

M E S P X W O R T M  
L K H U H R Y K Q O  
J G F A P T S P E S  
A N L E I T U N G D

# Dokumentation

Wort Uhr

Ulrich Radig

## Inhaltsverzeichnis

Einleitung .....	2
Software .....	2
Update der Firmware über das Web-Frontend .....	2
1. Voraussetzungen .....	2
2. Arduino-IDE vorbereiten .....	2
3. BIN-Datei erzeugen und hochladen.....	3
Funktionen der Uhrensoftware auf dem ESP .....	3
Standard Funktionen .....	3
Verbinden der Uhr mit dem eigenen WLAN.....	4
Zeitserver (NTP Server) .....	4
Automatische Helligkeitsregelung.....	4
Hardware.....	5
Pegelanpassung .....	5
Anschlussbeispiel .....	6
Aufbau.....	8
Buchstabenmatrix .....	8

## Einleitung

Diese Anleitung ist eine Sammlung von Informationen und kann unvollständig sein. Sie soll als Basis dienen, die Funktionen der Uhren Software zu beschreiben. Unter anderem enthält sie auch Beschreibungen vom Aufbau der Hardware und Tipps. Die Anleitung wird immer wieder erweitert und es steht jedem frei, Anpassungen zu machen.

## Software

### Update der Firmware über das Web-Frontend

Mit der Uhr-Version 2 ist es möglich, die Firmware OTA, das heißt über den in der Uhr eingebauten Webserver zu aktualisieren. Das ist in den meisten Fällen einfacher als der direkten Upload über die Arduino-IDE.

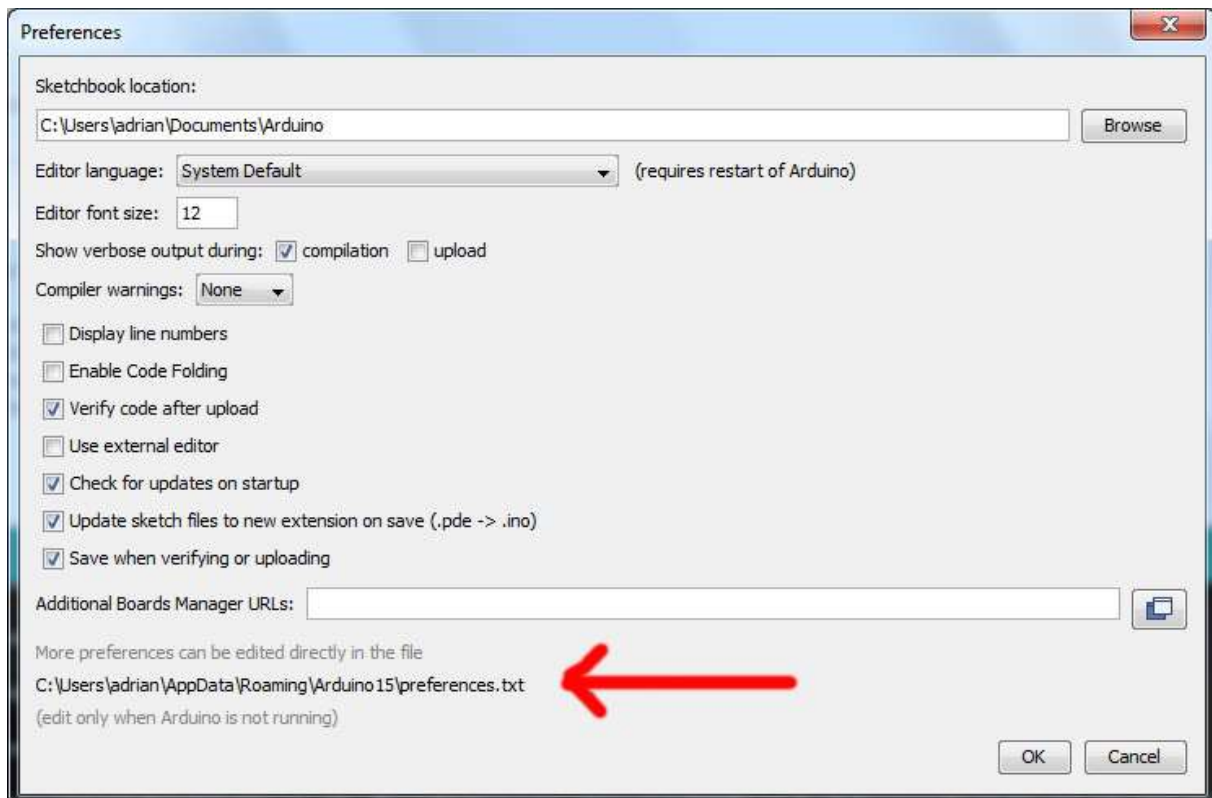
#### 1. Voraussetzungen

Voraussetzung für diese Vorgehensweise ist, dass die Uhr über WLAN erreichbar ist. Die Update-Schnittstelle erreicht man dann über die Adresse `http://<IP-Adresse der Uhr>:81/update` (Beispiel: `http://192.168.4.1:81/update`). Die Adresse der Uhr wird in der Grundeinstellung beim Start als Laufschrift ausgegeben. Hier kann nun die in der Arduino-IDE erzeugte BIN-Datei hochgeladen werden

#### 2. Arduino-IDE vorbereiten

In der Arduino-IDE ist es möglich, den Pfad zum Ablegen der kompilierten BIN-Datei, die die neue Firmware enthält anzugeben. Das Verzeichnis muss bereits existieren und kann beispielsweise über den Explorer angelegt werden (Beispiel: `C:\Users\xxx\Documents\Arduino\Bin-Ausgabe`).

Dieser Pfad kann nun in der Arduino-Preferences-Datei eingetragen werden. Dazu startet man die IDE und wählt <Datei>-<Voreinstellungen>. Oder <File><Preferences>. Nun auf die Zeile mit dem Pfad der Preferences-Datei klicken:



Es öffnet sich der Explorer mit dem Verzeichnis. Die Arduino-IDE muss nun beendet werden, dann öffnet man die Datei preferences.txt mit einem Texteditor.

Hier muss nun an beliebiger Stelle die Zeile `build.path=<Gewünschter Pfad>` eingefügt werden (Beispiel: `build.path=C:\Users\xxx\Documents\Arduino\Bin-Ausgabe`).

### 3. BIN-Datei erzeugen und hochladen

Startet man nun die Arduino-IDE und klickt auf die „Überprüfen“-Schaltfläche links oben, wird der Sketch kompiliert und im angegebenen Verzeichnis unter anderem eine .bin-Datei angelegt (Uhr\_Neu.ino.bin). Diese kann nun über die Adresse `http://<IP-Adresse der Uhr>:81/update` auf die Uhr übertragen werden.

## Funktionen der Uhrensoftware auf dem ESP

### Standard Funktionen

Es gibt 2 Möglichkeiten, die Software auf den ESP aufzuspielen. Im Verzeichnis "bin" des Github Repositories [https://github.com/Eisbaeeer/Ulrich-Radig\\_Wort\\_Uhr\\_Version\\_2](https://github.com/Eisbaeeer/Ulrich-Radig_Wort_Uhr_Version_2) befindet sich eine binary, welche mit einem beliebigen Tool direkt auf den ESP geladen werden kann.

Nach dem hochladen des Programmes, startet der ESP erstmalig im Modus Accesspoint. Die IP-Adresse wird als Laufschrift angezeigt. Die Uhr stellt ein WLAN mit dem Name (SSID) "Uhr" zur Verfügung. Das Kennwort lautet: 12345678

**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.

<b>WortUhr</b>	<b>11:01:30</b>	<b>Online</b>
Uhr	Effekte	Konfig

Startwerte speichern

WLAN speichern

SSID 
  
Passwort

**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.

<b>WortUhr</b>	<b>11:03:25</b>	<b>Online</b>
Uhr	Effekte	Konfig

Startwerte speichern

WLAN speichern

SSID 
  
Passwort

Zeitserver speichern

Zeitserver

**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.

<div>LDR Einstellungen speichern</div> <div>LDR für Helligkeitsregelung <span>Ja ▼</span></div> <div>LDR Kalibrierung <input type="text" value="30"/></div>
---

## Hardware

Für den Nachbau wird folgende Hardware/Software benötigt.

- NodeMCU (getestetes Board Lolin Version3)
- WS2812B oder WS2816 RGB LED Stripes
- Netzteil 5V 2A
- Arduino Version 1.8.5
- 470 Ohm Widerstand
- Diode oder Pegelanpasser
- Optional: LDR, 10 KOhm Widerstand

## 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 2. LED verstärkt sozusagen das Signal wieder auf die volle 5V.

Das heißt, die erste LED wird sozusagen als Pegelanpassung missbraucht.

(Bild1)

#### Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Power supply voltage	$V_{DD}$	+3.5~+5.3	V
Input voltage	$V_I$	-0.5~ $V_{DD}+0.5$	V
Operation junction temperature	$T_{opt}$	-25~+80	°C
Storage temperature range	$T_{stg}$	-40~+105	°C

**Electrical Characteristics** ( $T_A = -20 \sim +70^\circ\text{C}$ ,  $V_{DD} = 4.5 \sim 5.5\text{V}$ ,  $V_{SS} = 0\text{V}$ , unless otherwise specified)



## WS2812B

Intelligent control LED  
integrated light source

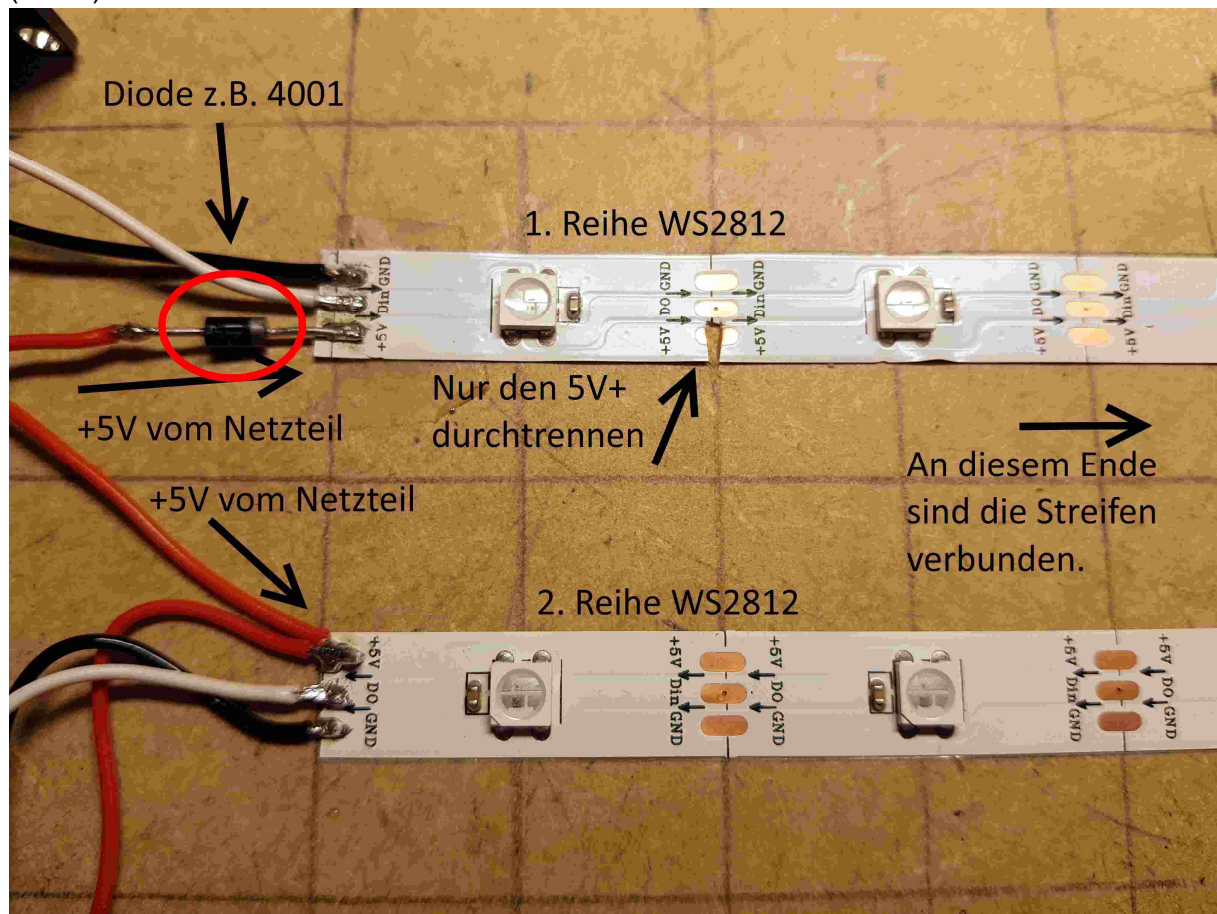
Parameter	Symbol	conditions	Min	Typ	Max	Unit
Input current	$I_I$	$V_I = V_{DD}/V_{SS}$	—	—	$\pm 1$	$\mu\text{A}$
Input voltage level	$V_{IH}$	$D_{IN}, \text{ SET}$	$0.7V_{DD}$	—	—	V
	$V_{IL}$	$D_{IN}, \text{ SET}$	—	—	$0.3 V_{DD}$	V
Hysteresis voltage	$V_H$	$D_{IN}, \text{ SET}$	—	0.35	—	V

### Anschlussbeispiel

Wie auf dem (Bild 2) 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!).

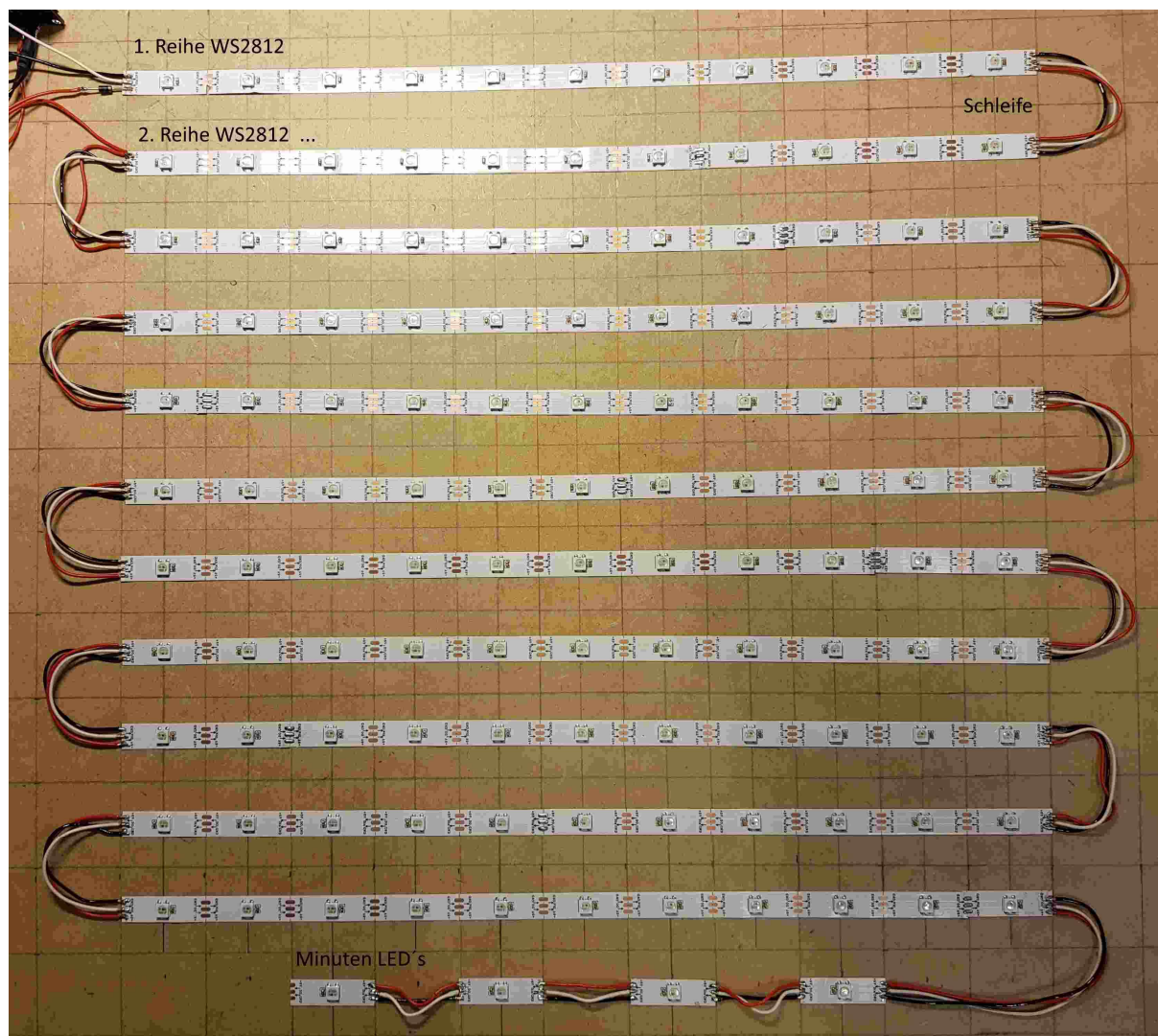
Am 2. Streifen wird dann zusätzlich vom Netzteil die +5V eingespeist. Die LED's vom 1. Stripe werden dann sozusagen "rückwärts" mit den nötigen +5V versorgt.

(Bild 2)



(Bild 3)





## Aufbau

### Buchstabenmatrix

Uhr Typ 114

E	S	K	I	S	T	L	F	Ü	N	F
Z	E	H	N	Z	W	A	N	Z	I	G
D	R	E	I	V	I	E	R	T	E	L
T	G	N	A	C	H	V	O	R	J	M
H	A	L	B	Q	Z	W	Ö	L	F	P
Z	W	E	I	N	S	I	E	B	E	N
K	D	R	E	I	R	H	F	Ü	N	F
E	L	F	N	E	U	N	V	I	E	R
W	A	C	H	T	Z	E	H	N	B	X
B	S	E	C	H	S	F	U	H	R	Y