WordClock mit der MobaLedLib

Die Entstehungsgeschichte

Anfang 2023 haben Hardi und ich überlegt, wie kann man die MobaLedLib mal ausreizen.

Bei mehreren Teams-Besprechungen mit unserem FRANK_TT hat Hardi immer an Frank vorbeigeschaut.
Man hat bemerkt, er ist in Gedanken. Doch was beschäftigt ihn so sehr??

Bis er meinte, so eine Uhr, wie Frank sie an der Wand hat könnte man doch mit Sicherheit auch mit der MobaLedLib machen.
Gemeint war keine normale Uhr, sondern eine auf die MobaLedLib zugeschnittene und mit dem Programm-Generator und Pattern-Generator programmierte Uhr.

Er hat mir dann von der Idee erzählt und ich war sofort Feuer und Flamme, denn an eine Wörteruhr (WordClock) hatte ich auch schon gedacht
und war am experimentieren mit einem anderen Projekt (WLED), was aber nur die Uhrzeit in Zahlen anzeigen kann, zwar auch in Laufschrift aber nicht so wie Franks Uhr.

Also haben wir angefangen zu probieren welche Module für uns in Frage kommen, ob auf Basis von Funkuhr oder Zeit über WLAN stellen usw. Doch das alles machte irgendwie Probleme.
Die Funkuhr zu stellen ging bei mir im Wohnzimmer nur mit einer ganz bestimmten Ausrichtung, WLAN wollten wir beide nicht, hat die MobaLedLib ja auch nicht.

Dann haben wir uns für das Uhrmodul DS3231 entschieden. Günstig und einfach. Zeit bringt der Compiler mit und wird beim Programmieren mit eingespielt.

Dann ging es weiter mit der Buchstabenmatrix, die sollte natürlich auch was ganz besonderes sein. Also haben wir fast 12 Wochen lang überlegt - wie sagt man zu den Zeiten in den verschiedenen Bundesländern? - was für Worte müssen unbedingt in der Matrix zusammenhängend wieder gegeben werden. Wir haben viel diskutiert und viele viele Entwürfe verworfen und von ganz vorne angefangen. - machen wir Sonderprogrammierungen (Geburtstage, sonst. Anlässe) - haben wir alle Buchstaben zumindest einmal in der Matrix - wie trennen wir Stunden und Minuten - zeigen wir nur 5 Minuten an oder wie können wir einzelne Minuten mit darstellen
Fragen über Fragen

Daraus resultierte diese Buchstabenmatrix. Unser Michael (raily74) hat diese dann so formatiert, dass die Buchstaben die bestmögliche Ausleuchtung haben.
Somit gibt es genau diese Schrifttypen nur bei uns in der MobaLedLib.

Und so sieht nun ein Beispiel aus (es gibt verschieden Schriften)



Wir hatten uns dann im Herbst 2023 in der Röhn zu einem kleinen Workshop mit ein paar Stummis getroffen.
Die Idee und Aufgabe des Treffens war, jeder Teilnehmer soll eine funktionsfähige und selbstgebaute WordClock mit nach Hause nehmen können.
Was Hardi und mich total positiv überrascht hat, ausnahmslos alle gebauten Uhren liefen von Anfang an ohne Probleme.
Keiner hatte einen Lötfehler in der Platine und auch nicht in der Verkabelung der Matrix.

Was kann die WordClock ??

Die Uhr kann die Uhrzeit in Wörtern anzeigen, sowie Termine (wenn eingepflegt), den Wochentag und die aktuelle Raumtemperatur (Temperatursensor erforderlich)
und kann auch auf Bewegung reagieren wenn ein Bewegungsmelder angeschlossen ist.
Der LDR dimmt die Helligkeit der RGBs, wenn der Raum Dunkler wird.
Je nach Bestückung ist dann ein 6/ 8 oder 10-poliger Wannenstecker erforderlich (6 PINs nur Uhr mit LDR, 8 PINs zusätzlich Thermometer und 10 PINs zusätzlich Bewegungsmelder)
Die Programmierung erfolgt über den Programm-Generator der MobaLedLib (ab Version 3.2.1) mit der im GitHub abgelegten Beispieldatei.

Stückliste der WordClock

alle für die Uhr benötigten Teile wollen wir in den Shop aufnehmen, so dass alle Teile aus einer Hand bezogen werden können.
Dazu auch die benötigten Gehäuse für die Elektronik und die Matrix.
Die aufgeführten Links und Lieferanten sind Beispiele, damit ersichtlich ist, welche Teile wir verwendet haben.

Steuereinheit / Platine:

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1	Platine	WordClock		werden wir im MLL-Shop anbieten
1	ESP32	ESP 32 mit **38PIN**	Aliexpress	AZ-Delivery ESP32 NodeMCU oder ESP-32 Dev Kit C V4
1	RTC Modul	RealTimeClock-Modul DS3231	Aliespress	RTC-Modul DS3231
1	Batterie RTC Modul	CR2032-Batterie		falls beim RTC-Modul nicht dabei
1	C4	Keramikkondensator, 100nF, RM 2.5mm	Z5U-2,5 100N	
1	C1	Keramikkondensator, 1µF	Z5U-5 1,0µ	
1	R13	Widerstand 10KΩ	METALL 10,0K	für LDR-Widerstand 5506
1	R1	Widerstand 4,7KΩ	METALL 4,70K	
1		Wannenstecker 6,8,10-polig		richtet sich nach Ausstattung der Uhr
1		USB-Mini Buchse Typ E	Aliexpress USB-Mini Buchse Typ E	zur Versorgung mit Fremdspannung, unbedingt nötig bei Bewegungsmelder
2		Buchsenleisten 19 Pin		für ESP32
1		Stiftleiste 6 Pin		für Modul RealtimeClock
1		Stiftleiste 4 Pin		für Modul RealtimeClock (optional)
1		WS2812 Breakout RGB-Modul		für Heartbeat
2		Buchsenleisten 3 Pin		für Breakout-Modul
3		Kurzhubtaster 6x6mm Höhe 4,3mm	Kurzhubtaster 6x6mm, Höhe: 4,3mm	**Achtung: kommen auf die Unterseite der Platine !!**
1		3D-Druck Gehäuse Elektronik		werden wir im MLL-Shop anbieten

Kabel zur Verbindung des Displays

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1,5 bis 1,8m		Flachbandkabel 6,8,10-polig	Reichelt/ MLL-Shop	Litzenanzahl richtet sich nach Ausstattung der Uhr
1		Postenstecker 6,8,10-Polig	Reichelt/ MLL-Shop	Litzenanzahl richtet sich nach Ausstattung der Uhr

Display / Matrix

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1		RGB LED Panel 16×16 WS2812B	Aliexpress 16x16 Matrix	
1		LDR Widerstand Typ 5506	Aliexpress LDR 5506	
1		Temperatur-Sensor DS1820 (Dallas)	DS 18S20	
1		Schrumpfschlauch	sollte in jeder Bastelkiste sein	
1		Buchstabenmatrix	Download auf GitHub Hardi S	werden wir versuchen im Shop anzubieten
1		Acrylglas-Scheibe mit Bohrungen	Acrylglas fertig konfiguriert - Kanten gelasert	Bohrung für Bewegungsmelder nicht berücksichtigt !!
4		Schrauben DIN7380 M3x10mm		Zur Befestigung vom Acrylglas
4		Mutten DIN933 M3		Zur Befestigung vom Acrylglas
10		Schrauben M2x4 Philips	Philips M2x4 selbstschneidend	Verschluss des 3D Druckgehäuses
1		3D-Druckgehäuse Ober- und Unterteil	Dateien im Github von Hardi	werden wir über den MLL-Shop anbieten

optionale Ausstattung der Platine mit Bewegungsmelder

Der Bewegungsmelder selber kommt in die Display-Matrix

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1		Bewegungsmelder MH-R602	Aliexpress MH-R602	kommt in das 3D Druckgehäuse, Acrylglas muss gebohrt werden
1	D1	BAT85 Schottky-Diode	BAT 85	
1	R2	Widerstand 10KΩ	Metall 10,0K	

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1	D2	Diode 1N4148	1N 4148	
1	R4	Widerstand 470Ω	Metall 470	
1	C3	Elko 100µF 25V	RAD 100/25	
1	Q1	MosFet N-Channel BS170	BS 170	ACHTUNG: es gibt 3 verschieden Bauarten – unbedingt vor Einbau testen
1	K1	Relais G5V1 5V	G5V-1 5DC	
1		Buchsenleiste 3 polig	optional	

optionale Ausstattung DCC auf Platine (noch Zukunft, da noch nicht in Software programmiert)

Anzahl	Bezeichnung	Beschreibung	Bestellnummer	Bemerkung
1	OK1	Opto-Koppler 6N137	6N 137	
2	R5,R7	Widerstand 1,0KΩ	Metall 1,00K	
1	R6	Widerstand 2,0KΩ	Metall 2,00K	
1	D3	Diode 1N4148	1N 4148	
1	DCC	Stiftleiste 2-Pin DCC-Anschluss		

Bestückung der Elektronik-Platine der Wordclock

Nachdem es nur wenige Bauteile sind, fangen wir wie immer mit der Bestückung bei den niedrigsten Bauteile an.
Dies sind hier die Widerstände und die Kondensatoren. Hier bitte auf die Werte und die richtigen Stellen achten.
Gefolgt wird die Bestückung dann mit der/ den Stiftleiste(n) für das Real-Time-Modul.

Beim Real-Time-Modul muss man die vorbestückten Pins auslöten und die Lötpads mit Entlötlitze sauber machen.

Sonst bekommen wir die Pins der Stiftleiste nicht rein.

Dann nehmen wir den ESP32, stecken die Buchsenleiten auf und stecken diese auf die Platine. Beim Verlöten bitte drauf achten das die Buchsenleiste komplett auf der Platine aufliegt. Gefolgt wird das Ganze von den Buchsenleisten für die Heartbeat.

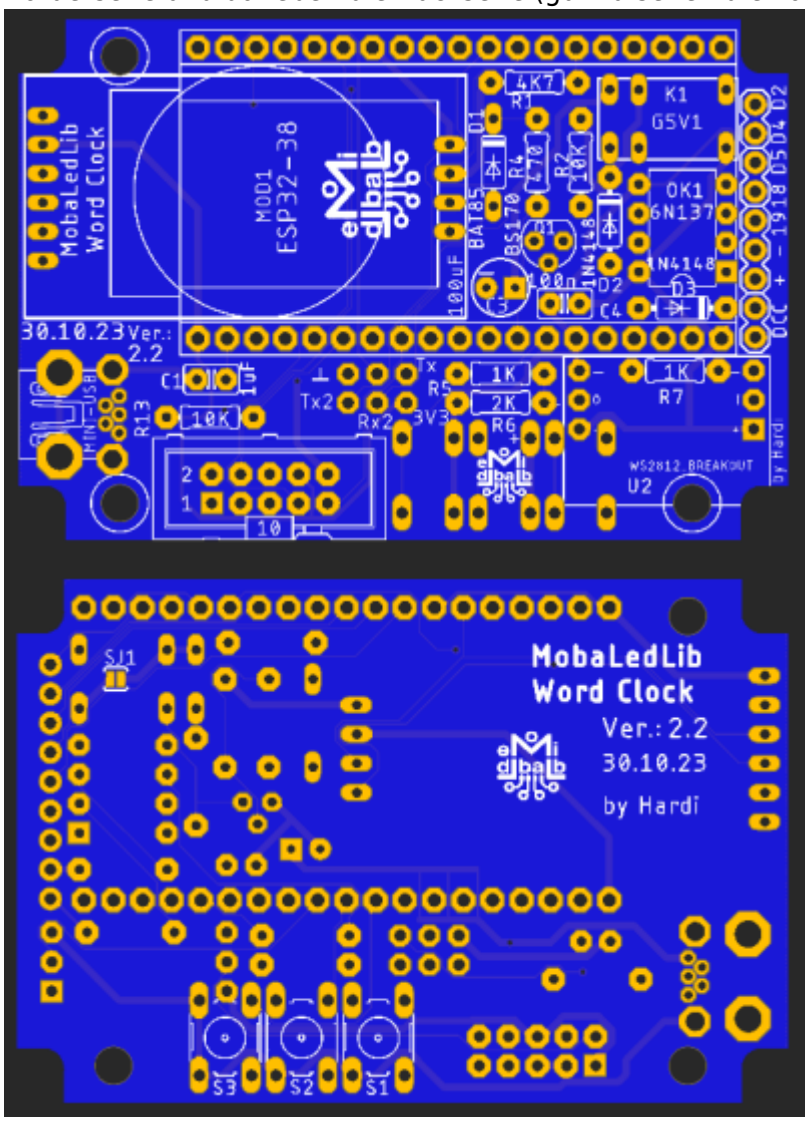
Jetzt kommt das RTC-Modul dran und wird zwischen den Buchsenleisten auf die Stiftleiste gedrückt und verlötet.

Nun noch den Wannenstecker einlöten, auch hier wieder drauf achten das er ganz auf der Platine aufliegt.

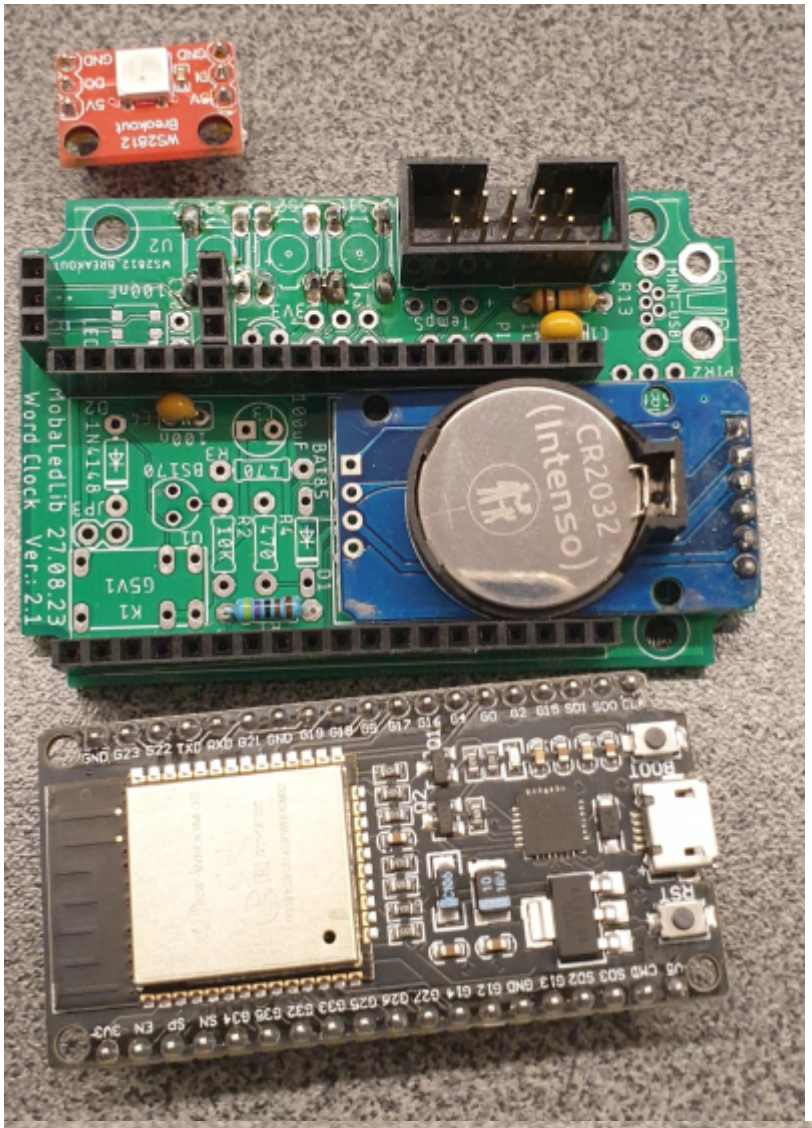
Zu guter Letzt kommen noch die 3 Taster dran.

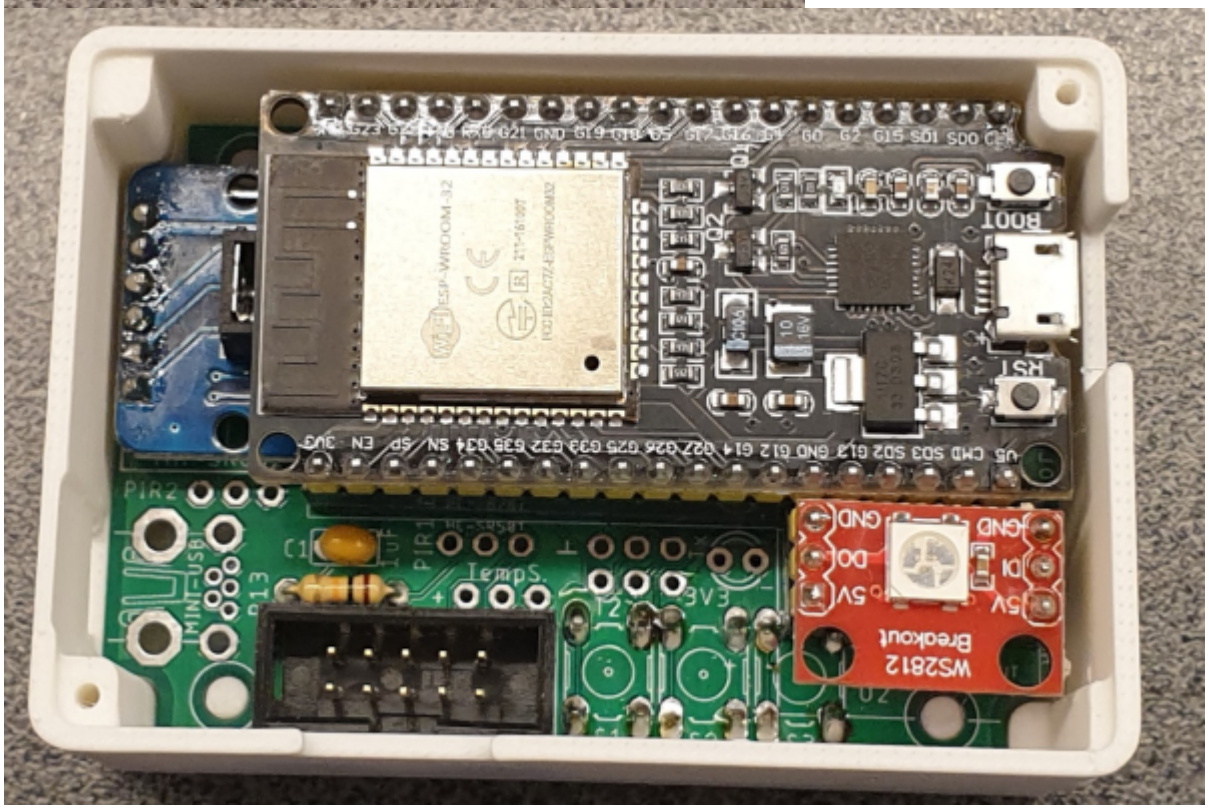
ACHTUNG: Die Taster kommen auf die Rückseite der Platine und werden auf der Oberseite Verlötet!!

Vorderseite und daneben die Rückseite (gut zu sehen die Taster auf der Rückseite)

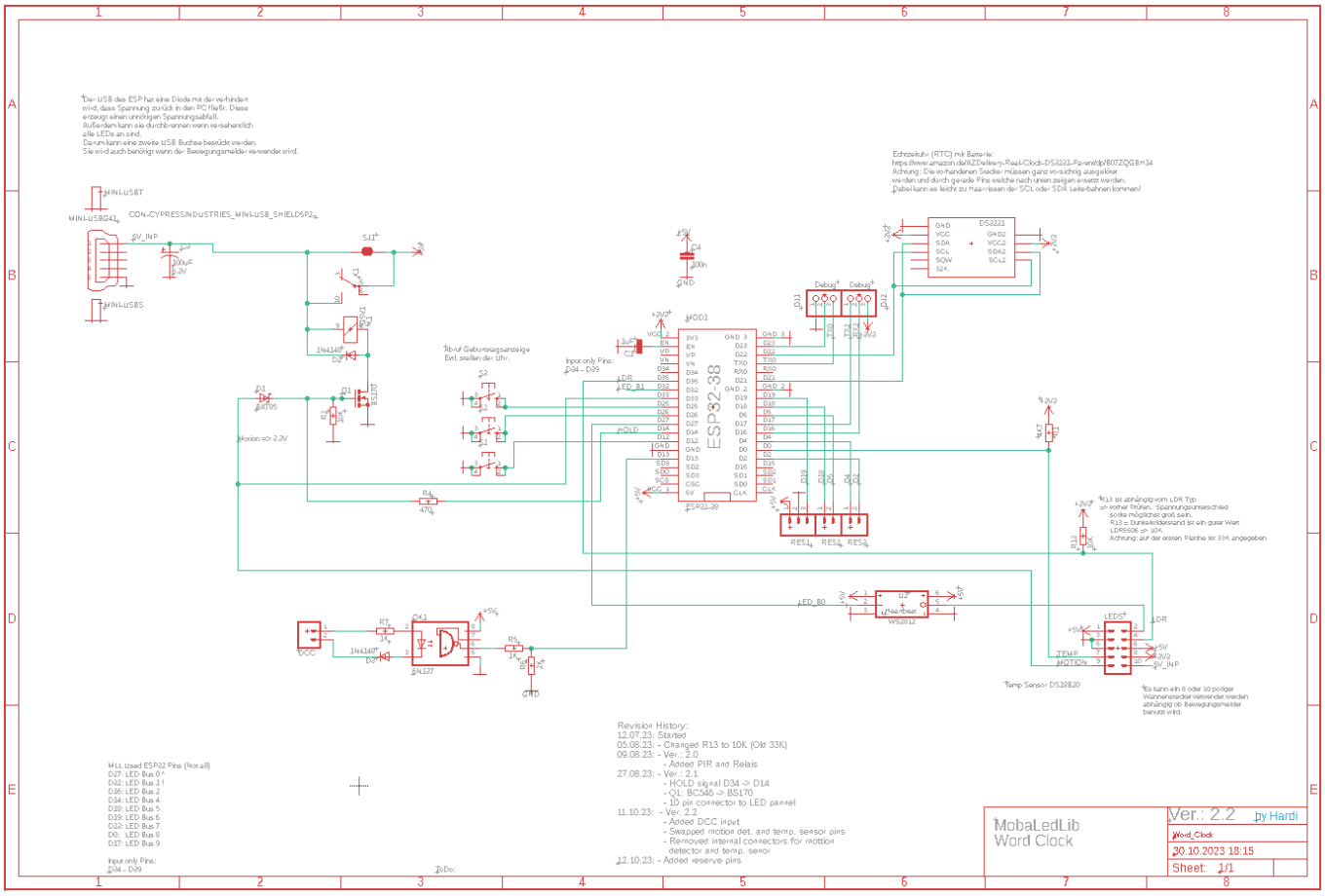


Hier ein Bilde der ersten von Felix (12 Jahre) und Nils (15 Jahre) gelöteten Platine (hier ist es noch die 1. Version)


WS2812 Breakout
MobaLedLib 27.08.23
Word Clock Ver.: 2.1
CR2032 (Intenso)
MINI-USB
PIR2
Temps
BOOT
RST

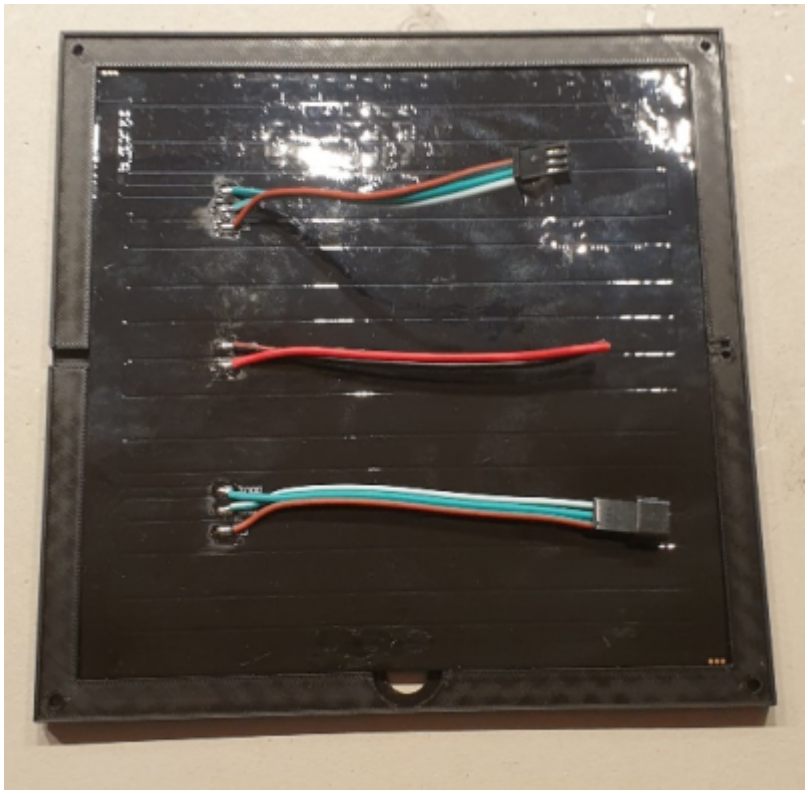

WS2812 Breakout
GND
DO
DI
5V
PIR2
Temps
MINI-USB
BOOT
RST

Schaltplan der WordClock



Verkabelung von RGB-Matrix im Gehäuse

Zunächst legt man die RGB-Matix 16×16 in den 3D-Druck Achtung die Position muss so sein, sonst passt es nicht !!

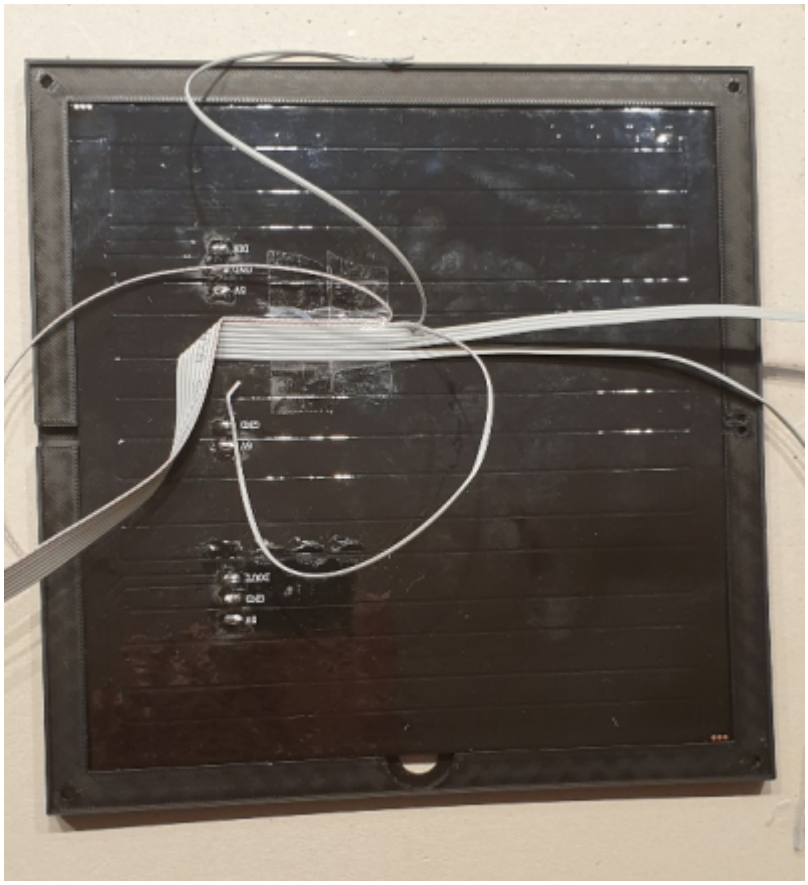


alle bunten Kabel werden nun abgelötet

Überprüfung ob alles passen wird (nach dem Ablöten Deckel drauf)



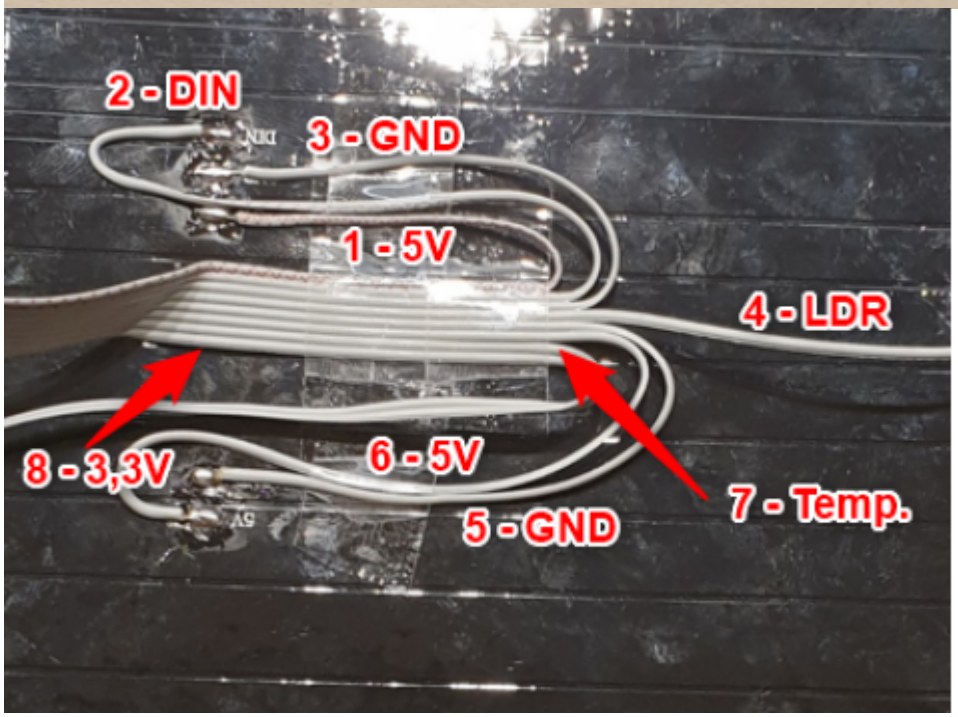
Verteilung der Kabel mit einfacher Fixierung durch Tesafilm
Wichtig: am Knick geht das Kabel aus dem Gehäuse raus zur Steuereinheit



Hier zur Info die Belegung des Kabels:
**Kabel 1 – +5V (rot markiert)
Kabel 2 – D-IN
Kabel 3 – GND
Kabel 4 – LDR
Kabel 5 – GND
Kabel 6 – +5V
Kabel 7 – Temperatur
Kabel 8 – +3,3V
Kabel 9 – Bewegungsmelder Motion)
Kabel 10 – 5V_Inp **

Grundverkabelung der Matix nach vorstehender Kabelbelegung, ohne Bewegungsmelder

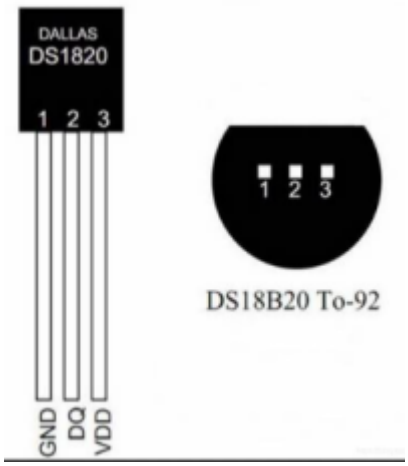

DOUT
GND
5V


2 - DIN
3 - GND
1 - 5V
4 - LDR
8 - 3,3V
6 - 5V
7 - Temp.
5 - GND

Dieses Kabelschema wurde so gewählt, damit es keine Kreuzungen gibt die innerhalb des Gehäuses unnötig auftragen und das spätere Verschließen erschweren würden.

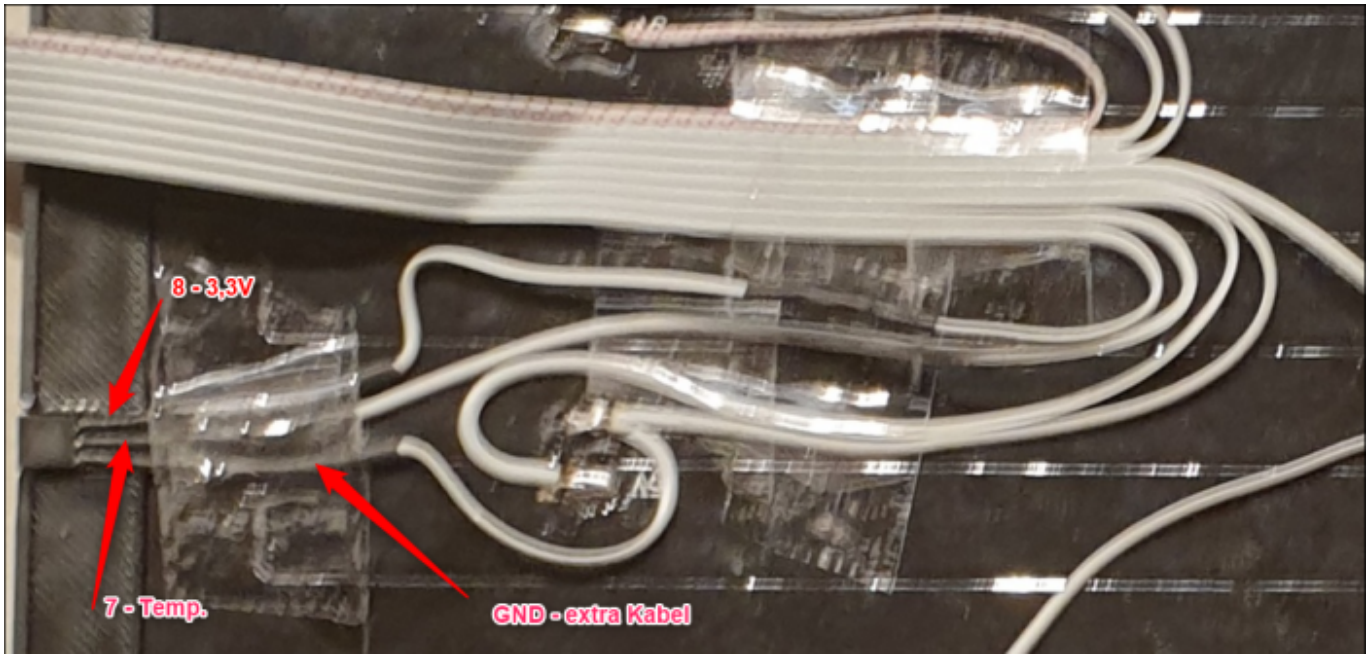
Nun kommen der LDR und der Thermo-Sensor an die Reihe

Der Thermosensor DS1820 hat folgende Belegung:



VDD bei diesem Sensor ist 3,3V – Versorgung über Kabel 8
DQ ist über Kabel 7 anzusprechen
GND des Sensors ist an GND auf der Matix angeschlossen

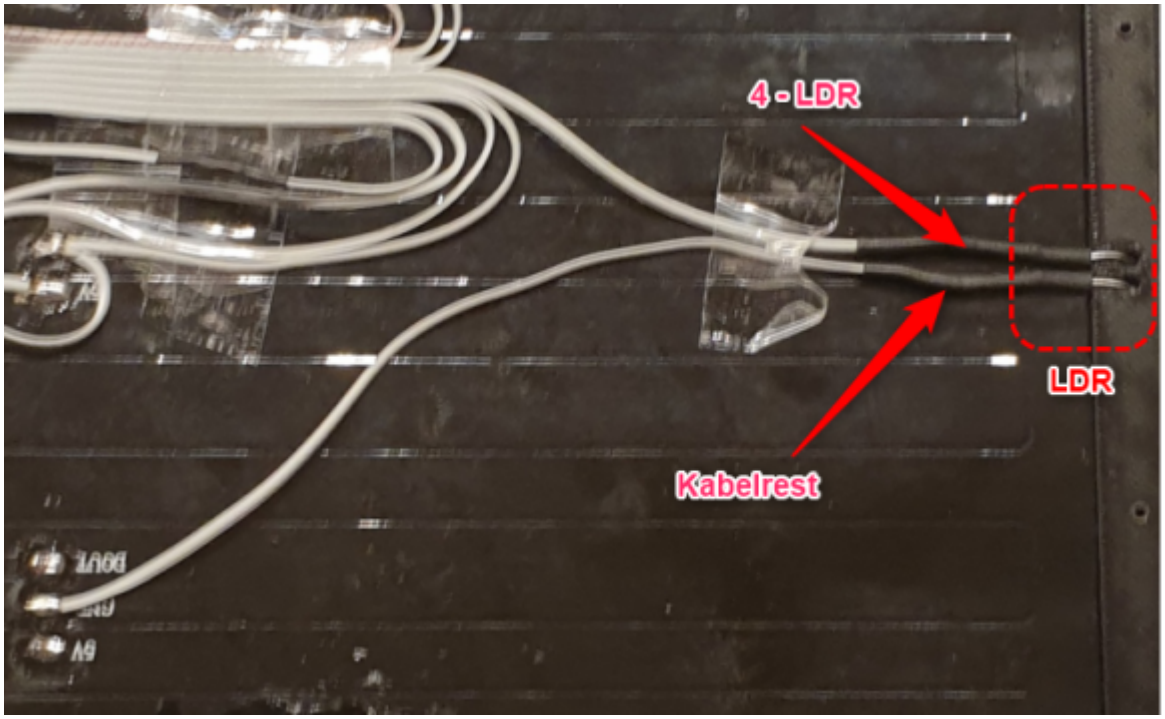
Der Einbau des Sensors erfolgt mit der flachen Seite nach oben, daraus ergibt sich folgendes Bild:



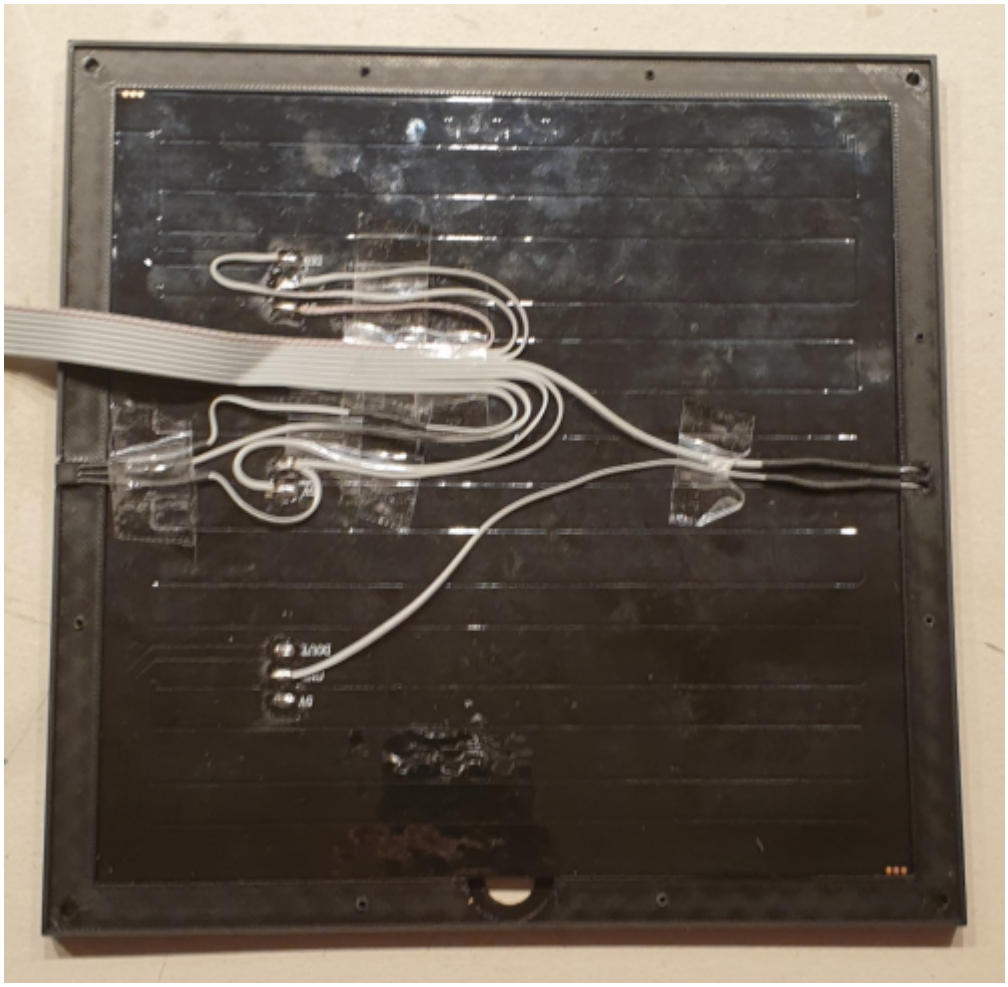
Sehr wichtig ist, den Sensor an der vorgesehenen Stelle nicht ins sondern ans Gehäuse zu setzen!! Der

Sensor nimmt sonst von innen die Temperatur mit und verfälscht den Wert

Die Verkabelung des LDR im Detail



Die komplette Verkabelung sieht nun so aus:



Nun kann der Pfostenbuchse an das Kabel gecimpt werden.

Achtung: die Pfostenbuchse hat einen kleinen Pfeil, dieser muss mit dem rot markierten Kabel 1 übereinstimmen, sonst ist die Pinbelegung falsch und es gibt Kurzschlüsse

Nun kann der hintere Deckel montiert werden.
Dazu das Kabel durch das Loch führen und nach unten in die Lasche (Zugentlastung) klemmen.
Nun den Deckel zuschrauben mit den M2x4 Philips-Schräubchen.

Jetzt kommt die geduckte Matrix dran. Hier sind die Löcher für die Schrauben und den LDR mit einer Lochzange auszustanzen.
Bei diesem Schritt das **A4-blatt noch nicht zuschneiden!!**

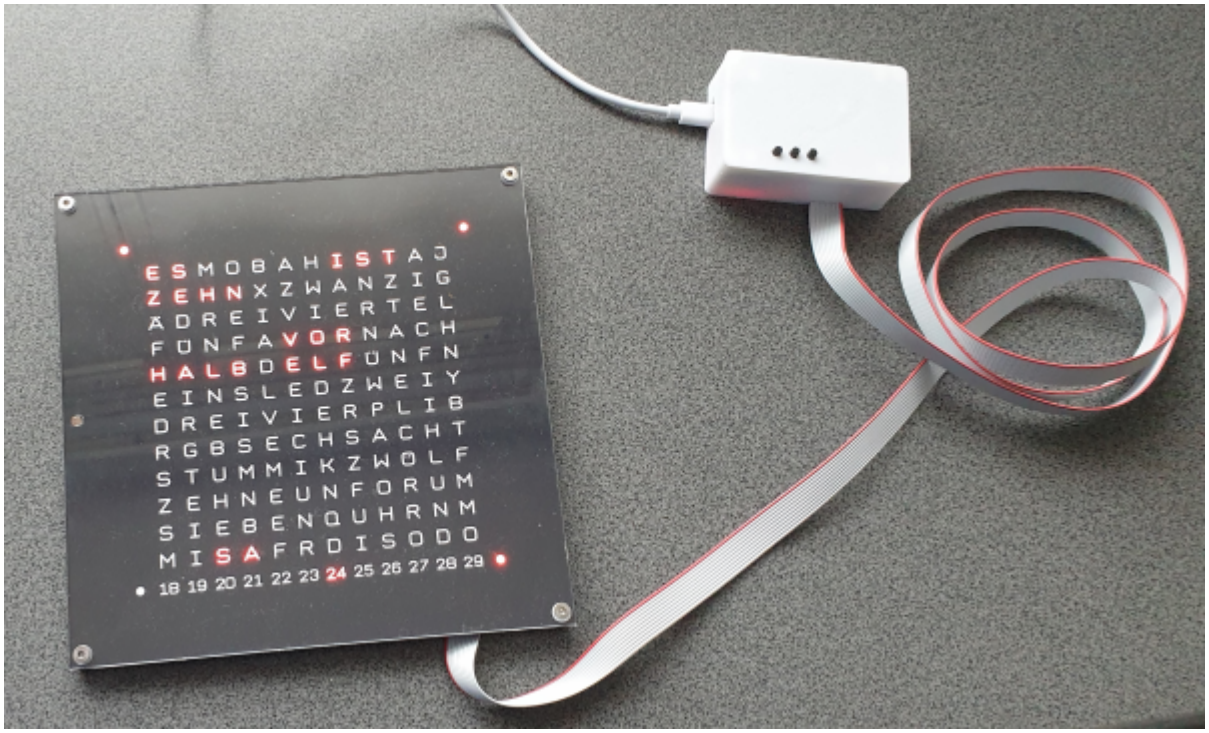
Nun beim Acrylglas die blaue Folie abziehen und vorsichtig zur Seite legen.

Als nächsten Schritt die Buchstabenmatrix auf die RGB-Matrix legen und nun das Acrylglas ganz oben aus, mit der unfolierten Seite auf die Matrixen.
Jetzt die DIN7380 M3x10mm Schrauben durch das Acrylglas, die Buchstabenmatrix und die RGB-Matrix stecken und mit den M3 Muttern fixieren.

Wir brauchen nun einen sehr scharfen Cutter (am Besten ein ganz neues Messer) und schneiden nun direkt am Acrylglas/ RGB-Matrix entlang die Buchstabenmatrix zurecht.
Dabei müssen wir auf der unteren Seite auf das Kabel aufpassen, nicht anritzen oder gar durchschneiden.

Jetzt noch den Stecker in die fertig gelötete Platine und es sollte was angezeigt werden.

So sollte die fertige Uhr mit der Elektronik im Gehäuse aussehen:



Programmablauf im Progamm-Generator und Bibliotheken

Hardi hat die MobaLedLib für die WordClock um einige Funktionen programmtechnisch erweitert, so wurden die Elemente / Bibliotheken

- WordClock_LDR_MLX (GitHub)
- WordClock_MotionDetect_MLX (GitHub)
- Word Clock_Text.MLX (GitHub)
- LED-Nummer definieren (MobaLedLib ab Vers. 3.2.1)
- Echtzeituhr (MobaLedLib ab Vers. 3.2.1)

in den Programmcode eingefügt.

Über die Arduino IDE muss unter Bibliotheken verwalten die

- Bibliothek **Streaming** von Mikal Hart, die neuere Version von Peter Polidoro kann auch installiert werden
- des Weiteren wird für den Temperatursensor Dallas DS1820 die Arduino-Bibliothek **DallasTemperature** (Miles Burton) benötigt, bitte alle vorgeschlagenen Bibliotheken wie **OneWire** mit installieren, sonst funktioniert der Sketch später nicht.
- die **RTCtime** (Fabiano Riccardi)
- die **Time** (Paul Stoffregen)
- die **TimerOne** (Paul Stoffregen)

Der LDR ist schon in der MobaLedLib-Bibliothek enthalten, braucht daher keine weitere Installation.

Eine Beispieldatei für den Prog-Gen. gibt es auf GitHub zum Download

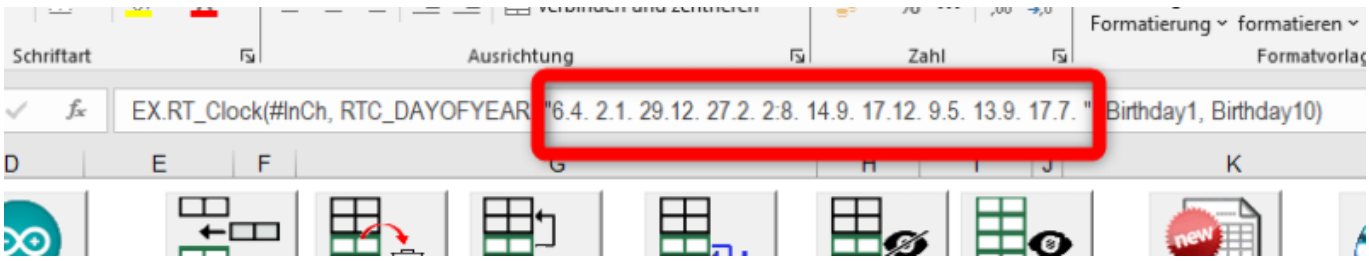
I0m Excel-Blatt des Prog.-Gen. haben wir versucht, die wichtigen Einstellungen gleich ganz an den Anfang des Programmablaufs zu stellen
Die Zeilen 18-26 definieren die einzelnen Anzeigezeiten der Termine/ Events
Für den Anwender beginnen die wichtigsten Zeilen mit der Zeile 47
Ab hier werden die anzuzeigenden Geburtstage definiert.
Hierbei ist wichtig das pro Block max. 10 Einträge möglich sind (wegen der Übersichtlichkeit so gemacht).

In der Zeile 47

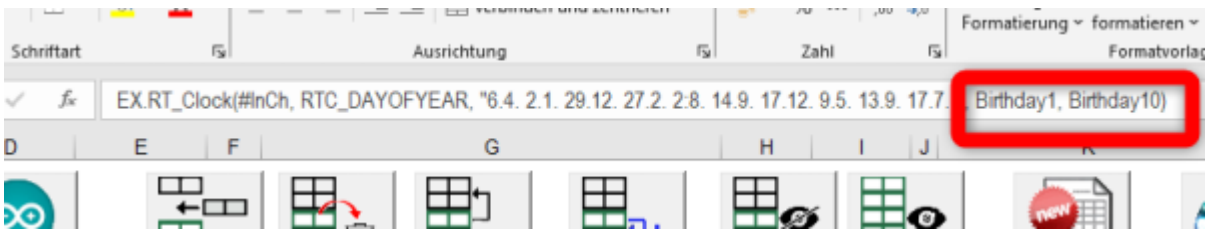


werden die „Geburtstagsdaten hinterlegt, sowie der Taster (SwitchD1) der Steuereinheit um die Geburtstage auch manuell aufrufen und durchklicken zu können.

In der Programmzeile sehen die Geburtstagsdaten so aus:

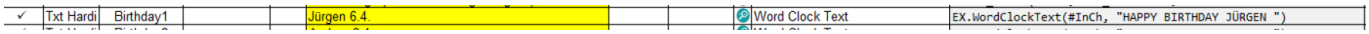


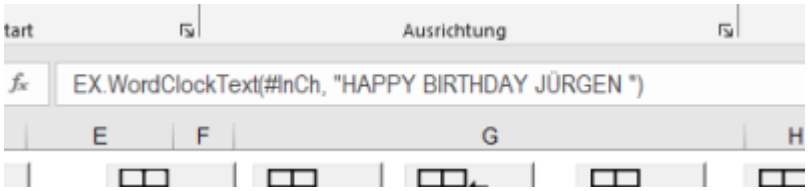
Des Weiteren werden in dieser Zeile auch die Variablen für den „Word Clock Text“ definiert:



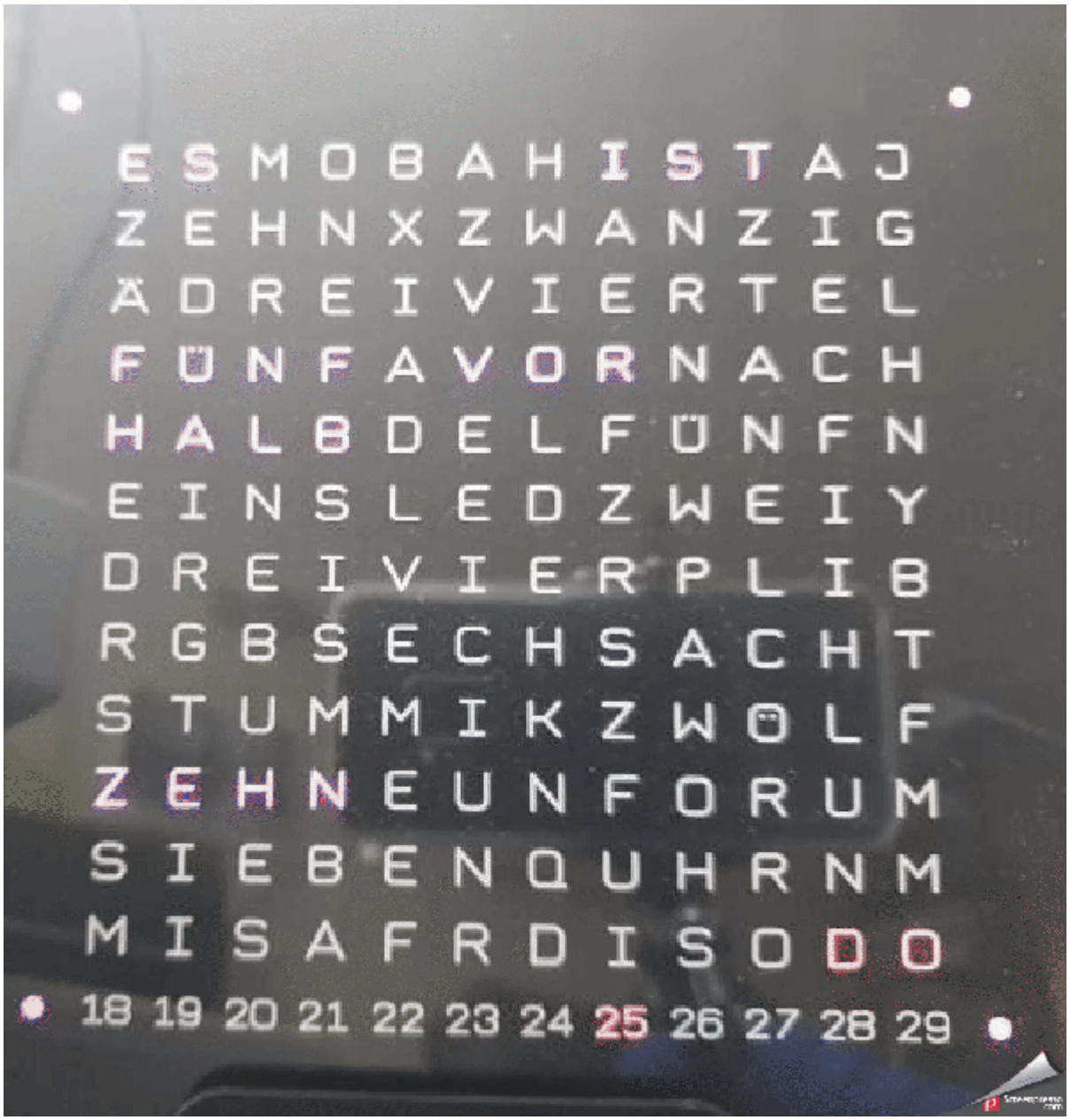
Ab der **Zeile 48** werden dann die Geburtstage den Variablen „Birthday1“ bis zum Beispiel „Birthday10“ definiert.
Hierbei muss man drauf achten das die in Zeile 47 definierten Daten ab hier in der Reihenfolge zugeordnet werden.

So hat in unserem Beispiel Jürgen am 6.4. Geburtstag und es soll der Text
„HAPPY BIRTHDAY JÜRGEN“
zufällig am 6.4. erscheinen, dann sieh die Zeile so aus:





So sieht es dann aus wenn der Text angezeigt wird **(Bitte auf das Bild klicken, damit die Effekte abgespielt werden)**



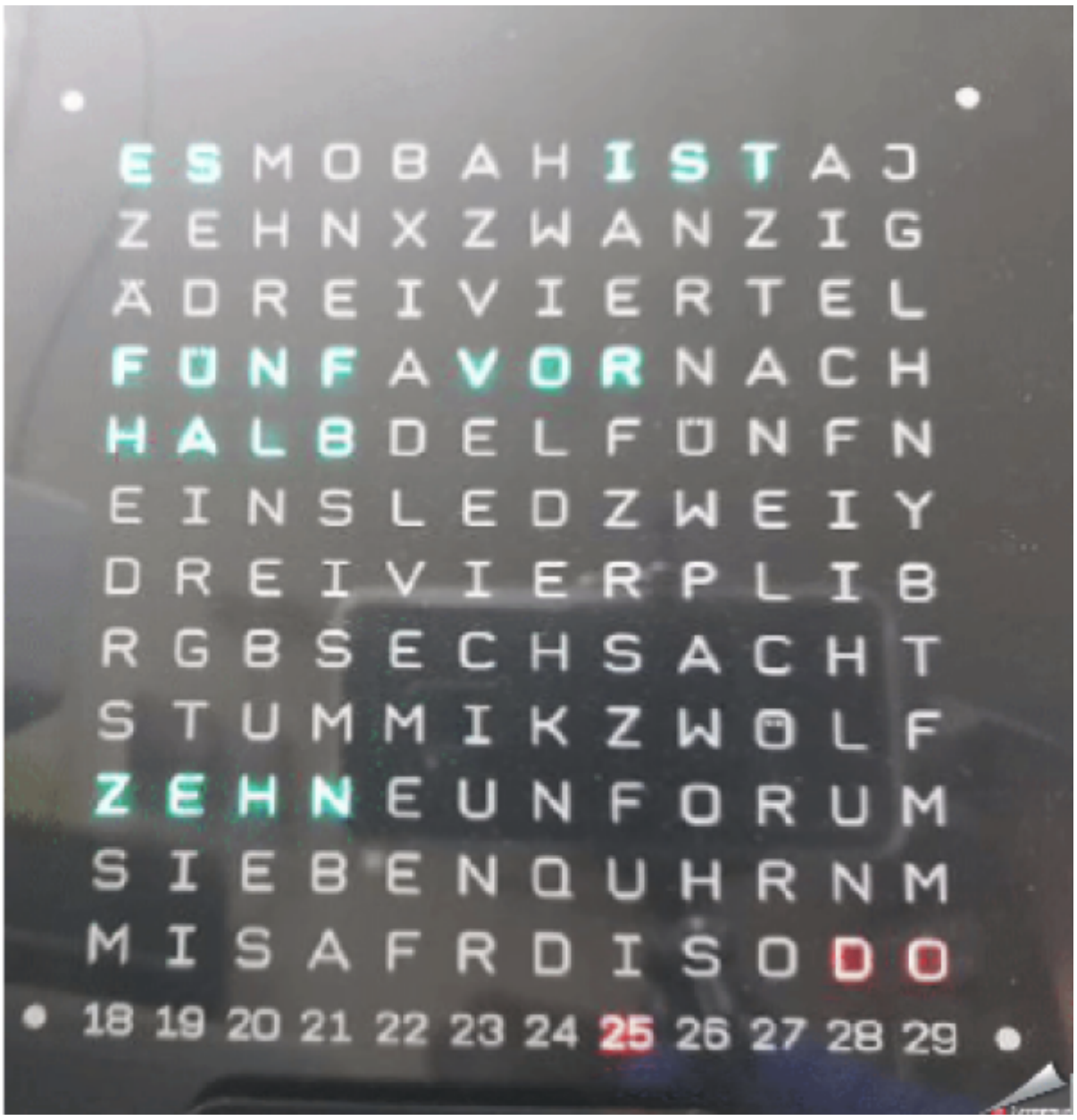
Sollte in einem Wort ein Buchstabe doppelt vorkommen, aber in der Matrix nur einmal enthalten sein, so wird dieser Buchstabe in einer anderen Farbe angezeigt. In diesem Beispiel das „P“ von „Happy“

Des Weiteren können .... und _ _ _ in den Texten verwendet werden.

Die Punkte im Wort lassen das Programm während der Darstellung in der Matrix auf den nächsten gleichen Buchstaben springen. Nimmt man ein „E(rich)“, so fängt er in der ersten Zeile ganz links an, möchte man aber das mittlere „E“, so muss man 3 Punkte von das E schreiben.
Somit werden die ersten 3 „E“ übersprungen.

Hat man einen Namen wie „Hans_Peter“ und der Name soll auf einmal auf der Matrix durchlaufen und nicht in 2 „Bildschirmen“ so arbeitet man hier mit dem Unterstrich zwischen den dazustellenden Wörtern.

Die Uhr in der Grundkonfiguration, wenn nur die Uhrzeit angezeigt wird sieht so aus:



Die unterste Buchstabenzeile sind die Wochentage und darunter sieht man die Temperaturanzeige.

Sollte die Temperatur höher oder niedriger als unser Bereich
in der Matrix sein, dann blinkt die Temperaturanzeige
entweder links (18) für tiefer oder rechts (29) für höher.

Die Punkte außerhalb sind die Minuten. Hier muss man etwas rechnen.
Wie in der Anzeige „ES IST FÜNF VOR HALB ZEHN“ und ein Punkt ist an, dann ist es 4 vor Zehn.

Zeigt die Uhr „ES IST FÜNF NACH HALB ZEHN“ an und zwei Punkte leuchten, dann ist es 7 nach halb Zehn.

Es müssen also die Punkte addiert oder subtrahiert werden.

Neue Version 1.0.1 der WordClock

Die WordClock hat softwarseitig einige Erweiterungen bekommen.
Seither konnte die Uhrzeit nur per neuem kompilieren des gesamten Programms korrigiert/eingestellt werden.
Dies konnte aber nur machen, wer die entsprechenden Dateien auf seinem Rechner hat.
Nachdem aber Hardi und Fromue einige der Uhren auch an Freunde verschenkt haben, hatten die nun keine Möglichkeit irgendwie
auf die Software zuzugreifen. Doof, denn wenn die Pufferbatterie im Steuergehäuse leer ist, ist die Zeit „verloren“, da sich das
RTC-Modul diese sich dann nicht mehr merken kann.

Daher wurden einige neue Dinge in die Software eingebaut:

- WLAN-Verbindung zum Router per WPS-Taste
- Uhr speichert bis zu 5 verschiedene Netzwerke und erkennt diese auch wieder
- Umstellung der Zeitzonen per Webinterface
- die StatusLED im Steuergehäuse zeigt farblich die Netzverbindung an
- Steuermenü über die serielle Schnittstelle (z.B. Arduino IDE), hier sind verschiedene Eingaben möglich:

```
Hier die wichtigsten Befehle:
- Eingabe " ? " im seriellen Monitor leitet auf die Hilfe mit
Befehlsübersicht
- " list " ruft die Variablen auf wie gespeicherte Netzwerke und Zeitzone
- " clear " hiermit können die Variablen gelöscht werden
(Auslieferungszustand herstellen)
- " restart " hiermit wird ein Neustart der Uhr ausgelöst (notwendig nach
Neuverbindung der Uhr mit einem Router um Webinterface zu aktivieren)
- " datetime " Möglichkeit die Uhrzeit und das Datum auf manuelle Weise zu
stellen
- " ssid " und " pwd " ermöglichen die manuelle Eingabe der WLAN-
Zugangsdaten, sollte kein WPS-Taster vorhanden sein, z.B. WLAN auf einer
```

```
Messe
- " scan " Suche und Status der Netzwerke rund um die Uhr
```

Anmerkung: Die WordClock lies sich auch schon mit der vorherigen Software per Serieller Schnittstelle stellen.

Die WordClock lässt sich derzeit nur mit PlatformIO im Programm-Generator kompilieren!!
Es werden die weiter unten aufgeführten neuen Bibliotheken der Version 1.0.1 benötigt!!

Bibliotheken für die MobaLedLib Version 1.0.1

Hier sind neue Bibliotheken für die WordClock Version 1.0.1 hinterlegt:

zum Dateidownload

(Stand der Dateien 06.Januar 2026)

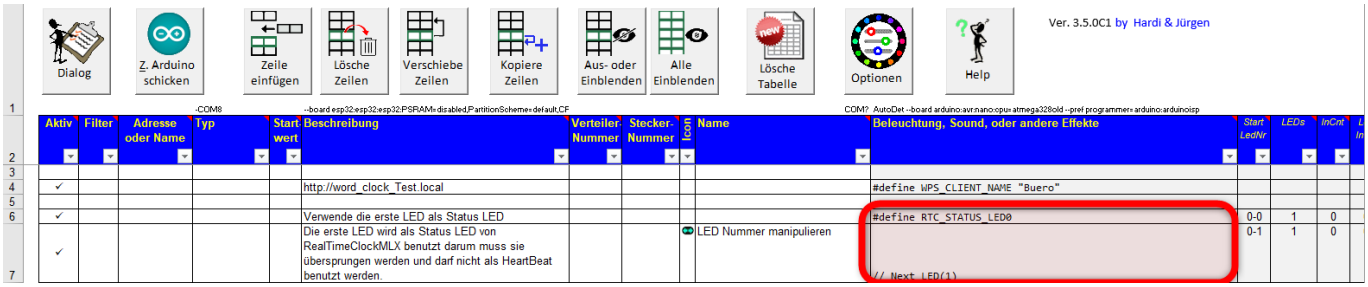
Wie man die PlatformIO installiert erklären wir noch. Aktuell hier mal der Link zum vorläufigen Skript für die PlatformI0-Installation:
https://github.com/Hardi-St/RealTimeClockMLX/blob/Wifi-Test/PIO%20Files.zip

Bedeutung der Farben der RGB im Steuergehäuse:

Wir haben der RGB im Steuergehäuse zur Unterscheidung der verschieden WLAN-Stadien unterschiedliche Farben zugewiesen:

- **orange** = Verbindungsaufbau
- **blau** = WPS-Taste am Router drücken
- **grün** = WLAN verbunden, Zeit ist synchronisiert

Diese erste RGB ist nur dann Status-RGB, wenn sie nicht als Heartbeat definiert ist.
Dazu werden anstatt der Heartbeat in der ersten Zeile, die erste und zweit Zeile mit folgenden Befehlen gefüllt:



Funktion der Zeile 4 wird beim Punkt Web-Interface erklärt!!

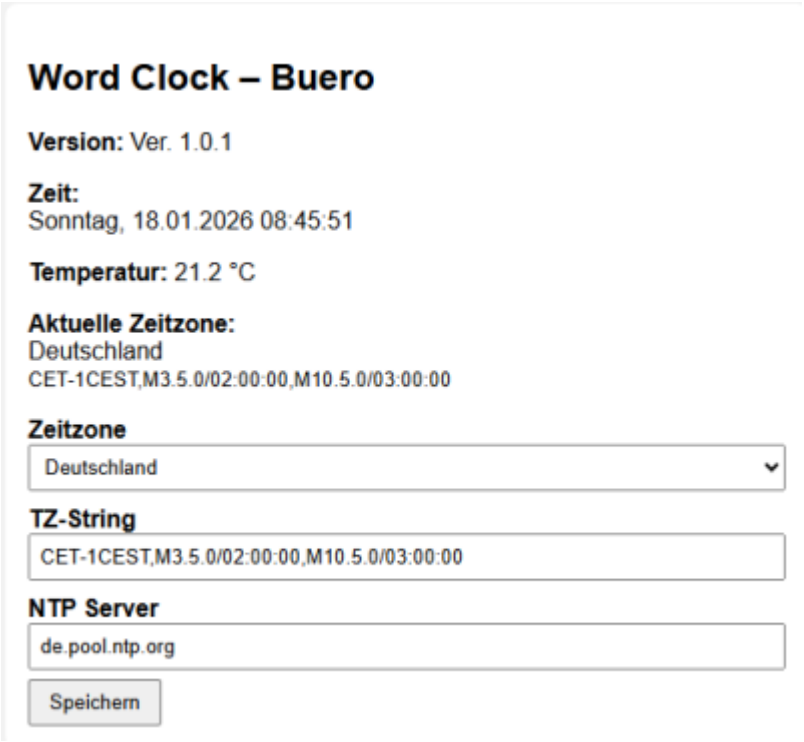
Lässt man die Heartbeat ganz normal in der Zeile und fügt
nichts hinzu, werden die zwei LEDs
oben im Display verwendet. Sie leuchten aber nur kurze Zeit.

Funktion des Web-Interface

Der Gedanke eines Web-Interface kam zunächst um den WLAN-Status der Uhr abzufragen und zu sehen ob sich die Uhr
und mit welchem Netzwerk sich die Uhr verbunden hat.

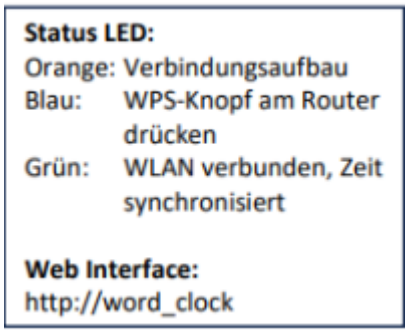
Es hat sich dann aber herausgestellt das es ein ganz lustiges und nettes Gimmick ist um im Heimnetzwerk die Uhrzeit und Temperatur
der WordClock zu sehen.
Zudem bietet das Webinterface die Möglichkeit die Zeitzone, falls jemand z.B. in England nutzen möchte, entsprechend einfach umzustellen.
Hierzu gibt es die Zeilen „Zeitzone, TZ-String und NTP-Server“. Nach Änderung auf speichern klicken und die Uhr stellt sich entsprechend um.

So sieht das Web-Interface aus:



Das Web-Interface wird im Internetbrowser mit http://word_clock aufgerufen

Ums sich diese ganzen Dinge einfacher merken zu können, gibt es einen Aufkleber, den man auf die Steuereinheit kleben kann:

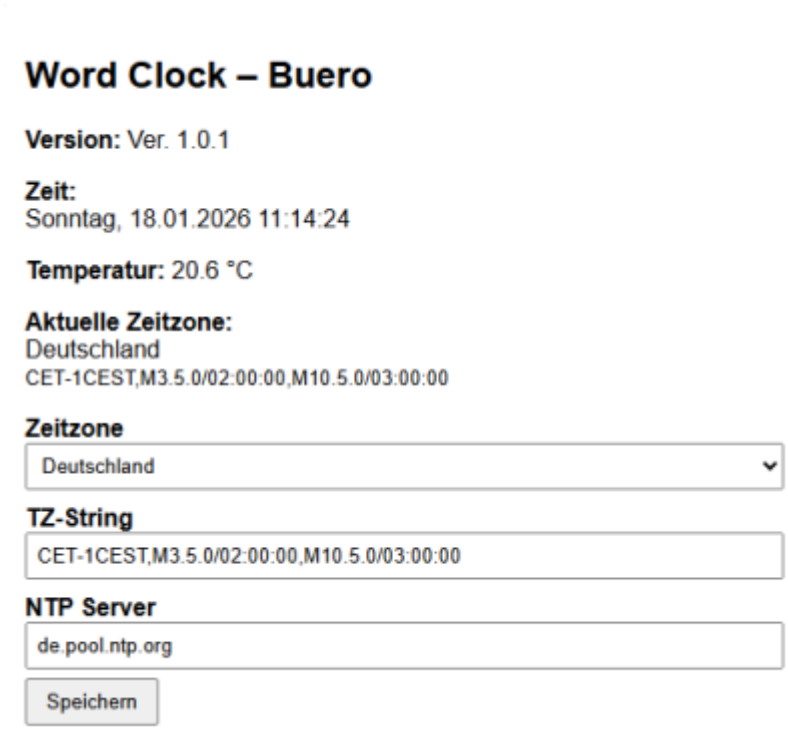


Hier die Datei:

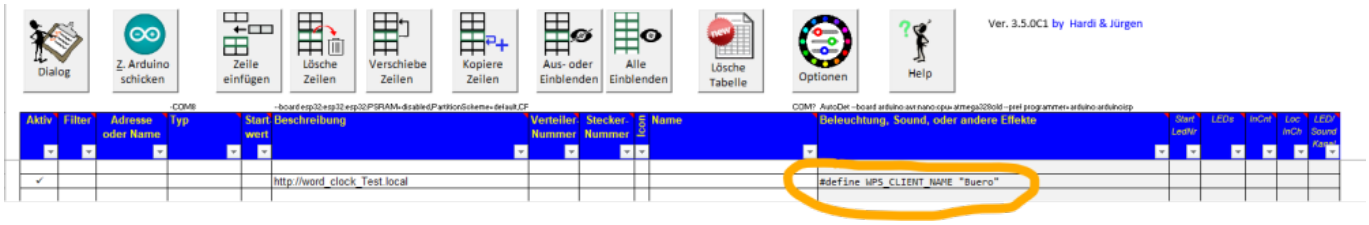
aufkleber.pdf

Unterscheidung mehrerer WordClocks im Web-Interface

Hat man mehrere WordClocks, dann kann man im Programm-Generator über ein #define auch den Namen erweitern, das dann z.B. WordClock - Buero angezeigt wird:



Hierzu fügt man im Programm-Generator ganz oben folgendes #define ein:
#define WPS_CLIENT_NAME „Buero“



??

Für Buero kann natürlich alles mögliche eingesetzt werden (Wohnzimmer, Kinderzimmer, Bad usw.).
Nun muss natürlich als Web-Aufruf **http://word_clock_buero** aufgerufen werden.

Dateien für die MobaLedLib (erste Version) mit Druckdateien für die Gehäuse

Die zur Installation und Fertigstellung des WordClock-Projektes sind hier folgende Dateien zusammengetragen:

- **eine Dokumentation** als PDF-Datei
- **die nötigen Arduino-Bibliotheken** (diese in das Verzeichnis Dokumente – Arduino – libraries kopieren)
- **Beispieldateien für den Prog_Generator** mit Anleitung
- **3D-Druckdateien für die Gehäuse der Elektronik und der Matrix**

Bitte beachten, die 3-Druckdateien sind als .step-Dateien hinterlegt.

Man braucht für gute Druckergebnisse nach Möglichkeit einen 3D-Drucker der mehrfarbig drucken kann.

Die für die WordClock benötigten Dateien geht es hier

zum Dateidownload
(Stand der Dateien ist der 21.04.2024)

Nachlese

//Im Programm-Generator gibt es noch viele viele Zeilen hinterlegt, die für den Programmablauf sorgen, aber hier enden wir mit der Erklärung.
Wir denken es wird niemand diese Programmierung nachvollziehen wollen.

Euch nun viel Spaß beim Nachbau der Uhr.
Bitte denkt dran, die Software ist funktionsfähig, aber dennoch in der Weiterentwicklung, je nachdem wie es zeitlich passt!!

Wir keinen Einfluss auf die Bibliotheken oder gar die Arduino IDE und somit auf die zukünftige Kompatibilität dieses Projektes

<color #22b14c>**An dieser Stelle nochmals herzlichen Dank an Michael (raily74) für seine tatkräftige Unterstützung.**
Ohne seine Hilfe wär das Projekt nicht so toll geworden.

© Dieses Projekt wurde durch Jürgen (fromue) zur Verfügung gestellt, 21.04.2024

Weitere Details von Frank (fbstr)

Nachdem fbstr die Wordclock nachgebaut hat, sind ihm noch ein paar Kleinigkeiten aufgefallen, welche von mir in der Anleitung nicht erwähnt wurden.
Er hat das ganze in der Tageszeitung unseres Stummi-Forums gepostet.

Hier geht´s direkt zum Beitrag in Forum

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
https://wiki.mobaledlib.de/anleitungen/bauanleitungen/workshop/wordclock?rev=1768768746

Last update: **2026/01/18 20:39**