

DIY Wordclock

Christof Pfannenmüller

1 EINFÜHRUNG

WORDCLOCKS sind in der Bastler oder DIY-Szene weit verbreitet und bergen eine gewisse Faszination. Auch wenn es für solche Wordclocks schon eine Vielzahl an Bauanleitungen gibt, kann trotzdem beim Bau einer solchen viel über die Programmierung und die Steuerung durch Mikrocontroller gelernt werden. Grundsätzlich sind alle Wordclocks sehr ähnlich aufgebaut. Fast immer wird eine Frontplatte mit passend angeordneten Buchstaben verwendet, die durch eine Matrix an LEDs von hinten erleuchtet ist. Dadurch, dass nur die zur aktuellen Stunde passenden Wörter leuchten, stechen diese aus den, auf den ersten Blick verwirrend angeordneten, Buchstaben der Frontplatte hervor. Wichtig bei einer Wordclock ist auch, dass jede LED nur den für sie vorgesehenen Buchstaben der Frontplatte beleuchtet und nicht zu einem Teil auch die benachbarten Buchstaben. Dies würde den absetzenden Effekt des Hervorscheinens der Buchstaben von den dunkleren minimieren. Dafür wird meist eine sogenannte Zwischenplatte angebracht, die das Licht der LED abschirmt, damit sicher gestellt werden kann, dass kein Nachbarbuchstabe beleuchtet wird. Wordclocks wurden bereits in vielen verschiedenen Sprachen erstellt, wobei grundsätzlich jede Sprache möglich ist. Die einzige Vorgabe ist alle Wörter, die zum „schreiben“ der Uhrzeit notwendig sind, auf der Frontplatte anzugeben. Dabei ist jedoch wichtig, dass die Reihenfolge der Wörter so gewählt wird, dass jede mögliche Uhrzeit auch dargestellt werden kann ohne ein Wort doppelt unterbringen zu müssen und trotzdem die Rei-



Abbildung 1. Die Frontplatte meiner Wordclock

henfolge der Wörter dem Sprachgebrauch entspricht. In diesem Zusammenhang ist etwa die Darstellung der Zeit 15:45 interessant, denn je nach Wohnort wird diese in unterschiedlichem Wortlaut verwendet. Vor allem in Sachsen, Thüringen und Franken wird die Bezeichnung „Dreiviertel vier“ verwendet, in Norddeutschland dagegen eher „Viertel vor vier“. Solche Feinheiten müssen also berücksichtigt werden, bei „Viertel nach drei“ bzw. „Viertel vier“ verhält es sich ähnlich.

2 ANFORDERUNGEN

Diese Wordclock sollte vor allem - im Gegensatz zu den meisten anderen - kleiner ausgelegt und nicht zum Aufhängen an der Wand ge-

dacht sein, sondern als modulare kleine Tischuhr, die bei Einbau einer Alarmfunktion, auch als Wecker dienen kann. Bei ersten Recherchen fand ich heraus, dass der Hersteller Qlocktwo mittlerweile eine solche Uhr in seinem Sortiment hat. Qlocktwo ist allem Anschein nach der einzige Hersteller von Wordclocks auf dem Europäischen Markt. Da der Hersteller auch Ersatzfrontplatten verkauft, entschied ich mich zuerst eine solche zu kaufen und eine eigene Frontplatte erst zum Schluss zu erstellen, wenn die Wordclock bereits funktionsfähig ist. Dadurch musste ich zwar die geplante Größe der Wordclock um ein paar Zentimeter auf 13,5cm Seitenlänge verkleinern, konnte mich aber erst einmal auf die Elektronik konzentrieren. Um zunächst einen Überblick zu bekommen sammelte ich alle möglichen Projektideen und mögliche Umsetzungen in einem, mit Latex erstellten, Lasten- und Pflichtenheft. Dabei wurde auch die in Latex integrierte Bibliographieverwaltung verwendet um alle Quellen für Ideen zu archivieren. Zur Versionskontrolle wurde während des Projektes für alle Dateien Github eingesetzt um jeweils den aktuellen Stand zu protokollieren. Das fertige Github-Repository ist unter der <https://github.com/pfanchri/Wordclock> zu finden.

3 HARDWARE

Eagle

Für die Platine wird Eagle als PCB-Software zum Entwurf der Platine verwendet, da es kostenlos verwendet werden kann und die Platinenentwicklung im Fablab auf Eagle ausgerichtet ist. Eagle arbeitet nach dem Bibliotheksprinzip, das heißt es werden für jedes Bauteil Daten über die elektrischen Anschlüsse und über das Package, also über die Verpackung des Bauteils, benötigt. In Eagle können somit zuerst der Schaltplan, also die elektrischen Verbindungen und Netze, erstellt werden. Diese werden in einem zweiten Schritt auf der Platine angeordnet und die Leiterbahnen geroutet. Hier kann leicht berücksichtigt werden, dass viele elektrische Bauteile in unterschiedlichen Packages zur Verfügung stehen. Die bei

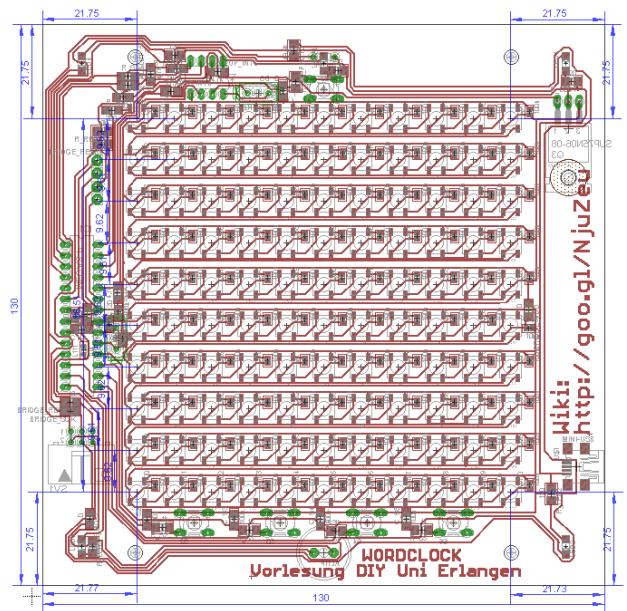


Abbildung 2. Das Layout in Eagle

der Wordclock verwendeten Bauteile waren zumeist in den mit Eagle ausgelieferten Bibliotheken bereits enthalten. Die Bibliotheksdaten für die LEDs waren zwar nicht in den Standardbibliotheken enthalten, werden dafür aber von Adafruit, einem Händler für diese Bauteile, bereitgestellt. Im Layout wurden zuerst alle LEDs an die durch die Frontplatte vorgegebenen Stellen platziert. Anschließend wurden die Taster angeordnet, sodass diese später optisch ansprechend platziert sind. Alle in der fertigen Wordclock nicht mehr sichtbaren Bauteile wurden als letztes platziert und mit Leiterbahnen versehen. Eagle konnte auch eingesetzt werden, um etwa eine Stückliste der verwendeten Teile auszugeben.

Elektronik

Das Ziel ist die Platine einlagig aufzubauen. Dies erhöht zwar den Aufwand für das Layout immens, dafür kann aber eine einlagige Platine verwendet werden, die wesentlich einfacher zu entwickeln ist, als eine zweilagige. Auch werden keine Durchkontaktierungen in der Platine benötigt, die im Fablab etwas aufwändiger sind. Da der Platz aber durch die Vorgabe der Frontplatte begrenzt ist, müssen auch kleine Bauteile verwendet werden, die etwas schwieriger zu löten sind. Versorgt werden soll die

Uhr mittels Strom aus einem USB-Ladegerät oder Computer damit kein besonderes Ladegerät notwenig ist. Dies legt die Versorgungsspannung auf 5 Volt fest. Als Stromanschluss wird eine kombinierte Micro-USB-Buchse, die für Stecker des Types A und B geeignet ist, verwendet. Die Uhr kann dadurch mit einem Ladekabel für Handys betrieben werden. Die Hauptarbeit übernimmt dabei ein Prozessor der Mega8-Reihe von Atmel, da dieser einfach zu programmieren und leicht verfügbar ist. Genau wird der Atmega168PA-PU verwendet, der den größten programmierbaren Flash seiner Baureihe zur Verfügung stellt. Da ein späterer Einbau einer Batteriepufferung möglich sein soll, verwenden wir hier die Pico-Power Ausführung des AVR, da diese weniger Strom im Schlafmodus verbraucht, was hierbei hilfreich sein kann. Als Baugröße verwende ich hierbei die PDIP Ausführung . Da bei diesem Gehäuse die Pins nahe beieinander liegen, kann der AVR gut zwischen der LED-Matrix und dem Rand der Platine eingebaut werden. Hierfür wäre eine andere Gehäuseform zu groß. Zum Programmieren des AVR wird eine Standart ISP-Stecker verwendet, da bereits damit zu rechen ist, das die Software mehrfach aufgespielt werden muss, bis diese reibungslos läuft. Die LED-Matrix zum Anzeigen der einzelnen Buchstaben der Uhr wird durch 110 WS2812B LEDs aufgebaut. Diese LEDs haben den Vorteil, dass sie eine eigene Datenleitung haben und der Mikrocontroller daher nur einen Pin bereitstellen muss, um alle hundertzehn Buchstaben ansteuern zu können. Dies funktioniert, indem ein, in die LED eingebauter Chip, dieses Datensignal empfängt und über den Data-Out-Pin der LED die folgenden LEDs mit dem Datensignal versorgt. Alle drei Grundfarben können durch den Chip in 255 Stufen bei den LEDs getrennt geregelt werden. Damit kann also mit jedem der 110 Buchstaben das gesamte Farbspektrum von 16777216 verschiedenen Farben dargestellt werden. Dem Datenblatt der WS2812B zufolge sollte bei jeder WS2812B ein Kondensator mit 100nF zwischen Versorgungsspannung und Ground im Layout eingebaut werden. Dies dient dazu die Versorgungsspannung der LED zu stabilisieren

und somit sicherzustellen, das die Ansteuerung richtig funktioniert. Da für die Versorgung des eingebauten Steuerchips der WS2812B auch im ausgeschalteten Zustand Strom benötigt wird, kann die Gesamte Matrix über einen P-Kanal-Mosfet von der Versorgungsspannung getrennt werden. Für die LEDs, welche die einzelnen Minuten und den Alarm anzeigen, werden dagegen „normale“ LEDs verwendet, deren Kathode vom Mikrocontroller auf Ground gezogen werden können und damit angeschaltet werden. Um keine ungewollten Aktionen beim Programmieren des Atmega auszulösen werden die LEDs auf die selben Pins des Mikrocontroller gelegt, die auch als Verbindung zum In-System-Programmer dienen. Beim Programmieren des AVR kann es also nun vorkommen, dass die LEDs aktiviert werden, es kann jedoch kein Schaden an Bauteilen erzeugt werden, obwohl diese Pins doppelt genutzt werden. Die Uhrzeit soll sowohl im Mikrocontroller selbst als auch in einem externen Realtime-Clock-Baustein gezählt werden können. Der externe Baustein hat hierbei wieder die Vorteile, dass der Atmega168 im Schlafmodus sein kann und somit wenig Strom benötigt wird. Als Clock-Baustein wird hierfür ein DS1337 verwendet, da dieser zwei externe Interrupt Kanäle besitzt und somit den Mikrocontroller durch einen Alarm aus dem Schlafmodus aufwecken kann. Die Real-Time-Clock wird vom AVR über das Two-Wire Protokoll angesteuert werden. Für das Einstellen der Uhrzeit und das Nutzen anderer Funktionen, wie etwa eines Alarms, wurden fünf Taster verbaut. Alle diese sind mit einem Kondensator entprellt und jeder Taster ist am AVR angeschlossen, sodass ein Tastendruck mit einem Pin-Change-Interrup detektiert werden kann. Außerdem wurde ein LDR (Light Dependent Resistor) also ein lichtabhängiger Widerstand eingebaut. Somit kann zu einem späteren Zeitpunkt die Helligkeit der Matrix in Abhängigkeit von der Umgebungshelligkeit gesteuert werden. So wird sichergestellt, dass die LEDs niemals blenden, was vor allem für den Einsatz als Wecker wichtig ist, aber diese auch zu jeder Tageszeit gut erkennbar sind. Damit die Uhr auch in der Lage ist ihre Benutzer zu wecken, wurde ein Signalgeber eingebaut,

der über einen internen Generator verfügt und somit nicht mit einer Sinusschwingung ange- regt werden muss, sondern nur durch anklem- men an die 5V Versorgungsspannung aktiviert werden kann. Dies ist durch den Mikrokontrol- ler wesentlich leichter zu realisieren. Die zwei nicht verwendeten Pins des Atmega wurden auf ein Via an einer freien Stelle der Leiterplat- te gelegt, außerdem wurde hier noch ein Via für Ground und +5 Volt eingefügt. Somit kann hier zu einem späteren Zeitpunkt auf einfache Weise eine Lösung zur Spannungsversorgung aus einer Batterie einbaut werden. Diese könnte sogar mit dem AVR kommunizieren und die Wordclock könnte dadurch in einen stromspa- renderen Modus wechseln.

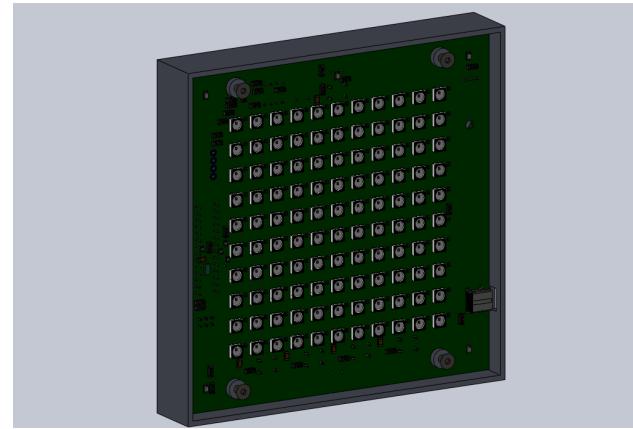


Abbildung 3. Das erstellte Modell mit der eingebauter Elektronik

Gehäuse

Um das Gehäuse der Wordclock planen zu können, erstellte ich zuerst ein CAD-Modell. Als Programm wurde dafür SolidWorks verwendet. Dies wurde auch vor dem Hinter- grund gemacht, damit aus den in Solid Works erstellten Dateien beispielsweise die Schnitt- muster für die Zwischenplatte extrahiert wer- den können. Da Eagle die Möglichkeit besitzt, das Layout der Platine im Intermediate Data Format (IDF) zu exportieren, welches für den Austausch von Layout und CAD-Programmen gedacht ist, konnte damit in SolidWorks ein genaues 3D Modell der Platine erstellt werden. Die einzelnen CAD-Modelle der elektrischen Bauteile dafür sind im Internet zu finden. So- mit konnte, aufbauend auf die Platine mit den angebrachten Bauteilen, ein gutes Gehäuse ge- plant und modelliert werden. Für den Proto- typ wurde das Gehäuse mit dem Onlinetool „Boxmaker“ aus Akryl ausgelasert. Die Zwi- schenplatte, die dazu dient die einzelnen LEDs gegeneinander abzuschirmen wurde ebenfalls zuerst in SolidWorks erstellt und anschließend mit dem Lasercutter aus einer MDF Platte. Die Dicke der Zwischenplatte ergibt sich aus dem Abstrahlwinkel der LEDs, dieser beträgt bei den WS2812B 120°. Für die Dicke der Zwi- schenplatte ergibt sich also mit einer Buchsta-

bengröße von etwa 8mm:

$$d = \frac{\frac{1}{2} * \text{Buchstabendurchmesser}}{\tan \frac{120^\circ}{2}} = 4,62 \quad (1)$$

Aus diesem Grund wurde eine Dicke von 5mm für die Zwischenplatte gewählt. Zusammen- gehalten wird die gesamte Uhr von vier M3 Schrauben, welche die Platine, die Zwischen- platte und das Gehäuse mit Abstandshalter fi- xieren, damit diese nicht verrutschen.

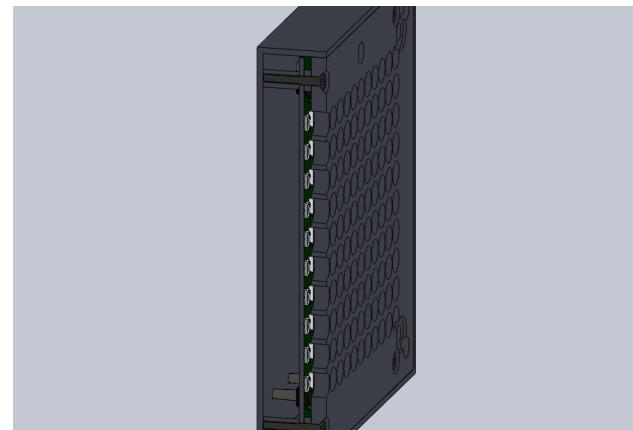


Abbildung 4. Schnitt durch das in SolidWorks erstellte Modell

Frontplatten

Für die Frontplatte wurde die Anordnung der Buchstaben in einer Illustrator-Datei erstellt. Dabei wurden alle Buchstaben so angeordnet, dass diese exakt über das entsprechende Loch in der Zwischenplatte passen. Für die erste Anordnung der Buchstaben wurde das selbe

Schema, welches auch bei der Qlocktwo und vielen weiteren Wordclocks im Internet zu finden ist, angewendet. Außerdem wurde eine Frontplatte erstellt, die es ermöglicht die Wordclock auch als Binäruhr zu verwenden. Hierfür

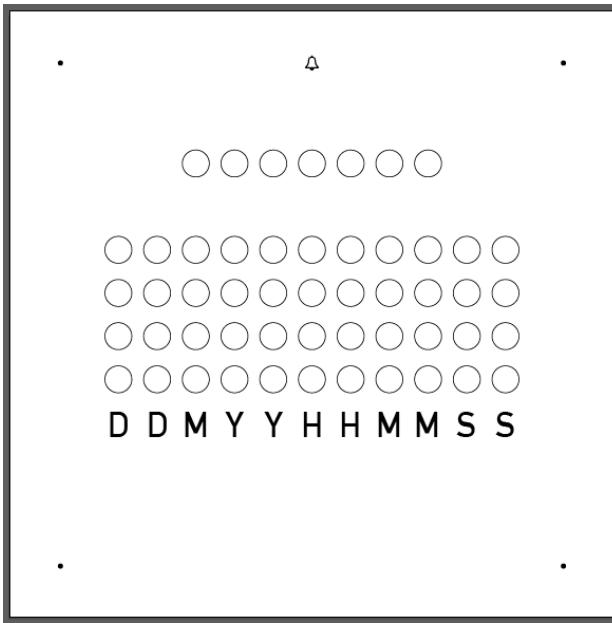


Abbildung 5. Eine Binäre Frontplatte für die Wordclock mit Anzeige für Wochentag (oben), Datum und Uhrzeit

muss nur die Software entsprechend angepasst werden. Da die Frontplatte idealerweise nur an den Buchstaben das Licht durchlässt und dort leicht milchig sein sollte, gibt es verschiedene Möglichkeiten zum Herstellen der Platte. Die simpelste Möglichkeit ist dabei, das Layout der Frontplatte auf mehrere Overheadfolien zu drucken und diese mit einem Blatt Papier als Diffusor gemeinsam zu laminieren. Problematisch ist hierbei allerdings die Verbindung der einzelnen Folien, da diese sich leicht lösen. Bei der binären Frontplatte wurden nur die Lichtdurchlässe aus der Holzplatte ausgeschnitten, die hierfür verwendet wurde. Diese wurden anschließend von hinten mit Papier verklebt, sodass die LEDs durch dieses diffus hindurchleuchten. Als aufwendigste Methode wurde eine Akrylscheibe von hinten zuerst farbig und anschließend mehrfach schwarz lackiert. Aus dieser wurde das in Illustrator erstellte Frontplattenlayout dann ausgraviert. Die Schwarze Lackierung dient hierbei dazu, dass die LEDs nicht durch die farbigen Stellen der Frontplat-

te hindurch leuchten. Eine weitere, aber nicht erprobte Methode, wäre das Ausschneiden einer Folie mit dem im Fablab verfügbaren Folienplotter, diese würde anschließend auf eine Milchglasscheibe geklebt. Alle Frontplatten sind mit einem Magnet an der Zwischenplatte befestigt, so wird kein Werkzeug zum Austauschen der Zwischenplatte benötigt.

4 SOFTWARE

Die Software für den Atmega168 wurde in Eclipse entwickelt. Der wesentliche Teil der Software ist die Ansteuerung der Matrix aus 110 WS2812B LEDs. Diese funktioniert über ein spezielles Protokoll: die erste LED der Matrix bekommt einen Strom an Bits vom Mikrocontroller. Von diesem Bitstrom entfernt die erste LED die vordersten 24 Bit und gibt alle weiteren ankommenden Bits an die nächste LED weiter, die nach dem selben Prinzip verfährt. Somit erhält jede LED 24 bit, von denen jeweils 8 die Helligkeit der drei Farben Grün, Rot und Blau repräsentieren. Die 8 Bit kodieren dabei die Helligkeit der jeweiligen Farbe in 256 Stufen. Die Übertragung zu den WS2812B erfolgt durch ein unipolares Non-Return-To-Zero-Signal. Eine 1 wird dabei durch einen nur 0,2us längeren Impuls als eine 0 übertragen; Aus diesem Grund ist, die Ansteuerung sehr zeitkritisch und Interrupts sollten während des Senden an die LED-Matrix deaktiviert sein. Zur

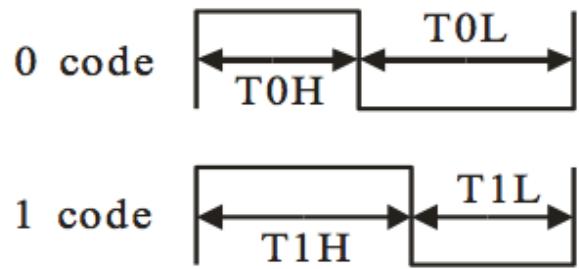


Abbildung 6. Ausschnitt aus dem Datenblatt der WS2812B

Ansteuerung wird die light_ws2812 V2.3 von cpldcpu verwendet, die auf Github unter der GNU General Public License veröffentlicht ist. Diese Bibliothek übernimmt vom Hauptprogramm einen selbst definierten Datentyp, der

für jede Farbe ein Array mit allen 110 LEDs der Matrix enthält. Den LEDs können somit getrennte Helligkeitswerte zugewiesen werden. Um Probleme bei der Ansteuerung der letzten LED zu umgehen, wurde das Array jedoch um eines länger gewählt, als die Matrix eigentlich ist. Dies führt dazu, dass die letzte WS2812B noch immer Daten weiterreichen muss und somit die Wahrscheinlichkeit durch einen Bitfehler eine falsche Helligkeit anzuzeigen sinkt. Die Verbindung zum DS1337 Baustein erfolgt, wie bereits erwähnt über das Two-Wire-Interface und verwendet dabei die eingebaute Peripherie des AVR. Softwareseitig baut die Kommunikation auf einer Bibliothek auf, die eigentlich für den Baustein DS1307 gedacht ist. Diese wurde von Davide Gironi ebenfalls unter der GNU General Public License zur Verfügung gestellt. Da sich die beiden Bausteine in Ihren Registern für die Stunden, Minuten und Sekunden sowie denen für das Datum aber entsprechen, funktioniert diese Library auch um das Datum im DS1337 zu speichern und aus diesem auch zu lesen.

5 AUSBLICK

Bisher noch nicht umgesetzt wurde die Einstellung der aktuellen Uhrzeit mit den fünf zur Verfügung stehenden Tasten. Dafür soll entweder die Möglichkeit des AVR genutzt werden einen Pin-change-Interrupt bei einem Tastendruck zu generieren. Alternativ könnte in der Hauptschleife ständig abgefragt werden, ob die Taste gerade gedrückt wurde. In einer späteren Version der Wordclock wäre es auch noch wünschenswert, wenn die Taster beschriften werden würden. Dies wäre auf dem Gehäuse ohne Probleme mit dem Lasercutter zu realisieren. Da die Bibliothek für die Kommunikation mit der Realtime-Clock aktuell die Verwendung der im Baustein DS1337 verfügbaren Alarne nicht unterstützt müsste die Kommunikation noch um einige Funktionen erweitert werden, damit auch die Alarne des Bausteins genutzt werden können. Intern nutzt die Bibliothek für die Konfiguration der Uhr einige Funktionen zum Senden und Empfangen von Daten über das Two-Wire-Interface. Diese könnten sicher



Abbildung 7. Foto der fertigen Wordclock

auch bei der weiteren Anpassung der Bibliothek an den DS1337 Baustein verwendet werden. Weitere Features wie der Empfang der aktuellen Zeit über den Sender DCF77, also der Umbau zu einer Funkuhr wären ebenfalls leicht möglich. Dafür könnten die bereits erwähnten Durchkontaktierungen in der Platine und die beiden ungebelagten Pins des Atmega genutzt werden. Zwischen der Platine und der Rückwand des Gehäuses wäre auch noch genügend Luft vorhanden, in den ein solcher Empfänger eingebaut werden könnte. Alternativ kann hier aber auch eine Lösung gut untergebracht werden, die die Uhr zu einer batteriebetriebenen Wordclock erweitert. Der einfachste Weg dazu wäre, eine USB-Powerbank zu nutzen, da diese integriert bereits Elektrische Schaltungen zum Laden der Akkus und eine Spannungsstabilisierung auf die benötigten 5V hat. Da die Buchstaben alle einzeln ansteuerbar sind, besteht auch die Möglichkeit die Matrix für noch weitere Anwendungen zu nutzen. Denn durch die vielen LEDs können auch Muster wie Kreise oder andere Formen auf der Matrix angezeigt werden. So kann etwa auch die Temperatur angezeigt werden oder alternativ das Datum oder der Wochentag. Dafür würden solche LEDs aktiviert, die zusammen den Umriss eines gewünschten Buchstabens oder einer Zahl bilden, der sich über die gesamte Matrix erstreckt.