

DIY Wordclock

Christof Pfannenmüller



1 EINFÜHRUNG

WORDCLOCKS haben in der Bastler oder DIY-Szene eine große Verbreitung und bergen eine gewisse Faszination.

Grundsätzlich sind die alle Wordclocks sehr ähnlich aufgebaut. Fast immer wird eine Frontplatte auf welcher die Buchstaben passend angeordnet sind, durch eine Matrix an LEDs von hinten erleuchtet. Dadurch, dass nur die zur aktuellen Stunde passenden Wörter erleuchtet sind, 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 vorgesehen Buchstaben der Frontplatte beleuchtet und nicht zu einem Teil auch die benachbarten Buchstaben, da dies den Effekt des Hervorscheins der erleuteten Buchstaben vor den dunkleren minimieren würde. Dafür wird meist eine so genannte Zwischenplatte angebracht, so kann sichergestellt werden, dass das Licht einer LED nicht auf einen anderen Teil der Frontplatte scheint, da es durch die Zwischenplatte abgeschirmt wird. Wordclocks wurden bereits in vielen verschiedenen Sprachen erstellt, wobei grundsätzlich jede Sprache möglich ist. Die einzige Bedingung ist, dass es möglich ist alle Wörter die zum "Schreiben" der Uhrzeit notwendig sind, auf der Frontplatte anzuordnen. Dabei ist jedoch wichtig, dass die Reihenfolge der Wörter so gewählt wird, dass jede mögliche Uhrzeit auch dargestellt werden kann, ohne aber ein Wort doppelt unterbringen zu müssen und trotzdem die Reihenfolge der Wörter dem Sprachgebrauch entspricht. In diesem Zusammenhang ist etwa die Darstel-

lung der Zeit 15:45 interessant, denn je nach Wohnort werden für diese verschiedene Zeiten im Umgang verwendet. Vor allem in Sachsen, Thüringen und Franken wird die Bezeichnung "dreviertel vier" verwendet, in Norddeutschland dagegen eher "viertel vor sechs". Solche Feinheiten müssen also berücksichtigt werden, bei "viertel nach drei" bzw. "viertel vier" verhält es sich ähnlich.

2 ANFORDERUNGEN

Die Wordclock sollte vor allem - im Gegensatz zu den meisten anderen - kleiner ausgelegt sein und nicht zum Aufhängen an der Wand gedacht sein, sondern als modulare kleine Tischuhr, die bei Einbau einer Alarmfunktion, auch als Wecker dienen kann. qlocktwo als Vorbild

Um zunächst einen Überblick zu bekommen sammelte ich alle möglichen Projektideen und mögliche Umsetzungen in

Zur Versionskontrolle wird während des Projektes für alle Dateien github eingesetzt um jeweils den aktuellen Stand zu protokollieren.

3 HARDWARE

Elektronik

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. 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 Vias 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 aufwändiger zu löten sind. Versorgt werden soll die LED mittels Strom aus einem USB-Ladegerät oder Computer damit kein besonderes Ladegerät notwendig ist. Dies legt die Versorgungsspannung auf 5 Volt fest. Als Buchse wird eine kombinierte Micro-USB-Buchse die für Stecker des Types A und B geeignet ist verwendet. Letzere stellt sicher, dass die Wordclock an einem handelsüblichen Ladekabel für Handys angeschlossen werden kann. Die Hauptarbeit übernimmt dabei ein Prozessor der Mega8-Reihe von AVR verwendet, 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, 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 Standard ISP-Stecker verwendet, da bereits damit zu rechnen ist, dass die Software mehrfach aufgespielt werden muss, bis diese reibungslos läuft. Die LED-Matrix zum Anzeigen der einzelnen Buchstaben der Matrix werden 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 ge-

samte 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 die sicherzustellen, dass 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-Interrupt 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 kann. So wird sichergestellt, dass die LEDs niemals blenden, was vor allem für den Einsatz als Wecker wichtig ist, aber 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 angeregt werden muss, sondern nur durch anklemmen an die 5V Versorgungsspannung aktiviert werden kann. Dies ist durch den Mikrocontroller wesentlich leichter zu realisieren. Die zwei nicht verwendenden Pins des Atmega wurden auf ein Via an einer freien Stelle der Leiterplatte gelegt, außerdem wurde hier noch ein Via für Ground und +5Volt eingefügt. Somit kann hier zu einem späteren Zeitpunkt einfach eine Lösung zur Spannungsüberbrückung einbaut werden. Diese könnte sogar mit dem AVR kommunizieren und die Wordclock könnte dadurch in einen stromsparenderen Modus wechseln.

Gehäuse

Um das Gehäuse der Wordclock planen zu können erstelle ich zuerst ein CAD-Modell. Als Programm wurde dafür das SolidWorks verwendet. Dies wurde auch vor dem Hintergrund gemacht, dass die aus den in Solid Works erstellten Dateien etwa die Schnittmuster für die Zwischenplatte extrahiert werden 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 erstellt werden. Die einzelnen CAD-Modelle der Bauteile dafür sind im Internet frei verfügbar. Somit konnte anhand der Platine mit den angebrachten Bauteilen ein gutes Gehäuse geplant werden. Für den Prototyp wurde das Gehäuse mit dem Onlinetool "Boxmaker aus Akryl" ausgelasert. Die Zwischenplatte, 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 Zwischenplatte ergibt sich also mit einer Buchstabengröße 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. Zusammengehalten wird die gesamte Uhr von vier M3 Schrauben, welche die Platine, die Zwischenplatte und das Gehäuse mit Abstandshalter fixieren, damit diese nicht verrutschen.

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, wie es 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 muss nur die Software entsprechend angepasst werden. Da die Frontplatte idealerweise das Licht nur an den Buchstaben das Licht durchlässt und dort leicht milchig sein sollte, gibt es verschiedene Möglichkeiten zum Erstellen der Platte. Die simpelste Möglichkeit ist dabei, das Layout der Frontplatte auf mehrere Overheadfolien zu drucken und diese mit einem Blatt Papier als Difusor gemeinsam zu laminieren. Problematisch ist hierbei allerdings die Verbindung der einzelnen Folien, da diese sich leicht löst. 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 noch eine Akrylscheibe von hinten zuerst farbig und anschließend mehrfach schwarz lackiert aus der das in Illustrator erstellte Frontplattenlayout dann ausgraviert wurde. Die Schwarze Lackierung dient hierbei dazu, dass die LEDs

nicht durch die farbigen Stellen der Frontplatte hindurchleuchten. Eine weitere, aber nicht erprobte Methode wäre das Ausschneiden einer Folie mit dem im Fablab verfügbaren Folienplotter und das aufkleben dieser auf eine Milchglasscheibe. Alle Frontplatten sind mit einem Magnet an der Zwischenplatte befestigt, so wird kein Werkzeug zum Austauschen der Zwischenplatte benötigt.

4 SOFTWARE

5 AUSBLICK

DCF beschriftung der Taster mit Lasercutter Batterienbetrieben Da die Buchstaben einzeln ansteuerbar sind, besteht auch die Möglichkeit die Matrix für noch weitere Anwendungen zu nutzen. So etwas temperatur anzeigen ; Datum Wochentag etc