

DIY: Pflichtenheft Wordclock

Christof Pfannenmüller

6. November 2015

Inhaltsverzeichnis

1	Über dieses Dokument	1
2	Lastenheft	1
3	Pflichtenheft	4
4	Umsetzung	6
4.1	Frontplatte	6
4.2	Gehäuse	6
4.3	Platine	6
4.3.1	Ansteuerung	6

1 Über dieses Dokument

Um die Entwicklung meiner Wordclock etwas zu ordnen möchte in diesem Dokument das Erstellen und die Aufgaben dabei etwas sortiert aufführen.

2 Lastenheft

Hier habe ich das bereits erstellte Lastenheft eingefügt.

Lastenheft Wordclock:

Weckfunktion: vorsehen einer Weckfunktion mit Ton und Snoozefunktion

Taster zum Ausschalten der Alarmfunktion, Snooze und zum Einstellen der Uhrzeit

Helligkeitsanpassung: über Sensor oder über Software und Schalter an der Rückseite

Sekundenanzeige

Datum und Wochentag anzeigen

Funkanbindung zum automatischen Uhrzeit einstellen

Stromreserve: Uhrzeit bleibt bestehen bei Stromausfall evtl. auch Anzeige und Weckfunktion ohne Strom

Ausschalter: kleinen Ausschalter an der Unterseite zum gänzlichen Ausschalten (Uhrzeit läuft über Batterie weiter)

USB Versorgung

24h Modus: Wecker klingelt nur alle 24h nicht alle 12 und die Weckfunktion kann somit angelassene werden

verschiedene Frontcovers möglich: in verschiedenen Sprachen, dafür verschiedene Möglichkeiten der LED Ansteuerung

Stabiles Gehäuse, schön klein

Standfuß (standsicher)

3 Pflichtenheft

Aufgabe	Beschreibung	Wie umzusetzen	bis	Status
<p>Weckfunktion</p> <p>Weckzeitpunkt einstellen</p>		<p>Lautsprecher/Tonger, Taster für Snoozefunktion mehrere Taster, mind. 2 (Mode, Up) besser 3 (Mode, Up, Down); Mode kurz drücken zeigt aktuellen Weckzeitpunkt an, lange drücken um in Modus Weckzeitpunkt kommen; Unterscheidungsmöglichkeit AM/PM notwendig</p>		
<p>Uhrzeit einstellen</p>		<p>Taster Up (Down) von Weckzeitpunkt verwenden; eigenen Taster um in Modus Uhrzeit einstellen zu kommen, oder mehrere Taster gleichzeitig drücken; wen gewünscht Möglichkeit für Datum einstellen (daraus Wochentag berechnen) Sensor für Umgebungslicht mit passender Elektronik; Loch in Gehäuse; Regelung der LEDs durch PWM (kompliziert) evtl. Controller oder durch IC (dafür ADU notwendig)</p>		
<p>Anpassung der Helligkeit</p>	<p>Anzeige heller bei heller Umgebung</p>			
<p>Sekunden-Anzeige</p>	<p>Sekunden zeigt auf der Matrix an</p>	<p>Knopf durch den in Sekundenmodus gewechselt wird; Anzeige der Sekunden auf der Matrix, evtl. nicht bei jedem Frontcover möglich (muss deaktiviert werden können); wo Sekunden gespeichert?</p>		
<p>Anzeige Datum und Wochentag</p>		<p>Wie anzeigen? Wo gespeichert? evtl. nicht bei jedem Frontcover möglich (muss deaktiviert werden können)</p>		
<p>Funkuhr</p>	<p>Einstellen der Uhrzeit nach Atomuhr</p>	<p>DCF77 Verbindung um Uhrzeit automatisch zu finden und einzustellen.</p>		
<p>Stromreserve</p>		<p>Uhrzeit bleibt erhalten und Wecker klingelt trotz fehlendem Strom; Anzeige trotzdem aktiv würde Speicherdauer verkürzen (Anzeige nur im Moment des Weckens oder gar nicht)</p>		

Aufgabe	Beschreibung	Wie umzusetzen	bis	Status
Uhr komplett ausschalten		Uhr über kleinen Schalter komplett ausschalten; Taster an Unterseite oder hinter Frontcover verstecken (bleibt Uhrzeit und Einstellungen erhalten?)		
Stromversorgung		Uhr wird über USB-Netzteil versorgt; USB Buchse verbauen		
24h Modus		Wecker klingelt nur um 6:00am, Uhr zeigt aber gleiches an für am/pm		
verschiedene Frontcovers		Frontcover austauschbar; verschiedene Farben möglich (evtl. bei Farben aktuelle Uhrzeit unterschiedlich gut erkennbar); verschiedene Sprachen (Umschalten der Matrix auf andere Sprache muss möglich sein, dafür Taster etc.)		
Stabiles Gehäuse		Gehäuse mit sicherem Stand; stabil um nicht kaputt zu gehen (herunterfallen); vorerst aus Holz später Alu etc (aus einem Teil)		
Programmieren der Uhr		Schnittstelle um Uhr auch später noch programmieren zu können; an Unterseite oder hinter Frontplatte verstecken		

4 Umsetzung

Der Aufbau der Wordclock soll von vorne nach hinten und von außen nach innen erfolgen:

Ich möchte zuerst die Größe der Frontplatte festlegen, dann ein entsprechendes CAD Modell erstellen und die Schaltung intern auf einer einzelnen Platine unterbringen, da dies den Verdrahtungsaufwand verringert. Das Gehäuse möchte ich zuerst nur aus Holz bauen bzw. am 3D-Drucker ausdrucken und dies aber so, dass es möglich ist zu späteren Zeitpunkten ein Gehäuse aus Aluminium oder Holz zu fräsen.

Ich möchte nicht nur selbstgestaltete Frontplatten verwenden können, sondern auch solche des Herstellers qlocktwo. Dadurch bin ich zwar an die genaue Größe der Frontplatte gebunden, kann aber auf die vielen verschiedenen Sprachen und Farben des Herstellers zugreifen.

4.1 Frontplatte

Für die Frontplatte möchte ich mehrere Sprachen machen, auf jeden Fall Deutsch und Englisch so wie nach Möglichkeit ein Dialekt (vorzugsweise Fränkisch)

Auflistung aller

4.2 Gehäuse

4.3 Platine

4.3.1 Ansteuerung

Für die LEDs stellt sich die Frage nach einer Matrix- oder Einzelansteuerung. Die folgende Tabelle soll die Vor- und Nachteile darstellen:

Vorteile Matrix	Nachteile Matrix	Vorteile Einzelansteuerung	Nachteile Einzelansteuerung
stromsparend	immer nur eine LED an; nicht so hell	Alle LEDs an; hell	Stromverbrauch größer
	Controller muss öfter aufwachen	lange Sleepzeiten des IC	
Dimmen über IC gleich integrierbar	Dimmen über abändern der Zeiträume einzelne LED leuchtet	gemeinsam Dimmen über Anode/Kathode aller LEDs	LEDs nicht unterschiedlich dimmbar

Der Verdrahtungsaufwand der beiden nahezu identisch, da ich die Schaltung auf einer einzigen Platine unterbringen will.

Zeitdauern der Ansteuerung: Gesamtzeit ein Durchlauf aller LEDs kleiner als $\frac{1}{30}$ einer Sekunde (damit kein Flackern sichtbar. Also Zeit für eine LED bei 100% ist $\frac{1}{10*11*30}$ also $\frac{1}{3300}$ Sekunde. Also etwa 0,303 ms. Jetzt würde sich die Zeit durch Dimmen noch weiter verkürzen, was sehr kompliziert würde. Also ist eine Einzelansteuerung und Dimmen aller LEDs über verbundene Anode/Kathode einfacher.

Ansteuern der LEDs über Schieberegister (kaskadierbar, benötigt also nur wenige Anschlüsse des IC) oder über LED-Treiber (eingebaute Möglichkeit des Dimmens; unabhängig von einander dimmbar; Ansteuerung über I2C)