



**Konzeption, Bau und Programmierung
einer lateinischen Wortuhr**

Abbildung 1

Maturitätsarbeit von Tamara Schneider

Klasse A6

Im Fach Informatik

Betreuer Chris Weber

Kantonsschule Limmattal 2020

**KONZEPTION, BAU UND PROGRAMMIERUNG
EINER LATEINISCHEN WORTUHR**

Eine Maturitätsarbeit an der
KANTONSSCHULE LIMMATTAL

vorgelegt von

TAMARA SCHNEIDER

Klasse A6

Im Fach Informatik

betreut von

Chris Weber

Abstract

Diese Arbeit befasst sich mit der Konzeption, dem Bau und der Programmierung einer lateinischen Wortuhr.

In der Einleitung soll eine Erklärung des Objekt «Wortuhr» knapp erklären. Nebst den Kernfragen der Arbeit wird aufgeführt, was der Hintergrund zur Wahl dieses Themas war und wie sich die Motivation dazu gestaltete. Kurz wird auch angesprochen, worin sich das Ziel der Arbeit gestaltete.

In einem nächsten Teil wird das Grundkonzept der Wortuhr erklärt. Dies bezieht sich auf ihre Funktionsweise und auf das für diese Arbeit gewählte Zeitprinzip.

Als erstes grosses Hauptthema wird die Konzeption der Hardware aufgeführt. Darin werden einzelne Schritte erläutert, die vorgenommen wurden, um eine Grundlage für den späteren Bau zu bilden. Dabei wird eher auf abstrakter Ebene gearbeitet.

Das zweite Hauptthema bildet der Bau des physischen Produkts. Hierbei werden alle nötigen Schritte bis zum Bau des Endprodukts aufgeführt und anhand einiger Darstellungen besser erklärt. Der Teil reicht von den einzelnen Funktionen der Uhr über den Aufbau und der Elektronik bis hin zum Bau eines Prototyps und schliesslich der endgültigen physischen Uhr. In Unterkapiteln werden zusätzlich gewisse Erkenntnisse, Problemstellungen und Lösungsansätze erklärt. Im dritten Hauptthema liegt der Fokus auf der Programmierung der Uhr. Es werden nochmals Themen aus vorherigen Kapiteln aufgegriffen und auf einer anderen Ebene neu erklärt. Schliesslich widmet sich der restliche Teil dieses Kapitels der Vorgehensweise, den Problemen und deren Lösungsansätze.

Abschliessend wird nochmals Stellung zur kompletten Arbeit genommen. Nebst dem Erkenntnisfortschritt werden ungelöste Fragen aufgeführt, die eigene Vorgehensweise der Arbeit und die Umsetzung des Konzepts werden beurteilt, allenfalls werden auch Änderungsmöglichkeiten erläutert.

In einem Ausblick wird zusammen mit dem Schlusswort die Darstellung einer weiteren Vorgehensweise zu künftigen Fortschritten nach dieser Arbeit folgen.

Inhalt

1	Einleitung	1
2	Grundkonzept der Wortuhr	4
2.1	Funktionsweise	4
2.2	Zeitprinzip	5
2.2.1	Zwei verschiedene Modi	5
3	Konzeption der Hardware	8
3.1	Konzeption der Zifferblätter	8
3.1.1	Allgemeine Regeln	8
3.1.2	Erstellung der Zifferblätter	10
3.2	Design	17
3.3	Bau eines Modells	19
4	Bau des physischen Produkts	21
4.1	Funktionen	21
4.2	Aufbau der Uhr	22
4.3	Elektronik	23
4.3.1	Die einzelnen Komponenten	23
4.4	Prototyp	26
4.4.1	Vereinfachungen in Bezug auf das Konzept	28
4.4.2	Erkenntnisse	29
4.5	Endprodukt	29
4.5.1	Problemstellungen	32
4.5.2	Lösungsansätze	32
5	Programmierung	33
5.1	Funktionen	33
5.2	Vorgehensweise	36
5.3	Problemstellungen	42
5.4	Lösungsansätze	42
6	Diskussion	43
7	Schlusswort und Ausblick	45
8	Danksagung	46
9	Literaturverzeichnis	47
10	Abbildungsverzeichnis	50
11	Anhang	51
12	Einhaltung rechtlicher Vorgaben	56

1 Einleitung

Was ist eine Wortuhr?

Uhren existieren in unterschiedlichen Varianten und eine mögliche Form dieser stellt auch die Wortuhr dar. Das grundsätzliche Prinzip der Zeitvermittlung unterscheidet sich nicht von einer herkömmlichen Uhr wie man sie aus dem Alltag kennt.¹ Das Ziel ist die exakte Angabe der Uhrzeit, die den genauen Zeitpunkt definiert. Der Hauptunterschied zwischen einer Wortuhr und einer Zifferuhr wird bereits mit den beiden Namen ausgedrückt. Während die Zifferuhr die Zeit in Ziffern ausdrückt, wird sie bei einer Wortuhr mit Worten formuliert. Würde die Uhrzeit in Ziffern ausgedrückt zum Beispiel «08:15 Uhr» lauten, so würde genau die gleiche Uhrzeit bei einer Wortuhr wie folgt formuliert werden: «Es ist acht Uhr fünfzehn» oder «Es ist Viertel nach acht Uhr».

Die Möglichkeit, die Zeit so unterschiedlich auszudrücken, kommt von der Umgestaltung des Zifferblattes. Im Gegensatz zu einer herkömmlichen Uhr, wie sie vorhin schon als Beispiel verwendet wurde, sind die Komponenten des Zifferblattes einer Wortuhr einzelne Buchstaben, die ihren festen Platz haben. Aufgrund der durchdachten Anordnung jedes einzelnen Buchstaben lassen sich Worte bilden, welche wiederum in einer ganz bestimmten Reihenfolge einen Satz ergeben. Dieser Satz gibt uns die Möglichkeit, die Uhrzeit wortwörtlich ablesen zu können.

Das Konzept dieser Arbeit

Zuerst will ich erklären, was meine Veranlassung bzw. mein Hintergrund zur Wahl dieses Themas war. Da sich die Suche nach einem geeigneten Thema für die Maturitätsarbeit als komplizierter als angenommen erwies, suchte ich den Kontakt zu Bekannten. Durch die Gespräche, die sich mit ihnen ergaben, wurde der Fokus zur Wahl eines Themas auf das eigene Interesse gelenkt. Schon immer war klar, dass meine Arbeit einen praktischen Teil beinhalten muss. Werden die eigenen Interessen nun mit der Tatsache kombiniert, dass es einen praktischen Teil beinhalten muss, so wird die Bandbreite zur Auswahl eines Themas bereits eingeschränkt. Durch weitere Inputs von Bekannten war schliesslich klar, dass meine Wahl in Richtung

¹ Unter dem hier verwendeten Begriff «herkömmliche Uhr» muss das Bild einer Uhr mit einem Zahlen-Zifferblatt mit Zeiger oder einer Digitaluhr verstanden werden, wobei die konkrete Art der Uhr (Synchronuhr, Atomuhr, Quarzuhr, etc.) nicht berücksichtigt wird. Entscheidend ist, dass die Zeitangabe mittels Ziffern ausgedrückt wird.

Informatik gehen sollte. Den Entschluss zur Behandlung dieses gewählten Themas fasste ich schlussendlich, als ich im Internet mehr Recherche vollzogen hatte und mir dadurch verschiedene Ideen zur Abänderung, Erweiterung und Ergänzung des Prinzips einer Wortuhr aufkamen.

So wie ich dieses Thema gewählt habe, umfasst es drei signifikante Bearbeitungsbereiche: Konzeption, Bau und Programmierung einer lateinischen Wortuhr. Angesichts der drei unterschiedlichen Teile kamen auch jeweils zu den verschiedenen Bereichen passende Kernfragen auf. Zunächst musste ich die zentrale Kernfrage aufzustellen. Diese gestaltete sich simpel. Das zentrale Thema bildet also die Frage, ob es mir gelingt, ein Konzept für eine Wortuhr mit lateinischer Zeitangabe² zu entwerfen und danach auch erfolgreich umsetzen zu können. Was die Konzeption betrifft, kam erst einmal die Frage auf, inwiefern es möglich ist, selbst von Grund auf eine voll funktionstüchtige Wortuhr zu designen, zu konstruieren und anschliessend zu programmieren. Im Internet lassen sich einige Anleitungen zum Bau einer Wortuhr finden, welche als Vorbild bzw. als Anhaltspunkt verwendet werden können. Ich musste jedoch selbst entscheiden, welche Komponenten³ enthalten sein sollten, welche Funktionen die Uhr schlussendlich aufweisen sollte und wie die Konzeption in dem Sinne ausfallen muss, damit alles aufeinander abgestimmt ist und somit zu einem erfolgreichen Ergebnis führen kann. Es muss immer im Hinterkopf behalten werden, dass die Wortuhren, die im Internet als Richtwert aufzufinden sind, niemals gänzlich den eigenen Vorstellungen entsprechen und deshalb ein eigener Plan bezüglich des Baus und der Programmierung entworfen werden muss.

Zum Bau der Uhr stellten sich Fragen, wie das Design ausfallen sollte, wie und mit welchem Material diese Vorstellung umgesetzt werden kann und wie die elektronischen Komponenten miteinander verkabelt werden müssen, damit sich ein funktionierender, geschlossener Stromkreislauf ergibt. Die bereits erwähnten Anleitungen aus dem Internet dienen tatsächlich nur als Anhaltspunkt, um eine grobe Vorstellung zu erhalten, wie eine solche Wortuhr aufgebaut sein könnte. Oftmals sind die Anleitungen jedoch nicht einmal so ausführlich, dass man jeden Schritt nachvollziehen könnte. Weiter stellte sich die Frage, wo die Elektronik bei der Uhr dann untergebracht wird, da es nur beschränkt Platz zur Verfügung hat.

² Dieser Schritt wird in einem späteren Kapitel genauer erläutert. Kurzgefasst handelt es sich bei der lateinischen Zeitangabe um zwei Zifferblätter, wobei das eine die uns bekannte Uhrzeit, lateinisch formuliert, angibt und das andere die lateinische Zeitrechnung

³ Dies bezieht sich hauptsächlich auf die elektronischen Komponenten

Die meisten Fragen befassten sich jedoch mit dem Bereich der Programmierung der Wortuhr. So kam die Frage auf, wie zwei verschiedene Modi programmiert werden können, zwischen denen erfolgreich gewechselt werden kann. Die weiteren Unklarheiten betrafen vor allem die auszugebende Uhrzeit. Weiter war unklar, wie die Zeitemstellung⁴ berücksichtigt werden muss, wie Sonnenaufgang und Sonnenuntergang miteinbezogen werden können und ob eine selbstgebaute und -programmierte Uhr überhaupt fähig ist, die Zeit konstant exakt anzugeben oder ob es zunehmende Abweichungen gibt.

In erster Linie soll ein vollendetes und voll funktionsfähiges Endprodukt vorgewiesen werden können. Dazu gehört, dass die Wortuhr gut aufgebaut und passend programmiert ist, so dass sie die Uhrzeit korrekt angibt. Zusätzlich soll anhand dieser Arbeit auch gezeigt werden, dass ohne grosse Vorkenntnisse zu einem solchen Thema, sei es bezüglich des Baus oder der Programmierung der Uhr, eine gewisse Komplexität in der Arbeit erreicht werden kann.

Bezüglich der Quellen lässt sich sagen, dass eher Internetseiten verwendet wurden, da dies so ziemlich die einzigen Quellen zur Informationsbeschaffung über Wortuhren sind. Zum Thema «Wortuhr» selbst konnte ich persönlich keine Quellen in Form von Büchern finden, jedoch versuchte ich dort, wo es mir möglich war, dennoch welche zu verwenden. Grundsätzlich wurde bei der Auswahl der Quellen vor allem darauf geachtet, inwiefern sie meinen Fortschritten in der Arbeit dienen könnten. So studierte ich nicht jede einzelne Anleitung einer Wortuhr, die sich finden liess. Sie musste gewissermassen Aspekte meiner Vorstellung enthalten, damit ich es überhaupt erst in Betracht zog, dass sie mir nützlich sein könnte.

⁴ Umstellung zwischen Sommer- und Winterzeit

2 Grundkonzept der Wortuhr

2.1 Funktionsweise

Die Funktionsweise einer Wortuhr unterscheidet sich in vielen Aspekten von einer alltäglichen Uhr. Abgesehen von dem neuen Design und der gänzlich anderen Darstellung der Zeit hat sich ein weiterer Hauptunterschied ergeben, der es überhaupt erst ermöglicht, die Zeit in dieser, eher unbekanntem, Form darzustellen. Dieser Hauptunterschied besteht darin, dass diese eher ungewöhnliche Darstellung der Zeit mithilfe gewisser Elektronikkomponenten erfolgt.

Die ausschlaggebendsten Komponenten stellen die **LEDs**⁵, eine **RTC**⁶ und ein **Mikrocontroller**⁷ dar. [1,2,3] Die Gesamtzahl der LEDs bildet eine **Matrix**, wobei das Zusammenspiel der einzelnen Leuchtdioden ermöglicht wird und dies somit die Grundlage zur Ausgabe der Zeit bildet. Die aktuelle Uhrzeit wird von der RTC bezogen und durch den Code, der mittels einer passend gewählten IDE⁸ erstellt und auf den Mikrocontroller hochgeladen wurde, mit den LEDs in Verbindung gebracht. Der mit der IDE erstellte und hochgeladene Code fällt so aus, dass genau gewählte Dioden zu einer gewissen Zeit aufleuchten, das heisst es wird im Programm definiert, welche Leuchtdioden zu welcher Zeit angesteuert werden.

Schliesslich befindet sich unmittelbar über der Matrix das Zifferblatt, das anstelle der Ziffern einzelne Buchstaben aufweist. Jeder LED wird jeweils ein Buchstabe zugeordnet. Durch die Definition im Code werden exakt gewählte LEDs zu einer bestimmten Zeit angesteuert, wie vorhin bereits erklärt. Dadurch ergibt sich ein Bild eines Satzes auf dem Zifferblatt, wobei die entsprechenden Buchstaben durch die darunter aufleuchtenden LEDs hervorgehoben werden und somit die einzelnen Worte und schliesslich als eine Einheit den zeitaussagenden Satz bilden.

Bsp.: «Es ist zehn Uhr.»

⁵ Engl.: **Light-Emitting Diode** (= lichtemittierende Diode). Im Falle dieser Arbeit werden sogenannte Neopixels verwendet. Auf diese Konkretisierung wird später genauer eingegangen. (4.3. **Elektronik**)

⁶ Engl.: **Real Time Clock** (elektronisches Bauelement, dessen gemessene Uhrzeit vom Mikrocontroller ausgelesen und weiterverarbeitet werden kann.) (4.3. **Elektronik**)

⁷ Ein Mikrocontroller ist genau genommen ein Chip (Halbleiterchip), der einen Prozessor und wichtige Komponenten wie Speicher, Zeitgeber, digitale und analoge Ein- und Ausgabegeräte, etc. aufweist. Durch die Programmierung wird ein Mikrocontroller zur Ausführung einfacher Aufgaben verwendet. (4.3. **Elektronik**)

⁸ Engl.: **Integrated Development Environment** = integrierte Entwicklungsumgebung, z.B. Arduino IDE (Vereint den Texteditor mit Syntax highlighting mit der Kompilierung des Quellcodes und mit dem Upload des Codes auf den Mikrocontroller {evtl. auch mit Start- und Stopfbefehl})

2.2 Zeitprinzip

2.2.1 Zwei verschiedene Modi

Bei dieser Wortuhr kann die Zeit in zwei verschiedenen Modi ausgegeben werden. Beide Modi erfolgen in lateinischer Sprache, jedoch unterscheiden sie sich in einem wesentlichen Punkt. Der erste Modus gibt die Uhrzeit lateinisch formuliert an. Das heisst, dass die Zeit grundsätzlich so ausgegeben wird, wie es für uns in der heutigen Zeit alltäglich ist, allerdings mit lateinischen Worten formuliert. Da aber unsere Zeitformulierung⁹ im klassischen Latein unüblich war bzw. nicht existierte, weist der zweite Modus die römische Tageseinteilung auf. Genau aufgrund der Inexistenz der lateinischen Zeitformulierung suchte ich Rat bei meiner Lateinlehrerin, welche mir hervorragend weiterhelfen konnte. Ich hatte mich im Vorfeld schon ein wenig selbst über die römische Tageseinteilung im Internet informiert. Von dort her war mir bereits klar, wie man die Stunden angeben würde, jedoch lag das Problem eher bei der Minutenanzeige. Von meiner Lateinlehrerin erhielt ich einige Beispiele zur Formulierung der Zeit, unter anderem auch mit speziellen Ausdrücken für gewisse Minutenangaben. Mithilfe dieser Angaben erkannte ich sofort, wie die Zeitangabe erfolgen musste. Genaueres wird anschliessend erläutert.

Lateinisch formulierte Zeit – Modus 1

Das Zeitprinzip dieses ersten Modus sieht wie folgt aus: Wie bereits oben erläutert wurde, basiert diese Darstellung auf unserer **heutigen alltäglichen Zeitformulierung**. Da die Zeitausgabe ausschliesslich lateinisch formuliert erfolgt, muss in der Darstellung die Wortstellung der lateinischen Sprache beachtet werden. Daher ist das Zifferblatt unterteilt in Stunden- und Minutenausgabe. Die Zeit wird in Worten in Fünfminutentakten ausgegeben. Um die Zeit exakt ausgeben zu können, sind zusätzlich noch Minutenanzeigen vorhanden. Damit können die Minuten 1-4, 11-14, 21-24, 31-34, 41-44 und 51-54 ebenfalls angezeigt werden. Zusätzlich muss beachtet werden, dass die Stundenangabe aufgrund Platzmangels nur bis zwölf reicht. Daher wurden noch Ausdrücke für Vormittag bzw. Nachmittag hinzugefügt, damit klar ist, worum es sich handelt.

⁹ Gemeint ist, dass jede Stunde/Minute definiert ist. Wir geben unsere Zeit wie folgt an: «Es ist acht Uhr.» oder «Es ist halb zehn Uhr/Es ist neun Uhr dreissig.» oder «Es ist Viertel vor zehn Uhr/Es ist neun Uhr fünfundvierzig.» Es gibt bei gewissen Ausdrücken zwei verschiedene Varianten zur Angabe, jedoch ist klar was gemeint ist.

Somit erfolgt die Zeitangabe nach dem folgenden Prinzip:

«Es ist {Stunde(n)} Uhr {Minuten} {a.m./p.m.}» =

«Est hora {Ordnungszahl¹⁰} minuta {Ordnungszahl} et {Ordnungszahl}¹¹/et {fester Ausdruck} {a.m./p.m.}»

Zusätzlich erfolgt noch die Angabe für die Minuten 1-4. [4]

Bsp.: «Est hora nona et quadrans a.m.» = «Es ist neun Uhr fünfzehn vormittags»

Römische Tageseinteilung – Modus 2

Das Zeitprinzip dieses zweiten Modus unterscheidet sich gänzlich von dem des ersten Modus. Hier wird die **tatsächliche römische Zeitrechnung** resp. **Tages- und Nachteinteilung** verwendet. Bei dieser Zeitrechnung wird nicht jede einzelne Stunde angegeben, sondern es wird ein etwas anderes Prinzip verfolgt: Die erste Stunde beginnt um etwa sechs Uhr morgens bzw. bei Sonnenaufgang. Ab da wird der Tag bis Sonnenuntergang in zwölf Stunden unterteilt, welche je nach Sommer – oder Winterzeit länger oder kürzer ausfallen. Gewisse Stunden haben ihren festen Zeitpunkt, so beginnt die erste Stunde bei Sonnenaufgang, die sechste Stunde ist immer mittags und die zwölfte Stunde endet mit dem Sonnenuntergang. Zusätzlich können die zwölf Stunden in vier Phasen à je drei Stunden unterteilt werden. Die Stunden eins bis drei bilden den Morgen (*mane*), die Stunden vier bis sechs den Vormittag (*antemeridiana/a.m.*), den Nachmittag (*postmeridiana/p.m.*) bilden die siebte bis und mit der neunten Stunde und schliesslich vergehen die Stunden zehn bis zwölf als Abend (*vesper*).

Die zwölf Stunden nach dem Sonnenuntergang bis zum nächsten Sonnenaufgang, genau genommen die Nacht (*nox*), werden in vier Phasen, genannt Nachtwachen (*vigiliae*), à je drei Stunden eingeteilt. Die erste Nachtwache verstreicht ungefähr von 18 – 21 Uhr, natürlich

¹⁰ Hier ist von den grammatikalischen Ordnungszahlen die Rede, d.h. z.B. erste/r/s, zweite/r/s, dritte/r/s, etc. Zur Angabe der Stunde werden immer Ordnungszahlen verwendet (z.B. «hora nona» = «die neunte Stunde»). Zur Angabe der Minuten werden ebenfalls Ordnungszahlen verwendet, sofern diese in Kombination mit dem Wort «minuta» vorkommen. Für gewisse Minutenangaben (15/ 30/ 45 min) gibt es auch andere Ausdrücke, wofür keine Ordnungszahlen verwendet werden. (z.B. «minuta vicesima» = «die zwanzigste Minute» aber «dodrans» = stehend für 45min)

¹¹ Je nach gewünschter Minutenangabe müssen zwei Ordnungszahlen miteinander kombiniert werden

kommt es ganz darauf an, wann der tatsächliche Sonnenuntergang ist (dies sind nur Richtwerte), denn auch hier müssen sowohl die Sommer- bzw. Winterzeit beachtet werden als auch die damit verbundenen Zeiten für Sonnenaufgang und Sonnenuntergang. Auch die Nachtwachen fallen kürzer oder länger aus, wie es oben schonmal für die Tageseinteilung erwähnt wurde. Die zweite Nachtwache reicht von 21 Uhr bis Mitternacht, die dritte von Mitternacht bis ungefähr drei Uhr morgens und die vierte von drei Uhr bis Sonnenaufgang. Auch hier gelten wieder einige feste Zeiten. Die erste Nachtwache beginnt bei Sonnenuntergang, die zweite endet um Mitternacht und die vierte endet bei Sonnenaufgang. [5]

Bei diesem Prinzip erfolgt die Zeitausgabe also auf folgende Art:

«Phase, Es ist {Ordnungszahl} Stunde/Nachtwache» =

«{Mane/a.m./p.m./Vesper/Nox} hora {Ordnungszahl} vigilia» [4]

Bsp.: «*p.m. hora septima.* » = «*Nachmittag. Es ist die siebte Stunde.*» (= «*Nachmittag. Es ist (ungefähr!) 13 Uhr.*»)

Bsp.: «*Nox. tertia vigilia.* » = «*Nacht. Es ist die dritte Nachtwache.*» (= «*Nacht, zwischen Mitternacht und drei Uhr in der Früh.*»)

3 Konzeption der Hardware¹²

Zu Beginn dieser Arbeit musste ich zuerst ein Konzept entwerfen, wonach ich mich während der ganzen Arbeit richten konnte, auch, um nicht zu viel Zeit zu verlieren, wenn während des Arbeitsprozesses etwas unklar sein bzw. werden sollte. Der Entwurf eines etwas ausführlicheren Konzepts diente der Vorbeugung von Zeitverlust, dem Aufbau einer klar strukturierten Vorgehensweise der Arbeit und es vereinfachte später den Bau des physischen Produkts. Durch meine Recherchen zum Thema «Wortuhr» im Internet ergab sich für mich immer mehr ein Bild der Konzeption der Hardware. [6] In den nachfolgenden Kapiteln wird jeweils auf einzelne Themen eingegangen, welche Grundlagen zur späteren Erstellung der Hardware darstellten.

3.1 Konzeption der Zifferblätter

Ein wichtiger Teil der Konzeption fällt den Zifferblättern zu. Aufgrund meiner gewählten Variante von Zifferblättern in lateinischer Sprache, muss die gesamte Konzeption der Zifferblätter aus eigener Arbeit erfolgen, da ein solches meines Erachtens nicht existiert. Trotz allem konnten andere Zifferblätter als Vorbild zum Aufbau und zur Anordnung der Wörter verwendet werden. Um also den Aufbau eines solchen Zifferblattes und die Anordnung der Ausdrücke zu verstehen, nachvollziehen und dann selbst nach diesem Prinzip arbeiten zu können, verwendete ich zur Hilfe ein Beispiel eines deutschen Zifferblattes, anhand dessen ich einige allgemeine Regeln aufstellen konnte. Diese Erkenntnisse dienten schliesslich der Erstellung der eigentlichen Zifferblätter.

3.1.1 Allgemeine Regeln

Mithilfe meiner Erkenntnisse, die ich aus dem Zeitprinzip mitnehmen konnte (2.2 Zeitprinzip) und ich aus der Analyse des Beispielzifferblattes gewann, formulierte ich einige allgemeine Regeln bezüglich der beiden Zeitformulierungen, des Aufbaus der Zifferblätter und der Anordnung der Wörter. Die Regeln 3 und 4 wurden im Zusammenhang mit dem Beispiel der deutschen Wortuhr erstellt und können auf das lateinische Prinzip angewandt werden.

¹² Der Begriff «Hardware» bezeichnet die Gesamtheit aller physischen Komponenten (elektronische und mechanische Teile) eines datenverarbeitenden Systems > Komplement zu Software (umfasst alle immateriellen Komponenten)

1. Lateinisch formulierte Uhrzeit

Allgemeiner Aufbau:

- *Est hora* (OZ¹³) → XX:00 → Volle Stunde
- *Est hora* (OZ) *et quadrans* → XX:15
- *Est hora* (OZ) *et dimidia* → XX:30
- *Est hora* (OZ) *et dodrans* → XX:45
- *Est hora* (OZ) *minuta* (OZ) {*et* (OZ)} → XX:XX → Minutengenau
(+zusätzliche Minutenanzeige)

(+ *antemeridiana/postmeridiana*)

2. Römische Tageseinteilung

- Phase, *hora* (OZ) (hora)
- Phase, (OZ) *vigilia*



Abbildung 2: Beispiel einer deutschen Wortuhr

3. Aufbau der Zifferblätter

- Feste Ausdrücke Anfang/Ende
- Feste Ausdrücke Mitte
- Minutenangaben
- Stundenangaben



Abbildung 3: Analyse des obigen Beispiels

4. Anordnung der Wörter

- Die Anordnung der Ausdrücke für die Stunden / Minuten erfolgen in der Regel ihrem Wert entsprechend von tief zu hoch, wenn es möglich ist.
- Passt die Buchstabenzahl nicht mehr in die Zeile hinein oder ein anderer Ausdruck stellt eine bessere Option zur Platzausnützung dar, so wird ein anderer Ausdruck verwendet,

¹³ Abkürzung für Ordnungszahl

auch wenn dieser erst später in der Reihenfolge käme. Die eigentliche Reihenfolge wird, wenn möglich, danach weitergeführt.

- c) Zwei Ausdrücke kombinieren → weniger Platzverbrauch → weniger LEDs
- i) Das gleiche Wort kommt zweimal vor
→ Bsp.: *dreiviertel oder viertel* → *(drei)viertel*
 - ii) Der Endbuchstabe eines Wortes entspricht dem Anfangsbuchstaben des nächsten Wortes
→ Bsp.: *Zehn und neun* → *zehneun*

3.1.2 Erstellung der Zifferblätter

Im folgenden Schritt wurden die Zifferblätter erstellt, d.h. ich wandte die Regeln 3 und 4 (**3.1.1 Allgemeine Regeln**) auf die beiden Modi meines gewählten Zeitprinzips an und erstellte so die Anordnung der entsprechenden Ausdrücke für beide Fälle.

1. Lateinisch formulierte Zeit

Regel 3:

Feste Ausdrücke Anfang/Ende

Est hora	antemeridiana (a.m.)	postmeridiana (p.m.)
----------	----------------------	----------------------

Abbildung 4: Feste Ausdrücke, die sich am Anfang oder am Ende des Zifferblattes befinden (Modus 1)

«Est hora» steht immer am Anfang und «a.m.» bzw. «p.m.» stehen ganz am Schluss, da diese ein Anhängsel darstellen, um anzugeben, ob es vormittags oder nachmittags ist.

Feste Ausdrücke Mitte

minuta	(quinta)	et
--------	----------	----

Abbildung 5: Feste Ausdrücke, die sich in der Mitte des Zifferblattes befinden (Modus 1)

Hierbei muss die Wortstellung im Satz beachtet werden. Wie dem allgemeinen Aufbau der Zeitformulierung in der ersten Regel (**Lateinisch formulierte Uhrzeit**) entnommen werden kann, muss «minuta» vor «et» aufgestellt sein, damit die Zeit sowohl mit den Ordnungszahlen als auch mit den speziellen Ausdrücken für 15/ 30/ 45 Minuten angegeben werden kann.

«Quinta» wird hier in Klammern aufgeführt, da es eigentlich kein fester Ausdruck, sondern eine Minutenangabe ist, jedoch hat «quinta» einen festen Platz, der von der Wortstellung im Satz vorgegeben wird. Somit ist die Reihenfolge «minuta» - «quinta» - «et»

→ **Bsp.:** Est hora octava minuta decima. (08:10 Uhr)

Est hora octava minuta quinta et vicesima. (08:25 Uhr)

Est hora octava et dodrans. (08:45 Uhr)

Minutenangaben

quinta (XX:05)	et	vicesima (XX:25)
		tricesima (XX:35)
		quinguagesima (XX:55)
decima (XX:10)		
vicesima (XX:20)		
quadragesima (XX:40)		
quinguagesima (XX:50)		

Abbildung 6: Ausdrücke zur Minutenangabe

Dies sind die Minutenangaben, die in Verbindung mit «minuta» und «et» stehen. Die Ordnungszahlen in der linken Spalte stehen unmittelbar nach «minuta». Die Ordnungszahlen in der rechten Spalte stehen nach «minuta quinta et».

et	quadrans (XX:15)
	dimidia (XX:30)
	dodrans (XX:45)

Abbildung 7: Spezialausdrücke zur Minutenangabe

Diese Minutenangaben stehen immer nach «et», wobei das «minuta» weggelassen wird.

→ **Bsp.:** Est hora octava et quadrans/dimidia/dodrans (08:15 Uhr/ 08:30 Uhr/ 08:45 Uhr)

Stundenangaben

Est hora	prima (01:00/ 13:00)	a.m. oder p.m.
	secunda (02:00/ 14:00)	
	tertia (03:00/ 15:00)	
	quarta (04:00/ 16:00)	
	quinta (05:00/ 17:00)	
	sexta (06:00/ 18:00)	
	septima (07:00/ 19:00)	
	octava (08:00/ 20:00)	
	nona (09:00/ 21:00)	
	decima (10:00/ 22:00)	
	undecima (11:00/ 23:00)	
	duodecima(12:00/ 00:00)	

Abbildung 8: Ausdrücke zur Stundenangabe (Modus 1)

In dieser Tabelle werden alle Stundenmöglichkeiten dargestellt. Zu Beginn steht immer «Est hora» danach steht eine Ordnungszahl und anschliessend kann ganz am Schluss a.m. oder p.m. angehängt werden.

Diese Regel 3 bildet eine Grundlage für Regel 4, da die Regel 4 sowohl die Besonderheiten in der Wortstellung berücksichtigt als auch gerade dazu dient, die komplette Anordnung der Wörter auf dem Zifferblatt vorzunehmen.

Regel 4:

Zuerst wurde kontrolliert, ob die **Regel 4.c)** angewandt werden kann. Bei genauerem Betrachten der Minuten- und Stundenangaben wurde schnell klar, dass **Regel 4.c)i)** angewandt werden kann, **4.c)ii)** hingegen nicht. Somit ergab sich für 4.c)i) folgende Situation: «decima» könnte sowohl mit «undecima» als auch mit «duodecima» kombiniert werden. Ich entschied mich in diesem Fall dazu, «decima» und «undecima» zu kombinieren, woraus sich folgendes ergab: *decima oder undecima* → *(un)decima*

Ansonsten liess sich diese Regel nicht mehr weiter anwenden, was mich dazu verleitete, mit der Anordnung der Wörter auf dem Zifferblatt zu beginnen. (Dies beinhaltet die Regeln **4.a)b)**)

Bei dieser Arbeit kam ich allerdings leider nicht ums Ausprobieren herum. Erst einmal musste ich nämlich die Anzahl Buchstaben pro Zeile bestimmen. Die Buchstabenanzahl pro Zeile durfte nicht zu gross werden, da die Zifferblätter ansonsten immer mehr in die Breite gehen würden, was aufgrund des Designs (3.2. Design) nicht gewollt war.

Um also die geeignete Anzahl Buchstaben pro Zeile zu ermitteln, probierte ich mit Rechnungen und Skizzen verschiedene Wortstellungen aus, wobei die Anzahl der Buchstaben pro Zeile variierte und die **allgemeinen Regeln** (3.1.1. Allgemeine Regeln) berücksichtigt wurden. Zuerst begann ich mit den Stundenangaben, da diese in der lateinischen Formulierung vor den Minutenangaben auftreten.

Um eine ungefähre Vorstellung von einer Zeilenlänge zu bekommen, entschloss ich mich dazu, das Problem erst einmal mathematisch anzugehen. Zuerst bestimmte ich die Gesamtzahl der Neopixels (= P_N). Da die Grösse der Buchstaben von den Neopixels bestimmt wird, hatte ich bereits einen Massstab für die mögliche Zeilenlänge (in cm) bei n Buchstaben. Somit setzte ich für die Variable n beliebige Zahlen ein, die in Zusammenhang mit den kombinierten Wortlängen (Buchstabenanzahl) dennoch Sinn ergeben könnten. Ich dividierte also die Gesamtzahl der Neopixels (P_N) durch die mögliche Anzahl Buchstaben (n) und erhielt dabei die Anzahl Zeilen (= Z).

Als Rechnung ergab sich also:

$$P_N / n = Z$$

Abbildung 9: Formel zur Ermittlung der Zeilenlänge bzw. Buchstabenanzahl pro Zeile

Indem ich die verschiedenen n – Werte auch mit Skizzen darstellte und dabei verschiedene Wortstellungen ausprobierte, konnte ich in kurzer Zeit eine angemessene Buchstabenanzahl pro Zeile bestimmen, welche $n = 13$ beträgt. Hierbei spielte die bildliche Darstellung eine wichtige Rolle zur Bestimmung des n -Werts, da ich somit auch einschätzen konnte, wie die Form und Grösse des Zifferblattes ausfallen würde, was aufgrund meines gewählten Designs auch zu beachten war.

Mit dem bestimmten n – Wert, welcher tatsächlich einen guten Ausgangspunkt darstellte, machte ich mich daran, die Anordnung der Ausdrücke aufzustellen. Da ich mich immer nach der Buchstabenanzahl pro Zeile richten konnte, war es einfach, zusätzlich die Regeln **4.a)** und **b)** zu berücksichtigen. Zuerst versuchte ich, mich möglichst an die Reihenfolge der Stunden nach ihren Werten zu halten, da dies aber dennoch nicht immer möglich war, platzierte ich

einige Ausdrücke gemäss Regel **4.b**) um. Um also auf eine sinnvolle Anordnung der Wörter zu kommen, begann ich mit «Est hora» und setzte danach die entsprechenden Ordnungszahlen für die Stunden ein. Dabei addierte ich immer die Anzahl Buchstaben der Wörter pro Zeile und stimmte mithilfe von Füllbuchstaben (x) die Buchstabenanzahl pro Zeile auf $n = 13$ ab. Füllbuchstaben (x) werden definitiv zwischen Wörter gesetzt, die nicht direkt nebeneinanderstehen dürfen, da die Zeit ansonsten schwieriger abzulesen wäre. Ansonsten werden sie tatsächlich nur zum Lückenfüllen verwendet.

Bsp.: «Est x^{14} hora»

Am nachfolgenden Beispiel soll das Prinzip meiner Vorgehensweise zur Anordnung der Wörter verständlich gemacht werden.

Bsp.: Zeilenlänge = Anzahl Buchstaben pro Zeile = $n = 13$. Die variierende Buchstabenanzahl der einzelnen Wörter wird ausgedrückt mit v . Es ergibt sich eine Zwischensumme (S_n) der Buchstabenanzahl und hierbei werden je nachdem noch Füllbuchstaben (x) dazu addiert. Die Wörter sollen möglichst der Reihe nach angeordnet werden (**Regel 4.a**)), ansonsten wird umgestellt (**Regel 4.b**):

1. Zeile: «Est» ($v=3$), Füllbuchstabe x ($v=1$), «hora» ($v=4$) \rightarrow es ergibt sich eine Zwischensumme $S_n=8$. Als nächstes muss zwangsläufig ein Füllbuchstabe x ($v=1$) eingesetzt werden, da «hora» und die Ordnungszahl ansonsten direkt nebeneinanderstehen würden, was das Ablesen der Zeit erschweren würde. ($S_n=9$) Um auf $n=13$ zu kommen, fehlen noch vier Buchstaben. Es wird bei den noch ausstehenden Ordnungszahlen der Stunden nachgeschaut, ob ein Wort $v=4$ aufweist. «Nona» zeigt $v=4$ auf, daher wird dieser Ausdruck ebenfalls auf diese Zeile gesetzt. (Auf der nächsten Zeile wird mit «prima» begonnen, da in der eigentlichen Reihenfolge der Stundenwerte fortgesetzt wird.
2. Zeile: «prima» ($v=5$) \rightarrow Zwischensumme $S_n=5$, «secunda» würde als nächstes folgen, da «secunda» aber $v=7$ aufweist, wird kontrolliert, ob ein Wort mit $v=8$ vorhanden ist. (Platz vollständig ausnutzen) \rightarrow «(un)decima» hat $v=8$ und wird eingesetzt. ($\rightarrow n=13$) Auf der nächsten Zeile wird mit «secunda» begonnen (Reihenfolge).
3. Zeile: «secunda» ($v=7$) $\rightarrow S_n=7$, Wort mit $v=6$ gesucht, «tertia» ($v=6$) \rightarrow geht in der Reihenfolge auf. ($\rightarrow n=13$)

¹⁴ Füllbuchstabe, repräsentiert gewissermassen einen Abstand zwischen den beiden Ausdrücken

4. Zeile: «quarta» ($v=6$) $\rightarrow S_n=6$, Wort mit $v=7$ benötigt, «septima» ($v=7$) \rightarrow «septima» einsetzen, ($\rightarrow n=13$), auf der nächsten Zeile wird mit «quinta» fortgefahren.
Etc.

Die Anordnung der Stunden erfolgte auf diese Weise, bis alle Ordnungszahlen verwendet wurden, dann wurden die *festen Ausdrücke Mitte* angereiht. Mit den Ordnungszahlen der Minuten verhält es sich genau gleich wie mit jenen der Stunden. Die Minutenangaben erfolgen auf einer neuen Zeile unmittelbar unter den *festen Ausdrücken*. Somit lässt sich das ganze Zifferblatt für den Modus 1 gestalten (*Anhang*).

2. Römische Tageseinteilung

Regel 3:

Feste Ausdrücke Anfang/Ende

mane	vesper	nox	a.m.	p.m.	hora
------	--------	-----	------	------	------

Abbildung 10: Feste Ausdrücke, die sich am Anfang oder Ende des Zifferblattes befinden (Modus 2)

Am Anfang stehen die Phasen der Tageseinteilung (2.2.1 Zwei verschiedene Modi), sowie «hora» am Anfang und am Ende zur Angabe der Stunde (bei der römischen Zeitrechnung wird ab der zehnten Stunde das «hora» nachgestellt).

Feste Ausdrücke Mitte

vigilia

Abbildung 11: Fester Ausdruck, der sich in der Mitte des Zifferblattes befindet (Modus 2)

«Vigilia» ist hierbei das einzige Wort, das fest in der Mitte steht, wobei auch dies nicht zwingend sein müsste. Es hätte genauso gut am Ende stehen können, doch aus optischen Gründen entschied ich mich dazu, «vigilia» als festen Ausdruck in der Mitte zu verwenden.

Minutenangaben

In diesem Modus sind keine Minutenangaben vorhanden.

Stundenangaben

mane	hora	prima (Sonnenaufgang)
		secunda
		terita
a.m.		quarta
		quinta
		sexta (Mittag)
p.m.		septima
		octava
		nona
vesper	decima	hora
	undecima	
	duodecima (Sonnenuntergang)	
nox	prima	vigilia
	secunda (Mitternacht)	
	tertia	
	quarta	

Abbildung 12: Ausdrücke zur Stundenangaben inkl. Phasen (Modus 2)

In der linken Spalte werden die Phasen aufgeführt, welche immer der Zeitangabe vorangestellt werden. Die mittlere und die rechte Spalte sind aufgeteilt in die Ausdrücke «hora» und «vigilia» und die einzelnen Ordnungszahlen. Von der ersten bis zur neunten Stunde wird «hora» der Ordnungszahl vorangestellt, ab der zehnten bis zur zwölften Stunde wird «hora» nachgestellt. «Vigilia» hingegen wird immer der Ordnungszahl nachgestellt. [5]

Auch bei diesem Modus dienen die Anwendungen der Regel 3 als Grundlage zur Regel 4, genau wie schon beim ersten Modus.

Regel 4:

Wie auch schon bei der lateinisch formulierten Zeit, kann die Regel 4.c)i) angewandt werden, 4.c)ii) hingegen nicht. Somit ergibt sich wieder die gleiche Konstruktion wie im Modus 1: «decima» kann mit «undecima» oder «duodecima» kombiniert werden. Ich wählte auch hier dieselbe Kombination: *decima und undecima* → *(un)decima*

Von der Anwendung der Regel 4 auf die lateinisch formulierte Zeit ist bekannt, dass die Buchstabenanzahl/Zeilenlänge $n=13$ ist. Somit konnte ich bei diesem zweiten Zifferblatt entsprechend zum ersten vorgehen, um die Ausdrücke anzuordnen. (3.1.2 **Erstellung der Zifferblätter**) Das ganze Zifferblatt des Modus 2 (**Anhang**) wurde grundsätzlich trotz einiger Unterschiede analog zu dem des Modus 1 gestaltet.

3.2 Design

Ebenfalls zu den Überlegungen der Konzeption der Hardware gehören diejenigen zum Thema «Design der Wortuhr». Das Design der Uhr ist das erste, was auffällt und ausserdem sehr einfach zu bestimmen. Hierbei kann man seinen Vorstellungen freien Lauf lassen oder sich nach Vorbildern aus dem Internet richten. Ich hatte mich an meiner Stelle erst einmal danach erkundigt, wie eine reguläre Wortuhr aussieht, welche Designaspekte sie aufweist, inwieweit die Beispielsuhr meiner Vorstellung entspricht und was ich übernehmen bzw. ändern möchte. Genaugenommen wurde ich aber von nichts so richtig überzeugt, daher entsprang das Design der Uhr meinen eigenen Vorstellungen.

Was sicherlich auffällt ist, dass die Wortuhren im Allgemeinen ziemlich schlicht gehalten sind. Die Form ist meist quadratisch, zur Zeitangabe wird ein einziges Zifferblatt verwendet, die Farbe ist eher unauffällig und die Höhe des Querschnitts der Uhr unterscheidet sich von Produkt zu Produkt, das heisst fabrikgefertigte Wortuhren fallen oftmals kompakter aus als selbstgefertigte. Es gab einige Anforderungen an das Design meiner Uhr, die gleich zu Beginn feststanden, andere entwickelten oder änderten sich während des Arbeitsprozesses.



Abbildung 13: Art der Wortuhr, die am häufigsten anzutreffen ist (quadratisch, niedrige Höhe, schwarz, schlicht)

Als erste Anforderung stand fest: Das Design muss zum Zeitprinzip passen. Das von mir gewählte Zeitprinzip, beinhaltend die Zeit in lateinischer Sprache formuliert und die beiden Modi, existiert meinen Kenntnissen zufolge nicht. Natürlich lässt sich das nur anhand meiner Recherchen im Internet annehmen. Aufgrund dessen sollte das Design dieser Uhr ebenfalls neuartig ausfallen, so wie man es noch nicht kennt. Der Stil der Optik sollte sich demnach dem eher Altmodischen/Antiken zuwenden, um dem Zeitprinzip gerecht zu werden. Hier kommt bei meiner Wortuhr noch die Schwierigkeit hinzu, dass sie zwei Zifferblätter aufweist, die entweder direkt nebeneinander fix platziert oder je nach gewähltem Modus ausgetauscht werden müssen. Um den Gebrauch jedoch zu vereinfachen, entschied ich mich dazu, die Zifferblätter nebeneinander aufzubauen. Allein durch diesen Faktor wird die Grundform der Uhr bestimmt, welche in diesem Fall rechteckig ausfällt.

Anhand dieser ersten Festlegungen entstand eine Idee, der zufolge die Uhr wie ein aufgeschlagenes Buch aussehen sollte. Dieses Design würde sich mit der Tatsache, dass zwei Zifferblätter untergebracht werden müssen, vereinbaren lassen, und es würde zugleich einen neuen und aussergewöhnlichen Stil der Wortuhr repräsentieren. Aufgrund einiger Komplikationen, die sich mit diesem Design ergeben hätten, habe ich diese Idee jedoch ziemlich schnell wieder verworfen und versucht, es einfacher zu halten.

Die zweite Idee führte gleichzeitig zum finalen Design. Die beiden Zifferblätter wurden nebeneinander platziert. Aufgrund der gewählten Methode¹⁵ zur Herstellung der Zifferblätter bildeten beide gemeinsam ein «grosses Zifferblatt», was bedeutet, dass sie zusammen auf einer einzigen grossen Scheibe aufgebracht sind. Umgeben von einem Holzrahmen, ergab sich die Möglichkeit, eine zusätzliche Frontscheibe einzubauen, womit sich das jeweils ausser Betrieb gesetzte Zifferblatt verdecken lässt. Eine Bedingung des zusätzlichen Frontglases war jedoch, dass es einfach zu verschieben sein muss.

Die Farbe blieb schlicht; die Wahl der Farbe der Zifferblätter fiel auf schwarz, jedoch wieder um dem «antiken Stil» gerecht zu werden, wurde es marmoriertes Schwarz. Die Farbe des Rahmes blieb naturfarben. Die Querschnittshöhe der Uhr sollte nicht zu hoch ausfallen, dies wurde jedoch hauptsächlich durch die Elektronik bestimmt, und nicht durch persönliche Vorstellungen.

¹⁵ Die Zifferblätter werden bestehen aus zwei Komponenten: Eine Hobbyglasscheibe und Selbstklebefolie. Die Buchstaben werden aus der Folie ausgeschnitten und die präparierte Folie wird anschliessend auf der Scheibe angebracht. (4.2. Aufbau der Uhr)

3.3 Bau eines Modells

Um dem physischen Produkt etwas näher zu kommen, beschloss ich, ein Modell zu bauen. Dies sollte dazu dienen, den Aufbau der Uhr noch besser zu verstehen. Der Nutzen des Modells besteht also schlussendlich darin, erkennen zu können, was so funktioniert wie ich es mir vorgestellt hatte und was nicht. Der Entschluss zum Bau des Modells kam daher, dass es manchmal schwierig war, etwas nur mit Skizzen richtig ein- bzw. abzuschätzen und ich mich vergewissern wollte, dass ich meine Vorstellungen des Designs auch korrekt umsetzen kann. Die Skizzen dienten trotz allem als Anhaltspunkt zur Herstellung des Modells. Mit der Arbeit dieses Kapitels bewegte ich mich langsam bereits weg vom Abstrakten und wandte mich dem Konkreteren zu. Dennoch zähle ich diesen Schritt zur Konzeption der Hardware, da ich mithilfe dieses Modells, den daraus resultierenden Erkenntnissen und meinen anschliessenden Gedanken bezüglich der Verbesserung des Konzepts der Hardware, den Bau des physischen Produkts besser kalkulieren und umsetzen konnte.

Vor dem Baubeginn dieses Modells hatte ich schon einige Skizzen angefertigt, die sowohl den äusseren Bau zeigten als auch die Elektronik im Inneren. Da die Skizzen mir jedoch keinesfalls dabei verhalfen, eine Grössenvorstellung des Ergebnisses zu erhalten, griff ich auf die erstellten Zifferblätter zurück. Die Grösse der Zifferblätter wird, wie bereits einmal erwähnt, von den Neopixeln bestimmt, da diese am Streifen verwendet werden und somit die Buchstabengrösse bzw. Buchstabenbreite vorgeben und diese wiederum bestimmen durch ihre Anzahl die Breite des Zifferblattes; die Höhe hingegen konnte selbst entschieden werden. Die Zifferblätter dienten also als Vorgaberichtlinien zum endgültigen Format des Modells.

Die Zifferblätter wurden auf eine Kartonplatte aufgelegt und ich schätzte ab, wie viel Platz nötig wäre, um die Elektronik rundherum anzubringen. Aufgrund meines zuerst gewählten Designs - die Form eines aufgeschlagenen Buches - bestimmte ich die Aussenmasse 70.5 x 43.5 cm.

Die Zifferblätterschablonen klebte ich auf die abgemessenen Partien auf. Mittels einfachster Materialien (Schaumgummi, Fimo - Knetmasse, Leim) fertigte ich «Neopixels» an, welche ich schliesslich auf den Zifferblättern an den entsprechenden Stellen, das heisst direkt mittig in den Buchstabenfeldern, anbrachte. Die Litzen, also die Kabel, stellte ich ebenfalls ganz vereinfacht mit



Abbildung 14: Bau des Modells: Zifferblattschablonen auf Karton



Abbildung 15: "Neopixels" auf das Zifferblatt geklebt und "Litzen"

Mit der fertigen «Elektronik» machte ich mich daran, einen Rahmen zu gestalten und schliesslich das Abdeckteil mit den Zifferblättern, welche ich aus Karton und Fotokarton anfertigte und darauf die Buchstaben an den exakt abgemessenen Stellen aufmalte; dies auch, um einen ersten Eindruck meiner Ergebnisse aus den letzten Kapiteln zu erhalten. Das Endergebnis zeigte sich so, dass es einen auf der Grundplatte fest angebrachten Rahmen gab und das Abdeckteil separat einen Rahmen aufwies. Diese beiden Teile können ineinandergesteckt werden und ermöglichen so eine einfache Anschauungsweise des Modells. Zusätzlich befestigte ich die Zifferblätter mit Magneten, was eine zusätzliche Art der Besichtigung ermöglicht.



Abbildung 17: Fertiges Modell mit Rahmen, Abdeckteil und Zifferblättern

Der Bau dieses Modells machte mir klar, dass das Buchform - Design einige Komplikationen mit sich bringen würde, wie zum Beispiel die Montage der Zifferblätter sowie deren Positionierung. Sie müssten vertieft in einen Rahmen eingesetzt werden. Dies würde meiner Meinung nach dem eigentlichen Zweck der Uhr entgegenwirken. Der Fokus sollte auf den Zifferblättern mit den Zeitangaben liegen und nicht auf dem Rahmen, somit würde sich dieses Design also eher kontraproduktiv verhalten. Ich stellte also fest, dass es sich kompliziert und die Sache unnötig erschwerend gestalten würde, das Design eines aufgeschlagenen Buches umsetzen zu wollen, allein schon aufgrund des Rahmens, welcher das vorhin beschriebene Problem zur Folge hat. Daher verwarf ich diese Design-Idee und überlegte mir Alternativen.

Pfeifenreinigern dar, dabei hielt ich mich ganz genau an meine Skizze der Elektronik. Um die Zifferblätterschablonen herum setzte ich einen Rahmen, der es mir ermöglichte, innerhalb dessen ein Raster zu platzieren, welches ich zuvor aus Fotokarton herstellte. Dieses Raster dient der Abtrennung der einzelnen Neopixels, damit sie sich nicht gegenseitig anleuchten. Zur Vervollständigung der Elektronik hatte ich mich dazu entschlossen auch die einzelnen Elektronikkomponenten (**4.3. Elektronik**) zu gestalten und anzubringen.



Abbildung 16: Elektronik, Rahmen um Neopixels, Raster

4 Bau des physischen Produkts

4.1 Funktionen

Zu den ersten Überlegungen gehören folgende Fragen, welche sich mit den Funktionen der Uhr beschäftigen:

1. Was soll mit der Uhr erreicht werden?
2. Wie erfolgt die Zeitübermittlung?
3. Wie soll die Bedienung¹⁶ der Uhr erfolgen?
4. Sollte eine manuelle Zeiteinstellung ebenfalls möglich sein?

Um später ein funktionierendes Programm schreiben zu können, ist es wichtig, bereits im Voraus zu wissen, welche Funktionen die Wortuhr schlussendlich aufweisen soll. Nur so hat man bereits einige Anhaltspunkte, nach denen man sich richten kann, ohne dass nachträglich überlegt werden muss, was das eigentliche Ziel ist.

Die Uhr, wie sie in dieser Arbeit entwickelt worden ist, sollte gemäss den oben aufgeführten Fragen folgendes erfüllen können:

1. Das Ziel der Uhr ist die korrekte Zeitausgabe in zwei verschiedenen Modi gemäss dem bereits aufgeführten Zeitprinzip in dieser Arbeit. (**2.2. Zeitprinzip**) Jeweils ein Modus und ein Zifferblatt bilden zusammen die entsprechende Kombination, daher erfolgt mit dem Wechsel des Modus auch ein Wechsel des Zifferblatts.
2. Die Zeitausgabe erfolgt mittels einer RTC und der LEDs gemäss der Funktionsweise der Uhr (**2.1. Funktionsweise**)
3. Die Bedienung der Uhr soll mit einer IR-Fernbedienung¹⁷ erreicht werden. [7]
4. Eine manuelle Zeiteinstellung sollte mithilfe einer IR-Fernbedienung möglich sein, für den Fall, dass Zeitabweichungen aufgrund der RTC auftreten sollten.

¹⁶ Die Bedienung der Uhr umfasst grundsätzlich alles, was nicht im Programm für dauerhaft vermerkt ist, sondern bei Bedarf ausgeführt wird (z.B. Wechsel zwischen den Modi, Ein- bzw. Ausschalten der Neopixels, Farbeinstellung, Helligkeitsänderung, etc.)

¹⁷ Infrarot – Fernbedienung (**4.3.1. Die einzelnen Komponenten**)

4.2 Aufbau der Uhr

Prinzipiell lässt sich sagen, dass der Aufbau der Uhr auf **vier Hauptbestandteilen** beruht. Hierbei liess ich mich zum Teil von Wortuhren inspirieren, die ich im Internet fand. [8,9,10] Diese vier Bestandteile stellen die folgenden dar:

Die **Grundplatte**, die **Elektronik**¹⁸, der **Rahmen** und die **Zifferblätter** (bzw. das **Zifferblatt**, da beide Zifferblätter auf einer Scheibe angebracht sind.)

Als essenzieller Grundbaustein des ganzen Produkts wird eine nichtleitende Grundplatte benötigt, worauf die Elektronik angebracht werden kann. Am besten eignet sich dafür ein Material wie zum Beispiel MDF¹⁹. [11] Eine Grundplatte aus MDF bringt einige Vorteile mit sich: Das Material ist leicht anzuschaffen, preiswert und bei entsprechend gewählter Dicke äusserst stabil. Diese Oberfläche ist bestens geeignet, um die Elektronik darauf zu platzieren, da die volle Garantie besteht, dass von der Oberfläche kein Strom geleitet werden kann.

Der zweite ausschlaggebende Punkt des Aufbaus bildet die Elektronik. Wie bereits erklärt, wird die Elektronik, die Neopixels und die weiteren elektronischen Komponenten umfassend (**4.3.Elektronik**), unmittelbar auf der Grundplatte platziert. Die Neopixel-Streifen werden an genau abgemessenen Stellen aufgeklebt und anschliessend mittels anzulötender Kabel miteinander verbunden. Die übrigen Elektronikteile werden in einem speziell dafür entworfenen Gehäuse untergebracht, da auf der Grundplatte nebst den verkabelten Neopixels und aufgrund der Querschnittshöhe nur beschränkt Platz zur Verfügung steht.

Als drittes wurde oben der Rahmen aufgeführt. Hierbei wird zwischen einem inneren und einem äusseren Rahmen unterschieden. Der innere Rahmen dient der Erstellung und Begrenzung des Uhrengehäuses, worauf dann das Zifferblatt zu liegen kommt. Der äussere Rahmen dient der Verkleidung des Uhrengehäuses und der Verbesserung der Optik. Zusätzlich dient der äussere Rahmen der Montierung der Frontscheibe zur Verdeckung des jeweils nicht gebrauchten Zifferblattes. Dafür werden Schienen auf der Innenseite des Rahmens oberhalb des Uhrengehäuses angebracht, worin die Frontscheibe hin- und herbewegt werden kann. Bei diesem Schritt muss allerdings darauf geachtet werden, dass eine Seite des Aussenrahmens entfernbar sein muss, um die Frontscheibe und das Zifferblatt entnehmen zu können, falls einmal Ausfälle bei den

¹⁸ Wird in einem späteren Kapitel ausführlich erläutert

¹⁹ Abkürzung für mitteldichte Holzfaserverplatte oder mitteldichte Faserplatte, Werkstoff aus dem Bereich der Faserplatten

Neopixels auftreten sollten. Durch die Entnahme der beiden Scheiben lässt sich der Fehler einfach beheben.

Der vierte und letzte Hauptbestandteil der Uhr bildet das Zifferblatt. Das Zifferblatt besteht grundsätzlich aus zwei Komponenten: Eine Hobbyglasscheibe und Selbstklebefolie. Die einzelnen Buchstaben werden aus der Folie ausgeschnitten und die präparierte Folie wird anschliessend auf dem Hobbyglas aufgebracht. Somit kann recht simpel ein präzises Zifferblatt zustande kommen, das es erlaubt, die Zeit korrekt abzulesen.

4.3 Elektronik

Die Elektronik bildet einen essenziellen Hauptbestandteil der kompletten Uhr, daher wird in dem folgenden Unterkapitel die gesamte Elektronik aufgeführt, und die einzelnen Komponenten und deren Funktionen werden erklärt. Auch an diesem Punkt richtete ich mich nach gewissen Vorbildern [8, 9, 10, 12]

4.3.1 Die einzelnen Komponenten

Die gesamte Elektronik setzt sich zusammen aus den folgenden Komponenten: Neopixels²⁰ (LEDs), eine RTC, eine IR- Fernbedienung inkl. IR-Receiver, ein Mikrocontroller und kleinere Teile wie Widerstände und Kondensatoren [1, 2, 7, 13, 14, 15]

1. **Neopixels:** Der grösste Bestandteil der Elektronik machen die Neopixels aus. Dies sind LEDs, die als Streifen verwendet werden können, allerdings weisen sie einen beträchtlichen Unterschied zu einem gewöhnlichen RGB -LED-Streifen auf. Die Neopixels können nämlich im Streifen selbst jeweils individuell angesteuert werden, während bei einem gewöhnlichen LED-Streifen immer mindestens drei LEDs gemeinsam angesteuert werden müssen. Genauso sind



Abbildung 18: Individuell angesteuerte Neopixels am Streifen

²⁰ Oftmals wird «Pixels» als verkürzte Variante verwendet.

sie viel praktischer, als wenn einzelne Leuchtdioden verwendet werden würden. Durch den Neopixel-Streifen entfällt viel mühsame Lötarbeit, da nur noch die Verkabelung der Streifen miteinander vorgenommen werden muss, was um einiges weniger aufwändig ist, als wenn einzelne Dioden verwendet werden würden, die einzeln verlötet werden müssten. Somit bildet diese Form von LEDs einen enormen Vorteil bei der Erstellung der **Matrix** und der Programmierung der Uhr.



Abbildung 19: Neopixels am Streifen, hier schon die Matrix bildend

2. **RTC:** Die RTC (**R**eal **T**ime **C**lock = Echtzeituhr) ist ein elektronisches Bauelement, das es uns erlaubt, die Zeit zu messen. Diese gemessene Uhrzeit kann von dem Mikro-

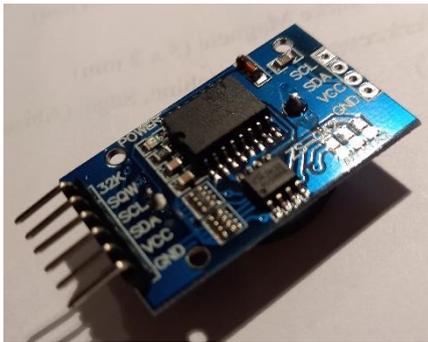


Abbildung 20: RTC (DS3231)

controller ausgelesen und anschliessend weiterverarbeitet werden. Damit haben wir die Möglichkeit, im Code zu definieren, zu welcher gemessenen Uhrzeit die entsprechenden Neopixels angesteuert werden sollen. [1]

3. **IR – Fernbedienung inkl. IR – Receiver:**

Die IR – Fernbedienung ist ein elektronisches Handgerät mithilfe dessen sich Geräte oder Maschinen aus einer gewissen Distanz bedienen lassen. Diese Fernbedienung funktioniert so, dass sie ein Signal im Bereich der Infrarotstrahlung an einen Empfänger (receiver) sendet, wobei Informationen übertragen werden. [7]

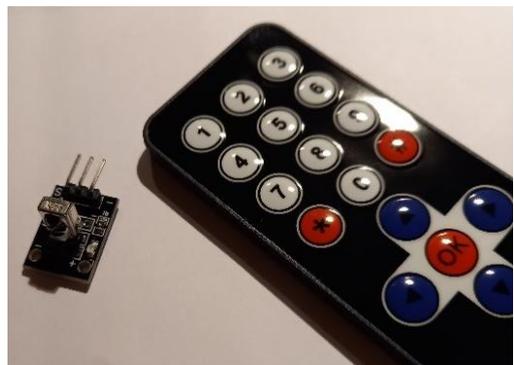


Abbildung 21: IR - Fernbedienung mit IR - Receiver

- 4. Mikrocontroller:** Der Mikrocontroller ist genau genommen ein Chip (Halbleiterchip), der einen Prozessor und wichtige Komponenten wie Speicher, Zeitgeber, digitale und analoge Ein- und Ausgabegeräte, etc. aufweist. Der Mikrocontroller wird zur Ausführung einfacher Aufgaben programmiert. In meiner Arbeit verwende ich den Arduino Nano. [2]

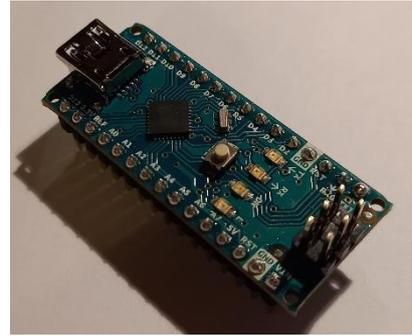


Abbildung 22: Arduino Nano



- 5. Widerstände:** Widerstände werden im elektrischen Stromkreis dafür eingesetzt, dass der durch sie hindurchfliessende Strom etwas gebremst wird. Dies ist zum Beispiel notwendig, wenn eine 4,5-Volt Spannung an ein 3-Volt Lämpchen angeschlossen wird. Der Widerstand sorgt hierbei dafür, dass die überflüssigen 1,5 Volt in Wärme umgewandelt und an die Umgebung abgestrahlt werden. [13]

Abbildung23: Widerstand

- 6. Kondensatoren:** Kondensatoren sind Bauelemente, die elektrische Ladung speichern können. Für meine Arbeit sind Elektrolyt-Kondensatoren (Elko) von Bedeutung. Diese Art von Kondensator wird durch eine besonders hohe Speicherkapazität ausgezeichnet. Bei fliessendem Strom «lädt» sich der Kondensator «auf», bei unterbrochenem Stromfluss wird die Ladung gespeichert, und er kann sich zu einem anderen Zeitpunkt wieder entladen. [14]



Abbildung 24: Elko - Kondensator

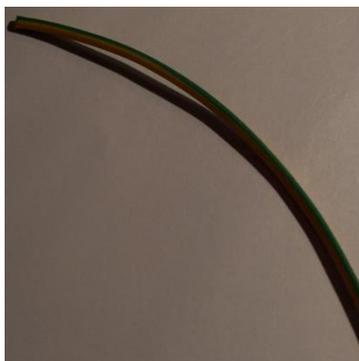


Abbildung 25: Litze

- 7. Litzen:** Litzen sind meist Kupferleiter, wobei diese aus einer Anzahl sehr dünner und biegsamer Drähte bestehen. Der Strom wird so geleitet. [15]

4.4 Prototyp

Bevor ich mich daran machte, die endgültige Fassung der Uhr zu bauen, stellte ich einen Prototyp her. Dieser dient als kleines Testobjekt. Das für den Bau verwendete **Material** kann einer Liste aus dem Anhang entnommen werden, genauso wie der **Bauplan**.

Grundsätzlich stellt der Aufbau des Prototyps eine stark vereinfachte Version des Aufbaus der Uhr dar (4.2 Aufbau der Uhr).

Zuerst schnitt ich eine Grundplatte aus MDF in der gewünschten Grösse zu, wobei mir, wie beim Modell auch schon, eine Zifferblattschablone zu Nutzen kam. Für die Zifferblattschablone des Prototyps verwendete ich der Einfachheit halber das erstellte Zifferblatt für die Lateinisch



Abbildung 26: Grundplatte mit Abmessungen für den Prototyp

formulierte Zeit, wobei ich die Ausdrücke für die Stunden und Minuten durch römische Ziffern ersetzte. Mithilfe der Schablone konnte ich auf der Platte Abmessungen vornehmen, welche den Platz der Neopixels markierten. Dies gestaltete sich einfach, indem ich das Raster der Schablone auf die Grundplatte übertrug. Die Neopixels konnte ich ganz einfach mittig auf den Markierungen aufkleben, da sie bereits einen Klebestreifen auf der Rückseite aufwiesen. Die Litzen knipste ich in der gewünschten bzw. benötigten Länge

ab, und an beiden Enden isolierte ich ein kleines Stück der Isolierung ab. An dieser Stelle kam auch schon der Lötkolben zum Einsatz. Zuerst benetzte ich sowohl die freien Litzen -Enden als auch die Kontaktstellen der Neopixels mit etwas Lötzinn. So gestaltete es sich sehr simpel, die Litzen an die Kontaktstellen der Neopixels zu löten, da nur beide mit Lötzinn benetzten Stellen gleichzeitig vom Lötkolben erhitzt werden mussten, während sie den Kontakt zueinander hatten. An jeweils eine Kontaktstelle der drei Anschlüsse wurde ein Jumper wire²¹ angelötet. [16]

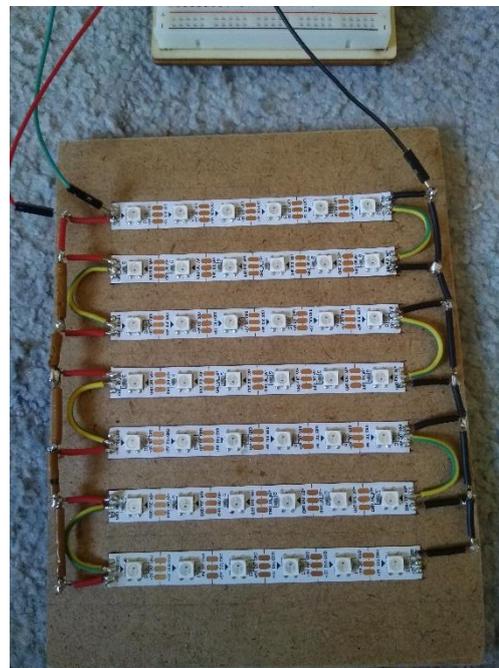


Abbildung 27: Aufgeklebte Neopixels und Verkabelung mit Litzen und Jumper wires

²¹ Jumper wire = Überbrückungskabel mit Stecker oder Stift an jedem Ende

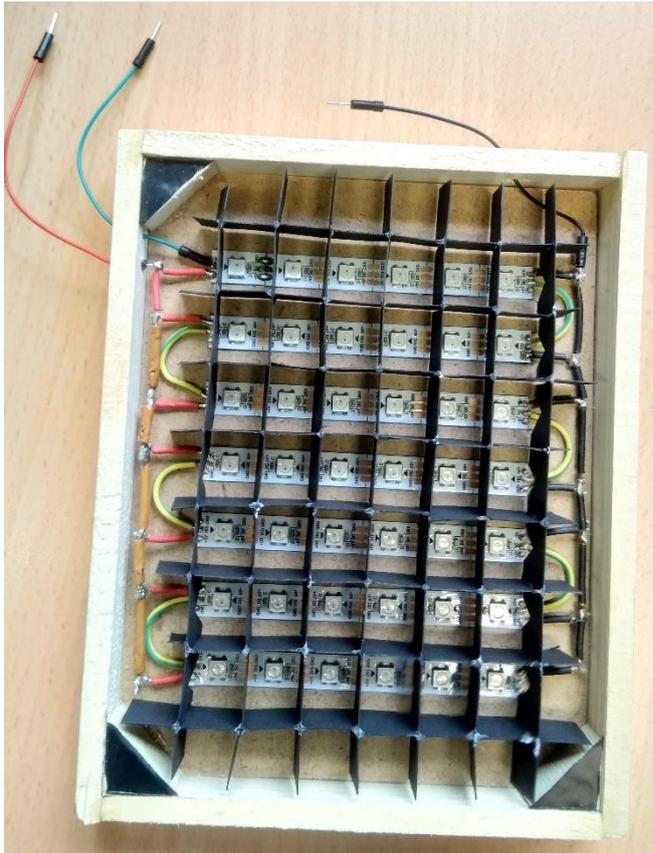


Abbildung 28: Prototyp mit dem Rahmen, den durchgezogenen Jumper wires, den Dreiecksleisten in den Ecken und dem Raster

Aus Gipserratten formte ich den Rahmen, wobei es hier beim Prototyp nur einen einfachen Rahmen gibt. Es wurden also Stücke in der gewünschten Länge zugesägt, anschliessend an den Rand der Grundplatte geklebt und zusammengepresst, um so zu trocknen. Ich bohrte drei Löcher in den Rahmen, durch welche ich die Jumper wire durchziehen konnte. Aus Fotokarton schnitt ich Streifen aus, die in der Länge, Breite und Höhe an das Uhrengehäuse angepasst waren. Aus ihnen formte ich ein Raster, dessen einzelne Felder der Grösse der Buchstaben angepasst waren. Dieses Raster kann in das Gehäuse eingesetzt, aber auch daraus ent-

fernt werden, da es keine besondere Befestigung dafür gibt. Von den Dreiecksleisten wurden kleine Stücke verwendet, die in den Ecken

des Uhrengehäuses zu liegen kamen. Angepasst auf die Grösse des Uhrengehäuses, schnitt ich ein Stück des Hobbyglases zu, um damit das Zifferblatt zu bilden. Da ich bereits eine Schablone des Zifferblattes hatte, stellte sich auch keine grosse Schwierigkeit darin, das Zifferblatt aus der Folie zu schneiden. Ich klebte die Schablone auf die Folie, schnitt mit dem Skalpell die Buchstaben aus, entfernte die Teile, welche schlussendlich von Licht durchschienen werden sollten, liess aber Feinheiten stehen, um den Buchstaben richtig zu erkennen. Die zugeschnittene Folie klebte ich auf das Glas.

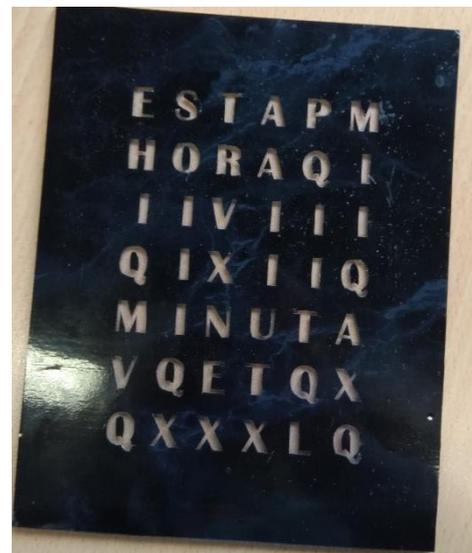


Abbildung 29: Zifferblatt mit Folie



Abbildung 30: Rückseite des Zifferblattes mit der Milchglasfolie und darunter die Magnete

Nun komme ich zu dem Teil, welcher sich der Montage des Zifferblattes widmet. In die Dreiecksleisten bohrte ich jeweils ein Loch und setzte einen Magneten ein. Analog zum Punkt des Magnetes in den Dreiecksleisten, bohrte ich Löcher in die Rückseite der Glasscheibe und setzte auch dort wieder Magnete ein unter Berücksichtigung, dass die Pole korrekt ausgerichtet sind. Die Rückseite des Zifferblattes beklebte ich dann noch mit Opal – und Milchglasfolie, um das Licht der Neopixels ein

wenig zu dimmen und einen schöneren Effekt zu erzeugen. Schliesslich kann das Zifferblatt ganz einfach auf dem Uhrengehäuse angebracht werden und der Prototyp ist bereit zum Einsatz.

4.4.1 Vereinfachungen in Bezug auf das Konzept

Wie es bereits im Kapitel 4.4. Prototyp zum Teil angesprochen wurde, habe ich beim Bau des Prototyps einige Vereinfachungen in Bezug auf das Konzept bzw. den Aufbau der Uhr vorgenommen.

Ganz offensichtlich ist der Prototyp erst einmal um einiges kleiner als das Endprodukt, wobei erwähnt werden muss, dass die Grösse meines Prototyps schon ziemlich gross gewählt ist für ein solches Testobjekt (19.5 x 15.5 cm). Dies habe ich jedoch bewusst so gewählt, um auch in Hinsicht auf das Programmieren eine grössere Möglichkeit bezüglich Testversuchen zu haben. So konnte ich vor dem Abschliessen des Baus des Endprodukts schon einige Funktionen mit dem Prototyp testen, natürlich in einem kleineren Rahmen, was jedoch keinen wirklichen Nachteil darstellte. Ganz im Gegenteil: der Prototyp eignete sich ganz hervorragend dazu, schnell kleine Versuche durchzuführen, ohne einen grossen Aufwand dafür betreiben zu müssen.

Eine weitere Vereinfachung stellt der Rahmen dar. Hier kommt der Prototyp mit nur einem einzigen Rahmen aus anstelle eines inneren und eines äusseren Rahmens. Es musste nur das Grundsätzliche getestet werden, das heisst ob die Bauart des Rahmens funktioniert. Die komplexere Variante bezüglich des Rahmens blieb dem Endprodukt vorbehalten.

Abgesehen davon, dass nur ein Zifferblatt zur Verfügung (Änderung in 5.4. Lösungsansätze) steht, habe ich auch die Darstellung der Zeitangaben stark vereinfacht. Anstelle der spezifischen Ausdrücke stehen lediglich römische Ziffern, sowohl für die Minuten als auch für die Stunden.

4.4.2 Erkenntnisse

Was ich mit dem Bau des Prototyps realisiert habe, ist, dass sich die genauen Skizzen, die vielen Überlegungen und Recherchen bezüglich des Baus einer Wortuhr und die Konzeption der Zifferblätter definitiv gelohnt haben. Aufgrund all meiner vorherigen Arbeitsschritte fiel es mir leicht, den Bau vorzunehmen, und dies mit grossem Erfolg. Der Prototyp ist voll funktionsfähig, bietet mir gute Möglichkeiten das Programmieren zu testen, und er gab Aufschluss darüber, dass ich meine Vorstellungen sehr wohl so umsetzen konnte, wie ich es wollte. Er zeigte auf, dass manche Methoden zur Gestaltung gewisser Komponenten des Gesamtbaus richtig gewählt waren und genauso für das Endprodukt übernommen werden konnten. Doch nicht nur Positives war ersichtlich. Ich konnte genauso auch Defizite in gewissen Bereichen ausfindig machen und dafür Lösungsansätze suchen. Ein Beispiel dafür wäre der Versuch, das Licht der Neopixels zu dimmen. Durch die Scheibe hindurch würde man die Neopixels erkennen, also sollten die Folien dies unerkennbar machen, sodass man wirklich nur eine schöne Färbung des Buchstabens erkennt. Zu Beginn versuchte ich dies mit Transparentpapier, wobei sich dann die Frage stellte, wie dieses unter das Zifferblatt gelegt werden sollte, ohne beschädigt zu werden und nebenbei erzeugte es keinen besonderen Effekt. Diese Erkenntnis verleitete mich dazu, die Verwendung von Klebefolien auszuprobieren, was sich als erfolgreich erwies.

4.5 Endprodukt

Basierend auf den vorherigen Kapiteln gilt es nun, sich dem Endprodukt zuzuwenden und den Bau anzugehen. Dafür besuchte ich die Werkstatt der Kantonsschule Limmattal, wo ich einen Grossteil der Bauarbeit erledigen konnte. Der Bau des Endprodukts erfolgte nach dem bereits erläuterten Aufbau der Uhr, und die grundsätzliche Vorgehensweise entsprach derjenigen aus dem Kapitel Prototyp. Mithilfe der Zifferblattschablonen legte ich auch hier, wie bereits beim Bau eines Modells, zuerst das Format der Uhr fest. Die Masse der fertigen Uhr betragen **60 x 42 cm**. Dies waren einige Zentimeter weniger als beim Modell, was von der Abänderung des Designs kam. Die Masse änderten sich jedoch in einem späteren Verlauf auf **60 x 51 cm**, der Grund dafür wird später erläutert (4.5.1. Problemstellungen und 4.5.2. Lösungsansätze).

Auch hier beginnend mit der Grundplatte, bildete ich Basis des Produkts, worauf ich dann den Rest aufbaute. Anschliessend folgen einige Impressionen vom Bau der Uhr auch mit einigen Erweiterungen zum Prototyp.

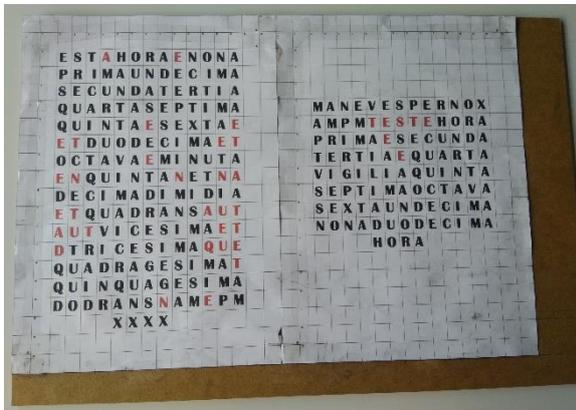


Abbildung 31: Mithilfe der Zifferblattschablonen wird die Grundplatte abgemessen

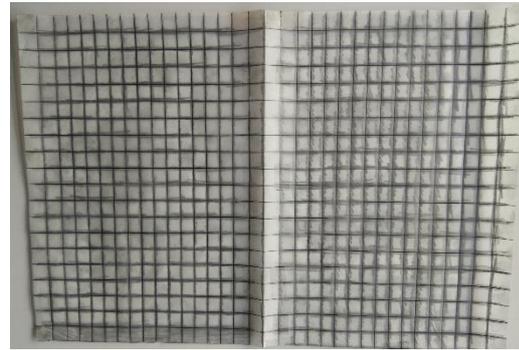


Abbildung 32: Darstellung des Transparentpapiers zur Übertragung des Rasters auf die Platte

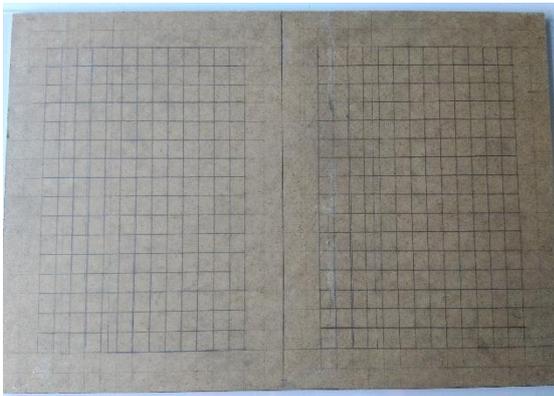


Abbildung 33: Grundplatte mit aufgezeichnetem Raster

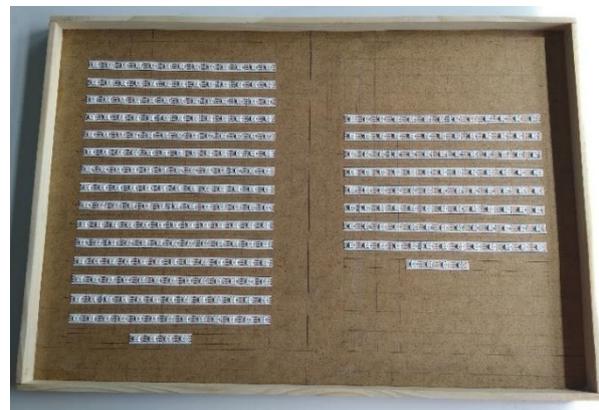


Abbildung 34: Aufgeklebte Neopixels und innerer Rahmen



Abbildung 36: Teile des Aussenrahmens

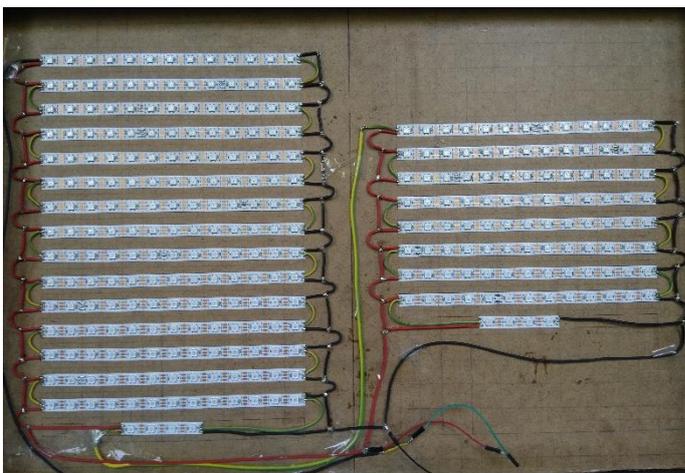


Abbildung 35: Fertige Matrix



Abbildung 37: Schiene



Abbildung 38: In Folie eingefasste Schienen auf Teile des Aussenrahmens



Abbildung 39: Teile des Aussenrahmens am Innenrahmen befestigt



Abbildung 40: Der Querschnitt sieht wie auf dem Bild aus. Zeigt vor allem die Schienen gut mit den beiden Glasscheiben



Abbildung 41: Ein kleiner Rahmen mit Kerben wird jeweils rund um die Matrix gebaut.

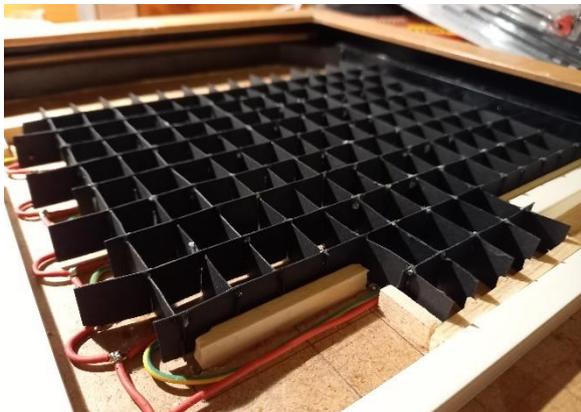


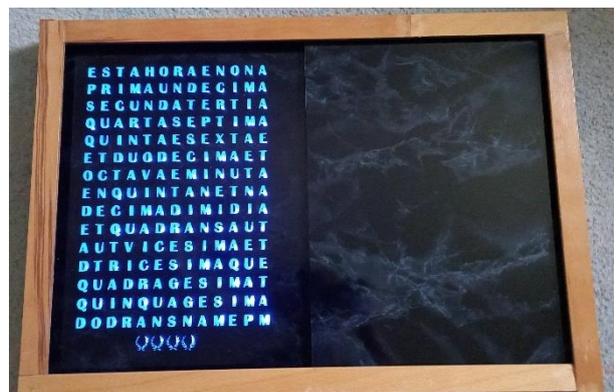
Abbildung 42: Das Raster in den Rahmen eingesetzt.



Abbildung 43: Das fertige Zifferblatt



Abbildung 44: Um für ein wenig Ordnung zu sorgen, werden Kabelkanäle angebracht. Rechts Abbildung 45: Fertige Uhr mit Frontscheibe



4.5.1 Problemstellungen

Trotz der ganzen Vorbereitungen für den Bau des Endprodukts stellten sich zwei Hauptprobleme:

Aufgrund der grossen Neopixelanzahl und deren Verteilung auf zwei Zifferblätter, musste ich den Platz möglichst sinnvoll ausnutzen, um die Verkabelung vorzunehmen und die übrigen Elektronikteile zu platzieren. Die zur Verfügung stehende Fläche war relativ knapp berechnet, da das Format der Uhr keine zu enormen Dimensionen annehmen sollte aufgrund der sowieso schon gegebenen Grösse. Die durch die grosse Zahl der Neopixels verursachte komplexere Verkabelung beanspruchte einen grösseren Teil der noch freien Fläche und liess nicht mehr allzu viel für die anderen elektronischen Komponenten übrig. An dieser Stelle muss auch erwähnt werden, dass der Platzmangel für die restlichen elektronischen Komponenten zusätzlich noch durch die beschränkte Höhe des Querschnitts verursacht wurde.

Ein zweites Problem stellten die Raster zur Neopixelabtrennung dar. Die Neopixels wiesen ringsum noch beträchtlich Freiraum auf, was es erschwerte, die Raster zu platzieren. Ganz ohne Befestigungsmöglichkeit hätten sich die Raster beim Einsetzen des Zifferblattes verschoben, was zu einer unerwünschten Streuung des Lichts geführt hätte.

4.5.2 Lösungsansätze

Als Lösungsansätze für das Platzproblem boten sich nicht viele Optionen. Aufgrund des zusätzlichen Höhenproblems gab es beinahe nur eine Möglichkeit, den Platzmangel zu beheben. Die von mir gewählte Lösung besteht darin, ein zusätzliches Gehäuseteil an das Endprodukt anzubauen und somit für zusätzlichen Platz zu sorgen. Dieses Gehäuseteil musste dieselbe Höhe²² wie das bestehende Endprodukt aufweisen. Da die Höhe beim eigentlichen Uhrengehäuse jedoch durch das Zifferblatt minimiert wird, stand beim Anbau also verhältnismässig mehr Platz zur Verfügung, um die Elektronik unterzubringen.

Das Rasterproblem liess sich einfach beheben. Um die Neopixels herum musste ich lediglich einen kleinen Rahmen anbringen, um so eine Abgrenzung zu schaffen, die die Raster an Ort und Stelle hält. Um zusätzlich für Sicherheit zu sorgen, dass die Raster nicht verrutschen

²² Es wird mit der Höhe des Aussenrahmens gerechnet

können, brachte ich kleine Einschub-Kerben an. Diese sorgen dafür, dass sich das Raster nicht oder nur ganz minimal bewegen lässt und somit die lichtundurchlässige Abtrennung der Neopixels gewährleistet ist.

5 Programmierung

An dieser Stelle waren zwei der drei grossen Hauptkapitel meiner Arbeit abgeschlossen, mit Erfolg. Das dritte und letzte grosse Thema bildete die Programmierung, da die Uhr nicht ohne den Code funktionieren kann. In den folgenden Kapiteln wird nochmals aufgeführt, welche Funktionen vorhanden sein sollen, wie ich bei der Programmierung vorgeing und welche Probleme und Lösungsansätze sich dabei ergaben.

5.1 Funktionen

Die Funktionen, welche hier besprochen werden, resultierten aus den Funktionen, die bereits beim Bau des physischen Produkts erläutert wurden, nur geht es in diesem Kapitel um den Teil der Umsetzung im Bereich der Programmierung. Das erste Kapitel zu den Funktionen (siehe oben) diente grundsätzlich zuerst der konkreten Formulierung der gewünschten Funktionen, um überhaupt einen Anhaltspunkt zur Umsetzung zu erhalten. Hier nochmals zusammengefasst die gewünschten Funktionen:

- korrekte Zeitausgabe in zwei verschiedenen Modi
- Die Zeitausgabe erfolgt mittels einer RTC und der LEDs
- Die Bedienung der Uhr soll mit einer IR-Fernbedienung erreicht werden

Die Funktionen hängen alle gewissermassen zusammen, das heisst das eine bildet die Grundlage für das andere, wobei das andere die Erweiterung des ersten darstellt.

So kommen die grundsätzlichen Basisfunktionen den Neopixels selbst zu. Diese werden ganz simpel mittels des Codes angesteuert, und erst wenn dieser Schritt der Ansteuerung verstanden ist, kann man sich den weiteren Funktionen zuwenden. Ganz vereinfacht gesagt muss nur geklärt sein, in welchem Fall die Neopixels eingeschaltet werden und wann nicht, wie der allgemeine Befehl zur Ansteuerung eines Pixels lautet und wie dieser mit anderen Funktionen kombiniert werden kann. Bevor man sich allerdings der direkten Ansteuerung der Pixels widmen

kann, muss zuerst einiges Grundlegendes verstanden werden, damit auch die einfachsten Befehle ausgeführt werden können. Beispiel dafür:

Ein erster Schritt beinhaltet sowohl die Definition des **Pins am Mikrocontroller**, an den die Neopixels angeschlossen werden, als auch die Definition der **Gesamtzahl der Neopixels**. Mit diesen beiden Angaben und einer dritten zur **Art der Neopixels** kann der Streifen als Objekt definiert werden, dafür wird er noch mit einem **Namen** benannt.

Bsp.: Adafruit_NeoPixel `pixels(PIXEL_COUNT, PIXELPIN, NEO_GRB + NEO_KHZ800);`

Ein anderer wichtiger Punkt ist die Zählweise der Neopixels. Wenn im Code eine Pixelzahl definiert wird, entspricht dies nicht unserer Zählweise im Alltag, indem man bei «eins» zu zählen beginnt. In der Programmiersprache entspricht die «0» unserer «1». Für den n -ten Pixel ergibt sich also die Formel $n - 1$. Für die Pixelansprache wird oftmals die Variable « i » verwendet, für welche ein Zahlenwert eingesetzt werden kann. Will man also zum Beispiel das 23. Neopixel ansteuern, so wird im Programm für die Variable i der Wert 22 eingesetzt, was bedingt durch die Zählweise, bei «0» beginnend, exakt dem 23. Pixel entspricht.

Wenn dann gewisse Grundlagen geklärt sind, folgt die direkte Ansteuerung der Neopixels. Zu der allgemeinen Ansteuerung eines einzigen Pixels gehören verschiedenen Befehle. [17] So sind die folgenden Befehle die ersten, die überhaupt benötigt werden:

```
pixels.begin();           pixels.show();           pixels.clear();
```

Diese Befehle sorgen für das Einschalten der Neopixels bzw. für deren Ausschalten.

An die Vorherigen anschliessend folgen einfache Befehle wie:

```
pixels.setPixelColor(i,R,G,B)23;           pixels.setBrightness();
```

Diese Befehle bewirken zum einen die Farbeinstellung eines Pixels und zum anderen die Helligkeit, in der die Pixels leuchten sollen.

²³ Die Variable i kann dafür verwendet werden, die Zahl des Pixels anzugeben, welches angesteuert werden soll. Die Variable R steht für rot, G für grün und B für Blau. Mit diesen Werten kann jede beliebige Farbe ausgegeben werden.

Natürlich gibt es noch viele weitere Befehle, mit denen man einen komplexeren Code schreiben kann, hierbei gehe ich aber nicht mehr weiter auf Details ein.

Die RTC wird an den Mikrocontroller angeschlossen, und im Code kann mittels Befehlen bestimmt werden, dass die Zeit ausgelesen werden soll. So können bedingte Befehle formuliert werden, dass genau ausgewählte Neopixels zu einer gewissen Zeit angesteuert werden sollen.

```
Bsp.: if((theTime.hour() == 1) || (theTime.hour() == 13)){  
    pixels.setPixelColor(i,R,G,B);  
}  
pixels.show();
```

Dieses Beispiel sagt aus, dass wenn die ausgemessene Stunde der RTC «1 Uhr» oder «13 Uhr» entspricht, das Pixel an der Stelle i angesprochen wird, sprich dieses aufleuchtet. So können die Neopixels mit der RTC kombiniert werden.

Die Zeitausgabe in zwei verschiedenen Modi wird zum einen durch die beiden Zifferblätter bestimmt, zum anderen aber auch durch die Programmierung, denn nur wenn ein Wechsel möglich ist, erfolgt auch die unterschiedliche Zeitausgabe. Da nun bekannt ist, wie die Funktionen der Neopixels und der RTC erfolgen, können so die beiden Modi gebildet werden, indem diese als eigene Funktionen²⁴ definiert werden. [18] Später können genau diese Funktionen abgerufen werden.

Die IR – Fernbedienung funktioniert ohne den IR – Receiver (Empfänger) nicht. Der Empfänger muss ebenfalls am Mikrocontroller angeschlossen werden, damit mit den empfangenen Informationen Befehle ausgeführt werden können. Den einzelnen Tasten werden verschiedene Werte zugeteilt, und wenn der Receiver einen bestimmten Wert empfängt, kann eine bestimmte Aktion ausgeführt werden, wenn diese im Code angegeben wurde.

²⁴ In der Arduino IDE wird mit dem void setup() und dem void loop() gearbeitet, welche beide Funktionen sind. Die Funktion void setup() wird beim Start des Sketches (das Programm) ein einziges Mal abgerufen, um Variablen, Bibliotheken, Pinmodes, etc. zu initialisieren. Die Funktion void loop() hingegen ist eine Endlosschleife, welche nach jedem Durchlauf erneut aufgerufen wird, womit es möglich gemacht wird, Variablen zu ändern oder ähnliches zu erreichen. Nebst diesen beiden Funktionen können auch eigene definiert werden, wobei diese Funktionen dann wiederum in das void loop() eingesetzt werden können.

```
Bsp.: if (irrecv.decode(&results)) { // wenn ein Signal empfangen wird
    Serial.print(" DEC: ");
    Serial.println(results.value, DEC); // wird dieses ausgegeben
    if (results.value == 16753245){ // dieser Wert entspricht der Taste 1
        pixels.setPixelColor(0,0,0,255);//wird dieser Wert empfangen, leuchtet das 1.Pixel blau
    }
    if (results.value == 16736925){ // Dieser Wert entspricht der Taste 2
        pixels.setPixelColor(3,255,0,0);// wird dieser Wert empfangen, leuchtet das 4. Pixel rot
    }
    pixels.show(); // Pixels beginnen zu leuchten
    irrecv.resume(); // Empfänger wird zurückgesetzt, um ein neues Signal empfangen zu können
}
```

Diese Codesequenz habe ich zusätzlich kommentiert, um klar zu zeigen, was bei diesem Vorgang geschieht und wie sich die Funktion der Fernbedienung gestaltet.

Nach diesem Prinzip können Befehle dazu erstellt werden, was beim Drücken einer bestimmten Taste geschehen soll.

5.2 Vorgehensweise

Aus dem vorherigen Kapitel ist bekannt, wie sich die Funktion im Bereich der Programmierung gestalten und wie sie ablaufen. Um auf diese Erkenntnisse zu kommen und anschliessend weiterfahren zu können, recherchierte ich viel im Internet und setzte mich mit Quellcodes anderer Wortuhren auseinander. Noch vor allem anderen war mir klar, dass die Programmierung mit der Arduino IDE erfolgen würde, und ich erkundigte mich danach, welche Programmiersprache verwendet wird: C++²⁵. [19]

Als nächstes musste ich wissen, welche elektronischen Komponenten verwendet werden sollten. An dieser Stelle kam wieder das Thema Funktionen auf, da diese, einfach gesagt, einen wichtigen und ausschlaggebenden Punkt der gesamten Programmierung darstellen. Dadurch, dass ich wusste, was meine Uhr können sollte, suchte ich gezielt nach Wortuhren, die ähnliche Funktionen aufwiesen und studierte deren elektronischen Komponenten und die damit

²⁵ C++ stellt eine Erweiterung der Programmiersprache C dar. Mit C++ ist eine effiziente und maschinennahe Programmierung möglich und auch eine Programmierung auf hohem Abstraktionsniveau.

verbundenen Quellcodes. Am Anfang gestaltete es sich noch relativ schwierig, die Quellcodes zu verstehen, wenn nicht einmal die Funktionen der einzelnen Komponenten bekannt waren. Trotz allem war mir klar, was meine Uhr benötigte, um richtig zu funktionieren. Somit stellte ich meine eigene Komponentenliste zusammen und machte mich daran, deren Funktionen kennenzulernen und zu verstehen.

Da die Neopixels den grössten Teil ausmachten, beschäftigte ich mich zuerst damit, deren Ansteuerung zu verstehen. Tatsächlich suchte ich zuerst im Internet nach einer Anleitung dafür, wie man die Neopixels ansteuern kann. Dabei stiess ich auf ein Youtube – Video, welches die ersten Schritte ganz einfach erklärte. [20] Mithilfe dessen schrieb ich auch ein ganz simples Programm, womit einzelne Neopixels zum Leuchten gebracht wurden. Schon bei diesem ersten Schritt lernte ich, dass man sogenannte libraries²⁶ (**Bibliotheken**) einbinden kann, welche die Arbeit etwas vereinfachen dürften. [21]

Bsp.: `#include <Adafruit_NeoPixel.h>`

```
#define PIN 6
#define NUMPIXELS 42
Adafruit_NeoPixel pixels = Adafruit_NeoPixel (NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
  pixels.begin();
  pixels.setPixelColor(0,0,0,100);
  pixels.setPixelColor(1,0,0,100);
  pixels.setPixelColor(2,0,0,100);
  pixels.show();
}
```

Mit dem Prototyp testete ich ausserdem ein Beispiel der Adafruit_Neopixel Bibliothek (strand-test). [22] Dieses Beispiel eignete sich perfekt, um meine Lötarbeit zu untersuchen, da alle Neopixels angesprochen wurden und ich somit sehen konnte, ob ich noch einige Verbesserungen vornehmen musste oder nicht. Glücklicherweise zeigte sich so, dass die Lötarbeit exakt war.

²⁶ Die IDE kann damit erweitert werden und durch die Libraries stehen mehr Funktionen zur Verfügung, was einen z.B. bei der Arbeit mit einer Hardware unterstützen kann

Als nächstes wandte ich mich der IR – Fernbedienung zu. Auch hier erkundigte ich mich im Internet über die Anwendung und fand dort einige hilfreiche Websites und Videos. [23, 24, 25, 26] Als erstes musste die Fernbedienung decodiert werden, d.h. die Werte der Tasten werden ermittelt. Dies ist nötig, da im Programm definiert werden muss, bei welchem empfangenen Wert eine bestimmte Aktion ausgeführt werden soll. Mithilfe meiner Quellen stellte dies kein Problem dar, und ich notierte die erhaltenen Werte in einer Tabelle. Der Code für eine Decodierung sieht wie folgt aus [23]:

```
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.print(" DEC: ");
    Serial.println(results.value, DEC); //Dezimal
    Serial.print(" HEX: ");
    Serial.println(results.value, HEX); //Hexadezimal
    Serial.print(" BIN: ");
    Serial.println(results.value, BIN); //Binär
    irrecv.resume();
  }
}
```

Vom IR – Empfänger wird jeweils ein Signal empfangen, dessen Wert im Serial Monitor (Arduino IDE < Tools < Serial Monitor) in der gewünschten Art (dezimal, hexadezimal oder binär) ausgegeben wird.

In einem weiteren Schritt versuchte ich einfache LEDs mit der Fernbedienung anzusteuern. Hierfür mussten die LEDs lediglich definiert werden, im `void setup()` als OUTPUT gesetzt

werden, und in der `if` – Schleife im `void loop()` konnte eine zusätzliche `if` – Schleife gesetzt werden, um zu definieren, bei welchem Signal die LED angesteuert wird. [24]

Bsp.: `int ledrot = 10;`

```
...
pinMode (ledrot, OUTPUT);
...
if (results.value == 16724175){
    digitalWrite (ledrot);
}
```

Und im letzten Schritt versuchte ich natürlich die Neopixels anzusteuern. Dies entsprach grundsätzlich dem genau gleichen Prinzip, hier musste einfach darauf geachtet werden, dass mit den Befehlen für die Neopixels gearbeitet wird. An dieser Stelle diente wiederum der Prototyp als Testobjekt.

Schliesslich musste ich noch die RTC testen. Auch für die RTC fand ich eine Library, deren Beispiel ich ausprobieren konnte, jedoch zuerst auf dem Serial Monitor. Dort wurde nach dem Upload des Codes auf den Mikrocontroller die Zeit ausgegeben. [27] Ich suchte weiter nach einer besseren Library mit einfacheren Funktionen. Schliesslich stiess ich auf eine Webseite mit Anleitung für eine Wortuhr, deren Code ebenfalls zu besichtigen war. In diesem Code wurde eine Library verwendet, die mir sehr gut gefiel und die ich als geeignet erachtete für mein eigenes Programm. [28] Ich band diese Library ein und schrieb ein kurzes Programm für meinen Prototyp. Die Befehle funktionierten einwandfrei. An dieser Stelle versuchte ich mich auch daran, eigene Funktionen zu erstellen für die einzelnen Wörter. Ich erstellte also Funktionen für die Ausdrücke, wobei die Neopixels für die einzelnen Buchstaben dort zusammengefasst wurden.

Bsp.: `void EST() { // Für das Wort «EST» müssen die ersten drei Neopixel angesteuert werden`

```
    pixel[0] = 1; //pixel[i] stellt eine Variable dar, die vorhin definiert wurde
    pixel[1] = 1;
    pixel[2] = 1;
}
```

Tatsächlich zeigte die Methode mit den Funktionen das gewünschte Ergebnis.

Da ich nun auch die Funktion der RTC kannte, war es an der Zeit, das Programm für das erste Zifferblatt zu schreiben. Dafür musste ich die eben gewonnenen Erkenntnisse miteinander

kombinieren. Als Hilfe dienten mir vor allem zwei andere Quellcodes, welche eine relativ einfache Methode zur Zeitausgabe verwendeten, ein dritter bestätigte mir, dass es möglich war, die Methode anzuwenden, welche ich mir überlegt hatte. [28,29,30]

Als allererstes musste ich Libraries einbinden und Objekte definieren. Dies geschieht immer vor allem anderen, da ansonsten Funktionen eventuell nicht erkannt werden und Befehle nicht ausgeführt werden können. Kurz gesagt, es treten Fehlermeldungen auf, wenn etwas nicht korrekt deklariert wurde.

Das Wichtigste dieses Schrittes war folgendes:

- Die Neopixels, deren Anschluss am Mikrocontroller und die Anzahl definieren und deren Library einbinden, anschliessend kann der Neopixelstreifen noch als Objekt definiert werden. Zusätzlich werden einer Variablen für die Pixels die entsprechenden Werte zugeordnet.
- Die Library der RTC einbinden und diese als Objekt definieren, ebenso ein Objekt für die aktuelle Uhrzeit.
- Die Library für die IR – Fernbedienung einbinden, den Pin am Mikrocontroller und ebenfalls Objekte für die Resultate und den Signalempfang definieren.

Als ich dies erledigt hatte, konnte ich damit beginnen, das Programm zu schreiben. Mir war von Anfang an klar, dass ich nicht immer jedes einzelne Neopixel aufschreiben wollte, wenn ich definierte, zu welcher Zeit welche Neopixels aufleuchten sollen. Ich entschied mich dazu, eine Methode zu verwenden, welche zu Beginn zwar etwas Geduld erforderte, jedoch war es im Anschluss einfacher, die Zeitdefinitionen zu schreiben. Als Vorbereitung für diesen Schritt hatte ich mir in meinem Dokument mit den Zifferblättern zu jedem Buchstaben die entsprechende Zahl notiert. So konnte ich schnell ablesen, welche Neopixels ich angeben musste.

(Anhang)

Meine Methode bestand darin, für die einzelnen Ausdrücke Funktionen zu erstellen, wie es vorhin schon einmal erklärt wurde. Diese Funktionen konnte ich später ganz einfach mit dem Namen aufrufen. Dies gab mir zum einen die Sicherheit, dass ich die richtigen Neopixels angegeben hatte, und zum anderen sah es auch besser aus, als wenn nur die Neopixelzahlen angegeben würden. Es war um ein Vielfaches übersichtlicher. Um alles noch übersichtlicher zu machen, schrieb ich die Funktionen in einen neuen Sketch und fügte diesen in das eigentliche Programm ein.

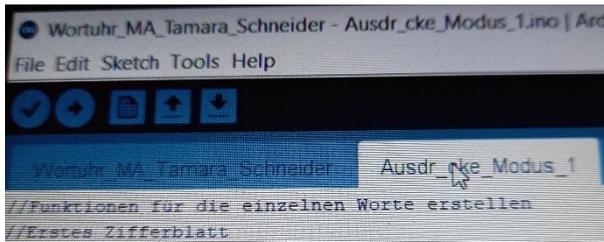


Abbildung 46: Sketch in Hauptprogramm eingefügt

Nachdem ich alle Funktionen erstellt hatte, wandte ich mich den Zeitdefinitionen der Modi zu, wobei ich auch diese als Funktion deklarierte. Hier schrieb ich ebenfalls alles in neuen Sketchen auf und fügte sie zum Programm hinzu. Analog zu dem Beispiel zur

RTC im Kapitel 5.1. **Funktionen**, schrieb ich alle nötigen Zeitdefinitionen auf, sowohl für den ersten als auch für den zweiten Modus. Dabei musste immer überlegt werden, wie die Zeitanzeige korrekt lauten würde, damit keine Fehler geschahen.

Will man zum Beispiel definieren, dass die Neopixel der Funktion für die Minuten 1 – 4 nacheinander aufleuchten, so würde man folgendes schreiben:

```
if((theTime.minute() < 5)){ //Wenn die Minute kleiner als 5 ist
  MINUTA(); //das Wort «Minuta» immer ausgeben
  switch(theTime.minute()){
    case 1: //wenn die Minute gleich 1 ist
      M1(); //Funktion M1(); ausgeben →Dass das Neopixel für die erste Minute aufleuchtet
      break;
    case 2: //Wenn die Minute gleich 2 ist
      M2(); //Funktion M2(); ausgeben
      break;
    case 3: //Wenn die Minute gleich 3 ist
      M3(); //Funktion M3(); ausgeben
      break;
    case 4: //Wenn die Minute gleich 4 ist
      M4(); // Funktion M4(); ausgeben
      break;
  }
}
```

In diesem Fall wird die if – Schleife ab der Minute 5 nicht mehr durchlaufen.

Nachdem die beiden Modi fertiggeschrieben waren, konnte ich das void setup() aufstellen und anschliessend das void loop(). Im void loop() deklarierte ich noch, dass beim Drücken einer

Taste der IR – Fernbedienung der Modus 1 eingeschaltet werden sollte, bei einer anderen Taste der zweite Modus. Der fertige Code konnte anschliessend auf den Mikrocontroller geladen und dieser an die Uhr angeschlossen werden.

5.3 Problemstellungen

Bei dieser Programmierarbeit ergaben sich einige Probleme. Zu der Zeit fand ich noch keine passende Funktion, um in den Zeitenangaben den Sonnenaufgang und Sonnenuntergang miteinzubeziehen. Daher fiel auch die andere Stundeneinteilung schwer, die über den Tag verteilt wäre. Ein anderes Problem stellte, auch immer noch, die IR – Fernbedienung dar, die nicht so genutzt werden konnte wie gewollt. Beim Drücken der Taste wurde der Modus zwar angesteuert, die Pixels flackerten kurz auf, erloschen dann aber gleich wieder. In Kombination mit der RTC schien es einen Störfaktor zu geben, der es verhinderte, dass die Neopixels richtig angesteuert werden. Die RTC weist wie vermutet eine kleine Zeitverzögerung auf, was allerdings keinen gravierenden Fehler darstellt. Das schwerwiegendste Problem lag jedoch beim Upload des Codes auf den Mikrocontroller und dessen Anschluss an die Uhr. Ich musste feststellen, dass das Programm zwar perfekt funktionierte, jedoch nur auf dem Prototyp. Beim Versuch den Code auf die tatsächliche Wortuhr zu laden, schlug dies fehl, sprich es wurde nichts ausgeführt, die Pixels blieben abgeschaltet.

5.4 Lösungsansätze

Das Problem des zweiten Modus mit den Sonnenaufgang – und Sonnenuntergangszeiten konnte ich nicht anders lösen als momentan ein allgemeines Programm zu schreiben, dabei werden weder die Sonnenaufgang- und Sonnenuntergangszeit miteinbezogen noch werden die anderen Stunden speziell eingeteilt. Momentan stellt der zweite Modus eine sehr vereinfachte Variante des eigentlichen zweiten Modus dar, sprich die Zeit wird im gewöhnlichen Stundentakt ausgegeben.

Die IR – Fernbedienung wird im Moment ebenfalls weggelassen und es wird im Programm manuell geändert, dass der eine oder andere Modus durchlaufen werden soll. Die RTC könnte mit einem DCF77 Empfänger unterstützt werden, damit es keine Zeitverschiebung gibt. [31]

Das Programm spiele ich nun auf dem Prototyp ab, da es dort funktioniert. Den Prototyp habe ich so erweitert, dass das Zifferblatt ausgetauscht werden kann, somit können trotzdem beide Modi verwendet werden. Für die eigentliche Wortuhr begann ich einen Code mit einem neuen Ansatz zu schreiben, jedoch besteht auch dort noch ein Problem mit der Zeit. Momentan werden die Neopixels nacheinander eingeschaltet und diese, welche zur aktuellen Zeitangabe nicht

mehr benötigt würden, schalten nicht ab, sondern bleiben erleuchtet. So würde nach 24 Stunden das ganze Zifferblatt leuchten.

6 Diskussion

Die Ausführung dieser Arbeit erfolgte in den einzelnen Schritten, die in den letzten Kapiteln behandelt wurden. Die Vorgehensweise scheint im Moment für mich immer noch ziemlich logisch im Ablauf zu sein. Die Arbeit beginnt im Bereich des Abstrakten und wendet sich immer mehr dem Konkreteren zu. Grundsätzlich lässt sich in gewissen Bereichen nicht viel zur Methodik sagen. Ich informierte mich darüber, wie eine Wortuhr aussieht und aufgebaut ist, so wie ich mich auch zu den Zeitformulierungen informierte. Dies war reine Recherchearbeit.

Des Weiteren lässt sich aber sagen, dass meine Methode zur Konzeption der Zifferblätter zwischenzeitlich ziemlich komplex ausfiel, was mir im Moment der Konzeptionierung der Zifferblätter nicht so bewusst war. Erst bei der Verfassung des schriftlichen Teils fiel mir auf, wie vernetzt und aufeinander aufbauend meine Gedankengänge bei diesem Schritt waren. Trotz allem denke ich, dass die einzelnen Schritte in der Vorgehensweise gut nachvollziehbar sind und man dies selbst analog zu meinen Erklärungen umsetzen könnte.

Meine Methode zum Bau baute auf meinen Erkenntnissen zum allgemeinen Aufbau einer Wortuhr auf. Ich erstellte mir Skizzen, jedoch überlegte ich mir oftmals die nächsten Schritte erst, während ich noch mit einer Sache beschäftigt war. Hierbei könnte man vielleicht sagen, dass mehr Vorausplanung teilweise nicht schlecht gewesen wäre. Ich hatte nie eine Blockade, was den Bau angeht, jedoch zogen sich manche Arbeitsschritte etwas in die Länge und ich musste mir gut überlegen, wie ich den nächsten Schritt anging. Trotz allem verfolgte ich genau meinen Plan, den ich mir bereitgelegt hatte, schon nur mit dem Bau des Modells.

Beim Stichwort Modell will ich erwähnen, dass dies ein aufschlussreicher Arbeitsschritt war und mir gut dazu diente, den Bau des Endprodukts zu überdenken und allfällige Änderungen vorzunehmen. Genauso behauptete ich auch, dass es eine der besten Entscheidungen in dieser Arbeit war, einen etwas grösseren Prototyp zu bauen, der mir nun sehr zu Nutzen kam im Fall meines Problems mit dem Upload des Codes auf die eigentliche Wortuhr. Bei der Programmierung bin ich mir nicht sicher, wie ich es anders hätte angehen sollen.

Die Methode, mir Vorbilder für meinen eigenen Quellcode zu suchen, mich nebenbei über einzelne Funktionen zu informieren und immer wieder Testversuche vorzunehmen, erschien mir als selbstverständlich. Die Testversuche waren sicherlich notwendig, jedoch hätte der Ablauf womöglich etwas anders ausfallen sollen. Hierbei bauten alle meine nächsten Schritte in der

Vorgehensweise auf den einzelnen Testversuchen auf, somit war es ein ständiger Kreislauf von: Informationsbeschaffung (evtl. Lösungsansatz) – Schreiben eines Testcodes – Testversuch – (evtl. Problemstellung).

Es wäre womöglich besser gewesen, hätte ich mich zu Beginn schon darüber informiert, welche Probleme auftreten könnten und wie man diese beheben kann. Durch meine Arbeitsweise habe ich vermutlich ziemlich viel Zeit verloren.

Ich kann definitiv sagen, dass ich in ganz verschiedenen Bereichen Erkenntnisfortschritte gemacht habe. Ich habe einiges in Bezug auf Konzeptionierung einer Sache gelernt und wie ich am besten vorgehe. Dabei stellte ich fest, dass es sich lohnt, alles sauber und verständlich aufzuschreiben, eventuell auch Skizzen zu erstellen, auch wenn es nur für den Eigengebrauch verwendet wird. Es verhilft zu einem besseren Verständnis des eigenen Handelns, vor allem wenn gewisse Themen zu einem späteren Zeitpunkt erneut aufgegriffen werden. Vor dieser Arbeit befasste ich mich fast nicht mit Elektronik, daher habe ich mit dem Bau der Uhr viel neues Wissen erlangt. Trotz der Probleme bei der Programmierung kann ich behaupten, dass ich hier womöglich die meisten Fortschritte gemacht habe. Zuvor hatte ich mich längere Zeit nicht mehr damit befasst, da dies zu einem früheren Zeitpunkt mal ein wenig in der Schule behandelt wurde, nun jedoch nicht mehr. Zu Hause hatte ich selbst nie etwas in Angriff genommen. Mit diesem Arbeitsschritt konnte ich also mein altes Wissen auffrischen und mir einiges an neuem aneignen.

Es bestehen noch einige ungelöste Fragen, die bereits in den Problemstellungen und Lösungsansätzen der Programmierung aufgegriffen wurden. Ich muss sagen, dass es mir eigentlich unerklärlich ist, weshalb das eigentlich funktionierende Programm nur auf dem Prototyp funktioniert. Ich war schlussendlich aber sehr froh, dass ich überhaupt die Möglichkeit hatte, auf den Prototyp umzusteigen. In Zukunft würde ich die Testversuche eher auf dem Endprodukt, also auf der richtigen Wortuhr ausprobieren anstatt nur auf dem Prototyp. Ich denke, dies war ein Fehler, der mir zu spät bewusst wurde.

Ich habe während dieser ganzen Arbeit auch vieles bezüglich meiner anfangs gestellten Fragen erreicht, so kann ich nun sagen, dass ich durchaus dazu imstande bin, eine Wortuhr zu designen und zu konstruieren. Es ist möglich und ich habe es auch geschafft, zwei verschiedene Modi zu programmieren zwischen denen gewechselt werden kann. Die Sommerzeit- und Winterzeitkorrektur kann mithilfe einer Library ganz einfach berücksichtigt werden. Die selbstgebaute Uhr ist dazu imstande, die Uhrzeit korrekt auszugeben, jedoch kann es schon zu Abweichungen kommen.

Zusammenfassend denke ich, dass meine Arbeit trotz noch einiger ungelösten Fragen und Problemen sehr aufschlussreich und gewissermassen trotzdem erfolgreich war.

7 Schlusswort und Ausblick

Abschliessend zu dieser Arbeit kann ich sagen, dass es trotz des grossen Aufwands eine gute Erfahrung für mich war und ich grundsätzlich froh bin, dass ich auf diese Weise mein Wissen erweitern konnte. Rückblickend auf den gesamten Prozess stelle ich für mich fest, dass mein Interesse für die Informatik nicht nur eine Phase war, sondern tatsächlich mehr dahintersteckt. Ich konnte mich leicht mehrere Stunden vor den Laptop setzen und nach Problemen und deren Lösungsansätzen suchen, ohne dass dabei die Motivation verloren ging und ich aufhören wollte. Ich denke, da es immer wieder kleine Zwischenerfolge in der Programmierung gab, spornte mich dies zusätzlich dazu an, weiterzumachen. Dadurch, dass ich mich nun so lange Zeit mit diesem Thema auseinandergesetzt habe, bin ich soweit, dass ich mir gut vorstellen könnte, in Zukunft einen Weg in Richtung Informatik einzuschlagen. Ich bin also sehr zufrieden dieses Thema gewählt zu haben; ich würde es auch definitiv nochmals tun.

In näherer Zukunft will ich mich noch weiter mit den ungelösten Problemen auseinandersetzen und nach möglichen Lösungen suchen. Dies betrifft die Suche nach einer Funktion für die Ausgabe der Zeit für Sonnenaufgang und Sonnenuntergang, die Suche nach einer Lösung für die Bedienung mittels der IR – Fernbedienung und die Suche nach dem Problem bezüglich des Codes für die Wortuhr.

8 Danksagung

Diese Arbeit erforderte von mir viel Eigenengagement. Dennoch gibt es einige Personen, denen ich danken möchte, da sie im Prozess eine wichtige Rolle spielten.

Zuallererst bedanke ich mich herzlich bei meinem Betreuer Chris Weber, der mich immer mit guter Kritik, passenden Verbesserungsmöglichkeiten und Anmerkungen unterstützte. Bei Fragen konnte ich mich immer an ihn wenden und mich auf eine schnelle und hilfreiche Antwort verlassen.

Als nächstes bedanke ich mich bei Herrn Lukas Gantert. Er ermöglichte es mir, in der Werkstatt arbeiten zu dürfen, was eine unglaubliche Hilfe beim Bau der Uhr darstellte. Zu Hause wäre dies niemals in solch einem Rahmen möglich gewesen. Zusätzlich stand er mir auch immer zur Verfügung, um meine Fragen zu beantworten.

Zuletzt bedanke ich mich herzlich bei meiner Lateinlehrerin Frau Dorothea Patti. Durch ihre Hilfe bei der lateinischen Formulierung der Uhrzeit hat sich aus meiner Sicht eine Erleichterung der Arbeit ergeben und zugleich konnten unglückliche Formulierungsfehler vermieden werden.

9 Literaturverzeichnis

[1] Thomas H-AL et al., Wikipedia: Echtzeituhr (Typen: Hardware-Uhr)

<https://de.wikipedia.org/wiki/Echtzeituhr>; 15.10.2020

[2] Laser & Co. Solutions GmbH (Alexander Huwaldt): Allgemeines über Mikrocontroller - Mikrocontrollerprogrammierung lernen mit myAVR

<http://einsteiger.myavr.de/index.php?id=5>; 11.10.2020

[3] W. Dvorak et al., Wikipedia: Leuchtdiode

<https://de.wikipedia.org/wiki/Leuchtdiode>, 30.10.2020

[4] M. Bornemann, P. Henningfeld, Duden (2015) Schulgrammatik extra Latein, Bibliographisches Institut GmbH, Berlin, S.30

[5] F. Kremser: Tageseinteilung (Tages- und Nachteinteilung)

<https://www.latein-grammatik.at/Tageseinteilung.htm>; 15.10.2020

[6] NeonZero et al., Wikipedia: Hardware

<https://de.wikipedia.org/wiki/Hardware>, 28.10.2020

[7] Ebuss et al., Wikipedia: Fernbedienung (Infrarot – Fernbedienung)

<https://de.wikipedia.org/wiki/Fernbedienung>, 03.11.2020

[8] Volvodani: Und noch eine Wordclock

<https://www.instructables.com/Und-noch-eine-Wordclock/>, 24.03.2020

[9] H. Sattler: WordClock

<http://www.harald-sattler.de/html/wordclock.htm>, 24.03.2020

[10] Super Make Something (2015, 31.August): Arduino + Neopixel Word Clock – Super Make Something Episode 6 [Youtube]

<https://www.youtube.com/watch?v=6tMTQ7Khe90&pbjreload=101>

[11] SWAN.dp et al., Wikipedia: Mitteldichte Holzfaserplatte

https://de.wikipedia.org/wiki/Mitteldichte_Holzfaserplatte#:~:text=Die%20mitteldichte%20Holzfaserplatte%20oder%20mitteldichte,aus%20dem%20Bereich%20der%20Faserplatten., 04.11.2020

[12] turi, Ledstyles.de, die LED Community: WS2812 Probleme

<https://www.ledstyles.de/index.php?thread/23548-ws2812-probleme/&pageNo=1>, 05.03.2020

[13] B. Hanus (2004), «Der leichte Einstieg in die Elektrotechnik», 1. Auflage, FRANZIS Verlag, Poing, Kapitel 9, Der Ohmsche Widerstand, S. 104

[14] A. Schommers (1991), «Elektronik – gar nicht schwer, Bd. 1, Experimente mit Gleichstrom», 6. Auflage, Elektor Verlag GmbH, Aachen, Kapitel 18, Kondensatoren, S. 140 ff.

[15] A. Schommers (1991), «Elektronik – gar nicht schwer, Bd. 1, Experimente mit Gleichstrom», 6. Auflage, Elektor Verlag GmbH, Aachen, Kapitel 2, Strom und Spannung, S. 16 f.

[16] Wikipedia: Jump wire

https://en.wikipedia.org/wiki/Jump_wire, 03.11.2020

[17] P. Burgess, Adafruit: Arduino Library Use

<https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use>, 03.11.2020

[18] (Autor unbekannt): Arduino Programmier – Handbuch (S. 4 f.)

http://www.netzmafia.de/skripten/hardware/Arduino/Arduino_Programmierhandbuch.pdf,
03.11.2020

[19] Poldi et al., Wikipedia: C++

<https://de.wikipedia.org/wiki/C%2B%2B>, 03.11.2020

[20] Makerblog.at – Arduino & Co (2016, 13. November): Arduino mit WS2812B Neopixel RGB LED Streifen [Youtube]

https://www.youtube.com/watch?v=UVISnxXh_VY

[21] Arduino: Libraries

<https://www.arduino.cc/en/Reference/Libraries>, 04.11.2020

[22] PaintYourDragon et al., GitHub: Adafruit_NeoPixel-examples-strandtest-strandtest.ino

https://github.com/adafruit/Adafruit_NeoPixel/blob/master/examples/strandtest/strandtest.ino,
11.03.2020

[23] AEQ-WEB (2019, 28. Januar): Infrarot Signale einer Fernbedienung decodieren mit dem Arduino und einem IR-Receiver [Youtube]

https://www.youtube.com/watch?v=93sX9izFq_g

[24] ArduinoLab (2016, 20. November): Fernbedienung am Arduino verwenden | Tutorial – ArduinoLab [Youtube]

<https://www.youtube.com/watch?v=kSPDHAQm9MA>

[25] Keebie81, instructables: Illuminated Neopixel Sign With IR Remote Control

<https://www.instructables.com/Illuminated-Neopixel-Sign-With-IR-Remote-Control/>,
03.03.2020

[26] Funduino: Nr.12 Fernbedienung - Eine Infrarotfernbedienung zur Ansteuerung von Arduino Mikrocontrollern verwenden

<https://funduino.de/nr-11-fernbedienung>, 03.03.2020

[27] Eric Ayars, GitHub: DS3231 – Examples - DS3231_test - DS3231_test.pde

https://github.com/tfg13/DS3231/blob/master/Examples/DS3231_test/DS3231_test.pde,
24.10.2020

[28] electromaker: Word Clock - Customisable And Easy To Build (Code)

<https://www.electromaker.io/project/view/word-clock-customisable-and-easy-to-build#code>,
24.10.2020

[29] SuperMakeSomething, GitHub: arduino-neopixel-word-clock/wordClockCode.ino

<https://github.com/SuperMakeSomething/arduino-neopixel-word-clock/blob/master/wordClockCode.ino>, 08.03.2020

[30] Bastelgarage.ch: MirrorClock Bausatz weiss made in Switzerland – Download zu MirrorClock

<https://www.bastelgarage.ch/bausatze/mirrorclock-bausatz-weiss-made-in-switzerland>

https://download.bastelgarage.ch/Bausaetze/03_MirrorClock/, 14.5.2020

[31] Meine Arduino – Projekte: DCF77 Empfänger zur Uhrzeitsynchronisation

<https://arduino-projekte.webnode.at/meine-projekte/dcf77-empfaenger/>, 18.03.2020

10 Abbildungsverzeichnis

Abb. 1: Beleuchtete Wortuhr.....1	Abb. 25: Litze.....25
Abb. 2: Beispiel einer deutschen Wortuhr.....9	Abb. 26: Grundplatte Prototyp.....26
https://fablab-karlsruhe.de/2017/11/21/wortuhr/	Abb. 27: Matrix.....26
Abb. 3: Analyse der deutschen Wortuhr.....9	Abb. 28: Rahmen, Raster, Ecken.....27
Abb. 4: Feste Ausdrücke 1 (Modus 1).....10	Abb. 29: Zifferblatt.....27
Abb. 5: Feste Ausdrücke 2 (Modus 1).....10	Abb. 30: Rückseite Zifferblatt.....28
Abb. 6: Ausdrücke Minutenangaben.....11	Abb. 31: Zifferblattschablonen.....30
Abb. 7: Spezialausdrücke Minutenangaben...11	Abb. 32: Transparentpapier.....30
Abb. 8: Ausdrücke Stunden (Modus 2).....12	Abb. 33: Grundplatte mit Raster.....30
Abb. 9: Formel für Zeilenlänge.....13	Abb. 34: Pixels und innerer Rahmen..30
Abb. 10: Feste Ausdrücke 1 (Modus 2).....15	Abb. 35: Fertige Matrix.....30
Abb. 12: Feste Ausdrücke 2 (Modus 2).....15	Abb. 36: Aussenrahmenteile.....30
Abb. 13: Art der Standardwortuhr.....17	Abb. 37: Schiene.....30
http://www.traweg.ch/wortuhr.html	Abb. 38: Schiene, Aussenrahmen.....31
Abb. 14: Bau des Modells.....19	Abb. 39: Aussenrahmen fertig.....31
Abb. 15: «Neopixels und Litzen».....20	Abb. 40: Querschnitt.....31
Abb. 16: Elektronik, Rahmen, Raster.....20	Abb. 41: Rahmen mit Kerben.....31
Abb. 17: Fertiges Modell.....20	Abb. 42: Raster in Rahmen gesetzt 31
Abb. 18: Leuchtende Neopixels.....23	Abb. 43: Fertiges Zifferblatt.....31
Abb. 19: Neopixels am Streifen.....24	Abb. 44: Kabelkanäle.....31
Abb. 20: RTC (DS3231).....24	Abb. 45: Fertige Uhr, Frontscheibe....31
Abb. 21: IR – Fernbedienung.....24	Abb. 46: Sketch im Hauptprogramm...41
Abb. 22: Arduino Nano.....25	
Abb. 23: Widerstand.....25	
Abb. 24: Elko – Kondensator.....25	

Alle Abbildungen ausser 2 und 13 sind eigene Abbildungen.

11 Anhang

Anhang 1: Zifferblatt Prototyp Modus 1

Anhang 2: Zifferblatt Prototyp Modus 2

Anhang 3: Zifferblatt Modus 1

Anhang 4: Zifferblatt Modus 2

Anhang 5: Materialliste

Anhang 6: Bauplan Prototyp

Anhang 7: Bauplan Wortuhr

Anhang 8: Schaltplan

Anhang 9: Quellcode

I. Zifferblatt Prototyp Modus 1

●	E	S	T	A	●
H	O	R	A	P	M
A	M	I	I	V	I
I	I	X	I	I	I
M	I	N	U	T	A
V	Q	E	T	Q	V
●	X	X	X	L	●

II. Zifferblatt Prototyp Modus 2

M	A	N	E	A	M
V	E	S	P	E	R
N	O	X	I	P	M
H	O	R	A	I	I
I	V	I	X	I	I
V	I	G	I	L	A
I	H	O	R	A	I

III. Zifferblatt Modus 1

E₀	S₁	T₂	A₃	H₄	O₅	R₆	A₇	E₈	N₉	O₁₀	N₁₁	A₁₂
P₂₅	R₂₄	I₂₃	M₂₂	A₂₁	U₂₀	N₁₉	D₁₈	E₁₇	C₁₆	I₁₅	M₁₄	A₁₃
S₂₆	E₂₇	C₂₈	U₂₉	N₃₀	D₃₁	A₃₂	T₃₃	E₃₄	R₃₅	T₃₆	I₃₇	A₃₈
Q₅₁	U₅₀	A₄₉	R₄₈	T₄₇	A₄₆	S₄₅	E₄₄	P₄₃	T₄₂	I₄₁	M₄₀	A₃₉
Q₅₂	U₅₃	I₅₄	N₅₅	T₅₆	A₅₇	E₅₈	S₅₉	E₆₀	X₆₁	T₆₂	A₆₃	E₆₄
E₇₇	T₇₆	D₇₅	U₇₄	O₇₃	D₇₂	E₇₁	C₇₀	I₆₉	M₆₈	A₆₇	E₆₆	T₆₅
O₇₈	C₇₉	T₈₀	A₈₁	V₈₂	A₈₃	E₈₄	M₈₅	I₈₆	N₈₇	U₈₈	T₈₉	A₉₀
E₁₀₃	N₁₀₂	Q₁₀₁	U₁₀₀	I₉₉	N₉₈	T₉₇	A₉₆	N₉₅	E₉₄	T₉₃	N₉₂	A₉₁
D₁₀₄	E₁₀₅	C₁₀₆	I₁₀₇	M₁₀₈	A₁₀₉	D₁₁₀	I₁₁₁	M₁₁₂	I₁₁₃	D₁₁₄	I₁₁₅	A₁₁₆
E₁₂₉	T₁₂₈	Q₁₂₇	U₁₂₆	A₁₂₅	D₁₂₄	R₁₂₃	A₁₂₂	N₁₂₁	S₁₂₀	A₁₁₉	U₁₁₈	T₁₁₇
A₁₃₀	U₁₃₁	T₁₃₂	V₁₃₃	I₁₃₄	C₁₃₅	E₁₃₆	S₁₃₇	I₁₃₈	M₁₃₉	A₁₄₀	E₁₄₁	T₁₄₂
D₁₅₅	T₁₅₄	R₁₅₃	I₁₅₂	C₁₅₁	E₁₅₀	S₁₄₉	I₁₄₈	M₁₄₇	A₁₄₆	Q₁₄₅	U₁₄₄	E₁₄₃
Q₁₅₆	U₁₅₇	A₁₅₈	D₁₅₉	R₁₆₀	A₁₆₁	G₁₆₂	E₁₆₃	S₁₆₄	I₁₆₅	M₁₆₆	A₁₆₇	T₁₆₈
Q₁₈₁	U₁₈₀	I₁₇₉	N₁₇₈	Q₁₇₇	U₁₇₆	A₁₇₅	G₁₇₄	E₁₇₃	S₁₇₂	I₁₇₁	M₁₇₀	A₁₆₉
D₁₈₂	O₁₈₃	D₁₈₄	R₁₈₅	A₁₈₆	N₁₈₇	S₁₈₈	N₁₈₉	A₁₉₀	M₁₉₁	E₁₉₂	P₁₉₃	M₁₉₄
				1₁₉₈	2₁₉₇	3₁₉₆	4₁₉₅					

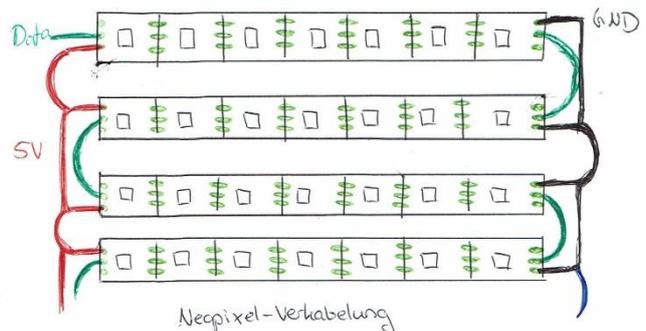
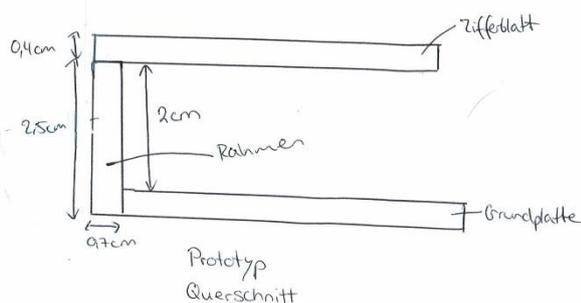
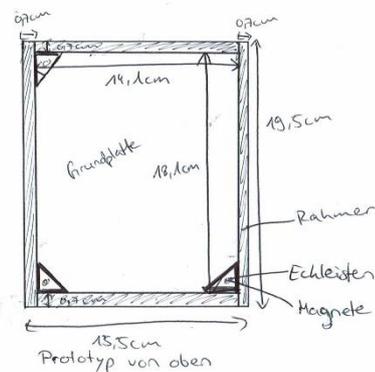
IV. Zifferblatt Modus 2

M₁₉₉	A₂₀₀	N₂₀₁	E₂₀₂	V₂₀₃	E₂₀₄	S₂₀₅	P₂₀₆	E₂₀₇	R₂₀₈	N₂₀₉	O₂₁₀	X₂₁₁
A₂₂₄	M₂₂₃	P₂₂₂	M₂₂₁	T₂₂₀	E₂₁₉	S₂₁₈	T₂₁₇	E₂₁₆	H₂₁₅	O₂₁₄	R₂₁₃	A₂₁₂
P₂₂₅	R₂₂₆	I₂₂₇	M₂₂₈	A₂₂₉	E₂₃₀	S₂₃₁	E₂₃₂	C₂₃₃	U₂₃₄	N₂₃₅	D₂₃₆	A₂₃₇
T₂₅₀	E₂₄₉	R₂₄₈	T₂₄₇	I₂₄₆	A₂₄₅	E₂₄₄	Q₂₄₃	U₂₄₂	A₂₄₁	R₂₄₀	T₂₃₉	A₂₃₈
V₂₅₁	I₂₅₂	G₂₅₃	I₂₅₄	L₂₅₅	I₂₅₆	A₂₅₇	Q₂₅₈	U₂₅₉	I₂₆₀	N₂₆₁	T₂₆₂	A₂₆₃
S₂₇₆	E₂₇₅	P₂₇₄	T₂₇₃	I₂₇₂	M₂₇₁	A₂₇₀	O₂₆₉	C₂₆₈	T₂₆₇	A₂₆₆	V₂₆₅	A₂₆₄
S₂₇₇	E₂₇₈	X₂₇₉	T₂₈₀	A₂₈₁	U₂₈₂	N₂₈₃	D₂₈₄	E₂₈₅	C₂₈₆	I₂₈₇	M₂₈₈	A₂₈₉
N₃₀₂	O₃₀₁	N₃₀₀	A₂₉₉	D₂₉₈	U₂₉₇	O₂₉₆	D₂₉₅	E₂₉₄	C₂₉₃	I₂₉₂	M₂₉₁	A₂₉₀
				H₃₀₃	O₃₀₄	R₃₀₅	A₃₀₆					

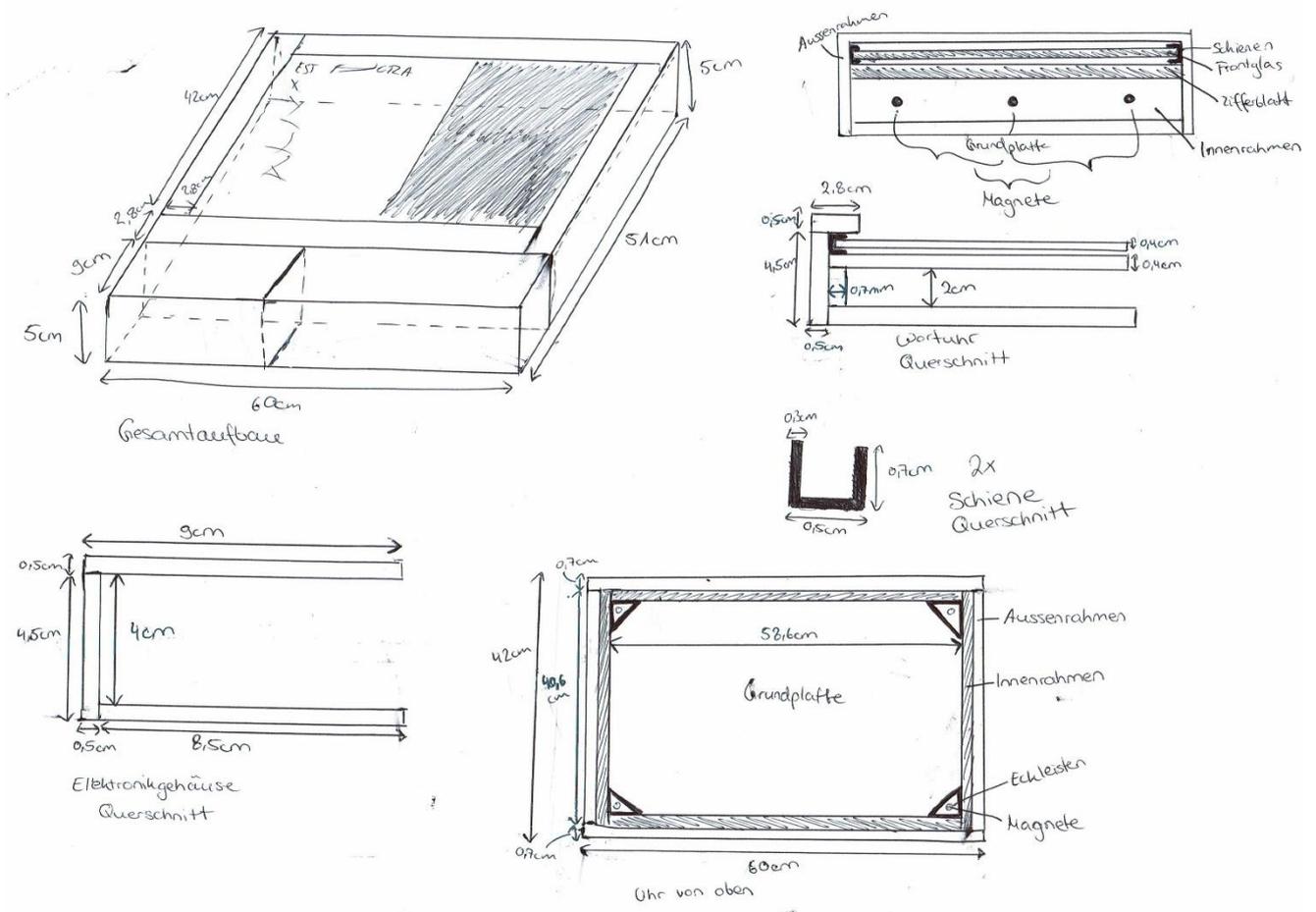
V. Materialliste

- MDF
- Gipsplatte
- Sperrholz
- Fotokarton (schwarz)
- Dreiecksleisten (20 x 20 x 1000 mm)
- Hobbyglas (4mm)
- Selbstklebefolie (schwarz-marmoriert, Opal, Milchglas)
- Superstarke Magnete (5 x 2 mm)
- Neopixels (WS2812B, 60 LEDs/m)
- Litzen (0.5mm², rot = 5V, schwarz = GND, grün-gelb = Daten)
- RTC (DS3231)
- IR – Fernbedienung inkl. IR – Receiver
- Arduino Nano
- Jumper wires
- Kleinteile (Widerstände, Kondensatoren)
- LötKolben und Lötzinn
- Werkzeug (Bohrmaschine, Säge, (Abisolier-)Zangen, Skalpell, Messinstrumente, Leim, etc.)

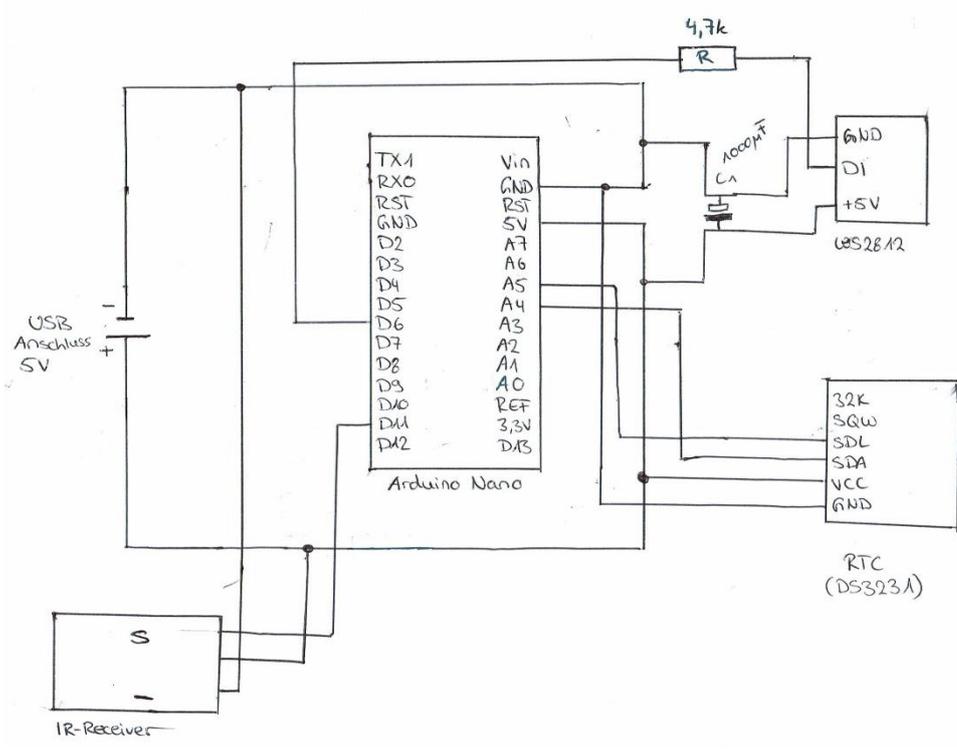
VI. Bauplan Prototyp

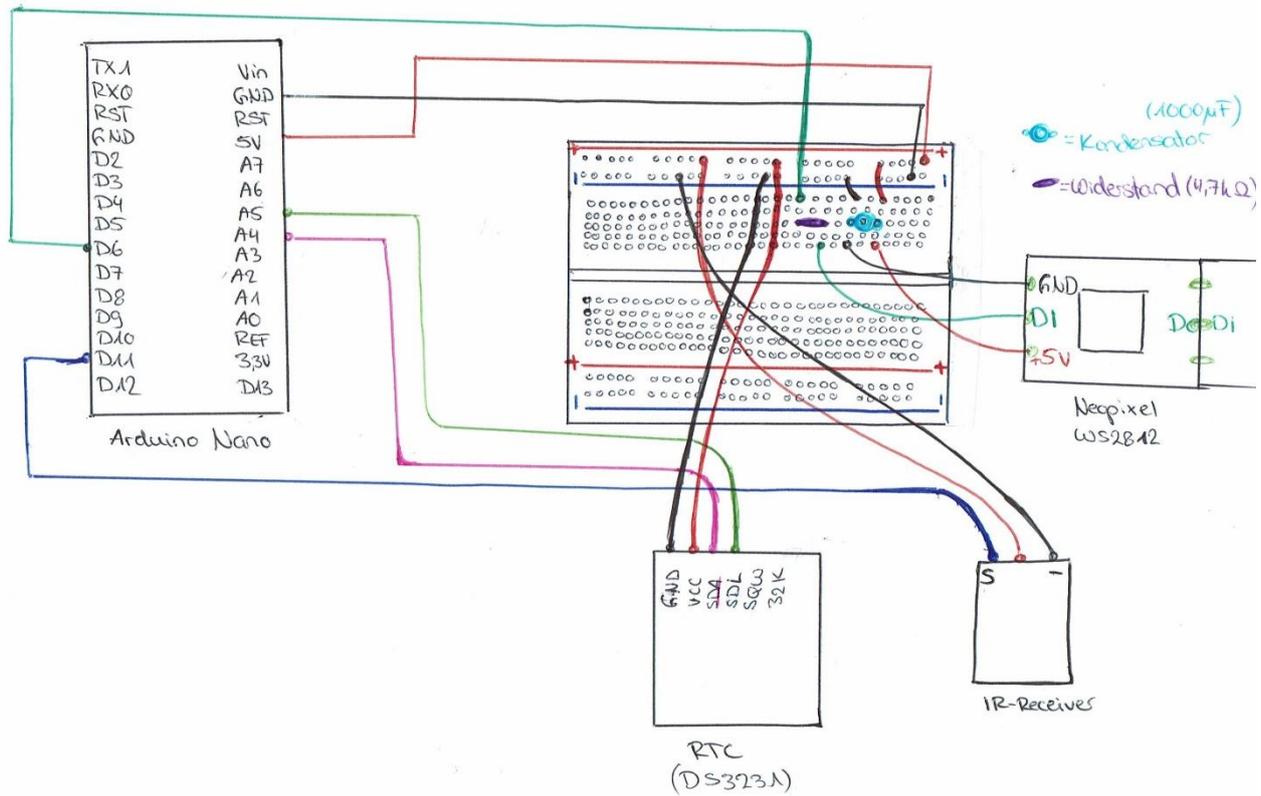


VII. Bauplan Wortuhr



VIII. Schaltplan





IX. Quellcode

Aufgrund der Grösse (wären etwa zusätzlich 20 Seiten gewesen) des Codes habe ich darauf verzichtet, den Quellcode hier im Anhang zu platzieren. Er lässt sich auf dem Datenträger nebst den anderen Dateien finden und aufrufen.

12 Einhaltung rechtlicher Vorgaben

Ich habe die Arbeit selbstständig und unter Aufsicht meines Betreuers verfasst und keine anderen als die angegebenen Hilfsmittel verwendet.

Ich nehme zur Kenntnis, dass meine Arbeit zur Überprüfung der korrekten und vollständigen Angabe der Quellen mit Hilfe einer Software (eines Plagiatserkennungstools) geprüft wird. Zu meinem eigenen Schutz wird die Software auch dazu verwendet, später eingereichte Arbeiten mit meiner Arbeit elektronisch zu vergleichen und damit Abschriften und eine Verletzung meines Urheberrechts zu verhindern.

Falls Verdacht besteht, dass mein Urheberrecht verletzt wurde, erkläre ich mich damit einverstanden, dass die Schulleitung meine Arbeit zu Prüfzwecken herausgibt.

Dietikon, 8. November 2020

Tamara Schneider