

Schweizerische Fachschule TEKO Zürich

Techniker HF Systemtechnik

Studienfach: Mikroprozessortechnik

Dozent: Christian Meier

Dokumentation Mikroprozessortechnik

Simon Says



Marc Mühlethaler

Klasse TEL/TIS-22

Abgabedatum: 16. Dezember 2024

Inhalt

HINTERGRUND DES PROJEKTS	3
MACHBARKEITSANALYSE	3
ARBEITSPLANUNG	3
TERMINPLAN	3
ARBEITSPROTOKOLL	4
PFLICHTENHEFT	5
MÖGLICHE ERWEITERUNGEN	5
BUDGET	5
SOFTWAREFLUSSDIAGRAMM	6
FINALVERSION	6
HARDWARE LAYOUT	8
MATERIALBESCHAFFUNG	8
STÜCKLISTE	9
SOFTWARE REGELN	10
SOFTWARE-DOKUMENTATION	10
VARIABLEN	10
FUNKTIONEN	10
LIBRARIES	12
ÜBRIGE 3RD-PARTY FRAGMENTE	12
<i>Sounds</i>	12
<i>Button Funktion</i>	12
VERSIONIERUNG	12
INBETRIEBNAHME & TESTPROTOKOLL	13
BEDIENUNGSANLEITUNG	13
HERAUSFORDERUNGEN	13
BUTTONREAKTION	13
SPIELMODUS / EINSTELLUNGEN	13
GEHÄUSEBAU	14
FAZIT	14
QUELLEN	14
ANHANG	15
ANHANGSVERZEICHNIS	15

Hintergrund des Projekts

Während meinem Lehrgang zum Techniker HF Systemtechnik an der schweizerischen Fachschule TEKO Zürich müssen wir im Fach Mikroprozessortechnik eine Projektaufgabe absolvieren. Es geht darum ein einfaches Projekt mit Mikroprozessoren umzusetzen. Dazu kann beispielsweise Arduino, ESP32 oder RaspberryPi benutzt werden. Da diese Aufgabe auch eine Vorbereitung auf die kommende Diplomarbeit sein soll muss sie als vollwertiges Projekt inklusive Projektplanung aufgezoogen werden.

Machbarkeitsanalyse

Nach einiger Recherche und Ideenfindung habe ich mich dazu entschieden ein **Simon Says** Spiel auf einem Arduino Mega2560 zu programmieren.

Simon Says ist ein Spiel, bei dem der Spieler eine immer länger werdende Sequenz aus farbigen Lichtern und dazu passenden Tönen wiederholen muss. Das Gerät zeigt eine Kombination an, die sich der Spieler merken und in der gleichen Reihenfolge durch Drücken der entsprechenden Tasten wiedergeben muss. Mit jeder Runde wird die Sequenz länger und schwieriger, bis ein Fehler gemacht wird. Ziel ist es, so viele Runden wie möglich korrekt zu meistern.

Solche Spiele sind bekannt und es gibt etliche Beispiele zur Umsetzung im Internet. Mein Ziel ist es aber dieses Spiel selbst zu programmieren und möglichst wenige fremde Codeblocks zu verwenden. So musste ich im Vorfeld testen, wie ich den Arduino genau ansprechen kann und ob meine Idee zur Umsetzung überhaupt funktioniert. Die für mich wichtigste Funktion zur Spiellogik ist das Befüllen und Vergleichen von Arrays. Wie dies funktioniert, wollte ich im Vorfeld austesten. So ist bereits im Vorfeld ein grosser Codeschnipsel entstanden, auf dem ich später aufgebaut habe.

Ich kam auch in Absprache mit dem Dozenten zum Schluss, dass das Projekt umsetzbar ist und dem Anspruch der Schule genügt.

Arbeitsplanung

Terminplan

Der Terminplan ist in mehrere Hauptphasen unterteilt. Darunter fällt die Organisation, die Entwicklung und die Dokumentation. Die einzelnen Arbeitsschritte sind den Hauptphasen unterstellt.

Da in diesem Ein-Mann-Projekt keine fixen Arbeitstage definiert sind, kann auch nicht genau prognostiziert werden, wann an der Umsetzung gearbeitet wird. Deshalb sind die meisten Arbeitspakete über einen Zeitraum von mehreren Tagen geplant. Obwohl sich Tätigkeiten überschneiden können, kann ein grundsätzlicher Ablauf der Entwicklung erkannt werden. Abhängigkeiten wurden im Terminplan nicht extra ausgewiesen, ergeben sich aber aus logischer Betrachtung. So kann zum Beispiel keine Hardware bestellt werden, wenn diese nicht zuvor definiert wurde.

Jedes Arbeitspaket verfügt über eine Aufwandabschätzung in Stunden. Dieser Aufwand soll im notierten Zeitraum geleistet werden. Zur Überprüfung der benötigten Zeitressourcen wird ein Arbeitsprotokoll erstellt. Die Aufwände aus dem Arbeitsprotokoll werden laufend in den Terminplan geschrieben, um so stets eine aktuelle Übersicht über den Stand des Projektes zu erhalten.

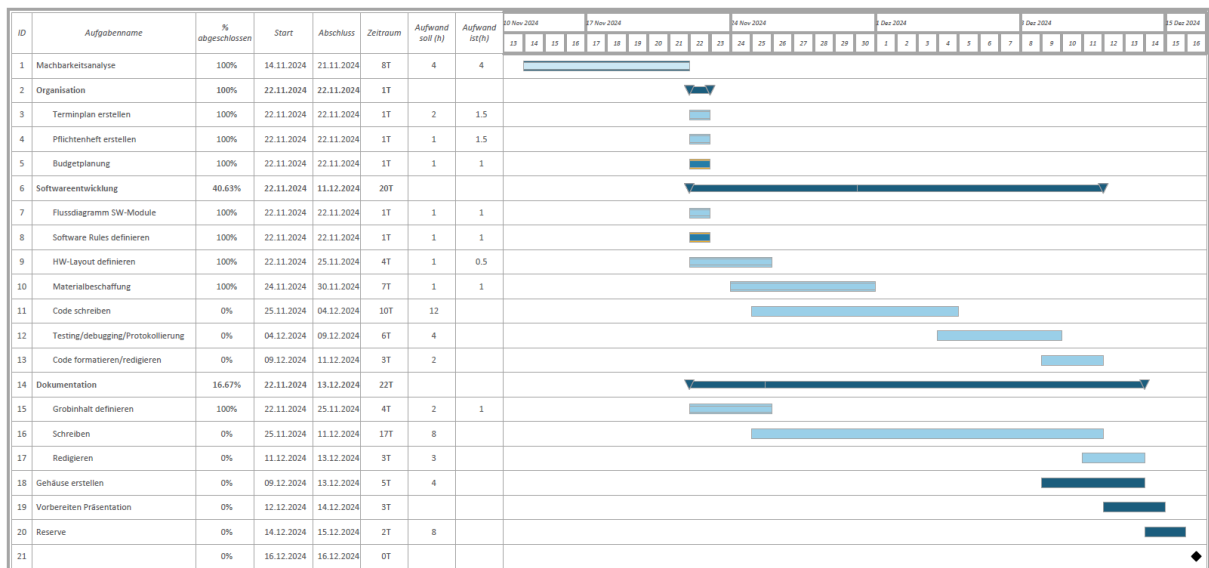


Abbildung 2: Terminplan stand 28.11.2024

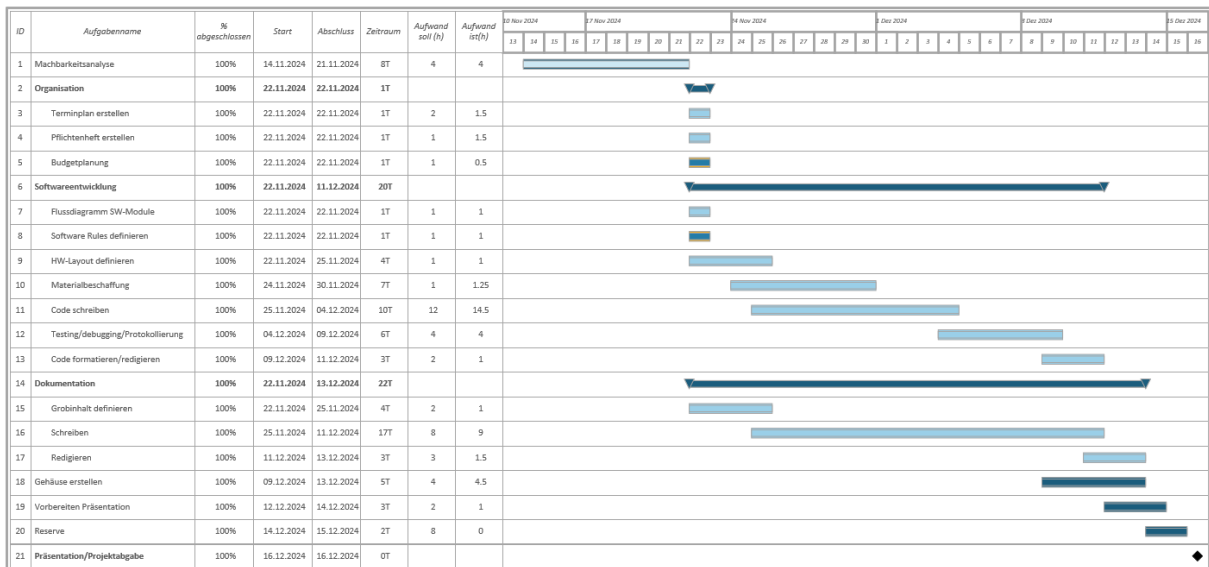


Abbildung 3: Terminplan nach Projektende

Arbeitsprotokoll

Auf dem Arbeitsprotokoll ([siehe Anhang](#)) wurden alle Arbeitsschritte nach Kategorie rapportiert. So kann nachvollzogen werden, wann wie viele Stunden für welches Arbeitspaket aufgewendet wurde. Die geleisteten Zeiten konnten so mit der ursprünglich geplanten Soll-Zeit im Terminplan verglichen werden. Daraus folgt wie viel Abweichung zwischen der ursprünglichen Planung und der effektiv geleisteten Arbeit resultiert.

Was auffällt ist, dass der Aufwand der Arbeitspakete mehrheitlich gut prognostiziert wurde. Einzig das Coding und die Dokumentation fiel aufwändiger aus. Ersteres liegt an den, während dem laufenden Projekt umgesetzten Zusatzfunktionen, auf die später noch eingegangen wird.

Der angesetzte Zeitplan jedoch hat sich etwas verschoben. So konnten die gemäss Terminplan gesetzten Zeiträume für die Umsetzung der Arbeitspakete nicht eingehalten werden. Auch dies ist begründet durch die Mehraufwände der umgesetzten Zusatzfunktionen. Es entstand eine ungefähre Verzögerung von 2 Tagen, die aber durch die geplante Reservezeitdauer abgefangen werden konnten.

Pflichtenheft

Aus dem Pflichtenheft ([siehe Anhang](#)) ist zu entnehmen, welche Funktionen bei der Planung des Projektes berücksichtigt wurden. Manche möglichen Erweiterungen, wie unten näher beschrieben, waren als Ideen schon zu diesem Zeitpunkt vorhanden und sind als "optional" kategorisiert ins Pflichtenheft mit eingeflossen.

Der Spalte "Erfüllt" ist zu entnehmen, ob die Funktionen zu Projektende umgesetzt werden konnten. Dies ist unter anderem abhängig vom Testprotokoll.

Bis auf eine Funktion wurden alle erfüllt. Die Latenz der Buttoneingabe (Testprotokoll T-014) könnte noch verbessert werden. Der Mangel wurde aber nur als "leicht" deklariert und somit nicht mehr verbessert, da dies gegebenenfalls hohen Coding-Aufwand bedeuten würde.

Mögliche Erweiterungen

Da der Zeitrahmen begrenzt ist und ich bei der Aufgabe mehr Relevanz auf die Ausführung der organisatorischen Bestandteile gelegt habe, konnte ich einige Ideen nicht in die Umsetzung des Endproduktes miteinberechnen. So sind die möglichen Erweiterungen bestenfalls optional dazu entstanden, aber nicht teil des Projektumfanges:

- Einbau in ein Gehäuse
- Schwierigkeitsstufen wählbar
- Zwei-Spieler Modus
- Spielmodus Spieler gegen Spieler
- Displayintegration
- Einstellungsmöglichkeiten (LED-Helligkeit, Lautstärke)
- Rekordanzeige
- Nonvolatiles Speichern der Einstellungswerte und Rekorde

Budget

Das Budget kann bis auf einen kleinen Teil ausser Acht gelassen werden.

Alles Material, welches es zum Erstellen dieses Projektes benötigt ist im von der Teko Zürich zur Verfügung gestellten «Starter Kit für Arduino» vorhanden. Deshalb ist kein Budget für Materialkosten nötig.

Auch Arbeitsressourcen werden nicht budgetiert, da diese nicht verrechnet werden.

Einzig Kosten, welche berücksichtigt werden, sind diejenigen für das optionale Material, welches für die optische Aufwertung des Endproduktes benötigt werden. Das Material kann anhand der Stückliste nachvollzogen werden. Die Kosten für das Zusatzmaterial sind mit CHF 50.- budgetiert.

Nach Abschluss des Projektes konnte festgestellt werden, dass das Budget mit CHF 34.50 für alle zusätzlichen Bauteile und ohne Verbrauchsmaterial und Werkzeug eingehalten werden konnte.

Softwareflussdiagramm

Mit dem Softwareflussdiagramm soll das enthaltene Programm auf einfache Art und Weise illustriert werden. Das erstellte und nachfolgend abgebildete Diagramm zeigt die zu Beginn des Projektes geplante Softwarearchitektur auf. Im Laufe des Projektes wurden diverse Zusatzfunktionen eingebaut, welche in diesem Bild nicht ersichtlich sind.

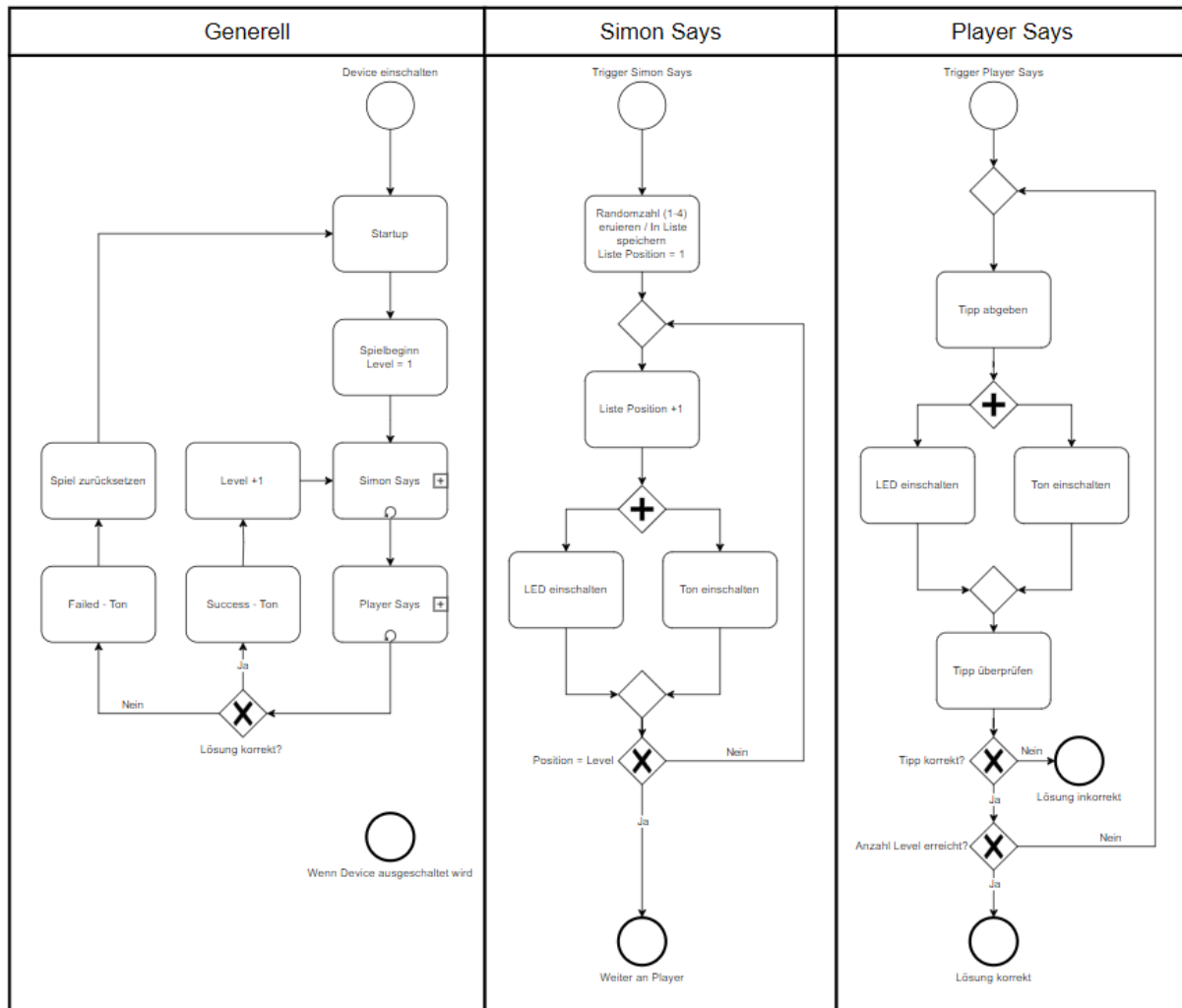


Abbildung 4: Softwareflussdiagramm V1.0

Finalversion

Wegen der vielen Änderungen im Projektverlauf wurde das Softwareflussdiagramm nach Abschluss des Projektes noch einmal aktualisiert. Es sind einige Vereinfachungen der Darstellung eingeflossen. So wird nicht mehr die Schaltung jedes LED's oder jedes Tonsignals vermerkt. Vielmehr wird dies im Prozessschritt markiert.

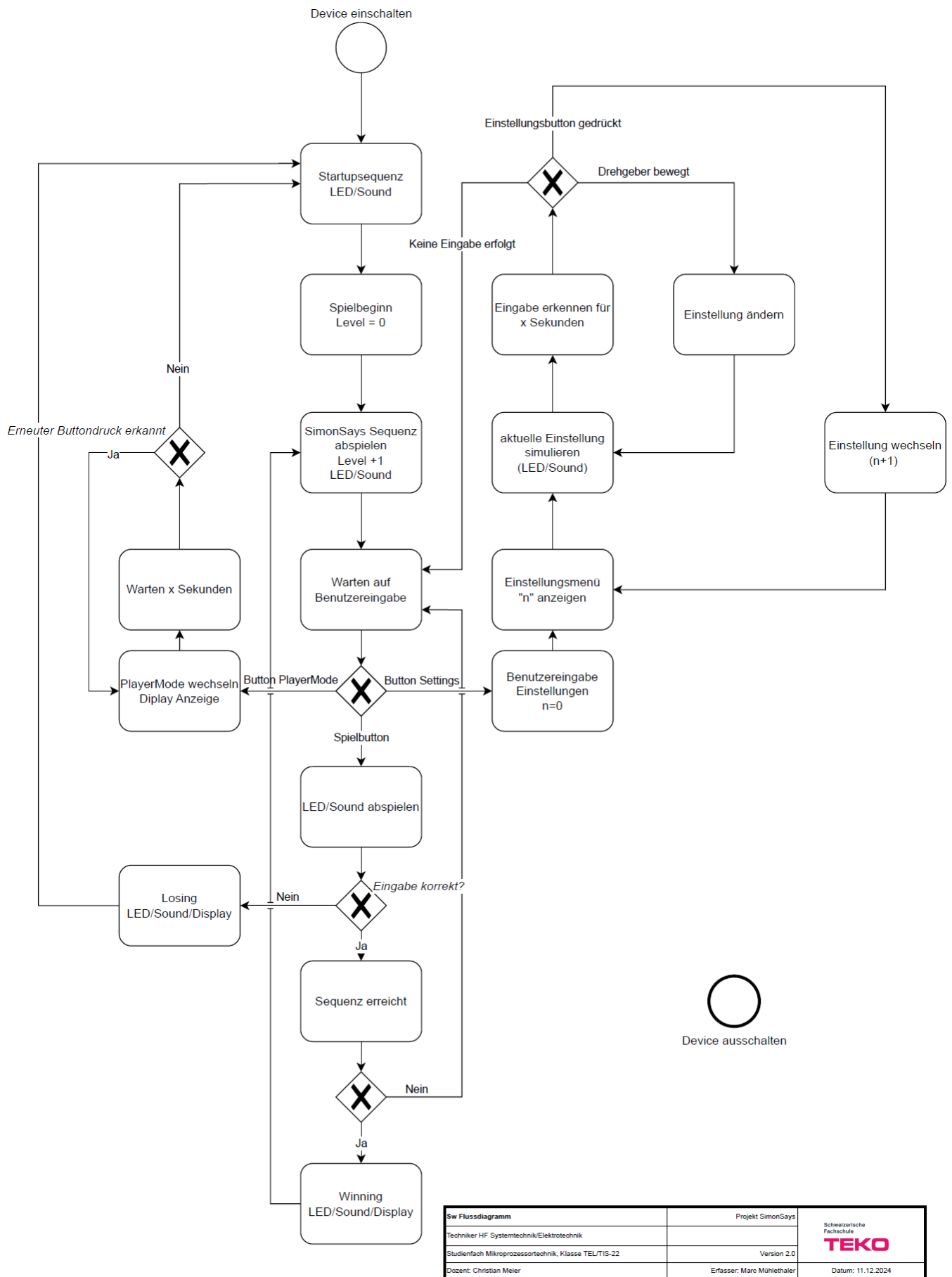
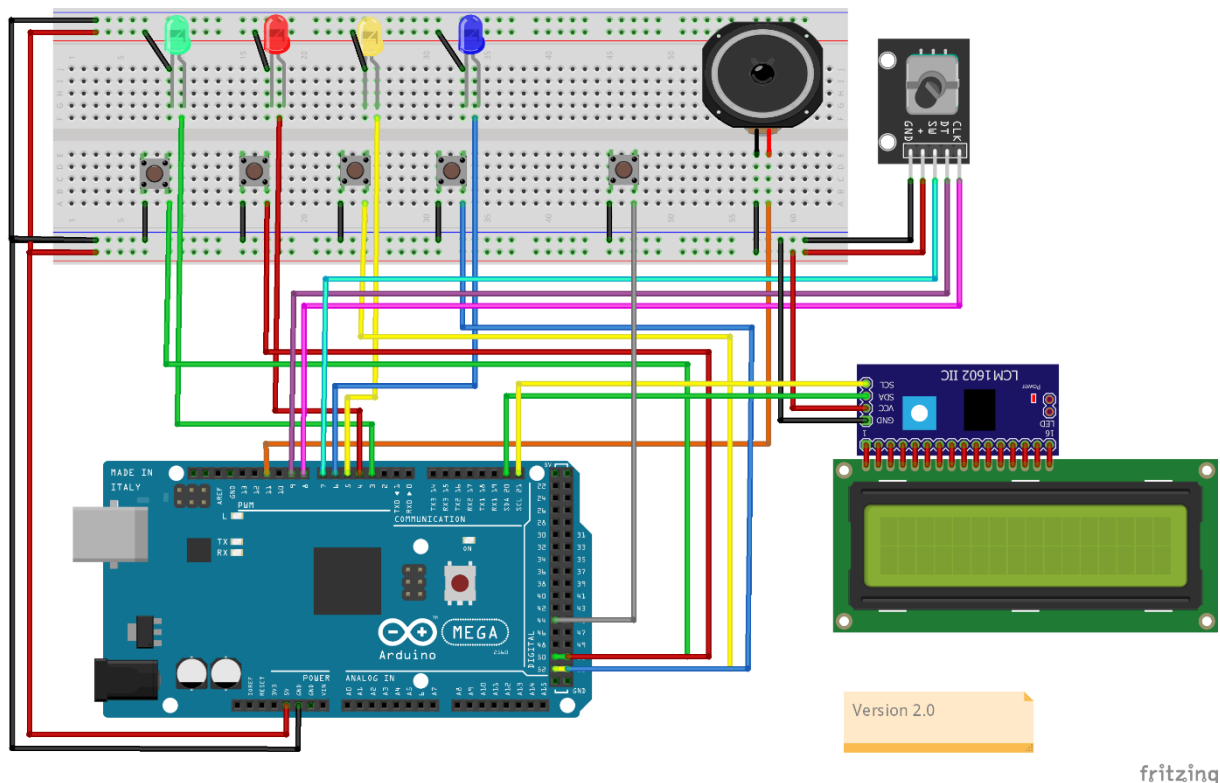


Abbildung 5: Softwareflussdiagramm V2.0

Hardware Layout

Mit Hilfe vom Elektronikschema soll aufgezeigt werden, wie die Bauteile untereinander verbunden werden müssen. Dies hilft zum einen die benötigte Hardware zu erfassen und auch im Fehlerfall eine effiziente Behebung zu gewährleisten.

Das nachfolgend aufgezeigte Hardware-Layout zeigt die Verkabelung des Endproduktes inklusive aller während dem Projekt hinzugefügten Bauteile auf. Der Übersichtlichkeit dienend wurde es in der Software "Fritzing" erstellt und mit einem generellen Breadboard umgesetzt. Die Endlösung verwendet zur Verbindung der einzelnen Komponenten andere Verbindungsmittel. Die Verschaltung der Elemente und PIN-Belegungen auf dem Arduino Mega 2560 entsprechen aber dem Original.



Materialbeschaffung

Nachdem das Hardware Layout definiert wurde, konnte auch der Materialbedarf benannt werden.

Die Auswahl des Lieferanten erfolgte nach nachfolgenden Kriterien, welche durch Internetrecherchen beurteilt werden konnten:

- Verfügbarkeit der Artikel
- Lieferdauer
- Preis
- Reputation
- Lieferkosten
- Lagerhaltung
- Alle Artikel aus einer Hand

Das Ziel war keine Artikel direkt aus dem asiatischen Raum zu importieren, sondern auf Waren zu setzen, welche sich bereits in der Schweiz befinden. So konnte die Lieferfrist klein gehalten werden. Natürlich mussten die Artikel am Lager verfügbar sein. Da das Budget bereits festgelegt wurde spielte auch der Preis eine wichtige Rolle. Unbekannte Verkäufer wurden nicht berücksichtigt.

Die Materialbeschaffung erfolgte schlussendlich wegen neuer Ideen während dem Projekt in zwei Schritten. Es wurden pro Bestellung je ein Lieferant berücksichtigt. Die bestellten Artikel und deren Bezugsorte können in der Stückliste nachvollzogen werden.

Stückliste

Anhand folgender Stückliste kann eingesehen werden, welches Material für die Umsetzung des Projektes benötigt wurde. Die für das Spiel benötigten Bauteile wurden teilweise dem «Starter Kit für Arduino» von Funduino entnommen. Um das Endresultat etwas ansehnlicher zu gestalten kann es zum Beispiel in ein Gehäuse montiert werden. Dazu benötigt man im besten Fall besser geeignete Buttons und Fassungen für die Elektrobauteile. Es folgt die gesamte Stückliste des fertigen Projektes:

Anzahl	Artikel	Bezugsort	Preis (CHF)
1	Arduino MEGA 2560	Starterkit Funduino	k.A.
1	Breadboard	Starterkit Funduino	k.A.
1	LED 5mm blau	Starterkit Funduino	k.A.
1	LED 5mm rot	Starterkit Funduino	k.A.
1	LED 5mm grün	Starterkit Funduino	k.A.
1	LED 5mm gelb	Starterkit Funduino	k.A.
4	Taster	Starterkit Funduino	k.A.
4	Widerstand 1kOhm	Starterkit Funduino	k.A.
1	LCD Display	Starterkit Funduino	k.A.
einige	Breadboardkabel	Starterkit Funduino	k.A.
1	EURO Kunststoff-Gehäuse Mittel Schwarz	Conrad	8.75
1	Arcade Button Micro grün	Conrad	2.20
1	Arcade Button Micro blau	Conrad	2.25
1	Arcade Button Micro rot	Conrad	2.85
1	Arcade Button Micro gelb	Conrad	2.85
4	LED Fassung Innenreflektor Chrom 5mm	Conrad	6.60
1	KY-040 Rotary Encoder Drehgeber	Bastelgarage.ch	3.90
1	Mini Lautsprecher 8Ohm 0.25W	Bastelgarage.ch	1.90
1	Einbau Drucktaster schwarz 7mm 2P	Bastelgarage.ch	1.40

Um alle Elemente zu verbauen und das Gehäuse zu präparieren wurde zudem folgendes Werkzeug und Verbrauchsmaterial verwendet:

- Bohrmaschine mit Bohrer (6mm, 7mm, 8mm, 20mm)
- Dekoupiersäge
- Schraubenzieher / Montageschrauben
- LötKolben / Lötzinn
- Klebstoff

Software Regeln

Softwareregeln dienen dazu Ordnung und Konsistenz sicherzustellen. Sie umfassen Vorgaben und Richtlinien, die bei der Entwicklung und Wartung von Software einzuhalten sind. Durch Vorgaben kann zum Beispiel die Teamarbeit, Kompatibilität, Fehlervermeidung und Wartbarkeit erhöht und somit die Qualität verbessert werden.

Ich habe folgende Regeln definiert:

Codestandards:

- Variablen in externem File (ausser Schleifenvariablen)
- Variablen wenn möglich gruppieren
- Variablen im camelCase Style (beginnen mit Kleinbuchstaben, haben Grossbuchstaben in der Mitte) und bezeichnen den Nutzen der Variable so gut wie möglich.
- Beispiel: «buttonPin» definiert den Pin für einen Button.
- Kommentare: Klar und prägnant die Codeabsicht dokumentieren
- Wiederholende Programmschritte werden in Funktionen geschrieben

Versionskontrolle:

- Im .ino-File wird nach jedem grösseren Arbeitsschritt die Version erhöht. Zudem wird notiert welche Änderungen vorgenommen wurden.
- GIT: Nach jedem Arbeitsschritt erfolgt ein git push.

Sicherheit:

- Code kann ohne zusätzliche Kontrolle deployed werden

Software-Dokumentation

Variablen

Alle verwendeten Variablen sind im File <arduino_config.h> abgespeichert. Dort ist auch deren Verwendungszweck auch genauer beschrieben

Funktionen

Um den Aufbau der Software besser verstehen zu können sind anschliessend die wichtigsten Funktionen und deren Verwendungszweck genauer beschrieben.

setup ()

Anfangswerte (Variablen, Pins und Bibliotheken) werden initialisiert.

loop ()

Die Funktion loop ist eine Endlosschleife, die nach jedem Durchlauf erneut aufgerufen wird. Sie wird verwendet, um das Spiel zu starten, die Computervorgabe abzuspielen und dann auf die Benutzereingabe zu warten. Je nachdem ob diese korrekt ist oder nicht wird angezeigt, ob das Level gewonnen oder verloren wurde. Danach wird entweder mit Erhöhung der Sequenz weitergefahren oder bei null begonnen.

printLCD(String lcdLine1, String lcdLine2 = "none")

Nimmt die auf dem Display auszugebenden Strings auf und printet sie. Wenn das Spiel gestartet ist und für die zweite Linie keine Eingabe erfolgt, werden die Variablen Level und Rekord ausgegeben.

startupSeq()

Gibt eine Ton und LED Abfolge als Startsequenz aus

simonPlays()

Die Funktion wird in jedem neuen Level durchschritten und speichert pro Durchlauf einen Zufallswert für die nächste Vorgabe der Sequenz in ein Array. Die gesamte Sequenz wird danach abgespielt. Dies ist die Spielsequenz die es danach vom Spieler zu lösen gilt.

checkUserSequence()

In dieser Funktion wird über die verschachtelte Funktion readButtons() geprüft ob die Benutzereingabe mit dem Array das in der Funktion simonPlays() generiert wurde übereinstimmt. Wenn ja wird sofern noch nicht die ganze Sequenz nachgespielt wurde, nochmals eine Eingabe abgefragt. Wenn eine fehlerhafte Eingabe gemacht wurde wird das Spiel mit der Funktion losing() beendet.

readButtons()

Es wird abgefragt ob und wenn ja welcher Button gedrückt wurde. Für die beiden Einstellungsbuttons sind weitere Funktionen verschachtelt.

settings(buttonPin)

In der Funktion settings() wird unterschieden ob der MultiPlayer- oder der Einstellungsbutton gedrückt wurde.
Multiplayer: mittels der Funktion changePlayerMode() wird der Spielmodus geändert. Der Button bleibt für gewisse Zeit aktiv und kann geändert werden, bevor das Spiel neu gestartet wird.
Einstellungen: mittels der Funktion rotary() können Einstellungen der Hardware vorgenommen werden.

changePlayerMode()

Hier wird der Zweispielermodus aktiviert oder deaktiviert.

rotary()

In dieser speziellen Funktion können Einstellungen der Hardware vorgenommen werden. Dies beinhaltet Helligkeit der LED's, Lautstärke der akustischen Signale und die Spielgeschwindigkeit. Pro Einstellung wird der neu gesetzte Wert jeweils simuliert und auf dem Display in Prozent-Schritten ausgegeben.

toneLED(int button)

Wann immer möglich wird dieses Codefragment benutzt um LED und Töne zu schalten

losing()

Notifikation via LED, Lautsprecher und Display über eine falsche Eingabe, die das Spiel anschliessend zurücksetzt.

winning()

Notifikation via LED, Lautsprecher und Display über eine korrekte Eingabe der gesamten Sequenz. Das Level wird um 1 erhöht.

Libraries

Um dieses Projekt umzusetzen, wurden folgende Libraries zu Hilfe genommen.

LiquidCrystal_I2C by Frank de Brabander

Die Bibliothek ermöglicht die Steuerung von I2C-Displays mit Funktionen, die denen der LiquidCrystal-Bibliothek sehr ähnlich sind.

RotaryEncoder by Matthias Hertel

Um den verwendeten Drehgeber KY-040 abzufragen wurde diese Library verwendet.

toneAC by Tim Eckel

Um das Feature der Lautstärkeneinstellung Software massig zu implementieren, wurde toneAC benutzt.

Übrige 3rd-party Fragmente

Sounds

Die Noten und Spielabfolgen für die akustischen Rückmeldungen wurden teilweise von Fazio Romadhona und seinem Simon Says Game übernommen. Insbesondere die Startabfolge, der Sound beim Meistern eines Levels und der Game-Over Ton habe ich kopiert.

Button Funktion

Mein erster Entwurf der Buttonfunktion war schlecht spielbar, da die Reaktionsgeschwindigkeit zu langsam ist. Ich habe die Idee von Urish übernommen, der auf der Webseite www.Wokwi.com sein Simon Says Game präsentiert. Dort wird in einer Schleife nur auf eine Benutzereingabe gewartet, was die Reaktionsgeschwindigkeit massiv verbessert. Meine Funktionen habe ich darum herum implementiert.

Versionierung

Da ich noch nicht viel Erfahrung bezüglich Softwareentwicklung habe, kenne ich auch den Prozess der Versionierung nicht wirklich. Ich habe dieses Projekt aber zum Anlass genommen meine ersten Schritte auf Github zu tätigen. Da ich aber nicht sehr viel Zeit zum Einlesen in dieses Thema hatte, habe ich mich auf die Grundfunktionen wie Hinzufügen und ändern von Dateien beschränkt. Jede grössere Änderung des Codes oder anderer relevanter Dateien hat ein Update auf Github erfahren mit den entsprechenden Kommentaren im commit.



Inbetriebnahme & Testprotokoll

Um die finale Funktionalität des Endproduktes zu testen, wurde ein Testprotokoll ([siehe Anhang](#)) erstellt. Dieses besteht aus Testfällen mit vordefinierten Testschritten und den dazugehörigen erwarteten Ergebnissen. Die Testfälle sind so definiert, dass nach ausführen aller Testfälle, die im Pflichtenheft genannten Funktionen garantiert werden können, sofern kein Mangel besteht.

Der Test wurde am 12.12.2024 mit nur leichten Mängeln abgeschlossen. Dank des Testprotokolls wurden einige kleinere Fehler im System gefunden, die aber noch während dem Ausführen des Tests behoben werden konnten. Beispielsweise wurde bei "T-007 Button Multiplayer" eine falsche Meldung auf den Display ausgegeben sobald das Gerät neu startete. Dies wurde durch eine kleine Codekorrektur direkt behoben und erneut getestet.

Das Ausführen des Testprotokolls am 12.12.2024 bedeutet somit auch, dass am Gerät bis zur Präsentation keine Änderungen mehr vorgenommen werden dürfen. Ansonsten müsste neu getestet werden.

Bedienungsanleitung

Üblicherweise erhält man zu einem gekauften Produkt eine Anleitung wie dieses zu verwenden ist. Auch bei diesem Projekt wurde eine Bedienungsanleitung ([siehe Anhang](#)) erstellt. Diese beinhaltet eine Funktionsbeschreibung über die Bestandteile des Geräts und eine Anleitung wie das Spiel zu verwenden ist.

Herausforderungen

Buttonreaktion

Leider war mit dem ursprünglich geplanten Code die Usereingabe nicht allzu schnell möglich. Gerade die ersten kurzen Sequenzen werden bei diesem Spiel schnell eingegeben. Dies war in den ersten Versionen wegen des überladenen loops nicht möglich.

Die Eingabegeschwindigkeit musste also verbessert werden. Aus Zeitgründen wurde die bereits erwähnte Buttonfunktion von Urish adaptiert.

Spielmodus / Einstellungen

Die bei der Buttonreaktion eingeführten Änderungen brachten zwar eine Verbesserung der Eingabeverzögerung, dann aber ein anders Problem zum Vorschein. Das Spiel sollte mit einem 2-Spielermodus attraktiver gestalten werden. Zudem sollte der Benutzer Einstellungen mithilfe eines Drehgebers verändern können. Diese Einstellung sollte zu jedem Zeitpunkt möglich sein, wenn nicht gerade der Computer etwas vorspielt. Dies führte dazu, dass im Code sehr viele Änderungen vorgenommen werden mussten, vor allem, um die zeitkritische Abfrage eines erneuten Buttonklicks zu bewältigen.

Auch die Abfrage des Drehgebers war kompliziert umzusetzen. Einstellungen sollten ohne grosse Latenz funktionieren, die gemachten Änderungen anzeigen und simulieren und nach einem Timeout, wenn nichts eingegeben wurde, wieder zum Spiel zurückkehren.

Dies konnte aus Mangel von Programmiererfahrung und Zeitgründen nur durch eine etwas unschöne und wohl komplizierte Codestruktur umgesetzt werden. Das Endergebnis allerdings funktioniert.

Gehäusebau

Der Bau des Gehäuses mit Haushaltsüblichen Werkzeugen war eine Herausforderung. Im speziellen die Ausschnitte für das Display und die Arduino-Anschlüsse sind nicht optimal gelungen. Dies wäre mit einer speziell dafür geeigneten Maschine auf CNC-Basis zu optimieren.

Fazit

Das einfache Spiel ist in der Basis simpel zu umzusetzen. Wenn man sich dann aber an Zeitangaben halten soll, die Arbeit im Projektmodus umsetzen möchte und während dem Verlauf der Umsetzung die Ansprüche an das Endprodukt erhöht, bedarf es schon ein bisschen mehr Aufwand.

Wenn man das fertige Spiel betrachtet und jemandem zum Spielen überlässt, kann man sich gar nicht vorstellen, wie viel Zeit in die Entwicklung der wenigen Features gesteckt wurde.

Ich konnte wieder viel neues dazulernen. Hier möchte ich besonders die Abwicklungsschritte eines Projektes, die Programmierung von Mikroprozessoren aber auch technische Details der verbauten Hardwarebestandteile benennen.

Würde ich das Projekt nochmals von vorne beginnen, müsste ich mir dazu Gedanken machen, ob ich wirklich so viel Zeit investieren möchte oder ich mich doch nicht lieber an das zu Beginn erstellte Pflichtenheft halten sollte. Die wenigen Zusatzfunktionen kosteten enormen Entwicklungs- und Testaufwand.

Ich bin mit dem fertigen Projekt aber sehr zufrieden und bin froh, die Extrameile gegangen zu sein. Ich habe grösstenteils selbstständig ein funktionierendes Spiel entwickelt, das in einer ansehnlichen Form daherkommt und ich meiner Familie und Freunden zum Bespielen überlassen kann.

Quellen

- > LiquidCrystal_I2C Library by Frank de Brabander
<https://docs.arduino.cc/libraries/liquidcrystal-i2c/>
- > toneAC Library by Tim Eckel
<https://bitbucket.org/teckel12/arduino-toneac/wiki/Home.>
- > RotaryEncoder Library by Matthias Hertel
<https://github.com/mathertel/RotaryEncoder>
- > Buttonfunktion Simon Says Game by "Urish"
<https://wokwi.com/projects/328451800839488084>
- > Sounds von: *Arduino Simple Simon Says Game* by Fazio Romadhona
<https://www.instructables.com/Arduino-Simple-Simon-Says-Game/>

Anhang

Anhangsverzeichnis

ANHANG 1: PFLICHTENHEFT	16
ANHANG 2: ARBEITSPROTOKOLL	18
ANHANG 3: TESTPROTOKOLL.....	20
ANHANG 4: BEDIENUNGSANLEITUNG	28

Pflichtenheft

22.11.2024

Mikroprozessortechnik Simon Says

Teko Zürich, Studiengang Systemtechnik, Marc Mühlethaler

Zielbestimmung

Das Endziel ist ein funktionstüchtiges Minispiel, das bekannt ist unter dem Namen "Simon Says". Der Computer gibt dem Spieler optisch eine Abfolge an Farbkombinationen vor, die der Spieler anschliessend wiederholen muss. Bei Spielbeginn startet die Sequenz bei einer Vorgabe. Schafft der Spieler es die Sequenz korrekt zu wiederholen steigt er ein Level auf und die Abfolge erhöht sich um eins. Schafft er es nicht startet das Spiel wieder bei einer Vorgabe. Computer und Spielereingaben werden optisch mittels farbiger LED und akustisch mittels Lautsprecherton quittiert. Zudem erfolgt eine optische und akustische Rückmeldung, wenn ein Level erreicht oder das Spiel verloren ist.

Anwendungsbereich

Das Produkt ist in erster Linie ein Übungsprojekt welches an der höheren Fachschule als Projektarbeit dient. Das Endprodukt kann zum kurzweiligen und spielerischen Zeitvertreib genutzt werden.

Zielgruppe

Alle Personen bei denen der oben erwähnte Anwendungsbereich zutrifft.

Betriebsbedingungen

Das Endprodukt kann von jedermann betrieben werden. Manipulationen an den elektrischen Bauteilen sind nur durch instruiertes Fachpersonal erlaubt. Kontakt mit flüssigen Substanzen ist nicht gestattet. Das Produkt darf nir mit der dafür vorgesehenen Stromversorgung benutzt werden.

Funktionen

Gruppe	Funktion	Gewichtung			13.12. Erfüllt
		Hoch	Niedrig	Optional	
Ausgabe	LED leuchtet	x			Ja
Ausgabe	Buzzer tönt je nach Farbwahl unterschiedlich	x			Ja
Spielverhalten	Levelerhöhung funktioniert	x			Ja
Spielverhalten	Computer spielt Sequenz vor	x			Ja
Eingabe	Eingabe gesperrt, wenn Computer am Zug		x		Ja
Eingabe	System reagiert auf Knopfdruck mit Anzeige von LED und Ton von Buzzer	x			Ja
Spielverhalten	Startmelodie verfügbar		x		Ja
Spielverhalten	System erkennt Eingabefehler	x			Ja
Ausgabe	System zeigt Fehler optisch und akustisch an		x		Ja
Ausgabe	System zeigt Erfolge optisch und akustisch an		x		Ja

Seite 1 von 2

Spielverhalten	Spiel beginnt bei Fehler von vorne	x			Ja
Eingabe	schnelle Benutzereingabe möglich	x			T-014
Einstellungen	LED-Helligkeit kann angepasst werden			x	Ja
Einstellungen	Lautstärke kann angepasst werden			x	Ja
Einstellungen	verschiedene Schwierigkeitsstufen wählbar			x	Ja
Ausgabe	Display zeigt aktuelles Level an			x	Ja
Ausgabe	Display zeigt aktuellen Rekord an			x	Ja

Qualität					
		Gewichtung			13.12.
Gruppe	Funktion	Hoch	Niedrig	Optional	Erfüllt
Programmcode	Für dritte verständlich		x		Ja
Dokumentation	Vorhanden	x			Ja

Benutzungsoberfläche					
		Gewichtung			13.12.
Gruppe	Funktion	Hoch	Niedrig	Optional	Erfüllt
Eingabe	Buttons zur Eingabe vorhanden	x			Ja
Ansicht	System in Chassis verbaut			x	Ja

Nichtfunktionale Anforderungen					
		Gewichtung			13.12.
Gruppe	Funktion	Hoch	Niedrig	Optional	Erfüllt
Programmcode	Variablen extern definiert	x			Ja
Dokumentation	Die Dokumentation enthält eine Stückliste		x		Ja
Dokumentation	Das System wird live demonstriert	x			Ja
Programmcode	Versionisierung auf Github			x	Ja

Technische Produktumgebung		
Funktion	Typ	Produkt
Software	Entwicklungsumgebung	Arduino IDE
Software	Elektronische Schaltungen	Fritzing
Hardware	Mikrocontroller	Arduino UNO MEGA 2560
Hardware	Elektronikbauteile	Gemäss Stückliste

Anhang 2: Arbeitsprotokoll

Arbeitsprotokoll

Mikroprozessortechnik: SimonSays
Marc Mühlethaler

1. Protokoll

Arbeitspaket	Anmerkung	Datum	Dauer (h)
Machbarkeitsanalyse	Codeversuche	14.11.2024	2
Machbarkeitsanalyse	Codeversuche	15.11.2024	1
Machbarkeitsanalyse	Codeversuche	17.11.2024	0.5
Machbarkeitsanalyse	Codeversuche	18.11.2024	0.5
Terminplan	Erstellt	22.11.2024	1.5
Flussdiagramm SW-Module	Erstellt	22.11.2024	1
Doku Grobinhalt	Beginn	22.11.2024	1
Pflichtenheft	Erstellt	22.11.2024	1.5
Software Rules definieren	erstellt	23.11.2024	1
HW-Layout definieren	Erstellt V1.0	24.11.2024	0.5
Materialbeschaffung	bestellt	24.11.2024	1.25
Doku schreiben	Stückliste, Machbarkeit	24.11.2024	2
Budgetplanung	Erstellt	25.11.2024	0.5
Code schreiben	V01.03.00 -Startup Sequenz mit Ton und LED -Button debouncer mit Millis -Winning und Losing Ton und LED Sequenz	26.11.2024	1.5
Code schreiben	V01.04.00 -richtiges Random mit randomSeed -Display mit Levelanzeige integriert -Multiplayer beginn	29.11.2024	3
Gehäuse erstellen	Gebaut	01.12.2024	3
Code schreiben	Multiplayer Button	02.12.2024	1
Code schreiben	Multiplayer Button Displayanzeige Einstellungsbutton	06.12.2024	4
Code schreiben	Multiplayer Button Displayanzeige Einstellungsbutton	07.12.2024	5
Gehäuse erstellen	Erweiterung Gehäuse mit 2 Buttons	08.12.2024	1.5
Testing/debugging/ Protokollierung	Finish der Displayanzeige Codeformatierung	09.12.2024	2
HW-Layout definieren	Hw Layout V2	09.12.2024	0.5
Doku schreiben	Sw Dokumentation Herausforderungen	09.12.2024	2.5
Doku schreiben	Bedienungsanleitung	11.12.2024	0.5
Testing/debugging/	Testprotokoll - Testcases definieren	11.12.2024	1.5

Seite 1 von 2

Protokollierung			
Code formatieren/redigieren	Code überarbeiten, kommentieren, Leichen entfernen	11.12.2024	1
Doku schreiben	Generelle Überarbeitung	12.12.2024	2
Testing/debugging/Protokollierung	Tests ausführen / Codekorrektur umsetzen	12.12.2024	0.5
Doku schreiben	Testprotokoll, Arbeitsprotokoll	13.12.2024	2
Doku Redaktion	Finales Layout, Korrekturlesen	13.12.2024	1.5
Vorbereiten Präsentation	Präsentationsablauf erstellen	15.12.2024	1

2. Aufwandsanalyse

Arbeitspaket	(h)	Zeit soll	Zeit ist	Differenz
Machbarkeitsanalyse		4	4	0
Terminplan		1.5	1.5	0
Pflichtenheft		1.5	1.5	0
Budgetplanung		0.5	0.5	0
Flussdiagramm SW-Module		1	1	0
Software Rules definieren		1	1	0
HW-Layout definieren		1	1	0
Materialbeschaffung		1	1.25	0.25
Code schreiben		12	14.5	2.5
Testing/debugging/Protokollierung		4	4	0
Code formatieren/redigieren		2	1	-1
Doku Grobinhalt		1	1	0
Doku schreiben		8	9	1
Doku Redaktion		3	1.5	-1.5
Gehäuse erstellen		4	4.5	0.5
Reserve		8	0	-8
Vorbereiten Präsentation		2	1	-1
Total:		55.5	48.25	-7.25

Testprotokoll

Mikroprozessortechnik Simon Says

Klassifizierung Öffentlich
Status in Arbeit
Projektname Mikroprozessortechnik Simon Says
Projektleiter Marc Mühlethaler
Version 1.0
Datum 11.12.2024
Auftraggeber TEKO Zürich
Autor Marc Mühlethaler

Änderungsverzeichnis

Version	Datum	Änderung	Autor
0.1	11.12.2024	Erstellung Testprotokoll	Marc Mühlethaler
0.1	11.12.2024	Beschreibung Testfälle	Marc Mühlethaler
1.0	12.12.2024	Tests ausgeführt	Marc Mühlethaler

Mängelklassen

Mängelklasse:

0 = mängelfrei

1 = belangloser Mangel

2 = leichter Mangel

3 = schwerer Mangel

4 = kritischer Mangel

Testfälle

ID / Bezeichnung	T-001	Stromversorgung
Beschreibung	Das Gerät lässt sich mit Strom versorgen und schaltet ein	
Testschritte	5V Batterie oder Netzteil an Batterieanschluss anschliessen	
Erwartetes Ergebnis	Gerät schaltet ein, Startprozess beginnt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-002	Display Hintergrundbeleuchtung
Beschreibung	Display Hintergrundbeleuchtung funktioniert	
Testschritte	Gerät mit Strom versorgen	
Erwartetes Ergebnis	Display Hintergrundbeleuchtung schaltet ein nach Stromversorgung	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-003	Startsequenz
Beschreibung	Bei Gerätstart wird die Startsequenz eingeleitet	
Testschritte	Gerät mit Strom versorgen	
Erwartetes Ergebnis	Gerät schaltet ein Displayausgabe: Starting Startton- und LED Sequenz beginnt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-004	LED
Beschreibung	Alle verbauten LED leuchten	
Testschritte	Abgedeckt mit T-003 Startsequenz	
Erwartetes Ergebnis	Alle LED haben während Startsequenz aufgeleuchtet	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-005	Audio Output
Beschreibung	Der Lautsprecher funktioniert	
Testschritte	Mit Startsequenz abgegolten	
Erwartetes Ergebnis	Lautsprecher hat während Startsequenz Ton abgespielt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-006	Button Input
Beschreibung	Alle Buttons funktionierten nach Design	
Testschritte	Gerät starten -> nach Startsequenz grüner Button drücken -> 20 Sekunden warten -> roter Button drücken -> 20 Sekunden warten -> gelber Button drücken -> 20 Sekunden warten -> blauer Button drücken -> 20 Sekunden warten -> MultiPlayer Button drücken-> 20 Sekunden warten -> Settings Button drücken	
Erwartetes Ergebnis	Farb-Buttons: Entsprechendes LED leuchtet auf, Tonsignal (für jeden Button unterschiedlich) wird ausgegeben MultiPlayer: Zeigt auf dem Display den aktuellen Player-Modus an Settings: Zeigt auf dem Display die Einstellung Helligkeit an	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-007	Button Multiplayer
Beschreibung	MultiPlayer Button funktioniert nach Design	
Testschritte	Wenn Benutzereingabe möglich Multiplayer Button drücken, innerhalb einiger Sekunden mehrmals drücken.	
Erwartetes Ergebnis	Bei jedem Drücken des Buttons ändert die Displayanzeige den Playermodus. Erfolgt drei Sekunden keine Eingabe startet das Gerät neu. Dies ist zu erkennen an der Startsequenz und dass wieder eine Sequenz abgespielt wird. Wenn 1 Spieler Modus ausgewählt wird nach der Vorgabe "Spieler am Zug" angezeigt. Wenn 2 Spieler Modus ausgewählt wird nach der Vorgabe "Spieler 1 am Zug" angezeigt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung	Display zeigt bei erstem Testdurchlauf fälschlicherweise eine Winningmeldung an nach Neustart. Dies wurde direkt behoben	

ID / Bezeichnung	T-008	Button Settings
Beschreibung	MultiPlayer Button funktioniert nach Design	
Testschritte	Wenn Benutzereingabe möglich Settings Button drücken, Wert verstellen, Button erneut drücken, Wert für Lautstärke verstellen, Button erneut drücken, Wert für Spielgeschwindigkeit verstellen. 10 Sekunden warten und die Schritte wiederholen	
Erwartetes Ergebnis	<p>Pro Buttonklick wird alternierend "Helligkeit", "Volume" und "Speed" sowie der aktuelle Einstellungswert in % angezeigt. Die jeweilige Einstellung wird am Gerät simuliert.</p> <p>Beim Drehen am Button wird die entsprechende Einstellung geändert. Dies wird wie folgt dargestellt:</p> <p>Helligkeit: LED leuchten im aktuellen Einstellungswert</p> <p>Sound: Es erfolgt eine Tonausgabe in der eingestellten Lautstärke</p> <p>Speed: Nachdem der Wert 1 Sekunde nicht verändert wurde erfolgt eine Spielsequenz mit LED und Ton der aktuellen Einstellung.</p> <p>Wird die Einstellung wiederholt wird der zuvor eingestellte Wert angezeigt und kann verändert werden</p>	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-009	Spielstand nach Setting
Beschreibung	Nach Einstellungen mittels Settings Button kehrt das Spiel zum aktuellen Level zurück	
Testschritte	3 Level durchspielen bis auf dem Display Level 4 angezeigt wird, Einstellungen vornehmen, warten bis wieder Eingaben möglich sind, Eingaben korrekt vornehmen	
Erwartetes Ergebnis	Nach den Einstellungen reagiert das Spiel wie zuvor eingestellt. Das Display zeigt Level 4 an. Nach 4 korrekten Eingaben wird die Winning Sequenz ausgelöst und das normale Leveldesign wird fortgesetzt.	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-010	Simon Says
Beschreibung	Korrektes Vorgeben der Sequenzen	
Testschritte	Spiel 5 runden korrekt durchspielen	
Erwartetes Ergebnis	Pro Runde hängt der Computer eine Ausgabe an die vorher gespielte Sequenz an	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-011	Spiellogik
Beschreibung	Korrekte Interpretation der vom Benutzer eingegebenen Versuchen	
Testschritte	<ol style="list-style-type: none">1. Gerät starten, gespielte Sequenz (nur 1) beobachten, falschen Button drücken.2. Sequenz beobachten, korrekte Eingabe ausführen3. Gerät starten, gespielte Sequenz (2) beobachten, ersten Button korrekt, 2. Button falschen drücken.	
Erwartetes Ergebnis	<ol style="list-style-type: none">1. Display gibt Game Over an, Losing Sequenz wird abgespielt, Spiel startet neu.2. Display gibt "Cool" aus und spielt die Winning Sequenz3. Ton und LED leuchtet beim ersten Button auf, beim zweiten falschen drücken gibt das Display Game Over aus, Losing Sequenz wird abgespielt, Spiel startet neu.	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-012	Levelanzeige
Beschreibung	Korrekte Anzeige des aktuellen Levels	
Testschritte	Mehrere Sequenzen korrekt eingeben und somit Levels erreichen	
Erwartetes Ergebnis	Im Eingabemodus erscheint das korrekte Level auf der 2. Displayzeile	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

ID / Bezeichnung	T-013	Rekordanzeige
Beschreibung	Korrekte Anzeige des aktuellen Rekordes	
Testschritte	Gerät einschalten, mehrere Sequenzen korrekt durchspielen, dann falsche Eingabe, mehr Sequenzen als vorher korrekt durchspielen.	
Erwartetes Ergebnis	Display zeigt in der zweiten Zeile unten rechts zuerst nur "Rec" an. Pro erreichtem Level wird die Anzeige erhöht. Wenn das Spiel verloren wurde bleibt der vorherige Wert erhalten bis entweder das Gerät neu gestartet wird oder eine Sequenz länger dauert als der aktuelle Rekord	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

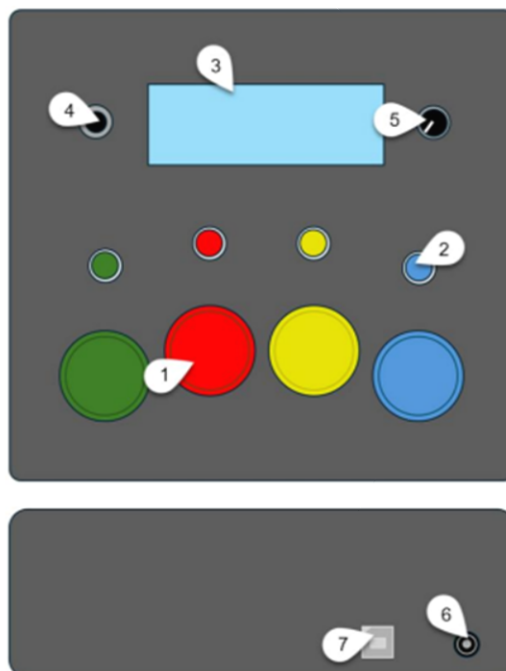
ID / Bezeichnung	T-014	Latenz bei Eingabe
Beschreibung	Buttons sind in einer angenehmen Eingabegeschwindigkeit zu drücken	
Testschritte	Mehrere Sequenzen durchspielen und die Buttons schnell nacheinander drücken	
Erwartetes Ergebnis	Jeder Buttondruck wird vom Spiel erkannt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	2	
Mangelbeschreibung	Bei geübten Spielern, die die Buttons sehr schnell drücken kommt es ab und zu vor, dass eine Eingabe nicht erkannt wird. Diese muss wiederholt werden.	

ID / Bezeichnung	T-015	Gehäuse
Beschreibung	Gehäuse ist solide verbaut	
Testschritte	Gehäuse gemäss Auflistung unten kontrollieren	
Erwartetes Ergebnis	<ul style="list-style-type: none">• Gehäuse fest verschlossen• Nichts klappert beim bewegen• Buttons, Display und LED's fest verschraubt	
Testdurchführung und Ergebnis		
Testdatum	12.12.2024	
Tester	Marc Mühlethaler	
Mängelklasse	0	
Mangelbeschreibung		

Bedienungsanleitung

Mikroprozessortechnik Simon Says

Teko Zürich, Studiengang Systemtechnik, Marc Mühlethaler



Bedien- und Anzeigeelemente:

- 1: Spielknöpfe
- 2: LED Anzeige
- 3: Display
- 4: Multiplayer Modus Knopf
- 5: Settings
- 6: Stromanschluss
- 7: Wartungsanschluss

Bedienung

Einschalten:

Einschalten: Gerät mit Stromversorgung verbinden. Hierfür entweder eine 5V Batterie oder das Netzteil verwenden

Ziel des Spiels:

Der Computer gibt eine Reihe an nachzuspielenden Farb- und Tonfolgen vor. Das Ziel des Spiels ist es die Sequenz mit Hilfe der Spielknöpfe nachzuspielen. Hat man eine Sequenz korrekt gespielt erhöht sich die Länge der Folge um eins. Wurde eine falsche Eingabe gemacht verliert man und das Spiel beginnt von neuem.

Multiplayer Modus:

Durch Drücken des Multiplayer Modus Buttons kann zwischen Ein- und Zweispieler Modus gewechselt werden. Das Gerät wechselt in den aktuell angezeigten Modus und startet neu so bald 5 Sekunden keine Eingabe mehr erfolgt.

Settings-Button:

Durch drücken des Einstellungs-knopf kann die Helligkeit der LED, die Lautstärke des Lautsprechers und die Spielgeschwindigkeit angepasst werden. Drücken des Knopfes ändert die Einstellungsmöglichkeit, drehen des Knopfes den jeweiligen Wert.