

PerfectPopCorn

Semesterarbeit
im Studiengang Informatik

Christian Bachmann
an der ZHAW Zürich

Betreuer: Peter Egli

1 Inhaltsverzeichnis

1	INHALTSVERZEICHNIS	1
2	EINLEITUNG	2
2.1	VISION.....	2
2.2	AUFGABENSTELLUNG.....	2
3	PROJEKTMANAGEMENT	4
3.1	PROJEKTPLAN.....	4
3.2	SOLL / IST ANALYSE.....	5
3.3	RANDBEDIENUNGEN	6
4	ANFORDERUNGSANALYSE	7
5	ANALYSE.....	7
5.1	ZU OPTIMIERENDE PROZESSE	7
5.2	EVA-PRINZIP	10
5.3	VERARBEITUNG VON MIKROFONSIGNALEN.....	11
5.4	TECHNISCHE UMGEBUNG	12
6	ANFORDERUNGEN.....	13
6.1	FUNKTIONALE ANFORDERUNGEN.....	13
6.2	NICHT FUNKTIONELLE ANFORDERUNGEN	15
7	DESIGN DER SOFTWARE	16
7.1	EINLEITUNG	16
7.2	DESIGN PATTERNS	16
7.3	ELEKTROBOARD (MIKROCONTROLLER-BOARD/INTERFACEPLATINE)	17
7.4	MOCKUPS	19
8	IMPLEMENTATION.....	20
8.1	TECHNOLOGIEN.....	20
8.2	ANSTEUERUNG DES MIKROFONS	21
8.3	FAST FOURIER TRANSFORMATION	22
8.4	AUSWERTUNG DES FREQUENZSSPEKTRUMS	22
8.5	SOFTWAREBAUSTEINE.....	23
8.6	ELEKTRISCHER SCHALTPLAN	24
8.7	APP-SCREEN.....	25
8.8	FOTOS DER TESTREIHEN.....	25
9	PROBLEME & LÖSUNGEN	26
9.1	ARDUINO WIFI SHIELD.....	26
10	FAZIT UND SCHLUSSWORT	27
	GLOSSAR	28
	LITERATURVERZEICHNIS	29
	BESTÄTIGUNG ÜBER DIE SELBSTÄNDIGKEIT	30

2 Einleitung

2.1 Vision

Verbrannte Mikrowellen Popcorns sollen der Vergangenheit angehören. Langweilige Wartezeiten vor der Mikrowelle können für Sinnvolleres genutzt werden.



Abbildung 1 Verbranntes Mikrowellen-Popcorn von Christian Bachmann

2.2 Aufgabenstellung

Nachfolgend ist die im EBS eingereichte Aufgabenstellung abgebildet.

2.2.1 Thema

Das perfekte Mikrowellen Popcorn erstellen durch Android App mit RealTime Soundanalyse und Internet of Things.

2.2.2 Ausgangslage

Im Detailhandel kann für die Mikrowelle Popcorn mit verschiedenen Zusätzen erworben werden. Je nach Mikrowelle und Umgebung wird die Popcornpackung verschieden lang aufgeheizt. Der Konsument kann da schon einmal 6 Minuten aber min. 2 ¼ Minuten vor der Mikrowelle ausharren. Dann gilt es denn richtigen Zeitpunkt zu finden um möglichst alle Popcorn aufgeplatzt und keine schwarzen Popcorns zu haben. Auf Social Media Kanälen können hunderte von Bilder von schwarzen Mikrowellen-Popcorns gefunden werden. Das mühsame Warten soll der Vergangenheit angehören und perfekte Popcorns sollen mit dieser Seminararbeit geschaffen werden.

2.2.3 Ziele der Arbeit

Das Endprodukt dieser Seminararbeit soll ein funktionierende Native Android App sein. Die App soll das Mikrofon des Telefons auslesen und den Input mittels Spektrum Analyse visuell darstellen. Die Mikrowelle soll, über die App ferngesteuert, den Strom entzogen werden können. Die Machbarkeit der Perfect Popcorn App mittels Soundanalyse soll geprüft werden. Wunschziel wäre es das Android Smartphone auf die Mikrowelle zu legen, mittels Mikrofon-Input eine Realtime Soundanalyse durchzuführen und den richtigen Zeitpunktes der Mikrowellenabstellung zu finden. Am gefundenen Zeitpunkt stellt die App stellt ferngesteuert die Mikrowelle ab.

2.2.4 Aufgabenstellung

- Erstellen einer Anforderungsanalyse
- Entwicklung einer Android App in Adobe Air
- Ansprechen des Android Mikrophone
- Verarbeitung des Mikrofon Inputs in Real Time
- Visualisierung des Mikrofon Inputs als einfache Spektrum Analyse
- Bau einer Elektronik um mittels Mikrocontroller 230V zuschalten
- Clientprogramm welches Befehle der App empfängt und den Mikrocontroller ansteuern kann
- Prüfen der Machbarkeit Soundanalyse für den richtigen Zeitpunkt der Mikrowellenabstellung und Beendung des Popcornhitzungsprozess
- Finetunning und Verbesserung

2.2.5 Erwartete Resultate

- Natives Android App verfügbar als APK
- Client Programm
- Physisch umgesetzte Elektronik
- Schriftlich festgehaltene Ergebnisse über die Machbarkeit
- Dokumentation

3 Projektmanagement

Das folgende Kapitel beschreibt die Einzelheiten der Projektplanung. Sie beinhaltet die Werkzeuge für die Umsetzung des Projekts, die Aufteilung in die verschiedenen Phasen, den Zeitplan und die Meilensteine sowie einige administrative Angaben.

3.1 Projektplan

Der Projektplan illustriert die Strukturierung des Projektes über die gut vier Monate lange Projektzeit.

Phase	März					April				Mai				Juni			
Woche	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
KickOff	x																
Aufgabenstellung		x															
Analyse der Optimierungsmöglichkeiten							x										
Implementation																	
Umsetzung der Basis-App																	
Implementation Möglicher Algorithmen																	
Darstellung App																	
Schaltplan / Aufbau Elektronik																	
Mikrocontroller Programmierung																	
Schnittstelle Mikrocontroller/App																	
Dokumentation																	
Präsentation																	
Milestones																	
	KickOff	Abgabe Aufgabenstellung	Websocket Kommunikation			Anforderungen aufgenommen					Elektronik funktioniert		Abgabe Draft	Applikation steht zur Verfügung		Präsentation	

Abbildung 2 - Projektplan

3.2 Soll / Ist Analyse

Auf der folgenden Tabelle ist die Soll / Ist Analyse der Aufwendungen für die Umsetzung des PerfectPopcorn Projekts abgebildet.

Das Erlernen von neuer Technologie wird nicht als Projektzeit gerechnet.

<i>Tätigkeit</i>	<i>Soll Stunden</i>	<i>Ist Stunden</i>
<i>Initialisierung</i>	3	2
<i>Dokumentation</i>	14	Zurzeit 15
<i>Analyse</i>	12	11
<i>Design und Architektur</i>	5	4
<i>Entwicklung/Elektronik</i>	20	21
<i>Vorbereitung Abgabe</i>	2	
<i>Reserve</i>	5	
<i>Total</i>	61	Aktuell 54

Tabelle 1- Soll-Ist Vergleich

3.3 RandBedienungen

3.3.1 Zeitlicher Rahmen

Der Offizielle Projektstart ist der 4. März 2015. Das Projekt muss bis spätestens 16. Juni 2015 abgegeben werden.

Der Aufwand für die Bearbeitung der Seminararbeit soll mindestens 50 Stunden umfassen.

3.3.2 Organisatorischer Rahmen

Nachstehend sind alle massgeblich involvierten Personen aufgeführt.

Person	Kontakt
Peter Egli, ZHAW (Lehrperson, Betreuer)	ZHAW Standort Zürich Lagerstrasse 41 / 8004 Zürich eglp@zhaw.ch
Dr. Reto Knaack, ZHAW (Studiengangs Leiter)	ZHAW Standort Zürich Lagerstrasse 41 / 8004 Zürich Reto.Knaack@zhaw.ch
Christian Bachmann, ZHAW (Student)	ZHAW Standort Zürich Lagerstrasse 41 / 8004 Zürich bachmch3@students.zhaw.ch

Tabelle 2 - Beteiligte Personen

4 Anforderungsanalyse

Dieses Kapitel beschreibt die Anforderungen an Perfect Popcorn. Die Anforderungen bilden die Basis für die Architektur, das Softwaredesign und die Implementation. Ihnen ist dem entsprechend ein sehr grosser Stellenwert zuzuschreiben. Um die Anforderungen zu definieren wurde zuerst eine Analyse durchgeführt.

5 Analyse

5.1 Zu optimierende Prozesse

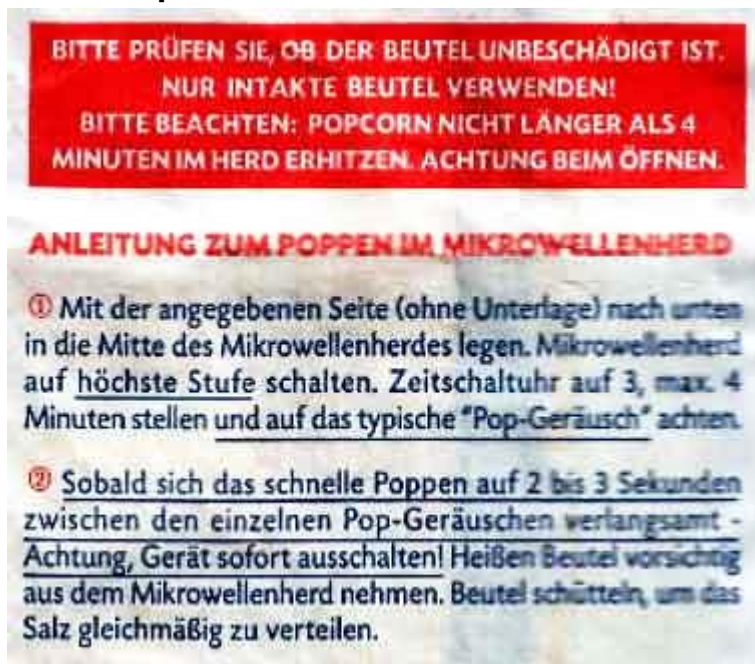


Abbildung 3 - Anleitung Zubereitung Mikrowellen Popcorn *Quelle chefkoch.de*

Der Zubereitungs-Prozess von Popcorn in einer Mikrowelle ist auf den meisten Packung definiert. Nun sollen die zu automatisierenden Teile dieses Workflows definiert werden. Dafür wird in einem ersten Schritt der gesamte Workflow des Zubereitungs-Prozess mit einem Ablaufs UML Diagramm visualisiert.

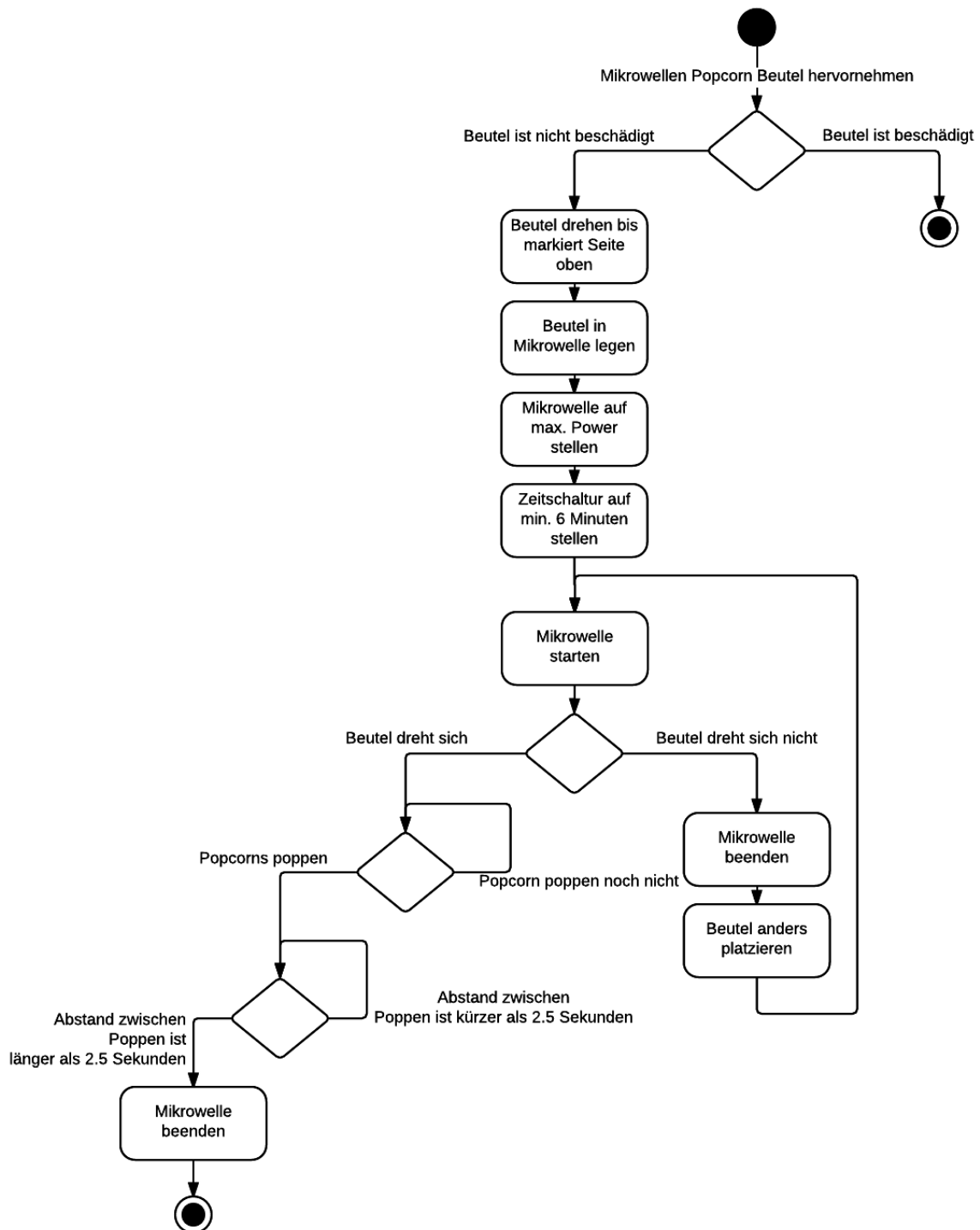


Abbildung 4 - Ablaufdiagramm Zubereitung Mikrowellen Popcorn

Die ersten Schritte, bis der Beutel von der Mikrowelle erwärmt wird, sind Straight Forward. Diese Schritte werden nicht als Probleme in Forums oder auf Social Media genannt. Diese offensichtliche Erkenntnis wird psychologisch unterstützt. Psychologische Studien haben gezeigt, dass die Fehlerraten bei Menschen besonders hoch sind, wenn ein Workflow durch Wartezeiten / Langweile unterbrochen wird.¹ Bei den ersten Schritten ist der Mensch beschäftigt und in Bewegung. Sowohl körperlich wie auch geistig gefordert.

¹ (Althof, 1999)

Während den folgenden Schritten, die Schritte des Popcorn-Wärmeprozess, steht oder sitzt der Mensch wartend vor der Mikrowelle. Seine Konzentration sinkt und die Gedanken schweifen ab. Die Versuchung etwas anderes während der Wartezeit zu erledigen ist gross. Durch Ausführen von anderen Prozessen sinkt die Konzentration und Priorisierung für den Popcorn-Wärmeprozess drastisch. Zusätzlich ist es schwierig einzuschätzen wie lange der Abstand zwischen dem Popcorn-Poppen ist. Insbesondere weil die Zeit sich während dem Warten gefühlsmässig verlängert. Der Mensch nimmt also kürzere Zeiten länger wahr. Durch das geht es länger bis der Mensch findet, dass das der Abstand zwischen den Pops nur noch 2.5 Sekunden beträgt.

Die in den letzten Sätzen erklärten Umstände werden durch das Teilen von negativen Erfahrungen in Forums oder Social Media untermauert. Sätze wie „Ich habe die Popcorns vergessen“ oder „Das waren niemals nur 2.5 Sekunden abstand. FAIL!“ sind Beispiele für Bildbeschreibungen von verkohlten Popcorns auf den verschiedenen Kanälen im Web.

Nun gilt es also diese zwei repetitiven Schritte maschinell zu optimieren.

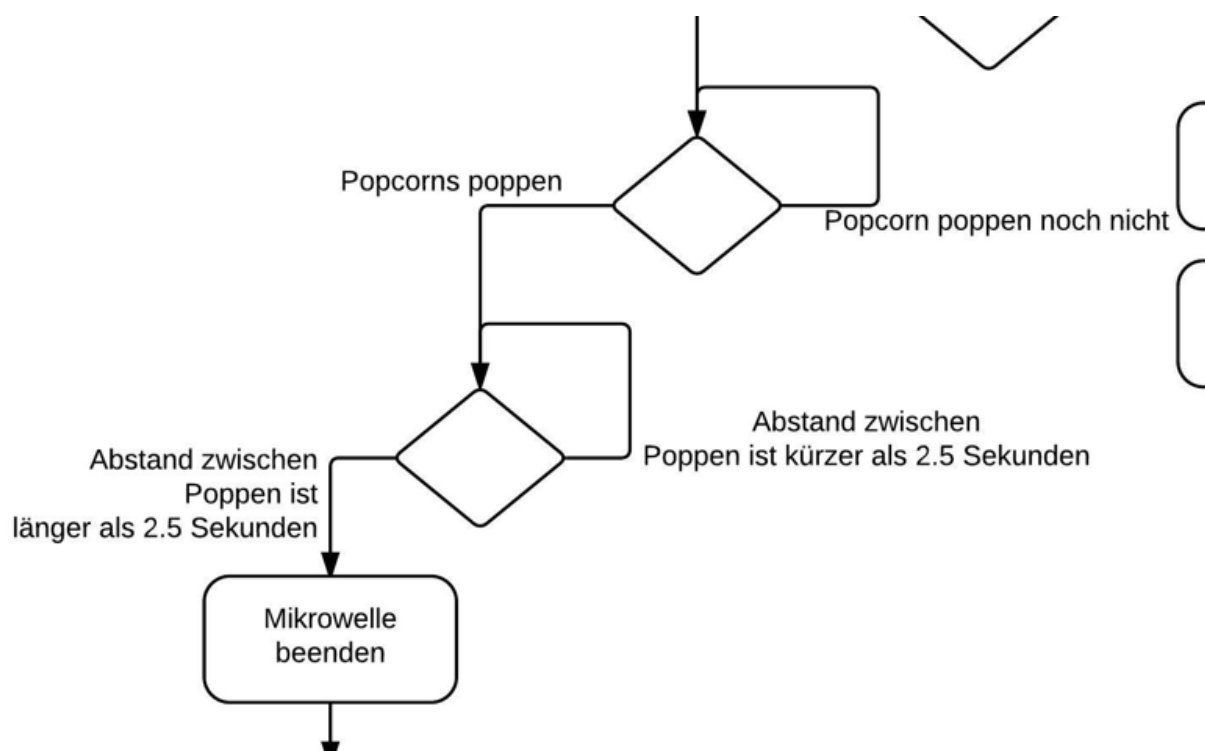


Abbildung 5 Zu optimierenden Schritte

5.2 EVA-Prinzip

Um den Prozess zu untersuchen bedienen wir uns dem EVA-Prinzip. Eingabe, Verarbeitung, Ausgabe. Wir analysieren wie der Mensch bei den Prozessen die Eingabe erhält wie er diese verarbeitet und welche Aktion der Mensch tätigt. Dabei erhoffen wir Erkenntnisse zu gelangen, wie es möglich ist die Prozesse maschinell zu optimieren und wie wir maschinell den Input erhalten, wie wir ihn maschinell verarbeiten und wir maschinell eine Ausgabe tätigen.

5.2.1 Eingabe

Ausgangslage des Prozesses ist, dass die Popcorns in der eingeschalteten Mikrowelle sind. Der Mensch nimmt irgendwann war, dass die Popcorns poppen und sollte dann wahrnehmen wenn der Abstand zwischen zwei Popps weniger als 2.5 Sekunden beträgt. Die Popcorns werden verschlossen in einem Beutel gewärmt. Dadurch kann einzig der Sinn Gehör das Poppen zuverlässig wahrnehmen. Alle anderen Sinne liefern keine zuverlässigen Inputs. Daher wird untersucht ob die Eingabe von der Maschine per Mikrofon aufgenommen werden könnte.

5.2.2 Verarbeitung

Die wahrgenommenen Geräusche werden vom menschlichen Gehirn gefiltert. Die Abstände zwischen werden mittels Zeitgefühl gemessen. Messungen von Abständen kann eine Maschine dank Time Clock genau vollziehen. Die Erkennung von Geräuschen ist da durchaus schwieriger und muss eingehender analysiert werden.

5.2.3 Ausgabe

Der Mensch drückt mit der Hand auf den Ausschalter. Um den Prozess 1:1 seitens Maschine auszuführen bräuchte man Robotics-Instrumente. Diese Variation können wir wegen dem zu hohen Aufwand direkt ausschliessen. Generell können wir das Ziel hinter der Ausgabe analysieren. Es geht darum die Mikrowelle auszuschalten als vom Strom zunehmen. Eine Maschine kann also durchaus den Strom unterbrechen. Eine weitere Variante wäre die Ausgabe mittels Alarm vom Menschen an die Maschine weiterzuleiten.

5.3 Verarbeitung von Mikrofonsignalen

Aus Kapitel 5.2.1 und 5.2.2 resultiert die Anforderung, dass PerfectPopcorn aus Mikrofonsignal entgegennehmen, verarbeiten und das Pop-Geräusch erkennen muss.

Wir bedienen uns für das Abbilden von Frequenzen des Audiosignals dem Spektrumanalysator. Dieses Gerät wird für Forschungszwecken erfolgreich eingesetzt. Der Spektrumanalysator möchten wir in einem weiteren Schritt mit der uns und in unserer Anwendern mit der uns zur Verfügung stehenden Hard- und Software umsetzen. Wir erhoffen uns weiter, dass ein Abbild des Pop-Geräuschs spezielle Merkmale aufweist um es von Umgebungsgeräuschen abzugrenzen.

5.3.1 Spektrumanalysator



Abbildung 6 - Spektrumanalysator *Quelle home.zhaw.ch*

Der Spektrumanalysator ist ein elektrisches Messgerät. Es stellt die erfassten Signale als Frequenzspektrum dar.

5.3.2 Spektrumanalysator FFT-Analyse

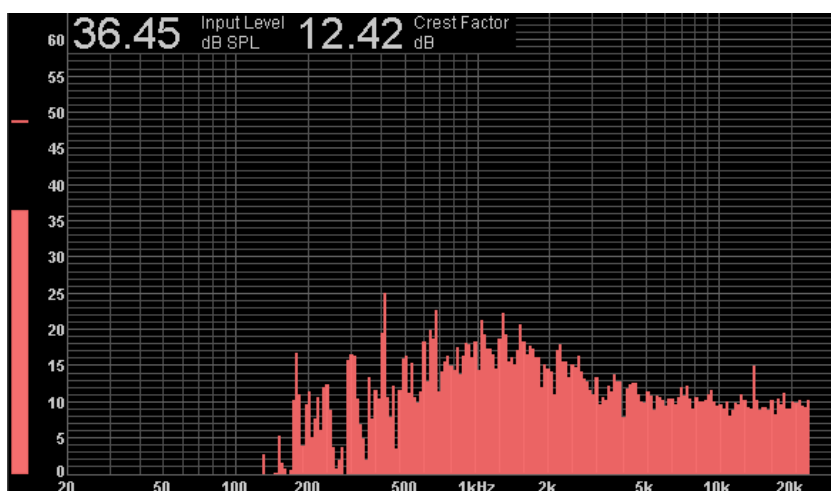


Abbildung 7 - Frequenzspektrum eines FFT-Analysators *Quelle www.tomshardware.de*

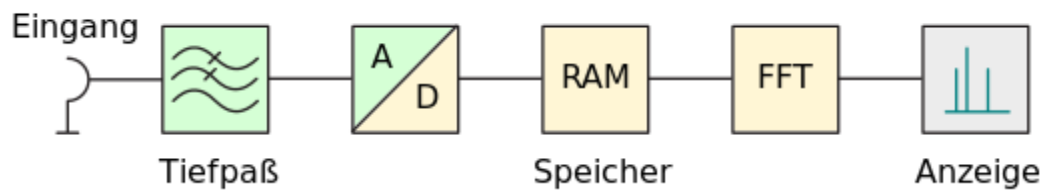


Abbildung 8 - Ablauf FFT-Analyse / Vereinfachtes Blockschaltbild *Quelle Wikipedia*

Der FFT-Analysator des Spektrumanalysators macht zuerst eine Tiefpassfilterung, um Störsignale zu vermeiden. Danach wird das analoge Signal in ein digitales Signal umgewandelt. Mit Hilfe der Fourier-Transformation kann aus den Daten das Frequenzspektrum errechnet werden. Üblicherweise wird die schnelle Fourier-Transformation verwendet. Daher auch die Abkürzung FFT für Fast Fourier Transformation. Die Ergebnisse der FFT werden danach visuell wiedergegeben. Beim dargestellten Frequenzspektrum ist die horizontale Achse, auch Abszisse genannt, die Frequenzachse und die vertikale Achse, auch Ordinate genannt, die Amplitude der Frequenz. Dadurch können die Frequenzen fast in Echtzeit analysiert werden. Dieser Vorgang wiederholt sich einige Male pro Sekunde.

Ein Spektrumanalysator soll nun innerhalb von PerfectPopcron nachgebaut werden.

5.4 Technische Umgebung

Im Kapitel 5.2 und 5.3 wurden die Anforderungen für die maschinelle Optimierung erörtert. Unser maschinelles System sollte daraus folgend per Mikrofon die Audiosignale entgegennehmen können, diese digitalisieren, eine FFT-Analyse durchführen, diese optisch darstellen und den Menschen über die fertigen Popcorns informieren und die Mikrowelle abstellen.

Ein Smartphone besitzt ein Mikrofon, CPU sowie Audioausgabe, um generell die ersten Punkte umzusetzen. Ob die Genauigkeit des Mikrofons und der Power der CPU für die FFT reicht, wird sich bei der Implementation zeigen.

6 Anforderungen

6.1 Funktionale Anforderungen

Die funktionalen Anforderungen legen die Funktionen von PerfectPopcorn fest. Die aus der Analyse resultierenden Anforderungen werden hier festgehalten.

Funktionale Anforderungen werden als *FREQ-Identifikation* bezeichnet

6.1.1 FREQ-101 Geräusche aufnehmen

Referenz	Kapitel 5.3
Beschreibung	Mittels Mikrofon können die Geräusche der Umgebung aufgenommen werden können
Techn. Risiko	Niedrig
Business Value	Hoch

6.1.2 FREQ-102 Durchführen einer Fast Fourier Transformation

Referenz	Kapitel 5.3
Beschreibung	Die Geräusche können mittels Fast Fourier Transformation ausgewertet werden können
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.3 FREQ-102 Darstellung des FrequenzSpektrums

Referenz	Kapitel 5.3
Beschreibung	Das Resultat der FFT soll als Frequenzspektrum dargestellt werden
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.4 FREQ-122 Bestimmen des Popcorn Popens

Referenz	Kapitel 5.3
Beschreibung	Anhand der FFT kann das Popcorn Popen bestimmen werden
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.5 FREQ-124 Visualisieren des Popcorn Popens

Referenz	Kapitel 5.3
Beschreibung	Das Popcorn Popen kann dem Anwender visuell dargestellt werden
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.6 FREQ-130 Bestimmen wenn der Erwärmungs Prozess beendet

Referenz	Kapitel 5.3
Beschreibung	Sobald der Abstand zwischen zwei Popcorn Pops weniger als 2.5 Sekunden beträgt muss die Beendigung des Erwärmungsprozess angestossen werden
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.7 FREQ-132 Kommunikation App<>Elektroboard

Referenz	Kapitel 5.3
Beschreibung	Das App muss das Beenden des Mikrowellenwärme Prozess an das Elektroboard senden. Das Elektroboard muss das App über den aktuellen Zustand informieren können.
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.8 FREQ-134 Darstellung des Elektroboardzustands

Referenz	Kapitel 5.3
Beschreibung	Die Zustände (Mikrowelle hat Strom, Mikrowelle hat kein Strom oder keine Kommunikation) müssen dargestellt werden
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.9 FREQ- 202 Web-Schnittstelle (REST API) Anzeige Status

Referenz	Kapitel 5.3
Beschreibung	Das Client-Programm muss eine Web-Schnittstelle zur Verfügung stellen um den Status des Elektroboards dar zustellen.
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.10 FREQ- 204 Web-Schnittstelle (REST API) Set Power

Referenz	Kapitel 5.3
Beschreibung	Das Client-Programm muss eine Web-Schnittstelle zur Verfügung stellen um den Power der Mikrowellen ein oder auszuschalten.
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.11 FREQ-210 Elektrische Schaltung

Referenz	Kapitel 5.3
Beschreibung	Eine elektrische Schaltung muss umgesetzt werden um vom Elektroboard die Mikrowelle Strom oder keinen Strom zu geben.
Techn. Risiko	Niedrig
Business Value	Niedrig

6.1.12 FREQ-212 Ansteuerung Elektrische Schaltung

Referenz	Kapitel 5.3
Beschreibung	Das Elektroboard muss die Elektronische Schaltung ansteuern können.
Techn. Risiko	Niedrig
Business Value	Niedrig

6.2 Nicht funktionelle Anforderungen**6.2.1 NREQ-15 Wartbarkeit**

UC-Referenz	Kapitel 5.3
Beschreibung	Die Wartbarkeit des Systems kann sichergestellt werden.
Techn. Risiko	Sehr niedrig
Business Value	Sehr niedrig

6.2.2 NREQ-20 Zeitnahe Kommunikation zwischen App und Elektronik Board

UC-Referenz	Kapitel 5.3
Beschreibung	Das App soll möglichst in real Time das Elektronik Board ansteuern können.
Techn. Risiko	Hoch
Business Value	Sehr hoch

6.2.3 NREQ-30 User über Status informieren

UC-Referenz	Kapitel 5.3
Beschreibung	Der User kann über den aktuellen Status visuel informiert werden.
Techn. Risiko	Hoch
Business Value	Sehr hoch

7 Design der Software

In diesem Kapitel wird ein System entworfen. Das System muss den Anforderungen, welche in Kapitel 6 definiert wurden, entsprechen.

7.1 Einleitung

Die Software Architektur bildet das Gerüst für die zu entwickelnde Applikation. Sie definiert welche Technologien in welchen Komponenten eingesetzt werden und wie über die Schnittstellen kommuniziert wird.

7.2 Design Patterns

7.2.1 Target-Action Pattern

Im Target-Action Design Pattern enthalten die Objekte die wichtigen Informationen um eine Message an ein anderes Objekt zu senden sofern ein Event ausgelöst wird.

Erklärt an einem einfachen Beispiel: Rezept-Verwaltung. Wir möchten ein neues Rezept erfassen. Wir haben einen Rezepteditier -View mit dem Rezept-Formular. In einer anderen View, der Navigation-View, sind Abbruch und Speicher Button enthalten. Der Speicher-Button muss die Rezepteditier-View darüber informieren, wenn er gedrückt wurde. So kann die Rezepteditier-View veranlassen, dass die Informationen gespeichert werden. Um dies umzusetzen braucht der Button 2 Dinge: Ziel (target) und die Aktion-Methode.

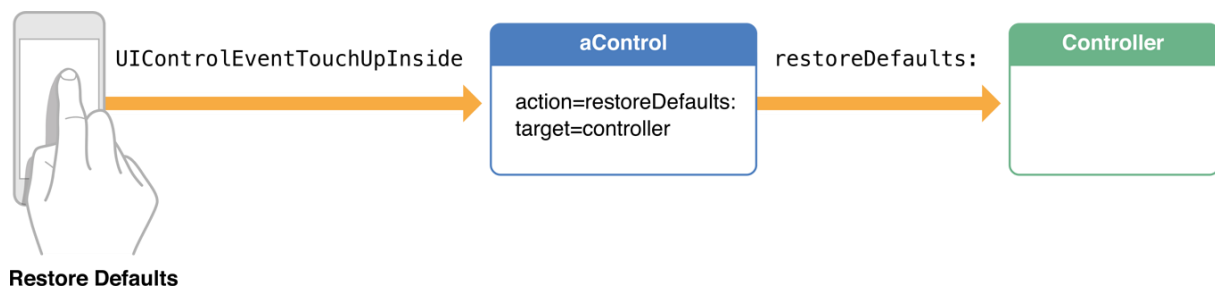


Abbildung 9 - Target-Action Pattern *Quelle <https://developer.apple.com>*

7.2.2 MVC, Model View Controller

Für die Umsetzung des Webservices soll das MVC-Pattern verwendet werden. Dadurch wird der Code in 3 Teile unterteilt:

- **Model**
Im Model wird die Businesslogik abgebildet. Das Model enthält Daten welche zur Darstellung benötigt werden.
- **View**
In der View werden die Daten dargestellt. Die View nimmt Benutzerinteraktionen entgegen und leitet sie an den Controller weiter.
- **Controller**
Der Controller ist die Steuereinheit. Er dient als Schnittstelle zwischen View und Model. Der Controller führt Benutzeraktionen aus und validiert diese, entscheidet welches View angezeigt werden soll oder ob ein anderer Controller angezeigt werden soll.

7.3 Elektroboard (Mikrocontroller-Board/Interfaceplatine)

Die zu erstellende elektronische Schaltung gilt es mittels eines Elektroboards zu steuern. Generell gibt es zwei verschiedene Arten die den Anforderung entsprechen.

7.3.1 Interfaceplatine (Velleman)

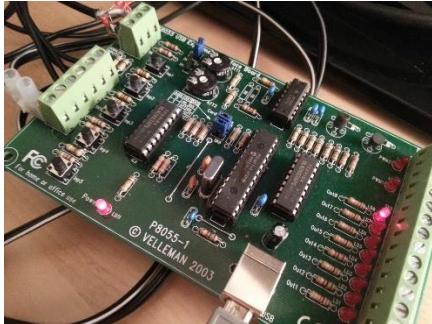


Abbildung 10 - Velleman Interfaceplatine

Die Interfaceplatine hat eine Vielzahl von digitalen und analogen Inputs und Outputs. An diesen können Sensoren, Schalter, Motoren, Signale und ähnliches angeschlossen werden. Die Interfaceplatine kann über elektronische Schaltungen erweitert werden. Die Interfaceplatine wird per USB² mit dem PC verbunden. Ein Treiber stellt die Kommunikation über den COM-Port zu Verfügung. Über diesen Treiber kann die Vielzahl von analogen und digitalen Inputs verarbeitet und die Outputs gesetzt werden. Das DLL kann also in die gewünschte³ Programmiersprache implementiert werden. Danach kann einfach aus einem Programm welches auf dem PC läuft die Befehle an die Interfaceplatine gesendet oder die Inputs empfangen werden. Für die Programmierung des Client-Programms muss keine zusätzliche Programmiersprache erlernt werden. Da die Logik auf dem PC ausgeführt wird, kann eine Interfaceplatine nur mittels PC betrieben werden.. Weiter gilt der Vorteil zu beachten, dass der Student bereits mit der Velleman P8055 Interfaceplatine gearbeitet hat.

7.3.2 Mikrocontroller Board (Arduino)



Abbildung 11 - Arduino Uno

Das Mikrocontroller-Board unterscheidet sich zur Interfaceplatine darin, dass er die analogen und digitalen Inputs und Outputs über seinen Chip (der Mikrocontroller-Chip) direkt auf dem Board verarbeitet. Die mögliche USB-Verbindung ist nur für die Programmübertragung oder für Inputs seitens PC zuständig. Eine spezielle Programmiersprache oder Erweiterungen sind meist zu erlernen. Der Student hat bis anhin noch nicht mit einem Mikrocontroller Board gearbeitet.

² Früher wurde RS-232 als serielle Schnittstelle zwischen PC und Interfaceplatine verwendet. Andere Technologien wären auch möglich.

³ Eine Programmiersprache welche DLL's implementieren kann ist nötig

7.3.3 Entscheidung

Das vorhandene Know-How und die bekannte Programmiersprache sprechen für die Interfaceplatine P8055 von Velleman. Beide Vorteile sind im Bereich Know-How anzusiedeln und bergen nur indirekten Vorteil für die Lösung PerfectPopcorn. Der Vorsprung dieses Know-How darf nicht zu hoch eingestuft werden, da bis anhin nur ein Projekt des Studenten mit der Interfaceplatine umgesetzt wurde. Das für die Lösung zusätzlich immer einen PC oder MiniPC betrieben werden müsste, ist ein entscheidender Nachteil der Interfaceplatine. Auch wenn es sich bei der Arbeit nur um einen Prototyp handelt, gilt es eine realistische Akzeptanz des Prototypes einzuhalten. Die Akzeptanz würde durch den Einsatz eines platzsparenden Arduino Boards klar gesteigert. Deshalb wird sich der Student das nötige Wissen aneignen müssen um PerfectPopcorn mit einem Arduino Board umzusetzen.

7.4 Mockups

7.4.1 Ansicht App

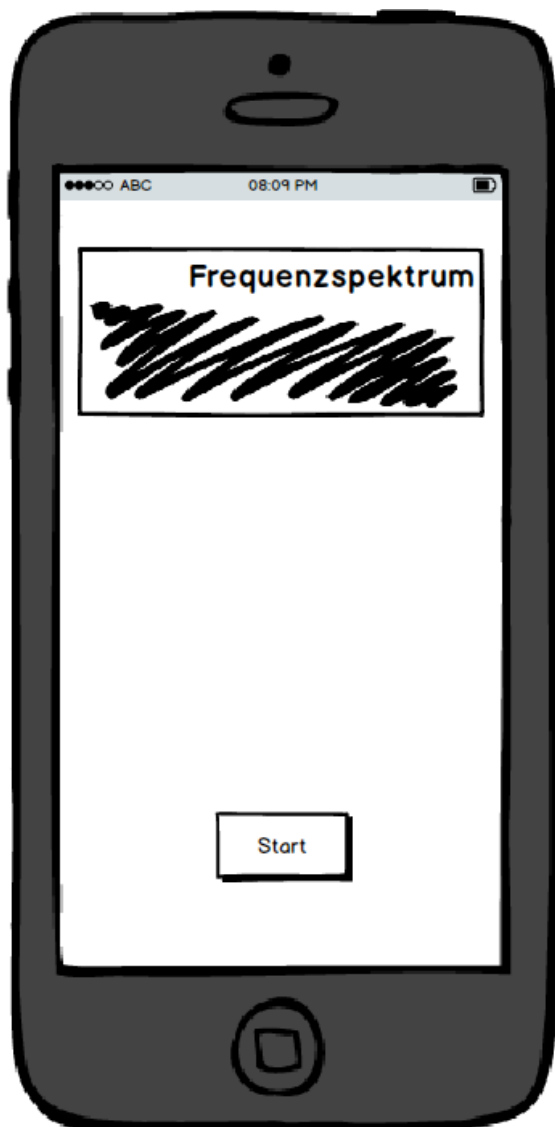


Abbildung 12 – Mockup der Android App

Die Android App soll übersichtlich alle Informationen darstellen.

Im oberen Bereich sollen Near Real Time die Ergebnisse der FFT (Fast Fourier Transformation) als Frequenzspektrum angezeigt werden. So können die Frequenzspektrum Abbildungen des Poppens und natürlich auch allen anderen Geräuschen mitverfolgt werden.

Die Info-Leiste im Mittelbereich informiert den User ob die Verbindung zum Mikrocontroller steht.

Über die Button-Leiste kann der User den Prozess starten und beenden können.

8 Implementation

8.1 Technologien

8.1.1 Adobe Air

(Adobe, 2015)

Mithilfe der *Adobe AIR*-Laufzeit können Entwickler den gleichen Code in native Anwendungen und Spiele sowohl für Windows und Mac OS-Desktop-Computer als auch iOS- und Android-Geräte einbinden und somit über eine Milliarde Desktopsysteme und mobile App Stores für über 500 Millionen Geräte erreichen.

8.1.2 ActionScript 3

(Adobe Learning, 2015)

ActionScript 3 wird umgangssprachlich *AS3* bezeichnet und ist die Programmiersprache für Runtime Environments *Adobe Air* und *Adobe Flash Player*. *ActionScript 3* wurde 2006 von Adobe eingeführt. Es unterscheidet sich komplett von den Versionen 1 und 2. *AS 3* ist eine objektorientierte Programmiersprache und enthält alle Grundprinzipien der Objektorientiertheit.

8.1.3 Arduino Programming Language

Die Arduino Programmiersprache wird auch „The Arduino Language“ genannt. Entwickelt wurde Sie basieren auf Wiring und wurde durch David Mellis entwickelt. Wiring und dadurch auch die Arduino Programmiersprache ähneln C/C++ stark. Für ein funktionsfähiges Programm sind mindestens folgende zwei Methoden zu definieren:

- `Setup()` Wird beim Programmstart einmalig aufgerufen
- `Loop()` Wird immer wieder durchlaufen bis das Arduino Board ausgeschaltet wird

8.2 Ansteuerung des Mikrofons

Um über ein Android App das Mikrofon ansteuern zu können, muss der spätere Anwender dazu die Berechtigung erlauben.

8.2.1 Android Manifest

Das Sicherheitsmodel von Android Applikation sieht vor, dass der Zugriff auf sicherheitsrelevante Dienste und Möglichkeiten oder persönliche Daten generell gesperrt ist. Muss die App auf diese Dienste zurückgreifen hinterlegt sie diese benötigten Funktionalitäten in einem Manifest. Bei der Installation der App wird der User über die geforderten sicherheitsrelevanten Funktionalitäten informiert und kann diese annehmen.

```
<manifest . . . >
    <permission android:name="android.permission.RECORD_AUDIO" . . . />
    <permission
        android:name=" android.permission-group.MICROPHONE" . . . />
</manifest>
```

Es können sowohl einzelne Permissions wie auch Group-Permissions gesetzt werden.

Da Mikrofone missbraucht werden könnten, fällt die Verwendung eines Mikrofons auch unter die sicherheitsrelevanten Funktionalitäten. Deshalb wurde für die PerfectPopcorn-App ein Manifest mit der entsprechenden Berechtigungs-Anforderung festgelegt.

8.2.2 Ansteuerung Mikrofon im Programmcode

Die eigentliche Ansteuerung eines Mikrofons über die App ist mit Adobe Air sehr einfach. Adobe Air stellt dafür eine eigene „Microphone“-Klasse zur Verfügung. Durch Aufruf der Methode Get, erhält man Zugriff auf das Mikrofon und kann Audio auswerten oder aufnehmen.

```
private var m_mic:Microphone;

. . .

m_mic = Microphone.getMicrophone();
```

8.3 Fast Fourier Transformation

Für die Implementation der Fast Fourier Transformation, wurde eine bestehende Berechnungsbibliothek verwendet. Diese Bibliothek wurde mit einer ehemaligen Flash Actionscript Bibliothek gefunden. Gerry Beauregard hat 2010 auf seinem Blog die Entwicklung dokumentiert. Er entwickelte die Fast Fourier Transformation für den Flash Player 10. Mit Hilfe dieser Bibliothek ist es möglich im Browser mit Hilfe des Flash Plugins eine Fast Fourier Transformation durchzuführen. Die Klassen-Bibliothek hat Gerry Beauregard veröffentlicht und optimiert.

Für PerfectPopcorn galt es die Scripts anzupassen und in Adobe Air App zu transportieren und integrieren. Nun kann in der PerfectPopcorn App die Audio Welt mittels Mikrofon erfasst werden und über die angepasste FFT von Gerry Beauregard visualisiert werden. PerfectPopcorn ist dadurch ein einfacher **Spektrumanalysator**.

8.4 Auswertung des Frequenzspektrums

Die entstandenen Frequenzspektrums aus den Popcorn-Pop Tönen gilt es nun zu analysieren und möglichst einfach von den Umgebungsgeräuschen abzugrenzen. Folgend sind 3 Screenshots von 3 unterschiedlichen Pops abgebildet.

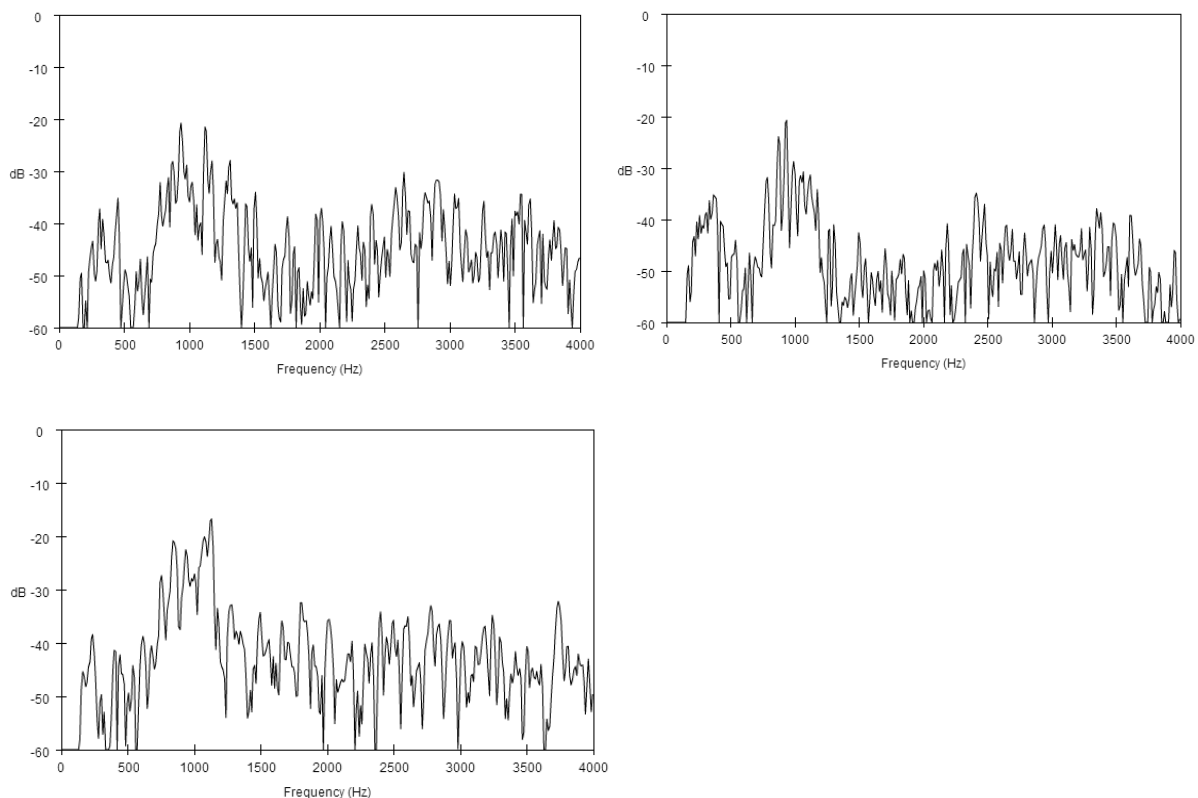


Abbildung 13 - 3 Screenshots der Popcorn Pop Töne

8.4.1 Feststellung

Das Poppen erhöht die Amplitude fast durchgehend über die alle gemessenen Frequenzen. Im Bereich von 1000 Herz ist es besonders hoch. Diese einfachen Erkenntnisse gilt es nun auszunutzen.

8.5 Softwarebausteine

8.6 Elektrischer Schaltplan

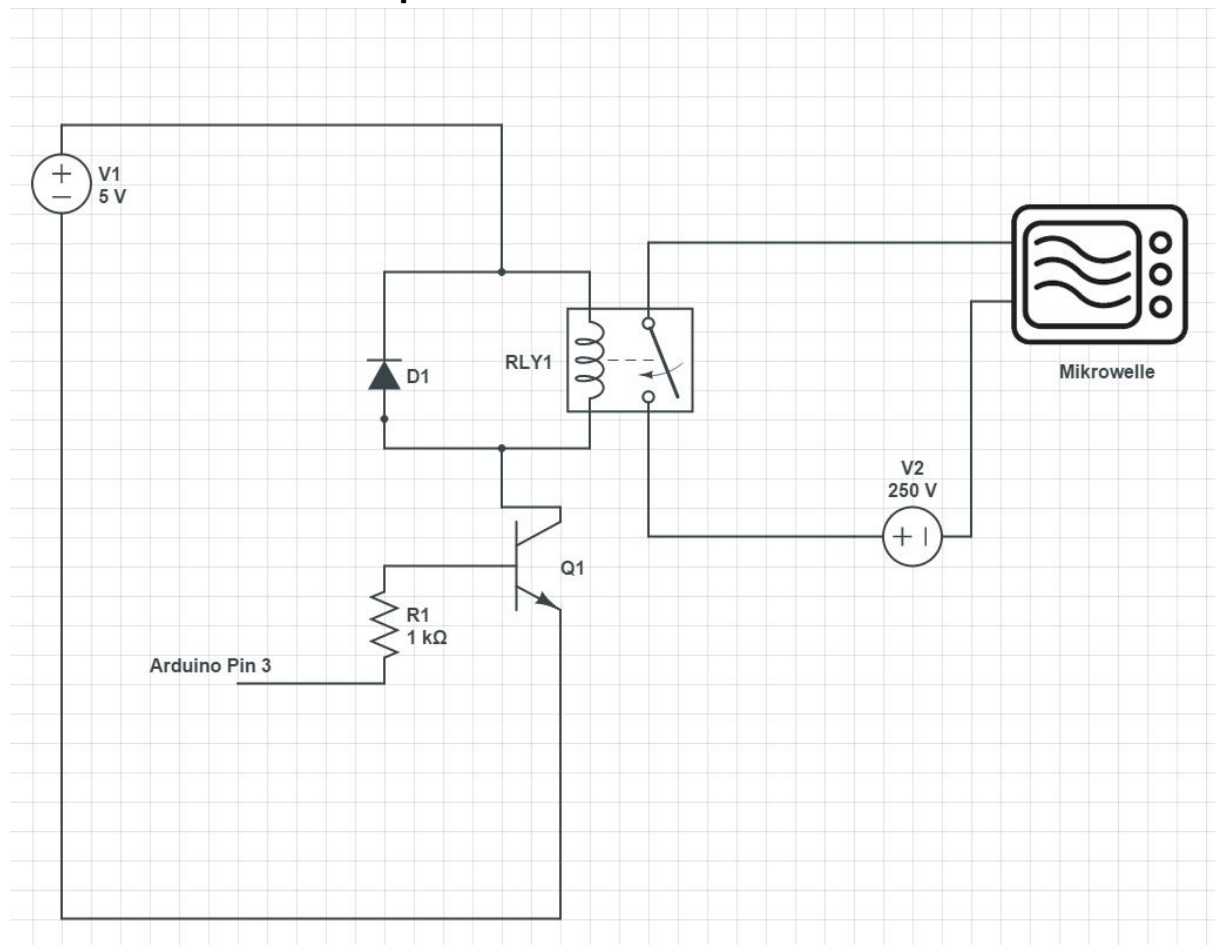


Abbildung 14 - Elektrischer Schaltplan

Ziel ist es mittels Arduino Output die Mikrowelle ein bzw. auszuschalten. Aus der Elektrotechnik sind 2 Komponenten bekannt mit welchen elektrische Stromkreise geschaltet werden können: Transistoren und Relays.

Für unsere Schaltung ist es zwingend nötig, die beiden Stromkreise (Steuerstromkreis und Laststromkreis) komplett zu trennen. Da wir den Laststrom mit 250V und ca. 750 Watt betreiben. Der Laststrom könnte einem Menschen das Leben kosten. Deshalb ist Vorsicht geboten. Natürlich könnte der Laststrom den gesamten Mikrocontroller zerstören würde er falsch verbunden. Die komplette Trennung ermöglicht ein Relais. Das Relais ist ein mechanischer Schalter. Das Relais arbeitet nach dem Prinzip des Elektro-Magneten: Fließt der Steuerstrom entsteht eine elektromagnetische Spannung durch eine Spule. Durch das schließt sich der Anker und der Laststromkreis ist geschlossen.

Das Arduino kann über den Pin3 bis zu 5V mit ca. 30-50mA output geben. Auch wenn dieser Output theoretisch als Steuerstrom für das Relais reichen würde, hat es nicht immer funktioniert. Dies liegt daran, dass das Arduino selbst auch Ampere und Strom für die Signalverarbeitung braucht. Um diesen Fehler vorzubeugen wird zuerst ein Transistor geschaltet der die 5V zuverlässig dem Relais weitergibt.

8.7 App-Screen

8.8 Fotos der Testreihen

9 Probleme & Lösungen

9.1 Arduino WifiShield

Das Arduino Uno Board funktioniert wie in Kapitel 7.3.2 beschrieben ohne am PC angeschlossen zu sein. Trotzdem sollte das AndroidApp mit dem Arduino-Board kommunizieren können. Dazu wird das Arduino Uno Board mit einem WifiShield erweitert. Das WifiShield ermöglicht es den Mikrocontroller des Arduino Uno Boards ins Wifi-Netz zu stellen und innerhalb dieses über eine IP-Adresse anzusprechen.

Das Ansprechen des Arduinobords von aussen, war nach dem Bezug des WifiShields nicht möglich. Nach Recherchen konnte herausgefunden werden, dass die WifiShields mit Firmware 1.0 ausgeliefert werden. Um korrekt darauf zugreifen zu können, war ein Firmware-Update von Nöten. Trotz Installation von Spezialsoftware konnte das Update nicht durchgeführt werden. In einem Mikrocontroller-Forum wurde darauf hingewiesen, dass dazu ein PC mit max. Windows 7 Betriebssystem nötig ist. Dieser Tipp verhalf zu einem erfolgreichen Update

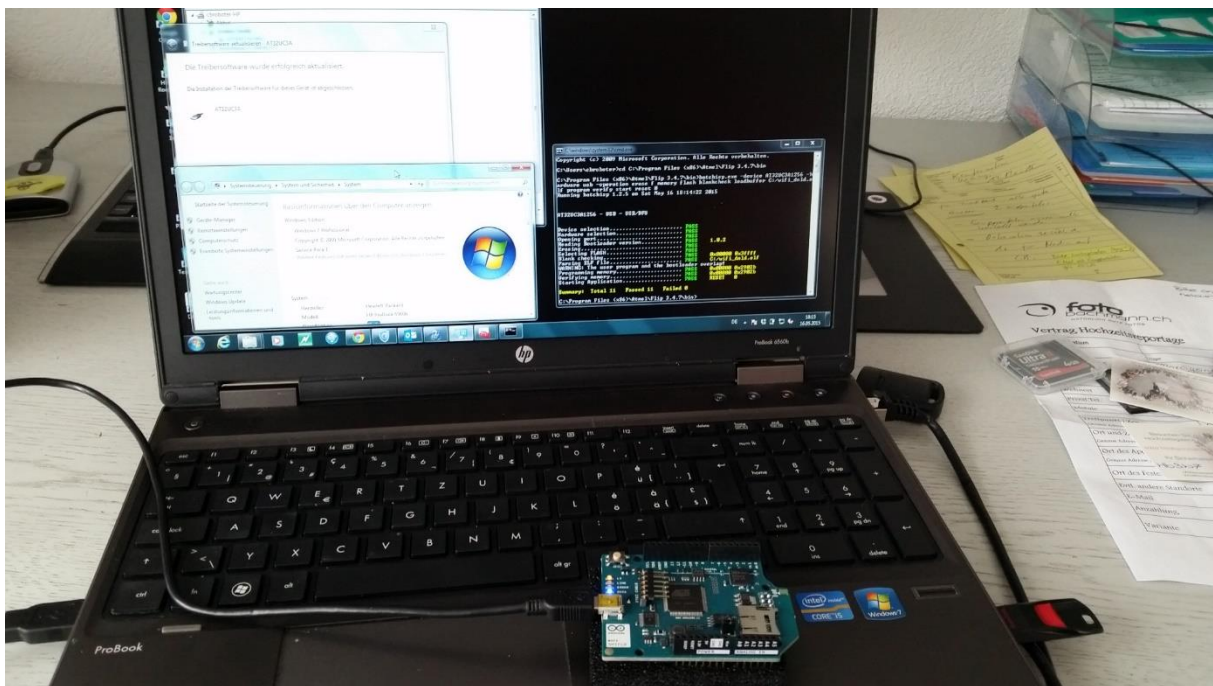


Abbildung 15 - Aktualisierung mittels WifiShield

10 Fazit und Schlusswort

Glossar

Code First

Die zielgerichtete Entwicklung für eine Datenbank, die noch nicht vorhanden ist und mittels Code First erstellt wird.

Microsoft MSDN

Entity Framework

Entity Framework ist die empfohlene Datenzugriffstechnologie von Microsoft für neue Anwendungen.

Microsoft MSDN

Github

Ein Cloud basierter Sourcecode Verwaltungs Dienst der Git einsetzt.

<https://github.com/>

Liedteile

Ein Liedteile prägen den Aufbau eines Liedes. Beispiele dafür sind Strophe, Refrain, Bridge, Liedteil 1, Liedteil 2,... Liedteile können mehrfach in einem Lied vorkommen oder mehrfach wiederholt werden.

WEB-API

API steht für WEB Application Programming Interface.

http://en.wikipedia.org/wiki/Web_API

Abbildungsverzeichnis

Abbildung 1 Verbranntes Mikrowellen-Popcorn von Christian Bachmann	2
Abbildung 2 - Projektplan.....	4
Abbildung 3 - Anleitung Zubereitung Mikrowellen Popcorn <i>Quelle chefkoch.de</i>	7
Abbildung 4 - Ablaufdiagramm Zubereitung Mikrowellen Popcorn.....	8
Abbildung 5 Zu optimierenden Schritte	9
Abbildung 6 - Spektrumanalysator <i>Quelle home.zhaw.ch</i>	11
Abbildung 7 - Frequenzspektrum eines FFT-Analysators <i>Quelle www.tomshardware.de</i>	11
Abbildung 8 - Ablauf FFT-Analyse / Vereinfachtes Blockschaltbild <i>Quelle Wikipedia</i>	12
Abbildung 9 - Target-Action Pattern <i>Quelle https://developer.apple.com</i>	16
Abbildung 10 - Velleman Interfaceplatine.....	17
Abbildung 11 - Arduino Uno.....	17
Abbildung 12 – Mokup der Android App.....	19
Abbildung 13 - 3 Screenshots der Popcorn Pop Töne.....	22
Abbildung 14 - Elektrischer Schaltplan	24
Abbildung 15 - Aktualisierung mittels WifiShield	26

Tabellenverzeichnis

Tabelle 1- Soll-Ist Vergleich	5
Tabelle 2 - Beteiligte Personen	6

Literaturverzeichnis

Adobe. (25. 02 2015). *Adobe.com*. Von www.adobe.com abgerufen

Adobe Learning. (25. 02 2015). *Adobe AS3 Learning*. Von Adobe.com:
<http://www.adobe.com/devnet/actionscript/learning.html> abgerufen

Althof, W. (1999). Fehlerwelten. Springer Fachmedien Wiesbaden GmbH.

<http://de.wikipedia.org/wiki/Spektrumanalysator>. (19. 04 2015). Von
<http://de.wikipedia.org/wiki/Spektrumanalysator> abgerufen

Microsoft. (30. 04 2015). *Einführung in die Programmiersprache C# und in .NET Framework*. Von
MSDN: <https://msdn.microsoft.com/de-de/library/z1zx9t92.aspx> abgerufen

Rupp, K. P. (2011). *Basiswissen Requirements Engineering*. dpunkt.verlag.

Wikipedia. (30. 04 2015). *C-Sharp*. Von Wikipedia: <http://de.wikipedia.org/wiki/C-Sharp> abgerufen

Bestätigung über die Selbständigkeit

Hiermit versichere ich, die vorliegende Semesterarbeit eigenständig und ausschliesslich unter Verwendung der angegebenen Hilfsmittel angefertigt zu haben.

Alle öffentlichen Quellen sind als solche kenntlich gemacht. Die vorliegende Arbeit ist in dieser oder anderer Form zuvor nicht als Semesterarbeit zur Begutachtung vorgelegt worden.

Aadorf 30.05.2015

Christian Bachmann