**Pflichtenheft**

Fachlicher Teil

Projekt 4 – Team 1

Windisch, 26. Februar 2020

Hochschule: *FHNW-Hochschule für Technik*

Studiengang: *Elektro- und Informationstechnik*

Team 1: *Silvan Burkard*

*Roman Fischer*

*Reto Gärtner*

*Julian Jeisy*

*Jenny Meier*

Auftraggeber: *Prof.* *Hans Gysin*

Betreuungspersonen: *Pascal Buchschacher  
 Dr. Anita Gertiser  
 Marie-Thérèse Rudolf von Rohr  
 Prof. Dr. Pascal Schleuniger  
 Albert Zihlmann*

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc33606698)

[2 Projektdefinition 3](#_Toc33606699)

[2.1 Ausgangslage 3](#_Toc33606700)

[2.2 Projektziele 3](#_Toc33606701)

[2.3 Nichtziele 3](#_Toc33606702)

[2.4 Lieferobjekte 3](#_Toc33606703)

[3 Lösungskonzept 4](#_Toc33606704)

[3.1 Blockschaltbild und Produkteeigenschaften 4](#_Toc33606705)

[3.2 Hardware 4](#_Toc33606706)

[3.3 Software 4](#_Toc33606707)

[3.4 Bedienung 4](#_Toc33606708)

[4 Testkonzept 4](#_Toc33606709)

[5 Projektvereinbarung 5](#_Toc33606710)

[6 Echtheitserklärung 6](#_Toc33606711)

[7 Literaturverzeichnis 7](#_Toc33606712)

# Einleitung

# Projektdefinition

## Ausgangslage

Um an Ausstellungen und Informationsveranstaltungen nicht unterzugehen, benötigt es eine herausstechende Attraktion. Hierfür eignen sich vor allem spektakulär leuchtende und interaktive Demogeräte. Im Rahmen des Projekt 4 des Studiengangs Elektro- und Informationstechnik, soll ein solches Demogerät entwickelt und realisiert werden. Das Ziel ist es mithilfe des Persistence of Vision (POV) Prinzips eine gut lesbare Schrift oder ein schönes Bild darzustellen. Als Anzeigeelement dienen LED-Bahnen, welche durch eine genügend schnelle Drehung, den gewünschten POV Effekt erzielen. Für die Interaktion wird eine Drahtlose Kommunikationsschnittstelle benötigt, welche via Smartphone, Tablet oder Laptop bedient werden können soll. In diesem Pflichtenheft wird auf die Anforderungen und umzusetzenden Funktonen des Demogerätes aufgezeigt.

## Projektziele

Damit klare Rahmenbedingungen herrschen, müssen die genauen Ziele definiert werden. In der nachfolgenden Tabelle sind die genauen Anforderungen an das Produkt ersichtlich.

|  |  |  |
| --- | --- | --- |
| Zielkriterium | Mindestanforderung | Wunschanforderung |
| Das POV kann ein Bild erkennbar darstellen. | Auf dem Halbkreis befindet sich mindestens ein LED pro cm. | Auf dem Halbkreis befindet sich drei LEDs pro cm. |
| Das POV erzeugt Bilder mit einer für den Menschen genügenden Bildwiederholfrequenz. | Die Wiederholfrequenz beträgt mindestens 15 Bilder pro Sekunde. | Die Wiederholfrequenz beträgt 30 Bilder pro Sekunde. |
| Das POV soll so konstruiert sein, dass es gut transportierbar ist. | Die Dimensionen sollen kleiner sein als 250x250x400mm (LxBxH) | Das Gerät kann zum Transport auseinandergenommen werden. |
| Das Gerät wird mittels eines aufladbaren Akkus betrieben. | Der Akku kann zum Aufladen ausgebaut werden. | Das POV verfügt über einen Netzanschluss zum Aufladen. |
| Die vom POV dargestellten Bilder sind farbig. | Es können sieben verschiedene Farben dargestellt werden (Rot, Grün, Blau und Mischfarben von diesen). | Jede Farbe kann zwischen 0 und 255 variiert werden. |
| Es können statische (unbewegte) Bilder dargestellt werden. | Unter Annahme einer konstanten Drehzahl wird der Programmablauf entsprechend getacktet. | Es ist eine Einrichtung zur Messung der Drehzahl vorhanden, welche den Bildaufbau synchronisiert. |
| Es kann über Schall mit dem POV interagiert werden. | Das Gerät kann über Klatschgeräusche gesteuert werden. | Das Gerät nimmt Stimmen wahr und kann den Schallpegel anzeigen (Oszilloskop). |
| Mittels einer Android App kann das POV kabellos gesteuert werden. | Mit dem Drücken bestimmter Tasten ändert sich das angezeigte Bild (z.B. Spiel). | Es kann auf der App ein Bild gezeichnet werden, welches dann vom POV angezeigt wird. |

## Nichtziele

## Lieferobjekte

Die Lieferobjekte entsprechen den Meilensteinen, welche im Drehbuch ersichtlich sind.

|  |  |
| --- | --- |
| **Lieferobjekt** | **Datum** |
| Pflichtenheft organisatorischer Teil | 10.03.2020 |
| Pflichtenheft technischer Teil 1. Version | 10.03.2020 |
| Pflichtenheft technischer Teil def. Version | 24.03.2020 |
| Statusbericht 1 | 29.04.2020 |
| Zwischenpräsentation | 07.04.2020 |
| Disposition und Einleitung Fachbericht | 28.04.2020 |
| Statusbericht 2 | 03.05.2020 |
| Statusbericht 3 | 17.05.2020 |
| Schlusspräsentation | 09.06.2020 |
| Fachbericht | 09.06.2020 |
| Statusbericht 4 | 09.06.2020 |

Tabelle Lieferobjekte

# Lösungskonzept

Dieser Abschnitt beinhaltet das Konzept für die LED POV Kugel.

## Blockschaltbild und Produkteeigenschaften

Die Abbildung 1 zeigt das Blockschaltbild für die Hardware der POV Kugel. Im nachfolgenden Kapitel 3.2 Hardware wird auf die einzelnen Blöcke genauer eingegangen.



Abbildung Blockschaltbild der POV Kugel

## Hardware

In diesem Kapitel wird auf die Hardware eingegangen, welche für die Realisierung der POV Kugel benötigt wird.

### Mikrocontroller

Der Mikrocontroller (Mc) ist das zentrale Element der Hardware. Er verarbeitet nicht nur die Daten des Mikrophons und des Bluetoothmoduls, sondern steuert auch die Schieberegister für die LEDs. Um diesen Anforderungen gerecht zu werden, muss der Mc genügend schnell sein und einen genug grossen RAM-Speicher besitzen. Für die Realisierung wird deshalb ein STM32 Mikrocontroller der Firma STMicroelectronics verwendet.

### UART

Für die Programmierung des Mc wird eine UART-Schnittstelle benötigt. Hierfür wird der FT2232D IC von FTDI Chip verwendet. Da es sich hierbei um einen Dual USB to UART Interface handelt, kann der Baustein sowohl für den Mc, wie auch für das Bluetooth-/Wlanmodul verwendet werden.

### Schieberegister

Die Daten des Mc für die LED-Bahn werden mithilfe von rund 25 Schieberegistern (HEF4894B) der Firma NXP verarbeitet. Pro Schieberegister können 4 LEDs angesteuert werden. Mithilfe des STR Pins werden zuerst alle Daten in das Schieberegister geschoben, bis dieses voll ist. Anschliessend werden die Daten gleichzeitig an die LEDs ausgegeben. Dadurch werden allfällige Verzögerungen, welche den POV-Effekt stören könnten, verkürzt. Die Kommunikation zwischen dem Mikrocontroller und den Schieberegister wird per SPI-Schnittstelle realisiert.

### LED-Bahn

Die POV Kugel besteht aus einer LED-Bahn mit rund 100 LEDs. Es werden 90° abgewinkelte RGB LEDs (MSDL0601RGB) der Firma Rohm verwendet, mitwelchen man 8 verschiedene Farben darstellen kann.

### Sensorik

Um die Position der LED-Bahn bestimmen zu können, wird ein Hall-Sensor verwendet. Der Hall-Sensor wird per I/O-Pin mit dem Mc verbunden.

### Interaktion durch Mikrophon

Eine Möglichkeit mit dem POV interagieren zu können erfolgt über den Schallpegel. Dazu ist am «Nordpol» der Kugel ein Elektret-Mikrophon montiert. Diese Art von Mikrophonen wird oft in kommerzieller Elektronik aufgrund ihrer kleinen Dimensionen, des guten Frequenzverhaltens und des niedrigen Preises eingesetzt. Sie bestehen aus einem Kondensator mit konstanter Ladung und einem JFET, welcher als Impedanzwandler dient. [1]

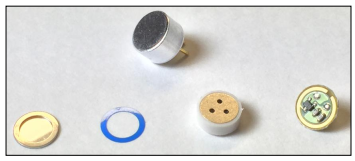


Abbildung : Aufbau eines Elektret-Mikrophons.

Das Mikrophonsignal wird mit einem diskret aufgebauten Verstärker auf eine Spannung gebracht, welche vom Mikrokontroller weiterverarbeiten werden kann (max. 3.3V). Elektret-Mikrophone haben meistens eine Empfindlichkeit zwischen -46dBV und -35dBV, umgerechnet entspricht dies einer Empfindlichkeit von 5…18mV/Pa [2]. Der Schalldruck einer Unterhaltung aus 1m Distanz liegt bei etwa 40…50dB (Referenzpegel: 20μPa) was im Mittel rund 4mPa entspricht [3]. Daraus ergibt sich ein Ausgangssignal der Amplitude 20…72μV. Für diesen Fall wird also eine Verstärkung von ungefähr 90…100dB benötigt. Da der Ausgang eines Elektret-Mikrophons sehr hochohmig ist, wird ein Transimpedanzverstärker verwendet. Das nachfolgende Bild zeigt, wie der grundsätzliche Aufbau des Verstärkers aussieht.

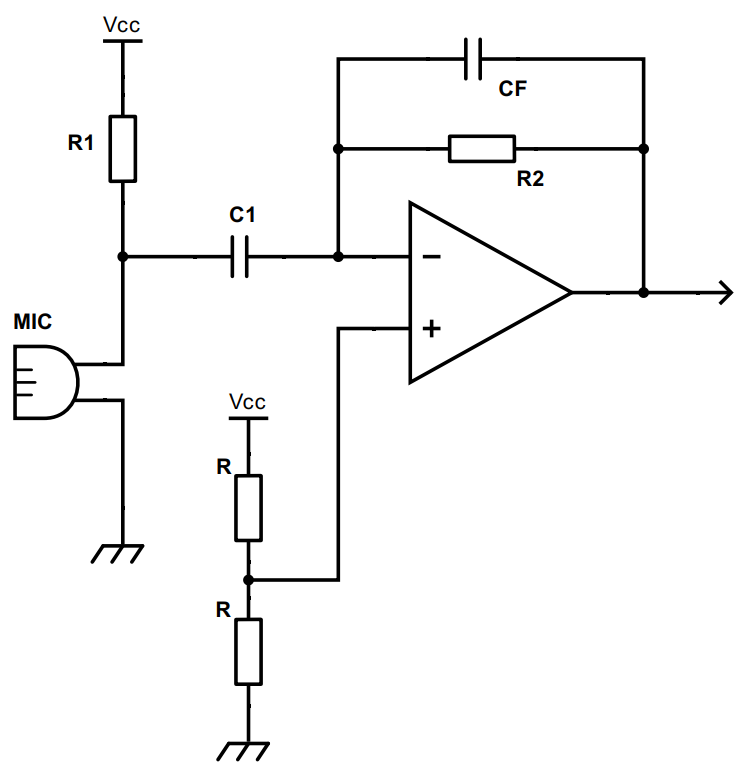


Abbildung : Aufbau des Mikrophonverstärkers. R1 speist das eingebaute FET des Mikrophons. C1 dient zum Entkoppeln des Signals und R2 bestimmt die Verstärkung. CF verhindert, dass der Verstärker zu schwingen beginnt.

Damit der Mikrocontroller das Signal auswerten kann, muss dieses zudem nach der Verstärkung mit einem Spannungsteiler angehoben werden, dass es im Bereich 0…3.3V schwingt. Um Aliasing-Effekte zu verhindern, muss ein Tiefpassfilter mit einer Grenzfrequenz kleiner als die halbe Sampling-Frequenz des ADCs integriert werden. Jedoch sollte diese Grenzfrequenz noch im hörbaren Spektrum liegen (>20kHz).

Durch die Konstruktion ist das Mikrophon mechanisch mit der Welle des Motors verbunden. Um zu verhindern, dass das Mikrophon zu viele Störgeräusche aufnimmt, können diese mit einem Hochpassfilter unterdrückt werden. Das kann direkt im Mikrophonverstärker durch R1 und C1 bewerkstelligt werden. Die störenden Frequenzen liegen bei den vielfachen der Motordrehzahl [4]. Das POV rotiert mit einer Drehzahl von 10…30Hz. Ein Hochpass mit einer Grenzfrequenz von einigen hundert Hertz könnte der grösste Teil der Störungen eliminieren. Dies würde gleichzeitig auch das meiste einer menschlichen Stimme herausfiltern, jedoch kann das Gerät immer noch Klatschgeräusche aufnehmen.

### Bluetooth-Modul

Für die serielle Datenübertragung zwischen der Android-App und des Mc, wird ein Bluetooth-Modul eingesetzt. Das ESP31-Wroom-32 Modul der Firma Espressif Systems besitzt nebst eines Bluetooth-Moduls zusätzlich noch ein Wifi-Modul, welches bei bedarf zusätzlich verwendet werden könnte. Damit wäre eine Bedienung per Webserver realisierbar.

### Mechanik

Die Mechanik soll aus drei Elementen bestehen:  
- Die runde Leiterplatte, auf welchen sich die LEDs befindet und sich um die Z-Achse dreht.  
- Der Standfuss, in welchem der Motor, Akku und ein Gewicht integriert ist.  
- Eine Befestigungsstange, an welcher sich der Magnet für den Hall-Sensor befindet.

## Software

In diesem Kapitel wird die Softwarestruktur des POVs behandelt. Dies beinhaltet die Programmierung des Controllers inklusive der Peripherie (Schieberegister mit SPI, Bluetooth, ADC, UART) und die Android App als Interaktionselement.

### Mikrokontroller

### Android App

Für die Entwicklung des Android Apps wird die offizielle IDE Android Studio verwendet. Apps können entweder mit Java, Kotlin oder C++ geschrieben werden. Da alle Teammitglieder Kenntnisse in Java haben, wird diese Sprache verwendet. Die grafische Oberfläche und das Layout der Apps werden mit XML-Dateien erstellt. In der folgenden Grafik ist das Grundgerüst einer App mit zwei Menus ersichtlich.

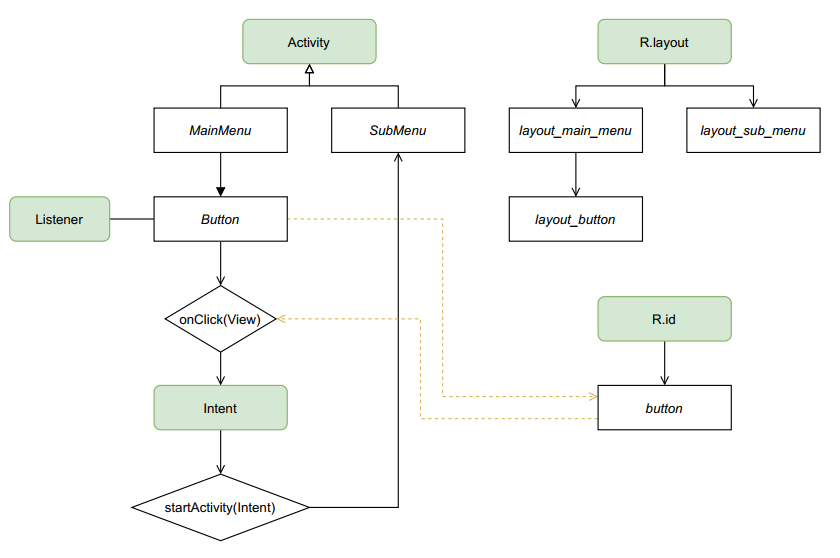


Abbildung : Prinzipschema einer App mit zwei Menüs und einem Knopf.

Jedes «Fenster» innerhalb einer App ist eine separate Klasse und wird als *Activity* bezeichnet. Innerhalb dieser Klasse wird u.a. definiert, wie auf Benutzereingaben reagiert wird. Einer *Activity* ist zudem eine XML-Datei zugewiesen, in welcher das Layout (Anordnung der Knöpfe, Farben, Animationen, etc.) festgelegt ist. Damit innerhalb der Klasse auf die Bedienelemente zugegriffen werden kann, gibt es die Klasse *R*, in welcher für jedes Element eine statische Identifikationsnummer abgelegt ist. Wie üblich in Java, müssen alle Bedienelemente mit einem entsprechenden *Listener* verknüpft sein, welcher auf Eingaben reagieren kann. Wenn z.B. in einem Menü ein Knopf gedrückt wird, kann mit *R.id.button* der Auslöser dieses Events bestimmt werden. Weitere Activities (z.B. das *SubMenu*) werden anschliessend mithilfe eines *Intents* gestartet.

## Bedienung

### Interaktion durch Android App

Das Bild unten zeigt eine Möglichkeit wie die App strukturiert werden kann. Es gibt ein Hauptmenü, dass man beim Starten der App sieht und ein Untermenü für die Bluetooth-Einstellungen. Weiter hat es eine Zeichenfläche, auf welcher mit dem Finger gezeichnet werden kann. Das Bild wird dann über übertragen und angezeigt. Weiter könnte ein «Keypad» eingebaut werden, um ein Spiel zu steuern.

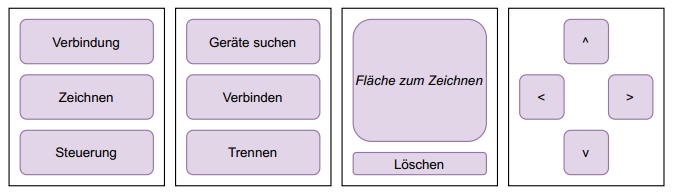


Abbildung 5: Mögliches Layout der App.

# Testkonzept

# Projektvereinbarung

# Echtheitserklärung

# Literaturverzeichnis

[1] J. Gentex Electro-Acoustic Products, „Apply Electret Microphones to Voice-Input Designs “, [Online]. Available: https://www.yumpu.com/en/document/read/36565851/apply-electret-microphones-to-voice-input-designs-gentex- [Abrufdatum 06.03.2020].

[2] J. Lewis, „Understanding Microphone Sensitivity“, 05.2012 [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/understanding-microphone-sensitivity.html [Abrufdatum 06.03.2020].

[3] wikipedia.org, „Schalldruck“, [Online]. Available: https://de.wikipedia.org/wiki/Schalldruck [Abrufdatum 06.03.2020].

[4] H. Kang, „The Study of DC Motor Noise and Vibration “, SAE Transactions, 1995, S. 2461–2467.

# Abbildungsverzeichnis

[Abbildung 1 Blockschaltbild der POV Kugel 5](file:///C:\Users\jenny\Documents\GitHub\pro4e\Pflichtenheft\Fachlich\FPH_teil_software_mic.docx#_Toc34608078)

[Abbildung 2: Aufbau eines Elektret-Mikrophons. 6](file:///C:\Users\jenny\Documents\GitHub\pro4e\Pflichtenheft\Fachlich\FPH_teil_software_mic.docx#_Toc34608079)

[Abbildung 3: Aufbau des Mikrophonverstärkers. R1 speist das eingebaute FET des Mikrophons. C1 dient zum Entkoppeln des Signals und R2 bestimmt die Verstärkung. CF verhindert, dass der Verstärker zu schwingen beginnt. 6](file:///C:\Users\jenny\Documents\GitHub\pro4e\Pflichtenheft\Fachlich\FPH_teil_software_mic.docx#_Toc34608080)

[Abbildung 4: Prinzipschema einer App mit zwei Menüs und einem Knopf. 8](file:///C:\Users\jenny\Documents\GitHub\pro4e\Pflichtenheft\Fachlich\FPH_teil_software_mic.docx#_Toc34608081)

[Abbildung 5: Mögliches Layout der App. 9](file:///C:\Users\jenny\Documents\GitHub\pro4e\Pflichtenheft\Fachlich\FPH_teil_software_mic.docx#_Toc34608082)

# Tabellenverzeichnis

[Tabelle 1 Lieferobjekte 4](#_Toc34608098)