

Multi-Room Sound Adapter

Nico Lang, Philipp Immler

Februar 2025

1 Projekt

1.1 Projektteam

Nico Lang

Wirtschaftsingenieure/Betriebsinformatik
Grießau
6651 Häselgehr AT
Nico.Lang@hak-reutte.ac.at

Philipp Immler

Wirtschaftsingenieure/Betriebsinformatik
Hoheneggweg 21a
6682 Vils AT
Philipp.Immler@hak-reutte.ac.at

1.2 Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen Hilfsmittel als die angegebenen benützt habe. Die Stellen, die anderen Werken (gilt ebenso für Werke aus elektronischen Datenbanken oder aus dem Internet) wörtlich oder sinngemäß entnommen sind, habe ich unter Angabe der Quelle und Einhaltung der Regeln wissenschaftlichen Zitierens kenntlich gemacht. Diese Versicherung umfasst auch in der Arbeit verwendete bildliche Darstellungen, Tabellen, Skizzen und Zeichnungen. Für die Erstellung der Arbeit habe ich auch folgende Hilfsmittel generativer KI-Tools (ChatGPT 3.5) zu folgendem Zweck verwendet: Inspiration und allgemeine Information. Auch Übersetzer (DeepL) wurden zur Hilfe genommen. Die verwendeten Hilfsmittel wurden vollständig und wahrheitsgetreu inkl. Produktversion und Prompt ausgewiesen.

.....
Ort, Datum

.....
Unterschrift Schüler/in

.....
Unterschrift Schüler/in

1.3 Abstract Deutsch

1.4 Abstract English

1.5 Danksagung

Inhaltsverzeichnis

1	Projekt	2
1.1	Projektteam	2
1.2	Eidesstattliche Erklärung	3
1.3	Abstract Deutsch	4
1.4	Abstract English	4
1.5	Danksagung	4
2	Einleitung	5
2.1	Einleitung Hardware	5
2.2	Einleitung Software	5
3	Planung	5
3.1	Festlegung Funktionsweise	5
3.2	Auswahl Hardwarekomponenten	5
3.3	Auswahl Technologien	5
3.3.1	Protokolle	5
3.4	Auswahl Softwaretools	6
3.4.1	Einleitung	6
3.4.2	Softwaretools Microcontroller	6
3.4.3	Softwaretools Smartphoneapp	7
4	Entwicklung	7
4.1	Design Platine	7
4.2	Bestückung Platine	7
4.3	Entwicklung Software Adapter	7
4.4	Entwicklung Smartphone-App	8
4.5	Design Adaptergehäuse	8
4.6	Fertigung Adaptergehäuse	8
5	Testen und Fehlerbehebung	8
5.1	Testen des Gesamtsystems	8
5.2	auftretende Fehler beheben	8
5.3	Test auf Cybersecurity	8
5.4	Auftretende Sicherheitslücken schließen	8
6	Einzelnachweise	8
6.1	Literaturverzeichnis	8
6.2	Abbildungsverzeichnis	8
6.3	Anhang	8

2 Einleitung

Hier befindet sich die allgemeine Einleitung der Diplomarbeit.

2.1 Einleitung Hardware

Der Teil der Hardware für folgende Diplomarbeit beschäftigt sich damit, einen sinnvollen internen Aufbau des Geräts zu erzielen, die am besten geeigneten Hardware-Komponenten zu finden, das System bzw. die einzelnen Komponenten zusammenzubauen (Platine) und zu testen. Dieser Teil der Diplomarbeit wird von Nico Lang übernommen. Zudem beschäftigt sich dieser Teil mit dem optischen Design des Geräts (Gehäuse) und bestimmt die technischen Anforderungen (Schnittstellen), die der Adapter letztendlich haben soll. Bei der Planung soll darauf geachtet werden, möglichst viele Kosten einzusparen, ohne dabei die Faktoren der Sicherheit und Qualität zu vergessen.

2.2 Einleitung Software

Der Teil der Software für folgende Diplomarbeit beschäftigt sich damit, einerseits die Software des Adapters, andererseits die Software der Smartphoneapp zu entwickeln. Dieser Teil der Diplomarbeit wird von Philipp Immler übernommen. Die Software des Adapters wird mit der Programmiersprache C++ codiert. Die Software der Smartphoneapp wird mit JavaScript codiert. Um eine bestmögliche Leistung und Effizienz zu garantieren, werden bei der Programmierung zahlreiche Bibliotheken und Frameworks verwendet. Bei der Entwicklung der Software wird ein großes Augenmerk auf Sicherheit und Effizienz gelegt.

3 Planung

3.1 Festlegung Funktionsweise

3.2 Auswahl Hardwarekomponenten

3.3 Auswahl Technologien

3.3.1 Protokolle

In diesem Kapitel geht es um die Recherche und Auswahl von Protokollen, die für den Austausch von Daten verwendet werden.

HTTP

Das Hyper Text Transfer Protocol wird für den Datenaustausch zwischen Microcontroller und Smartphone verwendet. Dabei läuft auf dem Microcontroller ein Webserver an den das Smartphone HTTP-Anfragen sendet. Dabei wird eine REST-API angewendet. Folgende Routen sind dabei auf dem Webserver aufrufbar:

Route	Anfragen-Typ	Funktion
/getInfo	GET	Client bekommt Infos vom Microcontroller
/getAvailableNetworks	GET	Client bekommt eine Liste im JSON-Format, gefüllt mit SSID und RSSI (Stärke) von verfügbaren Netzwerken in der Nähe des Microcontrollers
/setWiFiCredentials	POST	Client sendet SSID und Passwort des gewünschten Netzwerks an den Microcontroller
/setStreamUrl	POST	Client sendet die URL des gewünschten Audio-Streams an den Microcontroller

I2S

Das Inter Integrated Sound Protocol wird verwendet, um Audiodaten vom Microcontroller an den PCM5102 Digital-Analog-Wandler zu senden. Dabei werden die digitalen Buffer, die der Microcontroller vom Audio-Stream erhält, mittels I2S an den Digital-Analog Wandler gesendet, welcher die digitalen Daten in analoge Daten umwandelt, so dass sie dann anschließend auf der Lautsprecherbox ausgegeben werden können.

3.4 Auswahl Softwaretools

3.4.1 Einleitung

In diesem Kapitel geht es um die Recherche und Auswahl von geeigneten Softwaretools, welche für die App-Entwicklung, als auch für die Entwicklung der Software des Microcontrollers verwendet werden.

3.4.2 Softwaretools Microcontroller

Im folgenden werden die verwendeten Bibliotheken im Code des Microcontrollers aufgezählt und kurz beschrieben:

Arduino

Die Arduino-Bibliothek wird verwendet um den ESP32 ähnlich wie einen Arduino programmieren zu können. Es erleichtert dabei die Programmierung enorm, vorallem dann, wenn man schon Vorerfahrung mit der Programmierung von Arduinos hat.

WiFi

Die WiFi-Bibliothek wird verwendet, um die Funktionen der eingebauten WiFi-Antenne des ESP32 zu verwenden. In unserem Projekt wird sie verwendet, um zum einen einen Access Point bereitzustellen und zum Andern als WiFi-Client zu fungieren.

ArduinoJson

Die ArduinoJson-Bibliothek wird verwendet, um Daten in das JSON vformat zu

kodieren. Der Vorteil dabei ist, dass JSON ein weit verbreitetes Format in der Informatik ist und deshalb mit vielen Schnittstellen funktioniert.

WebServer

Die WebServer-Bibliothek wird verwendet, um einen Webserver auf dem ESP32 bereitzustellen. Dieser ist wichtig für den Datenaustausch mittels HTTP, zwischen ESP32 und Smartphone.

Audio

Die Audio-Bibliothek wird verwendet, um die Audio-Daten mittels I2S-Protokoll an den Digital-Analog-Wandler zu senden und diesen zu konfigurieren.

3.4.3 Softwaretools Smartphoneapp

Für die Entwicklung der Smartphoneapp wurde das "React Native Framework verwendet. Mithilfe von React Native ist es möglich eine zentrale Applikation zu entwickeln und diese dann auf mehreren Plattformen wie IOS, Android und auch im Web zu verwenden. React Native basiert auf React, welches ein Framework für die Frontend-Entwicklung ist. Außerdem wird die Radio-Browser-API für die bereitstellung diverser Internetradios verwendet.

4 Entwicklung

4.1 Design Platine

4.2 Bestückung Platine

4.3 Entwicklung Software Adapter

In diesem Kapitel wird der Übergang der Planung in die Entwicklung der Software des Adapters beschrieben. Zur Entwicklung der Software des Microcontrollers wird die IDE Visual Studio Code in Verbindung mit PlatformIO verwendet. Um die Entwicklung einfacher und übersichtlicher zu gestalten, wird die Arduino-Bibliothek in Verbindung mit C++ verwendet.

Programmablauf

Der Ablauf des Programmes wird mit folgendem UML-Ablaufdiagramm veranschaulicht: hier kommt das UML-Ablaufdiagramm her

Klassen

Bei der Entwicklung der Microcontroller-Software wurde aufgrund der Übersichtlichkeit und um die Design-Patterns der Softwareentwicklung einzuhalten, auf objektorientierte Programmierung gesetzt. Das folgende UML-Klassendiagramm veranschaulicht die Beziehung der verschiedenen Klassen zueinander: hier kommt das UML-Klassendiagramm her Im folgenden Teil werden die Klassen und deren Funktionen noch näher beschrieben:

StatusLED

Mithilfe der Klasse StatusLED wird die RGB-LED, welche am ESP32 angeschlossen ist, gesteuert.

NetworkManager

Die Klasse NetworkManager kümmert sich um alle Funktionen, die mit dem WiFi des ESP32 zu tun haben.

AudioManager

Die Klasse AudioManager regelt hauptsächlich das Senden der Audiodaten, vom ESP32 an den Digital-Analog Wandler, mittels I2S-Protokoll.

4.4 Entwicklung Smartphone-App

In diesem Kapitel wird der Übergang der Planung in die Entwicklung der Smartphone-App beschrieben.

4.5 Design Adaptergehäuse

4.6 Fertigung Adaptergehäuse

5 Testen und Fehlerbehebung

5.1 Testen des Gesamtsystems

5.2 auftretende Fehler beheben

5.3 Test auf Cybersecurity

In diesem Kapitel wird der gesamte Code auf Sicherheitslücken getestet.

5.4 Auftretende Sicherheitslücken schließen

In diesem Kapitel werden die bei den Tests aufgetretenen Sicherheitslücken geschlossen.

6 Einzelnachweise

6.1 Literaturverzeichnis

6.2 Abbildungsverzeichnis

6.3 Anhang