



AUDIO HOME

Hochschule Harz

Fach:

Dezentrale Gebäudeautomatisierung,
Spezialisierung: Ambient Assisted Living / Mobile Systeme

Dennis Naujokat und Jannik Sengstack

Abstract

In Haushalten mit Massenmarktprodukten der Medientechnik hat jedes Gerät (Fernseher, Radio, Computer, etc.) seine eigenen Lautsprecher. Bei Zimmerlautstärke sind diese dementsprechend nur im selben Raum hörbar. Um dies zu umgehen, gibt es Systeme, die von verschiedenen Audioquellen genutzt werden können und den Ton im ganzen Haus verteilen. Diese sind unter der Bezeichnung „Multiroom“ bekannt und werden von Herstellern wie Teufel oder Sonos vertrieben. Die Steuerung erfolgt hierbei manuell über eine App und die Geräte kosten oft über hundert Euro pro Komponente. Mit einem solchen System ist es möglich, dass der Ton dem Nutzer durch die Wohnung folgt, wobei jedoch bei herkömmlichen Systemen manuelles Umschalten notwendig ist. Das kann dazu führen, dass Nutzer den Ton einfach lauter stellen um ihn über mehrere Räume hinweg zu hören. Eine daraus resultierende Lärmbelästigung kann innerhalb eines Mehrpersonenhaushaltes, aber auch innerhalb der Nachbarschaft zu Problemen führen. So fühlten sich beispielsweise etwa 20% der Deutschen bereits 2011 stark durch Lärm in der Nachbarschaft belastigt.

Ziel des Projektes ist es, dieses Problem mit Hilfe einer automatischen Lokalisierung des Nutzers und entsprechender Anpassung der anzusteuernenden Wiedergabegeräte zu minimieren. Die Kosten sollen dabei möglichst gering gehalten und gegebenenfalls bestehende Hardware des Nutzers integriert werden. Dafür ist es vorgesehen, dass alle Ein- und Ausgabegeräte zentral vernetzt werden.

Es wird ein zentraler Server eingerichtet, welcher die Steuerung übernimmt. Die einzelnen Geräte werden dabei mittels Mikrocontroller mit dem Server vernetzt. Für die Lokalisierung werden raumbezogene Bluetooth Sender verwendet, die auf ein Gerät des Nutzers (Smartphone, Smartwatch) reagieren und so auf dessen Aufenthaltsraum innerhalb der Wohnung schließen lassen. Mit diesen Informationen kann die Audiowiedergabe entsprechend angepasst werden.

Inhalt

I.	Glossar	3
1.	Einleitung	4
2.	Problemstellung	4
3.	Lösungskonzept	5
4.	Grundlegende Technologien	6
4.1.	Multiroom	6
4.2.	PEAQ	6
4.3.	MQTT	6
4.4.	Multicast	6
4.5.	BLE	7
4.6.	ESP32	7
4.7.	Lyra-T	7
5.	Prototyp	8
5.1.	Audiostreaming	9
5.1.1.	Kommunikation	9
5.1.2.	Audioübertragung	10
5.1.3.	Konfiguration	11
5.2.	Lokalisierung	11
5.2.1	Präsenzerkennung	12
5.2.2	Datenübertragung	13
5.2.3	Konfiguration	13
5.3.	Zusammenspiel	14
6.	Evaluation	16
6.1.	Audio-Evaluation	16
6.1.1.	Versuch objektiver Beurteilung	16
6.1.2.	Subjektive Einschätzung	18
6.2.	Evaluation der Lokalisierung	18
7.	Fazit	22
II.	Literaturverzeichnis	23
III.	Abbildungsverzeichnis	23
IV.	Tabellenverzeichnis	23
V.	Anhang	24
V.1.	Git	24
V.2.	Bedienungsanleitung AudioHome	24
V.3.	Evaluation	26

I. Glossar

Begriff	Beschreibung	Verweis
ESP-IDF	Offizielles Entwicklungs-Framework für den ESP32	
ESP-ADF	Espressif Audio Development Framework	
MQTT	MQ Telemetry Transport	Siehe 4.3
PEAQ	Perceptual Evaluation of Audio Quality	Siehe 4.2
Line-In	Analoger Audioeingang	
Line-Out	Analoger Audioausgang	
BLE	Bluetooth Low Energy	Siehe 4.5
RSSI	Received Signal Strength Indication	Siehe 5.2.1

1. Einleitung¹

Im Jahr 2019 besaßen rund 75% der deutschen Haushalte eine Stereo-Anlage. Mehr als 80% der Haushalte besitzen einen Plasma-, LCD- oder HD-Fernseher.² Vor allem in Mehrpersonenhaushalten kann diese Abdeckung an Multimediasystemen zu Problemen durch Lärmbelästigung führen. Durch ein offenes Fenster, laut eingestellte Fernseher oder die Vibrationen einer Stereoanlage werden andere Haushalte im Mietgefüge belästigt. Lärmbelästigungen geschehen aber nicht nur durch Stereo-Anlagen oder Fernseher, sondern kann auch durch Kleinkinder oder Heimwerken hervorgerufen werden. Laut einer Umfrage aus dem Jahr 2011 fühlen sich mehr als 18% der Deutschen durch Lärm in der Nachbarschaft belästigt.³ Das Projekt AudioHome, welches im Rahmen der Lehrveranstaltungen Dezentrale Gebäudeautomatisierung und Ambient Assisted Living entstand soll dabei gezielt die Belästigung durch Multimedia verringern.

2. Problemstellung⁴

Aktuelle Multiroom-Systeme können zwar die Lautsprecher in den einzelnen Räumen per Smartphone ansteuern und die Lautstärke regulieren (siehe 4.1), wenn man jedoch durch seine Wohnung läuft und die Musik oder den Ton der Fernsehserie mitbekommen möchte ist ein manuelles Umschalten der Lautsprecher notwendig. Hierfür müsste die Steuerungs-App des Herstellers geöffnet werden. Bei schnellen oder kurzweiligen Raumwechseln ist dieses Vorgehen eher lästig und könnte den Nutzer dazu bewegen die Lautstärke zu erhöhen, um den Ton im Nebenraum zu verstehen. In Folge dessen, kann eine Lärmbelästigung trotz eines Multirom-Systems nicht ausgeschlossen werden.

¹ Geschrieben von Jannik Sengstack

² Quelle: (VuMA, 2019)

³ Quelle: (Umweltbundesamt, 2011)

⁴ Geschrieben von Jannik Sengstack

3. Lösungskonzept⁵

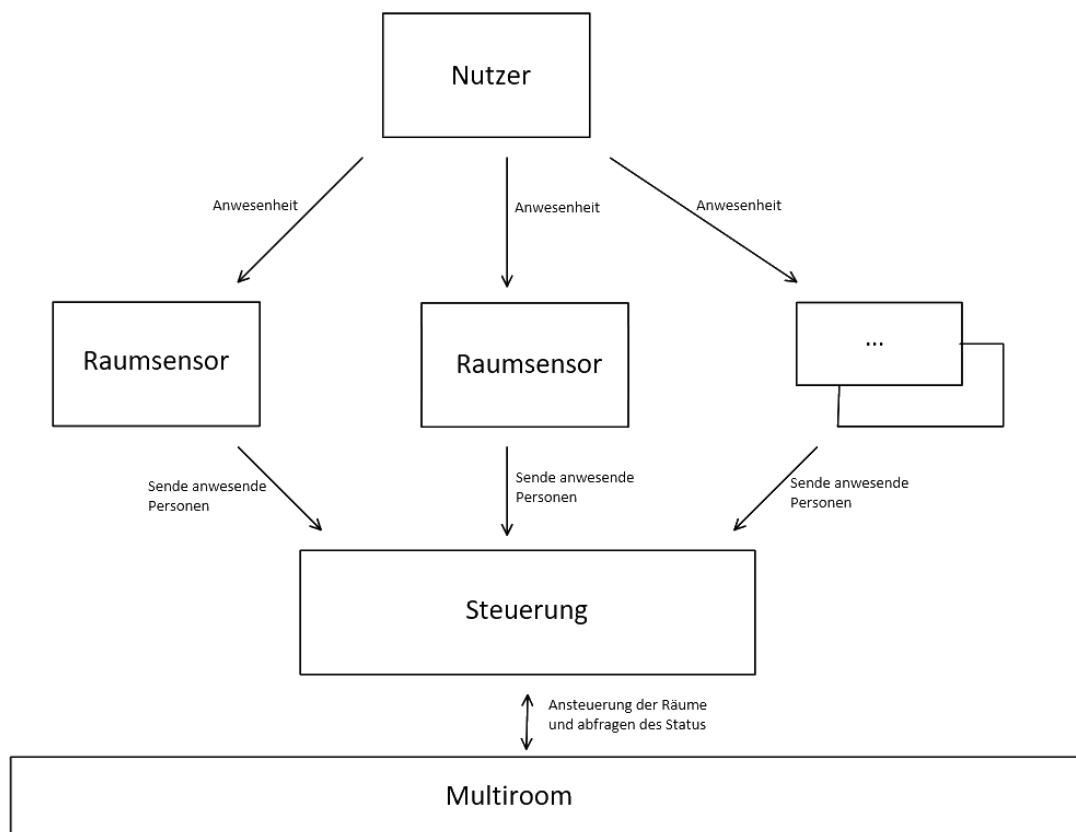


Abbildung 1: Konzeptzeichnung

Das Projekt AudioHome setzt an dieser Stelle an und soll ein herkömmliches Multiroom-System um eine automatische Erkennung des Aufenthaltsraumes des Nutzers erweitern. Dazu werden in Räumen Sensoren angebracht, welche die Anwesenheit des Nutzers erkennen können und diese Informationen an eine Steuereinrichtung weitergeben. Diese kann dann wiederum das Multiroom-System ansteuert. Dadurch soll erreicht werden, dass die Musik dem Nutzer durch die Wohnung folgt und der Nutzer die Wiedergabe nicht mehr manuell auf den neuen Raum übertragen muss. Durch die Automatisierung soll vermieden werden, dass Nutzer die Wiedergabelautstärke erhöhen, um die Wiedergabe auch in anderen Räumen hören zu können.

⁵ Geschrieben von Dennis Naujokat

4. Grundlegende Technologien

Bevor nun die genaue Umsetzung erklärt wird, werden nun noch einige grundlegenden Technologien und Hardware-Geräte, die bei der Realisierung verwendet wurden, kurz erläutert.

4.1. Multiroom⁶

Bei einem Multiroom-System handelt es sich im Allgemeinen um eine Vernetzung von Lautsprechern und Wiedergabegeräten, bei der jedes Wiedergabegerät auf allen Lautsprechern wiedergeben kann. Ziel ist es dabei die Wiedergabe nicht nur am Bereitstellungsort, sondern an jedem beliebigen Ort genießen zu können. Dabei können meist auch mehrere Lautsprecher, die sich beispielsweise im selben Raum befinden, zu einer Gruppe zusammengefasst werden, um diese wie einen einzigen verwenden zu können.

Obwohl auch eine analoge Verkabelung möglich ist, werden solche Systeme meist mit einer Kommunikation über das Netzwerk in Verbindung gebracht. Jedoch gibt es zum aktuellen Zeitpunkt noch keine Standardisierungen zur Steuerung und Übertragung der Daten. Bekannte Hersteller sind unter anderem Sonos und Teufel.

4.2. PEAQ⁷

Bei PEAQ (Perceptual Evaluation of Audio Quality) ist ein Verfahren, welches einen möglichst objektiven Vergleich der Audioqualität einer Aufnahme zum Originalstück ermöglichen soll. Original und Testsignal werden dabei mittels eines peripheren Modells verglichen, welches als Ergebnis zwei Werte liefert: den ODG (Objective Difference Grade) und den DI (Distortion Index). Der ODG ist dabei eine objektive Variante des im ITU-R-Standard definierten SDG (Subjective Difference Grade), welcher die Beeinträchtigung von -4 (störend) bis 0 (kaum merklich) einschätzt.

Der ODG selbst variiert von -4 bis 0.7, dies liegt dem zugrunde, dass das PEAQ-Verfahren auf ein neuronales Netz zurückgreift, welches mit Werten des SDG angelernt wurde. Positive Werte des ODG bedeuten dabei, dass Original und Testsignal zum Verwechseln ähnlich sind. Der DI dient dazu, um auch mehrere Aufnahmen mit einem ODG unter -3.6 untereinander Vergleichen zu können.

4.3. MQTT⁸

MQTT steht für „MQ Telemetry Transport“ und ist ein Protokoll zur Übertragung von Daten. Im Gegensatz zu TCP oder HTTP wird keine beidseitige Verbindung aufgebaut. Ein MQTT Client sendet („published“) seine Daten zu einem bestimmten Zeitpunkt an einen MQTT-Broker. Dieser Broker arbeitet die eingehenden Nachrichten ereignisgesteuert ab. Andere Geräte im Netzwerk können diese Daten vom Broker abfragen indem sie eine bestimmte Topic Adresse abonnieren („subscribed“). Jedes Topic ist ähnlich einer Baumstruktur aufgebaut. Die einzelnen Ebenen sind durch „/“ getrennt und ein Gerät im Netzwerk kann sowohl Ober- als auch Unterthemen abonnieren. Nachrichten können des Weiteren mit einer von drei „Quality of Service“-Einstufungen versehen werden: Sie wird einmal nur gesendet, unabhängig ob sie korrekt empfangen wird, sie wird mehrfach gesendet, bis der Empfang quittiert wurde (kann auch mehrfach empfangen werden) oder es wird sichergestellt, dass die Nachricht genau einmal empfangen wird.

4.4. Multicast⁹

Bei Multicast handelt es sich um eine Möglichkeit eine Punkt-zu-Mehrpunktverbindung auf OSI-Layer 3 zu realisieren. Der Vorteil ist hierbei, dass der Sender nur eine Nachricht senden muss und diese

⁶ Geschrieben von Dennis Naujokat

⁷ Geschrieben von Dennis Naujokat

⁸ Geschrieben von Jannik Sengstack und Dennis Naujokat

⁹ Geschrieben von Dennis Naujokat

von beliebig vielen Empfängern empfangen werden kann. Im Gegensatz zum Broadcast, der an alle Clients im selben Netz geht, muss sich beim Multicast der Empfänger erst auf die Multicastadresse registrieren.

Der Multicastadressraum reicht bei IPv4 von 224.0.0.0 bis 239.255.255.255. Das Management der IP-Multicastgruppen wird dabei standardmäßig von den Routern übernommen und erfolgt über das Internet Group Management Protocol (IGMP). Über dieses Protokoll können sich Clients in eine Multicastgruppe anmelden und die Router die IP-Multicast-Datenströme beziehen. Die Routing der Multicastströme erfolgt über spezielle Protokolle, auf welche hier nicht weiter eingegangen wird, da sich das Projekt nur im lokalen IP-Raum befindet.

4.5. BLE¹⁰

Bluetooth Low Energy ist ein Funkstandard, der auf dem Full-Stack Bluetooth basiert. BLE benötigt weniger Energie, daraus folgt eine geringere Reichweite von 10m bis maximal 40m bei freier Sicht auf das Ziel. Bluetooth Low Energy wird für die Positionsbestimmung innerhalb von Gebäuden, aber auch in Sensoren oder Fitnessarmbändern verwendet. Wegen des geringen Stromverbrauches sind die Akkus der Fitnessarmbänder länger haltbar und die Sensoren können auch im Niederspannungsbereich bei geringer Stromzufuhr arbeiten.

Standardmäßig wird die Bluetooth Version 4.0 gewählt, da viele IoT-Lösungen bereits mit dieser Version entwickelt wurden und sie demnach Kosten- und Energieeffizient ist. Durch die Version 5.1 und der „Angle of Arrival“ (AoA) Technologie, ist eine zentimetergenaue Ortung bei Sichtkontakt möglich.

4.6. ESP32¹¹

Der ESP32 ist ein 32-Bit Mikrocontroller des Herstellers Espressif. Standardmäßig auf diesem Controller integriert sind ein Bluetooth Modul, Wi-Fi, RF (Radio Frequency) -Transmitter und – Receiver, Schnittstellen (Pins) für Peripherie, integrierter Systemtakt (Clock generator), ein Prozessor und 4MB Programmspeicher. Dieser Mikrocontroller ist für den niedrigenenergiebedarf ausgelegt und wird deswegen in vielen IoT-Anwendungen genutzt.

4.7. Lyra-T¹²

Bei dem ESP32-LyraT V4.3 handelt es sich um eine Entwicklungs-Board für Audioanwendungen des Herstellers Espressif. Es basiert auf dem ESP32-Microcontroller und erweitert diesen um einen Chipsatz zur Verarbeitung von Audioströmen, sowie ein Mikrofon, Line-In- und Line-Out-Anschlüsse, Anschlüsse für Kopfhörer, 6 frei programmierbare Tasten, einen Micro-SD-Slot und ein Batteriemanagement. Zusätzlich wird von Espressif noch eine umfangreiche API zur einfacheren Programmierung zur Verfügung gestellt: die ESP-ADF.

¹⁰ Geschrieben von Jannik Sengstack

¹¹ Geschrieben von Jannik Sengstack

¹² Geschrieben von Dennis Naujokat

5. Prototyp¹³

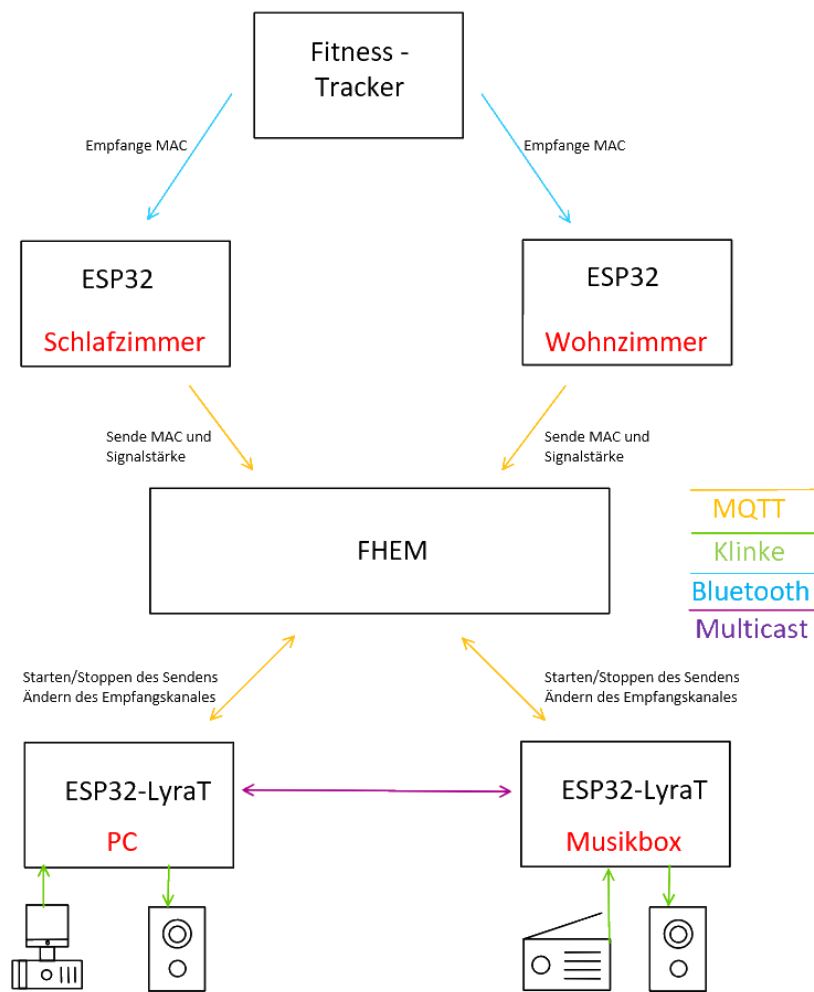


Abbildung 2: Prototypzeichnung

Zur Umsetzung des Konzeptes mussten zwei Teilaufgaben erfüllt werden. Zum einen musste ein einfaches Multiroom-System aufgebaut werden, welches sich leicht ansteuern lässt. Zum anderen musste ein System zur Präsenzerkennung entwickelt werden. Als weitere Anforderung, sollte sich das System leicht auf andere Multiroom-Systeme umstellen lassen, wodurch eine möglichst offene Art zur Kommunikation gewählt werden musste.

Bei der Suche nach offenen Kommunikationssystemen wurde dann schnell MQTT gefunden, welches bereits in der SmartHome-Technik viel Verwendung findet. Es ist leicht verständlich, leicht zu implementieren und auf den meisten Systemen lauffähig. Zudem ist das Protokoll relativ klein und verbraucht wenig Rechenzeit, wodurch es auch auf Microcontrollern zum Einsatz kommen kann. Da wie erwähnt in vielen SmartHome-Umgebungen MQTT bereits im Einsatz ist, wurde ein vorhandener MQTT-Server als gegeben vorausgesetzt. In der Testumgebung wurde dafür ein bereits vorhandener Server genutzt, welcher auf dem Node-JS-Paket „mosca“ basiert.

Ein weiterer wichtiger Faktor war, welches System die Steuerung übernehmen sollte. Da eine automatische Erkennung des Aufenthaltsraumes auch für andere Zwecke wie evtl. eine Steuerung der Beleuchtung genutzt werden kann, sollte das System auch in diesem Bereich möglichst offen und erweiterbar sein. Dadurch konnten wieder Parallelen zum SmartHome-Bereich gezogen werden. Um

¹³ Geschrieben von Dennis Naujokat

das System also mit SmartHome-Technologie kompatibel zu halten, böten sich auch deren Steuerungssysteme an. Im deutschen Bereich ist gerade die SmartHome-Managementsoftware FHEM sehr beliebt. Diese bietet auch eine sehr offene Programmierschnittstelle, wodurch der notwendige Steuercode leicht integriert werden konnte.

Im Folgenden werden nun das realisierte Multiroom-System und die Präsenzerkennung genauer erläutert.

5.1. Audiostreaming¹⁴

Bei der Wahl der Hardware für das Multiroom-System wurden zwei grundsätzliche Möglichkeiten in Betracht gezogen. Entweder der Einsatz eines Microcontroller mit zusätzlicher Hardware zur Audioverarbeitung oder Controllerboard mit integrierten Audioanschlüssen. Bei der ersten Variante muss sichergestellt werden dass die Hardware untereinander kompatibel ist. Zusätzlich gibt es nicht viele Audiocontroller die sowohl über einen Line-In- als auch über einen Line-Out-Anschluss verfügen. Bei einem Controllerboard hingegen ist die Kompatibilität zwar sichergestellt, jedoch verfügen die Meisten Boards nur über einen Line-Out-Anschluss und keinen Line-In, was wiederum das hinzufügen zusätzlicher Hardware benötigen würde. Zudem ist der Energieverbrauch von Controllerboard meist höher, da diese eher für komplexere Aufgaben entwickelt werden.

Das ESP32-LyraT-Board stellt eine Art Kombination beider Ansätze da. Es basiert auf einem ESP32-Microcontroller und erweitert diesen um zusätzliche Audiohardware. Zudem bringt das zugehörige ESP-ADF viele Funktionalität zur Audioverarbeitung mit. Daher wurde sich für den Einsatz dieses Boards entschieden.

5.1.1. Kommunikation¹⁵

Topic	Daten
AudioDevice/Line-In/<Eingabegerät Name>/Address	Adresse zum Abrufen des Audiostreams
AudioDevice/Line-In/<Eingabegerät Name>/plays	Status der Wiedergabe
AudioDevice/Line-In/<Eingabegerät Name>/log	Loginformationen
AudioDevice/Line-Out/<Ausgabegerät Name>	Empfang der Audioquelle

Tabelle 1: Audiostreaming, MQTT Topics

Für die Implementierung der Kommunikation wurde die in der ESP-IDF bereitstehende Bibliothek „mqtt_client.h“ verwendet. Diese ist leicht zu verwenden und läuft zuverlässig auf dem Microcontroller. Außerdem managend dieser automatisch einen Neuaufbau der Verbindung bei Verbindungsproblemen oder wenn der MQTT-Server zeitweise nicht erreichbar ist. Als Obertopic dient „AudioDevice“, darunter befinden sich die Topics „Line-In“ und „Line-Out“, welche die Eingabe- und Ausgabekanäle trennen.

Bei den Line-In-Geräten folgt dann der Name des Gerätes als Untertopic und die Daten werden dann in 3 weiteren Untertopic ausgetauscht. Unter dem Topic „Address“ wird beim Start die Adresse die Adresse veröffentlicht, unter welcher der Audiostream abgerufen werden kann. Das Topic „plays“ wird sowohl zum Senden als auch zum Empfangen des Wiedergabezustandes genutzt. Wird eine Nachricht empfangen, wechselt die Übertragung in den entsprechenden Zustand (starten oder beenden der Übertragung). Gesendet werden können muss der Zustand auch, da das Übertragen des Audiostroms des Eingabegerätes auch durch Betätigung der Play-Taste gestartet bzw. gestoppt werden kann und dies in diesem Fall auch nach außen mitgeteilt werden muss. Da das Board

¹⁴ Geschrieben von Dennis Naujokat

¹⁵ Geschrieben von Dennis Naujokat

standardmäßig über die UART Schnittstelle logt, werden noch zusätzlich Informationen über Statusänderungen des Microcontroller über das Topic „log“ übertragen. Dadurch muss nicht immer ein Gerät über die UART-Schnittstelle des Boards angeschlossen sein, um Loginformationen zu erhalten.

Bei der Audioausgabe werden die Informationen direkt unter dem Namen des Ausgabegerätes empfangen. Als Datenwert wird hier die Adresse, unter welcher der Audiostream empfangen werden kann, oder „none“ zum Beenden des Abspielens erwartet.

5.1.2. Audioübertragung¹⁶

Da es sich nur um einen Prototypen handelt und in der verfügbaren Zeit keine geeignete Bibliothek zum Streamen eines Audiopuffers gefunden werden konnte, wurde sich auf einfaches senden der Daten über einen Multicast entschieden. Die Audioeingabe und –ausgabe wurden über Standardmöglichkeiten der ESP-ADF mit einem Ein- bzw. Ausgabepuffer verknüpft. Die Puffergröße musste für eine stabile Übertragung der Daten so beschränkt werden, dass dieser in ein einzelnes Ethernetpaket passt. Bei einer größeren Datenmenge pro UDP-Paket konnte das Empfangsgerät (anderer LyraT) die Pakete nicht mehr empfangen. Dadurch ist die Puffergröße auf max. 1500 Bytes begrenzt. Für den praktischen Einsatz wurde die Größe jedoch auf 1300 Bytes festgelegt, da sonst schnell Probleme mit der Hauptspeichergröße des ESP32 auftraten. Diese treten auch bei dieser Puffergröße noch gelegentlich auf, jedoch deutlich weniger. Wenn diese Probleme zu häufig in kurzer Zeit auftreten, führt dies zu einem Absturz der WLAN-Schnittstelle. In diesem Fall wird das Board automatisch neugestartet, um die Probleme zu beheben. Weitere Versuche die Qualität durch Komprimierung der Daten zu erhöhen haben sich als konterproduktiv herausgestellt. Beim Versuch die Audiodaten im WAV-Format zu codieren, sind Probleme beim Decodieren entstanden (evtl. durch ausgelassene Pakete). Und der Versuch die Daten durch Einsatz eines Filters zu reduzieren wirkte sich auch negativ auf die Qualität aus.

Das Senden der Audiodaten erfolgt in einen separaten Thread, welcher in einer Schleife erst die Daten in den Puffer einließt und diesen dann als Multicast versendet. Zum Stoppen der Übertragung wird dieser Thread pausiert, um ihn dann später wieder aktiv werden zu lassen. Das Pausieren ist dabei so organisiert, dass dies immer am Ende eines Schleifendurchlaufes passiert, damit sichergestellt werden kann, dass sich der Thread immer in einem sicheren Zustand befindet.

Empfangen werden die Daten ebenfalls in einem einzelnen Thread, dieser arbeitet ähnlich wie beim Senden in einer Schleife. Zuerst wird auf ein eingehendes Paket auf der eingestellten Multicastadresse gewartet und dann die empfangenen Daten in den Puffer für die Audioausgabe geschrieben. Besonders ist hierbei, dass dieser Thread nicht wie der Thread zum Senden pausiert wird, sondern bei einem Wechsel der Abspieladresse oder einem Stoppen des Abspielens beendet wird. Dies hat seinen Ursprung darin, dass für eine neue Quelladresse auch ein neuer Socket generiert werden muss. Dazu sollte der alte Socket jedoch erst geschlossen werden, daher geschieht ein solches Stoppen oder Wechseln der Quelladresse immer in einem neuen Thread, um den aufrufenden Thread (z.B. den Eventhandler des MQTT-Client) nicht aufzuhalten, während auf das beenden des Empfangsthreads gewartet wird.

¹⁶ Geschrieben von Dennis Naujokat

5.1.3. Konfiguration¹⁷

Parameter	Typ	Beschreibung
AUDIOHOME_DEBUG	Boolean	Ausgabe von Debuginformationen
AUDIOHOME_RESTART_AFTER_NETWORK_COLLAPSE	Boolean	Neustart bei Zusammenbruch der Netzwerkverbindung nach Speicherproblemen
AUDIOHOME_RESET_PLAYS_ON_START	Boolean	Mitteilen nach Außen, das die Wiedergabe beendet wurde nach Neustart
AUDIOHOME_WIFI_SSID	String	SSID der WLAN-Verbindung
AUDIOHOME_WIFI_PASSWORD	String	Passwort der WLAN-Verbindung
AUDIOHOME_MULTICAST_TTL	Int	TTL der Multicastpakete
AUDIOHOME_MULTICAST_IPV4_ADDR	String	Adresse unter der die Audiodaten gesendet werden
AUDIOHOME_UDP_PORT	Int	Port unter dem die Audiodaten gesendet werden
AUDIOHOME_LOCAL_SEND_PORT	Int	lokaler Port der zum Senden genutzt wird
AUDIOHOME_LINE_IN_NAME	String	Name des Wiedergabegerätes
AUDIOHOME_LINE_OUT_NAME	String	Name der Lautsprecher
AUDIOHOME_MQTT_SERVER	String	Adresse des MQTT-Servers

Tabelle 2: Audiostreaming, Konfigurationsparameter

Die gesamte Konfiguration liegt in der „config.h“ damit bei Änderungen und dem Hinzufügen neuer Geräte nur eine Datei bearbeitet werden muss. Wichtige Parameter sind hierbei die SSID und Passwort des WLAN-Zugangs und die Multicastadresse und der zugehörige Port. Dabei ist zu beachten das die Kombination auf IP-Adresse und Port einmalig sein muss, das heißt bei jedem LyraT anders sein muss. Diese beiden Parameter zusammen ergeben die Adresse die beim Start gesendet wird und die dann im Fall einer Wiedergabe an die Lautsprecher (andere LyraT) weitergegeben wird.

Zur Zuordnung ist es des Weiteren wichtig, dass die Namen der Line-In- und Line-Out-Geräte eindeutig beschrieben werden. Dabei dürfen jedoch nur Buchstaben, Zahlen und der Unterstrich verwendet werden.

Der letzte wichtige Parameter, der eingestellt werden muss, ist die Adresse des MQTT-Servers. Alle anderen Parameter können und sollten einfach unverändert übernommen werden.

5.2. Lokalisierung¹⁸

Für die Lokalisierung kommen ESP32 Mikrocontroller in Einsatz. Alternativ könnte man hier auch mit einem Raspberry Pi pro Raum arbeiten. Der Raspberry Pi verfügt erst ab der Version 3 ein integriertes Bluetooth Modul, bei den Vorgängerversionen müsste man ein Bluetooth Adapter extra kaufen. Der Raspberry Pi 3 hat 1GB Arbeitsspeicher und einen 1200MHz 4-Kern ARM Cortex A-53 64-Bit Prozessor. Mit diesen Geräteeigenschaften und einem Preis von 34,02 €¹⁹ ist ein Raspberry Pi 3 für eine einfache Präsenzerkennung geeignet, da er Bluetooth Low Energy empfangen kann und genug Speicherkapazität für das benötigte Programm hat.

Im Vergleich hierzu steht ein ESP32 der NodeMCU Variante. Das Board lässt sich per USB-Verbindung zum PC programmieren und arbeitet dieses Programm dauerhaft ab. Mit einem Programmspeicher von 4MB sind und einem SRAM mit 512KB sind die Programme in ihrer Größe und Komplexität

¹⁷ Geschrieben von Dennis Naujokat

¹⁸ Geschrieben von Jannik Sengstack

¹⁹ Preis von reichelt.de (abgerufen am 14.08.20)

begrenzt. Der Preis eines NodeMCU ESP32 beträgt 9,74€²⁰. Da das Programm zur Lokalisierung eines BLE Gerätes nicht komplex ist und einfach in Dauerschleife abgearbeitet werden kann wurde sich für den NodeMCU ESP32 entschieden. Ein Raspberry Pi wäre für diese Anwendung überdimensioniert und kostet knapp 4-mal so viel wie der ESP32.

5.2.1 Präsenzerkennung²¹

Für die Präsenzerkennung wurden Bibliotheken aus dem Arduino Core für ESP32 benutzt. Unter anderem Bibliotheken für die Suche nach BLE Geräten in der Umgebung (<BLEScan.h>), das Abbilden der Bluetooth MAC-Adresse <BLEAddress.h>, oder das Auslesen der Daten des gefundenen Geräts (<BLEAdvertisedDevice.h>).

Die hier genutzte Präsenzerkennung benötigt einen ESP32, der ständig nach anderen BLE Geräten (hier: Fitnessarmband) in seiner Reichweite. Da BLE im Gegensatz zu „Full-Stack“ Bluetooth eine geringere Leistung und somit auch Reichweite hat deckt ein ESP nur einen Raum ab. Dies macht man sich zum Vorteil, denn so kann man Interferenzen mit anderen ESP vermindern. Der Unterschied zum „Full-Stack“ ist, dass bei BLE die Datenübertragung mit kleineren Datenpaketen in kleinen Schüben und nicht als ein dauerhafter Stream stattfindet. Die BLE Endgeräte (z.B. Fitnessarmbänder) sind darauf ausgelegt keine dauerhaften Übertragungen und nur geringe Datenmengen zu erhalten.²² Die geringere Übertragungsrate und die kleineren Datenmengen sorgen dafür, dass ein BLE Endgerät sehr Energieeffizient ist und oft bis zu mehreren Wochen/ Monaten mit einer Akkuladung auskommt. Bluetooth und BLE werden beim ESP32 über die gleiche Antenne gesendet und empfangen. Aufgrund der Nutzung des gleichen Frequenzbandes und der selben Hardware haben sie ein ähnliches Ausbreitungsverhalten und werden durch Wände gestoppt. Dieses Verhalten bietet einen weiteren Vorteil in Bezug auf die Präsenzerkennung, da man mit hoher Wahrscheinlichkeit sagen kann, dass sich das Fitnessarmband in einem Raum aufhält, wenn es von dem dortigen ESP mit dem „BLEScan“ gefunden wurde. Durch die Nutzung des 2,4-GHz-ISM-Bandes kommt es zu Überschneidungen mit anderen Funkverbindungen wie zum Beispiel dem Hausinternen WLAN-Netz. Um mögliche Interferenzen auf diesem Kanal zu vermeiden hat der ESP32 ein integriertes Feature namens „Adaptive Frequency Hopping“ (AFH). Das 2,4GHz Band lässt sich noch weiter unterteilen, denn in diesem Band gibt es mehrere Channels, die sich in ihrer Frequenz unterscheiden. AFH sorgt für eine reibungslose Nutzung des 2,4-GHz Bandes. Hierbei handelt es sich um die Erkennung ob einer der Channels des 2,4GHz Bandes schon in Benutzung ist, indem nach Interferenzen gesucht wird, und bei Interferenzen wird ein anderer Channel abgefragt, solange bis ein freier Channel gefunden wird. Über diesen kann die Verbindung aufgebaut und der Datenaustausch stattfinden.

Auch wenn die BLE Signale des ESP32 durch die Wände eines Raumes begrenzt werden, kann es beim Raumwechsel dazu kommen, dass mehrere ESP in der Wohnung das Fitnessarmband erkennen. Es kann nicht eindeutig festgestellt werden in welchem Raum sich das Fitnessarmband befindet. Um diesem Problem aus dem Weg zu gehen, wird neben dem Namen und der MAC-Adresse auch noch der RSSI-Wert ausgelesen. RSSI steht für „Received Signal Strength Indication“ und ist ein Wert, der die Signalstärke der Verbindung in widerspiegelt. Es wird die am Empfänger angekommene Signalstärke in Relation zur Maximalen Sendeleistung gesetzt, je näher der Empfänger am Sender ist, desto größer ist die dort ankommende Signalstärke. Der RSSI Wert ist ein Leistungspegel und wird in dBm angegeben. Trotz des Logarithmischen Verlaufs, kann man mit diesem Wert auf die ungefähre Entfernung schießen und ein zentraler Server der alle ESP Daten verarbeitet, kann die RSSI Werte

²⁰ Preis von reichelt.de (abgerufen am 14.08.20)

²¹ Geschrieben von Jannik Sengstack

²² Inside Bluetooth Low Energy, (Gupta, 2016)

vergleichen und weiß zu welchem ESP das Fitnessarmband die stärkste Verbindung hat, also die geringste Distanz hat.

5.2.2 Datenübertragung²³

Zum Übertragen der Daten vom jeweiligen ESP an einen zentralen Server wird die Bibliothek <PubSubClient.h> genutzt. Sie fügt Funktionen für den Umgang mit einem MQTT-Client hinzu, der Daten an einen Broker übertragen kann.

Für die Datenübertragung an den Audio Home Server wird sich bei jedem Sendevorgang erneut mit dem MQTT-Broker verbunden und dann eine Nachricht gesendet. Die Nachricht enthält den RSSI-Wert zu dem gefundenen BLE-Gerät. Bei mehreren BLE-Geräten im Raum werden mehrere Nachrichten gesendet. Dabei ist zu beachten, dass jedes Gerät ein eigenes Topic hat.

Audio Home Topic:

Lokalisierung/Schlafzimmer/00:00:00:00:00:00

In der untersten Ebene des Topics befindet sich eine 12 Ziffern lange Zahl, welche jedes BLE-Gerät eindeutig identifiziert. Anstatt der „0“ können hier Hex-Ziffern und Buchstaben stehen. Jedes Byte wird durch einen Doppelpunkt getrennt. Die ersten zwei Byte der Adresse stehen hier für den NAP (Non-significant Address Part), die Werte dieser zwei Byte werden für das Frequency Hopping und die Synchronisation verwendet. Das dritte Byte ist der UAP (Upper Address Part) teilt verschiedenste Bluetooth-Spezifizierte Algorithmen. Zusammen ergeben die drei Byte von NAP und UAP eine eindeutige Herstellerkennung diese ist als OUI (Organizationally Unique Identifier) festgelegt. Die letzten drei Byte sind für die Geräteerkennung, die Zuweisung geschieht vom Hersteller und hier muss auch drauf geachtet werden jede Kombination nur einmal zu verwenden. Dieser LAP (Lower Address Part) bildet zusammen mit dem UAP den signifikanten Teil der Mac-Adresse.²⁴

5.2.3 Konfiguration²⁵

Parameter	Typ	Beschreibung
SSID	String	SSID der WLAN-Verbindung
PASSWORD	String	Passwort der WLAN-Verbindung
SERVER	String	Adresse des MQTT-Servers
ZIMMER	String	Zimmer in der Wohnung (z.B.: Schlafzimmer)

Die Änderungen müssen direkt im Arduino-Sketch geändert werden und jeder ESP muss einmal mit diesen Daten geflasht werden. Es ist zu beachten, dass jeder ESP ein anderes Zimmer zugewiesen bekommt. Die anderen Daten sind für alle ESP in einer Wohnung gleich.

Bei falschen WLAN-Daten wird sich der ESP nicht mit dem Netzwerk verbinden können und sich selbst Neustarten. Kontrolliert werden kann dies in dem man nach dem flashen den Seriellen Monitor bei Arduino öffnet und sich die Ausgaben des ESP anschaut. Bei einer Verbindung zum Netzwerk wird die IP-Adresse ausgegeben, andernfalls müssen die Daten geändert und der ESP neu geflasht werden. Das gleiche gilt, wenn die Adresse des MQTT-Servers falsch angegeben wurde.

Bei falscher Benennung eines Zimmers kann es im späteren beim Kontrollieren am FHEM Server zu Problemen führen, da man nur erschwert nachvollziehen kann welcher Raum wirklich gemeint ist.

²³ Geschrieben von Jannik Sengstack

²⁴ https://macaddresschanger.com/what-is-bluetooth-address-BD_ADDR

²⁵ Geschrieben von Jannik Sengstack

Nach Vergabe der Zimmernamen werden die MQTT aufgebaut. Dies geschieht automatisch und der Benutzer muss nicht eingreifen.

5.3. Zusammenspiel²⁶

Die bisherigen Geräte senden bis lang nur ihre Daten über MQTT oder empfangen Befehle darüber. Im Folgenden wird daher auf eigentliche Logik eingegangen, die die Geräte koordiniert und steuert. Wie zuvor erwähnt wird dazu ein FHEM-Server eingesetzt auf welchem nur die „99_AudioHomeUtils.pm“ installiert werden muss und einmal der Perlbefehl „AudioHome_init()“ ausgeführt werden muss (siehe Anhang V). Dadurch werden die Basisgeräte initialisiert. Da das System zunächst davon ausgeht, dass der MQTT-Server auf demselben Gerät betrieben wird, muss dessen IP-Adresse gegebenen falls über den Manager angepasst werden.

MQTT2_CLIENT		
AudioHome_MQTT_Client		opened
MQTT2_DEVICE		
AudioHome_LineIn_Musikbox		
AudioHome_LineIn_PC		
AudioHome_MQTT_Listener		OK
at		
AudioHome_LineOut_Contact		Next: 11:21:37
dummy		
AudioHome_LineOut_Musikbox_Lautsprecher		
AudioHome_LineOut_PC_Lautsprecher		
AudioHome_Manager		OK
AudioHome_room_Schlafzimmer		spiele Nutzer: none
AudioHome_room_Wohnzimmer		spiele Nutzer: Test
AudioHome_user_Test	Aufenthaltsraum: Wohnzimmer	Plays <input type="text" value="PC"/>
notify		
AudioHome_LineIn_Notify		2020-08-07 10:47:59
AudioHome_ManagerLogic		active
AudioHome_setUserPlayNotify		2020-08-07 10:47:59

Abbildung 3: FHEM Benutzeroberfläche in Betrieb

Das Programm ist so geschrieben, dass es möglich viel vollautomatisiert generiert. Daher werden sowohl die Line-In- und Line-Out-Geräte sowie die Räume automatisch generiert. Die Audiogeräte, werden generiert, sobald unter den jeweiligen Topics (siehe Tabelle 1) die erste Nachricht empfangen wird und die Räume sobald unter dem Topic „Lokalisierung/<Raumname>/“ eine Nachricht eingeht. Die Räume sind nach der Generierung sofort einsatzbereit, denn Audiogeräten muss allerdings noch ein Raum zugewiesen werden. Um die Handhabung zu erleichtern zeigen diese Geräte das jeweils in der Benutzerschnittstelle von FHEM zusammen mit dem notwendigen Befehl an. Einzig die Nutzer müssen über den Manager manuell angelegt werden und ihnen dann eine Bluetooth-MAC-Adresse zugewiesen werden.

Der Aufenthaltsraum der Nutzer wird über die jeweilige MAC-Adresse ermittelt, indem der ihr entsprechende RSSI-Wert in allen Räumen verglichen wird. Dabei werden nur Werte berücksichtigt

²⁶ Geschrieben von Dennis Naujokat

die vorkonfiguriertes alter nicht überschritten haben (standardmäßig 10 Sekunden). Dies dient dazu, um sicherzustellen, dass es sich nicht um veraltete Werte handelt.

Wenn ein Gerät mitteilt, dass eine Wiedergabe beginnt, wird diese automatisch allen Nutzern im Raum zugewiesen. Nutzer können zudem über das Webinterface mittels des Dropdown-Menüs ein spezifisches Line-In-Gerät zum Abspielen auswählen bzw. auf diese Weise auch die aktuelle Wiedergabe ändern oder stoppen.

Beim Raumwechsel eines Nutzers wird seine aktuelle Wiedergabe automatisch auf den neuen Raum übertragen, wenn dort noch kein anderer Nutzer etwas abspielt. Der alte Raum spielt dann noch für eine kurze Zeit (standardmäßig 10 Sekunden) den Ton weiterhin ab, um bei kurzzeitigen oder häufigen Raumwechseln ein ständiges umschalten zu verhindern.

6. Evaluation²⁷

Zur Evaluations des Systems musste zum einen die Qualität des umgesetzten Multiroom-Systems beurteilt werden und zum anderen wie genau die Raumerkennung mittels BLE funktioniert. Diese beiden Aspekte werden im Folgenden genauer beleuchtet.

6.1. Audio-Evaluation²⁸

Bei der Evaluierung des Multiroom-Systems wurde sich primär auf die Beurteilung der Audioqualität gestützt und wie Zuverlässig das System arbeitet. Bei der Audioqualität sollte beurteilt werden wie viel schlechter diese durch den Einsatz des Multiroom-Systems wird. Für die Beurteilung wurde auf eine Implementierung des PEAQ-Algorithmus (siehe 4.2) von P. Kabal²⁹ für MATLAB zurückgegriffen. Als Testmusikstücke wurde sich auf 3 Stücke beschränkt, welche jeweils als Vertreter für 3 unterschiedliche Genre ausgewählt wurden.

Bei dem ersten Teststück, im Folgenden als Teststück 1 referenziert, handelt es sich um „Lost Soul (feat. Selphius)“ aus dem Album „Dreaming Stargazers“ von Frozen Starfall³⁰ und es wurde als Vertreter des Genre „electro house“ ausgewählt. Das zweite Teststück, folgend Teststück 2, ist „Said - Loli Patrol“ von „Dokueki Leal Arturo“³¹ und soll als Vertreter des Genre „Dubstep“ zeigen, wie das System auf starke Bässe reagiert. Und als Teststück 3 wurde der „Ritt der Walküre“ von Richard Wagner als Vertreter der klassischen Musik gewählt, um einen Kontrast zu den ersten beiden Stücken darzustellen.

Jedoch haben die Tests gezeigt, dass die Verzerrung durch die Hardware des LyraT so groß ist, dass der Algorithmus die Qualität sehr schlecht bewertet, obwohl diese subjektiv betrachtet nicht so viel schlechter ist. Aus diesem Grund wird dann im späteren auch eine subjektive Einschätzung der Tonqualität folgen.

6.1.1. Versuch objektiver Beurteilung³²

Da der PEAQ-Algorithmus sehr anfällig für asynchrone Dateien ist, wurde am Anfang jedes Teststücken ein lautes Klatschen eingefügt. Danach wurde die Originalquelle über einen LyraT abgespielt und über einen zweiten Empfangen und aufgezeichnet. Da damit zu rechnen war, dass die Hardware des LyraT selbst auch zu einer Verschlechterung der Audioqualität führt, wurde zusätzlich nochmals für jedes Teststück eine Aufnahme über einen separaten LyraT gemacht, welcher so konfiguriert wurde, dass dieser das Eingangssignal direkt an seinen eigenen Ausgang weiterleitet. Diese Aufnahme wurde in allen Test mit der Bezeichnung „Passthrough“ ausgestattet und ein Vergleich zwischen ihr und den Testaufnahmen sollte zeigen, um wie viel sich die Qualität nur durch das Multiroom-System verringert ohne Berücksichtigung der Hardware des LyraT.

Vergleichsstück 1	Vergleichsstück 2	ODG
Teststück 1 Original	Teststück 1 Passthrough	-3.466
Teststück 1 Original	Teststück 1 Multiroom-Aufnahme	-3.526
Teststück 1 Passthrough	Teststück 1 Multiroom-Aufnahme	-3.789
Teststück 2 Original	Teststück 2 Passthrough	-2.852
Teststück 2 Original	Teststück 2 Multiroom-Aufnahme	-3.7
Teststück 2 Passthrough	Teststück 2 Multiroom-Aufnahme	-3.716
Teststück 3 Original	Teststück 3 Passthrough	-3.298

²⁷ Geschrieben von Dennis Naujokat

²⁸ Geschrieben von Dennis Naujokat

²⁹ Quelle: (Kabal, 2003)

³⁰ Quelle: (Dreaming Stargazers, 2019)

³¹ Quelle: (Arturo, 2017)

³² Geschrieben von Dennis Naujokat

Teststück 3 Original	Teststück 3 Multiroom-Aufnahme	-3.788
Teststück 3 Passthrough	Teststück 3 Multiroom-Aufnahme	-3.778

Tabelle 3: Ergebnisse des PEAQ

Wie sich aus „Tabelle 3: Ergebnisse des PEAQ“ entnehmen lässt, bewertet der Algorithmus bereits den Vergleich zwischen Original und „Passthrough“ mit -2.852 bis -3.789 sehr schlecht. Subjektiv betrachtet eine solch schlechte Bewertung nicht nachvollziehbar, da zwar ein Verlust in der Audioqualität hörbar ist, dieser jedoch nicht so groß ist, wie es der ODG vermuten lassen würde. Die Ursache ist jedoch im direkten Vergleich der Graphen beider Tonaufzeichnungen schnell ersichtlich (siehe Abbildung 4). Eine der beiden Tonkanäle der Stereoaufnahme wird sehr stark zusammengestaucht.

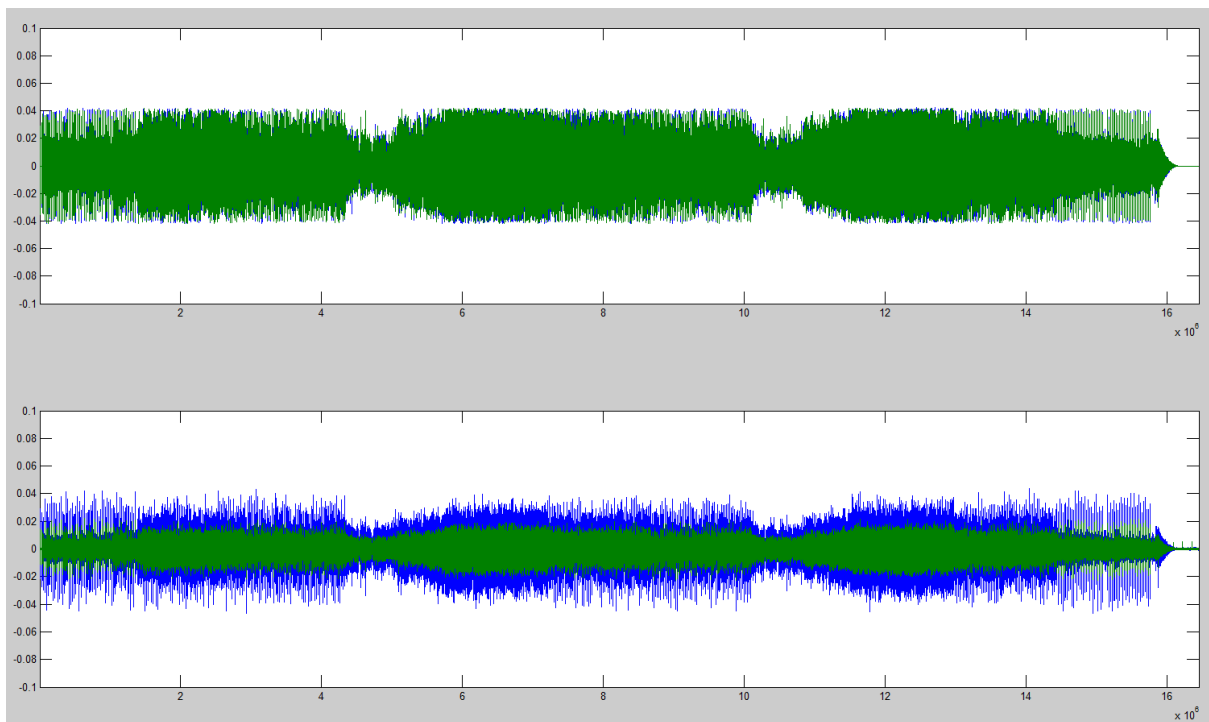


Abbildung 4: Vergleich Tonspuren Teststück 1 Original (oben) und Passthrough (unten)

Im nächsten Schritt wurden dann die Aufnahmen mit der „Passthrough“-Aufnahme verglichen, jedoch zeigen sich auch hier die Werte äußerst schlecht, obwohl die Graphen in diesem Fall zumindest optisch sehr ähnlich sind (siehe Abbildung 5).

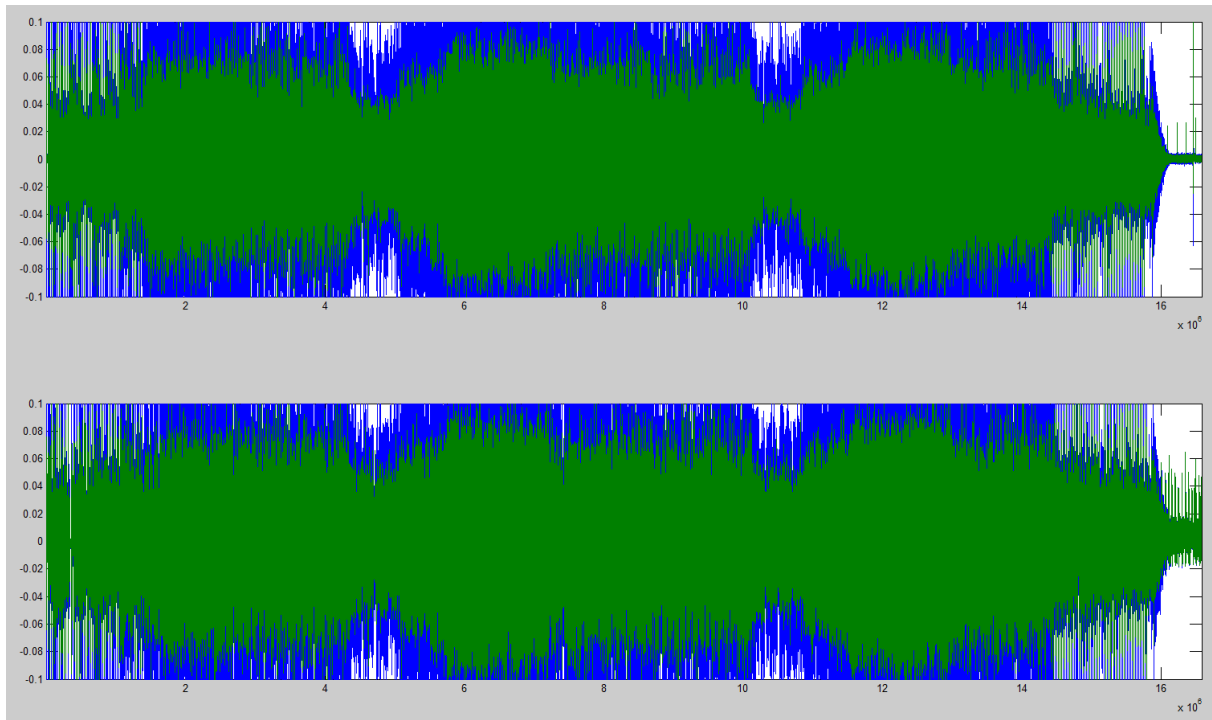


Abbildung 5: Vergleich Tonspuren Teststück 1 Passthrough (oben) und Multiroom-Aufnahme (unten)

Weitere Untersuchungen zeigen dann jedoch, dass im direkten Vergleich der Werte es Teilweise zu sehr großen Unterschieden kommt, bei denen Teilweise nicht einmal die Vorzeichen übereinstimmen (siehe Anhang Abbildung). Es wird vermutet, dass dies die Auswirkungen von kleinen Latenzschwankungen zwischen den Paketen ist. Die Schwankungen wären in dem Fall so klein, dass sie zwar kaum hörbar sind, jedoch doch zu einer Asynchronität der Werte führen. Dies wiederum würde die schlechten ODG-Werte erklären, da der Algorithmus wie zuvor erwähnt sehr anfällig für asynchrone Dateien ist. Diese Theorie konnte jedoch aufgrund der begrenzten Zeit des Projektes nicht weiter belegt werden.

Da als Folge daraus alle Ergebnisse des Algorithmus sehr schlecht ausfallen und diese so nur wenig Aussagekraft haben, wurde noch zusätzlich eine Subjektive Einschätzung der Audioqualität vorgenommen.

6.1.2. Subjektive Einschätzung³³

Subjektiv betrachtet ist der Qualitätsverlust durch die Hardware des LyraT hörbar, jedoch ist dies bei einem Preis von ca. 30€ nachvollziehbar und für diese Preisklasse ist die Qualität auch annehmbar. Das entwickelte Multiroom-System erzeugt ein deutlich hörbares Hintergrundrauschen. Hinzukommen Aussetzer durch verlorene und nicht übertragene Multicast-Pakete, die, wenn sie vermehrt auftreten, sehr störend wahrgenommen werden können. Die manchmal auftretenden ausfälle der Netzwerkschnittstelle (siehe 5.1.2) führen zu einem mehrsekündigen Ausfall der Übertragung. Das Auftreten und die Häufigkeit dieser Störungen ist nicht vorhersagbar. Solange die Übertragung stabil ist, ist die Tonqualität akzeptabel und im Vergleich zu einer direkten Tonweiterleitung innerhalb eines LyraT nicht viel schlechter.

6.2. Evaluation der Lokalisierung³⁴

Um die Genauigkeit der Lokalisierung zu bestimmen, wurde die Sendestärke bei bestimmten Entfernungen gemessen. Die gemessenen Sendestärken werden vom RSSI-Wert wiedergespiegelt.

³³ Geschrieben von Dennis Naujokat

³⁴ Geschrieben von Jannik Sengstack

RSSI wird in dBm angegeben. Wie diese Einheit vermuten lässt handelt es sich hierbei um einen Pegel, genauer einen Leistungspegel. Die Bezugsgröße ist hierbei kein beliebiger Wert, sondern ist die maximale Sendeleistung des Senders (hier: ESP).

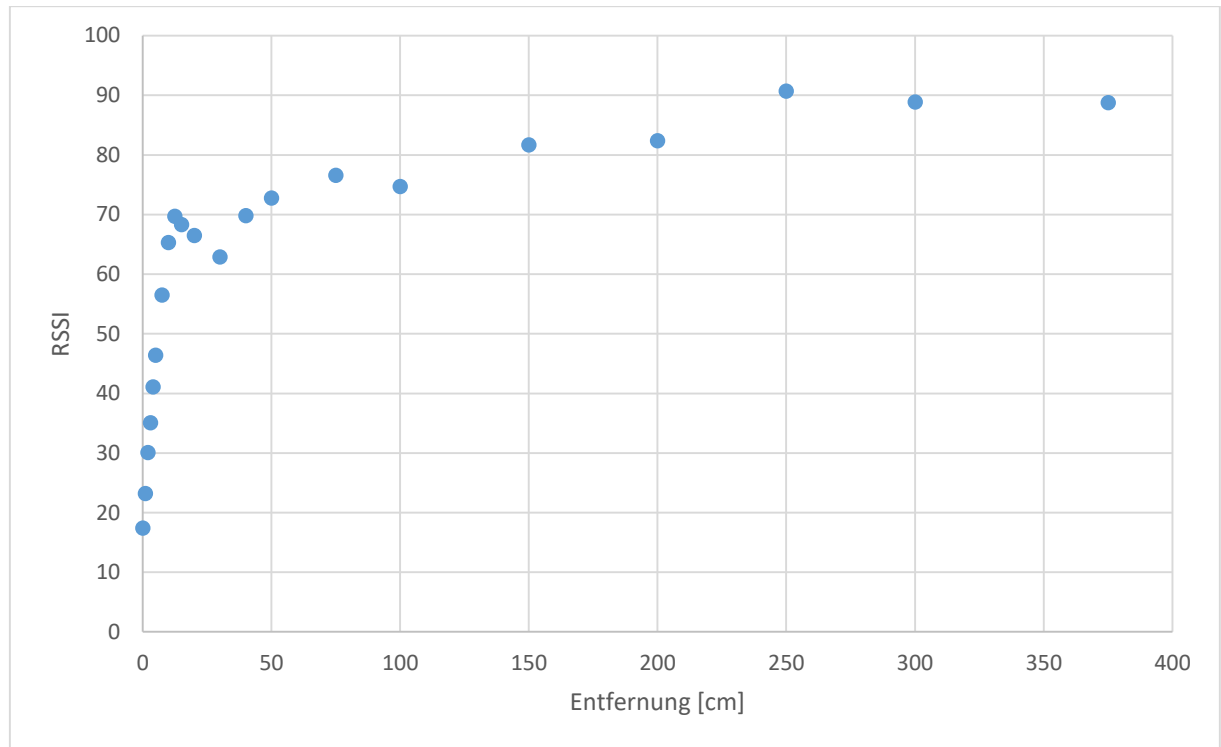


Abbildung 6: RSSI bei bestimmten Entfernungen

Die aufgenommene Kurve zeigt die Mittelwerte von mehreren Messungen der RSSI-Werte bei bestimmten Entfernungen. Die Besonderheit der Maßeinheit Dezibel ist, dass sie in einem Diagramm und in den Formeln zur Berechnung logarithmisch zur Basis 10 abgetragen wird. Bei einer logarithmischen Darstellung der Werte der x-Achse wird ein nahezu linearer Anstieg des RSSI-Wertes sichtbar.

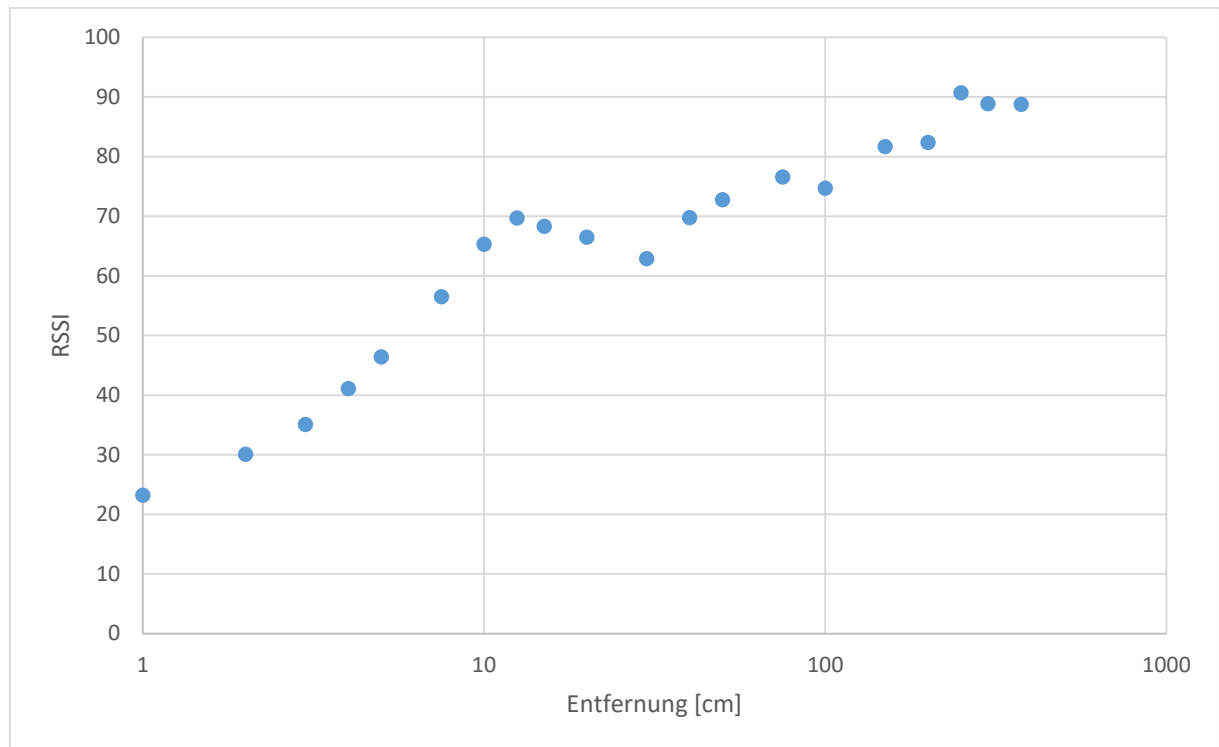


Abbildung 7: RSSI Verlauf bei logarithmischer Abtragung

Weiterhin ist zu beachten, dass die RSSI-Werte negativ geliefert werden. In den Abbildungen wurden das Vorzeichen gewechselt um eine vereinfachte Darstellung zu erhalten. Je größer der RSSI-Wert, desto größer auch die Entfernung zwischen Sender und Empfänger und je näher der RSSI-Wert an Null liegt, desto näher ist auch der Sender am Empfänger.

Abschließend hierzu lässt sich sagen, dass sich der Wahre Verlauf des RSSI-Wertes erst zeigt, wenn man ihn logarithmisch aufträgt. Dieser lineare Verlauf ist zu erwarten, denn wenn nur die Entfernung zwischen Sender und Empfänger vergrößert wird, sich sonst keine äußeren Bedingungen ändern, ist bei gleichbleibender Ausbreitung des Signals ein linearer Verlauf zu erwarten. Es treten vereinzelt Messungenauigkeiten durch zufällige Fehler auf, welche Vernachlässigt werden können.

Zur Verbesserung des RSSI-Wertes können Filter verwendet werden. Einer dieser Filter ist der Kalman Filter. Diesen Filter gibt es für drei Typen von Verläufen. Für lineare Systeme mit gleichbleibenden Komponenten und möglichst gleicher Störung des Signals wird der normale Kalman Filter benutzt. Der Erweiterte Kalman Filter (Extended Kalman Filter) wird bei nichtlinearen Systemen genutzt. Hierfür muss bei jeder Berechnung des gefilterten Wertes eine Jacobi-Matrix berechnet werden. Durch die Berechnung der Jacobi Matrix können Approximierte Werte errechnet werden, welche dann zum Filtern des RSSI-Wertes bei einer bestimmten Entfernung benutzt werden. Die dritte Variante des Kalman Filters, ist der Unscented Kalman Filter. Dieser wird auch für nichtlineare Systeme und deren Filterung benutzt, kommt aber im Gegensatz zum Erweiterten Kalman Filter ohne Berechnung der Jacobi-Matrix aus.³⁵ Wichtig sind diese Filter, wenn man mit Hilfe von mehreren ESP32 in einem Raum eine genaue Positionsbestimmung durchführen möchte. Dieses Verfahren heißt Trilateration und bestimmt die Position eines Objektes indem die Entfernung zu drei verschiedenen Ankerpunkten gemessen wird. Aus diesen Entfernungen kann eine Schnittmenge gebildet werden und so eine genaue Position bestimmt werden. Dieses Verfahren ist vor allem im

³⁵ BLE Indoor Localization based on Improved RSSI and Trilateration, (Mouhammad, et al., 2019)

Innenhausbereich Verbreitet und notwendig, da eine genaue Ortung via GPS innerhalb von Gebäuden nicht möglich ist.

7. Fazit³⁶

Zusammenfassend lässt sich sagen, dass das Projekt erfolgreich war. Es konnte eine automatische Steuerung des Multiroom-Systems realisiert werden und das System ist auch so konzipiert, dass es sich leicht mit anderen Multiroom-Systemen koppeln lässt (Siehe Anhang V.1).

Zu bemängeln ist jedoch manchmal die Reaktionszeit des Lokalisierungssystems. Es dauerte manchmal einige Sekunden bis das Fitness-Armband gefunden wurde. Dies kann daran liegen, dass dieses Armband nicht immer über BLE sichtbar ist und könnte mit einem anderen Gerät zur Nutzeridentifizierung besser funktionieren.

Des Weiteren ist die Audioqualität des erstellten Multiroom-Systems eher schlecht. Für einen Prototypen und zum Nachweis, dass der Ansatz funktioniert, waren die Ergebnisse ausreichend, wenn das System jedoch aktiv genutzt werden soll, müssen noch einige Verbesserungen vorgenommen werden. Zum einen sollte die Übertragung per Multicast durch einen besseren Streamingprotokoll ersetzt werden. Zudem sollten die Ursachen für das Hintergrundrauschen und für die unregelmäßig auftauchenden Speicherprobleme des LyraT genauer untersucht werden. Und für eine bessere Einschätzung des Qualitätsverlustes durch die Hardware des LyraT, könnte nochmals der PEAQ-Algorithmus mit vorheriger Trennung der Audiokanäle getestet werden.

Insgesamt ist das Ergebnis jedoch zufriedenstellend und zeigt, dass eine Automatisierung von Multiroom-Systemen zum automatischen Raumwechsel möglich ist und zukünftig vielleicht eine Sinnvolle Ergänzung zu diesen Systemen ist.

³⁶ Geschrieben von Dennis Naujokat

II. Literaturverzeichnis

- Arturo, D. L., 2017. *Said - Loli Patrol*. [Tonaufnahme].
- Dreaming Stargazers, S., 2019. *Lost Soul (feat. Selphius)*. [Tonaufnahme].
- Espressif Systems (Shanghai) CO., 2019. *Audio Development Framework - Espressif*. [Online]
Available at: <https://docs.espressif.com/projects/esp-adf/en/latest/get-started/index.html>
[Zugriff am 05 08 2020].
- Gupta, N. K., 2016. *Inside Bluetooth Low Energy*, s.l.: Artech House.
- IFAK, Ipsos & Research, G. M. a. C., 2019. *Bevölkerung in Deutschland nach Besitz ausgewählter Elektronikgeräte (Unterhaltungselektronik) im Haushalt in den Jahren 2016 bis 2019*. s.l.:VuMA.
- Kabal, P., 2003. *An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality*, s.l.: Department of Electrical & Computer Engineering, McGill University.
- Mouhammad, C. . S. et al., 2019. *BLE Indoor Localization based on Improved RSSI and Trilateration*, s.l.: IEEE.
- Rund, F. & Ulovek, K., 2018. *Comparison of two objective methods of quality assessment for digital audio broadcasting*. Prague, Czech Republic, IEEE.
- Teufel Blog Redaktion, kein Datum *Tausendsassa Multiroom – Wie WLAN-Audiosysteme private Haushalte erobern*. [Online]
Available at: <https://blog.teufel.de/definition-multiroom-audio-systeme/>
[Zugriff am 05 08 2020].
- Umweltbundesamt, 2011. *Anteil der Deutschen, die sich durch Lärm in der Nachbarschaft belästigt fühlen*. s.l.:Umweltbundesamt.
- Wagner, R., kein Datum *Ritt der Walküren*. [Tonaufnahme].

III. Abbildungsverzeichnis

Abbildung 1: Konzeptzeichnung.....	5
Abbildung 2: Prototypzeichnung.....	8
Abbildung 3: FHEM Benutzeroberfläche in Betrieb	14
Abbildung 4: Vergleich Tonspuren Teststück 1 Original (oben) und Passthrough (unten).....	17
Abbildung 5: Vergleich Tonspuren Teststück 1 Passthrough (oben) und Multiroom-Aufnahme (unten)	18

IV. Tabellenverzeichnis

Tabelle 1: Audiostreaming, MQTT Topics.....	9
Tabelle 2: Audiostreaming, Konfigurationsparameter	11
Tabelle 3: Ergebnisse des PEAQ	17

V. Anhang

V.1. Git

<https://github.com/PrinzDarknis/AudioHome>

V.2. Bedienungsanleitung AudioHome³⁷

Konfigurieren der ESPs zur Lokalisierung:

- Einstellen der WLAN-Zugangsdaten
 - WIFI_SSID
 - PASSWORD
- Einstellen des MQTT-Servers
 - SERVER
- Einstellen des Raumes der erfasst wird
 - ZIMMER

Konfiguration der LyraT:

- Einstellen der WLAN-Zugangsdaten
 - AUDIOHOME_WIFI_SSID
 - AUDIOHOME_WIFI_PASSWORD
- Benennen der Line-In- und Line-Out-Geräte
 - AUDIOHOME_LINE_IN_NAME
 - AUDIOHOME_LINE_OUT_NAME
 - Nur Buchstaben, Zahlen und Unterstrich verwenden
- Einstellen des MQTT-Servers
 - AUDIOHOME_MQTT_SERVER
- Einstellen der Multicastadresse
 - AUDIOHOME_MULTICAST_IPV4_ADDR
 - AUDIOHOME_UDP_PORT
 - Die Kombination aus Port und IP muss einmalig im System sein

Nutzen anderen Multiroom-Systeme:

Erstellen sie einen Adapter der Folgende MQTT-Nachrichten versenden bzw. verarbeiten kann:

- Topic: AudioDevice/Line-In/<Eingabegerät Name>/Address
 - Senden:
 - Beim Starten die Adresse über die das Audiosignal empfangen werden kann
- Topic: AudioDevice/Line-In/<Eingabegerät Name>/plays
 - Senden:
 - „play“ wenn eine Wiedergabe am Gerät gestartet wird
 - „none“ wenn die Wiedergabe am Gerät gestoppt wird
 - Empfangen:
 - „play“ wenn die Begonnen werden soll einen Audiostream zu senden
 - „none“ wenn der Audiostream beendet werden soll
- Topic: AudioDevice/Line-Out/<Ausgabegerät Name>
 - Empfangen:
 - Adresse die Abgespielt werden soll

³⁷ Geschrieben von Dennis Naujokat

Da das Format der Adresse nicht vorgeschrieben ist, können beliebige Informationen die zur Koordinierung des Multiroom-Systems nötig sind Versand werden.

Installation:

- Kopieren der „99_AudioHomeUtils.pm“ in <FHEM-Verzeichnis>/FHEM
- Neustarten von FHEM
- Einmaliges initialisieren des Systems durch Ausführung des Perlbefehls „AudioHome_init()“
 - Bsp: define AudioHome_ starter at +00:00:00 { AudioHome_init ()}
- Einrichten des MQTT-Servers im Device „AudioHome_Manager“
 - Bsp: set AudioHome_Manager MQTT-IP 127.0.0.1:1883

Anlegen eines Nutzers:

- Befehl: set AudioHome_Manager DefineUser **BENUTZERNAME**
- Dem Nutzer im Nutzerdevice eine Bluetooth-MAC-Adresse zuweisen
 - Bsp: set AudioHome_user_**BENUTZERNAME** BT-MAC 00:00:00:00:00:01

Audiogeräten Räume zuweisen:

- Die Line-In- und Line-Out-Geräte werden automatisch erkannt und angelegt
- Zuweisen des **Readings** „room“ an die Devices
 - Bsp: setreading AudioHome_LineIn_**GERÄTENAME** room **RAUMNAME**
 - Bsp: set AudioHome_LineOut_**GERÄTENAME** room **RAUMNAME**
- Der jeweilige Befehl wird auch bei Geräten ohne Raum im Interface angezeigt.

Auswahl/Ändern/Stoppen einer Wiedergabe über die Benutzeroberfläche:

- Über die das Dropdown-Menü des Nutzers das gewünschte Line-In-Gerät auswählen das Wiedergeben werden soll.
- Zum Stoppen der Wiedergabe **none** auswählen.

V.3. Evaluation

```
Vorzeichenunterschied: index: 79127 , ori: -6.683350e-03 , auf: 8.575439e-03
Vorzeichenunterschied: index: 79128 , ori: -1.885986e-02 , auf: 8.911133e-03
Vorzeichenunterschied: index: 79128 , ori: -5.889893e-03 , auf: 8.026123e-03
Vorzeichenunterschied: index: 79129 , ori: -1.611328e-02 , auf: 1.342773e-02
Vorzeichenunterschied: index: 79129 , ori: -4.638672e-03 , auf: 8.331299e-03
Vorzeichenunterschied: index: 79130 , ori: -1.284790e-02 , auf: 6.866455e-03
Vorzeichenunterschied: index: 79130 , ori: -3.540039e-03 , auf: 8.483887e-03
Vorzeichenunterschied: index: 79131 , ori: -3.997803e-03 , auf: 8.056641e-03
Vorzeichenunterschied: index: 79132 , ori: -6.591797e-03 , auf: 7.934570e-03
Vorzeichenunterschied: index: 79133 , ori: -2.459717e-02 , auf: 1.159668e-03
Vorzeichenunterschied: index: 79133 , ori: -8.819580e-03 , auf: 7.873535e-03
Vorzeichenunterschied: index: 79134 , ori: -2.407837e-02 , auf: 3.051758e-05
Vorzeichenunterschied: index: 79134 , ori: -8.941650e-03 , auf: 7.598877e-03
Vorzeichenunterschied: index: 79135 , ori: -7.751465e-03 , auf: 7.476807e-03
Vorzeichenunterschied: index: 79136 , ori: -6.896973e-03 , auf: 7.629395e-03
Vorzeichenunterschied: index: 79137 , ori: -1.184082e-02 , auf: 1.190186e-03
Vorzeichenunterschied: index: 79137 , ori: -7.659912e-03 , auf: 5.950928e-03
Vorzeichenunterschied: index: 79138 , ori: -9.216309e-03 , auf: 2.685547e-03
Vorzeichenunterschied: index: 79139 , ori: -9.246826e-03 , auf: 1.342773e-03
Vorzeichenunterschied: index: 79140 , ori: -7.293701e-03 , auf: 3.234863e-03
Vorzeichenunterschied: index: 79141 , ori: -5.981445e-03 , auf: 6.591797e-03
Vorzeichenunterschied: index: 79142 , ori: -1.351929e-02 , auf: 2.746582e-04
Vorzeichenunterschied: index: 79142 , ori: -7.812500e-03 , auf: 8.666992e-03
Vorzeichenunterschied: index: 79143 , ori: -1.193237e-02 , auf: 7.110596e-03
Vorzeichenunterschied: index: 79144 , ori: -1.473999e-02 , auf: 2.502441e-03
Vorzeichenunterschied: index: 79146 , ori: -1.190186e-02 , auf: 1.525879e-04
Vorzeichenunterschied: index: 79147 , ori: -9.429932e-03 , auf: 3.692627e-03
Vorzeichenunterschied: index: 79148 , ori: -1.358032e-02 , auf: 5.493164e-04
```

Abbildung 8: Ausschnitt Direktvergleich Passthrough (ori) und Multiroom-Aufnahme (auf)