

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-
ICT

Masterthesis

Ontwikkeling van een meerkanaals-, hogeresolutie-USB-naar-I2S-converter

Bram Plessers

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. ir. Nele MENTENS

PROMOTOR :

ir. Stijn VAN MOL

BEGELEIDER :

ing. Jelle BIESMANS

COPROMOTOR :

Dhr. Jo DELVEAU

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven



Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt



2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-
ICT

Masterthesis

Ontwikkeling van een meerkanaals-, hogeresolutie-USB-naar-I2S-converter

Bram Plessers

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

Prof. dr. ir. Nele MENTENS

PROMOTOR :

ir. Stijn VAN MOL

COPROMOTOR :

Dhr. Jo DELVEAU

BEGELEIDER :

ing. Jelle BIESMANS



KU LEUVEN

Woord vooraf

Door het afronden van deze masterproef eindigt ook mijn opleiding tot Industrieel Ingenieur in de Elektronica-ICT aan Universiteit Hasselt en KU Leuven. Gedurende de opleiding heb ik genoten van de vele interessante vakken waarbij ik veel kennis heb opgedaan. Ondanks dat deze periode niet altijd even makkelijk was, ben ik zeer tevreden met mijn behaalde prestaties. Deze prestaties zijn ook te danken aan het eerst behalen van een Professionele Bachelor Elektronica-ICT aan de PXL. Daar heb ik de praktische aspecten van Elektronica-ICT kunnen ervaren en uitvoeren terwijl dat de ingenieursopleiding mij meer theoretisch inzicht heeft gegeven. In mijn ogen de ideale combinatie. Voornamelijk wil ik PVS bedanken om mij de mogelijkheid te bieden om dit onderwerp te mogen uitvoeren als mijn masterproef en de ondersteuning in de weg daar naartoe.

Gedurende mijn masterproef bij PVS heb ik mij steeds toegewijd ingezet op het uitvoeren van het onderzoek met een zelfstandige en productieve houding met de gewenste ondersteuning wanneer nodig. Ik blik aangenaam terug op de deze periode waarbij ik mee heb mogen draaien in deze familiere bedrijfsomgeving.

Verschillende mensen hebben een grote bijdrage geleverd aan het goede verloop van deze masterproef. Deze mensen wil ik in het bijzonder bedanken.

Allereerst wil ik graag mijn externe promotor, ir. Stijn Van Mol, en externe co-promotor, dhr. Jo Delveau, willen bedanken. Door hun goede begeleiding, hulp, nalezen van de scriptie en open communicatie ben ik gekomen tot dit mooie resultaat. Verder wil ik ook de interne promotor, prof. dr. ir. Nele Mentens, en interne begeleider, ing. Jelle Biesmans, bedanken voor hun begeleiding, controleren van de scriptie en belangrijke inzichten binnen de masterproef. Eveneens zou ik graag dr. Bart Dreesen willen bedanken voor zijn contributie op het aspect van taal en communicatie. Verder zou ik het hele team van PVS nog willen bedanken voor de leerrijke omgeving waarin ik ondersteuning, leuke conversaties, tips en inzichten gekregen heb.

Tenslotte bedank ik nog graag mijn familie, vriendin en vrienden voor de jarenlange ondersteuning doorheen mijn reis in het onderwijs. Door jullie steun en geduld heb ik de mogelijkheid gehad om mijn opleiding tot een succesvol resultaat te volbrengen.

Bram Plessers

Oudsbergen, 20 januari 2023

Inhoudsopgave

Woord vooraf.....	1
Lijst van tabellen.....	5
Lijst van figuren.....	7
Verklarende woordenlijst.....	9
Abstract.....	11
Abstract in English.....	13
1. Inleiding	15
1.1. Situering	15
1.2. Probleemstelling.....	15
1.3. Onderzoeksvraag.....	16
1.4. Doelstellingen.....	16
1.5. Materiaal en methode	16
2. Literatuurstudie	17
2.1. Audio.....	17
2.1.1. Parameters.....	17
2.2. Universele Seriële Bus	21
2.2.1. Versies.....	21
2.2.2. Vormfactor	21
2.2.3. Apparaat klasse	23
2.2.4. Synchronisatie.....	24
2.3. I ² S	26
2.4. I ² C.....	27
2.5. USB naar I ² S.....	27
2.6. FPGA gebaseerde audio DSP	27
2.7. Microcontroller gebaseerde audio DSP	29
2.8. Raspberry Pi Pico (RP2040).....	30
2.8.1. Programmable Input/Output	31
2.9. Dante	33
2.9.1. Ultimo	33
2.9.2. Dante Controller.....	34
2.9.3. Dante Virtual Soundcard.....	35
2.9.4. Precision Time Protocol.....	35

3. Ontwikkeling	37
3.1. Hardware	37
3.1.1. Vereisten	37
3.1.2. Vergelijking	37
3.1.3. Keuze	40
3.1.4. Opstelling	40
3.1.5. Fysieke testopstelling	42
3.1.6. Elektronisch ontwerp	43
3.1.7. PCB	45
3.2. Software	49
3.2.1. Overzicht	50
3.2.2. USB	50
3.2.3. I ² S	52
4. Resultaten	55
4.1. PCB/Wandmodule	55
4.2. Windows	56
4.3. Data	57
4.4. Metingen	58
4.4.1. Configuratie	58
4.4.2. Testdata	60
4.4.3. Opgenomen grafiek	60
4.4.4. REW-analyse grafieken	61
4.5. Langdurige test	63
5. Conclusie	65
5.1. Toekomstig werk	65
Referentielijst	67
Bijlagen	73
Bijlage A: Samenvatting vergelijking mogelijke microcontrollers	75
Bijlage B: Wandmodule PCB-stack	77
Bijlage C: Topschema uit Altium Designer	79
Bijlage D: USB naar I2S schema in Altium Designer	81
Bijlage E: PCB-lagen	83
Bijlage F: BOM NWP400 USB naar I2S	85
Bijlage G: PIO-flowchart	87
Bijlage H: REW-metingen	89

Lijst van tabellen

Tabel 1: USB Standaarden en snelheden	21
Tabel 2: Pinout connectoren USB 1.0-2.0	22
Tabel 3: Pinout connector USB 3.0-3.1 (type A, type B en micro B)	22
Tabel 4: Pinout connector USB 3.0-3.1 (type C).....	23
Tabel 5: USB Audio Synchronisatie types	24
Tabel 6: Verbruik van ADC 1.0, ADC 2.0 en ADC 3.0	28

Lijst van figuren

Figuur 1: Hoofdbedrijf PVS met dochterondernemingen Audac, PROCAB en Caymon	15
Figuur 2: Audac NWP400 USB/Bluetooth Wandmodule	16
Figuur 3: ADC-principe	18
Figuur 4: Oscillogram voorbeeld lage sample rate	18
Figuur 5: Verschil in bit depth bij dezelfde sample rate	19
Figuur 6: Ruis bij een analoog en digitaal signaal	19
Figuur 7: Sinusgolf van 60 Hz (groen) met een harmonische van 180 Hz (rood) resulterend in de zwarte golf	20
Figuur 8: Frequentierespons	20
Figuur 9: Vormfactor USB-connector	21
Figuur 10: USB Audio operationeel model	23
Figuur 11: USB synchrone synchronisatie	25
Figuur 12: USB adaptieve synchronisatie	25
Figuur 13: USB asynchrone synchronisatie	26
Figuur 14: Eenvoudige I ² S systeem configuratie en interface timing	26
Figuur 15: I ² C toepassingsvoorbeeld	27
Figuur 16: DE10-Nano Opstelling	28
Figuur 17: Hardware Architectuur	28
Figuur 18: Sequentie diagram van het audiosysteem	29
Figuur 19: Raspberry Pi Pico Ontwikkelaarsbord Pinout	30
Figuur 20: Raspberry Pi Pico probe verbindingsdiagram	31
Figuur 21: PIO-state machine blok	31
Figuur 22: Dante Ultimo Network, Audio en Control Interfaces	33
Figuur 23: Dante Controller Applicatie Interface	34
Figuur 24: Dante Virtual Soundcard Applicatie Interface	35
Figuur 25: XMOS xCORE XU208 PCB-ontwerp	38
Figuur 26: STM32F4DISCOVERY	38
Figuur 27: PIC32 USB-audiomogelijkheden	39
Figuur 28: ESP32-S2 DevKitM-1	39
Figuur 29: RP2040 in QFN-56 package	39
Figuur 30: DIP Switches S1 CODEC Isolatie	40
Figuur 31: Testopstelling Raspberry Pi Pico en Dante Ultimo interne klokken	41
Figuur 32: Testopstelling Raspberry Pi Pico en Dante Ultimo externe klokken	41
Figuur 33: Testopstelling Raspberry Pi Pico en Dante Ultimo PDK	42
Figuur 34: Schematische voorstelling gehele hardware testopstelling	42
Figuur 35: Verschillende aanzichten Audac NWP400 USB/Bluetooth wandmodule	43
Figuur 36: Opstelling spanningsdeler	44
Figuur 37: PCB stackup	45
Figuur 38: Eurocircuits PCB bestellen overzicht	46
Figuur 39: Stencil bovenkant USB-naar-I ² S PCB	47
Figuur 40: Soldeerpasta op soldeer masker van USB-naar-I ² S PCB	47
Figuur 41: Opstelling aanbrengen soldeerpasta	47
Figuur 42: Componenten op soldeerpasta geplaatst	48
Figuur 43: Vapor phase soldeeroven temperatuur profiel	48
Figuur 44: Soldeer temperatuur profiel en vapor phase soldeeroven temperatuur profiel	48
Figuur 45: Imdes Jumbo Condens IT	49
Figuur 46: Close-up oscillator	49
Figuur 47: Close-up RP2040	49
Figuur 48: NWP400 firmware bestanden	49
Figuur 49: USB Descriptors NWP400 in Windows apparaatbeheer	50
Figuur 50: Circulaire buffer	51

Figuur 51: I ² S-lijnen op oscilloscoop.....	52
Figuur 52: I ² S-lijnen op oscilloscoop groter bereik	52
Figuur 53: Resultaat PCB USB naar I ² S.....	55
Figuur 54: Resultaat NWP400 USB/Bluetooth wandmodule.....	55
Figuur 55: Apparaatbeheer Audac High Definition Audio.....	56
Figuur 56: Apparaatbeheer Speakers	56
Figuur 57: Apparaatbeheer Microfoon	56
Figuur 58: Resultaat I ² S datalijnen	57
Figuur 59: Resultaat I ² S datalijnen groter bereik.....	57
Figuur 60: Originele kristaloscillator XOSC	57
Figuur 61: Master clock afkomstig van Ultimo chip	57
Figuur 62: Dante Controller configuratie REW.....	58
Figuur 63: Voorkeuringstellingen testapparatuur	59
Figuur 64: Testmeting instellingen testapparatuur.....	59
Figuur 65: Frequentieanalyse meting REW.....	59
Figuur 66: Opname Audacity meting REW.....	60
Figuur 67: Opgenomen grafiek REW	60
Figuur 68: Afspeelde grafiek REW.....	61
Figuur 69: Opgenomen grafiek REW	61
Figuur 70: Speaker opname van 1 kHz gedurende een uur	64
Figuur 71: Vergelijking opgenomen lied en origineel lied	64
Figuur 72: Microfoon opname van een 100 Hz sinusgolf gedurende twee uur	64

Verklarende woordenlijst

ADC	Analog Digital Converter
ANC	Active Noise Control
API	Application Programming Interface
ASRC	Asynchronous Sample Rate Converter
BOM	Bill-Of-Materials
CODEC	Coderen/Decoderen
DAC	Digital Analog Converter
DMA	Direct Memory Access
DSP	Digital Signal Processing
EOF	End Of Frame
FIFO	First In, First Out
FPGA	Field-Programmable Gate Array
HID	Human Interface Device
IC	Inter-Integrated Circuit
I²S	Inter-IC Sound
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IRQ	Interrupt Request
ISR	Input Shift Register
KVM	Keyboard, Video en Mouse
LPM	Link Power Management
NRND	Not Recommended for New Designs
OpenOCD	Open On-Chip Debugger
OSR	Output Shift Register
PBGA	Plastic Ball Grid Array
PCB	Printed Circuit Board
PDK	Platform Development Kit
PHY	Physical Layer Interface
PIO	Programmable Input Output
PoE	Power over Ethernet
PTP	Precision Time Protocol
QFN	Quad-Flat No-Leads
QSPI	Quad Serial Peripheral Interface
R&D	Research and Development
RMII	Reduced Media-Independent Interface
ROM	Read-only memory
RX	Receive data
SA	Standards Association
SBU	Side Band Use
SNR	Signal-to-noise Ratio
SOF	Start Of Frame
SPL	Sound Pressure Level
SWD	Serial Wire Debug
THD	Total Harmonic Distorstion
TQFP	Thin Quad Flat Package
TSSOP	Thin Shrink Small Outline Package
TX	Transmit data
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
XIP	eXecute In Place

Abstract

In het uitgebreide assortiment van de Audac-audioproducten is er een wandmodule met de mogelijkheid om audio af te spelen door middel van een USB-of Bluetoothverbinding. Deze masterproef onderzoekt de mogelijkheid om de chip, die de omzetting van USB naar I²S uitvoert, te vervangen door een alternatief die deze functie uitvoert met een hoger aantal bits per sample, data verstuurt in twee richtingen en het asynchroon-isochroon data USB-transfer type gebruikt. Een alternatief wordt gezocht omwille van de lage beschikbaarheid en de stopzetting van ondersteuning en fabricage door de huidige fabrikant.

In eerste instantie wordt er onderzoek uitgevoerd naar de technologieën zelf aan de hand van een literatuurstudie. Vervolgens wordt er een proefopstelling ontworpen waarbij er zowel van een interne klok als een externe klok gebruik gemaakt kan worden voor het ontwerpen van firmware. Het gebruik van de externe Dante-klok voor de opstelling zorgt ervoor dat alle klokken synchroon lopen en de noodzaak van een Asynchronous Sample Rate Converter (ASRC) wegvalt. Daarna wordt op basis van deze proefopstelling een printplaat ontworpen die voldoet aan de eisen van de printplaat die ingebouwd kan worden in een wandpaneel.

Tenslotte wordt aan de hand van kwaliteit testen geëvalueerd of het ontwerp en de firmware voldoen aan de vooropgestelde audiokwaliteit en dusdanig voldoen aan de hoge kwaliteitsnormen. Er kan geconcludeerd worden dat de Raspberry Pi RP2040 een volwaardige vervanger is voor de gedateerde chip.

Abstract in English

The extensive range of Audac audio products includes a wall module which enables the user to play audio through a USB or Bluetooth connection. This master's thesis explores the possibility of replacing the Integrated Circuit (IC) that performs the conversion from USB to I²S with an alternative to perform this function with higher bit depth, send data in two directions and use the asynchronous isochronous data USB transfer type. An alternative to the IC was sought due its low availability and because of the discontinued support from the current manufacturer.

The first step is to research the technologies themselves on the basis of a literature study. Next, a test setup is designed in which both an internal clock and an external Dante clock can be used to design firmware. Using the Dante clock for the whole setup ensures that all clocks are synchronised and eliminates the need for an Asynchronous Sample Rate Converter (ASRC). Then, based on this test setup, a circuit board is designed to meet the requirements to be integrated in the existing wall module PCB.

Finally, evaluations are performed on the PCB, using audio quality testing to evaluate whether the design and firmware meet the predetermined audio quality and as such meet the high-quality standards. It can be concluded that the Raspberry Pi RP2040 is a viable alternative for the outdated IC.

1. Inleiding

Dit inleidend hoofdstuk van de scriptie is gewijd aan de onderzoeksopzet. In dit hoofdstuk wordt allereerst de situering van de thesis geschetst. Ten tweede wordt de concrete probleemstelling beschreven. Vervolgens wordt de onderzoeksvraag, waar deze thesis antwoord op biedt, concreet besproken. Daarnaast worden de doelstellingen besproken waaraan deze thesis opgericht werd. Ten slotte worden het materiaal en de methodes besproken die gedurende deze thesis gebruikt werden.

1.1. Situering

Het onderzoek dat uitgevoerd wordt in het kader van de masterproef, gebeurt voor het bedrijf PVS, gelegen te Stevoort. PVS is een internationale speler op de markt van professionele audiosystemen en heeft drie verschillende merken met hun eigen specialiteiten. Het eerste merk, genaamd “Audac”, verzorgt alles wat betrekking heeft tot geluidsinstallaties voor zowel particulieren, alsook voor professionele doeleinden [1]. “PROCAB” is het tweede merk dat valt onder PVS en dit bedrijf voorziet alle soorten bekabeling die nodig zijn voor audiovisuele, IT en mobiele intranetoplossingen [2]. Het derde en laatste merk, genaamd “Caymon”, voorziet server racks, modulaire oplossingen voor systeem integraties, flightcases, etc. [3]. Figuur 1 toont de logo's van deze bedrijven met als hoofdbedrijf PVS en dochterondernemingen Audac, PROCAB en Caymon.



Figuur 1: Hoofdbedrijf PVS met dochterondernemingen Audac, PROCAB en Caymon

De afdeling waar dit onderzoek werd uitgevoerd was de R&D-afdeling van het merk Audac. Het meest indrukwekkende aan de producten van Audac is dat al deze producten binnenshuis worden ontwikkeld variërend van luidsprekers, versterkers, audioprocessors, wandmodules, etc. Omwille van deze binnenshuis ontwikkeling, zijn alle oplossingen dicht geïntegreerd zorgend voor een eenvoudig, samenwerkend geheel aangevuld met een persoonlijke en gespecialiseerde klantenservice.

1.2. Probleemstelling

Een audio-installatie bestaat of kan bestaan uit verschillende apparatuur zoals luidsprekers, versterkers, audioprocessors, wandmodules, etc. De probleemstelling bevindt zich in dit onderzoek bij de wandmodule die beschikt over Bluetooth en een USB Type C-interface. Via een van deze interfaces kan er verbinding met een toestel tot stand gebracht worden waardoor er vervolgens audio van het apparaat naar de installatie afgespeeld kan worden. De omzetting voor de USB-interface naar I²S werd uitgevoerd door een IC die de seriële USB-data omzette naar I²S seriële data. Deze IC werd echter NRND waardoor het niet langer ondersteund wordt vanuit de fabrikant om gebruik makend van de IC nieuwe ontwerpen uit te werken. Figuur 2 toont de voorkant van de Audac NWP400 wandmodule die ondersteuning biedt voor zowel Bluetooth als USB-verbinding functionaliteit. Op deze afbeelding is de USB Type C-verbinding te zien alsook het logo van Bluetooth. Een Bluetooth verbinding kan tot stand gebracht worden door de knop in te drukken en vervolgens te verbinden met een Bluetooth ondersteunend apparaat.

Deze IC had ook een bit depth van 16-bits met een sample rate van 32, 44.1 en 48 kHz in slechts één richting. Deze IC maakte ook gebruik van adaptief isochroon transfer type. De vereisten voor audio binnen de eigen doelstellingen van Audac liggen hedendaags anders bij 24-bits bit depth met een sample rate van 48 kHz waarbij de seriële data ook bi-directioneel gestuurd kan worden.



Figuur 2: Audac NWP400 USB/Bluetooth Wandmodule

1.3. Onderzoeksvraag

Aangezien de USB naar I²S ICs niet meer ondersteund worden in de toekomst en de audio eigenschappen van de wandmodule aan hogere vereisten moeten voldoen kwam de vraag uit de R&D-afdeling van Audac om een nieuwe module te ontwikkelen die deze omvorming van het USB-interface naar het I²S interface uitvoert aan de vooropgestelde parameters.

Dit onderzoek focust op de mogelijkheid om een meerkanaals bi-directioneel USB-audiosignaal om te zetten naar het I²S interface voor high definition audio met een sample rate van 48 kHz en een bit depth van 24-bits per kanaal zonder extra installatie of ontwikkeling van gepersonaliseerde drivers op eender welk besturingssysteem.

1.4. Doelstellingen

Deze masterproef beoogt een PCB-module te ontwerpen die geïntegreerd kan worden in de bestaande Audac USB/Bluetooth wandmodule stacked PCB's waarbij de bi-directionele seriële USB-data vertaald wordt naar de seriële I²S interface. Deze vertaling wordt voorzien door een low-level processor die efficiënt gebruikt maakt van de beschikbare middelen.

1.5. Materiaal en methode

Bovenstaande probleemstellingen en doelstellingen zijn benaderd vanuit zowel een praktisch als theoretisch perspectief. Over het algemeen is deze masterproef praktisch van aard.

Voor het programmeren van de microcontroller werd er gebruik gemaakt van Visual Studio Code met de C/C++ extension om de leesbaarheid van C en C++ code te verbeteren, alsook de OpenOCD voor het debuggen van de code. Dankzij deze OpenOCD is het mogelijk om stap voor stap doorheen de code te doorlopen. Aanvullend op deze configuratie werd er gebruik gemaakt van Putty om de seriële UART debug informatie uit te lezen aan de hand van de Picoprobe opstelling. Om de code te compileren werd gebruik gemaakt van GNU ARM C/C++ Compiler.

De configuratie van de Dante Ultimo PDK werd uitgevoerd aan de hand van de Dante Controller waarbij alle parameters en configuraties ingesteld kunnen worden. Aanvullend hieraan is een installatie van Dante Virtual Soundcard op de laptop waarmee een Dante zender of ontvanger gemuleerd kan worden op Windows of MacOS. Voor het testen van de doorvoer van audiodata werd er gebruik gemaakt van Audacity voor het generen en opnemen van audiodata alsook het programma genaamd REW voor het uitvoeren van een frequentie analyse.

Het ontwerpen van het PCB werd uitgevoerd in Altium Designer voor zowel de elektronische schakeling als de printplaat zelf. De componenten gebruikt op het PCB zijn componenten uit de collectieve PVS-bibliotheek.

2. Literatuurstudie

Vooraleer het onderzoek van start kon gaan was het belangrijk om een grondig en uitgebreid vooronderzoek uit te voeren in de vorm van een literatuurstudie. Zodanig is het mogelijk een goed beeld te vormen van de huidige stand van zaken in het domein van het onderzoek.

De doelstelling van de literatuurstudie is om concreet af te bakenen wat er juist verstaan wordt bij bepaalde onderwerpen en uit welke parameters de kwaliteit van een systeem, onderwerp, interface, en toepassing concreet bepaald kunnen worden.

Allereerst werd er onderzocht wat juist verstaan wordt onder de term audio en welke parameters allemaal bepalen wat de kwaliteit van de audio is. Enkele van deze parameters zijn kanalen, sample rate, etc. Ten tweede werd het serieel protocol van USB onderzocht en de verschillende eigenschappen die de fysieke hardware bepalen en de synchronisatie tussen host en apparaat. Vervolgens worden de bus systemen ontworpen door Philips onderzocht, gebaseerd op protocollen genaamd I²S en I²C. Waarbij dat I²S het elektronisch serieel bus systeem voor audio is en I²C het elektronisch serieel bus systeem voor multi-controller voor randapparatuur is. Daarna werd er onderzocht wat er juist moet gebeuren voor de omzetting van USB-data naar I²S data uit te voeren. Daarnaast werd er ook onderzocht welke literatuur er beschikbaar was die betrekking had op de verschillende aspecten van dit onderzoek. Zowel voor toepassingen met een FPGA als een microcontroller. Verder worden de eigenschappen van de Raspberry Pi Pico microcontroller besproken alsook de functie van PIO bij deze specifieke microcontroller. Tenslotte wordt de Dante Ultimo microcontroller besproken en wat het Dante protocol juist inhoudt. Alsook het Precision Time Protocol dat gebruikt wordt om klokken te synchroniseren bij een Dante netwerk.

2.1. Audio

Het eerste onderdeel dat grondig bestudeerd werd voor dit onderzoek is audio of audiosignaal. De oorsprong van het woord *audio* is uit het Latijns. Audio is hierbij afgeleid uit het werkwoord *audire* die vertaalt naar ‘vernemen, verstaan of horen’. De betekenis van audio of audiosignaal is een signaal dat informatie bevat over geluid. Dit audiosignaal is een elektrisch signaal dat zowel digitaal als analoog kan zijn.

“Geluidsdruk en elektrische wisselspanning spanning zijn ‘continu variërende’ grootheden. Dit betekent dat deze grootheden niet ‘springen’ van de ene naar de andere waarde, maar alle tussengelegen waarden ‘aannemen’” [4]. Om over de mogelijkheid te beschikken de analoge signalen op te slaan op een digitaal medium moet dit analoog signaal omgezet worden naar een digitaal signaal dat geïnterpreteerd kan worden door een computer. De omzetting gebeurt aan de hand van een ADC waarbij de analoge waardes omgezet worden naar getallen uit het binaire talstelsel [4].

2.1.1. Parameters

Bepalen hoe de kwaliteit van geluid is kan niet enkel subjectief gebeuren maar ook objectief. Een objectieve vaststelling gebeurt dan aan de hand van enkele parameters. Door het bestuderen van de parameters kan er bepaald worden of de geluidskwaliteit al dan niet voldoende is voor de toepassing.

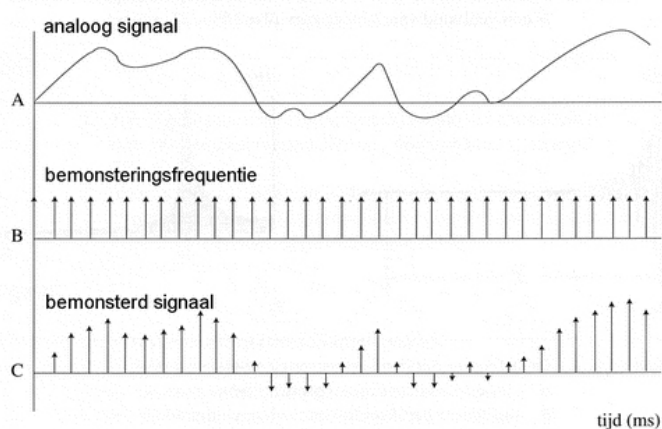
2.1.1.1. Kanalen

Een eerste parameter genaamd kanalen beschrijft het aantal individuele bronnen van audio die er aanwezig zijn. Voor een mono speaker opstelling wordt er slechts gebruik gemaakt van een enkele audiobron. Wanneer men dan gebruik maakt van een stereo speaker opstelling zijn er twee audiobronnen aanwezig. Wanneer men spreekt over meer dan twee kanalen wordt surround sound bedoeld waarvan talrijke verschillende opstellingen mogelijk zijn. Een voorbeeld hiervan is het 7.1 surround sound formaat dat zeer populair is voor thuiscinema en Blu-Rays.

2.1.1.2. Sample rate

Een volgende belangrijke parameter voor het omzetten van een analoog audiosignaal naar een digitaal audiosignaal is de sample rate. De grafieken in Figuur 3 tonen hoe dit ADC-principe juist werkt “In deelfiguur A is het analoge signaal afgebeeld. De pijltjes in deelfiguur B geven aan met welk ritme

(sample rate) de monsters worden genomen. In deelfiguur C zien we discrete waarden van de analoge golfvorm op de achtereenvolgende tijdstippen. De hoogtes van de pijltjes worden omgezet in binaire getallen en opgeslagen” [4]. Hoe hoger de densiteit van de pijltjes, hoe hoger de sample rate is en bijgevolg de benadering van het oorspronkelijk signaal beter is [4], [5].

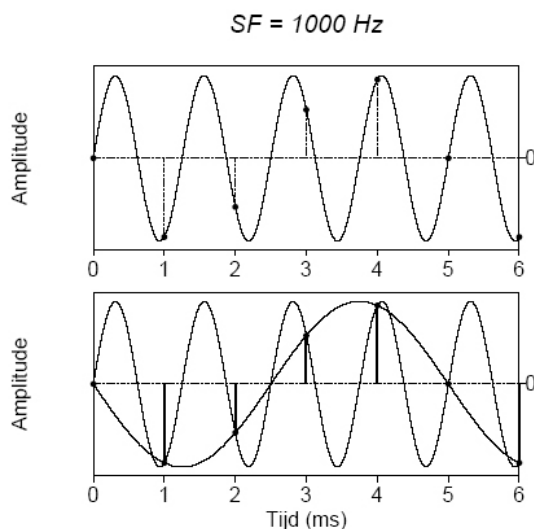


Figuur 3: ADC-principe [4]

Wanneer men een te lage sample rate gebruikt kan dit een zeer vertekend beeld geven van het origineel signaal. De regel die hiervoor wordt toegepast werd oorspronkelijk onderzocht door Harry Nyquist en vervolgens door Claude E. Shannon en is beter bekend als het Bemonsteringstheorema van Nyquist-Shannon. Waarbij het onderzoek van Nyquist, H. de sample rate voor signalen werd bepaald maar zonder rekening te houden met storingen [6]. Vervolgens werd de sample rate opnieuw onder handen genomen door Shannon, C. die mogelijke storingen wel in rekening bracht [7]. De regel die gehanteerd wordt is dat de sample rate minstens twee keer zo groot moet zijn als de hoogste frequentie van het signaal. Vergelijking 1 toont het bemonsteringstheorema van Nyquist-Shannon in formule vorm.

$$f_s \geq 2 \cdot f_{max} \quad (1)$$

Een voorbeeld van een te lage sample rate is te zien in Figuur 4. De bovenstaande grafiek toont hier een sinus met een frequentie van 800 Hz. De frequentie waarmee bemonsterd wordt is 1000 Hz en dus niet volgens het Nyquist-Shannon theorema. De zwarte puntjes tonen waar dat het signaal juist bemonsterd wordt en op de onderstaande grafiek is het signaal te zien dat gevormd wordt wanneer er gebruikt wordt gemaakt van een incorrecte sample rate.



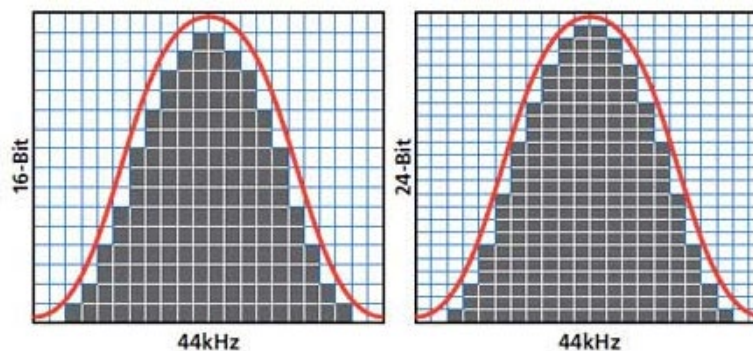
Figuur 4: Oscillogram voorbeeld lage sample rate [4]

2.1.1.3. Bit depth

Een andere parameter voor audio met als functie het omzetten van analoge audio naar digitale audio aan de hand van een ADC is de bit depth of ook wel beter gekend als de bit depth in audio toepassingen. De bit depth is het aantal bits die beschikbaar zijn in de ADC om het analoge signaal om te zetten in een digitale waarde. De bits beschrijven het aantal niveaus naar waar dat de analoge waarde omgezet kan worden. Vergelijking 2 toont aan hoe het aantal niveaus berekend kan worden. Voor 2 bits komt dit overeen met 4 niveaus. Voor 16-bits komt dit overeen met 4096 niveaus [4], [5].

$$\text{aantal niveaus} = 2^n \quad (2)$$

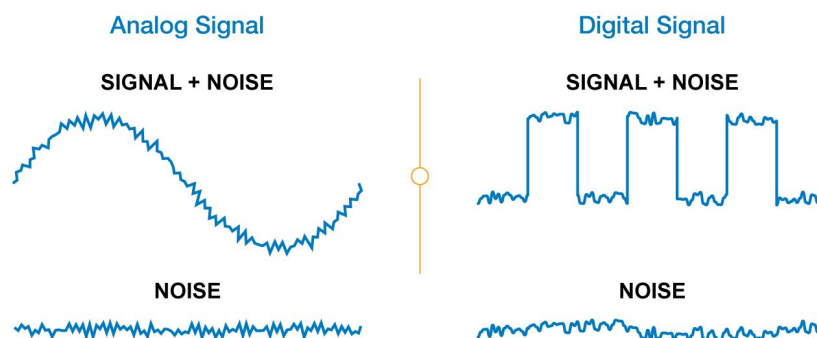
Figuur 5 toont hetzelfde signaal, aangeduid in het rood, bij dezelfde sample rate van 44 kHz. Het verschil is de bit depth, die bij de links grafiek 16-bits is en bij de rechts grafiek 24-bits is. De vergelijking tussen beide grafieken toont dat de 24 bit depth dichterbij het originele signaal komt. Hieruit kan geconcludeerd worden dat een hoger aantal bits ervoor zal zorgen dat het digitaal signaal dichterbij het originele analoge signaal in de buurt komt.



Figuur 5: Verschil in bit depth bij dezelfde sample rate [8]

2.1.1.4. Ruis

Voor een digitaal signaal is ruis, in de meest eenvoudige vorm, de enige ongewenste interferentie dat het digitaal signaal gaat degraderen. Een digitaal signaal is minder gevoelig aan ruis dan een analoge signaal. Een digitaal signaal is minder gevoelig omdat een digitaal signaal discrete signaalniveaus zijn, terwijl dat bij analoge signalen er een oneindig bereik van waarden is [9]. De vergelijking tussen ruis bij een analoge en digitaal signaal is geïllustreerd in Figuur 6.



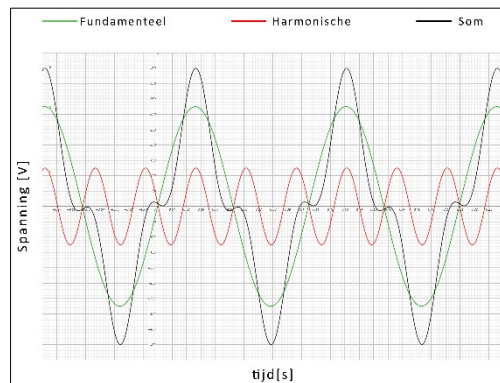
Figuur 6: Ruis bij een analoge en digitaal signaal [9]

De SNR definieert de verhouding waarmee het gewenste signaal tot de ruis ligt. Vaak uitgedrukt in decibel. De berekening voor de SNR uitgedrukt in dB gaat als volgt:

$$SNR_{dB} = 10 \cdot \log_{10} \left[\left(\frac{A_{signal}}{A_{noise}} \right)^2 \right] \quad (3)$$

2.1.1.5. Harmonische vervorming

De totale harmonische vervorming is een belangrijk criterium voor communicatiesignalen. THD is de verhouding van het totaal aantal harmonische componenten ten opzichte van de eerste harmonische/grondtoon van een sinus. Hoe lager de THD, hoe meer het werkelijke signaal overeenkomt met het theoretische signaal. Het signaal wordt bij een lagere THD meer accuraat gereproduceerd. THD komt vooral voor bij niet-lineaire elektronische lasten [10], [11]. Het resultaat van een harmonisch signaal op een normale sinus is geïllustreerd in Figuur 7. Op de figuur is het normaal signaal, een sinusgolf met een frequentie van 60 Hz, in het groen afgebeeld. De derde harmonische van de sinusgolf, met een frequentie van 180 Hz, in het rood afgebeeld. Het resulterend signaal gevormd door de som te nemen van beide golven is te zien aan de zwarte lijn.



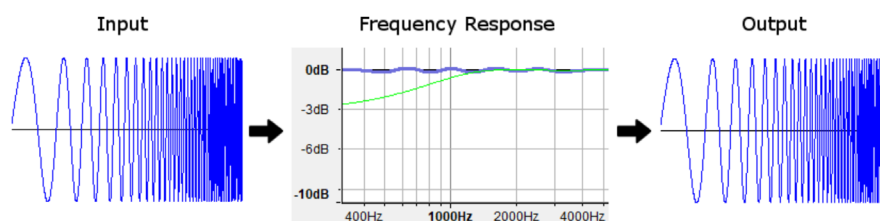
Figuur 7: Sinusgolf van 60 Hz (groen) met een harmonische van 180 Hz (rood) resulterend in de zwarte golf [10]

De som van de signalen toont aan dat er zowel constructieve als destructieve interferentie kan optreden voor de signalen.

2.1.1.6. Frequentierespons

Een frequentierespons beschrijft de reactie van een systeem op een sinusvormige ingang van variërende frequenties en analyseert het systeem in het frequentie domein. Het frequentierespons beschrijft dus het bereik van frequenties welke een component kan reproduceren. De frequentie voor audio wordt doorgaans getest voor een bandbreedte van 10 Hz tot 20 kHz omdat dit bereik algemeen beschreven wordt als het bereik van het menselijk gehoor [12]–[14].

De grafiek van een frequentierespons wordt uitgedrukt als het aantal decibels in functie van de frequentie afgebeeld op een logaritmische schaal [12]–[15]. Figuur 8 visualiseert de frequentierespons van een ideaal systeem waarbij de frequentierespons grafiek, in het blauw aangeduid, zo vlak mogelijk is bij de 0 dB lijn. De groene lijn toont een praktisch voorbeeld van een frequentierespons waarbij de lagere frequenties minder versterkt zijn dan de hogere. Het signaal varieert in dit geval van 400 Hz tot 5000 Hz.



Figuur 8: Frequentierespons [12]

De blauwe lijn op de grafiek toont een correct systeem dat juist afspeelt voor lage, middelmatige en hoge tonen in de juiste verhoudingen ten opzichte van elkaar. Het bereik dat een speaker haalt, bepaalt ook de kwaliteit van het systeem of speaker [15].

2.2. Universele Seriële Bus

De universele seriële bus, beter gekend als USB, is een welgekende standaard interface voor het sturen en ontvangen van seriële data. Er bestaan tal van apparaten die ondersteuning bieden voor de standaard, zoals: desktop, tablets, laptops, toetsenborden, muizen, camera's, printers, audio- en videoapparaten, embedded apparaten en zoveel meer. Dit maakt van de USB-interface een zeer veelzijdige, betrouwbare, snelle, energiezuinige, goedkope en sterk ondersteunde oplossing voor dataoverdracht [16], [17]. De standaarden van USB worden sinds het begin onderhouden en uitgebracht door de USB Implementers Forum.

De identificatie van een USB-apparaat gebeurt aan de hand van een descriptor die opgeslagen is in het ROM-geheugen van het USB-apparaat. De descriptor bevat alle data die verder zal beschrijven wat voor apparaat verbonden is, welke parameters dit apparaat gebruikt en wordt onderverdeeld in class, subclass en interfaces. [18] toont de verschillende mogelijkheden die beschikbaar zijn voor de descriptor parameters van class, subclass en interface. Aan de hand van de descriptor kan het apparaat ook gekoppeld worden aan standaard drivers in een besturingssysteem.

2.2.1. Versies

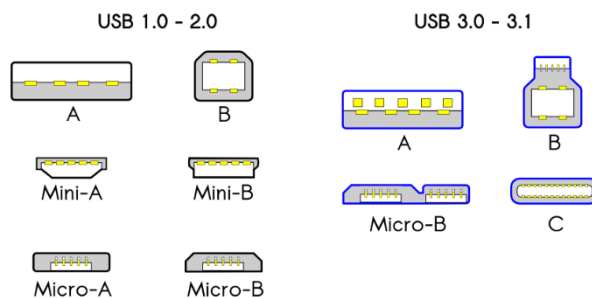
Doorheen de jaren is de snelheid van de USB-interface significant gestegen. De parameters in Tabel 1 geven van nieuwste standaard naar oude standaard weer welke de slogan is, de marketingnaam en met welke bandbreedte de standaarden theoretisch kunnen werken. De naamgeving van USB loopt vaak echter door elkaar heen waardoor er rond de juiste benamingen regelmatig verwarring ontstaat. Het is namelijk mogelijk om een Low-Speed USB 3.0 verbinding te hebben. Dit wil namelijk zeggen dat de fysieke connector een USB 3.0 layout heeft maar dat er snelheden van maximum 1.5 Mbps gehaald worden. Dit komt door het feit dat de USB-versies terugwaartse compatibiliteit hebben. Doorheen datasheets van verschillende fabrikanten kan dit soms verwarring veroorzaken naar de consument toe.

Tabel 1: USB Standaarden en snelheden [16], [19]

USB Slogan	USB Marketingnaam	Data Bandbreedte
USB4 20 Gbps	USB 4	20 Gbps
USB4 40 Gbps		40 Gbps
SuperSpeed+	USB 3.2	20 Gbps
SuperSpeed+	USB 3.2	10 Gbps
SuperSpeed	USB 3.0	5 Gbps
High-Speed	USB 2.0	480 Mbps
Full-Speed	USB 1.1	12 Mbps
Low-Speed	USB 1.0	1.5 Mbps

2.2.2. Vormfactor

Wanneer men spreekt over de vormfactor van een USB-apparaat bedoelt men de fysieke connector die verbindt naar de apparaten. Doorheen de jaren zijn hier verschillende types van ontstaan. Figuur 9 toont de verschillende connectoren per versie. Hierbij valt op te merken dat USB 3.0-3.1 over meer pinnen beschikt dan USB 1.0-2.0. Uit de vormfactoren zijn de meest voorkomende versies de USB type A en type C connectoren.



Figuur 9: Vormfactor USB-connectoren [20]

2.2.2.1. Pinout USB 1.0-2.0

Zoals voorheen vermeld, bestaan er voor andere USB-versies dezelfde connectoren types ondanks dat deze over een ander aantal pinnen beschikt. Zoals te zien in Figuur 9 bestaat voor USB 1.0-2.0 een type A, type B, Mini-A, Mini-B, Micro-A en Micro-B-uitvoering. In Tabel 2 is te zien welke signaal, kleur en omschrijving iedere pin heeft van de connector. Let wel op, de vierde pin kan beschikken over het ID-sigitaal voor USB OTG maar dit is enkel het geval indien de connector over vijf pinnen beschikt in plaats van vier pinnen. De ID-pin wordt gebruikt voor het bepalen welk van de twee apparaten juist het host-apparaat is en welke het slave-apparaat is. Wanneer de ID-pin op het PCB verbonden is met de grond wordt deze als een host gezien. Indien de ID-pin niet verbonden is op het PCB wordt deze als een slave gezien.

Tabel 2: Pinout connectoren USB 1.0-2.0 [21]

Pin	Signaal	Kleur	Omschrijving
1	Vcc	Rood	+5V
2	D-	Wit	Data-
3	D+	Groen	Data+
4	ID	/	USB OTG ID
4/5	GND	Zwart	Ground

2.2.2.2. Pinout USB 3.0-3.1

Zoals te zien in Figuur 9 bestaat voor USB 3.0-3.1 een type A, type B, Micro-B en type C. In Tabel 3 is te zien welke signaal, kleur en omschrijving iedere pin heeft van de connector voor type A, type B en micro B.

Tabel 3: Pinout connector USB 3.0-3.1 (type A, type B en micro B) [21]

Pin	Signaal	Naam	Omschrijving
1	Red	Vcc	+5 DC
2	White	D-	Data –
3	Green	D+	Data +
4	N/A	ID	OTG ID
5	Black	GND	Ground
6	Blue	StdB_SSRX-	Superspeed Transmit-
7	Yellow	StdB_SSRX+	Superspeed Transmit+
8	N/A	GND_Drain	Ground Signal Return
9	Purple	StdB_SSTX-	Superspeed Receive-
10	Orange	StdB_SSTX+	Superspeed Receive+

In Tabel 4 is te zien welke signaal, kleur en omschrijving iedere pin heeft van de connector voor type C. Het grootste voordeel ten opzichte van de vorige types van USB-connectoren is dat de connector omkeerbaar is en oriëntatie van de connector dus niet uit maakt. Het type C connector is het type USB-connector waar de industrie tegenwoordig het meest op inzet door de veelzijdige toepassingen die gelinkt zijn aan de USB type C connector. [22], [23] bespreken de wetgeving van het Europese parlement waarbij besloten is dat tegen 2024 er slechts één soort oplader (USB type C) bij draagbare apparaten geleverd mag worden. Dit omwille van de ecologische voordelen en de vele voordelen gelinkt aan USB type C. Zoals bijvoorbeeld de mogelijkheid tot zeer snelle snelheidsoverdracht voor visuele toepassingen die een hoge datasnelheid vereisen. Alsook biedt USB Type C de mogelijkheid om een hoger vermogen te leveren waardoor de mogelijkheid er is om apparaten op te laden.

Tabel 4: Pinout connector USB 3.0-3.1 (type C) [21]

Pin	Omschrijving	Pin	Omschrijving
A1	Ground	B12	Ground
A2	Superspeed differential pair 1, TX, positive	B11	Superspeed differential pair 2, RX, positive
A3	Superspeed differential pair 1, TX, negative	B10	Superspeed differential pair 2, RX, negative
A4	Bus power	B9	Bus power
A5	Configuration channel	B8	Side Band Use
A6	Differential pair 1, positive	B7	Differential pair 2, negative
A7	Differential pair 1, negative	B6	Differential pair 2, positive
A8	Side Band Use	B5	Configuration channel
A9	Bus power	B4	Bus power
A10	Superspeed differential pair 4, RX, negative	B3	Superspeed differential pair 3, TX, negative
A11	Superspeed differential pair 4, RX, positive	B2	Superspeed differential pair 3, TX, positive
A12	Ground	B1	Ground

2.2.3. Apparaat klasse

Zoals reeds besproken in de intro van hoofdstuk 2.2 Universele Seriële Bus zijn er veel soorten toepassingen voor het gebruik van USB. De toepassingen kunnen elk vallen onder hun eigen apparaat klasse die als een standaard gedefinieerd zijn door USB-IF. De USB-IF is een groep van bedrijven die zich inzetten op het ondersteunen en standaardiseren van de USB-markt. De groep bestaat uit grote bedrijven zoals Apple, HP, Intel, Microsoft, ... [24]

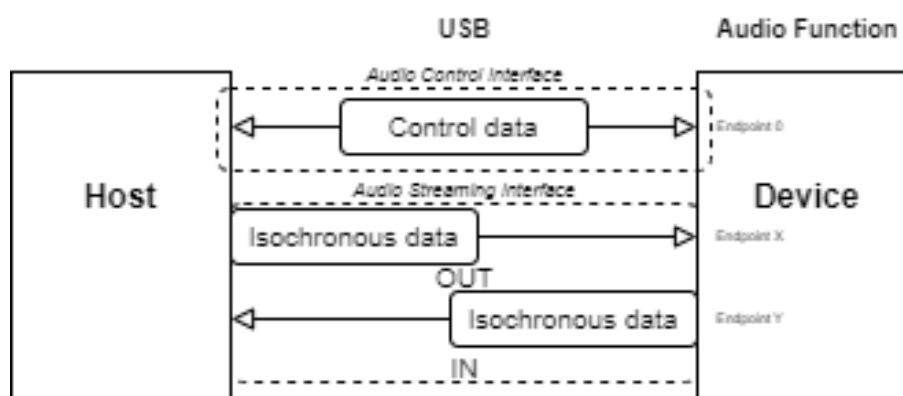
2.2.3.1. Human Interface Devices

Een eerste van zulke apparaat klasse is de HID. De HID is de interface tussen mensen en het apparaat. De HID behandelt alle input en output die ingegeven wordt door mensen. Typische voorbeelden van zulke apparaten zijn toetsenborden, muizen, etc. De HID-klasse wordt beschreven aan de hand van de USB-descriptor die geconfigureerd moet worden voor een HID. De juiste parameters staan in de documentatie van USB-IF [18], [25].

2.2.3.2. Audioapparaten

De audioapparaat klasse is een tweede soort apparaat klasse die van toepassing is op alle composieten apparaten die gebruikt worden voor het manipuleren van audio, voice en geluid gerelateerde functionaliteit. Dit telt zowel voor analoge als digitale audiodata en de functionaliteit voor het configureren van de geluidsomgeving, zoals bijvoorbeeld volume [26].

Het operationeel model voor de USB-audio wordt weergegeven in Figuur 10. In dit diagram is te zien dat de audio control interface ervoor zal zorgen dat de audio eigenschappen zoals bijvoorbeeld volume en het dempen van de audio mogelijk is. De audio streaming interface daarentegen wordt gebruikt voor het transporteren van de audiodata tussen de host en het apparaat [5].



Figuur 10: USB Audio operationeel model [5, p. 11]

2.2.3.3. USB Audio Device Class

De USB Audio Device Class beschrijft aan welke voorwaarden een USB-audio apparaat moet voldoen of over welke eigenschappen of parameters het USB-apparaat mag beschikken. Doorheen de jaren zijn hier drie verschillende iteraties van geweest. Hiervan zijn de Audio Device Class 1.0 en 2.0 tegenwoordig nog de meest gebruikte en ondersteunende versies. Het grote voordeel van Audio Device Class 3.0 ten opzichte van Audio Device Class 1.0 en 2.0 is dat de dataoverdracht tussen host en device gepauzeerd kan worden. Wanneer niet gepauzeerd stuurt de USB host isochroon data door waardoor er een groot aantal data verstuurd wordt. Dit betekent dat het energieverbruik van derde klasse het meest optimaal is waarbij de efficiëntie van de derde klasse ten opzichte van de originele 3.5mm connector zelfs gelijkaardig is [27], [28].

2.2.4. Synchronisatie

De synchronisatie tussen een host en het apparaat is belangrijk voor het transporteren van USB-audio aangezien te veel of te weinig data niet goed is voor de kwaliteit van audio. Wanneer er te veel data is zullen er namelijk delen overgeslagen worden en is er nog het risico dat een buffer overflow kan ontstaan. Wanneer er dan te weinig data is zullen er regelmatig pauzes gebeuren wanneer er audio wordt afgespeeld aangezien er geen data is om af te spelen. Alsook zijn er nog andere artefacten die kunnen voorkomen zoals klok afdrijven, jitter, etc. [5]

Voor USB-audio zijn er verschillende klokken aanwezig in het systeem. De eerste is de sample klok, de sample klok bepaalt de sample rate van de audiodata doorgestuurd zoals besproken in 2.1.1 Parameters. De tweede klok is de USB-bus klok, de USB-bus klok is afhankelijk van welke USB-versie er juist gebruikt. Een voorbeeld hiervan is een frequentie van 1 kHz voor een Full-Speed USB-apparaat en wordt aangeduid aan de hand van een SOF-pakket op de bus. De derde klok is de service klok, de service klok is de snelheid waarmee software loopt ten opzichte van verzoek pakketjes die verzameld kunnen worden tussen uitvoeringen. De vierde en laatste klok is een codec klok, de codec klok is de snelheid waarmee de audiodata wordt getransporteerd naar of van de codec module [5].

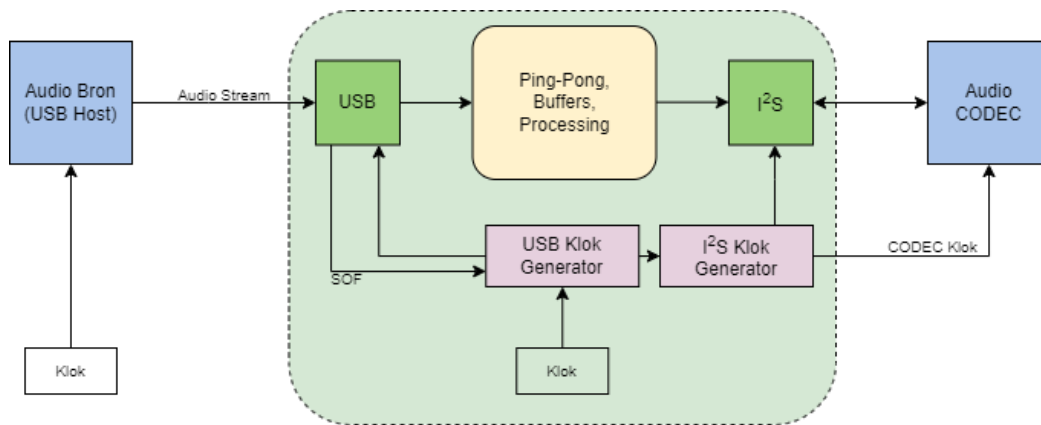
De USB-specificaties voorzien drie synchronisatie types voor de klokken. Tabel 5 toont de eigenschappen van de verschillende USB Audio synchronisatie types.

Tabel 5: USB Audio Synchronisatie types [5], [29]

Type	Host	Apparaat
Synchroon	Klok (fs) vergrendeld aan de USB SOF. Gebruikt impliciete feedback.	Klok (fs) vergrendeld aan de USB SOF. Gebruikt impliciete feedback.
Adaptief	Klok (fs) vergrendeld aan de sink. Gebruikt expliciete feedback.	Klok (fs) vergrendeld aan de data flow. Gebruikt impliciete feedback.
Asynchroon	Vrijlopende klok (fs). Voorziet impliciete feed forward naar het apparaat.	Vrijlopende klok (fs). Voorziet expliciete feedback naar de host.

2.2.4.1. Synchroon

Het eerste synchronisatie type is synchrone modus. Hierbij synchroniseert het apparaat zich iedere seconde met de SOF die iedere milliseconde ontvangen wordt vanuit de host via een USB-frame. Dit USB SOF-pakket wordt gegenereerd door de USB host. In Figuur 11 is te zien hoe het USB-apparaat gebruik maakt van het SOF-pakket om de interne oscillator te kalibreren en bijgevolg de sample klok instelt.

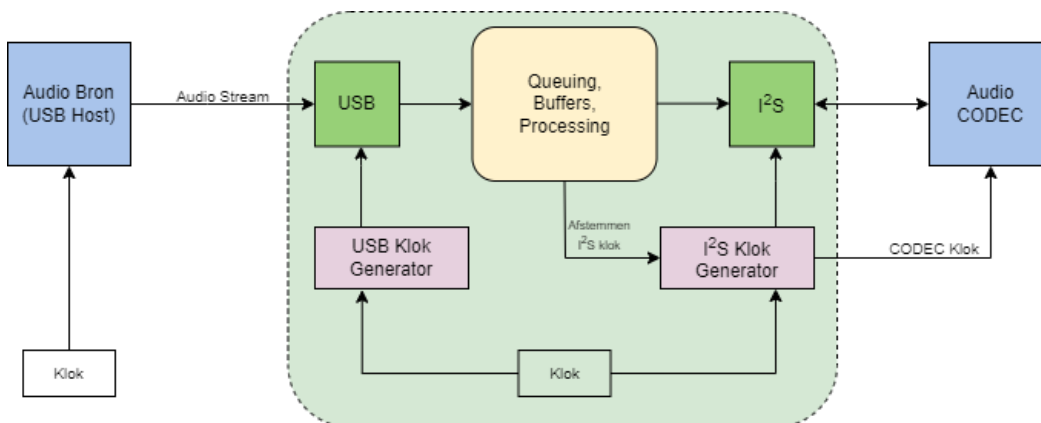


Figuur 11: USB synchrone synchronisatie [5, p. 14]

Voor dit synchroon proces wordt er iedere milliseconde een SOF-pakket gestuurd gevolgd door het aantal afgesproken samples dat voordien is afgesproken tussen audio host en apparaat. Dit SOF-pakket zal dienen om intern in het apparaat een klok te genereren. De klok zal ervoor zorgen dat de USB klok zich synchroniseert met de USB. Verder wordt hieruit ook de I²S klok gegenereerd [5], [29], [30].

2.2.4.2. Adaptief

De tweede synchronisatie modus voor USB is de adaptieve modus. Hierbij passen zowel de bron als het apparaat de datasnelheid aan afhankelijk van de snelheid ten opzichte van elkaar zoals te zien is in Figuur 12.

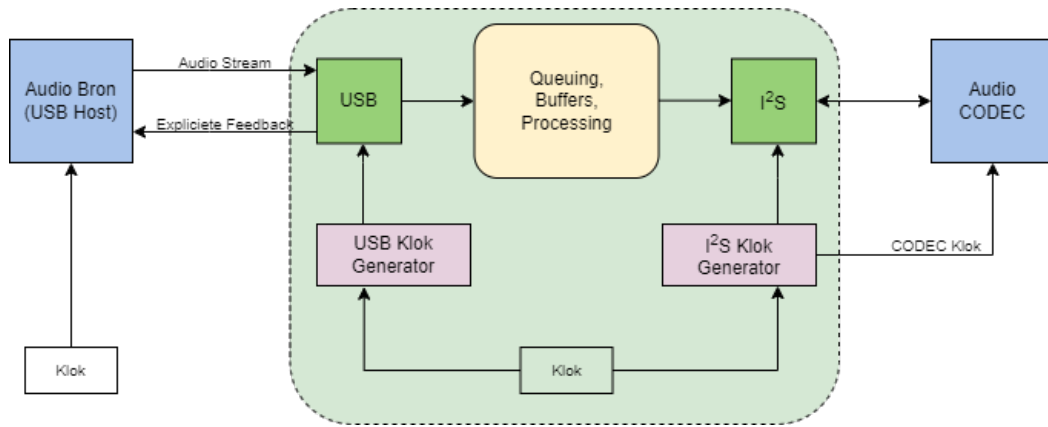


Figuur 12: USB adaptieve synchronisatie [5, p. 15]

Een adaptieve bron produceert data aan een snelheid die afhankelijk is van het apparaat. Het apparaat voorziet in deze modus geen feedback naar de host. Een adaptief apparaat en de USB-host spreken tijdens het maken van een verbinding een sample rate af. De sample rate is de snelheid waarmee er data over USB gestuurd zal worden. Er zal iedere milliseconde data gestuurd worden in een dataframe bestaande uit een SOF gevolgd door het aantal afgesproken samples. Aangezien er geen feedback is van het USB-apparaat naar de USB-host zal de snelheid van de datapakketten niet wijzigen. Het USB-apparaat houdt echter in de buffer bij of er al dan niet te snel of te traag data getransporteerd wordt vanuit de USB-host en zal evenredig de snelheid van de I²S dataoverdracht wijzigen om zo de buffer op het juiste pijl te houden [5].

2.2.4.3. Asynchroon

De laatste synchronisatie type is de asynchrone modus. Hierbij is het apparaat met de audiobron niet gesynchroniseerd via een gemeenschappelijke klok of het USB SOF signaal. Bij asynchrone apparaten is er een vrijlopende interne klok die de gewenste datasnelheid bepaalt. Er is een expliciete feedback naar de bron vanuit het apparaat die de snelheid van de data regelt zoals te zien is in Figuur 13.



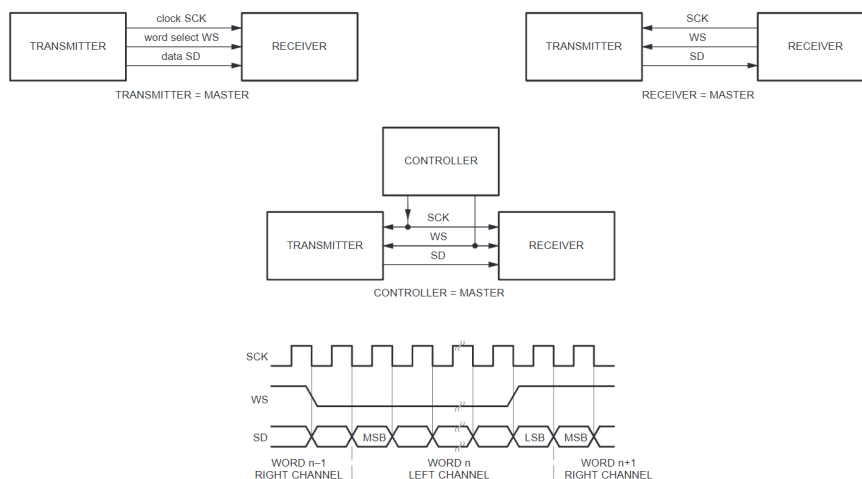
Figuur 13: USB asynchrone synchronisatie [5, p. 13]

Het proces dat de asynchrone modus overloopt is dat er eerst een sample rate wordt afgesproken tussen bron en apparaat. De snelheid is de snelheid waarmee data, die start met een SOF iedere milliseconde, verstuurd zal worden tussen de bron en het apparaat. Het apparaat heeft een buffer waarin de ontvangen data opgeslagen zal worden. Het apparaat zal de buffer en het verwerken van data voor zijn rekening houden. Aangezien de bron en het apparaat op een andere klok werken is er een mogelijkheid dat de klokken niet exact overeenkomen wat kan lijden tot een buffer overflow of underflow. Via de feedback verbinding tussen host en apparaat kan het apparaat aangeven aan de host hoe de buffer juist opgevuld is en of de data al dan niet aan een hogere of lagere snelheid doorgestuurd moet worden [5], [29], [30].

Verschillende bronnen [31]–[34] geven aan dat asynchrone modus de meest geavanceerde modus is voor USB audio overdracht. Omdat hierbij de datasnelheid volledig gecontroleerd kan worden door het apparaat zelf wat verschillende mogelijkheden met zich meebrengt zoals bijvoorbeeld het aanzienlijk lager aantal jitters en dat de klok ook slave kan zijn.

2.3. I²S

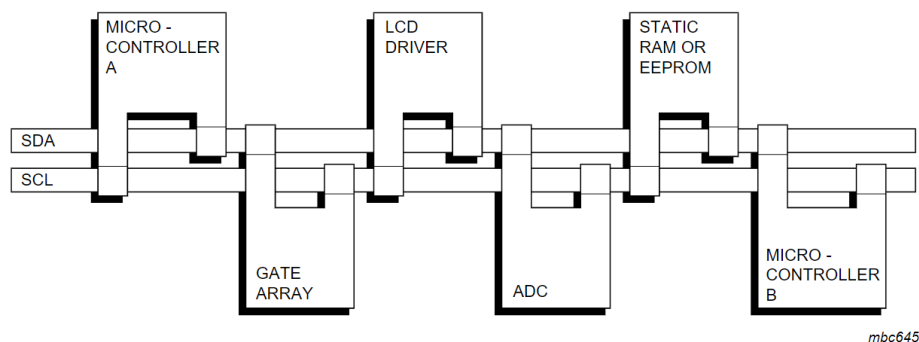
De I²S is een seriële databus specifiek ontworpen voor het gebruik bij digitale audio door Philips Semiconductors (tegenwoordig gekend als NXP Semiconductors en verder in de scriptie zo vernoemd) in 1996. De seriële data kan verwerkt worden door verschillende systemen zoals bijvoorbeeld: ADC, DAC, DSP, etc. Figuur 14 toont een standaard configuratie waarbij er drie lijnen te zien zijn. ‘SCK’ is de bit klok, ‘WS’ de woord select klok en ‘SD’ de datalijn is waar de werkelijk audiodata op zal getransporteerd worden. De bit klok zal een periode hebben met de lengte van een bit. De woord select klok zal wisselen bij een lengte van een woord aantal bits met een verschuiving van een bit naar links ten opzichte van de MSB op de datalijn zoals gedefinieerd in de standaard van NXP Semiconductors [35], [36].



Figuur 14: Eenvoudige I²S systeem configuratie en interface timing [36, p. 1]

2.4. I²C

De I²C standaard is een seriële multi-master databus ontworpen door NXP Semiconductors voor het doorsturen en ontvangen van digitale data. Enkele van de toepassingen voor I²C zijn: lcddrivers, leddrivers, applicatie georiënteerde schakelingen, etc. Voordelen van de interface zijn: laag aantal signalen, makkelijk om apparaten toe te voegen, extreem laag stroom verbruik, ruis onafhankelijk, ruim spanningsbereik en ruim temperatuurbereik. De illustratie in Figuur 15 toont het diagram van een voorbeeldtoepassing waarbij twee microcontrollers aanwezig zijn. Hiervan zijn ook de twee lijnen zichtbaar waarbij ‘SDA’ de seriële datalijn is en ‘SCL’ de seriële klok is [37].



Figuur 15: I²C toepassingsvoorbeeld [37, p. 7]

2.5. USB naar I²S

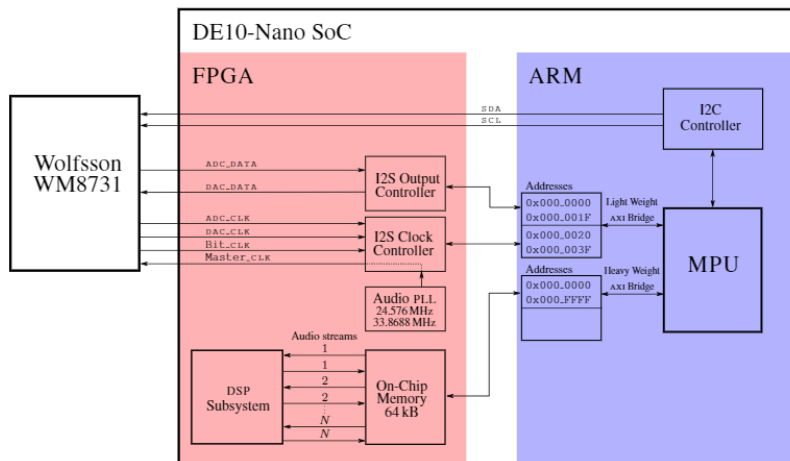
Zowel USB als I²S zijn seriële protocollen voor het verzenden van data. De manier waarop de data wordt verstuurd is verschillend voor iedere protocol. Voor USB zit de data verpakt in een pakket startend met een SOF en eindigend met een EOF welke respectievelijk het begin en einde van een datapakket aanduiden. Voor USB-audiodata in het geval van een stereo pakket bestaat de data uit het linkerkanaal gevolgd door het rechterkanaal waarbij elk kanaal het voor gedefinieerd aantal samples bevat (vb. 48 samples). Voor I²S is er een continue stroom van data volgend aan de bit klok en de word select klok waarbij de data per kanaal verbonden is aan het hoog of laag zijn van de word select klok.

Om de USB-audiodata om te zetten naar de datalijn van het I²S protocol moet de data gesynchroniseerd worden met de bit klok en word select klok. Omzettingen naar I²S zijn mogelijk door verschillende technieken.

2.6. FPGA gebaseerde audio DSP

[38]–[42] toont aan dat het verwerken van digitale signalen kan gebeuren door gebruik te maken van een FPGA. Ieder onderzoek verwerkt de data voor verschillende doeleinden.

[38] maakt gebruik van een Terasic DE10-Nano Development kit voor het verwerken van audio en het toepassen van filters hierop. De processor draait een Linux besturingssysteem waarop het programma uitgevoerd zal worden. De kostprijs van de development kit is 215 euro en wordt gebruikt in vele toepassingen en door universiteiten [43]. De kwaliteit van audio behaald in dit onderzoek is 16-bit bij een sample rate van 48 kHz met een zeer lage vertraging op het DSP. De opstelling van dit onderzoek is geïllustreerd in Figuur 16.



Figuur 16: DE10-Nano Opstelling [38, p. 5]

[39] onderzoekt het gebruik van een Xilinx XC3S50AN in het ontwerpen van een geluidskaat. De geluidskaat maakt verbinding met een pc door gebruik te maken van een USB-bus in full-duplex. De geluidskaat is compatibel met audiokwaliteit van 8-bit met een sample rate van 44.1 kHz voor een mono-installatie. De kostprijs van de FPGA is 26 euro [44].

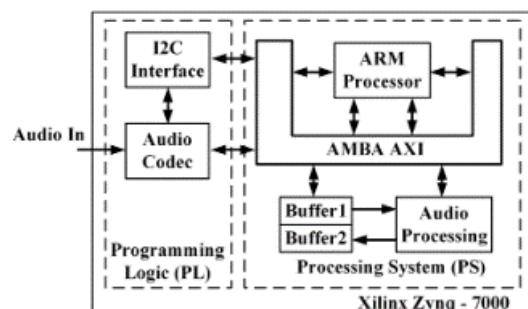
[40] onderzoekt de specifieke apparaat klasse voor USB van audio-apparaten waarbij gebruik werd gemaakt van de USB LPM aan een High Speed snelheid. Als controller werd er gekozen voor een Altera Stratix III FPGA met een kostprijs van 30 euro [45]. In dit onderzoek werd de vergelijking gemaakt tussen de 3 USB Audio Device Classes (1.0, 2.0 en 3.0) en welke het meest energie-efficiënt was. In Tabel 6 is het verbruik per klasse te zien bij eenzelfde datasnelheid voor de audio. De conclusie van dit onderzoek is dat USB ADC 3.0 het meest energie-efficiënt is, maar dat de technologie nog weinig, tot niet ondersteund wordt in apparaten. USB ADC 2.0 is nog het meest gebruikte apparaat klasse.

Tabel 6: Verbruik van ADC 1.0, ADC 2.0 en ADC 3.0

USB audio class	Phy (system) average current (ma)	Audio bit rate
Adc3.0	2.16 ma	320 kbps
Adc2.0	29.65 ma	320 kbps
Adc1.0	7.54 ma	320 kbps

[41] maakt gebruik van Xilinx XC3V200 voor het ontwerpen van een DSP-module in een KVM-systeem. De kostprijs van de FPGA is 45 euro [45]. Waarbij alle KVM-signalen samen worden verstuurd met het groot aantal audiodata dat wordt verstuurd bij het I²S protocol. Het verlies van audiodata was in dit onderzoek een belangrijk aspect.

[42] onderzoekt de analyse van binnenkomende audio waarbij het geluid van een gitaar geanalyseerd wordt op frequenties, door FFT, en amplitudes. In dit onderzoek wordt gebruik gemaakt van de Xilinx Zynq-7000 FPGA met een kostprijs van 70 euro [46]. In Figuur 17 is de hardware architectuur van dit onderzoek te zien.



Figuur 17: Hardware Architectuur [42, p. 2]

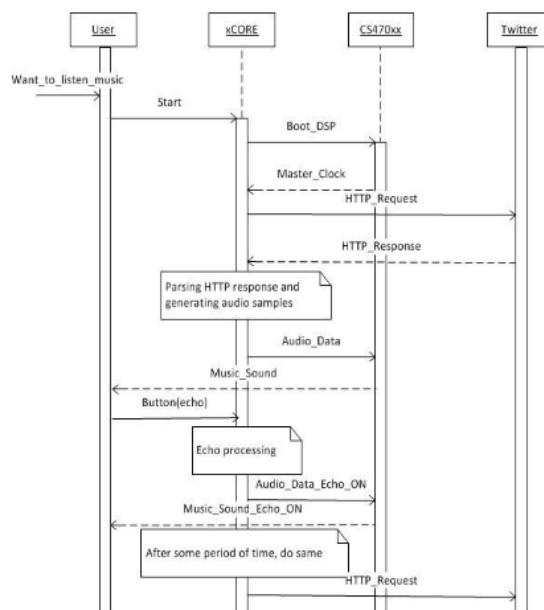
De voorgaande onderzoeken bewijzen dat het gebruik van een FPGA een goede optie is doordat FPGA's zeer low-level programmeerbaar zijn en bijgevolg zeer snel en energie-efficiënt zijn. Het grootste nadeel dat terugkomt in alle bovenstaande onderzoeken is dat de prijs van FPGA's relatief hoog zijn.

2.7. Microcontroller gebaseerde audio DSP

Ander onderzoek van [47]–[50] onderzoekt anderzijds het gebruik van een microcontroller voor het verwerken van digitale signalen. Ieder onderzoek verwerkt data voor een ander doeleinde of andere toepassing.

[47] onderzoekt het verwerken van data door gebruik te maken van Espressif ESP32. In dit onderzoek werd data van een temperatuursensor en versnellingsmeter uitgelezen en getoond op een OLED display door het I²C protocol. De kostprijs van een ESP32 is 1,30 euro [51].

[48] maakt gebruik van een XMOSE xCORE multicore microcontroller voor het afspelen van muzieknoden die via een Twitter bericht ontvangen worden. Door een bericht te sturen naar een profiel op Twitter door gebruik te maken van een bepaalde syntax decodeert de microcontroller de muzieknoden. Het gedecodeerde bericht is dan I²S audiodata van 16-bits met een sample rate van 44,1 kHz of 48 kHz. Figuur 18 toont de sequentie waarin bepaalde hardware en/of APIs worden aangesproken. De kostprijs van een XMOSE xCORE microcontroller is 4,80 euro [52].



Figuur 18: Sequentie diagram van het audiosysteem [48, p. 4]

[49] onderzoekt het opnemen, opslaan en afspelen van geluid door gebruik te maken van een ARM7 microcontroller. De audio heeft een kwaliteit van 8-bit met een sample rate van 8 kHz. De kostprijs van een ARM7 microcontroller is 9 euro [53]. Het onderzoek kan audio wegschrijven naar een SD kaart en afspelen van een SD kaart met een lage audiokwaliteit.

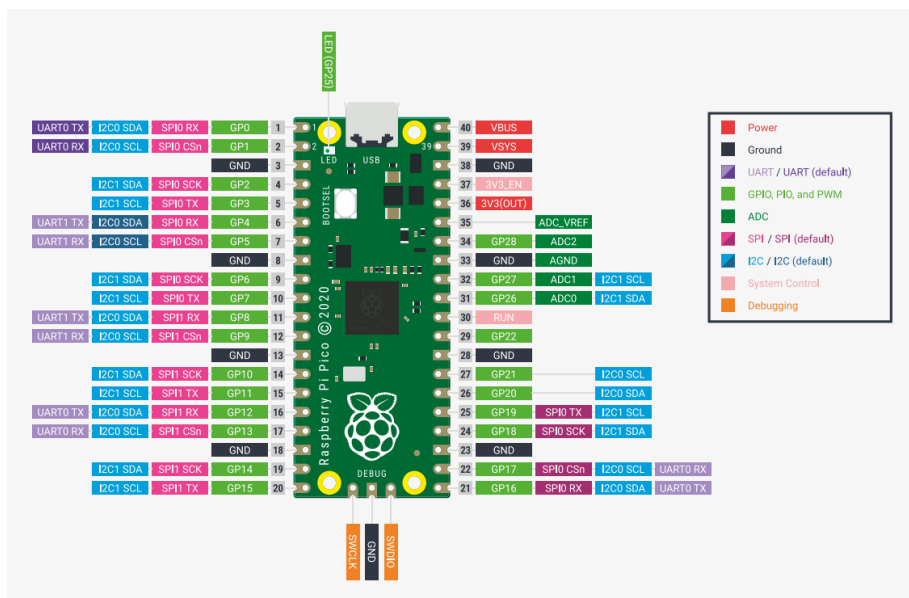
[50] deed onderzoek naar ANC in headsets door gebruik te maken van een PIC24H microcontroller met een kostprijs van 3 euro [54]. Het gebruik van een goedkope microcontroller in de ANC-toepassing zorgt ervoor dat de kostprijs van het gehele product verlaagt.

2.8. Raspberry Pi Pico (RP2040)

De Raspberry Pi Pico is een relatief nieuw ontwikkelaarsbordje uit 2019. De RP2040 is de chip aanwezig op dit bord en is de eerste microcontroller uitgebracht door Raspberry Pi waarbij hoge prestatie en lage kost voornamelijk de hoofdzaken waren tijdens het ontwerpen [55]. De technische specificaties voor de microchip zijn als volgt.

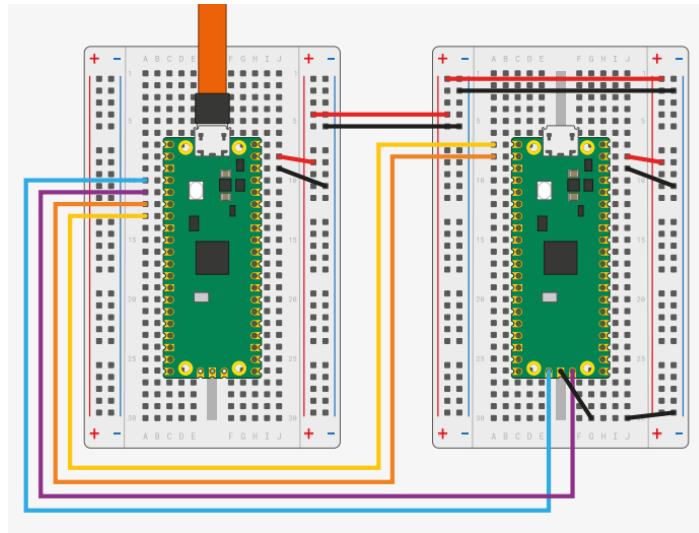
- Dual ARM Cortex-M0+ [133 MHz].
- 264 kB on-chip SRAM.
- Ondersteuning voor 16 MB off-chip Flash memory via QSPI chip.
- DMA controller.
- On-chip programmeerbare LDO om core spanningen te generen.
- 2 on-chip PLLs om USB en core kloks te genereren.
- 30 GPIO pinnen, waarvan 4 pinnen als analoge uitgangen gebruikt kunnen worden.
- Randapparatuur:
 - 2 UARTs
 - 2 SPI-controllers
 - 2 I²C controllers
 - 16 PWM-kanalen
 - USB 1.1 controller met host en apparaat ondersteuning.
 - 8 PIO-state machines.
- Kostprijs van 1,04 euro per microchip.

De pinout van het Raspberry Pi Pico ontwikkelaars bord is geïllustreerd in Figuur 19. Hierop zijn de ingangen en uitgang van het bordje aangeduid. Het programmeren van dit bordje kan gebeuren aan de hand van de BOOTSEL-knop die gedetecteerd wordt als een USB mass storage apparaat wanneer het bordje over USB verbonden wordt met een pc terwijl de BOOTSEL-knop wordt ingedrukt en na het insteken in de USB-poort terug wordt losgelaten. Vervolgens kan een gegenereerd bestand met de “.ulf2” extensie in de opslag van de Raspberry Pi Pico geplaatst worden en de Raspberry Pi Pico zal opnieuw opstarten met dit project actief. Een alternatieve oplossing voor het ontwikkelaarsbord te programmeren is aan de hand van de aanwezige SWD poort [55]–[57].



Figuur 19: Raspberry Pi Pico Ontwikkelaarsbord Pinout [56, p. 5]

Voor het debuggen van programma's voor de Raspberry Pi Pico kan er gebruik gemaakt worden van firmware ontwikkeld door Raspberry Pi zelf. De firmware, genaamd "picoprobe", zorgt ervoor dat programma's geüpload kunnen worden naar een RP2040 gebaseerde microcontroller door het programma via een USB-verbinding te sturen naar een eerste RP2040 gebaseerde microcontroller die dit programma zal uploaden naar een tweede RP2040 gebaseerde microcontroller aan de hand van UART en de SWD-interface [58]. Het verbindingsdiagram voor een debug opstelling bij gebruik van twee Raspberry Pi Picos zijn te zien in Figuur 20.

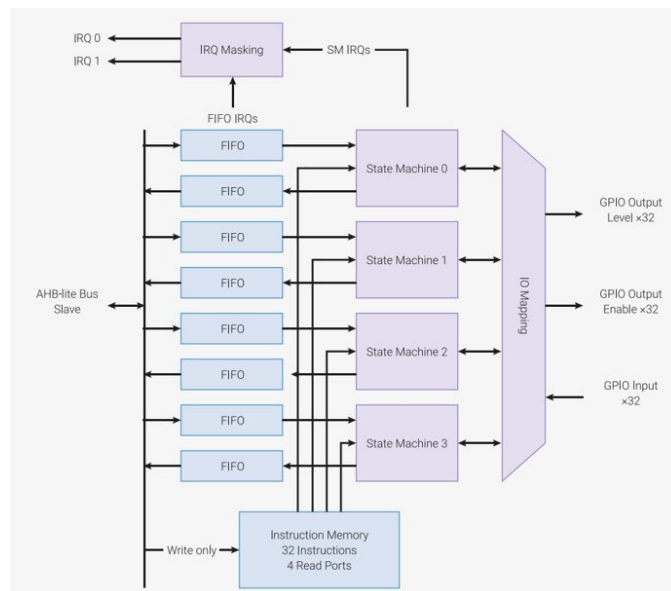


Figuur 20: Raspberry Pi Picoprobe verbindingsdiagram [58, p. 58]

2.8.1. Programmable Input/Output

Het grootste voordeel aan de RP2040 zijn de PIO-state machines aanwezig in de chip. Dit is namelijk een flexibel en zeer energie efficiënt kenmerk van de RP2040. De PIO wordt gebruikt voor het manipuleren van GPIO's, het verzenden van data en is gespecialiseerd voor IO en interfaces dankzij de precieze timing die behaald kan worden en de nauwe integratie met hardware. In de RP2040 zijn er twee state machines beschikbaar met ieder twee 32-bit shift registers, twee 32-bit scratch registers, vier maal 32-bit bus FIFO in iedere richting, fractionele klok verdeler, flexibele GPIO-mappen, DMA-interface en IRQ vlaggen [57].

Aan de hand van de PIO kunnen nieuwe interfaces gegenereerd worden zoals 8080 parallel bus, 6800 parallel bus, I²C, 3-pin I²S, SDIO, SPI, DSPI, QSPI, UART, DPI, VGA, etc. Het diagram van een enkele PIO-blok is te zien in Figuur 21. Hierin zien we hoe de data getransporteerd wordt naar de state machines via de FIFO en aangestuurd wordt via het instructie geheugen.



Figuur 21: PIO-state machine blok [57, p. 310]

Het PIO-state machine bestaat uit korte, binaire programma's die uitgevoerd worden en maakt gebruik van negen instructies. De instructies met parameters worden beschreven waarbij een basis instructie gevolgd kan worden door een extra side set instructie die lost staat van de originele instructie. De structuur van dit geheel ziet er als volgt uit:

`<instructie> (side <side_set_waarde>) ([vertraging_waard])`

`jmp <voorwaarde> <doel>`
 spring naar een doel in het programma aangeduid met een label of waarde.

`wait <polariteit> (gpio/pin/irq)* <*_num>`
 wacht instructie met het aantal vertraging in cycles.

`in <bron>, <bit_count>`
 shift `bit_count` bits van de bron in het ISR.

`out <bestemming>, <bit_count>`
 shift `bit_count` bits van de bron uit het OSR in de bestemming.

`push (indien_vol)`
 duw de inhoud van de ISR in de RX FIFO (32-bit word, ISR volledig nullen).

`pull (indien_leeg)`
 laad een 32-bit word van de TX FIFO in de OSR.

`mov <bestemming>, (op) <bron>`
 kopieer data van de bron naar de bestemming.

`irq set/nowait/wait/clear <irq_num>`
 configureer de IRQ vlaggen geselecteerd door de index.

`set <bestemming>, <waarde>`
 schrijf onmiddellijk de waarde naar de bestemming.

Het algemene PIO-programma bestaat uit verschillende Directives Assembly Programs. De directives zijn instructies die gebruikt worden door de assembler om het proces te automatiseren en de leesbaarheid van het programma zelf te verhogen. De directives zien er als volgt uit:

`.define (PUBLIC) <symbool> <waarde>`
 definitie van een variable met optioneel globaal (PUBLIC) argument, die doorheen het hele programma beschikbaar is.

`.program <naam>`
 start van een nieuw programma met een naam.

`<LABEL>`
 groep van uitspraken naar waar de `jmp` instructie naar toe gewezen kan worden.

`.wrap_target`
 specificeert de instructies waar de uitvoering doorgaat door program wrapping. Indien niet gespecificeerd start deze aan het begin van het programma.

`.wrap`
 specificeert de instructie naar welke het programma herstart (naar `.wrap_target` instructie).

`.lang_opt <lang> <name> <option>`
 specificeert een optie van het programma gerelateerd op de programmeertaal generator zoals bijvoorbeeld (c-sdk, python en hex).

`.word`
 slaat een 16-bit waarde op als een instructie in het programma.

`; of //`
 duiden een opmerking aan.

`.side_set <aantal> (opt) (pindirs)`
 Indien de directive aanwezig is geeft het aantal weer hoeveel bits er gebruikt worden voor het aantal side-set bits voor een maximaantal van vijf. Hierbij wordt gelijktijdig aan het uitvoeren van de instructie de pinnen/bits naar gelang de waarde gewijzigd. De bits worden bijvoorbeeld gebruikt voor het genereren van klokken voor communicatieprotocollen.

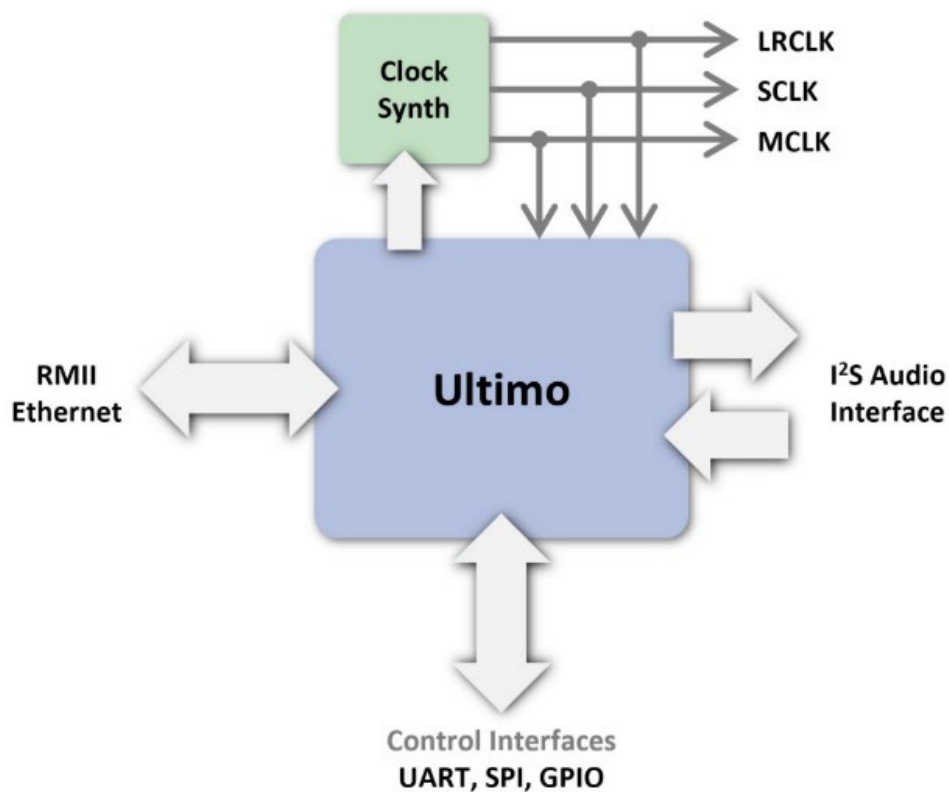
2.9. Dante

Dante is een platform van een leider in professionele audiovisuele netwerktechnologie genaamd Audinate. Audinate zijn Dante platform distribueert digitale audio en video signalen over ethernet netwerken en is ontworpen om alle voordelen van netwerken naar de professionele audiovisuele industrie te brengen. Dante is tegenwoordig een courante digitale medianetwerkoplossing die gebruik maakt van de standaard IP-infrastructuur naar netwerkkapparaten. Dit maakt interoperabiliteit eenvoudig en betrouwbaar. Dante distribueert ongecomprimeerde, meerkanaals audiovisuele data via een standaard Ethernet netwerk met een vertraging van enkele milliseconden en perfecte synchronisatie. Dante is een combinatie van software, hardware en hoge snelheid netwerkprotocollen die het gehele plaatje vormen [59, p. 12].

2.9.1. Ultimo

Dante Ultimo is een familie van de Dante microcontrollers die kosteffectief is, een laag kanaalaantal heeft en een groot bereik aan eigenschappen heeft. Hierbij is een ready-to-use Dante interface beschikbaar met maximaal vier bi-directionele kanalen van audio met een maximale sample rate van 96 kHz en een vertraging van 1 ms. Het doel van de Ultimo processor is om het digitale I²S audioformaat om te zetten naar het Dante netwerkprotocol [60, p. 10], [61].

Het diagram in Figuur 22 toont alle interfaces waarmee de Dante Ultimo chip mee gaat communiceren. Aan de bovenzijde van dit diagram is een aparte klok component te zien die gebruikt wordt voor het genereren van hoge kwaliteit, lage jitter audio klokken. De rechterzijde van het diagram toont het I²S interface waar de digitale audio stream zit. Aan de linkerzijde van het diagram zit de RMII Ethernet interface die in vergelijking met de originele MII Ethernet interface slechts de helft van de signalen gebruikt. De interface wordt gebruikt voor het Dante protocol en de communicatie over IP-infrastructuur. Aan de onderzijde van het diagram staan verschillende controle interfaces die de Ultimo chip ondersteunt. De interfaces zijn UART, SPI en GPIO [60], [61, p. 2].



Figuur 22: Dante Ultimo Network, Audio en Control Interfaces [61, p. 2]

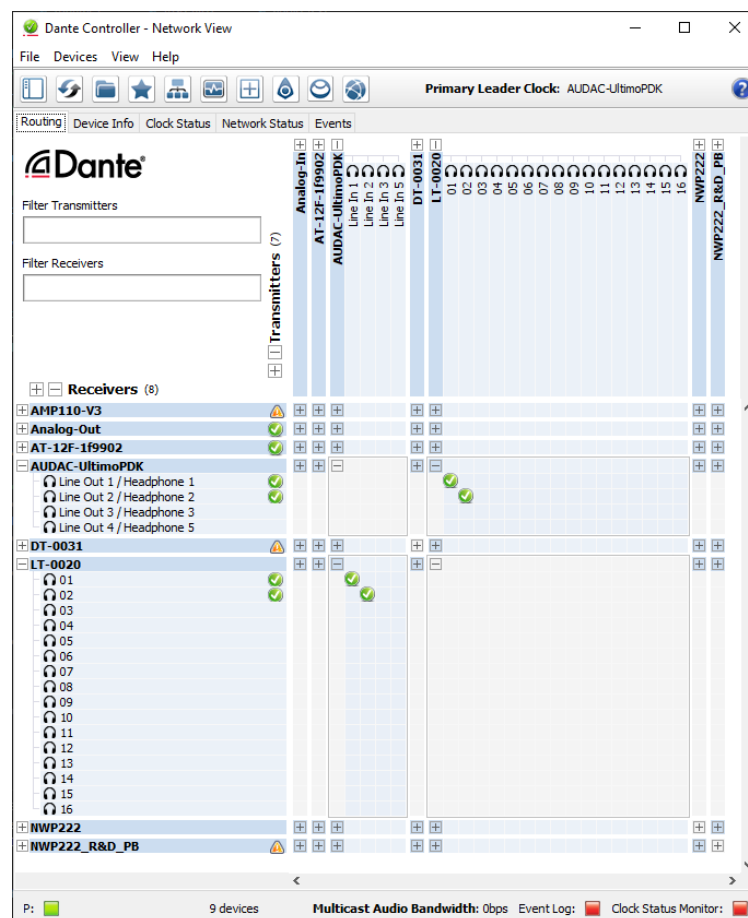
Er zijn verschillende toepassingen waarin de Dante Ultimo chip gebruikt wordt zoals bijvoorbeeld: versterkers, actieve luidsprekers, hoofdtelefoons, microfoons, ... [60], [61, p. 2]

2.9.2. Dante Controller

De Windows en MacOS softwaretoepassing genaamd Dante Controller van Audinate geeft gebruikers de mogelijkheid om de audio en video rond Dante netwerken de configureren en routen. Belangrijke eigenschappen van Dante Controller zijn:

- Alle Dante audio en video-apparaten en kanalen monitoren.
- Alle netwerk en klokken van Dante apparaten monitoren.
- Configureren van routen, van audio en video, op Dante apparaten.
- Vergrendelen en ontgrendelen van Dante configuraties op apparaten.
- Hernoemen van kanaalnummers naar namen om configuratie te vereenvoudigen.
- Het opslaan en herstellen van reeds geconfigureerde voorinstellingen.
- Netwerk status informatie bekijken.
- Etc. [59]

De illustratie in Figuur 23 is een screenshot uit de Dante Controller applicatie op een Windows platform. Hier zijn de zenders, die toegankelijk zijn in het netwerk, zichtbaar in de kolommen. De ontvangers, die toegankelijk zijn in het netwerk, zijn zichtbaar in de rijen. Via de Dante Controller interface kunnen zenders aan ontvangers verbonden worden. In het geval van Figuur 23 staan er twee ontvangers en een zender open. De zender genaamd 'AUDAC-UltimoPDK' is de Dante audionetwerk interface die geïntegreerd zit in de Audac XMP44 en over een Ultimo chip beschikt. Deze zender zendt enerzijds twee Dante Out kanalen uit naar ontvanger 'LT-0020' met twee Dante In kanalen. Waarbij 'LT-0020' een Dante Virtual Sound Card is. Deze uitgangen vormen een stereo uitgang. Anderzijds zendt de zender dezelfde vier kanalen naar een Dante Virtual Soundcard. (zie 2.9.3 Dante Virtual Soundcard) die wederom gelinkt zijn aan uitgangen. Via de eenvoudige interface is het mogelijk om de Dante te configureren en routen naar verlangen [59].



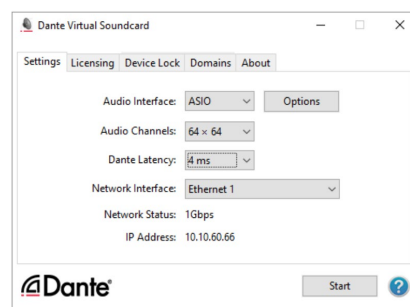
Figuur 23: Dante Controller Applicatie Interface

2.9.3. Dante Virtual Soundcard

De Dante Virtual Soundcard applicatie is een applicatie geschreven door Audinate voor Windows en MacOS. Dankzij Dante virtual Soundcard software is het mogelijk om de pc te laten functioneren als een Dante apparaat dat het toelaat om Dante audioverkeer te verzenden en ontvangen door gebruik te maken van een standaard Ethernet poort en geen extra toevoeging van gespecialiseerde hardware. Wanneer een Dante Virtual Soundcard geconnecteerd is met een Dante netwerk beschikt het over volgende eigenschappen:

- Starten en stoppen van de applicatie.
- Vergrendelen en ontgrendelen van de applicatie.
- Wijzigen van de apparaat vertraging.
- Selecteren van het aantal audiokanalen.
- Wijzigen van de Ethernet netwerk interface.
- Wijzigen van de audio sample rate, audio codering bit depth en naam van het apparaat [62].

De screenshot in Figuur 24 is van de Dante Virtual Soundcard op een Windows PC. Hierin kunnen de verschillende eigenschappen en parameters naar behoren worden ingesteld vooraleer de virtuele geluidskaart gestart kan worden. Na het instellen van de parameters kan de virtuele geluidskaart gestart worden mits de voorwaarde dat er een juiste licentie geactiveerd is in het tweede tabblad van de applicatie.



Figuur 24: Dante Virtual Soundcard Applicatie Interface [62, p. 1]

2.9.4. Precision Time Protocol

Audiosignalen zijn zeer tijdsgevoelige signalen aangezien het menselijk oor het vrijwel meteen zal merken indien er verschuivingen op het tijdsdomein plaatsvinden van verschillende kanalen of luidsprekers ten opzichte van elkaar. Omwille van deze reden is het zeer belangrijk om alle apparaten over het netwerk gesynchroniseerd zijn aan een dezelfde klok. Voor het synchroniseren van de meerdere verschillende klokken op het netwerk maken alle Dante apparaten gebruik van het IEEE 1588 PTP-standaard gedefinieerd door IEEE SA. De laatste versie in de standaard is IEEE Std 1588bTM-2022 volgens [63] ook wel gekend als PTPv2. De oudere standaard IEEE Std 1588TM-2002 volgens [64] ,ook wel gekend als PTPv1. De PTPv1 wordt ook nog ondersteunt door Dante en wordt zelfs nog als standaard instelling gebruikt. Van alle Dante ingeschakelde apparaten zal er één apparaat gekozen worden als PTP-leider klok. De leider klok zal een lokale klok genereren en deze ter beschikking stellen aan het netwerk waardoor de andere Dante apparaten dan kunnen synchroniseren aan de hand van de globale klok. De PTP volg klokken operatie maakt gebruik van een kristaloscillator om te verbinden met de PTP-leider klok. Conceptueel gezien heeft elke flank van een woord netwerktime in seconden en nanoseconden geassocieerd met zich. De kristaloscillator zal voortdurend worden bijgesteld aan de hand van gemeten compensatie met nauwkeurigheid in het netwerk zodanig dat de lokale klok overeenkomt met de globale PTP-leider klok [59, pp. 28–29], [60, pp. 31–32].

3. Ontwikkeling

In dit hoofdstuk van de scriptie wordt de effectieve ontwikkeling van USB-audio naar I²S voor meerkanaals bi-directionele data uitgewerkt en verduidelijkt voor zowel hardware als software. Het illustreert welke stappen ondernomen worden om tot een goed resultaat te komen waarbij het doel van de masterproef centraal staat.

3.1. Hardware

De hardware voor de masterproef lag voor een klein deel vast. De opstelling bestaat namelijk uit twee grotere delen. Een eerste deel bestaat uit een processor die zal dienen als een USB-geluidskaart die verantwoordelijk is voor de bi-directionele omzetting van de USB-data naar de I²S data. Het tweede hardware deel bestaat uit een processor die verantwoordelijk is om de I²S data te versturen over het netwerk gebruikmakend van het Dante protocol. Voor dit tweede deel wordt gebruik gemaakt van een Dante Ultimo die ondersteuning biedt voor audioproducten met een sample rate van 48 kHz voor een opstelling tot 4 x 4 kanalen [65]. Voor het eerste hardware deel moest er dus gekozen worden welke processor juist gebruikt ging worden.

3.1.1. Vereisten

De vereisten vanuit de R&D-afdeling van Audac zelf waren beperkt. De voornaamste vereiste was dat de stukprijs van de processor goedkoop moest zijn aangezien deze processor toegepast wordt in massaproductie van de wandmodule. Er werd geen voorkeur gegeven aan het type processor zoals FPGA, microcontroller, etc. Aanvullend op de kostprijs was het belangrijk dat de processor USB ondersteunt en over de mogelijkheid beschikt om de data om te vormen en door te sturen bij zeer precieze klokken en de flanken hiervan bij de snelheden die hierbij nodig zijn.

Om te bepalen welke snelheid er nodig is om de data te kunnen versturen en/of ontvangen gaat de berekening als volgt:

$$R = n_K \cdot n_B \cdot n_D \cdot f_s \quad (4)$$

Waarbij dat:

- $R = \text{bitsnelheid [Mbps]}$
- $n_K = \text{aantal kanalen [kanalen]}$
- $n_B = \text{aantal bits per pakket [bits]}$
- $n_D = \text{aantal richtingen [richtingen]}$
- $f_s = \text{sample rate [MHz]}$

Ingevuld voor de toepassing uit deze scriptie met een USB-opstelling waarbij er twee kanalen zijn, waarvan ieder kanaal van 24 audio bits in een pakket van 32-bits zit, de data verstuurd wordt in twee richtingen met een sample rate van 0,048 MHz bekomt men volgende ingevulde formule.

$$R = 2 [\text{kanalen}] \cdot 32 [\text{bits}] \cdot 2 [\text{richtingen}] \cdot 0,048 [\text{MHz}] = 6,144 [\text{Mbps}]$$

De bit rate van USB moet dus minstens een snelheid hebben van 6,144 Mbps. Uit Tabel 1 uit hoofdstuk 2.2.1 kan afgeleid worden dat er een USB-versie nodig is van 1.1 (Full-Speed) of hoger om te voldoen aan deze minimumsnelheid van dataoverdracht.

Andere vereisten waaraan de processor moet voldoen zijn: ondersteuning voor I²S, hoge kloksnelheid om monsters te nemen van het binnenkomende signaal en GPIO pinnen voor diverse doeleinde en mogelijkheden.

3.1.2. Vergelijking

De markt is overladen met allerlei verschillende soorten FPGA's en processoren die voldoen aan de vooropgestelde vereisten. Een categorie die meteen buiten het voornaamste vereiste viel zijn FPGA's. De prijs van een FPGA is aanzienlijk hoger dan de prijs van een microcontroller. Omwille van deze reden zal de vergelijking van processoren focussen op microcontrollers.

3.1.2.1. XMOS xCORE XU208

Een eerste microcontroller processor is de XMOS xCORE XU208. Dit is een 32-bit microcontroller met acht kernen die beschikt over High Speed USB [66]. XMOS is een minder gekend halfgeleider bedrijf dat zich voornamelijk inzet op hoge kwaliteit audioproducten, stem controle en automatisatie oplossingen [67]. Deze heeft de mogelijkheid tot een sample rate van 384 kHz met 32-bits en is stukken hoger dan vereist voor de applicatie in deze paper. Alsook is de stukprijs/1000 14,004 euro wat relatief hoog is. De beschikbaarheid van deze processoren zelf is enorm laag. Figuur 25 toont een PCB waarin de XMOS XU208 geïntegreerd is. Deze biedt ondersteuning voor 32-bit data bij een sample rate van 384 kHz.



Figuur 25: XMOS xCORE XU208 PCB-ontwerp [68]

3.1.2.2. STM32F0x2 en STM32F4

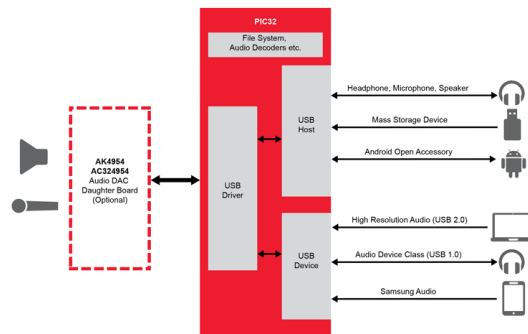
De tweede mogelijkheid zijn verschillende microcontrollers van STMicroelectronics waaronder specifiek STM32F0x2 of STM32F4. STMicroelectronics is een gevestigde waarde op de microcontroller en microprocessor markt met een zeer uitgebreid en uiteenlopend aanbod van microcontrollers. De STM32F0x2 is een 32-bit microcontroller met een enkele kern die beschikt over Full Speed USB [69]. [70] bevat een application note waarin een USB-audio bridge voorbeeld uitgewerkt is aan de hand van deze STM32F0x2 processor. De STM32F401 daarentegen is ook een 32-bit microcontroller met een enkele kern die beschikt over High Speed USB en valt meer onder de hogere kwaliteit microcontrollers [71]. [72] bevat een application note waarbij de STM32F4DISCOVERY gebruikt wordt om audio af te spelen en op te nemen via Flash memory en een USB-opslagapparaat. De beschikbaarheid van STMicroelectronics processoren is echter heel slecht sinds het begin van de pandemie in 2019 die globaal voor een chiptekort heeft gezorgd. De gevolgen hiervan zijn nog steeds duidelijk zichtbaar in de globale voorraad van deze microprocessors. Figuur 26 toont het STM32F4DISCOVERY ontwikkelingsbord van STMicroelectronics.



Figuur 26: STM32F4DISCOVERY [73]

3.1.2.3. PIC32MX7

Een volgende mogelijkheid is de PIC32MX7 van Microchip. Microchip is een bekend merk van microcontrollers met een goede reputatie en een zeer uitgebreid aanbod voor tal van toepassingen. De PIC32MX7 is een 32-bit microcontroller met een enkele kern die beschikt over Full Speed USB [74]. [75] is een application note waarin de PIC32 gebruikt wordt om te verbinden met een USB host en vervolgens data af te spelen of op te nemen via een headset die bestuurd kan worden of de audio kan aansturen door middel van een I²C verbinding. De beschikbaarheid van Microchip processoren is echter heel slecht sinds het begin van de pandemie in 2019 die globaal voor een chiptekort heeft gezorgd. De gevolgen hiervan zijn nog steeds duidelijk zichtbaar in de globale voorraad van deze microprocessors. Figuur 27 illustreert de USB-mogelijkheden die mogelijk zijn met de Microchip PIC32 microprocessors.



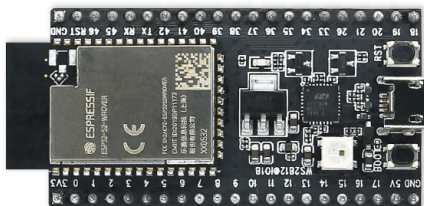
Figuur 27: PIC32 USB-audiomogelijkheden [76]

3.1.2.4. ESP32-S2

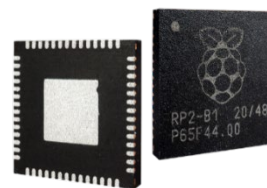
Een alternatieve microcontroller die onderzocht werd is de ESP32-S2 van Espressif. Espressif is een zeer bekend merk van microcontrollers, de ESP32 is een type microcontroller van Espressif die wordt toegepast in zeer veel IoT toepassingen door de veelzijdigheid en het gebruiksgemak. De ESP32-S2 is een krachtige, lage kost 32-bit microcontroller die beschikt over een enkele kern en Full-Speed USB [77]. [78] is documentatie van Espressif waarin het gebruik van de USB Device Driver voor de ESP32-S2 chip wordt uitgelegd. Voor ontwikkeling van USB wordt er gebruik gemaakt van de TinyUSB stack waardoor er veel voorbeelden en mogelijkheden beschikbaar zijn. Figuur 28 toont een ESP32-S2 development kit.

3.1.2.5. RP2040

Een laatste mogelijkheid die onderzocht werd is de RP2040 van Raspberry Pi. Dit is de eerste poging van Raspberry Pi om een microcontroller uit te brengen naast de beter gekende en zeer populaire Raspberry Pi en Raspberry Pi Zero die gebruikt worden als computer. De RP2040 is een hoge prestatie, lage kost 32-bit microcontroller die beschikt over twee kernen en Full Speed USB [57]. [79] is een voorbeeldproject waarin dat de Raspberry Pi Pico, een ontwikkelingsbord waarop de RP2040 staat, gebruikt wordt voor 32-bit stereo te sturen met een sample rate van 44,1 kHz. De beschikbaarheid van deze microcontroller is bij iedere leverancier zeer hoog. Figuur 29 toont het fysieke QFN-56 package waarin de RP2040 microcontroller van Raspberry Pi zit.



Figuur 28: ESP32-S2 DevKitM-1 [80]



Figuur 29: RP2040 in QFN-56 package [55]

3.1.3. Keuze

De vergelijking van voorgaande microcontrollers is samengevat in Bijlage A: Samenvatting vergelijking mogelijke microcontrollers. Hierin zijn de belangrijkste eigenschappen van de microcontrollers opgesomd zoals: aantal kernen, CPU klok, stukprijs, beschikbaarheid etc.

Uit deze vergelijking werd al rap duidelijk dat twee specifieke eigenschappen bij één bepaalde microcontroller veel beter waren dan bij al de rest. De beschikbaarheid en kostprijs van Raspberry Pi RP2040 zijn beter dan bij de andere microcontrollers terwijl deze over dezelfde en/of meer mogelijkheden beschikt. De keuze van microcontroller werd dan ook de RP2040 voor het onderzoek in deze scriptie.

3.1.4. Opstelling

De testopstelling is opgebouwd uit twee Raspberry Pi Picos, die beschikken over een RP2040, in een opstelling zijn opgezet waardoor het mogelijk is om van één van de twee Raspberry Pi Pico de seriële debug uit te lezen en te programmeren aan de hand van de Raspberry Pi Picoprobe firmware. Deze opstelling laat het toe om eenvoudig en snel te debuggen.

Een goede aanpak bij programmeren is het hoofdprobleem opdelen in kleinere deelproblemen en doelen opstellen die later het groter geheel kunnen vormen. De kleinere werkpakketen voor het onderzoek in deze scriptie zijn als volgt en vereisen voor ieder werkpakket een andere opstelling:

1. Implementeer 16-bit I²S interface op de Raspberry Pi Pico.
 - RP2040 genereert de klokken.
 - Enkel als producer van audio/data.
 - Gegeneerde sinus op bordje.
2. Implementeer 32-bit I²S interface op de Raspberry Pi Pico
 - Zelfde als vorig werkpakket maar met 32-bit resolutie.
3. I²S interface aansturen via USB op de Raspberry Pi Pico
 - USB Endpoints configureren (Upstream en downstream).
 - Enkel als producer van audio/data.
4. Implementeer 32-bit I²S slave interface op de Raspberry Pi Pico
 - Externe klokken worden binnengehaald op de RP2040 en worden gevolgd.
 - Gegeneerde sinus op bordje.
 - Producer en consumer van audio/data.

Bovenstaande werkpakketen of doelen kunnen onderverdeeld worden in twee grotere delen, namelijk de opstellingen met interne klokken en externe klokken.

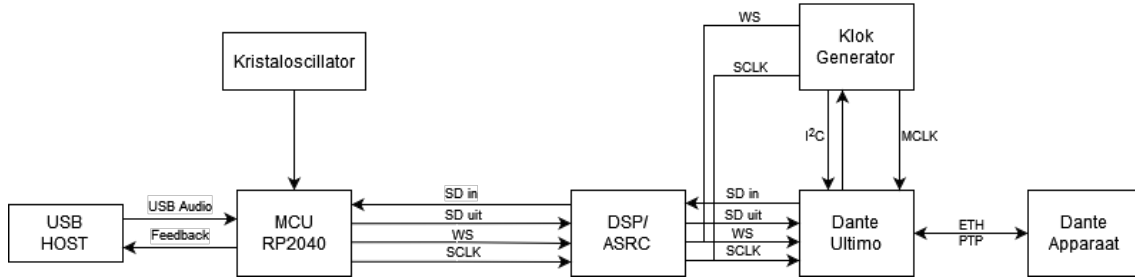
Afhankelijk van de opstelling moet de Dante Ultimo PDK geconfigureerd worden zodanig dat de Ultimo wel of geen klokken uitstuurt en al dan niet klokken binnen neemt. Figuur 30 beschrijft hoe de DIP-switches van S1 op de Dante Ultimo PDK moeten staan zodanig dat er klokken binnen genomen kunnen worden. Aan de hand van deze DIP-switches wordt de CODEC al dan niet geïsoleerd. Door deze DIP-switches te wijzigen kan er dus gewisseld worden tussen het binnen nemen van klokken of het uitsturen van klokken.

Switch #	Input	Output (if switch on)
1	I2S0_RXD1	DOUT_B (to/from codec)
2	I2S0_TDX1	DIN_B (to/from codec)
3	I2S0_RXD0	DOUT_A (to/from codec)
4	I2S0_TXD0	DIN_A (to/from codec)
5	I2C_SDA_CDC	SDA (to/from codec)
6	I2C_SCL_CDC	SCL (to/from codec)
7	nRST_CDC	RESET (to/from codec)
8	NC (Not connected)	NC (Not connected)

Figuur 30: DIP Switches S1 CODEC Isolatie [81, p. 18]

3.1.4.1. Interne klok

In het geval van de opstelling met de interne klok is de Raspberry Pi Pico met RP2040 zelf de master. Hierbij is de RP2040 niet afhankelijk van externe klokken en genereert het alle klokken zelf. Figuur 31 illustreert de schematische voorstelling waarbij de Raspberry Pi Pico gebruik maakt van de interne kristaloscillator voor het genereren van de I²S klok en het uitvoeren van instructies. Hierbij is de Dante Ultimo PDK ook een eigen klok waardoor dat de klokken niet volledig overeenkomen.

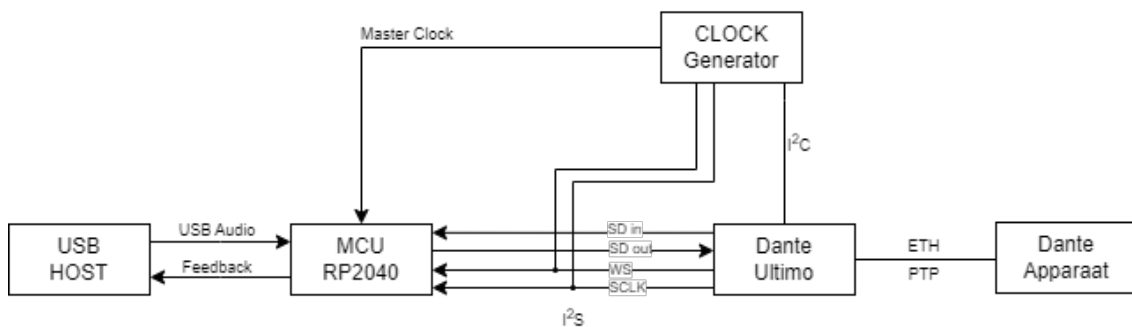


Figuur 31: Testopstelling Raspberry Pi Pico en Dante Ultimo interne klokken

Op Figuur 31 is van links naar rechts geïllustreerd hoe de opstelling juist opgebouwd is. Uiterst links is de USB Host zichtbaar die het USB-audiodata signaal gaat versturen en/of ontvangen. Naar deze USB Host is een asynchrone isochrone dataverbinding type met de Raspberry Pi RP2040 die de kristaloscillator volgt en het USB-audiodata signaal omvormt naar I²S en omgekeerd. Tussen de RP2040 en de Dante Ultimo PDK loopt een data upstream en downstream verbinding waartussen I²S data uitgewisseld wordt. De Dante Ultimo PDK maakt gebruik van eigen gegenereerde I²S klokken, door middel van een ASRC is het mogelijk om deze asynchrone klokken gelijk te krijgen.

3.1.4.2. Externe klok

In een andere opstelling wordt de Raspberry Pi Pico aangestuurd door een externe klok. De Raspberry Pi Pico staat in dit geval in een slave configuratie waarbij deze de externe klok volgt. Het voordeel hiervan is dat klokken doorheen de gehele opstelling gesynchroniseerd zijn wat in het geval van twee afzonderlijke klokken opgelost wordt door een dure ASRC te plaatsen. Figuur 32 illustreert de schematische voorstelling waarbij de Raspberry Pi Pico gebruik maakt van dezelfde klok als die van de Dante Ultimo processor. Hierbij worden de klokken voor I²S binnen genomen bij de Raspberry Pi Pico en wordt op basis van deze klokken bepaald waar de linker- of rechterflank van een audiopakket begint en zodanig dat de bits overeenkomen.



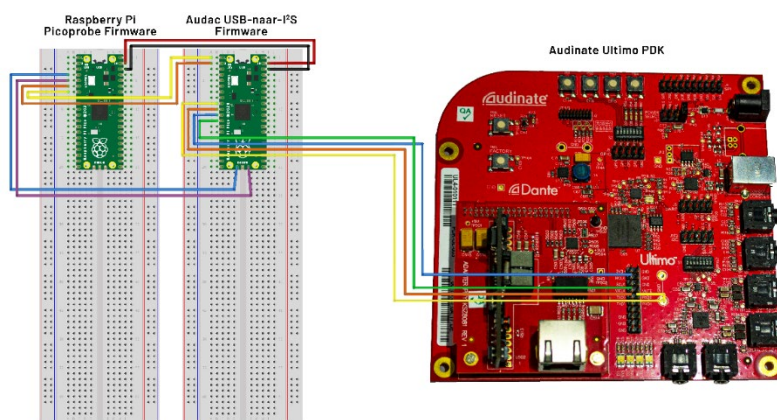
Figuur 32: Testopstelling Raspberry Pi Pico en Dante Ultimo externe klokken

Op Figuur 32 is van links naar rechts geïllustreerd hoe de opstelling juist opgebouwd is. T.o.v. Figuur 31 verandert er niets aan de bekabeling zelf, maar wijzigt wel de manier waarop data zich verplaatst voor de klokken van I²S. Uiterst links is de USB Host zichtbaar die het USB-audiodata signaal gaat versturen en/of ontvangen. Naar deze USB Host is een asynchrone isochrone dataverbinding type met de Raspberry Pi RP2040 die de klok ontvangt van dezelfde klokgenerator als die van de Dante Ultimo processor. Tussen de Raspberry Pi Pico zit de gedeelde klok generator en de I²S verbinding waartussen data uitgewisseld wordt door 'SD in' en 'SD out'. De Dante Ultimo processor is de verbinding naar het Dante netwerk.

3.1.5. Fysieke testopstelling

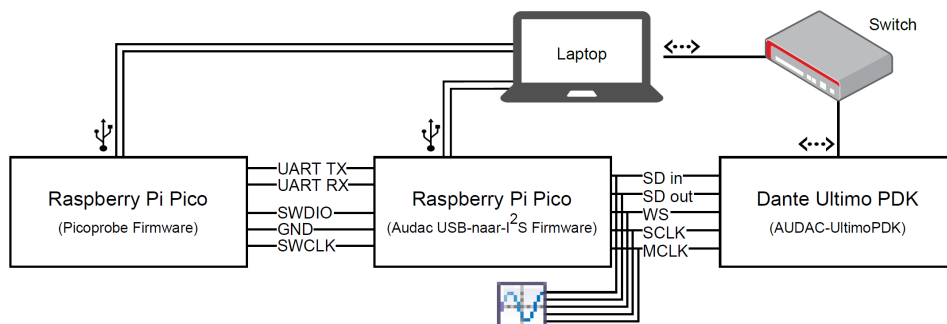
De fysieke verbindingen zijn voorgesteld in Figuur 33. In Figuur 33 staat uiterst links een eerste Raspberry Pi Pico waarop de Picoprobe firmware van Raspberry staat die gebruikt wordt om de tweede Raspberry Pi serieel te debuggen over UART en de tweede Raspberry Pi Pico serieel te programmeren aan de hand van het SWD-protocol. Aan de hand van deze UART TX en UART RX verbinding is het mogelijk om via de laptop een verbinding op te zetten aan de hand van Putty om te monitoren welke informatie er geprint wordt in de tweede Raspberry Pi Pico. Door gebruik te maken van het SWD-protocol is het mogelijk de tweede Raspberry Pi Pico te programmeren door middel van de eerste Raspberry Pi Pico. Nog extra voorzien is het doorverbinden van de grond tussen alle bordjes. Dit zorgt ervoor dat ze allemaal over dezelfde grond beschikken.

Vanuit de tweede Raspberry Pi Pico waarop de Audac USB-naar-I²S firmware staat is er nog een I²S-verbinding naar de Audinate Dante Ultimo PDK. Hierover zal de datacommunicatie van I²S over plaatsvinden alsook de master klok voor de processor van de Raspberry Pi Pico zal via een van deze datalijnen binnenkomen op een GPIO pin die geconfigureerd kan worden voor de Raspberry Pi Pico in de code zelf.



Figuur 33: Testopstelling Raspberry Pi Pico en Dante Ultimo PDK

In Figuur 34 is de schematische voorstelling van de gehele hardware testopstelling te zien. Het onderste deel is nogmaals een kortere samenvatting van de Raspberry Pi Picos in combinatie met de Audinate Dante Ultimo PDK eerder vermeld in Figuur 33. De gehele testopstelling, inclusief de oscilloscoop onderaan die de datalijnen van de I²S-verbinding tussen de Raspberry Pi Pico met Audac firmware en Dante Ultimo PDK monitort. Aan de hand van deze oscilloscoop is het mogelijk naar de datalijnen van I²S te kijken, op deze manier valt het meteen op indien bits niet of slecht binnenkomen. Naar de laptop lopen twee USB-verbindingen voor beide Raspberry Pi Picos en een ethernet verbinding naar een switch toe die op het netwerk zit, die op zijn beurt dan weer verbonden is met de Dante Ultimo PDK. De USB-verbinding naar de Raspberry Pi Pico met Picoprobe is enkel voor serieel te debuggen en serieel te monitoren. De verbinding naar de Raspberry Pi Pico met de Audac firmware dient om de geluidskaart te verbinden en voeding te geven aan de Raspberry Pi Pico, hierover zal dus ook de USB-data verstuurd worden.



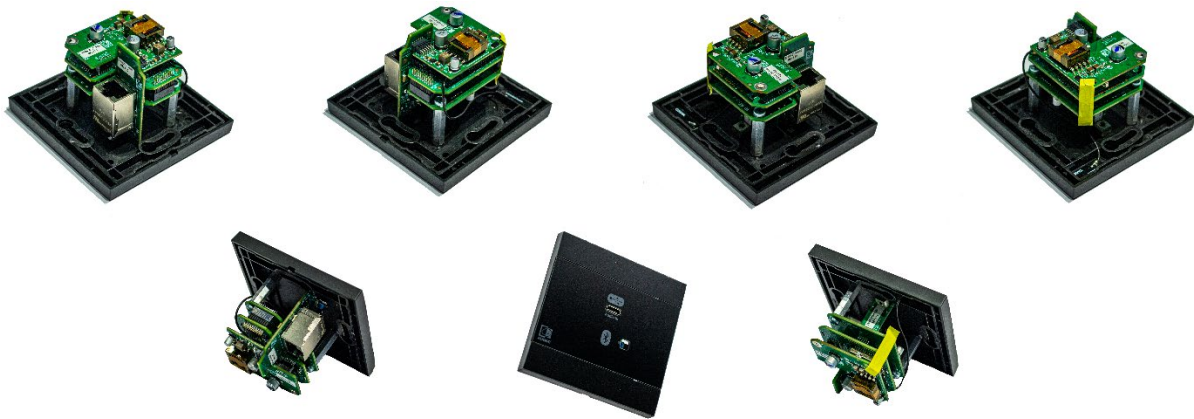
Figuur 34: Schematische voorstelling gehele hardware testopstelling

3.1.6. Elektronisch ontwerp

Voor de schakeling te ontwerpen en het schema hiervoor te tekenen is in deze scriptie gebruik gemaakt van de hardware designregels van de minimumvereisten van een RP2040 in [82]. In dit document werden alle minimumvereisten voor de RP2040 voorgesteld aan de hand van een ontwerp voorbeeld. Alsook het voorgaande Audac NWP400 ontwerp.

3.1.6.1. Audac NWP400 USB/Bluetooth Wandmodule

Het finale ontwerp werd geïntegreerd in de bestaande Audac NWP400 wandmodule waarin het een PCB van de PCB-stack zou vervangen. Figuur 35 toont de verschillende aanzichten van de bestaande Audac NWP400 wandmodule met USB en Bluetooth functionaliteit. De PCB-stack van 5 verschillende PCB's is hierop zichtbaar. In Bijlage B: Wandmodule PCB-stack staan de verschillende PCB's afgebeeld langs een zijde.



Figuur 35: Verschillende aanzichten Audac NWP400 USB/Bluetooth wandmodule

Voor de PCB's in Bijlage B: Wandmodule PCB-stack komt de letter A overeen met het PCB die verantwoordelijk is voor de binnenkomende spanning van de PoE, die ongeveer 48 volt is, om te zetten naar verschillende andere waarden zoals 5.5, 3.3 en 1.8 volt die verder in het schema worden gebruikt.

Het PCB aangeduid met letter B in Bijlage B: Wandmodule PCB-stack komt overeen met het PCB waarop de Dante Ultimo chip, ESP32, Ethernet PHY-controller en een DSP staan. De Dante Ultimo chip heeft een netwerkverbinding via de PoE connector en staat in voor de Dante netwerkverbinding om muziek af te spelen. De ESP32 microcontroller is de master microcontroller doorheen de PCB-stack en is verantwoordelijk voor het gehele overzicht, het aansturen van de I²S interface en de Bluetooth communicatie. De Ethernet PHY verziet de communicatie aan de hand van RMI-signalen. De DSP, die aangestuurd wordt over SPI, stuurt het I²S-signaal aan en moduleert dit naargelang de instructies op de SPI bus.

Het PCB aangeduid met letter C in Bijlage B: Wandmodule PCB-stack is het PCB die gebruik maakt van de verouderde PCM2707 IC die de omzetting van USB naar I²S voorziet en meteen ook het PCB waarvoor dit onderzoek opgestart is. Gesoldeerd aan deze PCB is nog een andere PCB waarop de USB Type C-connector staat voor de USB-audiodata.

Het PCB aangeduid met letter D aangeduid in Bijlage B: Wandmodule PCB-stack is het PCB waar de RJ45-PoE connector opstaat. Alsook de connectoren die de verschillende PCB's onderling verbinden met elkaar voor bijvoorbeeld SPI-verbinding.

De opbouw van de PCB-stack van de Audac NWP400 wandmodule was belangrijk voor de aanpassing van het PCB met letter C in Bijlage B: Wandmodule PCB-stack voor het onderzoek in deze scriptie. De schema's en PCB van het PCB layout moesten volledig vervangen worden met een nieuwe langetermijnoplossing. Voor het ontwerpen van de schema's en PCB wordt in deze scriptie gebruik gemaakt van Altium Designer versie 22.4.2.

3.1.6.2. Altium Designer

Altium Designer is een softwarepakket voor het ontwerpen van elektronische schema's en PCB's en wordt zeer veel gebruikt in de industrie voor deze toepassingen. Altium Designer is een relatief eenvoudig programma maar beschikt over talrijke geavanceerde handelingen voor geavanceerde gebruikers.

De componenten gebruikt in het schema voor de USB-naar-I²S converter zijn afkomstig uit de componenten bibliotheek van Audac zelf. De R&D-afdeling van Audac beschikt over een zelfgemaakte, uitgebreide bibliotheek waarin alle componenten staan die ze op de afdeling gebruiken of gebruikt hebben.

3.1.6.3. Elektronisch schema

Het ontwerp van het elektronische schema is uitgevoerd in het Altium Designer softwarepakket. Screenshots van de schema's zijn terug te vinden in Bijlage C: Topschema uit Altium Designer en Bijlage D: USB naar I2S schema in Altium Designer. De eerste screenshot in Bijlage C: Topschema uit Altium Designer toont het topschema waarin de verschillende connectoren naar andere componenten of verbindingen terug te vinden zijn.

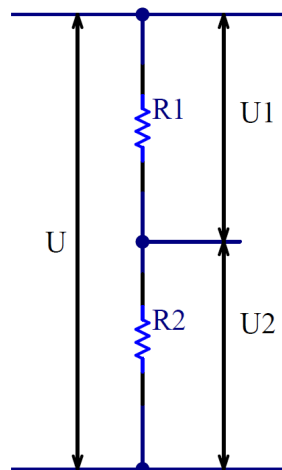
De tweede screenshot in Bijlage D: USB naar I2S schema in Altium Designer toont het werkelijke ontwerp waar de conversie van USB-naar-I²S plaats vindt voor bi-directionele, hoge resolutie audio. Op dit schema staan linksboven de verbindingen die naar of van het topschema komen en verbindingen vormen naar andere PCB's. Aan de linkerzijde in het midden staat het 16 Mbit NOR-flash geheugen dat beschikt over QSPI en XIP. Iets lager aan de linkerzijde staat een schakelaar die het mogelijk maakt de RP2040 te resetten alsook een spanningsdeler die aangeeft indien de 5 volt van een USB-bus aanwezig is en dan verlaagt naar een spanning van 3.3 volt. Het berekenen van een spanningsdeler, zoals geïllustreerd in Figuur 36, gaat door volgende formule te gebruiken:

$$U_2 = U \cdot \frac{R_2}{R_1 + R_2} \quad (5)$$

In dit geval waren U en U_2 gekend, maar R_1 en R_2 niet. Wanneer deze waardes dan worden ingevuld bekomen we volgende formule:

$$3.3 \text{ [volt]} = 5 \text{ [volt]} \cdot \frac{R_2}{R_1 + R_2}$$

Dit omgevormd naar R_1 en R_2 lost op naar: $R_1 = 0.515 \cdot k$ en $R_2 = k$ waarbij dat $k \in \mathbb{N}_0$, als we dan voor R_2 een willekeurige veelvoorkomende waarde nemen zoals bijvoorbeeld 7,5 k Ω bekomen we voor R_1 een waarde van 3863,64 Ω . De E12-waarde die hier het dichtste bij aan sloot was 3900 Ω .



Figuur 36: Opstelling spanningsdeler

In het midden bovenaan van Bijlage D: USB naar I2S schema in Altium Designer staat een opstelling met een kristaloscillator van 12 MHz. Het midden van de pagina toont de RP2040 met alle in- en uitgangen. De footprint en schematische voorstelling van de RP2040 waren niet aanwezig in de R&D-afdeling bibliotheek. Door de footprint en het schemasymbool te ontwerpen in Altium Designer en toe te voegen aan de uitgebreide bibliotheek van Audac is de footprint en schematische voorstelling nu beschikbaar om te gebruiken in nieuwe ontwerpen. Onder de in- en uitgangen van de RP2040 staat het spanningsdeel van de RP2040. Ook aan de benamingen “U1A” en “U1B” is af te lezen dat deze twee schema’s samen horen. Om het overzicht te bewaren wordt het spanningsgedeelte losgekoppeld van de in- en uitgangen.

De rechterzijde van de pagina toont twee connectoren die gebruikt kunnen worden om I²C, SPI, I²S of SWD te gebruiken en te debuggen. Dit vereenvoudigt het testen wanneer het PCB ontworpen, gefabriceerd is. Aan de hand van dit elektronisch schema konden de componenten geëxporteerd worden naar het PCB-ontwerp software deel van Altium Designer om het ontwerpen van het PCB te starten.

3.1.7. PCB

Om het ontwerpen van het PCB te starten wordt het elektronisch schema, met de correct ingestelde footprints, geïmporteerd naar een PCB-document voor Altium Designer. Hierdoor komen alle componenten in het PCB-document te staan waarna dat de ontwerper kan beginnen aan het ontwerpen van het PCB door de componenten te verplaatsen en te verbinden via baantjes en via’s.

3.1.7.1. Ontwerpregels

Om tot een goed ontwerp te komen zijn er talloze ontwerpregels die in Altium Designer zelf zitten maar ook vanuit de fabrikant worden opgelegd. [83] is een deel van de website van PCB-productiebedrijf genaamd Eurocircuits waarop de ontwerpregels binnen hun bedrijf staan. Om hier te kunnen bestellen moeten er aan deze vereisten voldaan worden. Enkele van deze regels hebben betrekking op het boren van gaten, soldeer masker, legende printen, etc.

3.1.7.2. Ontwerp

Voor het ontwerp zelf zijn de componenten eerst gerangschikt zodat deze ongeveer overeen komen met het schema. Alsook is het kost efficiënt om alle componenten aan een zijde te houden en hier rekening mee te houden. Vervolgens zijn de verbindingen tussen alle componenten op een zo efficiënt mogelijke manier verbonden. Dit betekent rekening houdend met zo weinig mogelijk via’s en zo kort mogelijke baantjes. De datalijnen voor USB moeten ook een gelijke lengte hebben voor de beste data-overdracht. Dit was in te stellen via Altium Designer door gebruik te maken van de differentiële paren eigenschap.

Het ontwerp van iedere laag in het PCB staan geïllustreerd in Bijlage E: PCB-lagen. Hierop is te zien dat ondanks dat alles best op een kant staat er toch weerstanden en condensatoren aan de achterzijde van het PCB zijn gezet. Een opmerkelijk detail bij de grond en 3.3 volt laag is dat deze opgedeeld is. Hierdoor kan vermeden worden dat signalen op de routingslaag erboven/onder een onderbreking zien in hun referentielaag. De stackup van het PCB is zichtbaar in Figuur 37. Hierin zijn de 4 verschillende signaal lagen te zien omringd met diëlektrische lagen, soldeer maskers en goud afwerking.

#	Name	Material	Type	Weight	Thickness	Dk	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask		0.01mm	3.5	
	Top Surface Fin...	Nickel_Gold	Surface Finish		0.04mm		
1	Top Layer		Signal	Toz	0.035mm		
	Dielectric 1	PP-016	Prepreg		0.11684mm	4.4	0.02
2	Layer 2		Signal	Toz	0.035mm		
	Dielectric 2	Core-038	Core		0.6096mm	4.6	0.02
	Dielectric 3	Core-038	Core		0.6096mm	4.6	0.02
3	Layer 3		Signal	Toz	0.035mm		
	Dielectric 4	PP-016	Prepreg		0.11684mm	4.4	0.02
4	Bottom Layer		Signal	Toz	0.035mm		
	Bottom Surface...	Nickel_Gold	Surface Finish		0.04mm		
	Bottom Solder	Solder Resist	Solder Mask		0.01mm	3.5	
	Bottom Overlay		Overlay				

Figuur 37: PCB stackup

Nadat het PCB-ontwerp klaar is, ondergaat deze nog een laatste controle in Altium Designer zelf via een Design Rule Check. Indien deze geen fouten meer aangeeft kan het ontwerp geëxporteerd worden.

3.1.7.3. Componenten

De componenten worden bij het exporteren geëxporteerd in een Excel-bestand waarin deze opgesomd worden. Bijlage F: BOM NWP400 USB naar I2S toont de BOM. Aan de hand van deze lijst kunnen de juiste componenten besteld worden.

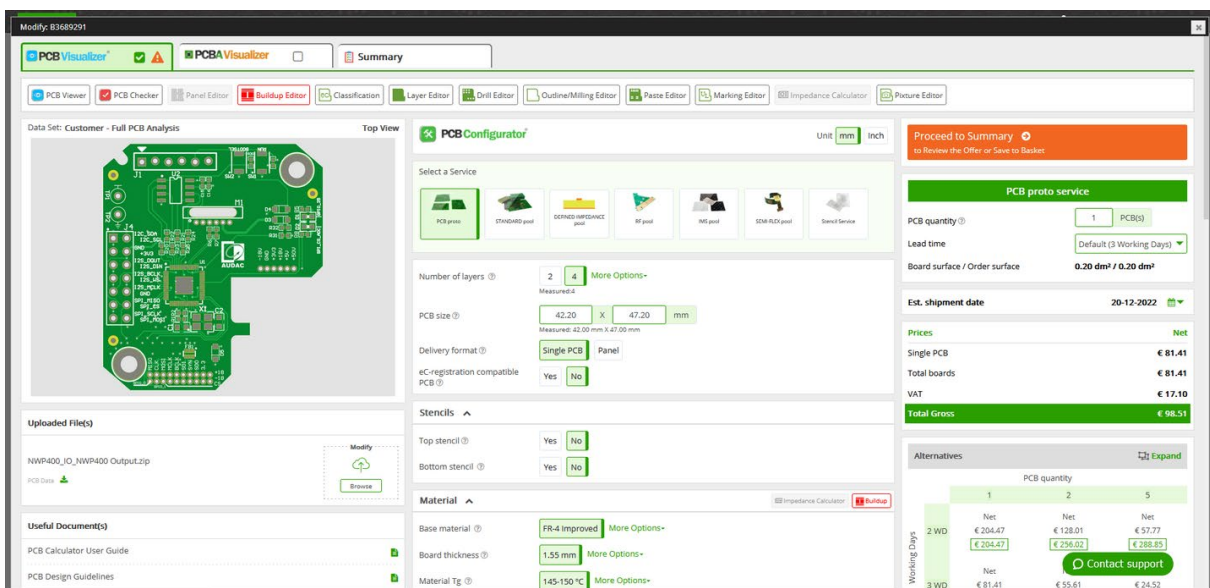
De resterende bestanden uit de genereerde zip bij het exporteren van het project worden gebruikt voor de fabricage van het PCB. De resterende bestanden zijn Gerberbestanden en boorbesteden. De Gerber bestanden zijn de voorstelling van de verschillende koperlagen en tussenlagen, het bestand wordt gebruikt voor het vervaardigen van het PCB. De boorbesteden bevatten de locaties en diameters van waar de gaten in het PCB geboord moeten worden.

3.1.7.4. Bestellen

Wanneer het ZIP-bestand geüpload wordt bij Eurocircuits is het mogelijk om een voorbeeld en schatting van het PCB te zien. Figuur 38 toont de website van Eurocircuits na het uploaden van het geëxporteerd ZIP-bestand uit Altium Designer.

Op deze site kan het PCB geconfigureerd worden naar verlangen. Het voornaamste is dat de bestanden nog gecontroleerd worden of het PCB al dan niet aan de regels voldoet van de producent zelf. Indien er problemen zijn zullen die in deze stap zichtbaar worden.

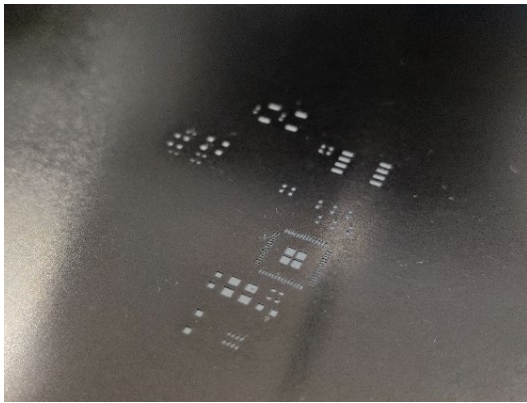
Gelijktijdig is er aan de rechterzijde van de figuur een schatting te zien van de kostprijs. Het prijsverschil per stuk is groot wanneer er een kleine oplage is of een grote oplage. Er worden namelijk altijd opstartkosten geteld voor het machinepark. Als deze kost dan gedeeld kan worden door meerdere PCB's daalt de stukprijs aanzienlijk aangezien de materiaalkost vrij laag is.



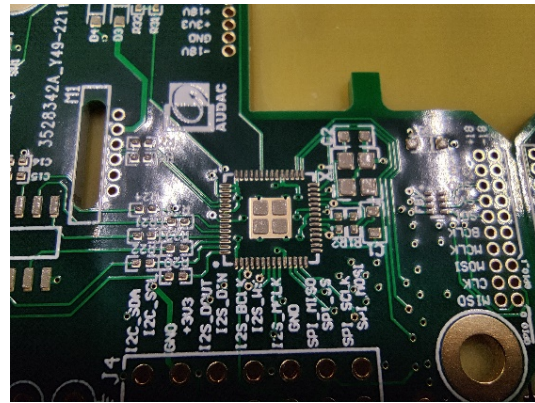
Figuur 38: Eurocircuits PCB bestellen overzicht

3.1.7.5. Soldeer pasta aanbrengen

Vooraleer dat de componenten geplaatst worden op het PCB moet er soldeer pasta gedeponerd worden op de soldeer maskers aangezien er gebruik gemaakt wordt van een oven om het tin te laten vloeien. Door middel van een stencil, een negatief van het soldeer masker, is het mogelijk om soldeer pasta op het soldeer masker te plaatsen. Figuur 39 toont de stencil die gebruikt is om soldeer pasta te deponeren op het PCB. Figuur 40 toont het PCB nadat de soldeer paste erop gedeponerd is door middel van de stencil.

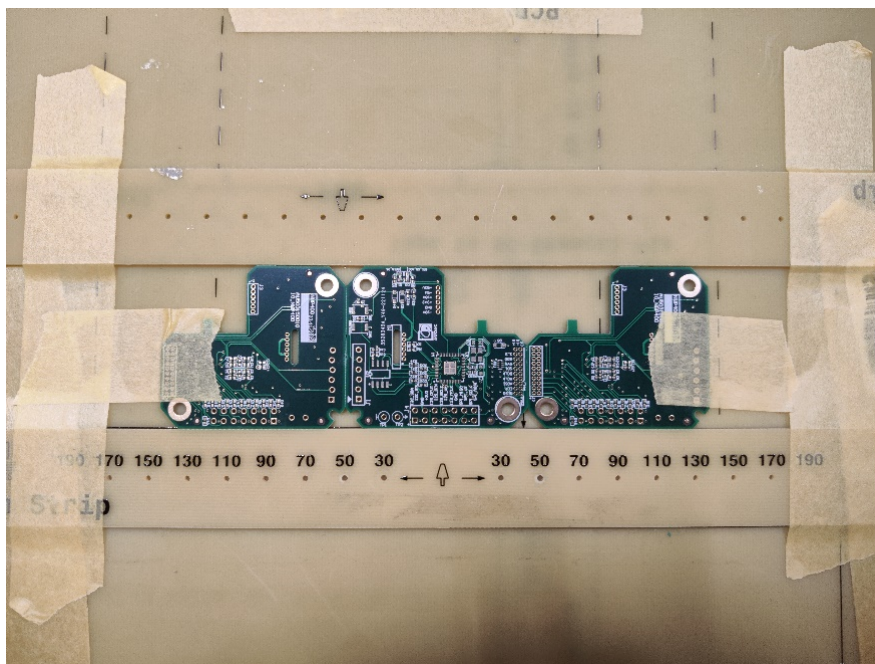


Figuur 39: Stencil bovenkant USB-naar-I²S PCB



Figuur 40: Soldeerpasta op soldeer masker van USB-naar-I²S PCB

Soldeerpasta aanbrengen via een stencil is een zeer nauwkeurig proces. De stencil en het soldeer masker van het PCB moeten perfect overeenkomen zodanig dat de soldeerpasta op de correcte plekken gedeponerd worden. Figuur 41 toont de opstelling waarmee de lege printplaat vastgehouden werd. Het PCB zit vastgeklemd tussen twee andere PCB's en twee latten. Zo is het zeker dat het PCB niet beweegt terwijl de soldeerpasta aangebracht wordt. Op deze opstelling wordt het soldeer masker nog aangebracht met plakband zodanig dat deze niet beweegt. Vervolgens wordt er door middel van een truweel waar soldeerpasta op zit over het soldeer masker gestreken. Zodanig worden de gaten van het stencil gevuld met soldeerpasta terwijl dat de overmaat aan soldeertin mee wordt genomen door het truweel. Het resultaat van dit proces is zichtbaar in Figuur 40.



Figuur 41: Opstelling aanbrengen soldeerpasta

3.1.7.6. Componenten plaatsen

Vervolgens moeten de componenten geplaatst worden op het PCB. De eenvoudigste manier is om deze manueel te plaatsen. Op industrieel niveau wordt dit voornamelijk uitgevoerd door een pick-and-place machine die de componenten zal plaatsen aan de hand van gegenereerde code uit Altium. In dit onderzoek werden de componenten manueel geplaatst. Door middel van het BOM-bestand en het PCB-document in Altium Designer is het mogelijk een component aan te klikken in de BOM waardoor vervolgens het component gearceerd wordt op het PCB document. Dit is een lang, tijdrovend proces waarbij enige precisie nodig is om de kleine componenten te plaatsen. Figuur 42 toont het PCB waarop de soldeerpasta gedeponerd is en de componenten manueel op geplaatst zijn.

3.1.7.7. Soldeeroven

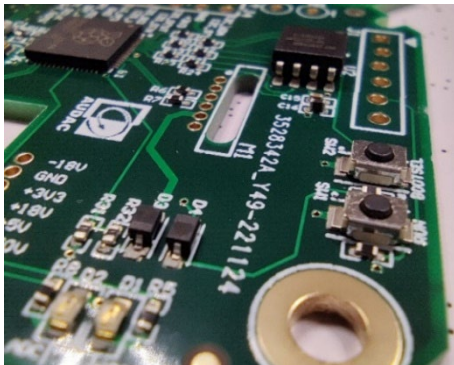
Nadat zowel de soldeer pasta en de componenten geplaatst zijn is de bestukte PCB klaar om geplaatst te worden in de soldeeroven. De oven bij PVS is een vapor phase soldeeroven, dat is een soort soldeeroven waarin PCB's met soldeer pasta en componenten geplaatst kunnen worden in een ruimte met een speciaal gas. Hierin zal het tin van de soldeer pasta beginnen vloeien en de componenten vast maken aan de soldeer maskers op het PCB.

De vapor phase soldeeroven is een speciale soldeeroven en maakt gebruik van een speciaal inerte vloeistof die verwarmd wordt. Het gas van deze vloeistof satureert de atmosfeer rond het PCB door het koken van de vloeistof. Dit resulteert in een hoog efficiënt en egaal soldeersel van het product [84]. Enkele voordelen van deze technologie zijn:

- Hoog efficiënte, gelijkmatige transmissie van hitte naar het PCB doorheen het hele product onafhankelijk van de vorm of grootte [85].
- Nauwkeurig regelen van de temperatuur [85].
- Door de inerte vloeistof waarin er gesoldeerd wordt is er een minimale vorming van oxidatie of vervuiling van gesoldeerde delen [85].
- Een korte verwerkingstijd bij een relatief lage temperatuur. Waardoor de thermische stress tot een minimum gehouden wordt [85].

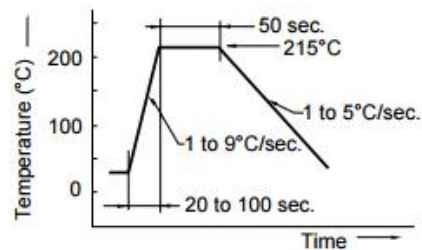
De enige nadelen aan de vapor phase soldeeroven is de hoge kost van de inerte vloeistof genaamd Galden LS230 waar men voor een fles van 500 ml een prijs betaald van € 229,00. Alsook dat het een vrij manueel intensief proces is waardoor het in productie minder of zelfs niet gebruikt wordt [86].

Figuur 43 toont een typisch temperatuurprofiel voor een vapor phase soldeeroven. Waarbij de korte termijn van 50 seconden aan de maximale temperatuur van 215 °C duidelijk aangeeft dat het over een korte verwerkingstijd gaat.



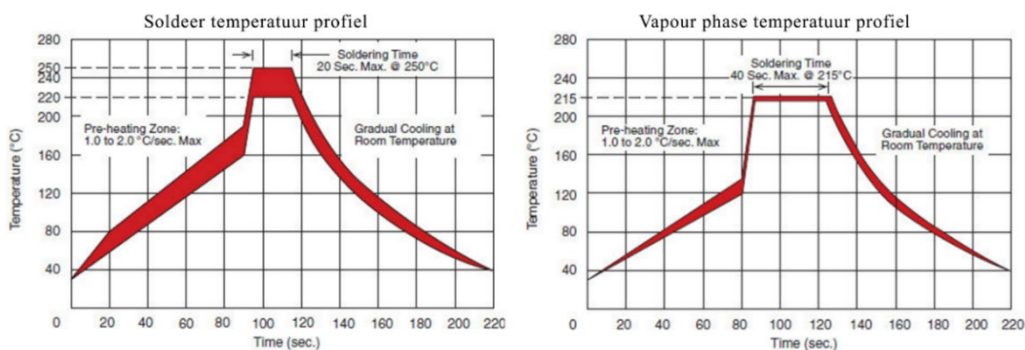
Figuur 42: Componenten op soldeer pasta geplaatst

Vapor Phase - Reflow



Figuur 43: Vapor phase soldeeroven temperatuur profiel [87]

Figuur 44 toont twee grafieken. De linkse grafiek illustreert het temperatuurprofiel bij een convectieoven. Waartegen de rechtse grafiek het temperatuurprofiel voor een vapor phase soldeeroven illustreert.

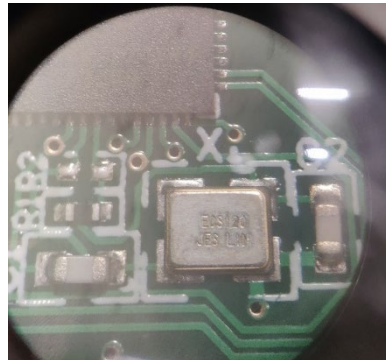


Figuur 44: Soldeer temperatuur profiel en vapor phase soldeeroven temperatuur profiel

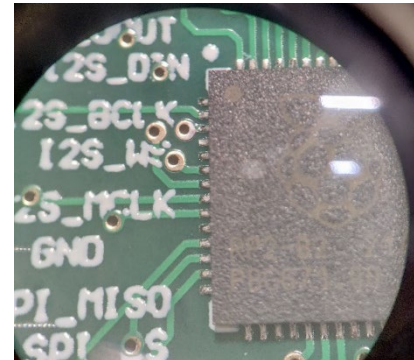
De vapor phase soldeeroven bij PVS is de Imdes Jumbo Condens-IT en wordt enkel gebruikt wanneer er testoplages worden gemaakt van PCB's en dus niet bij productie van grotere aantallen. Figuur 45 toont de behuizing van de Imdes Jumbo Condens-IT vapor phase soldeeroven. Figuur 46 toont het oscillator gedeelte van het PCB door een microscoop. Hierop is te zien dat de tin mooi gevloeid is op de soldeer maskers en de componenten. Figuur 47 is de Raspberry Pi RP2040 chip. Dit toont de kleinere verbindingen van de QFN-56 die gevloeid zijn tot de soldeer maskers. De hechtingen naar de soldeer masker gaan ook eenvoudiger door het gebruik van goud op de soldeer maskers. Bovendien zorgt dit voor een anti-corrosie beschermingslaag.



Figuur 45: Imdes Jumbo Condens IT



Figuur 46: Close-up oscillator



Figuur 47: Close-up RP2040

3.2. Software

De software ontwikkeld voor deze scriptie bestaat uit een programma geschreven in programmeertalen C en C++. Het gebruik van de C-programmeertaal maakt het mogelijk om op een laag level te programmeren. Om de gekozen microcontroller te debuggen en te programmeren wordt er gebruik gemaakt van OpenOCD in een configuratie met een tweede Raspberry Pi Pico waarop Raspberry Pi Picoprobe geprogrammeerd is. Dit maakt het mogelijk de NOR-flash van de Audac USB-naar-I²S te programmeren over een driedraads SWD-verbinding en serieel te monitoren over tweedraads UART. De code zelf werd geschreven in Visual Studio Code waarin de OpenOCD en C/C++ format extensies geactiveerd waren.

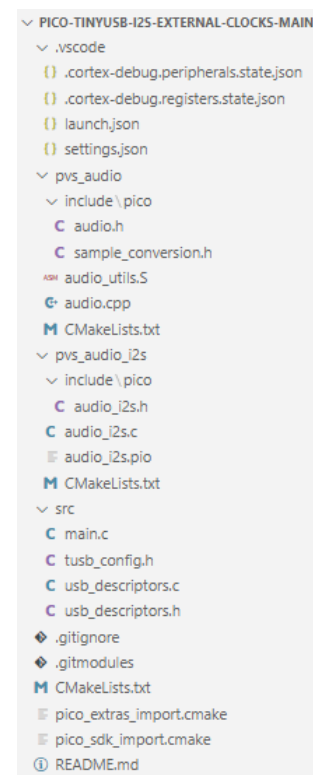
De mappen en de bestanden die deel uitmaken van de firmware zijn geïllustreerd in Figuur 48. De map genaamd: '.vscode' bevat de bestanden die betrekking hebben op het opstarten van de processor en de configuratie voor OpenOCD waardoor de code die gestart wordt doorstapt en gepauzeerd kan worden.

De map genaamd: 'pvs_audio' bevat verschillende functies die gebruikt worden om verschillende sample rates te converteren zodat het mogelijk is om bijvoorbeeld van 32-bit data naar 8-bit te gaan.

De volgende map genaamd: 'pvs_audio_i2s' bevat de bestanden voor het versturen en ontvangen van I²S data. Alsook het PIO-bestand voor de Raspberry Pi RP2040 om te synchroniseren met binnenkomende klokken, de registers in of uit te schuiven via registers afhankelijk of er data wordt ontvangen of verstuurd.

De laatste map genaamd: 'src' bevat het bestand waarin het hoofdprogramma zich bevindt. Aanvullend bevat het nog de TinyUSB bestanden voor de configuratie en de descriptor die de omschrijving van USB beschrijft.

Verder staan er nog een 'CMakeLists.txt' bestanden waarin gedefinieerd wordt waar dat de algemene bibliotheek, bord bestanden van Raspberry Pi Pico en algemene definities gevonden kunnen worden voor het uitvoeren van het hoofdprogramma.



Figuur 48: NWP400 firmware bestanden

3.2.1. Overzicht

Het algemene programma kan opgedeeld worden in twee verschillende aspecten zijnde: USB en I²S. Beide zijn seriële protocollen waarover dat audiodata in twee richtingen verstuurd wordt met een bit depth van 24-bits in een datapakket van 32-bits en een sample rate van 48 000 Hz.

Voor zowel USB als I²S worden er twee DMA-kanalen aangemaakt voor de inkomende en uitgaande data. De DMA-kanalen bestaan uit circulaire/ring buffers om efficiënt om te gaan met het geheugen.

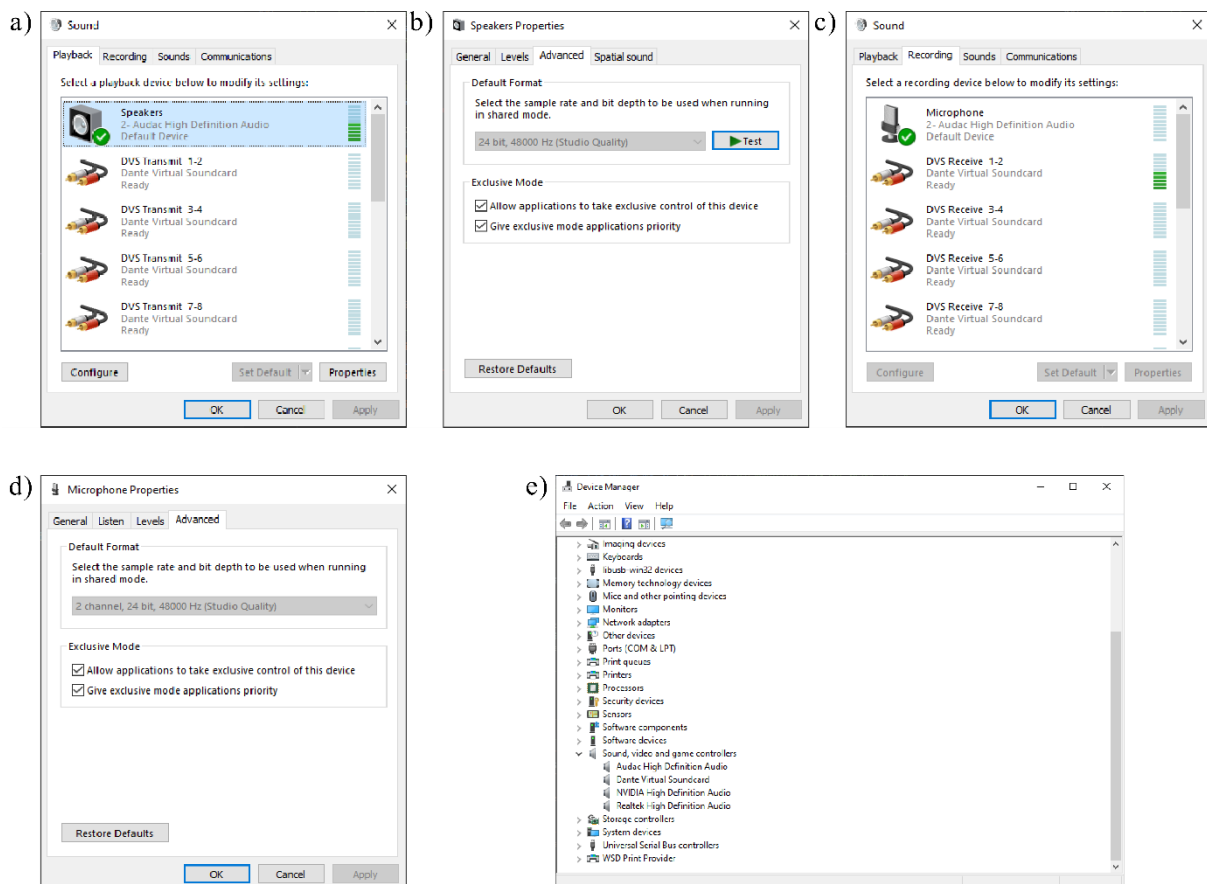
3.2.2. USB

De USB-verbinding wordt gebruikt als een geluidskaart. Een apparaat kan, door middel van een verbinding met de USB-connector, geluid afspelen en opnemen. Om een USB-verbinding te initialiseren wordt er gebruik gemaakt van de TinyUSB open-source bibliotheek [88]. TinyUSB is een open-source USB Host/apparaat stack voor embedded systemen.

3.2.2.1. Configuratie

Via het bestand genaamd 'tusb_config.h' wordt beschreven welke data er juist over de datalijnen gestuurd wordt en in welke grootteorde. Voor toepassing in deze scriptie worden er twee afzonderlijke endpoints aangemaakt. Beide endpoints hebben dezelfde configuratie aan een resolutie van 24-bits in een datapakket van vier bytes waarbij de data in een big endian configuratie is en verstuurd wordt via isochronous transfer. Het verschil in de endpoints zit in het ontvangen of versturen van data waarbij de 'TX' endpoint data verstuurt en de 'RX' endpoint data ontvangt. In het config bestand is ook een link naar 'usb_descriptor.h' waarin de descriptor omschreven wordt die aan apparaten zal duidelijk maken wat voor apparaat het USB-apparaat is.

Figuur 49 toont verschillende screenshots uit de geluidscontrole applicatie en apparaatbeheer applicatie uit Windows. Deel a en b uit Figuur 49 toont de USB-speaker (uitgaande audiodata), deel c en d toont de USB-microfoon (inkomende audiodata) en deel c toont tenslotte hoe het apparaat herkend wordt in het apparaatbeheer in Windows.



Figuur 49: USB Descriptors NWP400 in Windows apparaatbeheer

3.2.2.2. Descriptor

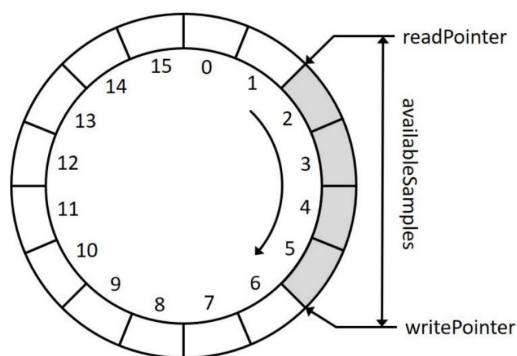
In het 'usb_descriptor.h' bestand staat de USB-descriptor aangegeven. Een USB-descriptor is een datastructuur dat informatie geeft over het USB-apparaat of de interface naar het gebruikerssysteem. Het beschrijft de mogelijkheden van het apparaat, de apparaat klasse, verkoper en productidentificatie en de types van geheugen gebruikt worden.

Er zijn verschillende soorten USB descriptors zoals: apparaat descriptors, configuratiedescriptors en interface descriptors. Een apparaat descriptor voorziet generieke informatie over het apparaat zoals de verkoper en product identificaties, de apparaatklasse en subklasse, en het aantal configuraties die ondersteund worden. Een configuratiedescriptor voorziet informatie over een specifieke configuratie van het apparaat, inclusief het aantal interfaces die ondersteund worden en aan de hoeveelheid vermogen die nodig is. Een interface descriptor voorziet informatie over een specifieke interface van het apparaat inclusief het aantal endpoints die gebruikt worden en de interface zijn klasse en subklasse.

USB descriptors worden voornamelijk opgeslagen in de firmware van apparaten en worden door het hoofdbesturingssysteem aangesproken wanneer een apparaat verbonden wordt. De host gebruikt de informatie uit de descriptor om te bepalen hoe deze moet communiceren met het apparaat en hoe het geheugen moet alloceren.

3.2.2.3. Buffer

Voor het bufferen van de inkomende en uitgaande data wordt er gebruik gemaakt van een circulaire buffer waarvan een illustratie zichtbaar is in Figuur 50. Een circulaire buffer is ook wel gekend als een circulaire queue of ring buffer en is een datastructuur dat gebruik maakt van een enkele vaste grootte buffer waarbij het begin en einde 'verbonden' zijn. Deze structuur laat het toe om voortdurend data te schrijven en lezen van deze enkele buffer waarbij oudere data overschreven wordt wanneer de buffer vol is en er nieuwe data is. De illustratie in Figuur 50 toont een eenvoudige circulaire buffer met 16 plaatsen waarin data opgeslagen kan worden. De schrijfpunter staat momenteel bij locatie 6 terwijl dat de leespointer staat bij locatie 1. Tussen deze locaties staat de binnengekomen data.



Figuur 50: Circulaire buffer [89, p. 13]

Een circulaire buffer wordt vaak gebruikt wanneer gegevens sneller worden gegenereerd of ontvangen dan ze kunnen worden verwerkt, of wanneer gegevens tijdelijk moeten worden opgeslagen voor latere verwerking. Het is een efficiënte manier om streaming gegevens te verwerken, omdat de gegevens niet hoeven te worden gekopieerd of verplaatst om plaats te maken voor nieuwe gegevens.

Om een circulaire buffer te implementeren, worden een start- en eindpointer gebruikt om de locatie van de oudste en nieuwste gegevens in de buffer bij te houden. Wanneer gegevens aan de buffer worden toegevoegd, wordt de eindpointer verhoogd en worden de gegevens geschreven naar de locatie waarnaar de eindpointer verwijst. Wanneer gegevens uit de buffer worden verwijderd, wordt de startpointer verhoogd en worden de gegevens op de door de startpointer aangegeven plaats verwijderd. Wanneer de eindpointer het einde van de buffer bereikt, loopt deze om naar het begin van de buffer, en wanneer de startpointer het einde van de buffer bereikt, loopt deze ook om. Hierdoor kan de buffer continu worden gebruikt zonder dat de pointers opnieuw moeten worden ingesteld of de gegevens moeten worden verplaatst.

3.2.3. I²S

De I²S data wordt, net zoals USB-data, in twee richtingen verstuurd voor 24-bits bij een frequentie van 48 000 Hz waarbij de data in een 32-bits data frame zit volgens het big endian formaat. De I²S data transporteert van en naar de Dante Ultimo chip die aanwezig is op een andere PCB in de PCB-stack.

De Dante ultimo chip wordt gebruikt om de I²S data te streamen of ontvangen over een Dante netwerk. Deze chip wordt dan gedetecteerd in Dante netwerk als een apparaat waarvan gestreamd kan worden of naar gestreamd kan worden. De naam van dit apparaat kan naar verlangen gewijzigd worden.

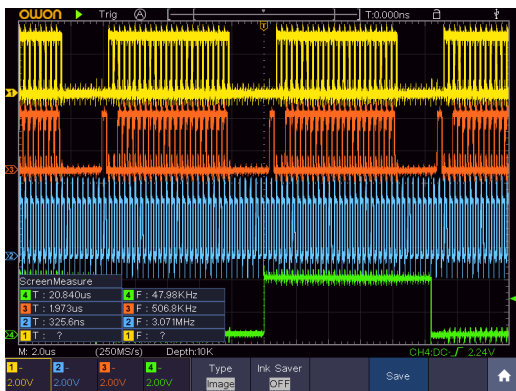
De opslag van de binnenkomende en uitgaande data gebeurt, net zoals bij USB, in circulaire buffers zoals besproken in 3.2.2.3 Buffer.

3.2.3.1. Configuratie

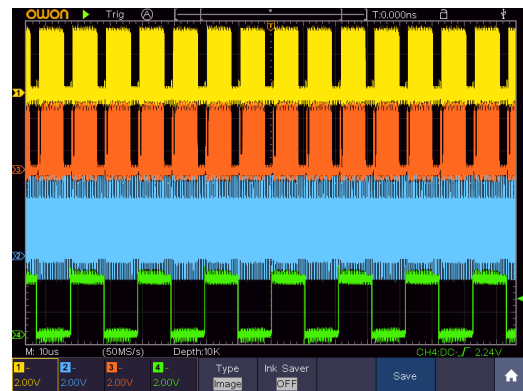
Via het bestand 'audio_i2s.h' wordt beschreven hoe de data juist opgebouwd is en op welke GPIO pinnen de verschillende datalijnen zitten voor I²S. Alsook het aantal bits en ondersteunende sample rates. Hierin staan ook de tijdelijke buffers die door worden gestuurd naar de circulaire buffer. Volgende struct toont de configuratie die ingesteld kan worden voor verschillende I²S toepassingen en configuraties.

```
typedef struct audio_i2s_config {  
  
    uint32_t fs;  
    uint32_t sck_mult;  
    uint8_t bit_depth;  
    uint8_t sck_pin;  
    uint8_t dout_pin;  
    uint8_t din_pin;  
    uint8_t clock_pin_base;  
    uint8_t dma_channel;  
    uint8_t pio_sm;  
    bool sck_enable;  
} audio_i2s_config_t;
```

In ditzelfde bestand zitten ook de functies voor het starten en stoppen van de I²S connecties voor zowel in master als slave configuratie. De functie om data te alloceren voor de twee DMA-buffers om de data te ontvangen en te versturen bevindt zich in ditzelfde bestand. Figuur 51 en Figuur 52 tonen de datalijnen voor I²S wanneer deze gemonitord worden op een oscilloscoop. Figuur 51 toont de lijnen waarbij, in volgorde van boven naar onder, eerst in het geel de inkomende datalijn zichtbaar is, vervolgens in het oranje de uitgaande datalijn, ten derde in het blauw de bitklok en als laatste in het groen de links- rechtsklok.



Figuur 51: I²S-lijnen op oscilloscoop



Figuur 52: I²S-lijnen op oscilloscoop groter bereik

3.2.3.2. PIO

Het PIO-programma wordt gebruikt voor het gehele transport I²S-gedeelte van de scriptie. Aan de hand van dit PIO-programma is het namelijk mogelijk om data te ontvangen en versturen door te wachten op de BCLK en LRCLK. De manier waarop het PIO-programma wordt uitgevoerd staat uitgelegd in 2.8.1 Programmable Input/Output en het werkelijke I²S-programma staat in volgende assembly code:

```
.program i2s_bidi_slave
public_entry_point:
    wait 1 pin 2
    wait 0 pin 2
start_l:
    wait 0 pin 1
    pull noblock
    push noblock
    wait 1 pin 1
loop_l:
    wait 0 pin 1
    jmp pin start_r
    out pins 1
    wait 1 pin 1
    in pins 1
    jmp loop_l

start_r:
    wait 0 pin 1
    pull noblock
    push noblock
    wait 1 pin 1
loop_r:
    wait 0 pin 1
    jmp pin continue_r
    jmp start_l
continue_r:
    out pins 1
    wait 1 pin 1
    in pins 1
    jmp loop_r
```

De uitvoering van de assembly code staat uitgevoerd aan de hand van een flowchart in Bijlage G: PIO-flowchart. Wanneer het programma wordt aangeroepen vanuit het C-bestand start het programma genaamd: *‘i2s_bidi_slave’* hierin start de PIO bij de *‘public_entry_point’*. In deze functie wordt er op een pin gewacht, die gelinkt is aan de WS klok, tot de waarde van deze laag is en bijgevolg bij het linkse kanaal van audiodata is. Vervolgens wordt er in de functie *‘start_l’* gewacht op de tweede bit klok die hoog wordt waarna dat het 32-bit input en output register in en uit worden geschoven. Hierna wordt er terug gewacht op de bit klok.

Vervolgens wordt er gekeken of de WS klok al dan niet hoog is en dus aan een ander kanaal is begonnen voor de functie *‘loop_l’*. Indien deze nog voor het linkerkanaal bezig is, wordt er een bit in het output shift register gezet waarna er gewacht wordt op de volgende bitflank. Na deze bit klok wordt er een bit gelezen van de data in lijn en shift de bit in het input shift register. Dan wordt er opnieuw begonnen aan deze functie om te kijken of de WS klok nog steeds op het linkerkanaal staat. Deze *‘loop_l’* functie wordt 32 keer uitgevoerd, voor iedere bit die verstuurd en ontvangen wordt.

Indien de WS klok hoog is, begint het PIO-programma aan de *‘start_r’* functie. Hierbij wordt eerst gewacht op de neergaande bitklok, waarna dat het 32-bit input en output register in en uit worden geshift. Vervolgens wordt er op de hoge bit klok gewacht.

Wanneer de bit klok dan weer laag is, begint *‘loop_r’* waarbij er wordt gekeken of de WS klok al dan niet hoog is. Wanneer deze hoog is, hoort de WS klok nog in het rechterkanaal en worden er twee bits geschoven in het output shift register. Daarna wordt er gewacht tot de bit klok hoog wordt en vervolgens shift er een bit in het input shift register. Vervolgens wordt er weer gewacht op de bit klok die laag wordt. Indien de WS klok nog steeds hoog is, voert deze de functie opnieuw uit. Deze *‘loop_r’* functie wordt 32 keer uitgevoerd, voor iedere bit die verstuurd en ontvangen wordt.

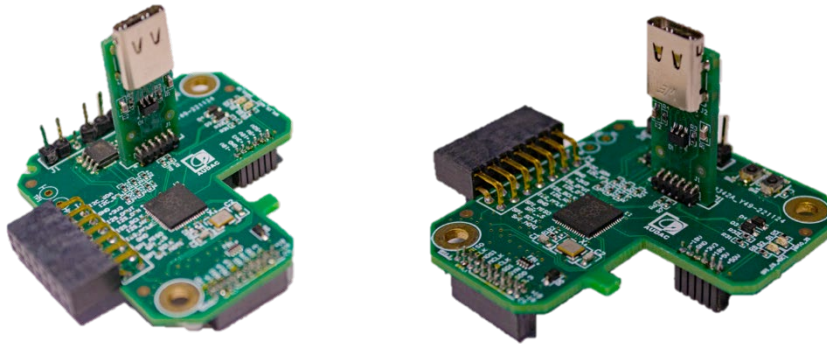
Wanneer de WS klok laag is bij de controle begint het gehele PIO-programma opnieuw bij de *‘start_r’* functie. Dit programma blijft zich voortdurend herhalen voor iedere 64 bits die verstuurd worden.

4. Resultaten

In dit hoofdstuk worden de resultaten, behaald in functie van deze scriptie, besproken. Allereerst wordt de finale PCB in het Audac NWP400 getoond en besproken. Daarna wordt de Windows configuratie van de USB-geluidskaart besproken. Vervolgens wordt de integriteit en kwaliteit van de data besproken. Uiteindelijk worden de uitgevoerde frequentieanalyses visueel gerepresenteerd en geanalyseerd.

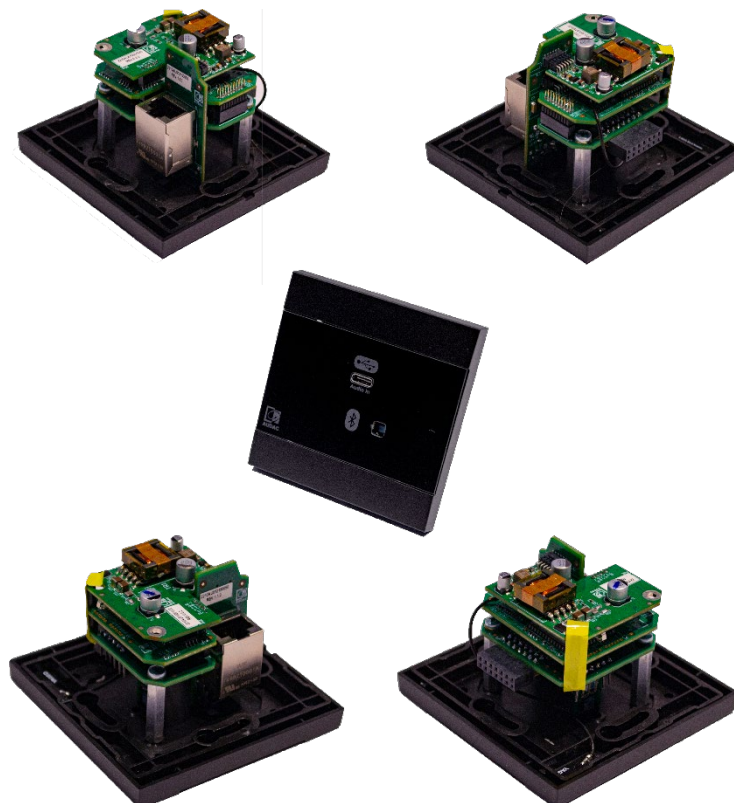
4.1. PCB/Wandmodule

Het resultaat van het ontworpen PCB is zichtbaar in Figuur 53. In dit ontwerp van het PCB zijn de verschillende debug punten zichtbaar en nog steeds bestukt. Dit zou in het absolute finale ontwerp niet zo zijn. Het PCB is zo ontworpen dat deze rechtstreeks geïntegreerd kan worden in de bestaande Audac NWP400 module.



Figuur 53: Resultaat PCB USB naar I²S

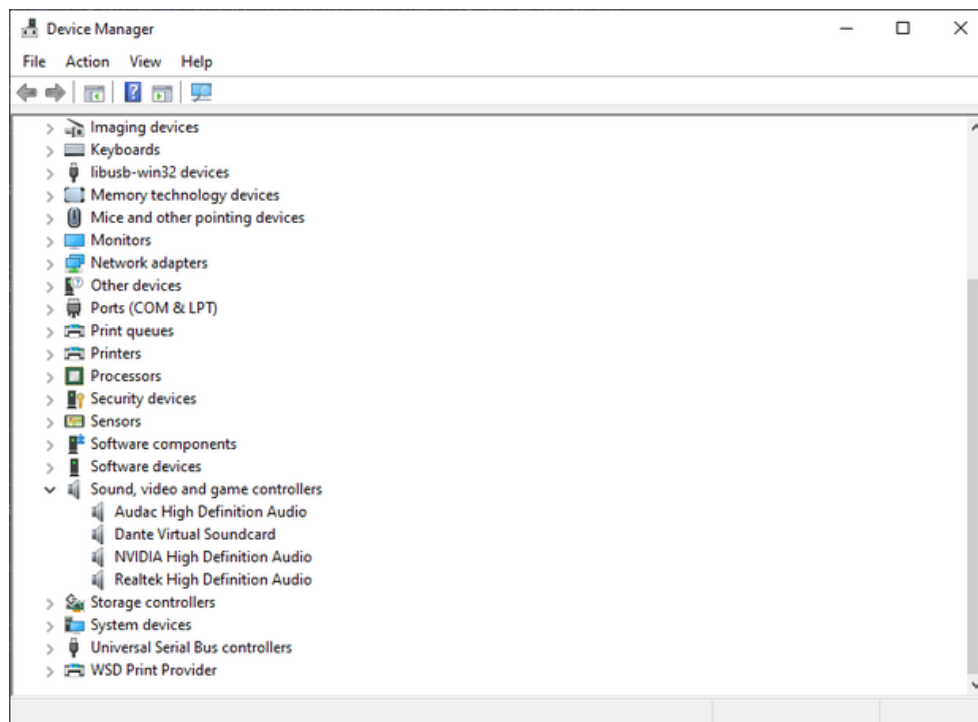
De verschillende aanzichten van de finale wandmodule zijn zichtbaar in Figuur 54. Hierop is te zien dat het PCB goed geïntegreerd is in de bestaande wandmodule. De connectoren komen goed overeen en de debug punten zijn nog steeds goed bereikbaar voor het debuggen van dit ontwerp van de opstelling.



Figuur 54: Resultaat NWP400 USB/Bluetooth wandmodule

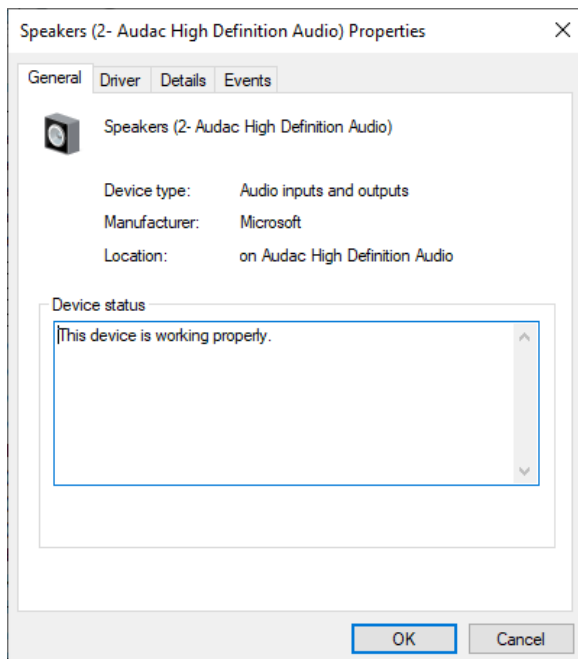
4.2. Windows

Door de USB-descriptor juist in te stellen, wordt de USB-geluidskaart correct herkend in Windows. Een USB-verbinding tussen de wandmodule en een Windows-apparaat zorgt ervoor dat het herkend wordt als een geluidsapparaat in het apparaatbeheer zoals zichtbaar is in Figuur 55.

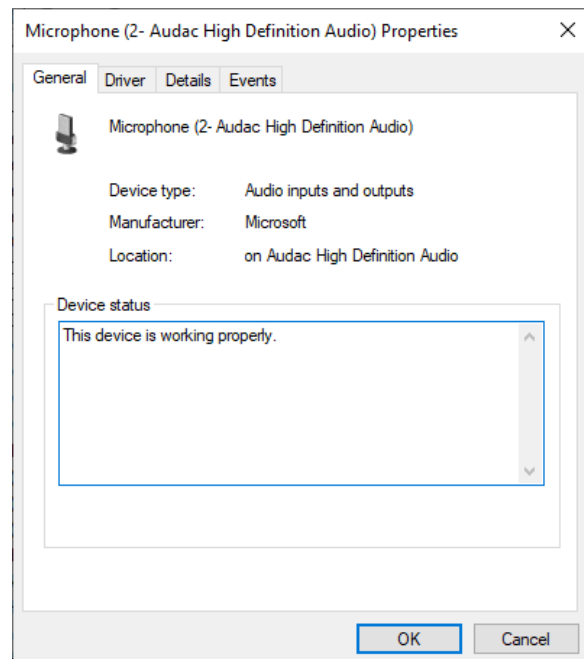


Figuur 55: Apparaatbeheer Audac High Definition Audio

De apparaten die beschikbaar zijn om audio naar of van te streamen zijn de speakers met een kwaliteit van 24-bits stereo bij een sample rate van 48000 Hz zichtbaar in Figuur 56 en de microfoon met een kwaliteit van 24 bit stereo bij een sample rate van 48000 Hz zichtbaar in Figuur 57.



Figuur 56: Apparaatbeheer Speakers



Figuur 57: Apparaatbeheer Microfoon

Door het juiste apparaat te kiezen in de geluidinstellingen van Windows is het nu mogelijk om audiodata te versturen en/of ontvangen via de wandmodule. De besturingsmogelijkheden voor audio zijn ook beschikbaar voor het aansturen van volume en het dempen van de audio.

4.3. Data

Er zijn verschillende datapunten waarvan het resultaat bekeken kan worden door middel van een oscilloscoop. Deze datapunten zijn de I²S-datalijnen en de master klok van het systeem. In Figuur 58 en Figuur 59 zijn de verschillende datalijnen van I²S zichtbaar. De volgende richtingen van data zijn omgeschreven vanuit het perspectief van de microcontroller.

De gele lijn is de binnenkomende data (D_IN), de oranje lijn is de uitgaande data (D_OUT), de blauwe lijn is de bit klok (BCLK) en tenslotte de groene lijn is de linker- en rechterklok (WSCLK). Figuur 58 toont de I²S-lijnen maar op een kleiner tijdsinterval van WSCLK periode. Hierop is te zien dat de linker- en rechterklok exact uit 32 klokslagen van de bit klok bestaat. De datalijnen voor inkomende en uitgaande data is in big endian formaat opgebouwd. Op het moment van opname werd zowel op de inkomende als de uitgaande datalijnen een sinus van 1 Hz afgespeeld.

Figuur 59 toont dezelfde datalijnen in een groter bereik. In deze afbeelding is linksonder ook de periode en de frequentie te zien van ieder signaal. De bit klok en de linker- en rechterklok, met respectievelijk een periode van 325 ns en een frequentie van 3 MHz voor de bit klok en een periode van 20.84 μ s en een frequentie van 48 kHz voor de linker- en rechterklok.



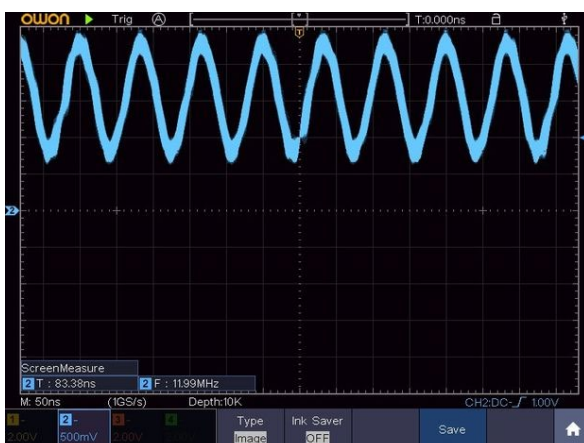
Figuur 58: Resultaat I²S datalijnen



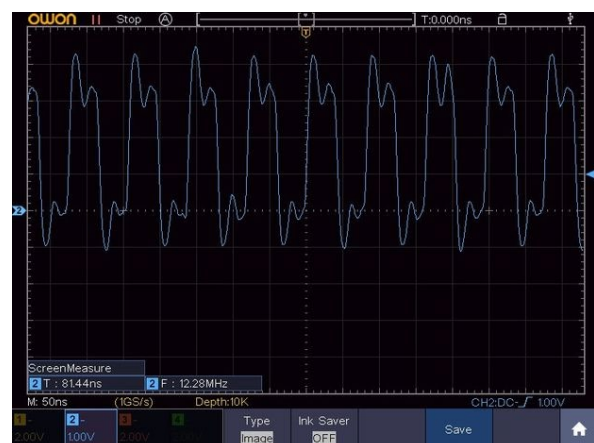
Figuur 59: Resultaat I²S datalijnen groter bereik

In Figuur 60 en Figuur 61 zijn twee master klokken zichtbaar. De originele kristaloscillator, afkomstig van de Raspberry Pi Pico, zichtbaar in Figuur 60 met een periode van 83.38 ns en een frequentie van 12 MHz.

In Figuur 61 de master klok afkomstig van de oscillator in de schakeling van de Dante Ultimo chip met een periode van 81.44 ns en een frequentie van 12.28 MHz. Door deze kleine afwijking van 280 kHz zijn er kleine wijzigingen uitgevoerd in de standaard bibliotheken van de Raspberry Pi Pico omdat de verhoudingen van klokken hierdoor anders lagen dan voordien.



Figuur 60: Originele kristaloscillator XOSC



Figuur 61: Master clock afkomstig van Ultimo chip

4.4. Metingen

In volgende hoofdstukken worden verschillende resultaten van metingen uitgevoerd in Room EQ Wizard besproken aan de hand van bijhorende grafieken. De testen werden afzonderlijk uitgevoerd per kanaal. De resultaten worden getoond en besproken per kanaal.

Omwille van de testopstelling waarbij deze meting werd uitgevoerd, wordt er enkel rekening gehouden met de omzetting van USB naar I²S en omgekeerd. Hierdoor kunnen alle andere factoren die normaal aanwezig zouden zijn bij een meting van een kamer uitgesloten worden zoals bijvoorbeeld kwaliteit microfoon, reflecties, absorptie, etc. De test geeft dus enkel een indicatie of de firmware al dan niet een goede omzetting uitvoert voor de seriële protocollen.

4.4.1. Configuratie

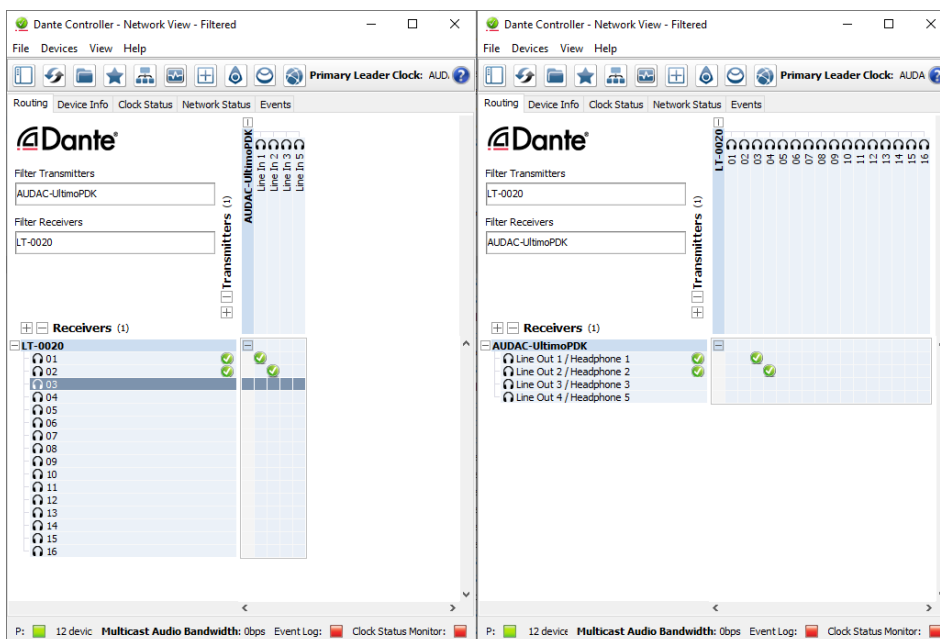
Door middel van de wandmodule door te lussen in de Dante Virtual Soundcard is het mogelijk om audio af te spelen over de USB-geluidskaart, die via de Dante Ultimo chip over het Dante netwerk wordt gestuurd. Over het Dante netwerk aan de hand van Dante Controller is het mogelijk deze terug door te verbinden naar de Dante Virtual Soundcard en daarop op te vangen om te analyseren.

Om de microfoon te testen is dit bijna hetzelfde proces, er wordt echter audiodata afgespeeld via de Dante Virtual Soundcard die dan via het Dante netwerk door lust aan de hand van Dante Controller naar de USB-geluidskaart via de Dante Ultimo chip waardoor deze geluid zal afspelen op de microfoon.

Figuur 62 toont de uitgelegde configuratie in Dante Controller. Volgende tekst is een korte samenvatting van deze configuratie:

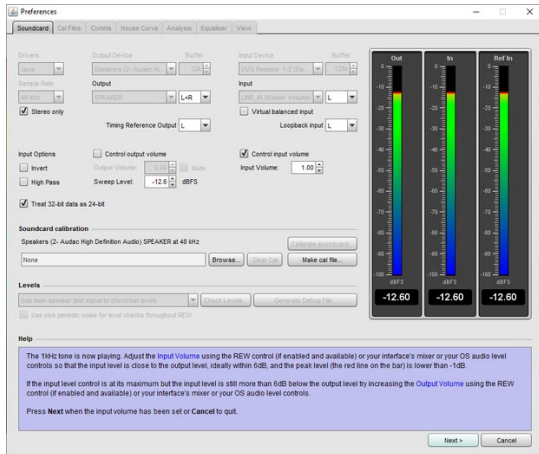
AUDAC-UltimoPDK (Line In 1) → LT-0020 (01)	Linkerkanaal speaker
AUDAC-UltimoPDK (Line In 2) → LT-0020 (02)	Rechterkanaal speaker
LT-0020 (03) → AUDAC-UltimoPDK (Line Out 1)	Linkerkanaal microfoon
LT-0020 (04) → AUDAC-UltimoPDK (Line Out 2)	Rechterkanaal microfoon

Dante Controller in Figuur 62 toont niet alle aanwezige Dante apparaten op het netwerk omdat er gefilterd wordt op de zender en ontvangers waarover het gaat in deze scriptie. De linkerhelft van de afbeelding toont de configuratie waarbij dat er op de speaker audio wordt afgespeeld over USB en opgevangen wordt via de Dante Virtual Soundcard op LT-0020. De rechterhelft van de afbeelding toont de configuratie waarbij er op de microfoon audio wordt afgespeeld via de Dante Virtual Soundcard op LT-0020.

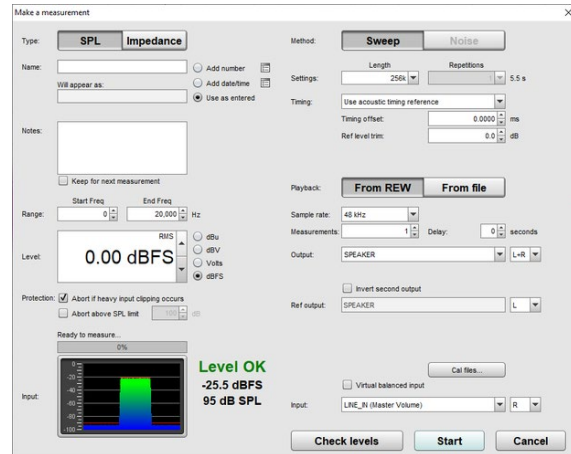


Figuur 62: Dante Controller configuratie REW

De configuratie in de REW-applicatie zelf sluit dicht aan bij de configuratie in Dante Controller. Deze connecties worden namelijk gebruikt als speaker en microfoon in REW. In REW wordt er namelijk audio afgespeeld over een speaker en opgenomen via een microfoon. Uit deze meting worden analyses uitgevoerd die getoond worden in verschillende grafieken. De configuratie van de audioapparaten is getoond in Figuur 63 en de testmeting die uitgevoerd wordt op deze instellingen is zichtbaar in Figuur 64. Aan de hand van deze testmeting is het duidelijk of de meting al dan niet correct is geconfigureerd. Links beneden in deze figuur zien we dat de maximale waarde op een maximale waarde van -25.5 dBFS komt waardoor deze ruim onder de waarde van 0 dBFS zit waarbij clipping zou kunnen optreden.

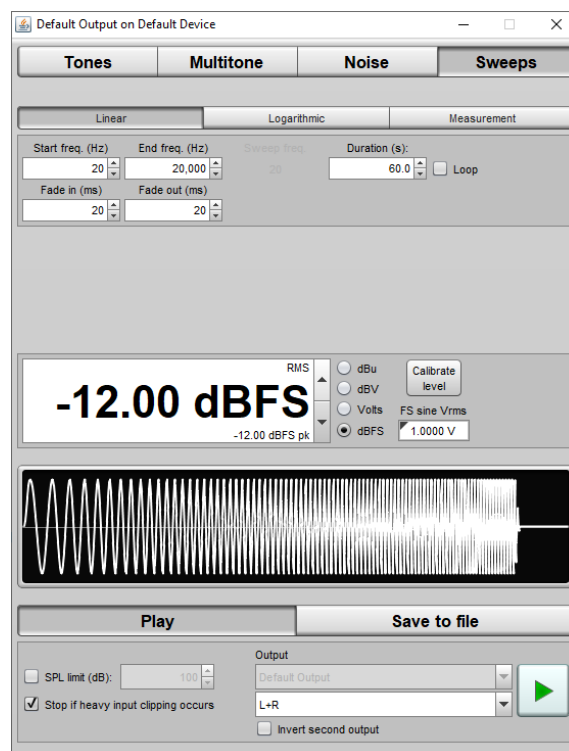


Figuur 63: Voorkeurstellingen testapparatuur



Figuur 64: Testmeting instellingen testapparatuur

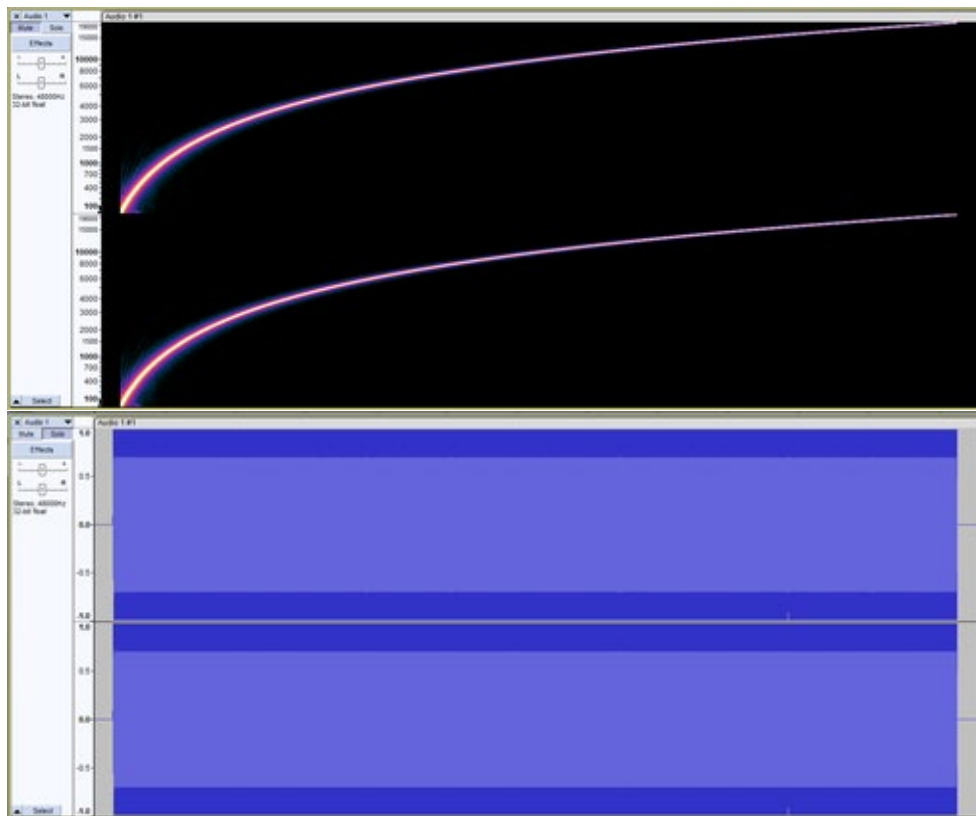
De testmeting die uitgevoerd wordt op de wandmodule is een frequentie analyse waarbij in de loop van 60 seconden, startend van een frequentie van 20 Hz, de frequentie geleidelijk aan zal oplopen tot deze aan het einde van de 60 seconden een maximale frequentie van 20 kHz bereikt heeft. REW voert dan een analyse uit tussen het uitgestuurde signaal en het binnenkomende signaal. De verwachtingen voor deze analyse zou dus moeten zijn dat er absoluut geen verschil is. Dit betekent dan dat de omzetting op een goede manier gebeurt en de data niet vervormd wordt in het proces van de omzetting van USB naar I²S. De configuratie van deze meting is zichtbaar in Figuur 65.



Figuur 65: Frequentieanalyse meting REW

4.4.2. Testdata

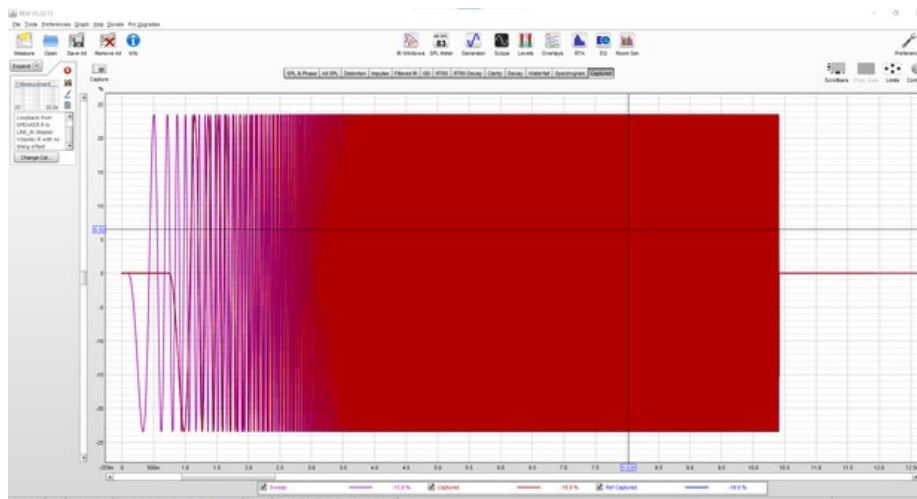
Wanneer de testdata uitgestuurd door REW opgenomen wordt in Audacity bekommt er een grafiek die zichtbaar is in Figuur 66. Bovenste grafiek in Figuur 66 toont het frequentiespectrum van deze meting. De onderste grafiek in Figuur 66 toont de amplitude in functie van de tijd. Het frequentiespectrum toont dat het binnenkomend signaal goed volgt op het verstuurde signaal. Op de gewone grafiek valt weinig af te lezen door hoge, uiteenlopende frequentie. Enkel de stabiele amplitude is hier goed zichtbaar.



Figuur 66: Opname Audacity meting REW

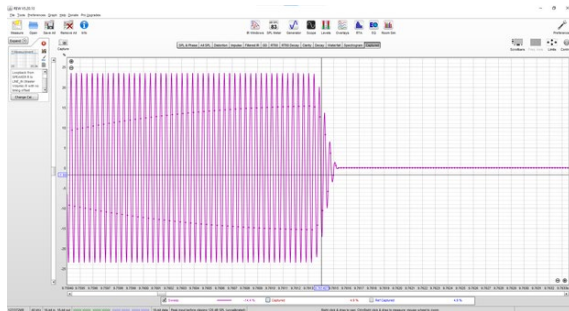
4.4.3. Opgenomen grafiek

Wanneer de frequentieanalyse uitgevoerd wordt in REW wordt de opgenomen en afgespeelde data ook getoond. In Figuur 67 staat de grafiek waarin de paarse lijn de afgespeelde data is en de rode lijn de opgenomen data is. Op het tijdsverschil na komen deze grafieken goed overeen totdat er hogere frequenties afgespeeld worden.

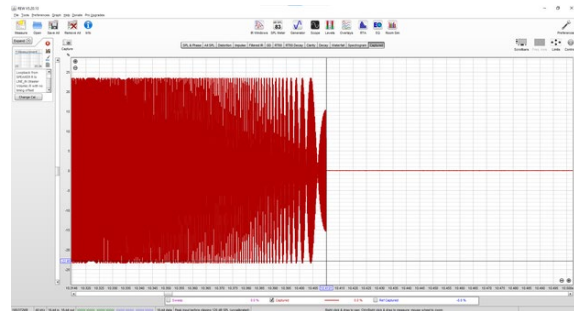


Figuur 67: Opgenomen grafiek REW

De resultaten van de grafieken in Figuur 67 staan afzonderlijk afgebeeld in Figuur 68 en Figuur 69. In Figuur 68 staat de grafiek van de verstuurde data. Op deze grafiek staan ook punten aangeduid, de punten geven aan waar de binnenkomende data juist een sample heeft genomen om de resulterende waarde te bekomen.



Figuur 68: Afspeelde grafiek REW



Figuur 69: Opgenomen grafiek REW

4.4.4. REW-analyse grafieken

Dit deel van de resultaten bespreekt de bekomen grafieken die beschikbaar zijn in Bijlage H: REW-metingen met de bijhorende titel die overeenkomt met iedere uitgevoerde analyse. Ieder hoofdstuk bespreekt kort welke data er juist getoond wordt op de grafiek en bespreekt daarna de resultaten van de grafiek zelf. De legende onderaan de grafiek geeft weer welke kleur lijn overeenstemt met welke waarde.

4.4.4.1. Bode diagram

SPL is een meting van de intensiteit van de geluidsgolf uitgedrukt in decibel. Het stelt de druk van de geluidsgolf voor relatief aan een referentieniveau en wordt gebruikt om het geluidsniveau uit te drukken. Hoe hoger de SPL-waarde, hoe luider het geluid [90]

Fase refereert naar de positie van een geluidsgolf ten opzichte van een referentiepunt in tijd. In audio refereert fase naar de timing van een geluidsgolf ten opzichte van andere geluiden en kan dit invloed hebben op de indruk van kwaliteit en helderheid van de audio. Fase is een belangrijk aspect in audiotoeepassingen. De fase verhouding tussen twee verschillende signalen kan invloed hebben op de algemene ervaring. Bijvoorbeeld, als twee signalen uit fase zijn kunnen ze elkaar annuleren of wanneer ze juist in fase zijn elkaar versterken voor een meer volle ervaring [91].

De SPL voor beide kanalen stemmen overeen met een en dezelfde versterking wat verwacht wordt bij de ingestelde waardes. Voor het rechterkanaal wordt deze uitgedrukt in dB per frequentie op een logaritmische schaal. Voor het linkerkanaal wordt dit uitgedrukt in dBFS per frequentie op een logaritmische schaal. Omdat de versterking hetzelfde blijft doorheen de hele test lijken deze op elkaar ondanks de andere eenheden of verdeling van schalen.

De fase voor beide kanalen stemmen ook overeen met een fase verschuiving van 0° . Dit wilt zeggen dat er geen faseverschuiving is in de audio gedurende de hele 60 seconden waarin de test afgelegd is geweest. De schalen voor het rechter- en linkerkanaal zijn de fase in functie van de frequentie.

4.4.4.2. Vervorming

Vervorming is een soort verandering van het originele audiosignaal die optreedt door ongewenste interferentie bij transport van het signaal of gewenste veranderingen aangelegd door een gebruiker. Vervorming kan vele vormen aannemen, waaronder het versterken of verzwakken van bepaalde frequenties, het introduceren van ruis of andere ongewenste signalen, en het wijzigen van de golfvorm van het signaal zelf [92]

Er zijn veel verschillende soorten vervorming die in een audiosysteem kunnen optreden en elk type heeft zijn eigen kenmerken en effecten op het geluid. Enkele veel voorkomende soorten vervorming zijn:

- Intermodulatievervorming: Dit treedt op wanneer twee of meer signalen zich op een niet-lineaire manier vermengen, waardoor nieuwe frequenties ontstaan die niet aanwezig waren in de oorspronkelijke signalen [93].
- Ruis: Ruis is elk ongewenst signaal dat aanwezig is in het geluid, zoals ruis, brom of elektrische interferentie. Ruis kan op elk punt in een audiosysteem worden geïntroduceerd en kan een aanzienlijke invloed hebben op de waargenomen kwaliteit van het geluid [92].
- Fasevervorming: Dit treedt op wanneer de faseverhouding tussen verschillende frequenties in een audiosignaal wordt gewijzigd, waardoor de golfvorm van het signaal verandert. Fasevervorming kan resulteren in een gebrek aan helderheid of definitie in het geluid. Dit treedt wel enkel op bij installaties waarbij dat er twee bronnen binnenkomen en er een tijdsverschil aanwezig is bij deze signalen [92].

Vervorming wordt over het algemeen beschouwd als een ongewenste eigenschap van audiosystemen, omdat het de natuurgetrouwheid van het geluid kan aantasten en de gehele kwaliteit van de luisterervaring kan verminderen. Sommige vormen van vervorming, zoals versterkervervorming, worden soms opzettelijk in audiosystemen geïntroduceerd om een specifiek geluidseffect te creëren.

De grafieken voor het rechter- en linkerkanaal voor vervorming komen vrij nauwkeurig overeen. De elementen ruis van harmonische of ruis blijven ver uit het hoorbare spectrum voor audio. De vervorming wordt uitgedrukt in dBFS in functie van de frequentie op een logaritmische schaal.

4.4.4.3. Impuls

Een impuls is een type signaal dat in de geluidstechniek wordt gebruikt om de prestaties van een audiosysteem te beoordelen. Een impuls wordt gekenmerkt door een zeer snelle stijgtijd en een zeer korte duur, en wordt vaak gebruikt om de respons van een systeem op een zeer korte, scherpe input te meten [94], [95].

Impulsen worden gebruikt om diverse kenmerken van een audiosysteem te meten, zoals de frequentierespons, de faserespons en de impulsrespons. De frequentierespons van een systeem is een maat voor de manier waarop het reageert op verschillende geluidsfrequenties, terwijl de faserespons de faseverhoudingen tussen verschillende frequenties meet. De impulsrespons is een maat voor hoe het systeem reageert op een korte, scherpe input, en kan worden gebruikt om de algemene kenmerken van het systeem te bepalen [94].

Impulsen worden in de audiosignaalverwerking ook gebruikt om de overdrachtsfunctie van een systeem te meten, die aangeeft hoe het systeem de frequentie- en fasekenmerken van eeningangssignaal beïnvloedt. Deze informatie kan worden gebruikt voor het ontwerpen van filters en andere verwerkingstechnieken om het geluid van een audiosignaal op een gewenste manier vorm te geven [95].

Naast hun technische toepassingen worden impulsen ook gebruikt als geluidseffecten in muziek en andere audioproducties, waar ze kunnen worden gebruikt om allerlei effecten te creëren, zoals explosies, geweervuur of andere scherpe, voorbijgaande geluiden [94]. De grafieken voor het rechter- en linkerkanaal stemmen ook voor de impuls goed overeen. De gemeten waarde is in beide gevallen een snel stijgende impuls. Deze impuls wordt gemeten als dBFS in functie van de lineaire tijd.

De window geeft in dit geval de regio aan waar dat de Fouriertransformatie op wordt uitgevoerd.

4.4.4.4. Waterval

Een watervalgrafiek is een soort grafiek die wordt gebruikt om de frequentierespons van een audiosysteem in de tijd te visualiseren. Het wordt een "watervalgrafiek" genoemd omdat de grafiek de frequentierespons in de tijd lijkt te "vallen", waarbij elke opeenvolgende schijf de respons op een ander moment weergeeft [96], [97].

Een watervalgrafiek toont gewoonlijk de frequentierespons van een systeem op de verticale as en de tijd op de horizontale as. Elke slice van de plot vertegenwoordigt de frequentierespons op een bepaald moment in de tijd, en de slices worden op elkaar gestapeld om de respons in de tijd weer te geven [96].

Watervalplots worden vaak gebruikt in de audioteknik om de frequentierespons van een systeem te visualiseren over een reeks frequenties, en om problemen zoals resonantie, ringing, of andere problemen die de prestaties van het systeem kunnen beïnvloeden, te identificeren. Ze worden ook gebruikt om de respons van een systeem op verschillende soorten ingangssignalen te visualiseren, zoals impulsen of geveegde sinusgolven. Het voordeel ten opzichte van de impulsrespons is dat de impulsrespons over een korte periode de reactie van het systeem bekijkt, terwijl dat de waterval over een langere periode bekeken wordt [96].

Watervalplots zijn nuttige hulpmiddelen om de frequentieresponskenmerken van audiosystemen te begrijpen, en worden vaak gebruikt in audio-onderzoek. De grafieken voor het rechter- en linkerkanaal komen zo goed als volledig overeen.

4.4.4.5. Spectrogram

Een spectrogram is een grafische weergave van het frequentiespectrum van een geluid in de tijd. Het is een visuele weergave van de spectrale inhoud van een geluid, en wordt gebruikt om de eigenschappen van een geluid, zoals toonhoogte, timbre en omhullende, te analyseren en te begrijpen [97], [98].

Een spectrogram wordt meestal uitgezet met de tijd op de x-as, de frequentie op de y-as en de intensiteit of het vermogen van de frequentiecomponenten op de z-as (weergegeven als een kleurenkaart). Het spectrogram toont hoe de spectrale inhoud van een geluid in de tijd verandert, zodat de gebruiker de evolutie van de spectrale inhoud van het geluid kan zien naarmate het vordert [98].

Spectrogrammen worden veel gebruikt in geluidstechniek en onderzoek om de eigenschappen van geluiden te analyseren, zoals spraak, muziek of omgevingsgeluiden. Ze kunnen worden gebruikt om specifieke frequenties of frequentiebereik binnen een geluid te identificeren, om de toonhoogte en het timbre van een geluid te analyseren, en om de omhullende en temporele kenmerken van een geluid te begrijpen.

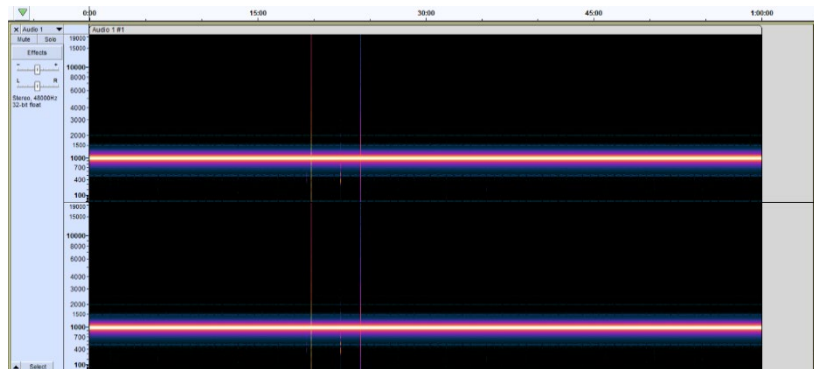
Spectrogrammen zijn ook nuttig voor het detecteren en analyseren van specifieke soorten geluiden, zoals vogelgeluiden of vocalisaties van dieren, en worden vaak gebruikt op gebieden als biologie en ecologie om het gedrag van dieren te bestuderen. Daarnaast worden spectrogrammen gebruikt bij de verwerking van audiosignalen om filters en andere verwerkingstechnieken te ontwerpen en te optimaliseren die worden gebruikt om het geluid van een audiosignaal vorm te geven.

De spectrogramgrafiek voor het rechter- en linkerkanaal komen perfect overeen. Dit wilt zeggen dat de versterking voor elke frequentie in elk tijdstip goed overeenkwam waardoor de ideale spectrogramgrafiek gevormd wordt in beide gevallen.

4.5. Langdurige test

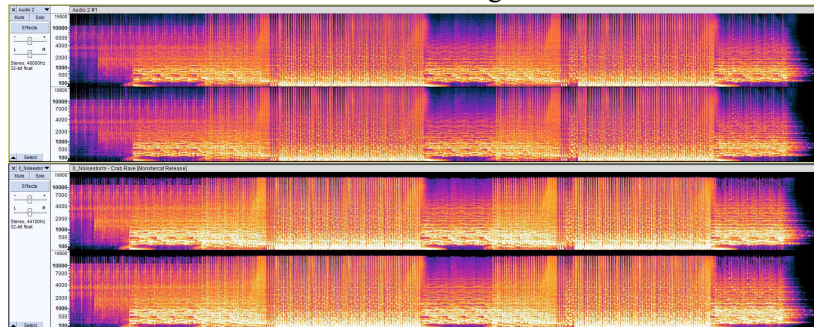
Door gebruik te maken van dezelfde testopstelling voor bij het REW-programma is het mogelijk een signaal af te spelen en terug op te nemen via Audacity. Figuur 70 toont de opname van een sinusgolf met een frequentie van 1 kHz gedurende een uur visueel getoond op het frequentiespectrum in functie van de tijd. De opname ziet er vrijwel perfect uit op uitzondering van enkele kortstondige frequentieverandering op drie individuele tijdstippen.

De reden van deze drie frequentieveranderingen zijn niet te verklaren en zullen waarschijnlijk te maken hebben met de firmware en hoe er omgegaan wordt met het geheugen. Door het gelimiteerd voorkomen van deze frequentieveranderingen is het moeilijk om te debuggen.



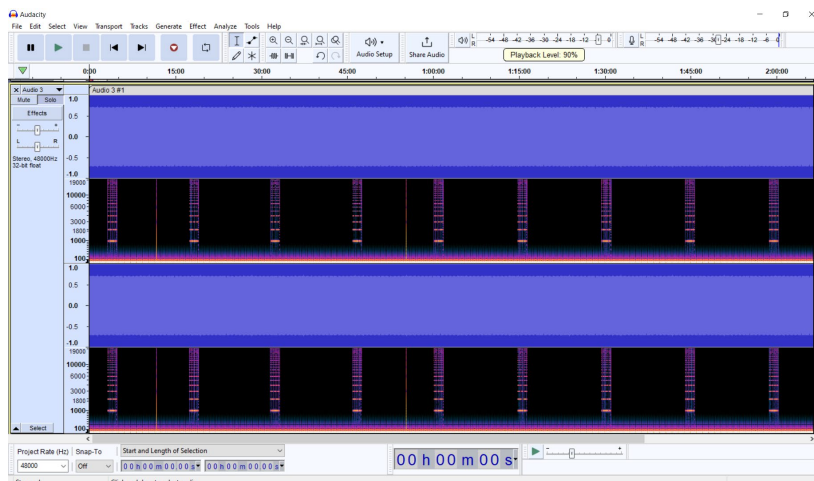
Figuur 70: Speaker opname van 1 kHz gedurende een uur

Een andere uitgevoerde test was een lied afspelen en dan het frequentiespectrum van het originele lied vergelijken met de opgenomen versie. In Figuur 71 staan twee geluidstracks in Audacity. Het bovenste van deze twee is de opgenomen versie terwijl de onderste versie de originele is. Deze twee frequentiespectra lijken relatief goed overeen te komen, al verschillen de frequentiespectra bij de lagere frequenties aanzienlijk meer dan bij de hoge frequenties. Door een geluidscontrole uit te voeren op het opgenomen signaal door middel van het geluid af te spelen op een aangekochte geluidskaart kan geverifieerd worden dat de twee verschillende tracks geen hoorbaar verschil hebben.



Figuur 71: Vergelijking opgenomen lied en origineel lied

Een laatste uitgevoerde test was op het microfoon gedeelte van de USB-geluidskaart. Deze test werd uitgevoerd door een sinusgolf met een frequentie van 100 Hz gedurende twee uur af te spelen via de Dante Virtual Soundcard op de microfoon en op te nemen. Figuur 72 toont deze opname in Audacity in 'Multi-View' waarbij zowel het frequentiespectrum als de gemeten amplitude in functie van tijd worden getoond. Hierop is te zien dat ook hier waarschijnlijk een fout zit in de firmware waardoor er regelmatig verschillende frequentieverandering tevoorschijn komen. Alsook is het op te merken dat deze veranderingen zich in een herhalend patroon bevinden. Maar de exacte oorzaak van dit probleem is niet bepaald.



Figuur 72: Microfoon opname van een 100 Hz sinusgolf gedurende twee uur

5. Conclusie

Aan de hand van de besproken resultaten en uitgevoerde metingen uit hoofdstuk 4 kan geconcludeerd worden dat de opzet van het onderwerp geslaagd is. De metingen tonen aan dat, ondanks enkele fouten in de buffer, de gekozen microprocessor de taak van de gedateerde chip efficiënt kan overnemen.

De integratie van de microprocessor in de bestaande wandmodule is geslaagd en past in de PCB-stack met een werkend proof of concept tot gevolg. Over het gehele project zijn er nog verbeterpunten die zeker in toekomstig werk geïmplementeerd kunnen worden.

5.1. Toekomstig werk

Om dit project tot een perfect resultaat te brengen, is het nodig om de firmware nog aan te passen op verschillende plekken. De voornaamste aanpassing is het nog beter afstemmen van de buffers en de DMA-handelingen zodanig dat de verdwenen samples die nu slechts af en toe voorkomen niet meer voorkomen.

In verband met de PCB zouden er enkele goede praktische veranderingen kunnen gebeuren. Een eerste verbeterpunt is het apart naar buiten brengen van de UART-connectie. Momenteel wordt deze connectie via de SPI-pinnen naar buiten gebracht aangezien die momenteel nog niet voor andere doeleinden gebruikt worden.

Ten tweede zou er nog een buck-converter geplaatst kunnen worden tussen de USB-voeding en de systeem voeding. Op deze manier kan de 5 V van USB omlaag gebracht worden naar een voeding van 3 V waardoor het systeem volledig gevoed zou kunnen worden via USB-spanning.

Ten derde mag de kristaloscillator, die nu nog toegevoegd is aan het ontwerp, volledig vervangen worden door de master klok van de Ultimo-chip. Dat zorgt ervoor dat de fabricagekost van de PCB verlaagt en de klok niet meer uitgeschakeld moet worden in de firmware. Door dit onderzoek is aangetoond dat de master klok van de Ultimo-chip voldoet als master klok voor de Raspberry Pi RP2040.

Als laatste is het nog mogelijk om de resterende GPIO-pinnen van de Raspberry Pi RP2040 naar buiten te brengen op het bordje aan de hand van testpunten of via's. Als het ontwerp zo wordt ontworpen is het mogelijk om de GPIO's alsnog te gebruiken voor testing/debugging wanneer de PCB bestukt is zonder rechtstreeks op de pinnen van de kleine QFN-56 IC-behuizing te moeten solderen.

Referentielijst

- [1] “Inspiring sound solutions | AUDAC.” <https://audac.eu/eu/> (accessed Oct. 03, 2022).
- [2] “PROCAB | Professional high quality cables and connectors.” <https://procab.be/be-en/> (accessed Oct. 03, 2022).
- [3] “Professional stands and racks I CAYMON.” <https://caymon.eu/> (accessed Oct. 03, 2022).
- [4] “5.3.2(2). Digitale opslag en analyse van geluid - Audiologieboek.” <https://audiologieboek.nl/content/5-3-22-digitale-opslag-en-analyse-van-geluid/> (accessed Oct. 14, 2022).
- [5] Microchip Technology Inc., “AN2582 Creating a USB Audio Device on a PIC32 MCU Using MPLAB Harmony Introduction,” 2017.
- [6] H. Nyquist, “Certain Topics in Telegraph Transmission Theory Classic Paper,” pp. 617–644.
- [7] C. E. Shannon, “Communication in the Presence of Noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949, doi: 10.1109/JRPROC.1949.232969.
- [8] “Sample Rate and Bit Depth | Articles | Tom S. Ray Audio Mastering.” <https://tomsrayaudiomastering.com/sample-rate-and-bit-depth/> (accessed Oct. 14, 2022).
- [9] S. Paonessa, “Reducing Signal Noise in Practice | Precision Digital.” <https://www.predig.com/whitepaper/reducing-signal-noise-practice> (accessed Nov. 13, 2022).
- [10] “Total Harmonic Distortion – what happened to my perfect wave?” <https://www.seencogroup.com/en/total-harmonic-distortion-what-happened-to-my-perfect-wave/> (accessed Nov. 13, 2022).
- [11] I. v. Blagouchine and E. Moreau, “Analytic method for the computation of the total harmonic distortion by the cauchy method of residues,” *IEEE Transactions on Communications*, vol. 59, no. 9, pp. 2478–2491, Sep. 2011, doi: 10.1109/TCOMM.2011.061511.100749.
- [12] “What is frequency response and how does it affect my music?” <https://www.soundguys.com/frequency-response-explained-16507/> (accessed Nov. 13, 2022).
- [13] “What is Frequency Response? - MATLAB & Simulink.” <https://nl.mathworks.com/discovery/frequency-response.html> (accessed Nov. 13, 2022).
- [14] “What is Frequency Response | Basics | Electrical Academia.” <https://electricalacademia.com/basic-electrical/frequency-response-basics/> (accessed Nov. 13, 2022).
- [15] “What is Frequency Response in Speakers? - Center Speaker Reviews and Buyer’s Guides.” <http://centerspeakerzone.com/what-is-frequency-response-in-speakers/> (accessed Nov. 13, 2022).
- [16] J. Axelson, “USB Complete: The Developer’s Guide, Fifth Edition,” 2015.
- [17] S. L. Garfinkel, “Boston Globe Online / Business / USB deserves more support,” 1999. https://simson.net/clips/1999/99.Globe.05-20.USB_deserves_more_support+.shtml (accessed Oct. 14, 2022).
- [18] USB Implementors Forum Inc., “Defined Class Codes | USB-IF,” 2022. <https://www.usb.org/defined-class-codes> (accessed Oct. 14, 2022).
- [19] USB Implementors Forum Inc., “USB Promoter Group USB4 Specification,” Aug. 2019.

- [20] A. Potts, “USB Explained: All the Different Types of USB cable and What They’re Used for,” 2021. <https://www.monkeyusb.com/blogs/news/usb-explained-all-the-different-types-of-usb-cable-and-what-they-re-used-for> (accessed Oct. 14, 2022).
- [21] S. Blessy, “10 USB Pinout Explained- USB A, B, C(Male and Female),” 2022. <https://www.etechnophiles.com/usb-pinout-ports-connectors/> (accessed Oct. 14, 2022).
- [22] Y. Yakimova, “Long-awaited common charger for mobile devices will be a reality in 2024 | News | European Parliament,” 2022. <https://www.europarl.europa.eu/news/en/press-room/20220930IPR41928/long-awaited-common-charger-for-mobile-devices-will-be-a-reality-in-2024> (accessed Oct. 17, 2022).
- [23] J. Porter and J. Vincent, “USB-C will be mandatory for phones sold in the EU ‘by autumn 2024’ - The Verge,” 2022. <https://www.theverge.com/2022/6/7/23156361/european-union-usb-c-wired-charging-iphone-lightning-ewaste> (accessed Oct. 17, 2022).
- [24] “Members | USB-IF.” <https://www.usb.org/members> (accessed Oct. 21, 2022).
- [25] M. Bergman *et al.*, “Device Class Definition for Human Interface Devices (HID),” 2001.
- [26] J. Scanlon *et al.*, “Universal Serial Bus Device Class Definition For Audio Devices,” Apr. 2019.
- [27] R. Triggs, “USB Audio Class 3.0 for USB Type-C headphones needs universal software adoption in Android and other operating systems,” 2018. <https://www.soundguys.com/android-usb-audio-class-3-0-18494/> (accessed Oct. 17, 2022).
- [28] J. Wong, “USB Audio Class – JimmyIoT,” 2020. <https://jimmywongiot.com/2020/04/08/usb-audio-class/> (accessed Oct. 17, 2022).
- [29] Inc. Silicon Laboratories, “USB Audio Simplified”, Accessed: Oct. 17, 2022. [Online]. Available: www.usb.org
- [30] C. Punter, “USB Audio – Synchronous / Asynchronous | HFA - The Independent Source for Audio Equipment Reviews,” 2011. <https://www.hifi-advice.com/blog/audiophile-insights/digital-insights/usb-audio-synchronous-asynchronous/> (accessed Oct. 17, 2022).
- [31] S. H. Angara, “Achieving Bit-Perfect USB Audio | Electronic Design,” 2016. <https://www.electronicdesign.com/technologies/embedded-revolution/article/21801786/achieving-bitperfect-usb-audio> (accessed Oct. 17, 2022).
- [32] Naim Audio Ltd, “Asynchronous USB,” Jan. 2013, Accessed: Oct. 17, 2022. [Online]. Available: https://www.naimaudio.com/sites/default/files/products/downloads/files/dac-v1_asynchronous-usb_mwp_jan13_0.pdf
- [33] AudioPhilleo LLC., “ASYNCHRONOUS USB - audiophilleo.” <http://www.audiophilleo.com/home/definition/asynchronous%20USB> (accessed Oct. 17, 2022).
- [34] S. C. Black, “What is Asynchronous USB? | Do You NEED It? - Home Studio Basics,” 2021. <https://homestudiobasics.com/what-is-asynchronous-usb/> (accessed Oct. 17, 2022).
- [35] NXP Semiconductors, “UM11732 I2S bus specification,” Feb. 2022.
- [36] “Philips Semiconductors I2S bus specification,” 1986.
- [37] NXP Semiconductors, “UM10204 I2C-Bus specification.” [Online]. Available: <http://www.nxp.com>

- [38] C. Wegener, M. Neupert, and S. Stang, “FPGA-accelerated Real-Time Audio in Pure Data Acoustic Interfaces View project Tickle View project FPGA-accelerated Real-Time Audio in Pure Data,” *19th Sound and Music Computing Conference*, pp. 168–175, doi: 10.5281/zenodo.6572969.
- [39] S. Dimitrov and S. Serafin, “Towards an open sound card-a bare-bones FPGA board in context of PC-based digital audio based on the AudioArduino open sound card system,” *Audio Mostly*, 2011.
- [40] H. Ren, J. Cui, W. Xie, C. Zhou, and K. Zhang, “Design of Low-Power USB Audio System Based on LPM Protocol,” in *Lecture Notes of the Institute for Computer Sciences: Simulation Tools and Techniques*, 11th ed., vol. 295, 2019, pp. 218–227.
- [41] C. Zhu, W. Wu, L. Hou, and J. Wang, “FPGA-Based Audio Signal Processing Module Design in KVMsystem”.
- [42] K. Vaca, M. M. Jefferies, and X. Yang, “An Open Audio Processing Platform with Zync FPGA”.
- [43] “Terasic - All FPGA Boards - Cyclone V - DE10-Nano Kit.” <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1046> (accessed Nov. 14, 2022).
- [44] “XC3S50AN-4TQG144I Xilinx | Distributors, Price Comparison, and Datasheets | FPGAs - Field Programmable Gate Arrays | Octopart.” <https://octopart.com/xc3s50an-4tqg144i-xilinx-7423549> (accessed Nov. 14, 2022).
- [45] “EP3C5F256C8N Intel / Altera | Distributors, Price Comparison, and Datasheets | FPGAs - Field Programmable Gate Arrays | Octopart.” <https://octopart.com/ep3c5f256c8n-intel+%2F+altera-7721024?r=sp> (accessed Nov. 14, 2022).
- [46] “XC7Z007S-1CLG225C Xilinx | Distributors, Price Comparison, and Datasheets | FPGAs - Field Programmable Gate Arrays | Octopart.” <https://octopart.com/xc7z007s-1clg225c-xilinx-76967862?r=sp> (accessed Nov. 14, 2022).
- [47] M. Babiuch, P. Foltýnek, and P. Smutny, “Using the ESP32 Microcontroller for Data Processing,” 2019.
- [48] S. Stanivuk, N. Cetic, J. Kovacevic, and M. Boric-Herget, “Twitter controlled real time audio synthesizer based on XMOS platform,” 2013.
- [49] S. Mateski and Z. Anastasovski, “Digital Sound Recorder with ARM microcontroller and SD Card ,” 2012.
- [50] C.-Y. Chang and S.-T. Li, “Active Noise Control in Headsets by Using a Low-Cost Microcontroller,” 2011.
- [51] “ESP32-S2 Espressif Systems | Distributors, Price Comparison, and Datasheets | RF Receivers, Transceivers | Octopart.” <https://octopart.com/esp32-s2-espressif+systems-103108191?r=sp> (accessed Nov. 14, 2022).
- [52] “XS1-L4A-64-TQ48-C4 XMOS | Distributors, Price Comparison, and Datasheets | Embedded Processors and Controllers | Octopart.” <https://octopart.com/xs1-l4a-64-tq48-c4-xmos-28874582?r=sp> (accessed Nov. 14, 2022).
- [53] “AT91SAM7S256D-MU Microchip | Distributors, Price Comparison, and Datasheets | Microcontrollers | Octopart.” <https://octopart.com/at91sam7s256d-mu-microchip-77759062?r=sp> (accessed Nov. 14, 2022).

- [54] “PIC24HJ32GP204T-I/ML Microchip | Distributors, Price Comparison, and Datasheets | Microcontrollers | Octopart.” <https://octopart.com/pic24hj32gp204t-i%2Fml-microchip-17006636?r=sp> (accessed Nov. 14, 2022).
- [55] Raspberry Pi Ltd, “Raspberry Pi Documentation - RP2040.” <https://www.raspberrypi.com/documentation/microcontrollers/rp2040.html#software-development> (accessed Oct. 17, 2022).
- [56] Raspberry Pi Ltd, “Raspberry Pi Pico Datasheet: An RP2040-based microcontroller board,” Jun. 2022. Accessed: Oct. 17, 2022. [Online]. Available: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>
- [57] Raspberry Pi Ltd, “RP2040 Datasheet: A microcontroller by Raspberry Pi,” Jun. 2022, Accessed: Oct. 17, 2022. [Online]. Available: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>
- [58] Raspberry Pi Ltd, “Getting started with Raspberry Pi Pico: C/C++ development with Raspberry Pi Pico and other RP2040-base microcontroller boards,” Jun. 2022, Accessed: Oct. 17, 2022. [Online]. Available: <https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf>
- [59] Audinate Pty Ltd., “Dante Controller User Guide,” 2022.
- [60] Audinate Pty Ltd., “Dante Ultimo Rev 4.0.x Technical Datasheet,” 2022.
- [61] Audinate Pty Ltd., “Dante Ultimo Marketing Datasheet,” 2021.
- [62] Audinate Pty Ltd., “Dante Virtual Soundcard,” 2022.
- [63] “IEEE Std 1588bTM-2022, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, Amendment 1: Precision Time Protocol (PTP) Mapping for Transport over the Optical Transport Network (OTN).” Technical Committee on Sensor Technology (TC-9), 2022.
- [64] The Institute of Electrical and Electronics Engineering, “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” pp. 1–154, Nov. 2002, Accessed: Nov. 14, 2022. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1048550>
- [65] “Dante Ultimo | Audinate | Pro AV’s Leading Networking Technology.” <https://www.audinate.com/products/manufacture-products/dante-ultimo> (accessed Dec. 04, 2022).
- [66] “XU208-256-TQ64-C10 - Xmos - 32 Bit Microcontroller, 8 Core, Programmable High Speed USB.” <https://www.newark.com/xmos/xu208-256-tq64-c10/32-bit-microcontroller-8-core/dp/68Y1459> (accessed Dec. 04, 2022).
- [67] “XMOS - Bringing technology to life.” <https://www.xmos.ai/> (accessed Dec. 04, 2022).
- [68] “USB to I2S Interface XMOS XU208 32bit 384kHz DSD256 - Audiophonics.” <https://www.audiophonics.fr/en/interface-modules/usb-to-i2s-interface-xmos-xu208-32bit-384khz-dsd256-p-14179.html> (accessed Dec. 04, 2022).
- [69] “STM32F0 - Arm Cortex-M0 - USB Microcontrollers - STMicroelectronics.” <https://www.st.com/en/microcontrollers-microprocessors/stm32f0-series.html> (accessed Dec. 04, 2022).
- [70] ST Microelectronics, “AN4711 Application note USB audio bridge example with STM32F0 MCUs,” 2016.

- [71] “STM32F4 - ARM Cortex-M4 High-Performance MCUs - STMicroelectronics.” <https://www.st.com/en/microcontrollers-microprocessors/stm32f4-series.html> (accessed Dec. 04, 2022).
- [72] “AN3997 Application note: Audio playback and recording using the STM32F4DISCOVERY,” 2011, Accessed: Dec. 04, 2022. [Online]. Available: www.st.com/stm32f4-discovery.
- [73] “STM32F4DISCOVERY - Discovery kit with STM32F407VG MCU - STMicroelectronics.” <https://www.st.com/en/evaluation-tools/stm32f4discovery.html> (accessed Dec. 04, 2022).
- [74] Microchip, “PIC32MX5XX/6XX/7XX,” 2019.
- [75] “AN2582 Creating a USB Audio Device on a PIC32 MCU Using MPLAB Harmony Introduction,” 2017.
- [76] “PIC32 USB Audio | Microchip Technology.” <https://www.microchip.com/en-us/solutions/consumer/audio-and-speech/pic32-digital-audio/pic32-usb-audio#> (accessed Dec. 04, 2022).
- [77] “ESP32S2 Series Datasheet SoC with Xtensa® SingleCore 32bit LX7 Microprocessor Supporting IEEE 802.11b/g/n (2.4 GHz WiFi),” 2022, Accessed: Dec. 06, 2022. [Online]. Available: <https://www.espressif.com/en/support/download/documents>.
- [78] “USB Device Driver - ESP32-S2 - — ESP-IDF Programming Guide latest documentation.” https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/api-reference/peripherals/usb_device.html (accessed Dec. 06, 2022).
- [79] “GitHub - elehobica/pico_sine_wave_i2s_32b: Raspberry Pi Pico sine_wave with 32bit I2S DAC.” https://github.com/elehobica/pico_sine_wave_i2s_32b (accessed Dec. 04, 2022).
- [80] “ESP32-S2 Wi-Fi MCU I Espressif.” <https://www.espressif.com/en/products/socs/esp32-s2> (accessed Dec. 06, 2022).
- [81] “Dante Ultimo PDK User Guide”, Accessed: Dec. 05, 2022. [Online]. Available: <https://dev.audinate.com/documentation/ultimo/pdk/userguide/pdf/latest/AUD-MAN-UltimoPDK-v3.5.pdf>
- [82] Raspberry Pi Ltd., “Hardware design with RP2040”, Accessed: Dec. 07, 2022. [Online]. Available: <https://datasheets.raspberrypi.com/rp2040/hardware-design-with-rp2040.pdf>
- [83] “PCB Design Guidelines - Design for Manufacture - Eurocircuits.” <https://www.eurocircuits.com/pcb-design-guidelines/> (accessed Dec. 08, 2022).
- [84] Dipl.-I. H. Leicht and A. Thumm, “Today’s Vapor Phase Soldering An Optimized Reflow Technology for Lead Free Soldering”, Accessed: Dec. 08, 2022. [Online]. Available: www.ibl-loettechnik.de,
- [85] “Vapor Phase Soldering :: Total Materia Article.” <https://www.totalmateria.com/page.aspx?ID=CheckArticle&site=ktn&NM=449> (accessed Dec. 08, 2022).
- [86] “Why Do We Need To Use Vapor Phase Reflow Technology.” <https://www.ourpcb.com/vapor-phase-reflow.html> (accessed Dec. 08, 2022).
- [87] “Mounting Techniques Overview.” <https://epci.eu/3-mounting-techniques-overview/> (accessed Dec. 08, 2022).
- [88] “Home - TinyUSB.” <https://docs.tinyusb.org/en/latest/> (accessed Dec. 27, 2022).

- [89] Z. Gingl, G. Makan, J. Mellar, G. Vadai, and R. Mingesz, “Phonocardiography and Photoplethysmography with Simple Arduino Setups to Support Interdisciplinary STEM Education,” *IEEE Access*, vol. 7, pp. 88970–88985, 2019, doi: 10.1109/ACCESS.2019.2926519.
- [90] “Sound pressure level.” <https://www.ecophon.com/en/about-ecophon/acoustic-knowledge/basic-acoustics/sound-pressure-level/> (accessed Jan. 12, 2023).
- [91] “Understanding Audio Phase | Universal Audio.” <https://www.uaudio.com/blog/understanding-audio-phase/> (accessed Jan. 12, 2023).
- [92] “What is Audio Distortion?” <https://www.premiumbeat.com/blog/what-is-audio-distortion/> (accessed Jan. 12, 2023).
- [93] “Intermodulation distortion (audio) - Glossary - Federal Agencies Digitization Guidelines Initiative.” <https://www.digitizationguidelines.gov/term.php?term=intermodulationdistortionaudio> (accessed Jan. 12, 2023).
- [94] “Impulse Responses.” https://www.roomeqwizard.com/help/help_en-GB/html/impulseresponse.html (accessed Jan. 12, 2023).
- [95] “What Is Impulse Response In Audio ? – SoundAssured.” <https://www.soundassured.com/blogs/blog/what-is-impulse-response-in-audio> (accessed Jan. 12, 2023).
- [96] “Waterfall Graph.” https://www.roomeqwizard.com/help/help_en-GB/html/graph_waterfall.html (accessed Jan. 12, 2023).
- [97] “RealTraps - How Does That Sound Look?” https://realtraps.com/art_dan.htm (accessed Jan. 12, 2023).
- [98] “Spectrogram analysis waterfall diagram — Sorama.” <https://www.sorama.eu/blog/spectrogram-analysis-waterfall-diagram> (accessed Jan. 12, 2023).

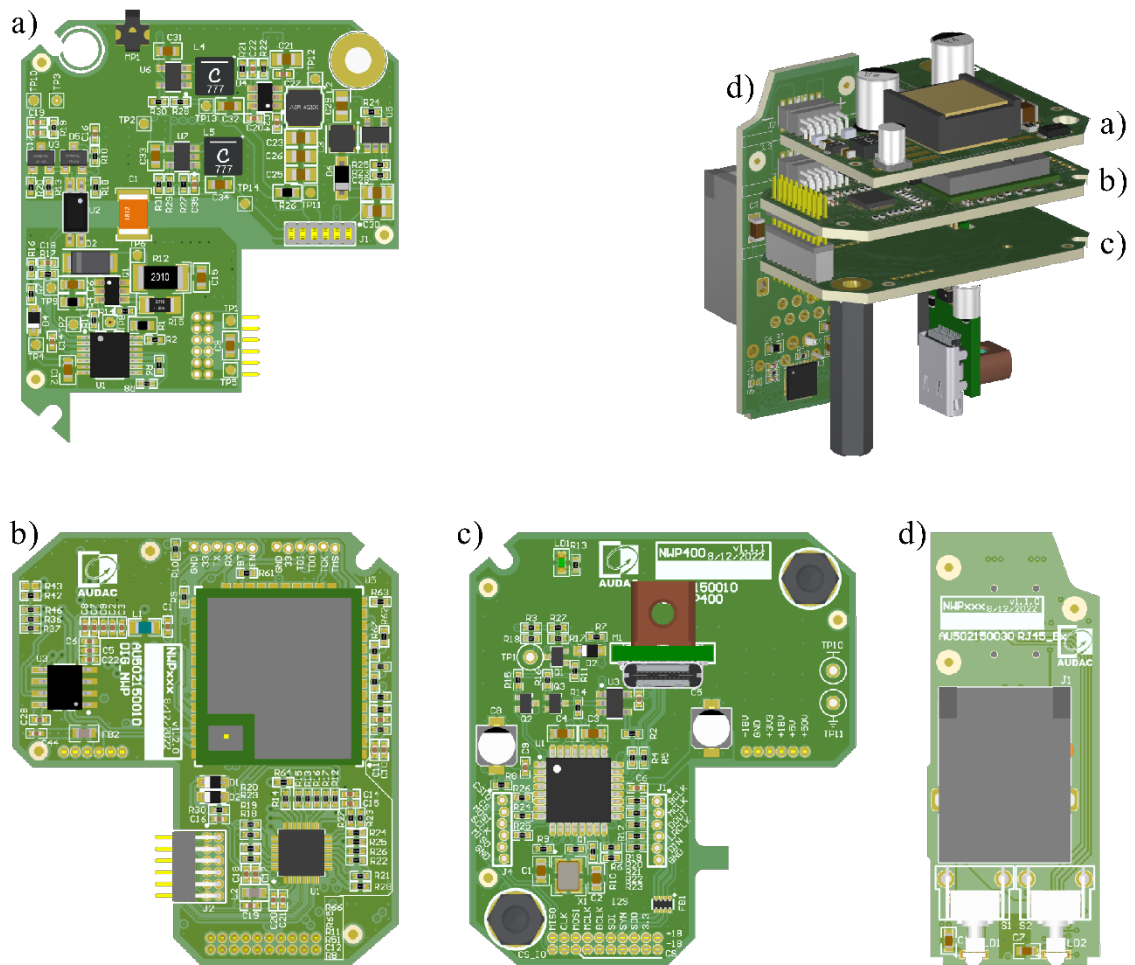
Bijlagen

Bijlage A: Samenvatting vergelijking mogelijke microcontrollers	75
Bijlage B: Wandmodule PCB-stack	77
Bijlage C: Topschema uit Altium Designer	79
Bijlage D: USB naar I2S schema in Altium Designer	81
Bijlage E: PCB-lagen	83
Bijlage F: BOM NWP400 USB naar I2S	85
Bijlage G: PIO-flowchart	87
Bijlage H: REW-metingen	89

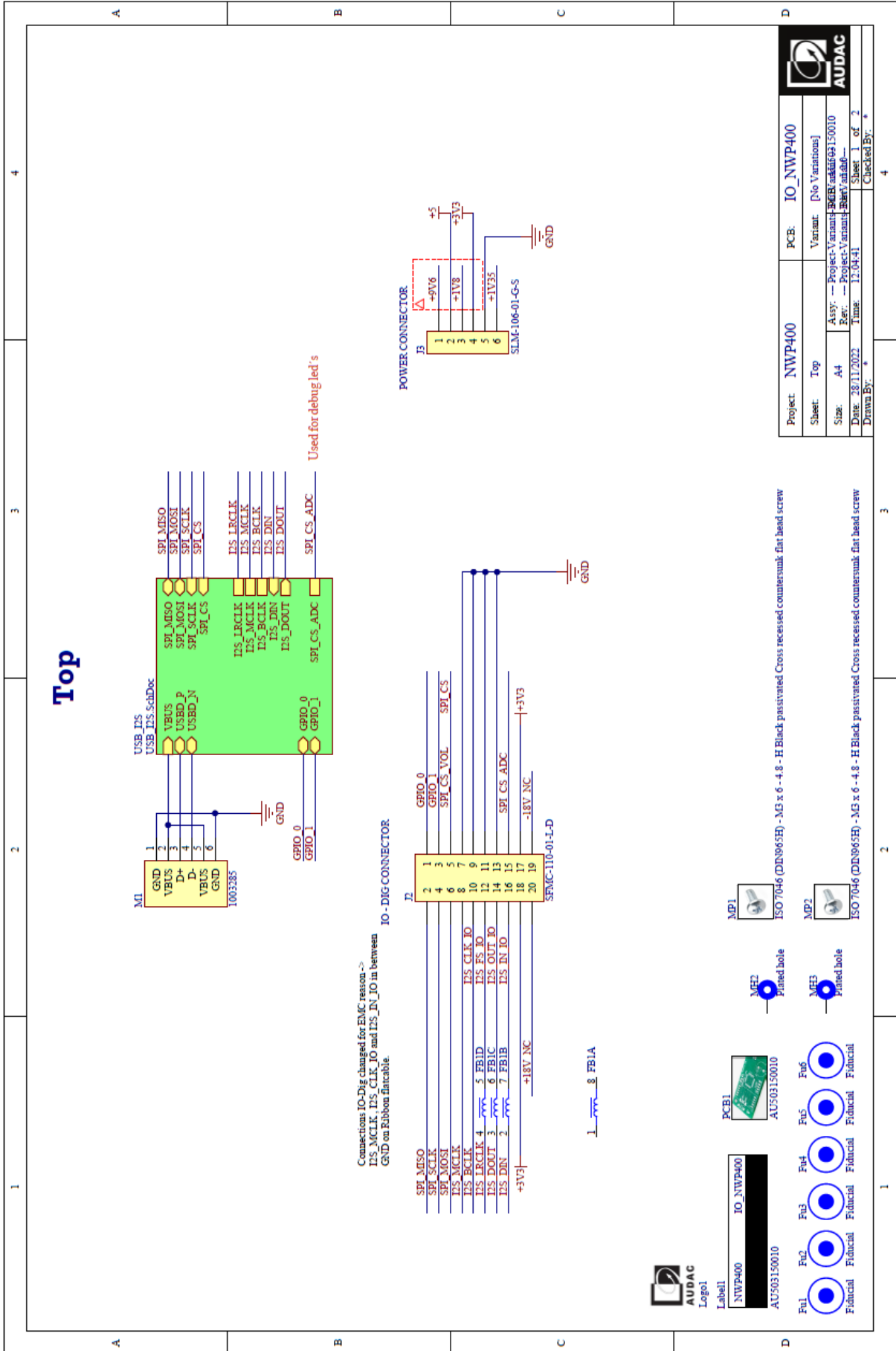
Bijlage A: Samenvatting vergelijking mogelijke microcontrollers

	XMOS XU208	STM32F0x2	STM32F401	PIC32MX7	ESP32-S2	RP2040
Processor type	xCore XU family	ARM Cortex-M0	ARM Cortex-M4	PIC32MX	Xtensa LX7	ARM Cortex-M0
Kernen	8	1	1	1	1	2
CPU klok	500 MHz	48 MHz	84 MHz	80 MHz	240 MHz	133 MHz
RAM	256 kB	6 kB	64 kB	32 kB	320 kB	264 kB
I²S	8x	1x	2x	2x	1x	PIO
IC Package	64-pin TQFP	TSSOP-20	48-pin UFQFPN	TQFP	QFN-56	QFN-56
Stukprijs/1000	€ 14,00	€ 1,39	€ 4,01	€ 4,26	€ 1,23	€ 0,76
USB Versie	2.0	1.1	2.0	1.1	1.1	1.1
Beschikbaarheid	Slecht	Zeer slecht	Zeer slecht	Slecht	Goed	Zeer goed

Bijlage B: Wandmodule PCB-stack

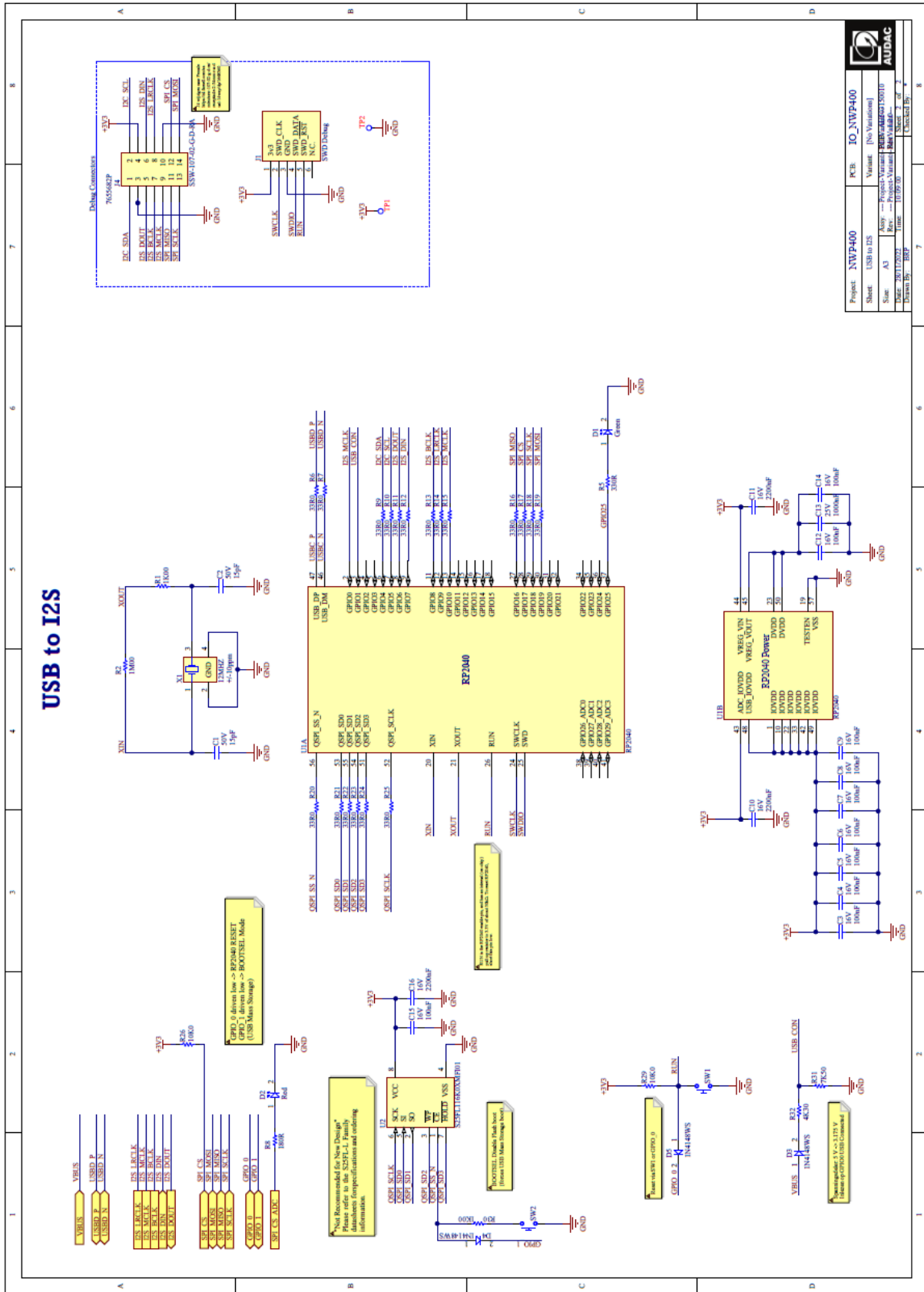


Bijlage C: Topschema uit Altium Designer



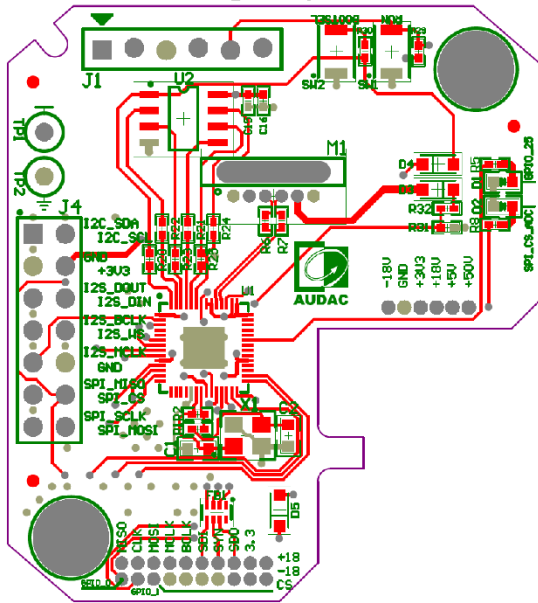
Project:	NWP400	PCB:	IO_NWP400
Sheet:	Top	Variant:	[No Variations]
Size:	A4	Asy:	--- Project-Variant-BGIB/au503150010
Date:	28/11/2022	Rev:	--- Project-Variant-BGIB/au503150010
Drawn By:	*	Time:	12:04:41
Checked By:	*	Sheet:	1 of 2

Bijlage D: USB naar I2S schema in Altium Designer

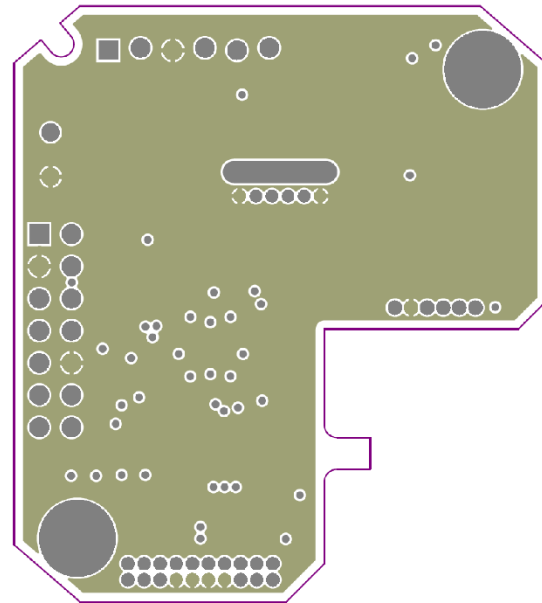


Bijlage E: PCB-lagen

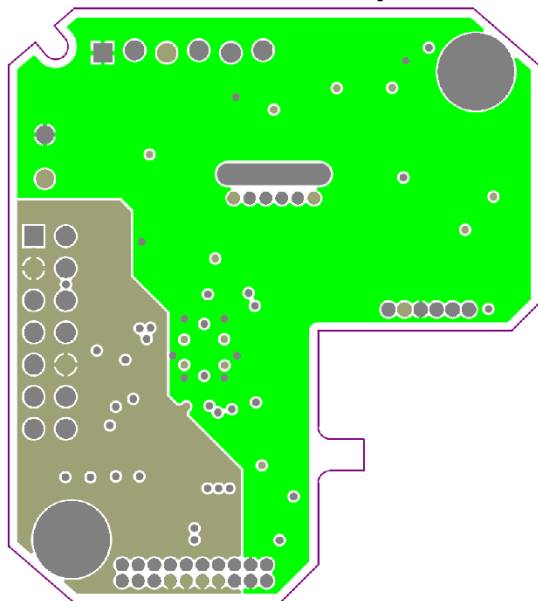
Top Layer



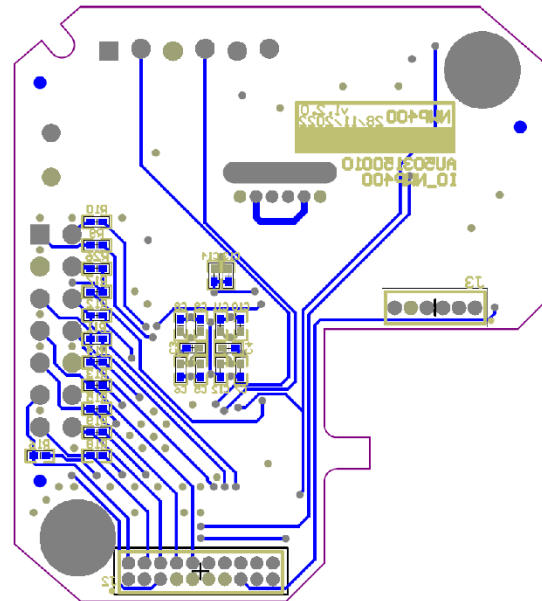
Ground Layer



Ground/3V3 Layer



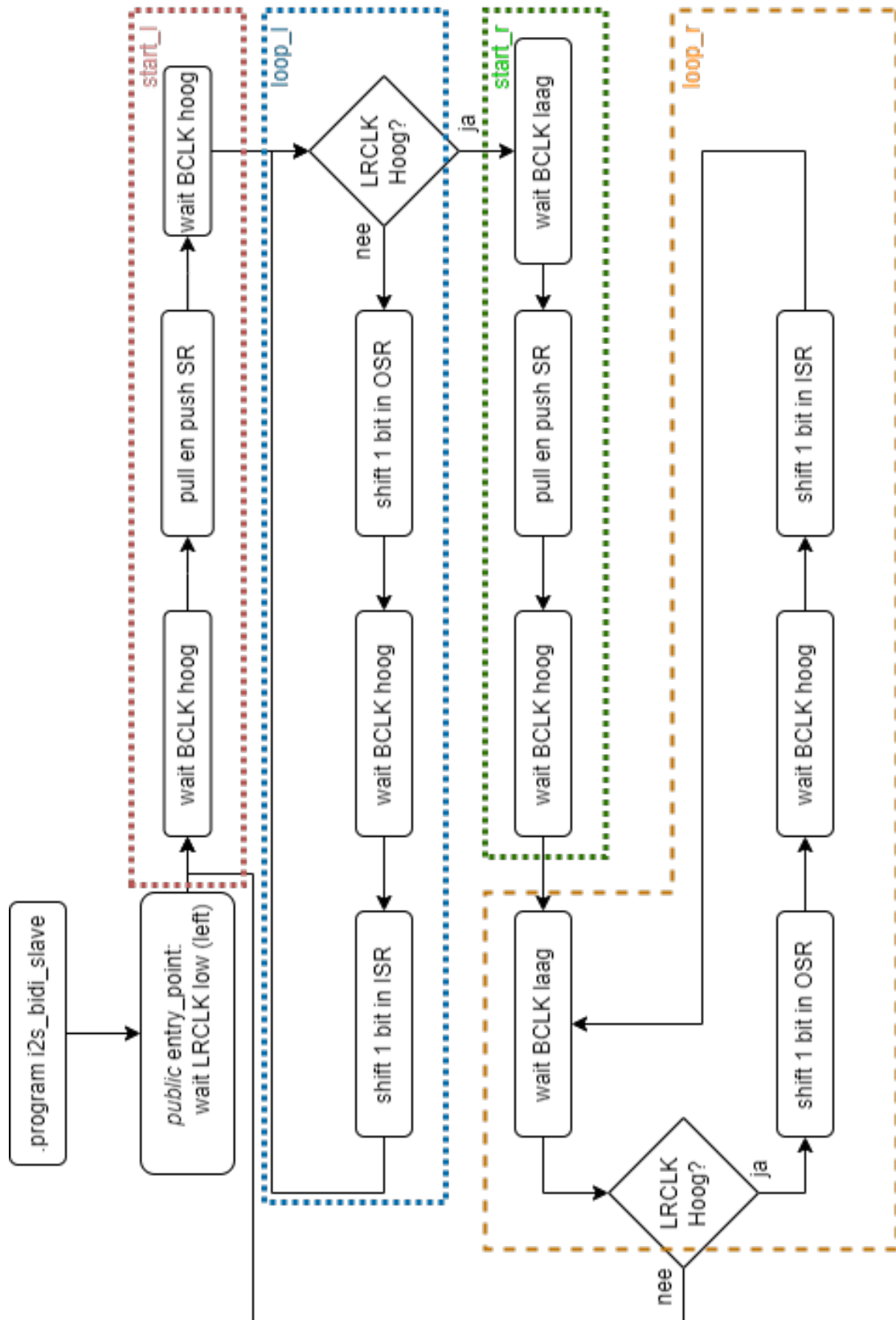
Bottom Layer



Bijlage F: BOM NWP400 USB naar I2S

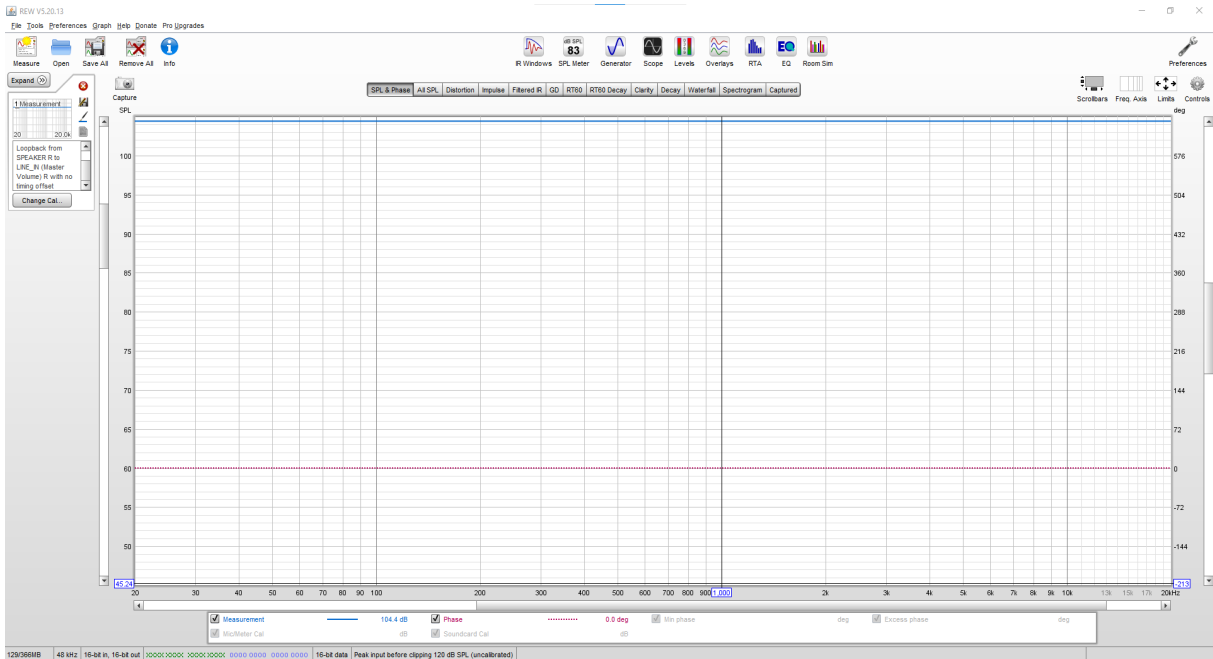
Fitted	Designator	Number	AU-number	Comment	Description	Footprint
Fitted	C1, C2	1002256		Capacitor	15pF /50V /10% ceramic multilayer COG (NPO) SMD 0603	0603_C
Fitted	C3, C4, C5, C6, C7, C8, C9, C12, C14, C15	1000266		Capacitor	100nF /16V /10% ceramic multilayer SMD COG (NPO) 0402	0402_C
Fitted	C10, C11, C16	1002536		Capacitor	2200nF /16V /10% ceramic multilayer SMD X5R 0402	0402_C
Fitted	C13	1002535		Capacitor	1000nF /25V /10% ceramic multilayer SMD X5R 0402	0402_C
Fitted	D1	1001567		Light Emitting Diode	LED SMD Green 0603	LED_0603_Green
Fitted	D2	1001568		Light Emitting Diode	LED SMD Red 0603	LED_0603_Red
Fitted	D3, D4, D5	1001015		Diode	1N4148WS Small signal Fast Switching diode 100V/0.15A SMD SOD-323	SOD-323
Fitted	FB1	1002999		Inductor	4x 600R / 100mA BEAD ARRAY SMD 0804	0804_B_8N
Fitted	J2	1002791		Mounting material	2x-10p Socket Straight P=1.27 H:4.57 THT	SFMC-110-01-L-D
Fitted	J3	1003166		Connector	6p Socket Straight P=1.27 H:4.57 THT	SOCKET 6x1.27 SLIM
Fitted	J4	1003166		Header 2x7 pitch 2.54mm 90° pins	Header 2x7 pitch 2.54mm 90° pins	Header 2x7.54mm 90°
Fitted	M1	1003285	AU513150020	Module	PCBA DWP345 USB CON ==> THT Pin Header 1x6 P1.27	Module USB3 con
Fitted	MP1, MP2	1001472		Mounting material	ISO 7046 [DIN965H] - M3 x 6 - 4.8 - H Black passivated Cross recessed counter	
Fitted	PCB1	1003290	AU503150010	NO COMPONENT	PCB	
Fitted	R1, R30	1000467		Resistor	1K00 /1% /62mW SMD resistor 0402	0402_R
Fitted	RS	1000458		Resistor	330R /1% /62mW SMD resistor 0402	0402_R
Fitted	R6, R7, R9, R10, R11, R12, R13, R14, R15, R16	1000438		Resistor	33R0 /1% /62mW SMD resistor 0402	0402_R
Fitted	R8	1000453		Resistor	180R /1% /62mW SMD resistor 0402	0402_R
Fitted	R26, R29	1000486		Resistor	10K0 /1% /62mW SMD resistor 0402	0402_R
Fitted	R31	1000483		Resistor	7K50 /1% /62mW SMD resistor 0402	0402_R
Fitted	R32	1002826		Resistor	4K30 /1% /62mW SMD resistor 0402	0402_R
Fitted	SW1, SW2	1003186		Switches	Switch Tactile N.O. SP5T Round Button PC Pins 0.05A 12VDC 1.5N SMD	Tactile-switch BSU-1000P
Fitted	U1	RP2040		RP2040		QFN-56 0.4mm
Fitted	U2	1001698		Integrated Circuit	S25F116IK0XMF01 16Mbit SPI Memory Flash SMD SO-8	SOIC-8
Fitted	X1	1002707		Crystal and oscillator	12MHz 12pF +/-10ppm Crystal Resonator SMD 4-Pad 3.2mmx2.5mm	XTAL_OSC_3.2x2.5
Not Fitted	R2	1000522		Resistor	1M00 /1% /62mW SMD resistor 0402	0402_R

Bijlage G: PIO-flowchart



Bijlage H: REW-metingen

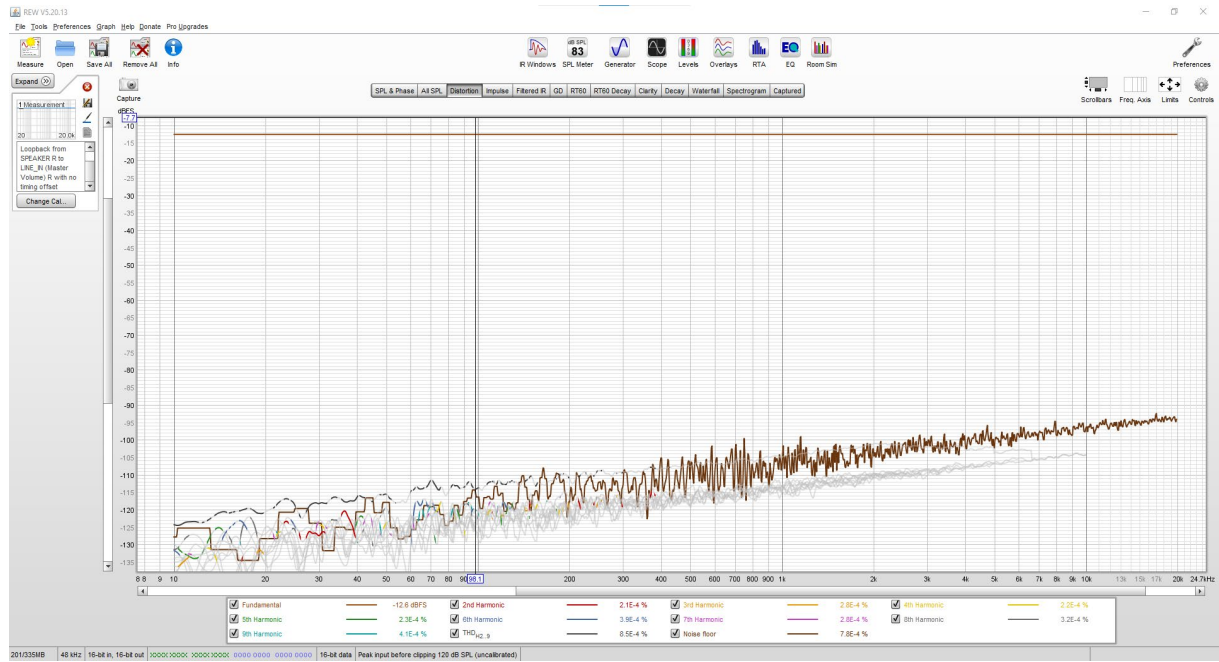
SPL en phase rechterkanaal



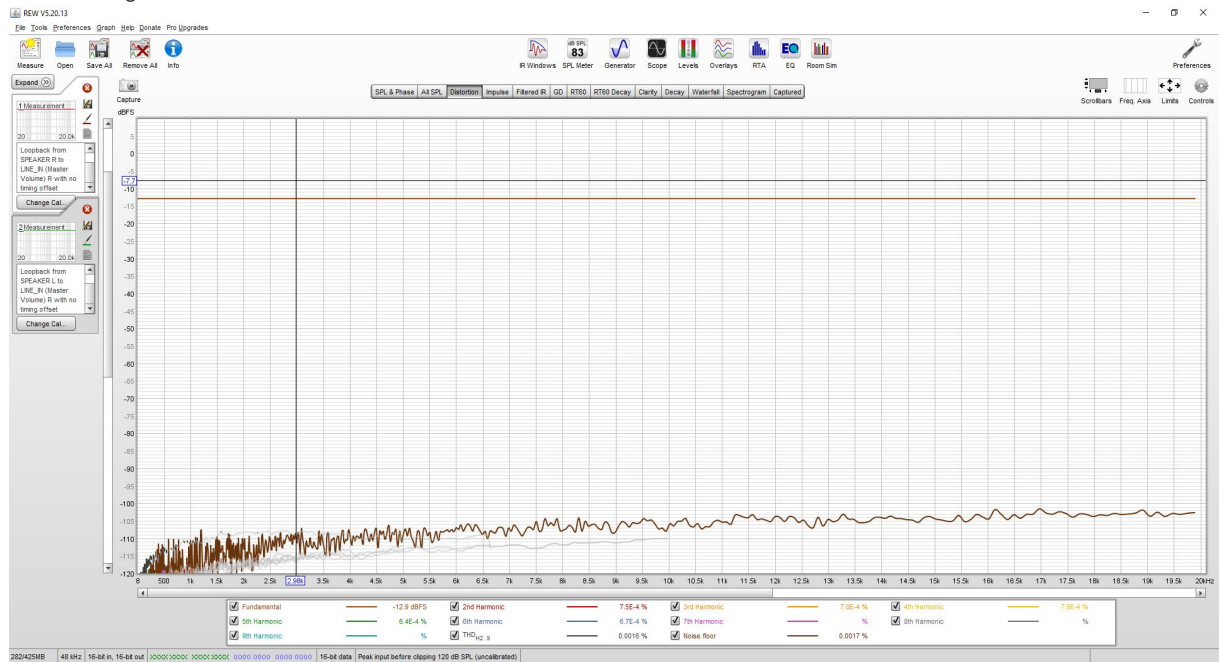
SPL en phase linkerkanaal



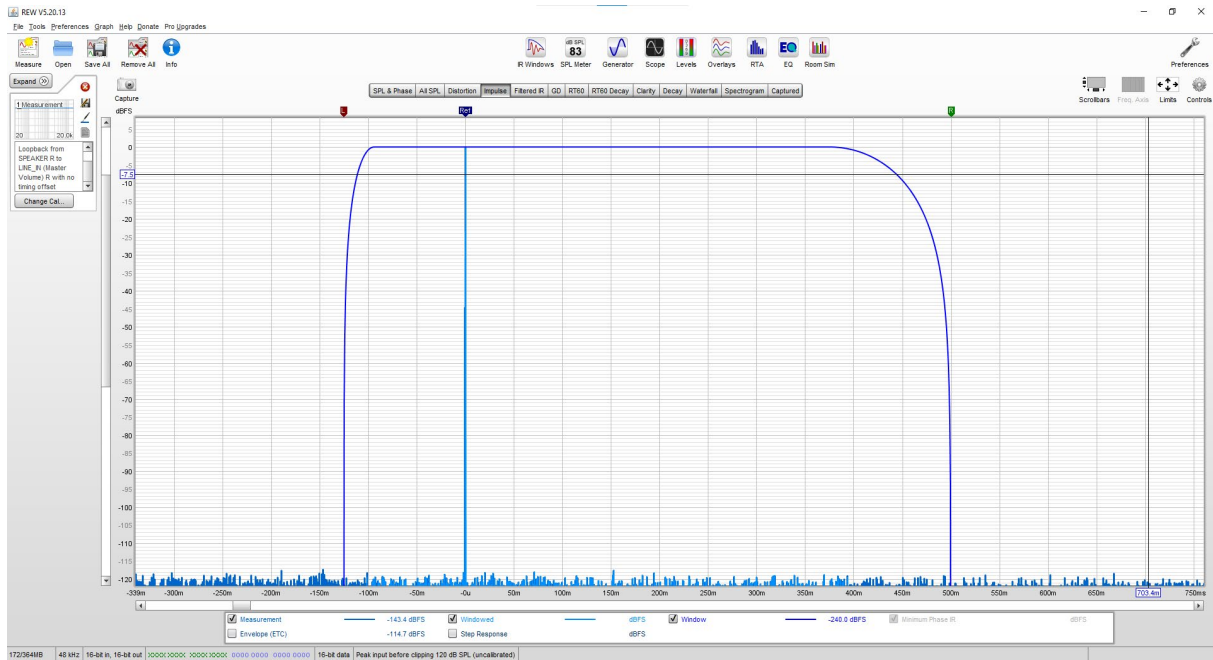
Vervorming rechterkanaal



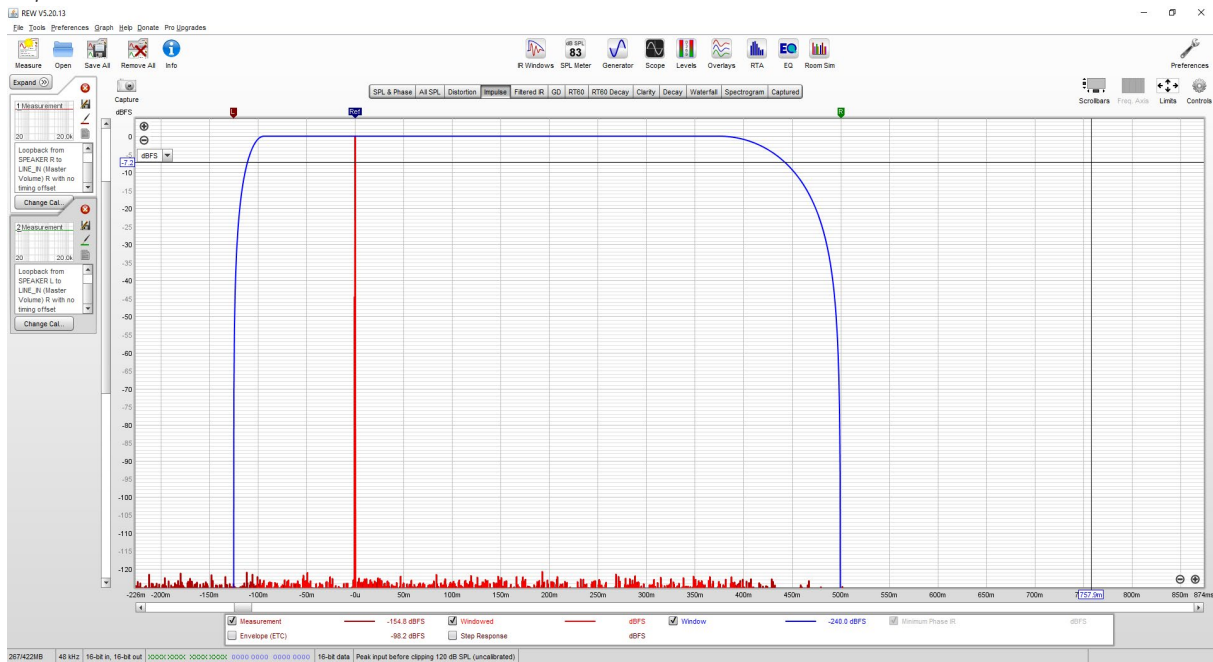
Vervorming linkerkanaal



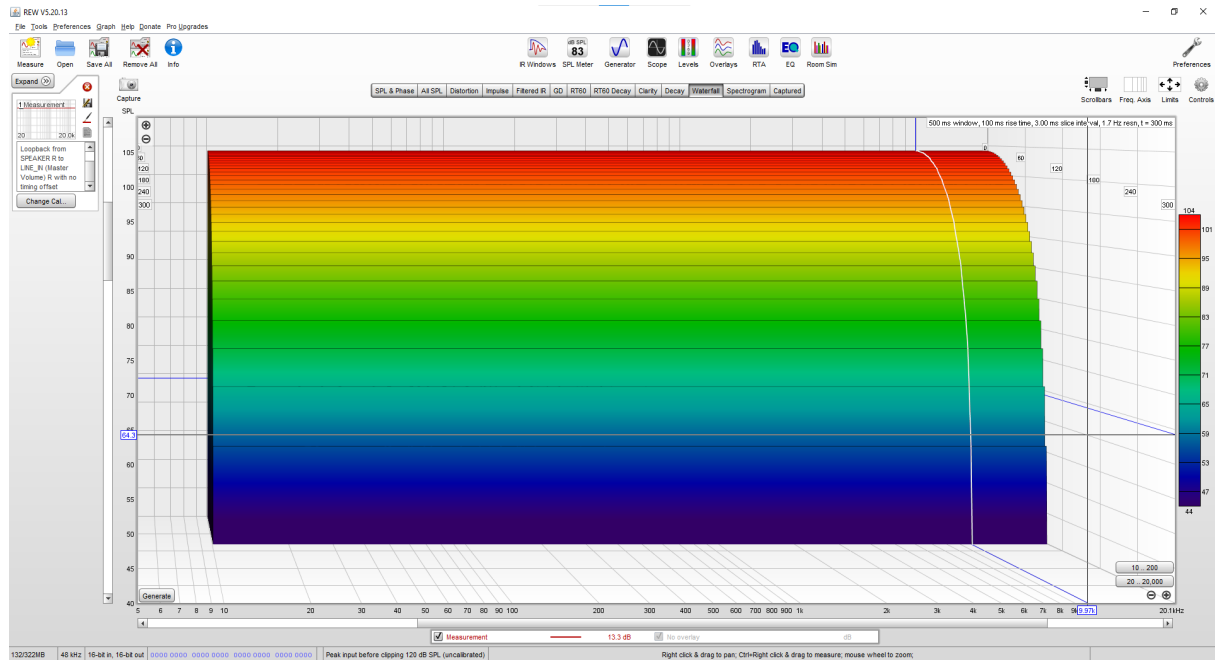
Impulse rechterkanaal



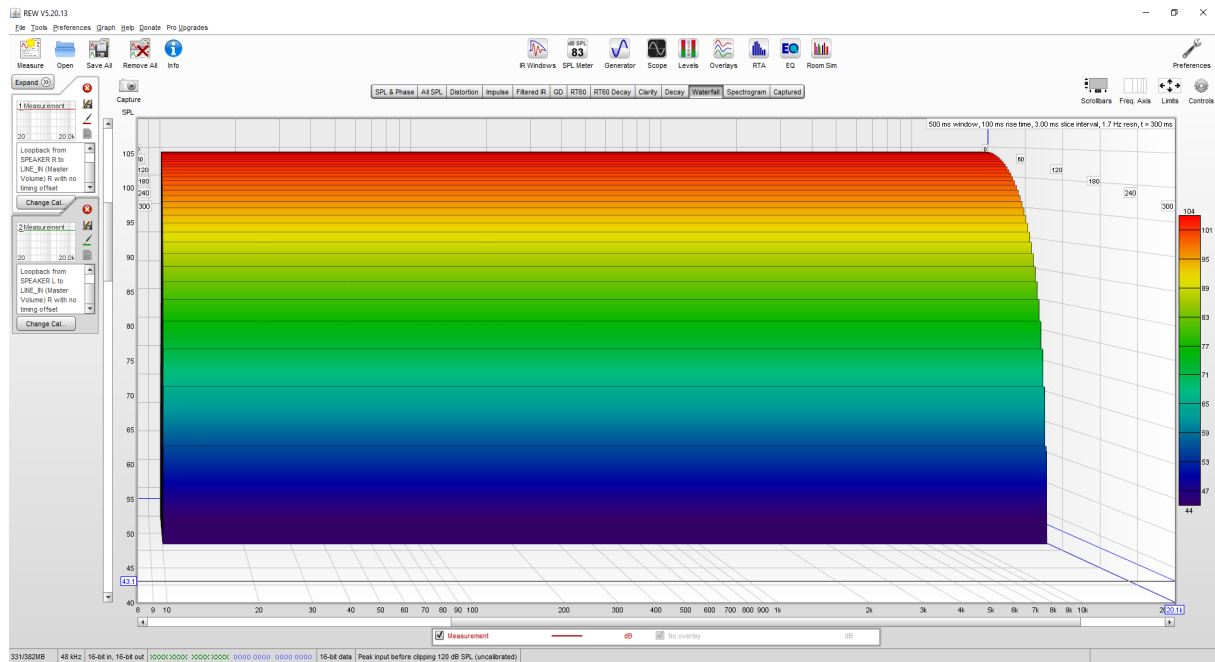
Impulse linkerkanaal



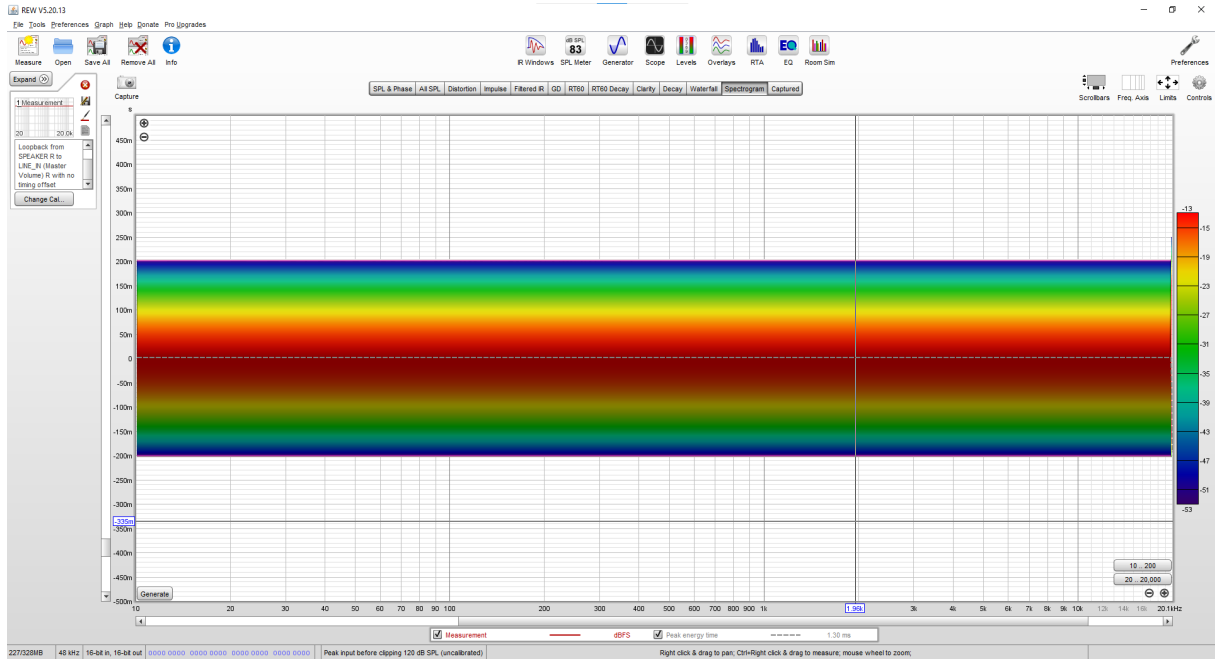
Waterval rechterkanaal



Waterval linkerkanaal



Spectrogram rechterkanaal



Spectrogram linkerkanaal

