

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Zpracování audio signálu na mikrokontroleru

Durák Marek

Vedoucí práce: Ing. Petr Honzík, Ph.D.
Studijní program: Elektronika a komunikace
Květen 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Durák** Jméno: **Marek** Osobní číslo: **507601**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Zpracování audio signálu na mikrokontroleru

Název bakalářské práce anglicky:

Audio Signal Processing on Microcontroller

Pokyny pro vypracování:

Seznamte se se základními postupy implementace zpracování akustických signálů. Zhodnoťte možnosti implementace zpracování signálů na dostupných mikrokontrolerech z hlediska požadavků na výpočetní výkon, komunikace s AD a DA převodníky s využitím sběrnice I2S a uživatelského rozhraní. Pro vybraný mikrokontroler navrhnete systém pro zpracování signálu v reálném čase včetně příslušného ovládání uživatelem. Implementujte vybrané algoritmy zpracování a ověřte funkčnost celého systému nejprve na vývojové desce, pak navrhnete vlastní hardwarové řešení.

Seznam doporučené literatury:

[1] Jiří Jan: Číslcová filtrace, analýza a restaurace signálů, Akademické nakladatelství, VUTIU, 2002 ISBN: 8021429119
[2] I2S bus specification [online]. Philips Semiconductors, 1986 [cit. 2023-01-19]. Accessible from:
<https://www.sparkfun.com/datasheets/BreakoutBoards/I2SBUS.pdf>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Honzík, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.02.2024**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Petr Honzík, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat panu Ing. Petru Honzíkovi, Ph.D. za veškerou ochotu, důvěru a podporu při vedení práce. Moc si toho vážím. Děkuji i své rodině za celkovou oporu při studiu, a stejně tak mým blízkým přátelům.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl všechny použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze 24. května 2024,

Abstrakt

Tato bakalářská práce se zabývá návrhem a stavbou zařízení pro číslicové zpracování akustického signálu v reálném čase. V práci jsou probírány jednotlivé úkony od výběru mikrokontroleru pro zpracování signálu, přes implementaci samotných efektů či digitálních filtrů až po realizaci samotné desky plošných spojů či návrhu uceleného zařízení s uživatelským rozhraním. Výstupem této práce je tak zařízení, které je schopné přijmout standardní linkový signál, popřípadě signál z elektrické kytary. Následně na něj aplikovat vybraný efekt či filtr uživatelem a odeslat jej na linkový, popřípadě sluchátkový výstup.

Klíčová slova: mikrokontroler, kodek, číslicové zpracování signálu v reálném čase, zvuk, filtrace, číslicové filtry

Vedoucí práce: Ing. Petr Honzík, Ph.D.

Abstract

This bachelor's thesis deals with the design and construction of a device for digital processing of acoustic signals in real-time. The thesis covers various tasks, starting from the selection of a microcontroller for signal processing, through the implementation of specific effects or digital filters, to the realization of the printed circuit board and the design of a complete device with a user interface. The outcome of this thesis is a device capable of receiving a standard line signal or a signal from an electric guitar, applying a user-selected effect or filter to it, and pass on to a line or headphone output.

Keywords: microcontroller, codec, real time digital signal processing, audio, effects, digital filters

Title translation: Audio Signal Processing on Microcontroller

Obsah

1 Úvod	1	4.1.5 Vstupy a výstupy	28
1.1 Cíl bakalářské práce	1	4.2 Modifikace pro finální produkt	29
1.2 Dostupná řešení z trhu	2	4.2.1 Mikrokontroler	29
1.3 Původní návrh řešení	3	4.2.2 Napájení	30
1.4 Návrh řešení	3	4.2.3 Další úpravy	30
2 Teoretický rozbor	5	4.3 Realizace finálního produktu	30
2.1 Zpracování akustických signálů	5	4.3.1 Modelování	32
2.1.1 Vzorkování	5	4.3.2 Fusion 360	32
2.1.2 Kvantizace	7	4.3.3 3D tisk pouzdra	32
2.2 I2S komunikace	8	4.3.4 UltiMaker Cura	33
2.3 Průzkum trhu mikrokontroleru	9	5 Diskuze	35
2.3.1 Mikrokontroler	9	5.1 Porovnání teoretických hodnot	
2.3.2 Nároky na mikrokontroler	9	filtrů s naměřenými hodnoty	35
2.3.3 Současná situace trhu	11	5.2 Známé nedostatky zařízení	36
2.4 Popis vybraných efektů a filtrů	11	5.3 Další možné úpravy zařízení	36
2.4.1 IIR	12	Závěr	37
2.4.2 FIR	13	Literatura	39
2.4.3 Ekvalizace vybraných pásem	14	A Seznam použitých zkratk	41
2.4.4 Reverb	14	B Schéma vývojové DPS	43
3 Softwarové zpracování	17	B.1 Část první	43
3.1 Tvorba softwaru	17	B.2 Část druhá	44
3.1.1 STM32CubeIDE	17	C Schéma Antinode DPS	45
3.1.2 MATLAB	17	C.1 Část první	45
3.2 Systém zpracování vzorků	18	C.2 Část druhá	46
3.3 Ekvalizace vybraných frekvenčních		C.3 Část třetí	47
pásem	19	C.4 Část čtvrtá	48
3.3.1 Stanovení analogového		D Seznam ostatních příloh	49
ekvivalentu a jeho úpravy	19		
3.3.2 Bilineární transformace	20		
3.3.3 Vyčíslení koeficientů pro			
požadované filtry	20		
3.4 Implementace algoritmů pro			
úpravu vzorků	22		
3.4.1 Stereo	22		
3.4.2 Mono	22		
3.4.3 Úprava hlasitosti	23		
3.4.4 Reverb	23		
3.4.5 Filtrování vybraných pásem	24		
3.5 Uživatelské rozhraní	25		
4 Hardwarové zpracování	27		
4.1 Tvorba vývojové desky	27		
4.1.1 Altium Designer	27		
4.1.2 Návrh DPS	27		
4.1.3 Klíčové komponenty DPS	27		
4.1.4 Uživatelské rozhraní	28		

Obrázky

1.1 Zjednodušené blokové schéma systému	1
2.1 Sinusový signál s frekvencí 100 Hz a amplitudou 1	5
2.2 Původní sinusový signál navzorkován s frekvencí 2 kHz	6
2.3 Původní navzorkovaný signál kvantizovaný s krokem 0,25.	7
2.4 Ukázka I2S komunikace	8
2.5 Blokový diagram struktury IIR filtru	12
2.6 Blokový diagram struktury FIR filtru	13
2.7 Převzaté blokové schéma algoritmu v efektu reverb	15
3.1 Převzaté blokové schéma systému pro zpracování vzorků akustického signálu z [10]	18
3.2 Navržené peaking filtry situovány do jednoho grafu s maximální zesílením i potlačením vybraných pásem	21
3.3 Fotografie zobrazovaných dat zařízení	25
4.1 Rozložení komponentů vývojové DPS	28
4.2 OLED display převzato z www.drakek.cz	29
4.3 Zapojení OZ jako napěťového sledovače pro levý kanál	29
4.4 Vrchní strana DPS Antinode ...	31
4.5 Spodní strana DPS Antinode ...	31
4.6 Vygenerovaný finální vzhled 3D modelu krabičky Antinode	32
4.7 Fotografie celého zařízení ze spodní strany bez spodního krytu	33
5.1 Postupně změřené peaking filtry situovány do jednoho grafu	35

Tabulky

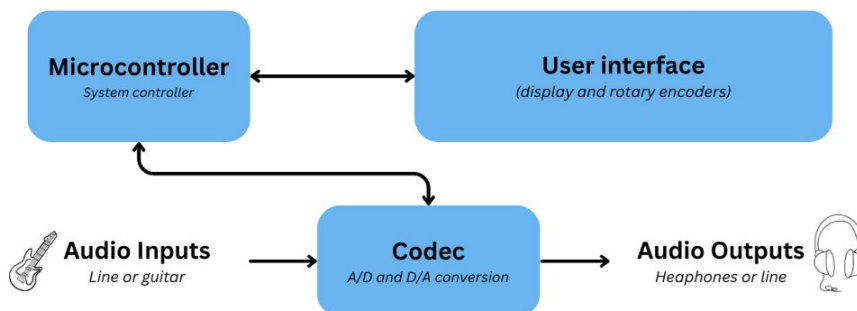
2.1 Porovnání vybraných mikrokontroléru	11
4.1 Porovnání mikrokontroleru vývojové desky a finálního zařízení	30

Kapitola 1

Úvod

1.1 Cíl bakalářské práce

Cílem této bakalářské práce je navrhnout kompaktní zařízení pro širokou škálu libovolných úprav a efektizaci zvukových signálů z elektrické kytary, popřípadě linkového signálu v reálném čase. Tedy ideálně s co nejmenší prodlevou. V průběhu přenosu signálu libovolně měnit efekty, upravovat parametry filtrů. Zároveň za pomoci displeje sledovat momentálně nastavený efekt a jeho parametry. Při návrhu je též věnována velká pozornost cenové dostupnosti, avšak nikoli na úkor kvality přenosu.



Obrázek 1.1: Zjednodušené blokové schéma systému

1.2 Dostupná řešení z trhu

Běžně dostupné komerční zařízení na trhu plnicí obdobnou funkci bychom mohli rozdělit do následujících kategorií. Kategorie, která zde nebyla pokryta, zahrnuje jednoúčelové profesionální zařízení pro úpravu zvuku, jako například Denoisery (odšumovače), Enhancery (zkreslovače signálu) a další. [1]

Pro úpravu akustického signálu můžeme najít zejména

- pedálové efekty pro kytaru,
- multiefekty pro kytaru,
- samostatné ekvalizéry,
- efektové procesory.

Pedálové efekty nabízí zpravidla právě jeden efekt, jedná se však o velmi robustní zařízení. Jsou určeny výhradně pro vstup kytary, až na výjimky nenabízejí vstup například linkového signálu. Lze je řadit sériově za sebe. Jelikož se jedná o kytarové příslušenství většinou pro jednoho kytaristu, nabízejí pedálové efekty zpracování převážně jednoho kanálu (mono).

Multiefekty obsahují více kytarových efektů v jednom pouzdru. Zvuk je uvnitř zařízení navzorkován a následně číslicově zpracován procesorem. Většinou nabízí zpracování právě jednokanálového signálu, levnější zařízení zpravidla nenabízí další stahování efektů či tvorbu vlastních. Lze tak hlavně měnit parametry již nahraných efektů.

Samostatné ekvalizéry nabízí filtrování akustického signálu v reálném čase, mohou zesilovat, popřípadě potlačovat části pásma frekvenčního spektra. Větší četnost pásem zařizuje podrobnější selektivitu potlačení či zesílení chtěné frekvenční složky. Nejčastěji se na trhu můžeme setkat s dvoukanálovými (levý a pravý kanál), to však není pravidlem, existují i ekvalizéry pro úpravu vícekanálového zvuku. Ve velké většině jde o jednoúčelové zařízení, až na výjimky neobsahují další efekty.

Efektové procesory nabízí širokou škálu efektů nejčastěji pro dvoukanálový akustický signál. Jsou zároveň částečně podobné kytarovým multiektům, liší se zpravidla počtem kanálů ke zpracování. Další rozdíly bychom našli například v zaměření efektů. Efekty pro elektrickou kytaru se zaměřují hlavně na úpravu signálu z kytary. Efektové procesory se zaměřují na zpracování komplexnějšího akustického signálu, například nahrávky celé kapely. Mezi jejich nejznámější efekty bychom mohli zařadit reverb, denoiser nebo taky grafický, popřípadě parametrický ekvalizér.

1.3 Původní návrh řešení

Jádrem původního návrhu byl mikrokontrolér od firmy Raspberry Pi model pico, který byl vybrán pro svoji příznivou cenu, vhodný výpočetní výkon a předešlé zkušenosti s tímto mikrokontrolerem. Mikrokontroler byl vybrán i přes hardwarovou absenci periferie I2S, která byla zvolena pro přenos diskretizovaného akustického signálu. Pro softwarové vytvoření I2S existuje totiž možnost využít programovatelnou periferii PIO, která se programuje v jazyku podobném Assembleru, i přes již existující knihovny od uživatelů daného MCU se tato možnost při řešení problému s komunikací ukázala jako velmi časově náročná a ne zcela intuitivní.

1.4 Návrh řešení

Návrh nového řešení používá mikrokontroler od firmy STMicroelectronics, který má periferii I2S již integrovanou. Zvažovány byly i mikrokontrolery od firmy Espressif, avšak nakonec bylo rozhodnuto zůstat u ARM architektury, právě pro předešlé zkušenosti s mikrokontrolery využívajícími tuto architekturu a ještě konkrétněji u mikrokontroleru výrobce STMicroelectronics pro již získané zkušenosti z předešlých projektů.

Kapitola 2

Teoretický rozbor

2.1 Zpracování akustických signálů

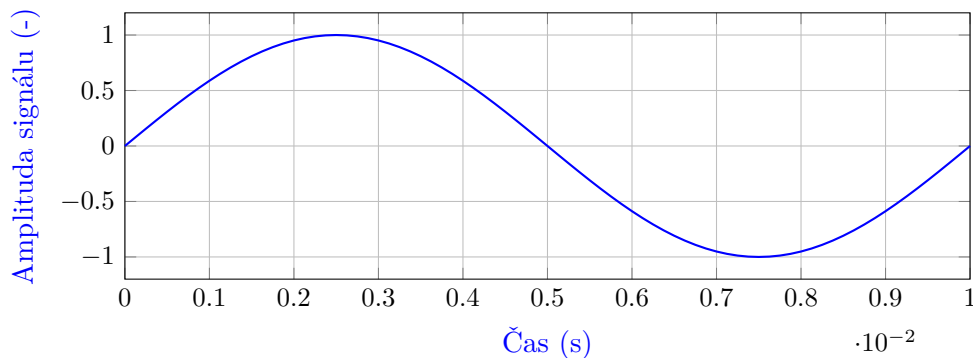
V této části bude vysvětleno, jak získat ze vstupního spojitého signálu 2.1 vzorky pro jejich další zpracování. [2]. Tedy převést v čase spojitý zvukový signál do času diskrétního. Výstupy této části zahrnují následné určení vzorkovací frekvence a bitové hloubky systému. [3]

Grafy v této části byly vytvořeny v prostředí MATLAB, jejich data byla následně exportována do L^AT_EX a s pomocí balíčku `pgfplots` byly vytvořeny jejich finální verze. Doporučení ke zvýšení kvality grafů vycházelo z [4].

2.1.1 Vzorkování

Vzorkování spojitého signálu, jedna z částí převodu analogového signálu na diskrétní. Tedy jej navzorkovat s jistou vzorkovací periodou. Dle vztahu (2.1) je možné získat ze vzorkovací periody T_s vzorkovací frekvenci f_s

$$f_s = \frac{1}{T_s}. \quad (2.1)$$



Obrázek 2.1: Sinusový signál s frekvencí 100 Hz a amplitudou 1

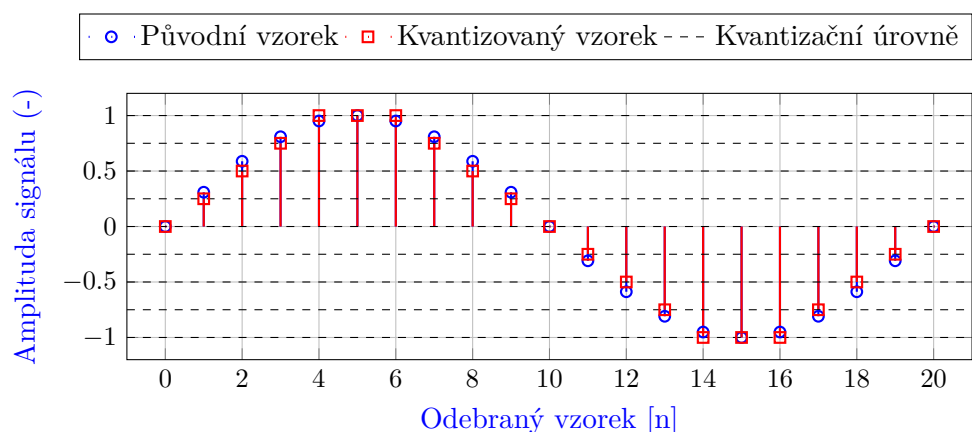
44.1, 48, 88.2 a 96 kHz [4]. Pro navrhovanou aplikaci byl stanoven $f_s = 48$ kHz.

2.1.2 Kvantizace

Kvantizace je proces přiřazení definované hodnoty k právě získanému vzorku. Tato operace je nejvíce ztrátová z celého procesu diskretizace spojitého signálu. V rámci kvantizace je přiřazena číslu konečná hodnota z nekonečného množství hodnot. Přiřazení nejbližší definované hodnoty $x_Q[n]$ ke skutečné hodnotě $x[n]$ lze sledovat na vztahu (2.3). Když $x_Q[n] \neq x[n]$, potom je dopouštěna kvantizační chyba $e[n]$ [3]

$$x_Q[n] = x[n] + e[n]. \quad (2.3)$$

Nyní bude analyzováno kvantizování navzorkovaného signálu 2.2, aby mohl být dokončen proces získání vzorků diskretizací spojitého signálu pro jejich další zpracování.



Obrázek 2.3: Původní navzorkovaný signál kvantizovaný s krokem 0,25.

Výsledně získané vzorky po kvantizaci, lze sledovat na obrázku 2.3, kvantizační krok Q_s byl pro demonstraci stanoven jako 0,25. Tento krok odpovídá bitové hloubce 3, bitovou hloubku b_d se znalostí rozkmitu amplitudy signálu A_{pp} a Q_s lze získat ze vztahu (2.4)

$$b_d = \log_2 \left(\frac{A_{pp}}{Q_s} \right). \quad (2.4)$$

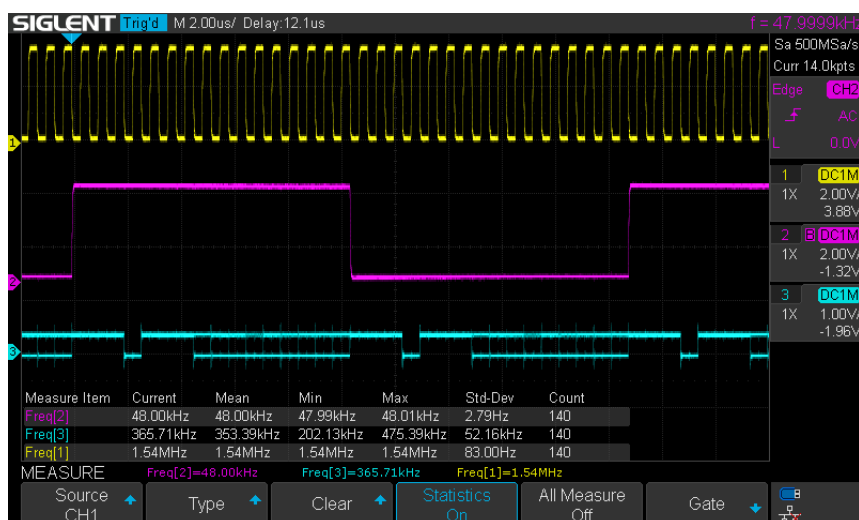
Důsledek kvantizační chyby $e[n]$ je vznik kvantizačního šumu, zvyšování bitové hloubky b_d nám tak může pomoci se snížením tohoto druhu zkreslení. Standartní bitové hloubky používané pro přenos akustických signálů jsou $b_d = 16, 24$ a 32 bitů. Pro navrhovanou aplikaci byla stanovena $b_d = 24$ bitů. [1]

2.2 I2S komunikace

I2S je seriová sběrnice, vytvořena právě pro přenos diskretizovaných akustických signálů mezi integrovanými obvody. [6] Právě tato sběrnice byla vybrána pro přenos diskretizovaných vzorků signálu mezi kodekem (A/D a D/A převodník) a MCU. Sběrnice je postavena na architektuře master-slave, jedno ze zařízení se tak stará o generaci řídicích kmitočtů. Jedná se o robustní tří vodičovou sběrnici, obsahující následující signály:

- SCK - Continuous serial clock (Taktovací frekvence sběrnice),
- WS - Word select (Výběr kanálu),
- SD - Serial data (Sériový přenos dat).

Sběrnice je určena pro přenos dvoukanalového signálu (stereo). Výběr jednotlivých kanálů má na starost linka WS, která úrovněmi logické nuly a logické jedničky přepíná mezi kanály. Kde logická nula vybírá levý kanál a logická jednička vybírá kanál pravý. Je zároveň možné tuto sběrnici použít pro přenos jednoho kanálu (mono). V této práci jsou však využívány oba kanály.



Obrázek 2.4: Ukázka I2S komunikace

Probíhající I2S komunikaci zachycenou na osciloskopu lze sledovat na obrázku 2.4, kde linka **SCK** je vyobrazena barvou žlutou. Linka **WS** je vyobrazena barvou fialovou a poslední linka **SD** je vyobrazena barvou modrou.

Linka WS je řízena právě vzorkovacím kmitočtem, v jedné periodě totiž vyzvedne vzorek z obou kanálů. Můžeme tak konstatovat, že vzorkovací frekvence užitá v této komunikaci je $f_s = 48 \text{ kHz}$. Jeden vzorek je složen z jednotlivých bitů, které reprezentují jeho hodnotu. Bity jsou zapisovány

na linku SD taktovacím kmitočtem SCK a řazeny ve formátu MSB first.

Taktovací kmitočet SCK s požadovanou vzorkovací frekvencí a bitovou hloubkou je nastaven dle rovnice (2.5), kde číslo dvě reprezentuje počet kanálů. V probíhající komunikaci na obrázku 2.4 je použita bitová hloubka 16 bitů. Změřený a zaokrouhlený kmitočet osciloskopem lze ověřit dosazením do rovnice (2.5), výstupem potom dostáváme $SCK = 1,536 \text{ MHz}$

$$SCK = 2 \cdot f_s \cdot b_d. \quad (2.5)$$

Nezmíněná volitelná linka ve sběrnici je MCLK (Hlavní taktovací frekvence), která je v tomto navrhovaném systému vyžadována kodekem. Většina I2S zařízení je schopna si tento kmitočet generovat sama, využití tohoto kmitočtu může být například k internímu řízení A/D a D/A převodníků.

V rámci I2S sběrnice je rovněž možné využít dvě linky SD pro duplexní provoz. Právě této možnosti je využito v navrženém systému. Kde jedna z linek SD přenáší data z A/D převodníku do MCU a druhá linka přenáší již zpracovaná data z MCU do D/A převodníku.

2.3 Průzkum trhu mikrokontroleru

Mikrokontrolér se v tomto projektu stará o veškeré zpracování dat, nakonfigurování kodeku a v neposlední řadě i o uživatelské rozhraní, jeho výběr je tedy klíčový pro další sekce. V této části jsou zároveň popsány nároky na MCU, podle kterých byl následně výběr prováděn.

2.3.1 Mikrokontroler

MCU je jednočipový počítač, tento integrovaný obvod obsahuje jeden nebo více CPU, společně spolu s pamětí podle jeho vnitřní architektury může mít jeden či více typů paměti. Například operační paměť RAM a paměť pro program, který je na něm uložen. To může být například typ paměti FLASH nebo EEPROM. MCU dále může obsahovat programovatelné vstupní/výstupní obvody, A/D a D/A převodníky, sběrnice pro komunikaci s ostatními zařízeními, časovače, DMA kontrolér a další vnitřní bloky podle potřeby odběratele. [7]

2.3.2 Nároky na mikrokontroler

Požadavky na MCU vycházejí z kvality přenosu akustického signálu, periférií pro nakonfigurování přenosu, využívaných sběrnic pro funkci celé aplikace a výpočetní náročnosti samotných efektů a uživatelského rozhraní. Důležité parametry, které přímo souvisí s požadavky na výběr MCU, jsou shrnuty zde:

2.3.3 Současná situace trhu

V této části jsou zhrnuty klíčové parametry vybraného mikrokontroléru od firmy STMicroelectronics. Pro porovnání s vybraným MCU, jsou zde ještě vysázeny parametry dvou dalších mikrokontrolerů od stejné firmy. Rozložení parametrů dle kterých byl výběr proveden se nachází v tabulce 2.1.

Mikrokontrolér	STM32L0x0	STM32F411	STM32H725
Procesor	ARM Cortex-M0+	ARM Cortex-M4	ARM Cortex-M7
Frekvence CPU	až 32 MHz	až 100 MHz	až 550 MHz
Flash paměť	až 128 kB	až 512 kB	až 1 MB
RAM	až 20 kB	128 kB	až 512 kB
GPIO piny	až 51	až 81	až 168
I2C	1	až 3	až 5
I2S	ne	až 5	až 4
FPU	ne	ano	ano
DMA	ano	ano	ano
Napájení	1,8 až 3,6V	1,7 až 3,6V	1,62 až 3,6V
Tržní Cena	≈ 50 Kč	≈ 120 Kč	≈ 300 Kč

Tabulka 2.1: Porovnání vybraných mikrokontrolerů

Finální rozhodnutí směřovalo právě k STM32F411, řada mikrokontrolerů F4 již patří do nejvýkonnějších mikrokontrolerů od dané firmy. Mikrokontroler byl vybrán v pouzdru 64-pin LQFP, pro snazší osazení v konfiguraci s velikostí FLASH paměti 512 kB. Vybraný MCU kromě veškerých požadovaných parametrů disponuje také skvělou cenou nebo velmi nízkou spotřebou 100 $\mu\text{A}/\text{MHz}$ při 1,7 V [9]. To nabízí možnost v dalších rozšířeních tento mikrokontroler provozovat z baterie. V tabulce 2.1 je vybraný MCU porovnán spolu s jedním z nejúspěšnějších (STM32L0x0) a s jedním z nejvýkonnějších (STM32H725).

2.4 Popis vybraných efektů a filtrů

V této části jsou popsány struktury IIR, FIR filtrů a následně efekty využívající jednu z těchto struktur. Tyto dvě struktury digitálních filtrů nachází široké uplatnění v diskrétní doméně. Podobně jako v analogové doméně, jsou tyto filtry navrženy pro odfiltrování nechtěných frekvenčních složek signálu.

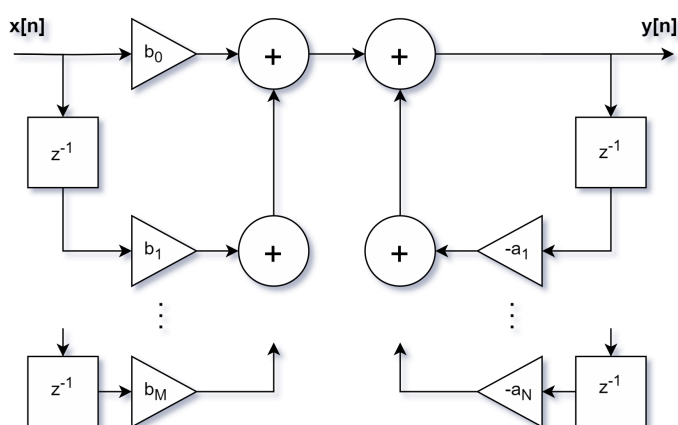
2.4.1 IIR

Filtry s nekonečnou impulsní charakteristikou (IIR) jsou pojmenovány podle odezvy na jednotkový impuls δ . Z odezvy na δ dostáváme totiž teoreticky nekonečnou impulsní charakteristiku. Můžeme se také setkat se slovním spojením rekurzivní typ digitálního filtru, IIR filtry jsou totiž postaveny na zpětnovazebním systému, a je tak nutné pracovat s jejich stabilitou. Systém IIR filtru používá tedy uložené výstupní vzorky ke zpracování přicházejících vstupních vzorků. Výstupní vzorek je spočítán diferenční rovnicí 2.6, která popisuje diskrétní konvoluci rekurzivní a nerekurzivní části filtru. Diferenční rovnice je zároveň shodná s realizací přímého algoritmu [2]

$$y[n] = \sum_{k=0}^M b_k \cdot x[n - k] - \sum_{k=1}^N a_k \cdot y[n - k], \quad (2.6)$$

kde:

- $y[n]$ je aktuální výstupní vzorek,
- $x[n]$ je aktuální vstupní vzorek,
- a_k a b_k jsou koeficienty filtru,
- N je počet zpoždění rekurzivní části filtru,
- M je počet zpoždění nerekurzivní části filtru.



Obrázek 2.5: Blokový diagram struktury IIR filtru

Realizace blokové struktury IIR filtru je vyobrazena na obrázku 2.5, blokový diagram je složen z jednotlivých zpožďovacích členů-paměťových registrů (z^{-1}) a k nim přiřazených zesílení, tedy koeficientů filtru (a_k, b_k). [5]

2.4.2 FIR

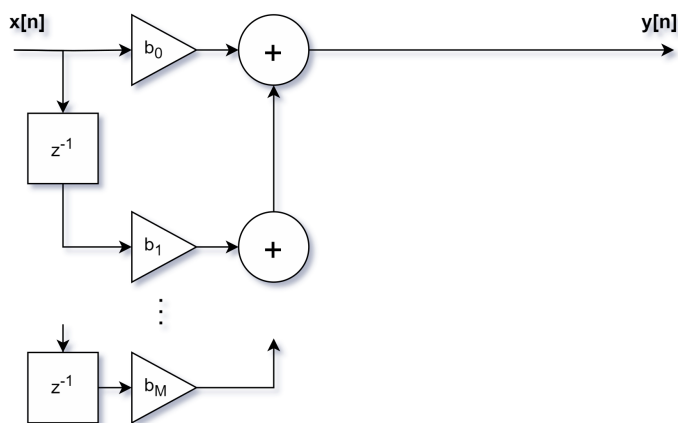
Filtry s konečnou impulsní charakteristikou (FIR) jsou postaveny na využívání pouze nerekurzivní části systému práce se vzorky. Systém tak operuje pouze se vstupními vzorky, jde tak vždy o stabilní typ filtru. Výstupní vzorek je spočítán diferenční rovnicí, která popisuje konečnou diskrétní konvoluci dle vztahu 2.7. Zároveň je shodná s realizací přímého algoritmu, hodnoty impulsní charakteristiky jsou přímo koeficienty filtru [2]

$$y[n] = \sum_{k=0}^{N-1} b_k \cdot x[n - k], \quad (2.7)$$

kde:

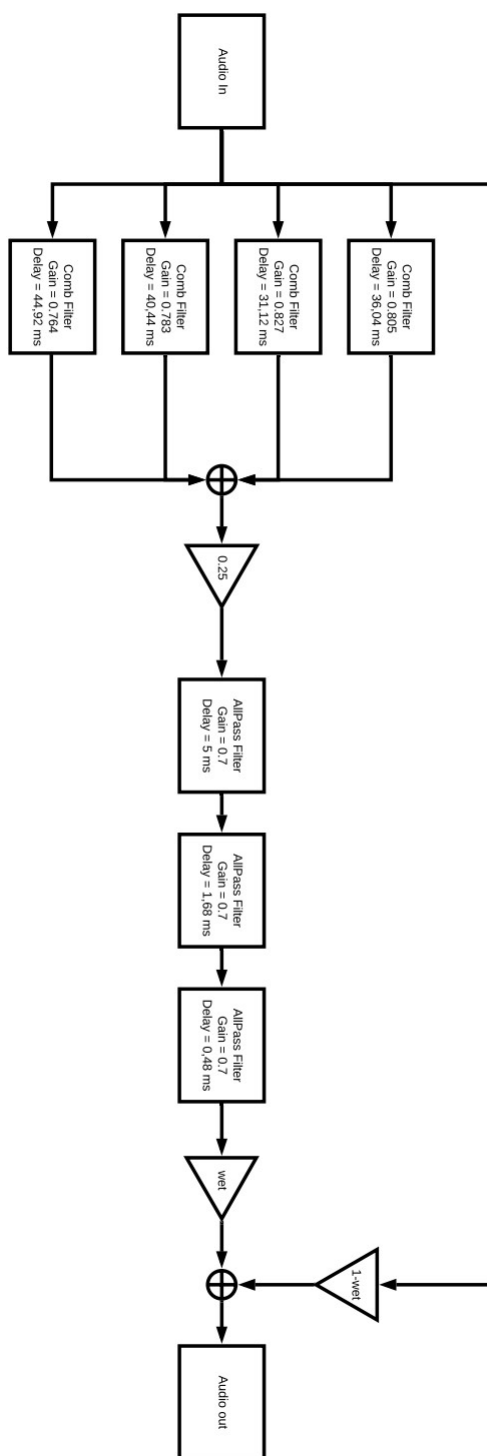
- $y[n]$ je aktuální výstupní vzorek,
- $x[n]$ je aktuální vstupní vzorek,
- b_k je koeficient filtru,
- N je řád filtru (počet zpoždění - 1).

Realizace blokové struktury FIR filtru je vyobrazena na obrázku 2.6, v porovnání se strukturou IIR je na první pohled zřejmé, že je zcela vynechána rekurzivní část filtru (zpětnovazební systém). Funkce jednotlivých bloků zůstává zachována. [5]



Obrázek 2.6: Blokový diagram struktury FIR filtru

Z těchto dvou možností tvorby digitálního filtru byl pro realizaci využit systém IIR filtru. IIR jsou při srovnatelné kvalitě zpracování signálu řádově méně náročné na výpočet, tedy i na ukládání daných vzorků. Zároveň je také možné realizovat návrh vycházející z analogových filtrů, což u FIR filtrů není možné. Tyto vlastnosti jsou vykoupené možnou nestabilitou filtrů, s kterou je nutné pracovat, nebo také nelineární fázovou charakteristikou, s kterou je potřeba počítat při návrhu těchto filtrů.



Obrázek 2.7: Převzaté blokové schéma algoritmu v efektu reverb

Kapitola 3

Softwarové zpracování

3.1 Tvorba softwaru

Vývoj a realizace softwaru se uskutečnila v STM32CubeIDE 3.1.1, právě z důvodu snadné a přehledné konfigurace požadovaných periférií. Návrh číselných filtrů se uskutečnil s pomocí programu MATLAB. V rámci softwaru byl vytvořen ucelený kód pro běh systému zpracování vzorků. Byla vytvořena knihovna pro nastavení kodeku. Ostatní převzaté části kódu jsou vždy označeny a je na ně odkázáno.

3.1.1 STM32CubeIDE

*STM32CubeIDE*¹ je vývojové prostředí určené k vývoji softwaru hlavně pro mikrokontrolery STM32 od společnosti STMicroelectronics. Vývojové prostředí (IDE) je postavené na open source platformě Eclipse. IDE umožňuje generování inicializačního kódu pro konfiguraci vybraných periférií nebo konfiguraci taktovacího systému. Další z velkých výhod je rozvinutý debugging, který umožňuje vyčítat živě data z vybraných registrů nebo sledovat živě žádané proměnné. Konkrétně debugging byl při programování zařízení hojně využíván.

3.1.2 MATLAB

MATLAB² je programovací jazyk a vývojové prostředí určené pro technické výpočty, modelování, simulace a analýzu dat. Vytvořený společností MathWorks a široce využíván v akademické i průmyslové sféře. MATLAB umožňuje rychlý vývoj algoritmů, numerické výpočty a vizualizaci dat. Klíčovou součástí MATLABu je integrované vývojové prostředí, které nabízí nástroje pro psaní kódu, ladění a analýzu výsledků.

¹<https://www.st.com/en/development-tools/stm32cubeide.html>

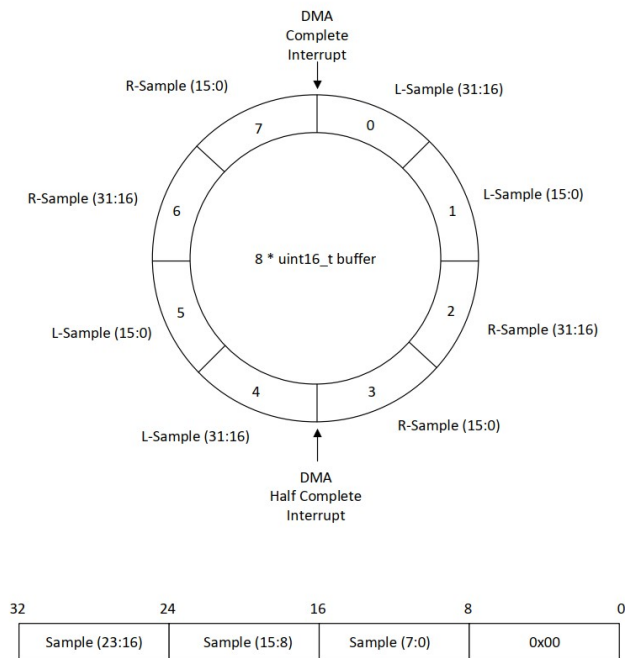
²<https://www.mathworks.com/products/matlab.html>

3.2 Systém zpracování vzorků

Implementace a návrh systému pro zpracování vzorků akustického signálu byl převzat z [10]. Převzatý systém využívá DMA kontroler ve spojení s přerušáním generovaným odesláním a přijmutím vzorků. Využití DMA (přímého přístupu k paměti) umožňuje snížit zátěž na CPU, nastavením DMA kontroleru není již dále potřeba využívat CPU pro paměťovou správu příchozích či odchozích vzorků. V rámci přijímání a odesílání vzorků akustického signálu jsou používány funkce z knihovny HAL. Konkrétně callbacky:

- I2SEx_TxRxDMAHalfCplt a
- I2SEx_TxRxDMACplt.

Díky dvojici callbackových funkcí generovaných přerušáním je možné předat DMA kontroleru polovinu vzorků k odeslání a ve stejnou chvíli zpracovávat druhou polovinu právě přijatých vzorků. Převzaté blokové schéma popisující plnění úložiště vzorků je na obrázku 3.1 Využití přerušování přispívá k plynulému běhu programu, je tak nutné, aby se CPU věnoval práci se vzorky pouze ve chvíli, kdy je přerušování vyvoláno, po jeho obslužení se může vrátit ke klasickému běhu programu. [7]



Obrázek 3.1: Převzaté blokové schéma systému pro zpracování vzorků akustického signálu z [10]

3.3 Ekvalizace vybraných frekvenčních pásem

Ekvalizace vybraných frekvenčních pásem využívá pro každé frekvenční pásmo jeden číslicový filtr. Každé frekvenční pásmo lze tak libovolně nastavit dle potřeby uživatele. Návrh číslicových filtrů využívá systému IIR, který byl vytvořen dle jednotlivých postupů z knihy [3]. Veškeré výpočty, vizualizace dat, kontrola stability systému byly provedeny v softwaru MATLAB 3.1.2. Postup návrhu těchto filtrů se může velmi lišit a jsou různé cesty jak vytvořit požadovaný číslicový filtr [2]. Kromě zesílení daného pásma je možné v průběhu přenosu měnit i jiné parametry než samotno zesílení. Například Q (jakost filtru), nebo střední frekvenci f_c . Kroky k vytvoření požadovaných peak/notch filtrů jsou následující:

1. Výběr analogového ekvivalentů, který splňuje požadované parametry.
2. Volitelným krokem jsou úpravy analogového ekvivalentu.
3. Bilineární transformace ze spojité (analogové) domény $H(s)$ do diskrétní (digitální) domény $H(z)$.
4. Stanovení pomocné proměnné K , tento krok obsahuje převod středního kmitočtu.
5. Stanovení koeficientů podle požadavků na filtr a parametry streamu.

3.3.1 Stanovení analogového ekvivalentu a jeho úpravy

Jako analogový ekvivalent je vybrána pásmová propust druhého řádu, kterou lze ve spojité doméně $H_{bp}(s)$ stanovit jako 3.1 [3]

$$H_{bp}(s) = \frac{\frac{H_0}{Q}s}{s^2 + \frac{1}{Q}s + 1}, \quad (3.1)$$

kde:

- $Q \Rightarrow$ je jakost filtru popřípadě šířka pásma definovaná jako $Q = f_c/f_b$, kde f_c je střední kmitočet pásmové propusti a f_b je frekvenční šířka pásma,
- $H_0 \Rightarrow$ je zesílení vybraného frekvenčního pásma, lze jej interpretovat jako $V_0 = 1 + G_0$, následně pro hladinové vyjádření jako $V_0 = 10^{G/20}$ kde G hladinová úroveň.

Pásmová propust druhého řádu je dále upravena následovně 3.2. Pro $V_0 = 1$ je tak získán přenos roven jedné. To je velmi důležité, jelikož ne vždy je potřeba pásmový filtr použít a díky tomuto je možné zachovat nezměněný amplitudový přenos

$$H(s) = 1 + H_{bp}(s) = \frac{s^2 + \frac{1+H_0}{Q}s + 1}{s^2 + \frac{1}{Q}s + 1} = \frac{s^2 + \frac{V_0}{Q}s + 1}{s^2 + \frac{1}{Q}s + 1}. \quad (3.2)$$

3.3.2 Bilineární transformace

O převod ze spojitě domény H_s do diskrétní domény H_z se v tomto postupu stará Bilineární (Tustinova) transformace za pomoci vztahu 3.3

$$s = \frac{2}{T} \frac{z - 1}{z + 1}. \quad (3.3)$$

Dosazením za proměnnou s do rovnice 3.2. S vhodnými matematickými úpravami, zjednodušeními a normalizací koeficientů získáváme 3.4

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}, \quad (3.4)$$

kde koeficienty a_0, a_1, a_2, b_1, b_2 jsou stanoveny následujícími rovnicemi

$$\begin{aligned} a_0 &= \frac{1 + \frac{V_0}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}, & a_1 &= \frac{2(K^2 - 1)}{1 + \frac{1}{Q}K + K^2}, & a_2 &= \frac{1 - \frac{V_0}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}, \\ b_1 &= \frac{2(K^2 - 1)}{1 + \frac{1}{Q}K + K^2} & a & & b_2 &= \frac{1 - \frac{1}{Q}K + K^2}{1 + \frac{1}{Q}K + K^2}. \end{aligned}$$

Pomocná proměnná K je v předchozích rovnicích stanovena jako $K = \tan(\omega_c T/2)$. Kde T je perioda vzorkovací frekvence a ω_c je střední úhlová frekvence, která byla získána ze střední frekvence za pomoci vztahu $\omega_c = 2\pi f_c$. K zároveň zařizuje korektní převedení f_c ze spojitě do diskrétní domény. Tuto korekci lze v angličtině najít jako prewarping. Koeficienty a_0 a a_2 obsahují proměnnou pro korigování zesílení V_0 , právě tyto koeficienty nejsou fixní a je nutné je s požadovaným zesílením měnit. Tyto koeficienty jsou v následujících sekcích stanoveny právě pro vybrané zesílení. Ostatní zůstávají konstantní. Kdyby bylo rozhodnuto měnit ještě jiné parametry jako například Q nebo f_c bylo by nutné přepočítávat i ostatní koeficienty. To však v této aplikaci není vyžadováno. Maximální doporučené zesílení či potlačení dle návrhu filtrů [3] je 20 dB.

3.3.3 Vyčíslení koeficientů pro požadované filtry

V této sekci jsou použity výsledky z 3.3.2 s danými požadavky na peak filtr frekvenčního pásma. Vzorkovací perioda T je do rovnic 3.3.2 dosazena za pomoci vzorkovací frekvence $T = 1/f_s$.

1. Spodní frekvenční pásmo s $f_c = 100$ Hz, $Q = 1,25$, $V_0 = 10$ a $f_s = 48000$ Hz. Dosazením do rovnic z 3.3.2 jsou získány koeficienty:

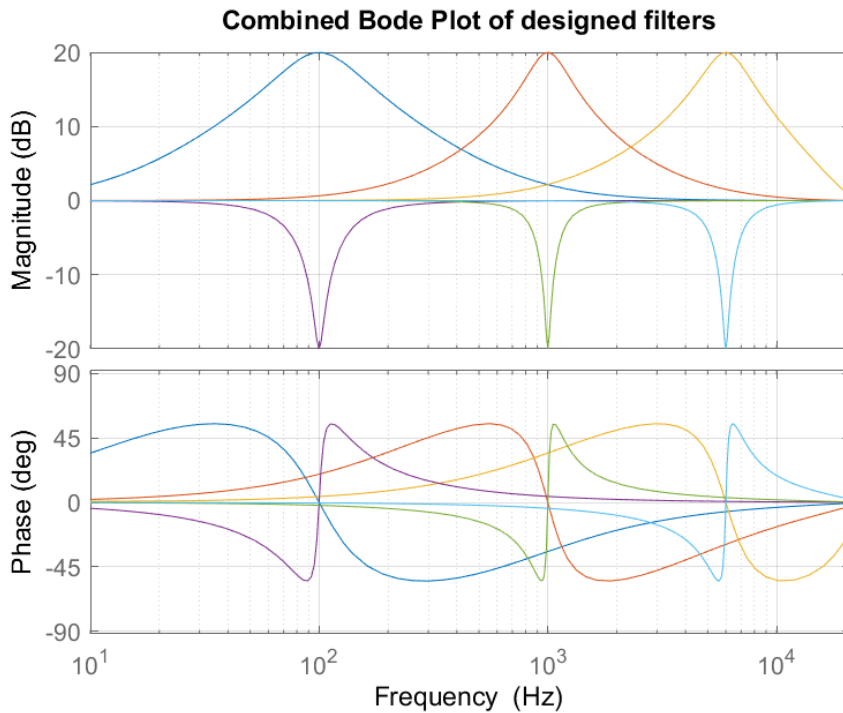
$$\begin{aligned} a_0 &= 1.046877, & a_1 &= -1.989412, & a_2 &= 0.942706, \\ b_1 &= -1.989412, & b_2 &= 0.989583, & K &= 0.006545 \quad a \quad Q = 1.250000. \end{aligned}$$

2. Střední frekvenční pásmo s $f_c = 1000$ Hz, $Q = 2,5$, $V_0 = 10$ a $f_s = 48000$ Hz. Dosazením do rovnic z 3.3.2 jsou získány koeficienty:

$$\begin{aligned} a_0 &= 1.228970, & a_1 &= -1.932443, & a_2 &= 0.720148, \\ b_1 &= -1.932443, & b_2 &= 0.949118, & K &= 0.065543 \quad a \quad Q = 2.500000 \end{aligned}$$

3. Vyšší frekvenční pásmo s parametry $f_c = 6000$ Hz, $Q = 2$, $V_0 = 10$ a $f_s = 48000$ Hz. Dosazením do rovnic z 3.3.2 jsou získány koeficienty:

$$\begin{aligned} a_0 &= 2.351990, & a_1 &= -1.201769, & a_2 &= -0.652432, \\ b_1 &= -1.201769, & b_2 &= 0.699558, & K &= 0.414214 \quad a \quad Q = 2.000000 \end{aligned}$$



Obrázek 3.2: Navržené peaking filtry situovány do jednoho grafu s maximální zesílením i potlačením vybraných pásem

Za pomoci funkce `bode` jsou získány z MATLABU grafy přenosových funkcí [12]. Tyto grafy byly zkombinovány do jednoho grafu i včetně přepočítaných

průběhu pro $V_0 = 0.1$ tedy maximální útlum v daných pásmech. Grafy na obrázku 3.2, tak přímo ukazují teoretické možnosti softwarové implementace. V horním grafu na obrázku 3.2 můžeme sledovat amplitudové spektrum a v dolním grafu spektrum fázové.

3.4 Implementace algoritmů pro úpravu vzorků

V této části bude provedena implementace úpravy vzorků diskrétního signálu k získání požadovaných efektů.

3.4.1 Stereo

Stereo tedy přenos dvou kanálů bez dalších úprav byl v rámci systému zpracování vzorků též převzat z github repozitáře [10]. Na tomto systému jsou následně postaveny další efekty 3.1.

```
case 0: // Stereo mode

    /* >>> Store samples for further processing <<< */
    lSample = (int32_t) (rxBuf[0]<<16)|rxBuf[1];
    rSample = (int32_t) (rxBuf[2]<<16)|rxBuf[3];

    /* >>> Here insert required processing <<< */

    /* >>> Return processed samples <<< */
    txBuf[0] = (lSample>>16)&0xFFFF;
    txBuf[1] = lSample&0xFFFF;
    txBuf[2] = (rSample>>16)&0xFFFF;
    txBuf[3] = rSample&0xFFFF;

    break;
```

Ukázka kódu 3.1: Stereo

3.4.2 Mono

Jeden z prvních implementovaných efektů byl právě mono efekt, tedy sjednocení obou kanálů do jednoho. Využití zasluhuje hlavně při použití vstupu elektrické kytary, kdy chceme naklonovat akustický signál do obou kanálů aby nám nehrálo právě jen jedno sluchátko. Implementace tohoto efektu je v ukázce kódu 3.2.

```
/* >>> Divide by 2 to reach same energy level <<< */
lSample >>= 1;
rSample >>= 1;
```

```
/* >>> Add together <<< */
lSample = rSample + lSample;
rSample = lSample;
```

Ukázka kódu 3.2: Mono efekt

3.4.3 Úprava hlasitosti

Další důležitá úprava vzorků, je ovládání úrovně hlasitosti. Systém úpravy hlasitosti získává vstupní data z uživatelského rozhraní konkrétně za pomoci rotačního enkóderu, tyto data vhodně zařadí do mezí hlasitosti konkrétně do hodnot $\langle 0, 1 \rangle$ kde 0 znamená absolutní ztišení a 1 neupravuje hlasitost jakkoli.

```
/* >>> Load current value from user interface <<< */
volumef = countf5;

/* >>> Volume adjustment <<< */
lSample = lSample*volumef;
rSample = rSample*volumef;
```

Ukázka kódu 3.3: Úprava hlasitosti

3.4.4 Reverb

Efekt reverb byl převzat včetně implementace z [10]. Následně vhodně zakomponován do kódu, pro získání stejného efektu byly sníženy jednotlivé uložště pro dočasné ukládání vzorků. Konkrétně nové uložště byly sníženy na polovinu původních. Autor kódu totiž používal $f_s = 96$ kHz a v navrhovaném systému je využívána $f_s = 48$ kHz. Shodně jako u úpravy zvuku je možné efekt regulovat uživatelem.

```
/* >>> Load current value from user interface <<< */
reverbf = countf5;

/* >>> Reverb effect implementation <<< */
float sum = (float) (lSample + rSample);
sum = (1.0f-reverbf)*sum + reverbf*Do_Reverb(sum);
```

Ukázka kódu 3.4: Reverb efekt

3.4.5 Filtrování vybraných pásem

Systém IIR filtrace v C jazyce byl postaven podle podobného systému z knihy [5]. Ve vytvořené knihovně efektů v přílohách můžeme najít funkci aktualizující koeficienty 3.6. Podle vstupů z uživatelského rozhraní je možné měnit zesílení od -20 dB do 20 dB v daném pásmu.

```
float IIRL_update_coef(float g)
{
    /* Update existing coefficient */
    IIRL_a0 = (1.0f + (g / QL) * KL + KL * KL) / (1.0f + (1.0
    f / QL) * KL + KL * KL);

    return IIRL_a0;
}
```

Ukázka kódu 3.5: Aktualizace koeficientu pro filtr spodního frekvence pásmu

Funkce `IIRL_update_coef` je volána hlavní funkcí `IIRL_left_filter` pro filtraci levého kanálu, koeficienty filtru jsou tak přepočítávány každý cyklus vyzvednutí a odeslání vzorků.

```
float IIRL_left_filter(float inSample)
{
    IIRL_update_coef(g);
    IIRL_update_coef1(g);

    /* Shift input samples */
    IIRL_left_x2 = IIRL_left_x1;
    IIRL_left_x1 = IIRL_left_x0;
    IIRL_left_x0 = inSample;

    /* Shift output samples */
    IIRL_left_y2 = IIRL_left_y1;
    IIRL_left_y1 = IIRL_left_y0;

    /* Compute new sample */
    IIRL_left_y0 = (IIRL_a0 * IIRL_left_x0 + IIRL_a1 *
    IIRL_left_x1 + IIRL_a2 * IIRL_left_x2)
    - (IIRL_b1 * IIRL_left_y1 + IIRL_b2 * IIRL_left_y2);

    return IIRL_left_y0;
}
```

Ukázka kódu 3.6: Filtrace vzorků levého kanálu ve spodním frekvence pásmu

V poslední řadě této úpravy diskretních vzorků, jsou vzorky vráceny do callbackových funkcí `I2SEx_TxRxDMAHalfCplt`, `I2SEx_TxRxDMACplt` a zpracová-

vány k odeslání DMA kontrolerem.

```
/* >>> Filter desired frequencies <<< */
IIRL_right_filter(rSample);
IIRL_left_filter(lSample);

/* Update sample with filter output value */
lSample = IIRL_left_y0;
rSample = IIRL_right_y0;
```

Ukázka kódu 3.7: Filtrování spodního frekvnenčního pásma

3.5 Uživatelské rozhraní

Výstup uživatelského rozhraní využívá jednobarevného OLED displeje s I2C sběrnici. Pro komunikaci s tímto displejem byla použita knihovna [13]. Displej je při zapnutí zařízení nejdříve inicializován a další komunikace je postavena čistě na přerušeních vyvolaných uživatelem. Je tedy nutné komunikovat s displejem jen při obslužení přerušení vycházejících z uživatelovi interakce. Ukázka typicky zobrazovaných dat displejem 3.3.



Obrázek 3.3: Fotografie zobrazovaných dat zařízením

Pro výběr a úpravu efektů jsou v uživatelském rozhraní využívány dva rotační enkóдеры. Rotační enkóдеры jsou připojeny k interním časovačům mikrokontroleru. Časovače pracují v kombinovaném režimu Encoder mode, což je mód připravený výrobcem pro připojení právě těchto zařízení a není tak nutné implementovat vlastní program pro jejich obsluhu.

Kapitola 4

Hardwarové zpracování

4.1 Tvorba vývojové desky

Pro potřebu testování přenosu a implementaci softwaru byla vytvořena vývojová deska. Návrh plošného spoje se uskutečnil v programu Altium Designer. Pro výrobu DPS byl zvolen výrobce *JLPCB*¹. Osazení hotové desky proběhlo ručně s použitím kombinované pajecí a horkovzdušné stanice.

4.1.1 Altium Designer

*AltiumDesigner*² je komplexní software pro návrh DPS. Spolu s vývojovými prostředími OrCad a Siemens EDA (dříve Mentor Graphics) tvoří úplnou špičku v oblasti návrhu analogových a digitálních obvodů. Altium Designer je placený software. Pro studenty technických škol je zde možnost zažádat o bezplatnou licenci po dobu studia na vysoké škole či univerzitě. Altium Designer též nabízí širokou škálu výukových videí, nejrozličnější školení v oblasti elektroniky a také kurzy práce v ECAD s možností získání následné certifikace.

4.1.2 Návrh DPS

V rámci návrhu DPS bylo vycházeno hlavně z kurzů od Altium Designer pro práci s daným softwarem. Kurzy jsou konkrétně poskytovány *AltiumEducation*³ v rámci absolvování těchto kurzů byl získán certifikát PCB Basic design course. Knihovny některých součástek byly převzaty z GitHub repozitáře [14], což je repozitář jednoho z tvůrců dalších kurzů pro Altium Designer. Většina ostatních součástek byla importována do Altium Designer za pomoci *SamacSys*⁴.

4.1.3 Klíčové komponenty DPS

Srdcem celé desky je již zmíněný MCU STM32F411 [15], taktovací frekvence f_t MCU byla navržena jako $f_t = 12,288$ MHz, a je zajištěna externím krys-

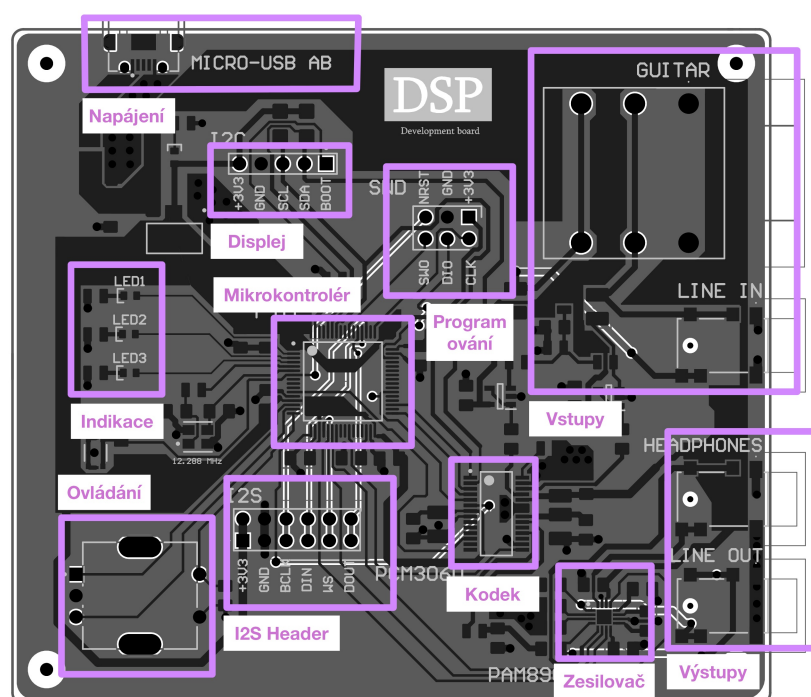
¹<https://jlpcb.com/>

²<https://www.altium.com/altium-designer>

³<https://education.altium.com/courses/>

⁴<https://www.samacsys.com/>

talem s požadovanou f_t [16]. Tato taktovací frekvence umožňuje nastavení standardních kmitočtů vzorkovací frekvence (48 kHz, 96 kHz). Jako A/D, D/A převodník neboli kodek byl zvolen IC PCM3060 [17] pro svoje méně náročné nastavení. Kodek totiž obsahuje celkovem 9 konfiguračních registrů a je zároveň možné jej nastavit čistě hardwarově, pro variabilitu je v této aplikaci však využíváno softwarové nastavení sběrnicí I2C. Napájení systému je řešeno pomocí micro USB type AB konektoru, ze vstupních +5V je za pomoci LDO regulátoru získáno +3.3V pro napájení mikrokontroléru a digitální části kodeku. Jako sluchátkový zesilovač byl použit IC PAM8908 [18], tento zesilovač pracuje ve třídě AB a jeho výkon je 25 mW na jeden kanál při zátěži $16\ \Omega$. Rozložení všech komponentů lze sledovat na obrázku 4.1.



Obrázek 4.1: Rozložení komponentů vývojové DPS

4.1.4 Uživatelské rozhraní

Uživatelské rozhraní je tvořeno rotačním enkóderem pro úpravu hlasitosti, nastavení parametrů filtru. Zobrazení vybraného filtru/efektu a dalších informačních údajů se stará OLED displej.

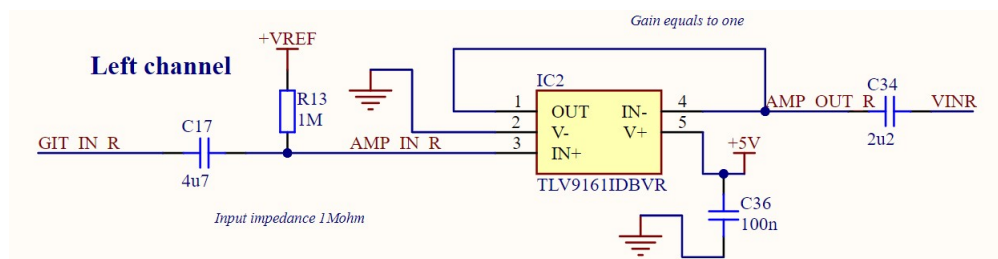
4.1.5 Vstupy a výstupy

Vstupní obvody zařízení umožňují připojení 3.5mm jacku jako linkového vstupu, dále nabízí možnost přímého připojení elektrické kytary pomocí 6.3 mm jacku. Jako vstupní obvod signálu z elektrické kytary byl využit



Obrázek 4.2: OLED display převzato z www.drateg.cz

operační zesilovač v zapojení → napěťový sledovač viz obrázek 4.3 vybraný ze schématu ???. Zapojení využívá vysokého vstupního odporu, následně je signál jednotkově zesílen a přiveden na výstup na který je nyní možné připojit obvod s nižším vstupním odporem. Zapojení na obrázku 4.3 používá přičtení stejnosměrné složky signálu o hodnotě poloviny napájecího napětí OZ, to umožňuje použití nesymetrického napájení. V daném zapojení je pak nutné použít oddělovací kondenzátory C17, C34 a limitovat stejnosměrnou složku pouze uvnitř zapojení. Výstupní signál ze zapojení 4.3 je přiveden na vstup levého kanálu kodeku. Shodně je zpracován i pravý kanál ikdyž by to pro připojení pouze elektrické kytary nebylo nutné. Analogový výstup z kodeku je přiveden na vstup sluchátkového zesilovače PAM8908 a zároveň vyveden jako linkový výstup signálu na 3.5mm jack. Výstup zesilovače je poté přiveden na 3.5mm jack pro připojení sluchátek.



Obrázek 4.3: Zapojení OZ jako napěťového sledovače pro levý kanál

4.2 Modifikace pro finální produkt

V porovnání s vývojovou deskou bylo při novém návrhu přistoupeno k některým modifikacím, vylepšením původního návrhu. Finální produkt dostal projektový název Antinode (kmitna).

4.2.1 Mikrokontroler

Největší změnou pro kterou bylo rozhodnuto, je výkonnější mikrokontroler. Jako vylepšení původního MCU byl vybrán jeden z nejvýkonnější řady F4 od firmy STMicroelectronics. Tedy konkrétně model STM32F469. Hlavním

důvodem pro toto rozšíření byla velikost paměti RAM, která je z velké části využita k ukládání vzorků pro následné další zpracování. Při testování náročnějších efektů docházelo k vyplnění podstatné části paměti. [19, 20]

Rozšíření na výkonnější MCU tak dovoluje naddimenzovat systém a popřípadě vytvořit prostor na složitější efekty.

Dalším rozdílem je zvýšení taktovací frekvence, to umožňuje vykonat více instrukcí za stejný čas a zpracovat tak více dat. Přehledný rozdíl mezi původním a novým MCU je možné sledovat v tabulce 4.1.

Mikrokontroler	STM32F411	STM32F469
Procesor	ARM Cortex-M4	ARM Cortex-M4
Frekvence CPU	až 100 MHz	až 180 MHz
Flash paměť	až 512 kB	až 2 MB
RAM	128 kB	384 kB
GPIO piny	51	71
I2C	až 3	až 3
I2S	až 5	až 2
FPU	ano	ano
DMA	ano	ano
Napájení	2.0 až 3.6V	2.0 až 3.6V

Tabulka 4.1: Porovnání mikrokontroleru vývojové desky a finálního zařízení

4.2.2 Napájení

Dalším rozdílem bylo použití USB-C konektoru místo původního Micro USB-AB, což umožňuje širší využití tohoto konektoru. Návrh byl proveden dle doporučení STMicroelectronics [21]. Zařízení je schopné bez další komunikace požádat až o 3A při 5V.

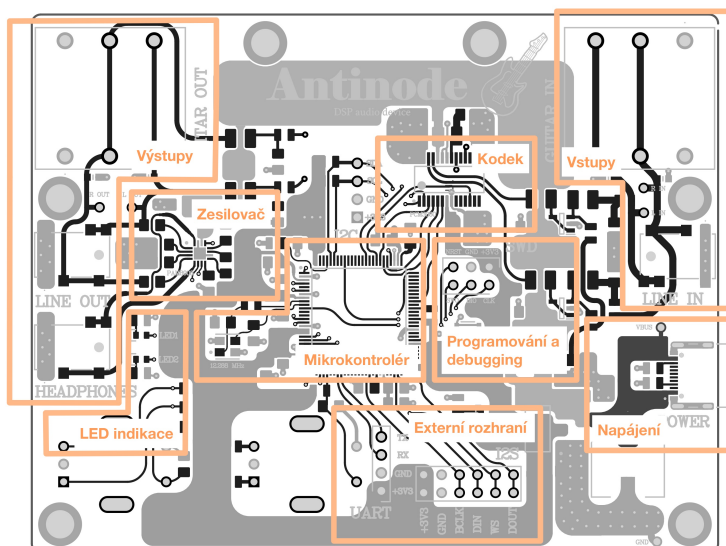
4.2.3 Další úpravy

Dále byl přidán konektor pro UART komunikaci, nebo také ESD ochrana rizikových konektorů, jako jsou připojení napájení, vstupy a výstupy zvukového signálu. Návrh ochrany obvodů vycházel z doporučení výrobce MCU [22]. Další úpravy zahrnují přidání druhého rotačního enkodéru pro ovládání zařízení a přidání 6,3 mm jacku jako výstupního konektoru pro linkový signál.

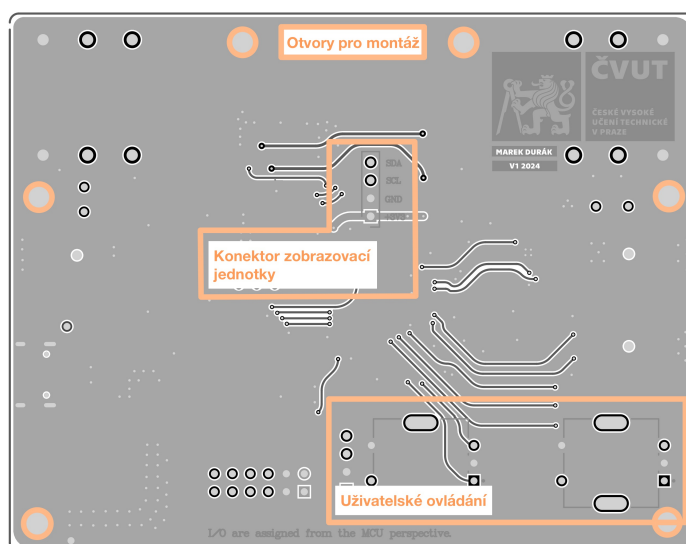
4.3 Realizace finálního produktu

K navržení DPS bylo přistoupeno tak, aby zařízení následně mohlo být zasazené do 3D vtištěného pouzdra. K navržení DPS byl použit opět software

Altium Designer. K výrobě DPS byl zvolen pro změnu výrobce *PCBWAY*⁵. Následné osazení nebo 3D tisk pouzdra se uskutečnil v domácích podmínkách. Ostatní komponenty zůstaly shodné s vývojovou deskou a jejich rozložení je možné sledovat z pohledu shora na obrázku 4.4 a ze spodní strany na obrázku 4.5.



Obrázek 4.4: Vrchní strana DPS Antinode



Obrázek 4.5: Spodní strana DPS Antinode

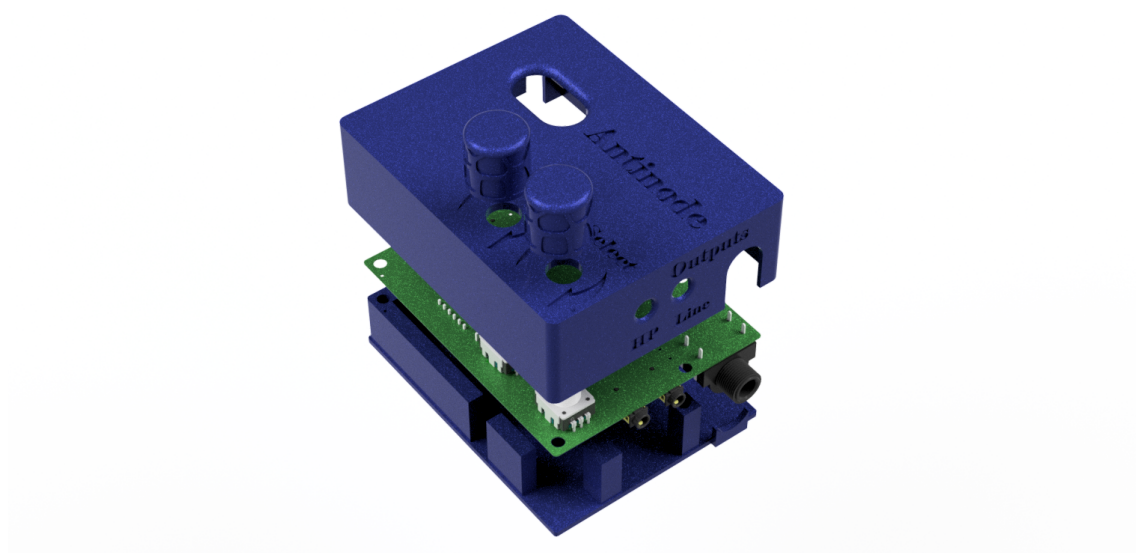
⁵<https://www.pcbway.com/>

■ 4.3.1 Modelování

Pro modelování 3D modelu pouzdra (krabičky) zařízení byl využit program Fusion 360. Krabička se skládá ze dvou částí spojenými šrouby M3 se schovanými hlavami. Další díly k modelování jsou plastové kryty rotačních enkodérů. Pro snazší modelování, nastavování mezer a určování přesných vzdáleností byl vyexportován z Altium Designer .step soubor desky plošných spojů. Dle něj byl následně vytvořen a vygenerován 3D model 4.6.

■ 4.3.2 Fusion 360

*Fusion 360*⁶ je cloudový software pro modelování objektu. Zajišťuje návrh, realizaci a samotnou simulaci objektů (CAD a CAE). Následně také export do výroby, či samotné řízení nástrojů (CAM). Fusion 360 je k dispozici pro studenty většiny technických škol zdarma po dobu studia. Je dokonce možné v něm navrhovat elektronické obvody skrz vcelku nově importované rozšíření EAGLE.



Obrázek 4.6: Vygenerovaný finální vzhled 3D modelu krabičky Antinode

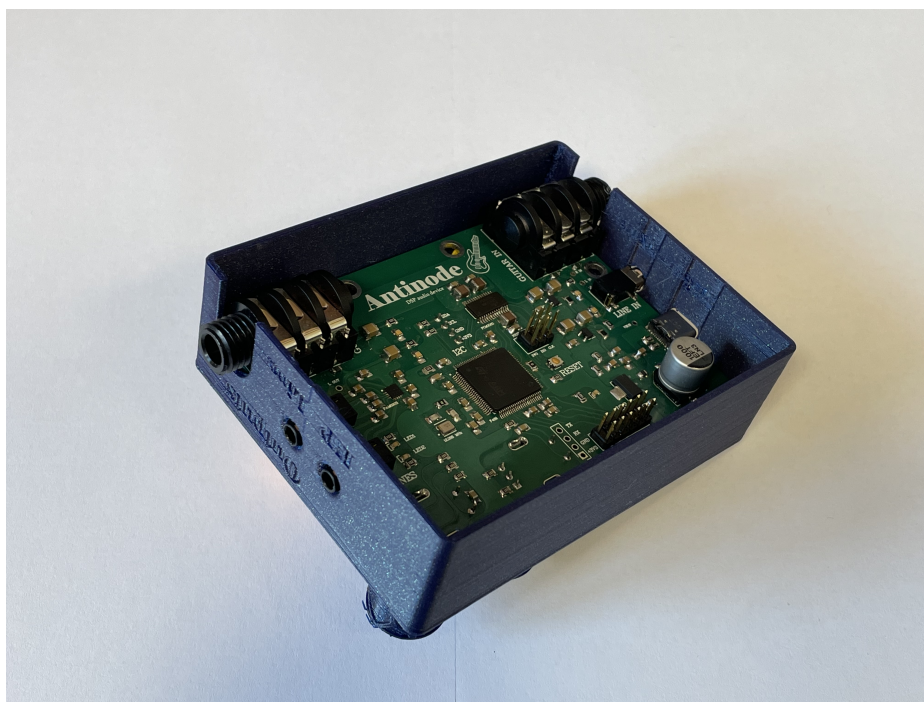
■ 4.3.3 3D tisk pouzdra

Příprava 3D tisku pouzdra zahrnuje vyexportování hotového modelu z Fusion 360 do .stl souboru, tento soubor je otevřen v programu UltiMaker Cura. V programu UltiMaker Cura jsou navoleny parametry pro požadovanou kvalitu 3D tisku a s již otestovanými konfiguracemi tiskárny jsou jednotlivé díly vytištěny 4.7.

⁶<https://www.fusion360.cz/>

4.3.4 UltiMaker Cura

*UltiMaker Cura*⁷ je open source software k vytvoření G-code ze vstupního 3D modelu pro 3D tiskárnu. Cura nabízí širokou škálu modifikací parametrů pro 3D tisk, včetně možnosti doinstalování vnitřních rozšíření. Jde o uživatelsky velmi přehledné prostředí. I přesto, že Cura spadá pod UltiMaker je možné ji používat pro širokou škálu tiskáren ostatních značek. Profily pro tiskárnu je možné importovat odzkoušené od jiných uživatelů.



Obrázek 4.7: Fotografie celého zařízení ze spodní strany bez spodního krytu

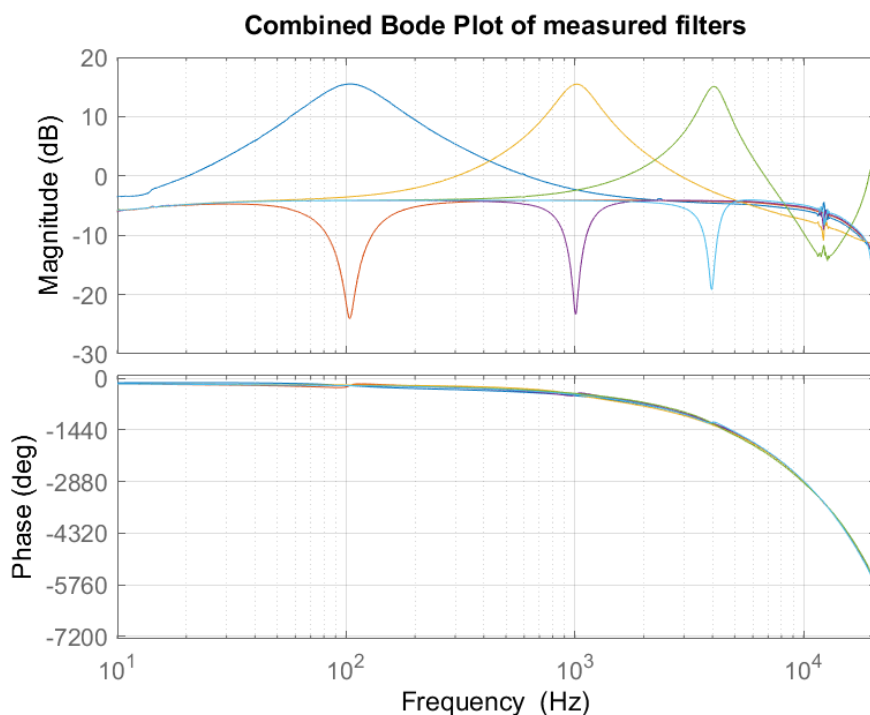
⁷<https://ultimaker.com/software/ultimaker-cura/>

Kapitola 5

Diskuze

5.1 Porovnání teoretických hodnot filtrů s naměřenými hodnoty

Zde jsou porovnány přenosové funkce z části 3.3.3 spolu s naměřenými přenosovými funkcemi. K měření přenosových funkcí byl použit dynamický signální analyzátor Photon+ od firmy Brüel & Kjær. Změřená data byla následně zpracována v MATLABU a vykreslena do přenosových funkcí shodně jako v sekci 3.3.3. Kombinovaný graf přenosových funkcí pro všechna pásma, měřena postupně s maximálním zesílením a maximálním útlumem je na obrázku 5.1.



Obrázek 5.1: Postupně změřené peaking filtry situovány do jednoho grafu

Při porovnání s návrhem peaking filtrů 3.2. Je možné konstatovat přesnost středových frekvencí f_c , mírné odlišení požadovaného útlumu či zesílení. Tato odlišnost může být způsobena nepřesným odečtením uživatelem stanovené hodnoty zesílení, tedy místo chtěného zesílení $V_0 = 10$ je krokováním rotačního enkóderu získaná hodnota například $V_0 = 9.87$. Největší rozdíl však postihuje fázové spektrum, v naměřeném fázovém spektru získáváme s rostoucí frekvencí značně stoupající hodnoty fáze. To značí systémem generované zpoždění na vyšších frekvencích.

■ 5.2 Známe nedostatky zařízení

V tuto chvíli není možné skládat jednotlivé efekty za sebe, je třeba tak vybrat právě jeden efekt popřípadě filtr a ten následně upravovat. Další z nedostatků souvisí s odesíláním dat do displeje při změně parametrů efektu/filtru, kdy je přerušena I2S komunikace a CPU se věnuje odesílání dat displeji. Tento nedostatek je možné vyřešit použitím DMA kontroleru pro odesílání dat displeji.

■ 5.3 Další možné úpravy zařízení

Jedno z možných rozšíření do budoucna, je vytvoření softwarového menu, pro snazší orientaci, správu efektu a filtrů. Menu může zároveň také umožnit skládání vybraných efektů za sebe. Se skládáním efektů za sebe je následně možné vytvořit grafický ekvalizér s vybraným počtem pásem. Vhodné rozšíření je implementace cyklické fronty pro správu ukládání přichozích vzorků. Toto rozšíření umožňuje optimalizovat program a snížit počet přepisů jednotlivých proměnných [23].



Závěr

V rámci práce bylo vytvořeno zařízení pro zpracování akustického signálu v reálném čase. Na zařízení byl implementován systém zpracování vzorků akustického signálu s velmi nízkou prodlevou. Softwarově bylo dosaženo prodlevy zpoždění dvou vzorků, časově stanovené na $42 \mu\text{s}$. Nejdříve byla vytvořena vývojová deska k testování funkčnosti systému, následně bylo vytvořené finální zařízení, které disponuje všestrannými potřebami pro užití v praxi. V rámci bakalářské práce byl vybrán pro daný systém vhodný mikrokontroler, který splňuje požadavky zamýšlené aplikace. S mikrokontrolerem byly vybrány ostatní periferie, které kompletují celý systém zpracování akustického signálu. Pro navržený systém byly vytvořeny digitální filtry vybraných frekvenčních pásem a instalovány další efekty. Také bylo vytvořeno uživatelské rozhraní pro ovládání zařízení. Celková výsledná cena vývojové desky s výrobou DPS v Číně, byla stanovena jako ≈ 800 Kč. Výsledná cena finálního produktu Antinode se dostává zhruba na dvojnásobek. Přesunutím výroby DPS do Česka by se výsledná cena ještě vzrostla.



Literatura

- [1] FORRÓ, Daniel. *Domáci nahrávací studio*. Praha: Grada, 1996. ISBN 80-7169-231-X.
- [2] JAN, Jiří. *Číslicová filtrace, analýza a restaurace signálů*. Brno: VUT, 1997. ISBN 80-214-0816-2.
- [3] ZÖLZER, Udo. *Digital audio signals processing*. 2nd edition Chichester: John Wiley & Sons, 2008. ISBN 0470997850.
- [4] KARPLUS, Kevin. *Applied Analog Electronics: A First Course in Electronics*. Singapore: World Scientific, 2023. ISBN 9811254419.
- [5] IBRAHIM Dogan a IBRAHIM Ahmet. *Practical Audio DSP Projects with the ESP32*. Elektor Verlag, 2023. ISBN 978-3-89576-567-4.
- [6] Philips Semiconductors. *I2S bus specification* [online]. 1986, [citováno 2024-2-12]. Dostupné z: www.sparkfun.com/I2SBUS.pdf
- [7] HINTZ Kenneth a TABAK Daniel. *Microcontrollers: architecture, implementation and programming*. United States of America: McGraw-Hill, 1992. ISBN 0070289778.
- [8] ARM. *CortexM4 technical reference manual* [online]. [citováno 2024-21-3]. Dostupné z: <https://developer.arm.com/documentation/100166/0001/?lang=en>
- [9] STMicroelectronics. *STM32F411 datasheet* [online]. [citováno 2024-18-2]. Dostupné z: www.st.com/resource/en/datasheet/stm32f411ce.pdf
- [10] YetAnotherElectronicsChannel. *STM32_DSP_Reverb* [online]. [citováno 2024-18-2]. Dostupné z: github.com/YetAnotherElectronicsChannel/STM32_DSP_Reverb
- [11] SMITH, Julius Orion. *"Schroeder Reverberators" in Physical Audio Signal Processing* [online]. [citováno 2024-16-5]. Dostupné z: ccrma.stanford.edu/jos/pasp/Schroeder_Reverberators.html
- [12] MATLAB Documentation. *MathWorks* [online]. [citováno 2024-2-12]. Dostupné z: www.mathworks.com/help/index.htmlf

- [13] ALEKSEEV, Aleksander. *stm32-ssd1306* [online]. [citováno 2024-18-2]. Dostupné z: github.com/YetAnotherElectronicsChannel/STM32_DSP_Reverb
- [14] Salmony, Philip. *AltiumDesigner-Libraries* [online]. [citováno 2024-18-2]. Dostupné z: github.com/pms67?tab=repositories
- [15] STMicroelectronics. *STM32F411 reference manual* [online]. [citováno 2024-18-2]. Dostupné z: www.st.com/rm0383-stm32f411xce-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
- [16] STMicroelectronics. *STM32 application note* [online]. [citováno 2024-18-2]. Dostupné z: www.st.com/an2867-guidelines-for-oscillator-design-on-stm8afals-and-stm32-mcusmpus-stmicroelectronics.pdf
- [17] Texas Instruments. *PCM3060 datasheet* [online]. 2014, [citováno 2024-25-2]. Dostupné z: www.ti.com/lit/ds/symlink/pcm3060.pdf
- [18] Diodes. *PAM8908 datasheet* [online]. 2022, [citováno 2024-25-2]. Dostupné z: www.diodes.com/assets/Datasheets/PAM8908.pdf
- [19] STMicroelectronics. *STM32F469 reference manual* [online]. [citováno 2024-21-3]. Dostupné z: www.st.com/rm0386-stm32f469xx-and-stm32f479xx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
- [20] STMicroelectronics. *STM32F469 datasheet* [online]. [citováno 2024-21-3]. Dostupné z: www.st.com/resource/en/datasheet/stm32f469ae.pdf
- [21] STMicroelectronics. *STM32 technical article* [online]. [citováno 2024-21-3]. Dostupné z: www.st.com/dm00496853-overview-of-usb-type-c-and-power-delivery-technologies-stmicroelectronics.pdf
- [22] STMicroelectronics. *STM32 technical article* [online]. [citováno 2024-21-3]. Dostupné z: www.st.com/an5612-esd-protection-of-stm32-mcus-and-mpus-stmicroelectronics.pdf
- [23] All about circuits. *Circular buffer technical article* [online]. 2022, [citováno 2024-25-2]. Dostupné z: www.allaboutcircuits.com/technical-articles/circular-buffer-a-critical-element-of-digital-signal-processors/



Příloha A

Seznam použitých zkratk

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAE	Computer Aided Engineering
I2S	Inter-IC Sound
PIO	Programmable Input Output
MSB	Most Significant Bit
MCU	Microcontroller Unit
CPU	Central Processing Unit
RAM	Random Acces Memory
DMA	Direct Memory Acces
DPS	Deska Plošných Spojů
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IDE	Integrated Development Enviroment
FPU	Floating Point Unit
LQFP	Low profile Quad Flat Package
I2C	Inter-Integrated Circuit
ARM	Advanced RISC Machine
EEPROM	Electrically Erasable Programmable Read Only Memory
A/D	Analog to Digital converter
D/A	Digital to Analog converter

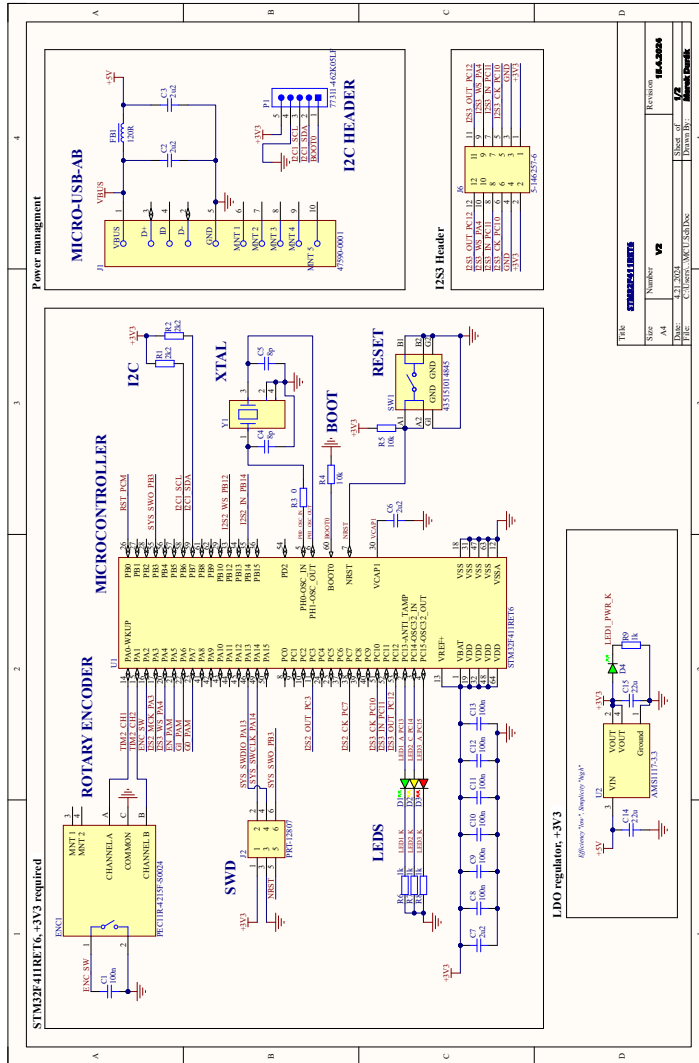
A. Seznam použitých zkratk

OLED	Organic Light Emitting Diode
IC	Integrated Circuit
kodek	kodek-dekodek
LDO	Low Dropout
ECAD	Electronic Computer Aided Design
OZ	Operační zesilovač
ESD	Electrostatic Discharge

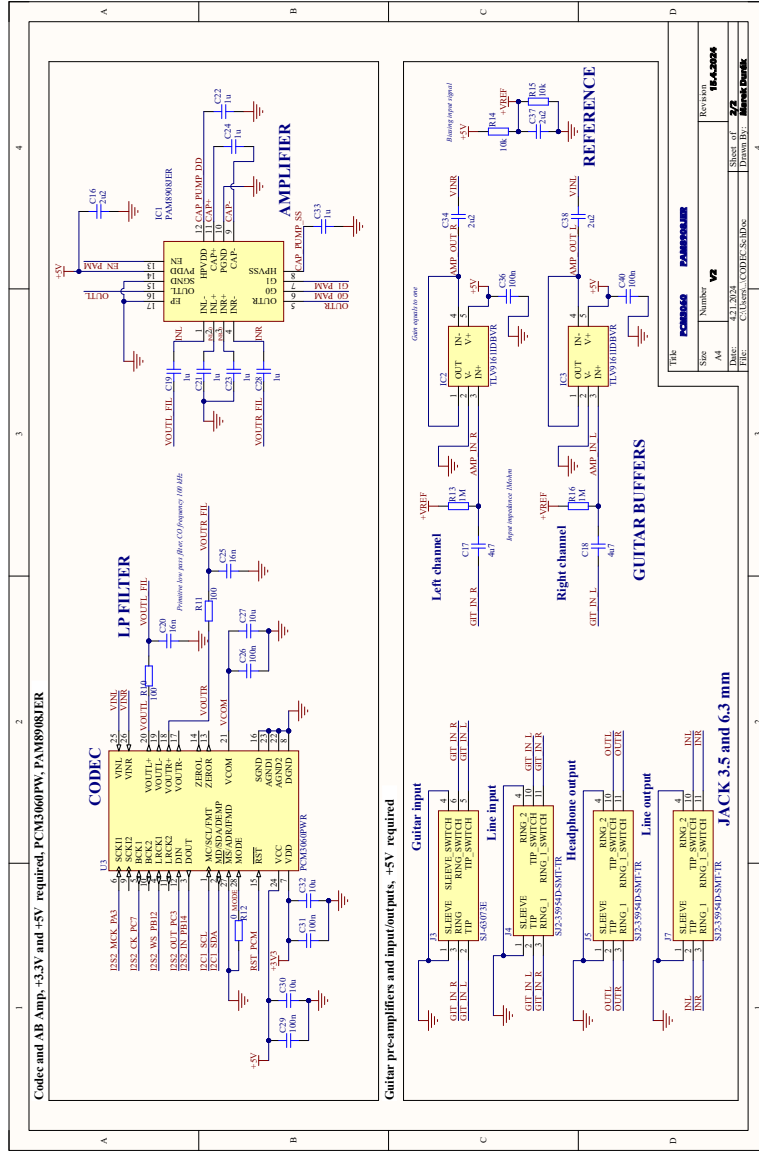
Příloha B

Schéma vývojové DPS

B.1 Část první



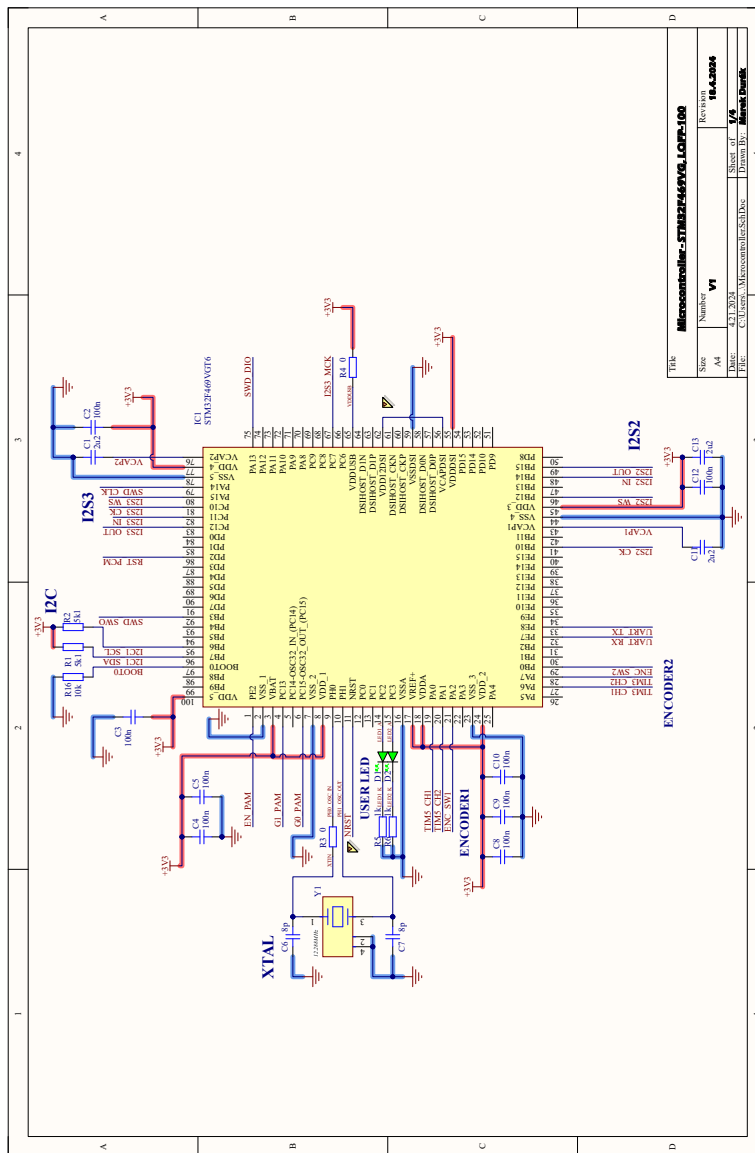
B.2 Část druhá



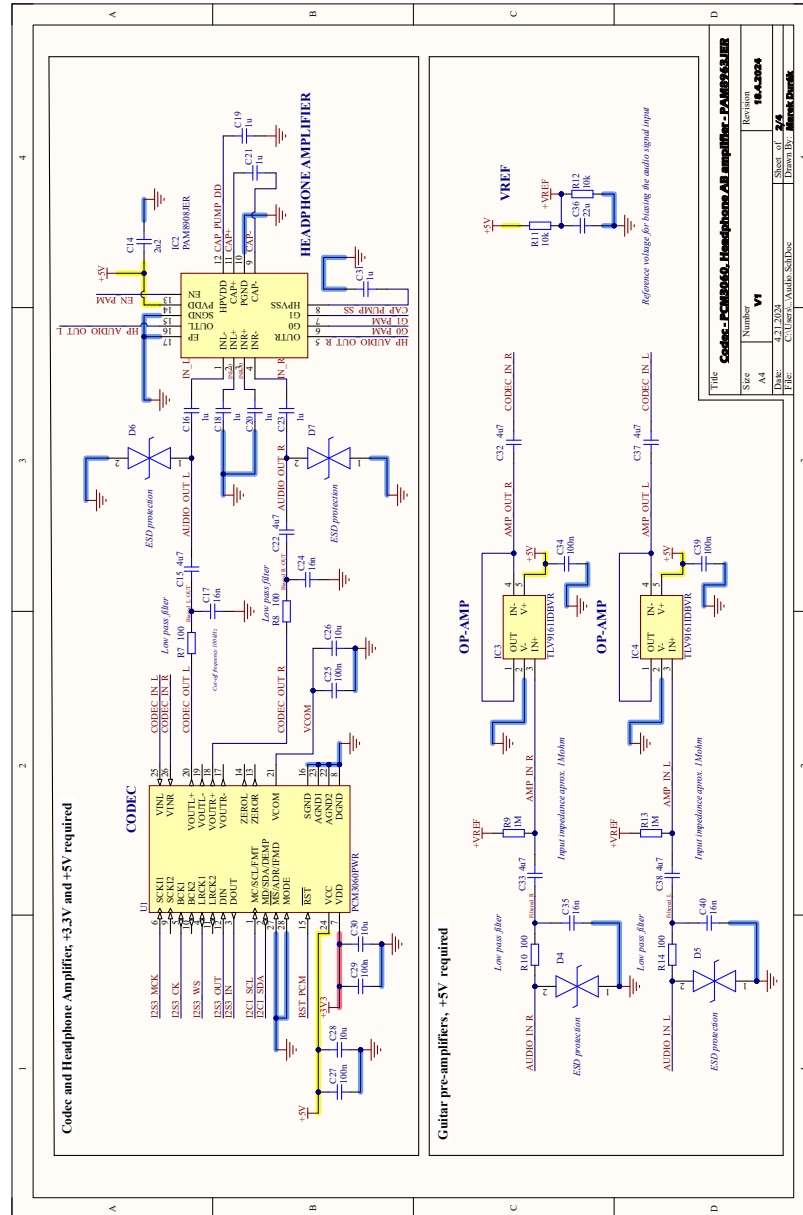
Příloha C

Schéma Antinode DPS

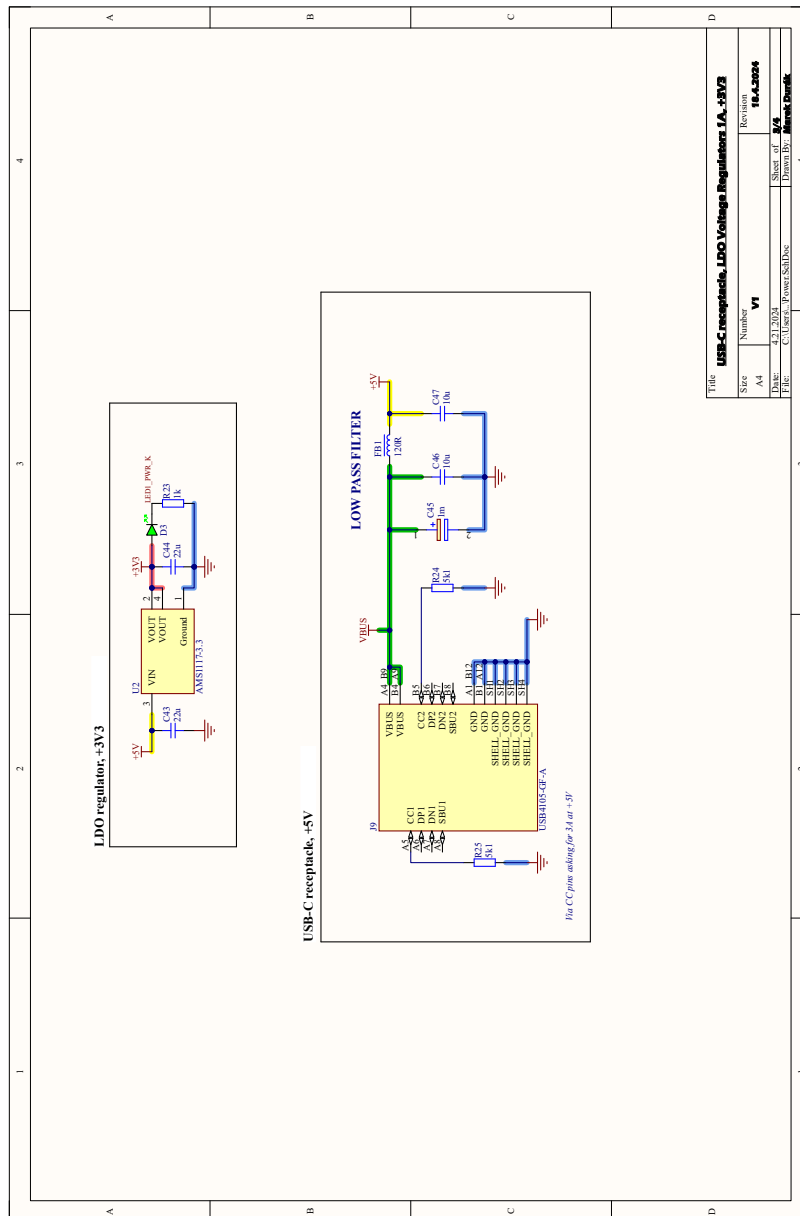
C.1 Část první



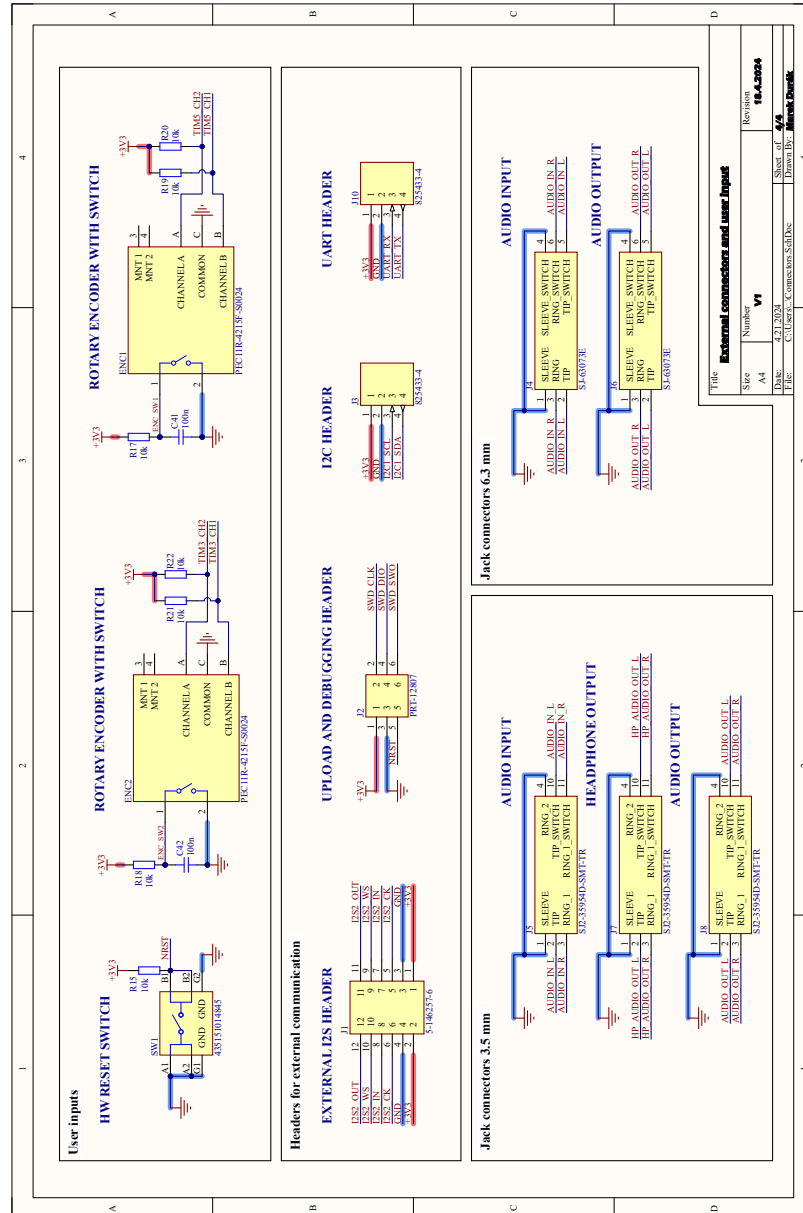
C.2 Část druhá



C.3 Část třetí



C.4 Část čtvrtá





Příloha D

Seznam ostatních příloh

Seznam externích příloh připojených k bakalářské práci je následující:

- Altium Designer → Altiovské projekty, vývojové desky a Antinode,
- Fusion 360 → Model krabičky pro Antinode,
- Matlab → Funkce pro získání koeficientů a vykreslování charakteristik,
- STM32 → STM32CubeIDE projekt obsahující všechny používané kódy.