

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

ŘÍDÍCÍ JEDNOTKA PRO DSP ZPRACOVÁNÍ A DISTRIBUCI  
ZVUKOVÉHO SIGNÁLU PRO EXTERNÍ AKTIVNÍ REPRODUKTORY

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. MARTIN ŠERÍK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# ŘÍDÍCÍ JEDNOTKA PRO DSP ZPRACOVÁNÍ A DISTRIBUCI ZVUKOVÉHO SIGNÁLU PRO EXTERNÍ AKTIVNÍ REPRODUKTORY

CONTROL UNIT FOR DSP AND AUDIO SIGNAL DISTRIBUTION TO ACTIVE EXTERNAL  
LOUDSPEAKERS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN ŠERÍK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. TOMÁŠ KRATOCHVÍL, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Martin Šerík

**ID:** 115283

**Ročník:** 2

**Akademický rok:** 2012/2013

## NÁZEV TÉMATU:

**Řídící jednotka pro DSP zpracování a distribuci zvukového signálu pro externí aktivní reproduktory**

## POKYNY PRO VYPRACOVÁNÍ:

Proveďte rozbor problematiky zpracování signálu pomocí DSP pro aktivní reproduktory. Navrhněte topologii řídicí jednotky pro zpracování vstupních analogových a digitálních zvukových signálů a jejich další distribuci pro externí aktivní reproduktory včetně datové komunikace.

Na základě rozboru navrhněte DSP systém včetně všech funkcí dle rozboru a pokynů vedoucího práce.

Navržený systém realizujte včetně sestavení příslušného SW. Ověřte správnou činnost realizovaného systému komplexním měřením včetně standardního protokolu o měření.

## DOPORUČENÁ LITERATURA:

[1] SMÉKAL, Z., SYSEL, P. Signálové procesory. Praha: Sdělovací technika, 2006.

[2] JÁN, J. Číslíkové zpracování, filtrace a analýzy signálů. Brno: Vutium, 2002.

[3] SMITH, W. S. The Scientist and Engineer's Guide to Digital Signal Processing [online]. Dostupné na [www: http://www.dspguide.com](http://www.dspguide.com)

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 24.5.2013

**Vedoucí práce:** doc. Ing. Tomáš Kratochvíl, Ph.D.

**Konzultanti diplomové práce:**

**prof. Dr. Ing. Zbyněk Raida**  
Předseda oborové rady

Výzkum realizovaný v rámci této diplomové práce byl finančně podpořen projektem  
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**  
operačního programu **Vzdělávání pro konkurenceschopnost**.



## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem  
a státním rozpočtem České republiky.

Tento příspěvek vzniknul za podpory projektu CZ.1.07/2.3.00/20.0007 WICOMT,  
financovaného z operačního programu Vzdělávání pro konkurenceschopnost



## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



## **ABSTRAKT**

Práca sa zaoberá rozborom problematiky číslicového spracovania signálu a návrhom riadiacej jednotky pre spracovanie a distribúciu zvukového signálu pre externé reproduktory. Je v nej popísaný spôsob funkcie delta-sigma AD prevodníkov, najpoužívanejších digitálnych filtrov a rýchlej Fourierovej transformácie. V práci je okrem výberu vhodného spôsobu prenosu spracovaného signálu do externých reproduktorov navrhnutý hardware zariadenia spolu s popisom vytvoreného firmwáre. V záveru práce sú diskutované funkcie a dosiahnuté parametre jednotky.

## **KLÍČOVÉ SLOVÁ**

DSP, AD prevod, digitálny filter, FFT, Ethernet, S/PDIF, ekvalizér

## **ABSTRACT**

The paper deals with analysis of the problems of digital signal processing and with design of the control unit for DSP and audio signal distribution to active external loudspeakers. It describes function of delta-sigma AD converters, the most widely used digital filters and Fast Fourier Transform. The processed signal will be send to external loudspeakers using an Ethernet interface. The paper also covers the design of signal processing unit's hardware and software. Achieved functions and parameters of the unit are discussed in conclusion.

## **KEYWORDS**

DSP, ADC, Digital Filter, FFT, Ethernet, S/PDIF, Equalizer

ŠERÍK, Martin. *Řídící jednotka pro DSP zpracování a distribuci zvukového signálu pro externí aktivní reproduktory*: diplomová práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2013. 103 s. Vedúci práce doc. Ing. Tomáš Kratochvíl, Ph.D.

## PREHLÁSENIE

Prehlasujem, že svoju diplomovú prácu na tému „Řídící jednotka pro DSP zpracování a distribuci zvukového signálu pro externí aktivní reproduktory“ som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich, autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení § 152 trestného zákona č. 140/1961 Sb.

BRNO .....

.....

(podpis autora)

# Pod'akovanie

Ďakujem konzultantovi diplomovej práce Ing. Petrovi Opravilovi a vedúcemu práce doc. Ing. Tomášovi Kratochvílovi, Ph.D. za odbornú pomoc a cenné rady pri spracovaní projektu. Ďalej chcem pod'akovať firme Bang & Olufsen s.r.o. za poskytnutú záštitu pri vypracovaní projektu.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	13
<b>1 Teoretický rozbor</b>	<b>15</b>
1.1 Ľudský sluch	15
1.2 Digitalizácia zvuku	18
1.3 Digitálne filtre	22
1.3.1 FIR filtre	22
1.3.2 IIR filtre	26
1.3.3 IIR filtre pre realizáciu digitálneho ekvalizéra	28
1.4 Fourierova transformácia	31
1.4.1 Rýchla Fourierova transformácia	31
1.5 Dátová komunikácia	34
1.5.1 Fast Ethernet	35
<b>2 Hardware SPU</b>	<b>37</b>
2.1 Bloková schéma zariadenia	37
2.1.1 MCU	38
2.1.2 IrDA prijímač	39
2.1.3 LCD	39
2.2 Vstup audio signálu	39
2.2.1 (Semi-)symetrický analógový vstup	39
2.2.2 S/PDIF digitálny vstup	43
2.3 Dátový vstup a výstup	45
<b>3 Firmware SPU</b>	<b>48</b>
3.1 Vývojové prostredie a RTOS	48
3.2 Štruktúra firmwaru	49
3.2.1 Audio vlákna	50
3.2.2 Ekvalizér	52
3.2.3 pcComm vlákno	53
3.2.4 Ostatné vlákna	53
<b>4 Komunikačný protokol</b>	<b>56</b>
4.1 Komunikácia s PC	56
4.2 Štruktúra UDP paketu	57
<b>5 Záver</b>	<b>58</b>

<b>Literatúra</b>	<b>60</b>
<b>Zoznam symbolov, veličín a skratiek</b>	<b>63</b>
<b>Zoznam príloh</b>	<b>64</b>
<b>A Schéma zapojenia SPU</b>	<b>65</b>
<b>B Návrhy motívu plošného spoja</b>	<b>72</b>
<b>C Osadzovacie plány DPS</b>	<b>75</b>
<b>D Zoznam súčiastok</b>	<b>78</b>
<b>E Fotodokumentácia</b>	<b>82</b>
<b>F Vývojové diagramy</b>	<b>84</b>
<b>G Meranie parametrov systému s SPU jednotkou</b>	<b>90</b>
<b>H btypes.h</b>	<b>94</b>
<b>I Prílohy na CD</b>	<b>103</b>
I.1 Dokumentácia . . . . .	103
I.2 Schémy SPU . . . . .	103
I.3 Plošné spoje SPU . . . . .	103
I.4 Podporné programy v SW MATLAB . . . . .	103
I.5 Firmware SPU . . . . .	103
I.6 Fotodokumentácia . . . . .	103

# ZOZNAM OBRÁZKOV

1	Bloková schéma systému ozvučenia obytnej jednotky . . . . .	13
1.1	Funkčný diagram stavby ucha . . . . .	16
1.2	Bloková schéma systému pre číslicové spracovanie signálu . . . . .	18
1.3	Principiálna schéma delta-sigma prevodníku . . . . .	19
1.4	Priamy prevod vzoriek na impulzy . . . . .	20
1.5	Sample & Hold výstup DA prevodníku . . . . .	21
1.6	Ideálna charakteristika rekonštrukčného filtra . . . . .	21
1.7	Frekvenčná odozva filtra s kľzavým priemerom . . . . .	23
1.8	Priebeh funkcie <i>sinc()</i> . . . . .	24
1.9	Orezaná funkcia <i>sinc()</i> a jej frekvenčná odozva . . . . .	25
1.10	Frekvenčné charakteristiky Hammingovho a Blackmanovho okna . . . . .	25
1.11	Odozva Chebyshevovho filtra s rôznym zvlnením . . . . .	28
1.12	FFT dekompozícia . . . . .	31
1.13	FFT syntéza . . . . .	33
1.14	Diagram FFT syntézy . . . . .	33
1.15	Motýlik – základný prvok FFT . . . . .	34
1.16	Štruktúra ethernetového rámca . . . . .	35
2.1	Bloková schéma hardwaru SPU . . . . .	37
2.2	Schéma zapojenia analógových vstupov . . . . .	41
2.3	Schéma zapojenia vstupného filtra . . . . .	41
2.4	Schéma zapojenia obvodu CS5361 . . . . .	42
2.5	Schéma zapojenia SPDIF vstupu . . . . .	44
2.6	Schéma zapojenia fyzickej vrstvy . . . . .	46
3.1	Bloková schéma firmwaru SPU . . . . .	50
3.2	Vývojový diagram pre audio vlákno . . . . .	51
3.3	Modul frekvenčných charakteristík filtrov sedem-pásmového ekvalizéru . . . . .	53
3.4	Vývojový diagram pre pcComm vlákno . . . . .	54
4.1	Štruktúra UDP paketu . . . . .	57
A.1	Schéma zapojenia napájania . . . . .	66
A.2	Schéma pripojenia periférií k STM32F4Discovery . . . . .	67
A.3	Schéma zapojenia AD prevodníka, 1. časť . . . . .	68
A.4	Schéma zapojenia AD prevodníka, 2. časť . . . . .	69
A.5	Schéma zapojenia S/PDIF modulu . . . . .	70
A.6	Schéma zapojenia fyzickej vrstvy Ethernetu . . . . .	71
B.1	Motív plošného spoja napájania, 102 x 60 mm, 1:1 . . . . .	72
B.2	Motív plošného spoja SPU jednotky, BOTTOM vrstva, 148 x 175 mm, 1:1 . . . . .	73

B.3	Motív plošného spoja SPU jednotky, TOP vrstva, 148 x 175 mm, 1:1	74
C.1	Osadzovací plán DPS SPU jednotky . . . . .	76
C.2	Osadzovací plán napájacej časti SPU jednotky, strana TOP . . . . .	77
C.3	Osadzovací plán napájacej časti SPU jednotky , strana BOTTOM . . . . .	77
E.1	Pohľad na predný a zadný panel zariadenia . . . . .	82
E.2	Pohľad na elektroniku v zariadení . . . . .	83
F.1	Vývojový diagram main vlákna . . . . .	84
F.2	Vývojový diagram led vlákna . . . . .	85
F.3	Vývojový diagram lcd vlákna . . . . .	86
F.4	Vývojový diagram irda vlákna . . . . .	87
F.5	Vývojový diagram audio vlákna . . . . .	88
F.6	Vývojový diagram pcComm vlákna . . . . .	89
G.1	Časový priebeh a modul frekvenčného spektra harmonického signálu na vstupe SPU jednotky . . . . .	91
G.2	Časový priebeh a modul frekvenčného spektra harmonického signálu na výstupe DA prevodníka v externých reproduktoroch . . . . .	91
G.3	Priebehy I <sup>2</sup> S signálov zmerané logickým analyzátorom . . . . .	92
G.4	Zaťaženie siete v závislosti na počtu prenášaných kanálov . . . . .	93

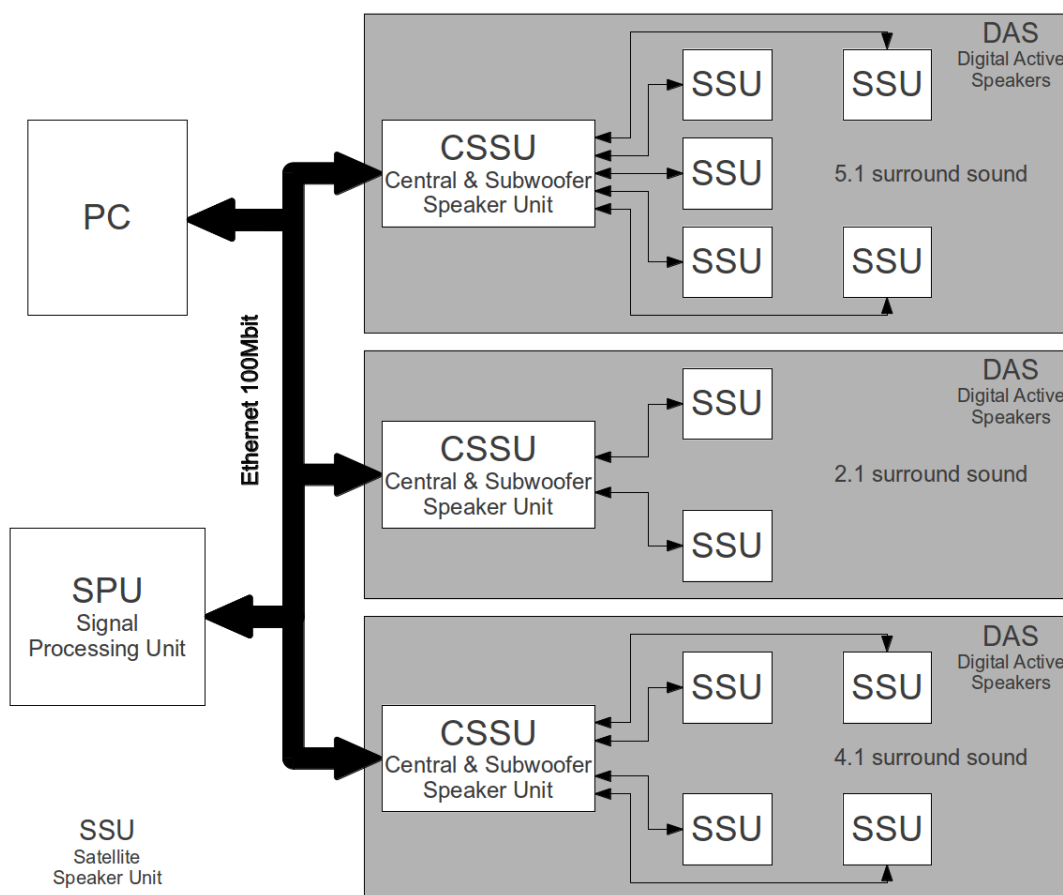
# ZOZNAM TABULIEK

1.1	Prehľad úrovni intenzity zvuku . . . . .	17
2.1	Výber aktívneho S/PDIF vstupu . . . . .	44
D.1	Zoznam súčiastok pre STM32F4Discovery schému . . . . .	78
D.2	Zoznam súčiastok pre AD_CS5361 schému . . . . .	79
D.3	Zoznam súčiastok pre SPDIF schému . . . . .	80
D.4	Zoznam súčiastok pre ETH_PHY schému . . . . .	80
D.5	Zoznam súčiastok pre SUPPLY schému . . . . .	81
G.1	Zaťaženie siete v závislosti na prenášanom počte kanálov . . . . .	93



# ÚVOD

Zadanie diplomovej práce je od firmy Bang & Olufsen s.r.o. Cieľom práce je preskúmať možnosti digitálneho spracovania, riadenia a distribúcie zvukového signálu pre ozvučenie obytnej jednotky. Následne je realizované zariadenie, na ktorom je implementovaná a prakticky odskúšaná funkcia systému v menšom merítku. Projekt je rozdelený na tri časti (vid' obr. 1). Jedna časť sa zaoberá návrhom a realizáciou riadiaceho softwaru a komunikačného protokolu (blok PC) [2], druhá časť realizáciou jednotky pre príjem, spracovanie a distribúciu zvukového signálu v digitálnej podobe, spolu s implementáciou komunikačného protokolu (blok SPU) a tretia časť je venovaná externým reproduktorom, ktoré prijímajú digitálny zvukový signál a prehrávajú ho (bloky DAS) [3]. Systém má byť plne digitálny, takže konverzia audio signálu do analógovej podoby sa deje až v externých reproduktoroch vo výstupnom filtre digitálneho zosilňovača.



Obr. 1: Blokovaná schéma systému ozvučenia obytnej jednotky

Cieľom diplomovej práce je navrhnuť a zrealizovať blok SPU. SPU má obsahovať

nasledujúce časti:

- DSP jednotku pre multikanálové spracovanie zvukového signálu,
- (semi-)symetrický analógový vstup zvukového signálu,
- digitálny S/PDIF vstup zvukového signálu,
- dátový výstup zvukového signálu pre externé aktívne reproduktory,
- obojsmernú dátovú komunikáciu umožňujúcu príjem dát z externých aktívnych reproduktorov,
- komunikačné rozhranie pre nastavenie, riadenie a monitoring systému,
- prijímač infračerveného diaľkového ovládania (poskytne B&O).

Diplomová práca je rozdelená na štyri logické celky. Prvý celok (kapitola 1 *Teoretický rozbor*) sa zaoberá rozborom základných techník číslicového spracovania signálov. Stručne je tu rozobraný princíp fungovania AD prevodníkov, spôsoby filtrovania signálu, fungovania rýchlej Fourierovej transformácie a výber vhodného typu dátovej komunikácie.

V druhej kapitole (2 *Hardware SPU*) sa pristupuje k vlastnému návrhu hardwaru SPU. Postupne sú tu popísané jednotlivé bloky zariadenia.

V tretej časti je popísaný firmware zariadenia spolu s výberom vývojového prostredia a systému reálneho času.

V kapitole 4 *Komunikačný protokol* je uvedený komunikačný protokol medzi SPU jednotkou a PC. Keďže cieľom diplomovej práce nebolo vymyslieť tento protokol, je popis značne zosťručený. Podrobnosti je možné nájsť v [2]. Ďalej sa v tejto kapitole nachádza štruktúra UDP paketu používaného pre prenos audio signálu.

# 1 TEORETICKÝ ROZBOR

V tejto kapitole sa oboznámime s funkciou ľudského sluchu, spôsobom vnímania zvuku, základom spracovania signálu pomocou digitálnych filtrov a princíp funkcie rýchlej Fourierovej transformácie.

Na záver kapitoly bude vybraný vhodný spôsob dátovej komunikácie.

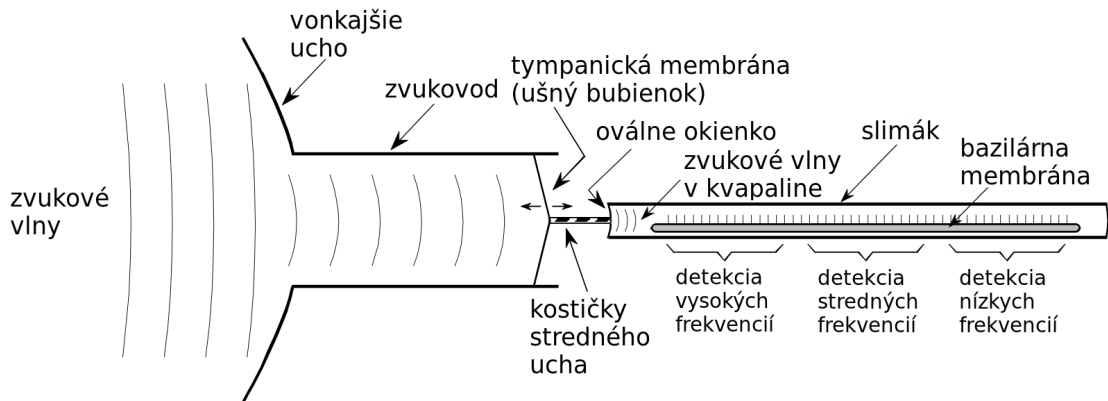
## 1.1 Ľudský sluch

Predtým než sa dostaneme k popisu fungovania ľudského sluchu, oboznámime sa so stavbou ucha, tak ako je schematicky zachytená na obr. 1.1. Ucho začína vonkajším uchom, ktoré je zložené z dvoch častí – ušnice a zvukovodu. Pomocou vonkajšieho ucha je zvuk smerovaný k citlivému strednému a vnútornému uchu. Rozhranie medzi koncom zvukovodu a začiatkom stredného ucha tvorí tympanická membrána, tiež nazývaná ušný bubienok. Za ňou sa smerom do lebky nachádzajú kostičky stredného ucha, konkrétne kladívko, nákovka a strmienok. Tieto slúžia na zosilnenie tlaku zvukovej vlny pred tým, než dosiahne vnútorné ucho. V ňom sú vibrácie premenené na nervové impulzy. Vnútorné ucho je tiež inak nazývané slimák alebo *kochlea*. Slimák má tvar slimačej ulity, jedná sa o rúrku dlhú zhruba 3 cm s priemerom 2 mm, naplnenú tekutinou. Na obr. 1.1 je kvôli prehľadnosti zakreslený ako rovný obdĺžnik. Na dne slimáka sa nachádza bazilárna membrána, podporná štruktúra pre viac ako 12 000 vlásokových buniek, vytvárajúcich kochleárny nerv. Bazilárna membrána má rôznu tuhosť v závislosti na vzdialenosti od bubienka. Chová sa ako frekvenčný analyzátor, na vysokých frekvenciách rezonuje tam, kde je pevná (v blízkosti oválneho okienka) a na nízkych frekvenciách rezonuje na pružných úsekoch (na druhom konci slimáka). Pri rezonancii vybudí nervové bunky, ktoré predávajú informáciu o budiacej frekvencii mozgu na ďalšie spracovanie. Tento spôsob vnímania zvuku sa označuje ako *rezonančný princíp*, alebo tiež *teória miesta*.<sup>1</sup>

Pri analýze zvuku sa uplatňuje aj druhý princíp, tzv. *frekvenčný*.<sup>2</sup> Nervové bunky sú schopné generovať krátke elektrické pulzy, nazývané akčné potenciály. Napríklad 200 hertzový zvukový signál môže byť reprezentovaný neurónom produkujúcim 200 akčných potenciálov za sekundu. Pre prevod frekvencií nad 500 hertzov sa používa niekoľko nervových dráh simultánne. Vďaka tomu je možné využívať frekvenčný princíp do asi 4 kHz. Nad touto hodnotou sa uplatňuje výlučne rezonančný princíp.

<sup>1</sup>v anglickej literatúre sa stretáme s pojmami *place principle*, prípadne *place theory* [1]

<sup>2</sup>v anglickej literatúre označovaný ako *volley principle* alebo *volley theory* [1]



Obr. 1.1: Funkčný diagram stavby ucha

V tab. 1.1 je uvedený vzťah medzi intenzitou zvuku a akustickým tlakom. Intenzita zvuku sa vyjadruje v logaritmickej škále jednotkami nazývanými *decibel SPL* (Sound Power Level). V tejto mierke je 0 dB SPL najslabší zvuk detekovateľný ľudským sluchom. Rozdiel medzi najslabším a najsilnejším zvukom, ktorý nespôsobuje bolesť, je asi 120 dB (milión násobok amplitúdy). Priemerný poslucháč dokáže rozoznať zmenu hlasitosti zvuku pri zmene jeho intenzity o 1 dB (12% zmena amplitúdy). Existuje teda len asi 120 úrovní hlasitosti, od najtichšieho šepotu po rachot rockového koncertu. Pre predstavu o jemnosti sluchového ústrojenstva uveďme, že pri počúvaní veľmi tichých zvukov kmitá bubienok s odchýlkou menšou ako priemer jedinej molekuly. [1]

Frekvenčný rozsah zvuku vnímaného ľudským sluchom je všeobecne uvažovaný medzi kmitočtami 20 Hz až 20 kHz (prípadne 16 kHz). S pribúdajúcim vekom sa rozsah hlavne v oblasti vysokých frekvencií znižuje. Citlivosť sluchu nie je na všetkých frekvenciách rovnaká. Najcitlivejší je na 3 kHz, kde dokáže detekovať 0 dB SPL. Pri 100 Hz ale nepočuje zvuk s intenzitou menšou ako 100 dB SPL. Priemerný poslucháč dokáže rozoznať dva tóny, vzdialené od seba o 0.3% na 3 kHz, na 100 Hz sú to 3%. Pre porovnanie, medzi susediacimi klávesmi na klavíri je rozdiel vo frekvencii asi 6%.

$L_p$ [dB SPL]	$p$ [Pa]	zdroj zvuku
0	0.000 020	prah počuteľnosti zdravej osoby, na 3 kHz
10	0.000 063	ľudský dych zo vzdialenosti troch metrov
30	0.000 632	ticho v divadle, noc v púšti
40	0.002 000	ticho na sídlisku v noci, ľudský šepot
50	0.006 325	tichá kaviareň
60	0.020	zvyčajná hlasitosť normálnej konverzácie
80	0.200	vysávač zo vzdialenosti jeden meter
90	0.632	nákladné auto zo vzdialenosti jeden meter
100	2	diskotéka, televízor alebo veža pri najvyššej hlasitosti
120	20	rockový koncert, vzlietnutie prúdového lietadla (100 m), na nízkych frekvenciách cíti telo vibrácie
130	63	vojenský dychový orchester zblízka, prah bolesti na všetkých frekvenciách
150	632	prúdový motor (30 m), bolesť v hrudníku, rozmazané videnie, poškodzovanie sluchu na všetkých frekvenciách
180	20 000	motor rakety (30 m), dunenie vráskavca ozrutného (1 m), výbuch sopky Krakatoa (r. 1883) vo vzdialenosti 160 km

Tab. 1.1: Prehľad úrovni intenzity zvuku ( $L_p$ ) vo vzťahu k akustickému tlaku ( $p$ ) a zdroju zvuku [5].

## 1.2 Digitalizácia zvuku

Aby sme boli schopní spracovať signály spojitého charakteru s pomocou číslicovej techniky, musíme ich previesť z analógového do číslicového tvaru. To je možné pomocou techniky, súhrnne označovanej ako AD prevod (ADC – Analog-to-Digital Conversion). Po spracovaní je možné previesť digitálny signál späť do analógového sveta pomocou číslicovo-analógového prevodu (DAC – Digital-to-Analog Conversion).

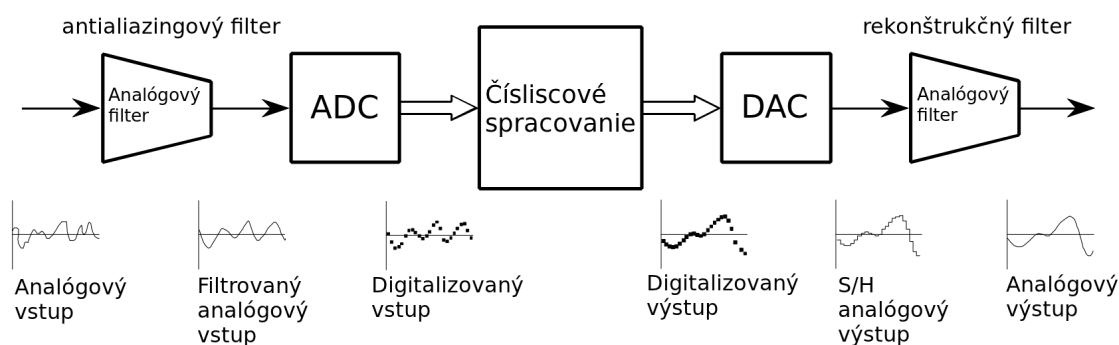
### AD prevod

Digitalizovaný signál sa od svojho analógového originálu líši hlavne v dvoch aspektoch: je *vzorkovaný* a *kvantizovaný*. Vzorkovanie prebieha tak, že odčítame hodnotu amplitúdy signálu každých  $T_{vz}$  sekundy, kde  $T_{vz}$  je perióda, s ktorou vzorkujeme signál. Ku kvantizácii dochádza už samotným vyjadrením amplitúdy signálu číslom s konečnou presnosťou. Kvantizácia signálu je určená bitovým rozlíšením použitého AD prevodníka.

V procese digitalizácie hrá významnú úlohu frekvencia, s akou vzorkujeme signál. Jej minimálna veľkosť je určená vzorkovacím, Nyquistovým, teorémom. Ten hovorí, že minimálna vzorkovacia frekvencia musí byť aspoň dva krát väčšia ako najvyšší kmitočet obsiahnutý vo vzorkovanom signáli. Pokiaľ tento predpoklad nesplníme dôjde k aliažingu, čiže degenerácii signálu.

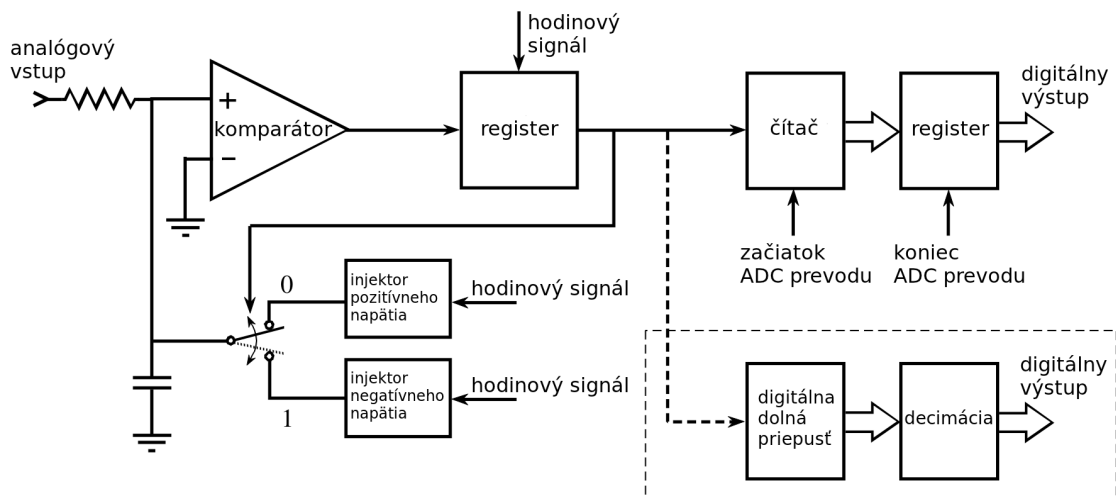
Aby sme zabránili aliažingu, vkladáme pred AD prevodník antialiažingový filter. Jedná sa o dolnú priepusť, ktorá frekvenčne obmedzí vstupný signál. Inými slovami odstráni všetky zložky s kmitočtom vyšším ako polovica vzorkovacej frekvencie.

Na obr. 1.2 vidíme pozíciu AD a DA prevodníkov v DSP systéme.



Obr. 1.2: Bloková schéma systému pre číslicové spracovanie signálu

V dnešnej dobe je najpoužívaným typom AD prevodníku pre všeobecné účely *Delta-sigma* prevodník. Na obr. 1.3 je zobrazený jeho blokový diagram. Vstupný



Obr. 1.3: Principiálna schéma delta-sigma prevodníku

komparátor porovnáva napätie na plusovej svorke s potenciálom zeme. Napätie na kladnej svorke je dané súčtom napätia na kondenzátore a napätia na analógovom vstupe. Kondenzátor je pripojený na elektronický prepínač riadený výstupom komparátora v okamžiku nástupnej (prípadne zostupnej) hrany hodinového signálu. Elektronický prepínač pripája na kondenzátor injektor negatívneho alebo pozitívneho napätia. Ak je na výstupe komparátora logická jednotka (na svorke + je väčšie napätie ako na svorke -), pripojí sa na kondenzátor záporné napätie a obdobne, ak je na výstupe komparátora logická nula (na svorke - je väčšie napätie ako na svorke +), pripojí sa na kondenzátor kladné napätie. Spätná väzba teda pôsobí proti napätiu na kladnej svorke komparátora, tak aby sa na nej udržovala napäťová úroveň potenciálu zeme.

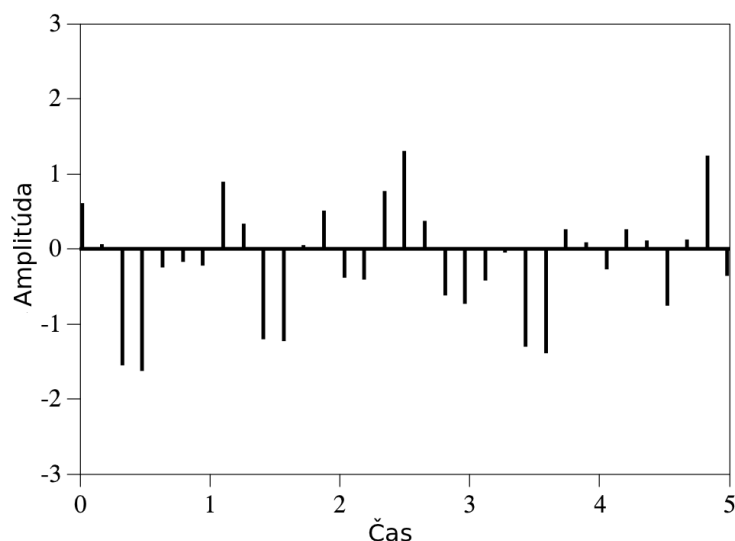
Predstavme si, že je na vstupe kladné napätie. Na výstupe komparátora je potom viac logických jednotiek ako núl. Obdobne, pokiaľ je na vstupe delta-sigma prevodníku záporné napätie, je na výstupe komparátora viac núl ako jednotiek. Po pripojení vstupu na zem bude na výstupe rovnaký počet jednotiek aj núl. Relatívny počet logických núl a jednotiek teda určuje úroveň vstupného signálu. V ďalšom kroku stačí počítať jednotky počas jedného cyklu prevodu, daného bitovou šírkou AD prevodníka. Napríklad pre 12 bitový AD prevodník znamená 4096 jednotiek maximálnu kladnú úroveň vstupného signálu, 0 jednotiek jeho maximálnu zápornú úroveň a 2048 jednotiek korešponduje s nulovým vstupným napätím.

Pre digitalizáciu signálu môžeme miesto počítania jednotiek, zapojiť za výstup komparátora digitálnu dolnú priepusť, ako je to naznačené na obr. 1.3. Priepusť prevedie (rekonštruuje) prúd jednotiek a núl na jednotnú úroveň, danú ich priemerom (podobne, ako by to spravila analógová dolná priepusť). Po decimácii získame na

výstupe vstupný signál v digitálnej podobe. Tento postup sa s výhodou používa pre digitalizáciu zvuku. Je ale nevhodný pre signály kódované v časovej oblasti, pretože pomocou tohto postupu nedostávame informáciu o tom, kedy bola jednotlivá vzorka získaná. [1]

## DA prevod

Najjednoduchší spôsob, ako konvertovať signál v digitálnej podobe do analógovej formy, je zobrať jednotlivé vzorky z pamäte a previesť ich na impulzy (viď obr. 1.4). Takýto signál obsahuje frekvenčné spektrum originálneho signálu spolu s jeho kópiami na násobkoch vzorkovacieho kmitočtu. Na odstránenie nadbytočných frekvencií, zaradíme za prevodník dolnú priepusť s medznou frekvenciou rovnou polovici vzorkovacieho kmitočtu.

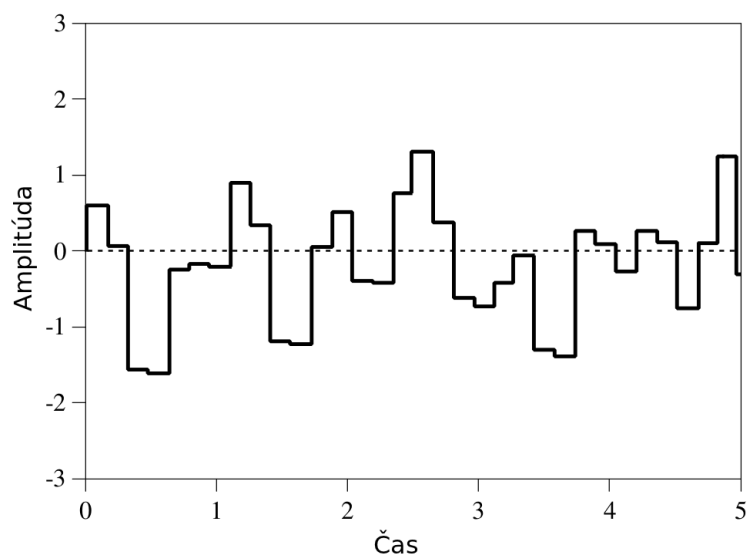


Obr. 1.4: Priamy prevod vzoriek na impulzy

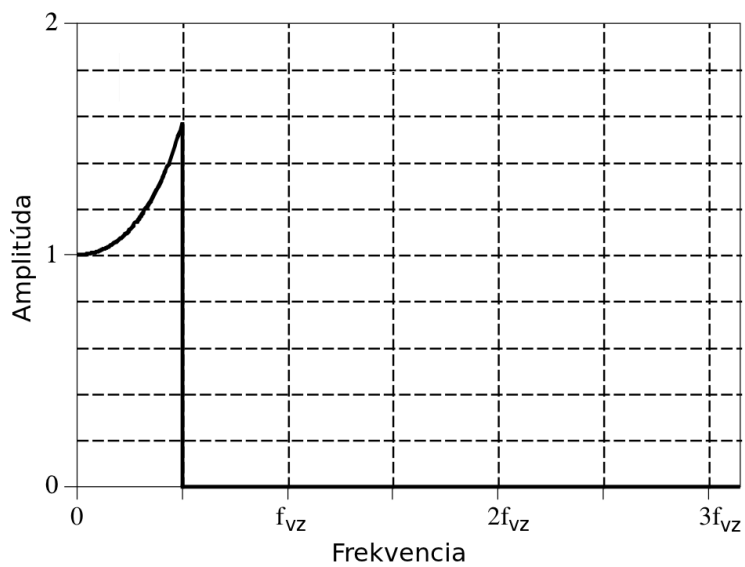
Problém je, že generovanie úzkych pulzov nie je vôbec jednoduché. Rieši sa to tak, že sa napäťová úroveň daná aktuálnou vzorkou podrží až do príchodu ďalšej vzorky (obr. 1.5).

Takýto Sample & Hold analógový signál zavádza do frekvenčného spektra skreslenie. Pôvodné spektrum je vynásobené funkciou  $\text{sinc}(\pi f/f_{vz})$ . Na obnovenie pôvodného spektra je potrebné zaradiť za DA prevodník rekonštrukčný filter, ktorý odstráni všetky frekvencie nad vzorkovacím kmitočtom a koriguje vplyv funkcie  $\text{sinc}(\pi f/f_{vz})$ . Ideálna charakteristika takéhoto filtru je na obr. 1.6. [1]





Obr. 1.5: Sample & Hold výstup DA prevodníku



Obr. 1.6: Ideálna charakteristika rekonštrukčného filtra

## 1.3 Digitálne filtre

Najpriamočiarejší spôsob implementácie digitálnych filtrov je konvolúciou vstupného signálu s impulznou odozvou digitálneho filtru. Keď je impulzná odozva použitá takýmto spôsobom, hovoríme o jadre filtru (filter kernel). Podľa toho aké jadro filtru navrhne, môžeme vytvoriť filter s takmer ľubovoľnou frekvenčnou charakteristikou. Filter implementovaný konvolúciou, má každú vzorku výstupného signálu vypočítanú vážením vstupných vzoriek (násobením koeficientom) a ich spočítaním.

Iný spôsob vytvárania digitálnych filtrov je pomocou rekurzie. Rekurzívne filtre používajú na rozdiel od filtrov realizovaných konvolúciou predchádzajúce *výstupné* vzorky. Namiesto použitia jadra filtru sú rekurzívne filtre definované sadou rekurzívnych koeficientov.

Rekurzívne filtre sú tiež nazývané filtre s nekonečnou impulznou odozvou (IIR – Infinite Impulse Response) a konvolučné filtre sú nazývané filtre s konečnou impulznou odozvou (FIR – Finite Impulse Response).

### 1.3.1 FIR filtre

FIR filtre môžeme rozdeliť podľa použitia do troch skupín:

- vhodné pre časovú oblasť (kľzavý priemer),
- pre frekvenčnú oblasť (sinc filtre s okienkovou funkciou – Windowed-sinc),
- filtre plne definované používateľom (vlastný dizajn).

#### Filtre realizujúce kľzavý priemer

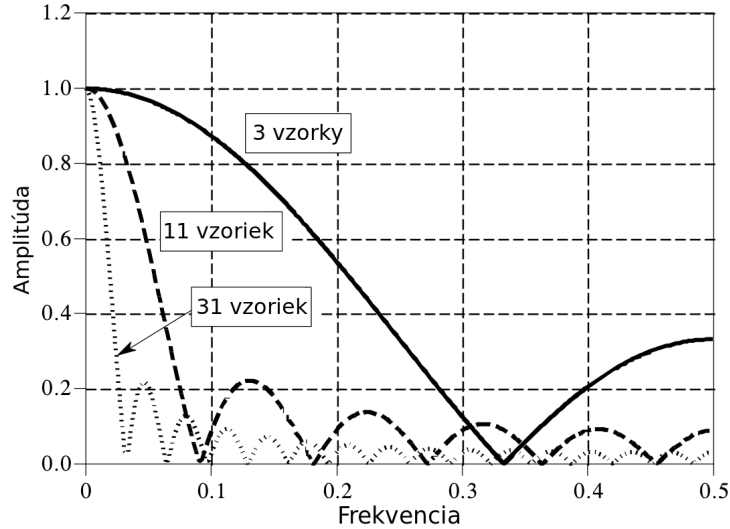
Filtre, ktoré realizujú kľzavý priemer (Moving Average Filters) sú najpoužívanejšími filtrami v číslicovom spracovaní signálu. Hlavne pretože sú jednoduché na použitie a pochopenie. Kľzavý priemer je optimálny na odstránenie náhodného šumu (bieleho šumu) pri zachovaní ostrých hrán odozvy na jednotkový skok. To ho robí výborným nástrojom pre signály kódované v časovej oblasti. Vo frekvenčnej oblasti sa veľmi neuplatní, pretože má len malú schopnosť oddeliť jedno pásmo frekvencií od druhého.

Spôsob implementácie kľzavého filtru je vyjadrený rovnicou 1.1:

$$y[i] = \frac{1}{M} \sum_{j=-(M-1)/2}^{(M-1)/2} x[i+j], \quad (1.1)$$

kde  $x[]$  je vstupný signál,  $y[]$  je výstupný signál,  $i$  je číslo vzorky a  $M$  je počet priemerovaných hodnôt (v tomto vzťahu musí byť  $M$  nepárne číslo).

Frekvenčná odozva filtru realizujúceho kľzavý priemer je na obrázku 1.7. Frekvencia na osi  $x$  je normovaná k vzorkovaciemu kmitočtu.



Obr. 1.7: Frekvenčná odozva filtru s kľzavým priemerom

Odozva sa môže vyjadriť funkciou:

$$H[f] = \frac{\sin(\pi f M)}{M \sin(\pi f)}, \forall f \neq 0, \quad (1.2)$$

čo je vlastne Fourierova transformácia obdĺžnikového pulzu. Kľzavý priemer realizuje veľmi zlú dolnú priepusť, kvôli pomalému roll-off faktoru a malému potlačeniu signálu v nepriepustnom pásme.

Veľká výhoda tohto typu filtru leží v možnosti jeho implementácie pomocou rýchleho a výpočtovo nenáročného algoritmu. Využijeme pritom, že súčet doterajších vzoriek vstupného signálu je uložený v predchádzajúcej výstupnej vzorke. Táto myšlienka je zachytená v rovnici 1.3. Člen  $x[i-q]$  je najstaršia vzorka vstupného signálu, ktorú musíme odpočítať, aby mal filter konečnú, konštantnú dĺžku.

$$y[i] = y[i-1] + x[i+p] - x[i-q], \quad (1.3)$$

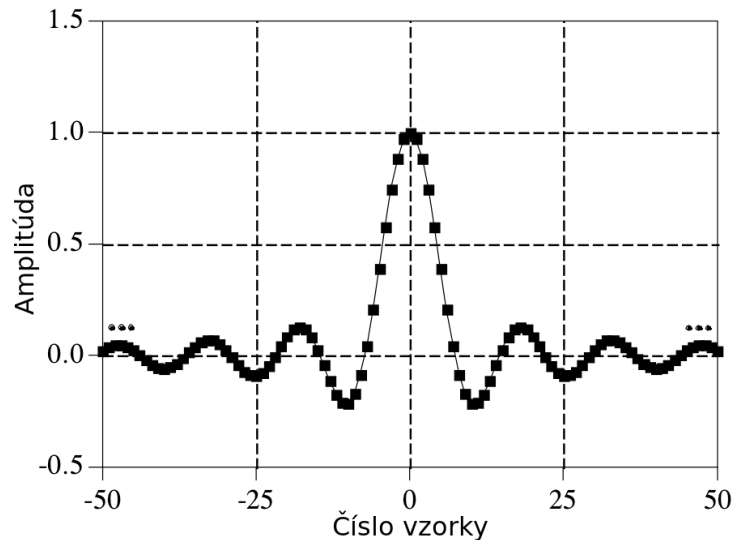
pričom  $p = (M-1)/2$ ,  $q = p+1$ ,  $x[]$  je vstupný signál,  $y[]$  je výstupný signál,  $i$  je číslo vzorky a  $M$  je počet priemerovaných hodnôt (nepárne číslo). Prvý bod výstupného signálu ( $y[]$ ) musí byť vypočítaný pomocou rovnice 1.1. [1]

### Windowed-sinc filtre

Windowed-sinc filtre majú impulznú odozvu funkcie  $\text{sinc}()$  vynásobenej zvoleným oknom. Sú používané na oddelenie jedného frekvenčného spektra od druhého. Sú veľmi stabilné, pri ich používaní nenarazíme na nepríjemné prekvapenia a dajú sa s nimi dosiahnuť výborné výsledky. Na rozdiel od frekvenčnej oblasti je v časovej

oblasti ich výkon slabý. V odozve na jednotkový skok nájdeme výrazný prekmit a zvlnenie. Pri implementácii klasickou konvolúciou sú relatívne pomalé, zrýchlenie dosiahneme použitím konvolúcie realizovanej pomocou rýchlej Fourierovej transformácie.

Modul kmitočtovej charakteristiky ideálnej dolnej priepusti má obdĺžnikový tvar. V časovej oblasti sa jedná o funkciu  $\text{sinc}()$ . Jadro filtra s ideálnou frekvenčnou odozvou je teda funkcia  $\text{sinc}()$ . Problém je, že  $\text{sinc}()$  je funkcia, ktorej amplitúda sa blíži k nule až v nekonečne (viď obr. 1.8). Takýto filter sa nedá prakticky implementovať. Po redukování počtu vzoriek (vynásobením obdĺžnikovým oknom) vznikne vo frekvenčnej odozve nežiaduce zvlnenie a zákmity (obr. 1.9). Na potlačenie týchto nežiaducich prejavov bola vytvorená sada okien, ktoré orežú funkciu  $\text{sinc}()$  vhodnejším spôsobom.



Obr. 1.8: Priebeh funkcie  $\text{sinc}()$

Existuje viacero okien s rôznymi vlastnosťami, napríklad Bartlettove, Hannove, Blackmanove alebo Hammingove okno. Z nich posledné dve sú najpoužívanejšie.

Hammingove okno je popísané funkciou 1.4

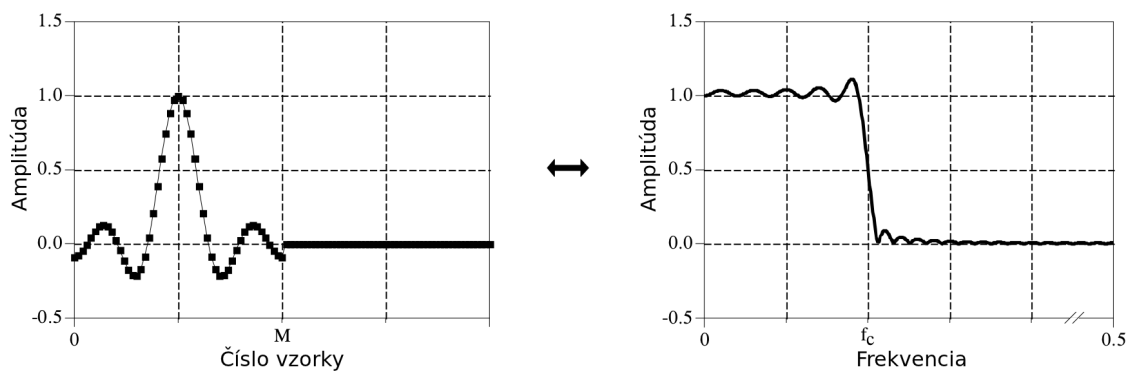
$$w[i] = 0.54 - 0.45 \cos(2\pi i/M) \quad (1.4)$$

a Blackmanove okno rovnicou 1.5

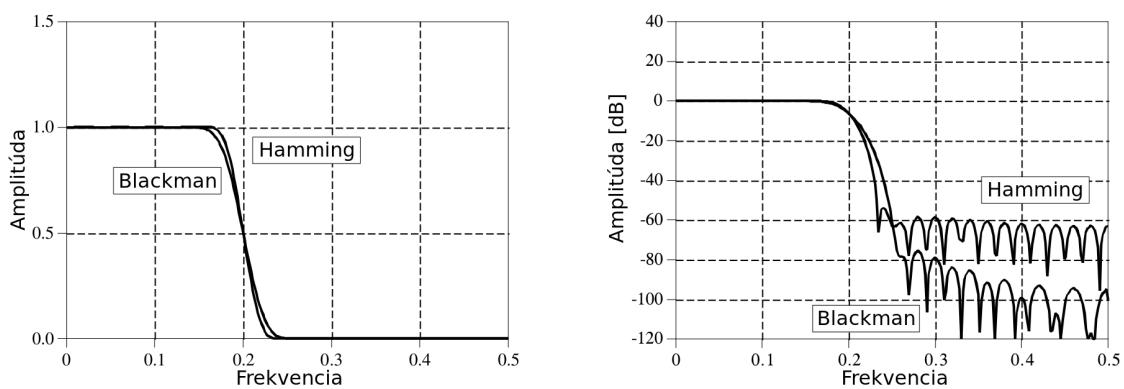
$$w[i] = 0.42 - 0.5 \cos(2\pi i/M) + 0.08 \cos(4\pi i/M), \quad (1.5)$$

kde  $i$  sa mení od 0 do  $M$ , výsledkom je teda filter dlhý  $M + 1$  vzoriek.

Z obr. 1.10 je vidieť, že Hammingove okno má lepší roll-off faktor ako Blackmanove (asi o 20 % [1]). To má zase lepší útlm v nepriepustnom pásme (-74 dB so zvlnením 0.02 %) oproti Hammingovmu oknu s -53 dB útlmom a zvlnením 0.2 %.



Obr. 1.9: Orezaná funkcia  $\text{sinc}()$  a jej frekvenčná odozva



Obr. 1.10: Frekvenčné charakteristiky Hammingovho a Blackmanovho okna

Jadro windowed-sinc filtru vypočítame pomocou rovnice 1.6

$$h[i] = K \frac{\sin(2\pi f_C(i - M/2))}{i - M/2} \left[ 0.42 - 0.5 \cos\left(\frac{2\pi i}{M}\right) + 0.08 \cos\left(\frac{4\pi i}{M}\right) \right], \quad (1.6)$$

kde  $f_C$  je medzná frekvencia vyjadrená ako hodnota medzi 0 a 0.5 (pomer k vzorkovacej frekvencii),  $M$  je dĺžka jadra filtru (párne celé číslo) a  $K$  slúži na prispôsobenie zisku na nulovej frekvencii. Pri  $i = M/2$  je  $h[i] = 2\pi f_C K$ . [1]

### 1.3.2 IIR filtre

IIR filtre umožňujú získať dlhú impulznú odozvu efektívnym spôsobom, bez nutnosti počítať konvolúciu. Tento spôsob je založený na využití rekurzív. Rovnica 1.7 ukazuje ako sa počíta hodnota aktuálnej vzorky. Využíva sa pri tom hodnota vstupného signálu, aj starších hodnôt výstupného signálu. Každá hodnota výstupného signálu sa určí vynásobením hodnôt vstupného signálu koeficientmi  $a$  a spočítaním s hodnotami výstupných vzoriek vynásobených koeficientmi  $b$ . V praxi sa nepoužíva viac ako tucet koeficientov (pri väčšom množstve sa stáva filter ľahko nestabilným).

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + a_3x[n-3] + \dots + b_1y[n-1] + b_2y[n-2] + b_3y[n-3] + \dots, \quad (1.7)$$

kde  $a$  a  $b$  sú koeficienty,  $x[]$  sú vstupné vzorky,  $y[]$  sú výstupné vzorky a  $n$  je poradové číslo vzorky.

Podobne ako sme rozdelili FIR filtre na tri skupiny, môžeme rozdeliť aj IIR filtre:

- vhodné pre časovú oblasť (s jedným pólom (Single pole)),
- vhodné pre frekvenčnú oblasť (Chebyshev),
- vlastný návrh (iteratívny dizajn).

### Jednopolové rekurzívne filtre

Pomocou jednopolových filtrov realizujeme celkom jednoducho ekvivalenty analógových RC a CR článkov.

Filter s koeficientmi  $a_0 = 0.15$  a  $b_1 = 0.85$  je ekvivalentom integračného RC článku. Keď zmeníme koeficienty  $a_0 = 0.93$ ,  $a_1 = -0.93$  a  $b_1 = 0.86$  získame obdobu derivačného CR článku. Pomocou týchto jednoduchých filtrov môžeme odstrániť jednosmernú zložku, vysokofrekvenčný šum, tvarovať signál, atď. Ľahko sa programujú a nie sú výpočtovo náročné.

Hodnoty koeficientov sa určia pomocou nasledujúcich rovníc:

- pre dolnú priepusť:

$$\begin{aligned}a_0 &= 1 - x \\ b_1 &= x\end{aligned}\tag{1.8}$$

- pre hornú priepusť:

$$\begin{aligned}a_0 &= (1 + x)/2 \\ a_1 &= -(1 + x)/2 \\ b_1 &= x\end{aligned}\tag{1.9}$$

Neznámu  $x$  v rovniciach 1.8 a 1.9 vypočítame nasledovne

$$x = e^{-1/d},\tag{1.10}$$

$d$  je požadovaná časová konštanta filtru. Inými slovami počet vzoriek, kým dosiahne amplitúda výstupného signálu 63.2 % amplitúdy vstupného signálu pre dolnú priepusť (respektíve 36.8 % pre hornú priepusť).

Vzťah medzi medznou frekvenciou a parametrom  $x$  je daný rovnicou 1.11

$$x = e^{-2\pi f_C},\tag{1.11}$$

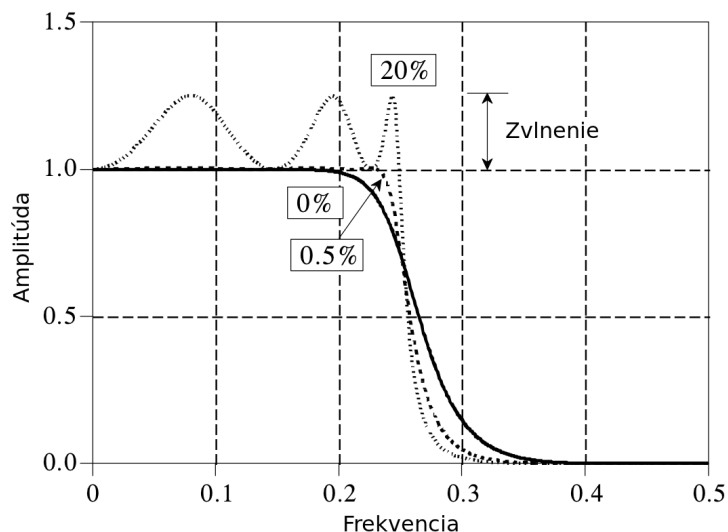
kde  $f_C$  je medzná frekvencia filtru.

## Chebyshevove filtre

Chebyshevove filtre sa používajú na oddelenie jedného frekvenčného pásma od druhého. Napriek tomu, že neposkytujú až také extrémne parametre ako windowed-sinc filtre, využívajú sa v mnohých situáciach. Hlavná výhodná vlastnosť Chebyshevových filtrov je rýchlosť, pretože namiesto konvolúcie sa na ich výpočet využíva rekurgia. Na zistenie koeficientov filtra sa využíva Z-transformácia.

Na obr. 1.11 je frekvenčná odozva Chebyshevovho filtru. Na dosiahnutie rýchlejšieho prechodu medzi priepustným a nepriepustným pásmom, sa využíva povolenie zvlnenia v priepustnom pásme. Existujú aj Chebyshevove filtre so zvlnením iba v nepriepustnom pásme, ale tie sa používajú iba zriedka. Dôležitým typom je tzv. eliptický filter. Ten má zvlnenie aj v priestupnom aj v nepriestupnom pásme. Tým pádom poskytuje najlepší roll-off na najmenší počet pólov. Nevýhodou je jeho zložitý návrh.

Limitácia rekurzívnych filtrov leží hlavne vo výpočtovej náročnosti. Napríklad filter šiesteho rádu má veľkosť niektorých koeficientov rádovo na úrovni  $10^{-10}$ . Takto malé hodnoty sa pri použití jednoduchej presnosti (single precision) čísel s plávajúcou desatinnou čiarkou strácajú v zaokrúhľovacom šume. Použitá presnosť čísel v programe teda limituje maximálny počet pólov Chebyshevovho filtru. Maximálny počet pólov pri jednoduchej presnosti závisí aj od medznej frekvencie a pohybuje sa v rozsahu 4 až 20 pólov pre frekvencie medzi 0.02 až 0.25 násobkom vzorkovacieho kmitočtu. [1]



Obr. 1.11: Odozva Chebyshevovho filtru s rôznym zvlnením

### 1.3.3 IIR filtre pre realizáciu digitálneho ekvalizéra

Pre realizáciu digitálneho ekvalizéra je nutné použiť filtre s nastaviteľným ziskom. Pre najnižšie pásmo kmitočtov je použitý filter typu *low shelving*, pre stredné pásma kmitočtov sú použité *peak* filtre a pre najvyššie pásmo je použitý *high shelving* filter.

Pre kvalitný ekvalizér je vhodné použiť filtre štvrtého radu. Pretože so zvyšujúcim sa radom filtru stúpa citlivosť IIR filtrov na kvantizáciu koeficientov, je vhodné realizovať filtre vyšších radov treťou kanonickou formou, čiže sériovým zapojením filtrov druhého radu.

V ďalšom texte uvedieme vzťahy pre výpočet koeficientov filtrov druhého radu.

#### Shelving filtre

Vstupné parametre pre shelving filtre sú:

- $F_c$ : medzná frekvencia filtru, frekvencia na ktorej sa zmení zisk filtru o polovicu,
- $F_s$ : vzrokovacia frekvencia,
- $S$ : sklon,
- $Gain_{dB}$ : zisk filtru.

Pre výpočet normalizovaného zisku  $Gain$  použijeme pre obidva typy filtrov vzťah 1.12.

$$Gain = 10^{(Gain_{dB}/40)}, \quad (1.12)$$

kde  $Gain_{dB}$  je požadovaný zisk filtru v decibeloch.



Koeficienty  $\alpha$  a  $\beta$ , používané v ďalších výpočtoch, určíme podľa rovníc 1.13 a 1.14.

$$\alpha = \frac{\sin(\theta_c)}{2} \cdot \sqrt{\left( \left( Gain + \left( \frac{1}{Gain} \right) \right) \cdot \left( \frac{1}{S} - 1 \right) + 2 \right)}, \quad (1.13)$$

$$\beta = 2 \cdot \alpha \cdot \sqrt{Gain}, \quad (1.14)$$

kde  $S$  je sklon a parameter  $\theta_c$  sa vypočíta nasledovne:

$$\theta_c = \frac{2\pi F_c}{F_s}. \quad (1.15)$$

Pre výpočet koeficientov low shelving filtru potom platia vzťahy 1.16

$$\begin{aligned} a_0 &= (Gain + 1) + (Gain - 1) \cdot \cos \theta_c + \beta \\ a_1 &= -2 \cdot (Gain - 1) + (Gain + 1) \cdot \cos \theta_c \\ a_2 &= (Gain + 1) + (Gain - 1) \cdot \cos \theta_c - \beta \\ b_0 &= Gain \cdot ((Gain + 1) - (Gain - 1) \cdot \cos \theta_c + \beta) \\ b_1 &= 2 \cdot Gain \cdot ((Gain - 1) - (Gain + 1) \cdot \cos \theta_c) \\ b_2 &= Gain \cdot ((Gain + 1) - (Gain - 1) \cdot \cos \theta_c - \beta) \end{aligned} \quad (1.16)$$

a pre high shelving filter vzťahy 1.17

$$\begin{aligned} a_0 &= (Gain + 1) - (Gain - 1) \cdot \cos \theta_c + \beta \\ a_1 &= 2 \cdot (Gain - 1) - (Gain + 1) \cdot \cos \theta_c \\ a_2 &= (Gain + 1) - (Gain - 1) \cdot \cos \theta_c - \beta \\ b_0 &= Gain \cdot ((Gain + 1) + (Gain - 1) \cdot \cos \theta_c + \beta) \\ b_1 &= -2 \cdot Gain \cdot ((Gain - 1) - (Gain + 1) \cdot \cos \theta_c) \\ b_2 &= Gain \cdot ((Gain + 1) + (Gain - 1) \cdot \cos \theta_c - \beta) \end{aligned} \quad (1.17)$$

## Peak filtre

Vstupné parametre pre peak filter sú:

- $F_c$ : stredová frekvencia filtru, frekvencia na ktorej je maximálny zisk filtru,
- $F_s$ : vzorkovacia frekvencia,
- $Q$ : kvalita filtru,
- $Gain_{dB}$ : zisk filtru pre  $f = F_c$ .

Normalizovaný zisk vypočítame podľa vzťahu 1.18

$$Gain = e^{(Gain_{dB} \cdot 0.115129254)}, \quad (1.18)$$

Výpočet koeficientov sa líši pre negatívny a pozitívny zisk.

Pre negatívny zisk (*cut*) je výpočet nasledovný:

$$CutValue = 1 + K \cdot \left( \frac{Q}{Gain} \right) + W, \quad (1.19)$$

kde

$$K = \tan(\theta_c/2), \quad (1.20)$$

a

$$W = K^2, \quad (1.21)$$

Parameter  $\theta_c$  vypočítame podľa rovnice 1.15.

Hodnoty koeficientov sa dajú získať pomocou vzťahov 1.22.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 4 \cdot \frac{(W - 1)}{CutValue} \\ a_2 &= \frac{1 - \frac{Q}{Gain} + W}{CutValue} \\ b_0 &= \frac{1 + \frac{K}{Q} + W}{CutValue} \\ b_1 &= 2 \cdot \frac{W - 1}{CutValue} \\ b_2 &= \frac{1 - \frac{K}{Q} + W}{CutValue} \end{aligned} \quad (1.22)$$

Pre pozitívny zisk (*boost*) je výpočet nasledovný:

$$BoostValue = 1 + \frac{K}{Q} + W, \quad (1.23)$$

kde  $K$  a  $W$  vypočítame podľa vzťahov 1.20 a 1.21.

Hodnoty koeficientov môžeme vypočítať pomocou vzťahov 1.24.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= \frac{1 + K \cdot \frac{Gain}{Q} + W}{BoostValue} \\ a_2 &= \frac{1 - \frac{K}{Q} + W}{BoostValue} \\ b_0 &= 2 \cdot \frac{1 + K \cdot \frac{Gain}{Q} + W}{BoostValue} \\ b_1 &= 2 \cdot \frac{W - 1}{BoostValue} \\ b_2 &= \frac{1 - K \cdot \frac{Gain}{Q} + W}{BoostValue} \end{aligned} \quad (1.24)$$

Vyššie uvedené vzťahy boli prebrané z aplikačných poznámok firmy STMicroelectronics *IIR filter design equations for Sound Terminal* [19].

## 1.4 Fourierova transformácia

Fourierova analýza je rodina matematických techník, založených na rozklade signálu na bazové signály (sínusoidy o rozličných frekvenciách). Poznáme štyri členy tejto rodiny:

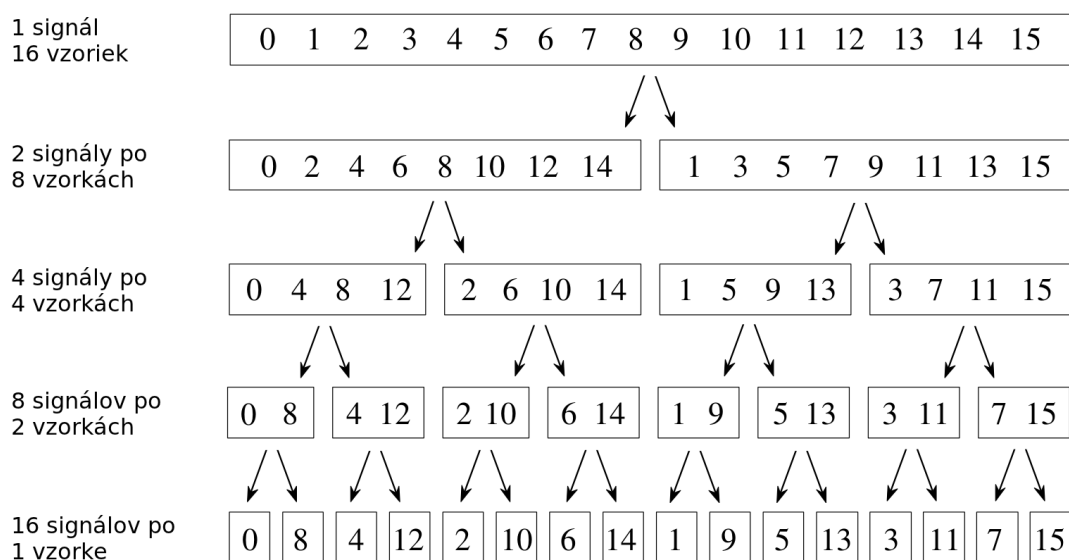
- Fourierova transformácia – pre aperiodické spojité signály,
- Fourierove rady – pre periodické spojité signály,
- DTFT (Discrete Time Fourier Transform) – pre aperiodické diskrétné signály,
- DFT (Discrete Fourier Transform) – pre periodické diskrétné signály.

Diskrétna Fourierova transformácia je jediná z transformácií, s ktorou vie pracovať číslicová technika.

DFT sa dá počítať viacerými spôsobmi, napr. riešením  $N$  lineárnych rovníc, korelačnou metódou alebo pomocou rýchlej Fourierovej transformácie. Rýchla Fourierova transformácia (FFT – Fast Fourier Transform) je oveľa rýchlejšia ako ostatné metódy a preto veľmi vhodná na implementáciu vo výpočtovej technike. V ďalšom texte si stručne priblížime spôsob jej fungovania.

### 1.4.1 Rýchla Fourierova transformácia

Prvým krokom FFT je dekompozícia signálu s  $N$  vzorkami na  $N$  signálov s jednou vzorkou. To je naznačené na obr. 1.12. V každom kroku je signál rozdelený na párne a nepárne číslované vzorky. Počet krokov potrebných na kompletnú dekompozíciu signálu je rovný  $\log_2 N$ , kde  $N$  je počet vzoriek.



Obr. 1.12: FFT dekompozícia

V dvojkovej sústave sa dekompozícia signálu dosiahne jednoduchou bitovou inverziou (na prvom mieste bude posledná číslica, na druhom predposledná, atď.). Napríklad, na mieste druhej vzorky podľa obr. 1.12 (hodnota  $1 = 0b0001$ ) bude po dekompozícii  $8 = 0b1000$  a na mieste vzorky s hodnotou  $10 = 0b1010$  je po dekompozícii  $5 = 0b0101$ .

V druhom kroku sa určí frekvenčné spektrum  $N$  jednovzorkových signálov. To je vcelku jednoduché, pretože frekvenčné spektrum jednobodového signálu je rovné samo sebe. Od tohto bodu už nepracujeme so signálom v časovej oblasti, ale so signálom v oblasti frekvenčnej.

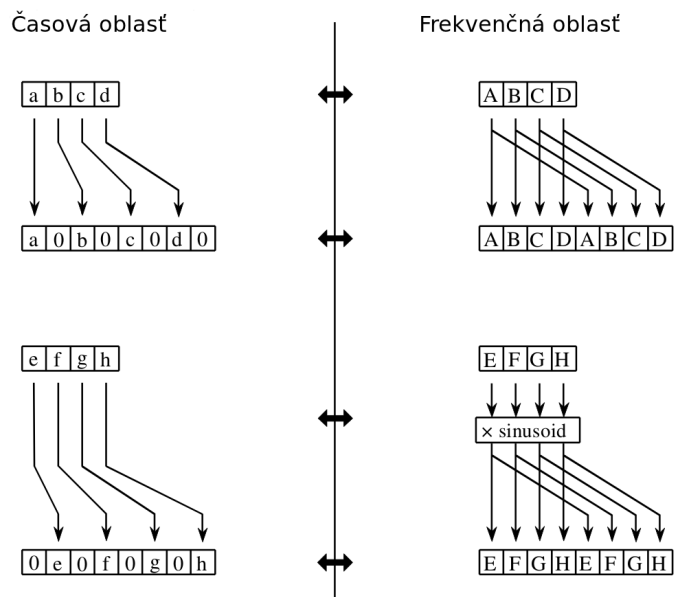
V poslednom kroku je potrebné  $N$  jednovzorkových signálov zložiť späť do jedného signálu o  $N$  bodoch. Keďže už pracujeme so signálom vo frekvenčnej oblasti, nemôžeme použiť postup, ktorý sme aplikovali na signál v časovej oblasti. V prvom kroku syntetizujeme 16 frekvenčných spektier (každý s veľkosťou jednej vzorky) do 8 frekvenčných spektier (každé s veľkosťou 2 vzorky). V druhom kroku spojíme 8 frekvenčných spektier do 4 frekvenčných spektier, a tak ďalej. V poslednom kroku získame jedno 16 vzorkové frekvenčné spektrum, ktoré je výstupom rýchlej Fourierovej transformácie vstupného 16 vzorkového signálu z časovej oblasti.

V procese syntézy frekvenčných spektier musíme postupovať proti zmenám, ktoré do frekvenčného spektra zaviedol rozklad signálu v časovej oblasti. Na obr. 1.13 je naznačené, čo sa deje so signálom vo frekvenčnej oblasti, keď ho meníme v časovej oblasti. Pri prekladaní signálu nulami na dvojnásobnú veľkosť sa spektrum signálu duplikuje. Posunutie signálu v časovej oblasti (konvolúcia s posunutou delta funkciou) je ekvivalentné násobeniu spektra signálu sínusoidou. Preto, keď chceme spojiť vzorky  $abcd$  a  $efgh$  do signálu  $aebfcgdh$  (postupujeme proti smeru rozkladu signálu 0 2 4 6 8 10 12 14 na signály 0 4 8 12 a 2 6 10 14 z obr. 1.12), musíme spektrum vzoriek  $abcd$  duplikovať a spektrum vzoriek  $efgh$  vynásobiť sínusoidou a duplikovať.

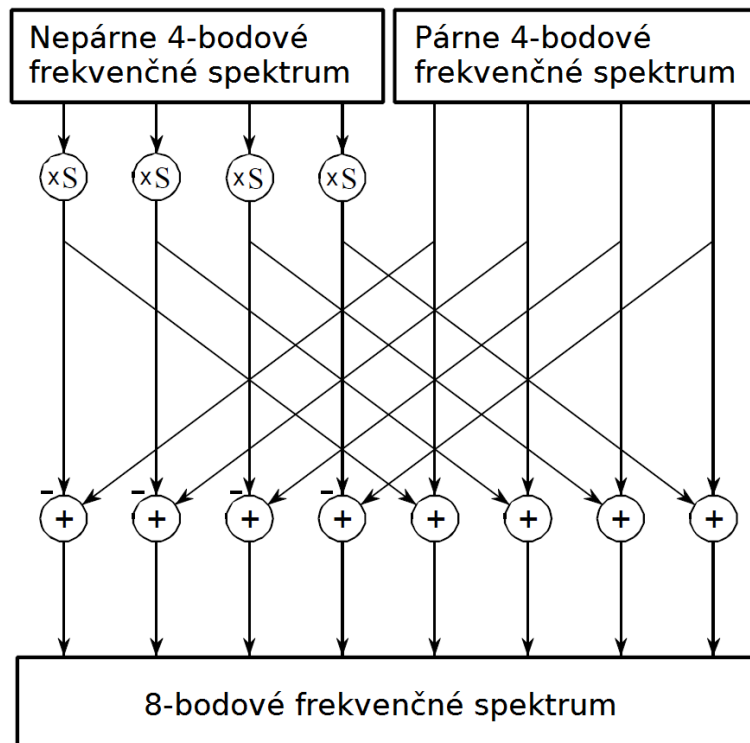
V ďalšom kroku je potrebné obidve spektrá spojiť dohromady. Táto operácia je zachytená na obr. 1.14. Vidíme, že pri počítaní prvých štyroch vzoriek výsledného signálu sú vzorky nepárnej časti spektra ( $EFGH$ ) násobené  $-1$ . To je dané jednou vlastnosťou Fourierovej transformácie – pre reálny vstupný signál s párnym počtom prvkov je spektrum symetrické a komplexne združené okolo svojho stredu. Čiže jedna polovica spektra má zápornú imaginárnu časť.

Všimnime si, že diagram FFT syntézy (obr. 1.14) je zložený zo základného obrazca, tzv. motýlika. Motýlik je základný výpočtový prvok FFT, ktorý transformuje dva komplexné vzorky na iné, požadovaným spôsobom upravené komplexné vzorky (obr. 1.15).

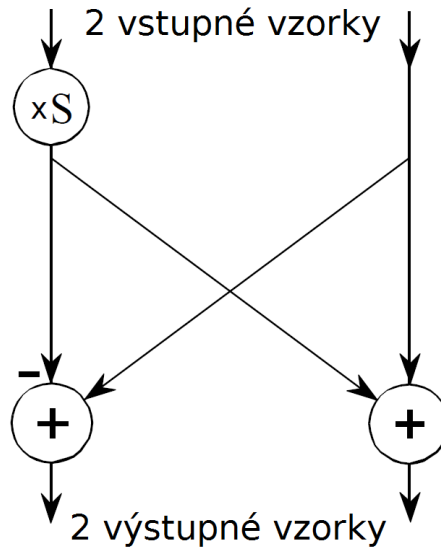
Keď je DFT počítaná pomocou korelácie, je rýchlosť výpočtu závislá na druhej mocnine počtu vzoriek ( $N^2$ ). Naproti tomu FFT závisí na hodnote danej výrazom  $N \log_2 N$ . 1024 bodová FFT je asi 300 krát rýchlejšia, ako 1024 bodová DFT reali-



Obr. 1.13: FFT syntéza



Obr. 1.14: Diagram FFT syntézy



Obr. 1.15: Motýlik – základný prvok FFT

zovaná pomocou korelácie. Nie len, že je hodnota výrazu  $N \log_2 N$  menšia ako  $N^2$ , ide aj o to, že rastie s menšou strmostou. Napríklad 32 bodová FFT je asi 10 krát rýchlejšia a 4096 bodová je už 1000 krát rýchlejšia ako rovnako dlhá DFT počítaná pomocou korelácie. [1]

## 1.5 Dátová komunikácia

Medzi externými reproduktormi a SPU bude potrebné vytvoriť komunikačné rozhranie, slúžiace ako na digitálny prenos hudby, tak aj na dátovú komunikáciu s reproduktormi a s počítačom.

Na prenos audio signálu v nekomprimovanej podobe je potrebné digitálne rozhranie s dostatočne vysokou rýchlosťou prenosu. Pri vzorkovacej frekvencii 48 kHz a použitia čísel s plávajúcou desatinnou čiarkou (float) je bitová rýchlosť pre jeden kanál takmer 1,6 Mb/s. Kvôli dostatočnej priepustnosti a veľkému rozšíreniu je na prenos audia použitá sieť 100 Mb ethernetu. Sieť sa využíva takisto na nastavovanie systému a logovanie servisných informácií.

K sieťovým protokolom by mohlo byť alternatívou využitie FireWire alebo USB. Prenosovými rýchlosťami predčia rýchly Ethernet (FireWire 400 a vyšší a USB 2.0 a vyššie). Problém pri využití týchto protokolov by bola maximálna dĺžka káblového spoja, obmedzená na 4.5 m pre FireWire a 5 m pre USB.

### 1.5.1 Fast Ethernet

Fast Ethernet implementovaný na krútenej dvojlinke je označovaný ako 100BASE-TX. Jedná sa o najrozšírenejšiu variantu rýchleho Ethernetu. Okrem krútenej dvojlinky je možné použiť ako prenosové médium aj optické vlákno (varianty označované ako 100BASE-FX, 100BASE-SX, 100BASE-BX a 100BASE-LX10).

Maximálna teoretická prenosová rýchlosť rýchleho Ethernetu je 100 Mb/s. V skutočnosti je táto rýchlosť kvôli rézii prenosu menšia.

Ethernet obsahuje fyzickú a linkovú vrstvu. V linkovej vrstve sa nachádza vrstva označovaná MAC (Medium Access Control), ktorá riadi prístup k zdieľanému médiu. MAC vrstva komunikuje s vrstvou PHY (Physical Layer Interface), ktorá zaisťuje kódovanie, obnovu taktu, kompenzáciu presluchovej, apod. MAC môže byť prepojená s PHY 4 bitovým 25 MHz synchronným paralelným rozhraním, známym ako MII (Media Independent Interface), alebo 2 bitovým 50 MHz variantom RMII (Reduced Media Independent Interface). Pre MII je definovaný 40 pinový konektor, ale v praxi je rozhranie využívané iba ako štandard rozhrania obvodov na doske plošných spojov. Často je úplne skryté vo vnútri integrovaného obvodu. [7] [4]

V roku 1995 bola schválená špecifikácia pre rýchly Ethernet označovaná ako IEEE 802.3u. Rýchly Ethernet je založený na efektívnejšom využití prenosového média oproti klasickému Ethernetu. Miesto kódovania Manchester je použité kódovanie 4B5B, doplnené viacúrovňovým kódovaním pre prenos po metalických vedeniach (MLT-3 – Multi Level Transmit). Vzdialenosť medzi dvoma jednotkami prepojenými metalickým káblom je do 100 m. Pri použití optického vlákna sa vzdialenosť predĺži na 412 m pri poloduplexnom, a na 2000 m pri plne duplexnom spojení. [4]

Štruktúra ethernetového rámca sa pre 10 Mb/s až 1000 Mb/s Ethernet nemení.

Preambula	SFD	Cieľ	Zdroj	Dĺžka/typ	Dáta	FCS
7 B	1 B	6 B	6 B	1 B	46 B - 1500 B	4 B

Obr. 1.16: Štruktúra ethernetového rámca

Podľa obr. 1.16 sa rámec delí na:

1. *Preambula* – obsahuje vzorku striedajúcich sa núl a jednotiek. Táto vzorka sa používa na časovú synchronizáciu pomalších implementácií Ethernetu (do 10 Mb/s). Pre rýchlejšie implementácie Ethernetu, ktoré sú synchronizované iným spôsobom, je táto informácia zbytočná, avšak ostáva zachovaná kvôli kompatibilite.
2. *SFD* – Takzvaný SOF (Start Of Frame) delimiter je jednobajtové pole, ktoré vymedzuje koniec úseku časovej synchronizácie. Obsahuje sekvenciu bitov 101011.

3. *Cieľová adresa* – obsahuje MAC adresu príjemcu. Cieľová MAC adresa môže byť typu UNICAST, MULTICAST alebo BROADCAST pre adresáciu viacerých príjemcov naraz a pre skupinové vysielanie.
4. *Zdrojová adresa* – obsahuje unicastovú MAC adresu zariadenia, ktoré daný rámec vyslalo.
5. *Dĺžka/typ* – má dva rôzne účely. Ak je táto hodnota menšia ako 0x600, tak určuje dĺžku rámca. Hodnota *typ* určuje ktorý protokol vyššej vrstvy je daným ethernetovým rámcom prenášaný.
6. *Dáta* – pole obsahuje samotné užitočné dáta, ktoré sú týmto rámcom prenášané. Obvykle sa jedná o paket vyššej vrstvy. Maximálna možná veľkosť prenášaného rámca je 1500 bajtov. Minimálna veľkosť pola dát je 46 bajtov. V prípade, že chceme prenášať menší objem dát, doplní sa balík automaticky na veľkosť 46 bajtov.
7. *FCS (Frame Check Sequence)* – obsahuje 32 bitový kontrolný súčet, ktorý je hardvérovo spočítaný zariadením, ktoré dané dáta odosiela. Príjemca po prijatí dát spočíta súčet znovu pomocou toho istého algoritmu. V prípade že sa súčty nezhodujú, prijaté dáta sú vyhodnotené ako poškodené. [10]



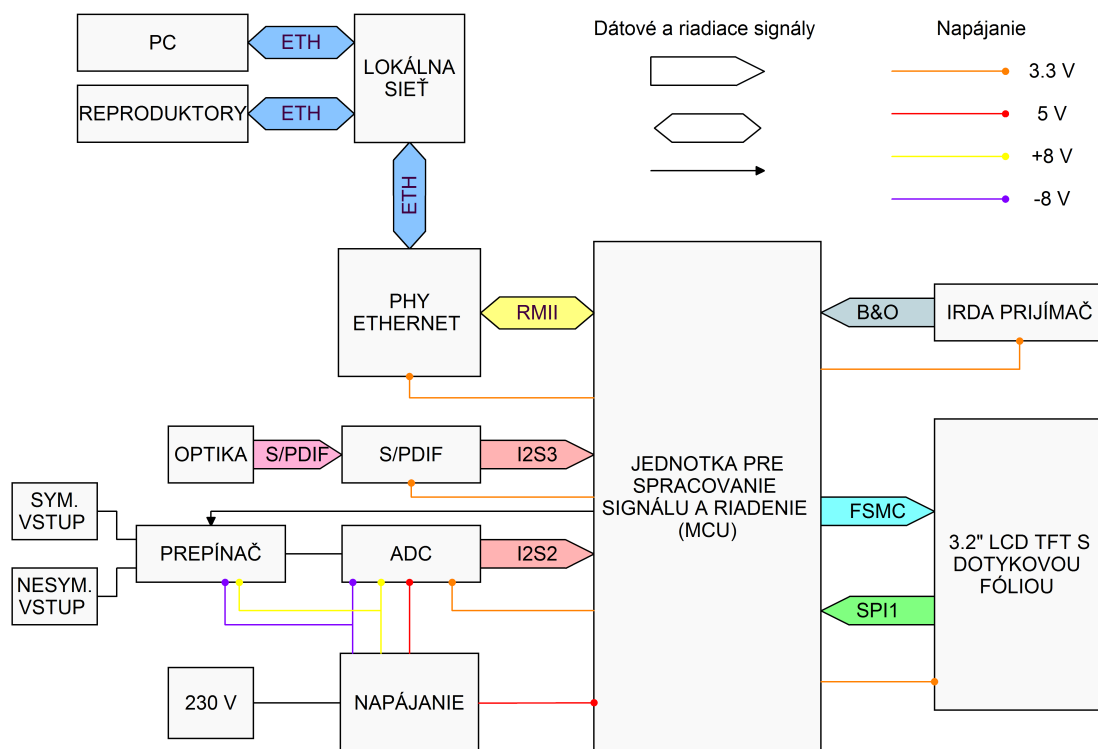
## 2 HARDWARE SPU

V tejto časti práce sa budeme zaoberať návrhom hardwaru SPU jednotky pozostávajúcej z nasledujúcich častí:

- MCU jednotka,
- vstup zvukového signálu:
  - analógový,
  - digitálny,
- dátový vstup a výstup.

### 2.1 Bloková schéma zariadenia

Na obr. 2.1 je znázornená bloková schéma jednotky. Tá sa skladá zo vstupných a výstupných blokov a bloku pre riadenie a spracovanie signálu.



Obr. 2.1: Bloková schéma hardwaru SPU

Centrom SPU je blok označený ako MCU (mikrokontrolérová jednotka). Blok je realizovaný vývojovým kitom STM32F4Discovery od firmy STMicroelectronics, ktorý je osadený mikrokontrolérom STM32F407VG s jadrom ARM Cortex M4F [11]. Tento procesor sa stará o riadenie, smerovanie vstupov a spracovanie audia.

Jednotka má tri vstupy zvukového signálu: symetrický a nesymetrický analógový vstup a digitálny optický vstup. Analógové vstupy zdieľajú spoločný AD prevodník. Z toho dôvodu je použitý prepínač (konkrétne CD4052B [20] od Texas Instruments) na aktivovanie iba jedného vstupu.

Za vstupmi sa nachádzajú prevodníky s výstupom v štandarde I<sup>2</sup>S (Inter-IC Sound).

Napájanie je realizované 10 VA transformátorom s 2 x 9 V výstupom a stabilizátormi na +8 V, -8 V a +5 V. Na získanie napätia 3.3 V je využitý stabilizátor osadený na kitu STM32F4Discovery.

Digitálne dátové rozhranie je realizované pomocou ethernetu. Použitý mikrokontrolér obsahuje rozhranie na úrovni MAC vrstvy a preto je ešte doplnený blok PHY ETHERNET s čipom realizujúcim fyzickú vrstvu sieťového rozhrania.

Na ovládanie zariadenia je možné využiť infračervené rozhranie alebo software v PC. IrDA rozhranie komunikuje s riadiacou jednotkou pomocou interného protokolu firmy Bang & Olufsen.

Jednotku ja ďalej možno softwarovo rozšíriť o ovládanie pomocou dotykového TFT displeja.

### 2.1.1 MCU

Ako MCU jednotka je s výhodou použitý vývojový kit od firmy STMicroelectronics STM32F4Discovery. Jeho cena neprevyšuje cenu samostatného procesoru STM32F4V07VG a odpadá nutnosť kupovať externý programátor. Programovanie mikrokontroléra prebieha cez mini USB.

Čip je založený na architektúre ARM Cortex M4F s maximálnou hodinovou frekvenciou jadra 168 MHz. F v názve Cortex M4F znamená, že jadro obsahuje jednotku pre hardwarovú podporu počítania s číslami s pohyblivou desatinnou čiarkou (FPU – Floating Point Unit). Najpoužívanejšiu DSP operáciu MAC (Multiply and Accumulate) zvláda vykonať za jeden hodinový cyklus. V mikrokontroléri sa nachádza 1 MB flash pamäte a 192 kB SRAM. [8]

STM32F4Discovery je osadený procesorom v púzdre LQFP100 a väčšina pinov je vyvedená na užívateľsky prístupné vývody. Z príslušenstva dostupného na kitu je okrem mikroprocesora využívaný stabilizátor na 3.3 V a programátor.

Mikrokontrolérová jednotka sa využíva na spracovanie audia, zostavovanie UDP datagramov určených pre externé reproduktory a TCP paketov pre komunikáciu s PC, riadenie z IrDA prijímača a zobrazovanie na displeji.

### 2.1.2 IrDA prijímač

IrDA prijímač a komunikačné rozhranie je duševným vlastníctvom firmy Bang & Olufsen a preto tu nie je možné uviesť podrobnosti o spôsobe jeho funkcie.

### 2.1.3 LCD

Pre ďalšie verzie systému bola jednotka doplnená o dotykový TFT 3.2" LCD displej s rozlíšením 320 x 240 pixelov. Pomocou neho bude možné nastavovať a monitorovať systém, podobne ako je to realizované pomocou ovládacieho software z PC.

Displej je pripojený k MCU pomocou FSMC (Flexible Static Memory Controller) rozhrania, čo je v podstate rozhranie na komunikáciu so synchronnými a asynchronnými pamäťami typu SRAM, ROM, FLASH a PSRAM. Driver displeja, SSD1289, je s týmto rozhraním plne kompatibilný [21].

Na displeji je odporová dotyková fólia, na ktorú je pripojený AD prevodník ADS7843 [22] s SPI rozhraním, pomocou ktorého je prenášaná informácia o mieste dotyku do MCU jednotky na ďalšie spracovanie.

## 2.2 Vstup audio signálu

SPU jednotka má vstupy pre analógový zvukový signál a digitálny zvukový signál. Na príjem analógového signálu slúžia 2 symetrické a 2 nesymetrické vstupy a na príjem digitálneho signálu optický S-PDIF konektor. V nasledujúcich dvoch sekciách bude popísané zapojenie analógovej a digitálnej časti.

### 2.2.1 (Semi-)symetrický analógový vstup

Pre digitalizovanie analógového zvukového signálu je použitý AD prevodník *CS5361* od firmy Cirrus Logic, Inc. Medzi jeho hlavné vlastnosti patrí:

- Delta-Sigma modulátor 5. radu,
- 24 bitová konverzia,
- nastaviteľná vzorkovacia frekvencia až do 192 kHz,
- 114 dB dynamický rozsah,
- antialiasingový filter,
- diferenčné vstupy,
- podporuje master aj slave mód,
- 5 V napájanie analógovej časti,
- 3.3 V - 5 V napájanie digitálnej časti.

Prevodník je používaný so vzorkovacou frekvenciou 48 kHz s 24 bitovou hĺbkou vzoriek. Hlavný hodinový signál, ktorý je 256 násobkom vzorkovacej frekvencie (čiže 12.288 MHz), generuje mikrokontrolér.

### Zapojenie AD prevodníka

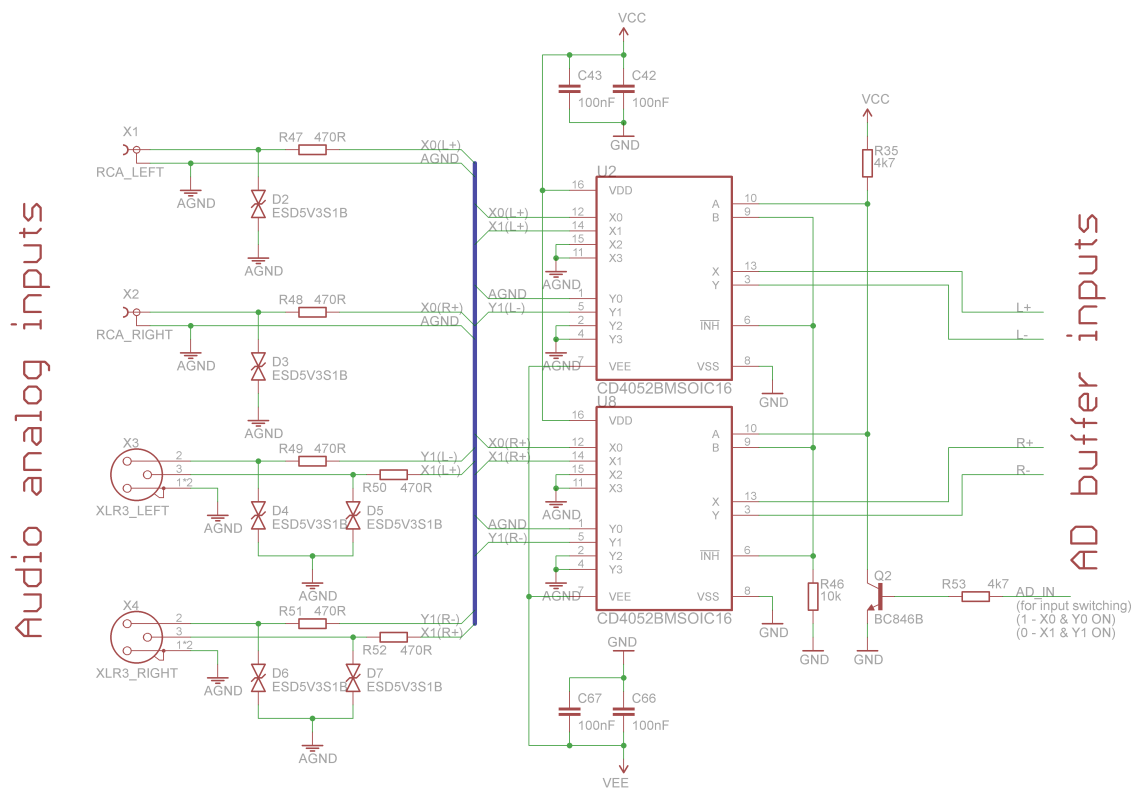
CS5361 má 2 analógové audio vstupy, jeden diferenčný pre ľavý kanál a jeden pre pravý kanál. To dovoľuje zapojiť jeden symetrický stereo vstup. Pre pripojenie nesymetrického vstupu je potrebné uzemniť zápornú časť diferenčného vstupu. Symetrický stereo vstup je realizovaný dvojicou konektorov typu *XLR3*, v schéme označených ako X3 a X4 a nesymetrický vstup dvojicou konektorov typu *RCA* (X1 a X2) (vid' obr. 2.2). Signály z analógových vstupov sú privedené do prepínača *CD4052BM* od Texas Instruments, ktorý má 8 vstupov a 2 výstupy [20]. Pre prepínanie štyroch výstupov (L+, L-, R+ a R-) budeme potrebovať dva prepínače. Transily D2 až D7 slúžia spoločne so 470  $\Omega$  rezistormi, zapojenými medzi konektormi a integrovaným obvodom, ako ochrana proti elektrostatickému výboju. Prepínanie vstupov je realizované signálom *AD\_IN*, pri 3.3 V je na pine A nulový potenciál, a na výstup prepínača sú privedené signály z nesymetrických vstupov. V prípade, že je na signálu *AD\_IN* potenciál zeme, tak sú pripojené symetrické vstupy. Výber vstupu je realizovaný mikrokontrolérom. Povoľovací vstup  $\overline{\text{INH}}$  je uzemnený, takže prepínač je permanentne v zopnutom stave.

Vybraný zvukový signál ďalej vedie do vstupného filtra, obr. 2.3. Jedná sa o plne diferenčný analógový vstupný buffer, plniaci viacero funkcií, a to filtrovanie jednosmernej zložky, izoláciu analógových vstupov, nízku výstupnú impedanciu a anti-aliasingovú filtráciu. Zapojenie je prevzaté z katalógového listu k obvodu CS5361 [9]. Šumový príspevok tohoto filtra je daný hlavne šumovou hladinou použitého operačného zosilňovača. Vybrali sme preto kvalitný operačný zosilňovač *OPA2134* od Texas Instruments, určený pre profesionálne audio zariadenia. Podľa katalógového listu nestúpne šumové napätie  $V_{\text{NOISE}}$  vo frekvenčnom rozsahu 20 Hz až 20 kHz nad 1.2  $\mu\text{V}_{\text{rms}}$ .

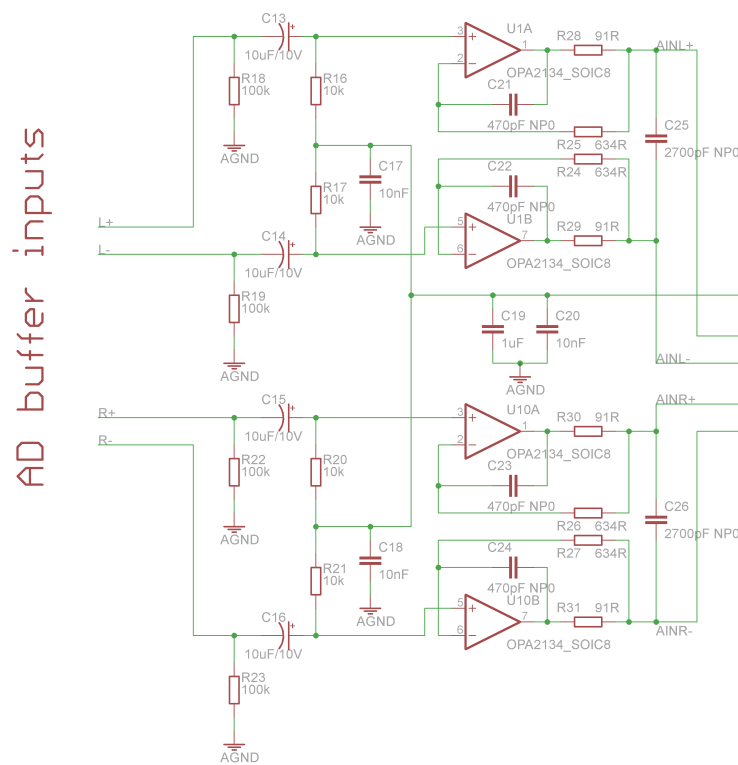
Na vstupe filtra je horná priepusť realizovaná kondenzátorom C13 a rezistorom R16. Podľa rovnice 2.1 je medzná frekvencia filtra 1.59 Hz. Je dôležité, aby bola táto hodnota aspoň jednu dekádu pod začiatkom počuteľného frekvenčného pásma.

$$f_M = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 10 \cdot 10^3 \cdot 10 \cdot 10^{-6}} = 1.59 \text{ Hz} \quad (2.1)$$

Ďalej nasleduje zapojenie s operačným zosilňovačom, ktoré zabezpečuje veľmi nízku výstupnú impedanciu, čím minimalizuje skreslenie pri vzorkovaní vnútornými vzorkovacími obvody AD prevodníka. Okrem tejto funkcie sa zapojenie chová aj

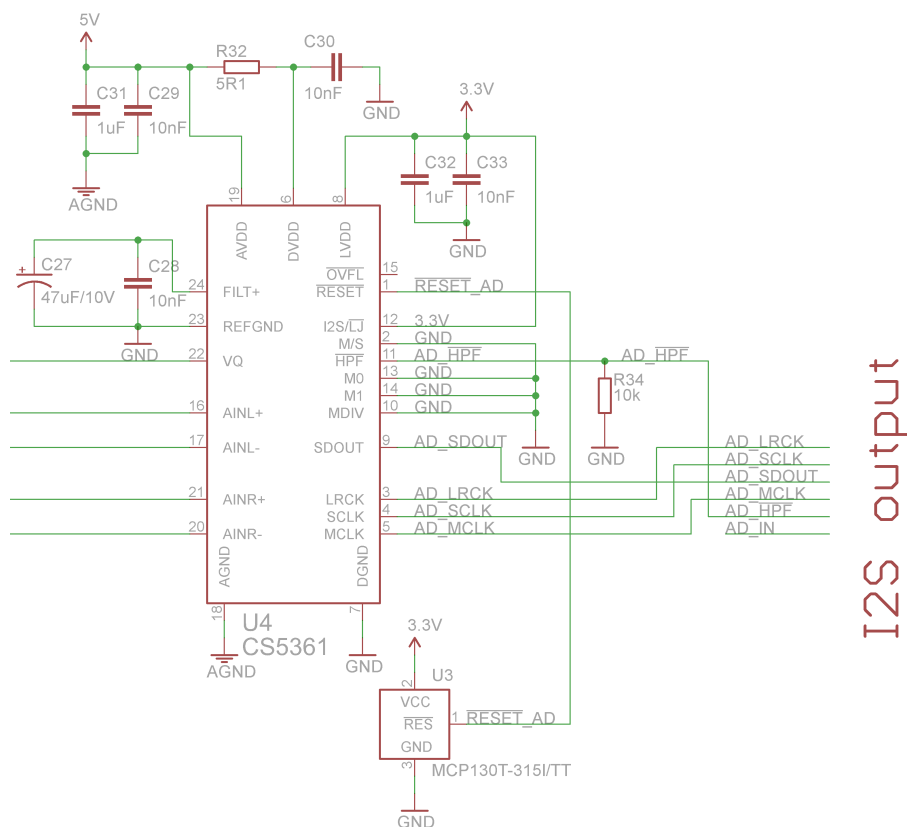


Obr. 2.2: Schéma zapojenia analógových vstupov a prepínača CD4052BM



Obr. 2.3: Schéma zapojenia vstupného filtra

ako dolná priepusť. Jej úlohou je zachytiť násobky vzorkovacej frekvencie AD prevodníka ( $n \cdot 6.144$  MHz, kde  $n = 1, 2, 3, \dots$ ), pretože na týchto frekvenciách je prevodník náchylný k aliažingu. Aby sme zachovali linearitu signálu, je nutné sa vyvarovať použitiu kondenzátorov s veľkým napäťovým koeficientom, ktorých typickým zástupcom sú keramické kondenzátory. Preto sme klasické keramické kondenzátory, ktoré ležia v ceste signálu, nahradili kondenzátormi so substrátom typu NP0<sup>1</sup>. V schéme sú označené reťazcom NP0 pri hodnote súčiastky. Viac informácií o zapojení vstupného filtra je možné nájsť v aplikačných poznámkach AN241 – *Analog Input Buffer Architectures* firmy Cirrus Logic [17].



Obr. 2.4: Schéma zapojenia obvodu CS5361

Zapojenie AD prevodníka nájdeme na obr. 2.4. CS5361 podporuje I<sup>2</sup>S a Left Justified formát výstupných dát; nastavením pinu I2S/L $\bar{J}$  sme vybrali I<sup>2</sup>S. Prevodník funguje ako slave s maximálnou vzorokovacíou frekvenciou do 50 kHz (nastavenie dvojicou M0 a M1). Keďže CS5361 nedokáže generovať vlastný hodinový signál sú hodiny generované mikrokontrolérom a privedené na pin MCLK. Pretože sme nastavili prevodník do slave módu, je potrebné k nemu priviesť aj ostatné signály (SCLK – vzorkovací kmitočet a LRCK, kde logická úroveň signalizuje aktívny kanál).

<sup>1</sup>vhodný by bol aj substrát C0G

Z prevodníka je k procesoru vyvedený ešte pin  $\overline{\text{HPF}}$ , ktorý je cez pull-down rezistor R34 pripojený na zem. Týmto pinom sa povoľuje digitálny hornopriepustný filter, ktorý odstraňuje DC offset. Podľa [9] je vhodné po ustálení filteru ( $\frac{10^5}{f_{VZ}}$  sekúnd) nastaviť  $\overline{\text{HPF}}$  do logickej 1. Tým sa uloží aktuálna hodnota DC offsetu, a ďalej sa pri jeho eliminácii bude pracovať s ňou.

Pre správny nábeh obvodu po pripojení napájania používame obvod MCP130T-315I. Tento obvod drží  $\overline{\text{RESET}}$  v nule, kým nestúpne napájacie napätie nad úroveň 3.15 V. Po dosiahnutí tejto úrovne podrží zariadenie v resete ešte asi 350 ms a potom pripojí na výstup interný pull-up rezistor. Pri poklese napájacieho napätia pod danú hranicu dôjde k zhodeniu signálu  $\overline{\text{RESET}}$  na nulu.

### 2.2.2 S/PDIF digitálny vstup

S/PDIF (Sony/Philips Digital Interconnect Format) je dátový linkový protokol, ktorý slúži na prenos audio signálu v digitálnej podobe medzi zariadeniami, a to pomocou optického alebo koaxiálneho rozhrania. Je založený na štandarde AES3, používanom v profesionálnej audio technike. Signál je vysielaný cez koaxiálny kábel s RCA konektormi alebo cez optický kábel s TOSLINK konektormi. Pomocou S/PDIF je možné prenášať 2 PCM kanály, alebo viac-kanálový komprimovaný formát, ako napríklad Dolby Digital alebo DTS. S/PDIF je štandardizovaný normou IEC 60958. [14].

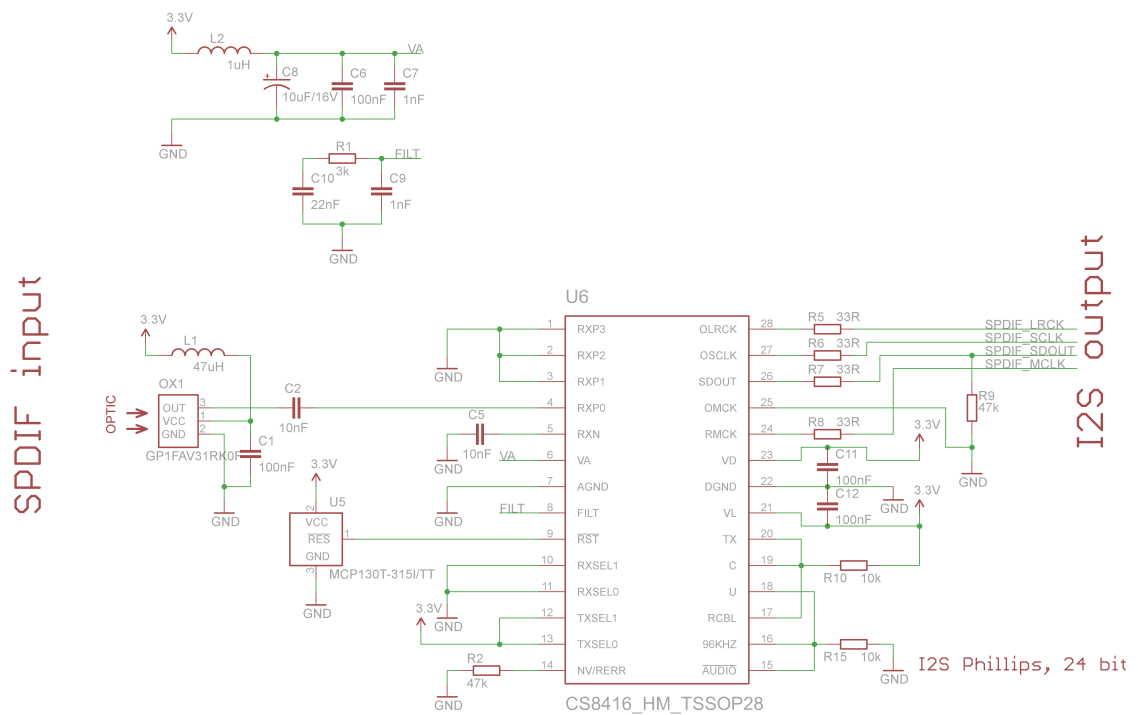
Na fyzickej vrstve je signál modulovaný BMC kódom (Biphase Manchester Code). To umožňuje získať hodinovú frekvenciu priamo z prijímaného signálu.

#### Zapojenie S/PDIF vstupu

Na prijímanie digitálneho audio signálu cez S/PDIF využijeme dekodér, ktorý prevedie vstupný signál do dátového formátu vhodného pre mikrokontrolér (konkrétne I<sup>2</sup>S). Pre prevod môžeme použiť napríklad prevodník DIR9001 od Texas Instruments [15] alebo CS8416 od firmy Cirrus Logic [16]. DIR9001 dosahuje vyššiu stabilitu hodinového signálu ako CS8416. Naproti tomu má CS8416 4 vstupy namiesto jedného a podporuje vzorkovacie frekvencie až do 192 kHz. Kvôli lepšej dostupnosti je použitý CS8416.

Na obr. 2.5 je schéma zapojenia prevodníka. CS8416 môže fungovať v dvoch režimoch: softwarovom alebo hardwarovom. Pre našu aplikáciu postačuje hardwarový režim, ktorý sa zvolí pomocou pull-down rezistoru R9 na pine SDOUT. V tomto režime pracuje prevodník plne autonómne a nepotrebuje komunikovať s mikrokontrolérom.

Optický konektor je pripojený na vstup RXP0. Výber aktívneho vstupu je realizovaný podľa tabuľky 2.1.



Obr. 2.5: Schéma zapojenia SPDIF vstupu

RXSEL0	RXSEL1	Vybratý vstup
0	0	RXP0
0	1	RXP1
1	0	RXP2
1	1	RXP3

Tab. 2.1: Výber aktívneho S/PDIF vstupu



Záporný pól symetrického vstupu RXN je podľa doporučenia z katalógového listu pripojený cez 10 nF kondenzátor na zem. Napájanie analógovej časti obvodu (pin VA) je filtrované cievkou L2 a kondenzátorom C8 a blokované kondenzátormi C6 a C7. Na tomto pinu musí byť čo najmenší šum, pretože ten priamo ovplyvňuje stabilitu hodinového signálu, získaného zo vstupného signálu. Na pin FILT sú pripojené súčiastky potrebné pre správnu funkciu PLL filtru. Podľa katalógového listu by malo byť pri použití týchto hodnôt možné detekovať signál so vzorkovacou frekvenciou 32 kHz – 192 kHz. Hodinový signál a ostatné signály sú odvodené od frekvencie S/PDIF signálu.

Po pripojení napájania sa chvíľu drží prevodník v resete vďaka obvodu MCP-130T-315I, ktorý pripojí na  $\overline{\text{RESET}}$  logickú 1, po uplynutí 350 ms po ustálení napájania nad hranicou 3.15 V.

Výstup je pomocou vstupov C a  $\overline{\text{AUDIO}}$  nastavený na 24 bitový I<sup>2</sup>S formát a vstupom U je nastavený hodinový výstup MCK na 256 násobok vzorkovacej frekvencie. Pomocou pinu TX sa nastavuje PDUR (Phase Detector Update Rate) bit v kontrolnom registri 0. Mení sa ním typ fázového detektoru používaného na zachytenie aktívneho vstupu. Pokiaľ je nastavený, tak je možné prijímať signál s maximálnym vzorkovacím kmitočtom len 108 kHz, výsledkom je ale vyššia stabilita kmitočtu odvodeného hodinového signálu.

Pomocou pinu RCBL nastavíme obvod do master módu.

## 2.3 Dátový vstup a výstup

Mikrokontrolér STM32F407 obsahuje ethernetovú perifériu, ktorá mu umožňuje prijímať a posilať dáta cez Ethernet v súlade s normou IEEE 802.3-2002. Periféria podporuje dve štandardné rozhrania s externou fyzickou vrstvou, a to MII (Media Independent Interface) a RMII (Reduced Media Independent Interface). Keď sa pozrieme na rozloženie alternatívnych funkcií jednotlivých pinov mikrokontroléra, zistíme že rozhranie MII sa prekrýva s I2S3 perifériou a znemožňuje jej použitie. Pretože I2S3 potrebujeme pre pripojenie S/PDIF vstupu, je riešením využiť rozhranie RMII, u ktorého k tejto kolízii nedochádza. [11]

V ďalšom texte je popísané zapojenie rozhrania fyzickej vrstvy Ethernetu. Pri návrhu sme siahli k často používanému čipu od Texas Instruments *DP83848*. Čip podporuje 10BASE-T a 100BASE-TX ethernetové protokoly a takisto aj obidve rozhrania MII a RMII.



príjem alebo vysielanie dát. Zelená LED pripojená na LED\_LINK zase indikuje úspešné vytvorenie spojenia. Pin LED\_SPEED (indikujúci rýchlosť pripojenia, 10 Mb/s alebo 100 Mb/s) nie je v tomto zapojení použitý. Všetky piny na ovládanie LED diód obsahujú interné zdvíhacie rezistory, takže v kludovom stave LED diódy nesvietia.

Použitie pinu 41 (RX\_ER – Receive Error) je dobrovoľné, pretože poškodené dáta sa zahadzujú na úrovni fyzickej vrstvy. COL (pin 42) - Collision Detect nie je u RMII potrebný. Kolízia sa určí s pomocou CRS (Carrier Sense, určené z CRS\_DV - Recieve Data Valid) a TX\_EN.

## 3 FIRMWARE SPU

V tejto kapitole sa budeme venovať softwarovému riešeniu SPU jednotky. V prvej sekcii vyberieme vývojové prostredie a systém reálneho času a v druhej sekcii popíšeme vytvorený firmware.

### 3.1 Vývojové prostredie a RTOS

Pre programovanie mikrokontroléra STM32F407VG je na výber z viacerých vývojových prostredí. Na stránkach STMicroelectronics ([12]) nájdeme zoznam vývojových prostredí podporujúcich rodinu procesorov STM32. Z nich môžeme uviesť vlastnosti niektorých vyskúšaných prostredí:

- *Atollic TrueSTUDIO*: vývojové prostredie založené na Eclipse. Priamo podporuje STM32F4Discovery. Jeho výhodou je v jednoduchom nastavení projektu a pohodlnom ladení programu (debugging). Voľná verzia, ktorú Atollic poskytuje, je ale obmedzená na maximálne 32 kB kódu a nemá sprístupnené všetky možnosti v porovnaní s profesionálnou verzou.
- *CrossWorks for ARM*: jedná sa o vývojové prostredie od firmy Rowley Associates. Priamo podporuje STM32F4Discovery. Jeho hlavnou výhodou je, že je multiplatformný, čiže je možné ho využívať vo Windowsoch, Mac OS X a rôznych distribúciach Linuxu. Ďalšia výhoda je veľmi svižné prostredie a rýchly debugger. Bohužiaľ neexistuje free verzia.
- *CooCox CoIDE*: jediné vývojové prostredie, ktoré je dostupné vo free verzii. Je založené na odľahčenej a orezanej verzii Eclipse a plne podporuje STM32F4Discovery kit. Na vývoj jednoduchých aplikácií a pri využívaní knihozien od ST je veľmi vhodné. Bohužiaľ neumožňuje vytvárať projekty z makefile súboru, čo bráni jednoduchému používaniu použitého RTOS. Preto bude software písaný v nasledovnom IDE.
- *Eclipse+GCC ARM+OpenOCD*: táto možnosť popisuje ako získať multiplatformné IDE, ktoré je úplne zadarmo a je plne postačujúce pre vývoj projektu. Nevýhodou je problematické nastavenie. IDE je potrebné poskladať z viacerých komponentov. Jedná sa o Eclipse s doinštalovanými doplnkami (podpora C/C++, hardwarový debugger), kompilátor (gcc pre ARM s podporou CortexM4 a hard floating point) a OpenOCD (programátor a debugger s podporou STlink/v2).

Pre takto veľký projekt je výhodné použiť niektorý z voľne dostupných systémov reálneho času (RTOS – Real Time Operating System). Asi najznámejší RTOS pre embedded systémy, freeRTOS, nepodporuje vo verzii pre CortexM4F kompilátor GCC. Možnosťou by bolo využiť port z CortexM3. Tu by sme však nemohli využívať

FPU jednotku. Aby sme sa vyhli nutnosti dopisovať do freeRTOS kód pre prácu s FPU, rozhodli sme sa použiť iný RTOS. Jedná sa o RTOS s označením *ChibiOS/RT* [13]. Jeho základné vlastnosti sú:

- efektívne a prenositeľné preemptívne jadro,
- výborný výkon pri prepínaní kontextov,
- statická architektúra (všetko je staticky alokované pri kompilácii),
- voliteľná podpora dynamických rozšírení, dynamických objektov,
- podporuje vlákna, virtuálne časovače, semaforey, mutexy, podmienené premenené, správy, mailboxy, príznaky udalostí, fronty, atď.,
- HAL vrstva, podporujúca veľké množstvo periférií, ako napr. Porty, Sériové rozhranie, ADC, CAN, EXT, GPT, I2C, ICU, MAC, PWM, RTC, SDC, SPI, UART, USB,
- podpora externých súčastí, napr. LWIP, FatFs, rozhranie pre grafické displeje.

ChibiOS/RT je pod neustálym aktívnym vývojom, na stránkach nájdeme dobrú dokumentáciu, rôzne návody a príklady. K STM32F4Discovery existuje demo aplikácia, kde je overená funkčnosť niektorých periférií (ADC, SPI, PWM) a názorne ukázaný spôsob práce so systémom. Na stránkach ChibiOS/RT sa takisto nachádza návod ako zostaviť vývojové prostredie.

Nevýhodou je, že v HAL vrstve chýba podpora pre I<sup>2</sup>S a FSMC periférie. Pri tvorbe firmwaru musela byť podpora týchto periférií doplnená svojpomocne.

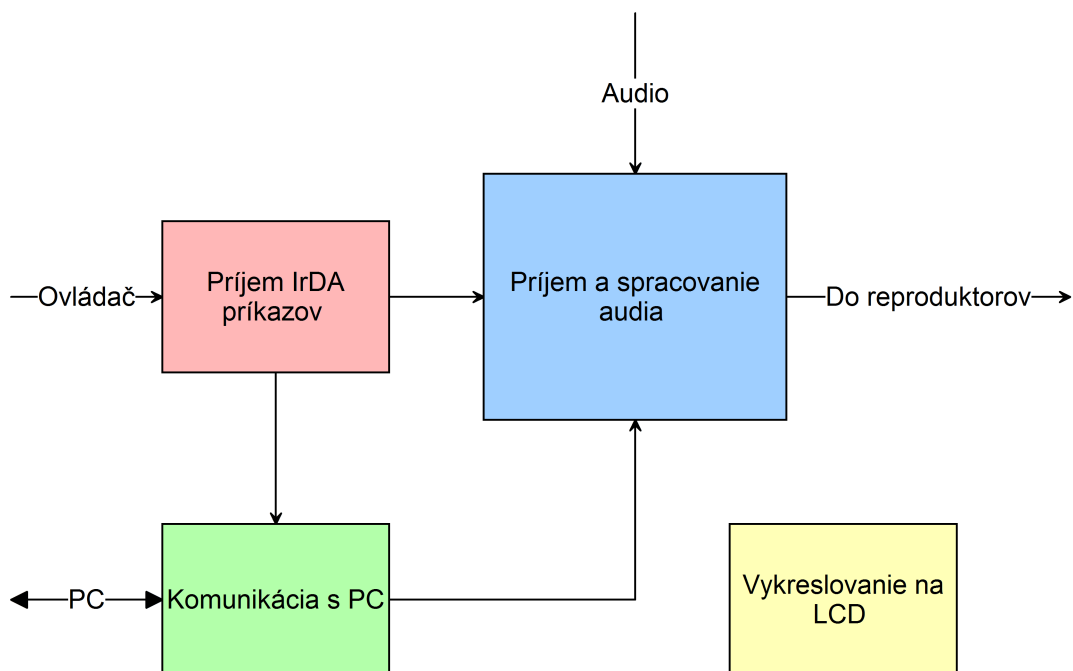
## 3.2 Štruktúra firmwaru

Firmware je rozdelený do logických celkov, ktoré si medzi sebou v prípade potreby predávajú informácie. Na obr. 3.1 je znázornená štruktúra firmwaru v zjednodušenej podobe formou navzájom previazaných blokov.

Hlavný blok pre príjem a spracovanie zvukového signálu je na obr. 3.1 znázornený modrou farbou. Pozostáva z dvoch vláken, jedno pre spracovanie signálu z S/PDIF prevodníka a druhé pre spracovanie signálu z AD prevodníka.

Blok pre príjem a spracovanie audia je riadený z infračerveného ovládača alebo pomocou softwaru v PC. V PC sa zobrazujú aj servisné informácie a aktuálne nastavenia (napríklad pri zmene hlasitosti pomocou IrDA ovládania, sa automaticky aktualizuje ukazovateľ hlasitosti v PC). O komunikáciu s PC sa stará vlákno nazvané *pcCommThread*.

Audio dáta sa posielajú pomocou UDP datagramov. Na začiatku každého paketu je 16 bajtová hlavička, v ktorej sa prenáša identifikačné číslo input streamu, output streamu, sériové číslo slave jednotky, pre ktorú je paket určený a časová známka. Viac informácií o štruktúre UDP paketu je v kapitole 4 *Komunikačný protokol*, na



Obr. 3.1: Bloková schéma firmwaru SPU

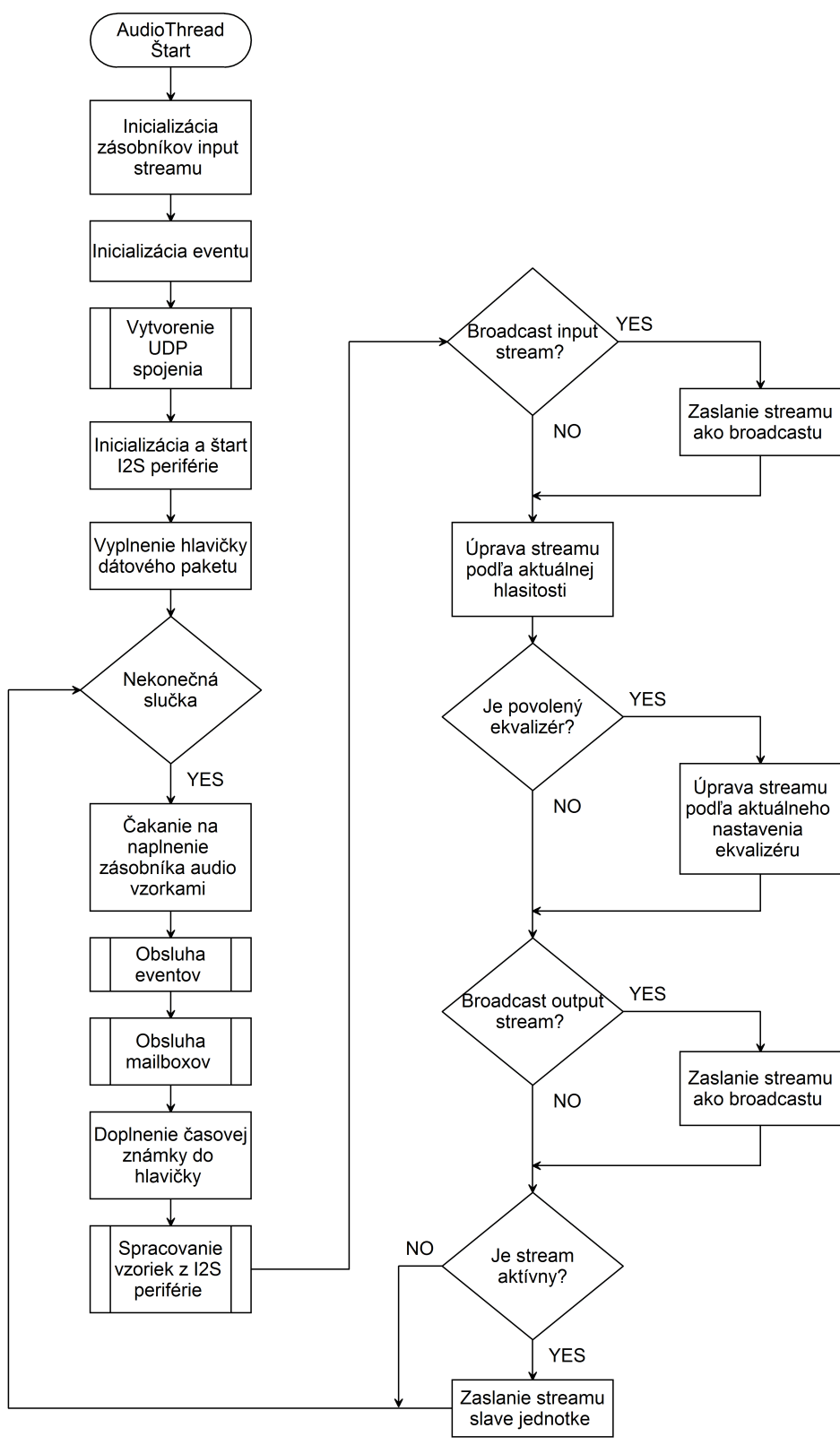
strane 56.

### 3.2.1 Audio vlákna

Na obr. 3.2 je znázornený obecný vývojový diagram pre obidve vlákna spracovávajúce audio.

V procese *Vytvorenie UDP spojenia* sa vlákno uspí, pokiaľ nie je input stream pripojený k nejakej slave jednotke. V nekonečnej slučke sa čaká na naplnenie zásobníka zo vzorkami. Zásobník má 1360 bajtov, 16 bajtov zaberá hlavička, takže na vzorky ostáva 1344 bajtov. Keďže sú vzorky spracovávané a odosielané vo formáte s plávajúcou desatinnou čiarkou (veľkosť 4 bajty), tak sa do jedného paketu zmestí 336 vzoriek audio signálu. I<sup>2</sup>S periféria pracuje v tzv. *double buffer móde*, ktorý procesor podporuje na hardwarovej úrovni. Pre vzorky sú vytvorené dve polia a periféria zapisuje raz do jedného, raz do druhého pola. Ešte je dobré spomenúť, že zapisovanie prebieha cez DMA (Direct Memory Access), takže jadro procesora sa môže venovať iným úlohám.

Po naplnení pola je vlákno prebudené (s využitím synchronizačného mechanizmu *semafor*). Ako prvé sa v procese *Obsluha eventov* skontroluje, či nie sú aktívne nejaké udalosti (eventy), ako napr. zmena hlasitosti, zmena pripojenia input streamu k output streamu, zmena pripojenia device k inej slave jednotke, zmena v nastavení



Obr. 3.2: Vývojový diagram pre audio vlákno

ekvalizéru, atď. Ak nejaká zmena nastala tak sa v tomto procese zrealizuje.

V procese *Obsluha mailboxov* sa zisťuje, či si ovládací software vypýtal, prípadne zrušil, zasielanie streamov pre grafické zobrazenie. To sa vyhodnocuje ďalej v programe. V prípade, že je potrebné zaslať stream, tak sa pošle ako broadcast.

Podľa I<sup>2</sup>S štandardu sa vzorky pre ľavý a pravý kanál striedajú. V SPU jednotke sa pracuje s kanálmi samostatne. Preto je potrebné tieto vzorky rozdeliť. O to sa stará proces *Spracovanie vzoriek z I<sup>2</sup>S periférie*. Okrem toho sa vzorky posúvajú a upravujú do rozsahu  $\langle -1, 1 \rangle$  a násobia sa ziskom nastaviteľným v ovládacom software. Nastavenie zisku slúži na vyrovnanie hlasitosti zvuku pre rôzne vstupy.

### 3.2.2 Ekvalizér

Súčasťou zariadenia je digitálny sedem-pásmový ekvalizér nastaviteľný v rozsahu  $\langle -15, 15 \rangle$  dB. Ekvalizér je realizovaný sadou siedmych filtrov zaradených v sérii, každý pre jedno pásmo. Na začiatku pásma je *low shelving* filter, nastavený na medznú frekvenciu 100 Hz. Potom nasleduje 5 *peak* filtrov postupne nastavených na frekvencie 150 Hz, 400 Hz, 1 kHz, 2.4 kHz a 6 kHz. Na konci pásma sa nachádza *high shelving* filter s medznou frekvenciou 12 kHz.

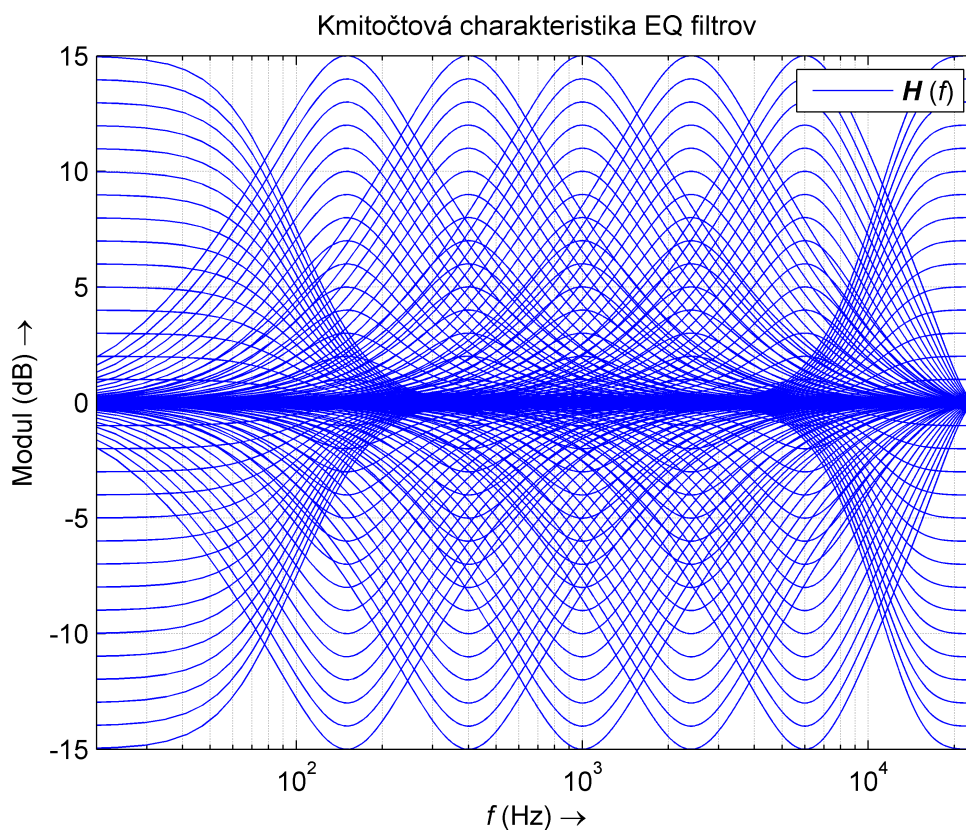
Filtre sú typu IIR 2. radu. Koeficienty filtrov sú normované, takže pre každý filter je potrebné uložiť 5 koeficientov. Keďže má každý filter 31 nastaviteľných ziskov, je potrebné vygenerovať 217 sád koeficientov. To je dohromady 1085 koeficientov. Koeficienty sú uložené v poli typu float a pri zmene ekvalizéru sa filter inicializuje vhodnými koeficientami. Výpočet koeficientov filtru, vrátane vytvorenia súboru s koeficientami, je realizovaný v Matlabe.

Na obr. 3.3 sú grafy modulov frekvenčných charakteristík pre všetky použité filtre.

Vzťahy pre výpočet koeficientov filtru sú uvedené v sekcii 1.3.3 *IIR filtre pre realizáciu digitálneho ekvalizéra* na strane 28.

Pre vyššiu kvalitu ekvalizéru by bolo vhodné použiť filtre 4. radu. Filter 4. radu je realizovaný sériovým zapojením dvoch filtrov druhého radu. Tým sa zvýši výpočtová náročnosť na dvojnásobok. Na použitom procesore nebolo možné realizovať takýto systém v reálnom čase.





Obr. 3.3: Modul frekvenčných charakteristík filtrov sedem-pásmového ekvalizéru

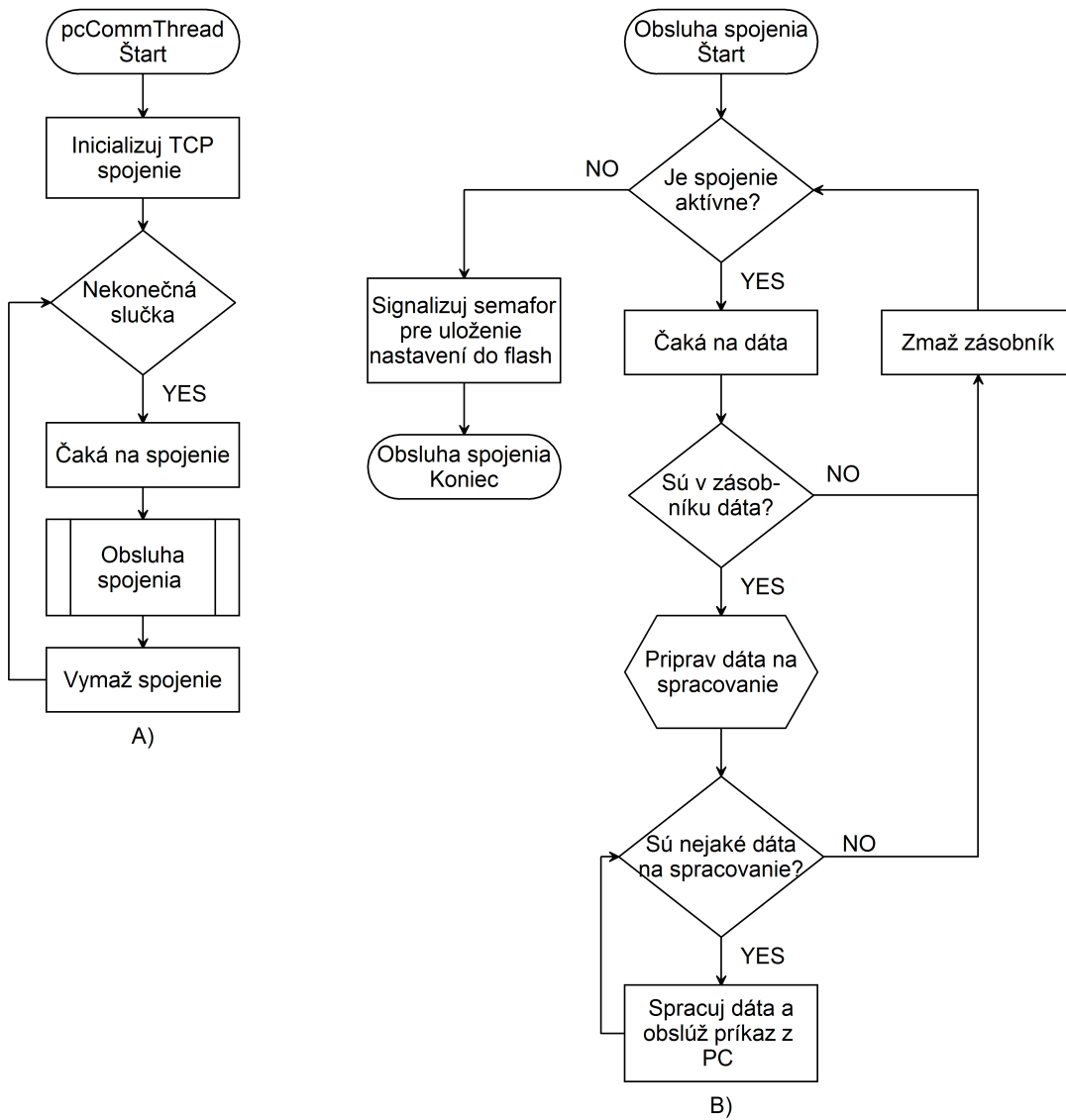
### 3.2.3 pcComm vlákno

Na obr. 3.4 vidíme vývojový diagram pre vlákno realizujúce komunikáciu s PC. V časti B je rozkreslený proces *Obsluha spojenia*. Interakcia s ostatnými vláknami je realizovaná pomocou udalostí (*events*). Podľa zmeny, ktorá sa má zrealizovať sa pošle všetkým počúvajúcim vláknam špecifický príznak, ktorý sa potom v cieľových vláknach vyhodnotí a zmena sa zrealizuje. Výhodou eventov je, že sa dajú použiť na synchronne predávanie správ viacerým vláknam, čo napríklad pomocou mailboxov alebo správ (*messages*) nie je možné.

Tu je treba podotknúť, že na začiatku každého príkazu z PC (a podobne aj odpovedi z SPU) sú vyhradené dva bajty, v ktorých je uložená veľkosť príkazu. To je potrebné, pretože Ethernet nezaručuje prijatie, alebo vyslanie správy v jednom celku. V prípade že príkaz neprišiel celý, tak sa čaká na prijatie ďalšieho paketu.

### 3.2.4 Ostatné vlákna

V programe sa používajú ešte nasledujúce vlákna:



Obr. 3.4: Vývojový diagram pre pcComm vlákno

- *main*: v main je realizovaná inicializácia operačného systému, načítanie nastavení z flash pamäte, vytvorenie vlákien a v prípade potreby zápis nastavení do flash.
- *ledThread*: v led vlákne sú realizované dve funkcie. Jedná sa o blikanie LED diódou s periódou 500 ms a detekciu stlačenia užívateľského tlačítka na kite STM32F4Discovery. Po stlačení tlačítka sa nastaví zariadenie do defaultných nastavení a mikrokontrolér sa resetuje. Toto vlákno existuje v projekte pre debugovacie účely.
- *lcdThread*: v lcd vlákne je realizované vykreslenie bitmapy na displej a jeho pohasnutie po 15 s (pokles jasů na 30 %, riadené PWM) a úplné zhasnutie po 60 s.
- *irdaThread*: irda vlákno obsluhuje príkazy prijaté z infračerveného ovládania. Takisto sa tu nachádza dekódovanie prijatých príkazov.

Vývojové diagramy k všetkým vláknam nájdeme v prílohe F *Vývojové diagramy* na strane 84.

## 4 KOMUNIKAČNÝ PROTOKOL

V tejto kapitole je stručne popísaný komunikačný protokol na riadenie a nastavenie systému pomocou programu v PC. Ďalej je popísaná štruktúra UDP paketu na prenos audio signálu medzi SPU jednotkou a externými reproduktormi.

V ďalšom texte sa bude pre označovanie jednotlivých blokov (podľa obr. 1 na strane 13) používať špecifické názvoslovie. Pod pojmom *master jednotka* (prípadne len *master*) sa bude rozumieť SPU (Signal Processing Unit), pod pojmom *slave jednotka* zase jeden DAS (Digital Active Speakers). Pre označenie vstupného audio kanálu pred spracovaním v SPU jednotke (napr. ľavý kanál z AD prevodníka) bude používaný termín *input stream*. Pre označenie audio toku smerujúceho do konkrétneho reproduktoru (napr. ľavý predný) sa bude používať termín *output stream*. Skupina output streamov určená pre konkrétnu slave jednotku je označená termínom *device*.

### 4.1 Komunikácia s PC

Medzi SPU a PC je vytvorené TCP/IP spojenie. Komunikácia je realizovaná formou textových správ s informáciami potrebnými pre nastavenie. Všetky nastavenia sú uložené v štruktúrach na to určených. Napr. v štruktúre typu BMaster.t sú uložené informácie o master jednotke (čiže SPU), konkrétne jej IP adresa, sériové číslo, verzia firmware a verzia hardware. Takto sú uložené aj informácie o input streamoch (popisuje konkrétny kanál konkrétneho vstupu), output streamoch (popisuje stream smerujúci do konkrétneho reproduktoru slave jednotky) a slave jednotkách (systém reproduktorov v nejakej miestnosti, napr. 2+1 alebo 5+1). Deklarácie štruktúr a komunikačné príkazy sa nachádzajú v súbore btypes.h, ktorého výpis je v prílohe H na strane 94.

Na vyžiadanie informácií z konkrétnej štruktúry slúžia príkazy typu GET\_x (napr. GET\_MASTER). Po obdržaní tohoto príkazu pošle SPU jednotka celú štruktúru naviazanú na tento príkaz. Príkazy tohto typu sa používajú prevažne po nadviazaní spojenia s PC softwarom, keďže software si po pripojení načítava informácie z SPU jednotky, kde sú permanentne uložené.

Pre zmenu nastavení sa používajú príkazy typu SET\_x (napr. SET\_OUTPUT\_STREAM). S týmto príkazom príde celá štruktúra, ktorou sa v SPU jednotke prepíše pôvodné nastavenie. Na tieto príkazy sa neposiela odpoveď.

Pre asynchrónne posielanie správ do PC sa používajú príkazy typu SEND\_x (napr. SEND\_INFRACOMMAND). Pomocou týchto príkazov sa aktualizujú nastavenia v PC, ak sa zmena udiala mimo PC softwaru (napr. prostredníctvom IrDA

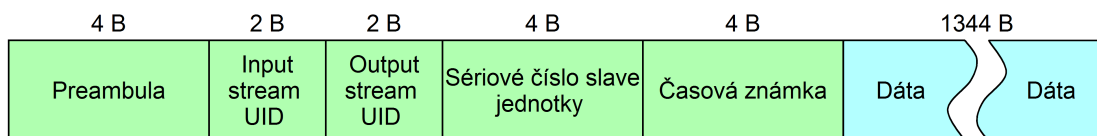
ovládača). Takisto sa používajú na posielanie informácií slúžiacich na logovanie do počítača (napr. teploty prijaté zo slave jednotky).

Posledná skupina príkazov typu GIVE\_x (napr. GIVE\_INPUTSTREAM) slúži na vyžiadanie, respektíve zrušenie (napr. GIVE\_STOPINPUTSTREAM), zasielania input alebo output streamu pre zobrazenie v PC formou grafu (časový priebeh audio signálu). Vtedy sa stream posiela ako broadcast.

Komunikačný protokol vymyslel Jakub Laník a jeho podrobný popis je možné nájsť v jeho diplomovej práci [2].

## 4.2 Štruktúra UDP paketu

Každý paket, ktorý sa posiela do slave jednotky, je zložený z hlavičky a vzoriek vo formáte float (dáta). Na obr. 4.1 je znázornená štruktúra paketu.



Obr. 4.1: Štruktúra UDP paketu

Preambula je dlhá 4 bajty a je pevne nastavená na hodnotu 0xFFFFFFFF alebo 0xFFFFFFFFE. LSB bit v preambule slúži na určenie typu streamu. Ak je nastavený na 1, tak sa jedná o input stream a ak je nastavený na 0, tak sa jedná o output stream. Zasielanie input streamu je len pre účely grafického zobrazenia signálu v PC, takže v slave jednotke sa kontroluje hlavička s 0xFFFFFFFF. Ak začína paket inou hodnotou, tak sa zahadzuje.

V nasledovnej dvojici dvoch bajtov sú uložené identifikačné čísla input a output streamu. Podľa UID output streamu smeruje slave jednotka audio signál do správneho reproduktora.

V hlavičke je ešte uložené sériové číslo slave jednotky, pre ktorú je stream určený a časová známka. V časovej známke je uložený relatívny čas vyslania paketu. Používa sa preto, aby bolo možné prijaté pakety v slave jednotke správne usporiadať, keďže prenosové médium nezaručuje prenos správ v poradí vyslania.

Na konci paketu sa nachádzajú samotné dáta s veľkosťou 1344 bajtov. Celý paket má teda dĺžku 1360 B.

## 5 ZÁVER

V prvej časti práce sú uvedené základné spôsoby spracovania signálov pomocou DSP techník. Jedná sa o priblíženie funkcie AD prevodníkov. Tie sú z veľkej väčšiny založené na použití delta-sigma modulácie, a preto je v práci vysvetlený práve tento druh prevodníkov. Základným prvkom DSP spracovania signálu sú digitálne filtre. Venovali sme sa bežne používaným typom filtrov, ako napr. filtre realizujúce kľzavý priemer, windowed-sync filtre, Chebyshevove filtre a špeciálnym typom filtrov vhodných pre realizáciu digitálneho ekvalizéru. Medzi základné techniky číslicového spracovania signálov patrí tiež rýchla Fourierova transformácia. V kapitole 1.4.1 je preto popísaný spôsob jej fungovania.

V ďalšej časti práce je navrhnutá topológia riadiacej jednotky pre spracovanie zvukového signálu. Centrálnou časťou jednotky je mikrokontrolér SMT32F407, o prevod audio signálu do štandardu I<sup>2</sup>S sa stará AD prevodník CS5361 a prevodník CS8416 a spracovaný signál sa posielajú do externých reproduktorov pomocou protokolu rýchleho Ethernetu. Jednotku je možné riadiť a nastavovať pomocou infračerveného ovládania alebo softwaru v PC. Pomocou IrDA ovládača je možné meniť hlasitosť alebo prehrávaný vstup (či sa v aktívnej slave jednotke prehráva analógový alebo digitálny vstup).

V tretej časti práce je popísaná funkcia firmwaru zariadenia. V mikrokontroléru sa využíva systém reálneho času, konkrétne sa jedná o ChibiOS/RT. Firmware bol vyvíjaný v prostredí zloženého z Eclipse, kompilátoru GCC pre ARM-y a OpenOCD, určeného pre ladenie aplikácie priamo v mikrokontroléru.

V poslednej časti práce je stručne popísaný komunikačný protokol medzi SPU jednotkou a PC softwarom. Pomocou softwaru v PC je možné meniť:

- získanie jednotlivých kanálov (pre vyrovnanie úrovne hlasitosti z rôznych vstupných zariadení),
- smerovanie kanálu do ľubovoľného reproduktora,
- hlasitosť pre jednotlivé kanály vrátane možnosti stíšiť kanál (*mute*),
- nastavenie ekvalizéra, prípadne jeho povolenie, resp. zakázanie,
- posielanie skupín output streamov (*device*) do konkrétnych slave jednotiek (napr. vstup z CD prehrávača sa prehráva v slave jednotke v kuchyni a vstup z televízie sa prehráva v obývacej izbe).

Pri zmene nastavení pomocou IrDA ovládania je táto informácia v PC softwaru automaticky aktualizovaná. Príkazy z ovládača sa v PC zaznamenávajú a je možné ich dohľadať v histórii príkazov. Z časových dôvodov sa nestihlo zrealizovať zaznamenávanie informácií zo slave jednotiek (napríklad teploty alebo príkazy z infračerveného ovládania slave jednotky). V PC je ďalej možné zobrazíť časový priebeh audio signálu pred a po spracovaní (napríklad po aplikácii ekvalizéru).

V prílohe G na strane 90 je uvedené meranie parametrov SPU jednotky spolu s externými reproduktormi, realizovanými v rámci diplomovej práce Jakuba Nedomy, vid' [3]. V protokole je zameraná latencia zvukového signálu, zaťaženie siete pri rôznom počte posielaných kanálov a porovnanie spektier vstupného a výstupného harmonického signálu.

Čelný a zadný panel zrealizovaného zariadenia je ukázaný v prílohe E na obr. E.1. Zapojenie elektroniky je patrné z obr. E.2.

# LITERATÚRA

- [1] SMITH, S. W. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Newnes: 1 edition (November 6, 2002). 650 s. ISBN-10: 075067444X. ISBN-13: 978-0750674447.
- [2] LANÍK, J. *Systém pro monitorování a nastavení signálového procesoru v digitálním reproduktorovém systému*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 71 s. Diplomová práce.
- [3] NEDOMA, J. *Aktivní reproduktor s digitálním vstupem zvukového signálu a s přídatnými funkcemi*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 82 s. Diplomová práce.
- [4] KOLKA, Z. *Počítačové a komunikační sítě*. Elektronické skriptá, UREL FEKT VUT.
- [5] ADAM, P. *Úvod do metód spracovania zvuku v súčasnom multimediálnom prostredí*: diplomová práca. BRATISLAVA: Univerzita Komenského. Fakulta matematiky, fyziky a informatiky. Katedra aplikovanej informatiky. 2006. 144 s.
- [6] *An Overview of the Electrical Validation of 10BASE-T, 100BASE-TX, and 1000BASE-T Device* [online]. [cit. 24. 04. 2012]. Dostupné z URL: <<http://cp.literature.agilent.com/litweb/pdf/5989-7528EN.pdf>>.
- [7] *Fast Ethernet* [online]. 2012, posledná aktualizácia 11. 04. 2012 [cit. 24. 04. 2012]. Dostupné z URL: <[http://en.wikipedia.org/wiki/Fast\\_ethernet](http://en.wikipedia.org/wiki/Fast_ethernet)>.
- [8] *UM1472: STM32F4DISCOVERY STM32F4 high-performance discovery board* [online]. 2012, posledná aktualizácia 19. 01. 2012 [cit. 25. 04. 2012]. Dostupné z URL: <[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/USER\\_MANUAL/DM00039084.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/DM00039084.pdf)>.
- [9] *CS5361 Product Data Sheet* [online]. 2005, posledná aktualizácia 02. 2005 [cit. 25. 04. 2012]. Dostupné z URL: <[http://www.cirrus.com/en/pubs/proDatasheet/CS5361\\_F2.pdf](http://www.cirrus.com/en/pubs/proDatasheet/CS5361_F2.pdf)>.
- [10] *Ethernet frame* [online]. 2012, posledná aktualizácia 02. 04. 2012 [cit. 25. 04. 2012]. Dostupné z URL: <[http://en.wikipedia.org/wiki/Ethernet\\_frame](http://en.wikipedia.org/wiki/Ethernet_frame)>.



- [11] *RM0090: STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs* [online]. 2011, posledná aktualizácia 16. 09. 2011 [cit. 25. 04. 2012]. Dostupné z URL:  
<[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/REFERENCE\\_MANUAL/DM00031020.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/DM00031020.pdf)>.
- [12] *STMicroelectronics: IDEs and Toolsets for MCUs* [online]. [cit. 16. 10. 2012]. Dostupné z URL:  
<[http://www.st.com/internet/com/software/ides\\_mcu.jsp](http://www.st.com/internet/com/software/ides_mcu.jsp)>.
- [13] *ChibiOS/RT Homepage* [online]. [cit. 16. 10. 2012]. Dostupné z URL:  
<<http://www.chibios.org/dokuwiki/doku.php?id=start>>.
- [14] *S/PDIF* [online]. 2012, posledná aktualizácia 16. 10. 2012 [cit. 4. 11. 2012]. Dostupné z URL:  
<<http://en.wikipedia.org/wiki/S/PDIF>>.
- [15] *DIR9001 Product Data Sheet* [online]. 2006, posledná aktualizácia 12. 2006 [cit. 4. 11. 2012]. Dostupné z URL:  
<<http://www.ti.com/lit/ds/symlink/dir9001.pdf>>.
- [16] *CS8416 Product Data Sheet* [online]. 2007, posledná aktualizácia 09. 2007 [cit. 4. 11. 2012]. Dostupné z URL:  
<[http://www.cirrus.com/en/pubs/proDatasheet/CS8416\\_F3.pdf](http://www.cirrus.com/en/pubs/proDatasheet/CS8416_F3.pdf)>.
- [17] *AN241 – Analog Input Buffer Architectures* [online]. 2003, posledná aktualizácia 10. 2003 [cit. 17. 11. 2012]. Dostupné z URL:  
<<http://www.cirrus.com/en/pubs/appNote/an241-1.pdf>>.
- [18] *DP83848 Product Data Sheet* [online]. 2008, posledná aktualizácia 05. 2008 [cit. 22. 11. 2012]. Dostupné z URL:  
<<http://www.ti.com/lit/ds/symlink/dp83848c.pdf>>.
- [19] *AN3984: IIR filter design equations for Sound Terminal* [online]. 2011, posledná aktualizácia 09. 2011 [cit. 07. 05. 2013]. Dostupné z URL:  
<[http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/application\\_note/DM00038862.pdf](http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/application_note/DM00038862.pdf)>.
- [20] *CD4051B, CD4052B, CD4053B Product Data Sheet* [online]. 2003, posledná aktualizácia 10. 2003 [cit. 08. 05. 2013]. Dostupné z URL:  
<<http://www.ti.com/lit/ds/symlink/cd4052b.pdf>>.

- [21] *SSD1289 Product Data Sheet* [online]. 2006, posledná aktualizácia 09. 2006 [cit. 07. 05. 2013]. Dostupné z URL:  
<[http://www.mcu-china.com/tft\\_doc/SSD1289\\_2.8.pdf](http://www.mcu-china.com/tft_doc/SSD1289_2.8.pdf)
- [22] *ADS7843 TOUCH SCREEN CONTROLLER* [online]. 2002, posledná aktualizácia 05. 2002 [cit. 07. 05. 2013]. Dostupné z URL:  
<<http://www.ti.com/lit/ds/sbas090b/sbas090b.pdf>

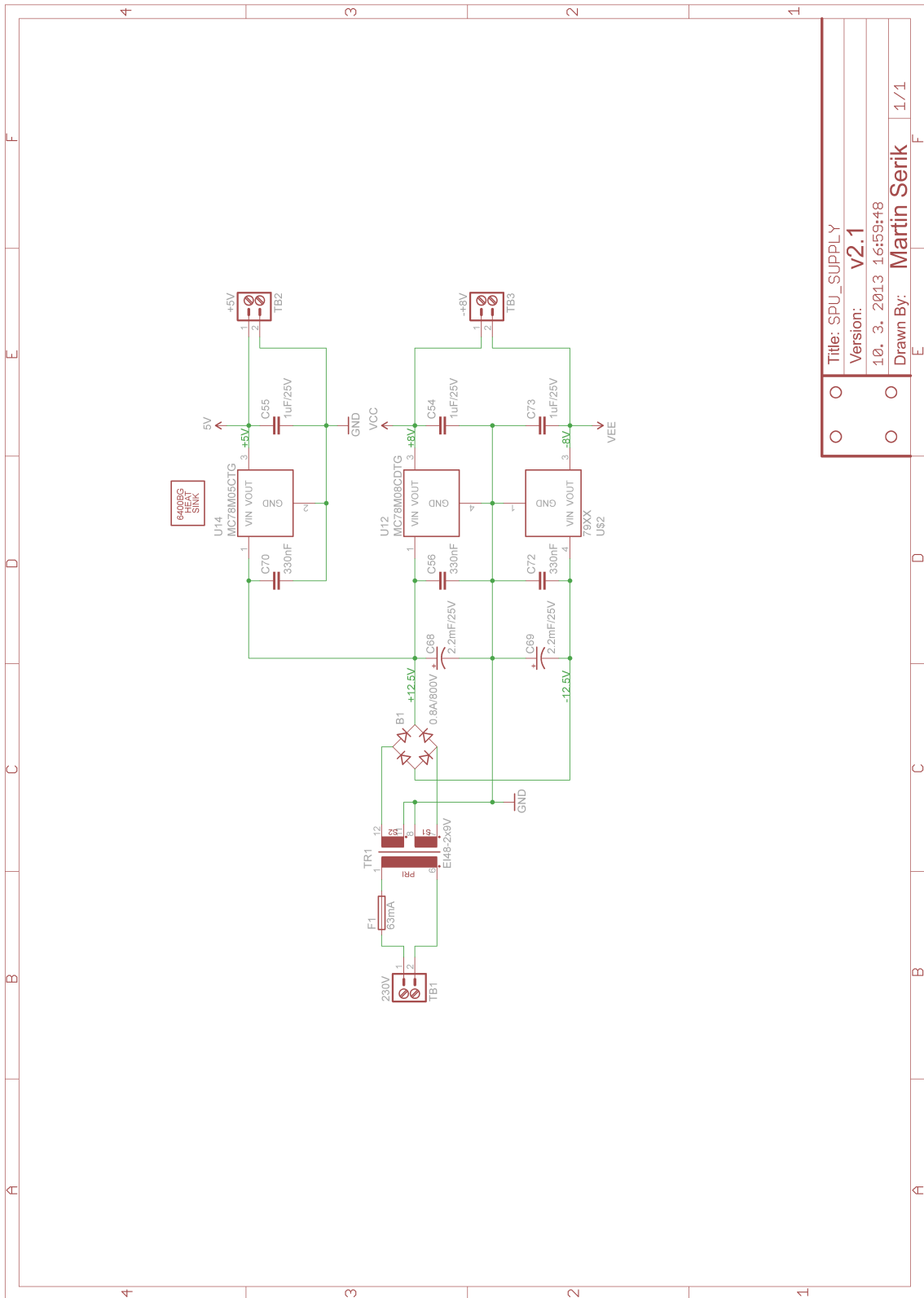
## ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

DSP	Digital Signal Processing – číslicové spracovanie signálov
ADC	Analog-to-Digital Conversion – analógovo-číslcová konverzia
DAC	Digital-to-Analog Conversion – číslicovo-analógová konverzia
SPL	Sound Power Level
Hi-Fi	High Fidelity – vysoká vernosť
FIR	Finite Impulse Response – konečná impulzná odozva
IIR	Infinite Impulse Response – nekonečná impulzná odozva
DTFT	Discrete Time Fourier Transform – Fourierova transformácia s diskretným časom
DFT	Discrete Fourier Transform – diskretná Fourierova transformácia
FFT	Fast Fourier Transform – rýchla Fourierova transformácia
IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
PHY	Physical Layer Interface
MII	Media Independent Interface
RMII	Reduced Media Independent Interface
FPU	Floating Point Unit – jednotka pre počítanie s číslami s pohyblivou desatinnou čiarkou
DMIPS	Dhrystone Million Instructions Per Second
MAC	Multiply and ACcumulate
SOF	Start Of Frame
FCS	Frame Check Sequence
IDE	Integrated Development Environment – integrované vývojové prostredie
S/PDIF	Sony/Philips Digital Interconnect Format
I <sup>2</sup> S	Integrated Interchip Sound

# ZOZNAM PRÍLOH

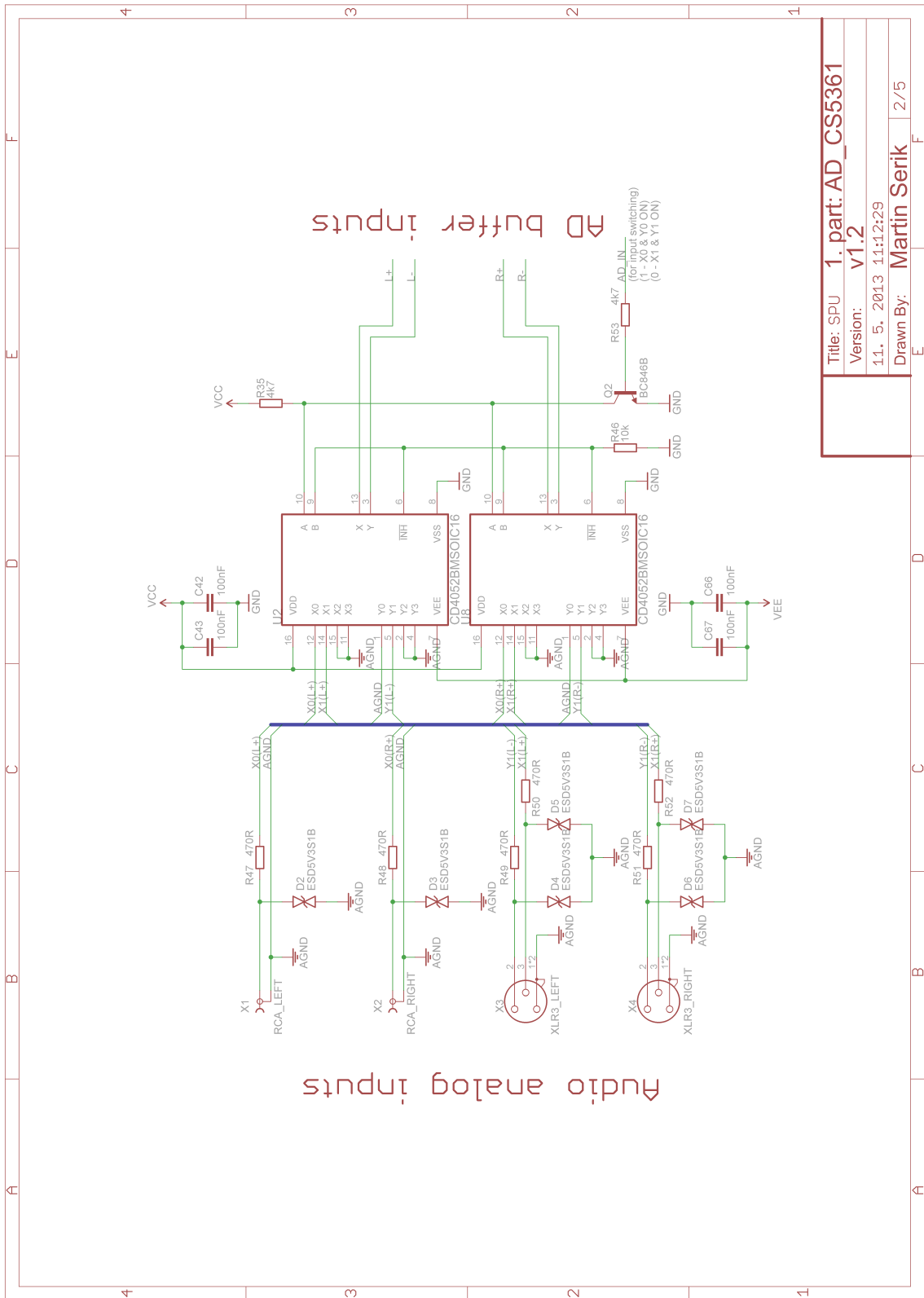
<b>A</b>	<b>Schéma zapojenia SPU</b>	<b>65</b>
<b>B</b>	<b>Návrhy motívu plošného spoja</b>	<b>72</b>
<b>C</b>	<b>Osadzovacie plány DPS</b>	<b>75</b>
<b>D</b>	<b>Zoznam súčiastok</b>	<b>78</b>
<b>E</b>	<b>Fotodokumentácia</b>	<b>82</b>
<b>F</b>	<b>Vývojové diagramy</b>	<b>84</b>
<b>G</b>	<b>Meranie parametrov systému s SPU jednotkou</b>	<b>90</b>
<b>H</b>	<b>btypes.h</b>	<b>94</b>
<b>I</b>	<b>Prílohy na CD</b>	<b>103</b>
I.1	Dokumentácia . . . . .	103
I.2	Schémy SPU . . . . .	103
I.3	Plošné spoje SPU . . . . .	103
I.4	Podporné programy v SW MATLAB . . . . .	103
I.5	Firmware SPU . . . . .	103
I.6	Fotodokumentácia . . . . .	103

## A SCHÉMA ZAPOJENIA SPU



Obr. A.1: Schéma zapojenia napájania

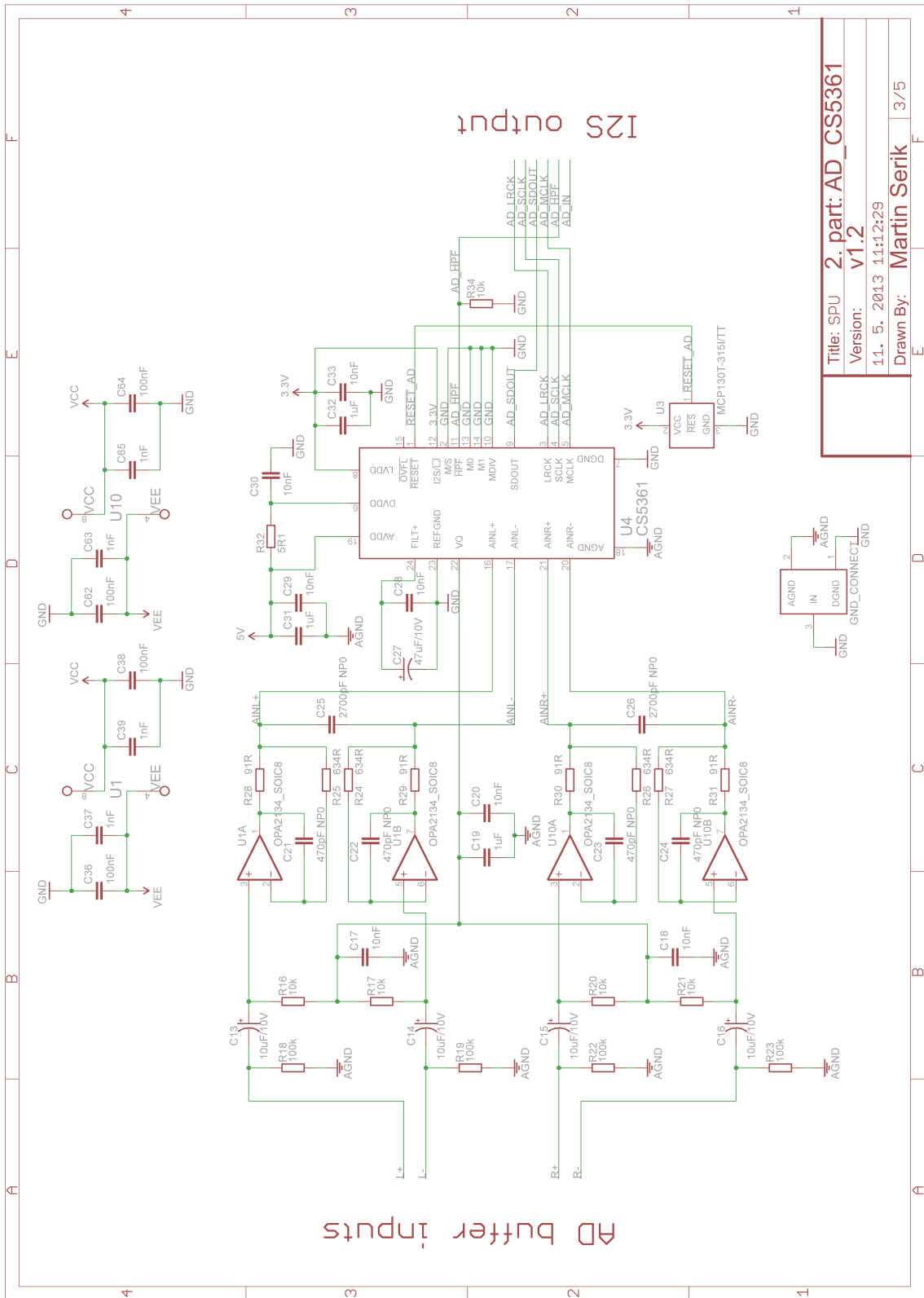




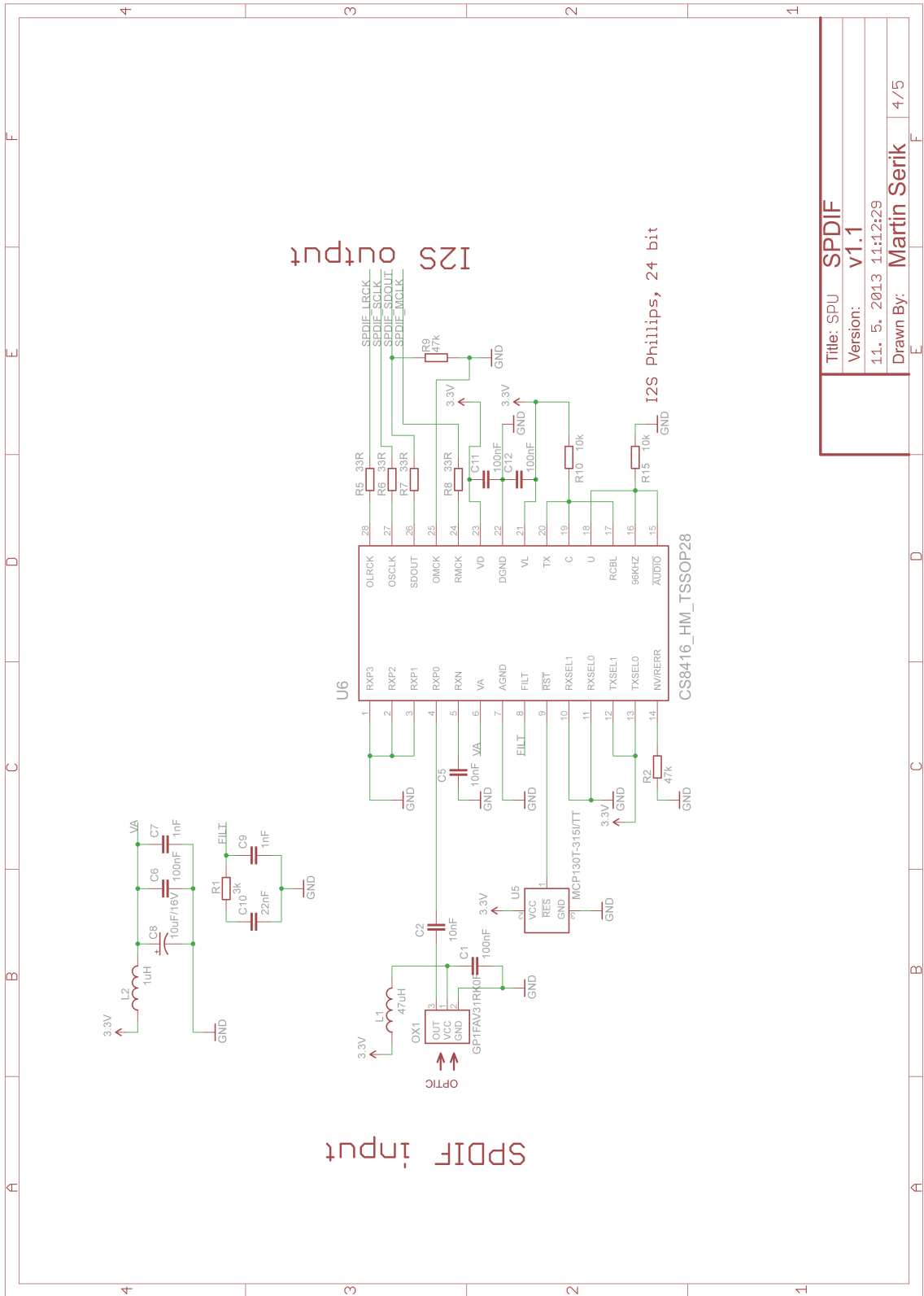
Obr. A.3: Schéma zapojenia AD prevodníka, 1. časť

Title: SPU 1. part: AD_CS5361	E
Version: v1.2	F
1.1. 5. 2013 11:12:29	F
Drawn By: Martin Serik	F
2/5	F



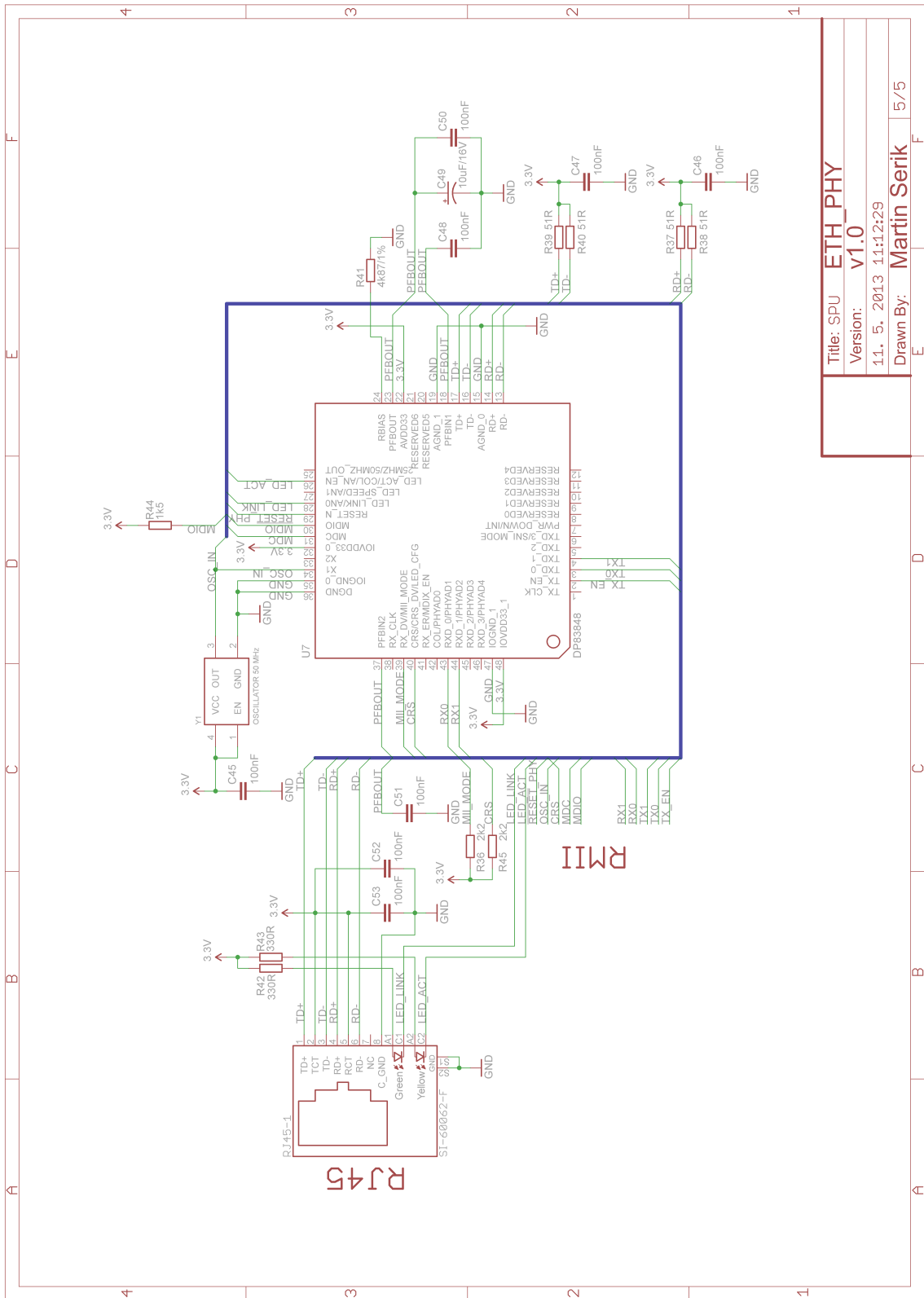


Obr. A.4: Schéma zapojenia AD prevodníka, 2. časť



Title: SPU SPDIF	
Version: v1.1	
1.1. 5. 2013 11:12:29	
Drawn By: Martin Serik	4/5

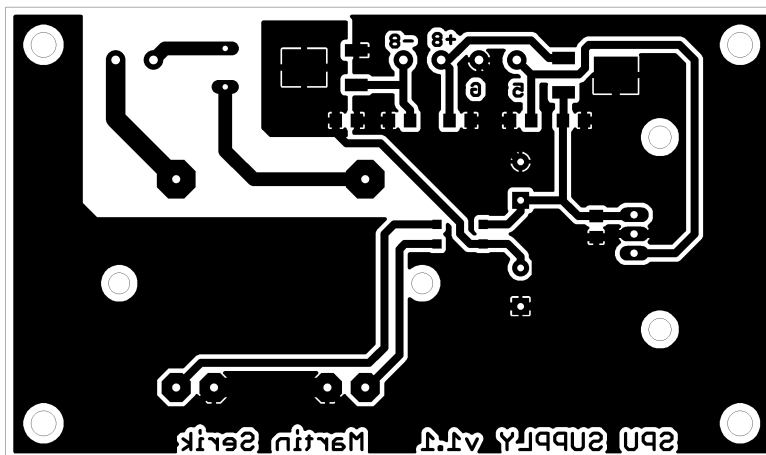
Obr. A.5: Schéma zapojenia S/PDIF modulu



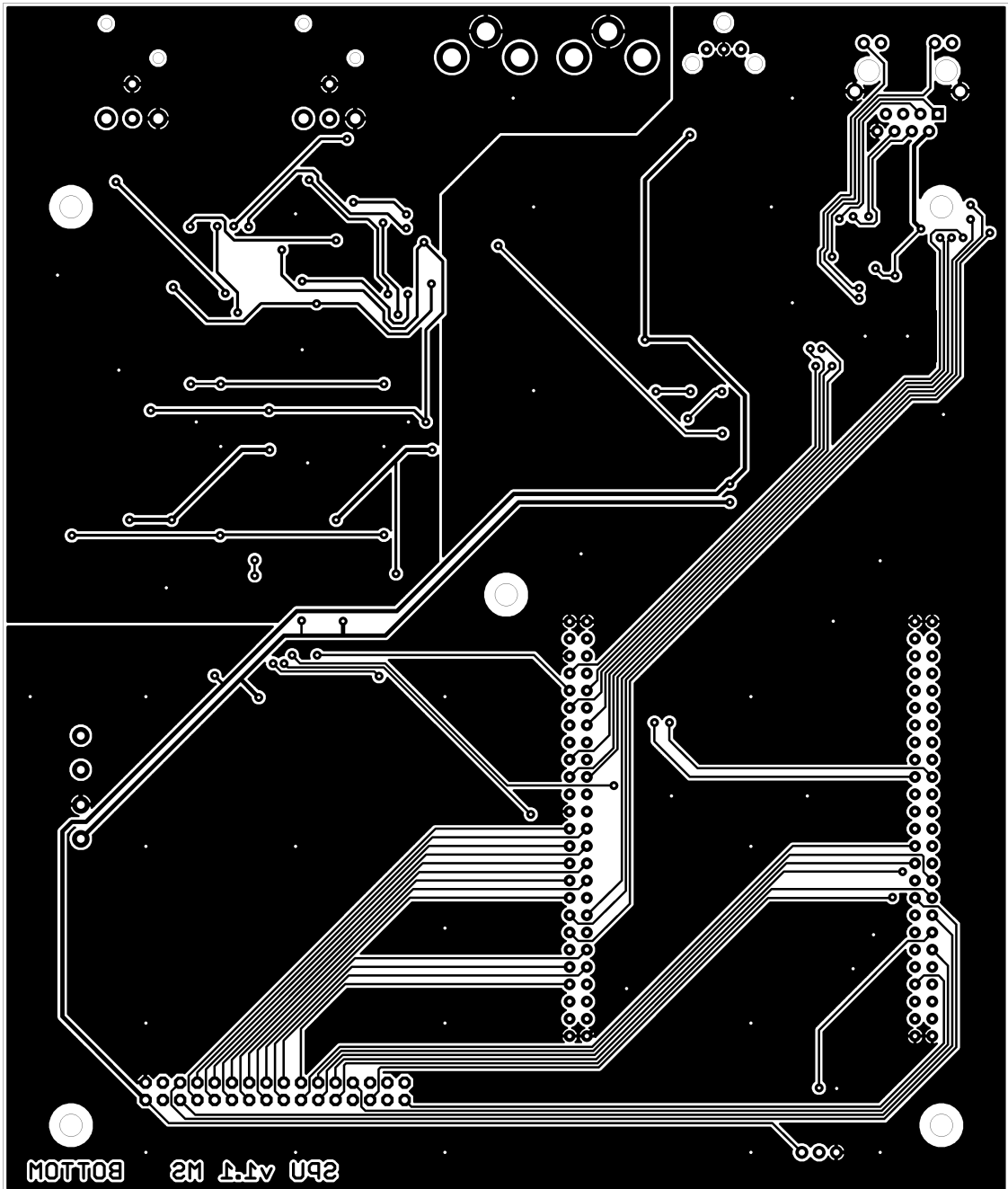
Title: SPU	ETH_PHY
Version:	v1.0
1.1. 5. 2013 11:12:29	
Drawn By:	Martin Serik
	5/5

Obr. A.6: Schéma zapojenia fyzickej vrstvy Ethernetu

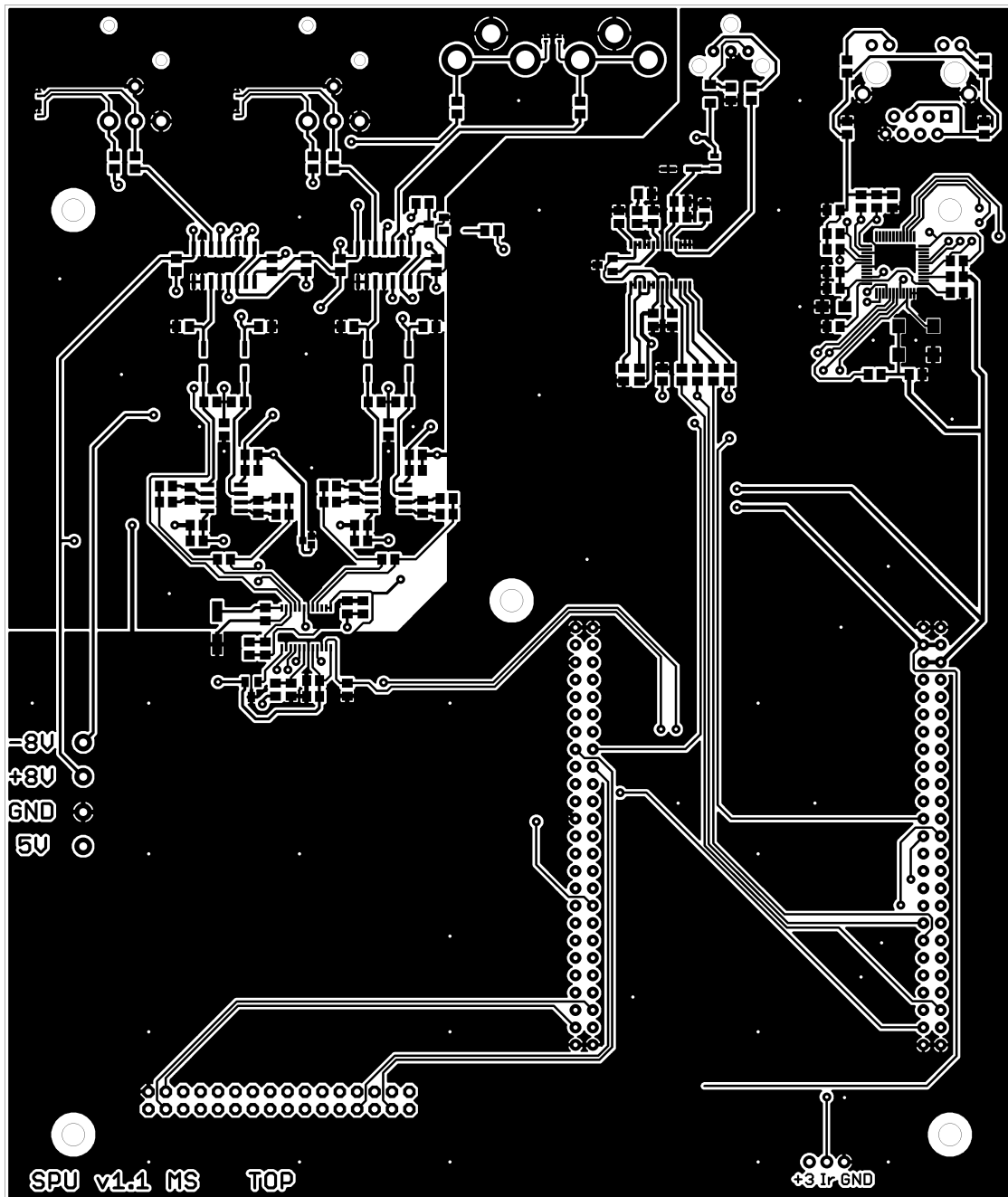
## B NÁVRHY MOTÍVU PLOŠNÉHO SPOJA



Obr. B.1: Motív plošného spoja napájania, 102 x 60 mm, 1:1



Obr. B.2: Motív plošného spoja SPU jednotky, BOTTOM vrstva, 148 x 175 mm, 1:1

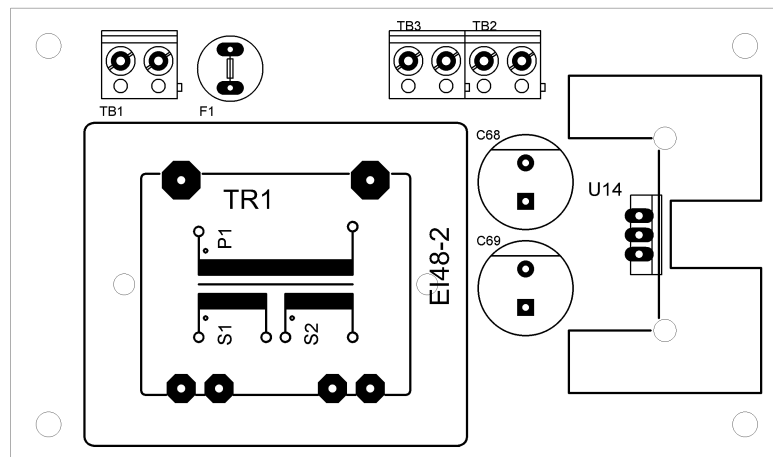


Obr. B.3: Motív plošného spoja SPU jednotky, TOP vrstva, 148 x 175 mm, 1:1

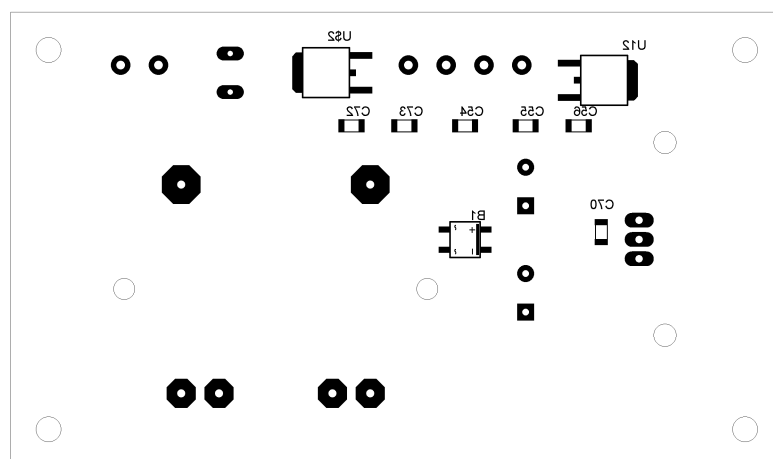
## C OSADZOVACIE PLÁNY DPS







Obr. C.2: Osadzovací plán napájacej časti SPU jednotky, strana TOP



Obr. C.3: Osadzovací plán napájacej časti SPU jednotky , strana BOTTOM

## D ZOZNAM SÚČIASTOK

Tab. D.1: Zoznam súčiastok pre STM32F4Discovery schému

Položka	Označenie	Púzdro	Hodnota
1.	KT1	KIT	-
2.	KT1	2ROW,25WAY	2ROW,25WAY
3.	TB4,TB5	5mm	2WAY
4.	JP21	2ROW,20WAY	2ROW,20WAY

Tab. D.2: Zoznam súčiastok pre AD\_CS5361 schému

<b>Položka</b>	<b>Označenie</b>	<b>Púzdro</b>	<b>Hodnota</b>
1.	U4	TSSOP24	-
2.	U3	SOT-23	3.15V
3.	U1,U10	SOIC8	-
4.	U2,U8	SOIC16	-
5.	Q2	SOT-23	300mW
6.	D2,D3,D4,D5,D6,D7	0402	6V
7.	C27	PANASONIC-D	47uF/10V
8.	C13,C14,C15,C16	1206	10uF/10V
9.	C19,C31,C32	0805	1uF/25V
10.	C36,C38,C42,C43,C62,C64,C66,C67	0805	100nF
11.	C17,C18,C20,C28,C29,C30,C33	0805	10nF
12.	C25,C26	0805	2700pF
13.	C37,C39,C63,C65	0805	1nF
14.	C21,C22,C23,C24	0805	470pF
15.	R18,R19,R22,R23	0805	100k
16.	R16,R17,R20,R21,R34,R46	0805	10k
17.	R35,R53,R54	0805	4k7
18.	R24,R25,R26,R27	0805	634R
19.	R47,R48,R49,R50,R51,R52	0805	470R
20.	R28,R29,R30,R31	0805	91R
21.	R32	0805	5R1
22.	XLR3_LEFT,XLR3_RIGHT	XLR	XLR3
23.	RCA_LEFT	RCA	White
24.	RCA_RIGHT	RCA	Red

Tab. D.3: Zoznam súčiastok pre SPDIF schému

Položka	Označenie	Púzdro	Hodnota
1.	U6	TSSOP28	-
2.	U5	SOT-23	3.15V
3.	OX1	GP1FAV31RK0F	RX
4.	L1	1008	47uH
5.	L2	0805	1uH
6.	C8	PANASONIC-B	10uF/16V
7.	C1, C6, C11, C12	0805	100nF
8.	C10	0805	22nF
9.	C2, C3	0805	10nF
10.	C7, C9	0805	1nF
11.	R2, R9	0805	47k
12.	R10,R15	0805	10k
13.	R1	0805	3k
14.	R5, R6, R7, R8	0805	33R

Tab. D.4: Zoznam súčiastok pre ETH\_PHY schému

Položka	Označenie	Púzdro	Hodnota
1.	U7	LQFP48	-
2.	RJ45-1	RJ45	-
3.	Y1	7.0X5.0	50 MHz
4.	C49	EIA3216	10uF/16V
5.	C45,C46,C47,C48,C50,C51,C52,C53	0805	100nF
6.	R41	0805	4k87/1%
7.	R36,R45	0805	2k2
8.	R44	0805	1k5
9.	R42,R43	0805	330R
10.	R37,R38,R39,R40	0805	51R

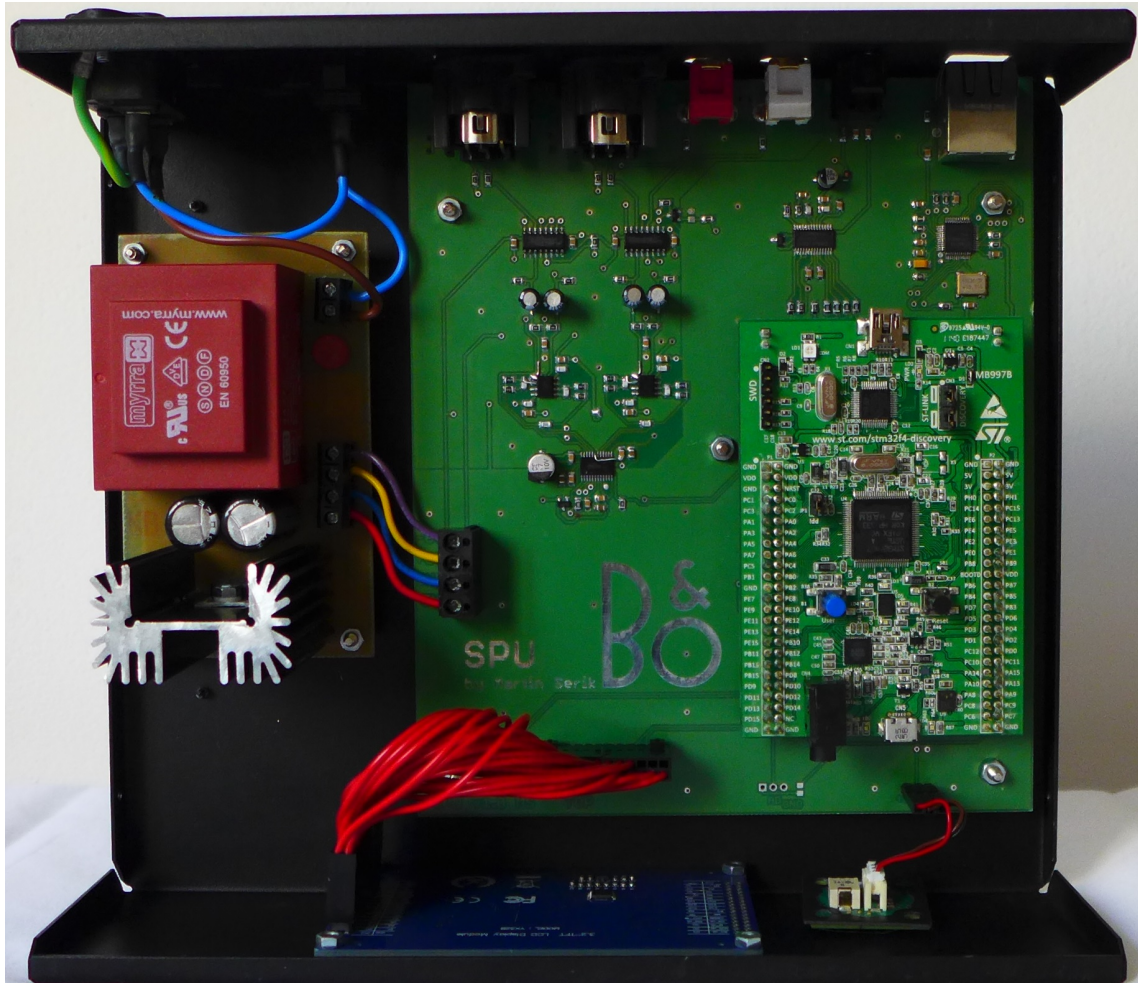
Tab. D.5: Zoznam súčiastok pre SUPPLY schému

Položka	Označenie	Púzdro	Hodnota
1.	U12	DPAK	8V
2.	U13	DPAK	-8V
3.	U14	TO-220	5V
4.	TR1	EI48-2	2x9V
5.	B1	SOIC4	0.8A/800V
6.	C56,C70,C72	1206	330nF
7.	C54,C55,C73	1206	1uF/25V
8.	C68,C69	Through Hole - 12.5 mm, 5 mm	2.2mF/25V
9.	F1	TR5	63mA/230V
10.	6400GB	Heatsink	2.7°C/W
11.	TB1,TB2,TB3	5mm	2WAY

## E FOTODOKUMENTÁCIA

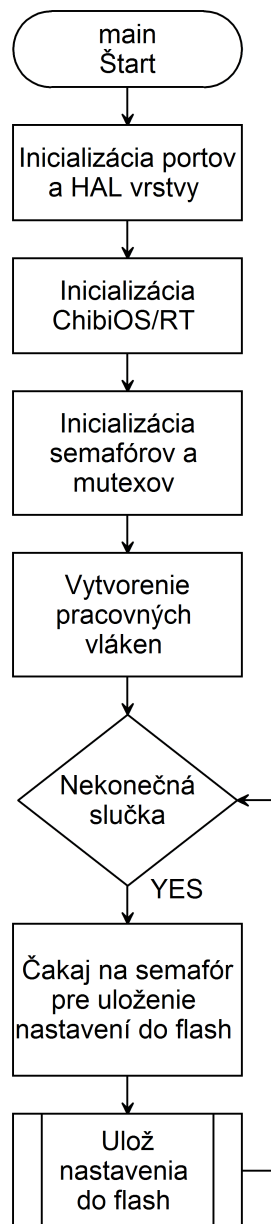


Obr. E.1: Pohľad na predný a zadný panel zariadenia



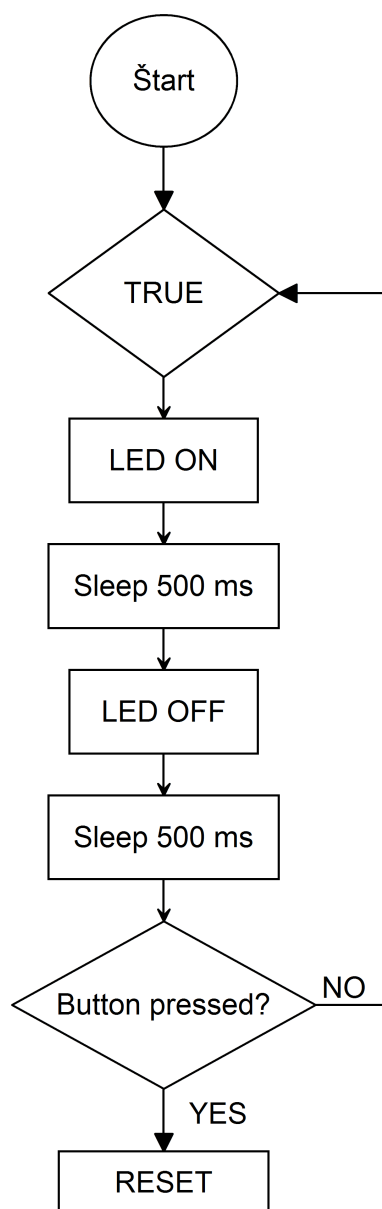
Obr. E.2: Pohľad na elektroniku v zariadení

## F VÝVOJOVÉ DIAGRAMY

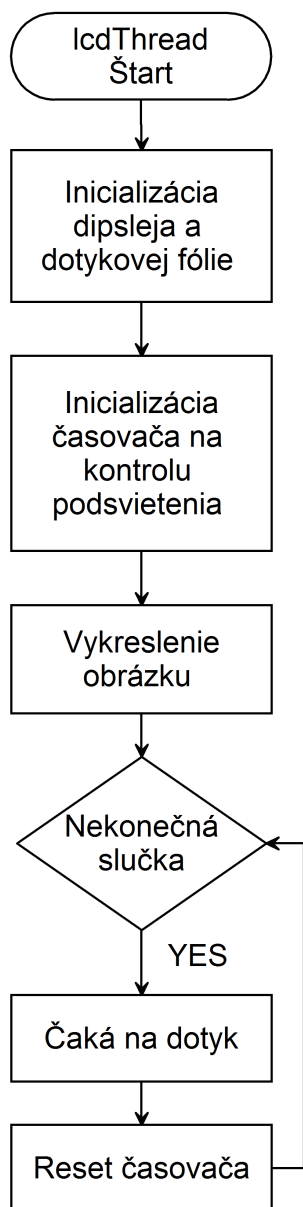


Obr. F.1: Vývojový diagram main vlákna

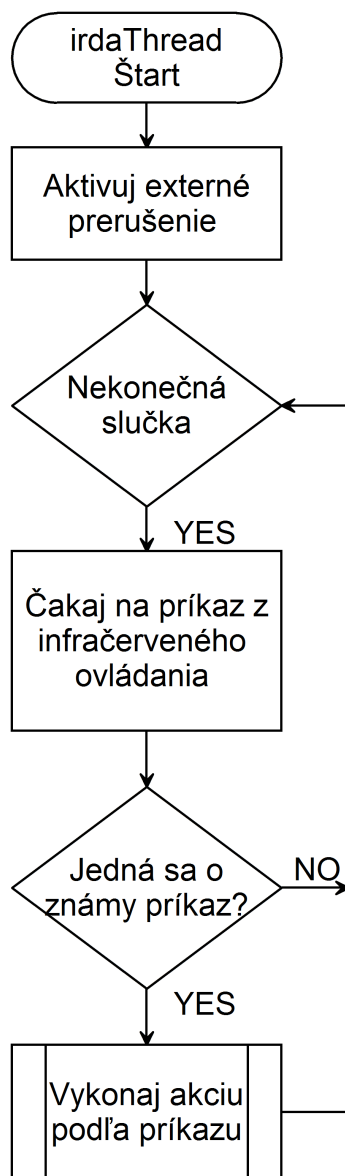




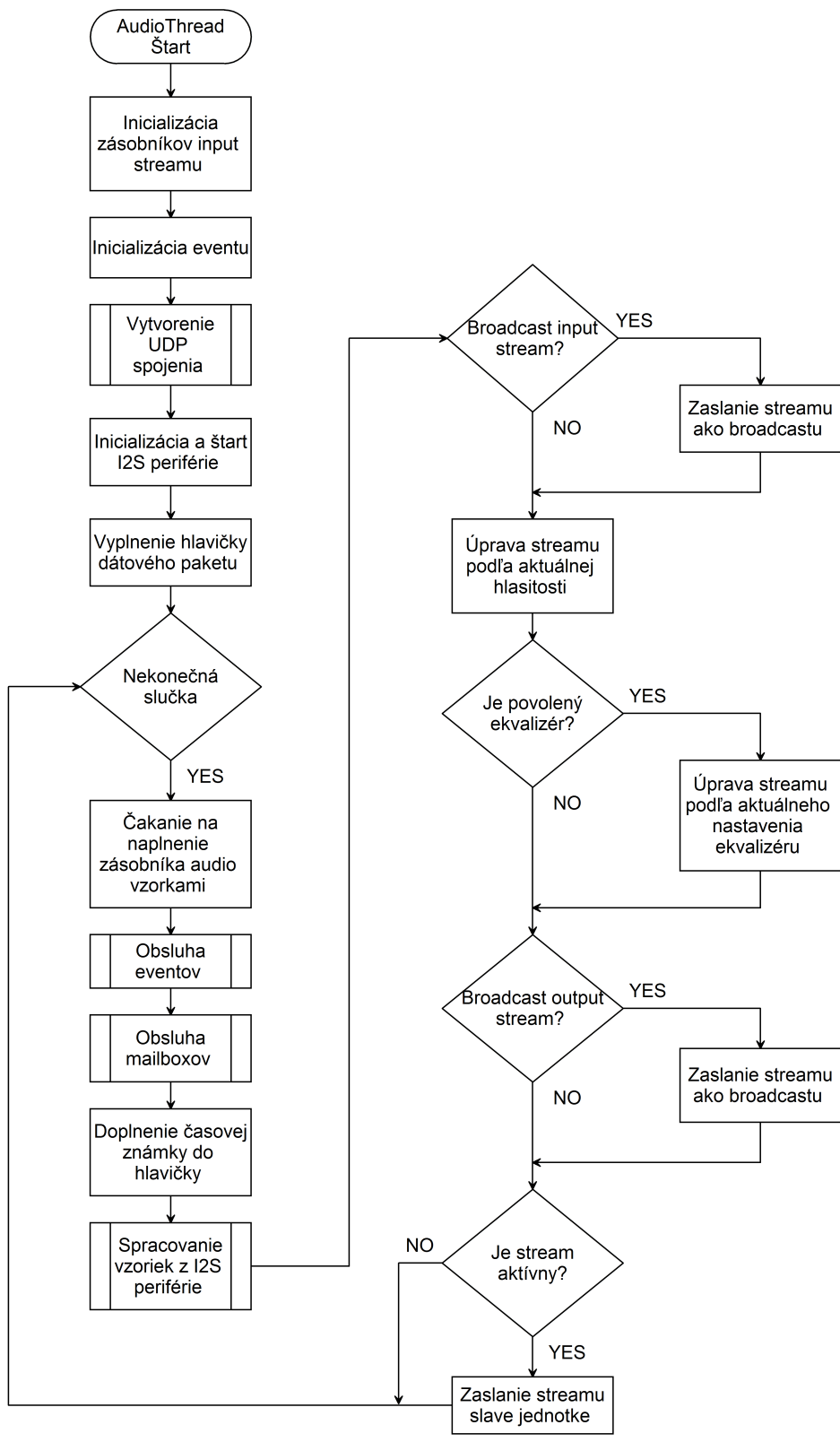
Obr. F.2: Vývojový diagram led vlákna



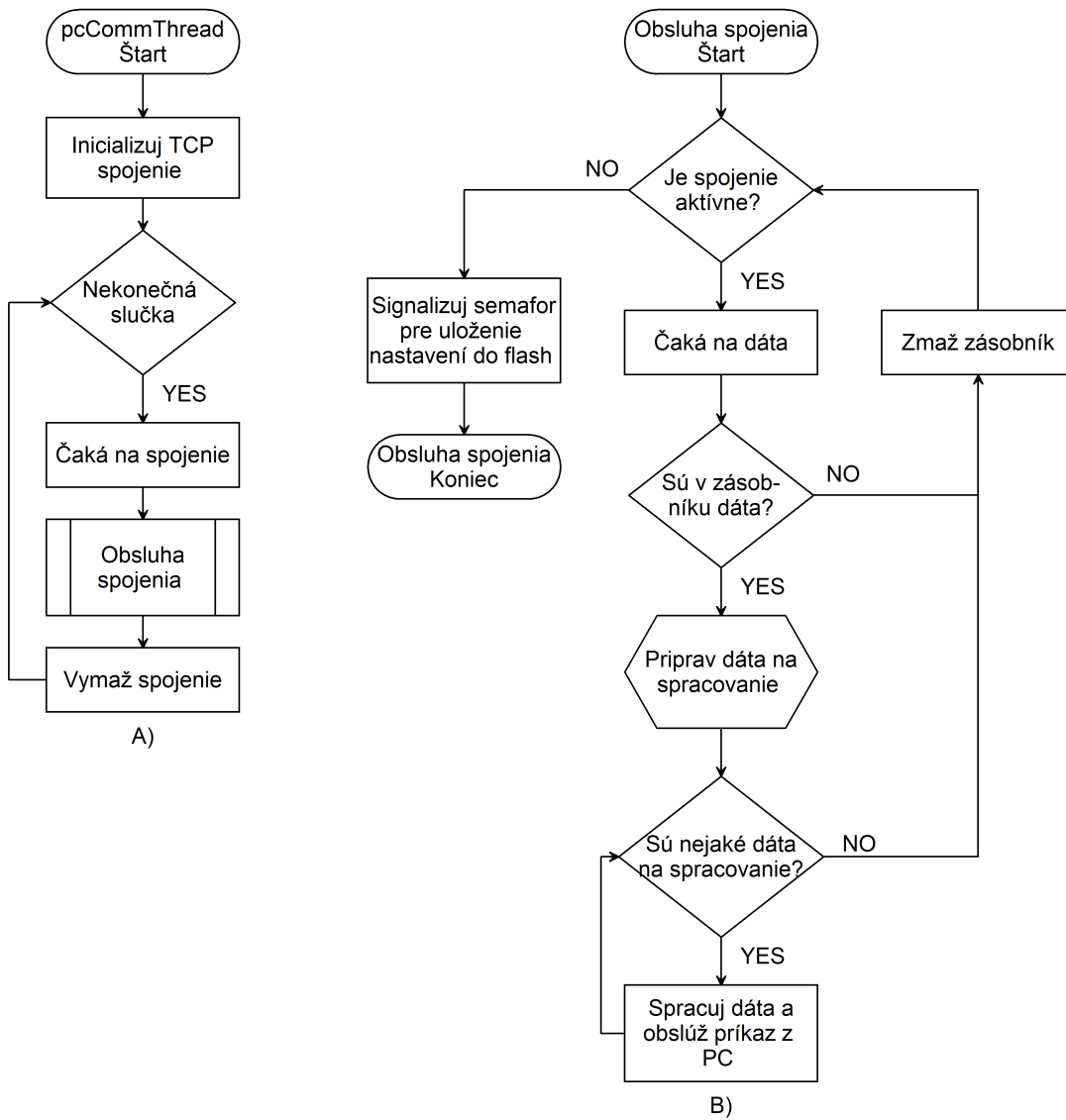
Obr. F.3: Vývojový diagram lcd vlákna



Obr. F.4: Vývojový diagram irda vlákna



Obr. F.5: Vývojový diagram audio vlákna



Obr. F.6: Vývojový diagram pcComm vlákna

# G MERANIE PARAMETROV SYSTÉMU S SPU JEDNOTKOU

Vypracoval Martin Šerík a Jakub Nedoma.

## Úvod

V meraní sa budú určovať parametre systému pre ozvučenie obytnej jednotky, zloženého z SPU a externých reproduktorov (podrobnosti k reproduktorom v [3]). Medzi zmeranými parametrami sa bude nachádzať porovnanie frekvenčného spektra harmonického signálu na vstupe a výstupe systému, latencia signálu a zaťaženie siete v závislosti na prenášanom počte kanálov.

## Zadanie

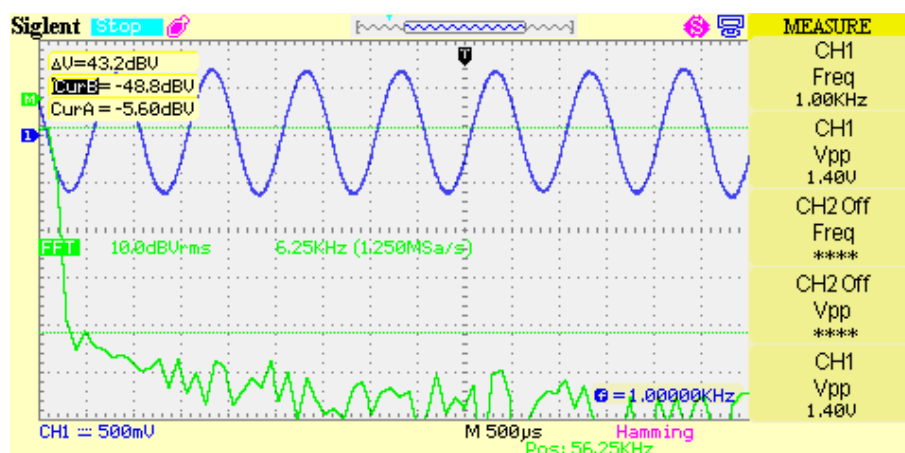
1. Frekvenčné spektrum – porovnanie frekvenčných spektier harmonického signálu na vstupe a výstupe systému.
2. Latencia signálu – čas, za ktorý sa po pripojení signálu na vstup SPU jednotky objaví signál na výstupe externých reproduktorov.
3. Zaťaženie siete – percentuálne vyjadrenie zaťaženia 100 Mb/s siete v závislosti na počte prenášaných kanálov.

## Vypracovanie

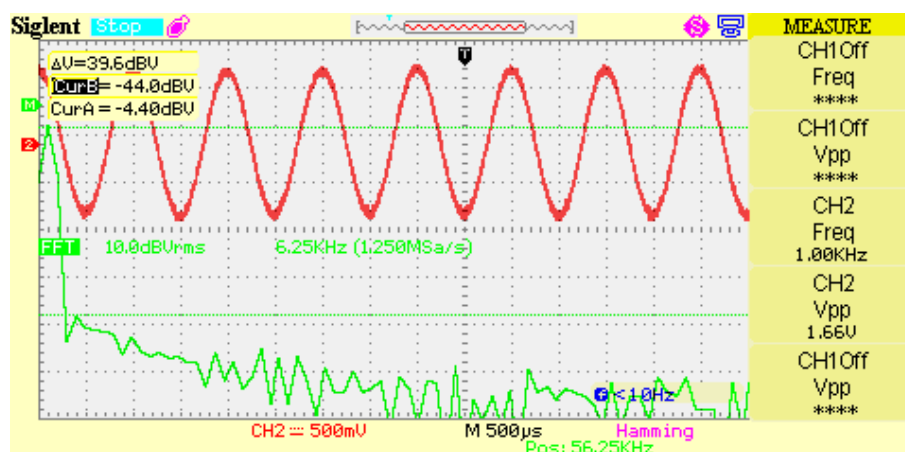
### 1. Frekvenčné spektrum

Na obr. G.1 je vidieť časový priebeh a modul spektra harmonického signálu pripojeného na vstup SPU jednotky. Jedná sa o sínusoidu s frekvenciou 1 kHz. Signál je generovaný zo zvukovej karty osobného počítača a nie je veľmi kvalitný. Odstup signál šum dosahuje hodnotu 43.2 dBV.

Na obr. G.2 je zobrazený časový priebeh a modul spektra signálu na výstupe systému. Odstup signál šum sa zhoršil na 39.6 dBV. To je spôsobené rušením z meniča, ktorým je napájaný DA prevodník v externých reproduktoroch, viď [3].



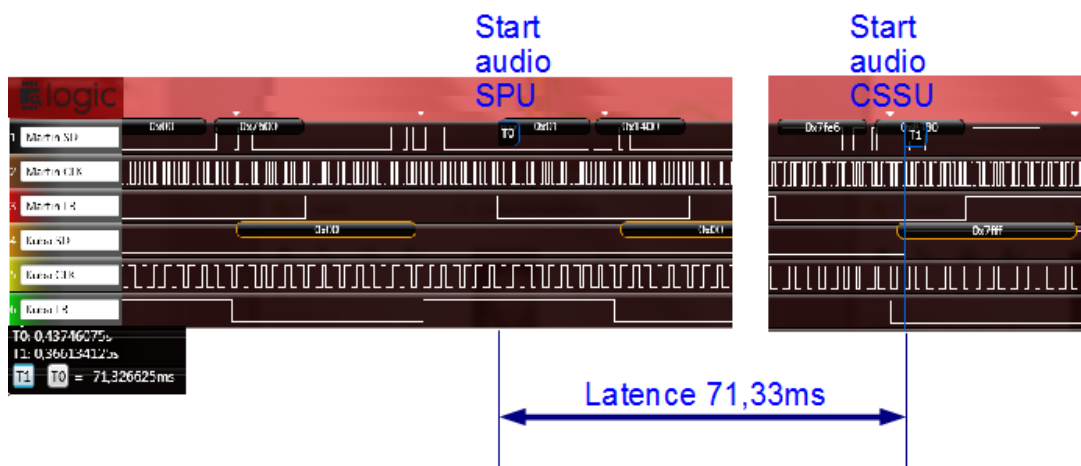
Obr. G.1: Časový priebeh a modul frekvenčného spektra harmonického signálu na vstupe SPU jednotky



Obr. G.2: Časový priebeh a modul frekvenčného spektra harmonického signálu na výstupe DA prevodníka v externých reproduktoroch

## 2. Latencia signálu

Na obr. G.3 je pomocou logického analyzátoru zachytený signál vo formáte I<sup>2</sup>S na výstupe AD prevodníka SPU jednotky a na vstupe DA prevodníka v externých reproduktoroch. Po spustení prehrávania sa začnú meniť posiadané hodnoty. Odmieraná latencia je potom čas, za ktorý sa zmena na vstupe premietne na výstup systému.



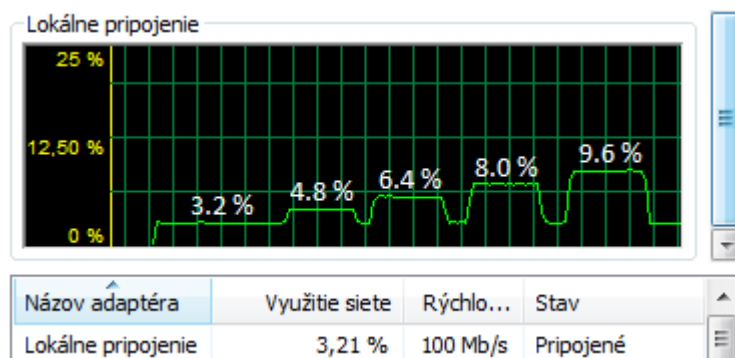
Obr. G.3: Priebehy I<sup>2</sup>S signálov zmerané logickým analyzátorom

Latencia signálu je zapríčinená použitím zásobníkov v SPU jednotke a v externých reproduktoroch. V SPU jednotke vzniká oneskorenie signálu 7 ms. To je doba, za ktorú sa naplní paket veľkosti 1344 B, ktorý sa posiela do externých reproduktorov. V externých reproduktoroch sa nachádza zásobník na 9 paketov. Z toho vyplýva oneskorenie signálu 63 ms [3]. Dohromady je teda latencia signálu celého systému 70 ms.

## 3. Zaťaženie siete

Na obr. G.4 je graf percentuálneho zaťaženia 100 Mb/s siete v závislosti na čase. V čase sa menil počet vysielaných kanálov v rozmedzí 2 až 6. V tab. G.1 sa nachádzajú hodnoty vzťahnuté k počtu kanálov.





Obr. G.4: Zaťaženie siete v závislosti na počtu prenášaných kanálov

Počet kanálov	Zaťaženie 100 Mb/s siete [%]
2	3.2
3	4.8
4	6.4
5	8.0
6	9.6

Tab. G.1: Zaťaženie siete v závislosti na prenášanom počte kanálov

## Použité meracie prístroje

- Osciloskop Siglent SDS 1072CML
- Logický analyzátor Saleae Logic 8-channel

## Záver

V meraní boli určené parametre systému s SPU jednotkou a externými reproduktormi. Odstup signál šum na výstupe externých reproduktorov sa pohyboval na úrovni 39.6 dB. Latencia signálu v rámci celého systému dosahovala hodnotu 71 ms a zaťaženie siete pre šesť prenášaných kanálov dosahovalo 9.6 % zo 100 Mb.

## H BTYPES.H

```
#ifndef BTYPES_H
#define BTYPES_H

#ifdef QT_CORE_LIB
#include <Qt>
#include <QStringList>

typedef quint8      uint8_t;
typedef quint16    uint16_t;
typedef quint32    uint32_t;
typedef qint8      int8_t;
typedef qint16     int16_t;
typedef qint32     int32_t;
typedef float      port_float_t;
typedef quint8     port_bool_t;

#define pFALSE      ((port_bool_t) 0)
#define pTRUE       ((port_bool_t) ~0)

#define to8bit()    toUtf8()
#define from8bit(x)  QString::fromUtf8(x)

typedef enum
{
    Refused, Closed
} CommunicationError_t;
#define COMMUNICATION_ERROR_TABLE \
    trUtf8("Nelze navázat spojení") \
    << trUtf8("Spojení bylo přerušeno")

namespace BaO
{
    typedef enum
    {
        Usb, Ethernet
    } CommunicationInterface_t;

```

```

#define COMMINTERFACELIST "USB" << "ETHERNET"

#define INFRA_COMMANDS_LIST "Volume_UP" \
    << "Volume_DOWN" << "Mute" \
    << "Device_1_asym" << "Device_2" << "Device_1_sym"
#define SIDE_LIST_QSTRING trUtf8("Center") \
    << trUtf8("Subwoofer") \
    << trUtf8("Levý_zadní") \
    << trUtf8("Levý_přední") << trUtf8("Pravý_zadní") \
    << trUtf8("Pravý_přední") << trUtf8("nic")

#define SIDE_LIST_SHORT_QSTRING \
    trUtf8("Cen") << trUtf8("Sub") \
    << trUtf8("LZ") << trUtf8("LP") << trUtf8("PZ") \
    << trUtf8("PP") << trUtf8("nic")

#define BAND_LIST "60Hz" << "150Hz" \
    << "400Hz" << "1KHz" \
    << "2.4KHz" << "6KHz" << "15KHz"

#define TEMPERATURE_NAMES_LIST trUtf8("satelit") \
    << trUtf8("zesilovač") \
    << trUtf8("další_zesilovač") \
    << trUtf8("zeslabovač") << trUtf8("konec")

#if (QT_VERSION < 0x050000)
    #define setSectionSizeMode setSizeMode
#endif
}

#define INI_FILE "bang.ini"

#else
#include "ch.h"
#include "hal.h"

typedef float      port_float_t;

```

```

typedef uint8_t      port_bool_t;

#endif

#ifdef QT_CORE_LIB
namespace BaO
{
#endif

typedef enum {ANALOG_UP_1, ANALOG_DW_1, MUTE, CIFFER_1,
             CIFFER_2, CIFFER_3} remoteCommand_t;

#ifdef QT_CORE_LIB
}
#endif

#define CONFIGURATION_TCP_PORT      15152
#define UDP_PORT                    25001
#define SAMPLE_COUNT                336
#define UDP_HEADER_LENGTH           16
#define PACKET_LENGTH                \
    ((SAMPLE_COUNT * 4) + UDP_HEADER_LENGTH)

#define GET_MASTER                   "GET_MASTER"
#define GET_SLAVE_COUNT              "GET_SLAVE_COUNT"
#define GET_SLAVE                    "GET_SLAVEN"
#define SEND_GPIO                    "SEND_GPIO"
// asynchronní – posiláš podle sebe
#define SEND_TEMPERATURE              "SEND_TEMP"
// SERIAL4bytyTEPLOTY2byty*počet
#define SEND_OUTPUTSTREAM             "SEND_OUTSTREAM"
#define SEND_INFRACOMMAND             "SEND_INFRACOMMAND"
// SERIAL4bytyCOMMAND2byty
#define SEND_SLAVE_EQUALISER          "SEND_EQUALISER"
// SERIAL4bytyGLOBAL1byteENABLED1byteSIZEOF(Bequaliser)
#define SET_OUTPUTSTREAM              "SET_OUTPUTSTREAM"
#define SET_INPUTSTREAM               "SET_INPUTSTREAM"
#define SET_GPIO                      "SET_GPIO"
#define SET_TEMPERATURE               "SET_TEMPERATURE"

```

```

#define SET_SLAVE_EQUALISER           "SET_EQUALISER"
// SERIAL4bytyGLOBAL1byteENABLED1byteSIZEOF( bequaliser )

//pouzije slave...

#define GIVE_INPUTSTREAM              "GIVE_INPUTSTREAM"
#define GIVE_STOPINPUTSTREAM "GIVE_STOPINPUTSTREAM"
#define GIVE_OUTPUTSTREAM            "GIVE_OUTPUTSTREAM"
#define GIVE_STOPOUTPUTSTREAM "GIVE_STOPOUTPUTSTREAM"

#define SET_SLAVENAME                 "SET_SLAVENAME"
// SERIAL4bytyNAME16bytu
#define SET_SLAVEINPUT                "SET_SLAVEINPUT"
// SERIAL4bytyUID2byty

#define SEND_SLAVEREFRESH            "SEND_SLAVEREFRESH"
//SERIAL4bytyCISLOuTebeVPolu4byty

#define EQUALISER_COEF_COUNT          7
#define TEMPERATURECOUNT            5

typedef enum
{
    NO_SIDE = 6, SIDE_CENTER = 0, SIDE_SUBWOOFER =1,
    LEFT_REAR = 2, LEFT_FRONT = 3, RIGHT_REAR = 4,
    RIGHT_FRONT = 5 , SIDE_ENUM_LAST
} side_enum_t;

typedef char name_t [16];

/*****
 * input stream
 *****/
struct BInputStream_t
{
    // tihle jenom vynasobit data z A/D
    port_float_t Gain; //readwrite
    // má vzorkovat nebo ne
    port_bool_t Active; //readwrite

```

```

    /// fixní nastaviš jenom na začátku programu
    uint16_t UID;                                //readonly
    /// nadřazený balík vstupních streamů
    ///(např CD bude mít číslo 100)
    uint16_t DeviceUID;                          //readwrite
    /// uživatelský název vstupu (např CD left)
    name_t name;                                 //readwrite in PC
    /// uživatelský název balíku vstupů (např CD)
    name_t DeviceName;
    /// side_enum_t
    uint8_t Side;
}  __attribute__((__packed__));

```

```

typedef struct BInputStream_t BInputStream_t;

```

```

/*****
 * output stream
 *****/
struct BEqualiser_t
{
    uint8_t Coef[EQUALISER_COEF_COUNT]; //readwrite
    port_bool_t Active;                 //readwrite
}  __attribute__((__packed__));

```

```

typedef struct BEqualiser_t BEqualiser_t;

```

```

/// nastavení výstupu může pro každý slave jinak
/// slave – celý systém pro jednu místnost (např 5+1)
/// output stream = reprák
struct BoutputStream_t
{
    BEqualiser_t Equaliser;              //readwrite

    port_bool_t Mute;                    //readwrite
    port_float_t Volume;                 //readwrite
    /// připojený vstup z A/D do konkrétního repráku
    uint16_t InputStreamConnectionUID;  //readwrite
    /// nadřazená jednotka v místnosti
    ///(hlavní jednotka – subwoofer)

```

```

    uint32_t SlaveSerialNumber;           //readOnly
    /// tohle nastaví PC – nezajímá tě
    uint8_t Side;                         //readOnly
    /// fixní identifikátor reproduktoru (např 100–103)
    uint16_t UID;                          //readonly
} __attribute__((__packed__));

typedef struct BoutputStream_t BoutputStream_t;

/*****
* Slave
*****/
struct BSlave_t
{
    uint32_t IP;                           //readonly in PC
    uint32_t SerialNumber;                 //readonly
    uint32_t FW_Version;                   //readonly
    uint32_t HW_Version;                   //readonly
    ///readonly in PC
    int16_t Temperatures [TEMPERATURECOUNT];
    uint16_t GPIO_In;                      //readonly in PC
    uint16_t GPIO_Out;                     //readwrite
    ///30
    name_t name;                           //readwrite in PC
    /// vstupní připojené zařízení
    uint16_t ConnectDeviceUID;

    BEqualiser_t Equaliser;                //readwrite
    ///výběr jesli použít lokální nebo globální eq
    port_bool_t useGlobalEq;
    ///globálně vypne nebo zapne equalizéry
    port_bool_t EqualisersEnabled;

    /// seznam výstupních streamů
    BoutputStream_t ** Outputs; //readonly in PC
    uint16_t OutputCount;        //readonly in PC
} __attribute__((__packed__));

typedef struct BSlave_t BSlave_t;

```

```

#ifdef QT_CORE_LIB
struct BSlaves_t
{
    uint16_t Count;
    BSlave_t ** Slaves;
} --attribute-- ((--packed--));

typedef struct BSlaves_t BSlaves_t;
#endif

/*****
* Master
*****/
struct BMaster_t
{
    uint32_t IP;           //readonly in PC
    uint32_t SerialNumber; //readonly
    uint32_t FW_Version;  //readonly
    uint32_t HW_Version;  //readonly
} --attribute-- ((--packed--));

typedef struct BMaster_t BMaster_t;

/*****
* Main
*****/
typedef struct
{
    BMaster_t * Master;
#ifndef QT_CORE_LIB
    BSlave_t ** Slaves;
    // ještě si sem přihodit počet slave jednotek
    uint8_t SlaveCount;
#else
    BSlaves_t * Slaves;
#endif

    BInputStream_t ** InputStreams;

```



```

uint16_t InputStreamCount;

BoutputStream_t ** OutputStreams;
uint16_t OutputStreamCount;

} BMain_t;

#define MEMBER_SIZE(type , member) \
    (sizeof(((type *)0)->member))

/*****
 * Plnění UDP hlavičky
 *****/
/*****
 * 0xFF FF FF F(F/E vstup/vystup) | 2byty INPUT UID |
 * 2byty OUTPUT UID | 4byty slave serial number |
 * 4byty timemarker
 *****/

#define _UDP_FILL(ptr , data , offset , size) \
    {memcpy(ptr + offset ,&data , size);}
#define _UDP_EXTRACT(ptr , retVal , offset , size) \
    {memcpy(&retVal , ptr+offset , size);}
#define TIME_MARKER_SIZE 4

#define STREAM_TYPE_INPUT 1
#define STREAM_TYPE_OUTPUT 0

//pokud se bude posila input stream tak bych chtěl
//obě uid stejny
//nejde pouzit s konstantou jako argument ale musi
//to byt promenna
#define UDP_FILL_PREAMBLE(ptr) \
    {memset(ptr ,0 xff ,3); ptr[3] |= 0xFE;}
#define UDP_FILL_STREAM_TYPE(ptr , type0vystup1vstup) \
    {ptr[3] &= 0xFE ;ptr[3] |= \
    (type0vystup1vstup & 1);}
#define UDP_FILL_INPUT_UID(ptr , uid) \

```

```

        {_UDP_FILL(ptr , uid , 4 , \
        MEMBER_SIZE( BInputStream_t , UID))}
#define UDP_FILL_OUTPUT_UID(ptr , uid) \
        {_UDP_FILL(ptr , uid , 6 , \
        MEMBER_SIZE( BoutputStream_t , UID))}
#define UDP_FILL_SERIAL_NUMBER(ptr , number) \
        {_UDP_FILL(ptr , number , 8 , \
        MEMBER_SIZE( BSlave_t , SerialNumber))}
#define UDP_FILL_TIME_MARKER(ptr , marker) \
        {_UDP_FILL(ptr , marker , 12 , TIME_MARKER_SIZE)}

#define UDP_EXTRACT_STREAM_TYPE(ptr , retVal8) \
        {retVal8 = ptr[3] & 0x1;}
#define UDP_EXTRACT_INPUT_UID(ptr , retVal16) \
        {_UDP_EXTRACT(ptr , retVal16 , 4 , \
        MEMBER_SIZE( BInputStream_t , UID))}
#define UDP_EXTRACT_OUTPUT_UID(ptr , retVal16) \
        {_UDP_EXTRACT(ptr , retVal16 , 6 , \
        MEMBER_SIZE( BoutputStream_t , UID))}
#define UDP_EXTRACT_SERIAL_NUMBER(ptr , retVal32) \
        {_UDP_EXTRACT(ptr , retVal32 , 8 , \
        MEMBER_SIZE( BSlave_t , SerialNumber))}
#define UDP_EXTRACT_TIME_MARKER(ptr , retVal32) \
        {_UDP_EXTRACT(ptr , retVal32 , 12 , TIME_MARKER_SIZE)}

#endif // BTYPES_H

```

# **I PRÍLOHY NA CD**

**I.1 Dokumentácia**

**I.2 Schémy SPU**

**I.3 Plošné spoje SPU**

**I.4 Podporné programy v SW MATLAB**

**I.5 Firmware SPU**

**I.6 Fotodokumentácia**