

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

HLUKOMĚR PRO EMBEDDED SYSTÉMY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ STEJSKAL

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

HLUKOMĚŘ PRO EMBEDDED SYSTÉMY

SOUND LEVEL METER FOR EMBEDDED SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ STEJSKAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MIROSLAV BOTTA

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Tomáš Stejskal

ID: 134615

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Hlukoměr pro embedded systémy

POKYNY PRO VYPRACOVÁNÍ:

Realizujte jednoduchý hlukoměr pro širokopásmové měření hladiny hluku s nastavitelnou časovou konstantou za pomoci mikrokontroléru typu ARM. Úlohou studenta v průběhu řešení diplomové práce bude návrh a vytvoření hardwaru čidla pro měření hluku s ohledem na nízkou spotřebu. K realizaci bude využit vývojový kit STM32F4 Discovery, pro který budou také vytvořeny nutné ovladače. Výsledkem práce bude sestava, chovající se jako digitální hlukoměrné čidlo.

DOPORUČENÁ LITERATURA:

[1] NOVÝ, Richard. Hluk a chvění. 3. vyd. V Praze: České vysoké učení technické, 2005, 141 s. ISBN 978-80-01-04347-9.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: Ing. Miroslav Botta

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem této práce je návrh a implementace hlukoměru pro emdedded systémy. Je vytvořeno hlukoměrové čidlo. Toto čidlo obsahuje mikrofon, mikrofonní předzesilovač a ADC převodník. Čidlo je připojeno k vývojovému kitu STM32F4 Discovery, kde je zvuk zpracován. Je provedena jeho filtrace, časové váhování, výpočet hladiny hluku a provedeno váhování dle vnímání hluku. Výsledná hladina hluku je poté odeslána pomocí seriové komunikace USART. Práce také obsahuje teoretický rozbor hluku a jeho vnímání člověkem. Je popsána také použitá vývojová platforma.

KLÍČOVÁ SLOVA

měření hluku, hlukoměr, hlukoměrové čidlo, mikrokontrolér, ARM Cortex®-M4F, ST® STM32F4 Discovery, ADC převodník, mikrofonní předzesilovač, akustický tlak, časové váhování, IIR Lattice, DSP, hladina hluku.

ABSTRACT

The aim of this work is the design and implementation of a sound level meter for emdedded systems. It is designed sound level meter sensor. This sensor includes a microphone, microphone preamplifier and ADC. This sensor is connected to a development kit STM32F4 Discovery, where sound is processed. It is processed filtration, time weighting, calculation of sound leve and loudness weighting. The sound level is then sent via the serial communication USART. This thesis includes a theoretical analysis of noise and its human perceptions. It also describes the development platform used.

KEYWORDS

noise measurement, sound level meter, sound level meter sensor, microcontroller, ARM Cortex®-M4F, ST® STM32F4 Discovery, ADC, microphone preamplifier, acoustic pressure, time weighting, IIR Lattice, DSP, sound level

STEJSKAL, Tomáš *Hlukoměr pro embedded systémy*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 73 s. Vedoucí práce byl Ing. Miroslav Botta

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Hlukoměr pro embedded systémy“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Miroslavu Bottovy za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	12
1 Teoretický úvod	13
1.1 Měření hluku	13
1.1.1 Zvuk	13
1.1.2 Šíření zvuku	13
1.1.3 Akustický tlak	14
1.1.4 Vlastnosti lidského ucha	15
1.1.5 Akustický tlak v kmitočtových pásmech	17
1.1.6 Váhové filtry	18
1.2 Číslicové filtry	18
1.2.1 IIR filtry	18
1.3 Vývojová platforma	21
2 Řešení studentské práce	23
2.1 Hlukoměrné čidlo	24
2.1.1 Mikrofon a mikrofonní předzesilovač	24
2.1.2 ADC převodník a jeho pomocné obvody	26
2.1.3 Digitální senzory teploty, tlaku a vlhkosti	30
2.1.4 DPS hlukoměrného čidla	31
2.2 Návrh IIR filtrů	33
2.2.1 IIR filtry s využitím DSP instrukcí	34
2.3 Zpracování vstupních dat v MCU	35
2.3.1 Vývojové prostředí MCU	35
2.3.2 Komunikace s hlukoměrným čidlem	36
2.3.3 Zpracování dat	38
2.4 Možnosti snížení energetické náročnosti	40
3 Výsledky studentské práce	42
3.1 Výsledný hlukoměr	42
3.1.1 Výstupní komunikace s hlukoměrným čidlem	43
3.1.2 Měření hlukoměru	45
3.2 Zhodnocení hlukoměru	49
4 Závěr	50
Literatura	52

Seznam symbolů, veličin a zkratk	54
Seznam příloh	57
A Příloha	58
A.1 Měření předzesilovače	58
A.2 Schéma DPS hlukoměrového čidla	60
A.3 Seznam součástek	61
A.4 Vývojový diagram ovladače hlukoměru	62
A.5 Návrh IIR filtrů v FDATool	63
A.6 Měření hlukoměru	64
A.7 Instalace vývojového kitu STM32F4 Discovery	67
A.7.1 USB ovladač STM32F4 Discovery	67
A.7.2 Překladač pro STM32F407VG	67
A.7.3 Vývojové prostředí pro STM32F4 Discovery	68
A.7.4 CooCox IDE – Založení nového projekt	68
A.8 Použití hardwarové FPU	69
A.9 Použití DSP knihovny	69
A.10 Připojení hlukoměru	71
A.11 Popis pinů hlukoměru	72
A.12 DVD	72

SEZNAM OBRÁZKŮ

1.1	Šíření zvuku od zdroje ve vlnoplochách [1]	14
1.2	Časový průběh akustického tlaku [2]	15
1.3	Oblast slyšitelnosti [2]	16
1.4	Křivky stejné hlasitosti [2]	16
1.5	Kmitočtové charakteristiky třetino-oktávové banky filtrů [2]	18
1.6	Příklad IIR filtru a jeho impulzní charakteristiky [3]	19
1.7	Struktura Lattice IIR filtru [5]	20
1.8	Vývojový kit STM32F4 Discovery	21
2.1	Blokové schéma vytvořené sestavy hlukoměru	23
2.2	Schéma mikrofonního předzesilovače	24
2.3	Závislost zesílení na kmitočtu (Gain)	26
2.4	Aplikační diagram AD7683 [8]	27
2.5	Schéma zapojení ADC převodníku	27
2.6	SPI sběrnice [11]	28
2.7	Obecný princip SPI komunikace [11]	29
2.8	Časovací diagram ADC převodníku [8]	30
2.9	DPS - obrazec spojů horní vrstvy (vlevo) a spodní vrstvy (vpravo)	32
2.10	DPS - Osazovací plán horní vrstvy (vlevo) a spodní vrstvy (vpravo)	32
2.11	Vyrobená neosazená DPS hlukoměrného čidla, vrstva Top (vlevo) a (Bottom) (vpravo)	33
2.12	Osazená DPS hlukoměrného čidla, vrstva Top (vlevo) a (Bottom) (vpravo)	33
2.13	Zachycená komunikace SPI	36
3.1	Hlukoměrná sestava	42
3.2	Redukce (vlevo) a kabel (vpravo) pro připojení hlukoměrného čidla	42
3.3	Příklad užití příkazu R10@	43
3.4	Měření kalibrační křivky při použití růžového šumu	46
3.5	Kalibrační mikrofon (vlevo) a hlukoměrné čidlo (vpravo)	46
3.6	Měřicí sestava v bezodrazové komoře (2)	47
A.1	Relativní zesílení vztažené k referenční hodnotě 1kHz (Relative Level (1kHz))	58
A.2	Celkové Harmonické zkreslení (Total harmonic distortion – THD)	58
A.3	Celkové harmonické zkreslení zahrnující šum (Total harmonic distortion plus noise – THD+N)	59
A.4	Harmonické zkreslení druhou až desátou harmonickou složkou (Distortion Product Ratio)	59
A.5	Schéma DPS hlukoměrového čidla	60

A.6	Vývojový diagram ovladače hlukoměru	62
A.7	Návrh IIR filtru v programu FDATool	63
A.8	Modulová kmitočtová závislost citlivosti hlukoměru při konstatním akustickém tlaku 74 dB _{SPL}	64
A.9	Použité měřicí přístroje, analyzátor (vlevo) a generátor zvuku (vpravo)	64
A.10	Laboratoř Studiové a hudební elektroniky ve které probíhalo měření .	65
A.11	Měřicí sestava v bezodrazové komoře (1)	65
A.12	Měřicí sestava v bezodrazové komoře (3)	66
A.13	Připojení hlukoměrového čidla pomocí kabelu	71
A.14	Připojení hlukoměrového čidla pomocí redukce	71
A.15	Popis pinů hlukoměru	72

SEZNAM TABULEK

1.1	Kmitočty oktávových pásem	17
1.2	Parametry STM32F4 Discovery [6]	22
2.1	Celkové změřené hodnoty předzesilovače	26
2.2	Parametry senzoru SHT21 [12]	31
2.3	Parametry senzoru MS5637-02BA03 [13]	31
3.1	Seznam použitých zařízení	47
3.2	Měření proudového odběru hlukoměrových komponent	48
3.3	Porovnání rychlosti zpracování dat s FPU	48
A.1	Seznam použitých součástek hlukoměrného čidla	61
A.2	Tabulka pinů hlukoměru	72

ÚVOD

V rámci této práce je navržen a vytvořen prototyp hlukoměru, který umožňuje širokopásmové měření hluku. Důvodem této práce bylo vytvoření prototypu hlukoměru pro embedded systémy, který bude jednoduchý, ale zároveň dostatečně přesný a bude navržen jako nízko-spotřebový.

Hlukoměr má nastavitelnou časovou konstantu pro různé typy měření (krátkodobé, dlouhodobé, . . .) a nastavitelné váhovací filtry, které umožňují korekci měřeného hluku podle toho, jak je vnímán člověkem.

Byl navržen a realizován hardware samostatného prototypu hlukoměrného čidla, který provádí převod střídavého akustického tlaku na elektrický signál pomocí elektretového mikrofону. Tento signál je zesílen a poté vzorkován ADC převodníkem, který výsledné vzorky signálu odesílá pomocí protokolu SPI. Tento protokol byl zvolen z důvodu jednoduchosti a tím i efektivnosti.

Následné zpracování vzorků signálu je provedeno pomocí mikrokontroléru STM32F407VGT6, umístěném na vývojovém kitu STM32F4 Discovery od firmy STMicroelectronics. Dochází zde k filtraci dat pomocí oktákové banky filtrů, která rozdělí měřený hluk do jednotlivých pásem podle jeho frekvence. Poté je provedeno časové váhování, pomocí kterého je získána efektivní hodnota signálu v jednotlivých pásmech, z které je poté spočítána hladina hluku.

V teoretické části je popsáno šíření zvuku v prostoru, principy jeho měření, vlastnosti lidského ucha a akustický tlak. Poté jsou popsány filtry, které byly využity k realizaci hlukoměru.

V části „Řešení studentské práce“ je popsán postup návrhu prototypu hlukoměrného čidla, číslicových filtrů, použité komunikace SPI a využití DSP instrukcí. Dále je popsáno vývojové prostředí a energeticky úsporné módy mikrokontroléru a jejich parametry.

Ve výsledcích práce je popsána výsledná hlukoměrná sestava. Jsou popsány výsledky měření parametrů předzesilovače, výsledného hlukoměru, energetické náročnosti, rychlosti zpracování dat a celkové zhodnocení celého systému.

1 TEORETICKÝ ÚVOD

V teoretické části semestrálního projektu je popsán princip měření hluku. Jsou popsány jeho základní vlastnosti a principy měření. Poté je popsán vybraný mikrokontrolér, který byl použit k řešení projektu. Dále je popsána komunikace podle standardu SPI a číslicové filtry typu IIR.

1.1 Měření hluku

V této kapitole je popsána teorie zvuku. Je popsána jeho definice, šíření, akustický tlak, vnímání hluku lidským sluchem a jeho rozklad do kmitočtových pásem.

1.1.1 Zvuk

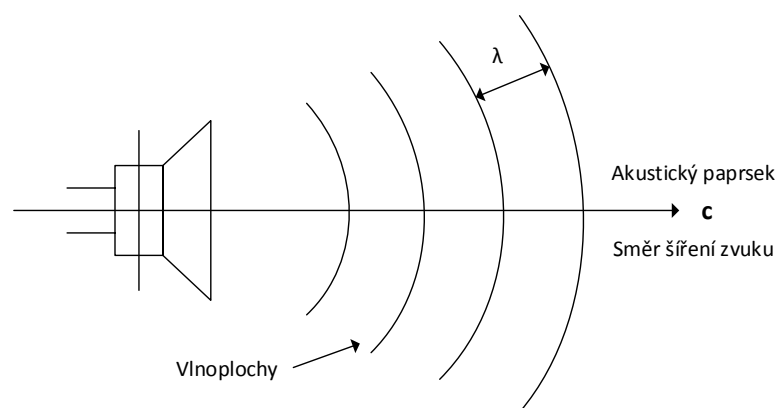
Podstatou zvuku je mechanické kmitání pružného prostředí. V případě slyšitelného zvuku jde o frekvenční rozsah 20-20000 Hz šířících se konečnou rychlostí daným prostředím. Akustická vlna ve vzduchu má rychlost 340 m/s. Daný frekvenční rozsah je dán frekvenčním rozsahem lidského ucha. Akustika se zabývá i akustickým kmitáním v jiných pásmech. Tato pásma se dělí na tři části, a jde o část infrazvukovou, slyšitelnou a ultrazvukovou. Tato práce se zajímá pouze o slyšitelnou část spektra [1].

1.1.2 Šíření zvuku

Zvuk se šíří v plynech, kapalinách a pevných látkách. V homogenním izotropním prostředí se šíří vlny přímočaře. Vlnění se dělí na příčné nebo podélné, podle toho, zda částice kmitají ve směru šíření vlny nebo kolmo na ni. U podélného vlnění je tedy směr vlnění jednoznačně dán směrem šíření vlnění, u příčného se musí také udávat rovina, ve které kmitání probíhá. U plynů a kapalin se vyskytuje pouze podélné akustické vlnění. Toto vlnění se šíří prostředím ve vlnoplochách směrem od zdroje. Vlnoplocha má takovou vlastnost, že ve všech bodech je v daném okamžiku stejný akustický stav. Kolmice na vlnoplochu se nazývá akustický paprsek. Akustické vlnění je zobrazeno na obr.1.1 [1]

Označíme-li periodu zvukového vlnění T . Je frekvence zvukového vlnění $f = 1/T$. Zvuková vlna bude tedy mít vlnovou délku:

$$\lambda = c_0 T = \frac{c_0}{f} = 2\pi \frac{c_0}{\omega} \quad [m], \quad (1.1)$$



Obr. 1.1: Šíření zvuku od zdroje ve vlnoplochách [1]

kde ω je úhlový kmitočet vlnění. Vlnová délka odpovídá vzdálenosti mezi jednotlivými vlnoplochami kmitající se stejnou frekvencí. Pro frekvence 20 – 20000 Hz, které lidské ucho slyší, odpovídají vlnové délky od 17,15 m do 17,15 mm.

1.1.3 Akustický tlak

Tlak, jako skalární veličina je definován jako síla působící na jednotku plochy:

$$p(t) = \frac{F(t)}{S} \text{ [Pa]}. \quad (1.2)$$

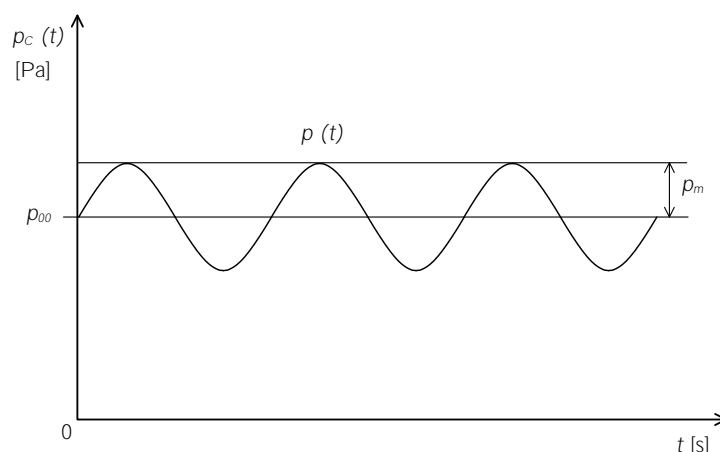
Součet akustického tlaku $p(t)$ a středního barometrického tlaku p_b tvoří celkový statický tlak p_c . V podstatě lze říci, že je akustický tlak namodulován na celkový statický tlak viz obr.1.2. Akustický tlak i v maximálních hodnotách je běžně o cca. tři řády nižší než barometrický tlak.

Z důvodu toho, jak lidské ucho vnímá zvuk, používá se pro vyjádření akustických veličin používá logaritmická stupnice. Lidské ucho totiž vnímá zvuky, které rostou geometrickou řadou, pouze jako přírůstky. Proto se jednotlivé veličiny v akustice vyjadřují jako hladiny.

Obecně lze hladinu libovolné veličiny vyjádřit jako:

$$L = \log \frac{x}{x_0} \text{ [Bel]}, \quad (1.3)$$

kde x_0 je vtažná hodnota k veličině x . Jednotkou je [Bel], avšak v praxi se užívá



Obr. 1.2: Časový průběh akustického tlaku [2]

„jemnější“ stupnice v jednotkách deci Bel [dB], která je vyjádřena vztahem:

$$L = 10 \log \frac{x}{x_0} \text{ [dB]}. \quad (1.4)$$

Při měření hluku se využívá efektivní hodnota akustického tlaku. Označuje se jako *hladina akustického tlaku* L (*Sound Pressure Level – SPL*) a je definována jako:

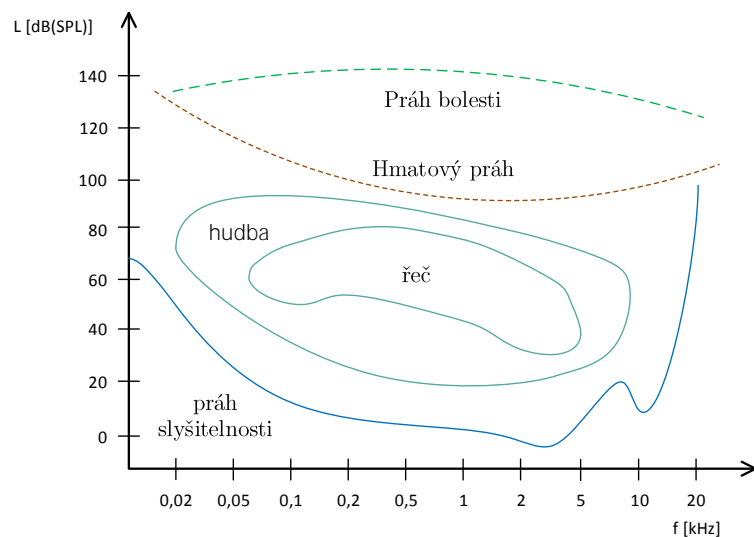
$$L_p = 10 \log \frac{p^2}{p_0^2} = 20 \log \frac{p}{p_0} \text{ [dB}_{\text{SPL}}], \quad (1.5)$$

kde vztažnou hodnotou je p_0 , která odpovídá prahu slyšitelnosti při kmitočtu 1 kHz a je rovna $2 \cdot 10^{-5}$ Pa. Jednotka hladiny akustického tlaku se označuje jako SPL, která má v dolní indexu nebo v závorce uvedenu jednotku dB – dB_{SPL}, dB(SPL), dB_(SPL). Důvodem je zdůraznění, že se jedná o hladinu akustického tlaku vztaženou k referenční hodnotě $2 \cdot 10^{-5}$ Pa [2].

1.1.4 Vlastnosti lidského ucha

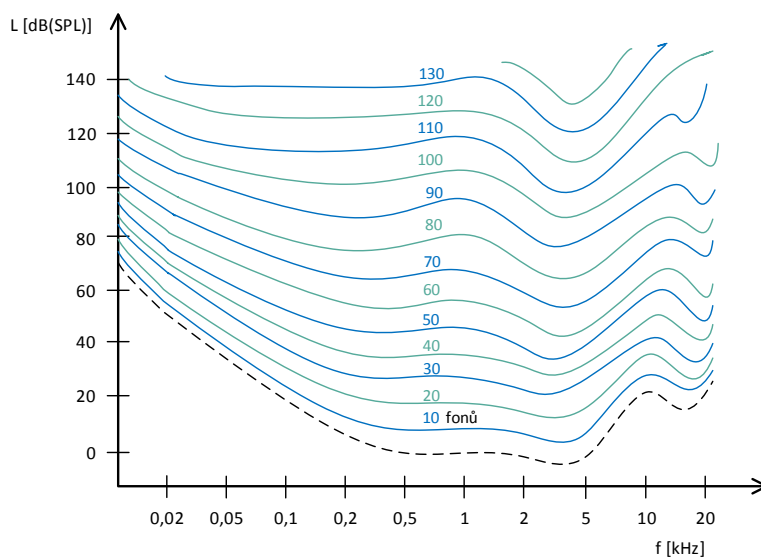
Lidské ucho vnímá zvuk v oblasti 20 Hz až 20 kHz. K tomu aby byl zvuk pro lidské ucho slyšitelný, musí jeho intenzita (hladina) překročit prahovou hodnotu vnímání. Tato hodnota je závislá na kmitočtu, proto se pro jednotlivé kmitočty hodnota prahu slyšitelnosti mění. Referenční hodnota tlaku ve vztahu (1.5) je dána prahem slyšitelnosti na frekvenci 1 kHz. Při hladinách zvuku 130 až 140 dB už vzniká pocit

bolesti, proto se tato oblast nazývá prahem bolesti. Vše ostatní mezi těmito dvěma prahy je tak oblastí slyšitelnosti viz obr.1.3 [1].



Obr. 1.3: Oblast slyšitelnosti [2]

Zvuky o stejné intenzitě jsou na různých kmitočtech vnímány různě. Nejvíce jsou vnímány zvuky v oblasti 2 až 3 kHz jak lze vidět na obr.1.4.



Obr. 1.4: Křivky stejné hlasitosti [2]

Tyto křivky se označují jako *křivky stejné hlasitosti* (Kingsburyho křivky, Fletcherovy-Munsonovy křivky). V tomto případě se jako jednotka hlasitosti používá jednotka fon [Ph], která vyjadřuje subjektivní vnímání změny hlasitosti [2].

1.1.5 Akustický tlak v kmitočtových pásmech

S ohledem na závislost vnímání hlasitosti zvuku na kmitočtu, se při měření hladiny zvuku měří jednotlivé hodnoty akustického tlaku v určitých kmitočtových pásmech. Zvukový signál je zaznamenán mikrofonom a poté je filtrován bankou kmitočtových filtrů typu pásmová propust. Poté jsou vypočteny jednotlivé hladiny akustického tlaku pro jednotlivé filtry podle:

$$L_{\Delta f} = 20 \log \frac{p_{\Delta f}}{p_0} \quad [\text{dB}_{\text{SPL}}]. \quad (1.6)$$

Při měřeních se používají oktákové a třetino-oktákové banky filtrů. Každý filtr je definován středním kmitočtem f_S a šířkou pásma B . Jejich velikost pro každý další filtr roste logaritmicky dle vzorce

$$f_{S_{i+1}} = \sqrt[n]{2} f_{S_i}, \quad B_{i+1} = \sqrt[n]{2} B_i, \quad f_{D_i} = f_{S_i} / \sqrt[n]{2}, \quad f_{H_i} = f_{S_i} \sqrt[n]{2} \quad [\text{Hz}]. \quad (1.7)$$

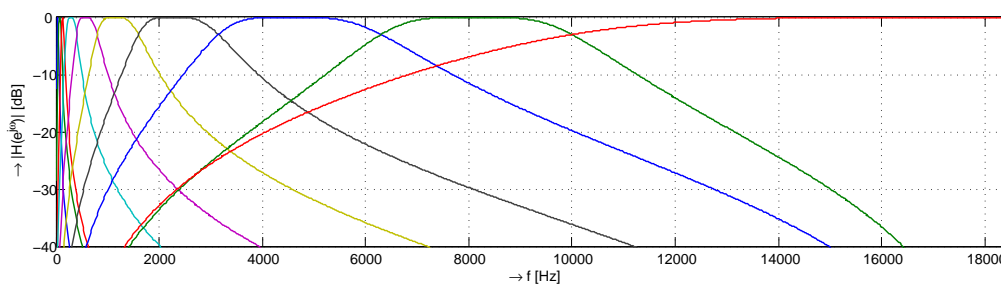
Střední kmitočty filtrů pro oktákovou banku jsou uvedeny v prostředním řádku v tab.1.1. Střední kmitočty pro třetino-oktákovou jsou všechny hodnoty uvedené v tabulce.

Tab. 1.1: Kmitočty oktákových pásem

f_H [Hz]	25	50	100	200	400	800	1600	3150	6300	12500
f_S [Hz]	31,5	63	125	250	500	1000	2000	4000	8000	16000
f_D [Hz]	40	80	160	315	630	1250	2500	5000	10000	20000

Na obr.1.5 jsou zobrazeny jednotlivé kmitočtové charakteristiky oktákové banky filtrů.

Tyto charakteristiky jsou výstupem z programu Matlab. Byly navrženy jako IIR filtry šestého řádu s využitím Butterworthovy aproximace [2]. Tato analýza se nazývá *analýza v kmitočtových pásmech s konstantní relativní šířkou* (Constant Percentage Bandwidth, CPB)



Obr. 1.5: Kmitočtové charakteristiky třetino-oktávové banky filtrů [2]

1.1.6 Váhové filtry

Jak bylo zmíněno v kapitole 1.1.4, vnímání hlasitosti zvuku je závislé na kmitočtu zvuku, proto je třeba výstupy z banky filtrů korigovat podle váhovacích filtrů. Používají se k tomu tři typy filtrů:

- Váhový filtr typu A: 40 Ph
- Váhový filtr typu B: 70 Ph
- Váhový filtr typu C: 100 Ph

Dříve se používal i filtr typu D. Jsou to vlastně korekční křivky, které se sčítají s výstupními hodnotami jednotlivých filtrů, které zvukoměr změřil. V praxi se nejvíce využívají filtry A a C. Po filtraci tímto filtrem, se jednotka měřené veličiny označuje podle typu použitého filtru (A filtr – dBA).

1.2 Číslicové filtry

Při tvorbě hlukoměru byla vytvořena oktávová banka filtrů. Byly použity filtry typu IIR z důvodu jejich menší výpočetní náročnosti oproti filtrům typu FIR.

1.2.1 IIR filtry

IIR filtry jsou filtry s nekonečnou impulzní charakteristikou. Vznikli jako číslicová obdoba analogových filtrů. Mají rekurzivní realizaci, tzn. že mají nelineární fázovou charakteristiku.

Přenos je dán podílem polynomů. Nulové body realizují nerekurzivní část, póly pak rekurzivní část. Řád filtru udává nejvyšší ze stupňů polynomu. Pokud jsou póly

uvnitř kružnice, tak je filtr stabilní [3]. Výstup lineárního filtru je definován podle:

$$y[n] = \sum_{l=1}^N a_l x[n-l] - \sum_{k=0}^M b_k y[n-k], \quad (1.8)$$

kde :

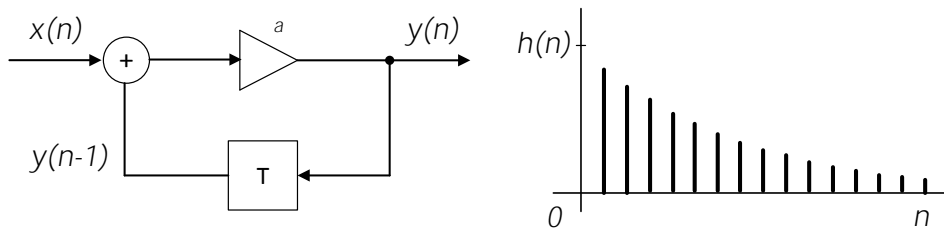
$y[n]$... výstup lineárního filtru,

a_i ... systémové koeficienty v dopředných vazbách,

b_i ... systémové koeficienty ve zpětných vazbách,

N ... počet zpožďovacích členů v nerekurzivní části systému,

M ... počet zpožďovacích členů rekurzivní části.



Obr. 1.6: Příklad IIR filtru a jeho impulzní charakteristiky [3]

Přenos IIR filtru je:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}. \quad (1.9)$$

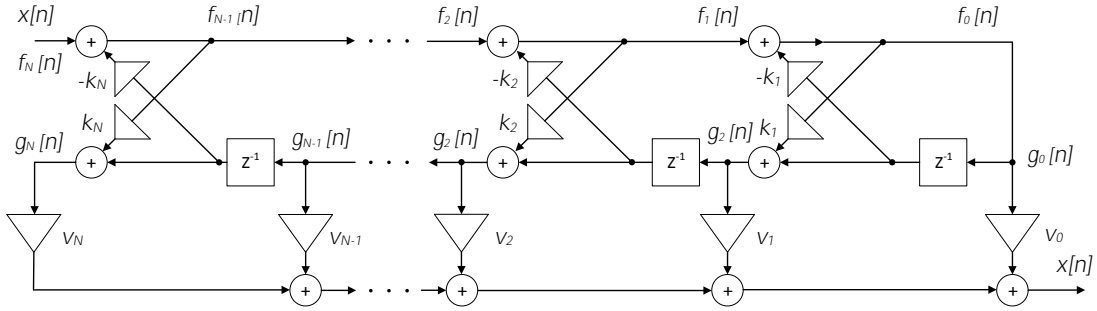
Realizace filtrů mají více možností:

- přímá forma
- kaskádní forma
- paralelní forma
- lattice forma

Při realizaci banky filtrů v této práci byla použita lattice forma a to kvůli nejlepší výpočetní výkonosti při použití DSP instrukcí. Její návrh je popsán v části řešení studentské práce.

Lattice forma IIR filtru

Tato forma filtru je založena na metodě Gray and Markel. Tato forma má obtížnější parametrizaci než ostatní struktury viz. 1.7 . Stavová reprezentace filtru poskytuje informace o struktuře jednotlivých matic [4].



Obr. 1.7: Struktura Lattice IIR filtru [5]

Tento filtr je popsán rovnicemi:

$$f_N(n) = x(n) \quad (1.10)$$

$$f_{m-1}(n) = (f_m(n) - k_m) \cdot g_{m-1}(n-1) \quad (1.11)$$

$$g_m(n) = k_m \cdot (f_{m-1}(n) + g_{m-1}(n)) \quad (1.12)$$

$$y(n) = (v_N \cdot g_N(n)) + (v_{N-1} \cdot g_{N-1}(n)) + \dots + (v_0 \cdot g_0(n)) \quad (1.13)$$

kde:

$\{k_N, k_{N-1}, \dots, k_1\}$... Reflexní koeficienty

$\{v_N, v_{N-1}, \dots, v_0\}$... Ladder koeficienty

Vyšetřit stabilitu filtru je snadné. Pokud jsou všechny koeficienty filtru menší než 1, je filtr stabilní.

1.3 Vývojová platforma

Pro realizaci hlukoměru byl použit vývojový kit STM32F4 Discovery od firmy ST[®] s integrovaným debuggerem ST-LINK/V2 [6]. Důvodem výběru této desky byl dobrý poměr cena/výkon. Vývojový kit je na obr.1.8.



Obr. 1.8: Vývojový kit STM32F4 Discovery

Na vývojovém kitu je umístěn mikrokontrolér (dále jen MCU) STM32F407VGT6 s 32-bitovým procesorem ARM Cortex[®]-M4F. MCU dále obsahuje hardwarovou FPU (Floating Point Unit – jednotka pro práci s čísly s plovoucí desetinnou čárkou), která umožňuje rychlejší operace s datovým typem float a kontrolér DMA, který významně šetří čas procesoru. MCU má 1 MB flash paměti, 192 KB operační paměti RAM a je v pouzdru LQFP100. Deska je napájena pomocí USB z počítače nebo z jiného 5 V zdroje. Deska dále obsahuje pomocné periferie, jako MEMS akcelerometr, audio senzor, audio DAC, atd. Pro vývoj jsou vyvedeny všechny piny mikrokontroleru. Další informace jsou v tab.1.2

Zdrojem hodinového signálu pro MCU je buď interní 16 MHz RC oscilátor nebo 8 MHz externí krystalový oscilátor. Spotřeba MCU je udávána typicky 87 mA/168 MHz. Dále má integrovanou kompletní sadu DSP instrukcí, které byly využity při tvorbě ovladačů.

Tab. 1.2: Parametry STM32F4 Discovery [6]

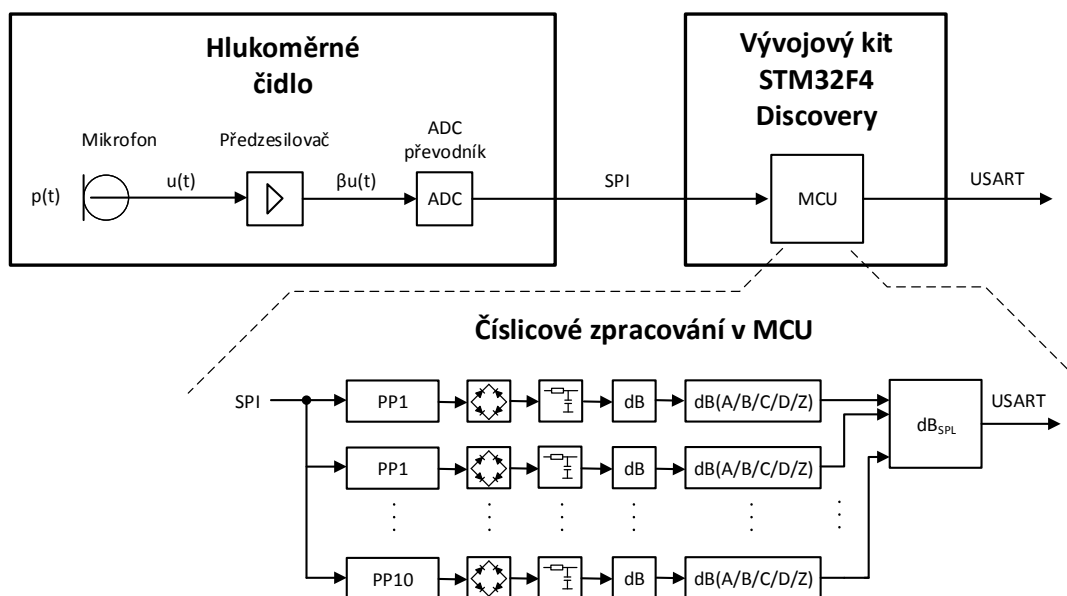
Parametry mikrokontroléru	
Frekvence	168 MHz
Flash paměť	1 MB
RAM	192 KB
Seriová komunikace	SPI, I2C, USART, UART, Ethernet, CAN, SDIO
ADC převodník	3 x 12-bit, 24 kanálů, 2.4 MSPS
DMA	16 streamů
časovače	12 x 16-bit, 2 x 32-bit
Debugger	ST-LINK/V2, SWD, JTAG
Oscilátor	Interní 16 MHz RC

Parametry vývojového kitu	
Oscilátor	Externí 8 MHz krystalový
Periferie	MEMS akcelerometr
	MEMS audio senzor
	Všesměrový digitální mikrofón
	audio DAC
LED	4 x user LED, 2 x USB OTG LED
Tlačítka	user a reset

2 ŘEŠENÍ STUDENTSKÉ PRÁCE

V rámci diplomové práce byl navržen a realizován prototyp hlukoměrného čidla, který byl poté připojen k vývojovému kitu STM32F4 Discovery a byly vytvořeny ovladače ke komunikaci s hlukoměrným čidlem. Na obr.2.1 je zobrazeno blokové schéma celé sestavy. Střídatá složka akustického tlaku je snímána mikrofonom. Získaný analogový signál je zesílen mikrofoniím předzesilovačem, a tento zesílený signál je přiveden do ADC převodníku, kde je vzorkován s frekvencí 48 KHz. Jednotlivé vzorky jsou odesílány do MCU pomocí sériové komunikace SPI.

Dále v MCU probíhá zpracování vzorků. Nejdříve je diskretní signál filtrován oktávovou bankou filtrů. Následně jsou signály v jednotlivých pásmech usměrněny. Poté dochází k časovému váhování a následně k výpočtu hladiny hluku pro jednotlivá pásma. Hodnoty pro tyto pásma jsou korigovány podle typu váhového filtru. Nakonec je z těchto hodnot spočtena výsledná hladina hluku, která je dále odeslána pomocí sériové komunikace USART. Typ časového váhování, váhování podle vnímání hlasitosti a počet vzorků které mají být posílány, lze nastavit pomocí specifických příkazů po USART.



Obr. 2.1: Blokové schéma vytvořené sestavy hlukoměru

2.1 Hlukoměrné čidlo

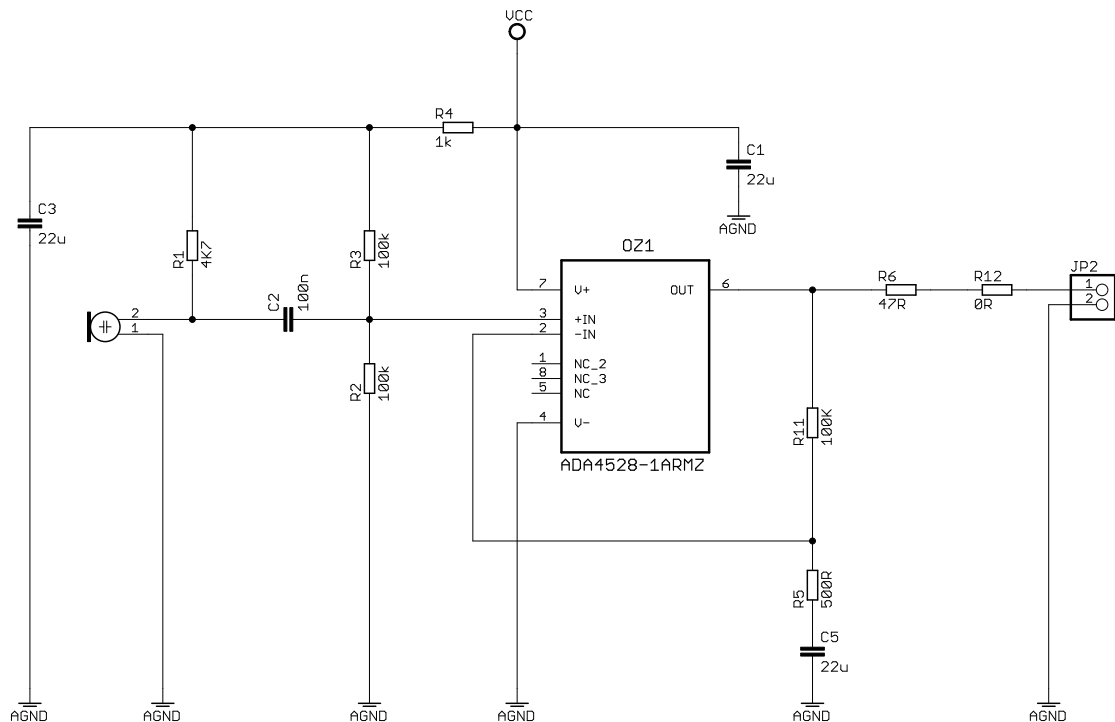
V této kapitole je popsán návrh prototypu hlukoměrného čidla. DPS čidla je rozdělena do tří základních částí:

- Mikrofon a mikrofonní předzesilovač
- ADC převodník a jeho pomocné obvody
- Digitální senzory teploty, tlaku a vlhkosti

Tvorba schémat elektrických obvodů a návrh DPS byly provedeny v návrhovém programu Cadsoft Eagle v6. Bylo využito jeho standardních knihoven a knihoven poskytnutých od výrobců zvolených součástek. Tyto externí knihovny jsou umístěny v Eagle projektu v příloze a na DVD.

2.1.1 Mikrofon a mikrofonní předzesilovač

Jako mikrofon byl zvolen elektretový mikrofon MCE-2500. Mikrofonní předzesilovač je postaven na neinvertujícím zapojení operačního zesilovače viz. obr.2.2



Obr. 2.2: Schéma mikrofonního předzesilovače

Mikrofon je napájen přes rezistor R1. Napájecí napětí je 1 V a má proudový odběr 0,4 mA. Pomocí kondenzátoru C2 je poté tato stejnosměrná složka oddělena. Poté je signál offsetován na 1,25 V z důvodu absence symetrického napájení a použití single-ended ADC převodníku s rozsahem vstupního napětí 0 – 2,5 V. Tento offset je realizován pomocí rozdělení napájecího napětí rezistory R2 a R3. Díky výrobním tolerancím rezistorů, je reálný offset 1,29 V.

Tento offsetovaný signál je dále přiveden na neinvertující vstup operačního zesilovače. Na invertující vstup je zapojena zpětná vazba sestávající se z rezistorů R5, R11 a kondenzátoru C5. Zesílení zesilovače je dáno vztahem:

$$u_{vyst} = u_{vst} \cdot \left(1 + \frac{R11}{R5}\right) \quad [V]. \quad (2.1)$$

Nulový rezistor R12 byl do obvodu přidán kvůli možností odpojení předzesilovače, z důvodů jeho měření.

Jako operační zesilovač (dále jen OZ) byl zvolen ADA4528 od výrobce Analog Devices, Inc. [7] v pouzdru MSOP-8. OZ je nízko šumový a Rail-to-Rail (umí pracovat i s napětími, blížícími se napájecímu napětí). Spotřeba OZ je výrobcem definována na 1,4 mA.

Jako pasivní součástky (rezistory a kondenzátory) byly zvoleny keramické, v pouzdrech 0603 a 0805.

Měření mikrofonního předzesilovače

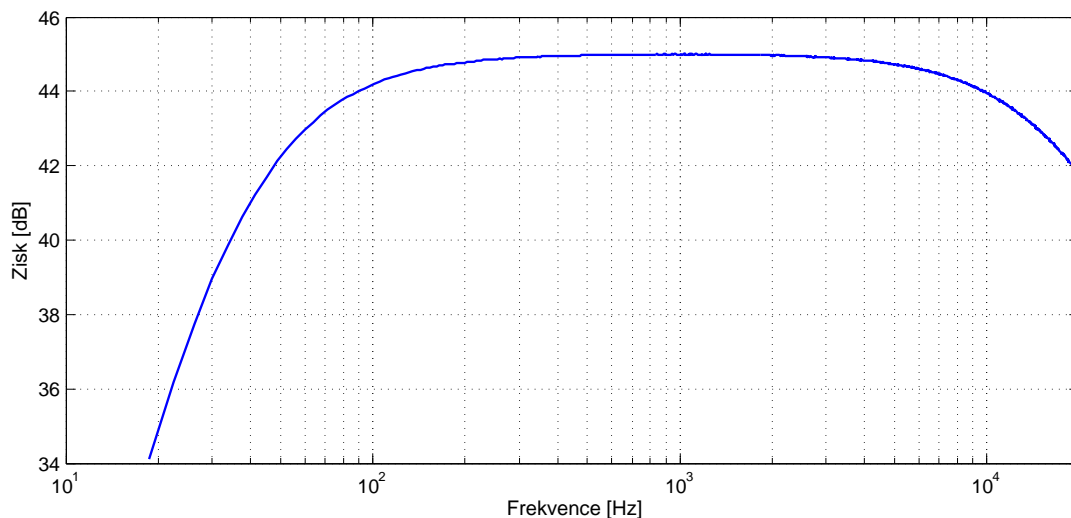
Vytvořený předzesilovač byl proměřen v laboratořích předmětu Studiová a hudební elektronika. K měření byl využit specializovaný software Audio Precision APx500. Při vstupním napětí 10 mV a frekvenčním rozsahu 20 – 20000 Hz byly změřeny tyto charakteristiky :

- Závislost zesílení na kmitočtu (Gain) 2.3
- Relativní zesílení vztahené k referenční hodnotě 1kHz (Relative Level (1kHz)) A.1
- Celkové Harmonické zkreslení (Total harmonic distortion – THD) A.2
- Celkové harmonické zkreslení zahrnující šum (Total harmonic distortion plus noise – THD+N) A.3
- Harmonické zkreslení druhou až desátou harmonickou složkou (Distortion Product Ratio) A.4

Závislost zesílení na kmitočtu je zobrazena na obr.2.3. Zesílení lze považovat za stabilní v oblasti 100 – 10000 Hz. V oblastech pod 100 Hz a nad 10000 Hz dochází

k poklesu zesílení. Tento pokles zesílení je kompenzován programově při výpočtech hladiny hluku v MCU.

Zbývající charakteristiky předzesilovače jsou umístěny v příloze A.1.



Obr. 2.3: Závislost zesílení na kmitočtu (Gain)

Celkové hodnoty, pro celé měřené pásmo, jsou uvedeny v tabulce 2.1. Oproti předzesilovači navrženém v rámci semestrálního projektu, byly vlastnosti předzesilovače vylepšeny. Především co se týká šumu. Odstup signálu od šumu je větší o 10 dB.

Tab. 2.1: Celkové změřené hodnoty předzesilovače

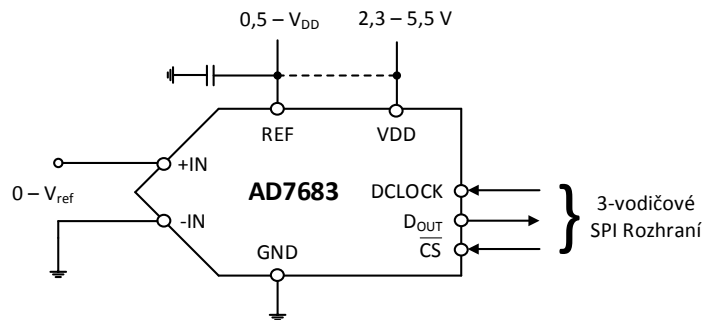
Celkové Harmonické zkreslení - THD [%]	0,0209
Celkové Harmonické zkreslení - THD [V]	0,000131
Celkové Harmonické zkreslení + šum - THD+N [%]	0,16
Celkové Harmonické zkreslení + šum - THD+N [V]	0,00101
Úroveň šumu [%]	0,159
Úroveň šumu [V]	0,001
Rozkmit zesílení [dB]	5,08
Odstup signálu od šumu [dB]	64

2.1.2 ADC převodník a jeho pomocné obvody

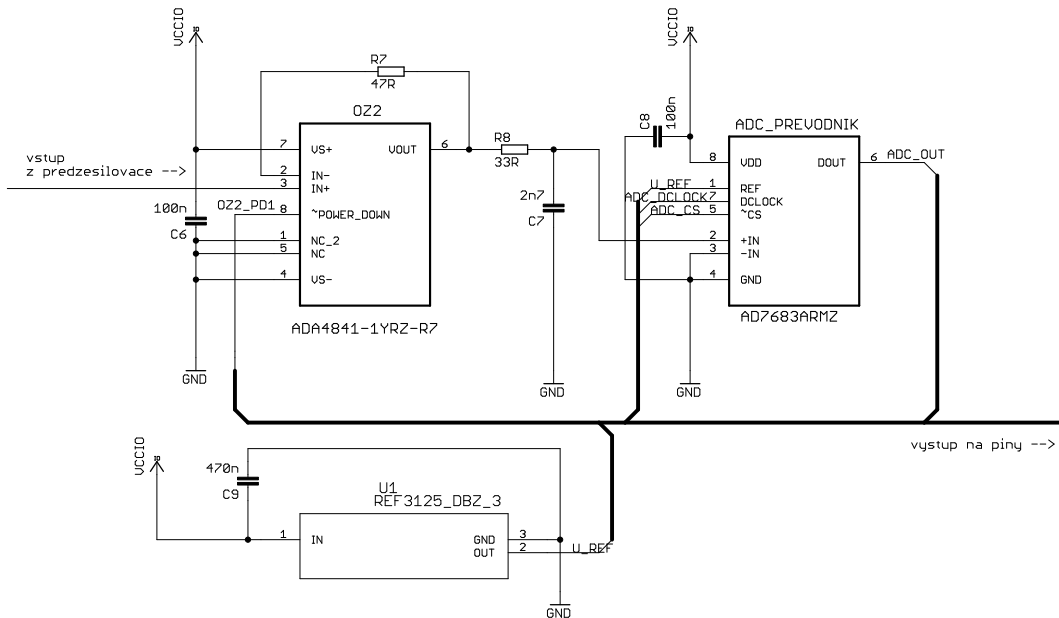
V této kapitole je popsán výběr ADC převodníku, jeho výstupní digitální komunikace a jeho začlenění do obvodu hlukoměrného čidla.

Návrh obvodu ADC převodníku

Pro digitalizaci signálu byl použit ADC převodník AD7683 od výrobce Analog Devices, Inc. [8] v pouzdru MSOP-8. Jde o 16-bitový pseudo diferenční převodník s postupnou aproximací. Maximální vzorkovací frekvence je 100 kSPS. Umožňuje nesymetrické napájení (single-supply) 2,7 – 5,5 V. Převodník má vstup pro napětovou referenci se vstupním rozsahem 0,5 V – napájecí napětí. Výstupní digitální rozhraní je tří-vodičové podporující SPI/QSPI/MICROWIRE/DSP rozhraní. Aplikační diagram převodníku je na obr.2.4. Reálné schéma zapojení této části je na obr.2.5.



Obr. 2.4: Aplikační diagram AD7683 [8]



Obr. 2.5: Schéma zapojení ADC převodníku

Jako pomocný obvod je k převodníku připojena napěťová reference REF3125 od výrobce Texas Instruments, Inc. [9], která poskytuje stabilní referenční napětí pro ADC převodník. Jako „buffer“ pro ADC převodník slouží operační zesilovač ADA4841 opět od Analog Devices, Inc. [10]. Tyto součástky byly připojeny k převodníku podle doporučeného schématu od výrobce.

Digitální komunikace ADC převodníku

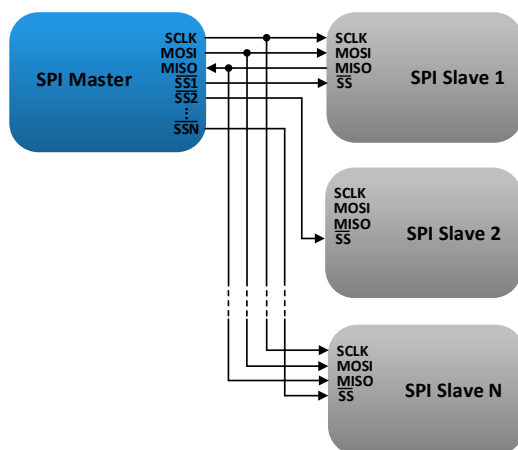
V této práci probíhá komunikace a ADC převodníkem pomocí SPI komunikace. SPI je standard pro synchronní sériovou komunikaci. Jde o jednoduchou komunikaci, která vznikla v roce 1979 u mikrokontrolérů Motorola. Existují dva typy zařízení: Master a Slave. Pracuje ve full-duplex módu pomocí čtyř signálových vodičů:

- SCLK – hodinový signál (clock) generovaný masterem, ostatní SPI signály jsou asynchronní k tomuto
- \overline{SS} – (Slave Select) výběr slave zařízení, se kterým chce master komunikovat
- MOSI – (Master OUT – Slave In) datový vodič pro komunikaci od master k slave
- MISO – (Master In – Slave Out) datový vodič pro komunikaci od slave k master

Existují dvě topologie:

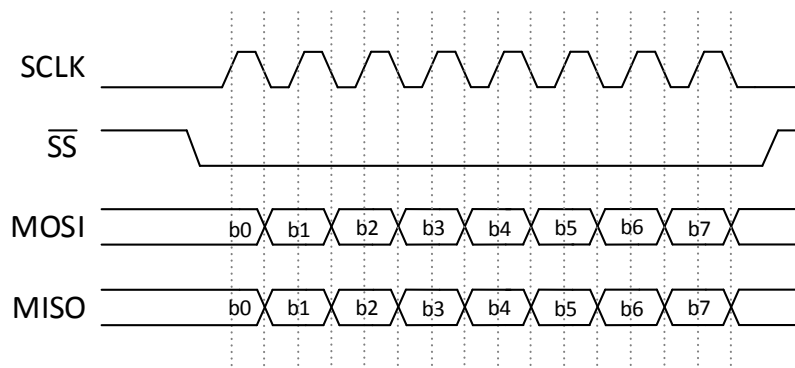
- Master připojený k jednomu slave (point-to-point topology)
- Master připojený k více slave

Sběrnice SPI je na obr.2.6.



Obr. 2.6: SPI sběrnice [11]

Komunikaci zahajuje master, tím že nastaví \overline{SS} pin zařízení (slave) se kterým chce komunikovat na log. 0 a začne gerovat hodinový signál (clock) na dohodnuté frekvenci. Master odesílá data na MOSI a zároveň přijímá data na MISO. Princip této komunikace je na obr.2.7.



Obr. 2.7: Obecný princip SPI komunikace [11]

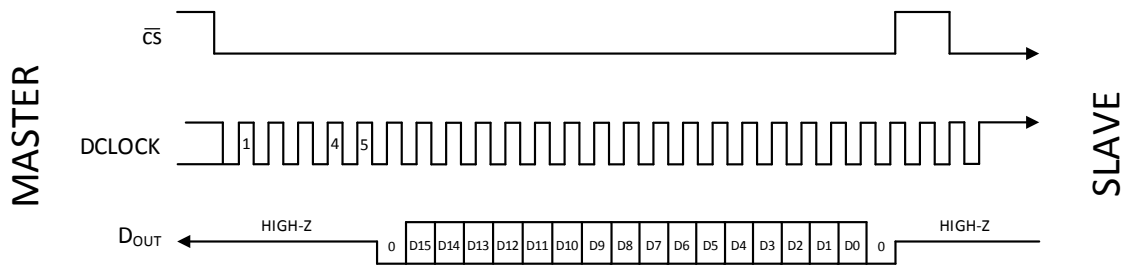
Existují čtyři módy komunikace, mód 0, 1, 2, 3. Tyto módy definují na jaké hraně hodinového signálu budou vzorkována data (CPHA) a v jaké logické úrovni bude hodinový signál, když nebude nic přenášeno (CPOL):

- **mód 0:** klidová polarita SCLK: CPOL=0, vzorkováno na vzestupné hraně SCLK CPHA=0
- **mód 1:** klidová polarita SCLK: CPOL=0, vzorkováno na sestupné hraně SCLK CPHA=1
- **mód 2:** klidová polarita SCLK: CPOL=1, vzorkováno na vzestupné hraně SCLK CPHA=0
- **mód 3:** klidová polarita SCLK: CPOL=1, vzorkováno na sestupné hraně SCLK CPHA=1

Komunikace s ADC převodníkem použitým na hlukoměrném čidle se od obecného modelu liší. Je využíváno tří-vodičového zapojení. ADC převodník reprezentuje v této komunikaci zařízení typu slave, MCU je master. Způsob komunikace je popsán v časovacím diagramu na obr.2.8. Vývody u ADC převodníku odpovídají těm pro SPI komunikaci takto:

- DCLOCK : SCLK
- \overline{CS} : \overline{SS}

- D_{OUT} : MISO



Obr. 2.8: Časovací diagram ADC převodníku [8]

Sestupná hrana na pinu \overline{CS} spustí konverzi převodníku. Po páté sestupné hraně DCLOCK (SPI clock) dojde k povolení D_{OUT} a je nastaven na log. 0. Po další periodě SPI hodinového signálu (SPI clock) začne vysílání 16-bitového výsledku konverze. První je vyslán nejvíce významný bit (MSB). K přenosu celé hodnoty je potřeba minimálně 22 period hodinového signálu (Na obr.2.8 je zobrazeno 24), ovšem je lepší použít o něco více.

ADC převodník nedokáže žádná data přijmat (absence vstupu pro data), tzn. že nelze nastavit, v jakém formátu nám převodník bude hodnoty vracet, nebo s jakou frekvencí. Z toho vyplývá, že nastavení vzorkovací frekvence je tedy plně v režii Masteru (v tomto případě MCU) a je závislé na tom, jak často jsou na pinu \overline{CS} generovány sestupné hrany. Maximální vzorkovací frekvence je 100 kSPS. Detailní popis SPI komunikace je v datasheetu ADC převodníku [8].

2.1.3 Digitální senzory teploty, tlaku a vlhkosti

Na DPS hlukoměrného čidla byly umístěny i digitální senzory teploty, tlaku a vlhkosti pro možné budoucí využití při korekci hodnot hladiny hluku. Využití je bude možné při pokračování, postaveném na této práci v rámci další závěrečné práce.

Jako senzor vlhkosti bylo zvolen senzor SENSIRION SHT21 [12] v pouzdru DFN. Senzor měří teplotu a vlhkost. Měření vlhkosti probíhá s přesností $\pm 2\%$ RH s rozsahem 0 – 100 % RH. Výstupní digitální rozhraní podporuje standard I²C. Další parametry jsou uvedeny v tab.2.2.

Na stejnou sběrnici byl také umístěn senzor tlaku MS5637-02BA03 od výrobce Measurement Specialties™ [13] v pouzdru QFN. Senzor měří tlak s přesností ± 2 mbar s rozsahem 10 – 2000 mbar. Senzor také měří teplotu s přesností ± 1 °C. Výstupní digitální rozhraní podporuje standard I²C. Další parametry jsou uvedeny v tab.2.3.

Tab. 2.2: Parametry senzoru SHT21 [12]

Parametry senzoru					
Vlhkost	Stav	Min.	Typ.	Max.	Jednotka
Rozlišení	12 bit		0,04		%RH
	8 bit		0,7		%RH
Přesnost	Typ.		±2,0		%RH
Hystereze			±1		%RH
Čas odezvy	$\tau_{63\%}$		8		s
Operační rozsah		0		100	%RH

Tab. 2.3: Parametry senzoru MS5637-02BA03 [13]

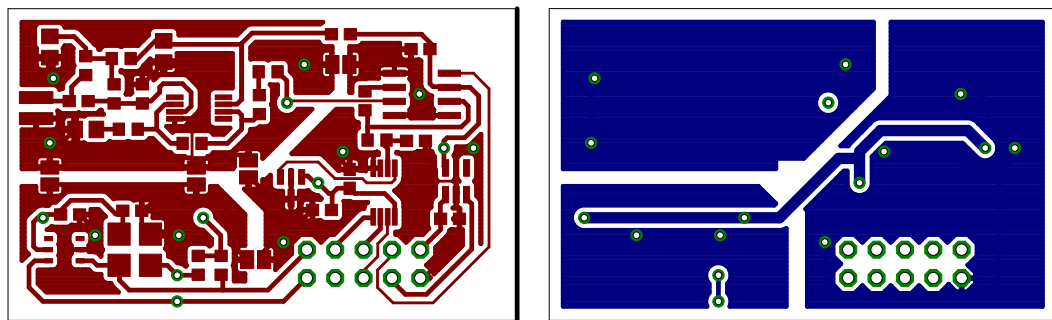
Parametry senzoru (při napájení 3 V)				
Tlak	Min.	Typ.	Max	Jednotka
Maximální rozsah	10		2000	mbar
Rozlišení (8192 - OS) (1)	0,11/0,062/0,039 /0,028/0,021/0,016			mbar
Chybovost při 25 °C	-2		+2	mbar
Čas odezvy (1)	0,5/1,1/2,1/4,1 /8,22/16,4			mbar
Dlouhodobá stabilita		±1		mbar/rok
Teplota	Min.	Typ.	Max.	Jednotka
Rozsah	-40		85	°C
Přesnost při 25 °C	-1		+1	°C
Poznámka: (1) Hodnota převzorkování: 256/512/1024/2048/4096/8192				

2.1.4 DPS hlukoměrného čidla

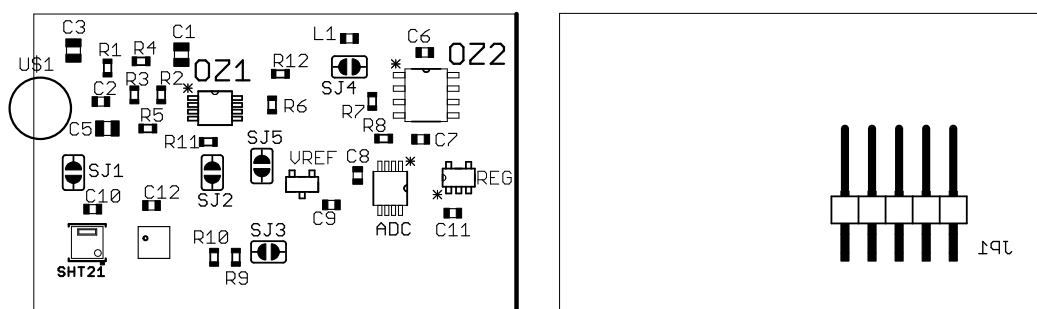
Všechny tři schémata zapojení byla realizována na jedné desce plošných spojů. Schéma celého systému je v příloze na obr.A.5. Všem třem částem byly odděleny země. Tím se zamezilo vzniku rušivých zemních smyček, jimiž by docházelo k rušení analogových obvodů těmi digitálními. Tyto tři oddělené země byly spojeny ve dvou místech, ale propojných míst bylo připraveno více. Kdyby vlastnosti obvodu byly špatné, je možnost propojení zemí realizovat na jiném místě. Tyto propoje jsou realizovány specifickými jumpery (tak jsou pojmenovány v knihovně programu Eagle), což jsou vlastně dvě měděné plošky blízko sebe, které se při pájení „přelíjí“ pájkou (cínem). Navržená deska je na obr.2.9. Vlevo je zobrazena horní vrstva (Top). Na této straně jsou umístěny všechny součástky kromě jedné. Vpravo je spodní vrstva

(Bottom), na které žádné součástky nejsou, pouze propoje, výstupní piny a „rozlitá“ zem, fungující jako stínění obvodu [14]. Výstup z hlukoměrného čidla je realizován oboustrannými kolíky S2G10 zahnutými o 90°

Na obr.2.10 je zobrazen osazovací plán DPS, který ukazuje rozmístění součástek na desce.



Obr. 2.9: DPS - obrazec spojů horní vrstvy (vlevo) a spodní vrstvy (vpravo)



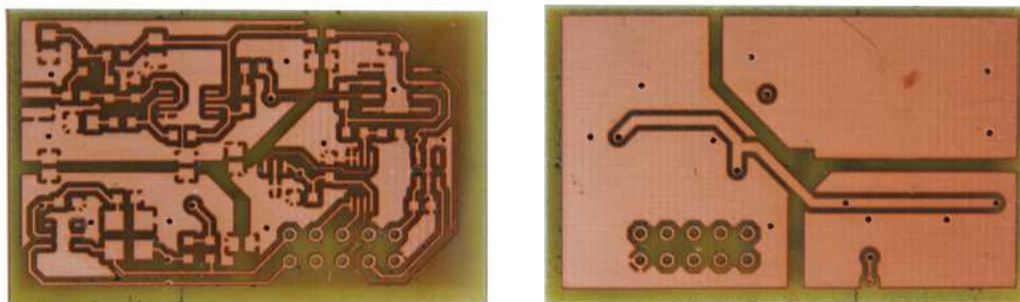
Obr. 2.10: DPS - Osazovací plán horní vrstvy (vlevo) a spodní vrstvy (vpravo)

Na výstupu z předzesilovače byl umístěn nevyužitý prokov, který slouží jako měřící kontakt.

Celý obvod je napájen přes lineární regulátor s výstupním napětím 3 V. Tento regulátor lze pomocí pinu SHUTDOWN vypnout. To lze provést přivedením logické 0 na tento pin. Po vypnutí regulátoru má celé hlukoměrné čidlo téměř nulový proudový odběr a tím lze dosáhnout velké úspory v případě, že není čidlo využíváno. Zdroj napájení hlukoměrnému čidlu poskytuje vývojový kit STM32F4 Discovery, a velikost napájecí napětí je 5 V.

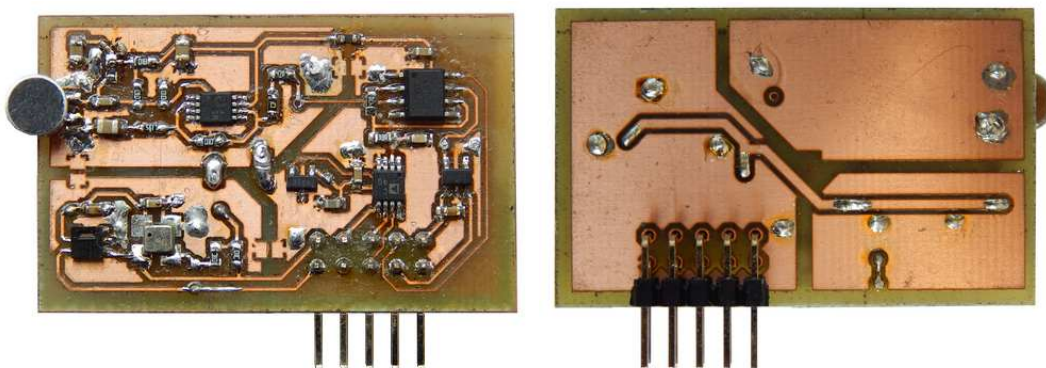
Pin SHUTDOWN obsahuje i operační zesilovač OZ2, který tvoří „buffer“ ADC převodníku. Pokud vypneme regulátor, a tím odpojíme od napájení celou desku, není nutné tento OZ vypínat. Jeho vypnutí se může využít v případě měření předzesilovače, jehož výstupní signál se tak nedostane k ADC převodníku.

DPS byla vyrobena v laboratořích ústavu UTKO a její vzhled je zobrazen na obr.2.11



Obr. 2.11: Vyrobená neosazená DPS hlukoměrného čidla, vrstva Top (vlevo) a (Bottom) (vpravo)

Osazování desky a prokovy na desce byly udělány ručně. Výsledný vzhled osazené DPS hlukoměrného čidla je na obr.2.12. Lze vidět zřetelné rozdělení zemí jednotlivých částí obvodu.



Obr. 2.12: Osazená DPS hlukoměrného čidla, vrstva Top (vlevo) a (Bottom) (vpravo)

Mikrofon byl umístěn na okraj desky na univerzálně navržené plošky, aby bylo možné mikrofon vyměnit za jiný typ obdobné velikosti.

2.2 Návrh IIR filtrů

Pro filtraci signálu pomocí oktávkové banky filtrů, bylo navrženo deset IIR filtrů typu pásmová propust. Tyto filtry byly navrženy v nástroji FDATool programu Matlab®. Pro dosažení požadovaných vlastností, tj. aby se propustná pásma jednotlivých filtrů protínaly v -3 dB, byly filtry navrženy pomocí butterworthovy aproximace. Filtry

byly navrženy pro práci se vzorkovací frekvencí 48 KHz. Příklad návrhu filtru v programu FDATool je v příloze na obr.A.7.

Navržené filtry byly vytvořeny jako přímá forma II, jako tři sekce druhého řádu. Protože je potřeba pro využití DSP instrukcí, aby navržený filtr byl ve formě Lattice, je třeba jej do této formy konvertovat. To lze provést pomocí: Edit -> Convert Structure ... -> Lattice Autoregressive Moving-Average (ARMA).

Pro získání přímých koeficientů šestého řádu, byly navržené filtry převedeny pomocí: Edit -> Convert to Single Section.

Tyto koeficienty lze exportovat do workspace programu Matlab® (File -> Export), nebo si lze nechat vygenerovat přímo hlavičkový soubor v jazyce C (Targets -> Generate C header).

Pro filtry v nízkých frekvencích by bylo zbytečné používat plnou vzorkovací frekvenci 48 KHz, proto první čtyři filtry z tab.1.1 byly navrženy pro vzorkovací frekvenci 1920 Hz (Tato frekvence byla zvolena tak, aby byla dobře soudělitelná s frekvencí 48 KHz). To znamená, že z načtených vzorků je příslušný počet vzorků vypuštěn a tím je signál podvzorkován. Avšak aby se předešlo aliasingu, je nejdříve nutné tyto vzorky vyfiltrout anti-aliasingovým filtrem. Ten je vytvořen jako dolní propust s mezní frekvencí 400 Hz. Toto nám zajistí úsporu značného výpočetního času MCU.

Obdobně byly podvzorkovány i další tři filtry, ovšem s vyšší vzorkovací frekvencí 12 KHz. Opět byl použit anti-aliasingový filtr, tentokrát s mezní frekvencí na 3000 Hz.

Poslední tři filtry už podvzorkovány nebyly.

2.2.1 IIR filtry s využitím DSP instrukcí

Jak bylo zmíněno, je vhodné využít k realizaci filtrů DSP instrukcí, které zvolený MCU podporuje. To umožní rychlejší výpočty. K realizaci těchto filtrů bylo využito knihovny CMSIS-DSP Library [5], která byla naimportována do projektu. V této knihovně je před-připraveno mnoho matematických funkcí a funkcí pro DSP. Co se týče filtrů, je připraveno více typů filtrů FIR, avšak pouze jeden typ IIR filtru, a to právě výše zmíněný lattice filtr.

Pro použití tohoto filtru je nutné jej nejdříve příslušně inicializovat. Jak to udělat je popsáno v .c souboru filtru. Je třeba definovat koeficienty filtru, které musí být v opačném pořadí, než jak je vygeneruje FDATool.

```
#define NL8 7
const float32_t NUM8[NL8] = {
    -0.0041904888635335893, -0.019206582213537587,
    -0.029004956191588598, -0.00041696147501512487,
    0.021144393774513465, 0.00077223500644650611,
```

```
-0.0022627156471827451};
```

```
#define DL8 6  
const float32_t DEN8[DL8] = {  
0.48495452926707433, -0.78799159103892447,  
0.92836075722116085, -0.87520299513719368,  
0.97511217625625679, -0.91324904172083543};
```

```
#define NUMBER_OF_TAPS_8 6
```

Dále je nutné připravit tzv. „state buffer“, nutný pro výpočty daného filtru, který má velikost zpracovávaného bloku dat + řád filtru:

```
float32_t StateBuffer8 [BLOCK_SIZE + NUMBER_OF_TAPS_8];
```

Poté je třeba vytvořit samotnou instanci filtru:

```
arm_iir_lattice_instance_f32 PP8_3150_5000Hz;
```

Následuje inicializace, kde jsou jako vstupní parametrem instance filtru, řád filtru, koeficienty čitatele, koeficienty jmenovatele, state buffer a velikost bloku dat:

```
arm_iir_lattice_init_f32( &PP8_3150_5000Hz, NUMBER_OF_TAPS_8,  
&DEN8[0], &NUM8[0], &StateBuffer8 [0], BLOCK_SIZE);
```

Poté již lze spustit filtraci:

```
for(i=0; i < numberOfStages; i++)  
{  
arm_iir_lattice_f32( S, inputBuffer + (i * BLOCK_SIZE),  
outputBuffer + (i * BLOCK_SIZE), BLOCK_SIZE);  
}
```

2.3 Zpracování vstupních dat v MCU

Vytvořené hlukoměrné čidlo popsané v předchozí kapitole bylo připojeno k vývojové kitu STM32F4 Discovery, kde jsou přijímány vzorky signálu a jsou zpracovány. Vývojový diagram celého ovladače hlukoměru je v příloze na obr.A.6.

2.3.1 Vývojové prostředí MCU

K programování MCU byl využito vývojové prostředí CoCox CoIDE v1.7.8, které je poskytováno zdarma. Nejnovější verze tohoto programu je verze 2, která je však ve fázi beta testování. Proto byla k vývoji využita stabilní verze 1, která je i více uživatelsky přívětivá (osobní názor autora). CoIDE je postaven na bázi Eclipse. Dalším důvodem k využití tohoto IDE byla rozsáhlá komunita vývojářů. Přímo v programu

jsou integrovány standartní periferní knihovny (STM32F4 standard peripherals library) od výrobce ST pro práci s perifériemi, které byly využity při řešení této práce. Tyto knihovny poskytují abstrakci při práci s registry MCU, díky tomu je programování snažší a přehlednější.

Pro programování byl využit programovací jazyk C. Jako překladač byl využit GNU Tools ARM Embedded verze 4.9 2014q4, který je instalován externě, a v CoIDE je nutno k němu nastavit cestu. Popis instalace CoIDE, jeho základní nastavení a nastavení překladače je popsáno v příloze v kapitole A.7.

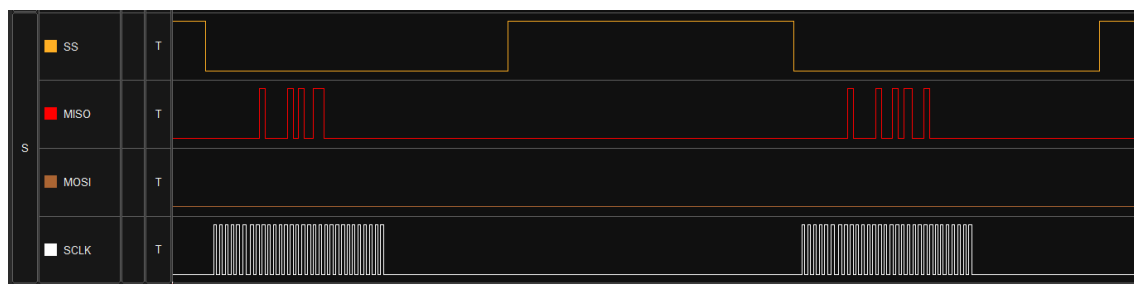
Jako další vývojová prostředí je možné využít např.: Keil μ Vision IDE, Atollic TrueSTUDIO[®], IAR-EWARM, ...

Při programování byla zapnuta optimalizace překladače -O1, která umožní změnu pořadí zpracovávaných operací tak, aby byly efektivněji prováděny. Zapnutí optimalizace přináší mnoho nevýhod (problému) při ladění programu, debugger se chová nelogicky a je třeba věnovat velkou pozornost při deklaraci proměnných a funkcí. Je třeba proměnné (a někdy i funkce) pracující s perifériemi deklarovat jako `volatile`, čímž zabráníme jejich optimalizaci.

Zapnutí optimalizace sice ztěžuje vývoj programu, ale zrychlení výpočtů je významné. Zapnutím došlo zhruba k čtyř-násobnému zrychlení. Byla testována i větší optimalizace -O2 a -O3, ale neposkytla nijak významně vyšší výkon a SPI komunikace s ADC převodníkem byla příliš nestabilní.

2.3.2 Komunikace s hlukoměrným čidlem

Komunikace s hlukoměrným čidlem probíhá pomocí standartu SPI. Komunikace probíhá přímo s ADC převodníkem. Pro tuto komunikaci jsou použity tři vodiče: \overline{CS} , DCLOCK a D_{OUT}. Teoretický princip komunikace byl popsán v kapitole „ADC převodník a jeho pomocné obvody“ 2.1.2. Komunikace s převodníkem byla zachytávána za pomoci logického analyzátoru BitScope BS05U. Zachycená komunikace je na obr.2.13



Obr. 2.13: Zachycená komunikace SPI

Komunikace SPI byla nastavena do módu 0. Tzn. že je hodinový signál v době nečinnosti na log. 0 a data vzorkována na vzestupné hraně. Byl využit kanál 3. Piny pro SPI3 byly u MCU namapovány na následující piny:

- SCLK (DCLOCK) : PB3
- \overline{SS} (\overline{CS}) : PD7
- MISO (SDATA) : PB4

Frekvence SPI hodinového signálu (SCLK) je nastavena na 2,25 MHz a je spočítána dle následujícího vzorce:

$$f_{CLK} = \frac{\text{frekvence sběrnice (APB1 bus)}}{\text{SPI dělička(SPI prescaler)}} \quad (2.2)$$

$$= \frac{42000000}{8} = 5250000 \text{ Hz} = 5,25 \text{ MHz.}$$

Dále je nastaveno, že první přijatý bit bude nejvýznamnější (MSB).

Časovač

K spuštění konverze ADC převodníku dochází sestupnou hranou na pin \overline{CS} . Z pohledu MCU je to pin \overline{SS} . Generování tohoto signálu probíhá za pomoci časovače MCU. Je využit časovač 2 (TIM2). Po přetečení registru časovače dochází k přerušení, ve kterém je negován pin SS, proto je časovač nastaven na dvojnásobnou frekvenci než je vzorkovací. Hodnota registru (perioda) pro časovač TIM2 je nastavena dle vzorce:

$$\text{Perioda (period)} = \frac{\text{frekvence sběrnice (APB1 bus)}}{2 \cdot \text{vzorkovací frekvence (sample frequency)}} \quad (2.3)$$

$$= \frac{42000000}{2 \cdot 48000} = 437,5 [-].$$

Výsledná hodnota byla testována pomocí osciloskopu a korigována na 436.

DMA

MCU STM32F4 disponuje DMA kontrolérem, který slouží k přímému přístupu k paměti. Dokáže kopírovat data z a do periférií. Tyto operace probíhají nezávisle na času procesoru (na pozadí) a tím umožňují značnou úsporu výpočetního výkonu. Pro práci s SPI je vyhrazen DMA kanál 2. Každý kanál má 7 streamů, pro odesílání je

definován stream 5, pro příjem pak stream 0. Jak bylo dříve zmíněno, odesílání a příjem u SPI probíhají zároveň. I když ADC převodník nedokáže žádná data přijímat, přesto musíme nějaká data odesílat, abychom mohli přijímat data od převodníku. Proto dochází k odesílání tzv. dummy-bytes (samé nuly).

Při generování sestupné hrany časovačem, je spuštěn transfer dat pomocí DMA. Jsou odeslány čtyři byty dat a čtyři byty jsou přijaty. při dalším přerušení od časovače, jsou přijatá data načtena z DMA bufferu. V přijatých čtyřech bytech je obsažena námi požadovaná 16-bitová hodnota výsledku AD konverze. Tato hodnota je poté bitovými rotacemi z oněch čtyř bytů extrahována. I když výrobce udává, že po páté periodě hodinového signálu je vrácena hodnota, počet period se liší podle frekvence hodinového signálu (SCLK). Proto je třeba přijatá data analyzovat (debugovat) a nastavit správně bitové posuvy. Každá hodnota AD převodu je hned při načtení zbavena offsetu a převedena na desetinné číslo, vyjadřující velikost napětí. Tato hodnota je uložena do 32-bitového datového typu float32_t.

2.3.3 Zpracování dat

V této kapitole je popsáno zpracování dat od vstupu do MCU, přes jejich filtraci a váhování až do výpočtu finální hodnoty hladiny hluku.

Bloky vstupních dat

Výsledky AD konverzí (vzorky signálu) jsou zpracovávány po blocích. Velikost těchto bloků je definována na 1000 hodnot. Během načítání oněch 1000 hodnot je zpracováváno 1000 předchozích hodnot. To znamená, že nedochází k žádné ztrátě hodnot měřené hladiny hluku. Toto je zajištěno příjmem dat do dvou nezávislých bufferů, mezi kterými je přepínáno. Naplnění obou bufferů značí signalizační proměnná. Pokud je nastavena, zapne se zpracování načtených dat.

Filtrace vstupních dat

První částí zpracování dat je jejich filtrace. Všech 1000 hodnot je filtrováno oktávovou bankou filtrů. Tzn. že data jsou 10-krát filtrována. Tím oddělíme jednotlivé harmonické složky signálu do oktáv. Jak bylo zmíněno v kapitole „Návrh IIR filtrů“ 2.2, byly použity IIR filtry typu lattice s využitím DSP instrukcí. Po filtraci je z takto získaných průběhů signálu potřeba získat efektivní hodnotu.

Časové váhování vstupních dat

Po každé filtraci je z filtrovaných dat vypočtena efektivní hodnota signálu pro dané pásmo. Efektivní hodnota je vypočtena pomocí časového váhování. Časové váhování

je vlastně filtr typu dolní propust (integrační článek) s dlouhou časovou konstatou. Jsou definovány 3 typy váhování:

- Fast – časová konstanta $\tau F = 125$ ms
- Slow – časová konstanta $\tau S = 1$ s
- Impulse – časová konstanta $\tau I = 35$ ms (pokud hodnota stoupá) a 1,5 s (pokud hodnota klesá)

K výpočtu je využito vzorce:

$$y[n] = (1 - \alpha) \cdot x[n] + \alpha \cdot y[n - 1], \quad (2.4)$$

kde:

- $y[n]$ aktuální hodnota časového váhování
- $x[n]$ aktuální vzorek signálu
- $y[n-1]$ předchozí hodnota časového váhování
- α koeficient časového váhování

Koeficient α je závislý na vzorkovací frekvenci a je vypočten dle vzorce:

$$\alpha = e^{\frac{-1}{f_{vz} \cdot \tau F}} \quad (2.5)$$

Před časovým váhováním je nutné vstupní vzorky dvoucestně usměrnit, tzn. provést ze všech vzorků výpočet absolutní hodnoty:

$$x[n] = |x[n]|. \quad (2.6)$$

Výpočet hladiny hluku

Nyní, když je známa efektivní hodnota signálu ve všech pásmech, je spočítána Hladina hluku dle vzorce:

$$L_{pb} = 20 \log \left(\frac{u_{ef}}{p_0 \cdot G} \right) - \eta_{dB} \quad [\text{dB}_{\text{SPL}}], \quad (2.7)$$

kde:

- L_{pb} ... Hladina hluku v daném pásmu
- u_{ef} ... Efektivní hodnota v daném pásmu
- p_0 ... Vztažná hodnota akustického tlaku = $2 \cdot 10^{-5}$ Pa
- G ... Zisk předzesilovače
- η_{dB} ... Citlivost mikrofону v dB

Takto jsou spočteny hodnoty hladin hluku pro všechny pásma. Než je spočtena výsledná hodnota, je provedeno váhování podle citlivosti lidského ucha na jednotlivé frekvence. Lze provést váhování podle pěti různých váhování: A, B, C, D a Z. Váhování je provedeno přičtením korekčních koeficientů k hladinám hluku v jednotlivých pásmech. Po váhování lze již spočítat výslednou hladinu hluku dle vzorce:

$$L = 20 \log \left(\frac{10^{\frac{L_{pb1}}{20}} + 10^{\frac{L_{pb2}}{20}} + \dots + 10^{\frac{L_{pbN}}{20}}}{N} \right) \quad [\text{dB}_{\text{SPL}}]. \quad (2.8)$$

2.4 Možnosti snížení energetické náročnosti

MCU lze nastavit frekvenci procesoru na které bude pracovat. Tato frekvence je přímo úměrná energetické náročnosti. Proto byl MCU podtaktován z maximální frekvence 168 MHz na 84 MHz. Tím došlo k snížení proudového odběru zhruba na polovinu. Na nižší frekvence již MCU podtaktovat nešlo, protože poté už procesor nebyl schopen stihnout zpracovat oněch 1000 vzorků, při načítání dalších.

Pokud není hlukoměru poslán příkaz na odesílání vzorků, je hlukoměrné čidlo vypnuto a MCU nastaven do úsporného módu. MCU STM32F4 má celkem 3 úsporné módy. Liší se tím, co vše bude vypnuto [15]:

1 SLEEP mód:

- Procesor Cortex-M4F je zastaven
- Periferie zůstávají zapnuté
- Jak povolit tento mód:
 - zavoláním funkce `__WFI()` nebo `__WFE()`
- Jak opustit tento mód:
 - jakékoli přerušení od periférií
 - systémovým časovačem
- Po probuzení:
 - MCU pokračuje tam kde skončil

2 STOP mód:

- všechny signály clock v 1,2V doméně jsou zastaveny
- PLL, HSI a HSE RC oscilátory jsou vypnuté
- obsah interní SRAM paměti a registrů je zachován
- napěťový regulátor je v nízko odběrovém módu
- Jak povolit tento mód:
 - zavoláním funkce `PWR_EnterSTOPMode(PWR_Regulator_LowPower, PWR_STOPEntry_WFI)`

- Jak opustit tento mód:
 - jakékoli externí přerušení
- Po probuzení:
 - MCU pokračuje tam kde skončil a použije HSI jako zdroj hodinového signálu
 - je doporučeno znovu zavolat funkci `SystemInit()` pro opětovné nastavení oscilátorů

3 STANDBY mód:

- Procesor Cortex-M4 je v hlubokém spánku
- Napěťový regulátor je vypnutý
- 1,2 V doména je vypnuta
- PLL, HSI a HSE RC oscilátory jsou vypnuté
- SRAM a obsah registrů je ztracen kromě RTC registrů, RTC backup registrů, backup SRAM a Standby obvodů
- Jak zapnout tento mód:
 - zavoláním funkce `PWR_EnterSTANDBYMode()`
 - nejdříve je vhodné vynulovat příznakové bity `PWR_ClearFlag(PWR_FLAG_SB | PWR_FLAG_WU)`
- Jak vypnout tento mód:
 - vzestupná hrana na Wakeup pinu (PA0)
 - RTC alarm
 - Tamper a time-stamp události
 - externí reset na NRST pinu
 - IWDG reset
- Po probuzení:
 - MCU je resetován

Výše zmíněné funkce na uspávání jsou součástí Standardních periferních knihoven od výrobce ST. Pro použití těchto funkcí je nutné naimportovat knihovnu `stm32f4xx_pwr.h`. Na vývojovém kitu je na pin PA0 připojeno uživatelské tlačítko (modré), tím je vlastně generována ona vzestupná hrana, která probudí MCU ze STANDBY módu (pro testovací účely). Pin PA0 je vyveden i samostatně. Toto tlačítko je nastaveno také jako zdroj externího přerušení, takže dokáže MCU probudit i ze STOP módu. Pro probuzení ze SLEEP módu, je využito přerušení od sériové komunikace USART.

3 VÝSLEDKY STUDENTSKÉ PRÁCE

V této kapitole je popsán výsledný prototyp hlukoměru, způsob jak s ním komunikovat a měření jeho výsledných parametrů. Nakonec je umístěno jeho závěrečné zhodnocení.

3.1 Výsledný hlukoměr



Obr. 3.1: Hlukoměrná sestava

Sestava na obr.3.1 je napájena z USB nebo z adaptéru s napětím 5 V. Pro připojení hlukoměrného čidla byla vytvořena redukce a kabel viz. obr.3.2. Kabel je vhodnější pro testovací účely, kdy je potřeba, aby mělo čidlo větší polohovací rozsah.



Obr. 3.2: Redukce (vlevo) a kabel (vpravo) pro připojení hlukoměrného čidla

3.1.1 Výstupní komunikace s hlukoměrným čidlem

Pro výstupní komunikaci bylo zvoleno rozhraní USART. Jde o tří-vodičovou komunikaci. Jeden vodič pro vysílání, jeden pro příjem a zem. Na vývojovém kitu STM32F4 Discovery byla tato komunikace definována na následujících pinech:

- PA2: vysílání (USART2 TX)
- PA3: příjem (USART RX)
- GND: zem

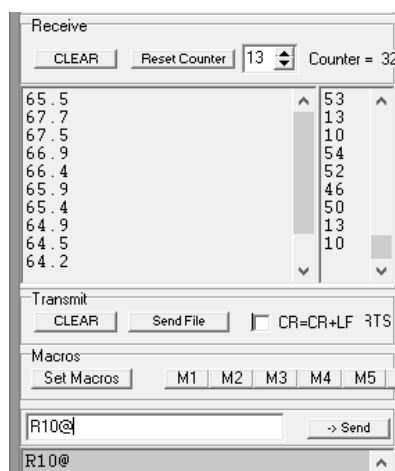
USART komunikaci byly nastaveny tyto parametry:

- Rychlost vysílání (Baud rate): 115200 Bd/s
- Počet datových bitů (Data bits): 8
- Parita (Parity): žádná (none)
- Počet Stop bitů (Stop bits): 1
- Inicializace (Handshaking): žádná (none)

Pro komunikaci s hlukoměrným čidlem byly vytvořeny příkazy pomocí znaků ASCII, které definují chování hlukoměru a odesílání hodnot:

- RXXX@
R ... čtení hodnot (Read)
XXX ... počet vzorků, které hlukoměr vrátí (1 – 999)
@ ... ukončovací znak

Příklad užití: R10@ – odeslání 10 hodnot hladiny hluku:



Obr. 3.3: Příklad užití příkazu R10@

- RCXX@
R ... čtení hodnot (Read)
C ... stálé odesílání hodnot (Continuous)
XX ... počet hodnot odeslaných za 1 s (1, 2, 3, 4, 6, 8, 12, 16, 24, 48)
@ ... ukončovací znak

Příklad užití: RC16@ – kontinuální odesílání hodnot, 16 hodnot za vteřinu

- S@
S ... zastavení odesílání hodnot (Stop)
@ ... ukončovací znak

Příklad užití: S@ – zastavení odesílání hodnot

- TX@
T ... nastavení časového váhování (Time weighting)
X ... typ časového váhování (F, S, I)
@ ... ukončovací znak

Příklad užití: TF@ – nastavení rychlého časového váhování, odesílání hodnot je zastaveno

- LX@
L ... nastavení váhování podle lidského vnímání (Loudness weighting)
X ... typ váhovacího filtru (A, B, C, D, Z)
@ ... ukončovací znak

Příklad užití: LC@ – nastavení C-váhování, odesílání hodnot je zastaveno

- AX@
A ... nastavení odesílání hodnot
X:
S ... vracena jedna celková hodnota
B ... vraceny hodnoty pro všechny pásma + celková hodnota
@ ... ukončovací znak

Příklad užití: AB@ – odesílání hodnot pro všechna pásma, odesílání hodnot je zastaveno

Vracené hodnoty jsou vraceny jako desetinná čísla s jedním desetinným místem. V případě vracení hodnot hladin hluku ze všech pásem, jsou jednotlivé hodnoty odděleny mezerou (ASCII: 32 (dec)). Po poslední hodnotě (výsledná) je oddělení provedeno pomocí odřádkování CR a LF (ASCII: CR = 12 (dec), LF = 10 (dec)). Komunikace byla testována na volně dostupném programu `Terminal.exe` [16].

Pokud je MCU v úsporném módu, je třeba brát na vědomí, že jej musíme nejdříve probudit. V případě SLEEP módu dojde k probuzení právě onou komunikací USART. V případě STOP módu je třeba použít nějaké externí přerušení (lze využít modré uživatelské tlačítko na desce). V případě STANDBY módu je nutné procesor buď resetovat, nebo opět využít uživatelské tlačítko.

3.1.2 Měření hlukoměru

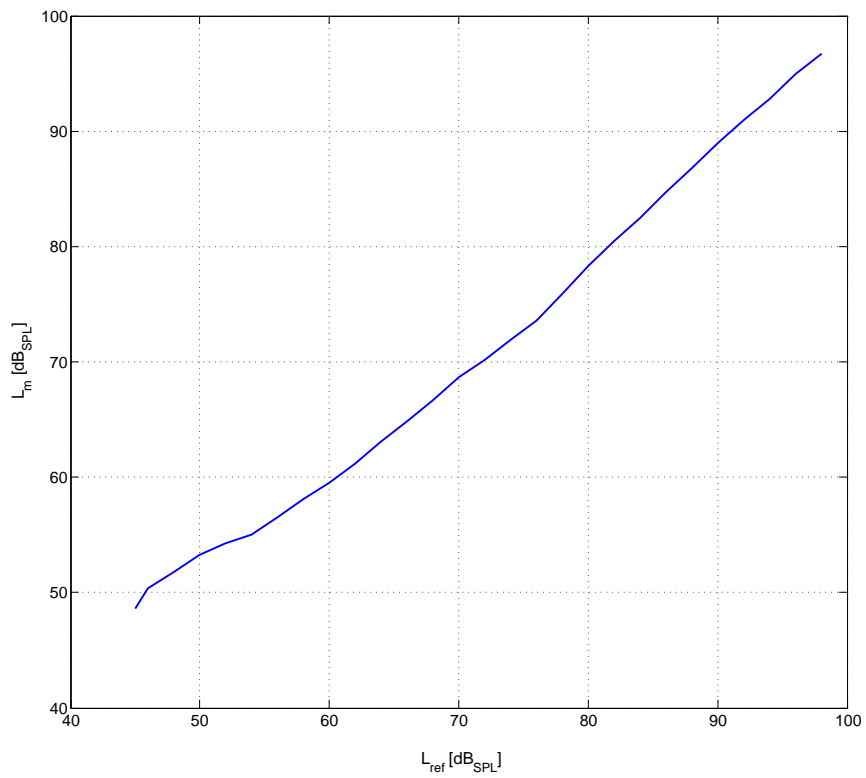
Vytvořenému hlukoměru byly proměřeny jednotlivé jeho části a zhodnoceny jejich parametry.

Měření přesnosti hlukoměru

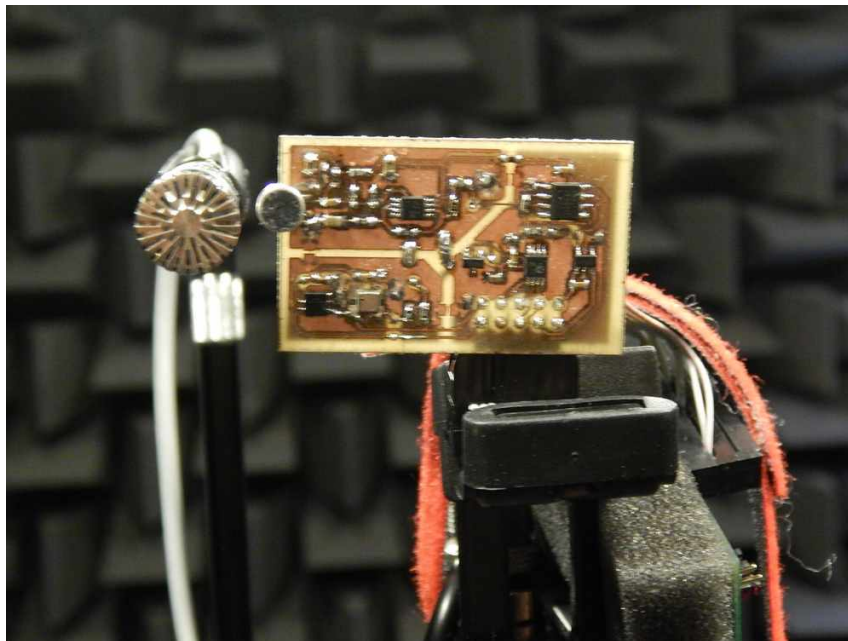
Zkonstruovaný hlukoměr byl proměřen v laboratorním prostředí předmětu *Studiová a hudební elektronika*. Hlukoměr byl umístěn do bezodrazové komory, kde byly jeho vlastnosti porovnány s referenčním laboratorním hlukoměrem. Oba hlukoměry byly umístěny ve stejné vzdálenosti od reproduktoru, tak aby ležely na stejné vlnoploše. Bylo provedeno kalibrační měření, kde byly naměřené hodnoty vytvořeného hlukoměru porovnány s hodnotami referenčními. Jako testovací zvuk byl použit růžový šum. Při měření bylo nastaveno váhování Z, tzn. že měřené hodnoty nepodléhají žádné korekci (váhování). Kalibrace byla provedena v oblasti 45 – 98 dB_{SPL} a je zanesena v grafu na obr.3.4

Dále byla změřena modulová kmitočtová závislost citlivosti hlukoměru při konstatním akustické tlaku 74 dB_{SPL}. Výsledná závislost byla zanesena do grafu na obr.A.8 v příloze. Použité přístroje jsou v tab.3.1

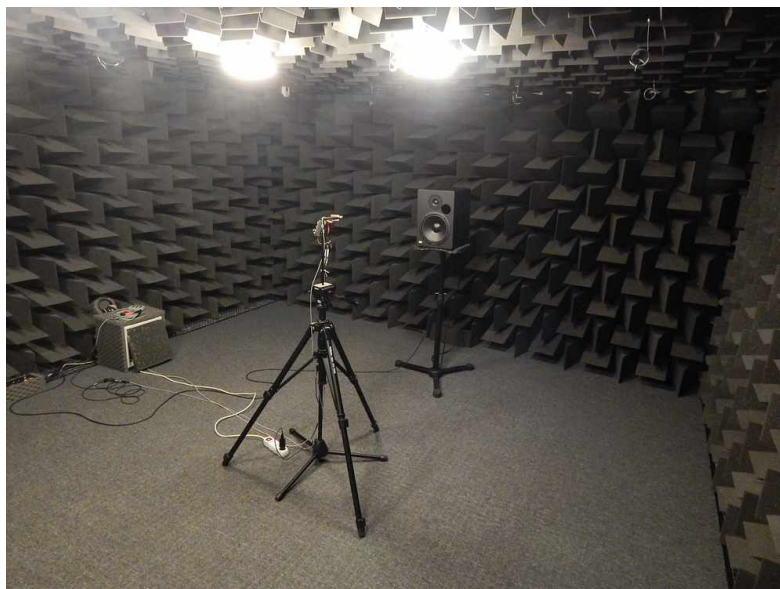
Následují fotografie z měření v bezodrazové komoře. Na obr.3.5 je kalibrační mikrofon (vlevo) a měřené hlukoměrné čidlo (vpravo). Na dalším obr.3.6 je zobrazena celá měřící soustava. Další obrázky z měření jsou umístěny v příloze A.6.



Obr. 3.4: Měření kalibrační křivky při použití růžového šumu



Obr. 3.5: Kalibrační mikrofon (vlevo) a hlukoměrné čidlo (vpravo)



Obr. 3.6: Měřicí sestava v bezodrazové komoře (2)

Tab. 3.1: Seznam použitých zařízení

Zařízení	Výrobce – Typ
Mikrofon	Brüerl Kjær – 4189
Mikrofonní předzesilovač	Brüerl Kjær – Nexus 2691
Analyzátor 1	Audio precision – APx 525
Analyzátor 2	NTI – XL2
Generátor	NTi – Minirator MR-PRO

Energetická náročnost hlukoměru

Všem částem hlukoměrné sestavy byl změřen proudový odběr. Proudový odběr byl změřen pro dvě frekvence jádra a pro tři různé úsporné módy. K měření byl využit ampérmetr Agilent 34410A. U měření proudového odběru MCU byly využito vyvedeného jumperu JP1, sloužícímu právě k tomuto účelu. Změřené hodnoty jsou v tab.3.2.

Nejmenší proudový odběr je ve STANDBY módu, ale „probuzení“ z tohoto módu je nejpomalejší a je proveden restart. Módy STOP a SLEEP po „probuzení“ restart neprovádí, ale mají větší proudový odběr. Frekvence procesoru nemá vliv na módy STOP a STANDBY, protože jsou oscilátory vypnuté. Díky tomu, že je hlukoměrné čidlo napájeno přes regulátor s možností vypnutí, je jeho spotřeba, v případě nevyužívání, nulová. V základu je nastaven SLEEP mód.

Tab. 3.2: Měření proudového odběru hlukoměrových komponent

Frekvence Procesoru 168 MHz:

	Spuštěný [mA]	SLEEP mód [mA]	STOP mód [mA]	STANDBY mód [mA]
Vývojový kit	92,4	70,2	46,4	45,75
MCU	47,8	26,1	2,02	0,818

Frekvence Procesoru 84 MHz:

Vývojový kit	69,73	58,3	46,4	46,6
MCU	24,21	14,36	1,948	0,82

Hlukoměrové čidlo:

	Zapnuté [mA]	Zapnuté + načítání hodnot [mA]	Vypnuté [mA]	
	3,17	3,45	0	

Komponenty hlukoměrového čidla:

Mikrofon [mA]	0,4			
Předzesilovač [mA]	1,9			
ADC převodník [mA]	0,28			

Měření rychlosti zpracování dat pomocí FPU

MCU má hardwarovou FPU (jednotku pro práci s čísly s plovoucí desetinou čárkou). To umožňuje značné zrychlení operací s těmito čísly. Bylo provedeno srovnání, jak rychle bude blok dat zpracován bez použití FPU, s využitím softwarové FPU a s využitím hardwarové FPU. Testovacím blokem dat bylo zpracování 1000 vzorků signálu, tzn. že testování proběhlo na reálném zpracování dat hlukoměrem. Výsledky jsou v tab.3.3.

Tab. 3.3: Porovnání rychlosti zpracování dat s FPU

Zpracování bloku dat (1000 vzorků) [ms]	
Bez FPU	224,8
Softwarová FPU	18,93
Hardwarová FPU	18,86

Jak lze vidět, hardwarová FPU poskytuje pouze nepatrně rychlejší výpočet oproti softwarové. Avšak porovnáme-li obě hodnoty s nepoužitím FPU, vidíme obrovský rozdíl. Při návrhu IIR filtrů bylo testováno, zda by filtrace nebyla rychlejší při práci

s datovým typem integer a bylo zjištěno, že nikoli. Z toho vyplývá že MCU s FPU dokáže pracovat rychleji s datovým typem float (32-bit) než s datovým typem integer (16-bit).

3.2 Zhodnocení hlukoměru

Z výsledků měření hlukoměru v kapitole 3.1.2 je patrné, že hlukoměr má oproti kvalitnímu laboratornímu hlukoměru jistou odchylku. Na obr3.4 lze vidět jeho korekční křivku, která udává odchylku od správné hodnoty hladiny hluku. V ideálním případě by křivka měla být lineární, se směrnicí rovnou jedné. Zde k mírným odchylkám dochází, nejvíce v nízkých hladinách hluku. To je způsobeno pravděpodobně šumem. Chyba měření je $\pm 2\text{dB}_{\text{SPL}}$.

Další, co má vliv na kvalitu měření hluku, je použitý elektretový mikrofon, jehož vlastnosti nejsou ideální a díky absenci obsáhlejší dokumentace, jsou některé vlastnosti neznámé.

Hlukoměr dokáže měřit hladiny hluku až do 98dB_{SPL} . To je dáno zvoleným vstupním rozsahem ADC převodníku a zesílením mikrofonního předzesilovače. Zesílení by mohlo být sice zmenšeno, ale tím více by se projevila chyba měření vlivem šumu.

Mikrofonnímu předzesilovači byly vylepšeny vlastnosti oproti tomu, který byl navržen v rámci semestrálního projektu. Jeho odstup signálu od šumu byl vylepšen o 10 dB.

Výsledný hlukoměr není samostatný systém, skládá se ze samostatného prototypu hlukoměrného čidla a samostatného vývojové kitu. To ale znamená, že ono hlukoměrné čidlo může být připojeno k jakémukoli zařízení podporujícímu komunikační protokol SPI.

Co se týče energetické náročnosti, dosahuje prototyp hlukoměrného čidla dobrých výsledků. Jeho proudový odběr je v aktivním stavu 3,45 mA a ve vypnutém stavu téměř nula. Tím lze dosáhnout značné úspory jeho vypínám ve stavu, kdy z něj nejsou načítány hodnoty. Oproti tomu vybraný MCU ve svých energeticky úsporných módech, nedosahuje tak dobrých hodnot. Při použití SLEEP módu klesne jeho spotřeba pouze na polovinu hodnoty, která je při zapnutém MCU. Pokud využijeme „hlubší“ módy úsporného režimu (STOP a STANDBY) jsou výsledky lepší, avšak MCU trvá déle „probuzení“ z těchto módů.

4 ZÁVĚR

V rámci této práce byla popsána teorie snímání zvuku. Byl popsán akustický tlak a jeho šíření v prostoru, to jak vnímá zvuk lidské ucho a principy jeho měření. Dále byla popsána použitá vývojová platforma a číslicové filtry, využití k realizaci hlukoměru.

Byla navržena a vytvořena hlukoměrná sestava chovající se jako digitální čidlo, které vrací hodnoty hladiny hluku pomocí sériové komunikace USART. Získávání hodnot je realizováno pomocí předdefinovaných příkazů a umožňuje zvolit počet vrácených hodnot, nastavit typ časového váhování (Fast, Slow, Impulse), typ váhovacího filtru (A, B, C, D, Z) a to, jestli má být vrácena pouze výsledná hodnota hladiny hluku, nebo i hodnoty v jednotlivých pásmech.

Parametry hlukoměrného čidla a celého hlukoměru byly proměřeny v laboratorním prostředí a zaneseny do této práce. Tyto měření umožnily posoudit přesnost hlukoměru a kvalitu mikrofonního předzesilovače.

Tvorba hlukoměru byla rozdělena do dvou částí. V první části byl navržen prototyp hlukoměrného čidla. Čidlo obsahuje elektretový mikrofon a mikrofonní předzesilovač. Tento předzesilovač je postaven na operačním zesilovači v neinvertujícím zapojení. Výsledný zesílený signál je offsetován na stejnosměrnou složku o polovině rozsahu ADC převodníku. Signál z předzesilovače je přiveden na vstup ADC převodníku, kde je signál vzorkován a výsledné hodnoty jsou odesílány do vývojového kitu, kde v mikrokontroléru probíhá zpracování těchto hodnot.

Pro vzorkování signálu byl použit 16-bitový ADC převodník AD7683. Hodnoty konverze převodník odesílá pomocí komunikace SPI. Jako vývojový kit byl zvolen STM32F4 Discovery od výrobce STMicroelectronics, obsahující mikrokontroler STM32F407VGT6, postavený na procesoru ARM Cortex[®]-M4F.

V druhé části byl vytvořen ovladač hlukoměru. Data jsou v mikrokontroléru načítána a zpracovávána po blocích (zvoleno 1000 vzorků). Během načítání onoho bloku dat, je prováděno zpracování předchozího bloku dat, tím nedochází ke ztrátě žádných vzorků. Během zpracování je provedena digitální filtrace signálu oktávovou bankou filtrů. Tím je vstupní signál rozdělen do pásem. Pro jednotlivá pásma je provedeno časové váhování, čímž je získána efektivní hodnota signálu. Z této hodnoty jsou vypočteny jednotlivé hladiny hluku v daných pásmech. Poté je provedena korekce těchto hodnot podle váhovacích filtrů, které odrážejí citlivost lidského ucha na jednotlivé frekvence hluku. Následně je vypočtena celková hladina hluku.

Výsledné zhodnocení hlukoměru je uvedeno v předchozí kapitole 3.2.

Celkově tato práce umožnila nahlédnout do problematiky snímání zvuku a jeho zpracování. Dále umožnila osvojení technik pro programování mikrokontrolérů architektury ARM. Také umožnila pochopení technik pro návrh a implementaci hardwa-

rových komponent hlukoměrného čidla, tj. návrh, výroba a osazení desky plošných spojů.

LITERATURA

- [1] NOVÝ, Richard. *Hluk a chvění*. 3. vyd. V Praze: České vysoké učení technické, 2005, 141 s. ISBN 978-80-01-04347-9.
- [2] SCHIMMEL, Jiří. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. *Elektroakustika* [online]. 1. vyd. Brno, 2013 [cit. 2014-12-14]. ISBN 978-80-214-4716-5. Dostupné z: <https://www.vutbr.cz/www_base/priloha.php?dpid=71624>.
- [3] JAKSCH, Ivan. TECHNICKÁ UNIVERZITA V LIBERCI. *Číslicové měřicí systémy: Číslicové filtry* [online]. Liberec, 2012 [cit. 2015-05-20]. Dostupné z: <http://www.rss.tul.cz/download/cms/11_ESF_cislic_filtry.pdf>. Učební text.
- [4] MITRA, Sanjit Kumar. *Digital signal processing: a computer-based approach*. 2nd ed. Boston: McGraw-Hill/Irwin, 2001, xviii, 866 p. ISBN 00-725-2261-5.
- [5] CMSIS-DSP: *Infinite Impulse Response (IIR) Lattice Filters*. Keil, Tool by ARM [online]. 2015 [cit. 2015-05-20]. Dostupné z: <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__i_i_r__lattice.html>.
- [6] ST: *STM32F4 Discovery* [online]. [cit. 2015-05-18]. Dostupné z: <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1848/PF252419?s_searchtype=partnumber>.
- [7] Analog Devices: *ADA4528 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADA4528-1_4528-2.pdf>.
- [8] Analog Devices: *AD7683 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <<http://www.analog.com/media/en/technical-documentation/data-sheets/AD7683.pdf>>.
- [9] Texas Instruments: *REF3125 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <<http://www.ti.com/lit/ds/symlink/ref3125.pdf>>.
- [10] Analog Devices: *ADA4841 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <<http://www.analog.com/en/ADA4841-1>>.
- [11] *Byte Paradigm sprl: Introduction to I₂C and SPI protocols* [online]. 2015 [cit. 2015-05-23]. Dostupné z: <<http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>>.

- [12] Sensirion AG: *SHT21 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT21_Datasheet_V4.pdf>.
- [13] Measurement Specialties™: *MS5637-02BA03 Datasheet* [online]. [cit. 2015-05-21]. Dostupné z: <<http://www.meas-spec.com/downloads/MS5637-02BA03.pdf>>.
- [14] HANÁK, Jan. VUT BRNO. *Návrh plošných spojů v Cadsoft Eagle 6* [online]. Brno, 2014 [cit. 2015-05-25]. ISBN 978-XXXXXXXXXX. Dostupné z: <<https://www.vutbr.cz/elearning/mod/resource/view.php?id=303715>>.
- [15] MAJERLE, Tilen. *STM32F4 Discovery: Power consumption modes for STM32F4* [online]. 2015 [cit. 2015-05-24]. <<http://stm32f4-discovery.com/2014/10/library-39-power-consumption-modes-for-stm32f4/>>.
- [16] HW-SERVER *TERMINAL - terminal emulation program for RS-232: Terminal.exe*. [online]. 2004 [cit. 2015-05-23]. Dostupné z: <<http://hw-server.com/terminal-terminal-emulation-program-rs-232>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A	Zesílení (zisk) – Gain
a_i	koeficienty filtru v dopředných vazbách – feedback filter coefficients
ADC	analogově digitální převodník – Analog-to-Digital Converter
ARM	pokročilý RISC procesor – Advanced RISC Machine
α	časová konstanta filtru – filter time constant
b_i	Systémové koeficienty ve zpětných vazbách – Feedforward filter coefficients
CPHA	Fáze hodinového signálu – Clock Phase
CPOL	Polarita hodinového signálu – Clock Polarity
DAC	Digitálně analogový převodník – Digital to Analog Converter
DMA	Přímý přístup do paměti – Direct Memory Access
DPS	Deska plošných spojů – printed circuit board
DSP	Digitální signálový procesor – Digital signal processor
η	citlivost mikrofonu – microphone sensitivity
f	frekvence – frequency
f_{vz}	Vzorkovací frekvence – Sampling frequency
F	Rychlé časové váhování – Fast Time weighting
FPU	Matematický koprocessor – Floating Point Unit
f_{Si}	střední kmitočet filtru – Medium frequency of filter
$F(t)$	síla – power
I	Impulzní časové váhování – Impulse time weighting
I ₂ C	multi-masterová počítačová sériová sběrnice – Inter-Integrated Circuit
IDE	Vývojové prostředí – Integrated Development Environment
IIR	Filtr s nekonečnou impulzní odezvou – Infinite Impulse Response

L	hladina – level
L_p	hladina akustického tlaku – Sound Pressure Level
$L_{\delta f}$	hladina akustického tlaku pro jednotlivé filtry banky filtrů – Sound Pressure Level for individual filters of filter bank
λ	vlnová délka – wavelength
MCU	Mikrokontrolér – Microcontroller Unit
MEMS	Mikro-elektronický mechanický systém – Micro-Electro-Mechanical Systems
MISO	Master vstup, Slave výstup – Master In, Slave Out
MOSI	Master výstup, Slave vstup – Master OUT, Slave In
OZ	Operační zesilovač – Operational amplifier
ω	úhlový kmitočet – angular frequency
p_C	celkový akustický tlak – total pressure
$p(t)$	akustický tlak – sound pressure
p_{00}	statický barometrický tlak – static/atmospheric pressure
RAM	operační paměť – Random-Access Memory
RX	příjmač – receiver
S	plocha – area
S	Pomalé časové váhování – Slow time weighting
SCLK	Sériový hodinový signál – Serial Clock
SNR	Odstup signál-šum – Signal to Noise Ratio
SPI	seriové rozhraní pro periferie – Serial Peripheral Interface
SS	Výběr Slave zařízení – Slave Select
T	perioda – period
THD	Celkové Harmonické zkreslení – Total harmonic distortion

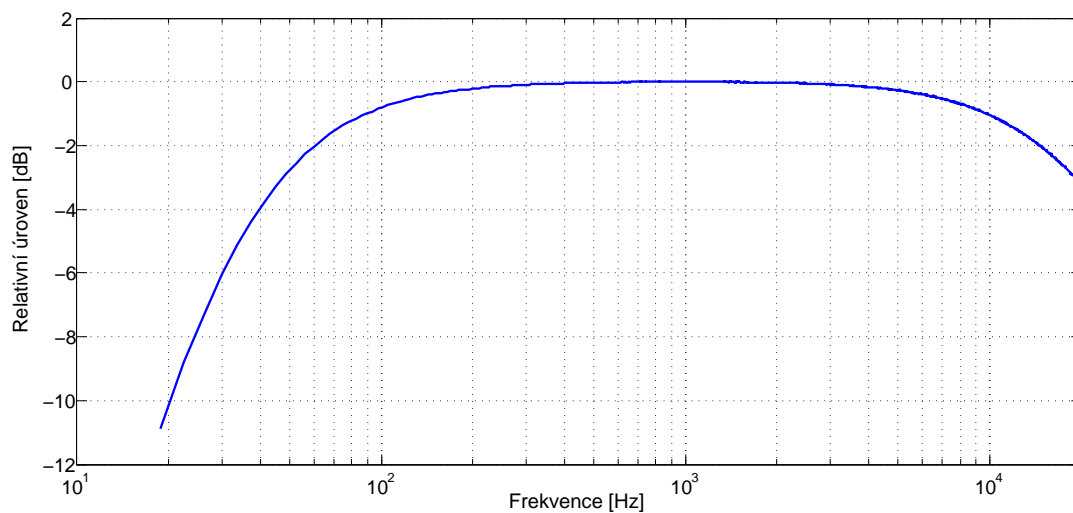
THD + N	Celkové Harmonické zkreslení zahrnující šum – Total harmonic distortion plus noise
TX	vysílač – transmitter
u_{ef}	Efektivní napětí – Effective voltage
USART	Synchronní / asynchronní sériové rozhraní USART – Universal Synchronous / Asynchronous Receiver and Transmitter
USB	univerzální seriové rozhraní – Universal Serial Bus

SEZNAM PŘÍLOH

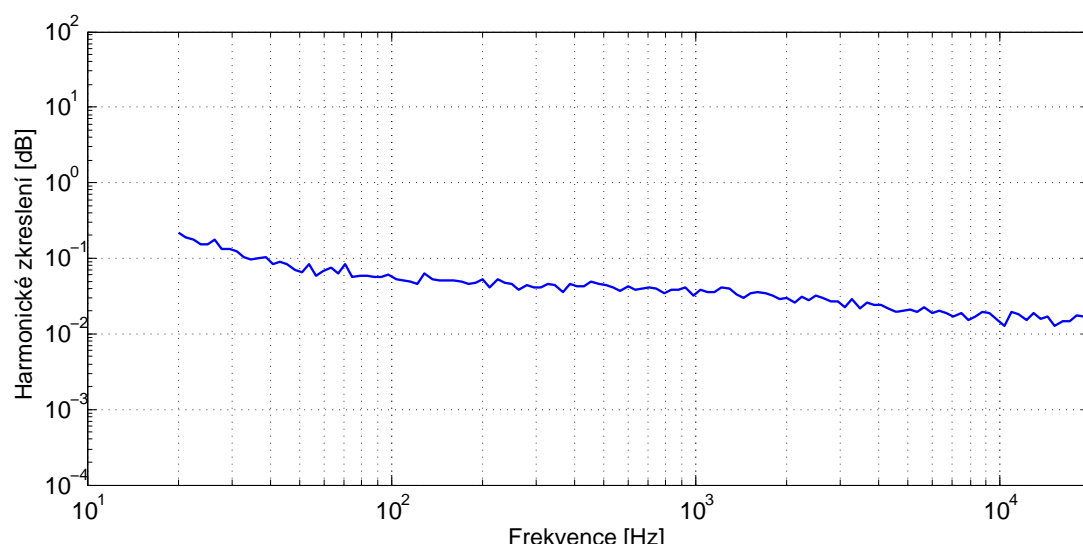
A Příloha	58
A.1 Měření předzesilovače	58
A.2 Schéma DPS hlukoměrového čidla	60
A.3 Seznam součástek	61
A.4 Vývojový diagram ovladače hlukoměru	62
A.5 Návrh IIR filtrů v FDATool	63
A.6 Měření hlukoměru	64
A.7 Instalace vývojového kitu STM32F4 Discovery	67
A.7.1 USB ovladač STM32F4 Discovery	67
A.7.2 Překladač pro STM32F407VG	67
A.7.3 Vývojové prostředí pro STM32F4 Discovery	68
A.7.4 CooCox IDE – Založení nového projekt	68
A.8 Použití hardwarové FPU	69
A.9 Použití DSP knihovny	69
A.10 Připojení hlukoměru	71
A.11 Popis pinů hlukoměru	72
A.12 DVD	72

A PŘÍLOHA

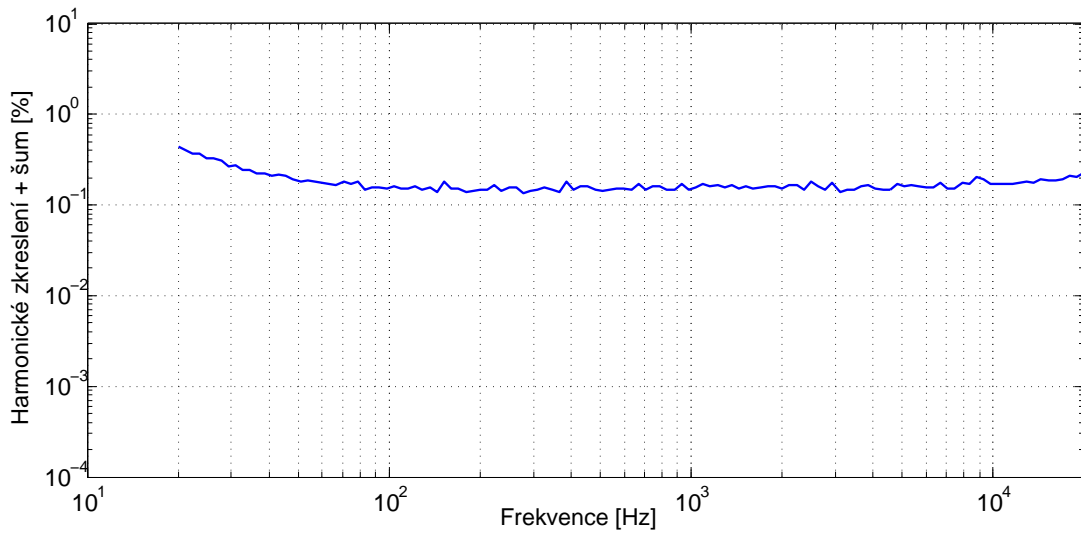
A.1 Měření předzesilovače



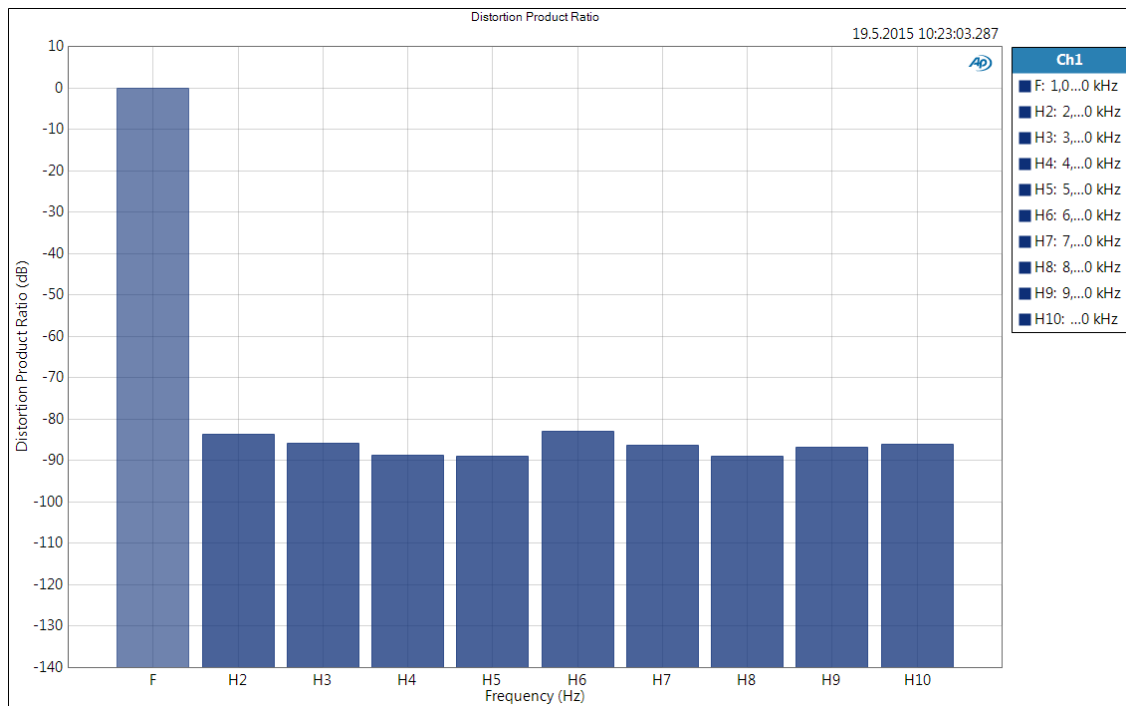
Obr. A.1: Relativní zesílení vztahované k referenční hodnotě 1kHz (Relative Level (1kHz))



Obr. A.2: Celkové Harmonické zkreslení (Total harmonic distortion – THD)

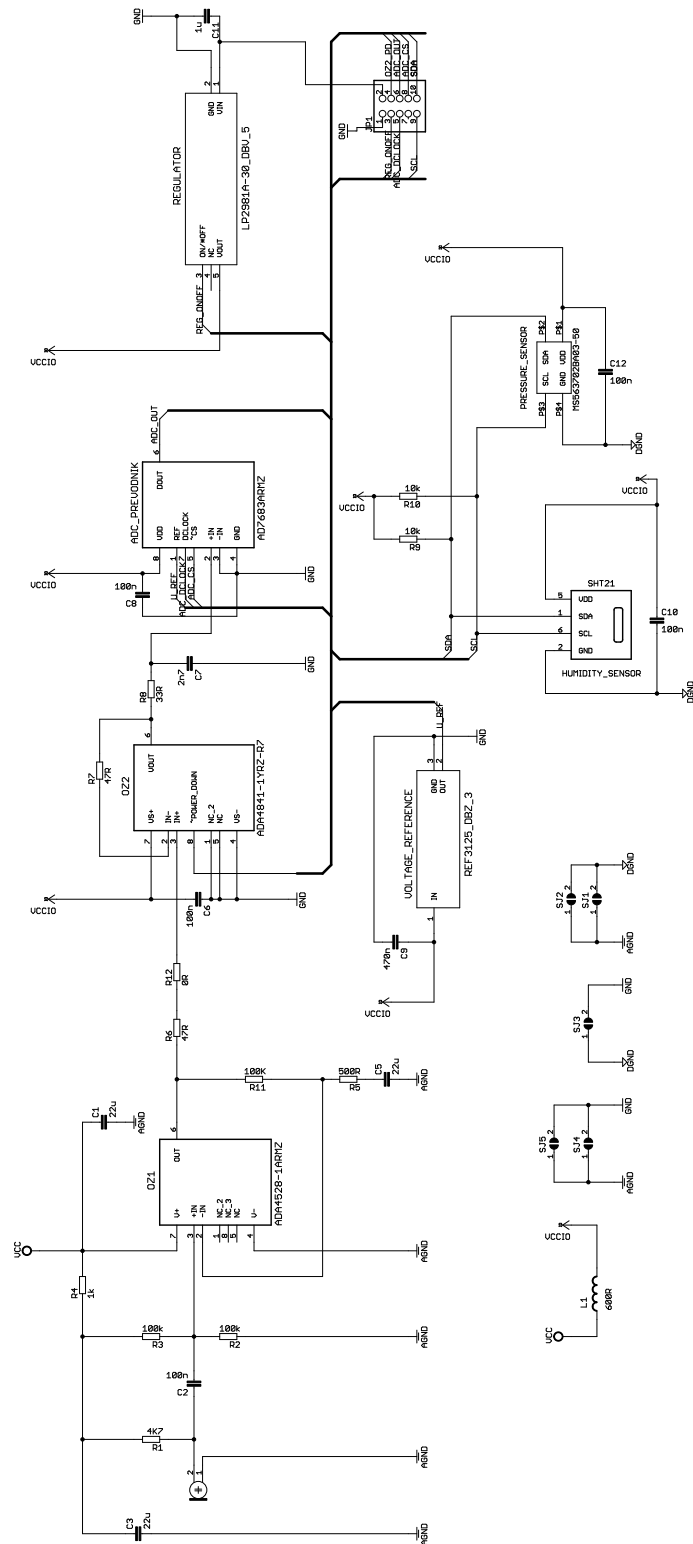


Obr. A.3: Celkové harmonické zkreslení zahrnující šum (Total harmonic distortion plus noise – THD+N)



Obr. A.4: Harmonické zkreslení druhou až desátou harmonickou složkou (Distortion Product Ratio)

A.2 Schéma DPS hlukoměrového čidla



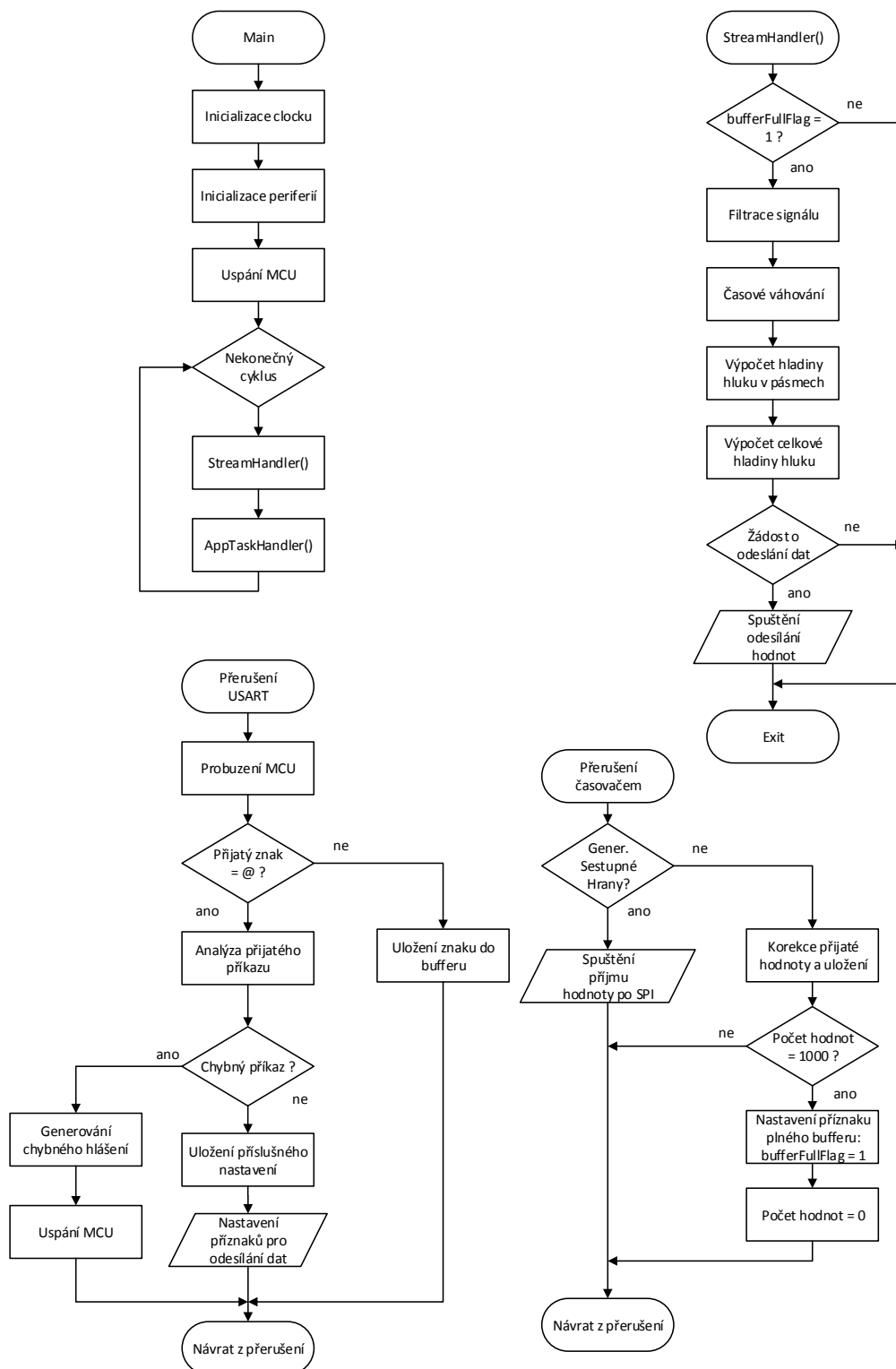
Obr. A.5: Schéma DPS hlukoměrového čidla

A.3 Seznam součástek

Tab. A.1: Seznam použitých součástek hlukoměrného čidla

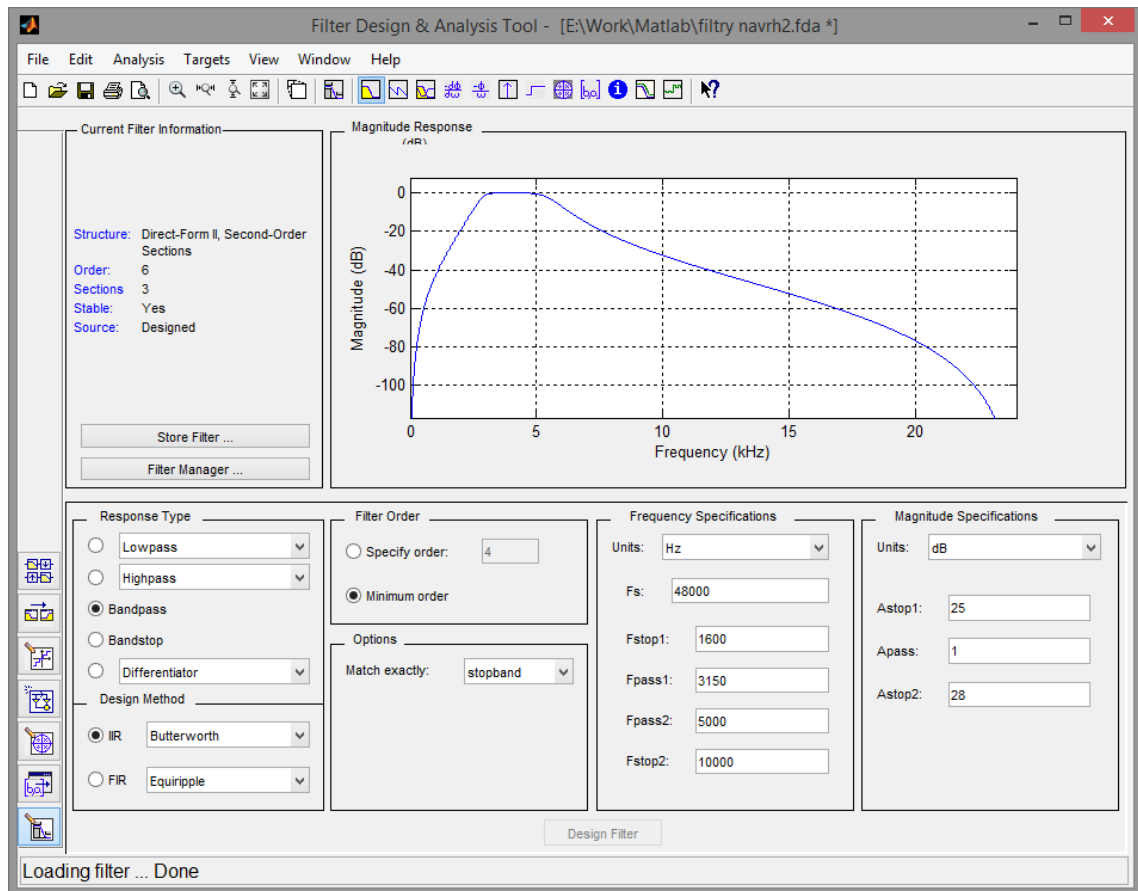
Part	Value	Device	Package	Description
ADC	AD7683ARMZ	AD7683ARMZ	SOP65P490X110-8N	16-Bit, 100 kSPS, Single-Ended PulSAR ADC in MSOP
C1	22u	C-EUC0805	C0805	CAPACITOR, European symbol
C2	100n	C-EUC0603	C0603	CAPACITOR, European symbol
C3	22u	C-EUC0805	C0805	CAPACITOR, European symbol
C5	22u	C-EUC0805	C0805	CAPACITOR, European symbol
C6	100n	C-EUC0603	C0603	CAPACITOR, European symbol
C7	2n7	C-EUC0603	C0603	CAPACITOR, European symbol
C8	100n	C-EUC0603	C0603	CAPACITOR, European symbol
C9	470n	C-EUC0603	C0603	CAPACITOR, European symbol
C10	100n	C-EUC0603	C0603	CAPACITOR, European symbol
C11	1u	C-EUC0603	C0603	CAPACITOR, European symbol
C12	100n	C-EUC0603	C0603	CAPACITOR, European symbol
JP1		PINH-D-2X5/90	2X05/90	PIN HEADER
L1	600R	INDUCTOR0603	0603	Inductors
MS563702BA03-50		ISL29102	ODFN6	
OZ1		ADA4528-1ARMZ	SOP65P490X110-8N	Zero-Drift Operational Amplifiers
OZ2		ADA4841-1YRZ-R7	SOIC127P600X175-8N	Low Power, Low Noise and Distortion, Rail-to-Rail Output Amplifier
R1	4K7	R-EU_R0603	R0603	RESISTOR, European symbol
R2	100k	R-EU_R0603	R0603	RESISTOR, European symbol
R3	100k	R-EU_R0603	R0603	RESISTOR, European symbol
R4	1k	R-EU_R0603	R0603	RESISTOR, European symbol
R5	500R	R-EU_R0603	R0603	RESISTOR, European symbol
R6	47R	R-EU_R0603	R0603	RESISTOR, European symbol
R7	47R	R-EU_R0603	R0603	RESISTOR, European symbol
R8	33R	R-EU_R0603	R0603	RESISTOR, European symbol
R9	10k	R-EU_R0603	R0603	RESISTOR, European symbol
R10	10k	R-EU_R0603	R0603	RESISTOR, European symbol
R11	100K	R-EU_R0603	R0603	RESISTOR, European symbol
R12	0R	R-EU_R0603	R0603	RESISTOR, European symbol
REG		LP2981A-30_DBV_5	DBV5	
SHT21		SHT21	DFN6/SHT	Humidity and Temperature Sensor
SJ1		SJ	SJ	SMD solder JUMPER
SJ2		SJ	SJ	SMD solder JUMPER
SJ3		SJ	SJ	SMD solder JUMPER
SJ4		SJ	SJ	SMD solder JUMPER
SJ5		SJ	SJ	SMD solder JUMPER
U\$1		MCE-2500		
VREF		REF3125_DBZ_3	DBZ3	

A.4 Vývojový diagram ovladače hlukoměru



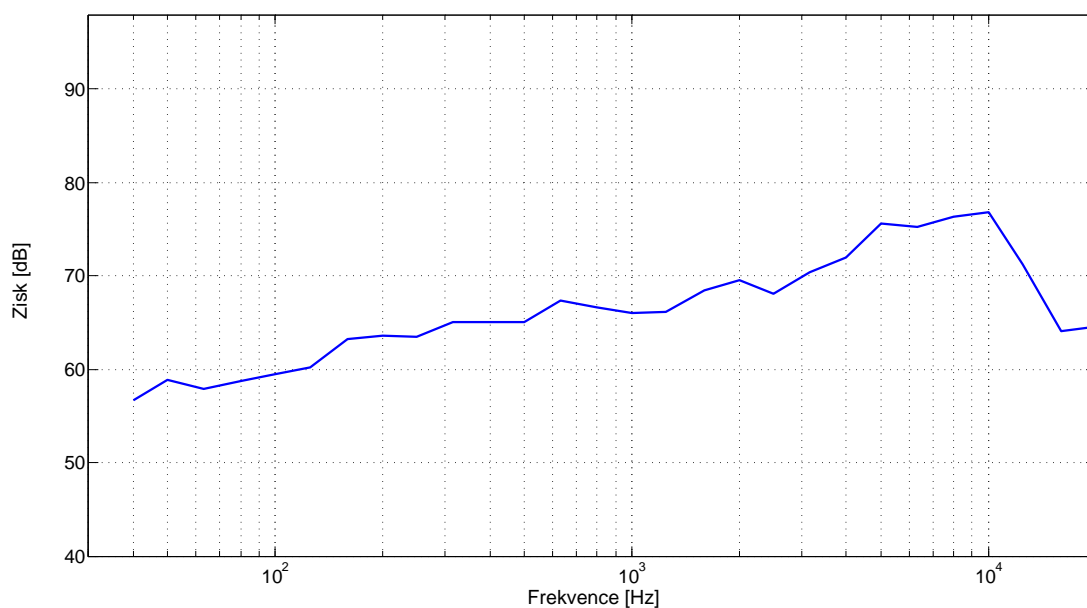
Obr. A.6: Vývojový diagram ovladače hlukoměru

A.5 Návrh IIR filtrů v FDATool



Obr. A.7: Návrh IIR filtru v programu FDATool

A.6 Měření hlukoměru



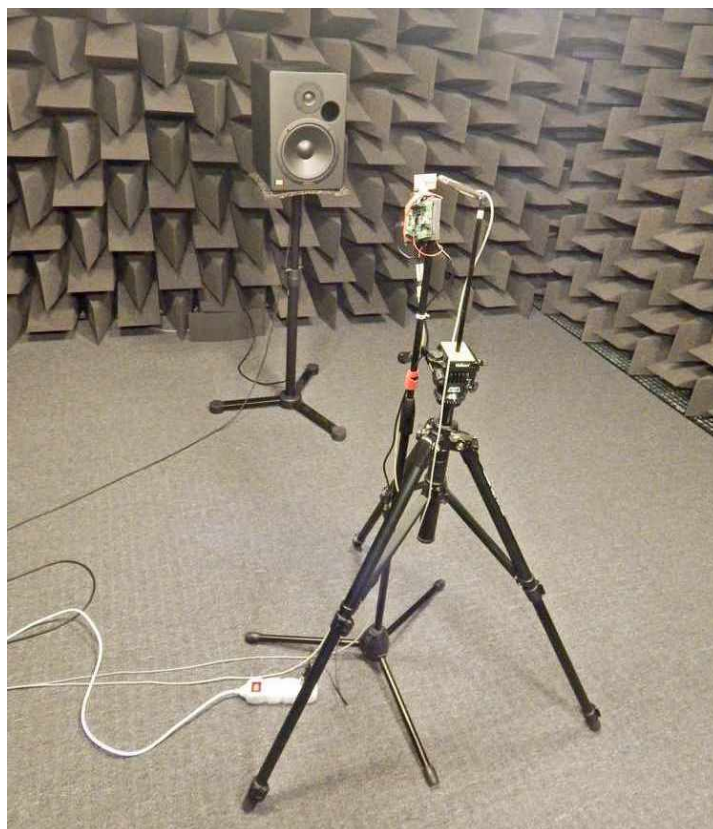
Obr. A.8: Modulová kmitočtová závislost citlivosti hlukoměru při konstantním akustickém tlaku $74 \text{ dB}_{\text{SPL}}$



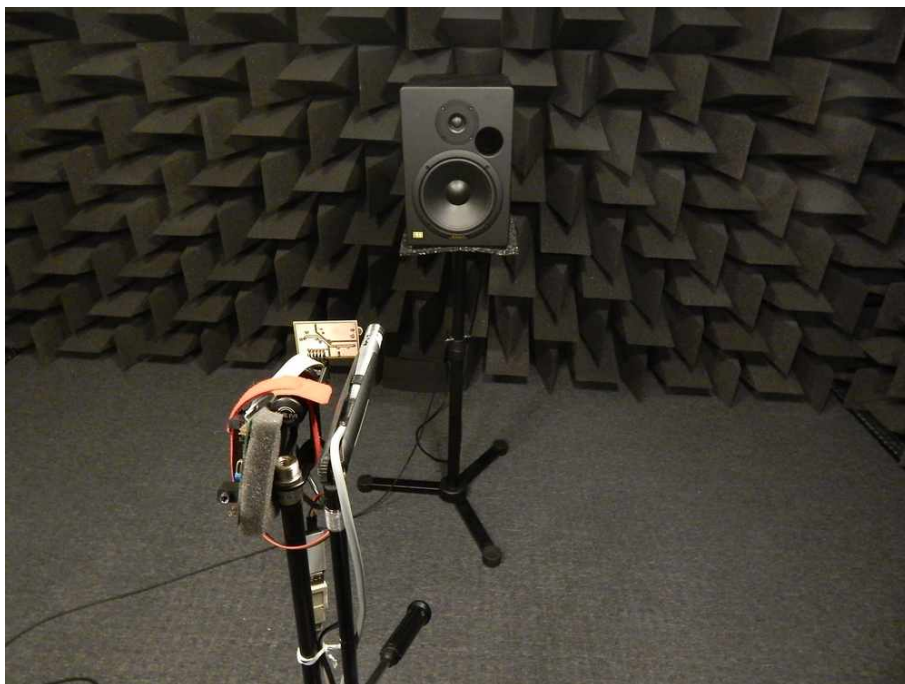
Obr. A.9: Použité měřící přístroje, analyzátor (vlevo) a generátor zvuku (vpravo)



Obr. A.10: Laboratoř Studiové a hudební elektroniky ve které probíhalo měření



Obr. A.11: Měřící sestava v bezdrazové komoře (1)



Obr. A.12: Měřící sestava v bezodrazové komoře (3)

A.7 Instalace vývojového kitu STM32F4 Discovery

A.7.1 USB ovladač STM32F4 Discovery

Driver hlukoměru byl vyvíjen na OS Microsoft Windows 8.1. Pro programování kitu STM32F4 Discovery je nejdříve nutné nainstalovat USB ovladače. Instalátor je dostupný na stránkách výrobce STMicroelectronics. Jméno souboru je „ST-LINK/V2 USB driver“, odkaz: <<http://www.st.com/web/en/catalog/tools/PF258167#>>.

Nejdříve je třeba nainstalovat ovladače a až poté připojit kit k PC. V případě instalace na Windows verzi 7 a nižší, pravděpodobně nebude s instalací ovladačů problém. V případě Windows verze 8 a 8.1 nastane problém s ověřením podpisu ovladačů. Proto je nutné ověření podpisu vypnout. Způsobů jak to provést je více, zde je popsán postup přes Aplikaci Metro:

1. Stiskněte kombinaci kláves Win + C poté: „Nasatvení“ -> „Změnit nastavení počítače“ -> „Aktualizace a obnovení“ -> „Obnovení“ -> „Spuštění s upřesněným nastavením“ -> „Restartovat“.
2. Počítač bude restartován a nabootuje do speciálního menu kde vyberete: „Zvolte možnost“ -> „Odstranit potíže“ -> „Upřesnit možnosti“ -> „Nastavení spouštění“ -> „Restartovat“.
3. Počítač je znovu restartován a zobrazí se nabídka. Zvolte nabídku číslo 7 (klávesou F7) Zakázat vynucení podpisu ovladače.
4. Počítač se znovu restartuje a nyní už můžete ovladače nainstalovat. Pouze musíte odsouhlasit, že chcete ovladače nainstalovat i přes to, že nejsou podepsané. Po dalším restartu se ověřování podpisů opět zapne.

Po instalaci můžeme již vývojový kit připojit k PC. Nyní už by měla komunikace fungovat. To lze poznat tak, že červená dioda na desce nebude blikat. Pokud bude blikat, značí to, že komunikace stále nefunguje, toto lze vyřešit instalací „STM32 ST-LINK Utility“ odkaz: <<http://www.st.com/web/en/catalog/tools/PF258168>>, kde se poté zkusíte připojit k zařízení (toto pomohlo při instalaci autorovi).

A.7.2 Překladač pro STM32F407VG

Jako překladač byl zvolen „GNU Tools ARM Embedded“ verze 4.9 2014q4, odkaz: <<https://launchpad.net/gcc-arm-embedded/+download>>. Stažený soubor nainstalujte.

A.7.3 Vývojové prostředí pro STM32F4 Discovery

K programování MCU byl využito vývojové prostředí CoCox CoIDE v1.7.8, odkaz: <http://www.cocox.org/download/Tools/CoIDE-1.7.8.exe>, které je poskytováno zdarma. Nejnovější verze tohoto programu je verze 2, která je však ve fázi beta testování. Proto byla k vývoji využita stabilní verze 1, která je i více uživatelsky přívětivá (osobní názor autora).

Nainstalujte CoCox pomocí staženého instalátoru a spusťte. Nejprve je nutné nastavit cestu k překladači „GNU Tools ARM Embedded“, který byl nainstalován v předchozím kroku. Zvolte „Project“ → „Select Toolchain Path“. Poté vyberte cestu k adresáři se spustitelným souborem překladače. V mém případě to bylo `C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2014q4\bin`.

A.7.4 CoCox IDE – Založení nového projektu

Nyní je provázání vývojového prostředí s překladačem hotovo a můžete začít vytvořením nového projektu. Zvolte: „Project“ → „New Project“, a nastavte jméno projektu a místo kde bude vytvořen. Pokračujte stiskem „Next“, poté zvolte „Chip“. Nyní vyberte: „ST“ → „STM32F4x“ → „STM32F407VG“. Pokračujte kliknutím na Finish. Nyní je vytvořen nový projekt a automaticky se otevře nabídka Repository. Kde si můžete navolit knihovny pro periferie, které budete využívat. Minimálně je nutné vybrat knihovnu „CMSIS BOOT“.

Předtím než zkusíte první projekt, je nutné správně nastavit zdroj a frekvenci systémových hodin (clock). Toto nastavení lze provést manuálně v systémových souborech, nebo využít nástroj „Clock configuration tool“ od výrobce ST, odkaz: <http://www.st.com/web/en/catalog/tools/PF257927>. Kde lze přehledně nastavit zdroj systémových hodin který chceme použít a na jaké frekvenci. Zdrojem systémových hodin může být buď interní RC oscilátor (HSI) nebo externí krystalový oscilátor. Pozor, hodnota interního oscilátoru je přednastavena správně, ale hodnota externího oscilátoru nikoli. Je třeba ji nastavit na 8 MHz (takový oscilátor je na STM32F4 Discovery). Poté se nastaví hodnota frekvence procesoru (maximum je 168 MHz).

Poté kliknete na tlačítko „Run“, tím dojde k výpočtu všech hodnot. Výsledný soubor se vygeneruje klikem na tlačítko „Generate“. Soubor se vygeneruje do stejného umístění, jako je tento konfigurační nástroj. Vygenerovaný soubor „system_stm32f4xx.c“ zkopírujte do adresáře „cmsis_boot“ v umístění projektu a nahraďte jím původní soubor. Poté je nutné v souboru „stm32f4xx.h“ změnit hodnotu `HSE_VALUE` z 25000000 na 8000000.

Nakonec pro aplikaci všech těchto nastavení je nutné využít funkce „SystemInit()“, jejíž volání je třeba umístit do souboru „startup_stm32f4xx.c“ do funkce

`void Default_Reset_Handler(void)` před volání funkce `main()`. Volání této funkce poté umístíte i do funkce `main()` v souboru `main.c`.

A.8 Použití hardwarové FPU

Pro použití hardwarové nebo softwarové FPU je nutné ji zapnout (dát překladači příkaz k jejímu zapnutí) a nalinkovat příslušné knihovny. Ve vývojovém prostředí CoCoX CoIDE to lze provést následovně:

V prostředí CoIDE kliknout pravým tlačítkem myši na kořenový adresář projektu. Poté zvolit „Configuration“ -> záložka „Compile“ -> z rozevírací nabídky zvolit „FPU soft“ nebo „FPU hard“. V okně „Defined Symbols“ se automaticky objeví `__FPU_USED`.

Nyní v kartě „Link“ -> okno „Linked Libraries“ -> „Add“ je třeba nalinkovat příslušné knihovny z adresáře, kde je nainstalován překladač. Pro:

- FPU soft:
 - `C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2014q4\arm-none-eabi\lib\armv7e-m\softfp\libm.a`
- FPU hard:
 - `C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2014q4\arm-none-eabi\lib\armv7e-m\fpv\libm.a`

Nyní je FPU zapnuta.

A.9 Použití DSP knihovny

Pro využití DSP je nutné do projektu naimportovat příslušné knihovny. Celý balík DSP knihoven a funkcí lze stáhnout ze stránek ARM[®]. Jméno souboru je `CMSIS-SP-00300-r4p3-00rel0.zip`, odkaz: <<https://silver.arm.com/browse/CMSIS>>. Pro stažení je nutné se přihlásit (registrace zdarma). Balík je nutné rozbalit. Poté se může umístit kamkoli, protože v projektu v CoIDE se pouze nalinkují cesty k příslušným knihovnám. Nebo ho umístit přímo do adresáře projektu.

DSP předpokládá využití FPU, proto je nutné ji nejdříve zapnout a to je popsáno v předchozí kapitole.

V prostředí CoIDE kliknout pravým tlačítkem myši na kořenový adresář projektu. Poté zvolit „Configuration“ -> záložka „Compile“ -> „Includepaths“ -> „Add“ a přidat tyto 2 cesty:

- `.../CMSIS-SP-00300-r4p3-00rel0/cmsis/include`

- .../CMSIS-SP-00300-r4p3-00rel0/arm/armcm4/include

Poté ověřit zda v okně „Defined Symbols“ jsou `__FPU_USED` a `__FPU_PRESENT`. Pokud ne, je nutno je přidat.

Nyní v kartě „Link“ -> okno „Linked Libraries“ -> „Add“ je třeba nalinkovat knihovnu:

```
...\\CMSIS-SP-00300-r4p3-00rel0\\CMSIS\\Lib\\GCC\\libarm_cortexM4lf_math.a
```

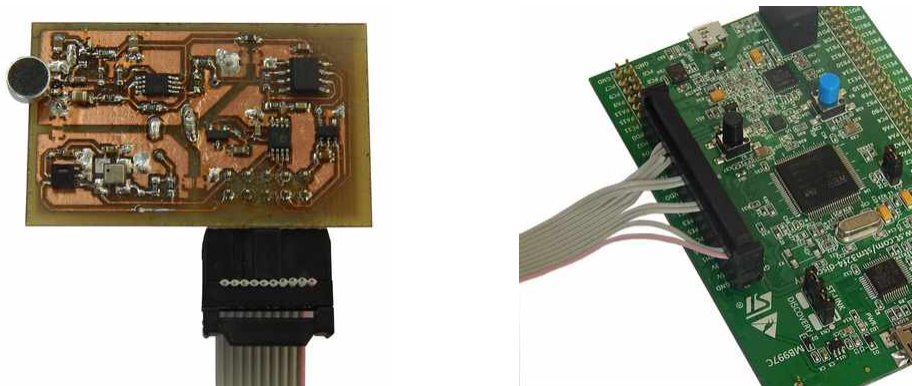
A nyní přidat tyto dva řádky do souboru `main.c` :

```
#define ARM_MATH_CM4  
#include <arm_math.h>
```

Pro použití některé z DSP před-připravených funkcí je nutné je přidat do projektu. Přidat do projektu je také nutné soubory `math_helper.h` a `math_helper.c` které jsou v adresáři `C:\\Work\\CooCox\\SOUND_LEVEL_METER\\DSP_lib\\CMSIS\\DSP_Lib\\Examples\\Common\\Include` a `Source`.

A.10 Připojení hlukoměru

Připojení hlukoměru je možné provést pomocí kabelu obr.A.13 nebo redukce obr.A.14.



Obr. A.13: Připojení hlukoměrového čidla pomocí kabelu



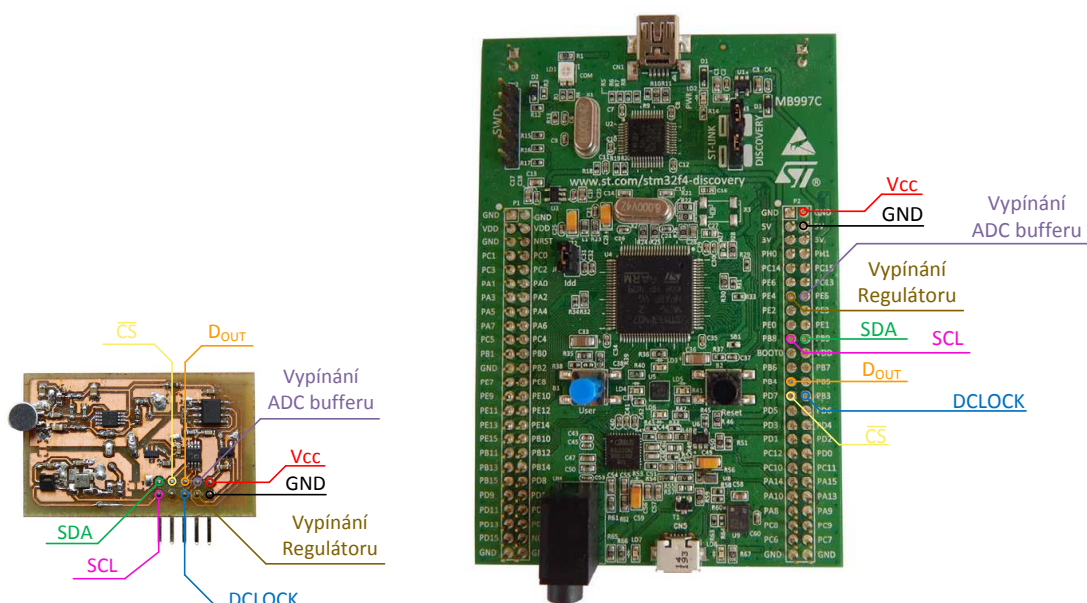
Obr. A.14: Připojení hlukoměrového čidla pomocí redukce

Při propojování je třeba dbát na správné zapojení. Při přepólování by mohlo dojít ke zničení hlukoměrového čidla.

Komunikace USART je nastavena na pinech:

- PA2: vysílání – TX
- PA3: příjem – RX

A.11 Popis pinů hlukoměru



Obr. A.15: Popis pinů hlukoměru

Tab. A.2: Tabulka pinů hlukoměru

Pin na hlukoměrném čidle	Pin na STM32F Discovery	Popis
V _{CC}	5 V	Napájení 5 V
GND	GND	Zem
Vypínání regulátoru	PE4	Vypínání čidla
Vypínání ADC bufferu	PE5	
SCL	PB8	I ² C hodinový signál
SDA	PB9	I ² C data
D _{OUT}	PB4	SPI data
DCLOCK	PB3	SPI hodinový signál
$\overline{\text{CS}}$	PD7	SPI spuštění konverze

A.12 DVD

Na přiloženém DVD byly umístěny následující soubory:

- Elektronická verze práce
- Coocox projekt

- Eagle projekt DPS hlukoměrného čidla
- Nyvržené filtry ve FDATool
- Program Terminal.exe
- Soubory pro STM32F4 Discovery:
 - Coocox CoIDE v1.7.8 instlátör
 - překladač v4.9 2014q4
 - STM32 ST-LINK Utility v3.5
 - STM32F4xx Clock Configuration v1.1.0
 - USB Driver v16.0.0.435
 - Dukomentace STM32F4
 - DSP Library 00300-r3p1-00rel0

Do Informačního systému byly odevzdány tyto soubory:

- Coocox projekt
- Eagle projekt DPS hlukoměrného čidla
- Nyvržené filtry ve FDATool
- Program Terminal.exe

Aby byl Coocox projekt v IS příloze funkční, je nutno do něj přidat DSP knihovnu: „CMSIS-SP-00300-r3p1-00rel0“.

Lze jí stáhnout z <<https://silver.arm.com/browse/CMSIS>> nebo zkopírovat z přiloženého DVD. Je nutné ji přejmenovat z „CMSIS-SP-00300-r3p1-00rel0“ na „DSP_lib“ a umístit do adresáře projektu. Tato knihovna byla z přílohy v IS odebrána kvůli její velikosti.