

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

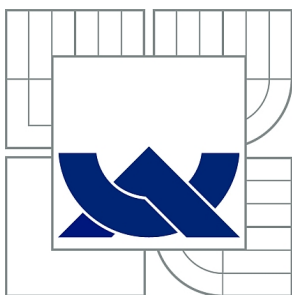
SYSTEM PRO ROZPOZNAVÁNÍ ZVUKU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

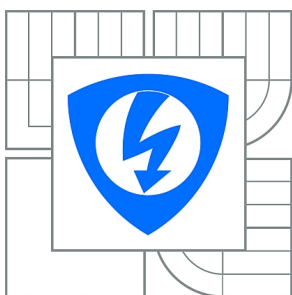
MARTIN MARŠÁL

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTEM PRO ROZPOZNÁVÁNÍ ZVUKU

EMBEDDED SOUND DETECTOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN MARŠÁL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ HYNČICA

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Martin Maršál

ID: 146055

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

System pro rozpoznávání zvuku

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte vestavný systém pro rozpoznání charakteristického zvuku (například tón sirény nebo tlesknutí).

- 1) Seznamte se s používanými typy mikrofonů, jejich vlastnostmi a možnostmi připojení k mikrokontroléru.
- 2) Navrhněte připojení mikrofonu k mikrokontroléru, ověřte vlastnosti a nasbírejte sadu zvuků pro další vyhodnocování.
- 3) Vyberte a vyzkoušejte vhodnou metodu pro rozpoznání zvuku, založenou na rychlé Fourierově transformaci.
- 4) Metodu rozpoznávání zvuku implementujte do mikrokontroléru.
- 5) Vyhodnoťte vlastnosti vytvořeného řešení.

DOPORUČENÁ LITERATURA:

[1] KARL, Holger a Andreas WILLIG. Protocols and architectures for wireless sensor networks: measuring energy consumption and lifetime. Chichester: John Wiley, 2007, xxv, 497 s. ISBN 978-0-470-51923-3.

[2] STALLINGS W.: Wireless Communications & Networks. 2.vyd. Upper Saddle River (NJ): Pearson Prentice Hall, 2005. ISBN 0-13-191835-4.

Termín zadání: 10.2.2014

Termín odevzdání: 26.5.2014

Vedoucí práce: Ing. Ondřej Hynčica

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

ABSTRAKT

V této bakalářské práci popisuji postup vytvoření systému rozpoznávání zvuku. Systém má za úkol rozpoznávat dva druhy zvuků (tlesknutí a zvuk požárního hlásiče). Zvuk tlesknutí si uživatel před použitím nejprve naučí na vlastní typ tlesknutí. Zvuk požárního hlásiče je předem definován podle typu alarmu. Vyhodnocování zvuků je prováděno na základě Fourierova rozvoje. Parametry systému jsou nastaveny podle požadavků na specifickost a senzitivitu detekce.

KLÍČOVÁ SLOVA

rozpoznání zvuku, identifikace zvuku, vyhodnocování zvuku, FFT, Goertzelův algoritmus, snímač zvuku

ABSTRACT

In this bachelor thesis I describe how to create a sound recognition system. System aims to detect two types of sounds (hand clap and the sound of the smoke detector). The sound of hand clap user before first using learns to hand clap own type. The sound of the smoke detector is pre-defined by the type of alarm. Evaluation of sounds is done based on Fourier series. The system parameters are set according to the requirements of specificity and sensitivity.

KEYWORDS

recognizing the sound, the sound identification, evaluation audio, FFT, Goertzel algorithm, sound sensor

MARŠÁL, M. *Systém pro rozpoznávání zvuku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 54 s. Vedoucí bakalářské práce Ing. Ondřej Hynčica.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Systém pro rozpoznávání zvuku“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 svému vedoucímu práce Ing. Ondřejovi Hynčicovi za perfektní spolupráci, čas strávený na konzultacích a odborné rady při řešení mé bakalářské práce.

Brno

.....
(podpis autora)

OBSAH

Úvod.....	8
1 Rozvržení práce.....	9
1.1 Podobné práce.....	9
2 Vyvojová platforma FRDM-KL25Z [1].....	10
2.1 Vývojářská komunita - mbed.....	10
2.2 Vlastnosti.....	10
2.3 Technické parametry.....	11
2.4 Blokové schéma.....	11
3 Snímač zvuku.....	12
3.1 Schéma zapojení.....	12
3.2 Mikrofony.....	12
3.2.1 Kondenzátorový mikrofon.....	12
3.2.2 Elektretový mikrofon.....	13
3.2.3 Dynamický mikrofon.....	13
3.2.4 Uhlíkový mikrofon.....	13
3.2.5 Piezoelektrický mikrofon.....	13
3.2.6 MEMS mikrofon.....	14
3.2.7 Shrnutí.....	14
3.3 Offset.....	15
3.4 Filtr.....	15
3.5 Operační zesilovač.....	16
3.6 Externí trigger.....	17
4 Fourierova transformace.....	18
4.1 DFT.....	18
4.1.1 Shrnutí.....	19
4.2 FFT.....	19
4.2.1 Algoritmus radix-2.....	19
4.3 Goertzelův algoritmus [13].....	22
4.3.1 Realizace.....	22
4.4 Srovnání Goertzelova algoritmu s FFT radix-2.....	24
4.5 Shrnutí.....	25
5 Postup řešení.....	26
5.1 Měření v MATLABu.....	26

5.2	Snímač zvuku	28
5.3	Software	30
5.3.1	Vzorkování vstupního signálu.....	30
5.3.2	Digitální FIR filtr	31
5.3.3	Vyhodnocení příchozích zvuků.....	32
5.4	Shrnutí.....	38
6	Testování systému	39
7	Závěr.....	41
	Literatura.....	43
	Seznam tabulek	45
	Seznam obrázků	45
	Seznam příloh	46
A	Měření v MATLABu.....	47
A.1	SIMULINK načítání mikrofону.....	47
A.2	Frekvenční spektrum tlesknutí	47
B	Program systému pro rozpoznávání zvuku.....	48
C	Návrh DPS snímače zvuku.....	52
C.1	Osazovací plán DPS snímače zvuku	52
C.2	Spodní vrstva DPS snímače zvuku	52
C.3	Schéma snímače zvuku	53
D	Přiložené CD	54

ÚVOD

V dnešní době se s moderními technologiemi setkáváme ve všech odvětvích našeho žití a setkáváme se s nimi opravdu každý den. Ani automatizovaná budova dnes není ojedinělého. Automatizovaná nebo také inteligentní budova by nám měla sloužit k tomu, aby nám naše bydlení ulehčovala, zjednodušovala a zároveň nás také chránila a přinášela vyšší komfort žití.

Ovládání obytných budov za pomoci dálkových ovladačů, počítačů a mobilních telefonů je praktické, ale ne vždy pro všechny periferie. Tímto narážím například na ovládání osvětlení v domě. Už ve starých filmech ovládali lidé světla za pomoci tlesknutí. Proč v dnešní moderní době je tento způsob řízení tak málo prakticky využíván? Jistěže v praxi se nacházejí takové systémy. Například snímače rozbití skla, ty ale nejsou využívány na frekventované ovládání.

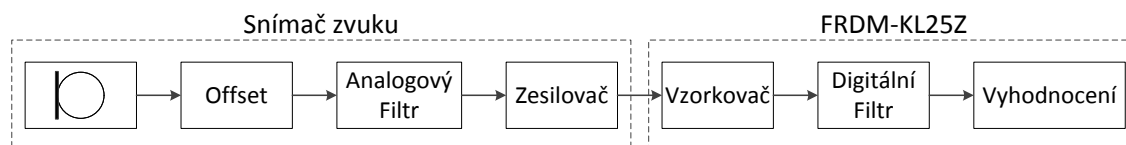
Z tohoto důvodu jsem se rozhodl vytvořit systém pro rozpoznávání zvuku a následné ovládání zařízení zvukem, v mém případě tlesknutím. Systém by měl rozpoznávat i další uživatelem zvolené zvuky, například požární hlásič i zmiňované rozbití skla. Tímto rozšířením se systém stává univerzálním nástrojem pro rozpoznávání zvukových podnětů v místnosti či budově. Také zastoupí funkci bezdrátové komunikace zařízení, které je pro větší počet zařízení velice nákladná.

1 ROZVRŽENÍ PRÁCE

Cílem mé práce bylo vytvoření vestavného systému na rozpoznání charakteristického zvuku. Jako charakteristický zvuk jsem si vybral tlesknutí a zvuk požárního hlásiče. Zvuk požárního hlásiče je zástupcem již zmiňované bezdrátové komunikace mezi zvukovými hlásiči a systémem.

Nejprve bylo zapotřebí vybrat nějakou metodu analyzování zvuků. Na výběr bylo více variant řešení, které jsou uvedeny následující kapitole. Pro rozpoznání zvuků jsem zvolil analyzování pomocí Fourierova rozvoje, který by měl mít větší pravděpodobnost správných výsledků. Metoda je výpočetně i časově náročná. Tím se zvýšily nároky na výpočetní výkon vyhodnocovacího procesoru. Proto jsem zvolil nový mikroprocesor KL25Z od firmy Freescale s 32-bitovou architekturou AMR.

Dalším krokem bylo vybrat snímač, který by se dal nastavovat podle potřeb systému. Na trhu se vyskytují vhodné snímače, z důvodu zdlouhavých dodacích lhůt výrobců jsem se však nakonec rozhodl k sestavení vlastního snímače s nastavitelným zesílením a nastavitelnou šířkou snímaného pásma. Na Obr. 1 je znázorněno blokové schéma systému na rozpoznání zvuku.



Obr. 1: Blokové schéma systému na rozpoznání zvuku

1.1 Podobné práce

Při hledání podobných prací jsem se setkal se systémy různého rozpoznávání.

- Rozpoznávání zvuků hmyzu v práci „Monitoring and Mining Insect Sounds in Visual Space“
<http://www.cs.ucr.edu/~yhao/AnimalSound/FullPaper.pdf>
- Systémy rozbití skla, které využívají Fourierovy transformace pro zjištění frekvenčního pásma.
http://cache.freescale.com/files/sensors/doc/app_note/AN4464.pdf
- Studentský projekt ETH Zürich, který řeší vyhodnocení tlesknutí pouze pomocí amplitudy časového průběhu.
<ftp://ftp.tik.ee.ethz.ch/pub/students/2013-FS/GA-2013-03.pdf>

2 VYVOJOVÁ PLATFORMA FRDM-KL25Z

[1]

FRDM-KL25Z je nová, velice levná vývojová platforma. Byla navržena firmou Freescale ve spolupráci s mbed. Microprocessor je vhodný pro jednoduché, ale i složitější aplikace. Je postavený na 32 bit ARM® Cortex™-M0+ mikrokontroléru. Mezi jeho vlastnosti patří dobrý výpočetní výkon, snadné ovládání I/O, nízká spotřeba, standardní konstrukce s možností rozšíření a vestavěný debug interface pro FLASH programování a run-control. Má stejné rozložení vnitřních pinů jako Arduino R3.

2.1 Vývojářská komunita - mbed

FRDM-KL25Z je plně podporován mbed platformou, takže budete mít přístup k bezplatným nástrojům. Její součástí je také C / C++ SDK pro programování periférií na vysoké úrovni. Mbed poskytuje sdílení knihoven, programů a příkladů v mbed komunitě. Nalezneme zde i online kompilátor, kde bezplatně vytváříte programy. Programy se ukládají na síťový disk, takže se k programům dostanete odkudkoliv. K tomu stačí se jen bezplatně zaregistrovat a vybrat typ desky.

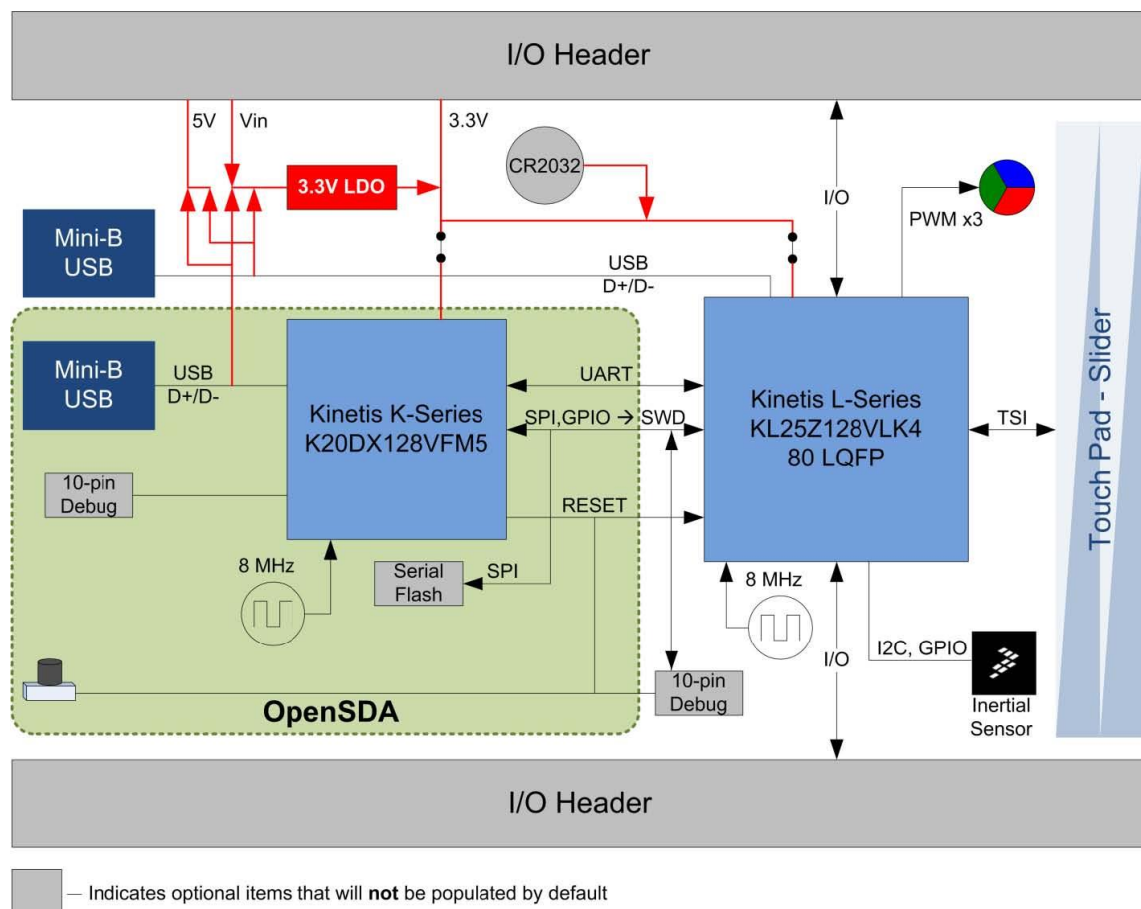
2.2 Vlastnosti

- Freescale KL25Z Kinetis KL2 MCU
 - 32 bit ARM® Cortex™-M0+ Core
 - 48 MHz, 16 KB RAM, 128 KB FLASH
 - 2xSPI, 2xI2C, 3xUART, 6xPWM, 6xADC, dotykový senzor, GPIO
- Vnější periférie
 - MMA8451Q - 3-osý akcelerometr
 - RGB LED
 - Kapacitní dotykový senzor
- Konstrukční parametry
 - Rozměry: 81 mm x 53 mm
 - Napájení: 5V USB nebo 4,5-9 V
 - FLASH programátor
- Vývojářské webové stránky: mbed
 - Online Compiler
 - Vysoká úroveň C/C++ SDK
 - Kuchařku s radami a příklady ostatních uživatelů

2.3 Technické parametry

- Napájení z USB nebo připojení 5 V – 9 V na externí napájení desky (VIN)
- Digitální IO 3,3 V a na jeden pin 4mA, všechny piny maximálně 400 mA
- Externí napájení desky na pinu Vin (5-9 V, 100 mA)
- RESET na pinu – stejné vlastnosti jako RESET tlačítko

2.4 Blokové schéma

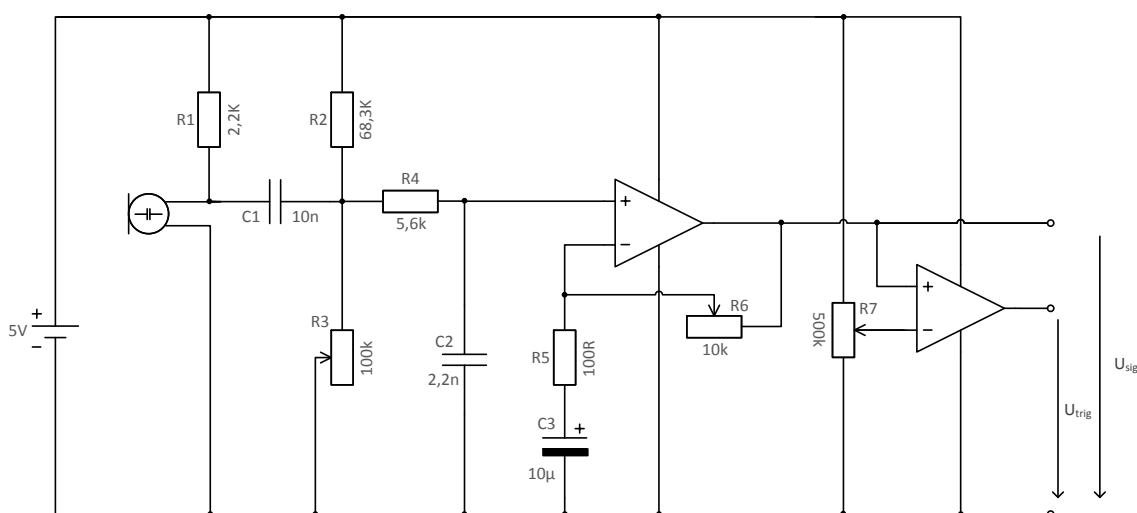


Obr. 2: Blokové schéma Freedom desky s mikroprocesorem KL25Z [2]

3 SNÍMAČ ZVUKU

Snímač zvuku je první součástí mého systému. Jeho úkolem je snímat okolní zvuky v požadovaném frekvenčním pásmu s určitým zesílením. Skládá se z mikrofону, offsetu, analogového aktivního filtru, operačního zesilovače a externího triggru, viz Obr. 1. Pro napájení snímače využívám usměrněný stejnosměrný zdroj napětí 5 V z mikroprocesoru. DPS snímače zvuku je navržena jako nastavný modul platformy FRDM-KL25Z.

3.1 Schéma zapojení



Obr. 3: Schéma zapojení snímače

3.2 Mikrofony

Pro snímač jsem mohl využít velké množství mikrofónů, které se liší nejen konstrukcí, ale především vlastnostmi přenášeného akustického tlaku na elektrický signál, který je dále zpracováván. Proto jsem před konstrukcí snímače nejprve provedl rešerši více druhů mikrofónů, abych našel mikrofón s vyhovujícími vlastnostmi pro můj systém.

3.2.1 Kondenzátorový mikrofón

Kondenzátorový mikrofón je založen na principu změny kapacity. Skládá se z pevné elektrody a velice tenké a lehké vodivé pohyblivé membrány, která tvoří druhou elektrodu kondenzátoru.

Dopadající zvukové vlny pohybují s membránou a tím mění kapacitu kondenzátoru. Změna kapacity moduluje procházející proud. Jedná se tedy o pasivní mikrofón.

Výroba kondenzátorového mikrofónu je velice nákladná přesto má vynikající frekvenční charakteristiku a další vlastnosti, proto se využívá pro studiové nahrávání.

[15]

3.2.2 Elektretový mikrofon

Elektretový mikrofon je typem kondenzátorového mikrofonu. Je rovněž založen na principu změny kapacity. Skládá se z velmi lehké, pohyblivé elektretové membrány a pevné elektrody. Elektret je materiál permanentně elektricky nabitý. Elektrody jsou od sebe rozmístěny velmi blízko a tvoří tak kondenzátor.

Dopadající zvukové vlny pohybují s elektretovou membránou. Pohybem způsobuje změnu kapacity, kterou vyhodnotí vestavěný zesilovač tvořen JFET tranzistorem, ten mění výstupní napětí. Jedná se tedy o pasivní mikrofon.

Elektretový mikrofon má téměř rovnou frekvenční charakteristiku, vysokou citlivost, malé zkreslení, je dostupný a má nízkou cenu. [3], [15]

3.2.3 Dynamický mikrofon

Dynamický mikrofon využívá princip elektromagnetické indukce. Je založen na pohybu cívky v magnetickém poli. Membrána je upevněna na pohyblivé cívce, která se nachází ve statickém poli permanentního magnetu.

Dopadající zvukové vlny rozkmitají membránu a spolu s ní, se začne pohybovat cívka v magnetickém poli a začne vytvářet střídavé napětí, přímo úměrné dopadajícímu zvuku. Jedná se tedy o aktivní mikrofon.

Dynamický mikrofon nepotřebuje napájení, je odolný proti vysokým hladinám zvuku, mechanicky odolný, nevýhody jsou malá citlivost a velké rozměry. [15], [16]

3.2.4 Uhlíkový mikrofon

Uhlíkový mikrofon je založen na principu změny elektrického odporu. Skládá se ze dvou kovových desek a velkého počtu uhlíkových zrněk. Jedna z desek je velmi tenká a tvoří membránu. Mezi membránou a pevnou deskou jsou těsně umístěna uhlíková zrnka, která jsou v trvalém kontaktu s deskami.

Zvukové vlny dopadající na membránu mění tlak zrněk, který je přímo úměrný změně elektrického odporu a mění tak stále procházející stejnosměrný proud. Jedná se tedy o pasivní mikrofon.

V dnešní době, se tyto mikrofony používají zřídka. Dříve se používaly v komunikační technice. Hlavní výhodou je, schopnost pracovat i na velmi malém napětí oproti ostatním mikrofonům, nevýhodou vysoký šum a zkreslení. [15]

3.2.5 Piezoelektrický mikrofon

Piezoelektrický mikrofon je založen na piezoelektrickém jevu vytvoření elektrického náboje na stranách krystalu při mechanickém namáhání. Skládá se z pouzdra a membrány, která je spojena s tenkým proužek piezoelektrického materiálu.

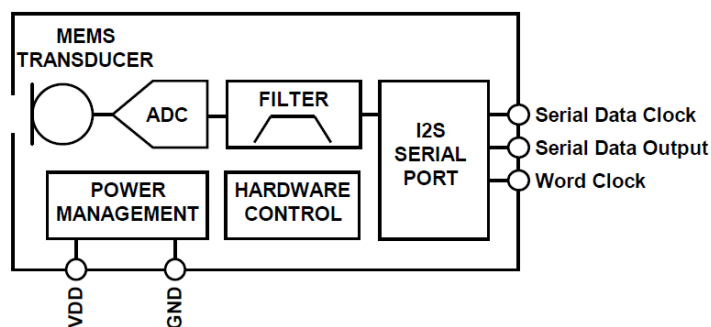
Působením zvukových vln na membránu vytváří mechanický tlak na piezokrystal, který vytváří úměrný elektrický náboj. Jedná se tedy o aktivní mikrofon.

Piezoelektrický mikrofon je citlivý, nevýhodou je špatná frekvenční odezva. [16]

3.2.6 MEMS mikrofon

MEMS mikrofon je vlastně kapacitní čidlo. Je založen na stejném principu jako kapacitní mikrofon zmíněný výše akorát v miniaturní podobě. Hojně se využívají u moderních telefonů. Vyrábějí se s analogovým nebo digitálním výstupem. U analogové varianty mikrofonu je za mikrofonem zařazen zesilovač, vstupem je pouze napájení a výstupem je analogový signál. Digitálních variant je více. Rozdíly jsou ve vnitřní struktuře i typech výstupů. Všechny digitální MEMS mikrofony obsahují ADC pro převod analogového signálu na digitální kvůli dalšímu zpracování signálu. Ve vnitřní struktuře se například nachází zesilovač nebo digitální filtr. Výstupy jsou stereo PDM modulovaný signál nebo I²S sériová komunikace.

MEMS mikrofony mají vynikající frekvenční charakteristiky, malý příkon, miniaturní rozměry a dobrou směrovou charakteristiku. [17]



Obr. 4: Blokové schéma I²S MEMS mikrofonu [17]

3.2.7 Shrnutí

Z rešerše mikrofonů vychází nejlépe elektretový mikrofon. Především díky své jednoduchosti, dostupnosti a dynamickým vlastnostem.

Prakticky jsem otestoval dva dostupné ze zmíněných mikrofonů, elektretový a dynamický. Z výsledků bylo zřejmé, že dynamický mikrofon není úplně vhodný, pro velkou dynamiku membrány je mikrofon velice necitlivý. Tento typ mikrofonu se většinou využívá v hudební technice, na zpěv z blízké vzdálenosti od mikrofonu. Naopak u elektretového mikrofonu je dynamika podstatně menší a citlivost vysoká a to díky volné a lehké elektretové membráně, která zachytí sebemenší zvlnění vzduchu, což je při tlesnutí výhodou. Vlastnosti elektretového mikrofonu jsou dostačující pro návrh snímače, a proto jsem si ho vybral pro konstrukci snímače zvuku.

Po vybrání mikrofonu jsem dále postupoval podle pravidel a požadavků na návrh snímače zvuku. Testováním jsem odstraňoval nedostatky a přidával vylepšení například

antialiasingový filtr nebo externí trigger, již výše detailně rozebrané. Níže zmíněné výpočty jsou shodné s reálným nastavením snímače.

3.3 Offset

Offset bylo nutné přidat kvůli nesymetrickému napájení operačního zesilovače. Je tvořen paralelní kombinací odporů R_2 a R_3 , které na výstupu posouvají signál z mikrofону. Bez těchto rezistorů by mikroprocesor, který podporuje pouze kladný analogový vstupní signál, zaznamenával pouze kladnou půlvlnu. Odpor R_3 je tvořen proměnným trimrem pro případné jemné donastavení výchozí pozice signálu.

Snímač zvuku je napájen stejnosměrným napětím 5 V . Analogové vstupy mikroprocesoru mají vstupní napětí pouze do $3,3\text{ V}$. Z tohoto důvodu není offset nastaven jako polovina napájecího napětí (5 V), ale na polovinu napětí mikroprocesoru ($3,3\text{ V}$) plus mezní napětí OZ. Mezní napětí je hodnota OZ, od které je možno propouštět signál. V mém případě jsem naměřil hodnotu $0,92\text{ V}$. Tato hodnota se projeví snížením měřeného intervalu mikroprocesoru na $2,4\text{ V}$. Při správném nastavení zesílení nemá toto snížení vliv na správnost snímání.

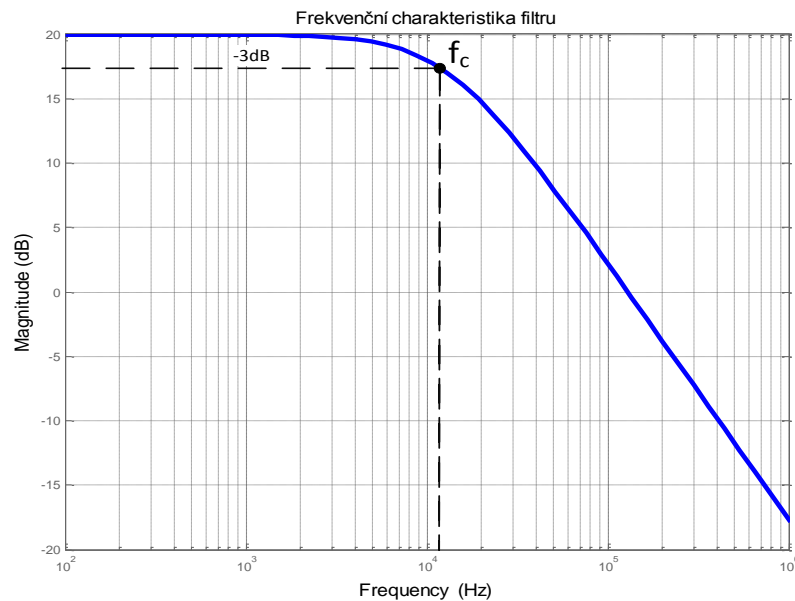
3.4 Filtr

Filtr představuje aktivní dolní propust, která má funkci antialiasingového filtru. Filtr je konstruován pomocí rezistorů, kondenzátorů a operačního zesilovače. Horní mezní kmitočet je nastaven tak, aby odpovídal hodnotě menší než je polovina vzorkovací frekvence A/D převodníku a tím zabránil aliasing efektu.

Horní mezní kmitočet určuje maximální hodnotu propustné frekvence výstupního signálu z mikrofону. Vypočet horního mezního kmitočtu jsem provedl dle následujícího vztahu (3.1).

$$f_h = \frac{1}{2\pi \cdot R_4 \cdot C_2} = \frac{1}{2\pi \cdot 5,6 \cdot 10^3 \cdot 2,2 \cdot 10^{-9}} = 12,918\text{ kHz} \quad (3.1)$$

Výsledná šířka pásma je interval mezi těmito mezními kmitočty. Ostatní kmitočty jsou tlumeny, znázorněno na Obr. 5.



Obr. 5: Frekvenční charakteristika dolní aktivní propustí

Kondenzátor C_1 s rezistory R_1 a R_2 odděluje výstupní signál z mikrofonu od stejnosměrné složky napětí, pomocí kondenzátoru C_1 . Společně s paralelní kombinací odporů na offsetování signálu tvoří dolní propust. Výpočet vstupního dolního mezního kmitočtu je proveden dle vztahu (3.2).

$$f_{vd} = \frac{1}{2\pi \cdot \left(\frac{R_2 \cdot R_3}{R_2 + R_3}\right) \cdot C_1} = \frac{1}{2\pi \cdot 28137 \cdot 10 \cdot 10^{-9}} = 566 \text{ Hz} \quad (3.2)$$

Požadovaná šířka pásma je závislá na zvuku, který chceme vyhodnocovat. V mém případě jsou to frekvence 2 – 5 kHz.

Vzhledem k požadovanému frekvenčnímu rozsahu je frekvence řezu zbytečně předdimenzovaná. Jak jsem zmínil výše, jedná se o antialiasingový filtr, kde se frekvence řezu volím jako polovina vzorkovací frekvence. Přesné a strmé oříznutí frekvencí se provádí až v mikroprocesoru digitálním filtrem, uvedeno níže. Při návrhu bylo počítáno i s poklesem o -3 dB na dekádu od mezní frekvence. Pokles téměř neovlivňuje používané frekvence.

3.5 Operační zesilovač

Operační zesilovač je součástí aktivního filtru, který zahrnuje funkci nastavitelného zesílení a slouží k zesílení vyfiltrovaného vstupního signálu. Zesílení měníme změnou poměru rezistorů, zapojených na výstupu operačního zesilovače. Jejich poměr měníme

potenciometrem R_6 , přičemž rezistor R_5 je konstantní. Výpočet zesílení vyplývá ze vztahu (3.3).

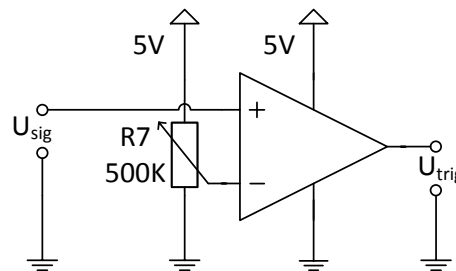
$$K = 1 + \frac{R_6}{R_5} = 1 + \frac{1600}{100} = 16 \quad (3.3)$$

Pro snímač jsem si vybral operační zesilovač LM358, který nepotřebuje záporné napájení. Pro moji aplikaci má zesilovač dostatečně velké frekvenční pásmo. Dosahuje vysokých zesílení a podporuje nízké kladné napájení. [4]

3.6 Externí trigger

Externí trigger má za úkol poslat signál do mikroprocesoru při překročení nastavené úrovně výstupního signálu. Jde tedy o nastavení citlivosti na příchozí zvuky. V obou případech vyhodnocení (tlesknutí a alarm) jde o hlasité zvuky. Z tohoto důvodu je trigger nastaven na nižší citlivost příchozích zvuků.

Externí trigger je tvořen druhým operačním zesilovačem, který je součástí integrovaného obvodu. Trigger je konstruován jako komparátor napětěových úrovní. Porovnává výstupní signál z aktivního filtru s referenční hodnotou nastavovanou potenciometrem R_7 , viz Obr. 6.



Obr. 6: Samostatné schéma externího triggeru

4 FOURIEROVA TRANSFORMACE

Fourierova transformace nebo také frekvenční spektrum je matematická transformace používaná k transformaci signálu mezi časovou a frekvenční oblastí. Rozkládá žádanou funkci na součet základních sinusových funkcí. Fourierova transformace (4.1) má rovněž inverzní Fourierovu transformaci danou vztahem (4.2). [5], [6]

$$F\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-2\pi ift} dt \quad (4.1)$$

$$F^{-1}\{G(f)\} = \int_{-\infty}^{\infty} G(f)e^{2\pi ift} df = g(t) \quad (4.2)$$

Aby Fourierova transformace existovala, musí funkce $g(t)$ splňovat následující podmínky.

- Funkce $g(t)$ musí být absolutně integrovatelná v intervalu $t \in (-\infty, \infty)$.
- Funkce $g(t)$ musí mít na každém konečném intervalu konečný počet nespojitostí a konečný počet maxim a minim. [6]

Numerické řešení pomocí diskrétní Fourierovy transformace (DFT) je typicky implementována jako rychlá Fourier transformace (FFT), která je výpočetně efektivnější. [7]

4.1 DFT

Modifikovaná forma Fourierovy transformace známá jako diskrétní Fourierova transformace slouží pro výpočet frekvenčního spektra pro signál s konečným počtem vzorků. Přímá diskrétní Fourierova transformace má tvar (4.3).

$$X(m) = \sum_{k=0}^{N-1} x(k)e^{-jmn\frac{2\pi}{N}} \quad m = 0, 1, \dots, N-1 \quad (4.3)$$

Existuje také inverzní diskrétní Fourierova transformace (4.4).

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)e^{+jmn\frac{2\pi}{N}} \quad m = 0, 1, \dots, N-1 \quad (4.4)$$

kde N je počet vzorků, index n označuje časový okamžik a m pořadí diskrétní frekvence. [7]

4.1.1 Shrnutí

Z důvodu vysoké výpočetní náročnosti DFT při použití mikroprocesoru jsem přikročil ke zjednodušeným výpočtům výpočtu frekvenčních spekter. Pro výpočet frekvenčního spektra tlesknutí jsem využil rychlou Fourierovu transformaci (FFT) a pro výpočet frekvenčního spektra zvuku požárního hlásiče Goertzelův algoritmus.

Oběma těmito metodám se věnuji v následujících kapitolách, kde je detailně popisují.

4.2 FFT

Časová náročnost výpočtu DFT vedla k zefektivnění algoritmů, které jsou dnes známy jako rychlá Fourierova transformace (FFT). Ta předpokládá, že standardní DFT zahrnuje nadbytečný počet zbytečných operací.

Postup pomocí DFT udává tedy velice přesné informace o frekvenčních složkách, ale s obrovským výpočetním úsilím. DFT se používala do roku 1960, kdy Cooley a Tukey objevili číselný algoritmus, který umožňuje vyhodnocení s výrazným snížením výpočetního úsilí. V současné době je FFT úspěšně využívána v různých oblastech kvůli její rychlosti, přesnosti a účinnosti. [8]

4.2.1 Algoritmus randix-2

Algoritmus randix-2 je nejpoužívanější FFT algoritmus. Využívá se zde vlastností DFT symetričnosti a komplexní sdruženosti kolem svého středu. Platí to pouze pro sudý počet reálného vstupního signálu. Algoritmus rozděluje počet prvků N na dvě části o velikosti $\frac{N}{2}$ a to na sudou reálnou část a imaginární lichou část. Lichá část je tedy identická s reálnou pouze s opačným znaménkem. Někdy nese označení „motýl“. [12]

Pro N vzorků může být DFT vyjádřena vztahem (4.5).

$$X(m) = \sum_{m=0}^{\frac{N}{2}-1} x(2m)W_N^{2km} + \sum_{m=0}^{\frac{N}{2}-1} x(2m+1)W_N^{k(2m+1)} \quad (4.5)$$
$$m = 0, 1, \dots, N-1, \quad W_N = e^{-j\frac{2\pi}{N}}$$

kde W_N je fázový faktor (twiddle factor). Popisuje rotující vektor, který se otáčí v krocích podle počtu vzorků N . [9]

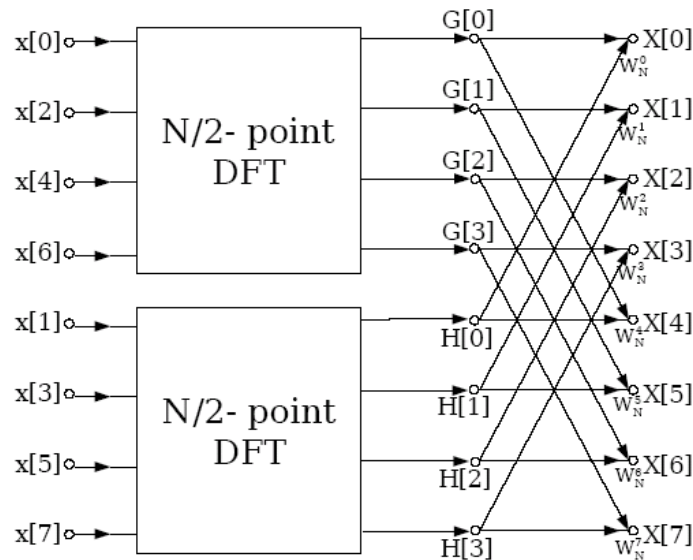
Dosazením $W_N^{\frac{km}{2}}$ za W_N^{2km} , získáme upravený vztah (4.6).

$$X(k) = G(k) + W_N^k H(k) \quad k = 0, 1, \dots, N-1 \quad (4.6)$$
$$W_N = e^{-j\frac{2\pi}{N}}$$

Matematicky zjednodušená rovnice (4.6) ukazuje, že všechny frekvenční výstupy $X(k)$ lze vypočítat jako součet $\frac{N}{2}$ vzorků indexovaných na sudé $G(k)$ a liché $H(k)$. [10]

4.2.1.1 DIT radix-2

Této metodě se říká dělení v čase (DIT - decimation-in-time), protože časové vzorky jsou střídavě rozděleny do skupin (polovin). Obr. 7 znázorňuje rozdělení vstupních dat na dvě poloviny $G(k)$ a $H(k)$. Protože jsou periodické s periodou $\frac{N}{2}$, mohou být využity pro výpočet výstupních frekvencí o délce N , a to pro $X(k)$ a $X(k + \frac{N}{2})$ s jiným fázovým faktorem. Pomocí této metody je zapotřebí $\frac{N^2}{2}$ komplexních součinů a $(\frac{N^2}{2} + N)$ komplexních součtů.



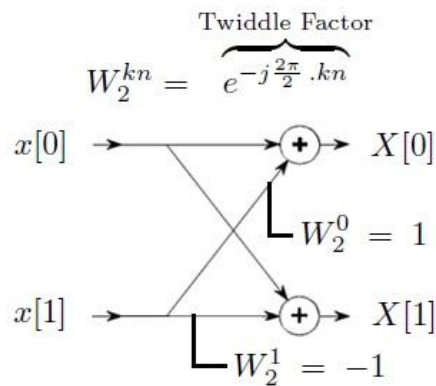
Obr. 7: Dělení v čase o délce N do dvou $N/2$ částí [10]

Ve srovnání s přímým výpočtem pomocí DFT pro všechny frekvence by bylo nutné N^2 komplexních součinů a $(N^2 - N)$ komplexních součtů. Výsledkem použití této metody je zkrácení výpočtu téměř na polovinu. [10]

4.2.1.2 Základní operace motýlku

Motýlek slouží k zjednodušení algoritmu. Základní operace motýlku vyžadují pouze $\frac{N}{2}$ násobení fázových faktorů v etapě, zobrazeno na Obr. 8. Po sloučení fázových faktorů do jedné větve vznikne jeden výstup s dvoubodovým signálem, jak popisuje rovnice (4.7). [11]

$$\begin{aligned} X[0] &= x[0] + x[1] \\ X[1] &= x[0] - x[1] \end{aligned} \quad (4.7)$$



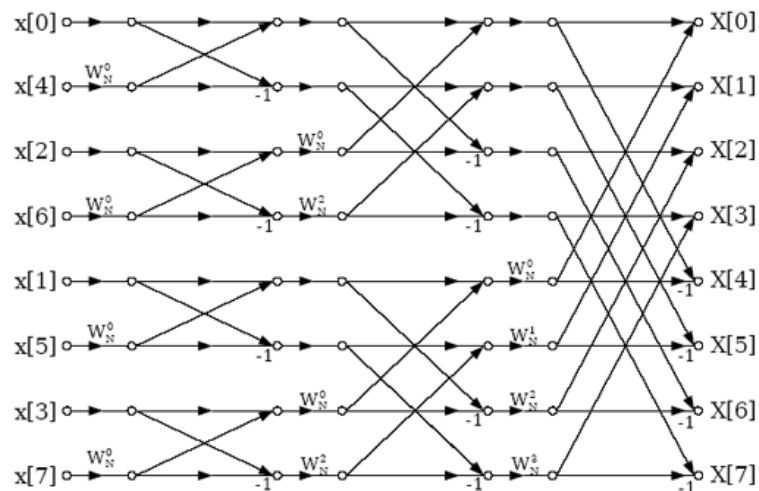
Obr. 8: Základní motýlek [11]

4.2.1.3 DIT radix-2 rekurzivní

Tato část popisuje další zjednodušení algoritmu. Algoritmus je rozdělen do etap, ve kterých se provádějí základní operace motýlků, znázorněno na Obr. 9. Počet etap M je určen vztahem (4.8). Z něj vyplývá, že vstupní počet vzorků je dán vztahem (4.9) v němž si všimněme základu čísla 2, proto se algoritmus nazývá radix-2.

$$M = \log_2(N) \tag{4.8}$$

$$N = 2^M \tag{4.9}$$



Obr. 9: Radix-2 DIT FFT algoritmus [10]

Tímto algoritmem docílíme pozoruhodných úspor. Například počet vzorků $N = 1024$ DFT by vyžadoval 1048576 komplexních součinů a 1047552 komplexních součtů přes přímý výpočet DFT. Pomocí DIT radix-2 rekurzivní FFT je třeba pouze 20480 komplexních součinů a 10240 komplexních součtů. Úspory se zvyšují s větším počtem vzorků N vstupujících do FFT. [10]

4.2.1.4 Bitová inverze

Bitová inverze je dalším zvýšením efektivity algoritmu. Jedná se o vhodné přeskupení indexů vstupních hodnot $x(k)$, viz Obr. 9. Při výpočtech pomocí MCU s bitovou inverzí je možnost výpočtu v jednom poli. To značně spoří paměť RAM. Bitová inverze je zobrazena v Tab. 1. [10], [11], [12]

Tab. 1: Bitová inverze pro N=8 [12]

Prvotní index		Výsledný index	
Dec	Bin	Bin	Dec
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

4.3 Goertzelův algoritmus [13]

Goertzelův algoritmus je jednou z dalších diskretních metod na převedení signálu z časové oblasti do frekvenční. Za určitých podmínek je tato metoda výpočetně rychlejší než rychlá Fourierova transformace. Algoritmus je rekurzivní filtr, který se zaměřuje na předem stanovenou frekvenci ve spektru. Nejčastěji se využívá na detekci tónů a pro analýzu spektra radiových frekvencí.

4.3.1 Realizace

„Goertzelův algoritmus je realizován jako IIR filtr druhého řádu.“ [13] Zásadním pojmem algoritmu je nahradit obecnou funkci filtru za identifikaci zadané frekvenční spektrální složky signálu. Tento filtr počítá jednu spektrální komplexní hodnotu DFT definovanou vztahem (4.10).

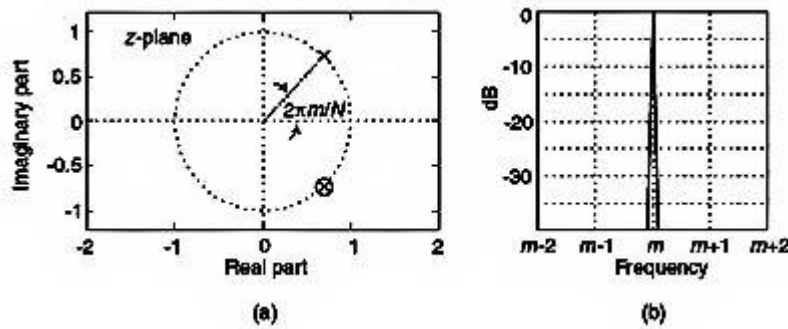
$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-jmk\frac{2\pi}{N}} \quad (4.10)$$

„Výstup filtru $y(n)$ je roven výstupní frekvenci koeficientu DFT, $X(m)$, když index $n=N, (\dots)$.“ [13] „Pro přesný výsledek DFT, musí index rezonanční frekvence m ležet v

intervalu $0 \leq m \leq N - 1$.“ [13] Převedením do z-roviny pomocí Z-transformace získáváme přenos Goertzelova filtru (4.11)

$$H_G(z) = \frac{Y(z)}{X(z)} = \frac{1 - e^{-j\frac{2\pi m}{N}} z^{-1}}{1 - 2 \cos\left(\frac{2\pi m}{N}\right) z^{-1} + z^{-2}} \quad (4.11)$$

s jednou nulou $z = e^{-j\frac{2\pi m}{N}}$ a sdruženými póly $z = e^{\pm j\frac{2\pi m}{N}}$. Nula a pól se vykrátí a zbyde nám pouze jeden pól na mezi stability, protože se nachází na jednotkové kružnici z-roviny, viz Obr. 10 a). Pro regulaci mez stability většinou znamená nestabilitu systému. U Goertzelova algoritmu to není tak nebezpečné z důvodu velkého počtu vzorků N , který obvykle bývá ve stovkách. Filtr zůstává stabilní pro tyto krátké časové sekvence, protože po uložení dat do interních registrů se na začátku nového bloku data resetují na nulu.



Obr. 10: a) rozložení pólů a nul filtru, b) odezva frekvence ve frekvenčním spektru [13]

Na Obr. 10 b) je znázorněn frekvenční rozsah filtru s frekvenční odezvou v žádané frekvenci $\frac{2\pi m}{N}$, která odpovídá periodické frekvenci $\frac{mF_s}{N}$ [Hz]. Z toho vyplývá, že žádanou rezonanční frekvenci m vypočítáme podle vztahu (4.12).

$$m = \frac{f \cdot N}{f_s} \quad (4.12)$$

Kde f je požadovaná frekvence, N počet vzorků a f_s vzorkovací frekvence.

Z důvodu realizace pomocí IIR filtru má filtr nekonečnou impulsní charakteristiku, a proto má úzkou frekvenční odezvu.

Diferenční rovnice v časové oblasti je vyjádřena vztahy (4.13)(4.14).

$$w(n) = 2 \cos\left(\frac{2\pi m}{N}\right) w(n-1) - w(n-2) + x(n) \quad (4.13)$$

$$y(n) = w(n) - e^{-j\frac{2\pi m}{N}} w(n-1) \quad (4.14)$$

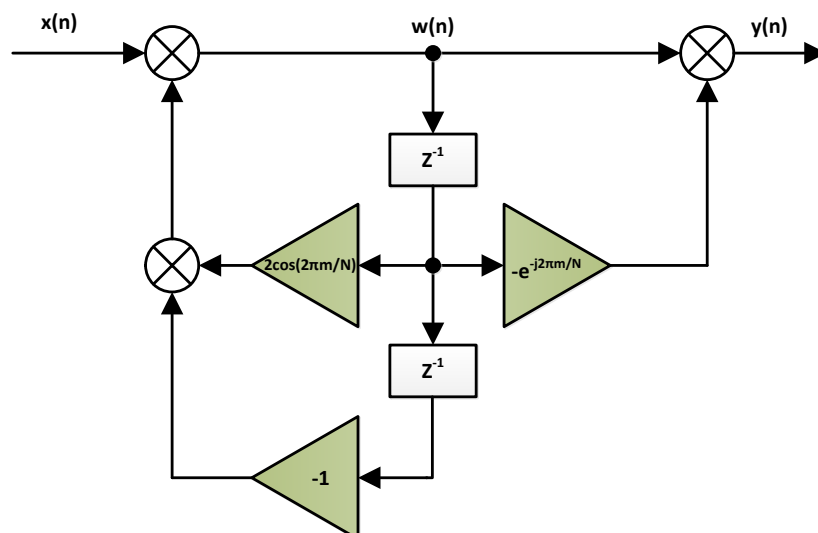
Z rovnic plyne $N + 2$ reálných násobení a $2N + 1$ reálných součtů pro výpočet jedné spektrální hodnoty DFT.

Počet prvků N může být libovolné číslo, čím bude větší, tím bude lepší frekvenční rozlišení a větší odolnost proti rušení. Zvýší se tím ale i počet výpočtů. Rezonanční frekvence m může být jakékoliv číslo v rozsahu 0 a $N - 1$.

Výkonu jedné spektrální hodnoty DFT vypočítáme ze vztahu (4.15).

$$|X(m)|^2 = |X(N - 1)|^2 = w(N - 1)^2 + w(N - 2)^2 - w(N - 1)w(N - 2) \left[2 \cos w\left(\frac{2\pi m}{N}\right) \right] \quad (4.15)$$

Skutečné blokové zapojení Goertzelova algoritmu je zobrazeno na Obr. 11, kde jsou i výpočty jednotlivých koeficientů pro softwarové začlenění.



Obr. 11: Blokové schéma zapojení Goertzelova algoritmu

4.4 Srovnání Goertzelova algoritmu s FFT radix-2

Srovnání Goertzelova algoritmu s FFT radix-2 pro detekci jednoho tónu, vychází nepopíratelně lépe pro Goertzelův algoritmus. Je to dáno tím, že nepočítá celé frekvenční spektrum jako FFT. To znamená, že pro výpočet malého počtu frekvencí je Goertzelův algoritmus efektivnější, viz Tab. 2. Proto se využívá pro detekci tónu a ne celých frekvenčních spekter.

Tab. 2: Matematické srovnání metody FFT radix-2 a Goertzelova algoritmu [13]

Metoda	Reálné násobení	Reálné sčítání
FFT	$2N\log_2(N)$	$N\log_2N$
Goertzelův algoritmus	$N+2$	$2N+1$

Další výhodou je, že pro stejný počet N vzorků, má Goertzelův algoritmus větší rozlišení frekvencí v určitém frekvenčním pásmu než FFT uvedeno na příkladu.

Například pro $N = 1000$ vzorků a $F_s = 10 \text{ kHz}$ má FFT rozlišení frekvencí 10 Hz , dle vztahu (4.16). Jak už jsem zmínil, Goertzelův algoritmus umožňuje vypočítat pouze část frekvenčního spektra, například pro $4000 - 5000 \text{ Hz}$ je rozlišení frekvence 1 Hz , podle výpočtu (4.17) je desetkrát větší než u FFT. Bohužel za cenu většího výpočetního úsilí, které plyne z Tab. 2.

$$x_{FFT} = \frac{f_s}{N} = \frac{10000}{1000} = 10 \text{ Hz} \quad (4.16)$$

$$x_{G_a} = \frac{f_s}{N} = \frac{10000}{1000} = 1 \text{ Hz} \quad (4.17)$$

Pro vyhodnocení tlesknutí je FFT nezbytnou, protože potřebuje vypočítat delší frekvenční spektrum, kde by byl Goertzelův algoritmus neefektivní. Naopak při rozpoznávání zvuku požárního hlásiče je FFT zbytečná, počítat celé frekvenční spektrum od 0 do $F_s/2$, když se požadovaná frekvence nemění. Proto jsem na vyhodnocení zvuku požárního hlásiče využil právě Goertzelův algoritmus podrobně rozebraný výše. [13], [14]

4.5 Shrnutí

Diskrétní Fourierova transformace je určena pro výpočet frekvenčního spektra s konečným počtem vzorků. DFT je výpočetně velice náročná a vzhledem k výkonu většiny mikroprocesorů nepoužitelná, proto jsem pro řešení svého snímače na rozpoznávání tlesknutí zvolil zjednodušený rekurzivní DIT radix-2 FFT algoritmus. Tento algoritmus je oproti DFT natolik zjednodušený a výpočetně méně zatížený, že ho dokážou analyzovat i výkonnostně slabší mikroprocesory. Pracuje na základě půlení počtu vzorků a na násobení fázových faktorů.

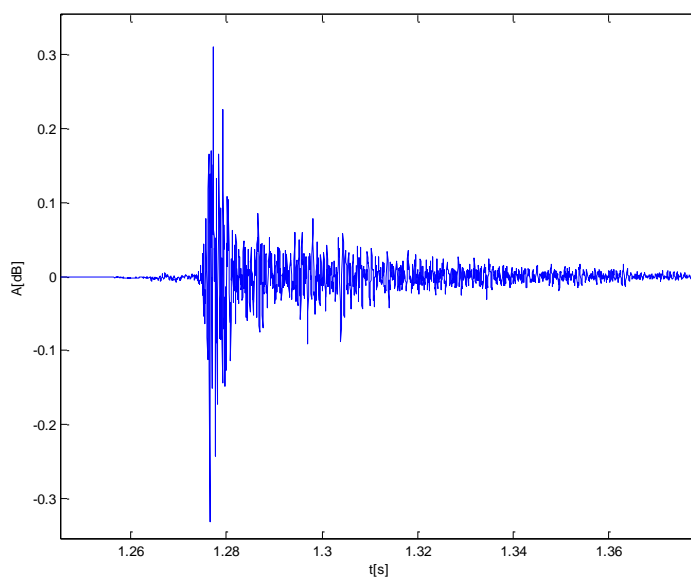
Goertzelův algoritmus se využívá pro detekci jednotlivých tónů, například kódování tlačítek u telefonních linek. Tento algoritmus je efektivní pouze pro výpočet malého počtu frekvencí. Zjednodušeně řečeno je Goertzelův algoritmus pásmová propust diskrétního IIR filtru druhého řádu. Ten propouští pouze požadovanou frekvenci, kterou sleduje a vyhodnotí její zesílení. Z toho plyne, že musíme znát požadovanou frekvenci. Proto se využívá pro definované systémy.

5 POSTUP ŘEŠENÍ

V této kapitole popisuji postup výroby systému pro rozpoznávání zvuku. Jako rozpoznávaný zvuk jsem zvolil tlesknutí a zvuk požárního hlásiče (alarm), o kterém jsem se již zmínil v kapitole Rozvržení práce.

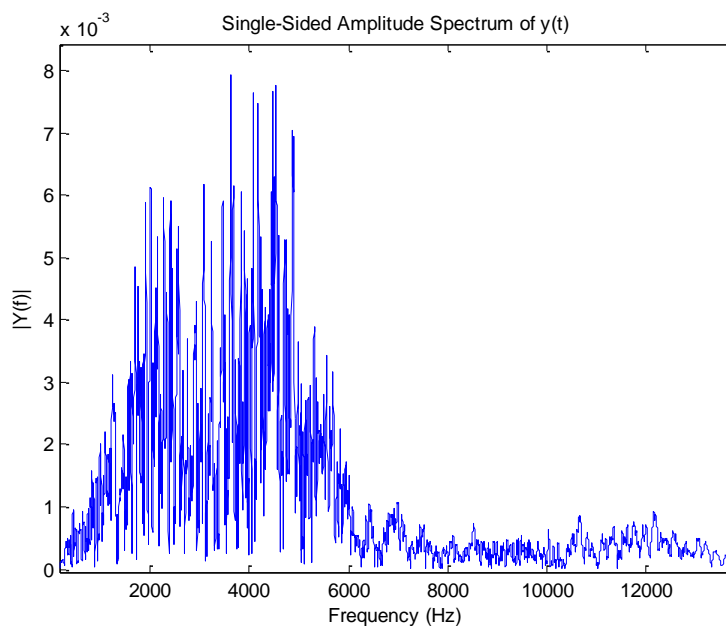
5.1 Měření v MATLABu

Nejprve jsem si v MATLABu naměřil časový průběh tlesknutí. Pro snímání zvuku jsem využil elektretový mikrofon od firmy Creative a externí zvukovou kartu od téhož výrobce. Mikrofon jsem připojil do vstupu LINE-IN externí zvukové karty. Externí zvukovou kartu a její vstup LINE-IN jsem využíval z důvodu ořezávání vyšších frekvencí než hlasových (přibližně 4 kHz) na normálních mikrofonních vstupech integrovaných zvukových karet. Z časového průběhu Obr. 12 jsem zjistil, že průběh trvá poměrně dlouhou dobu (přibližně 100 ms), kterou způsobuje setrvačnost membrány v mikrofonu.



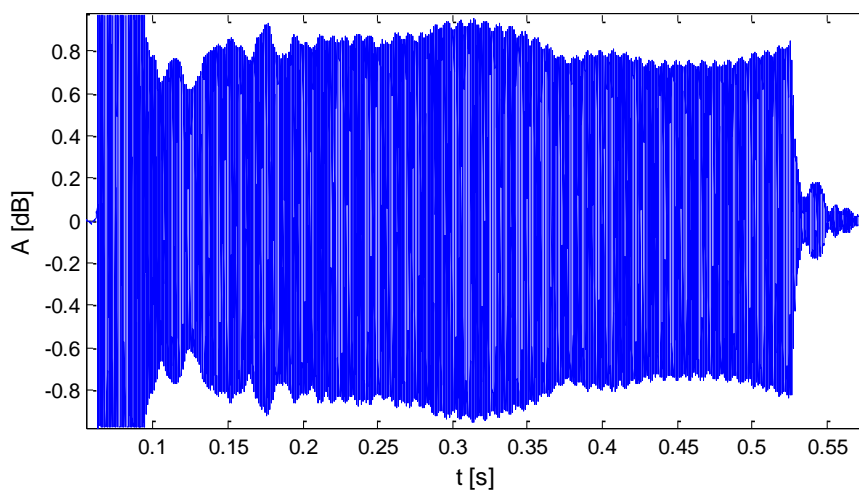
Obr. 12: Naměřený průběh tlesknutí

Tyto průběhy bylo potřeba analyzovat. Jak jsem zmínil v kapitole Rozvržení práce, vybral jsem si Fourierovu spektrální analýzu a opět ji provedl v MATLABu. Pro výpočet frekvenčního spektra tlesknutí jsem využil příslušné funkce. Výsledné spektrum obsahovalo velké množství různých frekvencí, zobrazeno na Obr. 13. Z průběhu je zřejmé, že analyzování a následné zpracování průběhu tlesknutí pomocí mikroprocesoru bude složité.



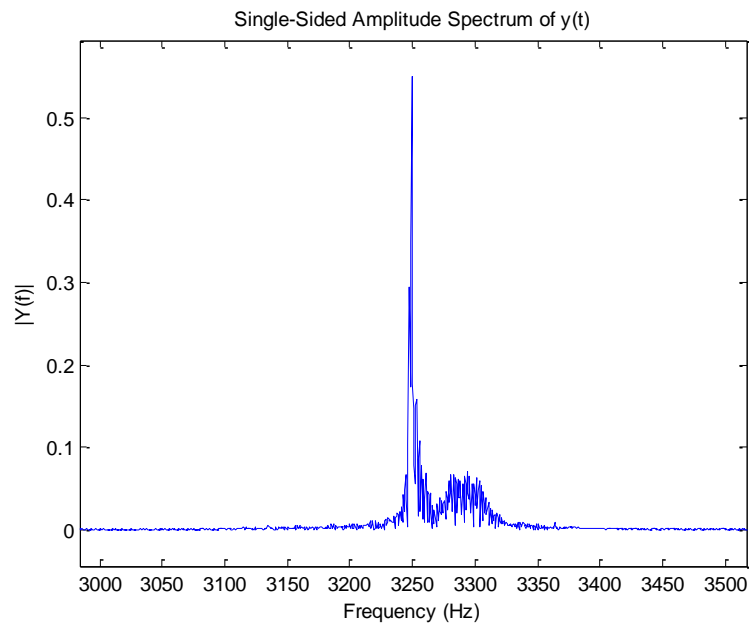
Obr. 13: Vypočítané frekvenční spektrum tlesknutí

Dále jsem provedl měření za pomoci MATLABu i pro zvuk požárního hlásiče. Tento zvuk je už od poslechu více harmonický než zmiňované tlesknutí. Pro snímání zvuku hlásiče jsem použil stejnou měřicí aparaturu jako u tlesknutí. Výsledný naměřený signál je zobrazen na Obr. 14. Časový průběh zobrazený na Obr. 14 se periodicky opakuje, proto není časově omezen jako u tlesknutí.



Obr. 14: Naměřený časový průběh zvuku požárního hlásiče

Z frekvenčního spektra zvukového signálu požárního hlásiče Obr. 15 je vidět, že skutečně obsahuje jednu dominantní harmonickou frekvenci, přibližně 3250 Hz.



Obr. 15: Vypočítané frekvenční spektrum zvuku požárního hlásiče

Při srovnání frekvenčních spekter obou signálů (Obr. 13, Obr. 15) je zřejmé, že vyhodnocení zvukového signálu požárního hlásiče bude nepopíratelně jednodušší než u tlesknutí. Prvním důvodem je to, že zvukový signál bude při opakovaném měření téměř pořád stejný. Je to dáno neměnností zvukového signálu z piezo-měniče požárního hlásiče na rozdíl od tlesknutí, které bylo při opakovaném měření pokaždé trochu jiné. Druhý důvod vyplývá z frekvenčního spektra požárního hlásiče, obsahuje pouze jednu dominantní frekvenci, která se bude daleko lépe vyhodnocovat.

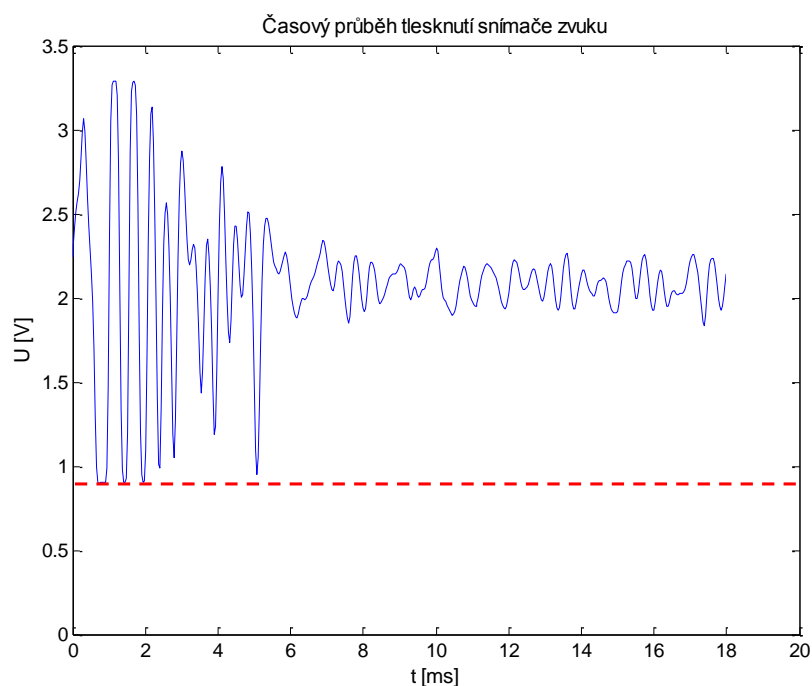
S těmito novými poznatky jsem se pustil do dalšího postupu výroby systému pro rozpoznávání zvuku.

5.2 Snímač zvuku

Po zjištění průběhů signálů zvuků bylo nutné zkonstruovat snímač zvuku. Podrobný popis jeho konstrukce je popsán v kapitole Snímač zvuku.

Snímač zvuku jsem zapojil na nepájivém kontaktním poli, kde jsem také prováděl různé úpravy. První úpravou byla dolní propust zapojena na výstupu snímače kvůli odfiltrování stejnosměrné složky. Tím se však zhoršily vlastnosti snímače, který byl pasivní dolní propustí tlumen. Úpravou byl odstraněn i offset. Po odstranění snímač snímal pouze kladnou polovinu signálu. Proto jsem tuto úpravu odstranil a stejnosměrnou složku filtruji až v softwaru.

Další úpravou je porovnávání výstupního signálu ze snímače pomocí operačního zesilovače z důvodu nastavování citlivosti pomocí elektroniky snímače a ulehčení práce mikroprocesoru. Po příchodu signálu externího triggru mikroprocesor začne vzorkovat výstupní signál ze snímače zvuku a ukládat do paměti (více v kapitole Externí trigger).

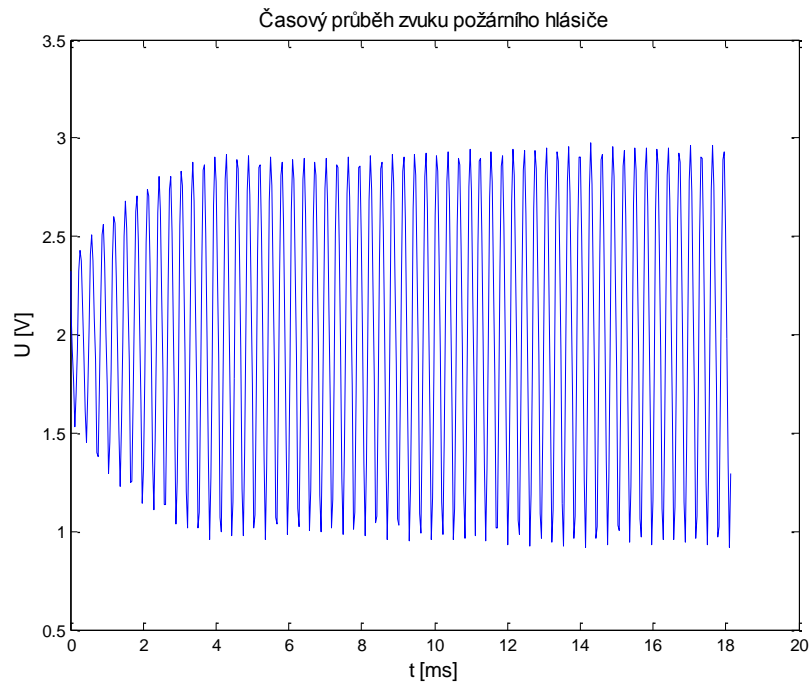


Obr. 16: Průběh tlesknutí snímaný snímačem zvuku měřený mikroprocesorem, $N=512$, $t=18,17$ ms

Na Obr. 16 je zobrazen časový průběh tlesknutí snímaný snímačem zvuku. Je zde vidět průběh pouze části celkového časového signálu tlesknutí Obr. 12. Jedná se o počáteční signál s nejvyššími amplitudovými pulsy (přibližně (30 ms)). Výběr této části signálu není náhodný, vedly k tomu dva důvody. Prvním důvodem je poměrně malý paměťový prostor pro nasbírané vzorky signálu při vysoké vzorkovací frekvenci. Tento problém lze vyřešit snížením vzorkovací frekvence, což by vedlo k menšímu počtu vzorků za stejný časový interval. Toto řešení by vedlo ke snížení přesnosti převodu mezi analogovým signálem a digitálním. Také by vznikla větší náchylnost na aliasing efekt. Druhým důvodem je počítání frekvenčního spektra ze signálu, který obsahuje zbytečně dlouhý časový průběh, ve kterém je obsaženo zkreslení a šumy stejné amplitudy jako zpracováváný signál. Kvůli těmto důvodům jsem využil pouze malou část časového průběhu, viz Obr. 16, který v takové míře nepodléhá těmto nežádoucím vlivům.

Na časovém průběhu Obr. 16 je vidět i výše zmiňovaný offset, který začíná na hodnotě 2 V a mezní hranice OZ je 0,92 V.

Snímání zvukového signálu požárního hlásiče provádím stejným snímačem, takže nastavení je totožné jako pro tlesknutí. V případě tohoto zvuku to nečiní žádný problém. Jak je vidět s porovnáním nezkráceného časového průběhu Obr. 14 průběhu s průběhem zkráceného signálu Obr. 17. Liší se pouze délkou signálu a amplitudovými změnami v čase, ale to při vyhodnocení nemá tak zásadní význam.



Obr. 17: Průběh požárního hlásiče snímaný snímače zvuku měřený mikroprocesorem, $N=512$, $t=18,17$ ms

5.3 Software

Program je psaný v jazyku C++. Veškerý kód je kompilovaný pomocí online kompilátoru na stránce mbed.org. Pro analýzu frekvenčního spektra tlesknutí jsem využil již hotovou knihovnu s rychlou Fourierovou transformací, která je volně dostupná na mbedu. Ta využívá rekurzivní DIT radix-2 FFT algoritmus podrobně popsany v kapitole Algoritmus radix-2. Pro analýzu frekvenčního spektra zvuku požárního hlásiče jsem napsal vlastní výpočetní algoritmus na základě metody Goertzelova algoritmu, který je popsán v kapitole Goertzelův algoritmus.

Programová část začíná vzorkováním signálu, poté následuje přesné odfiltrování nežádoucích frekvencí a v neposlední řadě vyhodnocení vstupních zvuků.

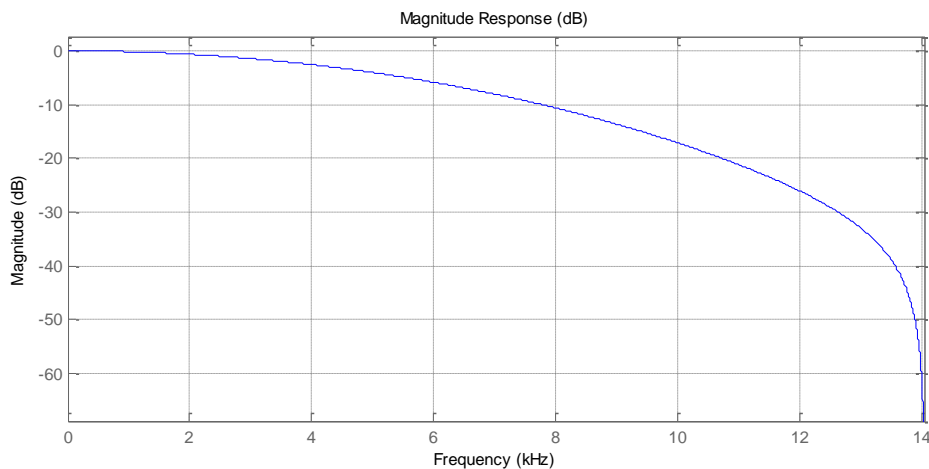
5.3.1 Vzorkování vstupního signálu

Vzorkování signálu začíná od příchodu externího triggru. Vzorkovací frekvence je dána základní třídou. Ta posílá signál v časovém intervalu do ADC. Výsledná vzorkovací frekvence se nepatrně liší s odlišností vstupního signálu. Její přibližná hodnota je $f_s = 28$ kHz. Vzorkování je zapouzdřeno ve smyčce, která zajišťuje zadaný počet vzorků. Počet vzorků se nastavuje do pomocného makra N , které je přidáno z důvodu šetření místa v paměti. Počet vzorků N je nastaven na hodnotu 512 a výsledná data se ukládají do pole prvků v paměti. S ohledem na dobu vzorkování 18 ms je tento počet dostačující. Výsledný diskretní průběh je zobrazen na Obr. 16.

Doba vzorkování je měřena z důvodu přesného výpočtu vzorkovací frekvence. Ta je důležitá pro přesný výpočet jednotlivých frekvencí pomocí Goertzelova algoritmu na vyhodnocení zvuku požárního hlásiče.

5.3.2 Digitální FIR filtr

Jak již bylo uvedeno výše, systém obsahuje i digitální FIR filtr pátého řádu. Filtr jsem do systému zařadil z důvodu přesného odfiltrování nežádoucích frekvencí a tím potlačení šumů. Ve snímači zvuku už je zařazen analogový filtr, ale ten slouží pouze k potlačení aliasing efektu. Frekvenci řezu jsem zvolil 5 kHz. Frekvenční charakteristika je zobrazena na Obr. 18.

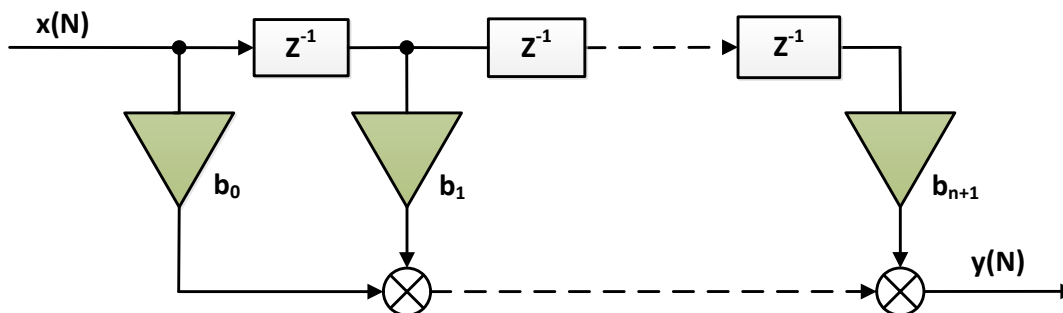


Obr. 18: Frekvenční charakteristika FIR filtru fo=5kHz

FIR filtr jsem zvolil díky jeho dobrým vlastnostem a jednoduchosti návrhu. FIR filtr obsahuje pouze dopřednou vazbu, proto je vždy stabilní. Výhodou FIR filtru oproti IIR je tedy stabilita, velká robustnost a konečná impulsní charakteristika. Nevýhody oproti IIR jsou zpoždění a poměrně vysoký řád pro strmé oříznutí.

Návrh a vypočítání koeficientů jsem prováděl pomocí MATLABu. Tyto koeficienty stanovují přenosovou charakteristiku filtru. Výsledný počet koeficientů je šest, protože platí $n + 1$ koeficientů na řád n . Koeficienty s velkou přesností desetinných míst jsem uložil do paměti, abych nezatěžoval mikroprocesor stejnými výpočty pořád dokola. Filtr zpožďuje signál o $\frac{n+1}{2}$, kde n je řád filtru.

Blokové schéma realizace FIR filtru pro programové použití je znázorněno na Obr. 19, kde $x(N)$ jsou vstupní data z A/D převodníku, $y(N)$ jsou výstupní data z filtru a b jsou koeficienty filtru. [18]



Obr. 19: Blokové schéma FIR filtru

5.3.3 Vyhodnocení příchozích zvuků

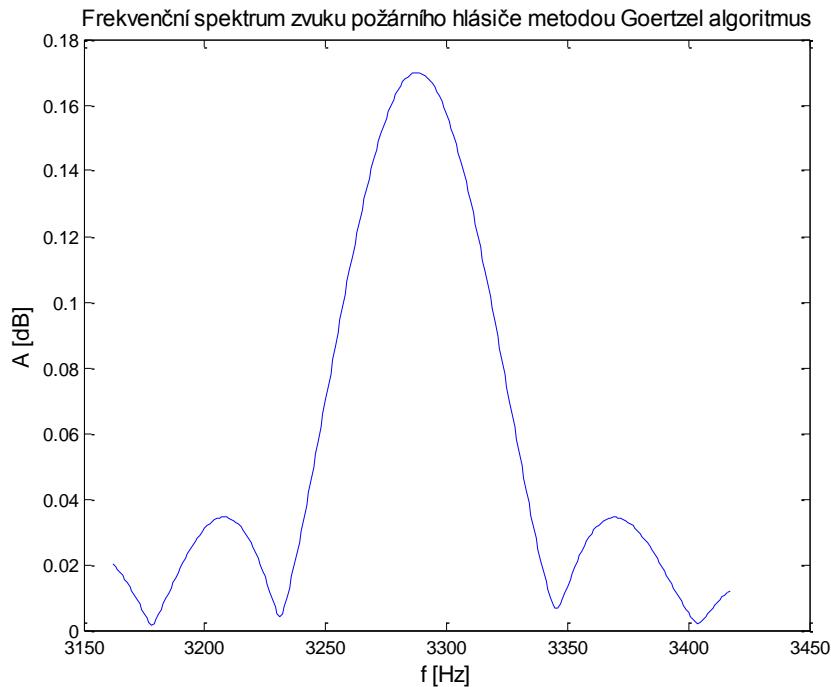
Vyhodnocení příchozích zvuků je nejdůležitější částí celého systému. Bylo potřeba vyhodnocovat dva příchozí zvuky, tlesknutí a zvuk požárního hlásiče.

Na začátku řešení této části jsem postupoval tak, že jsem našel pomocí FFT dominantní frekvence příchozích zvuků, které jsem následně zařadil do algoritmu vyhodnocení. Tento postup byl použitelný pouze pro zvuk požárního hlásiče, pro tlesknutí se ukázal tento postup jako nepoužitelný. Přesto u obou zvuků byl problém, jak vyhodnotit dominantní frekvenci. Musela by být určitá referenční hodnota amplitudy, kterou když dominantní frekvence překoná, vyhodnocení vyhodnotí shodu a tím je rozpoznání hotové. Bohužel amplituda zvuků není stále konstantní, mění se se vzdáleností od snímače nebo intenzitou tlesknutí. Navíc u vyhodnocení tlesknutí nastával ještě problém s podobnými zvuky, které obsahují stejnou frekvenci s nižší amplitudou než dominantní. Při přiblížení ke snímači se zvýší amplitudy určitých frekvencí a nastává chyba vyhodnocení špatného zvuku. Možnost odstranění tohoto problému je rozšíření vyhodnocovacích frekvencí. Tím se riziko špatného vyhodnocení sníží, ale stále přechovává problém s referenční hodnotou amplitudy.

Navrhl jsem proto vylepšené vyhodnocení systému. Nejprve jsem oddělil vyhodnocení zvuku požárního hlásiče od vyhodnocení tlesknutí a zvuky vyhodnocuji jednotlivě. Vzápětí jsem otestoval několik druhů rozpoznávacích podmínek a vybíral, která z nich nejlépe rozpozná daný zvuk.

5.3.3.1 Vyhodnocení zvuku požárního hlásiče

Na vyhodnocení zvuku požárního hlásiče jsem aplikoval Goertzelův algoritmus z důvodu vyhodnocování pouze jedné frekvence, viz Obr. 15. Pro zjištění amplitudy frekvence jsem napsal v programu funkci maximum, kterou zjišťuji, zda maximum leží na správné frekvenci. Pro zjištění maxima je zapotřebí více než jedna frekvence, proto vypočítávám Goertzelovým algoritmem interval s velkými kroky kolem vybrané frekvence. Výsledné frekvenční spektrum vyhodnocovací podmínky pro zvuk požárního hlásiče je zobrazeno na Obr. 20.



Obr. 20: Frekvenční spektrum zvuku požárního hlásiče vypočítané mikroprocesorem

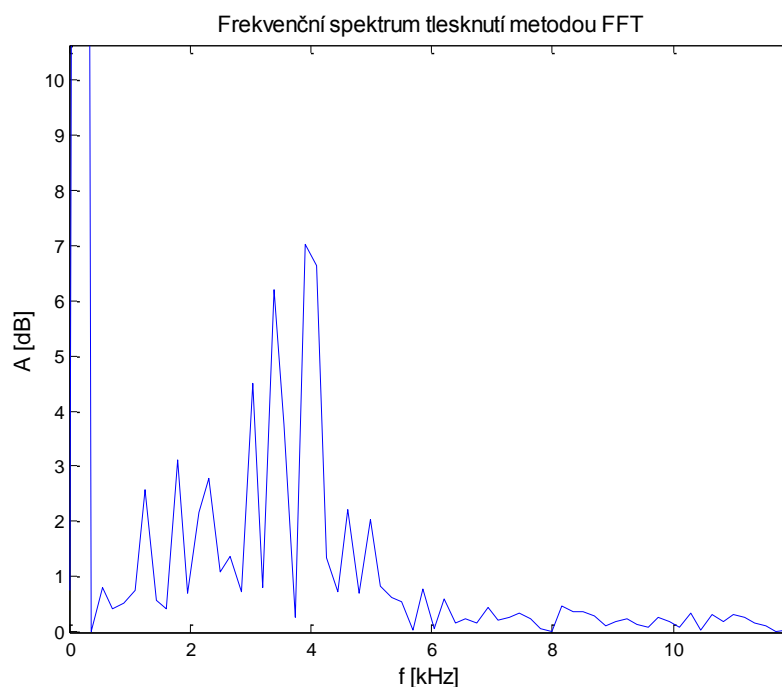
Vyhodnocení frekvence je pouze první podmínkou vyhodnocení. Druhou podmínkou je správný počet zero-cross (přechodů přes počátek signálu), který se nepočítá z celého signálu ale pouze pro 200 vzorků. Spojením těchto dvou podmínek mám identifikaci jak v časovém průběhu, tak i ve frekvenčním spektru. Pro správné vyhodnocení zvuku požárního hlásiče musí být splněny obě podmínky najednou.

5.3.3.2 Vyhodnocení tlesknutí

Vyhodnocení tlesknutí bylo oproti zvuku požárního hlásiče daleko složitější z důvodu velké rozmanitosti jednotlivých typů tlesknutí. Z tohoto důvodu jsem při postupování navrhl úpravu sjednocení určitého typu tlesknutí tím, že si uživatel před používáním systému nahraje pár (např. 10) tlesknutí, pro které se přizpůsobí podmínky vyhodnocování. I přes toto zjednodušení se mi podařilo najít takové podmínky, které zahrnují větší množství typů tlesknutí.

Začal jsem vybráním úseku časového průběhu tlesknutí s nejharmoničtějším průběhem už ze zkráceného časového průběhu. Když se podíváme na Obr. 16, vidíme na počátku průběhu nejlepší signál na vyhodnocení, vysokou amplitudu a přibližnou harmoničnost. Jedná se přibližně o interval od 1 ms do 6 ms. Vyhodnocení tlesknutí vyhodnocuji pomocí FFT radix-2 algoritmus, který musí mít počet vzorků násobky mocniny 2. Z tohoto důvodu má vybraná část pouze 128 vzorků, ale naopak to podstatně zrychluje výpočet. Data následně pošlu do rychlé Fourierovy transformace přiložené v samostatné knihovně. Funkce je volána pro reálná data $vrealFFT(Data, n)$, vstupními parametry této funkce jsou data a počet vzorků. Výstupní data se uloží do stejného pole jako data vstupní a to z důvodu šetření místa v

paměti. Výsledné frekvenční spektrum tlesknutí je zobrazeno na Obr. 21. Na začátku spektra v nízkých frekvencích je vidět stejnosměrná složka signálu, která nijak neovlivňuje vyhodnocovací spektrum.



Obr. 21: Frekvenční spektrum tlesknutí vypočítané mikroprocesorem

Pokračoval jsem ve výběru podmínek pro vyhodnocení tlesknutí:

1. podmínkou je stejně jako u požárního hlásiče správný počet zero-cross (přechodů přes počátek). Počet není jedno číslo, ale interval hodnot, z důvodu rozdílného typu tlesknutí při opakování.
2. podmínkou je správné umístění dominantní (maximální amplitudy) frekvence v intervalu frekvencí.
3. podmínkou je opět interval frekvencí. Zatímco podmínka č. 2 hledá dominantní frekvence v určitém intervalu, podmínka č. 3 kontroluje frekvenční pásmo, ve kterém mají být amplitudy frekvencí nižší než poměr maximální amplitudy a nastavené hodnoty. Jedná se o interval nízkých frekvencí.
4. a poslední podmínkou kontroluji výkon tří intervalů frekvenčního spektra tlesknutí. Výkonem se rozumí součet všech amplitud vzorků v daných frekvenčních intervalech. První interval počítá výkon nízkých frekvencí, druhý středních a třetí vysokých frekvencí. Podmínka je splněna tehdy, když výsledný výkon je největší ve druhém intervalu a zároveň rozdíl mezi druhým a prvním intervalem je větší než jedna.

Stejně jako u vyhodnocení zvuku požárního hlásiče mám podmínky v časovém průběhu i frekvenčním spektru. Za pomoci těchto čtyř podmínek vyhodnocuji výsledný zvuk tlesknutí.

5.3.3.3 Nastavení hodnot podmínek

Po výběru podmínek přišlo na řadu jejich optimální nastavení vzhledem k požadavkům systému. Nejprve jsem si zobrazoval průběhy různých zvuků a vyhodnocoval určité hodnoty podmínek. Tento postup byl pouze rámcový, neměl jsem žádná výsledná srovnávací data, a proto jsem přešel ke statistickému měření. Naměřil jsem si soubor trénovacích zvuků, který obsahuje vybrané vyhodnocovací zvuky (tlesknutí, zvuk požárního hlásiče) a ostatní zvuky. Především šlo o zvuky podobné vyhodnocovacím nebo o hlasité zvuky vyskytující se v daném místnosti (bouchnutí dveřmi atd.). Celkový počet zvuků v souboru je 130. Tyto zvuky jsou rozděleny do 13 skupin. Jedna skupina obsahuje 10 podobně způsobených zvuků (dopad klíčů o stůl, o zem atd.). A pro různá nastavení podmínek systému jsem prováděl binární klasifikační zkoušku. Každému nastavení podmínek náleží jedna matice změn (confusion matrix) o rozměru 3x3. Abych vyhodnotil přesnost systému pro oba vyhodnocovací zvuky zvlášť, rozdělíme matici 3x3 na dvě matice 2x2. To znamená, že pro jedno nastavení systému jsem provedl binární klasifikační zkoušku celého souboru zvuků. Poté jsem nastavení pozměnil a provedl zkoušku znovu. Na Obr. 22 je příklad matice pro výpočet sensitivity a specificity tlesknutí pro určité nastavení. Výpočet sensitivity a specificity je dán vztahem (5.1).

		Skutečný výsledek	
		Tlesknutí	Ostatní
Výsledek vyhodnocení systému	Tlesknutí	8 (True positive)	9 (False positive)
	Ostatní	2 (False negative)	101 (True negative)

		Skutečný výsledek	
		Hlásič	Ostatní
Výsledek vyhodnocení systému	Hlásič	10 (True positive)	0 (False positive)
	Ostatní	0 (False negative)	120 (True negative)

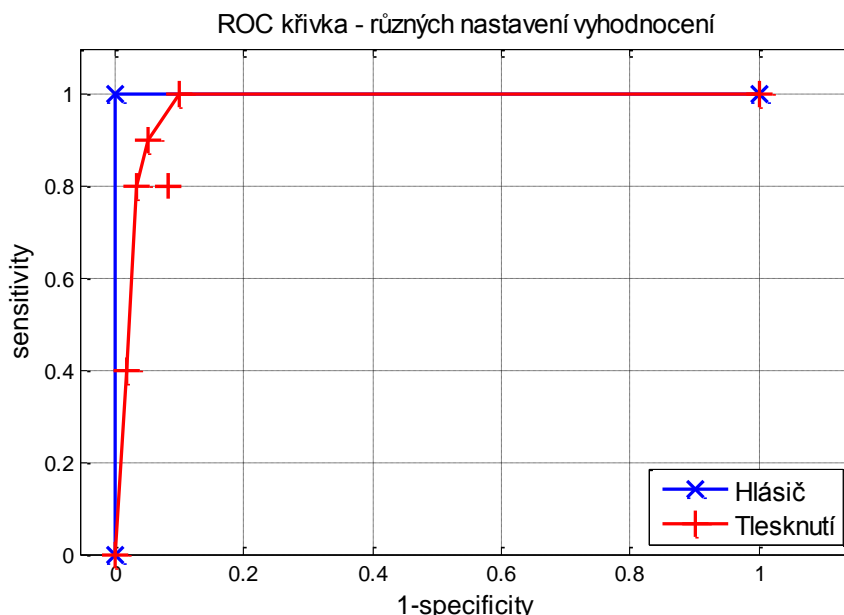
Obr. 22: Matice klasifikace a výpočet sensitivity a specificity

$$Sensitivity = \frac{True\ positive}{True\ positive + False\ negative}$$

$$Specificity = \frac{True\ negative}{True\ negative + False\ positive} \quad (5.1)$$

Výsledkem celkového zkoušení je křivka v tzv. ROC prostoru. Je to vztah mezi *sensitivitou* a $1 - \textit{specificitou}$. Na Obr. 23 je zobrazena křivka v ROC prostoru. Každý označený bod na křivce náleží jednomu nastavení podmínek systému. Cílem zkoušky je přiblížit se co nejvíce bodu (0; 1) v levém horním rohu. Mohou také nastat případy s různou prioritou vyhodnocení, poté se výsledný bod nastavení posouvá více vpravo nebo dolů.

Z křivky na Obr. 23 lze vidět, že žádný z testovaných zvuků nenarušuje vyhodnocení požárního hlásiče. Naopak u vyhodnocení tlesknutí je několik zvuků, které systém vyhodnotil chybně. Proto jsem zvolil nastavení podmínek pro tlesknutí umístěné nejbližše bodu (0; 1), jak je vidět na Obr. 23. Krajní bod (1; 1) je část, kdy systém vyhodnotí všechny příchozí zvuky za správné a bod (0; 0) nevyhodnotí žádný zvuk správně. Pouze jediný bod je pod křivkou, což je to dáno počtem měření. Tento bod náleží k výchozímu nastavení podmínek.

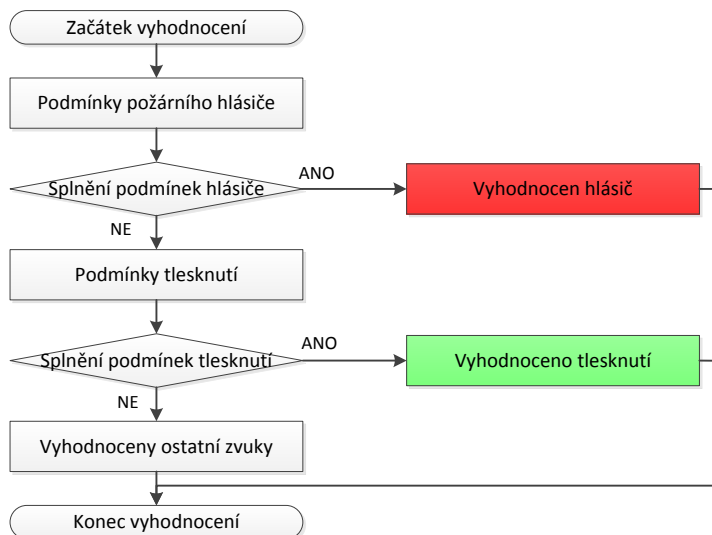


Obr. 23: ROC křivka - různých nastavení vyhodnocení systému

5.3.3.4 Výsledný vyhodnocovací algoritmus

Od vyhodnocení systému je vyžadováno, aby nevyhodnocoval chybně zvuky požárního hlásiče. Tento důvod je prostý, požární hlásič upozorňuje na výskyt požáru nebo případné nebezpečí. Ve většině případů je spojen s bezpečnostním systémem objektu, který při hlášení alarmuje hasiče. Proto systém především nesmí tento zvuk nevyhodnotit, ale i nezaměňovat za ostatní zvuky. U vyhodnocení tlesknutí je tato priorita jiná. Systém je nastaven na naučený typ uživatelského tlesknutí, a tudíž se snaží vyhodnotit pouze daný typ zvuku i za předpokladu nevyhodnocení správného zvuku. Určitě bude nepříjemné, když uživatel bude chtít rozsvítit světla a na první příkaz se to nepovede, ale nebude to tolik závažný problém jako u požárního hlásiče.

Z testovacích výsledků plyne, že požadavek na vyhodnocení požárního hlásiče alespoň z vyzkoušených zvuků je splněn. Vyhodnocení požárního hlásiče mělo při testování 100 % úspěšnost správného vyhodnocení.



Obr. 24: Vývojový diagram vyhodnocení

Výsledný vyhodnocovací algoritmus je zobrazen na vývojovém diagramu Obr. 24. Z diagramu plyne, že jako první se testuje příchozí zvuk na podmínky požárního hlásiče a pokud nejsou podmínky splněny, tak se testují podmínky tlesknutí.

Pokud není splněna ani jedna z podmínek (tlesknutí, hlásič), systém vyhodnotí ostatní zvuky. Když systém vyhodnotí tento stav, neprovádí žádný úkol, pouze přejde na začátek celého algoritmu. Zde čeká na příkaz externího triggru, který značí nově příchozí zvuk.

5.3.3.5 Vylepšení vyhodnocování tlesknutí

Z důvodu neuspokojivých výsledků vyhodnocení tlesknutí jsem navrhl vylepšení. Jedná se o opakované vyhodnocení dvou tlesknutí v určitém časovém intervalu, tzv. zdvojené tlesknutí. To znamená, že pro správné vyhodnocení tlesknutí musí po sobě následovat dva zvuky, které vyhodnotí podmínky vyhodnocení jako tlesknutí a navíc musí dodržet časový interval opakování. Při nedodržení tohoto intervalu je vyhodnocení stejné jako pro ostatní zvuky.

Náhodné zvuky se ve většině případů dvakrát stejně neopakují. Z tohoto zjištění vychází i moje úprava. S tímto vylepšením by se daly i zvolnit podmínky vyhodnocení a pokrýt tak většinu typů tlesknutí.

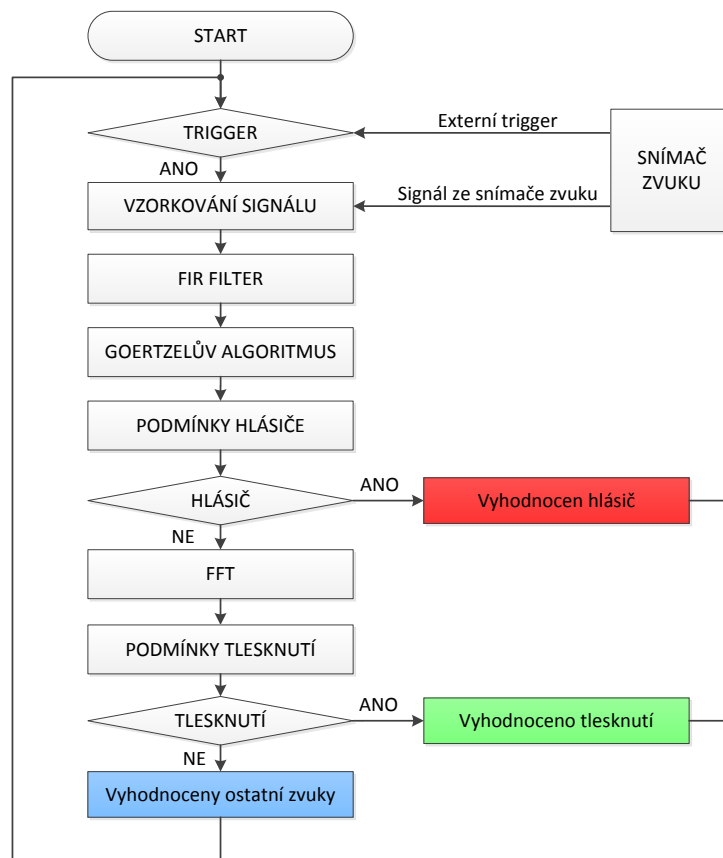
Toto vylepšení je realizováno, ale do systému není zařazeno. Výsledkem je vylepšení vyhodnocení, ale za cenu obtížnějšího ovládání.

Proto jsem při testování zvolil vyhodnocení pouze jednoho tlesknutí, jak je popsáno v kapitole Vyhodnocení tlesknutí. Při zařazení do ovládání budovy by byla tato alternativa nabídnuta uživateli.

5.4 Shrnutí

Po zapnutí systém čeká na příchozí zvuk. Jestliže příchozí zvuk překročí hranici nastavenou pomocí externího triggru, mikroprocesor začne okamžitě vzorkovat vstup snímače zvuku se vzorkovací frekvencí 28180 Hz. Stanovený počet vzorků se ukládá do jednorozměrného pole do FLASH paměti mikroprocesoru.

V dalším kroku je diskrétní signál přesně vyfiltrován digitálním FIR filtrem s frekvencí řezu 5 kHz. Poté je ze signálu vypočítáno úzké frekvenční spektrum pro určení zvuku požárního hlásiče. Pokračují vyhodnocovací podmínky požárního hlásiče, po kterých následuje rozdělení algoritmu. Při vyhodnocení zvuku požárního hlásiče celý algoritmus končí a přechází opět na začátek před vzorkování dat. Jestli není vyhodnocen jako požární hlásič, algoritmus pokračuje dál a vstupuje do funkce FFT pro výpočet frekvenčního spektra tlesknutí. Vypočítané spektrum se uloží do původního pole s časovými vzorky. Od této chvíle není možno počítat s časovým signálem. Spektrum je dále prověřeno podmínkami vyhodnocení tlesknutí. V tomto místě se algoritmus opět dělí na dvě cesty. Pokud jsou podmínky splněny, je zvuk vyhodnocen jako tlesknutí a algoritmus přejde opět na začátek. Pokud nejsou podmínky splněny, algoritmus vyhodnotí ostatní zvuky a přechází opět na začátek před vzorkování signálu. Celkový algoritmus systému pro rozpoznávání zvuku je zobrazen na Obr. 25.



Obr. 25: Celkový algoritmus systému pro rozpoznávání zvuku

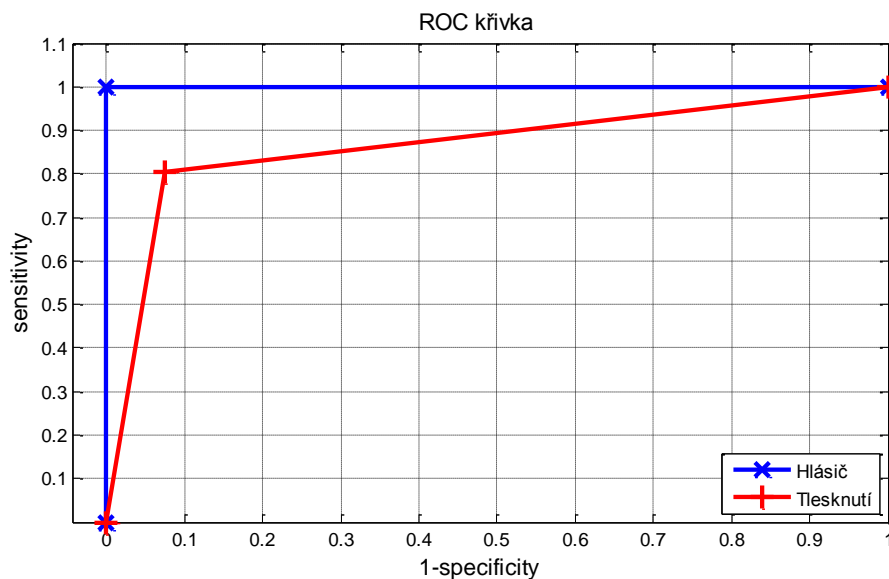
6 TESTOVÁNÍ SYSTÉMU

Testování probíhalo po dobu 6 hodin v místnosti se dvěma osobami. Test probíhal za normálních podmínek, kdy uživatel v průběhu měření tleskal a simuloval tím příkaz na rozsvícení světel. V měření také zazněl zvuk požárního hlásiče. Ostatní zvuky byly čistě náhodné a přirozené. Test měl určit pravděpodobnost správného nebo falešného vyhodnocení tlesknutí a požárního hlásiče. Systém byl nastaven na optimální nastavení vycházející z grafu Obr. 23, kdy senzitivita se rovná 0,9. Výsledná matice Obr. 26 a ilustrativní ROC křivka platná pouze pro body (0; 1) a (0,075; 0,8) je zobrazena na Obr. 27.

		Skutečný výsledek	
		Tlesknutí	Ostatní
Výsledek vyhodnocení systému	Tlesknutí	115 (True positive)	21 (False positive)
	Ostatní	28 (False negative)	255 (True negative)

		Skutečný výsledek	
		Hlásič	Ostatní
Výsledek vyhodnocení systému	Hlásič	4 (True positive)	0 (False positive)
	Ostatní	0 (False negative)	419 (True negative)

Obr. 26: Výsledná matice 6. hodinového klasifikačního měření



Obr. 27: Ilustrativní ROC křivka - 6 hodinového měření

Z matice na Obr. 26 vychází přesnost vyhodnocení správného výsledku tlesknutí na 88,42 % a vyhodnocení zvuku požárního hlásiče na 100 %.

Celkově bylo zaznamenáno 413 všech zvuků, které externí trigger vyhodnotil za hlasité. Skutečných tlesknutí bylo 143, z nichž systém vyhodnotil 136 jako tlesknutí. Správně systém vyhodnotil 115 tlesknutí, 21 tlesknutí vyhodnotil při jiných zvucích a na 10 tlesknutí nezareagoval a vyhodnotil je za ostatní. Skutečně byly zaznamenány 4

zvuky požárního hlásiče a systém vyhodnotil taktéž 4 zvuky. Žádná chyba při vyhodnocení zvuku požárního hlásiče nebyla zaznamenána. Během měření jsem objevil zvuky, které byly chybně vyhodnoceny jako tlesknutí. Šlo o jiné zvuky než naměřené v souboru trénovacích zvuků pro určení optimálního nastavení. Zvuku požárního hlásiče se přiblížil pouze zvuk vyzvánějícího telefonu, který však nebyl vyhodnocen jako tento zvuk.

Musím připomenout skutečnost, že podmínky tlesknutí jsou nastaveny na daný typ tlesknutí. Pokaždé se nepodaří tlesknout podle naučeného vzoru.

Dalšími poznatky z měření jsou problémová místa v místnosti například v rozích, kde se zvuk hodně láme a stává se tak zdrojem chybného vyhodnocení.

7 ZÁVĚR

Cílem mé práce bylo vytvoření systému pro rozpoznávání definovaných zvuků. Prvním je tlesknutí, které představuje ovládací prvek například pro osvětlení v místnosti. Druhým je zvuk požárního hlásiče, který zajišťuje bezpečnost a bezdrátovou komunikaci mezi systémem.

Systém se skládá ze dvou hlavních částí - snímače zvuku a vyhodnocovacího algoritmu. Snímač zvuku jsem vyrobil vlastní především z důvodu dlouhých dodacích lhůt výrobců snímačů. Snímač obsahuje mikrofon, antialiasingový filtr, zesilovač a externí trigger. Velkou pozornost jsem věnoval výběru vhodného mikrofону. Provedl jsem rešerši různých typů mikrofónů a na základě tohoto vyhodnocení jsem vybral elektretový mikrofon. Antialiasingový filtr je zařazen pro potlačení aliasingového efektu. Zesilovač slouží k zesílení výstupního signálu a externí trigger posílá signál při překročení definované zvukové úrovně do mikroprocesoru k zahájení vzorkování.

Hotovým snímačem jsem provedl naměření souboru 130 trénovacích zvuků různých typů. Především jde o hlasité zvuky, které se přirozeně vyskytují v místnosti. DPS snímače zvuku je navržena jako nastavný modul platformy FRDM-KL25Z, návrh je přiložen v příloze.

Výsledný vyhodnocovací algoritmus je rozdělen na dvě části, a to pro zvuk požárního hlásiče a pro tlesknutí. Vyhodnocení zvuku požárního hlásiče klade primární důraz na přesnost vyhodnocení při požáru. Pro vyhodnocení tlesknutí je systém nastaven na naučený typ uživatelského tlesknutí, a tudíž se snaží vyhodnotit pouze daný typ zvuku i za předpokladu nevyhodnocení správného zvuku.

Vyhodnocení zvuku požárního hlásiče je zpracováváno pomocí Goertzelova algoritmu, kterým lze počítat pouze zvolenou frekvenci, což je u zvuku požárního hlásiče možné. Tím šetřím výpočetní čas mikroprocesoru i zvětšuji přesnost výsledku. Tento algoritmus jsem naprogramoval do mikroprocesoru a je součástí vyhodnocení.

Vyhodnocení tlesknutí je řešeno pomocí rychlé Fourierovy transformace z důvodu sledování většího frekvenčního spektra. Díky zkrácení časového průběhu pouze na počátek zvukové vlny neobsahuje FFT tolik šumu. A proto lze provést i poměrně kvalitní vyhodnocení. Rychlou Fourierovu transformaci jsem implementoval pomocí převzaté knihovny do mikroprocesoru, který počítá dané hodnoty.

Pro výsledný verdikt vyhodnocení slouží optimálně nastavené podmínky, které jsou porovnávány s naměřenými hodnotami.

K ověření výsledných parametrů systému jsem provedl 6 hodinové měření, z jehož výsledků jsem vypočítal přesnost vyhodnocení tlesknutí na 88,42 % a vyhodnocení požárního hlásiče na 100 %. Tyto výsledky jsou obrazem systému pro vyhodnocení zvuku. Pro vylepšení vyhodnocení systému jsem vytvořil podmínku zdvojeného tlesknutí. To by mělo zvýšit přesnost výsledného vyhodnocení tlesknutí.

Do budoucna bych navrhl vylepšení systému a to především rozšířením podmínek vyhodnocení tlesknutí a jejich strojové určení. Vylepšení by mohlo přinést i nahrazení snímače zvuku za již zmiňovaný MEMS mikrofon.

LITERATURA

- [1] *Mbed* [online]. 2005 [cit. 2013-12-30]. Dostupné z: <http://mbed.org/handbook/mbed-FRDM-KL25Z>
- [2] FREEDOM KL25Z: BLOCK DIAGRAM. RAFAEL DEL REY. *Freescale* [online]. 2012, 12.6 [cit. 2013-12-30]. Dostupné z: http://cache.freescale.com/files/soft_dev_tools/hardware_tools/schematics/FRDM-KL25Z_SCH.pdf?fpsp=1
- [3] Electret Microphones. *Open Music Labs* [online]. 2013 [cit. 2013-12-30]. Dostupné z: <http://www.openmusiclabs.com/learning/sensors/electret-microphones>
- [4] Dual Operational Amplifiers: LM158, LM158A, LM258, LM258A LM358, LM358A, LM2904, LM2904V. Texas Instrument [online]. 2013 [cit. 2013-12-30]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm358.pdf>
- [5] Introduction to the Fourier Transform. *The Fourier Transform* [online]. 2009-2011 [cit. 2013-12-30]. Dostupné z: <http://www.thefouriertransform.com/transform/fourier.php>
- [6] PROF. ING. PAVEL JURA, CSc. *Signály a systémy: Část 1: Spojité signály* [online]. Druhé. Brno, 2010 [cit. 2013-12-30]. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=33240
- [7] KANKELBORG, C. *DFT and FFT* [online]. 2009 [cit. 2013-12-30]. Dostupné z: <http://solar.physics.montana.edu/kankel/ph567/examples/Octave/dft/dft.pdf>
- [8] RAHMAN, M. *Applications of Fourier transforms to generalized functions*. Boston, Mass.: WIT Press, 2011, s. 129-136. ISBN 1845645642.
- [9] The "Twiddle Factor". *AlwaysLearn* [online]. 2013 [cit. 2013-12-30]. Dostupné z: http://www.alwayslearn.com/dft%20and%20fft%20tutorial/DFTandFFT_FFT_TwiddleFactor.html
- [10] JONES. Decimation-in-time (DIT) Radix-2 FFT. *Connexions* [online]. 2006 [cit. 2013-12-30]. Dostupné z: <http://cnx.org/content/m12016/latest/>
- [11] OpenCL™ Optimization Case Study Fast Fourier Transform – Part 1: Radix-2 FFT Algorithm. *AMD* [online]. 2011 [cit. 2013-12-30]. Dostupné z: <http://developer.amd.com/resources/documentation-articles/articles-whitepapers/opencl-optimization-case-study-fast-fourier-transform-part-1/>
- [12] MAŠLÁŇ, Stanislav. Rychlá Fourierova transformace (FFT) pro AVR. *Elektronika kvalitně* [online]. 2012 [cit. 2013-12-30]. Dostupné z: <http://elektronika.kvalitne.cz/ATMEL/necoteorie/transformation/AVRFFT/AVRFFT.htm>

- [13] LYONS, Richard. Single tone detection with the Goertzel alhorithm. In: Embedded: cracking the code to systems development [online]. 19. 11. 2012 [cit. 2014-05-19]. Dostupné z: <http://www.embedded.com/design/real-world-applications/4401754/Single-tone-detection-with-the-Goertzel-algorithm>
- [14] MAHMOODA, M. VINOD KUMAR REDDY a Sagar NAYAKANTI. INTERNATIO. Implementation of Spectrum Analyzer using GOERTZEL Algorithm. International Journal of Scientific and Research Publications [online]. 2013, č. 3 [cit. 2014-05-19]. ISSN 2250-3153. Dostupné z: <http://www.ijsrp.org/research-paper-0313/ijsrp-p15135.pdf>
- [15] RUMSEY, Francis a Tim MCCORMICK. *Sound and recording*. 6th ed. Oxford: Focal Press, 2009, s. 48-53. ISBN 978-0-240-52163-3.
- [16] GLEN M. BALLOU, Glen M.editor. Handbook for sound engineers. 4th ed. Oxford: Focal, 2008. ISBN 02-408-0969-6.
- [17] LEWIS, Jerad. Analog and Digital MEMS Microphone Design Considerations. In: Industrial automation products [online]. Norwood, MA: Analog Devices, 2013 [cit. 29.5.2014]. ISBN 02062-9106ISSN 02062-9106. Dostupné z: http://www.analog.com/static/imported-files/tech_articles/Analog-and-Digital-MEMS-Microphone-Design-Considerations-MS-2472.pdf
- [18] DURAI, By C. Ramesh Babu. Digital signal processing. New Delhi: Laxmi Publications (P) Ltd, 2006. ISBN 81-700-8736-8.

SEZNAM TABULEK

Tab. 1: Bitová inverze pro $N=8$ [12].....	22
Tab. 2: Matematické srovnání metody FFT radix-2 a Goertzelova algoritmu [13]	25

SEZNAM OBRÁZKŮ

Obr. 1: Blokové schéma systému na rozpoznání zvuku.....	9
Obr. 2: Blokové schéma Freedom desky s mikroprocesorem KL25Z [2]	11
Obr. 3: Schéma zapojení snímače	12
Obr. 4: Blokové schéma I ² S MEMS mikrofonu [17].....	14
Obr. 5: Frekvenční charakteristika dolní aktivní propusti.....	16
Obr. 6: Samostatné schéma externího triggeru	17
Obr. 7: Dělení v čase o délce N do dvou $N/2$ částí [10].....	20
Obr. 8: Základní motýlek [11]	21
Obr. 9: Radix-2 DIT FFT algoritmus [10]	21
Obr. 10: a) rozložení pólů a nul filtru, b) odezva frekvence ve frekvenčním spektru [13].....	23
Obr. 11: Blokové schéma zapojení Goertzelova algoritmu	24
Obr. 12: Naměřený průběh tlesknutí.....	26
Obr. 13: Vypočítané frekvenční spektrum tlesknutí	27
Obr. 14: Naměřený časový průběh zvuku požárního hlásiče.....	27
Obr. 15: Vypočítané frekvenční spektrum zvuku požárního hlásiče	28
Obr. 16: Průběh tlesknutí snímáný snímače zvuku měřený mikroprocesorem, $N=512$, $t=18,17$ ms	29
Obr. 17: Průběh požárního hlásiče snímáný snímače zvuku měřený mikroprocesorem, $N=512$, $t=18,17$ ms.....	30
Obr. 18: Frekvenční charakteristika FIR filtru $f_0=5$ kHz.....	31
Obr. 19: Blokové schéma FIR filtru.....	32
Obr. 20: Frekvenční spektrum zvuku požárního hlásiče vypočítané mikroprocesorem	33
Obr. 21: Frekvenční spektrum tlesknutí vypočítané mikroprocesorem	34
Obr. 22: Matice klasifikace a výpočet sensitivity a specifcity	35
Obr. 23: ROC křivka - různých nastavení vyhodnocení systému	36
Obr. 24: Vývojový diagram vyhodnocení.....	37
Obr. 25: Celkový algoritmus systému pro rozpoznávání zvuku	38
Obr. 26: Výsledná matice 6. hodinového klasifikačního měření	39
Obr. 27: Ilustrativní ROC křivka - 6 hodinového měření	39

SEZNAM PŘÍLOH

A	Měření v MATLABu.....	47
A.1	SIMULINK načítání mikrofonu.....	47
A.2	Frekvenční spektrum tlesknutí	47
B	Program systému pro rozpoznávání zvuku.....	48
C	Návrh DPS snímače zvuku.....	52
C.1	Osazovací plán DPS snímače zvuku	52
C.2	Spodní vrstva DPS snímače zvuku	52
C.3	Schéma snímače zvuku	53
D	Příložené CD	54

A MĚŘENÍ V MATLABU

A.1 SIMULINK načítání mikrofону



A.2 Frekvenční spektrum tlesknutí

```
clear all
close all
clc

sim('mic')
X=rot90(A);
Fs = 44100; % Vzorkovací frekvence
L=22528; % Počet vzorku
T = 1/Fs; % Perioda vzorkování
t = (0:L-1)*T; % Casový vektor
plot(A)

NFFT = 2^nextpow2(L); % FFT
Y = fft(X,NFFT)/L;
f = (Fs/2*linspace(0,1,NFFT/2+1));

a=2*abs(Y(1:NFFT/2+1));

title('Signal Corrupted')
xlabel('time (seconds)')

plot(f,a)
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```

B PROGRAM SYSTÉMU PRO ROZPOZNÁVÁNÍ ZVUKU

```
#include "mbed.h"
#include "FFT.h"
#define N 512
#define n 5
#define PI 3.1415

DigitalOut myled(LED_BLUE);
DigitalOut myled1(LED_GREEN);
DigitalOut myled2(LED_RED);
AnalogIn mic (PTB0);
AnalogIn ext (PTB1);
Timer t;

////////////////////////////////////
int x,f,i; //pomocné proměnné
float Q1=0,Q2=0,Q0,coeff,w,cosine,sine,real,im,ko,Fs; //Goertzel
proměnné
float Data[N+1];
float Power[17];
float h[n+1]={0.00436686802192238, 0.106146818796024,
0.389486313182053, 0.389486313182053, 0.106146818796024,
0.00436686802192238}; //koeficienty pro FIR filtr z Matlabu pro
frekvenci řezu 5kHz
float max,V,max_1,max_2; //výpočty identifikace
int index,nn,id; //výpočty identifikace
float buffer,souc1,souc2,souc3;
bool ok,pod01,pod1,pod2,pod3,pod4,pod5; //podmínky
////////////////////////////////////

int main()
{
    myled=1;
    myled1=1;
    myled2=1;
    while(1)
    {
        //////////////////////////////////////
        //nulování proměnných
        index=0;
        souc1=0;
        souc2=0;
        souc3=0;
        max_1 =0;
        max_2 =0;
        id=0;
        //////////////////////////////////////
        //nulování boolovských proměnných
        ok=false;
        pod01=false;
        pod1=false;
        pod2=false;
    }
}
```

```

    pod3=false;
    pod4=false;
    pod5=false;
    //////////////////////////////////////
    //externí trigger
    if(ext.read()>0.7)
    {
        t.reset();
        x=0;
        myled=0.8;
        t.start();
    }
    //////////////////////////////////////
    //vzorkování dat
    while(x<N)
    {
        Data[x] = mic.read();
        x++;
    }
    //////////////////////////////////////
    //výpočet aktuální vzorkovací frekvence Fs
    t.stop();
    Fs=N/t.read();
    //////////////////////////////////////
    //Digitalní FIR filtr frekvence řezu 5kHz
    for(int k=0;k<N;k++)
    {
        buffer=0;
        for(x=0;x<n+1;x++)
        {
            buffer += h[x]*Data[k+x];
        }
        Data[k]=buffer;
    }
    //////////////////////////////////////
    //Gortzel algoritmus pro vyhodnocení požárního hlásiče
    x=0;
    max=0;
    for(f=3162;f<3418;f=f+16)
    {
        ko=(0.5+(N*f))/Fs;
        w=(2*PI*ko)/N;
        cosine=cos(w);
        sine=sin(w);
        coeff=2*cosine;
        Q1=0;
        Q2=0;
        Q0=0;
        for(i=0;i<N;i++)
        {
            Q0=coeff*Q1-Q2+Data[i];
            Q2=Q1;
            Q1=Q0;
        }
        real=(Q1-Q2*cosine)/(N/2);
        im=(Q2*sine)/(N/2);
        Power[x]=sqrt(real*real+im*im);
        if(Power[x] > max)//hledání maxima spektra požárního
hlásiče
    {

```

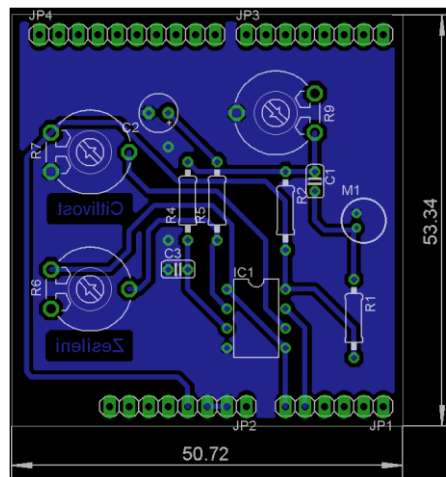
```

        max = Power[x];
        id=x;
    }
    x++;
}
////////////////////////////////////
//podmínka 01
index=0;
for(int k=0;k<200;k++)
{
    if((Data[k]>0.63) && (Data[k+1]<0.63))//zero-cross
    {
        index=index+1;
    }
}
if(index==23 && (id==7 || id==8))
{
    pod01=true;
}
////////////////////////////////////
//výsledná podmínka požárního hlásiče
if(pod01==true)
{
    myled=1;
    myled2=0.8;
    wait(0.5);
    myled2=1;
}
////////////////////////////////////
//rozdělení algoritmu na dvě části
else
{
    for(int k=50;k<178;k++)
    {
        Data[k-50]=Data[k];
    }
}
////////////////////////////////////
//FFT
nn=128;
vRealFFT(Data, nn);
myled=1;
////////////////////////////////////
//hledání maxima spektra tlesknutí
id=0;
max=0;
for(int k=2;k<42;k++)
{
    if(abs(Data[k]) > max)
    {
        max = abs(Data[k]);
        id=k;
    }
}
////////////////////////////////////
//podmínka1
if(index>8 && index<16)
{
    pod1=true;
}

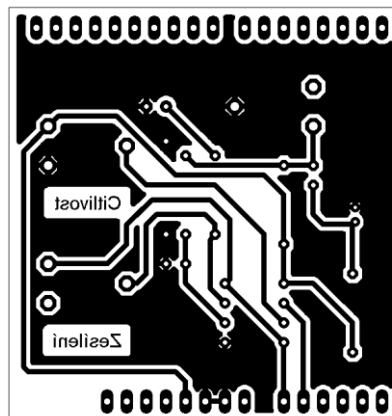
```


C NÁVRH DPS SNÍMAČE ZVUKU

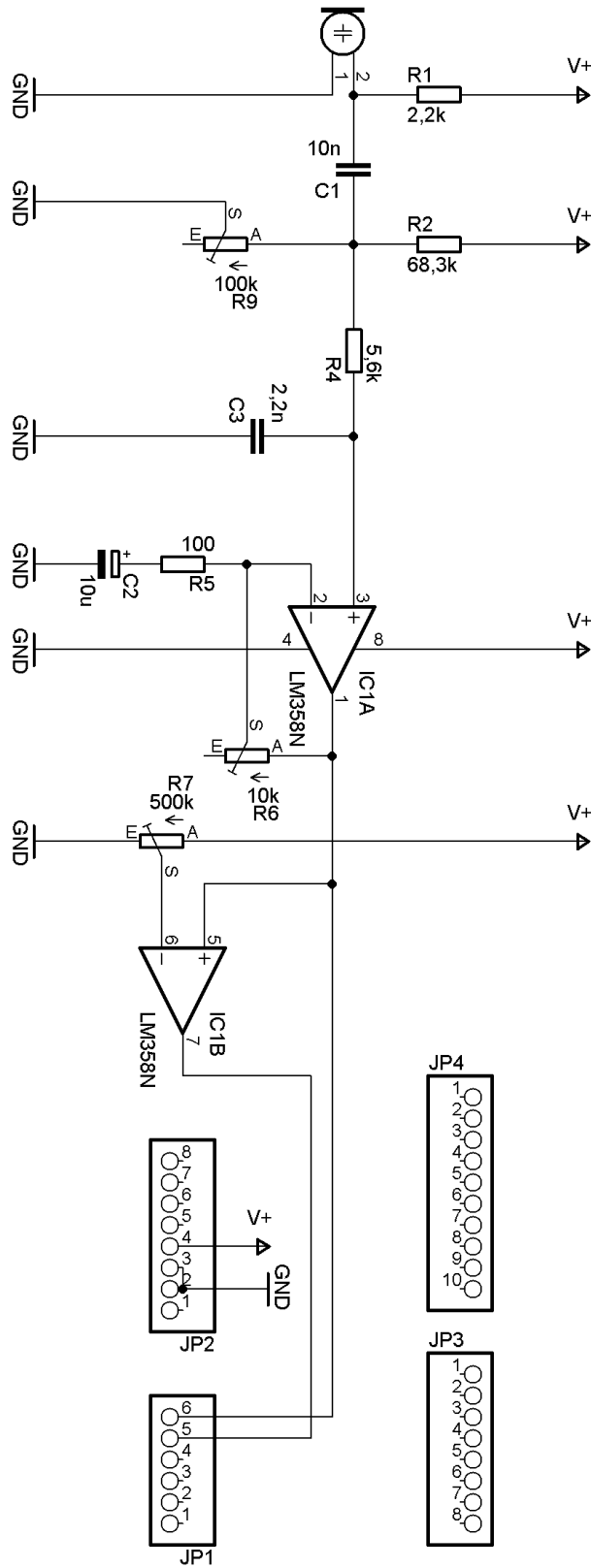
C.1 Osazovací plán DPS snímače zvuku



C.2 Spodní vrstva DPS snímače zvuku



C.3 Schéma snímače zvuku



D PŘILOŽENÉ CD

Na přiloženém CD se nachází kompletní text ve formátu .pdf, soubor 130 trénovacích zvuků, měřící programy z MATLABu, program a knihovna systému pro vyhodnocení zvuku pro FRDM-KL25Z, výpočty optimálního nastavení podmínek a návrhy DPS snímače zvuku v Eaglu.