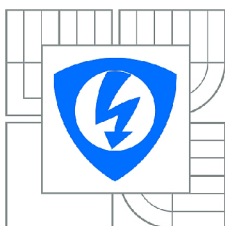


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTÉM PRO DETEKCI ZVUKU SOUND DETECTION SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

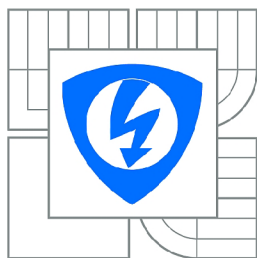
AUTOR PRÁCE
AUTHOR

YAROSLAV SAPRYKIN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ONDŘEJ HYNČICA

BRNO 2015



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: Yaroslav Saprykin

ID: 158222

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

System pro detekci zvuku

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření vestavného systému pro rozpoznávání předem daných určitých zvuků.

- 1) Seznamte se s používanými typy mikrofonů, jejich vlastnostmi, připojte vybraný mikrofon k mikrokontroléru.
- 2) Nasbírejte sadu zvuků pro další vyhodnocování.
- 3) Vyberte a vyzkoušejte vhodnou metodu pro rozpoznání zvuku, založenou například na rychlé Fourierově transformaci nebo dynamickém borcení času (DTW).
- 4) Metodu implementujte do mikrokontroléru a vyhodnoťte.

DOPORUČENÁ LITERATURA:

SLOSS, Andrew N, Dominic SYMES a Chris WRIGHT. ARM system developer's guide: designing and optimizing system software. Amsterdam: Elsevier, 2004, 689 s. ISBN 15-586-0874-5.

Termín zadání: 9.2.2015

Termín odevzdání: 25.5.2015

Vedoucí práce: Ing. Ondřej Hynčica

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Bakalářská práce je zaměřena na vytvoření systému pro detekci zvuku. Vytvoření jednoduchého algoritmu detekcí rozbití skla založeném na FFT a Goertzel algoritmech. Částí praktické realizací algoritmu je aplikace FIR filtru. Seznámení s metodou DTW a ověření teoretických poznatků na příkladu porovnání zvukových vzorků ptáků. Implementace vytvořeného algoritmu detekcí rozbití skla do mikrokontroléru na vývojové desce FRDM-KL25Z od firmy Freescale. Koncovým stupněm systému je návrh a praktická realizace modulu s elektretovým mikrofonem. Implementace algoritmu detekcí v C++.

Klíčová slova

detekce zvuku, systém rozpoznávání zvuku, algoritmus detekce, rozpoznávání zvuku, algoritmus detekcí rozbití skla, algoritmus rozpoznávání rozbití skla, detekce skla, rozpoznávání skla, modul s mikrofonem, elektretový mikrofon, Goertzel algoritmus, FFT, DTW.

Abstract

Bachelor thesis is focused on the developing sound detection system. Developing simple algorithm of detection breaking glass based on FFT and Goertzel algorithms. Part of practise realization of algorithm is application FIR filter. Acquaintance with the method DTW and verification of theoretical knowledges on the example of comparison sound samples of birds. Implementation of created algorithm of breaking glass detection to microcontroller based on development platform FRDM-KL25Z by Freescale. Terminal equipment of system are designed a practical realization module with electret microphone. Detection algorithm implementation in C++.

Keywords

sound detection, sound detection system, detection algorithm, sound recognition, breaking glass detection algorithm, breaking glass recognition algorithm, glass detection, module with microphone, electret microphone, Goertzel algorithm, FFT, DTW.

SAPRYKIN, Y. *Systém pro detekci zvuku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 61 s. Vedoucí Ing. Ondřej Hynčica.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma "Systém pro detekci 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 jsem si 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.

.....

V Brně dne

.....

(podpis autora)

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Ondřejovi Hynčicovi za perfektní spolupráci, odbornou pomoc na konzultacích a cenné rady při zpracování mé bakalářské práce.

.....

V Brně dne

.....

(podpis autora)

Obsah

1	Úvod	8
2	Rozbor	9
2.1	Knihovna vzorků	10
3	Výběr zařízení	11
3.1	Mikrokontrolér [2]	11
3.2	Mikrofony	11
3.2.1	Uhlíkový mikrofon	12
3.2.2	Piezoelektrický mikrofon	12
3.2.3	Dynamický mikrofon	13
3.2.4	Kondenzátorový mikrofon	14
3.2.5	MEMS mikrofon	15
3.2.6	Shrnutí výběru	15
3.3	Výběr modulu s mikrofonem	16
3.4	Připojení mikrofonu	17
4	Analýza	18
4.1	Měření v matlabu	18
4.2	Analýza pomocí FFT	18
4.3	Analýza pomocí spektrogramu	20
4.4	Metoda DTW	21
4.5	Shrnutí	26
5	Popis metod	27
5.1	Použití spektrogram	27
5.2	Aplikace FFT	27
5.2.1	Diskretní Fourierova transformace	27
5.2.2	Rychlá Fourierova transformace	28
5.2.3	Cooley-Tukey algoritmus	29
5.2.4	Výpočet FFT s pevnou čárkou	29
5.3	Digitální filtry	29
5.4	Goertzel algoritmus	31
5.5	Volba časového okna	32
5.6	Metoda DTW	32
6	Realizace algoritmu	34
6.1	Trigger	34
6.2	Rozpoznání rozbití skla	34
6.2.1	Klepnutí	34
6.2.2	Střepiny	36
6.3	Aplikace FIR filtru	39
6.4	Shrnutí algoritmu	39

7 Testování systému	41
7.1 O testu	41
7.2 Určování podmínek	41
7.3 Zpracování výsledků	42
8 Závěr	45
Literatura	48
9 Seznam použitých zkratk a symbolů	49
10 Seznam příloh	51
A Kód programy	52
A.1 Zdrojový kód algoritmu detekcí zvuku rozbití skla	52
B Analýza v programu Matlab	59
B.1 Zdrojový kód DTW algoritmu	59
C Obsah přiloženého CD	61

1. Úvod

Dneska se setkáváme s různými úlohami v automatizaci, někdy je potřeba jakýkoliv proces automatizovat, aby se usnadnila práce člověka a zvýšila se efektivita a přesnost toho procesu. Slyšíme o inteligentních budovách, moderních prostředcích v průmyslu a dalších zajímavých věcí. Jednou z nich je signalizace, která informuje člověka o jakémkoliv poruše v procesu nebo varuje ho o nebezpečí. Například člověku okamžitě přichází signál o faktu rozbití skla, což snižuje riziko pro sebe a svůj majetek. Jiný příklad když člověk dostává signál o napadení ptáků na zahradu. Pak se spouští odstrašovač ptáků. To pomůže ušetřit úrodu. Jelikož mechanismus se spouští jen v okamžiky aktivací signalizací šetří se energie. Je možné najít jinou aplikaci, kde člověk může být okamžitě informován.

V rámci potřeb dnešní automatizací systémy detekce poruchy mají velký význam v průmyslu a domácnosti. Proto v této bakalářské práci se budu zabývat rozvojem podobného systému. V mém případě systém bude využívat akustický jev. Jedná se o detekci zvuku, která spočívá v tom, že systém bude schopen rozpoznávat určité zvuky. Konkrétně vyhodnocovat přicházející zvukový signál pomocí algoritmu, který je uložen v paměti systému. Předpokládá se použití několika různých zvuků. Za složitý zvuk se bere zvuk rozbití skla. Zpravidla tento zvuk signalizuje o škodě nebo nebezpečí v našem životě. Což dává smysl takovou veličinu detekovat. Kromě složitého zvuku bude používán jednoduchý zvuk (křik ptáků).

Daná práce se věnuje vytvoření systému z hlediska hardwaru a softwaru. V kapitolách 2 a 3 se provede rozbor a výběr potřebných součástí systému. V kapitole 4 se věnuje analýze vybraných zvuků. V kapitole 5 se rozebírají jednotlivé metody pro vytvoření algoritmu. V kapitole 6 na základě provedeného rozboru se provádí realizace algoritmu. V kapitole 7 se testuje celý systém. V poslední kapitole se vyhodnocují dosažené výsledky prací.

Práce je psána v češtině, není mateřským jazykem pro autora, ale při všech obtížnostech chyby budou minimalizované.

2. Rozbor

Po seznámení se spoustou systémů detekcí určitých zvuků jsem přišel k závěru že, velkou část z nich tvoří systémy pro detekci rozbití skla, takzvané breaking glass alarmy. Pro detekci zcela zvláštních zvuků jakými jsou křiky ptáků nebo opakující zvuky existují zvláštní k tomu systémy. Přehled podobných systémů je uveden následně.

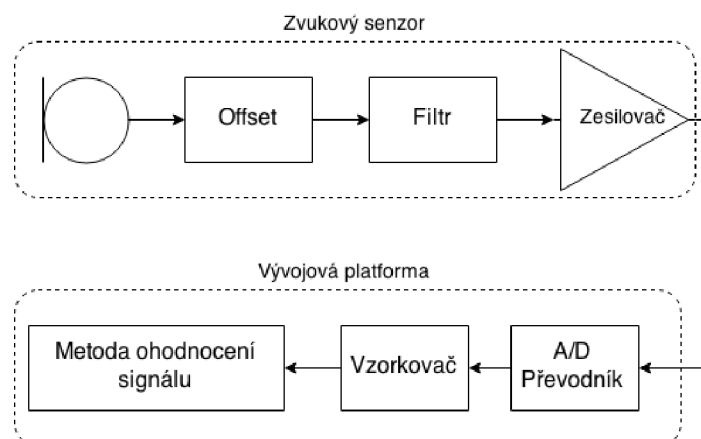
Podobné práce:

- Jednoduchý detektor rozbití skla založený na mikrokontroléru od Texas Instruments
<http://www.ti.com/lit/an/slaa351/slaa351.pdf>
- Robustní detektor rozbití skla založený na mikrokontroléru MSP430 od Texas Instruments
<http://www.ti.com/lit/an/slaa389/slaa389.pdf>
- Popis akustických vlastností signalizačních zvuků a experimenty pro jejich detekci v DETECTING ALARM SOUNDS
<http://www.ee.columbia.edu/~dpwe/pubs/crac01-alarms.pdf>
- Identifikace zvuku v Sound Object Localization and Retrieval in Complex Audio Environments
http://web.engr.illinois.edu/~dhoiem/publications/ICASSP2005_Hoie.pdf

Základem detekcí je měřicí řetězec, vstupem kterého bude akustická vlna přeměněná na digitální podobu pro další vyhodnocení. Akustický signál dá se převést na signál elektrický pomocí mikrofону. Tato část převodu by měla být co nejkvalitnější, aby chyby vzniklé v analogové části řetězce by se dalo lehce upravit v digitální. Požadavkami na mikrofon budou velký pracovní rozsah frekvence a dobrá citlivost postačující na široké spektrum zvuku rozbití skla a přesné reagování na definované zvuky. U snímaného mikrofónem signálu bude potřeba upravit offset a dále odfiltrovat šumy především použitím filtru horní propust. Malý elektrický signál je potřeba zesílit. K tomu se hodí operační zesilovač který by garantoval vhodný k dalšímu zpracování výstupní signál. Tento signál budeme vzorkovat s vhodnou periodou vzorkování. V té době bezprostředně bude záležet na metodě detekcí zvuku. Aby bylo možné mít kvalitní měření, tak je potřeba zvětšit počet vzorků a následně brát více odpočtů pro zjištění spektra. Tím se zvyšují nároky na digitální část. Bude potřeba hledat vhodný mikroprocesor s docela velkou pamětí pro tyto operace. Mezi vzorkováním a bezprostředně vyhodnocováním signálu by šlo používat digitální filtr, záleží však na tom, jaký typ zesilovače bude použit a jak velké zesílení bude mít. U vybraného mikrokontroléru pak bude potřeba testovat parametry: čas zápisu jednoho vzorku a další potřebné operace.

Vstupní částí toho řetězce je mikrofon. Tato část je nejdůležitější kvůli dalšímu zpracování signálu, chyby které vzniknou na vstupu, pak se projeví na výstupu řetězce. O podrobnostech a výběru vhodného typu mikrofónu zmíním v kapitole 3 mikrofóny.

Pro detekci zvukového signálu je potřeba najít vhodný mikrofon k převodu signálu akustického na digitální záznam hodnot. Z důvodu toho, že vyhodnocení analogového signálu je velice těžké a nevypočteně pro digitální techniku bude použit daný převod pomocí vybraného mikrofónu. Pro digitální vyhodnocování signálu existuje hodně nástrojů pro analýzu (viz kapitola 4).



Obrázek 2.1: Bloková schéma systému pro detekci zvuku

2.1. Knihovna vzorků

V úvodu bylo zmíněno o volbě jednoduchého a složitějšího druhu zvuku. Za jednoduchý se vybírá zvuk křiku ptáku. Za složitý se vybírá zvuk rozbití skla. Při této volbě se uvažovala dostupnost daných zvuků.

Zvuků křiku ptáků je velké množství v přírodě a svou volbu zastavím na těch, které jsou v blízké dostupnosti. Pro analýzu jsou vybrány zvuky havranů, sojek a kukaček. Zvuky křiků havranů a sojek jsou podobné (patří do jedné řady pěvců) a proto byly vybrány. Zvuk křiku kukaček je vybrán jako více odlišitelný zvuk od dvou ostatních. Celkem je nasbíráno 24 vzorků křiků havranů, sojek a kukaček.

Pro analýzu zvuku rozbití skla je nasbíráno kolem 20 zvuků rozbití skla, ze kterých většina tvoří zvuky rozbití okenního skla. Patří sem zvuky rozbití lahvi, nádobí, brýle, malých zrcadel. Všechny uvedené zvuky jsou zapsané a použitelné v analýze.

Všechny výše uvedené vzorky jsou v příloze k dané práci ve formě zápisů na CD.

3. Výběr zařízení

3.1. Mikrokontrolér [2]

V měřicím řetězce je potřeba převodu detekovaného signálu na diskretní. Číslicový signál pak je připraven k ohodnocení mikrokontrolérem. Srovnání vzorků a použití vyhodnocovacích metod si vyžaduje paměť. Z těchto požadavků je vybrána vývojová deska FRDM-KL25Z od firmy freescale. A to navíc z důvodu dostupnosti HW a potřebných nástrojů k programování mikrokontroléru KL25Z128VLK4.

KL25Z MCU obsahuje 32-bit ARM Cortex-M0+ jádro, které schopno zvětšovat maximální operační frekvenci až do 48 MHz pro veškeré potřebné operace výpočtů. Jednotka tvořena jednotlivě cyklicky rychlými I/O porty. Paměť jednotky je 128 KB flash a 16 KB SRAM. Zařízení podporuje správu napájení a módy kontrolérů pro efektivnost využití energií.

Obsahuje oscilátor kompatibilní s třemi rozsahy vstupního krystalu nebo rezonanční frekvence: 32-40 kHz (low freq. mode), 3-8 MHz (high freq. mode, malý rozsah) a 8-32 MHz (high freq. mode, velký rozsah). Jednotka KL25Z128 taktována od 8 MHz krystalu.

Analogová periferie je tvořena 16-bit SAR ADC a 12-bit DAC převodníky s w/ DMA podporou. Pro efektivitu převodů signálů je používán vysokorychlostní komparátor. Ve svém případě je potřeba analogový signál mikrofonu převést pomocí 16-bitového A/D převodníku s postupnou aproximací (SAR). Výhodou u toho převodníku je dostatečně vysoké rozlišení, vysoká rychlost. Doba převodu je dána hodinovým knitočtovým násobeným počtem bitů. V tomto případě teoretická doba je 2 mikrosekundy, což je zcela krátký časový interval.

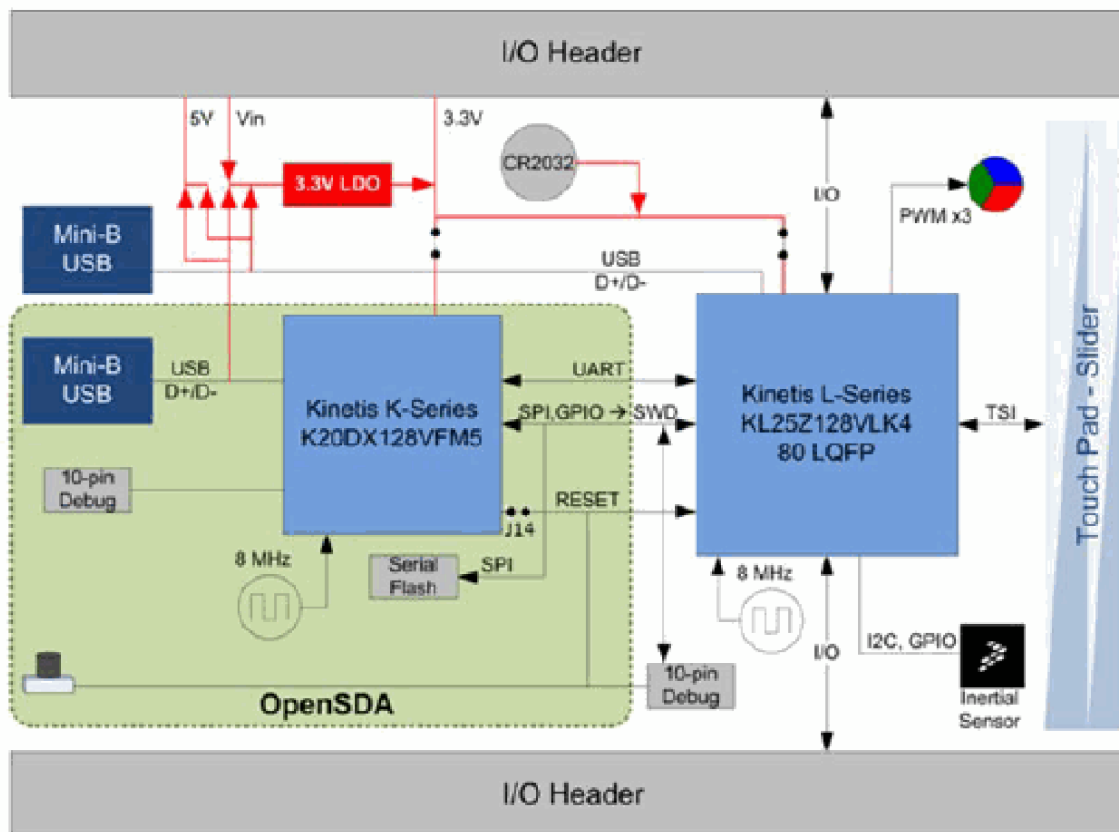
Komunikace desky s PC probíhá s použitím rychlého USB kontroléru. Napájení je flexibilní buď přes interface USB, baterií nebo vnějšími zdroje. Nominální napájení je 5 V. Deska je vybavena RGB LED diodou. Jednoduchý přístup k MCU I/O prostřednictvím Arduino R3 kompatibilními konektory. Požadované vstupy/výstupy pro danou aplikaci jsou uvedeny na obr. 3.1[2].

Na obr. 3.2 je uveden blok diagram nejdůležitějších jednotek tvořící vývojovou desku FRDM-KL25Z, který uvádí výrobce Freescale [2].

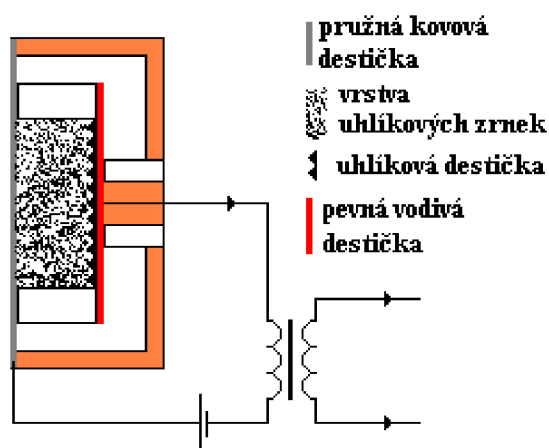
3.2. Mikrofony

Mikrofon je senzor tlaku přizpůsobený pro převod zvukové vlny v širokém spektrálním pásmu na elektrické kmity. Je první a důležitou částí v měřicím řetězci. Mikrofon je obvykle charakterizován svou citlivostí, směrovostí, frekvenčním pásmem, dynamickou vlastností, velikostí a cenou. Mikrofon se skládá z membrány a převodníku posunutí, který převádí odchylku membrány na signál elektrický. Zpravidla se mikrofony liší konstrukcí těchto dvou komponentů.

V následujících odstavcích je uveden přehled typů mikrofonů: jejich základní parametry, princip realizací, výhody a nevýhody použití.



Obrázek 3.2: FRDM-KL25Z blok diagram [2]

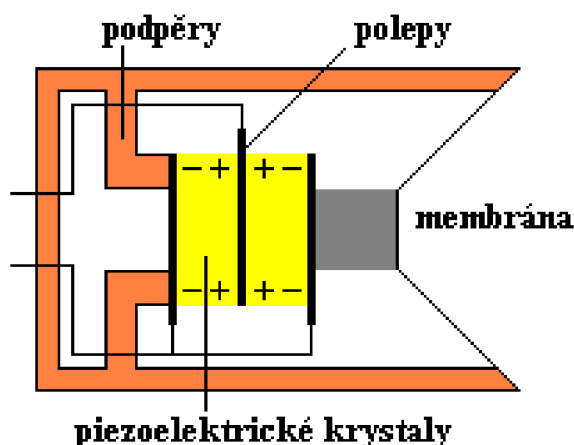


Obrázek 3.3: Schéma uhlíkového mikrofonu [4]

3.2.3. Dynamický mikrofon

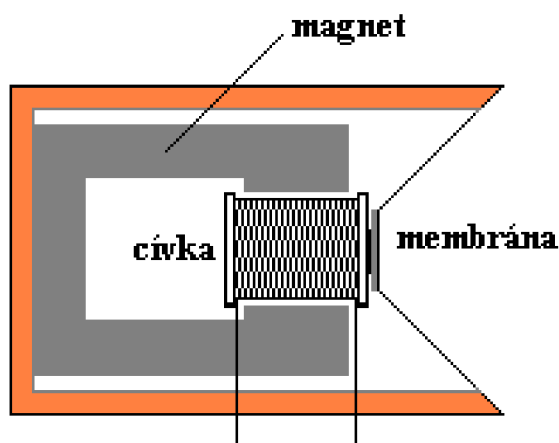
Principem dynamického mikrofonu je pohybující membrána cívkou v magnetickém poli, které je tvořeno permanentním magnetem. Membrána je vázána cívkou. Při pohybu se indukuje napětí, amplituda kterého je úměrná posunutí membrány. Viz obr. 3.5.

Ve srovnání s ostatními druhy mikrofonů je srovnatelně levný a zároveň spolehlivý. Může úspěšně pracovat v oblastech vysokého zvukového tlaku při relativně nízkém zkreslení. Nevýhodou je relativně malá citlivost.



Obrázek 3.4: Schéma piezoelektrického mikrofonu [4]

Zvláštním případem dynamického mikrofonu je mikrofon páskový. Membránou je kovový pásek umístěný v magnetickém poli permanentního magnetu. Ale i v takovém uspořádání vlastnosti přibližně odpovídají obyčejnému dynamickému mikrofonu.



Obrázek 3.5: Schéma dynamického mikrofonu [4]

3.2.4. Kondenzátorový mikrofon

Založen na principu změny kapacity kondenzátoru, který má velmi tenkou vodivou membránu a statickou vodivou destičku. Napětí mezi desky kondenzátoru se vypočítává podle vzorce:

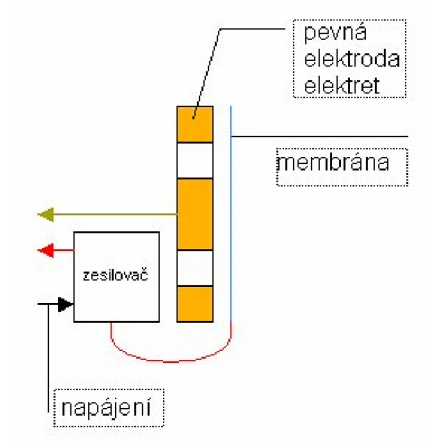
$$U = q * \frac{d}{\varepsilon_0 * \varepsilon_r * S} \quad [V] \quad (3.1)$$

,kde q - elektrický náboj na deskách kondenzátoru [C], d - vzdálenost mezi deskami [m], ε_0 - permitivita vakua [$F * m^{-1}$], ε_r - relativní permitivita prostředí mezi deskami kondenzátoru [-] (pro vzduch ~ 1), S - účinná plocha kondenzátoru [m^2].

Tím pádem kondenzátorový mikrofon převádí proměnlivou vzdálenost mezi membránou a statickou destičkou na signál elektrický. Elektrický signál se zesiluje dalšími elektrickými obvody.

Elektrétový mikrofon

Je zvláštním druhem kondenzátorového mikrofonu, který má podobný princip činnosti využití elektrického pole. Obsahuje elektret, který je dielektrickým materiálem krystalické struktury s pevnou elektrickou polarizací. Podle obr. 3.6 elektretový mikrofon se skládá z metalizované elektretové membrány a metalické desky, které jsou odděleny od sebe navzájem vzdušnou mezerou. Elektrodoou je snímáno napětí, které je výstupním signálem mikrofonu. Membrána mikrofonu dostává elektrický náboj během kmitů. Napájení se používá pro zesílení polarizací. Vstupní impedance daného druhu mikrofonu je vysoká. Zpravidla na vstupu mikrofonu je použit FET tranzistor, který spotřebuje malý proud a pracuje od nízkého zdroje napětí (až do 5 V). Elektretový mikrofon má řadu výhod. Je schopen pracovat v širokém frekvenčním pásmu. Má plošnou frekvenční charakteristiku (± 1 dB). Mikrofon je odolný vůči magnetickému poli, vibracím a má nízkou úroveň nelineárního zkreslení.



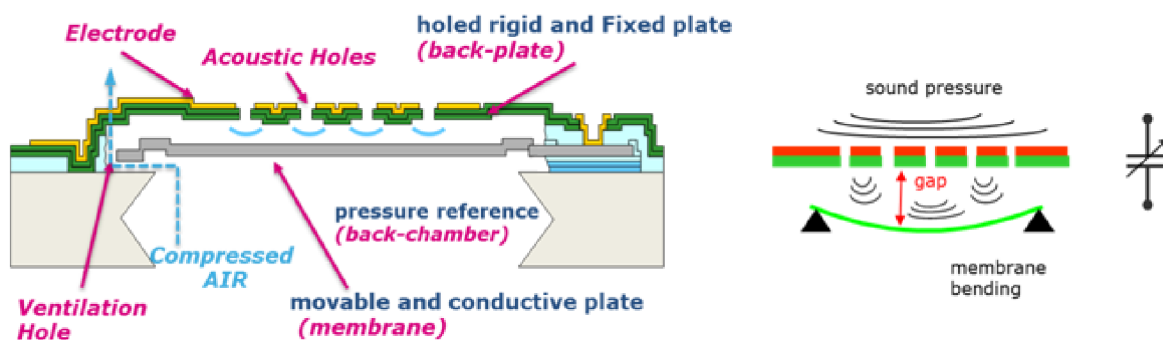
Obrázek 3.6: Schéma elektrétového mikrofonu [5]

3.2.5. MEMS mikrofon

Senzor využívá MEMS (Micro-Electrical-Mechanical-System) technologii a je v podstatě křemíkový kondenzátor. Kondenzátor tvoří dvě křemíkové plochy Obr. 3.7. Jedná z nich je fixovaná a druhá je pohyblivá. Pevný povrch je pokrytý elektrodou, tím je vodivý a plný akustických otvorů, které umožňují zvuku projít. Druhá deska je schopná pohybovat, protože je vázána pouze na jedné straně struktury. Větrací otvor umožňuje vyvádět ze zadní komory stlačený vzduch ven a tím se membrána vrací zpět. Komora umožňuje membráně pohybovat i uvnitř. V kombinaci s komorou vytvořenou balením má vliv na akustický výkon mikrofonu, případně na frekvenční odezvu. V důsledku změny kapacity (napětí) mezi dvěma povrchy je možné detekovat dopadající zvukovou vlnu. Obrovskou výhodou je malý rozměr. Nevýhodou je zkreslení kvůli vysoké amplitudě vstupního signálu.

3.2.6. Shrnutí výběru

Na základě provedeného rozboru mikrofonů je možné přijít k závěru že uhelní a piezoelektrický mikrofony stačit pro účel dané práci nebudou. Přestože oba mají větší citlivost než ostatní druhy mikrofonů. Nicméně kvůli úzkému pracovnímu frekvenčnímu pásmu a nevy-



Obrázek 3.7: Schéma MEMS mikrofonu [6],[7]
 Tabulka 3.1: Srovnávání typů mikrofonů

Typ	Pracovní frekvenční rozsah [Hz]	Nerovnost frekvenční charakteristiky [dB]	Citlivost na 1kHz [mV/Pa]	SNR [dB]
Uhelní	300-3400	20	1000	-
Piezoelektrický	100-5000	15	50	-
Dynamický	100-15000	12	3-5	-
Kondenzátorový (elektretový)	30-15000	5	1-10	60
MEMS	100-15000	-	4,2	63

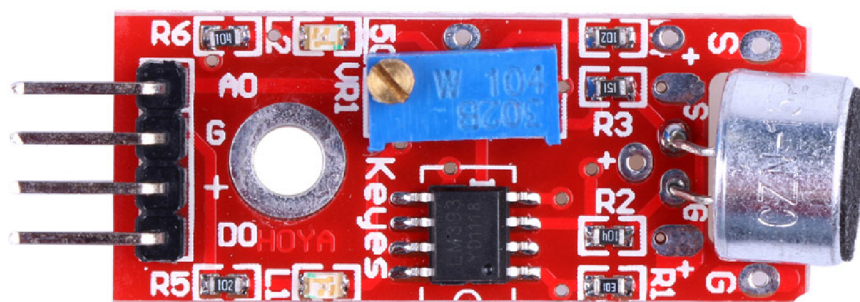
hovoujícím dynamickým vlastnostem dané druhy mikrofonů se uvažovat nebudou. Dobrou volbou pro detekci rozbití skla by mohl být dynamický mikrofon kvůli příjemnému pracovnímu frekvenčnímu rozsahu a schopnosti pracovat s vysokým tlakem zvukové vlny při nízkém zkreslení. Avšak nemá tak dobrou kvalitu zpracování nízkých zvuku jak mikrofony kondenzátorový a MEMS. Při srovnání kondenzátorového s MEMS mikrofonem je přednost dávana prvnímu kvůli lepšímu pracovnímu frekvenčnímu rozsahu a dostupnosti. Nicméně oba mají dobré dynamické vlastnosti, postačující citlivost a velký poměr signálu k šumu. Parametr velikost může vadit u některých aplikací, ale moc pro danou aplikaci nehrají roli. Kdyby bylo potřeba vybrat mikrofon podle velikosti, tak je lepší vybírat MEMS mikrofon. Tím pádem volbou je elektretový mikrofon, který nadále používán v hardwarové části dané prací. Viz tab. 3.1

3.3. Výběr modulu s mikrofonem

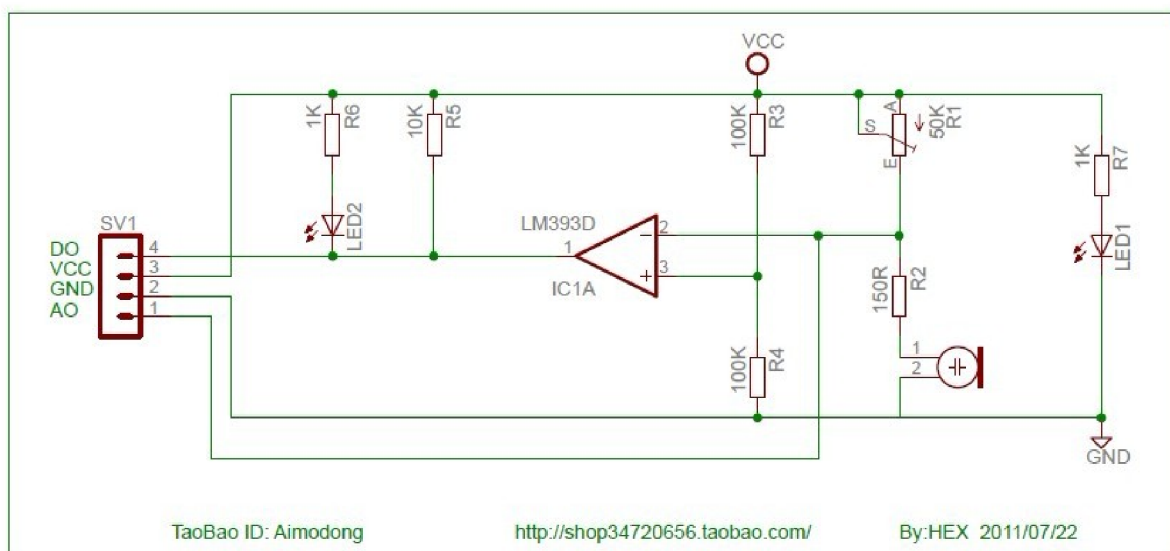
K vybranému mikrofonu z předchozí částí je potřeba navrhnout schémata zapojení. Výhodou elektretového mikrofonu je malý potřebný zdroj stejnosměrného napájení. Napájení až do 5 V může být zaručeno vývojovou deskou. Z komerčně dostupných a volně prodávaných modulů s mikrofonem byl vybrán modul s mikrofonem, který obsahuje elektretový mikrofon a potřebuje nízké napájení (5 V) a při tom má nevysokou cenu. Modul podle obr. 3.8 také obsahuje výstupy pro připojení k vývojové desce, elementy indikací, potenciometr pro kalibraci. Obsahuje jeden analogový AO a digitální DO výstupy, vstupy napájení + a G.

3.4. Připojení mikrofону

Na základě vybraného druhu mikrofónu je používán modul s mikrofónem a potřebným k němu zesilovačem pro zajištění dostatečnou úrovní výstupního signálu. Signál z modulu s mikrofónem se přenáší analogovým výstupem na vývojovou desku. Signál z modulu s mikrofónem se přenáší digitálním výstupem na vývojovou desku a působí jako trigger. Napájení a zem z vývojové desky jsou připojeny k příslušným vstupům modulu s mikrofónem. Trigger funguje, když intenzita zvuku dosahuje práhové hodnoty. Citlivost triggeru se nastavuje pomocí vestavného potenciometru. Napájení modulu s mikrofónem je indikováno LED L1 diodou. Dioda L2 indikuje o dosažení práhové hodnoty. Schéma daného modulu od výrobce je uvedeno na obr. 3.9.

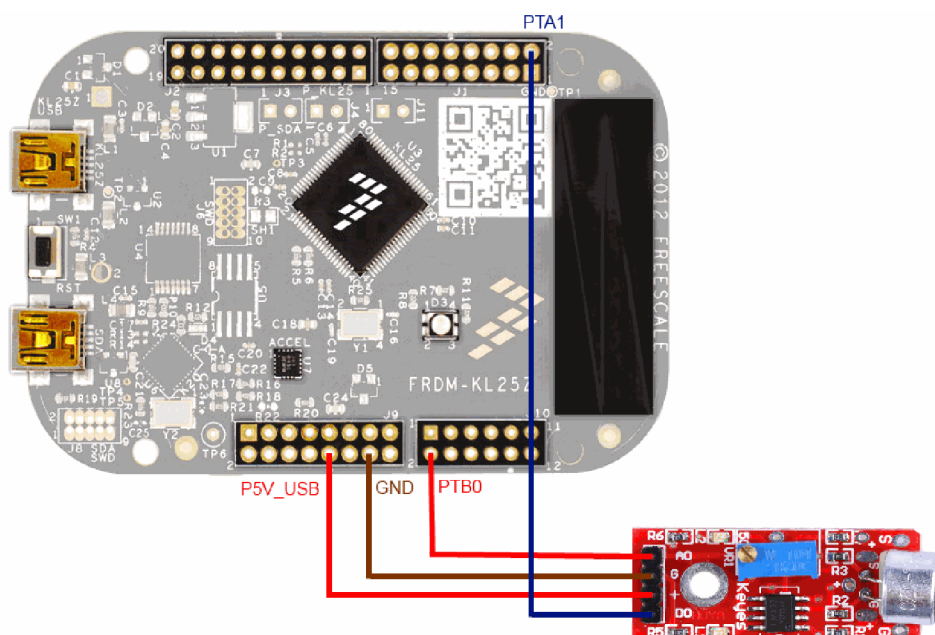


Obrázek 3.8: Keyses Microphone Sound Detection Sensor Module [10]



Obrázek 3.9: Scheme Keyses Microphone Sound Detection Sensor Module [11]

Připojení mikrofónu k vývojové desce FRDM-KL25Z je uvedeno na obr. 3.10. Navržené připojení může být odlišné pro jiné výstupy na FRDM-KL25Z (viz obr. 3.1).



Obrázek 3.10: Schema připojení microphone module k FRDM-KL25Z [2],[10]

4. Analýza

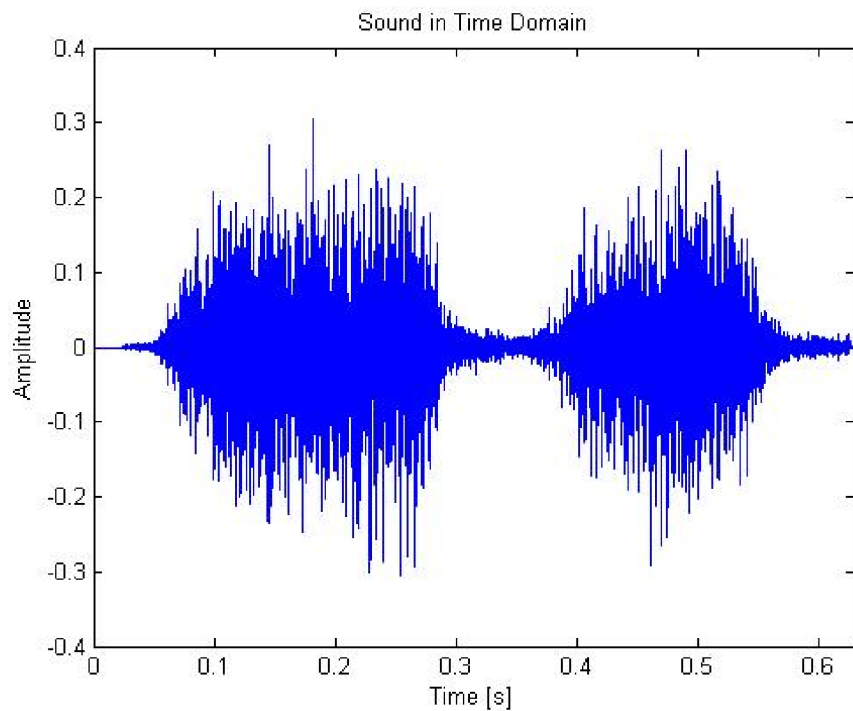
V této kapitole se bude řešit výroba systému pro detekci zvuku. Za vzorové rozpoznání byl vybrán křik ptáku, konkrétně křik sojky a složitější zvuk je zvuk rozbitého skla. O zvolených zvucích bylo řečeno v kapitole 2 rozbor.

4.1. Měření v matlabu

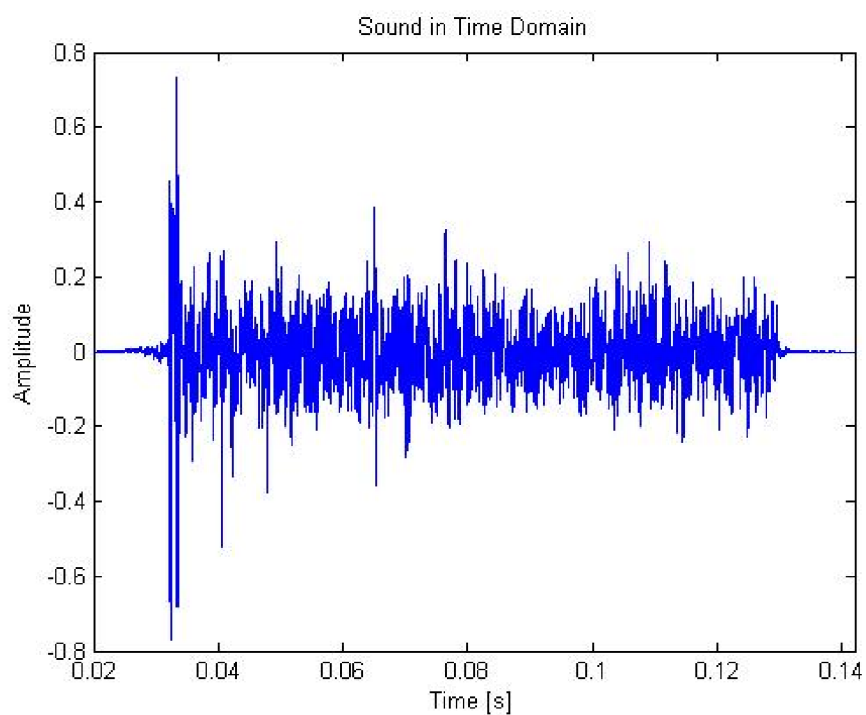
Za prvé byl změřen časový průběh obojích zvukových signálů v programu MATLAB. Používá se knihovna zvuků zvolená v kapitole 2 rozbor, která obsahuje sadu zvuků rozbití skla a křiků ptáků. Z časového průběhu křiku sojky je evidentně vidět podle obr.4.1, že jeden křik se skládá ze dvou, kde interval mezi křiky trvá přibližně 100 ms. Celkový dvojitý křik trvá poměrně dlouho, až do 600 ms. Vyhodnocování časového průběhu rozbití skla není tak jednoduché a proto byla zvolena myšlenka rozdělit daný signál po dvou částech. Ze celého signálu rozbití skla (obr. 4.3) je vidět začátek strmým skokem amplitudy čemuž dál bude říkat klepnutí o sklo. Klepnutí o sklo by mělo někdy skončit a z jednoho průběhu je zcela těžko poznat tento konec. Proto experimentálně průměrováním sady nasbíraných signálů bylo zjištěno, že klepnutí leží v časovém intervalu kolem 150 ms 4.2. Následující částí se bude dál říkat stěpiny skla a jejich úplný konec dopadu není možné zjistit kvůli různé výšce dopadu nebo odlišnosti situace.

4.2. Analýza pomocí FFT

Po provedení analýzy průběhů pomocí Fourierové transformací je možné zjistit, že výsledný spektrum rozbitého skla obsahuje široký rozsah frekvencí až do 20 kHz. Při dalším návrhu by bylo těžko hodnotit tak velký rozsah, jsou to velké nároky na mikroprocesor,

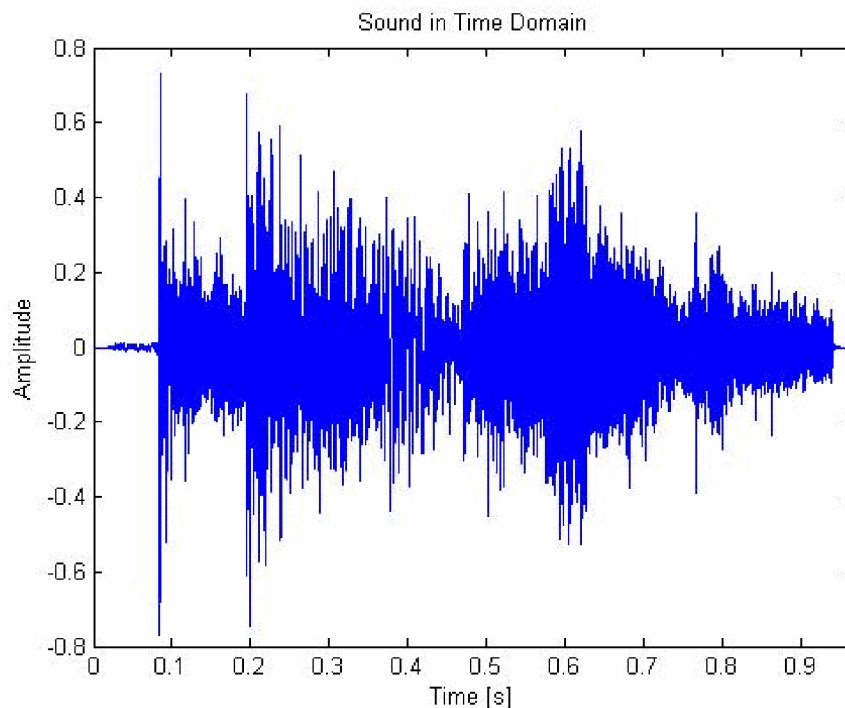


Obrázek 4.1: Časový průběh křiku sojky



Obrázek 4.2: Časový průběh klepnutí o sklo

proto byl zvolen menší rozsah 10 kHz. Ten rozsah by měl obsahovat význačné frekvence pro lepší hodnocení signálu. Důležitě si bylo uvědomit, že zvuk rozbitého skla se skládá ze dvou významných částí jak bylo řečeno předtím v 4.1 a to ze samotného klepnutí a



Obrázek 4.3: Časový průběh rozbití skla

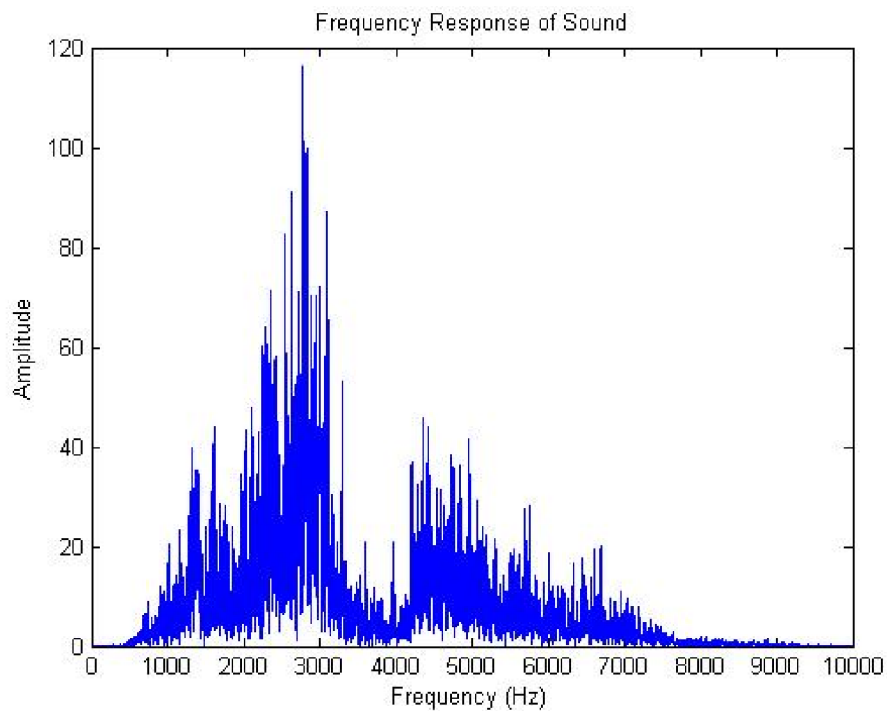
spadnutí střepin o zem. Klepnutí se projeví v rozsahu nízkých frekvencí Obr. 4.5. V tomto rozsahu spektra se často vyskytuje dominanta, která se vždy nachází kolem 300-400 Hz. Co se týká spadnutí střepin tak ty frekvence budou řadově vyšší, než u klepnutí. Aby bylo možné poznat které jsou to frekvence, používají se spektrogramy, o použití kterých se píše v další analýze.

Fourierové rozklad průběhu křiku sojky ukázal, že amplitudy spektra leží na frekvencích až do 4 kHz Obr. 4.4. Ze sady analýz různých křiků sojek vždycky se objevuje dominanta, která se nachází kolem 3 kHz, tím by se dalo velice zkrátit dobu rozpoznání křiku. Ale však by to nestačilo při hodnocení konkrétního druhu ptáku jakým je sojka. Z důvodu existenci velkého množství křiků ptáků, může nastat to, že by se našla stejná dominanta, ale u jiného druhu ptáku. Proto je potřeba najít jinou metodu analýzy.

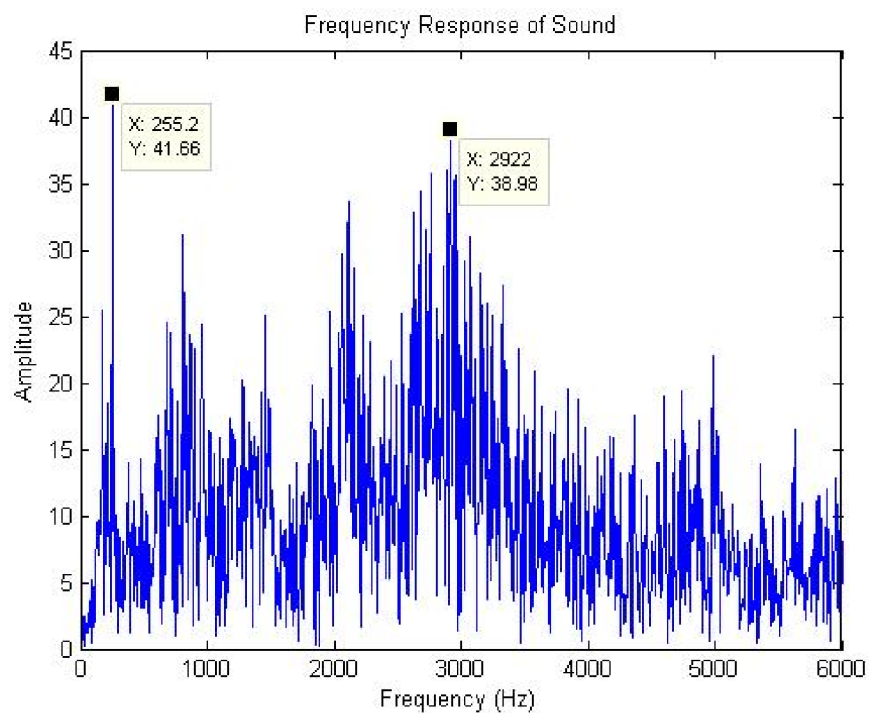
4.3. Analýza pomocí spektrogramu

Další metodou je zobrazení závislosti spektrální hustoty výkonu signálu na čase. Tímto způsobem je zobrazen signál rozbití skla, který má velké pásmo frekvence. Z grafu je okamžitě vidět hustoty na vyšších frekvencích Obr. 4.7. Hustota každé trvá určitou dobu a každý se nachází v různých místech spektrogramu. Po výkresu sady průběhů rozbitého skla jsem přišel k závěru, že jsou to střepiny, které padají po klepnutí o sklo. Od toho okamžiku dá se oříznout pásmo těchto frekvencí pro další analýzu rozbití skla podle spektrogramu. V úvahu jsou brány frekvence až do 10 kHz s kterými je možné pracovat dál.

Při výkresu spektrogramů sojky se podařilo vidět rovnoměrný rozklad hustot frekvence v okamžiky samotného křiku a pauzu mezi nimi. Viz Obr. 4.6. Přišel jsem k závěru, že bych zkrátit pásmo frekvencí až do 8 kHz, s uvažovanou dominantou ve 3 kHz, tím by se dalo dobře počítat při zpracování mikroprocesorem.



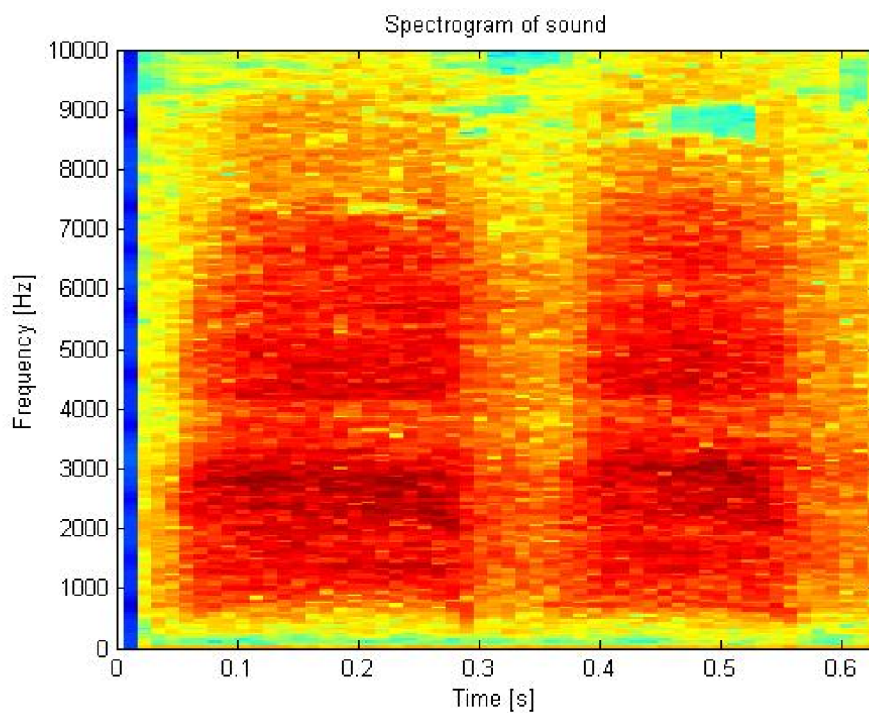
Obrázek 4.4: FFT křiku sojky



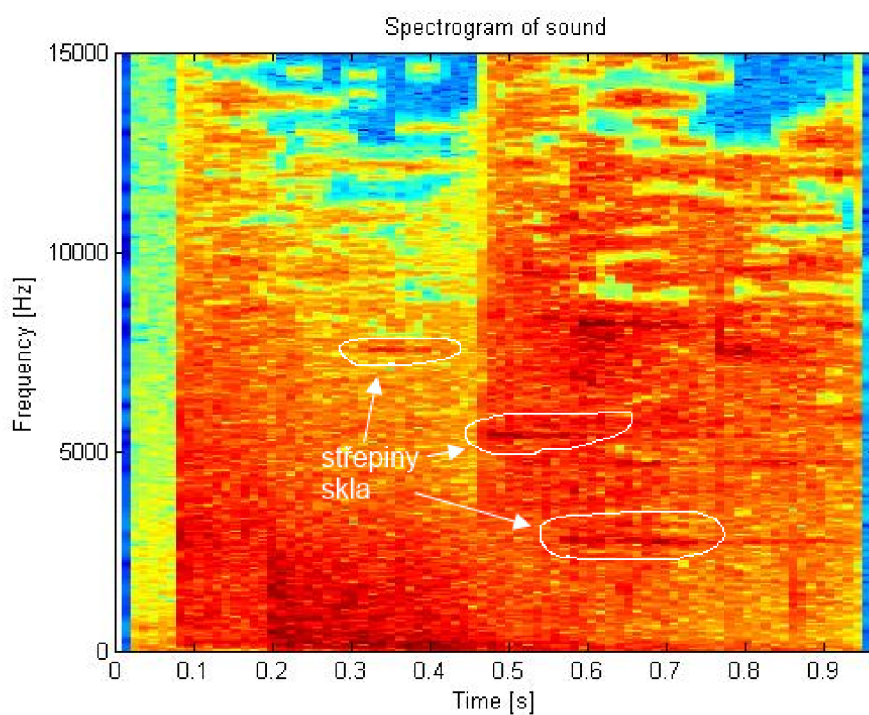
Obrázek 4.5: FFT rozbití skla

4.4. Metoda DTW

Metoda Dynamic Time Warping umožňuje srovnávat dva signály, které mohou mít různou časovou délkou. Dá se zjistit o kolik původní signál liší od druhého. Ve svém případě metoda



Obrázek 4.6: Spektrogram křiku sojky



Obrázek 4.7: Spektrogram rozbití skla

se aplikuje pro analýzu dvou signálů křiků ptáků. Každý průběh křiku je unikátní a to může platit jak pro křiky jednoho ptáku tak i pro křiky různých druhů ptáku.

Výsledkem metody DTW je Euklidova vzdálenost mezi danými časové průběhy, která říká o kolik se tento signál liší od původního. Hlavní příkon této metody je v tom, že při srovnání jej nevádí jestli původní křik byl rychlejší než druhý. Znamená to že je časově nezávislá.

V metodě DTW se používají spektrogramy obou signálů. Vektorem je brán časový interval signálu, který leží na zadané frekvenci. Porovnáním metodou DTW dvou vybraných vektorů pro stejné frekvence se zjistí jedno číslo, nazývané metrikou. Postup se opakuje pro všechny frekvence až do 8 kHz. Tak je získávána řada čísel. Pro získání výsledku aplikační metody DTW se sčítají jednotlivé prvky řady do jednoho čísla, které pak předvádí výslednou odlišnost dvou signálů.

Pro ověření funkčnosti metody Dynamického borcení času si zvolí tři skupiny křiků: skupina sojek, havranů a kukaček. V každé skupině je minimálně 7 vzorků jednoho druhu ptáku. Největší skupina sojek má 11 vzorků (viz kapitola č. Sada zvuků). Jenom skupina sojek má směs dvojitých a jednotlivých křiků. Ostatní skupiny mají průběh podobný mezi sebou. O podrobnostech vytvoření a výběru konkrétních vzorků je řečeno v kapitole 2 rozbor.

Test se provádí následujícím způsobem. Za prvé se srovnávají křiky mezi sebou v rámci jedné skupiny ptáku. Pak se srovnávají tři křiky z jedné skupiny se třemi křiky z jiné skupiny ptáků.

V první části testu bylo provedeno 66 srovnání ve skupině sojek, 55 srovnání ve skupině havranů a 21 srovnání ve skupině kukaček. Na základě získaných hodnot byly určeny maximální, minimální a průměrné metriky pro každou skupinu. Z tabulky č. 4.1 je možné přijít k závěru, že minimální metrika vyšla u skupiny sojek, maximální metrika u skupiny kukaček a největší průměr má skupina kukaček. Tím pádem průměrná metrika skupiny sojek a havranů je řadově menší než u skupiny kukaček, což znamená že nejvíce odlišují mezi sebou křiky ve skupině kukaček než v ostatních skupinách.

Tabulka 4.1: DTW: Porovnání vzorků mezi sebou ve skupině

	Min. metrika	Max. metrika	Průměr
Skupina sojek	0.0004926950	0.0045147307	0.00233165
Skupina havranů	0.0011780476	0.0045112447	0.002439135
Skupina kukaček	0.0081032426	0.0333751791	0.022136583

V další části testu se provádí srovnání vzorků mezi skupiny. Na základě zjištěných metrik z tabulek č. 4.2, 4.3, 4.4 je možné nalézt odlišnou hodnotu od ostatních metrik při srovnání vzorku 1 skupiny sojek s ostatními. To je způsobeno tím, že vzorek 1 skupiny sojek má dvojité křik.

Při určování a srovnání maximální, minimální a průměrné metriky mezi skupiny podle tabulky č. 4.5 je možné přijít k závěru, že nejmenší metrika je u skupiny sojek a havranů, maximální je u skupiny sojek a kukaček a největší výsledný průměr má skupina kukaček a havranů. Průměrná metrika při porovnávání skupiny sojek se skupinou havranů řadově menší, než při porovnávání skupiny sojek a havranů se skupinou kukaček. Podle průměru se dá říct, že nejvíce podobné jsou vzorky sojek a havranů oproti ostatním vzorkům.

Tabulka 4.2: DTW: Porovnání skupiny sojek se skupinou havranů

	Vzorek 1	Vzorek 2	Vzorek 3
Vzorek 1	0.0072706044	0.0027374822	0.0027251215
Vzorek 2	0.0048618573	0.0019933980	0.0019782640
Vzorek 3	0.0047873510	0.0024394912	0.0024335137

Tabulka 4.3: DTW: Porovnání skupiny sojek se skupinou kukaček

	Vzorek 1	Vzorek 2	Vzorek 3
Vzorek 1	0.0281505280	0.0186920915	0.0182445851
Vzorek 2	0.0273878417	0.0180718152	0.0180319810
Vzorek 3	0.0280369860	0.0184999439	0.0184909506

Tabulka 4.4: DTW: Porovnání skupiny kukaček se skupinou havranů

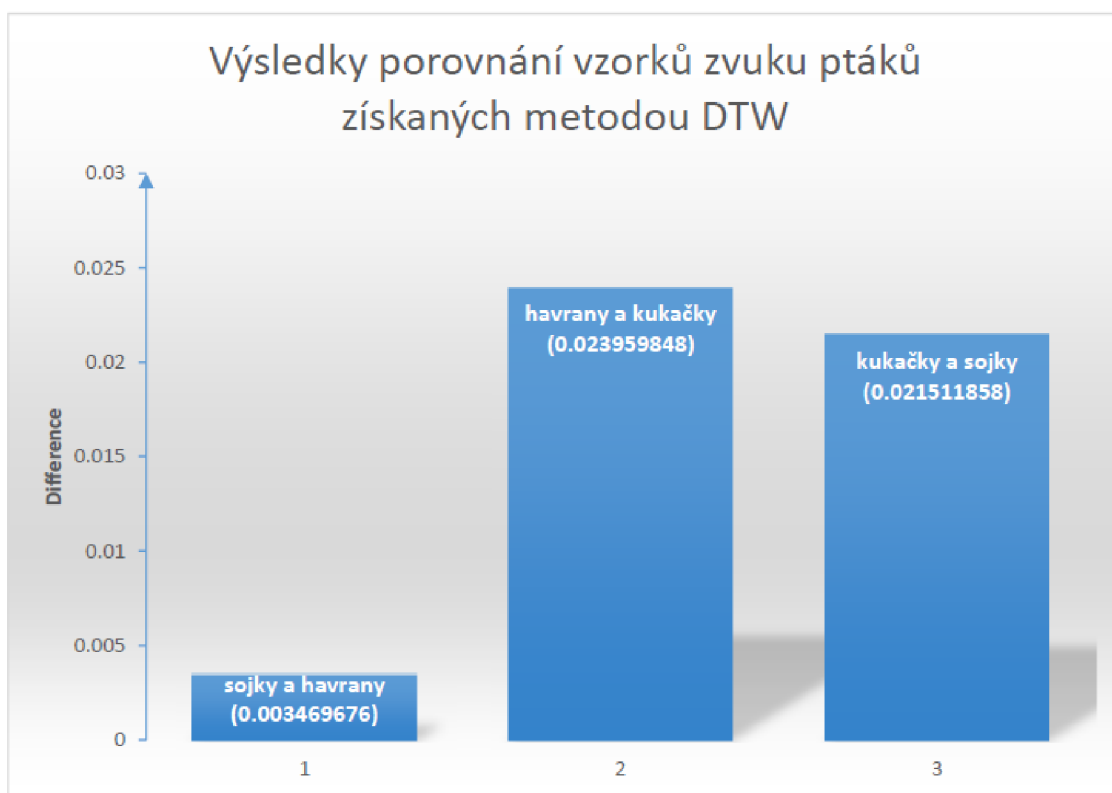
	Vzorek 1	Vzorek 2	Vzorek 3
Vzorek 1	0.0253761301	0.0246128677	0.0252623987
Vzorek 2	0.0239668379	0.0230731653	0.0232778098
Vzorek 3	0.0239093391	0.0229768255	0.0230932550

Tabulka 4.5: DTW: Porovnání vzorků mezi skupiny

	Min. metrika	Max. metrika	Průměr
Skupina sojek a havranů	0.0019782640	0.0072706044	0.003469676
Skupina sojek a kukaček	0.0180319810	0.0281505280	0.021511858
Skupina kukaček a havranů	0.0229768255	0.0253761301	0.023949848

Pro lepší představu získaných hodnot z tabulky č. 4.5 je uveden diagram na obr. 4.8. Za první sloupec je představen výsledek porovnání skupiny sojek se skupinou havranů. Za druhý a třetí sloupec jsou uvedeny výsledky porovnání skupin havranů a sojek se skupinou kukaček. Na vertikální ose je zobrazena šipka ve směru zesponu nahoru, která ukazuje výslednou diferenci dvou vzorků.

U skla metoda DTW se hodit nebude. Kvůli tomu, že není tady co řešit s rychlostí klepnutí o sklo. Jen by možná hodila při hodnocení dopadu střepin o zem, ale se zabývání analýzou střepin v rámci této bakalářské práci nebude, takže nebere se v úvahu střepiny, o tom jsem si zmínil v předchozí podkapitole Analýza pomocí spektrogramu.



Obrázek 4.8: DTW: Výsledky porovnání zvuků křiku ptáků mezi skupiny

4.5. Shrnutí

V dané kapitole byla provedena analýza zvolených podle kapitoly 2 druhů zvuků (zvuk rozbití skla a křik ptáku) pomocí metod uvedených v zadání bakalářské práce a jiných způsobů probraných z odborné literatury. Vytvoření potřebných skriptů a použití funkce pro analýzu se probíhalo v programu MATLAB.

První a základní analýzou bylo určování časových intervalů obou signálů. Charakterním příznakem zvuku rozbití skla stal strmý skok na začátku oproti křiku ptáku, který byl dvojitý a měl řadově stejný časový interval. U skla nebyl celý časový interval přesně stanoven, ale jenom začátek, který je pak nazván klepnutím o sklo.

Další analýzou bylo klasické zjišťování spekter obou signálů pomocí algoritmu rychlé Fourierovy transformací. Z výsledných spekter byl pak stanoven frekvenční rozsah pro oba signály a určeny největší spektrální složky.

Jinou formou představy bylo tady použití spektrogram, která je založena na Fourierově transformaci. Ze spektrogramu zvuku rozbití skla se dalo přesněji rozlišovat klepnutí a vysokofrekvenční složky. U křiku ptáku bylo jasně vidět rovnoměrné rozložení energií signálu.

Poslední částí bylo vyzkoušení metody Dynamického borcení času uvedeného v zadání bakalářské práce. Metoda se aplikovala s použitím spektrogram a jenom pro zvuk ptáků. Byl proveden test s porovnáním několika skupin zvuků ptáků. Zjistilo se, že zvuky skupiny ptáků se úspěšně podařilo rozlišovat mezi jednotlivými skupiny. Bohužel nejasnost použití metody DTW nastala při porovnání zvuků v rámci jedné skupiny. Proto danou metodu nemůžu považovat za správnou v dané situaci i když uvádí odpovídající realitě výsledky při porovnání mezi jednotlivé skupiny. Možná je potřeba danou metodu vyzkoušet na více vzorků s více druhy ptáků nebo používat jiné zvuky.

Tím pádem pro další realizaci systému pro detekci zvuku je vytvořen základ podle kterého nastal pokrok ve vývoje daného systému. Nadále se bude použítá metoda FFT.

5. Popis metod

V této kapitole jsou probírány specifické detaily návrhu. Řekne se o aplikaci matematického aparátu, používaných metodách, algoritmech a efektivitě jejich využití. Na základě čeho bude probíhat následující návrh z hlediska softwaru.

5.1. Použití spektrogram

V praxi se často používá postup odhadu spektrální hustoty nestacionárních signálů (jsou závislé na poloze počátku časové osy), který je založen na segmentaci signálu na krátké časové úseky, na kterých je prováděn odhad časově proměnného spektra nestacionárního signálu. Tento způsob je jednodušší, výpočetně méně náročný a nepotřebuje tak dlouhé výpočty jako FFT. Nevýhoda je v potřebnosti správného výběru délky segmentací. Když získáme krátké segmenty mají odhady velký rozptyl. Naopak když výsledné segmenty jsou příliš dlouhé informace bude jenom kvalitativního charakteru. Takže je potřeba dodržet proporce každého segmentu.

Výsledkem spektrální analýzy nestacionárních a lokálně stacionárních (lokálně nejsou závislé na poloze počátku časové osy) signálů jsou spektrogramy, často označované jako "hory" nebo "reliéfy", neboť jsou zobrazovány v trojrozměrném prostoru $(f, t, S(f))$ (frekvence, čas, spektrální hustota). Tento způsob zobrazení je velmi názorný, ale v mnoha případech může být velmi nečitelný. Proto je alternativně používáno dvourozměrné zobrazení pomocí hustotních spektrogramu, ve kterých je určité hodnotě spektrální hustoty přiřazen určitý odstín šedi. Výhoda zobrazení pomocí spektrogramu spočívá v tom, že je možné lehce vidět hustotu signálu o frekvenci v určitém čase.

5.2. Aplikace FFT

Fast Fourier transform je efektivní algoritmus pro spočtení diskretní Fourierovy transformace (DFT) a její inverze. Fourier analýza převádí časový prostor na frekvenční představu. FFT spočítá tyto převody faktorizací DFT maticí do násobku řídkých matic. Je používána v digitálním zpracování signálu.

5.2.1. Diskretní Fourierova transformace

V dnešní době v inženýrských aplikacích s rozvojem výpočetní techniky se často používá nástroj nazývaný diskretní Fourierovou transformací nebo zkrátka DFT. Je jednou z variant Fourierovy transformace a založen na Fourierově rozvoje. Výhodou je aplikace k diskretnímu signálu o konečný počet vzorků oproti obecné Fourierově transformaci, která vyžaduje celý analogový signál. Pro výpočet DFT 32 odpočtů je potřeba 1024 operace komplexního násobení a sčítání, což udělat ručně docela těžko. Příme DFT se popisuje:

$$S(k) = \sum_{n=0}^{N-1} s(n) * e^{-j\frac{2\pi}{N}nk}, \quad k = 0 \dots N - 1 \quad (5.1)$$

, kde je výsledkem přímého DFT spekter $S(k)$ pro N odpočtů při $k = 0 \dots N - 1$, obecně komplexní a je představen součtem $s(n)$ odpočty signálu $n = 0 \dots N - 1$.

Existuje ekvivalentní převod nazývaný zpětné DFT:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) * e^{j\frac{2\pi}{N}nk}, \quad n = 0 \dots N-1 \quad (5.2)$$

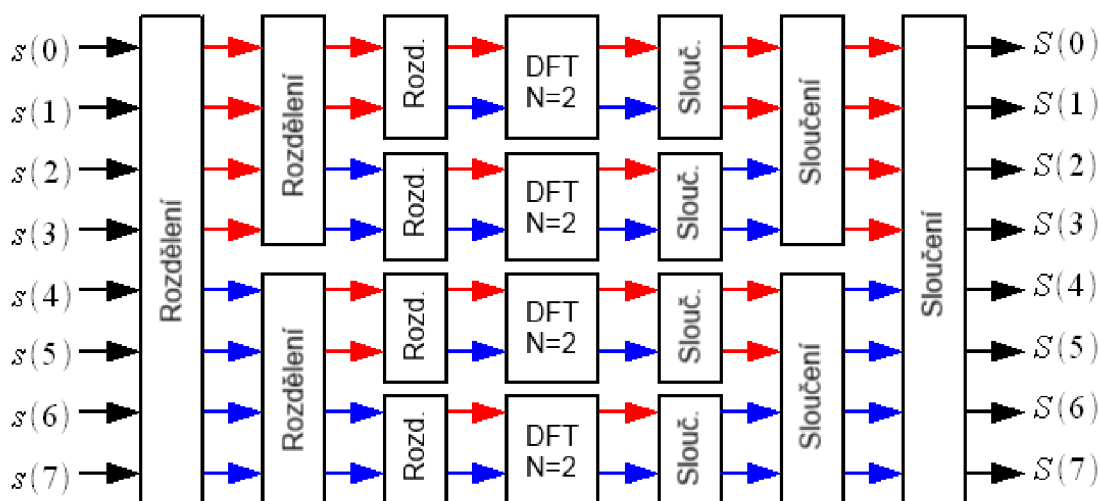
, kde je výsledkem zpětného DFT odpočty signálu $s(n)$ pro N odpočtů při $n = 0 \dots N-1$ a je představen součtem $S(k)$ spektru signálu $k = 0 \dots N-1$.

Koeficienty diskrétní Fourierovy transformace $S(k)$ při $k = 0 \dots N-1$ pro reálné vzorkované signály mají důležité vlastnosti:

- pro reálnou část koeficientů S_k platí $\Re(S_{N-k}) = \Re(S_k)$
- pro imaginární část koeficientů S_k platí $\Im(S_{N-k}) = -\Im(S_k)$
- hodnoty koeficientů S_k a S_{N-k} jsou komplexně sdružené, tj. $S_{N-k} = S_k^*$

5.2.2. Rychlá Fourierova transformace

Pro výpočet jednoho koeficientu DFT je nutné splnit N komplexních násobení a sčítání. Tím pádem výpočet celého DFT, který obsahuje N koeficientů, potřebuje N^2 pár operací. Počet operace se vzrůstá úměrně kvadrátu rozměrnosti DFT. Avšak když N není obecné číslo a může být rozloženo na činitele, tak se dá výpočetní proces zrychlit. Analyzovaný záznam se rozděluje po částech, vypočte jejich DFT a výsledky se sloučí (viz obr. 5.1 princip FFT). Takový způsob výpočtu DFT se nazývá rychlá Fourierova transformace nebo anglický Fast Fourier Transform (FFT).



Obrázek 5.1: Princip FFT

Při realizaci FFT je možné několik variant organizace výpočtů v závislosti od způsobu dělení posloupností odpočtů po částech. Vybírá se o kolik fragmentů rozbijí posloupnosti za každý krok (základ FFT).

V rámci své bakalářské práce výpočty spektra signálu se provádí pomocí FFT radix-2. Slovo radix označuje základ rychlé Fourierovy transformací. Další číslo je počet fragmentů, na které se rozbije signál na každé etapě probrání. V algoritmu radix-2 je rozměr analyzovaného záznamu musí být roven řádu dvojky, její dělení na půl se provádí až do získání dvou elementů posloupnosti. Výpočet jejich DFT nepotřebuje operace násobení - dva spektrální odpočty představují sebou sumu a rozdíl časových odpočtů.

5.2.3. Cooley-Tukey algoritmus

Nejpoužívanější způsob realizací FFT. Běžně použití tohoto algoritmu je rozdělením převodu po dvou rovných částech v každém kroku. Proto je obvykle ohraničen mocninou od dvou. Známé pod názvy radix-2 a mixem radixů. Patří tady i efektivní algoritmus split-radix. Hlavní myšlenkou je rekursivnost. Také dělí DFT po menších DFTs.

Existuje další množina používaných algoritmů Prime-faktor, Brunnův, Bluesteinův. Důležité se podívat na efektivitu těchto algoritmů.

5.2.4. Výpočet FFT s pevnou čárkou

Výpočet FFT lze provádět buď s pevnou nebo s pohyblivou řádovou čárkou. Oba způsoby mají své výhody a nevýhody dané rozdílnou délkou hodnoty. Při výpočtech FFT s běžnými 1024 odpočtů pro 1024 hodnot signálu se provede několik tisíc operace, kdy velkou roli bude hrát možnosti použité výpočetní jednotky. Dále se vysvětluje proč je dána přednost použití FFT s pevnou čárkou (integrity) pro daný mikrokontrolér KL25z.

Na první pokus jsem si snažil používat FFT funkci z mbed knihovny s parametry, které mají pohyblivou řádovou čárku (floaty). Výhodnost řešení je dosažení velkého dynamického rozsahu, to znamená značného poměru signálu k šumu. Výpočet je časově náročný, při operaci sčítání je nutné porovnávat velikosti obou elementů a normovat výsledky. Pro svou aplikaci v rámci bakalářské práci tak velký dynamický rozsah nehraje značnou roli, proto jsem přešel na funkci FFT s fixní čárkou. Také důvodem přejítí byla nemožnost provádět výpočty pro více než 512 odpočtů FFT kvůli omezené paměti mikrokontroléru.

Použití FFT s pevnou řádovou čárkou dalo větší rychlost výpočtu. Algoritmus pro výpočet motýlku je mnohem jednodušší než v případě pohyblivé řádové čárky, protože odpadne operace porovnávání a úpravy elementů. Samozřejmě je tady menší přesnost výpočtů než u pohyblivé čárky. Ale při zpracování signálu mikrofonem pracujícím v reálném čase přesnost tak velkou roli nehraje.

FFT potřebuje udělat několik tisíc operace aby transformovala signál o tisíc vzorků. Pro efektivitu vykonání FFT se proto radši používat celočíselné proměnné typů integer, než s čárkami. Při sbírání dat do celočíselného pole je potřeba hodnoty vynásobit aspoň tisícovkou, aby změna signálu z mikrofonu se projevila a pole nenaplnilo nuly.

5.3. Digitální filtry

Digitální filtrace signálů má velký význam v praxi. Velmi často se stává situace když je potřeba zbavit šumu, který je součástí detekovaného signálu. Obvykle zpracování signálů probíhá v podobě spektra a nastane to, že je nutné určit spektrum signálu jenom v určitých mezích. K tomu právě slouží digitální filtry. [15]

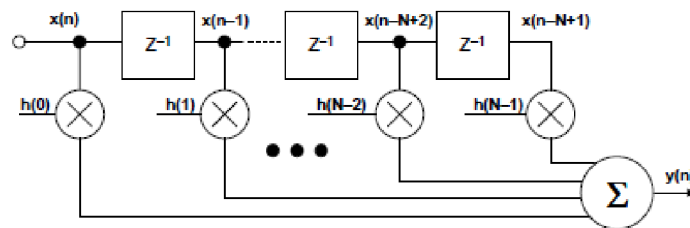
Filtry se popisují buď impulsní odezvou, odezvou na skok nebo frekvenční odezvou. Pro aplikaci v své bakalářské práci se bude věnovat frekvenční představě signálu po jeho filtraci. Existují 4 základní frekvenční odezvy podle propustností filtru: dolní propust LP, horní propust HP, pásmová propust BP a pásmová zádrž BR. V časové oblasti jsou důležité doba náběhu a překmit filtru. V praxi čím je kratší doba náběhu a menší překmit, tím je navržený filtr lepší.

Základně se filtry rozdělují podle způsobu tvorby na filtry s konečnou impulsní charakteristikou FIR a rekurzivní filtry IIR. Filtry s nekonečnou impulsní charakteristikou IIR kromě vstupního signálu se používají i výstupy zpracované v předchozím kroku.

Aplikace FIR filtru se popisuje rovnicí:

$$y[n] = h[k] * x[k] = \sum_{k=0}^{N-1} h[k]x[n - k] \quad (5.3)$$

kde $h[k]$ je řada koeficientů filtru a $x[k]$ je řada vstupních dat pro filtraci. Číslo N v rovnici představuje číslo větve filtru a vztahuje k výkonnosti filtru. FIR filtr s N větve vyžaduje N násobně akumulovaných cyklů. Grafická interpretace vzorce obecného FIR filtru je uvedena na obr. 5.2.



Obrázek 5.2: FIR filtr [15]

Druhým druhem je IIR filtr, výstupní odpočty kterého se popisují rovnicí:

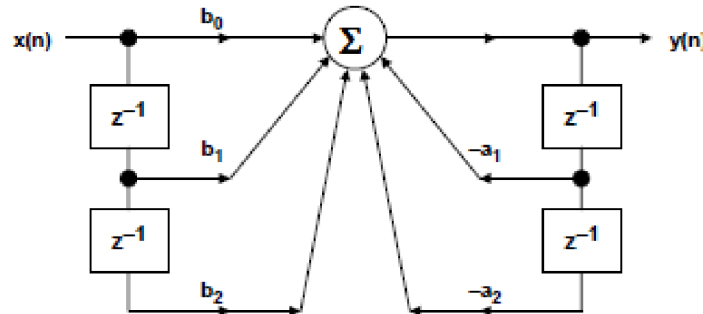
$$y[n] = \sum_{k=0}^M b_k x[n - k] + \sum_{k=1}^N a_k y[n - k] \quad (5.4)$$

Obvykle se představuje přenosem:

$$H[z] = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (5.5)$$

kde b_k jsou koeficienty přímé vazby nul, a_k jsou zpětnovazební koeficienty pólů. Index k odpovídá řadu filtru IIR.

Na obr. 5.3 je uvedena implementace filtru IIR druhého řadu.



Obrázek 5.3: HW realizace IIR filtru [15]

5.4. Goertzel algoritmus

Minulé se říkalo o diskrétní Fourierové transformaci na které je založen algoritmus rychlé Fourierovy transformací. Aby bylo možné zjistit spektrum signálu je nutné signál s určitou frekvencí vzorkovat. Potom ze znalosti počtů nasbíraných vzorků se vybíral počet odpočtů a algoritmus FFT vypočítal spektrum nadefinovaného záznamu. Jednotlivé složky pak leží s krokem $\Delta f = Fs/N$. Tím pádem je možné pracovat s celým spektrem signálu. Ale co když je potřeba určovat jenom několik hodnot spektra signálu. K tomu slouží nástroj Goertzel algoritmus, který je zvláštním případem realizací diskrétní Fourierové transformací ve formě rekurzivního filtru. Daný algoritmus nabídl Gerald Goertzel v roku 1958.

Výpočet k -tého DFT koeficientu probíhá následovně podle vzorce

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi k \frac{n}{N}}, k = 0, \dots, N - 1 \quad (5.6)$$

Možný převod na jiný tvar

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi k \frac{n-N}{N}} \quad (5.7)$$

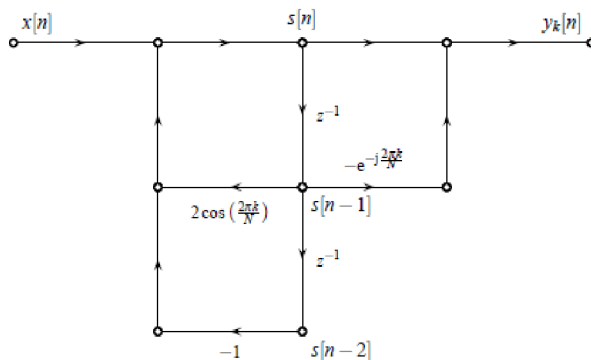
Hledaná hodnota může obsahovat výstupní vzorek v čase N IIR lineárního systému s impulzní odezvou. Přenos toho systému je odvozen ve Laplaceově transformaci a je druhého řádu.

$$H_k[k] = \frac{1 - e^{-j\frac{2\pi k}{N}} z^{-1}}{1 - 2 \cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}} \quad (5.8)$$

Daný přenos se popisuje s použitím stavových proměnných vzorcem

$$s[n] = x[n] + 2 \cos\left(\frac{2\pi k}{N}\right)s[n-1] - s[n-2] \quad (5.9)$$

Grafická interpretace stavového popisu je zobrazena na Obr. 5.4. Algoritmus iteruje reálná čísla podle předchozího vzorku $(N + 1)$ krát.



Obrázek 5.4: Signálový graf proudění Goertzel systému druhého řádu [24]

Značnou výhodou tohoto algoritmu oproti algoritmu FFT je menší počet prováděných operací pro výpočet DFT koeficientů. Například pro DTMF aplikaci je potřeba mít 8 ze

205 DFT koeficientů pro výpočet prvních harmonik osmi možných tonů. Když nejsou v první je možné pokračovat v počítání druhé atd.

Na rozdíl od FFT algoritmu nepotřebuje mít přesně definovanou délku signálu N a to nehraje roli na složitost počítání Goertzel algoritmu. Povoluje výpočet indexů k jak celočíselných tak s čarou. Zabere méně paměti při výpočtech než FFT.

5.5. Volba časového okna

Chyby v odhadu frekvence a amplitudy, které plynou z omezené doby záznamů detekovaných signálů, lze částečně kompenzovat vhodnou volbou časového okna. Existuje velké množství okenných funkcí. V praxi se nejčastěji používají časové obdélníkové okno, časové okna Hanning a Flat Top. Cílem použití oken je vyrovnání výsledného spektra po FFT transformací naměřeného signálu. Pak z vyrovnaného spektra by se dalo jednoduší zjistit jednotlivé složky signálu.

Důležitou vlastností efektivity okna je úroveň postranních laloků, když je příliš vysoká úroveň postranních laloků spektra v případě obdélníkového okna (-13dB), tak mohou být ztraceny sousední složky dominanty spektra. Důležité je také zajistit dynamický rozsah spektrální analýzy. Pro 16-bitový ADC převodník se to výpočte jako $D = 20 * \log_2^{16} = 96,32dB$. Na základě znalosti dynamického rozsahu detekovaného signálu a dynamického rozsahu převodníku se může být voleno určité okno. Vybrané okno je vhodné když jeho úroveň postranních laloků je menší než dynamický rozsah signálu. Pak se volí počet odpočtů FFT pro zajištění nutného frekvenčního rozlišení spektra vzhledem k frekvenci vzorkování a vlastnosti okenné funkce. Vybraná okenná funkce se vynásobí naměřenými vzorky, čímž pak zjištěné násobky můžou transformovaný pomocí FFT algoritmu.

Vzhledem k práci s zároveň úzkým a širokým spektrem signálu rozbití skla použijí časové okno Hanning. Nejvyšší postranní lalok má -31,5 dB, což je menší než u obdélníkového časového okna. Má největší pokles laloků 60 dB/dek, což zajistí lepší viditelnost výsledného spektra signálu.

5.6. Metoda DTW

Dynamic time warping (DTW) je dobrá technika pro nalezení optimální cesty mezi dvěma (časově - závislých) posloupnostmi pod určité rámce. Originálně, DTW byl použit pro porovnání různých řečových vzorů v automatickém rozpoznávání řeči. DTW úspěšně zpracovává časové deformace a různé rychlosti asociativních časově - závislých dat.

Objektem DTW je porovnání dvou (časově-závislých) sekvence $X := (x_1, x_2, \dots, x_i, \dots, x_{|X|})$ délky $|X|$ a $Y := (y_1, y_2, \dots, y_j, \dots, y_{|Y|})$ délky $|Y|$. Ty sekvence můžou být diskrétní signály (časově-závislých) nebo obecně vzorkované posloupnosti v ekvidistantních bodech v čase.

Dvě sekvence pak tvoří zkreslenou cestu W :

$$W = w_1, w_2, \dots, w_K \quad \max(|X|, |Y|) \leq K < |X| + |Y| \quad (5.10)$$

kde K je délka zkreslené cesty a k -tý element zkreslené cesty je:

$$w_k = (i, j) \quad (5.11)$$

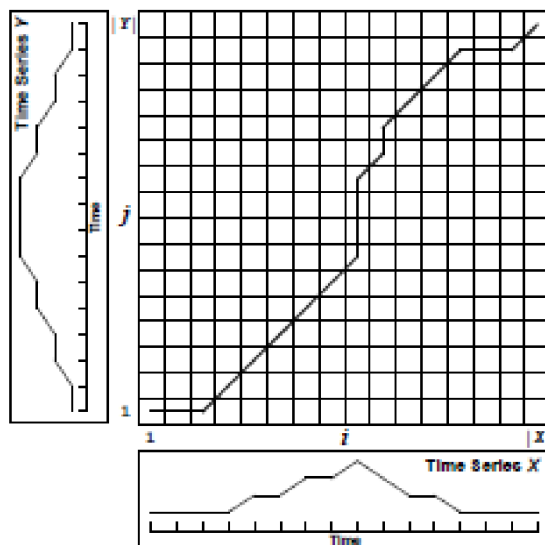
kde i je index od časové řady hodnot X a j je index od časové řady hodnot Y .

Optimální zkreslená cesta se vypočte jako:

$$Dist(W) = \sum_{k=1}^{k=K} Dist(w_{ki}, w_{kj}) \quad (5.12)$$

kde $Dist(W)$ je obyčejně Euklidova vzdálenost zkreslené cesty W a $Dist(w_{ki}, w_{kj})$ je vzdálenost dvou bodů řad hodnot s indexy (jeden od X a jeden od Y) v k -tém elementu zkreslené cesty. Grafická interpretace algoritmu DTW je uvedena na obr. 5.5.

Ve svém případě algoritmus DTW byl aplikován v kapitole 4 Analýza prostřednictvím programy Matlab při analýze dvou křiků ptáku ve formě vzorkovaných diskretních posloupností.



Obrázek 5.5: DTW pro dvě časové řady hodnot [13]

6. Realizace algoritmu

Na základě provedených měření z předchozí kapitoly 4 je tady věnováno samotnému vytvoření algoritmu pro detekci určitých zvuků.

6.1. Trigger

Při detekce jakéhokoliv zvuku je potřeba mít spouštěcí mechanismus celého algoritmu detekcí. A to nejen z důvodu potřebného zahájení algoritmu, ale také kvůli snížení požadavků na mikrokontrolér. Mikrokontrolér by neměl provádět žádné operace počítání v dobu ticha. V mém případě to bude zajištěno pomocí digitálního výstupu z modulu mikrofonu. Když nastane jakýkoliv šum výše nastavené potenciometrem úrovní, tak se trigger přepne. Z digitální log. 0 se přejde do log. 1, čímž bude zahájen algoritmus detekcí. Také je potřeba dávat pozor na napájení modulu. Když je na vstupu 5 V nebo 3,3 V hranice triggeru se bude lišit a je nutné nastavovat trigger znova. Tím pádem kolísání napájení značně ovlivňuje funkčnost triggeru a přesnost celého algoritmu detekce.

6.2. Rozpoznání rozbití skla

Z časového průběhu byl zjištěn velký skok na začátku a s rostoucím časem amplituda klesla. Po Fourierové transformaci tohoto intervalu se objevili dominanty v rozsahu 200-400 Hz. Jedná se o klepnutí o sklo. Podobný zvuk je zcela častý ve světě, proto dostatečným pro detekci nebude. Avšak použití ho v kombinaci s nalezením vysokofrekvenční složky garantuje správnou detekci. Ohodnocení frekvence až do 4 kHz proběhne až po nalezeném klepnutí. Po určitém počtu zpracovaných cyklů se vrací do počátečního stavu.

Důležité je uvědomit na to, že samotné klepnutí trvá vždy poměrně krátký čas kolem 200 ms (kapitola Analýza 4 4.1). Ostatní část signálu tvoří vysokofrekvenční složky a čas dopadu střepin je tedy neomezen. Uvažovat budu čas celého signálu rozbití skla jen kolem 1 s, čímž odpadne nejasnost v čase a zjednoduší realizace algoritmu detekce.

6.2.1. Klepnutí

Nejvíce efektivním a rychlým způsobem rozpoznávání zvuku rozbití skla je nalezení charakterního příznaku klepnutí o sklo. Pro určování klepnutí je potřeba nalézt dominanty v intervalu 200-400 Hz. K tomu se aplikuje funkce FFT s fixní čarou s určitou délkou odpočtů. Výběr délky odpočtů se určuje délkou záznamu zapsaných hodnot z mikrofonu. Také důležitým kritériem výběru je výpočet frekvence vzorkování. Podle výše uvedeného intervalu 200-400 Hz je potřeba vybrat frekvenci vzorkování aspoň 1 kHz (podle vzorkovacího teorému). Frekvence vzorkování je vypočítána podle vzorce

$$Fs = N/t \tag{6.1}$$

kde N - délka záznamu zapsaných hodnot z mikrofonu, t - čas zápisu toho záznamu. Ze znalosti toho, že časová interval klepnutí o sklo je kolem 200 ms (viz 4.1) a délka záznamu hodnot souvisí s délkou odpočtů FFT dá se najít odpovídající poměr ve vzorci, aby byly

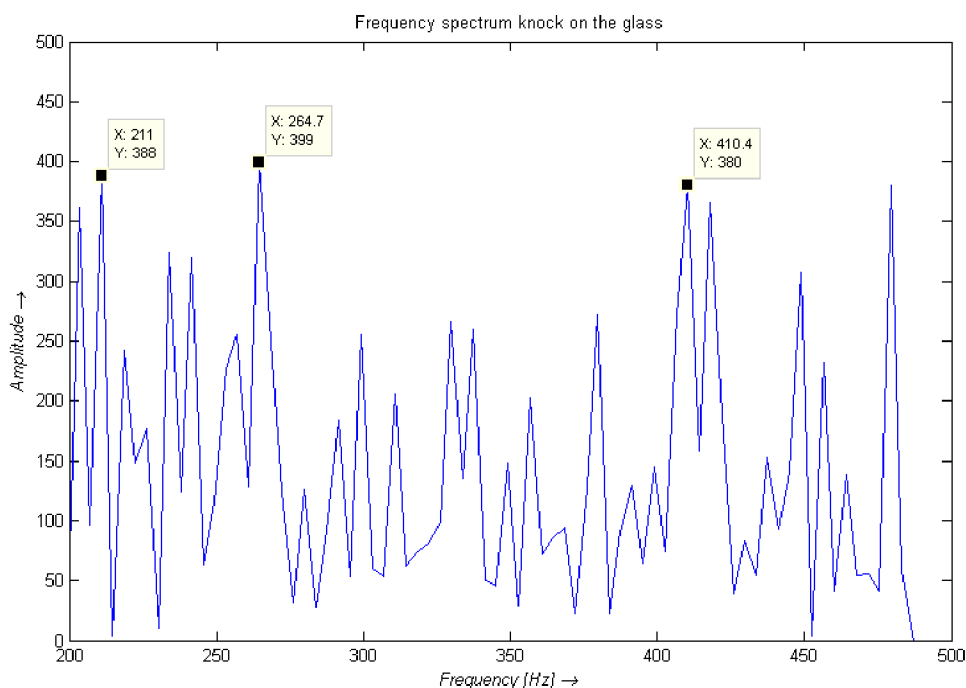
splněny všechny podmínky výpočtu. Tím pádem délka záznamu N je 256, čas zápisu toho záznamu t je $\tilde{0}.26$ s. Po dosažení do vzorce vyplývá požadovaná frekvence vzorkování $\tilde{1}$ kHz. Když délka záznamu N je 256, tak maximum možných 256 odpočtů FFT dá v tomto případě nejjemnější možný krok frekvenční osy. Frekvenční krok se vypočítá podle vzorce

$$\Delta f = F_s / FFTSIZE \quad (6.2)$$

kde F_s - frekvence vzorkování, $FFTSIZE$ - počet odpočtů FFT. Z toho vyplývá, že při frekvenci vzorkování $\tilde{1}$ kHz a 256 odpočtů FFT frekvenční krok Δf je $\tilde{3}.9$ Hz. Samozřejmě se dá dosáhnout jemnějšího kroku než $\tilde{3}.9$ Hz, ale omezovacím faktorem tady posloužilo splnění všech podmínek výpočtu dané úlohy. Kromě toho značnou limitou stalo použití vybrané FFT funkce, založenou na radix-2 algoritmu, která povoluje vybírat jenom 2^N odpočtů (resp. $2^8 = 256$).

Na základě provedené analýzy zvuku klepnutí o sklo (viz. 4.2) byly určeny tři podmínky jeho rozpoznávání. Za současné splnění všech těchto podmínek následuje to, že jedná se o klepnutí a algoritmus může pokračovat k další detekci vysokofrekvenčních složek.

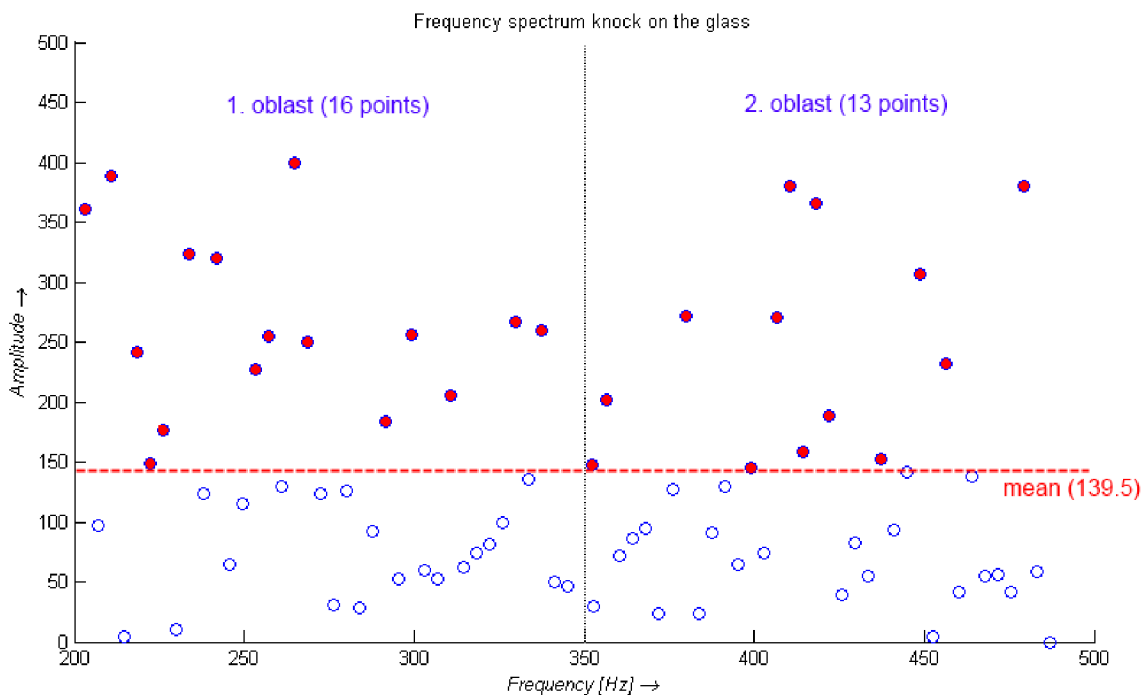
První podmínkou je nalezení třech největších složek spektra v intervalu 200-350 Hz. Když je alespoň jedná ze třech těchto složek leží v tom intervalu, tak podmínka se počítá za splněnou. Hledání třech největších složek spektra se podařilo realizovat pomocí jednoho while cyklu, což tři krát zrychluje algoritmus pro výpočet první podmínky, než při hledání maxima postupně za pomocí třech cyklů. Na následujícím obrázku (Obr. 6.1) je ukázka splnění této podmínky.



Obrázek 6.1: Spektrum klepnutí o sklo

Druhá podmínka vychází z rozložení energií signálu v intervalu 200-480 Hz. Interval je definován dvěma oblastmi v 200-350 Hz a 350-480 Hz. Za energii je tady považovány

hodnoty, které leží nad průměrnou úrovní spektra celého intervalu 200-480 Hz. Tím pádem se vypočte počet hodnot spektra v každé oblasti frekvenčně a amplitudově omezených a z toho přijde rozhodnutí. Když je v oblasti 200-350 Hz vypočteno více hodnot než v oblasti 350-480 Hz z toho následuje, že druhá podmínka je splněna. Níže je uveden názorný příklad (Obr. 6.2) kdy je druhá podmínka splněna. Pro lepší viditelnost je spektrum představeno diskrétními body pomocí matlab funkce scatter.



Obrázek 6.2: Spektrum klepnutí o sklo

Třetí podmínka garantuje, že detekovaný signál klepnutí o sklo má dostatečně vysokou úroveň pro detekci. Je zavedena proto, aby poměr užitečného signálu k šumu byl nad určitou úrovní. Je definována pomocí poměru největší spektrální složky k průměru celého intervalu spektra. Když je tento průměr větší než 3, tím je podmínka splněna.

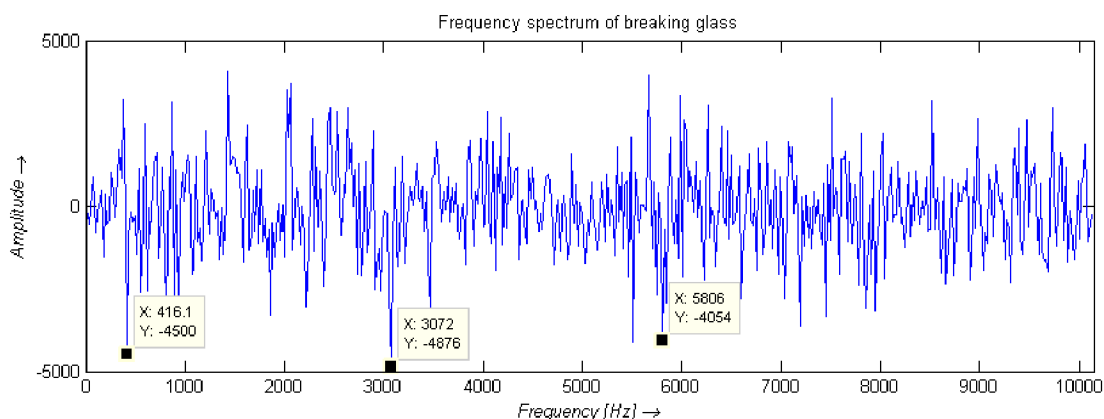
6.2.2. Střepiny

Po nalezení klepnutí algoritmus přejde do stavu střepiny. Aby bylo možné hledat spektrální složky na vyšších frekvencích zase se aplikuje FFT. S tím rozdílem, že detekce až do 10 kHz vyžaduje frekvenci vzorkování aspoň 20 kHz. Tím pádem čas jednoho zápisu záznamu střepin při všech splněných podmínkách (viz 6.2) trvá kolem 50 ms pro 1024 vzorků. Tak krátké nasbírané v čase intervaly je těžko ohodnocovat kvůli hodně odlišným průběhům každého. To vzniká z důvodu příliš krátkého zápisu záznamu, které nepokrývá celý interval zvuku rozbití skla (viz kapitola 4 analýza). Druhou příčinou je malý počet sbíraných vzorků, který následně po FFT transformaci pro 1024 odpočtů dá frekvenční krok $\Delta f = 20Hz$. Přesnost ohodnocování v tomto případě spadá, ale z jiné strany bude stačit při tak velkém rozsahu spektra 10 kHz.

Aby byl pokrýt celý interval rozbití skla je potřeba mít aspoň 15 krát větší čas zápisu a tím i 15 krát větší počet vzorků. Mít $N = 15 * 1024 = 15360$ nebo více vzorků jednoho

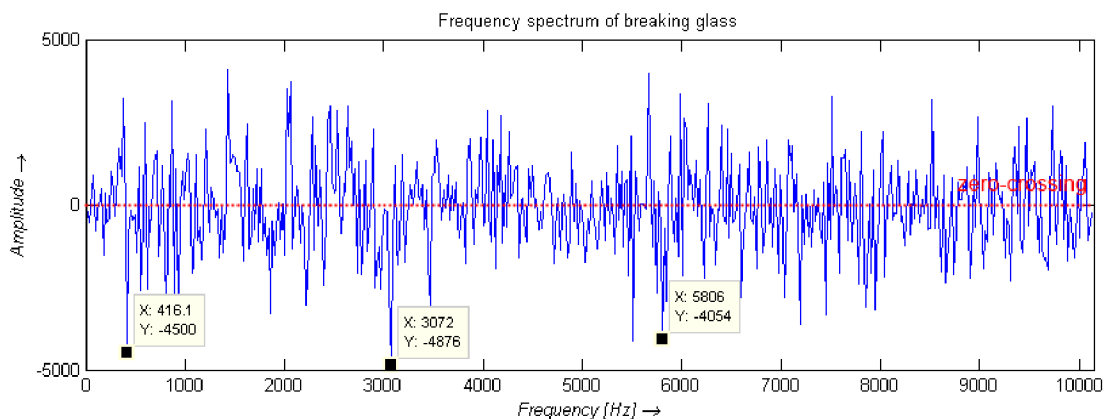
záznamu by asi nebylo dobře. To by zabralo na mikrokontroléru při implementaci v C jazyce $2B * 15360 = 30\text{KB}$ paměti pro celočíselný typ short. Lepší řešení je použití jednoho záznamu s 1024 vzorků při zvětšení času zápisu o 15 krát při stejné frekvenci vzorkování. Tím se myslí použití jednoho cyklu, který 15 krát načte záznam tak, že ve výsledku je získáván průměrný signál rozbití skla.

S průměrovaným signálem je dále lépe pracuje (viz obr. 6.3). Za prvé je to z důvodu zmenšování chyb nasbíraných dat z mikrofonu. Za druhé odpadá potřeba mít dlouhý záznam zápisu, což vede k efektivitě vykonání algoritmu detekcí rozbití skla.



Obrázek 6.3: Spektrum rozbití skla

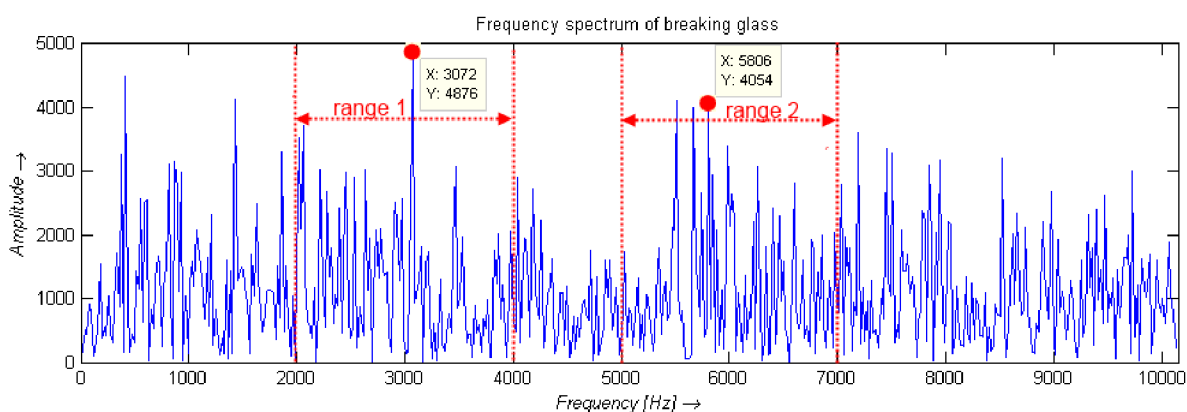
Následně je potřeba si nalézt a vybrat kritéria podle kterých bude provedena analýza získaného celého spektra rozbití skla. Podle analýzy (viz. kapitola 4) a porovnáním výsledného spektra z mikrofonu byly určeny vhodné podmínky detekcí. První z nich je tak zvaný zero-crossing kritérium, který spočívá v tom, že spočítáním průsečíků frekvenčních složek spektra s nulou se zjistí jejich počet a ten počet se pak porovnává s předem nadefinovaným počtem průsečíku charakterního pro zvuk rozbití skla. Tento kritérium hodně záleží na použitém počtu odpočtů FFT. V tomto případě se používá počet odpočtů FFT rovný 1024. Stejně důležité je tady pracovat s obousměrným spektrem (záporné a kladné hodnoty), protože při práci s absolutní hodnotami nenastane průsečík s nulou. Na obr. 6.4 je uvedena aplikace zero-crossing metody pro spektrum rozbití skla, její výsledkem je zjištěných 515 průsečíků s nulou. Algoritmus je implementován v C a je uveden v příloze, kde je kód celého programu.



Obrázek 6.4: Zero-crossing spektra rozbití skla

Z důvodu toho, že spektrum rozbití skla je zcela rozsáhlý a to klade vysoké požadavky na zpracování signálu. Hledání největších spektrálních složek spektra vyžaduje zbytečně dlouhé cykly pro nalezení každé. Po analýze řady získaných průběhů bylo zjištěno, že energie signálu rozložena ve stejných místech, jenom největší spektrální složky mění svůj řad. Pro hledání největších významných složek se aplikuje Goertzel algoritmus. S tím rozdílem, že pro hledání složek v určitém rozsahu je potřeba několika násobné použití Goertzel algoritmu. Násobek použití daného algoritmu je realizována pomocí cyklu for, který má definované počáteční a konečnou frekvence hledání s krokem vypočteným z frekvencí vzorkování. Samotná napsaná funkce Goertzel algoritmu je založena na IIR filtru druhého řadu. Výsledkem vykonání funkcí je vrácení amplitudy signálu vypočtené na zadané frekvenci. Kód Goertzel funkcí je zapouzdřen do for cyklu. Po vykonání daného cyklu je zjištěna maximální amplituda z nastaveného rozsahu, která je zapsaná společně s odpovídající frekvencí. Implementace kódu v C je uvedena v příloze.

Podle kapitoly 4 Analýza rozbití skla je nutné hledat největší spektrální složky v okolí 3 kHz a na dalších frekvencích, například v okolí 6 kHz. Proto je možné pracovat s několika rozsahy frekvence s použitím více cyklů Goertzel algoritmu. Grafickou interpretací použití cyklů s Goertzel algoritmem pro hledání největších spektrálních složek ve dvou významných rozsazích poslouží obr. 6.5. Značným příkonem toho použití je hledání největších složek na definovaných rozsazích oproti klasické transformací FFT celého spektru signálu a následného hledání maxima. Z jiné strany je potřeba předem provést dobrou analýzu rozbití skla a používat konkrétní typ skla, aby možno bylo přesně stanovit rozsahy pro hledání největších spektrálních složek spektra.



Obrázek 6.5: Hledání největších spektrálních složek rozbití skla pomocí Goertzel algoritmu

6.3. Aplikace FIR filtru

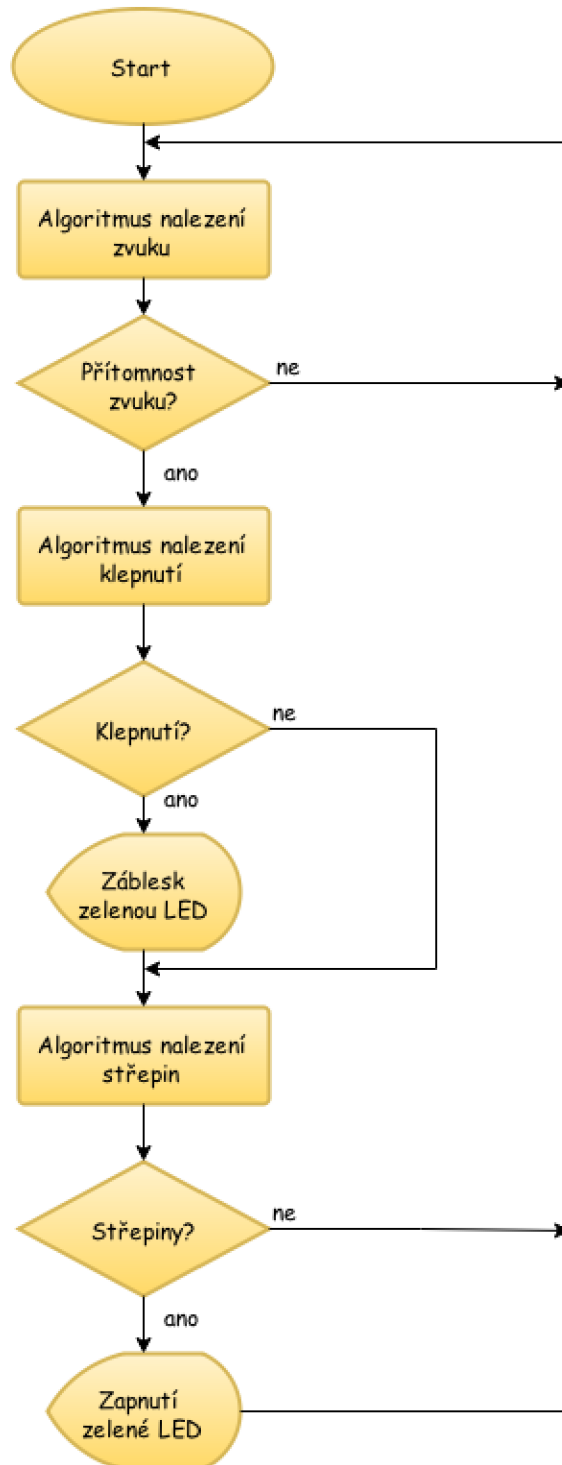
V průběhu detekcí klepnutí je potřeba odfiltrovat frekvence, které se nacházejí v blízkosti frekvencí vzorkování. Při detekci klepnutí hranice leží kolem 480 Hz, při frekvenci vzorkování 1 kHz podle 6.2.1. Na základě provedeného popisu digitálních filtrů v kapitole 5 se vybírá FIR filtr čtvrtého řádu, který má dostatečně dobrou impulsní charakteristiku. Výpočet koeficientů filtru se provádí v programu MATLAB a aplikuje se v kódu algoritmu pro zlepšení použití metoda FFT.

Pro vysokofrekvenční složky střepin skla je rovněž použit FIR filtr, ale pátého řádu. Takový řad je dostačující, aby byla zajištěna požadovaná frekvence řezu ~ 10 kHz. Pro dané frekvenci řezu jsou vypočítané koeficienty pomocí programu MATLAB. V obou případech jsou používány filtry typu dolní propust. Implementace filtru s konečnou odezvou je jednoduchá a dovoluje dosáhnout přesnosti použití rychlé Fourierovy transformací pro klepnutí o sklo a dopadajících střepin skla.

6.4. Shrnutí algoritmu

Na základě postupného rozboru jednotlivých částí algoritmu v předchozích odstavcích je tedy funkčnost algoritmu detekcí vypadá následovně viz. 6.6.

Daný program se skládá ze třech základních komponentů: nalezení aktivit, nalezení klepnutí o sklo a nalezení střepin skla. Přičemž funguje jenom v takovém pořadí, jak je to uvedeno. Nalezení aktivit spočívá v přepnutí stavu triggeru z log.0 do log. 1 a tím se objeví přítomnost zvuku okolí. Z toho okamžiku zapracuje algoritmus nalezení klepnutí, při úspěšném nalezení kterého bude krátce indikováno zelenou LED diodou. V opačném případě dioda nebude zapnutá. Nalezení klepnutí tedy nese jen informativní charakter. Program pokračuje v obou případech (buď po nalezení nebo chybnosti klepnutí) do stavu nalezení střepin skla. Touto poslední etapou přijde rozhodnutí jestli se jedná o střepiny skla. V případě nalezení střepin se rozsvítí zelenou LED diodou a tím se považuje zvuk rozbití skla za nalezený. Důležité si uvědomit, že zelená LED dioda svítí až do okamžiku přepnutí triggeru na začátku. Tam je dioda vypnutá. Jinak i v záporného případě, když nejsou střepiny nalezené, tak se považuje zvuk rozbití skla za nenalezený. Program se spouští znova a neustále opakuje.



Obrázek 6.6: Algoritmus detekcí rozbití skla

7. Testování systému

V předchozích kapitolách byla provedena analýza vybraných zvuků různými metodami. Na základě toho pak byl vytvořen postup jak daný zvuk detekovat. Takovým zvukem bylo rozbití skla, které se skládá ze dvou částí a to jsou klepnutí o sklo a střepiny. K detekování obou částí byly vytvořené potřebné k tomu algoritmy.

V dané kapitole se bude daný systém testovat a to s využitím statistických metod. K tomu aby bylo možné provést testování bylo potřeba vytvořit knihovnu podobných zvuků. Při analýze a vytvoření algoritmu se používal zápis s rozbitím klasického typu skla s tloušťkou 4 mm, která se používají v oknech.

Je důležité si uvědomit, že úspěšnost detekcí rozbití skla bude především záležet na typu použitého skla, na jeho geometrickém tvaru, na vzdálenosti spadnutí střepin a dalších parametrech. Ne vždy zvuk rozbití skla obsahuje charakterní klepnutí o sklo, jak je to u sklenici vína nebo podobných malých předmětů. Může stát i opačně kdy zvuk nebude obsahovat charakterní vyšší frekvence. Tím pádem detekování rozbití skla záleží na velkém množství faktorů.

7.1. O testu

Cílem daného testu bude zhodnotit kvalitu u binární klasifikaci (ROC analýza), která je reprezentována dvěma třídami. Jedná třída je definována s kladným řešením, druhá je s záporným řešením. ROC křivka ukáže závislost počtu správně klasifikovaných kladných příkladů od počtu chybně klasifikovaných záporných příkladů. Při tom klasifikátor má parametr podle kterého se provádí rozbíjení do dvou tříd, nazývaný prahem (cut-off value). V závislosti na prahu budou získané různé veličiny chyb prvního a druhého druhu. Množství zkoušek binární klasifikace je neomezeno a vybírá se podle konkrétní situací.

Pro pochopení významů chyb dvou druhů se rozebírá čtyřpólová sdružená tabulka (confusion matrix), která se vyrábí na základě výsledků klasifikací modelem a faktickou příslušností příkladů k třídám.

Tabulka podle obr. 7.1 obsahuje:

- TP (True Positives) - správně klasifikované kladné příklady (opravdově kladné události)
- TN (True Negatives) - správně klasifikované záporné příklady (opravdově záporné události)
- FN (False Negatives) - kladné příklady klasifikované jako záporné (chyba prvního druhu). Je tak zvaná lživá propustka když událost chybně se nenalezne (lživě záporné události).
- FP (False Positives) - záporné příklady klasifikované jako kladné (chyba druhého druhu). Je tak zvané lživé nalezení, tj při absenci událostí chybně se vynáší řešení o jeho výskytu (lživě kladné události).

7.2. Určování podmínek

K provedení testu se definují podmínky na základě kterých budou zjištěny výsledky testu. K testování daného systému s vytvořeným algoritmem detekcí se uvažuje třída, která

		Fakticky	
		kladně	záporně
Model	kladně	TP	FP
	záporně	FN	TN

Obrázek 7.1: Ukázka confusion matrix

obsahuje vzorky rozbití skla a je počítána za kladný výsledek. Třída se skládá ze zápisů rozbití různých druhů skla (okno v bytě a autech, sklenice, lahve). Z toho důvodu, že zvuky rozbití skla se velice od sebe liší a hodně záleží na jejich typu jsem přišel k závěru, že daná třída by mohla mít podtřídy.

Za první podtřídou byly vybrány zvuky rozbití skla velkých objektů v řadu velikosti metrů. Tato podtřída je tvořena především okny v budovách a autech. Patří sem ploché, autosklo a stavební sklo. Konkrétně jsou to dvojsklo, trojsklo nebo obyčejné používané sklo v oknech.

Druhá podtřída se skládá ze skla menší velikosti než v první podtřídě a jsou to sklenice, lahve, nádobí. Patří sem obalové, technické, umělecké sklo.

Tím pádem v testu zúčastní všechny nadefinované podtřídy a bude zjištěn počet úspěšných pokusů nebo neúspěšných a to s chybou prvního a druhého druhu. Za chybu prvního řádu se uvažuje propustky když aktuální třída se plete s predikovanou (například rozbití okna se plete se sklenicí). Chyba druhého řádu je představena tak, když zvuk rozbití skla se nalezne při jeho chybnosti.

7.3. Zpracování výsledků

Po definování podmínek provedení testu v předchozí části se provádí samotný test. Dané testy se provádí poměrně kvalitně a to s opakováním měření při pochybnostech ve správnosti výsledků (3 krát opakování).

Při testování vytvořeného systému pro detekci rozbití velkého skla byly získané následující výsledky (viz obr. 7.2). Z čehož vychází, že při celých 36 zápisech systém vyhodnotil 20 krát správně přítomnost zvuku rozbití skla a 6 krát správně vyhodnotil jeho absenci. Ve zbytku jsou 10 krát falešných detekování. Z nich tvoří 7 krát kdy systém detekoval absenci klepnutí nebo střepin skla při jejich přítomnosti. Naopak bylo 3 krát detekována přítomnost klepnutí nebo střepin skla při jejich absenci.

Tím pádem citlivost daného systému při detekci rozbití velkého skla je vypočítána následovně:

$$Sensitivity = \frac{TP}{TP + FN} * 100\% = \frac{20}{20 + 7} * 100 = 74\% \quad (7.1)$$

Specificita se vypočítá podle vzorce:

$$Specificity = \frac{TN}{TN + FP} * 100\% = \frac{6}{6 + 3} * 100 = 67\% \quad (7.2)$$

		Fakticky	
		kladně	záporně
Model	kladně	20	3
	záporně	7	6

Obrázek 7.2: Výsledky detekcí rozbití velkého skla

Při testování systému pro detekci rozbití malého skla vychází, že při celých 24 zápisech systém vyhodnotil 16 krát správně přítomnost zvuku rozbití skla a nulový počet správného nedetekování. Ve zbytku jsou 8 falešných detekování. Z nich tvoří 7 krát kdy systém detekoval absenci klepnutí nebo střepin skla při jejich přítomnosti. Jednou byla detekována přítomnost klepnutí o sklo při jeho faktické absenci.

Kromě srovnání zvuků rozbití skla mezi sebou by se hodilo srovnávat vzorky rozbití skla s jinými zvuky. Za jiné zvuky jsou považované zvuky domácností, které se můžou vyskytnout obyčejně (dopad předmětů, klepnutí, tlesknutí, hudba, rozhlas, hlas).

Při testování systému na ostatních zvuků byly získané následující výsledky (viz obr. 7.3). Celkem se nasbíralo 103 vzorků. Z nich 93 krát správně definoval absenci zvuku rozbití skla. Zbytkem jsou 10 krát falešných nalezení, při kterých fakticky zvuk rozbití skla chyběl, ačkoli snímač ukázal jeho přítomnost.

Specificita tedy vyšla:

$$Specificity = \frac{TN}{TN + FP} * 100\% = \frac{93}{93 + 10} * 100 = 90\% \quad (7.3)$$

Při všech provedených testech je celkový výsledek vychází podle obr. 7.4 pro celých 163 událostí. Chyby prvního a druhého řadů vzniklé buď velkou vzdáleností od mikrofonu nebo nedostačující kalibrací triggeru modulu s mikrofonem jsou přibližně stejné. Tím pádem odchylka funkčnosti algoritmu vyšla minimální a 135 událostí bylo úspěšně zpracovááno.

V procentním poměru celková citlivost a specificita daného systému detekcí zvuku vyšla:

$$Sensitivity = \frac{TP}{TP + FN} * 100\% = \frac{36}{36 + 14} * 100 = 72\% \quad (7.4)$$

		Fakticky	
		kladně	záporně
Model	kladně	0	10
	záporně	0	93

Obrázek 7.3: Výsledky detekcí rozbití skla pro jiné zvuky

$$Specificity = \frac{TN}{TN + FP} * 100\% = \frac{99}{99 + 14} * 100 = 87\% \quad (7.5)$$

		Fakticky	
		kladně	záporně
Model	kladně	36	14
	záporně	14	99

Obrázek 7.4: Celkový výsledek detekcí rozbití skla

8. Závěr

Cílem práce bylo seznámit se s používanými typy mikrofonů, jejich vlastnostmi a připojením vybraného mikrofonu k mikrokontroléru. Na začátku bylo provedeno seznámení s podobnými systémy pro detekci zvuku. Následně byl proveden návrh schématy pro vytvoření nezbytného zařízení. K tomu byly popsány jednotlivé součástky. V souladu s požadavky byla vybraná vývojová platforma KL25Z firmy Freescale. Byla popsána periferie a funkčnost vybrané desky. V práci se provedl rozbor koncového zařízení na základě porovnání jejich vlastností. Volba přišla na modul s elektretovým mikrofonem. Periferie a funkčnost vybraného modulu byla popsána a použita u připojení k vývojové desce.

Pro další vyhodnocování byla nasbíraná řada zvuků. Nadefinovány byly dva druhy zvuku. Za jednoduchý druh byl vybrán zvuk ptáku. Patřily sem křiky havranů, sojek a kukaček. Za složitý byl vybrán zvuk rozbití skla, který byl rozdělen do dvou kategorií. Do kategorie malého skla patřily lahvi, nádoby, brýle a další malé výrobky ze skla. Do kategorie velkého skla vešly okenní sklo v domácnosti a autech. Celkem v průběhu vývoje a testování systému bylo použito kolem 150 vzorků.

K rozpoznávání vybraných zvuků bylo nutné vybrat a vyzkoušet vhodnou metodu. Byl jsem se seznámen s metodami rychlé Fourierové transformací a dynamického borcení času. Při analýze zvuku křiku ptáků byla použita FFT metoda, analýza v časové oblasti a navíc analýza pomocí spektrogram. Bylo provedeno testování DTW metody pro porovnání zvuku křiků ptáků mezi sebou. V dané aplikaci metoda DTW měla výhodnost použití při porovnání vybraných křiků ptáků mezi jednotlivými řády ptáků.

Analýza zvuku rozbití skla probíhala převážně pomocí FFT metody. Analýza v časové oblasti a použití spektrogram pomohla rozdělit zvuk rozbití skla do dvou částí (klepnutí o sklo a dopadající střepiny skla). V důsledku se zjednodušil další postup vytvoření algoritmu detekcí. Analýza obojích druhů zvuků probíhala v programu MATLAB.

Pro vytvoření algoritmu detekcí zvuku rozbití skla nezbytným bylo se seznámit s realizací dříve uvedených metod na mikrokontroléru. Po analýze možných způsobů provedení realizací algoritmu přednost byla dána algoritmu Goertzel, protože jeho aplikace dovolila značně snížit počet prováděných operací pro hledání vysokofrekvenčních složek rozbití skla. Kromě realizací metod byl vypočítán a aplikován FIR filtr pro lepší funkčnost metody FFT. Použití celočíselné aritmetiky rovněž povolilo zkrátit počet prováděných mikrokontrolérem operací.

Celý systém se testoval uvedenými v kapitole 7 způsoby. Při testování řadu vzorků skla, kde se hodnotily úspěšné pokusy nalezení zvuku rozbití skla, vyšla citlivost kolem 74% a specifická 67%. Při testování systému s jinými vzorky než sklo se hodnotil počet úspěšně nedetekovaných zvuků rozbití skla. V tomto případě vyšel 90% výsledek specifické systémy při 103 událostí. Předvádí to, že systém nejlépe rozlišuje ostatní zvuky mezi sklem.

Celkový výsledek funkčnosti algoritmu detekcí zvuku při zpracovávání 165 událostech předvádí celkovou citlivost 72% a specifickou 87% detekování zvuku rozbití skla.

Získaný výsledek ukazuje, že vytvořený systém detekcí zvuku může najít úspěšnou aplikaci v životních podmínkách, ovšem potřebuje zdokonalení pro použití v průmyslu a výrobě. Toho je lze dosáhnout zlepšením kvality zařízení a provedením větším počtem testů v laboratorních podmínkách.

S použitím vytvořeného systému lze stavět různé systémy, jako jsou bezpečnostní systémy nebo systémy ohlašování. Při relativně levném a jednoduchém řešení daný systém poskytuje dostačující kvalitu pro využití v domácnosti a různých uživatelských aplikacích. Daný systém může být značně rozšířen a tím najít větší počet možných použití.

Literatura

- [1] SLOSS, Andrew N, Dominic SYMES, Chris WRIGHT: *ARM system developer's guide: designing and optimizing system software*. Amsterdam: Elsevier, 2004. 689 p. ISBN 15-586-0874-5.
- [2] FREESCALE. *FRDM-KL25Z User's Manual*. [online]. Rev 2.0. 2013 [cit. 2015-02-20]. Dostupné z: http://cache.freescale.com/files/32bit/doc/user_guide/FRDMKL25ZUM.zip?fsp=1.
- [3] ARM MBED. *mbed Compiler*. [online] [cit. 2015-02-20]. Dostupné z: <https://developer.mbed.org/compiler/>.
- [4] *Mikrofony*. [online]. 2010 [cit. 2015-02-23]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/377-mikrofony>.
- [5] HUCK: *Princip funkce mikrofону*. [online]. 3.06.2007 [cit. 2015-02-23]. Dostupné z: <http://www.haaf.cz/view.php?cisloclanku=2007060305>.
- [6] Antonín Vojáček: *MEMS mikrofony – obecný popis struktury a funkce*. [online]. 27.04.2010 [cit. 2015-02-23]. Dostupné z: <http://automatizace.hw.cz/mems-mikrofony-obecny-popis-struktury-a-funkce>.
- [7] STMICROELECTRONICS. *AN4426 Application note: Tutorial for MEMS microphones*. [online]. Rev 1. 2014 [cit. 2015-02-23]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/DM00103199.pdf.
- [8] ZVUKAŘ. *Porozumění specifikací mikrofону: Specifikace, technika*. [online]. 2012 [cit. 2015-02-23]. Dostupné z: http://www.digiton.ru/uploads/files/zvuko_3_2012.pdf.
- [9] FRADEN, Jacob: *Handbook of Modern Sensors: Physics, Designs, and Applications*. Third Edition. New York: Springer, 2010. 663 p. ISBN 978-1-4419-6465-6.
- [10] *Keyes Microphone Sound Detection Sensor Module for Arduino: Pinouts*. [online]. 2012 [cit. 2015-03-1]. Dostupné z: <http://www.idiymart.com/wholesale/arduino-microphone-sound-detection-sensor-module.html>.
- [11] *Keyes Microphone Sound Detection Sensor Module for Arduino: Scheme*. [online]. 2012 [cit. 2015-03-1]. Dostupné z: <http://www.marcmart.com/ebay/o/OT927/OT927.jpg>.
- [12] Theodoros Damoulas, Samuel Henry, Andrew Farnsworth: *Bayesian Classification of Flight Calls with a novel Dynamic Time Warping Kernel*. [online] NY: Cornell University, 2010 [cit. 2015-03-1]. Dostupné z: http://www.cs.cornell.edu/~damoulas/Site/papers_files/icmla2010.pdf.
- [13] MÜLLER, MEINARD: *Information Retrieval for Music and Motion*. NY: Springer, 2007. ISBN 978-3-540-74047-6.

- [14] Stan Salvador, Philip Chan: *FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space*. [online] Melbourne: Florida Institute of Technology, 2004 [cit. 2015-03-12]. Dostupné z: <https://gi.cebitec.uni-bielefeld.de/teaching/2007summer/jclub/papers/Salvador2004.pdf>.
- [15] ANALOG DEVICES. *Section 6 Digital Filters*. [online]. Rev 1. 2014 [cit. 2015-03-12]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/DM00103199.pdf.
- [16] Petr Vavřín, Pavel Jura: *Systémy, procesy a signály 2*. Brno: Vutium, 1999. 130 s. ISBN 802-14129-1-7.
- [17] Jiří Tůma: *Zpracování signálů získaných z mechanických systémů užitím FFT*. Praha: Sdělovací technika, 1997. 174 s. ISBN 80-901936-1-7.
- [18] Jan Uhlíř, Pavel Sovka: *Zpracování signálů získaných z mechanických systémů užitím FFT*. Praha: Vydavatelství ČVUT, 1995. ISBN 800-10130-3-0.
- [19] Vratislav Davídek: *Implementace algoritmů číslicového zpracování signálů v reálném čase*. Praha: Vydavatelství ČVUT, 2004. ISBN 802-14129-1-7.
- [20] Sergienko, A: *Digital signal processing*. St. Petersburg: Piter, 2002, 586 p. ISBN 5-318-00666-3.
- [21] MANDAL, Mrinal Kr a Amir ASIF: *Continuous and discrete time signals and systems*. New York: Cambridge University Press, 2007, xiv, 865 p. ISBN 05-218-5455-5.
- [22] Bhargavi Nisarga, Kripasagar Venkat: *A Robust Glass-Breakage Detector Using the MSP430* [online]. Texas Instruments. February 2008 [cit. 2015-4-10]. Dostupné z: <http://www.ti.com/lit/an/slaa389/slaa389.pdf>.
- [23] M.V.Swathi, Ch.Sireesha, K.Bindu, M.Triveni: *DESIGN AND IMPLEMENTATION OF ROBUST GLASS BREAK DETECTOR USING ULTRA-LOW POWER MSP430F2274*. [Project Report.] Hyderabad: Jawaharlal Nehru Technological University, ELECTRONICS AND COMMUNICATION ENGINEERING, 2013. 62 s.
- [24] Petr Sysel, Pavel Rajmic: *Goertzel algorithm generalized to non-integer multiples of fundamental frequency*. EURASIP Journal on Advances in Signal Processing. 56. 2012
- [25] Ricardo A. Losada. *Practical FIR Filter Design in MATLAB*. [online]. Rev 1. 2013 [cit. 2015-5-7]. Dostupné z: http://www.secs.oakland.edu/~ganesan/old/courses/CSE671SU08mathworks_firdesign.pdf.

9. Seznam použitých zkratek a symbolů

DFT	Diskrétní Fourierova transformace
FFT	Rychlá Fourierova transformace
DTW	Dynamické borcení času
FIR	Konečná impulsní odezva
IIR	Nekonečná impulsní odezva
ADC	Analogově číslicový převodník
DTMF	Dual-tone multi-frequency
HW	Hardware
DP	Dolní propust
LP	Low-pass
HP	High-pass
BP	Band-pass
BR	Band-rejection
MATLAB	Matrix laboratory
AO	Analog output
DO	Digital output
DC	Direct current
LED	Light-emitting diode
MEMS	Microelectromechanical systems
SNR	Signal-to-noise ratio
FET	Field-effect transistor
SAR	Převodník s postupnou aproximací
MCU	Microcontroller unit
DMA	Direct memory access
DAC	Digital-to-analog converter
SRAM	Static random-access memory

I/O	Input/output
USB	Universal Serial Bus
flash	Flash memory
ROC	Receiver operating characteristic

10. Seznam příloh

A. Kód programy

A.1. Zdrojový kód algoritmu detekcí zvuku rozbití skla

Uvedený skript představuje realizaci algoritmu detekcí zvuku rozbití skla.

```
1 // ===== //
2 // Sound Detection Algorithm //
3 // Using FFT function with fixed points //
4 // ===== //
5 //////////////////////////////////////
6
7 //////////////////////////////////////
8 // Includings
9 //////////////////////////////////////
10 #include "mbed.h"
11 #include "fix_fft.c"
12 #include "fhanning.h"
13 #include "goertzel.h"
14 #include "FastAnalogIn.h"
15 #include <math.h>
16
17 #define SPECTRUM 1
18
19 //////////////////////////////////////
20 // Settings and definitions
21 //////////////////////////////////////
22 //(+) Defines LED indicators
23 DigitalOut a(LED1); // LED 1 Red
24 DigitalOut c(LED2); // LED 2 Green
25 DigitalOut d(LED3); // LED 3 Blue
26
27 //(+) Defines pins microphone
28 FastAnalogIn mic(A0,0); // using FastAnalogIn library for ADC pin
29 // microphone
30 DigitalIn mic2(PTA1); // using for DAC pin microphone (trigger)
31
32 //(+) Defines variables
33
34 #define FFT_SIZE 256 // FFT size
35 #define log2FFT 8 // Power of 2
36 #define N (2*FFT_SIZE) // Array size
37 #define log2N (log2FFT + 1)
38
39 Timer timer; // Timer 1
40 Timer timer2; // Timer 2
41
42 //////////////////////////////////////
43 // Variables and arrays definition //
44 //////////////////////////////////////
45 short x[N], y1[N*4], fx[N]; // Signal and spectrum arrays
46 // LP FIR filter 10 kHz coefficients
47 float coef[6] = {0.338500748481199, -0.217211961301011, 0.637899352791997,
48 0.637899352791997, -0.217211961301011, 0.338500748481199};
```

```

47 // LP FIR filter 480 Hz coefficients
   float coef2[5] = {-0.24465372903124544, 0.010692541937508238,
   0.98930745806249099, 0.010692541937508238, -0.24465372903124544};
49 float FS,FS2;           // Sampling frequency
   float mean, mean2;     // Mean of spectrum
51 double ptr;
   int i,j;               // Addition variables
53 int cycles;            // Number of cycles

55 ////////////////////////////////////////////////////////////////////
   // MAIN PROGRAM
57 ////////////////////////////////////////////////////////////////////
   int main()
59 {
   a=1,c=1,d=1;          // Sets LEDs off
61
   ////////////////////////////////////////////////////////////////////
63 // READING DATA FROM MICROPHONE
   ////////////////////////////////////////////////////////////////////
65 ////////////////////////////////////////////////////////////////////
   // 1) READING GLASS BLOW
67 ////////////////////////////////////////////////////////////////////
   while (true)
69   {
   if(mic2.read()> 0.0) // if trigger on
71   {
   a=1,c=1,d=1;        // Sets LEDs off
73   cycles = 0;
   for (i=0;i<(N*4);i++)
75   {
   if(i>((N*4)/2)) break;
77   y1[i]=0;
   }
79   timer.reset();    // Reset timer 1
   timer2.reset();   // Reset timer 2
81   a=0.5;           // Switch red LED on while recording
   timer.start();    // Start timer 1
83
   for (i=0; i<N ;i++)
85   {
   wait_us(1000);
87   x[i] = (100000*(mic.read())-800000); // Read data microphone
89
   if(i>=(N/2)) break;
   }
91   timer.stop();      // Stop timer 1

93 ////////////////////////////////////////////////////////////////////
   // 2) READING GLASS HIGH FREQUENCY
95 ////////////////////////////////////////////////////////////////////
   timer2.start();   // Start timer 2
97
   short zeros=0;    // Crossing zeros counter
99   int l=15;        // Number of recording
   int m=1;          // Decrement
101

```

```

103     for (i=0;i<(N*4);i++)
104     {
105         if ((i == (N*4)/2) && (m > 0))
106         {
107             timer2.stop(); // Stop timer 2
108             i=0; m--;
109         }
110         wait_us(30.5); // Pause in reading
111
112         if(i>((N*4)/2)) break;
113
114         y1[i] += ((float)(100000*(mic.read())-800000))/((float)1); // Read
data microphone
115     }
116
117     FS=((float)(N/2))/timer.read();
118     FS2=((float)((N*4)/2))/timer2.read();
119
120     a=1,c=1,d=1;
121
122     //////////////////////////////////////
123     // ANALYSIS AND RECOGNITION
124     //////////////////////////////////////
125
126     //////////////////////////////////////
127     // Digital filtration with cut off frequency 10.0 kHz
128     //////////////////////////////////////
129     //filter(y1,N*4,coef,5);
130     for(int k=0;k<(N*2);k++)
131     {
132         float buffer=0.0;
133         for(int x=0;x<6;x++)
134         {
135             buffer += coef[x]*(float)(y1[k+x]);
136         }
137         y1[k] = buffer;
138     }
139     //////////////////////////////////////
140     // Digital filtration with cut off frequency 480.0 Hz
141     //////////////////////////////////////
142     for(int k=0;k<(N/2);k++)
143     {
144         float buffer2=0.0;
145         for(int i=0;i<5;i++)
146         {
147             buffer2 += coef2[i]*(float)(x[k+i]);
148         }
149         x[k] = buffer2;
150     }
151     //////////////////////////////////////
152     // Goertzel algorithm
153     //////////////////////////////////////
154     int fr,num,max=0;
155     float step = FS2/(N*2); // Frequency step
156     float mag; // Amplitude

```

```

157 for (fr=(145*step); fr <=(200*step); fr=fr+step) // from 2900 Hz to 4000 Hz
159 {
    mag = goertzel_mag(N*2,abs(fr),FS2,y1);
161     if(mag > max)
    {
163         max = mag; // Maximum amplitude
        num = abs(fr); // Frequency of maximum
165     }
    }
167 ////////////////////////////////////////////////////////////////////
// Fast Fourier Transform using
169 ////////////////////////////////////////////////////////////////////
    fix_fftr(x, log2N, 0); // Blow spectrum calculation
171 fix_fftr(y1, 10, 0); // High frequency spectrum calculation
    #if SPECTRUM
173 x[0]=0; // Zeroed first element of spectrum
    fx[0]=0; // Zeroed first element of spectrum
175 float f=0.0; // Frequency variable
    int start=0,final=0; // Limited indexes
177 int fincross; // Mean crosses
    int diff=0; // Sum elements of spectrum
179 do // Defines start frequency
    {
181     f += FS/(float)FFT_SIZE;
        start++;
183 } while(f<200.0);

185 final=start;
    fincross=start;
187 do // Defines final frequency
    {
189     f += FS/(float)FFT_SIZE;
        final++;
191     if(f<350.0) fincross++;
    } while(f<480.0);
193
    for (i=0; i<(N/2); i++)
195 {
        if(i<start)
197 x[i]=0;
        if(i>final)
199 x[i]=0;
        diff += abs(x[i]);
201 }

203 y1[0]=0;
    mean = (float)diff/(float)((N/2)-(start+final)); // Mean of spectrum
205
    ////////////////////////////////////////////////////////////////////
207 // DETECTION GLASS BLOW
    ////////////////////////////////////////////////////////////////////
209     bool cond1=false,cond2=false,cond3=false; // Condition variables
        short peak1=0,peak2=0,peak3=0; // The biggest three peaks
        defined zeros
211     short fp1=start,fp2=start,fp3=start; // Frequency indexes

```

```

    short crossmean1=0,crossmean2=0;           // Number mean crossed of
spectrum
213 float fcoef=0.0;                          // Frequency coefficient
    int n=start;                              // Set start index in while
    loop
215
//(+)  
217 // Finding 3 biggest peaks of spectrum
while(n<final)
{
219     if (abs(x[n]) > peak3)
        {
221         if (abs(x[n]) > peak2)
            {
223             peak3 = peak2;
                fp3=fp2;
225             if (abs(x[n]) > peak1)
                {
227                 peak2 = peak1;
                    peak1 = abs(x[n]);
229                 fp2=fp1;
                    fp1=n;
231             }
                else
233             {
                    peak2 = abs(x[n]);
235                 fp2=n;
                }
            }
        }
        else
237     {
239         peak3 = abs(x[n]);
241         fp3=n;
        }
    }
243
    if((n<fincross)&&(abs(x[n])>mean)) crossmean1++; // Counts amplitudes
above mean of spectrum
    if((n>fincross)&&(abs(x[n])>mean)) crossmean2++; // Counts amplitudes
above mean of spectrum
247
    n++;
249 }
    fcoef=FS/FFT_SIZE;
251
////////////////////////////////////
253 // CONDITION 1 CHECKING RANGE OF PEAKS
////////////////////////////////////
255
if((( fcoef*fp1 )>200.0)&&((fcoef*fp1 )<400.0)) cond1 = true;
257 if((( fcoef*fp2 )>200.0)&&((fcoef*fp2 )<400.0)) cond1 = true;
if((( fcoef*fp3 )>200.0)&&((fcoef*fp3 )<400.0)) cond1 = true;
259
////////////////////////////////////
261 // CONDITION 2 CHECKING PEAK RATIO
////////////////////////////////////
263

```



```

265 if((((float)peak1)/mean) > 3.0) cond2 = true;
267 ////////////////////////////////////////////////////
267 // CONDITION 3 CHECKING SIGNAL ENERGY
267 ////////////////////////////////////////////////////
269 if( crossmean1 > crossmean2 ) cond3 = true;
271 if((cond1==true) || (cond2==true) || (cond3==true)) {a=0; c=0; d=1;}
273 if(((cond1==true)&&(cond2==true)) || ((cond2==true)&&(cond3==true)) || ((cond1
==true)&&(cond3==true))) {a=1; c=1; d=0;}
if((cond1==true)&&(cond2==true)&&(cond3==true)){a=1; c=0; d=1;}
275 wait_ms(1000);
277 a=1;c=1;d=1;
279 ////////////////////////////////////////////////////
279 // DETECTION GLASS HIGH FREQUENCY
279 ////////////////////////////////////////////////////
281 bool cond4=false , cond5=false;
283 // Zero-crossing counter and meaning of spectrum
283 ////////////////////////////////////////////////////
285 mean2 = 0.0;

287 for (i=0;i<(N*4);i++)
{
289     if(i>((N*4)/2))
{
291         //mean2 = (float)sum2/(i-1);
break;
293     }
// Zero-crossing counter
295     if(i!=0)
{
297         if( ( float(y1[i-1])/float(y1[i]) < 0.0) && (y1[i]!=0)) zeros
++;
else if( (y1[i] == 0) && ((float)y1[i-1] > 0.0) && ((float)y1[i
+1] < 0.0) ) zeros++;
299         else if( (y1[i] == 0) && ((float)y1[i-1] < 0.0) && ((float)y1[i
+1] > 0.0) ) zeros++;
}
301         mean2 += ((float)abs(y1[i]))/((N*4)/2);
//sum2 += abs(y1[i]);
303     }
305 ////////////////////////////////////////////////////
305 // CONDITION 4 ZERO-CROSSING
305 ////////////////////////////////////////////////////
307 if(zeros>=480) cond4 = true;

309 ////////////////////////////////////////////////////
309 // CONDITION 5 HIGH FREQUENCY PEAKS
309 ////////////////////////////////////////////////////
311 if(((float)max/mean2) > 3.0) cond5 = true;
313
315 ////////////////////////////////////////////////////

```

```

// BREAKING GLASS DECISION
317 ///////////////////////////////////////////////////////////////////
bool blow = false , crash = false;
319
wait_ms(1000);
321
if((cond1==true)&&(cond2==true)&&(cond3==true)) // Blow detected => LED is
yellow
323 { blow = true; a=0; c=0; d=1; }

325 if((cond4==true)&&(cond5==true)&&(blow==true)) // Blow + crash detected =>
LED is green
{ crash = true; a=1; c=0; d=1;}
327
#endif
329 }
}
331 }

```

main.cpp

B. Analýza v programu Matlab

B.1. Zdrojový kód DTW algoritmu

Uvedený skript představuje realizaci metody DTW na příkladě porovnání zvuku dvou křiků ptáků.

```
% Similarity detection for sounds
2 % Dynamic time warping (DTW)
%% Reading two arrays from .txt
4 % Write as array S1 (bird sound 1) and S2 (bird sound 2)
  input1 = fopen('1.txt');
6 input2 = fopen('2.txt');
  S1 = fscanf(input1, '%f ');
8 S2 = fscanf(input2, '%f ');
  fclose(input1);
10 fclose(input2);

12 Fs = 44100;           % Discretization frequency
  Nsamps1 = length(S1); % Number of samples sound 1
14 Nsamps2 = length(S2); % Number of samples sound 2
  t1 = (1/Fs)*(1:Nsamps1); % Prepare time data sound 1
16 t2 = (1/Fs)*(1:Nsamps2); % Prepare time data sound 2

18 %=====
%% Fast Fourier Transform of Bird Sound
20 y_fft1 = abs(fft(S1)); % Retain Magnitude
  y_fft1 = y_fft1(1:Nsamps1/2); % Discard Half of Points
22 f1=Fs*(0:Nsamps1/2-1)/Nsamps1; % Prepare freq data

24 y_fft2 = abs(fft(S2)); % Retain Magnitude
  y_fft2 = y_fft2(1:Nsamps2/2); % Discard Half of Points
26 f2=Fs*(0:Nsamps2/2-1)/Nsamps2; % Prepare freq data

28 %=====
% Comparing two vectors
30 % Gets F – frequency vector F1, F2
  % Gets PSD vector P1,P2
32
  [Sv1,F1,T1,P1]=spectrogram(S1,1024,[],[],Fs);
34 [Sv2,F2,T2,P2]=spectrogram(S2,1024,[],[],Fs);
  %1) Defines length of time vectors
36 len=length(T1);
  len2=length(T2);
38 %2) Defines min length of time vectors to comparison
  if(len<len2)
40     len = len;
  else
42     len = len2;
  end
44 %3) Defines end index of detection
  i=1;j=1;
46 while(F1(i)<8000)
    i=i+1;
```

```

48 end
   while(F2(j)<8000)
50     j=j+1;
   end
52 %4) Defines min index for PSD vector
   if(i<j)
54     i = i;
   else
56     i = j;
   end
58 %5) Comparison between two vectors using DIW
   b=0;
60 if(i==j)
       for row = 1:i
62         v1 = P1(row,1:len);
           v2 = P2(row,1:len);
64         vector_dtw(row)=dtw(v1,v2,1024);
           b=b+vector_dtw(row);
66         end
   end
68
   fin = fopen('logdtw.txt','a');
70 fprintf(fin,'\n\n%1.10f ',b);
   fclose(fin);

```

dtw.m

C. Obsah příloženého CD

Na CD jsou uvedeny vzorky zapsaných zvuků rozbití skla (okna, lahvi, nádobí) a křiků ptáků (sojky, havrany, kukačky), které byly použité v průběhu analýzy a testování systému. Pro testování zvuku rozbití skla byly zapsané pomocí vybraného mikrofonu v průběhu prací jiné vzorky (dopad předmětů o zem, klepnutí o lahev, rozhlas, hlas a další zvuky domácností). Také jsou uvedeny skripty použité v programu MATLAB při realizaci metody DTW a zdrojový kód algoritmu detekcí rozbití skla. Analýza se prováděla v Matlab 2013a.