



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

AUTONOMNÍ SYSTÉM PRO AKUSTICKOU ANALÝZU V DIAGNOSTICE

AUTONOMOUS SYSTEM FOR ACOUSTIC ANALYSIS IN DIAGNOSTICS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Havlíček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Rostislav Huzlík, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	Bc. Michal Havlíček
Studijní program:	Kvalita, spolehlivost a bezpečnost
Studijní obor:	bez specializace
Vedoucí práce:	Ing. Rostislav Huzlík, Ph.D.
Akademický rok:	2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Autonomní systém pro akustickou analýzu v diagnostice

Stručná charakteristika problematiky úkolu:

Předmětem práce je návrh a realizace systému založeného na předloženém vývojovém kitu pro využití akustickou analýzu v rámci diagnostiky.

Cíle diplomové práce:

Proveďte literární rešerši na téma algoritmů pro analýzu zvuku v rámci diagnostiky.

Seznamte se s předloženým vývojovým kitem.

Zhodnoťte možnosti užití možných algoritmů pro analýzu zvuku v rámci diagnostiky s využitím předloženého vývojového kitu.

Navrhněte a realizujte zařízení pro získání a analýzu zvuku na předloženém hardwaru.

Zhodnoťte využití zařízení v praxi.

Seznam doporučené literatury:

MILÁČEK, Stanislav, 2001. Měření a vyhodnocování mechanických veličin. Praha: Vydavatelství ČVUT. ISBN 80-010-2417-2.

KREIDL, Marcel a Radislav ŠMÍD, 2006. Technická diagnostika: senzory, metody, analýza signálu. Praha: BEN - technická literatura. Senzory neelektrických veličin. ISBN 80-7300-158-6.

TÚMA, Jiří, 1997. Zpracování signálů získaných z mechanických systémů užitím FFT. Praha: Sdělovací technika. ISBN 80-901936-1-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato diplomová práce se zabývá možnostmi analýzy zvuku na vývojové platformě Sipeed na hardwaru MaixBit. Návrhem a sestavením autonomního zařízení, které po povelu z počítače za pomoci vytvořeného programu provede analýzu okolního zvuku a výsledné informace převede zpět na monitor počítače. Obsahem závěru je zhodnocení využití těchto zařízení v praxi.

ABSTRACT

The aim of this diploma thesis are possibilities of sound analysis with Sipeed development platform using MaixBit hardware. The main goal is design and development of an autonomous device that after a command from a computer analyzes the ambient sound and converts the resulting information back to the computer screen. Usage of these devices in real world scenarios is evaluated in conclusion.

KLÍČOVÁ SLOVA

Analýza zvuku, autonomní zařízení, akustická diagnostika, Sipeed, Maixpy IDE, Rychlá fourierova transformace, akustická kamera

KEYWORDS

Sound analysis, autonomous equipment, acoustic diagnostics, Sipeed, Maixpy IDE, fast furier transformation, acoustic Camera

BIBLIOGRAFICKÁ CITACE

HAVLÍČEK, Michal. *Autonomní systém pro akustickou analýzu v diagnostice*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/140173>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Rostislav Huzlík.

PODĚKOVÁNÍ

Tímto bych rád poděkoval panu Ing. Rostislavovi Huzlíkovi, Ph.D. za cenné připomínky a rady týkající se zpracování diplomové práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce na téma autonomní systém pro akustickou analýzu v diagnostice je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Rostislava Huzlíka, Ph.D a s použitím literatury a dalších zdrojů, které jsou všechny uvedeny v seznamu použitých zdrojů.

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

V Brně dne

.....

Bc. Havlíček Michal

OBSAH

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	15
ÚVOD	16
1 ANALÝZA ZVUKU.....	17
1.1 Zvuk.....	17
1.2 hluk	18
1.3 Akustika	18
1.4 Akustické veličiny	19
1.4.1 Akustický tlak:.....	19
1.4.2 Akustická rychlost:	19
1.4.3 Intenzita zvuku:	20
1.4.4 Akustický výkon:.....	20
1.5 Hladiny akustických veličin.....	20
1.5.1 Hladina akustického tlaku L_p :.....	20
1.5.2 Hladina akustické rychlosti L_u :.....	20
1.5.3 Hladina intenzity zvuku L_I :	21
1.5.4 Hladina akustického výkonu L_w :	21
1.6 Mikrofon	21
1.6.1 Vlastnosti mikrofonů	22
1.7 Zvukoměr, zvukový analyzátor	23
1.7.1 Kalibrace zvukoměrů:.....	24
1.8 Akustická holografie.....	25
1.9 Akustické kamery	25
1.9.1 Mikrofonní pole.....	26
1.9.2 Metody pro určení směru příchodu signálu.....	26
2 ALGORITMY PRO ANALÝZU A ZPRACOVÁNÍ DIAGNOSTICKÝCH SIGNÁLŮ.....	28
2.1 Frekvenční analýzy	28
2.2 CPB analýza.....	28
2.3 Řádová analýza	29
2.4 FFT analýza	29
2.5 Kepstrální analýza.....	30
2.6 Vlnová (wavelet) transformace.....	31
3 PLATFORMA SIPEED A SOFTWARE.....	32

3.1	Maix Bit v2.0.....	32
3.1.1	Čip Kendryte K210	33
3.1.2	Mikrofon MSM261S4030H0	34
3.2	Maixpy IDE	35
3.3	Open source knihovny	36
3.3.1	Maixhub:	37
3.3.2	Github.....	37
3.4	Využitý Software.....	37
3.4.1	Hercules.....	38
3.4.2	Matlab App designer	38
4	NÁVRH A REALIZACE ZVUKOVÉHO ANALYZÁTORU	41
4.1	Návrh funkcí.....	41
4.2	Návrh konstrukce.....	42
4.3	Komunikace.....	43
4.3.1	UART	43
4.3.2	Napěťový dělič	44
4.3.3	Arduino CH340G USB	45
4.4	Zdrojové kódy	45
4.4.1	Maixpy IDE.....	46
4.4.2	App designer	50
4.5	Implementace rozšiřujících funkcí	52
4.5.1	Rozšíření Maixpy IDE	54
4.5.2	Rozšíření app designer	55
4.6	Výsledný zvukový analyzátor verze 1	56
4.6.1	Ověření funkčnosti zařízení	57
5	NÁVRH NA VYLEPŠENÍ	59
5.1	Návrh konstrukce akustické kamery	59
5.2	Periferie využité ke konstrukci	60
5.2.1	Mikrofonní pole Sipeed.....	60
5.2.2	LCD display RY24049B0-00.....	61
5.2.3	Kamera OV2640	62
5.3	Zdrojové kódy	63
5.3.1	Maixpy IDE.....	63
5.3.2	App designer	65
5.4	Výsledný zvukový analyzátor verze 2.....	66

5.4.1	Ověření funkčnosti zařízení.....	69
5.5	Návrhy na budoucí vylepšení akustické kamery	72
6	ZHODNOCENÍ VYUŽITÍ ZAŘÍZENÍ V PRAXI.....	73
7	ZÁVĚR.....	75
8	SEZNAM POUŽITÝCH ZDROJŮ	77
9	SEZNAM OBRÁZKŮ A TABULEK.....	79
9.1	Seznam tabulek	79
9.2	Seznam obrázků.....	79
10	SEZNAM PŘÍLOH.....	81

ÚVOD

V průběhu poslední dekády dochází k rychlému vývoji v oblasti elektrotechniky a tím i na poli analýzy a diagnostických prostředků. A to především díky zlepšujícím se technologiím ve výpočetní technice, která díky lepším mikroprocesorům získává větší výpočetní výkon.

Zpracováním signálů se zabývá celý samostatný vědeckotechnický obor na rozhraní elektrotechniky a aplikované matematiky. V obecné rovině můžeme říct, že signál je v podstatě určitý typ zprávy, která může obsahovat velké množství různých informací. Při zpracování různých typů signálů je možné v praxi využít vícero různých algoritmů, díky těmto algoritmům je pak zpracování informací podstatně rychlejší a pro reálnou praxi představují komfortnější využití v podstatě na jakýchkoliv platformách. Jednou z těchto platform je pak v této práci použitý vývojový kit Maix bit, kterému bude v rámci práce věnována velká část pozornosti.

Tato práce pojednává o řešení tématu návrhu a sestavení autonomního systému pro akustickou analýzu. K tématu bylo přistupováno systematickým přístupem. Důležitá je základní znalost řešené problematiky, za tímto účelem byla provedena důkladná rešerše na téma zvuk, jeho veličiny a algoritmy pro jeho zpracování. Nezbytné pro úspěšné splnění požadovaných cílů je chápání problematiky v širokém úhlu pohledu, s tím souvisí také seznámení se s využívanými přístroji v diagnostické praxi. To obnáší základní znalost hlavních výrobců zvukoměrů, zvukových analyzátorů a akustických kamer. Za zmínku stojí především výrobci TC electronic, Norsonic, SVM tech, Fluke a NL Acoustics. Jejich produkty jsou špičkou v oboru akustické diagnostiky. Jedná se o profesionální přístroje v hodnotě desítek až statisíců korun. To je dáno především finanční nákladností na odladění softwarového zpracování produktů a každá ze jmenovaných firem si svoje produkty snaží vylepšovat na maximální možnou úroveň. Z tohoto důvodu je většina produktů těchto firem málo dostupná a prodávají se na základě komunikace s firmou po vysvětlení konkrétní potřeby zákazníka. Některé firmy dokonce nabízejí možnost přístroj v hodnotě statisíců korun pouze zapůjčit pro měření, případně taková měření sami provádějí.

Využití těchto přístrojů je opravdu široké. Může se jednat o diagnostiku zvuku i vibrací na libovolných strojních zařízeních. Různé aplikace v hudební branži. Zkoumání akustiky otevřených prostor jako jsou hudební sály nebo například tvorby akustických map celých výrobních závodů pro stanovení zdravotních limitů k omezení pobytu pracovníků na nevhodných místech. Dále například v automobilovém průmyslu ke zkoumání hlučnosti pneumatik, nebo zajišťování komfortu uvnitř vozidla.

1 ANALÝZA ZVUKU

Moderní technologie umožňují díky rozvoji výpočetní technologie kvalitní analýzy zvuku. Tato technologie se nazývá akustická diagnostika. Jedná se o bezdemontážní a bezkontaktní způsob zjišťování informací, které může zvuk obsahovat.

Akustická diagnostika je technologie využívaná v široké škále různých oborů, z nichž si uvedeme několik příkladů. Pro hudebníky je užitečné analyzovat zvuk z mnoha důvodů. Příkladem může být sledování spektra tónů pro muzikanty pro zlepšení jejich práce s hudebními nástroji. Dále například hudebníci využívají různých softwarových nástrojů, díky kterým jsou schopni jejich hlas změnit k nepoznání.

Dalším příkladem může být akustika staveb. Tímto se zabývá teorie o šíření zvuku v konstrukcích. Jedná se o téma, které by mohlo zajímat prakticky každého člověka, protože nikdo nechce slyšet každou věc, kterou si sousedé říkají ve vedlejší bytě.

V oboru strojírenství je využití opravdu široké. Především se jedná o způsob identifikace zdrojů hluku na technických zařízeních využitelné pro různé aplikace. Například pro prediktivní údržbu nebo diagnostiku stavu rotačních součástí jako jsou ložiska nebo převodovky.

1.1 Zvuk

Jedná se o mechanické vlnění pružného prostředí s kmitočtem od 20 Hz do 20 kHz. Tento rozsah je dán omezením slyšitelnosti lidského ucha přibližně v 18 letech. Závisí to však na konkrétním sluchovém aparátu, protože každý člověk je individuální.

Jak již bylo zmíněno zvuk vzniká vibracemi (vlněním) pevných látek, které získávají energii působením fyzikálních sil. Takto vyzářená energie se šíří v pružném prostředí, jako je vzduch, voda, dřevo atd. Přitom způsobí kmitání molekul, tedy akustické vlnění. Tyto vyvolané změny tlaku je člověk schopen registrovat, protože je náš sluchový aparát převádí na zvukové vjemy o určité intenzitě a frekvenci.

Vibrace mohou vznikat a není nezbytně nutné, aby přitom byl vydáván zvuk. Nicméně zvuk bez vibrací nemůže vůbec vzniknout vzhledem k jeho fyzikálním vlastnostem. Zvuk se šíří v různých látkách jinými rychlostmi. V pevných látkách je rychlost šíření zvuku větší než u o statních skupenstvích. Například pro ocel je rychlost přibližně 5000 metrů za sekundu. U kapalných látek je rychlost nižší než u pevných, příkladem je například voda, kde se zvukové vlny šíří rychlostí 1500 metrů za sekundu. U plyných látek jsou tyto rychlosti nejnižší. Dáno je to menší hustotou molekul a jejich vazbami pro šíření vln v tomto prostředí. Pro vzduch je přibližná hodnota šíření 340 metrů za sekundu. Ve vakuu se zvuk vůbec nešíří z důvodu absence molekul v tomto prostředí, díky kterým by se mohly zvukové vlny šířit [5].

1.2 hluk

Za hluk můžeme označit každý nežádoucí zvuk, který vyvolává nepříjemný nebo rušivý vjem, případně i poškozuje lidské zdraví, a to jak sluchový orgán, tak i nervovou soustavu. Může vést k akustickému traumatu, stresu, závratím, migrénám a řadě dalších zdravotních komplikací.

Nebezpečné může být i dlouhodobé působení hluku o nízké intenzitě. Tím může být například ventilátor elektronického zařízení jako je počítač, projektor nebo klimatizace. Na tyto vjemy si sice zvykneme, nicméně nervová soustava je nadále vnímá a mozek tedy musí tyto rušivé vjemy filtrovat. Unavuje se tak a snižuje efektivitu práce [6].

Samotná definice hluku je komplikovaná, jelikož nežádoucím může být pro každého člověka jiný zvuk. Přejímáme tedy rozlišení na kvalitativní a kvantitativní stránku hluku. Kvalitativní neboli subjektivní stránka je vzata na základě lidského faktoru. Při jejím hodnocení se uplatní statistické metody. Zkoumáme například, kolika lidem z daného vzorku je daný zvuk nepříjemný apod. Druhou variantou je pak stránka kvantitativní, neboli měřitelná pomocí fyzikálních veličin. Můžeme ji popsat číselnou hodnotou, a díky tomu mohly být zavedeny tzv. hygienické limity hluku, které můžeme najít v konkrétních legislativách. Můžeme se tedy setkat se situací, kdy místo odstranění zdroje hluku přikročíme ke snaze o jeho změnu. Například změnu frekvence, použití akustických materiálů nebo snížení intenzity [1].

Hluk je bezdotykově měřitelná diagnostická veličina a lze jej využít jako zdroj informací o technickém stavu zařízení. Na jeho vytváření u strojních zařízení se totiž podílejí třecí síly generované vzájemným pohybem drsných nebo nepřesně opracovaných ploch v místě styku dílů stroje. Hluk se pak šíří výrazným směrovým účinkem. Nevýhodou zvukové diagnostiky mohou být různé vnější rušení jako hluk okolních objektů, parazitní odrazy nebo interference v uzavřených prostorách uvnitř strojů. Některé specifické zdroje hluku u točivých strojů mohou být: ložiska, převodovky, spalovací motory, kompresory, elektrické točivé stroje a transformátory [4].

K měření hluku se téměř výhradně využívají kondenzátorové mikrofony. Kolísání tlaku vyvolá vydutí membrány, tím změnu kapacity, která se následně elektronicky vyhodnotí. Díky rozvoji výpočetní techniky jsme schopni vytvářet tzv. zvuková pole. Ta umožňují pomocí akustického tlaku v bodech sítě blízké zdroji hluku aproximovat zvuková pole ve vzdálenějších prostorách [2].

1.3 Akustika

Jedná se o široký vědní obor. Zabývá se zvukem v kompletním kontextu od jeho vzniku, přenosu v různých prostředích až po jeho vnímání lidskými smysly a zpracováním pomocí moderních technologií. Měřit můžeme různé veličiny jako je rychlost šíření akustických vln, přenášenou energii, a jejich intenzitu. Výpočty šíření vlnění, jejich odrazu a lomu nebo třeba ohybu, jsou v akustice velmi podobné vzorcům a principům z optiky. Pohybujeme se zde však v jiných vlnových délkách a frekvencích. Akustika se podle jevů, které zkoumá, dělí na akustiku: [6]

- a) Hudební – zabývá se vlastnostmi zvuku z hudebního hlediska.
- b) Fyzikální – zabývá se vlastnostmi a zdroji zvuku z fyzikálního hlediska

- c) Fyziologickou – zabývá se vnímáním a tvorbou zvuku u člověka
- d) Psychoakustiku – zabývá se subjektivním vnímáním zvuku u člověka a vlivem na nervovou soustavu.
- e) Elektroakustiku – zabývá se principem reprodukce a záznamu zvuku v elektrotechnice.
- f) Stavební – zabývá se akustikou stavebních konstrukcí, metody šíření zvuku v prostorech a odhlučnění budov. Stavební akustiku můžeme dělit na tři hlavní směry:
 - Zvukové izolace
 - Prostorová akustika
 - Urbanistická akustika [7]

1.4 Akustické veličiny

Zvuková vlna se v pružných prostředích chová jako vlna tlaková. Dle směru šíření vln rozlišujeme vlny podélné a příčné. Když částice prostředí kmitají ve směru šíření vlny, pak se jedná o vlnu podélnou. Naopak když částice prostředí kmitají ve směru kolmém na šíření vlny, jedná se o příčné vlny. Nutno také podotknout, že příčné vlny se objevují jen v pevných látkách, nikoli u plynů a kapalin. K popsání charakteristik zvuku se pak využívají různé veličiny, z nichž některé budou níže popsány.

1.4.1 Akustický tlak:

Označením jednotky je **p**. Jednotkou je pascal [**Pa**]. Jedná se o skalární veličinu. Udává nám tzv. střídavý tlak v jednom bodě (superponovaný barometrickému tlaku). Tato střídavost znamená, že kolísá kolem střední hodnoty, která je daná barometrickým tlakem. Matematicky se tedy jedná o součet odchylky a střední hodnoty tlaku.

Rozsah hodnot, kterých akustický tlak může dosahovat je velice široký. Můžeme hovořit o rozptylu až 7 řádů. Rozsah je dán pro spodní hodnotu tzv. prahem slyšitelnosti zvuku (přibližně $2 \cdot 10^{-5}$ Pa). Jedná se o nejmenší kolísání tlakové vlny, kterou je lidské sluchové ústrojí schopné registrovat. Na druhé straně horní ohraničení rozsahu nabývá přibližných hodnot $2 \cdot 10^2$ Pa. Překročení těchto hodnot tlaku může znamenat nenávratné poškození sluchu. Tato hodnota pak bývá označována jako práh bolesti. Důležitou poznámkou je také, že akustický tlak je jediná akustická veličina, která může být měřena přímo. Všechny ostatní veličiny měříme a dopočítáváme nějakou nepřímou metodou [6], [8].

1.4.2 Akustická rychlost:

Označení této jednotky je **v** a její jednotkou je [**m*s⁻¹**]. Jedná li se o šíření ve vzduchu, pak tato veličina navazuje na akustický tlak tím, že udává rychlost, jakou dosahuje kolísání molekul vzduchu kolem svojí rovnovážné polohy v důsledku působení změn tlaků. V tomto případě, narozdíl od předchozího, se jedná o vektorovou, energetickou veličinu. Má totiž daný směr a vlnový charakter, díky čemuž ji lze popsat rovnicí pro vlny. Jak již bylo zmíněno, tuto jednotku můžeme měřit pouze nepřímou metodou, a to na základě výpočtu hodnot z dvou různých mikrofonů. Tato jednotka je závislá na teplotě a tlaku prostředí, ve kterém se vlny šíří. V praxi se proto využívají tzv. akustické sondy. Rozsahy měření hodnot akustické rychlosti je opět dán prahem slyšitelnosti a bolesti [6], [8].

1.4.3 Intenzita zvuku:

Označení intenzity zvuku je **I**. Jednotkou této veličiny je $[\mathbf{W} \cdot \mathbf{m}^{-2}]$ Interpretuje velikost akustické energie, která prochází určitou plochou. Tedy udává, jak moc velká energie prochází m^2 plochy. Opět se jedná o nepřímo měřitelnou vektorovou veličinu s vlnovým charakterem a výpočet její hodnoty je dán násobkem mezi akustickým tlakem a akustickou rychlostí, Případně může být vypočtena jako podíl druhé mocniny akustického tlaku a akustického vlnového odporu prostředí [6], [8].

1.4.4 Akustický výkon:

Označením je **P** a jednotka $[\mathbf{W}]$. Akustický výkon popisuje akustické vlastnosti zdroje zvuku. Konkrétně se jedná o množství energie, která je vyzářena do okolí zdroje za daný časový úsek [6], [8].

1.5 Hladiny akustických veličin

Jak již bylo zmiňováno akustické veličiny mohou dosahovat hodnot v širokém rozsahu a to až 7 řádů od prahu slyšitelnosti až po práh bolesti. Z tohoto důvodu je nevýhodné jejich využívání v základním tvaru. Snaha tedy vedla k tomu tyto veličiny převést na lépe zpracovatelné hodnoty. Na základě Weberova-Fechnerova zákona byli zavedeny tzv. logaritmická měřítka pro akustické veličiny. Důvodem byl způsob, jak lidské ucho a mozek vnímají změnu akustických veličin. Lidské ucho totiž nereaguje na zvukový podnět lineárně, ale jeho vnímání je úměrné logaritmu daného podnětu. Jinak řečeno logaritmická funkce je vhodný převodník geometricky vnímaných změn signálu na aritmetickou řadu, například zdvojnásobí-li se akustický signál naše ucho jej bude vnímat jako plynulý přírůstek [6], [8].

Logaritmická vyjádření jednotlivých veličin se nazývají hladiny akustických veličin. Jejich společné označení je písmeno **L** a jako jednotka se v praxi využívá decibel $[\mathbf{dB}]$, tedy desetinná hodnota jednotky Bel [8].

1.5.1 Hladina akustického tlaku L_p :

Je definována vztahem z následující rovnice (2.1)

$$L_p = 10 \log \frac{p^2}{p_0^2} = 20 \log \frac{p}{p_0} \quad [\mathbf{dB}] \quad (2.1)$$

Kde: p [Pa] je efektivní hodnota akustického tlaku,

p_0 [Pa] je referenční hodnota akustického tlaku ($2 \cdot 10^{-5}$) [2], [4].

Dojde-li k desetinásobnému zvýšení akustického tlaku. Pak dojde ke zvýšení hladiny akustického tlaku o 20 dB [8].

1.5.2 Hladina akustické rychlosti L_u :

Je definována vztahem z následující rovnice (2.2)

$$L_u = 10 \log \frac{u^2}{u_0^2} = 20 \log \frac{u}{u_0} \quad [\mathbf{dB}] \quad (2.2)$$

Kde: u [Pa] je hodnota akustické rychlosti,

u_0 [Pa] je referenční hodnota akustické rychlosti ($2 \cdot 10^{-5}$) [2], [4].

1.5.3 Hladina intenzity zvuku L_I :

Je definována vztahem z následující rovnice (2.3)

$$L_I = 10 \log \frac{I}{I_0} \text{ [dB]} \quad (2.3)$$

Kde: I [$\text{W} \cdot \text{m}^{-2}$] je intenzita akustického signálu,

I_0 [$\text{W} \cdot \text{m}^{-2}$] je referenční hodnota intenzity zvuku (10^{-12}) [2], [4].

1.5.4 Hladina akustického výkonu L_W :

Je definována vztahem z následující rovnice (2.4)

$$L_W = 10 \log \frac{W}{W_0} \text{ [dB]} \quad (2.4)$$

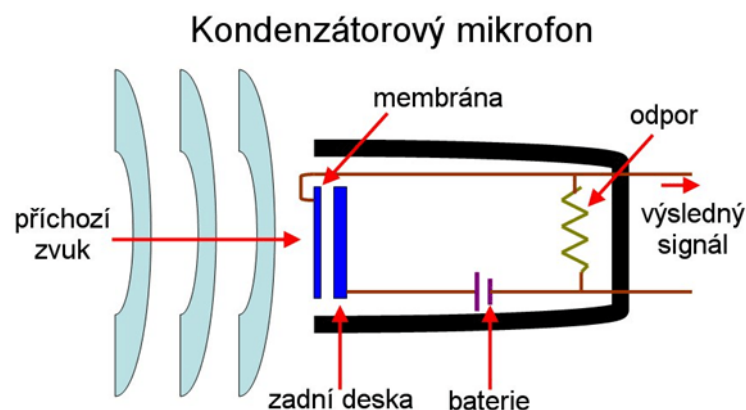
Kde: W [W] je akustický výkon,

W_0 [W] je referenční hodnota akustického výkonu (10^{-12}) [4], [2].

1.6 Mikrofon

Jedná se o hlavní součást zvukoměrných soustav, tzv. akusticko-mechanicko-elektrický měnič. Má přímý kontakt s prostředím, ve kterém měříme. Je využíván k zachycení zvukového signálu, tedy snímání kmitání vnějšího plynného prostředí a převádění těchto mechanických kmitů na elektrické veličiny. Tím zprostředkovává přenos akustické energie na elektrický signál, který pak můžeme dále zpracovávat. Elektrická odezva na akustické podněty by měla být v širokém pásmu kmitočtů, které jsme si vysvětlovali v kapitole akustických veličin. Kvůli tomu musí být napětí zpracovaných signálů dostatečně velké a ve velkém dynamickém rozsahu. Mikrofon by také měl být zajištěn proti změně svých vlastností v závislostech na vnějších statických podmínkách jako je teplota, tlak nebo vlhkost [6], [8].

Mikrofony, které jsou využívány na měření akustického tlaku, se vyznačují velkou, a navíc frekvenčně závislou směrovou citlivostí, a to v různých úhlech od osy. Pro nízké kmitočty bývá mikrofon téměř všesměrový. V akustické diagnostice však většinou potřebujeme lokalizovat zdroj zvuku a tím i místo případné závady. Pro použití takového mikrofonu ve volném zvukovém poli je nutné zařídit, aby vlny dopadaly na plochu membrány kolmo, tedy pod úhlem 90° [4].



Obrázek 1

Schéma kondenzátorového mikrofonu [12]

Dle toho, jakým způsobem dochází k převodu akustického tlaku na napětí, rozlišujeme několik druhů mikrofonů.

Piezelektrické se vyrábějí z piezelektrických krystalů, které na základě své mechanické deformace vytvářejí odpovídající elektrické napětí. Jsou značně odolné vůči vlhkosti a teplotě. Další často využívaný typ jsou dynamické mikrofony, které se vyrábějí v cívkovém, či páskovém provedení. Tímto páskem je pružný hliníkový úsek, který se využívá jako membrána. Principem je zde indukce napětí při pohybu vodiče v magnetickém poli. Dynamické mikrofony dosahují vysoké citlivosti a jsou využívány v těžkých provozech [8].

Nejpoužívanějším typem mikrofonu je elektrostatický neboli kondenzátorový. Je to dáno jeho jednoduchým provedením, lehce dosažitelnou lineární kmitočtovou charakteristikou v rozsahu celého slyšitelného spektra kmitočtů a jeho stálou citlivostí. Základním principem fungování je lehká membrána, která se dobře poddává akustickému kmitání vnějšího prostředí. Tím mění vzdálenost membrány, jako jedné elektrody, a pevné elektrody. Změnou vzdáleností elektrod je vytvořena změna kapacity na vzduchovém dielektriku, což na kondenzátoru generuje střídavé napětí. Toto napětí je pak přímo úměrné akustickému tlaku v širokém pásmu kmitočtů a ve velkém dynamickém rozsahu. Jeho nevýhodami jsou vyšší cena a malá mechanická odolnost. Na obrázku 1 můžeme vidět schéma kondenzátorového mikrofonu s jeho základními částmi [6], [12].

V této práci je využívána hardwarová deska s integrovaným mikrofonem technologie MEMS. Ve zkratce se jedná o spojení elektroakustického měniče a integrovaného AD převodníku pro převod na digitální výstup. Z důvodu větší přehlednosti bude tato technologie více vysvětlena v kapitole 3.1.2, která se věnuje konkrétním periferiím využitých v této práci se všemi vlastnostmi i požadavky.

1.6.1 Vlastnosti mikrofonů

- Směrnost

Velmi důležitým požadavkem na mikrofony jsou jeho směrové charakteristiky. Mikrofony pro měření v uzavřeném zvukovém prostoru mohou mít kulovou, tedy všesměrovou charakteristiku. Na druhou stranu pro použití na volných prostranstvích bývá vhodnější, když mikrofon snímá zvuk pouze z určitých směrů. Tohoto můžeme dosáhnout různými konstrukčními uspořádáními. Tyto charakteristiky mohou být řešeny také v rámci tzv. typu měřeného pole. Můžeme odlišit pole volné, difúzní a tlakové [8].

- Citlivost

Další vlastnost důležitá pro výběr zařízení je citlivost. Velká citlivost nám umožní lepší rozlišitelnost na menších intenzitách díky velkému rozsahu změny elektrické veličiny v závislosti na změně tlaku. Tento rozsah elektrických veličin zároveň umožňuje jednodušší zpětný převod z napětí na akustický tlak, jež působil na membránu mikrofonu [8].

- Frekvenční charakteristika

V závislosti na druhu snímaného signálu vybíráme mikrofony s různými frekvenčními rozsahy. Je dán konstrukčním provedením zařízení a udává rozsah ve kterém je mikrofon schopen měřit. U kondenzátorových mikrofonů je důležité polarizační napětí kondenzátoru. Může být externě polarizovaný případně před-polarizovaný, kterému již není třeba externě dodávat napětí [6].

- Maximální akustický tlak SPL

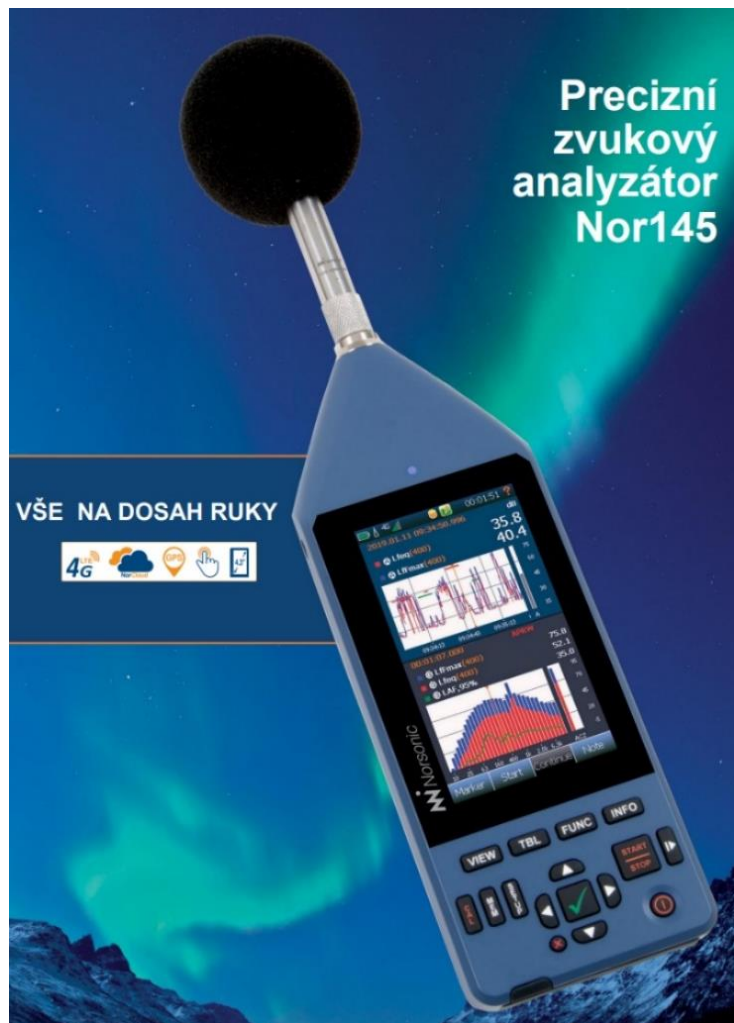
Zkratka SPL znamená v doslovném překladu maximální hladina akustického tlaku. Nejedná se však o maximální tlak, od kterého se mikrofon rozbije, nýbrž o hodnotu, od které začíná mikrofon slyšitelně zkreslovat přijímaný signál. Za počátek zkreslení, které již začíná být slyšitelné se považuje celkové harmonické zkreslení okolo 0,5% [8].

1.7 Zvukoměr, zvukový analyzátor

Zvukoměr je zařízení, které se využívá pro měření a zobrazení hladiny akustického tlaku, všechny zvukoměry obsahují mikrofon, zpracování signálu a indikační zařízení. Hlavní požadavky na přístroje jsou stálost měřených dat, přesnost měření a nenarušování měřeného pole [4].

Běžně se při komunikaci využívá také pojem hlukoměr. Jedná se pouze o jiné označení stejného přístroje, rozdíl je pouze v tom, v jakém odvětví a skupině lidí se tento přístroj využívá. Dříve se v akustické branži odlišoval pojem zvukoměr a zvukový analyzátor. Jak jsme si již řekli, tak zvukoměr slouží pouze pro akustická měření a zobrazení hodnoty na displej. Zvukový analyzátor disponuje také možností daný zvuk analyzovat například frekvenčně pomocí Fourierovy transformace a tím zjistit a zobrazit další potřebné informace o akustických veličinách. V dnešní době se tyto pojmy již tak striktně nerozlišují z důvodu velkého rozmachu výpočetní techniky, která disponuje stále větším výpočetním výkonem. Díky tomu se na trhu objevují varianty zvukoměrů za přijatelnou cenu i se zabudovanými funkcemi pro analýzu zvuku. Jednotlivé přístroje se pak od sebe mohou lišit dle konkrétního provedení výrobce. Můžeme je dělit například podle počtu měřících kanálů, formy výstupu, podpory navazujících software řešení, nebo typu konstrukce [6], [9].

Názorným příkladem může být přístroj na obrázku 2. Jedná se o zvukoměr firmy Norsonic, která se zabývá výrobou zvukoměrné techniky. Tento přístroj umožňuje naměřené hodnoty akustických veličin rovnou analyzovat. Zároveň disponuje dalšími užitečnými funkcemi jako je připojení ke 4G síti díky čemuž může zpracovaná data ihned nahrát například na cloudové úložiště a tím data ihned zobrazovat a využívat na vzdálených pracovištích od místa měření. Případně můžeme analyzovaná data ihned vyčítat na dotykovém, velkém displeji [10].



Obrázek 2 Zvukoměr/zvukový analyzátor Nor 145 [10]

1.7.1 Kalibrace zvukoměrů:

Pro zobrazování správné hodnoty hladiny akustického tlaku je třeba provádět na těchto přístrojích kalibraci. Kalibrace spočívá v nalezení koeficientu pro sjednocení efektivní hodnoty elektrického napětí, případně číslkové hodnoty v digitálních zvukoměrech s hodnotami hladin akustických veličin. Provádí se pomocí pistonfonu, případně tónového kalibrátoru [11].

Pistonfon je tvořen válcovou těsnou komorou, do které je vložen mikrofon kalibrovaného přístroje. V komoře se pak nachází píst, který pomocí kmitavého pohybu vytváří daný akustický tlak o známé frekvenci. [11].

Tónový kalibrátor je zařízení, které za pomoci reproduktoru v dutině, do které je vložen mikrofon kalibrovaného hlukoměru, generuje zvukový signál o kmitočtu 1000 Hz s hladinou akustického tlaku 94 dB [11].

Existuje také méně přesná metoda kalibrace než předchozí dvě. Jedná se o porovnávání hodnot nekalibrovaného zvukoměru s hodnotami, které ukazuje již kalibrovaný zvukoměr. Toto porovnání musí proběhnout v bezodrazové místnosti, ve které je generován zvuk o daném kmitočtu [11].

1.8 Akustická holografie

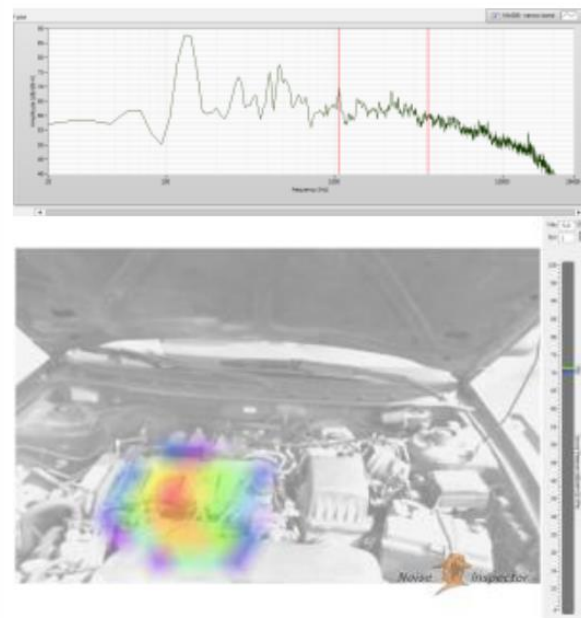
Jedná se skupinu různých metod sloužících pro rekonstrukci akustických veličin náležejících k objektu ve třírozměrném prostoru na základě měření ve vhodných rovinách. Prakticky můžeme za pomoci intenzitních sond nebo mikrofonů identifikovat zdroj a intenzitu vibrací objektu na jeho povrchu. Důvodem rozvoje těchto metod je, že se jedná o bezdotykovou metodu, která nijak neovlivňuje měřený objekt. Velkou výhodou je to především u málo hmotných objektů, u kterých by přidělení snímače vibrací mohlo mít za následek velkou nepřesnost měření [4].

1.9 Akustické kamery

Toto zařízení využívá spojení více mikrofonů v jedno tzv. mikrofonní pole. Při spojení s videokamerou, která je uložena uprostřed mikrofonního pole, je zařízení schopné zobrazit na monitoru počítače záznam videokamery spolu s výsledky výpočtu akustických polí. Jedná se tedy o velice silný diagnostický nástroj. Díky tomu lze toto zařízení využívat v široké škále různých oblastí. Některá možná praktická užití si vyjmenujeme pro různá odvětví. Ve strojírenství můžeme například identifikovat zhoršený stav ložisek rotačních součástí, případně využít pro prediktivní údržbu ve spojení s dalšími diagnostickými nástroji jako je termodiagnostika nebo vibrodiagnostika. V automobilovém průmyslu pomáhá vyhledávat závady na motoru viz. obrázek 3, nebo naopak sledovat dokonalost odhlučnění kabin vozů a slabá místa tohoto odhlučnění. Ve stavebnictví můžeme zase vyhledávat slabá místa odhlučnění budov s přesahem například k hudebníkům, kteří tuto technologii mohou využívat k dobrému nastavení ozvučení místností nebo velkých hudebních sálů [13].



Měření motoru automobilu



Ukázka výsledků analýzy

Obrázek 3

Ukázka měření akustickou kamerou [13]

1.9.1 Mikrofonní pole

Jedná se o nezbytnou součást zvukoměrů určených na určování směru zdroje akustického signálu. V závislosti na konstrukci je můžeme dělit na 3 typy.

- Lineární
- Rovinné
- Prostorové

Pole umožňuje rozpoznat rozložení akustického signálu v prostoru. Obecně se dá říct, že čím více mikrofonů tím bude měření přesnější [27].

1.9.2 Metody pro určení směru příchodu signálu

Akustický signál naměřený pomocí mikrofonního pole je pro lokalizaci zdroje nutné zpracovat některou z metod pro určování směru příchodu. Nejpoužívanějšími typy těchto metod jsou: [27]

- Spektrální odhad vysokého rozlišení

Jedná se o metody, které fungují na principu spektrální analýzy vysokého rozlišení. Příklady metod tohoto druhu jsou: odhad minimálního rozptylu (MV), autoregresivní modelování (AR) a vícenásobné klasifikace signálu (MUSIC). Všechny tyto metody umožňují lokalizovat více zdrojů najednou. Jedním z hlavních rozdílů pro použití je schopnost snímat blízké nebo vzdálené zdroje. Metody MV a MUSIC umožňují obě varianty. Metoda AR umožňuje využití pouze pro zdroje vzdálené [27].

Spektrální odhady se vyznačují vysokou přesností lokalizace při dvou zdrojích blízko u sebe, kdy odezva vykazuje ostřejší hranici mezi signály na rozdíl od metod tvarování přijímací charakteristiky, kde se dva zdroje blízko u sebe mohou lehce zaměnit za jeden [27].

- Odhad časových zpoždění

Tato metoda vykazuje na rozdíl od ostatních zmiňovaných menší potřebu na výpočetní výkon, díky tomu je v praxi hojně využívána. Nevýhodou však je možnost lokalizovat pouze jeden zdroj zvuku, a také vysoká náchylnost na rušení dozvuky. Můžeme ji najít také pod označením dvoukroková, vyplývá to z principu určování polohy zdroje rozděleného do dvou kroků [27].

Při prvním se odhadují časová zpoždění signálu sousedících mikrofonů. Na to se využívají různé výpočty, například křížová korelace, která určuje vzájemnou podobnost signálů. Ta je pak vyhodnocena na základě tzv. normovaného koeficientu křížové korelace. Ten se pohybuje v intervalu od mínus jedna do jedna a pro naprosto totožné signály je roven jedné. Při druhém kroku je pak z odhadnutého časového zpoždění vypočtena poloha zdroje na základě známé geometrie mikrofonního pole [27].

- Tvarování přijímací charakteristiky

Metoda tvarování přijímací charakteristiky je na rozdíl od odhadu časových zpoždění výpočetně velmi náročná. To znemožňuje nevyužívání na lokalizaci pohybujících se zdrojů zvuku. Výhodou je možnost rozeznání více různých zdrojů. Zařízení na určování poloh signálů se nazývá tvarovač neboli beamformer. Jeho fungování můžeme opět rozdělit do dvou kroků. V prvním kroku se provádí synchronizace signálu. V druhém kroku se pak provádí váhování signálu, sumace a finálním výpočet polohy zdroje [27].

V kroku synchronizace je nutné signály ze všech přijatých signálů zaostřit do jednoho zkoumaného bodu. To se provede zpožděním signálů přichozích z jednotlivých mikrofonů o vypočtený čas. V literatuře se jedná o pojem vnučování směru pohledu tvarovači. Tvarovač by totiž měl vykazovat největší zisk právě pokud jsou signály z jednotlivých mikrofonů zarovnány bez vzájemného posunutí [27].

Druhý krok váhování a sumace přiděluje váhu signálům ze všech mikrofonů a ty potom sčítá. Toto má vliv na finální šířku hlavní smyčky a postranních smyček na směrové charakteristice daného pole mikrofonů. Postranní smyčky bývají vytvořeny z důvodu překročení mezních frekvencí [27].

Nejjednodušším tvarovačem, který se v praxi využívá je tzv. tvarovač typu delay and sum. Jedná se o princip, kde sami určujeme, jaký zkoumaný bod budeme měřit pomocí nastavení časového posunutí signálů z mikrofonu o námi vypočtenou hodnotu. V dalším kroku opět sečteme časově posunuté signály pro získání výstupu tvarovače. Tyto kroky můžeme libovolně opakovat pro jakýkoli zvolený zkoumaný bod, a tím projít klidně celé zorné pole mikrofonního pole [27].

2 ALGORITMY PRO ANALÝZU A ZPRACOVÁNÍ DIAGNOSTICKÝCH SIGNÁLŮ

Zvukové kmity pomocí mikrofону měníme na signál elektrický. Tento signál je spojitý a abychom jej mohli zpracovat výpočetní technikou v číslicovém systému musíme jej do příslušného tvaru převést. V této práci bude využíván vývojový kit Sipeed, který bude popsán v další kapitole. Tato deska k tomuto převodu využívá formu převedení do číselné posloupnosti, tzv. digitalizaci. Jedná se o analogově-číslcový převod prováděný na tzv. AD převodníku. Takto vytvořený signál tvoří z matematického hlediska posloupnost. Operace vedoucí k těmto posloupnostem jsou vzorkování v čase a kvantování v amplitudě. Při tomto procesu je důležité zabránit jevu zvanému aliasing, ten vzniká, pokud je frekvence odebírání vzorků příliš malá v porovnání s maximální frekvencí signálu. Takto připravený signál již lze jednoduše dále zpracovávat pomocí různých algoritmů, které budou dále osvětleny [2], [4].

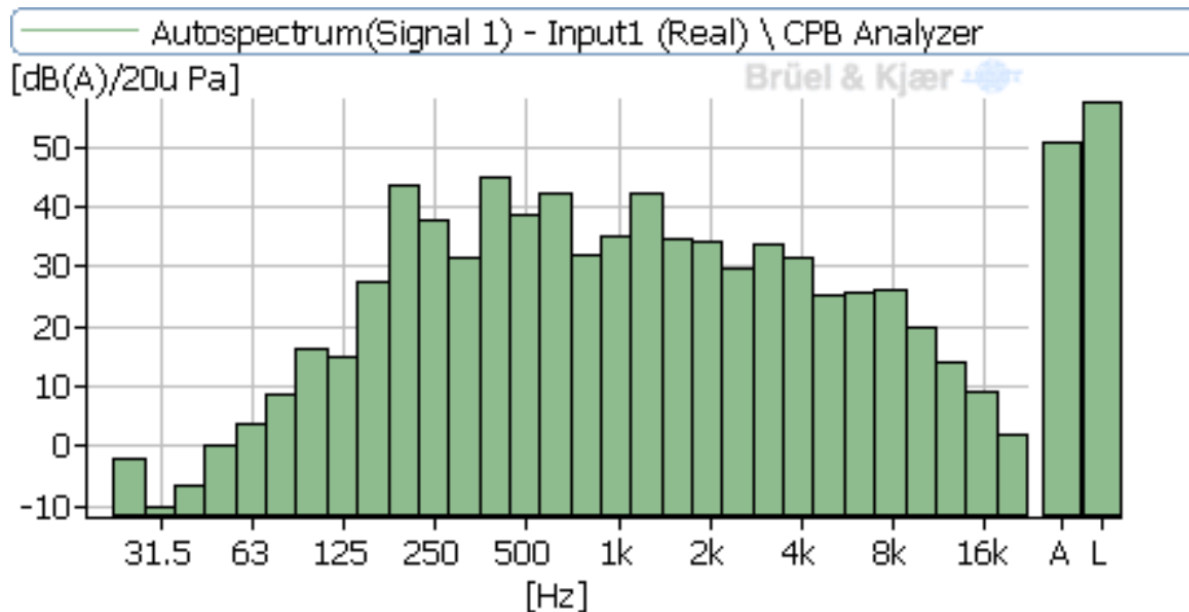
2.1 Frekvenční analýzy

Signál, který snímáme, můžeme nazvat jako souhrnnou charakteristiku (například hladina hluku). Tato charakteristika nám udává informace o energetickém obsahu signálu v závislosti na čase. Pro zpracování signálů z časové oblasti do frekvenční je výhodné využít některou z frekvenčních analýz. Jsou to důležité nástroje pro diagnostiku zvuku, obzvláště pak pro identifikaci jeho zdroje. Umožňují nám zobrazit frekvenční spektrum (rozklad na jednotlivé frekvenční složky). Spektrum představuje velmi zjednodušeně podíl jednotlivých takzvaně čistých tónů ve zvuku. Jinak řečeno, kolik harmonických složek je v signálu. Jedná se o funkce závislé na frekvenci. Díky tomu tedy můžeme zkoumat energetickou hladinu v závislostech na frekvencích, které potřebujeme znát. Případně můžeme zkoumat vliv na zvukový vjem při konkrétních známých frekvencích. Rozklad signálů na zmíněné frekvenční složky provádí analyzátor. U dřívějších analogových přístrojů se využívalo analogových filtrů o daném frekvenčním rozsahu. Dnes již využíváme filtrů digitálních. Vzájemný vztah mezi šířkami dílčích pásem frekvenčního rozsahu spektra nám udává dvě hlavní kategorie analyzátorů. Analyzátoři, jež charakterizuje logaritmická frekvenční osa jsou které mají konstantní relativní šířku pásma. Druhými jsou analyzátoři určeny konstantní absolutní šířkou pásma, které charakterizuje lineární frekvenční osa [2], [6].

Podle potřeb konkrétního měření volíme vhodný typ analýzy, nejpoužívanější jsou popsány dále v této kapitole.

2.2 CPB analýza

Jedná se o analýza s konstantní relativní šířkou pásma, která je charakterizována logaritmickou frekvenční osou. Tato analýza odpovídá velmi přesně fungování lidského sluchu. Využívá se tak hlavně v akustice, při měření hluku, případně ke sledování přechodových jevů a identifikaci záznějů. Šířka pásma bývá stejná pro celé měření. Střední pásem leží na frekvencích oktávy tedy na dvojnásobném kmitočtu. Oktávová analýza je příliš hrubá, používáme proto jejich podíly 1/3, 1/6, 1/12 [2], [6], [14]. Příklad zpracované analýzy CPB můžeme vidět na obrázku 4.



Obrázek 4 Příklad CPB analýzy [14]

2.3 Řádová analýza

Jedná se o speciální typ frekvenční analýzy, která využívá FFT. Podstatou je provedení úpravy naměřeného časového signálu. Pomocí dvoukanalového měření, kdy první kanál slouží k měření diagnostických veličin a druhý se využívá k synchronizaci počátku a konce záznamu zajistí, aby počet vzorků pro analýzu byl vždy stejný bez ohledu na otáčky stroje. Eliminuje tedy skluz otáček. Výsledné spektrum má místo frekvenční osy násobky otáčkové frekvence. Díky tomuto se tato analýza hodí k měření rotujících součástí, u kterých dochází ke změnám otáček. Jedná se například o převodovky nebo pohonné systémy vozů [14].

2.4 FFT analýza

Takzvaná rychlá Fourierova transformace (dále může být použito pouze „FFT“) je jedna z nejvyužívanějších numerických metod pro spočtení Fourierovy transformace. Využit se dá v různých oborech. Jedná se o efektivní algoritmus pro spočtení diskrétní Fourierovy transformace a její inverze. Řadí se do skupiny frekvenčních analýz s konstantní absolutní šířkou pásma. S výhodou ji využíváme k identifikaci zdroje hluku, protože umožňuje zobrazení amplitudy signálu na přesně dané frekvenci. Známe-li tedy frekvence jednotlivých částí stroje jako je například motor, můžeme měřit stroj v celém frekvenčním spektru a izolovat signály na známých frekvencích bez nutnosti demontáže, nebo jiné separace této části stroje [3], [6].

Důležitou vlastností diskrétní Fourierovy transformace je spektrální rozptyl. Jedná se o to, že ve zpracovaném signálu se začnou objevovat frekvence, které v původním signálu nebyly, proto výsledné spektrum neodpovídá spektru původně měřeného signálu. Tento jev vychází ze základní vlastnosti transformace o požadavku na nekonečnou dobu záznamu signálu. FFT algoritmus tedy předpokládá, že se časový záznam sledovaného signálu přesně opakuje po celou dobu trvání signálu. Jedná se tedy o předpoklad, že časový záznam je celý násobek period původního signálu. Není-li tomu tak a délka záznamu je desetinného násobku periody původního signálu tak tento algoritmus vytvoří v signálu tyto nespojitosti. Předcházet tomuto

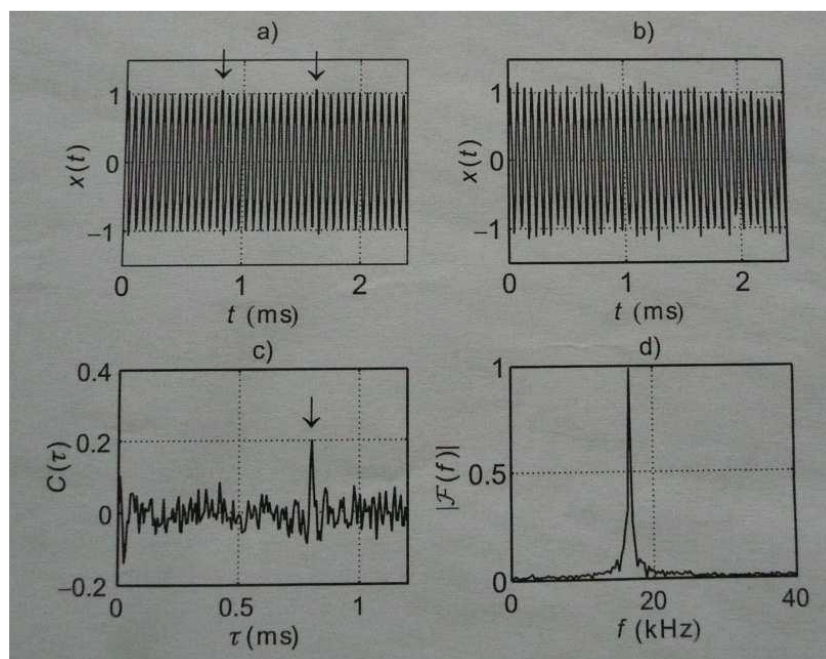
jevu pomáhá tzv. synchronní vzorkování. Toto se využívá především u periodických signálů. U neperiodických signálů se pak využívá funkce windowing. Ta funguje tak, že se časový záznam násobí různými časovými okny, které mají tvar vymyšlený tak, aby rozptýl co nejvíce potlačily [3].

Tato analýza je s výhodou využívána v rámci této práce z důvodu její integrace do vývojového prostředí, které mi bylo zapůjčeno a které bude představeno v další kapitole.

2.5 Kepstrální analýza

Název tohoto algoritmu byl vytvořen přesmyčkou prvních písmen slova spektrum. Tato analýza signálu funguje na základě rychlé Fourierovy transformace ovšem na logaritmické spektrum. Osa y bývá na grafech vyjadřována v decibelech. Vodorovná osa x nabývá rozměru času a je označována jako kvefrencce, tedy opět označení vytvořené přesmyčkou slova frekvence. Přednostní vlastností kepstrální analýzy je citlivost na ekvidistantně rozložené složky spektra signálu. Například pro projevy v postranních pásech při amplitudové modulaci, nebo na projevy harmonických skupin.

Pro lepší představu je využití kepstra znázorněno na snímku 5. V části a) můžeme pozorovat kepstrum harmonického signálu s frekvencí 16,8 kHz. Jeho amplituda je modulována obdélníkovým signálem o periodě 0,8ms. Tato modulace je na obrázku zvýrazněna šipkami. V části obrázku b) pozorujeme totožný signál, který ovšem obsahuje přidání normálně rozložený bílý šum. V tomto šumu již není patrná modulace signálu. Ve třetí části obrázku c) pozorujeme kepstrum zašuměného signálu, jehož modulační kmitočet se jasně projevuje špičkou na kvefreci, která odpovídá periodě modulace. V poslední části d) je vyobrazeno amplitudové spektrum zašuměného signálu, ve kterém nejsou zřetelná postranní pásma amplitudové modulace [4].

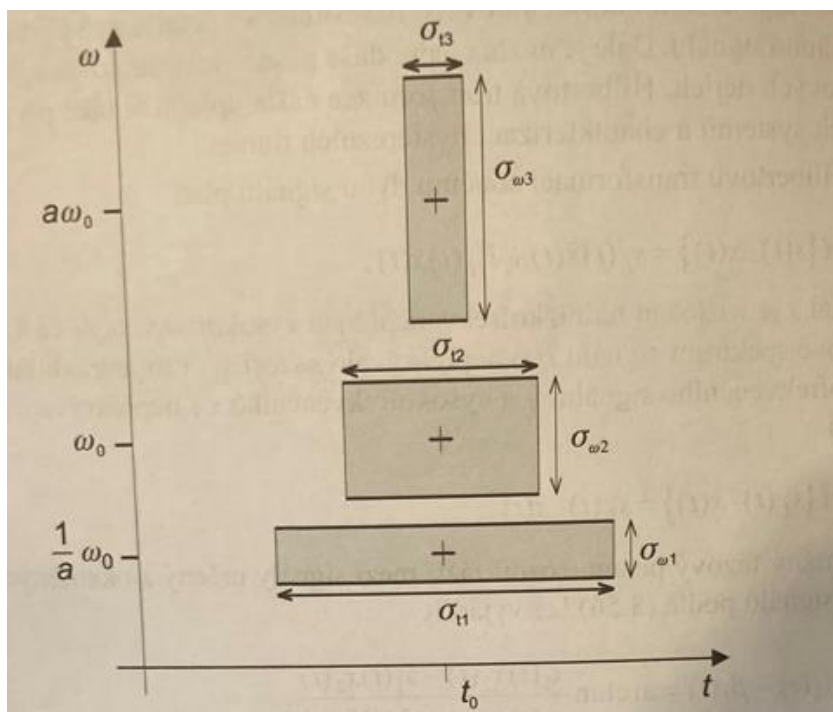


Obrázek 5 Grafické znázornění kepstrální analýzy [4]

2.6 Vlnová (wavelet) transformace

Vlnková transformace vznikla jako výsledek snahy o získání časově-frekvenčního popisu signálu. Jinak řečeno poskytuje informaci o tom, které frekvence se v signálu nacházejí a zároveň i v jaké poloze, tedy v jakém čase. Tato transformace je integrální [4].

Principem algoritmu je použití okna. To v čase ohraničí krátký úsek signálu a umožní z něj určovat spektrum v tomto daném, ohraničeném časovém intervalu. (short-time fourier transform, gaborova transformace). Princip neurčitosti nám však udává, že nelze současně určit přesně jak frekvenci, tak polohu jejího výskytu v čase. Proto při využití předchozích metod mezeru. Při užití časově konstantního širokého okna pro všechny kmitočty velkou rozlišitelnost ve frekvenci, ale malou v čase. Naopak při užití časově úzkého okna pak dosáhneme velkou rozlišitelnost v čase, avšak malou ve frekvenci. Toto je zobrazeno na snímku 6. Hlavní myšlenkou je pak vhodnou změnou šířky okna v čase dosáhnout optimálního poměru rozlišitelnosti v čase i frekvenci [4].



Obrázek 6 Rozlišitelnost v čase a frekvenci [4]

3 PLATFORMA SIPEED A SOFTWARE

V této kapitole se blíže seznámíme s předloženým vývojovým kitem od společnosti Sipeed, který byl k vypracování této diplomové práce využit.

Tato společnost vznikla v Číně. Zabývá se tvorbou hardware a software pro použití v několika různých odvětvích. Hlavní orientací je tzv. AIoT. Jedná se o spojení technologií „AI“ tedy umělé inteligence a „IoT“ internetu věcí. Tato metoda se zaměřuje na zlepšení vztahu člověka a stroje spolu se systematickým sběrem dat a jejich analýz. Na této myšlence vytvořila celý vlastní ekosystém. Ten zahrnuje kromě dále představených částí také cloud servis a kompletní technickou a servisní podporu [16].

Nyní budou popsány některé části této platformy a zapůjčená deska spolu s konkrétními perifériemi, které byly využity pro tuto práci. Bude také popsáno softwarové řešení programu, kterým je využit pro komunikaci s deskou. Zmíněna budou i další externí zařízení použita v této práci, jako je vývojová deska od firmy Arduino využita pro komunikaci s okolím.

3.1 Maix Bit v2.0

Jedná se o druhou verzi vývojové desky Bit, která byla součástí produktové řady Maix. Tento hardware je vhodný pro tuto práci především díky přidanému mikrofonu, který první verze neměla. Na tomto mikrofonu byly prováděny první kroky při plnění cíle seznámení se s možnostmi využití různých algoritmů pro zpracování analýzy zvuku.

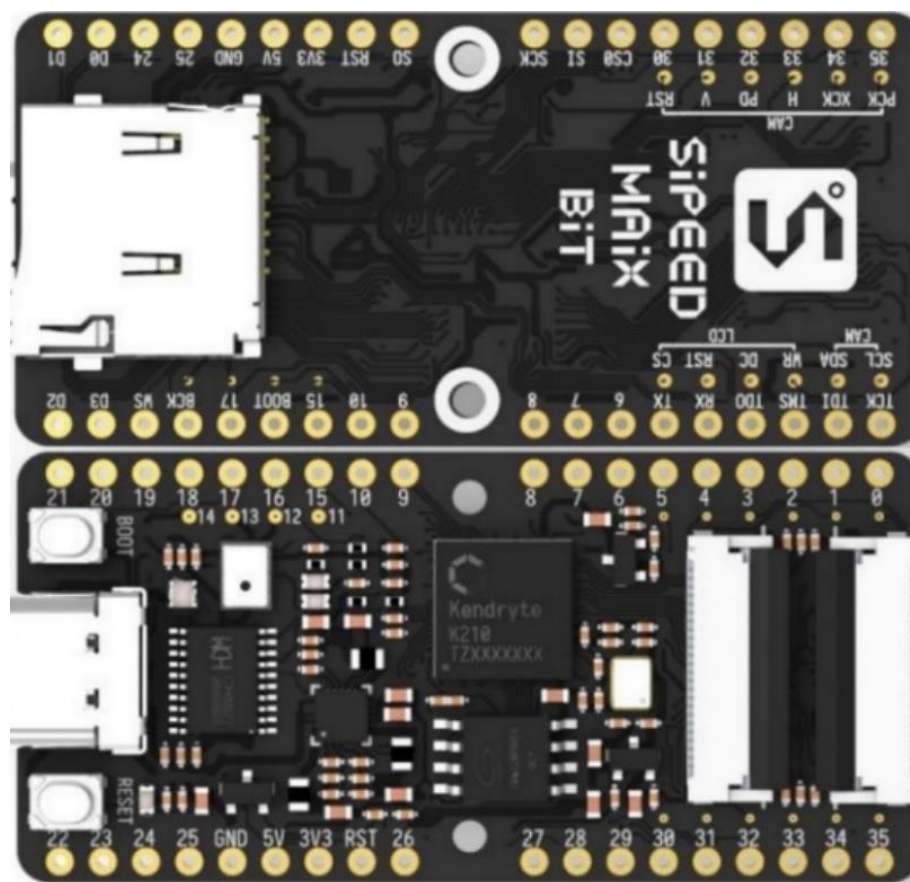
Celá deska je postavena na mikročipu, kterému bude věnován samostatný odstavec. Důležitými perifériemi jsou především USB-C rozhraní pro komunikaci s PC a dalšími zařízeními pomocí zabudované UART (Universal asynchronous receiver-transmitter) sběrnice k asynchronnímu sériovému přenosu dat. Na desce se nachází také sběrnice I²S (Inter Sound). Ta slouží pro komunikaci mezi mikrofonem a čipem [19].

Před prvním použitím tohoto hardware je nezbytné aktualizovat řídicí software pomocí příslušného ovladače z oficiální stránky výrobce. Toto nastavení umožní využití všech dostupných možností ovládání v rámci konkrétního typu desky. K zápisu tohoto nastavení slouží externí aplikace kflash_gui.

Deska nabízí širokou škálu možností práce s daty. Můžeme využít rozhraní pro čtení a zápis na MicroSD karty na zadní straně desky, které můžeme vidět na obrázku 7. Zpracovávaná data nebo výsledky měření mohou být promítnuty pomocí připojeného LCD displeje. Informace také můžeme promítat pomocí již zmíněného USB-C na počítač a zobrazovat v software Maixpy, kterému bude věnována následující kapitola. Rozhraní pro zpracování zvuku ale umožňuje i data přijímat, proto může být připojena i externí kamera pro využití v aplikacích jako je rozpoznávání obličejů za pomoci zmiňovaného výkonného čipu.

Modulový přístup této desky umožňuje připojení různých externích rozšíření, které firma také nabízí. Příkladem může být Wifi adaptér pro sdílení dat online, nebo mikrofonní pole pro možné rozšíření měření. Velké variabilitě přispívá i možnost naprogramování přiřazení jednotlivých číslovaných pinů desky na různá využití podle vlastních potřeb [19].

Využití může být opravdu široké jako například chytré domácnosti, chytré vzdělávací a logistické aplikace, roboti na úklid, diagnostika ve strojírenství, medicíně, stavebnictví nebo třeba i zemědělství.



Obrázek 7 Vývojová deska Maix Bit v2.0 [20]

3.1.1 Čip Kendryte K210

Čip je tvořený pro technologie zaměřené na rozpoznávání pomocí práce s AIoT. Obsahuje několik zabudovaných funkcí pro Analýzy na základě FFT algoritmů. První je akcelerátor výpočtu FFT, který zrychluje a zpřesňuje matematické výpočty potřebných hodnot. Dále má zabudovanou podporu komunikace a zpracování dat z mikrofonního pole. Tímto se daný algoritmus stává nejlepším řešením pro tuto desku ze všech představených algoritmů v předchozí kapitole této práce. Díky těmto vlastnostem je periferie vhodná pro využití v diagnostických nástrojích.

Mezi důležité výhody tohoto čipu patří kromě silné podpory FFT a zmiňované podpory mikrofonních polí také nízká spotřeba energie. Firma, která čip vyrábí pochází také z Číny a jmenuje se Canaan ldt. Dle jejího tvrzení dokáží díky využití offline databázi pro práci s daty ve spojení se SRAM pamětí dosáhnout takřka třetinové spotřeby oproti běžným konkurenčním čipům se stejným zaměřením. Výsledná udávaná průměrná spotřeba čipu se pohybuje na 0,3 W což umožňuje využít tento čip v široké škále aplikací.

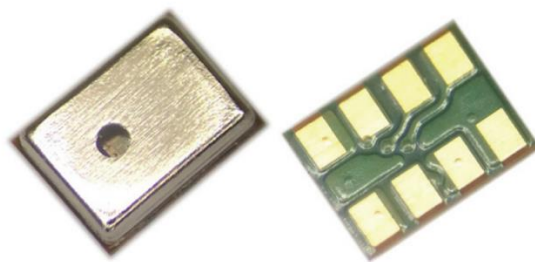
Programován může být v jazycích C, C++, Micropython a v dalších odvozeninách tohoto jazyka [17].

3.1.2 Mikrofon MSM261S4030H0

Na následujícím obrázku 8 můžeme vidět vzhled mikrofonu typu MEMS integrovaného na desce Maix Bit a mikrofonním poli. Jedná se o zařízení od firmy Sensing Microsystems Co., Ltd. Komunikující na zmiňovaném digitálním rozhraní I²S s čipem desky. Jedná se o všesměrový mikrofon s vysokým výkonem a spolehlivostí.

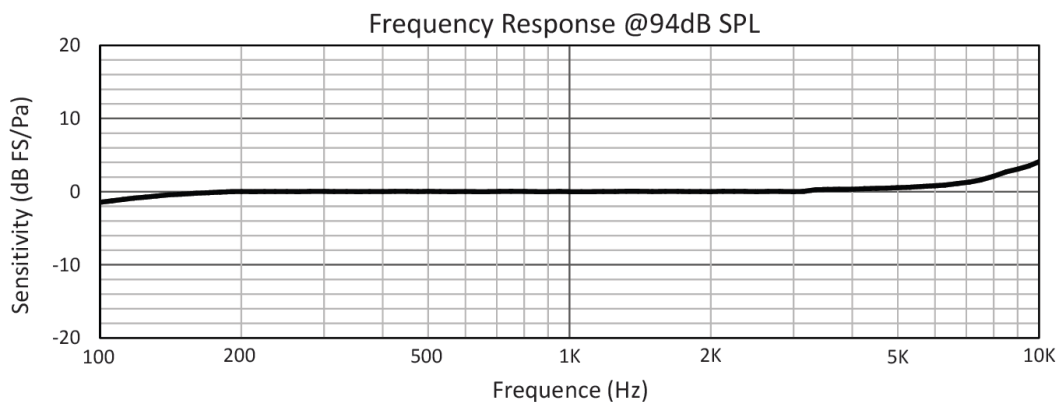
Jeho citlivost dosahuje rozmezí 26 dB (při 1 kHz a 1 Pa). Další důležitou vlastností je poměr mezi silou signálu a snímaným šumem nazývaný S/n ratio jehož nominální hodnota je 57 dB. Maximální měřitelná hladina tlaku je 140 dB, při napětí v rozmezí od -0,3 do 4 voltů. Pracovat dokáže v rozmezí teplot od -40 °C do 100 °C. Velice důležitou charakteristikou je také Frekvenční charakteristika. Tento zvukový snímač je schopen pracovat na frekvencích od 100 Hz do 10 kHz. Což vychází z obrázku 9. Můžeme zde vidět, že v rozmezí 100–200 Hz a 5–10 kHz se nachází místa s malou sensitivitou, to může znamenat výraznou chybu při měření.

Jeho výhodami jsou nízká cena, nízká spotřeba energie a digitální výstup. Ty umožňují široké využití například v mobilních telefonech, tabletech, sluchátkách a aplikacích jako jsou vývojové desky pro zařízení pracující se zvukem. Na obrázek 7 můžeme vidět umístění mikrofon na desce Maix Bit pod piny číslo 16 a 17 [22].



Obrázek 8 Mikrofon MSM osazený na desce Maix Bit a mikrofonním poli MicArray [22]

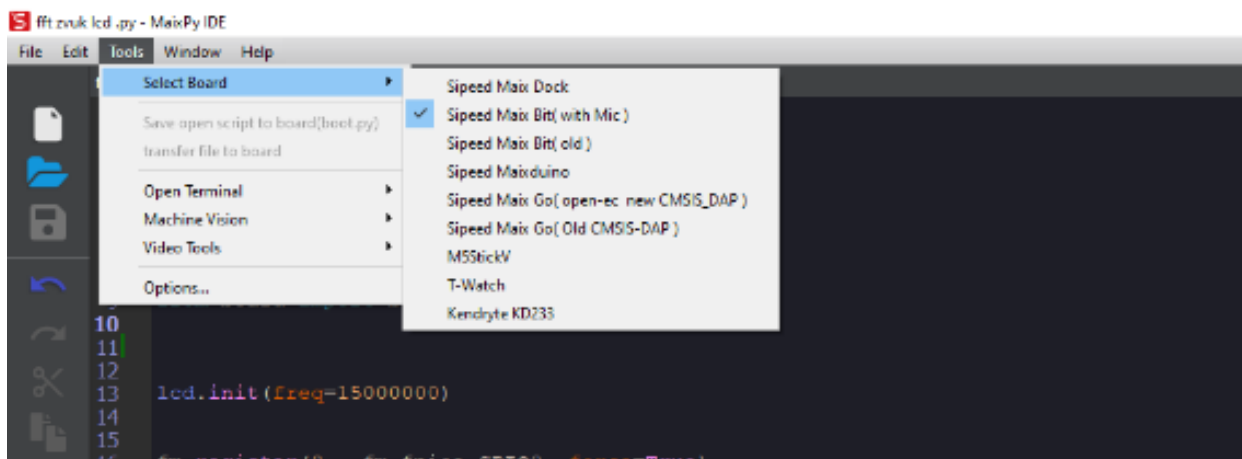
Technologie MEMS (electro-acoustic transducer sensor) spojuje elektroakustický snímač, který převádí proměnný příchozí akustický tlak na kapacitní změny s integrovaným obvodem ASIC, který tyto kapacitní změny převádí na analogový nebo digitální výstup. V našem případě byl zvolen digitální výstup [22].



Obrázek 9 citlivost (odezva) mikrofonu pro jeho frekvenční rozsah [22]

3.2 Maixpy IDE

Po představení hardwaru na platformě Sipeed, který mi byl zapůjčen je potřeba popsat si i základní části softwarových řešení této společnosti. Tato část tedy popisuje základy rozhraní, které je využito pro vytváření ovládacích programů a jejich integraci do mnou používané vývojové desky.



Obrázek 10 Printscreens aplikace MaixPy IDE s viditelným výběrem desek

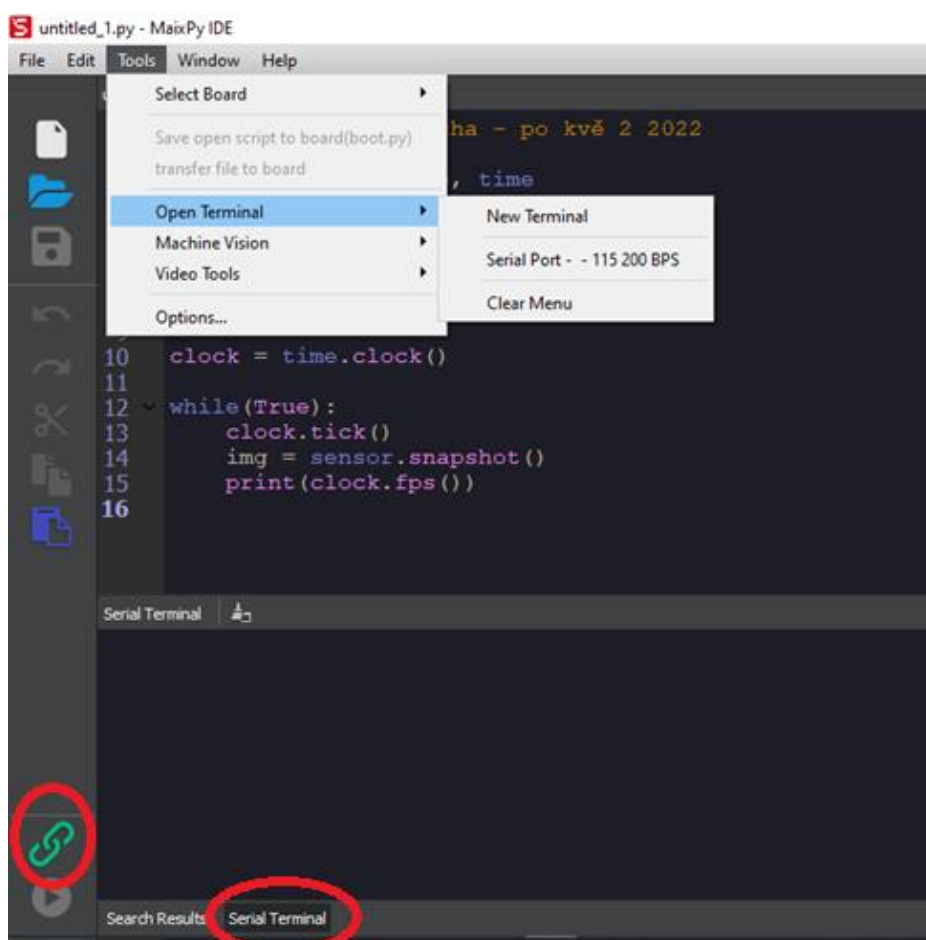
Maixpy se dá označit jako port Micropythonu pro řídicí mikročip Kendryte K210, tedy jako komunikační port mezi počítačem a tímto čipem. Funguje na všech možných operačních systémech, tedy na Windows, iOS i Linux. Jeho funkcí je usnadňovat a zrychlovat programování na K210. Tento port lze provozovat na Programu Maixpy IDE. IDE je rozhraní, které můžeme vidět na obrázku 10 a jeho základní funkce budou dále více popsány v rámci seznámení se s tímto prostředím. Historie tohoto portu započala v roce 2018, kdy byla vytvořena první deska opatřená Kendryte K210 a vznikla potřeba pro zjednodušení jeho ovládání. V první polovině roku 2019, kdy byl tento port integrován do zmíněného programovacího prostředí IDE, došlo k velkému pokroku ve využitelnosti těchto periférií v různých zařízeních. Posledním, zatím nejdůležitějším krokem bylo na konci roku 2020 přidání systematického výběru konkrétního typu vývojové desky což usnadnilo nastavování komunikace počítače a vývojové desky vzhledem ke drobným rozdílům u jednotlivých produktů firmy Sipeed. Možnost výběru konkrétní desky můžeme také vidět na obrázku 10 [15].

Kromě této funkce můžeme na snímku vidět zbytek rozhraní pro ovládání tohoto programu. Vlevo na postranní liště se nacházejí tlačítka pro vytvoření nového dokumentu, otevření dokumentu a poté pro uložení rozpracované práce. Pod nimi jsou tlačítka na vrácení akcí zpět a dopředu. Níže pak kopírovací a vkládací tlačítka pro text. V horní liště pak můžeme využít především složky Edit, kde najdeme různé stejné funkce jako v levé liště i s možnostmi pro hledání a nahrazování textu. A dále složky Tools, kde kromě výběru konkrétní desky můžeme provádět nahrávání konfiguračních programů na desky. Dále otvírat terminál což si ještě popíšeme dále v této části práce. Posledními možnostmi je využití integrovaných funkcí této aplikace pro zpracování videa a kontrolu syntaxe daného programu.

Jako důležité vnímám i obeznámit se se zmiňovaným programovacím jazykem Micropython. Ten je založený na gramatice jazyka Python3. Obsahuje většinu základní gramatiky tohoto jazyka, ovšem pro účely tohoto software nemusí být aplikovány některé syntaxe (soubory pravidel pro definování kombinací symbolů pro určení správné struktury

dokumentu). Toto vede ke snížení výpočetní zátěže. Jeho další funkcí je práce jako tzv. parser. Tedy program schopný kontrolovat syntaktickou správnost dokumentu čímž výrazně usnadňuje práci na programování v tomto rozhraní. Dokáže totiž hledat chyby v kódu a zobrazit pravděpodobný důvod jeho nefunkčnosti [15].

Další funkcí související s ovládáním předloženého vývojového kitu je samotné propojení IDE a Kendryte. To může být dosaženo dvěma různými způsoby, které můžeme vidět na obrázku 11. První možností je připojení komunikace pomocí zeleného tlačítka connect ve tvaru článku řetězu, který můžeme vidět zakroužkované ve spodní levé části programu. Toto tlačítko je propojené s funkcí seriál Terminal vpravo od ní, zde můžeme vidět hodnoty vypisované v sériové komunikaci během spuštění programu připojeného pomocí zmíněného tlačítka. Druhou variantu připojení můžeme vidět nahoře v liště Tools. Jedná se o otevření samostatného komunikačního okna na sériovém portu. Tato varianta je díky otevření v samostatném okně přehlednější a lépe se mi s ní po seznámení se s programem pracovalo.



Obrázek 11 Možnosti komunikace programu s čipem vývojové desky

3.3 Open source knihovny

Poslední velice důležitou součástí ekosystému společnosti Sipeed jsou tzv. open source knihovny. Jak z názvu vyplývá, jedná se o volně dostupná uložení informací, která jsou sdílena mezi tvůrci z celého světa, tak i se samotnými vývojáři z firmy Sipeed. Jsou zde sdíleny návody i postupy na různé aplikace a programová řešení pro všechny druhy vývojových desek. Vývojáři si zde mohou vzájemně předávat zkušenosti a řešení různých problémů, kterými se zabývají ve

smyslu pozitivního přínosu pro daná odvětví. Jedná se konkrétně o dvě knihovny Maixhub a Github v závislosti na aplikačních řešeních.

3.3.1 Maixhub:

Tato knihovna se zaměřuje na aplikace AI technologie a sdílení kreativních modelů této platformy. Poskytuje také funkci pro trénování a rozvoj AI modelů a funkci jejich sdílení.

Obecně stačí pouze připravit sadu dat, kterou chce vývojář trénovat a rozvíjet v rámci strojového učení. Není tedy potřeba vytvářet a programovat tréninková prostředí. Její hlavní přednost tedy je v rychlém trénování strojově se učících modelů. V současné době Maixhub podporuje trénování klasifikačních modelů a modelů detekce cílů.

Postup pro fungování na platformě je následující. Nejprve je potřeba určit si konkrétní cíl simulace. Jestli jde o klasifikaci nebo detekci. Dále je potřebné vytvořit sadu dat, ta musí splňovat podmínky návodu k použití konkrétního tréninkového prostředí. Následně již stačí pouze nahrát data do prostředí a spustit automatické trénování na cloud výpočetní kapacitě společnosti Sipeed. Poté už jen stačí čekat na výsledky tréninku. Po jeho skončení přijde uživateli na e-mail oznámení. V oznámení jsou uvedeny výsledky modelace a soubory s konkrétními daty [23].

3.3.2 Github

Tato knihovna se naopak zaměřuje na aplikace v oblasti AIoT. Github využívají vývojáři, instruktoři programování, studenti i podniky po celém světě. Je také založen na open source myšlence. Slouží primárně ke kontrole verzí a spolupráci vývojářů. Vytvořené a zde sdílené projekty se tedy vyznačují otevřeným zdrojovým kódem, což znamená, že s nimi může kdokoliv pracovat dále. Tento přístup by se dal nazvat jako organizovaná spolupráce v rámci jedné platformy. Platforma také podporuje správu verzí, která nabízí efektivnější způsob vytváření funkčních projektů.

Platforma umožňuje také spolupráci více různých vývojářů z celého světa na společných projektech. Spolupráce na kódu probíhá v takzvaných větvích a revizích, které fungují jako systém správy verzí. Tím umožňuje bezproblémovou spolupráci bez narušení integrity původního projektu a umožňuje uchování všech rozdělaných verzí kódů pro jejich případné znovuvyužití.

Github se dělí na jednotlivá úložiště, větve a revize. Zároveň obsahuje výukový program, který vás naučí základnímu postupu pro vytváření vlastních úložišť a větví. Tento program vytvořil vývojář Peter Cottle s původní myšlenkou vysvětlit větvení úložišť na Github. Účet zde je zadarmo a umožňuje spolupracovat s neomezeným počtem spolupracovníků na libovolném počtu veřejných adresářů s kompletní sadou funkcí. Ovšem pro práci na soukromých adresářích je potřeba placená verze, která odemkne všechny podporované funkce tohoto systému. S bezplatným účtem zároveň uživatel získá i podporu vývojářů komunity Github [24].

3.4 Využitý Software

K vypracování práce bylo využito několik software programů a ty nejdůležitější budou nyní popsány.

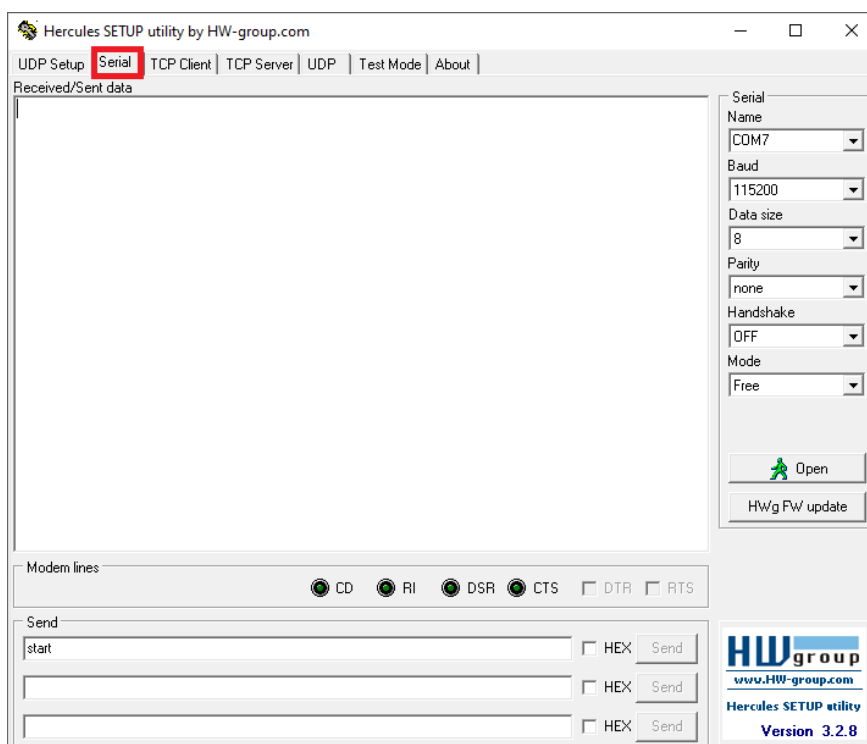
3.4.1 Hercules

Důležitou součástí využívanou při vytváření a debugingu řídicího kódu pro prostředí Maixpy bylo software Hercules, které si nyní popíšeme.

Aplikace Hercules je terminál pro obsluhu sériových portů (RS – 485 a RS – 232). Byla vytvořena pro interní užití společnosti HW group. Postupem času se rozšířila o hodně dalších funkcí spojených s komunikací a byla poskytnuta jako freeware. Důležitá vlastnost je implementovaný sériový terminál spolupracující s virtuálními porty [26].

Z pohledu této práce byla využita jako velice užitečný nástroj při tvorbě řídicích programů. Konkrétně pak při ověřování funkčnosti a nastavování spojení po USB a UART portu díky tomu, že umožňuje zapisovat a číst z jednotlivých komportů.

Vzhled rozhraní můžeme vidět na snímku 12. Na horní liště se nachází volby funkcí programu. Červeným rámečkem je označený sériová komunikace na UART portu. Na pravé liště můžeme navolit konkrétní parametry spojení. Hlavním je zde volba, na kterém portu komunikace probíhá a v jaké rychlosti přenosu. Ve spodní části můžeme využít funkce pro posílání zprávy.



Obrázek 12 Rozhraní aplikace Hercules

3.4.2 Matlab App designer

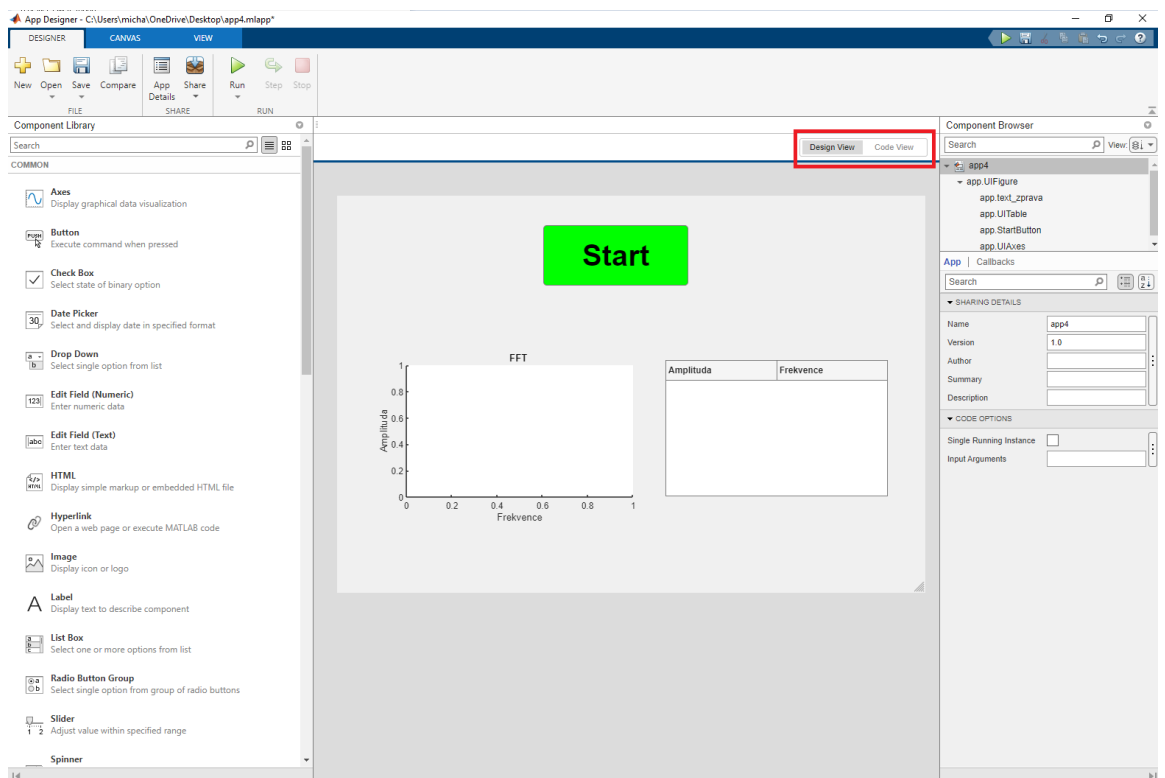
Pro účely spouštění měření a zobrazení výsledných hodnot byla využita aplikace Matlab. Konkrétně pak jeho část zvaná Matlab app designer, kterou si nyní popíšeme.

App designer je samostatný program Matlab, který zahrnuje uživatelské rozhraní pro optimalizaci úloh a výpočtů. Umožňuje vytvářet profesionální aplikace i přesto, že konkrétní uživatel není profesionální vývojář software. Funguje na principu přetahování vizuálních blokových komponent. Jejich umístěním na ploše app designeru vznikne návrh grafického

uživatelského rozhraní vaší vyvíjené aplikace. Po rozvržení bloků je nutné jim přiřadit požadovanou funkci. Jejich rychlého naprogramování lze docílit pomocí integrovaného editoru.

Všechny úlohy spojené s tvořenou aplikací se provádějí přímo zde. Do klasického příkazového řádku aplikace Matla však designer vypisuje stejným způsobem, jakým je v aplikaci Maixpy IDE vypisováno na sériový monitor. Vaše hotová aplikace může být vložena do více různých produktů Matlab, se kterými bude spolupracovat.

Na obrázku 13 je snímek obrazovky s rozhraním této aplikace. Uprostřed vidíme plochu pro vkládání programovatelných bloků. Konkrétně můžeme vidět rozvržení navržené aplikace pro navrhované zařízení. Nachází se zde start tlačítko pro spuštění měření a graf s tabulkou pro zobrazení naměřených hodnot. Vlevo je panel pro výběr bloků a vpravo pak lišta pro nastavení jejich parametrů. V červeném rámečku je označené přepínání mezi designovým rozhraním a rozhraním na tvorbu zdrojového kódu. Část s kódem bude popsána dále v kapitole věnované programu na ovládání tohoto rozhraní.



Obrázek 13 Ukázka rozhraní app designer

4 NÁVRH A REALIZACE ZVUKOVÉHO ANALYZÁTORU

Hlavním cílem této diplomové práce je návrh a realizace zařízení, které bude schopno na základě povelu odeslaného z počítače přes program Matlab autonomně nahrát aktuální hluk v místnosti. Tento zvuk poté interně analyzovat pomocí algoritmu FFT. Ten byl na základě seznámení se s vývojovým prostředím Sipeed sledán jako nejvhodnější varianta pro jednoduchou a rychlou analýzu naměřených signálů. Je to dáno navržením mikročipu Kendryte pro analytické a diagnostické využití v praxi, především tedy integrací podpůrných systémů na zpracování FFT jako je akcelerátor jeho výpočtů. V posledním kroku by toto zařízením mělo zpracovaná data odeslat nazpět do počítače pro zobrazení v grafu případně zápisu do tabulek pro další zpracovávání.

Součástí kapitoly je i širší seznámení s prostředím Matlab. Konkrétně pak jeho částí app designer. Ten umožňuje vytvoření grafického blokového rozhraní. Jednotlivým blokům pak můžeme naprogramovat libovolné funkce a tím si vytvořit požadované okno s potřebnými bloky a specifickými parametry.

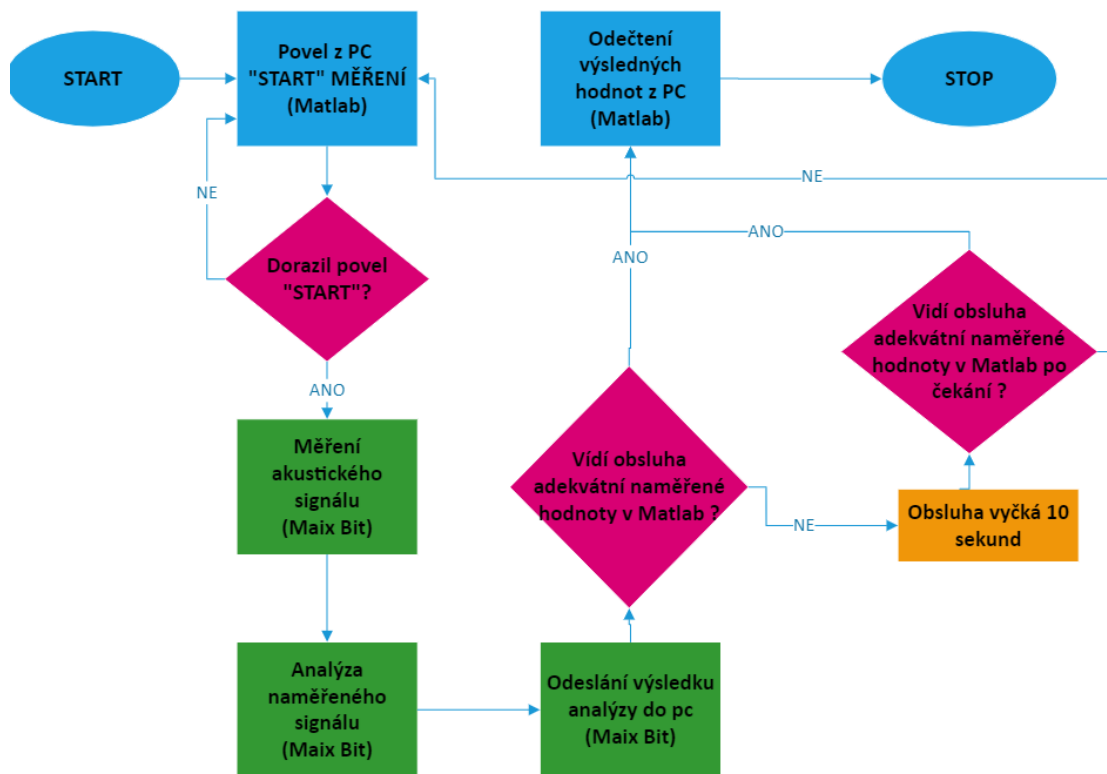
Na základě zadání a systematické přípravy byl vytvořen vývojový diagram pro návrh diagnostického zařízení viz. obrázek 14. Z tohoto vývojového diagramu a získaných informací v rámci seznámení se zapůjčenou vývojovou platformou byla provedena praktická realizace zařízení pro nahrávání zvuku a jeho následnou analýzu.

V rámci této kapitoly bude představeno hardwarové i softwarové řešení navrhovaného zařízení. Z tohoto důvodu bude také představeno softwarové rozhraní Herkules, které bylo k realizaci využíváno. Ze stejného důvodu bude také popsána externí hardwarová komunikační periferie od společnosti Arduino. Dále zde budou vysvětleny zdrojové kódy pro řízení zařízení i pro zobrazení naměřených hodnot.

4.1 Návrh funkcí

Návrh fungování daného zařízení vycházející z vytvořeného vývojového diagramu na snímku 14, který si nyní více přiblížíme a popíšeme si jednotlivé kroky při jeho průběhu. Počátek nebo, chceme-li start systému je na modrém vývojovém bloku. Modrá barva v tomto diagramu znamená, že se jedná o prostředí Matlab na řídicím počítači. Zde musí přijít iniciace programu obsluhou zařízení v podobě zmáčknutí tlačítka START. Tímto se provede odeslání příkazu do hardware Maix bit.

Práce probíhající na desce Maix bit v rámci mikročipu Kendryte k210 je v diagramu označena zelenou barvou. Mezi odesláním příkazu k začátku měření a provedením měření se nachází rozhodovací člen. Ten reprezentuje programový kód nahraný v Kendryte. Má na starosti, aby se program nespouštěl samovolně, ale pouze pokud si to obsluha přeje. Jakmile se měření spustí, všechny ostatní kroky na Maix bit by měly probíhat autonomně na základě nahraného zdrojového kódu. Tedy v prvním kroku zařízení pomocí integrovaného mikrofonu nahraje zvuk ve svém okolí. Poté na tento záznam aplikuje algoritmus FFT a v posledním kroku získaná data odešle nazpět do počítače na rozhraní Matlab.



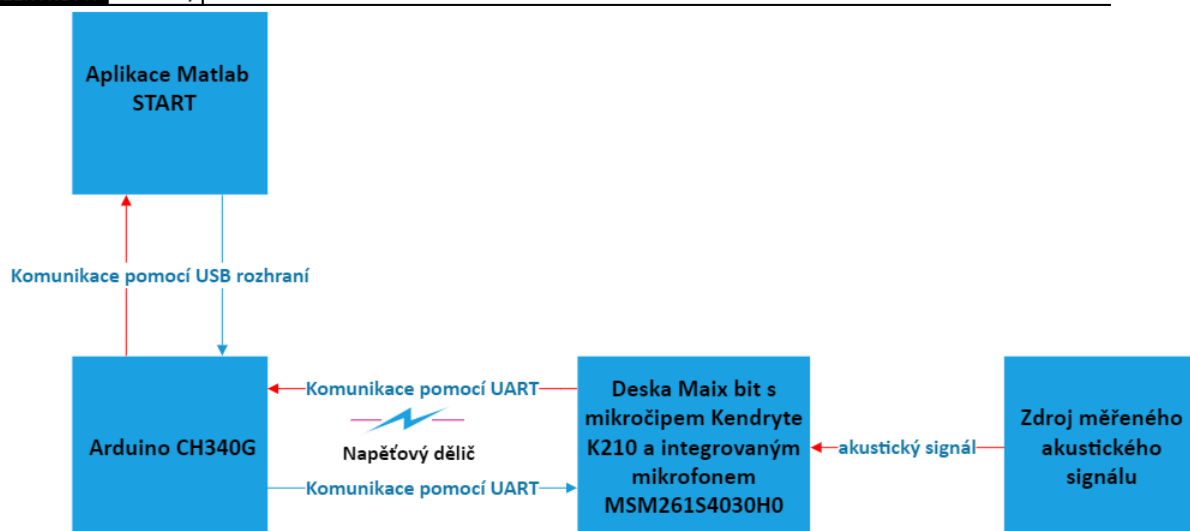
Obrázek 14 Vývojový diagram návrhu autonomního zařízení

Mezi bodem odeslání výsledku analýzy z prostředí Maix do prostředí Matlab se opět nachází rozhodovací člen. Je to z důvodu, kdyby nastal nějaký problém, kvůli kterému by nedorazila naměřená a vyhodnocená data do počítače. Případně pokud by data zobrazená v počítači vykazovala známky nesmyslných hodnot. Tím jsou myšleny hodnoty mimo předpokládaný rámec, případně data nečitelná. V tomto případě obsluha vyčká 10 sekund, a pokud se do této doby situace nezmění, je celý program v podstatě restartován opětovným spuštěním, tedy zmáčknutím tlačítka start. Tímto se celý program restartuje a spustí znovu.

4.2 Návrh konstrukce

Realizace konstrukce zařízení vychází z popsaného vývojového diagramu. Snaha při sestrojování bylo dosáhnout co možná největší jednoduchosti systému, a to především pro zajištění dobré funkčnosti a vyhnutí se systémovým chybám.

Navržené uspořádání použitých komponent spolu se způsobem jejich zapojení můžeme vidět na obrázku 15, který si dále popíšeme. Pro řízení a zobrazování dat je využíván program Matlab spuštěný na počítači. Ten je pomocí USB rozhraní spojen s deskou Arduino, která je podrobněji popsána v následující kapitole zabývající se komunikací. V té bude popsán také napěťový dělič, který je využíván pro umožnění komunikace na rozhraní UART mezi deskou Arduino a deskou Maix bit. Komunikační kanály jsou na obrázku zakresleny modrou šipkou pro data proudící z počítače do systému a červenou šipkou jsou naznačeny cesty dat proudící ze systému do počítače. Poslední částí je zdroj měřeného akustického signálu. Tím může být zpočátku signál generovaný uměle o přesně dané frekvenci pro ověření správného měření, případně pak při praktickém měření jakýkoliv zdroj hluku, který chceme analyzovat.



Obrázek 15 Navrhovaná konstrukce zařízení

4.3 Komunikace

Vzhledem k obsazení komunikačního portu usb kabelu z desky Maix Bit na komunikaci s programovacím prostředím a zobrazování sériového monitoru programu Maixpy IDE, nezbyvá na tomto portu prostor pro předávání i naměřených hodnot do počítače. K tomuto účelu je nezbytné využít externí zařízení. Zvolena byla desky od společnosti Arduino CH340G viz obrázek 18. Toto zařízení bude popsáno dále v samostatné kapitole. Zároveň je nezbytné mezi tyto desky implementovat tzv. napěťový dělič. Je to dáno tím, že komunikační porty UART na obou deskách fungují na jiném napětí. Podrobnější vysvětlení se nachází v dalším odstavci.

4.3.1 UART

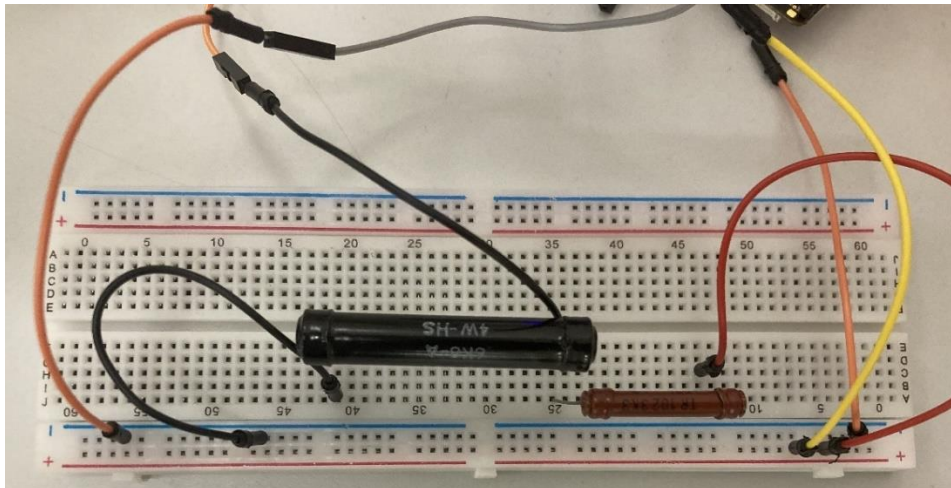
Jak již bylo řečeno v kapitole 3.1 Maix bit disponuje integrovanou sběrnici UART. Tedy univerzální asynchronní přijímač/vysílač. Přes tuto sběrnici komunikuje přes USB-C. Vzhledem k obsazenosti USB-C spojením s prostředím Maixpy je nutné komunikaci s Matlab zařídit přes piny desky. Tyto piny jsou běžně označovány jako RX a TX to můžeme vidět na obrázku 7 na pozicích pinů 4 a 5. Ovšem v případě této vývojové desky jsou tyto piny softwarovým nastavením blokovány pro komunikaci s LCD displejem. Vzhledem k možnosti libovolně přeprogramovat piny na desce však můžeme tento problém obejít. Komunikace UART tedy byla v mé práci přeprogramována na piny, které nejsou blokovány. Konkrétně pin 6 na TX neboli vysílač a pin 7 na RX neboli přijímač.

Pin 6 tedy funguje jako vysílač naměřených dat na sériový port. Nutno podotknout, že pro správnou funkci je potřeba tento vysílač připojit na pin desky Arduino označený jako RX, neboli přijímač. A zase naopak pro zpětnou komunikaci z Matlab do Maix bit je nezbytné připojit vysílací pin Arduino na pin 7, neboli přijímač desky Maix bit.

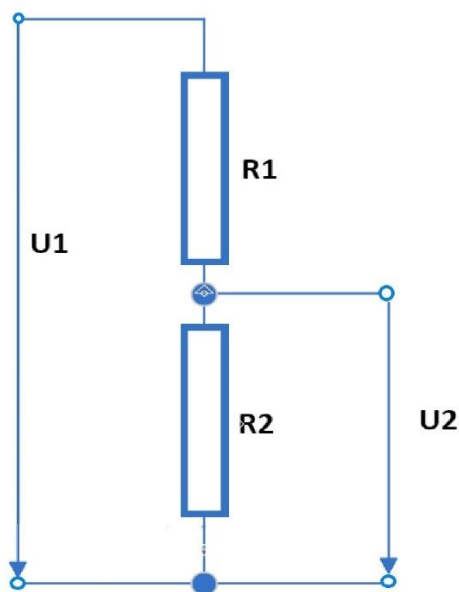
4.3.2 Napěťový dělič

Sběrnice UART na Maix bit funguje na napětí 3,3 V. Na Arduino je operační napětí 5 V.

Pro správné fungování tedy musel být do zapojení zařízení mezi Maix bit a Arduino CH340G vložen tzv. napěťový dělič, který můžeme vidět na obrázku 16. Toto zapojení vychází z výkresu na obrázku 17. Sestává se z dvou různých odporů o hodnotách $R_1=3300$ ohmů a $R_2=6800$ ohmů. Dále zde je uveden i způsob výpočtu v rovnici 5.1, který vedl k výběru odporů o vybraných hodnotách.



Obrázek 16 Napěťový dělič



Obrázek 17 Výkres zapojení děliče napětí

Rovnice pro výpočet výsledného napětí:

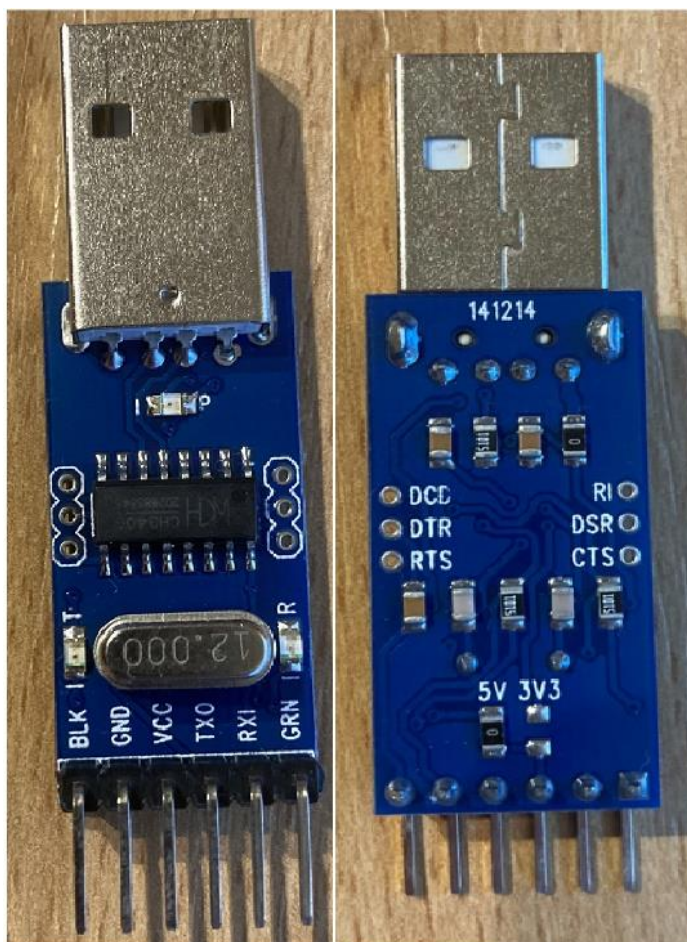
$$U_2 = U_1 * \frac{R_2}{R_1+R_2} = 5 * \frac{6800}{10100} = 3,36 \text{ V} \quad (5.1)$$

4.3.3 Arduino CH340G USB

Tato deska, jak již bylo zmíněno je v této práci využita ke komunikaci desky Maix Bit a počítače pomocí sériového rozhraní UART. Modul je běžně využíván pro různá zařízení komunikujících sériově. Především pro jejich debugging (odladování chyb), aktualizaci jejich firmwarů případně k přímé komunikaci pro desky bez USB rozhraní jako jsou desky Arduino pro mini a jiné.

Jeho základní charakteristiky jsou:

- operační napětí 5 V
- napětí logického obvodu 5 V
- rychlost zápisu do 115200 bps
- podpora USB 1.1, 2.0, 3.0 [25].



Obrázek 18 Arduino CH340G USB to TTL Adapter Module

4.4 Zdrojové kódy

V této kapitole budou popsány řídicí kódy pro řízení z počítače a autonomní práci zvukového analyzátoru.

4.4.1 Maixpy IDE

Tato část se zaměřuje na popis zdrojového kódu, který je uložen na mikroprocesoru Kendryte. Díky tomuto programu vykonává deska Maix bit požadovanou funkci. Popis bude rozdělen na tři části podle postupu, jakým způsobem deska program vykonává.

V první části, jejíž kód můžeme vidět na obrázku 19, se zaměříme na primární nastavení desky pro danou práci. Nejprve je potřeba zajistit, aby deska věděla, s jakými částmi jejího hardware a software budeme pracovat. Daný krok můžeme nazývat například načtení knihoven. Ty jsou rozděleny na několik druhů podle toho, jak je má deska indexované. Na snímku 19 můžeme tuto část ohraničit řádky 1 až 6. Nezáleží na pořadí, v jakém jsou v kódu zapsány. Důležité je nezapomenout žádnou, kterou budeme dále v kódu potřebovat. Bez jejich načtení by řídicí procesor nevěděl jak s kódem, který je v něm nahraný pracovat a hlásil by chyby. Můžeme vidět, že jsou načteny knihovny týkající se komunikace jako I²S a UART. Dále důležité nastavení jako gpio, fm nebo board_info, které umožňují přiřazovat různé funkce Maix bit na příslušné výstupní piny této desky. Pro funkci desky nepostradatelné ovládání audio a knihovna FFT algoritmu.

Následujících řádky 7 až 11 první části kódu jsou určeny pro uživatele na nastavení parametrů využívaných funkcí. Pro správné fungování zařízení je potřeba nastavit tyto parametry. Sample_rate je nazvána proměnná, kterou nastavujeme vzorkování. V programu Matlab je možnost nastavit hodnoty na rozsah 100 až 5000. Je to z důvodu, že čím větší vzorkování nastavíme, tím větší frekvenční spektrum program měří, a protože jeho přesnost je omezena do 5 kHz nemá smysl měřit větší hodnoty. Přesto můžeme ručně v Kendryte nastavit libovolnou hodnotu. Sampl_points je proměnná, kterou si uživatel nastaví množství jednotlivých odebraných vzorků z nahraného audia. Poslední důležité je nastavení fft_points. Díky tomuto nastavení si uživatel vybírá počet vzorků pro výpočet FFT. Kendryte umožňuje volit počet těchto bodů pouze ve formátu 2⁶ až 2⁹. Tedy v hodnotách, 64, 128, 256 a 512 bodů. Všechna tato nastavení tedy umožňují uživateli nastavit si potřebné hodnoty dle potřebné kvality výsledných hodnot.

```
1 # Inicializace potřebných knihoven
2 import os, uos, time, utime, image, lcd
3 from Maix import GPIO, I2S, FFT, Audio
4 from fpioa_manager import fm, *
5 from board import board_info
6 from machine import UART
7 # Uživatelské nastavení
8 sample_rate = 16000
9 sample_points = 2048
10 wav_ch = 2
11 fft_points = 512
```

Obrázek 19

Náhled kódu MaixPy 1/4

V druhé části zdrojového kódu, který je vidět na obrázku 20, již vidíme funkce, které slouží pro inicializace a přiřazení funkcí jednotlivým pinům a částem vývojového kitu. Pro řádky 12 až 15 se jedná o přiřazení pinů na kterých bude probíhat komunikace mezi mikročipem desky a jejím integrovaným mikrofonem.

Následuje samotná inicializace komunikace pro I²S, tedy funkce, která postupně říká programu, že informace obdržené z akustického signálu v digitální podobě bude uložena do paměti pod názvem proměnné RX. Dále se zadá konfigurace tohoto spojení společně s přiřazením módu komunikace. Nakonec v této části přiřadíme `sample_rate`, kterým si uživatel může nastavit počet bodů této komunikace uložených pod touto proměnnou.

Následující řádky 20 až 23 slouží k obstarání komunikace přes komporty mezi deskou Maix bit a počítačem přes vložené USB zařízení Arduino ch340g, které je popsáno v kapitole 4.3.3. První dva řádky obsahují nastavení pro přiřazení, na kterých pinech desky bude tato komunikace probíhat i s označení RX jako přijímače dat na port a TX jako pin po kterém informace ze zařízení odcházejí. Na posledním řádku této části kódu je nastavena proměnná UART. Pod tuto proměnnou se bude do paměti zařízení ukládat právě komunikace, které v tomto kroku zároveň přiřadíme všechny parametry. Velice důležitá je zde tzv `baudrate`, ta označuje rychlost komunikace a v našem případě je nastavena na maximální hodnotu kterou všechny komponenty zvládnou tedy 115200. Dále jsou nastaveny například hodnoty v kolika bitech a v jakém módu komunikace probíhá. Poslední nastavitelnou hodnotou je počet znaků, které se načítají do tzv `bufferu`, Do toho jsou data nahrána z jednoho zařízení a následně tato data na druhém zařízení načítáme.

```

12 # Přiřazení pinů komunikace mikrofon s Kendryte
13 fm.register(20, fm.fpioa.I2S0_IN_D0, force=True)
14 fm.register(19, fm.fpioa.I2S0_WS, force=True)
15 fm.register(18, fm.fpioa.I2S0_SCLK, force=True) )
16 # Inicializace komunikace mikrofonu s Kendryte
17 rx = I2S(I2S.DEVICE_0)
18 rx.channel_config(rx.CHANNEL_0, rx.RECEIVER, align_mode=I2S.STANDARD_MODE)
19 rx.set_sample_rate(sample_rate)
20 # Přiřazení pinů a inicializace komunikace UART
21 fm.register(7, fm.fpioa.UART1_RX)
22 fm.register(6, fm.fpioa.UART1_TX)
23 uart = UART(UART.UART1, 115200, 8, None, 1, timeout=1000, read_buf_len=4096)

```

Obrázek 20 Náhled kódu MaixPy 2/4

Po provedení nastavení pro nahrávání zvuku a komunikace mezi všemi perifériemi následuje poslední část kódu, která je vidět na obrázku 21. Nejprve je nutné zavést si dvě proměnné potřebné dále v kódu. Prvním je hodnota `odesílání`. Ta slouží k nastavování počtu dat odeslaných do PC. Druhou proměnnou je pak hodnota `run` sloužící jako funkční argument funkce `řídící` zbytek programu.

Ten se nazývá tzv. `smyčka` programu. Jedná se o část kódu, která zajišťuje, aby se vykonávali požadované funkce na daném zařízení. Smyčkou je označována z důvodu cyklického opakování zde konkrétně pomocí funkce `while`. Tato cyklická funkce je řízena parametrem `run`. Tedy dokud je jeho hodnota na hodnotě `true`, bude se celý následující kód neustále dokola opakovat.

Pro správné fungování zdrojového kódu na této platformě je nutné dodržet pravidlo sloupcového zarovnání tak jak to můžeme vidět právě na obrázku 21. Odskočení následných

funkcí po hlavním while značí, že tato část kódu se provede v dalších krocích. Tedy kód zarovnaný více doprava se bude provádět tzv. hlouběji v programu. Jedná se o funkce vnořené.

Druhou úrovní kódu je funkce if na řádce 31. Její nastavení znamená, že jestliže se na sériovém komportu budou nacházet nějaká data, vykoná se následující třetí úroveň vnoření kódu. Dle vysvětleného zarovnávání programu lze lehce vidět, že pokud se na komunikačním portu žádná data nenacházejí je vykonáván příkaz else na řádce 56. Ten má za úkol program na 10 milisekund pozastavit. Po uběhnutí tohoto času bude opět zjištěno, zda se již neobjevila nějaká data na komunikaci UART.

Na řádce 32 se nachází opět funkce while. Nachází se na třetí úrovni kódu a bude tedy provedena za předpokladu přijetí dat na sériový port. Úkolem této funkce je načíst přijatý příkaz z UART komunikace tedy počítače a uložit je do paměti pod názvem proměnné read_data. následně tato data vypsat na sériový monitor. Z důvodu přehledu, jaký příkaz byl přijat.

Následuje poslední, čtvrtá úroveň zdrojového kódu se třemi rozhodovacími funkcemi if. O tom, která z funkcí bude provedena rozhoduje přijatá informace z řídicího rozhraní Matlab.

První možností pro průběh této smyčky je na řádcích 35 až 45. Tato část je stěžejní pro fungování zařízení. Podmínka provedení této funkce je přijatý příkaz start. Jedná se o funkci, která provede zaznamenání zvukového záznamu. Tento záznam zvuku uložený pod proměnnou audio je definován pomocí počtu bodů neboli sample_points, které si uživatel sám dříve nastaví. Dále se pomocí integrované funkce FFT provede matematický výpočet tohoto algoritmu z uložených bodů audio a v rozsahu nastaveném uživatelem pomocí proměnné fft_points. Dostaneme výsledek fft_res v podobě reálných a imaginárních čísel. Z těchto hodnot jsme pak schopní pomocí funkcí FFT.freq a FFT.amplitude nechat dopočítat pro každý bod měření konkrétní frekvenci a amplitudu. Tyto hodnoty se ukládají do proměnných označených fft_freq a fft_amp. Důležité je zde zmínit, že pro výpočet frekvencí je využita polovina z nastavení fft_point. Je to z důvodu odfiltrování záporné složky výsledků a bude více vysvětleno v kapitole kódu pro Matlab. Dále se nacházejí příkazy pro zobrazení vypočtených dat na sériovém monitoru Kendryte z důvodu přehlednosti. Následně se v rámci tohoto if provede vnořená funkce for na páté úrovni kódu. Tento povel na řádcích 43 a 44 postupně zapisuje všechny hodnoty frekvencí a amplitud do matice o velikosti odpovídající počtu bodů spočteného FFT. Počet těchto měření je přednastaven v proměnné odeslání na 512, nicméně může být uživatelem nastaven na jinou hodnotu, a to příkazem z počítače o kterém si povíme dále. Formát zápisu do matice je první vypočtená frekvence, středník, první vypočtená amplituda a mezera. Tento zápis se opakuje postupně až do vypsaní nastaveného počtu zápisů.

Pokud však přijatá informace nebude povel Start, ale naopak z pc bude odeslán příkaz pro ukončení měření v podobě zprávy stop, bude hodnota run nastavena na hodnotu false. Tím se celý program vypne na základě neplatného rozhodovacího prvku pro hlavní funkci while na řádce 30.


```

27 odesilani=512
28 run=True
29 # Smyčka programu
30 while run==True:
31     if uart.any():
32         while uart.any():
33             read_data=uart.read()
34             print(read_data)
35             if read_data==b'start':
36                 audio = rx.record(sample_points)
37                 fft_res = FFT.run(audio.to_bytes(),fft_points)
38                 fft_freq = FFT.freq(int(fft_points/2), sample_rate)
39                 fft_amp = FFT.amplitude(fft_res)
40                 print(fft_freq)
41                 print(fft_amp)
42                 print(fft_res)
43                 for i in range(0,odesilani):
44                     uart.write(str(fft_freq[i])+';'+str(fft_amp[i])+chr(32))
45                     uart.write(chr(10))
46             if read_data==b'stop':
47                 run=False
    
```

Obrázek 21 Náhled kódu MaixPy 3/4

Poslední ze tří funkcí if je varianta vyobrazena na obrázku 22. Je spuštěná přijetím dat z PC v podobě příkazu nastav. Tato funkce byla přidána dodatečně po zprovoznění předchozího kódu jako rozšíření právě pro možnost regulace počtu naměřených dat a měřeného frekvenčního rozsahu. Na obrázku 26 můžeme vidět jakým vývojovým diagramem se tato funkce řídí.

Její průběh je následující. Po přečtení příkazu nastav se program na 2sekundy pozastaví. Během tohoto času obdrží z matlab hodnotu, na kterou má nastavit proměnnou odesílání. Zbytek této funkce slouží k úpravě formátu příchozích dat. Matlab totiž odesílá hodnoty ve formátu bitového kódu ve tvaru string. V tomto tvaru je příchozí hodnota brána jako jeden znak a není možné rozlišit jednotlivá příchozí čísla, řádky 89 až 92 tedy tento formát převedou na jednotlivé znaky, tak aby mohli být rozeznány jednotlivé číslice a přijatou hodnotu nastaví za proměnnou odeslání tak, aby s touto hodnotou mohl program dále pracovat. To samé se pak stane ještě jednou tentokrát pro proměnnou sample_rate. Ten se pak ještě propíše do proměnné rx, která zajišťuje správnou funkci FFT algoritmu.

```

87     if read_data==b'nastav':
88         time.sleep_ms(1000)
89         read_data = uart.read(),
90         read_data = read_data[0]
91         read_data = read_data.decode('utf-8')
92         cislo = int(read_data)
93         odesilani = cislo
94         print(str(odesilani))
95         time.sleep_ms(2000)
96         read_data = uart.read(),
97         read_data = read_data[0]
98         read_data = read_data.decode('utf-8')
99         cislo = int(read_data)
100        sample_rate = cislo
101        rx.set_sample_rate(sample_rate)
102        print(str(sample_rate))
    
```

Obrázek 22 Náhled kódu MaixPy 4/4

4.4.2 App designer

V této kapitole bude popsán zdrojový kód pro aplikaci uživatelského rozhraní v programu Matlab app designer. Tato aplikace je určena k ovládání vývojového kitu a zobrazování výsledných dat naměřených a zpracovaných pomocí navrženého zařízení. Zdrojový kód app designeru je rozdělen na dvě části. Na obrázku obrázku 23 můžeme vidět zašedlou plochu, na ní se nachází kód vytvořený aplikací, kterou uživatel nemůže měnit. Druhá část jsou plochy bílé. Jedná se o místa vytvořená vložením komponent, které jsou popsány v kapitole 3.4.2. Zdrojové kódy napsané v ploše určené konkrétním komponentům reagují na jejich úpravu v rámci uživatelského grafického rozhraní.

První bílou plochou v rámci vytvořeného programu v této aplikaci vidno na snímku 23 na řádcích 17 až 21. Jedná se o část pro nastavení tzv. vlastností, které slouží pro zavedení proměnné typu (app.) k výhodnému využití v rámci kódu napříč jednotlivými komponenty. Konkrétně je zavedeno s pro hodnoty načtené z UART portu a dvě pole hodnot pro rozčlenění amplitud a frekvencí.

Následuje funkce startupFcn začínající na řádce 29, která je spuštěna při načtení aplikace. Jejím účelem je připojit se na port UART komunikace. Rovnou je zde využita i zmiňovaná proměnná pro snadný přístup k datům v rámci celé aplikace. Druhým příkazem provedeným se startem aplikace je funkce pro vyčištění všech pamětí Matlab, aby měření nebylo ovlivněno náhodnými hodnotami.

```
15
16
17 properties (Access = private)
18     s;
19     poleampl;
20     polefreq;
21 end
22
23
24 % Callbacks that handle component events
25 methods (Access = private)
26
27     % Code that executes after component creation
28     function startupFcn(app)
29         app.s = serialport("COM11", 115200, "Timeout",100);
30         clear all
31     end
```

Obrázek 23

Program matlab app designer 1/3

Hlavní částí tohoto kódu je část provedení pomocí stisknutí tlačítka pro spuštění měření. Celý postup je vidět na obrázku 24 a vychází z vývojového grafu návrhu zařízení. V prvním kroku je tedy odeslán příkaz start na Maixpy. Poté program dvě vteřiny vyčká a načte do své paměti příchozí data, která ihned převede z typu (str) na jednotlivé znaky (char). Následně vytvoří prázdnou matici proměnných k zápisu amplitud a frekvencí. Na řádku 41 načte do proměnné n hodnotu nastaveného počtu měření v grafickém rozhraní aplikace, která je defaultně nastavena na hodnotu 256. Jedná se o polovinu hodnot zmíněných v části kódu Maixpy. Toto je dáno vnitřním způsobem zpracování FFT na Kendryte k210. Ten při výpočtech tohoto algoritmu tvoří zrcadlově výsledky v kladném i záporném spektru. Nicméně pro potřeby odečítání dat z grafu nám stačí kladné hodnoty. Ošetření tohoto problému jsme již řešili v kapitole 4.4.1.

Následuje cyklická funkce for, která se provede právě tolikrát, kolikrát si nastavím v grafickém rozhraní. Tato funkce slouží ke zpracování dat načtených z bufferu komportu ve formátu frekvence, středník, amplituda, mezera. Postupně tedy pro všechny průchody měření jednotlivé prvky separuje a zapíše je do předem připravených matic. Po ukončení smyčky a zápisu všech hodnot provede seskupení do jednotné matice a transpozici této společné matice. Tento krok je nezbytný pro správný zápis dat do tabulky v grafické části aplikace. V posledním kroku se pak z těchto hodnot vytvoří také graf FFT závislosti amplitudy na frekvenci.

```

34 function SpustitmenButtonPushed(app, event)
35     write(app.s,'start','char'); % pošli příkaz na Sipeed, Start měřen:
36     pause (2);
37     a = readline(app.s); % přečti odpověď ze Sipeed
38     a=char(a);
39     app.poleampl= [];
40     app.polefreq= [];
41     n=app.PoetmenEditField.Value;
42     for k=1:n
43         mezera=strfind(a,' ');
44         b=a(1:(mezera(1)-1));
45         strednik=strfind(b,';');
46         frek=str2num (b(1:strednik(1)-1));
47         ampl=str2num (b(strednik(1)+1:size(b,2)));
48         app.poleampl=[app.poleampl ampl];
49         app.polefreq=[app.polefreq frek];
50         a=a(mezera(1)+1:size(a,2));
51     end
52     matice = [app.polefreq;app.poleampl];
53     matice = matice';
54     app.UITable.Data = matice;
55     plot(app.UIAxes,app.polefreq,app.poleampl)
56
  
```

Obrázek 24

Program matlab app designer 2/3

V průběhu tvorby programů této práce byli postupně vymyšleny a integrovány nové funkce nad rámec původního návrhu. Byli vytvářeny na základě rozšiřování vývojového diagramu pro toto zařízení, který můžeme vidět na obrázku obrázku 26.

Na obrázku 25 můžeme vidět právě přidané funkce. Nejprve funkce tlačítka Ukončení měření. Slouží k odeslání příkazu stop na Kendryte, čímž přeruší chod jeho hlavní smyčka. Zároveň příkazem close all ukončí všechny procesy probíhající v matlab app designeru.

Druhou bílou plochou, kterou můžeme vidět je oblast pro výčet hodnoty zadané do textového pole v grafickém rozhraní, její využití již bylo zmíněno dříve a nyní bude využito i dále pro odesílání dat.

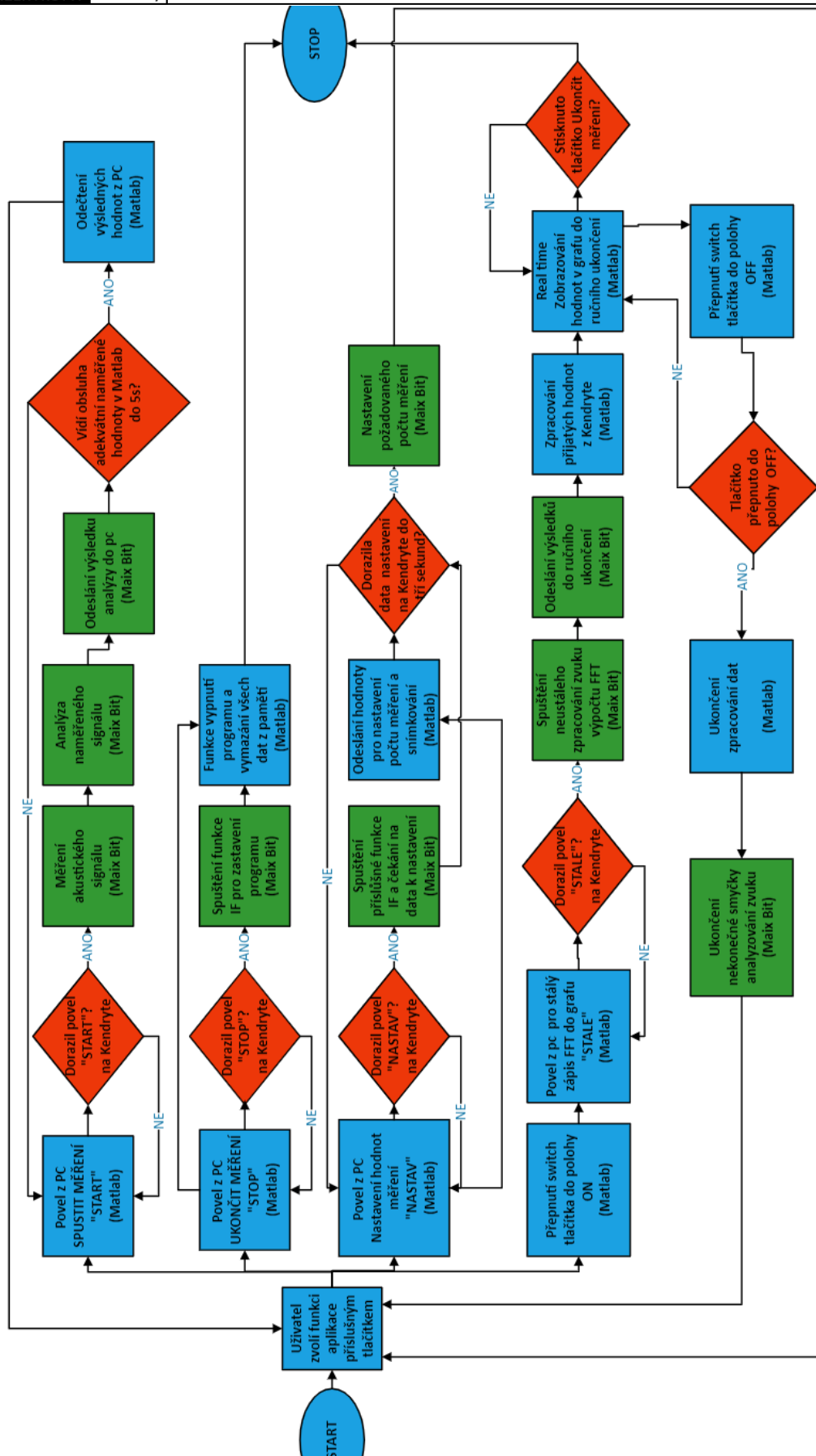
Poslední programovanou částí je tlačítko pro nastavení počtu měření. Zde je tedy stisknutím tlačítka zaslán příkaz nastav na Kendryte. Následuje usnutí programu na dvě sekundy, po kterých odešle na Kendryte uživatelem vyplněnou hodnotu v rozmezí 1-256. Hodnota je odeslána v datovém formátu přizpůsobenému vzájemné komunikaci vycházející z popsání v části komunikace ve zdrojovém kódu Maixpy. Dále opět dvě sekundy vyčká a odešle druhou hodnotu, tentokrát uživatelem vyplněné vzorkování v rozsahu 100-5000. Tímto je nastavení provedeno a nyní může uživatel využít libovolně jiné funkce pro měření.

```
72 function UkonitmenButtonPushed(app, event)
73     write(app.s, 'stop', 'char')
74     close all
75 end
76
77 % Value changed function: Poetmen0256EditField
78 function Poetmen0256EditFieldValueChanged(app, event)
79     app.Poetmen0256EditField.Value;
80 end
81
82 % Button pushed function: NastavithodnotyButton
83 function NastavithodnotyButtonPushed(app, event)
84     flush(app.s)
85     write(app.s, 'nastav', 'char'); % pošli příkaz na Sipeed, S
86     pause (1);
87     m = num2str (app.Poetmen0256EditField.Value)
88     write(app.s,m, 'char')
89     pause (2);
90     m = num2str(app.Nastavenvzorkovn1005000EditField.Value)
91     write(app.s,m, 'char')
```

Obrázek 25 Program matlab app designer 3/3

4.5 Implementace rozšiřujících funkcí

V průběhu realizace zařízení pro zpracování akustického signálu bylo postupně navrženo několik možných funkcí, které by bylo vhodné využít. Za účelem jejich implementace probíhalo rozšiřování původního vývojového diagramu (Obrázek 14 Vývojový diagram návrhu autonomního zařízení). Vždy v návaznosti na funkci byla vytvořena nová větev diagramu a na jeho základě upraveny zdrojové kódy. Dále v této kapitole budou vysvětleny vylepšení programů, které zatím nebyli osvětleny v předchozích textech. Výsledný diagram všech naprogramovaných funkcí vidíme na snímku 26.



Obrázek 26 Návrh vylepšení funkcí zařízení

4.5.1 Rozšíření Maixpy IDE

Do programu ovládajícího vývojovou desku Maix bit byly přidány poslední dvě funkce. První z nich je pouze malá změna určená ke zvýšení uživatelské jednoduchosti. Jedná se o připojení a využití zabudovaných led světel. Ty jsou načteny viz obrázek 27, dále jsou nastaveny tak, aby pro každou funkci svítily jinou barvou. Uživatel tak může lehce rozeznávat pouhým pohledem jakou operaci deska aktuálně provádí.

```
# Přiražení pinů Led světlo deska
fm.register(board_info.LED_R, fm.fpioa.GPIO0, force=True)
fm.register(board_info.LED_B, fm.fpioa.GPIO1, force=True)
fm.register(board_info.LED_G, fm.fpioa.GPIO2, force=True)
led_r = GPIO(GPIO.GPIO0, GPIO.OUT)
led_b = GPIO(GPIO.GPIO1, GPIO.OUT)
led_g = GPIO(GPIO.GPIO2, GPIO.OUT)
```

Obrázek 27 Nnačtení Led

Poslední přidanou funkcí je přepínač pro neustálé živé načítání hodnot do grafu. Data se zobrazují do doby ručního přepnutí a celý postup si popíšeme podle snímku 28. Spuštění probíhá pomocí funkce if. Jejím parametrem pro spuštění je příchod příkazu „stala“. V první části můžeme pozorovat konkrétní využití integrovaného led, které bude při spuštění této funkce svítit tmavě modrou barvou.

Následuje nekonečná smyčka typu while. Zde se odehrává zpracování a odeslání hodnot FFT totožné s již popisovanou funkcí v kapitole 4.4, až do bodu po funkci for. Jedinou změnou je fakt, že smyčka se opakuje neustále dokola a nezastaví se automaticky po odečtení požadovaného počtu měření. Po každém průběhu nekonečná smyčka na svém konci ověří pomocí funkce if, zda se na bufferu UART portu neobjevila nějaká data. Pokud ano, tyto data načte. Jedná-li se o příkaz „stop“ přeruší hlavní smyčku while. Tím ukončí živou analýzu dat a dále čeká na nový příkaz.

```
if read_data==b'stala':
    runn = True
    led_r.value(1) # svítí tmavě modrá - načítá a zobrazuje hodnoty
    led_b.value(0)
    led_g.value(1)
    while runn==True:
        audio = rx.record(sample_points)
        fft_res = FFT.run(audio.to_bytes(),fft_points)
        fft_freq = FFT.freq(int(fft_points/2), sample_rate)
        fft_amp = FFT.amplitude(fft_res)
        print(fft_freq)
        print(fft_amp)
        for i in range(0,odesilani):
            uart.write(str(fft_freq[i])+';'+str(fft_amp[i])+chr(32))
        uart.write(chr(10))
        time.sleep_ms(1000)
        if uart.any():
            while uart.any():
                read_data=uart.read()
        if read_data==b'stopp':
            runn =False
```

Obrázek 28 Funkce živého měření

4.5.2 Rozšíření app designer

Stejně jako pro aplikaci Maixpy muselo být i do rozhraní řídicí aplikace přidán kód pro zobrazování živého přenosu dat do tabulky a grafu. Řešený je na základě přepínacího tlačítka o stavech on a off. Ty jsou zde prezentovány za pomoci dvou funkcí if. Celý zdrojový kód této funkce můžeme vidět na snímku 29.

První po splnění podmínky o sepnutí tlačítka načte komunikaci s Kendryte a spustí vnořenou funkci while. Ta načítá příchozí hodnoty a provede jejich rozřazení a promítnutí v grafickém rozhraní. Druhá funkce if má parametr nastavení tlačítka zpět do pozice off. Po tomto přepnutí se odešle příkaz k zastavení činnosti na Kendryte a následně se přeručí hlavní smyčka.

Oproti původnímu kódu bylo do programovatelných oblastí programu vložena funkce flush. Ta má za úkol vymazat veškerá data v proměnné sloužící pro načítání dat z UART. Díky tomu nedojde k chybě v měření z důvodu načtení nějakých starých dat zanechaných v bufferu.

```

98      flush(app.s)
99      value = app.ZapnoutpromtnrealtimegrafuSwitch.Value;
100     if strcmp(value, 'On')
101         write(app.s, 'stala', 'char');
102         run = true;
103     while run == true
104         a = readline(app.s); % přečti odpověď ze Sipeed
105         a=char(a);
106         app.poleampl= [];
107         app.polefreq= [];
108         n=app.Poetmen0256EditField.Value;
109         for k=1:n
110             mezera=strfind(a, ' ');
111             b=a(1:(mezera(1)-1));
112             strednik=strfind(b, ';');
113             frek=str2num (b(1:strednik(1)-1));
114             ampl=str2num (b(strednik(1)+1:size(b,2)));
115             app.poleampl=[app.poleampl ampl];
116             app.polefreq=[app.polefreq frek];
117             a=a(mezera(1)+1:size(a,2));
118         end
119         matice = [app.polefreq;app.poleampl];
120         matice = matice';
121         disp(matice)
122         app.UITable.Data = matice;
123         plot(app.UIAxes, app.polefreq, app.poleampl)
124         value = app.ZapnoutpromtnrealtimegrafuSwitch.Value;
125         if strcmp(value, 'Off')
126             write(app.s, 'stopp', 'char');
127             run = false
128         end
129     end
130 end

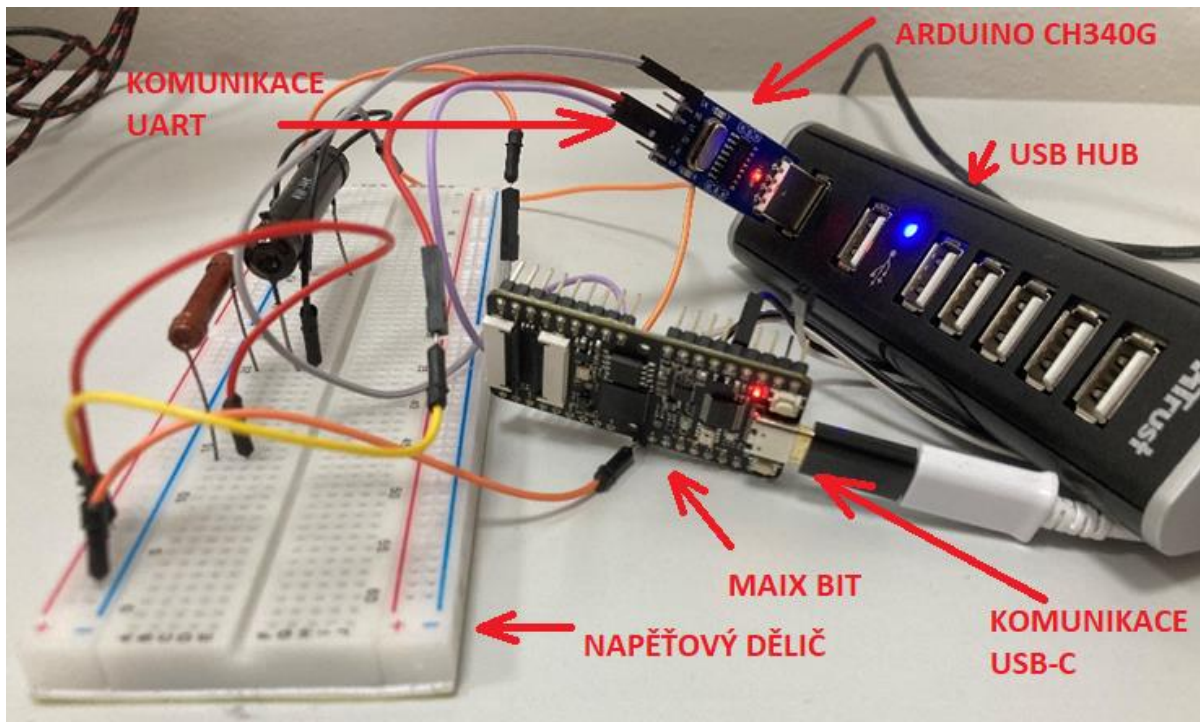
```

Obrázek 29

Funkce živého zobrazování grafu FFT app designer

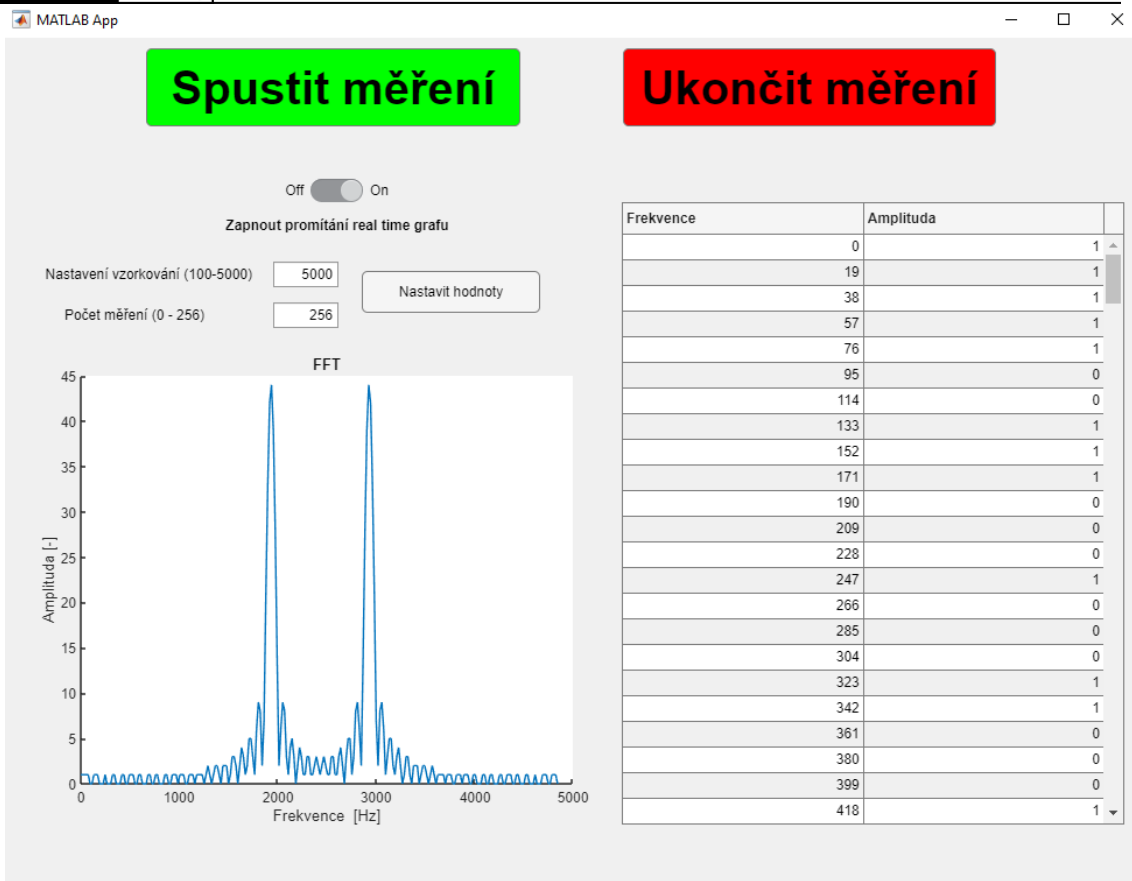
4.6 Výsledný zvukový analyzátor verze 1

Výsledné sestavení hardwarové části zařízení je popsáno na obrázku 30. Můžeme zde vidět způsob propojení jednotlivých komponent. Usb hub je využit k propojení zařízení s počítačem.



Obrázek 30 sestavení výsledného zařízení verze 1

Rozhraní uživatelské aplikace v programu Matlab app designer můžeme vidět na obrázku 31. Uspořádání je jednoduché. V horní části se nacházejí dvě tlačítka pro funkce spustit a ukončit měření. Pod tlačítkem spustit měření najdeme přepínací tlačítko pro zapnutí a vypnutí zobrazování zpracovaných dat v reálném čase. Ještě níž se nachází pole pro vyplnění požadovaného počtu měření a vzorkování spolu s tlačítkem pro nastavení těchto hodnot do Kendryte. Pro interpretaci hodnot slouží prostor grafu a tabulky, kde můžeme vidět zapsané jednotlivé hodnoty.



Obrázek 31 Ukázka realizace aplikace app designer se spuštěnou funkcí živé analýzy zvuku

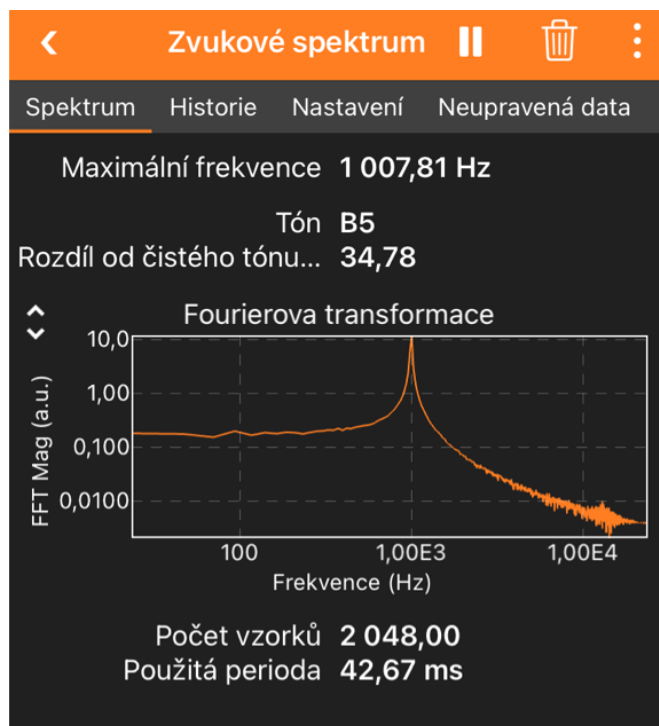
4.6.1 Ověření funkčnosti zařízení

V první řadě byly odzkoušeny všechny navržené funkce programu. Po ověření funkčnosti všech vytvořených procesů bylo přistoupeno k ověření správnosti měřených dat.

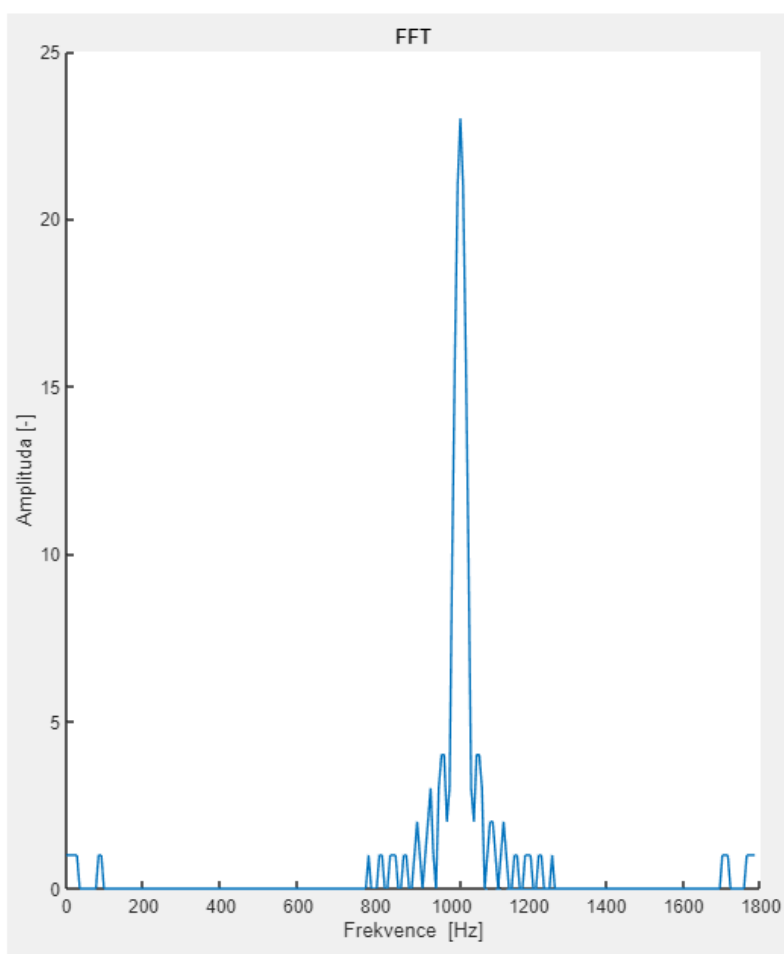
Kontrola správného fungování analýzy zvuku byla provedena na podobném principu jako jednoduchá kalibrace mikrofónů a hlukoměrů. K zařízení byl v jinak tiché místnosti přiblížen zdroj zvuku o předem dané frekvenci 1000 Hz. Po provedení měření odpovídaly výsledky analýzy nahraného zvuku akustickému zdroji. Tím byla ověřena správná funkčnost a nastavení algoritmů pro zpracování měření. Výsledek zkušebního měření můžeme vidět na obrázku 33.

Pro ověření, že zdroj akustického signálu odpovídal požadované hodnotě byla využita aplikace Phypbox v osobním telefonu. Pomocí tohoto programu byl signál zkušebně proměřen. Výsledek tohoto ověření můžeme vidět na obrázku 32, vidíme, že generovaný tón opravdu měl požadovanou hodnotu 1 kHz.

Nutno podotknout, že při měření nízkých a vysokých frekvencí může docházet k chybám. Ty mohou být způsobeny využitým mikrofónem. Na obrázku 9 (citlivost (odezva) mikrofónu pro jeho frekvenční rozsah [22]) vidíme graf, ze kterého plyne, že při měření tímto zařízením můžeme zanedbat výsledky v rozsahu spodní a vrchní hranice frekvenčního rozsahu. Konkrétně od 0 Hz do 100 Hz a v oblasti od 5 kHz dále.



Obrázek 32 Analýza provedená pomocí Phyphox na zvuku o frekvenci 1000 Hz



Obrázek 33 Výsledné zobrazení hodnot v prostředí Matlab

5 NÁVRH NA VYLEPŠENÍ

Navržené zařízení se podařilo sestavit a zprovoznit dle navržených parametrů i s dodatečnými vylepšeními. V problematice zpracování akustického signálu ovšem nedosahuje vysoké konkurenceschopnosti při porovnání se zařízeními běžně používanými v praxi. Z tohoto důvodu byla provedena metoda ke zlepšování produktů pod názvem PDCA (plan, do, check, act). Z výsledku této metody vyplývá následující tabulka 1 návrhů na zlepšení stávajícího zařízení, které se pokusím dále implementovat.

Tabulka 1 Výsledná tabulka metody PDCA

Definovaný nedostatek	Návrh zlepšení
Neschopnost zařízení určit směr zdroje	Implementace mikrofonního pole
Rozptyl měření daný šumem	Softwarová aplikace filtrů zvuku
Vzhled a kompaktnost zařízení	Zakoupení kompaktního děliče napětí pro komunikaci UART, zabudování vývojových desek do krytu
Sdílení dat	Připojení pomocí rozhraní wifi
Připojení kamery	Lepší prostorová orientace pro určení směru zdroje akustického signálu
Nemožnost zobrazit mapu akustických zdrojů a video z kamery	Připojení LCD displeje

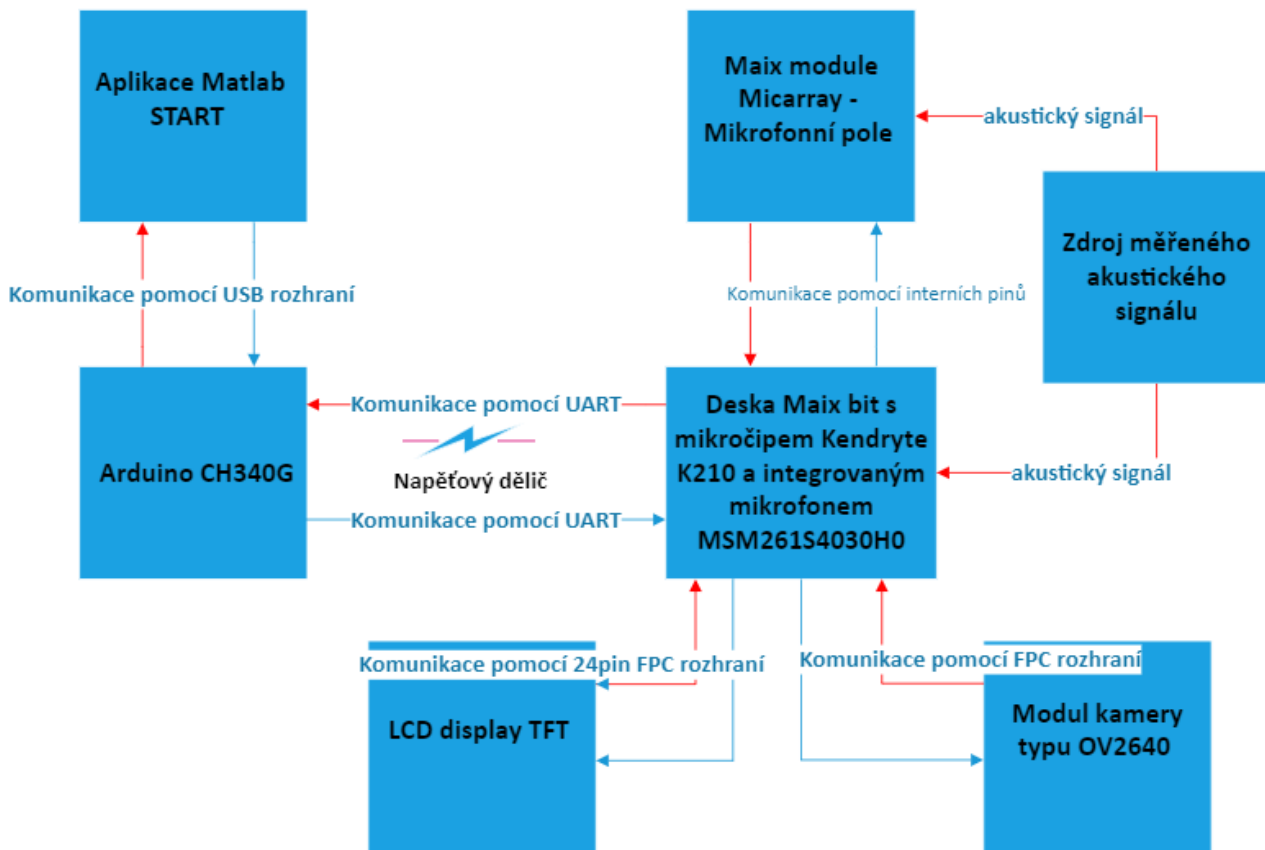
Dále byla stanovena priorita zlepšovacích návrhů na základě vypracování vah kritérií pomocí určení, které z definovaných vylepšení mají jaký stupeň obtížnosti provedení a potenciální zisk z této úpravy. Díky tomuto bylo rozhodnuto, že zařízení bude osazeno mikrofonním polem, videokamerou, LCD displejem a také bude pomocí součástí vytisknutých na 3D tiskárně upraveno uchycení jednotlivých částí. Zbylé úpravy nebudou provedeny z důvodu špatné proveditelnosti, vysoké ceny případně z důvodu nedůležitosti při zohlednění kritéria, že se jedná o školní projekt s důrazem na získání zkušeností v této oblasti.

5.1 Návrh konstrukce akustické kamery

Na základě zvolených vylepšení, která budou implementována pro zlepšení konkurenceschopnosti pomocí rozšíření funkcí zařízení byl navržen nový konstrukční diagram pro verzi zařízení dva. Tento návrh je zobrazen na obrázku 34.

Hlavní změnou je přidání tří komponent, tedy mikrofonního pole, kamery a lcd displeje. Mikrofonní pole pro určování polohy zdroje akustického signálu a kameru na vizualizaci prostoru v zorném poli mikrofónů pro lepší orientaci v prostoru při zobrazování. Na displej si můžeme nechat promítat libovolná data, buďto z kamery, nebo vypočtenou mapu akustických zdrojů z mikrofonního pole.

Platí zde stejné podmínky jako u prvního návrhu, tedy modré šipky definují výstupy dat z jednotlivých prvků a červené šipky vstupy dat do bloků.



Obrázek 34 Návrh na konstrukci vylepšené verze

5.2 Periferie využité ke konstrukci

V této kapitole budou blíže specifikovány charakteristiky periférií využitých k rozšíření první verze zařízení na akustickou kameru.

5.2.1 Mikrofonní pole Sipeed

Tato externí periferie společnosti Sipeed z řady modulů a analytických nástrojů byla vytvořena pro rozšíření možností zvukových analýz na vývojových deskách založených na mikročipu Kendryte K210 popsaného v kapitole 3.1.1. Pomocí tohoto pole lze snadněji identifikovat směrovost příchozího zvukového signálu. Jak toto pole vypadá je vidět na snímku 35.



Obrázek 35 Mikrofonní pole společnosti Sipeed Micarray [21]

Na desce je integrováno 7 mikrofonů typu MSM261S4030H0. Jeden se nachází uprostřed popisované desky. Šest dalších se pak nachází v kruhu o průměru 4 cm na okrajích desky, jak to můžeme vidět na Obr.10.). Dále zde můžeme vidět také 12 LED modulů rozmístěných také do kruhu, propojených ve dvoukanálovém zapojení. Pomocí těchto led světel je možné například vizualizovat právě směr příchozího zdroje hluku. Dokonce je podporováno rozhraní pro nastavování barvy v osmi bitech (umožňuje 256 barev) a nastavování intenzity svícení v pěti bitech (umožňuje 32 úrovní svícení). Díky tomuto můžeme i nastavovat intenzitu svícení pro odhadování intenzity hluku vizuálním vjemem [21].

Maximální hodnota měřitelné hladiny zvuku vychází z osazeného příslušenství a její hodnota je 140 dB. Taktéž citlivost odpovídá 26 dB (při 1 kHz a 1 Pa) a S/N ratio je 57 d. Pro širokou škálu uplatnění je výhodné, že lze pracovat na dvou různých taktovacích frekvencích v závislosti na spotřebě elektrické energie a přesnosti měření. V normálním módu pracuje na frekvencích od 1 do 4 MHz. Frekvence pro mód snížené spotřeby energie se pohybují v rozmezí od 150 do 800 kHz [21].

5.2.2 LCD display RY24049B0-00

Jedná se o malý lehký display připojený pomocí MCU rozhraní. Využívaný v různých aplikacích vyžadující malou spotřebu energie. Všechny potřebné údaje jsou uvedeny v tabulce 2.

Tabulka 2 Tabulka parametrů lcd displeje [29]

Parametr	Specifikace
Rozměry	42,72 x 60,26 x 2,45 [mm] (2,4" úhlopříčka)
Rozlišení	240 x 320 RGB
Svítivost	260 cd / m ²
Dotykový panel	NE
Rozhraní	8bit MCU
Pozorovací směry	Všesměrový
Skladovací teploty	-20 až 70 °C
Provozní teploty teploty	-10 až 60 °C

5.2.3 Kamera OV2640

Jedná se o malou barevnou kameru, kterou můžeme vidět na obrázku 36, určenou na připojování k externím zařízení pomocí esp32-cam modulu. V této práci je využita pro zobrazování aktuálního videozáznamu prostředí nacházejícího se v zorném poli mikrofonního pole. Důvodem je lepší orientace ve snímaném prostředí. Vzhledem ke konstrukci zařízení je kamerový záznam posunut o několik centimetrů nahoru vůči záznamu zvuku z mikrofonů. Všechny potřebné parametry jsou uvedeny v tabulce 3.



Obrázek 36 Kamera OV2640

Obrazový snímač	UXGA	1600x1200
Napájení	Jádro	1,2 V DC +5%
	Modul	2,5-3 V DC
	I/O	1,7-3,3 V
Odběr proudu	Pohotovostní	600 μ A
	Aktivní	140 mW
Teplotní rozsah	Skladování	-30 °C až 70 °C
	Pracovní	0 °C až 50 °C
Výstupní formáty (8bitové)	YUV (422/420) / YCbCr422	
	RGB565 / 555	
	8bitová komprimovaná data	
	8- / 10-bitová data RGB	
Hlavní Ray Angle	25° nelineární	
Citlivost	0,6V / Lux-sec	
Poměr S/N	40dB	
Velikost pixelu	2,2 x 2,2 μ m	
Temný proud	15mV / při 60°C	
Kapacita	12Ke	
Snímkování	UXGA / SXGA	15fps
	SVGA	30fps
	CIF	60fps

5.3 Zdrojové kódy

Pro správné fungování akustické kamery museli být přepracovány také řídicí programy. Hlavní rozdíly budou popsány v této kapitole.

5.3.1 Maixpy IDE

Původní kód první verze zařízení byl rozšířen o několik dalších funkcí. K tomuto bylo nezbytné nejprve načíst chybějící knihovnu vývojové desky, a to pro práci s mikrofonním polem. Dohromady byli přidány další tři rozhodovací funkce pro vykonání potřebných příkazů, které budou popsány podle obrázků

První je funkce pro promítání videa na lcd obrazovku. Její průběh je spuštěn příkazem „lcd_v“. Následuje restartování senzoru videokamery a nastavení potřebných parametrů snímání obrazu. Poté se senzor aktivuje a první dva snímky jsou přeskočeny z důvodu nízké kvality. Následuje nastavení správného zobrazení videa na lcd monitoru příkazem zrcadlení a rotace. Poté se spustí smyčka while, která posílá na obrazovku dostatek jednotlivých snímků k vytvoření videa. V našem případě je nastaveno 30 snímků za sekundu. Smyčka se ukončí přijetím příkazu „stop_v“. Celý kód vidíme na obrázku 37.

```

109  if read_data==b'lcd_v':
110      sensor.reset()
111      sensor.set_pixformat(sensor.RGB565)
112      sensor.set_framesize(sensor.QVGA)
113      sensor.run(1)
114      sensor.skip_frames(2)
115      lcd.mirror(True)
116      lcd.rotation(2)
117      runn=True
118  while runn == True:
119      img = sensor.snapshot()
120      lcd.display(img)
121  if uart.any():
122      while uart.any():
123          read_data=uart.read()
124  if read_data==b'stop_v':
125      runn =False

```

Obrázek 37 promítání videa na lcd displej

Pomocí obrázku 38 bude vysvětlena následující funkce if, spustitelná příkazem „lcd_s“. V prvním kroku jsou inicializovány všechny mikrofony a světla mikrofonního pole. V hlavní smyčce příkazu pak zařízení postupně vytvoří mapu zdrojů pomocí výpočtu na určování směru zdroje zvuku. A tu nám po zobrazovací úpravě stejně jako u předchozí funkce promítne na displej správně zorientovanou. Během toho zároveň rozsvítí přední led světla na základě vypočteného směru příchodu akustického signálu. K ukončení smyčky tentokrát slouží příkaz „stop_s“.

```

127  if read_data==b'lcd_s':
128      mic.init(i2s_d0=15, i2s_d1=21, i2s_d2=22,
129      runn=True
130  while runn==True:
131      imga = mic.get_map()
132      b = mic.get_dir(imga)
133      a = mic.set_led(b, (0,0,255))
134      imgb = imga.resize(180,180)
135      imgc = imgb.to_rainbow(1)
136      lcd.mirror(True)
137      lcd.rotation(2)
138      a = lcd.display(imgc)
139  if uart.any():
140      while uart.any():
141          read_data=uart.read()
142          print(read_data)
143  if read_data==b'stop_s':
144      runn =False

```

Obrázek 38 promítání akustické mapy na lcd a detekce směru zdroje

Poslední přidanou funkcí do prostředí Maixpy je možnost uložit aktuální snímky videa a akustické mapy na úložiště Maixpy. Z důvodu velké výpočetní zátěže nezvládne Maix bit zároveň předchozí funkce pro diagnostikování propojit a zároveň odesílat obrazy živě do počítače. Proto je zvolena možnost interního uložení. Proces je popsán na snímku 39. Spuštěn je příkazem „saveimg“ Jeho úkolem je provést kombinaci předchozích dvou popsaných procesů s rozdílem, že se nejedná o smyčku, ale jednou provedený příkaz. Při jeho průběhu jsou data uložena do zabudované paměti desky. V případě potřeby lze ukládat i na externí kartu.

```

146 if read_data==b'saveimg':
147     mic.init(i2s_d0=15, i2s_d1=21, i2s_d2=22,
148             sensor.reset())
149     sensor.set_pixformat(sensor.RGB565)
150     sensor.set_framesize(sensor.QVGA)
151     sensor.run(1)
152     sensor.skip_frames(2)
153     lcd.rotation(2)
154     img = sensor.snapshot()
155     img.save("kamera.jpg", quality=95)
156     imga = mic.get_map()
157     b = mic.get_dir(imga)
158     a = mic.set_led(b, (0,0,255))
159     imgb = imga.resize(180,180)
160     imgc = imgb.to_rainbow(1)
161     imgc.save("mapa.jpg", quality=95)
162     mic.deinit()
  
```

Obrázek 39 uložení aktuálního snímku videa a akustické mapy

5.3.2 App designer

Programovací část aplikace Matlab byla rozšířena tak, abychom byli schopni pohodlně ovládat všechny popsané vložené funkce.

Následující obrázek 40 se skládá z dvou částí programu, které jsou pro přehlednost zanechány odděleně pomocí řádku 165. Jejich společné zobrazení ukazuje jejich podobnost. Oba jsou tvořeny přepínacím tlačítkem. Principem fungování je odeslání zprávy po zapnutí tlačítka do polohy ON pro aktivaci dané funkce a zpět na OFF pro zastavení funkce. Důležitým prvkem je odeslání příkazu pouze v případě splnění logického členu and. Ten nám neumožní požadovat po Kendryte, aby zároveň odesílal na lcd display video i akustickou mapu. Pokud se uživatel pokusí spustit obě tlačítka najednou, bude upozorněn, že je potřeba nejdříve předchozí funkcí vypnout.

```

153     flush(app.s)
154     value_v = app.ZobrazvideoSwitch.Value;
155     value_s = app.ZobrazzvukovoumapuaurovnsrovostiLEDSwitch.Value;
156     if (strcmp(value_v, 'On') && strcmp(value_s, 'Off'))
157         write(app.s, 'lcd_v', 'char');
158     elseif strcmp(value_v, 'On')
159         fig=uifigure;
160         uialert(fig, 'Nejprve vypni promítání akustické mapy', 'Warning')
161     end
162     if strcmp(value_v, 'Off')
163         write(app.s, 'stop_v', 'char');
164     end
165 end
170     flush(app.s)
171     value_s = app.ZobrazzvukovoumapuaurovnsrovostiLEDSwitch.Value;
172     value_v = app.ZobrazvideoSwitch.Value;
173     if (strcmp(value_s, 'On') && strcmp(value_v, 'Off'))
174         write(app.s, 'lcd_s', 'char');
175     elseif strcmp(value_s, 'On')
176         fig=uifigure;
177         uialert(fig, 'Nejprve vypni promítání videa', 'Warning')
178     end
179     if strcmp(value_s, 'Off')
180         write(app.s, 'stop_s', 'char');
181     end

```

Obrázek 40 funkce pro zobrazení videa a výpočet směrovosti

Další dvě funkce, které jsou zobrazeny na snímku 41, jsou velmi jednoduché. Reagují na stisknutí tlačítka. První odešle příkaz na Kendryte pro uložení snímků do paměti. Druhá funkce na řádce 199 uloží hodnoty naměřené frekvence a amplitudy do dokumentu v počítači. Pro tuto funkci je potřeba nejprve provést spuštění měření analýzy fft.

```

192     flush(app.s)
193     write(app.s, 'saveimg', 'char');
194 end
195
196 % Button pushed function:
197 % Uložit hodnoty frekvence a amplitudy do CSV s
198 function Uložit_hodnoty_frekvence_a_amplitudy
199     writematrix(app.matice, 'app2.csv')

```

Obrázek 41 funkce uložení snímků a zápisu hodnot měření do csv dokumentu

5.4 Výsledný zvukový analyzátor verze 2

V této kapitole byla rozvedena implementace jednotlivých vylepšení aplikovaných na první verzi zařízení pro analýzu akustického signálu. Díky těmto vylepšením z tohoto zařízení vzniká

tzv. akustická kamera, která je schopna nejenom provádět diagnostiku přichozího signálu, ale také určovat směrovost zdroje signálu a vyobrazit fotografii snímaného prostoru.

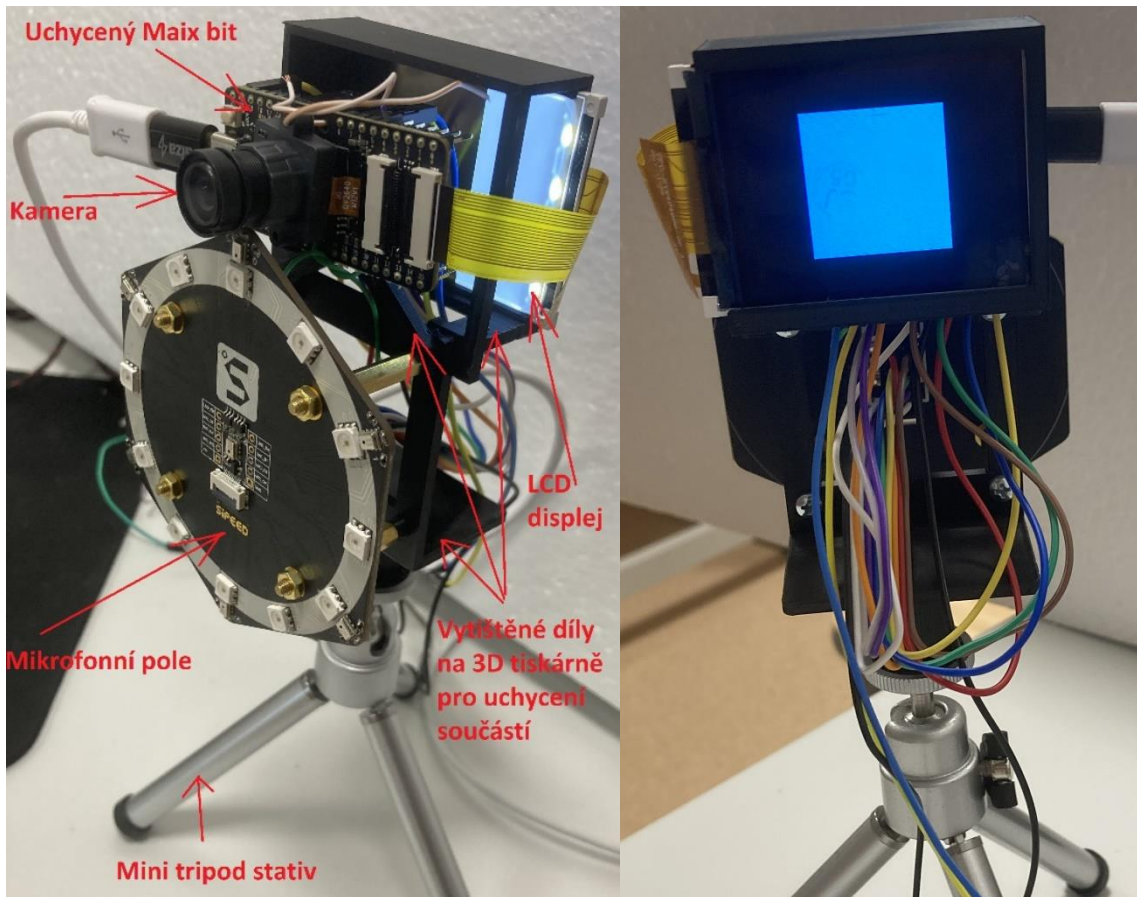
Pro správné fungování musely být upraveny také zdrojové kódy obou využívaných rozhraní a tyto změny budou vysvětleny dále v této kapitole.

Jednotlivé hardwarové prvky byly zapojeny dohromady pomocí dílů vytištěných na 3D tiskárně dle návrhu pro toto specifické použití. Sestavení celého zařízení můžeme pozorovat na dalším obrázku 42. V části a) vidíme zařízení zepředu i s popisky jednotlivých částí dále na obrázku b) vidíme pohled na zadní část zařízení, kde uživatel vidí na displeji zobrazované hodnoty.

Pro práci s touto verzí zařízení byla stanovena určitá zjednodušení k ulehčení dané problematiky. Především, že zdroje zvuku jsou statické, tedy při měření se vůči mikrofonnímu poli nepohybují a nacházejí se v jeho zorném poli. Vzdáleny dostatečně tak, aby na mikrofony dopadaly rovinné vlny. V posledním bodě počítáme s homogenním prostředím, tedy zanedbáváme útlum zvuku vlivem šíření v homogenním prostředí [27].

Druhou částí akustického analyzátoru je grafické rozhraní app designer. Jeho podobu můžeme vidět na snímku obrázku 43. Z důvodu lepší viditelnosti byla odříznuta levá část snímku. Ta zůstala beze změny a doplněná byla pouze pravá část.

Můžeme zespuď postupně vidět tlačítka na uložení hodnot frekvence a amplitud do csv souboru, na uložení aktuálních snímků na uložení v Maix bit nebo na SD kartu. Výše se nacházejí dvě přepínací tlačítka na ovládání lcd displeje, tedy která funkce bude na displeji zobrazena.



Obrázek 42 Sestavené zařízení akustická kamera a) zepředu, obrázek b) zezadu pohled na lcd displej



Obrázek 43 Výsledný vzhled rozhraní app designer

5.4.1 Ověření funkčnosti zařízení

Všechny zmíněné funkce byly postupně vyzkoušeny. Nejprve bylo provedeno měření a výsledná data byla uložena do počítače jako excelovská tabulka. Tu můžeme vidět na snímku 44.

	A	B	C
39	114	0	
40	117	0	
41	120	0	
42	123	1	
43	126	1	
44	129	1	
45	132	1	
46	135	2	
47	138	2	
48	141	1	
49	144	1	
50	147	0	
51	150	0	

Obrázek 44 export hodnot měření do excelu

Druhou funkcí jednoduchou pro vyzkoušení bylo aktivování videozáznamu na displeji. Tu můžeme vidět na snímku obrázku 45. Můžeme vidět, že na obraze jsou viditelné barevné pruhy. Toto zrnění je způsobeno problémem na kontaktech pro připojení kamery do desky.



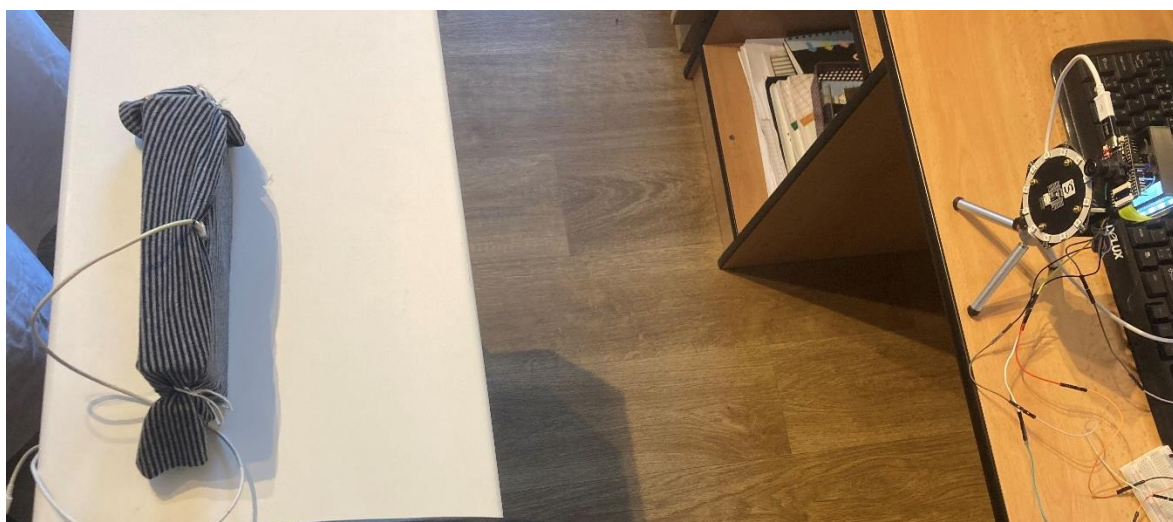
Obrázek 45 spuštěná funkce zobrazení obrazu z videokamery

Následně byla ověřena schopnost identifikace směru zdroje akustického signálu a zobrazování pomocí led světél mikrofonního pole. Pokusy s touto funkcí jsou zobrazeny na snímku 46. Na levé straně a) můžeme vidět odezvu zařízení při směřování zdroje zvuku zespodu. Na pravé části snímku b) vidíme zdroj směřovat z pravé horní části zorného pole zařízení. Tato funkce funguje správně. Vzdálenost, na kterou je zařízení schopno rozeznávat směry, je dána hlasitostí zdroje. Pro telefon na plný výkon se jedná o cca 2 metry.

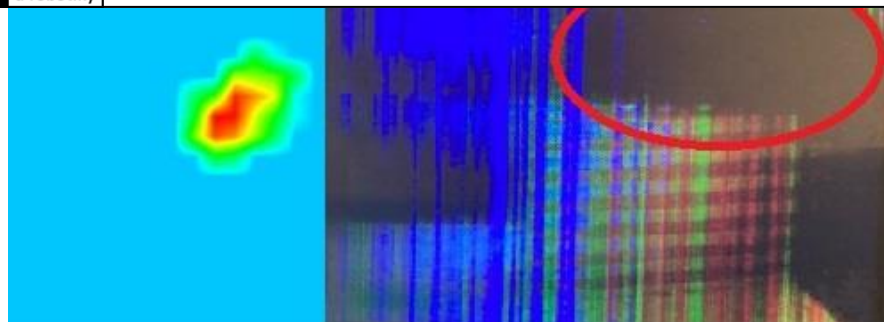


Obrázek 46 Ověřování funkčnosti rozpoznání směru zdroje akustického signálu a) signál zespodu b) signál z vrchu

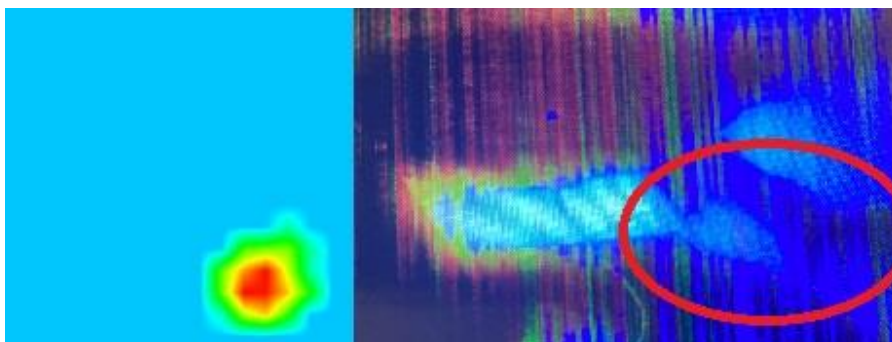
Funkce ukládání aktuálních snímků z videokamery a vypočtené akustické mapy byla odzkoušena praktickým měření zvuku ze zdroje při rozložení viz. obrázek 47. Využitý reproduktor byl postupně přesouván na obě strany nahoru i dolů, a přitom byly pořizovány snímky ukládané do paměti zařízení. Takto vzniklo 8 jednotlivých fotografií, které byly poté spojeny do odpovídajících dvojic. Ty můžeme vidět na obrázcích 48, 49, 50 a 51.



Obrázek 47 Rozestavení pro měření



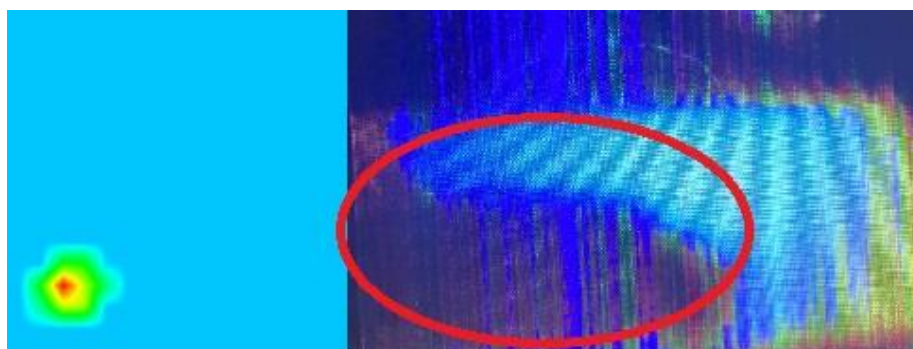
Obrázek 48 Zdroj nahoře vpravo



Obrázek 49 Zdroj dole vpravo



Obrázek 50 Zdroj nahoře vlevo



Obrázek 51 Zdroj dole vlevo

Bohužel kvalita snímku kamery je velmi nízká. Tento problém je způsoben poškozením připojovacího konektoru periferie. Nicméně pro názornou ukázkou jsou tyto snímky s úpravou využity. V červeném kroužku je vždy upřesněna pozice reproduktoru.

Vzhledem ke konstrukci zařízení je fotografie posunuta v prostoru o několik centimetrů výše vůči záznamu zvuku z mikrofonů.

Můžeme vidět, že pozice reproduktoru odpovídá zobrazení akustické mapy zdroje zvuku v prostoru. Tímto bylo odzkoušeno správné fungování i poslední implementované funkce.

5.5 Návrhy na budoucí vylepšení akustické kamery

Po sestavení a odzkoušení všech funkcí je nutné poznamenat, že zařízení je sice schopno provádět základní analýzu a vyhodnocení příchozího signálu. Stále ale disponuje velkou řadou problémů, kvůli kterým by bylo využití v technické praxi omezeno na jednoduché aplikace. Nyní budou vyjmenovány hlavní z těchto nedostatků jako návrh vylepšení pro sestavené zařízení k vylepšení její schopnosti řešit i komplexnější technické problémy.

Prvním bodem je návrh na výměnu videokamery za novou. Tím by byl odstraněn problém se špatně čitelnými fotkami. To by výrazně pomohlo v diagnostických schopnostech zařízení. Další možností by bylo pořídit kameru disponující automatickým zaostřováním, kterým použitý typ nedisponuje. Nahradit za lepší variantu by bylo záhodno i v případě napěťového děliče. Ten by šel pořídit v kompaktnější variantě tak, aby zařízení bylo lehčeji transportovatelné.

Druhým návrhem k vylepšení schopností akustické kamery je navýšení výpočetního výkonu osazením na vyšší řadu vývojové desky. Tím by se navýšili možnosti kombinace jednotlivých funkcí. Například pro funkci zobrazování akustické mapy je z důvodu nedostatku interní paměti zařízení schopno zobrazovat pouze v rozlišení 180x180 pixelů i přes rozlišení displeje 320x240pixelů. Na hranici výpočetních schopností naráží také funkce, která měla být v zařízení využita, ale nakonec to nebylo možné. Jedná se o přenášení obrazů v reálném čase do řídicího programu v počítači pomocí UART komunikace. Tato funkce nemohla být využita právě z tohoto důvodu. Navazuje na to další funkce, která nemohla být do zařízení integrována, a to provádění výpočtu FFT v reálném čase společně se všemi předchozími vyjmenovanými procesy. Souhrnně by tedy zařízení mohlo být schopno zároveň provádět výpočet FFT s přenosem do grafu, zároveň počítat směrovost zdrojů a zobrazovat v počítači akustickou mapu společně s videem z kamery.

Posledním bodem je softwarové odladění aplikací. Jedná se o velice komplikovanou záležitost zařídit, aby zařízení fungovalo zcela bez chyb a nehlásilo chybové hlásky. Bylo by tedy potřeba provést důkladný tzv. debugging, neboli odstraňování chyb v řídicích kódech. S vylepšením řídicích kódů souvisí také zavedení automatického rozpoznávání komunikačního portu v Matlab, ke kterému se přihlásilo Arduino v rámci počítače. Toto se v práci nezadařilo. Celý software je záležitostí, na kterou v reálné praxi firmy vynakládají velké finanční obnosy z důvodu přesnosti měření. Toto v případě studentské práce není možné, nicméně odladění přesnosti měření pomocí aplikace různých filtrů zpracování a šumů je v rámci dlouhého zkoušení proveditelné.

6 ZHODNOCENÍ VYUŽITÍ ZAŘÍZENÍ V PRAXI

První realizované zařízení pro analýzu zvuku, které jsem v této práci odzkoušel plnilo požadovanou funkci vyhodnocení akustického signálu. Pro zobrazení získaných dat byl použit Matlab app designer. Dále byly odzkoušeny i jeho další přidání rozšiřující funkce jako jsou ukončení měření, nastavování počtu měření a zobrazování grafu FFT aktuálně snímaného akustického signálu. Tyto funkce nevykazovali žádné procesní chyby a jsou zcela funkční.

Další částí zhodnocení zařízení v praxi je finanční konkurenceschopnost na českém trhu. Za tímto účelem byla vypočtena průměrná nákupní cena využitých součástek, která byla porovnána s cenami běžně v praxi využívaných zařízení. Tyto hodnoty byly vloženy do tabulky 4. V potaz musíme brát, že většina prodávaných zařízení disponuje více možnými funkcemi. Ovšem pro některé jednodušší aplikace se realizované zařízení jeví jako velice výhodné.

Je těžké zjistit ceny především pokročilých zvukových analyzátorů a akustických kamer. Jejich výrobci totiž produkty běžně nenabízejí jako jinou elektroniku přes internet. Ceny produktů, které se dali dohledat jsou vyplněny v tabulce níže.

Z níže uvedené tabulky vyplývá, že cena sestrojeného zařízení akustického analyzátoru (zařízení V1) se pohybuje na cenách nejlevněji prodávaných produktů na českém trhu. Ty jsou ovšem schopny měřit pouze hladinu akustického tlaku. Na toto využití tedy V1 konkurenci vyrovnává. Od produktů dražších jak 2500 Kč již můžeme pozorovat přesnější měření akustického tlaku než sestrojené zařízení. Nejlevnější prodávaná zařízení schopná zpracování alespoň základní analýzy se pohybují v hodnotě od 1400 Kč. Tímto sestrojené zařízení zdá být výhodnější variantou. Můžeme vidět, že další zvukové analyzátory jsou řádově dražší. Je to způsobeno vysokou náročností zpracování a odladění řídicích a analyzačních software řešení pro dané produkty.

Akustické kamery jsou velice drahými nástroji. V praxi si tyto zařízení vzhledem k ceně pořizují pouze největší společnosti na světě. Některé firmy však nabízejí možnosti zapůjčení zařízení, případně kompletní služby měření. Z tohoto důvodu by mohla být sestavená akustická kamera (zařízení V2) navrženého zařízení po vylepšení přesnosti měření dobrý pomocník na některé jednodušší aplikace v mapování zdrojů zvuku.

Tabulka 4 Ffinanční zhodnocení konkurenceschopnosti

Součástka	Cena [Kč]	Typ a označení zařízení	Cena [Kč]	Parametry
Komunikační převodník	180	Zvukoměr UNI-T UT353	650	Pouze akustický tlak (a.t.)
Maix bit v2	380	Zvukoměr FSM 130+	2541	Přesnější měření (a.t.)
Breadboard, odpory, spojovací dráty	100	Chauvin Arnoux C.A 1310	16390	Velice přesné (a.t.)
Zařízení V 1 celkem	660	Zvukový analyzátor Whadda WSAH8098	1400	FFT základní funkce
Mikrofonní pole	470	Zvukový analyzátor NTI Audio ML	22 000	Přesná diagnostika
Vytisknutí 3D dílů, spojovací materiál	200	Zvukový analyzátor EXTECH 407790A	110 000	Více druhů algoritmů
Kamera	100	Akustická kamera Polycom 2624	5800	Funkce zoom, nikoli diagnostika FFT
Lcd display	170	Akustická kamera Flir SI124	330 000	Dobré řešení útlumu šumů
Zařízení V 2 celkem	1600	Akustická kamera SoundCam	325 000	Různé druhy mikrofonního pole
Matlab	Studentská verze	Akustická kamera Fluke ii910	631 000	Detekce výbojů, úniků z tlakových zařízení

7 ZÁVĚR

Tato práce byla zpracována na základních principech systémového přístupu. V úvodu práce byla formou rešerše popsána oblast akustiky, vytyčení základních veličin a pojmů, způsob zpracování zvuku pro lidské ucho. Rovněž jsou popsány jednotlivé nástroje a techniky kterými je možné provést analýzu zvuku. Dále jsou zde popsány jednotlivé nejpoužívanější metody a algoritmy pro zpracování zvukového signálu

Praktická část této práce je věnována analýze zvuku na vývojové platformě Sipeed za pomoci hardwaru Maix bit. Po zpracování výše uvedené problematiky je proveden návrh zařízení pro získání a analýzu zvuku na zmíněném hardware a proveden návrh na vylepšení již funkčního zařízení, a to rozšířením na akustickou kameru.

Další možností, jakým způsobem dosáhnout lepšího praktického využití se nabízí zakomponování zvukových filtrů a čištění šupu. Běžně se tato filtrování provádí na analogovém signálu v časovém pásmu. Vzhledem k integrovanému mikrofonu technologie MEMS, kdy je naměřený signál ihned převáděn na digitální formu pro přenos na I²S port nelze tyto filtry do obvodu zařadit. Pro využití těchto filtrů, a tím zlepšení analytických vlastností tohoto zařízení v diagnostické praxi, by bylo vhodné využít externího zařízení pro vykonání těchto částí. Případně upravit datový výstup z mikrofonu a tato data pak dále zpracovávat.

Deska Maix Bit vyniká především díky zabudované technologii na čipu Kendryte K210 tzv. Fast Fourier Transform Accelerator. Díky níž dokáže zpracovávat velké množství dat. Zároveň tím nabízí možnost aplikovat i složitější zvukové analýzy pomocí metod navazujících na výpočet parametrů FFT a DFT.

Modulárnímu systém vývojových desek Sipeed představuje možnost ve zlepšení zařízení například dopojením modulu pro Wifi komunikaci a zasilání zpracovaných dat například na cloud uložení, které tato firma poskytuje a tím usnadňuje jejich zpracování.

Pro zvýšení využití realizovaného zařízení v praxi by bylo záhodno aplikovat postupně navržená vylepšení. Tímto by se rozšířila možnost využití na pestřejší paletu různých technických problémů.

I přes zjištěné výsledky se v průběhu zpracování praktické části zjistilo více různých problémů s kompatibilitou a stabilitou programů a zdrojových kódů. Tyto problémy mají za následek omezení využitelnosti zařízení. Jejich kompletní odstranění by bylo časově i finančně velmi náročné.

Výsledky měření mohou být zásadně ovlivněny. jak použitou přístrojovou technikou, tak metodikou měření. V mezinárodních normách jsou stanoveny zásady a podmínky měření, které je nutné v každé etapě od zpracování dat až po samotné vyhodnocování dodržovat. Tyto normy byli v průběhu návrhu a zpracování práce brány v potaz.

8 SEZNAM POUŽITÝCH ZDROJŮ

- [1] BRÜEL & KJÆR, Základy měření hluku, Sborník z konference v Hrotovicích, 2012.
- [2] TŮMA, Jiří. *Zpracování signálů získaných z mechanických systémů užitím FFT*. Praha: Sdělovací technika, 1997. ISBN 80-901-9361-7
- [3] MILÁČEK, Stanislav. *Měření a vyhodnocování mechanických veličin*. Praha: Vydavatelství ČVUT, 2001. ISBN 80-010-2417-2
- [4] REIDL, Marcel a Radislav ŠMÍD. *Technická diagnostika: senzory, metody, analýza signálu*. 4. díl. Praha: BEN-technická literatura, 2006. *Senzory neelektrických veličin*. ISBN 80-730-0158-6.
- [5] Cs.wikipedia [online]. [cit. 2022-02-01]. Dostupné z: <https://cs.wikipedia.org/wiki/Zvuk>
- [6] KŘENEK, Jakub. *Spektrální analýza zvuků v prostorové akustice*. Brno, 2013. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. Jiří Rozman, CSc.
- [7] ŠŤASTNÍK, S., ZACH, J., *Stavební akustika a zvukoizolační materiály*. Brno: Akademické nakladatelství CERM, s.r.o., 2002. 210 s. ISBN 80-214-2117-7
- [8] PORT, Martin. *SPEKTRÁLNÍ ANALÝZA ZVUKŮ V PROSTOROVÉ AKUSTICE*. Brno, 2011. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. Jiří Rozman, CSc.
- [9] PTÁČEK, Ladislav. *Technické kukátko-zvukoměr vs. hlukoměr*. <https://www.casopismuzikus.cz/workshopy/technicke-kukatko-zvukomer-vs-hlukomer> [online]. Praha: Muzikus, 2018, 3.9.2018 [cit. 2022-04-22]. Dostupné z: <https://www.casopismuzikus.cz/workshopy/technicke-kukatko-zvukomer-vs-hlukomer>
- [10] *Zvukový analyzátor Nor145*. In: EKOLA group, spol. s r.o. [online]. Praha: Ekolagroup, 2020 [cit. 2022-04-25]. Dostupné z: <http://www.ekolagroup.cz/cs/zvukovy-analyzator-nor145/>
- [11] JEDLIČKA, Petr. *Hlukoměr a detektor zvuků*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Stanislav Horký.
- [12] *Technika nahrávání – mikrofon*. In: *Muzikanti a kapely* [online]. Praha: Muzikanti a kapely, 2018 [cit. 2022-04-25]. Dostupné z: <https://muzikantiakapely.cz/magazin/technika-nahravani-mikrofon/>
- [13] *Identifikace zdrojů hluku. Národní referenční laboratoř pro komunální hluk* [online]. Ústí nad Orlicí: Zdravotní ústav se sídlem v Ostravě, 2015 [cit. 2022-04-28]. Dostupné z: <http://hluk.nrl.cz/Home/Novinka?id=111>
- [14] *Frekvenční analýza*. Ekosoftware [online]. Liberec: Ekosoftware s.r.o, 2020 [cit. 2022-04-29]. Dostupné z: <https://www.ekosoftware.cz/frekvencni-analyza>
- [15] Wiki.sipeed.com [online]. Baoan (Šen-čen, Čína): Sipeed, c2018–2021 [cit. 2022-02-16]. Dostupné z: <https://wiki.sipeed.com/soft/maixpy/en/index.html>
- [16] Sipeed. Sipeed [online]. Baoan, (Šen-čen, Čína): Sipeed, 2020 [cit. 2022-04-22]. Dostupné z: <https://www.sipeed.com/index-en.html>

- [17] Kendryte K210. Canaan [online]. Beijing, China: Canaan, 16n. 1. [cit. 2022-04-23]. Dostupné z: <https://canaan.io/product/kendryteai>
- [18] Twitter [online]. Šen-čen, Čína: Sipeed, 2018 [cit. 2022-04-23]. Dostupné z: <https://mobile.twitter.com/sipeedio>
- [19] Maix Bit. Sipeed [online]. Šen-čen, Čína: Sipeed, 2019 [cit. 2022-05-02]. Dostupné z: <https://1url.cz/DrMhS>
- [20] Maix-BIT v2.0 Specifications: Pdf soubor převzato z [19]. Šen-čen, Čína, 2019. Dostupné také z: <https://1url.cz/grMhO>
- [21] MicArray Microphone Array. Sipeed [online]. Šen-čen, Čína: Sipeed, 2019 [cit. 2022-04-21]. Dostupné z: <https://1url.cz/SrMh5>
- [22] MEMS microphone Data sheet. Šen-čen, Čína, 2015. Dostupné také z: <https://1url.cz/2rMhb>
- [23] MaixHub. MaixHub [online]. Baoan, (Šen-čen, Čína): Sipeed, 2019 [cit. 2022-04-30]. Dostupné z: <https://www.maixhub.com/>
- [24] GitHub. GitHub [online]. Šen-čen, Čína: Sipeed, 2022 [cit. 2022-04-30]. Dostupné z: <https://github.com/sipeed>
- [25] Geekbuing. Geekbuying [online]. Šen-čen, Čína: Geekbuying, 2012 [cit. 2022-04-30]. Dostupné z: <https://1url.cz/RrMxb>
- [26] Hercules. HW group [online]. Praha-Újezd: HW group, 2020 [cit. 2022-05-04]. Dostupné z: <https://www.hw-group.com/cs/software/aplikace-hercules-setup>
- [27] Lokalizace vzdáleného zdroje zvuku polem mikrofonů. Brno, 2011. Bakalářská práce. Vysoké učení technické v Brně FEKT. Vedoucí práce Ing. Zdeněk Havránek, Ph.D.
- [28] Láska kit. Lásakakit [online]. Rychnov nad Kněžnou: Lásakakit, 2022 [cit. 2022-05-01]. Dostupné z: <https://1url.cz/CrKb6>
- [29] Seedstudio. Seedstudio [online]. Shenzhen, Čína: Seeed Technology Co., 2008 [cit. 2022-05-18]. Dostupné z: <https://www.seeedstudio.com/2-4-TFT-LCD-p-4049.html>

9 SEZNAM OBRÁZKŮ A TABULEK

9.1 Seznam tabulek

Tabulka 1 Výsledná tabulka metody PDCA	59
Tabulka 2 Tabulka parametrů lcd displeje [29]	62
Tabulka 3 Specifikace kamery OV2640 [28].....	63
Tabulka 4 Finanční zhodnocení konkurenceschopnosti	74

9.2 Seznam obrázků

Obrázek 1 Schéma kondenzátorového mikrofonu [12].....	21
Obrázek 2 Zvukoměr/zvukový analyzátor Nor 145 [10]	24
Obrázek 3 Ukázka měření akustickou kamerou [13]”	25
Obrázek 4 Příklad CPB analýzy [14]	29
Obrázek 5 Grafické znázornění kepstrální analýzy [4]	30
Obrázek 6 Rozlišitelnost v čase a frekvenci [4]	31
Obrázek 7 Vývojová deska Maix Bit v2.0 [20].....	33
Obrázek 8 Mikrofon MSM osazený na desce Maix Bit a mikrofonním poli MicArray [22]	34
Obrázek 9 citlivost (odezva) mikrofonu pro jeho frekvenční rozsah [22].....	34
Obrázek 10 Printscreen aplikace MaixPy IDE s viditelným výběrem desek.....	35
Obrázek 11 Možnosti komunikace programu s čipem vývojové desky.....	36
Obrázek 12 Rozhraní aplikace Hercules	38
Obrázek 13 Ukázka rozhraní app designer.....	39
Obrázek 14 Vývojový diagram návrhu autonomního zařízení.....	42
Obrázek 15 Navrhovaná konstrukce zařízení.....	43
Obrázek 16 Napěťový dělič.....	44
Obrázek 17 Výkres zapojení děliče napětí	44
Obrázek 18 Arduino CH340G USB to TTL Adapter Module	45
Obrázek 19 Náhled kódu MaixPy 1/4	46
Obrázek 20 Náhled kódu MaixPy 2/4	47
Obrázek 21 Náhled kódu MaixPy 3/4	49
Obrázek 22 Náhled kódu MaixPy 4/4	49
Obrázek 23 Program matlab app designer 1/3	50

Obrázek 24	Program matlab app designer 2/3	51
Obrázek 25	Program matlab app designer 3/3	52
Obrázek 26	Návrh vylepšení funkcí zařízení	53
Obrázek 27	Nnačtení Led.....	54
Obrázek 28	Funkce živého měření.....	54
Obrázek 29	Funkce živého zobrazování grafu FFT app designer.....	55
Obrázek 30	sestavení výsledného zařízení verze 1	56
Obrázek 31	Ukázka realizace aplikace app designer se spuštěnou funkcí živé analýzy zvuku	57
Obrázek 32	Analýza provedená pomocí Phypox na zvuku o frekvenci 1000 Hz	58
Obrázek 33	<i>Výsledné zobrazení hodnot v prostředí Matlab</i>	58
Obrázek 34	Návrh na konstrukci vylepšené verze	60
Obrázek 35	Mikrofonní pole společnosti Sipeed Micarray [21].....	61
Obrázek 36	<i>Kamera OV2640.....</i>	62
Obrázek 37	promítání videa na lcd displej	64
Obrázek 38	promítání akustické mapy na lcd a detekce směru zdroje	64
Obrázek 39	uložení aktuálního snímku videa a akustické mapy	65
Obrázek 40	funkce pro zobrazení videa a výpočet směrovosti.....	66
Obrázek 41	funkce uložení snímků a zápisu hodnot měření do csv dokumentu	66
Obrázek 42	<i>Sestavené zařízení akustická kamera a) zepředu, obrázek b) zezadu pohled na lcd displej.....</i>	68
Obrázek 43	Výsledný vzhled rozhraní app designer	68
Obrázek 44	export hodnot měření do excelu	69
Obrázek 45	spuštěná funkce zobrazení obrazu z videokamery	69
Obrázek 46	Ověřování funkčnosti rozpoznání směru zdroje akustického signálu a) signál zespodu b) signál z vrchu	70
Obrázek 47	Rozestavení pro měření	70
Obrázek 48	Zdroj nahoře vpravo	71
Obrázek 49	Zdroj dole vpravo	71
Obrázek 50	Zdroj nahoře vlevo.....	71
Obrázek 51	Zdroj dole vlevo.....	71

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MEMS	Micro electro mechanical system
AD převodník	Analog digital převodník
SPL	Sound presure level
4G	Čtvrtá generace bezdrátového spojení
CPB	Constant percentage bandwidth
FFT	Fast Fourier transformation
AIoT	Artificial intelligence of things
IoT	Internet of things
AI	Artificial intelligence
USB-C	Universal serial bus typ C
PC	Personal computer
UART	Universal asynchronous receiver transmitter
I ² S	Inter sound
LCD	Liquid crystal display
Micro SD	Micro secure digital
SRAM	Static random access memory
C, C++	Programming languages
iOS	iphone operating system
IDE	Integrated development environment
RX	Recieve
TX	Transmit
USB HUB	USB distributor
MV	Minimum variance
AR	autoregresive modeling
MUSIC	Multiple signal classification
PDCA	metoda plan, do, check, act

10 SEZNAM PŘÍLOH

Příloha 1 Archyv se zdrojovými kódy