



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **ANALYZÁTOR SIGNÁLU ZALOŽENÝ NA DVB-T USB TUNERU**

SIGNAL ANALYSER BASED ON DVB-T USB TUNER

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN BELICA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. Dr. Ing. JAN ČERNOCKÝ,**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Belica Martin**

Obor: Informační technologie

Téma: **Analyzátor signálu založený na DVB-T USB tuneru**  
**Signal Analyzer Based on DVB-T USB Tuner**

Kategorie: Zpracování signálů

Pokyny:

1. Nastudujte problematiku zpracování a demodulace signálů.
2. Seznamte se s existujícími řešeními softwarově definovaného rádia (SDR).
3. Vyberte vhodný driver pro USB TV RTL Tuner pro Windows nebo Linux.
4. Navrhněte a naprogramujte modulární systém pro zpracování signálu z tuneru obsahující vybrané techniky analýzy signálu a vizualizaci.
5. Otestujte systém na datech z reálného provozu.
6. Vytvořte plakátek a/nebo krátké video demonstrující Vaši práci.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Černocký Jan, doc. Dr. Ing.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 66 Brno, Štefánekova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Táto práca sa zaoberá problematikou detekcie a analyzovania rádiových signálov pomocou DVB-T USB tunera. Na analýzu signálu je nutné najprv nejaký signál zachytiť, v tomto prípade pomocou USB tunera. Následne je možné nad zachyteným signálom aplikovať rôzne metódy, ktoré nám poskytnú potrebné informácie o danom signále. Základom analýzy je diskretná Fourierova transformácia, pomocou ktorej sa zobrazuje spektrum signálu. Na záver sa aplikujú rôzne demodulačné a filtračné algoritmy, ktoré pomáhajú analyzovaný signál lepšie interpretovať užívateľovi.

## Abstract

This bachelor thesis deals with the detection and analysis of a radio signals with the DVB-T USB tuner. The first step of the analysis is to capture some radio signal, in this case using a USB tuner. The next step is the application of signal processing algorithms. The most important algorithm for analysis is a discrete Fourier transform that is used to obtain a signal spectrum. Finally, several demodulation and filtering algorithms are used to better interpret the analyzed signal to the user.

## Kľúčové slová

analýza signálov, spracovanie signálov, USB DVB-T tuner, RTL2832, SDR, RTL-SDR, softvérovo definované rádio

## Keywords

signal analysis, signal processing, USB DVB-T tuner, RTL2832, SDR, RTL-SDR, software defined radio

## Citácia

BELICA, Martin. *Analýzátor signálu založený na DVB-T USB tuneru*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Dr. Ing. Jan Černocký,

# Analyzátor signálu založený na DVB-T USB tuneru

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána doc. Dr. Ing. Jana Černockého. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
Martin Belica  
15. mája 2017

## Poďakovanie

Ďakujem môjmu vedúcemu práce, pánovi doc. Dr. Ing. Janovi Černockému, za vedenie tejto práce, za pomoc a cenné rady, ktoré mi poskytol pri jej tvorbe. Ďalej by som sa rád poďakoval môjmu kamarátovi Renému Činčurovi, za pomoc pri programovaní aplikácie.

# Obsah

<b>Zoznam použitých skratiek</b>	<b>3</b>
<b>1 Úvod</b>	<b>4</b>
<b>2 Predstavenie témy</b>	<b>5</b>
2.1 SDR	5
2.2 USB DVB-T tuner (RTL-SDR)	6
2.3 Existujúce riešenia	7
2.3.1 HDSDR	8
2.3.2 Gqrx SDR	9
<b>3 Teoretické základy SDR</b>	<b>10</b>
3.1 I/Q signál	10
3.2 Rýchla Fourierova transformácia	11
3.3 Oknová funkcia	12
3.4 Rádiové signály	14
3.4.1 Antény	15
3.5 Modulácia a demodulácia signálu	15
3.5.1 Analógové modulácie	15
3.5.2 Digitálne modulácie	18
<b>4 Návrh riešenia</b>	<b>19</b>
4.1 Bloková schéma aplikácie	19
4.2 Práca s USB tunerom	19
4.3 Spracovanie signálu	20
4.4 Uživatelské rozhranie	21
<b>5 Použité technológie</b>	<b>23</b>
5.1 Jazyk C++	23
5.1.1 Knižnica Simple-FFT	23
5.2 Qt framework	24
5.2.1 Knižnica QCustomPlot	24
5.2.2 OpenGL	24
5.3 Ovládač	25
<b>6 Implementácia a riešenie problému</b>	<b>26</b>
6.1 Zabezpečenie komunikácie s ovládačom	26
6.2 Implementácia spektrálnej analýzy	26

6.3	Vykresľovanie grafu a spektrogramu . . . . .	27
6.4	Výsledné užívateľské rozhranie . . . . .	28
<b>7</b>	<b>Testovanie</b>	<b>29</b>
7.1	Rozhlasové vysielanie . . . . .	29
7.2	Dialkové odomykanie automobilu . . . . .	30
7.3	PMR rádiostanica . . . . .	31
7.4	Teplotná odolnosť . . . . .	31
7.5	Vyhodnotenie testovania . . . . .	32
<b>8</b>	<b>Záver</b>	<b>33</b>
	<b>Literatúra</b>	<b>34</b>
	<b>Prílohy</b>	<b>35</b>
<b>A</b>	<b>Obsah priloženého pamäťového média</b>	<b>36</b>
<b>B</b>	<b>Inštalácia a spustenie</b>	<b>37</b>
<b>C</b>	<b>Návod na obsluhu</b>	<b>38</b>
<b>D</b>	<b>Plagát z konferencie Excel@FIT 2017</b>	<b>39</b>

# Zoznam použitých skratiek

**A/D** - Analog to Digital, analógovo-digitálny (prevodník)  
**AFC** - Automatic Frequency Control, automatické doladovanie frekvencie  
**AGC** - Automatic Gain Control, automatické vyrovnávanie citlivosti  
**AM** - Amplitude Modulation, amplitúdová modulácia  
**ASK** - Amplitude-shift Keying, amplitúdové klúčovanie  
**DFT** - Discrete Fourier Transform, diskrétna Fourierova transformácia  
**DVB-T** - Digital Video Broadcasting – Terrestrial, pozemné digitálne vysielanie  
**EHF** - Extremely High Frequency, milimetrové vlny  
**FFT** - Fast Fourier Transform, rýchla Fourierova transformácia  
**FM** - Frequency Modulation, frekvenčná modulácia  
**FSK** - Frequency-shift Keying, frekvenčné klúčovanie  
**GSM** - Global System for Mobile Communications, globálny systém mobilných komunikácií  
**HF** - High Frequency, krátke vlny  
**I/Q** - In-phase/Quadrature, synfázový a kvadratúrny (signál)  
**LF** - Low Frequency, dlhé vlny  
**LSB** - Lower Sideband, dolné postranné pásmo  
**MF** - Medium Frequency, stredné vlny  
**Msp/s** - Mega samples per second, počet miliónov vzoriek za sekundu  
**PM** - Phase Modulation, fázová modulácia  
**PMR** - Public Mobile Radio, súkromná mobilná rádiová služba  
**PSK** - Phase-shift Keying, fázové klúčovanie  
**QAM** - Quadrature Amplitude Modulation, kvadrátne amplitúdová modulácia  
**RTL** - Realtek  
**RTL-SDR** - Realtek Software Defined Radio  
**SDR** - Software Defined Radio, softvérovo definované rádio  
**SHF** - Super High Frequency, centimetrové vlny  
**SSB** - Single Sideband modulation, modulácia s jedným postranným pásmom  
**TCP** - Transmission Control Protocol, protokol riadenia prenosu  
**UDP** - User Datagram Protocol  
**UHF** - Ultra High Frequency, ultra krátke vlny  
**USB** - Upper Sideband, horné postranné pásmo  
**USB** - Universal Serial Bus, univerzálna sériová zbernica  
**VHF** - Very High Frequency, veľmi krátke vlny  
**VLF** - Very Low Frequency, veľmi dlhé vlny  
**WAV** - Waveform Audio File Format, bezstratový zvukový formát

# Kapitola 1

## Úvod

Zachytávanie signálov pomocou jednoduchého prijímača a následné spracovanie týchto signálov pomocou počítača resp. softvéru je moderná rádiová technológia, ktorá sa nazýva *SDR (software defined radio)*.

Rádiový prijímač je v tomto prípade veľmi jednoduché zariadenie, ktoré obsahuje iba zopár vstupných obvodov a signál je ďalej spracovaný softvérovo. To znamená, že v prípade technológie SDR, prijímač neobsahuje žiadne ďalšie obvody, ktoré by zachytávaný signál ďalej spracovávali. Výstupom tohto prijímača je 8 bitový I/Q (in-phase/quadrature) signál zachytený na zadanej frekvencii. Použitie takéhoto jednoduchého zariadenia na účely SDR s čipsetom RTL sa nazýva RTL-SDR

Softvérovo definované rádio je technológia, ktorá sa začala používať iba nedávno aj napriek tomu, že tento pojem je známy už desiatky rokov. Používanie SDR sa rozšírilo hlavne z dôvodu dostupnosti veľkého výpočetného výkonu dnešných počítačov. Technológie SDR a RTL-SDR sú popísané v kapitole 2.

Zachytený signál je pomocou ovládača privedený do počítača, kde sa ďalej spracováva pomocou softvérových algoritmov ako je napr. výpočet rýchlej Fourierovej transformácie alebo demodulácia. Na analýzu signálu je vhodné analyzovaný signál nejakým spôsobom graficky interpretovať, napríklad formou grafov alebo štatistík. Konkrétne sa budeme zaoberať grafom spektra a spektrogramom. V prípade zachytávania zvukového signálu je dôležité vedieť tento zvuk prehrať a pod. Teoretickým základom SDR sa venuje kapitola 3

Cieľom tejto práce je vytvoriť aplikáciu, ktorá dokáže zachytiť signál pomocou USB tunera na požadovanej nosnej frekvencii s istou šírkou pásma a ďalšími parametrami. Aplikácia musí následne umožniť zachytený signál spracovať pomocou algoritmov používaných pri SDR a spracovaný signál interpretovať užívateľovi. Návrh riešenia je ukázaný v kapitole 4.

Na tvorbu aplikácie je potrebné zvoliť vhodný programovací jazyk a taktiež rôzne knižnice, ktoré uľahčia prácu pri programovaní. Použité technológie pri implementácii tejto práce je možné nájsť v kapitole 5.

Jadru práce, ktorou je implementácia samotnej aplikácie je určená kapitola 6, v ktorej je znázornená bloková schéma celého systému a taktiež sú tu popísané všetky implementačné detaily, použité pri tvorbe tejto práce.

Po vytvorení funkčného systému je veľmi dôležité vykonať testovanie. Toto testovanie má za účel odhaliť možné chyby či už implementácie, alebo aj samotného hardvéru. Pri testovaní môžu nastať chyby, ktoré nie sú odstrániteľné (hlavne hardvérové), ale aj chyby závislé od okolia. O testovaní aplikácie je možné sa dočítať v kapitole 7.

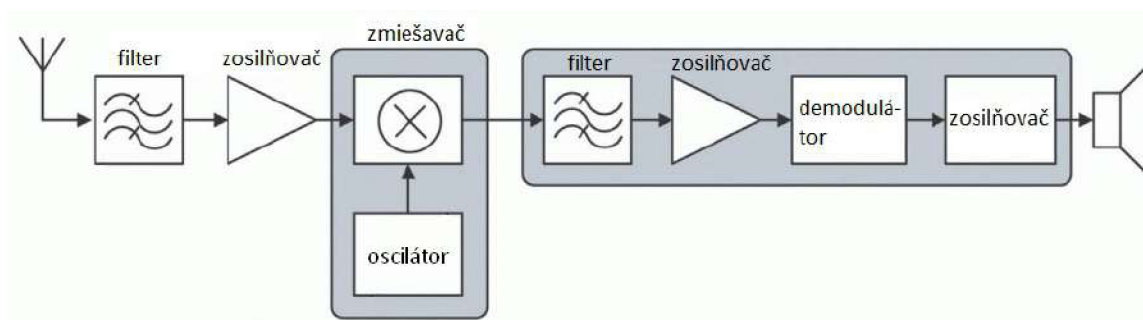


## Kapitola 2

# Predstavenie témy

### 2.1 SDR

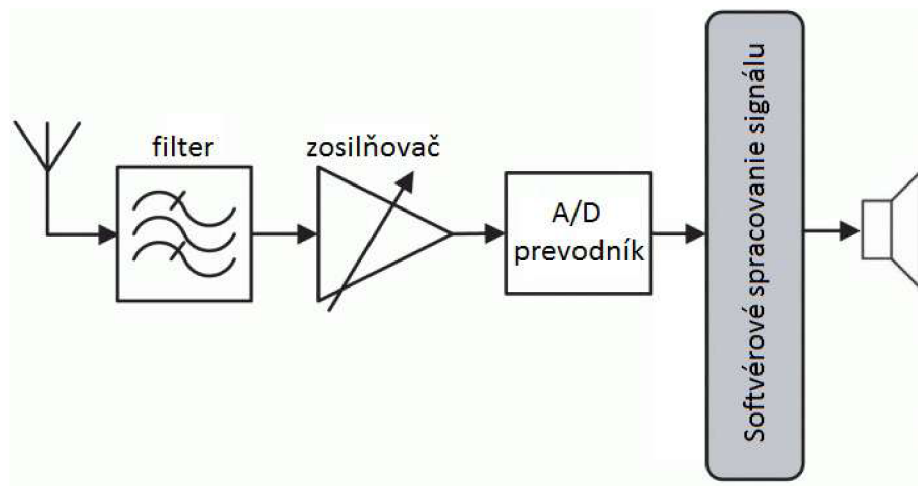
SDR [5] je skratka pre Software Defined Radio, čo je v súčasnosti moderná technológia v oblasti rádiovkej komunikácie. V tomto systéme sú všetky jeho komponenty ako napr. filtrácia, demodulácia atď. implementované pomocou softvéru na osobných počítačoch alebo vo vstavaných zariadeniach. Tento stav sa dá dosiahnuť tak, že signál premeníme do digitálnej podoby čo najbližšie pri anténe. Potom tieto čísla spracujeme v počítači matematickými postupmi a algoritmami. Pojem SDR je známy už relatívne dlho. Približne v roku 1992 vznikli prvé odborné články. Ale donedávna sa o SDR hovorilo iba na teoretickej úrovni, pretože v minulosti sa všetky komponenty rádiovkej komunikácie realizovali pomocou špecializovaného hardvéru. Nástupom vysokého výpočetného výkonu za relatívne malú cenu sa táto teória môže využívať aj v praxi. Približne v roku 2002 vzniklo prvé komerčné riešenie SDR. Po desiatich rokoch sa definícia SDR ustálila a odvtedy sa táto technológia využíva hlavne v oblastiach armády, telekomunikačného priemyslu, ale aj na rádioamatérske účely.



Obr. 2.1: Bloková schéma rádioprijímača realizovaného pomocou hardvérových komponentov

Na obrázku 2.1 je zobrazená bloková schéma bežného rádioprijímača<sup>1</sup>. Z obrázku je zrejmé, že aj jednoduché zariadenie ako je rádioprijímač obsahuje množstvo hardvérových komponentov, pričom každý z nich vykonáva špecifickú činnosť. V prípade, že by sme chceli zmeniť funkčnosť tohto systému napr. zmeniť typ demodulácie, museli by sme pridať ďalšiu súčiastku alebo vymeniť pôvodnú za novú.

<sup>1</sup>Obrázok prevzatý z: [http://pandatron.cz/elektronika2/psdr\\_blok1.gif](http://pandatron.cz/elektronika2/psdr_blok1.gif)



Obr. 2.2: Bloková schéma zariadenia na princípe SDR

V prípade technológie SDR (obrázok 2.2)<sup>2</sup> je celý systém flexibilný a dá sa prispôbiť presne na požadovaný účel, bez nutnosti pridávania alebo výmeny hardvérových komponentov. Samozrejme, sú tu isté obmedzenia, ktoré sa týkajú hardvéru (kvalita, rýchlosť a presnosť zachytávania signálu) a tiež nutnosti naprogramovať potrebné algoritmy.

## 2.2 USB DVB-T tuner (RTL-SDR)

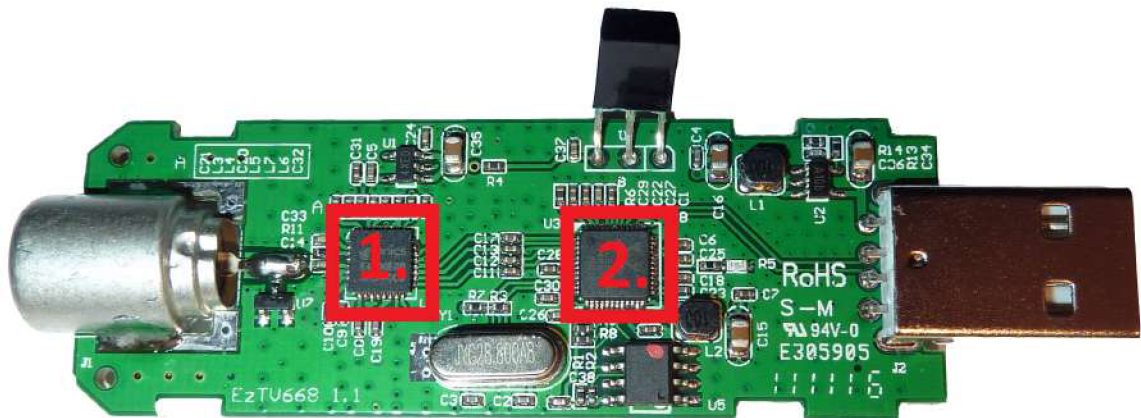
V tejto práci budeme na zachytávanie signálu používať USB DVB-T tuner. Konkrétne s využitím čipu RTL2832U, z čoho pochádza pojem RTL-SDR. Primárne sa toto zariadenie používa na sledovanie televízie alebo počúvanie rádia na osobnom počítači. My sa ale budeme snažiť využiť potenciál tohto tunera naplno. A to tak, že zachytávaný signál nebude iba v pásme DVB-T a FM, ale v celom rozsahu frekvencií, ktoré fyzicky umožňuje toto zariadenie spracovať.

Celý USB tuner je veľmi jednoduché a relatívne lacné (približne 7 EUR) zariadenie obsahujúce iba minimum komponentov. Základnými a najdôležitejšími komponentami sú vstupný tuner R820T od firmy RafaelMicro a čipset RTL2832U od firmy Realtek obsahujúci 8-bitový analógovo-digitálny prevodník. Doska plošných spojov s jednotlivými komponentami je zobrazená na obrázku 2.3.

Zachytený signál z antény je čo najskôr prevedený do číselnej podoby. Na to slúži analógovo-digitálny prevodník, ktorý prevádza spojitý signál na číslo. Pri prevode spojitého signálu na číslo je nutné dodržať Shannonov vzorkovací teorém. Tento teorém hovorí o tom, že pri vzorkovaní spojitého signálu je nutné odobrať aspoň dve vzorky z periódy najvyššej frekvencie v harmonickom signále. To znamená, že vzorkovacia frekvencia musí byť dvakrát väčšia, než najvyššia harmonická zložka signálu. Aby bol Shannonov teorém zachovaný, je použitý filter s dolnou priepustou nazývaný tiež anti-alias filter. Výstupom analógovo-digitálneho prevodníka sú dva číslicové signály I/Q (in-phase a quadrature signály, sekcia 3.1), ktoré sú pomocou zbernice USB 2.0 posielané do počítača. Opačným smerom t.j. z počítača do tunera sú posielané užívateľské požiadavky na zachytávanie sig-

<sup>2</sup>Obrázok prevzatý z: [http://pandatron.cz/elektronika2/psdr\\_blok2.gif](http://pandatron.cz/elektronika2/psdr_blok2.gif)

<sup>3</sup>Obrázok prevzatý z: <http://osmocom.org/projects/sdr/wiki/rtl-sdr>



Obr. 2.3: Doska plošných spojov<sup>3</sup> (1. - tuner R820T, 2. - čipset RTL2832U)

nálu ako napr. na akej frekvencii a s akou šírkou pásma sa má zachytávať alebo povolenie hardvérového AGC<sup>4</sup> a pod.

Aby bola zabezpečená komunikácia medzi tunerom a počítačom, je nevyhnutné použiť ovládač (sekcia 5.3), ktorý vytvorí prepojenie medzi týmito dvoma komponentami. Ovládač musí byť schopný komunikovať cez zbernicu USB 2.0, po ktorej je zasielaný zachytený signál z tunera do počítača. Opačným smerom musí ovládač umožňovať nastavenie parametrov tunera, a to v čo najväčšom možnom rozsahu funkcií, ktoré poskytuje daný hardvér.

Zariadenie použité v tejto práci má nasledujúce parametre<sup>5</sup>:

- **Rozlíšenie A/D prevodníka** - 8 bitov
- **Maximálna rýchlosť A/D prevodníka** - 3,2 Msps<sup>6</sup>
- **Dynamický rozsah** - 48 dB
- **Šírka pásma** - približne 2,8 MHz
- **Frekvenčný rozsah** - približne 25 až 1750 MHz

Na zachytávanie a analýzu rádiových signálov pomocou technológie SDR (sekcia 2.1) je možné použiť aj iné zariadenia, priamo postavené na tento účel, avšak riešenie s využitím spomínaného USB tunera je lacnejšie. Na druhej strane je ale potrebné použiť spomínaný ovládač, ktorý bude komunikovať medzi tunerom a počítačom cez USB zbernicu.

## 2.3 Existujúce riešenia

Na trhu sa dá nájsť množstvo produktov s podobným zameraním. Jeden z príkladov je systém AKRS (softvér) spolu s IZ225 (hardvér)<sup>7</sup> od firmy URC, ktorý sa používa prevažne na armádne účely. Tento systém má oproti USB DVB-T tuneru neporovnateľne vyšší výkon (rýchlosť spracovania, presnosť, podrobnosť, atď.), ale aj cenu, pretože tento produkt (AKRS a IZ225) stojí približne jeden milión českých korún.

<sup>4</sup>Automatic Gain Control - Automatické vyrovnávanie citlivosti

<sup>5</sup>Zdroj údajov: <http://sdr.ipip.cz/rtl-sdr>

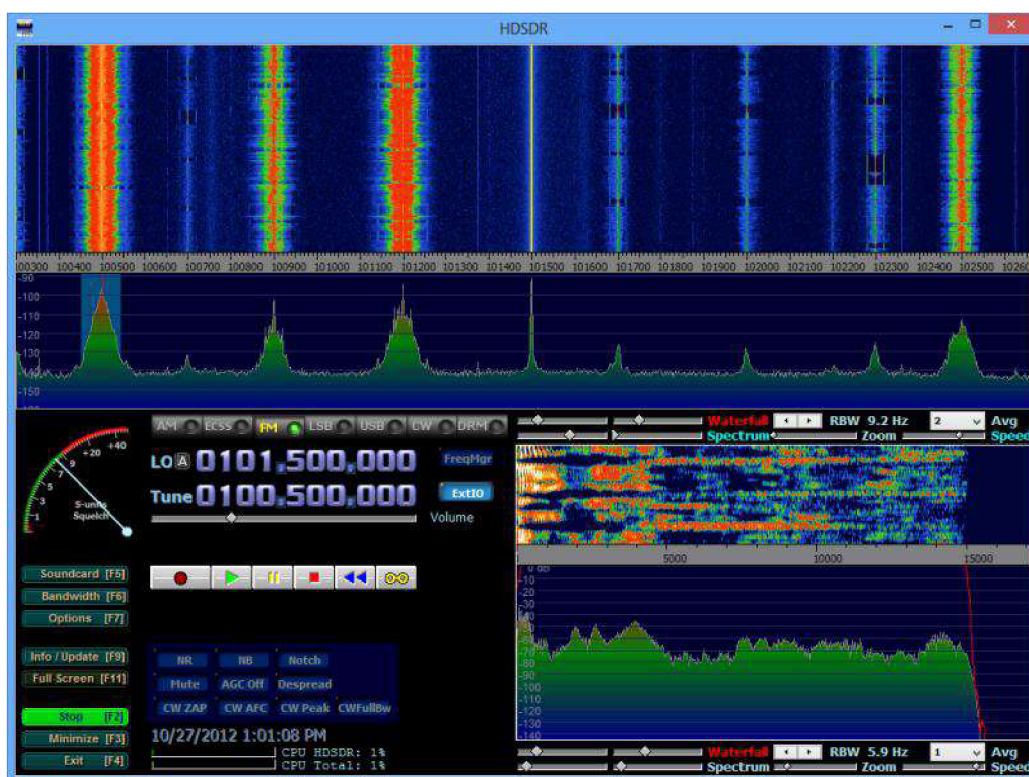
<sup>6</sup>Mega samples per second - počet miliónov vzoriek za sekundu

<sup>7</sup>Viac informácií na stránke: <http://www.urc-systems.cz/produkty/commses/>

Existujú aj nekomerčné riešenia, ktoré sú voľne dostupné a určené na rádioamatérske resp. študijné účely podobne ako aj táto práca. Väčšina riešení je navrhnutá práve pre čipset RTL2832U (sekcia 2.2).

### 2.3.1 HSDR

HSDR<sup>8</sup> (High Definition Software Defined Radio) je jedným z najznámejších voľných softvérov pre SDR a je určený pre platformu Windows. Tento softvér je založený na starom WinRAD<sup>9</sup> programe a navyše obsahuje ďalšie rozšírenia. Okrem základných funkcií ako je vykresľovanie grafu spektra a spektrogramu (horná polovica obrázku 2.4) na základe použitia algoritmu rýchlej Fourierovej transformácie (sekcia 3.2), obsahuje tento program aj analýzu demodulovaného signálu (obrázok 2.4 vpravo dole). Užívateľ má teda možnosť vidieť spektrum zachytávaného, ale aj demodulovaného signálu. Samozrejmosťou je audio výstup demodulovaného signálu a tiež možnosť tento signál nahráť a uložiť.



Obr. 2.4: Ukážka aplikácie HSDR

Pri prvom kontakte s touto aplikáciou môže jej ovládanie pôsobiť trochu zložité. Napríklad na počúvanie rozhlasového vysielania je potrebné zvoliť frekvenciu rádiovej stanice a vhodne nastaviť vzorkovaciu frekvenciu a šírku pásma. V opačnom prípade bude počuteľný iba šum. Napriek tomu je podľa mňa ovládanie tohto programu po chvíli používania veľmi jednoduché.

Táto aplikácia má aj zopár nedokonalostí. Jedným z príkladov je zadávanie frekvencie. Ak sa užívateľovi podarí zadať frekvenciu mimo rozsahu, ktorý podporuje hardvér, aplikácia

<sup>8</sup>Dostupný na: <http://www.hdsdr.de/>

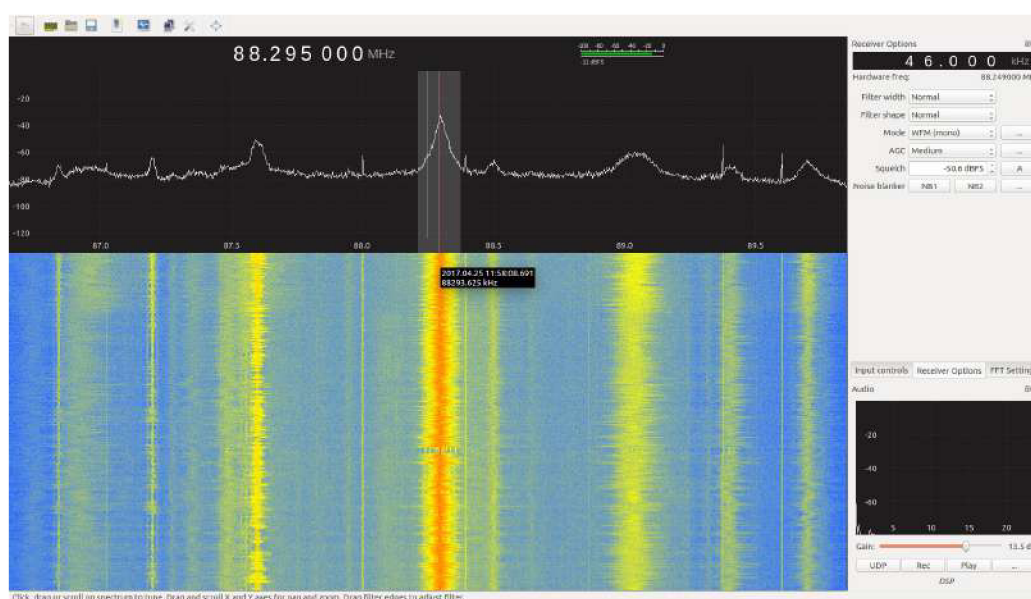
<sup>9</sup>Dostupný na: <http://www.winrad.org/>

sa zastaví s chybovým hlásením a spustí sa odznova. Ďalším problémom je veľká náročnosť na procesor, hlavne pri zapnutom automatickom doladovaní frekvencie (AFC<sup>10</sup>).

Napriek drobným chybám a možno pre niektorých užívateľov zložitému ovládaniu, je tento program pre softvérovo definované rádio celkom kvalitne spracovaný.

### 2.3.2 Gqrx SDR

Druhým príkladom programu pre softvérovo definované rádio je Gqrx SDR<sup>11</sup>. Táto aplikácia je určená pre platformy Linux a Mac a je postavená na GNU Radio a Qt frameworku. Rovnako ako aplikácia HDSDR (sekcia 2.3.1) je aj táto aplikácia bezplatná. Okrem funkcií, ktoré sú bežné pre všetky SDR aplikácie, obsahuje tento program navyše množstvo rôznych filtrov a korekcií. Dokonca podporuje vzdialené ovládanie tejto aplikácie pomocou sieťového protokolu TCP<sup>12</sup> a tiež spätné streamovanie audia pomocou protokolu UDP<sup>13</sup>. Na obrázku 2.5 je zobrazený natívny vzhľad aplikácie so zapnutým zachytávaním rádiového prenosu.



Obr. 2.5: Ukážka aplikácie Gqrx SDR

Ovládanie tohto programu je zložitejšie resp. menej intuitívne ako je to pri programe HDSDR. Niektoré dôležité ovládacie prvky ako napr. nastavenie šírky pásma alebo vzorkovacej frekvencie by mohli byť na základnom paneli, čo by určite uľahčilo prácu s aplikáciou. Asi najväčším problémom je nastavovanie vzoriek rýchlej Fourierovej transformácie, pretože pri zvolení veľkých hodnôt sa program zastaví na niekoľko sekúnd a následne pracuje veľmi spomalene. Taktiež chýba možnosť obnovenia pôvodných nastavení bez nutnosti reštartovania aplikácie.

V konečnom dôsledku je prevedenie tejto aplikácie veľmi kvalitné, hlavne bohaté možnosti nastavenia parametrov (filtre, korekcie, grafy, atď.). Program je vhodný pre pokročilejších rádioamatérov alebo pre užívateľov operačných systémov Mac a Linux.

<sup>10</sup>AFC - automatic frequency control

<sup>11</sup>Dostupný na: <http://gqrx.dk/>

<sup>12</sup>TCP - Transmission Control Protocol

<sup>13</sup>UDP - User Datagram Protocol

## Kapitola 3

# Teoretické základy SDR

### 3.1 I/Q signál

Charakteristickým črtom moderných digitálnych prijímačov je I/Q (in-phase/quadrature) signál. Výstupom použitého USB tunera je taktiež tento typ signálu. Princípom tohto signálu je, že harmonický signál je popísaný rotujúcim vektorom, pričom rotácia prebieha v komplexnej rovine a čase. Signál I zodpovedá kosínusovej resp. synfáznej zložke (in-phase) a signál Q zodpovedá sínusovej resp. kvadrátúrnej zložke (quadrature). Táto dvojkanálová koncepcia sa môže zdať zbytočne zložitá, avšak jej hlavnou výhodou je možnosť rozlíšiť zložky demodulovaného signálu patriace dolnému postrannému pásmu od horného postranného pásma.

Všeobecný modulovaný signál sa dá zapísať nasledovne:

$$s(t) = A \cos(\omega t + \varphi) \quad (3.1)$$

Kde  $A$  je amplitúda harmonického signálu,  $\omega$  je nosná uhlová frekvencia a  $\varphi$  fázový uhol.

Pre trigonometrický rozklad výrazu 3.1 je možné použiť nasledujúcu rovnicu:

$$\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \quad (3.2)$$

Po aplikovaní rovnice 3.2 na výraz 3.1 vznikne vzťah:

$$A \cos(\omega t + \varphi) = A \cos(\omega t) \cos \varphi - A \sin(\omega t) \sin \varphi \quad (3.3)$$

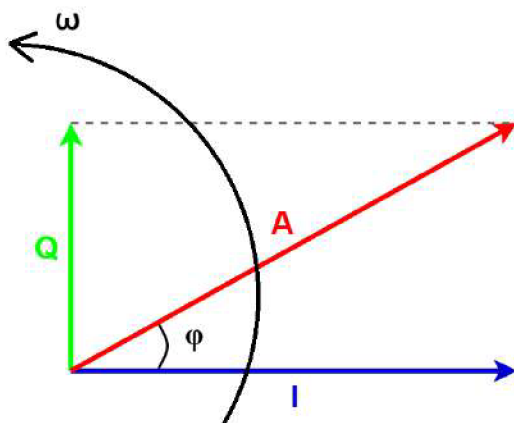
Ak sa zložke I (in-phase) priradí výraz  $A * \cos \varphi$  a zložke Q (quadrature) výraz  $A * \sin \varphi$ , dostane výraz 3.3 zrozumiteľnejšiu formu:

$$A \cos(\omega t + \varphi) = I \cos(\omega t) - Q \sin(\omega t) \quad (3.4)$$

Výsledný vzťah 3.4 je graficky znázornený na obrázku 3.1 a vyplýva z neho, že zložky I a Q:

- sú na seba kolmé a dá sa nimi popísať vzorec pre všeobecnú moduláciu
- spoločne rotujú v komplexnej rovine rýchlosťou  $\omega$
- nesú informáciu o amplitúdovej alebo fázovej modulácii

Informácie použité v tejto kapitole boli čerpané z literatúry [5] a [10]



Obr. 3.1: Princíp I/Q signálu znázornený pomocou vektorov

## 3.2 Rýchla Fourierova transformácia

Základným algoritmom pri analýze signálov je rýchla Fourierova transformácia alebo skrátene FFT. Je to veľmi efektívny algoritmus na výpočet diskkrétnej Fourierovej transformácie (DFT), ktorá je definovaná nasledovne:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n}, \quad k = 0, \dots, N-1 \quad (3.5)$$

Vzťah 3.6 je možné zapísať aj tzv. Eulerovým zápisom nasledovne:

$$X[k] = \sum_{n=0}^{N-1} x[n] \left[ \cos\left(2\pi \frac{k}{N} n\right) - j \sin\left(2\pi \frac{k}{N} n\right) \right], \quad k = 0, \dots, N-1 \quad (3.6)$$

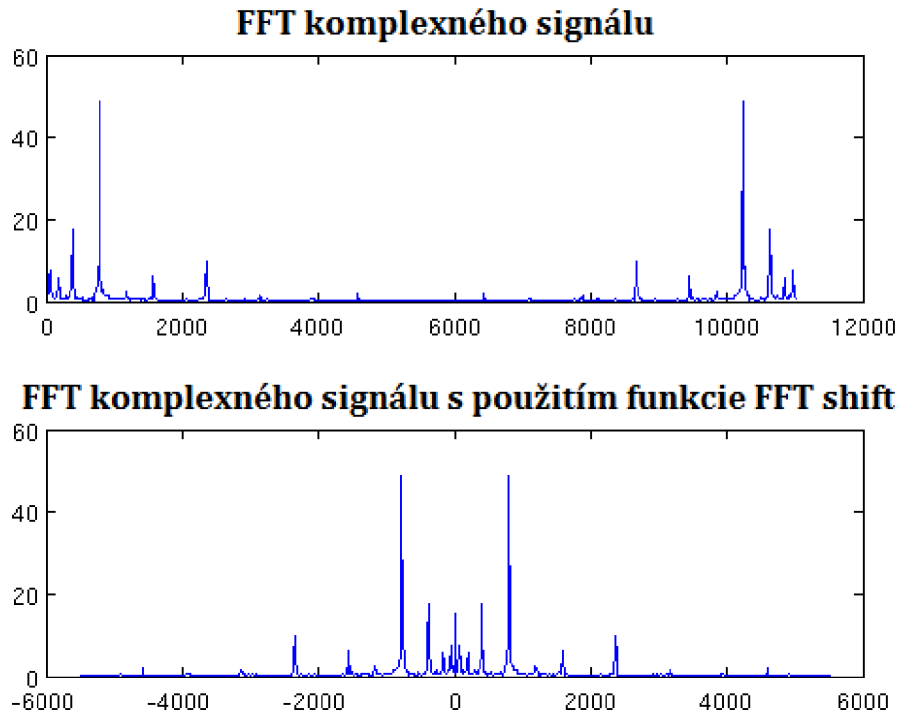
Kde  $X[k]$  je  $k$ -ty výstupný komponent DFT,  $k$  je index výstupu DFT,  $x[n]$  je vzorka vstupného signálu,  $n$  je poradie vzorky vstupného signálu a  $N$  je počet frekvenčných bodov vo výstupe DFT (hodnota  $N$  sa tiež nazýva FFT bins).

Fourierova transformácia slúži na prevod signálu z časovej oblasti do frekvenčnej. To znamená, že zistíme z akých frekvencií harmonických signálov sa signál skladá a aké dosahujú amplitúdy. Pomocou DFT teda dokážeme určiť spektrum zachytávaného signálu. Vypočítané spektrum je následne vhodné zobrazit pomocou spektrogramu.

V prípade, že na vstupe DFT je reálny signál, vo výstupe nastáva tzv. symetria. To znamená, že každá výstupná vzorka  $k$  je symetrická s  $N-k-1$  vzorkou resp. ich amplitúdy sú rovnaké. V prípade, že chceme zobrazit spektrum reálneho signálu, je potrebné obmedziť zobrazované vzorky na polovičný rozsah t.j. 0 až  $\frac{N}{2}$ . Ak je na vstupe DFT komplexný signál, táto symetria platiť nemusí, pretože spektrum komplexného signálu okolo nulovej frekvencie je vo všeobecnosti nesymetrické. Keďže výstupom USB tunera je I/Q t.j. komplexný signál, pri zobrazovaní spektra je potrebné použiť všetky vzorky DFT. Navyše na správne zobrazenie spektra pomocou grafu je nutné použiť funkciu na posun (angl. FFT shift). Táto funkcia slúži na posun nulových frekvencií do stredu spektra. Princíp FFT shiftu je zobrazený na obrázku 3.2

Podrobnejšie informácie o Fourierovej transformácii v súvislosti s analýzou rádiových signálov je možné nájsť v literatúre [4] a [8].

<sup>1</sup>Obrázok prebratý z: <https://i.stack.imgur.com/jmYGR.png>

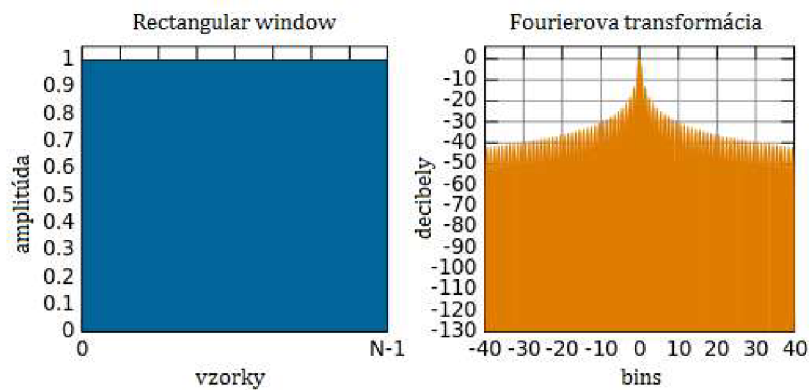


Obr. 3.2: Výstup FFT pred a po aplikovaní funkcie FFT shift<sup>1</sup>

### 3.3 Oknová funkcia

Pri spektrálnej analýze signálu sa často stáva, že v spektre sa okrem základnej frekvencie zobrazia aj nežiadúce frekvencie, ktoré nie sú súčasťou skúmaného signálu. Na odstránenie tohto problému je často využívaná oknová funkcia (angl. window function). Je to matematická funkcia, ktorá nadobúda nulové hodnoty mimo určitý interval. Ak sa vstupný signál vynásobí oknovou funkciou, signál bude tiež nulový tam, kde aj oknová funkcia. Vybrané oknové funkcie:

- **Obdĺžniková** - angl. Rectangular je najjednoduchšia oknová funkcia, ktorá nijak nemení vstupný signál

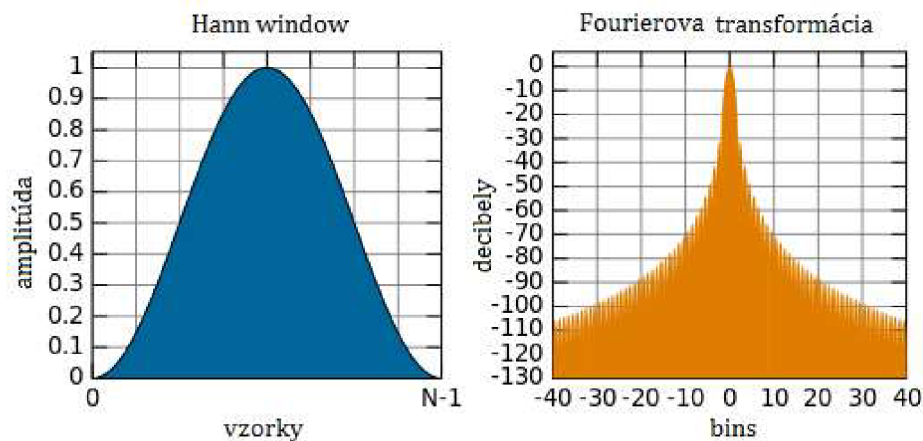


Obr. 3.3: Obdĺžniková funkcia a jej spektrum



- **Hann window** - univerzálna funkcia

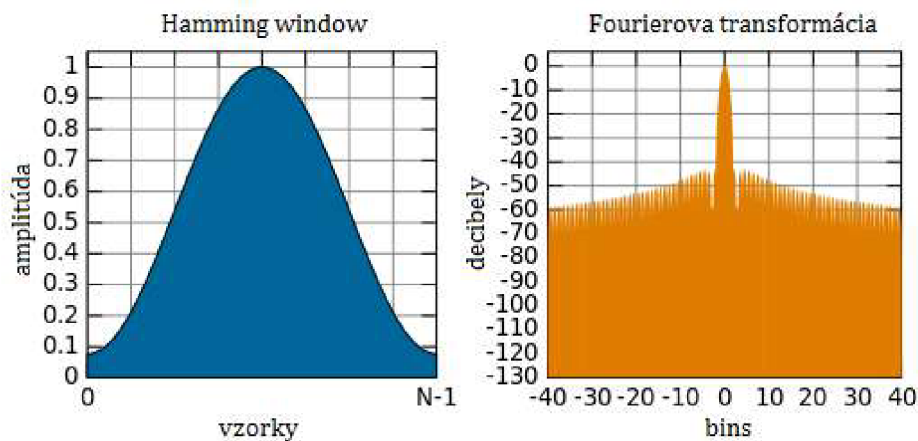
$$w(n) = 0.5 * \left( 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right) \quad (3.7)$$



Obr. 3.4: Funkcia Hann window a jej spektrum

- **Hamming window** - funkcia vhodná pre frekvenčne podobné signály

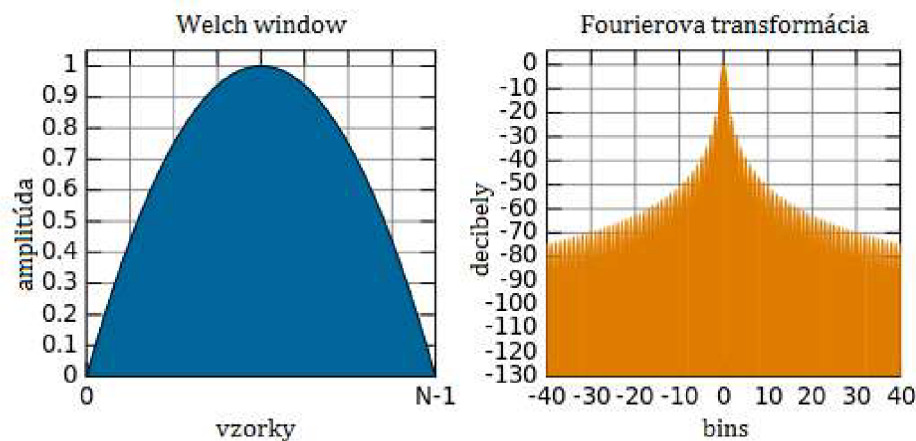
$$w(n) = \alpha - \beta \cos \left( \frac{2\pi n}{N-1} \right) \quad (3.8)$$



Obr. 3.5: Funkcia Hamming window a jej spektrum

- **Welch window** - jednoduchá parabolická funkcia

$$w(n) = 1 - \left( \frac{n - \frac{N-1}{2}}{\frac{N-1}{2}} \right)^2 \quad (3.9)$$



Obr. 3.6: Funkcia Welch window a jej spektrum

Kde  $w(n)$  je výstup oknovej funkcie,  $n$  je index výstupu DFT,  $N$  je počet frekvenčných bodov DFT a  $\alpha$  a  $\beta$  sú konštanty.

Obrázky použité v tejto sekcii boli prebraté<sup>2</sup>. Viac informácií o oknových funkciách je možné nájsť v literatúre [2] a [9].

### 3.4 Rádiové signály

Rádiový signál je súčasťou elektromagnetického spektra a využíva sa prevažne na komunikáciu. Frekvencia rádiových vln sa väčšinou pohybuje v rozsahu od 3 kHz do 300 GHz. Rádiové vlny sa pohybujú rýchlosťou svetla a na ich vysielanie sa používa rádiový vysielateľ. Na prijímanie týchto vln sa zas používa rádiový prijímač. Jedným z príkladov takéhoto prijímača je aj použitý USB tuner (sekcia 2.2). Signály používané v rádiových komunikáciách sú klasifikované nasledovne:

- **3 - 30 kHz** - veľmi dlhé vlny (VLF)
- **30 - 300 kHz** - dlhé vlny (LF)
- **300 - 3000 kHz** - stredné vlny (MF)
- **3 - 30 MHz** - krátke vlny (HF)
- **30 - 300 MHz** - veľmi krátke vlny (VHF)
- **300 - 3000 MHz** - ultra krátke vlny (UHF)
- **3 - 30 GHz** - centimetrové vlny (SHF)
- **30 - 300 GHz** - milimetrové vlny (EHF)

V našom prípade dokáže použitý USB tuner pracovať s frekvenciami od 25 do 1750 MHz. Jedná sa teda o pásma veľmi krátkych a ultra krátkych vln. Pásmo veľmi krátkych vln sa používa prevažne na rozhlasové a televízne vysielanie. Tento typ rádiových vln sa šíri tzv. priamymi vlnami, ktoré sa čiastočne ohýbajú okolo zemského povrchu.

<sup>2</sup>Zdroj obrázkov: [https://en.wikipedia.org/wiki/Window\\_function](https://en.wikipedia.org/wiki/Window_function)

Pásmo ultra krátkych vln je využívané predovšetkým na prenos mobilného GSM signálu a ďalej sa tiež využíva na letecké a námorné systémy. Šírenie týchto vln je ovplyvňované odrazmi od prekážok, čo obzvlášť platí v mestských zástavbách.

Viac informácií o jednotlivých pásmach signálových vln je možné nájsť v literatúre [6].

### 3.4.1 Antény

Anténa je zariadenie k vysielaniu alebo príjmu rádiových signálov. Je to časť vysokofrekvenčného vedenia upravená tak, aby účinne vyžarovala energiu do priestoru. Antény sa delia na vysielacie a prijímacie (každá anténa môže ako vysielateľ, tak aj prijímať). Vysielacia anténa slúži k premene elektrickej energie na energiu elektromagnetických vln. Prijímacia anténa má presne opačný princíp. Tu sú uvedené základné typy antén:

- **Drôtová anténa** - najjednoduchšia anténa, ktorá dokáže vysielateľ v horizontálnej aj vertikálnej pozícii
- **Rámová anténa** - smerová anténa, ladený paralelný obvod s veľkou cievkou
- **Feritová anténa** - tvorená feritovou tyčinkou, na ktorej je navinutá cievka
- **Parabolická anténa** - využíva parabolický povrch na vysielanie resp. príjem signálu
- **Dipólová anténa** - využíva priamy vodič v určitom mieste prerušený
- **Slučková anténa** - využíva vodič tvarovaný do slučky

## 3.5 Modulácia a demodulácia signálu

Pri analyzovaní rádiovkej komunikácie je dôležité, aby bolo možné, si analyzovaný signál vypočítať. Je preto potrebné tento signál demodulovať. Proces demodulácie signálu je opačný ako proces modulácie. Pri modulácii meníme charakter nosného signálu pomocou modulačného signálu (znázornené na obrázkoch<sup>3</sup> 3.7 a 3.8).

Poznáme dve základné kategórie modulácií signálu, ktoré sú **analogová** a **digitálna**.

### 3.5.1 Analogové modulácie

Analogová modulácia je typ modulácie, kde je použitý spojité signál, ktorý je následne modulovaný nosnou frekvenciou.

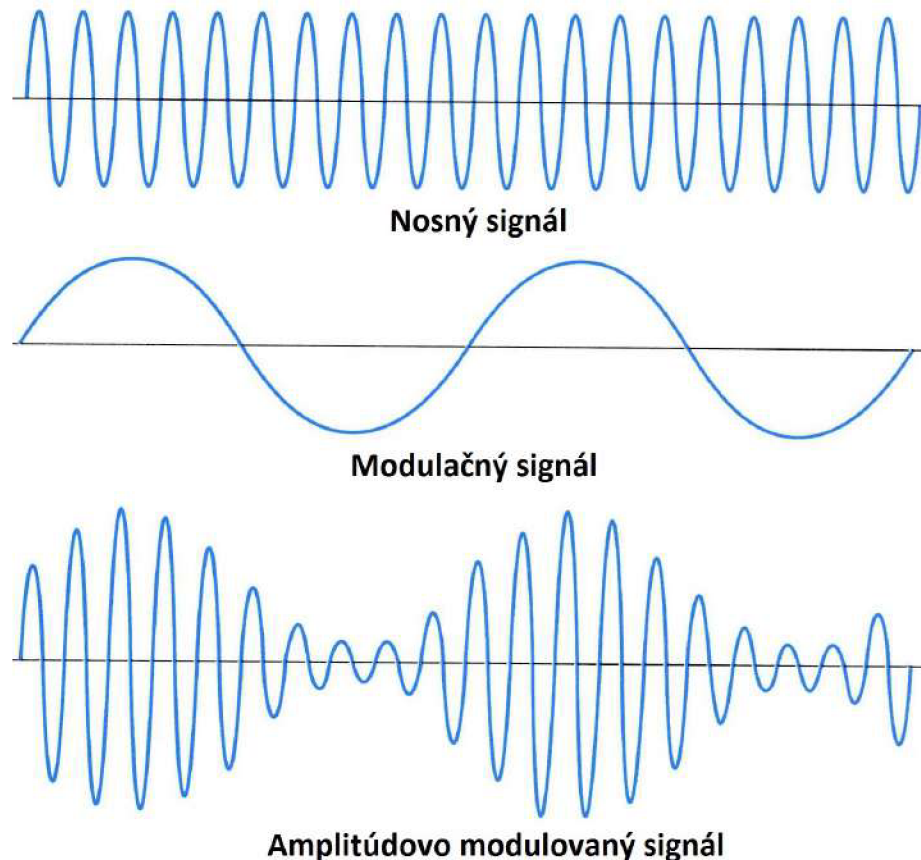
- **Amplitúdová modulácia (AM)** - hodnota modulačného signálu ovplyvňuje výšku amplitúdy modulovaného signálu
- **Frekvenčná modulácia (FM)** - hodnota modulačného signálu ovplyvňuje frekvenciu modulovaného signálu
- **Fázová modulácia (PM)** - hodnota modulačného signálu fázovo ovplyvňuje výšku amplitúdy modulovaného signálu
- **Single side band (SSB, LSB/USB)** - obsahuje iba jedno postranné pásmo

---

<sup>3</sup>Obrázky prevzaté z: <http://ironbark.xtelco.com.au/subjects/DC/lectures/7/>

- **Kvadratúrne amplitúdová modulácia (QAM)** - je ako analógová, tak aj digitálna modulačná schéma, ktorá kóduje buď dva analógové signály, alebo dva digitálne bity; poznáme aj rozšírené typy QAM modulácie ako sú napr. QAM16, QAM64, QAM256, ktoré dokážu za jednu jednotku času preniesť až 256 bitov

### Amplitúdová modulácia



Obr. 3.7: Princíp amplitúdovej modulácie signálu

Amplitúdová modulácia je jednou z najjednoduchších a najstarších spojitých modulácií používaných v rádiovkej komunikácii. Vo frekvenčnom spektre tejto modulácie je nosná vlna a dve postranné pásma (angl. dual side band). Často sa však niektorá z týchto zložiek odstraňuje a tak vzniká modulácia s jedným postranným pásmom (angl. single side band).

Pri amplitúdovej modulácii sa mení amplitúda nosnej vlny podľa hodnoty modulačného signálu (obrázok 3.7). Frekvencia aj fáza modulovanej vlny ostáva nemenná.

Nasledujúce vzťahy sú rovnako platné ako pre amplitúdovú, tak aj pre frekvenčnú moduláciu a popisujú nosný a modulačný signál:

$$n(t) = A_n \sin(2\pi ft) \quad (3.10)$$

Kde  $n$  je nosný signál s maximálnou amplitúdou  $A_n$  a  $f$  je kmitočet.

$$m(t) = A_m \sin(2\pi Ft) \quad (3.11)$$

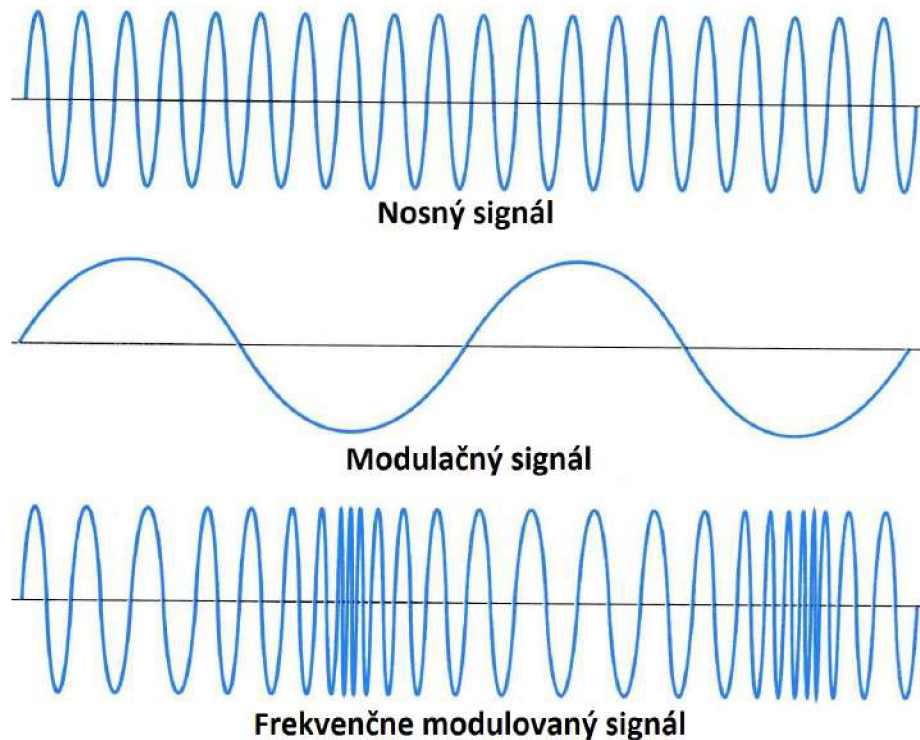
Kde  $m$  je modulačný signál s maximálnou amplitúdou  $A_m$  a  $F$  je kmitočet.

Rovnica pre amplitúdovo modulovanú nosnú vlnu je nasledovná:

$$a(t) = A_n \sin(2\pi ft) + \frac{A_m}{2} \cos(2\pi(f - F)t) - \frac{A_m}{2} \cos(2\pi(f + F)t) \quad (3.12)$$

Kde  $a$  modulovaná nosná vlna,  $A_n$  je amplitúda nosného signálu 3.10,  $A_m$  je amplitúda modulačného signálu 3.11,  $f$  je frekvencia nosného signálu a  $F$  je frekvencia modulačného signálu. Princíp amplitúdovej modulácie je zobrazený na obrázku 3.7.

### Frekvenčná modulácia



Obr. 3.8: Princíp frekvenčnej modulácie signálu

Frekvenčná modulácia sa prevažne využíva v pásme veľmi krátkych vln pre prenos zvuku napr. rozhlasového vysielania alebo vysieláčiek. Pri tejto modulácii sa mení frekvencia nosnej vlny, pričom amplitúda a fáza ostávajú konštantné (obrázok 3.8) a platí nasledujúci vzťah:

$$b(t) = A_n \cos[2\pi ft + \beta \sin(2\pi Ft)] \quad (3.13)$$

Kde  $b$  je nosný modulovaný signál,  $A_n$  je amplitúda nosného signálu 3.10 a  $\beta$  je index (činiteľ) frekvenčnej modulácie. Princíp frekvenčnej modulácie je zobrazený na obrázku 3.8.

Uvedené vzťahy pre amplitúdovú a frekvenčnú moduláciu boli čerpané z [11] a [1]. Viac informácií o modulácii a demodulácii analógových signálov je možné nájsť v literatúre [3], [4] a [7].

### 3.5.2 Digitálne modulácie

Digitálna modulácia je typ modulácie, kde je použitý diskretný signál. Zmeny nosného signálu sú prevádzané na základe konečných stavov modulačného slovníka.

- **Amplitúdové kľúčovanie (ASK)** - je to forma amplitúdovej modulácie, ktorá reprezentuje digitálne dáta ako variácie amplitúdy v nosnom signále
- **Frekvenčné kľúčovanie (FSK)** - je to forma frekvenčnej modulácie, v ktorej sú informácie prenášané pomocou diskretných zmien frekvencie nosného signálu
- **Fázové kľúčovanie (PSK)** - je to forma fázovej modulácie,
- **Kvadratúrne amplitúdová modulácia (QAM)** - vysvetlené v predchádzajúcej podkapitole [3.5.1](#)

## Kapitola 4

# Návrh riešenia

Návrh riešenia tejto práce sa skladá z troch základných častí. V prvom rade je potrebné zachytiť analyzovaný signál pomocou tunera a preniesť ho do počítača. Na zabezpečenie komunikácie slúži ovládač, ktorý je popísaný v sekcii 4.2. Keď už sa signál nachádza v počítači, je možné nad ním aplikovať rôzne algoritmy na spracovanie signálu. Práci so signálmi sa venuje sekcia 4.3. Spracovaný signál je následne vhodné užívateľovi prezentovať pomocou užívateľského rozhrania, o ktorom je písané v sekcii 4.4. Taktiež je vhodné užívateľovi umožniť zmenu parametrov zachytávaného signálu.

### 4.1 Bloková schéma aplikácie

Tvorba aplikácie sa riadila podľa blokovej schémy uvedenej na obrázku 4.1. Pri prvotnom spustení aplikácie je potrebné inicializovať USB tuner. V prípade, že nie je pripojený resp. nie je možné s ním komunikovať, užívateľovi sa zobrazí chybové hlásenie. Ak inicializácia prebehne úspešne, nasleduje inicializácia počiatočných nastavení, potrebných pre zachytávanie signálu. Následne je možné načítavať zachytené dáta do vyrovnávacej pamäte (angl. buffer). Po naplnení buffera sa aplikuje zvolená oknová funkcia a vypočíta sa FFT s aplikovaním algoritmu FFT shift. Výstup FFT je normalizovaný a prevedený na reálne číslo, ktoré je možné vložiť do grafu spektra a spektrogramu.

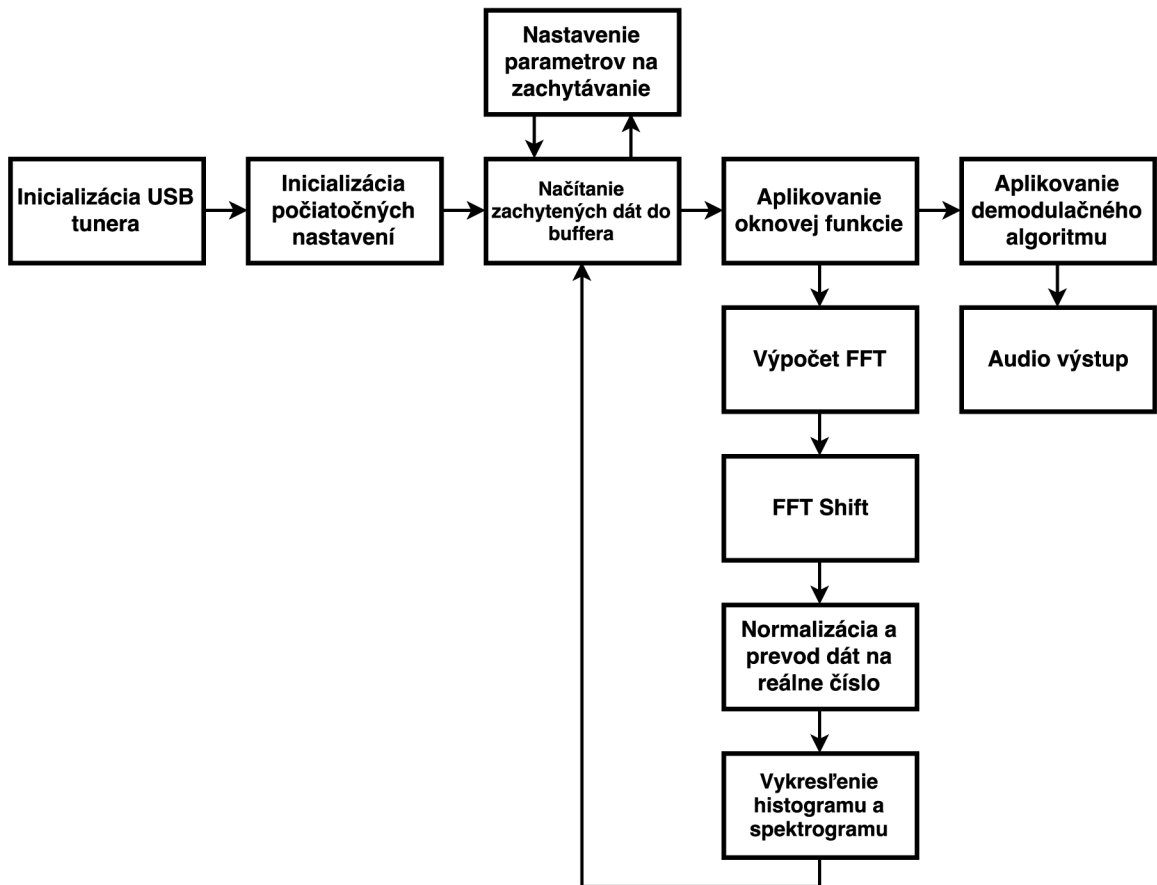
Po aplikovaní oknovej funkcie sa môže taktiež použiť demodulačný algoritmus, ktorého výstup je privedený na audio sústavu počítača.

Podrobnejšie informácie o implementácii jednotlivých blokov sú popísané v kapitole 6.

### 4.2 Práca s USB tunerom

Aby bolo možné pracovať s USB tunerom, je nevyhnutné implementovať alebo použiť riadiaci program, nazývaný tiež ovládač alebo driver. Tento ovládač zabezpečí komunikáciu medzi tunerom a počítačom resp. aplikáciou. Jeho hlavnou úlohou je zabezpečiť komunikáciu medzi zariadeniami, a to v oboch smeroch. V smere z aplikácie do tunera sa prenášajú informácie o aktuálnych užívateľských nastaveniach ako napríklad frekvencia, vzorkovacia frekvencia, šírka pásma a pod. Opačným smerom sa posiela zachytený signál vo forme I/Q signálu (sekcia 3.1). Dáta v oboch smeroch musia byť posielané po zbernici USB 2.0.

Implementácia ovládača pre USB tuner je veľmi zložitá a časovo náročná, pretože si vyžaduje pozorné preštudovanie spôsobu komunikácie. Navyše je dostupné iba veľmi obmedzené množstvo dokumentácie k použitému DVB-T tuneru. Najrozumnejším spôsobom



Obr. 4.1: Bloková schéma návrhu aplikácie

na zabezpečenie komunikácie s tunerom je preto použitie už existujúceho ovládača. Na internete existujú riešenia, ktoré sú vhodné pre použitý typ USB tunera a sú dokonca voľne šíriteľné. Takéto voľne dostupné riešenia ale majú aj svoje nevýhody. Hlavnou nevýhodou je, že môžu obsahovať rôzne neošetrené chyby alebo použité metódy a algoritmy sú neefektívne.

Napriek uvedeným nevýhodám je použitie už existujúceho ovládača tou najlepšou voľbou vzhľadom na účel tejto práce, ktorým je demonštrovať základnú funkčnosť softvérového definovaného rádia (sekcia 2.1).

### 4.3 Spracovanie signálu

Zachytený signál, ktorý je privedený pomocou zbernice USB do počítača je vo forme I/Q signálu (sekcia 3.1) a je potrebné ho ďalej spracovať. Na analýzu signálu v reálnom čase je nutné spracovať veľké množstvo vstupných dát za čo najmenšiu časovú dobu. Na spracovanie signálov sa používajú rôzne algoritmy, ktoré sú popísané v kapitole 3. Hlavným algoritmom pri analýze je rýchla Fourierova transformácia (sekcia 3.2), ktorá určí spektrum daného signálu. Takto určené spektrum je veľmi vhodné interpretovať užívateľovi vo forme grafov. Na vykresľovanie grafov sa musí výstup Fourierovej transformácie ďalej spracovať pomocou algoritmov používaných pri tvorbe grafov. Pokiaľ sa jedná o zvukový signál, napr. rádiové vysielanie, užívateľ ocení aj možnosť, tento signál si vypočítať. Na tento účel



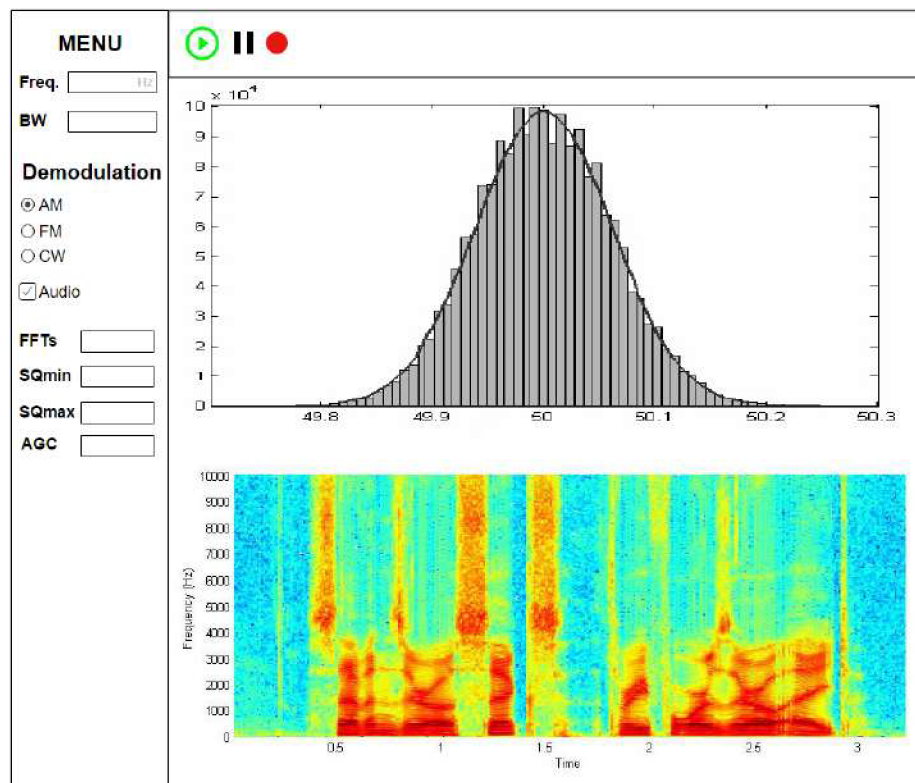
poslúžia demodulačné algoritmy. V tejto práci sa budeme konkrétne zaoberať amplitúdovou a frekvenčnou demoduláciou (sekcia 3.5.1)

## 4.4 Uživatelské rozhranie

Hlavnou myšlienkou tejto práce je vytvoriť čo najjednoduchšiu aplikáciu na analýzu signálov. To znamená, že pri návrhu užívateľského rozhrania je potrebné brať ohľad na cieľovú skupinu užívateľov, ktorými budú prevažne rádioamatéri. Pre túto skupinu užívateľov platí, že grafické rozhranie aplikácie má byť v prvom rade funkčné a prehľadné a na estetickú stránku nie je až tak potrebné brať ohľad. Aby bola táto práca vhodná aj pre začínajúcich rádioamatérov, musí byť užívateľské rozhranie jednoduché na používanie.

Aby užívateľské rozhranie splnilo svoj účel, musí sa skladať z dvoch hlavných častí. Prvá časť umožňuje užívateľovi meniť parametre analýzy signálu ako napríklad nastavenie nosnej frekvencie, zmena šírky pásma alebo spustenie, či zastavenie zachytávania. Druhá časť interpretuje užívateľovi analyzovaný signál vo forme grafov.

Na základe týchto požiadaviek bol vytvorený mockup (návrh), ktorý je zobrazený na obrázku 4.2.



Obr. 4.2: Mockup užívateľského rozhrania

V ľavej časti návrhu sa nachádza menu, v ktorom užívateľ môže nastavovať vstupné parametre analýzy. Význam jednotlivých ovládacích komponentov:

- **Freq.** - zadanie nosnej frekvencie
- **BW** - zadanie šírky pásma

- **Demodulation** - výber typu demodulácie signálu (amplitúdová, frekvenčná, morzeovka)
- **Audio** - povolenie audio výstupu do počítača
- **FFTs** - voľba počtu FFT bins (sekcia 3.2)
- **SQ min/max** - nastavenie squelch (slúži na odpojenie audio výstupu v prípade, že sila signálu nie je vyhovujúca zvolenému intervalu)
- **AGC** - slúži na automatické vyrovnávanie citlivosti

Tlačidlá v hornej časti návrhu slúžia na spustenie a pozastavenie analýzy a tretie tlačidlo s červeným kruhom slúži na nahrávanie vstupného I/Q signálu (sekcia 3.1).

Stredná plocha aplikácie slúži na vykresľovanie analyzovaných dát. Horný graf zobrazuje aktuálne spektrum zachyteného signálu a spodný graf (nazývaný spektrogram) zobrazuje zachytené spektrum signálu v čase.

## Kapitola 5

# Použité technológie

Pre tvorbu tejto práce je nutné zvoliť vhodné technológie, podľa ktorých sa bude ďalej odvíjať návrh a implementácia celej práce. Práve preto je kľúčové, si dobre preštudovať a premyslieť, aké technológie zvoliť.

### 5.1 Jazyk C++

Na implementáciu aplikácie bol zvolený jazyk C++. Tento objektovo orientovaný programovací jazyk je veľmi vhodný na prácu s knižnicami, ktoré sú nevyhnutne potrebné na riešenie tejto práce. Vzhľadom na to, že je to jeden z najpoužívanejších programovacích jazykov na svete, je veľmi jednoduché nájsť na internete príklady, informácie a odpovede pri riešení problémov. Ďalšou výhodou tohto programovacieho jazyka je, že je rýchlejší ako v súčasnosti tiež často používané interpretované jazyky ako napríklad Java alebo C#.

#### 5.1.1 Knižnica Simple-FFT

Knižnica Simple-FFT<sup>1</sup> implementuje výpočet rýchlej Fourierovej transformácie (sekcia 3.2). Táto knižnica je napísaná v programovacom jazyku C++ a jej hlavnými prednosťami sú:

- **Free & Open-source knižnica** - nie je potrebné platiť za túto knižnicu a riešiť problémy s licenciou
- **Knižnica zložená iba z hlavičkových súborov** - stačí vložiť súbory do projektu a používať (nie je potrebný build)
- **Dokáže pracovať aj s 2D poľami** - obzvlášť vhodné pri práci s I/Q signálmi
- **Malá veľkosť** - iba zopár hlavičkových súborov

Existuje aj zopár nevýhod pri tejto knižnici. Hlavnou nevýhodou je, že použitý algoritmus nie je najrýchlejší zo všetkých známych algoritmov. Taktiež nedokáže pracovať so vzorkami, ktoré nie sú mocninou dvoch.

Aj napriek uvedeným nevýhodám je táto knižnica veľmi vhodná pri riešení danej problematiky, pretože počet vzoriek Fourierovej transformácie môže byť prednastavený na mocniny dvoch a rýchlosť výpočtu je takisto dostatočujúca.

---

<sup>1</sup>Dostupná na: <https://github.com/divanov/Simple-FFT>

## 5.2 Qt framework

Na tvorbu grafického rozhrania bol použitý Qt framework. Medzi hlavné výhody tohto frameworku patrí, že je kompatibilný s viacerými platformami a na nekomerčné účely je bezplatný. Qt sa používa prevažne s programovacím jazykom C++, ale dá sa použiť aj s inými jazykmi. Jeho ďalšou výhodou je veľmi dobre a prehľadne spracovaná dokumentácia. Na programovanie s týmto frameworkom je možné použiť vývojové prostredie Qt Creator, ktoré podstatne zjednodušuje a spríjemňuje prácu. Vytvárané aplikácie používajú natívny vzhľad operačného systému a preto sa veľmi jednoducho prispôbia.

Keďže pri analýze signálov je takmer nevyhnutné pracovať s rôznymi stavmi a zmenami signálu, je veľmi dobré, že tento framework obsahuje množstvo rozšírení a technológií, ktoré umožňujú pracovať s grafikou. Na vykresľovanie spektrogramu a grafu spektra, ktoré sú v tejto práci kľúčovými, sú použité knižnice QCustomPlot (sekcia 5.2.1) a grafická knižnica OpenGL (sekcia 5.2.2).

### 5.2.1 Knižnica QCustomPlot

Táto knižnica slúži na vizualizáciu jednoduchších, dvojrozmerných grafov a tiež poskytuje dostatočný výkon na vykresľovanie dát v reálnom čase. Práve preto je veľmi vhodná na účely tejto práce a to konkrétne na vykresľovanie grafu spektra. Je určená pre Qt framework a má veľmi dobre spracovanú dokumentáciu. Hlavnými výhodami tejto knižnice sú:

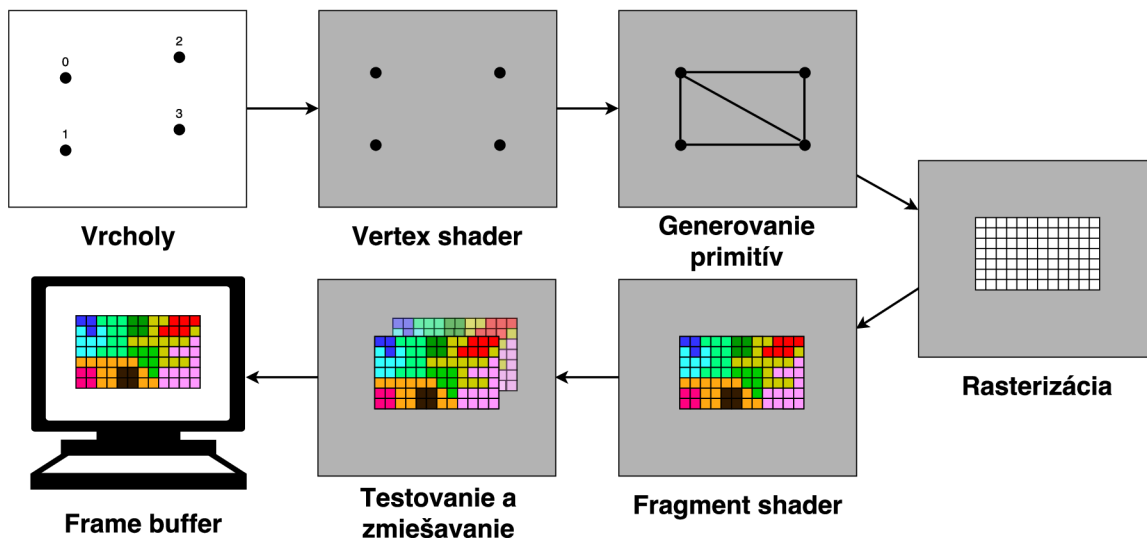
- Voľná dostupnosť
- Aktívny prístup užívateľov k riešeniu problémov (fóra a pod.)
- Veľmi jednoduchá a prehľadná na používanie
- Dobre zdokumentovaná

Uvedená knižnica má aj zopár nevýhod ako napríklad, že dokáže pracovať iba s karteziánskou sústavou t.j. nepodporuje rôzne výsekové grafy a taktiež nepodporuje multithreading.

Napriek uvedeným nevýhodám je táto knižnica vhodná na použitie v tejto práci, pretože na vykresľovanie grafu spektra je potrebné pracovať práve s karteziánskou sústavou. Navyše je veľmi jednoduché nastavovať hodnoty na osách X a Y v reálnom čase podľa potreby. Na druhej strane, na vykresľovanie zložitejších grafov, ako je spektrogram, už táto knižnica nie je dostačujúca práve kvôli absencii multithreadingu.

### 5.2.2 OpenGL

Súčasťou aplikácie je dvojrozmerný graf, ktorý je potrebný v reálnom čase obnovovať s veľkým množstvom dát. Na vykreslenie takéhoto grafu existujú objekty a renderovacie metódy s použitím knižnice Qt. Keďže tieto metódy pracujú na hlavnom vlákne v procesore, tak pri takomto veľkom množstve dát by spracovanie nebolo optimálne. Preto je nutné hľadať alternatívnu cestu ako vygenerovať tento graf, v ktorom sa obnovuje veľké množstvo dát v malých intervaloch. Riešením je využitie dedičnosti od Qt widget-u s podporou použitia OpenGL funkcií. Rozhodol som sa použiť verziu OpenGL 3.2 (Core Profile), ktorá si vyžaduje napísať vlastný GLSL (Graphic Library Shading Language) program. Tento program sa skladá z Vertex shader-u a Fragment shader-u, ktoré sú implementačne veľmi jednoduché. Najefektívnejším spôsobom ako dosiahnuť takýto pohyblivý graf, je využitie textúr a



Obr. 5.1: Schéma vykresľovania v OpenGL

vďaka dobrej optimalizácii grafickej karty, by sa mal výsledok niekoľkonásobne zrýchliť. Na obrázku 5.1 je znázornená postupnosť jednotlivých činností potrebných pre vykreslenie objektu v OpenGL.

### 5.3 Ovládač

Na zabezpečenie komunikácie medzi USB tunerom a počítačom bol zvolený už existujúci ovládač rtl-sdr<sup>2</sup>, ktorý je voľne šíriteľný a modifikovateľný podľa ustanovení Všeobecnej zverejňovacej licencie GNU. Tento ovládač je písaný v jazyku C a je určený pre platformu Linux. Poskytuje podporu pre viaceré zariadenia obsahujúce čipset RTL2832U, používané na softvérovo definované rádio. Medzi tieto zariadenia patrí aj tuner R820T (sekcia 2.2), ktorý je použitý v tejto práci.

Medzi hlavné pozitíva tohto ovládača patrí, že je voľne dostupný a modifikovateľný a tiež, že obsahuje všetky potrebné funkcie, ktoré sú dôležité pre komunikáciu s USB tunerom. Na druhej strane, neexistuje žiadna dokumentácia k tomuto ovládaču. Dokonca aj samotný zdrojový kód je veľmi slabo komentovaný. Táto skutočnosť sťažuje prácu pri vytváraní spojenia medzi USB tunerom a aplikáciou, pretože bude nutné podrobne študovať zdrojové kódy ovládača.

<sup>2</sup>Dostupný na: <https://github.com/pinkavaj/rtl-sdr/tree/master/src>

## Kapitola 6

# Implementácia a riešenie problému

Nasledujúca kapitola popisuje implementáciu aplikácie podľa návrhu uvedeného v Kapitole 4 s použitím technológií uvedených v Kapitole 5. Prvým krokom pri vytváraní aplikácie je vytvorenie blokovej schémy (sekcia 4.1) návrhu celého systému. Po premyslení návrhu aplikácie je možné implementovať jednotlivé bloky podľa teoretických základov uvedených v Kapitole 3.

### 6.1 Zabezpečenie komunikácie s ovládačom

Na zabezpečenie komunikácie aplikácie s ovládačom slúži trieda *DeviceDriver*. Táto trieda implementuje metódy, pomocou ktorých sú zasielané požiadavky na ovládač. Používané metódy:

- *start()* - aktivovanie zachytávania dát
- *stop()* - deaktivovanie zachytávania dát
- *setFrequency()* - nastavenie nosnej frekvencie
- *setBandwidth()* - nastavenie šírky pásma
- *setSamplingRate()* - nastavenie vzorkovacej frekvencie
- *setAgc()* - povolenie automatického vyrovnávania citlivosti
- *read()* - čítanie zachytených dát

Hlavným účelom uvedených metód je zavolať príslušné metódy implementované v ovládači rtl-sdr. To má za následok zjednodušenie práce pri implementácii samotnej aplikácie a zlepšenie prehľadnosti kódu t.j. všetky potrebné funkcie sú prístupné cez túto triedu. Trieda *devicedriver* navyše ošetruje prípady, keď USB tuner nie je pripojený alebo ho používa iná aplikácia. V prípade neošetrenia týchto prípadov, by mohlo dôjsť k pádu resp. neočakávanému správaniu aplikácie.

### 6.2 Implementácia spektrálnej analýzy

Zachytené dáta z USB tunera je vhodné uložiť do nejakej medzipamäte. Na to slúži metóda *read()* z triedy *DeviceDriver*. Nad uloženými dátami sa aplikuje zvolená oknová funkcia

(viz. sekcia 3.3). Aplikácia oknovej funkcie prebieha veľmi jednoducho. Cyklicky sa prechádzajú všetky prvky zachytených dát a prenasobia sa príslušným vzorcom pre zvolenú oknovú funkciu a zároveň sa dáta uložia do špeciálneho poľa komplexných čísel. Nasleduje výpočet rýchlej Fourierovej transformácie (viz. sekcia 3.2), ktorá pracuje so vstupným komplexným poľom a počtom prvkov (binov) Fourierovej transformácie. Po vypočítaní FFT je nutné vykonať posun (angl. shift), ktorý zabezpečí správnu reprezentáciu spektra (viz. obrázok 3.2). Na posun spektra je implementovaná funkcia *fftShift()*, ktorej úlohou je rozdeliť vstupné pole na polovicu a vzniknuté dve časti medzi sebou vymeniť. Po správnom posunutí spektra je dobré vykonať normalizáciu dát. Na to slúži vzorec pre odhad spektrálnej hustoty výkonu:

$$x[i] = A * \log \left( 4 * \frac{r^2[i] + j^2[i]}{4 * bins^2} \right) \quad (6.1)$$

Kde  $\mathbf{x}[i]$  je aktuálne spracovávaný prvok spektra,  $\mathbf{i}$  je index aktuálneho prvku,  $\mathbf{A}$  je amplitúdový posun,  $\mathbf{r}[i]$  je reálna zložka aktuálneho prvku,  $\mathbf{j}[i]$  je imaginárna zložka aktuálneho prvku a  $\mathbf{bins}$  je celkový počet prvkov FFT. Zmenou hodnoty  $\mathbf{A}$  je možné meniť intenzitu amplitúdy podľa potreby užívateľa.

Po uvedenej normalizácii môže nastať situácia, keď niektoré hodnoty budú mimo rozsahu stupnice a môže dôjsť k pretečeniu dát. Z tohto dôvodu sa po normalizácii musí overiť, či výsledná hodnota nepresahuje maximálnu alebo minimálnu možnú hodnotu. V prípade, že hodnota je mimo rozsah, do výsledku sa zapíše buď maximálna, alebo minimálna možná hodnota.

V tejto chvíli sú dáta pripravené na vykreslenie pomocou grafu a spektrogramu, čomu sa venuje nasledujúca podkapitola.

### 6.3 Vykresľovanie grafu a spektrogramu

Po aplikovaní všetkých potrebných algoritmov spektrálnej analýzy sa môže analyzovaný signál vykresliť pomocou grafov. Na vykresľovanie grafov boli použité knižnice QCustomPlot a OpenGL uvedené v sekcii 5.2.1 a 5.2.2.

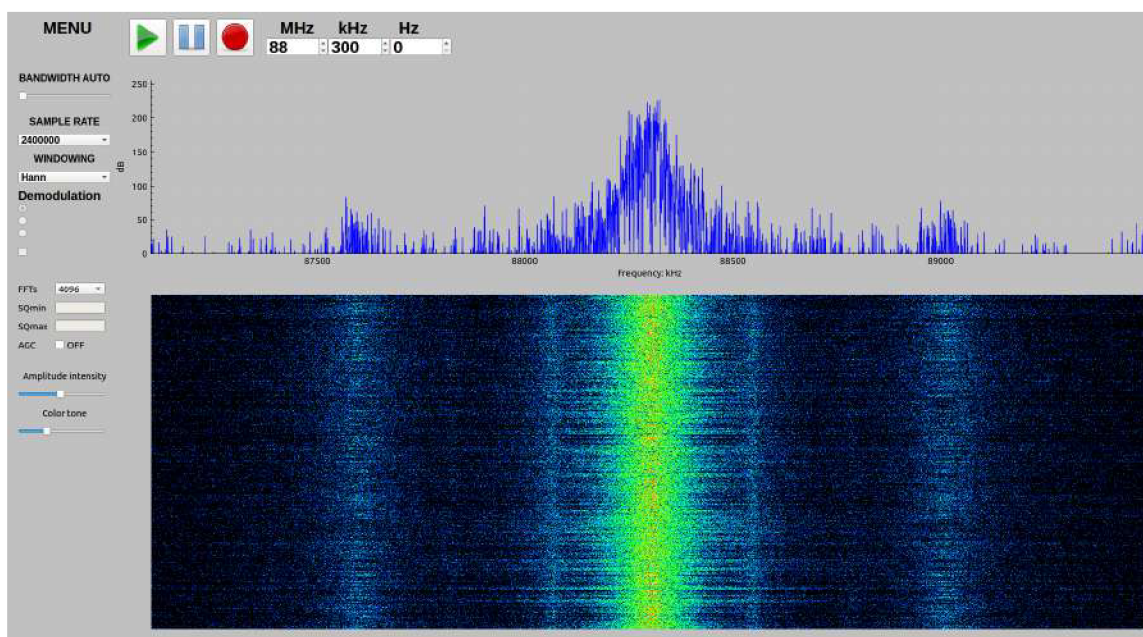
Na zobrazenie grafu spektra slúži trieda *Histogram*. Táto trieda obsahuje metódy, pomocou ktorých je možné za behu aplikácie meniť rozsah osí X a Y. To je veľmi dôležité hlavne pri zmene nosnej frekvencie alebo šírky zachytávaného pásma.

Zobrazenie spektrogramu sa realizuje pomocou grafickej knižnice OpenGL, pretože knižnica QCustomPlot by už nebola dostačujúca na také veľké množstvo dát (podrobnosti sú uvedené v sekcii 5.2.2). Tvorba spektrogramu je implementovaná v triede *Spectrogram*, kde je definovaná funkcia *rainbow()*, ktorá prevádza bajtové hodnoty 0 až 255 pomocou HSV <sup>1</sup> algoritmu na výslednú farbu. Výstupy tejto funkcie sa následne vykresľujú na jednotlivé pozície pixelov spektrogramu, pre konkrétnu frekvenciu. V prípade, že počet FFT binov je väčší alebo menší ako počet pixelov grafu, dochádza k decimácii alebo interpolácii pixelov. Pomocou metódy *max()* sa realizuje decimácia a interpolácia sa vykonáva pomocou aritmetického priemeru okolitých bodov.

<sup>1</sup>Hue Saturation Value - farebný model

## 6.4 Výsledné užívateľské rozhranie

Implementácia užívateľského rozhrania bola realizovaná podľa mockup návrhu uvedeného v sekcii 4.4. Avšak počas implementácie a testovania aplikácie som zistil, že zadávanie nosnej frekvencie do jediného textového poľa je veľmi nepohodlné. Preto konečná verzia aplikácie obsahuje textové polia rozdelené na tri časti - MHz, kHz a Hz. Každé textové pole sa dá ovládať aj kolieskom na myši. Taktiež niektoré textové polia, kde je možný výber iba z niektorých hodnôt ako napríklad zadávanie vzoriek pre rýchlu Fourierovu transformáciu, boli nahradené rozbalovacími zoznamami. Oproti plánovanému návrhu boli pridané aj niektoré funkcie, s ktorými sa pôvodne nepočítalo ako napríklad povolenie AGC a výber oknovej funkcie. Ovládanie týchto funkcií bolo pridané do Menu aplikácie. Výsledné užívateľské rozhranie je zobrazené na obrázku 6.1.



Obr. 6.1: Vzhľad užívateľského rozhrania



# Kapitola 7

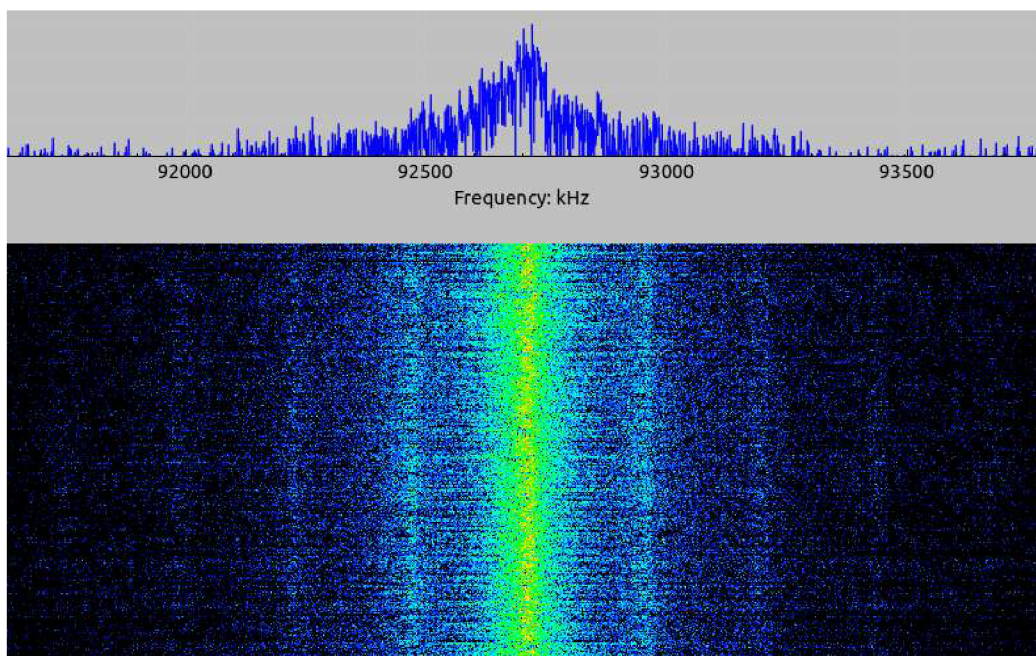
## Testovanie

Pre overenie správnosti návrhu a implementácie systému je dôležité vykonať testovanie. Testovanie bude prebiehať s použitím USB tunera, ktorý je uvedený v sekcii 2.2. Do tunera je pripojená jednoduchá drôtová anténa (sekcia 3.4.1).

Cielom testovania je overiť:

- Správnosť komunikácie medzi tunerom a aplikáciou
- Reakcia na zmenu vstupných hodnôt
- Správnosť vykresľovania grafu spektra a spektrogramu

### 7.1 Rozhlasové vysielanie



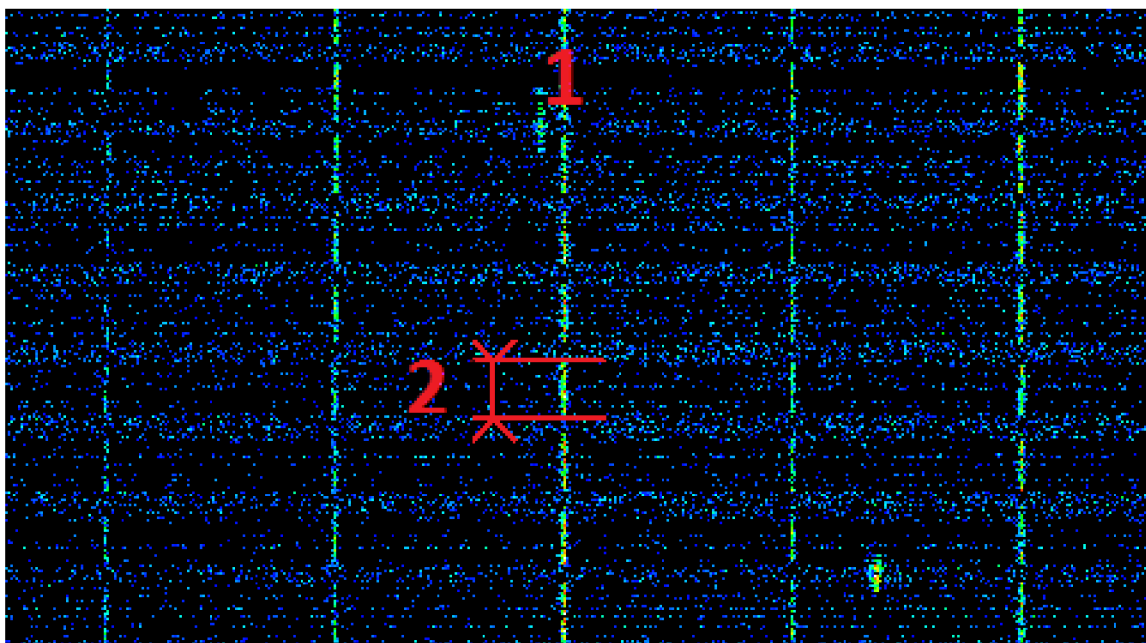
Obr. 7.1: Vybraná časť grafu spektra a spektrogramu, zobrazujúca prítomnosť rozhlasového vysielania (stred oboch grafov)

Prvým testom je pokus o zachytenie rozhlasového vysielania. Rozhlasové vysielanie má relatívne silný signál (hlavne v mestách) a preto by nemal byť problém nejakú rozhlasovú stanicu zachytiť. Na testovanie som zvolil frekvenciu 92,7 MHz, na ktorej by mala vysielat istá rozhlasová stanica. Po nastavení nosnej frekvencie a spustení zachytávania je možné vidieť nasledovný výstup spektrogramu, zobrazený na obrázku 7.1

Z obrázku 7.1 je zrejme, že na zvolenej frekvencii je prítomnosť nejakého rádiového signálu. Z uvedeného spektrogramu je tiež možné vidieť, že šírka zachyteného signálu je pomerne veľká, čo je typické práve pre rádiové vysielanie. Na úplné overenie správnosti výstupu stačí použiť bežný rádiový prijímač s nastavením frekvencie na hodnotu 92,7 MHz.

## 7.2 Diaľkové odomykanie automobilu

V tomto teste sa jedná o zachytenie signálu z kľúča s diaľkovým odomykaním automobilu. Tento typ signálu by mal mať oveľa menšiu šírku ako to bolo v predchádzajúcom teste s rádiovým vysielaním. Diaľkové ovládania automobilov pracujú vo väčšine prípadov na frekvencii 433,92 MHz.

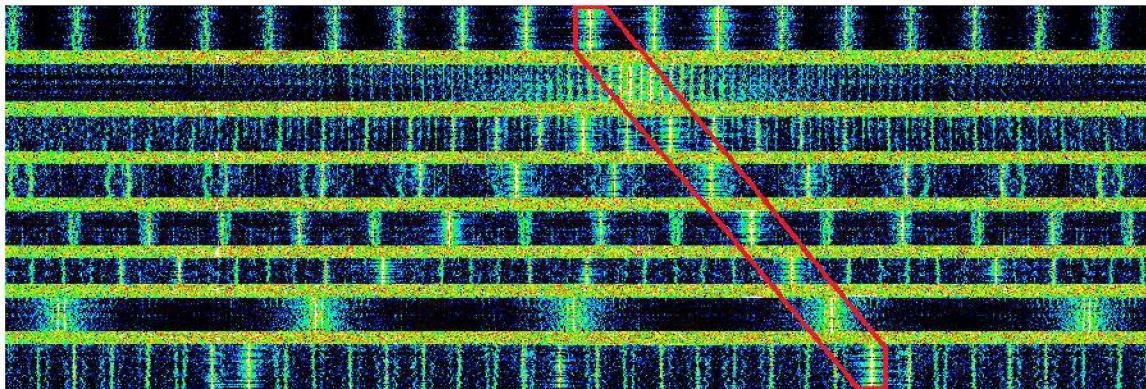


Obr. 7.2: Vybraná časť spektrogramu zobrazuje impulzy kľúča diaľkového odomykania, kde nosný signál je označený číslom 1 a dĺžka impulzu je označená číslom 2

Na obrázku 7.2 je vidno prítomnosť nosného signálu (označený číslom 1) a prítomnosť buď harmonických signálov, alebo vnútorného rušenia hardvéru. Z uvedeného obrázku je taktiež vidno, že signál vysielaný z kľúča je odosielaný v určitých impulzoch (dĺžka jedného impulzu je označená číslom 2).

### 7.3 PMR rádiostanica

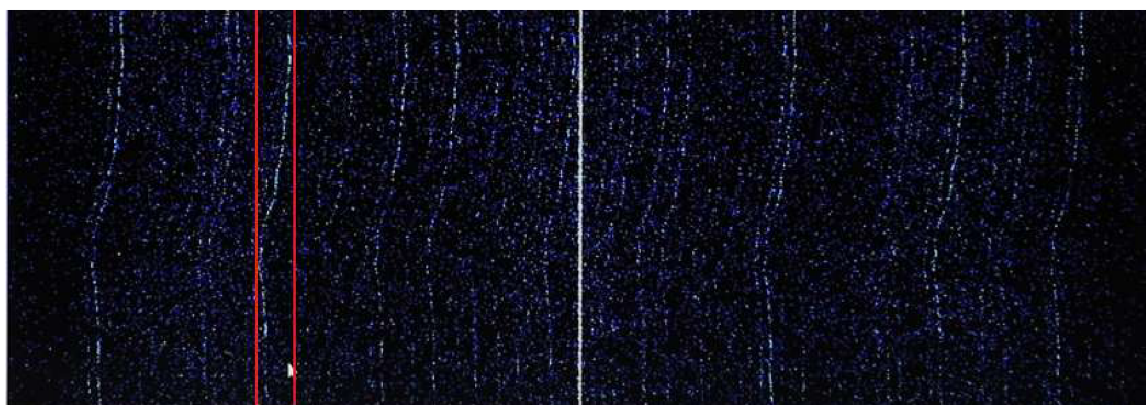
Na nasledujúci test bola použitá rádiostanica pracujúca v pásme PMR<sup>1</sup>. Účelom tohto testu je otestovať reakciu systému na zmenu frekvencie. Pásmo PMR má vyhradených osem kanálov, ktoré sú od seba posunuté o 12,5 kHz. Prvý kanál pásma PMR pracuje na frekvencii 446,00625 MHz. Princípom testovania bude vyslať signál pomocou rádiostanice a po pár sekundách zvolit nasledujúci kanál a týmto spôsobom prejsť všetkých osem kanálov.



Obr. 7.3: Vybraná časť spektrogramu zobrazuje vysielanie rádiostanice v pásme PMR, kde červené orámovanie znázorňuje posun frekvencie medzi kanálmi

Výsledok testu je zobrazený na obrázku 7.3, v ktorom je možné vidieť posun frekvencie (červené orámovanie). Všetky ostatné frekvencie, zobrazené v spektrograme sú opäť buď harmonické frekvencie, hardvérová interferencia alebo chyba v ovládači.

### 7.4 Teplotná odolnosť



Obr. 7.4: Vybraná časť spektrogramu zobrazuje posun parazitných frekvencií následkom teplotných rozdielov okolia (napr. v červenom orámovaní)

V predchádzajúcich testoch boli prítomné frekvencie, ktoré mohli pochádzať z vnútorného rušenia hardvérových súčiastok. Pri takomto lacnom zariadení je to očakávaný jav.

<sup>1</sup>PMR - personal mobile radio

Práve preto chcem týmto posledným testom poukázať na odolnosť USB tunera voči zmenám okolitej teploty. Priebeh testu bude vyzerat nasledovne. Pomocou aplikácie sa nájde a nastaví vhodný externý signál (najlepšie s malou šírkou) okolo ktorého je vidno aj parazitné frekvencie. Následne sa začne tuner zohrievať pomocou nejakého zariadenia generujúceho teplo. Po chvíli zohrievania sa tuner náhle ochladí studeným vzduchom. Priebeh testu je možné vidieť na videu v priloženom CD.

Na obrázku 7.4 je zachytený výsledok testu. Zmena okolitej teploty nemala vplyv na nosnú frekvenciu, čo je dobrý výsledok, ale ovplyvnila okolité frekvencie. Zmena resp. posun frekvencie je zvýraznený červeným orámovaním.

## 7.5 Vyhodnotenie testovania

Testovaním aplikácie boli overené všetky vopred stanovené body. Funkčnosť ovládača a správnosť spojenia medzi USB tunerom a aplikáciou bola overená spustením zachytávania signálov. Reakcia na zmenu vstupných hodnôt bola skontrolovaná pri vykonávaní testov, a to pri zadávaní a zmene frekvencií a ostatných parametrov. Správnosť vykresľovania grafov je zobrazená na predchádzajúcich obrázkoch, avšak pri niektorých testoch boli zistené frekvencie neznámeho pôvodu.

# Kapitola 8

## Záver

Cieľom tejto práce bolo vytvoriť systém, ktorý dokáže analyzovať rádiový signál zachytený pomocou DVB-T USB tunera a aplikovať algoritmy používané pri SDR, ktoré spracovávajú signál. Následne by systém mal byť schopný spracovaný signál vhodne interpretovať užívateľovi.

Po preštudovaní problematiky softvérovo definovaného rádia (sekcia 2 a 3) bola navrhnutá (kapitola 4) a následne implementovaná aplikácia na zachytávanie a analýzu rádiových signálov pomocou technológie RTL-SDR. Výsledná aplikácia je implementovaná v jazyku C++ s použitím Qt frameworku a ďalších podporných knižníc (kapitola 6).

Aplikácia je schopná zachytiť signál na zvolenej nosnej frekvencii s ďalšími užívateľskými parametrami. Aplikácia následne dokáže spracovať vstupný signál a formou grafu spektra a spektrogramu ho zobrazit užívateľovi. Bohužiaľ sa nepodarilo spojzdnit demoduláciu a následný audio výstup analyzovaného signálu.

Ciele, ktoré boli vytýčené, sa mi teda podarilo takmer úplne splniť. Základné požiadavky na funkčnosť aplikácie sú splnené, niektoré rozširujúce časti a funkcie systému (rôzne typy demodulácií) by bolo vhodné implementovať resp. opraviť a dokončiť pre dosiahnutie lepšej kvality celého systému.

V budúcnosti je možné pridať množstvo ďalších rozširujúcich funkcií, pretože to návrh tejto aplikácie a princíp softvérovo definovaného rádia jednoducho umožňuje.

# Literatúra

- [1] Gladišová, I.; Mihalík, J.: *Modulované signály*. LČSOV FEI TU Košice, 2016, ISBN 978-80-553-2442-5.
- [2] Harris, F. J.: *On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform*. Proceedings of the IEEE, vol 66, Január 1978.
- [3] Haykin, S.: *Communication Systems, 4th Edition*. Wiley, 2001, ISBN 0-471-17869-1.
- [4] Haykin, S.; Moher, M.: *An Introduction to Analog and Digital Communications, 2nd Edition*. Willey, 2007, ISBN 978-0-471-43222-7.
- [5] Míšek, P.: Softwarově Definované Rádio v provozních souvislostech. Október 2012, [Online; navštívené 7.4.2017].  
URL [http://www.crk.cz/FILES/SDR\\_V\\_PROVOZNICH\\_SOUVISLOSTECH.PDF](http://www.crk.cz/FILES/SDR_V_PROVOZNICH_SOUVISLOSTECH.PDF)
- [6] OM1AMM: Rozdelenie frekvencií. 2012, [Online; navštívené 1.5.2017].  
URL <http://ontheair.webnode.sk/frekvencne-pasma/rozdelenie-frekvencii/>
- [7] Prokeš, A.: *Rádiové Komunikační Systémy*. Ústav radioelektroniky FEKT VUT v Brně, 2013.
- [8] Valkama, M.: *Complex-Valued Signals and Systems*. Tampere University of Technology, [Online; navštívené 25.4.2017].  
URL [http://aaronsher.com/wireless\\_com\\_SDR/docs/complex.pdf](http://aaronsher.com/wireless_com_SDR/docs/complex.pdf)
- [9] Zavacký, J.: Krátkodobá Fourierova transformácia. December 2014, [Online; navštívené 3.5.2017].  
URL <http://www.posterus.sk/?p=17657>
- [10] Žalud, V.: *Moderní radioelektronika*. BEN, 2000, ISBN 80-86056-47-3.
- [11] Žatkovič, A.: Amplitúdová modulácia. [Online; navštívené 1.5.2017].  
URL <http://alzat.spseke.sk/radiote/modula/am/am.htm>

# Prílohy

## Príloha A

# Obsah priloženého pamäťového média

K tejto práci je priložené CD, ktorého obsah je rozdelený do nasledujúcich zložiek:

- **Aplikácia** - obsahuje všetky zdrojové súbory a použité knižnice
- **Latex** - obsahuje zdrojové súbory tohto textu v jazyku  $\text{\LaTeX}$
- **Text** - obsahuje text tejto práce vo formáte PDF
- **Video** - obsahuje nahrávku z testovania aplikácie



## Príloha B

# Inštalácia a spustenie

Ako prvé je potrebné pripojiť do USB zbernice počítača zariadenie používajúce čipset RTL2832U a do súboru `/etc/modprobe.d/blacklist.conf` pridať záznam `blacklist dvb_usb_rtl28xxu`. Na spustenie priloženej aplikácie je potrebné používať operačný systém na platforme Linux so stiahnutým a nainštalovaným frameworkom Qt. K funkčnosti aplikácie je tiež potrebné priložiť aj použité knižnice a to Simple-FFT, ktorá je dostupná na stránke <https://github.com/d1vanov/Simple-FFT>, knižnicu QCustomPlot, ktorá bola v tejto práci použitá vo verzii 2.0.0-beta a je dostupná na stránke <http://www.qcustomplot.com/index.php/download>. Na zabezpečenie komunikácie s tunerom je ešte potrebné priložiť ovládač, dostupný na stránke <https://github.com/pinkavaj/rtl-sdr/tree/master/src>. Potom sa môže v zložke so zdrojovými súborami vytvoriť adresár build a v tomto adresári pomocou konzoly zadať príkazy:

- `qmake ..`
- `make -j8`
- `./SpectralAnalyser`

Všetky potrebné zdrojové súbory vrátane knižníc a ovládača sú na priloženom CD v priečinku Aplikácia a pripravené na preklad pomocou príkazu `qmake` alebo v prostredí Qt Creator.

## Príloha C

# Návod na obsluhu

Obsluha aplikácie je veľmi jednoduchá. Pri spustení programu sú už prednastavené všetky potrebné vstupy a užívateľovi stačí zvoliť nosnú frekvenciu pomocou vstupného poľa rozdeleného na tri časti. V ľavej časti sa nastavuje frekvencia v MHz, v strednej časti v kHz a v pravej časti v Hz. Potom už len stačí zapnúť zachytávanie pomocou zeleného tlačidla play.

Na úpravu parametrov analýzy slúžia jednotlivé komponenty v časti menu. Ich význam je nasledovný:

- **Bandwidth** - nastavenie šírky pásma pomocou posuvníka; v polohe úplne vľavo, je nastavená automatická šírka pásma
- **Sample rate** - nastavenie vzorkovacej frekvencie zachytávania
- **Windowing** - výber oknovej funkcie
- **Demodulation** - výber demodulácie (nie je implementované)
- **FFTs** - nastavenie počtu vzoriek FFT (FFT bins)
- **SQ min/max** - nastavenie squelch (funguje iba s demoduláciou)
- **AGC** - povolenie hardvérového AGC (automatické vyrovnávanie citlivosti)
- **Amplitude intensity** - nastavenie koeficientu amplitúdy
- **Color tone** - nastavenie farby spektrogramu

V hornej časti obrazovky sú tri tlačidlá, ktoré slúžia na riadenie zachytávania. Na pozastavenie zachytávania slúži prostredné modré tlačidlo so znakom pause. Na opätovné spustenie zachytávania slúži zelené tlačidlo play. Zachytený I/Q sinál je možné nahráť pomocou červeného tlačidla record. Nahrávka sa ukladá do súboru v priečinku build vo formáte wav.



## Príloha D

# Plagát z konferencie Excel@FIT 2017

**27** **Excel@FIT 2017**

**Analyzátor signálu založený na DVB-T USB tuneri**

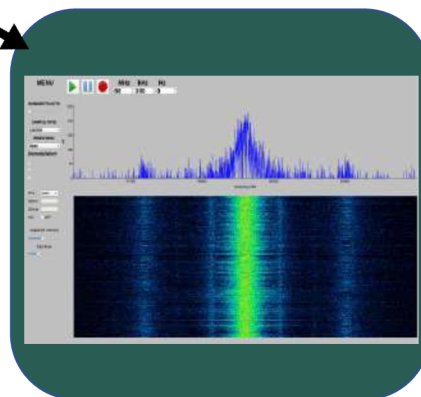


### USB DVB-T tuner

- **Lacné (7€ = 200CZK)**
- **Rozsah 25-1750MHz**
- **Spracovanie vstupu pomocou RTL2832U**
- **Výstup vo forme I/Q signálu**

### Aplikácia na analýzu signálu

- **Nastavenie parametrov zachytávaného signálu**
- **Aplikovanie algoritmov (FFT, AM, FM, apod.)**
- **Vykreslenie grafov (histogram a spektrogram)**
- **Audio výstup**
- **Vhodné napr. pre rádioamatérov**



- Použitá technológia:**
- **SDR – Softvérovo Definované Rádio**
  - **Jazyk C++**
  - **Qt framework**
  - **OpenGL**

**Martin Belica**