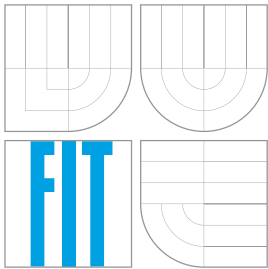


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **FM VYSÍLAČ NA PLATFORMĚ RASPBERRY PI**

FM TRANSMITTER ON RASPBERRY PI

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**JAN KOHOUTEK**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Ing. ZDENĚK VAŠÍČEK, Ph.D.**

BRNO 2016

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

Ústav počítačových systémů

Akademický rok 2015/2016

**Zadání bakalářské práce**

Řešitel: **Kohoutek Jan**

Obor: Informační technologie

Téma: **FM vysílač na platformě Raspberry Pi**  
**FM Transmitter on Raspberry Pi**

Kategorie: Vestavěné systémy

**Pokyny:**

1. Seznamte se s platformou Raspberry Pi, zejména možnostmi připojení externích periferií a podpory bezdrátového přenosu souborů. Prostudujte možnosti realizace FM vysílače, princip kódování stereo signálu a princip přenosu RDS informací.
2. Navrhněte systém, který bude umožňovat audio reprodukci v radiovém pásmu FM. Systém koncipujte tak, aby jej bylo možné ovládat externě např. pomocí chytrého telefonu. Ovládáním je myšlena konfigurace parametrů (např. nosná frekvence) a možnost vložení vlastních audio souborů.
3. Zpracujte studii na výše uvedené téma.
4. Navržený systém realizujte a vyhodnoťte jeho parametry.

**Literatura:**

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodu 1 a 2 zadání.

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: **Vašíček Zdeněk, Ing., Ph.D., UPSY FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačových systémů a sítí  
612 66 Brno, Božetěchova 2



---

doc. Ing. Zdeněk Kotásek, CSc.  
vedoucí ústavu

## Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat přehrávač, který bude umožňovat prehrávání vlastních souborů v pásmu FM. Přehrávač tvoří vysílač založený na platformě Raspberry Pi a mobilní aplikace umožňující bezdrátové ovládání přehrávače. Technická zpráva je členěna na tři části. Uvodní část se věnuje technologiím pro bezdrátový přenos dat a přenosu audio signálu v pásmu FM. Následuje popis navrženého řešení a jeho implementace. Závěrečná část popisuje navržený prototyp a vyhodnocení jeho parametrů.

## Abstract

Aim of this bachelor thesis is to desing and implement player, which will be able to play its own files in FM band. The player consists of transmitter based on Rapsberry Pi and mobile application that allows wireless control of this player. Technical report is divided into three main parts. The initial part describes technologies for transmission of audio signal in FM band. Followed by description of suggested solution and its implementation. Concluding part describes the designed prototype and assesment of its parameters.

## Klíčová slova

Vysílání FM, Raspberry Pi, Bluetooth, WiFi

## Keywords

FM broadcasts, Raspberry Pi, Bluetooth, WiFi

## Citace

KOHOUTEK, Jan. *FM vysílač na platformě Raspberry Pi*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Vašíček Zdeněk.

# FM vysílač na platformě Raspberry Pi

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdeňka Vašíčka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jan Kohoutek  
18. května 2016

## Poděkování

Rád bych na tomto místě poděkoval vedoucímu mé práce panu Ing. Zdeňku Vašíčkovi, Ph.D. za jeho obětavost, trpělivost a velmi dobré rady při práci na projektu.

© Jan Kohoutek, 2016.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Teoretický úvod</b>	<b>4</b>
2.1	Princip vysílání analogové a digitální informace v pásmu FM	4
2.1.1	Vysílání FM	5
2.1.2	Radio data system	5
2.1.3	Digitální vysílání	8
2.1.4	Z pohledu zákona	8
2.2	Princip bezdrátového přenosu dat	8
2.2.1	WiFi	8
2.2.2	Bluetooth	9
2.3	Existující řešení	9
2.3.1	I <sup>2</sup> C rozhraní	10
<b>3</b>	<b>Rozbor problematiky a návrh řešení</b>	<b>12</b>
3.1	Využití linux platformy	13
3.2	Navržené řešení	14
3.2.1	Modul vysílače pro Raspberry Pi	14
3.2.2	Přehrávač skladeb	15
3.2.3	Management skladeb	15
3.2.4	Mobilní aplikace	15
<b>4</b>	<b>Realizace</b>	<b>17</b>
4.1	Vysílací obvod	17
4.2	Jádro přehrávače	18
4.2.1	Vlákno přehrávání hudby	20
4.2.2	Vlákno pro ovládání	20
4.3	HTTP server a HTML stránka	23
4.3.1	Klient	23
4.4	Android aplikace	24
4.4.1	Třída definující uživatelské rozhraní	24
4.4.2	Třída pro komunikaci skrze bluetooth	25
<b>5</b>	<b>Vyhodnocení parametrů navrženého řešení</b>	<b>27</b>
5.1	Prototyp	27
5.2	Rychlost přenosu	28
5.3	Kvalita zvuku	28

<b>6 Závěr</b>	<b>29</b>
<b>Literatura</b>	<b>30</b>
<b>Přílohy</b>	<b>33</b>
Seznam příloh . . . . .	34
<b>A Obsah CD</b>	<b>35</b>
<b>B Schéma vysílacího obvodu</b>	<b>36</b>
<b>C Navržený tištěný spoj</b>	<b>37</b>
<b>D Seznam potřebných součástek pro obvod</b>	<b>38</b>

# Kapitola 1

## Úvod

Hudba nás obklopuje a pro některé je nedílnou součástí každodenního života. A to ať při řízení vozu, oslavě s přáteli či rodinou, nebo při práci. Hlavní motivací tohoto projektu je fakt, že poskytované přenosné přehrávače na dnešním trhu nabízí v drtivé většině možnost přehrávání jen z jediného zdroje reflektujícího výběr jednoho člověka. Hudební vkus je ale skutečně velmi individuální věcí a tak vyvstaly tendence vyřešit problém i pro skupinu posluchačů, tak aby si každý přišel na své.

Přenos audia pomocí FM technologií je dnes stále nejrozšířenější volbou. Přijímačem tohoto signálu je vybavena velká většina hudebních přehrávačů, od těch nejzákladnějších, až po ty velmi sofistikované.

Hlavním cílem projektu je návrh a vytvoření FM vysílače přenášejícího hudbu na platformě Raspberry Pi, s možností bezdrátového ovládání. Tento přehrávač má být schopen nejen vysílat hudbu, ale také přijímat požadované hudební soubory a to bezdrátově, od několika uživatelů paralelně. Pro ovládání je doporučen mobilní telefon. Záměr je také co nejjednodušší a nejpohodlnější ovládání a používání.

K dosažení cíle je tedy zapotřebí navrhnout vhodný FM vysílač, přizpůsobený pro zvolenou platformu Raspberry Pi. Vytvořit program, schopný přijímat a přehrávat hudební soubory a vyvinout jednoduché uživatelské rozhraní, zaměřené na mobilní platformy. Celek poté spojit v prototyp a ten otestovat.

Tato práce je členěna následovně. První kapitola je věnována problematice vysílání v pásmu FM a bezdrátovému přenosu dat (kapitola 2), následovaná rozborem zadání. Další část pojednává o návrhu samotné práce (sekce 3.2) a v kapitole 4 je popsána realizace tvořeného řešení. Práci uzavírá vyhodnocení výsledků sestaveného prototypu (kapitola 5).

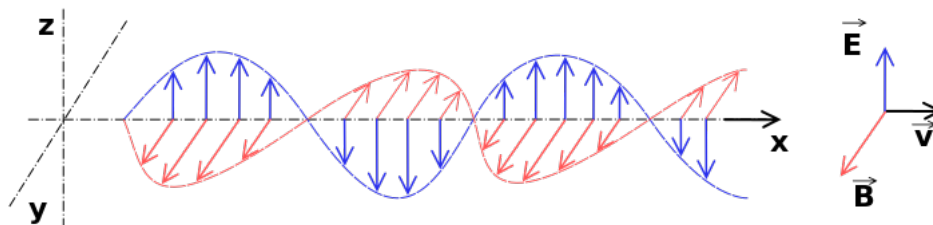
## Kapitola 2

# Teoretický úvod

V této kapitole jsou naznačeny a vysvětleny principy vysílání v pásmu FM a bezdrátové přenosy dat. Obsahuje také souhrn existujících řešení na trhu.

### 2.1 Princip vysílání analogové a digitální informace v pásmu FM

Přenos informací vzduchem má na svědomí elektromagnetické vlnění. To vzniká při zapojeném oscilačním obvodu, který generuje elektrické pole, doprovázené polem magnetickým. Tento jev se dá reprezentovat jako dvojice vektorů elektrické intenzity a magnetické indukce. Vektory jsou na sebe vzájemně kolmé a oba jsou kolmé na směr šíření. Existenci těchto vln zjistil skotský vědec James Clerk Maxwell. Který v roce 1864 uvedl, že i světlo je druhem elektromagnetického vlnění a musí tedy existovat i vlnění s porovnatelně nižšími frekvencemi. Což dokázal ve své laboratoři Heinrich Hertz roku 1887 a vznikl pojem „rádiové vlny (rádiové záření)“. [19][18][17]



Obrázek 1: Ukázka šíření elektromagnetického záření. Kde vektor  $\vec{E}$  označuje elektrickou intenzitu,  $\vec{B}$  magnetickou indukci a  $\vec{V}$  reprezentuje směr šíření.<sup>1</sup>

Elektromagnetické záření se dělí na části podle vlnové délky. Jednou z nich je i viditelné světlo (760 nm – 390 nm). Další jsou například mikrovlny, infračervené záření anebo Gama záření. Rádiové vlny mají délky od 1 m až po tisíce kilometrů a dále se člení opět podle vlnové délky. Jednotlivé oblasti jsou pak používány pro různé účely. Například pro leteckou a námořní navigaci, meteorologické služby, WiFi, mobilní telefony a v neposlední řadě také rádiové a televizní vysílání. Pro FM (frekvenční modulace) vysílání se užívá pásmo

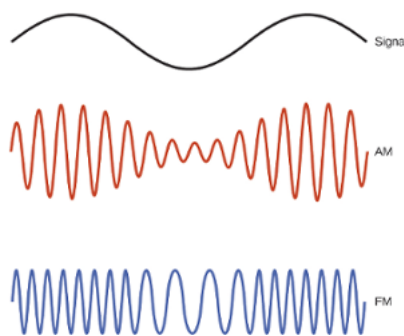
<sup>1</sup>Zdroj: [https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Onde\\_electromagnetique.svg/714px-Onde\\_electromagnetique.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Onde_electromagnetique.svg/714px-Onde_electromagnetique.svg.png)



velmi krátkých vln (anglicky Very High Frequency (VHF)) ve frekvenčním rozsahu 30 – 300 MHz.[18][15]

### 2.1.1 Vysílání FM

Frekvenční modulace byla objevena a dokumentována poprvé v roce 1932 Edwinem Armstrongem. Oproti do té doby používané amplitudové modulaci (AM), přinášela mnohem lepší kvalitu přenášené informace a menší rušení. Vysílání zvuku, který je lidem slyšitelný (20 Hz – 20kHz), se přenáší kombinací signálu zvuku s frekvencí z pásma vysílání. Ta se nazývá nosná a pro FM v České republice musí být v rozsahu 87,5 – 108 MHz, podle rozhodnutí Českého telekomunikačního úřadu. Amplitudová modulace funguje na principu změny amplitudy nosné frekvence, na závislosti hodnoty zvukového signálu. Výhodou tohoto způsobu je jednoduché sestavení vysílačů a přijímačů. Největší nevýhoda a hlavní důvod přechodu na FM je fakt, že amplituda přenášeného signálu, je snadno ovlivňována rušivými prvky, jako jsou přírodní podmínky, nebo elektronická zařízení. Tyto problémy eliminovala frekvenční modulace signálu. Opět jde o kombinaci nosné frekvence s datovou, zde je ale podle hodnoty signálu přenášené informace upravována frekvence (viz obrázek 2). Změny frekvence jsou omezeny maximální odchylkou, aby nedošlo k rušení jiné nosné frekvence. [18][17][23]



Obrázek 2: Srovnání amplitudové a frekvenční modulace signálu.<sup>2</sup>

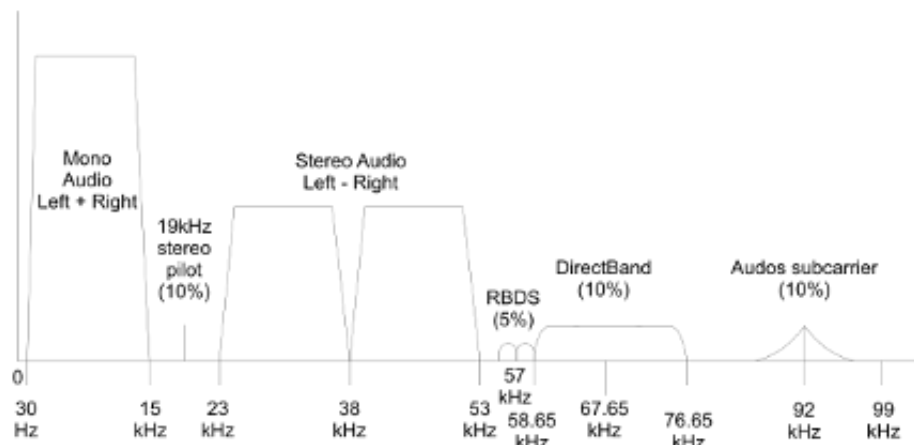
FM vysílání se těšilo velkému komerčnímu úspěchu. Velká změna ale přišla v druhé polovině padesátých let dvacátého století, kdy kvalita vysílání značně vzrostla díky kódování a vysílání signálu ve dvou kanálech. Frekvence pilotního signálu byla stanovena na 19 kHz. Dvojnásobek (38kHz) byl určen jako subnosná frekvence pro pásma jednotlivých kanálů. Levý i pravý kanál mají stanovené frekvenční pásmo 23 – 53 kHz. Nadále je také vysílán mono signál, kódován na frekvenčním rozsahu od 30 Hz do 15 kHz. Roku 1969 bylo objeveno i kvadrofonní vysílání, tedy se čtyřmi kódovanými zvukovými kanály, později dokonce Dolby FM, jež však nebylo komerčně úspěšné. [12]

### 2.1.2 Radio data system

V sedmdesátých a osmdesátých letech bylo velmi populární FM vysílání rozšířeno téměř po celém světě. Při stále větším a větším počtu přidělovaných frekvenčních pásem stanicím,

<sup>2</sup>Zdroj: [https://cnx.org/resources/24aa43fe5369f27fd056a5a9064ade6abe354d1b/CNX\\_Chem\\_06\\_01\\_AMFM.jpg](https://cnx.org/resources/24aa43fe5369f27fd056a5a9064ade6abe354d1b/CNX_Chem_06_01_AMFM.jpg)

<sup>3</sup>Zdroj: [https://en.wikipedia.org/wiki/FM\\_broadcasting#/media/File:RDS\\_vs\\_DirectBand\\_FM-spectrum2.svg](https://en.wikipedia.org/wiki/FM_broadcasting#/media/File:RDS_vs_DirectBand_FM-spectrum2.svg)



Obrázek 3: Typická skladba kódovaného rádiového spektra.<sup>3</sup>

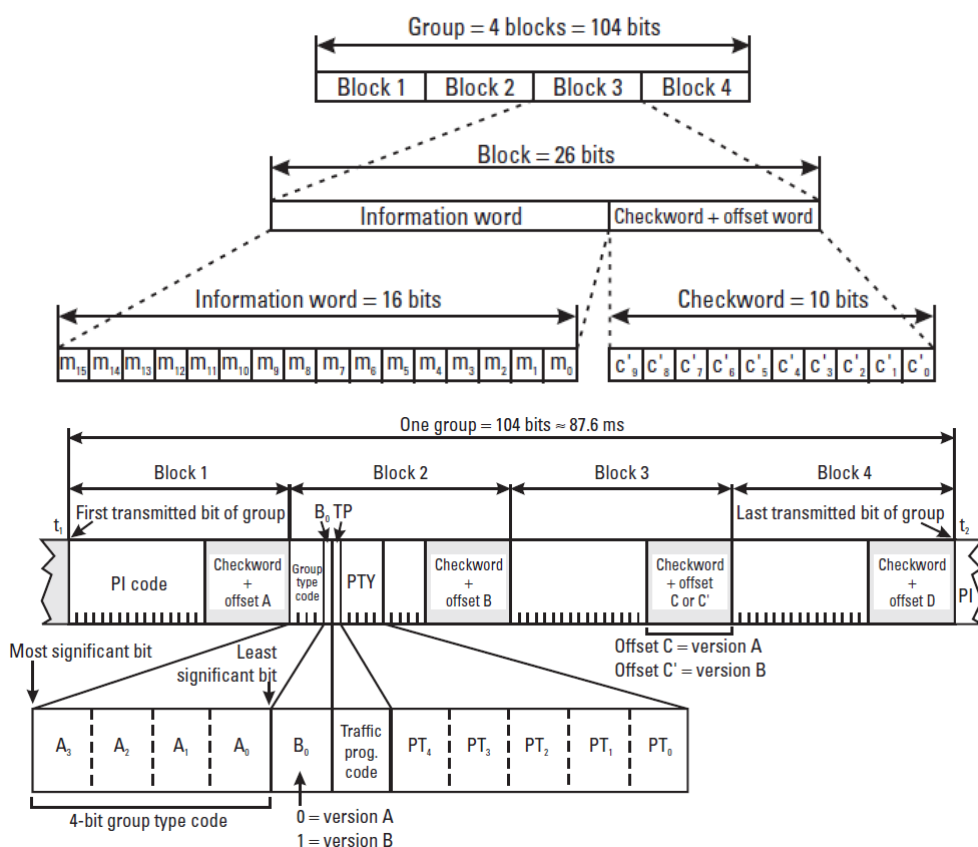
bylo někdy těžké naladit přesně požadovanou stanici na přijímači. Toto řeší Radio Data System (RDS), který je na trhu od roku 1987 a jehož efektivní evoluce stále pokračuje. Stojí za ním EBU (European Broadcasting Union), což je sdružení vysílacích stanic, se sídlem v Ženevě. Základním účelem RDS je k vysílání přidat i doplňkové informace.

Následuje výčet některých z nich:

- **Programme type (PTY)** – Slouží k určení typu vysílacího programu. Ten je dán tabulkou dostupnou z [5].
- **Alternative frequencies (AF)** – Stejná stanice může být na různých vysílacích vysílána na jiné frekvenci. Díky AF je automaticky přijímač přeladěn.
- **Clock time (CT)** – Dovoluje nastavení data a času přijímače podle vysílané informace.
- **Programme Identification (PI)** – Identifikace stanice pro možnost automatického vyhledávání, či přeladění.
- **Programme Service (PS)** – Umožňuje přenos a zobrazení až osmi znaků s názvem stanice.
- **Radiotext (RT)** – Vysílaná textová zpráva, až 64 znaků. Používaná pro zobrazení autora skladby, nebo názvu.
- ...

Jak je uvedeno i na obrázku 3, základní frekvenci RDS informací je 57 kHz. Opět násobek pilotní frekvence (19 kHz). Díky použití násobků pilotní frekvence, takzvaných harmonických signálů, je dosaženo co nejmenšího rušení. Subnosná frekvence je modulována pomocí binárního klíčování fázovým posunem (BPSK). Odchylka fáze je  $\pm 90^\circ$ . Jde o nejrobustnější ze všech typů klíčování fázovým posunem, protože vyžaduje nejvyšší úroveň šumu nebo zkreslení, aby došlo při dekódování k chybě. Je však schopné kódovat pouze 1 bit, takže není vhodné pro aplikace vyžadující vysoké přenosové rychlosti. Rychlost přenosu se dá vypočítat ze subnosné frekvence, podělené 48 (hodinový signál). Základní přenosová rychlost je tedy 1187,5 bitů za sekundu. [14][20]

Přenášená data se dělí na *skupiny*. Každá má 104 bitů a dále se člení na *bloky* po 26 bitech, které se skládají z 16bitového *informačního slova* a *kontrolního slova* (10 bitů). Přenos dat je naprosto synchronní. V toku nejsou žádné mezery ani mezi bloky, nebo skupinami. První blok každé skupiny vždy obsahuje PI kód. První čtyři bity z druhého bloku reprezentují typ přenášené informace (některé uvedené v seznamu výše). Pátý bit určuje verzi informace. Následuje bit označený jako TP, jež ve spolupráci s TA přepínačem pomohou posluchači nepropásnout hlášení o dopravní situaci (Traffic announcement/ programme). Pokud je TP nastaven, jedná se o stanici, která má ve svém vysílání zařazeny informace o dopravě. Dynamicky nastavovaný přepínač TA je potom indikátorem, že zrovna vysílání o dopravě běží. Tyto informace jsou tedy nejpoužitelnější v autorádiích, která mohou pozastavit přehrávání nosičů a spustit příjem dopravních informací. Součástí druhého bloku každé skupiny je také PTY informace. [20]



Obrázek 4: Složení datového toku přenosu RDS. Převzato z [20]

RDS implementace a forma se můžou lišit napříč světem. Například ve spojených státech je oficiálně používán Radio Broadcast Data System (RBDS), rozdíly v obou verzích jsou ale zanedbatelné a dokonce i zařízení jsou kompatibilní. Dne 3. listopadu 2015 se konal sjezd společnosti RDS Forum, jejíž členy jsou přední světoví výrobci vysílačů a přijímačů FM signálů. Na této schůzi v Berlíně se ustanovila nová podoba RDS druhé verze, tedy RDS2. Ta by měla přinést řadu vylepšení, jako jsou lepší a přesnější dopravní informace, nebo předpovědi počasí, možnost přenášet obraz (logo stanice), UTF-8 texty, delší texty, dokonce texty formátované jako HTML/CSS. To vše zároveň se stálou podporou přijímačů, jež mají vestavěn RDS první generace. [14][4]

### 2.1.3 Digitální vysílání

Předpovědi pro zastaralé analogové vysílání nebyly příznivé, když se začalo po roce 2000 na scénu rádiového vysílání dostávat digitální rádio. To pracuje na vyšších frekvencích a tak může poskytovat i rychlejší datové přenosy. Zvukový signál je nejdříve komprimován, poté digitální modulací převeden a vyslán. I v České republice několik stanic provozuje tento systém. Ovšem přechod na digitální vysílání je poměrně finančně náročný a vysílání není kompatibilní s přijímači analogového signálu. I díky zlepšujícímu se RDS je tedy stále nejoblíbenější a nejpoužívanější analogové FM vysílání. Výhody digitálního rádia jsou ale značné. Díky vysílání v „multiplexu“, může být jednou frekvencí vysíláno více stanic. Při kompresi zvuku například kodekem MPEG4 je kvalita vysoká a to i v méně přístupných oblastech. Zároveň je tato metoda i energeticky úspornější a nabízí širokou škálu doplňujících informací. [21]

### 2.1.4 Z pohledu zákona

Kontrolu a přidělování kmitočtových pásem v České republice zajišťuje Český telekomunikační úřad (ČTU). Podle vydané vyhlášky tímto úřadem v roce 2014 [22], je možné vysílat na zařízení bez vlastnění licence. Jsou to však zařízení s maximálním výkonem 50 nW. Jejich předpokládaný dosah je tedy v jednotkách metrů. Navíc ovšem toto zařízení musí nést od výrobce označení CE. Pro možnost tohoto označení je třeba prohlášení o shodě, podle zákona 426/2000 Sb [24]. Tento zákon uvádí celou řadu nutných technických požadavků, jimž musí zařízení odpovídat. Mimo jiné jsou to požadavky na „ochranu zdraví a bezpečnosti uživatele a jakékoli další osoby“ [24], efektivní využití kmitočtového pásma, obsah prvků zajišťující bezpečnost dat uživatelů atp. Splnění požadavků je nutno doložit kvalitní dokumentací, sadou kontrol a zkoušek, kontrolou jakosti autorizovanou osobou a řadou dalších nezbytností. Vývoj vlastního vysílacího zařízení je tedy běh na dlouhou trať. Eliminace těchto problémů je možná koupí již na trh uvedeného řešení. Některá z nich jsou uvedena v sekci 2.3. [25]

Dalším právním prvkem je samozřejmě dodržování autorských zákonů a práv pro vysílání souborů resp. hudby.

## 2.2 Princip bezdrátového přenosu dat

Bezdrátové technologie pro přenosy dat fungují na principu rádiového záření. (Zmíněno v sekci 2.1). Liší se použitými způsoby kódování binárních dat a frekvencemi přenosu. Využívané jsou tedy nejen pro přenos zvuku, nebo obrazu, ale i pro komunikaci mezi dvěma a více zařízeními, ať už na vzdálenost centimetrů, nebo desítek tisíc kilometrů.

### 2.2.1 WiFi

Jednou z těchto technologií je hojně využívané Wi-Fi (WiFi). Jde o obchodní značku neziskové organizace Wi-Fi Alliance<sup>4</sup>, fungující od roku 1999. Její certifikace výrobků, užívající WiFi, je založena na standardu IEEE 802.11 z roku 1997. Jde o sadu specifikací pro implementaci bezdrátových lokálních sítí. Tu spravuje Institute of Electrical and Electronics Engineers (IEEE). Specifikace se týkají především fyzické vrstvy a části vrstvy linkové, zvané Media Access Control (MAC), z OSI modelu. Specifikací z rodiny 802.11 je celá řada.

<sup>4</sup>Wi-Fi Alliance: <http://www.wi-fi.org/>

V průběhu historie vždy přinášely jistá vylepšení. Pracují na frekvenci 2,4 GHz, 5 GHz, nebo 60 GHz. Jeden ze standardů dokonce komunikuje na frekvenci 900 MHz, což přináší výhody nižší spotřeby a delšího dosahu signálu. Naopak díky vysokým frekvencím lze dosáhnout přenosových rychlostí až 100 Gbit/s. Spojení mezi dvěma body je typu „half-duplex“, komunikace tedy v čase probíhá pouze jedním směrem. WiFi funguje na bázi přístupových bodů, ke kterým je možné se bezdrátově připojit a odesílat požadavky. Když přístupový bod získá požadavek, odešle ho po připojeném kabelu do internetové sítě a naopak. Velká výhoda WiFi sítí je kompatibilita s většinou operačních systémů na trhu. [10][13]

Samozřejmě bylo nutné zajistit bezpečnost přenášených dat. Vzniklo tedy několik bezpečnostních technik, jako WEP, WPA nebo WPS. Vzhledem k tomu, že je ale signál šířen vzduchem a teoreticky nelze zabránit útočníkovi v přístupu k síti, nejsou zabezpečení sto-percentní. [13]

### 2.2.2 Bluetooth

Další technologií pro bezdrátové přenosy, využívající technologii rádiových vln, je Bluetooth (BT). V dnešní době za standard bluetooth komunikace zodpovídá Bluetooth Special Interest Group (SIG), nezisková organizace poskytující licence výrobkům a výrobcům bluetooth využívajících zařízení. BT je vyvíjeno už od roku 1989. Jeho frekvenční rozmezí je mezi 2402 až 2480 MHz. Při komunikaci si rozdělí přidělené pásmo na 79 kanálů a pak skokově mění frekvenci pro odesílání bitů. Tato metoda se nazývá frequency-hopping spread spectrum. Komunikace mezi zařízeními se řídí modelem master/slave, kde jedno zařízení přebírá kontrolu nad přenosem dat. Master může odesílat data až sedmi zařízením v roli slave. Takto vytvořené sítě se nazývají piconets (pikosít). Bluetooth je určeno především pro energeticky nenáročné aplikace, proto je také dosah komunikace velmi omezen (10 m). Bluetooth disponuje několika předpřipravenými profily. Ty jsou specifické pro některé druhy použití a umožňují ušetření energie. Jedná se například o komunikaci s bezdrátovou myší, nebo klávesnicí, ovládání handsfree zařízení, stream hudby, komunikace se zdravotním zařízením apodobně. [11][9]

Prozatím poslední čtvrtá verze jádra (2010) přinesla novinku v podobě Bluetooth low energy (BLE), energeticky velmi úsporný režim komunikace. Pro komunikaci dělí frekvenční pásmo na 40 kanálů a využití najde především u dlouhodobě běžících senzorů. [9]

## 2.3 Existující řešení

Podobných výrobků, jakým je cíl tohoto projektu je samozřejmě na trhu celá řada. Tedy FM vysílače přehrávající hudbu. Ve valné většině jde o vysílače „do auta“, které se připojují do auto konektoru s napájením 12, nebo 24 V. Pro nastavení nosné frekvence slouží většinou LED displej a sada ovládacích tlačítek. Existují i varianty s dálkovým ovládním. Problém ale je, že přehrávaná data čerpají z externího úložiště (USB, paměťová karta), na kterém je zpravidla výběr hudby jeho majitele. Zároveň jen vyjímečně je umožněno napájení i jiným způsobem, než auto zásuvkou. Dalšími variantami jsou například vysílače, získávající data přes audio vstup, nebo skrze bluetooth stream. Napájené mohou být USB konektorem, nebo jsou kompletně součástí mobilního telefonu. Problém s jedním úložištěm ovšem přetrvává.

Výše uvedené problémy je možné eliminovat použitím vhodného vysílacího čipu, v kombinaci s vlastním navrženým obvodem. Na trhu je opět na výběr několik variant. Každá má

<sup>5</sup>Zdroje: <http://cdn.electroworld.cz/images/img-large/3/154353.jpg>, <http://www.best-apple.cz/450-1266-thickbox/in-car-led-stereo-fm-transmitter-pro-iphone-ipod-ipad.jpg>



Obrázek 5: Ukázka komerčních vysílačů.<sup>5</sup>

své přednosti, či nedostatky.

Prvním z nich je čip od společnosti Texas Instruments a nese název **SN761634**. Jde o jednoduché řešení vysílání FM signálu v rozsahu 76 – 108 MHz. Přičemž napájení je poměrně nenáročné (2,5 – 4 V). Výstupní síla je přitom od -7 – 4 dBm. Bohužel však nedisponuje RDS vysíláním. [6]

Druhého kandidáta má na svědomí firma Japan Radio. Čip **NJM2295A** si klade trochu vyšší nároky na napájení. Rozmezí od 2,7 V do 7,0 V je poměrně široké, nicméně je obvod schopen vysílání na frekvencích 50 – 350 MHz. Ani tento ovšem nepodporuje RDS informace. [1]

Dalším možným řešením je čip **QN8027**, zástupce firmy Quintic. Frekvenční rozsah vysílání je 76 – 108 MHz, tedy v pásmu FM. Napájecí napětí je od 2,7 do 4,2 V. Napájení dokonce sám odpojí, pokud na vstup alespoň jednu minutu není přiveden audio signál. Disponuje i vysíláním RDS/RBDS informacemi. [8]

Posledním zmíněným řešením je **Si4713** z dílny Silicon Laboratories. Rozsah frekvencí odpovídá FM standardu (76 – 108 MHz). Napájení je v rozmezí 2,7 V až 5,5 V. Podporuje i kódování RDS/RBDS. Krom I<sup>2</sup>C rozhraní je schopen komunikace i přes SPI sběrnici. Dokonce dovoluje digitální zvukový vstup. [7]

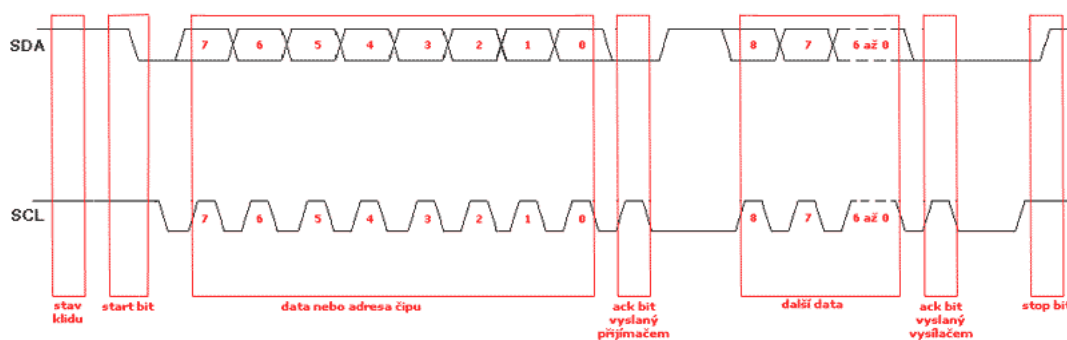
Je na místě upozornit, že bez udělení licence anebo vyhotovením a potvrzením posudku o shodě s předpisy pro vysílání, od Českého telekomunikačního úřadu, je nelegální s těmito čipy vysílat. (Více v sekci 2.1.4.) Každý výše zmíněný obvod také poskytuje komunikaci skrze I<sup>2</sup>C rozhraní, jež je popsáno v sekci 2.3.1.

### 2.3.1 I<sup>2</sup>C rozhraní

Jde o jednoduché dvou vodičové komunikační rozhraní, vyvinuté roku 1982 firmou Philips Semiconductors (nyní NXP Semiconductors). Pro komunikaci stačí jen dvě obousměrné linky. Jedna přenáší data (SDA), druhá pro hodinový signál (SCL). Může dosahovat přenosové rychlosti až 3,4 Mbit/s. V jednosměrné komunikaci dokonce 5 Mbit/s. Komunikace probíhá modelem master/slave. Jedno zařízení připojené na sběrnici v roli master odesílá data všem ostatním (slave). [2]

Pokud jsou oba spoje ve stavu vysoké impedance, pak je klidový stav. Nic se neodesílá. Startovací bit je rozpoznán změnou stavu na datovém vodiči, při nastaveném hodinovém signálu v log. 1. Potom začnou proudit data. Změna hodnoty na datovém vodiči je provedena jen při nulové hodnotě hodinového signálu. Odeslaných prvních 7 bitů může být adresou požadovaného zařízení pro komunikaci. Pokud je připraveno přijímat další data, odpoví ACK bitem. Data jsou v nule, hodinový signál nastaven. Data jsou posílána po bytu. Pokud

jsou odeslána všechna data, vysílač dá znamení tím, že odešle devátý bit s hodnotou v logické nule. Pak je přenos ukončen a soustava uvedena do původního stavu. [2]



Obrázek 6: Komunikace pomocí I<sup>2</sup>C rozhraní.<sup>6</sup>

<sup>6</sup>Zdroj: [http://f1.hw.cz/design/i2c\\_pic/i2c01.gif](http://f1.hw.cz/design/i2c_pic/i2c01.gif)

## Kapitola 3

# Rozbor problematiky a návrh řešení

V této kapitole jsou podrobně rozebrány jednotlivé body zadání. Jsou uvedeny důvody pro zvolení daných nástrojů a metod. V druhé části je popsán návrh realizace projektu, jež se dá rozdělit do několika hlavních částí a pro každou z nich je uveden návrh realizace.

Pro bezdrátové vysílání hudby, je stále nejlepší volbou FM rádiové pásmo. Jde o nejpoužívanější metodu dnešní doby, jak přenášet hudební soubory. Běžné analogové vysílání je už ale přece jen zastaralé. Z důvodů kvalitnějšího přenosu a možnosti vysílat a přijímat různorodé informace se používá digitální vysílání. Jedním z požadavků zadání je i vysílání RDS informace. Proto je třeba zvolit vhodný FM vysílač, schopný tyto informace nejen vysílat, ale i za chodu měnit. Jak je patrné ze souhrnu existujících řešení (sekce 2.3), je nejlepší volbou sestavení vlastního vysílacího obvodu, nejlépe za použití čipu, zastávajícího vysílání. Toto řešení nejen poskytuje poměrně široké možnosti ovládání, ale je výhodné i z hlediska prostorového řešení. Obvod je možno koncipovat tak, aby byl připojen přímo na vstupně výstupní piny Raspberry bez dalších zbytečných propojovacích kabelů, nebo jiných součástí.

Bezdrátové ovládání i přenos souborů je bezpochyby nejlepší cesta k pohodlnosti použití přehrávače. Nutnost přesouvat se kvůli výběru skladby, nebo přepnutí na další, přímo k přehrávači je tak eliminována. Pro bezdrátové ovládání je doporučen mobilní telefon. V dnešní době téměř každý disponuje touto chytrou technologií, proto je na tyto platformy mířeno především. Ovládací aplikace s jednoduchým a snadno použitelným rozhraním, musí být schopna minimálně těchto funkcí:

- Změna RDS informace
- Změna vysílací frekvence FM
- Pauza v přehrávání
- Přehát další skladbu (právě přehrávanou přeskočit)
- Přetočit právě přehrávanou na začátek (spustit znovu)

Od dob, kdy přestaly telefony sloužit jen k přenosu hovorů a textových zpráv, se používají pro nejrůznější účely. Jedním z nich je i kapesní přehrávání hudby a tak není divu, že většina vlastníků podobných technologií nosí svou oblíbenou hudbu stále u sebe. Je proto vhodné přizpůsobit mobilním platformám i přenos hudebních souborů k přehrávání.



### 3.1 Využití linux platformy

Platformy s instalovanou distribucí linuxu jsou k použití velmi výhodné. A to nejen snadným vzdáleným přístupem, konfigurací, nebo použitím, ale mají i velkou uživatelskou základnu s dostatkem užitečných návodů a rad. Otevřený software u většiny nástrojů navíc dává možnosti vlastních úprav. Jednou z takových platform je **Raspberry Pi**. Jak uvádí [3], jde o levný a prostorově velice úsporný<sup>1</sup> počítač, postavený na jediné desce tištěného spoje, který je ale vybaven základními prvky pro běžné použití. Byl vyvíjen od roku 2006 jako výukový kit určený pro školy a děti, aby měli možnost vzdělávat se v programování a ovládání různých prvků a to za co nejdostupnější cenu. Je vhodný nejen pro běžné použití, nebo výuku a psaní programů, ale nabízí i možnost propojení s vnějším světem přímo skrze piny. To u běžných domácích počítačů, nebo konzolí není zvykem.

Vzhledem k tomu, že prototyp byl vytvořen na platformě Raspberry Pi 2 B, bude v popisu zmiňován právě tento typ. Některé zajímavé vlastnosti, nebo srovnání s dalšími typy bude uvedeno. Raspberry Pi platforma je osazena procesorem od firmy Broadcom<sup>2</sup>, BCM2836. Tento jednočipový systém je vybaven čtyřmi jádry typu ARM Cortex-A7, běžícími na frekvenci 900 MHz s 32bitovou šířkou slova a instrukční sadou ARMv7. Procesorové jednotky typu ARM jsou oblíbené pro jejich vysoký výpočetní výkon, při nízké spotřebě energie. Díky tomuto faktu jde dnes o jeden z nejpoužívanějších typů řídicích jednotek v mobilních zařízeních. Využití ale nalézají i v dalších odvětvích, jako jsou vestavěné systémy a prostory, kde není snadné provozovat spolehlivé chlazení. Díky nízkému příkonu nejsou tyto procesory tolik náchylné na teplotu. ARM jednotky se řídí filosofií RISC. Jejich instrukční sada tedy je sice omezená, zato velmi dobře optimalizovaná. Systém dále disponuje grafickým akcelerátorem Broadcom VideoCore IV. Ten je schopen skrze HDMI rozhraní zobrazovat i FullHD rozlišení<sup>3</sup>. A v neposlední řadě je součástí procesorového obvodu RAM paměť sdílená s grafickým čipem o kapacitě 1 GB. [16]

K napájení jednotky je zapotřebí USB 5 V napětí s dodávaným proudem alespoň 1,8 A<sup>4</sup>. Tato skutečnost podporuje možnost běhu i na baterii a to dostatečně dlouho. Pro potřeby přehrávače je předpokládána maximální doba běhu 12 hodin. Čehož je schopna dosáhnout i lehce nadprůměrná baterie.

Dále je Raspberry vybaveno několika USB porty. U těchto je nastavené omezení maximálního napájecího proudu 600 mA – 1,2 A. Pro naše potřeby, tedy napájení WiFi a Bluetooth adaptéru je to dostačující. Nejnovější přírůstek do rodiny Raspberry, tedy typ 3, je navíc již v základu vybaven vestavěným WiFi i Bluetooth.

Operačních systémů je dostupná celá řada.[3] Některé jsou připravené pro přehrávání multimédií, jako OpenELEC, nebo OSMC. Jiné poskytují oblíbené nástroje, jako Visual Studio při Windows 10 IoT Core. Avšak doporučovaný a nejvíce podporovaný je systém Raspbian<sup>5</sup>. Jde o optimalizovanou verzi Debian distribuce přímo pro Raspberry platformu. Je stále ve vývoji a nabízí velmi širokou škálu balíčků.

Na trhu se samozřejmě vyskytuje celá řada mikrokontrolérů a mikropočítačů, jež by mohli být vhodnými alternativami. Jedním z nich je BeagleBone Black, podobně koncipovaný počítač na jedné desce. Další jsou například NanoPC-T1, Radxa Rock2 Square, nebo ODROID-C1+. Každý ze zmíněných disponuje lepší či horší výbavou, v poměru k ceně.

<sup>1</sup>Velikostí kreditní karty (56 × 85 mm)

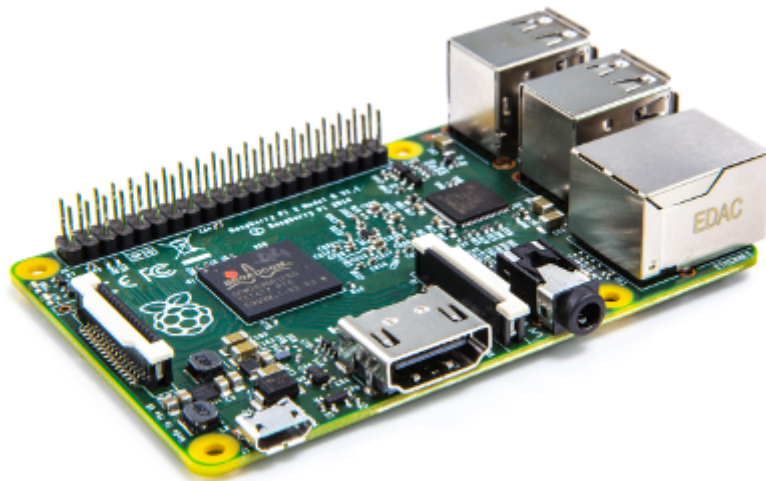
<sup>2</sup><http://www.broadcom.com>

<sup>3</sup>1920 × 1080 obrazových bodů.

<sup>4</sup>Záleží na připojených perifériích.

<sup>5</sup><https://www.raspbian.org/>

Raspberry Pi má ovšem zdaleka nejširší komunitu vývojářů a je snadno dostupné i v našich zemích.



Obrázek 7: Raspberry Pi 2 Model B v1.1 2014<sup>6</sup>

## 3.2 Navržené řešení

V této sekci je shrnut návrh realizace, rozdělený do čtyř podčástí. První z nich pojednává o návrhu tištěného obvodu a částí pro FM vysílač (podsekce 3.2.1). Následuje popis návrhu samotného přehrávače a jeho ovládání (podsekce 3.2.2), poté nahrávání souborů (podsekce 3.2.3) a nakonec návrh ovládací aplikace (podsekce 3.2.4).

### 3.2.1 Modul vysílače pro Raspberry Pi

Elektronický obvod zajišťující vysílání v pásmu FM je z valné většiny inspirován již existujícím podobným výrobkem od známé skupiny Adafruit<sup>7</sup>. Je navržen s ohledem na úsporu místa a pro komunikaci skrze I<sup>2</sup>C rozhraní (popsané v sekci 2.3.1), což je běžně a hojně využívaný způsob komunikace s perifériemi. Je vybaven čipem Si4713, který je schopen vysílat informace RDS/RBDS[7].

Úprava spočívá především v možnosti tento obvod připojit přímo na Raspberry skrze I/O piny. Jednak se tím odstraní nutnost zbytečných propojovacích kabelů na napájení, uzemnění, signál reset apod., zároveň je ale možné piny přivádět i stereo audio. Raspbian a jeho ovladače pro čipovou sadu BCM2836 používají na audio výstup PWM (pulzně šířkovou modulaci). Tento signál je vyveden na dva piny, reprezentující levý a pravý kanál stereo konektoru 3,5 JACK. Existují techniky jak přesměrovat tento signál na dva piny pro obecné použití. Tím lze signál o frekvenci 50 MHz dopravit na připojený modul. Vysoké frekvence je však nutné filtrovat a potřebný filtr je třeba vytvořit, jako součást modulu. Dále je možné audio dopravovat pomocí 3,5 JACK – 3,5 JACK propojovacím kabelem z výstupu Raspberry, na vstup vysílačeho obvodu. Poslední z možností je HDMI výstup na

<sup>6</sup>Zdroj: <http://core0.staticworld.net/images/article/2015/02/raspberry-pi-2-angle-100569133-orig.png>

<sup>7</sup><https://www.adafruit.com/>

Raspberry. Tato možnost je ale nejnáročnější, protože je potřeba z výstupního signálu extrahovat zvukovou stopu a tu poté dopravit na 3,5 JACK vstup obvodu. Nicméně výstupní kvalita zvuku je nejvyšší.

### 3.2.2 Přehrávač skladeb

Samotné jádro přehrávače je koncipováno tak, aby se po startu rozběhlo přehrávání, bez nutnosti vnějšího zásahu. Po připojení napájení a přidání skladby do fronty, se automaticky začne přehrávat.

Pro přehrávání souborů se jeví jako nejlepší volba VLC přehrávač. Jeho jádro tvoří libVLC knihovna, napsaná v jazyce C, která ke svému ovládní nabízí API, implementovanou v celé řadě jazyků. Další výhodou je velké množství podporovaných formátů<sup>8</sup> a to v základní verzi. Program pro obsluhu přehrávání tedy musí s touto knihovnou pracovat. Zároveň je ale nutná komunikace s vysílacím obvodem, resp. čipem. Vhodná volba je dvou-vodičové komunikační rozhraní I<sup>2</sup>C a i tuto funkcionalitu je nutné implementovat.

Program pro přehrávání tedy ustanovuje a spouští FM vysílání, zpracovává hudební soubory a přehrává je ve správném pořadí. Musí být ale také schopen reagovat na vnější zásahy, tedy změny funkčnosti od ovládací aplikace. Za běhu lze měnit nosnou frekvenci signálu, přepisovat RDS informace anebo přehrát další skladbu. Zároveň je schopen nepřetržitě fungovat i delší časový interval.

### 3.2.3 Management skladeb

Původní návrh nahrávání souborů pro přehrání počítal s bluetooth přenosem. Ovšem vzhledem k tomu, že bluetooth síť pro paralelní přenos dat od více uživatelů do jednoho uzlu není nativní funkcionalitou bluetooth zařízení, bylo by nutné vytvořit aplikaci, která by danou funkčnost implementovala. Což je úkol přinejmenším nesnadný, protože by bylo třeba pokrýt velké množství různých zařízení i operačních systémů.

Po zavržení bluetooth přenosu se zrodila myšlenka nahrávat soubory skrze wifi síť. Tedy přehrávač jako přístupový bod bezdrátové komunikace a standardní HTML stránka, zobrazovaná připojeným uživatelům. Počáteční verze počítala s osvědčeným a volně dostupným HTTP Apache serverem a PHP skripty pro obsluhu nahrávání. Díky radám vedoucího práce bylo ale toto rozhodnutí přehodnoceno a finální verze pracuje s HTTP serverem v jazyce python. A to hlavně díky „přiblížení se“ jádru přehrávače. Jednou z možností je využití služeb modulu Tornado, který efektivně obsluhuje i větší počet paralelních připojení.

Na straně klienta je zobrazována HTML stránka řízená primárně JavaScriptem. Vzhledem k faktu, že je cíleno především na mobilní platformy, bylo by na místě, použít některý z volně dostupných frameworků (aplikační rámec). Tím se docílí nejen přirozeného vzhledu, ale i ovládní. Vhodným kandidátem je komplexní framework<sup>7</sup>.

### 3.2.4 Mobilní aplikace

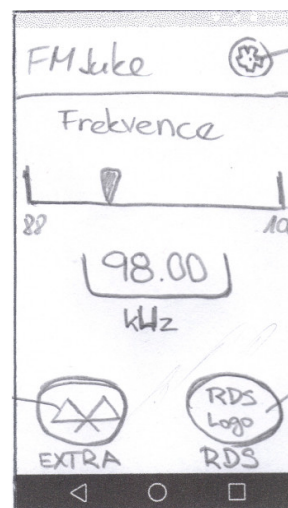
Pro ovládní přehrávání byl doporučen chytrý telefon. Stejně jako při transferu souborů (rozebrán v sekci 3.2.3) i zde původní návrh spoléhá na bluetooth technologii. Vzhledem k tomu že je potřeba připojení jen jednoho uživatele v čase a aplikace pro mobilní platformu musí být vytvořena, od komunikace pomocí bluetooth nebylo upuštěno. Díky dostupnosti vývojářských nástrojů a testovacích zařízení, je zvolena platforma Android pro návrh

<sup>8</sup>Seznam audio formátů: <http://www.videolan.org/vlc/features.php?cat=audio>

a vytvoření aplikace, která zajistí možnost bezdrátového a pohodlného ovládání přehrávání. Aplikace musí být schopná se připojit kdykoliv při běhu přehrávače. Díky tomu je možné aplikaci spouštět jen v případech potřeby a šetřit tak baterii mobilního zařízení, popřípadě se mohou v ovládání přehrávání jednotliví uživatelé měnit. Samozřejmě přehrávač musí na tuto situaci být připraven. S důrazem na jednoduchost a uživatelskou přívětivost aplikace, je návrh co nejvíce minimalistický. Rozhraní počítá se zobrazeným seznamem skladeb, včetně právě přehrávané a ovládacími prvky pro ovlivnění přehrávání. Poskytnuty musí být možnosti měnit RDS a nosnou frekvenci vysílání.



(a) Hlavní obrazovka jednoduché aplikace



(b) Obrazovka nastavení

Obrázek 8: Původní návrh mobilní aplikace

# Kapitola 4

## Realizace

Tato kapitola detailně rozebírá implementaci a tvorbu projektu. Podobně jako kapitola předchozí (kapitola 3.2), i realizace může být rozdělena na několik částí. První z nich je popis obvodu pro vysílání v pásmu FM (sekce 4.1), následuje implementace python programu zajišťujícího přehrávání (sekce 4.2). Další sekce je věnována HTTP serveru a nahrávání souborů (sekce 4.3) a poté je popsána android aplikace pro ovládání (sekce 4.4).

### 4.1 Vysílací obvod

Jak je zmíněno v návrhu (v sekci 3.2.1), je potřeba přivést z I/O pinů Raspberry napájení a řídicí signály. Schéma uvedené v příloze B popisuje použité součásti a spoje. Vysílací jednotkou je již zmiňovaný čip Si4713. Nyní budou uvedeny některé jeho důležité vstupy a výstupy. Na pinu č. 4 je výstup pro vysílaný signál, tedy na anténu. Pin č. 5 je vstupem pro signál RESET. Pokud je na tomto pinu logická nula, pak je vysílač ve stavu reset a nevysílá. Další v řadě, tedy pin označený číslem 6, slouží ke změně adresy I<sup>2</sup>C komunikačního protokolu. Pro naše potřeby je nastaven stále na logickou 0<sup>1</sup>. Následující dva piny v pořadí slouží právě pro komunikaci s čipem. Připojení označené jako SCL je hodinový signál pro komunikaci, SDA přenáší data. RCLK vstup slouží pro přívod referenčního taktu z oscilátoru. Piny číslo 10 a 11 jsou určeny k napájení. Vstup č. 15 má funkci vstupu pravého kanálu analogového stereo audio signálu. Podobně jako č. 16, jež zastává levý kanál. Pin GP03 je v daném případě využíván pro synchronizaci hodinového signálu. Následující dva piny, tedy číslo 18 a 19 jsou výstupní piny obecného použití, např. pro LED kontrolky. V projektu nevyužité. Součástky Q1 a Q2 jsou unipolární tranzistory, zde ve funkci pull-up rezistorů.

Součástka JP2 označuje 40-pinový konektor, přímo odpovídající Raspberry Pi 2. Vstupy číslo 12 a 33 slouží pro příchozí signál PWM. Ten zastupuje na Raspberry audio výstup a má vysokou frekvenci. Ta je potřeba filtrovat na úroveň slyšitelného analogového zvuku. Raspberry generuje PWM signál s frekvencí asi 50MHz. Výpočet účinku filtru je následující.

$$f_0 = \frac{1}{2\pi RC} \quad (1)$$

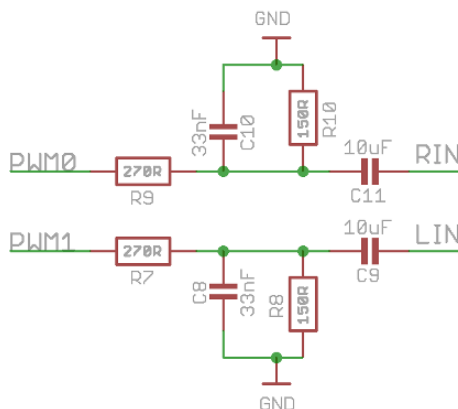
$$f_0 = \frac{1}{2 \times \pi \times 270 \times 33 \times 10^{-9}} \quad (2)$$

$$f_0 = 17862,5076\text{Hz} \quad (3)$$

---

<sup>1</sup>Adresa: 0x11

V rovnici 1  $f_0$  zastupuje mezní frekvenci,  $R$  je hodnota odporu rezistoru a  $C$  kapacita kondenzátoru. Výsledek (rovnice 3) tedy téměř dosahuje hodnoty 20kHz, což je běžně udávaná hranice lidem slyšitelného zvuku.



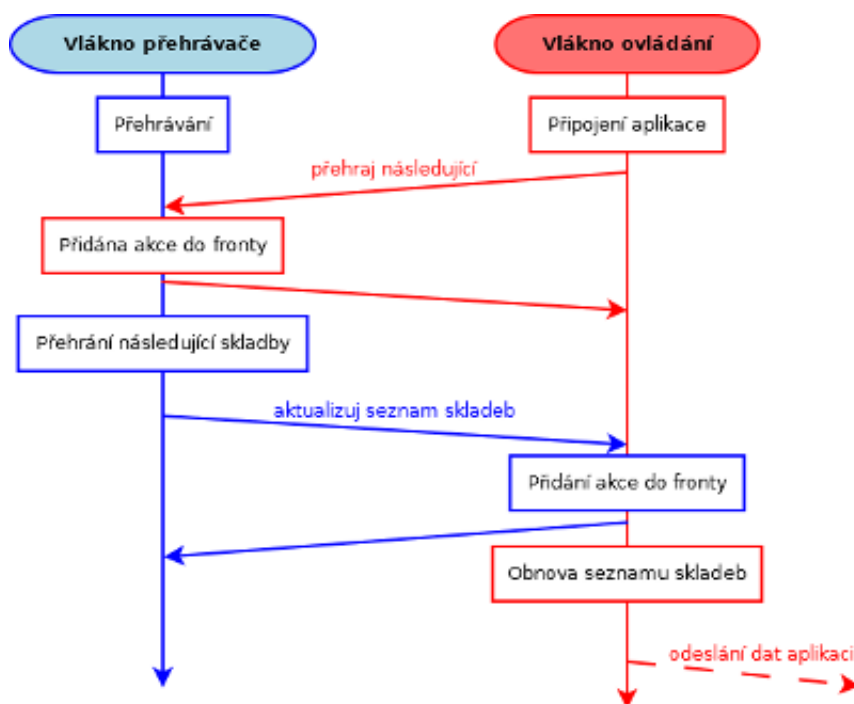
Obrázek 9: Filtr vysokých frekvencí. PWM0 reprezentuje pravý zvukový kanál stereo audia. R9 spolu s C10 tvoří dolní propust, pro filtrování v rámci slyšitelné frekvence. Díky R10 je sníženo napětí a konečně C11 kapacitor filtruje stejnosměrné napětí. Výstup RIN je připojen na čip Si4713. Analogicky pro levý kanál (PWM1).

Ovládání modulu z programu je zajišťováno python skriptem, který využívá již implementované třídy `Adafruit_Si4713` a `Adafruit_I2C`. Ty byly vytvořeny právě pro ovládání adafruit modulu s použitím Si4713. Zahrnují nastavení pinů a komunikaci skrze I<sup>2</sup>C, pro ovládání vysílacího čipu. Třída importovaná přehrávačem `TransmitterFM` ve svém průběhu nejprve nastaví defaultní hodnoty, poté spustí vysílání (funkcí `begin()`). Po startu jsou vysílači nastaveny i další hodnoty, jako je vysílací frekvence, nebo defaultní RDS informace. Třída dále implementuje dvě metody. První `set_rds(rds)` ověří správnou délku RDS řetězce a poté vysílacímu čipu tento řetězec poskytne pro jeho vysílání (pomocí `setRDSbuffer(rds)`). Druhá metoda `set_freq(freq)`, jak název napovídá, nastavuje nosnou vysílací frekvenci. Frekvence je nejprve upravena dle mezí obvodu, tedy 76 – 108 MHz, a voláním `tuneFM(freq)` uvedena v platnost. Poslední dvě zmiňované metody jsou volány z jádra přehrávače. Asynchronní volání přichází jako příkaz z ovládací aplikace, na změnu nosné frekvence, či RDS.

## 4.2 Jádro přehrávače

Hlavní jádro přehrávače, je tvořeno v jazyce python. Pro spuštění přehrávání bez externího zásahu, je spouštěn shellovský skript `startUp.sh` ihned po automatickém přihlášení uživatele. Na Raspberry s operačním systémem raspbian toho lze dosáhnout editací souboru `.bashrc` v domácím adresáři. Skript nejprve pomocí příkazu `hciconfig hci0 up piscan` spustí a nastaví bluetooth zařízení do režimu `discoverable`, což umožní spárování s dalšími zařízeními. Komunikace s ovládací mobilní aplikací je umožněna právě jen spárovaným zařízením. Dalším krokem je nastavení hlasitosti zvukového výstupu (nástrojem `amixer`). Poté jsou smazány všechny případné hudební soubory z minulé relace a logovací soubor. Nakonec jsou jako samostatné procesy spuštěny skripty `PlayerServer.py` (sekce 4.3) a `coreFMJuke.py`

Poslední zmiňovaný soubor `coreFMJuke.py` zajišťuje vytvoření a spuštění instancí tříd `TransmitterFM` (popsané v sekci 4.1) pro zahájení rádiového vysílání FM, `PlayerThr` (sekce 4.2.1), jakožto samostatné vlákno řízení přehrávání a `Arbiter` (sekce 4.2.2), jako vlákno obsluhující bluetooth připojení a ovládání uživatelem. Jak bylo popsáno výše, hlavní řízení mají na starosti python vlákna z modulu `threadings`. Jedno tedy pro řízení přehrávání, druhé pro obsluhu ovládání. Tyto dvě vlákna se vzájemně ovlivňují. Jejich nutná komunikace mezi sebou je umožněna rozhraním, které definují některé volané metody. Aby nedošlo k předání řízení, disponuje každé z vláken FIFO frontou akcí. Potom volané metody z rozhraní jen přidají požadovanou akci do fronty a vrací se řízení volajícímu vláknu. Metody rozhraní je tedy možné volat zcela asynchronně. Ovšem instance třídy vlákna přehrávače, musí uchovávat referenci na instanci arbitra a naopak.



Obrázek 10: Ukázka mezivláknové komunikace. Šipky naznačují tok řízení vlákna. Modrá barva reprezentuje vlákno přehrávání hudby, červená vlákno ovládací aplikace. Metody jsou volány tak, aby nebylo předáno řízení.

Mimo výše uvedené, implementace obsahuje také modul `constants.py`. Ten je sdílen vícero částmi programu a obsahuje UUID identifikátor bluetooth služby, cestu k adresáři pro nahrávání souborů anebo funkci pro logování. Funkce `get_track()` slouží pro vyhledání nejstaršího přidaného souboru, podle data poslední modifikace. Tím je zaručeno, že se budou skladby přehrávat ve správném pořadí. Pokud není nalezen žádný soubor, funkce vrací `False`. Podobný způsob řazení využívá i funkce `sorted_list(reverse)`, která ovšem vrací seznam všech souborů. (Popřípadě prázdný seznam.) Parametr `reverse`, typu boolean, určuje směr řazení.

## 4.2.1 Vlákno přehrávání hudby

Třída `PlayerThr` implementovaná v souboru `VlcPlayer.py` zodpovídá za chod přehrávání požadovaných souborů. Nahraný modul `vlc.py` obsahuje implementaci API pro obsluhu `libVLC` knihovny, použité pro přehrávání. Po startu vlákna se nastavují defaultní hodnoty. Je vytvořena instance VLC přehrávače, RDS informace nastavena boolean přepínačem `rds_song_like` do stavu, kdy odpovídá názvu právě přehrávané skladby. Poté je zahájeno pětisekundové čekání, což z testů vyplynulo jako dostatečná doba, pro inicializaci a spuštění vysílání FM. Následným voláním metody `play()` je spuštěno přehrávání.

Funkce `play()`, jako první krok, zadá vláknu ovládání požadavek na obnovení fronty skladeb. Poté sama vybere nejstarší skladbu (`get_track()`). Pokud je některá k dostání, nastaví přehrávač a danou skladbu mu přiřadí pomocí API volání funkce `set_mrl(track)`. Další API funkcí `play()` se spustí přehrávání souboru. Jelikož přehrávání média má na starosti samostatně běžící proces `vlc`, pokračuje se další fází, což je nastavení RDS. Avšak to jen v případě, že má být totožné s názvem přehrávaného souboru. V případě, že žádný soubor ve frontě nečeká, je nastaven příznak `no_track` a RDS nastaveno do původního stavu. Následuje hlavní smyčka (funkce `main_loop()`).

Nastavení přehrávače řeší metoda `setup_player()`. Ta pomocí volání API funkcí vytvoří instanci `media_player` třídy, nastaví jí hlasitost na 120% a přiřadí jí metodu, jež bude asynchronně volána po skončení skladby (`cb_song_end()`). Tato obsahuje pouze přidání akce „next“ do fronty.

Pro zajištění stálého chodu, implementuje vlákno nekonečný cyklus typu `while`. V průběhu cyklu se kontroluje dříve zmíněná fronta akcí, implementována jako pole řetězců. Smyčka je popsána obrázkem 11. Každá požadovaná akce je zpracována voláním příslušné metody. Pokud je nastaven příznak `no_track`, je testována přítomnost čekajícího souboru. Každý průchod smyčkou uspí vlákno na jednu sekundu pomocí funkce `time.sleep()`. To z důvodu, aby procesorový čas byl přiřazován častěji i ostatním vláknům a nebyl naplno vytížen i když se nepřehrává.

Metoda `rew()` má za úkol přehrát skladbu od začátku. Dosáhne toho tak, že nad instancí přehrávače zavolá API funkci `set_time()` a jako parametr předá hodnotu nula. Podobně metoda `pause()` pouze spouští API funkci, která přepne stav „pozastaveno“. Takže z přehrávání se přejde do pauzy. Po dalším volání této metody přehrávání pokračuje.

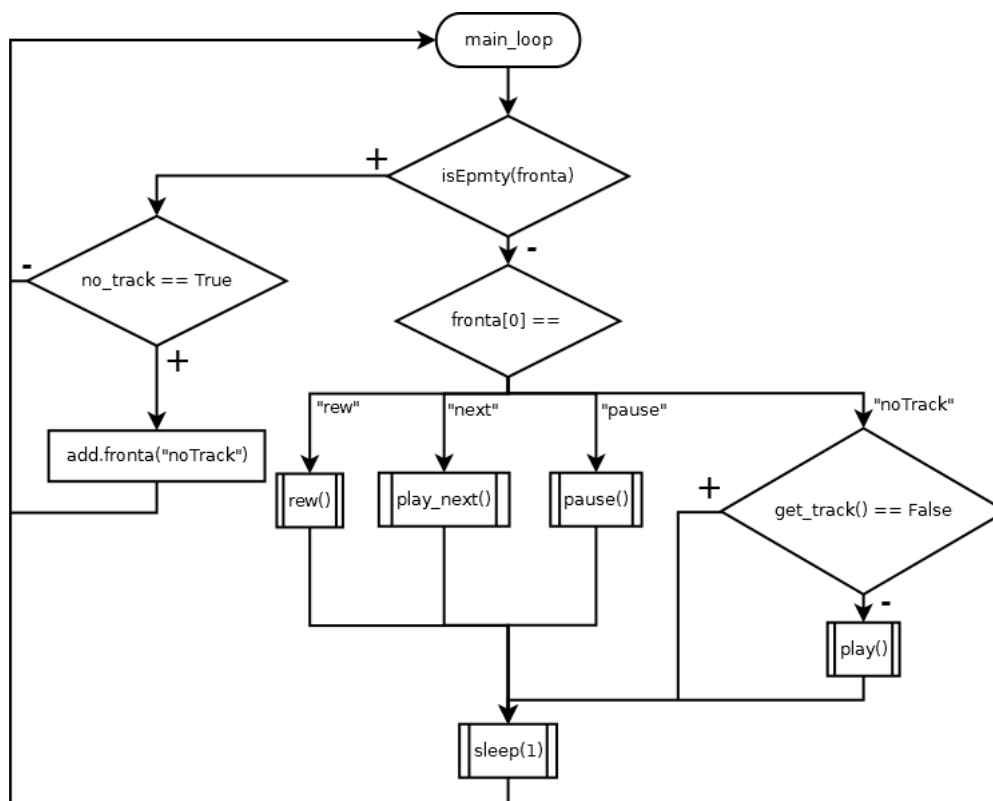
`play_next()` zajišťuje spuštění následující skladby ve frontě. Nejprve ukončí přehrávání (metoda `stop_play()`), poté vymaže soubor odpovídající zastavené skladbě a volá metodu `play()`. Ta opět vybere čekající soubor, inicializuje a spustí přehrávání a tak dále.

Metody definující komunikační rozhraní jsou ve zdrojovém kódu rozeznatelné podle prefixu `ext_`. Jsou to `ext_set_arbiter(Arbiter)`, jež uloží referenci na vlákno obsluhující ovládání, `ext_play_next()`, `ext_pause()`, `ext_rew()` a `ext_rds_song()`. Poslední zmiňovaná nastavuje příznak, který zajistí podobu RDS informace totožnou s názvem skladby, jež je právě přehrávána. Ostatní odpovídají názvům a přidávají pouze akci do fronty.

## 4.2.2 Vlákno pro ovládání

Třída `Arbiter`, je analogicky k přehrávači (sekce 4.2.1), definována jako vlákno. Také samozřejmě disponuje vlastní FIFO frontou akcí. Je odpovědná za připojení uživatelské ovládací aplikace a provedení příchozích požadavků. Komunikace s aplikací probíhá pomocí bluetooth technologií, s architekturou server–klient. Server zastupuje právě toto vlákno, klient je naopak ovládací aplikace. Pro bluetooth implementaci je využito modulu `PyBluez`, což velmi zjednodušuje práci se zdroji bluetooth.





Obrázek 11: Diagram naznačující průběh hlavní smyčky přehrávače. `fronta` zastupuje frontu akcí. `get_track()` vrací název přehrávané skladby, nebo `False`.

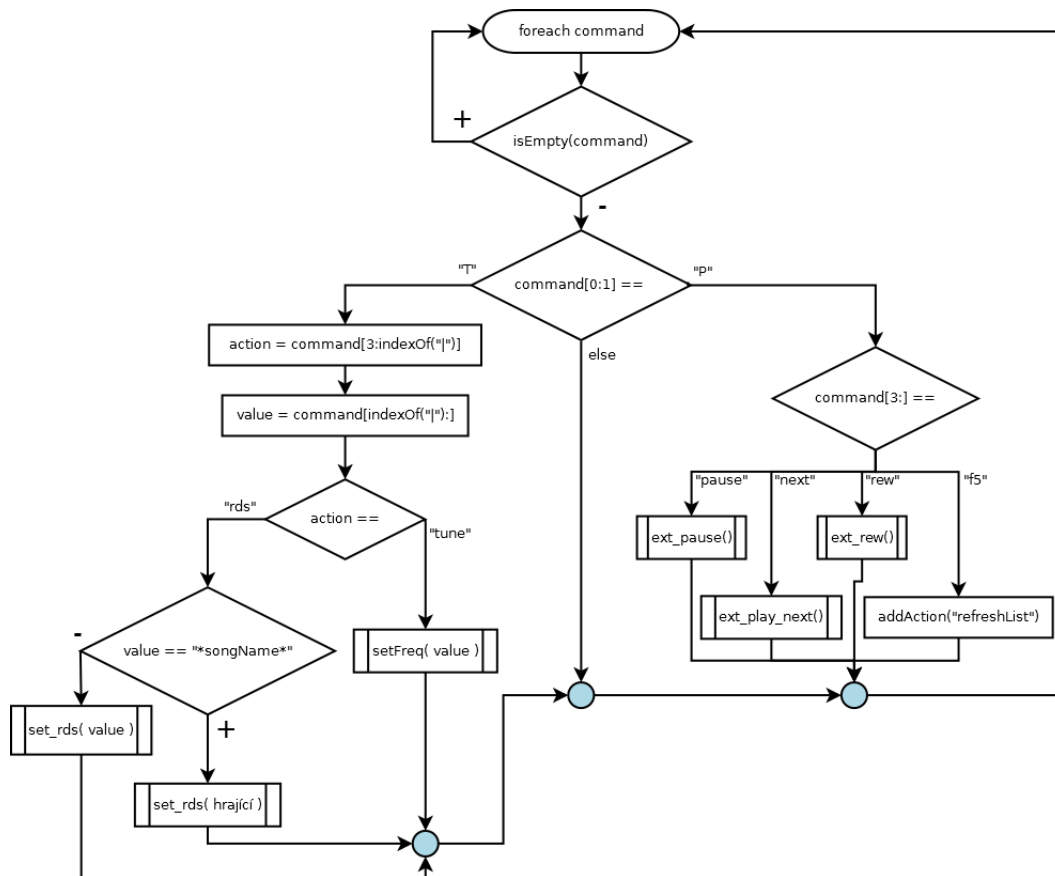
Po startu vlákna jsou nastaveny základní hodnoty, jako reference na instance tříd reprezentující přehrávač i vysílač. Dále je založen RFCOMM komunikační kanál na server socketu. Socket je pak nastaven na očekávání právě jednoho připojení. Toto čekání je nastaveno jako blokující. Umožněno je to faktem, že vlákno ovládání nemá žádnou další funkcionalitu a pokud se uživatel nepřipojí, nemusí zatěžovat procesor. Přehrávání tedy běží i bez připojeného ovládacího prvku.

Jakmile je navázáno spojení, je nový komunikační socket nastaven na neblokující (funkcí `setblocking(False)`). Aktivní instance třídy `arbitr` je poskytnuta vláknu přehrávače pro pozdější komunikaci a pokračuje se hlavní smyčkou.

Ta je opět typu `while` a v jejím průběhu se kontroluje `fronta` akcí. Pokud není prázdná, akce se provede voláním příslušné metody. Dílčím krokem v průchodu je však ještě neblokující čtení z připojeného socketu. Díky neblokujícímu způsobu se nečeká na data, ale je možné v práci pokračovat i když není co číst. Pokud je vstupní buffer prázdný, je generována výjimka typu `bluetooth.btcommon.BluetoothError`. Po jejím zachycení je vlákno uspano na jednu sekundu<sup>2</sup>, než je spuštěn další cyklus. Jiná výjimka, `IOError`, je generována v případě, že došlo ke ztrátě spojení. V tomto případě je vlákno restartováno, tedy uvedeno do stavu ihned po inicializaci metodou `run()`. Vláknu přehrávače je taktéž odebrána reference na tuto instanci. Tak je zajištěno, že bude znovu blokujícím způsobem očekávat další připojení. Tato technika umožňuje uživateli se odpojit a znovu připojit bez nutnosti přerušování běhu přehrávače.

<sup>2</sup>Funkce `time.sleep(1)`

Po úspěšném přečtení dat ze socketu, jsou data rozdělena do jednotlivých příchozích požadavků. Jako oddělovač slouží znak „&“, který je v mobilní aplikaci přidáván za každý požadavek. Toto pokryje případy, kdy jsou příkazy odesílány rychleji, než jsou čteny. Zpracování takto vytvořené fronty příkazů je názorně uvedeno na obrázku 12.



Obrázek 12: Diagram popisující řešení příchozího příkazu ze strany ovládací aplikace. Hodnoty v hranatých závorkách určují indexové rozsahy, oddělené dvojtečkou. Pokud index není uveden, jde o krajní mez ([ od : do ]). Příklad příchozího příkazu „T::tune|108,0“

Pokud je volána metoda `send_list(refresh)`, je odeslán na mobilní aplikaci seznam čekajících skladeb k přehrání. Získán je pomocí dříve zmiňované (v sekci 4.2) funkce `sorted_list(reverse)`. Pokud je vrácený seznam prázdný, odešle se pouze jeden záznam s tímto vyrozuměním. Na základě předaného parametru `refresh`, je buď přepsán stávající seznam kompletně, tedy nahrazeny všechny záznamy, anebo je pouze doplněn. Doplnění seznamu je umožněno proměnnou `list_cnt`, jež uchovává hodnotu velikosti seznamu odeslaného naposledy. Defaultně je nastavena na 1. Po každém doplnění skladeb, odesláním seznamu, se hodnota aktualizuje na délku fronty souborů. Odesílaný seznam skladeb má upravený rozsah od indexu daného `list_cnt` do konce. Tím je dosaženo správného doplnění nově přidávaných souborů, od poslední aktualizace.

Díky neblokujícímu čtení ze socketu, lze asynchronně odesílat data. K tomu slouží metoda `send_data(data)`, jež pouze volá funkci `send(data)` nad bluetooth socketem.

I vlákno ovládací aplikace má určeno externí rozhraní metodou s prefixem `ext_` v názvu. V daném stádiu vývoje je to pouze jediná a to `ext_send_list()`. Ta je volána z jiného vlákna a aby nebylo předáno řízení, pouze přidává akci do fronty.

## 4.3 HTTP server a HTML stránka

Architektura klient–server je použita i pro webové rozhraní, vytvořené k přijímání souborů. Díky zvolenému jazyku python, lze využít některé již implementované funkce. V roli HTTP serveru vystupuje přehrávač. Tedy soubor `PlayerServer.py`, který implementuje neblokující server pomocí Tornado, ze stejnojmenného modulu. V hlavní nekonečné smyčce server kontroluje přístupy na definované URI. Pro každou z nich poté určí obslužnou rutinu (handler), jež je spuštěna po přístupu na danou URI adresu klientem.

V projektu jsou pouze tři definované adresy. První z nich je adresa domény, kterou obsluhuje třída `MainHandler`. Ta pomocí šablony generuje hlavní stránku v HTML a funkcí `render()` ji dopravuje na klientskou stranu. Díky technice šablon a již několikrát zmínovaným funkcím `sorted_list()` a `get_track()` (viz popis v kapitole 4.2) lze jednoduše generovat stránku, s frontou skladeb i právě přehrávanou.

Třída `ListUpdatesHandler` ošetřující obnovu seznamu skladeb, generuje šablonu ze souboru `list.html`. Ta jako parametr přijímá frontu souborů a výsledek odesílá na klienta. Adresa již je tato třída přiřazena („`/list/updates`“), je volána klientskou stranou na pozadí.

Poslední obslužnou třídou je `TrackUploadHandler`. Ta je také volána AJAX požadavkem od klienta a slouží k nahrání a uložení hudebního souboru, určeného k přehrání. Nahráný soubor je nejprve kontrolován na typ MIME, pokud není označen jako audio, je vrácena chyba a soubor není uložen. V případě, že je nahrán soubor, se stejným názvem, jako je už některý ve frontě, je implicitně tento soubor přepsán. Tím se brání zahlcení fronty přehrávání stejnou skladbou<sup>3</sup>. Může ale nastat i situace, že soubor s totožným názvem je již přehráván. Tady je přepsání obsahu nežádoucí a tato skutečnost se ošetřuje tak, že k názvu nahrávaného souboru se přidá řetězec „`0_`“. Při úspěšném pokusu o nahrání souboru je tento zapsán a uložen do interního úložiště přehrávače. Výsledek operace je klientovi odeslán jako JSON hodnota proměnné `status`. V případě úspěchu `success`, jinak `error`.

Server běží v přehrávači jako vlastní proces, díky čemuž jsou přesuny dat poměrně rychlé i při více paralelně běžících připojeních. Více o rychlostech v kapitole 5.

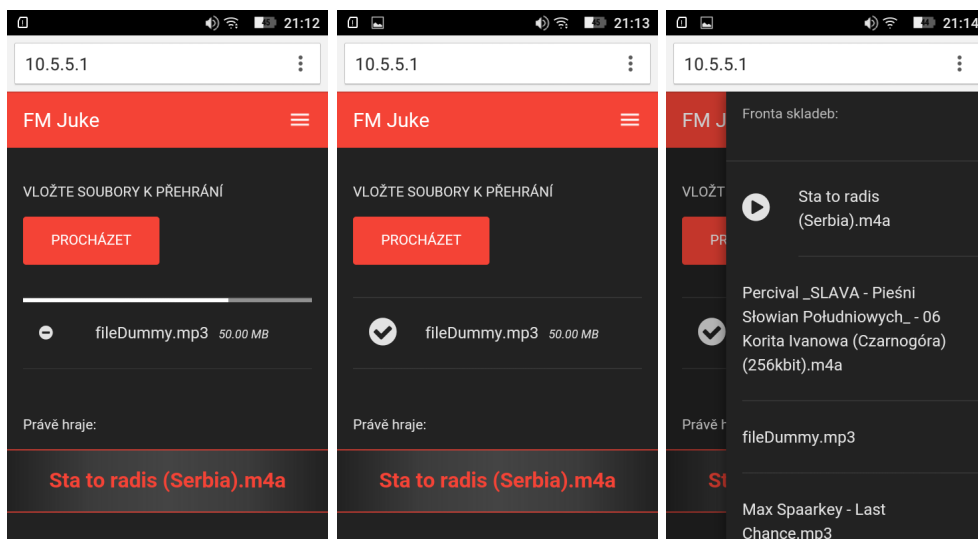
### 4.3.1 Klient

Klientská stránka pro nahrávání (upload) souborů je řízena především javascriptem. Využívá přitom služeb aplikačního rámce `framework7`. Ten nabízí pohodlné použití HTML, JS a CSS nástrojů pro tvorbu internetových stránek, přizpůsobených mobilním platformám. A to jak vizualizací, tak ovládáním. Hlavní skript `fmJuke.js` potom inicializuje tento framework a implementuje funkci `getList()`. Ta již podle názvu získává a obnovuje seznam čekajících skladeb. Pomocí AJAX požadavku na adresu „`/list/updates`“ obdrží aktuální frontu souborů. Pokud fronta není prázdná, zkontroluje se vzorek (10 znaků) s již prezentovaným seznamem na stránce a při odlišnostech je seznam aktualizován. Dále se ověřuje jméno přehrávaného souboru. Pokud se liší, je nejen aktualizován element, jež jej zobrazuje, ale opět i celý seznam. Tato funkce je volána každé tři sekundy, což je dostatečně krátký časový interval pro udržování aktuálního stavu. Díky použití AJAX techniky není uživatel rušen obnovením stránky. Vše se děje na pozadí.

I nahrávání souborů využívá metodik odesílání na pozadí. Samotný upload souborů je řešen skriptem `upload.js`, ve spolupráci s běžným HTML5 formulářem. Po přidání souboru je spuštěna funkce `upload()`, která vytvoří a do existující stránky přidá nový element,

<sup>3</sup>Může nastat pouze v případě, že se o to pokusí uživatel/é

zobrazující průběh nahrávání, název souboru, jeho velikost a prostředek pro zrušení procesu nahrávání. Díky vytváření nových elementů je možné posílat i několik souborů paralelně. To se ovšem odráží na rychlosti spojení.



Obrázek 13: Ukázka webové stránky pro upload souborů.

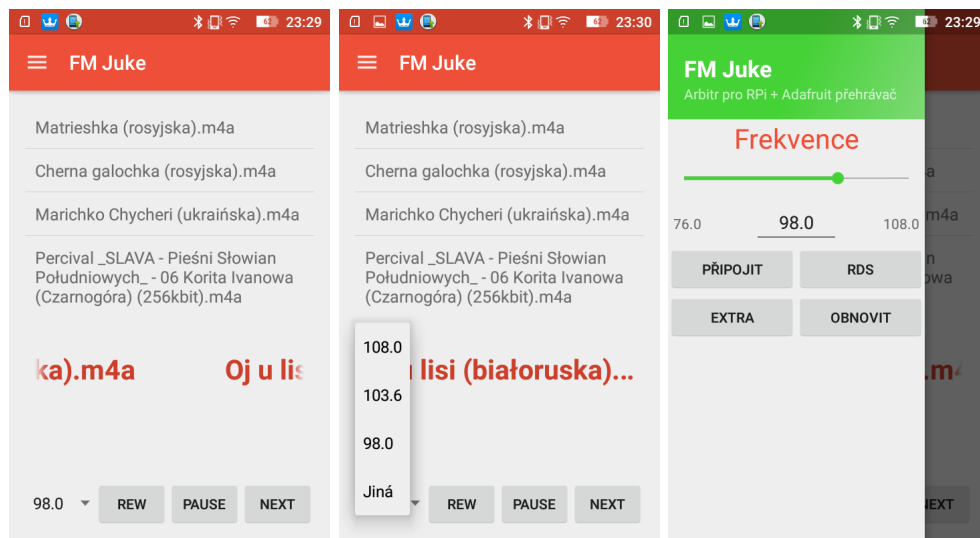
## 4.4 Android aplikace

Jak je uvedeno v kapitole 3.2.4, je ovládací aplikace pro chytrý telefon vyvinuta na platformě Android. Aplikace komunikuje s přehrávačem výhradně prostřednictvím vlákna Arbiter (popsaného v sekci 4.2.2), za použití technologie bluetooth. Detailnější náhled na odesílané příkazy je popsán obrázkem 12. S ohledem na co nejjednodušší ovládání a minimalismus, je aplikace tvořena pouze jednou jedinou aktivitou<sup>4</sup>, která ale obsahuje vše důležité. Prostředí aplikace se skládá ze dvou hlavních částí, tedy hlavní obrazovky a výsuvného panelu s nastavením. Základní obrazovka obsahuje název hrající skladby, frontu následujících skladeb a ovládání přehrávání. Navíc je zde přidáno rozbalovací menu, pro snadnou změnu vysílací frekvence, bez nutnosti přechodu do nastavení. Nastavovací panel se vysouvá z levé strany. Poskytuje nastavení frekvence posuvníkem, pro rychlou a snadnou změnu, nebo možnost zadání přesné vysílací frekvence. Dále pak sadu akčních tlačítek, jejichž názvy, resp. popisy, odpovídají funkcím jež zastupují. PŘIPOJIT se pokusí znovu navázat bluetooth spojení s přehrávačem, RDS zobrazí panel pro zadání RDS informací. Jedna z voleb je i nastavení RDS dle právě hrající skladby. EXTRA v daném stavu neplní žádnou funkci a konečně OBNOVIT kompletně přepíše a aktualizuje seznam souborů čekajících na přehrávání.

### 4.4.1 Třída definující uživatelské rozhraní

Samotný zdrojový kód aplikace obsahuje řadu volání a přiřazení hodnot, které slouží k propojení metod a proměnných s prvky už. rozhraní. Definice hlavní třídy `PlayerActivity` defaultně nastavuje hodnoty rozsahu vysílací frekvence, dané vysílacím obvodem a její

<sup>4</sup>Hlavní třídou obsluhující uživatelské rozhraní.



Obrázek 14: Ukázka aplikace

počáteční hodnotu. Konstanta `mAdress` obsahuje MAC adresu bluetooth zařízení serveru, tedy Raspberry. Ta je jedinečná pro každé zařízení a je využívána pro navázání spojení. Při spuštění aplikace je spuštěna metoda `onCreate()`. V průběhu funkce se naplní prvky UI základními hodnotami a přiřadí se jim vázané funkce (callback). Nastaví se bluetooth a ověří se jeho dostupnost. Pokud není bluetooth součástí zařízení, aplikace je ukončena. Následuje metoda `onStart` ve které je ověřeno zapnutí BT, popřípadě je uživateli zobrazen dotaz na povolení bluetooth komunikace. Při nepovolení je aplikace opět ukončena. V opačném případě se pokusí spojit se serverem.

Funkce `getFreqFromProgress(Int)` a `getProgressFromFreq(Double)`, slouží k přepočtu vysílací frekvence na hodnotu posuvníku a opačně. Posuvník má nastavené hodnoty od nuly po  $(\text{maximálníFrekvence} - \text{minimální frekvence}) \times 10$ . Nastavený krok je roven jedné celé. Tímto je zajištěna přesnost posuvníku při změně hodnoty na jedno desetinné číslo. Při přepočtu se samozřejmě hodnota posuvníku dělí, popřípadě hodnota frekvence násobí deseti.

Rolovací menu na hlavní obrazovce je naplněno maximálně třemi hodnotami posledních volených frekvencí a volbou „Jiná“. Při zvolení této možnosti je vyvoláno dialogové okno, které umožňuje zadání nové, uživatelem stanovené frekvence, která je nejen odeslána pro přeladění vysílače, ale je také uložena do rozbalovacího menu. V případě že jsou uloženy více než tři hodnoty, je ta nejstarší z nich odstraněna. Tak se docílí nabídky tří posledních zadaných hodnot vysílací frekvence. Při změně frekvence a to jakýmkoliv způsobem, je volána metoda `tuneAll(freq)`. Ta nejprve zaokrouhlí předanou hodnotu frekvence na jedno desetinné číslo, poté ji upraví takovým způsobem, aby odpovídala vysílačem stanovenému rozsahu. V dalším kroku je nová hodnota odeslána přehrávači a zároveň je prezentována uživateli.

#### 4.4.2 Třída pro komunikaci skrze bluetooth

Bluetooth komunikaci zajišťuje druhá definovaná třída `BluetoothService`. Tato třída obsahuje konstantu `UUID`, což je 128-bitový jedinečný identifikátor služby. Pro potřeby komunikace zařízení – zařízení byl vygenerován náhodně. Identifikátor slouží k nalezení okol-

ních zařízení a možnosti identifikace jejich služeb. Třída také disponuje proměnnou určující momentální stav připojení. (`STATE_NONE`, `STATE_CONNECTING`, `STATE_CONNECTED`). Zároveň implementuje dvě další třídy, definované jako vlákna.

První z nich, `ConnectThread`, při svém průběhu vytvoří socket za použití protokolu `RFCOMM`, což je běžně používaný protokol pro komunikaci a přenášení binárních dat. Dále se pokusí připojit k serveru. Při neúspěšném pokusu o připojení je navrácen původní stav vlákna pomocí metody `connectionFailed()`. Naopak při úspěchu, toto vlákno zaniká a spouští druhé vlákno.

`ConnectedThread` nejprve z připojeného socketu zajistí vstupní a výstupní streamy dat a poté ve svém běhu v nekonečné smyčce blokujícím způsobem čte ze vstupního streamu. Definuje také metodu pro zápis na výstupní proud, která je ale volána asynchronně a to tak, že nadřazenou třídou definovaná funkce `write(byte[])`, nejprve ověří stav připojení a pokud se potvrdí jeho platnost, zkopíruje instanci připojeného vlákna `mConnectedThread`. Pomocí této kopie poté volá metodu pro zápis na datový stream. Díky metodě `start()`, která ukončí běžící vlákna a nastaví stav na počáteční, je možné běžící aplikaci i po ztrátě spojení znovu připojit.

Komunikace a předávání dat mezi hlavními třídami zajišťuje instance systémové třídy `android.os.Handler`. Přetížením metody `handleMessage(Message)` dostaneme správné reakce na příchozí příkazy. Ať jde o aktualizaci seznamu skladeb, anebo upozornění pro uživatele vypsané na obrazovku. Aktualizace fronty skladeb je provedena algoritmem, který kontroluje první příchozí záznam, jež reprezentuje přehrávanou skladbu. Pokud první znak záznamu je roven „R“, pak je stávající seznam zcela vymazán a znovu naplněn. Tedy kompletně obnoven. Jiný případ nastane, pokud první znak rovná se „T“. Potom je seznam pouze doplněn o nově přidávané skladby a zároveň, pokud se zobrazený název přehrávané skladby liší od první příchozí, pak je element reprezentující přehrávaný soubor aktualizován na novou hodnotu.

Odchozí požadavky jsou posílány metodou `sendMessage(String message)`, která zkontroluje připojení, přidá za příkaz oddělovací znak (kvůli frontě na bufferu) a volá výše zmíněnou funkci `write(byte[])`.

## Kapitola 5

# Vyhodnocení parametrů navrženého řešení

Kapitola popisuje vytvořený prototyp pro testování (sekce 5.1) a uvádí provedená měření rychlosti datových přenosů (sekce 5.2). Jednou z velkých nevýhod vyplývajících z testů, je nemožnost připojení uživatelů k internetu, pokud jsou napojeni na síť přehrávače.

### 5.1 Prototyp

Pro účely testování a také jako jeden z bodů zadání, byl vytvořen a otestován prototyp. Následuje výčet součástí, použitých k jeho realizaci.

- Raspberry Pi 2 Model B V1.1 2014
- OS – Raspbian GNU/Linux 8 (jessie)
- Kernel – Linux raspberrypi 4.1.19-v7+ #858 SMP
- FM Transmitter – Adafruit Si4713 FM RDS Radio
- WiFi – TP-Link WN725N v2
- Bluetooth - TRUST Bluetooth 4.0 USB adapter
- 2 × rezistor 270Ω
- 2 × rezistor 150Ω
- 2 × kondenzátor 33nF
- 2 × kondenzátor 10μF
- Nepájivé pole – 400 bodů
- Propojovací kabely

Dále pro testování a vývoj android aplikace sloužil telefon Lenovo A1000, vybaven Androidem 5.0 Lollipop, s jádrem 3.10.64jenkins.

## 5.2 Rychlost přenosu

Wifi připojení přes USB modul je poměrně stabilní a přenosové rychlosti dosahují v průměru 3,16 MB/s. Jakmile je ovšem přenášeno paralelně více souborů od více uživatelů, rychlost se rapidně snižuje. Pro měření rychlostí přenosu byl využit nástroj netcat.

Aktivních připojení	Průměrná rychlost
1	3,16 MB/s
3	858,8 kB/s
5	571,3 kB/s

Tabulka 5.1: Změny rychlosti přenosu WiFi při paralelních připojeních. Naměřená rychlost je udávána pro jednoho uživatele.

Při úvaze že průměrný hudební soubor, má velikost 10 MB, je možné k přehrávání teoreticky odeslat až 18 souborů za jedinou minutu. To ale samozřejmě jen v ideálních podmínkách. Záleží na síle rušení bezdrátového signálu i na vzdálenosti, překážkách apod.

## 5.3 Kvalita zvuku

Jak bylo uvedeno v kapitole 3.2, je možné zvukový signál do modulu pro vysílání přivést vícero způsoby. Prvním z nich je přesměrování PWM výstupu na piny obecného použití na Raspberry a filtrovat vysoké frekvence na zvukový výstup. Tato metoda je výhodná díky tomu, že nejsou zapotřebí žádné propojovací kabely, nebo pomůcky. Také má čipová sada na Raspberry podporu hardwarové tvorby PWM, proto nezatěžuje procesor. Nevýhodou však je, že navržený filtrovací obvod (obrázek 9), není příliš kvalitní a tak i zvukový výstup není nikterak přesvědčivý.

Lepší metodou s ohledem na kvalitu zvuku je použití krátkého propojovacího kabelu typu 3,5 JACK – 3,5 JACK. Připojeného na výstup Raspberry a vstup vysílacího obvodu. Vzhledem k faktu, že FM vysílání zvukový signál opět trochu znehodnotí, je asi nejlepší volbou právě tato možnost.

Nejkvalitnějšího zvuku je možno dosáhnout pomocí HDMI výstupu. Hlavní nevýhodou tohoto přístupu je ovšem potřeba konvertoru HDMI – 3,5 JACK.



## Kapitola 6

# Závěr

Cílem projektu bylo navrhnout a vytvořit FM vysílač postavený na platformě Raspberry Pi. Vysílání hudby mělo být ovládáno bezdrátově, stejně jako nahrávání dat.

Bylo třeba navrhnout a sestavit vhodný vysílač v pásmu FM. Vyvinout jeho obsluhu a přehrávání hudby. Vytvořit bezdrátový ovládací nástroj pro chytrý telefon a rozhraní pro nahrávání souborů, také bezdrátovou technologií.

Podařilo se vytvořit funkční systém, přehrávající hudbu pomocí python programu na Raspberry. Obvod zajišťující vysílání byl sestaven jako modul pro přímé připojení skrze vstupně výstupní piny. Bezdrátové ovládání je zajištěno aplikací pro mobilní platformu android, komunikující prostřednictvím bluetooth. A přidávání hudebních souborů skrze wifi připojení. Všechny části byly sestaveny v jeden prototyp, jež byl testován a vyhodnocen.

Při tvorbě projektu vyvstaly některé myšlenky a nápady, které by bylo vhodné prozkoumat, popřípadě realizovat při budoucí práci. Jednou z nich je celkové vypuštění bluetooth technologií z návrhu. Dalším vhodným prvkem by bylo vytvoření sady skladeb, uložené v paměti přehrávače. Z ní by se náhodně vybíraly jednotlivé soubory v případě, že by nebyly uživateli nahrané jiné. Využitelné zejména pokud by hudba plnila roli pouhé zvukové kulisy. Pokud by byla vytvořena druhá verze ovládací aplikace, bylo by na místě využít při její implementaci některých snadno dostupných nástrojů, pro tvorbu multiplatformních web based aplikací, jako např. Cordova. Ve spojení s frameworky pro tvorbu takovýchto projektů by bylo možné vytvořit skutečně moderní nástroj. Razantním vylepšením by měl projít i filtr vysokých frekvencí na modulu vysílacího obvodu. Nový návrh by měl poskytnout zvuk o srovnatelné kvalitě s výstupem 3,5 JACK na Raspberry.

Velkou kapitolou budoucnosti projektu by mohla být možnost připojení internetu. Tím by se dala nabídka přehrávání rozšířit o hudbu ze streamovacích služeb, jako je např. Spotify, nebo portálu YouTube. Velkou výhodou by byl i fakt, že uživatelé připojení na síť přehrávače, by neztratili možnost stálého připojení na internet.

Z pohledu osobního rozvoje byl pro autora projekt velice přínosný a zajímavý. Naskytla se dobrá příležitost vyzkoušet si práci v několika odvětvích, jako je návrh hardwarového obvodu, tvorba mobilní aplikace, nebo webového serveru. To vše zároveň na projektu, který je dobře použitelný i v praxi.

# Literatura

- [1] NEW JAPAN RADIO CO.: *NJM2295A FM IF IC for remote keyless entry system*. [Online; cit. 2016-5-17]. Dostupné z: [http://www.njr.com/semicon/PDF/NJM2295A\\_E.pdf](http://www.njr.com/semicon/PDF/NJM2295A_E.pdf).
- [2] NXP SEMICONDUCTORS: *UM10204 I2C-bus specification and user manual*. [Online; cit. 2016-5-16]. Dostupné z: [http://cache.nxp.com/documents/user\\_manual/UM10204.pdf](http://cache.nxp.com/documents/user_manual/UM10204.pdf).
- [3] RASPBERRY PI FOUNDATION: *Raspberry Pi*. [Online; cit. 2016-5-17]. Dostupné z: <https://www.raspberrypi.org/>.
- [4] RDS FORUM: RDS2 Info Day Conference Report. *Rds.org.uk*. [Online; cit. 2016-5-16]. Dostupné z: <http://www.rds.org.uk/2010/RDS2Day.htm>.
- [5] EN 50067: *Specification of the radio data system (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 to 108,0 MHz*. Brusel:European Committee for Electrotechnical Standardization, 1998, [Online; cit. 2016-5-15]. Dostupné z: [http://www.interactive-radio-system.com/docs/EN50067\\_RDS\\_Standard.pdf](http://www.interactive-radio-system.com/docs/EN50067_RDS_Standard.pdf).
- [6] TEXAS INSTRUMENTS INCORPORATED: *SN761634 FM stereo transmitter*. ©2008 – 2010, [Online; cit. 2016-5-17]. Dostupné z: <http://www.ti.com/lit/ds/symlink/sn761634.pdf>.
- [7] SILICON LABORATORIES: *Si4712/13-B30 FM Radio transmitter with receive power scan*. ©2008, [Online; cit. 2016-5-17]. Dostupné z: <http://www.silabs.com/Support%20Documents/TechnicalDocs/Si4712-13-B30.pdf>.
- [8] QUINTIC CORPORATION: *QN8027 High Performance Digital FM Transmitter for Portable Devices*. ©2009, [Online; cit. 2016-5-17]. Dostupné z: <http://down.cosou.com/xintechsz.com/QN8027.pdf>.
- [9] BLUETOOTH SIG: Bluetooth Specifications. *Bluetooth.com*. ©2016, [Online; cit. 2016-5-16]. Dostupné z: <https://www.bluetooth.com/specifications>.
- [10] WI-FI ALLIANCE: Who We Are. *Wi-fi.org*. ©2016, [Online; cit. 2016-5-16]. Dostupné z: <http://www.wi-fi.org/who-we-are>.
- [11] Bluetooth: In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 10. 5. 2016, [Online; cit. 2016-5-15]. Dostupné z: <https://en.wikipedia.org/wiki/Bluetooth>.

- [12] FM broadcasting: In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 12. 5. 2016, [Online; cit. 2016-5-15]. Dostupné z: [https://en.wikipedia.org/wiki/FM\\_broadcasting](https://en.wikipedia.org/wiki/FM_broadcasting).
- [13] IEEE 802.11: In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 16. 5. 2016, [Online; cit. 2016-5-15]. Dostupné z: [https://en.wikipedia.org/wiki/IEEE\\_802.11](https://en.wikipedia.org/wiki/IEEE_802.11).
- [14] Binary phase-shift keying (BPSK): In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 17. 4. 2016, [Online; cit. 2016-5-17]. Dostupné z: [https://en.wikipedia.org/wiki/Phase-shift\\_keying#Binary\\_phase-shift\\_keying\\_.28BPSK.29](https://en.wikipedia.org/wiki/Phase-shift_keying#Binary_phase-shift_keying_.28BPSK.29).
- [15] Rádiové vlny: In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 24. 11. 2015, [Online; cit. 2016-5-15]. Dostupné z: [https://cs.wikipedia.org/wiki/R%C3%A1diov%C3%A9\\_vlny](https://cs.wikipedia.org/wiki/R%C3%A1diov%C3%A9_vlny).
- [16] Přehled 64-bitových procesorů ARM: In: *Wikipedia: the free encyclopedia*. St. Petersburg (Florida): Wikipedia Foundation, last modified on 25. 2. 2016, [Online; cit. 2016-5-12]. Dostupné z: [https://cs.wikipedia.org/wiki/ARM#P.64-bitov.C3.BDch\\_procesor.C5.AF\\_ARM](https://cs.wikipedia.org/wiki/ARM#P.64-bitov.C3.BDch_procesor.C5.AF_ARM).
- [17] Albert, A. L.: *Electrical communication*. 1954, [Online; cit. 2016-5-12]. Dostupné z: [https://www.princeton.edu/ssp/joseph-henry-project/henry-bell/albert\\_elect\\_comm.pdf](https://www.princeton.edu/ssp/joseph-henry-project/henry-bell/albert_elect_comm.pdf).
- [18] Halliday, D.; Resnick, R.; Walker, J.: *Fyzika*. VUTIUM, 2014, ISBN 978-80-214-4123-1.
- [19] Hoag, J. B.: *Basic radio, the essentials of electron tubes and their circuits*. 1942, [Online; cit. 2016-5-12]. URL <http://www.tubebooks.org/Books/hoag.pdf>
- [20] Kopitz, D.; Marks, B.: *RDS: The Radio Data System*. Artech House, ©1999, ISBN 0-89006-744-9.
- [21] RKROPY: Co je Digital Audio Broadcasting. *Digitalradiodab.cz*. ©2014, [Online; cit. 2016-5-17]. Dostupné z: <http://www.digitalradiodab.cz/co-je-dab.html>.
- [22] ČESKO: ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD. *Všeobecné oprávnění č. VO-R/10/05.2014-3 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu*. [Online; cit. 2016-5-12]. Dostupné z: [https://www.ctu.cz/cs/download/oop/rok\\_2014/vo-r\\_10-05\\_2014-03.pdf](https://www.ctu.cz/cs/download/oop/rok_2014/vo-r_10-05_2014-03.pdf).
- [23] ČESKO: ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD. *Část plánu využití rádiového spektra č. PV-P/22/11.2013-7*. [Online; cit. 2016-5-15]. Dostupné z: [http://www.ctu.cz/cs/download/plan-vyuziti-radioveho-spektra/rok\\_2013/pv-p\\_22-11\\_2013-07.pdf](http://www.ctu.cz/cs/download/plan-vyuziti-radioveho-spektra/rok_2013/pv-p_22-11_2013-07.pdf).
- [24] ČESKO: Zákon č. 426 ze dne 23. října 2000 kterým se stanoví technické požadavky na rádiová a na telekomunikační koncová zařízení. In: *Sbírka zákonů České republiky*. 2000, částka 119, s. 5738-5750. Dostupný také z: <http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=c&id=3507>.

[25] ČESKO: ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD. Koncová zařízení. *Ctu.cz*. ©2016,  
[Online; cit. 2016-5-17]. Dostupné z: <https://www.ctu.cz/koncova-zarizeni>.

# Přílohy

## Seznam příloh

<b>A</b>	<b>Obsah CD</b>	<b>35</b>
<b>B</b>	<b>Schéma vysílacího obvodu</b>	<b>36</b>
<b>C</b>	<b>Navržený tištěný spoj</b>	<b>37</b>
<b>D</b>	<b>Seznam potřebných součástek pro obvod</b>	<b>38</b>

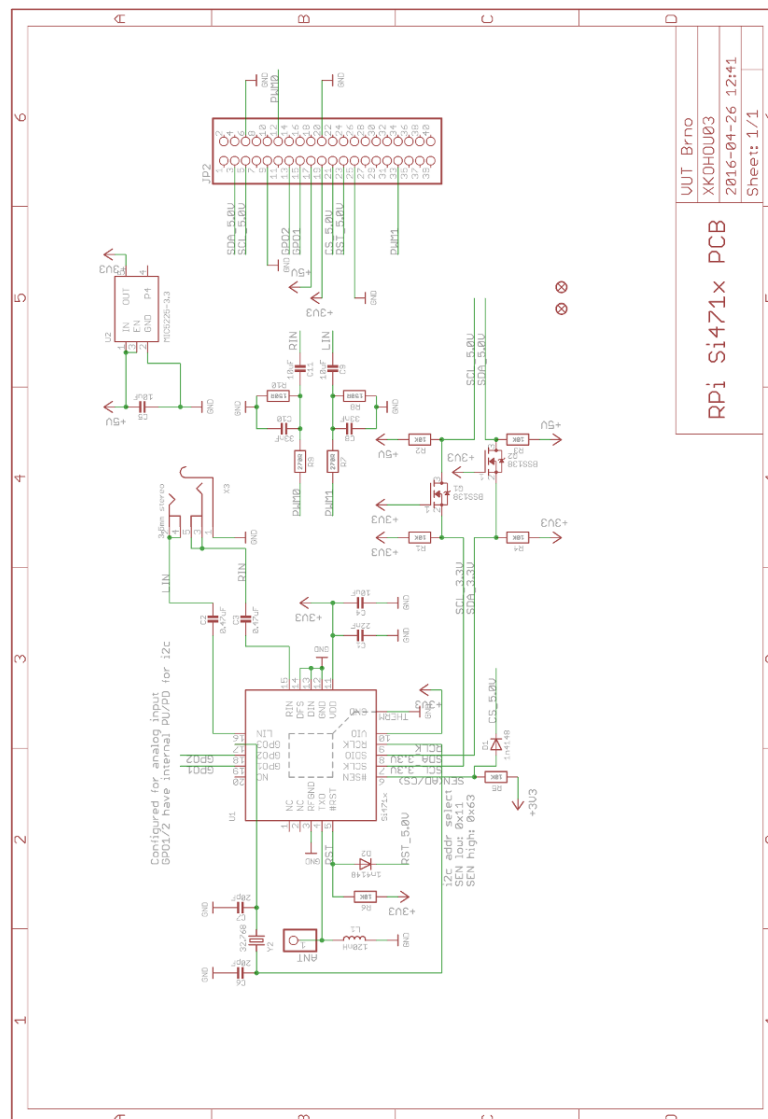
# Příloha A

## Obsah CD

- Android – Adresář se soubory pro kompilaci mobilní aplikace.
- EAGLEsrc – Zdrojové soubory programu EAGLE navrženého vysílacího obvodu (REV B).
- PlayerPy – Python sada skriptů pro přehrávač Raspberry.
- technickaZprava2016 – Obsahuje zdrojové soubory technické zprávy k BP.
- xkohou03.pdf – Soubor pdf obsahující technickou zprávu.
- startUp.sh – Skript určený pro automatické spouštění pro Raspberry po loginu.
- LICENCE – Soubor s licenčním ujednáním.
- README – Návod ke spuštění.

# Příloha B

## Schéma vysílacího obvodu



UUT Brno
XK0H0U03
2016-04-26 12:41
Sheet: 1/1

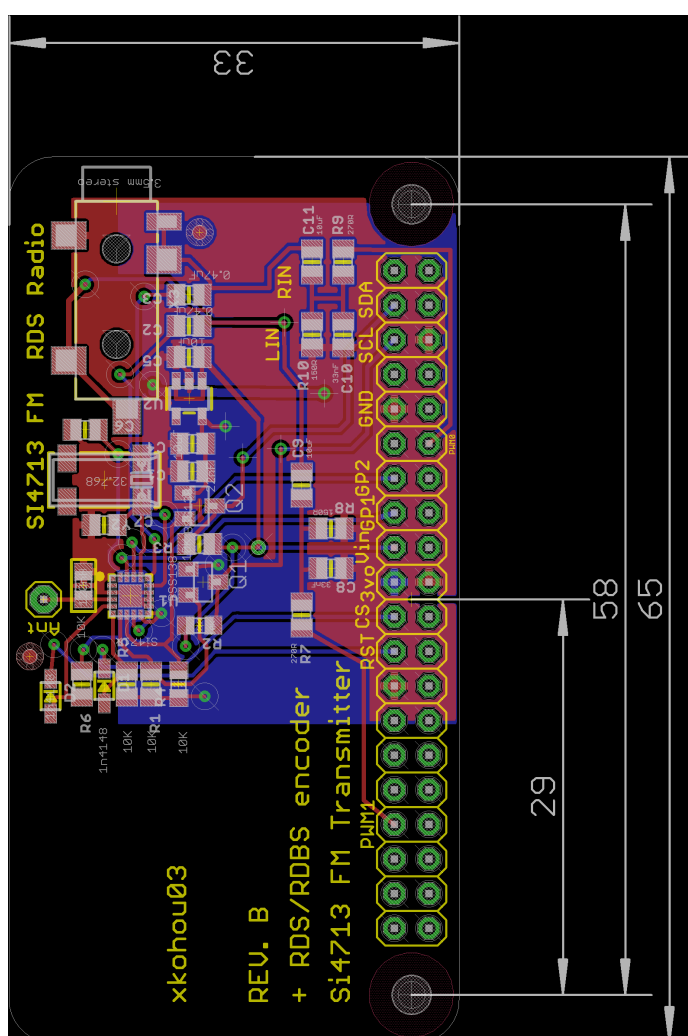
RPi Si471X PCB

Obrázek 15: Schéma vysílacího obvodu s čipem Si4713



## Příloha C

# Navržený tištěný spoj



Obrázek 16: Tištěný spoj obvodu s čipem Si4713

## Příloha D

# Seznam potřebných součástek pro obvod

- Si4713 – Vysílací čip
- Mic5225-3.3 – Konvertor napětí
- JP2 – 40pinoý konektor
- BSS138 – 2 × bipolární tranzistor
- 3.5 JACK stereo
- Oscilátor – 32.768 kHz
- Kondenzátory – (2 × 20 pF, 2 × 33 nF, 2 × 0.47  $\mu$ F, 3 × 10  $\mu$ F)
- Rezistory – (2 × 150  $\Omega$ , 2 × 270  $\Omega$ , 6 × 10 k $\Omega$ )
- Cívka – (120 nH)
- Diody – (2 × 1n4148)