

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

## RADIOVÝ DAB PŘIJÍMAČ

DAB RADIO RECEIVER

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Ondřej Pišťák

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Dřínovský, Ph.D.

BRNO 2021

# Bakalářská práce

bakalářský studijní program **Elektronika a komunikační technologie**

Ústav radioelektroniky

**Student:** Ondřej Pišťák

**ID:** 211315

**Ročník:** 3

**Akademický rok:** 2020/21

**NÁZEV TÉMATU:**

## Radiový DAB přijímač

### POKYNY PRO VYPRACOVÁNÍ:

Podrobně se seznamte s možností příjmu DAB signálu. Dále se zaměřte na analýzu vlastností jednotlivých softwarově definovaných přijímačů radiových signálů DAB. Na základě vašich vlastností navrhnete DAB přijímač realizovaný na platformě Raspberry Pi. Pro tuto platformu vyberte vhodný USB přijímač, případně HAT pro RPi platformu. K přijímači navrhnete ovládací panel s tlačítky a displejem pro zobrazení informací o přijímané radiové stanici. Ověřte základní funkčnost vámi navrženého přijímače např. sestavením na nepájivém poli.

Realizujte vámi navržené komponenty DAB přijímače na desce plošných spojů. Jednotlivé moduly oživte. Přijímač oživte instalací a úpravou vhodného SW pro platformu RPi. Přijímač doplňte o dálkové IR ovládání jeho základních funkcí.

### DOPORUČENÁ LITERATURA:

[1] BCM2711 ARM Peripherals [online], 2020. Raspberry Pi (Trading) [cit. 2020-05-18]. Dostupné z: [https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi\\_DATA\\_2711\\_1p0.pdf](https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0.pdf)

[2] BCM2835 ARM Peripherals [online], 2018. 406 Science Park Milton Road Cambridge CB4 0WW: Broadcom Europe [cit. 2020-05-18]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/BCM2835-ARM-Peripherals.pdf>

[3] UGreen DAB Board Instructions v8 [online], 2019. Frobenstrasse 37, 4053 Basel: uGreen [cit. 2020-05-18]. Dostupné z: [https://ugreen.eu/wp-content/uploads/2019/08/DABBoard\\_Instructions\\_v8.pdf](https://ugreen.eu/wp-content/uploads/2019/08/DABBoard_Instructions_v8.pdf)

**Termín zadání:** 8.2.2021

**Termín odevzdání:** 27.5.2021

**Vedoucí práce:** Ing. Jiří Dřínovský, Ph.D.

**prof. Ing. Tomáš Kratochvíl, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Práce je zaměřena na návrh a konstrukci softwarově definovaného rádia, které umí přehrávat standard DAB+. Zpracování signálu zachyceného pomocí antény zajišťuje USB přijímač RTL-SDR BLOG s 8 – bitovým A/D převodníkem RTL2832U a platforma Raspberry Pi 3 (Model B). Práce je také zaměřena na návrh a realizaci ovládacích prvků k aplikaci pro jednoduché přepínání stanic pomocí tlačítek na ovládacím panelu. Pro zobrazení právě přehrávané stanice slouží připojený LCD display.

## **Klíčová slova**

SDR, Softwarově definované rádio, DAB+, Digitální rozhlasové vysílání, Raspberry Pi

## **Abstract**

This work is focused on the design and construction of software-defined radio that can play DAB+ standard. The received signal is processed by USB receiver RTL-SDR BLOG with 8 bits resolution A/D convertor RTL2832U and small single-board computer Raspberry Pi 3 (Model B). The work deals also with the design and realization of the control panel by which the radio is controlled. The LCD display is connected to Raspberry Pi and used for the screen information about currently playing radio program.

## **Keywords**

SDR, Software Defined Radio, DAB+, Digital Audio Broadcasting, Raspberry Pi

## **Bibliografická citace**

PIŠŤÁK, Ondřej. Radiový DAB přijímač. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/133595>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Jiří Dřínovský.

## Prohlášení autora o původnosti díla

<b>Jméno a příjmení studenta:</b>	<i>Ondřej Pišták</i>
<b>VUT ID studenta:</b>	<i>211315</i>
<b>Typ práce:</b>	<i>Bakalářská práce</i>
<b>Akademický rok:</b>	<i>2020/21</i>
<b>Téma závěrečné práce:</b>	<i>Rádiový DAB přijímač</i>

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 27. května 2021

-----  
podpis autora

## **Poděkování**

*Děkuji vedoucímu semestrální práce Ing. Jiřímu Dřínovskému, Ph.D. za odborné konzultace, pomoc a další cenné rady při zpracování mé bakalářské práce.*

*Rád bych poděkoval programátorovi Janu van Katwijkovi za odborné rady týkající se vybrané aplikace pro přehrávání DAB vysílání.*

## **Acknowledgement**

*I would like to thank my supervisor Ing. Jiří Dřínovský, Ph.D. for professional consultation and useful advice on my thesis.*

*I would like to thank programmer Jan van Katwijk for helpful advice about application for playing DAB.*

V Brně dne: 27. května 2021

-----  
podpis autora

# Obsah

<b>SEZNAM SYMBOLŮ A ZKRATEK .....</b>	<b>9</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>10</b>
<b>ÚVOD .....</b>	<b>11</b>
<b>1. TEORIE .....</b>	<b>12</b>
1.1 OBECNÝ POPIS DAB.....	12
1.1.1 DAB v ČR.....	12
1.2 DAB+ .....	13
1.3 BLOKOVÉ SCHÉMA DAB / DAB+.....	13
1.3.1 Zdrojové kódování (komprese).....	14
1.3.2 Kanálové kódování.....	14
1.3.3 Prokládání .....	15
1.3.4 OFDM.....	15
1.3.5 DQPSK.....	16
1.4 DÁLKOVÉ OVLÁDÁNÍ .....	17
1.4.1 Protokol JVC.....	17
1.4.2 Protokol Panasonic.....	18
1.4.3 Protokol SIRC.....	18
1.4.4 Protokol NEC.....	19
<b>2. NÁVRH DAB PŘIJÍMAČE .....</b>	<b>20</b>
2.1 LCD DISPLAY .....	20
2.2 NÁVRH OVLÁDACÍHO PANELU .....	21
2.3 VÝBĚR VHODNÉHO DAB PŘIJÍMAČE PRO RASPBERRY PI .....	22
2.4 VÝBĚR DAB APLIKACE.....	23
2.5 VÝBĚR DÁLKOVÉHO OVLÁDÁNÍ A IR PŘIJÍMAČE .....	25
<b>3. REALIZACE DAB PŘIJÍMAČE .....</b>	<b>27</b>
3.1 VÝBĚR A INSTALACE OS .....	27
3.2 ZAPOJENÍ PERIFERÍ.....	27
3.2.1 Zapojení displeje.....	27
3.2.2 Zapojení IR přijímače .....	27
3.2.3 Zapojení ovládacích tlačítek.....	28
3.3 INSTALACE DAB PŘEHRÁVAČE.....	28
3.3.1 Instalace knihoven .....	28
3.4 POPIS DOPLNĚNÝCH FUNKCÍ.....	30
3.4.1 Potřebné knihovny .....	30
3.4.2 Definování maker.....	30
3.4.3 Deklarace funkcí .....	31
3.4.4 Definice funkcí .....	32
3.5 DESKA PLOŠNÝCH SPOJŮ .....	39
3.6 AUTOMATICKÉ SPUŠTĚNÍ PO ZAPNUTÍ .....	41
3.6.1 Vytvoření scriptu.....	41
3.6.2 Automatické spuštění .....	41



<b>4. ZÁVĚR.....</b>	<b>42</b>
<b>LITERATURA.....</b>	<b>43</b>

### **SEZNAM PŘÍLOH**

**PŘÍLOHA 1: ZDROJOVÝ KÓD LINKOVACÍHO SKRIPTU – CmakeLists.txt**

**PŘÍLOHA 2: ZDROJOVÝ KÓD HLAVNÍHO PROGRAMU V C++ – main.cpp**

**PŘÍLOHA 3: NÁVRH DESKY PLOŠNÝCH SPOJŮ OVLÁDACÍHO PANELU – ovladani.brd**

**PŘÍLOHA 4: SCHÉMA ZAPOJENÍ OVLÁDACÍHO PANELU – ovladani.sch**

# SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

AM	Amplitudová modulace
CD	Compact Disc
ČR	Česká republika
DAB	Digital Audio Broadcasting
DQPSK	Differential Quadrature Phase Shift Keying
EEP	Equal Error Protection
FM	Frekvenční modulace
FEC	Forward Error Protection
GPIO	General Purpose Input / Output
HE-AACv2	High Efficiency Advanced Audio Coding
IR	Infra Red
LCD	Liquid Crystal Display
MPEG	The Moving Picture Experts Group
OFDM	Orthogonal Frequency Division Multiplexing
QPSK	Quadrature Phase Shift Keying
RMS	Root Mean Square
RPI	Raspberry Pi
SDR	Software Defined Radio
UEP	Unequal Error Protection

## SEZNAM OBRÁZKŮ

Obrázek 1: Blokové schéma DAB vysílače [2] .....	13
Obrázek 2: Blokové schéma DAB přijímače [2] .....	14
Obrázek 3: OFDM rozložení subnosných [27] .....	15
Obrázek 4: Diagram modulačních schémat QPSK a DQPSK [28] .....	16
Obrázek 5: Protokol JVC - časový průběh [33] .....	17
Obrázek 6: Protokol Panasonic- časový průběh [33] .....	18
Obrázek 7: Protokol SIRC – úrovně [33] .....	18
Obrázek 8: Protokol SIRC - časový průběh [33] .....	19
Obrázek 9: Protokol NEC - časový průběh [34] .....	19
Obrázek 10: LCD display [29] .....	20
Obrázek 11: I2C expandér [30] .....	20
Obrázek 12: Blokové zapojení expandéru [21] .....	21
Obrázek 13: Zapojení tlačítka se Schmittovým klopným obvodem .....	22
Obrázek 14: Náhled RLT-SDR klíčenky [24] .....	23
Obrázek 15: GUI aplikace dab-mini [25] .....	24
Obrázek 16: Testovací zapojení komponent .....	24
Obrázek 17: Dálkové ovládání (vlevo), IR přijímač (vpravo dole) [36] .....	25
Obrázek 18: Blokové schéma IR přijímače [35] .....	26
Obrázek 19: Časový signál z ovladače (nahore), výstup z IR přijímače (dole) [35] .....	26
Obrázek 20: Import knihoven [38], [40] .....	30
Obrázek 21: Použitá makra [26], [39] .....	31
Obrázek 22: Deklarace LCD a IR přijímače [26], [40] .....	31
Obrázek 23: Deklarace tlačítek .....	32
Obrázek 24: Definice LCD funkcí 1 [26] .....	32
Obrázek 25: Definice LCD funkcí 2 [26] .....	33
Obrázek 26: Definice IR přijímače [40] .....	34
Obrázek 27: Nastavení pinů a přerušení [38] .....	35
Obrázek 28: Definice funkcí tlačítek 1 .....	36
Obrázek 29: Definice funkcí tlačítek 2 .....	37
Obrázek 30: Definice funkce dálkového ovladače .....	38
Obrázek 31: Návrh desky s tlačítky .....	39
Obrázek 32: Výsledné zapojení a deska ze strany součástek .....	40
Obrázek 33: Výsledné zapojení a deska ze strany spojů .....	40

# ÚVOD

Tato bakalářská práce se zabývá návrhem a konstrukcí radiového přijímače, který umí přijímat a zpracovávat signál DAB / DAB+ a využívá k tomu platformu Raspberry Pi 3. Díky tomu, že zvolená platforma má malé rozměry bude celé rádio kompaktní. Pro výběr správných komponent přijímajících digitální rozhlasové vysílání je třeba seznámit se s vlastnostmi a parametry vysílání DAB a DAB+. K tomu slouží norma ČSN ETSI EN 300 401 V2.1.1 [2]. V ní jsou uvedené veškeré podstatné informace týkající se digitálního rozhlasového vysílání včetně pásma, kde se ve standardu DAB a DAB+ vysílá. Vybraný modul pro příjem signálu musí být tedy malý a kompatibilní s platformou, na které se bude rádio realizovat. Výběr vhodného zobrazovacího prvku je podmíněn malým odběrem energie a pěkným grafickým výstupem. Ovládací panel by měl být přehledný a velikostně úsporný. Na platformě Raspberry Pi bude spuštěn 32-bitový operační systém Raspbian a SD karta o velikosti 16 GB. Tyto informace poslouží k tomu, aby se na platformu neinstalovala nebo nevytvářela příliš velká aplikace z důvodu omezeného místa na Raspberry Pi. Po návrhu všech potřebných komponentů je potřeba ověřit správnost tohoto návrhu jeho realizací a vyzkoušením celého DAB přijímače s ovládacím panelem a displejem na nepájivém poli. Dalším krokem je vytvoření desky plošných spojů, která nahradí nepájivé pole. Poslední část je implementace dálkového ovládání do programu. Celé rádio bude vloženo do krabičky, která se vytiskne na 3D tiskárně.

# 1. TEORIE

V této kapitole budou popsány vlastnosti vysílacích standardů DAB/ DAB+, výhody a nevýhody tohoto vysílání, porovnání DAB a DAB+ standardů a popis různých bloků DAB přijímače a jejich vlastnosti. Dále si uvedeme potřebné informace a komponenty k návrhu přijímače.

## 1.1 Obecný popis DAB

Digitální rozhlasové vysílání (Digital Audio Broadcasting) zkráceně DAB je technologie, která vznikla za účelem nahradit AM a FM rozhlasové vysílání. Byla vytvořena v rámci projektu Eureka 147/DAB v roce 1990. DAB je vhodný pro mobilní příjem a je vybaven ochranným intervalem, který společně s modulací OFDM (Orthogonal Frequency Division Multiplexing) a diferenčním QPSK (Quadrature Phase-Shift Keying) zaručí odolnost užitečného signálu vůči vícecestnému šíření. Kvalita zvuku je srovnatelná s kvalitou CD. Při vysílání ve standardu DAB je také možné přenášet další informace, jako jsou například informace o vysílacích stanicích, textové zpravodajství a další. Vysílání DAB lze využít v pásmu 3 MHz až 3 GHz. Bližší informace o vysílacím standardu DAB lze nalézt v [1], [2], [3], [4] a [5].

### 1.1.1 DAB v ČR

V České republice bylo DAB vysíláno ve III. televizní pásmu (174 – 240 MHz) a v pásmu L (1452 – 1492 MHz). Od začátku roku 2021 se již vysílá v ČR jen ve III. televizním pásmu. Šířka kanálu DAB byla stanovena jako 1,536 MHz a přenosová rychlost na 1,8 Mbit/s. Tento systém využívá diferenciální modulaci QPSK a OFDM, díky těmto modulacím je signál odolný proti chybám způsobených vícecestným šířením signálu. DAB využívá audio formát MPEG Audio layer II (MP2).

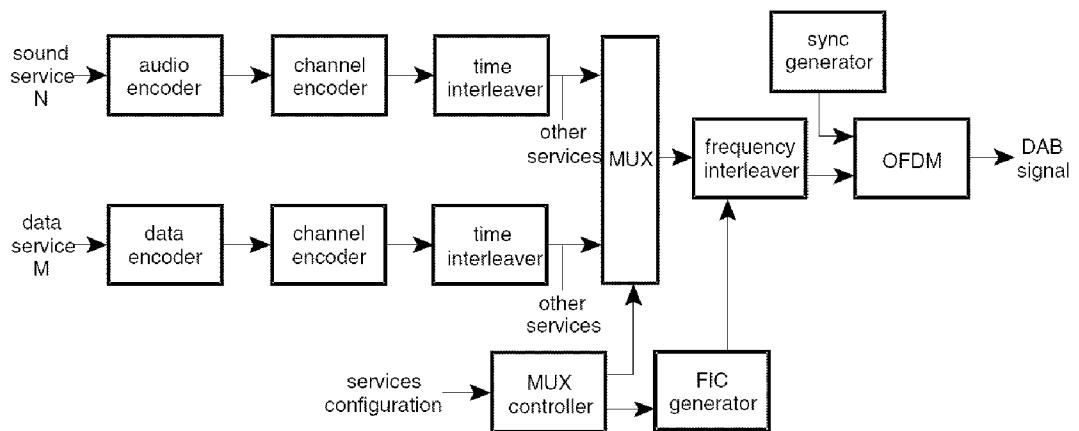
V Praze roce 1999 proběhlo v pásmu L první DAB vysílání v České republice, které bylo spuštěno z vysílače na Žižkovské věži. Větší rozsah vysílání zahájila až firma Teleko a v roce 2013 bylo toto vysílání dostupné pro více než polovinu obyvatel České republiky. Později v tomto roce se objevila další firma s tímto typem vysíláním, a to RTI cz s.r.o. Jako první zahájil v roce 2015 vysílání vysokým výkonem Český rozhlas s multiplexerem ČRo DAB+ z vysílače Žižkov v Praze. V Čechách multiplex Českého rozhlasu využívá kanál 12C a na Moravě kanál 12D. Dne 4. března 2021 byl vypnut poslední vysílač v pásmu L, čímž se veškeré digitální rozhlasové vysílání přesunulo do III. televizního pásma [1], [2], [3], [4], [5], [6], [7] a [43].

## 1.2 DAB+

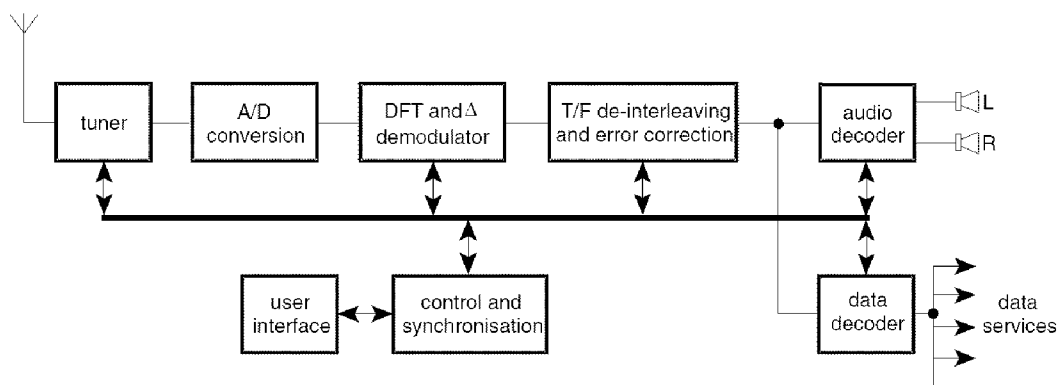
DAB+ je vylepšená verze systému DAB, která byla uvedena v roce 2017. Oproti standardu DAB se zde používá audio formát MPEG-4 HE-AACv2, což umožňuje začlenění více kanálů do jedno multiplexu. Zvuková kvalita je sice velmi podobná, ale velkou výhodou je právě snížení datového toku. Dalším výrazným vylepšením je začlenění Reedova-Solomonova samoopravného kódu [1], [2], [3], [5], [6] a [7].

## 1.3 Blokové schéma DAB / DAB+

Na obrázcích 1 a 2 je zobrazeno blokové schéma systému DAB / DAB+. Audio signál je nejprve zdrojově kódovaný (komprimován) pomocí formátu MPEG Audio layer II nebo MPEG-4 HE-AACv2 v závislosti na druhu systému DAB / DAB+. Po komprimaci signálu je prováděno kanálové kódování, jehož úkolem je zabezpečit data proti chybám. Poté se ještě data časově prokládají, což zlepší odolnost proti případným výpadkům sekvence bitů. Tyto operace se provedou i s datovým vstupem a obě tyto větve vstupují do multiplexeru. V multiplexeru se zpracují data tak, aby odpovídala konfiguraci daného módu. Za multiplexerem se výsledná data opět prokládají, ale tentokrát frekvenčně. Tento signál vstupuje do bloku OFDM, kde je modulován pomocí modulace OFDM a poté se vytvořené subnosné frekvence modulují pomocí diferenciální QPSK. Tím je signál připraven na vysílání [1], [2], [3], [6] a [7].



Obrázek 1: Blokové schéma DAB vysílače [2]



Obrázek 2: Blokové schéma DAB přijímače [2]

### 1.3.1 Zdrojové kódování (komprese)

Zdrojové kódování slouží ke snížení přenosové rychlosti. V systému DAB jde především o komprimaci zvuku, ale i některých dodatečných informací, které se přidávají k celkovému datovému toku. K této komprimaci slouží MPEG Audio layer II. použitý v systému DAB a MPEG-4 HE-AACv2 pro systém DAB+.

MPEG Audio layer II (Moving Picture Experts Group) je standard používaný při ztrátové kompresi audiovizuálních dat. Historicky byly vytvořeny 3 vrstvy standardu MPEG (MPEG-1). Redukce datového toku je založena na omezeném vnímání lidských smyslů. V našem případě se využívá psychoakustický model lidského sluchu a maskování zvuků vnímaných lidským ušním. Pomocí banky filtrů se signál rozdělí na 32 subpásem. Data z psychoakustického modelu a data vycházející z banky filtrů se dále musí kvantovat a kódovat. Výsledný datový tok se vytvoří pomocí rámcování. Pro systém DAB jsou uvedeny povolené vzorkovací frekvence 48 kHz a 24 kHz.

MPEG-4 HE-AACv2 (HE-AAC- High Efficiency Advanced Audio Coding) je standard, který se používá ke ztrátové kompresi stejně jako předchozí typ. Patří mezi jeden z nejpokročilejších formátů zvukové komprese. Je využit v systému DAB+, jelikož MPEG Audio layer II, použitý v DAB, neměl dostačující vlastnosti ke snížení bitové rychlosti a zvýšení kvality. Povolené vzorkovací frekvence jsou 16 kHz, 24 kHz, 32 kHz a 48 kHz [1], [2], [3], [6], [7] a [8].

### 1.3.2 Kanálové kódování

Kanálové kódování představuje ochranu proti chybám, které mohou vzniknout například vlivem šumu a ostatních nežádoucích vlastností přenosového kanálu. Přidáním takového kódu ovšem zvětšíme datový tok, protože přidáváme redundantní informaci k původnímu audio signálu.

V systému DAB je možné použít UEP (Unequal Error Protection), což je nerovnoměrné zakódování dat, nebo EEP (Equal Error Protection). UEP se používá především pro audio, ale je možné s ním kódovat i jiná data. V paketovém módu se

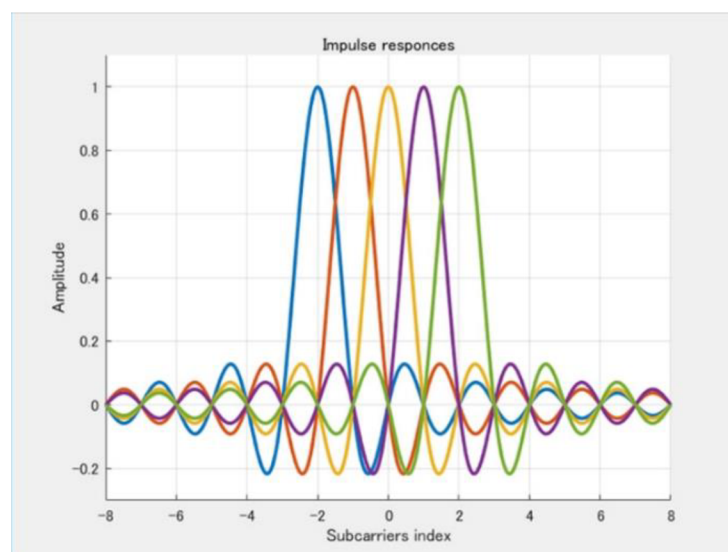
využívá FEC (Forward Error Correction) k dalšímu zabezpečení a prokládání dat. Kódů, které se používají pro ochranu dat je velké množství a většinou se jich v jednom systému používá více na různých místech. Detailnější informace o problematice kanálového kódování lze nalézt v [1], [2], [3], [6] a [7].

### 1.3.3 Prokládání

Prokládáním se řeší problém případného výpadku bitové sekvence (několik bitů po sobě). Opravné kódy jsou schopné opravit jen chyby šumového charakteru. Časové prokládání nám tedy zaručí že původní posloupnost bitů se před vysláním systematicky zamění. V dekodéru se opět posloupnost zamění tak, aby odpovídala původní sekvenci bitů. V případě výpadku řady bitů na přenosové cestě k dekodéru se již bity mohou opravit, jelikož chybné bity jsou rozprostřeny [1], [2], [3] a [7].

### 1.3.4 OFDM

OFDM (Ortogonal Frequency Division Multiplexing) je modulační metoda, která využívá rozdělení původního datového toku do více pomalejších datových toků. Ty jsou modulovány až na několik tisíc subnosných frekvencí a na jednotlivé subnosné se používají další modulace. Ve standardu DAB je použito modulace DQPSK. Díky ortogonalitě je každé maximum subnosné překryto minimem ostatních subnosných frekvencí. Tím je také dána spektrální účinnost OFDM, která je oproti ostatním modulacím velmi vysoká. Zavedením ochranného intervalu se zamezí chybám, které vznikají v důsledku vícecestného šíření signálu. Na obrázku 3 je zobrazeno jen pár takových subnosných frekvencí pro ucelení představy, ale v praxi se používají stovky až tisíce těchto ortogonálních nosných frekvencí [1], [2], [3], [7], [9] a [10].

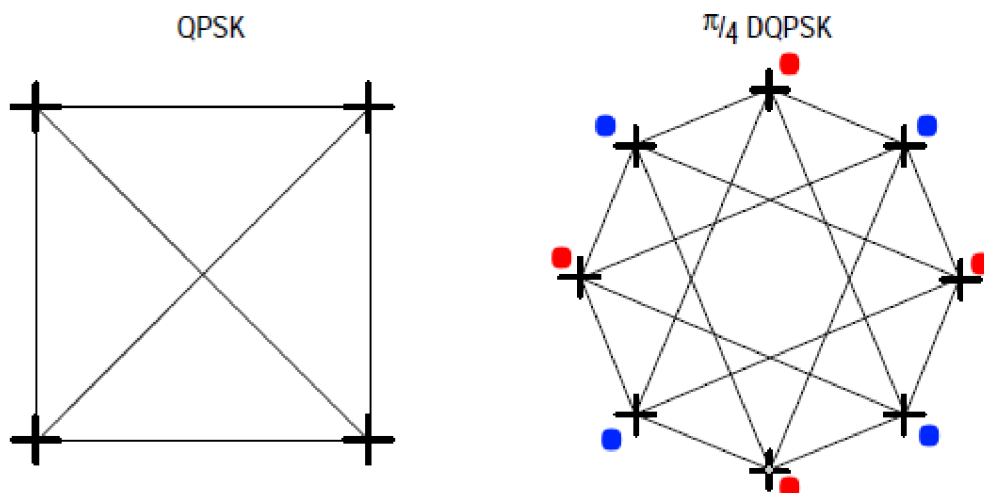


Obrázek 3: OFDM rozložení subnosných [27]



### 1.3.5 DQPSK

Digitální fázová modulace QPSK (Quadrature Phase-shift keying) funguje na principu změny fáze, která může nabývat čtyř hodnot (stavů). Stavů jsou závislé na vstupních hodnotách dvojice prvků (I, Q). Výsledné stavy jsou vzájemně posunuty o  $90^\circ$ . Pokud nastane změna v hodnotě I a Q zároveň (v diagramu prochází přes střed obou os), objeví se parazitní 100 % AM modulace. Proto se používá  $\pi/4$ -DQPSK, která zamezuje přechodu přes střed diagramu. Ve zkratce této modulace je D, které značí diferenciální kódování. Tento typ kódování odstraňuje chyby případného invertování prvků (0 => 1 a naopak). Na obrázku 4 si můžete prohlédnout diagram QPSK a  $\pi/4$ -DQPSK ( $\pi/4$ -QPSK) [1], [2], [3], [9] a [11].



Obrázek 4: Diagram modulačních schémat QPSK a DQPSK [28]

## 1.4 Dálkové ovládání

Dálkové ovládání (ovladač) je nástroj, který umožňuje ovládat nebo řídit určitou činnost jiného zařízení bez přímého přístupu k tomuto zařízení. V dřívějších dobách toto ovládání používalo ultrazvukové vlny pro komunikaci. Dnes se však využívá infračerveného (IR) záření, které je digitálně kódované a přenesené na IR přijímač zařízení, které chceme ovládat. Obvykle se využívá ovládání hlasitosti, ztlumení, přepínání programu, vypnutí / zapnutí a podobné funkce. Infračervené záření je v okolí 950 nm, což je ve spektru pro člověka nevnímání. Hlavní důvod použití tohoto záření je, že uživatel není rušen paprskem, který je vysílán z ovladače. V dálkových ovladačích se IR záření moduluje, protože jinak by bylo rušeno od okolních zdrojů (např. Slunce, žárovka). IR dioda nevysílá po celou dobu kontinuálně, ale je buzena nosným obdélníkovým signálem nejčastěji 36, 38 a 40 kHz. Ten je poté modulovaný impulsním signálem, který nese informaci o povelu. Přesné informace, co značí logickou 1, logickou 0, frekvenci nosné a další, jsou uvedeny v tzv. komunikačním protokolu. Těchto protokolů je velké množství a jsou především firemní záležitostí. Protokoly jsou vzájemně nekompatibilní [31], [32], [33].

### 1.4.1 Protokol JVC

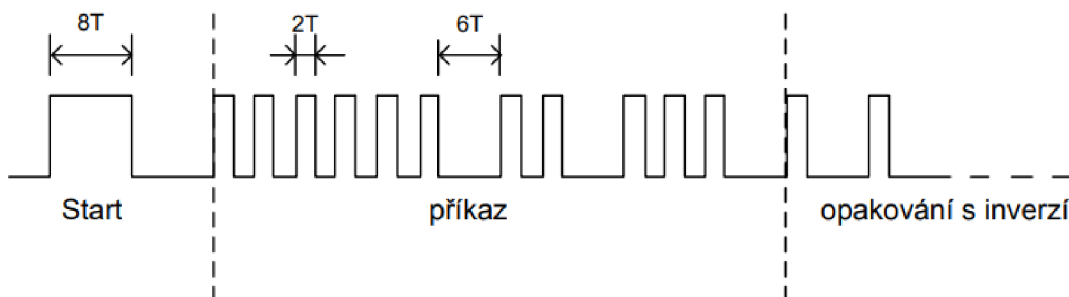
Tento protokol přenáší 8-bitové adresy a 8-bitové povely. Kmitočet nosné vlny je 38 kHz a využívá se kódování bitů do vzdálenosti mezi impulsy. Jeden impuls trvá 526  $\mu$ s, což je 20 period nosného kmitočtu 38 kHz. Logická 1 představuje v tomto protokolu dobu trvání (mezi impulsy) 2,1 ms a logická 0 1,05 ms. Na začátku povelu je vždy start impuls, který trvá 8,4 ms a po něm mezera trvající 4,2 ms. Poté následuje adresa a za ní povel [33].



Obrázek 5: Protokol JVC – časový průběh [33]

### 1.4.2 Protokol Panasonic

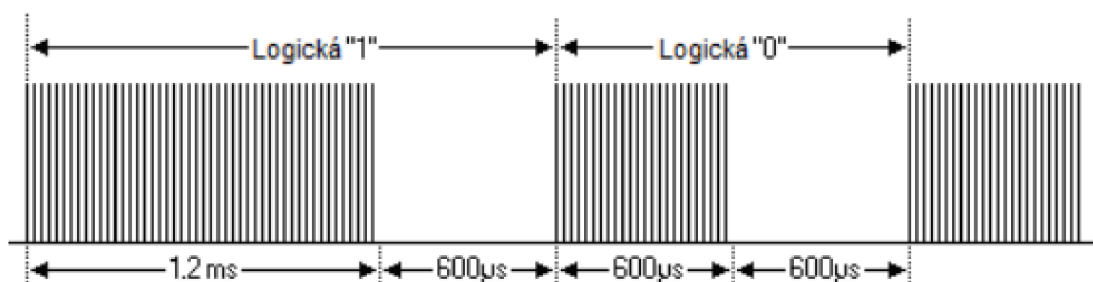
Protokol Panasonic přenáší 6-bitový povel a 5-bitový kód, a to s nosným kmitočtem 38 kHz. Jako základní časový úsek T je pro Evropu stanoveno 454–460  $\mu$ s. Logická 1 se označuje jako 2T úrovně H a 6T úrovně L. Logická 0 se kóduje jako 2T úrovně H a 2T úrovně L. Opět se zde na začátku vysílá start impuls a má délku 8T (úroveň H). Celková vyslaná zpráva má délku 22 bitů, jelikož se vysílá 6 bitů pro povel, 5 bitů pro kód a poté se opět vysílají tyto bity, ale v invertované podobě [33].



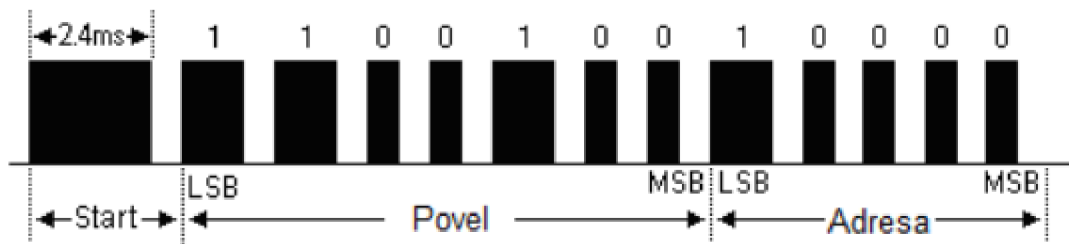
Obrázek 6: Protokol Panasonic – časový průběh [33]

### 1.4.3 Protokol SIRC

Tento protokol využívá několik verzí s různými velikostmi (12 bitů, 15 bitů a 20 bitů). U 12-bitového protokolu se používá 5 bitů na adresu a 7 bitů na povel. Nosný kmitočet je 40 kHz. Startovací impuls trvá 2,4 ms a za ním následuje mezera 0,6 ms. Jako první se po start impulsu vysílá povel a následně adresa. Opakování příkazů je po 45 ms [33].



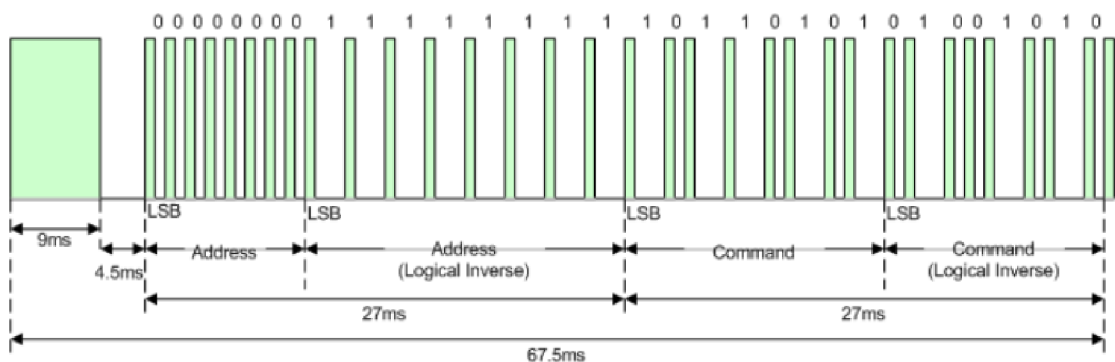
Obrázek 7: Protokol SIRC – úrovně [33]



Obrázek 8: Protokol SIRC – časový průběh [33]

#### 1.4.4 Protokol NEC

Nosná frekvence tohoto protokolu je 38 kHz a základní burst má délku 562,5  $\mu$ s. Logická 0 je definovaná jako burst trvající 562,5  $\mu$ s a následná mezera 562,5  $\mu$ s. Logická 1 se kóduje jako burst 562,5  $\mu$ s a mezera 1,6875 ms. Celkové časy pro trvání logické 1 a logické 0 jsou 2,25 ms (logická 1) a 1,125 ms (logická 0). Protokol využívá 8 - bitovou adresu a 8 – bitový příkaz, přičemž se vysílají tato data i inverzně. Dohromady tedy 4 byty dat + start impuls, který je tvořen pulsem trvajícím 9 ms a následnou mezerou 4,5 ms [34].



Obrázek 9: Protokol NEC – časový průběh [34]

## 2. NÁVRH DAB PŘIJÍMAČE

V této kapitole se budeme zabývat návrhem DAB přijímače. Návrh se bude skládat z několika částí a to: Výběr vhodného LCD displeje, návrh ovládacího panelu, návrh a provedení DAB+ přijímače na platformě Raspberry Pi 3 a výběr dálkového ovládání.

### 2.1 LCD display

Tato součástka je přidána k DAB+ přijímači pro zobrazování názvů stanic tak, jako je tomu u normálního digitálního rádia. LCD displej, jak již napovídá zkratka LCD (Liquid Crystal Display), je tvořen z tekutých krystalů. Pomocí této konstrukce displeje je spotřebováváno velmi málo energie pro vlastní funkčnost. Pro naše využití není potřeba nějak velkého displeje. K zobrazení základních informací byl vybrán LCD 2x16 (2 řádky 16 znaků) RG1602A s modrým podsvícením. Samotný displej může pracovat v 4-bitovém nebo v 8-bitovém módu. U 4-bitového módu se sice ušetří připojení 4 pinů na GPIO v Raspberry Pi, ale programově to znamená, že 8-bitová data se budou muset posílat nadvrát. Já jsem se rozhodl, že bude z konstrukčního hlediska nejideálnější použít sériový přenos dat pomocí komunikačního protokolu I2C. Displej se následně připojí pouze přes dva signálové piny (hodinový, datový). Dohromady jsou tedy potřeba pouze 4 propojovací vodiče, pokud počítáme i napájení a zem. K tomu je ovšem potřeba hardwarová úprava samotného LCD. Přesněji jde o doplnění displeje o převodník (expandér) na I2C typu PCF8574. Vzhled využití zobrazovací komponenty si můžete prohlédnout na obrázku 10 společně s expandérem na sériové rozhraní (obrázek 11) a jeho blokovým zapojením (obrázek 12) [19], [20] a [21].

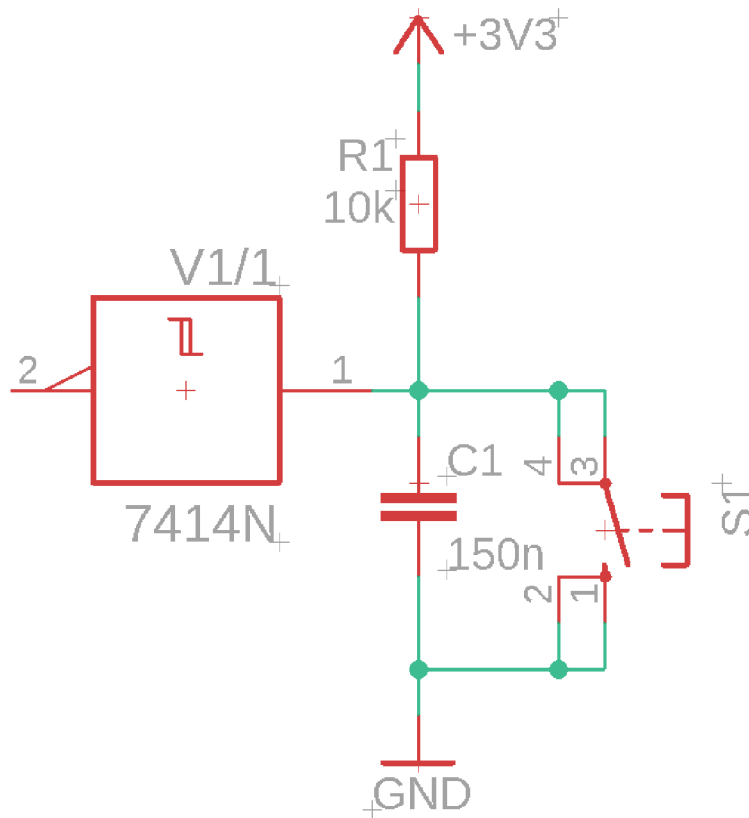


Obrázek 10: LCD display [29]



Obrázek 11: I2C expandér [30]





Obrázek 13: Zapojení tlačítka se Schmittovým klopným obvodem [44]

### 2.3 Výběr vhodného DAB přijímače pro Raspberry Pi

Na různých stránkách lze najít DAB přijímače, které se od sebe liší parametry a především cenou. Nejznámější jsou nejspíš: uGreen DAB Board, MonkeyBoard DAB receiver a RTL-SDR Blog V3.

Modul od firmy uGreen využívá chip Si4684, který zvládá přijímat DAB, FM a má podporu I2S (Inter-IC Sound). Deska obsahuje 40 pinový konektor, který je kompatibilní s Raspberry Pi 1 B+, Pi 2, Pi 3, Pi 4 B, Pi Zero [22].

MonkeyBoard DAB receiver je navrhovaný podle normy ETSI EN 300 401 s DAB/DAB+ citlivostí -99 dBm a FM citlivostí -106 dBm. Nabízí podporu I2S stejně jako deska uGreen [23].

RTL-SDR klíčenka (dongle) byla původně konstruována pro příjem DVB-T HDTV. Obsahuje 8-bitový A/D převodník RTL2832U. Rozmezí přijímaných frekvencí je 500 kHz – 1766 MHz a šířka pásma až 2,4 MHz [24].

Při výběru bylo zohledněno více aspektů než jen kvalita modulu a jeho citlivost, ale i rozsah popisu v dokumentaci a informace k jednotlivým přijímačům, složitost implementace, typ připojení k Raspberry Pi a především cena modulu. S ohledem na tyto parametry byl vybrán USB přijímač RTL-SDR, který je zobrazen na obrázku 14.



Obrázek 14: Náhled RLT-SDR klíčenky [24]

## 2.4 Výběr DAB aplikace

Naprogramování celého DAB přijímače bez využití cizích knihoven není možno realizovat v takto krátkém časovém období, proto je nutné jiné řešení. Našel jsem dvě možnosti, první je, že využijeme programové prostředí GNU radio a propojíme příslušné bloky. Druhou možností je již využít vytvořenou aplikaci s free softwarovou licencí. Rozhodl jsem se, použít již vytvořenou aplikaci Qt-DAB-3.6, jejíž tvůrce je Jan van Katwijk. Jedná se o aplikaci, která je primárně určena na platformy Linux a Raspberry Pi. Jako optimální se mi jevila aplikace dab-mini, která je doporučena pro Raspberry Pi a má minimální grafické rozhraní. Tato aplikace má velice jednoduché a intuitivní ovládání. Skládá se z listu stanic, které lze přepínat tlačítkem v grafickém rozhraní nebo přímým kliknutím na název stanice. Pro výběr kanálů je možné také použít tlačítko stejně jako pro přepnutí stanice nebo rozvinout list kanálů a vybrat kliknutím na příslušnou volbu. V aplikaci je také možné vytvořit seznam oblíbených stanic nebo například ztlumit zvuk, k čemuž slouží tlačítko „mute“. Příslušné grafické rozhraní je zobrazeno na obrázku 15 [25]. Instalace této aplikace je velmi jednoduchá. Postačí následovat instrukce pro instalaci na Raspberry Pi uvedené v [25].





Obrázek 15: GUI aplikace dab-mini [25]



Obrázek 16: Testovací zapojení komponent

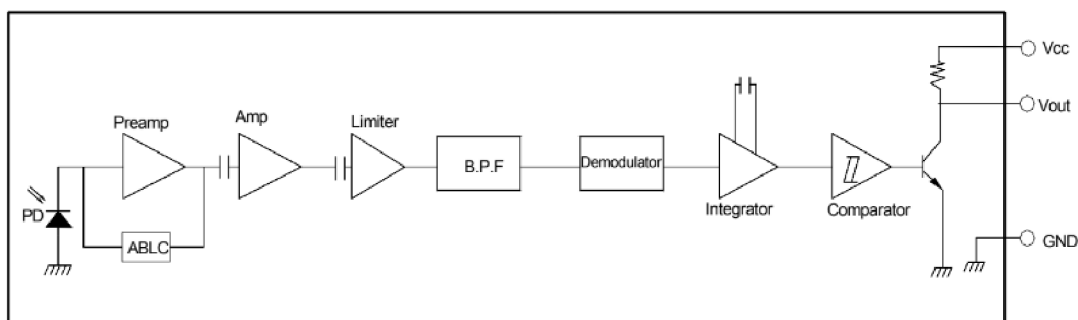
Na obrázku 16 je zobrazeno reálné zapojení všech potřebných komponent k přijímači. Platforma Raspberry Pi se nachází vlevo a je do ní zapojeno několik dalších zařízení (myš, klávesnice a obrazovka). Vpravo na fotografii se nachází USB přijímač, který je zapojen pomocí prodlužovacího kabelu USB tak, aby anténa přijímače dosáhla na potřebná místa. LCD a tlačítko se Schmittovým obvodem na nepájivém poli je připojeno na 40 pinový konektor k Raspberry Pi. Display zobrazuje pouze výstup z testovacího programu, který při stisku tlačítka vyvolá příslušnou funkci pro zobrazení textu „vypis“ a čísla udávajícího počet zmáčknutí tlačítka.

## 2.5 Výběr dálkového ovládání a IR přijímače

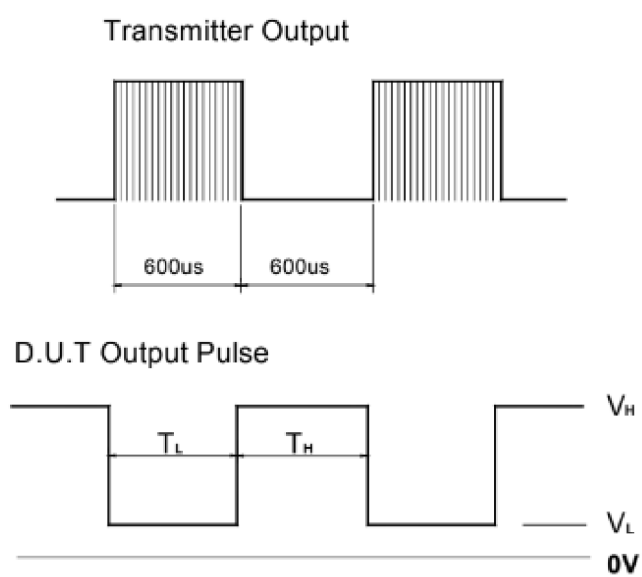
Dálkový ovladač byl vybrán na základě několika požadavků. Plánované rádio by nemělo být moc složité na ovládní, takže ovladač nemusí mít příliš mnoho tlačítek. Požadovaný dosah ovladače nebyl předem definován, ale předpokládané vzdálenosti jsou jednotky metrů. Dálkový ovladač, často používaný pro Arduino, by měl naše základní požadavky dostatečně splňovat. Tento ovladač se pořizuje zároveň i s IR přijímačem. Ovladač pracuje s protokolem NEC, který má nosnou frekvenci 38 kHz. Výrobce uvádí, že maximální vzdálenost ovladače od IR přijímače je 8 metrů, ale tato hodnota je závislá na prostředí a různých rušících faktorech. Typ IR přijímače je OS-1838B a v jednom pouzdře je fotodetektor a předzesilovač. Tento modul je napájen 5 / 3,3 V a je vhodný pro přijímání signálu pomocí protokolu NEC a RC5. Na obrázku 17 si můžete prohlédnout tyto komponenty [35].



Obrázek 17: Dálkové ovládání (vlevo), IR přijímač (vpravo dole) [36]



Obrázek 18: Blokové schéma IR přijímače [35]



Obrázek 19: Časový signál z ovladače (nahore), výstup z IR přijímače (dole) [35]

## 3. REALIZACE DAB PŘIJÍMAČE

Tato kapitola bude pojednávat o konkrétním řešení DAB přijímače pomocí Raspberry Pi 3 B+. Samozřejmě existuje více variant, jak zprovoznit přijímač DAB na jakékoliv platformě pomocí různých komponent a doplňků.

### 3.1 Výběr a instalace OS

Na platformu Raspberry Pi je možné nainstalovat jakýkoliv operační systém, ale nejčastěji se vybírá Raspbian (Buster). Tento operační systém je upravená verze Debianu (Buster) pro RPI. Já jsem tedy zvolil doporučený OS (Raspbian / Debian Buster) s grafickým zobrazením z hlavních stránek Raspbery Pi. Nejrychlejší realizace je přes vytvoření bootovací SD karty a následné vložení do slotu pro SD karu na RPI. Minimální velikost SD karty s ohledem na operační systém a použitý program je 16 GB. Po zapojení RPI do sítě se začne načítat OS a výsledkem je grafické zobrazení plochy a hlavního panelu [37].

### 3.2 Zapojení periferií

#### 3.2.1 Zapojení displeje

Zapojení LCD displeje s I2C expandérem je velmi jednoduché. Zapotřebí jsou pouze 4 propojovací vodiče. Na expandéru jsou označené vývody:

Vcc – napájecí pin 5 V, je nutné jej připojit na druhý pin v RPI pin konektoru.

GND – zemnicí pin, připojen na pin 9 v RPI pin konektoru.

SCL – pin na hodinový signál, zapojený na pin 5 v RPI pin konektoru.

SDA – datový pin, zapojený na pin 3 v RPI pin konektoru.

Po zapojení napájení do RPI by se ve správném případě měl display rozsvítit [21], [38].

#### 3.2.2 Zapojení IR přijímače

Přijímač signálu z dálkového ovládání má 3 piny, z nichž je jeden signálový, který přenáší potřebná data pro dekodování signálu. Tyto vývody jsou označeny následovně:

- (mínus) - toto označení je pro pin, který má být přiveden na GND (na RPI pin 25).

S – signálový (datový) pin, připojen na pin 37 v RPI.

(bez označení) – napájení přijímače 3,3 V, pin 17 na RPI [35].

### 3.2.3 Zapojení ovládacích tlačítek

Každé z šesti tlačítek je zapojené podle schématu na obrázku 13 a celkové schéma zapojení se nachází v příloze (PŘÍLOHA 4). U tlačítek je využité napájení o hodnotě 3,3 V a jsou zapojeny k RPI následovně:

S1 – připojen na pin 16 (GPIO 4), zajišťuje vypnutí RPI.

S2 – připojen na pin 18 (GPIO 5), zobrazí počet stanic v kanále.

S3 – připojen na pin 22 (GPIO 6), přepíná na předchozí kanál.

S4 – připojen na pin 40 (GPIO 29), přepíná na následující kanál.

S5 – připojen na pin 38 (GPIO 28), přepíná na předchozí stanici.

S6 – připojen na pin 36 (GPIO 27), přepíná na následující stanici.

Číslování GPIO pinů je podle [38].

## 3.3 Instalace DAB přehrávače

V kapitole návrhu DAB přijímače je popis aplikace, která využívá GUI zobrazení pro přepínání stanic a kanálu. Po kratší rozpravě s tvůrcem této aplikace mi bylo doporučeno využít pouze základní část (core) tohoto programu. Důvodem byly jednodušší úpravy v hlavního programu a implementace potřebných ovládacích prvků. Tato základní aplikace se jmenuje terminal-DAB a tvůrcem je Jan van Katwijk. Aplikace má velmi jednoduché ovládání, a to přímo v terminálu [39].

### 3.3.1 Instalace knihoven

Pro instalaci těchto knihoven je nutné připojit RPI k internetu. A do terminálu postupně zadat následující příkazy [39].

```
sudo apt-get update
sudo apt-get install git cmake
sudo apt-get install build-essential g++
sudo apt-get install pkg-config
sudo apt-get install libsndfile1-dev
sudo apt-get install libfftw3-dev
sudo apt-get install portaudio19-dev
sudo apt-get install zlib1g-dev
sudo apt-get install libusb-1.0-0-dev
sudo apt-get install libsamplerate0-dev
sudo apt-get install curses
sudo apt-get install libfdk-aac-dev
```

Nyní stáhneme potřebné programy pro DAB stick. Pomocí těchto příkazů zapsaných postupně do terminálu.

```
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr/
```

```
mkdir build  
cd build  
cmake ../-DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON  
make  
sudo make install  
sudo ldconfig  
cd ..  
rm -rf build  
cd ..
```

Pokud není na RPI knihovna wiringPi je nutné ji nainstalovat. Tato knihovna umožňuje práci s GPIO a obsahuje další potřebné funkce na ovládání rádia z ovládacího panelu.

```
sudo apt-get install wiringpi
```

Pokud se správně knihovna nainstaluje lze zobrazit číslování pinů příkazem.

```
gpio readall
```

Tato knihovna umožňuje i práci se sběrnici I2C, ale je nutné použít příkaz (do terminálu).

```
gpio load i2c
```

Dále stáhneme aplikaci terminal-DAB pomocí následujících příkazů.

```
cd /home/pi  
mkdir DAB_OP  
cd DAB_OP  
git clone https://github.com/JvanKatwijk/terminal-DAB-xxx  
cd terminal-DAB-xxx
```

Nyní je zapotřebí zkopírovat text v souboru *CMakeLists.txt* (PŘÍLOHA 1) do stejnojmenného souboru ve složce *terminal-DAB-xxx*. Tento soubor nastavuje některé parametry programu a zajišťuje kompilaci a linkování potřebných programů a knihoven. Veškeré úpravy této aplikace jsem prováděl jen v hlavním programu *main.cpp*, který je také potřeba nahradit stejnojmenným souborem z přílohy (PŘÍLOHA 2). V tomto souboru jsou funkce na ovládání rádia z panelu tlačítek a dálkového ovládání. Po nahrazení těchto souborů je možné pokračovat v instalaci aplikace pomocí následujících příkazů.

```
mkdir build  
cd build  
cmake ../-DRTLSDR=ON  
make
```

```
sudo make install
```

Nyní je aplikace nainstalovaná a připravena na spuštění. Aplikace však potřebuje různé parametry jako například číslo kanálu, proto je potřeba ji spustit přes terminál zadáním následujícího příkazu.

```
terminal -DABOP -rtlsdr -C 12D
```

Tento příkaz zajistí spuštění aplikace s výběrem kanálu 12D a pro přehrávání DAB je potřeba přepnout na další nebo předchozí stanici. To lze udělat pomocí ovládacího panelu, dálkového ovládání (šipka nahoru nebo šipka dolů) a také pomocí klávesnice (klávesa + nebo -) [39].

## 3.4 Popis doplněných funkcí

### 3.4.1 Potřebné knihovny

Jako první je zapotřebí v hlavním programu importovat knihovny, které chceme používat. To se provádí pomocí příkazu *include*. Na obrázku 20 je takováto forma zobrazena [38],[40].

```
68
69 //*****
70 // přidáno pro interrupt + display
71 #include <wiringPiI2C.h>
72 #include <wiringPi.h>
73 #include <stdio.h>
74
75 //*****
76 // přidáno pro IR
77 #include <string.h>
78 #include <errno.h>
79 #include <sys/time.h>
```

Obrázek 20: Import knihoven [38], [40]

### 3.4.2 Definování maker

V programovacím jazyce C++ je možné používat makra. Ta slouží ke zjednodušení čitelnosti programu a ulehčují globální změny určitých parametrů. V programu *main.cpp* jsou potřebná makra k ovládní displeje převzatá od tvůrce webových stránek Lewise Loflina [26] a makra pro dekódování IR signálu od uživatele fóra amfasis [40]. Forma těchto maker je zobrazena na obrázku 21.

```

86 // Define some device parameters
87 #define I2C_ADDR 0x27 // I2C device address
88 // Define some device constants
89 #define LCD_CHR 1 // Mode - Sending data
90 #define LCD_CMD 0 // Mode - Sending command
91 #define LINE1 0x80 // 1st line
92 #define LINE2 0xC0 // 2nd line
93 #define LCD_BACKLIGHT 0x08 // On
94 // LCD_BACKLIGHT = 0x00 # Off
95 #define ENABLE 0b00000100 // Enable bit // pridano pro LCD
96 //*****
97 // přídáno pro IR
98 #define IR_PULSES 35
99 #define TIMEOUT 20000 /* 20 ms timeout value (first pulse should be shorter) */
100 #define BIT1 1000
101 #define IR PIN 25

```

Obrázek 21: Použitá makra [26], [40]

### 3.4.3 Deklarace funkcí

Deklarace funkce uvádí jméno, datový typ a návratové hodnoty funkce. Říká také, že někde dále v programu se nachází i její definice. V programu *main.cpp* jsou deklarace využity pro práci s LCD, IR přijímačem a tlačítky na ovládacím panelu [26], [40].

```

110 void lcd_init(void); // pridano pro LCD
111 void lcd_byte(int bits, int mode);
112 void lcd_toggle_enable(int bits);
113
114 // added by Lewis
115 void typeInt(int i);
116 void typeFloat(float myFloat);
117 void lcdLoc(int line); //move cursor
118 void clrLcd(void); // clr LCD return home
119 void typeLn(const char *s);
120 void typeChar(char val);
121 int fd; // seen by all subroutines
122
123 // přídáno pro IR
124 //*****
125 int pulseIn (int pin, int timeout, long *us_zero, long *us_one);
126 int getIRkey();
127 void IRthread(void);
128 int remotekey;

```

Obrázek 22: Deklarace LCD a IR přijímače [26], [40]



```

273 // pridano pro interrupt
274
275 void Button1(void);
276 void Button2(void);
277 void Button3(void);
278 void Button4(void);
279 void Button5(void);
280 void Button6(void);

```

Obrázek 23: Deklarace tlačítek

### 3.4.4 Definice funkcí

Definice funkcí obsahuje vlastní řešení, tedy co se má stát se vstupními hodnotami (parametry funkce). Forma definice obsahuje datový typ a jméno funkce, po kterých následuje datový typ očekávané vstupní hodnoty (parametr). Následující obrázek 24 uvádí potřebné definice funkcí ke správnému chodu programu.

```

132 // float to string
133 void typeFloat(float myFloat) {
134     char buffer[20];
135     sprintf(buffer, "%.2f", myFloat);
136     typeLn(buffer);
137 }
138
139 // int to string
140 void typeInt(int i) {
141     char array1[20];
142     sprintf(array1, "%d", i);
143     typeLn(array1);
144 }
145
146 // clr lcd go home loc 0x80
147 void ClrLcd(void) {
148     lcd_byte(0x01, LCD_CMD);
149     lcd_byte(0x02, LCD_CMD);
150 }
151
152 // go to location on LCD
153 void lcdLoc(int line) {
154     lcd_byte(line, LCD_CMD);
155 }
156
157 // out char to LCD at current position
158 void typeChar(char val) {
159
160     lcd_byte(val, LCD_CHR);
161 }
162

```

Obrázek 24: Definice LCD funkcí 1 [26]

```

171 void lcd_byte(int bits, int mode) {
172     //Send byte to data pins
173     // bits = the data
174     // mode = 1 for data, 0 for command
175     int bits_high;
176     int bits_low;
177     // uses the two half byte writes to LCD
178     bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT ;
179     bits_low = mode | ((bits << 4) & 0xF0) | LCD_BACKLIGHT ;
180
181     // High bits
182     wiringPiI2CReadReg8(fd, bits_high);
183     lcd_toggle_enable(bits_high);
184
185     // Low bits
186     wiringPiI2CReadReg8(fd, bits_low);
187     lcd_toggle_enable(bits_low);
188 }
189
190 void lcd_toggle_enable(int bits) {
191     // Toggle enable pin on LCD display
192     delayMicroseconds(500);
193     wiringPiI2CReadReg8(fd, (bits | ENABLE));
194     delayMicroseconds(500);
195     wiringPiI2CReadReg8(fd, (bits & ~ENABLE));
196     delayMicroseconds(500);
197 }
198
199 void lcd_init()
200 {
201     // Initialise display
202     lcd_byte(0x33, LCD_CMD); // Initialise
203     lcd_byte(0x32, LCD_CMD); // Initialise
204     lcd_byte(0x06, LCD_CMD); // Cursor move direction
205     lcd_byte(0x0C, LCD_CMD); // 0x0F On, Blink Off
206     lcd_byte(0x28, LCD_CMD); // Data length, number of lines, font size
207     lcd_byte(0x01, LCD_CMD); // Clear display
208     delayMicroseconds(500);
209 }

```

Obrázek 25: Definice LCD funkcí 2 [26], [38]

```

215 int pulseIn(int pin, int timeout, long *us_zero, long *us_one)
216 {
217     long start = micros();
218     *us_zero = 0;
219     *us_one = 0;
220     while (digitalRead(pin) == LOW)
221     {
222         if(micros() - start > timeout)
223             return 0;
224     }
225     *us_one = micros() - start;
226     start = micros();
227     while (digitalRead(pin) == HIGH)
228     {
229         if(micros() - start > timeout)
230             return 0;
231     }
232     *us_zero = micros() - start;
233     return 1;
234 }
235 int getIRkey()
236 {
237     long high[IR_PULSES];
238     long low[IR_PULSES];
239     int i;
240     int result;
241     while(digitalRead(IR_PIN) == HIGH)
242         ;
243     for (i = 0; i < IR_PULSES; i++)
244     {
245         pulseIn(IR_PIN, TIMEOUT, &low[i], &high[i]);
246     }
247     result = 0;
248     for (i = 1; i < IR_PULSES; ++i)
249     {
250         if(low[i] > BIT1)
251             result |= (1 << (32 - i));
252     }
253     return result;
254 }

```

Obrázek 26: Definice IR přijímače [38], [40]

Pomocí příkazů z knihovny wiringPi je potřeba nastavit piny, ke kterým jsou připojeny tlačítka, jako vstupní a dále nastavit obsluhu přerušení od těchto tlačítek. Na obrázku 27 je uveden programový kód, který zajišťuje tyto záležitosti [38].

```

787
788     if (wiringPiSetup () == -1) exit (1);
789
790     fd = wiringPiI2CSetup(I2C_ADDR);
791     wiringPiSetup();
792     pinMode(4, INPUT);
793     pinMode(5, INPUT);
794     pinMode(6, INPUT);
795     pinMode(27, INPUT);
796     pinMode(28, INPUT);
797     pinMode(29, INPUT);
798
799     wiringPiISR(4, INT_EDGE_RISING, Button1);
800     wiringPiISR(5, INT_EDGE_RISING, Button2);
801     wiringPiISR(6, INT_EDGE_RISING, Button3);
802     wiringPiISR(27, INT_EDGE_RISING, Button4);
803     wiringPiISR(28, INT_EDGE_RISING, Button5);
804     wiringPiISR(29, INT_EDGE_RISING, Button6);

```

Obrázek 27: Nastavení pinů a přerušení [38]

U ovládání programu tlačítka je potřeba definovat funkce tlačítek tak, aby se předávala informace pomocí struktury *message*. Tato struktura je definovaná v hlavní programu a předává potřebná data do hlavní smyčky, která kontroluje požadavky na ovládání. Funkce tlačítek jsou volané pomocí přerušení, jelikož každé tlačítko při stisku (z logické 0 na logickou 1) vyvolá přerušení a jeho obsluhu funkcí.

Definovaná funkce pro dálkový ovladač je tvořena smyčkou, kde se dekoduje signál a převede se na číslo. Každé číslo odpovídá jinému tlačítku na ovladači. Ve smyčce je pomocí příkazu *if* vybrán odpovídající kód pro tlačítko a tím i zavolaná příslušná funkce. Je však potřeba, aby smyčka na dekodování signálu z ovladače nějak nezpožďovala hlavní smyčku programu. To lze zajistit rozdělením vláken v procesoru, čímž pracují tyto dvě smyčky paralelně a navzájem se nezpožďují. Rozdělení vláken v procesoru se dělá pomocí příkazu *std::thread „jméno vlákna“ = std::thread(&„název funkce“)*, který je součástí základní knihovny *thread*. Po dokončení požadovaných úkonu pomocí tohoto vlákna je doporučeno opět vlákna sjednotit [45]. To se provede pomocí funkce *join* z téže knihovny. Na obrázcích níže je možné si prohlédnout konkrétní zpracování [39].

---

```

1063 void Button5(void)
1064 {   message m;
1065     int myindex_service;
1066     myindex_service=index_currentService-1;
1067     if (myindex_service == -1)
1068     {
1069         myindex_service= serviceNames.size ()-1;
1070     }
1071     const char *LCD_NAMESERVICE = serviceNames[myindex_service].c_str();
1072     m. key = S_SET_PREVSERVICE;
1073     m. string = "";
1074     messageQueue. push (m);
1075
1076     ClrLcd();
1077     lcdLoc(LINE1);
1078     typeLn(LCD_NAMECHANNEL);
1079     lcdLoc(LINE2);
1080     typeLn(LCD_NAMESERVICE);
1081 }
1082 void Button4(void)
1083 {   message m;
1084     int myindex_service;
1085     myindex_service=index_currentService+1;
1086
1087     if (myindex_service >= serviceNames.size ())
1088     {
1089         myindex_service= 0;
1090     }
1091
1092     const char *LCD_NAMESERVICE = serviceNames[myindex_service].c_str();
1093     m. key = S_SET_NEXTSERVICE;
1094     m. string = "";
1095     messageQueue. push (m);
1096
1097     ClrLcd();
1098     lcdLoc(LINE1);
1099     typeLn(LCD_NAMECHANNEL);
1100     lcdLoc(LINE2);
1101     typeLn(LCD_NAMESERVICE);
1102 }

```

Obrázek 28: Definice funkcí tlačítek 1

```

1103 void Button3(void)
1104 { message m;
1105   m. key = S_SET_PREVCHANNEL;
1106   m. string = "";
1107   messageQueue. push (m);
1108
1109   ClrLcd();
1110   lcdLoc(LINE1);
1111   typeLn(LCD_NAMECHANNEL);
1112 }
1113 void Button6(void)
1114 { message m;
1115   m. key = S_SET_NEXTCHANNEL;
1116   m. string = "";
1117   messageQueue. push (m);
1118
1119   ClrLcd();
1120   lcdLoc(LINE1);
1121   typeLn(LCD_NAMECHANNEL);
1122 }
1123 void Button2(void)
1124 {
1125   int my_services_number;
1126   my_services_number= serviceNames.size ();
1127
1128   ClrLcd();
1129   lcdLoc(LINE1);
1130   typeLn("Services:");
1131   lcdLoc(LINE2);
1132   typeLn("  ");
1133   typeInt(my_services_number);
1134 }
1135 void Button1(void)
1136 {
1137   ClrLcd();
1138   lcdLoc(LINE1);
1139   typeLn("RPI shutdown");
1140   system("poweroff");
1141 }
1142

```

Obrázek 29: Definice funkcí tlačítek 2

```

1198 // funkce pro dekodovani ovladace
1199 void IRthread(void)
1200 {
1201     while(1)
1202     {
1203         remotekey=getIRkey();
1204
1205         if(remotekey == 16730805) // IR controler button down (prev service)
1206         {
1207             Button5();
1208         }
1209
1210         if(remotekey == 16718055) // IR controler button up (next service)
1211         {
1212             Button4();
1213         }
1214
1215         if(remotekey == 16716015) // IR controler button left (prev channel)
1216         {
1217             Button3();
1218         }
1219
1220         if(remotekey == 16734885) // IR controler button right (next channel)
1221         {
1222             Button6();
1223         }
1224
1225         if(remotekey == 16726215) // IR controler button ok (show number of services in channel)
1226         {
1227             Button2();
1228         }
1229
1230         if(remotekey == 16756815) // IR controler button hash (#) (shutdown RPI)
1231         {
1232             Button1();
1233         }
1234     }
1235 }

```

Obrázek 30: Definice funkce dálkového ovladače

Na tomto výpisu z programu (obrázek 30) je názorné přiřazení funkcí ke kódu z dálkového ovládání. Jsou zde využity stejné funkce jako pro tlačítka z ovládacího panelu.

Funkce na tlačítkách dálkového ovládání:

Šipka dolů – přepnutí na předchozí stanici

Šipka nahoru – přepnutí na následující stanici

Šipka vlevo – přepnutí na předchozí kanál

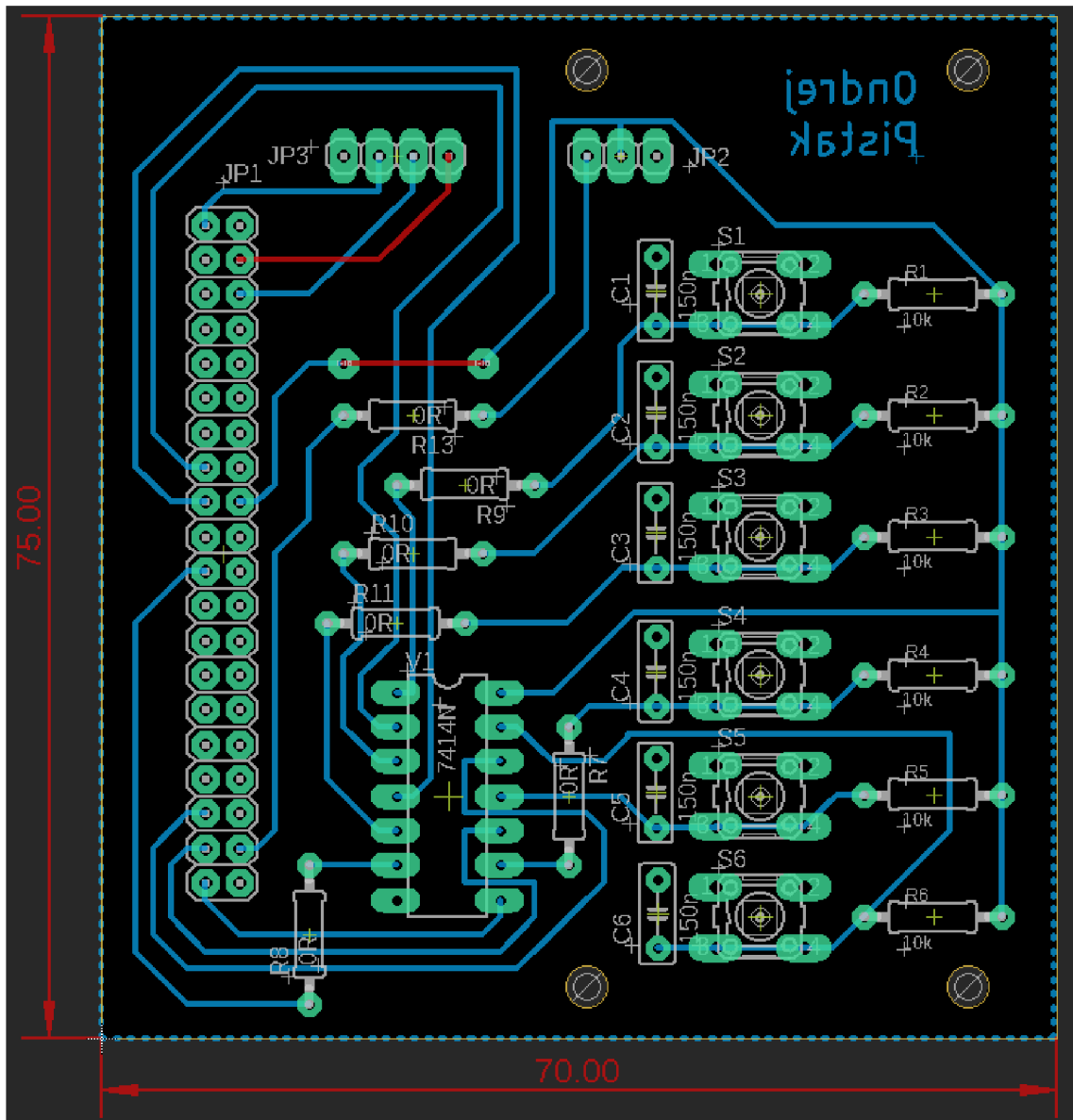
Šipka vpravo – přepnutí na následující kanál

OK (prostřední tlačítko) – zjištění počtu stanic v kanále

# – vypnutí Raspberry Pi

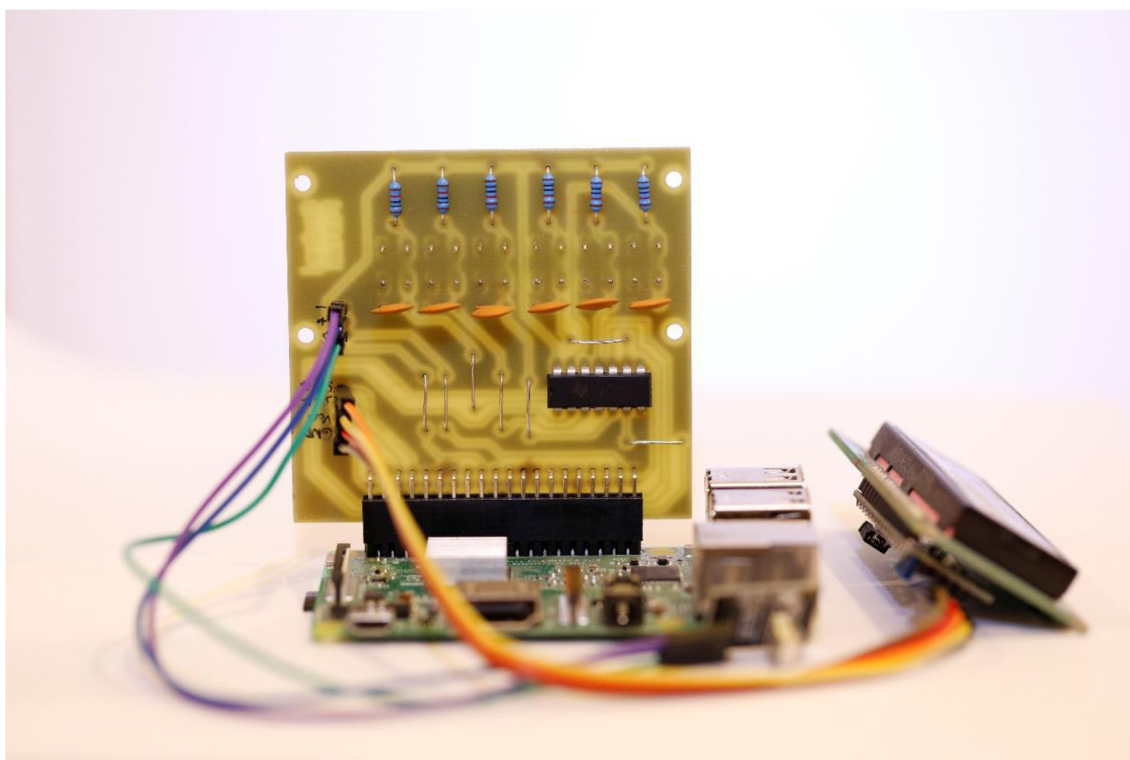
### 3.5 Deska plošných spojů

Deska je navržena tak, aby se dala mechanicky připojit kolmo k desce RPI. Toto zapojení zajistí 90° dutinková lišta napájená na desce s tlačítky. Celkový návrh desky je jednostranný, ale ze strany spojů jsou napájena tlačítka, aby bylo možné dobře navrhnout krabičku. Deska tedy obsahuje dutinkovou lištu, integrovaný Schmittův klopný obvod, kondenzátory, rezistory, propojky, tlačítka a pin konektory, na které se připojí IR přijímač a LCD. Návrh desky byl vytvořen v prostředí Eagle 9.6.1. Výrobu desky zajistila dílna Ústavu radioelektroniky. Návrh i výsledná realizace desky jsou zobrazeny níže na obrázcích 31, 32 a 33.

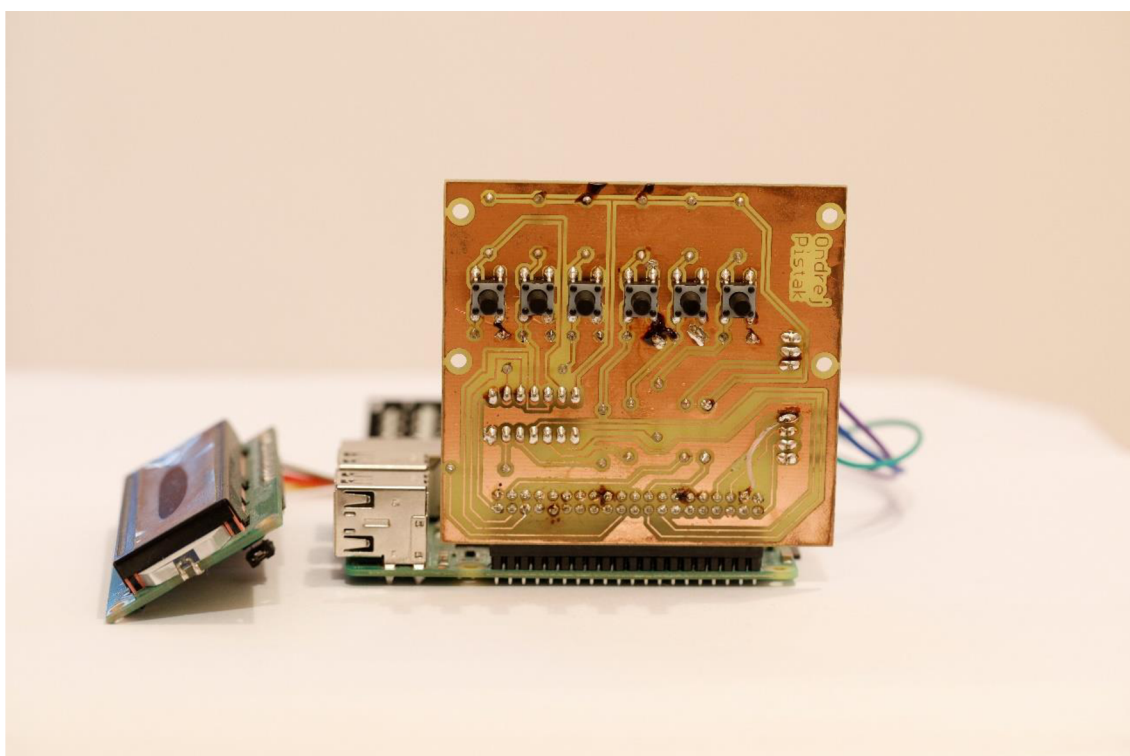


Obrázek 31: Návrh desky plošných spojů s tlačítky (pohled ze strany součástek)





Obrázek 32: Výsledné zapojení a deska ze strany součástek



Obrázek 33: Výsledné zapojení a deska ze strany spojů

## 3.6 Automatické spuštění po zapnutí

Nyní jsme ve fázi, kdy by mělo rádio fungovat a je možné ho ovládat pomocí tlačítek na desce nebo dálkového ovládání. V dalším kroku, který zajišťuje automatické zapnutí programu po načtení operačního systému RPI, je nutné zasáhnout do souboru systému v RPI. Je možné že tento zásah (při špatném provedení) způsobí vážné chyby v systému, proto je nutné systém zálohovat na jinou SD kartu. Zálohování se provádí pomocí externí čtečky (SD karta - USB) a aplikaci SD Card Copier, která je v základním vybavení systému (Raspbian buster) RPI. Pokud je systém zálohován na jiné SD kartě je možné pokračovat v této poslední části. Jako první vytvoříme script, který zapíše do terminálu příkaz s parametrem kanálu na spuštění programu pro přehrávání DAB. A tento script budeme následně spouštět při načítání operačního systému [41], [42].

### 3.6.1 Vytvoření scriptu

Vytvoříme textový soubor (na ploše) *DAB\_runner.sh*. Do tohoto souboru se zapíše na první řádek `#!/bin/bash` a na další řádek příkaz s parametrem pro spuštění přehrávání DAB `terminal-DABOP-rtlsdr -C 12D` a soubor uložíme a zavřeme. Momentálně máme textový soubor, který ovšem nejde spustit. Pro spustitelný soubor je potřeba přiřadit mu správná přístupová práva. To se provede příkazem zadaným do terminálu:

```
sudo chmod +x /home/pi/Desktop/DAB_runner.sh
```

Nyní by mělo být možné soubor spustit [41].

### 3.6.2 Automatické spuštění

Pro automatické spuštění je třeba zasáhnout do souboru *autostart*, který využívá RPI pro automatické spuštění grafických programů. Do terminálu tedy zadáme příkaz na změnu tohoto souboru `sudo nano /etc/xdg/lxsession//LXDE-pi/autostart`, kde `sudo` znamená práva admina a `nano` zápis do souboru (zobrazení v terminále). Na poslední řádek je potřeba zapsat spuštění předem vytvořeného skriptu *DAB\_runner.sh*. Tento příkaz vypadá následovně: `@lxterminal -e bash /home/pi/Desktop/DAB_runner.sh`, kde `lxterminal -e` zajišťuje spuštění v terminálu a `@` znamená, že se při nepovedeném spuštění tohoto skriptu provede opakované spuštění. Po doplnění tohoto příkazu do souboru *autostart* můžeme soubor uložit a zavřít. Po restartování RPI by se po načtení operačního systému měl spustit program na přehrávání DAB [42].

## 4. ZÁVĚR

Tato bakalářská práce se zabývá jednou z možností, jak vytvořit DAB přijímač za použití tuneru RTL-SDR Blog V3 a platformy Raspberry Pi.

V první části, tedy teoretické, je popsán princip vzniku signálu DAB. Veškeré detailní informace jsou stanoveny normou ČSN ETSI EN 300 401. Z blokového schématu je zřejmé, že zpracování signálu je v podstatě inverzní k jeho vytvoření. Tyto teoretické znalosti nám dále pomohly s návrhem (výběrem) vhodné aplikace.

Část, která se zaměřuje na návrh obsahuje teoretickou konstrukci a výběr komponent. Jako první v této kapitole se zabýváme výběrem vhodného LCD displeje. Prioritou pro tuto volbu byla podpora komunikačního protokolu I2C, který redukuje počet obsazených pinů GPIO na Raspberry Pi. Při práci s displejem je zapotřebí naprogramovat nebo najít vhodné knihovny (popřípadě funkce), které umí komunikovat s tímto LCD pomocí I2C. Funkce, které jsou využity v testovacím programu a později i v programu terminal-DAB pro zobrazení na display napsal Lewis Loflin [26]. Výběr aplikace, která signál zpracovává byla výrazně ovlivněna počtem projektů nalezených na různých fórech, internetových stránkách a jejich dokumentací. Realizace zpracování signálu v prostředí GNU Radio by byla velmi zajímavá a více názorná, bohužel jsem neměl dostatek času na bližší prozkoumání možností této aplikace. Vybraná aplikaci, kterou rádio využívá je terminal-DAB od tvůrce Jana van Katwijkka [39]. Tuto aplikaci jsem doplnil o funkce pro ovládání pomocí tlačítek a dálkového ovládání. Funkce, které dekodují IR signál jsou od uživatele fóra amfasis [40].

Vytvořené DAB rádio splňuje všechny zadané požadavky. Lze jej ovládat prostřednictvím ovládacího panelu (pomocí tlačítek) a také pomocí IR dálkového ovladače. Po načtení rádia jsou potřebné informace o stanici (číslo kanálu, název stanice) zobrazeny na LCD.

## LITERATURA

- [1] HOEG, Wolfgang a Thomas LAUTERBACH. *Digital Audio Broadcasting: Principles and Applications of Digital Radio* [online]. Second Edition. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons, 2003 [cit. 2020-12-06]. ISBN ISBN 0-470-85013-2. Dostupné z: [http://maxwell.sze.hu/~ungert/Radiorendszerek\\_satlab/Segedanyagok/Ajanlott\\_ir\\_odalom/DAB.pdf](http://maxwell.sze.hu/~ungert/Radiorendszerek_satlab/Segedanyagok/Ajanlott_ir_odalom/DAB.pdf)
- [2] ČSN ETSI EN 300 401 V2.1.1: *Systémy rozhlasového vysílání – Digitální rozhlasové vysílání (DAB) pro pohyblivé, přenosné a pevné přijímače*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2017.
- [3] PĚCHA, Jakub. *Generátor signálů DAB v Matlabu* [online]. Pardubice, 2017 [cit. 2020-12-06]. Dostupné z: [https://dk.upce.cz/bitstream/handle/10195/71859/PechaJ\\_GeneratorSignalu\\_JP\\_2018.pdf?sequence=2&isAllowed=y](https://dk.upce.cz/bitstream/handle/10195/71859/PechaJ_GeneratorSignalu_JP_2018.pdf?sequence=2&isAllowed=y) Diplomová práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky. Vedoucí práce Jan Pidanič.
- [4] Hradňanský, V. *Multimódový VKV rozhlasový přijímač* [online]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2016. 100 s., 18 s. příloh. [cit. 2020-12-06]. Dostupné z: <http://mtt.ieee.cz/studentska-soutez/soutez2016/MMSE.pdf>. Diplomová práce. Vedoucí práce: doc. Ing. Jiří Šebesta, Ph.D.
- [5] ETSI TS 102 563 V2.1.1. *Digital Audio Broadcasting (DAB); DAB+ audio coding (MPEG HE-AACv2)* [online]. Sophia Antipolis Cedex- FRANCE: European Telecommunications Standards Institute, 2017 [cit. 2020-12-06]. Dostupné z: [https://www.etsi.org/deliver/etsi\\_ts/102500\\_102599/102563/02.01.01\\_60/ts\\_102563v020101p.pdf](https://www.etsi.org/deliver/etsi_ts/102500_102599/102563/02.01.01_60/ts_102563v020101p.pdf)
- [6] MIKULÁŠTÍK, Karel. *Perspektivy digitálního rozhlasu T-DAB v České republice* [online]. 2009 [cit. 2020-12-06]. Dostupné z: <https://www.lupa.cz/clanky/perspektivy-digitalniho-rozhlasu-t-dab-v-cesku/>
- [7] *Digital Audio Broadcasting* [online]. 2020 [cit. 2020-12-06]. Dostupné z: [https://cs.wikipedia.org/wiki/Digital\\_Audio\\_Broadcasting](https://cs.wikipedia.org/wiki/Digital_Audio_Broadcasting)
- [8] *Advanced Audio Coding* [online]. 2018 [cit. 2020-12-06]. Dostupné z: [https://cs.wikipedia.org/wiki/Advanced\\_Audio\\_Coding](https://cs.wikipedia.org/wiki/Advanced_Audio_Coding)
- [9] VLČEK, Jiří. *Aktuální stav digitálního rozhlasového vysílání v Evropě* [online]. Ostrava, 2010 [cit. 2020-12-06]. Dostupné z: [https://dspace.vsb.cz/bitstream/handle/10084/78730/VLC155\\_FEI\\_B2647\\_2612R059\\_2010.pdf](https://dspace.vsb.cz/bitstream/handle/10084/78730/VLC155_FEI_B2647_2612R059_2010.pdf). Bakalářská práce. Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky. Vedoucí práce Marek Dvorský.
- [10] *Orthogonal frequency-division multiplexing* [online]. 2020 [cit. 2020-12-06]. Dostupné z: [https://en.wikipedia.org/wiki/Orthogonal\\_frequency-division\\_multiplexing](https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing)

- [11] PROKEŠ, Aleš. *RÁDIOVÉ KOMUNIKAČNÍ SYSTÉMY* [online]. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2013 [cit. 2020-12-06]. Dostupné z: [https://moodle.vutbr.cz/pluginfile.php/240820/mod\\_resource/content/1/RKS%20s%20kripta.pdf](https://moodle.vutbr.cz/pluginfile.php/240820/mod_resource/content/1/RKS%20s%20kripta.pdf)
- [12] HOJKA, Jiří, Jiří BOLTÍK a Jiří NOBILIS. *Radioelektronická zařízení I: učební text pro SPŠ s výukou studijního oboru 26-68-6 Elektronická a sdělovací zařízení*. Praha: Státní nakladatelství technické literatury, 1986.
- [13] *Stabilizovaný zdroj* [online]. 2020 [cit. 2020-12-07]. Dostupné z: [https://cs.wikipedia.org/wiki/Stabilizovan%C3%BD\\_zdroj](https://cs.wikipedia.org/wiki/Stabilizovan%C3%BD_zdroj)
- [14] *Transformátor* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://cs.wikipedia.org/wiki/Transform%C3%A1tor>
- [15] *Stabilizátor napětí* [online]. 2020 [cit. 2020-12-07]. Dostupné z: [https://cs.wikipedia.org/wiki/Stabiliz%C3%A1tor\\_nap%C4%9Bt%C3%AD](https://cs.wikipedia.org/wiki/Stabiliz%C3%A1tor_nap%C4%9Bt%C3%AD)
- [16] *Usměrnovač* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://cs.wikipedia.org/wiki/Usm%C4%9Br%C5%88ova%C4%8D>
- [17] *Návrh stejnosměrného zdroje napětí* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <http://www.elweb.cz/clanky.php?clanek=65>
- [18] *LM7805* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://www.ti.com/>
- [19] *Displej z tekutých krystalů* [online]. 2020 [cit. 2020-12-11]. Dostupné z: [https://cs.wikipedia.org/wiki/Displej\\_z\\_tekut%C3%BDch\\_krystal%C5%AF](https://cs.wikipedia.org/wiki/Displej_z_tekut%C3%BDch_krystal%C5%AF)
- [20] *Specification for LCD Module 1602A* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://www.openhacks.com/uploads/productos/eone-1602a1.pdf>
- [21] *PCF8574 Remote 8-bit I/O expander for I2C* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/18215/PHILIPS/PCF8574T.html>
- [22] *uGreen DAB Board* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://ugreen.eu/>
- [23] *The DABMonkey* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://www.monkeyboard.org/>
- [24] *RTL-SDR Blog V3 Datasheet* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://www.rtl-sdr.com/wp-content/uploads/2018/02/RTL-SDR-Blog-V3-Datasheet.pdf>
- [25] *Qt-DAB* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <https://github.com/JvanKatwijk/qt-dab>
- [26] *I2C LCD to Raspberry Pi in C* [online]. 2020 [cit. 2020-12-11]. Dostupné z: <http://www.bristolwatch.com/rpi/i2clcd.htm>
- [27] *Using OFDM signals in wireless communication* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://www.mathworks.com/discovery/ofdm.html>
- [28] *about  $\pi/4$  DQPSK modulation* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <http://i56578-swl.blogspot.com/2015/09/about-4-dqpsk-modulation-here-used-in.html>

- [29] *16x2 LCD Display* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://www.hobbytronics.co.uk/lcd-16-2-backlight-blue>
- [30] *LCD Sériové rozhraní IIC/I2C/TWI/SP 1602* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://arduino-shop.cz/arduino/1071-lcd-seriove-rozhrani-iic-i2c-twi-sp-1602.html>
- [31] *Dálkové ovládání* [online]. 2021 [cit. 2021-24-05]. Dostupné z: [https://cs.wikipedia.org/wiki/D%C3%A1lkov%C3%A9\\_ovl%C3%A1d%C3%A1n%C3%AD](https://cs.wikipedia.org/wiki/D%C3%A1lkov%C3%A9_ovl%C3%A1d%C3%A1n%C3%AD)
- [32] *Dálkové ovládání* [online]. 2021 [cit. 2021-24-05]. Dostupné z: [https://en.wikipedia.org/wiki/Remote\\_control](https://en.wikipedia.org/wiki/Remote_control)
- [33] *Dálkové ovládání TV přijímače* [online]. 2021 [cit. 2021-24-05]. Dostupné z: [https://moodle.vutbr.cz/pluginfile.php/239090/mod\\_resource/content/1/TVT%20uloha%2001.pdf](https://moodle.vutbr.cz/pluginfile.php/239090/mod_resource/content/1/TVT%20uloha%2001.pdf)
- [34] *Protokol NEC* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>
- [35] *IR přijímač* [online]. 2021 [cit. 2021-24-05]. Dostupné z: [https://www.hwkitchen.cz/user/related\\_files/infrared-prijimac-1838b.pdf](https://www.hwkitchen.cz/user/related_files/infrared-prijimac-1838b.pdf)
- [36] *Dálkové ovládání* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <https://soyoo.com/wp-content/uploads/2018/08/ira.jpg>
- [37] *Raspberry Pi OS* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <https://www.raspberrypi.org/software/>
- [38] *Knihovna wiringPi* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <http://wiringpi.com/>
- [39] *DAB program* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <https://github.com/JvanKatwijk/>
- [40] *Dekódování IR signálu* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?t=9207>
- [41] *Spustitelný soubor* [online]. 2015 [cit. 2021-24-05]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?t=104909>
- [42] *Automatické spuštění* [online]. 2020 [cit. 2021-24-05]. Dostupné z: <https://www.youtube.com/watch?v=zeB5TimDNj8>
- [43] *Digitální rozhlas DAB+* [online]. 2021 [cit. 2021-24-05]. Dostupné z: <http://www.digitalradiodab.cz/>
- [44] *Hardware debounce* [online]. 2021 [cit. 2021-24-02]. Dostupné z: <https://www.logiswitch.net/switch-bounce-debounce-tutorial/method-4-hardware-debounce-for-spst-switches>
- [45] *Multi-threading* [online]. 2020 [cit. 2021-24-02]. Dostupné z: <https://www.cplusplus.com/reference/thread/thread/>