

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Fakulta Elektrotechnická

Bezdrátový mikrofon napojený do Wi-Fi sítě

květen 2014

Vypracoval: Kryštof Čermák

Vedoucí práce: doc. Ing. Leoš Boháč, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 23. 5. 2014

.....

podpis

Poděkování

Chtěl bych poděkovat zejména doc. Ing. Leoši Boháčovi, Ph.D. za vedení této práce a jeho hodnotné připomínky. Poděkování patří také mojí rodině, která mě podporovala po celou dobu mého studia i při psaní této bakalářské práce.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Čermák Kryštof**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síťové a informační technologie

Název tématu: **Bezdrátový mikrofon napojený do Wi-Fi sítě**

Pokyny pro vypracování:

Navrhněte a realizujte bezdrátový mikrofon, který bude napájen z baterie. Snímaný akustický signál bude za mikrofonem digitalizován, pakelizován a poslán do sítě na server nebo multicastem k více cílům. Zařízení musí být co nejmenší a musí co nejvíce šetřit baterie.

Seznam odborné literatury:

[1] Eady, F.: Implementing 802.11 with Microcontrollers: Wireless Networking for Embedded Systems Designers. Newnes 2005. ISBN 978-075067-865-0.

Vedoucí: doc. Ing. Leoš Boháč, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Boris Šimák, CSc.
vedoucí katedry




prof. Ing. Pavel Řipka, CSc.
děkan

V Praze dne 9. 12. 2013

Anotace:

Tato bakalářská práce se zabývá návrhem bezdrátového mikrofonu napojeného do Wi-Fi sítě a jeho realizací podle zadaných parametrů a s tím spojených problémů implementace na konkrétním řešení. Dále jsou zde popsány standardy a principy použité při realizaci a podrobněji je popsán operační systém OpenWrt.

Klíčová slova:

Mikrofon, Audio, Wi-Fi, OpenWrt

Summary:

This bachelor thesis describes the design of a wireless microphone connected to Wi-Fi network and its implementation on the basis of the given parameters and the problems connected to specific solutions of implementation. It further describes the standards and principles used in the implementation and further describes operating system OpenWrt.

Index Terms:

Microphone, Audio, Wi-Fi, OpenWrt

Obsah:

1. ÚVOD	1
2. ROZBOR ZADÁNÍ.....	2
2.1. VSTUPNÍ MĚNIČ.....	2
2.2 DIGITALIZACE SIGNÁLU	3
2.2.1 Telekomunikační standardy.....	4
2.2.2 Kompresní formáty.....	4
2.3 PAKETIZACE.....	5
2.3.1 Referenční modely.....	6
2.3.2 TCP protokol.....	6
2.3.3 UDP protokol.....	7
2.2.4 RTP protokol.....	7
2.2.5 Multicast.....	8
2.2.6 Negativní efekty přenosu zvuku paketovou sítí.....	8
2.4 STANDARD IEEE 802.11.....	9
3. NÁVRH ŘEŠENÍ.....	10
3.1 PRŮZKUM TRHU	10
3.2 SCÉNÁŘE POUŽITÍ.....	10
3.2.1 Mikrofon jako přístupový bod	10
3.2.2 Mikrofon jako klient	11
4. HARDWARE.....	14
4.1 ŘÍDÍCÍ ČIP.....	14
4.1.1 Mikrokontrolér	14
4.1.2 SoC.....	14
4.1.3 Rozhraní	14
4.1.4 Výběr cílového zařízení	15
4.2 NAPÁJENÍ	16
4.3 ZVUKOVÁ KARTA.....	17
5. OPENWRT	18
5.1 O SYSTÉMU	18
5.1.1 Historie.....	18
5.1.2 Vlastnosti systému	18
5.1.3 Hardwarové požadavky.....	18
5.1.4 Licence	18
5.2 SOFTWARE VYBAVENÍ	19
5.2.1 Struktura.....	19
5.2.2 Balíčkovací systém	19
5.2.3 Flash layout.....	19
5.2.4 Konfigurační nástroje	20
5.2.5 CLI.....	21
5.2.6 Busybox	21
5.2.7 OSS.....	21

5.2.8 Další nástroje.....	22
6. SOFTWARE	23
6.1 KONFIGURACE OPENWRT	23
6.1.1 První připojení	23
6.1.2 Konfigurace síťového připojení	23
6.1.3 Konfigurace Wi-Fi.....	24
6.1.4 Extroot.....	24
6.1.5 Sysupgrade	25
6.1.6 Audio	25
6.1.7 Další software	25
6.1.8 Failsafe.....	25
6.2 SHELL	26
6.2.1 Skripty v mikrofonu	26
6.2.2 Skripty v přijímači	27
6.3 WEBOVÉ ROZHŘANÍ	27
6.3.1 Bootstrap.....	27
6.3.2 Ovládací prvky	28
6.4 SLEPÉ VÝVOJOVÉ CESTY	29
6.5 PROBLÉMY VÝSLEDNÉ REALIZACE	30
7. ZÁVĚR	31
8. LITERATURA	32
8.1 SEZNAM POUŽITÉ LITERATURY.....	32
8.2. SEZNAM OBRÁZKŮ	34
8.3. SEZNAM TABULEK.....	35
8.4 SEZNAM POUŽITÝCH ZKRATEK	36
PŘÍLOHY	37
OBSAH PŘILOŽENÉHO CD.....	38

1. Úvod

V této práci se budu zabývat popisem procesu realizace bezdrátového mikrofonu připojeného do Wi-Fi sítě. Od rozboru dostupných platforem přes implementaci vybraných dílčích platforem a problémů s tím spojených. Základním a určujícím prvkem této práce je zadaný standard bezdrátové komunikace nazvaný Wi-Fi, který pod sebou slučuje celou řadu standardů vydaných organizací IEEE pod označením 802.11.

Jelikož jsou Wi-Fi sítě v dnešní době nejrozšířenějším volně dostupným řešením připojení koncového klienta do sítě internet tak se naskytla možnost připojení zařízení denní potřeby, které může usnadnit jejich ovládání, umožnit jejich automatizaci nebo je využít ke zcela novým účelům. Tento trend byl pojmenován jako „Internet of Things“ neboli volně přeloženo jako „Internet věcí“. Do této kategorie spadá i tento bezdrátový mikrofon připojený do Wi-Fi sítě.

Při zpracování tohoto zadání jsem se snažil dosáhnout co nejvíce univerzálního a modulárního zařízení které bude zároveň uživatelsky přívětivé a které najde využití ve všedních situacích, jako nahrazení analogového bezdrátového mikrofonu, nebo i jako dílčí nástroj pro řešení komplexnějších problémů.

Cílem této práce je realizovat co nejmenší, baterií napájený mikrofon, který bude pomocí bezdrátové technologie Wi-Fi připojen do sítě kam bude vysílat zaznamenaný akustický signál určený k reprodukci nebo dalšímu zpracování.

2. Rozbor zadání

Přenos zvukového signálu pomocí IP sítě začíná v místě zdroje akustického signálu, nebo jeho blízkém okolí, kde je umístěn mikrofon. Analogový signál je navzorkován pomocí A/D převodníku a dále zpracováván. Dochází zde k použití kodeku, který příchozí signál upraví, zkomprimuje a přetransformuje do datového toku. Kodek může být implementován pomocí samostatného integrovaného čipu, nebo pouze jako softwarový ovladač.

V dalším bloku dochází k transformaci datového toku do jednotlivých paketů. Tento proces se ve většině případů zajišťuje podle standardizovaného protokolu zajišťujícího, že jednotlivé pakety bude možné u příjemce složit dohromady a poskládat z nich ideálně stejný datový tok jako na vysílací straně.

Následně jsou pakety odeslány do IP sítě. V některých případech může stačit komunikace pouze v lokální síti, avšak některé aplikace mohou vyžadovat připojení do sítě internet - například pro zpracování, uložení datového toku v „cloudu“ nebo pro prostou hlasovou komunikaci napříč internetem s jedním nebo více uživateli.



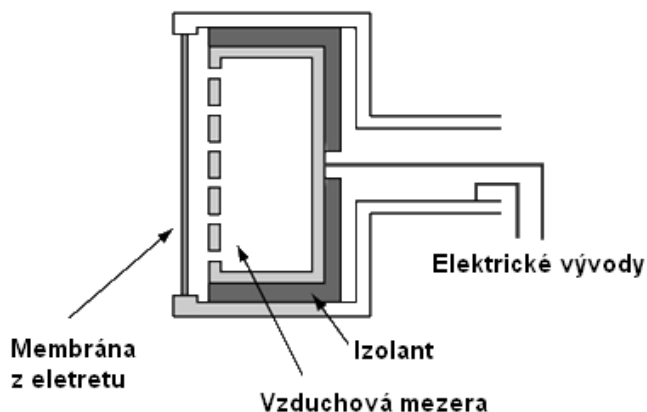
Obrázek 1 – Obecné blokové schéma přenosu akustického signálu

2.1. Vstupní měnič

Na začátku celého řetězce se nachází zdroj akustického signálu, zvuku. Tento signál charakterizujeme jako mechanické kmitání, ze kterého je pro nás podstatné pásmo přibližně 16 Hz až 20 kHz, které můžeme nazývat jako „slyšitelný zvuk“. Tento signál je snímán mechanicko-elektrickým měničem, mikrofonom, který slyšitelné pásmo přemění na elektrický signál.

Pro přeměnu akustického signálu na elektrický se využívá několika různých fyzikálních principů. Nejrozšířenějším z nich je elektrostatický princip.

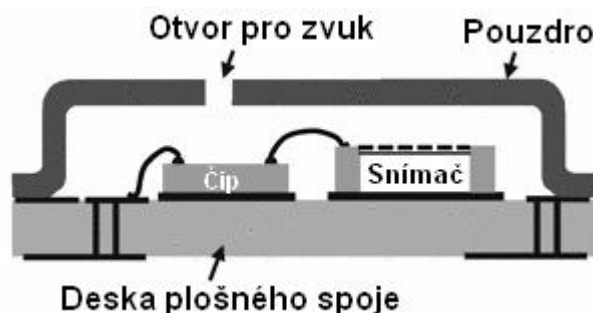
Tento princip využívá vlastnosti deskového kondenzátoru, kdy se při změně vzdáleností desek mění kapacita kondenzátoru. U elektrostatického mikrofону slouží jedna elektroda jako pohyblivá membrána a druhá elektroda je statická. K této elektrodě může být připojeno stejnosměrné polarizované napětí nebo může být pokryta speciálním dielektrickým materiálem nesoucím permanentní náboj. Tento materiál se nazývá elektret a mikrofony využívající tento princip jsou označovány jako elektretové. [2]



Obrázek 2 - Průřez elektretovým mikrofonem [1]

Elektrostatického principu také často využívají mikrofony na bázi MEMS neboli Micro Electro-Mechanical Systems což jsou miniaturní mikrofony s integrovanou elektronikou v pouzdru o ploše několika mm². [3] To je velkou výhodou při další integraci do co nejmenšího zařízení.

Tento typ mikrofonu v sobě může využívat více typů snímání akustického tlaku, avšak nejčastěji využívá kapacitního principu. Dále v sobě může integrovat další pomocné obvody, nebo i A/D převodník a výstupní data už mohou být čistě digitální.



Obrázek 3 - Architektura MEMS mikrofonu [3].

2.2 Digitalizace signálu

V dalším bloku následujícím po mikrofonu je blok převádějící vstupní analogový signál na digitální tok dat. Tento proces může být proveden, v případě použití jednoduššího mikrokontroléru, na nízké úrovni pomocí A/D převodníků zpracování vlastním programem nebo použitím integrovaného enkodéru, který bývá většinou přidružen k dekodéru. Tento celek se nazývá kodek a může synchronně pracovat obousměrně.

Základní dělení kodeků je na ztrátové a bezztrátové podle druhu komprese signálu. V telekomunikační technice se zpravidla používají ztrátové komprese z důvodu minimalizace datového toku.

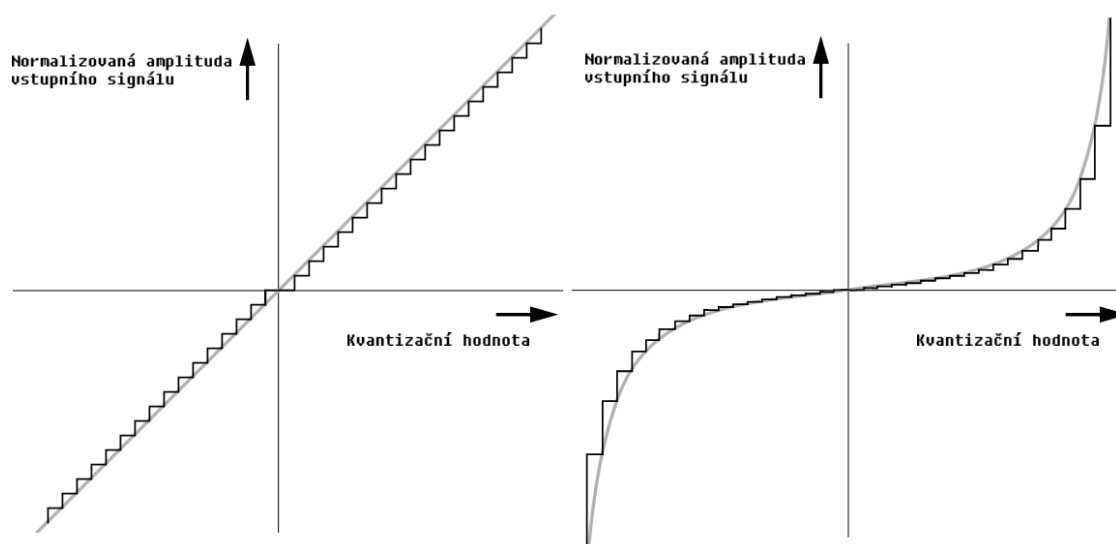
Pro datovou komunikaci kodeky používají sběrnice I²S, celým názvem Inter-IC Sound definované firmou Philips Semiconductor už v roce 1986. Ta je navržena pro zvuková zařízení. [4]

Mezi další rozhraní, které se používají pro komunikaci s kodeky, je rozhraní USB, celým názvem Universal serial bus.

2.2.1 Telekomunikační standardy

Některé standardy pro kódování hlasového a audio signálu nalezneme v doporučeních ITU-T G.710 až G.729. Mezi běžně používaný standard v telekomunikačních sítích patří PCM definovaný v G.711. Tento standard je používán zejména díky své jednoduchosti a minimálním nárokům na výpočetní výkon díky absenci další komprimace. Jeho přenosová rychlost je 64 kbit/s.

U standardu G.711 se používá vzorkovací frekvence 8 kHz, která vyplývá ze vzorkovacího teoremu a šířky pásma. Tato vzorkovací frekvence stačí pro přenos hlasového signálu, avšak pro jiné zvuky může být tento rozsah limitující. Jednotlivé vzorky mají 8bitovou hloubku a vznikají pomocí pulsně kódové modulace s použitím nelineární kvantizační křivky, která je přizpůsobena nelinearitě sluchového vnímání a tím snižuje kvantizační zkreslení vznikající ze „zaokrouhlování“ velikosti vzorků.



Obrázek 4 - Srovnání lineární a nelineární (μ -law) kvantizační křivky. [5]

Standard G.711 existuje ve dvou verzích, μ -law pro Severní Ameriku a Japonsko, a a-law pro zbytek světa. Liší se pouze tvarem kvantizační křivky, která je u μ -law složena z logaritmických křivek a u a-law je jejich tangentou. [6]

Pro přenos kvalitního zvukového záznamu je však nutné použít vyšší přenosové rychlosti. I přes zvyšující se přenosovou kapacitou IP sítě je nutné použít kompresi pro snížení požadavků na přenosovou rychlost, aby bylo možné používat mikrofon například i s využitím pomalejší mobilní datové sítě.

2.2.2 Kompresní formáty

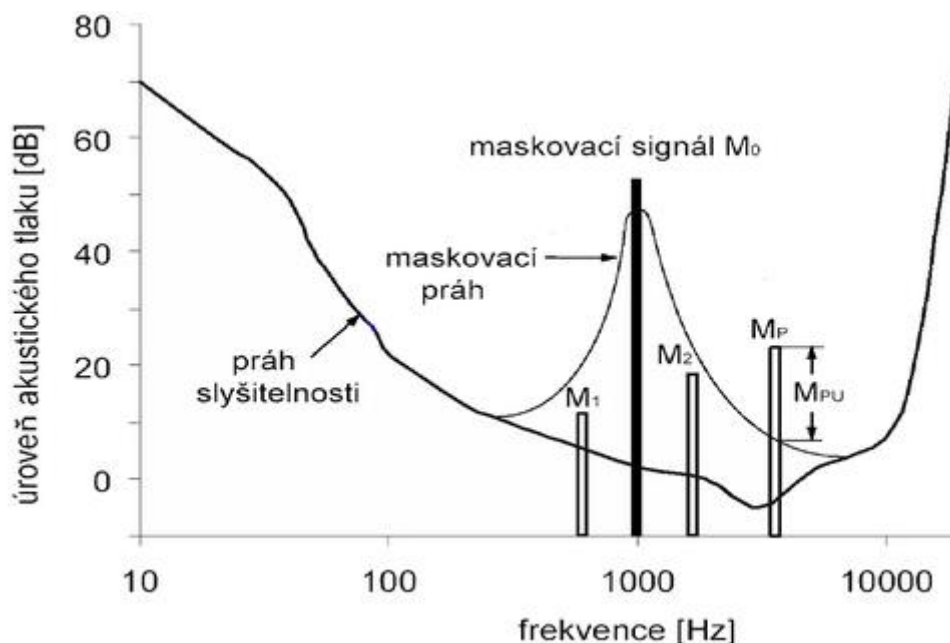
V následujících odstavcích rozeberu přehled nejznámějších kompresních audio formátů vhodných pro přenos audio signálu.

MP3

Kompresní ztrátový formát MP3, celým názvem MPEG-1, nebo MPEG-2 Audio Layer III byl publikován už v roce 1993 organizací Moving Pictures Experts Group.

MP3 existuje v několika revizích podporujících vzorkovací frekvence od 8 kHz do 48 kHz a datový tok pohybující se do 320 kbit/s. [7]

Formát MP3 využívá pro kompresi psychoakustického modelu, na jehož základě byl vytvořeno tzv. časové a frekvenční maskování které využívají vlastností lidského ucha a vnímání zvuku.



Obrázek 5 – Frekvenční maskování [8]

Vorbis

Formát Vorbis je na rozdíl od formátu MP3 otevřený formát. Dosahuje vyšších zvukových kvalit při stejné bitové rychlosti, ovšem za cenu potřeby vyššího výpočetního výkonu.[9]

Ztrátový kompresní formát Vorbis se používá v kombinaci s kontejnerovým formátem OGG. Tento formát je určený pro manipulaci a „stream“ jak audia, tak videa.

Formát OGG podporuje i bezztrátový kompresní audio formát FLAC který dosahuje kompresního poměru okolo 50% originálního CD [10]. U ztrátových audio formátů se tato hodnota okolo 10%.

Nevýhodou kompresních formátů při použití pro přenos v reálném čase je jejich výpočetní náročnost a přidání zpoždění, které vzniká jejich zpracováním.

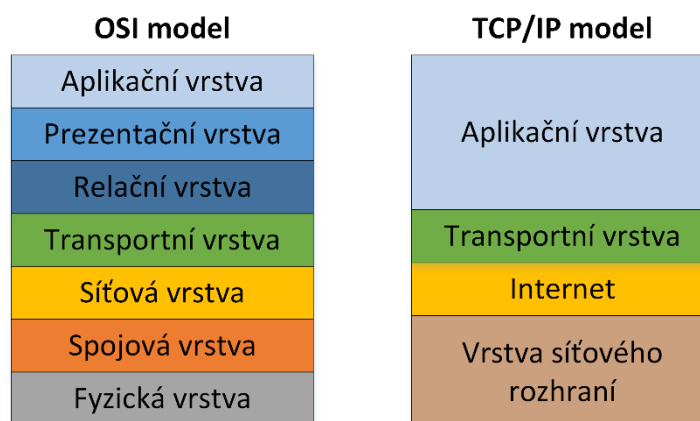
2.3 Paketizace

Po digitalizaci signálu dostaneme na výstupu datový tok, který je potřeba dále rozdělit do paketů a poslat sítí na určené místo. Aby bylo možné využít stávající infrastrukturu IP sítě, tak musíme použít standardizované síťové komunikační protokoly. Tyto protokoly předem definují, jakým způsobem budou jednotlivé prvky komunikovat. Internetové protokoly pracují na různých vrstvách referenčních modelů počítačových sítí.

2.3.1 Referenční modely

Základní referenční model normalizoval Mezinárodní normalizační úřad (ISO) pod názvem ISO OSI model. Tento model popisuje architekturu komunikace na sedmi vrstvách od aplikační vrstvy až po fyzickou vrstvu.

Dalším referenčním modelem je TCP/IP model popisující komunikaci na čtyřech vrstvách. Tento model je oproti OSI modelu jednodušší a vykazuje větší flexibilitu.



Obrázek 6 – OSI a TCP/IP model

Oba tyto modely mají velmi podobný popis transportní vrstvy, která je klíčovou vrstvou pro spolehlivost a rychlost komunikace, což jsou podstatné parametry pro přenos zvukového signálu.

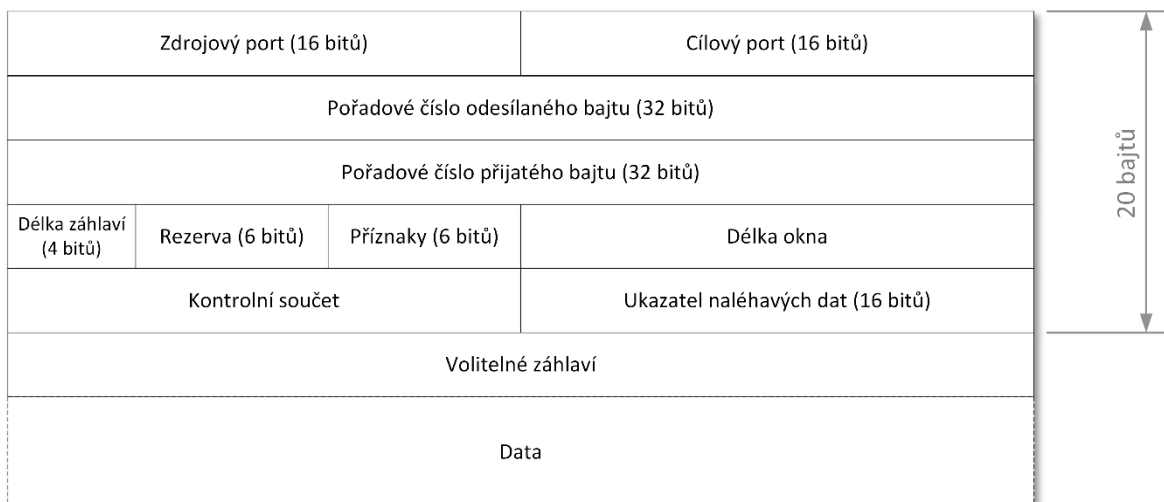
Tato vrstva přejímá vstupní data od aplikace a dělí je do menších částí, datagramů nebo segmentů a po přidání záhlaví je předává další vrstvě. V opačném směru tato vrstva skládá příchozí data zpět do datového toku.

Na této vrstvě jsou definovány dva hlavní komunikační protokoly a to TCP a UDP protokol. Tyto protokoly zajišťují spojení mezi aplikacemi běžícími na počítačích připojených v síti. Tyto procesy se adresují pomocí čísla portu. Číslo portu je 16 bitové číslo v rozsahu 0 až 65535. Pro adresaci v síti se toto číslo portu používá v kombinaci s IP adresou. [11]

2.3.2 TCP protokol

TCP protokol, celým názvem Transmission Control Protocol, je spojovanou službou kde se nejprve vytvoří plně duplexní spojení mezi serverem a klientem a poté se potvrzuje přijetí dat mezi oběma účastníky komunikace a v případě přijetí poškozených dat se tato data vyžádají znovu. Tato vlastnost zaručuje doručení všech dat a jejich integritu. [12]

Z těchto vlastností vychází struktura záhlaví, která se přidává ke všem datům převzatým z vyšší vrstvy, které se po spojení s uživatelskými daty nazývají TCP segment. Tento segment poté dále předává nižší vrstvě.



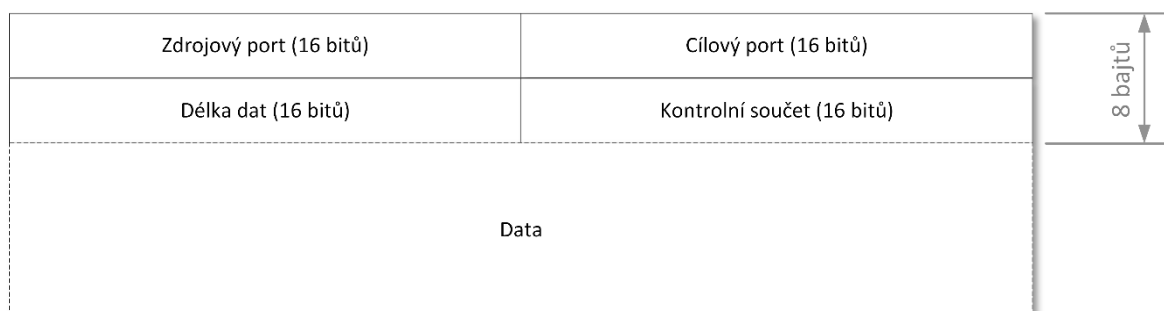
Obrázek 7– TCP záhlaví [11]

2.3.3 UDP protokol

UDP protokol, celým názvem User Datagram Protocol, je protokol, který na rozdíl od protokolu TCP, nenavazuje spojení před přenášením dat a ani neprobíhá potvrzení přijetí jednotlivých rámců. Protokol UDP pouze odešle rámeček dat, nazvaný datagram, a nezajišťuje další řízení přenosu dat, o které se případně musí starat vyšší vrstva. Z tohoto důvodu je protokol UDP rychlejší než TCP, ale není tak spolehlivý – data nemusí dorazit do cíle nebo mohou dorazit v nesprávném pořadí.

Protokolu UDP nalezne využití v aplikacích, kde hraje velkou roli rychlost přenosu. Jedná se zejména o přenos multimediálního obsahu v reálném čase, ale využívá ho i protokol DNS nebo DHCP.

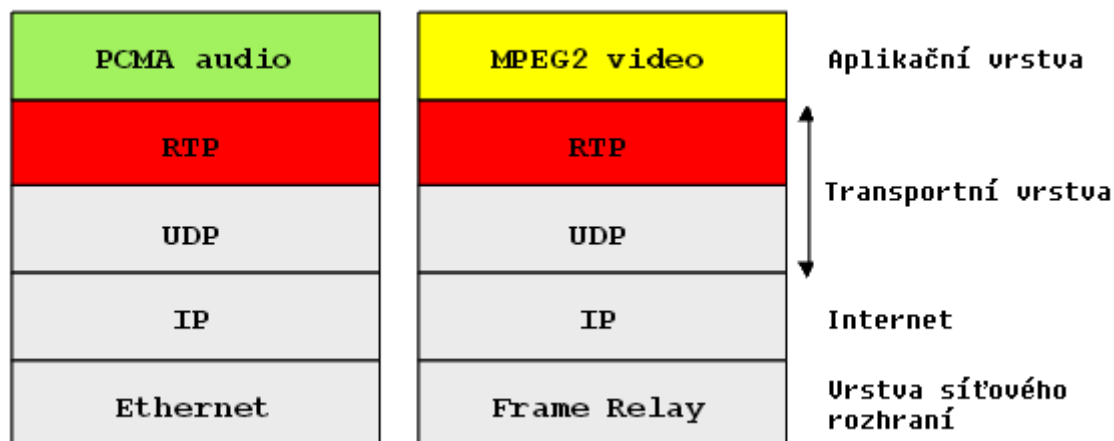
Záhlaví protokolu UDP je velké pouze 8 bajtů a kromě zdrojového a cílového portu, údaje o délce dat obsahuje také nepovinný kontrolní součet.



Obrázek 8- UDP záhlaví [11]

2.2.4 RTP protokol

Real time protocol je protokolově neutrální a flexibilní standard pro přenos multimediálního obsahu v reálném čase. Na transportní vrstvě může používat například protokol UDP.



Obrázek 9 – Umístění RTP protokolu v TCP/IP modelu [16]

K protokolu RTP je také přidružen kontrolní protokol RTCP – RTP Control Protocol, který se využívá ke zpětné vazbě, synchronizaci mezi audiem a videem a dalším účelům jako úprava přenosové rychlosti, detekce kolizí a smyček nebo ukončení spojení. [14]

2.2.5 Multicast

Multicast je komunikační metoda umožňující komunikaci v dané skupině uživatelských stanic. Oproti broadcastu, který přeposílá data všem uživatelům zároveň a unicastu, který umožňuje komunikovat pouze mezi dvěma uživateli, multicast umožňuje efektivně využít omezenou šířku přenosové cesty mezi serverem odesílajícím data a multicastovou skupinou jenž tato data přijímá.

Multicast využívá IP adresy v rozsahu 224.0.0.0 až 239.255.255:255, z něhož je rozsah 239.255.0.0/16 vyhrazen pro lokální síť. Pro přiřazení do multicastové skupiny se využívá protokol IGMP

Nevýhodou multicastu je, že síťová infrastruktura ho musí podporovat, a to jak na 3. vrstvě, v podobě podpory ve směrovačích, tak na 2. vrstvě na síťových přepínačích [15].

2.2.6 Negativní efekty přenosu zvuku paketovou sítí

Datová síť použitá k přenosu paketů obsahujících zvuková data, má oproti sítím s komutací okruhů, svoje nevýhody, které jsou pro přenos dat v reálném čase velkou výzvou.

Zpoždění

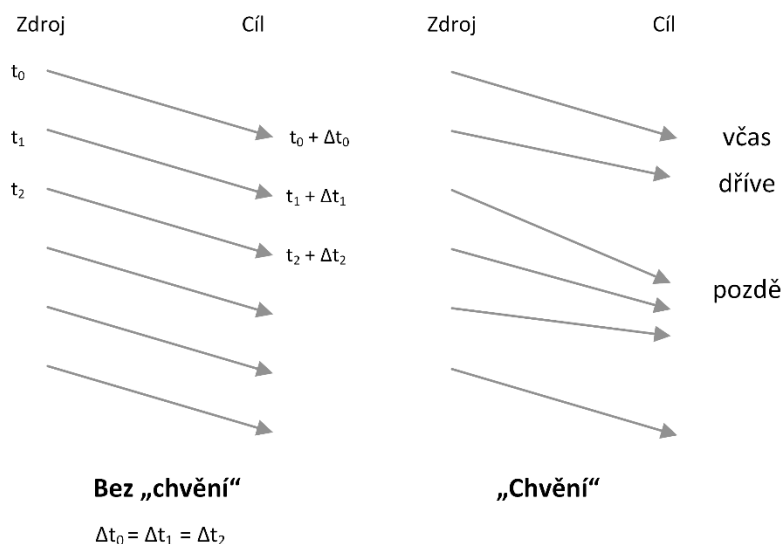
V paketových sítích vzniká větší přenosová prodleva, protože v každém průchozím prvku musí být zpracována hlavička paketu a ten musí být nadále směrován dál.

Ztráta paketů

Protože přenosy v reálném čase si zpravidla nemohou zažádat o opětovný přenos ztraceného nebo poškozeného paketu, tak může docházet k výpadkům signálu. Tyto výpadky mohou mít ve větší zřetelný dopad na kvalitu přenášeného zvuku.

Jitter - „chvění“

Paket se v datových sítích s komutací paketů může šířit odlišnými cestami a tím pádem i s nestejnými dobami zpoždění. Což může negativním způsobem ovlivňovat reprodukci přenášeného signálu. Tento efekt lze odstranit použitím protokolů jako je RTP který v každém paketu obsahuje tzv. timestamp, časový údaj, který v kombinaci s vyrovnávací pamětí dokáže tento efekt potlačit. [16]



Obrázek 10 - Jitter neboli „chvění“ signálu [16]

2.4 Standard IEEE 802.11

Tento standard pro bezdrátovou komunikaci vyvinula organizace IEEE - Institute of Electrical and Electronics Engineers. Původní standard 802.11 byl vydán v 1997 a od té doby byla vydána řada doplňků, která tento standard rozšiřují.[18] Pro moji práci jsou důležité pouze tři doplňky - *b*, *g* a *n*. Právě tyto doplňky podporuje většina dostupných zařízení.

Standard	Přenosový mechanismus	Frekvence	Podporovaná přenosová rychlost [Mbit/s]
802.11b	DSSS	2,4 GHz	1; 2; 5,5; 11
802.11g	OFDM,DSSS	2,4 GHz	6; 9; 12; 18; 24; 36; 48; 54
802.11n	OFDM	2,4 / 5 GHz	7,2 - 150

Obrázek 11 – Přehled Wi-Fi standardů [17]

3. Návrh řešení

3.1 Průzkum trhu

Před tím, než jsem začal se samotným návrhem, jsem se pokusil udělat průzkum trhu a srovnání produktů a zařízení s podobnými kritérii. Hlavním kritériem pro zařazení do srovnání byla podpora standardu Wi-Fi, dále následovaly parametry jako provoz na baterii, kompaktní rozměry a samozřejmě přítomnost mikrofonu, nebo možnost připojení externího mikrofonu.

Po důkladném prohledání lokálních i globálních prodejců, dodavatelů a výrobců, jsem došel k závěru, že podobný výrobek na trhu není. U zařízení, která nejlépe splňovala zadané parametry, jsem se často setkal se klamavým popisem „wifi“, ale při důkladném prozkoumání jsem zjistil, že používají pro přenos pouze shodné frekvenční pásmo 2,4 GHz, ale nejedná se o standard IEEE 802:11.

Dalším typem zařízení byla zařízení pro „stream“ audia i videa, čemuž však odpovídala zvýšená cena, rozměry a zejména spotřeba. Tyto zařízení byla vybavena nejčastěji proprietárním systémem nebo mobilním operačním systémem Android.

V tomto segmentu je tedy vidět velká mezera na trhu, která by při doladění produktu mohla být snadno využita.

3.2 Scénáře použití

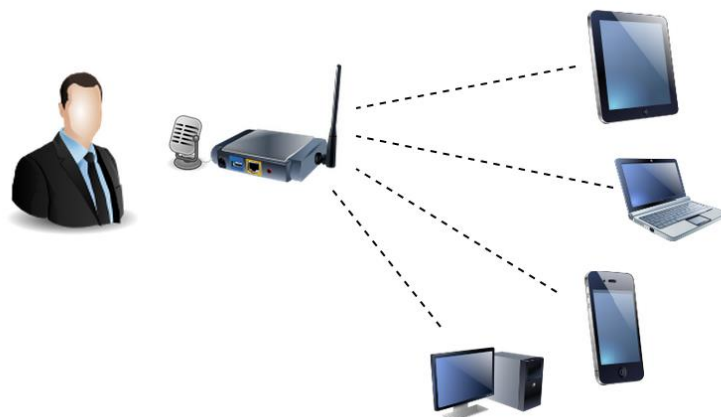
Při návrhu tohoto zařízení jsem si nejprve musel ujasnit, jak bude samotné zařízení pracovat v síti. Nabízí se více možností, jak lze zařadit bezdrátový mikrofon do sítě.

V této kapitole budou rozebrány různé varianty napojení bezdrátového mikrofonu do sítě při implementaci s použitím technologie Wi-Fi. A dále možnosti komunikace v různých typech sítí.

3.2.1 Mikrofon jako přístupový bod

V tomto režimu funguje „mikrofon“ jako Wi-Fi přístupový bod. Jednotliví klienti se k němu připojí a pomocí webové aplikace nebo specifické aplikace si spustí předem definovaný přenos zvuku. V závislosti na zátěži sítě a počtu klientů může být využito přenosu pomocí unicastového vysílání nebo v případě většího zatížení se uživatelé připojí multicastového vysílání.

Webová aplikace je v tomto scénáři provozována na http serveru, který běží na „mikrofonu“. Pro větší uživatelský komfort může být využito lokální úpravy DNS serveru, který také běží na „mikrofonu“, a odpovídá na předem známé doménové jméno adresou, na kterém běží webová ovládací aplikace. Odpadne tím zadávání adresy a dojde ke zjednodušení a zrychlení i pro běžného uživatele.



Obrázek 12 - Wi-Fi mikrofon jako přístupový bod

Při použití tohoto řešení se dále nabízí možnost, že samotný „mikrofon“ bude připojen např. pomocí Ethernetu do sítě internet a může všem připojeným klientům poskytovat připojení k internetu. V tomto případě by „mikrofon“ sloužil jako Wi-Fi směrovač a mikrofon s http serverem by na něm běželi pouze jako postranní aplikace. Nevýhodou tohoto řešení je umístění mikrofonu v prostoru omezené připojeným kabelem. Dalším problémem by mohla být kvalita přenosu zvuku při vyšším zatížení směrovače. Bylo by nutné implementovat QoS nebo nějak jinak omezit rychlost pro ostatní datovou komunikaci.

Toto řešení by mohlo najít uplatnění například na konferencích pro vícejazyčný simultánní překlad, kdy by každý posluchač nemusel mít specializovaný přijímač, ale mohl by použít vlastní notebook, tablet nebo smartphone.

3.2.2 Mikrofon jako klient

V tomto případě je „mikrofon“ připojen na již existující infrastrukturu a komunikuje buď na lokální síti nebo „streamuje“ i mimo ní. Hlavní otázkou zůstává konfigurace připojení do stávající Wi-Fi sítě. Konfigurace závisí na tom, jakým uživatelským rozhraním je „mikrofon“ vybaven. V případě, že chceme dosáhnout co nejmenších rozměrů a spotřeby energie tak není na ovládací prvky místo. Konfigurace tedy musí proběhnout přes Wi-Fi.

Konfigurace může být provedena následujícím způsobem:

1. „Mikrofon“ vytvoří přístupový bod a uživatel se připojí pomocí osobního počítače, nebo pomocí telefonu vybaveného Wi-Fi.
2. Uživatel se přihlásí stejně jako v prvním případě, připojí se na webové rozhraní a spustí skenování okolních Wi-Fi sítí poté vybere požadovanou síť a vyplní přihlašovací údaje.
3. „Mikrofon“ se po dokončení konfigurace restartuje a přepne do klientského režimu s danými přístupovými údaji.
4. Při dalším spuštění dříve než přejde do konfiguračního módu tak zjistí známé Wi-Fi sítě v dosahu a podle předem dané logiky se připojí.

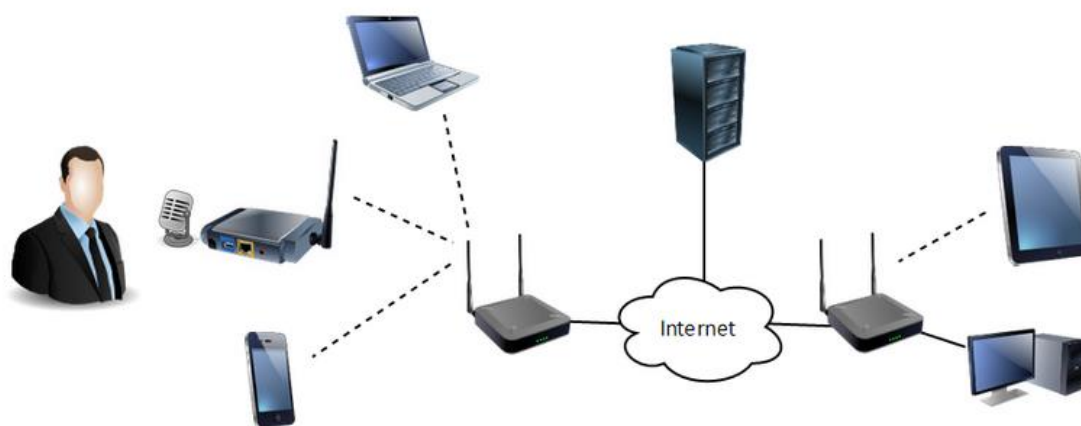
Mezi další možnost snadného připojení „mikrofonu“ do sítě je technologie WPS, neboli „Wi-Fi protected setup“, kde jednou z možností připojení klienta k přístupovému bodu je připojení stiskem tlačítka na Wi-Fi směrovači. Nevýhodou tohoto řešení je, že technologie WPS je dostupná pouze u části Wi-Fi směrovačů, které jsou na trhu dostupné.

Po připojení „mikrofonu“ do sítě se nabízí více možností, jak komunikovat s koncovými zařízeními do kterých bude posílán digitalizovaný audio signál.

Uživatelé v lokální síti budou moc přistupovat unicastem přímo k „mikrofonu“

Mikrofon bude v místní síti propagován pomocí audio serveru, jako je například Pulseaudio, který umožní standardizovaným protokolem připojení více posluchačů pomocí protokolu RTP.

Mikrofon bude odesílat zvuk na server, umístěný v síti internet, který bude tento „stream“ dále redistribuovat pomocí multicastu.



Obrázek 13 - Wi-Fi mikrofon jako klient

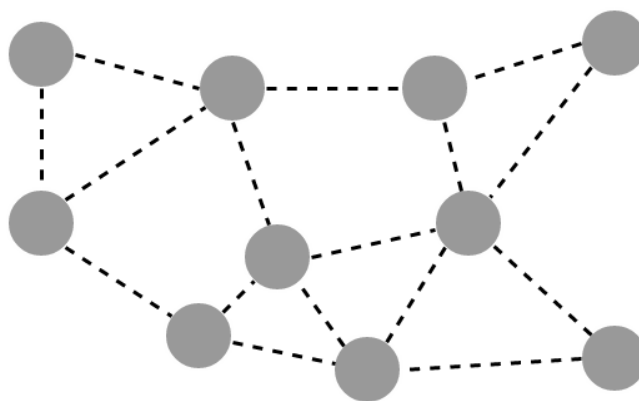
Mikrofon v síti s polygonální topologií

V polygonální topologii jsou nejčastěji propojeny sousední jednotky. Této vlastnosti se využívá zejména k dosažení většího pokrytí a v případě výpadku, ke komunikaci alternativní trasou.

Polygonální topologie je nativní topologie technologie ZigBee, definované ve standartu 802.15.4, ale existuje i specifikace pro Wi-Fi standard IEEE 802.11 pod označením 802.11s.

Standard IEEE 802.11s má oproti ad hoc režimu IEEE 802.11g velkou výhodu v tom, že všechny stanice nemusí být v dosahu. Umožňuje tedy při stejném počtu klientů větší rozestupy a celkově je lépe navržen lépe pro vytvoření infrastruktury. Bohužel však tento standard nebyl plošně implementován. Jeho implementace je možná dodatečně pouze na linuxových systémech.

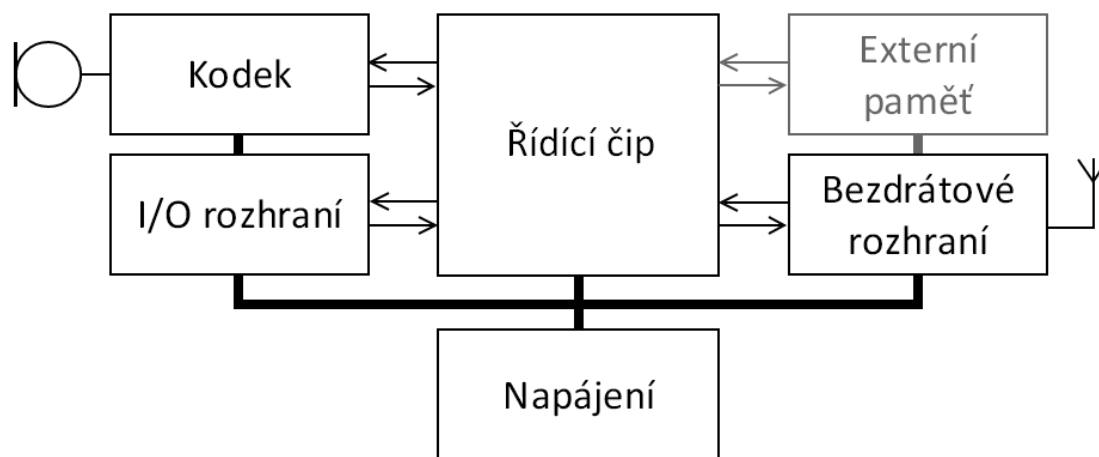
Standard ZigBee vytváří síť například pomocí zařízení ve funkci koordinátoru, které dále určuje používané kanály a další parametry. Pro připojení k síti IP musíme však použít další zařízení – gateway, neboli výchozí bránu.



Obrázek 14 – Polygonální topologie

4. Hardware

Před výběrem finální platformy k realizaci je třeba si určit požadavky a priority. K tomuto účelu poslouží základní blokové schéma a rozbor jeho dílčích součástí.



Obrázek 15 – Blokové schéma bezdrátového mikrofonu

4.1 Řídicí čip

Hlavním prvkem celé řídicí struktury bývá zpravidla mikrokontrolér integrující v sobě mikroprocesor, paměť a další, například komunikační, rozhraní.

4.1.1 Mikrokontrolér

Mikrokontroléry rozdělujeme podle mnoha parametrů. Pro tzv. embedded (vestavěné) systémy se nejčastěji používají 8,16 nebo 32bitové mikrokontroléry, jejichž výkon je dále určen frekvencí a velikostí a rychlostí RAM paměti. Dalším limitujícím parametrem je velikost interní programové paměti (většinou technologie Flash) určené pro program. U mikrokontrolérů se složitějším programovým vybavením nebo operačním systémem je často také přítomna externí programová a operační paměť.

4.1.2 SoC

Jako pokročilejší alternativa mikrokontrolérů se jeví kategorie integrovaných obvodů nazvaných SoC (System on a Chip) neboli Systém na čipu. Hlavním rozdílem bývá řádově vyšší výkon a integrace speciálních bloků, jako může být například Wi-Fi modul.

4.1.3 Rozhraní

Pro připojení externího příslušenství je důležitým parametrem počet vstupně-výstupních pinů. Jako nejčastěji používané komunikační rozhraní se také integrují rozhraní:

- USART, SPI, USB (sériová rozhraní)
- I²C, I²S (sériové sběrnice rozhraní)
- Ethernet

4.1.4 Výběr cílového zařízení

Po zvážení všech potřebných parametrů bylo vybráno zařízení TP-LINK WR703N, které nejlépe splnilo požadovaný profil. Toto zařízení je určeno jako přenosný Wi-Fi směrovač s možností připojení USB modemu pro sítě třetí a čtvrté generace. Výrobce je dodáván s proprietárním firmware, který má velmi omezené funkce, ale lze ho snadno nahradit operačním systémem OpenWrt s nepřebírnými možnostmi. Svou oblibu si tento směrovač získal hlavně z důvodu nízké ceny, která se pohybuje okolo 22 US dolarů. V české distribuci bohužel není, ale lze nalézt alternativy jako je model MR3020, který obsahuje téměř stejný hardware od stejného výrobce pouze ve větším pouzdru.

Mezi jeho klíčové vlastnosti obou těchto zařízení patří:

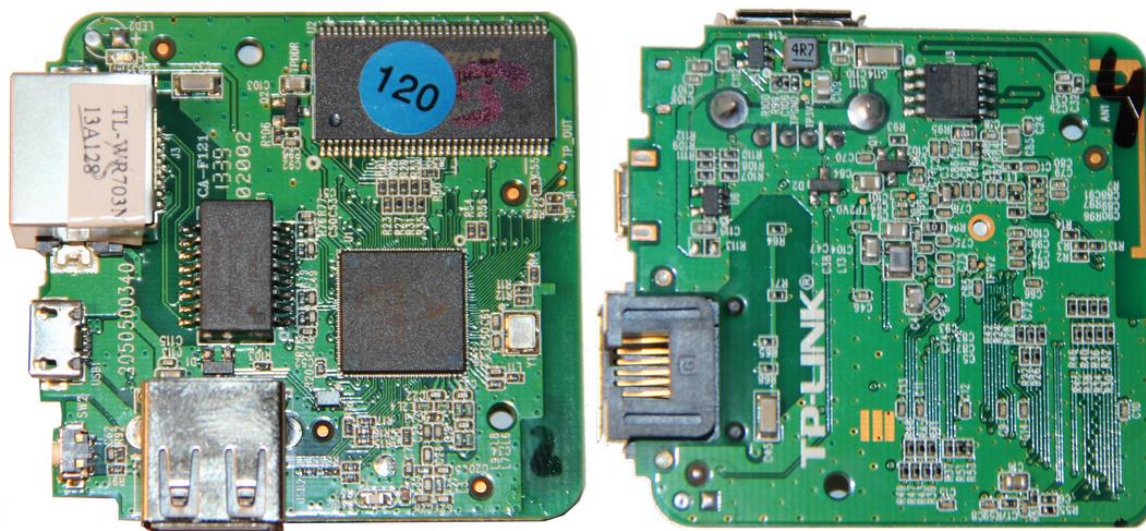
- Podpora linuxového operačního systému OpenWrt
- 4 MB paměti Flash, 32 MB paměti RAM
- Integrovaný Wi-Fi obvod s podporou standardu IEEE 802.11b/g/n
- Port USB 2.0
- Rozměry 57x57x18 mm (67x74x22 mm u MR3020)



Obrázek 16 – TP-LINK WR703N [19]

Směrovač WR703N využívá SoC Atheros AR9331 [19] ve kterém je integrovaný Wi-Fi obvod AR9285, který je pomocí interní PCI Express sběrnice připojen k 32bitovému procesoru AR7240 [20]. Tento procesor je postaven na architektuře MIPS 24K a je taktován na frekvenci 400 MHz. Dále je k němu připojena operační paměť DDR a Flash paměť o velikosti 4 MB.

Nevýhodou je, že výrobce oficiálně neposkytuje dokumentaci k SoC. Ta je dostupná pouze velkým odběratelům. Protože se však o toto zařízení začala hojně zajímat internetová komunita okolo operačního systému OpenWrt, tak pomocí reverzního inženýrství vzniklo celé schéma a plošný spoj. Později na Internet unikl i celá dokumentace SoC AR9331 který dále vývojářům posloužil k lepší implementaci OpenWrt.



Obrázek 17 – Deska směrovače TP-LINK WR703N

4.2 Napájení

Pro napájení bezdrátových zařízení se nejčastěji používají primární, nebo nabíjecí (sekundární) baterie doplněné o nabíjecí modul. V úvahu se také musí brát samovybitení sekundárních baterií, které se liší podle použité technologie.

Primární články	- Lithiový článek (1,5 – 3,7 V) - Alkalický článek (1,5 V)
Nabíjecí(sekundární) články	- NiMH (1,2 V) - Li-Pol / Li-Ion (3,7 V)

Tabulka 1 – Přehled nepoužívanějších článků

Po zvážení kapacity a velikosti napájecího článku byl zvolen Lithium polymerový článek o kapacitě 650 mAh a nominálním napětí 3,7 V a rozměrech 43x18x8 mm.



Obrázek 18 – Lithium polymerový článek

Podle typu baterie a jejího nominálního napětí dále podle potřeby připojíme obvod pro regulaci napětí. K tomuto účelu se nejlépe hodí stejnosměrné měniče s výrazně vyšší efektivitou než klasické napěťové regulátory v kombinaci se sériovým zapojením více

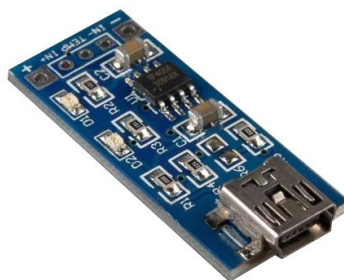
článků. Podle rozdílů vstupního a výstupního napětí určíme, jestli se bude jednat o zvyšující (step-up) nebo snižující (step-down) měnič.

V tomto případě padla volba na zvyšující měnič z 3,7 V na 5 V s maximálním výstupním proudem 500 mA. Deska plošného spoje s měničem obsahuje také přidáný ochranný integrovaný obvod pro ochranu lithiových článků proti přepětí a podpětí



Obrázek 19 – Zvyšující stejnosměrný měnič [21]

Aby bylo možno mikrofon udělat co nejvíce kompaktní a baterie se nemusela dobíjet externě tak bylo nutné zvolit nabíjecí obvod kompatibilní s daným typem sekundárního článku. Po prozkoumání možných obvodů byl vybrán nabíjecí obvod s integrovaným obvodem TP4056. Tento obvod podporuje změnu nabíjecího proudu pomocí výměny rezistoru R_{PROG} na plošném spoji a dále také připojení termistoru umístěného u baterie, který snímá teplotu baterie a zabraňuje přehřátí při nabíjení. Jeho účinnost je oproti řešení se stabilizátorem velmi vysoká, více než 85 %. Tento fakt velmi prodlouží životnost baterie. [22]



Obrázek 20 – Nabíjecí obvod

4.3 Zvuková karta

Protože směrovač WR703N disponuje pouze sériovou linkou a USB rozhraním, tak jediná možnost připojení zvukového kodeku je pomocí tohoto rozhraní.

Byla zvolena USB 2.0 zvuková karta s integrovaným obvodem CM119 a zvukovým vstupem a výstupem v podobě jack 3,5mm konektoru. Tato karta podporuje vzorkovací frekvence 44,1 kHz a 48 kHz.



Obrázek 211 – USB zvuková karta

5. OpenWrt

5.1 O systému

OpenWrt je operační systém založený na GNU/Linuxu a je určený zejména pro síťová a embedded zařízení.

5.1.1 Historie

Systém OpenWrt vznikl v roce 2004 díky uvolnění firmware směrovače WRT54G od společnosti Linksys pod licencí GPL, na jejímž základě je nutné uveřejnit zdrojové kódy. Tento základ posloužil tvůrcům systému jako reference k vytvoření linuxové distribuce, která je nadále komunitně vyvíjena a do dnešního dne byla portována na několik desítek architektur, jako jsou ARM, MIPS nebo i x86 a x64 [23].

5.1.2 Vlastnosti systému

Hlavním určením tohoto operačního systému je nahradit předinstalovaný firmware ve spotřebních směrovačích a přizpůsobit si systém podle svých potřeb přidáním funkcí, které výrobce do vlastního firmware nezahrnul. Od tohoto se odvíjejí hlavní přednosti a nedostatky tohoto systému. Mezi přednostmi se jednoznačně řadí možnosti programového vybavení pro síťovou komunikaci a propracovaný systém konfigurace síťových rozhraní ať se jedná o Wi-Fi nebo Ethernet. Jako jeden nedostatek lze vytknout snad jen komplikovanost systému, která může být pro běžného uživatele nepřekonatelnou překážkou.

Operační systém OpenWrt se ovládá zpravidla pomocí příkazové řádky připojením pomocí protokolu Telnet nebo SSH, případně v nouzi připojením sériové linky k desce. Dále je možné zařízení konfigurovat pomocí webového rozhraní (např. LuCI), které je však vhodné jen pro základní konfiguraci zařízení sloužícího jako síťový směrovač.

5.1.3 Hardwarové požadavky

Pro instalaci operačního systému OpenWrt stačí najít zařízení s podporovanou architekturou, alespoň 4 MB systémové paměti a 16 MB operační paměti RAM. Všechny zařízení, na kterých byl OpenWrt úspěšně spuštěn lze nalézt přímo na stránkách projektu OpenWrt. K většině těchto zařízení je nadále aktivně udržován repozitář aktuálních binárních obrazů systému OpenWrt a kompatibilních balíčků.

5.1.4 Licence

OpenWrt se řadí k volně šiřitelnému software vydávaného pod licencí GNU GPL verze 2. Tato licence umožňuje dále upravený systém šířit zadarmo nebo za poplatek, ale pouze za předpokladu, že příjemce bude mít stejná práva k dalšímu šíření tohoto software. [24] Pod touto licencí je poskytována většina svobodného software včetně samotného linuxového jádra, které je základem každého operačního systému Linux.

5.2 Softwarové vybavení

5.2.1 Struktura

Operační systém OpenWrt se skládá z několika na sobě závislých součástí. První a hlavní součástí je „kernel“ neboli linuxové jádro. V případě OpenWrt může být jádro ořezáno až na minimum do velikosti 1024 kB. Další funkcionality lze přidávat instalací modulů do kernelu. Některé lze zahrnout již při kompilaci celého systému, což se provádí nástrojem zvaným „toolchain“. Protože však OpenWrt běží většinou na platformách s malým výpočetním výkonem postačujícím pouze pro konkrétní účel, tak se provádí takzvaná křížová kompilace, kdy kompilujeme systém na jiné architektuře, než pro kterou je určený.

Další důležitou součástí systému, která se však neřadí do samotného OS, je „bootloader“ neboli zavaděč systému. Zavaděč je spuštěn při každém spuštění zařízení a jeho úkolem je inicializovat hardware zařízení a nahrát jádro operačního systému do operační paměti. Protože zavaděč pracuje na nejnižší úrovni a pracuje se samotným hardware tak musí být specificky nastavený pro každé zařízení. Mezi nejpoužívanější bootloadery patří The Universal Bootloader, zkráceně U-Boot. Zavaděč má v paměti zařízení vyhrazený svůj oddíl. U OpenWrt je tento oddíl velký zpravidla 128 kB. [25] Tento zavaděč má svoje vlastní rozhraní, příkazový řádek, který obsahuje sadu základních příkazů, které lze využít zejména při odstraňování chyb nebo v případě zablokování zařízení a potřebě přehrání paměti.

Poslední částí systému je uživatelsky přepisovatelná paměť, která obsahuje přepisovatelné konfigurační soubory a programové vybavení. Toto programové vybavení lze snadno rozšířit například pomocí balíčkovacího systému.

5.2.2 Balíčkovací systém

Balíčkovací systém je software zjednodušující instalaci, správu, aktualizaci a odstranění programových balíčků. Tento program umožňuje pomocí několika základních příkazů stahování softwarových balíčků z lokálního nebo centrálního internetového úložiště.

Systém OpenWrt používá balíčkovací systém OPKG, který vychází z balíčkovacího systému DPKG, který je použit v linuxovém operačním systému Debian.

Balíčkovací systém řeší i závislosti mezi balíčky, takže si při instalaci určitého balíčku, který pro svou funkci vyžaduje nějaký další, stáhne všechny potřebné balíčky.

Balíčky pro systém OpenWrt se nachází v centrálním internetovém úložišti, kde jsou pro každou architekturu oddělené kompatibilní balíčky, které jsou neustále aktualizovány například při každé revizi operačního systému.

5.2.3 Flash layout

Naprostá většina zařízení nepoužívá konvenční pevné disky, ale Flash paměť. Tato nevolatilní Flash paměť může být architektury NAND nebo NOR. Každá z těchto architektur má svoje výhody a nevýhody, například paměť NOR má řádově vyšší časy potřebné pro smazání bloku paměti a pomalejší zápis, ale oproti paměti NAND má vyšší čtecí rychlosti.

Tato paměť bývá ve většině případů externí a je připojena přímo na řídicí čip. V případě, že je k zařízení připojena paměť pomocí jiného rozhraní, například USB nebo SATA, tak je mezi samotnou pamětí a řídicím čipem další vrstva nazvaná FTL – Flash translation layer, která se sama stará například o adresaci a chybové bloky.

Nad touto Flash pamětí je vytvořen speciální systém tzv. Flash layout, který dělí paměť do dvou nebo více vrstev na další oddíly, které usnadňují práci s daty.

Flash Layout (TP-Link MR3020)					
vrstva 0	spi0.0: 4096 kB				
vrstva 1	mtd0	mtd1	mtd2	mtd3	mtd4
velikost	128 kB	1024 kB	2816 kB	64 kB	64 kB
název	<i>u-boot</i>	<i>kernel</i>	<i>rootfs</i>	<i>config</i>	<i>art</i>
přípojný bod	-	-	/	-	-
soub. systém	-	-	SquashFS	-	-

Tabulka 2 – Flash layout OpenWrt [26]

Vrstva 0 je samotná Flash paměť připojená přes sériové rozhraní SPI.

Vrstva 1 je abstraktní vrstva, která rozděluje paměť na jednotlivé MTD (Memory technology device) oddíly. MTD vytváří rozhraní mezi Flash pamětmi, mající specifické ovladače pro každou technologii a operačním systémem.

V základní konfiguraci je paměť rozčleněna do pěti paměťových oddílů. V prvních dvou oddílech je již zmíněný zavaděč U-Boot a linuxové jádro – kernel. V oddílu mtd2 je připojen hlavní oddíl určený pro programové vybavení který využívá souborového systému SquashFS, který je určen pouze pro čtení. Jeho hlavní výhodou je komprese dat před uložením do paměti čímž šetří až 30 % místa a podpora techniky zvané „Failsafe“, která umožňuje v případě zablokování vlivem špatné konfigurace „naboťování“ do speciálního předdefinovaného stavu, kdy je zařízení dostupné na IP adrese 192.168.1.1 protokolem Telnet a lze jej vrátit do původního stavu po instalaci, změnit heslo nebo načíst prepisovatelný oddíl a pozměnit konfiguraci, která zapříčinila zablokování.

Kromě oddílu určeného pouze pro čtení se po prvním startu vytvoří také prepisovatelný oddíl využívající souborového systému JFFS2 do kterého jsou instalovány přídatné programy a zapisovány všechny provedené změny konfigurace. Mezi výhody tohoto souborového systému patří podpora žurnálování, které před provedením změny v paměti zapíše danou akci do záznamu nazvaného žurnál, což při nečekaném přerušení zápisu této změny umožňuje snadnější regeneraci paměti[26].

Oba tyto souborové systémy bývají spojeny v jednom MTD oddílu, přičemž je přidána další abstraktní vrstva, ve které jsou odděleny.

V případě, že nedostačuje systémová paměť, lze využít techniky Extroot která umožňuje přenesení celého prepisovatelného souborového oddílu na další přídatné médium jako je například USB Flash disk nebo SD kartu. Tento přídatný oddíl překryje původní a je dále využíván jako primární úložiště.

Jako poslední dva oddíly jsou zde oddíl „config“, který obsahuje blíže nespecifikovaná data a oddíl „art“, plným názvem „Atheros radio test“, který obsahuje kalibrační data nutná pro spuštění rádiového modulu Wi-Fi.

5.2.4 Konfigurační nástroje

Operační systém OpenWrt je v konfiguraci velmi podobný klasickým linuxovým distribucím jako je např. Debian. Díky své orientaci na použití v síti má však propracovanější systém konfigurace síťových rozhraní.

Tento systém se nazývá UCI („Unified Configuration Interface“). Jeho základem jsou konfigurační soubory umístěné ve složce `/etc/config`. Tyto soubory mají jednotnou syntaxi a strukturu. Lze je upravovat přímo pomocí textového editoru nebo přes příkazovou řádku pomocí programu `uci`, který však jen modifikuje identické soubory. Může ale velmi zjednodušit automatizovanou úpravu nastavení například pomocí skriptů.

Stěžejní konfigurační soubory:

<code>/etc/config/network</code>	Nastavení síťových rozhraní, směrování a přepínače
<code>/etc/config/wireless</code>	Nastavení Wi-Fi modulu a bezdrátových sítí
<code>/etc/config/dhcp</code>	Nastavení DHCP serveru a lokálního DNS serveru
<code>/etc/config/dropbear</code>	Nastavení SSH serveru
<code>/etc/config/firewall</code>	NAT, filtrování paketů, přesměrování portů atd.
<code>/etc/config/fstab</code>	Nastavení připojených oddílů

Tabulka 3 – Konfigurační soubory OpenWrt

5.2.5 CLI

Pro ovládání operačního systému bez grafického prostředí se používá příkazový řádek. Program, který interpretuje příkazy napsané do této příkazové řádky, nebo do souboru, se nazývá CLI („Command line interface“), v unixových systémech je nazýván shell. Tento program vytváří prosté vstupní a výstupní rozhraní k operačnímu systému.

Napříč všemi operačními systémy jsou používány různé varianty shellů. OpenWrt používá Ash shell, upravenou odnož shellu převzatého z operačního systému Debian. Lze však doinstalovat i jiné příkazové interprety. Většina unixových shellů má však téměř shodnou syntaxi takže se dají snadno substituovat.

Ash shell a jeho rozšířenější alternativu Bash, lze použít také jako skriptovací jazyk pro vytváření efektivních shellových skriptů.

5.2.6 Busybox

BusyBox je souborem základních příkazů potřebných pro uživatelské operace v linuxových operačních systémech. Busybox je určen přímo pro vestavěné linuxové systémy do kterých spadá i OpenWrt. Skládá se ze sady příkazů splňujících normu POSIX. Tento název je zkrácením názvu „Portable Operating System Interface“ definovaném v normě IEEE 1003.1. Ta definuje jednotné uživatelské rozhraní a stěžejní funkcionalitu. K těmto příkazům patří příkazy, jako jsou `cd`, `ls`, `mv`, `mount`, `ping` a řada dalších. Tyto příkazy jsou často zbavené některých svých méně využívaných funkcí a všechny jsou zkompileovány do jednoho binárního balíčku. [27]

5.2.7 OSS

Open Sound System je zvukové rozhraní, které už v dnešní době nahradil ve většině distribucí rozhraní ALSA, ale protože má OSS menší výpočetní a paměťové nároky tak je pro OpenWrt ideální.

Toto rozhraní obsahuje mimo jiné také 2 speciální soubory, do kterých lze zapisovat, ale lze z nich i číst zvuková data.

`/dev/dsp` výstup 8bit unsigned, lineární kvantování

`/dev/audiod` výstup 8bit, μ -law kvantování [28]

5.2.8 Další nástroje

Při vývoji bezdrátového mikrofonu jsem vyzkoušel velké množství programů a platforem, ale ve finálním výrobku zůstala pouze malá část.

Jedním z těchto nástrojů je netcat, celým názvem „The GNU Netcat project“, což je nástroj umožňující komunikaci napříč sítě pomocí protokolu TCP nebo UDP. Program netcat může pracovat v několika módech.

Mód připojení – netcat se připojí na danou IP adresu a konkrétní port a může snadno obousměrně komunikovat se serverem.

Mód naslouchání – netcat v tomto módu spuštěn s parametrem *-l* a na předem daném portu čeká na připojení klientů a naslouchá příchozí komunikaci pomocí protokolu UDP nebo TCP.

Mód tunelování – tento režim je kombinací obou předchozích módů a funguje tak, že netcat naslouchá na určitém portu a zároveň je připojený na cílový server který oproti zdrojovému může být na jiném portu a využívat i jiný protokol pro připojení.

Mód skenování – v tomto režimu netcat vyzkouší, jaké porty jsou na cílové IP adrese dostupné pro komunikaci. Při použití protokolu TCP k tomu použije standardizovaný proces nazvaný „handshake“ a při použití UDP protokolu odešle malé množství dat. [29]

6.1.3 Konfigurace Wi-Fi

Konfigurace Wi-Fi se částečně liší od ostatních rozhraní. Nejprve je nutné si v souboru `network` připravit *interface* tak jako pro připojení *lan*. Zde záleží na režimu provozu daného zařízení. První možnost je mít zařízení v režimu přístupový bod a poskytovat ostatním Wi-Fi připojení, druhá možnost je mít zařízení v režimu klient a připojovat se na danou Wi-Fi síť.

Konfigurace pro Wi-Fi mikrofon v režimu klient.

```
config interface 'wwan'
    option proto 'static'
    option ipaddr '192.168.2.10'
    option netmask '255.255.255.0'
    option gateway '192.168.2.1'
    option dns '8.8.8.8'
```

Konfiguraci bezdrátového rozhraní nalezneme v souboru `/etc/config/wireless`. Při poškození tohoto souboru, např. vlastní špatnou konfigurací, lze základ tohoto souboru obnovit pomocí příkazu `wifi detect > /etc/config/wireless` který vygeneruje nový soubor s nastavením parametrů rádiové části.

```
config wifi-device radio0
    option type mac80211
    option channel 6
    option hwmode 11n
    option path 'platform/ar933x_wmac'
    list ht_capab SHORT-GI-20
    list ht_capab SHORT-GI-40
    list ht_capab RX-STBC1
    list ht_capab DSSS_CCK-40
    option htmode HT20
    option disabled 0

config wifi-iface
    option device radio0
    option network wwan
    option mode sta
    option ssid WifiMikrofon
    option encryption psk
    option key mikrofon1234
```

Konfigurace přístupového bodu proběhne obdobně s několika rozdílnými parametry. Stejně jako u síťového rozhraní musíme při změnách obnovovat rozhraní v tomto případě bez pomoci init skriptů pouze kombinací příkazů `wifi down` ; `wifi up` který restartuje Wi-Fi rozhraní.

6.1.4 Extroot

Při vývoji mikrofonu bylo nutné zkusit různé varianty, s čímž byla spojena instalace velkého množství software do Flash paměti zařízení. Je zde však dostupný USB port, na který lze připojit USB hub a na něj dále připojit externí paměť zároveň se zvukovou kartou. Tato přidaná paměť může být s využitím techniky extroot snadno připojena ke kořenovému svazku paměti zařízení, a tím značně rozšířit její paměť.

Tato funkcionalita spočívá v překopírování celé programové paměti do externího úložiště, které je naformátováno na souborový systém Ext4, a nastavení tabulky souborového systému aby načítala kořenovou složku z externího úložiště místo vlastní paměti.

6.1.5 Sysupgrade

Protože se systém OpenWrt neustále vyvíjí, tak vznikají nové verze, které například opravují předchozí chyby nebo přidávají nové funkcionality. Pro nainstalování nové verze slouží program `sysupgrade`, který však smaže všechny složky a soubory, které nejsou vypsány ve v konfiguračním souboru `/etc/sysupgrade.conf`.

Tento process je nutné podstoupit například pokud nejde nainstalovat balíček z úložiště což bývá nejčastěji způsobeno tím, že nové balíčky potřebují novější verzi linuxového jádra.

6.1.6 Audio

Systém OpenWrt ve výchozím stavu nemá nainstalovanou podporu zvukových zařízení, ale lze ji snadno doinstalovat balíčkovacím systémem OPKG ve formě modulů jádra `kmod-usb-audio` a `kmod-sound-core`. To že jsou ovladače nainstalované správně, ověříme příkazem, který by měl vypsát nainstalované zvukové karty.

```
cat /proc/asound/cards

0 [Device ]: USB-Audio - USB PnP Sound Device
             C-Media Electronics Inc. USB PnP Sound Device at usb-ehci-platform-1.1
```

6.1.7 Další software

Pro provoz webového rozhraní byl na druhé zařízení nainstalován webový server Nginx. Při instalaci se vytvořila složka s konfiguračními soubory v umístění `/etc/nginx`, kde stačí editovat konfigurační soubory a nakonec pomocí init skriptů `/etc/init.d/nginx enable` a `/etc/init.d/nginx start` server povolit a spustit.

Poté jsem nainstaloval podporu serverově orientovaného skriptovacího jazyka PHP ve verzi 5, aby bylo možné spouštět příkazy odeslané z webového rozhraní přímo v příkazové řádce druhého zařízení, které zároveň vytváří přístupový Wi-Fi bod pro mikrofon.

Pro zjednodušení přístupu k webovému rozhraní byl upraven DNS server ze kterého uživatel, který se připojuje k webovému rozhraní, dostává odpovědi na DNS dotazy. Byl přidán lokální DNS záznam, který přiřazuje dotazu na „*mikrofon.wifi*“ IP adresu webserveru kde běží webové rozhraní, takže když uživatel zadá do svého prohlížeče tuto adresu, tak mu bez psaní IP adresy vyskočí rovnou webové rozhraní.

6.1.8 Failsafe

Při vývoji na tomto systému se při častých změnách konfigurace může stát, že se celé zařízení zasekne a systém se při startu neinicilizuje, jak by měl. Pro tyto situace byl vytvořen tzv. „failsafe“ mód pro nouzovou změnu klíčových konfiguračních souborů nebo obnovu do původního nastavení.

V nejhorší možné situaci se může stát, že nezbyvá nic jiného než celý systém smazat a začít znovu. Není však potřeba mazat celou paměť, ta je totiž důmyslně rozdělena do několika partií a stačí smazat pouze jedinou část, kde se nachází proměnná data, samotné linuxové jádro zůstane nedotčeno.

```
mtd -r erase rootfs_data
rm -r /overlay/*
```


Těmito příkazy se smažou všechna data, která uživatel nastavil a systém se vrátí do stavu, v jakém byl po instalaci.

6.2 Shell

Pro ovládání obou zařízení byla vytvořena sada shellových skriptů které vykonávají řadu činností. Část z nich běží na bezdrátovém Wi-Fi mikrofonu a další část běží na druhém zařízení pro kooperaci s mikrofonem a ovládání z webového rozhraní.

6.2.1 Skripty v mikrofonu

V bezdrátovém mikrofonu běží na pozadí dva skripty, které se spouští při startu. Toho je docíleno zapsáním do souboru skriptu `/etc/rc.local`, který se spouští na konci bootovacího procesu.

První jednoduchý skript má na starost každých 15 sekund zkontrolovat, jestli je zařízení připojené na Wi-Fi a jestliže tomu tak není tak se zjistí dostupné Wi-Fi sítě a pokusí se připojit na předdefinovanou síť.

Druhý a důležitější skript se stará o zbytek ovládání celého mikrofonu. Běží v nekonečném while cyklu a po vytvoření TCP socketu pracuje v režimu naslouchání a čeká na příchozí data na předem daném portu

```
...
while [ : ]
do
    $NETCAT -l -p $SERVERPORT | while read CMD
do
...

```

V další části porovná příchozí data, a když se některý příchozí řetězec shoduje s předdefinovaným tak vykoná příkaz, který k němu náleží. Po skončení tohoto příkazu opět naslouchá na daném portu na příchozí zprávu.

```
...
case $1 in
"REC") echo REC
# Zarizeni zacne nahravat audio signal a pomoci nc ho preposle
cat /dev/audio | $NETCAT -u $SINKIP $AUDIOPORT
;;
"KILL") echo KILL
#Ukonci procesy spojene s prijimanim nebo vysilanim na audiportu
PID=$(ps -w | grep "nc" | grep $AUDIOPORT | awk '{print $1}')
kill -9 $PID
;;
...

```

Tento skript zajišťuje všechny základní úkony, které jsou při komunikaci s druhou stranou potřeba. Je tu možnost zahájení přenos z mikrofonu, existuje i druhá varianta kdy lze obrátit logiku přenosu a na tomto zařízení naopak přijímat zvuková data z druhé strany. Další možností je například změna komunikačního portu, na němž běží přenos zvuku, ten může být už využíván jinou službou, nebo blokován firewallem. Z dalších příkazů je zde možnost odeslání údajů o kvalitě a podrobnostech o Wi-Fi připojení, nebo příkaz který ukončí všechny běžící přenosy.

6.2.2 Skripty v přijímači

V druhém zařízení, které slouží buď ke konfiguraci mikrofonu, nebo i pro spuštění a reprodukci přenosu z mikrofonu, je uloženo několik menších skriptů. Velká část z nich slouží pro ovládání přenosu z webového rozhraní. Mezi tyto patří například skript s názvem *zahajit.sh*, který po stisknutí tlačítka Play ve webovém rozhraní zahájí přenos zvuku.

```
#!/bin/ash
SINKIP="192.168.2.10"
APORT="9999"
CTRLP="19999"

echo "KILL" | nc -c $SINKIP $CTRLP
sh /usr/skripty/ukoncit.sh -q
sleep 1
topo=`cat topo`
if [ "$1" != "-q" ]; then
    echo "Zahajuji prenos zvuku..."
fi
if [ "$topo" = "default" ]; then
    echo "REC" | nc -c $SINKIP $CTRLP ; nc -lu -p $APORT > /dev/audio &
    echo "default" > topo
else
    echo "PLAY" | nc -c $SINKIP $CTRLP ; cat /dev/audio | nc -u $SINKIP $APORT
    echo "reverse" > topo
fi
echo "1" > state
exit 0
```

Další skripty, které se nacházejí na tomto zařízení, zajišťují ukončení nebo pozastavení přenosu, ovládání tlačítkem na zařízení a ověření připojení bezdrátového mikrofonu.

6.3 Webové rozhraní

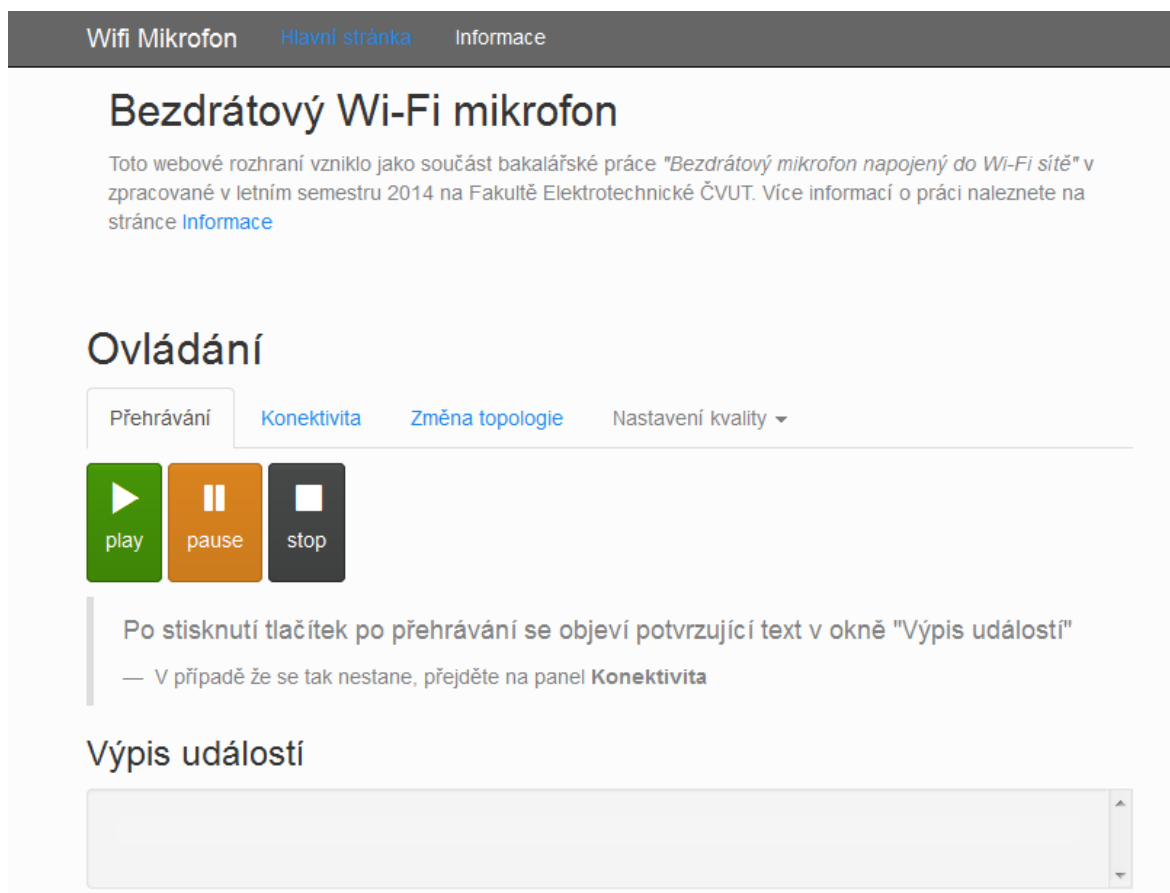
6.3.1 Bootstrap

Ovládací webové rozhraní bylo vytvořeno pomocí open-source řešení Bootstrap, které dává volně k dispozici svoje zdrojové kódy i s popisem struktury a dalšími prostředky po usnadnění tvorby webu založeném na tomto prostředí.

Bootstrap využívá více technologií. Základní struktura je napsána pomocí jazyku HTML doplněného kaskádové styly CSS. Dále využívá programovacího jazyka JavaScript, který slouží k vytváření dynamických prvků, jako jsou např. tlačítka a další aktivní prvky. V další ukázce kódu lze vidět, jak vypadá část zdrojového kódu menu využívajícího prvků z prostředí Bootstrap. [30]

```
...
<h2 id="nav-tabs">Ovládání</h2>
<div class="bs-component">
  <ul class="nav nav-tabs" style="margin-bottom: 10px;">
    <li class="active"><a href="#prehravani" data-toggle="tab">Přehrávání</a></li>
    <li class=""><a href="#pripojeni" data-toggle="tab">Konektivita</a></li>
    <li><a href="#topologie" data-toggle="tab">Změna topologie</a></li>
  </ul>
</div>
...
```

Mezi jednu z hlavních výhod tohoto prostředí patří dynamické přizpůsobování podle velikosti obrazovky, tak aby bylo možné snadno ovládat web i z přístrojů s menším rozlišením obrazovky, například mobilních telefonů. [30]



Obrázek 222 – Webové rozhraní

6.3.2 Ovládací prvky

Aby mohlo webové rozhraní interagovat se samotným zařízením a posílat příkazy, tak bylo nutné využít serverově orientovaného skriptovacího jazyka PHP, jehož prostředí běží a zároveň komunikuje s webovým serverem NGINX v rámci jednoho zařízení.

Docílit spouštění příkazů na příkazové řádce zařízení s OpenWrt nebo jiným linuxovým systémem je pomocí jazyka PHP možné díky příkazu `shell_exec`, který vezme předaný argument a ten spustí na příkazové řádce. Po vykonání příkazu vrátí zpět celý výstup jako datový typ string, ten je poté možné vypsát na webu nebo dále zpracovat. [31]

S využitím tohoto příkazu byl napsán krátký PHP skript `commands.php`, který kromě spouštění příkazů také ukládá všechny výstupní data do souboru uloženém na serveru.

```
<?php
    $q=$_REQUEST["q"];
    $output = shell_exec($q);
    echo $output;

    $saveLog = "/www/" . "log" . ".txt";
    $obsahSoub = file_get_contents($saveLog);
    file_put_contents($saveLog, $output . $obsahSoub);
?>
```

V mém řešení jsem dále využil jednoduchého skriptu v jazyce JavaScript. V tomto skriptu je definovaná funkce *klik*, která je volána při stisknutí tlačítek, jenž jí využívají. Tyto tlačítka jí předají svůj argument funkce, ve kterém je napsaný konkrétní příkaz ke spuštění na příkazovém řádku, tato funkce je pak dále předá PHP skriptu *commands.php*. Funkce *klik* využívá technologie AJAX, konkrétně rozhraní XMLHttpRequest, které umožňuje asynchronní výměnu dat mezi serverem a uživatelem bez nutnosti znovunačtení celého webu. Tohoto rozhraní využívá k zobrazení výstupních dat v okně „Výpis událostí“, kde se po provedení daných příkazů, zobrazí výstupní data. [32]

```
function klik(str,sessionID)
{
  if (str.length==0)
  {
    return;
  }
  var xmlhttp=new XMLHttpRequest();
  xmlhttp.onreadystatechange=function()
  {
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
      getHTML = document.getElementById("output").innerHTML;
      document.getElementById("output").innerHTML+xmlhttp.responseText
      + "\n" + getHTML;
    }
  }
  xmlhttp.open("GET","commands.php?q="+str,true);
  xmlhttp.send();
}
```

6.4 Slepé vývojové cesty

Při vývoji bezdrátového mikrofonu jsem vyzkoušel řadu způsobů, jak lze zvukový signál přenášet po síti. Některé však byly pro konečné řešení nepoužitelné.

Jedno ze softwarových řešení, jenž jsem vyzkoušel, byl program FFserver, součást balíčku FFmpeg, které se používá pro převod, nahrávání a úpravu multimediálních souborů.

FFserver je program podporující stream audia i videa s využitím protokolu RTP a podporou multicastu. [33]

```
root@OpenWrt:~# ffmpeg -re -f ul16le -ac 1 -i /dev/dsp -re -f rtp -ar 44100
rtp://224.0.0.100:7000
```

Přenos audia se podařilo uskutečnit pomocí konfigurace nastavení ffserveru a výše zmíněného příkazu. Výstup byl však více než neuspokojivý. Po připojení na stream byly slyšet pouze útržky přenášeného zvuku se zpožděním přibližně 30 sekund.

S programem FFserver se však podařilo zprovoznit multicastové vysílání v místní síti, avšak v konečné verzi software nebylo úspěšně implementováno z důvodu nekompatibility software.

Podobného výsledku bylo dosaženo i s dalším programem DarkIce, který převádí audio vstup a posílá ho do stream serveru IceCast, přes který je poté distribuován dále.

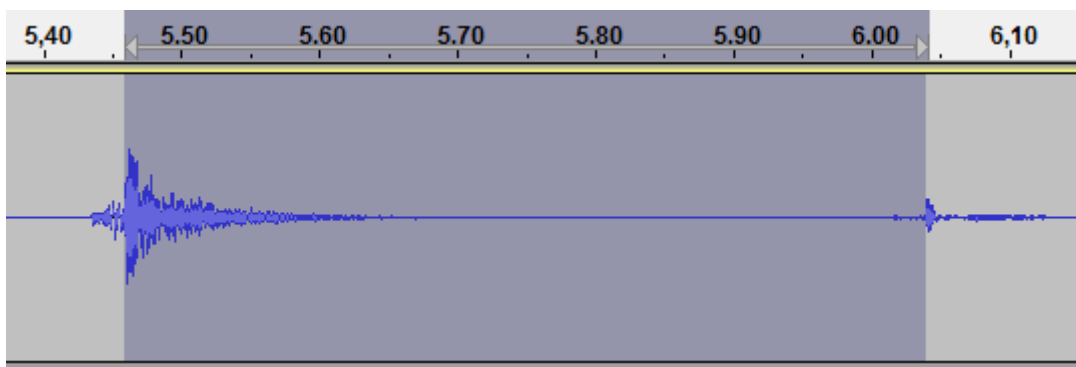
Po pátrání, co je příčinou tohoto problému, jsem dospěl k závěru, že díky využití komprimovaných zvukových formátů, které tyto servery používají, je výkon mikrofonu,

potazmo zařízení WR703N, je nedostatečný pro kontinuální převod audia v reálném čase v kombinaci se jeho přeposíláním do sítě.

Také jsem zjistil, že procesor obsažený v SoC Atheros AR9331, neobsahuje FPU – Floating point unit, což je jednotka v procesoru, která se stará o práci čísla s proměnnou řádovou čárkou, která je při náročných procesech převodu zvuku do kompresního formátu potřebná.

6.5 Problémy výsledné realizace

Implementace řešení proběhla úspěšně, pouze s jedním nečekaným problémem. Po spuštění přenos zvuku z bezdrátového mikrofonu na druhé zařízení funguje, s jedinou vadou a tou je zpoždění při přenosu. Délka zpoždění se rovnala přibližně 583 ms, což je pro použití pro nahrazení analogového mikrofonu nepoužitelné. Uplatnění by toto zařízení našlo i se zpožděním například jako baby monitor.



Obrázek 233 – Zpoždění při přenosu

7. Závěr

V této práci jsem se z počátku věnoval teoretickému rozboru zadání a jeho dílčích součástí a s nimi souvisejícím teoretickým podkladem nutným pro další návrh řešení.

Dále jsem se věnoval výběru koncového zařízení a popisem operačního systému OpenWrt, na kterém stojí velká část řešení bezdrátového mikrofonu. Práce s operačním systémem OpenWrt pro mě byla velkým přínosem, který určitě najde uplatnění v budoucnu.

Při realizaci jsem čelil řadě překážek, které se mi až na výjimky podařilo úspěšně překonat. Jednou z nich je zpoždění zvuku, vznikající před odesláním do sítě. Zde by byl prostor k dalšímu vylepšení.

Možnost vylepšení bych viděl hlavně na straně hardware, kde by bylo vhodné využít více otevřené platformy využívající systém OpenWrt a integrace větší paměti, která by nebyla tolik omezující při implementaci softwarového vybavení a umožnila přidání zlepšujících funkcí.

Výsledkem práce je funkční a univerzální řešení, které je vhodné k přenosu zvuku v síti a nenáročné na použití i pro běžného uživatele. Nadále bych toto řešení rád rozvíjel a využil nabytých zkušeností k jeho zdokonalení.

8. Literatura

8.1 Seznam použité literatury

- [1] NAVE, Carl R. Microphones. In: *Georgie State University* [online]. [cit. 2014-05-20]. Dostupné z: <http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/mic2.html>
- [2] ORSÁG, Petr. Měření základních charakteristik mikrofonů. Brno, 2011. Bakalářská práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Ing. ZDENĚK HAVRÁNEK, Ph.D.
- [3] FEIERTAG, Gregor, Matthias WINTER a Anton LEIDL. Flip chip packaging for MEMS microphones. ScienceDirect [online]. 2010, s. 4 [cit. 2014-05-20]. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1877705810006685>
- [4] PHILIPS SEMICONDUCTORS. I2S Bus Specification. [online]. 1986 [cit. 2014-05-20]. Dostupné z: <https://www.sparkfun.com/datasheets/BreakoutBoards/I2SBUS.pdf>
- [5] POLAND, K. L. Quantization and Pulse Code Modulation. T1 - Basic Transmission Theory [online]. [cit. 2014-05-20]. Dostupné z: http://www2.uic.edu/stud_orgs/prof/pesc/part_3_rev_F.pdf
- [6] BENEŠ, Petr. Přenos audio signálu pomocí protokolu IP. Praha, 2013. Bakalářská práce. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ. Vedoucí práce Ing. Pavel Bezpalec, PhD.
- [7] ARMS, Caroline a Carl FLEISCHHAUER. MP3 (MPEG Layer III Audio Encoding). Sustainability of Digital Formats [online]. [cit. 2014-05-20]. Dostupné z: <http://www.digitalpreservation.gov/formats/fdd/fdd000012.shtml>
- [8] ŠVEC, Jiří. Kompresce zvuku? Jen podvod na uši!. In: AVmania [online]. 2009 [cit. 2014-05-20]. Dostupné z: <http://avmania.e15.cz/kompresce-zvuku-jen-podvod-na-usi>
- [9] FAIGL, Jan. MP3 nebo Ogg?. In: Root.cz [online]. 2002 [cit. 2014-05-20]. Dostupné z: <http://www.root.cz/clanky/mp3-nebo-ogg/>
- [10] BEURDEN, Martijn. Lossless Codec Audio Comparison. Xiph.org [online]. 2013 [cit. 2014-05-20]. Dostupné z: <http://www.xiph.org/flac/comparison.pdf>
- [11] DOSTÁLEK, Libor a KABELOVÁ. Velký průvodce protokoly TCP/IP a systémem DNS. Praha: Computer Press, 2000, s. 216-234. ISBN 80-7226-323-4.
- [12] LU, Wei, Carolyn MATTHEWS, Lydia PARZIALE a Nicolas ROSSELOT. IBM. TCP/IP Tutorial and Technical Overview: An IBM Redbooks publication [online]. 2006 [cit. 2014-05-20]. ISBN 9780738494685. Dostupné z: <http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>
- [13] LE BOUDEC, Jean-Yves, Andrzej DUDA a Patrick THIRAN. Real Time Protocol (RTP). [online]. [cit. 2014-05-20]. Dostupné z: <http://icapeople.epfl.ch/thiran/CoursED/RTP.pdf>
- [14] LEDVINA, Jiří. RTP – Real Time protocol. [online]. 2007 [cit. 2014-05-20]. Dostupné z: http://www.kiv.zcu.cz/~ledvina/Prednasky-PDS-2007/09a-rtp_VoIP.pdf
- [15] DVOŘÁK, M. Multicast na Ostravské univerzitě. [online]. 2006 [cit. 2014-05-21]. Dostupné z: http://www.osu.cz/cit/dokumenty/Multicast_na_OU.pdf

- [16] VOŽŇÁK, Miroslav. TECHNICKÉ PRINCIPY IP TELEFONIE. [online]. 2004 [cit. 2014-05-20]. Dostupné z: http://www.phonet.cz/archiv/dok_osta/ipt-2004_Principy_IPtel.pdf
- [17] KOCUR, Z. a M. ŠAFRÁNEK. Fyzická vrstva Wi-Fi. [online]. 2008 [cit. 2014-05-20]. Dostupné z: <http://access.feld.cvut.cz/view.php?cisloclanku=2008050006>
- [18] IEEE 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York: IEEE, 2013. Dostupné z: <http://standards.ieee.org/getieee802/download/802.11-2012.pdf>
- [19] TP-Link TL-WR703N. In: *Openwrt.org* [online]. [cit. 2014-05-20]. Dostupné z: <http://wiki.openwrt.org/toh/tp-link/tl-wr703n>
- [20] AR9285: Single-chip PCIe based on 802.11n 1-stream specification. In: [online]. 2013 [cit. 2014-05-20]. Dostupné z: <http://www.qca.qualcomm.com/wp-content/uploads/2013/11/AR9285.pdf>
- [21] 1 PCB Li-ion Lipo Battery 3.7V step up Converter. *Ebay.com* [online]. [cit. 2014-05-20]. Dostupné z: <http://www.ebay.com/itm/1-PCB-Li-ion-Lipo-Battery-3-7V-step-up-Convert-to-Output-Power-DC-5V-DLX-UPDC5V-/251115596858>
- [22] TP 4056: 1A Standalone Linear Li-Ion Battery Charger. In: [online]. [cit. 2014-05-20]. Dostupné z: <http://www.ndatasheet.com/download.php?id=770462>
- [23] Table of Hardware. In: *Openwrt.org* [online]. [cit. 2014-05-20]. Dostupné z: <http://wiki.openwrt.org/toh/start>
- [24] GNU GPLv2. *GNU General Public License version 2*. Free Software Foundation, Inc., 2007. Dostupné z: <http://www.gnu.org/copyleft/gpl.html>
- [25] Das U-Boot Environment. In: *Openwrt.org* [online]. [cit. 2014-05-20]. Dostupné z: <http://wiki.openwrt.org/doc/techref/bootloader/uboot.config>
- [26] The OpenWrt Flash Layout. In: *Openwrt.org* [online]. [cit. 2014-05-20]. Dostupné z: <http://wiki.openwrt.org/doc/techref/flash.layout>
- [27] General Questions. In: *Busybox.net* [online]. [cit. 2014-05-20]. Dostupné z: <http://www.busybox.net/FAQ.html>
- [28] Open Sound System. In: *4front-tech.com: Digital Audio Solutions* [online]. [cit. 2014-05-20]. Dostupné z: <http://www.4front-tech.com/pguide/intro.html>
- [29] LOS, Petr. Netcat: Švýcarský armádní nůž pro TCP/IP. In: *Abclinuxu.cz* [online]. 2013 [cit. 2014-05-20]. Dostupné z: <http://www.abclinuxu.cz/clanky/netcat-svycarsky-armadni-nuz-pro-tcp-ip>
- [30] Getting started: An overview of Bootstrap. In: *Getbootstrap.com* [online]. [cit. 2014-05-20]. Dostupné z: <http://getbootstrap.com/getting-started/>
- [31] Shell_exec. In: *Php.net* [online]. [cit. 2014-05-20]. Dostupné z: <http://www.php.net/manual/en/function.shell-exec.php>
- [32] AJAX. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2014-05-20]. Dostupné z: <http://en.wikipedia.org/wiki/AJAX>
- [33] Fserver documentation. In: *Ffmpeg.org* [online]. [cit. 2014-05-20]. Dostupné z: <https://www.ffmpeg.org/ffserver.html>

8.2. Seznam obrázků

Obrázek 1 – Obecné blokové schéma přenosu akustického signálu	2
Obrázek 2 - Průřez elektretovým mikrofonem [1]	3
Obrázek 3 - Architektura MEMS mikrofonu [3].....	3
Obrázek 4 - Srovnání lineární a nelineární (μ -law) kvantizační křivky. [5]	4
Obrázek 5 – Frekvenční maskování [8].....	5
Obrázek 6 – OSI a TCP/IP model	6
Obrázek 7– TCP záhlaví [11]	7
Obrázek 8- UDP záhlaví [11]	7
Obrázek 9 – Umístění RTP protokolu v TCP/IP modelu [16]	8
Obrázek 10 - Jitter neboli „chvění“ signálu [16].....	9
Obrázek 11 – Přehled Wi-Fi standardů [17].....	9
Obrázek 12 - Wi-Fi mikrofon jako přístupový bod.....	11
Obrázek 13 - Wi-Fi mikrofon jako klient.....	12
Obrázek 14 – Polygonální topologie	13
Obrázek 15 – Blokové schéma bezdrátového mikrofonu.....	14
Obrázek 16 – TP-LINK WR703N [19]	15
Obrázek 17 – Deska směrovače TP-LINK WR703N.....	16
Obrázek 18 – Lithium polymerový článek	16
Obrázek 19 – Zvyšující stejnosměrný měnič [21].....	17
Obrázek 20 – Nabíjecí obvod	17
Obrázek 21 – USB zvuková karta	17
Obrázek 22 – Webové rozhraní	28
Obrázek 23 – Zpoždění při přenosu	30

8.3. Seznam tabulek

Tabulka 1 – Přehled nejpoužívanějších článků	16
Tabulka 2 – Flash layout OpenWrt [26]	20
Tabulka 3 – Konfigurační soubory OpenWrt	21

8.4 Seznam použitých zkratek

A/D	Analogově-digitální
ARM	Advanced RISC Machine
CD	Compact disc
CLI	Command line interpreter
CSS	Cascading Style Sheets
DC	Direct current
DDR	Double data rate
DHCP	Dynamic host
DNS	Domain Name System
DSSS	Direct Sequence Spread Spectrum
FLAC	Free Lossless Audio Codec
FPU	Floating point unit
GNU	GNU's Not Unix
GPL	General Public License
I ² C	Inter-Integrated Circuit
I ² S	Inter-IC Sound
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JFFS2	Journalling Flash File System version 2
LAN	Local Area Network
MEMS	Microelectromechanical systems
MIPS	Microprocessor without Interlocked Pipeline Stages)
MP3	MPEG-1 (2) Audio Layer III
MPEG	Moving Picture Experts Group
MTD	Memory Technology Device
OFDM	Orthogonal frequency-division multiplexing
OS	Operační systém
OSI	Open Systems Interconnection model
OSS	Open Sound System
PCI	Peripheral Component Interconnect
PCM	Pulse-code modulation
QoS	Quality of Service
RAM	Random-access memory
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SATA	Serial ATA
SD	Secure Digital
SoC	System on a chip
SPI	Serial Peripheral Interface
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USART	Universal asynchronous receiver/transmitter
USB	Universal Serial Bus

Přílohy



Příloha 1 – Bezdrátový mikrofon



Příloha 2 – Příjímač bezdrátového mikrofonu

Obsah příloženého CD

/Krystof_Cermak.pdf	Text bakalářské práce
/fotografie/	Fotografie zařízení
/skripty/	Shellové skripty
/webove_rozhvani/	Kompletní webové rozhraní