

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

DIPLOMOVÁ PRÁCE

2015

Bc. Petr Beneš

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

**Zařízení pro vysoce kvalitní přenos
audiosignálu IP sítí**

květen 2015

Vypracoval: Bc. Petr Beneš

Vedoucí práce: Ing. Pavel Troller, CSc.

Čestné prohlášení

Prohlašuji, že jsem svou diplomovou práci na téma „Zařízení pro vysoce kvalitní přenos audiosignálu IP sítí“ vypracoval samostatně a uvedl jsem všechny použité zdroje a literaturu.

Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry. V případě publikace práce nebo její významné části budu uveden jako spoluautor.

V Praze dne _____

Podpis

Poděkování

Rád bych poděkoval mému vedoucímu diplomové práce panu Ing. Pavlu Trollerovi, CSc. za jeho připomínky, cenné rady a čas, který mi věnoval při tvorbě mé práce. A dále mému spolubydlícímu a kamarádovi Filipu Zavadilovi, který mi pomáhal s prací.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Beneš Petr**

Studijní program: Komunikace, multimédia a elektronika
Obor: Sítě elektronických komunikací

Název tématu: **Zařízení pro vysoce kvalitní přenos audiosignálu IP sítí**

Pokyny pro vypracování:

Navrhněte a zkonstruuje prototyp(y) zařízení, které bude schopno přenosu audiosignálu z normalizovaného vstupu (symetricky 600 Ohm, 0.707V) na obdobně normalizovaný výstup na druhém zařízení. Zařízení budou vzájemně propojena IP sítí. Přenos mezi zařízeními bude obousměrný (samostatné linky tam/zpět) a navíc bude obousměrně přenášena signalizace v podobě jednovodičového logického stavu (např. přizemnění) pro dálkové řízení připojených zařízení.

Pro konstrukci využijte dostupných hardwareových komponent, jako je např. Raspberry Pi, externí zvuková karta s USB rozhraním atd.

Obslužný program bude sloužit jen pro ověření funkcí hardwareového řešení.

Seznam odborné literatury:

- [1] IP audio codec. Wwww.wikipedia.org [online]. [cit. 2014-10-21]. Dostupné z: http://en.wikipedia.org/wiki/IP_audio_codec
- [2] Vlček, K.: Komprese a kódová zabezpečení v multimediálních komunikacích, Praha, BEN-Technická literatura, 1999.

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2015/2016



prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 13. 11. 2014

Anotace:

Tato diplomová práce se zabývá základními pojmy a to internetovými protokoly. Dále obsahuje přehled základních druhů kódování a nejpoužívanější hlasové a zvukové kodeky. Cílem práce je seznámit se s přenosem audio signálu přes IP síť za pomoci zařízení Raspberry Pi (2) a zvukové karty.

Klíčová slova:

Internetový protokol, Audio, Kodek, Audio over IP, Raspberry Pi (2), U-Control UCA202, GPIO

Summary:

This diploma thesis describes the basic concepts of network, specifically internet protocols. It also contains an overview of the basic types of coding and most widely used voice and audio codecs. The aim is to get acquainted with the transmission of audio over IP network with the help Raspberry Pi 1 or 2 devices and a sound card.

Index terms:

Internet protocols, Audio, Codec, Audio over IP, Raspberry Pi (2), U-Control UCA202, GPIO

Obsah:

1.	ÚVOD	1
2.	ZÁKLADY IP SÍTÍ	2
2.1.	INTERNETOVÉ PROTOKOLY	3
2.1.1.	Síťová vrstva	3
2.1.2.	Transportní vrstva	6
2.1.3.	Aplikační vrstva.....	7
3.	KODEKY A JEJICH VLASTNOSTI	8
3.1.	DRUHY KÓDOVÁNÍ.....	8
3.1.1.	PCM.....	8
3.1.2.	DPCM	10
3.1.3.	ADPCM.....	10
3.1.4.	LPC	10
3.1.5.	CELP.....	11
3.2.	DRUHY KODEKŮ	12
3.2.1.	G.711.....	13
3.2.2.	G.722.....	14
3.2.3.	G.723.1.....	15
3.2.4.	G.726.....	15
3.2.5.	G.728.....	15
3.2.6.	G.729.....	16
3.2.7.	GSM.....	17
3.2.8.	SILK (v programu Skype)	18
3.2.9.	Opus.....	19
3.2.10.	iLBC	20
3.2.11.	MP3	21
3.2.12.	AAC	22
3.2.13.	Tieline.....	25
3.2.14.	AptX.....	26
3.3.	POROVNÁNÍ NĚKTERÝCH KODEKŮ	28
3.3.1.	MOS	28
4.	REALIZACE	31
4.1.	DŮVODY K VÝBĚRU	31
4.1.	RASPERRY PI.....	32
4.2.	RASPERRY PI 2	33
4.3.	U-CONTROL UCA202.....	34
4.4.	INSTALACE A KONFIGURACE ZAŘÍZENÍ	35
4.4.1.	Operační systém(y)	35
4.4.2.	Mplayer.....	37
4.4.3.	VLC player.....	38
4.4.4.	Konfigurace externí zvukové karty UCA202.....	39
4.4.5.	Instalace potřebných programů - node.....	43
4.4.6.	GPIO.....	44
4.4.7.	Ukázkové zapojení.....	45

4.5.	OVĽÁDÁNÍ A SAMOTNÝ SKRIPT	47
4.5.1.	<i>Ovládání</i>	47
4.5.2.	<i>Funkčnost</i>	49
4.5.3.	<i>Samotný skript</i>	51
4.5.4.	<i>Zlepšení celého systému</i>	52
5.	VYHODNOCENÍ	53
6.	LITERATURA	55
6.1.	SEZNAM OBRÁZKŮ	58
6.2.	SEZNAM TABULEK	60
PŘÍLOHY	61

1. Úvod

Ve své práci se budu zabývat přenosem zvukového signálu pomocí protokolu IP. V práci jsou popsány referenční modely. Dále je zde důležitá část a to jsou internetové protokoly. Je zde stručně popsán IP protokol u síťové vrstvy a u transportní TCP, UDP a RTP. Druhy nejpoužívanějších kódování a „rů“ celé diplomové práce používané kodeky. Na ukázkou jsem do diplomové práce zahrnul jen ty nejpoužívanější protokoly hlasové a zvukové. Podobným tématem jsem se už zabýval ve své bakalářské práci, proto využiji svoji bakalářskou práci jako základní osnovu a z části teoretický základ pro moji diplomovou práci.

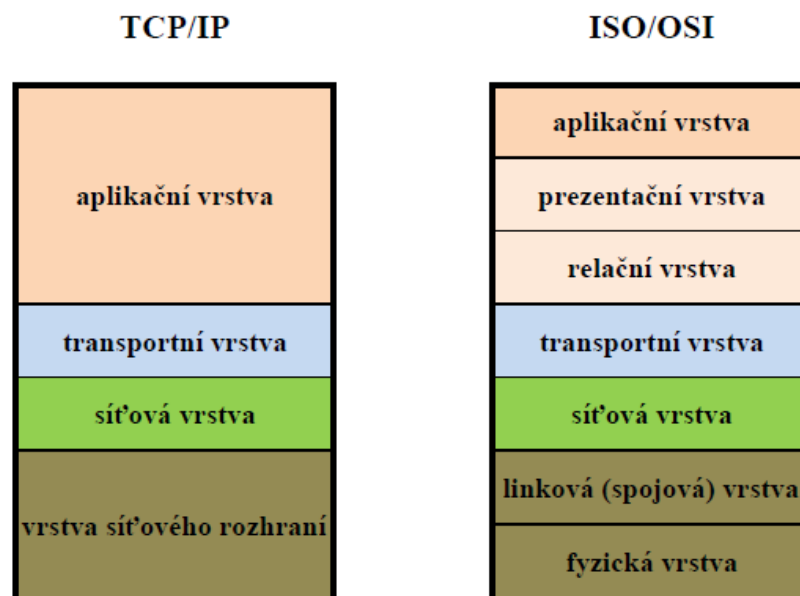
V dnešní době máme hodně možností, jak si můžeme přehrávat zvuk pomocí IP protokolu tedy pomocí internetu. Například to můžou být různé hudební servery (Youtube, Stream, online rádia, televize atd.). Všechny tyto portály využívají streamování zvuku popř. i obrazu. Výhodou je, že nemusíme nic „stahovat“. Bohužel i zde je nevýhoda a to v podobě odkázání se na rychlost odesílání ze zdroje (serveru, PC, mobilu atd.) k přijímači.

V praktické části se budu věnovat realizaci a zapojení mých zvolených výrobků. Pomocí malého PC a zvukové karty. V mé diplomové práci jsem použil dva typy Raspberry Pi a to starou verzi a tento rok vyšlou (2015) verzi dva. Dále jsem použil USB zvukovou kartu U-Control UCA202, která odpovídá mým požadavkům.

2. Základy IP sítě

Máme dva referenční modely a to jsou ISO/OSI a TCP/IP. Referenční model OSI (*Open Systems Interconnection*) vypracovala organizace ISO (*International Standards Organization*), která se pokoušela vytvořit univerzální síťovou architekturu (pouze jako síťový model bez protokolů). Model OSI má sedm vrstev a byl definován v roce 1983 zmíněnou organizací OSI. Úkolem každé vrstvy je poskytnutí služby vyšší vrstvě. U odesílatele data procházejí směrem dolů od aplikační vrstvy k vrstvě fyzické, kde každá vrstva přidá datu svoje příslušné záhlaví. U příjemce to funguje opačným procesem od fyzické vrstvy k aplikační (každá vrstva si odebere vlastní záhlaví). [1]

Referenční model TCP/IP (*Transmission Control Protocol/Internet Protocol*) obsahuje sadu protokolů pro komunikaci v počítačové síti a je hlavním protokolem celosvětové sítě dnešního Internetu. TCP/IP vznikl v rámci projektu ARPANET v polovině 80. let minulého století. Všechny vrstvy jsou přesně definované. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší. [1]



Obr. 1: Referenční model TCP/IP a ISO/OSI

2.1. Internetové protokoly

Co je to protokol? Protokol je standard (nařízení), podle kterého počítače (koncové zařízení) komunikují a přenášejí mezi sebou data. Protokoly mají různé typy. Mohou být realizovány hardwarově, softwarově nebo kombinací obou. Každý protokol má svou typickou vlastnost a funkci.

2.1.1. Síťová vrstva

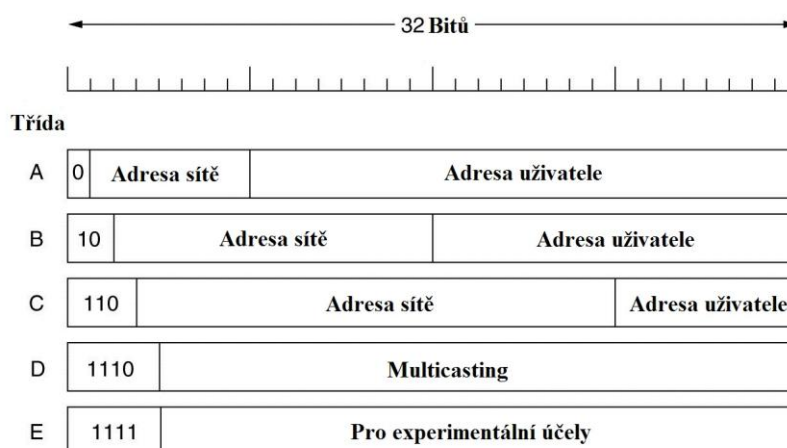
- IP + IPv4 a IPv6

Internet Protocol je základní protokol síťové vrstvy a celého Internetu. Protokol IP poskytuje datagramovou službu celé rodině protokolů TCP/IP. Provádí vysílání datagramů na základě síťových IP adres obsažených v jejich záhlaví. Poskytuje vyšším vrstvám síťovou službu bez spojení. Každý datagram je samostatná datová jednotka, která obsahuje všechny potřebné údaje o adresátovi i odesilatel. Datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě. Jelikož IP protokol není tak spolehlivý, proto musíme zjistit spolehlivost ve vyšších vrstvách čili protokolem TCP nebo aplikací.

IP adresa je číslo, které nám jednoznačně určuje síťové rozhraní v počítačových sítích. V dnešní době se používá IPv4. Jelikož v roce 2011 došlo k rozdělení posledních bloků IP adres, tak se bude (má) přecházet na novější verzi IPv6.

- IPv4

IPv4 je 32bitová adresa, která je zapsaná dekadicky (0 - 9) po jednotlivých oktetech (8 bitů) oddělena tečkou. Např. 192.168.1.150. Takových to čísel existuje celkem $2^{32} = 4\,294\,967\,296$. Určitá část adres je rezervována pro vnitřní potřeby protokolu a nemohou být přiděleny (viz níže). IP adresa se dělí do několika tříd (obr. 2). V tabulce jsou shrnuty všechny důležité informace, jako je např. počet sítí a uživatelů v ní. [1]



Obr. 2: Třídy IP

třída	začátek bin. kódu	první bajt (8 bitů)	maska	počet	
				síť	uživ. v síti
A	0	0 - 127	255.0.0.0	2^7	2^{24}
B	10	128 - 191	255.255.0.0	2^{14}	2^{16}
C	110	192 - 223	255.255.255.0	2^{21}	2^8
D	1110	224 - 239	multicast		
E	11111	240 - 255	rezerva		

Tab. 1: Třídy IP s parametry

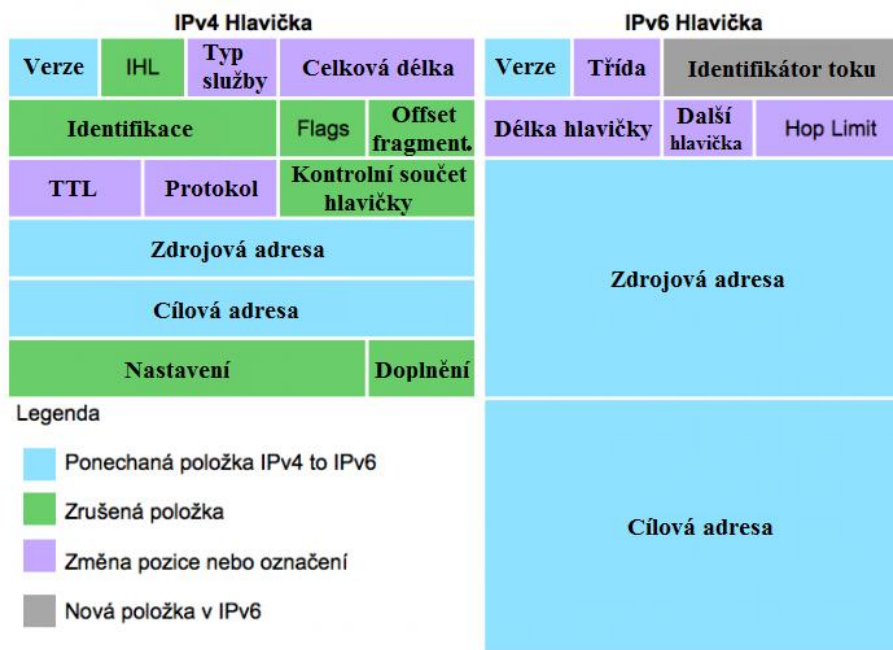
U IP adres máme také speciální vyhrazené IP adresy. Tyto IP adresy nemohou, jak již bylo zmíněno, být použity pro veřejnou síť. Jako jsou například:

- 0.0.0.0 – odkaz na neznámou síť, „tento počítač v této síti“
- 255.255.255.255 – broadcast přes všechny sítě
- privátní sítě: třída A: 10.0.0.0 – 10.255.255.255
třída B: 172.16.0.0 – 172.31.255.255
třída C: 192.168.0.0 – 192.168.255.255
- celá třída E je určena především na vědecké účely

- o IPv6

Adresový rozsah IPv6 je oproti IPv4 mnohem větší. Má 128bitovou adresu, která je zapsána hexadecimálně (0 - F) po osmi skupinách oddělenými dvojtečkou. Máme zde tedy 2^{128} možných adres. Např. 2015:08a3:08b3:08c3:08d3:9934:0db8:8a2e. [1]

Na obrázku jde vidět rozdíl mezi hlavičkami IPv4 a IPv6.

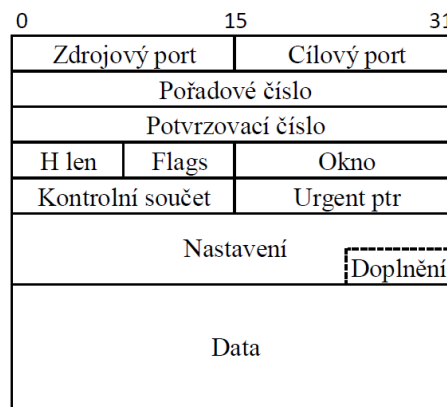


Obr. 3: Rozdíl mezi IPv4 a IPv6 - převzato [8]

2.1.2. Transportní vrstva

- TCP

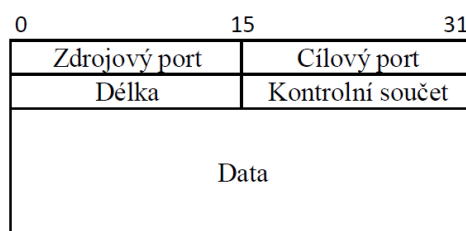
TCP (*Transmission Protocol Control*) je jeden ze základních prvků protokolů internetu. Vytváří virtuální okruh mezi koncovými aplikacemi, tedy spolehlivý přenos dat. Protokol je vytvořen tak, aby garantoval spolehlivé doručování a zároveň doručování ve správném pořadí. TCP je hodně využívaným protokolem v internetu. V internetu ho využívají různé protokoly, které pak zároveň rozlišuje podle portů např. HTTP (port 80), SMTP (port 25), DNS (port 53), FTP (port 21) a další. [1]



Obr. 4: Hlavička TCP

- UDP

UDP (*User Datagram Protocol*) je „opak“ TCP. Je to tzv. nespolehlivý protokol. UDP protokol přenáší datagramy mezi počítači v síti, ale na rozdíl od TCP nezaručuje, zda přenášený paket neztratí či nezmění pořadí paketů nebo zda některý paket nedoručí vícekrát. Velkým problémem je, že aplikace po ztrátě jednoho paketu nemůže dostat následující pakety do té doby, dokud není ztracený paket znovu poslán. To způsobuje hlavně problémy u real-timeových aplikací jako je internetové rádio, reálné multiplayerové hry a VoIP. Zde se klade důraz na rychlost doručení data, než je dostávat ve správném pořadí a kompletní. UDP protokol používá podobně jako TCP čísla portů. [1]



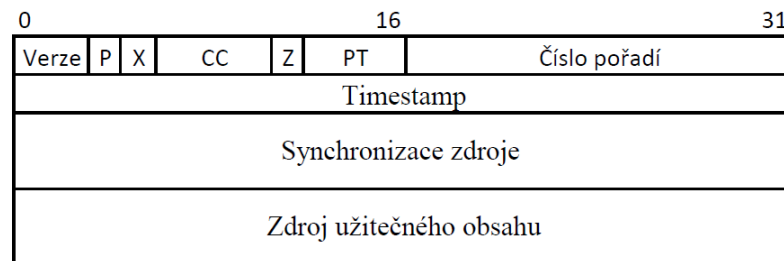
Obr. 5: Hlavička UDP

2.1.3. Aplikační vrstva

- **RTP**

RTP (*Real-time Transport Protocol*) je jeden z důležitých protokolů pro doručování zvukových a obrazových (neboli video) dat po internetu. Využívá již zmiňovaný protokol UDP. Tento protokol je oproti ostatním „novější“. Byl poprvé publikován v roce 1996. Vyvinula ho Audio-Video Transport Working Group.

Tento protokol se nejčastěji používá pro proudové přenosy, telefonie, videohovory a celkově na jakýkoliv stream v reálném čase. Může být použit pro unicastové, tak multicastové služby.



Obr. 6: Hlavička RTP [9]

Vysvětlivky:

- P - 1 bit, slouží, jako doplnění např. používají ho některé šifrovací algoritmy
- X - 1 bit, slouží pro rozšíření (více informací v hlavičce RTP)
- CC - 4 bity, počet identifikátorů (zdrojů užitečného obsahu)
- Z - 1 bit, slouží jako záložka
- PT - 7 bitů, popisuje formát užitečného obsahu
- Timestamp - 32 bitů, moment kdy odebíráme první vzorek z užitečného obsahu

3. Kodeky a jejich vlastnosti

Tato kapitola je stěžejní pro tento projekt. Zjistíme co je to kodek a na co ho potřebujeme. Dále si řekneme a objasníme pár důležitých druhů kódování, jako jsou např. PCM atd. Podíváme se na ty nejvíce používané kódování a zjistíme, na jakém principu pracují. Dále si probereme jednotlivé druhy kodeků, jak hlasových tak zvukových. Větší část textu je převzata z bakalářské práce [1].

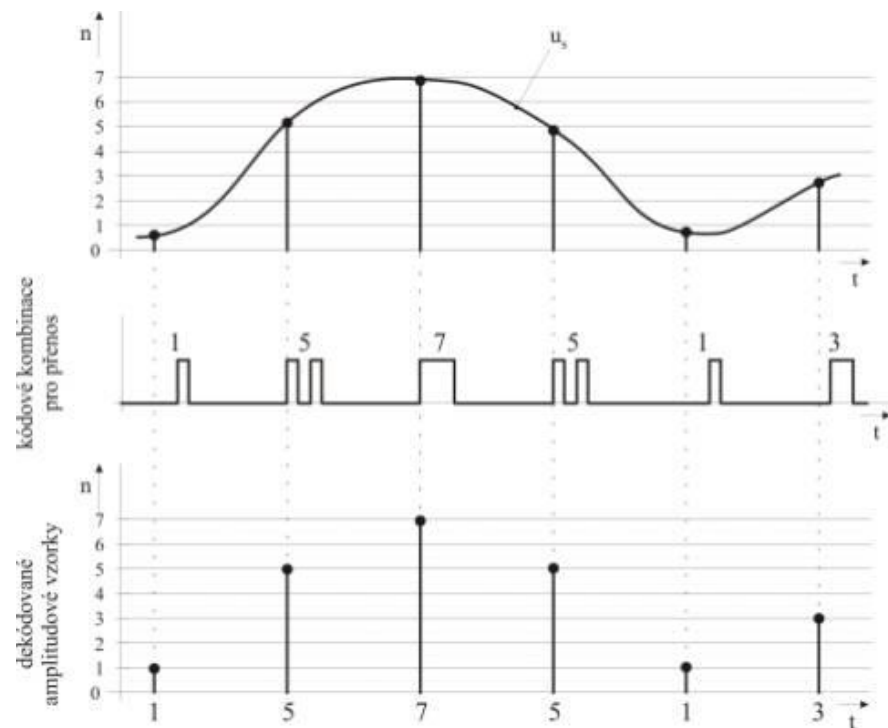
3.1. Druhy kódování

Kódování je převod data, jako je např. zvuk, obraz, atd. pomocí stanoveného kódu nebo kódové tabulky. Např. obrázek se kóduje tak, že se rozloží na určité pixely (čtverce v obrazu) a ty se pak zakódují. My se zde budeme zabývat kódováním zvuku. [1] [2]

3.1.1. PCM

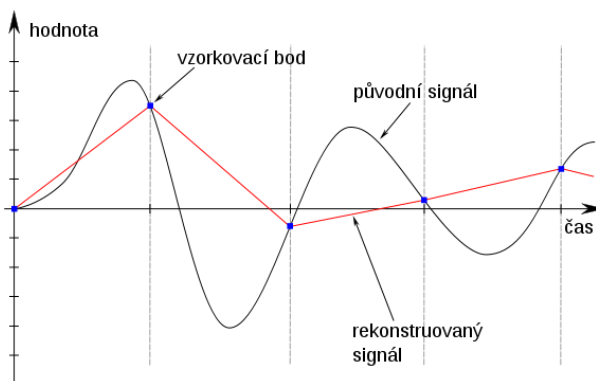
PCM (*Pulse Code Modulation*) je modulační metoda převodu analogového zvukového signálu na signál digitální. Je to nejznámějším a nejpoužívanějším kódováním. Kódování PCM se skládá ze tří částí:

- vzorkování – je proces, kde hodnota analogového signálu se odečítá v diskrétním čase (viz Obr. 9 a Obr. 8)
- kvantování – je diskretizace oboru hodnot signálu, je to proces ztrátový a nevratný
- komprese – diskrétní hodnoty jsou reprezentovány pomocí n-bitových čísel, čím více bitů jeden vzorek obsahuje, tím menší chyba při PCM kódování vzniká (objem dat se zvětšuje)

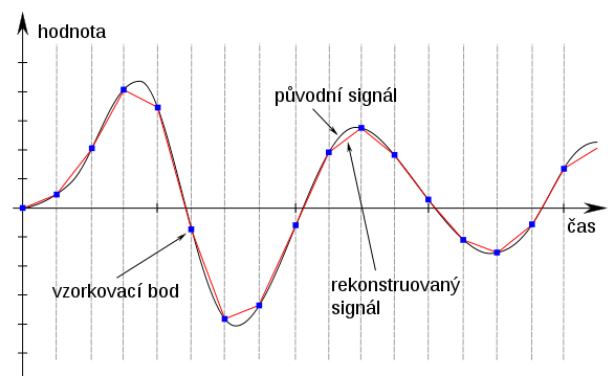


Obr. 7: Princip PCM (modulace a demodulace)

Na horním obrázku (Obr. 7) vidíme, jaký je princip PCM modulace a demodulace. Na následujících dvou obrázcích máme zobrazený signál, který je vzorkován s nízkou frekvencí (Obr. 9 převzat z [10]). Zde vidíme, že zrekonstruovaný signál je hodně špatný (vůbec se nepodobá původnímu signálu). A signál, který je vzorkován při vyšší frekvenci (Obr. 8 převzat z [10]).



Obr. 9: Signál s nízkou vzorkovací frekvencí



Obr. 8: Signál s vysokou vzorkovací frekvencí

Zde je stručná tabulka, ve které jsou popsány nejdůležitější parametry týkající se kódování různé kvality zvuku.

kvalita digitálního záznamu	rozlišení a kanály	vzorkovací frekvence
Telefonní kvalita	8 bit / mono	11025 Hz
Rozhlasová kvalita	8 bit / mono	22050 Hz
CD kvalita	16 bit / stereo	44100 Hz
DVD kvalita	24 bit / 5+1 zvuk	192000 Hz

Tab. 2: Porovnání kvality záznamu a jejich parametry

3.1.2. DPCM

DPCM (*Differential Pulse Code Modulation*) je druh PCM kódování a používá se na snížení množství přenesených dat. Navzorkovaná data se nekódují, ale pouze jejich rozdíl oproti odhadnutému průběhu signálu. Jelikož navzorkovaný průběh a odhadnutý průběh jsou si značně podobné, tak jejich výsledný rozdíl má mnohem menší dynamický rozsah a je možné ho tedy zakódovat pomocí menšího počtu bitů (množství dat se snižuje) [1].

3.1.3. ADPCM

ADPCM (*Adaptive Differential Pulse Code Modulation*) je varianta DPCM kódování, která se liší velikostí kvantizačního kroku. Byl vyvinut v roce 1970 pro kódování hlasu. Je to vylepšená verze, která se adaptivně přizpůsobuje. Výsledkem je opět snížení dynamického rozsahu oproti DPCM a zároveň menší počet bitů k zakódování [1].

3.1.4. LPC

LPC (*Linear Predictive Coding*) je kódovací algoritmus ztrátové komprese. Pracuje na jiném principu než PCM či ADPCM, ty pracovali s průběhem signálu. Kdežto metoda LPC vychází z předpokladu, že hlas je generován zdrojem (hlasivkami v krku). Metoda se snaží vytvořit model hlasového ústrojí člověka. Kde na hlasivkové šterbině vzniká bzukot, který charakterizuje hlasitost, frekvenci a intenzitu generovaných zvuků v krku a ústech, které tvoří jakousi „trubku“, která pak způsobuje různé rezonance, což dává hlasu jeho charakteristické zabarvení pro každého člověka. LPC analyzuje řeč prostřednictvím filtrů (inversní filtrace). Inversní filtrace je odstranění vyfiltrovaných signálů (např. ticho). Zbytek signálu se zpracuje. A opět převede na podobný původní signál řeči. Pro přijatelnou kvalitu výsledné řeči se používá 30 až 50 rámců za vteřinu (opakování vstupního signálu) [1].

3.1.5. CELP

CELP (*Code Excited Linear Prediction*) je vylepšená verze LPC. V současné době je nejpoužívanějším algoritmem kódované řeči. CELP není konkrétní kodek. Je to spíše obecný termín pro skupinu kodeků. LPC řeší jen signál bzučáku a ostatní složky neumí vyfiltrovat. Nedokáže určit zvuky, které neodpovídají LPC modelu, platí to pro zvuky generované např. různou pozicí jazyku (výslovnost). Kdežto CELP řeší u hlasu i různé „zbytky“ jako je např. šum, sykot, atd. K tomu mu slouží tzv. codebook. Codebook je tabulka obsahující rezidua (zbytek po odfiltrování). Kodér posléze porovná rezidua s tabulkou. Dekodér postupuje opačně. Podle tabulky obnoví rezidua. Výsledkem je lepší hlasový signál [1].

Mezi modifikace CELP patří ACELP (*Algebraic Code Excited Linear Prediction*), LD-CELP (*Low Delay Code Excited Linear Prediction*) a CS-CELP (*Conjugate Structure Algebraic Linear Code Prediction*).

3.2. Druhy kodeků

Kodek je zařízení nebo algoritmus, který slouží ke zmenšení velkého objemu audio či video dat, které jsou jinak příliš obrovské. Slovo kodek vzniklo složením slov kodér a dekodér (neboli **k**omprese a **de**kompresa), tj. zařízení (program) jež je na jedné straně schopné data zakódovat a na druhé straně opět rozkódovat. Kodeky velmi často používají ztrátovou kompresi (existují i bezztrátové) a proto dekódovaná data nejsou totožná s daty, která byla zakódována na začátku. Jako příklad si můžeme uvést mikrofon a zvukovou kartu. Data získaná z mikrofonního vstupu kodér zpracuje a předa je dál. Poslaná data dorazí do dekodéru, který opět převede do původní podoby a pošle na audio výstup zvukové karty. [1]

Často používané širokopásmové zvukové kodeky (hlasové):

- G.711
- G.722
- G.723.1
- G.726
- G.728
- G.729
- GSM
- SILK (Skype)
- Opus
- iLBC

Často používané zvukové (rozhlasové) kodeky:

- MP3
- AAC
- Tieline
- aptX

Na následujících stránkách si probereme jednotlivé kodeky zvláště.

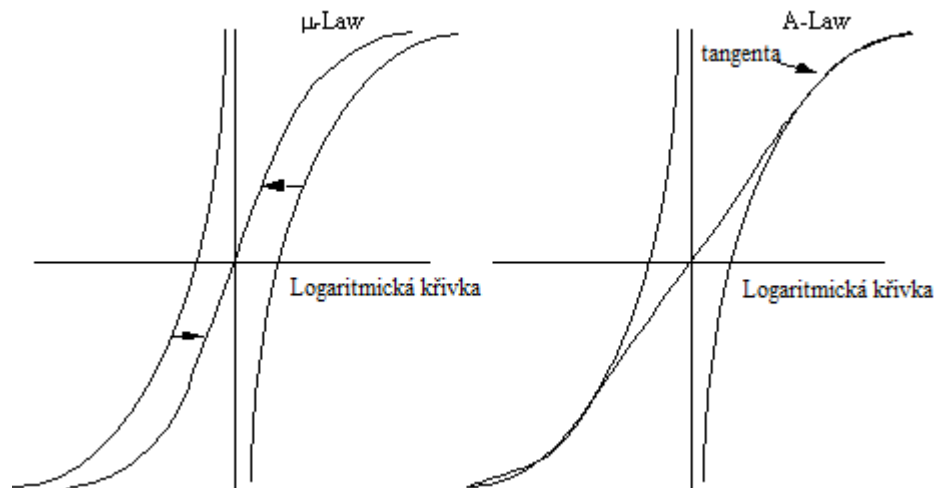
3.2.1. G.711

G.711 (*Pulse Code Modulation of Voice Frequencies*) je základní (nejběžnější, nejrozšířenější a nejjednodušší) kodek, který se používá v telekomunikacích. Byl schválen jako ITU-T standard v roce 1988 [3]. Je určen pro vysoké přenosové rychlosti (64 kbit/s). Kodek G.711 má dvě verze: v Evropě + zbytku světa (a-law) v Severní Americe a Japonsku (μ -law). Liší se pouze v logaritmické kompresi signálu.

V dnešní době máme už verzi G.711.1, která byla schválena v roce 2008 [12]. Je rozšířena o možnost použít vyšší vzorkovací frekvence 16 kHz (oproti 8kHz). Datový tok je tedy 64, 80 nebo 96 kbit/s. G.711.1 je kompatibilní s G.711 na 64 kbit/s a proto efektivní nasazení do stávajících G.711 na bázi Voice over IP (VoIP infrastruktury) se nepředpokládá.

Dále máme verzi G.711.0 také je známá jako G.711 LLC. Tato verze využívá bezztrátové komprese dat ke snížení šířky pásma až o 50%.

V následujícím obrázku (Obr. 10 převzat a upraven z [11]) můžeme vidět rozdíl mezi a-law a μ -law. Z obrázku je pak patrné, že u a-law je tangenta a u μ -law je složena z logaritmických křivek.



Obr. 10: Porovnání a-law s μ -law

G.711 poskytuje dvě rozšířené vrstvy. První prodloužení zlepšuje kvalitu G.711 na nízkých frekvencích (50-300 Hz), obvykle přenášená pomocí G.711. Druhé prodloužení kóduje na vysokých frekvencích až 7 kHz pomocí různých rozšířených vrstev. G.711.1 funguje na 80 nebo 96 kbit/s v závislosti na použití jedné nebo dvou prodlužovacích vrstev [12].

G.711		64 Kbps
G.711	50-300 Hz	80 Kbps
G.711	4-7 KHz	80 Kbps
G.711	50-300 Hz 4-7 KHz	96 Kbps

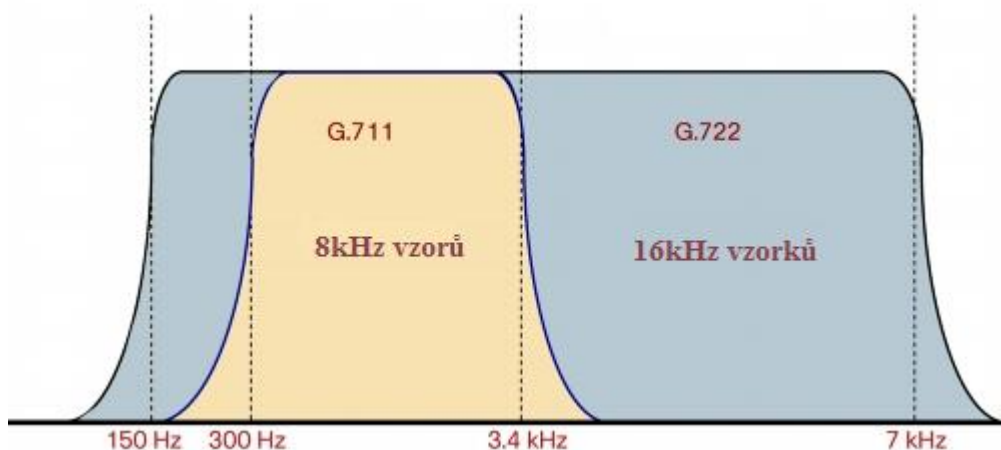
Obr. 11: G.711 s prodlužovacíma vrstvami

3.2.2. G.722

G.722 je kodek standardu ITU-T (schválen 1988), který poskytuje 7kHz širokopásmový zvuk s datovou rychlostí 48, 56 a 64 kbit/s. Používá kódování SB-ADPCM (díličí pásmo ADPCM). Budeme-li mít frekvenci 16 kHz (při použití 14 bitů), tak se nám zdvojnásobí čistota a kvalita zvuku.

Další kodeky se 7kHz šířkou pásma jsou G.722.1 a G.722.2. Tyto kodeky jsou varianty G.722 a používají různé kompresní patentované technologie. G.722.1 je založena na Siren7 kodeku a nabízí nižší přenosové rychlosti komprese. Novější typ G.722.2 také známý pod názvem jako AMR-WB (*Adaptive Multirate Wideband*) je založena na kódování ACELP a ten nabízí nižší přenosové rychlosti komprese, stejně jako schopnost rychle se přizpůsobit různým kompresím. Při přetížení sítě se šířka pásma vrátí na normální (původní) úroveň, sníží se komprese a tak se opět vyšší kvalita zvuku.

Na obrázku (Obr. 12 převzat a upraven [13]) je zobrazen kodek G.711 a G.722. Je na něm znázorněna šířka pásma při 64 kbit/s.



Obr. 12: Porovnání G.711 a G.722

3.2.3. G.723.1

G.723.1 (*Dual Rate Speech Coder*) byl schválen jako ITU-T standard v roce 1996. U tohoto kodeku máme dvě přenosové rychlosti. A to 6,3 kbit/s (s algoritmem MP-MLQ) ten se používá k lepší kvalitě kódování hlasu a 5,3 kbit/s (s algoritmem ACELP) ten je zase lépe přizpůsobený pro použití v sítích se smíšeným provozem (hlas/data). Tento kodek se používá většinou u VoIP kvůli jeho nízkým požadavkům na šířku pásma. G.723.1 je zvukový kodek v H.324 ITU-T, který je doporučen pro H.324 terminály nabízející audio komunikace.

Kodek G.723 provádí konverzi analogového signálu na datový tok 64 kbit/s (PCM) a také používá vícepásmový digitální filtr. Ten zvolí frekvenci hlásek. Ty analyzuje a vysílá na kanálu IP pouze informace o aktuálním stavu fonému v řeči signálu. Tato transformace (algoritmus) umožňuje snížit míru kódované informace na 5,3 - 6,3 kbit/s bez viditelné zhoršení kvality řeči.

3.2.4. G.726

G.726 je hlasový kodek, který byl schválen v roce 1990. Kodek má datový tok 32 kbit/s (standard). Je složený z kodeků G.721 (byl představen v roce 1984) a kodekem G.723 (byl představen v roce 1988). Datový tok může být 16, 24, 32 nebo 40 kbit/s. Kodek používá kódování ADPCM. G.721 je standardní kodek aplikovaný u bezdrátových telefonů DECT. Může být také používán ve spojení s poloviční rychlostí kodeku G.711 (32 kbit/s) což zdvojnásobuje využitelné kapacity sítě. Dále je určen k používání u videokonferencí.

3.2.5. G.728

G.728 je hlasový kodek, který byl standardizován ITU-T v roce 1992. Datový tok u tohoto kodeku je 16 kbit/s. Používá kódování LD-CELP (*Low - Delay Code Excited Linear Prediction*). Kodek se používá častěji ve videokonferencích než v IP telefonii. Je široce používán pro aplikace, které vyžadují velmi nízké algoritmické zpoždění. Zpoždění kodeku je pouze 5 vzorků (0,625 ms).

3.2.6. G.729

G.729 je hlasový kodek, který byl standardizován v roce 1996. Umožňuje efektivněji využít limitované pásmo pro přenos hlasu. Má dvě hlavní verze: verzi A a B. Nabízí nízký datový tok 8 kbit/s s použitím metody CS-ACELP. Je to však "náročný" kodek z hlediska času na zpracování procesoru, a proto pro některé VoIP telefony a adaptéry mohou být špatnou volbou. Některé telefony zvládnou pouze jeden G.729 hovor najednou. To může způsobit při volání problémy, jestliže je povolen pouze kodek G.729.

CS-ACELP kodér pracuje s rámci řeči o délce 10 ms, což při vzorkovací frekvenci 8000 Hz odpovídá 80 vzorkům. Každý 10 ms rámec řečového signálu je analyzován pro získání parametrů CELP modelu (koeficienty LPC, indexy slov z pevné a adaptivní kódové knihy, zisk). Tyto parametry jsou kódovány a přenášeny komunikačním kanálem.

V dekodéru jsou tyto parametry použity k obnovení excitačního signálu a koeficientů syntetického filtru. Řečový signál je pak rekonstruován filtrací excitačního vektoru přes syntetický filtr.

3.2.7. GSM

GSM (*Global System for Mobile communications*) je digitální mobilní radiový systém, který je používán v celé Evropě. Původně byly jen dva kodeky (FR a HR). Tyto používali systém založený na LPC kódováním. Po sléze se přidali další.

V GSM telefonech rozlišujeme tyto základní typy kodeků a způsobů kódování:

- FR

GSM (*Full-Rate*) byl vyvinut na počátku roku 1990. Jeho přenosová rychlost je 13 kbit/s. Kodek používá lineární prediktivní kódování (RPE-LTP). Měl první digitální řečové kódování používané v digitálních systémech GSM jako je např. mobilní telefon [7].

- EFR

GSM (*Enhanced Full-Rate*) byl vyvinut v roce 1995, kdy ho ETSI vybrala jako průmyslový kodek používaný u sítě GSM. Byl vyvinut za účelem zlepšení kvality (nástupce za Full-Rate). Jeho přenosová rychlost je 12,2 kbit/s. Používá kódování ACELP bylo lepší než co používal Full-Rate (RPE-LTP).

- HR

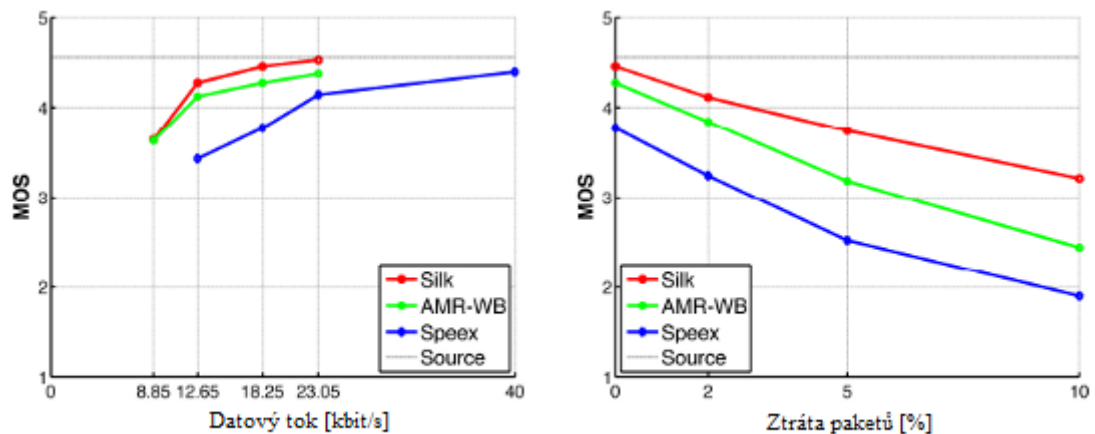
GSM (*Half-Rate*) byl také vyvinut v roce 1990 [7]. Jeho přenosová rychlost je 5,3 kbit/s. Kodek používá vektorový součet VSELP (technika pro kompresi řeči). Také byl primárně vyvinut pro mobilní telefony.

- AMR

AMR (*Adaptive Multi-rate*) je nejvíce používaným kodekem v GSM. Byl vyvinut v roce 1988. AMR pracuje v osmi přenosových rychlostech v rozmezí od 4,75 do 12,2 kbps a byl speciálně navržen pro zlepšení propojení robustnosti. Používá kódování ACELP. Kodek používá různé komprese dat např., pokud je ticho, tak nic nepřenáší, dává zpětnou vazbu uživateli, že je spojení stále připojené, atd.

3.2.8. SILK (v programu Skype)

SILK je audio kompresní formát. Tento kodek je využíván u programu Skype, který byl vyvinut Skype Limited v roce 2009 [14]. Jeho přenosová rychlost je 6 až 40 kbit/s. SILK může použít vzorkovací frekvenci 8, 12, 16 nebo 24 kHz. Při vyšší přenosové rychlosti poskytuje lepší kvalitu zvuku díky nižšímu kvantizačnímu šumu. Může také využít nízkého algoritmického zpoždění 25ms. Komprimované audio je obsažen v balení po 20, 40, 60, 80 nebo 100 ms. Je volně ke stažení. Obrázek níže je převzat z [14].



Obr. 13: Přehled MOS/datový tok [kbit/s] + MOS/ztráta paketů [%]

3.2.9. Opus

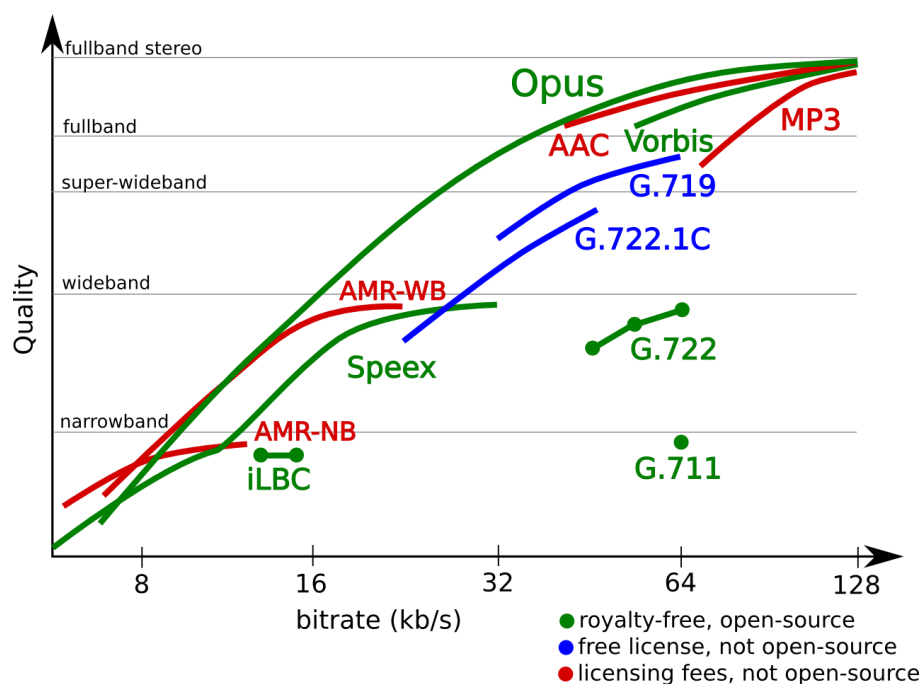
Opus je otevřený a tzv. „royalty-free“. Je to vysoce univerzální zvukový formát, který je ztrátový. Byl navrhnout na internetové využití, jako jsou přenosy hlasu, videokonference, streamování hudby atd. Má velký potenciál nahradit dosavadní audio komprese na internetu. Tento zvukový formát je standardizován IETF. V roce 2012 byla vytvořena první verze. Dnes už je verze 1.1.1-beta (říjen 2014). [16]

Opus „stojí“ na dvou hlavních pilířích a to na formátu SILK (vyvinut společností Skype viz kapitola SILK (v programu Skype) 3.2.8) a Celt (vyvinut Xiph.org Foundation). Jak už bylo zmíněno, opus je bezplatný, kromě pár patentů, které jsou s tímto formátem spjaty.

Parametry zvukového formátu jsou: datová rychlost od 6 – 510 kbit/s, frekvence je od 8 kHz do 48 kHz, velikost rámců je od 2,5 – 60 ms, podpora až 255 kanálů, mono, stereo, řeč a hudba.

Hlavní rozdíl oproti zvukovým formátům pro vysokou kvalitu audia (AAC, MP3) je jeho nízká latence. Kde naopak u zvukových formátů s nízkou latencí (G.711, GSM) má opus vysokou kvalitu audia.

Obrázek níže popisuje kvalitu různých kodeků v závislosti na bitové rychlosti. Z obrázku je patrné, jak sami autoři zvukového formátu zamýšleli to, že opus bude lepší než většina již zmiňovaných formátů. Obrázek je převzat z [16].



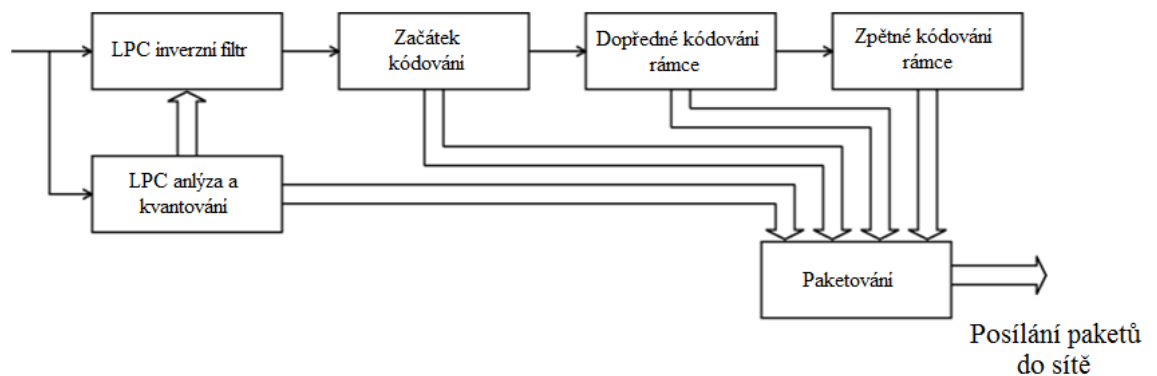
Obr. 14: Porovnání kvality na bitové rychlosti

3.2.10. iLBC

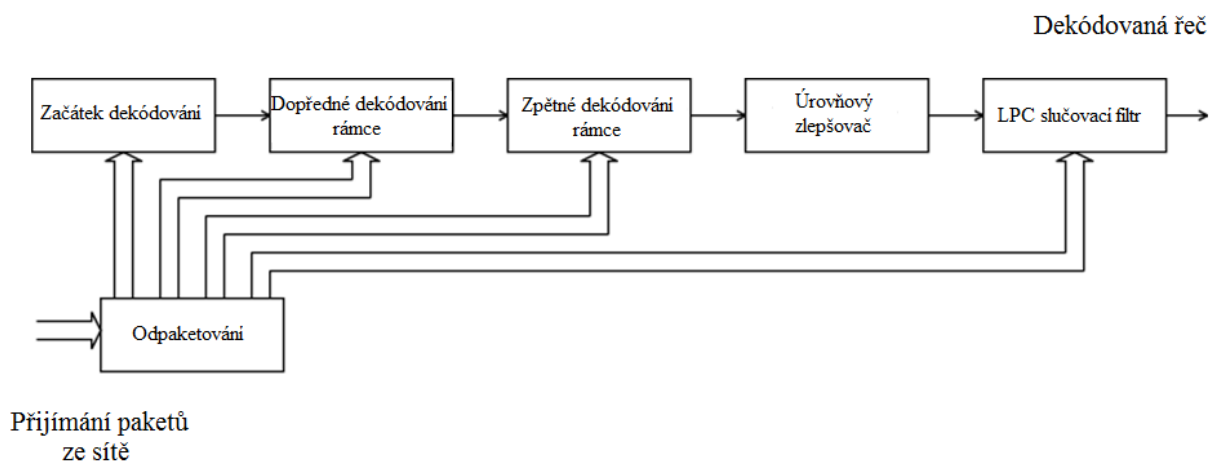
iLBC (*Internet Low Bitrate Codec*) je open-source kodek (zcela zdarma) a slouží pro mluvené slovo, což je vhodné pro robustní hlasové komunikace přes IP (také pro streamování audio). Byl standardizován Global IP Sound (GIPS) v roce 2004. Kodek je určen pro úzké řečové pásmo. Datový tok má rychlost 13,33 kbit/s a kódování rámečků délku 30 ms. U datového toku při rychlosti 15,2 kbit/s má rámeček délku 20 ms. Hlavní výhodou použití iLBC je zlepšení kvality hlasu a to zejména v IP sítích. iLBC řečové snímky jsou komprimovány nezávisle. To umožňuje lepší kodekovou toleranci pro ztráty paketů. U iLBC kodeku může dojít tedy ke zhoršení kvality řeči v případě ztráty snímku, které se vyskytují v souvislosti se ztracenými nebo zpožděnými IP pakety [3] [1].

V následujících dvou obrázcích je vysvětleno kódování a dekódování iLBC kodeku.

Příchozí řeč



Obr. 16: Kódování iLBC



Obr. 15: Dekódování iLBC

3.2.11. MP3

MP3 (známe také pod názvem *MPEG-1 nebo MPEG-2 Audio Layer III*) je zvukový formát, který používá formu ztrátové komprese. Byl navržen v MPEG (Motion Picture Experts Group) pro standard MPEG-1, který byl publikován v roce 1993 a později pro standard MPEG-2. Ten byl publikován o 2 roky později. U kodeku MP3 se nejčastěji setkáváme s datovým tokem 128 kbit/s. Poměr mezi CD kvalitou (WAV) a MP3 je "přibližně" 1/11 datového toku. Tato hodnota představuje pomyslnou hranici, kde lidské ucho ještě nepozná kazy a vady komprimace. Hodnota MOS je mezi 2,22 až 3,66, podle toho jaký byl zvolen snímač zvuku [1].

Srovnání přenosových rychlostí je v Tab. 4 [15]. V další tabulce (Tab. 3) máme pak srovnání vzorkovacích frekvencí podle standardů [15].

MPEG-1	MPEG-2	MPEG-2.5
-	8	8
-	16	16
-	24	24
32	32	32
40	40	40
48	48	48
56	56	56
64	64	64
80	80	80
96	96	-
112	112	-
128	128	-
144	144	-
160	160	-
192	-	-
224	-	-
256	-	-
320	-	-

Tab. 4: Srovnání přenosových rychlostí

MPEG-1	MPEG-2	MPEG-2.5
-	-	8000
-	-	11025
-	-	1200
-	16000	-
-	22050	-
-	24000	-
32000	-	-
44100	-	-
48000	-	-

Tab. 3: Srovnání vzorkovacích frekvencí

V dnešní době je formát MP3 jedním z oblíbených formátů při uchovávání a přehrávání hudby na počítačích a veškerých moderních zařízeních. A to hlavně díky své malé velikosti oproti CD kvalitě a jeho rozsahu frekvencí.

3.2.12. AAC

AAC (*Advanced Audio Coding*) je jeden z nejpokročilejších formátů komprese zvuku (ztrátová komprese). Byl vyvinut jako následovník zvukového formátu MP3. Dosahuje lepší zvukové kvality záznamu než u MP3 při podobné bitové rychlosti. Tento zvukový formát je standardizován ISO a IEC. Je stanoven v MPEG-2 část 7. První vydání bylo v roce 1997 [17].

Parametry zvukového formátu jsou: datová rychlost do 256 kbit/s na kanál, frekvence je od 8 kHz do 96 kHz, podpora až 48 kanálů při celém frekvenčním rozsahu plus 16 LFE (*low frequency effects*) kde je omezena frekvence na 120 Hz .

AAC formát obsahuje mnoho upravených AAC profilů. V praxi se moc nepoužívá obecný AAC formát, ale spíše jeho upravené a vylepšené verze. Má tyto tři „základní“ profily (typy): AAC-LC nebo také LC-AAC (*Low-Complexity*), AAC Main (*Main*) a AAC-SSR (*Scalable Sampling Rate*). Později v roce 1999 zařazen do MPEG-4 část 3.

Jednotlivé typy AAC:

- AAC-LC

AAC (*Low-Complexity*) je profil s nízkou složitostí. Tento profil je nejpoužívanější, nejjednodušší a nejvíce podporovaný formát. AAC-LC je lepší pro hovor, kdežto pro hudbu a zpěv je vhodnější HE-AAC. [17]. Na Obr. 18 je znázorněn vývoj AAC profilů.

- AAC-Main

AAC (*Main*) funguje jako predikce v případě bezztrátové komprese zvuku, probíhá však ve frekvenční (ne časové doméně). Tento formát je výpočetně více náročnější. Vytvořen pro nízké datové toky, ale jelikož výsledky nebyly dostačující. Lepší výsledky jsou u HE-AAC. Nezískal tento profil velkou podporu. [17].

- AAC-SSR

AAC (*Scalable Sampling Rate*) je profil se škálovatelnou vzorkovací frekvencí. Vytvořen firmou Sony. Před aplikováním MDCT (*Modified Discrete Cosine Transform*) se zvukový záznam rozdělí na 4 subpásma. Výhoda je v adaptovatelnosti jednotlivých subpásem na složitost signálu (pro malé frekvence zvýším frekvenční rozsah a pro malé frekvence zvýším časový rozsah). Nevýhodou je vznik zkreslení při zpracování signálu. Proto se dnes už v praxi nepoužívá [17].

- AAC-LTP

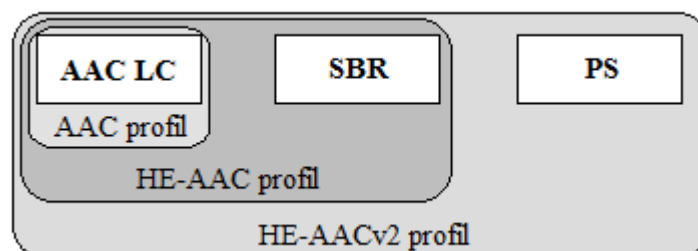
AAC (*Long Term Prediction*) je profil s dlouhou predikcí. Je zde použit LC s jednoduchou dopředně adaptivní predikcí (v časové doméně). Taktéž jako SSR se dnes již už nepoužívá [17].

- AAC-LD

AAC (*Low Delay*) je profil z krátkým zpožděním. Opět využívá stávající LC, tak jak LTP, kde omezuje latenci mezi vstupem a výstupem kodéru. Latence se pohybuje v hodnotách přibližně 20 ms, kde standardní hodnota se pohybuje kolem 100 ms [17].

- HE-AAC

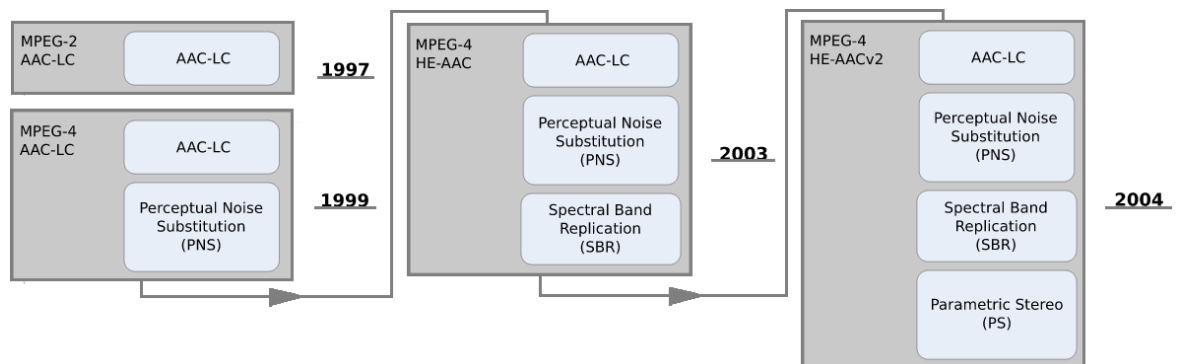
HE (*High Efficiency*) AAC nebo taktéž AACplus je vysokoefektivní formát, který je ztrátový. Je hlavně optimalizován pro nízké datové rychlosti jako je např. streamování hudebních služeb. HE-AAC má dvě verze a to verze 1 (označována jako HE-AACv1) a verze 2 (označována jako HE-AACv2). Hierarchická struktura AAC profilu je zobrazena v Obr. 17.



Obr. 17: Hierarchická struktura AAC profilu

HE-AACv1 používá SBR (*Spectral Band Replication*) pro zvýšení účinnosti komprese ve frekvenční oblasti (využívá kombinaci AAC-LC s SBR). Je vhodný pro streamování hlasových přenosů. Funguje tak, že reprodukuje vyšší frekvence, díky přenosu nižších frekvencí, které se ukládají (v šířce pásma) 1,5 kbit/s [18].

HE-AACv2 používá PS (*Parametric Stereo*) pro zvýšení účinnosti komprese stereo signálu (využívá kombinaci AAC-LC, SBR a PS). PS (parametrické stereo) spočívá v analýze stereofonní informace v záznamu, kde vytváří postranní informace o rozdílech mezi levým a pravým kanálem. Původní 2 signály jsou sečteny a následně komprimovány jako mono záznam. Díky tomu má při datovém toku mezi 48 a 64 kbit/s vynikající výkon v porovnání s konkurencí [17].



Obr. 18: Vývoj AAC s roky

3.2.13. Tieline

Tieline je společnost zabývající se kodeky. Jejich kodeky nesou stejný název jako firma. Tieline kodeky se specializují na spolehlivost, nízké zpoždění vysílací kvality zvuku při nízké přenosové rychlosti v telekomunikačních sítích. Kódové zpoždění je podobné jako u AAC-LD. Všechny Tieline kodeky obsahují různé sady algoritmů, které se upravují dle vlastních potřeb [19]. V Tab. 5 (viz níže) je srovnání Tieline kodeků s jinými (tabulka je upravena a převzata z [19]).

Máme několik typů kodeků [19]:

- TIELINE MUSIC

Tieline Music je hlavně optimalizován pro přehrávání hudby a hlasu, při nízkých přenosových rychlostech. Poskytuje až 15 kHz v mono kvalitě zvuku s 20ms kódovým zpožděním a s přenosovou rychlostí nižší jak 24 kbit/s. Tyto parametry jsou vhodné pro telefonní linky, internetové připojení a GSM.

- TIELINE MUSICPLUS

Tieline MusicPlus má vysokou toleranční ztrátu pro IP připojení. Poskytuje až 20 kHz ve stereo kvalitě zvuku s 20ms kódovým zpožděním a s přenosovou rychlostí nižší jak 96 kbit/s. Tyto parametry jsou vhodné v dnešních IP a 3G sítích.

- TIELINE VOICE G3

Tieline Voice G3 poskytuje až 7,5 kHz ve zvukové kvalitě s přenosovou rychlostí nižší jak 9,6 kbit/s. Jeho výkonnost se může zvýšit až na úroveň 14,4 a 16,8kbit/s. Zvuková kvalita je pak podobná s G.722 při 64 kbit/s přes ISDN.

	Tieline Music	Tieline MusicPlus	Tieline Voice G3	AAC-LD	G.722
Minimální přenosová rychlost [kbit/s]	24	48	9,6	64	64
Kódové zpoždění [kHz]	20	20	20	20	8
Kvalita zvuku [ms]	15	20	až 7,5	15	7

Tab. 5: Srovnávací tabulka

3.2.14. AptX

AptX dříve také označován jako Apt-X byl vyvinut v roce 1980 v Belfastu (původní algoritmus). Byl zaměřen na snížení přenosové rychlosti a dosažení významné úspory přenosové rychlosti při zachování kvality zvuku. Byl nejlépe střeženým tajemstvím v profesionálním audio průmyslu [20].

Od svého uvedení se rozrůstal velice rychle. Byl v podobě software, firmware, televizního a rozhlasového vysílání a hlavně u bezdrátového audia. Je podporován ve vysoce výkonných zařízeních. AptX kodek lze použít místo SBC.

Máme několik typů AptX kodeků:

- APTX

Kodek aptX slouží zejména pro bezdrátové audio aplikace a také pro automobilový průmysl. Ale zejména se používá pro real-time streaming (vysoce kvalitní stereofonní zvuk) přes Bluetooth (A2DP). Mezi zdrojovým zařízením (smartphone, tablet nebo notebook) a koncovým zařízením (Bluetooth stereo reproduktor, sluchátka). Ovšem důležité je, aby technologii aptX měli obě zařízení.

- ETHANCED APTX

Ethanced AptX je rozšířená verze AptX (vyvinut 2000). Tato verze poskytuje vysoce kvalitní kódování s poměrem komprese 4:1. Můžeme si vybrat mezi přenosovou rychlostí 16, 20 a 24 bity. Rozšířené aptX zvládne až 4 stereo páry (AES3).

- APTX LIVE

AptX Live je zvukový kodek s nízkou složitostí a je ideální pro poskytování vícekanálového zvuku. Vhodný pro bezdrátové mikrofony, bezdrátové telefony a monitorovací zařízení. Kompresní poměr je 8:1 pro 16 bitový digitální zvukový tok a pro 24 bitový zvukový tok je kompresní poměr 12:1 [21].

- APTX LOSSLESS

AptX Lossless (bezztrátový) podporuje zvuk s vysokým rozlišením až do vzorkovací frekvence 96 kHz. Je tedy určen pro profesionální a spotřebitelské aplikace. Velikost komprese dosažitelná s čistým bezeztrátovým kódováním je závislá na charakteru zvukového materiálu. V průměru se kompresní poměr pohybuje v hodnotě 2:1 v 16 bitovém režimu.

3.3. Porovnání některých kodeků

V následujících tabulkách a grafu zjistíme, který kodek (vybrány jen některé) je nejrychlejší a který je naopak nejpomalejší. A vysvětlíme si pojem MOS. Tyto výsledky jsem převzal s mé bakalářské práce [1].

3.3.1. MOS

MOS (*Mean opinion score*) je hodnocení telefonní sítě z pohledu uživatele na kvalitu sítě. Hodnocení u multimediálních kodeků (audio, hlasové i video telefonie), zvláště když jsou pásma používána ke kompresi. Střední průměrné hodnocení (MOS) je číslo, které stanoví kvalitu médií po kompresi a nebo přenosu. MOS je vyjádřeno jako číslo v rozmezí 1 až 5, kde 1 je nejnižší vnímaná kvalita zvuku a 5 je naopak nejvyšší možnou naměřenou (a zároveň vnímanou) kvalitou zvuku. MOS testy pro hlasové kodeky jsou stanoveny podle doporučení ITU-T P.800 [1].

MOS	kvalita	zhoršení
5	výborná	nepatrné
4	dobrá	znatelné, ale ne nepříjemné
3	střední	mírně nepříjemné
2	špatná	nepříjemné
1	slabá	velmi nepříjemné

Tab. 6: MOS tabulka hodnocení

Kompenzační / dekompenzační kodek systémů a zpracování digitálního signálu (DSP) se běžně používají v hlasové komunikaci. Může být nakonfigurován pro zachování šířky pásma. Zde máme kompromis mezi kvalitou hlasu a šířkou pásma ochrany. Nejlepší kodeky poskytují největší šířku pásma ochrany a zároveň při výrobě přenášejí nejmenší zhoršení kvality hlasu. Šířka pásma lze měřit vícekrát [1].

Nevýhodou při získávání MOS výsledků je, že jsou jak časově, tak i finančně velice náročné. Každé vyhodnocení vyžaduje najímání řadu expertů, kteří výsledky většinou odhadují.

Nejčastějším vyhodnocením MOS (na doporučení podle ITU-T P.800) je použití některých anglických vět jako jsou např.:

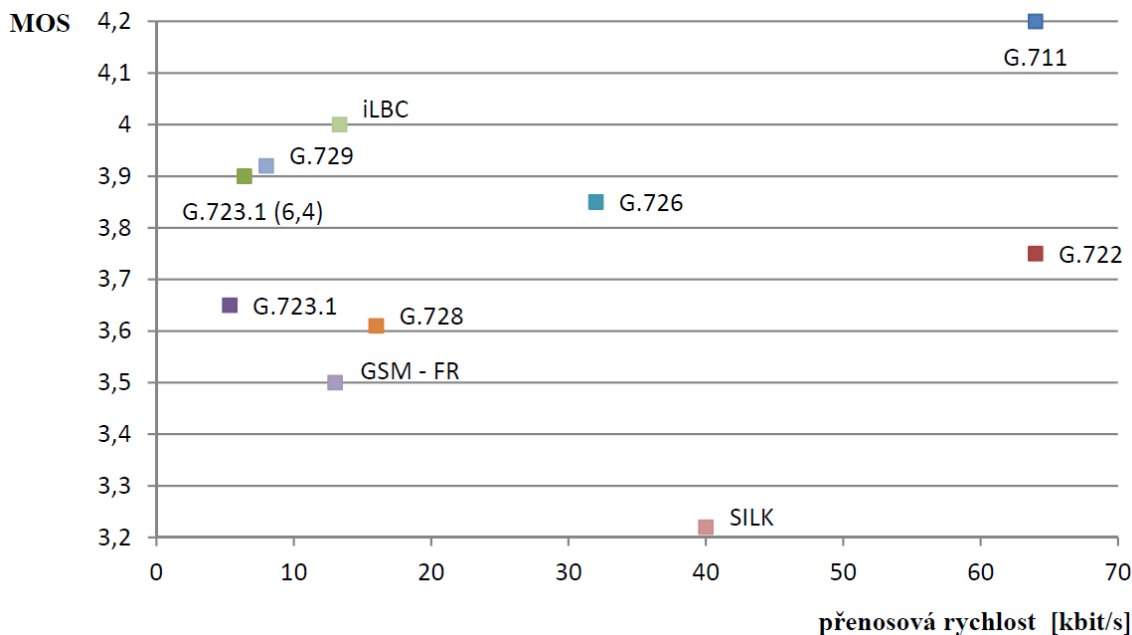
- You will have to be very quiet. / Budete muset být velice tichý.
- There was nothing to be seen. / Nebylo nic vidět.
- They worshipped wooden idols. / Uctívali dřevěné modly.
- Did he need any money? / Potřeboval peníze?
- I want a minute with the inspector. / Chci minutu s inspektorem.

V následující tabulce máme porovnání všech kodeků s vlastnostmi [6].

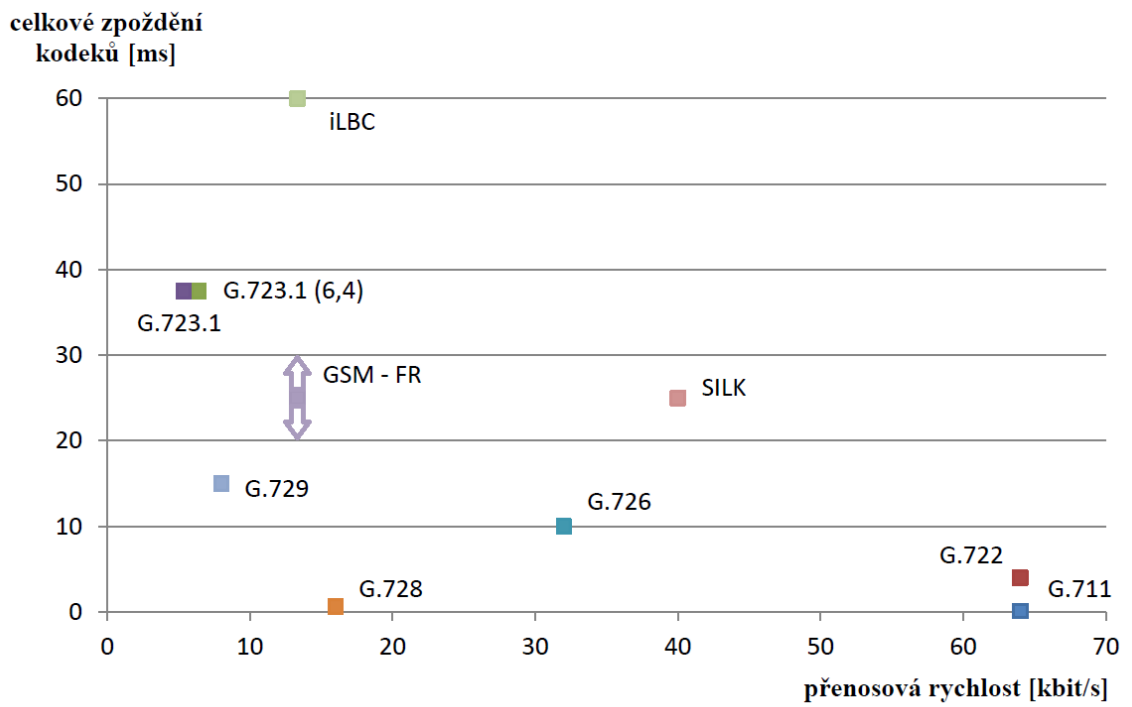
kodek	typ	přenosová rychlost [kbit/s]	MOS	celkové zpoždění kodeků [ms]
G.711	PCM	64	4,2	0,125
G.722	SB-ADPCM	64	3,75	4
G.723.1	ACELP	6,4	3,9	37,5
G.723.1	MP-MLQ	5,33	3,65	37,5
G.726	ADPCM	32	3,85	10
G.728	LD-CELP	16	3,61	0,625
G.729	CS-ACELP	8	3,92	15
GSM - FR	RPE-LTP	13,3	3,5	20-30
SILK	LPT	40	3,22	25
iLBC	LPC	13,33	4	60

Tab. 7: Základní vlastnosti kodeků

Na následujícím obrázku je porovnání MOS s přenosovou rychlostí daných (vybraných) kodeků. A v další je porovnání celkového zpoždění kodeků s přenosovou rychlostí [1].



Obr. 19: MOS / přenosová rychlost kodeků



Obr. 20: Celkové zpoždění / přenosová rychlost

4. Realizace

Tato kapitola je věnovaná samotné realizaci diplomové práce. Nalezneme zde důvody, proč jsem si zvolil toto téma a z jakého důvodu jsem vybral dané zařízení. Dále jsou zde popsány jednotlivé zařízení, které jsou potřeba pro realizaci práce.

4.1. Důvody k výběru

Jeden ze základních důvodů, proč jsem si vybral tuto práci je, že na toto téma jsem už měl svoji bakalářskou práci. Z tohoto důvodu jsem se rozhodl o pokračování v bakalářské práci a její upravení do jiné podoby než jsem poprvé zkoušel.

Dalším z důvodů je seznámení se více s danou technologií VoIP (*Voice over IP*) v našem případě se dá pojednávat spíše o technologii AoIP (*Audio over IP*). Zjistit jaké kodeky se v dnešní době nejvíce používají a jaké mají parametry oproti ostatním.

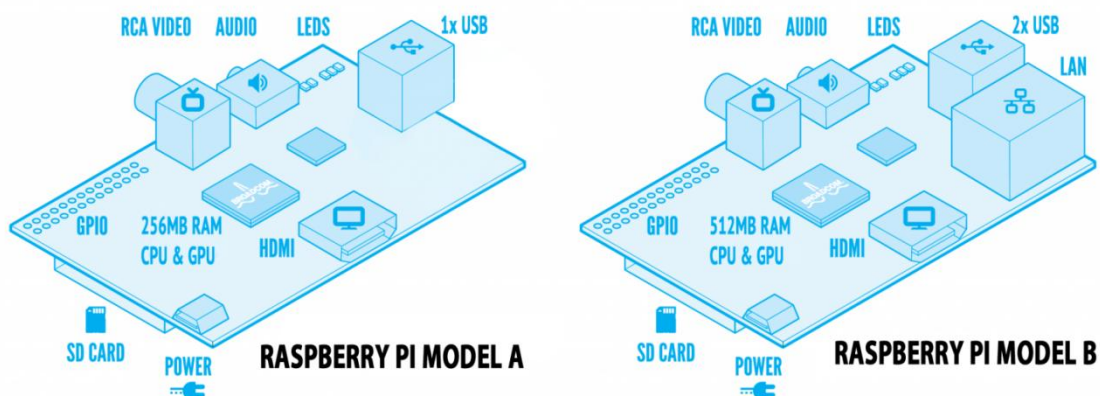
Při výběru produktu jsem se nemusel dlouho rozhodovat, neboť už z určité části mám zkušenosti s Raspberry Pi. Ten se mi osvědčil jako dobrý pomocník při mé bakalářské práci. Výběr byl, jak už bylo řečeno, jasný. Ale jelikož tento rok (2015) vyšla nová verze zmiňovaného Raspberry Pi, tak jsem se rozhodl o jeho koupi. Na dalších stránkách bude porovnání obou zařízení (Raspberry Pi a Raspberry Pi 2).

Dalším zařízením, které jsem potřeboval, bylo sehnat zvukovou kartu. Přemýšlel jsem, jakou kartu bych mohl použít a zda by nestačila pouze integrovaná. Po čase hledání mi byla doporučena USB zvuková karta. Konkrétně to byl typ U-Control UCA202 od výrobce Behringer. Proto jsem ji použil i v mé diplomové práci.

Výhodou daných zařízení je jejich cena a velikost oproti ostatním profesionálním zařízením. Jako je například od firmy 2N produkt 2N® NetSpeaker.

4.1. Raspberry Pi

Raspberry Pi (také někdy značeno jako RPi) je v dnešní době jeden z nejznámějších a nejpoužívanějších minipočítačů na světě. Vyvíjí ho britská nadace Raspberry Pi Foundation. Cílem této nadace bylo seznámit žáky s technikou a informatikou. Kde díky jeho malé velikosti (přibližně velká jako kreditní karta), možnosti vlastní konfigurace a nízké ceně se stal jedním z oblíbených zařízení u uživatelů. RPi má dvě různé varianty a to A a B variantu. Varianta A nemá ethernetový port oproti B a také je rozdíl v rychlosti procesoru. Dále se liší v počtu USB a ve spotřebě energie. V této práci jsem použil verzi B, tedy se dvěma USB a ethernetovým portem.



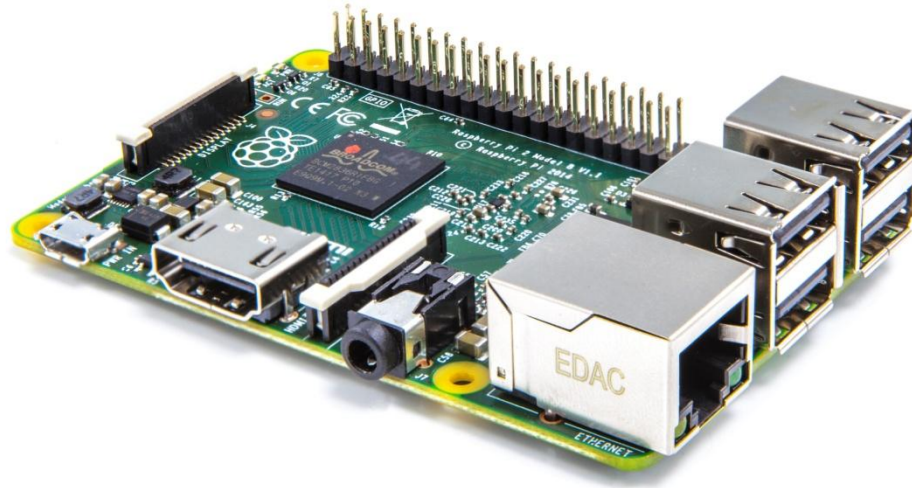
Obr. 21: Porovnání modelů RPi [22]

Z obrázku (Obr. 21) je patrné z jakých konektorů se RPi skládá. Jeho technické parametry nejsou v dnešní době vůbec výjimečné ani dobré. Procesor je zde ARM s taktem 700 MHz, operační paměť 256 MB (512 MB) a velikost diskové paměti je podle SD karty (doporučené jsou min. 4 GB). Tyto technické parametry mají už i nejlevnější mobilní telefony. Ty novější mají několika násobně lepší parametry než RPi.

Výhoda RPi spočívá v jeho univerzálnosti oproti např. mobilním telefonům. Lze ho použít jako mikropočítač nebo jako multimediální stanice u televize (souvisí s programem XBMC).

4.2. Raspberry Pi 2

Raspberry Pi 2 je následovníkem starší verze RPi. Ve vyvinuta stejnou firmou, jak jeho předchůdce. Má pouze vylepšený hardware jinak vše zůstalo „při starém“. Na obrázku níže (Obr. 22) je vzhled nové generace RPi. Zde si můžeme všimnout některých drobných změn. A to je například zdvojnásobení USB portů, odstranění konektoru RCA Video a přemístění některých součástí.



Obr. 22: Vzhled Raspberry Pi 2 [23]

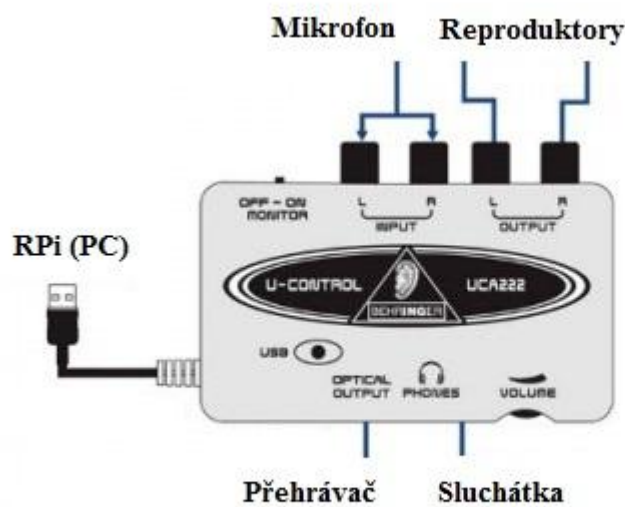
V následující tabulce je zobrazen seznam rozdílů případně neměnných parametrů Raspberry Pi verze jedna a verze dva.

	Raspberry Pi	Raspberry Pi 2
cena	cca 1 000 ,-	cca 1 000 ,-
velikost	85 x 55 mm	85 x 56 mm
hmotnost	42 g	70 g
napájení	přes micro USB	přes micro USB
procesor	700 MHz	900 MHz
počet jader	1x	4x
RAM	512 MB	1 GB
počet USB	2x	4x
uložiště (HDD)	SD karta	micro SD karta (max 128 GB)

Tab. 8: Porovnání verzí Raspberry Pi

4.3. U-Control UCA202

U- Control UCA202 je jednoduchá externí zvuková karta přes USB s mnoha možnostmi nastavení. Toto zařízení vyvinula německá firma Behringer. Zvuková karta se dá použít, jak pro PC, tak i pro MAC a jiné zařízení. Má velmi nízkou latenci. Má v sobě ASIO (Audio Stream Input/Output) driver. Její maximální vzorkovací frekvence je 48 kHz. Stereofonní sluchátkový výstup s vlastním regulátorem hlasitosti. CINCHE pro vstup a výstup a S/PDIF digitální optický výstup. Karta umí také monitorování vstupu.



Obr. 23: Zvuková karta UCA202 a její popis

Jedním z hlavních prvků u této externí zvukové karty je to, že má nízkou spotřebu, což se hodí při připojení k RPi. Dále splňuje Nyquistovu větu a je tedy nutné, aby zvuková karta pracovala, alespoň se vzorkovací frekvencí 40 kHz (lidský sluch 20 Hz - 20 kHz). A posledním důležitým prvkem je, aby nevyžadovala žádné plug-in nebo případně, aby měla co nejmenší nároky na instalaci (existence ovladačů pro linux).

4.4. Instalace a konfigurace zařízení

Instalace a konfigurace RPi je velice jednoduchá (v některých partiích). Neboť RPi slouží jako jednoduchý nástroj pro začínající programátory. V této kapitole si řekneme, jaký operační systém vybrat (je nejlepší) a který byl použit v práci. Dále se budeme zabývat instalací potřebných programů, zařízení a konfigurací.

4.4.1. Operační systém(y)

Jako první věc je důležité si uvědomit hned na začátku, na co se bude používat RPi a k němu pak vyberu potřebný operační systém. V dnešní době je nezměrné množství operačních systémů, ale jen některé se hodí pro takové zařízení jako je RPi. Výběr operačního systému byl velice jednoduchý k dané potřebě. U RPi je doporučený operační systém Raspbian (linuxová verze). Ale můžeme použít i jiné operační systémy, které jsou uvedeny na stránkách Raspberry Pi.

- Raspbian

Je to jedna z linuxových distribucí. Je speciálně vytvořena a testována přímo pro RPi (2). Tento operační systém lze použít také v grafickém prostředí, které se hodně podobá linuxové distribuci Debianu. Pro začátečníky s linuxem (Raspbianem popř. ostatní operační systémy), bych doporučoval přepnutí do grafického prostředí (jednodušší na orientaci). Je to přijatelnější pro běžného uživatele než samotný příkazový řádek.

- Instalace Raspbianu

Přesný popis instalace je uveden na adrese, která se nachází v literatuře [24]. Tento návod je bohužel v angličtině, ale je zde i videonávod, ze kterého to lze pochopit. Zde si vysvětlíme stručnou popis instalace.

1. Stažení operačního systému Raspbian
2. Rozbalení souboru
3. Stažení programu na zápis na SD kartu
4. Zapsání Raspbianu na SD kartu
5. První bootování Raspbianu
 - i. Nastavení jazyka (klávesnice a systému)
 - ii. Určení lokace RPi (nastavení času)
 - iii. Restart
 - iv. Přihlášení do operačního systému
 - v. (Příkaz *startx* (zapnutí grafického prostředí))

- OS - ostatní

Jelikož je to malý počítač, jak už je to v předchozích kapitolách zmíněno, tak i na toto zařízení, lze nahrát jiný operační systém. Např. Raspberry Pi není, tak výkonná jako nová verze, tak bych zde doporučoval jakoukoliv odlehčenou verzi linuxové distribuce. Windows a podobné OS nejsou pro toto zařízení vhodné.

Naopak na Raspberry Pi 2 je speciálně vytvořen Microsoft Windows 10 for Raspberry Pi 2. Zde jen pro informaci uvádím link, odkud je možné stáhnout a být developerem na samotný Windows 10.

<https://www.windowsondevices.com/signup.aspx>

Pro ověření správného nainstalování operačního systému a zároveň zjištění informací ohledně zařízení zjistíme pomocí příkazu „*uname*“.

Raspberry Pi & Raspberry Pi 2

```
pi@raspberrypi ~ $ uname --all
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l GNU/Linux
pi@raspberrypi ~ $ uname --version
uname (GNU coreutils) 8.13
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
```

Obr. 24: Příkaz *uname* na RPi (2)

4.4.2. Mplayer

Mým účelem bylo najít program pro přehrávání audio stop, který by uměl přehrávat jak bezztrátové i ztrátové formáty, tak zároveň pracovat i se streamem. V mém případě bylo důležité najít hlavně bezztrátové formáty. Jelikož je na internetu mnoho programů, co tuto vlastnost mají, tak si stačí vybrat jakýkoliv, který máme rádi. Některé ovšem nemusejí jít nainstalovat. Z tohoto důvodu jsem si vybral do své práce program Mplayer (pro testování). Tento program splňuje všechny moje požadavky a zároveň nezatěžuje příliš procesor RPi. Výběr přehrávače si může zvolit každý z nás. Zde se meze nekladou.

- Instalace Mplayeru¹

Instalace je opravdu jednoduchá. Důležitou podmínkou při instalaci Mplayeru je, že RPi musí být připojen do internetu (pokud nemáme program na USB flash disku). Při instalaci se budou stahovat různé knihovny a všechny potřebné soubory. Dále stačí zadat do terminálu následující příkazy. Pokud nemáme spuštěné grafické prostředí, tak rovnou zadáváme příkazy².

1. Aktualizace knihovny

```
apt-get update
```

2. Samotná instalace Mplayeru

```
apt-get install mplayer
```

3. Tento krok je dobrovolný - instalace Mplayeru do grafického prostředí

```
apt-get install mplayer-gui
```

4. Restart RPi

```
reboot
```

¹ Lze nainstalovat i Mplayer2

² Příkazy je nutné zadávat jako superuživatel (root)! Např. *sudo apt-get install...*

4.4.3. VLC player

Dalším z alternativních přehrávačů a zároveň jeden z nejpoužívanějších přehrávačů na veškeré multimediální soubory, jak na PC, tak i na jiných zařízeních se stal VLC player. Výhoda tohoto přehrávače je, že umí přehrávat různé druhy zvukových formátů (jak už bylo zmíněno) a zároveň je možné přes něj streamovat zvukové soubory. Proto jsem si ve finální verzi práce zvolil tento přehrávač.

- Instalace VLC playeru

Instalace je, tak jak u MPlayeru, velice jednoduchá. Postup je velice podobný, jak u předchozí instalace přehrávače. Pro nejsnadnější instalaci je dobré být připojený k internetu. Pak budeme zadávat tyto následující příkazy do terminálu.³

1. Aktualizace knihovny

```
apt-get update
```

2. Samotní instalace VLC playeru

```
apt-get install vlc
```

3. Restart RPi

```
reboot
```

³ Příkazy je nutné zadávat jako superuživatel

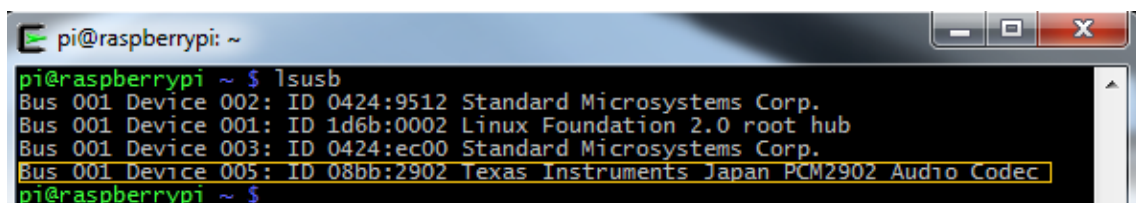
4.4.4. Konfigurace externí zvukové karty UCA202

Jak externí zvukovou kartu UCA202 si vysvětlíme v této podkapitole. Návod je už trochu více komplikovaný, ale nijak složitý. Hned na začátku je důležité mít nejnovější verzi Raspbianu, neboť ze starší verzí je toto zařízení nekompatibilní. Respektive starší verze Raspbianu ignoruje určité příkazy.

- Instalace externí zvukové karty
 1. Připojit USB zvukovou kartu
 2. Zjištění, zda je zvuková karta zapojena

lsusb

Výsledkem musí být seznam všech připojených zařízení do RPi (viz obrázek níže). Na obrázku je pak vyznačena naše připojená USB zvuková karta. V našem případě je to zvuková karta s názvem Texas Instruments Japan PCM2902 Audio Codec.

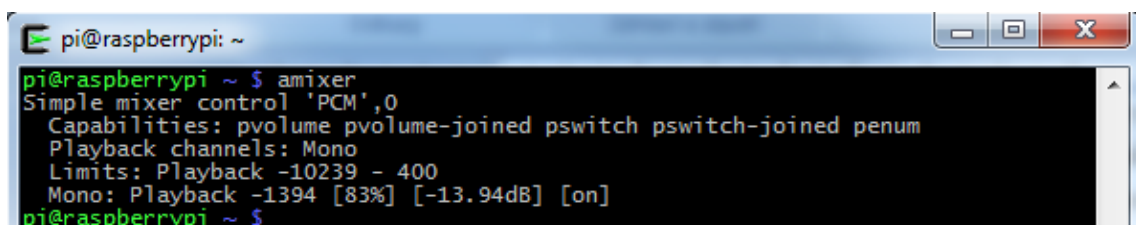


```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ lsusb  
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 005: ID 08bb:2902 Texas Instruments Japan PCM2902 Audio Codec  
pi@raspberrypi ~ $
```

Obr. 25: Příkaz *lsusb* (výpis USB zařízení)

3. Zobrazení aktuálního nastavení zvukové karty a její stručné informace

amixer



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ amixer  
Simple mixer control 'PCM',0  
Capabilities: pvolume pvolume-joined pswitch pswitch-joined penum  
Playback channels: Mono  
Limits: Playback -10239 - 400  
Mono: Playback -1394 [83%] [-13.94dB] [on]  
pi@raspberrypi ~ $
```

Obr. 26: Příkaz *amixer*

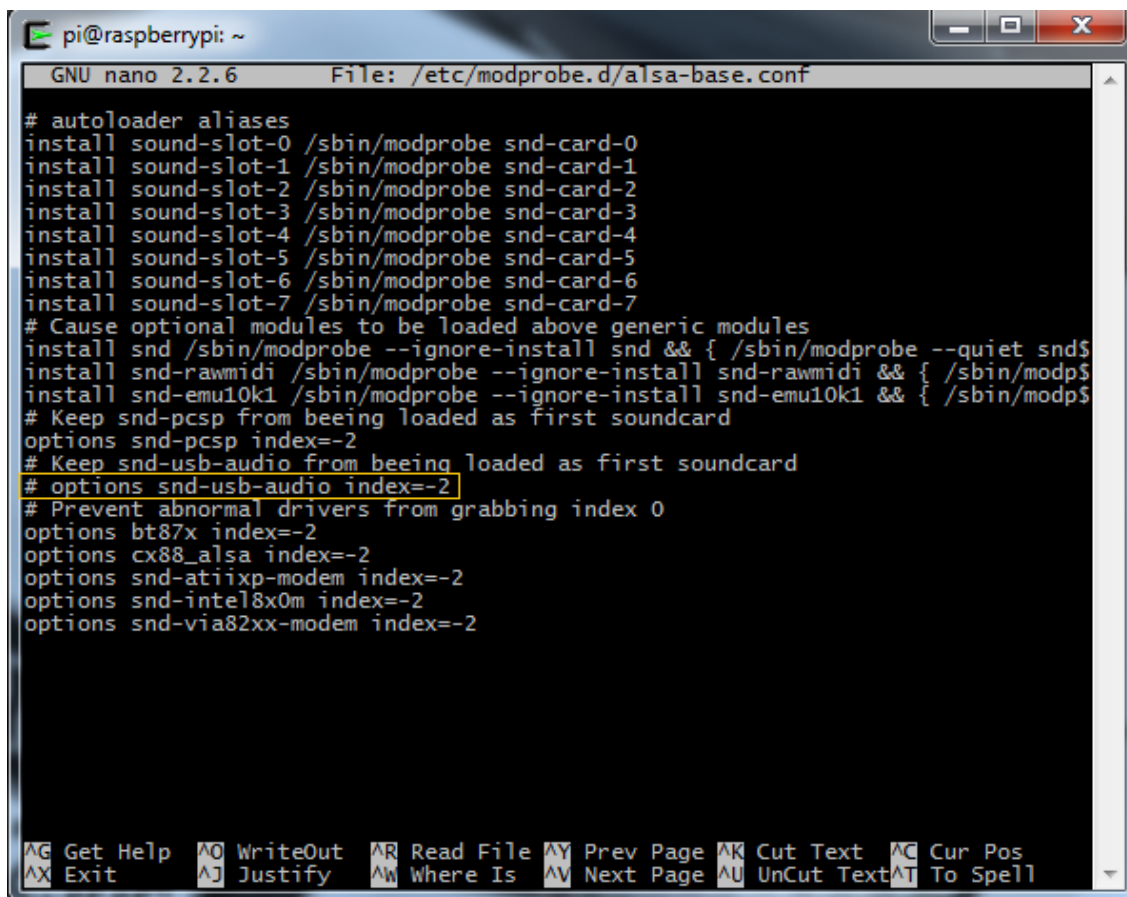
Zde na obrázku (Obr. 26) je zobrazen příkaz *amixer*. Toto je původní verze ještě bez jakéhokoliv nastavení zvukové karty. Na straně 41 je ukázka příkazu *amixer*, jak má vypadat.

4. Samotná úprava konfiguračního souboru

Proto, abychom měli nastavenou zvukovou kartu stále, je zapotřebí, abychom přepsali konfigurační soubor. Důležitým aspektem je, že po vypojení naší externí karty se zařízení vrátí do původního nastavení. Po znovu připojení zvukové karty je důležité opět restartovat RPi, aby se nám karta zaktivovala.

Konfigurační soubor najdeme pod názvem `alsa-base.conf`. Pomocí jakéhokoliv textového editoru ho otevřeme a následně upravíme (v mém případě je použit textový editor `nano`). Ten konfigurační soubor najdeme pomocí příkazu⁴:

```
nano /etc/modprobe.d/alsa-base.conf
```



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/modprobe.d/alsa-base.conf
# autoloader aliases
install sound-slot-0 /sbin/modprobe snd-card-0
install sound-slot-1 /sbin/modprobe snd-card-1
install sound-slot-2 /sbin/modprobe snd-card-2
install sound-slot-3 /sbin/modprobe snd-card-3
install sound-slot-4 /sbin/modprobe snd-card-4
install sound-slot-5 /sbin/modprobe snd-card-5
install sound-slot-6 /sbin/modprobe snd-card-6
install sound-slot-7 /sbin/modprobe snd-card-7
# Cause optional modules to be loaded above generic modules
install snd /sbin/modprobe --ignore-install snd && { /sbin/modprobe --quiet snd$
install snd-rawmidi /sbin/modprobe --ignore-install snd-rawmidi && { /sbin/modp$
install snd-emu10k1 /sbin/modprobe --ignore-install snd-emu10k1 && { /sbin/modp$
# Keep snd-pcsp from beeing loaded as first soundcard
options snd-pcsp index=-2
# Keep snd-usb-audio from beeing loaded as first soundcard
# options snd-usb-audio index=-2
# Prevent abnormal drivers from grabbing index 0
options bt87x index=-2
options cx88_alsa index=-2
options snd-atiixp-modem index=-2
options snd-intel8x0m index=-2
options snd-via82xx-modem index=-2

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Obr. 27: Výpis konfiguračního souboru `alsa.base.conf`

⁴ Příkazy je nutné zadávat jako superuživatel (`sudopříkaz...`)

Na obrázku je řádek, který budeme upravovat. Zde máme dvě možnosti:

- a) Řádek s názvem „option snd-usb-audio index=-2“ zakomentovat (na to použijeme znak #)

Řádek pak bude vypadat takto: `# option snd-usb-audio index=-2`

- b) Řádek s názvem „option snd-usb-audio index=-2“ přepsat

Řádek pak bude vypadat takto: `option snd-usb-audio index=0`

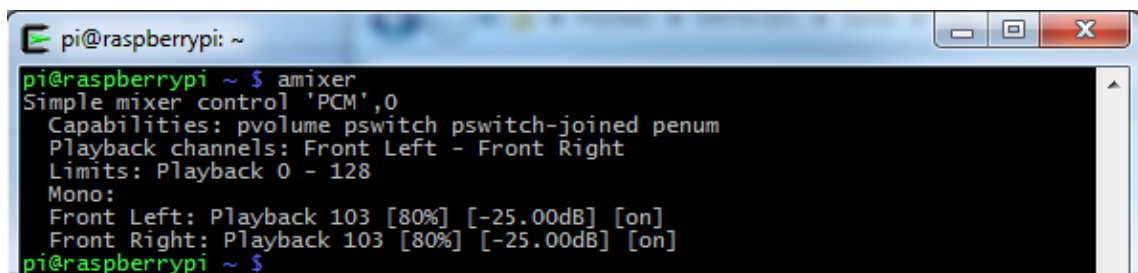
V tomto případě jsem zvolil variantu za a). Pak stačí konfigurační soubor uložit.

5. Restart RPi

`reboot`

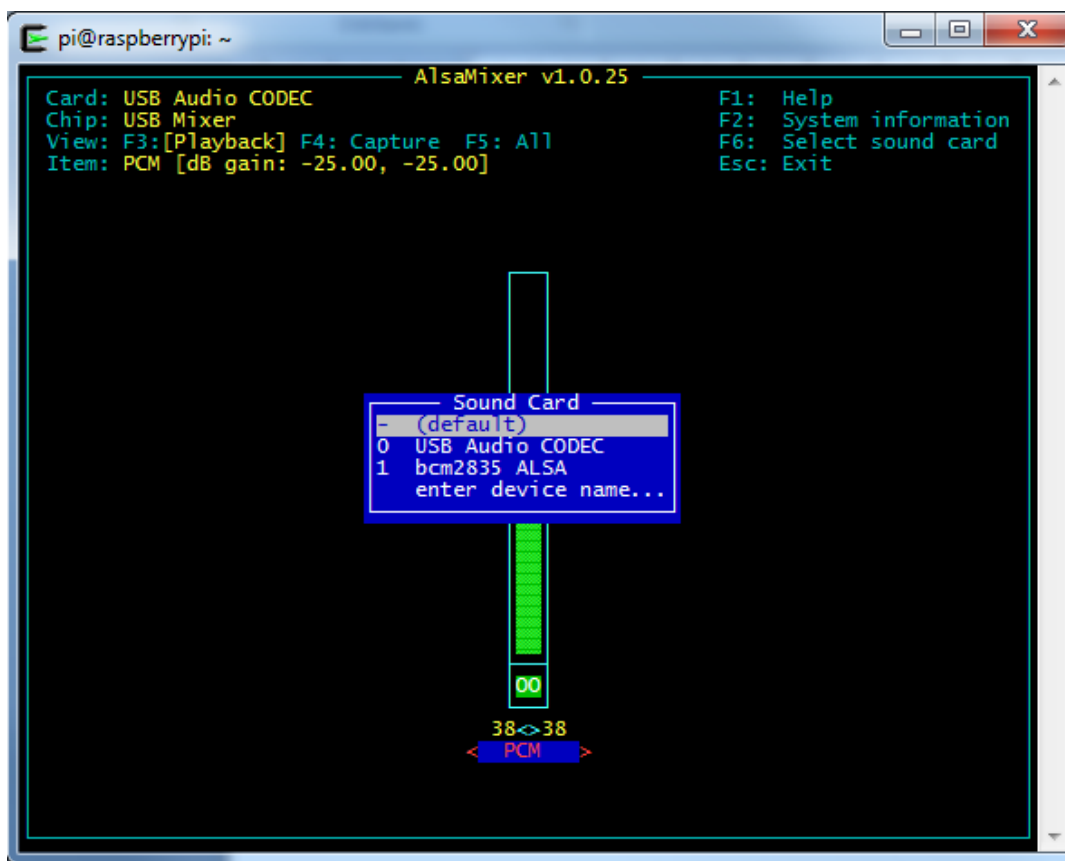
6. Ověření změny nastavení (amixer, alsamixer)

`amixer`



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ amixer
Simple mixer control 'PCM',0
Capabilities: pvolume pswitch pswitch-joined penum
Playback channels: Front Left - Front Right
Limits: Playback 0 - 128
Mono:
Front Left: Playback 103 [80%] [-25.00dB] [on]
Front Right: Playback 103 [80%] [-25.00dB] [on]
pi@raspberrypi ~ $
```

Obr. 28: Ověření změny v amixer

alsamixer

Obr. 29: Ověření změny v alsamixeru

Pomocí příkazu `alsamixer`, lze zjistit více informací o audio v RPi. Na obrázku je znázorněn výpis po změně na externí zvukovou kartu. Pomocí tlačítka F6, lze zobrazit výpis všech zvukových karet. Kde pod názvem `bcm2835 ALSA` je ukrytý výstup jack 3,5 mm a HDMI.

7. Otestování funkčnosti

Zde stačí spuštění jakéhokoliv audio souboru. Případě spuštění audio streamu.

4.4.5. Instalace potřebných programů - node

V první řadě je důležité vymyslet, jaký program použít na zobrazení webového serveru (pro tuto variantu jsem se rozhodl) a co bude dobré pro jeho komunikaci a spouštění přehrávače. Proto jsem se rozhodl použít program `nodejs`. Tento program jsem nainstaloval pomocí příkazu `sudo apt-get install nodejs`. Bohužel po dlouhém zkoušení jsem byl nucen najít jiný program. Nodejs neuměl nainstalovat balíčkovací systém NPM, který potřebujeme na GPIO. NPM je podobný balíčkovací systém jako nejznámější `apt-get`. Nyní si ukážeme, jaké příkazy použít pro nainstalování programu `node` (novější verze, než je možné samotným `apt-get install`). Starší verze nepodporovala naši knihovnu pro GPIO tak, jak samotný program `nodejs`.

- Instalace node

Jak už je zde zmíněno, tak bylo nutné nainstalovat novou verzi programu `node`. Nelze to udělat standardním způsobem pomocí `apt-get`, ale musíme si ho zkompileovat sami.

1. Stáhnout soubor

```
wget http://nodejs.org/dist/v0.10.2/node-v0.10.2.tar.gz
```

2. Rozbalit

```
tar -xzf node-v0.10.2.tar.gz
```

3. Ve složce `node-v0.10.2`⁵

```
./config
```

```
Make
```

```
Sudo make install
```

Pokud nechcete čekat, je zde druhá možnost (lepší).

1. Stáhnout soubor

```
curl -sLS https://apt.adafruit.com/add | sudo bash
```

2. Nainstalovat

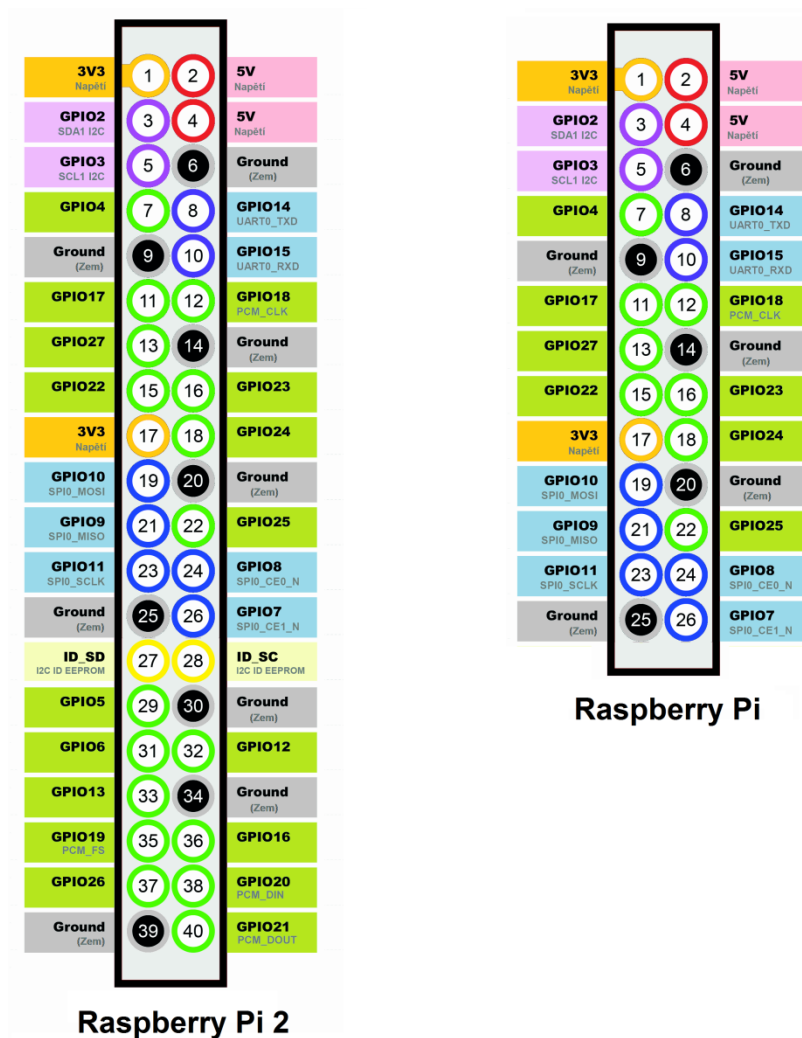
```
sudo apt-get install node
```

⁵ Tento bod může trvat až několik hodin (testováno na RPi)

4.4.6. GPIO

GPIO (general-purpose input/output) slouží pro ovládání vstupních a výstupních vlastností zařízení. Jsou to takzvané piny, které nemají zvlášť definovaný účel. Lze si je tedy nastavit dle vlastních požadavků. Umožňují například nastavení pinu jako vstup nebo výstup. Dále pak je lze vypnout/zapnout, mohou nabývat hodnot 1 (true) nebo 0 (false), může se na ně zapisovat či číst a mnoho dalšího. Na GPIO piny lze také připojit i jiná zařízení která komunikují přes I2C, SPI nebo UART (RS232). Hodí se tedy pro různé displeje, čidla, převodníky, teploměry atd.

U Raspberry Pi se nachází těchto pinů celkem 26 viz Obr. 30. Kde jen s některými lze pracovat. Jsou to piny, které jsou popsány GPIO(číslo). Kdež to u Raspberry Pi 2 je těchto pinů o něco více. Zde jich je 40. Barevné označení slouží pro lepší orientaci např. tmavě modré kolečka dole jsou používány pro SPI či fialové kolečka pro I2C atd.



Obr. 30: Piny na RPi a RPi 2

Důležitou poznámkou, kterou opravdu musíme dodržovat je maximální napětí, které nesmí přesáhnout hranici 3,3 V. Pokud přivedeme více napětí s největší pravděpodobností zničíme RPi. Pozor taktéž u číslování pinů. Fyzické piny neodpovídají GPIO signálů v procesoru.

Pro ovládání GPIO je nutné použít některý s programovacích jazyků (Python, C+, javascript, atd.). v mém případě jsem si zvolil program Node (javascript) kde je použit tzv. npm balíčkovací systém, který právě využívá GPIO piny. Programů v balíčkovacím systému npm je hodně. V této práci jsem si zvolil npn onoff (viz návod níže). Ale lze použít i rpi-gpio nebo pi-gpio a mnohé další.

1. Instalace npm (je možné, že se nainstaluje sám s novým Node)

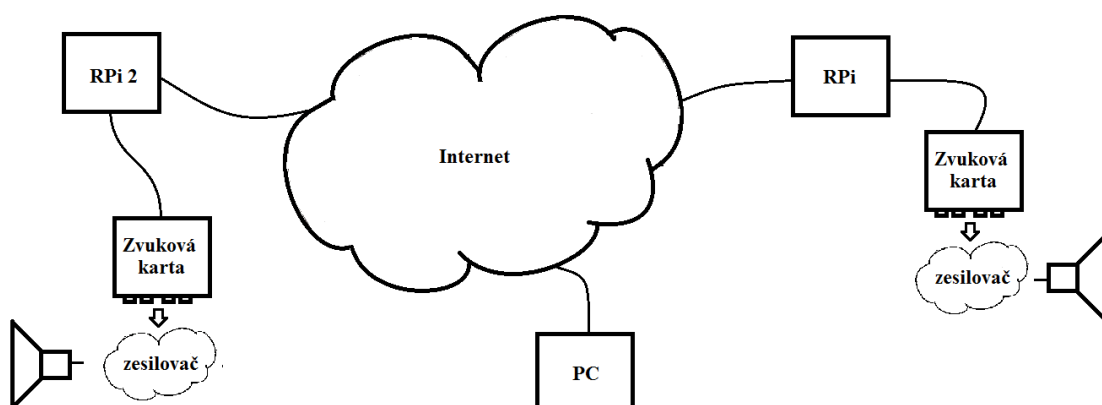
```
Sudo apt-get install npm
```

2. Instalace OnOff

```
Npm install onoff
```

4.4.7. Ukázkové zapojení

Pro sestavené mé práce jsem si zvolil dva druhy RPi (více na str. 32 a 33). Na Obr. 31 je vyobrazeno moje ukázkové zapojení sítě s použitím všech zařízení, které jsem měl k dispozici. V přílohách na str. 61 lze najít vyfocené obrázky s fyzickými zařízeními. Nyní si ukážeme schéma zapojené mé práce.



Obr. 31: Schéma ukázkového zapojení

Na obrázku jsou vyobrazeny všechny zařízení, tak jak jsem je měl zapojeny. PC slouží pouze jako konfigurační zařízení, přes které se můžeme vzdáleně napojit na naše minipočítače. Z obrázku je patrné zapojení všech zařízení do jedné sítě. Dále lze vidět propojení RPi (2) s externí USB zvukovou kartou, kterou jsem pro testování měl zapojenou do repro soustavy. Zkoušel jsem zapojení, jak s pomocí jacku 3,5mm, tak s pomocí cinche (Cinche byly mnohem lepší pro pocit z hudby, než samotný jack). Níže je vyobrazeno zapojení s použitím GPIO pinů.

Pro streamování hudby jsem použil již zmiňovaný VLC player, který byl nainstalovaný spolu s dalšími programy, které jsou zde vypsány. VLC player byl použit na obou koncích ukázkového zapojení. Při použití pro streamování VLC playeru (vysílací část) a Mplayer (přijímací část), docházelo k velkému zpoždění.

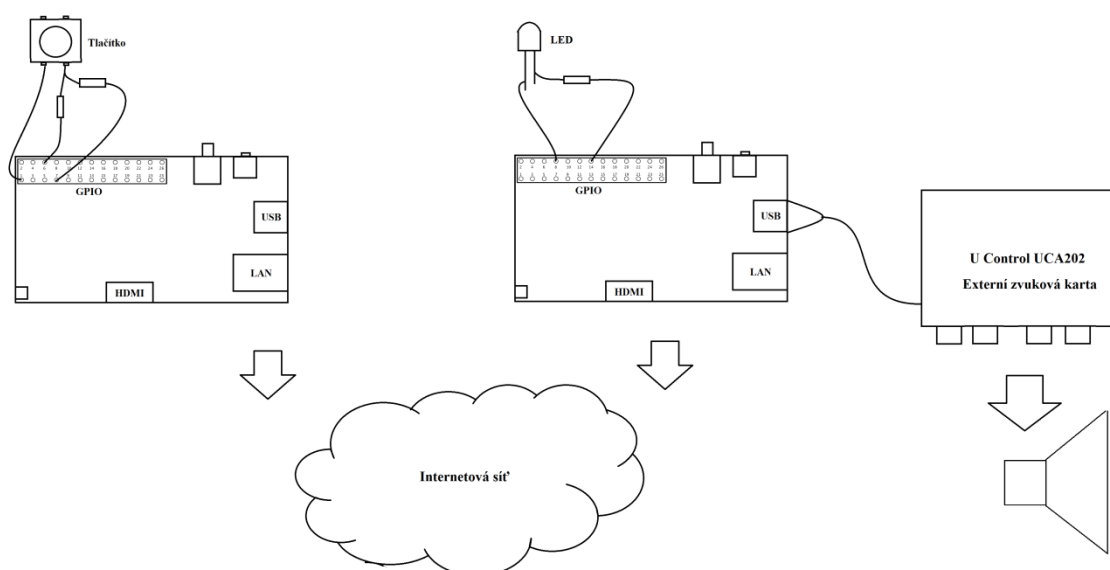
Ve vysílací části stačilo vždy zadat pouze jeden delší příkaz pro zapnutí streamu. Viz ukázka níže.

```
colc audio.mp3 -sout
'#transcode{acodec=kodek,channels=2,ab=128,
samplerate=44100};std{access=protokol,mux=ts,dst=port}'
```

V přijímací části pak např.

```
colc http://IP_adresa_vysilace:port
```

Na obrázku je schéma zapojení RPi s využitím GPIO pinů.



Obr. 32: Schéma zapojení RPi pomocí GPIO pinů

4.5. Ovládání a samotný skript

Ovládání je vytvořeno s co nejjednodušším užíváním. Tak, aby to bylo tzv. user friendly, česky přívětivé a jednoduché. Proto jsem zvolil nastavování přes webový server. Neboť spousta běžných uživatelů mají doma obyčejný modem (router), který umí nastavit. Proto jsem se snažil o stejnou koncepci.

4.5.1. Ovládání

Ovládání je vytvořeno pomocí programu node, který slouží jako webový server. Kde na každém RPi je tento server spuštěn. Node také slouží pro veškerou komunikaci mezi zařízeními. Na tento program je napojena celá funkčnost této diplomové práce.

Bohužel v mé práci nefunguje cron tak, jak bych si představoval. Proto je nutné při prvním spuštění RPi se nalogovat a zadat jednoduchý příkaz. Tento příkaz je uvedený níže.

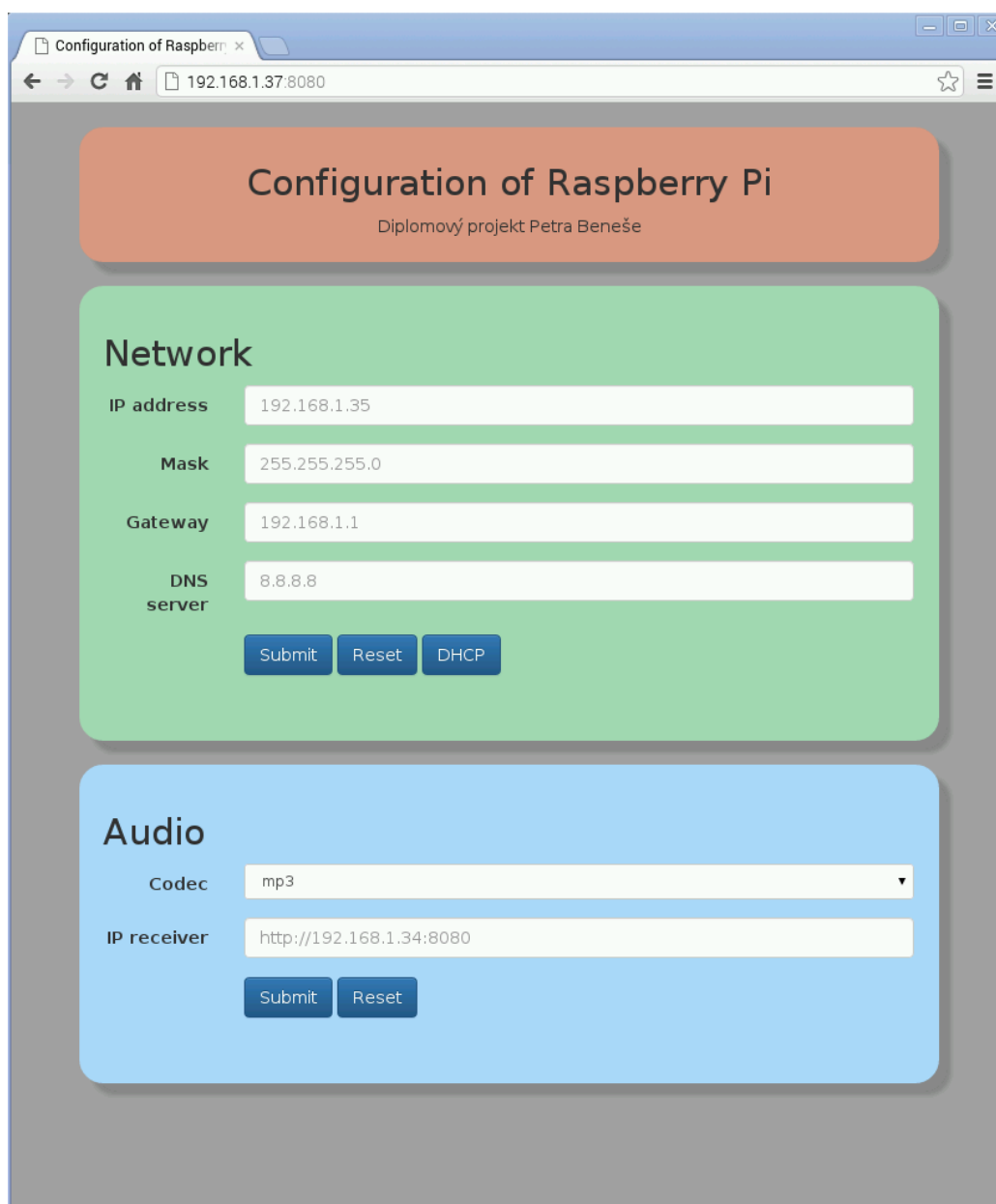
Node /rpiaudio/server.js

Po sléze se v konzolovém řádku vypíše, že je server aktivní. Poté stačí z jakéhokoli počítače v síti. Případně pokud má RPi nastavenou veřejnou IP adresu, tak odkudkoliv se na něho připojit. Výjimka nastává pouze při úplně novém spuštění RPi, zde je pochopitelně nutné nainstalovat patřičné programy, nahrát soubory a zjistit si svoji IP adresu.

Program funguje tak, že pokud uživatel zná svoji IP adresu (IP adresu RPi) a má spuštěný node. Tak mu stačí pouze do prohlížeče napsat IP adresu RPi a port, který je v programu nastaven na číslo 8080. Čili samotný příkaz bude vypadat viz níže, případně viz Obr. 33.

http://IP_Adresa_RPi:port

http://192.168.1.37:8080



Obr. 33: Vzhled webového serveru na RPi

V zeleném rámečku si můžeme všimnout samotného nastavení síťové části. Kde je možnost nastavit si vlastní parametry jako jsou např. IP adresa, maska, brána a DNS server (pro překlad adres).

Jsou zde tři tlačítka sloužící pro uložení statické adresy (tlačítko „Submit“). Dále je zde „Reset“ sloužící pro smazání aktuálního vyplnění formuláře a „DHCP“. Toto tlačítko je zde z důvodu toho, že pokud uživatel zadá statickou adresu a bude například chtít přemístit zařízení, kde mají spuštěný DHCP server, tak nebude muset jít do samotného nastavení zařízení, ale bude moci pouze stlačit tlačítko a skript to vykoná za něho.

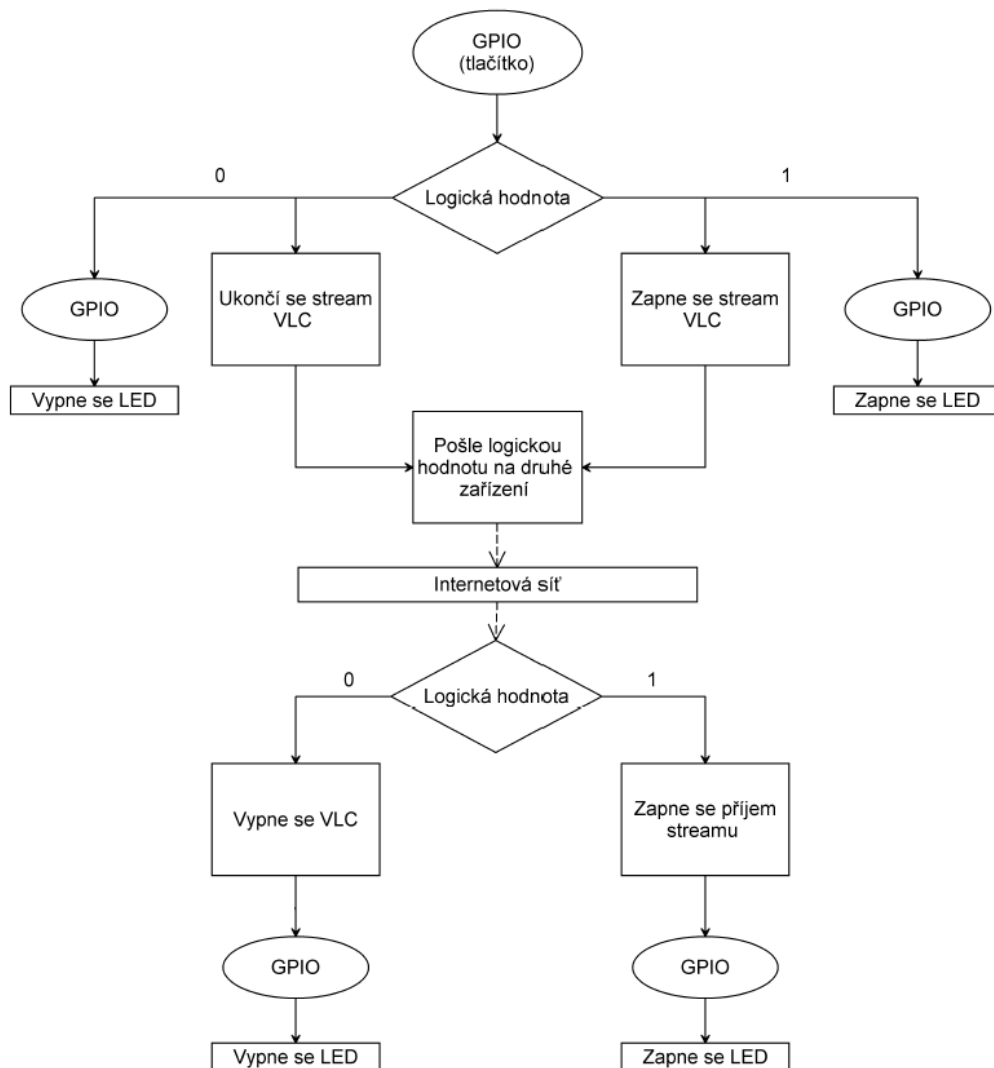
Modrý rámeček slouží pro vybrání kodeku, přes který chceme přenášet naše audio. Zde máme několik možností, který kodek si vybereme. Pochopitelně lze jednoduše upravovat tyto kodeky a případně přidávat či ubírat. Zde se meze nekladou. Pouze je důležité, aby byly tyto kodeky kompatibilní s VLC playerem, který v práci používáme.

Dále se v modrém rámečku nachází kolonka IP receiver, což je IP adresa přijímače. Tady se jen podle předlohy vyplní IP adresa a port a vše se opět potvrdí tlačítkem „Submit“. „Reset“ opět vymaže aktuální záznam z prohlížeče.

Zapínání a vypínání přenosu je napsáno v kapitole níže.

4.5.2. Funkčnost

V této kapitole vysvětlím, alespoň stručně, jak vlastně práce funguje a jak se zapíná/vypíná. Proto jsem si vytvořil stručný popis celé architektury práce.

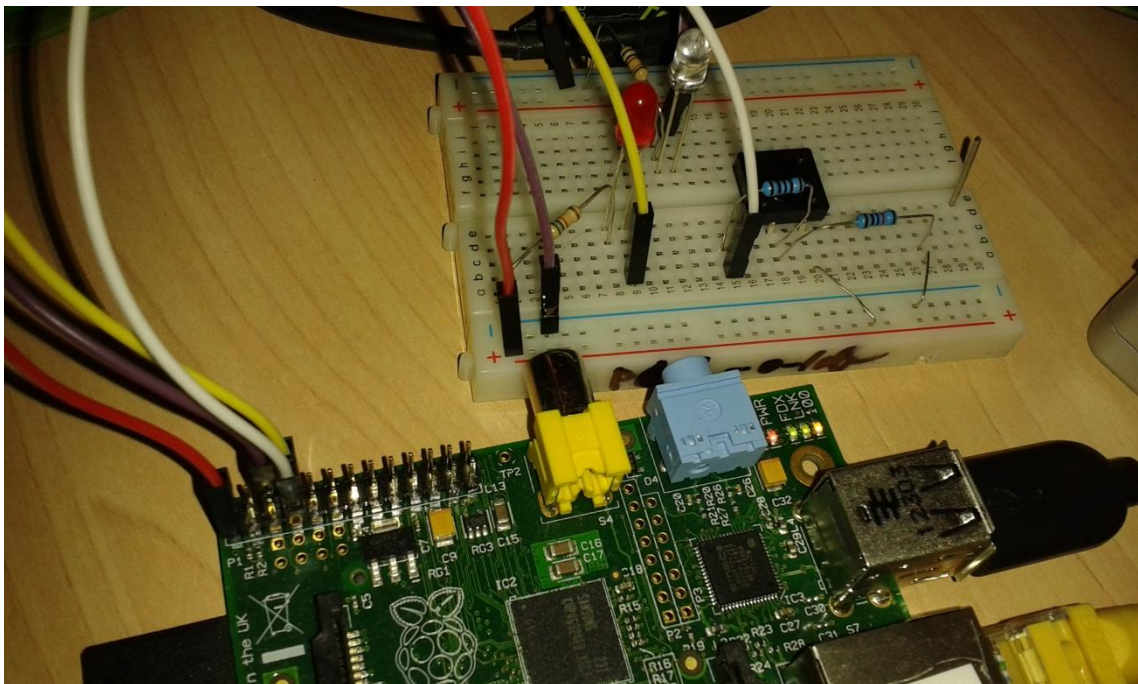


Obr. 34: Architektura diplomové práce

Z Obr. 34 lze jednoduše zjistit, jak funguje celá práce. Hlavním bodem je samotné tlačítko, které mi nahrazuje logický vstup a LED diody, které mi naopak nahrazují logický výstup. Pokud na obou zařízeních RPi běží náš skript (je puštěn node a vše bude správně nastaveno), tak po zmáčknutí tlačítka se z logické nuly stane jednička a podle grafu se vykoná daná činnost. V našem případě se zjistí, zda je na vstupu logická jedna nebo nula, když to bude jednička. Tak skript zapne streamování pomocí VLC playeru a zároveň se rozsvítí LED dioda, která je zapojena na jeden námi definovaným pinem. Při nule nastává opačný jev a to, že se vypne stream a zhasne LED dioda.

Skript potom pošle logickou hodnotu na druhé zařízení, kde se zjistí, co za hodnotu přišlo a podle toho bude pracovat. Opět je zde rozhodovací člen, který nám buď zhasne nebo rozsvítí LED diodu.

Na obrázku níže je vyfoceno zapojení v praxi.



Obr. 35: Zapojení v praxi

4.5.3. Samotný skript

V následující podkapitole si ukážeme různé výřezy mého kódu. Pro práci jsem využil programovací jazyk JavaScript. A programy k němu viz kapitola Instalace a konfigurace zařízení 4.4. Zde si ukážeme jen podstatné části celého projektu. Na začátek si ukážeme, jak jsou vyřešeny GPIO piny v práci. Dále se podíváme na samotný kód programu, jako je vysílání a příjem streamu. Nastavení síťové části.

Kousek kódu pro GPIO ovládání.

```
/***/ GPIO
try {
  var Gpio = require('onoff').Gpio,
      led = new Gpio(14, 'out'),
      button = new Gpio(4, 'in', 'both');

  button.watch(function(err, value) {
    led.writeSync(value);
    console.log("tlacitko zmacknuto ! hodnota: " + value );
    if (value){
      console.log("STREAMING STARTED");
      runCommand(codec);
      console.log("EMITTER: Vyslan signal na:"+recieverUrl+"/listen")
      send(recieverUrl, "/listen")
    } else {
      console.log("ukonci se");
      send(recieverUrl, "/stop")
      runCommand("killall vlc");
    }
  });
}
```

Vybrání požadovaného audio kodeku. Kde přenos je zajištěn v mém případě pomocí protokolu HTTP. Lze změnit na např. RTP atd.

```
function audio(json){
  recieverUrl= json.ip_address_reciever;
  emitterUrl = json.ip_address_emitter;

  var codecs ={
    "mp3" : "cvlc audio.mp3 --sout '#transcode{acodec=mp3,...}:std{access=http,...}'",
    "aac" : "cvlc audio.mp3 --sout '#transcode{acodec=mp3,...}:std{access=http,...}'",
    "mp4a" : "cvlc audio.mp4 --sout '#transcode{acodec=mp4a,...}:std{access=http,...}'",
    "flac" : "cvlc audio.flac --sout '#transcode{acodec=flac,...}:std{access=http,...}'",
    "wma" : "cvlc audio.wma --sout '#transcode{acodec=wma,...}:std{access=http,...}'",
  }

  console.log("CODEC SELECTED");
  codec = codecs[json.codec]

  sendResponse(res, "EMITTER: vybran kodek")
}
```

Nastavení síťové části pro statické a pro DHCP. ⁶

```
function network(json){
    var interfaces = "auto eth0\n" +
        "\niface lo inet loopback" +
        "\niface eth0 inet static" +
        "\naddress "+ json.ip_address +
        "\nnetmask " + json.mask +
        "\ngateway " + json.gateway+
        "\ndns-server "+ json.DNSserver+
        "\n\nwpa-roam /etc/wpa_supplicant/wpa_supplicant.conf";

    app.get('/dhcp', function (req, res) {

        var interfaces ="auto lo\n" +
            "\niface lo inet loopback" +
            "\niface eth0 inet dhcp"+
            "\n\nallow-hotplug wlan0"+
            "\niface wlan0 inet manual"+
            "\nwpa-roam /etc/wpa_supplicant/wpa_supplicant.conf"+
            "\niface default inet dhcp";
```

4.5.4. Zlepšení celého systému

Ve své diplomové práci jsem se zabýval hlavně přenosem přes http protokol, což v streamové sféře není zrovna ideální. Diplomová práce by tedy šla zlepšit v použití jiného protokolu, jako je například protokol RTP či jiné. Dále by bylo dobré v praktickém zapojení použít nejnovější verzi RPi. Je to hlavně z důvodu rychlosti celého systému.

Jako zvukovou kartu by bylo lepší použít takovou kartu, která by byla přímo napojena na RPi, čímž je myšleno zapojení na GPIO piny. V aktuálním stavu není zrovna ideální mít externí zvukovou kartu přes USB. Velikostí je to nepraktické.

Jako další vylepšení bych viděl zapnutí programu node hned na začátku při spuštění zařízení. Zde je možnost z několika variant a to např. crontab, /etc/rc.local nebo /etc/init.d. Automatické zapnutí jsem zkusil pouze přes crontab, což není zrovna nejlepší volba. Neboť se mi přes něj nepodařilo zapnout stream. Z neurčitého důvodu se program node zapne, ale stream už ne. Pokud by se mělo toto zařízení nasazovat do provozu, bylo by dobré tento problém vyřešit. Jako daleko lepší volba je např. použití /etc/init.d.

⁶ Pro to, abychom nemuseli psát stále příkaz sudo při změně statické adresy. Stačí napsat jednoduchý příkaz a to *sudo chmod a=rwx /cesta/k/souboru/nebo/samotný/soubor*

5. Vyhodnocení

Tato práce slouží jako teoretický základ na zkonstruování praktického výrobku. Je zde pojednáno o základech IP sítě. Jako jsou referenční modely, několik základních protokolů, co se používají pro přenos jakéhokoliv audio/video. Jsou zde vypsány jedny z hlavních kodeků (hlasové a zvukové - rozhlasové), taktéž druhy kódování. Některé kodeky jsem převzal ze svojí bakalářské práce a zahrnul je do diplomové práce. Slouží pouze jako teoretický základ.

V kapitole 4 je popsána samotná realizace celého projektu. Na začátku jsou probrány důvody výběru mé práce. A dále popsány stručně jednotlivé zařízení, se kterými pracuji. Na straně 33 je porovnání těchto dvou verzí RPi (Raspberry Pi). Je zde ještě stručně pojednáno o externí zvukové kartě UCA202 od firmy Behringer, která je zapojena pomocí USB.

V podkapitole 4.4 je zahrnuta veškerá instalace a konfigurace zařízení. Její postup, jak je správně nainstalovat a co je případně potřeba k jejich instalaci. V práci jsem se zabýval zvukem. Hledal jsem, který přehrávač je ideální pro mou práci. Našel jsem MPlayer a VLC player. Z praxe jsem usoudil, že lepší bude VLC player, neboť kombinace obou nebyla příliš dobrá vůči zpoždění přenosu.

Pro správné přehrávání zvuku se stará integrovaná zvuková karta, která má dva výstupy a to jack 3,5 mm a HDMI. Jelikož jsem potřeboval lepší zvukový převodník, tak jsem zvolil UCA202 (byla mi doporučena). Je zde tedy popsán postup, jak ji správně nainstalovat a čeho se vyvarovat při instalaci.

V samotné práci je potřeba pracovat s GPIO piny. O tyto piny se nám stará program node, ve kterém je v kapitole 4.4.5 vše vysvětleno, jak naprogramovat piny ke správné funkčnosti. Program node jsem použil i z důvodu kompatibility webového serveru, ovládání GPIO pinu pomocí onoff balíčku atd.

Je zde také kapitola ohledně zapojení zařízení a jeho ovládání přes webové rozhraní. V práci jsou také zahrnuty kusy kódů, které jsou potřeba ke správné funkčnosti. A v poslední řadě je zde možnost vylepšení stávajícího projektu.

V práci jsem si otestoval přenos zvuku pomocí IP sítě. Zde jsem si potvrdil, že pakety se streamem chodí správně. Dále jsem se seznámil s GPIO piny, které jsem nastavil dle požadavků. Práce funguje dle předpokladů tak, že při vstupu logické jedna na určitý pin se spustí stream, kde na druhé straně se tento stream začne zachytávat a vyhodnotí se logická jedna nebo nula. Kde na druhém zařízení na jeho výstupním pinu začne vysílat také logickou jedna nebo nula.

V praktické části jsem simuloval logický vstup na GPIO pinu vysílače pomocí tlačítka a na přijímači výstupním GPIO pinu pomocí LED diody. Vše pracovalo, tak jak bylo nastaveno.

6. Literatura

[1] BENEŠ, P. *Přenos audio signálu prostřednictvím protokolu IP*. Praha: ČVUT – ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, 2013. s. 58.

[2] VOZŇÁK, M. *Voice over IP* 1. vyd. Ostrava: VŠB – TECHNICKÁ UNIVERZITA OSTRAVA, 2009. s. 176

[3] VLK, B. *možnosti videokonference v pbx asterisk* 1. vyd. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2009. s. 55.

[4] ŠKROPIL, Vladislav; PROCHÁZKA, Tomáš. *Elektrorevue.cz* [online]. 19.12.2002 [cit. 2011-12-08]. Komprese dat u ISDN. Dostupné z WWW: <<http://www.elektrorevue.cz/clanky/02068/index.html#kap4>>.

[5] ŠIMÁK, Boris. *Principy zpracování hlasu v klasické a IP telefonii* [online]. Katedra telekomunikační techniky (FEL ČVUT), 2011 [cit. 2011-12-09]. Itpoint.cz. Dostupné z WWW: <<http://www.itpoint.cz/ip-telefonie/teorie/princip-zpracovani-hlasu-ip-telefonie.asp>>.

[6] *Voip Think* [online]. 2011 [cit. 2011-12-09]. Codec Summary table. Dostupné z WWW: <<http://www.en.voipforo.com/codec/codecs.php>>.

[7] *CouthIT* [online]. 2011 [cit. 2011-12-09]. SPEECH CODECS. Dostupné z WWW: <<http://www.couthit.com/speech-codecs.asp>>.

[8] Cisco: IPv6 Extension Headers Review and Considerations. Cisco [online]. 2006 [cit. 2014-12-28]. Dostupné z: http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html

[9] HUJKA, Petr. *Elektrorevue* [online]. 2003 [cit. 2014-12-30]. Dostupné z: <http://www.elektrorevue.cz/clanky/03018/index.html>

[10] *Wikipedia* [online]. 2013 [cit. 2013-02-23]. Dostupné z: http://cs.wikipedia.org/wiki/Pulzn%C4%9B_k%C3%B3dov%C3%A1_modulace

- [11] Morpheustechology. *3.0 Analog & Digital Conversions* [online]. 2011 [cit. 2013-02-23]. Dostupné z: <http://www.morpheustechology.com/ebooks/Digital/03%20A&D%20Conversions.htm>
- [12] Voip Think. *G711.1 - G729.1 - G722.1 - G722.1* [online]. 2012 [cit. 2013-02-23]. Dostupné z: <http://www.en.voipforo.com/codec/new-codecs-g7111-g7291.php>
- [13] VoipSupply.com. *Cisco HD Voice* [online]. 2013 [cit. 2013-02-23]. Dostupné z: <http://www.voipsupply.com/cisco-hd-voice>
- [14] SILK Audio Codec: SILK: Super Wideband Audio Codec. *Skype Developer - SILK* [online]. 2013 [cit. 2013-02-23]. Dostupné z: <http://dev.skype.com/silk>.
- [15] MP3. In: *Wikipedia: the free encyclopedia* [online]. 2014 [cit. 2014-12-28]. Dostupné z: <http://en.wikipedia.org/wiki/MP3>
- [16] *Opus* [online]. 2014 [cit. 2014-12-30]. Dostupné z: <http://www.opus-codec.org/>
- [17] Úvod do metód spracovania zvuku v súčasnom multimedialnom prostredí: Časť 4.3: MPEG-2/4 Audio (AAC). *Zvuk.atrip* [online]. 2006 [cit. 2014-12-31]. Dostupné z: http://zvuk.atrip.sk/index.php?site=4_3
- [18] *Abouttech: HE-AAC Definition: What is The HE-AAC Format?* [online]. 2014 [cit. 2015-01-01]. Dostupné z: <http://mp3.about.com/od/glossary/g/He-Aac-Definition-What-Is-The-He-Aac-Format.htm>
- [19] Audio Samples. *Tieleine* [online]. 2014 [cit. 2015-01-02]. Dostupné z: <http://www.tieline.com/Audio-Samples>
- [20] About aptX®. *AptX* [online]. 2015 [cit. 2015-01-02]. Dostupné z: <http://www.aptx.com/about>
- [21] AptX Live®. CSR [online]. 2015 [cit. 2015-01-02]. Dostupné z: <http://www.csr.com/products/59/aptx-live>
- [22] What's the Difference Between Raspberry Pi Model A and B?. *Hack Things* [online]. 2013 [cit. 2015-04-11]. Dostupné z: <http://www.hackthings.com/raspberry-pi-model-a-and-b/>

[23] RASPBERRY PI 2 ON SALE NOW AT \$35. *Raspberry Pi* [online]. 2015 [cit. 2015-04-11]. Dostupné z: <https://www.raspberrypi.org/raspberry-pi-2-on-sale/>

[24] QUICK START GUIDE. *Raspberry Pi* [online]. 2015 [cit. 2015-04-11]. Dostupné z: <https://www.raspberrypi.org/help/quick-start-guide/>

6.1. Seznam obrázků

Obr. 1: Referenční model TCP/IP a ISO/OSI.....	2
Obr. 2: Třídy IP.....	3
Obr. 3: Rozdíl mezi IPv4 a IPv6 - převzato [8].....	5
Obr. 4: Hlavička TCP.....	6
Obr. 5: Hlavička UDP.....	6
Obr. 6: Hlavička RTP [9].....	7
Obr. 7: Princip PCM (modulace a demodulace).....	9
Obr. 8: Signál s vysokou vzorkovací frekvencí.....	9
Obr. 9: Signál s nízkou vzorkovací frekvencí.....	9
Obr. 10: Porovnání a-law s μ -law.....	13
Obr. 11: G.711 s prodlužovacími vrstvami.....	14
Obr. 12: Porovnání G.711 a G.722.....	14
Obr. 13: Přehled MOS/datový tok [kbit/s] + MOS/ztráta paketů [%].....	18
Obr. 14: Porovnání kvality na bitové rychlosti.....	19
Obr. 15: Dekódování iLBC.....	20
Obr. 16: Kódování iLBC.....	20
Obr. 17: Hierarchická struktura AAC profilu.....	23
Obr. 18: Vývoj AAC s roky.....	24
Obr. 19: MOS / přenosová rychlost kodeků.....	29
Obr. 20: Celkové zpoždění / přenosová rychlost.....	30
Obr. 21: Porovnání modelů RPi [22].....	32
Obr. 22: Vzhled Raspberry Pi 2 [23].....	33
Obr. 23: Zvuková karta UCA202 a její popis.....	34
Obr. 24: Příkaz uname na RPi (2).....	36
Obr. 25: Příkaz lsusb (výpis USB zařízení).....	39
Obr. 26: Příkaz amixer.....	39
Obr. 27: Výpis konfiguračního souboru alsabase.conf.....	40
Obr. 28: Ověření změny v amixer.....	41
Obr. 29: Ověření změny v alsamixeru.....	42

Obr. 30: Piny na RPi a RPi 2.....	44
Obr. 31: Schéma ukázkového zapojení	45
Obr. 32: Schéma zapojení RPi pomocí GPIO pinů	46
Obr. 33: Vzhled webového serveru na RPi	48
Obr. 34: Architektura diplomové práce	49
Obr. 35: Zapojení v praxi	50

6.2. Seznam tabulek

Tab. 1: Třídy IP s parametry	4
Tab. 2: Porovnání kvality záznamu a jejich parametry	10
Tab. 3: Srovnání vzorkovacích frekvencí.....	21
Tab. 4: Srovnání přenosových rychlostí.....	21
Tab. 5: Srovnávací tabulka.....	25
Tab. 6: MOS tabulka hodnocení.....	28
Tab. 7: Základní vlastnosti kodeků	29
Tab. 8: Porovnání verzí Raspberry Pi.....	33

Přílohy

Seznam příloh:

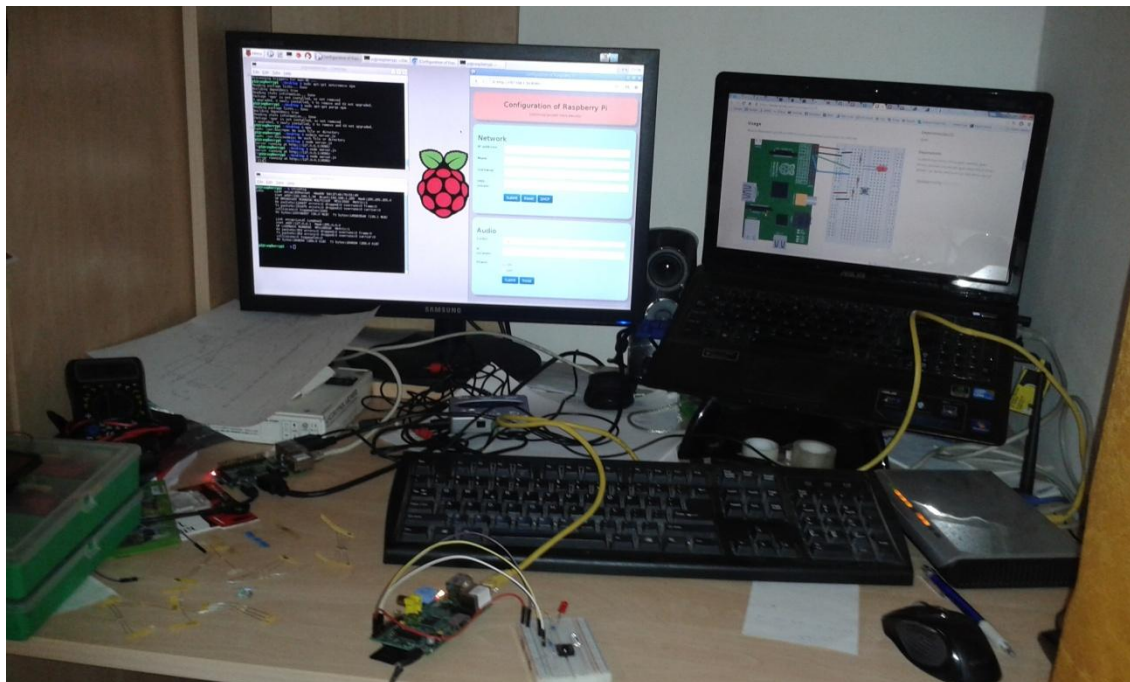
Příloha 1 - Praktické zapojení a testování

Příloha 2 - Pohled shora na zapojení

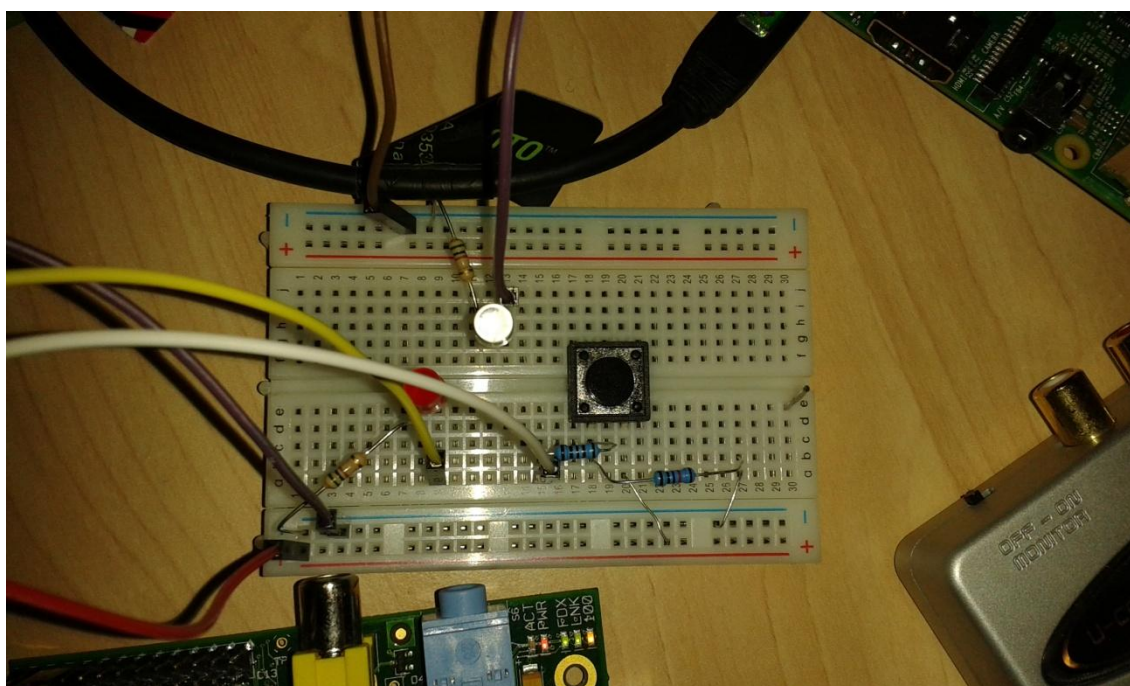
Příloha 3 - Pohled z boku na zapojení

Příloha 4 - Zapojení finální práce a testování

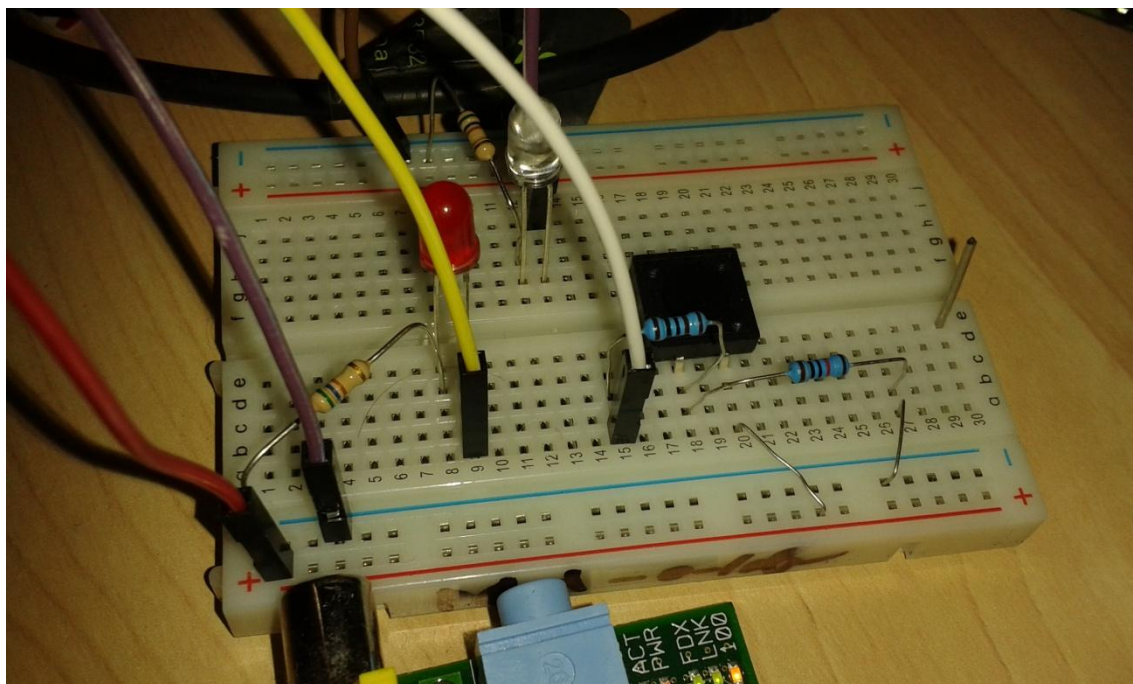
Příloha 5 - Zbylá fotodokumentace a videa na CD



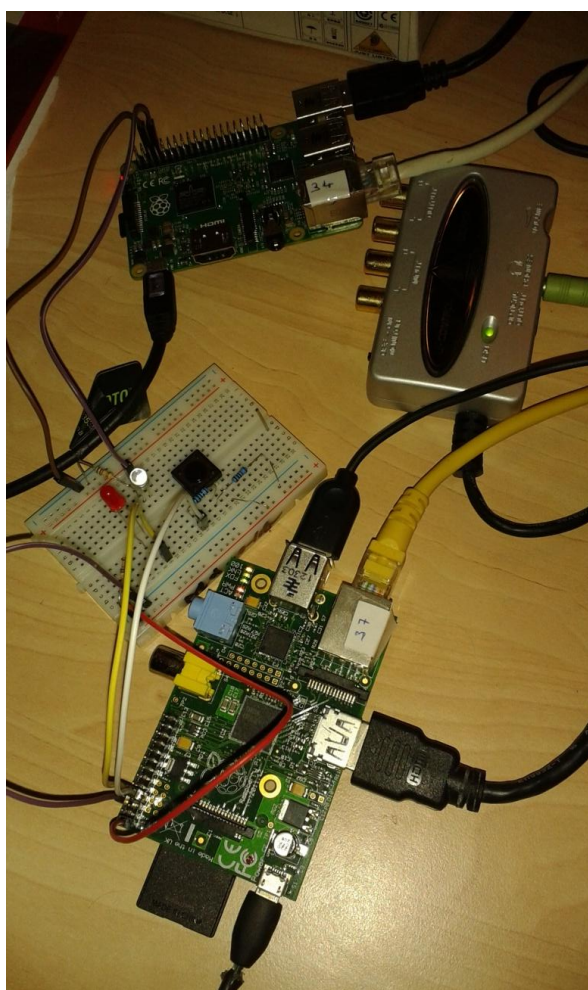
Příloha 1 -Praktické zapojení a testování



Příloha 2 -Pohled shora na zapojení



Příloha 3 - Pohled z boku na zapojení



Příloha 4 - Zapojení finální práce a testování