

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

DIPLOMOVÁ PRÁCE

**Návrh bluetooth přijímacího modulu pro bezdrátový
poslech hudby**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Miroslav VANC**
Osobní číslo: **E17N0037P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a aplikovaná informatika**
Téma práce: **Návrh bluetooth přijímacího modulu pro bezdrátový poslech hudby**
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování

1. Uveďte problematiku, výhody a nevýhody bezdrátového přenosu hudby s ohledem na v současnosti používaná řešení.
2. Navrhněte a zrealizujte přijímací modul komunikující s vysílacím zařízením prostřednictvím bluetooth. Koncipujte modul tak, aby bylo možné co nejtěsnější připojení ke sluchátkům.
3. Vytvořte aplikaci pro vysílací zařízení, přes kterou bude možno přijímací modul ovládat.
4. Zhodnoťte navržené zařízení (případě uveďte možnosti dalšího vylepšení či rozšíření).

Rozsah diplomové práce: **40 – 60 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování diplomové práce: **tištěná/elektronická**

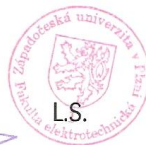
Seznam doporučené literatury:

Profile Specifications [online]. [cit. 2019-06-05]. Dostupné z:
<https://www.bluetooth.com/specifications/profiles-overview/>

Vedoucí diplomové práce: **Ing. Kamil Kosturik, Ph.D.**
Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: **4. října 2019**
Termín odevzdání diplomové práce: **28. května 2020**


Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 4. října 2019

Abstrakt

Předkládaná diplomová práce je zaměřena na problematiku bezdrátového přenosu audiodat. Pro demonstraci souvisejících problémů a návrhů vhodných řešení jsou do rámce této práce zahrnuty návrh a realizace bluetooth modulu pro vložení do sluchátek.

Klíčová slova

Audio, Bluetooth, profil A2DP, SBC audio kodek, ESP32, XS3868

Abstract

The master thesis presents the matters of the wireless data transmission. For the demonstration of related issues and suggestions of appropriate solutions, the thesis also incorporates making a layout of and constructing a bluetooth module, which should be connected to headphones.

Key words

Audio, Bluetooth, A2DP profile, SBC audio codec, ESP32, XS3868

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....
podpis

V Plzni dne 17.6.2020

Miroslav Vanc

Poděkování

Tímto bych rád poděkoval Ing. Petru Weissarovi, Ph.D. za cenné profesionální rady a připomínky a dále vedoucímu diplomové práce Ing. Kamilu Kosturikovi, Ph.D. za metodické vedení práce.

Obsah

OBSAH	8
ÚVOD	9
SEZNAM SYMBOLŮ A ZKRATEK	10
1 VÝHODY A NEVÝHODY BEZDRÁTOVÉHO PŘENOSU	11
2 BLUETOOTH	12
2.1 PRINCIP PŘENOSU PŘES RÁDIOVÝ KANÁL.....	13
2.2 BLUETOOTH STACK.....	14
2.2.1 <i>Baseband</i>	15
2.2.2 <i>Link Manager Protocol</i>	16
2.2.3 <i>L2CAP</i>	16
2.2.4 <i>SDP</i>	17
2.2.5 <i>RFCOMM</i>	17
2.3 BLUETOOTH PROFILY.....	17
2.3.1 <i>A2DP</i>	18
2.3.2 <i>AVRCP</i>	19
2.3.3 <i>SPP</i>	20
2.4 BLUETOOTH LOW ENERGY.....	21
2.4.1 <i>LE Audio</i>	22
3 AUDIO KODEKY	23
3.1 KODEK SBC.....	24
4 NÁVRH A REALIZACE BT-PCB	26
4.1 MODUL XS3868.....	26
4.2 MODUL ESP32.....	28
4.2.1 <i>Návrh PCB</i>	28
4.2.2 <i>Software</i>	34
4.2.3 <i>Aplikace pro zařízení Android</i>	37
5 ZÁVĚR	39
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	40
PŘÍLOHY	1

Úvod

Žijeme v době, kdy poměrně velká část elektronických zařízení využívá moderní bezdrátové technologie. Tyto technologie poskytují pohodlná řešení pro síťové a mobilní komunikace a jsou také hojně skloňovány v souvislosti s nadcházejícím fenoménem IoT. Ve spoustě odvětvích se staly symbolem zjednodušení a zpohodlnění uživatelského přístupu. Můžeme si ale být jisti, že je bezdrátové řešení v každém případě výhodné? Není někdy prosazováno na úkor kvality dotčeného systému? Mluvím teď zejména o aplikacích, kde kvalita datového přenosu sehrává klíčovou roli. Jde jistě hlavně o to, jaké jsou na konkrétní aplikace kladena očekávání.

Téma méj diplomové práce se zabývá bezdrátovým přenosem hudby s využitím bezdrátové technologie Bluetooth. Má tedy za úkol detailně prozkoumat, jakým způsobem tento přenos audiodat probíhá, následně poukázat na možné problémy nebo nevýhody, které s tímto typem přenosu souvisí, a pokud je to jen trochu možné, navrhnout použitelná řešení. Pro lepší názornost je součástí práce také návrh a realizace Bluetooth modulu, který poslouží k praktické demonstraci problémů souvisejících s bezdrátovým přenosem a pokusí se je alespoň částečně eliminovat. V rámci této práce budu tento vytvářený modul pro lepší přehlednost a srozumitelnost nazývat zkratkou BT-PCB.

Seznam symbolů a zkratek

BT	Bluetooth
PCB.....	Printed Circuit Board
IoT	Internet of Things
ISM.....	Industrial, Scientific and Medical
BR/EDR.....	Basic Rate / Enhanced Data Rate
LE	Low Energy
OUI.....	Organizationally Unique Identifier
SCO	Synchronous Connection-Oriented
ACL	Asynchronous Connection-Less
SRC.....	Source
SNK	Sink

1 Výhody a nevýhody bezdrátového přenosu

Prvním a nejvíce zjevným přínosem bezdrátových systémů je samotný fakt, že k přenosu dat není potřeba propojovacích kabelů. Výhoda tedy spočívá hlavně v daleko větší prostorové flexibilitě těchto systémů. U audio systémů a sluchátek odpadají nepříjemnosti při nechtěném vytrhnutí kabelu a zamezuje se tak eventuálnímu poškození zařízení. Výhodou může být také relativně nízká cena, kdy vlivem současného trendu je v komerční oblasti vyšší poptávka právě po bezdrátových systémech a velká konkurence na trhu s elektronikou přispívá ke snížení ceny těchto výrobků.

Hlavním problémem u bezdrátových audio aplikací může být omezená rychlost datového přenosu. Šířka pásma rádiového kanálu nemusí vždy stačit pro přenos dostatečně velkého toku dat, který by zajistil věrnou zvukovou reprodukci původního audio signálu. Digitální signál navíc musí mimo jiné projít fází modulace a demodulace, což do systému zavádí určité zpoždění. Souvisejícím faktorem je pak také spotřeba energie a výdrž baterie použité v zařízení [1]. Je zřejmé, že v těchto systémech je nutno hledat kompromis mezi kvalitou přenosu a průměrnou dobou provozu zařízení na jedno nabití.

Přenos přes rádiové médium je také náchylnější na elektromagnetické rušení. Vysokofrekvenční vlny mohou být během propagace rušeny od jiných zařízení pracujících na podobných frekvencích, a přenos tak nemusí vždy být zcela spolehlivý [2]. Pro bezdrátový přenos je samozřejmě nutná přítomnost speciálních modulačních a demodulačních obvodů, vysílací a přijímací antény a dalších nezbytných rádiových komponent, jejichž provedení výrazně ovlivní výsledný audio signál. Na přijímací straně musí být kromě radiopřijímače i odpovídající dekodér, převodník DAC, filtry a zesilovač signálu.

V současnosti se pro komerčně využívaný bezdrátový přenos audia používají zpravidla technologie Bluetooth nebo WiFi. Původně měl Bluetooth sloužit pouze jako bezdrátová varianta linky RS-232 pro obyčejný přenos malého množství dat mezi dvěma zařízeními na malou vzdálenost. Postupem času se technologie přenosu zdokonalila a v roce 2004 byla na komerční trh poprvé uvedena první bezdrátová sluchátka určená pro poslech hudby [3]. Se stoupající rychlostí datového přenosu a vývojem efektivnějších audiokodeků lze v

současnosti dosáhnout téměř stejně kvalitního poslechu jako u klasických kabelových systémů.

2 Bluetooth

Bluetooth je standardizovaná bezdrátová technologie vhodná pro přenos dat mezi zařízeními v krátké vzdálenosti od sebe pomocí ultra-vysokofrekvenčních rádiových vln. Pracuje v nelicencovaném pásmu 2.4 GHz, rezervovaném pro průmyslové, vědecké a zdravotnické aplikace (ISM). V tomto pásmu kromě něj pracují i další moderní bezdrátové technologie, jako WiFi, Zigbee a další [4]. V současnosti existují v rámci Bluetooth dva typově odlišné systémy - původní Bluetooth Classic (BR/EDR) a novější Bluetooth Low Energy, který byl vydán jako součást BT verze 4.0 a nadále expanduje ve verzích 5.0 a 5.1. Ten je v principu navržen pro systémy s požadavkem na velmi nízkou spotřebu [5].

Každé Bluetooth zařízení by mělo mít přidělenou vlastní unikátní 6-byteovou adresu, která sestává z unikátního identifikátoru výrobce (OUI) a komplementární jedinečné kombinace bitů. Při vyhledávání jiných zařízení se uživateli pro přehlednost často místo adresy zobrazuje název zařízení. Podle výkonu se Bluetooth zařízení dělí do 3 tříd s různě definovanými operačními vzdálenostmi. Zařízení náležící do 1. třídy má výkon 100 mW a dosah 100 metrů. Zařízení 2. třídy má výkon 2,5 mW a dosah 10 metrů. Zařízení 3. třídy má výkon 1 mW a komunikuje na vzdálenost menší zhruba 10 centimetrů [4]. Naprostá většina BT audio zařízení náleží do 2. třídy, která poskytuje rozumný kompromis mezi elektrickým příkonem a fyzickým dosahem.

Audio aplikace jsou dnes jednou z největších oblastí využití technologie BT a zařízení podporující audio přenosy tvoří až 47 % celkové produkce Bluetooth zařízení. V roce 2019 bylo vyexpedováno 4,2 miliard Bluetooth zařízení, z nichž 1,1 miliardy byla typu „audio-sink“. Podle současných statistik je trh s bezdrátovým audiem na vzestupu, 90 % přenosných reprosoustav nyní využívá technologii Bluetooth. Po rozšíření nového LE audia na komerční trh během let 2020 a 2021 se předpokládá až dvojnásobný nárůst počtu expedovaných audio zařízení a částečné pokrytí trhu s pomůckami pro sluchově postižené [6].

2.1 Princip přenosu přes rádiový kanál

Pro čtenářovu lepší představu o funkcionalitě BT a tím i pro lepší porozumění problematice přenosu audia přes BT je zde nejprve nastíněn princip přenosu dat mezi BT zařízeními. Následující text se bude zabývat převážně principem přenosu dat u systémů BT Classic (BR/EDR). Relativně nový systém BT Low Energy zavádí ve fyzických vrstvách zařízení nové a lehce odlišné metody pro přenos dat, které jsou ve stručnosti vysvětleny v kapitole 2.4. Veškeré informace v kapitolách 2.1 a 2.2 vycházejí ze specifikace Bluetooth verze 4.2.

Aby byl zaručen spolehlivý provoz v pásmu 2.4 GHz, používá se pro BT komunikaci metoda adaptivních frekvenčních skoků, kde zařízení vysílají skrze 79 frekvenčních kanálů v rozmezí 2.402 - 2.480 GHz. Frekvenční krok mezi dvěma sousedními kanály odpovídá 1 MHz. Využívá se zde binární frekvenční modulace GFSK nebo i komplexnější modulace vyhrazené pro vyšší přenosové rychlosti (EDR). Různé formy fyzické vrstvy BT kontroléru využívají různé modulace a obecně podporují rychlosti od 1 Mbps do 3 Mbps. Rovněž podporují několik výkonostních režimů v rozmezí 1 mW až 100 mW, jak je uvedeno v předchozí kapitole [7].

Pro komunikaci v rámci rádiového kanálu vytváří BT zařízení síť zvanou „piconet“, kde zaujímají klasický komunikační model **Master - Slave**. Zařízení **M** zajišťuje řízení datových transakcí v rámci piconetu a může komunikovat až se sedmi různými zařízeními **S**, která však přímo mezi sebou komunikovat nemohou. Zařízení **M** jim poskytuje synchronizaci v podobě hodinového signálu a specifického schématu frekvenčních skoků po dostupných 79 frekvencích. Toto schéma skoků je tvořeno pseudonáhodnou posloupností, vycházející z unikátní adresy zařízení **M**, ale je adaptivní a pokud se v dosahu vyskytnou zařízení, která pracují na jedné stálé frekvenci v dotčeném pásmu, potom BT tyto zarušené frekvence přestane využívat [7].

Přenosový kanál používá časový multiplex, kde čas je rozdělen do „slotů“ o trvání 625 μ s. Data jsou vysílána po paketech, které mohou mít různou velikost a pro každý z nich může být vyhrazeno více po sobě jdoucích slotů. V případě, že by každý paket zabíral jeden slot, tak v jedné sekundě je možno přenést až 1600 paketů. Po odvysílání nebo přijetí každého jednoho paketu se provede skok na jiný frekvenční kanál. Při použití schématu

TDD (Time Division Duplex) je možno dosáhnout plně duplexní komunikace, kdy zařízení **M** vysílá v sudých slotech a zařízení **S** v lichých [7].

V rámci fyzického rádiového kanálu je po vzájemném připojení mezi zařízeními **M** a **S** vytvořen fyzický link. Tím je zaručena možnost obousměrného datového transportu mezi těmito zařízeními. Fyzický link v zásadě slouží jako transportní médium pro jeden nebo častěji více tzv. logických linků. Ty jsou na fyzickém linku časově multiplexovány do jednotlivých slotů. Logické linky mohou podporovat různé typy logických transportů jako synchronní (SCO), asynchronní (ACL), isochronní nebo „broadcast“ přenosy [7].

Z hlediska vyšších protokolových vrstev je vytvoření spojení u dvou Bluetooth zařízení proces, který lze rozdělit do tří fází. Při vyhledávání okolních zařízení je nejprve vyslán dotaz („inquiry“) a všechna naslouchající zařízení dopoví odesláním své adresy, jména, případně dalších informací. Následuje fáze připojování („paging“), při které se vytváří spojovací linka mezi oběma zařízeními. Po jejím úspěšném vytvoření jsou již zařízení ve stavu připojení („connection“). V tomto stavu mohou připojená **S** zařízení setrvat v aktivním režimu, kdy se aktivně účastní komunikace s **M**, nebo mohou přejít do jednoho z dostupných low-power režimů [7].

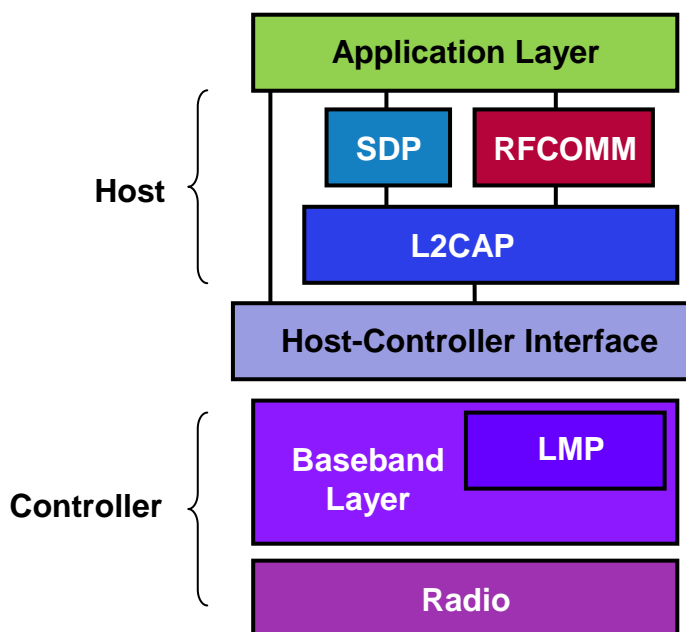
Pokud mají zařízení mezi sebou sdílet svoje vlastní údaje, jako je adresa, jméno, seznam použitelných BT profilů atd., musí provést takzvané párování („pairing“). To zahrnuje autentizační proces, kdy je potřeba uživatelsky schválit takovéto spojení. Autentizační proces může být pro každé zařízení jiný, od jednoduchých „odkliknutí“ po složitější zadávání šestimístného číselného kódu. Po procesu párování je pro obě zařízení možné, aby po dostatečném přiblížení k sobě automaticky vytvořily pouto („bonding“) bez jakéhokoli přičinění ze strany uživatele [7].

2.2 Bluetooth stack

Architektura základního systému BT sestává ze dvou oddělených logických entit „Host“ a „Controller“. Entita „Host“ obsahuje služby a protokoly vyšších vrstev BT stacku a je společný pro oba systémy BT Classic a BT Low Energy, zatímco „Controller“ obsahuje nízkourovňové protokolové vrstvy, které jsou pro oba systémy rozdílné. Tato část proto existuje ve třech variantách: „BR/EDR Controller“, „LE Controller“ nebo kombinace

obou pro zařízení Dual-Mode s podporou BT Classic i BT Low Energy. Mezi logickými entitami „Host“ a „Controller“ se nachází asimilační rozhraní „Host-Controller Interface“ (HCI) [7]. V následujícím textu se budu věnovat protokolové sadě, která se opět týká převážně systémů BT Classic (BR/EDR).

Základní protokoly využívané v nízkourovňové vrstvě „Controlleru“ BT systému jsou rádiový protokol PHY, protokol LC (Link Control) a LMP (Link Manager Protocol). Ve vyšších vrstvách systému BT, které spadají do logické entity „Host“ je pro činnost BT systému nutná implementace protokolu L2CAP (Logical Link Control and Adaptation Protocol) a protokolu servisní vrstvy BT stacku SDP (Service Discovery Protocol) [7].



Obr. 2.1 Obecná architektura BT systému [7]

2.2.1 Baseband

Nejnižší vrstvy architektury zodpovědné za vysílání a přijímání paketů dat jsou řízeny pomocí několika funkčních bloků. „Baseband Resource Manager“ je zodpovědný za veškerý přístup na rádiové médium. V jádru komponenty má plánovač s rozvrhem času pro všechny entity, které požádají o přístup na fyzický kanál. Musí brát ohled na všechny žádosti o užití fyzické vrstvy kontroléru - přenos dat, skenování, vytváření spojení, nastavení viditelnosti a připojitelnosti zařízení. Pomocí dalších bloků se ovládá i časování

a frekvence nosné ve fyzické vrstvě, která transformuje tok dat z rádiového média do patřičných formátů [7].

Na nižší vrstvě pracuje také Link Controller, jehož úkolem je enkódování a dekodování datových paketů a informačních dat o parametrech fyzického kanálu a logických linků. Komunikuje skrze řídicí linkový protokol, kterým nastavuje a řídí datový tok a vysílá potvrzení nebo žádosti o opětovnou transakci [7].

2.2.2 Link Manager Protocol

Řídicí protokol pro baseband a fyzické vrstvy (LMP) je přenášen spolu s uživatelskými daty asynchronním transportem ACL, který je vytvořen, jakmile Slave vstoupí do piconetu. Pomocí LMP se řídí operace zařízení v piconetu a jsou poskytovány služby využitelné v nižších vrstvách architektury. Skrze tento protokol komunikuje blok Link Manager, který takto zajišťuje tvorbu a úpravu logických linků a taky dohlíží na aktuálnost parametrů fyzických linků mezi zařízeními. Provádí to prostřednictvím komunikace s Link Managerem vzdáleného BT zařízení. Protokol LMP umožňuje vytváření nových logických linků i nových logických transportů a jejich řízení [7].

2.2.3 L2CAP

Nad rozhraním mezi oběma logickými entitami poskytuje vrstva L2CAP jakousi „kanálovou abstrakci“ pro účely aplikací a služeb ve vyšších uživatelských vrstvách. Provádí segmentaci a znovusestavení aplikačních dat a multiplexuje, resp. demultiplexuje více těchto L2CAP kanálů do, resp. z logických linků. Protokol má svůj vyhrazený pevný kanál, přenášený transportem typu ACL. Aplikační data z vyšší vrstvy mohou využít jakýkoli logický link, který podporuje protokoly L2CAP [7].

Pomocí protokolu L2CAP komunikuje Channel Manager, který má na starost tvorbu, řízení a uzavírání L2CAP kanálů pro transport servisních protokolů a uživatelských dat z aplikační vrstvy. Používá protokol L2CAP ke komunikaci s Channel Managerem u vzdáleného zařízení pro vytvoření těchto kanálů a pro jejich správné zakončení. Také komunikuje s Link Managerem v nižší vrstvě, který podle potřeby vytváří nové logické linky pro přenos L2CAP kanálů [7].

Funkční blok L2CAP Resource Manager je zodpovědný za správné řazení podávaných PDU fragmentů do vrstvy baseband a řídí rozvrh pro muxování L2CAP kanálů. Je to zapotřebí z toho důvodu, že architektonický model počítá jen s omezeným datovým bufferem v rámci logické entity „Controller“ [7].

2.2.4 SDP

Protokol SDP umožňuje aplikacím ve vyšších vrstvách zjišťovat, jaké služby jsou ve vzdáleném zařízení dostupné a jak jsou tyto služby pro dotyčné zařízení charakterizovány. Vzdálené zařízení, které vysílá zpět seznam dostupných služeb, by mělo také poskytnout návod pro žádající zařízení, jak může tyto služby vyvolat a využít. Dostupnost služeb u vzdáleného zařízení se může dynamicky měnit, například může záviset na blízkosti obou zařízení. Proto je třeba protokol speciálně navržený pro prostředí BT, který v těchto podmínkách umí pracovat [7].

2.2.5 RFCOMM

Jedná se o transportní protokol, který poskytuje emulaci sériového portu skrze kanály L2CAP. V principu dokáže zprostředkovat až 60 simultánních sériových linek mezi dvěma BT zařízeními. Používá se pro pokrytí aplikací, které pro komunikaci mezi zařízeními využívají sériové porty. Tento protokol mohou kromě počítačů, mikrokontrolérů a příslušných akčních členů využívat také modemy a jiná komunikační zařízení [7].

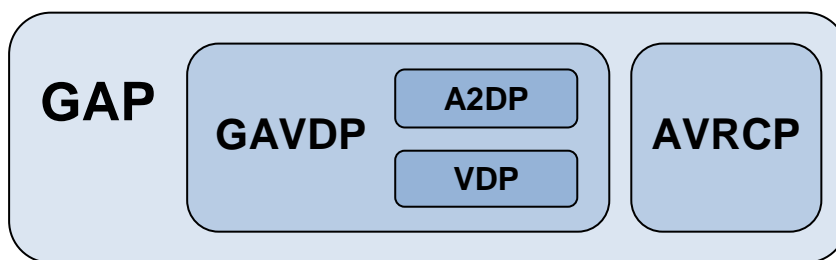
2.3 Bluetooth profily

Obecně jsou speciální funkční komponenty v rámci BT definovány v takzvaných profilech. Profil GAP (Generic Access Profile) funguje jako přístupový bod pro komunikaci mezi dvěma BT zařízeními. Reprezentuje základní funkcionalitu a služby společné pro všechna BT zařízení, jako například procedury pro nalezení dostupných zařízení, procedury pro nalezení dalších služeb, volbu módu připojení, bezpečnostní a autentizační služby. Pro různé módy připojení definuje specifické parametry připojení a přístupové procedury užívané všemi transporty, potažmo všemi protokoly a aplikačními profily [7].

Pro účely této práce budou v následujícím textu podrobněji prozkoumány pouze ty profily, které byly explicitně použity v rámci aplikačního softwaru nahraného do vytvořeného modulu.

2.3.1 A2DP

Profil A2DP (Advanced Audio Distribution Profile) definuje protokoly a procedury, které realizují distribuci mono nebo stereo audia vyšší kvality („high quality stereo“) skrze kanály typu ACL. Liší se tak od standardního BT audia, které je navrženo pouze pro komunikační účely a přenos hlasu a využívá pouze úzké frekvenční pásmo na kanálech typu SCO. Audio data jsou zde komprimována do patřičného formátu pro efektivní využití omezeného pásma. A2DP je podmíněn přítomností profilů GAP a GAVDP, které definují obecné procedury pro nastavení A/V streamingu. Samotný A2DP se zaměřuje na parametry a procedury použité výhradně pro audio streaming [8].



Obr. 2.2 Podmíněnosti BT audio a video profilů [8]

Zařízení využívající tento profil mohou v rámci jednoho *piconetu* plnit jednu ze dvou určených rolí, a to buď roli zdroje audiostreamu, nebo roli přijímače. V klasickém schématu figuruje mobilní telefon jako zdroj a libovolný audiosystém jako přijímač. A2DP ke své funkci v obou těchto rolích využívá stack složený z protokolů Baseband, LMP, L2CAP a SDP, definovaných v základní specifikaci. Součástí protokolové sady je zde navíc i protokol AVDTP (Audio/Video Distribution Transport Protocol), který zajišťuje pro obě strany kompatibilní nastavení streamovacích parametrů a dále ovládá samotný streaming [8].

Ve zdrojovém zařízení SRC musí být dostupná procedura „Send Audio Stream“ a naopak v přijímacím zařízení je nutná inverzní procedura „Receive Audio Stream“. Pokud je verze protokolu AVDTP u přijímacího zařízení SNK pro SRC neznámá, pak ještě před založením GAVDP spojení musí zdrojové zařízení prostřednictvím servisního protokolu

SDP vyslat dotaz k přijímacímu zařízení, které mu obratem pošle číslo verze svého protokolu AVDTP. Na verzi tohoto protokolu totiž závisí několik příkazů týkajících se nastavení audio streamingu. Posléze je potřeba založit spojení pro streaming - patřičné procedury jsou definovány ve specifikaci profilu GAVDP. Tehdy se obě zařízení musí shodnout na nejvhodnějších parametrech pro datový přenos. A2DP specifikuje parametry týkající se audia, tedy použitelnou vzorkovací frekvenci, bitovou hloubku, atd. Tyto parametry se potom dosadí do příslušných procedur [8].

Profil má tedy na starosti nastavení, řízení a manipulaci toku audiodat od zdroje k přijímači. Mezi problémy, které vyvstávají u přenosu audia, patří zpoždění signálu s hodnotami přibližně 100 - 150 milisekund, které vzniká vlivem doby zpracování rádiového signálu, buffering dat a enkódování / dekodování audio toku. Dalším omezením se týká velikosti nebo „šířky“ přenášených audiodat. Ta musí být s určitou rezervou menší než použitelná šířka BT linky, a to proto, aby bylo umožněno opětovné vyslání chybně přijatých dat. Tímto způsobem se v datovém přenosu zamezuje ztrátám paketů. Je nutné si také uvědomit, že kromě samotného audia se v datovém toku přenáší i patřičné protokolové pakety [8].

V rámci tohoto profilu jsou dále definovány požadavky pro interoperabilitu kodeků a jejich parametry. Obě BT zařízení musí z definice profilu podporovat kodek SBC, mezi další volitelné kodeky definované v profilu A2DP patří i MPEG-1,2 Audio, MPEG-3,4 AAC a kodeky ATRAC. Pokud vysílač podporuje enkódování některého z volitelných kodeků a zároveň přijímač podporuje jeho dekodování, mohou se audiodata přenášet ve formátu příslušného kodeku. Pokud ale chce zdrojové zařízení vysílat data, jejichž kodekový formát není podporován na přijímací straně, pak tato data musí být transkódována do formátu SBC [8].

2.3.2 AVRCP

Profil AVRCP definuje procedury potřebné k zajištění interoperability mezi BT zařízeními s A/V řídicími funkcemi. Používá sadu příkazů „AV/C Digital Interface Command Set“ a realizuje jejich snadnou implementaci. Zařízení využívající tento profil zaujmají standardní model pro vysílání a příjem řídicích audio a video funkcí a používají specifický příkazový formát pro řídicí signály, které jsou poté transportovány pomocí protokolu AVCTP [9].

BT kontrolér zde překládá uživatelskou akci na A/V řídicí signál a vysílá ho na vzdálené zařízení. Kromě řídicích příkazů se v rámci tohoto profilu mohou přenášet metadata přímo spjatá s konkrétním A/V obsahem. Profil používá v nižších vrstvách systému protokoly totožné s těmi, co používá profil A2DP, ve vyšší vrstvě má ale vlastní speciální AVCTP protokol, který definuje procedury a zprávy vyhrazené pro řízení A/V zařízení [9].

V rámci tohoto profilu mohou zařízení opět zaujmout jednu ze dvou rolí. Kontrolér iniciuje transakci vysláním příkazového rámce cíli, který po jeho přijetí způsobně vygeneruje rámeček odezvy. Zde je například důležitá dostatečná responsivita a rychlá reakce na odeslaný příkaz. Latence pro přijetí příkazu je závislá na konkrétní aplikaci, ale většinou se pohybuje kolem 100 - 200 ms [9].

2.3.3 SPP

Profil SPP (Serial Port Profile) definuje protokoly a procedury, které jsou použity pro komunikaci mezi zařízeními skrze virtuální sériový port. Je podmíněn přítomností profilu GAP a zároveň je podmínkou pro správnou funkci jiných profilů, jako je třeba HSP, nebo profilu určeného pro přenos souborů. Ke své funkci využívá protokoly ve vrstvě Baseband, LMP, L2CAP, SDP a navíc RFCOMM, který poskytuje transportní kanál emulující klasický sériový port [10].

Zařízení v rámci tohoto profilu mohou opět zastávat jednu ze dvou rolí. Iniciátor vytváří spojení s druhým zařízením, akceptorem. SPP v přeneseném významu poskytuje mezi těmito dvěma zařízeními jednoduché programovací rozhraní pro specifické uživatelské aplikace (API). Tyto aplikace ale nemohou fungovat stejně, jako kdyby využívaly klasickou fyzickou sériovou linku, proto je pro celkovou interoperabilitu na úrovni BT nutnost dodržet chování specifické pro tento profil [10].

Profil vyžaduje podporu minimálně pro jednoslotové pakety, což zajišťuje přenosové rychlosti do 128 kb/s. Dále je zde vyžadována podpora pro autentikaci a enkrypci, i když se nemusí v aplikacích uplatnit. Nepoužívá se zde bonding, pouze párování. Protokol RFCOMM je zde použit pro transport uživatelských dat, řídicích signálů pro modemy a konfiguračních příkazů [10].

2.4 Bluetooth Low Energy

Stejně jako Bluetooth Classic využívá metodu frekvenčních skoků v rozprostřeném spektru 2.402 - 2.480 GHz, ale využívá zde pouze 40 kanálů s frekvenční vzdáleností 2 MHz od sebe. Tři z těchto kanálů jsou vyhrazeny pro takzvaný „advertising“ - periodické vysílání krátkých datových sekvencí vhodných pro snadné poskytování různých informací uložených ve vysílajícím zařízení. Stejně jako systém BT Classic používá modulaci GFSK. Různé formy fyzické vrstvy podporují přenosové rychlosti v rozsahu 125 Kbps až 2 Mbps a opět několik výkonostních režimů v rozmezí od 1 mW po 100 mW. Narozdíl od BT Classic podporuje BT Low Energy několik topologií - klasické schéma „point-to-point“, dále režim „broadcast“, ale i komplikovanější síťová schémata „mesh networking“. Ani v případě LE nejsou podporovány přímé fyzické linky mezi více **S**. Narozdíl od BT Classic není v rámci sítě počet **S** limitován [5].

Fyzický kanál u systémů LE je časově dělen na tzv. „eventy“. Data jsou mezi LE zařízeními přenášeny v paketech umístěných v těchto eventech. Existují dva typy eventů - „connection“ eventy a „advertising“ eventy. Při přenosu „advertising“ dat není potřeba přímého připojení mezi tzv. „advertiser“ a „scannery“ a může být použito při jednosměrné komunikaci nebo jako prostředek pro založení obousměrné komunikace. Iniciující zařízení čekají právě na takovýto „advertising“ paket, který v sobě nese možnost připojení, a posléze mohou vyslat žádost o připojení. Pokud advertiser od iniciátora žádost přijme, může začít vysílání standardních „connection“ eventů. Iniciátor pak zde přijímá roli **M** piconetu a advertiser se stává **S**. Komunikace přeskočí na jeden z 37 datových kanálů a dokud si **M** a **S** vyměňují data v rámci jednoho connection eventu, kanál se nemění. **M** ho může kdykoli přerušit [7].

Řídící protokol pro nejnižší vrstvy se v systému LE označuje LL (Link Layer), narozdíl od systému Classic, kde se jedná o LMP (Link Manager Protocol), a řídí se jím fyzická a linková vrstva BT stacku. Defaultní logický transport pro zařízení v piconetu je LE ACL, skrze který se přenáší signály LL. Nad linkovou vrstvou je opět abstrakce ve formě L2CAP pro služby a aplikace na vyšších úrovních. Dále jsou ve vyšších vrstvách ještě dvě dodatečné protokolové vrstvy - SMP (Security Manager Protocol), který používá pevný L2CAP kanál pro zajištění bezpečnostních funkcí a ATT, přes který lze

komunikovat malé množství dat a většinou slouží pro zjišťování dostupných služeb a funkcí zařízení [7].

2.4.1 LE Audio

Ostře sledovaným tématem je vydání nového standardu pro přenos audia v rámci Bluetooth Low Energy ohlášeného na veletrhu CES v lednu 2020. LE Audio má být vydáno jako součást verze Bluetooth 5.2 a má přinést revoluční změny do všech oblastí, kde se používá bezdrátový přenos zvuku. Kromě klasických audio-sink systémů, jako jsou sluchátka, headsety nebo reprosoustavy, sem spadají například i pomůcky pro sluchově postižené. Ti by s pomůckami využívajícími LE Audio měli být schopni čerpat ze všech výhod, které má v budoucnu tento standard přinést [11].

Hlavním pilířem LE Audia v oblasti BT přenosu hudby má být využití audiokodeku LC3 (Low Complexity Communications Codec). Zatímco u všech klasických audiokodeků (SBC, aptX, LDAC...) bylo třeba k přenosu dat používat BT Classic pro jeho vyšší dosažitelnou přenosovou rychlost, novému kodeku mohou pro stejně kvalitní nebo i lepší přenos stačit parametry méně energeticky náročného BT Low Energy. Při poslechovém testu „Standard Stereo Listening Test“ audio v komprimovaném formátu LC3 a šířkou dat pouze 192 kb/s a 160 kb/s získalo lepší skóre než audio ve formátu SBC se 345 kb/s. LC3 kodek ale zároveň může být použit jako volitelný kodek v rámci BT Classic, podobně jako aptX, LDAC, a další [11].

Další velkou výhodou pro LE Audio představuje možnost komunikace s více zařízeními najednou, tzv. multistreaming, vlastnost vycházející z původního konceptu BT Low Energy. Vznikne tak standard pro nezávislé, synchronizované toky audiodat. Toho mohou využít například oddělená „true-wireless“ sluchátka, u nichž se tím optimalizuje princip příjmu audia. Lze si představit i další možnosti aplikací, a to zejména vezmeme-li v úvahu využití režimu „broadcast“, přítomného pouze v rámci BT Low Energy [11].

3 Audio kodeky

V rámci profilu A2DP je standardizováno několik audiokodeků. Primárně je zde popsán nízko-komplexní kodek SBC (Sub Band Codec), který je vytvořen speciálně pro Bluetooth a je minimálním požadavkem u všech Bluetooth audio zařízení [8]. A2DP ze zásady umožňuje i použití jiných, lepších kodeků, například MP3, MP2, AAC, aptX, nebo jiných.

Důvodů proč je u bezdrátového přenosu audia potřeba kodek je několik. Nekomprimovaná audio data požadují přenosovou rychlost 1.4 Mb/s a i když například přenos u variant EDR může mít 2 až 3 Mb/s, je tak jako tak velmi obtížné nekomprimovaná data přenášet. Tyto přenosové rychlosti totiž určují maximální rychlost pro přenos kompletní datové informace, tedy včetně patřičných protokolových hlaviček, pro oba směry vysílání. V reálných podmínkách navíc hraje roli vzdálenost mezi vysílačem a přijímačem a prostředí mezi nimi. Záleží na tom, jestli je prostor volný, nebo jestli se v něm vyskytují překážky, přenosovou rychlost ovlivní třeba i vlhkost ovzduší. Faktická rychlost přenosu audio dat je tedy nižší [12], [8].

Pro přenos audio dat se používá asynchronní kanál, využívající například pakety typu 2-DH5 nebo 3-DH5. Tyto mohou nést maximální množství audio dat pro 2 Mb/s a 3 Mb/s EDR přenosové rychlosti, přičemž každý tento paket zabírá 5 slotů. Po každých 5 slotech přichází na řadu přijímací zařízení, které buď pouze potvrdí svoji přítomnost, nebo vyšle zpět menší či větší paket. V případě, že je tento zpětný paket větší než jeden slot, přenos samotného audia z druhé strany musí čekat a tok dat tak ztrácí na rychlosti [12], [7].

Dále se pro přenos audia v rámci profilu A2DP vyžaduje zapouzdření jednotlivých paketů audio dat do transportních protokolů L2CAP a AVDTP, které si z celkového počtu dat v jednom audio paketu vyhradí navíc 16 bytů. Např. 2-DH5 paket může nést maximálně 663 bytů čistých audiodat a jeho přenosová rychlost je 1414 kb/s. To ale bohužel v reálných podmínkách nestačí - pásmo 2.4 GHz je většinou zarušené od různých WiFi zařízení nebo dalších BT zařízení. Paket 3-DH5 sice může přenášet 2143 kbps, na druhou stranu zařízení využívající EDR s přenosovou rychlostí 3 Mb/s vyžaduje vyšší vysílací výkon a vyšší poměr úrovně signálu vůči šumu. Mohou vzde vznikat krátkodobá

přerušení toku a o spolehlivosti přenosu by se dalo mluvit pouze na vzdálenost pár metrů [12].

3.1 Kodek SBC

Je povinný provšechna zařízení podporující standard A2DP. Podporuje vzorkovací kmitočty 16, 32, 44.1 nebo 48 kHz, a 16-b rozlišení. Maximální výrobci používaná přenosová rychlost takto komprimovaného audia je 345 kb/s, i když jsou zde v principu možné i vyšší rychlosti. Je to jednoduchý, výpočetně rychlý kodek, který obsahuje primitivní psychoakustický model využívající adaptivní PCM modulaci. V rámci A2DP specifikace se vyskytuje v dvojí kvalitě - vysoké či střední [8].

Lze u něj provádět nastavení různých parametrů, např. nastavení celkového zpoždění algoritmu, počtu vzorků v bloku, rozdělení frekvenčních pásem nebo volba algoritmu alokace bitů. Parametry používané ve většině aplikací jsou nastaveny pro mód „joint stereo“, signál se rozděluje do 8 frekvenčních pásem, audio rámeček je rozdělen do 8 frekvenčních pásem a 16 bloků audio vzorků a využívá se hlasitostní algoritmus alokace bitů. Velký vliv na přenosovou rychlost má nastavení hodnoty „bitpool“. Pokud je rádiové pásmo zarušené a dochází ke ztrátám paketů, nebo pokud jsou zařízení příliš daleko od sebe, vysílací zařízení může dynamicky redukovat hodnotu bitpool, aby se zabránilo výpadkům audio toku, dokud není spojení opět stabilní [12], [8].

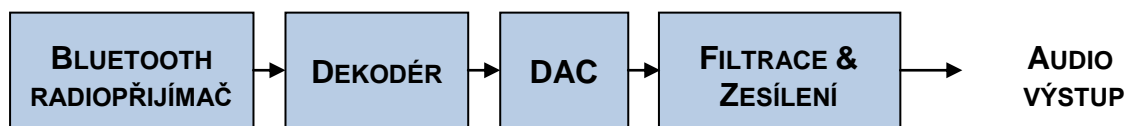
Většina výrobců sluchátek nastavuje hodnotu bitpool na 53, to znamená rychlost 328 kb/s při použití doporučeného nastavení. Použití vyššího bitpoolu by teoreticky znamenalo lepší zvukovou kvalitu, ale zde je překážka v podobě vnitřních limitů Bluetooth stacku v rámci užívaného operačního systému. Tyto limity jsou uměle vytvořeny, a to kvůli nedostatečnému množství certifikačních testů a nekompatibilitě při používání vyšších hodnot bitpoolu, nebo atypických profilů u různých přijímacích zařízení [12].

SBC má také funkci dynamické alokace kvantizačních bitů pro různá frekvenční pásma, od spodních po horní, s různými vahami. Pokud by tedy byl celý bitrate zaplněn nízkými a středními frekvencemi, pak vyšší frekvence budou zcela odříznuty [12], [8].

Celkově je SBC velmi flexibilní kodek a při standardních 328 kb/s může poskytnout obstojnou zvukovou kvalitu. Dále může být konfigurován pro aplikace s potřebou nízkého zpoždění, atd. Standard v profilu A2DP nedefinuje pevné nastavení kodeku, pouze uvádí doporučení, a parametry přenášeného audia většinou nejsou zobrazovány v rozhraní dostupném pro uživatele. Proto se může stát, že vlivem špatného nastavení parametrů kodeku může být jeho zvuková kvalita degradována. Někdy je tak kodek SBC neprávem označován jako nedostatečně kvalitní [12], [8].

4 Návrh a realizace BT-PCB

Nadcházející část diplomové práce se již zabývá návrhem a realizací BT modulu, přičemž zohledňuje poznatky uvedené v minulých kapitolách. Na následujícím řetězcovém schématu je ilustrován žádoucí princip funkce a přenosu signálu u konstruovaného zařízení BT-PCB.



Obr. 4.1 Blokové schéma toku audio signálu

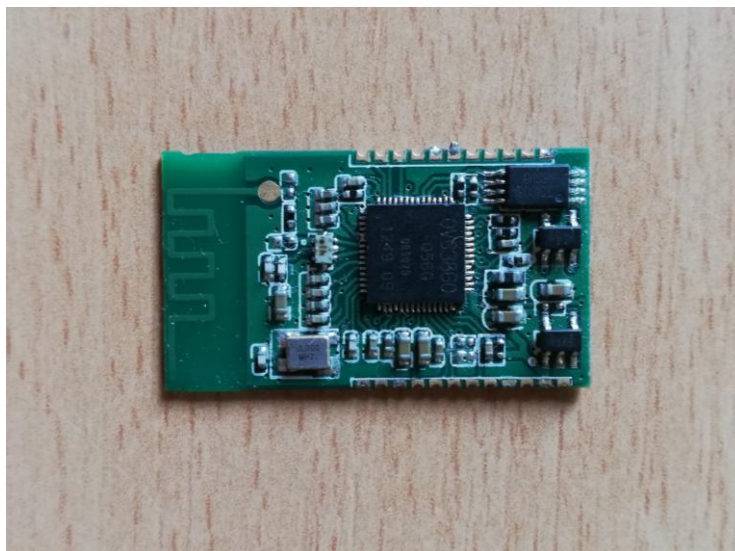
Audio signál je vysílán přes Bluetooth linku s využitím profilu A2DP od zařízení Android k přijímací části BT-PCB. V samotném modulu dochází k následnému dekódování komprimovaného audio signálu. Dále je pomocí DAC převodníku konvertován na analogový signál a nakonec patřičně zesílen, filtrován a přizpůsoben pro zátěž ve formě sluchátkových reproduktorů.

4.1 Modul XS3868

Pro první prototyp BT-PCB byl použit modul XS3868, vystavěný kolem procesoru OVC3860. Tento mikroprocesor je přímo navržen pro přenos stereo audio dat přes BT, pro použití v aplikacích jako sluchátka, reprosoustavy, mini-soundboxy atd. Podporuje verzi Bluetooth 2.0 + EDR a používá standardní SBC dekódování. Podporuje BT profily A2DP, AVRCP, HSP (headset profile) a HFP (hands-free profile), je tedy zároveň určen pro účely komunikace. Uvnitř tohoto čipu je integrovaný hi-fi audio kodek s dostatečně kvalitním (20bit stereo, 90db SNR) DAC převodníkem, o něco méně kvalitním (16bit mono) ADC převodníkem, integrovaným zesilovačem pro sluchátka (40mW @ 32ohm) a vstupem pro mikrofon [13].

Procesor je navržen s ohledem na nízkoodběrový provoz (1,8 V; 26mA při normální aktivitě). Zajímavostí u tohoto čipu je přítomnost kompletního power managementu, zahrnující spínaný regulátor, nabíjecí obvod pro baterie Li-ion / Li-pol, a Low Iq lineární regulátory. Není zde tedy nutná implementace externích regulátorů a modul je tedy možné

napájet přímo z článku Li-pol. Maximální provozní vzdálenost je podle specifikace do 10 metrů od vysílacího zařízení. Pro komunikaci používá rozhraní UART, skrze které lze systém konfigurovat, nebo k němu posílat AT příkazy [13].



Obr. 4.2 Modul XS3868

Modul XS3868 byl pro první prototyp BT-PCB vybrán zejména díky jeho relativně nízké ceně, snadné dostupnosti a celkem uspokojivým parametrům. Z praktických důvodů jsou zde také vyvedeny piny pro snadné ovládání přehrávání hudby přímo od přijímacího zařízení skrze podporu AVRCP (pozastavení, přetáčení, hlasitost). Dále má 2 vstupy pro indikační LED diody, které střídavě blikají, pokud modul není připojen k žádnému vysílacímu zařízení. Jakmile se připojí, bliká pouze jedna z nich. Modul má ještě několik dalších pinů, které naleznou využití při aplikacích s hlasovou komunikací, z časových důvodů jsem je ale hlouběji nezkoumal.

Nevýhodou u tohoto modelu je velmi nedostatečná dokumentace, a to jak samotného procesoru, tak i systému okolních obvodů. Modul je koncipován ve stylu „ready-made“ pro bezprostřední použití ve výše zmíněných audio aplikacích a není třeba jej pro tyto speciálně nastavovat. Není tedy příliš uzpůsoben k detailnímu programování. Množství AT příkazů je omezené a lze nastavit jen základní parametry.

V praxi tento modul neodpovídá požadavkům, které jsou na mou práci kladeny. Zaprvé kvalita přenosu signálu zde není příliš vysoká. To mohlo být způsobeno několika faktory. Modul XS3868 je vybaven BT verzí 2.0, zatímco na mém mobilu je již verze 4.2,

kteřá je sice zpětně kompatibilní, ale i tak tím patrně mohly být způsobeny některé chyby a inkonzistence v přenosu. Výstupní signál z modulu nebyl nijak upraven externími přizpůsobovacími nebo filtračními obvody, což jistě mohlo sehrát významnou roli. Tento problém je samozřejmě řešitelný, ale vzhledem k následujícímu bodu, byla další práce s tímto modulem přerušena.

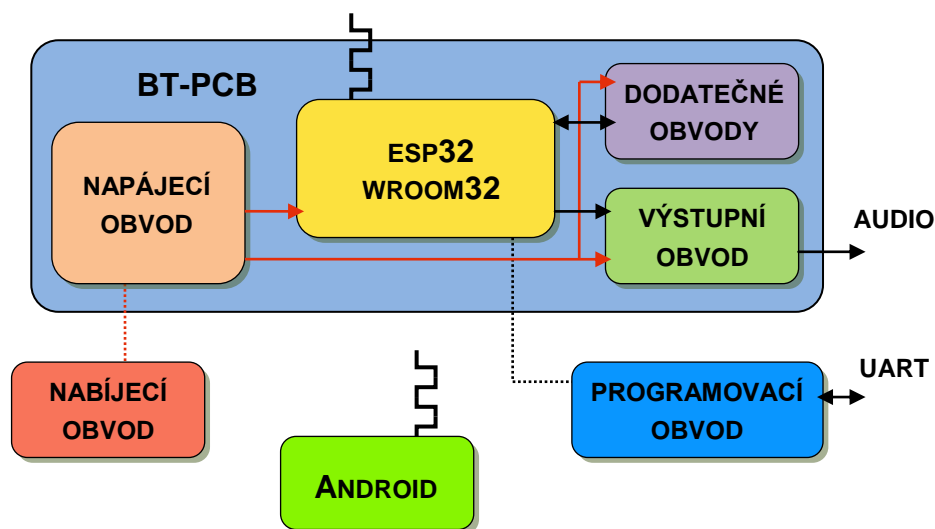
Kvůli nedostatečné dokumentaci vnitřní architektury a funkcí modulu se objevuje problém se zajištěním efektivní konfigurace a možností vlastního řízení modulu. Jestliže žádný z katalogových listů neposkytl detailnější informace k funkci modulu a procesoru, pak také nemohla být řádně vyvinuta mobilní aplikace, která by pomocí vyslaných konkrétních signálů tento systém řídila. Proto byl první prototyp odložen a byla započata práce na druhé verzi BT-PCB s využitím jiného modulu.

4.2 Modul ESP32

Druhá a finální verze zamýšleného BT-PCB je realizovaná za účasti modulu ze série ESP32 - WROOM32. Tento modul podporuje kromě klasického BT také Bluetooth Low Energy a WiFi. Je tedy vhodný mimo jiné i pro senzorické low-power aplikace, nebo pro připojení k internetu. Je však zároveň schopný plnit i náročnější úlohy, jako je právě například zpracování audiodat.

4.2.1 Návrh PCB

Funkční schéma BT-PCB se dá rozdělit do několika bloků. Jádro schématu tvoří samotný modul ESP32, který slouží jako přijímač BT signálu a zároveň v sobě integruje funkci dekódování komprimovaného signálu. Napájení celého systému je řešeno v napájecím obvodu. Pro možnost opětovného nabíjení baterie je zde popsán i nabíjecí obvod. Výstupní obvod slouží ke konverzi digitálního I2S signálu na analogový a k jeho zesílení, filtraci, případně přizpůsobení. Dodatečné obvody zahrnují „on-board“ dotykové základní ovládání pomocí MEMS mikrofону nebo kapacitních dotykových obvodů, napěťový dělič pro sledování napětí baterie a uživatelsky nastavitelnou LED. Pro možnost přeprogramování a manipulace s nahraným softwarem je vytvořen externí obvod pro komunikaci s počítačem přes rozhraní UART. Principiální blokové schéma BT-PCB je uvedeno na obrázku 4.3.



Obr. 4.3 Blokové schéma BT-PCB

- **Modul ESP32 - WROOM32**

Základ modulu WROOM32 tvoří dvoujádrový procesor ESP32-D0WDQ6. Obě jádra mohou být řízena individuálně a mají nastavitelnou frekvenci od 80 do 240 MHz. Tento procesor je díky přesným synchronizačním obvodům a podpoře několika low-power módů vhodný pro řadu mobilních, senzorických a IoT aplikací. Dále je vybaven potřebnými rádiovými komponenty, kalibračními obvody, zesilovačem výkonu, nízkošumovými zesilovači přijímaného signálu, filtračními obvody a tak dále [14].

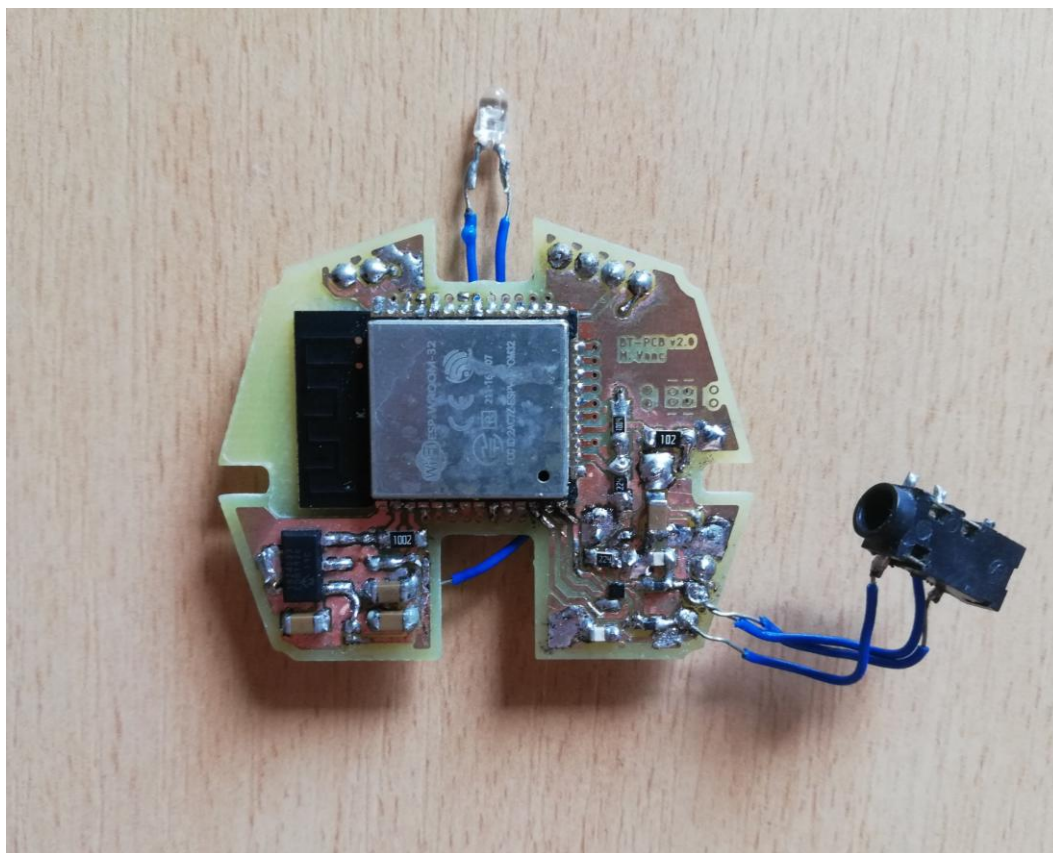
Modul v sobě integruje širokou škálu periférií jako jsou například kapacitní dotykové senzory, ADC, DAC, Hallova sonda, komunikační rozhraní pro SD karty, SPI, UART, I2C, SDIO, IR komunikace, GPIO, řízení PWM a pro audio aplikace také nezbytné rozhraní I2S. Modul podporuje rychlosti až do 150Mbps a využívá anténu s maximálním výkonem 20 dBm. Použitý operační systém pro ESP32 je freeRTOS s LwIP. Průměrná spotřeba modulu se pohybuje kolem 80 mA. Minimální proud dodaný z externího zdroje by měl být minimálně 500 mA. Pro low-power aplikace je možnost samotnou CPU odpojit a uživateli stačí pro monitorovací a vyhodnocovací funkce využít koprocesor. Proud v režimu spánku se může dostat až pod 5 μ A [14].

Podpora BT verze 4.2 BR/EDR verze u tohoto modulu zaručuje plnou kompatibilitu s

použitím pro tuto práci zvoleného zařízení Android, který taktéž podporuje BT verzi 4.2 BR/EDR. Mimo to modul obsahuje specifikaci BT Low Energy. Fyzická vrstva BT kontroléru je vybavena rádiovým vysílačem s podporou všech tří výkonostních tříd a rádiovým přijímačem typu NZIF s citlivostí -97 dBm. V rámci dostupného BT stacku je zde podporován audio kodek CSVD pro kódování hlasového audiosignálu a kodek SBC pro kódování audia vyšší kvality. Audio dekodéry jsou implementovány pouze softwareově v rámci Bluedroid stacku dostupném v knihovnách ESP-IDF [14].

Přesun dekódovaných audio dat do výstupního obvodu je realizován pomocí periferie I2S, která je implementována uvnitř modulu a může pro výstup využít libovolný pin GPIO. Data mohou být přesouvána v 16-bitovém, 24-bitovém nebo v 32-bitovém formátu.

Na obrázku 4.4 je vyfotografován výsledný produkt BT-PCB, který sestává z modulu ESP32, napájecího obvodu, výstupního obvodu s připojeným audio jack konektorem a dodatečnými obvody včetně vysunutých modrých LED diod.



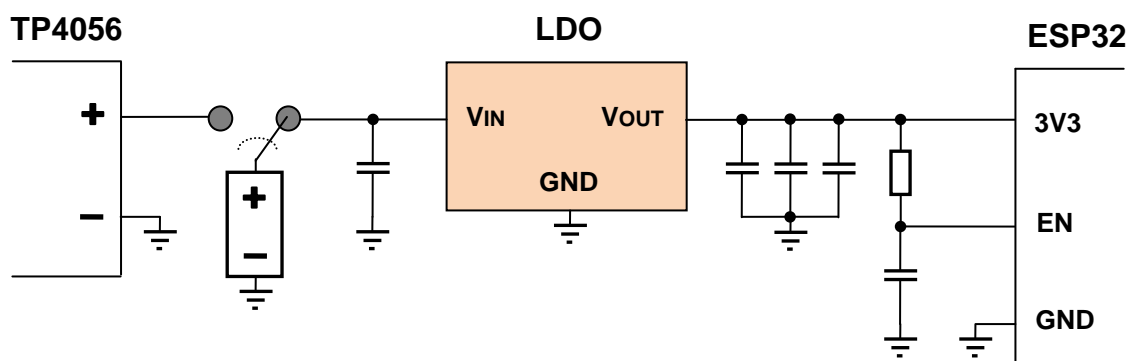
Obr. 4.4 Finální podoba BT-PCB s připojeným modulem ESP32

- **Napájecí a nabíjecí obvod**

Pro napájení modulu BT-PCB ve standardním poslechovém režimu je použit článek Li-pol s nominální hodnotu napětí 3,7 V. Napětí z článku, které po jeho plném nabití dosahuje hodnoty zhruba 4,2 V, je nejdříve pro snížení impedance zdroje připojeno přes kapacitor k zemní ploše a poté pomocí LDO regulátoru sníženo na 3,3 V. Konkrétní regulátor použitý v obvodu je dimenzován na vstupní napětí až 6 V a pracuje s napěťovým úbytkem 200 mV. Toto napětí je dále stabilizováno pomocí kapacitorů na výstupu LDO, přivedeno na pin 3V3 a přes 10k Ω rezistor na pin EN modulu ESP32.

V případě připojení modulu k počítači pro účely testování a nahrávání programu není potřeba napájení z bateriového článku. Vstupní napětí lze poskytnout přímo z konektoru USB v počítači skrze externí programovací obvod. Vstupní napěťová tolerance LDO regulátoru dovoluje snížení 5V napětí na potřebných 3,3 V.

Funkce nabíjení je realizována pomocí obvodu TP4056, který je speciálně navržen pro nabíjení lithium-ionových a lithium-polymerových článků. Na vstupu očekává 5 V a lze jej připojit ke zdroji napětí skrze konektor microUSB. Na výstupu potom poskytuje pevnou hodnotu napětí 4,2 V, která odpovídá hodnotě plného nabití bateriového článku. V režimu OFF/CHARGE je článek připojen k nabíjecímu obvodu a v závislosti na přítomnosti externího zdroje je nebo není nabíjen. V režimu ON je článek připojen k zátěži a poskytuje napájení dalším obvodům modulu BT-PCB.



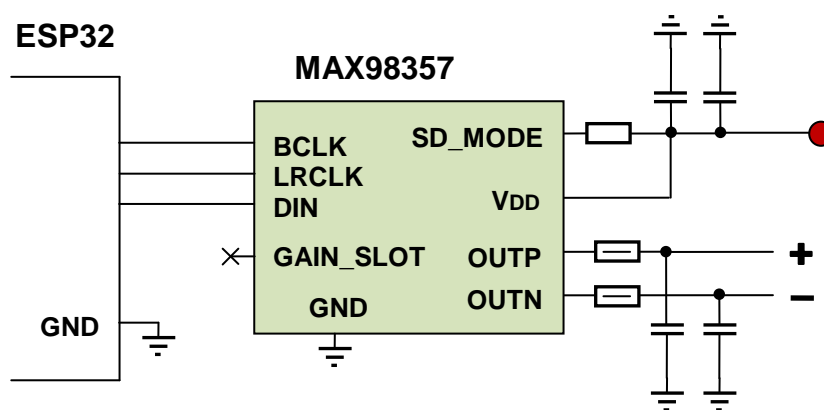
Obr. 4.5 Schéma nabíjecího a napájecího obvodu

- **Výstupní obvod**

Výstupní obvod slouží ke konverzi I2S digitálního signálu vystupujícího z I2S periferie modulu ESP32 na analogový. Součástí obvodu je dále zesilovací a filtrační člen, případně zde mohou být další přizpůsobovací členy pro finální úpravu audio signálu.

Základem tohoto obvodu je integrovaný obvod MAX98357B. Jedná se o součástku, která v sobě integruje DAC převodník, zesilovač třídy D a digitální audio rozhraní s podporou pro digitální vstupní signály typu I2S MSB-justified. Není zde potřeba externího hodinového signálu MCLK. Je možno použít standardní vzorkovací kmitočty od 8 kHz do 96 kHz a 16-b, 24-b nebo 32-b rozlišení. Má vysokou toleranci pro širokopásmový jitter na synchronizačních signálech BCLK a LRCLK. Na výstupu je schopen dodat výkon 1,8 W do zátěže o hodnotě 8 Ω s 10% celkovým harmonickým zkreslením při napájení 5 V. Je zde také možnost „standby“ módu pro nižší spotřebu [15].

Pro napájení je použit pin VCC, který je schopen přijmout hodnoty napětí od 2,5 V po 5,5 V. Napájecí napětí je na vstupu do obvodu stabilizováno dvěma kapacitami s hodnotami 10 μ F a 0,1 μ F. Obvod má vstup pro tři digitální signály I2S formátu - synchronizační BCLK a LRCLK a datový signál DIN. Bohužel výstup je pouze jednokanálový, kdy připojením pinu SDMODE přes různou rezistanci k napájecímu napětí lze zvolit pravý nebo levý kanál, případně lze stereo signál obsažený v I2S smíchat půl na půl. Pro stereo poslech by bylo třeba dvou těchto čipů nebo použít dvoukanálovou variantu obvodu. Pin GAIN slouží k volbě zesílení - pokud je ponechán nepřipojený, pak je výstupní signál zesílen o 9 dB [15].



Obr. 4.6 Schéma výstupního obvodu

Další prvky ve výstupním obvodu jsou filtrační členy na výstupních svorkách realizované pomocí feritových kroužků a 220pF kapacitorů zapojených paralelně k zemní ploše. Toto zapojení tvoří dolní propust' pro eliminaci nežádoucího rádiového rušení na audio linkách.

V souvislosti s výstupním audio signálem jsou pro úplnost jsou dále uvedeny parametry použitých sluchátkových reproduktorů. Jejich citlivost dosahuje 99dB SPL, což je u sluchátek zhruba průměrná hodnota. Tento parametr v podstatě udává efektivitu proměny elektrického výkonu v akustický tlak a je relevantní vždy pouze ve vztahu k použitému zesilovači. Výrobce dále udává frekvenční rozsah zahrnující kmitočty od 20 Hz do 20 kHz. Hodnota impedance činí zhruba 39 Ω , ale je samozřejmě frekvenčně závislá.

- ***Dodatečné obvody***

Pro funkci zjišťování hodnoty napětí na bateriovém článku a tedy zjišťování procenta nabití je mezi kladným a záporným pólem článku vytvořen napěťový dělič. Je realizován pomocí dvou rezistorů s hodnotami 1M Ω a 220 k Ω a jeho střed je přiveden na pin GPIO, který by měl být softwareově nastaven jako vstup pro ADC převodník. Poměr hodnot rezistorů je volen s ohledem na maximální rozsah vstupního napětí převodníku. Proud protékající děličem je v průměru roven zanedbatelné hodnotě zhruba 3 μ A.

Do návrhu plošného spoje je dále zařazen mikrofon pro ovládání poklepáním, ale v závislosti na použitém kódu by mohl sloužit i pro hlasový vstup. Pro ovládání dotykem by mohly sloužit i kapacitní dotykové senzory integrované v modulu ESP32. Součástí BT-PCB je i uživatelsky nastavitelná modrá LED dioda, která může sloužit pro různé indikační účely.

- ***Externí programovací obvod***

Po připojení k BT-PCB slouží tento obvod k nahrání programu z počítače do procesoru ESP32 přes sériové rozhraní UART. Obvod využívá převodník CP2102, který zajišťuje převod mezi počítačovým rozhraním USB a rozhraním UART obsaženým v modulu ESP32. Dále obsahuje obvodovou část se dvěma tranzistory NPN, která obstarává logiku pro korektní nahrávání programu. Je zde přítomné tlačítko napojené na pin EN,

tisknuté pro hardwareový reset modulu ESP32, a tlačítko BOOT tisknuté za účelem nahrání programu.

Z praktických důvodů a pro zaručení vyšší kompaktnosti byl modul navržen tak, že tento programovací obvod k němu lze připojit až po otevření krytu sluchátka. Program se typicky do modulu ESP32 nahrává teprve poté, co je celý modul BT-PCB vyjmut z krytu sluchátka. Vstupní napětí může být připojeno skrze tento programovací obvod.

4.2.2 Software

Pro efektivní práci s ESP32 modulem je nejprve nutné nastavit si vývojové prostředí. To zahrnuje instalaci sady speciálních programovacích nástrojů (Git, cross-compiler, CMake build tool, Python) a dalších nástrojů potřebných pro účely kompilace, sestavení a konfigurace programů. Poté je třeba stáhnout ESP-nativní framework ESP-IDF (Espressif IoT Development Framework), který obsahuje dostupné knihovny a příklady kódu psané v jazyce C, využitelné při vývoji aplikací u modulů ESP32. Dále je užitečné nastavit si v systému environmentální proměnné - cesty pro přístup ke všem těmto programovacím nástrojům. Posledním krokem je instalace softwareových balíčků programovacího jazyka Python, které jsou nutné pro manipulaci v rámci prostředí ESP-IDF.

V této fázi je již možné nahrát kód do modulu přímo přes příkazový řádek s využitím patřičných nástrojů. Před procesem kompilace je zde možnost využití konfiguračního nástroje s vlastním grafickým uživatelským rozhraním, který umožňuje počáteční nastavení vestavěných komponent a periférií modulu. Po tomto nastavení přichází na řadu kompilace a sestavení konkrétního kódu. Zde je mimo jiné využit nástroj CMake Build tool, protože je nutné k samotnému kódu přidat speciální soubor MakeFile, ve kterém je explicitně uvedeno, které soubory a knihovny se budou v rámci kódu kompilovat. Pokud nedojde při kompilaci k chybě, je možné konkrétní kód nahrát přímo do flash paměti modulu.

Je samozřejmě možné takto postupovat a opakovaně kompilovat změny v kódu prostřednictvím příkazového řádku, pro tyto účely je však vhodné implementovat ESP-IDF a související nástroje do libovolného integrovaného vývojového prostředí, kde je práce s kódem výrazně snazší, pohodlnější a hlavně přehlednější. Pro tento účel jsem využil vývojové prostředí Visual Studio Code. Každý projekt je přehledně umístěn do samostatné

složky, kde jsou přítomny uživatelské zdrojové kódy, výše zmíněné CMake soubory, konfigurační soubor a separátní složka se soubory vzniklými po kompilaci. Po otevření projektu ve Visual Studio Code zde vznikne také složka `.vscode`, kam se ukládá zvláštní nastavení prostředí platné pouze pro daný projekt. Pro efektivitu při nahrávání kódu byly v této složce vytvořeny takzvané tasky v souboru typu `json`, které fungují jako zkratky jednotlivých příkazů - tedy `build` a `flash`.

- ***Schéma přenosu dat***

Aplikace by měla s využitím speciálních BT profilů zajistit kontinuální přenos audio dat do modulu a na výstup. Dále by měla umožnit základní audio ovládání, jak ze strany zařízení Android, tak ze strany modulu ESP32.

V první fázi jsou data vysílána ze zařízení Android v komprimovaném formátu rádiovým kanálem. Pokud jsou audio soubory ve ztrátovém formátu uloženy v zařízení, tak v případě, že pro tento formát není v systému podpora enkódování, musí být soubory transkódovány do výchozího SBC formátu. Po přijetí na straně modulu ESP32 musí být komprimovaná data dekodována. Toto dekodování probíhá softwareově, to znamená, že zde není přítomný žádný fyzický audio kodek. Program pro dekodování je uložen v knihovněch `Bluedroid` stacku, který je součástí `ESP-IDF`.

Dekodovaná data jsou rozhraním I2S transportována na vyvedené piny modulu, odkud putují do výstupního obvodu. Zde jsou digitální data převedena do analogové podoby, zesílena a patřičným způsobem filtrována.

- ***Využití profilu A2DP v aplikaci***

V samotném programovém kódu je využito několik BT profilů. Za prvé pro přenos audiodat ze zařízení Android je využit profil A2DP v režimu přijímání. V kódu je třeba předem inicializovat pevnou paměť pro umístění kalibračních dat pro konkrétní fyzickou vrstvu. Dále je nutná správná konfigurace periferie I2S, přičemž se zde volí piny pro výstupní signál v I2S formátu (`LRCLK`, `BCLK` a `DATA`). Poté proběhne konfigurace řadiče BT pro běh v režimu BR/EDR a zároveň dojde k nastavení pro komunikaci se zařízením Android. Po inicializační fázi je vytvořena aplikační úloha, která v rámci

aplikační vrstvy vyřizuje a zpracovává různé eventy vznikající při komunikaci mezi oběma zařízeními. Pro tento účel využívá paměťovou frontu vyhrazenou pro 10 příchozích eventů. Dále je třeba nastavení jména BT zařízení, registrace funkcí zpětného volání pro profily GAP a A2DP, nastavení viditelného a připojitelného módu a režim zabezpečení párování.

Pro přídatné ovládací funkce jsou využity profily SPP a AVRCP. Modul ESP32 je v rámci profilu SPP nastaven do režimu „acceptor“, čili je potřeba, aby iniciace komunikace proběhla ze strany vysílacího zařízení Android. Je možné nastavit modul tak, aby při přijetí konkrétního znaku vykonal určitou akci, například přepnul logickou úroveň na některém z GPIO. Tato akce však může obnášet i zpětnou komunikaci s vysílacím zařízením, modul tedy může odpovědět skrze stejný sériový port na určitý „dotaz“ od Androidu. Profil AVRCP je přímo uzpůsoben pro komunikaci s audio nebo video zařízením a ve své specifikaci obsahuje řadu základních příkazů, jako play, pause, next, previous, volume, atd.

Scénář použití profilu AVRCP konkrétně pro mou práci je následující. Obě zařízení by měly být schopny pracovat jako kontrolér i jako cíl. Mobil bude ve výchozím stavu kontrolér, který má k dispozici všechny standardní příkazy. Sluchátka bych chtěl aby mohly posílat minimálně příkaz play pause. ovládací rozhraní by mělo být realizováno pomocí MEMS-mikrofonního senzoru.

Profil SPP je využit pro získání informace o napětové hodnotě baterie. Mikrokontrolér čeká na specifický znak vyslaný po emulované sériové lince a jakmile ho obdrží, provede jednorázové měření s využitím ADC převodníku přítomném na modulu. Aplikace by také mohla fungovat tak, že stav baterie by byl monitorován kontinuálně, např. jednou za několik minut. V případě, že by procento nabití baterie kleslo pod uživatelsky stanovenou úroveň, by mohl mikrokontrolér do zařízení Android odeslat speciální znak, po jehož přijetí by telefon krátce zavibroval v definovaném „patternu“.

- ***MicroPython***

Pro zajímavost zde také uvádím možnost programovat modul ESP32 v jazyce MicroPython. Tento programovací jazyk je kompletní reimplementací jazyka Python 3, odkud si vypůjčuje malou část standardní knihovny, a je optimalizovaný pro chod na

mikrokontrolérech a jiných izolovaných systémech [16][16]. Před spuštěním kódu je nutno nejprve do mikroprocesoru nahrát speciální firmware určený pro běh s použitím MicroPythonu.

MicroPython je narozdíl od jazyka C objektově orientovaný a před nahráním kódu do paměti mikrokontroléru se zdrojový kód nemusí předem kompilovat do strojového kódu. Místo toho je zde použit speciální program, tzv. interpret, který zdrojový kód v reálném čase a přímo „interpretuje“. Ladění programu se tak stává daleko flexibilnějším než v případě kompilačních jazyků jako je C. Dále má tento jazyk k dispozici velké množství zabudovaných funkcí pro rychlé použití v kódu [16].

Tato alternativa tedy umožňuje psát kód pro modul ESP32 v jazyce vyšší úrovně, což je relativně nové a neobvyklé řešení a ještě více se tak zpohodlnňuje práce při vyvíjení vlastních aplikací. Problém však představuje prozatímní nezastoupenost některých „céčkovských“ funkcí a knihoven. Například pro práci s komponentou Bluetooth BR/EDR zatím v oblasti jazyka MicroPython nebyly vytvořeny žádné ovladače. Tuto variantu jsem tedy musel vzhledem k podstatě mé práce zavrhnout.

4.2.3 Aplikace pro zařízení Android

Prototyp mobilní aplikace je vytvořen pomocí online nástroje AppInventor. Je zde využit modul BT Client pro snadné použití BT sériového portu. Funkcí několika tlačítek, co jsou použity v aplikaci je vyslat znak konkrétní znak po sériovém portu, kdy pro některé znaky je očekávána odpověď od modulu, která se pak zobrazí v přítomném textovém poli.

Stávající verze aplikace slouží pro ověření správné funkcionality při BT komunikaci a pro výše zmíněné zjištění úroveň nabití baterie. Na obrázku 4.5 je screenshot domovské obrazovky aplikace. Grafické rozhraní je v této rané verzi realizováno spíše provizorně, důraz je zde kladen především na správnou funkcionality aplikace v součinnosti s modulem BT-PCB.



Obr. 4.7 Grafické rozhraní mobilní aplikace

V budoucnosti a za použití vyspělejších nástrojů by mohla aplikace zároveň sloužit jako vlastnoručně navržená hudební knihovna s možností ovládání pomocí AVRCP funkcí a s přídatnými SPP funkcemi.

5 Závěr

V rámci této diplomové práce byly nejprve předneseny výhody a nevýhody bezdrátového přenosu. Dále byl detailně popsán princip bezdrátové komunikační technologie Bluetooth a rozebrány jednotlivé protokoly, které jsou jádrem této technologie. V návaznosti zde byly uvedeny příslušné BT profily použité při konkrétní aplikaci pro tuto diplomovou práci. Nakonec byla ve stručnosti uvedena problematika kodeků se zvláštní kapitolou pro kodek SBC, který byl v konkrétní aplikaci použit.

Pro účely diplomové práce byla dále navržena a realizována deska plošného spoje s funkcí bezdrátového audio přijímače. V jejím základu byl použit modul od společnosti Espressif - ESP32 podporující technologie Bluetooth Classic, Bluetooth Low Energy, WiFi. První prototyp desky plošného spoje obsahoval modul XS3868, který se ale vzhledem k požadavkům této práce neosvědčil. Dalším postupem byl návrh dílčích obvodů, tedy nabíjecího a napájecího obvodu, výstupního zesilovacího obvodu a pomocných řídicích nebo indikačních obvodů. Následně byl třeba naprogramovat použitý mikrokontrolér ESP32 pro danou aplikaci, tj. zajistit správný průtok audio dat a v rámci programu spřáhnout ovládací funkce a dotazovací funkce mezi modulem a vlastní navrženou aplikací pro zařízení Android.

V průběhu řešení tohoto projektu se objevily některé problémy, související zejména s výběrem vhodných součástí a komponent. Například nebylo lehké vybrat samotný BT modul, který by splňoval nároky pro zpracování audia vyšší kvality, přitom byl uživatelsky programovatelný a konfigurovatelný, zároveň i cenově dostupný. Pro co nejtěsnější připojení modulu k vybraným sluchátkům byl výsledný modul navržen s ohledem na celkovou velikost a tvar. Pro připojení do vnitřku krytu sluchátka zde byly také nutné fyzické zásahy pro zvětšení vnitřního prostoru.

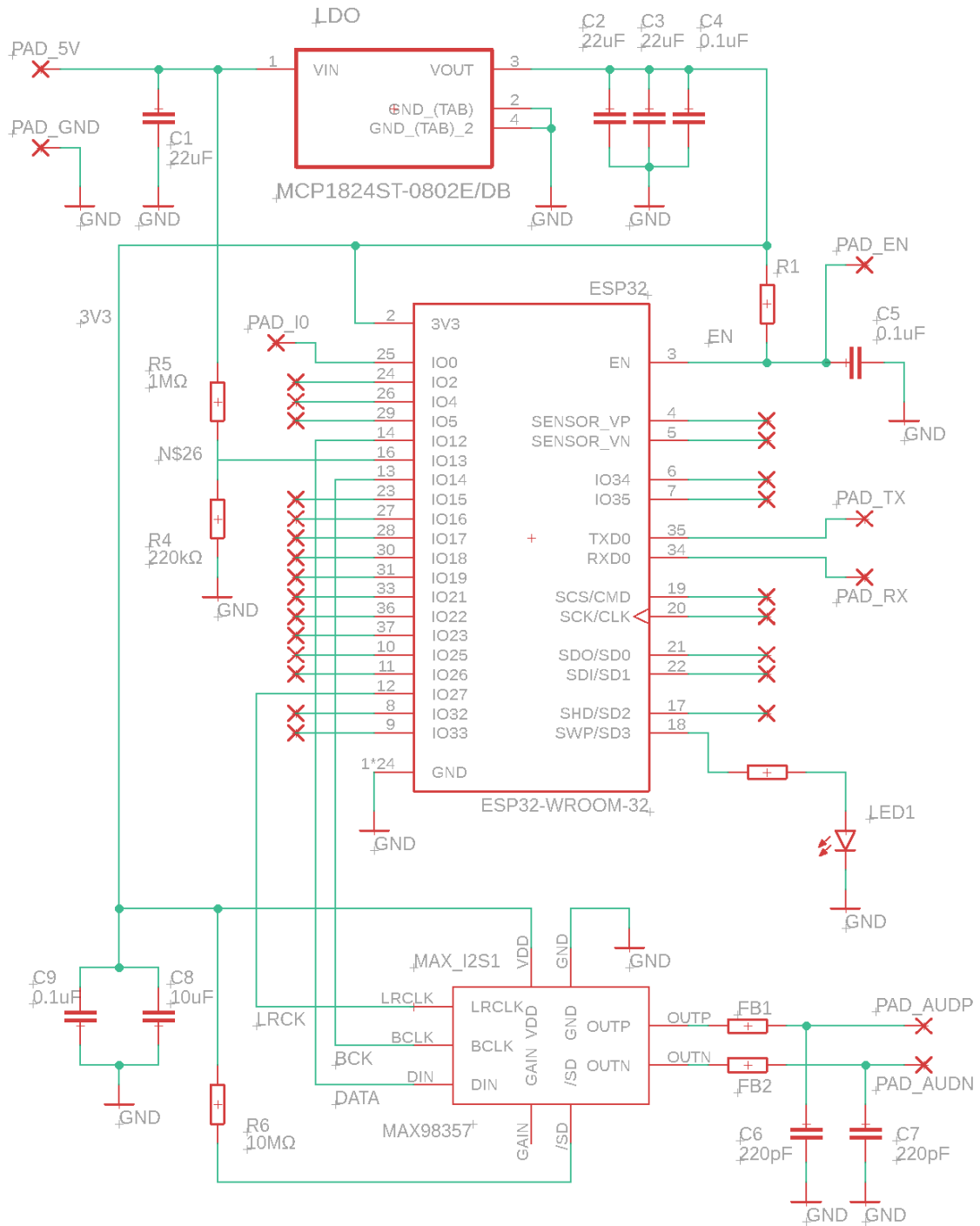
Výsledkem práce je funkční prototyp BT audio přijímače s dostatečnou zvukovou kvalitou a aplikace, která tento modul dokáže patřičným způsobem ovládat.

Seznam literatury a informačních zdrojů

- [1] J. Strickland, “How Wireless Speakers Work.” [Online]. Available: <https://electronics.howstuffworks.com/gadgets/audio-music/wireless-speakers5.htm>.
- [2] Victrola, “Wireless vs Wired Headphones.” [Online]. Available: <https://victrola.com/blogs/articles/wireless-vs-wired-headphones-which-has-the-best-sound-quality>.
- [3] R. Triggs, “A quick history of Bluetooth,” *Android Authority*, 2018. [Online]. Available: <https://www.androidauthority.com/history-bluetooth-explained-846345/>.
- [4] Jimblom, “Bluetooth Basics.” [Online]. Available: <https://learn.sparkfun.com/tutorials/bluetooth-basics>.
- [5] Bluetooth SIG, “Radio Versions,” 2020. [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/>.
- [6] Bluetooth SIG, “Bluetooth Market Update 2020,” 2020.
- [7] Bluetooth SIG, *Specification of the Bluetooth System Version 4.2*. 2014.
- [8] Bluetooth SIG, *Advanced Audio Distribution*, V1.3.2. 2019.
- [9] Bluetooth SIG, *Audio/Video Remote Control*, V1.6.2. 2019.
- [10] Bluetooth SIG, *Serial Port Profile*, v12 ed. 2012.
- [11] R. Triggs, “LE Audio LC3 Explained.” [Online]. Available: <https://www.soundguys.com/bluetooth-le-audio-lc3-explained-28192/>.
- [12] ValdikSS, “Audio over Bluetooth,” 2019. [Online]. Available: <https://habr.com/en/post/456182/>.
- [13] Omnivision Technologies Inc., *OVC3860*. 2010.
- [14] Espressif Inc., *ESP32-WROOM-32 Datasheet*. 2018.
- [15] Maxim Integrated Products Inc., *MAX98357A-MAX98357B*. 2019.
- [16] C. Bell, *MicroPython for the Internet of Things*. 2017.

Přílohy

Příloha A - Elektrické schéma modulu BT-PCB



Příloha B - Návrh desky plošného spoje pro modul BT-PCB

