VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

ANALÝZA LORAWAN PROVOZU POMOCÍ SDR

LORAWAN TRAFFIC ANALYSIS VIA SDR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MICHAL JÁL**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. ONDŘEJ HUJŇÁK**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Jál Michal**
Program: Informační technologie
Název: **Analýza LoRaWAN provozu pomocí SDR**
LoRaWAN Traffic Analysis via SDR
Kategorie: Počítačové sítě

Zadání:

1. Nastudujte problematiku softwarově definovaných rádií.
2. Seznamte se s technologií LoRaWAN se zaměřením na spojení s koncovými zařízeními a jeho radiové vlastnosti.
3. Navrhněte implementaci LoRaWAN snifferu pomocí SDR rádia LimeSDR.
4. Navržené řešení implementujte a s jeho pomocí analyzujte komunikaci vybraného LoRaWAN uzlu.
5. Zhodnoťte dosažené výsledky, definujte omezení Vašeho řešení a diskutujte možná použití a rozšíření.

Literatura:

- LoRaWAN 1.0.3 Specification, LoRa Alliance, 2018
- https://wiki.myriadrf.org/LimeSDR_Quick_Start
- KNIGHT, Matthew; SEEBER, Balint. Decoding LoRa: Realizing a modern LPWAN with SDR. In: *Proceedings of the GNU Radio Conference*. 2016.
- AREF, Mohamed; SIKORA, Axel. Free space range measurements with Semtech LoRa technology. In: *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*. IEEE, 2014. p. 19-23.
- ROBYNS, Pieter, et al. A multi-channel software decoder for the LoRa modulation scheme.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body ze zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Hujňák Ondřej, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 31. července 2020
Datum schválení: 31. října 2019

Abstrakt

Tato práce se zabývá implementací LoRaWAN snifferu za použití softwarové definovaného radia LimeSDR. A analýzou LoRaWAN komunikace za pomoci tohoto zařízení. Součástí práce je popis modulace LoRa a představení síťové vrstvy LoRaWAN.

Abstract

This id about implementing LoRaWAN sniffer using software-defined radio LimeSDR. And analyzing LoRaWAN communication with this device. Part of the work is a description of LoRa modulation and introduction of the LoRaWAN network layer.

Klíčová slova

Softwerově definované rádio, LoRa, LoRaWAN, IoT, Sniffer

Keywords

Software-defined radio, LoRa, LoRaWAN, IoT, Sniffer

Citace

JÁL, Michal. *Analýza LoRaWAN provozu pomocí SDR*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Hujňák

Analýza LoRaWAN provozu pomocí SDR

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana inženýra Ondřeje Hujňáka Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Michal Jál
31. července 2020

Obsah

Kapitola 1

Úvod

V této práci si uděláme lehký úvod do problematiky softwarově definovaných radí. Přičemž si vysvětlíme základní pojmy a především se pokusím rozumě vysvětlit - proč softwarově definovaná rádia používají k reprezentaci signálu takzvané IQ vzorky.

Pokusíme se pochopit principy LoRa modulace a vysvětli její výhody a nevýhody. Představíme si LoRaWAN síťovou vrstvu včetně typů zařízení a jež se v této síti vyskytují. Lehce se podíváme i na zabezpečení přenosu na těchto sítí.

Následně se zaměříme na hlavní část návrh LoRawan snifferu za použití LimeSDR. Důkladně projdeme výběr softwaru použitého k implementaci. Otestujeme základní předpoklady funkčnosti všech vybraných možností. Po výběru si ukážeme možnosti a výhody rozšiřujících rozhraní pro wireshark extcap. Za pomocí tohoto rozšíření implementujeme vybrané řešení.

Po té vyzkoušíme naše zařízení při odchytávání v sítí LoRaWAN. Provedeme analýzu zachycených dat a vyhodnotíme výsledky. V závěru pak již jen definujeme omezení naší implementace a pokusíme se navrhnout možná rozšíření či vylepšení.

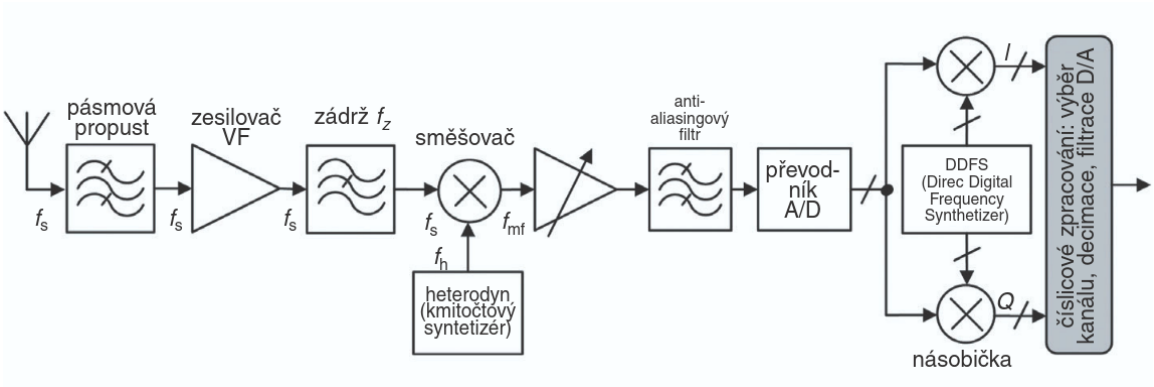
Kapitola 2

Úvod do: Softwarově definované rádia

Softwarově definované radio je systém pro zachytávaní a zpracování rádiových vln, ve kterém jsou jednotlivé části jako (směšovač, filtr, zesilovač, modulátor...) ne nutně všechny implementovány za pomocí softwaru. Výhodou tohoto řešení je možnost změn parametrů přijímaní bez nutnosti fyzických úprav hardwaru. Z pracování signálu probíhá poté digitálně za pomocí softwaru, který můžeme libovolně modifikovat.

2.1 Teoretický princip SDR

V zjednodušené podobě si můžeme představit ideální SDR pouze jako anténu připojenou k AD převodníku přes dolno propustní filtr, kterým odfiltrujeme signál po nejvyšší chtěné frekvence. V praxi by tato implementace kladla příliš vysoké nároky na AD převodník, který by musel disponovat dostatečným dynamickým rozsahem a rychlostí vzorkovaní. Představme si, že chceme za pomocí takového SDR přijímat běžné FM radiové vysílání. Rádio vysílá v pásmu 87,5 – 108 MHz. To znamená, že dle Shannonova teorému bychom museli vzorkovat minimálně rychlostí 216 Msps takový převodník by byl zbytečně drahý na výrobu. [3]



Obrázek 2.1: SDR se zpracováním na mezním kmitočtu [10]

2.2 Reálné typy implementace SDR

Nejčastějším typem softwarově definovaného radia je přijímač s digitálním zpracováním na mezní frekvenci. Tento typ vychází s superheterodyn radí kde demodulátor nahradíme AD převodníkem a kvadraturním syntezátorem. [10]

2.2.1 Základní součásti SDR a jejich funkce

Anténa

Anténa je zařízení určené k přijmu nebo vysílání rádiových signálu. Slouží k přeměně elektromagnetických vln na elektrickou energii v případě příjímání a nebo naopak k přeměně elektrické energie na elektromagnetické vlny pokud chceme vysílat.

Dělíme je na všesměrové a směrové. V všesměrové antény jsou určeny k získávání nebo vysílání signálu do všech směrů. V porovnání ze směrovými anténami které vrhají signál jen do určité výseče v prostoru.

Důležitým parametrem je zisk, ten je udávaný jednotkách dBi nebo dBd. Což je poměr zisku s ideální izotropní anténou dBi, nebo podíl proti ideálnímu dipólu dBd.[17]

Zesilovač

Slouží k zesílení signálu neboli k zvýšení amplitudy signálu.

Filtr

Umožňuje nám odstranit nechtěné části signálu. V tomto případě máme namysli lineární filtry a to především dolnopropustní filtr, který nám umožňuje utlumit vyšší frekvence jež nepotřebujeme.

Směšovač

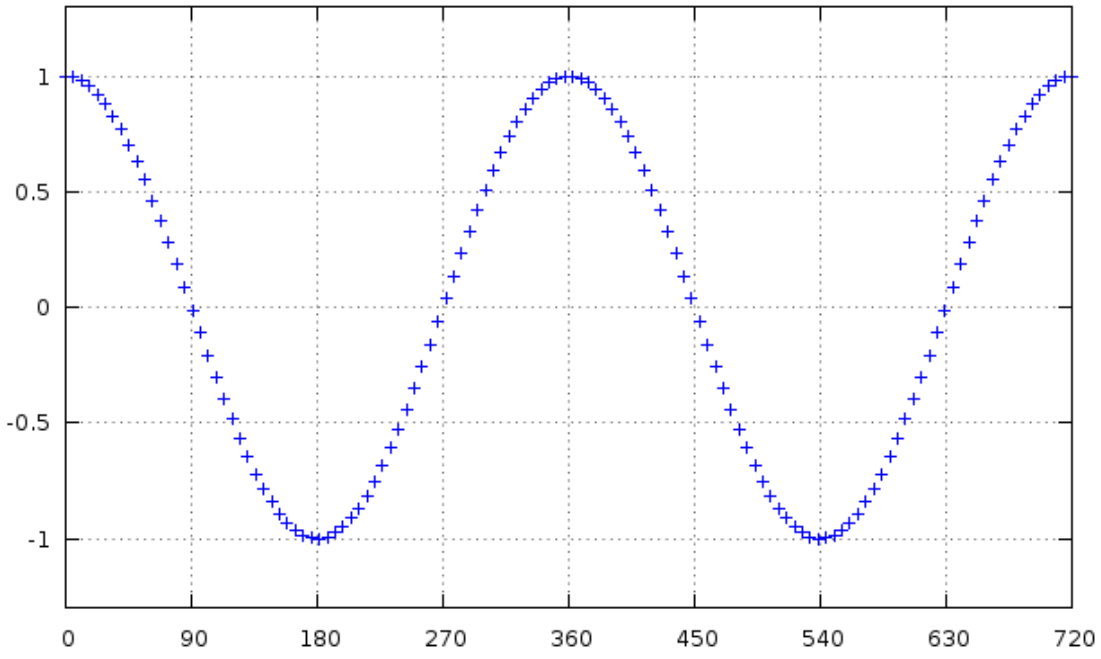
Slouží k posunu požadováného signálu na jinou frekvenci. Směšovač dostane na vstupu dva signály a na výstup nám dá součet nebo rozdíl těchto dvou signálů. Směšovač vracející součtovou frekvenci nazýváme up-convertor a v případě rozdílové frekvence down-convertor. V SDR potřebujeme posunout signál na nižší frekvence využívám tedy nejčastěji down-convertor. [3] Díky posunutí signálu ve frekvenci již nepotřebujeme extrémně rychlý převodník, ale stáčí nám převodník z dvojnásobnou vzorkovací frekvencí než je naše požadovaná šířka pásma.

AD převodník

AD převodník je součástka určená k převodu spojitého signálu na diskrétní. Převod spojitého signálu na diskrétní nazýváme vzorkování. Pokud nechceme ztratit informaci musíme signál vzorkovat s minimálně dvojnásobnou frekvencí než je nejvyšší frekvence, kterou chceme získat. Pro nás důležitými parametry převodníku je rychlost vzorkovaní a rozlišení. [16]

2.3 I/Q signály

Proto abychom mohly začít pracovat s SDR potřebujeme porozumět reprezentaci signálu signálu pomocí I/Q vzorků.



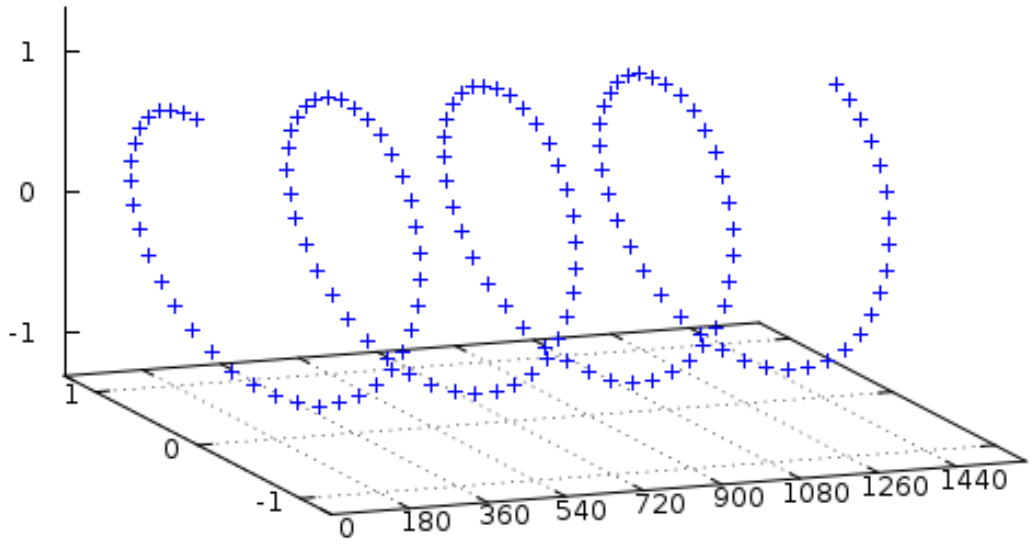
Obrázek 2.2: Funkce cosinus v 2D prostoru [1]

Proč ale vlastně potřebujeme nějaké I/Q data, nestačili by nám pouze vzorky amplitudy signálu. V čem je vlastně háček? Proč používat dvě množiny dat místo jedné.

Problém nastává s negativními frekvencemi. Jelikož přecházíme do základního pásma kdy náš požadovaný frekvenční rozsah převedeme středem k nule. Získáme tak záporné frekvence. Musíme se tedy nějak vypořádat se znázorněním této informace. Vyřešíme znázorněním funkce v prostoru. K tomuto účelu využijeme komplexní exponenciálu pomocí ní jsme schopni určit nejen amplitudu a frekvenci, ale jsme dokonce schopni rozlišit záporné frekvence od kladných. Nyní již potřebujeme jen v hodný způsob jak reprezentovat tuto komplexní exponenciálu v počítači. K čemuž se nám přímo nabízejí komplexní čísla. Jediný rozdíl je že reálnou část nazýváme In-Phase a imaginární složku Quadrature. Tyto signály jsou vzájemně posunuty o 90 stupňů (1/4 periody).

Druhým důvod proč je dobré použít reprezentaci pomocí I/Q signálu je, že neztrácíme informaci o tom v jaké úhlové části jsme vzorek pořídily. Získáme tak přesnější informace o maximální síle signálu.

Pokud se na obrázek kosínusovky v prostoru 2.3 podíváme zepředu uvidíme reálnou část signálu neboli I . Pohlédneme-li však na ni z vrchu uvidíme stejný signál posunutý ve fázy a to je naše Q kvadraturní část.[1]



Obrázek 2.3: Funkce cosinus v 3D prostoru - komplexní exponenciála [1]

2.4 LimeSDR

V této práci budeme pracovat s softwarově definovaným radiem LimeSDR USB. Jedná se o hardware vyráběný firmou Lime microsystems. Kromě této desky najdeme v nabídce tohoto výrobce další SDR například LimeSDR PCIe určeného pro pro implementaci do různých embedded zařízení. Pro toto rozhraní vyrábějí i vylepšenou verzi LimeSDR QPCIe s podporou 4x4 MIMO. V rámci krautfundingového projektu vznikla také levnější varianta LimeSDR MINI, která má pouze jeden chanel pro příjem a jeden pro vysílání.

2.4.1 LimeSDR USB

LimeSDR USB je cenově dostupné softwarově definované rádio vhodný především v aplikacích kde nedostačují parametry RTL-SDR a zároveň není třeba hardwaru, který se dá považovat za opravdu spolehlivé laboratorní zařízení jako USRP firmy ettus. Můžeme jej využít pro celou škálu různých aplikací v bezdrátových přenosů. [8]

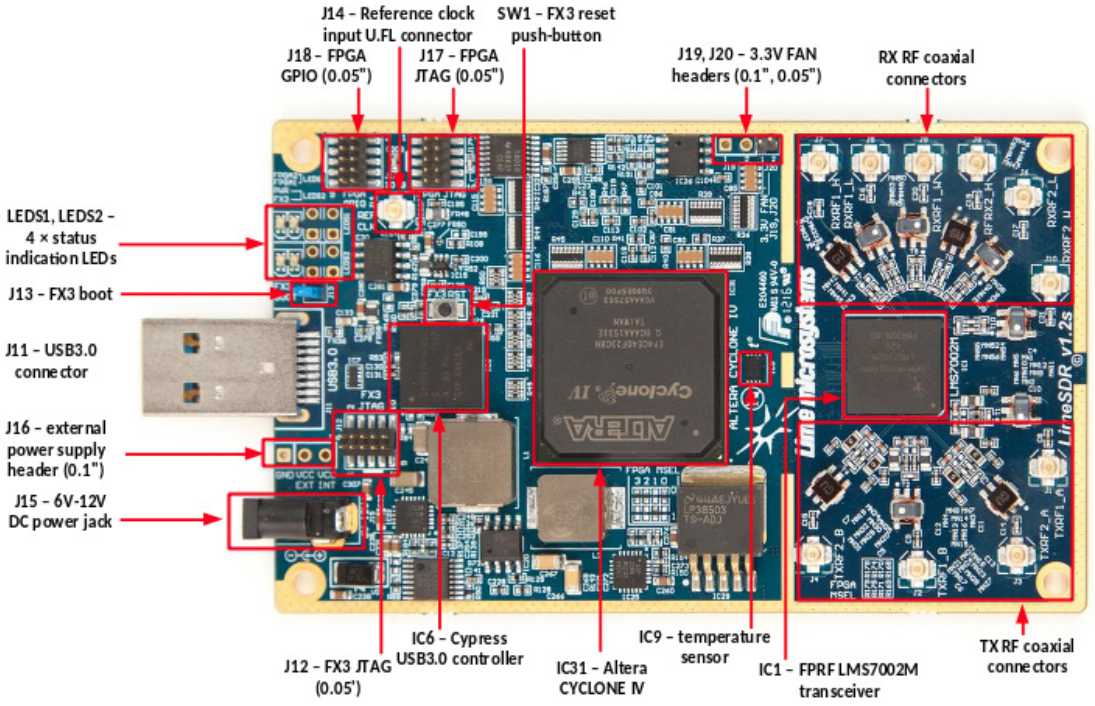
Možná využití

- Radar
- Základové stanice (BTS)
- Vysílání audio či audiovizuálního obsahu DBT, FM, AM
- Brány pro IoT
- HAM radio

- Odposlouchávání bezdrátových periferií či jejich emulace
- Sledování pozice letadle přes systém ADS-B
- Měření charakteristiky antén
- A mnoho dalších aplikací

Popis parametrů

LimeSDR 2.4 má dva přijímací a dva vysílací kanály, avšak není možné na dvou kanálech přijímat rozdílné frekvence dva kanály zároveň lze využít pouze v konfiguraci MIMO. Na desce nalezneme 10 U. FL konektorů. Čtyři sloužící pro vysílání jsou v párech pro kanál A a B. Zbylých šest slouží pro příjem na dvou kanálech a jsou rozděleny podle zapojeních zesilovačů na LNA L pro frekvence nižší než 1.5 GHz, LNA H pro vyšší jak 1.5 GHz a LNA W pro celý frekvenční rozsah. [9]



Obrázek 2.4: LimeSDR [9]

RF Transceiver	Lime Microsystems LMS7002M MIMO FPRF
FPGA	Altera Cyclone IV EP4CE40F23 – also compatible with EP4CE30F23
USB 3.0 kontroler	Cypress USB 3.0 CYUSB3014-BZXC
Oscilátor	Rakon RPT7050A @30.72MHz
Frekvenční rozsah	100 kHz – 3.8 GHz
Šířka pásma	61.44MHz
RF konecktry	10 U.FL konektorů (6 RX, 4 TX)
Výstupní výkon (CW)	Up to 10dBm
Multiplexing	2x2 MIMO
Zdroj napětí	Z USB nebo pomocí externího zdroje
Status indicators	Programovatelné LED
Rozměry	100mm x 60mm

Kapitola 3

LoRA & LoRaWAN

3.1 LoRa

LoRa je technologie pro komunikaci na velkou vzdálenost. Její název vznikl zkrácením slov Long Rang. Byla navržena za účelem komunikace zařízení s nízkou spotřebou na velké vzdálenosti s odolností vůči rušení. Odolnost vůči rušení je zvláště důležitá jelikož LoRa podobně jako jiné LPWAN technologie je provozována v bezlicenčních pásmech.

LoRa je tedy proprietární protokol pro modulaci signálu založený na chirp spread spectrum modulaci (CSS). Za návrhem stojí společnost Semtech, která je současně jediným oficiálním výrobcem čipů pro LoRa modulaci a demodulaci.

3.1.1 Princip a parametry LoRa Modulace

Princip modulace spočívá na základě generování Chirp signálu. Chirp signál lineárně zvyšuje nebo snižuje frekvenci než dosáhne definované hodnoty. V našem případě se jedná o zvolenou šířku pásma. Chirp s rostoucí frekvencí označujeme jako upchirp a s klesající frekvencí downchirp. Tento typ modulace řadíme mezi systémy s rozprostřeným spektrem. Rozprostření dat a jejich namodulování na nosný signál zajišťuje odolnost vůči slábnutí signálu. Odolnost vůči slábnutí signálu můžeme ovlivnit nastavením právě Spreding Factoru. Dále charakterizujeme LoRa modulaci pomocí Bandwith a Coding Rate. [4]

Spreding Factor (SF)

Spreding Factor určuje kolik bitů je zakódováno v jednom chirpu/symbolu, ale také říká kolik chipů je v jednom symbolu. V každém symbolu je tedy 2^{SF} chipů. Pro jednoduchost si můžeme představit jeden chip tak, že vezmeme jeden Chirp a rozdělíme jej na 2^{SF} dílků. Každý dílek nám pak reprezentuje změnu frekvence v dané šířce pásma. Kódování probíhá tak, že začneme generovat chirp od určitého dílku, který reprezentuje požadovanou hodnotu a frekvenci zvyšujeme až než dosáhneme maxima poté provedeme skok na nejnižší frekvenci a pokračujeme se zvyšováním až po hodnotu na které jsme začali. LoRa má celkem 7 možností konfigurace spreding factoru 6 až 12. Jeden symbol pak nese 6 až 12 bitů informace. Zvolený hodnota spreding factoru ovlivňuje schopnost dekódovat slabší signály. Čím vyšší je tím jednodušší je dekódovat signál. Cenou za zvýšení počtu chipů na symbol je snížení datové rychlosti.

Bandwidth (BW)

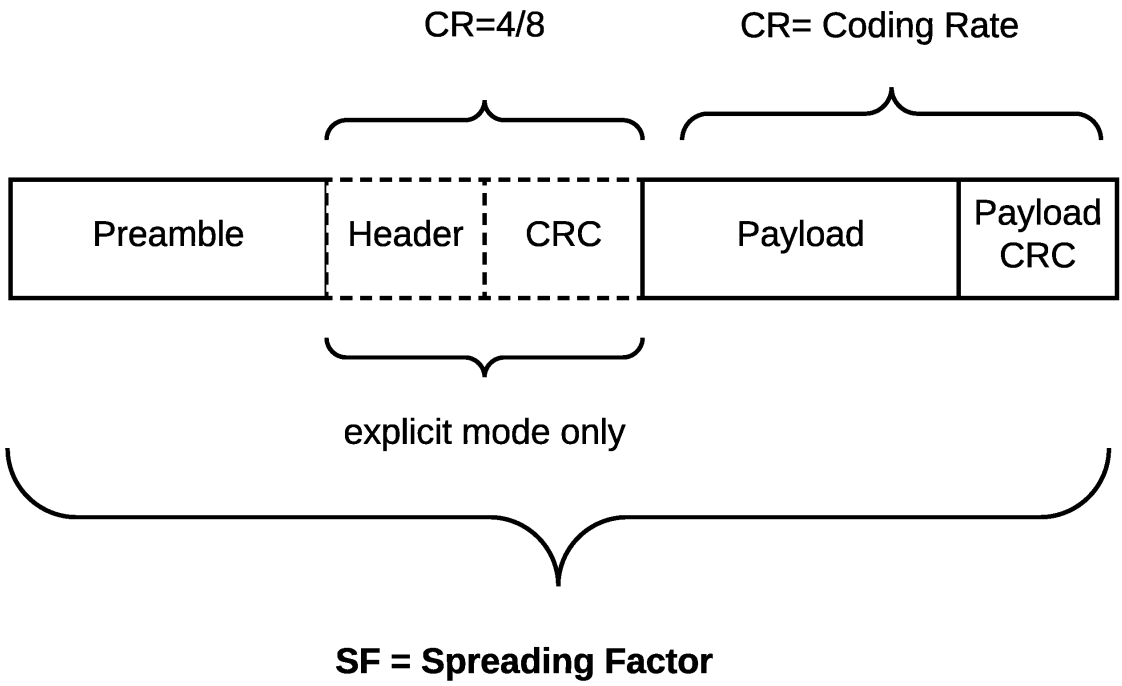
Neboli šířka pásma, je rozdíl mezi nejvyšší a nejnižší frekvencí v přenášeném signálu. Má vliv na datovou rychlost a obsazenost spektra. Při volení této hodnoty se musíme řídit normami definujícími parametry vysílání v požadovaném frekvenčním rozsahu. V našich podmínkách se o dodržování stará Český telekomunikační úřad. Provoz je tedy možný pod všeobecním opravněním VO-R/10/01.2019-1. Šířka pásma je konfigurovatelná z předdefinovaných hodnot v rozsahu 125 kHz až 500 kHz pro moduly SX1272/73 a v rozsahu 7,8 kHz až 500 kHz pro SX1276/77/78.

Code Rate (CR)

Lora moduly také umožňují využiti samoopravného kódu. Code rate udává rozšíření délky dat oproti jejich původní délce. Toto rozšíření nám dává možnost detekovat a opravit chyby vzniklé při přenosu. LoRa umožnuje kodov= poměry: 4/8, 4/7, 4/6, 4/5.

3.1.2 Struktura LoRa rámce

Každý LoRa rámec začíná preambulí, následovaný hlavičkou a uživatelskými daty. Hlavička se vyskytuje pouze v explicitním modu. V implicitním modu jsou parametry kodování předem známe tudíž hlavička není potřeba. [14] [15]



Obrázek 3.1: LoRa - rámec

Preambule

Preambule slouží především k detekci LoRa rámce přijímačem. Její délku můžeme ovlivnit. Pokud délku preambule nastavíme na nulu dostaneme preambuli obsahující dva upchirpy

se zakodovaným synchronizačním slovem následované 2.25 downchirpy. Zvětšením délky hlavičky přidáme na začátek tolik upchirpů kolik jsme nastavily v registru délky preambule. Preambule je tedy vždy dlouhá n + 4.25 chirpu/symbolu.

Synchronizační slovo tak je definováno v datasheetu SX1276. Slouží k rozlišení typu sítě. Hodnota může být nastavena na 0x00 pro obecný přenos v s modulací LoRa, 0x12 pro privátní sítě LoRaWAN a nebo hodnota 0x34 pro veřejné LoRaWAN sítě poskytované lokálním operátorem. Slouží k tomu aby se koncové brány a koncová zařízení nemuseli zabývat dekódováním zpráv, které nejsou pro ně určené.

Hlavička rámce

V hlavičce jsou obsaženy informace o délce uživatelských dat (Payloadu), přítomnost CRC a nastavení coding rate pro zbytek zprávy. Hlavička samotná je vždy kodovaná s coding rate 4/8 zdůvodů co nejvyšší odolnosti vůči vzniklým chybám. Hlavička není povinnou součástí rámce. Proto rozlišujeme dva módy implicitní (hlavička není přítomna) a explicitní (hlavička je obsažena v rámci). V případě implicitního módu je zbytek zprávy kódován s coding ratem předem známým všem zařízením pro které je zpráva určena.

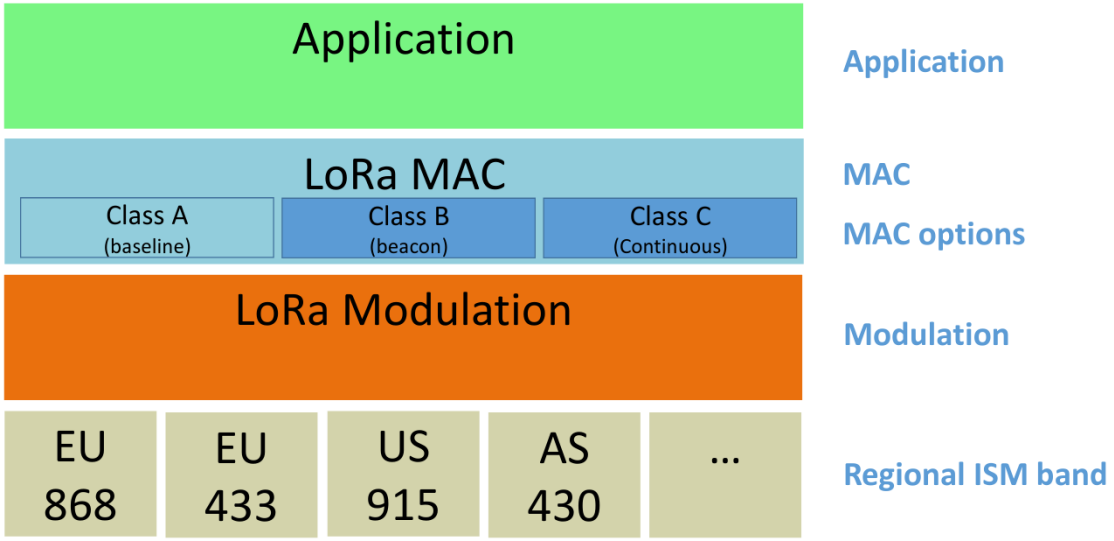
Payload

Obsahuje užitečná přenášená data. V našem případě zde očekáváme vyšší vrstvu LoRaWAN. Zakódovaná pomocí coding rate definovaným buď v hlavičce nebo implicitně.

3.2 LoRaWAN™

Tato kapitola je založena převážně na informacích získáních z technické dokumentace LoRaWAN. [6][5]

Jedná se otevřený komunikační protokol implementující vyšší vrstvy komunikace. Správu protokolu zajišťuje LoRa Alliance. LoRaWAN MAC (Media Access Control) je vyšší vrstva řídící přenos dat aplikace po fyzické vrstvě LoRa PHY. Vrstvy si můžeme prohlédnout na obrázku 3.2. Umožňuje nám obousměrnou komunikaci a to pomocí takzvaných přijímacích okének (receive window). V těchto přijímacích okénkách se zařízení přepíná do naslouchacího módu a čeká zda mu od brány nedorazí nějaká zpráva. Každé zařízení v síti se musí autentizovat. Zařízení se autentizují vůči řídícímu serveru nikoliv bráně. Při komunikaci zařízení odešle zprávu, kterou zachytí všechny dostupné brány v okolí. Brány poté předají zprávu příslušnému řídícímu serveru standardním IP spojením. Pokud řídící server přijme stejnou zprávu od více bran jednoduše ji zahodí. Server se pak již postará o předání zprávy případnému aplikačnímu serveru. Zařízení komunikující tímto protokolem se dělí na tři třídy.



Obrázek 3.2: LoRaWAN - vrstvy [6]

3.2.1 Struktura sítě

LoRaWAN sítě se dělí na několik částí. Koncová zařízení, brány, řídící servery a aplikační servery. Propojení mezi koncovými zařízeními a branami je uskutečněno pomocí právě sítě LoRa. Brány a koncová zařízení navzájem tvoří hvězdicovou topologii kdy koncové zařízení může být v dosahu více bran.

Koncové zařízení

Uživatelské zařízení připojené do sítě. Toto zařízení vysílá zprávy které zachytávají brány, které zprávu zpracují a předají dále. Zařízení může zprávy také přijímat kdy může zařízení přijímat je specifikováno jeho třídou. Většinou se jedná o zařízení s minimálním odběrem energie určené například k měření teploty nebo jiné typy senzorů.

Brána (Gateway)

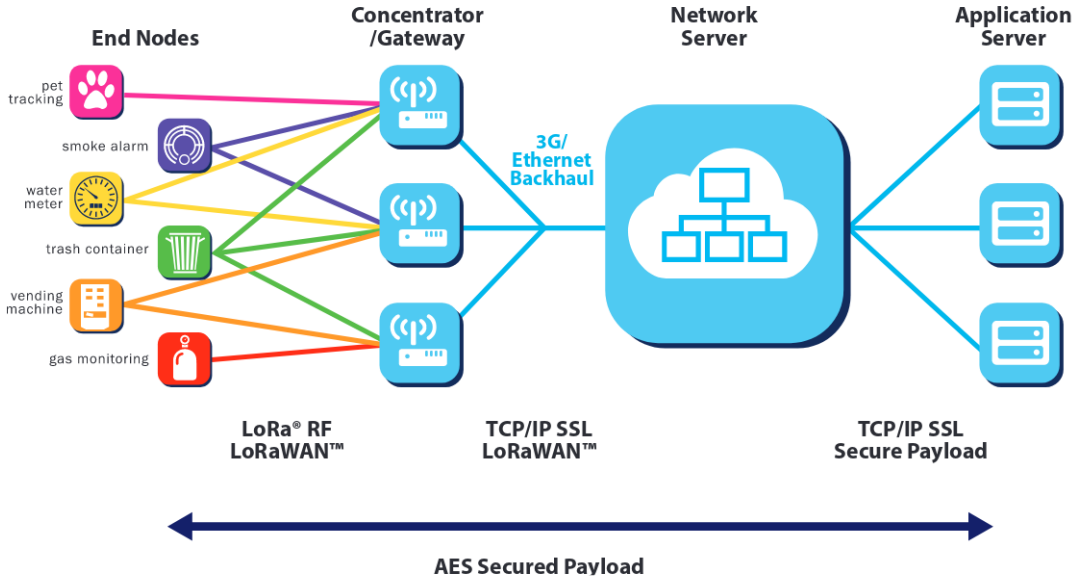
Brána slouží k přeposílání dat přijatých od koncového zařízení řídicímu serveru. Brány zajišťuji pokrytí území sítí LoRa. Komunikace mezi bránou a řídícím serverem probíhá po IP síti. Brány jsou tedy něco jako BTS stanice v mobilních sítích.

Řídící server (Network Server)

Řídící server spravuje komunikaci v síti LoRaWAN poskytované určitým operátorem. Rolemi řídícího serveru jsou inicializace koncového zařízení, zajištění šifrování a dešifrování řídících MAC příkazů, přeposílání dat na správný aplikační server. Kontrola a třídění zpráv od bran aby nedocházelo k zdvojení zpráv pro aplikační server, jelikož jedno koncové zařízení může být v dosahu více bran. Také se stará o výběr vhodné brány pří komunikaci směrem ke koncovému zařízení. A v případě, že mu to koncové zařízení umožní, tak i k řízení jeho parametrů rádiového přenosu.

Aplikační server (Application Server)

Konečné zpracování uživatelských dat odeslaných koncovými zařízeními jež spravuje. Data dešifruje a uloží je do databáze. Pokud má připravená data k odeslání nějakému zařízení tak je zašifruje a předá řídícímu serveru, který vybere vhodnou bránu k vyslání informace pro koncové zařízení.



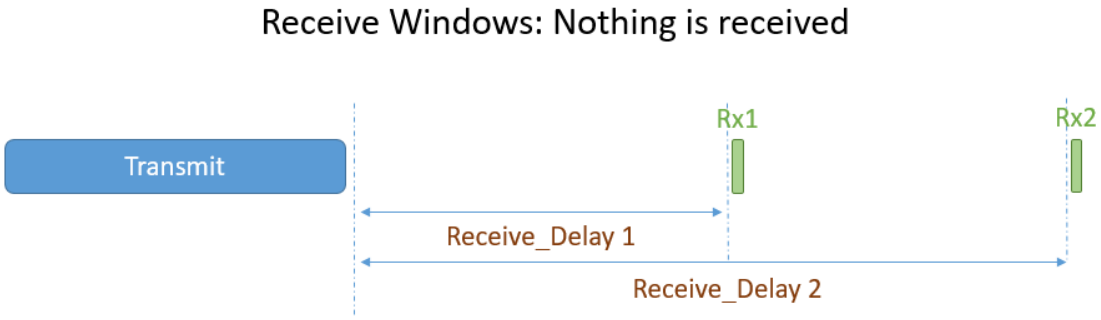
Obrázek 3.3: LoRaWAN - struktura sítě [13]

3.2.2 Třídy zařízení

Koncová zařízení dělíme do tří tříd A, B a C. Jednotlivé kategorie určují způsob komunikace s bránou a tím také výrazně ovlivňují spotřebu koncového zařízení. Všechna koncová zařízení musejí implementovat minimálně funkce třídy A. Pokud zařízení podporuje více tříd je poté možné volitelně používat funkce vyšších tříd za cenu zvýšení spotřeby.

Třída A

Koncový zařízení třídy A zajišťují obousměrnou komunikaci pomocí pomocí takzvaných downlink okének po odeslání zprávy. To znamená, že server nemá možnost posílat pravidelně data těmto zařízením. Může s nimi komunikovat pouze bezprostředně po té co se zařízení samo rozhodne odeslat nějaká data. Jak mile zařízení odešle zprávu otevře přijímací okénko. Pokud v prvním okénku neobdrží žádnou zprávu otevírá okénko druhé. Druhé okénko je otevřeno s jinými parametry definovanými pomocí MAC příkazů složí ke zmírnění dopadu zarušení.



Obrázek 3.4: Přijímací okénka třída A [13]

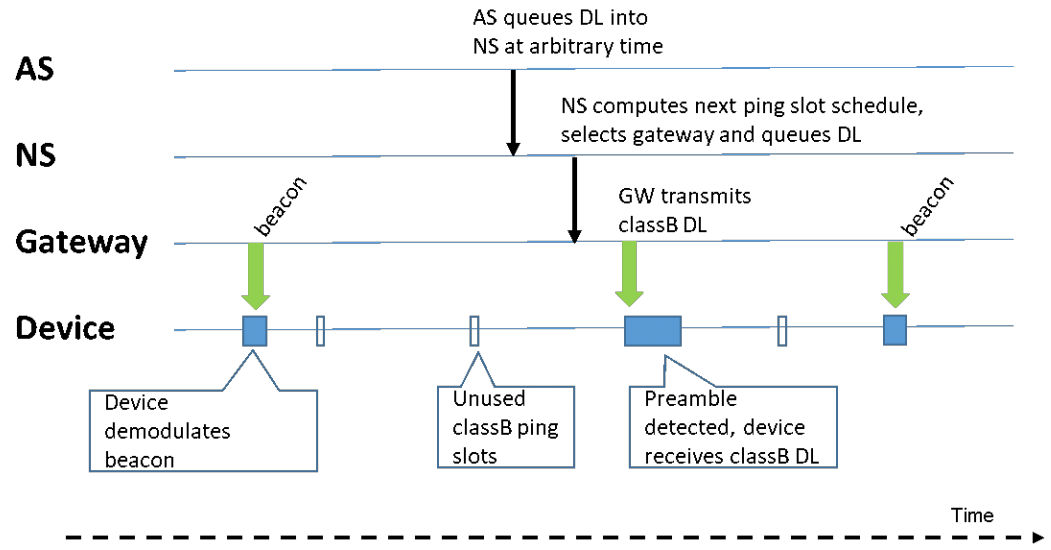
Obrázek 3.4 znázorňuje časování přijímacích okének v případě, že aplikační server nepředal žádnou zprávu. Druhé přijímací okénko je otevíráno pouze v případě, že v prvním nedošlo k přijetí žádné zprávy. Zpoždění přijímacích okének jsou definovány regionálními parametry a mohou být změněny pomocí MAC příkazů.

Třída B

Třída B obrázek 3.5 je rozšíření vlastností třídy A o plánování dodatečných přijímacích časových okének. Zařízení se v tomto modu pravidelně probouzí a pokouší se přijmout zprávou beacon nebo data které mu zasílá jeho aplikační server. Pokud zařízení při otevření okénka nezachytí Preamble opět se co nejrychleji uspý a čeká do dalšího okénka (Pink slot). Beacon zprávy slouží k udržení synchronizace pravidelného otevírání okének.

Třída C

Koncová zařízení operující v modu Třídy C mají přijímací okénka téměř stále otevřená. Přijímací okénka se zavírají jen v případě, že zařízení potřebuje odesílat data. Zařízení implementují přijímací okénka třídy A, ale otevírají své přijímací okénko i v době čekání mezi otevíráními okénky Třídy A. Tato třída je nejnáročnější na spotřebu elektrické energie, avšak umožňuje obousměrnou komunikaci s nejlišší prodlevou.



Obrázek 3.5: Časování okének a příjem zpráv Třídy B [12]

3.2.3 Formát zpráv MAC

Všechny zprávy v LoRaWAN sítích z hlediska fyzické vrstvy jsou definovány protokolem LoRa popsaným výše. Datové pole fyzické zprávy má ztrukturu MAC zpráv. Tyto zprávy mají tři varianty. První varianta slouží jak pro přenos uživatelských dat tak pro řízení komunikace v systému je tím pádem nejpoužívanější. Varianta druhá slouží k připojení zařízení, nebo znovu připojení do sítě. Třetím typem se připojení potvrzuje.

PHYPayload:	MHDR : 8	MACPayload			MIC : 32
MACPayload:	FHDR : 56..176	FPort : 8	FRMPayload (encrypted)		
FHDR:	DevAddr : 32	FCtrl : 8	FCnt : 16	FOpts : 0..120	
MHDR:	MType : 3	RFU : 3	Major : 2		
FCtrl: Uplink:	ADR : 1	ADRAckReq : 1	ACK : 1	FPending : 1	FOptsLen : 4
FCtrl: Downlink:	ADR : 1	ADRAckReq : 1	ACK : 1	RFU : 1	FOptsLen : 4
FOpts:	MACCommand_1 : 8..40	...	MACCommand_n : 8..40		
MACCommand:	CID : 8	Args : 0..32			

Obrázek 3.6: Formát zprávy MAC [6]

MAC hlavička (MHDR)

Specifikuje typ zprávy zprávy - položka MType. Položky RFU a Major pak slouží k specifikaci různých typů rámců.

MType	Popis
000	Join-request
001	Join-accept
010	Unconfirmed DataUp
011	Unconfirmed Data Down
100	Confirmed DataUp
101	Confirmed Data Down
110	Rejoin-request
111	Proprietary

MAC Payload

Obsahuje hlavičku rámce (FHDR) ve které se nachází adresa zařízení (DevAddr), počítadlo rámců (FCtrl) a volitelně nastavení rámce (FOpts) sloužící pro přenos řídících příkazů MAC.

Dále může obsahovat FPort slouží pro identifikaci na aplikační vrstvě. Toto položka je poviná v pokud je FRMPayload přítomen. FRMPayload pak přenáší samotná data, které zařízení potřebuje odeslat.

MIC

Slouží k zajištění integrity přenášené zprávy.

3.2.4 Bezpečnost

Protokol LoRaWAN byl od začátku navrhován s vysokým důrazem na bezpečnost. Zabezpečení je založené na standardu AES kryptografie. Celý systém je založen na dvou 128 bitů dlouhých klíčích (AppKey, NwkKey), které jsou koncovému zařízení přiděleny výrobcem. NwkKey je nutní pro OTAA aktivaci v případě že zařízení podporuje ABP není NwkKey vyžadován. V tomto případě musí zařízení přidělena adresa již výrobcem. AppKey slouží šifrování uživatelských data na cestě k aplikačnímu serveru. Pro nás je hlavní, že bez znalosti klíčů není možné LoRaWAN zprávy analyzovat.

3.2.5 OTAA

Zařízení odešle zprávu obsahující DevEUI, AppEUI a AppKey. Ke zprávě je připojen kontrolní součet MIC vygenerovaný CMAC AES. Zpráva typu join-request s obsahem DevNonce a AppEui je odeslána. V momentu kdy síťový server obdrží správu zkontroluje zda hodnota DevNonce nebyla ještě použita. Vygeneruje odpověď join-accept s obsahem DevAddr, AppNonece. Zpráva je šifrována pomocí AppKey, který musí být síťovému serveru známý. Nyní mají jak síťový server tak koncové zařízení náhodná čísla DevNonce, AppNonce. A mohou vygenerovat nezávisle na sobě stejné relační klíče AppSKey a NwkSKey. Aplikační relační klíč je dále používán při šifrování přenášených dat a MAC příkazy jsou šifrovány síťovým relačním klíčem.

Rozdíl ve specifikaci verze 1.0 a 1.1

Verze 1.0

Používá pouze jeden klíč AppKey, ze kterého vygeneruje při připojování do sítě dva klíče (AppSKey, NwkSKey) na základě poskytnuté náhodné sekvence od řídícího serveru. AppSKey je používán k šifrování samotných uživatelských dat a NwkSKey zabezpečení řídících signálů. Problém tohoto řešení je, že Řídící server má všechny potřebné údaje pro dešifrování uživatelských dat. V případě připojení bez Aktivace jsou klíče AppSKey a NwkSkey uloženy od výroby v zařízení.

Verze 1.1

Pro zajištění zvýšení bezpečnosti byl přidán nový klíč NwkKey. AppKey je nyní známí jen pro aplikační server a koncové zařízení stejně jako v předchozí verzi je na základě náhodné posloupnosti s tohoto klíče vygenerován nový dočasný klič AppSKey sloužící k šifrování uživatelských dat. NwkKey pak slouží k vygenerování třech nových dočasných klíčů FNwkSIntKey, SNwkSIntKey, NwkSEncKey. První dva klíče se používají pro výpočet a verifikaci kontrolních součtů. Klíčem NwkSEncKey jsou pak šifrovány určité části hlavičky hlavně řídící MAC příkazy.

Kapitola 4

Návrh LoRaWAN snifferu

4.1 Výběr softwaru pro zpracování signálu s SDR

Při výběru vhodného softwaru pro implementaci našeho snifferu musíme brát v potaz využití SDR LimeSDR definovaného v zadání. To nás ovšem příliš příliš neomezuje, jelikož ovladače pro LimeSDR jsou dostupné jak pro operační systém Linux tak pro Windows. Tudíž při výběru dalšího softwaru nebudeme omezeni na jeden operační systém. Následně je třeba zvolit vhodný nástroj pro zpracování dat s SDR.

Zde máme na výběr s různých nástrojů pro zpracování signálů GNU Radio, Pothosware, Mathlab, Octave. Pro nás ideální software musí mít možnost zpracovávat a nastavovat parametry LimeSDR. Proto se v dalších sekci zaměříme na srovnání GNU Radia a Pothosware.

4.1.1 GNU Radio

Gnu radio je open-source software poskytující vývojové prostředí pro implementaci zpracovaní signálu v reálném čase. Poskytuje sadu bloků, které skládáme za sebe a vytváříme tak komplexní systém. Poskytuje nám vlastně jednoduchý způsob jak proměnit na-vzorkovaný signál s nějakého SDR například v přijímač FM radia nebo v našem případě dekódovat pakety LoRa sítě.

Knihovny

Pro komunikaci a nastavení LimeSDR jsem otestoval v rámci GNU Radia dvě knihovny kompatibilní s LimeSDR. První testovaná knihovna byla OsmoSDR, která spolu s knihovnou SoapyOsmo pro vytváří most pro nastavení SDR s podporou SoapySDR jako je naše LimeSDR. Tuto možnost jsme otestovali a podařilo se sní nakonfigurovat SDR pro příjem požadovaných frekvencí. Nastavení však nebylo zcela intuitivní hlavně z hlediska volby kanálu a cesty LNA. Poté jsem vyzkoušel knihovnu gr-limesdr. Velkou výhodou této knihovny je, že všechny popisy atributů vypovídají o tom co vlastně v SDR nastavujeme na rozdíl od předchozí testované knihovny. Je zde jasné nastavení kanálu a LNA cesty. Knihovna umí také sama detekovat připojené zařízení. V neposlední řadě nám umožňuje k nastavení parametrů využít soubor vygenerovaný pomocí LimeSuit a tak velmi přesného manuálního ladění.

Zpracování signálu - implementaci demodulace a dekódování signálu LoRa jsem vyzkoušeli dvě knihovny obě stejného jméno gr-lora. První knihovna vytvořená Matt Knightem. Jež se zasloužil za osvětlení principu LoRa modulace svou prací zabývající reverzním inže-

nýrství této modulace. Bohužel této knihovně chybí jedna pro nás důležitá vlastnost a to dekódování explicitní hlavičky LoRa. Vzhledem k tomu, že knihovna postrádá tuto funkcionalitu a její další vývoj je už 4 roky neaktivní nebude pro nás nejvhodnější. Druhá knihovna vyvíjená převážně Pieter Robynsnem splňuje naše požadavky mnohem lépe. Umožňuje dekódování s explicitní hlavičkou, kterou používají všechny rámce v rámci sítí LoRaWAN. Detailněji si popíšeme vlastnosti a nastavení této knihovny v sekci zabývající se vytvořením extcap rozhraní pro WireShark.

4.1.2 Pothosware

Pothosware je velmi podobný nástroj Gnu radiu jak funkcionalitou tak uživatelským prostředím. Poskytuje nám podobnou sadu bloků pro zpracovávání signálu. Hlavním vývojářem tohoto nástroje je Josh Blum jenž je také jedním s důležitých autorů Gnu radia. Jeho důvodem pro vytvoření Pothoswaru byla neochota komunity Gnu radia akceptovat některé radikálnější úpravy api pro vytváření nových bloků, navrhované změny ve správě bufferu a implementaci zpráv.

Knihovny

Na rozdíl od Gnu radia zde nepotřebujeme instalovat žádnou doplňkovou knihovnu pro nastavení a příjem dat s LimeSDR. Vystačíme si s blokem SourceSDR který využívá Soapy. Opět však toto nastavení není tak jednoduché jako v případě knihovny gr-limesdr pro Gnu radio.

Pro práci s LoRou zde najdeme knihovnu LoRa-SDR. Knihovna poskytuje dva hlavní bloky jeden pro demodulaci a druhý pro dekódování. Po prozkoumání parametrů nastavení příjmu LoRa v této knihovně se jevila jako dobrý kandidát pro tuto práci. Bohužel při pokusech s příjmem s reálných zařízení na místo dat vygenerovaných touto knihovnou se ukázalo, že kompatibilita s reálnou implementací v zařízeních není příliš dokonalá. Při pokusech o příjem LoRa paketu odeslaného za pomoci čipu SX1276 se nepodařilo dekódovat žádnou zprávu ani při jednom z mnoha zkoušených nastavení. Z tohoto důvodu se knihovna jeví jako použitelná pouze pro simulační účely.

4.2 Vybraný software

Na základě zjištěných informací o jednotlivých nástrojích a jejich knihovnách jsme se rozhodli pro nástroj Gnu radio. Pro následnou analýzu dekódovaných dat použijeme WireShark. WireShark má implementovaný disector pro LoRa a LoRaWAN tudíž nám velmi ulehčí práci při analýze komunikace. Jak dostat data do v potřebném formátu do WireSharku se dozvíme v sekci o implementaci extcap rozhraní.

4.2.1 Definice použitého softwaru

Software	Verze	Zdroj
GNU Radio	3.8.1.0	arch repo
gr-lora	rpp0/gr-lora	github
gr-limesdr	myriadrf/gr-limesdr	arch aur
WireShark	WireShark 3.2.5	arch repo

Kapitola 5

Implementace extcap rozhraní pro WireShark

Implementace našeho sniferu se skládá z GNU Radia a jeho vybraných knihoven. Data jsme se rozhodly analyzovat s pomocí programu wireshark. Pro zjednodušení interakce mezi wiresharkem, GNU radiem a SDR využijeme vytvoříme extcap rozhraní, které nám umožní provést veškeré potřebné nastavení parametrů pro příjem a dekódování LoRy přímo ve wiresharku. Ve výsledku tedy uživatel tohoto sniferu pouze otevře wireshark nastaví parametry příjmu pohodlně v grafickém prostředí a spustí zachytávání.

5.1 Co je extcap?

Extcap rozhraní jsou univerzální rozšiřující moduly, které umožňují externím programům chovat se stejně jako rozhraní integrovaná ve wiresharku. Používáme jej v případě kdy zdrojem je například nějaký nestandardní kus hardwaru nebo softwaru.[18] Vždy můžeme dosáhnout zachytávání pomocí přímého zápisu do souboru wiresharku, ale díky extcapu získáme pohodlné rozhraní ovládatelné přímo v našem oblíbeném síťovém analyzátoru. Rozhraní je možné vytvořit v jakémkoliv programovacím či skriptovacím jazyku. Musíme se pouze držet textového rozhraní pro komunikaci s wiresharkem definovaného v programátorské příručce.[19]

Iplementace je založena na úkázkovém zdrojovém kódu pro python z dokumentace wiresharku zveřejněné githubu`https://github.com/wireshark/wireshark/blob/master/doc/extcap_example.py`.

5.2 LoRa interface for sdr: loradump

5.2.1 Popis funkcionality

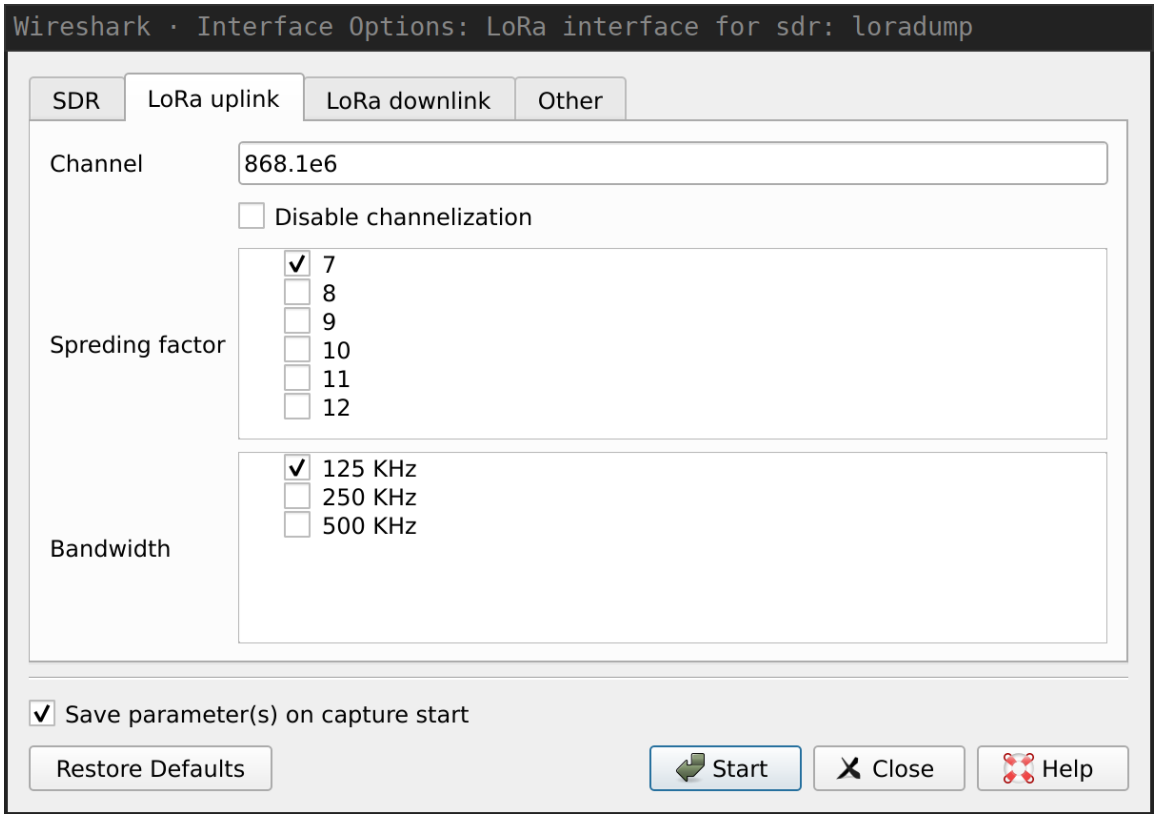
Jak již bylo uvedeno výše rozhraní nám umožní nastavit všechny parametry příjmu a dekódovaní přímo ve wiresharku. Musíme tedy na implementovat parsování argumentů ze kterých následně vygenerujeme GNU Radio flow graph. Flow graph nám popisuje veškeré nastavení bloků pro GNU Radio ve značkovacím jazyku yaml. Vygenerovaný yaml soubor poté přeložíme pomocí grcc (GNU Radio Companion Compiler) na spustitelný soubor v pythonu nebo c++. V našem případě je výsledkem spustitelný soubor v pythonu, který

spustí SDR a začne provádět dekódování LoRa signálu. Teď nám již zbývá pouze předat dekódované zprávy do wiresharku

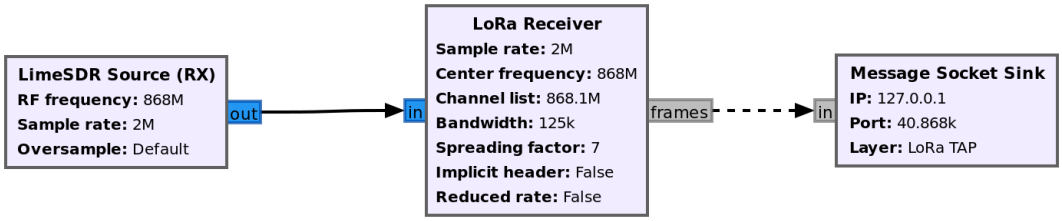
5.2.2 Konfigurace rozhraní

V této sekci si popíšeme možnosti konfigurace parametrů pomocí vytvořeného extcap rozhraní.

Sekce	Parametr	Typ
sdr	RF frequency	double
	Sample rate	double
	Gain	int
LoRa uplink	Channel	double
	Disable channelization	bool
	Spreding factor	select 7-12
	Bandwidth	select 125 KHz, 250 KHz, 500 KHz
LoRa downlink	Channel	double
	Disable channelization	bool
	Spreding factor	select 7-12
	Bandwidth	select 125 KHz, 250 KHz, 500 KHz
Other	Remove LoRa PHY header	bool
	Show GNU Radio QT Sink	bool



Obrázek 5.1: loradump nastavení ve wiresharku



Obrázek 5.2: Ekvivalentní flow graph nastavení v obrázku 5.1

Význam většiny parametrů s této tabulky jsme popsaly v sekci popisující LoRa modulaci. 3.1 Proto se zaměříme hlavně na popis parametrů nesouvisejících s LoRou. U parametrů týkajících se LoRa modulace se zaměříme pouze na jejich implementaci a dopady zvolených kombinací.

Sekce - sdr

Zde musíme správně nastavit středovou frekvenci signálu (RF frequency). Úzce souvisí s nastavením další hodnoty vzorkovací frekvence (Sample rate), která určuje šířku nevzorkovaného pásma. Tyto hodnotu nastavujeme podle toho na jakých kanálech chceme přijímat. Chceme-li například přijímat na kanálu definovaného frekvencí 868.1 MHz musíme nastavit SDR tak, aby celá šířka kanálu LoRA byla v obsažena v šířce na vzorkovaného signálu s SDR. Zvolíme tedy například středovou frekvenci pro SDR 868 MHz a vzorkovací frekvenci na 500 KHZ. Tím si můžeme být jistí že signál na požadovaném kanálu bude obsažen v nevzorkovaném signálu. Důvodem proč volíme vyšší vzorkovací frekvenci než je nutné přesnost zpracovaní signálu. Pro většinu testů v této bakalářské práci byla používaná vzorkovací frekvence 2 MHz. Touto vzorkovací frekvencí pokryjeme všechny kanály LoRaWAN sítí běžně používané v Evropě.

Dalším neméně důležitým parametrem v nastavení SDR je zisk (gain). Tato hodnota určuje míru zesílení signálu s antény. Při našem testování byla pro většinu testů nastavena tato hodnota na 10 dB. Jelikož se vysílač nacházel velmi blízko přijímače na nastavení této hodnoty příliš nezáleželo.

Sekce - LoRa uplink, LoRa downlink

Prvně si vlastně vysvětlíme proč rozlišujeme mezi nastavením pro odesílání (uplink) a přijímáním (downlink). Důvodem je, že zprávy odeslané bránou jsou modulovány jinak než zprávy odesílané koncovým zařízením. Brány totiž provádí inverzi IQ signálu což v praxi znamená že chirp začíná na nejvyšší frekvenci a končí nejnižší frekvencí. To zabraňuje tomu aby koncová zařízení zachytávali svoje zprávy mezi sebou a brány zase nezachytávali komunikaci ostatních bran.https://www.overleaf.com/project/5dd996b59aae2e000124bdc5

Jediný parametr, který nám není známy s popisu LoRa modulace, je vypnutí rozdělení na kanály (Disable channelization). Tímto nastavením vypneme v LoRa Receiver bloku rozdělení na jednotlivé kanály. Což nám umožní dekódovat na všech kanálech obsažených v nastavené Šířce pásma. Toto nastavení má negativní vliv na citlivost při detekci. Jeho implantace v bloku je hlavně z důvodů nedokončení rozdělování do jednotlivých kanálů, které aktuálně podporuje pouze jeden kanál. U parametrů šířky pásma (Bandwidth) a rozkládacího faktoru (Spreding factor) je nutné zmínit, že vytváří nové LoRa Receiver

bloky. Tedy pokud máme vybrány SF 7, 8 a šířku pásma 125 KHz a 250 KHz vytvoříme pak 4 LoRa Receiver bloky. Dekódovaní s více různými parametry má tedy vyšší požadavky na výkon při zpracování.

Sekce - Other

V této sekci můžeme vypnout odstraňování LoRa PHY vrstvy. Tato funkcionalita je výchozím nastavení zapnuta. Vrstvu chceme odstranit, abychom mohly správně interpretovat data ve wiresharku. Implementace disektorů totiž nepodporuje zobrazení LoRa PHY vrstvi. Wireshark očekává, že data zabalená v LoRaTAP vrstvě, budou obsahovat pouze LoRaWAN strukturu nikoliv radiovou vrstvu.

Další možností je zapnout QT Sink blok. Můžeme si tak zobrazit výsledek Furierovy transformace ve formě vodopádového grafu. Vidíme tak zastoupení frekvencí v našem signálu v závislosti na čase. Graficky znázornění nám umožní identifikovat, signál na požadovaných frekvencích. Víme tak zda naše analyzované zařízení vůbec něco vysílá nebo ne.

5.3 Úpravy knihovny gr-lora

Upravená verze knihovny je dostupná na přiloženém CD. Pro většinu funkcionalit nejsou úpravy nutné.

Oprava kompatibility s GNU Radiem 3.8

Aktuální verze knihovny vyžaduje GNU Radio verze 3.8 a však ne všechny její funkce byli důkladně otestovány na této verzi. K tomuto problému nás přivedl pokus o implementaci možnosti přijímat více kanálů na jednou s využitím vypnuté rozdělení na kanály, které zatím nebylo dokončeno. Bohužel se však ukázalo, že tato funkcionalita způsobuje nekompatibilitu. Odstranění této chyby nebylo příliš náročné jednalo se pouze o změnu názvu funkce jednoho filtru z standardních knihoven GNU Radia.

Přidání metadat do LoRaTAP

Jelikož naše řešení umožňuje přijímat více kanálů s různými nastaveními zároveň rozhodl jsem se přidat informace o tom s jakým nastavením byl paket zachycen do LoRaTAP hlavičky. Původní verze knihovny zde uvádí pouze SNR. Nám by se hodilo vědět s jakým rozprostíracím faktorem a jakou šířkou pásma byl paket zachycen. V ideálním případě bychom přidaly i frekvenci na které byl zachycen. To se však ukázalo být poměrně komplikovanější z důvodu ne úplně funkční implementace výpočtu posunutí od základní frekvence. Proto byla nakonec tato funkce vypnuta.

Problémy se SF 11, 12

Při testování jsme také narazily na problémy s dekódováním rozprostíracího faktoru 11, 12. Při analýze problému jsme se prvně pokusily vyloučit možnost nedostatečného výkonu. Vytvořili jsme tedy jednoduchý experiment kdy jsme signál odeslaný s těmito parametry uložili prvně do souboru a poté jsme se jej pokusili dekódovat při přehrávání. Problém však přetrval i dále. Zkompilovali jsme tedy knihovnu s parametry pro debugování a začali jsme hledat problém. Zjistili jsme, že knihovna má problém s nalezením hlavičky. Jako možné řešení jsme upravili v kódu magickou konstantu určující kdy je autokorelace přijatého signálu

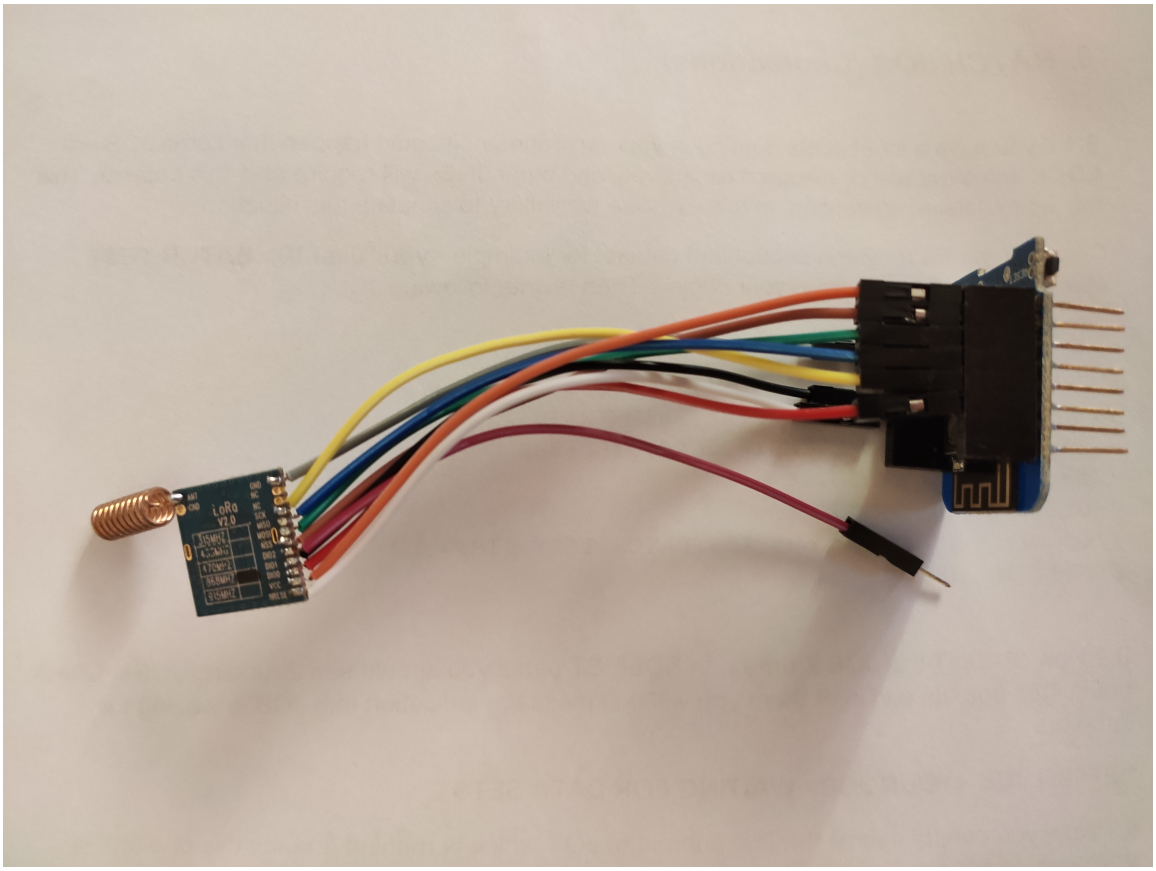
s ideálním upchirpem úspěšná. Bohužel ani tato úprava nevedla k vyřešení problému. Dosáhly jsme sice toho, že hlavička byla detekovaná ale synchronizace už správně nepřehnula. Řešení problému bohužel nebylo úspěšně nalezeno. Nepovedlo se ani s jistotou identifikovat jeho příčinu.

Kapitola 6

Analýza komunikace

6.1 Vybrané koncové zařízení

Jako koncové zařízení jsme zvolili NiceRF LoRa1276 připojený k mikrokontroleru WeMos D1 Mini. 6.1 Mikrokontroler jsme vybíraly pouze na základě podpory vývojové platformy Arduino a dostatečného množství pinů. WeMos D1 byl použit především proto, že jsme jej měli dostupný v době sestavování zařízení.



Obrázek 6.1: WeMos D1 mini s NiceRF LoRa 1276

6.1.1 NiceRF LoRa1276

Jedná se o desku obsahující čip Semtech SX1276 zařizující LoRa modulaci. Všechny desky založené na tomto čipu mají dobrou podporu v arduino knihovnách.[2]

Specifikace

- Frekvence: 868 MHz
- Citlivost až -139 dBm
- Maximální výstupní výkon: 20 dBm
- 13mA - režim přijímače
- Proud v režimu spánku: <200 nA
- Přenosová rychlost: FSK, 1,2-300 Kbps
- LoRa TM, 0.018-37,5Kbps
- Ochrana proti elektrostatickému výboji
- 127 dB dynamický rozsah RSSI
- Pakety až 256 bajtů s FIFO a CRC
- Frekvenční rozprostření

Zapojení

Modul disponuje SPI rozhraním je tedy nutné správně zapojit tuto sběrnici k vybranému mikrokontroleru. V případě našeho mikrokontroleru jsou piny dány knihovnami přidávajícími podporu do android platformy. Zapojíme tedy sběrnici podle na definované piny mikrokontroleru. S dalších pinů modulu budeme potřebovat pouze DIO0 a DIO1 sloužících k monitorování stavu přenosu. Tyto piny můžeme zapojit na libovolné volné digitální piny.

6.1.2 Knihovny pro práci s čipem

Zde si krátce představíme použité knihovny pro ovládání čipu LoRA1276 za použití arduino platformi. Popíšeme k čemu a jak jsme je při testování využívaly. Zdrojové kódy pro mikrokontroler použité při testování jsou dostupné na přiloženém CD. Jedná se pouze o mírně upravené ukázkové kódy.

Arduino LoRa

Arduino knihovana pro přijímání a vysílání za použití LoRa modulace. Podporuje většinu mudulů obsahující čipy Semtech SX1276/77/78/79. Tato knihovna implementuje pouze nastavení modulu pro odeslání a příjem zpráv neobsahuje implementaci vyšší vrstvy LoRaWAN. Při práci s touto knihovnou je důležité mít na paměti, že neimplementuje řízení pracovního cyklu vysílání. [11]

Využili jsme ji v prvotních fázích výběru softwaru pro implementaci snifferu a při ověřování možnosti zachytávání s určitými parametry. Umožňuje nám nastavení jednotlivých parametrů přenosu LoRa, snadno tak ověříme kompatibilitu našeho snifferu.

Arduino-LMIC library

Jedná se o port implementaci IBM LMIC knihovny do arduino platformy. IBM LMIC je implementace LoRaWAN vrstvy napsaná v jazyce C pro čipy Semtech SX1272/76. Podporuje většinu funkcionality LoRaWAN Třídy A a Třídy B zařízení. Knihovna simuluje chování konocového LoRaWAN zařízení tudíž není možné nastavit parametry modulace, ty jsou voleny stejně jako u hardwarových zařízení v závislostí na úspěšnosti probíhající komunikace.[7]

Toto knihovnou budeme simulovat analyzované koncové zařízení v režimu aktivace ABP a později OTAA.

6.2 Analíza pomocí našeho snifferu

V této sekci si popíšeme veškerá nastavení snifferu při prováděné analýze a vyhodnotíme dosažené výsledky. Pokusíme se analyzovat komunikaci zařízení v režimu aktivace ABP tak v režimu OTAA.

Společné vlastnosti experimentu

V obou experimentech je zařízení nastaveno tak aby bylo schopné komunikovat s komerční sítí ČRA. Jelikož se jedná pouze o experiment v obou případech bude přenášená zpráva obsahovat text "Hello world!"zakódovaný pomocí ASCII. Nebudeme se tedy zabývat pokusem o dokódování dat která se v LoRa sítích většinou přenášejí různě komprimovaná pro co nejrychlejší odeslání a úsporu pracovního cyklu.

6.2.1 ABP aktivace

Začneme pokusem při přenosu s ABP aktivací. Downlink zprávy nejsou v pokusu očekávány. Program pro nahraný do koncového zařízení se nachází na CD ve složce Arduino/cra-abp.

Paramety zařízení:

Všechny hodnoty jsou MSB formátu.

Adresa zařízení: 2601139A

Sítový klíč: A50B54399F05A1B57AC8E861219CC0BD

Aplikační klíč: 3BC480AD4EA8FD55B5E8131D0E7C08F7

Nastavení příjmu na snifferu

Nejprve si ve wiresharku nastavíme interpretaci LoRaTAP data s kódovým slovem 0x00 jako LoRaWAN. V nastavení disektoru LoRaWAN přidáme šifrovací klíče spolu s adresou zařízení. Díky tomu bude wireshark schopný automaticky dekódovat přijatou zprávu a ověřit kontrolní součet.

SDR frekvence: 868 MHz

Vzorkovací frekvence: 2 MHz

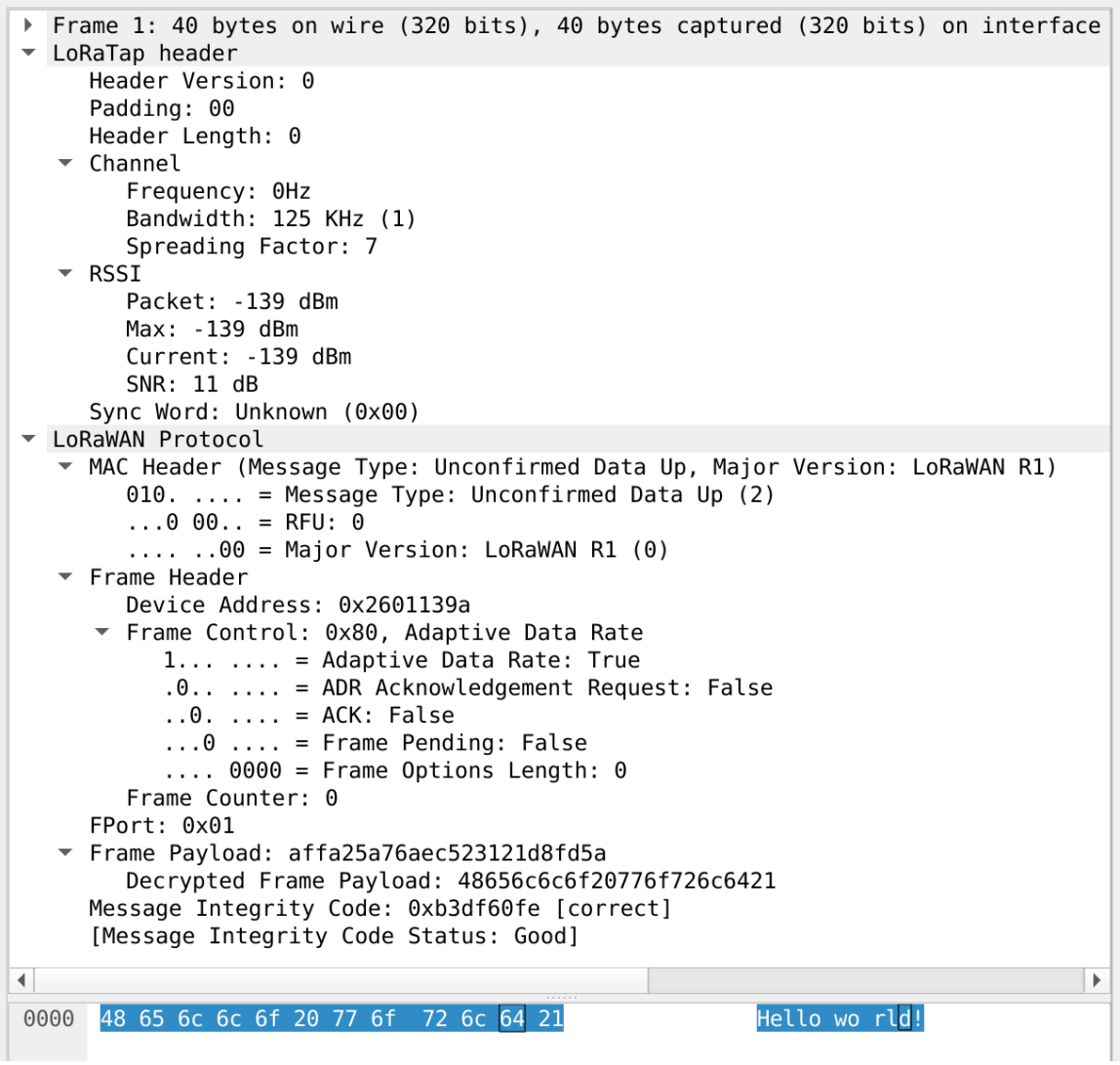
Zisk: 30 dB

LoRa uplink

- Vypnuté rozdělení na kanály
- Rozprostírací faktor: 7, 8, 9, 10
- Šířka pásma: 125 KHz

Other:

- Odstranění LoRa PHY hlavičky
- Zobrazení QT Sink - není podstatné



Obrázek 6.2: Wireshark - ABP aktivace

Na obrázku 6.2 vidíme zachycenou zprávu.

LoRaTAP header

Nejedná se o data která by byla reálně přenášená v tomto paketu je to pouze obálka, kterou gr-lora a wireshark používají k zobrazení metadat o parametrech přenosu. Bohužel většina těchto metadat není knihovnou gr-lora podporována proto si popíšeme alespoň těch pár co se zde nacházejí

SNR: 11 dB - Vyjadřuje sílu užitečného signálu v poměru k šumu

Bandwidth: 125 KHz - Šířka pásma se kterou byl paket přenesen

Spreding Factor: 7 - Použitý rozprostírací faktor

LoRaWAN MAC Header

Zde je nejdůležitější první hodnota 010 Unconfirmed Data Up. Znamená že naše zařízení odesílá tuto zprávu bez toho aniž by očekávalo přijetí ACK potvrzení pokud není zapnutá funkce ADR.

Frame Header

Device Address: 0x2601139a - adresa zařízení, ve skutečnosti je tato adresa přenášena v převráceném tvaru ale pro pohodlnost nám wireshak adresu otočil.

Frame control:

- Adaptive Data Rate: True - je aktivován algoritmus výpočtu úpravy rozprostírací ho faktoru tak aby nedocházelo k zbytečnému využívání pracovního cyklu
- ADR Acknowledge Request: False - slouží algoritmu ADR při zvyšování rozprostírací ho faktoru jelikož je signál pro bránu příliš slabý
- ACK: False - Potvrzení předchozího requestu v tomto případě je to poprvé co se zařízení v síti objevilo není co potvrzovat
- Frame pending: False - Používá se pouze pro downlink komunikaci zančí že brána má kdyspozici další data k odeslání.
- Frame option length: 0 - Neposíláme žádné další frame options
- Frame counter: 0 - První odeslaná správa hodnota je zkaždou odeslanou zprávou zvyšována používá se při výpočtu kontrolního součtu MIC

- FPort 0x01 - Označení portu příjemce na aplikačním serveru
- Frame payload - obsahuje zakódovanou zprávu v dolní části můžeme vidět obsah zprávy po dekódování
- Message Integrity Code - slouží k ověření zda přijatá zpráva nebyla poškozen. Je vypočten po zašifrování zprávy za pomocí síťového klíče

6.2.2 OTAA aktivace

Pokusíme se zachytit kompletní aktivaci OTAA v síti ČRA. Program koncového zařízení tentokrát nalezneme CD ve složce Arduino/cra-otaa.

Paramety zařízení:

Všechny hodnoty jsou MSB formátu.

EUI zařízení: 008BD1DE6994969D

EUI aplikace: 70B3D57ED003238E

Aplikační klíč: 1FBE00DF1EC49266E0A887562D07D6E1

Nastavení příjmu na snifferu

Nastavení wiresharku ponecháme stejné jako v předchozím případě. Tentokrát však nebudeme přidávat žádné klíče jelikož ty budeme muset teprve odvodit ze správ join request a join accept.

SDR frekvence: 868 MHz

Vzorkovací frekvence: 2 MHz

Zisk: 30 dB

LoRa uplink

- Vypnuté rozdělení na kanály
- Rozprostírací faktor: 7, 8, 9, 10
- Šířka pásma: 125 KHz

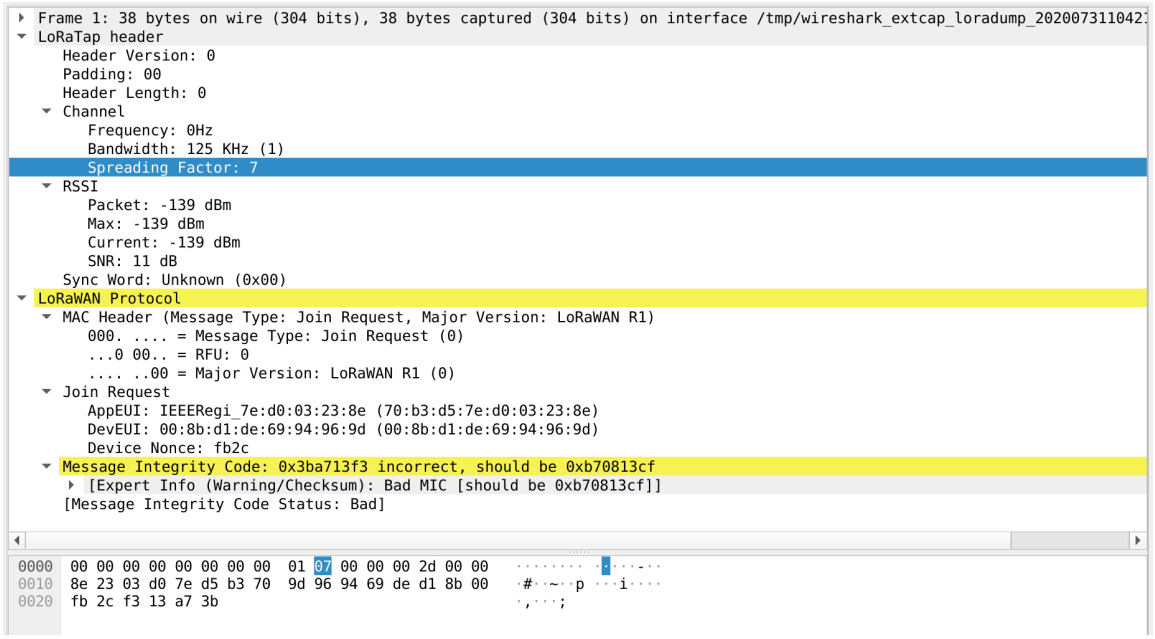
LoRa downlink

- Vypnuté rozdělení na kanály
- Rozprostírací faktor: 7, 8, 9, 10
- Šířka pásma: 125 KHz

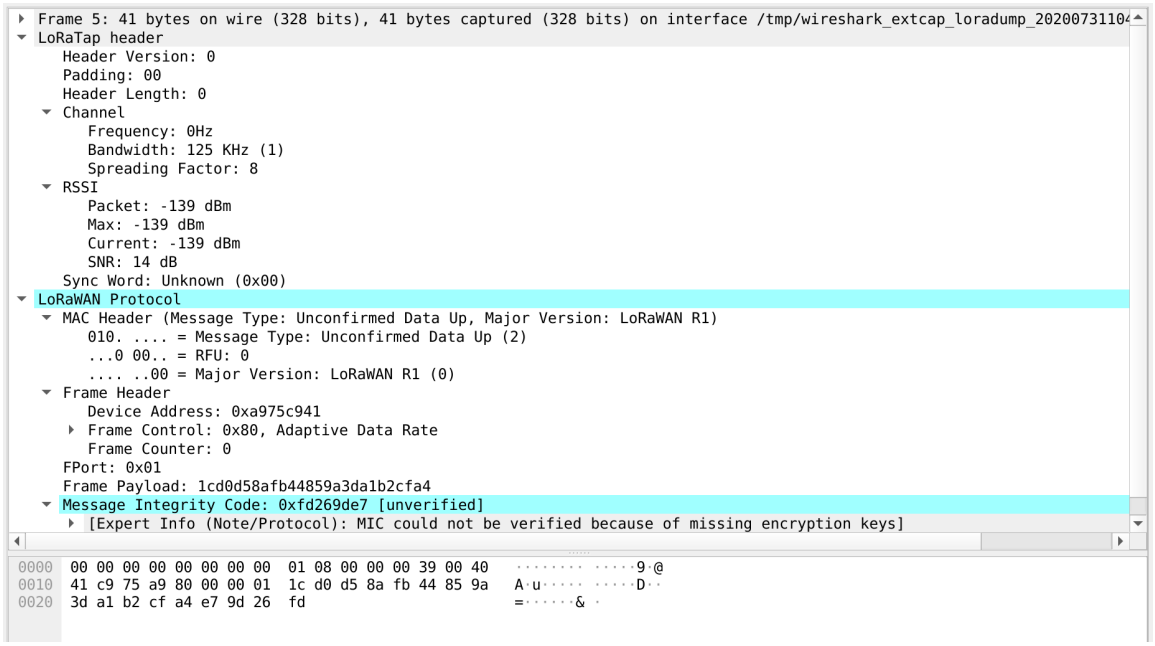
Other:

- Odstranění LoRa PHY hlavičky
- Zobrazení QT Sink - není podstatné

Můžeme si všimnout, že zpráva na obrázku 6.3 nenese žádná užitečná data. Jedná o join-request zprávu na kterou očekáváme odpověď odpověď join-accept. Zařízení po této zprávě odeslalo join-requst požadavek ještě dvakrát. Třetí v pořadí měl již vyšší hodnotu rozprostíracího faktoru. Bohužel však ani po této zprávě jsme nezachitili join-accept další zachycená zpráva je na obrázku 6.4 a jedná se již o zprávu typu unconfirmed data up. Ztoho usuzuji, že i přesto že se nacházíme poměrně blízko brány se bohužel našemu SDR nepodařilo join-accept zprávu zachytit. Z toho důvodu nemůžeme pokračovat v dalším analýze těchto paketů.



Obrázek 6.3: Wireshark - join-request

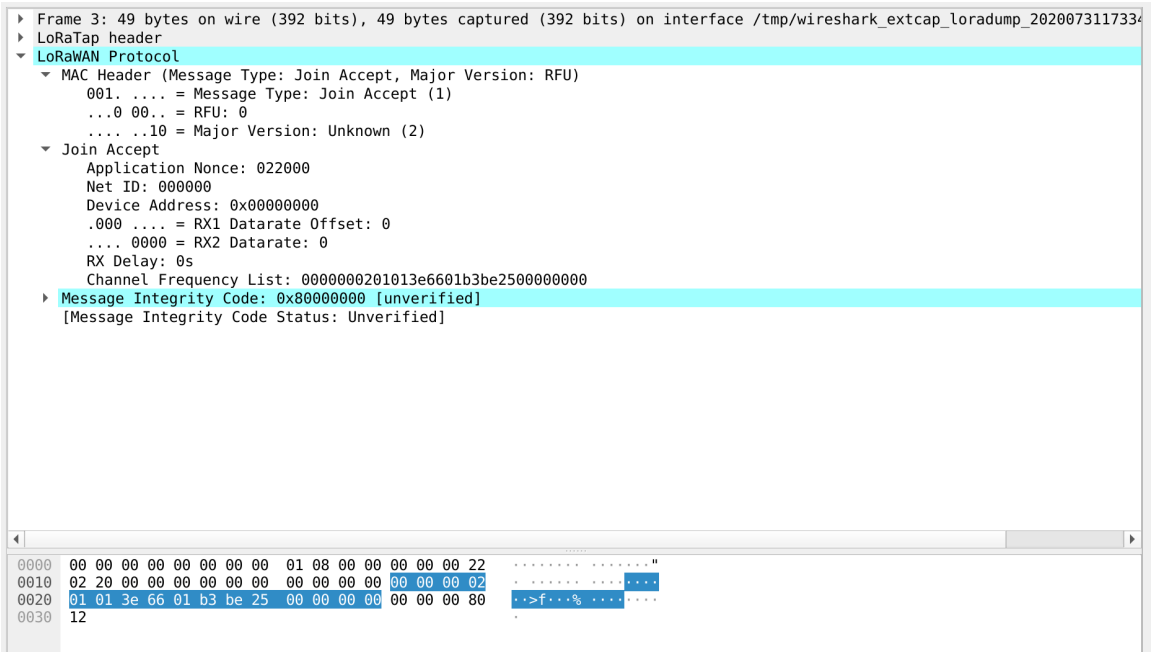


Obrázek 6.4: Wireshark - unconfirmed data up

Po prvním nevydařeném pokusu se nevzdáme a jestli nestačí že se LoRaWAN brána nachází na paneláku vedle nás tak se pokusíme dostat ještě blíže.

Druhý experiment tedy provedeme z auta přímo u mobilního vysílače na kterém je brána umístěna. Postup je opět úplně stejný použijeme stejné nastavení přijímače i vysílače jako v předchozím pokusu.

Experiment s počátku probýhá naprosto stejně jsou odeslány dva join-requesty s rozptylovacím faktorem 7 na, které nedostáváme žádnou odpověď. V momentě kdy však zařízení přejde na vyší rozptylovací faktor 8 spatříme ve viresharku správu typu join-accept6.5.



Obrázek 6.5: Wireshark - OTAA join-accept

Hodnoty uvedené v disektoru viresharku pro nás teď nemají nijak velký význam jelikož celá zpráva až na MAC hlavičku je šifrována metodou AES ECB. Už od prvního pohledu působý zpráva velmi podivně obsahuje totiž ohromné množství nul a je poměrně velmi dlouhá. Délka však nemusí být problém zpráva může obsahovat list kanálů.

Pokusíme se tedy provést dešifrování této zprávy. Po dešifrování ovřešíme zda 7 až 10 bajt odpovídají adrese zařízení použité v následující zprávě. Bohužel se neschodují.

Závěrem tohoto experimentu bohužel je neúspěch při dekódování OTAA aktivace. I přes to, že předpoklady pro správné odchycení aktivace, jsem se snažili co nejvíce maximalizovat. Předem byla ověřena možnost dekódovat zprávy odeslané bránou principem přehození IQ vzorků. Pro další pokusy bychom potřebovali získat bránu, kterou by jsme mohly podle potřeb nakonfigurovat a také kontrolovat data přes ni procházející.

Kapitola 7

Závěr

7.1 Zhodnocení dosažených výsledků

V rámci práce se podařilo implementovat LoRaWAN sniffer s použití GNU Radia a především knihonvy gr-lora. Sniffer se dá pohodlně ovládat přímo s použitého sítového analyzátoru wireshark, který také zabezpečuje interpretaci zachycených dat. Podařilo se nám úspěšně zachytit a dekódovat zprávu odeslanou koncovým uzlem v ABP aktivační metodou. Při pokusech s OTAA aktivací jsme již tak úspěšní nebyly jelikož jsme nedokázali zachytit odpověď brány z důvodů příliš slabého signálu. S výsledným řešením jsme tedy schopni analyzovat komunikaci v síti LoRaWAN za podmínky, že přijímaný signál je výrazně silnější než šum.

7.1.1 Omezení použitého řešení

- Dekódování pouze signálů nad úrovní šumu
- Nefunkční dekódování rozkládacího faktoru 11 a 12
- Výpočetní náročnost při zachytávání s více parametry přenosu

7.2 Možné využiti a rozšíření

Vzhledem k dosaženým výsledkům je využití tohoto řešení velmi úzké. Pro plnohodnotnou analýzu LoRaWAN sítě jej můžeme využít jen za předpokladu že se koncové zařízení i brána nachází ve velmi malé vzdálenosti od našeho snifferu ideálně v rámci jedné místnosti. Toto omezení vylučuje použití k analýze komunikace v komerčních LoRaWAN sítích. Zařízení se tak hodí spíše pro využití při implementaci vlastní LoRaWAN sítě.

Z hlediska identifikace koncového zařízení jej můžeme použít k ověření nastavení aktivační metody a vysílací parametrů.

7.2.1 Rozšíření

Seznam možných rozšíření seřazený podle odhadované náročnosti jejich implementace.

- Přidání podpory pro další softwarově definovaná radia pr. RTL-SDR
- Přidání dalších metadat do struktury loratap v rámci knihny gr-lora

- Úprava knihovny gr-lora pro dekódování kódového slova
- Nalezení příčiny problému dekódování SF 11 a 12
- Vylepšení dekódování zpráv s nízkou sílou signálu

Přidání podpory pro další SDR do našeho extcap rozhraní je velmi jednoduché. Jediným důvodem proč jsme neimplementovaly do extcap rozhraní všechna běžně dostupná sdr je nutnost mít toto sdr kdyspozici pro testování.

Úpravy knihovny gr-lora by již tak jednoduché nebyly. Přidání dalších metadat by vyžadovalo větší seznámení se zdrojovými kódy této knihovny. Zbylé opravy a vylepšení by pak vyžadovali důkladné porozumění knihovně a nalezení způsobu detekování LoRa modulace pod hranicí šumu.

Literatura

[1] *Amateur Radio* [online]. [cit. 2019-11-30]. Dostupné z: https://www.pe0sat.vgnet.nl/sdr/iq-data-explained/.

[2] Co., N. W. T. *NiceRF LoRa1276 Product specification* [online]. [cit. 2020-03-01]. Dostupné z: https://www.nicerf.com/Upload/ueditor/files/2020-03-05/LORA1276-100mW%20long%20range%20Spread%20Spectrum%20modulation%20wireless%20transceiver%20module%20V2.2-bdc587bf-32c2-413c-8e4f-a309a8a00abd.pdf.

[3] Dobeš, J. *Moderní radiotechnika*. 1. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-7300-132-2.

[4] KNIGHT, B. *Decoding LoRa: Realizing a Modern LPWAN with SDR* [online]. 2015 [cit. 2020-01-02]. Dostupné z: https://pubs.gnuradio.org/index.php/grcon/article/view/8/7.

[5] LORA ALLIANCE, I. *LoRaWAN 1.1 Specification* [online]. 2017 [cit. 2020-01-01]. Dostupné z: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[6] LORA ALLIANCE, I. *LoRaWAN 1.0.3 Specification* [online]. 2018 [cit. 2019-12-21]. Dostupné z: https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf.

[7] matthijskooijman. *Arduino-LMIC library* [online]. Dostupné z: https://github.com/matthijskooijman/arduino-lmic.

[8] myriadrf. *LimeSDR Made Simple Part 1: Introduction* [online]. Dostupné z: hhttps://myriadrf.org/news/limesdr-made-simple-part-1/.

[9] myriadrf. *LimeSDR-USB hardware description* [online]. Dostupné z: https://wiki.myriadrf.org/LimeSDR-USB_hardware_description.

[10] Prokeš, A. *Softwarově definované rádio* [online]. Dostupné z: http://www.urel.feec.vutbr.cz/web_pages/projekty/clanky/Prokes_SW_radio.pdf.

[11] sandeepmistry. *Arduino LoRa* [online]. Dostupné z: https://github.com/sandeepmistry/arduino-LoRa.

[12] SEMTECH. *An In-depth Look at LoRaWAN® Class B Devices* [online]. [cit. 2020-2-06]. Dostupné z: https://lora-developers.semtech.com/library/tech-papers-and-guides/lorawan-class-b-devices/.

[13] SEMTECH. *What are LoRa® and LoRaWAN®?* [online]. [cit. 2020-2-05]. Dostupné z: https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/.

[14] SEMTECH. *Software Whitening and CRC on SX12xx Devices* [online]. 2013 [cit. 2019-12-01]. Dostupné z: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000HSOn/5O4_iba4ULdi6rfa1q7oENCQ9LGFWGwTjS32Loibdoo.

[15] SEMTECH. *LoRa™ Modulation Basics* [online]. 2015 [cit. 2019-12-05]. Dostupné z: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Ju/xvKUc5w9yjG1q5Pb2IIkpolW54YYqGb.frOZ7HQBcRc.

[16] WIKIPEDIA. *Analog-to-digital converter* [online]. [cit. 2019-12-10]. Dostupné z: https://en.wikipedia.org/wiki/Analog-to-digital_converter.

[17] WIKIPEDIA. *Antenna (radio)* [online]. [cit. 2019-12-10]. Dostupné z: https://en.wikipedia.org/wiki/Antenna_(radio).

[18] WIRESHARK. *Extcap - The Wireshark Network Analyzer 3.2.5* [online]. [cit. 2020-6-08]. Dostupné z: https://www.wireshark.org/docs/man-pages/extcap.html.

[19] WIRESHARK. *Wireshark Developer's Guide* [online]. Dostupné z: https://www.wireshark.org/download/docs/developer-guide.pdf.