

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra telekomunikační techniky

Přehledový přijímač / monitor rádiových sítí IoT

Ondřej Šulc

Školitel: Ing. Pavel Troller, CSc.
Obor: Komunikační systémy a sítě
Leden 2019

Poděkování

Děkujeme ...

Prohlášení

Fakt sám ...

Abstrakt

Rozvíjíme ...

Klíčová slova: IoT, SDR-RTL, LoRa, Sigfox, Přehledový přijímač

Školitel: Ing. Pavel Troller, CSc.
Pestitelský ústav,
Zárivá 232,
12000 Praha 2

Abstract

We develop ...

Keywords: IoT, SDR-RTL, LoRa, Sigfox, Scanner

Title translation: Scanner/Monitor of IoT radio networks

Obsah

1 Úvod	1	6.2 Architektura	46
Část I		6.3 Třídy IoT technologií	46
Seznámení s problematikou		6.4 kontrolní aplikace	46
2 Internet věcí	5	6.4.1 webserver.py	46
2.1 Rozdělení technologií pro IoT ...	5	6.4.2 index.html	46
2.1.1 Dle využívaných frekvencí	5	7 Výsledky	47
2.1.2 Dle architektury sítě	6	8 Závěr	49
2.1.3 Dle rozsahu sítě	7		
3 Softvérově definované rádio	9		
3.1 Úvod do SDR	9		
3.2 Fungování SDR	9		
3.3 RTL-SDR	12		
3.4 Další dostupná SDR	13		
4 Technologie IoT	17		
4.1 LoRa	17		
4.1.1 Obecné informace	17		
4.1.2 Specifikace	18		
4.1.3 Pokrytí	18		
4.1.4 Fyzická vrstva (LoRa PHY) .	19		
4.1.5 Softwarová demodulace	23		
4.2 SigFox	26		
4.2.1 Obecné informace	26		
4.2.2 Specifikace	27		
4.2.3 Pokrytí)	27		
4.2.4 Rádiový protokol Sigfox)	28		
Část II			
Přehledový přijímač			
5 Hardware	37		
5.1 RTL-SDR	37		
5.2 Raspberry Pi	38		
5.2.1 Displej	38		
5.3 Napájení	38		
5.4 Kryt	40		
5.4.1 Autodesk Fusion 360	40		
5.4.2 Model	41		
5.4.3 Tisk	42		
5.5 Zapojení	42		
6 Software	43		
6.1 Požadavky a z nich vyplývající software	43		
6.1.1 librtlsdr a její utility	43		
6.1.2 GNU Radio	44		
6.1.3 Flask a Flask-socketIO	46		

Obrázky

3.1 Implementace rádia na základě SDR[11]	10
3.2 Klasifikace SDR dle místa konverze domény[andr\IeC {\a}sretzler2014]	11
3.3 Blokové schéma SDR s přímou kvadraturní konverzí RF na BB[19]	12
4.1 Specifikace LoRaWAN v jednotlivých regionech [5]	19
4.2 Pokrytí technologií LoRaWAN ve světě [4]	20
4.3 Pokrytí LoRaWAN Českými radiokomunikacemi [26]	21
4.4 Ukázka signálu lora [16] (upraveno)	22
4.5 Diagonální prokládání při $FS = 7$ a $CR = 4$, hodnota čipu třetího symbolu je vyznačena tučně. [28] .	23
4.6 Vlevo synchronizace pomocí Schmidl-Cox algoritmu a vpravo algoritmus použitý v [28]	24
4.7 Pokrytí Sigfox ve světě[33]	28
4.8 Pokrytí Sigfox v České republice [24]	29
4.9 Příklad modulace DBPSK [18] .	29
4.10 Modulace amplitudy při změně fáze [9]	30
4.11 Struktura rámce Sigfox pro Uplink (nahore) a konkrétní podoba rámce OOB (dole). Každá buňka reprezentuje půslabiku (4 bity) [9]	31
5.1 Výhody RTL-SDR blog V3 oproti běžným RTL DVB-T [30]	39
5.2 Prostředí Autodesk Fusion 360 a navrhnutý model krytu	41
5.3 Diagram zapojení a dostupnosti portů přehledového přijímače	42
6.1 Flowgraf pro příjem LoRa	45
6.2 Flowgraf pro příjem Sigfox	46

Tabulky

3.1 Přehled dospuných SDR	15
4.1 Parametry přenosu Sigfox pro Uplink a Downlink	28
4.2 Hodnoty F.TYPE v závislosti na délce zprávy a pořadí opakování [9]	31
4.3 Pole rámce Downlinku (délka pole v bitech)	33



Kapitola 1

Úvod

Nejlepší práce na světě



Část I

Seznámení s problematikou

Kapitola 2

Internet věcí

Internet věcí je velmi rychle se rozvíjející trend v oblasti telekomunikační techniky. Vzhledem k snaze propojit do sítě stále více a více zařízení, která mají často specifické požadavky je potřeba vytvořit nové technologie, které by lépe vyhovovali tomuto použití. Tento proces již započal a na trhu existuje velké množství variant. Každá technologie je uzpůsobená pro určité způsoby využití, a tak se v mnohém liší. Mají však i společné atributy vyplývající z obecných požadavků pro IoT.

Je potřeba aby takové sítě byly velkokapacitní a mohly tak obsloužit velké množství zařízení. Dle odhadů bylo do sítě IoT v roce 2017 připojeno již přes dvacet miliard zařízení a jejich počet zatím bude nadále prudce stoupat. [36]

I přesto že povětšinou se jedná o zařízení generující jen malý síťový provoz, celkový provoz sítě je díky jejich velkému množství obrovský.

Dále je, vzhledem k motivaci připojit zařízení co nejvíce, nutné, aby v síti mohla pracovat i velmi jednoduchá zařízení, která jde snadno a levně vyrobit. Jen tak bude možné vybudovat rozsáhlé senzorové sítě o tisících prvků nebo připojit opravdu většinu zařízení každé domácnosti.

Posledním univerzálním požadavkem, kterým se IoT od běžných sítí odlišují jsou velké nároky na energetickou úspornost. Připojená zařízení totiž musí často fungovat nezávisle na rozvodných sítích elektrické energie, a tak musí spoléhat na akumulátory. Ty mají omezenou velikost a kapacitu a často je jejich výměna či dobítí nepraktické nebo dokonce nemožné. I přesto však musí být podobná zařízení schopná provozu i několik let.

Dale povidani o tom proc je dulezite do IoT integrovat SDR (rychly vyvoj, velka kompativibilita, univerzalni gateway, cognitive radio)

2.1 Rozdělení technologií pro IoT

2.1.1 Dle využívaných frekvencí

Velkým faktorem při výběru technologie pro použití pro účely internetu věcí jsou používané frekvence. Rádiové spektrum je totiž omezeným zdrojem, a tak jsou pro volné použití vyhrazeny jen určité frekvence, a i pro vysílání v těchto bezlicenčních pásmech existují omezení a pravidla. Další pásma lze využívat na základě licence od Českého telekomunikačního úřadu. Sítě pro

Výhodou bezlicenčních pásem je především jednoduchost a cena nasazení. Není třeba žádat o žádné licence, čekat na jejich schválení, platit za ně. Na druhou stranu je pak ale nutné počítat s tím, že na stejných frekvencích může vysílat i kdokoli jiný a způsobovat tak vzájemnou interferenci. Dále je nutné vždy dodržovat závazná pravidla pro použití dané frekvence. To se týká zejména maximálního vysílaného výkonu, což může ohrozit požadovaný dosah. Dalším omezením často bývá maximální povolená doba využívání daných prostředků. Můžeme tak například být omezeni počtem zpráv které je možné za určitý časový úsek přenést.

Náklady na provoz sítě pracující v licencovaném pásmu pak bývají větší, protože se do nich musí započítat poplatky za licenci. Na druhou stranu pak má provozovatel mnohem větší svobodu s nakládáním s daným pásmem a může tak mít vysílače s vyšším výkonem a nemusí se bát rušení od jiných zařízení, než od těch jejichž provoz sám v síti povolí.

Nejčastější topologií bývá hvězda, její název je odvozen od tvaru hvězdy, kdy uprostřed je vždy centrální prvek a z něho se rozebíhají paprsky ke všem připojeným zařízením. Ta jsou propojena jen s centrálním prvkem, a ne spolu navzájem. Výhoda této topologie je, že pokud selže nebo se odpojí jedno z připojených zařízení nezpůsobí to žádné komplikace pro ostatní. Dále je v této topologii přítomný řídicí prvek, který má většinou řádově vyšší výpočetní kapacitu a dostupné zdroje a může tak síť řídit a jednotlivé uzly mohou být o to jednodušší a úspornější. To také nahrává modelu, kdy provozovatel vytvoří pokrytí určitého území svou technologií a uživatelé se již nemusí o nic starat a jen za poplatek využívají v již vytvořené síti svá koncová zařízení. Na druhou stranu v případě selhání centrálního prvku přestává fungovat celá síť. To lze samozřejmě řešit redundancí takového prvku, avšak to dále přispívá, k již tak vysokým nákladům na budování infrastruktury pro síť tohoto typu.

komunikuje s vnějším světem.

■ 2.1.3 Dle rozsahu sítě

Správnou technologii budeme posuzovat i na základě rozsahu. Naše požadavky se zásadní promění v závislosti na tom, zda vytváříme mezinárodní síť pokrývající velké území nebo chceme pokrýt jen jednu domácnost. Pro pojmenování rozsahu sítě jsou standardně používány tyto termíny:

PAN (Personal area network) Tento akronym popisuje síť, která zahrnuje osobní prostor člověka. V dnešní době to může znamenat spojení zařízení jako mobilní telefon, chytré hodinky, sluchátka, hrudní pás nebo i senzor cukru v krvi. Často se pro tento typ sítí používají technologie jako Bluetooth, Ant, MiWi nebo Zigbee.

LAN (Local area network) Místní síť je název pro síť, která je většinou fyzicky omezena jednou budovou, může to být domácnost, škola nebo menší firma. Typickými zástupci bezdrátových technologií této kategorie jsou WiFi a Z-Wave.

MAN (Metropolitan area network) Tento druh sítě propojuje zařízení v rámci větších oblastí jako jsou města a univerzitní kampusy. V tomto rozsahu jsou typickými zástupci WiMax a IQRF.

WAN (Wide area network) Síť pokrývající velká území, například celé státy. Zástupci z této kategorie jsou LoRaWAN, Sigfox nebo NB-IoT.

Kapitola 3

Softwérově definované rádio

3.1 Úvod do SDR

Značná část historie použití rádiových vln využívala rádia, která byla implementována čistě v hardware. Pro každou úpravu vlastností rádia, tak bylo zapotřebí ho fyzicky upravit. S rozmachem výpočetní techniky však přišel vývoj i v oblasti rádia a na světlo světa se tak nejdříve dostala SCR (Software controlled radio), která umožňovala v omezené míře ovlivňovat některé funkce a později i rádia SDR, která mohou být použita univerzálně díky digitálnímu zpracování signálu (DSP -Digital signal processing).

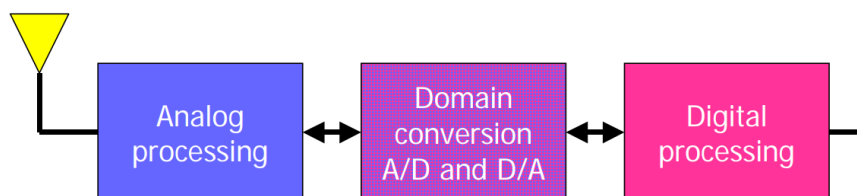
První známe nasazení SDR v praxi bylo v armádě, jednalo se o systém SPEAKEasy. Systém vyvinula DARPA v roce 1991 a umožňoval díky softwarové implementaci komunikovat prostřednictvím 10 různých vojenských protokolů na frekvencích od 2 MHz po 2 Ghz. SPEAKEasy byl připraven i na přidání dalších modulací a protokolů. O rok později již vyšel první článek o SDR na IEEE, napsal ho Joe Mitola a je tak považován za kmotra SDR. [21]

I přes tyto slibné začátky a podchycení jeho teoretických možností se SDR rozšiřovalo poměrně pomalu a jeho význam narostl až v podlední době s rozmachem levných výkoných integrovaných čipů. Velkou motivací využití SDR do budoucna jsou tzv. kognitivní rádia, která se dokáží přizpůsobit aktuálnímu stavu spektra.

3.2 Fungování SDR

Základním znakem každého SDR je, že je značná jeho část realizována jako software běžící na programovatelném a konfigurovatelném HW zařízení. Toto zařízení můžeme nazvat rádiovou platformou (radio platform) a SW část jako aplikační rámec (application framework). Pokud je použitý jednotný standard tak tyto části mohou tvořit univerzální a znovupoužitelné komponenty a tak ušetřit prostředky a usnadnit jakékoliv upgrady.

Jedním z možných standardů je armádou používaná Softwarová komunikační architektura (Software Communication Architecture) vyvinutá v programu JTRS (Joint Tactical Radio System). Tuto architekturu používá většina armádních SDR. [11] Jinou možností je de facto standard pro rádio amatéry -



Obrázek 3.1: Implementace rádia na základě SDR[11]

GNU Rádio, kterému je v této práci věnována samostatná kapitola ??.

Pokud na rádio nechceme koukat jako na krabičku ke které se připojí anténa, můžeme identifikovat jednotlivé funkční bloky:

1. Převod z RF (rádiová frekvence) do IF (mezifrekvence)
2. Převod do základního pásma
3. Demodulace
4. Uživatelské rozhraní

Pokud se jedná o SDR můžeme ale zvolit i jednodušší vysokoúrovňový pohled a identifikovat tak tři prvky modelu: Zpracování analogového signálu (front-end), konverze domény (A/D, D/A) a digitální zpracování signálu (back-end), viz obrázek 3.1

Fron-end se stará o převedení rádiového signálu na frekvenci a šířku pásu, kterou dokáže zpracovat digitální back-end. Tvoří ho analogové zesilovače, směšovače filtry a oscilátory.

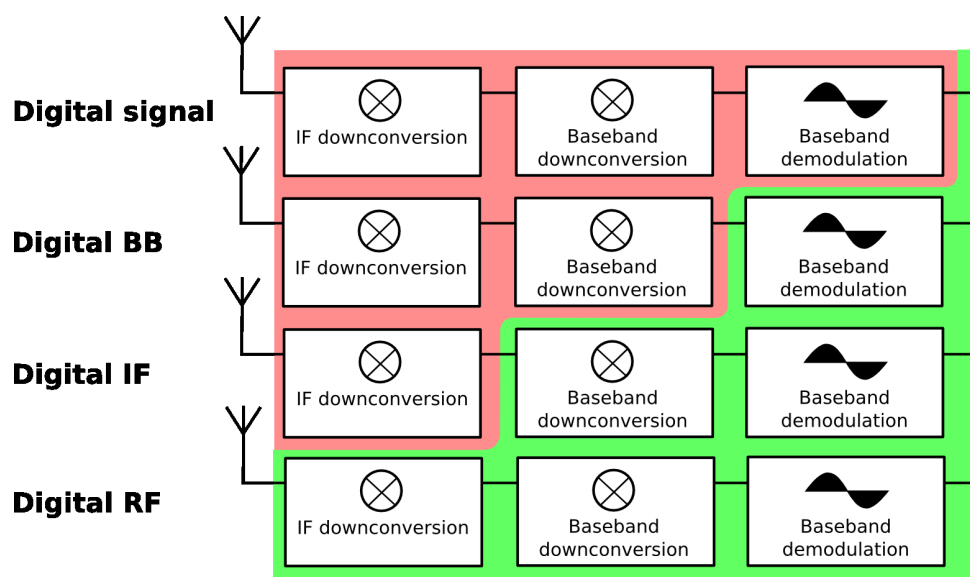
Konvertor domény jak již název napovídá převádí analogový signál na digitální a obráceně. Souží mu k tomu vysokorychlostní širokopásmové A/D a D/A převodníky, jejich vlastnosti zásadně ovlivňují možnosti výsledného SDR.

Poslední prvek tedy backend kde probíhá digitální zpracování je založen na FPGA a/nebo DSP programovatelných počítačích, na kterých běží software. Převod mezi analogovou a digitální doménou se může odehrát v různých místech zpracování viz obrázek 3.2 a podle toho lze systémy kategorizovat.

Digitální signál (Digital signal) V toto případě se v podstatě nejedná o SDR, vše je implementováno v HW. Výstupní signál je však digitální.

Digitální základní pásmo (Digital Baseband) Zde se již část zpracování signálu odehrává v SW. Signál v základním pásmu je navzorkován a modulace se tak odehrává pomocí DSP. Podobné systémy používají rádio amatéři například pro příjem BPSK31 pomocí rádiového přijímače a zvukové karty počítače na kterém běží SW pro demodlaci.

Digitální mezifrekvence (Digital IF) V této kategorii probíhá vzorkování již na mezifrekvenci a předchází mu konverze z rádiové frekvence



Obrázek 3.2: Klasifikace SDR dle místa konverze domény [andr\IeC {\a}sretzler2014]

a filtrování implementované v HW. Podobné systémy jsou využívány radioamatéry nebo například v námořních rádiích, DSP zde slouží k filtrování šumu a demodulaci.

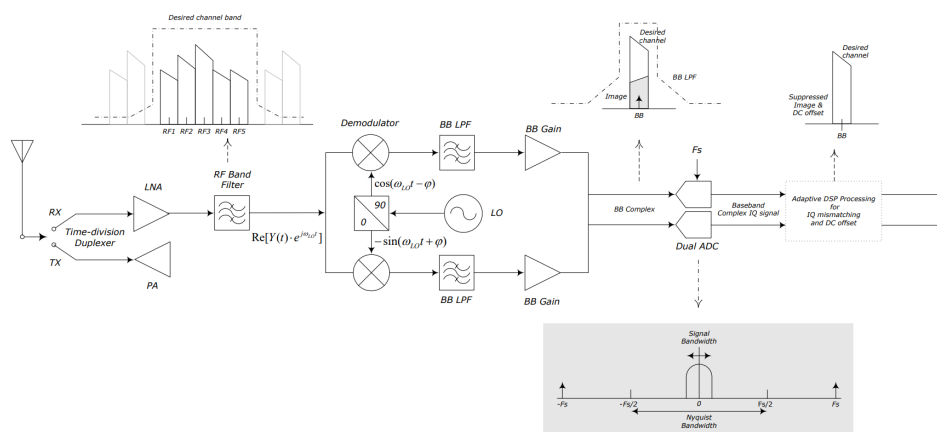
Digitální rádiová frekvence (Digital RF) I při vzorkování přímo RF je potřeba signál nejdříve zesílit a filtrovat až poté je možný jeho převod pomocí ADC. Následně probíhají všechny úkony již v SF. Příkladem systému z této kategorie může být HPSDR Mercury [38].

Toto rozdělení však nepočítá s přímou kvadraturní konverzí RF kde je vynechán mezikrok převodu na IF a převádí se rovnou na BB. V takovém případě se RF signál s reálnými hodnotami po zesílení a filtraci smíchá s výstupem oscilátoru s komplexním výstupem (sínus a kosínus), prožene filtrem dolní propust který odstraní vysoké frekvenční komponenty jako boční pásmo vzniklé smíšením a poté navzrokuje dvěma ADC. [andr\IeC {\a}sretzler2014]

Popsaný postup je výhodný zejména díky své jednoduchosti, není potřeba filtrování IF a cena tak může být nižší. Další výhoda spočívá v tom, že Nyquistova vzorkovací frekvence pro komplexní I/Q signál je dvojnásobná oproti reálnému signálu a tak je možné vzorkovat mnohem širší pásmo.

Mezi nevýhody a také důvody proč se dříve používali složitější systémy patří možnost projevení se zrcadlového obrazu signálu a stejnosměrné složky v přijatém signálu. Zrcadlový obraz je způsoben rozdílem ve fázi a amplitudě mezi I a Q kanály. Tyto vady výrazně degradují přijatý signál avšak je možné je efektivně eliminovat korekcí rovnováhy IQ v digitálním zpracování.

Tento systém jsem takto podrobně rozepsal zejména protože ho používá většina dostupných SDR pro radio amatéry včetně RTL-SDR použitého v této práci.



Obrázek 3.3: Blokové schéma SDR s přímou kvadraturní konverzí RF na BB[19]

Možnost přidat výhody a nevýhody <http://www.winradio.com/home/facts.htm>

3.3 RTL-SDR

Pokud si v minulosti chtěl někdo pořídit SDR musel počítat s částkami mnoha tisíc nebo i desítek tisíc korun a musel mít zároveň mít k dispozici velmi výkonný HW na kterém provádět výpočty. To se změnilo v roce 2012, kdy na scénu přišlo RTL-SDR - původně DVB-T USB tuner, který však s alternativními ovladači může být použit jako SDR a dá se pořídit za cenu do 500 Kč.

Celé to začalo v roce 2008, kdy Realtek představil chipset RTL2832U s DVB-T COFDM demodulátorem, hlavním účelem tohoto čipu a na něm postavených USB donglů byl příjem evropského standardu pro pozemní digitální televizní vysílání. Nabízel však i příjem digitálního rádia DAB a analogového FM. Tento malý detail se později stal velmi podstatným.

V roce 2010 totiž Eric Fry při pokusech o napsání ovladačů tohoto DVB-T donglu odposlouchával USB pakety během používání FM aplikace pro Windows. Eric zjistil, že narozdíl od příjmu DVB-T, kdy demodulace probíhá přímo na čipu, při příjmu FM, DAB a DAB+ demodulace probíhá až v SW počítače a přes USB se tedy přenáší I/Q vzorky, jeho primárním cílem však byli ovladače pro Linux a tak nezačal s vývojem pro použití jako SDR.

Informace se sice rozšířila, ale až do roku 2012 nenastal žádný větší pokrok. V tomto roce se o tento DVB-T přijímač od Realteku začal zajímat Antti Palosaari. Ten potvrdil možnost využití jako SDR díky přístupu k 8-bitovým I/Q vzorkům a začal s vývojem potřebného SW. Díky navázání spolupráce s organizací Osmocom, která se v té době vyvíjela vlastní SDR založené na tuneru E4000, což byl jeden z tunerů používaných v DVB-T přijímačích

Realtek, práce nabrala obrátek a brzy byl vyvinut driver SW pro jednoduché použití těchto USB donglů jako SDR. Nejdůležitějším krokem bylo nejspíše vyvinutí ovladače pro linux, o to se v organizaci Osmocom postaral Steve Markgraf. [29] [8]

Tento počín nastartoval éru RTL-SDR a na jeho základě tak vzniklo obrovské množství projektů všeho druhu, jmenovat budu jen pár těch zajímavějších (pokud čtete tu práci ve formátu PDF můžete kliknutím na projekt přejít na stránky, které se mu věnují):

Spektrální analyzátor, Odposlech GSM, Sledování letadel pomocí ADSB, Generátor nahodných čísel, Příjem dat meteorologických balónů, Sledování meteoritů, nebo Odposlech vysílaček městské policie

3.4 Další dostupná SDR

RTL-SDR je sice i v současné době to nejlevnější SDR, jeho kvalita však může být pro mnohá použití nedostatečná a tak v posledních letech vzniklo mnoho dalších dostupných SDR. Ta jsou sice o něco dražší, ale díky tomu, že jde o HW pro použití jako SDR navržený, mají větší rozsah, rozlišení, vzorkovací frekvenci a některá mohou kromě příjmu i vysílat.

USRP (Universal Software Radio Peripheral) USRP je celá řada produktů od společnosti Ettus Research. Hlavním vývojářem je Matt Ettus a první USRP spatřilo světlo světa již v roce 2008. V současnosti začínají ceny okolo 20 tisíc korun a v základu má USRP rozsah od 70 MHz po 6 GHz, tento rozsah však lze rozšířit pomocí přídatných desek. Starší hardware je uvolněn jako open-source a veškeré ovladače také. Z těchto důvodů jsou rádia USRP populární ve vědě, na univerzitách i mezi amatéry. USRP je kromě jiných podporované v GNU-Radio a Matlab SimuLinku a kromě příjmu zvládá i vysílat. [27]

bladeRF Již v roce 2012 v reakci na RTL-SDR začal vývoj bladeRF. Momentálně se nachází již ve své druhé verzi a zajímavostí je, že jedna z variant obsahuje i FPGA pro HW akceleraci digitálního zpracování. Cena začíná na 11 tisících za verzi bez FPGA, verze s FPGA stojí 16 tisíc korun. [20]

HackRF Toto SDR vyvinuté Michaelem Ossmanem v roce 2013 se díky úspěšné kampani na Kickstarter.com začalo prodávat v roce 2014 a jeho cena je cca 7000 Kč. Michael Ossman se již dříve proslavil vývojem zařízením pro odposlech bluetooth Ubertooth a HackRF navazuje jako další velmi kvalitní produkt. HackRF dokáže vysílat i přijímat a má rozsah má od 30 MHz po 6 GHz a zvládá i 2x2 MIMO. [22]

AirSpy AirSpy je asi nejlevnější alternativa k RTL-SDR a podobně jako ono neumí vysílat. Velkou popularitu AirSpy získalo zejména díky úzkému propojení s SDR# což je software pro analýzu a demodulaci rádiových

signálů. V nejnovější verzi má rozsah od 24 MHz po 1,8 GHz a podporuje i externí hodiny pro použití vyžadující synchronizaci. [3]

Název	USRP B205mini	bladeRF 2.0 Micro	HackRF One	AirSpy R2	RTL-SDR
Cena (Kč)	20 000	11 000	7000	5000	500
Frekvenční rozsah (MHz)	70-6000	47-6000	1-6000	24-1700	24-2200
ADC rozlišení (bits)	12	12	8	12	8
Šířka pásma (MHz)	56	56	20	10	16
Možnost vysílání	Full Duplex	Full Duplex 2x2 MIMO	Half Duplex	Ne	Ne
Přesnost hodin (PPM)	2	0,26	30	0,5	85
FPGA (kLE)	Ne (dražší verze Ano)	49	Ne jen CPLD	Ne	Ne

Tabulka 3.1: Přehled dostupných SDR

[27] [2] [13] [15]

Kapitola 4

Technologie IoT

Po předchozím průzkumu provedeném v rámci předmětu B2MPROJ6 - Projekt jsem se rozhodl zaměřit na dvě technologie pro IoT, jedná se o LoRaWAN a Sigfox. Tyto technologie jsem zvolil, protože jako jediné mají rozvinutou infrastrukturu v ČR a zároveň vysílají v pásmu, které se nachází v rozsahu RTL-SDR, které jsem plánoval pro přehledový přijímač použít.

V následující kapitole tyto technologie představím jak z hlediska obecného, tak z hlediska rádiové vrstvy.

4.1 LoRa

4.1.1 Obecné informace

LoRaWAN je klasická ukázka Low-Power Wide-Area Network (LPWAN) navržené pro prostředí internetu věcí. Vyniká tedy hlavně v oblasti životnosti na baterii, dosahu, ceně a celkové kapacitě sítě.

Každá implementace této technologie jako LPWAN se skládá ze dvou částí. LoRaWAN definuje komunikační protokol a síťovou architekturu a LoRa zajišťuje fyzickou vrstvu přenosu [4].

LoRa (Long Range) je fyzická vrstva používaná v sítích LoRaWAN. Je, jak již název napovídá, navržena pro komunikaci na velké vzdálenosti a na rozdíl od mnohých starších systémů, které využívají klíčování frekvenčním posunem (FSK), používá chirp rozprostřené spektrum [5]. Díky tomu je možné pomocí jen několika málo bran nebo základových stanic pokrýt velká území. Například celé město jednou jedinou branou. Z toho plyne možnost velmi rychle, s malými náklady a infrastrukturou pokrýt území celé země.

LoRaWAN definuje komunikační protokol a síťovou architekturu. Zvolená topologie síťové architektury je hvězda. To má oproti smíšené topologii výhodu v tom, že každý uzel sítě přenáší jen svoje data a nemusí tak využívat své zdroje pro přenos pro něj irelevantních informací. Zároveň je taková síť jednodušší na správu a má vyšší celkovou kapacitu. Na druhou stranu je dosah omezen na dosah jednoho uzlu.

Takováto architektura sítě by se dala připodobnit klasické buňkové mobilní síti, ale na rozdíl od ní není nikdy uzel přiřazen specifické základové stanici, při pohybu uzlu tedy odpadá handover. Místo toho paket z každého uzlu při-

jmou všechny brány v dosahu a předají ho na společný server. Ten vyhodnotí přijaté pakety, duplikované zahodí, zvolí ideální bránu pro odpověď, adaptivní přenosovou rychlost a podobně.

Další úspora baterie spočívá v časování komunikace. Uzly v síti nejsou synchronizované s bránami. K řízení provozu je použit protokol ALOHA. Zařízení tak vysílají jen pokud mají data k odeslání nebo v naplánovaných intervalech a nemusí se budit pro udržování synchronizace. Pro komunikaci směrem k uzlům existují tři způsoby podle tříd zařízení. Třída A má pro downlink definovaná okna jen vždy po ukončení uplinku. Třída B má navíc naplánovaná další periodicky se opakující okna, kdy je zařízení připraveno přijímat a třída C určená zejména pro zařízení, pro která není energetická náročnost tak kritická, je připravena na příjem kontinuálně.

Pro zachování velké celkové kapacity i při poměrně malém počtu bran musí tyto být schopné přijímat data od mnoha uzlů zároveň. Toho je docíleno využitím adaptivních rychlostí přenosu a vícekanálovým a vícemodemovým vysílačem. Díky použití rozprostřeného spektra lze také využít ortogonalitu signálů používajících jiný činitel rozptření. Pokud by ani tak celková kapacita nestačila, je technologie LoRaWAN velmi dobře horizontálně škálovatelná. Přidáním dalších bran v oblasti mohou zařízení vysílat menším výkonem a vyšší přenosovou rychlostí a tím omezit přeslechy a kapacita tak může být rozšířena 6-8krát.

I přes důraz na nízkou energetickou náročnost LoRaWAN řeší i bezpečnost provozu. Na síťové úrovni je řešena autenticita uzlů a bran a na aplikační úrovni je použito AES šifrování pro skrytí dat před neoprávněným odposlechem.

■ 4.1.2 Specifikace

- modulace: CSS - Cvrlikavé rozptřené spektrum
- maximální velikost zprávy: 256 Bajtů
- počet zpráv za den: neomezený
- citlivost: -140dBm
- dosah v terénu: až 40km v terénu, 15 km v příměstském prostředí a 2 – 5 km ve městě
- zabezpečení: šifrování AES128
- výdrž na bateriích: 5-15 let (podle hustoty komunikace) [23]

■ 4.1.3 Pokrytí

Vzhledem k tomu, že LoRa aliance je otevřená a sestává se ze stovek členů zahrnujících mezinárodní telekomunikační operátory i výrobce techniky a senzorů a její účel je vytvořit mezinárodní síť s možností roamingu zařízení, je pokrytí téměř celosvětové. Sama aliance na svých stránkách publikuje tuto mapu 4.2. V Česku spravuje komerční LoRaWAN síť operátor České Radiokomunikace a pokrývá většinu území, viz ??

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 + 8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kbps				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Obrázek 4.1: Specifikace LoRaWAN v jednotlivých regionech [5]

4.1.4 Fyzická vrstva (LoRa PHY)

Následující část se zabývá protokolem LoRa. Vzhledem k tomu, že je protokol LoRa proprietární a neexistuje k němu tak veřejně dostupná dokumentace, vycházím zejména ze dvou prací [16] [28]. Tyto práce napsali výzkumníci, kteří pomocí reverzního inženýrství protokol analyzovali. Kromě reverzního inženýrství disponovali také zdroji ze stránek vlastníka technologie Semtech, které již v tuto chvíli nejsou k dispozici.

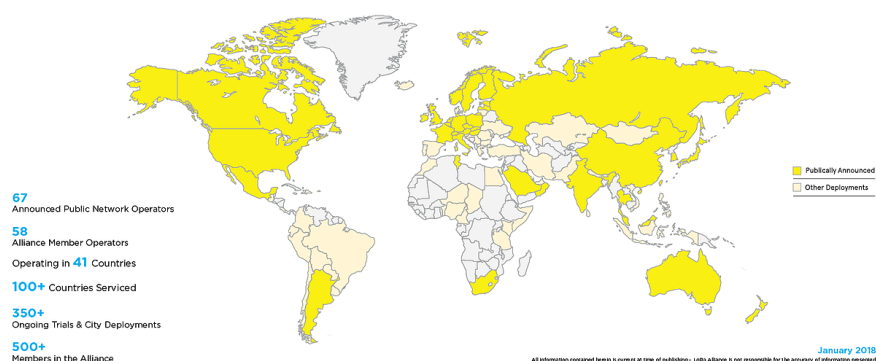
Modulace

Modulační schéma LoRa je založeno na Chirp Spread Spread Spectrum (Cvrlikající rozprostřené spektrum) modulaci [12] a definuje jeden “cvrk” jako jeden symbol. Cvrk lze popsat jako signál, který v průběhu svého trvání lineárně mění svoji frekvenci a to buď na nižší nebo na vyšší a rozlišujeme tak cvrk nahoru adolů.

Část spektra kterou cvrk zabývá je jeho šířka pásma, v Evropě to může být 125 kHz nebo 250 kHz. To kolik kolik daný symbol/cvrk nese dat určuje činitel rozprostření, ten nabývá hodnot 7-12 bitů. Rychlost jakou se frekvence cvrku mění (první derivace frekvence) je dána vztahem $k = \frac{BW}{2^{SF}}$ kde k je rychlost cvrku, BW je šířka pásma a SF činitel rozprostření.

Celkový počet možných symbolů je dán vztahem 2^{SF} a konkrétní hodnota na cvrk modulována pomocí časového posuvu cvrku. To znamená, že cvrk začíná na jiné frekvenci a ve chvíli kdy dojde na konec šířky svého pásma začíná a okamžitě znovu druhé hranice pásma. Hodnota tohoto posuvu tak určuje hodnotu symbolu a ten je tak v podstatě kvantován na 2^{SF} časových

LoRaWAN™ NETWORKS



Obrázek 4.2: Pokrytí technologií LoRaWAN ve světě [4]

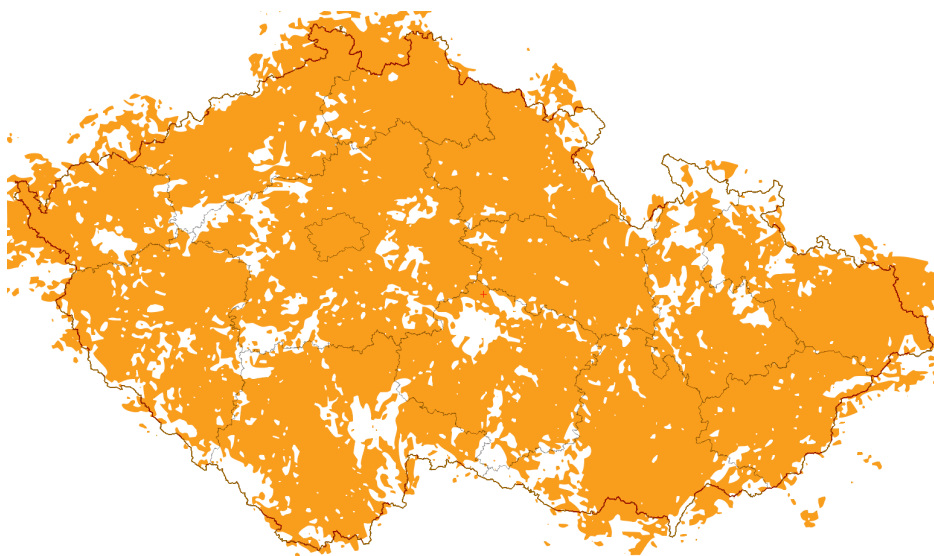
intervalů, těmto intervalům říkáme čipy.

■ Prokládání

Jako v každé jiné modulaci, musíme i zde počítat s chybami způsobenými šumem, interferencí, a časovými nebo frekvenčními posuny. Tyto chyby mohou způsobit, že hodnota čipu nebude dobře odečtena z modulovaného symbolu. Například poryv šumu může posunout vrchol v FFT spektru na jinou hodnotu čipu a tak jej znehodnotit.

Aby bylo možné minimalizovat dopad poryvů šumu na chybu jen jednoho bitu v symbolu je použito prokládání. Několik chipů je dohromady vepsáno do mřížky $\{0, 1\}^{SF \times (4+CR)}$, kde CR (Coding Rate) značí počet paritních bitů a nabývá hodnot 1 až 4. Pokud tedy bude použit $SF = 7$ a $CR = 4$ dostaneme matici $\{0, 1\}^{7 \times 8}$, příklad je na obrázku 4.5. K získání kódového slova je pak potřeba číst bity po diagonále matice. Na rozdíl od patentu LoRy, kde se uvádí, že směr diagonálního čtení bitů z mřížky je směrem dolů, v praxi lze pozorovat opačný směr. Tímto způsobem tak první chip obsahuje všechny nejméně významné bity (LSB - Least significant) všech kódových slov, druhý čip všechny druhé bity všech slov a tak dále. Díky tomu v případě ztráty celého čipu dojde k chybě jen v jednom bitu na kódové slovo.

Dalším způsobem jak zvýšit odolnost proti rušení vysílání je použití módu redukované rychlosti (reduced rate mode). V případě použití tohoto módu jsou první dvě řady prokládací matice zahozeny a její rozměr se tak změní na $\{0, 1\}^{SF-2 \times (4+CR)}$ což způsobí, že z ní následně vyčteme o dvě kódová slova méně. Zahozené řádky obsahují nejméně významné bity chipů, které jsou náchylnější k chybám protože odpovídají užším frekvenčním intervalům v FFT spektru. Z toho vyplývá, že mód redukované rychlosti obětuje rychlost přenosu dat ve prospěch odolnosti proti šumu. Hlavička fyzické vrstvy LoRa



Obrázek 4.3: Pokrytí LoRaWAN Českými radiokomunikacemi [26]

je v tomto módu vysílána vždy, kdežto užitečná data je v případě použití SF 11 nebo 12.

■ Kódování

Po přečtení kódových slov z prokládací matice mají tato délku $4 + CR$. Kvůli zamezení vzniku stejnosměrné složky byla slova v části rámce s užitečnými daty XOR-ována 9-bitovým lineárním posuvným registrem se zpětnou vazbou (LSFR Linear feedback shift register) (whitening). A proto musí po synchronizaci projít stejným procesem znovu. Přesný algoritmus není v patentu určen a jeho výběr je tedy na každém výrobci zvlášť.

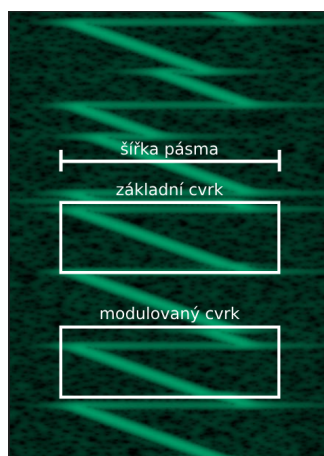
Na několika testovacích zařízeních [28] reverzním inženýrstvím zjistilo použití upraveného $4/(4 + CR)$ Hammingova kódu. Ve výsledku tak z každého kódového slova po dekódování získáme 4 bity dat. Ty jsou pak naparsovány do struktury rámce lora.

■ Struktura rámce

Na fyzické vrstvě LoRa definuje rámec jako strukturu složenou z následujících polí. Pole jsou uvedena ve stejném pořadí jako v rámci.

Preamble Sekvence základních cvrků, která slouží k časové a frekvenční synchronizaci. Počet cvrků není pevně dán.

Symbole synchronizace rámce Dva modulované cvrky co mohou být použity pro identifikaci sítě. Hardwarový přijímač zahodí rámec, které obsahují synchronizační symboly co neodpovídají jeho nastavení.



Obrázek 4.4: Ukázka signálu lora [16] (upraveno)

Symbole synchronizace frekvence Dva sdružené cvrky následované sdruženým cvrkem s periodou $\frac{T}{4}$ určené pro přesnou frekvenční synchronizaci.

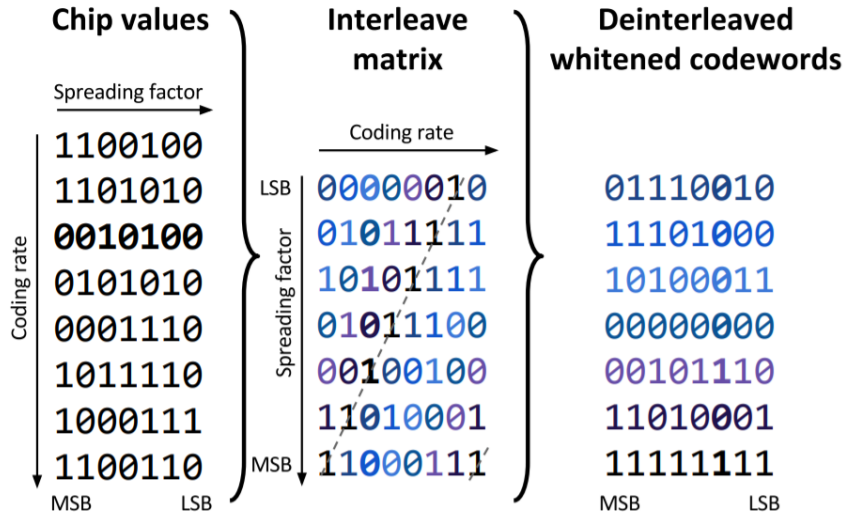
Hlavička (nepovinná) Hlavička obsahuje délku užitečných dat, použitou přenosovou rychlost, indikuje použití cyklického redundantního součtu (CRC - Cyclic redundancy check) a jednobajtovou kontrolní sumu hlavičky. Pro modulaci hlavičky je vždy použito $CR = 4$ a mód redukované rychlosti. Pokud hlavička vysílána není (implicitní mód) musí mít jak přijímač tak vysílač předem stejně nastavené CR a také zdali je použito CRC.

Užitečná data Pole o proměnné délce obsahující data vrstvy přístupu k médiu (MAC - Media access control) a případné dvoubajtové CRC těchto dat.

■ Struktura hlavičky

Délka hlavičky není ve specifikaci nikdy přímo určena. Lze jí však vydedukovat z toho, že hlavička je vždy vysílána v módu redukované rychlosti, má $CR = 4$ a SF minimálně 7. Z toho vyplývá že hlavička se musí vejít do mřížky $\{0, 1\}^{7-2x8}$ a to odpovídá 5 kódovým slovům. Každé slovo má 8 bitů a dohromady je to bitů 40. Jakékoliv zbývající bity jsou použity pro užitečná data.

Po dekódování díky redundantním bitům dostáváme $40 \frac{4}{8} = 20$ bitů nebo 2,5 bajtu. V [28] experimentálně vyzkoušeli pořadí hlavičky. První bajt udává délku datového obsahu, následuje půlslabika udávající CR a přítomnost MAC CRC a poslední bajt obsahuje kontrolní součet hlavičky, z něj je však používá jen 5 LSB bitů.



Obrázek 4.5: Diagonální prokládání při $FS = 7$ a $CR = 4$, hodnota čipu třetího symbolu je vyznačena tučně. [28]

4.1.5 Softwarová demodulace

[28] dokázali implementovat kompletní PHY vrstvu LoRa ve frameworku GNU Radio. Jejich zdrojové kódy jsou open source a dostupné na Githubu. Funkčnost příjmu signálu LoRa mého scanneru vychází z jejich práce. V této kapitole je popsán princip fungování.

Detekce a synchronizace

Aby mohl být signál demodulován musí být nejdříve detekován. K tomu slouží preamule která má dva opakující se cvrky čehož dokáže využít použitý Schmidl-Cox algoritmus. Ten definuje dvě veličiny $P(d)$ a $R(d)$, ty jsou definované takto [31]:

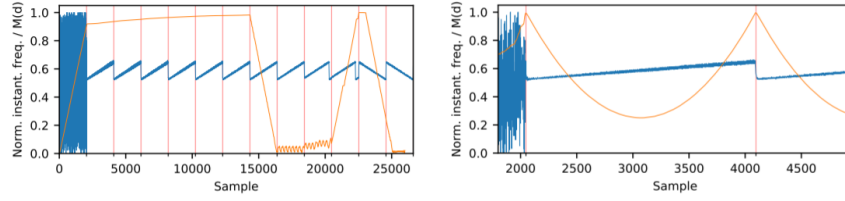
$$P(d) = \sum_{m=0}^{L-1} (x_{t+m}^* x_{t+m+L}) \quad (4.1)$$

$$R(d) = \sum_{m=0}^{L-1} |x_{t+m+L}|^2 \quad (4.2)$$

kde L je délka symbolu, t je index vzorku komplexního signálu x a x^* je jeho komplexně sdružený signál. Veličiny $P(d)$ a $R(d)$ jsou použity k výpočtu časové metriky $M(d)$:

$$M(d) = \frac{|P(d)|^2}{R(d)^2} \quad (4.3)$$

Časová metrika $M(d)$ v podstatě počítá normalizovanou autokorelaci délky L přes dva symboly, maximum bude mít ve chvíli kdy v signálu budou za sebou



Obrázek 4.6: Vlevo synchronizace pomocí Schmidl-Cox algoritmu a vpravo algoritmus použitý v [28]

dva totožné symboly. Díky tomu, že oba symboly jsou chybami způsobenými přenosem (interference, frekvenční odchylka nosné (CFO - Carrier frequency offset), odchylka vzorkovací frekvence) ovlivněny stejně, tak tyto chyby téměř neovlivní výsledek korelace.

I přesto že tento algoritmus detekuje preamble velmi dobře, nicméně ani to není vzhledem k plošině maxima dostatečně přesné pro signály LoRa. Aby ?? tenhle problém vyřešili museli vymyslet nové řešení. To se zakládá použití Schmidl-Coxovi metriky pro přibližné určení okna ve kterém se nachází druhý symbol preamble a následném zpřesnění pomocí ideálního lokálně vygenerovaného cvrku. Jeho okamžitá frekvence $\omega_l(t)$ a normovaná okamžitá frekvence signálu Lora $\omega(t)$ jsou vzájemně korelovány a index vzorku jež odpovídá maximální hodnotě této funkce je považován za počátek symbolu. Použití omažité frekvence místo komplexních hodnot je odůvodněno chybami CFO, které by bez korekce mohly ovlivnit přesnost synchronizace.

Výsledek je na obrázku 4.6. Poslední součástí tohoto řešení je určení prahové hodnoty maxima korelačního koeficientu okamžitých frekvencí lokálně gerovaného cvrku a přijátého. Pokud je tato hodnota menší než prahová je daný rámec zahozen, protože se buďto jedná o falešně pozitivní detekci rámce nebo o nepovedenou synchronizaci.

Demodulace

Po úspěšné synchronizaci následuje fáze demodulace. Při příjmu modulovaného cvrku s neznámým časovým posuvem $x(t + \hat{t})$, může být hodnota cvrku zrekonstruována navzorkováním signálu vzorkovací frekvencí chipů a výpočtem:

$$i = \text{Gray}(\arg \max(|FFT(x(t + \hat{t}) \odot \overline{x(t)})|)) \quad (4.4)$$

Kde $\overline{x(t)}$ značí komplexně sdružený základní cvrk, \odot značí multiplikaci po prvcích, $|FFT(x)|$ značí velikost Rychlé Fourierovi transformace x , a *Gray* je Grayovo kódování.

Oproti teorii má však v praxi FFT demodulace nevýhodu v tom, že je citlivá na odchylku frekvence, která způsobuje posun hodnot FFT a tím i odečítaných hodnot chipů. Tím pádem je potřeba přesná synchronizace frekvence, kterou je navíc potřeba aplikovat na každý kanál LoRa zvlášť. Separace kanálů a následná synchronizace každého z nich je však v softawaru příliš náročná operace a tak [28] přišli s novou metodou demodulace, která je nezávislá na

frekvenci a umožňuje demodulaci na všech kanálech současně v reálném čase. V porovnání s FFT metodou je však méně robustní.

Nejdříve je potřeba spočítat okamžitou úhlovou frekvenci $\omega[t] = \frac{d\varphi[t]}{dt}$. Poté je potřeba $\omega[t]$ vyhladit a decimovat konstantním decimačním faktorem $\frac{s_f T}{2^{SF}}$ kde s_f je vzorkovací frekvence. Díky tomu je pak počet vzorků v $\omega[t]$ shodný s 2^{SF} následně je vypočítán digitální gradient f :

$$D_t[\omega[t]] = \omega[t + 1] - \omega[t] \quad (4.5)$$

Tuto operaci si lze představit jako filtr horní propust okamžité frekvence nebo jako druhou derivaci fáze. Protože frekvence základního cvrku se lineárně zvyšuje s k - $\omega(t) = kt + f_0$ je její derivace $\omega'(t)$ rovna k . Pro modulované cvrky se však v D_t objeví ostré špičky v místech přechodu mezi vysokou a nízkou frekvencí. Přítomnost takových špiček indikuje časový posun \hat{t} , nepřítomnost naopak idikuje časový posun 0 - základní cvrk.

Dalším problémem při demodulaci je zpoždování/předbíhání hodin v jednotlivých zařízeních. Kristalové oscilátory v LoRa vysílači a SDR se budou zákonitě navzájem předbíhat nebo zpožďovat, rozdíl jejich frekvencí je předem neznámý, ale v průběhu času se musí projevit. To může způsobovat problémy zejména v případě delšího datového obsahu v kombinaci s vyšším SF. V patentu LoRa jsou pro účely korekce tohoto jevu použity pilotní symboly, které pomohou sledovat časování. Ve skutečnosti se však zdá, že k jejich použití nebylo přistoupeno. Je tedy nutné využít techniku slepého odhadu, která využívá převzorkování přijatého signálu N -krát. Aby tato technika byla funkční je potřeba aby hodnota N odpovídala následujícímu vztahu $|\Delta t| < \frac{N}{2}$, kde Δt je chyba časování na symbol. Prvním krokem je synchronizace popsaná v 4.1.5. Pokud je chyba časování na symbol $|\Delta t| < \frac{N}{2}$, lze $|\Delta t|$ určit následujícím způsobem:

1. Symbol je demodulován běžným způsobem jak je popsáno v 4.1.5 a je tak získána hodnota chipu i a časového posunu \hat{t} .
2. Na přijímači je lokálně vygenerován základní cvrk modulován na i což způsobí časový posun \hat{t}_1 lokálního signálu.
3. Protože lokálně generovaný cvrk není ovlivněn rozdílem oscilátorů vysílače a přijímače můžeme vzájemné zpoždění oscilátorů definovat takto $\Delta t = \hat{t}_1 - \hat{t}$. Nyní stačí na přijímači opravit \hat{t} připočtením vzájemného zpoždění k přijatému signálu.

Již zmíněná podmínka $|\Delta t| < \frac{N}{2}$ je daná tím, že při jejím nesplnění dekodér špatně určí hodnotu chipu i a chyba se tak bude šířit i do dalších symbolů. Hodnota N tak není určena přímo ale jako interpolace z \hat{t} do $\hat{t} + \Delta t$. Vyšší hodnoty N by dále vylepšovali přesnost korekcí zpoždění ale také by zvyšovali náročnost výpočtu.

■ Dekódování

Ve fázi dekódování jsou hodnoty chipů zpětně proloženy a tím jsou získána kódová slova s $4 + CR$ bity. Prvních 8 kódových slov lze přímo dekódovat jako hlavičku fyzické vrstvy. Datový obsah však musí být nejprve XORován bělicí posloupností. I přesto že dle výrobce LoRa Semtech má jít o sekvenci generovanou 9-bitovým LFSR ve skutečnosti je dle výzkumu [16] a [7] použita posloupnost mírně odlišná, která však není veřejně zdokumentována.

Pokud však známe původní kódové slovo i přijaté vybělené lze snadno zjistit sekvenci použitou pro XORování následovně $w^{(j)} = c_w^{(j)} \oplus c^{(j)}$. Pro zjednodušení lze vyslat všechna kódová slova nulová a tím dostaneme rovnici $w^{(j)} = c_w^{(j)} \oplus 0$ a na výstupu tak máme přímo samotnou bělicí sekvenci, kterou můžeme uložit a poté načítat z tabulky.

Po odbělování přichází poslední krok a to Hammingovo dekódování kódových slov. U LoRa jsou datové bity umístěny na jiných pozicích než je běžné a to na indexech 0,1,2 a 3 bajtu místo indexů 1,2,3 a 5. Po extrakci datových bitů jsou pak pomocí paritních bitů detekovány a případně opraveny chyby a tím získána původní data.

■ 4.2 SigFox

■ 4.2.1 Obecné informace

Sigfox je další LPWAN síť. Původně tato technologie vznikla ve Francii odkud se rozšířila do zbytku světa. Stejně jako LoRaWAN používá hvězdicové architektury a má i podobné využití a cíle. Na rozdíl od LoRaWAN však nevyužívá rozprostřené spektrum, ale vysílání na velmi úzké šířce pásma.

Stejně je i to, že nepoužívá synchronizovanou síť a šetří tak zdroje normálně na synchronizaci vynakládané a místo synchronizace je použit náhodný přístup k médium, kdy zařízení každou zprávu pošle třikrát v různých časech a na různých frekvencích. Na rozdíl od LoRaWAN však nemá třídy zařízení, a tak jediná možnost pro downlink je potom co zařízení ukončí uplink.

Dalším omezením je povolený počet zpráv, který je pro koncová zařízení limitován na obsazení pásma 1 % času což při dvanácti bytových zprávách odpovídá 144 zprávám denně [34]. Brány mohou obsazovat médium po 10 % času. Při maximálním počtu zařízení to odpovídá 4 zprávám denně na každé zařízení. Tato omezení vyplývají z použité technologie a použití nekomerčních frekvencí na které není potřeba žádná licence.

Architektura sítě je velmi podobná LoRaWAN. Koncová zařízení komunikují přes Sigfox s bránami, které jsou připojeny přes veřejný internet k serverům Sigfox, které zajišťují správu stanic, spektra a celkového fungování sítě, zpracování zpráv, ukládání dat a také slouží k přístupu do sítě skrz webový portál nebo API. Servery Sigfox také zajišťují účtování a případnou analýzu dat.

Bezpečnost je zajištěna na několika vrstvách. Za prvé zařízení v síti nejsou přímo dostupná z internetu, což slouží podobně jako firewall. Za druhé je systém odolný proti replay útokům, díky použití unikátního ID pro každou

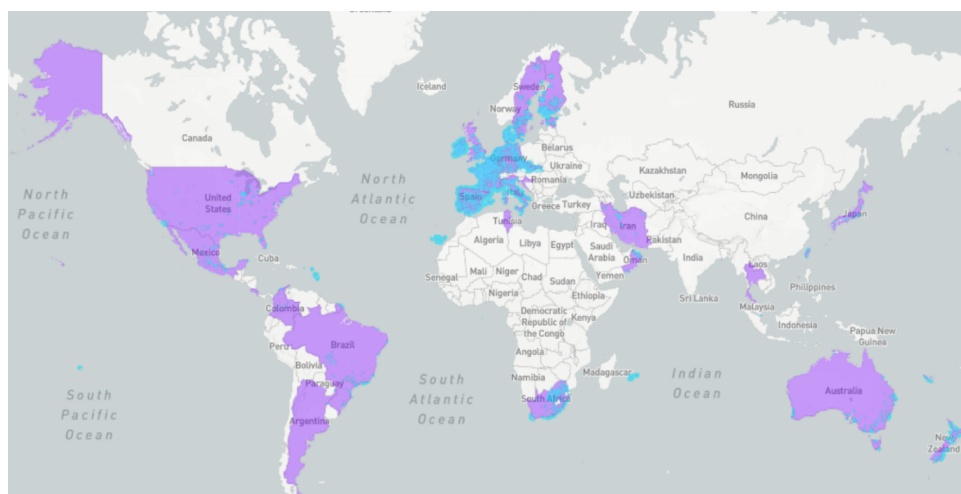
zprávu. Každé zařízení má také unikátní symetrický autentizační klíč, pomocí kterého je generován Message Authentication Code (MAC), který zajišťuje autenticitu a integritu každé zprávy. V případě nutnosti zašifrovat samotné zprávy je možné použít vlastní šifrování na aplikační úrovni nebo využít zabudovaný systém na bázi AES [32]. Od brány dál směrem k serverům a uživateli je použito standardní zabezpečení používané na internetu.

■ 4.2.2 Specifikace

- technologie: UNB (Ultra Narrow Band)
- modulace: DBPSK
- způsob příjmu: bez synchronizace, MIMO
- velikost zprávy: 0-12 Bytů (96 bitů)
- rychlost přenosu: 100 bitů/s
- doba přenosu a zpracování: 4-6 s
- frekvence: 868MHz (ETSI), 915 MHz (FCC)
- počet zpráv za den: 144
- maximální počet zpráv na BTS denně: 9 000 000
- vysílací výkon: 25mW / 14 dBm
- budget link: 162 dB
- zpětný kanál: 4 zprávy po 8 Bytech denně
- dosah v terénu: až 50 km v terénu, 3 km ve městě pro indoor
- spotřeba: 5 mA – 45 mA při vysílání, 0 mA v klidu
- výdrž na bateriích: 5-15 let (až 20 let na dvě AA baterie)
- zabezpečení: hash, šifrování možné na aplikační úrovni
- SLA: 99 % [25]

■ 4.2.3 Pokrytí)

V rámci celosvětového měřítko je SigFox rozvinut zejména v Evropě, kde je nejsouvislejší pokrytí. Dále se je síť ve výstavbě ve většině zemí Ameriky a v Austrálii. Na mapě jsou fialově vyznačeny země, kde probíhá budování sítě a modrou aktuální pokrytí. V České republice provozuje síť SigFox operátor SimpleCell ve spolupráci s operátorem T-mobile CZ. S výstavbou sítě začali již v létě 2015 v rámci testovacího provozu. Aktuálně pokrývají 96 % obyvatelstva a 94 % území a v plánu mají pokrýt 98 % obyvatelstva a 96 % území ČR



Obrázek 4.7: Pokrytí Sigfox ve světě[33]

	Uplink	Downlink
Šířka pásma	100 Hz	1500 Hz
Přenosová rychlost	100 baud	600 baud
Modulace	DBPSK	GFSK
Vysílací výkon	25 mW	500 mW
Frekvence	868,0 - 868,6 Mhz	869,40 - 869,5 Mhz
Střída	1 %	10 %

[40]

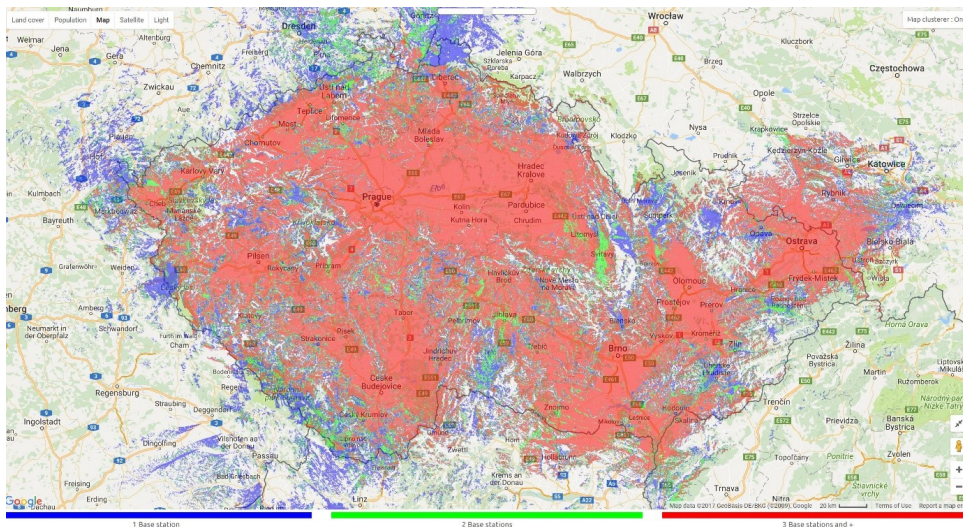
Tabulka 4.1: Parametry přenosu Sigfox pro Uplink a Downlink

[35]. V mapě jsou červenou barvou označena místa, která pokrývají tři a více bran, což je ideální stav. Zelenou barvou jsou označena místa pokryta dvěma branami a modrou ta jen jednou.

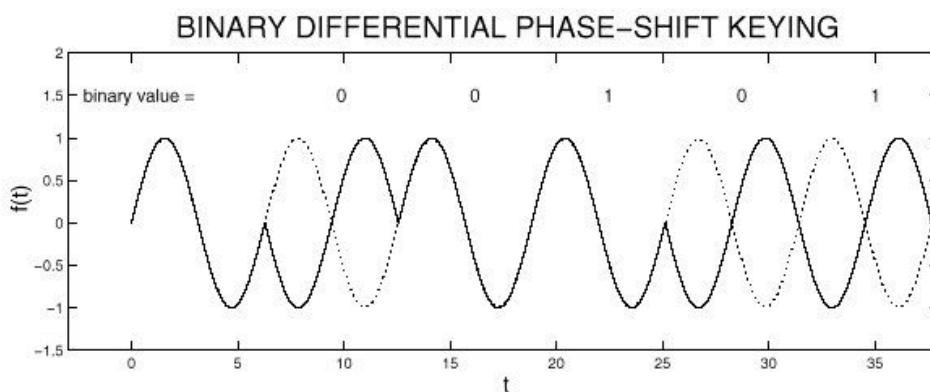
4.2.4 Rádiový protokol Sigfox)

Základním poznávacím znamením technologie Sigfox je použití velmi úzkého pásma (UNB - Ultra Narrow Band). To dovoluje vysílání na větší vzdálenosti díky úzkému zaměření vysílacího výkonu každého zařízení. Navíc díky použití UNB může na jednom místě operovat velké množství zařízení bez přílišného vzájemného rušení.

Sigfox používá různá modulační schémata pro uplink a downlink, důvodem je optimalizace využití spektra. Různá je i šířka pásma, přenosová rychlost, vysílací výkon i struktura rámce. Popíšu tedy oba směry zvlášť a zaměřím se přitom na parametry používané v Evropské unii. V jiných regionech se parametry liší, důvodem jsou různé alokace spektra.



Obrázek 4.8: Pokrytí Sigfox v České republice [24]

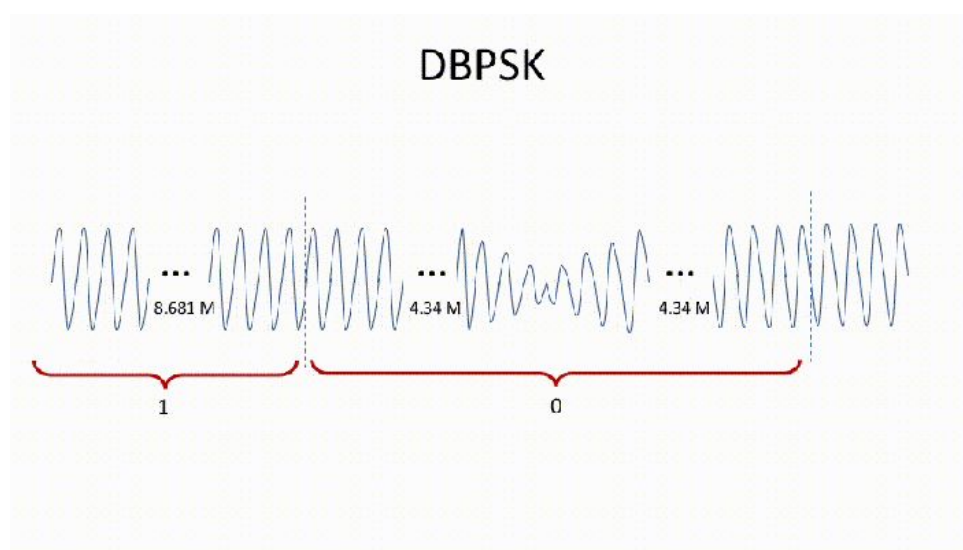


Obrázek 4.9: Příklad modulace DBPSK [18]

Fyzická vrstva uplinku

Sigfox pro Uplink používá modulaci typu DBPSK (Differential Binary Phase Shift-Keying). Tato modulace stejně jako standartní BPSK používá fázi pro určení symbolu, narozdíl od ní však nejen zaznamená přechod z 0 do 1, ale i bez referenční fáze rozliší zda je daný symbol 0 nebo 1. Toto je možné díky tomu, že symbol neurčuje konkrétní fáze, ale její změna, odtud také plyne její pojmenování.

Na obrázku 4.9 je znázorněn příklad fungování této modulace. Pokud nenastává žádná změna ve fázi signálu identifikujeme 1 pokud změna nastane pak je symbol určen jako 0. V tomto příkladu každý symbol trvá jednu periodu signálu a to by při frekvenci 868 MHz kterou Sigfox v Evropě používá znamenalo přenosovou rychlost 868Mb/s což nejen nedává smysl pro daný účel, ale navíc tak Sigfox ani ve skutečnosti fungovat nemůže kvůli nemožnosti



Obrázek 4.10: Modulace amplitudy při změně fáze [9]

dekódovat podobný signál na jakoukoliv použitelnou vzdálenost.

Verze DBPSK Sigfoxem používaná pro každý symbol sadu period signálu a hodnotu symbolu určí podle toho zda se průběhu vysílání této sady změní fáze (0) či nikoliv (1). Demodulace tak není tolik ovlivněna lokální proměnlivostí fáze.

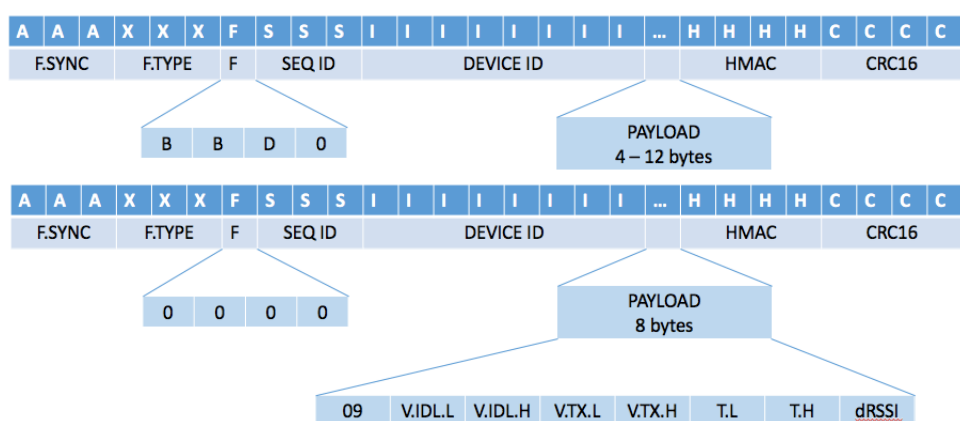
V případě evropské verze je přenosová rychlost uplinku 100 bit/s a frekvence signálu 868,1 Mhz. Z toho vyplývá že každý bit/symbol odpovídá 8 681 000 periodám. Velké množství period využívá přijímač pro statistické určení posunu fáze každého bitu. V případě 1 bude celá sada mít fázi stejnou a v případě 0 se fáze v prostřed posune.

Pokud by se však fáze posunula při plné amplitudě znamenalo by to velký negativní dopad na spektrální efektivitu. Aby se tento efekt zmenšil je také okolo posunu fáze modulována amplituda. Grafické znázornění této modulace je na obrázku 4.10.

■ Struktura rámce uplinku

Na fyzické vrstvě Sigfox definuje rámec jako strukturu složenou z následujících polí. Pole jsou uvedena ve stejném pořadí jako v rámci. V prvním opakování není použito žádné specifické kódování a data tak jsou přímo čitelná. V druhém a třetím opakování jsou však pole F.TYPE, SEQ.ID, DEVICE.ID a PAYLOAD zakódovány.

Při druhém opakování hodnota každého bitu závisí na dvou předchozích, pokud jsou stejné tak je hodnota aktuálního bitu prohozena. Při třetím opakování závisí hodnota aktuálních dvou bitů vždy na hodnotě dvou předchozích !!!!!!! uplne nevím jak !!!! možná jen tak, že pokud je objeden dozadu nula tak prohazuji hodnotu.



Obrázek 4.11: Struktura rámce Sigfox pro Uplink (nahore) a konkrétní podoba rámce OOB (dole). Každá buňka reprezentuje půslabiku (4 bity) [9]

Délka zprávy	1. opakování	2. opakování	3. opakování
12 bajtů	0x94C	0x971	0x997
8 bajtů	0x611	0x6BF	0x72C
4 bajty	0x35F	0x598	0x5A3
1 bajt	0x08D	0x0D2	0x302
1 bit	0x06B	0x6E0	0x034
RX OOB	0xF67	-	-

Tabulka 4.2: Hodnoty F.TYPE v závislosti na délce zprávy a pořadí opakování [9]

Následkem použití jiného kódování pro každé opakování, není i přes identický obsah nikdy vyslán stejný rámec třikrát. Změny fáze jsou vždy na jiných místech, to může pomoci při příjmu zejména v podmínkách kdy se signál ztrácí v šumu.

Speciálním typem rámce uplinku je rámec OOB (Out of Band) který slouží jako potvrzení příjmu rámce downlinku. Lyší se v několika ohledech, jednak není opakován a druhak nepřenáší uživatelsky užitečná data. Místo nich v PAYLOAD poli, které je vždy dlouhé 8 bajtů, zasílá informace o napětí během nečinnosti i během vysílání, teplotu, RX RSSI.

F.SYNC Nebo také Frame Synchronization jak již název napovídá slouží pro synchronizaci. Objevuje se na začátku každého rámce. V rámci této části se střídá 0 a 1 po dobu 3 půslabik (12 bitů), objevuje se tedy 6 změn fáze. Tyto změny fáze využívá přijímač k synchronizaci hodin aby později korektně identifikoval začátek každého bitu.

Synchronizace je ukončena ve chvíli, kdy se vzor změní z 01 na 00 nebo 11 v závislosti na typu rámce. Typ rámce je určen v následujícím poli F.TYPE.

F.TYPE Frame Type je část rámce, dle které lze rozeznat o kolikáté opa-

kování (každý rámec se opakuje třikrát) vysílání se jedná a také délku užitečných dat. Přesné hodnoty nejsou nikde definovány, ale z pokusů o reverzní inženýrství [9] vyplývá tabulka 4.2. (mohu/nemohu potvrdit) Zajímavé je, že opakování a délka dat nemají samostatné indikační bity, ale existují konstanty pro jejich kombinace.

F.FLAGS Tato půslabika je rozdělená do tří částí. První dva bity určují počet bajtů přidaných do rámce jako výplň pro dosažení jedné z předdefinovaných délek. To znamená, že pokud by délka zprávy vyšla na 6 bajtů, tak bude odeslána v osmi bajtovém rámci a dva bajty tak budou přidány a hodnota těchto bitů bude 10 neboli 2.

Třetí bit značí zda je na tento rámec vyžadována odpověď ve formě downlinku. Pokud je tento bit 1 síť by měla v následujícím intervalu pro downlink odpovědět. Poslední bit je vždy 0.

V případě OBB rámce jsou všechny bity 0.

SEQ.ID Sequence ID je 12 bitové pole, které se inkrementuje s každým odeslaným rámcem. Slouží jako určitá ochrana proti znovuoodeslání zachycených zpráv. Limitem této ochrany je, že se SEQ.ID každých 2048 zpráv opakuje a tak v určitých situacích nemusí backend znovuoodeslanou zprávu zachytit. Bity jsou seřazeny od nejvýznamějšího po nejméně významný.

DEVICE.ID Toto pole je dlouhé 32 bitů a představuje unikátní ID Sigfox zařízení. Bajty jsou řazeny od nejméně významného, ale bity každého z nich jsou řazeny od nejvýznamějšího bitu.

PAYLOAD Toto pole obsahuje přenášená data a může být dlouhé 12, 8 nebo 4 bajty.

HMAC HMAC (hash-based message authentication code) je druh MAC pro nějž výpočet je použita kryptografická hešovací funkce a tajný klíč. Slouží jak ke kontrole integrity dat, tak k autentifikaci. Každé Sigfox zařízení má svůj tajný privátní klíč vypálený přímo na čipu. Délka pole je 2 bajty.

CRC Poslední pole je CRC s délkou 16 bitů.

■ Downlink

Pro komunikaci směrem ze sítě do zařízení Sigfox používá modulační schéma GFSK, vyšší výkon, širší pásmo a navíc má i větší poměr vysílacího času, viz 4.1. Mohlo by se zdát, že jsou Uplink a Downlink kanály nevyvážené ve prospěch Downlinku, ale opak je pravdou jelikož Downlink je vyslán pro všechna zařízení v dosahu a Uplink se počítá pro každé zařízení zvlášť. V Evropě po započtení všech omezení vychází pro každé zařízení 140 Uplink zpráv denně a jen 4 Downlink zprávy.

Přesný obsah rámce ani další detaily momentálně neznám. Pouze lze něco málo vyčíst z tabulky 4.3.

Preamble (91)	Frame Sync (13)	ECC (32)	Payload (0 - 64)	MAC (16)	FCS (8)
---------------	-----------------	----------	------------------	----------	---------

Tabulka 4.3: Pole rámce Downlinku (délka pole v bitech)



Část II

Přehledový příjmač

Kapitola 5

Hardware

V této kapitole shrnu veškerý potřebný hardware pro přehledový přijímač. Jedinou povinnou součástí určenou v zadání bylo SDR, ostatní komponenty však vyplynuly z potřebných vlastností přijímače, a tak nezbylo příliš prostoru pro invenci. Cíl bylo zhotovit přijímač tak aby byl mobilní, mohl fungovat na baterie a vyhověl zadání v tom smyslu, že kromě příjmu dat ze sítí IoT bude schopen je také předat po IP protokolech dál a sám přehledně zobrazit.

5.1 RTL-SDR

I přesto, že konkrétní druh SDR nebyl v zadání určen, byla jeho volba jen formální záležitostí. Jak už totiž vyplývá z kapitoly `refchapter:sdr` RTL-SDR je de facto standard pro amatérské využití. Mezi ostatními SDR se vyjímá především cenou, která je desetinnásobně až stonásobně menší. Jeho hlavní nevýhodou oproti dražším variantám je nemožnost vysílání, tato nevýhoda však vzhledem k určení není relevantní. Dalším potenciální nevýhodou by mohl být rozsah. Vzhledem k tomu, že jsem se však rozhodl soustředit pouze dva druhy IoT sítí - LoRa a Sigfox a obě pracují v bezlicenčním pásmu ISM 868Mhz, je i tato nevýhoda bezvýznamná. Ostatní nedostatky oproti dražší konkurenci jsou již tak malé, že by pro účely této práce nedávalo smysl volit jinak.

V případě dalšího vývoje v budoucnosti je však možné, že bude potřeba SDR upgradovat. To může být motivováno technologií NB-IoT, kterou u nás provozuje operátor Vodafone. Tato technologie funguje na stejných pásmech jako LTE a tak se vzhledem k nemožnosti příjmu frekvencí nad 1766 Mhz může NB-IoT dostat mimo rozsah RTL-SDR. Upgrade by pak mohl proběhnout dvěma způsoby, buďto přidáním konvertoru, nebo výměnou samotného SDR. To už jsou však úvahy náležící pozdějšímu pracem. V obou případech se však díky využití GNU Radio bude jednat o jednoduchou výměnu, bez nutnosti větších zásahů do software přehledového přijímače.

Specifikovat vybrané SDR pouze jako RTL-SDR však nestačí. Toto označení se používá pro nepřeberné množství USB DVB-T tunerů používajících čipset RTL2832U. Ty se mezi sebou mohou lišit ve několika ohledech. Prvním je použitý tuner. Nejžádanější variantou pro účely SDR je tuner E4000 od firmy Elonics, má totiž nejširší rozsah z používaných tunerů.

citeosmocom Jeho nevýhodou však je dostupnost, tento tuner se již nevyrábí a tak jsou RTL-SDR s tímto tunerem ke koupi již jen ze zapomenutých zásob obchodů a nebo z druhé ruky. Jejich cena tak oproti ostatním variantám v poslední době stoupla i přesto, že ještě před několika lety se nijak nelišila. V poslední době je tak nejběžnějším tunerem R820T2, ten má sice rozsah o něco menší než E4000 (ale větší než ostatní tunery), nemá v něm však mezery a je tak i vzhledem k dostupnosti nejvhodnější variantou.

Druhým zásadním parametrem, kterým se můžou RTL-SDR lišit je konektor antény. Vzhledem k tomu, že byly původně určeny pro sledování televize, mají často konektor PAL nebo MCX. [30] To v zásadě nepředstavuje problém, ale pro práci s rádiem se častěji používá konektor SMA a proto existuje také větší množství a výběr antén k tomuto konektoru.

Posledním zásadním parametrem odlišujícím různá RTL-SDR je přítomnost teplotně kompenzovaného oscilátoru (TXCO). Ten je užitečný, protože při použití se RTL-SDR zahřívá a pokud není jeho oscilátor tepelně kompenzován, tak se jeho frekvence mění. Zahřívání na plnou provozní teplotu trvá několik minut, během této doby zůstane nalazená frekvence u SDR-RTL s TXCO stabilní, kdežto u ustatných se bude posouvat.

Po zvážení těchto skutečností a také kvůli dalším úpravám mířeným na zlepšení použití jako SDR jsem zvolil RTL-SDR Blog V3. Tato varianta RTL-SDR byla speciálně upravena pro použití ne jako DVB-T tuner ale jako SDR. Kromě tuneru R820T2, konektoru antény SMA a TXCO má i další vylepšení. Nejdůležitějším vylepšením pro použití v přehledovém přijímači je použití hliníkového krytu, který funguje jako chladič pro interní komponenty a zároveň jako elektromagnetický štít. To spolu s lepším návrhem PCB také snižuje množství šumu. Další vylepšení nejsou pro účely této práce až tolik podstatná spíše přispívají k znovupoužitelnosti tohoto SDR v dalších projektech. Konkrétně se jedná o softwarově spínatelné bias tee (napájení např. nízkofrekvenčního zesilovače z RTL-SDR), režim přímého vzorkování (zprístupňuje frekvence 500 kHz - 24 MHz přímo přes SMA konektor) a mnoho dalších maličkostí, jejich kompletní výčet je v obrázku 5.1

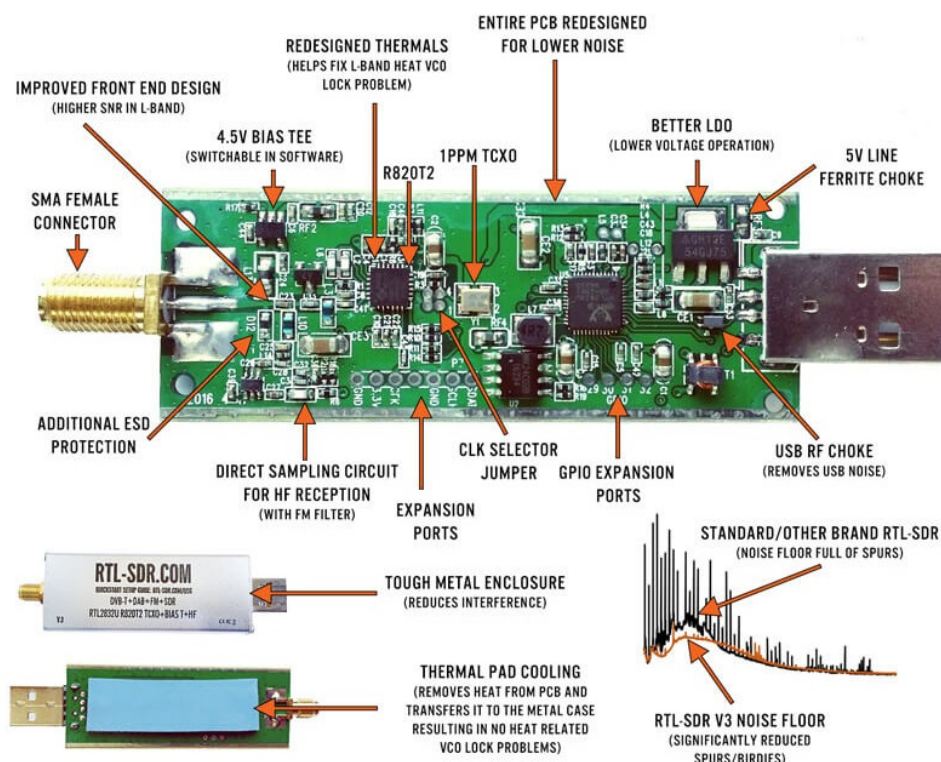
■ 5.2 Raspberry Pi

■ 5.2.1 Displej

■ 5.3 Napájení

Vzhledem k potřebě fungování přehledového přijímače na baterky bylo nutné řešit napájení složitěji, než jen zapojením Raspberry Pi do USB nabíječky. Bylo potřeba zvolit správnou baterii, DC-DC měnič s výstupem 5V pro napájení Raspberry PI (A přes něj ostatních komponent), způsob nabíjení baterií, vypínač a propojení těchto komponent.

Jako zdroj energie jsem zvolil články typu 18650 Li-Ion. Je to stejný typ



Obrázek 5.1: Výhody RTL-SDR blog V3 oproti běžným RTL DVB-T [30]

článků jako se používá v přenosných počítačích, elektromobilech, vaporizérech a power bankách. Využívá technologii Li-Ion a díky tomu netrpí pamětovým efektem jako starší Ni-Cd a má vysokou hustotu energie. K dispozici jsem měl články značky Samsung model ICR18650-26. Kapacita jednoho článku je 2,55 Ah a nominální napětí 3,7 V. Články jsem použil tři, vložil do zakoupeného držáku a zapojil paralelně. Dostal jsem tak baterii 1s3p s kapacitou 7,65 Ah a nominálním napětím 3,7 V (min 3V, max 4,2V) z čehož vyplývá kapacita ve watthodinách okolo 28 Wh.

Vzhledem k tomu, že Raspberry Pi vyžaduje napětí 5 V bylo potřeba napětí z baterie zvýšit na tuto hodnotu. Zároveň jsem potřeboval vyřešit jakým způsobem budu baterii nabíjet. Stejná napětí a problémy však řeší každá power banka a tak jsem se uchýlil k již připravenému řešení v podobě booster modulu určeného do powerbanky. Tyto moduly dokáží zvýšit napětí z baterie na 5V, ohlídat napětí na bateriích aby nekleslo pod 3V (při menších napětích dochází k poškození článků) a zároveň při zapojení zdroje dokáží baterii nabíjet a ohlídat, že její napětí nepřesáhne hodnotu 4,2 V, nad kterou dochází k poškození článků a i případnému vzplanutí.

Vyzkoušel jsem několik druhů (z Aliexpress se dají podobné moduly koupit za malé částky) a nakonec se rozhodl pro HCX-PCB-429. Tento modul nejen deklaruje maximální výstupní proud 2A, ale doopravdy ho i dosahuje, což

se většině ostatních nepodařilo. Další jeho výhodou je, že při připojování a odpojování nabíječky nepřeruší dodávku elektřiny na 5 V výstupu, to u těchto modelů bohužel nebývá běžné. Jeho poslední výhodou oproti dalším vyhovujícím je jeho velikost, ostatní kvalitnější moduly byly navrženy pro powerbanky s dvěma výstupy a měli tím pádem větší rozměry.

Raspberry Pi samotné nemá žádný vypínač, musel jsem tedy přidat vlastní. Použil jsem jednoduchý kolébkový vypínač a umístil ho v obvodu až za booster modul, díky tomu lze baterii napájet i ve vypnutém stavu a spotřeba samotného modulu je minimální, je na neustálé připojení k baterii stavěn.

Samotné Raspberry Pi 3 má při plné zátěži spotřebu okolo 730 mA [10], RTL-SDR má spotřebu okolo 280 mA [30], displej má spotřebu menší než 400 mA [1] dohromady tedy méně než 1410 mA což při 5 V voltech dává 7 W a přibližná doba na baterii tedy bude okolo 4 hodin.

5.4 Kryt

Abych z již představených jednotlivých komponent dokázal vyrobit přehledový přijímač, bylo potřeba je dohromady umístit do ochranného krytu. Přestože bývá pro účely závěrečných prací běžné sestavit kryt z plexiskla s vyřezanými otvory, rozhodl jsem se jít jinou cestou - navrhnout a poslat na 3D tiskárnu vytisknout kryt přizpůsobený na míru komponentům.

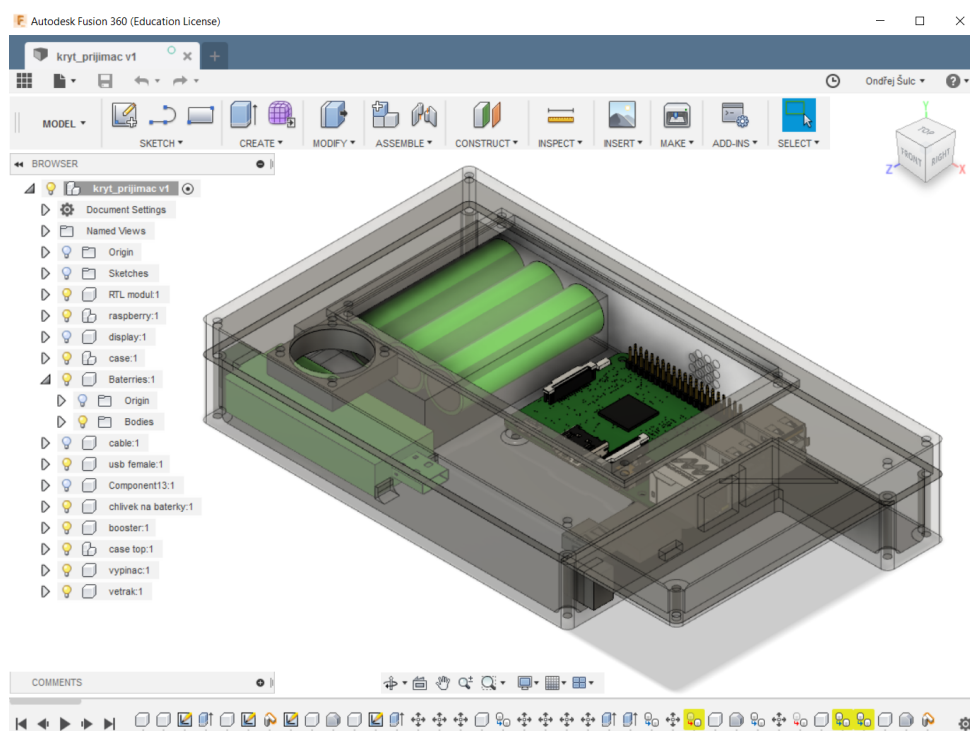
Vzhledem k tomu, že jsem neměl mnoho předchozích zkušeností s návrhem 3D modelů, dal jsem na doporučení kolegy Jaroslava Bartoše, který se 3D tiskem zabývá a pro návrh zvolil CAD software Autodesk Fusion 360. V rámci studia mají studenti všech vzdělávacích institucí k dispozici licenci zdarma

5.4.1 Autodesk Fusion 360

Fusion 360 of Firmy Autodesk je varianta jejich CAD programu zaměřená zejména na kolaboraci a na návrh modelů určených pro výrobu, ať už pomocí 3D tiskáren nebo obrábění, soustružení, řezání vodním paprskem, laserem či plazmou [6]. Umožňuje také simulaci 3D modelů, kde jsou k dispozici nástroje jako pro statické zatížení, tepelnou analýzu, optimalizaci tvaru a podobně. Aplikace vznikla již v roce 2013 a od té doby jsou neustále přidávány nové funkce.

Princip celé aplikace je založen na cloudu, který je používán jako pro ukládání dat, tak pro náročnější výpočty. Samotná klientská aplikace je napsána v WebGL/HTML5 a je vidět snaha o maximální jednoduchost uživatelského prostředí. Kdo již někdy pracoval s jejím velkým příbuzným Autodesk Fusion Inventor, ten jistě najde hodně známých prvků. Podobně jako Inventor také podporuje parametrické i přímé modelování [37].

Velkou výhodou, kterou jsem často využil při modelování komponent přijímače je také schopnost otevřít téměř libovolný formát 3D modelu, já jsem tak mohl importovat již hotové modely Raspberry Pi, Li-Ion článků nebo různých konektorů. Poslední velmi užitečnou funkcí je již v úvodu zmíněná podpora kolaborace, díky ní jsem mohl svůj návrh průběžně konzultovat s již zmíněným



Obrázek 5.2: Prostředí Autodesk Fusion 360 a navrhnutý model krytu

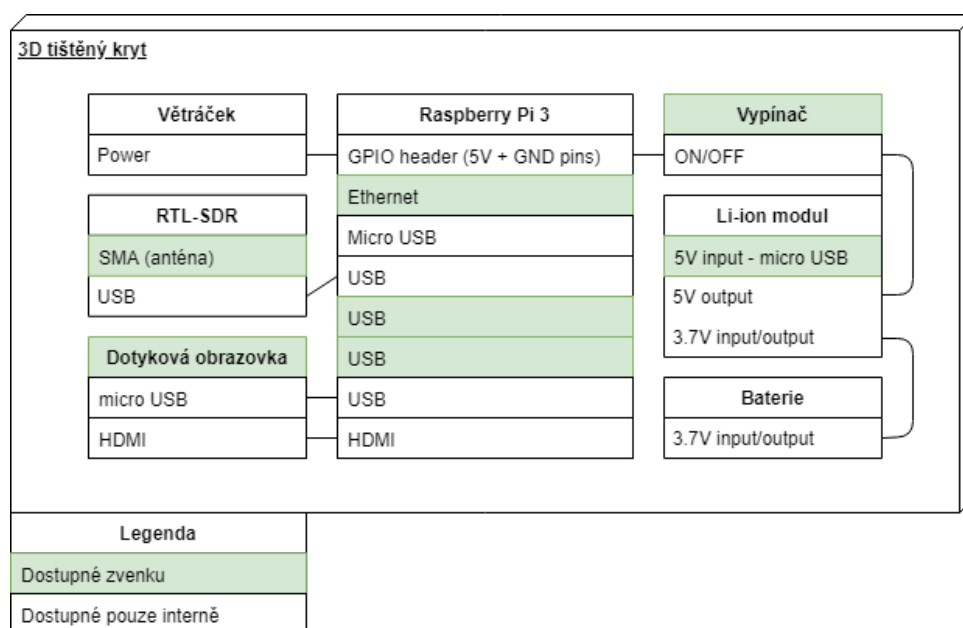
kolegou a provádět jeho úpravy tak aby šel výsledný model bez problému vytisknout.

5.4.2 Model

Tvorbu modelu jsem započal vymodelováním všech komponent. Modeloval jsem je do různé úrovně detailu podle toho jak moc na jejich členění mohl záviset výsledný kryt. Například držák s Li-Ion články stačilo vymodelovat jako přibližný kvádr, do krytu jsem ho totiž jen vlepil na rovnou plochu. Oproti tomu Raspberry Pi které mělo mít některé konektory dostupné zvenku, bylo potřeba mít vymodelované do detailu, naštěstí jsem však našel velmi podrobný model na internetu a vyhnul se tak zdlouhavé práci.

Po vytvoření modelů komponent následovalo jejich poskládání tak aby všechny potřebné porty byly dostupné zvenku, výsledný kryt se vešel na tisknutelnou plochu tiskárny a také aby dva největší producenti tepla - RTL-SDR a Raspberry Pi nebyly hned vedle sebe. Jako poměrně náročné se ukázalo, zpřístupnění LAN a volných USB portů zároveň s uschováním veškerých propojení komponent uvnitř krytu. Z tohoto důvodu nemá kryt tvar kvádru, ale je na jedné straně vykousnutý.

Výsledný kryt se skládá ze dvou částí. Ve vrchní části je zabudován displej a větráček a v dolní zbytek komponent. Ke spodní části bylo nutné ještě přilepit záslepku otvoru, který vznikl vykrojením původního kvádru. Záslepka byla vytištěna zvlášť, protože by ji bez opor nebylo možné vytisknout jako součást



Obrázek 5.3: Diagram zapojení a dostupnosti portů přehledového přijímače

spodního ani horního dílu. Díly jsou spojeny šesti šrouby, celý kryt má tak velkou pevnost a působí robustně.

5.4.3 Tisk

O tisk se postaral můj kolega Jaroslav Bartoš. Použita byla tiskárna Průša MK2 a materiál petg. Tisk byl nastaven na nižší rychlost kvůli menší chybivosti a tisk vrchního dílu tak trval 6 hodin včetně záslepky a spodního 10 hodin. Vytisknutý kryt byl hned na první pokus použitelný bez větších úprav, jediný komponent co nešel zasadit bez úprav krytu byl displej jehož připravený výřez byl asi o půl milimetru menší, než displej samotný a bylo tak potřeba otvor trochu zvětšit.

5.5 Zapojení

Při zapojení se centrálním prvkem stalo Raspberry Pi a k němu jsou připojeny všechny ostatní komponenty. Konkrétní porty a dostupnost komponent a volných portů zvenku lze vyčíst z diagramu na obrázku 5.3.

Kapitola 6

Software

V této kapitole se budu věnovat veškerému software potřebnému k fungování přehledového přijímače ať už se bude jednat o SW vytvořený třetími stranami nebo mnou. Nejprve shrnu požadavky, pak představím použitý software z nich vyplývající a nakonec nastíním architekturu celého systému, to jak jsou jednotlivé programy propojeny a jaká mají rozhraní.

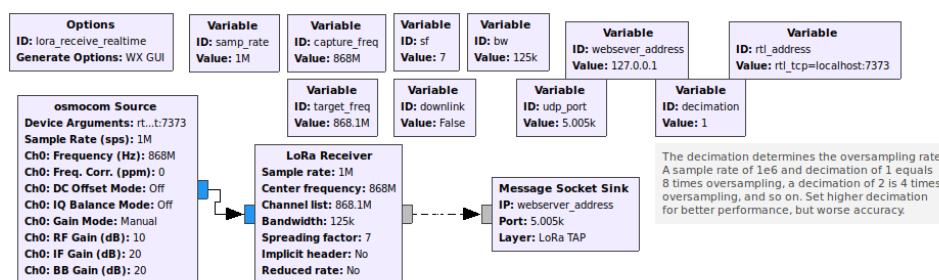
6.1 Požadavky a z nich vyplývající software

Všechny požadavky vycházejí ze zadání diplomové práce. Hlavním požadavkem je přijmout a dekodovat zprávy vysílané pomocí technologií IoT a následně je zobrazit a to jak lokálně přímo na přehledovém přijímači, tak vzdáleně přes IP protokoly. Tento hlavní požadavek lze rozložit do jednotlivých částí.

První část se týká příjmu zpráv z IoT zařízení z hlediska HW to obstarává RTL-SDR a použitý softwér ho tak musí umět ovládat a připravit přijaté I/Q vzorky pro dekodování, o to se postará knihovna librtlsdr a její utility které jsou součástí projektu rtl-sdr. Druhý požadavek tedy dekodování zpráv bude splněn pomocí toolkitu GNU Radio ve kterém bude přijatý signál demodulován a zprávy dekodovány. Požadavek na zobrazení zpráv lokálně a vzdáleně jsem se rozhodl uspokojit zároveň a to pomocí webového uživatelského rozhraní založeného na microframeworku Flask a jeho rozšíření Flas-socketIO zajišťující zpětnou vazbu uživatelského rozhraní a celého systému v reálném čase.

6.1.1 librtlsdr a její utility

Tato knihovna byla vyvinuta v rámci softwarového projektu rtl-sdr od organizace Osmocom (Open source mobile communications). Knihovna je jeho ústřední částí sloužící k ovládání a přístupu k RTL-SDR a v podstatě všechny uživatelské aplikace využívající RTL-SDR jí potřebují ke svému fungování. Tato knihovna závisí na další knihovně libusb, to je dáno použitím rozhraní USB pro připojení RTL-SDR k počítači [17]. Kromě librtlsdr projekt obsahuje také několik utilit pro použití v terminálu:



Obrázek 6.1: Flowgraf pro příjem LoRa

těchto parametrů ve formě hlavičky k dekódovaným datům. Bohužel však v metodě používané pro výpis dekódovaných dat byl bug, který způsoboval korupci parametrů odesílaných do hlavičky. Tuto metodu není nutné používat pokud je použit blok pro odeslání zpráv a tak byla oprava pro účely této práce jednoduchá, ale přijít na to co chybu způsobuje zabralo mnoho hodin debugování. A o výsledku budu informovat autora, protože si myslím, že tento bug byl důvod proč přidání parametrů do hlavičky neprovedl on sám.

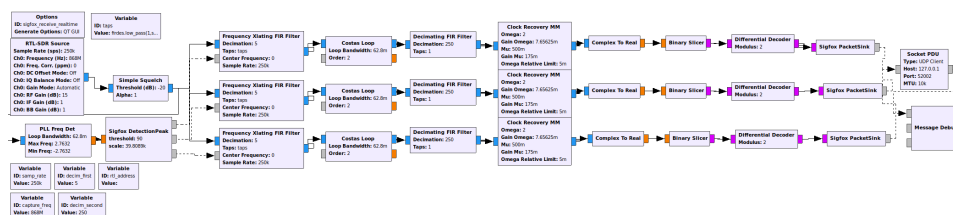
Blok receiver je složen ze třech bloků channelizer, controler a debugger. Tyto bloky implementují postup popsany v kapitole 4.1.5. V rámci jeho konfigurace je nutné nastavit požadovaný kanál, faktor rozproštění a rozlišovací práh. Jeho výstupem je již dekódovaný rámec, který lze pomocí posledního bloku Message Socket Sink odeslat do dalších aplikací.

■ Sigfox

Tento flowgraf byl původně součástí upravené verze programu Scapy nazvané scapy-radio, která do něj přidávala podporu pro protokoly Bluetooth, Zigbee, Zwave a konečně Sigfox [39]. Obsahuje dva bloky, které nejsou standartní součástí GNU radia, jinak je složen z již připravených bloků.

První z nestandardní blok se jmenuje Sigfox DetectionPeak a je implementován v pythonu. Jeho účel je detekovat frekvence nosných třech opakování vysílání Sigfox a tyto frekvence předat do třech samostatných větví flowgrafu, každá demodulující jen jedno vysílání. Blok má dva parametry, paramater threshold udává o kolik se musí lišit následující detekovaná frekvence aby jí bylo možné považovat za následující opakování a parametr scale pomáhá normalizovat frekvenci.

Blok Sigfox PacketSink dekóduje demodulovaný signál. Je implementován v C++ a má dvě fáze. V první fázi detekuje preambuli a jakmile narazí na F.SYNC následovaný F.TYPE přečte ho a zjistí z něho délku zprávy, druhé části dekódu následujících několik bajtů, dle délky indikované v preambuli. Tato data přidá k preambuli a vytvoří paket.



Obrázek 6.2: Flowgraf pro příjem Sigfox

6.1.3 Flask a Flask-socketIO

6.2 Architektura

Softwarovou výbavu přehledového přijímače lze rozdělit do několika bloků. Ústřední částí je kontrolní aplikace, která spouští, propojuje a ovládá ostatní bloky. Druhý blok se věnuje přímo modulu RTL-SDR, zabývá se jeho ovládáním a distribucí digitalizovaných vzorků rádiového spektra. Další blok tvoří skripty využívající GNU Radio, ty zpracovávají vzorky z bloku RTL-SDR a po dekodování rámce je předávají kontrolnímu bloku. Posledním blokem je samotný klient, který ve webovém prohlížeči zobrazuje přijaté rámce a umožňuje změnu nastavení. Tyto bloky fungují samostatně a tak lze každý provozovat na jiném stroji, musí však být zajištěna možnost jejich vzájemné komunikace a kompatibilní operační systém.

6.3 Třídy IoT technologií

Jednotlivé standardy IoT podporované přehledovým přijímačem jsou dekodované v těchto třídách

6.4 kontrolní aplikace

Tento blok jsem implementoval v jazyce python a skládá se z několika souborů.

6.4.1 webserver.py

Webserver.py je hlavní skript tohoto bloku. Jeho kostru tvoří microframework Flask a jeho rozšíření Flask-socketIO. Pomocí něho vytvářím server který publikuje index.html. Dále v tomto skriptu vytvářím instance tříd definujících jednotlivé podporované standardy IoT. bibcit

6.4.2 index.html

Pro dekodování zpráv využívá instance tříd pro jednotlivé technologie. Tyto třídy je možné psát manuálně nebo generovat pomocí grafického prostředí gnuradio-companion. Já jsem bloky připravil



Kapitola 7

Výsledky



Kapitola 8

Závěr

Bibliografie

- [1] 52-Pi. *5-Inch-800x480-HDMI-TFT-LCD-Touch-Screen*. URL: https://wiki.52pi.com/index.php?title=5-Inch-800x480-HDMI-TFT-LCD-Touch-Screen_SKU:Z-0053.
- [2] rtl-sdr.com admin. *REVIEW: AIRSPY VS. SDRPLAY RSP VS. HACKRF*. 2016. URL: <https://www.rtl-sdr.com/review-airspy-vs-sdrplay-rsp-vs-hackrf/>.
- [3] Airspy. *What is Airspy?* URL: <https://airspy.com/>.
- [4] LoRa Alliance. *LoRa Alliance technology*. URL: <https://www.lora-alliance.org/technology>.
- [5] LoRa Alliance. *LoRaWANTM White Papers*. URL: <https://www.lora-alliance.org/lorawan-white-papers>.
- [6] Autodesk. *Revoluční cloudová 3D CAD aplikace. Vytvářejte cokoliv*. 2018. URL: <https://www.fusion360.cz/>.
- [7] Josh Blum. *LoRa modem with LimeSDR*. Červ. 2016. URL: <https://myriadrf.org/blog/lora-modem-limesdr/>.
- [8] coherent-receiver. *RTL-SDR Introduction*. 2018. URL: <https://coherent-receiver.com/publications>.
- [9] Paul Disk91. *The Sigfox radio protocol*. Lis. 2017. URL: <https://www.disk91.com/2017/technology/sigfox/the-sigfox-radio-protocol/>.
- [10] Raspberry Pi Dramble. *Power Consumption Benchmarks*. URL: <https://www.pidramble.com/wiki/benchmarks/power-consumption>.
- [11] Bertalan Eged a Benjamin Babják. “Universal Software Defined Radio Development Platform”. In: *Dynamic Communications Management*. 2006.
- [12] Claire Goursaud a Jean-Marie Gorce. “Dedicated networks for IoT : PHY / MAC state of the art and challenges”. In: *EAI endorsed transactions on Internet of Things*, (lis. 2015).
- [13] ITEAD. *Airspy R2*. URL: <https://www.itead.cc/airspy.html>.
- [14] Simonyi Károly. *RTL Multi-User Server*. 2014. URL: https://github.com/simonyiszk/rtl_mus.

- [32] Sigfox. *Make things come alive in a secure way*. Ún. 2017. URL: https://www.sigfox.com/sites/default/files/1701-SIGFOX-White_Paper_Security.pdf.
- [33] Sigfox. *SIGFOX has you covered*. URL: <https://www.sigfox.com/en/coverage>.
- [34] Sigfox. *Sigfox Technical Overview*. Květ. 2017.
- [35] Simplecell. *Simplecell - connecting things*. URL: <https://simplecell.eu/>.
- [36] Statista. *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)*. Ún. 2018. URL: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.
- [37] CAD Studio. *Autodesk Fusion 360 - cloudový 3D CAD/CAM/CAE*. URL: <https://www.cadstudio.cz/fusion360>.
- [38] tapr. *HPSDR Mercury tech specification*. URL: https://www.tapr.org/kits_merc.html.
- [39] cyber tools. *scapy-radio*. 2016. URL: <https://bitbucket.org/cybertools/scapy-radio/src>.
- [40] Juan Carlos Zuniga a Benoit Ponsard. "SIGFOX System Description". In: *IETF 96*. <https://www.ietf.org/proceedings/96/slides/slides-96-lpwan-10.pdf>. Čvc 2016.