FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Multiplatformní nástroj pro odposlech dat z rádiového přenosu pomocí SDR
Student:	Bc. Martin Šimůnek
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Počítačová bezpečnost
Katedra:	Katedra informační bezpečnosti
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

Analyzujte technologii SDR(softwarově definovaného rádia) a jeho možnosti pro odchytávání bezdrátových zařízení.
Zaměřte se zejména na zařízení pracující v pásmu 433MHz a 868MHz.
Prozkoumejte existující řešení pro odchytávání a analýzu těchto zařízení.
Využijte SDR a navrhněte nástroj pro odposlech dat přenášených rádiovým signálem v pásmu 433MHz a 868 MHz.
Nástroj by měl být tvořen knihovnou a grafickým rozhraním pro odposlech a zpracování naměřených dat.
Podporována budou zejména zařízení typu: bezdrátový teplotní senzor, bezdrátové senzory v automobilu a bezdrátové měřiče tepla v domácnostech.
Nastroj bude umožňovat analyzovat přenos, a to jak nešifrovaný, tak i šifrovaný.
Pokud bude k dispozici klíč pro dešifrování, bude nástroj umožňovat i dešifrování, a to zejména pro bezdrátové měřiče tepla v domácnostech.
Navržené řešení zrealizujte a řádně otestujte.

Elektronicky schválil/a prof. Ing. Róbert Lórencz, CSc. dne 6. února 2023 v Praze.

Diplomová práce

MULTIPLATFORMNÍ NÁSTROJ PRO ODPOSLECH DAT Z RÁDIOVÉHO PŘENOSU POMOCÍ SDR

Bc. Martin Šimůnek

Fakulta informačních technologií
Katedra informační bezpečnosti
Vedoucí: Ing. Pavel Kubalík, Ph.D
3. května 2023

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů kódu

Tímto bych chtěl poděkovat vedoucímu této diplomové práce Ing. Pavlovi Kubalíkovi, Ph.D za rady, časté konzultace a pomoc, kterou mi poskytl při psaní této práce. Dále bych chtěl poděkovat rodině za poskytnutí zázemí pro psaní této práce a svým přátelům a známým, kteří byli velkou oporou.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 3. května 2023 .....

Abstrakt

Tato diplomová práce se zabývá analýzou technologie softwarově definovaného rádia (SDR) a tím, jak tuto technologii, která je dnes snadno dostupná pro koncového uživatele, využít k odposlechu běžných typů bezdrátových senzorů v rámci bezlicenčních/ISM pásem. Analýza se týká zařízení, které pracují v evropských bezlicenčních subGHz pásmech, a to konkrétně v pásmu 433 MHz a pásmu 868 MHz. Tato zařízení jsou většinou malé domácí senzory s bateriovým napájením. V rámci práce se rozebírají zabezpečení takovýchto senzorů a možná bezpečností rizika spojená s jejich užíváním a zabezpečením. Dalším bodem, který tato práce řeší, je navržení a sestavení multiplatformní odposlechové aplikace s malými externími závislostmi pro běžné typy senzorů za pomocí SDR technologie na základě analýzy dostupných technologií a již realizovaných projektů. Konkrétní aplikace stojí na levném hardwaru RTL-SDR V3 USB zařízení s čipem RTL2832U od firmy Realtek, které umožňuje zachytávat rádiové přenosy a výstupem jsou IQ data, která se zpracovávají v softwaru. V poslední fázi je takto sestavená aplikace testována na fyzických senzorech a zhodnocena, stejně tak jako jsou zhodnoceny údaje získané ze senzorů oproti tomu, co lze běžně vyčíst a získat z manuálů a volně dostupných datasheetu od výrobce.

Klíčová slova SDR, softwarově definované rádio, 433MHz, 868Mhz, RTL, rtl2832U, RTL-SDR, RTLSDRV3, odposlech komunikace, bezdrátová komunikace, bezdrátové senzory, RTL-SDR, ISM, bezlicenční pásmo, bezdrátový přenos, senzor, subGhz, bezpečnost senzorů

Abstract

This thesis analyses Software Defined Radio (SDR) technology and how this technology, which is now readily available to the end user, can be used to eavesdrop on common types of wireless sensors within the license-exempt/ISM bands. The analysis concerns devices operating in the European licence-exempt sub-GHz bands, specifically the 433 MHz and 868 MHz bands. These devices are mostly small battery powered home sensors. The security of such sensors and the possible safety risks associated with their use and security are discussed. Another point addressed in this thesis is to design and build a multi-platform eavesdropping application with small external dependencies for common sensor types using SDR technology based on the analysis of available technologies and already implemented projects. The specific application is based on a low-cost RTL-SDR V3 USB device hardware with RTL2832U chip from Realtek, which allows to capture radio transmissions and output IQ data that is processed in software. In the last stage, the application thus built is tested on physical sensors and evaluated, as well as the data obtained from the sensors are evaluated against what can be commonly obtained from manuals and freely available datasheets from the manufacturer.

Keywords SDR, software defined radio, 433MHz, 868Mhz, RTL, rtl2832U, RTLSDR, RTL-SDRV3, eavesdropping, wireless communication, wireless sensors, RTL-SDR, ISM, ISM bands, sensor, subGhz, sensor security

Seznam zkratek

ASK	Amplitude shift keying
FSK	Frequency shift keying
PSK	Phase shift keying
QAM	Quadrature amplitude modulation
PAM	Pulse amplitude modulation
PCM	Pulse coded modulation
PWM	Pulse width modulation
PPM	Pulse position modulation
OOK	On-Off keying
PC	Personal Computer
IQ	In phase, quadrature
SDR	software defined radio
PIR	passive infrared
HW	hardware
SW	software
API	aplication programming interface
ISM	industrial scientific and medical
IF	intermediate frequency
AGC	automatic gain control
GUI	graphical user interface

Úvod

V dnešním světě rychle přibývá množství různých zařízení, které nám mají ulehčit život a snížit náklady lepším monitorováním a rozhodováním se. Tato zařízení do budoucna mají i mít využití v ochraně zdraví, nahrazovat péči o seniory či duševně choré lidi doma místo v nemocnicích a institucích, umožňovat stálý dohled a zároveň ulehčovat jejich život ve známém prostředí[1].

Ke svému správnému fungování tato zařízení potřebují data, a tato data poskytují různé druhy senzorů. Vzhledem k množství a nutnosti rovnoměrného rozprostření senzorů, často tato zařízení komunikují bezdrátově, aby byla možná, co nejsnazší instalace, a nebylo třeba vést kabely ke každému senzoru ve zdech budov nebo nevzhledných lištách. Do budoucna se počítá s tím, že bezdrátové senzorové sítě budou ovlivňovat skoro každé odvětví lidské činnosti[2].

Již dnes si můžeme všimnou zahušťujících se sítí bezdrátových senzorů. Tempo expanze se má ještě urychlit a největší expanze bezdrátových senzorů se očekává v rámci odvětví monitoringu prostředí.

Možností komunikovat je několik, přes wi-fi, jako velká část IoT zařízení, skrz bluetooth tj. technologie na frekvenci 2,4 GHz nebo 5 GHz nebo pomocí levnějšího způsobu, a to pomocí rádiových signálů nižších než 1GHz v bezlicenčních pásmech spolu s nějakou centrálou, která je pak napojena dál. Taková to centrála může mít podobu nějaké krabičky napojené na internet, nebo jen displaye, co ukazuje venkovní teploty nebo centrálního systému pro alarm.

Pro mnoho senzorů je komunikace na subGHz pásmech , tj na pásmech nižších než 1GHz velmi výhodná. Tato bezlicenční pásma nabízejí nižší spotřebu při komunikaci a větší dosah, menší propustnost není problémem pro jednoduché senzory. Problémem těchto pásem v současné době zůstává legislativa[3], přesto se očekává, že tato pásma budou hrát důležitou roli.

Dá se říci, že trend je zlevňování a zvyšování množství takovýchto zařízení v domácnostech[4] i průmyslu. Zařízení pro bezdrátové odečty vody, elektřiny, tepla, vytápění, a mnoho dalšího často v reálném čase přímo v domě se stávají čím dál populárnější.

Zlevňování a rozšíření těchto zařízení však může být rizikem, zvláště u zařízení, která užívají bezlicenční pásma nižší nižší než 1Ghz ke komunikaci s centrálou, může dojít k odposlechu nebo i narušení integrity dat. Na rozdíl od wi-fi, nebo bluetooth nemá tato komunikace žádné široce rozšířené standardy. Existují různé pokusy o standarty, většinou proprietární jako je ZIGBEE, který je asi nejznámějším standardem v subGHz pásmu. Většina výrobců ale standardy nepoužívá a často používají co nejednoduší proprietární implementace, na laciném hardwaru, které často ani neumožňují nějaké zabezpečení jako je šifrování nebo ochrana integrity dat.

Dalším důvodem pro neimplementace šifrování jsou nároky na baterie[5]. Pro ušetření malé mono-článkové baterie, která je nejčastějším způsobem napájení takovýchto senzorů, je komunikace omezena na minimum a potřebné předzpracování dat také, často nedochází k žádnému předzpracování informací či šifrování dat, což vede ke zcela nechráněným přenosům dat. Toto vytváří jakýsi ”postranní kanál”o budově, domácnosti, či o celém areálu, ze kterého lze vyčíst soukromé informace. Senzory lze i obelstít a v případě potřeby senzory snadno napodobit.

S příchodem levných SDR přijímačů tak nastává otázka, jestli je možno tyto informace doposlechnout a zneužít.

U budov se takto dá například získat informace o tom, ve které místnosti někdo je (pokud například v místnostech je detektor pohybu pro alarm), ve které místnosti se topí a na kolik stupňů (chytré regulace vytápění) , lze odchytit kód od dálkového ovládání garáže (zvláště starší typy dálkových ovládání lze snadno napodobit). Ve výsledku se tedy jedná o zajímavý způsob získávání informací či manipulaci s nimi.

V Současné době nejzajímavější pásma na evropském trhu z tohoto hlediska jsou 433 MHz a 868 MHz[6]. Obě dvě pásma jsou užívána pro přenos malého množství dat.

Dosah 433 Mhz je nejčastěji kolem 300 metrů bez překážek (může být ale i více) , v případě budov záleží na materiálu, ze kterých je budova postavena. A v případě 868 MHz se dosah nejčastěji pohybuje okolo 500 m (opět může se dát zvětšit). V tomto případě se jedná o hodnoty, kterých zařízení dosahují vzhledem k limitům stanoveným lokálními regulačními úřady a nejčastěji používanými technologiemi. Jinak dosah může být i větší.

Zachytávat tato často nijak nešifrovaná data je levné, stačí na to levné zařízení, které vzešlo z televizního DVB-T tuneru s čipem (RTL2832U[7]) . Takováto zařízení se dají pořídit od 1500 Kč a dokáží přijímat rádiový signál, který následně převádějí na digitální IQ data, která lze jednoduše zpracovávat pomocí software. Takováto řešení se nazývají softwarově definované rádia. A jde o silně modulární a levné systémy.

0.1 Motivace

Jak již bylo naznačeno, v domácnostech, veřejných budovách i podnicích se rozmáhá používání chytrých systémů, které ke svému správnému fungování vyžadují velké množství dat ze senzorů. Ať již jde o bezpečnost, úspory, nebo snadnější řízení systémů.

Problémem však je, že spousta z těchto systémů je navržena tak, aby bylo co nejsnazší je instalovat, udržovat a zároveň byla co nejlevnější. Z tohoto důvodu tato zařízení bývají stavěna na levném a slabém HW, často napájena jednoduchými mono-článkovými bateriemi. Data jsou pak často přenášena nechráněně, zařízení nejsou nijak identifikována, kontrola integrity dat je pouze na úrovni chybového přenosu a data nejsou ověřována ani šifrována. To je něco, co teoreticky nevadí, například, u teplotních senzorů, které jsou určeny pouze k orientačnímu zjišťování teploty. Stejně nezabezpečené přenosy dat jsou však stále používány u senzorů pohybu a dalších senzorů podle nichž se řídí třeba i vytápění budov, což může vést k velkým zranitelnostem systémů, které na nich závisí.

Takto nashromážděna data mohou mít přidanou informační hodnotu pro nekalé elementy, případně lze tyto senzory rušit a mást aktivně. Teoreticky také senzory mohou být využívané i v autech a dopravě a jejich ovlivnění může mít vážnější následky ovlivňující bezpečnost provozu.

0.2 Cíl

Cílem této práce je tedy vytvořit jednoduchou multiplatformní aplikaci, která bude za pomocí levného USB zařízení, RTL-SDR blog V3, odchytávat a číst data, která odesílají bezdrátové senzory a bude nabízet možnost dešifrování šifrovaných dat (v případě je nutná znalost klíče a módu šifry senzoru).

Krom tohoto, aplikace by měla umět zapisovat nezpracovaná zachycená IQ data. A nabízet omezený možnosti analýzy vysílání senzorů jakým je například ukládání zpracovaných a filtrovaných dat.

Aplikace bude fungovat hlavně pro evropská bezlicenční pásma nižší než 1Ghz a to 868 Mhz a 433 Mhz

Cílem by měla být uživatelsky příjemná aplikace schopná odposlechnutí komunikace běžných senzorů. Druhotným cílem je také poukázání na problematiku zabezpečení bezdrátových senzorů

a poskytnou analýzu možné zneužitelnosti některých senzorů.

Aplikace dále může sloužit jako základ pro sbírání dat z nesourodých systémů. Pro implementaci centrály, která by umožnila agregaci přijímaných dat.

Nejprve je třeba pokrýt daná ISM pásma, k čemu jsou, jak je lze využívat, jaké zákony regulují jejich využití. Z tohoto důvodu představím příslušný zákon o komunikačních službách. V tomto zákoně jsou definované podmínky využití daného pásma a technické parametry vysílačů.

Dále je nutné pokrýt, jakým způsobem funguje rádiový přenos dat ze senzorů do sběrných bodů, jaké modulace se využívají.

Po teoretických pasážích představím program využívající RTL-SDR zařízení pro odchytávání a analýzu zachycených dat. Tento program bude multiplatformní s grafickým uživatelským rozhraním.

Následně představím senzory, se kterými jsem se mohl setkat, naměřit je a zhodnotit v rámci testování aplikace.

Kapitola 1
Rešerše

Tato kapitola se zaměří na to, jaké existují již nástroje a aplikace pro práci s SDR, jaké existují knihovny pro tvorbu takových aplikací. Dále se také zaměří na existující programovací nástroje a jazyky, které se dají využít v rámci tvorby aplikace.

V rámci rešerše se budu zaměřovat hlavně na již existující projekty, které využívají SDR k monitorování rádiových frekvencí a případně k zachytávání dat z nich. Vyberu a porovnám přednostně projekty, které jsou blízko zadání této práce.

Dále porovnám knihovny a frameworky, které jsou k dispozici a vhodné pro dosažení cílů této práce. Bude se jednat o knihovny pro práci s RTL-SDR zařízením, grafické a šifrovací knihovny. Rešerše poskytne přehled o dostupných technologiích a nástrojích, ze kterých bude vycházet návrh.

1.1 Existující řešení pro SDR

Vzhledem k všestrannosti SDR a zároveň jeho příznivé ceně existuje mnoho projektů, jež používají jako svou bázi právě SDR, mnoho těchto projektů je proprietárních a těmi se zabývat v této práci nebudu, důvodem je že SDR je dnes základem mnoho radiokomunikačních zařízení na zcela jiné úrovni, než je v této práci. Příkladem prvního proprietárního využiti jsou armádní vysílačky, které lze přelaďovat dle potřeby mezi několika pásmy. Dále vysílače mobilního signáálu 4G a 5G jsou založeny na SDR.

1.1.1 Rtl_433

Rtl_433 [8] je open source řešení univerzální centrály pro chytrou domácnost, které umožňuje sbírat data z nešifrovaných senzorů na frekvencích 433 Mhz, 868 Mhz a ostatních ISM pásmech a poté je zpracovávat a posílat ve zprávách pro komunikaci s nějakým systémem chytré domácnosti. Celé řešení je psáno jako C monolitický program, který slouží jako server a zpracovává a odesílá zprávy. Toto řešení integruje MQTT protokol pro komunikaci s dalšími systémy.

Celkově se řešení zdá optimalizované a výhodou je otevřený zdrojový kód, který v některých případech může sloužit jako základ pro další použití, tvůrce programu je stále aktivní a na projektu pracuje. Aplikace je stavěná jako server, který běží na pozadí a nabízí pouze CLI jako uživatelské rozhraní. V rámci aplikace není podporováno šifrování. Aplikace je dobře optimalizovaná a funkční. Lze stáhnout jako deb balíček či zdrojový kód a zkompilovat. Aplikace je multiplatformní.

Aplikace je zaměřená jako prostředník pro chytrou domácnost a podporuje například OpenHab,HomeAsssitant,Domoticz a NodeRed.

1.1.2 Sdr#

Sdr# [9] je aplikace psaná v C# (.NET), dále nabízí pro C# vývojové balíčky a API pro rozšíření aplikace. Aplikaci lze zdarma stáhnout a používat. Umožňuje poměrně rozsáhlé možnosti pro analýzu radiového spektra, hlavně jeho vizuální stránku. Lze zapínat či vypínat různé druhy zobrazení, filtrování vln a mnoho dalšího. Tento nástroj je velmi vhodný k prohlédnutí si radiových pásem, jak moc jsou využívána či rušena, zda nějaké zařízení na nich aktuálně vysílá, jaké konkrétně využívá frekvenční rozpětí, zda není v okolí zdroj nějakého rušení, atp.

Toto již hotové řešení je spíše vhodné jako analyzátor rádiového spektra, než nějaké nějaké konkrétní řešení jako výše uvedená aplikace na sběr dat ze senzorů.

1.1.3 Sdr++

SDR++ [10] je aplikace podobná SDR# s tím rozdílem, že se jedná o opensource, multiplatformní aplikaci napsanou v C++. Aplikace umožňuje podobně jako SDR# spektrální analýzu rádiových vln, které jsou přijímány pomocí SDR. Podporuje několik druhů SDR zařízení. Nabízí celou řadu filtrů a signálových demodulátorů. Je možnost pro aplikaci psát pluginy. Narozdíl od SDR# nenabízí SDK balíček.

Ve své podstatě se jedná o velmi podobné řešení jako výše popsané SDR#, které se pyšní že je bez zbytečností a rychlé. Ve srovnání použitelnosti je tato aplikace ekvivalentní s SDR#.

1.1.4 GNU radio

GNU radio [11] je velmi robustní opensourcové řešení/framework vyvíjené v Pythonu a C++, nabízející bloky pro zpracovávání signálu a implementaci softwarových rádií. Jednotlivé bloky lze spojovat programově v jazyce C++ nebo Pythonu či přes "kreslící"nástroj lze tvořit flow graphy, což je pospojování bloků za sebou, kdy každý blok má konkértní účel nad přijatými daty. Lze takto spojovat blok čtení dat, filtrování, převádění, atp. GNU radio umožňuje zpracovávání signálu rychle, robustně a kvalitně.

Hodně složitý a obsáhlý framework, spíše sloužící k samotnému zpracovaní signálů, než k interpretaci dat. Vhodné na rychlé prototypování. Velké množství externích závislostí.

1.1.5 Shrnutí

Již existují různé řešení, které umožňují analyzovat či dokonce poslouchat rádiové signály jako takové, velká část se zaměřuje na běžné rádiové vysílání typu FM, získávání zvuku skrze rádio a analýzu rádiových vln v okolí. Jedno řešení je však velmi blízko cíle této práce, tímto řešením je aplikace rtl_433 a to jednak díky svému opensource přístupu, minimálních externích závislostí a zároveň možností zachytání bezdrátových senzorů. Rozdíl je v zaměření, kdy toto řešení má sloužit hlavně jako robustní server pro přijímaní dat ze senzorů a jejich přeposílání do nějaké centrály chytré domácnosti. Tato aplikace neposkytuje příjemné uživatelské prostředí, e v mnoha místech je aplikace příliš složitá a nabízí pro projekt nepotřebné funkcionality.

1.2 Existující knihovny pro RTL-SDR

V této sekci se budou rozebírat již existující knihovny pro komunikaci se zařízeními obsahujícími čip RTL2832U od firmy REALTEK. Tato knihovna se stane důležitým bodem pro návrh a řešení

samotné aplikace a hlavním vstupním bodem pro ovládání zařízení.

1.2.1 Librtlsdr

Librtlsdr[12] je knihovna v psaná v jazyce C od projektu OSMOCOM, který je zaměřený na opensource mobilní komunikaci. Knihovna je základem mnoha aplikací pro komunikaci s levnými zařízeními, která jsou základy amatérských SDR. Tato konkretní knihovna umožňuje z čipu RTL2832U od firmy Realtek, původně užívaný pro potřeby přijímání pozemního digitálního rozhlasového a televizního vysílání, získávat IQ, 16 bitové vzorky, kdy I a Q jsou 8 bitová čísla. Díky tomuto se cenově dostupné amatérské SDR velmi rozšířilo. Dá se říci že tato knihovna to umožnila spolu s příchodem čipu RTL2832U.

Knihovna je opensource a je závislá pouze na knihovně libusb. Obě knihovny mají své binární soubory připravené a lze je využít pro Windows i pro Linux. Na stránce OSMOCOM lze najít kompilované Windows binární soubory, bohužel tyto sobory jsou kompilovány špatně a nefungují spolu s libusb tak jak by měly.

Knihovna librtlsdr je snadná na použití a ovládání rtl-sdr zařízení.Nabízí API pro nastavení zařízení a získání IQ vzorků pro další zpracování. Tato knihovna je limitována USB 2.0 propojením mezi knihovnou a zařízením, které neumožňuje samplovat rychleji než 2.4 Mega samplu za sekundu. V případě vyšších rychlostí samplování jsou některé vzorky ztraceny.

Knihovna nabízí navíc synchronní i asynchronní získávání informací přímo ze samotného USB zařízení.

1.2.2 Libsdr

Libsdr [13] je C++ opensource knihovna postavená nad librtslsdr od OSMOCOM a libpthread. Tato malá C++ knihovna byla stvořena jako projekt jehož cílem bylo dosáhnout lepšímu porozumění SDR technologie ze strany autora. Knihovna je spíše zaměřená na přijímaní dat, která jsou interpretovatelná jako zvuk, než jako digitální data. Mezi příklady je například FM rádio. Knihovna má slabou dokumentaci, žádnou komunitu a jako projekt je opuštěna již od roku 2015.

1.2.3 LibSDR

LibSDR [14] je C# knihovna psaná bez externích závislostí nebo procesů, není závislá na knihovně od OSMOCOMU. Část kódu této knihovny je založená na kódu SDR# aplikace. Knihovna klade důraz na přenositelnost a jednoduchost použití API, nikoli na maximální výkon jako například SDR#. Knihovna přiznává single thread zaměření.

V současné době se knihovna již nezdá býti udržovaná. Poslední příspěvek je více než 2 roky starý. Dokumentace ke knihovně je velmi stručná.

1.2.4 Pyrtlsdr

Pyrtlsdr [15] je wrapper pro knihovnu librtlsdr od OSMOCOMu pro jazyk Python. Knihovna poskytuje wrappery pro většinu funkcí z librtlsdr, na této knihovně také závisí, podle dokumentace tato knihovna neposkytuje wrappery na všechny funkce knihovny od OSMOCOMU.

Knihovna dále rozšiřuje nad knihovnou od OSMOCOMU některé své vlastní funkce jako je rtlsdraio - asynchronií input output operace v rámci pythonu a rtlsdrtcp posílání IQ dat ze zařízení pomocí tcp protokolu.

1.3 Existující knihovny pro GUI

Frameworky pro GUI byly vybírány na základě jediné vlastnosti a to multiplatformitě. Frameworky které zde jsou porovnány všechny nabízí multiplatformní řešení. Krom této základní vlastnosti se také porovnává hlavně strmost učící křivky a jednoduchosti kompilace a slinkování takové knihovny s projektem. Většina knihoven má více API pro více programovacích jazyků.

1.3.0.1 Qt

Qt[16] je jedno z nejznámějších multiplatformní řešení pro grafiku v rámci C++ a Pythonu. Řešení je spravováno a podporováno firmou Qt group a nabízí dva druhy licencování a to komerční i opensource. Kód je volně přístupný. Aplikace v tomto frameworku mohou mít velice přehledné a pěkné UI. Framework je vhodný pro velké firemní projekty, vyznačuje se silnou podporou,dostupností mnoha manuály a komunitními návody a velmi obsáhlou dokumentací a developer blogy. Nabízí nástroje pro snadné budování vzhledu aplikace. Má vše, co je potřeba pro velké projekty. QT framework umožňuje nativní vzhled na všech cílových platformách.

Nevýhodou je velikost a složitost frameworku. Z náhledu do dokumentace a do API volání je jasné, že projekt je zaměřen hlavně na potřeby složitých GUI aplikací. Podle diskusí na různých stránkách má QT velmi prudkou křivku učení. Samotná webová stránka je velmi nepřehledná, až odrazující od užití.

1.3.0.2 WxWidgets

WxWidgets[17] je opensource framework psaný v C++ jako multiplatformní řešení uživatelského grafického interface. Framework má API pro Python, Pearl, C# a C++. Řešení je využitelné pro komerční i opensource projekty. Samotné řešení je již poměrně známé a dobře otestované časem i uživately. Tento framework nabízí poměrně širokou komunitní i komerční podporu. Vzhled aplikací psaných v tomto frameworku je poněkud hůře vypadající se srovnáním s Qt, ale stále si udržuje nativní vzhled na cílových platformách.

Jde o velký a složitý projekt s opět poměrně strmou učící křivkou, hlavně z důvodů složitého API. Tento projekt je zaměřen spíše na budování komplexních GUI aplikací. Na rozdíl od Qt je tento projekt založen čistě na přístupu programovacím, takže zde není k dispozici žádný pomocný nástroj na vybudování GUI. Což dělá tento projekt ještě složitější na použití.

Jendá se o všestranný framework, užívající nativní volání systému místo emulace.Celkově jde o komplikovaný framework, složité API a horší vzhled.

1.3.0.3 GTK

GTK[18] je Opensource grafický framework psaný v C++, multiplatformní pro počítačovou grafiku. Otestovaný spolehlivý, s obstojným vzhledem, který ale na cílových platformách nevypadá nativně a vypadá i zastarale pokud je použit například pro OS Windows. Framework nabízí pestrou paletu API a funkcionalit, které jsou vhodné pro menší i velké projekty a s méně ostrou učící křivkou. Původně vznikl pro potřeby programu GIMP.

Framework je veliký, a velmi špatně se kompiluje a linkuje, zvláště pokud má spolupracovat s nějakým vývojovým prostředím.

1.3.0.4 Nuklear

Nuklear[19] je extrémně malý a úsporný framework psaný jako jediný hlavičkový soubor C, který nabízí velmi jednoduchou tvorbu GUI, s pestrou škálou funkcí a prezentuje se jako řešení, vhodné pro menší projekty. Framework je přenositelný, funguje na Windows i Linux, je snadno kompilovatelný. Projekt se vyznačuje velmi mírnou učící křivkou, žádné externí závislosti, jednoduchá API volání. Nabízí možnost fungování nad directX, či jinými řešeními, která komunikují s HW.

Framework je malý, neotestovaný, malá komunita, ne moc příkladů užití, špatná dokumentace. Sám je jen nástrojem komunikujícím s jinými frameworky, které řeší vykreslování nebo vstupy.

1.3.0.5 Scitter

Scitter[20] je framework pro cros-platform GUI development. C++ API, nabízí poměrně variabilní tvorbu UI a to kombinací technologií HTML, CSS a JavaScript. Binární soubory jsou k dispozici ke stažení zdarma. Mocný nástroj, menší než Qt, stejná vizuální kvalita a nativní vzhled aplikací. Jednoduché API. Closed source řešení.

K tomuto frameworku je nutná znalost CSS, HTML a JavaScriptu pro vytváření grafické části. Špatná dokumentace. Komunita se zdržuje hlavně na fóru, na který je registrace momentálně omezená.

1.4 Existující Šifrovací knihovny

Výběr knihoven s kryptografickými funkcemi je zaměřený hlavně na rozsah implementovaných algoritmů a jednoduchosti poskytovaného API. Porovnávány jsou hlavně dobře zavedené šifrovací frameworky, které se běžně užívají.

1.4.1 Bouncy Castle

Bouncy castle[21] je Open-source knihovna pod licencí MIT, psaná v Javě a C#. Jedná se o moderní API nabízející šifrování, dešifrování, protokoly výměny klíče a další kryptografické funkce. Projekt vznikl v Austrálii jako náhrada kryptografické knihovny v Javě, jelikož na technologie v USA, konkrétně na kryptografické funkce, platilo embargo.

1.4.2 Openssl

Openssl[22] je open-source knihovna psaná v C od projektu openSSL. Opensource implementace bezpečnostních protokolů, funkcí a metod. Využívaná v rámci opensource projektů i v rámci komerční sféry. Knihovna je otestovaná komunitou i časem, velmi robustní. Funguje na Windows i Linux. Pro užití v tomto programu stačí pouze pro dešifrovací algoritmy, snadno použitelná s podrobnou dokumentací i příklady.

1.4.3 Crypto++

Crypto++ [23] je open-source knihovna psaná v C++, která nabízí rozsáhlé množství kryptografických funkcí. Krom standardních kryptografických funkcí a schémat nabízí také méně známé algoritmy, které ovšem také mají své standardizování a schválení některými světovými organizacemi či státy , nebo implementuje algoritmy z různých kryptografických soutěží, kteří se dostaly do finále (Například šifra RC6).

API je celkem složité, ale dokumentace je k němu obsáhlá, i když obsahuje málo příkladů užití.

1.4.4 LibreSSL

Open-source [24] kryptografická knihovna psaná v C, která vznikla z openssl, po zranitelnosti krvácejícího srdce. Odstraněna je podpora starších kryptografických standardů jako je SSL a plně se soustředí na implementace TLS s tím, že se chce vyhnout chybám, které se objevují v

OpenSSL a odstranit ty části kódu, které by potenciálně mohli být neubezpečené. V současnosti je to primární kryptografická knihovna v systému open BSD.

1.5 Filtry

V rámci přijatého signálu je dobré udělat nějaké zpracování. Obvykle u normálního přijímače je toto řešeno v HW nějakou dolní propustí. To bývá poměrně jednoduchý obvod. Bohužel v rámci SDR je toto ponecháno pro DSP. Digital signal processing musí tyto filtry implementovat jako SW.

Pro zvýšení kvality zpracovaného signálu, hlavně od zdrojů, které nejsou blízko antény nebo jsou třeba za zdí, je potřeba aspoň jednoduchý filtr.

V rámci filtrů je možnost postavení si vlastního filtru, nebo využití již implementovaných řešení v open-source projektech jako je GNU radio nebo RTL_433.

Krom gnu rádia a jeho stavebních bloků jsem nenašel knihovny, které by filtry implementovaly. Nejsnazší způsob nabízí Python v kombinaci s knihovnou scipy a numpy. V rámci jazyků jako je C nebo Java je nutné si tyto filtry naprogramovat sám nebo využít online prototypovacích nástrojů, které vygenerují pseudokód daného filtru.

1.6 Programovací jazyky

V rámci vývojových nástrojů je třeba nejprve zvolit jazyk. Každý jazyk nabízí jiné možnosti vývoje a jeho rychlosti. Kdy nižší jazyky nabízejí větší kontrolu nad systémem, ale potenciálně v nich lze udělat chybu snáze a vyšší jazyky jako Java nabízí rychlejší a jednodušší vývoj, nabízí v základu už řešení některých komponent, jako je například GUI, ale jsou obecně těžkopádnější.

1.6.1 C

Starý programovací jazyk C, který je vhodný pro nízkoúrovňové přenositelné aplikace. Imperativní jazyk bez objektů. Jazyk se hodí hlavně na aplikace, kde je důraz na výkon, a také na psaní operačních systémů. „Jazyk C byl navržen pro a na UNIX os“[25].V jazyce C se dodnes programují například moduly jádra Linuxu. Jazyk je kompilovaný a je k němu velké množství moderních optimalizačních kompilátorů. Tento jazyk je poměrně minimalistický, což může komplikovat některé věci, které jsou jednoduše implementovatelné ve vyšších jazycích. Syntaxe jazyka C je základem syntaxí dalších programovacích jazyků.

Jazyk C je obtížně využitelný pro multiplatformní programování, jelikož nenabízí abstrakce některých systémových funkcí jako je například multithreading, souborový systém a další.

1.6.2 C++

C++ je vylepšení jazyka C, které zachovává vlastnosti jazyka C, ale přidává několik vylepšení. „C++ je lepší C“[26]. Mezi tato vylepšení patří například přetěžování operátorů, přetěžování funkcí, objekty a vše co s nimi souvisí. Templaty a kontejnery od C++ 11 ve standardní knihovně, které mohou odstínit programátora od manuální práce s pamětí, abstrakce, které v ráímci jazyka zlepšují platformní přenositelnost kódu a mnoho dalšího, C++ je živý jazyk, který se mění a stejně tak se postupně mění práce s ním.

Syntaxe jazyka zůstává stejná jako u C, jazyk zůstává poměrně nízkoúrovňový, ale velmi výkonný se spoustou vlastností moderního objektově orientovaného jazyka.

1.6.3 Java

Objektově orientovaný jazyk inspirovaný C++, který se soustředí hlavně na přenositelnost. Syntaxe zůstává podobná jazyku C. Díky své silné multiplatformitě se jedná o jeden z nejrozšířenějších jazyků na světě a také díky jednoduchosti a relativní bezpečnosti.

Velkou nevýhodou je pomalejší běh tohoto jazyka například oproti jazyku C (v některých případech) . Java je jak kompilovaný tak intrepretovaný jazyk. Výsledkem kompilace je bytekód, který běží na virtuálním stroji[27]. Java je díky overheadu virtuálního stroje ,ve kterém běží kód, pomalejší oproti jazyku typu C++ či C, který běží přímo na hardwaru, ale kód je schopen běžet všude, na silných strojích je overhead zanedbatelný.

1.6.4 Python

Python je programovací jazyk, jenž je jednoduše čitelný a pochopitelný [28]. Python je interpretovaný jazyk, takže spíše než programům se výsledek podobá skriptům. Syntaxe jazyka je podobná C s několika změnama pro lepší čitelnost kódu. Jazyk je velmi jednoduchý na naučení a užití, výsledný kód má dobrou přenositelnost. Jazyk ale obsahuje velké množství frameworků a závislostí, je těžký a na slabších systémech pomalý, vhodný je na rychlé zpracování dat a podpůrné skripty.

1.7 Vývojová prostředí OS

Vývojové prostředí se dost odvíjí od použitého jazyka a také preferencí uživatele. Vývojových prostředí je na trhu několik, nicméně zde se budeme držet základního dělení na Linux based a Windows based, kdy každý z těchto operačních systémů nabízí své nástroje na vývoj včetně některých, které jsou multiplatformní.

1.7.1 Linux

Prostředí se spoustou jednoduchý a mocných nástrojů na vývoj a debugování C/C++ aplikací, které například ve Windows nejsou tak kvalitní či jejich užití je horší. Další rozhodující faktor je zkušenost s C/C++ aplikacemi a jejich vývojem převážně na systémech s Linuxem, kdy integrace mnohých knihoven a opensource řešení je daleko snazší než ve Windows.

V rámci vývoje Javy nebo Python aplikací je Linux stejně dobrý jako Windows. V případě Pythonu je nevýhoda platforem založených na Linux ta, že Linux má integrovaný Python 2 v základu, který by se již neměl používat a někdy může být zprovoznění Pythonu 3 problémové.

1.7.2 Windows

Prostředí se zaměřením převážně na uživatelský komfort,pro vývojáře je Windows dělán až druhotně. Vývoj aplikací nižší úrovně jako C/C++ je obtížnější hlavně kvůli nutnosti ručního nastavení a integrace různých knihoven a balíčků a nástrojů pro debugování programů psaných v C/C++.

Pro vývoj v jazyce Java je stejně vhodný jako Linux a pro vývoje v jazyce Python ještě vhodnější, jelikož Python není součástí systému, takže je snazší nainstalovat a využívat jazyk Python bez nutnosti nějaké složitější konfigurace.

1.8 Vývojové nástroje

V této sekci se jedná o představení známých vývojových nástrojů, které jsou k dispozici pro vývoj a psaní kódů. jedná se o klasické nástroje na verzování, automatické sestavení a nástroje pro psaní kódu, označované jako IDE (integrated development environment).

1.8.1 Subversion

Systém správy verzí zdrojových kódů postavený jako náhrada CVS, ale při zachování podobného fungování a odstranění nedostatků a přidání vylepšení oproti CVS. Systém je na rozdíl od gitu centralizovaný. Vhodný pro týmovou spolupráci. Výhodou Subversion je lepší zacházení s binárními soubory, než například u GITu.

1.8.2 GIT

Dnes již známý klasický vývojový nástroj pro distribuovanou správu verzí, původně vytvořen pro účely vývoje linuxového jádra. Dnes běžně dostupný nástroj pro vývoj softwaru, zvláště vhodné je využití je pro spolupráci v týmu. Nástroj není vhodný pro verzování binárních souborů. Umožňuje nastavení automatické generace dokumentace, kompilace atd.

1.8.3 MAKE

Starý automatizovaný způsob umožňující automatické vybudování aplikace od základu, kdy soubor zvaný makefile popisuje kroky, které se stanou po zavolání příkazu make. Make je multiplatformní, na Windows se dá spustit pod linuxovým subsystémem. Jeho podpora v rámci vývojových nástrojů je ale slabší oproti modernější verzi CMake.

Makefile má zastaralou strukturu, jeho obsah připomíná bash skripty, rozlišuje využívání TAB a mezer jako oddělovačů, což někdy může vést ke špatně nalezitelným a opravitelným chybám zvláště u některých druhů textových editorů, které TAB převádí do podoby mezer.

1.8.4 CMAKE

Automatizovaný způsob snadného vybudování aplikace od základu, bez nutnosti někde psát posloupnost příkazů pro kompilování a linkování. Cmake má strukturu připomínající volání funkcí, na rozdíl od Make. Cmake nabízí oproti make i další funkce, jako vyhledávání knihoven a cesty k nim pro linkování. Nástroj má být abstrakcí nástroje Make, jelikož CMake generuje makefily a následně pomocí Make je vykonává. Cmake je nativně podporován nástroji jako je Clion, a podporuje ho i nástroj Visual Studio od Microsoftu a další IDE.

1.8.5 Clion

V současné době asi nejlepší nástroj pro vývoj C/C++ který funguje velmi dobře out of the box, nabízí intuitivní rozhraní, spousty integrovaných nástrojů pro snadnější vývoj a snadné nastavení i zprovoznění. Nabízí nástroje pro refactoring kódu, jeho statickou inspekci, integrovanou nápovědu generovanou z doxygenstyle komentářů, rychlé možnosti oprav kódu, automatické přeformátovaní kódu a integrovaný interface pro version control system.

1.8.6 Eclipse

Vývojové prostředí pro jazyk Java, ale lze tento nástroj rozšířit o další pluginy například o plugin pro jazyk C/C++ a další jazyky. Jednu dobu velmi populární nástroj, který stále má co nabídnout. Při prvním spuštění je třeba tento nástroj nastavit, není automaticky nastavitelný jako například Clion, je nutno vyhledání a instalace správných pluginů a jejich následné nastavování.

1.8.7 MS Visual studio 2022

Multifunkční integrované vývojové prostředí určené pro jazyky C/C++, .NET, Java, Python a další. Vývojové prostředí umožňuje integraci s dalšími frameworky, jako je například unity, nově nabízí snadnou správu balíčků knihoven, které lze instalovat z centrálního repozitáře. Umožňuje integraci s CMake projekty, integruje version control systém typu git a lze rozšířit o další debugovací nástroje.

Prostředí je těžkopádné a nepřehledné, takže je obtížnější s ním pracovat pokud s ním není předchozí zkušenost.

Kapitola 2
Analýza

Tato kapitola se zaměří na analýzu potřebných technologií, prvního rámce využívání frekvenčního pásma 433 Mhz a 868 Mhz, problémy rádiového přenosu, zabezpečení dat, rozdílné druhy senzorů a vybrané knihovny a nástroje pro vytvoření aplikace.

2.1 Rádiové spektrum

V rámci této práce se budu zabývat pouze určitými částmi rádiového spektra. Rádiové spektrum je myšleno ve znění zákona č 127/2005 Sb(zákon o elektronických komunikacích)[29]. Jedná se o elektromagnetické vlny o kmitočtu do 3000 GHz. Tyto vlny se šíří volným prostorem bez vedení. Konkrétně se jedná o ISM pásmo. ISM pásmo je bezlicenční pásmo, tj může ho za předem stanovených podmínek využívat každý. ISM pásmo je mezinárodně stanovené.

2.1.1 ISM

Zkratka ISM je složenina ze tří anglických slov Industrial,Scientific,Medical. Jedná se o rádiové frekvence vyhrazené k volnému užívání. Důvodem k tomuto je problém rušení. Mnoho těchto pásem je rušeno díky využívání různých technologií (typickým příkladem je mikrovlnná trouba, která ruší pásmo 2.4Ghz, které se běžně užívá pro Wi-fi či bluetooth) Tyto pásma stát nemůže pronajímat (licencovat) vzhledem k rušení, tak jsou za stanovených podmínek volně k dispozici.

Pásma jsou dnes běžně využívána k datovým přenosům, vzhledem k specifičnosti zařízení která je ruší a toho, že výskyt těchto zařízení není častý, zařízení bývají stíněná a neběží často (mikrovlnná trouba), dají se využít i pro spolehlivé datové přenosy.

Níže jsou vypsána pásma označena pro použití ISM dle čtú [30].

- 6 765 – 7 000 kHz
- 13 550 – 13 570 kHz
- 26 957 – 27 405 kHz
- 40,02 – 40,98 MHz
- 432 – 438 MHz
- 2 300 – 2 450 MHz
- 2 450 – 2 483,5 MHz
- 2 483,5 – 2 500 MHz

- 5 725 – 5 830 MHz
- 5 830 – 5 850 MHz
- 5 850 – 5 925 MHz
- 24 – 24,05 GHz
- 24,05 – 24,25 GHz
- 61 – 62 GHz
- 119,98 – 122,25 GHz
- 122,25 – 123 GHz
- 241 – 248 GHz

V rámci České Republiky je pásmo 868 MHz, které spadá do ISM v některých evropských státech, brané jako bezlicenční pásmo, ale není ISM, je vyhrazené pro konkrétní účely. V rámci Evropy, toto pásmo je regulováno na úrovni národních vlád. Což je jeden z důvodů, proč toto pásmo není využívané tolik, jako pásmo 433 Mhz.

2.1.1.1 SubGhz vs WIfi

Jednou z velkých výhod subGhz bezlicenčních pásem je dosah, tato pásma mají při započtení omezení, která na ně jsou kladena legislativně, větší dosah než třeba WIFI, a to při stejných podmínkách. Například 433 Mhz má dosah téměř 5x větší ve stejných podmínkách než signál WIFI na 2.4Ghz [31]. Na druhou stranu se počítá s menší přenosovou rychlostí než u WIFI.

2.1.2 Právní rámce ISM

V České republice se o rádiové spektrum, jeho využívání a licencování stará ČTÚ-Český telekomunikační úřad, který vznikl jako nástupce Českého telekomunikačního úřadu, zřízeného dříve.

V ČR upravuje využití rádiového spektra Plán využití rádiového spektra[32] a všeobecná oprávnění[33].

2.1.2.1 433 Mhz a 868 Mhz

Obě pásma jsou v rámci ČR používána pro malé datové přenosy na krátké vzdálenosti. A obě pásma jsou bezlicenční. Obě pásma mají však svá technická omezení. Pásmo 433Mhz se nesmí používat k zvukovým přenosům, jako jsou vysílačky, sluchátka. 868 Mhz má striktnější předpisy a nespadá dle právního rámce Čr mezi ISM pásma[30] přesto, že se jedná o bezlicenční pásmo a je vymezeno pro specifické použití.

2.1.2.2 433 Mhz

- 433 Mhz má omezenšjší dosah než 868 Mhz, často používaná hlavně na senzory s krátkým dosahem.
- v ČR/Evropě je 433 Mhz frekvence je vysílací výkon zákonem omezen do 10mW.
- Kanál je hojně využíván bezdrátovými senzory, takže je nutno počítat se silným rušením v tomto pásmu.
- Kanál je vhodný na kratší vzdálenosti.

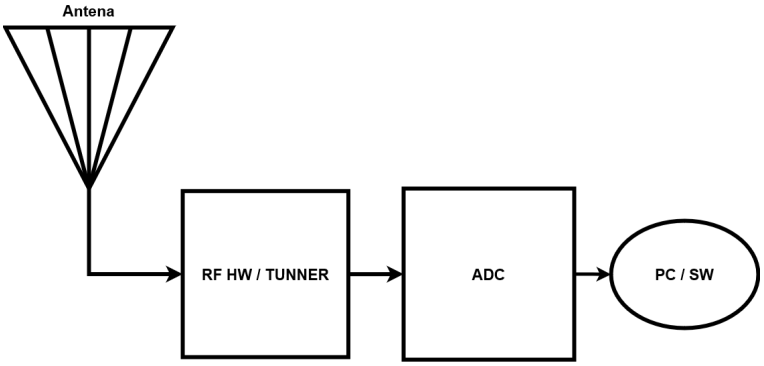
2.1.2.3 868 Mhz

- 868 Mhz má vysílací výkon omezen na 25 mW a má striktnější přepisy užití než 433 Mhz.
- Díky omezení 433 Mhz v Evropě je 868 Mhz vhodnější na větší vzdálenosti díky možnosti většího vysílacího výkonu 25mW.
- Kanál je méně rušený a je vhodnější pro bezproblémový přenos dat.
- Kanál je využíván podobně jako 433 Mhz + dálková ovládání.
- Kanál nespadá do ISM pásem, každý evropský stát si to může definovat po svém.

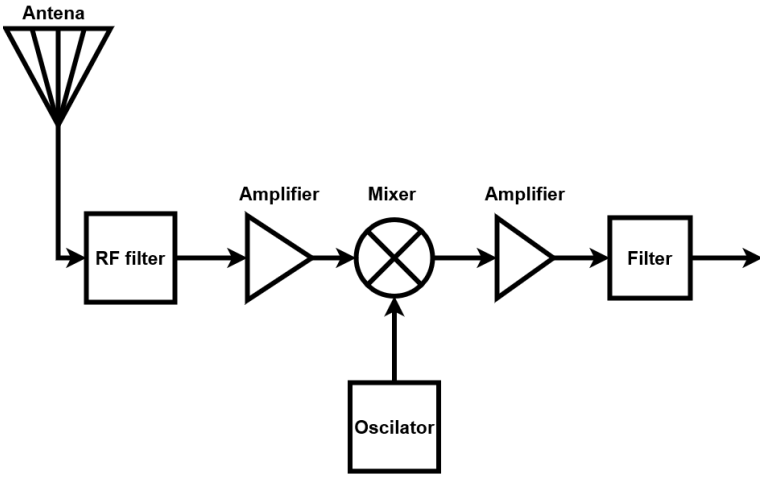
2.2 SDR

SDR neboli softwarově definované rádio je jednoduchý a modulární způsob, jakým lze zpracovávat signál posílaný pomocí rádiových vln, ať již jde o příjem takovéhoto signálu nebo jeho vysílání[34].

Celý HW v rámci SDR přijímače je omezen na minimum. Anténa na příjem rádiového signálu spolu s ADC (analog to digical converter) a nějaký dodatečný HW mezi anténou a převodníkem a výpočetní systém (Osbní počítač, smartphone, FPGA, embeded zařízení). Níže lze vidět rozdíl mezi SDR a klasickým přijímačem.



Obrázek 2.1 Diagram SDR, skládající se pouze z antény, tunneru a analog-digitálního převodníku.



Obrázek 2.2 Diagram superheterodyne přijímače.

Zařízení schopné přijímat a převádět rádiový signál na digitální může stát od stovek korun až po desítky tisíce korun, vše záleží na účelu, výrobci a vlastnostech daného zařízení.

Nejjednodušší verze zařízení však obsahuje absolutní minimum, které umožní signál přijmou a převést do digitální podoby. S digitalizovaným signálem lze pak zacházet různě, filtrovat, zpracovávat a rozšiřovat dle potřeby.

2.3 Technolgie-SDR

Jak už bylo zmíněno v úvodu technolgie SDR, je moderní způsob stavby rádiového přijímače, kdy co největší část samotného přijímače je implementována v programu, místo HW. Jediné povinné součástky, které zůstávají v HW je samotná anténa a převodník z analogu na digitál. Zbytek je poté proveden softwarově. Někdy se jako synonymum k SDR užívá DSP (digital signal processing), obojí znamená to samé, akorát se liší faktor. SDR je většinou v podobně nějakého donglu k PC, zatímco DSP je samostatná jednotka, která vše potřebné má v sobě uzavřené.

2.3.1 Výhody

- Flexibilita - opravení chyby, přidání nové funkcionality se dá získat pouhým stáhnutím nového SW.
- Využitelnost - v rámci SDR lze užít jeden systém pro různá spektra, přeladit SDR je jednoduché, stačí na to jeden příkaz. Případně lze spektra analyzovat, vizualizovat. A to vše poměrně jednoduše na levném HW zařízení.
- Jemné ladění - v SDR není problém jemně ladit až v řádech HZ.
- Životnost - díky podstatné části zařízení v SW není třeba sytém vylepšovat po HW stránce.
- Levný HW - méně součástek co se implementují v HW znamená levnější HW.

2.3.2 Nevýhody

- HW nároky - implementace některých součástí v PC může být problémová po stránce výkonu PC.
- Omezení faktory třetích stran - některé součásti propojení zařízení a PC mohou způsobovat úzké hrdlo, například skrze USB připojení je vzorkování při rychlosti větší než 2400000 vzorků za sekundu nestabilní. Na vině je zde USB připojení.
- Složitost - některé součásti rádia jsou snadno implementovatelné v HW, ale jejich softwarová implementace je složitější.

2.3.3 Historie

V minulosti byla rádia realizovaná pouze pomocí HW součástek a tudíž neexistoval způsob, jak spolu rádia od různých výrobců využívajících různé technologie spárovat, pokud k tomu nebyly vysloveně vyrobená, případně byly potřeba zásahy do samotného HW. Tyto skutečnosti ve výsledku dost limitovaly možnosti využití takového zařízení a také limitovali jejich vzájemné míchání, protože se musí k funkčnosti využívat jeden homogenní systém. Jedna z možností je vytvořit vyměnitelný(modulární) HW, ale opět to přináší velké náklady, navíc výměna takového HW nemusí být uživatelsky přívětivá. Druhá možnost je nechat v HW jen absolutní minimum a zbytek zpracovávat pomocí SW řešení- Softwarově definovaném rádiu.

Obrázek 2.3 RTL-SDR zařízení do usb

SDR neboli softwarově definované rádio je pojem, který se začal objevovat na počátku 90. let 20. století s rozvojem osobních počítačů a nárůstem výpočetního výkonu, který zároveň začínal být dostupnější, mobilních telefonů a miniaturizací. Se stále levnějším a výkonnějším výpočetním HW začala být možnost zpracovávat signál pouze pomocí SW realizovatelná. Myšlenka, že zařízení půjde přenastavit na jiné pásmo, jiná data nebo kompletně jiné využití jen za pomocí změny programu, byla velmi populární.

Velkou roli v rozšíření sehrála organizace Wireless Inovation Forum, spolupracující s IEEE, založená roku 1996. Cílem je zlepšit možnosti bezdrátové komunikace a inovovat v tomto odvětví, a to jak pro průmysl, tak civilní a obranné využití[35].

SDR postupně nabralo na popularitě v průmyslu i v běžném životě. Moderní armádní telekomunikace je stavěna jako SDR, i spousta radioamatérských zařízení začala využívat SDR.Začali tak vznikat programovatelné, versatilní systémy.

Softwarově definované rádio je myšlenka, jak generalizovat použití rádia, princip, jak umožnit zpracovávat radiovou komunikaci modulárním způsobem a nahradit či podpořit hardware implementaci softwarovou implementací, kterou lze upravit, změnit, aktualizovat dle potřeb. Tento způsob je velmi populární i dnes a stále se rozšiřuje jeho využití, a to v průmyslu, armádě a radioamatérství.

2.3.4 SDR a superhet

SDR je, jak již bylo řečeno, dalším evolučním krokem rádia. V současné době patří mezi nejrozšířenější radio přijímače ty, které využívají principu superheterodyne a nebo SDR.

Superheterodyne přijímač je klasické HW rádio, které se skládá z několika součástí. na zachycení a zpracování RF signálu, je levný a má dobré výsledky k požadovanému použití, nevýhodu však je, že použití je velmi omezené, bez fyzického zásahu do obvodů nelze nijak změnit to, co je zpracováváno, případně jak.

Oproti tomu SDR se skládá jen z antény, zesilovače a ADC převodníku, dále je vše možné implementovat pomocí SW. Veškeré filtry a zpracování je možné implementovat v jazyku jako je například C, python a další.

HW část SDR systému poskytuje IQ data, mnohá zařízenía také umí vzorkovat jen I nebo Q data.

2.3.5 Současnost SDR

V současné době se u komerčních zařízení stalo SDR defacto standardem, kombinuje se spolu s klasickými HW řešeními, specializovanými ASIC obvody a FPGA pro DSP. Důvodem je právě jednoduchá rozšiřitelnost, která může být aplikována pomocí sw patche ovladače/firmwaru a v některých případech i levnost daného řešení. Leadrem SDR rozvoje v této části jsou hlavně armáda a elektronický boj[36].

2.3.6 Budoucnost SDR

Předpokládá se strmí růst SDR zařízení, a to hlavně v průmyslu, kdy nové technologie jako 5G, rozšíření IoT, senzorů a dalšího hardware bude klást větší a větší nároky na flexibilitu a dostupnost robustních řešení. SDR se spojí se specializovanými programovatelnými HW obvody jako jsou FPGA a budou vznikat hybridní řešení. Nicméně zde budou překážky ve vývoji nástrojů a potřebného SW, který už v současné době zaostává za HW součástkami a brzdí celkový rozvoj hybridních řešení. Chybějí nástroje pro programování GPP a FPGA jednoduchou cestou[36].

2.4 RTL-SDR

V rámci této práce bude využíváno levného RTL-SDR BLOG V3 2.4 USB zařízení od RTL-SDR.com. RTL-SDR je levný USB dongle o velikosti flash disku, který vznikl z DVB-T tuneru na základě čipsetu RTL2832U od firmy Realtek, u kterých byla objevena možnost přenastavení čipu k jiným účelům. Z těchto tunerů s tímto čipem šla získat nezpracovaná data o průběhu signálu,tzv IQ data. Díky tomu se tento chipset stal základem těchto přijímačů spolu se vytvořeným SW. Chip obsahuje rychlý i když méně přesný analog-digitální převaděč.

RTL-SDR zařízení je pouze schopné přijímat nikoliv vysílat, což je jeden z důvodů proč je toto zařízení samotné tak levné a v současnosti je jedno z nejlevnějších na trhu, přičemž oproti původním DVB-T tunerům nabízí některá rozšíření, ať již nabízející funkcionality navíc jako direct sampling či bias tee.

Ze zařízení lze získat IQ data o velikosti 8 bitů na vzorek s přesností pouze 7 bitů. Což dělá toto zařízení slabší než konkurence, která využívá 16 bitů.

- Maximální stabilní vzorkovací frekvence: 2.4 MHz
- Maximální vzorkovací frekvence: 3.4 MHz (nestabilní)
- ADC: RTL2832U 8-bit
- Proud: 280 mA
- Impendace: 50 Ohms
- frekvenční rozsah: 500 kHz – 1766 MHz
- frekvenční rozsah přímého módu: 500 kHz – 24 MHz

Zařízení navíc umožňuje v případě potřeby vzorkovat pouze I nebo Q signál.

2.4.1 RTL2823U

RTL2823U čip je vysoko-výkonnostní demodulátor pro COFDM (coded orthogonal frequency-division multiplexing) DVB-T s USB 2.0 interfacem, Umožňuje samplovat na IF, low-IF nebo zero-IF výstup díky 28.8 MHz krystalu. Čip má zabudovaný ADC převodník a měl by poskytovat stabilní příjem [7].

Tento čip, jeho vydání a následné zkoumání vedlo k získání velmi levných SDR přijímačů, které se dají zakoupit jako je SDR-RTL V3, nebo zísat použitím levných DVB-T zařízení a správných knihoven.

Obrázek 2.4 celé sestavené RTL-SDR zařízení s anténou

2.5 IQ data

Pojem IQ je zkráceně pro ”in-phase”a ”qadrature”. Čísla, které dostaneme z AD převodníku, tak lze chápat jako komplexní číslo. Složené z I části a Q části. Zvykem je že I signál odkazuje na cosinový průběh signálu a Q na sínový průběh signálu.

Výhodou je, že jakákoliv modulace rádiového signálu může být vytvořena rozdílnými amplitudami. Práce s takto digitalizovaným signálem je poměrně snadná, kupříkladu aplitudu lze získat pouhým umocněním I a Q složky a následným sečtením a odmocněním.

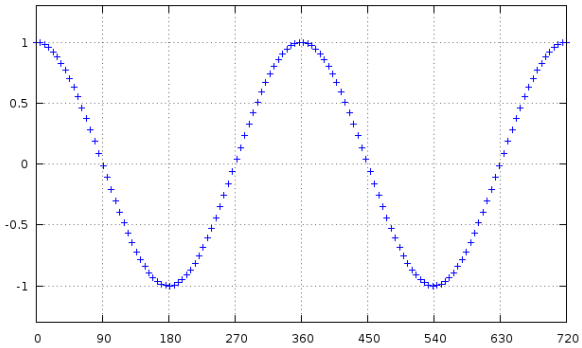
2.5.1 Vzorkování klasicky

Klasické vzorkování signálu je problematické, z několika důvodů. Pokud vzorkuji signál, každý vzorek je bodem na analogovém signálu viz obázek 2.5. Problémem je, že neznáme frekvenci, zjištění frekvence není snadné vzhledem k problému $\cos(x) = \cos(-x)$ [37].

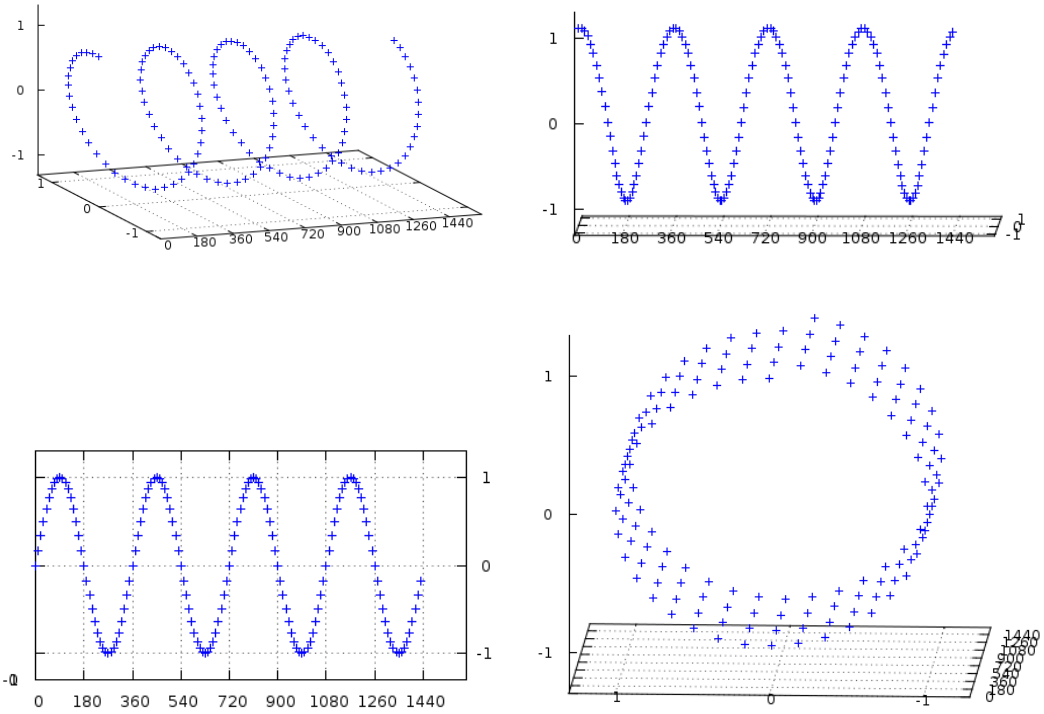
2.5.2 Vzorkování IQ

Jedním z dalších problémů je nemožnost si být jistý amplitudou, jelikož by to vyžadovalo vzorkování při každém vrcholu. Což nemůžeme zaručit. Nicméně IQ data tuto nevýhodu a některé další nemají, a proto se také využívají.

V rámci tohoto je základní představa že ”nevzorkujeme”křivku, vzorkujeme v 3D prostoru spirálu, která podle toho, jak se na ni díváme, dává reálnou část, která mimo jiné reprezentuje klasickou křivku průběhu signálu. Takže v případě čtení pouze této části bychom měli dostat



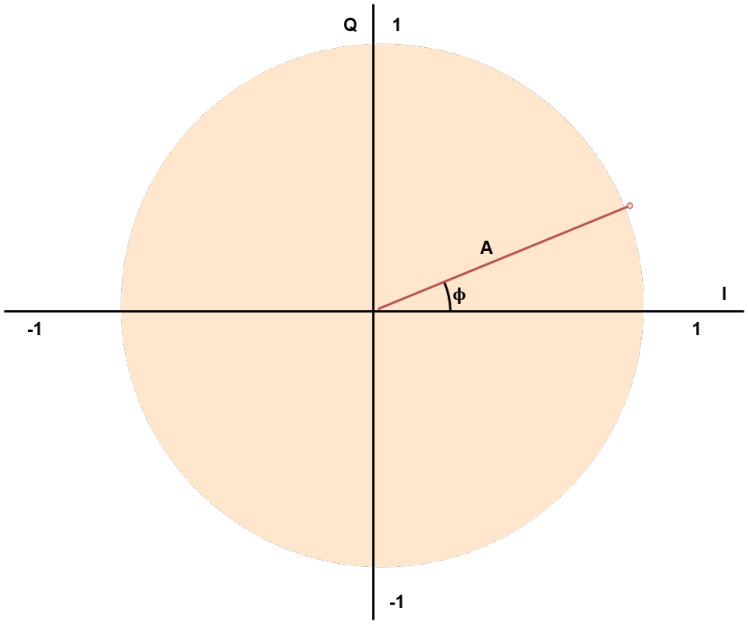
Obrázek 2.5 Graf ukazující průběh vzorků při klasickém vzorkování, převzato z[37]



Obrázek 2.6 Graf ukazující průběh vzorků při IQ vzorkování, celkový 3D pohled, pohled ze strany cosinového signálu (I), pohled ze strany sinového signálu (Q) a pohled zepředu, převzato z[37]

klasický sinusový průběh. V případě že se díváme na ”Imaginární”část, tak je celá křivka pouze posunuta o 90 stupňů 2.6.

Reálná I část:
$I = A * \cos \phi$
Imaginární Q část:
$Q = A * \sin \phi$



Obrázek 2.7 Pohled na IQ data jako komplexní číslo

Když dáme dohromady reálnou a IQ část dostáváme bod na kružnici. IQ data jsou v tomto případě stejné jako komplexní číslo, určené reálnou a imaginární částí 2.7. Díky tomuto lze snadno zjistit vlastnosti signálu, který je přijímaný. Existují na to i specifické metody, které to umožňují získat rychleji a optimálněji, ale pro použití zde stačí tyto základní vzorce.

Amplituda:
$A = \sqrt{I^2 + Q^2}$
Úhel fáze:
$\phi = \arctan Q/I$

Vzorce výše jsou velmi důležité pro demodulaci nosné vlny a získání dat z modulací jako je FSK či ASK. Další výhodou IQ vzorkování je, že některé operace, které by v klasickém případě nemuseli být jednoznačné, jako je skládání signálu, jsou nyní jednoznačné.

2.6 Modulace

V telekomunikaci je modulace způsob jakým ovlivnit vlastnosti průběhu signálu, kterému se říká nosná vlna dalším signálem, kterému říkáme modulační signál, který obsahuje data která chceme přenést. V kontextu této práce nás zajímá pouze takové modulace, na které je možné běžně narazit a dají se považovat za základní modulace. Jedná se o přenos digitálních dat pomocí modulací jako jsou FSK,ASK,PSK. V této práci se nezabývám analogovým přenosem analogových dat jako je FM,AM a PM.

Takovýto signál se šíří poté jako rádiové vlny od vysílače do prostoru a k přijímači.

Pro přenos digitálních dat v analogu (rádiovém signálu) se využívají hlavně dva druhy modulace a to amplitudová a frekvenční.

2.6.1 Modulace pro přenos digitálních dat skrze analogové médium

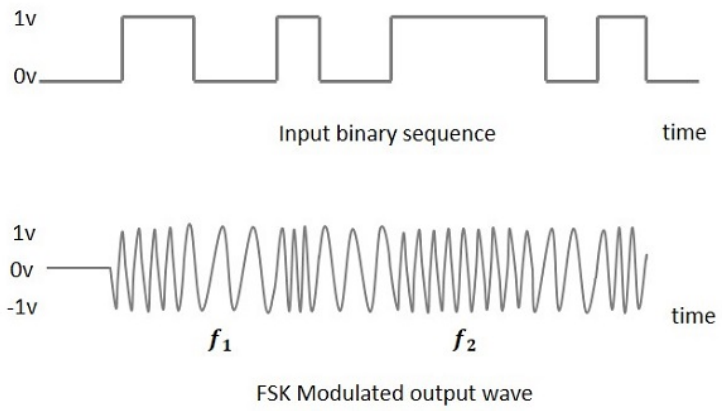
Následující způsoby modulace jsou určené pro přenos digitálních dat po analogovém médiu v tomto případě za pomoci rádiových vln.

V nejsnazším provedení je přenos 1 bit na stav. To znamená, že pokud je stav zapnuto/-vypnuto, tak každý z těchto stavů znamená 1 bit. Pro zvýšení propustnosti je možno využít více stavů pro přenos více bitů najednou v analogovém prostředí, je to teoreticky možno využít nekonečno stavů. reálně se používají mocniny 2 jako počet stavů, čím větší množství užívaných stavů je, tím je menší odolnost proti šumům, jelikož jednotlivé stavy si mohou být velmi podobné a roste nárok na vysílač a kvalitu používaného pásma.

2.6.1.1 FSK

Frequency-shif keying, neboli klíčování frekvenčním posunem je druh modulace nosné rádiové vlny pro přenos digitálních dat. Tato modulace využívá změny frekvence nosné vlny a je poměrně robustní modulací, kde není třeba souvislé fáze pro dosažení skoro optimálního výsledku [38]. Ideálně změny stavů by měly být diskrétní, aby se předešlo mezistavům. Modulace umožňuje vytvořit přenosový kanál. Ve své nejjednodušší podobě bude nosná frekvence měněna tak, aby odpovídala binární jedničce či nule.

Nejjednodušší typ modulace je BFSK (Binary Frequency-shif keying). Kdy je přenášen 1 bit na stav a jsou užívány 2 stavy. 2.8

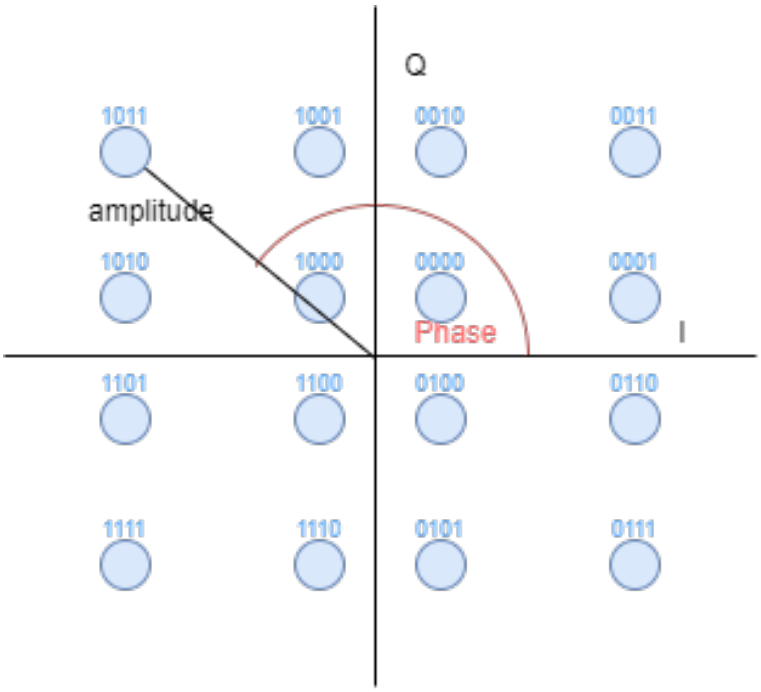


Obrázek 2.8 BFSK modulace na nosné vlně [39]

2.6.1.2 QAM

Quadrature amplitude modulation, je způsob modulace nosné vlny za pomocí dvou signálů, které mají posun o 90 stupňů a sčítají se dohromady. QAM tedy využívá amplitudové i fázové modulace [40]. Toto je stejný princip jako mají IQ data, akorát zde se jedná o výstup, který se moduluje na vlnu. U těchto dvou vstupních signálů se mění jen amplituda. PSK a FSK lze považovat za speciální případy QAM.

QAM má hlavně využití pro přenos digitální televizního signálu a pro použití v rámci ADSL modemů pro analogové telefonní linky. Na obrázku je vidět princip QAM 2.9



Obrázek 2.9 4-QAM modulace

[39]

2.6.1.3 PSK

Phase shif keeying, je způsob přenášení dat pomocí měnění fáze signálu nosné vlny [41]. Tato modulace je vytvářena ovlivněním sinového a cosinového signálu. Využití této modulace je u bluetooth, RFID/NFC zařízení nebo wi-fi. Nejednoduší implementace je BPSK 2.10, v této variantě jedna fáze znamená buď to bit 1 nebo bit 0, pro zvýšení propustnosti lze použít fáze v počtu mocnin 2.

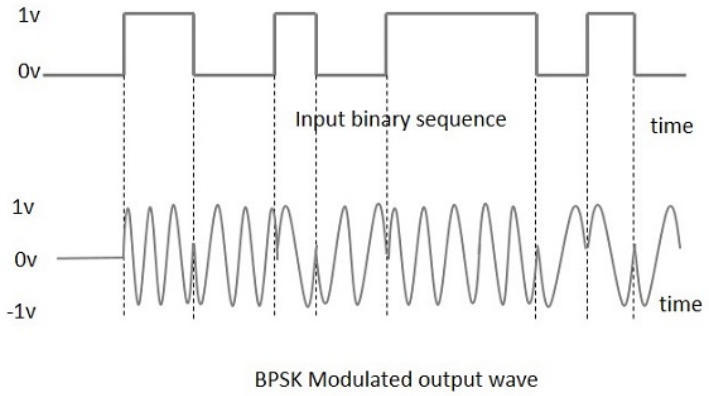
2.6.1.4 ASK

Amplitude shif keying 2.11 je poslední běžná metoda modulace pro RF přenos dat a velmi rozšířená kvůli nízké spotřebě na straně modulátoru [43]. Rozšířená je i u malých a levných vysílačů jako jsou senzory teplot. Jak již název říká, mění se amplituda. Nejjednodušší implementace se nazývá On-Off keying, kdy změna amplitudy znamená změnu 1 bitu, nejčastěji je změněná amplituda považována za 1 a nezměněná za 0.

2.6.1.5 Aplikace

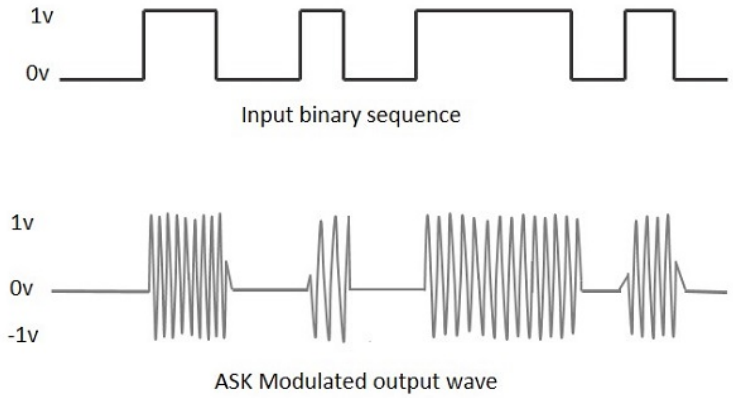
Nejčastěji aplikované modulace pro rádiový přenos v rámci levných a jednoduchých senzorů, se kterými jsem se setkal, jsou ASK a FSK pro svou jednoduchost, jednoduchou implementaci a dobré vlastnosti, která stačí pro přenos malého množství dat na krátké vzdálenosti. Osobně jsem se nesetkal s domácím senzorem, který by používal QAM nebo PSK, co tedy neznamená že neexistují, nicméně v oblasti levných domácích a automobilových senzorů nejsou rozšířenou možností.

U zkoumaných senzorů a v aplikaci se tak budu zaměřovat hlavně na ASK a FSK modulace, a jejich využití.



Obrázek 2.10 BPSK modulace

[42]



Obrázek 2.11 ASK modulace

[44]

2.6.2 Modulace pro přenos analogových dat a digitálních dat po digitálním kanálu

Modulace jako je FSK či ASK většinou nepřenášejí data přímo, pouze umožňují vytvoření přenosového kanálu. U ASK a jeho verze OOK je problémem zjistit, kdy přenos skončil. Kdy byl ztarcen signál, nebo zda je skutečně přijímaná tak dlouhá posloupnost nul.

Nejen z tohoto důvodu se používá druhá modulace, která kóduje digitální či analogová data po již vytvořeném digitálním kanálu.. Nejčastěji se takovéto modulace využívají pro přenos malých binárních dat.

2.6.2.1 PCM

Modulace, která je zmíněna pro úplnost, jedná se o způsob jakým lze digitálně reprezentovat analogové signály. Ve své podstatě jde o to, že signál je vzorkován pomocí uniformě rozdělených bodů a poté kvantován na nejbližší hodnoty[45].

Hlavní využití této modulace je při ukládání hudby, nebo v digitalizovaných telekomunikacích. Jedná se o to jak digitalizovat analogová data (nejčastěji zvuk).

2.6.2.2 PAM

Pulse amplitude modulation, ve svém principu jsou data kódována do amplitudy v nosné vlně v jednotlivých pulzech[46]. Ve výsledku jsou z nosné vlny „odebrány“ vzorky v určitých místech. Teoretické množství přenesených dat jedním pulzem je nekonečné, ale pro digitální data se používají mocniny 2. Příliš vysoké hodnoty přenášení dat v jednom pulzu jsou náchylné na šum.

Modulace PAM je používána v některých druzích Ethernetu.

2.6.2.3 PWM

Pulse width modulation, je jednoduchý princip modulace pomocí šířky pulzu, kdy pulz má stejnou sílu a mění se jen jeho šíře[47]. Teoreticky má velmi jemné modulační možnosti, ale reálně se užívají nižší mocniny 2 a nejčastěji se používají 2 druhy délek pulzů, kdy jeden pulz znamená 1 bit nebo 0 bit. Tento způsob modulace má omezenou propustnost a používá se pro přenos malého množství dat. Typicky je tento způsob užíván v bezdrátových senzorech pohybu nebo je užíván v počítačích jako způsob, kterým jsou hlášeny otáčky ventilátorů.

2.6.2.4 PPM

Pulse position modulation je další jednoduchý princip modulace pomocí vzdálenosti mezi jednotlivými pulzy, respektive postavení pulzů v závislosti na přenášených datech [48]. Pozice pulzů opět může být teoreticky nekonečné množství, ale časté jsou mocniny 2 a velmi často se používají jen 2 vzdálenosti pro jednoduchý přenos binárních dat. Narozdíl od PWM, zde se dá říci, že informace je přenášená pomocí délky mezery.

2.6.3 Kódování Manchester

Způsob kódování dat, který se využívá pro přenos dat na fyzické vrstvě ISO/OSI modelu (ethernet). Toto kódování funguje pomocí hran přechodů signálu[49]. Pokud signál přechází z nízkého stavu do vysokého, jedná se o binární 1, pokud přechází stav z vysokého do nízkého, jedná se o binární 0 či opačně, záleží na použitém typu kódování.

Toto kódování se využívá například u TPMS senzorů od firmy Continental v kombinaci s FSK modulací. Proto je zde zmíněn, jelikož je to další možnost, jak poslat digitální data a označit jasně jejich začátek a konec.

2.7 Bezpečnost Dat

Důležitou součástí komunikačních technologií jakéhokoliv typu je zabezpečení dat. V tomto ohledu zabezpečení dat může znamenat několik věcí, zde se v rámci zabezpečení dat používá CIAAN model. Pokud zabezpečení splňuje model CIAAN jsou data považována za zcela zabezpečená.

2.7.1 CIAAN model

CIAAN model, je moderní model řešící zabezpečení a ochranu dat. Je to slovo tvořené z prvního písmene všech principů, který tento model obsahuje [50].

- C - confidentality
- I - integrity
- A - availability
- A - authentification
- N - non repudiation

2.7.1.1 Konfidentalita

Tato součást znamená, že data, která jsou nějakým způsobem ukládána, nebo transportována jsou chráněná proti neoprávněnému přístupu, porozumění a odhalení[51]. Pokud A posílá data B, poté jen a pouze B je schopen tato data přečíst.Pokud nějaké E odposlechne tato data, je pro něj v rozumném čase nemožné přečíst tato data.

K zajištění tohoto slouží šifrování. Šifry a šifrování umožňují právě, aby poslaná data dávala smysl jen příjemci s tajným klíčem, což je také příjemce, kterého považujeme, že je oprávněn data číst.

V rámci hodnocení zabezpečení dat senzorů bude stačit pokud senzory využívají nějaký šifrovací mechanismus pro zabezpečený přenos dat.

2.7.1.2 Integrita

Data musí být ověřitelná, že došla či byla uložená tak, jak byla vytvořena a nebyla poškozená, ať již z důvodu technické závady nebo zlého úmyslu [52]. Zde je nutno si uvědomit, že i zašifrovaná data nemusí být odolná proti poškození integrity. Kupříkladu textová šifrová zpráva, ve které bylo změněn jeden bit po odšifrování nemusí dávat smysl. Pokud se ale jedná třeba o 32 bitové číslo, nemůžeme si být jisti, zda nebylo během přenosu změněno.

K tomuto účelu může sloužit například HMAC/MAC zpráva, nebo HASH. V po příchodu a dešifrování dat, mohu HASH spočíst a zjistit, zda skutečně je dodržena integrita přijatých dat.

V rámci fyzického přenosu dat se mohou pro zajištění integrity používat i různé detekční, samoopravné kódy či kontrolní součty. Ovšem tyto nástroje nezajišťují integritu dat proti úmyslným útokům, spíše se z nich dá odvodit selhání přenosového kanálu.

2.7.1.3 Dostupnost (Aviability)

Data musí být k dispozici pokaždé, pokud o ně systém či uživatel žádá[53]. Toto je důležité hlavně pro servery, datové trezory a jiné podobné využití. Zajistit 100% dostupnost je defacto nemožné i pro ohromné servery, které mají spousty redundantních součástí a linek.

Zajistit dostupnost bezdrátového senzoru, který funguje na pásmech citlivých na šum, která jsou plná rušení, je velký problém, až neřešitelný, pokud by například zdroj rušení byl blíže, či silnější než samotný senzor.

2.7.1.4 Autentizace

Ověření, nejčastěji uživatele, nebo původce dat, že je skutečně tím, za koho se vydává[54]. Takové ověření může mít různé podoby, uživatelské jméno a heslo, otisk prstu, elektronická klíčenka, nfc kartička a další, v dnešní době kombinace několika faktorů (2 fázové ověření například do banky), elektronická zařízení se mohou autentizovat například fyzicky neklonovatelnou funkcí tzv. PUF.

Autentizaci konkrétního senzoru, nebo rádiového vysílače dat považuji za poměrně důležitou věc, které by senzory měly být schopny.

2.7.1.5 Nepopiratelnost (Non repudiation)

Nepopiratelnost odeslané zprávy. Systém nebo uživatel nemůže popřít odeslání něčeho, nebo provedení nějaké akce[55], typickým příkladem je požadavek bance, kdy banka musí mít důkaz o tom, že zákazník o takovou akci skutečně žádal.

U bezdrátových senzorů by jistě bylo prospěšné kdyby toto bylo implementováno.

2.7.2 Útoky

V rámci útoků na bezdrátová zařízení lze předpokládat všechny běžné typy bezdrátových útoků, jako je denial of service, napodobování daného zařízení a odposlouchávání komunikace[56]. V rámci nezabezpečených senzorů je asi největší hrozbou rušení (denial of service) a odposlouchávání(evasdropping), případně vydávání se za určitý senzor a vysílání falešných dat.

2.7.3 Proč zabezpečit i zdánlivě nevinná data

Otázka zůstává, co jsou nevinná data v tomto kontextu. Data ze senzorů obecně jsou pro vnitřní použití nějakého systému, to že se nám tato data zdají nezneužitelná, neznamená, že tomu skutečně tak není. V době sociálních sítí, kdy se firmy i jednotlivci naučili zneužít sebemenší informaci proti někomu či za účelem zisku. V době, kdy každá stránka, která nepoužívá HTTPS je podezřelá, si myslím, že veškerá data by měla být zabezpečena.

Hypotetickým příkladem budiž senzor teploty, který měří teplotu v kanceláři a v případě přílišného horka zapne klimatizaci, v opačném případě zapne topení. Tento senzor nemá žádné zabezpečení, jen se identifikuje nějakým číslem. Co když nějaká zlomyslná strana si opatří nějaké zařízení schopné napodobovat tento senzor a vysílat vždy špatné informace o teplotách. V krajním případě takto může prodražit kancelářský provoz. Pokud takovýto senzor bude umístěn například v chladícím boxu, může znehodnotit chlazené věci apod.

2.7.4 Nevýhody zabezpečení

Jednou z hlavních nevýhod implementace bezpečnostních protokolů, šifrování a zpracování a ověřování dat je náročnost na výpočetní výkon a paměť, a tudíž na cenu a spotřebu energie. První i druhé je důležité.

Výkonnější HW znamená větší cenu takovýchto zařízení, a to jak senzorů tak jednotky, která tato data přijímá. Implementace zabezpečovacích protokolů může také znamenat větší náklady na SW části senzoru a nižší životnost baterie, což dále zvyšuje cenu nutné údržby[57]. V době soupeření o IoT systémy a chytrou domácnost, kdy firmy soupeří o to, aby se na tomto trhu prosadily, je jedním z lákadel i nízká cena těchto zařízení a běžného zákazníka zabezpečení zajímá velmi okrajově a není ochoten dát kvůli tomu znatelně větší obnos peněz za systém, co zdánlivě bude fungovat stejně.

Druhý implementační problém je spotřeba energie a . Každá operace nad odeslanými nebo přijatými daty je zvýšení spotřeby energie, což u bezdrátových senzorů, jejichž napájení je většinou nějaká forma baterie problém. Problém to činní opět hlavně koncovému uživateli.

2.7.5 Očekávání

Vzhledem k tomu, jak se k bezpečnosti staví zařízení typu IoT, která jsou často sofistikovanější než pouhé bezdrátové senzory, neočekávám velkou úroveň zabezpečení. V případě šifrování dat očekávám šifry typu AES, pokud vůbec data budou šifrována. Identifikace nebo integrita dat bude indikována pouze okrajově. Spíše je možné že se výrobci budou spoléhat na security by obscurity, kdy není zveřejněná modulace a význam jednotlivých částí přenesených dat.

2.8 Senzory

Senzory nebo někdy také čidla, jsou zařízení, která měří nějaký fyzický jev a převádějí jej na formu, kterou může interpretovat stroj či člověk. Příkladem může být teploměr s rtutí, který informaci o fyzické veličině, jako je teplo, převádí na informaci vizuální (zaplnění sloupce rtutí). V rámci této práce nás budou zajímat jen elektronické senzory, a to konkrétně bezdrátové senzory.

2.8.1 K čemu senzory používáme

Senzory všeho druhu používáme k získání námi či strojem vyhodnotitelných dat. Na základě těchto získaných dat se snažíme reagovat adekvátně, abychom dosáhli optimálního stavu nebo se tomuto stavu přiblížili, zvýšení ekonomičnosti a bezpečnosti provozů továren, zabezpečení domovů a komfortu v nich, nebo bezpečnosti přepravy lidí a materiálů, dálková ovládání a zabezpečení.

Ve výsledku se snažíme získat co nejvíce dat z co největšího okolí, získaná data zpracovat a vyhodnotit je. Toto je nezbytné pro pokračující automatizaci ve všech odvětví.

2.8.2 Rozšíření senzorů v době informační

V době informační se množství senzorů, které denně užíváme, dramaticky zvýšilo. Zatímco v době 20. století běžná domácnost užívala hlavně senzory teplotní ve formě teploměrů, v 21. století spektrum užívaných senzorů rapidně narostlo, a to hlavně v domácnostech, ale i průmyslu.[58]

Důvodů k masivnímu nárůstu je několik. Nejhlavnější důvod je dostupnost/cena , jednoduchost instalace, a také to, že se tímto způsobem dá ušetřit poměrně slušné množství peněz či že domácnost díky senzorům se dá ovládat centralizovaně, vždy mít přehled o všem.

Nejen pohodlnost, ale také se do popředí dostává nový termín Ambient Assisted Livin, který umožňuje domácnosti uzpůsobit tak, aby se dalo starat převážně o starší nebo nemocné lidi, a přitom nebylo nutné je svážet na specializovaná pracoviště.[59]

Senzory se takto postupně dostaly všude a jsou nedílnou součástí našich životů. Senzory má již každé moderní osobní či nákladní auto,na základě těchto senzorů fungují asistenti jízdy. Jsou nedílnou součástí systémů alarmů, systémů kontroly teploty v domácnostech, klimatizačních jednotkách, chytrých budov, IoT, domácích asistentů.

2.8.3 Zapojení senzorů

Masivní rozšíření senzorů také přispívá jednoduchost jejich instalace. Vést dráty, je obtížné a někdy i nemožné, zvláště například ve starých bytech nebo kulturních památkách, kde záleží i na estetické stránce věci a neekonomické například pokud se jedná o zabezpečení enormních skladovacích prostor, kde by se vedly desítky metrů kabelů, jen kvůli pár senzorům.

Z tohoto důvodu jsou populární baterií napájené bezdrátové senzory, které přenášejí data v určitých intervalech přes bezlicenční pásma ISM. Díky tomuto je snadná instalace, která nepotřebuje vedení kabelů. Toto řešení však má své velké nevýhody a jednou z těchto nevýhod je špatná bezpečnost.

2.8.4 Radiový přenos

Stejně jako jiné rádiové zařízení, dosah rádiového přenosu je leckdy daleko větší než je potřeba, mimo budovy se dá zachytit senzor pohybu, stejně dobře jako uvnitř. Stejně tak jde i v cizím bytě zachytávat věci z bytu vedlejšího. Narozdíl od kabelu je tedy snadný odposlech, který nepotřebuje narušení nebo blízkost jakýchkoliv fyzických vodičů.

Krom odposlechu ale lze například mást přijímač, který data přijme bez jaké kontroly, což pokud se jedná o nějakou řídící jednotku, může dojít k tomu, že bude vyhodnocovat nevalidní data, případně pokud přijímá nějaké složitější packety, jde s nimi manipulovat a donutit systém udělat něco, co by neměl, v případě že není dostatečně ošetřený vstup, nebo se vyskytuje nějaká chyba při zpracovávání netradičních hlaviček.

Tyto problémy mají hlavně přenosy na frekvencích 433 Mhz a 868 Mhz, kde neexistují dominantní standarty zabezpečující přenos, narozdíl například od bluetooth nebo wi-fi standartů v pásmu 2.4 GHz.

2.8.5 Rušení přenosu

Rádiový přenos lze záměrně rušit různými rušičkami, či se může jednat o rušení od jiných zařízení. S obojím je nutné se vypořádat.

Jak již bylo v dřívějších kapitolách naznačeno, jedním z problémů volných ISM pásem je rušení. Jednak způsobené provozováním specifického vybavení a také tím, že vzhledem k zákonům upravujícím využitelnost těchto pásem a malé množství těchto pásem vytváří husté ”osídlení”.

Senzory musí tedy počítat s výpadky a implementovat opatření, aby vždy jednou za určitý časový interval bylo téměř jisté, že data dojdou v pořádku. Jedním z takovýchto opatření může být pseudonáhodnost vysílání dat. Řekněme vysílací okno, které se bude pohybovat mezi 50 a 90 sekundami od posledního vysílání. Díky takovéto implementaci je zajištěno, že i když dva stejné senzory vyslaly signál shodně , tak je pravděpodobnost, že signál vyšlou shodně i příště malá a tudíž je větší pravděpodobnost že vyslaná data se nebudou překrývat a v pořádku budou přečtena

.

2.8.6 Zranitelnost senzorů

V předchozí kapitole jsem probral bezpečnost dat, tak jak by to ideálně mělo fungovat. Zde chci probrat, jak lze zneužít zranitelné senzory, případně systémy na ně napojené.Může se jednat od rušení až po odposlech a upravování dat.

2.8.6.1 Závažnosti těchto zranitelností

Závažnost zranitelností senzorů je poměrně malá až střední. K tomu přispívá několik faktorů. Prvním faktorem je nutná blízkost k cíli. Vzhledem k omezení dosahu senzorů, zvláště přes zdi, je nutné být od cíle na 50-100 metrů. Je nutná nějaká doba pozorování pro zjištění druhu senzorů a připravení útoku.

Dále senzory většinou nejsou napojeny na kritické systémy nebo vysílají kritická data. Ano i tato data se dají zneužít, ale v krátkodobém hledisku samotná nepředstavují až takovou hrozbu, když uniknou. Mohou ale třeba dát nějaké informace, jako například že v dané budově funguje alarm, nebo v ní nikdo není, protože se tam netopí a podobně. Problém nastává pokud skutečně senzory jsou napojeny na nějakou kritickou infrastrukturu či je jejich vstup součástí nějaké logiky. V tu chvíli stále však platí že je nutno mít potřebné znalosti a být na místě s potřebným vybavením.

Na druhou stranu vybavení na takovýto útok je možné koupit za pár stovek až tisíc korun. A s rozšiřující se úlohou senzorových sítí je možné předpokládat příchod zařízení usnadňující takové to útoky a zároveň nárůst tšchto útoků.

2.8.6.2 Narušení soukromí

V případě řešení, která pracují s citlivými daty, jako jsou dálkové odečty spotřeby energií, vody či měřiče tepla, už je narušení soukromí legitimní problém.

Pokud se k tomu přiedají bezdrátové senzory alarmu, jako jsou PIR, které hlásí pohyb, dá se říci, že mohu sledovat obyvatele ve vlastním domě a zjistí. kde se pohybuje, případně spotřebu bytu, teplotu vytápění, atd.

2.8.6.3 Ovlivnění fyzického světa

Asi největší zranitelností senzorů je ovlivnění fyzického světa, s rozvojem automatizovaných systémů je toto čím dál větší a také závažnější problém. Alarmové systémy ovládající klasické alarmy až po ty požární, mohou napáchat značné finanční škody během minut, pokud buýdou spuštěny se zlým úmyslem. Samotné napodobování teplotních čidel a vysílání falešných údajů může mít ten samý efekt při přílišném vytápění nebo chlazení.

2.8.6.4 Možné dopady do budoucna

Do budoucna se počítá s větší úlohou senzorových sítí, a to jak v průmyslu, dopravě, tak i v domácnostech a zdravotnictví. V rámci masového rozšíření můžeme počítat se zatraktivněním útoků na tyto sítě, jejich odposlech a případně jejich ovlivňování. Zvláště to poslední jmenované může být vážný problém, a je potřeba tudíž zajistit, aby k tomuto nedocházelo.

Zvláště v automobilovém průmyslu, v rámci rozvoje jízdních asistentů a dalších součástí, které již aktivně mohou člověku zasahovat do řízení. je použití nijak nechráněných senzorů problematické. V současných modelech se počítá s velkou chybovostí TPMS, nicméně to se může do budoucna změnit a v takovém případě by se měl změnit i přístup k zabezpečení.

Krom rozšíření již existujících senzorů se počítá i rozšířený nových typů senzorů, které umožní například pacientům se sníženými možnostmi orientace, nebo mentálních schopností zůstávat doma, předpokládám, že takové senzory už budou dodávat data, jejichž zneužitelnost bude vysoká.

2.8.7 Analýza konkrétních senzorů

2.8.7.1 AURIOL

Levný senzor z obchodu Lidl od značky Auriol 2.12, v ceně zhruba 300 Kč. Venkovní senzor napájen 2 bateriemi typu AA, 1.5V, bez uvedené životnosti baterie. Vhodný pro teplotní rozsah od -20°do +60°Celsia. Pásmo vysílání je 433 Mhz s vysílacím výkonem 0.024 W. Přijímač je také napájen 2 bateriemi typu AA [60].

Senzor se páruje se stanicí tak, že se stanice zapne do párovacího režimu a poté zapne senzor, první senzor, který je zachycen, je se stanicí spárován dokud je v dosahu stanice, v opačném případě se pokusí stanice spárovat s novým senzorem.

Výdrž baterie byl stanoven experimentálně pomocí běžného používání na zhruba 6 měsíců v chladném počasí s běžně dostupnými alkalickými bateriemi, životnost přijímače je zhruba 18 měsíců.

Dosah senzoru je silně omezen zdmi, nicméně jde zachytit v celé zděné dvoupatrové budově, pokud je senzor položen u zdi mimo tuto budovu. Hrubý odhad experimentálně, kdy už nejsou data posílána čitelná, je +20 Metrů.

Obrázek 2.12 Senzor teploty a meteostanice značky Auriol

2.8.7.2 GoGEN ME SENZOR 12

Obrázek 2.13 Senzor teploty značky Gogen

Senzor pro meteostanici od firmy Gogen 2.13. Venkovní senzor napájen 2 bateriemi typu AA, 1.5V, bez uvedené životnosti baterie. Vhodný pro teploty od -20°do 70°Celsia. Pásmo vysílání je 433 Mhz[61].

K senzoru není vlastněná meteostanice, nicméně lze senzor nastavit tak, aby šel spárovat s meteostanicí k senzoru AURIOL. Což vede k tomu, že musí být ”protokol”stejný jako u systému značky Auriol.

Ochrana přenášených dat či samotný přenos by měl být natolik stejný, že lze zaměnit tyto dvě značky, což mne vede k doměnce, že se může jednat pouze o rebrandování jednoho produktu.

2.8.7.3 CT60M sonoff PIR2

Obrázek 2.14 Pir senzor

Bezdrátové čidlo pohybu 2.14 za 400 Kč, k alarmu. Mezi přednosti tohoto čidla by měla patřit nízká spotřeba a velká výdrž baterie. Čidlo nezachytává domácí mazlíčky do váhy zhruba 25 kg. A mělo by to být kompatibilní s produkty od konkurence a běžnými domácími systémy[62].

Senzor je napájen 2x AAA bateriemi ty by měly vydržet okolo 2 let ve vnitřních vytápěných prostorách. Komunikuje na frekvenci 433 Mhz. Jinak provozní teploty jsou od -10°až do 50°C s dosahem až 12 metrů, dosah by měl být až 70 metrů ve volném prostoru.

2.8.7.4 ALA51

Bezdrátové čidlo pohybu 2.15, v ceně 800 Kč, vhodné k alarmu a zabezpečení domácností, skladů, továren, obchodů. Nezachytává mazlíčky do 25 Kilo, údajně kompatibilní s velkou částí alarmů na trhu. Dosah detekce až 9 metrů, dosah rádia až 70 metrů v otevřeném prostoru[63].

Obrázek 2.15 Pir senzor

Senzor je napájen pomocí 4 AAA baterií, s výdrží přes 1 rok ve vnitřních prostorách. Přístroj obsahu „smart power saving mód“, provozní teplota je od -10°až do 65°C.

2.8.7.5 Sonoff DW2-RF

Obrázek 2.16 Magnetický senzor na okna

Bezdrátové čidlo otevření dveří a oken 2.16, za 200 Kč, slibuje snadné propojení s existujícími zabezpečovacími systémy. Systém opět napájen 2x AAA monočlánky 1.5 V, slibuje 20 metrů dosahu [64], o výdrži baterie se nezmiňuje.

Čidlo vysílá pouze při otevření (oddálení magnetu). Je doporučeno ho propojit s centrálou, která komunikuje spolu s WI-FI pro integraci do smart domácnosti.

2.8.7.6 GS-WDS07

Bezdrátové čidlo 2.17 otevření dveří a oken, v ceně 200 Kč[65]. Bez návodu. Slibuje kompatibilitu s existujícími systémy, vhodnost pro zabezpečení oken, dveří a skříní. Je napájen jedinou 1.5 V AAA batterií. Dosah signálu nikde není zmíněn, stejně jako předpokládaná výdrž baterie. Tento senzor vysílá při otevření i zavření oken a dveří.

2.8.7.7 Senzory tlaku v pneumatikách značky FORD

Běžný bezdrátový senzor správného nahuštění pneumatik od automobilky značky Ford 2.18. Který hlídá tlak v pneumatikách. Senzor stojí okolo 1200 kč za jeden kus. Jedná se o běžnou výbavu dnešních vozů. Vzhledem k umístění takovýchto senzorů je senzor poháněn lithiovou baterií s výdrží cca 5-10.let podle oficiálních stránek výrobce [67].

Obrázek 2.17 Magnetický senzor na okna

Obrázek 2.18 TPMS značky FORD [66]

Tyto senzory pracují na frekvenci 433 Mhz v Evropě, a jsou uzpůsobeny na velmi krátkou vzdálenost z důvodu blízkého umístění antény ve vozidle, a také pro šetření integrované baterie.

Tyto senzory mají částečně chráněnou integritu dat pomocí CRC[68].

2.8.7.8 E-ITN 30

Senzory firmy Apator jako bezdrátový indikátor topných nákladů, v ceně zhruba 1300 Kč za čidlo topných nákladů. Čidla jsou integrovanou lithiovou baterií, a přibližná životnost této baterie je odhadována na 10 let. Podle manuálu jsou data přenášena na frekvenci 868Mhz bez udání modulace a data jsou šifrována šifrou, která není zveřejněna[69].

Obrázek 2.19 Indikátor topných nákladů

[70]

2.8.7.9 JA-60S

Detektor kouře 2.20 od firmy Jablotron na dvě baterie typu AA, s očekávanou životností 1 roku. V návodu není k zjištěni radiová frekvence, na které systém pracuje, nicméně je zmíněn dosah 100 metrů ve volném terénu[71] z čehož bych usuzoval 433 Mhz.

Senzor má životnost okolo 10 let, konkrétní jednotka která je k dispozici je starší více než 10 let.

2.8.7.10 EZ-7901

Dálkové ovládání a dálkově ovládané zásuvky značky Ecolite 2.21, fungující na frekvenci 433 Mhz [72]. Životnost baterie v ovladači či podrobnosti o radiovém ovládání zásuvky nejsou uvedeny. Zásuvky a ovladač mají „nastavitelný kód“, který umožňuje vybrat kanál, který dané zásuvky využívají. Změna kanálu však není doporučována výrobcem.

Obrázek 2.20 Detektor kouře

Obrázek 2.21 Dálkově ovládaná zásuvka

Kapitola 3
Návrh

Tato kapitola se zaměří na návrh výsledné aplikace, jednoduchého GUI a jednoduchého rozhraní a případné rozšiřitelnosti.

V základu návrhu je důležité si ujasnit jaké funkce aplikace bude mít, konkrétně co aplikace má dělat, jaké má mít funkce a co naopak nejsou její funkce. Návrh samotné aplikace by měl respektovat vybrané knihovny, frameworky a nástroje a respektovat jejich omezení. Výsledkem návrhu bude monolitická aplikace.

3.1 Požadavky na aplikaci

Aplikace musí umět získávat a zpracovávat rádiové signály v pásmech 433 Mhz a 868Mhz.

Nástroj by měl být tvořen knihovnou zpracovávající získaná data a grafickým rozhraním, které budou získaná data vypisovat uživateli v hexadecimální podobě.

Nástroj by měl být rozšiřitelný o nové typy senzorů.

3.2 Funkčnost aplikace

Aplikace musí být schopna spolupracovat se zařízením RTL-SDR V3, nastavit ho a získávat z něj IQ data. IQ data aplikace musí být schopna zpracovat na amplitudu a frekvenci.

Aplikace musí umět načítat data z uložených souborů, nejen ze zařízení RTL-SDR.

Aplikace má být schopna uložit zpracovaná i nezpracovaná data, a to jak IQ data, tak amplitudová a frekvenční do souboru.

Aplikace by měla být dále schopna ze zpracovaných dat vyčíst přenášená data a vypsat v jejich hexadecimální podobě, v případě dopsat o jaký druh senzoru se jedná, pokud je možno senzory rozeznat.

Aplikace by měla umožňovat také dešifrování za předpokladu, že je známa šifra a její klíč.

3.3 Vlastnosti aplikace

Výsledný kód musí být přenositelný mezi operačním systémem Linux a Windows, aby se zajistil požadavek multiplatformity.

Aplikace by měla usnadňovat analýzu rádiového signálu v pásmech 433 Mhz a 868 Mhz.

Aplikace bude mít grafické uživatelské rozhraní.

Aplikace bude umožňovat ovládání RTL-SDR zařízení. Aplikace by měla umožňovat snadné přepínání mezi sledovanými frekvencemi, a to 433 Mhz a 868 Mhz. Zapnínání a vypínání signálu bez nutnosti resetovat danou aplikaci.

Senzory v rámci aplikace by se měly dát rozšířit v případě nového senzoru, senzory a jejich demodulace by se měly dle potřeby dát zapínat či vypínat.

3.4 Výběr programovacího jazyka

Výběr jazyku programování je základní krok, od kterého se dále odvíjí vybrané frameworky, vývojová prostředí a další. K zvážení je jednak vhodnost jazyka a jednak jeho znalost. Přidávat si práci buď nevhodně vybraným jazykem (například v jazyce C psát backend webové aplikace) nebo tím, že je vybrán zcela neznámý jazyk nemá žádnou přidanou hodnotu.

Po zvážení nabízených jazyků se kterými jsem již obeznámen, požadavků jako je mupltiplatformita aplikace i výkonnostních požadavků na zpracovávání přijatých dat v SW jsem se rozhodl pro jazyk C++. Konkrétně standart C++17, který podporuje ve svém standartu moderní platformně nezávislé abstrakce jako například abstrakci souborového systému. C++ je velmi silný imperativní/objektový jazyk, co nabízí kombinaci nízké a vysoké úrovně abstrakce, nabízí kontejnery a mnoho dalšího. Jazyk je kompilovaný a k dispozici je mnoho moderními optimalizačními kompilátorů.

Další výhodou jazyka je multiplatformita standardních knihoven, které lze využít. Příkladem jsou knihovny řešící vícevláknovost, mutexy a další věci, u kterých je v rámci například C těžké dosáhnout přenositelnosti.

Důvodem pro nevybrání pythonu je především vysoká úroveň abstrakce, kdy věci jako cykli jsou velmi pomalé a musí se řešit přes funkce například v knihovně numpy, což by porušovalo požadavek na malé externí závislosti. Jazyky jako java nebo C# nebyly vybrány především z důvodů neobeznámenosti autora s těmito jazyky a tudíž předpokládané vyšší obtížnosti.

3.5 Výběr knihoven

Výběr vhodného frameworku/knihovny se řídí několika faktory, na první místě je vhodnost daného frameworku pro práci tohoto typu, zda nabízí to, co je třeba z funkcionalit, jestli je knihovna přenositelná mezi rozdílnými OS, jak náročné frameworky jsou na výpočetní prostředky a také kompatibilita se zvoleným programovacím jazykem, zda nejsou pro potřeby projektu příliš robustní či složité.

Jedním z kritérií je i to, zda už mám s touto knihovnou nějaké zkušenosti, jak kvalitní je dokumentace, komunita a jak strmá je učící křivka. Roli hraje i to, zda je projekt živý, nebo mrtvý. Zda někdo v projektu opravuje chyby které se objeví a aktivně usiluje o jeho vylepšení.

Každý framework či knihovna však má trošku rozdílný styl hodnocení a výběru, jelikož je něco jiného vybírat knihovnu pro GUI a například knihovnu poskytující šifrovací a zabezpečovací funkcionality.

3.5.1 GUI

V rámci gui jsou hlavními kritérii výběru přenositelnost a jednoduchost na naučení/ovládání. Osobně se snažím vyhnout robustním frameworkům, které umí kdejakou maličkost a jejich užití nebo linkování v rámci projektu je velmi komplikované, případně mají ohromné externí závislosti a nebo komplikovaná API, zdlouhavé dokumentace apod.

Když jsem porovnal všechny frameworky v rámci rešerše, vybral jsem framework nuklear. Důvodů je několik. Hlavním důvodem je jednoduché API, které se dobře učí i přes to, že dokumentace je poměrně krátká. Autor však komunikuje, a ochotně poradí pokud někdo má nějakou otázku.

Celá knihovna je psaná jako hlavičkový soubor v jazyce C, takže není problém s kompilací. Renderovací engine lze vybrat externí dle platformy, nebo konkrétních požadavků, a poté jen přidat konkrétní hlavičkový soubor.

Ostatní frameworky mají poměrně strmou učící křivku, komplikované API, velké závislosti, problém s linkováním, nebo jak je dnes moderní, využívají webový engine, takže vyžadují znalost HTML,CSS a JS. Vzhledem k neznalosti těchto jazyků by zvolení takového frameworku přidělalo mnoho problémů.

3.5.2 SDR

Pro svou jednoduchost, přímost , malé externí závislosti a kompatibilní jazyk C jsem zvolil základní knihovnu librtlsdr od OSMOCMu. Všechny ostatní knihovny staví na tomto řešení, jako svém základu. Knihovna nabízí jednoduché a dobře dokumetované api a binární soubory pro Windows. Ostatní knihovny tuto knihovnu nerozšiřují o žádné využitelné funkcionality v rámci tohoto projektu.

Navíc je projekt stále udržován a rozvíjen aktivní a nápomocnou skupinou lidí.

3.5.3 Dešifrování

Výběr knihovny s kryptografickými možnostmi byl složitý, všechny zmiňované knihovny mají velké množství kryptografických funkcí a šifrovacích algoritmů. Vybrána byla knihovna OpenSSL, je to robustní knihovna s řadou šifrovacích algoritmů. Knihovna má poměrně jednoduchá api právě pro šifrování a dešifrování. Dále s knihovnou již mám zkušenosti, a v neposlední řadě příklady lze najít na komunitní wiki. Projekt je živý a komunita okolo něj velká. JKnihovna je psaná v C.

3.6 Výběr nástrojů

Výběr nástrojů k užívání v rámci projektu byl zaměřen hlavně na to aby zvolené řešení podporvalo vývoj v jazyce C++, jak se daná řešení dobře integrují, jak se dají snadno nastavovat, a jak fungují out-of-the-box.

V rámci vývoje C++ aplikace,je volba vývojového prostředí linuxového typu jasná. Linux jako OS založený na C, má spoustu časem ověřených nástrojů podporující debugování a vývoj aplikací v C a C++. Nástrojů jako je Valgrind, kompilátory jako je GCC, a debugery. Tyto nástroje jsou pro vývoj v C++ lepší a dají se rychle instalovat a užívat, narozdíl od podobných nástrojů na platformě Windows. Jako operační systém byl zvolen debian based systém ubuntu.

Jako IDE byl zvolen produkt Clion, se kterým mám osobní zkušenosti a jsem s ním, jeho rozhraním a out-of-box fungováním více než spokojen. Navíc pordporuje VCS a Cmake.

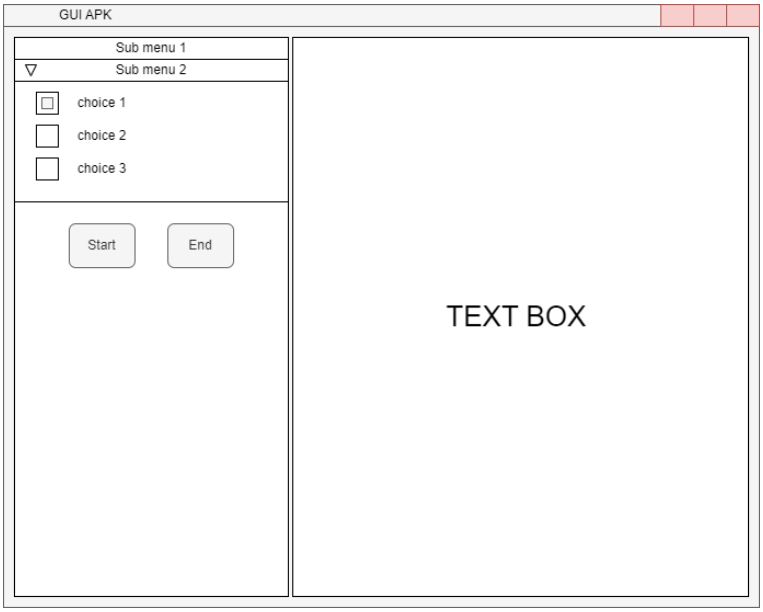
Pro správu verzí byl zvolen git, jelikož s ním již mám zkušenosti a snadno se ovládá a jeho integrace v Clionu je pohodlná. Snadno se nastavuje.

Poslední zvolený nástroj je Cmake, který podporuje Clion a pro jednoduché aplikace je snažší a přehlednější než make.

Pro testování multiplatformity na Windows byl zvolen nástroj Visual Studio od společnosti Microsoft, jelikož tento nástroj slibuje integraci CMake projektu,a tudíž zkompilovat kód pod MSVC v tomto IDE, by mělo být snadné. Nástroj podporuje git integraci, nicméně místy je velmi složitý a nepřehledný. Funkčně je ekvivalentní s Clionem. Toto IDE je zvoleno jen jako testovací nástroj pro multiplatformitu zdrojového kódu a otestování funkčnosti.

3.7 GUI

Hlavní návrh gui 3.1 je dělení na menu část, která vypisuje informace, získaná data v textové podobě. Poměr stran gui, na část nastavení a část vypisující, je přibližně 3:7. Takto je zachován dostatečný prostor pro přehledné menu a ovládací prvky, zatímco výpisová část zůstává přehledná a funkční.



Obrázek 3.1 Návrh gui rozhraní, vlevo ovládací prvky, v právo textbox pro výpis dat

3.8 Popis zpracování dat

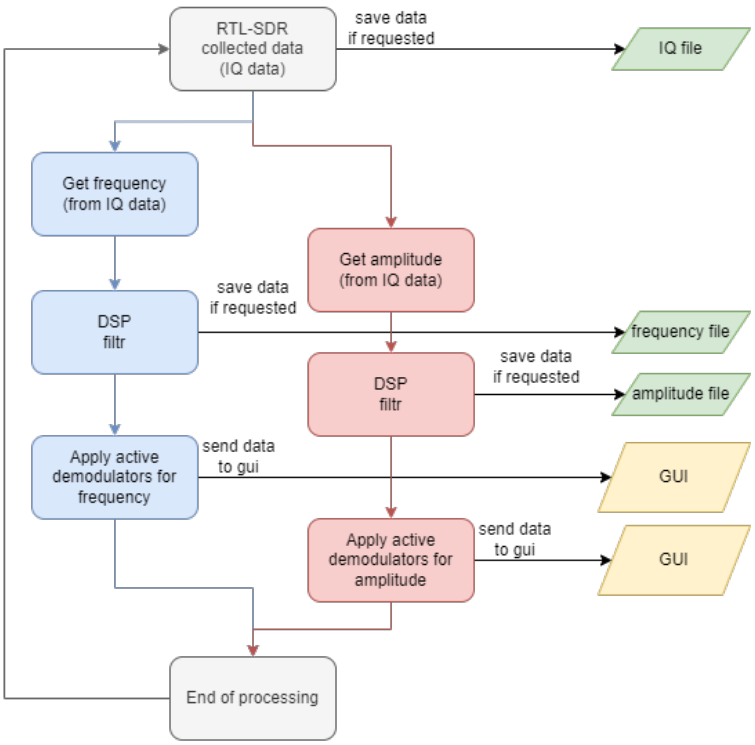
Samotné zpracování dat musí proběhnout uvnitř callback funkce z knihovny librtlsdr.

Aplikace získané IQ samply zpracuje, a to tak, že nejprve převede získané data na dvě posloupnosti a to frekvenční posloupnost a amplitudovou posloupnost. Takto získané posloupnosti zpracují DSP filtry, aby signál byl lehce vyčištěn. Implementace filtrů bude převzata a upravena z projekltu rtl_433 [8]. Následně takto získaná data v případě potřeby uloží do souboru či aplikuje demodulátory dat. Případné výsledky boudou odeslány do gui. Názorně je to zpracováno na grafu 3.2.

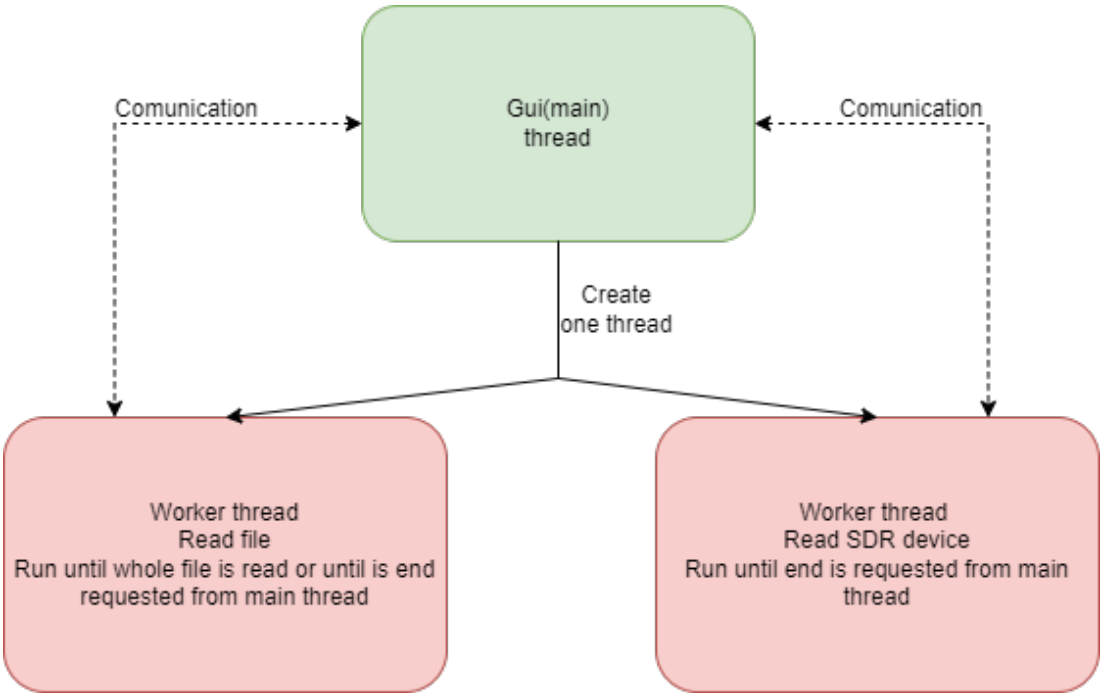
3.9 Vlákna

Návrh aplikace počítá s rozdělením na dvě vlákna 3.3. Jedno vlákno hlavní, starající se o správu pracovních vláken a vykreslující a spravující GUI a příkazy z něj, a druhé pracovní vlákno, které bude mít na starosti pouze vytahovat data z RTL-SDR, případně z souboru a zpracovávat tato data. Vlákna spolu budou komunikovat pomocí třídy, která zajistí bezpečnou mezi vláknovou komunikaci.

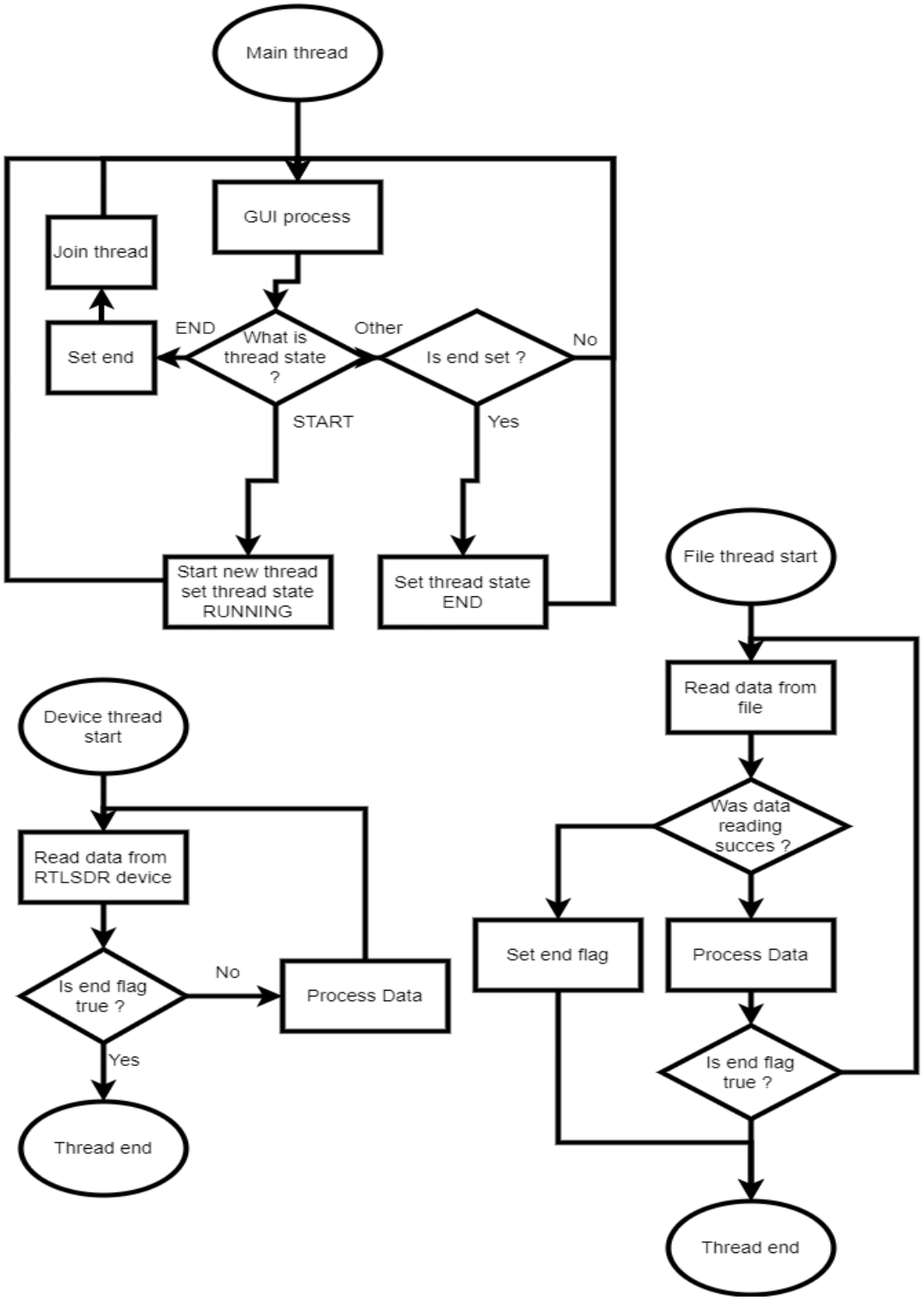
Hlavní vlákno bude mít na starosti start a ukončení pracovních vláken, a dozvědět se o ukončení vláken, pokud vlákna skončila očekávaným způsobem 3.4.



Obrázek 3.2 Zpracování dat uvnitř callback funkce



Obrázek 3.3 Diagram zobrazující vlákna a vztahy mezi nimi.



Obrázek 3.4 Diagram zobrazující životní cykly jednotlivých vláken.

3.10 Přidání nových senzorů

Přidání nového senzoru, s modulací, kterou již aplikace zvládá by mělo být snadné. Možnosti jsou dvě, buď to přidávat pomocí textových souborů, do kterých lze napsat informace o senzoru, třeba ve formě CSV, nebo ve formě hlavičkového souboru, jako instance třídy, které budou seřazené v globálním statickém poli.

První přístup má výhodu, že není třeba znovu kompilovat kód, aby se dal přidat nový druh senzoru, prostě se přečtou dodané soubory v rámci adresáře a informacemi z nich se naplní objekty, které se pak budou používat jako reprezentace jednotlivých senzorů.

Nevýhoda takovéhoto přístupu je vytváření nějaké třídy nebo funkce, co daný soubor bude číst, poté parsovat a nakonec s nimi naplní příslušné objekty. Dalším problémem je, že bude problém s komentováním souboru, obsahující takovýto popis. Toto bude muset zvládat řešit funkce nebo třída která soubor bude číst.

Druhý přístup má výhodu, že všechny problémy se čtením souboru, odstraňováním komentářů a podobně, už řeší kompilátor a preprocesor, autor jen vyplní potřebné údaje do konstruktoru, a přidá daný senzor do globálního statického pole.

Nevýhoda tohoto přístupu je nutnost překompilovat aplikaci po každém přidání nového senzoru.

3.11 Shrnutí

Aplikace je navrhnuta jako nástroj pro analýzu rádiového přenosu v ISM pásmech pro bezdrátové senzory. GUI aplikace byla navrhnuta po vzoru již existujících aplikací. Aplikace umožní čtení již naměřených IQ dat ze souboru, jejich uložení ve formě amplitudy nebo frekvence. Aplikace dále umožní demodulaci získaných dat, pro která je již zavedená demodulace a případné dešifrování těchto dat.

Kapitola 4
Řešení

V rámci této kapitoly se bude rozebírat již hotová aplikace, jak je tato aplikace řešena, jaké třídy a nezatříděné funkce jsou použity.

V rámci řešení byla napsána monolitická multivláknová aplikace s grafickým uživatelským rozhraním v jazyce C++. Snaha v rámci řešení byla hlavně o snadnou přenositelnost a vyhnutí se přílišným externím závislostem, jak již bylo diskutováno v návrhu.

4.1 Datové typy

V rámci tvorby aplikace byla snaha vyhnout se užívání kontejnerů, z důvodů problému, které činí míchání kontejnerů a referencí s pointery pro zpracování v rámci knihoven psaných v C. Toto se ne všude podařilo a na mnoha místech je použita například třída string. Nicméně v rámci polí kontejnery užity nejsou a jsou místo toho užity statické pole nebo dynamicky alokovaná.

V rámci řešení také byla snaha o použití jednodušších datových typů. nejčastěji se jedná o použití float místo double, kdy není přesnost až takovým problémem.

4.2 Třídy a funkce

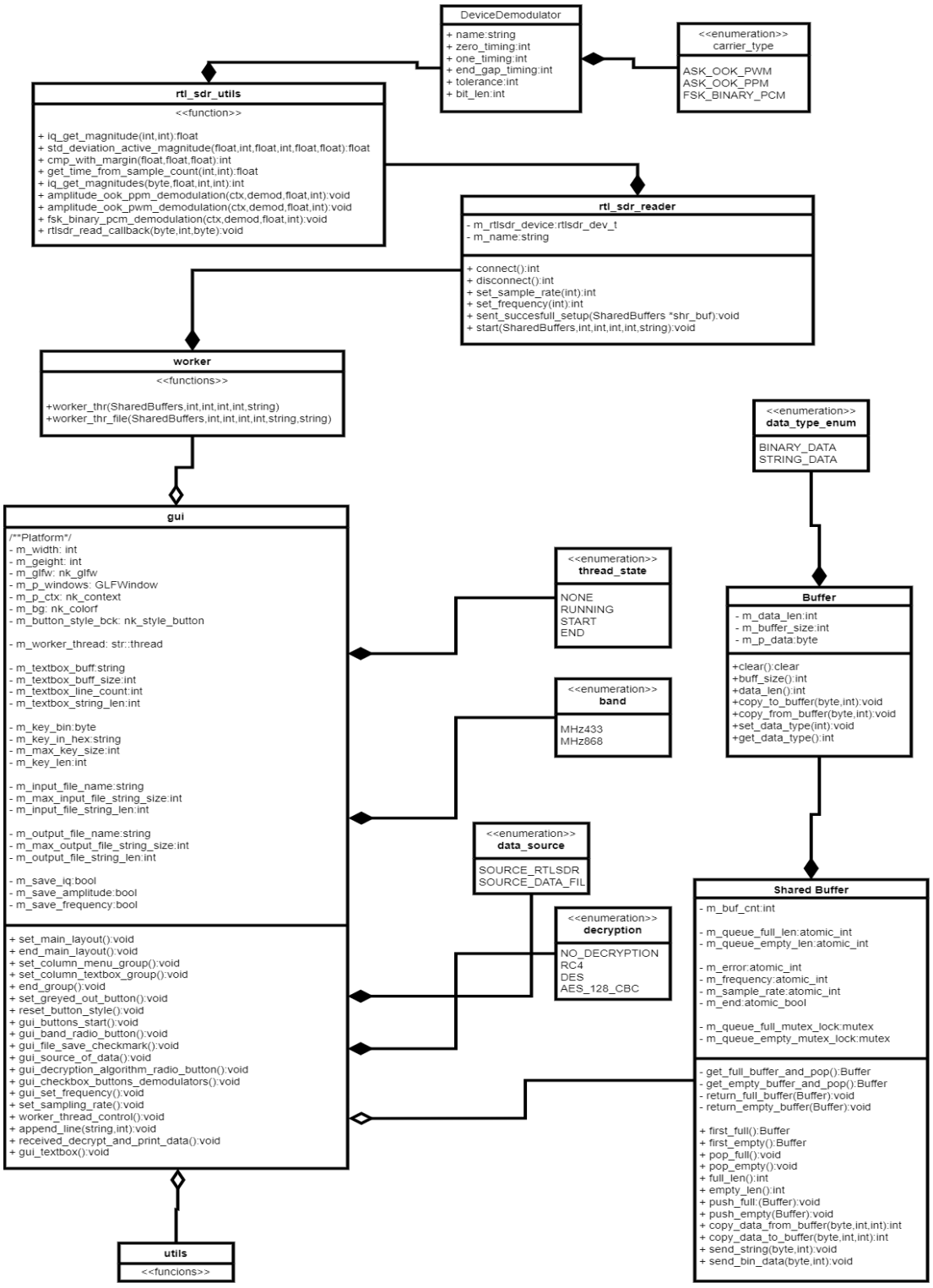
Hlavní část aplikace bude rozdělena do několika spolupracujících tříd a funkcí 4.1. Gui třída je navržena tak, že ji lze zcela odstranit a vyměnit za jinou implementaci. Provázanost ostatních tříd je však poněkud těsnější. Přístup kombinuje imperativní funkce a objekty.

4.2.1 Gui

Třída spravující GUI rozhraní, operace nad knihovnou Nuklear a výpisy dat předané z pracovního vlákna aplikace, kromě GUI rozhraní také tato třída ovládá vytváření a spouštění pracovního vlákna a jeho následné ukončování. Tato třída běží ve svém vlastním hlavním vlákně.

Gui vlákno má mít na starosti také malé předzpracování zobrazovaných dat, v tomto případě se jedná o převod binárních dat do hexadecimální podoby nebo dešifrování přenesených dat. U obojího se jedná o malé a rychlé operace nad daty, které by měli dávat dohromady řádově desítky bajtů.

V rámci návrhu třídy se pak o zobrazování budou starat zobrazovací metody.



Obrázek 4.1 Diagram tříd a funkcí, inspirovaný UML diagramem, problém je s částmi souboru, který je psaný jako C funkce, jelikož UML diagram pro to nemá konstrukt.

4.2.2 SharedBuffers

Třída starající se o předávání dat mezi gui vláknem a pracovním vláknem. Třída by měla umožňovat bezpečnou komunikaci mezi vlákny, a být uzpůsobená tak, aby byla chráněna před problémy jako je data-race a deadlock.

Hlavním účelem třídy je mít frontu několika bufferů, které se plní a předávají do jiné fronty hlavnímu vláknu ke zpracování. Dalším účelem je předávání dalších dat, jako jsou nastavení zařízení apod.

4.2.3 RTL_sdr_reader

Třída řešící práci se samotným zařízením rtl_sdr, třída řeší nastavení samotného zařízení, a spuštění asynchronního sběru dat ze zařízení, až do doby, než je vyslán příkaz ke zrušení.

4.2.4 RTL_sdr_dev_context

Třída sloužící pro předávání stálého kontextu v rámci jednotlivých volání callbacků. Cokoliv má být stálé a předávat se mezi jednotlivými voláními callbackové funkce v rámci asynchronního čtení dat, musí být k dispozici z této třídy a již inicializované při předání asynchronní funkci.

Tato třída je hlavní součást vlákna, které čte data přímo ze zařízení.

4.2.5 Bit

Jednoduchá třída řešící sestavování odposlechnutých bitů do bajtů.

4.2.6 File_read

Třída, která řeší alternativní mód aplikace a to čtení IQ dat ze souboru místo čtení těchto dat ze samotného zařízení.

Třída je hlavní součástí alternativního vlákna, které řešeí čtení dat ze souboru.

4.2.7 File_save

Třída na ukládání surových naměřených dat či zpracovaných dat jako je amplituda a frekvence. Třída poskytuje odkazy na výstupní streamy a řeší otevření i uzavření souborů.

4.3 nezatříděné knihovní funkce

Funkce, které nejsou součástí žádné třídy z důvodu nevhodnosti nebo snazší kompatibility se zvolenými knihovnami.

4.3.1 Rtl_sdr_reader_util

Část kódu, obsahující funkce pro zpracování signálu a demodulaci známých zařízení včetně callbacku pro asynchroní sbírání a zpracování dat ze zařízení RTL-SDR.

4.3.2 Utils

Část kódu obsahující pomocné funkce hlavně pro výpisovou část GUI jako jsou převody z bajtů na hexadecimální reprezentaci či obráceně.

4.4 Makra

Pro konstanty jsem se rozhodl využívat speciální soubor obsahující všechna programátorsky definovaná makra, která fungují jako konstanty v kódu. Cílem tohoto souboru je pouze zvýšit přehlednost, tento soubor obsahuje pouze konstanty, které jsou užívány v RTL-SDR části a neobsahuje konstanty pro třídu GUI nebo třídu SharedBuffers.

4.5 Decryption

Kolekce hlavičkových souborů, které nabízejí funkce na dešifrování a stojí nad openSSL knihovnou. Případné další algoritmy je nutno přidat ručně. Implementovány byly algoritmy RC4,DES a AES v CBC modu.

4.6 Přidání nových senzorů a modulací

Přidání nových senzorů je řešeno skrze statické globální pole, statické inicializace objektů reprezentující popsání demodulátorů senzoru, kdy každý senzor má svůj hlavičkový soubor.

Přidání tímto způsobem bylo zvoleno vzhledem k možnosti využití existujících vlastností kompilátorů a preprocesorů, které umožňují přidávat komentáře k hlavičkovým souborům. Dále také relativní jednoduchosti oproti importování nebo psaní parseru souborů, které by bylo potřeba číst.

Výsledný objekt typu DeviceDemodulator se musí přidat do souboru devices_collection.h kam musí být vložená hlavička, a dále samotný statický objekt do statického pole demodulátorů a přidána hodnota do statického pole, které určuje, které demodulátory jsou automaticky vybrány po zapnutí aplikace.

4.7 Shrnutí

Vytvořená aplikace odpovídá návrhu, při vytváření byl kladen důraz na využíváni jednoduchých datových struktur a omezení používání kontejnerů. Všechny floating point proměnné jsou typu float.

Výpis kódu 4.1 Třída objektu zařízení - demodulátor

```
   class DeviceDemodulator{
public:
    DeviceDemodulator(  /* name of the sensor */
                        const char *dev_name,
                        /* tyope of carrier form enum of carriers */
                        carrier_type dev_carrier,
                        /* if applicable give time in microsecond
                        to represent bin 0 */
                        unsigned dev_zero_timing,
                        /* if applicable give time in microsecond
                        to represent bin 1 */
                        unsigned dev_one_timing,
                        /* add end gap, after this time, bit seqence
                        will be considered ended */
                        unsigned dev_end_gap_timing,
                        /* time tolerance, be careful to avoid
                        ambiguous definition.
                        50 for 0 and 100 for 1 with 50 tolerance
                        and you get 75 that is undefined behavior */
                        unsigned dev_tolerance,
                        /* Length of total bits transmitted
                        in one transmission */
                        unsigned dev_bit_len
                      );

    const char* name;
    const carrier_type carrier;
    const unsigned zero_timing;
    const unsigned one_timing;
    const unsigned end_gap_timing;
    const unsigned tolerance;
    const unsigned bit_len;
};
```

Výpis kódu 4.2 Ukázka přidaného zařízení

```
  #include "device_demodulator.h"

const DeviceDemodulator pir ("generic␣pir␣sensor",
                                    ASK_OOK_PWM,
                                    1300,
                                    420,
                                    2500,
                                    450,
                                    25
);
```

Výpis kódu 4.3 Hlavičkový soubor devices_collection.h s ukázkou přidaných zařízení

```
  static DeviceDemodulator const devices_demodulators[] = {
        /*!Add devices to this static array*/
    temp_sensor,
    pir,
    door_sensor_open,
    generic_window_sensor_open,
    tpms_ford,
    remote_sockets
};

/*!Default activated demodulators
the length of demodulators_usage must be same as
devices_demodulators */
static int demodulators_usage[] = {0,0,0,0,0,1};
```

Kapitola 5
Testování

Tato kapitola se zaměří testování nastavení zařízení SDR, testování funkčnosti aplikace, testování GUI a testování na skutečných senzorech a vyvozování závěrech o nich, a zda se senzory dají přidat bezproblémově.

Testování je důležitá součást vystavění aplikace a její funkčnosti. V rámci testování se chci zaměřit hlavně na funkčnost aplikace, stejně tak na to, zda jsem schopen opravdu odposlechnout senzory, které mám k dispozici.

5.1 Testování aplikace

Otestování aplikace jako takové je možno dělat 2 způsoby pro zachování funkčnosti, unit testy, které nejsou tak rozšířené pro C++ aplikace za pomoci frameworku jako je boost, nebo ruční testy pro zajištění optimální funkčnosti aplikace.

Vzhledem k potřebě otestovat apliakci jako celek, jsem zvolil ruční testování s debugovacími výpisy, které z finální verze aplikace byli odstraněny.

5.1.1 Testování funkčnosti

Základní testování je zda vše funguje jak má z programovacího hlediska, při tomto testování je odstíněno GUI a testování probíhá pomocí logovacích výpisů, které jsou poté porovnány s očekávanýma hodnotama a zda vše odpovídá, případně zda jsou volány správné metody a funkce.

Testování zahrnovalo spuštění pod nástrojem valgrind pro kontrolu leaků při dynamické alokaci a testování jednotlivých funkcí zda vracejí očekávané výsledky pro dané vstupy.

Poslední součást testování bylo spuštění aplikace nad souborem již (ručně)dekódovaných dat a porovnání s výsledky z aplikace, zda vše souhlasí .

5.1.2 Testování GUI

Základní testování, zda gui funguje jak má, probíhalo pomocí debugovacích výpisů, kdy kliknutím na ovládací prvek bylo vždy vypsáno, jaký ovládací prvek byl použit a jaká je jeho nová hodnota.

Když UI bylo otestováno po funkční stránce, bylo UI předvedeno několika testovacím subjektům, kteří si mohli ozkoušet práci s tímto UI na jednoduchém zadání. V rámci tohoto testování se ukázalo, že UI funguje jak má, je vcelku přehledné a spolupracuje dobře s aplikací.

5.1.2.1 Ohlasy

Gui bylo představeno několika testovacím subjektům, konkrétně sedmi, kteří ho měli zkusit použít a zhodnotit. Vzhled GUI byl hodnocen kladně, jako přehledný a graficky příjemný. Velkým plus byl zatmavený-šedivý vzhled, jelikož dost připomíná dnes hodně populární dark mode.

Problematické se ukázalo zaškrtávaní demodulátorů, kde je těžké rozlišit jaké demodulátory jsou zapnuté a jaké vypnuté. Toto je jen problém prvního dojmu, který uvedl jeden z 7 testujících. A neshledávám ho vážným nedostatkem GUI návrhu.

5.1.3 Testování přenositelnosti kódu

Přenositelnost kódu se ukázala jako problematická. Testování mělo proběhnout tak, že je přenesen Cmake a nezměněný kód do nástroje Visual Studio a tento nástroj zkompiluje a slinkuje výsledný program. Tento test selhal, jelikož integrace CMake není v rámci Visual Studia není tak snadná a převést projekt z Linux do Windows také přináší své vlastní problémy hlavně na straně využívaného kompilátoru.

Tento test selhal a bylo nutno upravit i kód pomocí podmíněných macro bloků s WIN32 makrem a dalšími úpravami. Dále bylo nutné Cmake projekt převést do projektu typu visual studia. Tento převod byl proveden ručně. Po úpravách programu a vytvoření projektu pro Visual Studio již projekt je bez problému přenositelný.

Krom samotného vytvoření nového projektu pro Visual studio se ukázal také problém s akceptovatelnou velikostí lokálních statických polí, kdy Windows hlásil při volání callback funkce přetečení stacku kvůli příliš velkým statickým polím.

Tento problém byl vyřešen výměnou velkých lokálních statických polí za dynamicky alokovaná pole v kódu.

5.2 Seznámení se s RTL-SDR

Prvním krokem k vytvoření funkční aplikace, která stojí na zařízení jako je RTLSDR, je zjištění jak tato zařízení správně nastavit. Díky knihovně od osmocomu máme k dispozici několik nastavení daného zařízení pomocí volání jednoduchých funkcí a také funkce automatického nastavení.

5.2.1 Anténa

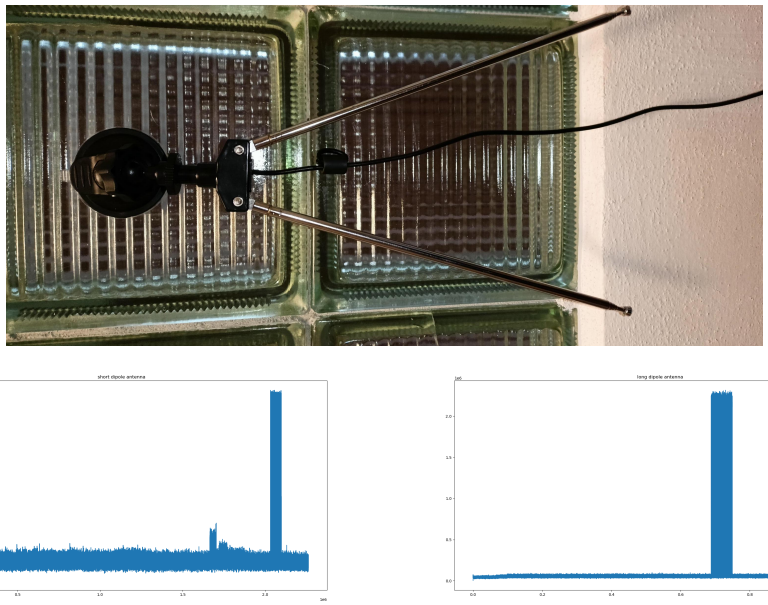
K zařízení je dodávaná klasická dipólová anténa, s dvěma délkami. Využití větší antény dává čistší signál ze senzorů 5.1.

5.2.2 Nastavení RTL-SDR

5.2.2.1 Frekvence a vzorkování

Nastavení frekvence je přímočaré, lze nastavit cokoliv v daném rozsahu pomocí funkcí, nastavit 433 Mhz nebo 868 Mhz není problém a je to přímočaré volání funkce. Rychlost převodníku, neboli sample rate, se nastavuje hůře. Je potřeba experimentovat s tím, jaké nastavení lze použít.

Čím více samplů za sekundu, tím lépe, zde neplatí, jelikož jsou pak kladeny přílišné nároky na výkon zpracování a rychlost HW. Zdá se že frekvence je volně nastavitelná mezi 1 Milionem samplů za sekundu a více, pod 1 milion samplů nastavení nefunguje jak by bylo očekáváno, je zde velká mezera nenastavitelných hodnot. Dále nastavit hodnota mezi 230 kilo samply a 300 Kilo samply za sekundu. Experimentováním jsem zjistil, že hodnota 250 000 samplů za vteřinu je dostatečná pro účely mé aplikace.



Obrázek 5.1 Dipólová anténa a grafy ukazující amplitudy s rozdílnými anténami, krátkou vlevo a dlouhou vpravo

5.2.2.2 AGC

První funkcionalitu, kterou zařízení má, je AGC mód. Teoreticky tato funkce umožňuje automatický gain control, což by mělo umožňovat zisk největšího Signal to noise poměru, problém je, že je tento mód vychází z čipu, který je nastaven pro televizní vysílání a pro jiné druhy signálu naopak tento poměr může zhoršovat, proto bylo nutno tuto funkci testovat.

Testování odhalilo, že funkci je lepší ponechat vypnutou, AGC nefunguje v rámci toho, co je od SDR potřeba tak, jak bychom očekávali, výsledný signál je v horší kvalitě a hůře předvídatelný pro zpracování což je vidět na tomto grafu 5.2.

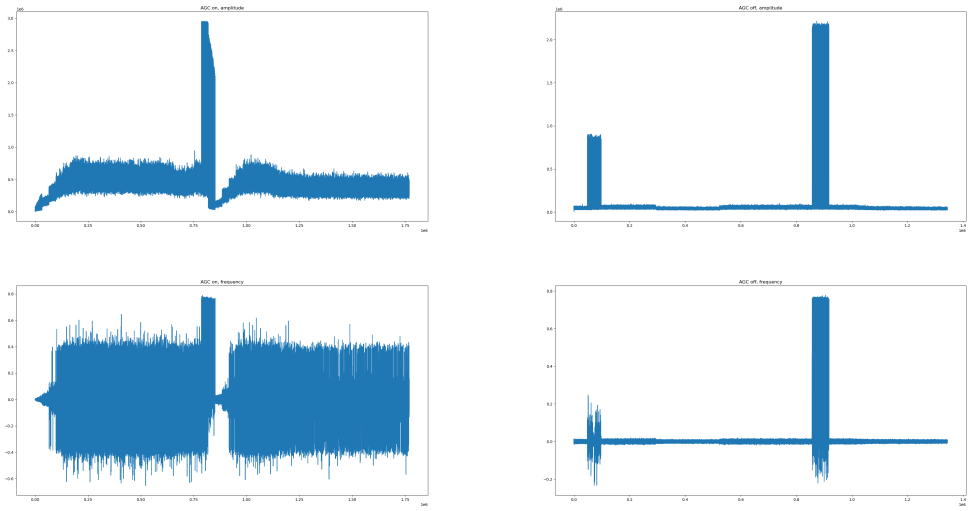
5.2.2.3 Tuner bandwidth

Nastavení šíře pásma má automatickou volbu, která funguje dobře, v tomto případě je dokonce doporučeno používat automatické nastavení šíře pásma, které se upraví k vhodné sample rate. V tomto případě nebyl důvod jakkoliv experimentovat. V teorii je možné takto dosáhnout filtrování horní propusti 5.3.

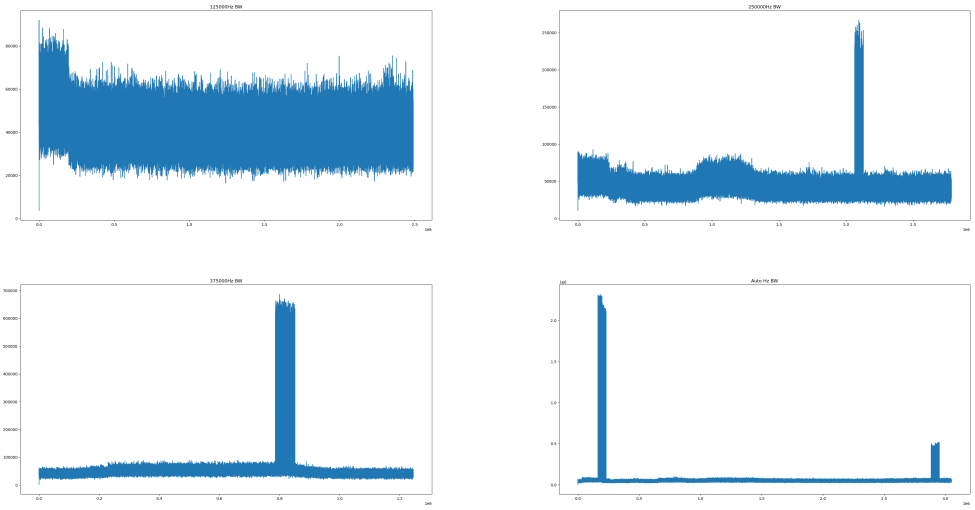
5.2.2.4 Tuner gain

Nastavení správné citlivosti vstupního signálu je důležité pro získání co nejlepšího SNR (signal to noise ratio) . Nastavení takové hodnoty se však může měnit v závislosti na okolí a okolních podmínkách. Přístupy jsou dva, buď manuálně nastavovat podle potřeby, nebo nastavovat automaticky. Manuální nastavení bude lepší než automatické pro dané podmínky, pokud se ale podmínky budou měnit, bude vždy nutná změna tohoto nastavení.

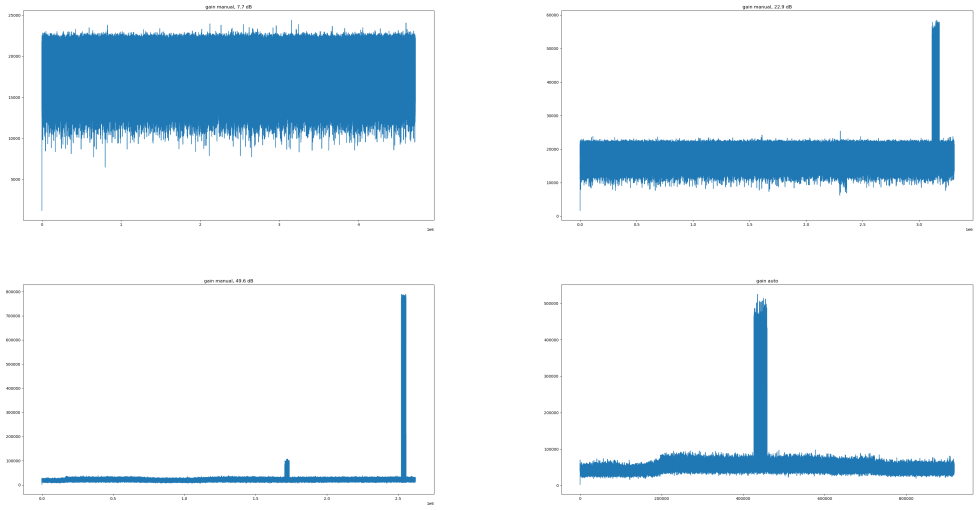
Přestože fungovalo vysoké nastavení gainu dobře pro některé případy 5.4, rozhodl jsem se ponechat automatiku, abych se vyhl možnému přenastavování gainu, pokud bude třeba. V tomto případě se automatika ukázala jako rozumné nastavení.



Obrázek 5.2 Grafy ukazující amplitudy a frekvence se zapnutým AGC (levo) a vypnutým (pravo), je vidět, že vypnuté AGC poskytuje daleko čistší a předvídatelná signál signál.



Obrázek 5.3 Grafy ukazující amplitudy s rozdílnými nastaveními šířky pásma



Obrázek 5.4 Grafy ukazující amplitudy s rozdílnými nastaveními citlivosti

5.2.2.5 Frequency correction

Nastavení korekce frekvence pro ”rozjíždějící se”oscilátor přijímače oproti oscilátoru vysílače. V datasheetu RTL-SDR V3 se uvádí, že tato hodnota by měla být mezi 0-2. Zkoušení ukázalo, že korekce nemá u tohoto zařízení téměř žádný smysl, neboť odchylka je velmi malá, nicméně hodnota 1 vykazovala v některých krajních případech o malinko lepší výsledky, proto toto nastavení je užito s hodnotou 1 pro toto zařízení.

5.2.2.6 Tuner IF gain

Experimentování s tímto nastavením nepřineslo žádné pozitivní výsledky oproti nepoužití tohoto nastavení,takže tato funkce není užita.

5.2.2.7 Shrnutí

Většina nastavení vyžaduje nějakou míru experimentování s nimi, aby se dalo přiblížit potřebné kvalitě přijímaného signálu, nicméně pro mnoho nastavení fungují automatické volby, které jsem také často volil. Největší vliv má nastavení tuner gainu a tunner bandwidth, nicméně pro mé účely stačí jeho automatická verze. AGC je nejlepší vypnuté a frequency correction je nastaveno na doporučení z diskusí o SDR, nicméně vliv je nepatrný.

5.3 Analýza rádiových signálů ze senzorů

Důležité pro testování aplikace a zároveň přidávání nových zařízení, které aplikace umí odposlechnout a demodulovat je schopnost analýzy signálu, který aplikace přijímá. Aplikace se zaměřuje hlavně na modulace amplitudové a frekvenční. Avšak analýza jako taková neznámého zařízení musí být provedena ručně.

5.3.1 Fyzická příprava

Prvním krokem k analýze konkrétního zařízení vysílajícího v pásmu 868 Mhz nebo 433 Mhz je co nejlepší izolace přijímače (v tomto konkrétním případě dipólové antény) a vysílače 5.5.

Obrázek 5.5 Zařízení připravené na odposlech a analýzu bezdrátového teplotního senzoru v izolovaném prostředí.

Snažíme se docílit toho, aby přijímač byl v blízkosti vysílače a nebyl rušen žádnými jinými zařízeními. V ISM pásmech se tohoto dosahuje obtížně, zvláště v případě hustě zabydlené městské oblasti, nicméně existuje několik možností, jak tohoto dosáhnout. Cokoliv podobné Faradayovi kleci funguje. Ideálním prostředím je třeba i kufr auta, plechová krabice a nebo lednice. Anténu i senzor, který chceme analyzovat umístíme do takovéhoto prostoru společně a vyvedeme kabel do zařízení. To zapojíme a spustíme aplikaci s požadovaným ukládáním dat.

5.3.2 SW příprava

Vzhledem k tomu, že neznáme jaká modulace je užívána, musíme uložit frekvenční i amplitudová data. která dále analyzujeme. Co však je známo téměř u každého senzoru, je pásmo, ve kterém se data přenášejí. Protože zařízení může být naladěno jen na jedno pásmo, musíme toto vybrat, nebo měření opakovat. Nicméně frekvenci zařízení lze často vyvodit. Pokud je zařízení krátkého dosahu a určené pro evropský trh, pravděpodobně to bude 433Mhz pásmo.

Samotná rychlost sbíraných dat je 250000 samplů za vteřinu, tato hodnota je v rámci aplikace neměnná a stačí pro drtivou většinu senzorů.

5.3.3 Měření

Po nastavení SW a HW samotné měření je zcela automatický proces. Nicméně zde je možno narazit na nějaká úskalí. Prvním je ohromné množství dat, která jsou naměřená a zapsána do souborů. Samotný IQ sample je 16 bitů *250 000 takovýchto vzroků za sekundu, frekvence i amplituda jsou 32 bitové floaty, oba s frekvencí 250 000 floatů za sekundu. Což ukazuje velký problém, a to náročnost dat jako takových. Například běžný venkovní teplotní senzor vysílá v

rozmezí mezi 1 - 10 minut, své údaje. Takže 10 minut měření a ukládání floatů je velmi paměťově náročné, v závislotsi na tom, jak často senzor vysílá své údaje.

Pro takovéto situace je lepší ukládat surová IQ data, a frekvenci a amplitudu zkoumat poté z takto uložených dat.

5.3.4 Vyhodnocení dat

Nejsnazším způsobem jak vyhodnotit data ze senzoru, která byla získána, je si tato data vizualizovat. Nejsnazší způsob je asi knihovna matplotlib v pythonu a načtení dat ze souboru pomocí numpy.

Prohlédnutím grafu jsme schopni okamžitě zjistit body zájmu. Pokud se jedná o graf amplitudy, body zájmu jsou zde velké výkyvy oproti normálnímu šumu. tyto výkyvy budou směřovat vzhůru a budou mít pravidelný tvar.

Pokud se jedná o frekvenční modualci je takto nutno zobrazit frekvenční graf a zde výchylky budou na obě strany.

Identifikovaná místa lze buď interaktivně zvětšit a nebo lze na základě grafu data oříznout. Jelikož raw IQ samply jsou dvojice 8 bitových čísel, takže lze použít třeba klasický příkaz DD pro jejich oříznutí. To samé lze udělat s amplitudou nebo frekvencí , kde jsou čísla 32 bitové floaty.

Následně lze podle vzhledu amplitudy identifikovat použitou druhou modulaci.

Stejně tak jde pokračovat v případě frekvence

5.4 Levné senzory

V rámci práce jsem se zabýval hlavně senzory, které lze volně a levně pořídit do domácnosti, ať se již jedná o senzory pohybu, teplot, vlhkosti a dalších. V tomto výběru nejsou zahrnuty žádné senzory, které mají profesionální uplatnění.

5.4.1 Teplotní senzory

Základní malý typ senzoru, k dostání poměrně všude, dvě prodávané verze, jen senzor teploty, nebo senzor teploty a vlhkosti vzduchu. Napájení je klasickými AA 1,5 voltu monočlánky. Senzor je pro venkovní i vnitřní užijí k běžně dodávané meteostanici, celkově se tento senzor řadí mezi levnější.

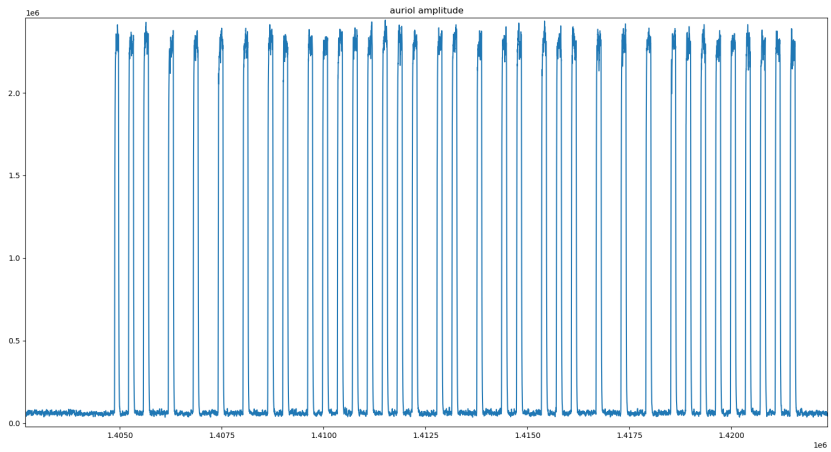
K testování a měření byly použity dva senzory jeden značky AURIOL a jeden značky GOGEN.

5.4.1.1 AURIOL

Tento konrétní senzor byl v rámci práce zkoumán jako první a byl zkoumán nejdéle ze všech ostatních senzorů, jelikož k němu byl nejsnazší přístup. Díky jednoduchosti daného senzoru a dat která odesílá mohl být tento senzor analyzován podrobně a posílaná data mohla být tak zmapována přesně. Jak vypadá odeslání jednoho balíčku dat lze vidět na obrázku 5.6.

5.4.1.2 Přenos dat

Senzor využívá pásmo 433 Mhz pro přenos dat do své centrály. Samotná modulace není uvedena v přiložených návodech, ani není uvedeno jak často tento senzor vysílá, ale experimentálně bylo ověřeno že se jedná o ASK (Amplitude shift keying) modulaci, a dále, že senzor vysílá každou minutu (tento čas není přesný má odchylky pár sekund, zda to je záměrně, aby nedocházelo k vzájemnému rušení pokud je senzorů více nebo náhoda nechci spekulovat) a vyšle celkově 12 pulsů. Senzor také vyšle data vždy okamžitě po vložení baterií. Využívaná modulace je, jak již



Obrázek 5.6 Odeslání balíčku dat ze senzoru auriol [66]

bylo řečeno, ASK a konkrétně typ On-Off keying (OOK). Dalším experimentem bylo určeno, že nad tímto modulovaným kanálem je ještě využita modulace typu Pulse Position modulation (PPM). Senzor vyšle celkově informaci o 36 bitech.

5.4.1.3 Ochrana přenášených dat

V rámci přenosu dat nebyla detekována žádná ochrana integrity dat, nebo jejich důvěrnost, v rámci přenosu dat je jen přenesena informace o číslu senzoru, které se mění po každém vložení baterií. A podle tohoto je pak identifikováno zařízení v rámci meteostanice.

5.4.1.4 Interpretace dat

Interpretace dat je obtížná, vzhledem k tomu že není zdokumentován žádný protokol, ohledně těchto dat. Zbývá jen experimentální cesta. Tato cesta spočívala v identifikaci měnících se bitů v závislosti na teplotě prostředí (spolu s kontrolním měřením), do kterého byl teploměr vložen a při výměně baterie.

Interpretace dat byla částečně udělána , první 2 nibbly (nibble = 4 bity) dávají dohromady náhodné číslo, které se mění po každém vložení baterie. Posledních 2 nibbly jsou vždy 0., 6. nibble je vždy 0xF, 2. nibble se zdá neměnný. 3.,4. a 5. nibble určuje teplotu. Při teplotě 26.4 stupňů na teploměru dostávám hodnotu 264. Lze předpokládat, že teplota je přenášená v desetinách stupňů. 12 bitové číslo se znaménkem podělené 10 udává teplotu. Což tereticky umožňuje rozsah hodnot od -204,8°do 204,7°Celsia. Teploměr má také indikátor výměny, a toto je potvrzeno vložením slabších baterií.

5.4.1.5 Shrnutí významu nibblů v rámci jedné zprávy

Odchycené nibbly v rámci jednoho přenosu: I I X T T T F 0 0 význam nibblů viz tabulka 5.1.

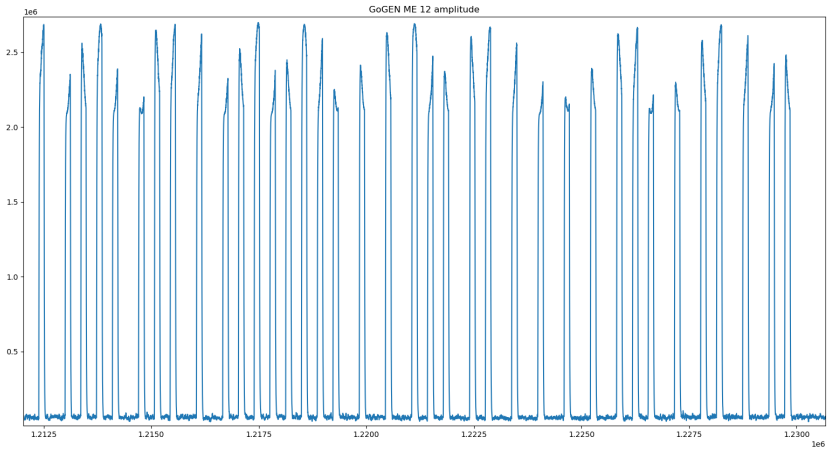
5.4.1.6 GoGEN ME SENZOR 12

Senzor, který se svou kompatibilitou se senzory značky Auriol velmi ulehčil své zkoumání. Díky kompatibilitě musí mít senzor podobné modulace přenosu dat, podobnou strukturu přenášených

Tabulka 5.1 Význam nibblů k senzoru značky Auriol

I	Nibbly označené I dávají dohromady náhodné 8 bitové číslo které se mění po restartování senzorů, pravděpodobně identifikátor
T	Nibbly označení jako T udávají hodnotu znaménkové teploty na 1 desetinné místo vynásobené 10
X	Nibble označený jako X je neznámý
0	Nibble označený jako 0 jsou vždy 0

dat a stejnou interpretaci. Graf přenosu je zachycen zde 5.7.



Obrázek 5.7 Odeslání balíčku dat ze senzoru GoGEN ME 12
[66]

5.4.1.7 Přenos dat

Data jsou přenášena v pásmu 433 Mhz, samotná modulace není nikde uvedena, ale vzhledem ke kompatibilitě s centrálou značky Auriol, lze předpokládat stejnou modulaci. Modulace je typu ASK , konkrétně OOK, kombinovaná s PPM modulací. Tento senzor ale má navíc senzor vlhkosti, který Auriol nemá, Proto došlo k pokusu, kolik bitů je přeneseno. Je přeneseno přesně 36 bitů.

5.4.1.8 Ochrana přenášených dat

V rámci přenosu dat nebyla detekována žádná ochrana přenášení dat, v rámci přenosu dat je jen přenesena informace o číslu senzoru, které se mění po každém vložení baterií.

5.4.1.9 Interpretace dat

Narozdíl od senzoru značky Auriol je však tento senzor vybaven i snímáním vlhkosti, která musí být taky přenášena. Podle experimentu jsou oproti senzoru značky Auriol používány poslední 2 nibbly , které u senzoru auriol jsou 0. Vlhkost se udává v procentech a není nijak škálovaná. Vzhledem k tomu, že tento senzor lze párovat se stanicí značky Auriol a ukazuje správné hodnoty si myslím, že jediná změna oproti Auriolu jsou poslední dva nibbly.

5.4.1.10 Shrnutí významu nibblů

Odchycené nibbly v rámci jednoho přenosu: I I X T T T F H H význam nibblů viz tabulka 5.2.

Tabulka 5.2 Význam nibblů k senzoru značky Gogen

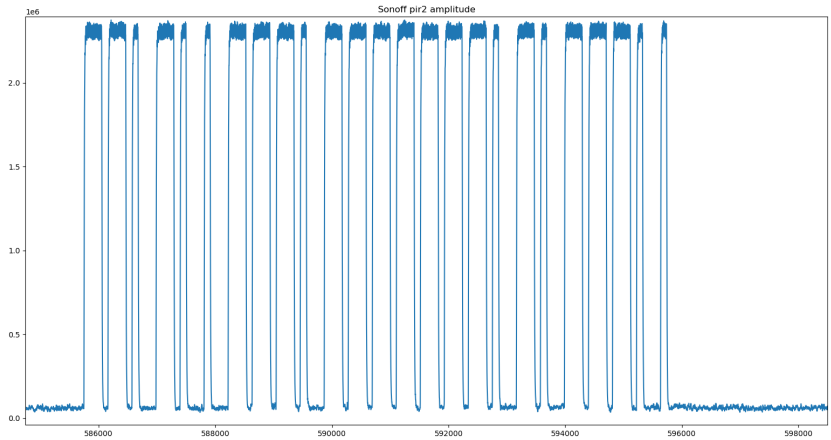
I	Nibbly označené I dávají dohromady náhodné číslo, které se mění po vložení baterií, pravděpodobně identifikátor
T	Nibbly označení jako T dávají hodnotu znaménkové teploty na 1 desetiné místo vynásobené 10
X	Nibble označený jako X je neznámý
H	Nibble označený jako H dávají dohromady vlhkost vzduchu a jsou místo nibblů 0, které jsou u senzoru Auriol

5.4.2 Senzory pohybu PIR

Dnes asi nejrozšířenější druh senzorů, běžně užíván k zabezpečení domovů i podniků. Pasivní infračervený detektor pohybu. Tento levný detektor běžně instalovaný v každé místnosti se nejčastěji užívá jako součást poplašného systému, ale třeba místy také jako součást automatických světel, otevírání dveří a podobně. Nejčastěji se s ním ale setkáme v rámci nějakého systému alarmu.

5.4.2.1 CT60M sonoff PIR2

Levný dveřní PIR senzor, který byl odposlechnut v rámci testování. Senzor detekuje pohyb vždy v určitých intervalech. Odposlech byl zaměřen na přenášená data a bezpečnost. Přenášená data nebylo možno interpretovat z důvodu jejich neměnnosti. Graf zachycující průběh vysílání je zde 5.8.



Obrázek 5.8 Odeslání balíčku dat ze senzoru PIR2
[66]

5.4.2.2 Přenos dat

Senzor používá pásmo 433 Mhz k přenosu dat. Opět nejsou žádné informace k tomu, jak jsou data přenášená, takže vše bylo ověřováno experimentálně. Jedná se o přenos pomocí ASK modulace, jako nosného kanálu, kdy je využito OOK modulování spolu s Pulse Width modulation (PWM), kde jsou data přenášena pomocí šíře pulzu. Data jsou opakovaně přenesena 24x za sebou, aby se zajistilo spolehlivé přenesení. Přeneseno je vždy 25 Bitů.

5.4.2.3 Ochrana přenášených dat

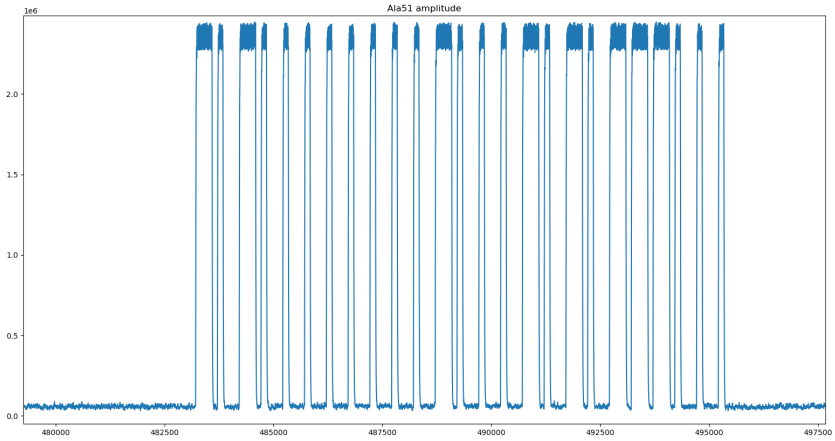
Integrita samotných dat není nijak hlídána, opakování dat je pravděpodobně z důvodu zvýšení šance na přenos dat bez problémů. Bezpečnost dat také není nijak zabezpečena.

5.4.2.4 Interpretace dat

K interpretaci dat není dostatek informací, poslaná data se nemění ani při výměně baterie, není tedy možnost jak data interpretovat. Ani experimentálně.

5.4.2.5 ALA51

Kvalitnější PIR senzor, který při vysílání spouští pokaždé bezdrátový zvonek v místě zkoušení. Lze tedy očekávat, že tento produkt vysílá nějaká neměnná data, bez ochrany. Navíc po 3 detekcích se senzor zamkne a další 3 minuty nedetekuje nic. Přenos dat je zaznamenán v grafu 5.9.



Obrázek 5.9 Odeslání balíčku dat ze senzoru pir ala51
[66]

5.4.2.6 Přenos dat

Senzor používá pásmo 433 Mhz, lze přenastavit na 868 Mhz dle návodu, nevyzkoušeno, návod neříká jak tohoto dosáhnout, pouze zmiňuje tuto možnost. Přenos dat je opět vytvářen jednoduchou ASK modulací typu OOK (on-off keying). Toto je kombinované s PWM pro přenos dat. Přeneseno je opět vždy 25 bitů.

5.4.2.7 Ochrana přenášených dat

Integrita samotných dat není nijak hlídána, opakování dat je pravděpodobně z důvodu zvýšení šance na přenos dat bez problémů, v tomto případě ale pouze 4x a většinou jen 2 přenosy jsou celé, může být vadným či starým kusem.

5.4.2.8 Interpretace dat

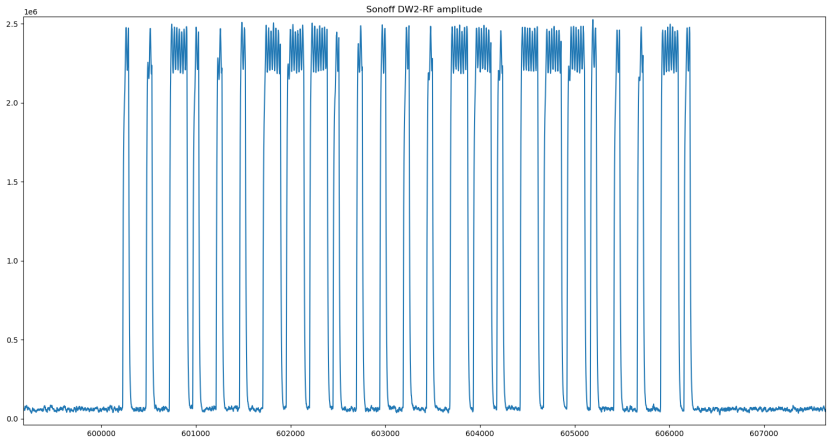
K interpretaci dat není dostatek informací, poslaná data se nemění ani při výměně baterie, není tedy možnost jak data interpretovat. Ani experimentálně, datasheet k tomuto kusu nebyl nalezen.

5.4.3 Senzory pohybu otevření dveří/oken

Další senzor s poměrně širokým rozšířením využitím hlavně pro alarmy a poplašné systémy. Toto čidlo umí vyslat signál v případě otevření zabezpečených dveří, oken či dalších míst. Běžné využití je v domácnostech, ale i obchodech, kde takovéto senzory občas nahrazují zvonky na dveřích. Tento senzor je vetšinou založen na detekci blízkého magnetu.

5.4.3.1 Sonoff DW2-RF

Magnetický senzor na okna a dveře, kde se vyšle signál, když je senzor vzdálen od magnetu. V rámci testování byl zkoumán přenos dat, ochrana přenášených dat a jejich interpretace. Graf zachycující přenos jednoho balíčku dat 5.10.



Obrázek 5.10 Odeslání balíčku dat ze senzoru otevřených dveří Sonoff DW2-RF [66]

5.4.3.2 Přenos dat

Zařízení funguje jen na pásmu 433 Mhz. Opět nejsou uvedeny žádné konkrétní informace o typu a způsobu přenosu dat. Experimentálně bylo zjištěno, že se jedná o přenos pomocí ASK modulace, typu OOK, s tím že je to typ PWM. Přenáší se 25 bitů. opakovaně, nicméně kvalita je kolísavá, a ne vždy se přenese všech 25 bitů. A počet opakování je také nestálý

5.4.3.3 Ochrana přenášených dat

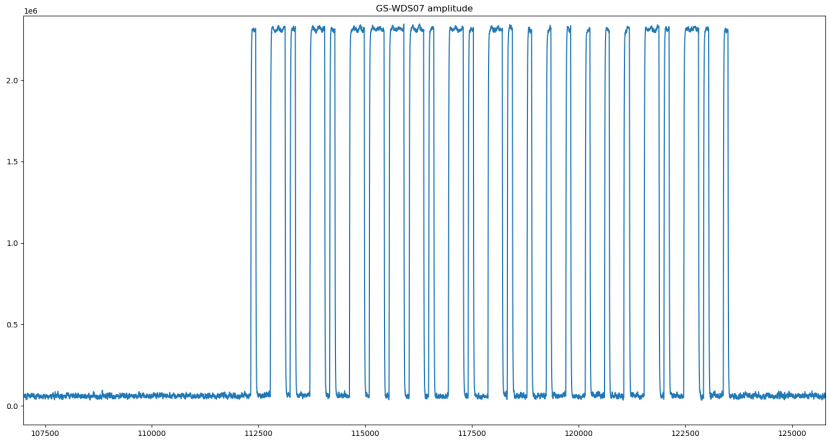
Jako u předchozích zařízení i zde je jediný způsob ochrany dat jejich přenášení několikrát za sebou, pro případ, kdyby jeden přenos nevyšel. U tohoto zařízení nevychází velká část přenosů, někdy je přenos ukončen v půlce, jindy je amplitudová modulace velmi malá a k nerozeznání od šumu.

5.4.3.4 Interpretace dat

Vyslaná data se nijak nemění. vysílá se jedna sekvence dat při každém spuštění, kdy je oddálen magnet. Vysílaná data jsou podobná jako vysílá PIR senzor.

5.4.3.5 GS-WDS07

Jednoduchý magnetický senzor, který vysílá krom oddálení magnetu také jeho přiblížení, jinak se neliší od předchozího senzoru. Testovaný byl přenos dat a dále ochrana dat a pokus o analýzu těchto dat. Graf jednoho přenosu dat 5.11.



Obrázek 5.11 Odeslání balíčku dat ze senzoru otevřených dveří GS-WDS07 [66]

5.4.3.6 Přenos dat

Jediná dodaná informace je že zařízení funguje na 433 Mhz. Nic konkrétního uvedeno není. Opět experimentálně bylo ověřeno ASK modulace, konrétně OOK verze a PWM přenos dat nad tím nastavený. Přeneseno je 25 bitů. Kvalita přenosu je velmi rozkolísaná, ne vždy je přeneseno 25 bitů.

5.4.3.7 Ochrana přenášených dat

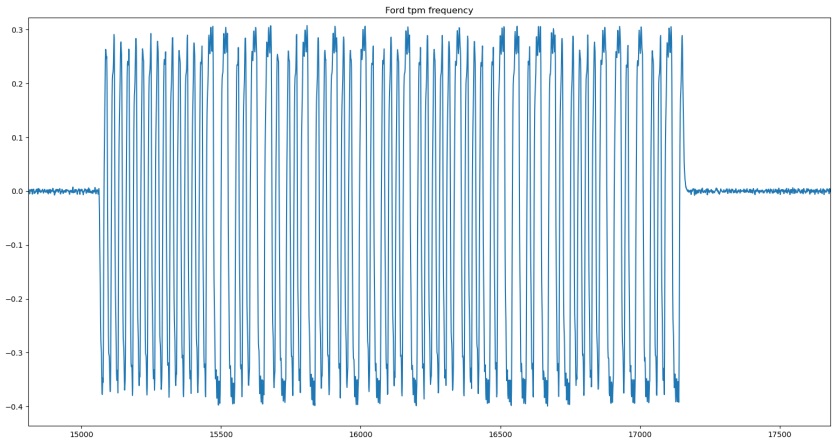
Jediná ochrana přenášených dat je opakování, pokud dojde k záměně bitů, tak přijímač může použít majoritu pro získání správných dat. Jinak není nijak zajištěna ochrana dat.

5.4.3.8 Interpretace dat

Interpretace dat stejně jako v předchozích případech je obtížná, jelikož nemám s čím porovnávat, nicméně zde se mění 6. nibble jeden bit, pro indikaci zda je senzor otevřen či uzavřen, zbytek dat zůstává neměnný.

5.4.4 Senzory tlaku v pneumatikách značky FORD

Klasické TPMS senzory značky Ford vyráběné firmou Continental. Tyto senzory jsou jediný FSK modulovaný senzor, ke kterému jsem se fyzicky dostal a demoduloval. Graf naměřeného přenosu je zde 5.12.



Obrázek 5.12 Odeslání balíčku dat ze senzoru TPMS značky Ford [66]

5.4.4.1 Přenos dat

Podle manuálu je přenos dat na frekvenci 433 Mhz pro Evropu a může využívat buď amplitudovou modulaci, nebo frekvenční modulaci. Konkrétně měřený typ využíval frekvenční modulaci, která s největší pravděpodobností užívá manchaster coding pravděpodobně bi-phase enccoding, jelikož samotná modulace se zdá být FSK. A odpovídá tomu, jak vypadá manchestrovo kódování bi phase.

5.4.4.2 Ochrana přenášených dat

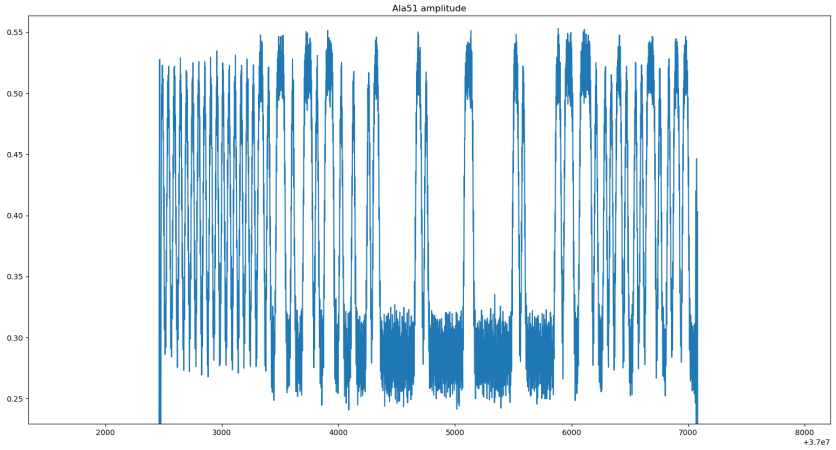
Přenášená data jsou v rámci přenosu chráněna proti chybě kanálu jednoduchým CRC, který slouží jako ochrana integrity dat. Jakákoliv ochrana dat jako je šifrování, nebo ověření původu naměřených dat, však dle všeho není implementována.

5.4.4.3 Interpretace dat

Z důvodů nedostatku času a údajů nebyla interpretace dat zkoumána. Prvních několik bitů se zdá být preambuli, ta je neměnná, zbytek bitů a jejich účel není znám. Teoreticky je zachyceno 160 bitů, první 4 bajty se zdají být preambule, zbývá 128 bitů dat. Vzhledem k teoretickému Manchesteru kódování je dvojice dat 1 bit. Celkem tedy je přeneseno 64 bitů informace.

5.4.5 E-ITN 30

Systém pro dálkový odečet tepla měřený uprostřed velkého a rušného paneláku, zachycení senzorů bylo problematické a výsledky nejsou jasné, jelikož není snadné určit, která část naměřených dat odpovídá tomuto senzoru viz graf 5.13.



Obrázek 5.13 Pravděpodobné zachycení dálkového senzoru topných nákladů

5.4.5.1 Přenos dat

Přenos dat je na frekvenci 868 MHz, toto zařízení je jediné, které toto pásmo využívá. Systém data vysílá každé 4 minuty + pseudonáhodné rozmezí, aby se zabránilo překryvu, a tím znemožnění odečtu, pokud by dva systémy vysílaly ve stejný čas. Přesný přenos dat se mi bez rušení nepodařilo zachytit. Odhaduji že senzor využívá FSK modulaci spolu s nějakým kódováním, ale nejsem schopen určit druh kódování.

5.4.5.2 Ochrana přenášených dat

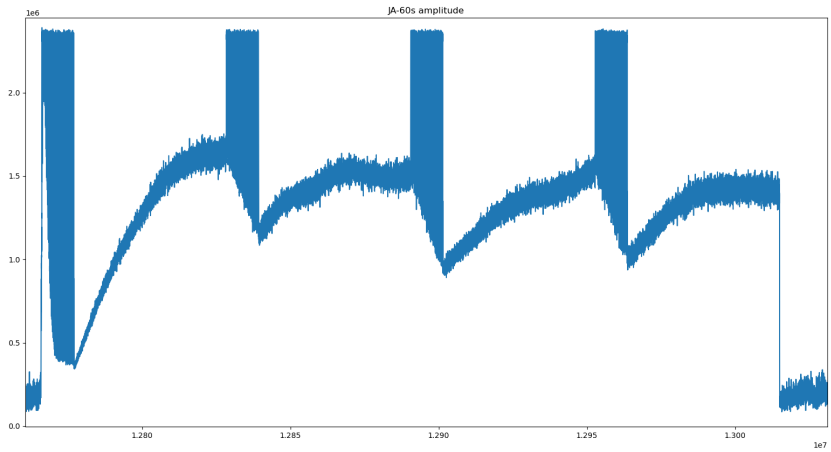
Návod k tomuto zařízení zmiňuje šifrování dat, takže lze předpokládat tuto ochranu dat. Jaký algoritmus, v případě jeho nastavení či jakým způsobem jsou klíče nastaveny a uloženy už není zmíněno. Nicméně zařízení jsou malá a komunikují jen jednosměrně podle manuálu, takže předpoklad je uložený klíč v zařízení.

5.4.5.3 Interpretace dat

Nemohla proběhnout vzhledem k silnému okolnímu rušení a nepřístupnosti ke konkrétnímu odstíněnému senzoru a šifrové ochraně dat.

5.4.6 JA-60S

Tento starý detektor kouře byl darován pro testování mé aplikace. Vzhledem ke stáří detektoru, které je větší než maximální životnost zařízení podle návodu, neproběhl pokus o interpretaci nebo zjištění ochrany dat. Ačkoliv zařízení kouř detekuje, samotné vysílání již nefunguje. Na grafu 5.14 lze vidět, že zařízení již nevysílá tak jak by mělo.



Obrázek 5.14 Nekvalitní odesílání dat ze senzoru kouře JA-60S [66]

5.4.6.1 Přenos dat

Přenos dat je v pásmu 433 Mhz, a celkově se jeví jako ASK modulace spolu s kódováním Manchaster. Signál jako takový je rozkolísaný a nedokáži ho nijak analyzovat. Zdá se že vysílání neprobíhá v pořádku, toto přisuzuji věku zařízení, který již překročil životnost zařízení danou výrobcem a dlouhodobému neužívání tohoto zařízení.

Bez funkčního vysílání ze zařízení nelze analyzovat ochranu ani interpretovat přenášená data.

5.4.7 EZ-7901

Dálkově ovládané zásuvky značky Ecolite, tyto konkrétní zásuvky jsou ještě funkční, ale poměrně staré, což lze vidět například z grafu 5.15 vysílání ovladače, kde jsou na konci jednotlivých pulzů rozeznatelné špičky, které by zde být neměli.

5.4.7.1 Přenos dat

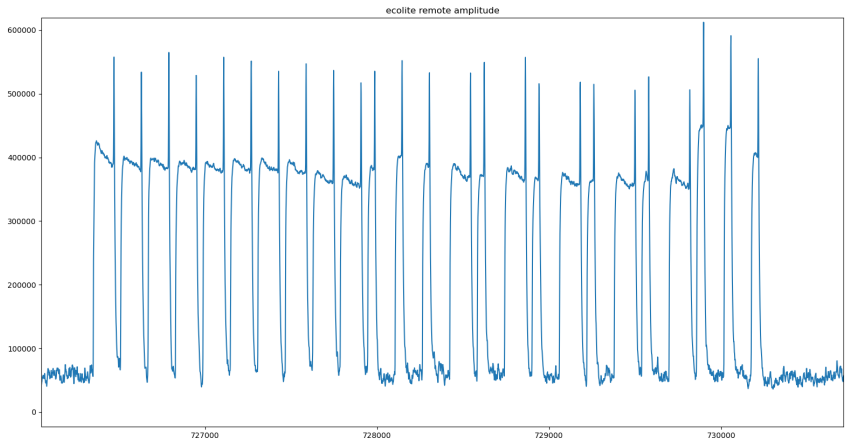
Ovladač funguje v pásmu 433 Mhz, a jedná se o modulaci ASK typu OOK, což je viedět z grafu 5.15, d8le y grafu lze vyčíst že se jedná o modulaci typu PWM. Přeneseno je 25 bitů v rámci jednoho příkazu zásuvce.

5.4.7.2 Ochrana přenášených dat

Přenášená data se nezdají být nijak chráněná proti zneužití, zapínání zapojených zásuvek je vždy provedeno stejným množstvím a stejnou posloupností přenesených bitů, stejně tak vypnutí. Lze se domnívat, že není implementována žádná ochrana přenášených dat.

5.4.7.3 Interpretace dat

První dva nibbly dat jsou vždy 0, následují 3 nibbly dat, které se mění v závislosti na použité zásuvce, lze se domnívat, že určují konkrétní zásuvku, další nibble se mění pouze v závislosti na tom, jestli zásuvku chci vypnout či zapnout. Poslední přenesený bit je vždy 1.



Obrázek 5.15 Odesílání dat z dálkového ovládání zásuvek, poslední výkyv při konci každého pulzu je známka poškozeného odesílání

[66]

5.4.7.4 Shrnutí významu nibblů

Odchycené nibbly v rámci jednoho přenosu: 0 0 I I I Z 8 význam nibblů viz tabulka 5.3.

Tabulka 5.3 Význam nibblů k dálkovému ovládání zásuvek značky Ecolite

I	Nibbly označené I dávají dohromady číslo pravděpodobně identifikátor zásuvky
0	Nibbly označení jako 0 jsou vždy 0
Z	Nibble který se mění v závislosti na tom zda zásuvka se má zapnout či vypnout, konkértně se mění mezi hodnotou B a E
8	Nibble tvořený jedním bitem, vždy dá dohromady 8

Kapitola 6

Závěr

Analýza technologie SDR ukázala její velký potenciál pro oblasti komeční, průmyslové ale i domácí sféry. Díky levným ADC převodníkům může SDR mít budoucnost i jako centrální řešení pro agregaci dat od nekompatibilních výrobců různých systémů. Již existují náznaky takovýchto programů.

Samozřejmě díky své všestrannosti je SDR velmi vhodné i na odchytávaní existujících senzorů a to hlavně v ISM pásmech, nicméně není problém SDR přenastavit i na pásma licenční.

Pro senzorové sítě a přenos malého množství dat jsou v rámci ČR vyhrazena dvě ISM pásma, jedno pásmo je 433 Mhz a druhé 868 Mhz, tato pásma mají téměř totožné využití, ačkoliv pásmo 868 Mhz je regulováno v každém státě Evropy jinak a z toho důvodu většina senzorů funguje v pásmu 433 Mhz.

Vytvořený nástroj se ukázal jako funkční a levné řešení postavené na SDR přijímači za pár stovek korun. Nástroj je psaný v C/C++ a přenositelný mezi platformou Linux a Windows (otestováno). Umožňuje zachycení provozu v ISM pásmech a uložení jako IQ vzorků, případně jako amplitudy a frekvence.

Dále nástroj implementuje možnost přidání nových senzorů, či rozšíření stávajících, tak aby bylo možné detekovat a číst data, která senzor posílá nehledě na to, zda jsou šifrovaná či ne. Senzor stačí analyzovat a přidat parametry, případně rozšířit detekční funkci, pokud daná modulace není zpracovávána. Pokud daná modulace už zpracovávána je, pouze by mělo jít přidat informace o konkrétním senzoru. V rámci knihovny jsou implementovány dvě hlavní modulace ASK a FSK a několik dalších spojených modulací.

Poslední vlastností této aplikace je zabudování openSSL knihovny, která nabízí možnost dešifrování a případně rozšíření dešifrování o další šifry a módy, pokud je třeba.

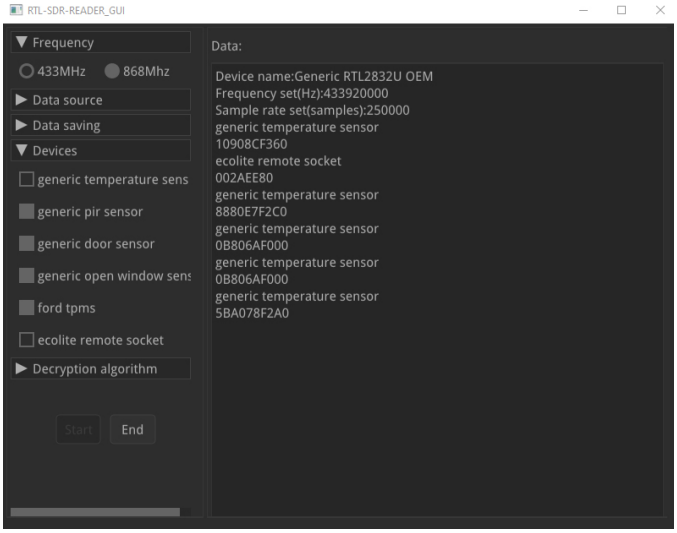
Celá aplikace byla řádně otestována, jelikož krom testování sloužila také jako analyzátor senzorů, které jsem odchytával. Aplikace funguje tak, jak bylo předpokládáno.

V rámci tvorby aplikace a analýzy senzorů jsem dospěl k závěru, že současná generace domácích senzorů, které fungují na ISM pásmech, je ve většině případů nezabezpečená, a že není moc snaha o to, to měnit, a ani zákazník nejspíše nemá zájem o takováto řešení.

Na mnoha stránkách se senzory, ať již například k ovládání vytápění, nebo alarmům je zabezpečení samotného přenosu dat zmíněno až jako poslední věc, pokud vůbec, jediný testovaný senzor, který jsem mohl naměřit, měl oficiálně šifrování, ostatní spíše spoléhali na security by obscurity, že se neví jaká modulace, jak často a nebo jak interpretovat data.

Příloha A

Příloha - ukázka hotové aplikace



Obrázek A.1 Rozhraní běžící aplikace

Bibliografie

1. CHAN, Marie; CAMPO, Eric; ESTÈVE, Daniel; FOURNIOLS, Jean-Yves. Smart homes — Current features and future perspectives. *Maturitas*. 2009, roč. 64, č. 2, s. 90–97. ISSN 0378-5122. Dostupné z DOI: https://doi.org/10.1016/j.maturitas.2009.07.014.
2. ALAM, Muhammad Raisul; ALI, Mohd Alauddin Mohd; REAZ, Mamun Bin Ibne. Wireless Sensor Networks. *Computer*. 2012, roč. 41, č. 10, s. 92–95. Dostupné z DOI: 10.1109/mc.2008.441.
3. CASTELLS-RUFAS, David; GALIN-PONS, Adrià; CARRABINA, Jordi. The regulation of unlicensed sub-GHz bands: Are stronger restrictions required for LPWAN-based IoT success? *arXiv preprint arXiv:1812.00031*. 2018. Dostupné z DOI: doi.org/10.48550/arXiv.1812.00031.
4. BLACKMAN, James. *Disposable dollar-priced IOT – how massive will it really get?* [online]. RCR Wireless News, 2022-01-25 [cit. 2023-04-25]. Dostupné z: https://www.rcrwireless.com/20220125/fundamentals/disposable-dollar-priced-iot-how-massive-will-it-really-get.
5. REN, Yi; OLESHCHUK, Vladimir A; LI, Frank Yong; GE, Xiaohu. Security in mobile wireless sensor networks-A survey. 2011.
6. UNIFORE. *433MHz or 868MHz wireless alarm system, what's the difference?* [online]. Shenzhen Meidasi Technology Development Co, 2015-08-13 [cit. 2023-04-25]. Dostupné z: https://www.burglaryalarmsystem.com/technology-news/433mhz-or-868mhz-wireless-alarm-system-what-s-the-difference.html.
7. *"DVB-T COFDM DEMODULATOR+USB 2.0"*. "Realtek Semiconductor Corp.", 2010. "Rev.1.4".
8. MERBANAN. *Merbanan/RTL_433: Program to decode radio transmissions from devices on the ISM bands (and other frequencies)* [online]. merbanan, 2021-04-15 [cit. 2023-04-14]. Dostupné z: https://github.com/merbanan/rtl_433.
9. *SDR# and Airspy downloads* [online]. airspy.com, 2023-02-27 [cit. 2023-04-25]. Dostupné z: https://airspy.com/download/.
10. RYZERTH [online]. SDR++, 2023-04-20 [cit. 2023-04-20]. Dostupné z: https://www.sdrpp.org/.
11. *The Free & Open Source Radio Ecosystem · Gnu Radio* [online]. GNU Radio, 2023-04-20 [cit. 2023-04-20]. Dostupné z: https://www.gnuradio.org/.
12. LAFORGE. *SDR (Software Defined Radio) "RTL-SDR* [online]. Osmocom, 2022-07-25 [cit. 2023-04-20]. Dostupné z: https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr.

13. MATUSCHEK, Hannes. *HMATUSCHEK/LIBSDR: A simple software defined radio (SDR) library* [online]. 2015-06-14. [cit. 2023-04-20]. Dostupné z: https://github.com/hmatuschek/libsdr.

14. PORT, Roman. *Roman-port/LIBSDR: Libsdr is easy to use, low latency, and Modular Software Defined Radio Library, fully written in C# .NET code with no external dependencies or processes.* [online]. 2021-06-09. [cit. 2023-04-20]. Dostupné z: https://github.com/Roman-Port/LibSDR.

15. D, Roger. *PYRTLSDR/pyrtlsdr: A python wrapper for librtlsdr (a driver for Realtek RTL2832U based SDR's)* [online]. 2022-08-01. [cit. 2023-04-20]. Dostupné z: https://github.com/pyrtlsdr/pyrtlsdr.

16. *Cross-platform software design and Development Tools* [online]. Qt Group, 2023-04-15 [cit. 2023-04-15]. Dostupné z: https://www.qt.io/.

17. TEAM, wxWidgets. *Wxwidgets - Latest news* [online]. wxWidgets, 2023-02-13 [cit. 2023-04-15]. Dostupné z: https://www.wxwidgets.org/.

18. TEAM, The GTK. *The GTK project - a free and open-source cross-platform Widget Toolkit* [online]. The GTK Team, 2023-04-05 [cit. 2023-04-05]. Dostupné z: https://www.gtk.org/.

19. DUMBLOB. *Immediate-mode-ui/nuklear: A single-header ANSI C immediate mode cross-platform GUI library* [online]. Immediate Mode UIs, 2023-03-09 [cit. 2023-02-17]. Dostupné z: https://github.com/Immediate-Mode-UI/Nuklear.

20. SCITER. *Embeddable HTML/CSS/javascript engine for modern UI development* [online]. Terra Informatica Software, 2023-03-08 [cit. 2023-04-05]. Dostupné z: https://sciter.com/.

21. *The legion of the bouncy castle* [online]. Legion of the Bouncy Castle Inc., 2021-04-21 [cit. 2023-04-25]. Dostupné z: https://www.bouncycastle.org/.

22. OPENSSL FOUNDATION, Inc. *OpenSSL* [online]. OpenSSL Foundation, 2023-03-22 [cit. 2023-03-23]. Dostupné z: https://www.openssl.org/.

23. DAI, Wei. *Crypto++® Library 8.7* [online]. Crypto++, 2021-01-01 [cit. 2023-04-25]. Dostupné z: https://www.cryptopp.com/.

24. [online]. OpenBSD Foundation, 2023-04-08 [cit. 2023-04-25]. Dostupné z: https://www.libressl.org/.

25. KERNINGHAN, Brian W.; RITCHIE, Dennis M. *The C programming language*. Prentice Hall, 1998. ISBN 0131103628.

26. STROUSTRUP, Bjarne. An overview of C++. In: *Proceedings of the 1986 SIGPLAN workshop on Object-oriented programming*. 1986, s. 7–18.

27. NIEMEYER, Patrick; KNUDSEN, Jonathan. *Learning java*. "O'Reilly Media, Inc.", 2005.

28. PYTHON, Why. Python. *Python Releases Wind*. 2021, roč. 24.

29. ÚŘAD, Český telekomunikační. *Rádiové Spektrum* [online]. Český telekomunikační úřad, 2018-01-01 [cit. 2023-04-18]. Dostupné z: https://www.ctu.cz/radiove-spektrum.

30. PETR, Hubík; TOMÁŠ, Břinčil [online]. Český telekomunikační úřad, 2023-04-11 [cit. 2023-04-26]. Dostupné z: https://spektrum.ctu.cz/kmitocty?filter%5C%5BapplicationIds%5C%5D%5C%5B0%5C%5D=52.

31. TUSET-PEIRÓ, Pere; ANGLÈS-VAZQUEZ, Albert; LÓPEZ-VICARIO, José; VILAJOSANA-GUILLÉN, Xavier. On the suitability of the 433 MHz band for M2M low-power wireless communications: propagation aspects. *Transactions on Emerging Telecommunications Technologies*. 2014, roč. 25, č. 12, s. 1154–1168. Dostupné z DOI: https://doi.org/10.1002/ett.2672.

32. ÚŘAD, Český telekomunikační. *Plán Využití Rádiového Spektra* [online]. Český telekomunikační úřad, 2022-08-31 [cit. 2023-04-18]. Dostupné z: https://www.ctu.cz/plan-vyuziti-radioveho-spektra.
33. ÚŘAD, Český telekomunikační. *Všeobecná Oprávnění* [online]. Český telekomunikační úřad, 2023-04-15 [cit. 2023-04-18]. Dostupné z: https://www.ctu.cz/vseobecna-opravneni.
34. SADIKU, M.N.O.; AKUJUOBI, C.M. Software-defined radio: a brief overview. *IEEE Potentials*. 2004, roč. 23, č. 4, s. 14–15. Dostupné z DOI: 10.1109/MP.2004.1343223.
35. EWING, Martin. *The abcs of software defined radio: Why your next radio will be SDR*. First. ARRL, 2012. ISBN 978-0-87259-632-0.
36. CORP., NATIONAL INSTRUMENTS. *Software defined radio: Past, present, and future* [online]. NATIONAL INSTRUMENTS CORP., 2023-04-11 [cit. 2023-04-25]. Dostupné z: https://www.ni.com/en/perspectives/software-defined-radio-past-present-future.html.
37. KUISMA, Mikael Q. *I/Q Data for Dummies* [online]. Mikael Q Kuisma, 2023-03-19 [cit. 2023-04-25]. Dostupné z: http://whiteboard.ping.se/SDR/IQ.
38. MIDDLESTEAD, Richard W. FREQUENCY SHIFT KEYING (FSK) MODULATION, DEMODULATION, AND PERFORMANCE. In: *Digital Communications with Emphasis on Data Modems: Theory, Analysis, Design, Simulation, Testing, and Applications*. 2017, s. 207–225. Dostupné z DOI: 10.1002/9781119011866.ch5.
39. [online]. Tutorials Point, 2023-04-25 [cit. 2023-04-25]. Dostupné z: https://www.tutorialspoint.com/digital_communication/digital_communication_frequency_shift_keying.htm.
40. SHARMA, Ravindra H; BHATT, Kiritkumar R. A review on implementation of QAM on FPGA. *International Journal of Innovative Research in Computer and Communication Engineering*. 2015, roč. 3, č. 3, s. 1684–1688.
41. HOEHER, P.; LODGE, J. "Turbo DPSK": iterative differential PSK demodulation and channel decoding. *IEEE Transactions on Communications*. 1999, roč. 47, č. 6, s. 837–843. Dostupné z DOI: 10.1109/26.771340.
42. [online]. Tutorials Point, 2023-04-25 [cit. 2023-04-25]. Dostupné z: https://www.tutorialspoint.com/digital_communication/digital_communication_phase_shift_keying.htm.
43. HANNAN, Mahammad A.; ABBAS, Saad M.; SAMAD, Salina A.; HUSSAIN, Aini. Modulation Techniques for Biomedical Implanted Devices and Their Challenges. *Sensors*. 2011, roč. 12, č. 1, s. 297–319. ISSN 1424-8220. Dostupné z DOI: 10.3390/s120100297.
44. [online]. Tutorials Point, 2023-04-18 [cit. 2023-04-18]. Dostupné z: https://www.tutorialspoint.com/digital_communication/digital_communication_amplitude_shift_keying.htm.
45. WAGGENER, Bill. *Pulse code modulation techniques: With applications in communication and Data Recording*. Van Nostrand Reinhold, 1995.
46. CONTRIBUTOR, TechTarget. *What is Pulse Amplitude Modulation (PAM)?: Definition from TechTarget* [online]. TechTarget, 2011-03-24 [cit. 2023-03-02]. Dostupné z: https://www.techtarget.com/whatis/definition/pulse-amplitude-modulation-PAM.
47. YU, Z.; MOHAMMED, A.; PANAHI, I. A review of three PWM techniques. In: *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*. 1997, sv. 1, 257–261 vol.1. Dostupné z DOI: 10.1109/ACC.1997.611797.
48. HAMKINS, Jon. Pulse Position Modulation. In: *Handbook of Computer Networks*. John Wiley & Sons, Ltd, 2007, s. 492–508. ISBN 9781118256053. Dostupné z DOI: https://doi.org/10.1002/9781118256053.ch32.
49. SHELDON, Robert. *What is Manchester encoding?: Definition from TechTarget* [online]. TechTarget, 2023-04-18 [cit. 2023-04-19]. Dostupné z: https://www.techtarget.com/searchnetworking/definition/Manchester-encoding.

50. MITKARI, Mayuresh. *Ciaan of security* [online]. Medium, 2021-09-30 [cit. 2023-03-19]. Dostupné z: https://medium.com/@mayureshmitkari/ciaan-of-security-c7a9c8571a61.

51. *Data confidentiality: Identifying and protecting assets* [online]. 2023-03-10. [cit. 2023-03-10]. Dostupné z: https://www.nccoe.nist.gov/data-confidentiality-identifying-and-protecting-assets-and-data-against-data-breaches.

52. BROOK, Chris; LORD, Nate. *What is Data Integrity? definition, types & tips* [online]. Fortra LLC, 2022-11-07 [cit. 2023-04-01]. Dostupné z: https://www.digitalguardian.com/blog/what-data-integrity-data-protection-101.

53. DELAWARE, University of [online]. University of Delaware, 2023-02-15 [cit. 2023-02-15]. Dostupné z: https://www1.udel.edu/security/data/availability.html.

54. UNIVERSITY, Boston. *Understanding Authentication, Authorization, and Encryption* [online]. Boston University, 2023-04-18 [cit. 2023-04-18]. Dostupné z: https://www.bu.edu/tech/about/security-resources/bestpractice/auth/.

55. STANDARDS, National Institute of; TECHNOLOGY. *DIGITAL SIGNATURE STANDARD (DSS)*. Washington, D.C., 2023-02. Tech. zpr., Federal Information Processing Standards Publication 186-5,Published: February 3, 2023. U.S. Department of Commerce. Dostupné z DOI: 10.6028/NIST.FIPS.186-5.

56. JESÚS RUGELES URIBE, José de; GUILLEN, Edward Paul; CARDOSO, Leonardo S. A technical review of wireless security for the internet of things: Software defined radio perspective. *Journal of King Saud University - Computer and Information Sciences*. 2022, roč. 34, č. 7, s. 4122–4134. ISSN 1319-1578. Dostupné z DOI: https://doi.org/10.1016/j.jksuci.2021.04.003.

57. AYYILDIZ, Cem; CETIN, Ramazan; KHODZHAEV, Zulfidin; KOCAK, Taskin; SOYAK, Ece Gelal; GUNGOR, V. Cagri; KURT, Gunes Karabulut. Physical layer authentication for extending battery life. *Ad Hoc Networks*. 2021, roč. 123, s. 102683. ISSN 1570-8705. Dostupné z DOI: https://doi.org/10.1016/j.adhoc.2021.102683.

58. PATHAN, A.S.K.; LEE, Hyung-Woo; HONG, Choong Seon. Security in wireless sensor networks: issues and challenges. In: *2006 8th International Conference Advanced Communication Technology*. 2006, sv. 2, 6 pp.–1048. Dostupné z DOI: 10.1109/ICACT.2006.206151.

59. PALUMBO, Filippo; ULLBERG, Jonas; ŠTIMEC, Ales; FURFARI, Francesco; KARLSSON, Lars; CORADESCHI, Silvia. Sensor Network Infrastructure for a Home Care Monitoring System. *Sensors*. 2014, roč. 14, č. 3, s. 3833–3860. ISSN 1424-8220. Dostupné z DOI: 10.3390/s140303833.

60. KG, OWIM GmbH & Co. *TEMPERATURE STATION*. OWIM GmbH & Co. KG, 2015. Version 03/2015. Č. Z31743A / C012015-PL / HU / CZ / SK.

61. A.S., ETA. *ME SENZOR 12*. ETA a.s., [b.r.].

62. S.R.O., HADEX spol. *CT60M Wireless Type Inactive Mode Infrared Intrusuion Dector*. HADEX spol s.r.o., [b.r.].

63. S.R.O., SHX Trading. *Wireless Dual Pet Immune PIR Dector*. SHX Trading s.r.o., [b.r.].

64. SHENZEN SONOFF TECHNOLOGIES CO., Ltd. *Sonoff DW2-RF*. Shenzen Sonoff Technologies Co., Ltd., [b.r.]. Ver. V 1.2.

65. *AliExpress - online shopping for popular electronics, fashion, home ...* [online]. aliexpress.com, 2023-02-20 [cit. 2023-02-20]. Dostupné z: https://www.aliexpress.com/item/1005003104522385.html?spm=a2g0o.order_list.order_list_main.51.21ef1802PNwVSO.

66. *Senzor TPMS tlaku v Pneu Ford* [online]. Auto Kora top s.r.o., 2023-03-20 [cit. 2023-03-20]. Dostupné z: https://www.fordshop.cz/p/senzor-tpms-tlaku-v-pneu-ford#.

67. *When one TPMS sensor fails - redi-sensor* [online]. Continental Automotive Systems, 2022-06-23 [cit. 2023-04-26]. Dostupné z: https://www.redi-sensor.com/when-one-tpms-sensor-fails/.

68. KRSTIC, Milos; SAVIC, Nemanja; KRAEMER, Rolf; JUNGHANS, Marek. Applying tire pressure monitoring devices for traffic management purposes. In: *2012 International Symposium on Signals, Systems, and Electronics (ISSSE)*. 2012, s. 1–6. Dostupné z DOI: 10.1109/ISSSE.2012.6374295.

69. S.R.O., APATOR METRA. *ELEKTRONICKÝ INDIKÁTOR TOPNÝCH NÁKLADŮ S INTEGROVANÝM RÁDIOVÝM VYSÍLAČEM E-ITN 30*. APATOR METRA s.r.o., [b.r.]. M2019/01a.

70. NETUP.CZ [online]. Apator Metra s.r.o., 2023-03-20 [cit. 2023-03-20]. Dostupné z: https://www.metra-su.cz/cs/produkt/elektronicky-indikator-topnych-nakladu-e-itn-30-s-integrovanym-radiovym-vysilacem.

71. JABLOTRON. *Bezdrátový ionizační detektor kouře JA-60S*. Jablotron, [b.r.]. MFH51201.

72. ECOLITE. *Dálkově ovládaná zásuvka EZ-7901*. Ecolite, [b.r.].

Obsah přiloženého média

- `readme.txt` ..... stručný popis obsahu média a strom souborů
- `rtl_sdr_reader` ..... adresář se zdrojovými kódy pro Windows a Linux
  - `lin_rtlsdr` . zdrojové kódy závěrečné práce spolu s návodem ke spuštění a projektem pro Cmake
  - `win_rtlsdr` zdrojové kódy závěrečné práce spolu s návodem ke spuštění a projektem pro MS VS 2022
- `latex_thesis` ..... zdrojová forma práce ve formátu LaTeX
- `thesis` ..... adresář s PDF této práce
  - `DP_Martin_Simunek_2023.pdf` ..... text práce ve formátu PDF