Systém pro sběr entropie z okolního elektromagnetického šumu

Bc. Jan Němec

Diplomová práce
2022

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav elektroniky a měření

Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

ıéno a příjmení:	**Bc. Jan Němec**
sobní číslo:	**A20209**
:udijní program:	**N1032A020003 Bezpečnostní technologie, systémy a management**
ɔecializace:	**Bezpečnostní technologie**
ɔrma studia:	**Prezenční**
ėma práce:	**Systém pro sběr entropie z okolního elektromagnetického šumu**
ėma práce anglicky:	**System for Entropy Collecting from Ambient Electromagnetic Noise**

Zásady pro vypracování

1. Vypracujte rešerši na téma současného stavu antén, včetně metod realizace a uplatnění při sběru entropie z okolního elektromagnetického šumu.
2. Vytvořte prototyp sběru signálu pomocí softwarem definovaného rádia a následného generovaní náhodných čísel.
3. Aplikujte prototyp do vybraného kryptografického systému.
4. Navrhněte metodu testování funkčnosti prototypu.
5. Otestujte a optimalizujte prototyp na základě navržené metody.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BALANIS, Constantine A. Antenna Theory: Analysis and Design. 4th ed. Hoboken, New Jersey: John Wiley, 2016. ISBN 9781118642061.
2. COLLINS, Travis F., Robin GETZ, Di PU a Alexander M. WYGLINSKI. Software-Defined Radio for Engineers. USA. Artech House Publishers, 2018. ISBN 9781630814571.
3. GONG, Lishuang, Jianguo ZHANG, Haifang LIU, Luxiao SANG a Yuncai WANG. True Random Number Generators Using Electrical Noise. IEEE Access [online]. 2019, 7, 125796-125805 [cit. 2021-11-30]. ISSN 2169-3536.
4. NAĎ, Andrej. Implementace Diehard testů pro testování generátorů pseudonáhodných čísel. Zlín: Univerzita Tomáše Bati ve Zlíně, 2018, 90 s. Dostupné také z: http://hdl.handle.net/10563/43244. Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav elektroniky a měření. Vedoucí práce Žáček, Petr.

Vedoucí diplomové práce: **Ing. Stanislav Kovář, PhD.**
Ústav bezpečnostního inženýrství

Datum zadání diplomové práce: **3. prosince 2021**
Termín odevzdání diplomové práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

Ing. Milan Navrátil, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 7. února 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.5.2022

Bc. Jan Němec v.r.

podpis studenta

ABSTRAKT

Hlavním cílem práce je navrhnout systém pro sběr entropie z okolního elektromagnetického šumu a následná úprava do podoby náhodných čísel. V úvodu práce je nastíněný současný stav v oblasti antén, softwarově definovaných rádií a generátorů náhodných čísel, včetně metod realizace a uplatnění. Další část práce se zabývá návrhem systému pro sběr entropie pomocí softwarově definovaného rádia. Využité hardwarové nástroje jsou tvořeny softwarovým rádiem HackRF One, v kombinaci s teleskopickou anténou ANT500, disponující frekvenčním rozsahem od 75 MHz do 1 GHz. Softwarové nástroje, používané v práci, zahrnují GNU Radio Companion, pro tvorbu základní struktury ovládání rádia a programovací jazyk Python, pro následné zpracování zachyceného signálu a implementace do aplikace. Zachycená entropie je podrobena analýze kvůli ověření náhodnosti a jsou stanoveny parametry pro dosažení nejlepších výsledků. Výzkum zahrnuje frekvenční analýzu, nezávislou na pozici bytu v čísle, výpočet entropie generovaných čísel a vývoj průměrné hodnoty závislé na pozici bytu v čísle. Závěrem práce je aplikace, ilustrující uplatnění náhodných čísel v kryptografii a využívající blokovou šifru AES-256 v GCM módu, který eliminuje nutnost zadávat inicializační vektor.

Klíčová slova: HackRF One, softwarově definované rádio; entropie, generátor náhodných čísel, skutečný generátor náhodných čísel, RNG, TRNG, Python, GNU Radio Companion

ABSTRACT

The main goal set for this thesis is to propose a method for gathering entropy from surrounding electromagnetic noise, followed by the transformation of said entropy into a random number. The introduction contains brief research on antennas, software-defined radios, and random number generators. Research would be focused on the realization method and usage in practice. Next in line is developing a technique for extracting entropy from electromagnetic noise using software-defined radio HackRF One, combined with telescopic whip antenna ANT500 designed for frequencies from 75 MHz to 1 GHz. Used software tools combine GNU Radio Companion to create basic radio controlling structure and programming language Python, processing harvested signal and implementation into showcase application. Acquiring entropy is subjected to analysis to test randomness and determine the best parameters. The set of tests comprises frequency analysis, independent of byte position, entropy calculation, and convergence of average byte value depending on byte position in a tested number. Lastly, all acquired knowledge serves to develop a showcase application to generate encryption keys for AES-256 in GCM mode, eliminating an initialization vector.

Keywords: HackRF One, software-defined radio; entropy, random number generator, true random number generator, RNG, TRNG, Python, GNU Radio Companion

Tento prostor bych rád využil abych poděkoval především mému školiteli doktoru Stanislavu Kováři za hodiny a hodiny práce, které strávil nad touto prací při kontrolování a konzultování. Také musím poděkovat svým přátelům za mentální podporu a odreágování když to bylo nejvíc potřeba, abych mohl nabrat síly před další probdělou nocí.

Ordinary people, they operate within a certain set of parameters, right? Rules. Limits. Then there's blokes like me, yeah? We cheat.
- John Constantine

OBSAH

ÚVOD

Základní myšlenkou práce je využití entropie, nacházející se v elektromagnetickém šumu, pro generování skutečně náhodných čísel. Využitelnost náhody jako proměnné nekončí pouze v oboru bezpečnosti, kde je často využívaná pro tvorbu šifrovacích klíčů, přístupových hesel, nebo PIN kódů, atd. Její využití je možné také v simulacích a designu.

Teoretická část práce je zaměřená na anténní rešerši, za účelem výběru vhodné antény pro fyzickou realizaci experimentu, uvedení do problematiky softwarově definovaných rádií, představení zařízení, které bude hardwarovým středobodem praktické části práce a samozřejmě náhodné generátory čísel.

První kapitola teoretické části postupně rozebírá základní typy antén, jejich charakteristiky a dělení do podkategorií, včetně příkladů realizací publikovaných v posledních letech. Další kapitola představuje softwarově definované rádio, jeho klady a zápory. Součástí je vysvětlení pojmů základní pásmo a kvadraturní vzorkování, jež jsou nedílnou součásti této radiotechniky. Konec kapitoly je věnovaný rádiu HackRF One [1], jeho nejdůležitějším součástkám, parametrům a činnostem při přijmu signálu z hardwarového hlediska. Poslední část teorie je zaměřená na generátory náhodných čísel, jejich vysvětlení a dělení. Kapitola obsahuje i důležité normy a standardy, například NIST 800-22 rev.1 [2]. Uvedený je rozdíl mezi pseudonáhodnými a skutečnými generátory náhodných čísel, spolu s několika příklady. Kapitola také pojednává o využití generátorů náhodných čísel v praxi, včetně metod testování náhodnosti čísla.

Praktická část je rozdělena do čtyř částí, které na sebe navazují. První kapitola pojednává o instalaci a manipulaci se softwarovým vybavením, potřebným pro ovládáním softwarově definovaného rádia. Následuje návrh sběru entropie a její transformace do náhodného čísla, včetně prezentace tři metod a jejich srovnání. Nejlepší metoda je v další kapitole podrobena sérii testů pro ověření náhodnosti a identifikace nedostatků pro jejich případné adresování ve finální implementaci. Což vede k poslední kapitole, která je rozdělena do dvou částí. První je implementace navrženého generátoru čísel do procesu tvorby uživatelského prostředí a druhá je implementace do tvorby klíčů využitelných v kryptografickém systému AES-256 [3].

I. TEORETICKÁ ČÁST

1 ANTÉNNÍ TECHNIKA

Jednou z nejdůležitějších částí jakéhokoliv radiového přijímače, nebo vysílače je anténa. Proto bude tomuto prvku věnována úvodní kapitola, která bude obsahovat průzkum vědeckých publikací vydaných v posledních letech, s cílem zjistit momentální trendy v designu a použití antén.

Kapitola se, vyjma vědeckých prací, opírá o publikaci "Antenna Theory: Analysis and Design"od C. A. Balanise [4].

Následující stránky budou obsahovat stručný popis typů antén, které jsou rozřazeny do základních šesti typů. Rozřazení je následující:

- drátové antény,
- otvorové antény,
- mikropásky,
- anténní pole,
- reflektorové antény,
- čočkové antény. [4]

Pokud se jedná o přijímací anténu, pak je její principiální funkcí, zlepšovat schopnost zařízení, pohlcovat elektromagnetické (EM) záření z okolí. Analogicky, vysílací anténa má schopnost zvětšovat EM záření, šířící se do okolí. V obou případech se toho dosáhne manipulací s tzv. směrovostí antény a resonanční frekvencí, kdy změnou tvaru dochází ke změně vyřazovacího diagramu v horizontální i vertikální rovině a změna velikosti vede ke změně resonanční frekvence. Obecně platí, že čím menší anténa, tím vyšší kmitočet přijímá. Tento efekt je dán přímou úměrou mezi vlnovou délkou a velikostí antény. [4]

1.1 Drátové antény

Drátové antény jsou pravděpodobně nejznámější a nejpoužívanější typ. Nachází se v prakticky každém mobilním či komunikačním zařízení, například anténa pro bezdrátové nabíjení telefonu, příjem mobilního signálu, RFID karty, nebo Wi-Fi anténa na přístupovém bodě. Drátové antény lze rozdělit do těchto podkategorií:

- prutové antény,
- dipól,
- smyčkové antény. [4]

1.1.1 Prutové antény

Jedná se o nejstarší a nejjednodušší druh antény. Vyvedený samostatný vodič s toroidní vyzařovací charakteristikou, který prochází středem toroidu. Prutové antény mohou být také teleskopické, čímž umožňují variabilně měnit frekvenci největšího zisku. Obecně platí, že největší zisk má anténa s délkami v rozsahu od $\lambda/2$ až $5\lambda/8$. Nejběžnější je však délka $\lambda/4$. [4]

Prvním příkladem je prutová anténa, navržená pro velmi široké frekvenční pásmo s impedancí 50 Ω v celém pásmu. Publikace [5] uvádí měření od frekvence 1 Hz po 2,5 GHz. Parametr S_{11} dosahoval -10 dB ve frekvenčním pásmu 194 MHz až 2124 MHz, -20 dB na frekvenci přibližně 750 MHz, jak odhalilo měření. Realizace spočívala ve vytištění na flexibilní plošný obvod a stočení do tubusu. První polovina tubusu je rozdělená na zemnící část, kterou půlí přívodní vodič a spolu tvoří koplanární vlnovod. [5]

Druhá zkoumaná anténa není pouze prutová anténa, ale kombinace prutové antény se čtyřmi cívkovými anténami. Prototyp byl testovaný ve frekvenčním rozsahu 1 GHz až 2 GHz. Signál byl přivedený pomocí koaxiálního kabelu s impedancí 50 Ω. Lepší výsledky anténa vykazovala přibližně mezi 1,25 GHz až 1,4 GHz, kde parametr S_{11} dosahoval hodnoty menší nebo rovny -10 dB. Anténa byla designovaná jako prutová anténa, okolo které se stáčely čtyři cívkové antény. Její hlavní výhodou je kardiodní charakteristika cívkové antény, jež však vyzařuje i dozadu. Přidání prutové antény toto vyzařování omezilo. [6]

1.1.2 Dipóly

Dipólem se rozumí anténa, která má základní vyzařovací charakteristiku toroidního tvaru. Ve zjednodušené verzi se jedná o dva rozevřené, elektricky nespojené, vodiče. Dipóly je například možné použít jako součást fázově řízeného anténního pole. Toto bylo využito v radarovém systému Duga-3 [7] poblíž jaderné elektrárny v Černobylu. [4]

První zkoumaný dipól je součástí fázově řízeného pole pro aplikaci v 5G sítích, tedy pro aplikace na frekvenci 28 GHz. Anténa disponuje S_{11} parametrem nižším než -10 dB ve frekvenčním pásmu 26 GHz až 31 GHz, na frekvenci 28,7 GHz skoro -30 dB. Jedná se o magneto-elektrický dipól, tištěný na několikavrstvou desku plošných spojů (PCB), který na povrchu nese 4 radiační elementy. Protože se jedná o vícevrstvé PCB může tato dipólová anténa být nízkoprofilová a tištěná spolu s ostatními anténami v poli. [8]

Další zkoumaný dipól je od skupiny ruských vědců, kteří vyrobili mikropáskové dipólové pole pro využití v radarovém systému. Frekvenčně testované pásmo bylo mezi 650 MHz až 750 MHz, největší zisk byl 13 dB na frekvenci 710 MHz a nejnižší 12

dB na frekvenci 650 MHz. Vzhled antény je založený osmiúhelníku, který je z každé strany osazený dvojicí dipólů, což umožňuje měnit vyzařovací charakteristiku po celém obvodu krokovým řízením a tedy skenovat po celém obvodu. [9]

1.1.3 Smyčkové antény

Smyčkové antény jsou nejčastěji využívané pro frekvence mezi 3 MHz až 3 GHz. Na rozdíl dipólu, se jedná o uzavřenou smyčku, či cívku, což ovlivňuje toroidní vyzařovací charakteristiku, nicméně vodič neprochází středem toroidu, ale vzniká zde volný prostor. Tento vytvořený gradient lze použít například pro EMS zkoušky [10], nebo jako sonda blízkého pole. Charakteristické pro smyčkovou anténu je, že se jedná o více, či méně planární anténu. [4]

První příklad je miniaturní smyčková anténa, navržená pro použití v bezdrátových sluchátkách, jejíž velikost dosahuje pouze 7 milimetrů v průměru. Kvůli funkci sluchátek bylo nutné, aby disponovala přívětivými parametry pro frekvence 2,4 GHz, 5 GHz a 6 až 10 GHz. Účinnost antény je variabilní, resp. závislá na zkoumaných frekvencích. Nejnižší naměřená účinnost byla 50% na 2,4 GHz a nejvyšší, přibližně 94%, na 6 až 10 GHz. Parametr S_{11} byl na frekvenci 2,4 GHz téměř -30 dB, na zbylých zkoumaných frekvencích -10 dB. [11]

Druhá anténa je navržená pro frekvenční pásmo 2,3 až 3,1 GHz. Jedná se o čtveřici smyčkových antén, přičemž všechny mají dva porty. První dvě používají protifázový proud, zatímco zbývající soufázový proud. Pro úsporu místa jsou naskládané na sobě a jejich S_{11} je ve zmíněném frekvenčním pásmu nižší než -10 dB, v nejnižším bodě téměř -20 dB. Autoři anténu srovnávali s dalšími typy a dospěli k závěru, že jejich anténa měla nejmenší rozměry a druhý největší frekvenční rozsah. [12]

1.2 Otvorové antény

Otvorové antény se stávají populárnější, kvůli zvýšené poptávce po vyšších kmitočtech. V podstatě fungují podobně jako optické kabely a záření se odráží od stran. Velkou výhodou je možnost instalace tak, aby pouze vyúsť antény byl na povrchu a celé tělo bylo přitom bezpečně uloženo uvnitř předmětu, nebo objektu. Vzniklý otvor je možné překrýt dielektrikem, čímž dojde ke kompletnímu zakrytí. Toto je výhodné zejména pro aplikace ve vesmírném, nebo leteckém průmyslu. Otvorové antény se dělí na:

- horn antény,
- vlnovody.

1.2.1 Horn antény

Horn antény představují silně směrové antény, vyzařující ve směru od otvoru rohu. Mohou nabývat různých tvarů, jak kuželu s elipsovitou nebo kruhovou podstavou, tak jehlanu se čtvercovou a obdélníkovou podstavou. [4]

První příklad je miniaturní horn anténa s frekvenčním pásmem 33 GHz až 67 GHz. Vytvořena byla na 3D tiskárně s využitím technologie selektivního laserového vytavování (SLM) a pomocí stereolitografie (SLA), tedy selektivního polymerního vytvrzování. Při srovnání dvou výrobních technologií byl změřený S_{11} a autoři dospěli k závěru, že SLA metoda tisku antén dosahuje lepších výsledků. S_{11} po celém frekvenčním pásmu nebyla vyšší než -10 dB. [13]

Druhý příklad je zajímavý tím, že se jedná o nastavitelnou horn anténu. Stěny horn antény jsou tvořeny z nevodivého dutého materiálu, kterým může protékat voda. Voda v tomto případě tvoří odrazivou plochu pro elektromagnetické záření, to umožňuje měnit tvary výsledné horn antény. Autoři hovoří o čtyřech nastaveních, E rovina, H rovina, pyramida, vlnovod. Všechna nastavení antény mají S_{11} parametr pod -10 dB ve frekvenčním rozsahu 2,5 GHz až 5 GHz. [14]

1.2.2 Vlnovody

Vlnovody, stejně jako horn antény, jsou trubky z elektricky vodivého materiálu, uvnitř kterého dochází k odrazům, proto se jedná o silně směrovou anténu s rozdílem, že vyúsť vlnovodu je obdélník, čtverec, kruh, nebo elipsa. Nedochází zde k rozšíření struktury, které by upravovalo vyzařování v blízkosti antény, takže bezprostřední blízkosti má větší emise po stranách. [4]

Příklad vlnovodu byl již zmíněný v článku, využívajícím vodu jako odrazivou plochu [14].

1.3 Mikropásky

Mikropásky jsou typy antén, které jsou konstruovány na tištěných spojích. Její konstrukce se skládá z pásky, která může mít různé tvary, nevodivého substrátu a zemnění. Podobně jako otvorové antény, i mikropásky mohou být využity v leteckém a vesmírném průmyslu, protože nevytváří téměř žádný odpor. [4]

Mikropáskové antény jsou často součástí PCB a byly již uvedeny v minulých případech. Poprvé u antény využívané v 5G sítích [8] a podruhé u skládané smyčkové antény [12].

1.4 Reflektorové antény

Reflektorové antény jsou běžně používané pro satelitní vysílání, či jiné aplikace vyžadující bezdrátové vysokorychlostní připojení na vzdálenost několika kilometrů. [15] Jejich funkce je vcelku jednoduchá, protože využívají geometrického tvaru usměrnění odražených vln. Antény disponují vysokou směrovostí, proto se pro ladění udává kromě směru natočení, také azimut. Reflektorové antény je možné rozdělit do dvou podkategorií, které specifikují geometrii reflektoru:

- parabolický reflektor,
- koutový odražeč. [4]

1.4.1 Parabolický reflektor

Parabolický reflektor představuje nejčastější typ reflektorové antény, protože existuje ve dvou variantách. První varianta má zdroj záření před reflektorem, druhá varianta má zdroj záření v reflektoru a září reflektor. Při správném geometrickém tvaru reflektoru a umístění zářiče dochází k homogenizaci záření a anténa je natolik směrová, že může navázat datové spojení i na vzdálenost desítek kilometrů. [4]

1.4.2 Koutový odražeč

Koutový odražeč má tvar kuželu, nebo jehlanu a funkci stejnou jako parabolický reflektor. Jeho schopnost homogenizace však není natolik dokonalá a svazek EM paprsků se s narůstající vzdáleností více rozchází. Výhodou je jednodušší výroba. [4]

Představený koutový odražeč je podobně jako [14] zajímavý tím, že je nastavitelný. Jeho nastavitelnost je dána odrazivou plochou, ve formě plazmy. Vyrobená anténa se skládá ze zářivkových těles ve dvou řadách a byla testována pro frekvenční pásmo 2 GHz až 4 GHz. S_{11} parametr nebyl vyšší než -10 dB ve frekvenčním rozsahu 1,9 GHz až 3 GHz. Zářičem v tomto případě byla jednoduchá prutová anténa. Využitím plazmy, bylo možné udělat z všesměrové antény koutový odražeč a výrazně omezit vyzařovaní v nežádoucích směrech. [16]

1.5 Anténní pole

Anténní pole představuje soustavu antén ve 2D prostoru. Nemusí se vždy jednat o identický typ antény, což potvrzuje například oblíbená Yagi-uda anténa, která zahrnuje kombinaci reflektoru, dipólu jako zářiče a směrovače tvořeného smyčkovou anténou. Nejčastější anténní pole jsou:

- Yagi-Uda,

- mikropáskové pole,
- otvorová síť,
- štěrbinový vlnovod. [4]

1.5.1 Yagi-Uda

Tento typ antény je vcelku speciální, protože kombinuje několik typů antén. Skládá se z reflektoru, který limituje záření za anténu. dipólu fungujícího jako zářiče a série smyčkových antén, které díky tvorbě gradientu ve středu svého obvodu směrují záření. Tato anténa byla často používaná pro staré televizní vysílání. [4]

Příklad je založený na publikaci [17], protože přístup autorů byl trochu odlišný a celá anténa je složena z prutových antén. Anténní pole se skládá ze 14 Yagi-Uda antén v kruhové konfiguraci. Nejvyšší zisk anténa dosahuje na 3,1 GHz při spuštěných 4 anténách, a také má v takový moment největší směrovost. Protože se jedná o Yagi-Uda anténu složenou z prutových antén, vzniká problém s vyzařováním za anténu, což se promítne do vyzařovacího diagramu, ale právě při čtyřech spuštěných segmentech je tento efekt nejmenší. [17]

1.5.2 Mikropáskové pole

Nejčastější využití mikropásku je právě v poli. Jednotlivé pásky jsou naskládané do různých tvarů, což ovlivňuje jejich zisk a také vyzařovací charakteristiku, čímž umožňuje lépe přizpůsobit zařízení dané aplikaci. [4]

Mikropásková pole jsou jedny z nejoblíbenějších moderních metod designu antén, protože je lze tisknout najednou na PCB a při kombinaci s fázovým řízením pro určení směrovosti, umožňuje levnou a rychlou tvorbu sofistikovaných anténních systémů. Příklady použití byly již popsány u 5G [8] a radarových systémů [9].

1.5.3 Otvorová síť

Stejně jako u mikropáskového pole, otvorová síť skládá jednotlivé výstupy vlnovodů, nebo rohových antén do pole. Rozdílem je vznik pole z jedné velké otvorové antény překrytím vyústě antény mřížkou z elektricky vodivého materiálu. [4]

Příklad otvorového pole z publikace [18] je navržený pro frekvenční spektrum 300 MHz až 1 GHz. Autoři navrhli osmiúhelníkové pole a provedli, jak simulace, tak reálné měření. Výsledný S_{11} parametr ukázal u simulací hodnotu nižší než -10 dB pro frekvence 0,4 až 1 GHz, což koresponduje s reálně naměřenými hodnotami. [18]

1.5.4 Štěrbinový vlnovod

Štěrbinový vlnovod je speciální úprava vlnovodu, přičemž se již nejedná o celistvou trubku s maximálně dvěma otvory, ale po stranách vlnovodu jsou vytvořeny otvory. Tyto otvory vytváří emise okolo vlnovodu, čímž limitují směrovost. [4]

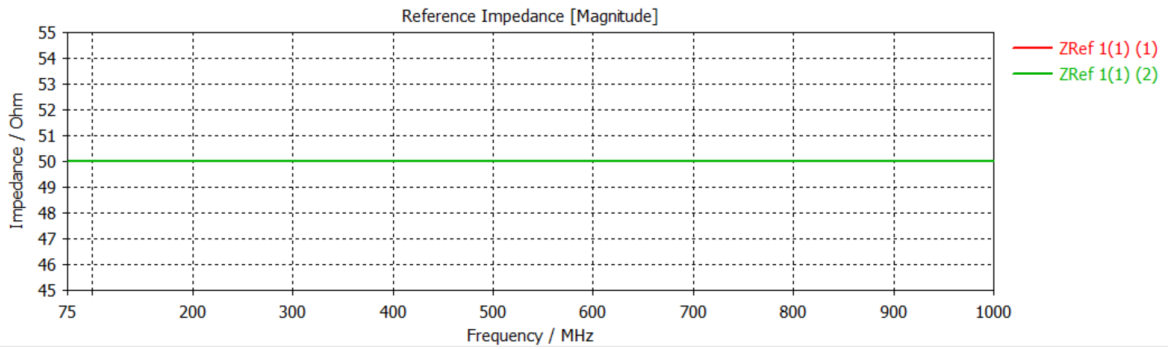
Příkladem štěrbinového vlnovodu může být vlnovod, navržený pro aplikaci ve vesmírném průmyslu se zabudovanou filtrovací funkcí přijatého signálu. Této funkce bylo docíleno vložením elektrický vodivých panelů, do spodní části vlnovodu. Horní část vlnovodu je opatřena obdélníkovými otvory o velikosti přibližně $\lambda/2$. S_{11} parametr byl přibližně -20 dB pro frekvenční pásmo 12,25 GHz až 12,75 GHz, tedy pracovní spektrum antény. [19]

1.6 Výběr vhodné antény pro sběr entropie

Zhodnocením vlastností všech šesti základních typů antén, byl vytvořen závěr, že antény nedisponující všesměrovou charakteristikou, by pro sběr entropie byly vhodné pouze za speciálních případů, jako je například sběr entropie z kosmického záření, kde by byla vhodná reflektorová anténa. Všesměrovostí disponují některé varianty anténních polí, nebo drátové antény. Z těchto dvou možností byla vyřazena anténní pole, protože se jedná o prototypiální návrh, kde by nebylo možné využít vlastnosti například mikropáskových polí být přímo na plošném spoji. Budoucí výzkum v oblasti, by však aplikaci tohoto typu antény měl blíže zkoumat.

Z drátových antén byly na výběr tři možnosti, z nichž jedna nebyla vhodná již z počátku. Smyčkové antény jsou vhodné pro blízká pole a snímají hlavně magnetickou složku signálu, práce se však zabývá spíše polem vzdáleným, které je dynamičtější a hůře ovlivnitelné bez komplexní techniky. Z toho důvodu se rozhodovalo mezi elektrickým dipólem a monopólem. V tomto případě byl vybrán elektrický monopól, protože existují varianty teleskopických prutových antén, které mají širší frekvenční rozsah a jsou tedy nejvhodnější pro tvorbu prototypu. Zvolená teleskopická prutová anténa disponuje potřebnými vlastnostmi, tedy frekvenční variabilitou a všesměrovostí.

Využívaná anténa bude ANT500 [20], čili se bude jednat o teleskopickou prutovou anténu, která je doporučená k softwarově definovanému rádiu HackRF One a disponuje frekvenčním rozsahem 75 MHz až 1 GHz. Vzhledem k velikosti frekvenčního rozsahu bylo vhodné, alespoň v simulačním prostředí programu CST Studio Suite [21], ověřit, zda impedance antény zůstává 50 Ω po celém frekvenčním spektru. Tato simulace byla provedena pro plně rozvinutou anténu o délce 88 cm, i pro složenou anténu o délce 20 cm.

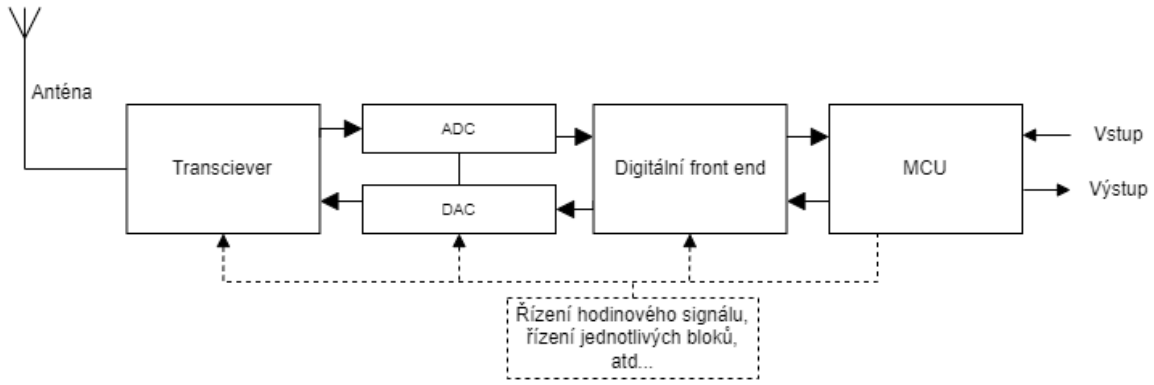


Obrázek 1.1 Změna impedance antény podle frekvence (1): 20 cm; (2): 88 cm

Podle grafu zobrazeného na Obrázek 1.1 se impedance s rostoucí frekvencí ani v jednom případě nemění. Tato skutečnost je platná pouze pro uvedený frekvenční rozsah, jelikož se jedná o pracovní pásmo antény. Na vyšších frekvencích však může docházet k impedančnímu kolísání. Nicméně vyšší frekvence nebyly předmětem zkoumání, neboť horní mez testované antény je 1 GHz.

2 SOFTWAROVĚ DEFINOVANÉ RÁDIO

Softwarově definované rádio (SDR) je zařízení využívající programovatelné obvody pro zpracování přijímaného, nebo vysílaného signálu. SDR umožňuje používat různé komunikační protokoly, měnit frekvenční rozsah, nebo jinak pracovat s přijatým signálem, pouhou změnou ovládacího softwaru, bez nutnosti zásahu do hardwarové části. [22]



Obrázek 2.1 Základní blokové schéma SDR - převzato z [22]

SDR se skládá ze čtyř základních bloků, při zanedbání filtrů, zesilovačů a spojení převodníků do jednoho bloku, jak je znázorněno na blokovém schématu na Obrázek 2.1. Při pohledu zleva, schéma začíná anténou, která je naladěna pro impedanci 50 Ω, jak je dobrým zvykem. Prvním blokem je transceiver, tedy kombinace vysílače a přijímače. Ten přesouvá přijatý signál obvykle do základního pásma a to buď přímo do převodníku signálu, nebo jej předtím rozděluje na I a Q složku. Další prvek je zmíněný převodník signálu ADC nebo DAC, označení se liší na základě převodu, tedy analogový signál na digitální (ADC) nebo naopak (DAC). Předposlední blok je digitální front end, který obsahuje zásobník pro bity, programovatelné hradlové pole pro rozdělení do kanálů, nebo synchronizaci. Posledním blokem je mikrokontroler (MCU), který ovládá celé rádio, pomocí elektricky ovládaných přepínačů nebo ovládacích vstupů na čipech, celé rádio nastavuje, ovládá v průběhu činnosti a zajišťuje přenos dat. [22]

2.1 Základní pásmo

SDR se pyšní velkým frekvenčním rozsahem, což zajišťuje směšovač, který umožňuje přesouvat sledované frekvenční pásmo do základního pásma, tedy poblíž 0 Hz. Kvůli tomuto přesunu, se u vzorkovací frekvence nevyužívá jednotka Hertz, ale *sps* neboli Sample per second, česky vzorek za vteřinu. Tato změna existuje, protože se aplikuje vzorkovací frekvence pouze do několika MHz, která ovšem vzorkuje frekvenci na vstupu antény řádově vyšší, například v GHz. Což by neodpovídalo Shannon-Kotelníkovu teorému [23]. Přesun do základního pásma, ovšem kromě výhody pracovat s širším frek-

venčním rozsahem, nese i nevýhodu. Ta se projevuje ve formě špičky, která je generovaná na laděné frekvenci a způsobená hodinovým signálem používaným ve směšovači. Tato špička je označovaná jako DC spike. Používané metody eliminace špičky nejsou aplikovatelné pro bezdrátová zařízení, proto se využívá spíše principu ladění rádia na frekvenci blízkou frekvenci požadované a následné doladění na požadovanou frekvenci. Tato metoda je mnohem jednodušší a dovoluje zachovat data na požadované frekvenci. [22]

2.2 Kvadraturní vzorkování

Kvadraturní, komplexní nebo I/Q vzorkování, jsou názvy pro vzorkování, které využívá SDR i mnoho další digitálních přijímačů a vysílačů. Jeden z názvů napovídá, že vzorkování pracuje s jistou I, která znamená in-phase, česky ve fázi a Q složkou, která znamená quadrature, tedy kvadratická. Na I lze nahlížet jako na reálnou složku a na Q jako na imaginární složku posunutou o 90°. [22]

2.2.1 Základní pojmy

Uvedené pojmy jsou nezbytné pro správné pochopení obsahu této podkapitoly:

- vzorkování,
- kvantování,
- komplexní číslo.

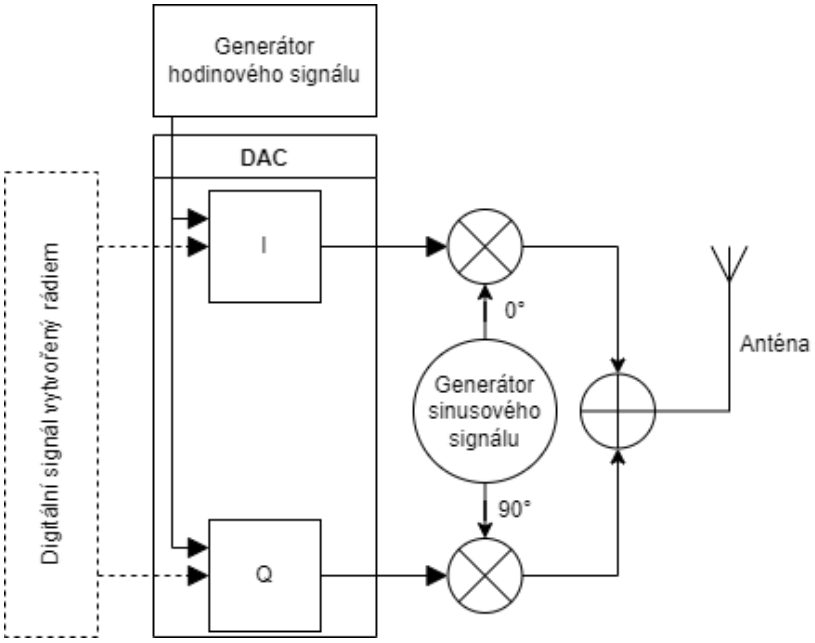
Vzorkováním se rozumí proces snímání výšky signálu, při zvolené frekvenci, vzorkovací frekvenci. Při dostatečné vzorkovací frekvenci, je pak možné zrekonstruovat vzorkovaný signál, propojením všech nasnímaných bodů. Dostatečnou frekvencí se rozumí alespoň dvakrát větší vzorkovací frekvence, než je nejvyšší frekvence vzorkovaného signálu. Toto tvrzení bylo již několikrát potvrzeno a nese název Shannon-Kotelníkův teorém [23]. V případě kvadraturního vzorkování však dochází k záznamu, jak I, tak Q hodnoty, tedy v jednom vzorku, je obsaženo dvakrát tolik informace a lze říci, že vzorkovací frekvence je rovná nejvyšší frekvenci vzorkované [24]. [22]

Kromě vzorkování, je důležité pochopit proces kvantovaní. Ten probíhá současné se vzorkovacím procesem a dochází při něm k omezení možných hodnot snímaného signálu, ze spojitého spektra na předem definované úrovně. Vlivem této diskretizace signálu, dochází ke ztrátě informace. Tato ztráta se zmenšuje s počtem možných úrovni signálu, přičemž počet kvantovacích úrovni se u přístrojů uvádí nejčastěji v bitech. U použitého zařízení HackRF One je kvantovací rozsah 8 bitů, snímání signálu tedy může nabývat 2^8 úrovní, čili 256 různých hodnot. [22]

Poslední velmi důležitý pojem, spojený s kvadraturním vzorkováním, je komplexní číslo. Toto číslo je na rozdíl od čísla reálného vektor, tedy obsahuje složku reálnou a složku imaginární. Jeho zápis je značený malým písmenem i, nebo v elektrotechnice malým písmenem j. Tento rozdíl ve značení je zavedený z důvodu vyvarování se záměny s okamžitou hodnotou elektrického proudu, značenou i. [22]

2.2.2 SDR v podobě vysílače

Z pohledu SDR jako vysílače, je výhodná kvadraturní modulace, zejména z hlediska modulace signálu. Protože vysílaný signál je násobená I a Q složka, které jsou k sobě ortogonální a vysílač může pouhou změnou amplitudy těchto složek, měnit amplitudu a fázi výsledného signálu. [22]



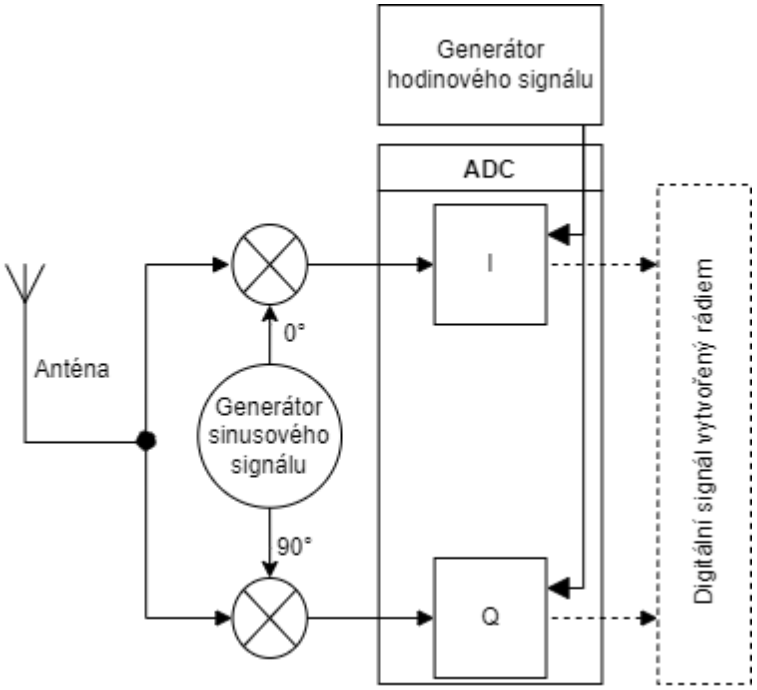
Obrázek 2.2 Základní blokové schéma vysílače - převzato z [22]

Z blokového diagramu na Obrázek 2.2 je vidět, že výstupy z převodníku signálu jsou dva, první je I složka, a druhý je Q složka. Z hlediska fáze se s I složkou vůbec nemanipuluje a je násobena čistým sinusovým signálem. Q složka je na druhou stranu násobena sinusovým signálem posunutých o 90°, tedy kosinusovým signálem. Výsledné komplexní složky jsou spojeny a vyslány do zesilovače, nebo přímo antény. [22]

2.2.3 SDR v podobě přijímače

Z pohledu přijímače je využití kvadraturního vzorkování způsob levného dosažení širokého pásma. Jak již bylo uvedeno v podkapitole zabývající se pojmy, není potřeba dosáhnout dvojnásobku vzorkované frekvence, ale pouze její hodnoty. SDR přesouvá

sledované pásmo do základního pásma, což znamená, že z hlediska vzorkování, již nezáleží na původní frekvenci. Jinými slovy nezáleží, zda byla frekvence 100 MHz nebo 2,4 GHz. [22]



Obrázek 2.3 Základní blokové schéma přijímače - převzato z [22]

Jak je tedy uvedeno na Obrázek 2.3, analogový signál je rozdělený do dvou větví, *I* větev vstupuje do ADC převodníku beze změny a *Q* větev je před vstupem do převodníku posunuta o 90°. Tím se zdvojnásobí množství informace o signálu, který potom může být zpracován. Tento proces je možné sledovat také na blokovém schématu rádia HackRF One v čipu MAX2837 na Obrázek 2.5. Fakt, že signál je předávaný ovládacímu softwaru pro další zpracování ve formátu komplexního čísla, umožňuje provádět efektivnější rychlou Fourierovu transformaci (FFT), která vyžaduje informaci o fázi. Pokud tuto informaci nemá k dispozici, převod do frekvenční domény vyžaduje více výpočetního výkonu. [22]

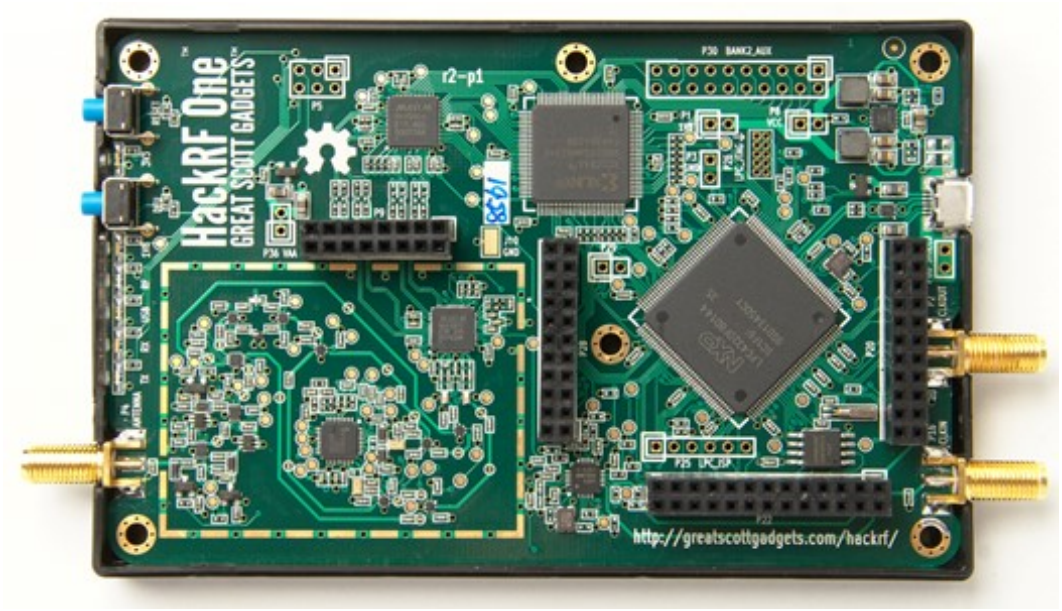
2.3 HackRF One

V této práci používané SDR je HackRF One, které bylo vytvořeno společností Great Scott Gadgets, specializující se na vybavení pro penetrační testery. Toto zařízení je zaměřeno zejména pro hardwarový hacking, nebo odposlech komunikace. Samotné SDR disponuje následujícími parametry:

- poloduplexní transceiver – rádio může buď vysílat, nebo přijímat,

- operační frekvence 1 MHz – 6 GHz,
- kvadraturní vzorkování,
- vzorkovací frekvence 2 Msps – 20 Msps,
- rozlišení I–8 bitů a Q–8 bitů,
- napájení skrze USB,
- SMA konektory pro anténu i generátor hodinových signálů,
- 50 Ω impedance,
- vysílání maximálně 3,3 V a 50 mA. [1]

Uvnitř rádia se nachází i konektory pro rozšiřující karty, připojení externího generátoru hodinového signálu, nebo debugging. Směšovač a transceiver jsou opatřeny stíněním, protože celé zařízení jinak chrání pouze plastový obal. Odhalené rádio ilustruje Obrázek 2.4. [1]



Obrázek 2.4 HackRF One [1]

2.3.1 Složení HackRF One

Rádio obsahuje všechny základní součásti SDR, které je možné vidět na Obrázek 2.1. HackRF One je však rozšířen o směšovač signálu, který zajišťuje mnohem širší frekvenční rozsah. Jednotlivé důležité obvody jsou jmenovitě tyto:

- MCU – LPC4320 [25],

- složitý programovatelný logický obvod (CPLD) – XC2C64A [26],
- ADC/DAC převodník - MAX5864 [27],
- transciever – MAX2837 [28],
- směšovač - RFFC5072 [29],
- generátor hodinových signálů - Si5351C [30].

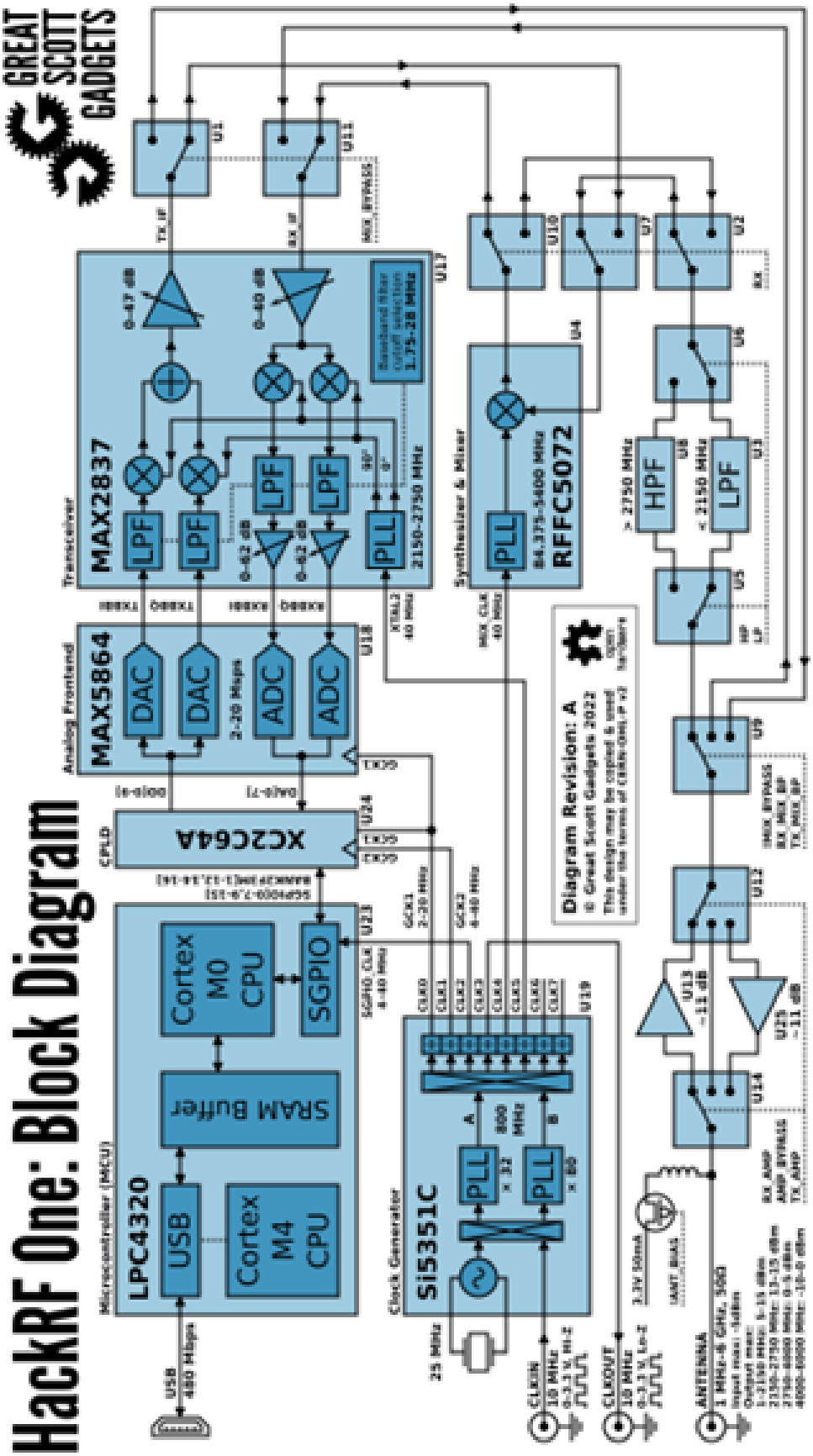
MCU, nacházející se v rádiu, je vyrobený firmou NXP Semiconductors a založený na ARM architektuře, přičemž obsahuje hlavní procesor M4 s pracovní frekvencí 204 MHz, stejně jako pomocný procesor M0. Dále disponuje SRAM pamětí 264 kB pro kód uživatele. CPLD je od firmy Xilinx, která je součástí společnosti AMD. CPLD je složené z CMOS obvodů s výrobní technologií 0.18 μm, což umožňuje měnit nastavení za běhu. Převodník signálu od společnosti Maxim Integrated umožňuje převádět signál oběma směry. Disponuje vzorkovací frekvenci 22 Msps pro oba směry převodu a jeho digitální výstupy mají rozlišení 10 bitů. Transciever je rovněž od společnosti Maxim Integrated. Je navržený pro práci ve frekvenčním pásmu od 2,3 GHz až 2,7 GHz, tedy pro Wi-Fi, či 4G/LTE aplikace. Pro přepínání mezi vysíláním a přijímáním vyžaduje pouze 2 μs. Další důležitou vlastností je integrovaná a plně nastavitelná dolní propust, pro frekvence od 1,75 MHz po 28 MHz. Směšovač je vyrobený společností Qorvo a jedná se o vysokofrekvenční směšovač signálu. Pracuje na frekvenčním rozsahu od 30 MHz do 6 GHz, jak na vstupu, tak na výstupu, čímž vytváří rádiu široký frekvenční rozsah. Generátor hodinových signálů, vyrobený společností Skywork Solutions, disponuje vlastností resynchronizace hodinového signálu, pokud nastane jeho rozladění postupem času. Jedná se tedy o programovatelný generátor hodinového signálu. Nabízí osm výstupů, přičemž maximální výstupní frekvence je 200 MHz a maximální vstupní frekvence je 28 MHz.

2.3.2 Popis funkce při přijmu signálu

Rádio je využívané pouze jako přijímač, proto je popis funkce zaměřený jen na tuto činnost, bez ohledu na vysílací vlastnosti. Nicméně funkce se příliš neliší, proto uvedený popis lze aplikovat i pro vysílací část. [1]

Z blokového schématu na Obrázek 2.5 je zřejmé, že přijaty signál je vhodné nejdříve zesílit, použitím předzesilovače na 10 dBmW, přičemž signál následně pokračuje buď do směšovače anebo rovnou do transceiveru. Výběr cesty záleží na laděné frekvenci, zařízení je však navržené primárně pro frekvence okolo 2,4 GHz. V případě, že signál vstupuje do směšovače, pak je pomocí fázového závěsu konvertovaný na frekvenci 2,4 GHz. V transceiveru je signál nejdříve rozdělený na dva. Z toho jeden je posunutý o 90°,

čímž se stává Q složkou signálu, nebo také imaginární částí a druhý zůstává nezměněný, proto je z něj I složka, tedy reálná složka. Tyto dvě složky prochází filtrem, který frekvenci mění opět, tentokrát na základní pásmo, tedy poblíž 0 Hz. Tento transformovaný signál dále postupuje do dolní propusti, která podle nastavené šířky pásma propouští mezi 1,75 MHz až 28 MHz. Následujícím blokem je A/D převodník, kde vzorkovací frekvence, pokud není pevně nastavena jinak, koresponduje s frekvencí dolní propusti a převádí signál. Díky využití kvadraturního vzorkování, je tedy možné získat frekvenční rozsah 20 MHz, ačkoliv zařízení disponuje pouze 20 Msps. Digitální signál postupuje do CPLD, který zastává funkci asistenta pro MCU a díky jeho vnitřnímu zásobníku, může předávat data. Mikrokontroler následně zakóduje data pro přenos skrze USB a posílá je do počítače, případně jiného ovládacího zařízení. V případě vysílání zařízení funguje opačně. [1]



Obrázek 2.5 Blokový diagram SDR [1]

3 GENERÁTORY NÁHODNÝCH ČÍSEL

Generátory náhodných čísel mají dvě základní rozdělení, jmenovitě generátory pseudonáhodných čísel a generátory kvantové. První z nich používá formu matematické funkce. Druhý, též označovaný jako pravý generátor náhodných čísel, vychází principu nepředvídatelných fyzikálních jevů pro generování náhodných čísel. [31]

3.1 Normy a standardy spojené s generováním náhodných čísel

Náhodná čísla jsou používaná pro bezpečnostní aplikace, ať už se jedná o generování šifrovacích klíčů, nebo přístupových hesel a pinů, proto existují také standardy a normy, které mají zajistit kvalitu generovaných hodnot. Mezi nejpoužívanější standardy patří tyto:

- NIST SP 800-22 rev.1 [2],
- NIST SP 800-90 [32], [33], [34],
- AIS 20 [35],
- AIS 31 [36],
- ISO/IEC 18031:2011 [37].

3.2 Generátory pseudonáhodných čísel

Generátory pseudonáhodných čísel, také známé jako deterministické generátory náhodných čísel, využívají matematických modelů k tvorbě řady zdánlivě náhodných čísel [31]. Důvodem proč jsou pseudonáhodné je skutečnost, že vyžadují inicializační hodnotu, anglicky zvanou seed, od které se výsledná číselná řada odvíjí. V případě odhalení inicializační hodnoty, lze automaticky předpovědět všechna další generovaná čísla. [35]

Pseudonáhodné generátory se dělí na lineární a kryptografické, přičemž hlavní rozdíl spočívá v jejich chování. Zatímco u lineárních pseudonáhodných generátorů stačí znát dostatečný počet předchozích stavů pro odvození stavů následujících, nebo odvození klíče, u kryptografických by tento postup neměl být realizovatelný. Jejich testování se provádí jako u všech generátorů náhodných čísel statistickými testy, pro kryptografické pseudonáhodné generátory čísel to může být například NIST SP 800-90A rev.1 [32].

Z hlediska testování lze generátory rozdělit do čtyř kategorií K1 až K4, přičemž pouze K3 a K4 se považují za validní pro aplikaci v kryptografii [31]. Tyto kategorie jsou specifikované v dokumentu pro klasifikaci pseudonáhodných generátorů a vysvětleny takto:

- K1 - vysoká pravděpodobnost, že generovaná čísla jsou rozdílná,
- K2 - generovaná čísla jsou nerozeznatelná od skutečně náhodných podle specifických testů, obsahující i monobit,
- K3 - pro útočníka by mělo být nemožné zjistit jakoukoliv předchozí nebo následující hodnotu,
- K4 - pro útočníka by mělo být nemožné zjistit vnitřní stav generátoru. [31]

3.2.1 Známé pseudonáhodné generátory čísel

Zde jsou dva známé pseudonáhodné generátory čísel:

- Metoda středu čtverce [38],
- Dual EC DRBG [39].

První zmíněný navrhl John von Neumann, který by mohl být označen za otce moderní výpočetní techniky, protože je i autor von Neumannova schématu. Tento generátor čísel využívá druhé mocniny. Inicializační hodnota je umocněna na druhou a následně jsou zleva doplněny nuly, pokud je to nutné pro získání středu výsledného čtverce. Tento střed představuje výstup a zároveň další inicializační hodnotu pro následující iteraci. Generátor nebyl příliš vhodný, protože časem začal generovat opakující se hodnoty, nebo pracuje v cyklu [38]. Problémem se zabýval Bernard Widynski, který metodu doplnili o Weylovu sekvenci a značně ho zredukoval [40].

Druhý generátor budil značnou kontroverzi, protože se nacházel v první verzi doporučení amerického národního institutu pro standardy a technologie. Obsahem tohoto doporučení NIST SP 800-90A[39] byl kromě způsobů testování deterministických náhodných generátorů čísel, také algoritmus využívající dvojité eliptické křivky. Podle odborníků však tento generátor obsahoval několik chyb, způsobených národní bezpečnostní agenturou (NSA), proto jej úřad v další verzi NIST SP 800-90A rev.1 [32] odstranil.

3.3 Skutečné generátory náhodných čísel

Skutečné generátory náhodných čísel sdílejí jednu společnou vlastnost, kterou je využívání fyzikálních jevů a entropie, jež vzniká v jakémkoliv reálném systému. Tato entropie může vycházet z výrobních vad, nebo nepředvídatelných fyzikálních interakcí [41]. Entropie, kterou tyto generátory mohou získávat z uzavřených systému, myšleno například sledování svazku fotonů, kde vliv okolí je minimální, nebo žádný [42]. Další

variantou je sledování entropii otevřeného systému, čili světa. Monitorovat je možné elektromagnetický šum, termodynamické jevy [43], nebo kosmické záření [44].

Největší úskalí při tvorbě skutečných generátorů náhodných čísel je způsob transformace entropie do podoby náhodného čísla. Podle standardu NIST 800-90B [33] je vhodné využívat kryptografických hashovacích algoritmů, nicméně tento standard vznikl za účelem aplikace v kryptografii. Pro jiné aplikace nemusí být příliš vhodný. Tyto generátory je možné rozdělit na skutečné generátory náhodných čísel a skutečné generátory náhodných bitových sekvencí. Z pohledu realizace je skutečný generátor náhodných bitových sekvencí podstatně jednodušší na realizaci a nevyžaduje robustní systém a rozlišovací schopnost. Což je způsobeno tím, že pouze deviace ve sledované veličině, nemusí být natolik velké, aby bylo vhodné je jako náhodná čísla interpretovat přímo.

3.3.1 Příklady skutečných generátorů náhodných čísel

Následující seznam obsahuje několik skutečných generátorů náhodných čísel, zahrnující navržené metody, nebo již vyráběné hardwarové řešení:

- Sledování rozdílů v latenci paměti [45],
- Využití I/Q modulátoru na optických kabelech [42]
- Quantis QRNG [46].

První metoda využívá latenci v paměti, přesněji latenci v feroelektrické paměti s náhodným přístupem (krátce FRAM). Autoři v článku [45] uvádí, že využívá rozdílů v latenci feroelektrických kondenzátorů nacházejících se uvnitř paměťové buňky. Podle provedených testů, vygenerovaná čísla byla náhodná, bez nutnosti postprocessingu, za normálních teplot. Když se ale čip zahřál, bylo nutné generované bitové sekvence předat do hashovacího algoritmu SHA-256, aby sekvence byla náhodná.

Další možnou metodou je využití I/Q modulátoru pro optické kabely, jak se uvádí v článku vědců z Rakouského technologického institutu [42]. Představený generátor náhodných čísel aplikuje již existující technologie, jako tomu bylo v předchozím příkladu, ale využívá ji jiným způsobem, než bylo zamýšleno. Autoři totiž sledují fluktuace ve svazku fotonů pro generování skutečně náhodných čísel.

Poslední příklad je komerční řešení od firmy ID Quantique [46]. Poskytované řešení je připojitelné do PCIe slotu v počítači a stejně jako v předchozím případě využívá fluktuací světelného paprsku. Jedná se tedy o uzavřený systém. Zjednodušeně lze říct, že představuje kombinaci CMOS senzoru a LED diody. Sesbíraná entropie je následně zprocesována skrze hashovací algoritmus, jak doporučuje NIST SP 800-90B [33], nebo může být zpřístupněna přímo.

3.4 Využití generátorů náhodných čísel

Generátory náhodných čísel lze využít v celé řadě oborů, nejznámější využití je ale pravděpodobně v kryptografii. Další možností je využívat náhodná čísla v simulacích, případně v designu.

3.4.1 Využití v kryptografii

V kryptografii je možné náhodné generátory čísel využít primárně v případě šifrovací klíče, když není potřeba například podmínka prvočíselnosti, jak požaduje třeba šifra RSA[47], ale i zde najde své uplatnění. U symetrických šifer jako třeba AES [3], náhodné číslo může sloužit jako inicializační vektor, který zajišťuje rozličný výstup, i když jsou jednotlivé bloky identické.

3.4.2 Využití v simulacích

Simulace využívají náhodná čísla primárně pro přiblížení se k realitě, u některých proměnných. Existuje také disciplína stochastických simulací, které pracují exkluzivně s náhodně generovanými čísly. Další méně jednoznačný příklad využití v simulacích, představují online kasina. Tato kasina simulují reálnou hru skrze náhodný generátor čísel.

3.4.3 Využití v designu

Náhodná čísla využívaná v designu, mají dvojí vlastnosti. První nabízí možnost, vytvořit rychle a jednoduše barevné pozadí. Druhá vlastnost je daleko zajímavější, jelikož pomocí náhodných čísel, je možné generovat struktury připomínající skutečné prostředí. Mezi těmito strukturami můžou být pouze stromy nebo celé geografické prostředí.

Obrázek 3.1 Procedurálně generované prostředí [48]

Na Obrázek 3.1 je zobrazeno prostředí, ze hry No Man's Sky od studia Hello Games [49]. Tato hra využívá procedurální generování pro celý svět, od zvuků, přes prostředí, po bytosti. Procedurální generování využívá právě náhodných čísel, jako součásti pro změny ve výstupech. Hra tím dosáhla snad největší unikátnosti v historii herního průmyslu [48].

3.5 Testování náhodných generátorů náhodných čísel

Aktuálně neexistuje exaktní způsob, jak označit číslo za skutečně náhodné, proto se ve většině případů používá statistických testů, které následně vedou k vyhodnocení, zda vygenerované číslo splňuje předem dané požadavky. Protože se jedná o statistické testy, je nutné je provést na velkém množství vygenerovaných hodnot a dosáhnout v průměru požadované hodnoty. Mezi tyto testy patří například:

- Frekvenční analýza,
- Výpočet entropie,
- Diehard testy [50],
- Soubor testů z NIST SP 800-22 rev.1 [2],
- Soubor testů z NIST SP 800-90B [33],
- Soubor testů z dokumentů BSI [35], [36].

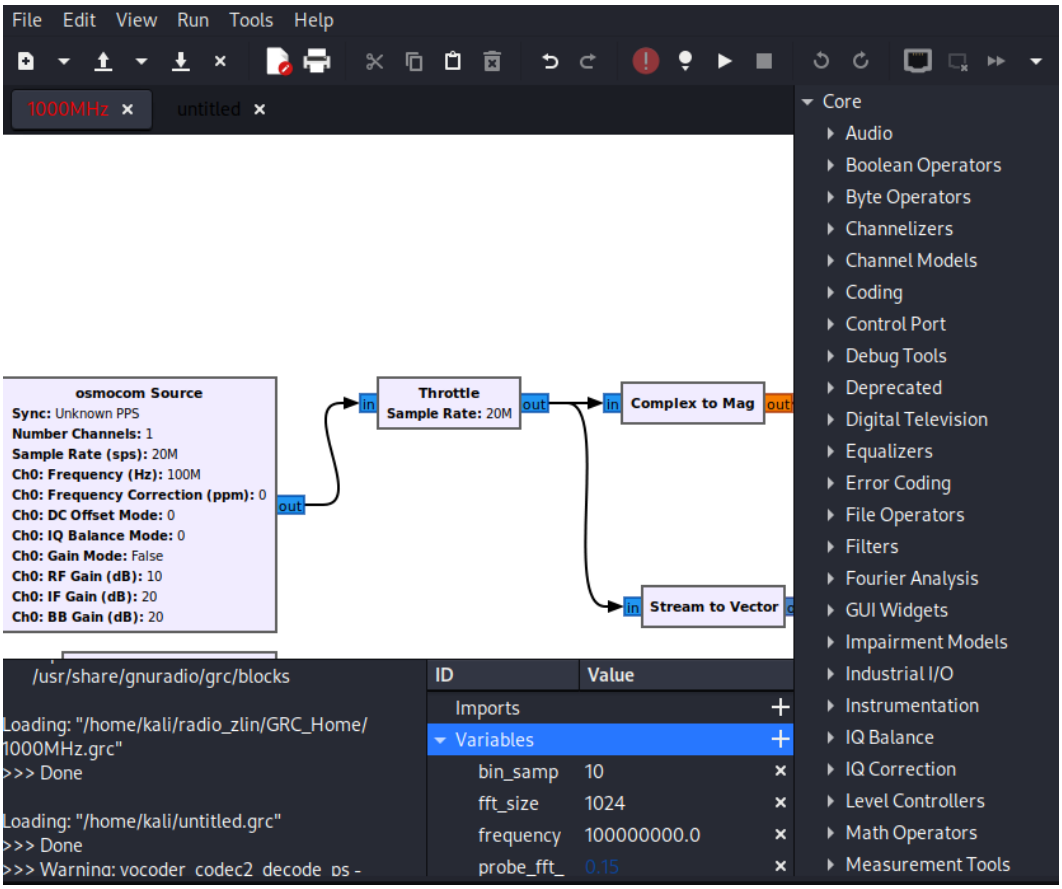
Mezi uvedenými soubory testů je dominantní zejména test monobit. Jedná se o test, který koresponduje s entropií, s tím rozdílem, že uvádí poměr mezi počtem logických

jedniček a nul v sekvenci, přičemž ideální je hodnota 0,5. Oproti tomu entropie uvádí jako ideální hodnotu 1 a při deviaci neuvádí, kterým směrem se poměr naklání. [50]

II. PRAKTICKÁ ČÁST

4 GNU RADIO COMPANION

GNU Radio Companion [51] je dominantně linuxová aplikace, využívaná pro ovládání celé řady softwarově definovaných rádií. Výhoda spočívá v možnosti exportovat vytvořený blokový model do jazyka Python, nebo C++, kde si uživatel si může kód upravit, případně doplnit.



Obrázek 4.1 Uživatelské prostředí

V blokovém schématu jsou specifikovány proměnné, parametry a sondy. Proměnné slouží k nastavení hodnoty přímo v aplikaci, zatímco parametr lze měnit při každém spuštění vygenerovaného kódu skrze linuxový terminál. Sondy jsou funkce sledující vstupní hodnotu s danou vzorkovací frekvencí.

4.1 Zjednodušená instalace

V úvodu kapitoly bylo zmíněno, že aplikace GNU Radio Companion je primárně určená pro použití v linuxovém prostředí, i přes to je možné plně aplikaci i její funkce využívat na operačním systému Windows skrze linuxový emulátor. V práci bylo použito distribuční prostředí Anaconda [52]. Balíčkovací software umožňující vytvářet separované prostředí pro programovací jazyky Python nebo R. Toto umožňuje zamezit problémům

se vzájemnou kompatibilitou jednotlivých knihoven a zároveň prostředí sdílet mezi uživateli pomocí YAML souborů.

V elektronických přílohách se nachází soubor enviroment.yml, který stačí pomocí jednoho příkazu importovat do počítače a začít využívat všechny balíčky a knihovny v přesně specifikovaných verzích využívat. [52]

```
conda env create -f environment.yml
```

Obrázek 4.2 Příkaz pro importování prostředí

Prostředí se poté aktivuje opět příkazem, což umožní pracovat s knihovnami, případně instalovat nové a zároveň ovlivnit pouze spravované prostředí. Jedná se tedy o poloviční sandboxovou aplikaci. Nutno však podotknout, že prostředí má stále přístup k souborům v počítači, tedy je nutné stále dbát na zásady práce s cizími kódy a nespouštět, nebo neinstalovat z neověřených zdrojů. [52]

```
conda activate gnuradio
```

Obrázek 4.3 Příkaz aktivaci prostředí

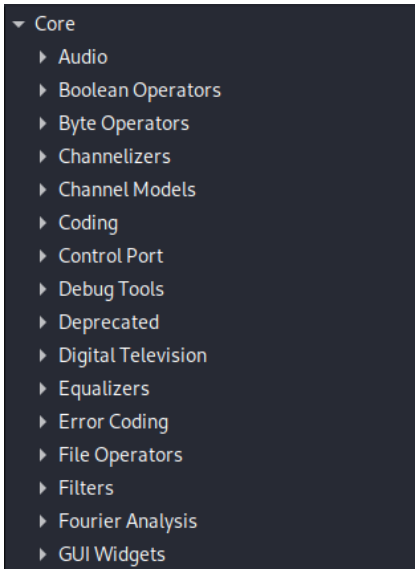
4.2 Zásady při tvorbě programu

Blokové schéma musí vždy začínat zdrojovým blokem, tedy zdrojem signálu. V tomto případě blokem Osmocom, který obsahuje všechny potřebné komponenty pro nastavení rádia, včetně knihoven pro přenos informací po USB. Výstupem bloku je komplexní číslo, které obsahuje reálnou a imaginární složku I/Q signálu. Toto komplexní číslo může být použito přímo, nebo jako v tomto případě, posunuto do bloku měnící datový typ. Například horní blok za zdrojem, ilustrovaný na Obrázek 5.1, mění komplexní číslo na formát s plovoucí nulou, který reprezentuje velikost signálu. Nejnižší hodnota, která tedy nastane, je 0. Nakonec musí být umístěný výstupní blok, což může být grafické prostředí, sonda, nebo ukládání do souboru. Při ukládání do souboru, je nutné mít na paměti, že nedochází ke kompresi dat a každý vzorek obsahuje 32 až 64 bitů. To se může značně ovlivnit volnou kapacitu disku.

4.3 Jednotlivé bloky

Na pravé straně aplikace jsou dostupné bloky. Mezi základními bloky lze nalézt například boolovské operátory, zdroje signálu, filtry, nástroje pro převzorkování signálu, a jiné. Dále jsou dostupné i specializované bloky, které dovolují například dekódovat

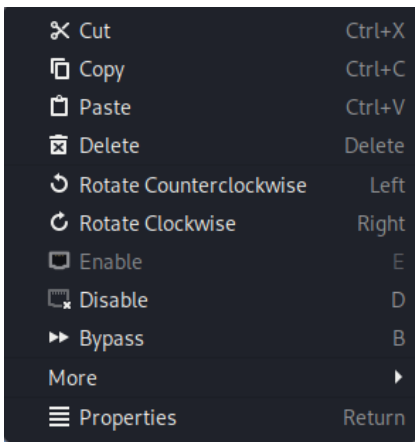
Wi-Fi pakety, DVB dekodéry a kodéry, nebo bloky pro modulaci a demodulaci signálu.



Obrázek 4.4 Postranní panel s bloky

V práci byly použité bloky pro konverzi datových typů, proměnné, parametry, výpočet průměrné hodnoty, FFT, dekadický logaritmus, zdroj signálu z SDR Osmocom a sondy pro zjištění hodnoty signálu v reálném čase.

Každý blok sdílí identické kontextové menu, obsahující funkce jako je natočení bloku, odstranění, vypnutí, nebo bypass, který způsobí, že program bude blok ignorovat a pokračovat dále. Nejdůležitější fází je nastavení bloku. Každý blok obsahuje vlastní set nastavení, doplněný záložkou s dokumentací. Ta obsahuje buď popis nastavení, nebo odkaz na online dokumentaci, spolu s příklady využití bloku a doporučené nastavení.



Obrázek 4.5 Kontextové menu

4.4 Horní lišta

Nástrojová lišta představuje další důležitou část programu, protože obsahuje prostředky nutné pro vygenerování kódu, spuštění, uložení, nebo otevření. Mimoto obsahuje také možnosti pro zpřístupnění chybového logu, vytvoření diagramu a vyhledání bloku.

Obrázek 4.6 Nástrojová lišta

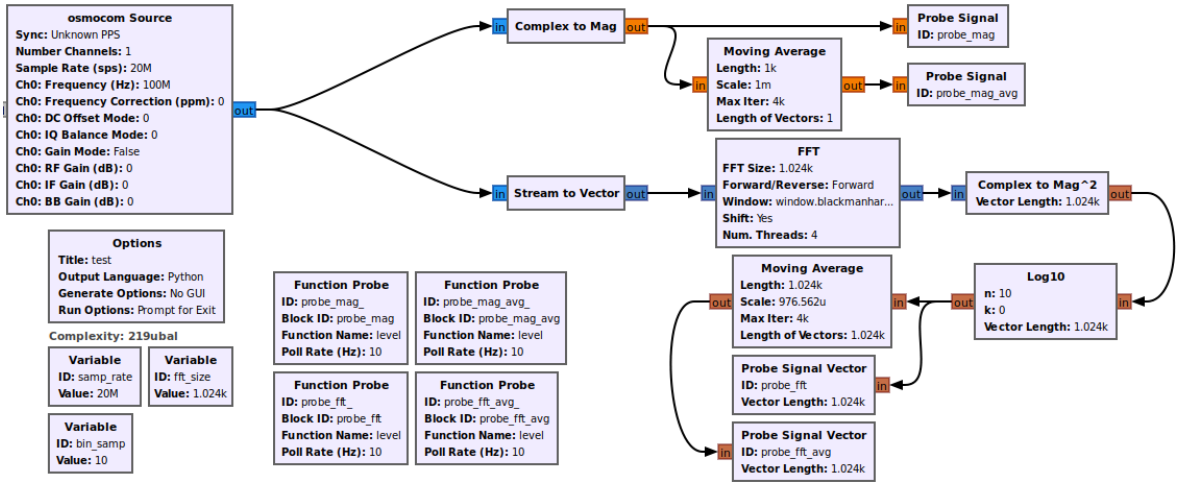
Na Obrázek 4.6 jsou barevně zvýrazněny nejdůležitější funkce. Růžově znázorněné jsou funkce pro vytvoření, otevření, nebo zavření schématu. Fialovou barvu má tlačítko pro vygenerování Python kódu, který se uloží na stejné místo jako schéma a sdílí společné jméno. Zeleně jsou znázorněny funkce spuštění a vypnutí programu, aniž by ovlivnily Python kód. Také nabízí možnost separátního spouštění, což funguje jako pojistka pro případy, kdy zdroj signálu vyžaduje restart konzole. Nástroje označené žlutou barvou, zapínají a vypínají blok, což je užitečné zejména v případě, že tvůrce chce nejdříve využít grafického prostředí pro nahlédnutí na přijímaný signál, před tím, než ho začne ukládat do souboru, ale zároveň nechce mazat, nebo přidávat další bloky.

5 NÁVRH ZPŮSOBU PRO SBĚR ELEKTROMAGNETICKÉHO ŠUMU

Návrh sběru entropie z EM šumu je založený na předpokladu, že EM záření lze snímat ve frekvenční a časové doméně. Ke sběru entropie se využívá zmíněné SDR HackRF One. Ovládání je realizováno kombinací aplikace GNU Radio Companion a programovacího jazyka Python.

5.1 Získávání okolního elektromagnetického šumu

Pro zachycení signálu bylo vytvořené blokové schéma, které neobsahuje žádné bloky s grafickým prostředním a využívá výše zmíněný zdroj Osmocom pro propojení s rádiem. Další operace jsou vyobrazeny na Obrázek 5.1.



Obrázek 5.1 Navržené blokové schéma

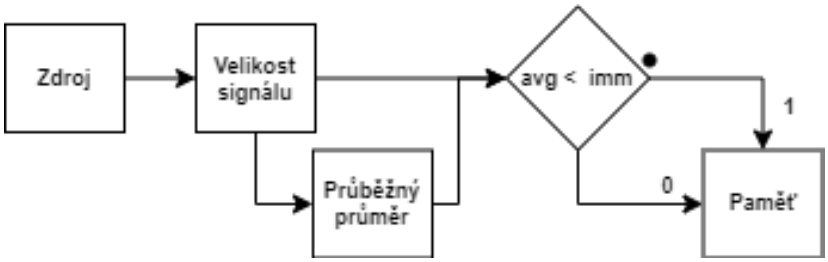
Teleskopická anténa ANT500 [20], skládající se z šesti segmentů, je připojena k SDR pomocí SMA-F konektoru. Její nejmenší velikost je 20 cm a největší 88 cm. Tabulka 5.1 popisuje jednotlivé délky antény podle počtu segmentů. Přičemž délka závisí na sledované frekvenci.

Tabulka 5.1 Závislost segmentů antény na velikosti.

Počet segmentů	Délka antény [cm]
1	20
2	33,6
3	47,2
4	60,8
5	74,4
6	88

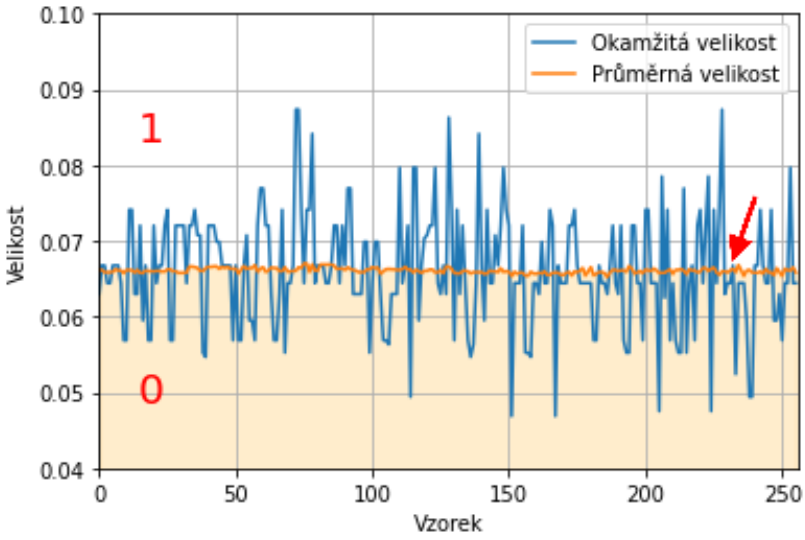
5.2 Zpracování zachycených dat

Při zachytávání bitové sekvence v časové doméně se používá jednoduchého rozhodovacího bloku, který srovnává okamžitou velikost signálu vůči průměrné velikosti signálu. Jestliže je průměrná hodnota vyšší zapíše se do paměti logická nula, naopak nižší hodnota vede k zápisu logické jedničky.



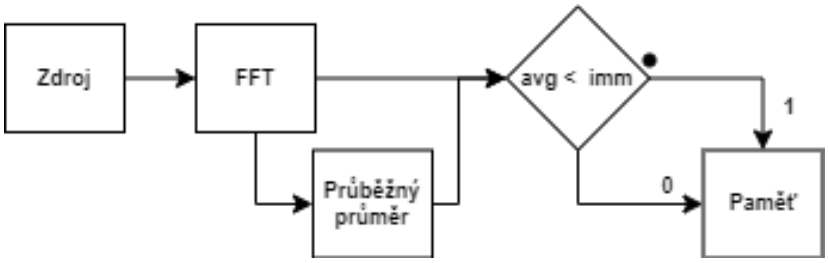
Obrázek 5.2 Transformace entropie v časové doméně

Obrázek 5.3 ilustruje časový průběh zachyceného signálu. Signál neosciluje pravidelně a šipkový ukazatel poukazuje na téměř nepozorovaný přesah průměrné hodnoty, těsně před tím, než se průměr zvedne. Tato situace je však vyhodnocena a zapsána jako logická jedna.



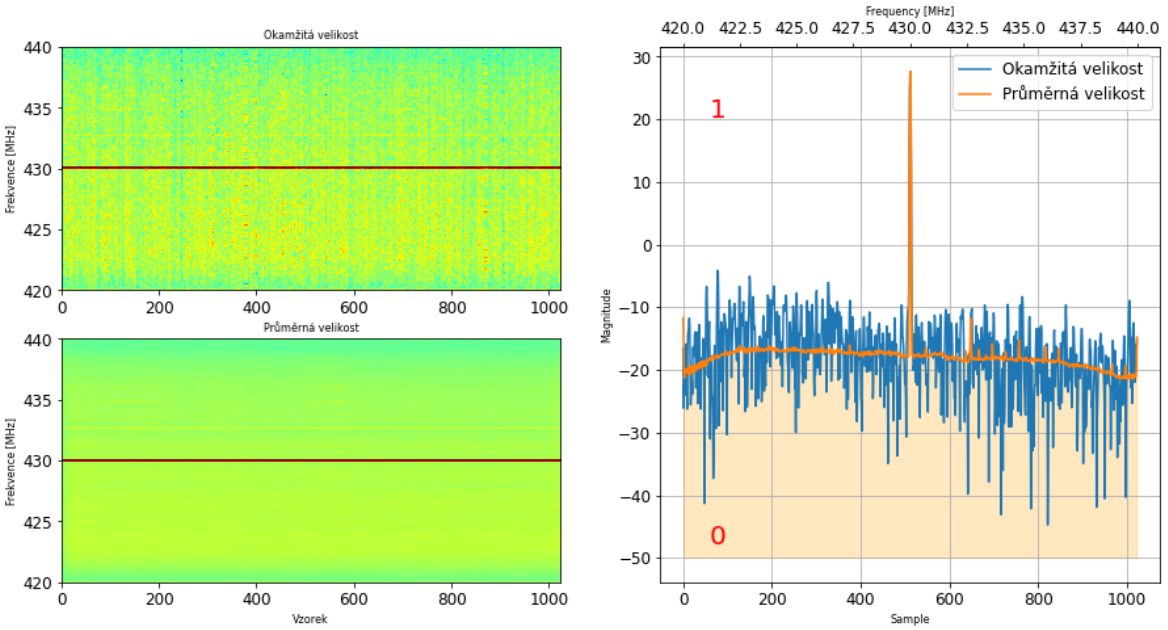
Obrázek 5.3 Transformace entropie v časové doméně

Další zkoumanou variantou, je využití signálu ve frekvenční doméně. Pro rozložení signálu s komplexní schránou na jeho frekvenční složky je využito FFT. Výsledek bude mít opět průměrný výstup a okamžitý výstup, jako u časové domény a stejným způsobem funguje tvorba čísla. V tomto případě, má však okno FFT určitou délku, náhodné číslo je tedy generováno v blocích.



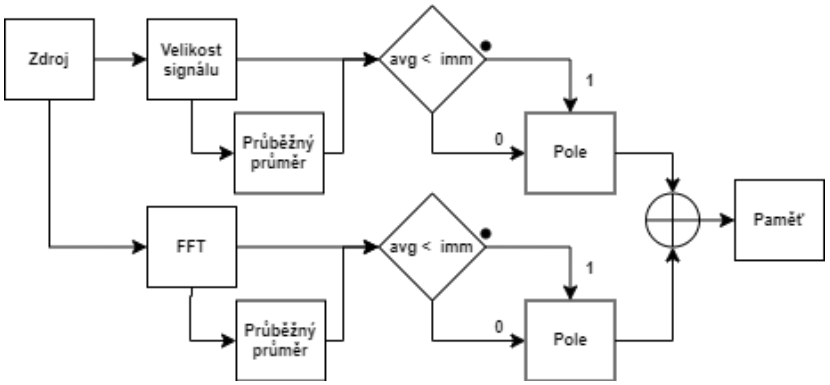
Obrázek 5.4 Transformace entropie ve frekvenční doméně

Vizualizace uvedená na Obrázek 5.5 znázorňuje okamžitou a průměrnou hodnotu ve frekvenčním pásmu 420 MHz až 440 MHz. Výrazný vrchol viditelný uprostřed zkoumaných průběhů, je tzv. „stejnosměrný vrchol“, způsobený SDR. Tento jev je způsobený posunem frekvence do základního pásma.



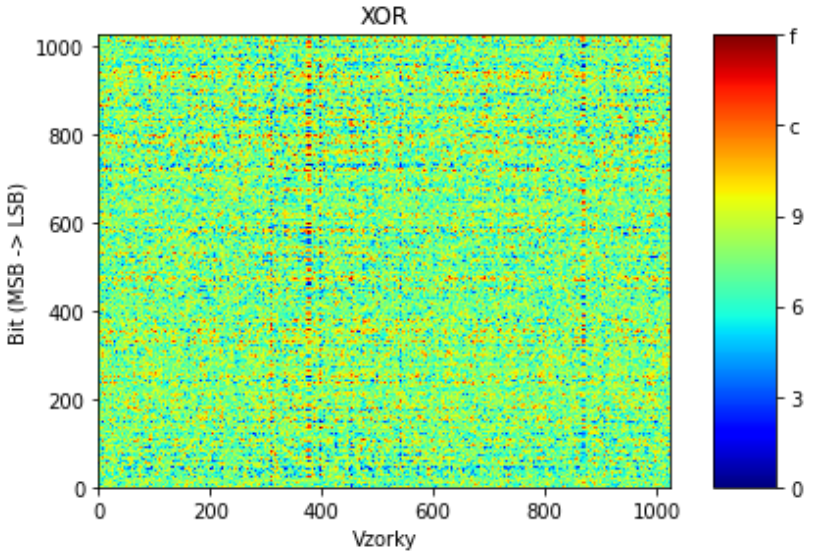
Obrázek 5.5 Signál ve frekvenční doméně

Poslední variantou je kombinace předcházejících skrze logickou operaci exkluzivního součtu, jak ilustruje Obrázek 5.6.



Obrázek 5.6 Transformace entropie kombinací předchozích

Téměř stejnobarevný panel na Obrázek 5.7 je způsobený nedostatečným rozlišením. Rovnoměrný počet nul a jedniček na daný pixel reprezentuje zelená barva, naopak deviace jedním nebo druhým směrem se mění k červené nebo modré barvě.



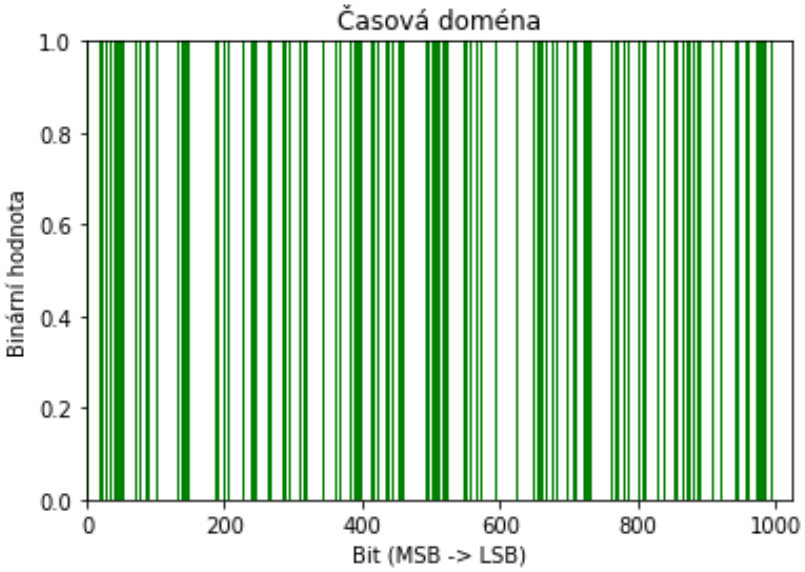
Obrázek 5.7 Výstup využitím exkluzivního součtu

5.3 Generování náhodných čísel

Generování náhodných čísel je vyřešeno uvnitř programu ovládajícího rádio, pomocí slovníkové proměnné, ve které se nachází čtyři proměnné typu pole, do nichž se průběžně ukládají zachycené vzorky. V momentně, kdy je nasbíraný dostatek dat pro vytvoření čísla, jsou vygenerované hodnoty zpracovány a je sestavena potencionálně náhodná binární sekvence.

5.3.1 Využitím časové domény

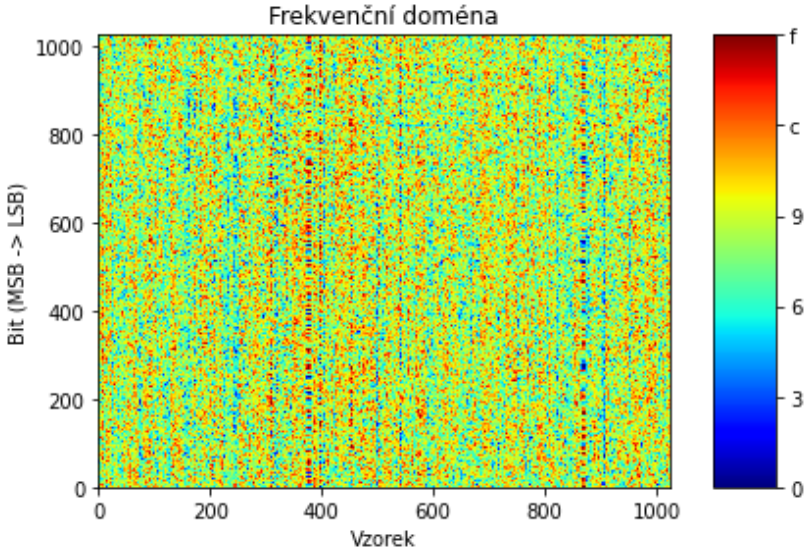
Při vizualizaci binární sekvence získané z časové domény, je možné využít obyčejný graf, který vytvoří strukturu podobnou čárovému kódu. Na Obrázek5.8 pozorované žádné opakující se trendy.



Obrázek 5.8 Vizualizace čísla v časové doméně

5.3.2 Využitím frekvenční domény

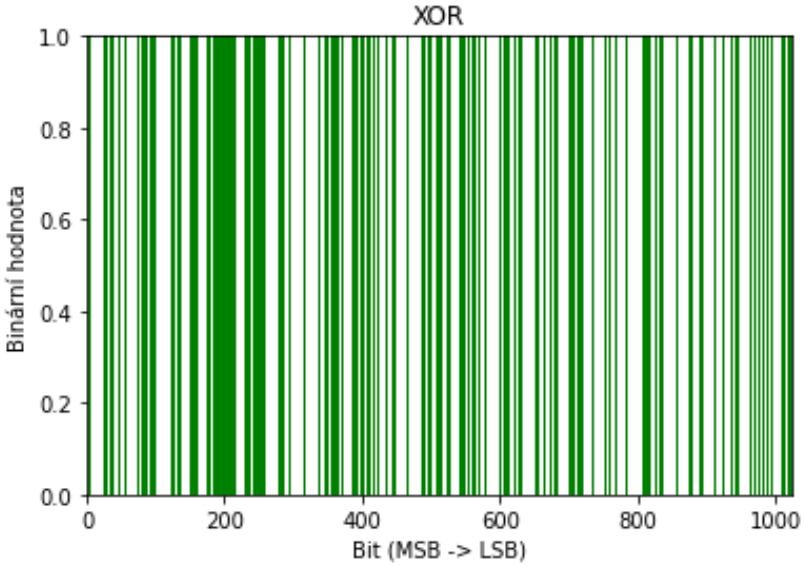
Binární sekvence získané z frekvenční domény jsou zobrazeny jako 2D panel obsahující jedno náhodné číslo za jeden krok FFT. Jinými slovy, transformované okno je identické tomu u časové domény. Výsledky ilustrované v Obrázek 5.9 odhalují, na základě četnosti červených a modrých pixelů, dlouhé sekvence jedniček a nul. Analýza výsledků však neodhalila známky periodického opakování číslic.



Obrázek 5.9 Vizualizace čísla ve frekvenční doméně

5.3.3 Využitím exkluzivního součtu

Čárový graf nenaznačuje periodicitu binární sekvence. Okolí dvoustého nejdůležitějšího bitu odhaluje delší sekvence jedniček, které se na jiných místech vyskytují.



Obrázek 5.10 Vizualizace čísla exkluzivního součtu

5.4 Výběr vhodné metody generování

Základní test zvaný monobit, ze standardu NIST 800-22a[2] pro testování generátoru náhodných čísel, byl zvolený jako metoda ověřování generovaných dat. Test analyzuje výskyt jedniček a nul v sekvenci a vrací jejich poměr v podobě zlomku. Podle zmíněného

standardu se za náhodnou sekvenci pokládá jakékoliv číslo s výsledkem rovným nebo vyšším než 0,01.

Tabulka 5.2 Výsledky monobit testu

	Monobit [%]
Časová doména	2,6
Frekvenční doména	7,8
Exkluzivní součet	93,5

Jako nejvhodnější varianta se jeví exkluzivní součet, který vykázal 93,5% testovaných čísel jako úspěšných. Průměrná hodnota přitom byla 0,4, tedy blízko k ideálu 0,5.

6 OVĚŘENÍ NÁHODNOSTI

Pro ověření náhodnosti byly sledovány tři základní charakteristiky generovaných čísel:

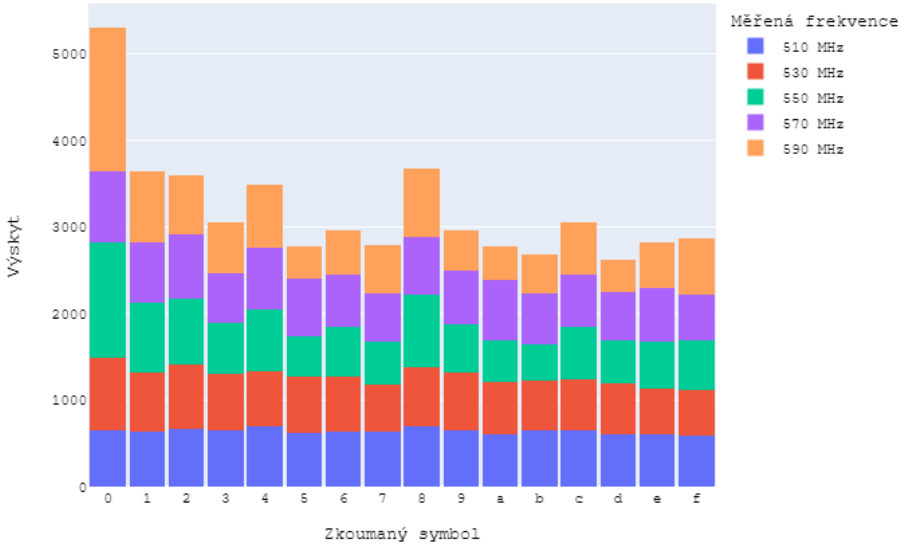
- frekvence výskytu symbolů,
- průměrná hodnota symbolů závislá na pozici v čísle,
- entropie čísla.

6.1 Frekvenční analýza

Frekvenční analýza sleduje výskyt znaků nezávisle na jejich pozici v analyzovaném vzorku. Pro větší čitelnost bylo náhodné číslo převedeno do hexadecimální soustavy a rozděleno po jednotlivých symbolech.

6.1.1 Podle frekvence

Sledované frekvence se pohybovaly mezi 100 MHz až 1 GHz s šířkou pásma 20 MHz, tedy 45 měření. Kvůli přehlednosti je ilustrovaný pouze jeden graf s pozitivními i negativními výsledky. Legenda vždy zobrazuje středovou frekvenci naladěnou při měření.



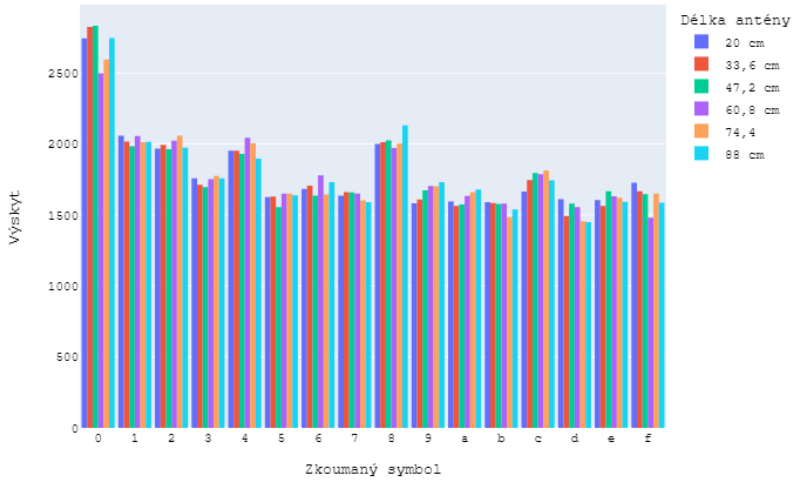
Obrázek 6.1 Frekvenční analýza 500 MHz až 600 MHz

Obrázek 6.1 indikuje, že že na frekvencích 510, 530 a 570 MHz je přibližně rovnoměrné rozložení symbolů, zatímco na frekvencích 550 a 590 MHz mírně dominuje symbol nuly. Výsledky ukazují, že s narůstající frekvencí, bez ohledu na délku antény, dochází k nárůstu výskytu nul. To je s největší pravděpodobností způsobeno vypnutým

vnitřním zesilovačem SDR a zkoumanými frekvencemi, které se nachází mimo ideální kmitočet transcieveru.

6.1.2 Podle délky antény

Při zjišťování vlivu délky antény na rozložení znaků byla zanedbána frekvence nastavená při měření. Délka antény se měnila vždy po jednom segmentu pro jednodušší kvantifikování a minimalizování možnosti chyby.



Obrázek 6.2 Frekvenční analýza podle délky antény

Výsledky měření ukazují, že optimální velikost antény, bez nutnosti manuálního nastavení, je 60,8 cm. Tedy mít vysunuty čtyři segmenty.

6.2 Průměrná hodnota na pozici symbolu

Průměrná hodnota určuje, zda některá pozice v generovaném čísle nemá tendenci konvergovat k jiné hodnotě, než ostatní. Tím by došlo ovlivnění náhodnosti čísla. Obrázek 6.3 zobrazuje pouze vybrané pozice, nicméně všechny zkoumané pozice mají stejný průběh a konvergují k číslu 7 s deviací $\pm 0,2$.

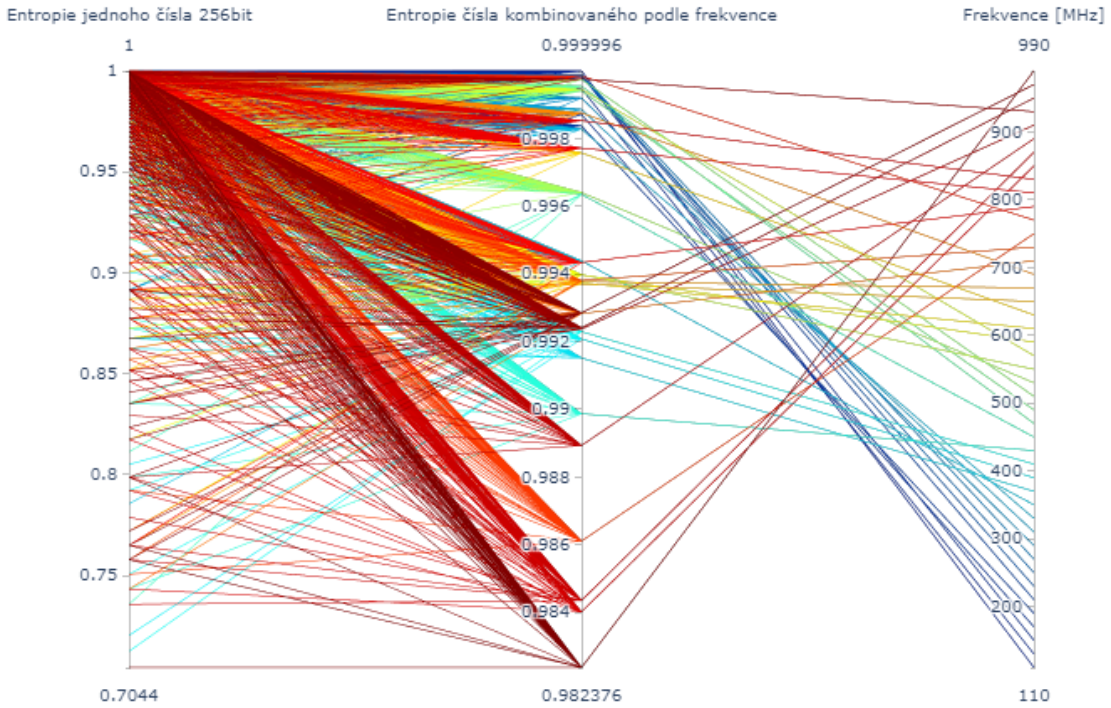
Obrázek 6.3 Konvergence průměrné hodnoty

6.3 Entropie čísel

Entropie čísel popisuje poměr bitů v daném čísle, podobně jako monobit test, pouze s tím rozdílem, že v ideálním případě je průměrná entropie čísla rovna 1. Tento test počítal entropii všem generovaným číslům a rozřazoval je podle frekvence a délky antény.

6.3.1 Podle frekvence

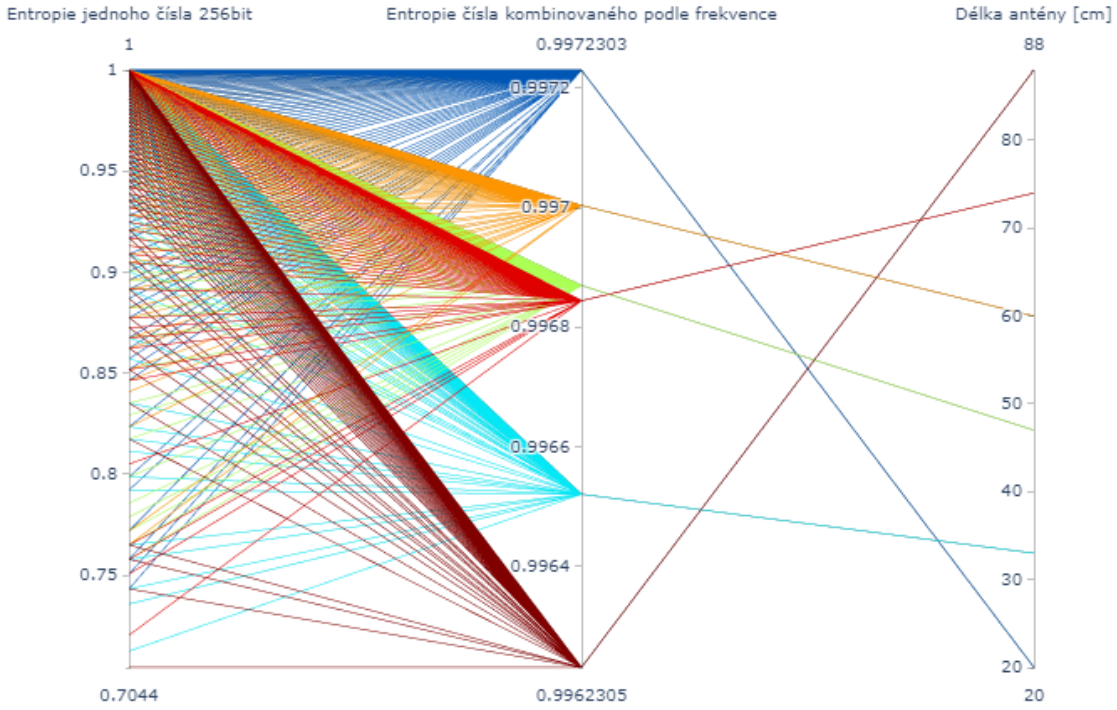
Obrázek 6.4 uvádí frekvence, jež nejsou příliš vhodné pro generování náhodných čísel zvolenou metodou. Zatímco nižší frekvence do 500 MHz mají tendenci vykazovat vyšší entropii, frekvence vyšší častěji vykazují entropii nižší. Ovšem i přes tyto výsledky zůstává nejnižší entropie jednoho čísla rovna 0,7.



Obrázek 6.4 Entropie čísla podle frekvence

6.3.2 Podle délky antény

Zkoumáním vlivu délky antény bylo zjištěno, že největší rozptyl hodnot entropie vykazuje anténa při délce 33,6 cm a 88 cm. Naopak nejmenší rozptyl při délce 60,8 cm, což potvrzuje poznatky získané frekvenční analýzou.



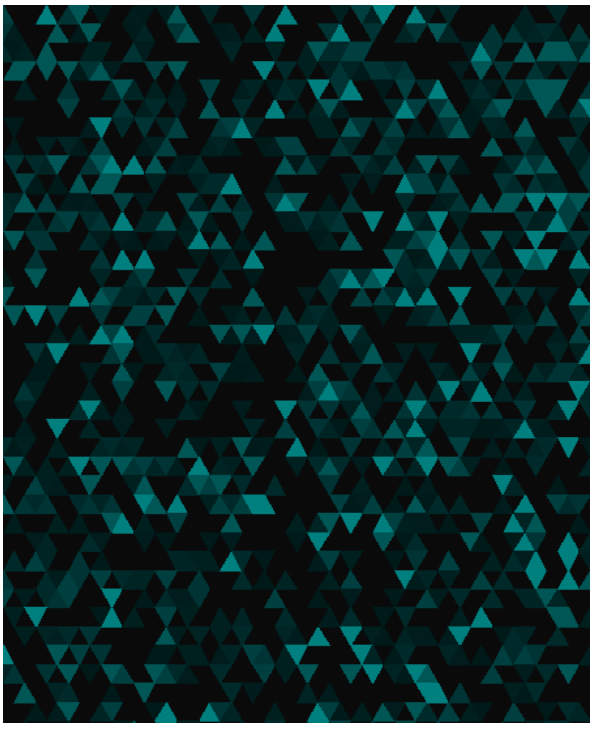
Obrázek 6.5 Entropie čísla podle délky antény

7 VYUŽITELNOST V PRAXI

Jak bylo zmíněno v teoretické části práce, náhodné generátory mají širší využití, než jen v kybernetické bezpečnosti. Součástí využití navržené metody jsou dva příklady, oba použité v aplikaci, která je výstupem práce. První příklad je pozadí grafického prostředí aplikace a druhý je implementace generátoru náhodných čísel do kryptografického systému.

7.1 Využití v designu

Aby aplikace neměla monotónní pozadí, je vhodné vložit alespoň dvoubarevný podklad. Protože autor nevyniká grafickým vnímáním, využil popsaného náhodného generátoru k vytvoření náhodného a originálního pozadí.



Obrázek 7.1 Pozadí vygenerované náhodným generátorem

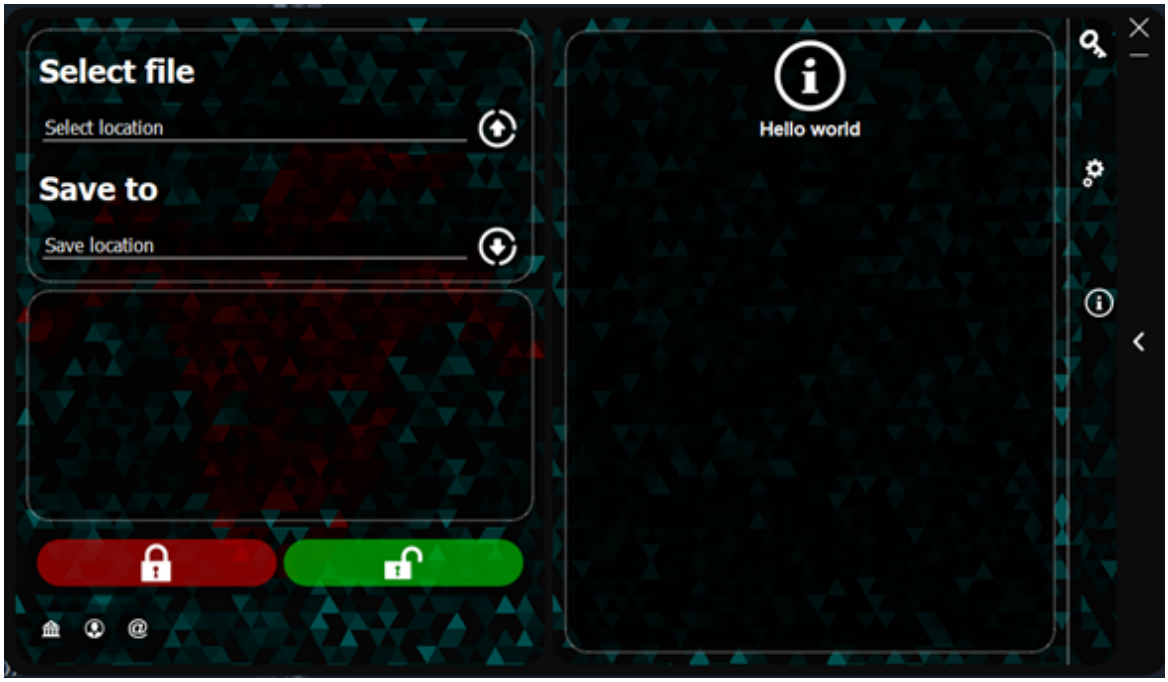
Výsledné pozadí, ilustrované na Obrázek 7.1, je reprezentováno tyrkysovými dlaždicemi na černém podkladu. Náhodné rozdělení alfa kanálu, tedy části RGBA kódu, určujícího průhlednost, zajistilo eliminaci jednolitého tyrkysového pozadí. Náhodné rozdělení průhlednosti odstraňuje symetričnost a monotónnost, čímž zpříjemňuje pohled na výsledné uživatelské prostředí, což ilustruje Obrázek 7.2.

7.2 Implementace v kryptografickém systému

V závěru práce byla navržená metoda, implementovaná do uživatelsky přívětivé aplikace, umožňující generovat klíč a zároveň s ním šifrovat soubor. Mimoto také importovat sdílený klíč, sloužící dešifrování. Aplikace byla napsaná v jazyce Python, s využitím PyQt5 knihoven pro inicializaci grafického prostředí a knihoven PyCrypto pro přístup k AES-256 [3].

7.2.1 Uživatelský prostředí

Uživatelské prostředí se skládá ze dvou hlavních částí. Levá strana, ilustrovaná v Obrázek 7.2, obsahuje výběr souboru, místa uložení a konzoli. Do té jsou vypisovány důležité informace, jako vybraná cesta, stav procesu šifrování/dešifrování, nebo vygenerování klíče. Pod konzolí se nachází dvojice tlačítek, barevně i symbolicky značící šifrování a dešifrování. Poslední funkce, v levé straně aplikace, jsou otevíratelné informace o autorovi, instituci a kontaktní email. Pravá strana je složena ze záložek, týkající se informací o ovládání, či licenci.



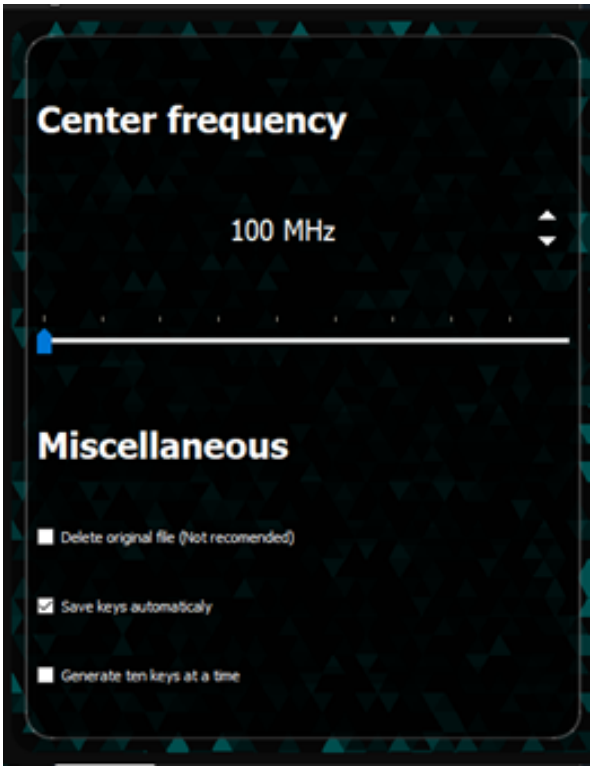
Obrázek 7.2 Uživatelské prostředí aplikace

Další záložka spravuje klíče, jak ilustruje Obrázek 7.3. Tato záložka je jednoduchá, protože horní část obsahuje seznam načtených klíčů, které uživatel vybírá kliknutím a do konzole se vypíše aktuálně vybraný klíč. Pod uvedeným seznamem se nachází fialové tlačítko, které spouští část programu zajišťující generování klíče. Pod tímto tlačítkem je už jen možnost ručně zadat klíč, importovat soubor s klíči a samozřejmě exportovat vygenerované klíče. Poslední záložka je nastavení aplikace, jak vyobrazuje Obrázek 7.4.

V této záložce je možné nastavit frekvenci, ze které bude generováno náhodné číslo, se základní hodnotou nastavenou na 100 MHz, jež se projevila jako jedna z vhodných frekvencí. Uživateli je však dovoleno využít jakoukoliv frekvenci v rozsahu 100 MHz až 990 MHz. Pod tímto nastavením jsou dodatečné funkce, například vymazání originálního souboru, což není doporučeno, proto je defaultně vypnuto. Vymazání souboru by mohlo vést k množství problémů, pokud při šifrování, nebo dešifrování dojde k chybě a bylo by nutné operaci opakovat. Další operací je automatické uložení klíče, aby se předešlo nedopatřením způsobeným zapomenutým uložením. Tato možnost je v základu zapnuta. Poslední možností je generovat deset klíčů najednou, což je časově náročná operace, ale užitečná v případě šifrování většího množství souborů vlastním klíčem

Obrázek 7.3 Záložka pro management klíčů

Poslední záložka je nastavení aplikace, jak vyobrazuje Obrázek 7.4. V této záložce je možné nastavit frekvenci, ze které bude generováno náhodné číslo, se základní hodnotou nastavenou na 100 MHz, jež se projevila jako jedna z vhodných frekvencí. Uživateli je však dovoleno využít jakoukoliv frekvenci v rozsahu 100 MHz až 990 MHz. Pod tímto nastavením jsou dodatečné funkce, například vymazání originálního souboru, což není doporučeno, proto je defaultně vypnuto. Vymazání souboru by mohlo vést k množství problémů, pokud při šifrování, nebo dešifrování dojde k chybě a bylo by nutné operaci opakovat. Další operací je automatické uložení klíče, aby se předešlo nedopatřením způsobeným zapomenutým uložením. Tato možnost je v základu zapnuta. Poslední možností je generovat deset klíčů najednou, což je časově náročná operace, ale užitečná v případě šifrování většího množství souborů vlastním klíčem.



Obrázek 7.4 Záložka s nastavením

7.2.2 Klíčové části hlavního zdrojového kódu

Pro zachování čitelnosti zdrojového kódu, je kód rozdělený na hlavní část, která spouští grafické prostředí nebo vykonává šifrovací funkce, management souborů a také volá druhou část zdrojového kódu. Druhá část kódu řídí SDR a transformuje získanou entropii na náhodný klíč, probraná bude v následující části.

Hlavní část zdrojového kódu se skládá ze dvou tříd. První třída, hlavní, inicializuje všechny funkce a provádí téměř všechny operace, druhá třída, vedlejší pojmenovaná Worker na Obrázek 7.9, obstarává vytvoření podprocesu a sama pracuje jako samostatné vlákno.

Mezi importované knihovny, viditelné na Obrázek 7.5, patří PyQt5 obsluhující grafické prostředí. Další důležitou knihovnou je knihovna PyCrypto zprostředkovávající šifrovací algoritmus. Zbylé knihovny se starají o multiprocessing a tvoření časových razítek.

```
from PyQt5 import QtCore, QtGui
from PyQt5.QtWidgets import QMainWindow, QApplication, QFileDialog
from PyQt5.QtCore import QObject, QThread, pyqtSignal
import sys
from datetime import datetime
from time import sleep
from user_interface import Ui_MainWindow
from Crypto.Cipher import AES
import subprocess
```

Obrázek 7.5 Importované knihovny

Pro aplikaci byla použita AES šifra v GCM módu, který nevyžaduje inicializační vektor pro svou funkci. Algoritmus si tento vektor sám bezpečně odvozuje z klíče během operací uvnitř volané funkce.

```
def encryptFile(self):
    for i in self.filenames:
        if "fey" not in i:
            with open(i, "rb") as file:
                cipher = AES.new(bytes.fromhex(self.currentKey), AES.MODE_GCM)
                ciphertext, tag = cipher.encrypt_and_digest(file.read())
                encrypted = cipher.nonce + ciphertext + tag
                Encrypted = bytes.fromhex(encrypted.hex())
                with open(i+".fen","wb") as saveF:
                    saveF.write(Encrypted)
                self.application.textEdit_console.append("Encryption succesful")
```

Obrázek 7.6 Kód pro dešifrování

Dešifrování je realizováno v módu dešifruj a ověř, kdy dochází k ověření správnosti zadaného klíče. Pokud by se dešifrovalo bez ověření, aplikace by nerozeznávala klíč a výsledný soubor by se jevil jako poškozený, ačkoliv by byl pouze dešifrovaný špatným klíčem.

```
def decryptFile(self):
    for i in self.filenames:
        if "fey" not in i:
            with open(i, "rb") as file:
                hexFile = file.read()
                cipher = AES.new(bytes.fromhex(self.currentKey), AES.MODE_GCM, hexFile[:16])
                try:
                    decrypted = cipher.decrypt_and_verify(hexFile[16:-16], hexFile[-16:])
                    with open(i[0:-4],"wb") as saveF:
                        saveF.write(decrypted)
                    self.application.textEdit_console.append("Decryption succesful")
                except ValueError:
                    self.application.textEdit_console.append("Decryption failed")
```

Obrázek 7.7 Kód pro šifrování

Spouštění rádia v hlavní třídě začíná získáním specifikované operační frekvence a následně spuštění separátního vlákna. Toto vlákno následně volá třídu, které předává parametry týkající se množství vygenerovaných čísel a bitů pro konkrétní frekvenci.

```
def radioStart(self):
    frequencySlider = float(int(self.application.spinBox_frequency.value())*10**6)
    try:
        self.thread = QThread()
        self.worker = self.Worker()
        self.worker.moveToThread(self.thread)
        if self.application.checkBox.isChecked():
            self.thread.started.connect(lambda: self.worker.run(10,frequencySlider,256))
        else:
            self.thread.started.connect(lambda: self.worker.run(1,frequencySlider,256))
        self.worker.finished.connect(self.thread.quit)
        self.worker.finished.connect(self.worker.deleteLater)
        self.thread.finished.connect(self.thread.deleteLater)
        self.thread.start()
        self.worker.random.connect(lambda x: self.application.listWidget.addItem(x))
    except:
        self.application.textEdit_console.append("Generation failed")
```

Obrázek 7.8 Spuštění rádia

Třída, volající aplikaci pro ovládání rádia, vytváří dvě proměnné. První signalizuje dokončení procesu a druhá nese náhodné číslo. Tato třída také definuje funkci run, jež spouští vlastní proces starající se o Python kód ovládající rádio. Uvedené řešení není ideální, protože z již zmíněných důvodů, je nutné nejdříve ukončit konzoli a až poté navázat opětovné spojení s rádiem. Zdroj Osmocom spoléhá na korektní ukončení uživatelem, naopak zdroj Soapy může operaci zastavení realizovat samostatně. V obou případech je ale uvedené řešení nutné. Program se také nachází v nekonečné smyčce, protože knihovna libusb, která zajišťuje komunikaci, může v případě déle trvajícího procesu ukončování, oznámit chybovou hlášku. Z toho důvodu, se v případě selhání spojení, proces uspí na nejméně dvě a půl vteřiny, aby měl program možnost procesy ukončit před opětovným pokusem o navázání kontaktu.

```
class Worker(QObject):
    finished = pyqtSignal()
    random = pyqtSignal(str)
    def run(self,rounds,freq,bins):
        i = 0
        while True:
            try:
                self.random.emit(subprocess.check_output("python hackrf.py --frequency "+str(freq)+" --bins "+str(bins), shell=True, universal_newlines=True).split("\n")
                i += 1
                if i == rounds:
                    break
            except:
                i += 1
                sleep(5*(i/2))
        self.finished.emit()
```

Obrázek 7.9 Zavolání aplikace ovládající rádio

7.2.3 Klíčové části zdrojového kódu řídící rádio

Kód ovládající SDR, byl z větší části vytvořený v aplikaci GNU Radio Companion, podle schématu vyobrazeném na Obrázek 5.1. Vygenerovaný kód obsahuje knihovny gnuradio, které vnitřně volají další knihovny, jako zmíněnou libusb, zprostředkovávající komunikaci s rádiem. Další knihovny se starají o multithreading, časování, přístup k systémovým funkcím a nebo signalizaci. Právě knihovna zajišťující multithreading je zdrojem problémů, kvůli kterým musí být aplikace volána jako separátní proces.

Zmíněné problémy spočívají v tom, že knihovna gnuradio vytváří vlákna, která korektně neukončuje, naopak je nastavuje jako daemon vlákna, čili aplikace musí být ukončena před tím, než může být znovu spuštěna. Bohužel tyto vlákna vytváří i na pozadí, uvnitř knihovny a nelze to tedy změnit pouhou změnou parametru.

```
from gnuradio import blocks
from gnuradio import fft
from gnuradio.fft import window
from gnuradio import gr
from argparse import ArgumentParser
from gnuradio.eng_arg import eng_float
from gnuradio import eng_notation
import sys
import signal
import osmosdr
import time
import threading
import functools
```

Obrázek 7.10 Knihovny pro obsluhu rádia

Důležitou součástí aplikace byla možnost vkládat parametry při volání kódu. Parametry jsou dva, jmenovitě operační frekvence a binární délka generovaného čísla.

```
def argument_parser():
    parser = ArgumentParser()
    parser.add_argument(
        "-f", "--frequency", dest="frequency", type=eng_float, default=eng_notation.num_to_str(float(100e6)),
        help="Set freq [default=%(default)r]")
    parser.add_argument(
        "-b", "--bins", dest="bins", type=intx, default=eng_notation.num_to_str(int(256)),
        help="Set binary lenght [default=%(default)r]")
    return parser
```

Obrázek 7.11 Management argumentů

Proměnné nacházející se uvnitř aplikace ovládající SDR jsou celkem tři. První je pevně stanovená vzorkovací frekvence rádia nastavená na maximum 20 Msps, aby byl zajištěna plná šířka pásma 20 MHz. Zbylé dve jsou parametry předané při volání kódu.

```
self.samp_rate = samp_rate = 20e6
self.frequency = frequency
self.fft_size = bins
```

Obrázek 7.12 Proměnné pro práci s radiem

Na Obrázek 7.13 ilustruje příklad programového řešení sond, zahrnující celkem čtyři sondy. Ukázka se zaměřuje na sondu, sledující průměrnou velikost signálu v časové doméně. Pro sondu je nastavená podmínka, zajišťující přeskočení startu rádia a uspání na přibližně 10 ms, pokud je hodnota velikosti signálu rovna nule. Druhá podmínka slouží pro zastavení zaznamenávání, pokud je v časové doméně dostatečný počet bitů pro pokročení k dalšímu kroku.

```
def _time_magnitude_avg_probe(stop):
  global time_avg
  while True:
    val = self.probe_time_avg.level()
    if val != 0:
        time_avg.append(val)
    if len(time_avg) >= bins and len(_time) >= bins and stop():
        break
    try:
      try:
        self.doc.add_next_tick_callback(functools.partial(self.set_time_magnitude_avg,val))
      except AttributeError:
        self.set_time_magnitude_avg(val)
    except AttributeError:
      pass
    time.sleep(1.0 / (100))
global stop_threads
stop_threads = False
self._time_magnitude_avg_thread = threading.Thread(target=_time_magnitude_avg_probe, args =(lambda : stop_threads, ))
self._time_magnitude_avg_thread.daemon = True
self._time_magnitude_avg_thread.start()
```

Obrázek 7.13 Příklad programového řešení sondy

Při generování náhodného čísla bylo využito všech čtyř zachycených hodnot. Nejprve dochází v nekonečné smyčce k pravidelné kontrole, zda existuje dostatečný počet dat pro vygenerování čísla. Pokud ano, pak je využito posledních záznamů ve frekvenční doméně a všech hodnot v časové doméně. Z frekvenční a časové domény jsou

vytvořeny samostatné binární sekvence, mezi kterými je následně provedený exkluzivní součet. Dalším krokem je zastavení komunikace s rádiem a převod binárního čísla na hexadecimální hodnotu, z důvodu délky při výpisu. Před vrácením hodnoty dojde k rychlé kontrole, zda je délka čísla 64 symbolů. Pokud délka nesouhlasí, jsou na začátek řetězce přidány potřebné nuly a výsledné číslo je předáno do hlavní aplikace.

```
while True:
    if len(time_avg) >= options.bins and len(_time) >= options.bins:
        sample["Time"] = _time[:options.bins]
        sample["Time average"] = time_avg[:options.bins]
        sample["Frequency"] = freq[-1]
        sample["Frequency Average"] = freq_avg[-1]
        first = ["0" if sample["Time"][i] <= sample["Time average"][i] else "1" for i in range(options.bins)]
        second = ["0" if sample["Frequency"][i] <= sample["Frequency Average"][i] else "1" for i in range(options.bins)]
        random = ["0" if first[i] == second[i] else "1" for i in range(options.bins)]
        stop_threads = True
        break
tb.stop()
tb.wait()
res = hex(int("".join(random),2))[2:]
while True:
    if len(res) != 64:
        res = "0"+res
    else:
        break
return res
```

Obrázek 7.14 Ukázka kódu generujícího náhodné číslo

ZÁVĚR

Práce pojednává o využití elektromagnetického šumu pro generování náhodných čísel. Úvod práce obsahuje rešerši z oblasti antén, zahrnující rozdělení do kategorií a subkategorií, které byly doplněny o příklady. Závěr rešerše uvádí nejvhodnější anténu pro účely této práce, již je teleskopická anténa ANT500 [20]. Tato anténa disponuje všesměrovostí, charakteristickou pro prutové antény a nastavitelností resonanční frekvence pomocí změny délky.

Rešerši následuje představení softwarového rádia HackRF One [1], které tvoří perfektní testovací nástroj, protože disponuje volně nastavitelným frekvenčním rozsahem od 30 MHz do 6 GHz o šířce pásma 2 MHz až 20 MHz. Což dovolovalo navrženou metodu optimalizovat tak, aby při případném vlastním hardwarovém řešení, nebylo potřeba využívat finančně nákladných obvodů. Detailně popsána je funkce zařízení jako přijímače a jsou uvedeny dílčí obvody, nacházející se uvnitř rádia.

Teoretická část práce je zakončena objasněním náhodných generátorů čísel. Kapitola uvádí aktuální standardy a normy, ovlivňující směr vývoje generátorů a také jejich využití v praxi. Generátory jsou rozděleny na pseudonáhodné a skutečné, a doplněny o příklady jako pseudonáhodný generátor náhodných čísel vytvořený Johnem von Neumannem [38], nebo jeho rozšíření o Weylovu sekvenci [40].

Úvod praktické části provádí instalací a seznámením se s programem GNU Radio Companion [51], který je oblíbeným linuxovým nástrojem pro vývoj aplikací řídících SDR. Po představení programu dochází k návrhu metody v této aplikaci.

Navržená metoda obsahuje zdrojový blok Osmocom pro komunikaci s rádiem HackRF One, který následuje blokové schéma, jenž se dělilo na část zachytávající časovou doménu snímaného signálu a část zachytávající frekvenční doménu. Každá část zahrnuje dvě sondy, jednu pro zachycení průměrné velikosti signálu a jednu pro průměrný signál. Způsoby transformace entropie do podoby náhodných čísel, využitelných v kryptografickém systému, jsou tři.

První metoda využívá časovou doménu. Zachycená velikost signálu v čase je kvantovaná podle průměrné hodnoty velikosti signálu, nicméně tato metoda se neprojevila jako náhodná a monobit test [2] prošel pouze ve 2,6 % případů. Druhá metoda provádí stejnou operaci ve frekvenční doméně a její výsledky jsou mírně lepší, až 7,8 %. Poslední metoda je kombinací předchozích, s použitím exkluzivního součtu. Výsledky testu odhalují 93,5 % náhodnost generovaných bitových sekvencí, přičemž průměrná hodnota testu je 0,4, což se blíží k ideálu 0,5. Metoda exkluzivního součtu nesla výhodu rozvázání informačního spojení generovaného čísla s elektromagnetickým signálem, ze kterého vzešla.

Tato metoda je vhodná pro následující testování. Testy se skládaly z frekvence výskytu symbolů, průměrné hodnoty symbolů závislých na pozici v generovaném čísle a nakonec entropie čísla. Všechny tyto testy jsou prováděny ve frekvenčním rozsahu 100 MHz až 1 GHz, rozděleném po 20 MHz pro identifikaci nevhodných frekvencí a odhalení vlivu délky antény.

Frekvenční analýza podle hexadecimální hodnoty má blokově zjistit rozložení symbolů. Tento test ukázal, že některé frekvence generují rovnoměrné rozložení a jiné mají tendenci generovat symboly blíže k nule. Délka antény neukázala žádné výraznější deviace, nicméně anténa o délce 60,8 cm generuje přibližně nejvyváženější výsledky. Výpočet průměrné hodnoty podle symbolu zjišťuje, zda je generátor pozičně nezávislý. Tento test je prováděný kvůli existenci stejnosměrné špičky ve středu frekvenční domény. Výsledky testu však odhalují, že průměrná hodnota na všech pozicích konverguje k průměrné hodnotě 7 s deviací $\pm 0,2$. Výsledky testu entropie indikují, že nejlepších výsledků dosahují nižší frekvence. Entropie však ani na vyšších frekvencích není nižší jak 0,7. Při analýze podle délky antény se ukazuje, že nejvyšší entropii má plně složená anténa a nejnižší plně rozložená. Podle předchozích výsledků frekvenční analýzy, zvláštní pozornost je zaměřena na velikost 60,8 cm. Všechny délky antény i frekvence však mají stejný trend poloviční Gaussovy funkce s vrcholem v jedničce. Není tedy nutné jakkoliv upravovat program nebo rozsahy.

Závěr praktické části představuje dvě využití návrhu. Prvním využitím je vytvoření pozadí aplikace, pro eliminaci monotónnosti designu. Druhé využití je pro generování šifrovacích klíčů v algoritmu AES-256 v GCM módu [3]. Aplikace obsahuje nastavení frekvence pro vygenerování klíčů, exportování a importování klíčů pro snazší management a možnost ručního zadávaní klíče. Využitá šifra je používaná v módu ověř a dešifruj, aby se mohlo předejít uživatelské chybě s výběrem špatného klíče.

Další rozšíření práce by mělo obsahovat rozšíření o testy obsažené ve standardech jako je NIST 800-22 rev.1 [2]. Další cesta kterou se tato práce bude ubírat je testování vyšších frekvencí. Poslední varianta je využití lepších a menších typů antén, což je spojené s vyššími frekvencemi a explorace možnosti využití nanoantén pro co největší miniaturizaci, samotné zařízení nemuselo být příliš velké při zmenšení frekvenčního rozsahu a předdefinovanému způsobu zpracování signálu.

Na závěr mělo být zmíněno, že výsledky této práce budou také začátkem září tohoto roku prezentovány na konferenci EMC Europe 2022 v Göteborgu ve Švédsku pod záštitou IEEE a také toto téma vyhrálo 1. místo v mezinárodní soutěži Studentské odborné a tvůrčí činnosti (STOČ) v sekci Měřící diagnostické a bezpečnostní systémy. Téma bude dále zkoumáno a rozšiřováno na doktorském studiu.

SEZNAM POUŽITÉ LITERATURY

[1] Great Scott Gadgets: HackRF One [online]. 2022, [cit. 2022-05-15]. Dostupné z: `https://greatscottgadgets.com/hackrf/one/`

[2] A. Ruhkin et al.: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [online]. 2010, [cit. 2022-04-24]. Dostupné z: `https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final`

[3] National Institute of Standards and Technology: Specification for the Advanced Encryption Standard (AES) [online]. 2001, [cit. 2022-04-24]. Dostupné z: `https://csrc.nist.gov/publications/detail/fips/197/final`

[4] Balanis, C.: *Antenna Theory: Analysis and Design*. Wiley, Čtvrté vydání, 2016, ISBN 9781118642061, [cit. 2022-05-15].

[5] Du, Z.-M.; Wong, S.-W.; Chen, R.-S.; aj.: Wide Bandwidth Ratio of 10-to-1 CPW-Fed Whip Antenna with Improved Radiation Patterns. *IEEE Transactions on Antennas and Propagation*, 2022: s. 1–1, doi:10.1109/TAP.2021.3138508, [cit. 2022-05-15].

[6] Qian, W.; Xiaoli, X.: Design of Combination Antenna of Whip Antenna and Quadrifilar Helix Antenna. In *2007 8th International Conference on Electronic Measurement and Instruments*, 2007, s. 1–669–1–672, doi:10.1109/ICEMI.2007.4350539, [cit. 2022-05-15].

[7] Nazaryan, A.: The Massive Russian Radar Site in the Chernobyl Exclusion Zone [online]. 2014, [cit. 2022-05-15]. Dostupné z: `https://www.newsweek.com/hunt-russian-woodpecker-246670`

[8] Scalise, G.; Boccia, L.; Arnieri, E.; aj.: Design of ME-dipole antennas for 5G phased array applications at 28 GHz. In *2021 15th European Conference on Antennas and Propagation (EuCAP)*, 2021, s. 1–4, doi:10.23919/EuCAP51087.2021.9411477, [cit. 2022-05-15].

[9] Kondratieva, S.; Shmachilin, P.; Gadzhiev, E.; aj.: Cylindrical AESA of microstrip dipols for the ground communication system. In *2020 International Conference Engineering and Telecommunication (En T)*, 2020, s. 1–5, doi:10.1109/EnT50437.2020.9431246, [cit. 2022-05-15].

[10] ČSN EN 61000-4-8 ED.2: Elektromagnetická kompatibilita (EMC) - Část 4-8: Zkušební a měřicí technika - Magnetické pole síťového kmitočtu - Zkouška odolnosti.

Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, ročník Třídící znak: 33 3432, říjen 2010: str. 44, [cit. 2022-05-15].

[11] Woo, T.; Kim, D.; Park, C. Y.; aj.: Compact Wideband Loop Antenna for Earbuds. *IEEE Access*, ročník 10, 2022: s. 47340–47347, doi:10.1109/ACCESS.2022.3171034, [cit. 2022-05-15].

[12] Chen, Z.; Hu, W.; Gao, Y.; aj.: Compact Wideband Circularly Polarized Loop Antenna Based on Dual Common and Differential Modes. *IEEE Antennas and Wireless Propagation Letters*, 2022: s. 1–1, doi:10.1109/LAWP.2022.3174400, [cit. 2022-05-15].

[13] Vaske, A.; Akar, A.; Neubauer, B.: Coplanar Waveguide Fed U-Band Horn Antenna Manufactured Using 3D Printing and Electroplating. In *2022 16th European Conference on Antennas and Propagation (EuCAP)*, 2022, s. 1–4, [cit. 2022-05-15].

[14] Luo, Z.; Jiang, H.; Wu, M.; aj.: Novel Multi-pattern Reconfigurable Horn Antenna Made of Pure Water. In *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2022, s. 1251–1254, doi:10.1109/EEBDA53927.2022.9744861, [cit. 2022-05-15].

[15] Ubiquiti Inc.: airFiber 60 LR [online]. 2022, [cit. 2022-05-15].
Dostupné z: `https://store.ui.com/collections/operator-airfiber/products/airfiber-60-lr`

[16] Jusoh, M. T.; Lafond, O.; Colombel, F.; aj.: Performance and Radiation Patterns of a Reconfigurable Plasma Corner-Reflector Antenna. *IEEE Antennas and Wireless Propagation Letters*, ročník 12, 2013: s. 1137–1140, doi:10.1109/LAWP.2013.2281221, [cit. 2022-05-15].

[17] Wen, Y.; Qin, P.-Y.; Wei, G.-M.; aj.: Circular Array of Endfire Yagi-Uda Monopoles with A Full 3600 Azimuthal Beam Scanning. *IEEE Transactions on Antennas and Propagation*, 2022: s. 1–1, doi:10.1109/TAP.2022.3161306, [cit. 2022-05-15].

[18] Zhang, Y.; Brown, A. K.: Octagonal Ring Antenna for a Compact Dual-Polarized Aperture Array. *IEEE Transactions on Antennas and Propagation*, ročník 59, č. 10, 2011: s. 3927–3932, doi:10.1109/TAP.2011.2163742, [cit. 2022-05-15].

[19] Liang, X.; Yuan, W.; Zhang, L.; aj.: Filtering Waveguide Slot Array Antenna for Ku-Band Applications. In *2018 IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2018, s. 456–457, doi:10.1109/APCAP.2018.8538133, [cit. 2022-05-15].

[20] Great Scott Gadgets: ANT500 [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://greatscottgadgets.com/ant500/

[21] 3DEXPERIENCE® Company - Dassault Systèmes®: CST Studio Suite 3D EM simulation and analysis software [online]. 2020, [cit. 2020-01-26].
Dostupné z: https://www.3ds.com/products-services/simulia/products/cst-studio-suite/

[22] Wyglinski, A.; Getz, R.; Collins, T.; aj.: *Software-Defined Radio for Engineers*. Artech House mobile communications series, Artech House, 2018, ISBN 9781630814595, [cit. 2022-05-15].

[23] Shannon, C.: Communication in the Presence of Noise. *Proceedings of the IRE*, ročník 37, č. 1, jan 1949: s. 10–21, doi:10.1109/jrproc.1949.232969, [cit. 2022-05-15].
Dostupné z: https://doi.org/10.1109/jrproc.1949.232969

[24] Dunne, B. E.: The What, How and Why of Complex Sampling for SDR Transceivers. In *ASEE North Central Section Conference*, 2019, s. 1–9, [cit. 2022-05-15].

[25] NXP Semiconductors: LPC4320 [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://cz.mouser.com/ProductDetail/NXP-Semiconductors/LPC4320FBD144551?qs=QGdnF3v6kLQtRsRnWVFR9w%3D%3D

[26] Xilinx: XC2C64A [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://cz.mouser.com/ProductDetail/Xilinx/XC2C64A-7VQG44I?qs=rrS6PyfT74cwFvjklnTetA%3D%3D

[27] Maxim Integrated®: MAX5864 [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://www.maximintegrated.com/en/products/analog/data-converters/analog-front-end-ics/MAX5864.html

[28] Maxim Integrated®: MAX2837 [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://www.maximintegrated.com/en/products/comms/wireless-rf/MAX2837.html

[29] Qorvo: RFFC5072 [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://cz.mouser.com/ProductDetail/Qorvo/RFFC5072?qs=p6lVfQR1GSotLfz%2F9VJJTg%3D%3D

[30] Skyworks Solutions, Inc.: Si5351C [online]. 2022, [cit. 2022-05-15].
Dostupné z: https://cz.mouser.com/ProductDetail/Skyworks-Solutions-Inc/SI5351C-B-GM?qs=p9T7GgSe1IEq510PZmN4nA%3D%3D

[31] Schindler, W.; Killmann, W.: Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, ISBN 978-3-540-36400-9, s. 431–449, [cit. 2022-05-15]. Dostupné z: `https://link.springer.com/content/pdf/10.1007/3-540-36400-5_31.pdf`

[32] Barker, E.; Kelsey, J.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators [online]. 2015, [cit. 2022-04-24]. Dostupné z: `https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final`

[33] M. S. Turan et al.: Recommendation for the Entropy Sources Used for Random Bit Generation [online]. 2018, [cit. 2022-04-24]. Dostupné z: `https://csrc.nist.gov/publications/detail/sp/800-90b/final`

[34] Barker, E.; Kelsey, J.: Recommendation for Random Bit Generator (RBG) Constructions [online]. 2016, [cit. 2022-04-24]. Dostupné z: `https://csrc.nist.gov/publications/detail/sp/800-90c/draft`

[35] Schindler, W.; Wolfgang, K.: Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators. 2002, [cit. 2022-05-15]. Dostupné z: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_Functionality_Classes_Evaluation_Methodology_DRNG_e.pdf?__blob=publicationFile&v=2`

[36] Schindler, W.; Wolfgang, K.: A proposal for : Functionality classes and evaluation methodology for true (physical) random number generators. 2002, [cit. 2022-05-15]. Dostupné z: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile&v=1`

[37] ISO/IEC 18031:2011: Information technology — Security techniques — Random bit generation. *International Organization for Standardization*, ročník ICS : 35.030, listopad 2011: str. 142, [cit. 2022-05-15].

[38] Von Neumann, J.: Various techniques used in connection with random digits. *John von Neumann, Collected Works*, ročník 5, 1963: s. 768–770, [cit. 2022-05-15].

[39] Barker, E.; Kelsey, J.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators [online]. 2015, [cit. 2022-04-24].
Dostupné z: `https://csrc.nist.gov/publications/detail/sp/800-90a/archive/2012-01-23`

[40] Widynski, B.: Middle-Square Weyl Sequence RNG. 2017, doi:10.48550/ARXIV.1704.00358, [cit. 2022-05-15].
Dostupné z: `https://arxiv.org/abs/1704.00358`

[41] Gong, L.; Zhang, J.; Liu, H.; aj.: True Random Number Generators Using Electrical Noise. *IEEE Access*, ročník 7, 2019: s. 125796–125805, doi:10.1109/ACCESS.2019.2939027.
Dostupné z: `https://ieeexplore.ieee.org/document/8822724`

[42] Vokić, N.; Milovančev, D.; Pacher, C.; aj.: True Random Number Generation in an Optical I/Q Modulator. In *2020 European Conference on Optical Communications (ECOC)*, 2020, s. 1–4, doi:10.1109/ECOC48923.2020.9333271, [cit. 2022-05-15].

[43] Maybee, B.; Hodgson, D.; Beige, A.; aj.: A Physically-Motivated Quantisation of the Electromagnetic Field on Curved Spacetimes. *Entropy*, ročník 21, č. 9, aug 2019: str. 844, doi:10.3390/e21090844, [cit. 2022-05-15].
Dostupné z: `https://doi.org/10.3390%2Fe21090844`

[44] Gamil, H.; Mehta, P.; Chielle, E.; aj.: Muon-Ra: Quantum random number generation from cosmic rays. In *2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2020, s. 1–6, doi:10.1109/IOLTS50870.2020.9159728, [cit. 2022-05-15].

[45] Rashid, M. I.; Ferdaus, F.; Talukder, B. M. S. B.; aj.: True Random Number Generation Using Latency Variations of FRAM. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, ročník 29, č. 1, 2021: s. 14–23, doi:10.1109/TVLSI.2020.3018998, [cit. 2022-05-15].

[46] ID Quantique: Quantis QRNG PCIe New Generation [online]. 2022, [cit. 2022-05-15].
Dostupné z: `https://www.idquantique.com/random-number-generation/products/quantis-qrng-pcie/`

[47] Rivest, R. L.; Shamir, A.; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, ročník 21, č. 2, 1978: s. 120–126, [cit. 2022-05-15].

[48] Cook, A.: 6 mind-blowing No Man's Sky numbers [online]. 2016, [cit. 2022-05-15]. Dostupné z: `https://www.redbull.com/int-en/no-mans-sky-for-pc-and-ps4-statistics`

[49] Hello Games: No Man's Sky [online]. 2016, [cit. 2022-05-15]. Dostupné z: `https://www.nomanssky.com/`

[50] Naď, A.: *Implementace Diehard testů pro testování generátorů pseudonáhodných čísel*. Diplomová práce, Univerzita Tomáše Bati ve Zlíně, Ústav elektroniky a měření. Vedoucí práce Žáček, Petr, Zlín, 2018, [cit. 2022-05-15]. Dostupné z: `https://digilib.k.utb.cz/handle/10563/43244`

[51] GNU Radio project: GNU Radio Companion [online]. 2022, [cit. 2022-05-15]. Dostupné z: `https://www.gnuradio.org/`

[52] Anaconda Inc.: Anaconda Distribution [online]. 2022, [cit. 2022-05-15]. Dostupné z: `https://www.anaconda.com/products/distribution`

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PIN	Personal Identification Number
NIST	National Institute of Standards and Technology
AES	Advanced Encryption Standard
EM	Elektromagnetické
RFID	Radio Frequency Identification
GHz	Gigahertz
MHz	Megahertz
λ	lambda
Ω	Omega
S_{11}	koeficient odrazu
dB	decibel
PCB	Printed circuit board
EMS	Elektromagnetická susceptibilita
SLM	Selective laser melting
SLA	Stereolitograpfie
SDR	Softwarově definované rádio
ADC	Analog-Digitální převaděč
DAC	Digital-Analogový převaděč
MCU	Master Controling Unit
sps	Snímky za sekundu
DC	Stejnosměrný proud
I	In-Phase
Q	Quadrature
i	imaginární jednotka
j	imaginární jednotka (v elektrotechnice)
FFT	Fast Fourier Transformation
USB	Universal Serial Bus
SMA	SubMiniaturní verze A
CPLD	Complex programmable logic device
ARM	Advanced Reduced Instruction Set Computer Machines
CMOS	Complementary Metal–Oxide–Semiconductor
LTE	Long Term Evolution

AIS	Anwendungshinweise und Interpretationen
IEC	International Electrotechnical Commission
SP	Special Publication
EC	Eliptic Curve
DRBG	Deterministic random bit generator
NSA	National Security Agency
QRNG	Quantum Random Number Generator
FRAM	Ferroelectric Random Access Memory
SHA	Secure Hash Algorithm
ID	Identification
PCIe	Peripheral Component Interconnect Express
LED	Light Emiting Diode
RSA	Rivest–Shamir–Adleman
BSI	Bundesamt für Sicherheit in der Informationstechnik
DVB	Digital Video Broadcasting
SMA-F	SubMiniature version A - Female
RGBA	Red Green Blue Alpha

SEZNAM OBRÁZKŮ

SEZNAM TABULEK