



Diplomová práce

Přípravek pro demonstraci různých modulací signálu

Studijní program:

N2612 Elektrotechnika a informatika

Studijní obor:

Mechatronika

Autor práce:

Bc. Jiří Čejka

Vedoucí práce:

Ing. Lubomír Slavík, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



Zadání diplomové práce

Přípravek pro demonstraci různých modulací signálu

<i>Jméno a příjmení:</i>	Bc. Jiří Čejka
<i>Osobní číslo:</i>	M20000181
<i>Studijní program:</i>	N2612 Elektrotechnika a informatika
<i>Studijní obor:</i>	Mechatronika
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Prostudujte problematiku modulací vysokofrekvenčního signálu.
2. Vytvořte software pro generování a příjem různých typů modulace signálu (AM, FM, ASK, FSK, PSK a QAM) pomocí SDR.
3. Realizujte měření signálu na dostupných přístrojích v laboratoři měření a navrhnete laboratorní úlohu.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40-50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

- [1] MACHÁČEK, Zdeněk a Pavel NEVŘIVA. *Modulované signály: učební text*. Ostrava: Vysoká škola báňská – Technická univerzita, 2012. ISBN 978-80-248-2600-4.
- [2] HANUS, Stanislav a Jiří SVAČINA. *Vysokofrekvenční a mikrovlnná technika: přednášky*. Vyd. 2., dopl. Brno: Vysoké učení technické, 2002. ISBN 80-214-2222-X.
- [3] Stewart Robert, Barlee Kenneth, Atkinson Dále: *Software Defined Radio using MATLAB & Simulink and the RTL-SDR*, Strathclyde Academic Media, 2015, ISBN: 099297871

Vedoucí práce: Ing. Lubomír Slavík, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2022

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Přípravek pro demonstraci různých modulací signálu

Abstrakt

Práce se zabývá tvorbou programu a úlohy do laboratoře pro prezentaci základních modulací (AM, FM, ASK, PSK, FSK a QAM). Pro kompletní prezentaci modulací se využívá SDR. Program kromě prezentace modulací ovládá i SDR, které se používá pro vysílání a příjem. Pro tvorbu celého programu je použitý programovací jazyk Python.

Klíčová slova: SDR, AM, FM, ASK, FSK, PSK, QAM, Python, ADALM PLUTO

Signal modulation demonstration tool

Abstract

The thesis deals with the creation of a program and a task for the laboratory for the presentation of basic modulations (AM, FM, ASK, PSK, FSK and QAM). SDR is used for the complete presentation of modulations. In addition to the presentation of the modulation, the program also controls the SDR, which is used for transmission and reception. The Python programming language is used to create the entire program.

Keywords: SDR, AM, FM, ASK, FSK, PSK, QAM, Python, ADALM PLUTO

Poděkování

Rád bych poděkoval vedoucímu mé práce za jeho pomoc a vedení, a že mi dal volnou ruku v tvorbě programů. Dále bych rád poděkoval mé snoubence za pochopení, které měla během mého stráveného času u počítače, a jistě velmi náročnou kontrolu mé diplomové práce. Nakonec bych rád poděkoval všem, kteří mi během tvorby této práce pomohli, a tedy přispěli ke vzniku tohoto díla.

Obsah

Seznam zkratek	13
1 Úvod	14
2 Softwarově definované rádio	15
2.1 Popis SDR	15
2.2 ADALM PLUTO	16
2.2.1 Podpora ADALM PLUTO	18
2.2.2 Parametry	19
3 Modulace	22
3.1 AM	22
3.2 FM	26
3.3 ASK	29
3.4 FSK	32
3.5 PSK	35
3.6 QAM	38
4 Program	41
4.1 Knihovny	42
4.2 Modulace	43
4.3 PySDR	46
4.3.1 Vysílání	47
4.3.2 Příjem	48
4.3.3 Demodulace	50
4.3.4 Popis GUI	55
4.4 PyModulace	61
4.4.1 Popis GUI	62
5 Úloha pro studenty	67
5.1 Řešení úloh	67
5.1.1 Úloha 1 - rozpoznání modulace a rekonstrukce modulovaného signálu	67
5.1.2 Úloha 2 - příjem signálu s neznámou modulací	70
5.1.3 Odpovědi na otázky	70
6 Závěr	72

Použitá literatura	74
Přílohy	75
A Seznámení se základními modulacemi	75
A.1 Úloha 1 - rozpoznání modulace a rekonstrukce modulovaného signálu	75
A.2 Úloha 2 - příjem signálu s neznámou modulací	78
A.3 Protokol	78

Seznam obrázků

2.1	Rozdíl mezi SDR a analogovým rádiem [3]	15
2.2	Blokový popis SDR [4]	16
2.3	ADALM-PLUTO [5]	17
2.4	PlutoSDR AD9361 a Xilinx Zynq [5]	17
2.5	PlutoSDR - AD9361 [7]	18
2.6	Datový tok v PlutoSDR [6]	18
2.7	Frekvenční charakteristika antény [9]	21
2.8	Vyzařovací charakteristika antény [10]	21
3.1	Kategorizace modulací [11]	22
3.2	AM - průběh modulovaného, modulačního signálu a nosného signálu	23
3.3	AM - Frekvenční spektrum	25
3.4	FM - Modulační signál má průběh sinus [13]	28
3.5	FM - Modulační signál má průběh obdelník [13]	29
3.6	Frekvenční a fázové spektrum FM [13]	29
3.7	Konstelční diagram 8-ASK [13]	30
3.8	Časový průběh 8-ASK [13]	31
3.9	4-ASK na pravé polorovině	32
3.10	Časový průběh 4-FSK	33
3.11	Frekvenční spektrum 4-FSK	34
3.12	Konstelční diagram 4-FSK	35
3.13	Konstelční diagram Q-PSK [13]	36
3.14	8-PSK	37
3.15	O-QPSK [14]	38
3.16	Konstelční diagram 16-QAM [13]	39
3.17	Časový průběh 16-QAM	40
4.1	Frekvenční spektrum a waterfall v PySDR	46
4.2	Zobrazení demodulovaného signálu AM nebo FM	51
4.3	Zobrazení zachyceného signálu	52
4.4	Diagram demodulace	54
4.5	Diagram zpracování paketu	55
4.6	Pohled na celé GUI PySDR	56
4.7	Pohled na parametrizační část GUI PySDR	57
4.8	Nastavení příjmu GUI PyModulace	58
4.9	Nastavení vysílání GUI PyModulace	59

4.10	Nastavení demodulace GUI PyModulace	60
4.11	Menu pro nastavení zobrazení GUI PyModulace	60
4.12	Okno s chybovou hláškou GUI PyModulace	60
4.13	Pohled na celé GUI PyModulace	63
4.14	Pohled na parametrizační část GUI PyModulace	65
4.15	Pohled na toolbar pro zobrazování GUI PyModulace	66
5.1	Průběh 1	68
5.2	Průběh 2	68
5.3	Průběh 3	68
5.4	Průběh 4	69
5.5	Průběh 5	69
5.6	Průběh 6	69
5.7	Průběh 7	70
5.8	Průběh 8	70
A.1	Průběh 1	75
A.2	Průběh 2	76
A.3	Průběh 3	76
A.4	Průběh 4	76
A.5	Průběh 5	77
A.6	Průběh 6	77
A.7	Průběh 7	77
A.8	Průběh 8	78

Seznam tabulek

2.1	Parametry PlutoSDR [5][6]	20
2.2	Parametry částí PlutoSDR [6]	20
2.3	Parametry antén PlutoSDR [9][10]	20

Seznam zdrojových kódů

4.1	Responzivní designe ukázka	41
4.2	Rekurzivní algoritmus	43
4.3	QAM	45
4.4	PSK	45
4.5	FSK	45
4.6	AM	45
4.7	FM	46
4.8	Vysílání	48
4.9	Vytvoření objektu pro SDR	49
4.10	Paralelizace	49
4.11	Obálka signálu	52
4.12	Okamžitá frekvence	52
4.13	Paralelizace v PyModulace	61
4.14	Matplotliv v GUI	62

Seznam zkratek

ADC	Digital to Analog Converter
AM	Amplitudová modulace
ASK	Amplitude-shift Keying
DAC	Analog to Digital Converter
DSB	Double Side Band
DSB-SC	Double Side Band Suppressed Carrier
FFT	Fast Fourier Transform
FM	Frekvenční modulace
FSK	Frequency-shift Keying
GUI	Graphical User Interface
IQ	In-phase, Quadrature
LSSB	Lower Single Side Band
LSSB-SC	Lower Single Side Band Suppressed Carrier
QAM	Quadrature Amplitude Modulation
PSK	Phase-shift Keying
RAM	Random Access Memory
SDR	Softwarově definované rádio
SSB-SC	Single Side Band Suppressed Carrier
USSB	Upper Single Side Band
USSB-SC	Upper Single Side Band Suppressed Carrier

1 Úvod

Prvním cílem této práce je vytvoření programu, který v kooperaci s SDR (softwarově definované rádio) přiblíží základní modulace studentům. Program bude umožňovat úpravu parametrů jednotlivých modulací a následné zobrazení této modulace. Modulovaný signál je pak možné vysílat a přijímat pomocí SDR. Kromě běžných modulací AM (amplitudová modulace) a FM (frekvenční modulace) program umožňuje pracovat uživatelům i s modulacemi ASK (amplitude-shift keying), FSK (frequency-shift keying), PSK (phase-shift keying) a QAM (quadrature amplitude modulation).

Druhým cílem této práce je vytvoření úlohy pro studenty, aby se seznámili se základními modulacemi. Tato úloha bude využívat vytvořeného programu z prvního cíle práce a vybavení v laboratoři. Úloha by měla obsahovat vysílání a příjem dat.

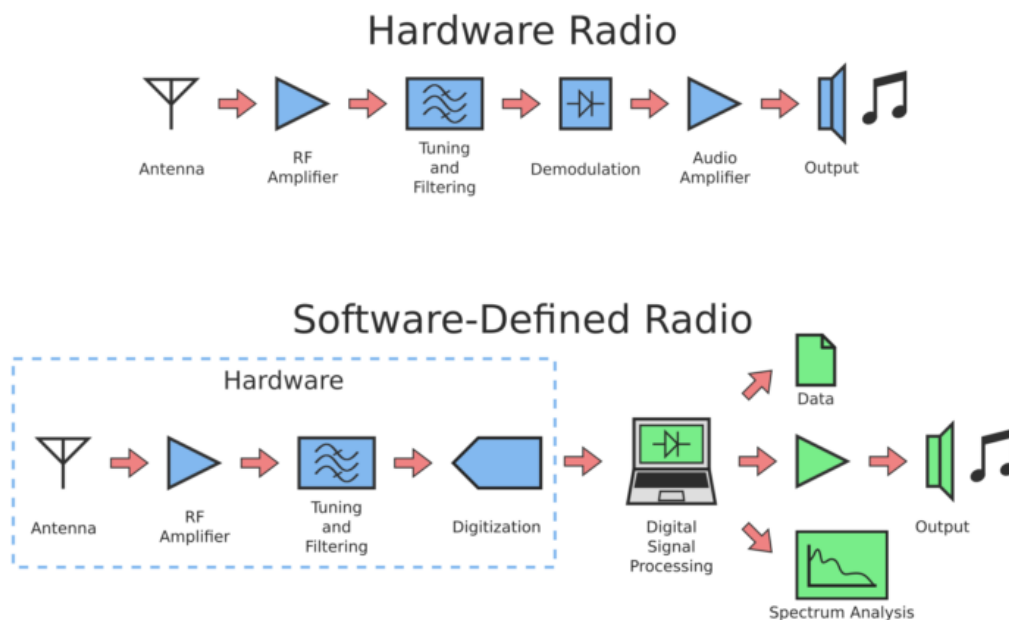
Na začátku práce je popsáno SDR a jaké SDR bude použito. Dále následuje teorie týkající se všech použitých modulací ve vytvořených programech. Po teorii následuje popis programů a jaké knihovny se používají, jak programy fungují, jaké nastaly problémy a popis GUI. Následují vypracované vytvořené úlohy pro studenty. Předposlední je závěr a za ním je příloha, kde je možné najít vytvořené úlohu.

2 Softwarově definované rádio

2.1 Popis SDR

SDR (softwarově definované rádio) je technologie, která umožňuje příjem a přenos rádiových signálů pomocí počítače a specializovaného hardwaru. SDR se liší od tradičního rádia v tom, že jeho funkce jsou definovány softwarem, což znamená, že jeho schopnosti a vlastnosti lze snadno upravit, viz obrázek (2.1). Úprava se provádí pomocí softwaru. Tato flexibilita umožňuje použití SDR pro širokou škálu aplikací, od amatérského rádia po profesionální použití v telekomunikacích. [1]

SDR je používáno v rádiové komunikaci a může být použito pro různé účely, jako je příjem a vysílání rádiových signálů, vytváření rádiových map, analýza frekvenčního spektra nebo vývoj nových technologií. Jeho schopnosti a flexibilita jsou klíčem k jeho širokému využití a v oblasti bezdrátové komunikace se stává stále více rozšířeným. Obrázek (2.1) zobrazuje flexibilitu SDR proti klasickému analogovému rádiu. [2]

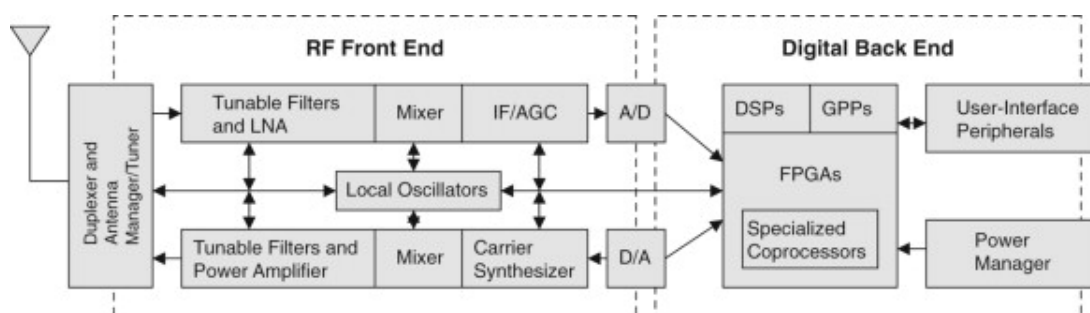


Obrázek 2.1: Rozdíl mezi SDR a analogovým rádiem [3]

Jednou z hlavních výhod SDR je jeho schopnost pracovat v širokém rozsahu frekvencí a modulací. To umožňuje přístup k širokému spektru rádiových služeb a dělá z SDR vysoce univerzální technologii. Funkce SDR jsou navíc definovány softwarově, to znamená, že je lze snadno aktualizovat a přizpůsobit aktuálnímu účelu, což z něj činí cenný nástroj pro mnoho aplikací. [2]

SDR má řadu dalších výhod. Lze provádět implementaci pokročilých algoritmů zpracování signálu, umožňujících vytvoření sofistikovaných rádiových systémů. Umožňuje také snadnou integraci různých rádiových technologií, a díky tomu lze vytvořit hybridní rádiové systémy, které kombinují nejlepší vlastnosti různých technologií.

Obrázek (2.2) zobrazuje blokově z čeho se SDR skládá a na jaké části se dělí. SDR má dvě hlavní části, a to analogovou (RF Front End) a digitální (Digital Back End) část. V analogové části dochází hlavně ke snížení frekvence přijímaného signálu na frekvenci, která je nejvhodnější pro AD převodník a filtraci. Do digitální části patří software, paměť atd.



Obrázek 2.2: Blokový popis SDR [4]

Závěrem lze říci, že SDR je vysoce flexibilní a všestranná technologie, která má velké množství různých aplikací v oblasti rádiové komunikace. Jeho schopnost pracovat v širokém rozsahu frekvencí a modulací, stejně jako jeho přizpůsobivost a snadná aktualizace, z něj dělá cenný nástroj pro různé účely. Jeho rostoucí relevance v oboru je důkazem jeho schopností a potenciálu pro budoucí rozvoj. [2]

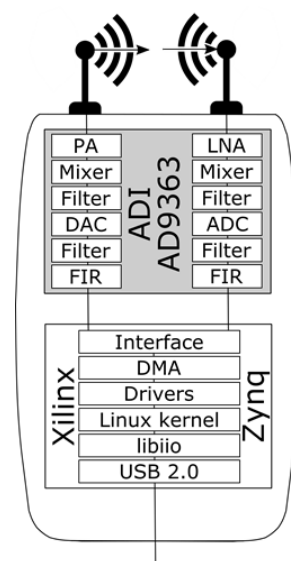
2.2 ADALM PLUTO

Pro ADALM PLUTO můžeme použít i název PlutoSDR, který také budeme používat dále v textu. Tím, že PlutoSDR je SDR, dokáže zachytit jakýkoli signál, který je v rozsahu PlutoSDR (viz kapitola 2.2.2). Zachycená data zpracuje a pošle do PC pro další zpracování. Kromě přijímání signálu PlutoSDR dokáže i vysílat. Pro vysílání je potřeba software, který to umožňuje (což umožňuje většina kompatibilních softwarů).



Obrázek 2.3: ADALM-PLUTO [5]

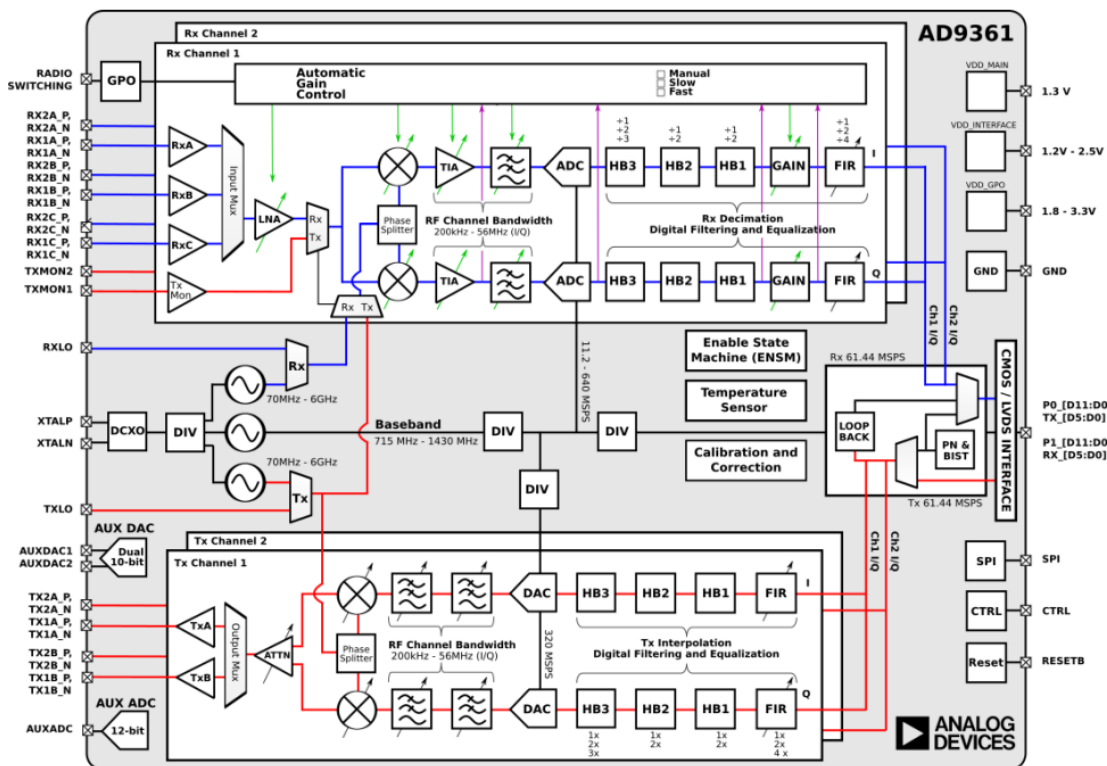
PlutoSDR je již od výrobce určeno k výuce. V praxi to znamená nižší výkon při vysílání a PlutoSDR je testováno jen na rozmezí 10 °C až 40 °C [6]. Běžná SDR mají testované rozmezí 0 °C až 70 °C nebo -40 °C až +85 °C.



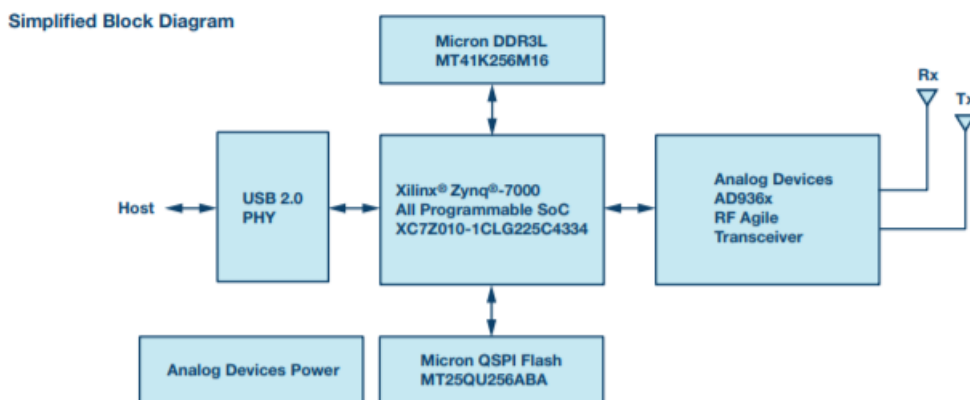
Obrázek 2.4: PlutoSDR AD9361 a Xilinx Zynq [5]

Obrázek (2.4) zobrazuje blokové zapojení PlutoSDR. Kromě zapojení je možné vidět dvě nejdůležitější části PlutoSDr, a to AD9363/AD9361 (existují dvě verze PlutoSDR, není v nich rozdíl) a Xilinx Zynq.

Obrázek (2.5) je podrobnější pohled na AD9363 respektive AD9361. Obrázek (2.6) ukazuje datový tok a komunikaci v PlutoSDR mezi jednotlivými prvky.



Obrázek 2.5: PlutoSDR - AD9361 [7]



Obrázek 2.6: Datový tok v PlutoSDR [6]

2.2.1 Podpora ADALM PLUTO

PlutoSDR podporuje práci s velkým množstvím softwarů. Kromě programů podporuje i některé programovací jazyky a knihovny. Mezi podporované programy patří:

- MATLAB
- GNU Radio

- IIO Oscilloscope
- Pluto Web
- SDR Sharp
- a další

Soupis a rešerši podporovaných programů pro PlutoSDR lze najít zde. [1] Softwarů pro SDR je však velké množství. Jsou to softwary univerzální, ale mohou být i zaměřené na konkrétní činnost. Příkladem konkrétní činnosti je zpracování televizního signálu. Soupis většiny softwarů je zde. [8] Kromě programů podporuje PlutoSDR i jazyky a knihovnu. Podporovaná knihovna je libiio. Podporované jazyky jsou [5]:

- C
- C++
- C#
- Python

2.2.2 Parametry

Soupis základních parametrů PlutoSDR je v tabulce (2.1). Hodnoty v závorce jsou platné při odebrání limitu na čipu AD9361 respektive AD9363. Odebrání limitu na čipu není nic složitého a ani nic rizikového. Toto omezení bylo pravděpodobně dáno kvůli vysokým podmínkám na čip výrobce, ale čip sám o sobě zvládá větší frekvence, než jsou povoleny z továrny.

PlutoSDR lze připojit i k něčemu jinému než je počítač. Přes USB 2.0. PlutoSDR ho lze připojit k flash disku nebo ho můžeme připojit k PlutoSDR přes USB LAN (Ethernet) nebo WiFi. Při použití připojení USB 2.0 dojde k pomyslnému omezení vzorkování pokud chceme minimální ztráty (v programech lze nastavit i pořadí maximální vzorkování, pouze dojde ke ztrátám vzorků). Omezení je způsobeno maximální přenosovou rychlostí USB 2.0. Podle výrobce je maximální vzorkování někde mezi 7,5 a 12 *MSPS*. Omezení lze obejít ukládáním vzorků do paměti a vyhodnocením až později.

Tabulka 2.1: Parametry PlutoSDR [5][6]

Frekvenční rozsah	325 (70) MHz - 3,8 (6) GHz
Frekvenční pásmo	200 kHz - 20 (56) MHz
ADC	12-bit flexibilní
DAC	12-bit flexibilní
Vzorkování	25,2 kSPS – 61,44 MSPS
Připojení antény	SMA samice
Impedance	50 Ω
Provoz	Half a Full duplex
Napájení	USB 2.0
Připojení k PC	USB 2.0
Frekvenční přesnost	± 25 ppm
Počet vstupů	1
Počet výstupů	1

Tabulka (2.2) popisuje základní parametry důležitých částí PlutoSDR jako například ARM, RAM (random access memory) a jiné. Tabulka (2.3) zachycuje základní parametry antény, která je v balení k PlutoSDR. K anténě také patří obrázek (2.7), který zobrazuje frekvenční charakteristiku antén. Dále k ní patří obrázek (2.8), zobrazující vyzařovací charakteristiku antény. Charakteristika zobrazuje vertikální a horizontální vyzařování pro sedm frekvencí.

Tabulka 2.2: Parametry částí PlutoSDR [6]

ARM Cortex A9	667 MHz
Xiling Zynq	28 000 logických buněk
DDR3L	512 MB
QSPI Flash	32 MB

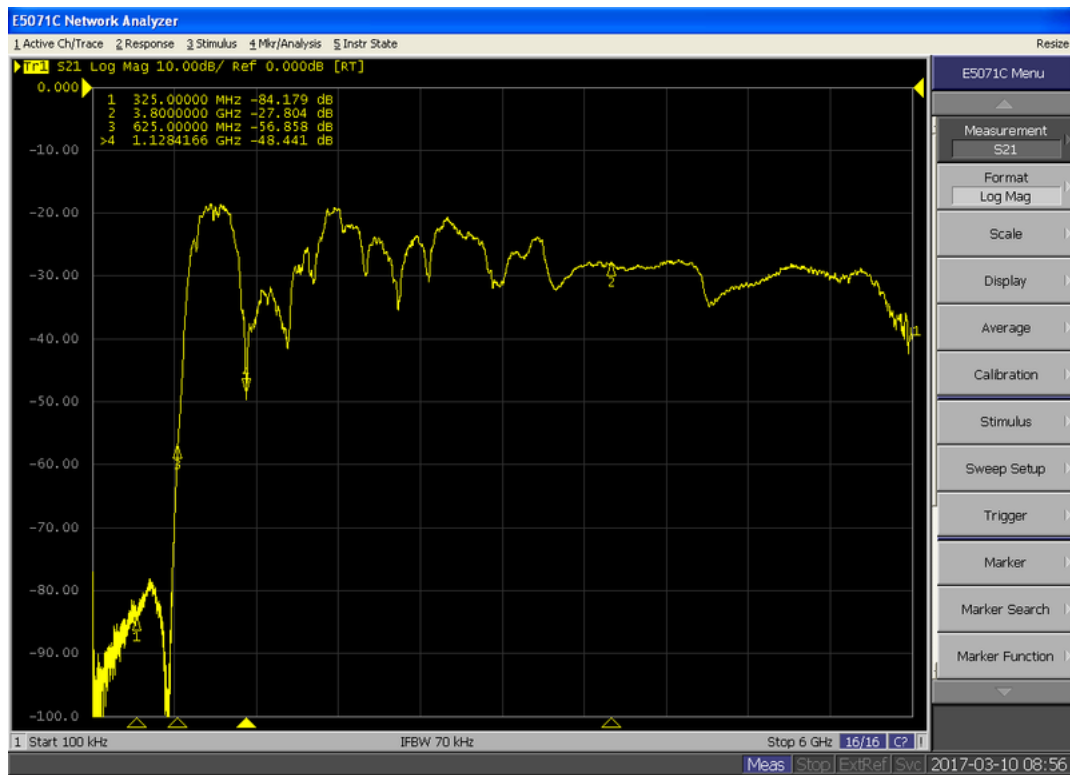
Tabulka 2.3: Parametry antén PlutoSDR [9][10]

Frekvenční rozsah	824 – 894 MHz
	1,71 – 2,17 GHz
Zesílení	2 dBi
Polarizace	Lineární

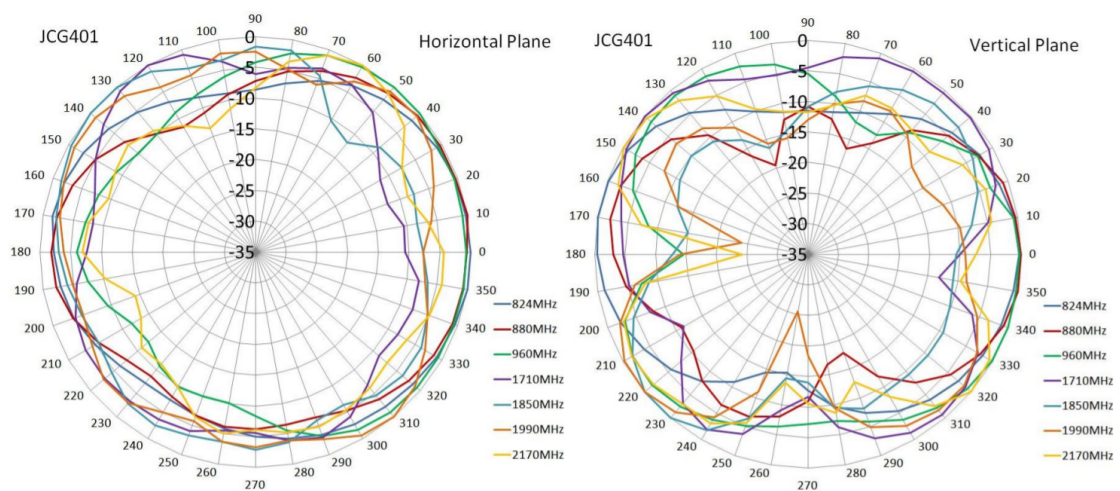
Antény k PlutoSDR, které jsou dodávány výrobcem jsou JCG401 od firmy Jinchang Electron. Anténa má samozřejmě impedanci 50 Ω a dokáže operovat od -40 $^{\circ}\text{C}$ do 85 $^{\circ}\text{C}$. Vydrží vibrace od 10 do 55 Hz s amplitudou 1,5 mm po dobu 2 *hodin*. [10]

Měření frekvenční charakteristiky antény proběhlo s využitím E5071C ENA Series Network Analyzer a měřením S-parametru. Konkrétně to byl parametr S_{12} a S_{21} (indexy označují porty/antény). [9]

Anténa má plně vyhovující parametry od 625 MHz. [9] Výjimkou je malý propad na 1,12 GHz. Průběh charakteristiky pod 625 MHz neznamená, že nepůjde zachytit třeba FM vysílání, jen je potřeba, aby vysílání bylo silné. Je vyzkoušené, že lze zachytit FM vysílání na frekvencích pod 625 MHz.



Obrázek 2.7: Frekvenční charakteristika antény [9]

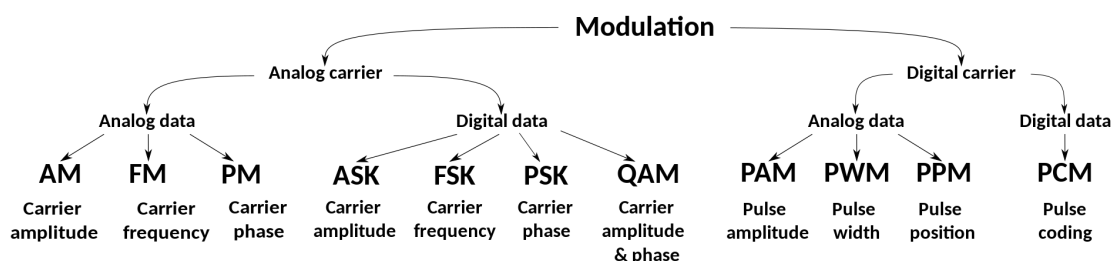


Obrázek 2.8: Vyzařovací charakteristika antény [10]

3 Modulace

Modulace se dělí do několika skupin. Pro znázornění těchto skupin, stejně tak i toho, do kterých patří jednotlivé modulace, které jsou popsány v této práci a další, je zde obrázek, který toto vše zachycuje v diagramu (3.1). U každé modulace je zapsáno pomocí jakého parametru signálu se vysílají data, respektive je u každé modulace její velmi stručný princip.

Základní dělení je na analogové modulace a digitální modulace. Obě tyto kategorie se dále dělí na základě typu dat (analogová data a digitální data).



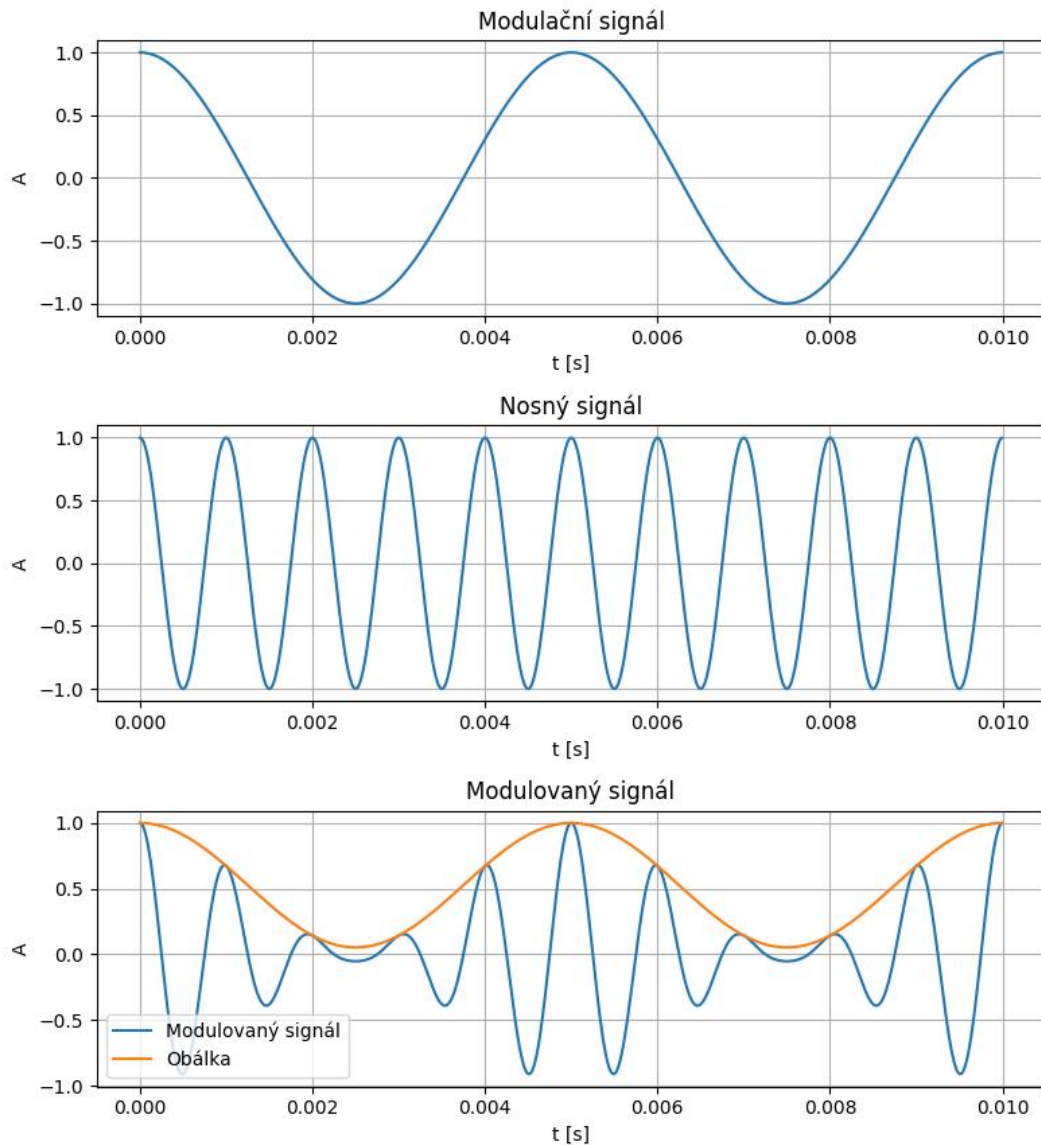
Obrázek 3.1: Kategorizace modulací [11]

3.1 AM

Amplitudová modulace, jak již napovídá název, funguje na principu změny amplitudy na základě aktuální hodnoty modulačního signálu. Princip je znázorněný na obrázku (3.2), kde je zobrazený průběh modulačního signálu a následný průběh modulovaného signálu. Modulovaný signál, pokud není potlačena nosná a má obě postranní pásma, je popsán vztahem

$$g(t) = A_c[1 + m(t)]. \quad (3.1)$$

Amplituda nemodulované nosné je A_c a je to konstanta. Dále $m(t)$ je modulační signál. A modulovaný signál je $g(t)$. Modulační signál může být analogový nebo i digitální.



Obrázek 3.2: AM - průběh modulovaného, modulačního signálu a nosného signálu

AM má několik parametrů, které popisují modulaci a její vlastnosti. První takový parametr je hloubka pozitivní modulace AM a je definována vztahem

$$H_{pm} = \frac{A_c - A_{min}}{A_c} \cdot 100. \quad (3.2)$$

Další parametr je podobný hloubce pozitivní modulace. Tento parametr se nazývá hloubka negativní modulace a je definována vztahem

$$H_{nm} = \frac{A_{max} - A_c}{A_c} \cdot 100. \quad (3.3)$$

Podobným parametrem předchozím dvěma je hloubka modulace. Hloubka modulace

je definována vztahem

$$Hm = \frac{A_{max} - A_{min}}{2A_c} \cdot 100, \quad (3.4)$$

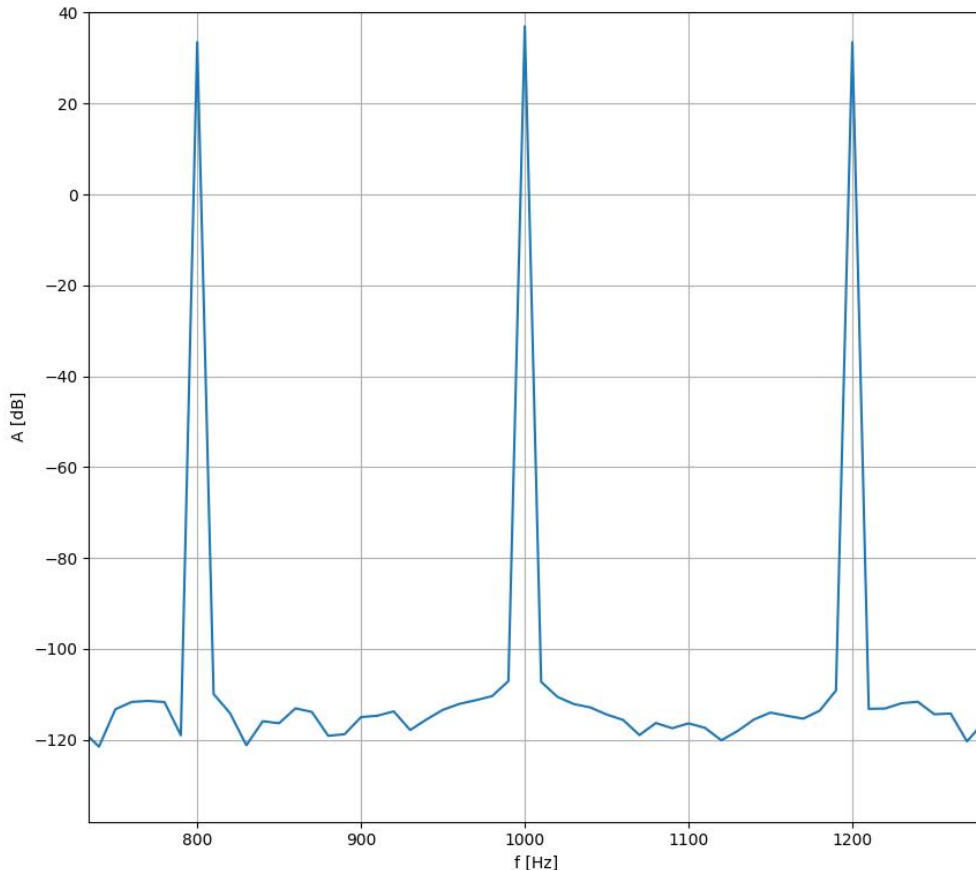
kde pro vztahy pro tyto parametry platí, že A_c je amplituda nosného signálu, A_{min} je minimální magnituda modulačního signálu a A_{max} je maximální magnituda modulačního signálu. Všechny parametry, které popisují hloubku modulace vychází v procentech. Hloubka modulace může vyjít více jak 100%, ale není to běžné. Mezi parametry patří střední výkon AM signálu. Střední výkon, pokud v modulačním signálu není konstantní složka, je definován vztahem

$$P_v = \frac{1}{2}A_c^2 + \frac{1}{2}A_c^2 \langle m_{(t)}^2 \rangle \cdot 100, \quad (3.5)$$

kde $\frac{1}{2}A_c^2$ je výkon nosné a $\frac{1}{2}A_c^2 \langle m_{(t)}^2 \rangle$ je výkon postranních pásem. Dalším parametrem je účinnost modulace, která popisuje kolik výkonu patří informacím. Účinnost modulace je definována vztahem

$$E = \frac{\langle m_{(t)}^2 \rangle}{1 + \langle m_{(t)}^2 \rangle} \cdot 100. \quad (3.6)$$

Nejvyšší účinnost je při $Hm = 100$ % a účinnost je poté 50 %. Těto účinností se dosahuje, pokud modulační signál má obdélníkový průběh.



Obrázek 3.3: AM - Frekvenční spektrum

Obrázek (3.3) zobrazuje frekvenční spektrum modulovaného signálu při $A_c = 1$. Z obrázku je patrné, že frekvenční šířka modulovaného signálu je dvakrát větší než šířka modulačního signálu. Z toho vyplývá, a je to patrné i na obrázku frekvenčního spektra, že modulovaný signál má dvě shodná postranní pásma, která jsou zrcadlově symetrická. Tato dvě pásma se označují jako horní a dolní. Na základě ponechání bočních pásem a nosné se nazývají různé typy.

Obrázek (3.3) je frekvenční spektrum modulovaného signálu na obrázku (3.2). Modulační signál má frekvenci 200 Hz a frekvence nosné je 1000 Hz , proto postranní pásma mají frekvenci 800 Hz a 1200 Hz .

AM, která má vše, se nazývá DSB (double side band) AM a má nejmenší účinnost, protože se výkon dělí mezi dvě postranní pásma a nosnou. AM, která má potlačenou nosnou se nazývá DSB-SC (double side band suppressed carrier). Postranní pásma má zachovaná. Když se porovná AM a DSB-SC, tak u DSB-SC je hloubka modulace nekonečná a chybí nosná v modulovaném signálu. Účinnost DSB-SC je 100 %, protože není potřeba energie na nosnou. Pokud má vysílač omezený špičkový výkon, tak lze pomocí DSB-SC přenést větší výkon pomocí postranních pásem. Přesněji můžeme přenést čtyřikrát více energie než pomocí AM.

AM, která má potlačené jedno postranní pásmo a potlačenou nosnou se nazývá SSB-SC (single side band suppressed carrier). Podle využívaného postranního

pásma se SSB-SC dělí na další modulace. Pokud se používá horní pásmo, tak se tato modulace nazývá USSB (upper single side band). Když se používá dolní pásmo, tak se nazývá LSSB (lower single side band). Obě varianty se používají s potlačenou nosnou, takže existují i USSB-SC (upper single side band suppressed carrier) a LSSB-SC (lower single side band suppressed carrier). [12]

Výhoda SSB-SC je v malém frekvenčním pásmu, kde šířka pásma modulovaného signálu je shodná se šířkou modulačního signálu, jinak řečeno data se přenáší jen jednou. Další výhodou je větší účinnost než u AM a SSB-SC má soustředěný výkon jen v jednom pásmu. SSB-SC je robustnější než AM. [12]

Nevýhoda SSB-SC je v kvalitě demodulace. Kvalita demodulace je závislá na frekvenční koherenci nosné, a tedy je třeba zajistit fázové koherence. Zajištění fázové koherence je nákladné. Pro nezkraslený přenos musí fázový posuv složek růst lineárně s frekvencí. [12]

3.2 FM

Frekvenční modulace patří k nejstarším modulacím (druhá po AM) a dále patří mezi nejpoužívanější analogové modulace. FM se používá především na velmi krátkých vlnách. FM byla nástupcem AM. Porovnání FM a AM:

- FM je odolnější proti rušení.
- FM má jednodušší modulátor než AM.
- FM umožňuje demodulace slabých signálů (napětově slabých signálů).
- AM má jednodušší demodulátor.

Komplexní obálka FM je popsána vztahem

$$g(t) = A_c e^{j\theta(t)}. \quad (3.7)$$

Reálná obálka, která je složkou komplexní obálky, je konstantní. Reálná obálka, respektive reálná složka, je dána vztahem

$$R(t) = \|g(t)\| = A_c, \quad (3.8)$$

kde A_c je amplituda nosné. Fáze $\theta(t)$, která je ve vztahu (3.7), je funkcí modulačního signálu. Tato funkce je lineární, naopak od komplexní obálky, která je také funkcí modulačního signálu. Výsledný signál je dán vztahem (v časové oblasti)

$$v(t) = A_c \cos(\omega_c t + \theta(t)), \quad (3.9)$$

kde $v(t)$ je modulovaný signál a ω_c je úhlová rychlost nosné. Fáze je určena vztahem

$$\theta(t) = D_f \int_{-\infty}^t m(\tau) d\tau, \quad (3.10)$$

kde D_f je konstanta a určuje frekvenční rozpětí signálu $v(t)$ neboli D_f je zesílení FM.

FM lze rozdělit podle frekvenčního pásma. Šířka pásma je závislá na velikosti D_f . FM se dělí na úzkopásmovou a širokopásmovou. Úzkopásmovou FM má $D_f < 1$ a širokopásmovou FM má $D_f > 1$. Úzkopásmová FM třeba využívají radioamatéři a širokopásmová FM se využívá v rozhlase.

K FM patří několik důležitých vztahů, které popisují FM. Okamžitá frekvence říká jaká je frekvence signálu v určitý okamžik. Okamžitá frekvence je dána vztahem

$$f_{i(t)} = f_c + \frac{1}{2\pi} \left[\frac{d\Theta(t)}{dt} \right], \quad (3.11)$$

kde f_c je frekvence nosné. Vztah pro okamžitou frekvenci lze přepsat ještě do jiné podoby, ze které je více patrné, proč je to frekvenční modulace. Toto přepsání je možné díky obrázku (3.5). Výsledný vztah vypadá takto

$$f_{i(t)} = f_c + \frac{1}{2\pi} D_f m(t), \quad (3.12)$$

kde $m(t)$ je modulační signál.

Dále tu je odchylka frekvence signálu od nosné frekvence, která je popsána vztahem

$$f_{d(t)} = f_{i(t)} - f_c, \quad (3.13)$$

který lze přepsat na

$$f_{d(t)} = \frac{1}{2\pi} \left[\frac{d\Theta(t)}{dt} \right]. \quad (3.14)$$

S odchylkou od nosné frekvence je spjatý parametr maximální odchylka frekvence, která určuje maximální rozdíl mezi nosnou frekvencí a frekvencí signálu. Maximální odchylka frekvence je definována vztahem

$$\Delta F = \max \left\{ \frac{1}{2\pi} \left[\frac{d\Theta(t)}{dt} \right] \right\}. \quad (3.15)$$

Pro FM je maximální odchylka frekvence přímo dána zesílením FM modulace D_f . Výsledný vztah je

$$\Delta F = \frac{1}{2\pi} D_f m_{max}, \quad (3.16)$$

kde m_{max} je maximální hodnota modulačního signálu. S parametrem maximální odchylky frekvence je spjatý další parametr. Odchylka frekvence mezi maximem a minimem je definována vztahem

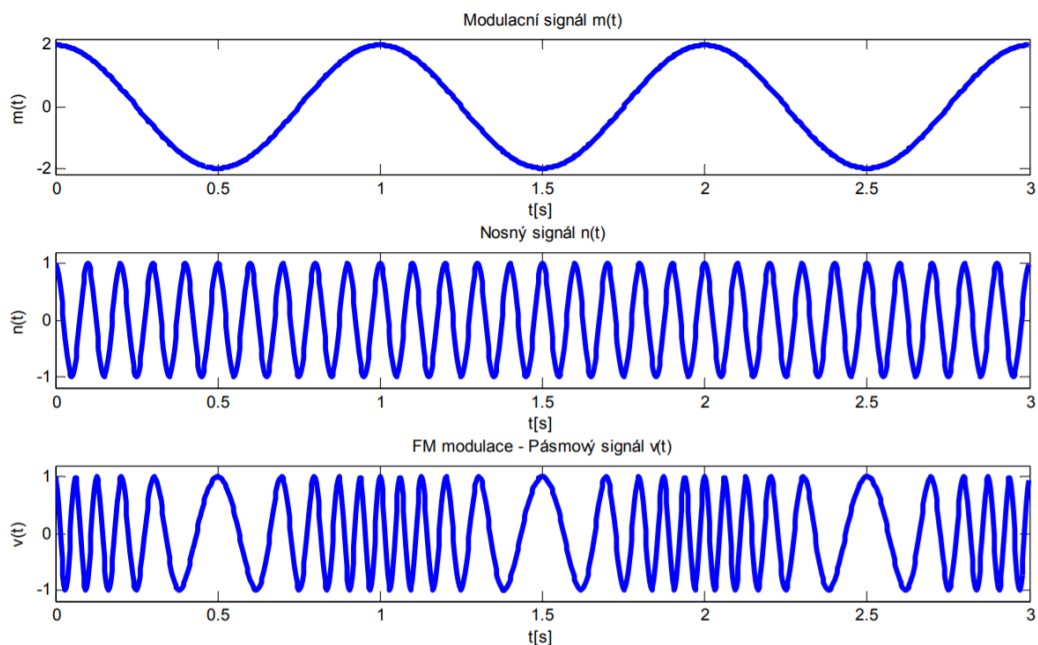
$$\Delta F_{pp} = \max \left\{ \frac{1}{2\pi} \left[\frac{d\Theta(t)}{dt} \right] \right\} - \min \left\{ \frac{1}{2\pi} \left[\frac{d\Theta(t)}{dt} \right] \right\}. \quad (3.17)$$

Jako poslední je tu parametr index frekvenční modulace, který je dán vztahem

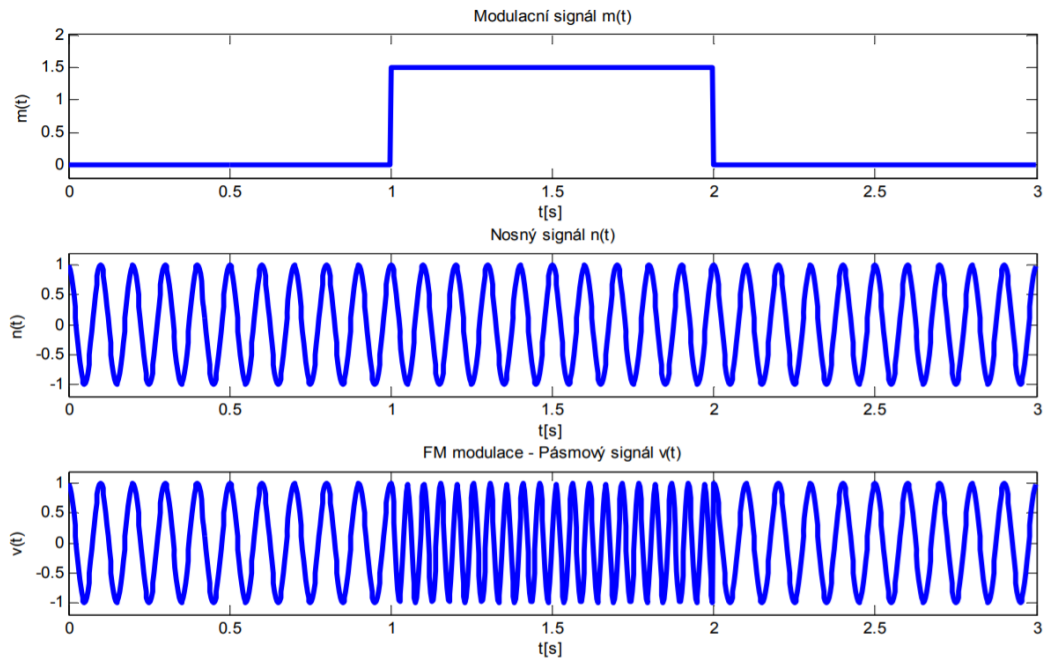
$$\beta_F = \frac{\Delta F}{B}, \quad (3.18)$$

kde B je šířka pásma modulačního signálu.

Obrázky (3.4) a (3.5) zobrazují princip FM. Rozdíl mezi obrázky je pouze v modulačním signálu, kde u prvního obrázku (3.4) je modulační signál sinus a u druhého (3.5) to je obdélník. Na prvním obrázku lze pozorovat plynulou změnu frekvence, jak se mění aktuální hodnota modulačního signálu. Druhý obrázek je nejen dobrý pro znázornění FM pro digitální signál, ale také pro určení vztahu (3.12).

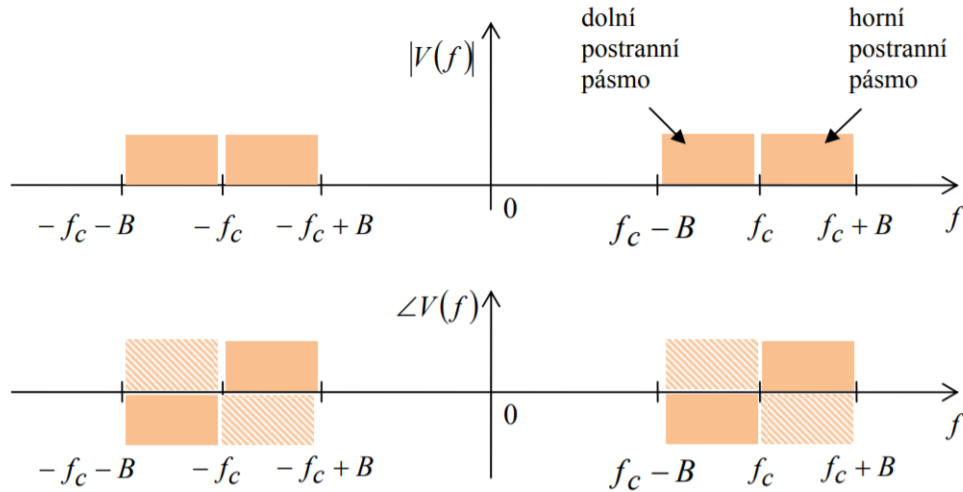


Obrázek 3.4: FM - Modulační signál má průběh sinus [13]



Obrázek 3.5: FM - Modulační signál má průběh obdelník [13]

Pro FM neplatí princip superpozice a ani jiné principy, které vychází z linearity soustavy. Další zajímavou vlastností FM je nemožnost přímo určit pozici a rozložení harmonického signálu, který je modulovaný FM, ve frekvenčním spektru. [12]



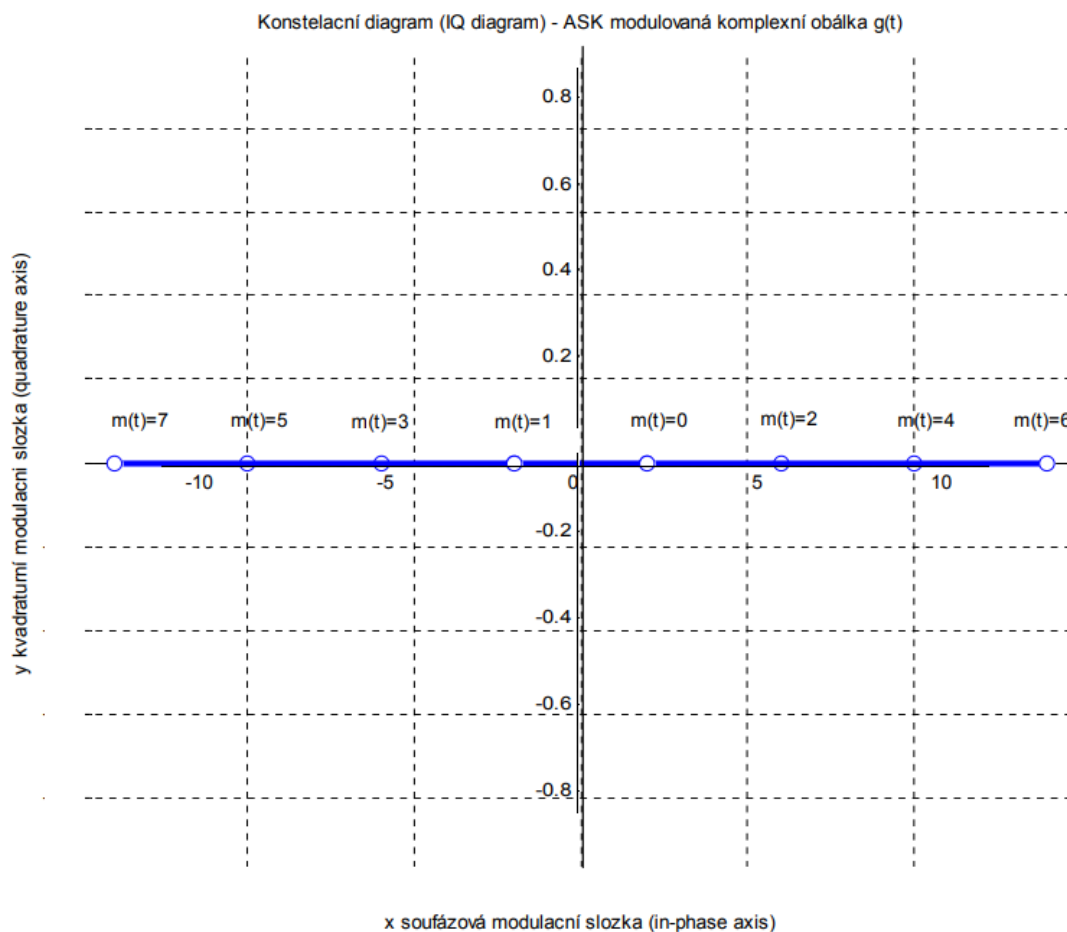
Obrázek 3.6: Frekvenční a fázové spektrum FM [13]

3.3 ASK

Plný název modulace ASK je Amplitude Shift Keying. Modulace ASK patří mezi digitální modulace. ASK je podobná analogové amplitudové modulaci, protože k pře-

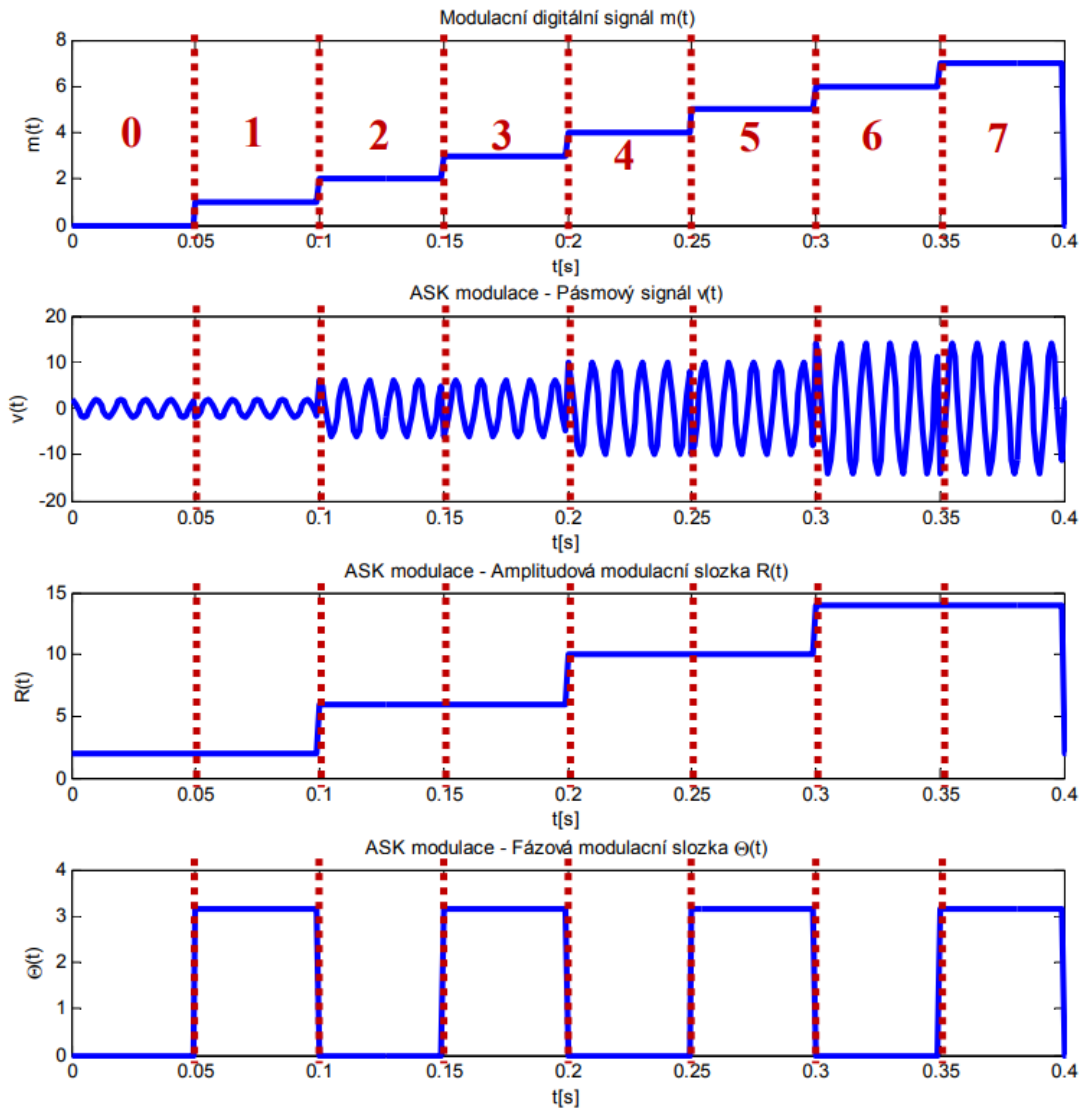
nosu informací využívá změnu amplitudy. ASK ke klíčování využívá změnu amplitudy. ASK může přenášet binární nebo vícestavové informace.

Symbols jsou rozmístěny pouze po reálné ose, jak je možné vidět na obrázku (3.7), který zobrazuje konstelační diagram ASK. Ten zobrazuje všechny vektory rozkladu modulačního signálu v komplexní rovině. Počet vektorů odpovídá počtu stavů ASK. Protože je využívána pouze reálná osa a je konstantní amplitudová složka pro dva stavy, tak mají tyto dva vektory rozdílnou fázovou modulační složku. Fázová modulační složka je 0° nebo 180° .



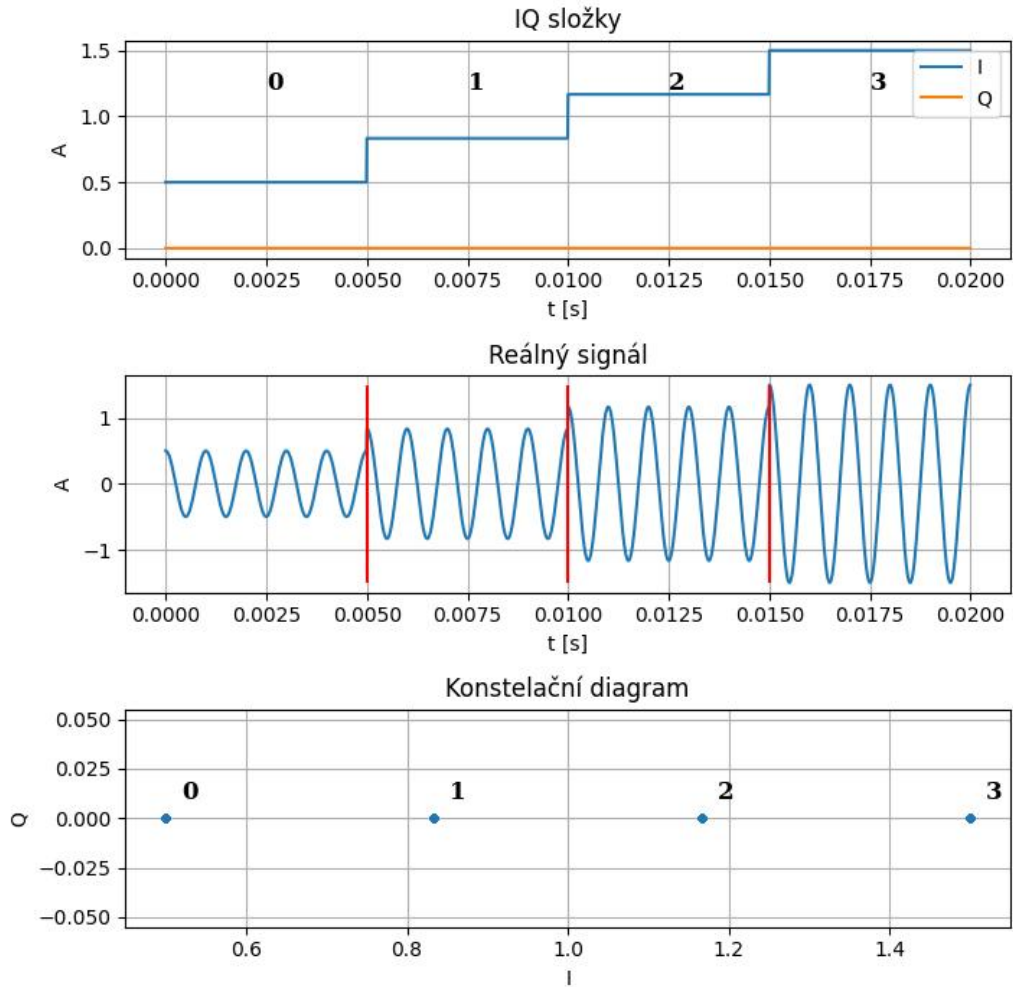
Obrázek 3.7: Konstelační diagram 8-ASK [13]

Obrázek (3.8) zobrazuje průběh signálu při použití 8-ASK a přenášená data jsou 0 až 7. V datech je několik čísel, které mají shodnou amplitudu s jiným číslem (např. 0 a 1). Rozdíl mezi těmito čísly je ve fázi (toto lze pozorovat i na konstelačním diagramu (3.7)).



Obrázek 3.8: Časový průběh 8-ASK [13]

Kromě výše zmíněné 8-ASK, která je symetrická okolo osy Q, tak může být ASK nesymetrická okolo osy Q. Příklad této možnosti je na obrázku (3.9), kde se používá pro definici symbolů pouze pravá polovina respektive kladná část osy I. V tomto případě je symbol definovaný pouze amplitudou.

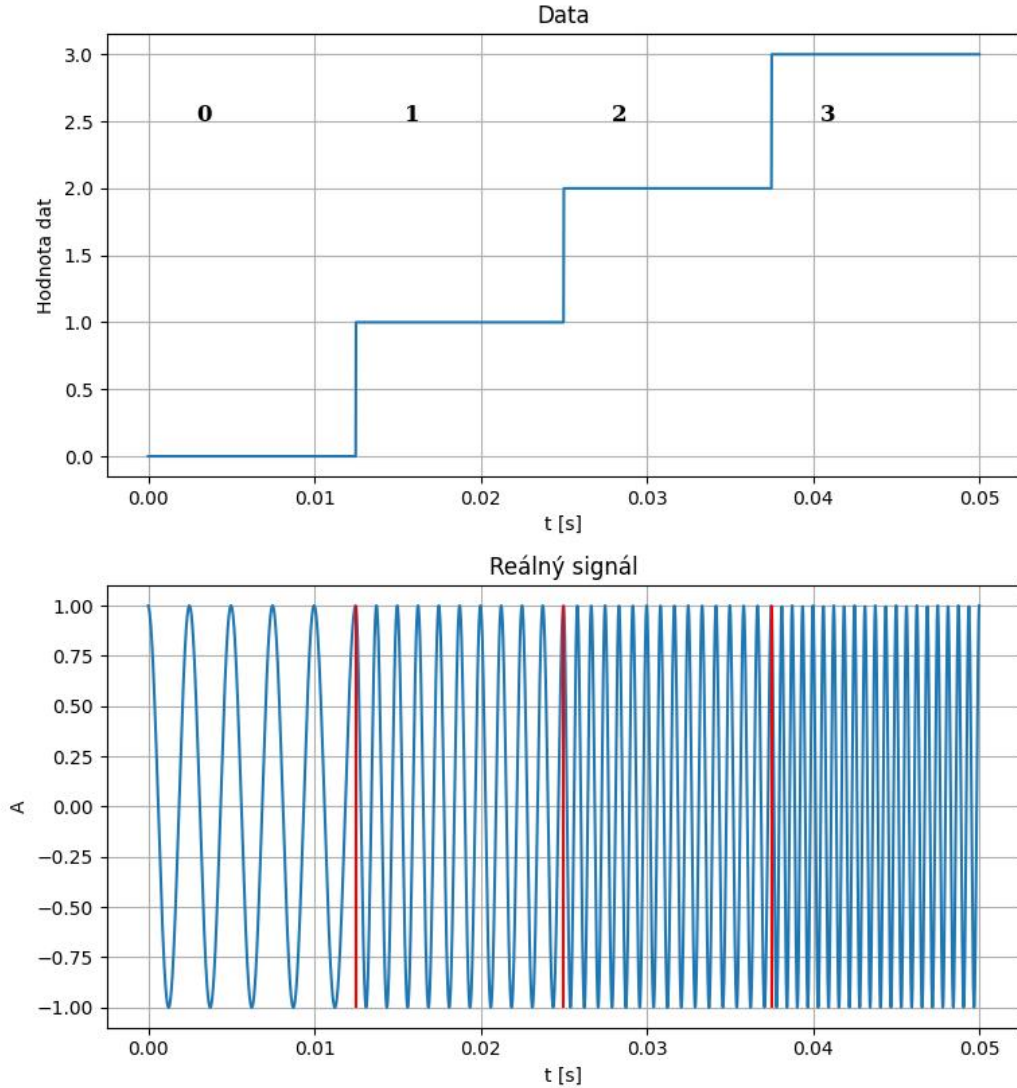


Obrázek 3.9: 4-ASK na pravé polorovině

ASK modulace je nevýhodná pro větší počet symbolů kvůli energetické účinnosti. ASK modulace se hlavně používá pro menší počet symbolů.[13]

3.4 FSK

Plný název modulace FSK je Frequency Shift Keying. Modulace FSK patří mezi digitální modulace. Tato modulace, stejně jako ASK modulace, je založená na klíčování digitálních dat. FSK modulace je digitálním ekvivalentem FM. Podobnost FSK a FM je možné pozorovat na obrázku (3.10). Z průběhu signálů na obrázku je patrná podobnost FSK a FM. Při čtyřech různých hodnotách dat má modulovaný signál (na obrázku je to reálný signál) čtyři rozdílné frekvence, to je možné vidět na obrázku (3.10) a obrázku (3.11).



Obrázek 3.10: Časový průběh 4-FSK

Amplitudová složka FSK je konstantní a fázová modulační složka FSK je proměnná. Fázová modulační složka se spojitě mění na základě přenášené hodnoty. Frekvence pro přenášenou hodnotu je konstantní, z toho vyplývá, že kanálové symboly jsou ve stejné vzdálenosti od středu konstelace. Stejná vzdálenost od středu konstelace znamená shodný modul vektoru a odlišují se fází. Zmíněné vlastnosti lze vidět na obrázku (3.12). Množina datových symbolů může být definována výrazem [13]

$$d_n = \{\pm 1, \pm 3, \dots, \pm(MAX_STAV - 1)\}, \quad (3.19)$$

kde d_n je datový symbol. Protože je frekvence konstantní, tak jsou kanálové symboly stejně daleko od středu konstelace, jinak řečeno modul symbolů je konstantní, a tedy stejný pro všechny symboly. Následkem toho platí

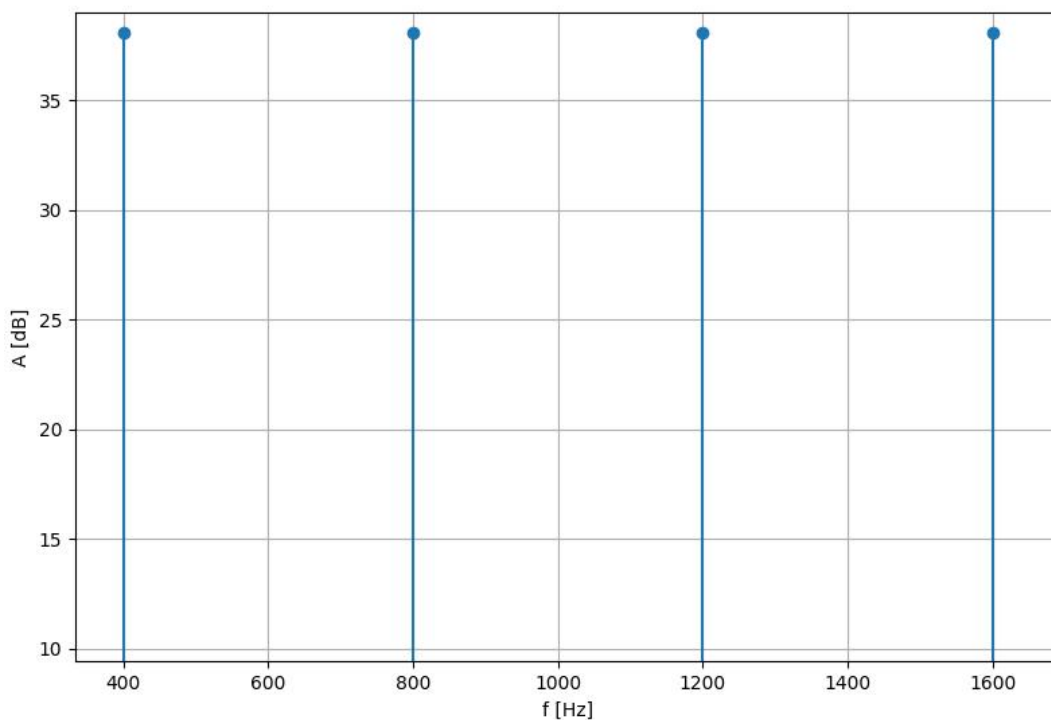
$$\|d_n\|^2 = 1. \quad (3.20)$$

A z toho vyplývá, že všechny symboly mají konstantní energii. Rozložení frekvencí pro jednotlivé symboly je ekvidistantní. Rozdíl frekvencí je kladné reálné číslo. Modulační index tedy lze definovat vztahem

$$K = \Delta f \cdot T_S, \quad (3.21)$$

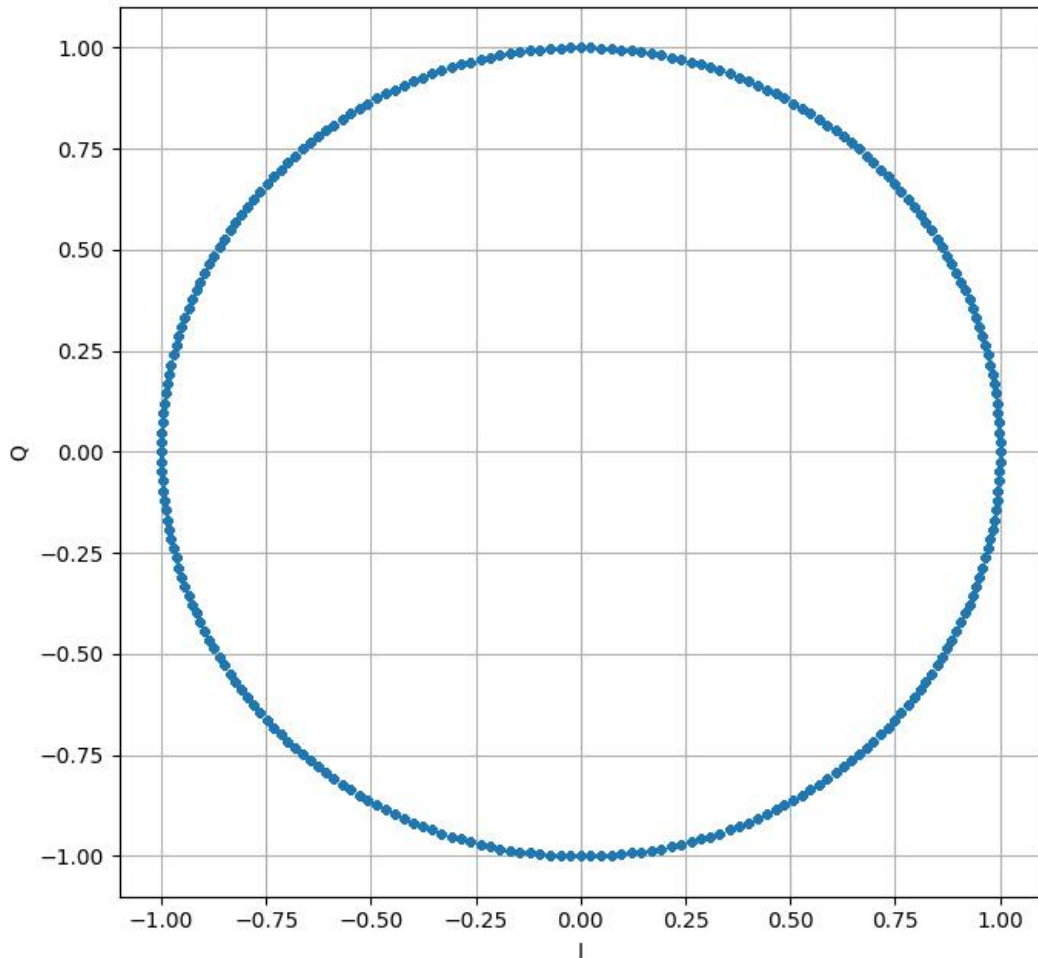
kde Δf je rozdíl frekvencí mezi sousedními symboly a T_S je perioda vzorkování.

Obrázek (3.11) a obrázku (3.10) patří k sobě. Frekvenční rozdíl mezi symboly je 400 Hz , je to patrné na obrázku (3.11). Hodnota dat (0 až 3) a odpovídající frekvence (400, 800, 1200 a 1600 Hz) jde zleva doprava.



Obrázek 3.11: Frekvenční spektrum 4-FSK

Obrázek (3.12) zobrazuje konstelační diagram FSK modulace (přesněji je to 4-FSK). Konstelační diagram odpovídá popisu z předchozí části této kapitoly. Modul je konstantní respektive vzdálenost od středu je konstantní.



Obrázek 3.12: Konstelační diagram 4-FSK

FSK má i jiné varianty než M-FSK. Příkladem je CPFSK (continuous phase FSK), kde je změna fáze plynulá, tak jak je to na obrázku (3.10). Nebo tu je GMSK (Gaussi minimum shift keying), která je vylepšenou CPFSK.[14]

3.5 PSK

Plný název PSK je Phase Shift Keying. PSK modulace patří mezi digitální modulace. Jako ASK nebo FSK používá klíčování digitálních dat. Jak již napovídá název, modulace využívá pro přenos dat změnu fáze. Přesněji to je změna fáze vektoru symbolu (v konstelačním diagramu), tato změna je viditelná na obrázku (3.13) nebo na obrázku (3.14).

PSK je složena symetrickou ekvidistantní množinou kanálových symbolů. Množina kanálových symbolů může být popsána vztahem [13]

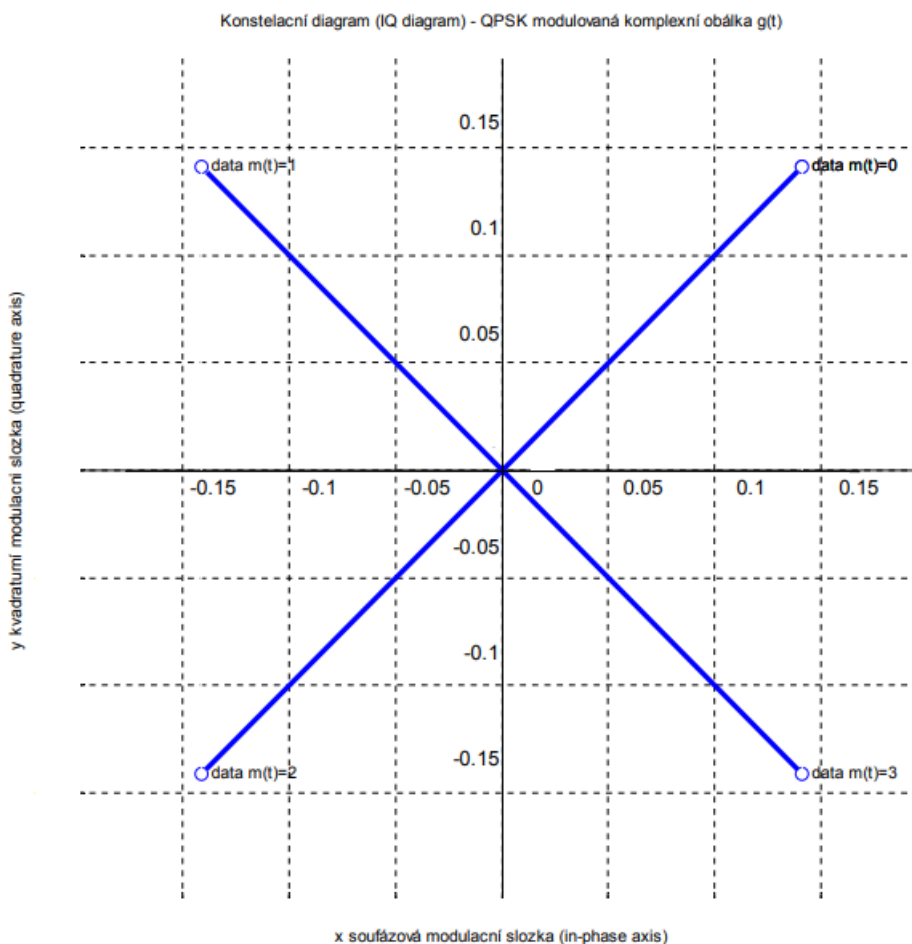
$$d_n = q_n \in \left\{ e^{j \frac{2\pi}{MAX_STAV} stav} \right\}_{stav=0}^{stav=MAX_STAV-1} . \quad (3.22)$$

Kanálové symboly jsou q_n a datové symboly jsou d_n . Modul symbolů je konstantní, což je patrné z obrázků konstelačního diagramu (3.13) nebo (3.14). Vztah pro modul je tedy

$$|d_n|^2 = 1. \quad (3.23)$$

Protože modul symbolů je konstantní, energie kanálových modulů je konstantní. Jinak řečeno vzdálenost symbolů od středu je pro všechny shodná a symboly se rozlišují pomocí jejich fáze.

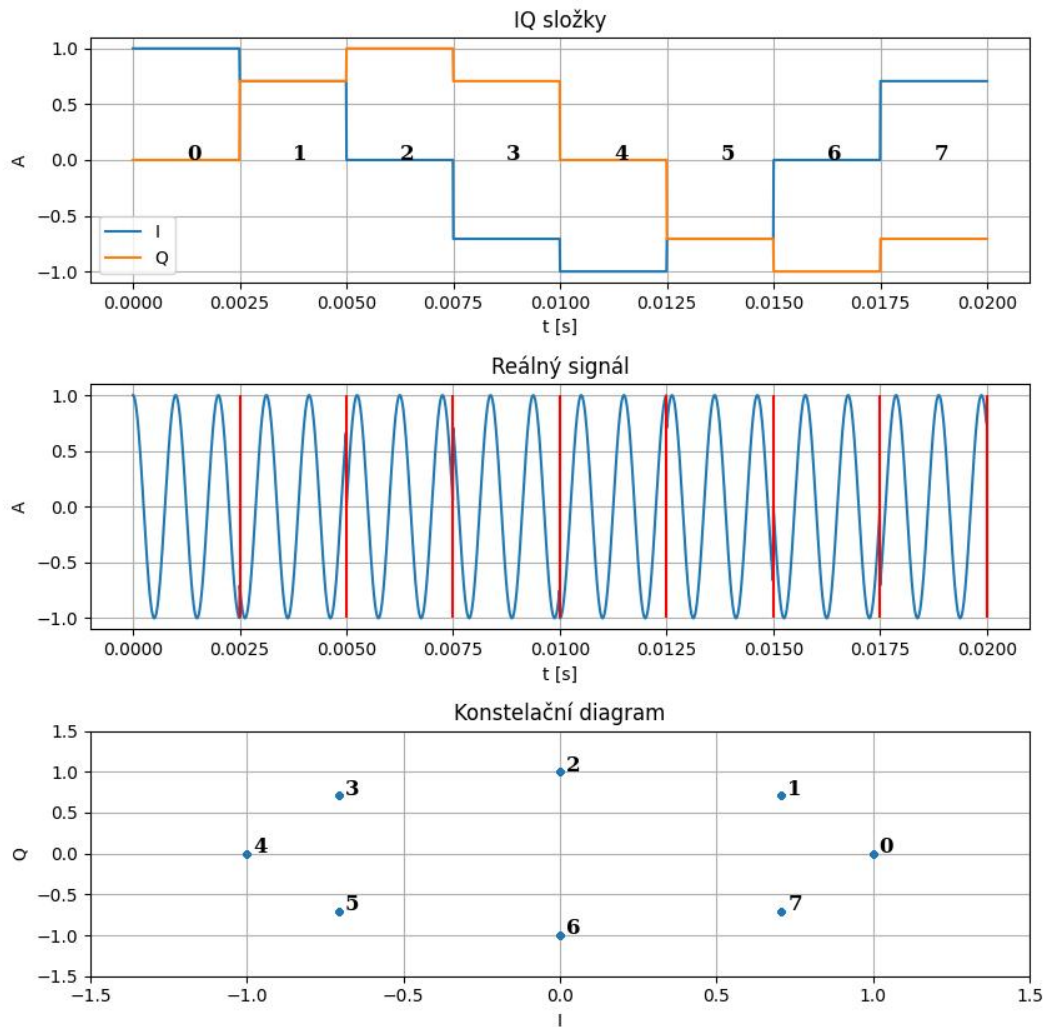
Jako u ASK nebo FSK, tak i u PSK modulace existuje více variant rozdělené na základě počtu stavů. Příkladem je B-PSK (dvoustavová) nebo Q-PSK (čtyřstavová). Čtyřstavová PSK má dvě varianty. Jedna se značí 4-PSK a má kanálové symboly $d_n \in \{1, j, -1, -j\}$. Druhé provedení se značí Q-PSK a má kanálové symboly $d_n \in \left\{ \frac{1+j}{\sqrt{2}}, \frac{-1+j}{\sqrt{2}}, \frac{-1-j}{\sqrt{2}}, \frac{1-j}{\sqrt{2}} \right\}$. Obrázek (3.13) zobrazuje konstelační diagram Q-PSK



Obrázek 3.13: Konstelační diagram Q-PSK [13]

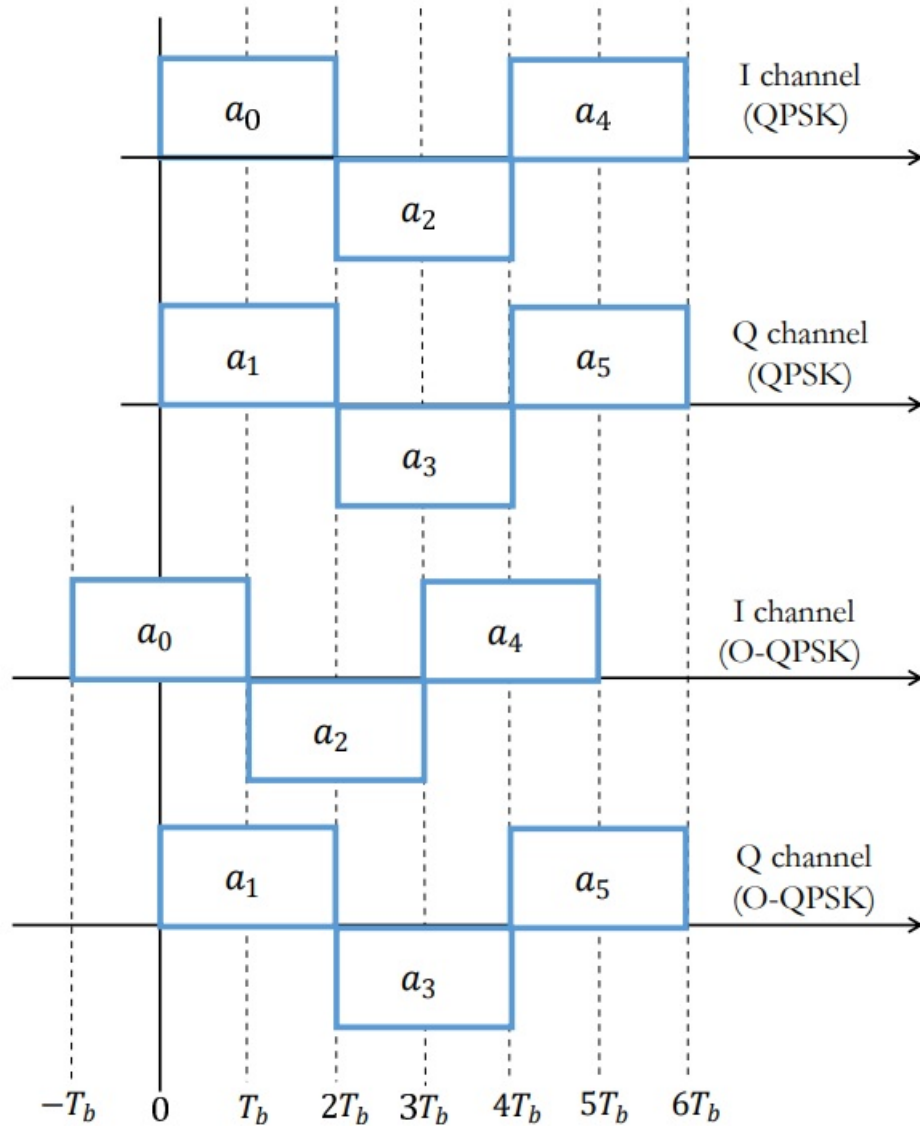
Obrázek (3.14) zobrazuje 8-PSK a její vlastnosti. První graf zobrazuje změnu IQ (In-phase, Quadrature) složek pro data 0 až 7. Druhý graf zobrazuje průběh

reálného signálu složeného z IQ složek. Poslední graf zobrazuje konstelační diagram pro 8-PSK a u každého symbolu je napsaná jeho hodnota.



Obrázek 3.14: 8-PSK

PSK modulace má i jiné varianty, které nejsou definované počtem úrovní. Příkladem může být O-QPSK (offset QPSK), kde jsou I a Q složky posunuté, je to vidět na obrázku (3.15). Nebo jiné varianty jsou definované posunem symbolů po jednotkové kružnici o určitý úhel. Například to je 8PSK s posunem o $\frac{3\pi}{8}$. [15][14]



Obrázek 3.15: O-QPSK [14]

3.6 QAM

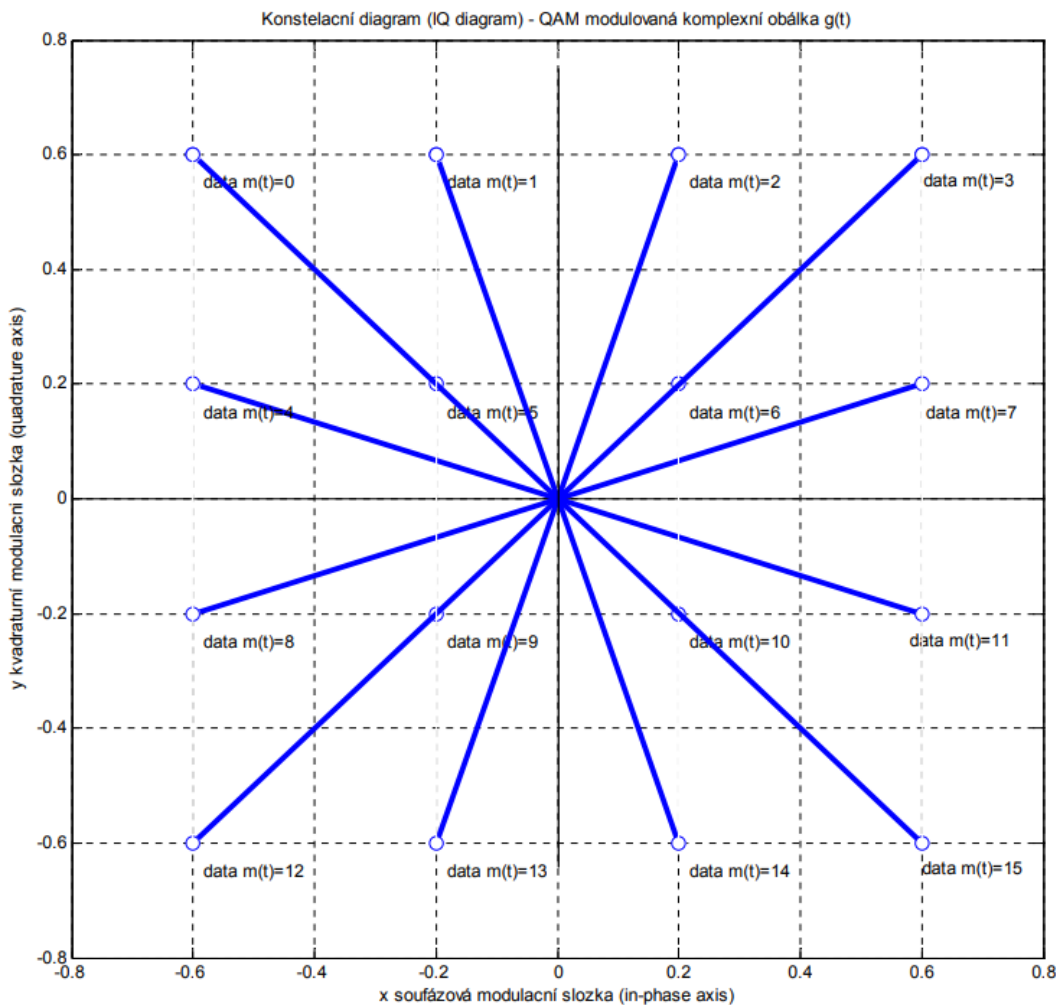
Plný název QAM je Quadrature Amplitude Modulation. QAM patří mezi digitální modulace. Princip QAM je v podstatě kombinace ASK modulace a PSK modulace.

QAM využívá změny velikosti a fáze vektoru v konstelačním diagramu, respektive změnu amplitudy a fáze. Jako ASK, FSK a PSK je QAM tvořena množinou kanálových symbolů. Tato množina je symetrická a ekvidistantní. Množina může

být definována vztahem [13]

$$d_n = q_n \in \left\{ d^{(stav)} \right\}_{stav=1}^{stav=MAX_STAV} \subset \left\{ d^{(stav_1, stav_2)} = (2 * stav_1 - MAX_STAV_1 - 1) + j(2 * stav_2 - MAX_STAV_2 - 1) \right\}_{stav_1=1, stav_2=1}^{stav_1=MAX_STAV_1, stav_2=MAX_STAV_2} \quad (3.24)$$

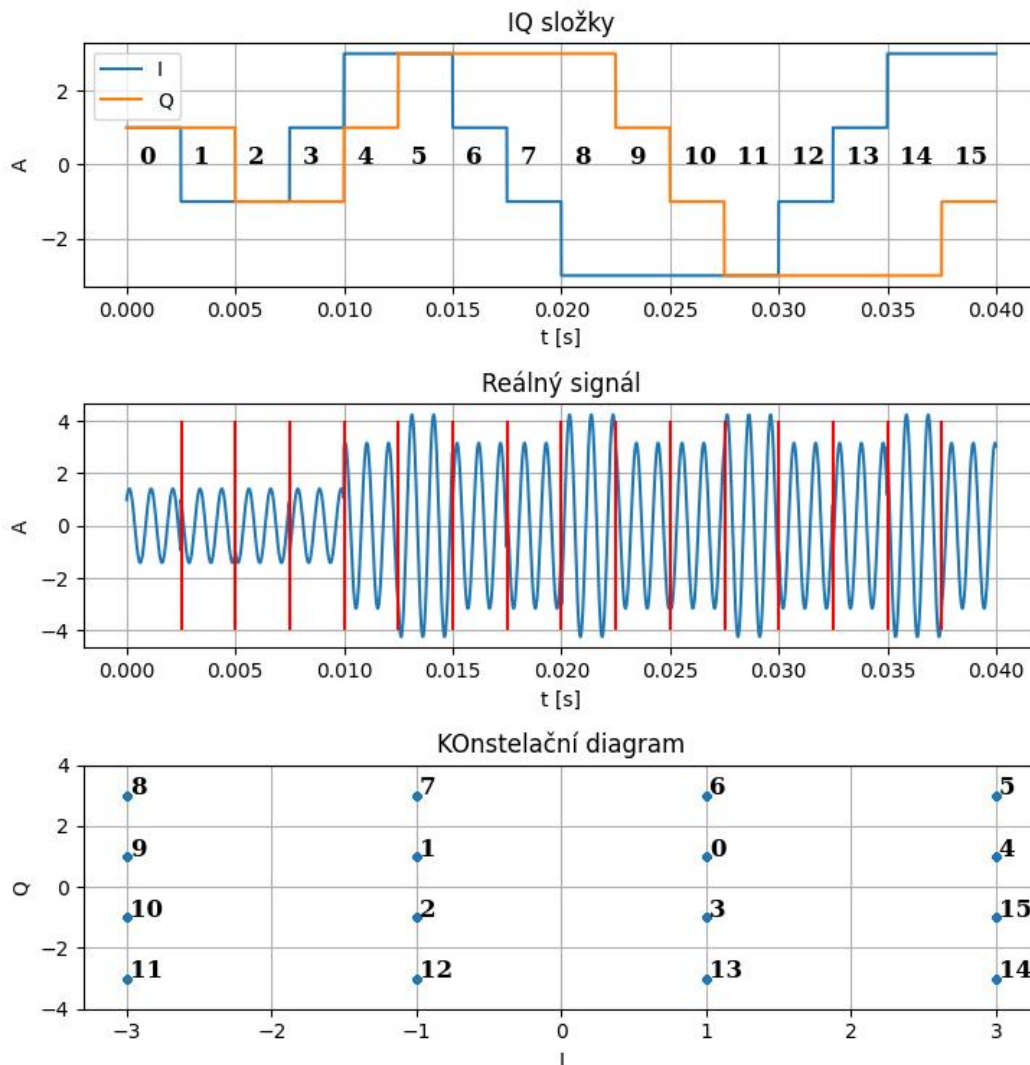
Kanálové symboly jsou q_n a datové symboly jsou d_n . U některých variant QAM jsou některé krajní kanálové symboly nepoužívány. Výsledný konstelační diagram pro 16-QAM je na obrázku (3.16).



Obrázek 3.16: Konstelační diagram 16-QAM [13]

Symboly v konstelačním diagram jsou v pravoúhlém rastru, jak lze vidět na obrázku (3.16). Kvůli rozmístění mají symboly různé moduly vektoru, a také mají různé fáze.

Obrázek (3.17) zobrazuje princip QAM respektive princip 16-QAM. Na obrázku jsou tři grafy. První graf od shora jsou IQ složky, která odpovídají datům (0 až 15). Druhý graf je výsledný modulovaný průběh. Třetí graf je konstelační diagram pro 16-QAM, která je na tomto obrázku.



Obrázek 3.17: Časový průběh 16-QAM

K QAM existuje podobná modulace, kterou je APSK. APSK modulace kombinuje modulace ASK a PSK. Rozdíl mezi QAM a APSK je v rozmístění symbolů v konstelačním diagramu. APSK má symboly rozmístěné na kružnici, tedy množina kanálových symbolů je kruhová a symetrická.

4 Program

V této kapitole jsou popsány výsledné programy a problémy, které se objevily během tvorby programů. Programy jsou dva. Jeden pracuje s SDR a umožňuje vysílat a přijímat signál. Ten se nazývá PySDR. Druhý program je o zobrazování, vysvětlování a parametrizaci modulací. Tento program se jmenuje PyModulace.

Pro základ GUI (graphical user interface) u obou programů a vizualizaci, jak bude vypadat, jsem použil program Qt Designer. Qt Designer umožňuje navrhnout GUI a to převést do kódu. Výsledný kód pro vizualizaci využívá knihovnu PyQt5. Celé GUI v Qt Designeru není dělané, pouze začátek pro vizualizace a rozvržení, kde co bude.

Jak PySDR tak i PyModulace mají pro každý prvek tooltip, protože tyto programy budou studenti používat jednou možná dvakrát, tak ať je pro ně jednoduché zjistit, co jednotlivý prvek dělá. Celkově jsem se snažil vytvořit GUI, které bude lehké na pochopení a půjde se v něm jednoduše zorientovat.

GUI pro oba programy jsou responzibilní. Responzibility jsem docílil nestingem layoutů. Využití více layoutů je nejrozumnější cesta pro responzibilní design, jinak bych veškeré přepočítávání a kontrolu musel vytvořit sám. Toto je už vytvořené a vyzkoušené mnohými uživateli knihovny *PyQt5*. Metoda má například už vyřešenou minimální velikost prvku, aby byl stále použitelný. Jsou tři layouty, ze kterých jsem vybíral, a to horizontální (*QHBoxLayout*), vertikální (*QVBoxLayout*) a mřížkový (*QGridLayout*). Z názvů je patrné, který typ layoutu je na co. Vkládání layoutu do layoutu jsem nakonec vytvořil responzibilní GUI pro oba programy. Tyto GUI jsou vidět na obrázcích (4.6) a (4.13).

Jak kombinování layoutů funguje je v ukázce kódu (4.1), kde je začátek a konec. Jsou tři hlavní layouty vnější, levý (pro ovládací prvky) a pravý (pro zobrazení). V levém layoutu se ještě objevuje mřížkový layout, pokud má být na řádce více prvků. Nakonec se levý a pravý layout vloží do vnějšího a ten se vloží do *centralwidget*. U levého a pravého layoutu je nastavená hodnota roztažnosti, to znamená, v jakém poměru zaberou prostor.

```
1 outerLayout = QHBoxLayout()
2 leftLayout = QVBoxLayout()
3 rightLayout = QVBoxLayout()
4
5 leftLayout.addWidget(self.lModulace)
6 leftLayout_0 = QGridLayout()
7 leftLayout_0.addWidget(self.cBmodulace,0,0)
```

```

8 | leftLayout_0.setColumnStretch(1,50)
9 | leftLayout.addLayout(leftLayout_0)
10 | leftLayout.addSpacing(20)
11 | leftLayout.addWidget(self.label1)
12 | leftLayout_1 = QGridLayout()
13 | leftLayout_1.addWidget(self.hSfrekSign,0,0,1,1)
14 | leftLayout_1.addWidget(self.sBfrekSign,0,1,1,1)
15 | leftLayout_1.addWidget(self.lFrekSig,0,2,1,1)
16 | leftLayout.addLayout(leftLayout_1)
17 | .
18 | .
19 | .
20 | outerLayout.addLayout(leftLayout,1)
21 | outerLayout.addLayout(rightLayout,5)
22 |
23 | self.centralwidget.setLayout(outerLayout)

```

Zdrojový kód 4.1: Responzivní designe ukázka

4.1 Knihovny

V programu používám několik knihoven. Pro matematické operace používám knihovnu *numpy*. Pro zobrazování grafů v PyModulace používám *matplotlib*, vizuálně je podobná zobrazování v Matlabu. Pro GUI používám *PyQt5* a k té je spojená i knihovna *sys*. K zobrazování v reálném čase v PySDR používám *pyqtgraph*. Ke zpracování signálu používám knihovnu *scipy* a *commpy*. Pro paralelizaci programu jsem použil knihovnu *threading*. Všechny tyto knihovny jsou velmi používané a známé, takže jsou i otestované.

Pro samotné SDR je potřeba knihovna *adi*, která umožňuje pracovat se zařízeními od firmy Analog Devices, dále je potřeba mít pro knihovnu *adi* nainstalováno *libio*. Knihovna *libio* je také od firmy Analog devices. Knihovna *libio* je interface pro I/O zařízení.

Pro modulaci a demodulaci jsem si udělal vlastní knihovnu. Tato knihovna má na starost zadaná data, respektive modulační signál, převést na modulovaný signál podle zadaných parametrů. Také umožňuje zachycený signál demodulovat a identifikovat rámce s daty, vyhodnotit ho a vrátit obsah rámců. Z této knihovny je možné používat všechny podprogramy samostatně.

V PySDR i PyModulace používám paralelizaci. Paralelizaci používám k urychlení programu, protože jinak je GUI zpomalený. Zpomalení bylo viditelné hlavně v PySDR, kde k výpočtům dochází často, když je zapnutý příjem a trvá znatelně dlouho než proběhnou všechny výpočty a než dojde k vyhodnocení z GUI. Do vláken jsem dal modulace, demodulace a výpočty spojené se zobrazením přijatého signálu.

Nejnovější verzi obou programů je možné najít na GitHub a Pypi. [16][17] Je možné je najít pod názvem PyModulationPlutoSDR. Protože jsem dal programy

na Pypi, vyřešil jsem distribuci na počítače ve škole, teď stačí jen nainstalovat Python, stáhnou balíček a spustit. Všechny nutné knihovny se automaticky nainstalují i s mým balíčkem.

4.2 Modulace

Každá modulace je reprezentována dvěma funkcemi. Jedna funkce se používá v PySDR a druhá v PyModulace. Jedna funkce je zaměřená pro vysílání a druhá na zobrazování modulace. Pro modulaci ASK používám rekurzivní algoritmus pro určení symbolu odpovídající zadané hodnotě.

Kód níže ukazuje funkci *ASK*, která odpovídá modulaci ASK a rekurzivní funkci *PokracovaniASK*, která určí symbol pro zadanou hodnotu. Nejdříve se ve funkci *ASK* načtou parametry jako počet symbolů, jakým způsobem jsou data generovány nebo jakou polovinu bude ASK používat. Po parametrizaci je samotná modulace. Na konci funkce *ASK* je převod z IQ vzorkování na reálný signál, který bude vysíláný SDR. Tímto postupem jsou dělány všechny modulace.

Samotný rekurzivní algoritmus vypočítává amplitudu symbolu. Algoritmus je pododstavcem (4.2). Používá se, když ASK využívá celou rovinu. Algoritmus neustále odečítá od zadaného čísla dvojkou, dokud výsledek není jedna nebo nula. Od jedničky nebo od nuly odečte polovinu, po odečtení jsou tyto čísla stejně vzdáleny od středu, a je tedy jasné, jestli patří do pravé nebo levé poloroviny. Tento výsledek se vynásobí dvojnásobným počtem volání funkce *PokraovnASK* plus 2 respektive $k + 2$, kde k je dvojnásobný počet volání, proto se vždy ke k přičítá dva.

```
1 def PokracovaniASK(x,n,k): #Vypocitava hodnoty zesileni pro 4-↔
   ASK a vetsi
2   x = x-2 # Snizeni na rozsah 0 - 1
3   if x <= 1:
4       y = (x - 0.5) * (k+2)
5   else:
6       k = k+2
7       y = PokracovaniASK(x,n,k)
8   return y
9
10 def ASK():
11     n = int(self.cBnbitmodulace.currentText())
12
13     """Vyber tvorby sekvence"""
14     if self.rBgenSekv.isChecked():
15         xInt = NahodSymboly(n,2*n) # Pocet cisel 2x vetsi nez ↔
           je rad, aby se zabezpecil dostatecny rozsah hodnot
16         delka = 2 * n * self.opakovani # Delka generovaného ↔
           signálu
17     if self.rBrostSekv.isChecked():
```

```

18     xInt = np.arange(0,n)
19     delka = n * self.opakovani # Delka generovaného signalu
20     if self.rBvlSekv.isChecked():
21         xInt = NaCislo(n)
22         delka = len(xInt) * self.opakovani # Delka generovaneho↔
           signalu
23
24     """Vyber poloroviny"""
25     if self.cbpolorovina.currentText() == "Celá rovina": ↔
           polorovina = 0
26     if self.cbpolorovina.currentText() == "Pravá polorovina": ↔
           polorovina = 1
27     if self.cbpolorovina.currentText() == "Levá polorovina": ↔
           polorovina = 2
28
29     """Generovani amplitud"""
30     xAmp = np.zeros(len(xInt), dtype=np.float64)
31     if polorovina == 0: # Podle toho jaka se pouziva ↔
           polorovina se vypocitaji aplmlicity pro symboly
32         for i in np.arange(len(xInt)):
33             if xInt[i] <= 1: # Vypocte aplitudu pro cisla 0 a 1
34                 xAmp[i] = (xInt[i] - 0.5) * 2
35             else: # čVypocte aplitudu pro čísla 2 a více
36                 k = 2
37                 xAmp[i] = PokracovaniASK(xInt[i],n,k) # ↔
           Rekurzivni vypocet aplitud pro symboly
38     elif polorovina == 1:
39         xAmp = xInt / (n - 1) + 0.5
40     elif polorovina == 2:
41         xAmp = xInt / (n - 1) - 1.5
42     xSymbol = xAmp * (np.cos(0) + 1j * np.sin(0)) # Vytvoceni ↔
           komplexnich čísel
43     samples = np.repeat(xSymbol, self.opakovani)
44     out = IQnaReal(samples,len(samples), np.arange(delka) / ↔
           self.sample_rate)

```

Zdrojový kód 4.2: Rekurzivní algoritmus

Jinak *QAM* používá pro určení symbolu *if a case* (ukázka kódu 4.3), *PSK* dělí 360° podle řádu modulace a z vypočteného úhlu vypočítá hodnotu symbolu (ukázka kódu 4.4), *FSK* si spočítá frekvenci k zadané hodnotě (ukázka kódu 4.5), *AM* a *FM* fungují na matematickém popisu těchto modulací (ukázka kódu 4.6 a 4.7).

```

1 match n:
2     case 4:
3         i = 0
4         for a in xInt:
5             match a:
6                 case 0:
7                     xSymbol[i] = 1 + 1j
8                 case 1:
9                     xSymbol[i] = -1 + 1j
10                case 2:
11                    xSymbol[i] = -1 - 1j
12                case 3:
13                    xSymbol[i] = 1 - 1j
14            i += 1

```

Zdrojový kód 4.3: QAM

```

1 if n == 4:
2     xDegrees = xInt * 360 / 4.0 + 45 # 45, 135, 225, 315 ←
3     stupnu
4 else:
5     xDegrees = xInt * 360 / n
6 xRadians = xDegrees * np.pi / 180.0 # prevod na radiany
7 if n == 2:
8     xSymbol = np.cos(xRadians) + 1j * np.floor(np.sin(xRadians)←
9         )) # vytvori komplexni symbol
10 else:
11     xSymbol = np.cos(xRadians) + 1j * np.sin(xRadians) # ←
12     Vytvori komplexni symbol

```

Zdrojový kód 4.4: PSK

```

1 f = (carrierFreq + Delta_f * xInt - (n * Delta_f)/2) + Delta_f←
2     /2
3 delta_phi = f * np.pi / (self.sample_rate / 2.0)
4 phi = np.cumsum(delta_phi)
5 samples = np.cos(phi) + 1j*np.sin(phi)

```

Zdrojový kód 4.5: FSK

```

1 samples = modulation_index * cosModul * cosCarrier + ←
2     cosCarrier

```

Zdrojový kód 4.6: AM

```

1 samples = np.cos(2*np.pi*carrierFreq*t + D_f*np.sin(2*np.pi*←
      modulatorFreq*t))

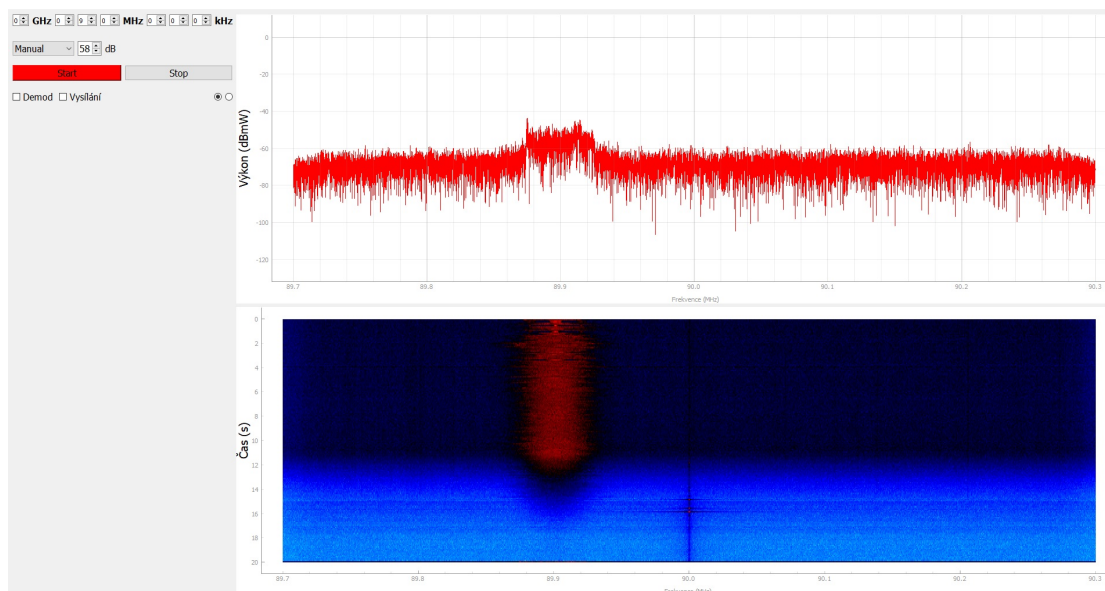
```

Zdrojový kód 4.7: FM

U FSK jsem musel vyřešit plynulý přechod mezi rozdílnými frekvencemi, protože při pouhé okamžité změně frekvence byly v průběhu skokové změny. Tento problém jsem vyřešil kumulativní sumou, která zařídila plynulý přechod, který je vidět na obrázku (3.10) a výsledný kód je vidět v ukázce (4.5). Tato varianta FSK se označuje CPFSK (kapitola 3.4).

4.3 PySDR

Program umožňuje příjem a vysílání signálu. Zachycený signál, pokud se nezapne demodulace, se pouze zobrazí ve frekvenčním spektru a v tzv. waterfallu. Waterfall je spektrogram, kterému se pravidelně obnovují data. Frekvenční spektrum a waterfall je možné vidět na obrázcích (4.1) a (4.6).



Obrázek 4.1: Frekvenční spektrum a waterfall v PySDR

Pro spuštění přijímání je potřeba zmáčknout tlačítko. Pro příjem je možné nastavit několik parametrů, například lze nastavit střední frekvenci nebo zesílení. Zesílení má tři módy: slow-attack, fast-attack a manuální. Střední frekvenci lze měnit během přijímání a zesílení lze měnit během vysílání. Pro ukončení přijímání stačí vypnout program nebo zmáčknout tlačítko „Stop”, které je vidět na obrázku (4.7).

Jedním z problémů, které jsem řešil v tomto programu bylo pravidelné zobrazování dat, které nesmělo výrazně ovlivnit funkci GUI. Na začátku, když se zapl příjem a s ním i pravidelné vykreslování dat, GUI fungovalo sekaně. Jako první jsem přešel z knihovny *matplotlib*, kterou jsem používal na zobrazování dat, na knihovnu

pyqtgraph, která měla být méně náročná na výpočty. Knihovna *pyqtgraph* zlepšila rychlost GUI, ale ne dostatečně. Když nepomohla změna knihovny, tak jsem začal používat paralelizaci, která pomohla, ale přinesla jiný problém, náhodně se zasekl waterfall a nevykresloval nová data. Když jsem zkoumal tento problém objevil jsem dvě možná řešení. První řešení je nevykreslovat data v jiném než v hlavním vlákně a druhé řešení je specifická funkce knihovny *pyqtgraph*, která vykreslí data v jiném vlákně, ale k této funkci jsem toho moc nenašel, takže jsem zkusil první možnost. První možnost vyřešila sekání waterfallu, ale zpomalila GUI. Zpomalení GUI jsem vyřešil správnou volbou rozlišení waterfallu a správnou volbou času, který je schopen zobrazit. Nyní vliv waterfallu zůstal, ale je srovnatelný s tím jaký byl, když jsem vykresloval data ve vedlejších vlákně.

4.3.1 Vysílání

Pokud se zvolí vysílání, je možné vybrat modulaci a nastavit vybrané parametry. U digitálních modulací je možné nastavit počet kvantizačních úrovní nebo tvorbu vysílaných dat. Jsou tři metody jak vytvořit vysílaná data. První metoda je náhodné vygenerování. Náhodně vygenerovaná sekvence má dvojnásobnou délku než je počet kvantizačních úrovní. Druhou metodou je vlastní sekvence, kde je možné napsat vlastní hodnoty respektive text. Třetí možností pro vygenerování dat je rostoucí sekvence. Rostoucí sekvence pokryje všechny možné hodnoty pro danou digitální modulaci. V obou variantách modulací lze nastavit atenuator, který má jako základní hodnotu -30 dB.

Modulační signál lze vytvořit zadáním parametrů do tabulky. Vytvoření modulačního signálu využívá principu Fourierovy řady, do tabulky se totiž zadává frekvence, amplituda a fáze. Po kompletním zadání celého řádku se vypočítá a zobrazí modulační signál, který byl vytvořen ze všech hodnot v tabulce. Touto metodou lze vytvořit kompletně vlastní průběh modulačního signálu. Počet řádků v tabulce je patnáct, tento počet je dostatečný pro vytvoření specifického signálu.

Pro odstartování vysílání je potřeba zmáčknout tlačítko. Během vysílání lze měnit parametry modulace, ale k načtení změněných parametrů dojde až po opětovném zmáčknutí tlačítka pro odstartování vysílání.

Pro vysílání je potřeba použít objektu SDR, který se vytvoří při zapnutí přijímání, proto je potřeba si dát pozor na to, kdyby objekt nebyl vytvořený respektive SDR nebylo připojeno. Pokud objekt není vytvořený nebo dojde k jiné chybě, tak dojde k zachycení této chyby a dojde k otevření okna s možnou příčinou. Toto okno je vidět na obrázku (4.12). Nejdříve dojde k nastavení parametrů a následně dojde k nahrání dat do SDR k vysílání. Toto je v ukázce kódu (4.8).

```

1 try:
2     self.sdr.tx_destroy_buffer()
3     self.sdr.rx_destroy_buffer()
4     self.sdr.tx_rf_bandwidth = int(self.sample_rate)
5     self.sdr.tx_lo = int(self.center_freq)
6
7     # Vysilani
8     self.sdr.tx_cyclic_buffer = True # Povoleni cyklickeho ↔
9     vysilani
10    self.sdr.tx(self.tx_samples) # start vysilani
11 except:
12     self.dlg = QMessageBox(self.centralwidget)
13     self.dlg.setIcon(QMessageBox.Critical)
14     self.dlg.setWindowTitle("Problém")
15     # self.dlg.setInformativeText('More information')
16     self.dlg.setText("SDR není řpijeno nebo se nelze ↔
17     řpijit.")
18     self.dlg.exec()
19 else:
20     self.tx = True
21     self.pBstartVysilani.setStyleSheet("background-color: ↔
22     green") # Nastaveni barvy tlacitka pro indikaci

```

Zdrojový kód 4.8: Vysílání

4.3.2 Příjem

Po zmáčknutí tlačítka pro příjem dojde k volání funkce *cStart*, která vytvoří objekt pro SDR a provede jeho nastavení. Je i ošetřeno, pokud by SDR nebylo připojeno a nebylo by možné vytvořit objekt pro SDR. Pokud není připojené SDR a zmáčkne se tlačítko pro příjem otevře se okno, které uživateli zdělí, že SDR není připojeno. Toto okno je vidět na obrázku (4.12). Jednotlivé kroky nastavení jsou explicitně pojmenované. Toto popisuje ukázkou kódu (4.9).


```

1 def cStart():
2     try:
3         self.sdr = adi.Pluto("ip:192.168.2.1") # Vytvori objekt↔
4             SDR, který je přímo spojený s reálným SDR
5     except:
6         self.dlg = QMessageBox(self.centralwidget)
7         self.dlg.setIcon(QMessageBox.Critical)
8         self.dlg.setWindowTitle("Problém")
9         self.dlg.setText("SDR není řpipojeno nebo se nelze ↔
10             řpipojit.")
11         self.dlg.exec()
12     else:
13         self.provoz = True
14         self.sdr.sample_rate = int(self.sample_rate)
15         self.sdr.rx_lo = int(center_freq)
16         self.sdr.rx_rf_bandwidth = int(self.sample_rate)
17         self.sdr.rx_buffer_size = self.num_samps
18         self.sdr.gain_control_mode_chan0 = "slow_attack"
19         self.sdr.rx_destroy_buffer() # Pro bezpecnost, pokud by↔
20             tam neco bylo
21
22         ZmenaZesileniPrijem()
23         self.timer.start()
24         self.pbstart.setStyleSheet("background-color: green") #↔
25             Nastaveni barvy tlacitka pro indikaci

```

Zdrojový kód 4.9: Vytvoření objektu pro SDR

Jak jsem již zmínil, tak pro zpracování přijatých dat a zobrazení používám threading. V PySDR při uběhnutí stanoveného času na časovači (*timer1*) dojde k zavolání funkce *ZobrazeniFFT*. Tato funkce přebere data z SDR, vyvolá thread, který použije funkci *Vlakno*, aktivuje thread a zobrazí FFT (fast Fourier transform). Zobrazení FFT mimo thread je, protože zobrazení FFT a waterfallu způsobovalo problémy. Funkce *Vlakno* vypočte FFT, hodnoty nahraje do veřejných proměnných a nakonec provede zobrazení waterfallu. Toto je popis ukázky kódu (4.10).

```

1 def Vlakno(self):
2     mag_samples, f1 = self.VypocetFFT(self.rx_samples, self.↔
3         num_samps)
4
5     # Predani promennych do tridy
6     self.x = f1
7     self.y = mag_samples
8
9     # Odebrani jedne rady dat a nahrani nove rady

```

```

9     self.img_array = np.roll(self.img_array, 1, 0) # Posunuti ↵
        o jedno
10    self.img_array[-1:] = self.y
11
12    # Nastaveni widgetu waterfallu
13    freq = np.arange(self.sample_rate / -2, self.sample_rate / ↵
        2, self.sample_rate / self.num_samps) + self.↵
        center_freq
14    image_height = abs(0-20)
15    image_width = abs(freq[0]-freq[-1])
16    self.img.setRect(QRectF(freq[0], 0., image_width, ↵
        image_height))
17
18    def ZobrazeniFFT(self):
19        try:
20            self.rx_samples = self.sdr.rx() # Vzorky z buffru
21        except:
22            self.dlg = QMessageBox(self.centralwidget)
23            self.dlg.setIcon(QMessageBox.Critical)
24            self.dlg.setWindowTitle("Problém")
25            # self.dlg.setInformativeText('More information')
26            self.dlg.setText("SDR není řpijeno nebo se nelze ↵
                řpijit.")
27            self.dlg.exec()
28        else:
29            if self.rBzobrazeni1.isChecked(): self.img.setImage(np.↵
                transpose(self.img_array), autoLevels=False)
30            self.data_line.setData(self.x, self.y) # Update ↵
                zobrazeni FFT
31
32            # řVytvoení vlákna
33            self.t1 = threading.Thread(target=self.Vlakno) # ↵
                Vytvoreni vlakna
34            self.t1.start() # Odstartovani vlakna pro vypocet a ↵
                zobrazeni vysledku

```

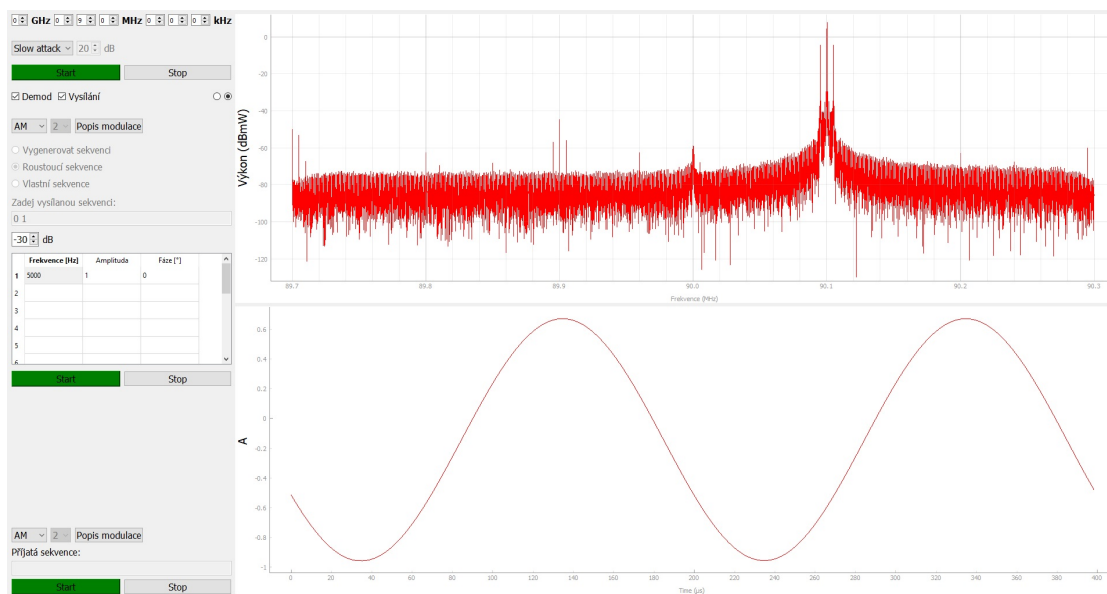
Zdrojový kód 4.10: Paralelizace

4.3.3 Demodulace

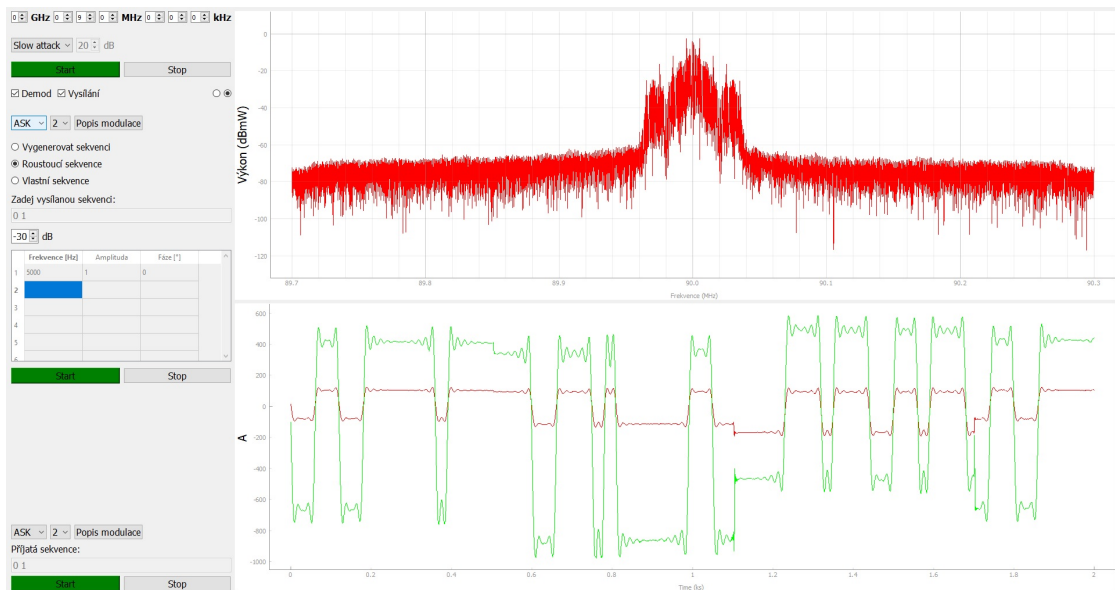
Pokud se zvolí demodulace, je možné vybrat modulaci a nastavit vybrané parametry. U digitálních modulací je možné nastavit počet kvantizačních úrovní. Analogové modulace nemají nastavitelné parametry, protože předpokládám, že je v modulovaném signálu i maximální hodnota modulačního signálu. Tím, že výsledek demodulace AM a FM je pouze zobrazení signálu a cílem pro studenty je dostat stejný průběh

signálu jako byl vyslán, tak se neřeší amplituda. A navíc jeden blok vzorků z SDR obsahuje několik period modulačního signálu, a tedy i maximální a minimální hodnotu. A proto AM a FM nemají nastavení promodulovanosti a indexu modulace při vysílání.

Pro odstartování demodulování je potřeba zmáčknout tlačítko. Během demodulování lze měnit parametry modulace, ale k načtení změněných parametrů dojde až po opětovném zmáčknutí tlačítka pro odstartování demodulování. Výsledná sekvence je zobrazovaná jako text. U analogových modulací dochází k zobrazování demodulovaný signál na místě pro waterfall (obrázek 4.2), ale toto musí být potvrzeno přes vybrané zobrazení 2 (kapitola 4.3.4). Na stejném místě je možné zobrazit I a Q složku přijatého signálu, je to možné vidět na obrázku (4.3). Toto je možné jen, když je zapnutá demodulace jiná než AM nebo FM a je aktivované zobrazení 2.



Obrázek 4.2: Zobrazení demodulovaného signálu AM nebo FM



Obrázek 4.3: Zobrazení zachyceného signálu

Například pro demodulaci AM jsem nejdříve zkusil obálkový detektor, který vrátil sice obálku modulovaného signálu, ale byla velmi zvlněná. Po dlouhém hledání a zkoušení jsem našel řešení v Hilbertově transformaci, díky které bylo snadné získat hladkou obálku pomocí pár řádků kódu (ukázka kódu 4.11). V Hilbertově transformaci jsem našel i způsob jak demodulovat FM a FSK, protože přidáním několika dalších řádků jsem znal frekvenci signálu v každém okamžiku (ukázka kódu 4.12). Hodně času jsem strávil na demodulaci PSK, kde jsem zkoušel demodulovat přes průběh fáze. Nakonec jsem to vyřešil tak, že jsem přišel na to, o kolik jsou symboly otočené oproti originálu, a to vyřešilo problém demodulace PSK. Na začátku práce jsem se snažil všechny digitální demodulace vysílat přijímané frekvenci plus smyšlená nosná, abych získal plynulé průběhy, ale časem jsem zjistil, že to není třeba. Stačilo předat SDR hodnoty IQ složek, tak jak jsou například na obrázku (3.17), ale vyfiltrované, aby se omezila šířka kanálu. Díky tomu stačilo pro demodulaci napsat funkci, která vybere vhodný vzorek pro daný symbol a nastavit správně prahování pro symboly.

```

1 analytic_signal = signal.hilbert(samples)
2 amplitude_envelope = np.abs(analytic_signal)

```

Zdrojový kód 4.11: Obálka signálu

```

1 analytic_signal = signal.hilbert(samples)
2 instantaneous_phase = np.unwrap(np.angle(analytic_signal))
3 instantaneous_frequency = (np.diff(instantaneous_phase) / ←
    (2.0*np.pi) * sample_rate)

```

Zdrojový kód 4.12: Okamžitá frekvence

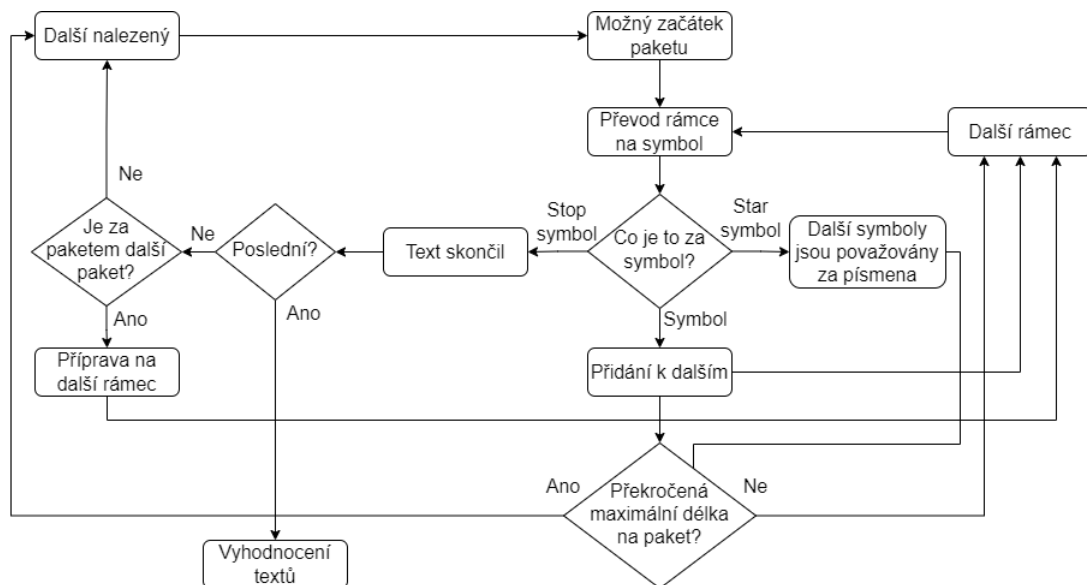
Pro demodulaci AM, FM a FSK je potřeba offset frekvence o -100 kHz . Je to potřeba, protože už při vysílání je také offset použit (100 kHz), jinak by při vysílání bylo potřeba oříznou levé pásmo. Pro vzdělávací účely je lepší nechat frekvenční spektrum modulace netknuté.

Princip demodulace ASK, PSK, QAM, FSK a zpracování demodulovaných dat je možné vidět na obrázku (4.4), kde je diagram programu. Krok „Jakému symbolu vzorek odpovídá“ je demodulace pomocí prahování. Následně se v demodulovaných datech hledá start symbol, který značí začátek paketu. Hledání se provádí pomocí korelace. Možný nalezený paket se převede na text a následně je text vyhodnocený. Jak probíhá převod paketu na text je na obrázku (4.5).



Obrázek 4.4: Diagram demodulace

Dokud není nalezen start symbol, tak nedojde k převedení na text, protože text je za start symbolem. Stop symbol značí konec paketu. Start a stop symboly jsem vybral, aby obsahovaly minimální a maximální hodnotu daného typu modulace, je to potřeba, protože dochází k normalizaci při zpracování. Kdyby v přijatých datech chybělo maximum, tak by došlo ke špatné demodulaci. Po zpracování každého paketu dojde k vyhodnocení textu. Vyhodnocením textu znamená, že se zjistí, jestli všechny texty jsou stejné, případně jaký text je nejčastější. Výsledek je zobrazený v GUI.

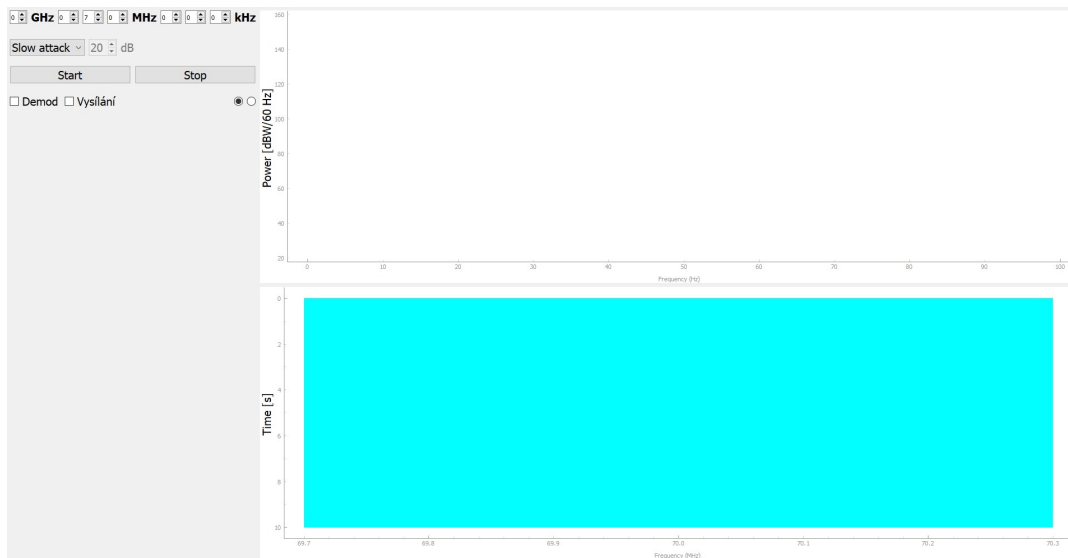


Obrázek 4.5: Diagram zpracování paketu

4.3.4 Popis GUI

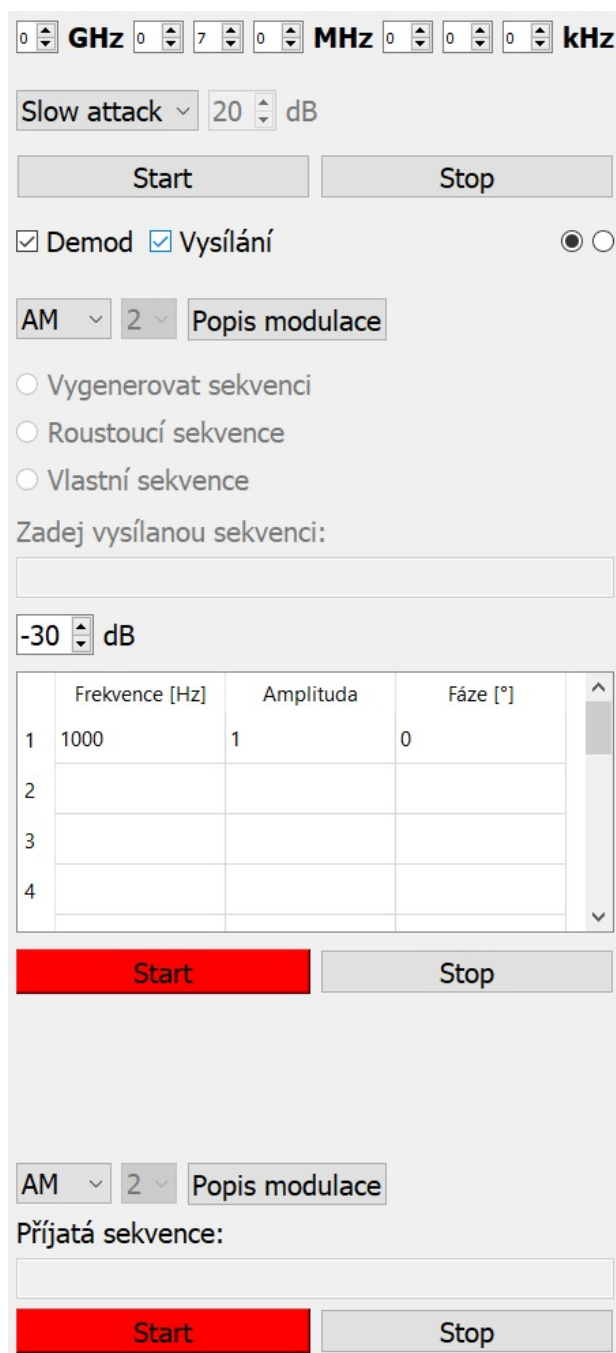
Každý prvek má vlastní tooltip pro jednodušší ovládání. Také je toto GUI responzibilní, takže je jedno, na jak velké obrazovce bude program otevřen, kromě toho je možné si dát program třeba i na půl strany obrazovky a všechny prvky budou viditelné a použitelné.

Na obrázku (4.6) je vidět pohled na celé GUI PySDR. Check box pro vysílání je zaškrtnutý a check box pro demodulaci není (levá strana nahoře), což znamená, že všechny prvky, které jsou spojené s demodulací jsou neviditelné. Jinak řečeno zaškrtnutí check boxu určí jaké prvky budou vidět. Na pravé straně jsou dva typy zobrazení. Jeden typ zobrazení, ten nahoře, je FFT a druhý typ, ten dole, je waterfall.



Obrázek 4.6: Pohled na celé GUI PySDR

Obrázek (4.7) zobrazuje podrobněji levou stranu GUI PySDR. Na obrázku lze vidět všechny prvky, kromě zobrazovacích prvků. Tato část je GUI je rozdělena na dva typy prvků, ty, které jsou vidět neustále a ty, které jsou vidět pouze, když je zaškrtnutý check box. Check boxy ovládají viditelnost prvků pro vysílání a pro demodulaci.

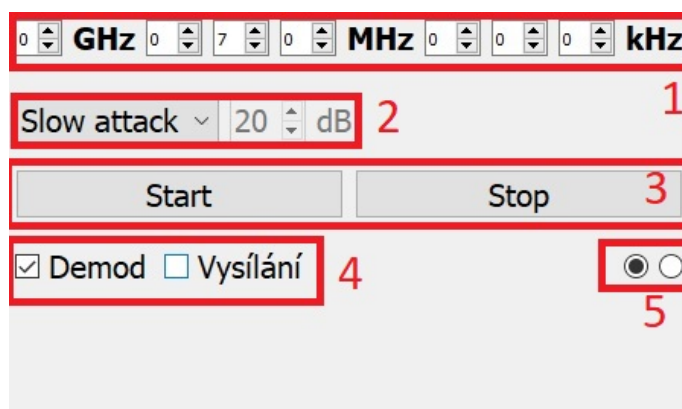


Obrázek 4.7: Pohled na parametrizační část GUI PySDR

Obrázek (4.8) zobrazuje levou stranu GUI PySDR, když není ani jeden z check boxů zaškrtnutý. Popis prvků:

1. Umožňuje nastavit přijímanou frekvenci. Spin boxy jsou propojené takže, když by měla hodnota ve spin boxu přesáhnout deset, tak se k hodnotě spin boxu, který odpovídá vyššímu řádu, přičte jedna a aktuální spin box je nastaven do nuly. Toto platí i opačným směrem.

2. Combo box a spin box patří k sobě. V combo boxu lze vybrat metodu zesílení pro příjem. A spin box je používán jen tehdy, když je v combo boxu nastaveno „Manual”.
3. Funkce tlačítek „Start” a „Stop” je očividný, spouští a vypínají příjem.
4. Check boxy jsou pro vybrání, co uživatel chce dělat, jestli chce vysílat nebo jestli chce demodulovat.
5. Jsou pro výběr, jestli se má zobrazit tvořený signál, který se tvoří v tabulce, která je pro tvorbu modulačního signálu (více v dalším odstavci) nebo waterfall. Pokud se demoduluje, tak rozhodují o tom, jestli se zobrazí waterfall nebo demodulace AM a FM. Také umožňují při demodulaci zobrazit část přijmutého signálu pro modulace ASK, PSK, FSK nebo QAM.



Obrázek 4.8: Nastavení příjmu GUI PyModulace

Obrázek (4.9) zobrazuje prvky, které patří k vysílání. Popis prvků:

1. Combo box pro výběr modulace. Ovlivňuje nejvíce prvků, jestli jsou uzamknuté nebo ne.
2. Combo box, který patří pouze k digitálním modulacím. Nastavuje počet úrovní, respektive počet symbolů.
3. Tlačítko pro zapnutí PyModulace.
4. Jsou pro výběr typu vytvoření sekvence pro vysílání. Patří pouze k digitálním modulacím. Vygenerování sekvence a rostoucí sekvence jsou pouze číselné hodnoty, které reprezentují symbol.
5. Zde se zobrazuje vysílaná sekvence a pokud je zvolená vlastní sekvence, tak se zde zapisuje. Patří pouze k digitálním modulacím.
6. Spin box pro nastavení atenuátoru.

7. Table widget je pro zadávání parametrů pro jedinečný signál pro vysílání, respektive modulační signál. Lze zadat až patnáct prvků pro Fourierovu řadu. Tento prvek patří k analogovým modulacím.
8. Tlačítka pro odstartování nebo zastavení vysílání. Tlačítko „Start” je červené, protože se nevysílá, bylo by zelené, kdyby se vysílalo. Změna barev je pro indikaci, jestli se vysílá. Pro načtení změny parametrů při vysílání je potřeba zmáčknout tlačítko „Start” znovu nebo vypnout a zapnout vysílání.

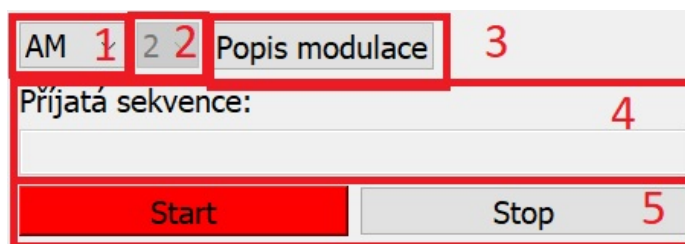


Obrázek 4.9: Nastavení vysílání GUI PyModulace

Obrázek (4.8) zobrazuje část GUI PySDR, která je zodpovědná za demodulaci. Popis prvků:

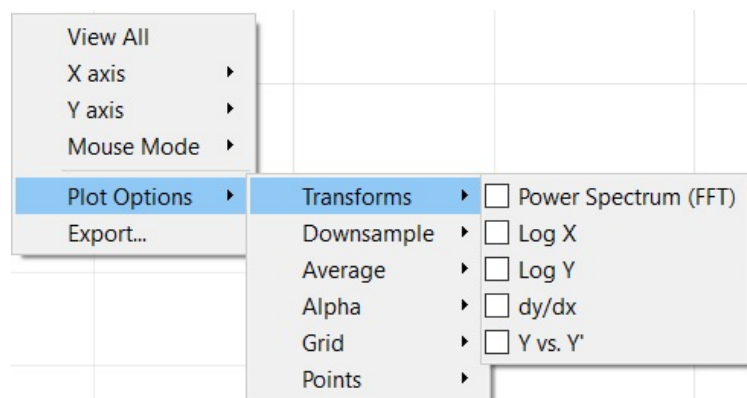
1. Combo box pro výběr demodulace.
2. Combo box, který patří pouze k digitálním modulacím. Nastavuje počet kvantizačních úrovní.
3. Tlačítko pro zapnutí PyModulace.
4. Zde se zobrazí přijatý text, pokud se demoduluje digitální modulace.
5. Tlačítka pro odstartování a zastavení demodulace. Tlačítko „Start” je červené, protože se nedemoduluje, bylo by zelené, kdyby se demodulovalo. Změna barev

je pro indikaci, jestli se demoduluje. Pro načtení změny parametrů při demodulaci je potřeba zmáčknout tlačítko „Start“ znovu nebo vypnout a zapnout demodulaci.



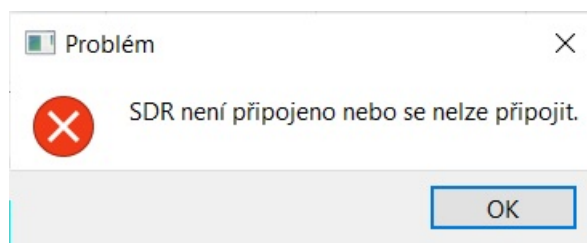
Obrázek 4.10: Nastavení demodulace GUI PyModulace

Pokud se na jeden z grafů, který je vidět na obrázku (4.6), klikne pravým tlačítkem myši, tak se otevře menu pro nastavení zobrazení. Toto menu je na obrázku (4.11). V tomto menu jde nastavit limity os, mřížku, změnit osy na logaritmické nebo upravit vzorkování. Také přes toto menu lze nechat spočítat FFT zobrazovaného signálu, ale tato metoda nezmění osy (pojmenování a jednotky zůstanou stejné).



Obrázek 4.11: Menu pro nastavení zobrazení GUI PyModulace

Součástí GUI jsou i okna s chybovými hláškami. Tyto okna se objeví například, pokud uživatel nebude mít připojené SDR a bude chtít zapnout příjem nebo bude chtít použít 2-QAM, které není definované. Příklad okna s chybovou hláškou je na obrázku (4.12). Tento typ okna je součástí knihovny *PyQt5*.



Obrázek 4.12: Okno s chybovou hláškou GUI PyModulace

4.4 PyModulace

Účel programu je seznámit studenty s modulacemi. V programu mohou měnit parametry modulace, jako počet úrovní, promodulovanost atd. Změny v parametrech se okamžitě promítnou ve dvou grafech. Pro analogové a digitální modulace se typy zobrazení liší. Analogové modulace mají v jednom grafu zobrazený průběh modulačního, modulovaného a nosného signálu a v druhém grafu je FFT těchto tří signálů. Digitální modulace mají tři grafy. Pouze dva jsou v jednu chvíli viditelné, takže se mezi nimi přepíná. Jeden graf je vidět neustále, a to je konstelační diagram. Dva grafy, které se střídají jsou graf průběhu IQ složek a graf výsledného signálu, když se IQ složky dají dohromady.

I PyModulace používá paralelizace. Zde je využit pro výpočty potřebné k vizualizaci, což znamená pro výpočet modulovaného signálu, FFT nebo konstelačního diagramu. Když dojde k změně hodnoty parametru v GUI, dojde k vyvolání funkce *Modulace*. Tato funkce vytvoří a spustí vlákno, které vypočte modulovaný signál dle vybrané modulace viz ukázka kódu (4.13) pod odstavcem.

```
1 def Modulace():
2     if self.cBmodulace.currentText() == "AM":
3         t1 = threading.Thread(target=AM()) # Vytvoreni vlakna
4         t1.start()
5     if self.cBmodulace.currentText() == "FM":
6         t2 = threading.Thread(target=FM()) # Vytvoreni vlakna
7         t2.start()
8     if self.cBmodulace.currentText() == "ASK":
9         t3 = threading.Thread(target=ASK()) # Vytvoreni vlakna
10        t3.start()
11    if self.cBmodulace.currentText() == "FSK":
12        t4 = threading.Thread(target=FSK()) # Vytvoreni vlakna
13        t4.start()
14    if self.cBmodulace.currentText() == "PSK":
15        t5 = threading.Thread(target=PSK()) # Vytvoreni vlakna
16        t5.start()
17    if self.cBmodulace.currentText() == "QAM":
18        t6 = threading.Thread(target=QAM()) # Vytvoreni vlakna
19        t6.start()
```

Zdrojový kód 4.13: Paralelizace v PyModulace

Dostat zobrazení průběhu do GUI je složitější, protože žádný připravený widget nevyužívá knihovnu *matplotlib*. Jak je možné docílit využívání *matplotlib* v GUI ukazuje ukázka kódu (4.14). Je potřeba třída (*MatplotlibWidget*), která do *QWidget* vytvoří grafy a toolbar. Aby v PyModulace došlo k úpravě *QWidget*, tak mám funkci *init_widget*, ta vytvoří objekt ze třídy *MatplotlibWidget* a vloží ho do GUI.

```

1 class MatplotlibWidget(QWidget): # Vytvori 2 grafy
2     def __init__(self, parent=None):
3         super(MatplotlibWidget, self).__init__(parent)
4         self.figure = Figure()
5         self.figure.set_tight_layout([50, 500, 100])
6         self.canvas = Canvas(self.figure)
7
8         self.axis = self.figure.add_subplot(211)
9         self.axis2 = self.figure.add_subplot(212)
10        toolbar = NavigationToolbar(self.canvas, self)
11
12        # spoji graf a toolbar
13        self.layoutvertical = QVBoxLayout(self)
14        self.layoutvertical.addWidget(toolbar)
15        self.layoutvertical.addWidget(self.canvas)
16
17    def init_widget(self): # Inicializace zobrazeni
18        self.matplotlibwidget = MatplotlibWidget()
19        self.layoutvertical = QVBoxLayout(self.zobrazeniModulace)
20        self.layoutvertical.addWidget(self.matplotlibwidget)

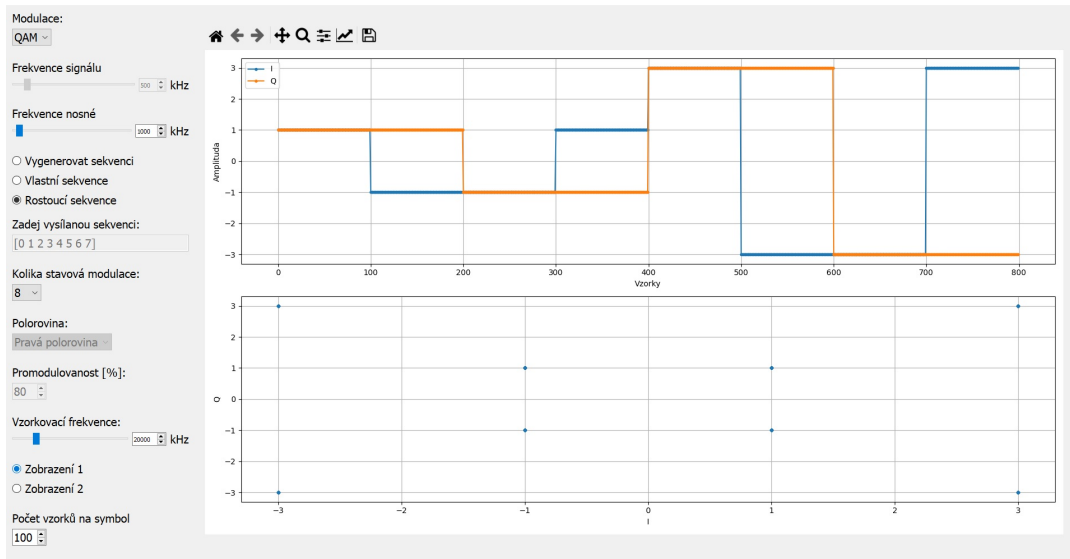
```

Zdrojový kód 4.14: Matplotliv v GUI

4.4.1 Popis GUI

Každý prvek má vlastní tooltip pro jednodušší ovládání. Také je toto GUI responzibilní, takže je jedno na jak velké obrazovce bude program otevřen, kromě toho je možné si dát program třeba i na půl strany obrazovky a všechny prvky budou viditelné a použitelné.

GUI, které jde vidět na obrázku (4.13), je rozděleno podobně jako v PySDR. Na levé straně jsou parametry modulací a na pravé straně je zobrazení. Parametry modulace jsou vždy zobrazeny a jsou pouze blokovány nebo odblokovány podle splnění kritérií. Je to takto uděláno, aby bylo neustále vidět, kolik je různých parametrů, a kdy jsou platné, i když to nejsou všechny, které existují.

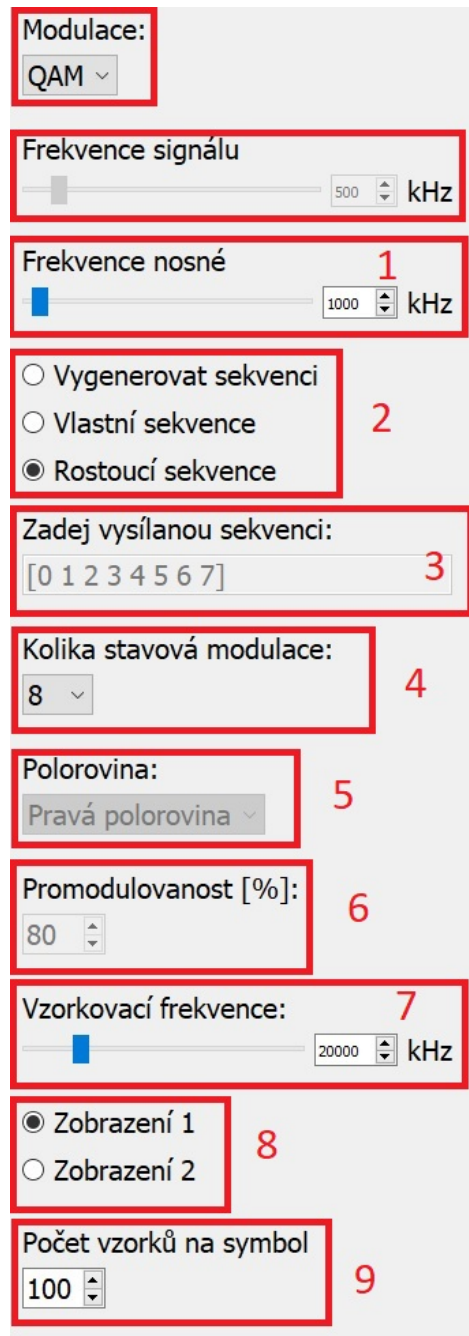


Obrázek 4.13: Pohled na celé GUI PyModulace

Obrázek (4.14) vyobrazuje levou část GUI PyModulace, kde jsou parametry modulací. Všechny prvky jsou neustále viditelné a mění se pouze, pokud je možné je použít. Popis prvků:

1. Combo box pro výběr modulace.
2. Slider bar a spin box pro zadání frekvence modulačního signálu. Kombinace dvou prvků je zde proto, aby bylo možné zadat konkrétní frekvenci, a také je díky tomu možné plynule měnit frekvenci modulačního signálu. Vztahují se pouze k analogovým modulacím.
3. Slider bar a spin box pro zadání frekvence nosné vlny. Důvod pro volbu kombinací těchto prvků je stejná jako u prvku nad ním. Vztahují se k oběma variantám modulací.
4. Jsou pro výběr typu vytvoření sekvence pro vysílání. Patří pouze k digitálními modulacím. Vygenerování sekvence a rostoucí sekvence jsou pouze číselné hodnoty, které reprezentují symbol.
5. Zde se zobrazují „vysílaná“ data. Pokud je vybráno na prvku nad „vlastní sekvence“, tak se sem zapisuje. Můžeme sem zapsat pouze čísla, a pokud číslo je mimo rozsah modulace, tak je smazáno. Zadávaná čísla odpovídají symbolům dané modulace.
6. Combo box, který patří pouze k digitálními modulacím. Nastavuje počet úrovní, respektive počet symbolů.
7. Combo box pro určení jakou část komplexní roviny může modulace ASK používat.

8. Spin box pro nastavení promodulovanosti u AM a pro nastavení modulačního indexu u FM.
9. Slider bar a spin box pro zadání vzorkovací frekvence. Kombinace dvou prvků má stejný důvod jako u těch předchozích. Vztahují se ke všem modulacím.
10. Jsou pro změnu typu zobrazení u digitálních modulacích. Mění horní zobrazení, kde místo IQ složek se zobrazí reálný signál, který je vytvořen z IQ složek.
11. Spin box pro nastavení počtu vzorků na symbol u digitálních modulací a analogových modulací nastavuje počet zobrazených vzorků nosné, modulačního a modulovaného signálu.



Obrázek 4.14: Pohled na parametrizační část GUI PyModulace

Obrázek (4.15) vyobrazuje toolbar, který patří k zobrazování průběhů. Toolbar je nahoře vlevo v oblasti pro zobrazení. Tento toolbar je z knihovny *matplotlib*, kterou používám pro zobrazování. Popis prvků (zleva doprava):

1. Vráťí přiblížení a posuvy v zobrazení na základní pohled.
2. Posuv mezi kroky ve změnách.
3. Pohyb v zobrazení.

4. Přiblížení v zobrazení.
5. Otevře okno pro nastavení layoutu zobrazení, například jaký je odstup od horní hrany nebo od spodní atd.
6. Otevře okno pro nastavení samotného zobrazování, například nastavení os nebo nastavení křivky (její barva, tloušťka atd.).
7. Pro uložení aktuálního zobrazení.



Obrázek 4.15: Pohled na toolbar pro zobrazování GUI PyModule

5 Úloha pro studenty

Práci s programy a seznámení s modulacemi jsem rozdělil do dvou úloh. V jedné úloze se studenti seznámí s modulacemi a v druhé úloze zkusí příjem a vysílání v základních modulacích.

První úloha je v PyModulace. Studenti dostanou několik různých průběhů signálů nebo konstelační diagramy a mají identifikovat modulaci a případně podstatné parametry. K identifikaci mají použít PyModulace, ve kterém jsem i vytvořil průběhy a konstelační digramy, které studenti dostanou na identifikaci. Když udělají identifikaci, mají za úkol se pokusit tyto průběhy nebo konstelační diagramy zrekonstruovat. Během identifikace a rekonstrukce si studenti vyzkouší všechny modulace a jejich parametry, takže zjistí jak se modulace chovají při změně parametrů a jaké jsou charakteristiky dané modulace.

V druhé úloze dělají studenti v PySDR a na spektrálním analyzátoru, ale také potřebují SDR. Vyučující vysílá a studenti mají za úkol najít, na jaké frekvenci vysílá a identifikovat modulaci. Pro nalezení frekvence používají spektrální analyzátor. Identifikace digitální modulace probíhá pomocí vybírání demodulace a sledování řádku, kde se objeví vysílaný text, pokud je to správná demodulace. A identifikace analogové modulace probíhá pomocí sledování zobrazení, kde se ukazuje demodulovaný signál.

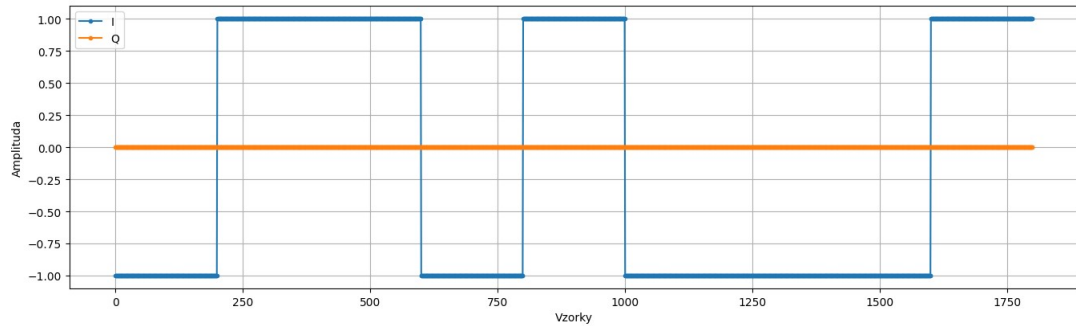
Na konci zadání úloh je, co má být v protokolu a otázky, které mají zodpovědět.

5.1 Řešení úloh

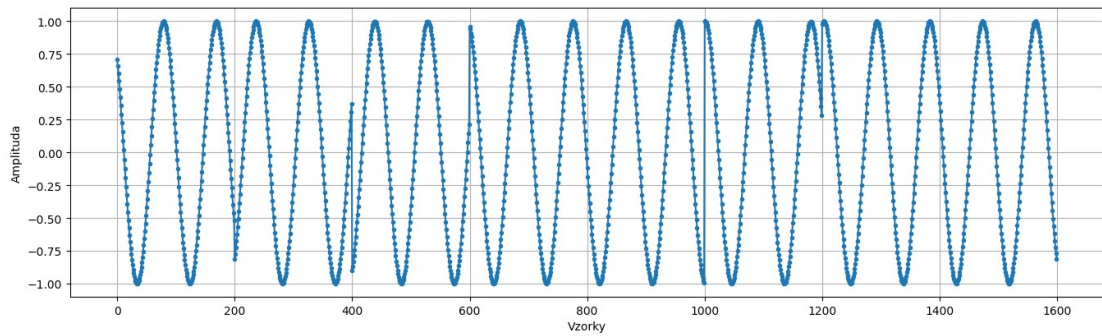
5.1.1 Úloha 1 - rozpoznání modulace a rekonstrukce modulovaného signálu

Obrázek (5.1) je 2-ASK nebo 2-PSK a sekvence je [0 1 1 0 1 0 0 1]. Obrázek (5.2) je 4-PSK nebo 4-QAM a sekvence je [3 2 3 2 2 0 1 1], ale tento obrázek nemusí studenti zrekonstruovat, protože průběh závisí na vzorkovací frekvenci, frekvenci nosné vlny a hlavně nelze jednoduše určit jednotlivé posuny fáze, takže ani nelze zjistit sekvenci. Obrázek (5.3) je konstelační diagram 4-PSK nebo 4-QAM a jediné, co lze říci o sekvenci je, že obsahuje všechny hodnoty. Obrázek (5.4) je 16-QAM a průběh IQ složek je vygenerován rostoucí sekvencí. Obrázek (5.5) je to 4-ASK a je to vygenerováno pomocí rostoucí sekvence. Obrázek (5.6) je FM s modulační frekvencí 300 Hz, nosnou frekvencí 4020 Hz a index 6. Obrázek (5.7) je AM s modulační frekvencí 300 Hz, nosnou frekvencí 1200 Hz a promodulovaností 60 %. Obrázek

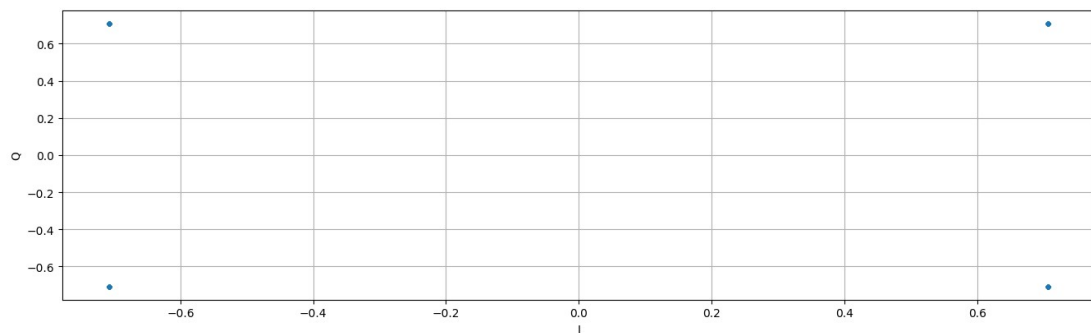
(5.8) je 4-ASK a je to vygenerováno pomocí vlastní sekvence, která je $[3\ 0\ 2\ 1\ 0]$, používá se celá rovina, frekvence nosné je 1 kHz , vzorkovací frekvence je 15 kHz a počet vzorků na symbol je 50.



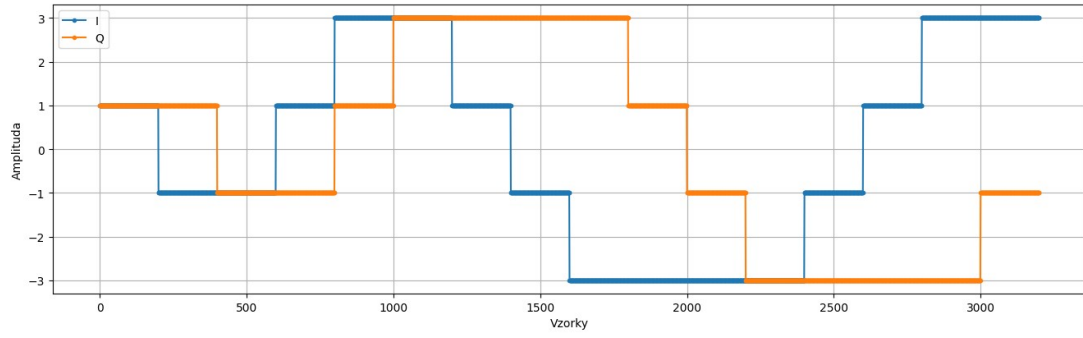
Obrázek 5.1: Průběh 1



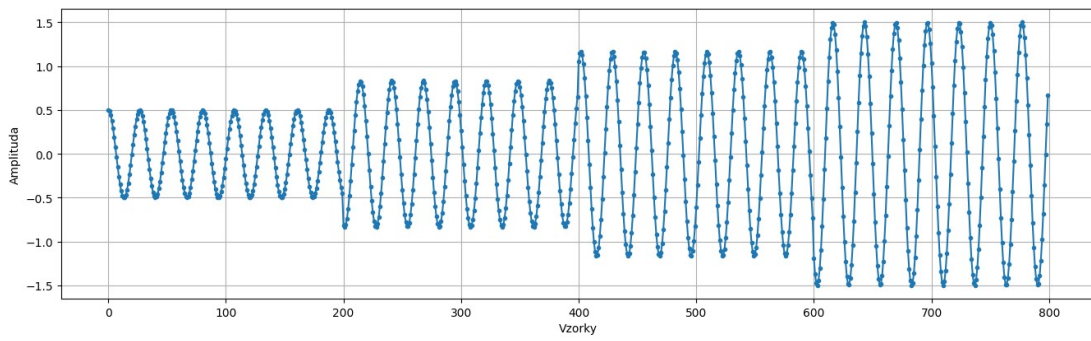
Obrázek 5.2: Průběh 2



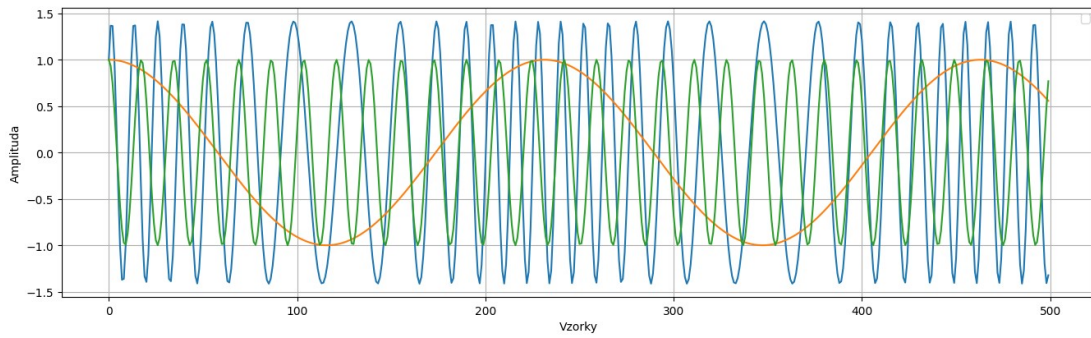
Obrázek 5.3: Průběh 3



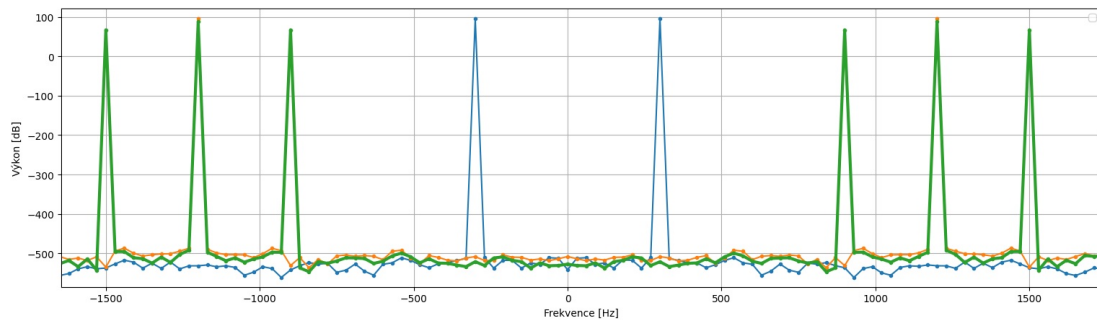
Obrázek 5.4: Průběh 4



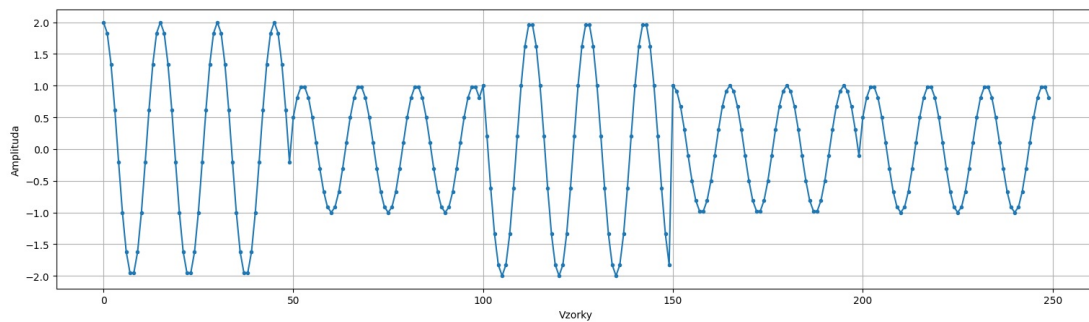
Obrázek 5.5: Průběh 5



Obrázek 5.6: Průběh 6



Obrázek 5.7: Průběh 7



Obrázek 5.8: Průběh 8

5.1.2 Úloha 2 - příjem signálu s neznámou modulací

Druhá úloha je na vyučujícím, co si vybere za modulaci, a co bude vysílat. Může vysílat text (ASK, PSK, FSK a QAM) nebo specifický signál (AM a FM). Studenti si také mohou posílat, co oni chtějí.

5.1.3 Odpovědi na otázky

1. Proč je vysílaný signál vyučujícím silnější než ostatní? - Vysílač je mnohem blíže než ostatní.
2. Jaké jsou rozdíly mezi 2-ASK a 2-PSK? - Pokud 2-ASK využívá celou rovinu a 2-PSK má na reálné ose proti sobě symboly, tak žádný, protože rozdíl mezi symboly je pouze v úhlu. Celkově PSK nemusí mít symetricky rozložené symboly.
3. Jaké jsou rozdíly mezi 4-PSK a 4-QAM? - Pokud jsou definovány symboly, tak jak jsou v PyModulace, není mezi nimi rozdíl. Pokud by byly symboly definovány jinak, a to mohou, tak už by byl mezi nimi rozdíl.
4. Jaké jsou rozdíly mezi AM a ASK? - Princip je stejný, informace ukládají do amplitudy modulovaného signálu. ASK je pro digitální signál a AM je pro analogový signál.

5. Jaké jsou rozdíly mezi FM a FSK? - Je to stejné jako mezi AM a ASK.
6. Proč si myslíte, že se zřídka používají vysokoúrovňové modulace ASK a PSK (16 a více)? - U ASK je problém, že se data nalézají v amplitudě modulovaného signálu a amplituda se velmi lehce a tedy, když je malý rozdíl mezi úrovněmi, tak lehce dojde k zarušení. U PSK je to podobné, mezi úrovněmi je malý rozdíl a tedy je modulace náchylnější na chyby.

6 Závěr

Cíle práce jsem splnil. Vytvořil jsem dva programy v jazyce Python, určené pro výuku základních modulací AM, FM, ASK, FSK, PSK a QAM. Dále jsem vytvořil zadání ke dvěma laboratorním úlohám včetně kontrolních otázek.

Program PySDR (kapitola 4.3) umožňuje s pomocí SDR ADLAM PLUTO (kapitola 2.2) vysílat modulovaný signál a signál i přijímat včetně jeho demodulace. Při vývoji programu jsem musel řešit řadu problémů, např. pomalé GUI při zapnutém příjmu, nebo demodulace signálu. Ve finále se nejvíce osvědčila Hilbertova transformace, pomocí které bylo možno velmi rychle získat frekvenci a obálku modulovaného signálu u modulací AM, FM a FSK. Pro ostatní modulace se při vysílání osvědčilo předat SDR hodnoty IQ složek a pro demodulaci pak napsat funkci, která vybere vhodný vzorek pro daný symbol a vhodné prahování pro symboly.

Program PyModulace (kapitola 4.4) umožňuje pracovat se základními modulacemi podrobněji. V programu je možné měnit parametry modulace a sledovat, jak to ovlivní modulovaný signál, frekvenční spektrum, IQ složky nebo konstelační diagram.

Nejnovější verzi obou programů je možné najít na GitHub a Pypi. [16][17] Lze je najít pod názvem PyModulationPlutoSDR. Protože jsem dal programy na Pypi, vyřešil jsem distribuci na počítače ve škole, teď stačí jen nainstalovat Python, stáhnou balíček a spustit.

První laboratorní úloha využívá program PyModulace, který je zaměřen na pochopení principu jednotlivých modulací s možností měnit všechny potřebné parametry a sledovat jejich vliv na průběh modulovaného signálu.

Druhá laboratorní úloha je zaměřena na praktickou činnost s SDR. Učitel vysílá signál s neznámou modulací na neznámé frekvenci a studenti mají za úkol zjistit všechny parametry signálu, jak pomocí spektrálního analyzátoru, tak pomocí programu PySDR.

Oba programy skýtají ještě možnost rozšíření. U programu PySDR je možné například vysílání signálu rozšířit o kódování signálu tak, aby i při zarušení nedošlo ke ztrátě dat, nebo jej upravit tak, aby bylo možné ovládat SDR i od jiných výrobců. Program PyModulace by bylo možné rozšířit o další typy modulací nebo umožnit nastavení i dalších parametrů již stávajících modulací.

Použitá literatura

- [1] ČEJKA, Jiří. *Laboratorní úlohy pro vysokofrekvenční měření*. Liberec, 2020. Bakalářská práce. Technická univerzita v Liberci.
- [2] STEWART, Robert, Barlee KENNETH a ATKINSON. *Software Defined Radio using MATLAB & Simulink and the RTL-SDR*. Strathclyde Academic Media, 2015. ISBN 0992978718.
- [3] *RTL-SDR V3 Teardown and Analysis* [online]. 2023. [cit. 2023-01-02]. Dostupné z: <https://www.oneelectrontech.com/rtl-sdr-v3-teardown-and-analysis/>.
- [4] FETTE, Bruce A. Chapter 1 - History and Background of Cognitive Radio Technology. In: FETTE, Bruce A. (ed.). *Cognitive Radio Technology (Second Edition)*. Second Edition. Oxford: Academic Press, 2009, s. 1–26. ISBN 978-0-12-374535-4. Dostupné z DOI: <https://doi.org/10.1016/B978-0-12-374535-4.00001-1>.
- [5] *ADALM-PLUTO: Software-Defined Radio Active Learning Module* [online]. Norwood, Massachusetts, U.S.: Analog Devices, 2020 [cit. 2020-05-22]. Dostupné z: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html>.
- [6] *ADALM-PLUTO SDR Active Learning Module* [online]. Norwood, Massachusetts, U.S.: Analog Devices, 2017 [cit. 2020-05-22]. Dostupné z: <https://www.analog.com/media/en/news-marketing-collateral/product-highlight/ADALM-PLUTO-Product-Highlight.pdf>.
- [7] *ADALM-PLUTO Transmit: Transmit Architecture* [online]. Norwood, Massachusetts, U.S.: Analog Devices, 2019 [cit. 2020-05-22]. Dostupné z: <https://wiki.analog.com/university/tools/pluto/users/transmit>.
- [8] *THE BIG LIST OF RTL-SDR SUPPORTED SOFTWARE* [online]. 2022. [cit. 2022-12-18]. Dostupné z: <https://www.rtl-sdr.com/big-list-rtl-sdr-supported-software/>.
- [9] *ADALM-PLUTO Antennas* [online]. Norwood, Massachusetts, U.S.: Analog Devices, 2018 [cit. 2020-05-22]. Dostupné z: <https://wiki.analog.com/university/tools/pluto/users/antennas>.
- [10] *GSM Antenna: JCG401* [online]. Zhenye Road, South Street, Jiaxing, Zhejiang, China [cit. 2023-01-05]. Dostupné z: https://wiki.analog.com/_media/university/tools/pluto/users/jcg401.pdf.

- [11] *Modulation* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2023-01-05]. Dostupné z: https://en.wikipedia.org/wiki/Modulation%5C#/media/File:Modulation_categorization.svg.
- [12] HANUS, Stanislav a Jiří SVAČINA. *Vysokofrekvenční a mikrovlnná technika: přednášky*. 2. vyd. Doplnění. Brno: Vysoké učení technické, 2002. ISBN 80-214-2222-X.
- [13] MACHÁČEK, Zdeněk a Pavel NEVŘIVA. *MODULOVANÉ SIGNÁLY: učební text*. Ostrava: Vysoká škola báňská - Technická univerzita, 2012. ISBN 978-80-248-2600-4.
- [14] VISWANATHAN, Mathuranathan. *Digital Modulations using Python*. 1. vyd. 2019. ISBN 9781712321638.
- [15] SHARMA, Amit Ramkripal. *3pi/8 OFFSET 8PSK MODULATION IN EDGE GSM*. San Diego, 2012. Diplomová práce. San Diego State University.
- [16] ČEJKA, Jiří. *PyModulationPlutoSDR* [online]. [cit. 2023-01-07]. Dostupné z: <https://pypi.org/project/PyModulationPlutoSDR/1.0.3/>.
- [17] ČEJKA, Jiří. *PyModulationPlutoSDR* [online]. [cit. 2023-01-07]. Dostupné z: <https://github.com/jiCejka/PyModulationPlutoSDR>.
- [18] *Cognitive radio technology*. 2nd ed. Amsterdam: Elsevier, 2009. ISBN 978-0-12-374535-4.

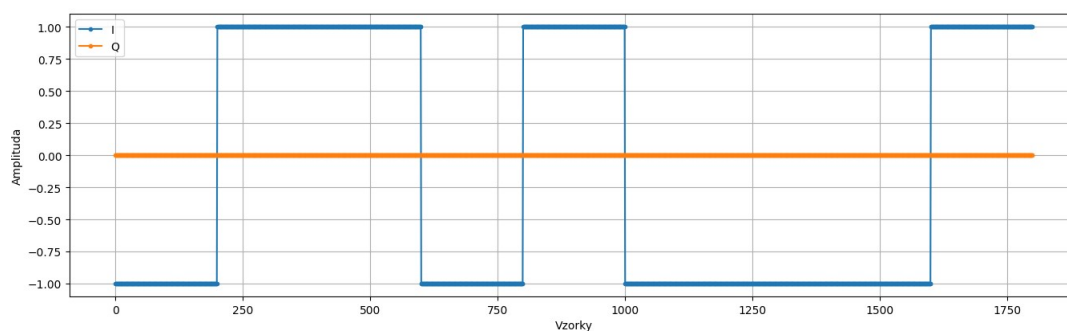
A Seznámení se základními modulacemi

V těchto úlohách se seznámíte se šesti základními modulacemi (AM, FM, ASK, FSK, PSK a QAM). Jedna úloha je více o seznámení s modulacemi a druhá úloha je více praktická. V první úloze máte několik obrázků a máte pomocí PyModulace rozpoznat, o jakou modulaci jde a pokusit se to, co je na obrázku vytvořit v PyModulace. V druhé úloze si vyzkoušíte příjem signálu s neznámou modulací a poté si můžete vyzkoušet i sami něco vysílat.

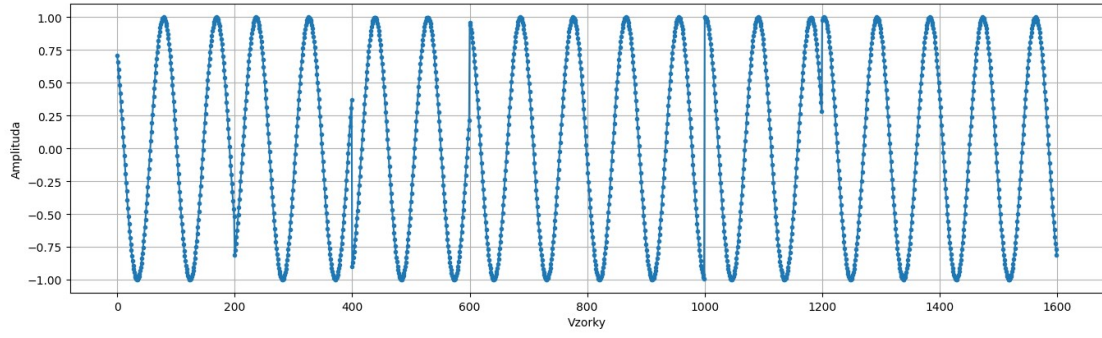
A.1 Úloha 1 - rozpoznání modulace a rekonstrukce modulovaného signálu

Postup je jednoduchý. Otevřete PySDR, zaškrtněte příjem nebo vysílání, a pak klikněte na tlačítko „Popis modulace“. To vám otevře PyModulace, kde se budete snažit identifikovat modulaci a dělat rekonstrukci signálů na obrázcích. Pokud budete mít problémy, že nevíte co jaký prvek dělá, najedte kurzorem na prvek a vyskočí nápověda pro daný prvek.

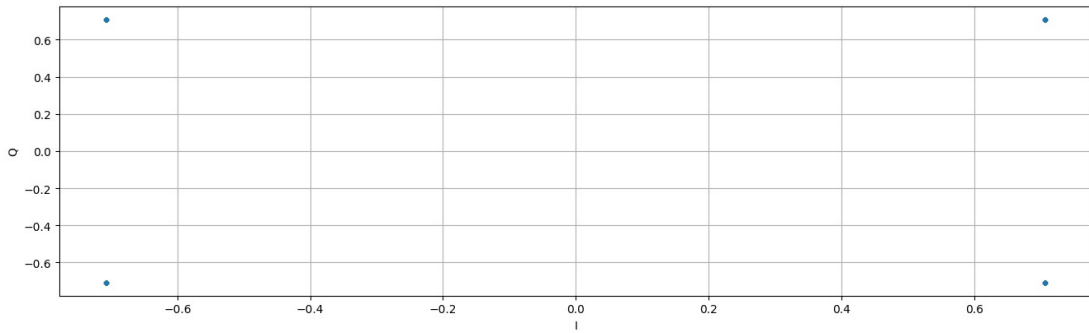
Z následujících obrázků určete, o jaký typ modulace jde a jaké jsou hodnoty parametrů vztahených k dané modulaci (vztahené parametry je možné nastavovat) a jsou validní pro daný obrázek. Po jejich identifikaci se pokuste daný obrázek zrekonstruovat. Obrázek (A.2) není třeba replikovat. Není potřeba identifikovat a zrekonstruovat všechny, vyberte si pět a u nich to udělejte.



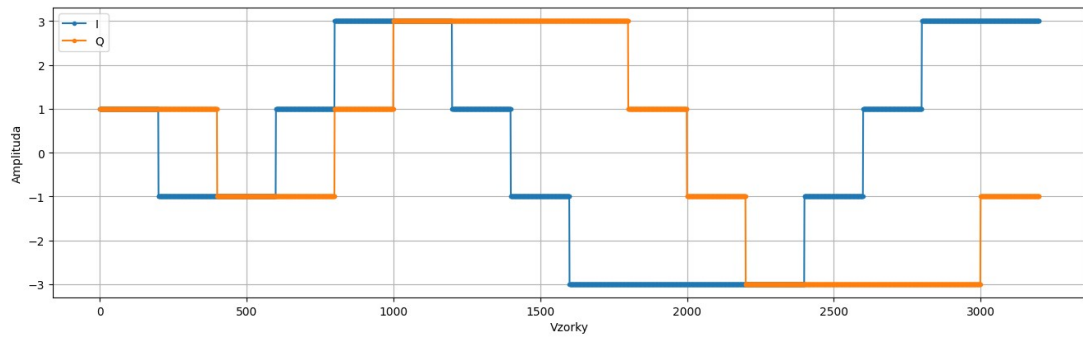
Obrázek A.1: Průběh 1



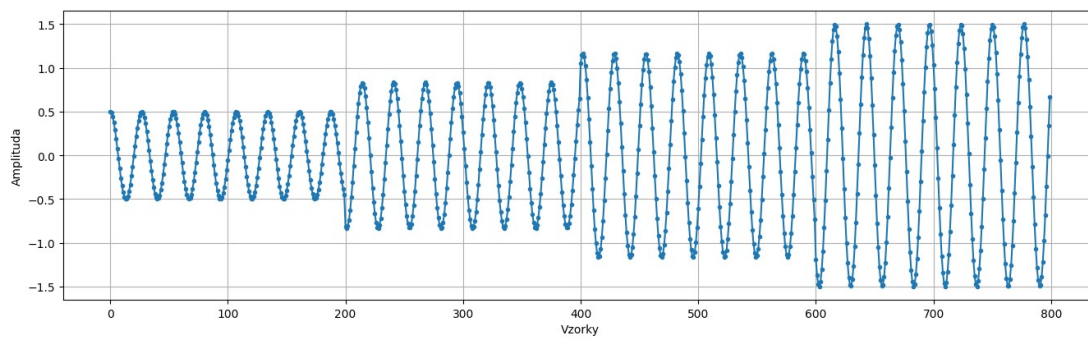
Obrázek A.2: Průběh 2



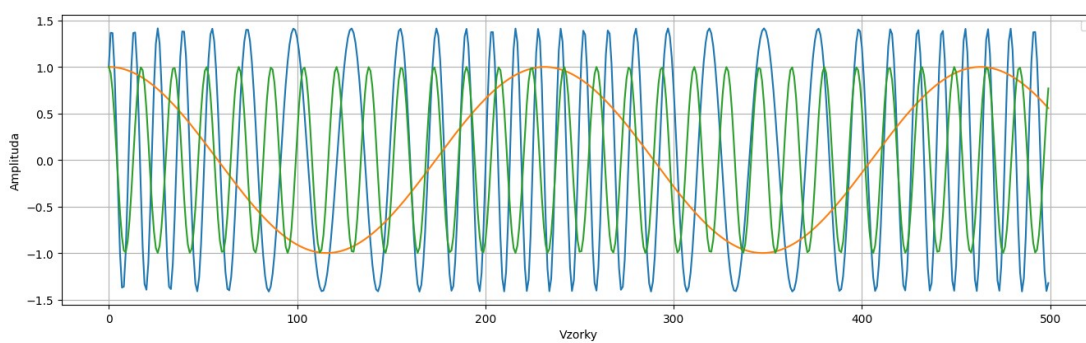
Obrázek A.3: Průběh 3



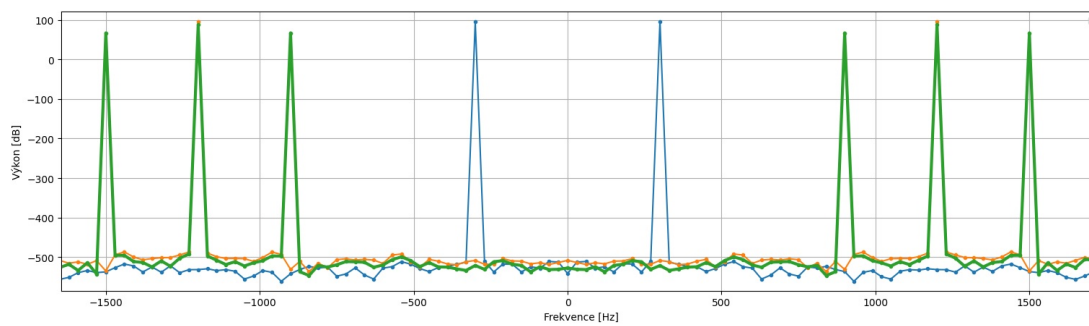
Obrázek A.4: Průběh 4



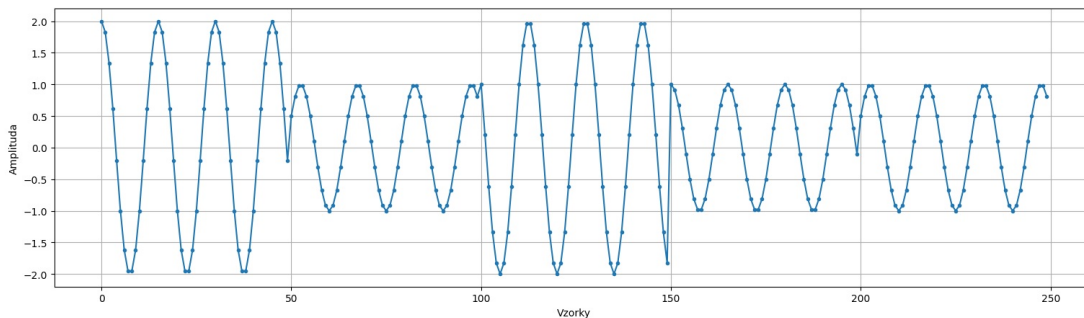
Obrázek A.5: Průběh 5



Obrázek A.6: Průběh 6



Obrázek A.7: Průběh 7



Obrázek A.8: Průběh 8

A.2 Úloha 2 - příjem signálu s neznámou modulací

Vyučující vysílá signál s neznámou modulací a vaším úkolem je přijít na to, jakou modulací vysílá a co vysílá. Vysílá někde mezi 300 a 600 MHz. Pro nalezení na jaké frekvenci vysílá použijte spektrální analyzátor a pro zjištění toho, co vysílá použij PySDR.

Poté co jste přišli na to, co posílá vyučující, také můžete vysílat a přijímat mezi sebou. Vysílejte na frekvenci v rozmezí 430 - 440 MHz nebo 442 - 450 MHz.

A.3 Protokol

Do protokolu napiš, jaké modulační metody jsou na obrázku a přiložte vaši rekonstrukci. Jakou modulaci používal vyučující (napište i kolika úrovně byla), co vysílal (obrázek nebo text) a na jaké frekvenci (dobrá bude fotka spektrálního analyzátoru). Taky do protokolu dejte, co jste vysílali. Můžete dát obrázek toho, co jste vy vysílali, a co zachytili jiní studenti.

Zodpovězte několik otázek:

1. Co jste věděli o modulacích předtím?
2. Co jste se dozvěděl nového?
3. Proč vysílaný signál vyučujícím je silnější než ostatní?
4. Jaké jsou rozdíly mezi 2-ASK a 2-PSK?
5. Jaké jsou rozdíly mezi 4-PSK a 4-QAM?
6. Jaké jsou rozdíly mezi AM a ASK?
7. Jaké jsou rozdíly mezi FM a FSK?
8. Proč si myslíte, že se zřídka používají vysokoúrovňové modulační metody ASK a PSK (16 a více)?