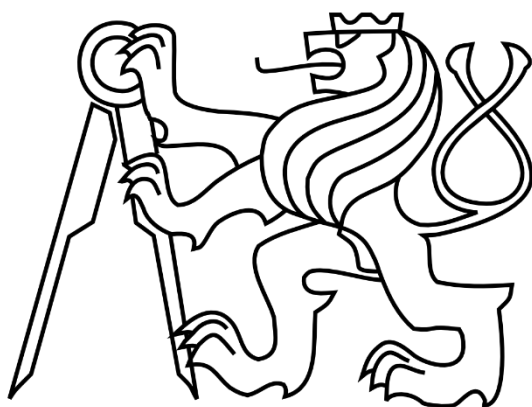


České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra teorie obvodů



SDR radiopřijímač v Matlabu

Diplomová práce

Autor: Bc. David Tomášek, DiS.
Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Komunikační systémy
Vedoucí práce: prof. Ing. Pavel Zahradník CSc.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tomášek** Jméno: **David** Osobní číslo: **457345**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra telekomunikační techniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Komunikační systémy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

SDR radiopřijímač v Matlabu

Název diplomové práce anglicky:

SDR Radio using Matlab

Pokyny pro vypracování:

K PC připojte modul SDR přijímače přes rozhraní USB. Ověřte činnost této konfigurace SDR přijímače. Modifikujte programové vybavení firmy Mathworks pro připojení SDR přijímače přes vstupy zvukové karty. Uvažujte generaci taktovacího signálu pro nastavení přijímaného kmitočtu řízenou z prostředí Matlab.

Seznam doporučené literatury:

- [1] Dokumentace dostupná na <https://www.mathworks.com/help/supportpkg/rtlsdradio/index.html> [on-line]
- [2] Dokumentace dostupná na <https://www.mathworks.com/help/audio/guide/real-time-audio-in-matlab-1.html> [on-line]
- [3] Dokumentace dostupná na http://www.ok1cjb.cz/index.php?option=com_content&view=article&id=543:3-312&catid=33:sdr-radio&Itemid=45 [on-line]

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Pavel Zahradník CSc., katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **15.02.2017**

Termín odevzdání diplomové práce: **26.05.2017**

Platnost zadání diplomové práce: **30.09.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

10.5.2017

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu diplomové práce prof. Ing. Pavlovi Zahradníkovi CSc. za jeho odborné rady, ochotu a trpělivost při zpracování této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Písku, dne 07. 01. 2018

podpis autora
Bc. David Tomášek, DiS.

Abstrakt

Diplomová práce má za cíl popsat využití Softwarově definovaného rádia (dále jen „SDR“) v prostředí Matlab. Práce obsahuje úvod do problematiky SDR a I/Q signálů, s kterými SDR pracuje. Funkce SDR přijímače je vysvětlena na dvou nejvíce používaných variantách: RTL-SDR a SDR DR2B. Dále jsou popsány principy zpracování real-time signálu v prostředí Matlab a základy pro vytváření GUI pomocí Guide. Výsledkem práce je zprovoznění obou variant SDR přijímače v prostředí Matlab, včetně generátoru kmitočtu přes rozhraní USB pro potřeby ladění SDR DR2B. Tento krok je v závěru práce prezentován na vytvořené aplikaci v Matlabu, která využívá v práci popsané postupy.

Klíčová slova

Softwarově definované rádio, Matlab, real-time zpracování signálu, grafické uživatelské rozhraní.

Abstract

This thesis aims to describe the use of Software defined radio (“SDR”) receiver in the Matlab environment. Thesis contains introduction to SDR and I/Q signals, which are used by SDR. The function of the SDR receiver is explained on two of the most used variants: RTL-SDR and SDR DR2B. Further, principles of real-time signal processing in the Matlab environment and the basics for creating a GUI with Guide are described. Result of the thesis is both SDR receiver variants working in the Matlab environment, including the frequency generator via USB for SDR DR2B tuning. This step is presented at the end of the thesis by created application in Matlab, which uses procedures described in thesis.

Key words

Software defined radio, Matlab, real-time signal processing, graphical user interface.

Obsah

1	SOFTWAREVĚ DEFINOVANÉ RÁDIO	4
1.1	POJEM „SOFTWAREVĚ DEFINOVANÉ“	4
1.2	SDR (SOFTWAREVĚ DEFINOVANÉ RÁDIO)	5
1.3	I/Q SIGNÁLY.....	6
2	DRUHY SDR	8
2.1	RTL-SDR (SDR Z DVB-T TUNERU)	9
2.1.1	<i>Zadig (změna ovladačů, DVB-T > RTL-SDR)</i>	11
2.2	SDR DR2B.....	13
2.2.1	<i>YU1LM/QRP</i>	13
2.2.2	<i>DR2B</i>	14
2.2.3	<i>Schéma a popis funkčnosti</i>	16
3	OSCILÁTOR ŘÍZENÝ MATLABEM	20
3.1	AD9850.....	20
3.1.1	<i>Specifikace AD9850</i>	20
3.1.2	<i>Výhody a nevýhody AD9850</i>	20
3.1.3	<i>Poznámka</i>	21
3.1.4	<i>Popis funkčnosti AD9850</i>	22
3.1.5	<i>Programování AD9850</i>	23
3.2	FTDI UM245R	26
3.2.1	<i>Ovládání UM245R z Matlabu.</i>	27
3.3	FREKVENČNÍ GENERÁTOR PŘES USB (UM245R + AD9850)	28
3.3.1	<i>Průběhy z frekvenčního generátoru</i>	30
3.4	SDR DR2B S OSCILÁTOREM ŘÍZENÝM PŘES USB.....	32
4	SDR V MATLABU	33
4.1	TOOLBOXY	33
4.2	PODPŮRNÉ BALÍČKY (SUPPORT PACKAGES).....	33
5	REAL-TIME ZPRACOVÁNÍ SIGNÁLU V MATLABU	34
5.1	VYTVORENÍ SYSTÉMOVÝCH OBJEKTŮ.....	35
5.1.1	<i>Vstupní zařízení</i>	35
5.1.2	<i>Výstupní zařízení</i>	36
5.1.3	<i>Filtry, demodulátory</i>	36
5.2	SMYČKA	36

5.3	UVOLNĚNÍ SYSTÉMOVÝCH OBJEKTŮ	36
6	GUI V MATLABU	37
6.1	METODY VYTVÁŘENÍ GUI	37
6.1.1	<i>App Designer</i>	37
6.1.2	<i>GUIDE</i>	38
6.1.3	<i>Programové GUI</i>	38
6.1.4	<i>Výběr nejvhodnějšího GUI</i>	38
6.2	GUI POMOCÍ GUIDE.....	39
6.2.1	<i>Funkce Callback</i>	39
6.2.2	<i>Struktura</i>	40
7	APLIKACE V MATLABU	41
8	PŘÍSLUŠENSTVÍ	43
8.1	ANTÉNA MINIWHIP	43
9	ZÁVĚR.....	44
10	ZDROJE	45
11	TERMINOLOGICKÝ SLOVNÍK.....	47
12	PŘÍLOHY NA CD/DVD	48

Obrázky

U každého obrázku je v závorce uveden původ. Číslo odkazuje na referenci.

Obr. 1 – Blokové schéma RTL-SDR [1].....	5
Obr. 2 – Ukázka funkce cosinus [3].....	6
Obr. 3 – Bod $0,69 + 0,40i$ v komplexní rovině [3].....	7
Obr. 4 – RTL SDR [vlastní].....	10
Obr. 5 – RTL SDR, pohled na PCB [vlastní].....	10
Obr. 6 – Program Zadig [vlastní].....	11
Obr. 7 – Program SDR# [vlastní].....	12
Obr. 8 – SDR DR2B v1 [vlastní].....	14
Obr. 9 – SDR DR2B v2 [vlastní].....	15
Obr. 10 – Schéma SDR DR2B [10].....	16
Obr. 11 – 10 MHz signál na vstupu 74AC74 [vlastní].....	17
Obr. 12 – 2,5 MHz výstupní signály z 74AC74 [vlastní].....	17
Obr. 13 – Schéma detektoru s 74HC4053 [11].....	18
Obr. 14 – Nabíjení kondenzátorů v polárním zobrazení [11].....	19
Obr. 15 – Pohled na AD9850 [vlastní].....	20
Obr. 16 – Blokové schéma AD9850 [14].....	22
Obr. 17 – Časování zápisu do registru [14].....	23
Obr. 18 – Paralelní zápis frekvence do AD9850 [14].....	24
Obr. 19 – Sériový zápis frekvence do AD9850 [14].....	25
Obr. 20 – Pohled na FTDI UM245R [vlastní].....	26
Obr. 21 – Schéma frekvenčního generátoru s UM245 a AD9850 [vlastní].....	28
Obr. 22 – Frekvenční generátor 500 KHz [vlastní].....	30
Obr. 23 – Frekvenční generátor 1 MHz [vlastní].....	30
Obr. 24 – Frekvenční generátor 2 MHz [vlastní].....	31
Obr. 25 – Frekvenční generátor 5 MHz [vlastní].....	31
Obr. 26 – Frekvenční generátor 10 MHz [vlastní].....	31
Obr. 27 – SDR DR2B s oscilátorem řízeným přes USB 2 [vlastní].....	32
Obr. 28 – SDR DR2B s oscilátorem řízeným přes USB 2 [vlastní].....	32
Obr. 29 – Matlab logo [28].....	33
Obr. 30 – Aplikace v Matlabu – RTL-SDR, 89 MHz [vlastní].....	41
Obr. 31 – Aplikace v Matlabu – SDR DR2B, 6 MHz [vlastní].....	42

Tabulky

Tabulka 1 – Použité tunery v RTL-SDR [6].....	9
Tabulka 2 – Časování zápisu do registru [14].....	23

1 Softwarově definované rádio

Tato kapitola seznamuje čtenáře s pojmem softwarově definované rádio. Nejdříve je vysvětleno, co znamená obecný výraz „Softwarově definované“. Následuje vysvětlení pojmu „Softwarově Definované Rádio“. Kapitola končí popisem I/Q signálů, které slouží jako přenašeč informace v oblasti SDR.

1.1 Pojem „Softwarově definované“

Pojem Softwarově definované(-ý) (Software defined) je moderní trend, kterým se ubírá mnoho oblastí okolo nás. Tento pojem se začal používat na konci 90. let minulého století, avšak jeho rozmach se muselo počkat až na začátek nového tisíciletí, kdy jeho používání exponenciálně roste. Jak už název napovídá, můžeme se s tímto pojmem setkat v oblasti IT.

Co si ale můžeme představit pod pojmem Softwarově definovaný? Svými slovy bych vysvětlil, že se jedná o snahu výrobců, či amatérů oprostít se od zařízení, která jsou již z hlediska hardwarového designu vyrobena pouze za jedním určitým účelem, případně nabízející pouze omezenou funkcionalitu v oblasti, ve které pracují. Hlavní myšlenkou softwarově definovaného zařízení je přesun hardwarových funkcí na softwarovou vrstvu.

Z toho samozřejmě plynou výhody i nevýhody

Klady [+]:

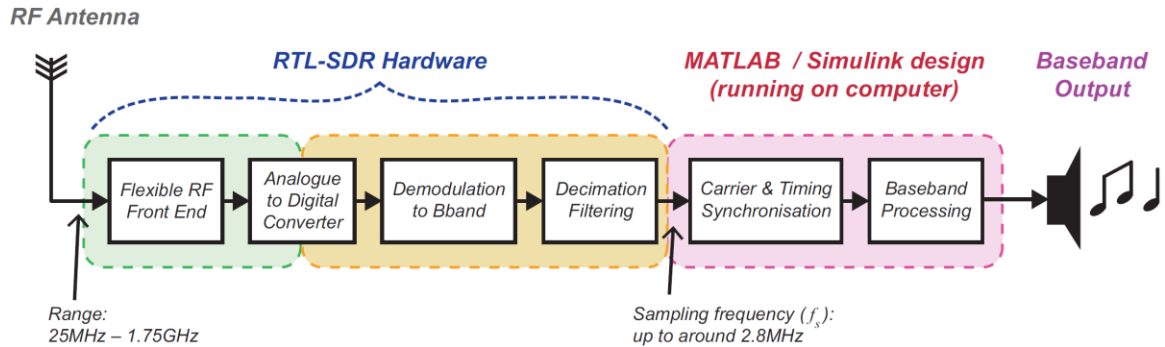
- Jednoduchý upgrade – hardware se při požadavku na spuštění nové služby může stát zastaralým a je potřeba ho vyměnit. Softwarovou změnu lze provést i na starším zařízení.
- Rychlost změny – jelikož se změny provádí v zásadě softwarově je možná rychlá změna parametrů zařízení. Při velkém počtu zařízení se čas změny dramaticky snižuje.
- Nižší náklady – jak na údržbu, tak na nákup nového zařízení. Vyplývá z předchozích bodů.

Zápory [-]:

- Vyšší náročnost na hardware – hardware zde není specializovaný pro ten daný účel. Zpracování se „virtualizuje“, což vede k vyšším požadavkům na výkon hardware.

1.2 SDR (Softwarově definované rádio)

Softwarově definované rádio (SDR) je rádio ve kterém jsou procesy jako zpracování, filtrování, demodulace signálu a další neprováděny specializovaným hardwarem, ale softwarem, běžícím na klasickém počítači či jiném systému.



Obr. 1 – Blokové schéma RTL-SDR [1]

Definice dle ITU-R SM.2152 (09/2009):

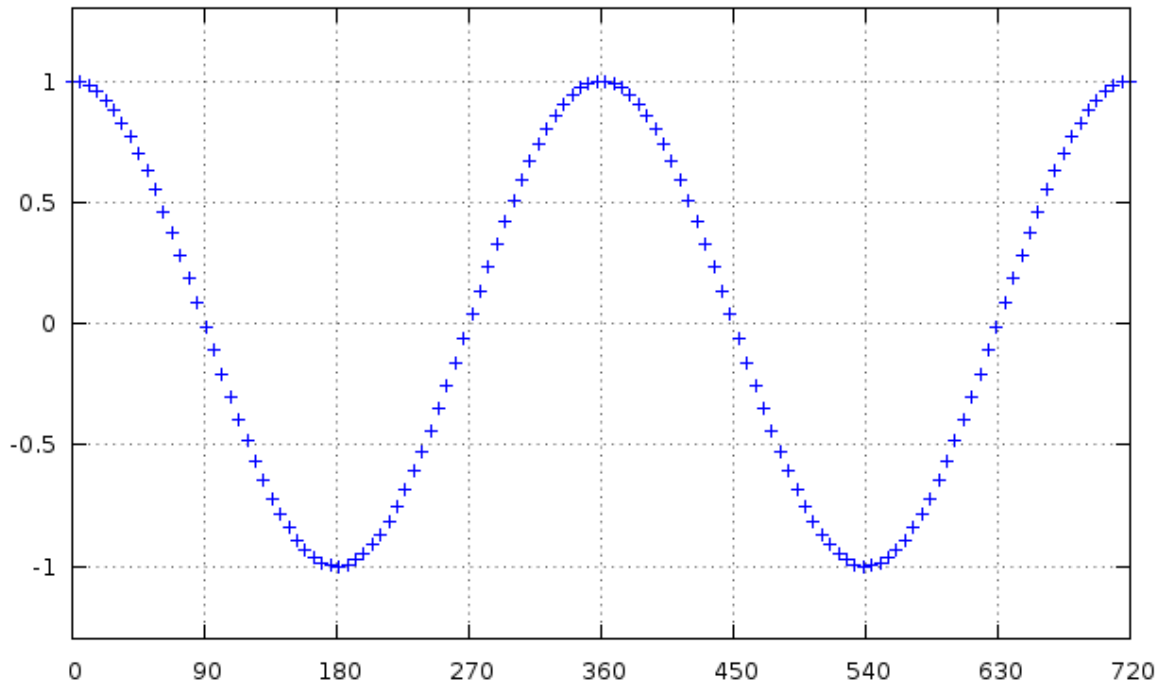
“Software-defined radio (SDR): A radio transmitter and/or receiver employing a technology that allows the RF operating parameters including, but not limited to, frequency range, modulation type, or output power to be set or altered by software, excluding changes to operating parameters which occur during the normal pre-installed and predetermined operation of a radio according to a system specification or standard.” [2]

Mezi výhody softwarově definovaného rádia oproti klasickým rádiům patří:

- Větší frekvenční rozsah – stovky MHz, až jednotky GHz.
- Šířka pásma přijímaného signálu (stovky KHz až desítky MHz) umožňuje příjem několika stanic/signálů paralelně na jednom zařízení.
- Takřka neomezené možnosti, jak zpracovat přijímaný signál.

1.3 I/Q signály

Výstup z SDR (Softwarově Definované Rádio) je reprezentován pomocí I/Q signálu. Abychom mohli začít s řešením jakýchkoliv problémů z oblasti SDR je nutné porozumět základním charakteristikám I/Q signálu a jak se s těmito signály dále pracuje.



Obr. 2 – Ukázka funkce cosinus [3]

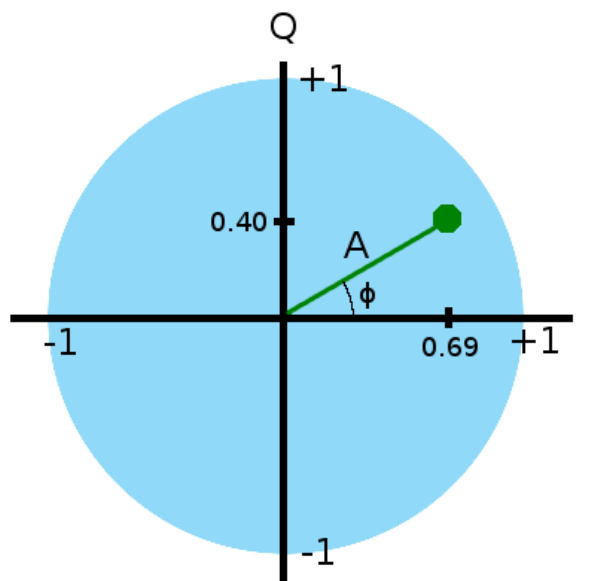
Proč nejsou data reprezentována pouze jednou množinou dat, ale dvěma množinami v podobě I/Q signálu má 2 hlavní důvody [3]:

- První důvod je, že při digitalizaci signálu z SDR, vzorkujeme reálně signál v rozsahu $-fs/2$ až $+fs/2$ (fs = vzorkovací kmitočet) okolo střední frekvence. Pokud bychom vzorkovali pouze jednou množinou dat, nebyli bychom schopni od sebe rozlišit složky záporné a kladné frekvence, jelikož $\cos(x) = \cos(-x)$, viz obr. 2.
- Druhý důvod je, že při použití jednoho signálu můžeme sílu/amplitudu signálu určit pouze v 0° , 180° atd. Vzorek signálu nemusí být přesně v maximu, a tudíž přicházíme o přesnou informaci.

I/Q signály tyto nedostatky eliminují. Jak už může být z názvu patrné, I/Q signál má dvě složky, které dohromady tvoří komplexní signál:

- I = In-Phase, neboli reálnou složku.
- Q = Quadrature, neboli imaginární složku.

Tyto signály jsou navzájem posunuté o 90° ($1/4$ periody), čímž je lze dle definice označit společně jako kvadraturní signály. Na první pohled někoho může zarazit imaginární složka signálu. I když jsou tyto signály svojí povahou reálné, je vhodné, je při jejich zpracování zapsat ve tvaru komplexního čísla.



Obr. 3 – Bod $0,69 + 0,40i$ v komplexní rovině [3]

I/Q signály mohou být znázorněny ve 3 doménách [3; 4]:

- 1) Klasický zápis v komplexní rovině, pomocí reálné a imaginární složky.

$$I/Q = I + Qi$$

- 2) Zápis pomocí polárních souřadnic = absolutní hodnoty vektoru a úhlu, který svírá s osou X.

$$I = A * \cos(\varphi)$$

$$Q = A * \sin(\varphi)$$

Kde A je amplituda a φ je úhel.

- 3) Zápis čísla jako komplexní exponent.

$$I/Q = A * e^{i*\varphi}$$

Kde A je amplituda a φ je úhel.

Nejčastěji se setkáme s 1. zápisem, jelikož se jedná o nejjednodušší zápis I/Q dat a je podporován v Matlabu.

2 Druhy SDR

V následujícím textu jsou popsány dvě z nejrozšířenějších řešení Softwarově Definovaného Rádía:

- Prvním řešením je RTL-SDR, což je SDR rádio vytvořené pomocí softwarové modifikace levného DVB-T donglu do USB.
- Druhé řešení má podobu DRxxx rádia, které lze postavit s nízkými náklady v domácích podmínkách.

Oba tyto systémy jsou velmi oblíbené mezi radioamatéry, hlavně pro jejich dostupnost, nízkou cenu, a že společně dokážou přijímat signál od frekvence 30 KHz až do 1,75 GHz (hodnoty se mohou lišit dle konfigurace).

Nejzajímavější frekvenční pásma z hlediska SDR příjmu jsou [1]:

Amatérské pásmo 160m	1,81 – 2,00 MHz
Amatérské pásmo 80m	3,50 – 3,80 MHz
Amatérské pásmo 40m	7,00 – 7,20 MHz
FM rádia	87,5 – 108 MHz
Letecké pásmo	108 – 137 MHz
RF klíče	433 MHz
LTE	800 MHz
IoT	863 – 870 MHz
ADS-B signály	1090 MHz
GPS	1227 / 1575 MHz

2.1 RTL-SDR (SDR z DVB-T tuneru)

První RTL-SDR vzniklo tak, že Antti Palosaari a Eric Fry společně se členy projektu Osmocom přišli na to, že je možné modifikovat DVB-T přijímač (Digital Video Broadcasting – Terrestrial, neboli pozemní digitální televizní vysílání) tak, že úpravou ovladačů jej přepnuli z režimu, ve kterém normálně pracuje, tedy přijímá frekvence DVB-T vysílání (174 - 862 MHz), do režimu, kde přijímač posílá na výstup přímo data v podobě I/Q signálu a je možné jej naladit na větší rozsah než ve kterém je přijímáno DVB-T vysílání. V tomto případě byl použit DVB-T přijímač v podobě donglu do USB, který obsahuje populární 8 bitový digitální demodulátor Realtek RTL2832U (z toho také plyne název "RTL-SDR"), který je schopen vzorkovací frekvence až do 3,2 MSPS (Milion Samples Per Second – milionů vzorků za sekundu) [5].

Kromě demodulátoru Realtek, který je v DVB-T přijímači klíčovým prvkem je důležité, avšak sekundární, také hledět na použitý tuner, který přímo ovlivňuje rozsah frekvence, kterou na přijímači naladíme, na seznam použitých tunerů se můžeme podívat níže.

Obrovská výhoda těchto přijímačů je jejich dostupnost a nízká cena, dongle do USB s použitým demodulátorem Realtek RTL2832U a tunerem Rafael Micro R820T2 lze koupit už kolem 200Kč.

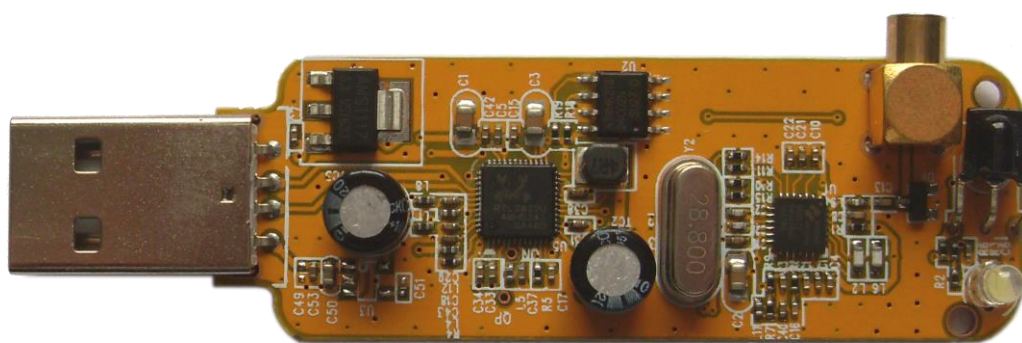
Tuner	Frekvenční rozsah
Elonics E4000	52 – 2200 MHz s mezerou od 1100 MHz do 1250 MHz (může se lišit)
Rafael Micro R820T(2)	24 – 1766 MHz
Rafael Micro R828D	24 – 1766 MHz
Fitipower FC0013	22 – 1100 MHz (FC0013B/C, FC0013G má oddělený vstup L-pásmu, který není na mnoho donglech připojen)
Fitipower FC0012	22 – 948,6 MHz
FCI FC2580	146 – 308 MHz a 438 - 924 MHz (mezera mezi frekvencemi)

Tabulka 1 – Použité tunery v RTL-SDR [6]

Níže můžeme vidět příklad DVB-T donglu s tunerem Rafael Micro R820T2 a MCX konektorem.



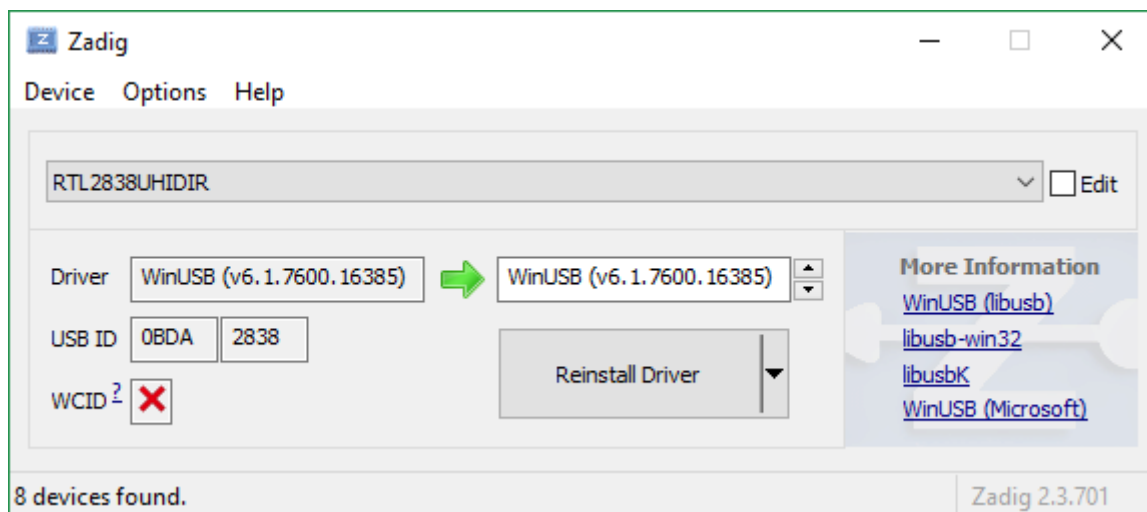
Obr. 4 – RTL SDR [vlastní]



Obr. 5 – RTL SDR, pohled na PCB [vlastní]

2.1.1 Zadig (změna ovladačů, DVB-T > RTL-SDR)

Program Zadig slouží k jednoduché přeinstalaci ovladače zařízení na generický ovladač USB (WinUSB). To se dělá z důvodu, abychom mohli DVB-T dongle používat jako RTL-SDR [7].



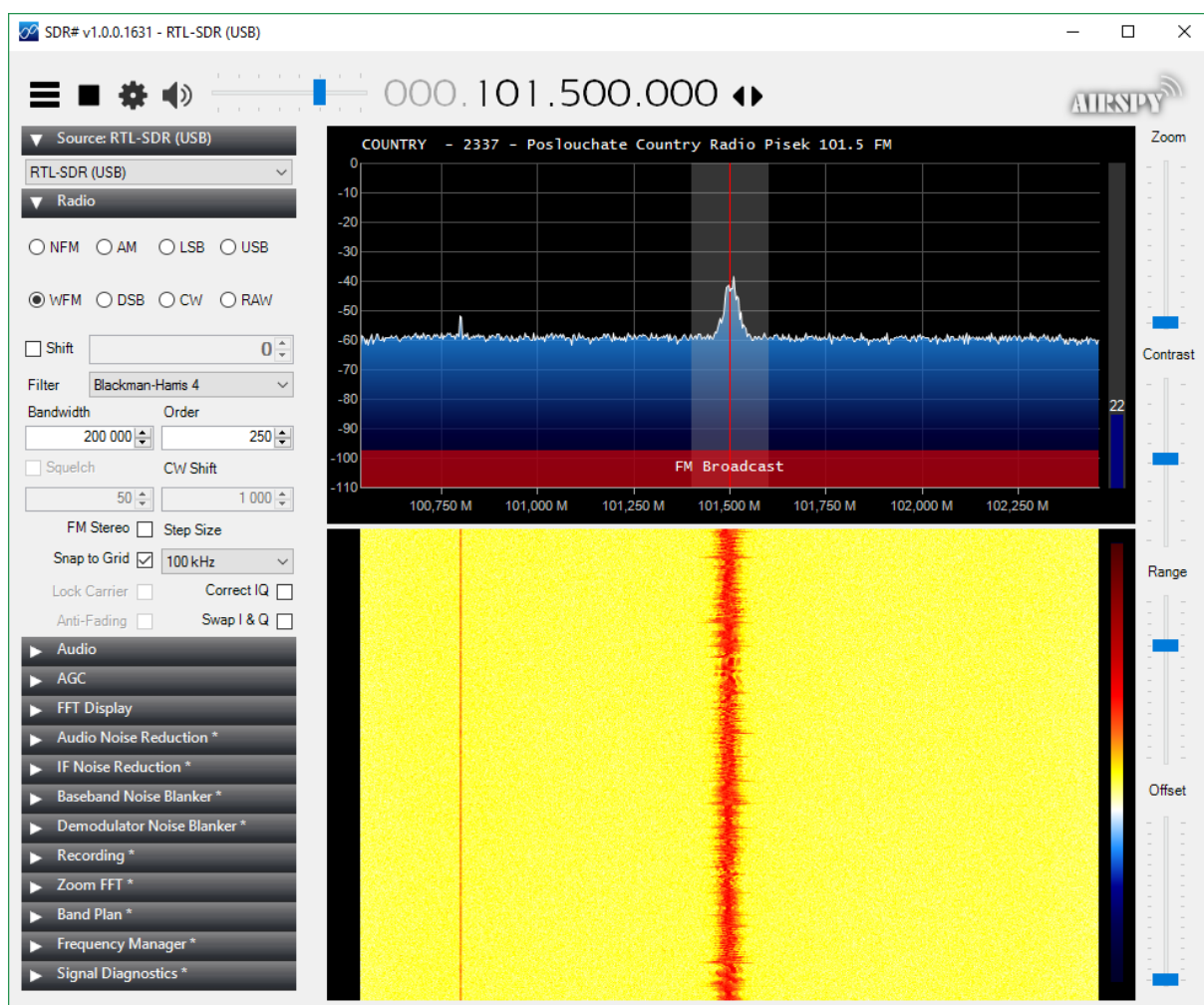
Obr. 6 – Program Zadig [vlastní]

Změna ovladačů se provede pomocí následujících kroků (návod je psaný pro Windows 10):

- 1) Program Zadig stáhneme na stránkách výrobce (<http://zadig.akeo.ie>).
- 2) DVB-T dongle se zapojí do USB portu.
- 3) Přes správce se spustí program Zadig (neinstaluje se).
- 4) V záložce [Options] se zaškrtně položka [List All Devices].
- 5) Z menu vybereme DVB-T dongle – většinou s RTL2832U v názvu. Popř. odpojíme nepotřebná zařízení a připojíme pouze DVB-T dongle.
- 6) Jako cílový driver zvolíme [WinUSB].
- 7) Klikneme na tlačítko [Replace driver].

Funkčnost ovladačů si můžeme rovnou vyzkoušet v nějakém SDR programu, osobně doporučuji SDR# od společnosti Airspy:

- 1) Stáhněte program SDR# ze stránek výrobce (<https://airspy.com>).
- 2) Po rozbalení spusťte [install-rtlsdr.bat] a nainstalujte ovladače pomocí Zadig, pokud jste ještě tak neučinili.
- 3) Spusťte program pomocí [SDRSharp.exe].
- 4) Pod záložkou [Source] zvolte [RTL-SDR (USB)].
- 5) V nastavení (ozubené kolečko) zvolte vzorkovací kmitočet 2.4 MSPS (u vyšší frekvence může docházet ke ztrátě vzorků).
- 6) Spusťte příjem tlačítkem [Start] a naladěte například FM stanici z rozsahu 87,5 – 108 MHz.



Obr. 7 – Program SDR# [vlastní]

2.2 SDR DR2B

2.2.1 YU1LM/QRP

Než se pustíme do popisu SDR DRxxx, přesněji DR2B řekneme si něco málo o jeho autorovi.

Tasić Siniša – Tasa alias YU1LM/QRP je tvůrce několika SDR návrhů, které jsou volně dostupné pro nekomerční použití. Tasa už se už na základní škole projevil zájem o radioamatérství a ve 13 letech postavil svoje první HAM rádio. To ho tak nadchlo, že ve svém koníčku pokračoval v následujících letech a později se z hobby stalo i povolání. S vývojem nových technologií se Tasa v roce 1998 začal zabývat vývojem SDR. Dle něj je SDR budoucností rádií, poskytuje totiž maximální flexibilitu a možnosti pouhou SW úpravou, což by jinak bylo těžce docílitelné úpravou HW. Od roku 2005 publikoval na internet svoje návrhy SDR rádií. Až do dnešní doby mají tyto SDR velkou popularitu mezi radioamatéry a Tasa se díky nim stál také známý a uznávaný v komunitě radioamatérů [8].

Jeho SDR návrhy mají společných několik klíčových věcí:

- Jednoduchost – každé SDR má kolem 50-70 obvodových součástek a složitost obvodu není taková, aby jej bylo obtížné sestavit v domácích podmínkách.
- Dostupnost – většina součástek je až na výjimky (klopné obvody) běžně dostupná u větších prodejců elektronických součástek.
- Kvalita – příjem je kvalitou srovnatelný s komerčními SDR.
- Výstup – výstup je řešen přímým výstupem IQ signálu přes klasický audio jack (k převedení dat do digitální podoby je potřeba A/D převodník, k čemuž poslouží zvuková karta s mikrofonním či line-in vstupem).
- Nízká cena – v závislosti na návrhu a použitých součástkách začíná cena od cca 200 Kč.

2.2.2 DR2B

Jedním z nejoblíbenějších Tasovo návrhů SDR je DR2B. DR2B vyniká tím, že je jednoduché na sestavení, a tudíž se hodí pro SDR začátečníky. Jednoduchost ale neznamená, že by toto SDR zaostávalo za jinými návrhy [9].

Mezi hlavní parametry DR2B patří:

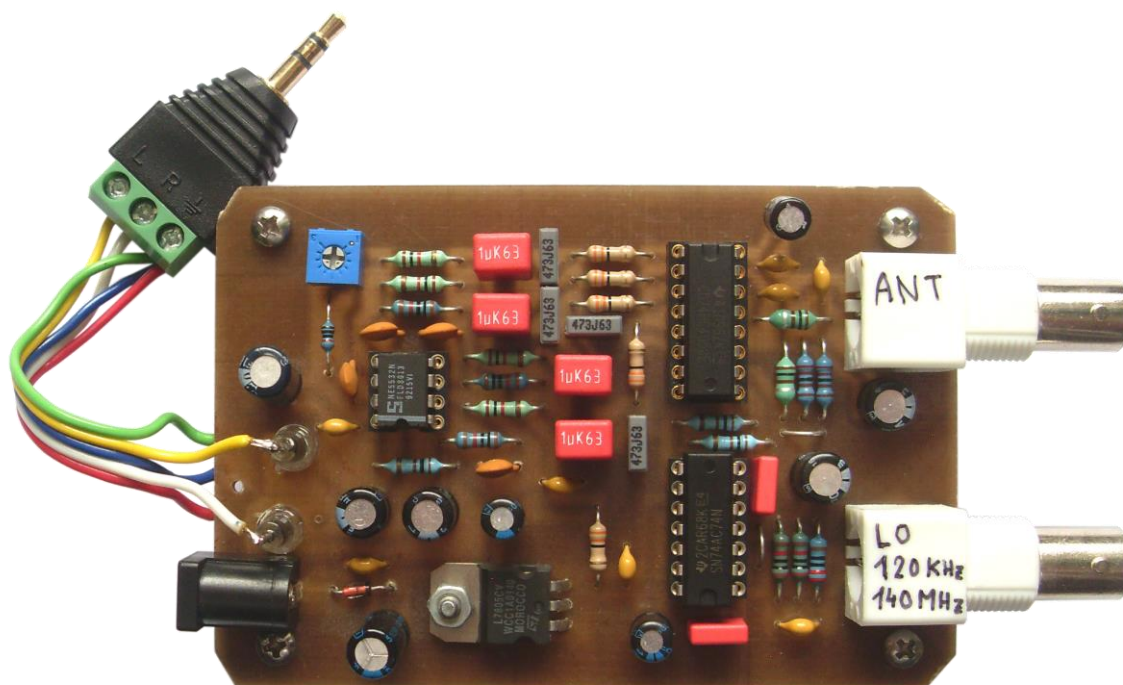
- Přijímací rozsah 30 KHz – 35 MHz
- Ladění pomocí externího LO (lokální oscilátor)
- Nízký příkon <100 mA při 12 V

Nevýhody DR2B:

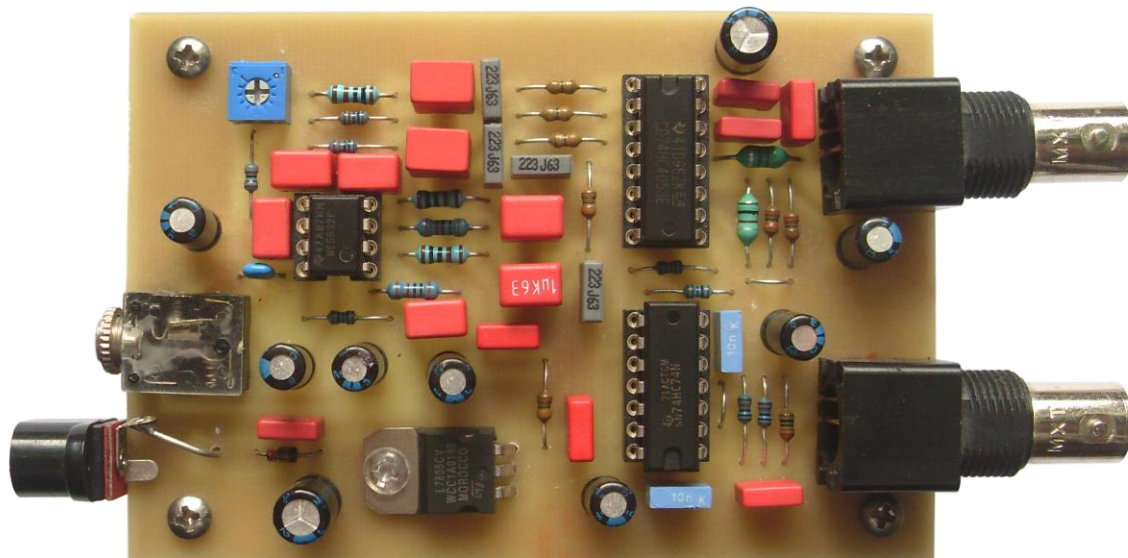
- Kvůli děličce kmitočtu na vstupu je nutná 4x vyšší frekvence LO. Vzhledem k přijímacímu rozsahu to znamená frekvenci 120 KHz – 140 MHz.
- Potlačení zrcadlového signálu není konstantní, ale je proměnné s frekvencí.

Před připojením a spuštěním je DR2B nutno nastavit, aby potlačení zrcadlového obrazu bylo co nejlepší:

- 1) Vstupní oscilátor musí do DR2B dodávat signál
- 2) Pomocí multimetru se na rezistoru R30 = 8,2 K Ω a trimeru R31 = 5 K Ω nastaví v sérii R = 10 K Ω .
- 3) Naladte signál ve vzdálenosti 12 KHz od nuly, nebo na vstup připojte generátor signálu.
- 4) Upravte 5 K Ω trimer R31 tak, aby zrcadlový signál byl co nejmenší.



Obr. 8 – SDR DR2B v1 [vlastní]



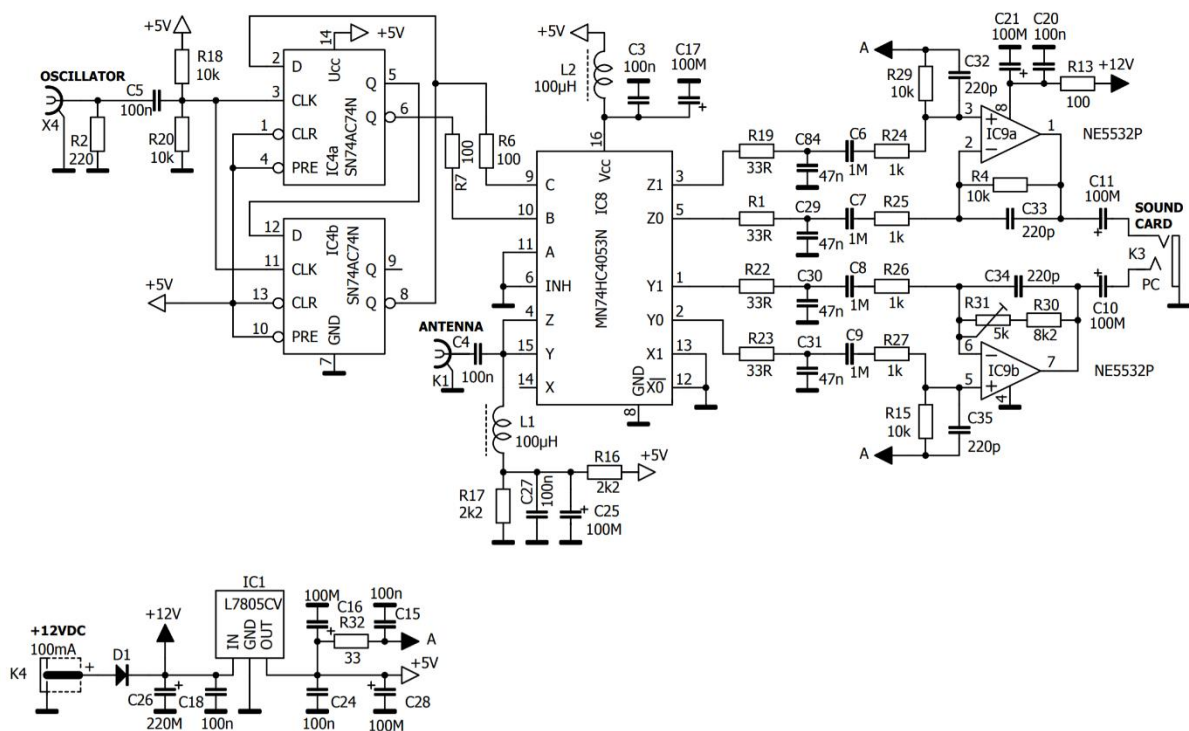
Obr. 9 – SDR DR2B v2 [vlastní]

2.2.3 Schéma a popis funkčnosti

Princip fungování DR2B není tak těžký, jak by se mohlo na první pohled zdát. Na obrázku je uvedeno schéma DR2B, na kterém si popíšeme jeho nejdůležitější části a jak fungují.

Pro správnou funkci SDR DR2B je nutné mít na vstup přivedeny 2 typy signálů:

- První typ signálu je vysokofrekvenční signál z antény.
- Druhý typ je obdélníkový signál z generátoru kmitočtu, který má 4x větší hodnotu než frekvenci, kterou chceme přijímat s SDR (v dalších krocích si vysvětlíme proč tomu tak je).

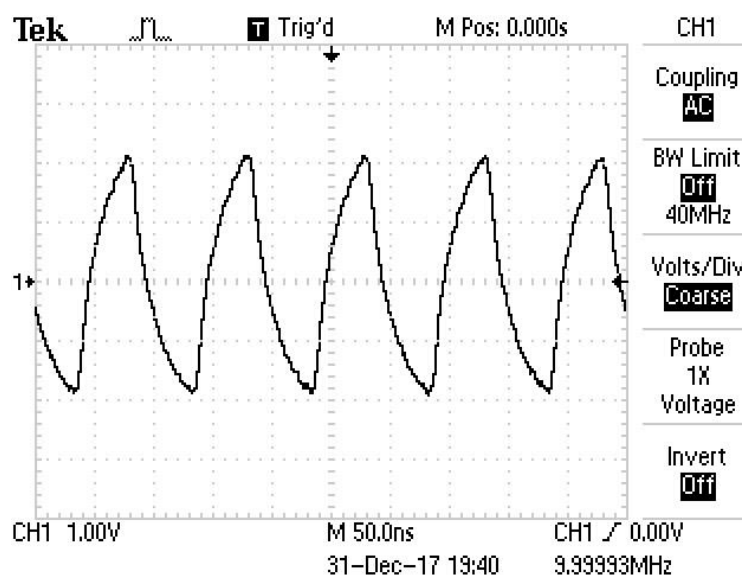


Obr. 10 – Schéma SDR DR2B [10]

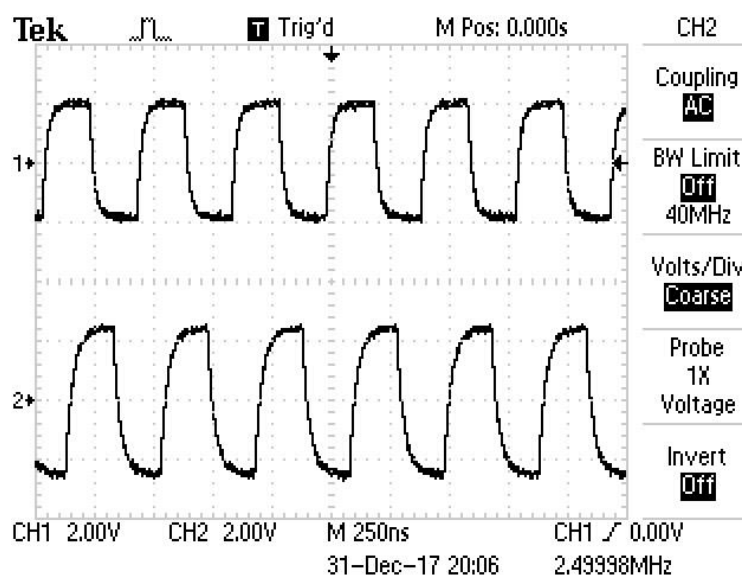
Popis funkčnosti SDR DR2B:

74AC74 – zpracování signálu začíná integrovaným obvodem, který je tvořen dvěma klopnými obvody typu D. Tyto klopné obvody jsou v sériovém zapojení. Po přivedení hodinového signálu z frekvenčního generátoru na vstupy klopných obvodů dojde ke generování obdélníkového signálu na jejich Q/Q' výstupech. Výstupní signály jsou celkově 4 a jsou vůči sobě posunuté o 90° (0°, 90°, 180° a 270°). Každý jednotlivý signál má už z principu D klopných obvodů čtvrtinovou frekvenci původního kmitočtu (10 MHz -> 2,5 MHz) [11].

Na druhém obrázku si můžeme všimnout posunu mezi výstupními signály z 74AC74 (Veškeré obrázky z osciloskopu byly vytvořeny pomocí osciloskopu Tektronix TDS 1001B - 40 MHz, 500 MS/s s firmware v22.01.).

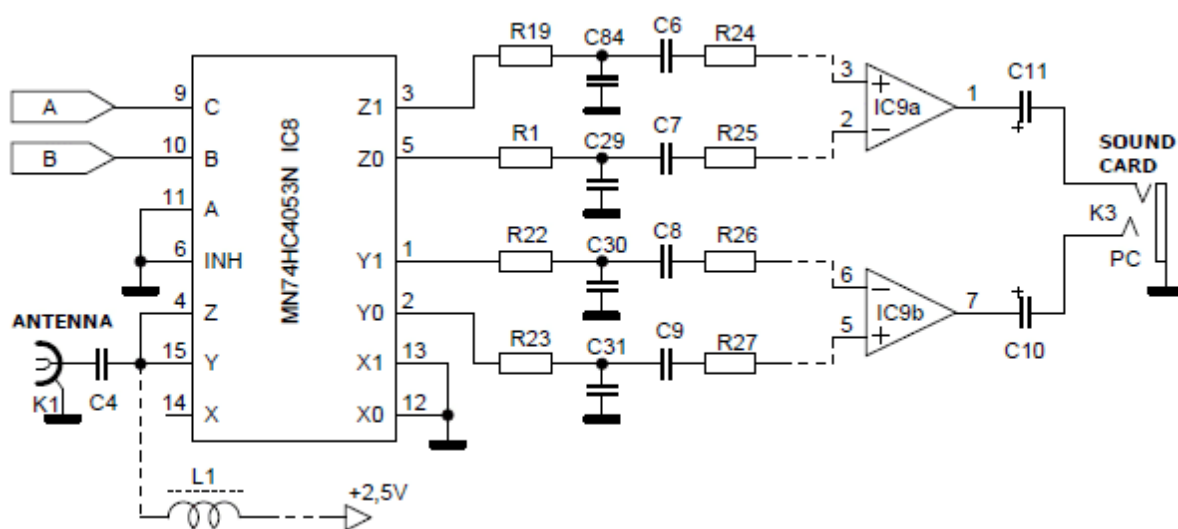


Obr. 11 – 10 MHz signál na vstupu 74AC74 [vlastní]



Obr. 12 – 2,5 MHz výstupní signály z 74AC74 [vlastní]

74HC4053 – integrovaný obvod, který následuje je tvořen trojitým analogovým multiplexorem / demultiplexorem. Tento integrovaný obvod se běžně používá jako směšovač signálu. V DR2B či jiných SDR má však trochu odlišnou funkci a to, že slouží jako přepínač vstupního signálu. Přepínače v tomto zapojení se nazývají kvadratické detektory a slouží pro konverzi vysokofrekvenčního signálu (RF) z antény na mezifrekvenční signál (IF). Kvadratických detektorů existuje několik, varianta použitá v DR2B se nazývá **Taylor detektor** (dle autora Dana Tayloe N7VE). Do Taylor detektoru je přiveden signál z výstupu klopných obvodů, přesněji se jedná o 2 signály z obou výstupů Q', kde první je posunut o 0° a druhý o 90°. Tyto 2 signály zde slouží jako řídicí signály k přepínání vstupního vysokofrekvenčního signálu z antény na výstupy 74HC4053 [11; 12; 13].

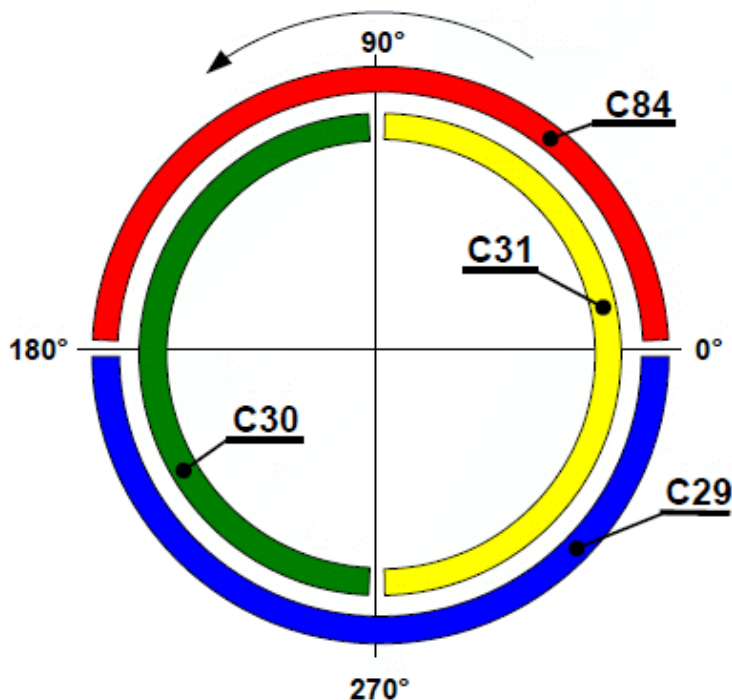


Obr. 13 – Schéma detektoru s 74HC4053 [11]

Taylor detektor se používá ve více SDR a stal se populární z několika hledisek [13]:

- Ztráta konverze je menší jak 1 dB. Když vezmeme v úvahu klasický diodový směšovač, který má ztráty kolem 7 dB, získáme oproti němu 6 dB zisk, což znamená dvojnásobnou úroveň užitečného signálu bez šumu.
- 4 vzorky signálu na periodu.
- Nízké hodnoty zrcadlových signálů. Díky tomu lze snížit požadavky na filtry, které by znamenaly další ztráty v obvodu.
- Vysoký činitel jakosti ($Q = 3500$ při 7 MHz).
- Kompaktní a jednoduchý návrh vzhledem k jiným řešením.

Integrátory a operační zesilovače – poslední stupeň v DR2B jsou integrátory tvořené z odporů a kondenzátorů. Tyto kondenzátory se při otevření jednotlivých výstupů nabíjejí vždy po dobu poloviny periody, čímž dochází ke vzniku vzorku přijímaného signálu. V operačních zesilovačích se poté vzorky signálu sečtou. Součtem vzorků z 0° a 180° výstupu v prvním operačním zesilovači dojde k vytvoření signálu „I“ (in-phase) a součtem vzorků z 90° a 270° výstupu vznikne signál „Q“ (quadrature) v operačním zesilovači druhém.



Obr. 14 – Nabíjení kondenzátorů v polárním zobrazení [11]

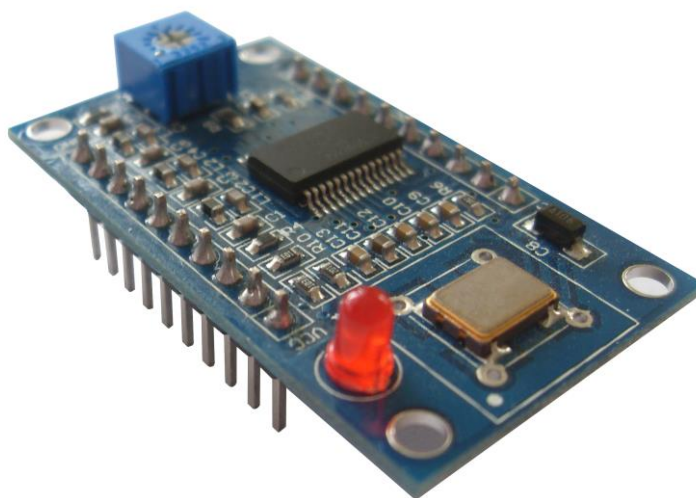
(Zvuková karta) - po všech krocích přichází na řadu už jen zvuková karta, která se stará o převedení vzorků do digitální podoby. Přijímaný frekvenční rozsah je dán vzorkovacím kmitočtem zvukové karty, který se běžně pohybuje kolem 48–192 KHz. Jelikož se vzorkuje okolo střední frekvence, je přijímaný rozsah např. u vzorkovací frekvence 192 KHz a 1 MHz signálu v rozsahu 0,904 – 1,096 MHz. Méně, ale také důležité je pak i rozlišení DAC převodníku, které je standardně 8–24 bit.

3 Oscilátor řízený Matlabem

SDR DR2B je návrh, který pro svojí funkci potřebuje generátor kmitočtu.

3.1 AD9850

AD9850 je přímý digitální frekvenční syntezátor neboli DDS (Direct Digital Synthesizer) od společnosti Analog Devices. Obecně frekvenční syntezátor slouží ke generování přesného, nejčastěji sinusového nebo obdélníkového signálu s proměnou frekvencí a fází. Je důležité zmínit, že výstupní signál se generuje za pomoci krystalového oscilátoru na vstupu, který má pevnou frekvenci. Analogový protějšek k frekvenčním syntezátorům bychom mohli najít například ve formě PLL (Phase Locked Loop) [14].



Obr. 15 – Pohled na AD9850 [vlastní]

3.1.1 Specifikace AD9850

Mezi hlavní parametry AD9850 jsou:

- 125 Mhz krystalový oscilátor na vstupu
- Příkon 380mW při 125Mhz (5 V)
- Příkon 155mW při 110Mhz (3,3V)

3.1.2 Výhody a nevýhody AD9850

Výhody [+]:

- Malá spotřeba (možnost napájení z USB)
- Rychlá změna frekvence
- Vysoká přesnost výstupní frekvence (zlomky Hz)
- Nízká cena hotového SMD modulu (Ebay, Aliexpress, ... cca 10€)

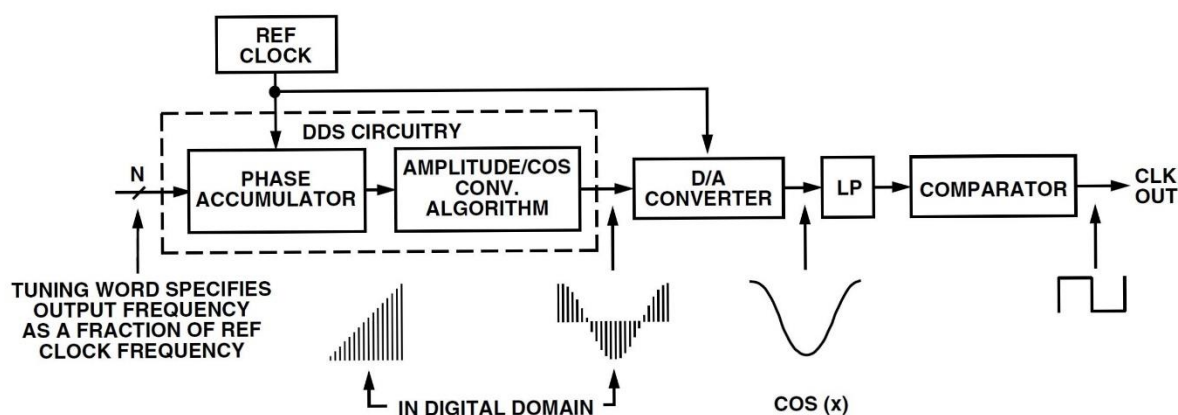
Nevýhody [-]:

- Poměrně nízký frekvenční rozsah 0-40 MHz – na vstupech většiny SDR DR od YU1LM/QRP se nachází multiplexor, který vstupní frekvenci dělí čtyřmi, což znamená, že maximální frekvence, která lze naladit na SDR je kolem 10 MHz

3.1.3 Poznámka

I když je možné z principu nastavit výstupní kmitočet na hodnotu vstupních hodin, je nutné při práci s AD9850 dbát na dodržení Nyquistova teoremu. Z toho plyne, že maximální výstupní kmitočet je poloviční, teoreticky tedy 62,5 MHz. Přímo od výrobce je však doporučeno, aby maximální frekvence nepřekračovala 33 % kmitočtu referenčních hodin, kvůli vzniku aliasingu. Prakticky lze tak AD9850 používat do kmitočtu kolem 40 MHz.

3.1.4 Popis funkčnosti AD9850



Obr. 16 – Blokové schéma AD9850 [14]

Když se podíváme na blokové schéma AD9850, zjistíme že obsahuje několik komponent, na kterých si nyní stručně popíšeme jeho funkci [14; 15]:

- 1) [REF CLOCK] referenční hodiny / oscilátor – jedna z nejdůležitějších složek frekvenčního syntezátoru, od které se odvíjí rozsah výstupní frekvence je krystalový oscilátor. Z této skutečnosti vyplývá, že přesnost výstupní frekvence je přímo daná přesností právě krystalového oscilátoru na vstupu. Oscilátor má pevně danou frekvenci, která se v případě AD9850 rovná 125 MHz.
- 2) [TUNING WORD] ladící kmitočet – kmitočet, kterého chceme pomocí frekvenčního syntezátoru generovat. Jedná se o 32-bit číslo, které je vztaženo k frekvenci referenčních hodin. Z frekvence referenčních hodin a velikosti čísla můžeme odvodit minimální krok změny frekvence:

$$\frac{f}{\text{word (bit)}} = \frac{125\,000\,000}{2^{32}} = 0,0291 \text{ [Hz, -, Hz]}$$

- 3) [PHASE ACCUMULATOR] fázový akumulátor – jedná se o srdce frekvenčního syntezátoru. V AD9850 a podobných syntezátorech je to tvořen modulo-M čítačem. Ten funguje na principu, že při každém cyklu vnitřních hodin přičítá k vnitřnímu stavu akumulátoru binární hodnotu ladícího kmitočtu. Jakmile hodnota dosáhne mezní hodnoty, která je rovna 2^{32} , dochází k přetečení akumulátoru a akumulátor začne dále počítat, avšak s počáteční hodnotou, která je rovna zbytku z přetečení. Zjednodušeně tak můžeme říci, že čím vyšší máme binární hodnotu ladícího kmitočtu, tím rychleji dochází k opětovnému přetečení akumulátoru.
- 4) [AMPLITUDE/COS CONV. ALGORITHM] – algoritmus pro převod akumulované hodnoty amplitudy na cosinusový průběh signálu.
- 5) [D/A CONVERTER] D/A převodník – rychlý 10-bit převodník k získání analogového průběhu signálu z digitálního.
- 6) [LP] Low Pass filter – dolní propust k potlačení signálu na harmonických frekvencích.
- 7) [COMPARATOR] komparátor – ze sinusového signálu se pomocí komparátoru vytvoří obdélníkový signál. Na většině prodávaných plošných spojů s AD9850 a jiných se nachází potenciometr, kterým lze ovládat prahová hodnota komparátoru, a tudíž měnit střihu (anglicky „duty cycle“) obdélníkového signálu.

3.1.5 Programování AD9850

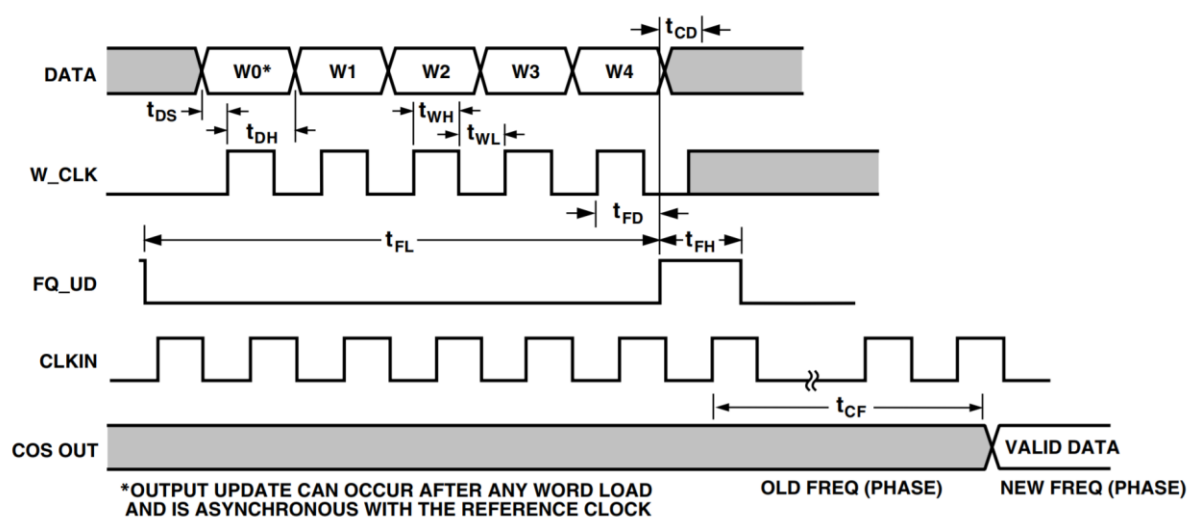
Programování AD9850 je možné provést dvěma způsoby, a to buď paralelně anebo sériově. Oba varianty spočívají v zápisu parametrů do 40 bitového registru, který je součástí AD9850 [14].

Piny, které se využívají pro zápis:

- W_CLK (Word Load Clock)
- FQ_UD (Frequency Update)
- D0 až D7 – pouze pro paralelní zápis
- DATA – pouze pro sériový zápis

Rozdělení 40 bitové slovo je rozděleno na:

- 32 frekvenčních bitů (Freq)
- 5 bitů fázové modulace (Phase)
- 2 řídicí bity (Control)
- 1 bit vypínací funkce (Power-Down)



Obr. 17 – Časování zápisu do registru [14]

Symbol	Definice	Minimální čas
t_{DS}	DATA SETUP TIME	3,5 ns
t_{DH}	DATA HOLD TIME	3,5 ns
t_{WH}	W_CLK HIGH	3,5 ns
t_{WL}	W_CLK LOW	3,5 ns
t_{CD}	CLK DELAY AFTER FQ_UD	3,5 ns
t_{FH}	FQ_UD HIGH	7,0 ns
t_{FL}	FQ_UD LOW	7,0 ns
t_{FD}	FQ_UD DELAY AFTER W_CLK	7,0 ns
t_{CF}	OUTPUT LATENCY FROM FQ_UD	
	FREQUENCY CHANGE	18 hodinových cyklů
	PHASE CHANGE	13 hodinových cyklů

Tabulka 2 – Časování zápisu do registru [14]

1) Paralelní zápis:

40 bitové slovo se v tomto případě nahraje do registru pomocí 8 bitové sběrnice reprezentované 8 vstupními piny D0 až D7. K nahrání celého slova je tedy zapotřebí jej rozdělit do 5 bytů (40 / 8). Při nahrávání 40 bitového slova do registru, je třeba mít na paměti, že všechny bity musí být seřazeny od nejvýznamnějšího bitu (MSB – Most Significant Bit) do nejméně významného bitu (LSB – Least Significant Bit) [14].

Zápis probíhá v následujícím pořadí:

Word	Data[7]	Data[6]	Data[5]	Data[4]	Data[3]	Data[2]	Data[1]	Data[0]
W0	Phase-b4 (MSB)	Phase-b3	Phase-b2	Phase-b1	Phase-b0 (LSB)	Power-Down	Control	Control
W1	Freq-b31 (MSB)	Freq-b30	Freq-b29	Freq-b28	Freq-b27	Freq-b26	Freq-b25	Freq-b24
W2	Freq-b23	Freq-b22	Freq-b21	Freq-b20	Freq-b19	Freq-b18	Freq-b17	Freq-b16
W3	Freq-b15	Freq-b14	Freq-b13	Freq-b12	Freq-b11	Freq-b10	Freq-b9	Freq-b8
W4	Freq-b7	Freq-b6	Freq-b5	Freq-b4	Freq-b3	Freq-b2	Freq-b1	Freq-b0 (LSB)

Obr. 18 – Paralelní zápis frekvence do AD9850 [14]

Paralelní zápis má následující kroky:

- a. Vzestupnou hranou na pin FQ_UD zahájíme nahrávání 40 bitového slova. V AD9850 dojde k posunutí ukazatele na první registr.
- b. Na 8 bitovou sběrnici (piny D0 až D7) přivedeme první byte.
- c. Vzestupnou hranou na pin W_CLK dojde k nahrání bytu do prvního registru a posunutí ukazatele na registr druhý.
- d. Opakujeme první 2 body ještě 4x, abychom nahráli celé 40 bitové slovo.
- e. Vzestupnou hranou na pinu FQ_UD dojde k posunutí ukazatele zpět na první registr a AD9850 nastaví frekvenci výstupu dle dat z registru.

2) Sériový zápis:

Jak už z názvu vyplývá, zápis 40 bitového slova probíhá sériově, a to prostřednictvím pinu DATA. Na rozdíl od paralelního zápisu je pořadí celého slova otočené, tudíž zápis začíná nejméně významným bitem (LSB – Least Significant Bit) a končí nejvýznamnějším bitu (MSB – Most Significant Bit) [14].

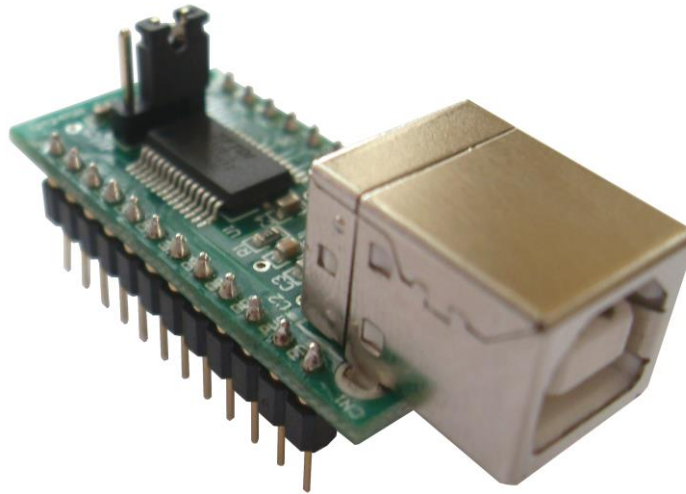
W0	Freq-b0 (LSB)	W14	Freq-b14	W28	Freq-b28
W1	Freq-b1	W15	Freq-b15	W29	Freq-b29
W2	Freq-b2	W16	Freq-b16	W30	Freq-b30
W3	Freq-b3	W17	Freq-b17	W31	Freq-b31 (MSB)
W4	Freq-b4	W18	Freq-b18	W32	Control
W5	Freq-b5	W19	Freq-b19	W33	Control
W6	Freq-b6	W20	Freq-b20	W34	Power-Down
W7	Freq-b7	W21	Freq-b21	W35	Phase-b0 (LSB)
W8	Freq-b8	W22	Freq-b22	W36	Phase-b1
W9	Freq-b9	W23	Freq-b23	W37	Phase-b2
W10	Freq-b10	W24	Freq-b24	W38	Phase-b3
W11	Freq-b11	W25	Freq-b25	W39	Phase-b4 (MSB)
W12	Freq-b12	W26	Freq-b26		
W13	Freq-b13	W27	Freq-b27		

Obr. 19 – Sériový zápis frekvence do AD9850 [14]

Sériový zápis probíhá v následujícím pořadí:

- a. Vzestupnou hranou na pin FQ_UD zahájíme nahrávání 40 bitového slova. V AD9850 dojde k posunutí ukazatele na první registr.
- b. Na pin DATA přivedeme první bit.
- c. Vzestupnou hranou na pin W_CLK dojde k nahrání bitu do prvního registru a posunutí ukazatele na registr druhý.
- d. Opakujeme první 2 body ještě 39x, abychom nahráli celé 40 bitové slovo.
- e. Vzestupnou hranou na pinu FQ_UD dojde k posunutí ukazatele zpět na první registr a AD9850 aktualizuje frekvenci na výstupu dle dat v registru.

3.2 FTDI UM245R



Obr. 20 – Pohled na FTDI UM245R [vlastní]

FTDI UM245R je modul od společnosti FTDI (Future Technology Devices International). Modul slouží k vytvoření rozhraní z USB na paralelní FIFO (First In First Out) port. Pro svoji funkci využívá integrovaný obvod FR245RL.

Bit Bang Mode – přepne FT245RL do módu, kdy je možné klasickou paralelní sběrnici používat jako 8 vstupně/výstupních GPIO (General Purpose input/output) pinů.

Mezi nejdůležitější parametry patří [16]:

- 24 pin DIP pouzdro s 15 mm šířkou
- USB -> paralelní oboustranná komunikace
- přenosová rychlost až 300 kB/s s VCP ovladači
- přenosová rychlost až 1 MB/s s D2XX ovladači
- asynchronní i synchronní big bang mode
- integrovaná 1024 bit EEPROM

Ovladače:

FTDI nabízí k programování UM245R 2 typy ovladačů:

- 1) [VCP – VIRTUAL COM PORT] – VCP ovladač zajistí, že se UM245R chová jako klasický COM port. Zařízení je tedy přístupné jako jakékoliv jiné COM port zařízení/převodník.
- 2) [D2XX Direct Drivers] – umožňuje přímý přístup k zařízení pomocí DLL knihoven. Software poté komunikuje se zařízením voláním jedné z několika DLL funkcí/metod.

3.2.1 Ovládání UM245R z Matlabu.

Komunikace s modulem FTDI UM245R není v programovém prostředí Matlab přímo podporována. Pro komunikaci s UM245R je použit .NET Wrapper, který je volně ke stažení přímo na stránkách výrobce [17].

Ten slouží jako prostředník v komunikaci mezi programem, který jej volá a daným zařízením, které chceme ovládat. V prostředí Matlabu se po načtení NET Wrapperu zpřístupní třída objektu, spolu s metodami, jak lze zařízení volat. Pro ovládání zařízení, v našem případě UM245R, stačí už jen ve skriptu volat tyto metody (např. metoda Write pro zápis).

Postup ovládání UM245R z prostředí Matlab:

- 1) Nejdříve je nutné načíst .NET Wrapper (FTD2XX_NET.dll musí být ve složce se skriptem):

```
NET.addAssembly([mfilename('fullpath'), '..\FTD2XX_NET.dll']);  
UM245R = FTD2XX_NET.FTDI;
```

- 2) Dostupné metody můžeme zobrazit zadáním příkazu:

```
methodsview(UM245R);
```

- 3) Připojení k zařízení (0 = první zařízení):

```
OpenByIndex(UM245R, 0);
```

- 4) Nastavení masky (I/O), BitBang módu:

```
mask = uint8(255);           % '11111111' = výstupy  
mode = uint8(1);            % '1' = bitbang mód  
x = SetBitMode(UM245R, mask, mode);
```

- 5) Zápis

```
Write(UM245R, <byte který chceme zapsat>, 1, 0);
```

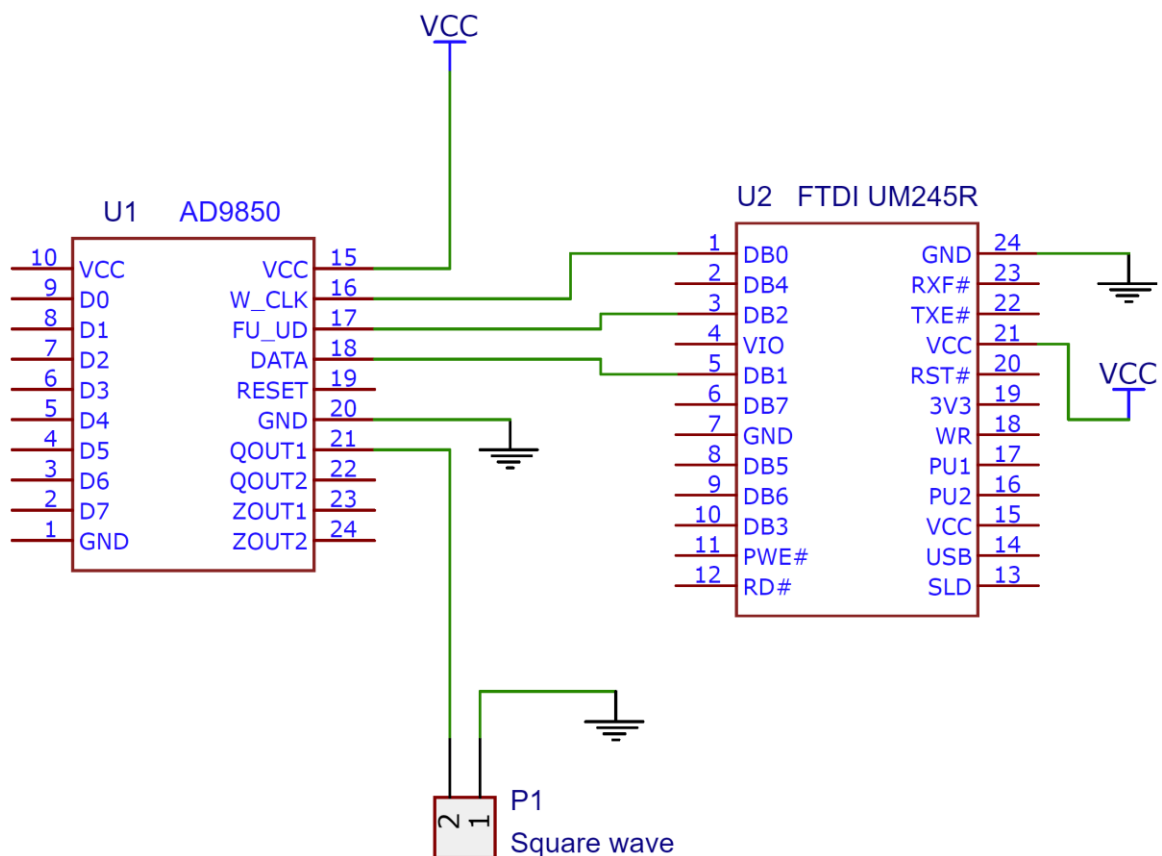
- 6) Ukončení komunikace

```
Close(UM245R)
```

3.3 Frekvenční generátor přes USB (UM245R + AD9850)

AD9850 vyžaduje pro paralelní zápis frekvence 8 stavových a 2 řídicí bity (piny). Pro sériový zápis je potřeba pouze 1 stavový a 2 řídicí bity (piny). UM245R je zařízení s 8 vstupně-výstupními piny, tudíž zápis frekvence do registru AD9850 nelze v našem případě provést paralelně a jediný možný způsob je sériový zápis.

Na obrázku níže je schéma zapojení frekvenčního generátoru. Jak už bylo zmíněno v popisu AD9850, je tento integrovaný obvod napájen přímo z USB portu přes UM245R. Toto je možné díky jeho nízkému příkonu 380mW při 125Mhz (5 V).



Obr. 21 – Schéma frekvenčního generátoru s UM245 a AD9850 [vlastní]

Pokud aplikujeme získané informace z předchozích kapitol, respektive jak ovládat UM245R v prostředí Matlab a jak provést zápis frekvence do registru AD9850, můžeme vytvořit skript pro Matlab viz. níže. Ten slouží pro připojení UM245R, zápisu frekvence 10 MHz do registru AD9850 a následné ukončení komunikace.

```

NET.addAssembly([mfilename('fullpath'), '..\FTD2XX_NET.dll']);

UM245R = FTD2XX_NET.FTDI;
OpenByIndex(UM245R,0);
mask = uint8(255);      % '11111111' = výstupy
mode = uint8(1); % '1' = bitbang mód
x = SetBitMode(UM245R, mask, mode);

W_CLK = 0; % W_CLK -> UM245R pin #
DATA = 1; % DATA -> UM245R pin #
FU_UD = 2; % FU_UD -> UM245R pin #

bit_W_CLK = uint8(de2bi(2.^W_CLK,8));
bit_DATA = uint8(de2bi(2.^DATA,8));
bit_FU_UD = uint8(de2bi(2.^FU_UD,8));

%FQ_UD -> start seriového přenosu
byte_send = bit_FU_UD;
write = Write(UM245R, bi2de(byte_send), 1, 0);
write = Write(UM245R, 0, 1, 0);

% Výpočet frekvence = <sys clock> * <frequency tuning word>/2^32
frekvence = 10e6;
freq = uint32(frekvence * 4294967295/125000000);
freq_bit = de2bi(freq,32);

% Zápis frekvence - zapisují se bity uspořádané do bytu (8 I/O portů)
for bit = 1:32
    binary = uint8(freq_bit(bit));

    byte_send = binary .* bit_DATA;
    write = Write(UM245R, bi2de(byte_send), 1, 0);

    byte_send = bitor(binary .*bit_DATA, bit_W_CLK);
    write = Write(UM245R, bi2de(byte_send), 1, 0);

    byte_send = binary .* bit_DATA;
    write = Write(UM245R, bi2de(byte_send), 1, 0);
end

% Zápis řídicích, vypínacích, a fázových bitů -> vše 0
for bit = 1:8
    byte_send = bit_W_CLK;
    write = Write(UM245R, bi2de(byte_send), 1, 0);
    write = Write(UM245R, 0, 1, 0);
end

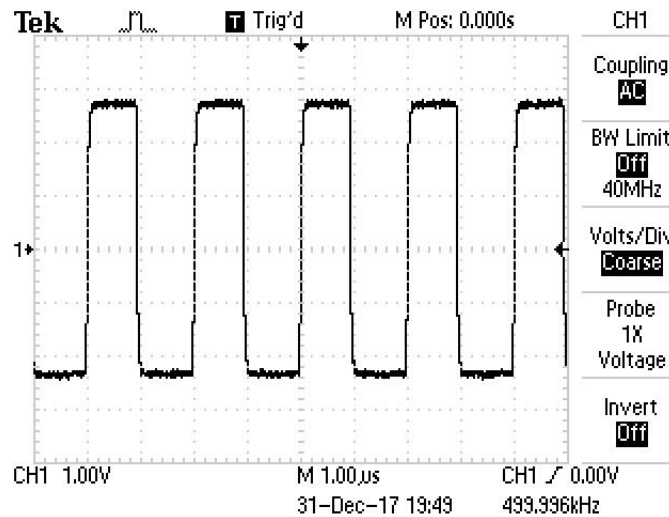
% Nastav frekvenci (FQ_UD puls)
byte_send = bit_FU_UD;
write = Write(UM245R, bi2de(byte_send), 1, 0);
write = Write(UM245R, 0, 1, 0);

Close(UM245R)

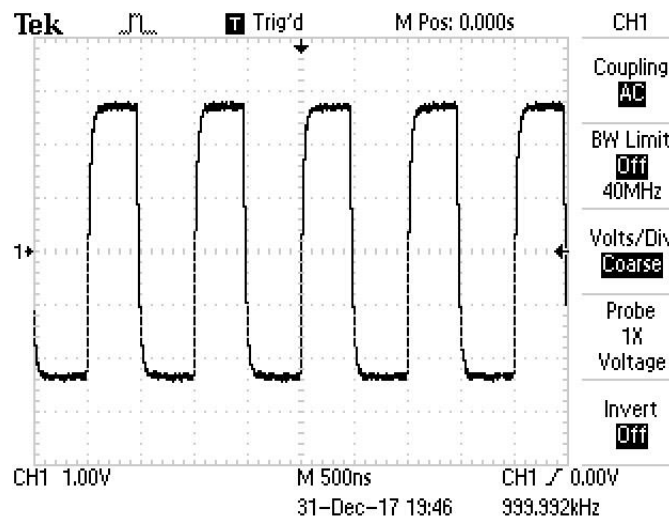
```

3.3.1 Průběhy z frekvenčního generátoru

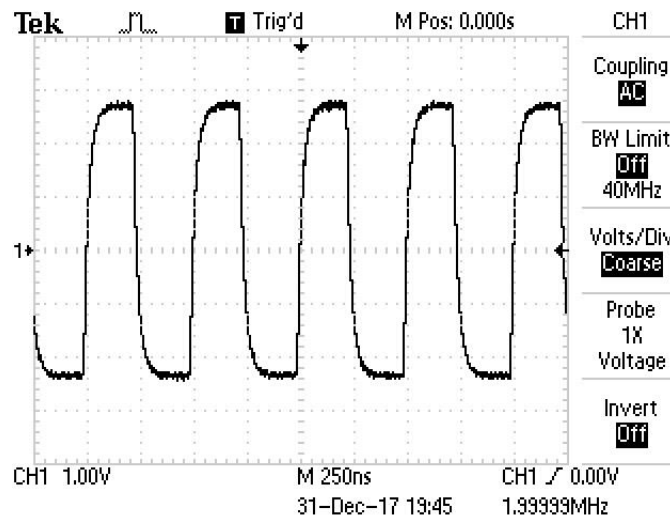
Níže jsou zobrazeny průběhy signálu z obdélníkového výstupu AD9850 od 500 KHz do 10 MHz. V popisu obrázku je hodnota nastavené frekvence, vpravo dole v obrázku je frekvence zobrazená oscilátorem. Jak si můžeme všimnout je rozdíl mezi frekvencí nastavenou a generovanou naprosto zanedbatelný, vzhledem k použití generátoru v našem projektu.



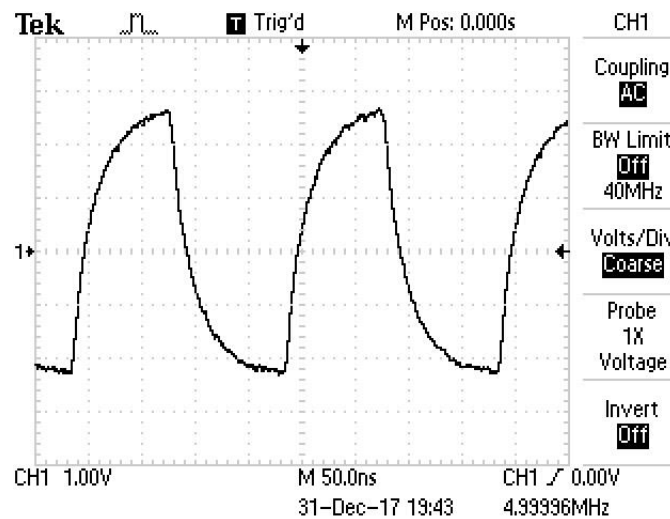
Obr. 22 – Frekvenční generátor 500 KHz [vlastní]



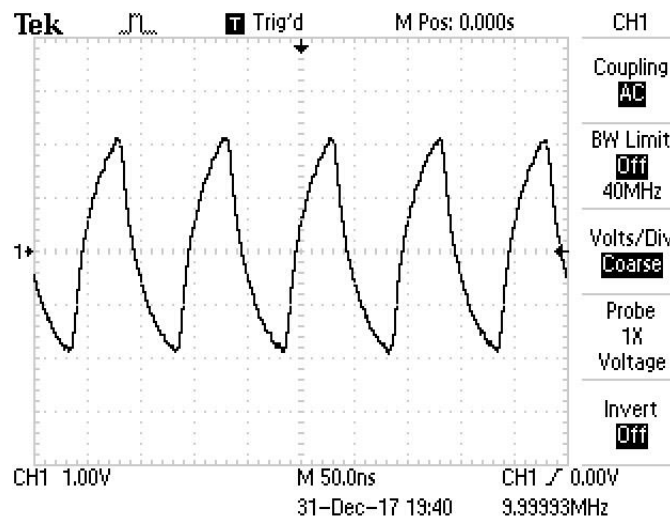
Obr. 23 – Frekvenční generátor 1 MHz [vlastní]



Obr. 24 – Frekvenční generátor 2 MHz [vlastní]



Obr. 25 – Frekvenční generátor 5 MHz [vlastní]



Obr. 26 – Frekvenční generátor 10 MHz [vlastní]

3.4 SDR DR2B s oscilátorem řízeným přes USB

Výsledkem předchozích kapitol může být vytvořený modul jako na obrázku níže.



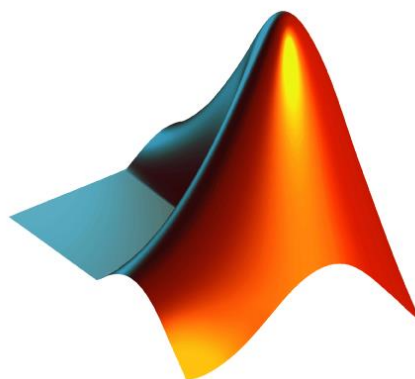
Obr. 27 – SDR DR2B s oscilátorem řízeným přes USB 2 [vlastní]



Obr. 28 – SDR DR2B s oscilátorem řízeným přes USB 2 [vlastní]

4 SDR v Matlabu

Matlab je skriptovací jazyk a zároveň interaktivní nástroj pro řešení komplexních úloh. Program Matlab je vyvíjen a udržován společností MathWorks. První Matlab spatřil světlo světa už v roce 1984 a od té doby je neustále zlepšován a zdokonalován. V oblíbě je hlavně mezi studenty vysokých škol a vědeckými pracovníky. Jeho největší síla oproti konkurenčním nástrojům tkví v obrovském množství oborů, v kterých lze Matlab aplikovat. Nabídka oborů je navíc s každou novou verzí Matlabu rozšiřována dle současných technologických trendů, tak aby byl Matlab stále aktuální. Mezi oblastí, v kterých můžeme Matlab použít patří například: matematika, statistika, řízení, zpracování signálu, bezdrátové komunikace, zpracování obrazu, finance a mnoho dalších. Další jeho obrovská výhoda spočívá v rychlosti programování. Matlab v tomto ohledu nefunguje zcela jako ostatní skriptovací či programovací jazyky, kde se k výsledku musíme dopracovat tvořením složitých kódů a funkcí. Matlab nabízí velké množství již připravených funkcí a postupů, tudíž programování je spíše formou skládání jednotlivých částí dohromady, abychom vytvořili funkční celek [18].



Obr. 29 – Matlab logo [28]

4.1 Toolboxy

Matlab v základu nabízí pouze omezené možnosti. Jeho funkcionalitu můžeme rozšířit pomocí Toolboxů. Toolboxy jsou stejně jako Matlab vyvíjené přímo společností MathWorks a jedná se o rozšíření placená, které se instalují přímo s instalací Matlabu či dodatečně.

K našemu účelu, tedy zprovoznění SDR v Matlabu je nutné nainstalovat několik Toolboxů. Potřebné Toolboxy se liší dle toho, jestli chceme pracovat s RTL-SDR připojeným přes USB, či s klasickým SDR připojeným přes zvukovou kartu.

Toolboxy pro SDR:

- **Audio System Toolbox** (pouze u SDR přes zvukovou kartu) – zpracování vstupního signálu z mikrofonního či line-in portu na zvukové kartě.
- **Communication Toolbox** – demodulace FM signálu.
- **DSP System Toolbox** – čtení/záznam IQ signálu pomocí souboru, spektrální analyzátor v samostatném okně.
- **Signal Processing Toolbox** – filtrace signálu (dolní propust).

4.2 Podpůrné balíčky (Support Packages)

Další možností, jak Matlab rozšířit je nainstalováním podpůrných balíčků (Support Packages). Tyto balíčky většinou přidávají podporu pro určitý hardware, např. RTL-SDR či Arduino. Podpůrné balíčky jsou sice volně ke stažení, ale mnohé z nich se neobejdou bez přítomnosti určitých Toolboxů.

Z hlediska SDR je potřeba pouze jediný a to **RTL-SDR Support from Communications System Toolbox** [19], který umožňuje přímou komunikaci s RTL-SDR a tedy získání I/Q signálu pro další zpracování.

5 Real-time zpracování signálu v Matlabu

Signál z SDR není statický, ale jedná se o kontinuální signál, který je přiváděn buď přes USB či přes zvukovou kartu. Abychom jej mohli zpracovávat je nutné jednotlivé příchozí vzorky ukládat do paměti. Po uložení X vzorků z nich vytvoříme tzv. rámec (frame), s kterým lze pracovat, jako by se jednalo o signál statický. Rámce jsou postupně vytvářeny vždy po době, která je daná vzorkovacím kmitočtem a velikostí rámce [20].

$$\text{doba mezi rámci} = \frac{\text{vzorkovací kmitočet}}{\text{velikost vzorku}} \text{ [s; Hz; -]}$$

Než uplyne doba do vytvoření dalšího rámce, měl by už být předchozí rámec zpracovaný, aby nedocházelo ke ztrátě dat. Vytváření a zpracování rámců už je defaultně podporováno v toolboxech, které při vytváření aplikace v pro SDR využijeme.

Real-time zpracování signálu probíhá ve 3 krocích [21]:

- 1) **Vytvoření systémových objektů** – objekty se vytváří na začátku skriptu. Patří mezi ně vstupy a výstupy signálu, filtry, scope a další. Objekty mají při vytváření defaultně nastavené parametry, ty lze ale dle potřeby měnit, a to zadáním názvu parametru a jeho hodnoty do závorky za objekt, například:

```
input = comm.SDRRTLReceiver('CenterFrequency', 89.0e6);
```

- 2) **Volání objektů** – zde dochází k volání vstupního systémového objektu, čímž dojde k vytvoření a načtení rámce. Následuje zpracování rámce (filtrace, demodulace, ...). Nakonec se v tom samém cyklu zpracovaný rámec pošle na výstupní objekt. Volání objektů v Matlab 2016b a novějším se provádí pomocí „<objektu, který signál zpracovává>(<signál>)“:

```
signal = input();  
signal_lpf = lpf(signal);  
output(signal_lpf);
```

- 3) **Uvolnění systémových objektů** – uvolnění systémových prostředků a spojení, které jsou systémovými objekty využívány.

5.1 Vytvoření systémových objektů

5.1.1 Vstupní zařízení

Zobrazení všech zařízení pro čtení:

```
read_devices = getAudioDevices(audioDeviceReader);
```

Objekt pro RTL-SDR přijímač:

```
input = comm.SDRRTLReceiver(...  
    '0',...  
    'CenterFrequency', 89.0e6,...  
    'SampleRate', 2.4e6,...  
    'SamplesPerFrame', 2.4e3,...  
    'EnableTunerAGC', true,...  
    'OutputDataType', 'double');
```

Objekt FM demodulátoru využívající RTL-SDR přijímač:

```
demod_broadcast = comm.FMBroadcastDemodulator(...  
    'SampleRate', 2.4e6, ...  
    'FrequencyDeviation', 75000, ...  
    'AudioSampleRate', 48000, ...  
    'Stereo', true);
```

Objekt pro vstupní směšovač (mikrofon, nebo line-in vstup):

```
input = audioDeviceReader(...  
    'Device', read_devices{1},...  
    'NumChannels', 2,...  
    'SamplesPerFrame', 2.4e3,...  
    'SampleRate', 192e3,...  
    'BitDepth', '16-bit integer');
```

Objekt pro čtení ze souboru:

```
input = dsp.AudioFileReader(...  
    'test_vstup.wav',...  
    'SamplesPerFrame', 192e3,...  
    'PlayCount', inf);
```

5.1.2 Výstupní zařízení

Zobrazení všech zařízení pro zápis:

```
write_devices = getAudioDevices(audioDeviceWriter);
```

Objekt pro výstupní směšovač (reproduktory, sluchátka, ...):

```
output = audioDeviceWriter(...  
    'Device', write_devices{1}, ...  
    'SampleRate', 48e3);
```

Objekt pro zápis do souboru:

```
output = dsp.AudioFileWriter(...  
    'FileFormat', 'WAV', ...  
    'test_vystup.wav', ...  
    'SampleRate', 48e3);
```

5.1.3 Filtry, demodulátory

Objekt dolní propusti vytvořené pomocí FIR filtru:

```
lpf = dsp.LowpassFilter(...  
    'SampleRate', 2.4e6, ...  
    'FilterType', 'FIR', ...  
    'PassbandFrequency', pass, ...  
    'StopbandFrequency', stop, ...  
    'PassbandRipple', 0.1, ...  
    'StopbandAttenuation', 50);
```

Objekt FM demodulátoru:

```
demod = comm.FMDemodulator(...  
    'SampleRate', 2.4e6, ...  
    'FrequencyDeviation', 75000);
```

5.2 Smyčka

```
while true  
  
    signal = input();  
    signal_lpf = lpf(signal);  
    output(signal_lpf);  
  
end
```

5.3 Uvolnění systémových objektů

```
release (input);  
release (output);
```

6 GUI v Matlabu

GUI neboli grafické uživatelské rozhraní slouží pro interakci mezi uživatelem a aplikací. Pomocí uživatelského rozhraní můžeme například v běžícím skriptu provádět změny jednoduchým kliknutím na obrazovku, což by jinak muselo být řešeno zadáním příkazu do příkazového řádku. Již z tohoto plynou výhody jako rychlejší ovládání aplikace a také, že člověk interagující s vytvořeným uživatelským rozhraním nemusí mít znalosti kódu aplikace či samotného Matlabu.

6.1 Metody vytváření GUI

Matlab pro vytvoření GUI nabízí hned 3 způsoby – App Designer, GUIDE, programové GUI. Tyto metody jsou popsány v následujících kapitolách [22; 23].

6.1.1 App Designer

První a zároveň nejmodernější způsob vytváření GUI představuje App Designer. Ten se spouští z Matlabu zadáním příkazu [appdesigner] do příkazového řádku. Vytvořená aplikace je uložena pouze v jediném [*.mlapp] souboru.

Klady [+]:

- V okně App Designeru jsou integrovány oba primární elementy při vývoji aplikace, a to vytváření grafických prvků rozhraní v jedné záložce a psaní kódu v záložce druhé.
- Vytvořením grafického prvku dojde automaticky ke generování funkce, která je daným prvkem volána (callback).
- App Designer spravuje část kódu tak, že ji není možné upravit. Tím se předejde, aby uživatel smazal důležitou část kódu.

Zápory [-]:

- Poslední kladný bod může být pro zkušeného uživatele záporný, jelikož není možné upravovat kód do funkčních celků, tak aby byl přehledný.
- Není možná editace [*.mlapp] souboru jiným programem než App Designer.

6.1.2 GUIDE

GUIDE je zkratka pro Graphical User Interface Development Environment neboli v překladu „vývojové prostředí pro grafické uživatelské rozhraní“. GUIDE se spouští zadáním příkazu [guide] do příkazového řádku Matlabu. Po spuštění se otevře okno, ve kterém se umísťují jednotlivé prvky GUI. Oproti App Designeru je výsledná aplikace tvořena dvěma soubory, první je klasický Matlab script [*.m], druhý je soubor s GUI prvky [*.fig]. Podmínkou pro fungování aplikace je, aby oba soubory měli stejný název.

Klady [+]:

- Umožňuje úplnou kontrolu nad kódem, včetně mazání volaných funkcí (callback).
- Vytvořením grafického prvku dojde automaticky ke generování funkce, která je daným prvkem volána (callback).
- Velké množství parametrů, které lze u prvku změnit.

Zápory [-]:

- Prvky GUI jsou méně vzhledné než u App Designer.

6.1.3 Programové GUI

Poslední způsob vytváření GUI je pomocí psaní kódu, který definuje parametry elementů přímo v Matlab scriptu. Výsledná aplikace je tak obsažena v jednom [*.m] souboru.

Klady [+]:

- Umožňuje vytvořit pixel-perfect uživatelské rozhraní.

Zápory [-]:

- Při velkém počtu prvků je tato metoda nejvíce časově náročná jak na vytvoření, tak úpravu GUI.

6.1.4 Výběr nejvhodnějšího GUI

Při výběru GUI metody jsem se rozhodl pro GUIDE z těchto důvodů:

- Metodu programového vytváření GUI jsem zavrhl, jelikož by byla obtížně udržovatelná při počtu 50+ prvků GUI.
- V App Designeru mi nevyhovuje nemožnost upravovat celý kód.
- Pomalá odezva aplikace v App Designeru. Totožná aplikace vytvořená pomocí GUIDE byla daleko rychleji reagující (Matlab R2016b).

6.2 GUI pomocí GUIDE

Následující kapitola popisuje nejdůležitější prvky při vytváření GUI. GUI se v tomto textu rozumí uživatelské rozhraní vytvořené pomocí metody GUIDE, pokud není uvedeno jinak [24].

6.2.1 Funkce Callback

Nejdůležitější i nejtěžší na pochopení je, jak funguje práce s proměnnými uvnitř GUI.

Callback je v Matlabu nazývána funkce, která je vyvolána objektem v GUI. Volání může být například kliknutím myši na tlačítko, puštění tlačítka myši, stisknutí klávesy, puštění klávesy, změna položky v menu a dalších. Pokud vytváříme GUI pomocí GUIDE nemusíme se o vytváření callback funkce starat, je automaticky vytvořena po přidání objektu a uložení GUI.

Všechny Callback funkce mají 3 vstupní argumenty:

```
function testovaci(hObject, eventdata, handles)
```

hObject – objekt v GUI, který callback zavolal. Na první pohled se nemusí tato proměnná jevit jako užitečná, jelikož všechny informace o objektech jsou uloženy v proměnné handles. Avšak díky této proměnné můžeme několika podobnými objekty volat pouze jeden callback a uvnitř něho poté rozlišit jaký objekt jej vyvolal, čímž se vyhneme abychom v kódu měli X téměř stejných callback funkcí.

eventdata – proměnná obsahující informace o akci, která callback zavolala. Akcí se zde rozumí kliknutí myši, nebo stisknutí kláves na klávesnici.

handles – struktura, která obsahuje všechny objekty v daném GUI, to znamená jejich popis, vzhled, vnitřní stavy proměnných a dalších. Handles však neslouží pouze pro tento účel, ale můžeme do něj ukládat vlastní proměnné, které chceme sdílet mezi funkcemi. Struktura handles uvnitř callback funkce je pouze kopií originálních handles, která se vytváří při zavolání callback funkce a zaniká po jejím ukončení. Pokud v této struktuře uděláme změny a neaktualizujeme lokální handles do struktury GUI, přijdeme o tyto změny. K uložení změn či získání aktuálních dat z GUI se používá funkce guidata.

Uložení kopie handles do GUI se provede pomocí kódu níže. Prvním argumentem udáváme cíl a druhým argumentem zdroj dat [25].

```
guidata(hObject, handles)
```

Pro získání aktuálních handles z GUI pak použijeme funkci dle kódu níže. Do argumentu zadáváme pouze zdroj dat.

```
handles = guidata(hObject)
```

Na funkcích výše si můžeme všimnout, že je v obou případech použita proměnná hObject, která odkazuje pouze na jediný objekt v GUI, což by nedávalo smysl, když chceme pracovat s celou strukturou GUI. Funkce guidata však pracuje tak, že pokud volaným argumentem není přímo GUI, poté je použito GUI, v kterém se daný objekt nachází.

6.2.2 Struktura

Jakýkoliv kód uvnitř GUI musí být uvnitř nějaké funkce. Část kódu umístěna mimo funkci, vyústí v chybu v kódu. Pro lepší přehled je možné použít na konci funkce ukončovací znak end, avšak toto není podmínkou, důležité je, aby ukončovací znak (ne-) byl u každé funkce v aplikaci.

Při vytváření GUI narazíme na 3 různé části:

- 1) Inicializační funkce.
- 2) Callback funkce vztažené ke GUI.
- 3) Callback funkce vztažené k objektům v GUI.

Nově vytvořené GUI obsahuje 3 základní funkce, které jsou povinné pro všechna GUI [26]:

Inicializační funkce – její obsah se neupravuje, ale je spravován automaticky pomocí GUIDE. Ve funkci jsou přítomny callback funkce, které jsou vztaženy přímo ke GUI (ne k objektům v GUI), například: CreateFcn, DeleteFcn, SizeChangedFcn, CloseRequestFcn a dalších.

```
function varargout = defaultni(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @defaultni_OpeningFcn, ...
                  'gui_OutputFcn',  @defaultni_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

OpeningFcn – je druhá funkce a zároveň i první callback funkce. Tato funkce se provede ještě před tím, než se zobrazí GUI pro uživatele. Mohou se zde vytvářet proměnné v handles struktuře, například nastavení vstupních parametrů aplikace, nebo načtení dat z externího zdroje.

```
function defaultni_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);
```

OutputFcn – je třetí a poslední callback funkce vytvořená v prázdném GUI. Funkce slouží ke generování výstupu z proměných do příkazového řádku.

```
function varargout = defaultni_OutputFcn(hObject, eventdata, handles)

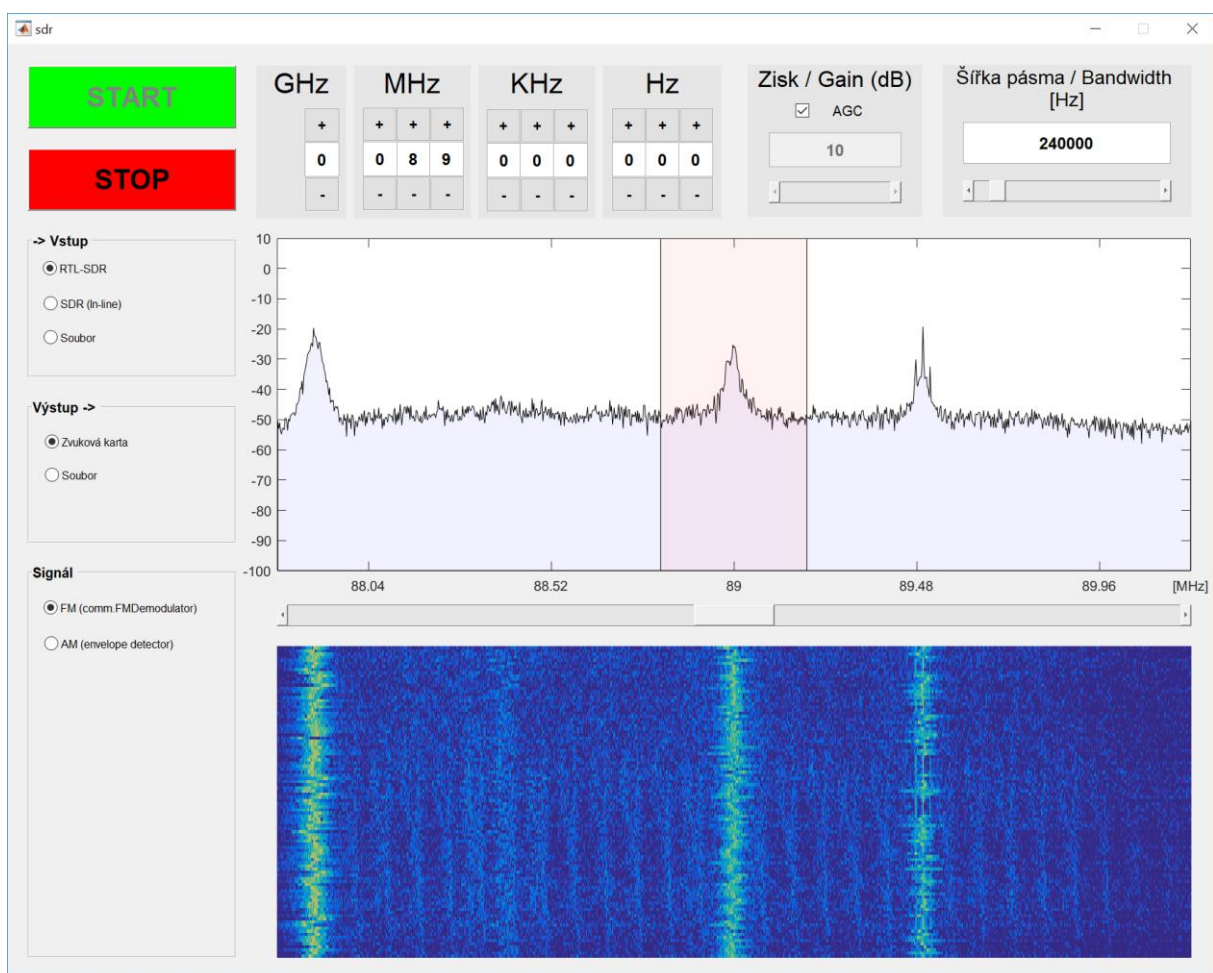
varargout{1} = handles.output;
```

7 Aplikace v Matlabu

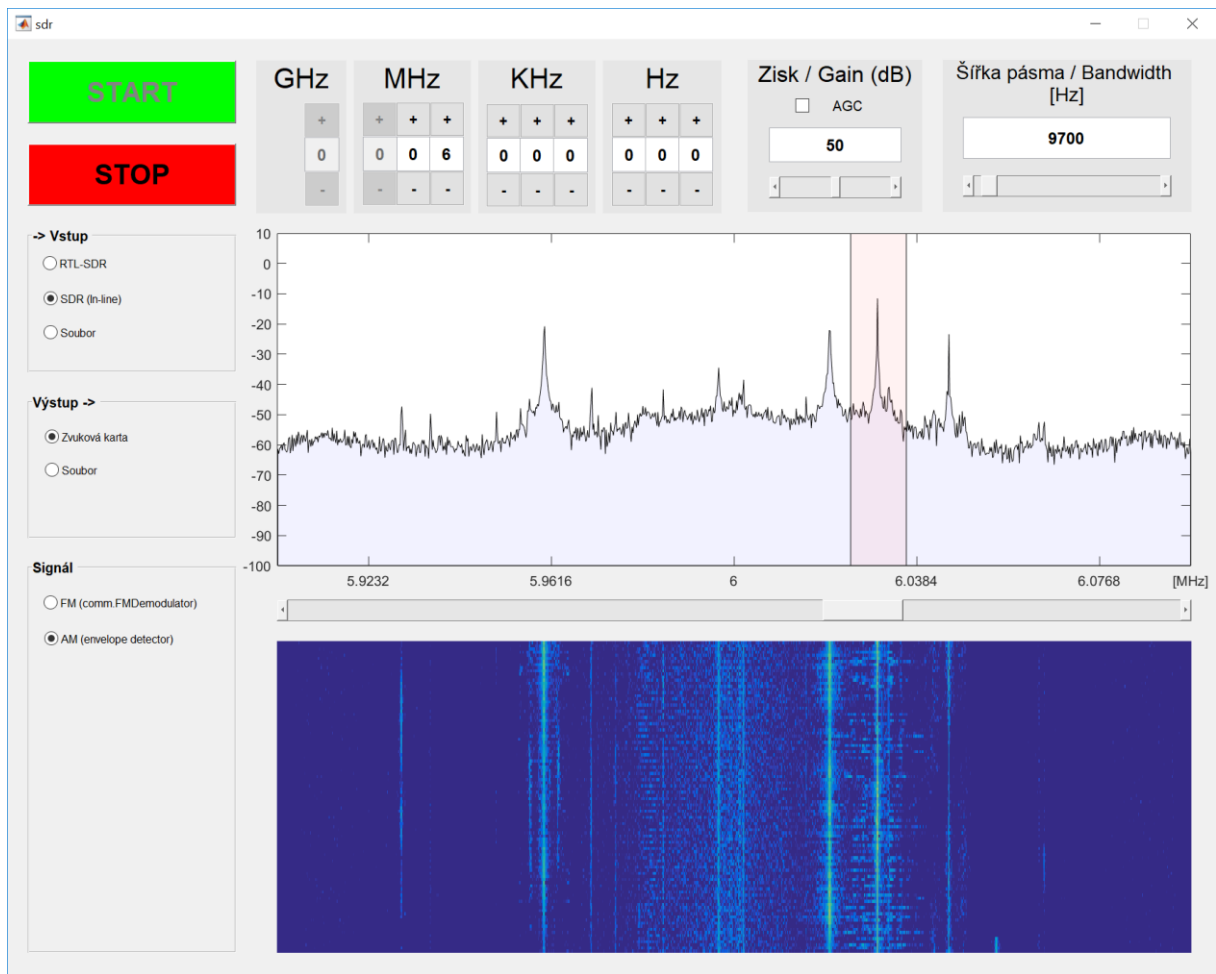
Pro demonstraci real-time zpracování signálu z SDR v propojení s GUI byla vytvořena aplikace, která umožňuje:

- Zpracování signálu z daného vstupu a výstupu.
- Real-time změna demodulace.
- Real-time změna ladící frekvence.
- Real-time změna zisku (popř. AGC).
- Real-time změna polohy filtrovaného pásma v FFT (červené pole).
- Real-time změna šířky pásma.
- Zobrazení průměrované FFT a Waterfall.

První obrázek je příjem pomocí RTL-SDR s generickou anténou dodávanou s donglem. Druhý obrázek je z SDR DR2B s anténou Miniwhip, která je popsána v poslední kapitole této práce.



Obr. 30 – Aplikace v Matlabu – RTL-SDR, 89 MHz [vlastní]



Obr. 31 – Aplikace v Matlabu – SDR DR2B, 6 MHz [vlastní]

8 Příslušenství

8.1 Anténa Miniwhip

Abychom byli schopni získat signál, který se zpracuje pomocí SDR a následně aplikací v Matlabu, je potřeba anténa. Pro tyto účely jsem zvolil aktivní anténu Miniwhip, která už je dnes legendární mezi radioamatéry. První článek o této anténě se objevil v časopisu Vernon, ročník 2006 a autorem byl Roelof Bakker, neboli PA0RDT. Design této byl od té doby kopírován a upravován, že za několik let vzniklo mnoho variant této antény. Dnes jsou tyto varianty dostupné na internetu, buď pro vlastní výrobu, nebo ve formě stavebnic. [27]

Technické specifikace:

- Frekvenční rozsah 10 KHz – 20 (30) MHz.
- Napájení 12-15 V.
- Příkon 150 mA.

Klady [+]:

- Ideální frekvenční rozsah pro SDR DRxxx.
- Kompaktní rozměry.

Zápory [-]:

- Součástky z některých návrhů mohou být těžko k dostání.
- Pro co nejlepší příjem se doporučuje umístění na nevodivý stožár, ideálně do výšky 6 m nad vodivou plochou. Toto může být překážka pro někoho, kdo bydlí například ve městě. Přijatelný signál, lze získat instalací antény na nevodivou tyč připevněnou například na balkon či z okna.

9 Závěr

Cílem této práce bylo zprovoznění SDR radiopřijímače v Matlabu. SDR radiopřijímačem se v tomto smyslu rozumí 2 typy. První typ je RTL-SDR, tedy SDR vytvořené z DVB-T dongle do USB. Druhý typ SDR má podobu SDR DR2B.

K dosažení tohoto cíle bylo nejprve nutné zprovoznit generátor frekvence pro SDR DR2B, aby jej bylo možné naladit a přijímat signál. Tento úkol byl vyřešen prostřednictvím frekvenčního generátoru AD9850, programovaného z UM245R, který je následně připojen přes klasický USB port. Jelikož Matlab nenabízí pro zařízení FTDI podporu, byla komunikace zprostředkována pomocí NET Wrapperu.

Následně bylo možné připojit oba SDR radiopřijímače a pracovat s nimi v Matlabu. Pro práci s nimi bylo využito real-time zpracování signálu, které je podporováno mnohými toolboxy v Matlabu.

Ovládání aplikace přes příkazový řádek nebylo ideální. Z tohoto důvodu bylo v Matlabu vytvořena aplikace pomocí prostředí Guide pro vytváření GUI.

Pokud bych měl navrhnout náměty pro řešení podobných projektů s SDR DR2B, asi bych doporučil vyzkoušení zesilovače pro signál vedený z AD9850, například LM7171. U frekvence vyšší jak 30 MHz klesalo jeho napětí pod 1 V_{pp}, které je nutné pro fungování SDR DR2B.

Dále by stálo za úvahu vyzkoušet jiný generátor frekvence. AD9850 nabízí poměrně nízký frekvenční rozsah 0–40 MHz pro potřeby SDR DR2B. Ten je totiž navrhnutý tak, že frekvence lokálního oscilátoru je nejprve vydělena čtyřmi, což znamená, že dokážeme naladit maximálně kolem 10 MHz. Jako alternativa k AD9850 se hodí AD9851, který posunuje horní hranici frekvenčního rozsahu na 70 MHz (krystalový oscilátor pracuje na frekvenci 180 Mhz). Jako další alternativa by bylo použít mezi radioamatéry oblíbený frekvenční generátor Si570 od společnosti Silicon Labs, který umožňuje generování signálu v závislosti na konfiguraci až do 1,4 GHz, což daleko převyšuje schopnosti SDR DR2B.

10 Zdroje

- [1] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett. Software Defined Radio using MATLAB & Simulink and the RTL-SDR. [Online] <http://www.desktopsdr.com>.
- [2] Report ITU-R SM.2152, Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS). [Online] 2009. https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-SM.2152-2009-PDF-E.pdf.
- [3] Kuisma, Mikael Q. I/Q Data for Dummies. [Online] <http://whiteboard.ping.se/SDR/IQ>.
- [4] What is I/Q Data? [Online] <http://www.ni.com/tutorial/4805/en/>.
- [5] rtl-sdr.org wiki. [Online] <https://rtlsdr.org>.
- [6] Osmocom RTL-SDR. [Online] <http://osmocom.org/projects/sdr/wiki/rtl-sdr>.
- [7] RTL-SDR.com Quick Start Guide. [Online] <https://www.rtl-sdr.com/rtl-sdr-quick-start-guide>.
- [8] YU1LM/QRP site. [Online] <http://yu1lm.qrpradio.com>.
- [9] HF SDR S/H Sample and Hold Receiver DR2B. [Online] <http://yu1lm.qrpradio.com/DR2B%20HF%20SDR%20RECEIVER-YU1LM.pdf>.
- [10] OK1CJB - schéma DR2B by YU1LM. [Online] http://www.ok1cjb.cz/downloads/DR2B_sch.pdf.
- [11] OK1CJB - Úvod do SDR. [Online] <http://radio.ok1cjb.cz/index.php/za%C5%99%C3%ADzen%C3%AD/23-rx-tx/21-93650>.
- [12] Product data sheet - 74HC4053. [Online] <http://titan.physx.u-szeged.hu/~ophome/dlabor/74HCT4053.pdf>.
- [13] Tayloe, Dan. Tayloe mixer. [Online] http://www.wparc.us/presentations/SDR-2-19-2013/Tayloe_mixer_x3a.pdf.
- [14] Product data sheet - AD9850. [Online] <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9850.pdf>.
- [15] Analog Devices - Fundamentals of Direct Digital Synthesis (DDS). [Online] <http://www.analog.com/media/en/training-seminars/tutorials/MT-085.pdf>.
- [16] FTDI UM245R datasheet. [Online] http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM245R.pdf.
- [17] Ellingson, Roger M. Test_FTD2XX_NET_BitBang. [Online] <https://www.mathworks.com/matlabcentral/fileexchange/28875-test-ftd2xx-net-bitbang>.
- [18] Matlab, popis. [Online] <https://www.mathworks.com/products.html>.
- [19] RTL-SDR Support from Communications System Toolbox. [Online] <https://www.mathworks.com/hardware-support/rtl-sdr.html>.
- [20] Matlab - Process streaming signals and large data with System objects. [Online] <https://www.mathworks.com/discovery/stream-processing.html>.
- [21] *Real-Time Audio in MATLAB*. [Online] <https://www.mathworks.com/help/audio/gs/real-time-audio-in-matlab.html>.

- [22] Create Apps with Graphical User Interfaces in MATLAB. [Online]
<https://www.mathworks.com/discovery/matlab-gui.html>.
- [23] Differences Between App Designer and GUIDE. [Online]
https://www.mathworks.com/help/matlab/creating_guis/differences-between-app-designer-and-guide.html.
- [24] Write Callbacks in GUIDE. [Online]
https://www.mathworks.com/help/matlab/creating_guis/write-callbacks-using-the-guide-workflow.html.
- [25] Matlab - guidata. [Online] <https://www.mathworks.com/help/matlab/ref/guidata.html>.
- [26] Initialize UI Components in GUIDE Apps. [Online]
https://www.mathworks.com/help/matlab/creating_guis/initializing-a-guide-gui.html.
- [27] OK1CJB - MiniWhip. [Online]
http://www.ok1cjb.cz/index.php?option=com_content&view=article&id=888.
- [28] Matlab, logo. [Online] <https://www.mathworks.com/company/newsletters/articles/the-mathworks-logo-is-an-eigenfunction-of-the-wave-equation.html>.

11 Terminologický slovník

Termín	Zkratka	Význam
Softwarově definované rádio	SDR	Rádio, kde výstupem z hardwaru jsou I/Q signály a veškeré další zpracování řeší softwarová vrstva.
RTL-SDR	RTL-SDR	Levné SDR z DVB-T donglu. Název odvozen z digitálního demodulátoru Realtek RTL2832U, který obsahuje.
Hardware	HW	Fyzické vybavení počítače.
Software	SW	Programové vybavení počítače.
Operační systém	OS	Základní softwarové vybavení počítače, např. OS Windows 10.
I/Q signál	I/Q	Komplexní signál, který využívá SDR během zpracování. I = In-phase, Q = Quadrature
Milion Samples Per Second	MSPS	Vzorkovací kmitočet – milionů vzorků za sekundu.
Digital Video Broadcasting – Terrestrial	DVB-T	Pozemní digitální televizní vysílání.
Automatic Gain Control	AGC	Obvod, regulující amplitudu signálu na jeho výstupu.
Fast Fourier Transform	FFT	Rychlá Fourierova transformace – algoritmus pro převod signálu z časové domény do frekvenčního spektra.

12 Přílohy na CD/DVD

S vedoucím diplomové práce jsme se dohodli, aby zdrojový kód hotové aplikace nebyl veřejně dostupný, jak na internetu, tak v příloze na disku. I přes tuto skutečnost je možné podobnou aplikaci vytvořit s dostupnými informacemi z této práce, případně jiných zdrojů.

Příložený disk obsahuje následující soubory:

- Text diplomové práce ve formátu PDF.
- Skript pro zápis frekvence do registru AD9850 pomocí UM245 – je nutné stáhnout NET Wrapper ze stránek FTDI a vložit do stejné složky.