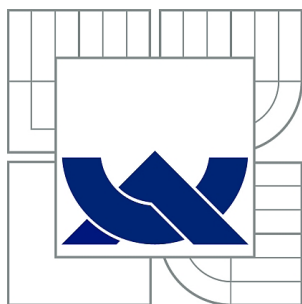
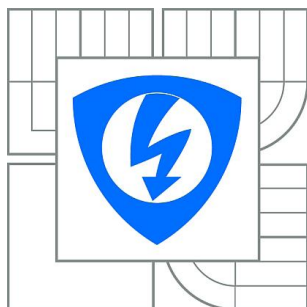




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ZDROJ MODULAČNÍHO SIGNÁLU PRO MIKROVLNNÝ MAJÁK

MODULATION SIGNAL SOURCE FOR MICROWAVE BEACON

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL BELLOŇ

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Ing. MIROSLAV KASAL, CSc.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Michal Belloň

ID: 119355

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Zdroj modulačního signálu pro mikrovlnný maják

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s modulacemi WSJT pro komunikaci s velmi slabými signály. Proved'te studii jak by bylo možné generovat dva modulační signály v kvadratuře pro SSB modulaci v módu JT4x.

Navrhněte obvody modulátoru, včetně návrhu plošného spoje a modulátor realizujte.

MMSE: Napište kód (program) pro modulátor včetně komunikace po sériové sběrnici a parametry modulátoru ověřte.

DOPORUČENÁ LITERATURA:

[1] Taylor, J. WSJT [online][cit. 14. 12. 2011]. Dostupné na <http://physics.princeton.edu/pulsar/K1JT/wsjt.html>.

[2] PINKER, J. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004.

Termín zadání: 11.2.2013

Termín odevzdání: 24.5.2013

Vedoucí práce: prof. Ing. Miroslav Kasal, CSc.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Česky:

Tato práce je zaměřena na konstrukci zdroje modulačního signálu pro mikrovlnný maják. Výstupní signál tohoto zdroje je shodný se signálem, který generuje PC software WSJT (Weak Signal by Joe Taylor) na výstupu zvukové karty. Signál na výstupu zařízení je modulován pomocí WSJT modulace a jejího konkrétního JT4 módu. Zařízení je konstruováno tak, že má na svém výstupu složku I a Q tohoto signálu.

English:

This project is focused on construction of source of modulation signal for microwave beacon. The output signal is the same as signal, which is generated by PC software WSJT (Weak Signal by Joe Taylor) on PC sound card output. Signal on device output is modulated using by WSJT modulation and its specific JT4 mode. The device output has I and Q components of signal.

Klíčová slova

Česky:

WSJT, modulace, mód JT4, velmi slabé signály, maják

English:

WSJT, modulation, JT4 mode, very weak signal, beacon

Bibliografická citace

BELLOŇ, M. *Zdroj modulačního signálu pro mikrovlnný maják*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 34 s., 4 příl., Vedoucí diplomové práce prof. Ing. Miroslav Kasal, CSc..

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Zdroj modulačního signálu pro mikrovlnný maják jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

V Brně dne

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Prof. Ing. Miroslavu Kasalovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....
podpis autora

1. Obsah

1.	Obsah.....	5
2.	Úvod	1
3.	Stručný popis použitých technik.....	1
3.1.	Frekvenční modulace	1
3.1.1.	Modulace FSK, M-FSK.....	1
3.1.	WSJT modulace a módy JT2 a JT4	2
3.2.	Modulace SSB	3
3.3.	Možnosti generace signálu.....	4
3.4.	Přímá digitální syntéza kmitočtu	5
3.4.1.	Stručný popis generace signálu	5
3.4.2.	Výstupní spektrum generátoru DDS.....	6
3.5.	Popis kódování JT4	7
3.5.1.	Zdrojové kódování.....	7
3.5.2.	Konvoluční kódování	8
3.5.3.	Prokládání.....	9
3.5.4.	Připojení synchronizační sekvence.....	9
3.5.5.	Modulace.....	9
4.	Návrh vlastního řešení.....	11
4.1.	Blokové schéma zařízení	11
4.2.	Výběr vhodných součástek.....	12
4.2.1.	Mikroprocesory	12
4.2.2.	Operační zesilovače	12
4.2.3.	D/A převodníky.....	13
4.2.4.	Stabilizátory napětí.....	13
4.3.	Hardware zařízení – jednotka DDS	14
4.3.1.	Napájení.....	14
4.3.2.	Mikroprocesor	14
4.3.3.	D/A převodník	15
4.3.4.	Dolní propust.....	15
4.3.5.	Převodník z UART na USB	17
5.	Software zařízení	18
5.1.	Struktura a stručný popis programu	18
5.2.	Popis jednotlivých funkcí programu.....	19

5.2.1. Nastavení registrů procesoru	19
5.2.2. Komunikace přes rozhraní UART	19
5.2.3. Funkce fill_freqs	20
5.2.4. Funkce unpack_message	21
5.2.5. Funkce set_freq_toDDS	22
5.2.6. Funkce calculate_transmission_gap_time	22
5.2.7. Blok DDS	23
5.2.8. Blok přepínání tónů	24
5.3. Popis programu DDS komunikátor	25
5.3.1. Zdrojový kód programu	26
5.3.2. Stručný popis aplikace GENJT4	26
5.4. Popis aplikace WSJT	27
6. Parametry a fotografie zařízení	28
6.1. Základní parametry zařízení	28
6.2. Výstupní průběhy signálů	29
6.3. Výstupní spektrum a test zařízení	30
6.4. Fotografie zařízení	31
7. Závěr	32
8. Použitá literatura	33
Přílohy	34

Seznam obrázků

Obr. 1 - Ilustrační foto	1
Obr. 2 – SSB modulátor, převzato ze [7]	3
Obr. 3 – Wienův oscilátor, převzato z [12]	4
Obr. 4 – Blokové schéma přímé číslicové syntézy, převzato ze [4]	5
Obr. 5 – Příklad výstupního spektra generátoru DDS, převzato z [2]	6
Obr. 6 - Blokové schéma zařízení	11
Obr. 7 - Schéma zapojení napájecího obvodu	14
Obr. 8 - Ilustrační fotografie	14
Obr. 9 - Schéma zapojení D/A převodníku	15
Obr. 10 - Schéma zapojení dolní propusti	15
Obr. 11 - Amplitudová charakteristika dolní propusti	16
Obr. 12 - Fázová charakteristika dolní propusti	16
Obr. 13 - Schéma zapojení převodníku UART - USB	17

Obr. 14 - Vývojový diagram programu pro procesor	18
Obr. 15 – Blokové schéma DDS, převzato z [4]	23
Obr. 16 – DDS komunikátor.....	25
Obr. 17 – Aplikace GENJT4	26
Obr. 18 – Program WSJT.....	27
Obr. 19 - Výstupní průběh signálu.....	29
Obr. 20 - Fázový posuv výstupních signálů (90°).....	29
Obr. 21 - Spektrum výstupního signálu	30
Obr. 22 – Ověření funkce zařízení	30
Obr. 23 - Fotografie modulátoru	31
Obr. 24 - Fotografie převodníku.....	31

Seznam tabulek

Tab. 1 - Některé parametry jednotlivých módů modulace WSJT.....	2
Tab. 2 - Převod jednotlivých znaků na číselné hodnoty	7
Tab. 3 - Konstanty k pro odstup tónů pro jednotlivé JT4 módy	10
Tab. 4 - Vypočtené kmitočty tónů	10
Tab. 5 - Přehled některých prodáváných mikroprocesorů.....	12
Tab. 6 - Přehled některých prodáváných operačních zesilovačů	12
Tab. 7 - Přehled některých prodáváných D/A převodníků	13
Tab. 8 - Přehled některých prodáváných stabilizátorů.....	13

2. Úvod

V radiotechnice a u radioamatérů je všeobecně požadováno domluvit se na co největší vzdálenost s co nejmenším vysílacím výkonem, a také pokud možno, co nejjednodušším způsobem. Pro komunikaci na velké vzdálenosti (tisíce kilometrů) se v současné době používají různé druhy WSJT (weak signal communication, by K1JT) modulace. Modulace WSJT využívá audiopásmo, které zabírá běžná řeč a dokáže ho zpracovat většina běžných transceiverů. NF signál se po demodulaci většinou zpracovává na PC pomocí zvukové karty a softwaru WSJT.



Obr. 1 - Ilustrační foto

Tato práce je zaměřena na vytvoření zařízení, které by nahradilo vysílací část programu WSJT bez použití PC.

3. Stručný popis použitých technik

3.1. Frekvenční modulace

Frekvenční modulace pracuje na principu, kdy kmitočet výstupního signálu závisí na amplitudě modulačního signálu. Také rychlost změny kmitočtu odpovídá rychlosti změny amplitudy a fáze modul. signálu. Informaci tedy nese kmitočet signálu. Rozdíl mezi kmitočtem bez modulace a s modulací s maximální amplitudou se nazývá frekvenční zdvih, který se značí Δf . U FM modulace se dále udává tzv. modulační index, což je poměr mezi změnou kmitočtu výstupního signálu a maximálním kmitočtem modulačního signálu.

$$m_{FM} = \frac{\Delta f_{max}}{f_{modmax}} \quad (1)$$

Protože na vyšších modulačních kmitočtech se snižuje poměr signál/šum, při modulaci jsou zesíleny vyšší kmitočty, což se nazývá preemfáze a při demodulaci jsou vyšší kmitočty potlačeny (deemfáze).

3.1.1. Modulace FSK, M-FSK

FSK (Frequency Shift Keying) modulace patří mezi digitální modulace. Může mít dva stavy (dva různé kmitočty), které odpovídají vstupnímu signálu (1,0). Je jednoduchá na modulaci i na demodulaci. Pro určení jestli je přijímán jeden nebo druhý kmitočet, je potřeba poměrně malý poměr signál/šum.

3.1. WSJT modulace a módy JT2 a JT4

Módy JT4 a JT2 modulace WSJT jsou navrženy pro použití u extrémně slabých signálů. Úroveň těchto signálů bývá pod hladinou okolního šumu. Výstupem WSJT modulace je nízkofrekvenční signál v audiopásmu. JT4 a JT2 používají dopřednou ochranu proti chybám s konvolučním kódováním a prokládáním.

Mód JT4 používá 4-FSK modulaci a každý datový bit obsahuje zároveň i synchronizační bit. Symbolová rychlost je 4,375 baudů/s. Odstup jednotlivých tónů se dá nastavit.

Mód JT2 využívá 2-FSK modulaci pro synchronizaci a diferenciální BPSK modulace pro přenos dat. Synchronizační bity mohou být opět vysílány spolu s datovými bity. Symbolová rychlost je také 4,375 baudů/s.

WSJT modulace a její módy JT2 a JT4 jsou určeny k následné SSB modulaci na nosný kmitočet, který bývá např. 144MHz, 2,3GHz, 10GHz.

WSJT modulace se využívá při komunikaci na velkou vzdálenost. Často jí bývá modulován signál pro majáky.

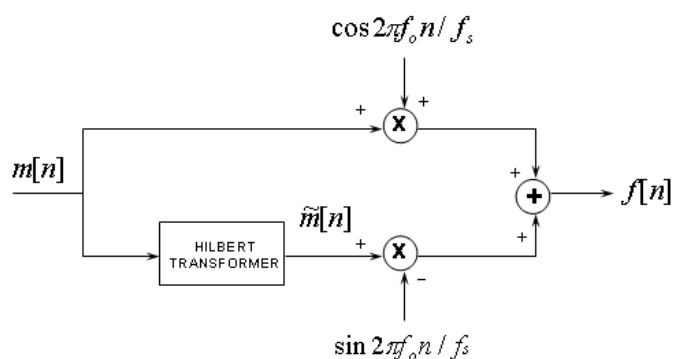
Tab. 1 - Některé parametry jednotlivých módů modulace WSJT

Mód	Rozestupy tónů [Hz]	Šířka pásma [Hz]	Minimální S/N [dB]
JT2	4,375	8,75	-24
JT4A	4,375	17,5	-23
JT4B	8,75	35	-22
JT4C	17,5	70	-21
JT4D	39,375	157,5	-20
JT4E	78,75	315	-19
JT4F	157,5	630	-18
JT4G	315	1260	-17

3.2. Modulace SSB

Tato modulace se používá hlavně v krátkovlnných pásmech, kde se používá především amplitudová modulace. U SSB modulace se jedná o provoz na jednom postranním pásmu. Tato modulace byla objevena v poválečných letech. Důvodem proč byla tato modulace vymyšlena, byla menší šířka pásma oproti standardní amplitudové modulaci a dále neefektivní využití výkonu vysílače.

Single-Sideband Modulation



Obr. 2 – SSB modulátor, převzato ze [7]

Při klasické amplitudové modulaci se přenáší nosná a dvě postranní pásma. Přitom asi 66% výkonu se přenáší v nosné vlně a cca 17% ve zbylých dvou postranních pásmech. Příjímáči však stačí přenášet jen jedno postranní pásmo aby byla celá přenášená informace správně dekodována. Čili bude-li mít vysílač například výkon 100W, pro přenos informace bude stačit výkon pouze 17W. Modulace SSB se používá i u spojení, kde je použita vnitřní modulace signálu WSJT. Jsou to většinou spojení na velmi velké vzdálenosti (řádově tisíce kilometrů).

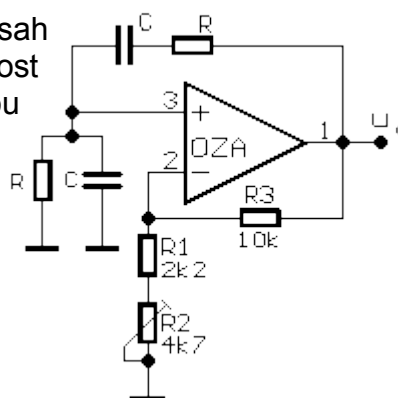
Důvod popisu SSB modulace v této práci je následující. SSB modulátor potřebuje ke správné modulaci mít na jednom vstupu původní signál a na druhém vstupu původní signál, ale s fází posunutou o 90°. Problém je, že pro posunutí fáze signálu nemůže být použit například klasický RC filtr, protože fázi signálu je potřeba posunout na všech kmitočtech o stejný úhel. Protože zařízení navrhované v této práci je generátor WSJT signálu, který bude později připojen do SSB modulátoru, bude tento generátor obsahovat oba dva výstupy signálu (s posunutím 0° a s posunutím 90°).

3.3. Možnosti generace signálu

Pro generaci signálu JT4 existuje několik možností. Buď můžeme signál generovat analogově nebo digitálně. Při analogové generaci můžeme použít například laditelný RC generátor nebo u digitálního způsobu modulace můžeme použít přímou číslicovou syntézu nebo smyčku fázového závěsu (PLL).

RC generátor

Výhodou RC generátoru je vysoký rozsah přeladění, jednoduchá konstrukce a dobrá dostupnost integrovaných obvodů pro RC oscilátory. Nevýhodou tohoto řešení je malá přesnost výstupního kmitočtu oscilátoru a složitost naladění kmitočtu pomocí mikroprocesoru. Pro generaci signálu JT4 je toto řešení prakticky nepoužitelné.



Obr. 3 – Wienův oscilátor, převzato z [12]

Smyčka fázového závěsu:

Mezi vlastnosti fázového závěsu patří přesnost kmitočtu, vysoká stabilita kmitočtu a velký rozsah přeladění. Mezi nevýhody tohoto typu oscilátoru patří poměrně složitý návrh a velký počet obvodových prvků. Dále mezi nevýhody tohoto typu oscilátoru patří, že má dlouhou dobu přeladění. Pro generaci signálu JT4 by toto řešení také nebylo nejvhodnější.

Přímá číslicová syntéza:

Dovoluje velký rozsah přeladění výstupního kmitočtu, dále jednoduché nastavení kmitočtu pomocí mikroprocesoru, malý kmitočtový krok a má také velkou rychlost přeladění. Přímá číslicová syntéza může být realizována buď pomocí speciálního obvodu nebo pomocí mikrokontroléru. Tento typ generátoru potřebuje na výstupu rekonstrukční filtr.

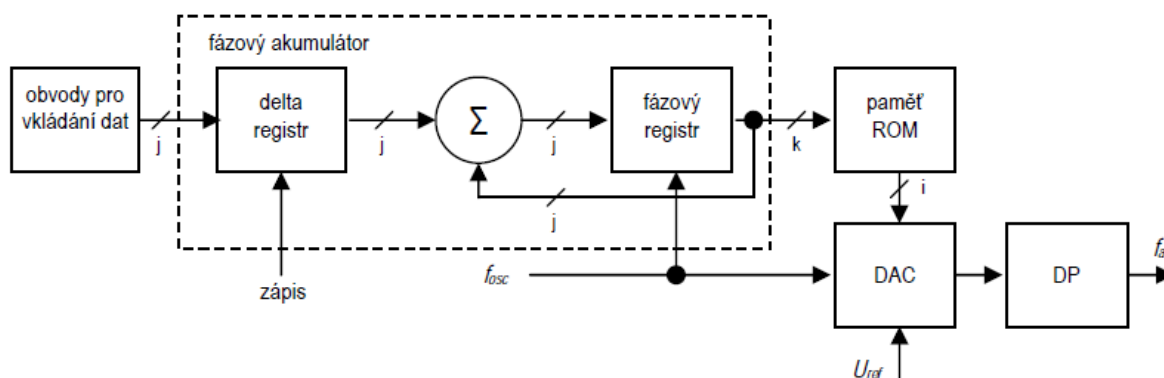
Přímá číslicová syntéza byla vybrána jako nejvhodnější řešení pro generaci signálu JT4. Jako obvod pro generaci signálu byl zvolen mikrokontrolér a to z důvodu snazšího naprogramování a větší dostupnosti.

3.4. Přímá digitální syntéza kmitočtu

3.4.1. Stručný popis generace signálu

U přímé digitální syntézy (DDS – Direct digital synthesis) se kmitočet generuje posíláním vzorků jedné periody sinusového průběhu uložených v paměti ROM na výstupní D/A převodník, za kterým následuje dolní propust. Výstupní kmitočet generátoru je určen krokem, s jakým se čtou jednotlivé vzorky z paměti.

K adresování jednotlivých vzorků v paměti slouží tzv. fázový registr, který nese informaci o aktuální fázi generovaného signálu. Na velikosti tohoto registru závisí minimální krok výstupního kmitočtu. K nastavení kmitočtu slouží tzv. delta registr, jehož hodnota určuje krok při čtení vzorků sinusového průběhu z paměti. Tyto registry dohromady tvoří fázový akumulátor.



Obr. 4 – Blokové schéma přímé číslicové syntézy, převzato ze [4]

Hodnota fázového registru se mění s každým hodinovým impulsem. Sumační člen ve fázovém akumulátoru slouží k tomu, aby si registr pamatoval svou předchozí hodnotu a k ní se potom přičetla hodnota delta registru. Jestliže hodnota fázového registru dosáhne svého maxima, začne se počítat od nuly. Jak bylo výše popsáno, šířka fázového registru ovlivňuje minimální kmitočtový krok. Čím je jeho šířka větší, tím je možné dosáhnout menší krok.

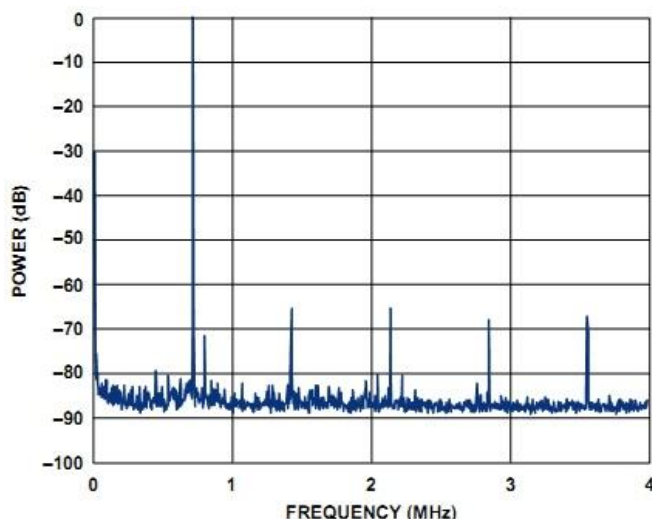
Výstupní kmitočet DDS se vypočítá takto:
$$f_{vyst} = \frac{D \cdot f_{osc}}{2^n} \quad (2) ,$$

kde: D - hodnota delta registru,
 n - bitová šířka fázového registru
 f_{osc} - vzorkovací kmitočet signálu.

Aby vzorky sinusového průběhu nezabíraly v paměti příliš místa, využívá se jeho symetrie a do paměti se ukládá jen část jeho průběhu. Celý signál se potom získá dopočítáním zbývajících vzorků. Dále se nepoužívá příliš velká bitová hloubka vzorků signálu, aby nezabíraly mnoho paměti a jejich načítání bylo rychlejší. Nejčastěji se používá bitová hloubka vzorků 8 až 16 bitů.

3.4.2. Výstupní spektrum generátoru DDS

Výstupní spektrum syntezátoru by mělo v ideálním případě obsahovat pouze jednu spektrální složku. Protože je sinusový signál složen ze vzorků, spektrum obsahuje také zrcadlové složky tohoto signálu a rovnoměrně rozprostřený kvantizační šum.



Obr. 5 – Příklad výstupního spektra generátoru DDS, převzato z [2]

Skutečné spektrum DDS signálu obsahuje více složek oproti ideálnímu. Přídavné složky jsou různé kombinační kmitočty, které vznikají harmonickým a také intermodulačním zkreslením D/A převodníku a následného zesilovače. Jsou to např. násobky základní harmonické ($2 \cdot f_0$, $3 \cdot f_0$) nebo složky vzniklé směřováním vzorkovacího kmitočtu s požadovaným kmitočtem např. $f_{osc} - 2 \cdot f_0$, $f_{osc} - 3 \cdot f_0$.

- Výhody:
- Rychlé přeladění výstupního kmitočtu, lehce realizovatelné modulace, kde se mění frekvence nebo fáze.
 - Velký rozsah přeladění výstupního kmitočtu a velmi malý kmitočtový krok
 - Referenční oscilátor je optimalizován pro jeden kmitočet
 - Lze generovat signál s libovolným průběhem
- Nevýhody:
- Lze generovat pouze signály s nižší frekvencí než je polovina referenčního kmitočtu

3.5. Popis kódování JT4

3.5.1. Zdrojové kódování

Čistá textová zpráva se může skládat pouze z těchto znaků:

- čísla 0 – 9
- 26 znaků A – Z
- 6 dalších symbolů mezera + - . / ?

Celkem je možných 42 vstupních znaků.

Zpráva JT4 může obsahovat až 13 těchto znaků, které jsou uloženy v proměnných [char 1] až [char 13]. Například volačka a lokátor vysílací stanice můžou být popsány jako „G4JNT IO90IV“.

Zpráva je kódována a generována postupně, jednotlivé znaky jsou převedeny na hodnoty od 0 – 41 podle pořadí znaku v tabulce.

Tab. 2 - Převod jednotlivých znaků na číselné hodnoty

Znak	Hodnota	Znak	Hodnota	Znak	Hodnota	Znak	Hodnota
0	0	B	11	M	22	X	33
1	1	C	12	N	23	Y	34
2	2	D	13	O	24	Z	35
3	3	E	14	P	25	mezera	36
4	4	F	15	Q	26	+	37
5	5	G	16	R	27	-	38
6	6	H	17	S	28	.	39
7	7	I	18	T	29	/	40
8	8	J	19	U	30	?	41
9	9	K	20	V	31		
A	10	L	21	W	32		

Z jednotlivých kódů znaků se nyní vypočítají 3 dlouhé integery.

Integery jsou zformátovány takto:

$$\begin{aligned} N1 &= [\text{char } 1] & N2 &= [\text{char } 6] \\ N1 &= N1 * 42 + [\text{char } 2] & N2 &= N2 * 42 + [\text{char } 7] \\ N1 &= N1 * 42 + [\text{char } 3] & N2 &= N2 * 42 + [\text{char } 8] \\ N1 &= N1 * 42 + [\text{char } 4] & N2 &= N2 * 42 + [\text{char } 9] \\ N1 &= N1 * 42 + [\text{char } 5] & N2 &= N2 * 42 + [\text{char } 10] \\ \\ N3 &= [\text{char } 11] \\ N3 &= N3 * 42 + [\text{char } 12] \\ N3 &= N3 * 42 + [\text{char } 13] \end{aligned}$$

N1 a N2 mají maximální hodnotu $42^5 - 1 = 130691231$, což odpovídá 27-bitovému integeru.

N3 je 17-ti bitový integer, který má maximální hodnotu $42^3 = 74087$.

Celkově je tedy potřeba $27 + 27 + 17 = 71$ bitů. Jeden bit je přidán jako indikace, že zpráva je čistý text, čili celkově získáme 72 bitů zdrojových dat.

Po přidání jednoho bitu se výsledné integery upraví takto:

$$\begin{aligned} N1 &= N1 * 2 + [\text{bit } 15] \text{ z } N3 & (\text{or } N3 \setminus 32768 \text{ MOD } 2) & (28 \text{ bits total}) \\ N2 &= N2 * 2 + [\text{bit } 16] \text{ z } N3 & (\text{or } N3 \setminus 65536 \text{ MOD } 2) & (28 \text{ bits total}) \\ N3 &= N3 \text{ MOD } 32768 + 32768 & & (16 \text{ bits total}) \end{aligned}$$

72-bitová sekvence se dá vytvořit jako: $N1 * 2^{16+28} + N2 * 2^{16} + N3$

Na konec této 72-bitové sekvence se připojí 31 nulových bitů.

3.5.2. Konvoluční kódování

Sekvence 103 bitů dat se nyní zakóduje pomocí FEC s poměrem $\frac{1}{2}$. Délka konvolučního kodéru je omezena na 32 bitů. Vstupní data (se zahrnutými nulami) jsou postupně čtena od MSB bitu. Bity jsou zapisovány souběžně na pravou stranu nebo na nejméně významnou pozici dvou 32-bitových posuvných registrů (reg. 0) a (reg. 1). Každý posuvný registr naplňuje Exclusive-OR paritní generátor ze zpětnovazebního výstupu, hodnoty generátorů jsou 32-bitové s hodnotami 0xF2D05351 a 0xE4613C47. Generace parity začíná v momentě, kdy se objeví první bit v registru (který musí být na začátku vyprázdněn) a pokračuje dokud registr nebude zcela zaplněn. Při každém posunutí jednoho ze 103 bitů datové sekvence se generuje jeden paritní bit z každého generátoru. Celkem se vygeneruje 206 bitů. Každému vstupnímu bitu odpovídají 2 paritní bity na výstupu.

Tab. 3 - Konstanty k pro odstup tónů pro jednotlivé JT4 módy

Mód JT4x	k
JT4a	1
JT4b	2
JT4c	4
JT4d	9
JT4e	18
JT4f	36
JT4g	72

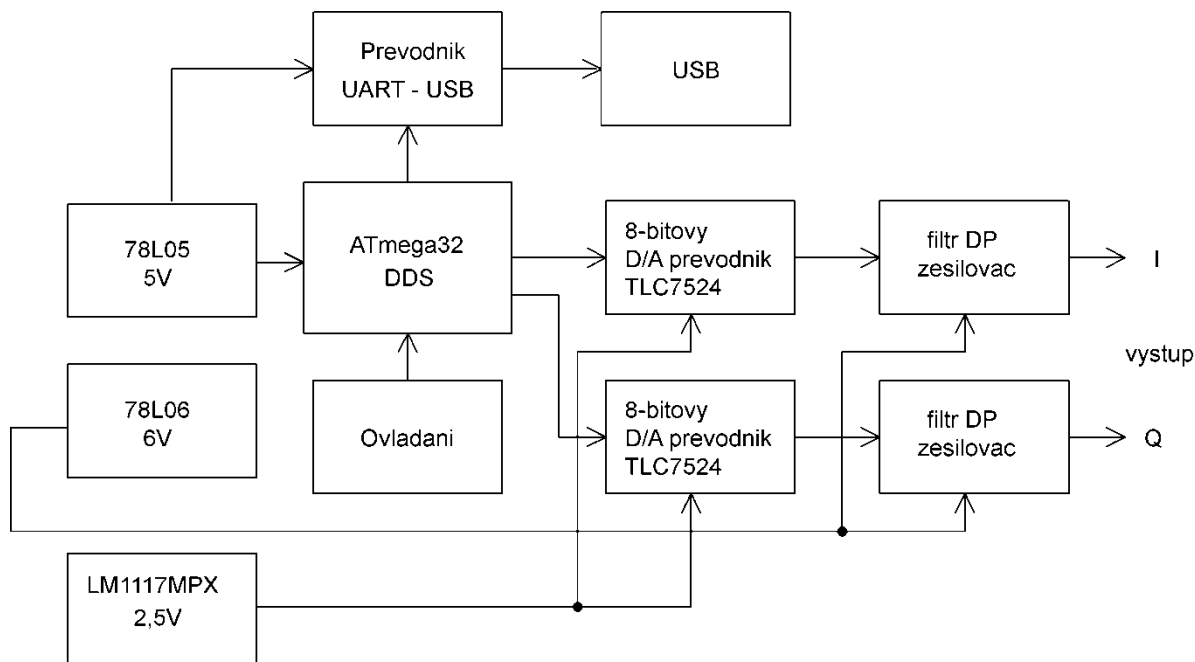
Jednotlivé tóny na sebe navazují s nepřerušenou fází, aby se nezvyšovala šířka pásma výstupního signálu.

Tab. 4 - Vypočtené kmitočty tónů

Mód	Tón 0[Hz]	Tón 1[Hz]	Tón 2[Hz]	Tón 3[Hz]	Tón 1,5[Hz]	k
JT4A	1263,89	1268,26	1272,63	1277	1270,46	1
JT4B	1257,33	1266,08	1274,83	1283,58	1270,46	2
JT4C	1244,2	1261,7	1279,2	1296,7	1270,46	4
JT4D	1211,39	1250,79	1290,19	1329,59	1270,46	9
JT4E	1152,33	1231,08	1309,03	1388,58	1270,46	18
JT4F	1034,2	1191,7	1349,2	1506,7	1270,46	36
JT4G	797,95	1112,95	1427,95	1742,95	1270,46	72

4. Návrh vlastního řešení

4.1. Blokové schéma zařízení



Obr. 6 - Blokové schéma zařízení

Modulační signál je generován pomocí jednoho mikroprocesoru ATmega32, kde je implementována přímá číslicová syntéza kmitočtu (DDS - Direct Digital Synthesis). Za mikroprocesorem s DDS následují dva 8 - bitové D/A převodníky s obvodem TLC7524, kde jeden je pro kanál I a druhý pro kanál Q. Za nimi se nachází dolní propust pro potlačení zrcadlových kmitočtů spolu s oddělovacím stupněm, aby nebyl D/A převodník příliš zatěžován.

Na plošném spoji je vyveden port B, kde jsou připojeny informační LED (v tomto případě jedna informuje o chybě při přenosu dat z PC do modulátoru a druhá vysílání signálu). Aby nebyl procesor příliš zatěžován, odesílaná zpráva je zakódována na PC a do procesoru se přes sériovou linku posílají pouze zakódovaná data s údajem o kmitočtech výstupního signálu.

Posílání dat bude popsáno níže v softwarové části. Pro komunikaci s PC bude použit převodník z UART na USB. Tento převodník je implementován přímo na desce. Napájecí napětí pro digitální obvody je použito 5V a pro operační zesilovače je použito 6V. Napájecí napětí 2,5V je použito jako reference pro D/A převodníky.

4.2. Výběr vhodných součástek

4.2.1. Mikroprocesory

Tab. 5 - Přehled některých prodáváných mikroprocesorů

Výrobce	Atmel	Atmel	Atmel	Microchip	Microchip
Typ	AVR ATMega16, (L)	AVR ATMega8, (L)	AVR ATmega32	PIC18F2220	PIC18F2455
Paměť	Flash: 16kB SRAM: 1kB EEPROM: 512B	Flash: 8kB SRAM: 1kB EEPROM: 512B	Flash:32kB SRAM:2kB EEPROM:1kB	Flash: 4kB SRAM: 512B EEPROM: 256B	Flash: 24kB SRAM: 2kB EEPROM: 256B
Funkce	2 x 8bit Timer 1x 16bit Timer 4x PWM, 8x 10bit A/D	2x8bit,1x16bit Timer, 8x A/D, 3xPWM,	2 x 8-bit, 16bit Timer 8 x 10bit A/D	3x 16bit Timer, 10x A/D, 2x PWM,	3x 16Bit Timer, 10x 10bit A/D, 2x PWM
Kom. rozhraní	USART,SPI,JTAG I2C, ISP	USART,SPI,JTAG, I2C,ISP	USART,SPI,JTAG I2C,ISP	USART,SPI, I2C	EUSART,SPI, I2C,USB,,
Počet 8bit I/O portů	4	3	4	4	3
Kmitočet	max 16MHz 8MHz (L)	max 16MHz 8MHz (L)	max 16MHz	max 25MHz	max 48MHz
Napájecí napětí	4,5 - 5,5V 2,7 - 5,5V(L)	4,5 - 5,5V 2,7 - 5,5V(L)	2,7 – 5,5V	2 - 5,5V	2 - 5,5V
cena [Kč]	cca 100	cca 70	cca 120	cca 200	cca 200

4.2.2. Operační zesilovače

Tab. 6 - Přehled některých prodáváných operačních zesilovačů

Typ	MC4558	TS272	TLC272	TL082	TL062D
Napájecí napětí	(±3) – (±22)V	3 – 16V	3 – 16V	(±5) –(±18)V	6 – 36V
Vstupní napětí	max ±15V	-0,3 – 18V	-0,3 – 18V	max ±15V	max ±15V
Vstupní odpor	cca 2MΩ	cca GΩ	cca 10 ¹² Ω	cca 10 ¹² Ω	cca 10 ¹² Ω
Slew rate	2,2V/μs	5,5V/μs	3V/μs	13V/μs	3,5V/μs
cena [Kč]	3		11	7	5

4.2.3. D/A převodníky

Tab. 7 - Přehled některých prodáváných D/A převodníků

Typ	DAC0808	AD7224	AD5330	TLC7524
Napájecí napětí	4,5 – 18V	11,4 – 16,5V	2,5 – 5,5V	5 – 16,5V
Počet bitů	8	8	8	8
Doba odezvy	150ns	90ns	8μs	100ns
Cena	18Kč	189Kč	83Kč	24Kč

4.2.4. Stabilizátory napětí

Tab. 8 - Přehled některých prodáváných stabilizátorů

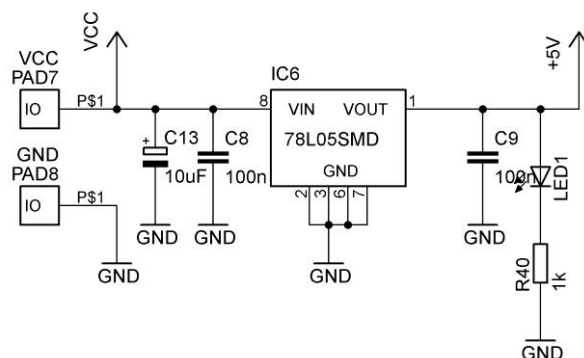
Typ	78L05SMD	78L06SMD	LM1117MPX
Max. vstupní napětí	30V	30V	20V
Výstupní napětí	5V	6V	2,5V
max. výstupní proud	100mA	100mA	800mA
cena	2Kč	2Kč	12Kč

4.3. Hardware zařízení – jednotka DDS

4.3.1. Napájení

V zařízení jsou použity celkem tři napájecí obvody, kde dva jsou typu 78L05 a 78L06 a třetí je typu LM1117MP.

Všechny stabilizátory využívají základní zapojení z katalogového listu. Napájecí napětí 5V je využito pro napájení digitální části, 6V je použito pro napájení operačních zesilovačů a 2,5V je použito jako referenční pro výstupní D/A převodníky.



Obr. 7 - Schéma zapojení napájecího obvodu

4.3.2. Mikroprocesor

Za mikroprocesor v jednotce DDS je použit ATmega32 od firmy Atmel. Tento procesor má 4 osmibitové porty, dále několik čítačů a časovačů, různé komunikační rozhraní a několik přerušovacích kanálů.

Podrobný popis procesoru je uveden v tabulce výše. Hlavní důvod použití tohoto mikroprocesoru byla poměrně velká flash paměť, která je zde 32kB. Využití portů procesoru je takové, že ke dvěma jeho portům jsou připojeny D/A převodníky. Jeden z portů slouží ke komunikaci s další jednotkou zařízení. Poslední port procesoru je využit k ovládání jednotky DDS.

Taktovací kmitočet procesoru je 16MHz, což je jeho maximální možná hodnota.



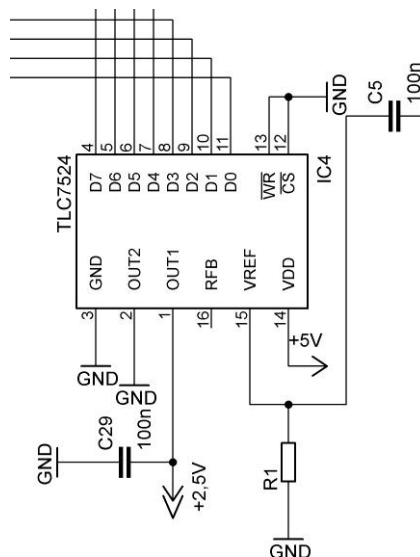
datasheet.seekic.com

Obr. 8 - Ilustrační fotografie

4.3.3. D/A převodník

V tomto zařízení je použit 8-bitový paralelní převodník typu TLC7524. Jedná se o převodník od firmy Texas instruments.

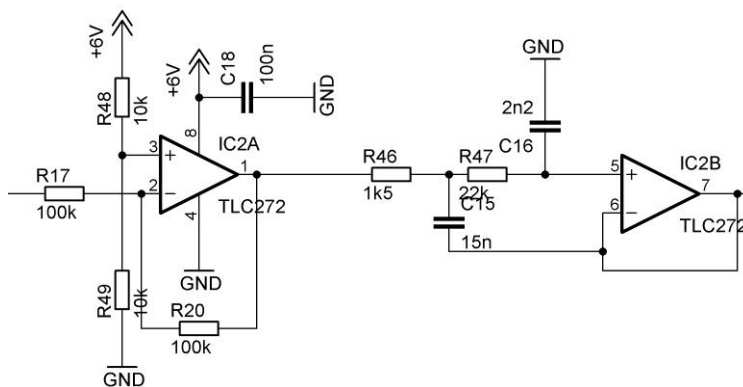
Doba odezvy tohoto převodníku je okolo 100ns, což odpovídá rychlosti vzorkování přibližně 10 megasamplů za sekundu. Napájecí napětí tohoto převodníku může být od 5 do 16,5V. Referenční napětí převodníku může být přibližně od -25V do +25V. V tomto případě je použito referenční napětí 2,5V. Převodník je zapojen v tzv. napětovém módu. Piny WR a CS slouží k ovládání D/A převodníku. Nyní jsou zapojeny tak, aby se výstupní signál měnil okamžitě při změně vstupního 8-mi bitového signálu.



Obr. 9 - Schéma zapojení D/A převodníku

4.3.4. Dolní propust

Aby byly potlačeny zrcadlové kmitočty a vyšší harmonické, je za D/A převodníkem zapojena dolní propust spolu s oddělovacím zesilovačem. Protože vzorkovací kmitočet DDS je poměrně nízký (cca 50kHz), obyčejný RC článek by nestačil, jelikož má malou strmost. Proto je použita dolní propust druhého řádu, která je typu Sallen-key. Aby nebyl



Obr. 10 - Schéma zapojení dolní propusti

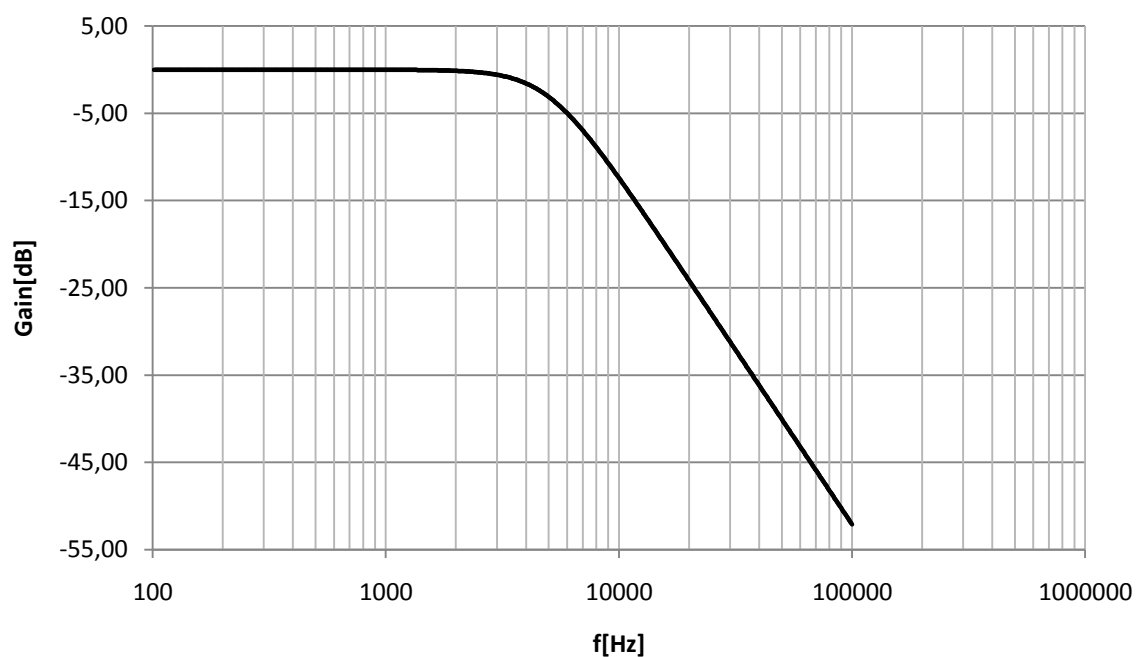
ovlivňován touto propustí D/A převodník je před ní zapojen ještě oddělovací zesilovač, který má zesílení 1.

Pro navržení hodnot součástek dolní propusti byl využit program FilterPro od firmy Texas Instruments. V tomto programu je možné navrhnout mnoho různých typů filtrů, různých řádů. Mezní kmitočet dolní propusti byl zvolen na $f_m = 5\text{kHz}$, což je dostačující, protože pomocí DDS budou generovány pouze tóny do kmitočtu cca 2kHz.

Použitím výše zmíněného programu byly vypočteny následující hodnoty součástek dolní propusti pro mezní kmitočet 5kHz:

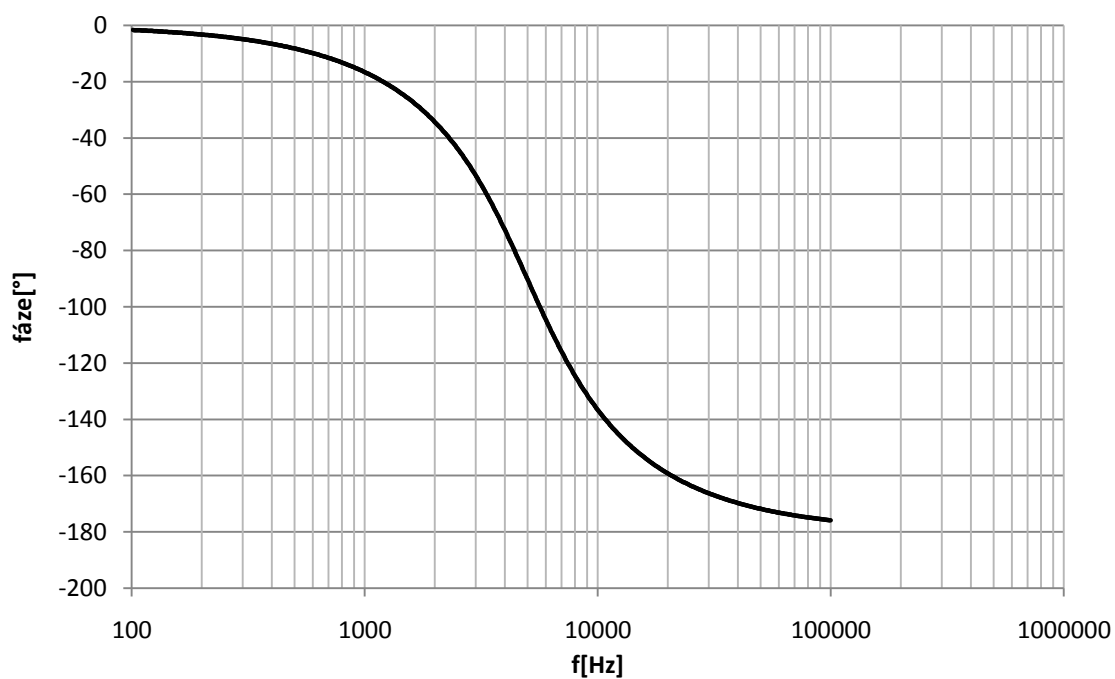
$$R46 = 1k5, R47 = 22k, C15 = 15nF, C16 = 2n2$$

Amplitudová charakteristika dolní propusti



Obr. 11 - Amplitudová charakteristika dolní propusti

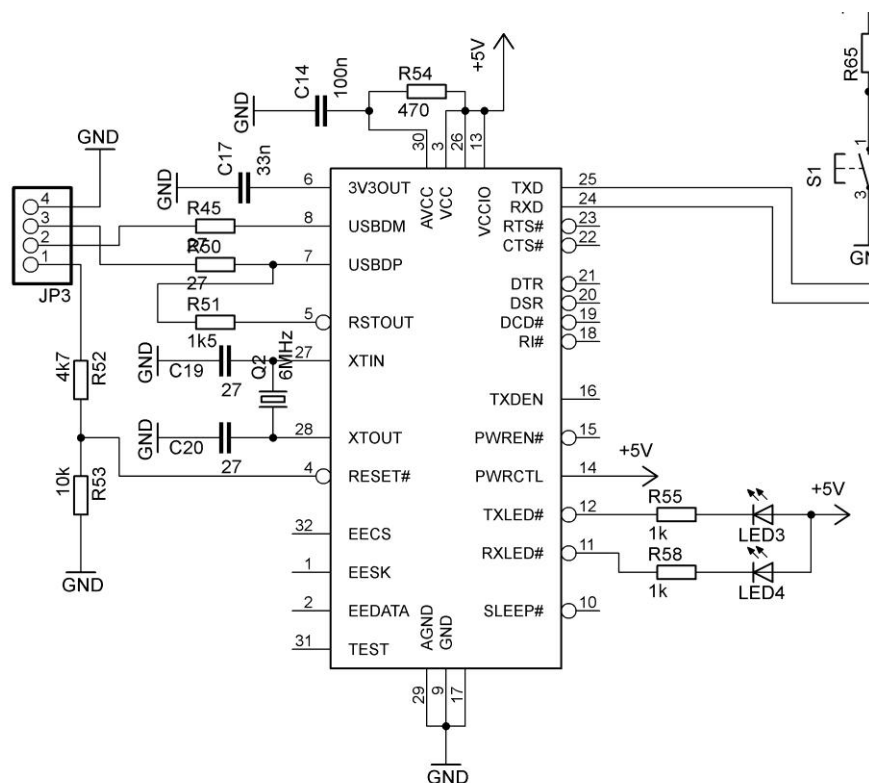
Fázová charakteristika dolní propusti



Obr. 12 - Fázová charakteristika dolní propusti

4.3.5. Převodník z UART na USB

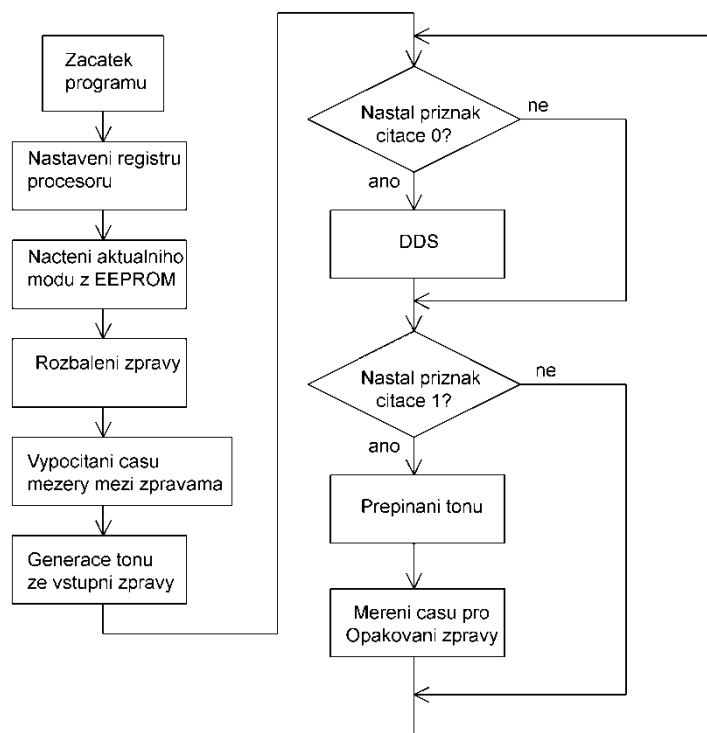
Pro komunikaci zařízení s PC je využito rozhraní UART (Universal Asynchronous Receiver/Transmitter). Většina PC dnes již však nemá rozhraní RS232 (UART), z tohoto důvodu je použit převodník z UART na USB. Zapojení využívá známý obvod FT232. Obvod obsahuje rozhraní UART, kde je možné nastavit například počet datových bitů, počet stop bitů, paritu a také rychlost. Dále obsahuje rozhraní USB, které podporuje USB1.1 a USB 2.0. Dále obvod obsahuje vyrovnávací paměti a blok pro převod těchto dvou rozhraní.



Obr. 13 - Schéma zapojení převodníku UART - USB

5. Software zařízení

5.1. Struktura a stručný popis programu



Obr. 14 - Vývojový diagram programu pro procesor

Na začátku programu se provede nejdříve nastavení registrů procesoru. Zde se nastavují například porty procesoru, čítače, přerušení apod. Dále následuje načtení aktuálního vysílacího módu z paměti EEPROM (do zvláštní proměnné se uloží frekvence tónů daného módu, které budou následně použity pro vysílání jednotlivých tónů). K načtení aktuálního módu slouží funkce **fill_freqs()**.

Po této operaci následuje rozbalení zprávy, která je uložena v EEPROM. Protože program GENJT4 na PC generuje čísla tónů ve formátu, kdy jsou v jednom bytu uloženy 4 čísla tónů (celkem je vygenerováno 52 bytů), je potřeba tyto byty rozdělit tak, aby čísla každého tónu byla uložena zvlášť v jedné proměnné typu char (čili je vytvořeno pole o velikosti $4 \times 52 = 208$ bytů). Rozbalení zprávy je potřeba z důvodu snadnějšího čtení tónů. Pro rozbalení zprávy slouží funkce **unpack_message()**.

Nyní se provede vypočítání času mezery mezi začátky vysílaných zpráv.

Pro počítání této mezery se využívá čítač pro přepínání tónů, který generuje kmitočet 4,375Hz.

K vypočítání času mezery je potřeba vstupní hodnota, která je opět uložena v EEPROM. K této operaci slouží funkce **calculate_transmission_gap_time()**.

Nyní už následuje funkce pro generaci tónu ze vstupní zprávy. Je to funkce **set_freq_to_DDS()**.

Po těchto počátečních funkcích již následuje hlavní cyklus celého programu. Blok DDS obsahuje funkci pro generování tónů. Tento blok je řízen čítačem 1. Další blok, který následuje, se nazývá Přepínání tónů. Slouží k postupnému vyčítání tónů z rozbalené zprávy. Je řízen čítačem 2. Posledním blokem je Měření času pro opakování zprávy. V tomto bloku se měří čas potřebný pro opakování zprávy. Tyto tři základní bloky běží neustále v cyklu.

5.2. Popis jednotlivých funkcí programu

5.2.1. Nastavení registrů procesoru

Jako výstupní porty pro signál z DDS jsou použity PORTA a PORTC. Dále je potřeba nastavit čítače procesoru.

Čítač 0 je využit pro časování bloku DDS. Jeho kmitočet je nastaven na 62,5kHz, což je vzorkovací kmitočet pro generování signálu.

Čítač 1 je využit pro časování bloku Přepínání tónů. Generovaný kmitočet tohoto čítače je nastaven na hodnotu 250kHz a poté je pomocí komparace jeho generovaný výstupní kmitočet snížen na přibližně 4,375Hz (4,374989063Hz). Kmitočet není zcela přesný, protože nebyl dostupný vhodnější krystal. V tomto zařízení je použit krystal s frekvencí 16MHz.

Poslední čítač 2 slouží ke komunikaci přes rozhraní UART.

Rozhraní UART je nastaveno následovně:

- 8 bitů, 1 stop bit, lichá parita
- Asynchronní mód
- Rychlost: 9600 baudů/s
- Aktivované přerušení

5.2.2. Komunikace přes rozhraní UART

Při příjmu dat přes rozhraní UART se nejdříve aktivuje přerušení, které spustí funkci pro přijetí dat. Na začátku této funkce se zapne čítač 2, který má periodu 16,384ms. Čítač 2 měří dobu přenosu přes rozhraní UART. Slouží k zastavení příjmu dat. Po aktivaci čítače se již začínají přijímat data. Správnost dat se ověřuje pomocí parity. Jsou – li data bezchybná, ukládají se postupně do přijímacího bufferu `rx_buffer[]`. Velikost tohoto bufferu je 55 bytů. Naplní-li se tento buffer, data se přestanou načítat. K ukončení načítání dat slouží čítač 2. Čítač je potřeba použít, protože při přenosu není definováno, který znak by byl použit jako ukončující. Perioda čítače je proto nastavena tak, aby její doba odpovídala době, která je potřeba pro odeslání 55 bytů při rychlosti 9600 baudů/s. Nastane – li během přenosu chyba (vlivem chybné parity nebo přetečení dat), změni se hodnota proměnné `transmission_err_flag` na 1, nedojde k zapsání chybných dat a přenos se stává celý chybným.

Při přetečení čítače 2 se aktivuje přerušení, kdy se spustí funkce, ve které se přijatá data uloží z přijímacího bufferu do paměti EEPROM.

Datový paket posílaný přes rozhraní UART:

Byte 0: Přenáší se typ módu (JT4A - 1, JT4B - 2,...), každému módu odpovídá určité číslo

Byte 1: Rezerva

Byte 2: Doba mezi vysíláním (v minutách)

Byte 3 – 55: Byty, kde se přenáší zpráva

5.2.3. Funkce fill_freqs

Funkce **fill_freqs** slouží k vypočítání hodnot delta registru pro blok DDS z požadovaných frekvencí.

Celkem se počítají 4 hodnoty (4 tóny na jeden JT4 mód).

Vzhledem k tomu, že blok DDS je 24-bitový, vypočítaná čísla jsou také 24-bitová. Poté jsou tato čísla rozdělena do tří 8-mi bitových čísel kvůli menší náročnosti na výpočet v bloku DDS.

V proměnné **freqs** jsou uloženy kmitočty jednotlivých tónů všech módů.

Kmitočty tónů uložené v proměnné freqs:

```
const flash float freqs[7][4] = {      //frekvence pro mody
    {1263.89, 1268.26, 1272.63, 1277}, // JT4A
    {1257.33, 1266.08, 1274.83, 1283.58},
    {1244.2, 1261.7, 1279.2, 1296.7},
    {1211.39, 1250.79, 1290.19, 1329.59},
    {1152.33, 1231.08, 1309.03, 1388.58},
    {1034.2, 1191.7, 1349.2, 1506.7},
    {797.95, 1112.95, 1427.95, 1742.95} //JT4G
};
```

Zdrojový kód funkce fill_freqs:

```
void fill_freqs(unsigned char mode)
{
    float fl_buff;
    long buff;
    char i;

    for(i=0;i<4;i++)
    {
        fl_buff = (16777216.0 * freqs[mode][i]) / 62500.0; // (2^24 * f_vyst) / f_osc
```

```

buff = fl_buff;

if (fl_buff - buff >= 0.5) buff++;

actual_mode_freqs_hi[i] = (buff & 0x00FF0000) >> 16;
actual_mode_freqs_mid[i] = (buff & 0x0000FF00) >> 8;
actual_mode_freqs_lo[i] = buff & 0x000000FF;
}
}

```

Základem této funkce je cyklus for. Výsledné vypočítané hodnoty delta registru se ukládají do proměnných **actual_mode_freqs_hi** až **actual_mode_freqs_lo**.

5.2.4. Funkce unpack_message

Tato funkce je určena k rozbalení zprávy, kde jsou uloženy čísla tónů, které se mají vysílat. Protože jsou čísla tónů uložena tak, že v jednom bytu jsou uloženy 4, je potřeba tato čísla roztrždit tak, aby každé z nich bylo uloženo v jednom bytu. Roztržení těchto čísel je potřeba z důvodu jejich snadnějšího čtení.

Zdrojový kód funkce unpack_message:

```

void unpack_message(void)
{
    unsigned char i;

    for(i=0;i<52;i++)
    {
        unpacked_msg[i*4] = (eep_packed_msg[i] & 0xC0) >> 6;
        unpacked_msg[i*4+1] = (eep_packed_msg[i] & 0x30) >> 4;
        unpacked_msg[i*4+2] = (eep_packed_msg[i] & 0x0C) >> 2;

        if (i == 51) break;
        else unpacked_msg[i*4+3] = eep_packed_msg[i] & 0x03;
    }
}

```

Funkce **unpack_message** vytváří z pole o velikosti 52 bytů pole o velikosti 208 bytů.

Příklad:

Vstupní hodnota je: 0x52 → 01 01 00 10
Výstupní hodnoty budou: 1, 1, 0, 2

5.2.5. Funkce set_freq_toDDS

Funkce **set_freq_toDDS** je určena ke generaci tónů z rozbalené zprávy. Jejím vstupním parametrem je pouze ukazatel na jednotlivé hodnoty v poli **unpacked_msg**.

Zdrojový kód funkce:

```
void set_freq_toDDS(unsigned char msg_pointer)    // generuje tony ze vstupni zpravy
{
    unsigned char actual_tone = unpacked_msg[msg_pointer];

    f_buff_hi = actual_mode_freqs_hi[actual_tone];
    f_buff_mid = actual_mode_freqs_mid[actual_tone];
    f_buff_lo = actual_mode_freqs_lo[actual_tone];
}
```

5.2.6. Funkce calculate_transmission_gap_time

Funkce vypočítává čas mezery mezi zprávami ze sekund do počtu cyklů, které mají kmitočet 4,375Hz. Pro měření doby mezi zprávami je využit čítač, který slouží k taktování přepínání tónů. Je použit, protože více čítačů není k dispozici.

Zdrojový kód funkce calculate_transmission_gap_time:

```
void calculate_transmission_gap_time(void)
{
    float float_buff;

    float_buff = eep_msg_pprtys[2] * 262.5 - 207;
    new_transmission_delay = float_buff;
    if ((float_buff - new_transmission_delay) > 0.5)
        new_transmission_delay++;
}
```

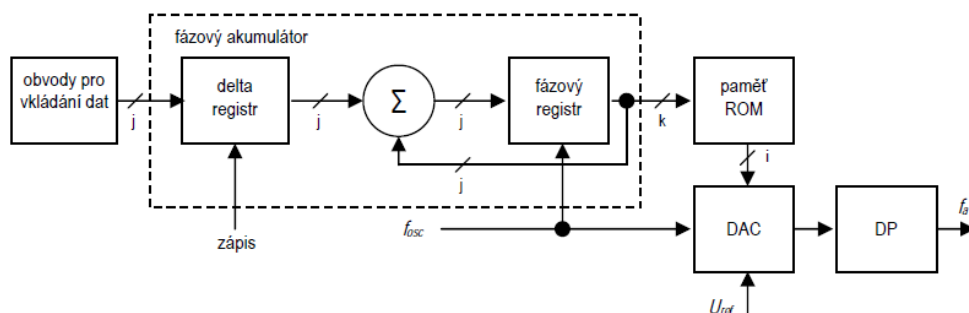
Funkce pracuje přibližně tak, že se vypočítá doba (respektive počet cyklů), která uplyne mezi koncem jedné zprávy a začátkem druhé zprávy. V paměti EEPROM je uložena pouze hodnota, která odpovídá době, po které se opakují zprávy (v minutách).

Příklad výpočtu počtu cyklů:

120s ->	$120 \cdot 4,375 = 525$ cyklů
Délka zprávy ->	207 cyklů
Rozdíl ->	$525 - 207 = 318$ cyklů

5.2.7. Blok DDS

Blok DDS je jednou z nejdůležitějších částí celého zařízení. V tomto bloku se generují výstupní tóny zařízení. Tyto tóny mají sinusový průběh. Přímá číslicová syntéza je vytvořena podle jejího blokového schématu, které je na obrázku 13.



Obr. 15 – Blokové schéma DDS, převzato z [4]

Jako zdroj vzorkovacího kmitočtu zde slouží čítač 0. Čítač generuje vzorkovací kmitočet 62500 Hz, což je v tomto případě dostatečné. Blok DDS dále obsahuje fázový akumulátor, který je kvůli menší výpočetní náročnosti napsán v assembleru. Fázový akumulátor je 24-bitový. Výstupní signál je kvadrurní, výstupem z bloku DDS je sinusový i kosinusový signál. Posunutí fáze výstupního signálu se docílilo přečtením dvou vzájemně posunutých vzorků ze sinusové lookup tabulky. Lookup tabulka je 8-bitová.

Lookup tabulka sinusovky:

```
/*const flash */unsigned char sin_vzorky[256]= {           //sinus lookup tabulka
    0x80,0x83,0x86,0x89,0x8C,0x8F,0x92,0x95,0x98,0x9B,0x9E,0xA2,
    0xA5,0xA8,0xAA,0xAD,0xB0,0xB3,0xB6,0xB9,0xBC,0xBE,0xC1,0xC4,
    0xC6,0xC9,0xCB,0xCE,0xD0,0xD3,0xD5,0xD7,0xDA,0xDC,0xDE,0xE0,
    0xE2,0xE4,0xE6,0xE8,0xEA,0xEB,0xED,0xEE,0xF0,0xF1,0xF3,0xF4,
    0xF5,0xF6,0xF8,0xF9,0xFA,0xFA,0xFB,0xFC,0xFD,0xFD,0xFE,0xFE,
    0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0xFE,0xFE,0xFD,
    0xFD,0xFC,0xFB,0xFA,0xFA,0xF9,0xF8,0xF6,0xF5,0xF4,0xF3,0xF1,
    0xF0,0xEE,0xED,0xEB,0xEA,0xE8,0xE6,0xE4,0xE2,0xE0,0xDE,0xDC,
    0xDA,0xD7,0xD5,0xD3,0xD0,0xCE,0xCB,0xC9,0xC6,0xC4,0xC1,0xBE,
    0xBC,0xB9,0xB6,0xB3,0xB0,0xAD,0xAA,0xA8,0xA5,0xA2,0x9E,0x9B,
    0x98,0x95,0x92,0x8F,0x8C,0x89,0x86,0x83,0x80,0x7C,0x79,0x76,
    0x73,0x70,0x6D,0x6A,0x67,0x64,0x61,0x5E,0x5A,0x58,0x55,0x52,
    0x4F,0x4C,0x49,0x46,0x43,0x41,0x3E,0x3B,0x39,0x36,0x34,0x31,
    0x2F,0x2C,0x2A,0x28,0x25,0x23,0x21,0x1F,0x1D,0x1B,0x19,0x17,
    0x15,0x14,0x12,0x11,0x0F,0x0E,0x0C,0x0B,0x0A,0x09,0x07,0x06,
    0x06,0x05,0x04,0x03,0x02,0x02,0x01,0x01,0x01,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x02,0x02,0x03,0x04,0x05,
    0x06,0x06,0x07,0x09,0x0A,0x0B,0x0C,0x0E,0x0F,0x11,0x12,0x14,
    0x15,0x17,0x19,0x1B,0x1D,0x1F,0x21,0x23,0x25,0x28,0x2A,0x2C,
    0x2F,0x31,0x34,0x36,0x39,0x3B,0x3E,0x41,0x43,0x46,0x49,0x4C,
    0x4F,0x52,0x55,0x58,0x5A,0x5E,0x61,0x64,0x67,0x6A,0x6D,0x70,
    0x73,0x76,0x79,0x7C};
```

Zdrojový kód bloku DDS:

```
if(TIFR & 0x01) // citac pro DDS
{
    //Rd = Rd + Rr
    #asm("ADD R7,R8"); // ACC_lo = ACC_lo + f_buff_lo;
    #asm("ADC R4,R9"); // ACC_mid = ACC_mid + f_buff_mid + carry;
    #asm("ADC R5,R6"); // ACC_hi = ACC_hi + f_buff_hi + carry;

    SIN_DA = sin_vzorky[ACC_hi];
    COS_DA = sin_vzorky[(unsigned char)(ACC_hi + 64)];
    TIFR |= 1<<0; // vymazani priznaku pretečení citace
}
```

5.2.8. Blok přepínání tónů

Tento blok zabezpečuje přepínání jednotlivých tónů, které jsou uloženy v paměti EEPROM (v paměti jsou uloženy čísla těchto tónů). Celkem se posílá 207 těchto tónů, přičemž perioda jejich přepínání je řízena čítačem 1. Frekvence přepínání tónů je již výše zmíněná 4,375Hz, což odpovídá periodě 228ms. Pro posílání tónů je využita výše popsaná funkce **set_freq_toDDS**. Během vysílání bliká informační LED. V případě kdy se zrovna nevysílá žádná zpráva, je možné generovat jeden souvislý tón. V tomto bloku se také počítá počet uběhlých cyklů, aby bylo možné změřit délku mezery mezi zprávami. Příznak, který určuje zda má být zpráva vysílána se nazývá **wait_to_transmission_flag**. Zpráva se začne posílat, jakmile dojde k napočítání potřebného počtu cyklů, jejichž počet byl vypočítán ve výše popsané funkci **calculate_transmission_gap_time**.

Zdrojový kód bloku pro přepínání tónů:

```
if(TIFR & 0x10) // prepinani tonu s kmitoctem 4,375Hz
{
    four_375_cntr++; // pocet ubehlych 4.375 casovych cyklu

    if(msg_ptr < 207) // posilani jednotlivych tonu do DDS
    {
        set_freq_toDDS(msg_ptr++);
        TX_LED = (msg_ptr & 0x01);
    }
    else if(wait_to_transmission_flag == 0) // cekani nez bude zprava znovu vysilana
    {
        f_buff_hi = 0x00; // 0 Hz - signal, pokud neni vysilana zprava
        f_buff_mid = 0x00;
        f_buff_lo = 0x00;

        TX_LED = 0;
    }
}
```

```

wait_to_transmission_flag = 1;
transmission_end_time = (four_375_cntr - 1);
}

if(((four_375_cntr - transmission_end_time) >= new_transmission_delay) && wait_to_transmission_flag) //
mereni casu pro opakovani zpravy
{
wait_to_transmission_flag = 0;
msg_ptr = 0;
}

TIFR |= 1<<4;
}

```

5.3. Popis programu DDS komunikator

Program DDS komunikátor slouží k odesílání dat z PC do modulátoru. Data se posílají přes sériovou linku UART. V aplikaci se dají nastavit různé parametry pro vysílání zprávy. Jsou to např. mód (JT4A, JT4B, ...), dále doba mezi zprávami a číslo sériového portu (COM1, COM2,...). Dále program umožňuje vložení souboru se zprávou. Tento soubor se generuje pomocí programu GENJT4.

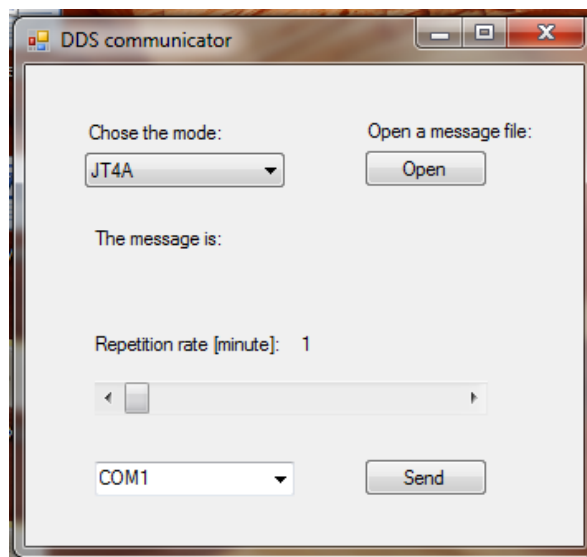
Možnosti nastavení programu:

Módy:

- JT4A
- JT4B
- JT4C
- JT4D
- JT4E
- JT4F
- JT4G

Doba opakování: 1 – 30 min

Číslo COM portu: 1 - 10



Obr. 16 – DDS komunikátor

5.3.1. Zdrojový kód programu

Aplikace je napsána v programovacím jazyku C#. Je to objektový programovací jazyk. Klíčovou vlastností aplikace DDS komunikátor je, že je nutné převést znaky se zprávou (zpráva je uložena v obecném souboru), které vygeneroval program GENJT4 z formátu ASCII do formátu HEX. Jinak by zpráva nemohla být správně přečtena.

Pro správnou funkci této aplikace je potřeba mít nainstalovaný dotNET .

Konverzní tabulka z ASCII do HEX:

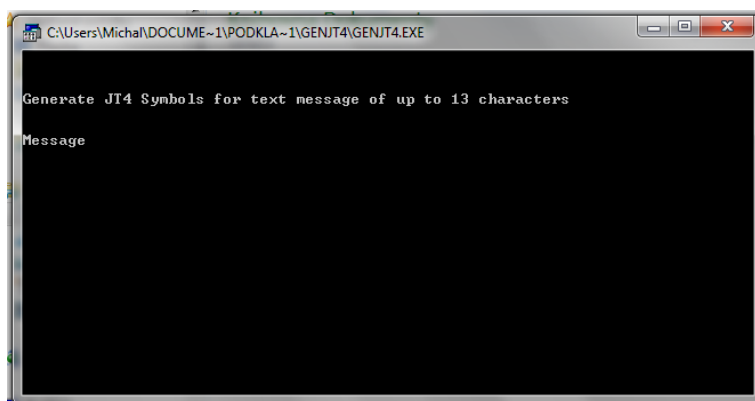
```
ASCIIconvTable['0'] = 0;
ASCIIconvTable['1'] = 1;
ASCIIconvTable['2'] = 2;
ASCIIconvTable['3'] = 3;
ASCIIconvTable['4'] = 4;
ASCIIconvTable['5'] = 5;
ASCIIconvTable['6'] = 6;
ASCIIconvTable['7'] = 7;
ASCIIconvTable['8'] = 8;
ASCIIconvTable['9'] = 9;

ASCIIconvTable['A'] = 0x0A;
ASCIIconvTable['B'] = 0x0B;
ASCIIconvTable['C'] = 0x0C;
ASCIIconvTable['D'] = 0x0D;
ASCIIconvTable['E'] = 0x0E;
ASCIIconvTable['F'] = 0x0F;
```

5.3.2. Stručný popis aplikace GENJT4

Aplikace GENJT4 má za úkol vygenerovat soubor s čísly tónů ze vstupní zprávy. Tato zpráva může mít až 13 znaků. Posloupnost výstupních tónů je generována podle protokolu JT4, který je popsán v bodě 3.4. Tato aplikace nebyla

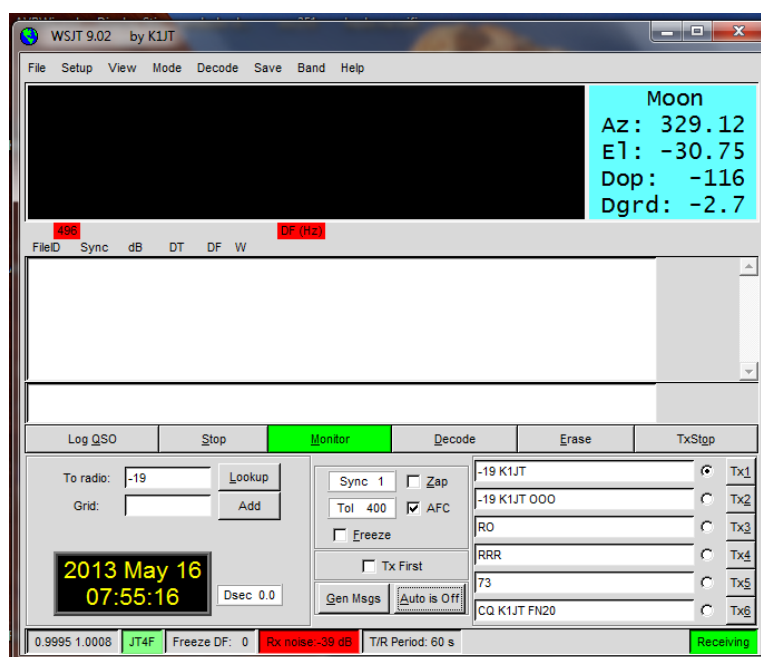
programována speciálně pro tuto práci, ale je převzatá ze stránek WSJT z důvodu obtížnosti jejího naprogramování.



Obr. 17 – Aplikace GENJT4

5.4. Popis aplikace WSJT

Program WSJT je určen pro příjem signálů WSJT pomocí zvukové karty počítače. Obsahuje mnoho funkcí a umožňuje dekódovat signály s nejrůznějšími módy. Jsou to například JT4A, JT4B až JT4G, dále pak módy JT65 a další. Tento program je také použit pro dekódování signálů ze zařízení, popsaného v této práci. Program signál analyzuje vždy po uplynutí jedné minuty. To znamená, že zprávu je potřeba poslat vždy na začátku minuty.



Obr. 18 – Program WSJT

Program WSJT dále obsahuje analyzátor spektra a indikaci úrovně vstupního signálu.

Mezi nedostatky tohoto programu patří, že nedokáže spolupracovat s každou zvukovou kartou. Další nedostatek tohoto programu je v tom, že při generaci signálu vytváří chyby; zřejmě vlivem nekompatibilní zvukové karty.

Vysvětlení některých zkratk používaných v programu:

- Sync – minimální úroveň synchronizačního prahu
- DT – rozdíl času mezi vysílačem a přijímačem
- DF – rozdíl kmitočtu mezi vysílačem a přijímačem
- Sync dB – úroveň synchronizačního signálu
- W – šířka pásma synchronizačního signálu
- * -- normální zpráva

6. Parametry a fotografie zařízení

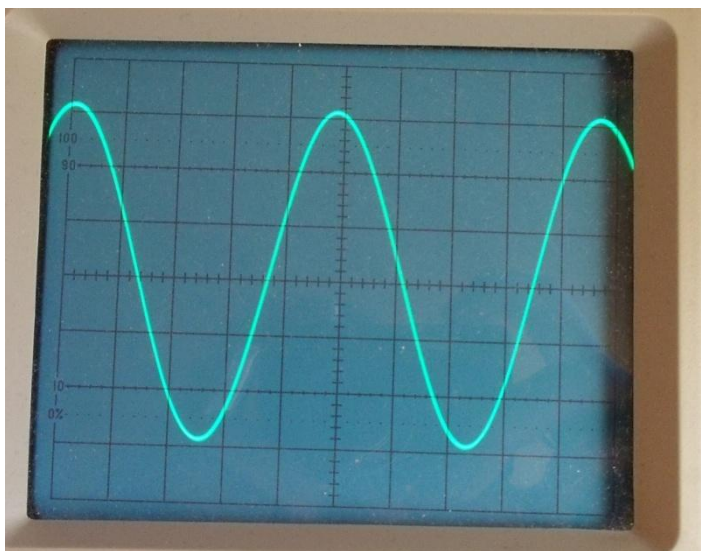
6.1. Základní parametry zařízení

V tomto odstavci se nachází výpis několika základních parametrů navrženého zařízení.

Napájecí napětí:	7 – 12V
Podporované módy:	JT4A, JT4B, JT4C, JT4D, JT4E, JT4F, JT4G
Doba mezi vysíláním zpráv:	1 – 30min
Délka vstupní zprávy:	max 13 znaků
Frekvenční rozsah:	800 – 2000Hz
Amplituda výstupní napětí:	max 5V
Informační LED:	indikace chyby, indikace vysílání
Rozhraní pro propojení s PC:	převodník UART -> USB
Odebíraný proud:	cca 60mA
Výstupní impedance:	1k Ω (výstupní potenciometr)

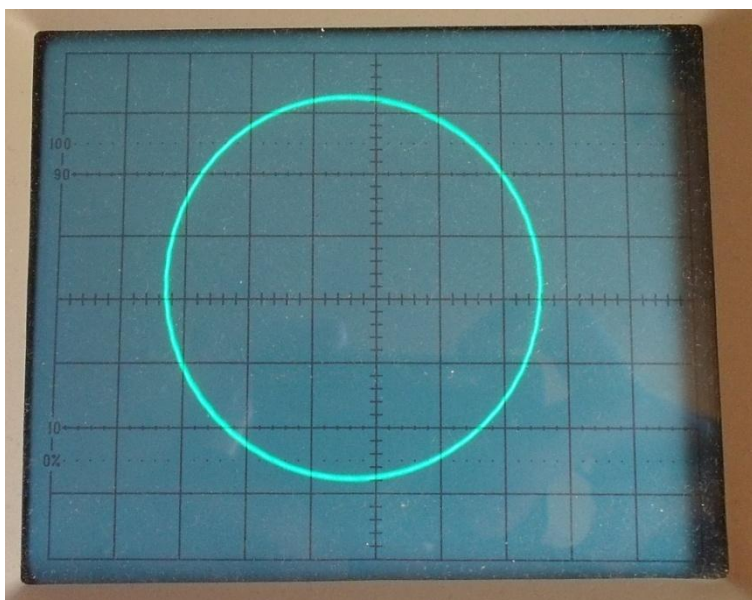
6.2. Výstupní průběhy signálů

Na obr. 17 se nachází průběh výstupního signálu modulátoru. Z obrázku je zřejmé, že průběh je sinusový a poměrně hladký. Fotografie byla pořízena během posílání zprávy.



Obr. 19 - Výstupní průběh signálu

Na obr. 18 je zobrazen fázový posun mezi výstupními signály modulátoru. Osciloskop je v módu x-y, což podle tvaru obrazce znamená, že fázový posun mezi signály je 90° , což bylo požadováno.



Obr. 20 - Fázový posuv výstupních signálů (90°)

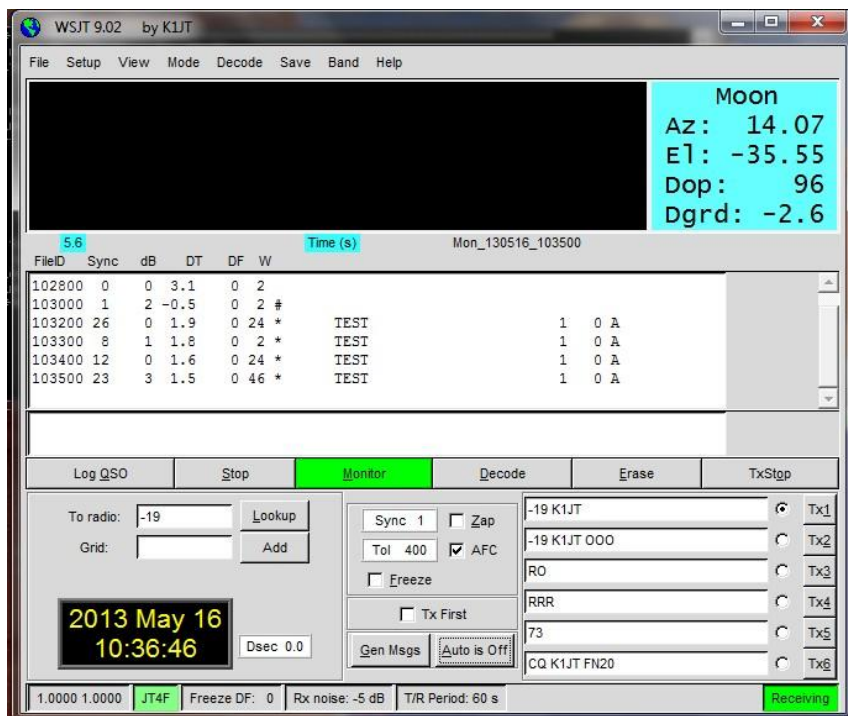
6.3. Výstupní spektrum a test zařízení

Spektrogram výstupního signálu modulátoru je na obr. 19. V tomto případě byl zvolen vysílací mód JT4F. Z obrázku vyplývá, že spektrum výstupního signálu modulátoru je poměrně čisté a neobsahuje nežádoucí harmonické složky signálu.



Obr. 21 - Spektrum výstupního signálu

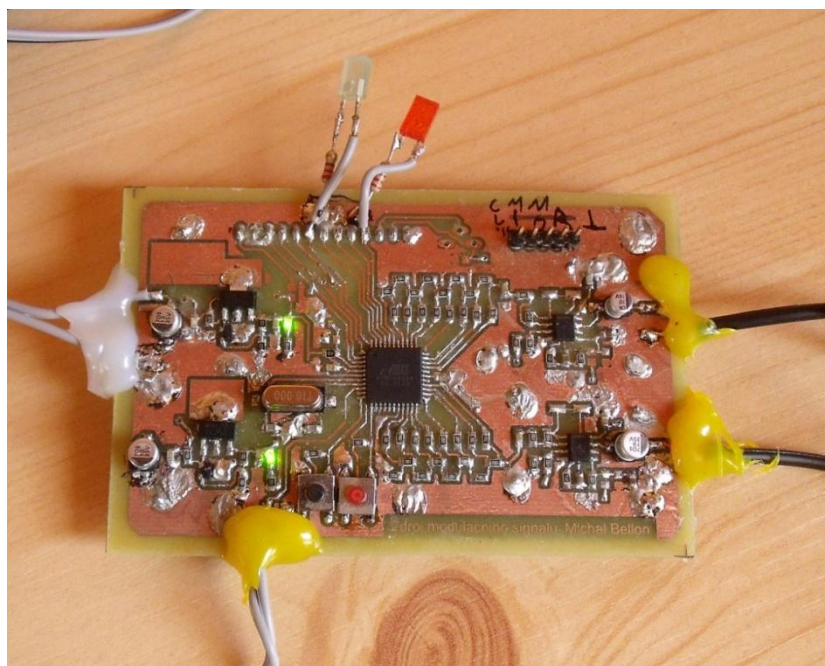
Na obr. 20 je vidět, že zařízení funguje. Do zařízení byla odeslána zkušební zpráva (TEST). Z obrázku je pozorovatelné, že zpráva se odesílá po minutě a je správně dekodována.



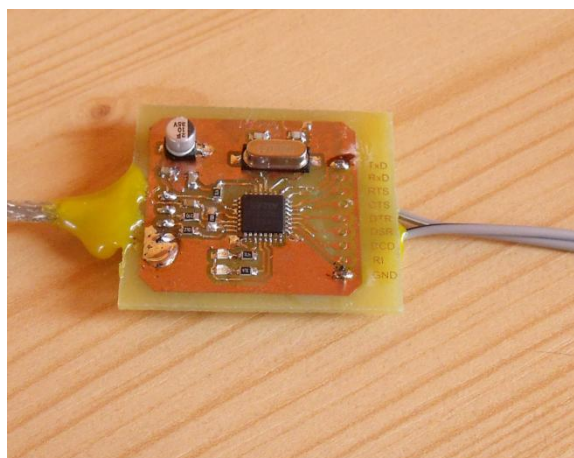
Obr. 22 – Ověření funkce zařízení

6.4. Fotografie zařízení

Na obr. 21 je fotografie starší verze modulátoru, která nepoužívá integrované D/A převodníky, ale odporovou R2R síť. Dále používá externí převodník UART -> USB, který je na obr. 22.



Obr. 23 - Fotografie modulátoru



Obr. 24 - Fotografie převodníku

7. Závěr

Cílem diplomové práce bylo zkonstruovat zařízení pro generaci signálu WSJT. V první části diplomové práce (semestrální projekt 1) byly prostudovány způsoby, jak by se dal výše požadovaný signál generovat. V další části diplomové práce (semestrální projekt 2) byl navržen hardware zařízení a v poslední části této práce software zařízení. Jako způsob pro generaci WSJT signálu byla zvolena přímá číslicová syntéza (DDS). A to z důvodu, že je to nejjednodušší řešení. Pro realizaci DDS není použit žádný speciální obvod, ale tato syntéza je realizována softwarově v mikrokontroléru AVR ATmega32.

Co se týče hardwarové části, jsou navrženy dvě verze tohoto zařízení. První verze má jako D/A převodníky použity R2R odporové sítě. A druhá verze, která je popsána v této práci používá jako D/A převodníky speciální integrované obvody. Deska druhé verze zařízení zatím nebyla z časových důvodů zapájena. V obou verzích zapojení je jako dolní propust použit filtr 2. řádu. Dále je v zapojení použit pro komunikaci s PC použit převodník z rozhraní UART na USB.

Důvod pro návrh druhé verze zapojení je ten, že integrované D/A převodníky mají obecně lepší vlastnosti než jednoduché R2R sítě. Dále na desce první verze zapojení nebyl integrován převodník UART – USB. Schéma zapojení zařízení je příloze 1, plošný spoj a rozmístění součástek v příloze 2 a v příloze 3.

V třetí části diplomové práce byl navržen software zařízení. Tento software byl psán v jazyku C a jako překladač byl použit program CodeVisionAVR. Pro naprogramování aplikace pro PC byl použit jazyk C#. Software byl napsán tak, aby se dal později případně modifikovat. Zdrojový kód programu pro procesor i pro PC aplikaci je uložen na doprovodném CD k diplomové práci.

Celkově se zadání diplomové práce podařilo splnit. Navržené zařízení bylo testováno pomocí programu WSJT, jeho skutečné parametry však nebyly změřeny.

Navržené zařízení se dá využít k jednoduché komunikaci mezi radioamatéry a dále pak například k experimentování při komunikaci s velmi slabými signály. Zařízení se dá připojit téměř k libovolné vysílačce.

8. Použitá literatura

- [1] PANDATRON. AVR DDS generátor [online]. [cit. 12. srpna 2011]. Dostupný z WWW: <http://pandatron.cz/?2800&avr_dds_generator>
- [2] PANDATRON. Nové DDS [online]. [cit. 26. dubna 2011]. Dostupný z WWW: <http://pandatron.cz/?2672&nove_dds_snizuji_rozmary_i_spotrebu>
- [3] TAYLOR, J., WSJT modulation [online]. [cit. 2012]. Dostupný z WWW: <<http://physics.princeton.edu/pulsar/K1JT/wsjt.html>>
- [4] MACKŮ, R. Návrh generátoru využívající přímou číslicovou syntézu (DDS) [online]. [cit. 2006]. Dostupný z WWW: http://www.urel.feec.vutbr.cz/~fryza/downloads/dp_macku_07.pdf
- [5] ZEDNÍK, L. Kapesní měřič indukčnosti a kapacity [online]. [cit. 2012]. Dostupný z WWW: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=62505
- [6] ČESKOSLOVENSKÝ DX KLUB, SSB, USB a LSB [online]. [cit. 2013]. Dostupný z WWW: <http://dx.cz/modules.php?name=Encyclopedia&op=content&tid=2>
- [7] MATHWORKS, Single Sideband Modulation via the Hilbert Transform [online]. [cit. 2012]. Dostupný z WWW: <http://www.mathworks.com/help/signal/examples/single-sideband-modulation-via-the-hilbert-transform.html>
- [8] DEVBOOK.CZ, Programování v jazyce C – Základy [online]. [cit. 2012]. Dostupný z WWW: <http://www.devbook.cz/programovani-v-jazyce-c-zaklady>
- [9] TME. Transfer Multisort Elektronik [online]. [cit. 2013]. Dostupný z WWW: <<http://www.tme.eu/cz/>>
- [10] FRÝZA, T., Mikroprocesorová technika, Elektronické skriptum. Brno: FEKT VUT v Brně, 2008
- [11] DATASHEETCATALOG. Katalogové listy elektronických součástek [online] [cit. 2013]. Dostupný z WWW: <<http://www.datasheetcatalog.com>>

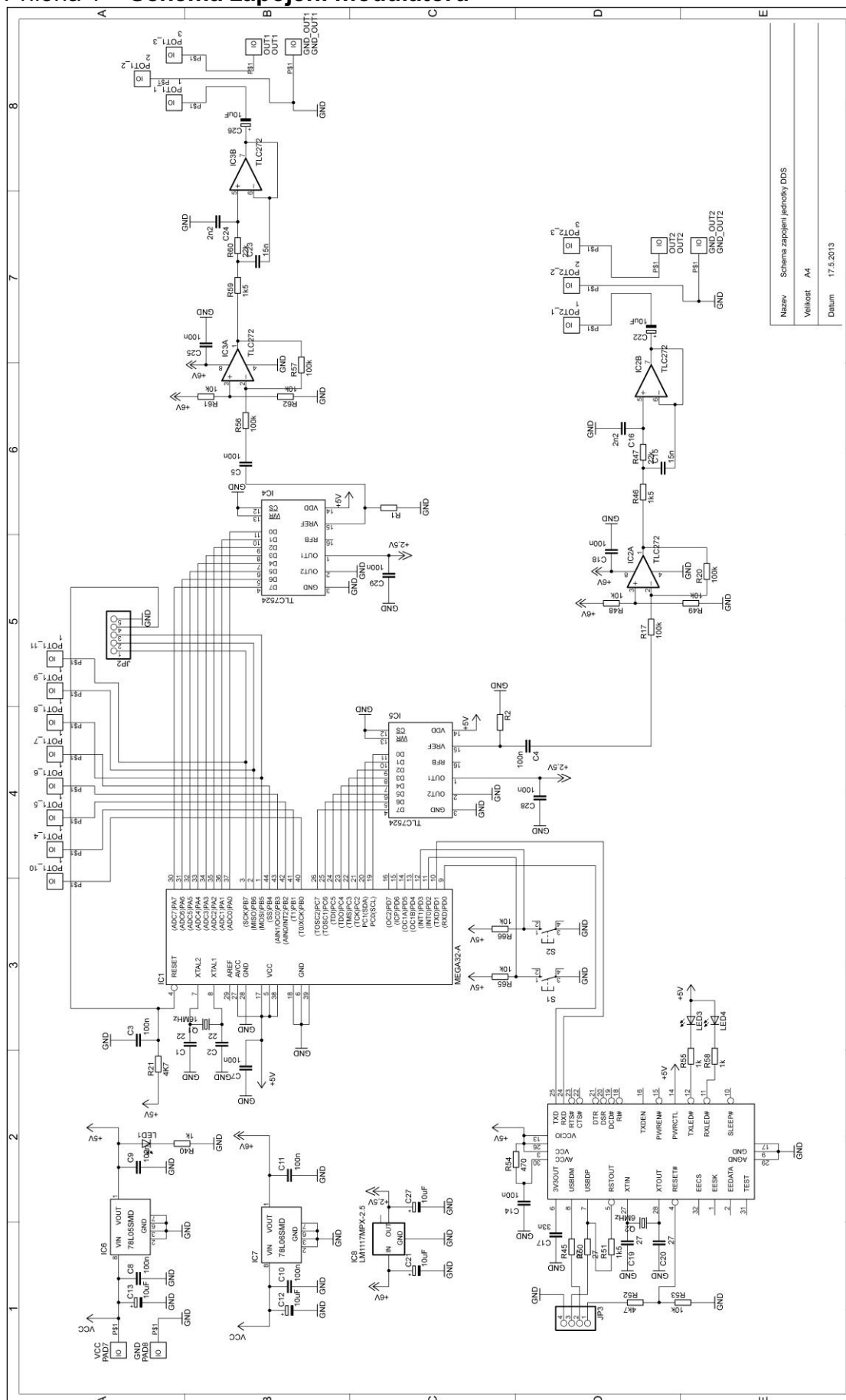
- [12] RONEŠOVÁ, A., Oscilátor s Wienovým článkem [online] [cit. 2009]. Dostupný z WWW: <http://home.zcu.cz/~ronesova/index.php?menuitem=aes_mwg>
- [13] TEXAS INSTRUMENTS, Katalogové listy elektronických součástek [online]. [cit.2013]. Dostupný z WWW: <<http://www.ti.com/>>

Přílohy

Seznam příloh

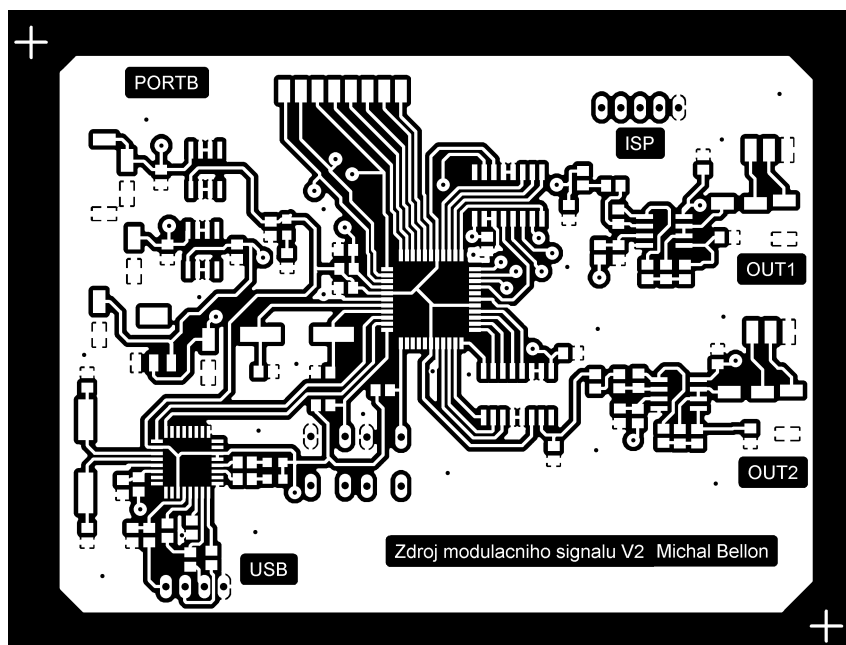
- Příloha 1 – Schéma zapojení modulátoru
- Příloha 2 – Plošný spoj modulátoru
- Příloha 3 – Rozmístění součástek na plošném spoji
- Příloha 4 – Seznam součástek

Príloha 1 – Schéma zapojení modulátoru

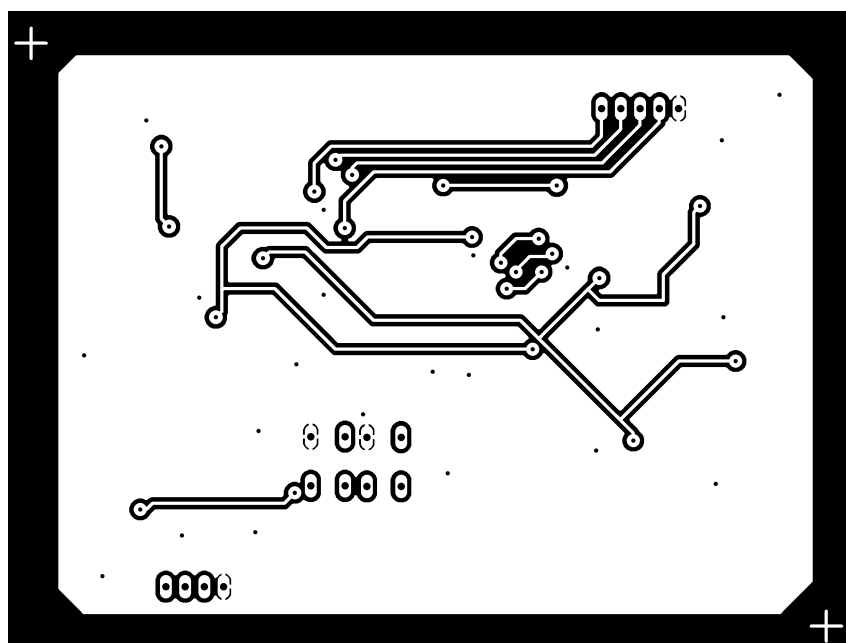


Příloha 2 – Plošný spoj modulátoru

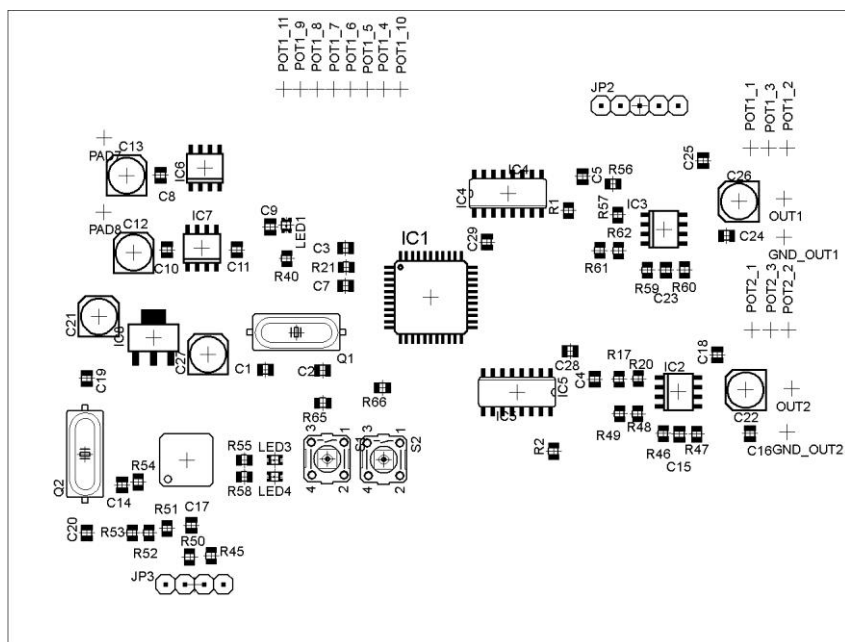
Horní strana:



Dolní strana:



Příloha 3 - Rozmístění součástek



Příloha 4 – Seznam součástek

Partlist

Exported from DDS.brd at 16.5.2013 18:56:13

EAGLE Version 5.6.0 Copyright (c) 1988-2009 CadSoft

Part	Value	Package	Library	Position (mm)	
Orientation					
C1	22	C0805	rc1	(28.98 28.56)	
R180					
C2	22	C0805	rc1	(36.54 28.44)	R0
C3	100n	C0805	rc1	(39.56 44.64)	R0
C4	100n	C0805	rc1	(72.46 27.28)	R90
C5	100n	C0805	rc1	(70.84 54.1)	
R270					
C7	100n	C0805	rc1	(39.54 39.64)	R0
C8	100n	C0805	rc1	(15.14 54.24)	
R270					
C9	100n	C0805	rc1	(29.64 47.42)	
R270					
C10	100n	C0805	rc1	(16 44.4)	
R270					
C11	100n	C0805	rc1	(25.2 44.4)	
R270					
C12	10uF	153CLV-0505	rc1	(11.6 44)	R90
C13	10uF	153CLV-0505	rc1	(10.67 54.22)	R90
C14	100n	C0805	rc1	(10.1 13.3)	R90
C15	15n	C0805	rc1	(83.64 20.05)	R90
C16	2n2	C0805	rc1	(92.96 20.13)	
R270					
C17	33n	C0805	rc1	(19.2 8)	R90
C18	100n	C0805	rc1	(88.64 30.49)	R90
C19	27	C0805	rc1	(5.4 27.4)	
R270					
C20	27	C0805	rc1	(5.4 7)	R90
C21	10uF	153CLV-0505	rc1	(7 35.6)	R90
C22	10uF	153CLV-0505	rc1	(92.36 25.87)	
R180					
C23	15n	C0805	rc1	(81.84 41.71)	R90
C24	2n2	C0805	rc1	(89.83 46.26)	R0
C25	100n	C0805	rc1	(86.74 56.15)	R90
C26	10uF	153CLV-0505	rc1	(91.45 50.77)	
R180					
C27	10uF	153CLV-0505	rc1	(21.4 30.6)	R90
C28	100n	C0805	rc1	(69.2 31)	
R180					
C29	100n	C0805	rc1	(58.2 45.4)	R90
GND_OUT1	GND_OUT1	PAD_1.5X3	PAD	(97.42 46.06)	R0
GND_OUT2	GND_OUT2	PAD_1.5X3	PAD	(97.86 20.36)	R0
IC1	MEGA32-A	TQFP44	atmel	(50.8 38.1)	R0
IC2	TLC272	SO08	linear	(83.48 25.63)	
R270					
IC3	TLC272	SO08	linear	(81.58 47.12)	
R270					
IC4	TLC7524	SO16	tlc7524	(61 51.8)	R0
IC5	TLC7524	SO16	tlc7524	(62.2 25.6)	
R180					
IC6	78L05SMD	SO08	linear	(21 55.2)	R0

IC7	78L06SMD	SO08	linear	(20.6 44.6)	R0
IC8	LM1117MPX-2.5	SOT223	lm1117	(14.2 32.8)	R0
JP2		1X05	pinhead	(78.4 63.4)	R0
JP3		1X04	pinhead	(19.6 0.2)	R0
LED1		CHIPLED_0805	led	(31.76 47.63)	
R180					
LED3		CHIPLED_0805	led	(30.2 16.6)	R90
LED4		CHIPLED_0805	led	(30.2 14.4)	R90
OUT1	OUT1	PAD_1.5X3	PAD	(97.45 51.16)	R0
OUT2	OUT2	PAD_1.5X3	PAD	(98.39 26.06)	R0
PAD7	VCC	PAD_1.5X3	PAD	(7.68 59.18)	R0
PAD8	GND	PAD_1.5X3	PAD	(7.65 49.38)	R0
POT1_1	1	PAD_1.5X3	PAD	(93 57.8)	R90
POT1_2	2	PAD_1.5X3	PAD	(97.8 57.8)	R90
POT1_3	3	PAD_1.5X3	PAD	(95.4 57.8)	R90
POT1_4	1	PAD_1.5X3	PAD	(44.6 65.6)	R90
POT1_5	1	PAD_1.5X3	PAD	(42.4 65.6)	R90
POT1_6	1	PAD_1.5X3	PAD	(40.2 65.6)	R90
POT1_7	1	PAD_1.5X3	PAD	(38 65.6)	R90
POT1_8	1	PAD_1.5X3	PAD	(35.8 65.6)	R90
POT1_9	1	PAD_1.5X3	PAD	(33.6 65.6)	R90
POT1_10	1	PAD_1.5X3	PAD	(46.8 65.6)	R90
POT1_11	1	PAD_1.5X3	PAD	(31.4 65.6)	R90
POT2_1	1	PAD_1.5X3	PAD	(93.2 33.8)	R90
POT2_2	2	PAD_1.5X3	PAD	(98 33.8)	R90
POT2_3	3	PAD_1.5X3	PAD	(95.6 33.8)	R90
Q1	16MHz	HC49UP	crystal	(33.09 33.47)	
R180					
Q2	6MHz	HC49UP	crystal	(5.2 17.4)	R90
R1		R0805	rc1	(69 49.6)	R90
R2		R0805	rc1	(67 17.8)	R90
R17	100k	R0805	rc1	(75.62 27.31)	
R270					
R20	100k	R0805	rc1	(78.18 27.39)	R90
R21	4K7	R0805	rc1	(39.64 42.12)	R0
R40	1k	R0805	rc1	(31.8 43.24)	R90
R45	27	R0805	rc1	(21.8 4)	R90
R46	1k5	R0805	rc1	(81.48 20.11)	
R270					
R47	22k	R0805	rc1	(85.92 20.04)	R90
R48	10k	R0805	rc1	(78.08 22.71)	
R270					
R49	10k	R0805	rc1	(75.72 22.74)	R90
R50	27	R0805	rc1	(19 3.8)	R90
R51	1k5	R0805	rc1	(16 7.6)	R90
R52	4k7	R0805	rc1	(13.6 7)	R90
R53	10k	R0805	rc1	(11.4 7)	
R270					
R54	470	R0805	rc1	(12.2 13.7)	
R270					
R55	1k	R0805	rc1	(26.2 16.6)	R0
R56	100k	R0805	rc1	(74.84 53.08)	R0
R57	100k	R0805	rc1	(75.52 48.99)	R90
R58	1k	R0805	rc1	(26.2 14.4)	R0
R59	1k5	R0805	rc1	(79.38 41.71)	
R270					
R60	22k	R0805	rc1	(84.32 41.74)	R90
R61	10k	R0805	rc1	(73.08 44.31)	
R270					
R62	10k	R0805	rc1	(75.54 44.29)	R90
R65	10k	R0805	rc1	(36.56 24.14)	

R180					
R66	10k	R0805	rcl	(44.41 26.14)	
R180					
S1		B3F-10XX	switch-omron	(37.18 16.78)	
R270					
S2		B3F-10XX	switch-omron	(44.58 16.71)	
R270					
U\$1	FT232BM32LQFP	32LQFP	ft232bm	(18.6 16.6)	R90