

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

PŘÍMÝ ČÍSLICOVÝ SYNTEZÁTOR PRO MIKROVLNNÉ  
APLIKACE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. VOJTĚCH DLUHÝ

Brno 2015



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

## PŘÍMÝ ČÍSLICOVÝ SYNTÉZÁTOR PRO MIKROVLNNÉ APLIKACE

DIRECT DIGITAL SYNTHESIZER FOR MICROWAVE APPLICATION

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. VOJTĚCH DLUHÝ

VEDOUCÍ PRÁCE  
SUPERVISOR

prof. Ing. MIROSLAV KASAL, CSc.

BRNO 2015



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Vojtěch Dluhý

**ID:** 134470

**Ročník:** 2

**Akademický rok:** 2014/2015

## NÁZEV TÉMATU:

**Přímý číslicový syntezátor pro mikrovlnné aplikace**

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s řešením přímých číslicových syntezátorů, zejména s integrovanými obvody Analog Devices. Navrhněte přímý číslicový syntezátor do 120 MHz s řízením po sériové sběrnici a s možností synchronizace externím frekvenčním normálem. Zvláštní pozornost věnujte spektrální čistotě generovaného signálu.

Dílčí obvody syntezátoru realizujte a ověřte jejich vlastnosti. Navrhněte plošný spoj syntezátoru. Napište ovladač program pro ovládání syntezátoru po sériové sběrnici s vhodným uživatelským rozhraním. Syntezátor sestavte a experimentálně ověřte jeho vlastnosti.

## DOPORUČENÁ LITERATURA:

[1] DOBEŠ, J., ŽALUD, V. Moderní radiotechnika. Praha: BEN - technická literatura, 2006.

[2] KASAL, M. Frekvenční syntéza v komunikačních systémech - Experimentální družice. Edice Habilitační a inaugurační spisy, sv. 169. Brno: Nakladatelství VUTUM, 2005.

**Termín zadání:** 9.2.2015

**Termín odevzdání:** 13.8.2015

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

**Konzultanti diplomové práce:**

**doc. Ing. Tomáš Kratochvíl, Ph.D.**

*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**

Cílem této diplomové práce je seznámit čtenáře se základy číslicové kmitočtové syntézy a navrhnout přímý číslicový syntežátor s obvodem AD9951 od společnosti Analog Devices. Zařízení je řízeno z PC rozhraním USB. Zařízení pracuje s vnitřním oscilátorem, s možností připojení externího kmitočtového normálu o frekvenci 10 MHz. Na vstupu zařízení je tranzistorový zdvojovač kmitočtu. Výstupní signál je filtrován dolní propustí na kmitočtu 120 MHz a zesílen trojbodovým zesilovačem ERA-3+.

## **KLÍČOVÁ SLOVA**

frekvenční syntéza, kmitočtová syntéza, syntežátor, přímý číslicový syntežátor, AD9951, kmitočtový zdvojovač

## **ABSTRACT**

The aim of this thesis is introduce readers to the basics of digital frequency synthesis and design of direct digital synthesizer with circuit AD9951 by Analog Devices. The device will be controlled from a PC via USB. The device works with internal oscillator, with the ability to connect an external frequency standard of 10 MHz. On input is frequency doubler with transistor. Output signal is filtered by low-pass filter and amplified by monolithic amplifier ERA-3+.

## **KEYWORDS**

frequency synthesis, synthesizer, DDS, direct digital synthesis, AD9951, frequency doubler

DLUHÝ, Vojtěch *Přímý číslicový syntežátor pro mikrovlnné aplikace*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2015. 57 s. Vedoucí práce byl prof. Ing. Miroslav Kasal, CSc.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Přímý číslicový syntezátor pro mikrovlonné aplikace“ 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.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Prof. Ing. Miroslavu Kasalovi, CSc. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. A také všem, kteří mi byli nápomocni každou dobrou radou.

Brno .....

.....

(podpis autora)

# OBSAH

<b>Úvod</b>	<b>10</b>
<b>1 Úvod k syntezátorům</b>	<b>11</b>
1.1 Syntezátory s fázovým závěsem . . . . .	11
1.2 Přímé číslicové syntezátory . . . . .	12
<b>2 Návrh syntezátoru</b>	<b>14</b>
2.1 Napájecí zdroje . . . . .	14
2.2 Obvody řízení . . . . .	16
2.3 Obvod číslicové syntézy AD9951 . . . . .	16
2.4 Výstupní filtr . . . . .	17
2.5 FW mikroprocesoru . . . . .	19
2.6 SW pro počítač . . . . .	19
<b>3 Realizace navrženého syntezátoru</b>	<b>20</b>
3.1 Napájecí zdroj . . . . .	20
3.2 Převodník USB/UART . . . . .	20
3.3 Mikroprocesor . . . . .	20
3.4 Obvody syntézy . . . . .	21
3.5 Ovládací SW . . . . .	22
3.6 FW mikroprocesoru . . . . .	23
3.7 Mechanické provedení . . . . .	25
3.8 Oživení a programování . . . . .	25
<b>4 Experimentální měření obvodu</b>	<b>28</b>
4.1 Zobrazení časových průběhů . . . . .	28
4.2 Zobrazení kmitočtových spekter . . . . .	30
4.3 Fázový šum . . . . .	37
<b>5 Závěr</b>	<b>41</b>
<b>Literatura</b>	<b>43</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>44</b>
<b>Seznam příloh</b>	<b>45</b>
<b>A Schéma zdroje</b>	<b>46</b>

<b>B</b>	<b>Schéma řídicích obvodů</b>	<b>48</b>
<b>C</b>	<b>Schéma obvodu syntezátoru</b>	<b>50</b>
<b>D</b>	<b>Desky plošných spojů</b>	<b>52</b>
<b>E</b>	<b>Zdrojový kód FW</b>	<b>54</b>



# SEZNAM OBRÁZKŮ

1.1	Smyčka fázového závěsu . . . . .	11
1.2	Příklad výstupního spektra PLL s viditelným fázovým šumem[1] . . .	12
1.3	Blokové schéma DDS . . . . .	12
1.4	Typické spektrum obvodu DDS [2] . . . . .	13
2.1	Blokové schéma zařízení . . . . .	14
2.2	Schéma 1,8 V s obvodem LM317 . . . . .	15
2.3	Blokové schéma obvodu AD9951[3] . . . . .	17
2.4	Modulová a fázová charakteristika výstupního filtru . . . . .	18
2.5	Schéma výstupního filtru . . . . .	19
3.1	Hlavní okno ovládacího programu . . . . .	22
3.2	Detailní nastavení obvodu AD9951 v ovládacím SW . . . . .	23
3.3	Vývojový diagram FW mikroprocesoru . . . . .	24
3.4	Rozmístění hlavních částí obvodu . . . . .	25
3.5	Fotografie osazené prototypové desky . . . . .	26
3.6	Časový průběh UART komunikace mezi PC a mikroprocesorem . . .	27
3.7	Časový průběh UART komunikace mezi procesorem a AD9951 . . . .	27
4.1	Časový průběh výstupního signálu o kmitočtu 1 MHz. . . . .	28
4.2	Časový průběh výstupního signálu o kmitočtu 10 MHz. . . . .	29
4.3	Časový průběh výstupního signálu o kmitočtu 100 MHz. . . . .	29
4.4	Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 200 MHz. .	30
4.5	Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 20 MHz. . .	30
4.6	Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 200 MHz. .	31
4.7	Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 20 MHz. .	31
4.8	Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 200 MHz. .	32
4.9	Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 20 MHz. .	32
4.10	Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 200 MHz. .	33
4.11	Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 20 MHz. .	33
4.12	Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 200 MHz. .	34
4.13	Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 20 MHz. .	34
4.14	Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 200 MHz. .	35
4.15	Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 20 MHz. .	35
4.16	Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 50 kHz s markery, při použití externího zdroje hodinového signálu. . . . .	36
4.17	Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 200 MHz, při použití externího zdroje hodinového signálu. . . . .	36
4.18	Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 50 kHz s markery. . . . .	37

4.19	Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 50 kHz s markery. . . . .	38
4.20	Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 50 kHz s markery. . . . .	38
4.21	Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 50 kHz s markery. . . . .	39
4.22	Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 50 kHz s markery. . . . .	39
4.23	Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 50 kHz s markery. . . . .	40
D.1	DPS - strana součástí (horní) . . . . .	53
D.2	DPS - strana spojů (spodní) . . . . .	53

# ÚVOD

Tato diplomová práce je zaměřena na seznámení se základními principy syntezátorů harmonických signálů, na obecné porovnání jejich vlastností a jejich využití v praxi. Dále se zabývá návrhem přímého číslicového syntezátoru do kmitočtu 120 MHz. Harmonické signály lze generovat několika různými způsoby. Prvním a nejjednodušším jsou oscilátory s RC či LC články, které generují čistě analogový harmonický signál. Jejich největší nevýhodou ovšem je jejich velice špatná přeladitelnost. Dále je možné generovat signály sinusového průběhu pomocí krystalových oscilátorů, které jsou zcela nepřeladitelné. Další možností jsou generátory se smyčkou fázového závesu a také přímé číslicové generátory, kterým se budu ve své práci také věnovat. Další a nejpodstatnější částí této práce je návrh a konstrukce přímého číslicového syntezátoru s obvodem AD9951 řízeným z PC sběrnici USB. USB komunikaci obstarává obvod FT232 s procesorem PIC16F628A.

Vstupní signál externího normálu je upraven zdvojovačem s bipolárním tranzistorem na kmitočet potřebný pro správnou funkci obvodu AD9951, výstupní signál je filtrován filtrem 7. řádu s Caurovou aproximací. Výstupní signál je zesílen trojbovým zesilovačem ERA-3+.

Dovolené napájecí napětí je 12 V, které je dále upraveno čtyřmi zdroji (+1,8 V, +1,8 V, +3,3 V a +5 V).

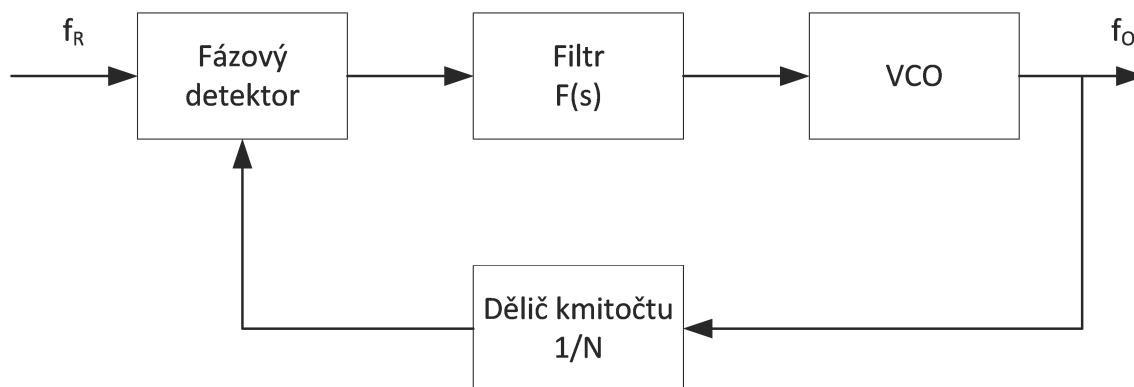
Zařízení je jednodeskové konstrukce s motivem plošných spojů po obou stranách. Vstup externího normálu a výstup signálu je na konektorech SMA.

# 1 ÚVOD K SYNTEZÁTORŮM

Frekvenční syntezátory jsou integrované obvody používané v moderní technice jako náhrada oscilátorů pro generování harmonických signálů různých kmitočtů. Dnes jsou vyráběny nejčastěji v SMD provedení integrovaných obvodů s rozličným počtem vývodů, podle jejich obvodové složitosti. Nedílnou součástí syntezátorů je obvod pro zvyšování vstupního kmitočtu z oscilátoru, tímto obvodem může být jak násobič kmitočtu, tak smyčka fázového závěsu (PLL). Dalším prvkem je samotný obvod generující výstupní signál, tímto může být jak smyčka fázového závěsu nebo obvod přímé číslicové syntézy. Hlavními výhodami syntezátorů oproti oscilátorům jsou široká možnost přeladění výstupního kmitočtu a jednoduchost tohoto úkonu, který se provádí většinou určitým řídicím slovem z připojeného mikroprocesoru.

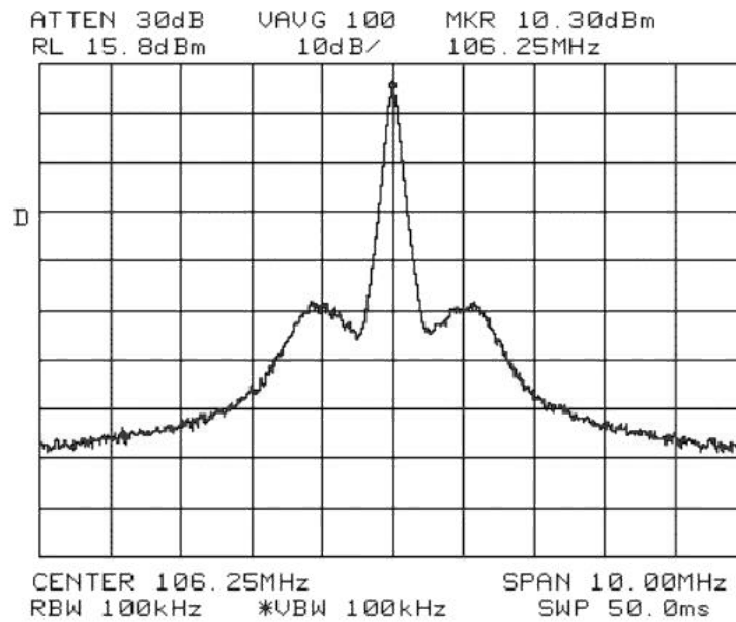
## 1.1 Syntezátory s fázovým závěsem

Syntezátory s fázovým závěsem (PLL - smyčka fázového závěsu – Phase-Locked loop) pracují, jak plyne z jejich názvu, se smyčkou fázového závěsu. Ta sestává ze čtyř základních prvků (obr. 1.1), kterými jsou fázový detektor, filtr smyčky, napětím řízený oscilátor (VCO) a volitelně ve zpětné vazbě dělič kmitočtu. Fázový detektor zjišťuje



Obr. 1.1: Smyčka fázového závěsu

rozdíl fáze mezi vstupním (referenčním) a zpětnovazebním signálem, kdy na jeho výstupu je signál odpovídající jejich rozdílu. Stejnoseměrná složka výstupního signálu fázového detektoru řídí VCO a jeho výstupní kmitočet. Čím více se fáze výstupního signálu blíží fázi vstupního signálu tím stabilnější je signál řídící VCO. Výstupní signál z VCO se zpětnou vazbou přes dělič kmitočtu vrací do detektoru fáze. Takto postupně dochází ke stabilizaci výstupního kmitočtu. Dosažením požadovaného výstupního kmitočtu se smyčka fázového závěsu tzv. uzamyká.



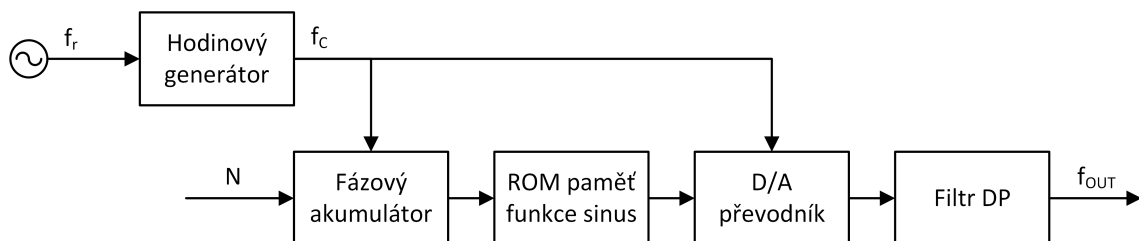
Obr. 1.2: Příklad výstupního spektra PLL s viditelným fázovým šumem[1]

Změna výstupního kmitočtu se provádí změnou dělicího poměru ve zpětné vazbě. Výstupní signál pak má kmitočet  $f_O$  dle vzorce 1.1.

$$f_O = N \cdot f_R \quad (1.1)$$

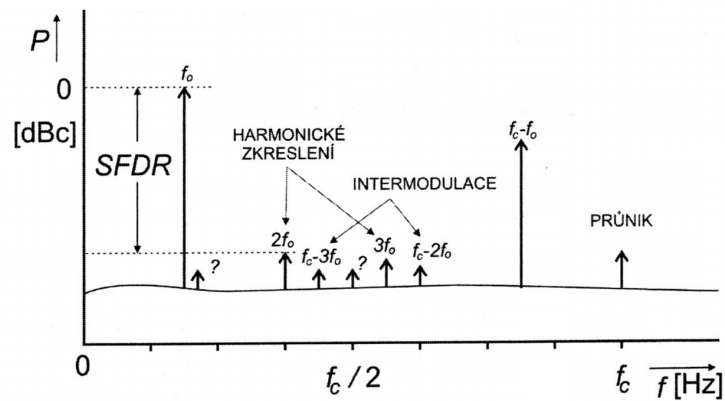
kde  $f_O$  je výstupní kmitočet,  $N$  je dělicí poměr a  $f_R$  je vstupní (referenční) kmitočet. Výstupní spektrum tohoto typu syntezátoru obsahuje dvě hlavní parazitní složky, které jsou způsobeny parazitní fázovou a amplitudovou modulací. Příklad výstupního signálu s fázovým šumem je na obr. 1.2.

## 1.2 Přímé číslicové syntezátory



Obr. 1.3: Blokové schéma DDS

Přímé číslicové syntezátory (DDS) pracují na principu vyčítání hodnot uložených v paměti ROM daným kmitočtem. S hodinovým kmitočtem se zvyšuje hodnota fá-

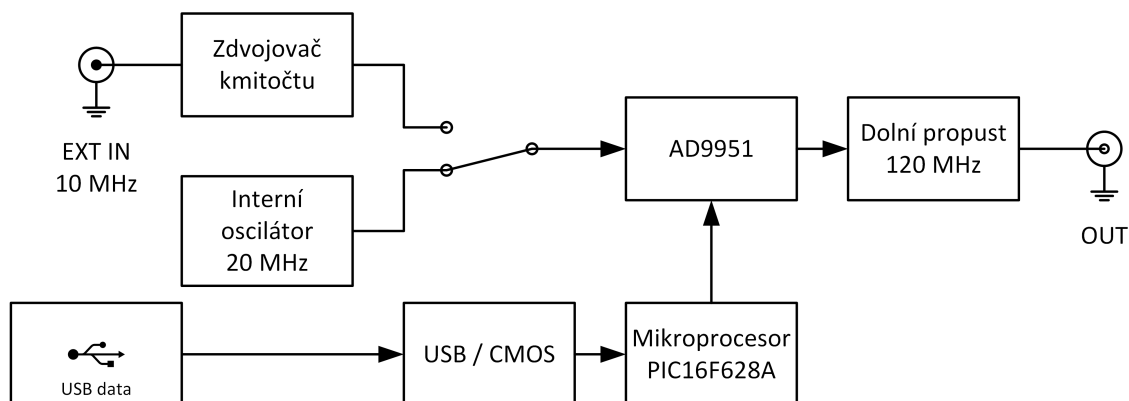


Obr. 1.4: Typické spektrum obvodu DDS [2]

zového akumulátoru o hodnotu odpovídající ladícímu slovu. Tato hodnota je ukazatelem na hodnoty uložené v ROM paměti. Výstupem z paměti je diskrétní průběh funkce sinus, který vstupuje do A/D převodníku, který jej převádí na hodnotu analogovou, byť značně zkreslenou (schodovitou). Na výstupu je proto filtr typu dolní propust, který odfiltruje vyšší kmitočty, které způsobují ono schodovité zkreslení. Základní blokové schéma je na obr.1.3.

Typické spektrum včetně parazitních signálů můžeme pozorovat na obr.1.4. Můžeme zde vidět zrcadlení kolem  $f_c/2$  a také parametr SFDR, který ukazuje amplitudový odstup parazitních spektrálních složek od požadovaného kmitočtu.

## 2 NÁVRH SYNTÉZÁTORU



Obr. 2.1: Blokové schéma zařízení

Navrhovaný syntezátor se bude skládat z následujících základních částí:

- dva zdroje +1,8 V pro analogovou a digitální část obvodu AD9951
- zdroj +3,3 V pro I/O obvody IO AD9951, mikroprocesor PIC16F628A a část převodníku LT232
- zdroj +5 V pro převodník LT232 a trojbodové zesilovače ERA-3+
- analogový přepínač ADG1219
- interní krystalový oscilátor 20 MHz
- zdvojovač vstupního kmitočtu z externího vstupu, tj. z 10 MHz na 20 MHz
- AD9951 - obvod přímé číslicové syntézy fy Analog Devices
- PIC16F628A - mikroprocesor fy Microchip pro řízení obvodu AD9951
- LT232 - převodník USB na UART fy FTDI
- filtr typu dolní propust na kmitočtu 120 MHz

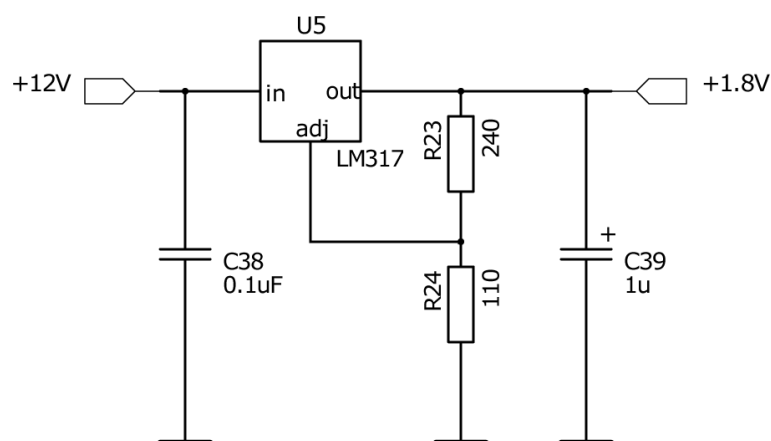
Jednotlivým částem se budu podrobněji věnovat v následujících podkapitolách.

Celé zařízení bude na jediné desce plošných spojů (DPS). Čímž bude splněn požadavek na minimální velikost.

### 2.1 Napájecí zdroje

Zařízení bude napájeno z externího zdroje s nominálním napětím 12 V (např. palubní síť osobního automobilu nebo externí zdroj). Z tohoto napětí budou dva obvody LM317 v zapojení napěťového stabilizátoru vytvářet napětí +1,8 V pro obvod AD9951 (mimo jeho I/O obvody), jeden stabilizátor MC33269DT-3.3 s výstupním napětím +3,3 V pro I/O obvody IO AD9951, mikroprocesor PIC16F628A a I/O část převodníku FT232 a jako poslední obvod 7805 pro napájení obvodu LT232.

**Zdroje 1,8 V** budou tvořeny základním katalogovým zapojením obvodu LM317 nastaveným pomocí rezistorového děliče na výstupní napětí 1,8 V. Maximální výstupní proud obvodu je 0,5 A, což je pro analogovou část obvodu AD9951 plně dostačující, jelikož celý obvod má při napájení 1,8 V příkon maximálně 171 mW<sup>1</sup>. Výstupní napětí ze stabilizátoru bude filtrováno kapacitorem, aby bylo zajištěno jeho nízké zvlnění, které by mohlo ovlivňovat kvalitu výstupního signálu. Rezistory ve zpětné vazbě jsou vypočteny dle vzorců v katalogovém listě obvodu[4]. Zapojení regulovatelného stabilizátoru je zobrazeno na obr.2.2.



Obr. 2.2: Schéma 1,8 V s obvodem LM317

**Zdroj 3,3 V** bude tvořen katalogovým zapojením obvodu NCP1117DT33[5], jehož výstupní napětí bude opět filtrováno kapacitorem, přičemž nemusí být filtrováno tak dokonale jako napájení analogové části, jelikož použité digitální obvody jsou schopny pracovat s rozptylem napájecího napětí od 3,0 V do 4,0 V. Maximální výstupní proud tohoto zdroje činí 1 A, což je dostatečná hodnota, jelikož součet odběrů všech zde připojených obvodů nepřesahuje 0,5 A.

**Zdroj 5 V** bude tvořen katalogovým zapojením obvodu MC7805[6], jehož výstupní napětí bude opět filtrováno kapacitorem. Maximální výstupní proud tohoto obvodu je 1 A, což je pro napájení obvodu FT232 a trojbodových zesilovačů ERA-3+ plně dostačující. K tomuto zdroji je přes rezistor připojena LED dioda, která signalizuje přítomnost napájecího napětí.

Celkové schéma zdroje je umístěno v příloze A.

<sup>1</sup>Maximální spotřeba (příkon) 171 mW platí při AVDD=DVDD=1,8 V pro obě napájecí větve.



## 2.2 Obvody řízení

Zařízení bude ovládáno z PC pomocí sériové sběrnice USB. Aby bylo možné vybraný mikroprocesor připojit k této sběrnici, je nutné mu předřadit převodník z USB na UART. Jako převodník byl vybrán obvod společnosti FTDI typu FT232[8], jehož výhodou je možnost napájení I/O obvodů napětím 3,3 V, což umožní přímé propojení s mikroprocesorem PIC16F628A, který bude také napájen napětím 3,3 V. Z počítače budou pomocí USB přenášeny do mikroprocesoru instrukce, vytvářené pomocí grafického uživatelského rozhraní SW, pro ovládání obvodu DDS AD9951. Obvod FT232 se počítači hlásí jako virtuální COM port, což velice zjednoduší programování SW, než kdyby musel obsahovat obsluhu USB portu. K obvodu FT232 je také připojena svítivá dioda (LED), která bude indikovat probíhající komunikaci na straně UARTu.

Mikroprocesor PIC16F628A[7] bude sloužit pouze pro překlad instrukcí přenášených z PC do formátu sériových dat srozumitelného pro obvod AD9951. Procesor obsahuje další, v tomto zařízení nevyužité, periferie jako jsou A/D převodník, čítače atp. Pro jednoduchost obsluhy není v obvodu procesoru umístěno tlačítko reset. Reset celého zařízení lze provést odpojením a následným připojením napájecího napětí. Schéma řídicího obvodu je umístěno v příloze B.

## 2.3 Obvod číslicové syntézy AD9951

Srdcem celého zařízení bude obvod DDS AD9951 společnosti Analog Devices[3]. Z katalogového listu a blokového schématu (obr.2.3) můžeme vyčíst:

- maximální hodinový kmitočet 400 MHz
- integrovaný 14-bitový D/A převodník
- 32-bitové ladící slovo
- fázový šum  $\leq -120\text{dBc/Hz}$  @ 160 MHz
- sériová komunikace
- podpora 5 V signálů na datových vstupech
- PLL násobička vstupního kmitočtu
- synchronizace více těchto obvodů a další

Z uvedených parametrů můžeme s jistotou říct, že maximální možný výstupní kmitočet obvodu může být 200 MHz. Na délce ladícího slova a hodinového kmitočtu závisí krok přeladění, který při užití vzorce 2.1 pro výpočet výstupního kmitočtu  $f_{OUT}$  ve vztahu 2.2, vychází na  $\Delta f = 0,093$  Hz.

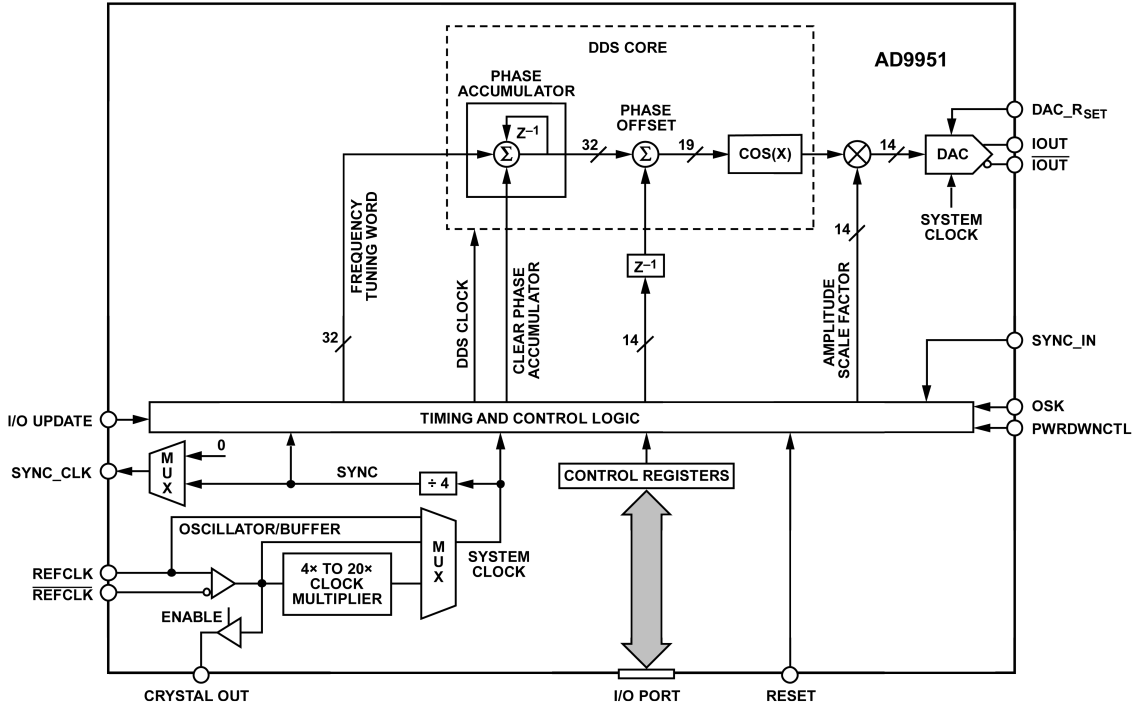
$$f_{OUT} = (FTW) \cdot (f_S) / 2^{32} \quad (2.1)$$

$$\Delta f = (FTW) \cdot (f_S) / 2^{32} - (FTW - 1) \cdot (f_S) / 2^{32} \quad (2.2)$$

kde  $f_S=400$  MHz a FTW je ladící slovo.

$$R_{SET} = 39,19/I_{OUT} \quad (2.3)$$

Pro nejlepší SFDR je potřeba omezit výstupní proud  $I_{OUT}$  na 10 mA, proto dle vzorce 2.3 vypočítáme rezistor, který bude připojen k vývodu DAC\_RSET. Jeho hodnota činí dle řady E12  $R_{SET}=3,9k\Omega$ .



Obr. 2.3: Blokové schéma obvodu AD9951[3]

## 2.4 Výstupní filtr

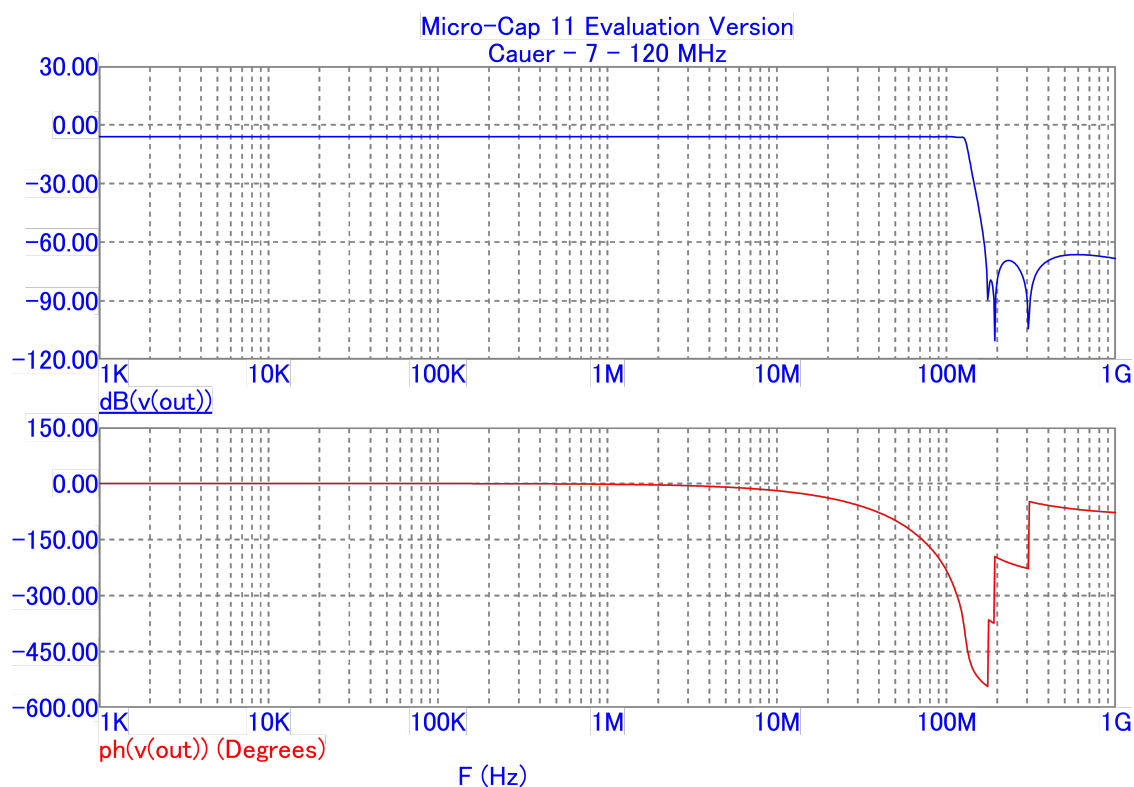
Na výstupu obvodu AD9951 je sice již analogový signál, ovšem značně zkreslený, což je způsobeno vyčítáním číselných hodnot z tabulky ROM. Proto na výstupu obvodu syntézy musí být umístěn výstupní filtr s mezním kmitočtem na nejvyšší nastavitelné hodnotě, který zde slouží jako filtr anti-aliasingový<sup>2</sup>. V tomto případě se jedná o 120 MHz. Při posouzení několika aproximací filtrů dolní propusti jsem se uchýlil k návrhu Caueirova přičkového filtru sedmého řádu, který má velkou strmou a dostatečný útlum v nepropustné části ( $> 60$  dB). Výstup zařízení je požadován nesymetrický, zatímco výstup obvodu je symetrický. Nesymetrizaci by bylo

<sup>2</sup>Anti-aliasingový filtr slouží k odfiltrování vyšších harmonických, které jsou na výstupu A/D převodníku nežádoucí.

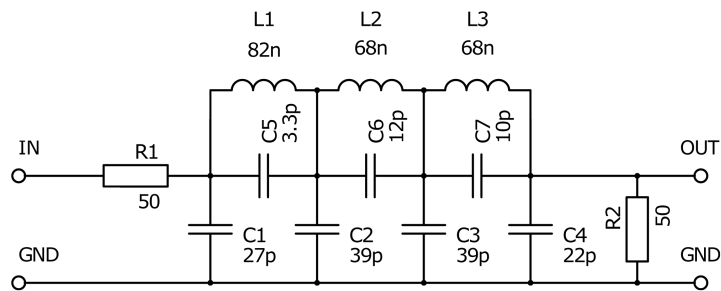
možno provést až těsně před výstupním konektorem, poté by však musel být filtr na výstupu obvodu symetrický, což by zvýšilo jeho obvodovou složitost. Proto je mezi výstup obvodu syntézy a filtr zařazen transformátor MABAES0060 fy Tyco Electronics[10], který provede onu nesymetrizaci, avšak také znemožní přenos nižších kmitočtů (<300 kHz). Dále by bylo možné signál znesymetrizovat připojením jednoho výstupního vodiče přes rezistor ke kladnému napájení AVDD (+1,8 V). Tím by ale došlo ke snížení výstupního napětí.

Pro návrh výstupního filtru jsem využil volně dostupný SW s názvem Elsie[11], který kromě návrhu filtrů umožňuje i doladění filtru změnou hodnot součástek, vykreslení jejich charakteristik a dalších funkcí.

Zapojení filtru s hodnotami součástek můžeme vidět na obr.2.5 a jeho modulovou a fázovou charakteristiku na obr.2.4. Celkové schéma signálových obvodů je v příloze C.



Obr. 2.4: Modulová a fázová charakteristika výstupního filtru



Obr. 2.5: Schéma výstupního filtru

## 2.5 FW mikroprocesoru

Programovým vybavením mikroprocesoru je FW (firmware). Náplní tohoto programu je obstarávat obsluhu I/O bran procesoru, všech jeho periférií a také provádět výpočet všech operací, které jsou od něj požadovány.

FW procesoru použitého v této aplikaci provádí naslouchání na branách UARTu, kterými přichází z PC instrukce pro ovládání zařízení, zpracování těchto instrukcí do instrukcí srozumitelných obvodu syntezátoru. Komunikace mezi zařízením a procesorem bude probíhat po sériovém rozhraní USB, které však musí být konvertováno na UART, který je přiveden na vstupní porty mikroprocesoru. Tuto konverzi obstarává obvod LT232. Instrukce z PC obsahují identifikaci nastavované hodnoty (registru) a onu hodnotu. Mikroprocesor si tyto hodnoty uloží do paměti RAM a upravené do tvaru srozumitelného obvodu syntezátoru je vyšle pomocí SPI ven. Dále si je uloží do paměti EEPROM, což umožní prvotní inicializaci obvodu AD9951 po zapnutí a nahrání posledních nastavených parametrů.

## 2.6 SW pro počítač

Ovládací program pro osobní počítač s OS Windows disponuje grafickým rozhraním, které přispívá snadnější obsluze zařízení. Z tohoto GUI (grafické uživatelské rozhraní - Graphic User Interface) lze ovládat veškeré registry v obvodu syntézy AD9951. V hlavním okně programu lze přímo nastavit kmitočet nebo fázi výstupního signálu. Ve druhém programovém okně lze nastavit veškeré hodnoty všech registrů obvodu syntezátoru, na kterých závisí jeho funkčnost, jako je hodnota vstupní násobičky apod.

## 3 REALIZACE NAVRŽENÉHO SYNTÉZÁTORU

Zařízení je koncipováno jako jednodeskové s požadavkem na minimální rozměry. S ohledem na relativní jednoduchost výroby jsem volil většinu součástek v provedení pro povrchovou montáž. Tam kde nebyla vhodná alternativa v SMT provedení jsem zůstal u vývodových součástek, zejména u konektorů. Velikost 1206 pasivních součástek je určitým kompromisem mezi požadavkem na minimální rozměry a schopností osadit je ručně.

### 3.1 Napájecí zdroj

Napájení celého zařízení je provedeno pomocí čtyř zdrojů s lineárními stabilizátory. Jedná se o katalogová zapojení obvodů

- MC7805 pro výstupní napětí +5 V (obvod LT232 a trojbodové zesilovače ERA-3+)
- NCP1117DT33 s výstupním napětím +3,3 V (I/O porty LT232, PIC16F628A, I/O porty AD9951)
- LM317 s výstupním napětím +1,8 V pro digitální část obvodu AD9951
- LM317 s výstupním napětím +1,8 V pro analogovou část obvodu AD9951

Obvody LM317 jsem zvolil pro špatnou dostupnost lineárních stabilizátorů pro tato napětí. Proudové zatížení zdrojů je popsáno v předchozí kapitole 2.1.

### 3.2 Převodník USB/UART

Pro připojení zařízení k PC je použita USB sběrnice, kterou v zařízení syntezátoru představuje obvod LT232. Tento obvod je převodníkem mezi USB a RS232, příp. UART. V této aplikaci plně dostačuje sběrnice UART, kterou lze přímo propojit s mikroprocesorem. V zapojení obvodu LT232 je připojena i LED dioda pro indikaci probíhající komunikace na straně UART. Tato svítivá dioda je připojena k vodičům indikace obou směrů komunikace, tudíž indikuje jak vysílání, tak příjem dat. Obvod je napájen napětím 5 V pro část USB komunikace a 3,3 V pro část komunikace UART. Možnost rozděleného napájení je výhodná zejména tehdy, kdy je potřeba přímo propojit dvě zařízení či obvody a to bez nutnosti dalších převodníků úrovní.

### 3.3 Mikroprocesor

Mikroprocesor v tomto zařízení slouží k překladu dat přijatých převodníkem USB/UART z PC do formátu sériových dat obvodu syntezátoru AD9951. Pomocí tohoto da-

tového slova lze nastavit veškeré parametry obvodu DDS (viz kap. obvodu AD9951). Pro použití v této aplikaci jsem zvolil procesor PIC16F628A fy Microchip. Tento obyčejný mikroprocesor svým výkonem pro tuto jednoduchou činnost překlada plně dostačuje. Procesor běží na hodinovém kmitočtu 4 MHz z interního oscilátoru, jelikož přijímá asynchronní data a následně je vysílá ven, přičemž sám tvoří i hodinový signál pro datovou komunikaci s obvodem AD9951. Další periferie mikroprocesoru (A/D převodníky, časovače apod.) jsou v této aplikaci nevyužity.

### 3.4 Obvody syntézy

Hlavním prvkem sekce přímé číslicové syntézy je obvod AD9951 fy Analog Devices. Jeho nejvyšší vstupní kmitočet je 400 MHz, který je zároveň kmitočtem hodinovým. Pro dosažení takového kmitočtu bez nutnosti přivádět jej na vstupní piny obvodu je obvod vybaven násobičkou vstupního kmitočtu s nastavitelnou hodnotou 4x až 20x. V této aplikaci využívám násobení hodnotou 20x, díky čemu lze na vstup obvodu připojit signál o kmitočtu 20 MHz. Požadavek na vstupní kmitočet, externího normálu však byl 10 MHz, proto musí být mezi vstupním konektorem a obvodem syntézy zapojen násobič kmitočtu, který byl realizován jako násobič s bipolárním tranzistorem BC847, na jehož nelinearitě vznikají vyšší harmonické, z nichž je následně vyfiltrována pouze jediná, o kmitočtu 20 MHz. Tím je dosaženo potřebného vstupního kmitočtu pro obvod AD9951. Impedanční oddělení je provedeno pomocí tříbodového zesilovače ERA-3+, který disponuje zesílením přibližně 10 dB.

Následujícím prvkem je analogový prepínač typu ADG1219 fy Analog Devices[9], který slouží k prepínání hodinového signálu mezi externím vstupem, přesněji výstupem násobiče, a výstupem interního krystalového oscilátoru s kmitočtem 20 MHz. Prepínání se provádí zkratovací propojkou (jumperem), která při prepnutí na externí vstup zároveň vypíná interní oscilátor do stavu StandBy, čímž jednak sníží spotřebu, ale také omezení vyzařování a rušení z tohoto obvodu.

Poslední analogovou částí zařízení je výstupní filtr typu dolní propust vypočítaný na kmitočet 120 MHz, který zaručí vyfiltrování výstupního signálu, aby nebyl zkreslen vyčítáním hodnot z tabulky ROM. Za tímto filtrem tedy signál bude sinusový. Vybraný filtr je Cauerův filtr sedmého řádu. Jelikož výstup obvodu AD9951 je symetrický a výstupní filtr byl pro zjednodušení konstrukce zvolen jako nesymetrický, je nutné výstupní signál z obvodu AD9951 znesymetrizovat. Toho je dosaženo pomocí 50  $\Omega$  transformátorku v miniaturním provedení od společnosti Tyco Electronics. Na výstupu filtru je opět tříbodový zesilovač ERA-3+, který zajistí zesílení výstupního signálu na potřebnou úroveň a zároveň zajistí impedanční přizpůsobení.

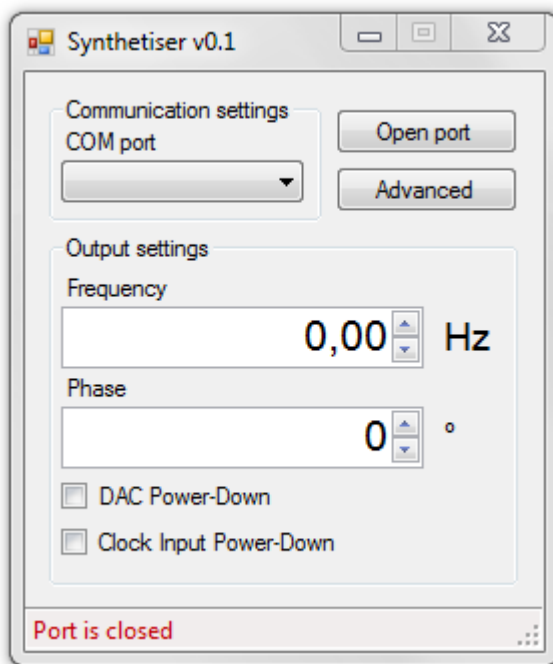
## 3.5 Ovládací SW

Ovládací SW v počítači je naprogramován v jazyce C# a umožňuje kompletní ovládní obvodu syntežátoru AD9951. Komunikace s HW probíhá skrz USB komunikaci, která je v zařízení pomocí obvodu FT232BL překládána na UART, který je veden do mikroprocesoru. Parametry UART komunikace jsou pevně nastaveny jak v SW, tak ve FW mikroprocesoru:

- rychlost = 9600 baudů
- parita = bez parity
- počet datových bitů = 8
- počet stop bitů = 1
- řízení (handshake) = bez řízení

**Hlavní okno programu** (zobrazeno na obr.3.1) umožňuje nastavit nejdůležitější parametry, jako je komunikační COM port, výstupní kmitočet syntežátoru, posuv jeho fáze, případně vypnutí výstupního DAC, resp. vypnutí vstupu hodinového signálu. Jednotlivé ovládací prvky jsou přímo překládány do slov určených přímo pro obvod syntežátoru, kromě dvou zatrhávacích položek, které jsou v procesoru dále upravovány.

Obvod AD9951 je možné ovládat až po otevření vybraného COM portu, který je tak daný i pro okno nastavení jednotlivých registrů obvodu (viz dále).



Obr. 3.1: Hlavní okno ovládacího programu

**Detailní nastavení** jednotlivých registrů obvodu AD9951 je v druhém okně ovládacího SW (obr.3.2). V tomto okně je možné nastavit jednotlivé parametry všech ovládacích registrů. Veškeré nastavené parametry jsou však odeslány až po potvrzení. Ne všechny nastavitelné parametry tohoto obvodu jsou potřebné pro tuto aplikaci, avšak je možné je proto využít v jiných aplikacích. Ať už by se jednalo přímo o toto zapojení nebo o jiné zapojení s možností ovládaní prostřednictvím tohoto SW.

Obr. 3.2: Detailní nastavení obvodu AD9951 v ovládacím SW

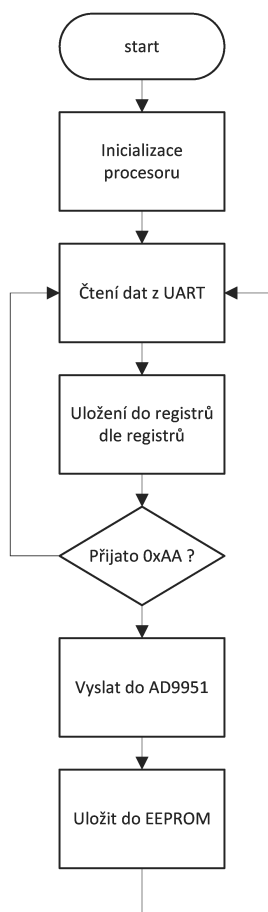
### 3.6 FW mikroprocesoru

Ovládací program mikroprocesoru je psán v jazyce C. Celý program je prakticky jen překladač dat mezi komunikací UART na sériovou komunikaci srozumitelnou obvodu AD9951. Komunikaci s obvodem AD9951 je možné provozovat buď dvou- nebo třídrátově. Pro zjednodušení zapojení byla zvolena dvoudrátová komunikace, nebude



totiž vyžadována komunikace ze syntezátoru do mikroprocesoru. Komunikace tedy probíhá primárně po dvou vodičích, jedním je datový vodič SDAT a druhým je hodinový CLK. Potvrzení o zaslání správných dat je provedeno vodičem IOUPDATE, po kterém dojde k nahrání zaslanych dat ze vstupních posuvných registrů obvodu do jeho pracovních registrů. Ihned po zapnutí provede mikroprocesor reset obvodu AD9951, čímž jej uvede do základního stavu, aby se předešlo nežádoucím čtením z datových vodičů, které probíhá ihned po přivedení napájecího napětí k obvodu AD9951.

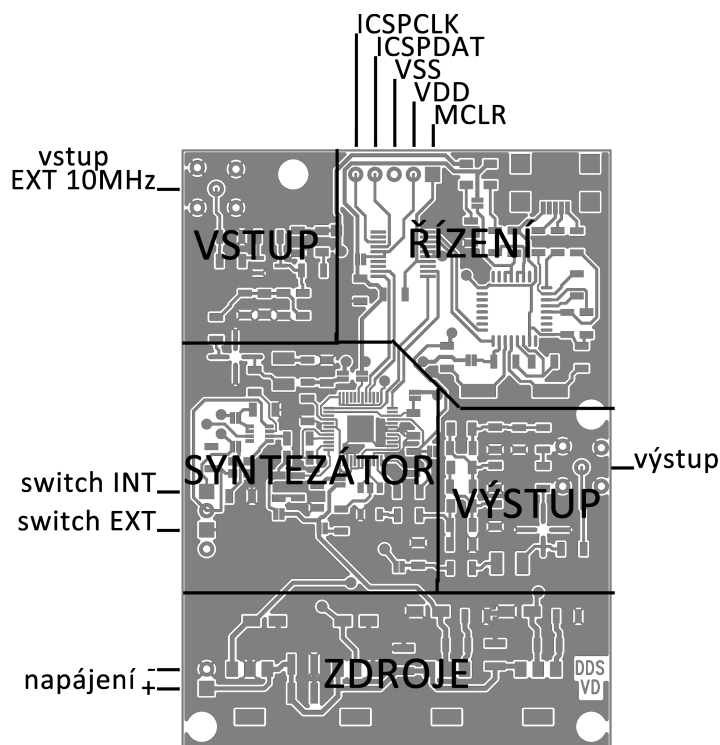
Vývojový diagram FW je na obr.3.3. Program vyčkává na zaslání datového slova z PC, dále dle prvního (adresního) bajtu rozhodne o který registr se jedná a uloží zbylá data do příslušných registrů. Po příjmu potvrzovacího slova (0xAA) odešle tyto informace do syntezátoru a zároveň je uloží do své interní paměti EEPROM. Data z EEPROM jsou využívána po zapnutí zařízení pro prvotní inicializaci obvodu AD9951, který nemá vnitřní paměť, ve které by si mohl bez napájení pamatovat nastavené parametry. Po zapnutí dojde tedy k nahrání posledních nastavených dat.



Obr. 3.3: Vývojový diagram FW mikroprocesoru

### 3.7 Mechanické provedení

Zařízení je realizováno jako jednodeskové s motivem na obou stranách a součástkami na straně jedné. Jednotlivé hlavní části jsou logicky rozvrženy a umístěny na desce aby nebylo nutné vést propojovací cesty napříč celou DPS. Hlavní části tedy jsou: zdroj, digitální část (převodník USB/UART, mikroprocesor), vstupní část obvodu, syntezátor a výstupní část obvodu. Rozvržení součástek je na obr.3.4.



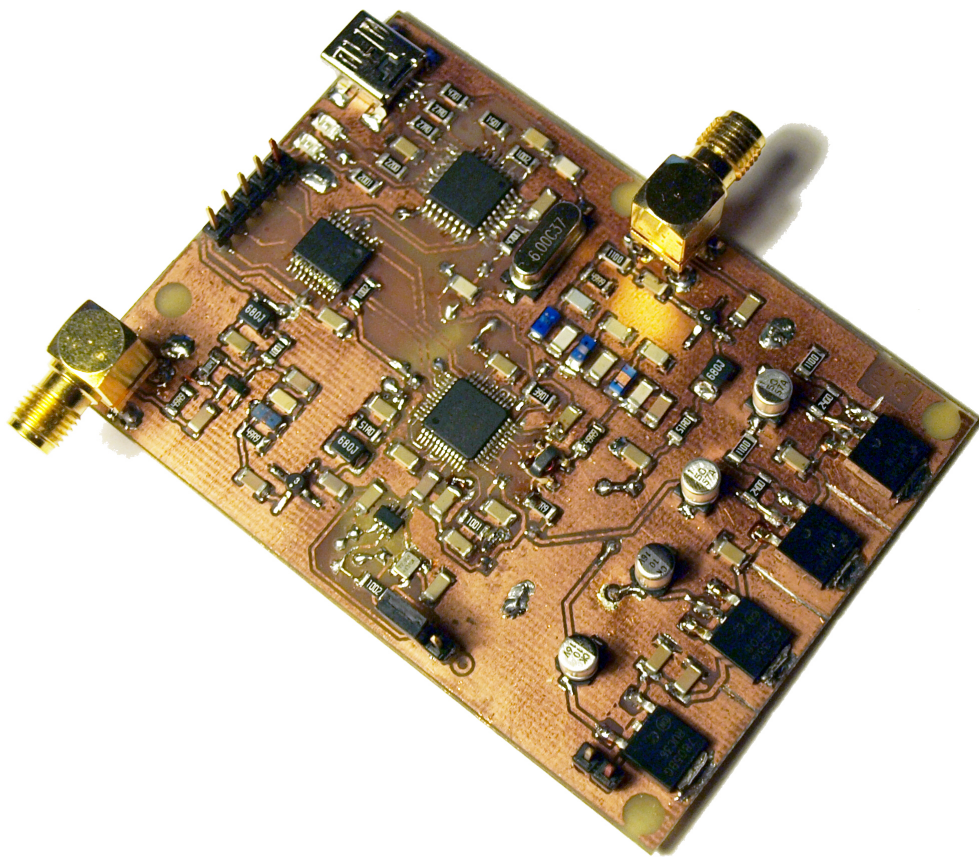
Obr. 3.4: Rozmístění hlavních částí obvodu

Vyrobená prototypová DPS je s ručními prokvy proletováním. Po osazení a oživení zdrojů byl osazen zbytek součástek a zkontrolováno jejich správné propojení. Po připojení zařízení USB kabelem k počítači a zapnutí došlo v PC k instalaci ovladačů virtuálního sériového (COM) portu, po kterém bylo zahájeno programování mikroprocesoru.

Motivy oboustranné desky plošných spojů jsou zobrazeny v příloze D. Fotografie osazené desky ze strany součástek je na obr.3.5.

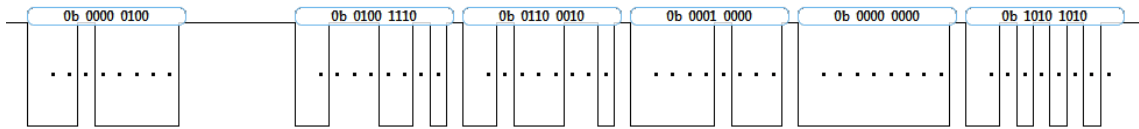
### 3.8 Oživení a programování

Po osazení, oživení a zkontrolování napětí na všech integrovaných obvodech jsem se mohl pustit do programování mikroprocesoru. Program mikroprocesoru je psán



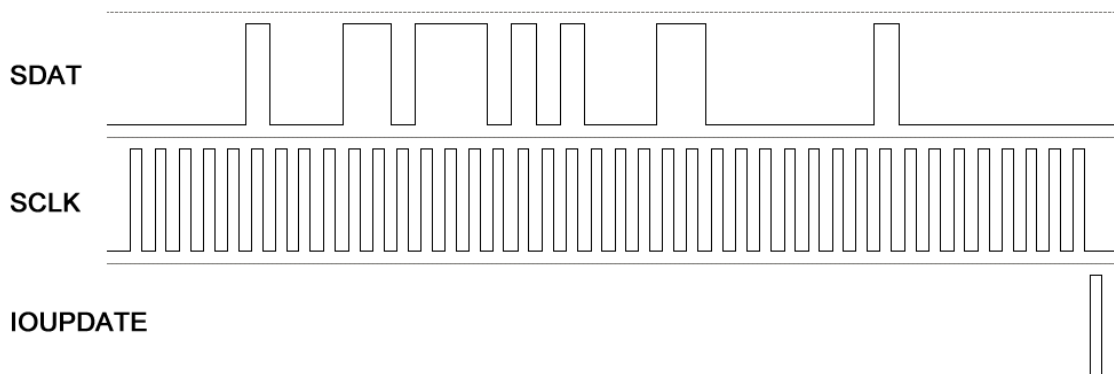
Obr. 3.5: Fotografie osazené prototypové desky

v jazyce C v programovacím prostředí MPLAB IDE. Program je pro svou jednoduchost napsán pouze v jednom souboru zdrojového kódu. Pro správnou funkčnost této aplikace, čili generování výstupního kmitočtu, postačuje nastavit činitel násobení vstupního kmitočtu fázovým závěsem a požadovaný výstupní kmitočet obvodu. Procesoru je programován pomocí rozhraní ICSP, které umožňuje změnu FW procesoru přímo v obvodu. K tomu slouží pětipinová kolíková lišta umístěná u procesoru. Pořadí pinů je následující: 1-MCLR, 2- $V_{DD}$ , 3- $V_{SS}$ , 4-ICSPDAT a 5-ICSPCLK. Prvním krokem programování bylo zprovoznit UART komunikaci mezi PC a mikroprocesorem. V SW v počítači i ve FW v mikroprocesoru jsou parametry komunikace nastaveny pevně, bez možnosti změny. Příklad průběhu UART komunikace při zápisu ladícího slova z PC do procesoru je zobrazen na obr.3.6. První bajt představuje adresu registru FTW (0x04 Frequency tuning word - Ladící slovo kmitočtu), následující čtyři bajty obsahují slovo představující zadaný kmitočet, přepočítaný dle 2.1 a poslední bajt je potvrzovací (0xAA). Přijetím 0xAA mikroprocesor ví, že všechna data byla pro tuto chvíli odeslána a může začít s jejich zpracováním.



Obr. 3.6: Časový průběh UART komunikace mezi PC a mikroprocesorem

Dalším krokem programování bylo zprovoznění sériové komunikace s obvodem AD9951. Zde se jedná o řízení pomocí čtyř vodičů. První vodič vede resetovací signál, který je využit pouze chvíli po zapnutí a který provede nastavení obvodu AD9951 do výchozího stavu. Druhý z vodičů vede datový signál, třetí vodič vede hodinový signál a čtvrtý slouží k potvrzení zaslaných dat. Syntezátor reaguje vždy na náběžnou hranu hodinového signálu. Další podmínkou zasílání sériových dat do obvodu AD9951 je začít datový přenos nejdůležitějším datovým bitem (MSB - Most Significant Bit). Mikroprocesor proto musí z UARTu přijatá data vysílat v opačném pořadí. Pro příklad je na obr.3.7 zobrazena komunikace do obvodu AD9951 se stejnými daty jako příklad komunikace UART z obr.3.6.



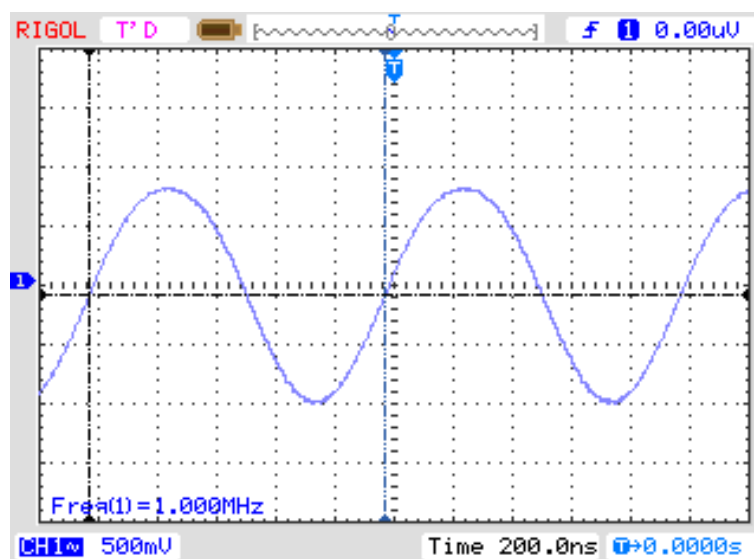
Obr. 3.7: Časový průběh UART komunikace mezi procesorem a AD9951

## 4 EXPERIMENTÁLNÍ MĚŘENÍ OBVODU

Po zdárném zprovoznění zařízení jsem dospěl k měření parametrů obvodu. Jako hlavní sledovaný parametr se jeví SFDR (Spurious free dynamic range)<sup>1</sup>, který je změřen na několika výstupních kmitočtech, čistota spektra výstupního signálu a dále fázový šum. Změřené parametry výstupních signálů pro jednotlivé kmitočty jsou zobrazeny v následujících odstavcích. K zobrazení časových průběhů byl použit osciloskop RIGOL 1102E a k zobrazení spektra byl použit analyzátor Rohde&Schwarz ETC. Všechny parametry byly měřeny při užití vnitřního oscilátoru a využití interního násobiče kmitočtu na hodinovém vstupu obvodu AD9951. Veškerá měření byla provedena v domácích podmínkách se zapůjčenými měřicími přístroji.

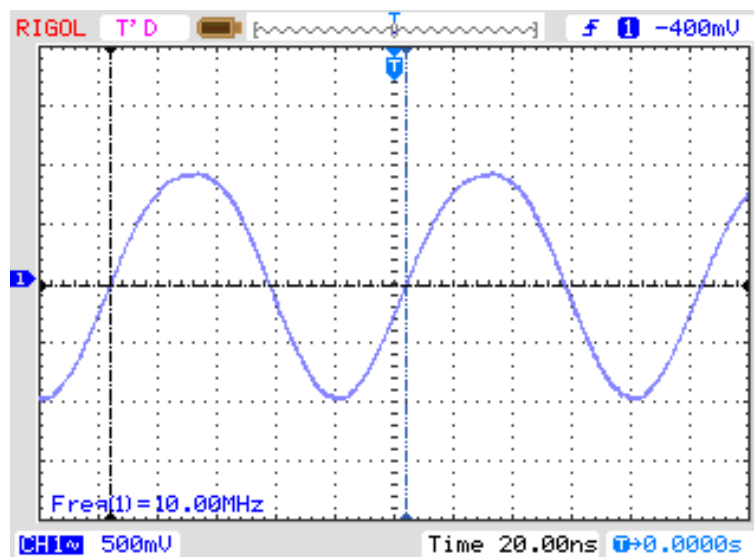
### 4.1 Zobrazení časových průběhů

Na následujících obrázcích (obr.4.1, obr.4.2, obr.4.3) můžeme vidět výstupní signály o kmitočtech 1 MHz, 10 MHz a 100 MHz.

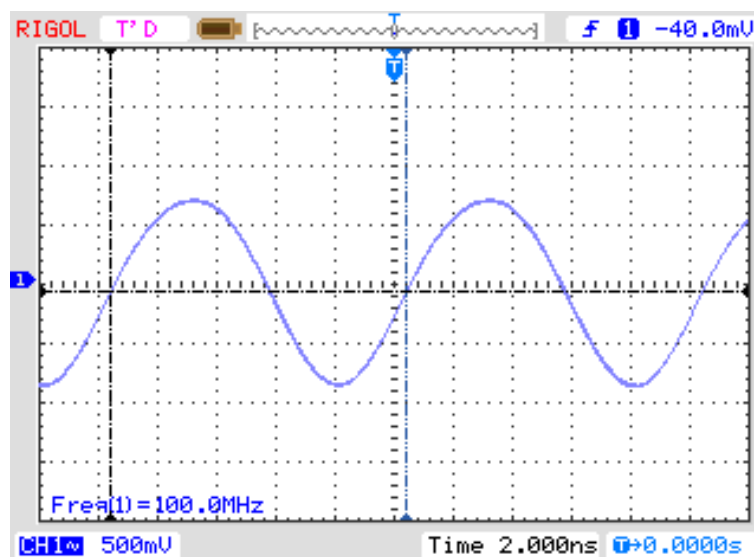


Obr. 4.1: Časový průběh výstupního signálu o kmitočtu 1 MHz.

<sup>1</sup>SFDR značí útlum mezi první harmonickou a druhou nejvyšší harmonickou



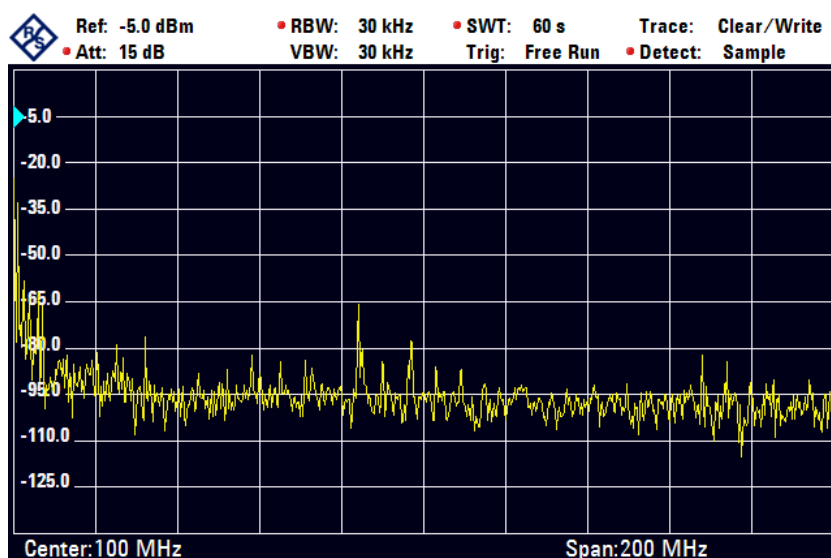
Obr. 4.2: Časový průběh výstupního signálu o kmitočtu 10 MHz.



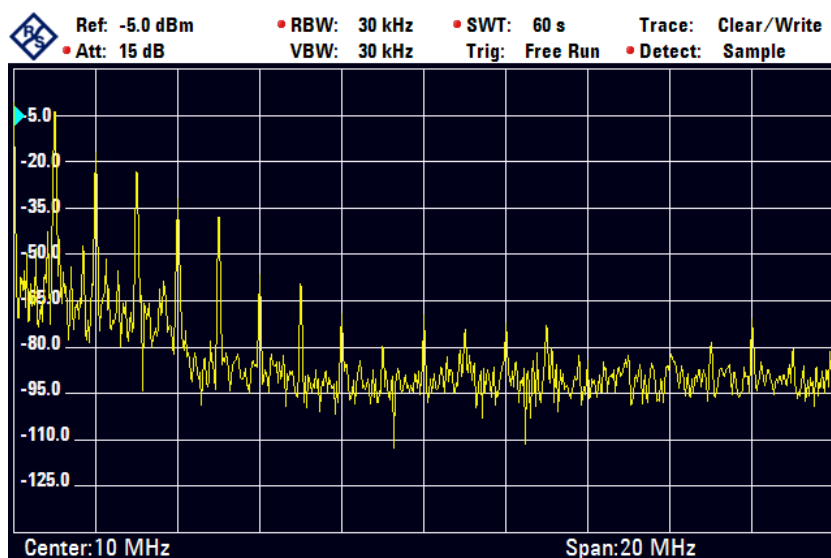
Obr. 4.3: Časový průběh výstupního signálu o kmitočtu 100 MHz.

## 4.2 Zobrazení kmitočtových spekter

Dále se podíváme na spektra výstupních signálů při několika nastavených kmitočtech. Prvním je nízký kmitočet 1 MHz zobrazený v širokém pásmu 200 MHz (obr.4.4). Můžeme pozorovat množství vyšších harmonických kmitočtů, pak parazitní složky na kmitočtu cca 85 MHz a jejich druhé harmonické kolem kmitočtu 170 MHz. Při bližším zobrazení (pásmo 20 MHz) na obr.4.5, můžeme pozorovat množství vyšších harmonických, které není ovlivněno výstupním filtrem, který je laděn až nad kmitočet 120 MHz.

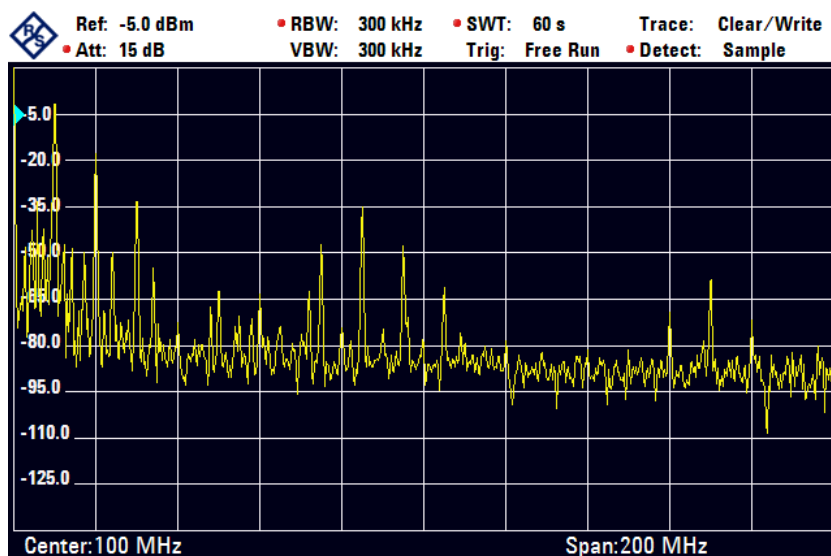


Obr. 4.4: Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 200 MHz.

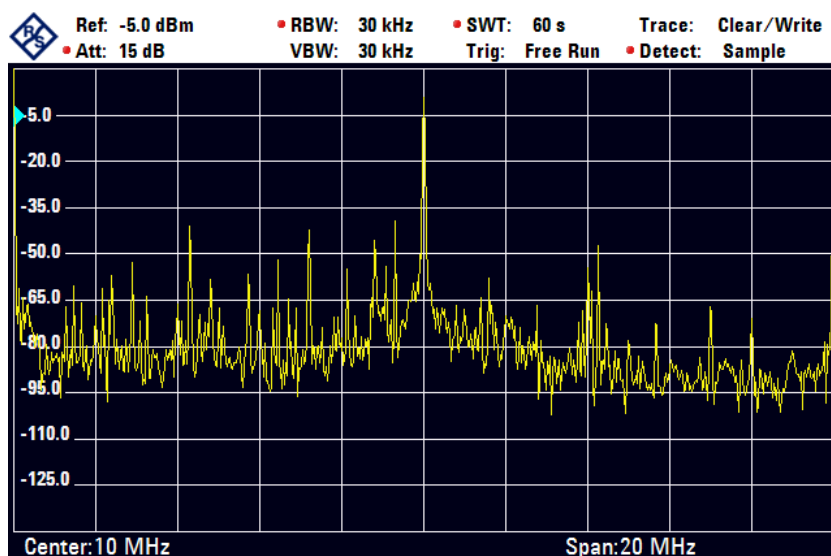


Obr. 4.5: Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 20 MHz.

Další obrázek (obr.4.6) ukazuje spektrum výstupního signálu na kmitočtu 10 MHz v širokém pásmu 200 MHz. Můžeme pozorovat poněkud zarušený kmitočet 10 MHz a tři jeho vyšší harmonické kmitočty a další parazitní prvky, včetně parazitní složky na kmitočtu 85 MHz. V užším pásmu 20 MHz (obr.4.5) můžeme sledovat kromě kmitočtu 10 MHz i další parazitní složky neznámého původu.



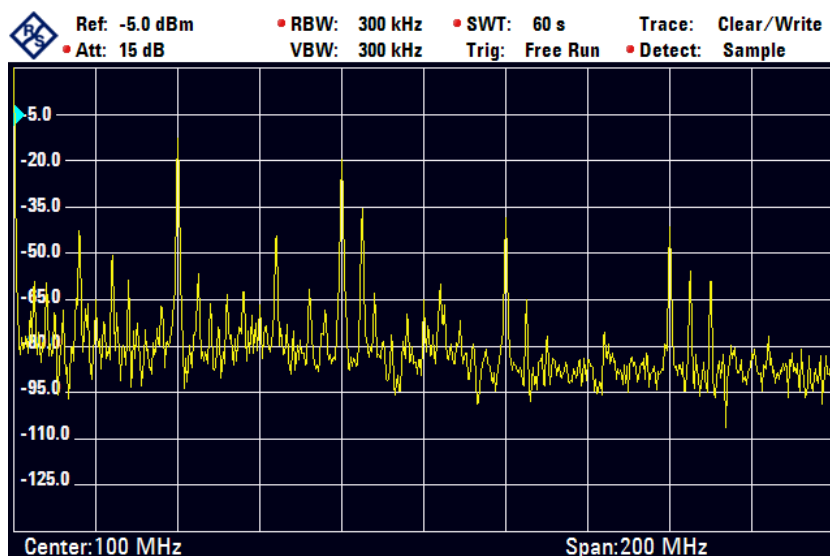
Obr. 4.6: Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 200 MHz.



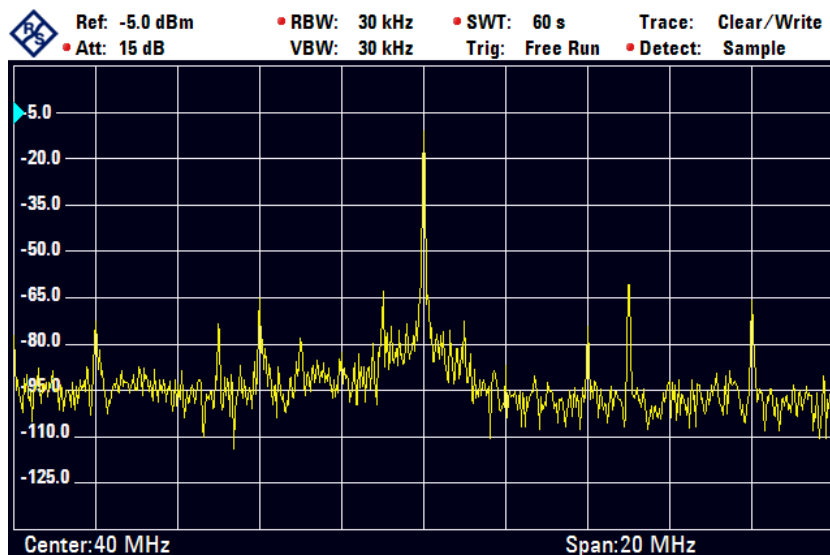
Obr. 4.7: Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 20 MHz.



Dalším sledovaným kmitočtem bylo 40 MHz, který můžeme vidět na obr.4.8 a obr.4.9. Můžeme pozorovat základní signál a čtyři harmonické kmitočty, spolu s parazitním signálem na kmitočtu 85 MHz i s jeho druhou harmonickou na 170 MHz. V užším pásmu můžeme vidět hlavní kmitočet a několik parazitních signálů.

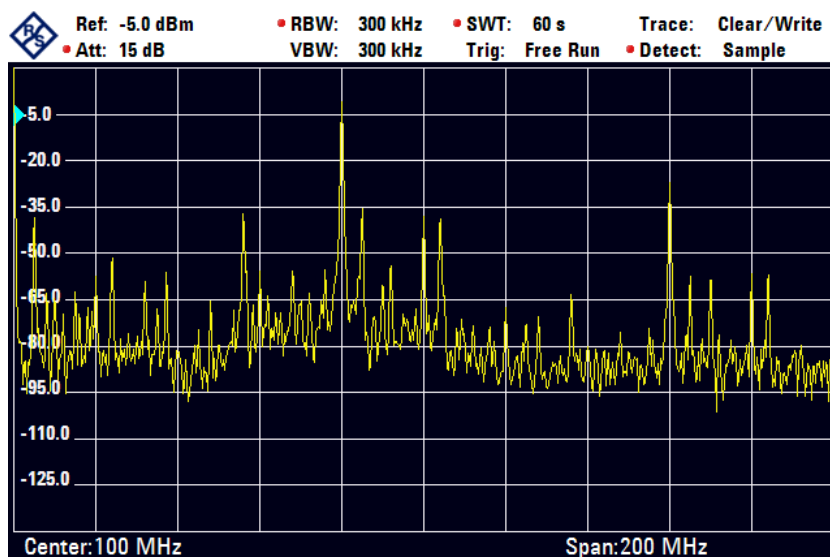


Obr. 4.8: Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 200 MHz.

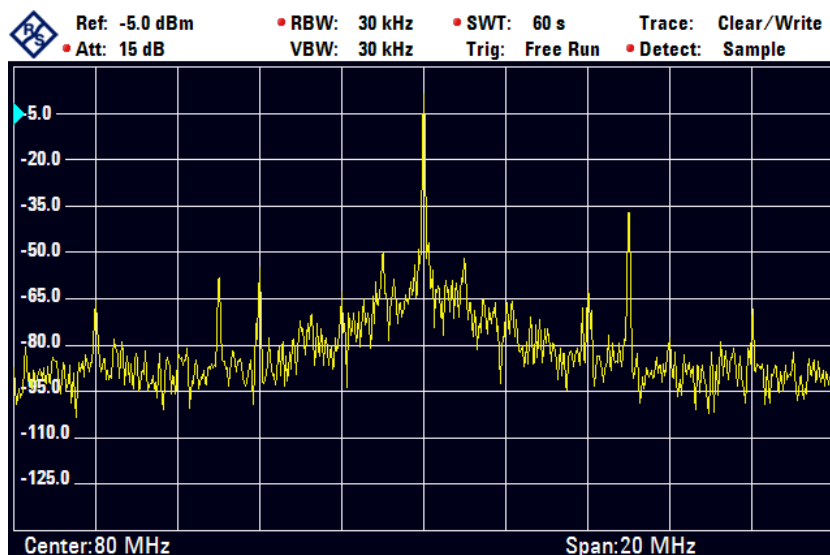


Obr. 4.9: Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 20 MHz.

Následujícím kmitočtem bylo 80 MHz, které jsou zobrazeny na obr.4.10 a obr.4.11. Na spektrogramech můžeme pozorovat kromě žádoucího kmitočtu 80 MHz i jeho druhou harmonickou, ale také parazitní složku na kmitočtu 85 MHz a také jeho druhou harmonickou.

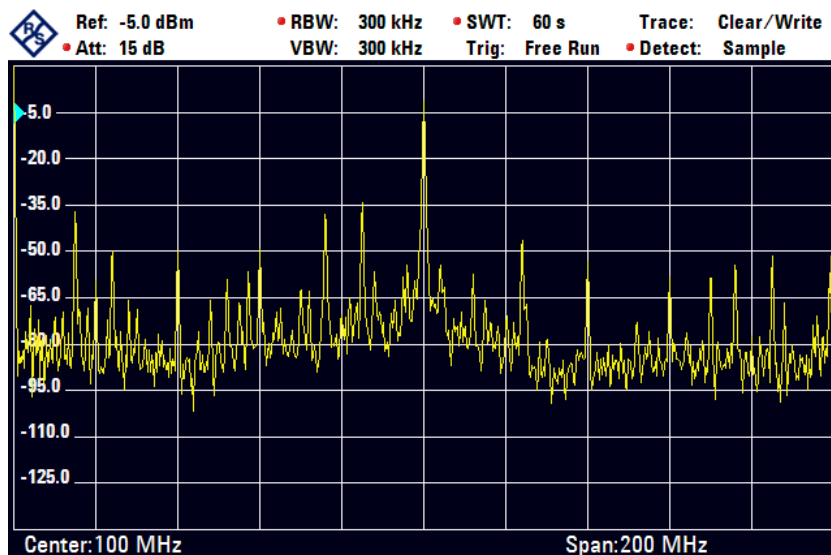


Obr. 4.10: Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 200 MHz.

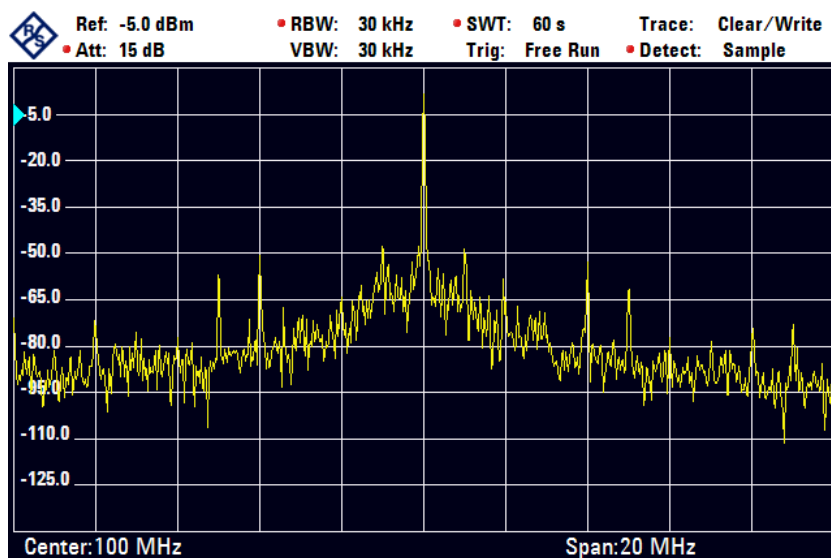


Obr. 4.11: Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 20 MHz.

Předposledním sledovaným kmitočtem bylo 100 MHz, jehož spektrogramy můžeme vidět na obr.4.12 a obr.4.13. Opět můžeme pozorovat kromě žádoucí složky i několik složek parazitních.

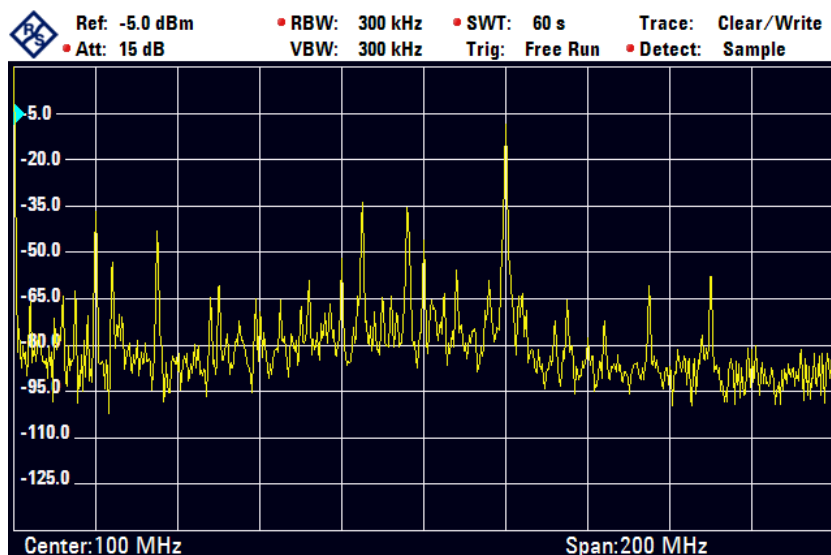


Obr. 4.12: Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 200 MHz.

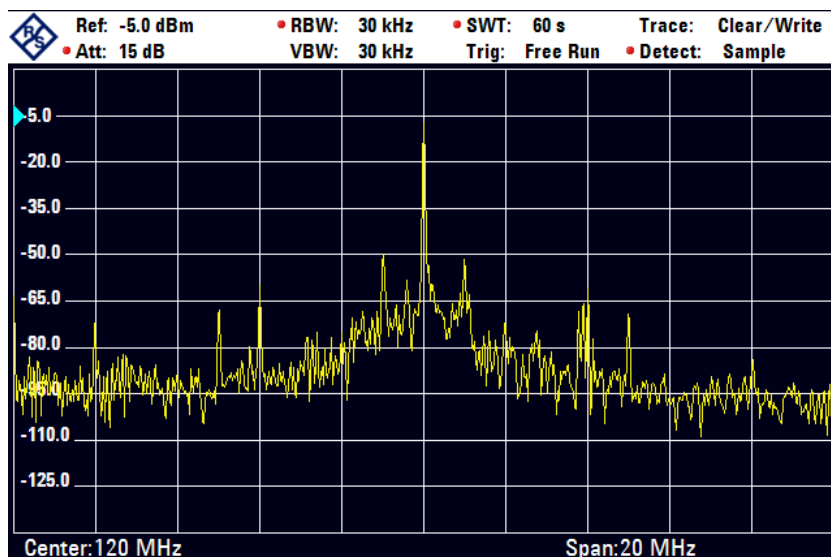


Obr. 4.13: Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 20 MHz.

Posledním kmitočtem byl nejvyšší možný, a to 120 MHz. Zobrazení jeho spektra můžeme vidět na obr.4.14 a obr.4.15. V širokém pásmu můžeme pozorovat kromě chtěného signálu 120 MHz i několik nechtěných složek, a to na kmitočtech 85 MHz (i jeho druhou harmonickou na 170 MHz) a 95 MHz.

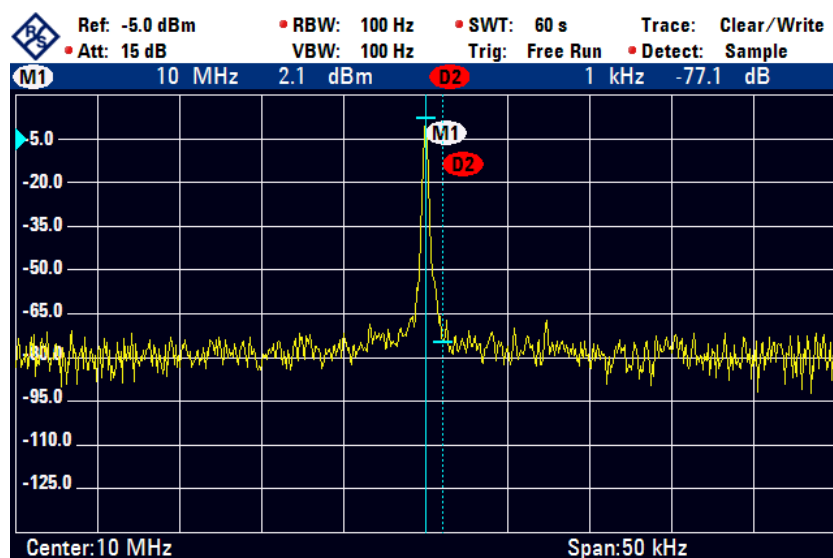


Obr. 4.14: Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 200 MHz.

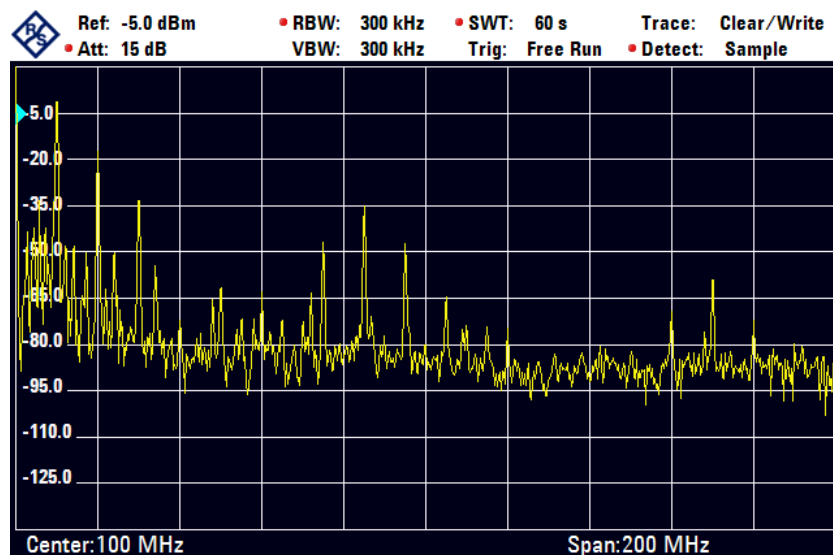


Obr. 4.15: Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 20 MHz.

Bylo také provedeno měření s generátorem připojeným k externímu vstupu. Použitý generátor však nesplňoval potřebné kvalitativní parametry, zejména stabilitu kmitočtu. Měření bylo provedeno pouze pro ověření činnosti externího vstupu. Pro ukázkou je zobrazeno spektrum signálu o kmitočtu 10 MHz v pásmu 50 kHz (obr.4.16) a 200 MHz (obr.4.17).



Obr. 4.16: Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 50 kHz s markery, při použití externího zdroje hodinového signálu.

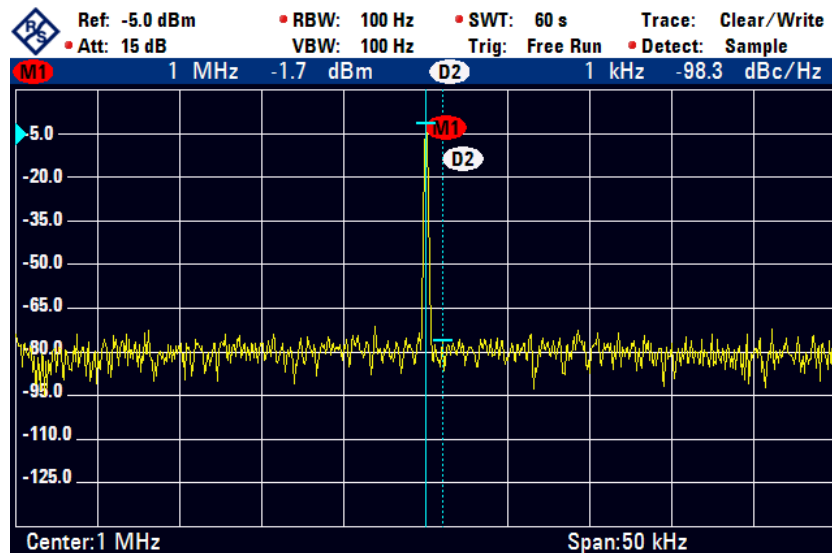


Obr. 4.17: Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 200 MHz, při použití externího zdroje hodinového signálu.

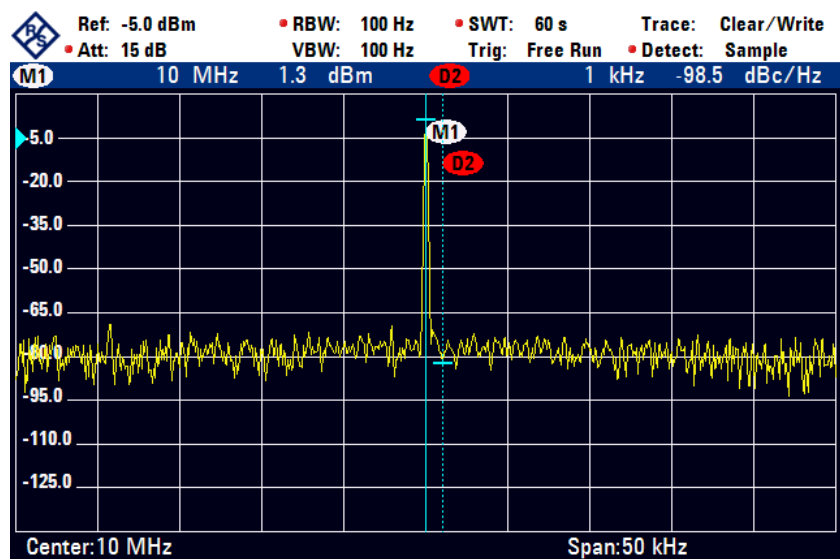
## 4.3 Fázový šum

K měření fázového šumu byl přímo využit analyzátor Rohde&Schwarz ETC, který umožňuje přímé zobrazení hodnoty fázového šumu. Spektrogramy se zobrazenými markery i hodnotami jsou zobrazeny na obr.4.18 až obr.4.23. Fázový šum byl měřen při offsetu 1 kHz od základního kmitočtu při zobrazení šířky pásma  $\pm 50$  kHz. Hodnoty fázového šumu byly naměřeny následující:

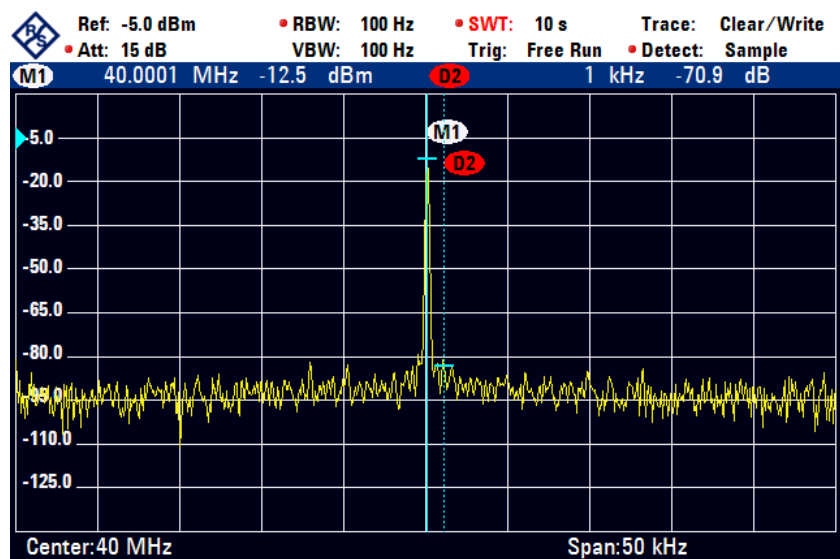
- -98,3 dBc/Hz @ 1 MHz
- -98,5 dBc/Hz @ 10 MHz
- -70,9 dBc/Hz @ 40 MHz
- -70,9 dBc/Hz @ 80 MHz
- -74,8 dBc/Hz @ 100 MHz
- -67,4 dBc/Hz @ 120 MHz



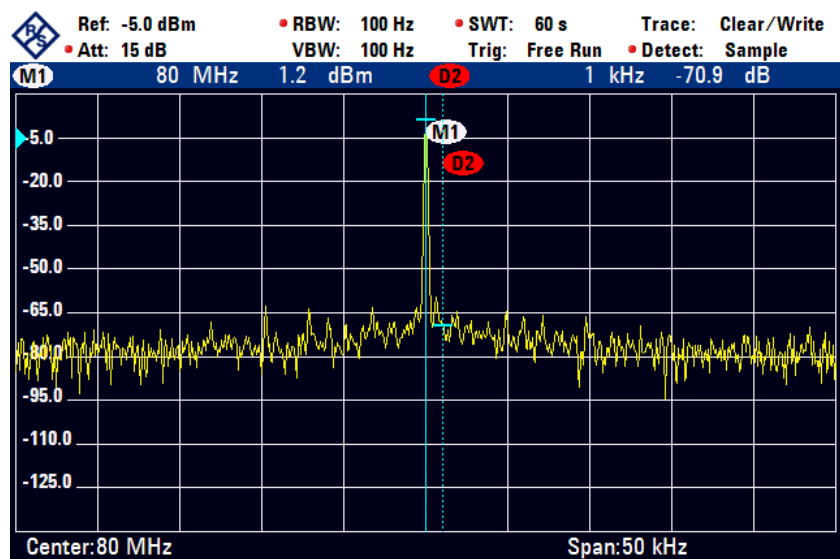
Obr. 4.18: Spektrum výstupního signálu o kmitočtu 1 MHz v pásmu 50 kHz s markery.



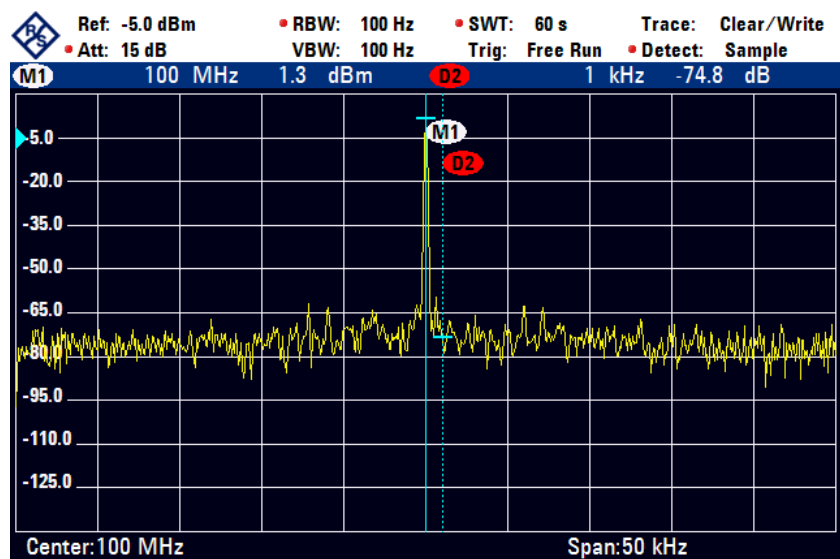
Obr. 4.19: Spektrum výstupního signálu o kmitočtu 10 MHz v pásmu 50 kHz s markery.



Obr. 4.20: Spektrum výstupního signálu o kmitočtu 40 MHz v pásmu 50 kHz s markery.

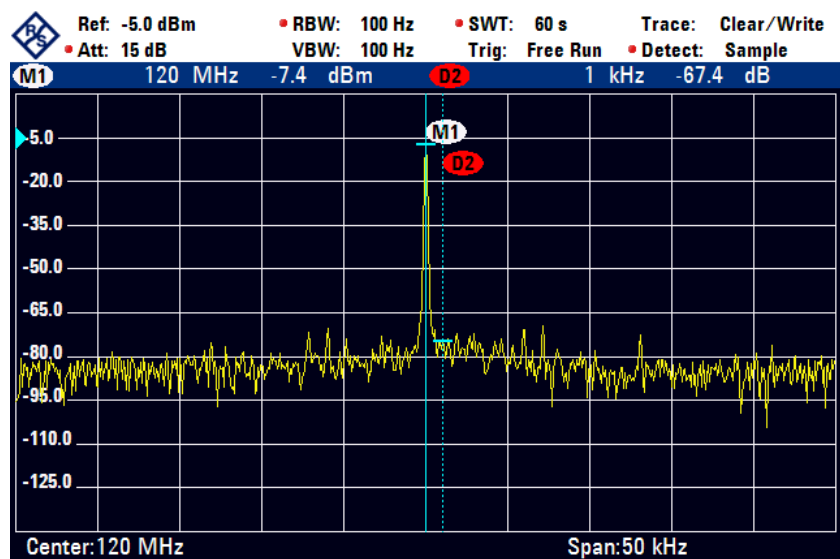


Obr. 4.21: Spektrum výstupního signálu o kmitočtu 80 MHz v pásmu 50 kHz s markery.



Obr. 4.22: Spektrum výstupního signálu o kmitočtu 100 MHz v pásmu 50 kHz s markery.





Obr. 4.23: Spektrum výstupního signálu o kmitočtu 120 MHz v pásmu 50 kHz s markery.

## 5 ZÁVĚR

Frekvenční syntéza je v poslední době jedna z nejpoužívanějších metod generování harmonických signálů, ať už se jedná o syntezátory na principu fázového závěsu nebo přímé číslicové syntézy. Velkým pokrokem v tomto směru byl značný vývoj ve výrobě a miniaturizaci integrovaných obvodů, čímž je dosaženo značné miniaturizace obvodů syntezátorů oproti klasickým oscilátorům. Taktéž se syntezátory vyznačují velkým rozsahem přeladění generovaného výstupního signálu. Musí však být jejich výstup dobře filtrován, jelikož především přímé číslicové syntezátory mají výstupní signál, díky vyčítání konečného množství hodnot z paměti ROM, značně zkreslený.

Další částí této práce je návrh syntezátoru na principu přímé číslicové syntézy. Pro tuto činnost byl vybrán obvod AD9951 fy Analog Devices. Jako zdroj vstupního (referenčního) hodinového signálu pro tento obvod slouží krystalový oscilátor o kmitočtu 20 MHz nebo externí kmitočtový normál 10 MHz připojený konektorem SMA. Aby mohl syntezátor pracovat na svém maximálním hodinovém kmitočtu 400 MHz, je nutné využít vnitřního fázového závěsu jako násobičku vstupního kmitočtu s hodnotou 20. I přes využití vnitřní PLL je nutné do cesty externího vstupu zařadit tranzistorový zdvojovač kmitočtu, který zajistí na vstup obvodu AD9951 požadovaný kmitočet 20 MHz. Výstupní signál zdvojovače je oddělen a zesílen pomocí trojbodového vf zesilovače ERA-3+. Přepínání mezi zdroji referenčního hodinového signálu je zajištěno polovodičovým přepínačem ADG1219 fy Analog Devices.

Výstupní signál z obvodu AD9951 je pomocí miniaturního transformátoru s převodem 1:1 a impedancí 50  $\Omega$  nesymetrizován, tím se zjednodušilo zapojení následného filtru. Filtr typu dolní propust má Cauerovu aproximaci sedmého řádu s mezním kmitočtem 120 MHz se zvlněním 0,1 dB v pásmu propustnosti a útlumem -65 dB v pásmu nepropustnosti. Před výstupním konektorem SMA je umístěn druhý trojbodový zesilovač ERA-3+, který zesiluje vyfiltrovaný výstupní signál.

Obvod AD9951 je řízen mikroprocesorem PIC16F628A fy Microchip pomocí sériové SPI dvoudrátové datové komunikační sběrnice. Zařízení s PC komunikuje pomocí USB, která je však na desce konvertována převodníkem FT232BL fy FTDI na UART, který je veden do mikroprocesoru. Mikroprocesor zajišťuje překlad a úpravu dat z PC do syntezátoru a také jejich uložení do paměti EEPROM, které je využito při zapnutí zařízení. V tu chvíli mikroprocesor načte z paměti uložená data poslední konfigurace a vyšle je do obvodu AD9951, čímž je provedena počáteční inicializace.

Experimentálně změřené výsledky, zobrazené v kapitole 4, ukazují časové průběhy a kmitočtová spektra generovaných signálů. Můžeme pozorovat například potlačení druhé harmonické o cca 30 dB a třetí o dalších zhruba 10 dB. Ve spektrech však jsou vidět i nestálé parazitní složky, zejména při širokopásmovém pohledu. Tyto složky se s každým měřením měnily, takže pravděpodobně nebyly generovány

syntezátorem. Fázový šum výstupního signálu se ve vzdálenosti 1 kHz od nosné pohyboval v rozsahu od cca -70 dBc/Hz do -98 dBc/Hz, což je horší než ve výrobní specifikaci obvodu syntezátoru. Pro ověření činnosti vstupu externího signálu byl použit méně kvalitní generátor hodinového kmitočtu, čemuž odpovídají i naměřené výsledky z předešlé kapitoly.

## LITERATURA

- [1] Cascading PLLs. *Cascading PLLs* [online]. [cit. 2014-12-13]. Dostupné z: <<http://www.oocities.org/fudinggepll/cascadingPLL.html>>
- [2] KASAL, M. *Frekvenční syntéza v komunikačních systémech - Experimentální družice*. Edice Habilitační a inaugurační spisy, sv. 169. Brno: Nakladatelství VUTIUM, 2005.
- [3] Katalogový list součástky : AD9951, výrobce : Analog Devices. [online]. [cit. 2015-05-19]. Dostupné z URL: <[http://www.analog.com/static/imported-files/data\\_sheets/AD9951.pdf](http://www.analog.com/static/imported-files/data_sheets/AD9951.pdf)>.
- [4] Katalogový list součástky : LM317, výrobce : Fairchild. [online]. [cit. 2015-05-19]. Dostupné z URL: <<https://www.fairchildsemi.com/datasheets/LM/LM317M.pdf>>.
- [5] Katalogový list součástky : NCP1117DT33, výrobce : ON Semiconductor. [online]. [cit. 2015-05-19]. Dostupné z URL: <[http://www.onsemi.com/pub\\_link/Collateral/NCP1117-D.PDF](http://www.onsemi.com/pub_link/Collateral/NCP1117-D.PDF)>.
- [6] Katalogový list součástky : MC7805, výrobce : ON Semiconductor. [online]. [cit. 2015-05-19]. Dostupné z URL: <[http://www.onsemi.com/pub\\_link/Collateral/MC7800-D.PDF](http://www.onsemi.com/pub_link/Collateral/MC7800-D.PDF)>.
- [7] Katalogový list součástky : PIC16F628A, výrobce : Microchip. [online]. [cit. 2015-05-19]. Dostupné z URL: <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>>.
- [8] Katalogový list součástky : FT232, výrobce : FTDI. [online]. [cit. 2015-05-19]. Dostupné z URL: <[http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232BL\\_BQ.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232BL_BQ.pdf)>.
- [9] Katalogový list součástky : ADG1219, výrobce : Analog Devices. [online]. [cit. 2015-08-04]. Dostupné z URL: <<http://www.analog.com/media/en/technical-documentation/data-sheets/ADG1219.pdf>>.
- [10] Katalogový list součástky : MABAES0060, výrobce : Tyco Electronics. [online]. [cit. 2015-08-04]. Dostupné z URL: <[http://cdn.macom.com/datasheets/MABAES0060\\_V6.pdf](http://cdn.macom.com/datasheets/MABAES0060_V6.pdf)>.
- [11] Program pro výpočet filtrů a přičkových struktur. *Elsie*. [cit. 2014-12-10]. Dostupné z: <<http://tonnesoftware.com/elsie.html>>.

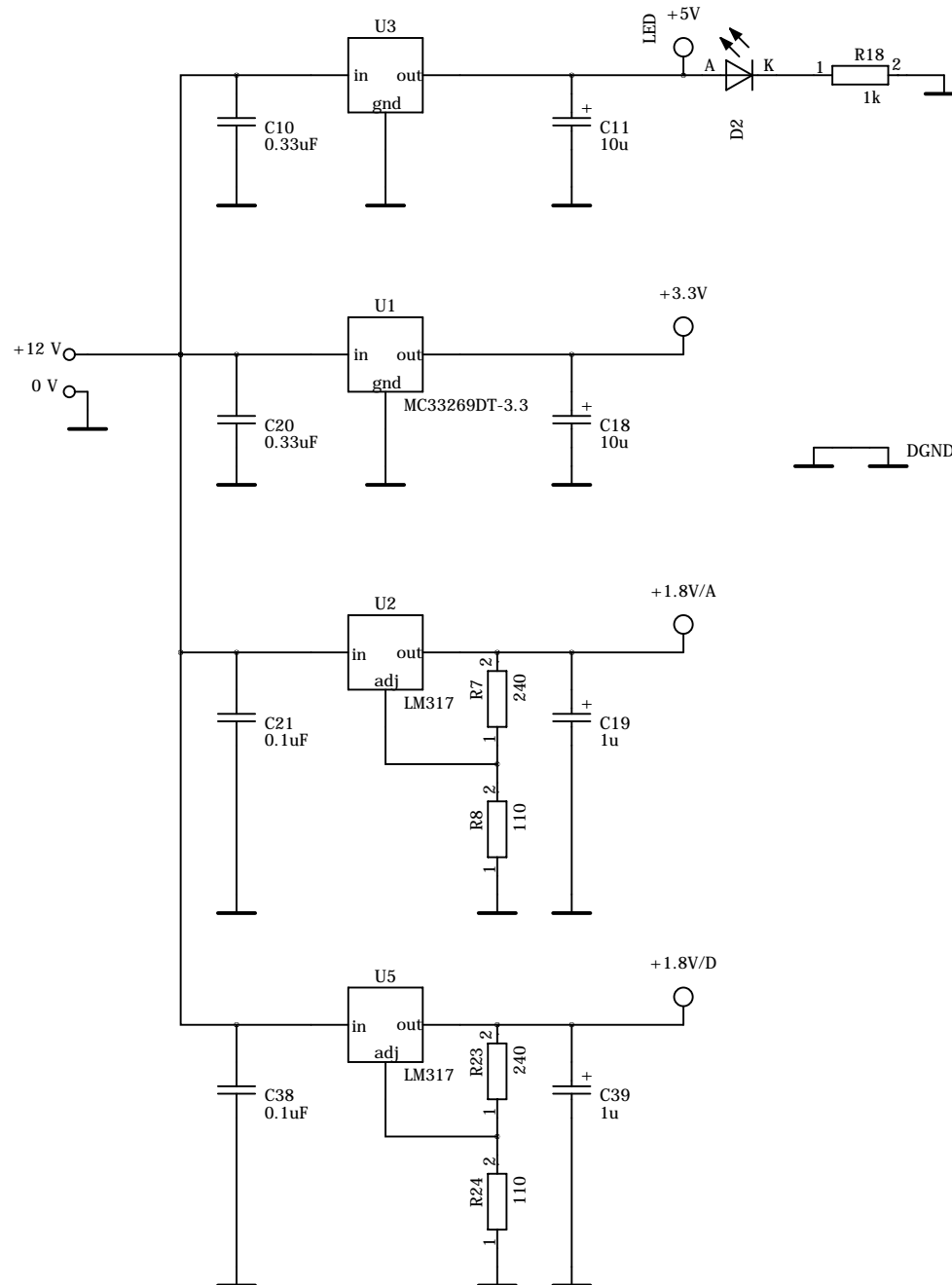
## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D	analogově-digitální převodník – Analog to Digital Converter
D/A	digitálně-analogový převodník – Digital to Analog Converter
DDS	přímá číslicová syntéza – Direct Digital Synthesis
DPS	deska plošných spojů
FW	firmware
GUI	grafické uživatelské rozhraní - Graphic User Interface
I/O	vstupně/výstupní – Input/Output
ICSP	sériové programování v obvodu – In-Circuit Serial Programming
IO	integrovaný obvod – Integrated Circuit
OS	operační systém – Operating System
PC	osobní počítač – Personal Computer
PLL	smyčka fázového závěsu – Phase-Locked loop
PWM	pulzně-šířková modulace – Pulse-Width Modulation
SFDR	spurious free dynamic range
SMD	součástky pro povrchovou montáž – Surface Mounted Device
SMT	technologie povrchové montáže – Surface Mount Technology
SW	programové vybavení – Software
UART	universální asynchronní přijímač vysílač – Univesral Asynchonus Receiver Transmitter
VCO	napětím řízený oscilátor – Voltage Controlled Oscilator
$\nu_f$	vysokofrekvenční
$f_{out}$	výstupní kmitočet

## SEZNAM PŘÍLOH

A Schéma zdroje	46
B Schéma řídicích obvodů	48
C Schéma obvodu syntezátoru	50
D Desky plošných spojů	52
E Zdrojový kód FW	54

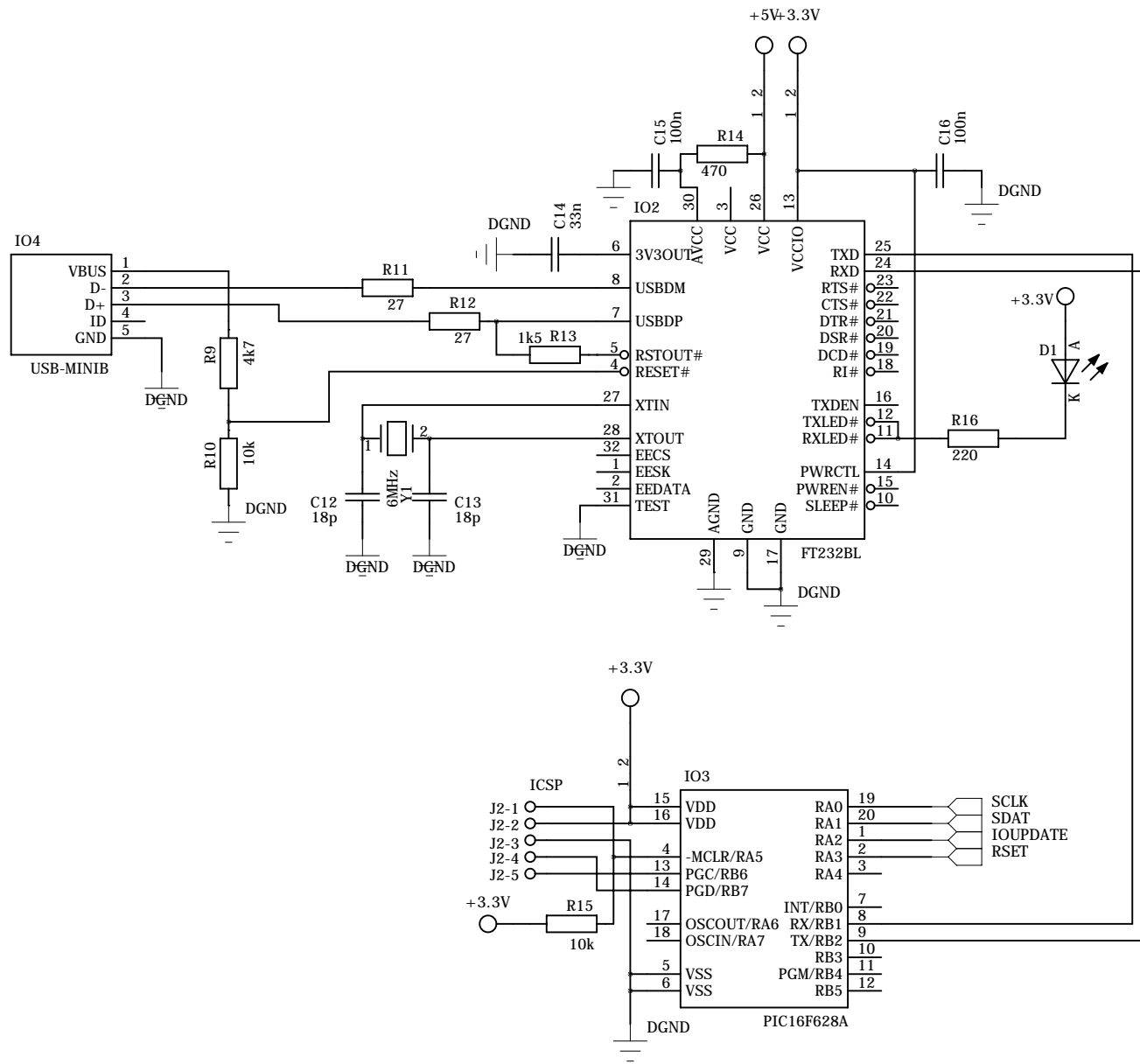
## A SCHÉMA ZDROJE



kreslil: Bc. Vojtech Dluhy	FEKT VUT v Brne	
projekt: <b>Diplomova prace</b>		
název: <b>ZDROJ</b>		
datum poslední změny: <b>4.8.2015</b>	formát: <b>A4</b>	list: <b>1 / 3</b>

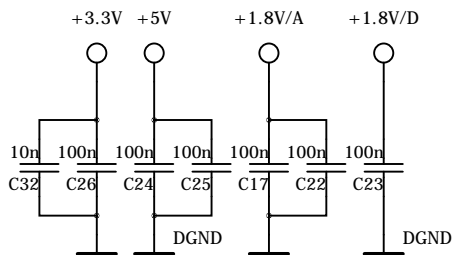
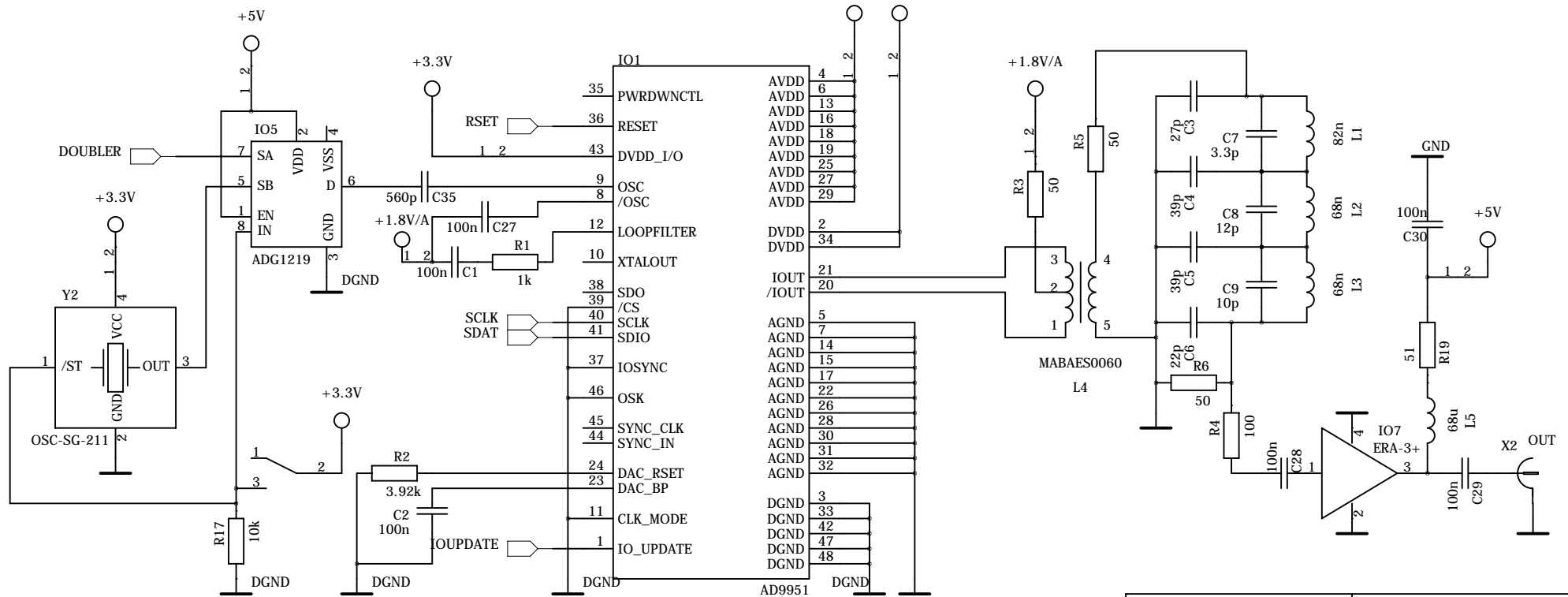
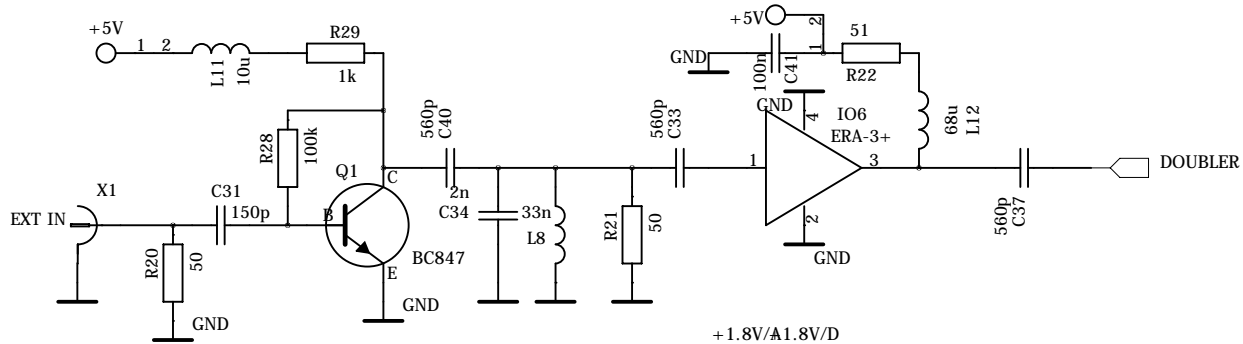


## B SCHÉMA ŘÍDÍCÍCH OBVODŮ



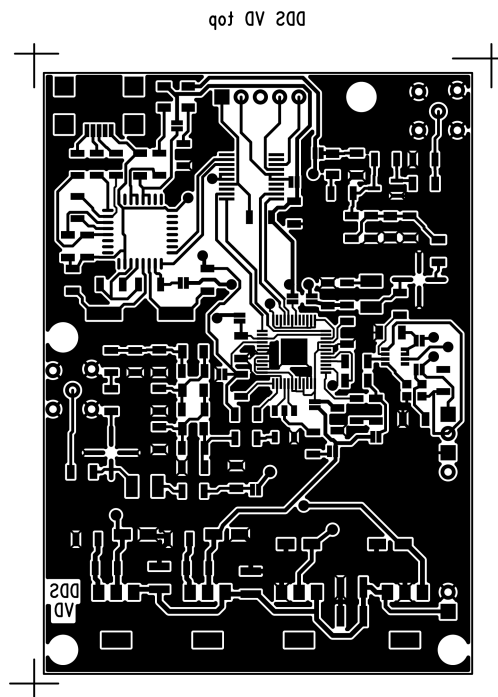
kreslil: Bc. Vojtech Dluhy	FEKT VUT v Brne	
projekt: <b>Diplomova prace</b>		
název: <b>RIZENI</b>		
datum poslední změny: <b>4.8.2015</b>	formát: <b>A4</b>	list: <b>2 / 3</b>

## C SCHÉMA OBVODU SYNTEZÁTORU

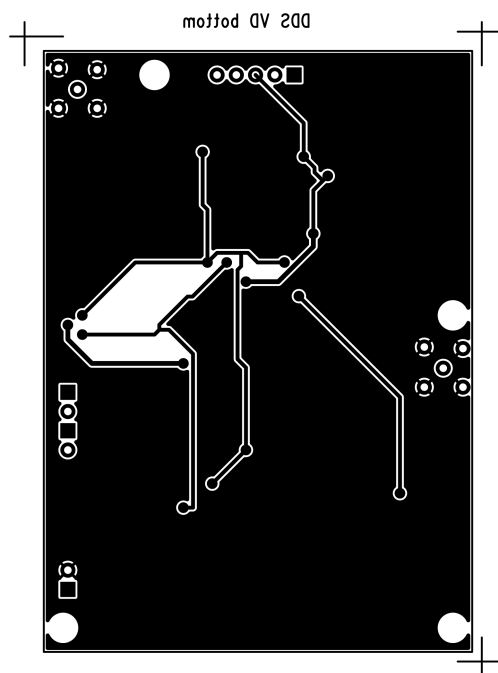


kreslil: Bc. Vojtech Dluhy	FEKT VUT v Brne	
projekt: Diplomova prace		
název: SYNTEZATOR		
datum poslední změny: 4.8.2015	formát: A4	list: 3 / 3

## D DESKY PLOŠNÝCH SPOJŮ



Obr. D.1: DPS - strana součástek (horní)



Obr. D.2: DPS - strana spojů (spodní)

# E ZDROJOVÝ KÓD FW

Soubor DP.c

```
1 #include <htc.h>
2 #include <stdio.h>
3
4 #define _XTAL_FREQ 4e6
5
6 //Definice nazvu
7 #define SDAT PORTAbits.RA1
8 #define SCLK PORTAbits.RA0
9 #define IOUPDATE PORTAbits.RA2
10 #define RESET PORTAbits.RA3
11
12 //Definice maker
13 #define BIT(x,n) (((x) >> (n)) & 1) //makro pristupu k jednotlivym bitum slova
14
15 //Definice globalnich promennych
16 volatile char prijate_slovo, zmena;
17 volatile char CFR11, CFR12, CFR13, CFR14, CFR21, CFR22, ASF1, ASF2, ARR, FTW1, FTW2, FTW3, FTW4,
18 POW1, POW2, POM;
19
20 //Predpisy funkci
21 void putch(unsigned char byte);
22 unsigned char getch();
23 void datawrite(char data);
24 void EEPROMwrite(char addr, char data);
25 char EEPROMread(char addr);
26 void write();
27
28 //Hlavni program
29 void main(){
30     CMCON = 0x07; //nastaveni pinu portu A jako digitalni
31     TRISA = 0x00; //nastaveni portu A jako vystupni
32     TRISB1 = 1; //nastaveni portu B1 jako vstupni --+--> pro spravnou funkci
33     UART
34     TRISB2 = 1; //nastaveni portu B2 jako vstupni --|
35     PORTA = 0x00; //inicializace portu A
36     PORTB = 0x00; //inicializace portu B
37     TXSTA = 0x24; //nastaveni registru UART pro vysilani
38     RCSTA = 0x90; //nastaveni registru UART pro prijem
39     SPBRG = 25; //nastaveni hodnoty delicky, kterou se ridi rychlost komunikace
40     UART
41
42     SCLK = 0; //nulovani vystupu SCLK (port A0)
43     SDAT = 0; //nulovani vystupu SDAT (port A1)
44     IOUPDATE = 0; //nulovani vystupu IOUPDATE (port A2)
45     RESET = 1; //nastaveni portu RESET (port A3)
46     __delay_ms(10);
47     RESET = 0; //nulovani portu RESET
48
49 //cteni ulozenych registru z pameti EEPROM
50 CFR11 = EEPROMread(0x00);
51 CFR12 = EEPROMread(0x01);
52 CFR13 = EEPROMread(0x02);
53 CFR14 = EEPROMread(0x03);
54 CFR21 = EEPROMread(0x04);
55 CFR22 = EEPROMread(0x05);
56 ASF1 = EEPROMread(0x06);
57 ASF2 = EEPROMread(0x07);
58 ARR = EEPROMread(0x08);
59 FTW1 = EEPROMread(0x09);
60 FTW2 = EEPROMread(0x0A);
61 FTW3 = EEPROMread(0x0B);
62 FTW4 = EEPROMread(0x0C);
63 POW1 = EEPROMread(0x0D);
64 POW2 = EEPROMread(0x0E);
65
66 datawrite(0x00); //CFR1
67 datawrite(CFR11);
68 datawrite(CFR12);
69 datawrite(CFR13);
70 datawrite(CFR14);
71 datawrite(0x01); //CFR2
72 datawrite(0x18);
73 datawrite(CFR21);
```

```

72     datawrite(CFR22);
73     datawrite(0x02);           //ASF
74     datawrite(ASF1);
75     datawrite(ASF2);
76     datawrite(0x03);           //ARR
77     datawrite(ARR);
78     datawrite(0x04);           //FTW0
79     datawrite(FTW1);
80     datawrite(FTW2);
81     datawrite(FTW3);
82     datawrite(FTW4);
83     datawrite(0x05);           //POW0
84     datawrite(POW1);
85     datawrite(POW2);
86     IOUPDATE = 1;
87     __delay_ms(10);
88     IOUPDATE = 0;
89
90
91     __delay_ms(10);
92     while(1){
93
94
95
96         prijate_slovo = getch();
97         switch (prijate_slovo){
98             case 0x00 :
99                 CFR11 = getch();
100                CFR12 = getch();
101                CFR13 = getch();
102                CFR14 = getch();
103                break;
104             case 0x01 :
105                 CFR21 = getch();
106                 CFR22 = getch();
107                 break;
108             case 0x02 :
109                 ASF1 = getch();
110                 ASF2 = getch();
111                 break;
112             case 0x03 :
113                 ARR = getch();
114                 break;
115             case 0x04 :
116                 FTW1 = getch();
117                 FTW2 = getch();
118                 FTW3 = getch();
119                 FTW4 = getch();
120                 break;
121             case 0x05 :
122                 POW1 = getch();
123                 POW2 = getch();
124                 break;
125             case 0xF0 :
126                 POM = 0x00;
127                 CFR11 &= 0xCF;
128                 break;
129             case 0xF1 :
130                 POM |= (1 << 5);
131                 CFR11 &= 0xDF;
132                 CFR11 |= 0x20;
133                 break;
134             case 0xF2 :
135                 POM |= (1 << 4);
136                 CFR11 &= 0xEF;
137                 CFR11 |= 0x10;
138                 break;
139             case 0xF3 :
140                 POM |= (1 << 4);
141                 POM |= (1 << 5);
142                 CFR11 |= 0x30;
143                 break;
144             case 0xAA :
145                 write();
146                 break;
147         }
148     /*     putchar(0x00);
149           putchar(CFR11);
150           putchar(CFR12);

```



```

151     putch(CFR13);
152     putch(CFR14);
153     putch(0x01);
154     putch(CFR21);
155     putch(CFR22);
156     putch(0x02);
157     putch(ASF1);
158     putch(ASF2);
159     putch(0x03);
160     putch(ARR);
161     putch(0x04);
162     putch(FTW1);
163     putch(FTW2);
164     putch(FTW3);
165     putch(FTW4);
166     putch(0x05);
167     putch(POW1);
168     putch(POW2);
169 */
170
171     }
172 }
173
174
175 //Podprogram odeslani znaku UARTem
176 void putch(unsigned char byte)
177 {
178     /* output one byte */
179     while(!TXIF) /* set when register is empty */
180         continue;
181     TXREG = byte;
182 }
183
184 //Podprogram prijmu dat z UARTu
185 unsigned char getch() {
186     /* retrieve one byte */
187     while(!RCIF) /* set when register is not empty */
188         continue;
189     return RCREG;
190 }
191
192 //Podprogram vysilani dat SPI do obvodu AD9951
193 void datawrite(char data){
194
195     for (int i=7; i>=0; i--)
196     {
197         RA1 = BIT(data, i);
198         __delay_ms(1);
199         RA0 = 1;
200         __delay_ms(1);
201         RA0 = 0;
202         __delay_ms(1);
203     }
204 }
205 }
206
207 void EEPROMwrite(char addr, char data)
208 {
209     EEADR = addr;
210     EEDATA = data;
211     WREN = 1;
212     EECON2 = 0x55;
213     EECON2 = 0xAA;
214     WR = 1;
215     __delay_ms(50);
216     WREN = 0;
217 }
218
219 char EEPROMread(char addr)
220 {
221     char data;
222     EEADR = addr;
223     RD = 1;
224     data = EEDATA;
225     return(data);
226 }
227
228 //Podprogram zapisu vseh registru do obvodu AD9951
229 void write(){

```

```

230     char INSTRBYTE;
231     INSTRBYTE=0b00000000; //registr Control Function Register No. 1
232     datawrite (INSTRBYTE); //CFR1
233     datawrite (CFR11);
234     datawrite (CFR12);
235     datawrite (CFR13);
236     datawrite (CFR14);
237     IOUPDATE = 1; //provedeni update registru
238     __delay_ms(10);
239     IOUPDATE = 0;
240
241     EEPROMwrite(0x00, CFR11);
242     EEPROMwrite(0x01, CFR12);
243     EEPROMwrite(0x02, CFR13);
244     EEPROMwrite(0x03, CFR14);
245
246     INSTRBYTE=0b00000001; //registr Control Function Register No. 2
247     datawrite (INSTRBYTE); //CFR2
248     datawrite (0x18);
249     datawrite (CFR21);
250     datawrite (CFR22);
251     IOUPDATE = 1;
252     __delay_ms(10);
253     IOUPDATE = 0;
254
255     EEPROMwrite(0x04, CFR21);
256     EEPROMwrite(0x05, CFR22);
257
258     INSTRBYTE=0b00000010; //registr Amplitude Scale Factor
259     datawrite (INSTRBYTE); //ASF
260     datawrite (ASF1);
261     datawrite (ASF2);
262     IOUPDATE = 1;
263     __delay_ms(10);
264     IOUPDATE = 0;
265
266     EEPROMwrite(0x06, ASF1);
267     EEPROMwrite(0x07, ASF2);
268
269     INSTRBYTE=0b00000011; //registr Amplitude Ramp Rate
270     datawrite (INSTRBYTE); //ARR
271     datawrite (ARR);
272     IOUPDATE = 1;
273     __delay_ms(10);
274     IOUPDATE = 0;
275
276     EEPROMwrite(0x08, ARR);
277
278     INSTRBYTE=0b00000100; //registr Frequency Tuning Word
279     datawrite (INSTRBYTE); //FTW0
280     datawrite (FTW1);
281     datawrite (FTW2);
282     datawrite (FTW3);
283     datawrite (FTW4);
284     IOUPDATE = 1;
285     __delay_ms(10);
286     IOUPDATE = 0;
287
288     EEPROMwrite(0x09, FTW1);
289     EEPROMwrite(0x0A, FTW2);
290     EEPROMwrite(0x0B, FTW3);
291     EEPROMwrite(0x0C, FTW4);
292
293     INSTRBYTE=0b00000101; //registr Phase Offset Word
294     datawrite (INSTRBYTE); //POW0
295     datawrite (POW1);
296     datawrite (POW2);
297     IOUPDATE = 1;
298     __delay_ms(10);
299     IOUPDATE = 0;
300
301     EEPROMwrite(0x0D, POW1);
302     EEPROMwrite(0x0E, POW2);
303 }

```