

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

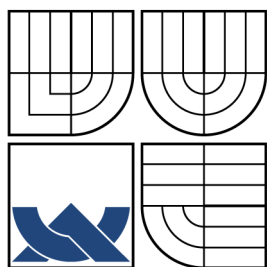
PŘÍMÝ ČÍSLICOVÝ SYNTEZÁTOR PRO MIKROVLNNÉ
APLIKACE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

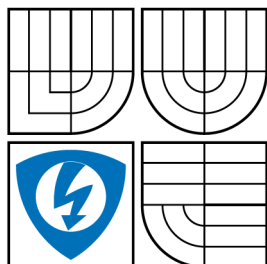
AUTOR PRÁCE
AUTHOR

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

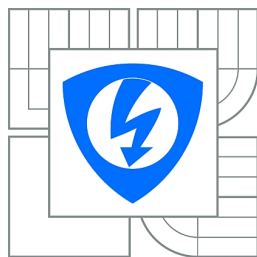
AUTOR PRÁCE
AUTHOR

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: 21.5.2015

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 s návrhem přímého číslicového syntezátoru s obvodem AD9951 od společnosti Analog Devices. Zařízení bude řízeno z PC přes USB. Zařízení pracuje s vnitřním oscilátorem, s možností připojení externího kmitočtového normálu o frekvenci 10 MHz.

KLÍČOVÁ SLOVA

frekvenční syntéza, kmitočtová syntéza, syntezátor, přímý číslicový syntezátor, 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.

KEYWORDS

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

DLUHÝ, Vojtěch *Přímý číslicový syntezátor pro mikrovlnné aplikace*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2015. 42 s. Vedoucí práce byl prof. Ing. Miroslav Kasal, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Přímý číslicový syntezátor pro mikrovlnné 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.

Brno

.....

(podpis autora)

OBSAH

Úvod	9
A Úvod k syntezátorům	10
A.1 Syntezátory s fázovým závěsem	10
A.2 Přímé číslicové syntezátory	11
B Návrh syntezátoru	13
B.1 Napájecí zdroje	13
B.2 Obvody řízení	15
B.3 Obvod číslicové syntézy AD9951	15
B.4 Výstupní filtr	17
B.5 FW mikroprocesoru	18
B.6 SW pro počítač	18
C Realizace navrženého syntezátoru	19
C.1 Napájecí zdroj	19
C.2 Převodník USB/UART	19
C.3 Mikroprocesor	19
C.4 Obvod přímé číslicové syntézy	20
C.5 Ovládací SW	20
C.6 FW mikroprocesoru	22
C.7 Mechanické provedení	24
C.8 Oživení a programování	25
D Závěr	27
Literatura	28
Seznam symbolů, veličin a zkratk	29
Seznam příloh	30
A Schéma zdroje	31
B Schéma řídicích obvodů	33
C Schéma obvodu syntezátoru	35
D Desky plošných spojů	37

SEZNAM OBRÁZKŮ

A.1	Smyčka fázového závěsu	10
A.2	Příklad výstupního spektra PLL s viditelným fázovým šumem[1] . . .	11
A.3	Blokové schéma DDS	11
A.4	Typické spektrum obvodu DDS [2]	12
B.1	Blokové schéma zařízení	13
B.2	Schéma 1,8 V s obvodem LM317	14
B.3	Blokové schéma obvodu AD9951[3]	16
B.4	Schéma výstupního filtru	16
B.5	Modulová a fázová charakteristika výstupního filtru	17
C.1	Hlavní okno ovládacího programu	21
C.2	Detailní nastavení obvodu AD9951 v ovládacím SW	22
C.3	Vývojový diagram FW mikroprocesoru	23
C.4	Rozmístění hlavní částí obvodu	24
C.5	Fotografie osazené prototypové desky	25
C.6	Časový průběh UART komunikace mezi PC a mikroprocesorem . . .	26
C.7	Časový průběh UART komunikace mezi procesorem a AD9951	26
D.1	DPS - strana součástek (horní)	38
D.2	DPS - strana spojů (spodní)	38

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 zá-
věsu a také přímé číslicové generátory, kterým se budu ve své práci také věnovat.

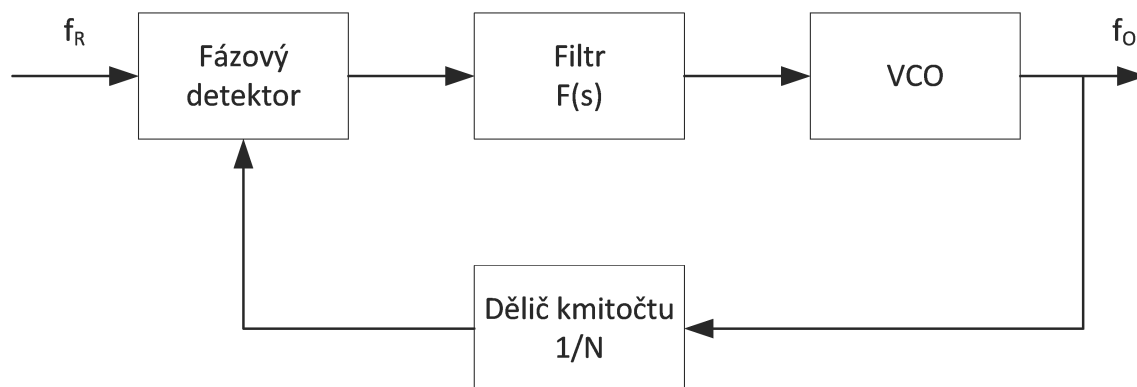
Další podstatnou částí této práce je návrh a konstrukce přímého číslicového syntezátoru s obvodem AD9951 řízeným z PC sběrnici USB. USB komunikaci obstarává obvod FT232 s procesorem PIC16F628A. Vstupní signál je upraven zdvojovačem s bipolárním tranzistorem na kmitočet potřebný pro správnou funkci obvodu AD9951, výstupní signál je filtrován filtrem 7. řádu s Caurovou aproximací. Výstupní signál je zesílen trojbodovým zesilovačem ERA-3+. Napájení je provedeno čtyřmi zdroji (+1,8 V, +1,8 V, +3,3 V a +5 V).

A ÚVOD K SYNTÉZÁTORŮM

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 syntezeátorů oproti oscilátorům jsou široká možnost přeladění výstupního kmitočtu a jednoduchost tohoto úkonu, který se provádí většinou určitým řídicím slovem z připojeného mikroprocesoru.

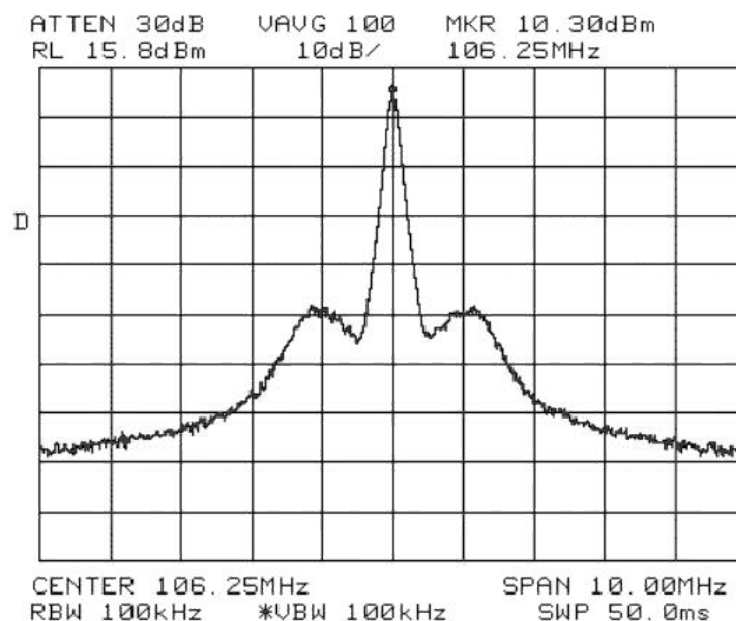
A.1 Syntezátory s fázovým závěsem

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. A.1), kterými jsou fázový detektor, filtr smyčky, napětím řízený oscilátor (VCO) a volitelně ve zpětné vazbě dělič kmitočtu. Fázový detektor zjišťuje



Obr. A.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.



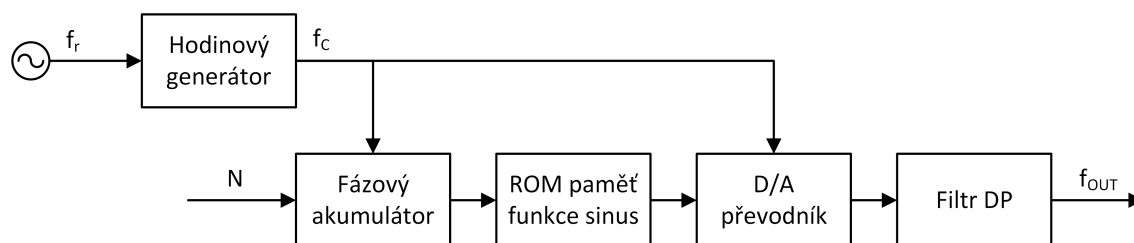
Obr. A.2: Příklad výstupního spektra PLL s viditelným fázovým šumem[1]

Změna výstupního kmitočtu se provádí změnou dělicího poměru ve zpětné vazbě. Výstupní signál pak má kmitočet f_O dle vzorce A.1.

$$f_O = N \cdot f_R \quad (\text{A.1})$$

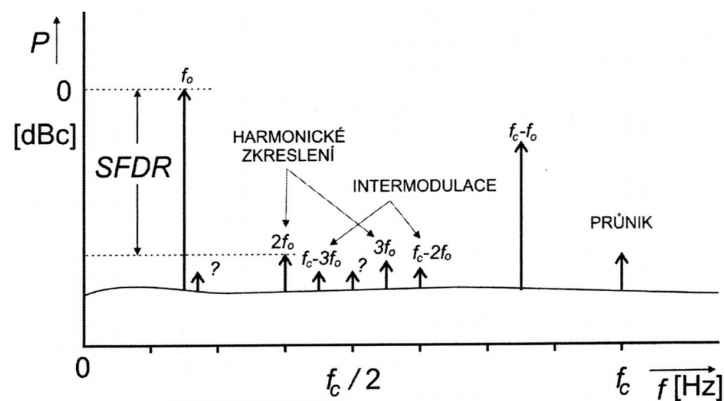
kde f_O je výstupní kmitočet, N je dělicí poměr a f_R je vstupní (referenční) kmitočet. Výstupní spektrum tohoto typu syntezátoru obsahuje dvě hlavní parazitní složky, které jsou způsobeny parazitní fázovou a amplitudovou modulací. Příklad výstupního signálu s fázovým šumem je na obr. A.2.

A.2 Přímé číslicové syntezátory



Obr. A.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-

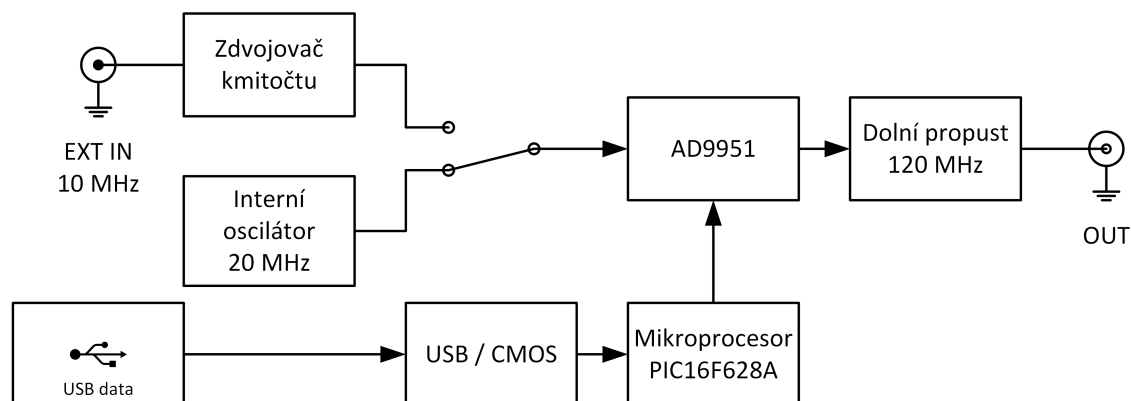


Obr. A.4: Typické spektrum obvodu DDS [2]

zového akumulátoru o hodnotu odpovídající ladícímu slovu. Tato hodnota je ukazatelem na hodnoty uložené v ROM paměti. Výstupem z paměti je diskretní 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.A.3.

Typické spektrum včetně parazitních signálů můžeme pozorovat na obr.A.4. Můžeme zde vidět zrcadlení kolem $f_c/2$ a také parametr SFDR, který ukazuje amplitudový odstup parazitních spektrálních složek od požadovaného kmitočtu.

B NÁVRH SYNTEZÁTORU



Obr. B.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+
- interní krystalový oscilátor 20 MHz s možností přepnutí ke konektoru k připojení externího kmitočtového normálu
- zdvojovač kmitočtu ze vstupního konektoru, 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

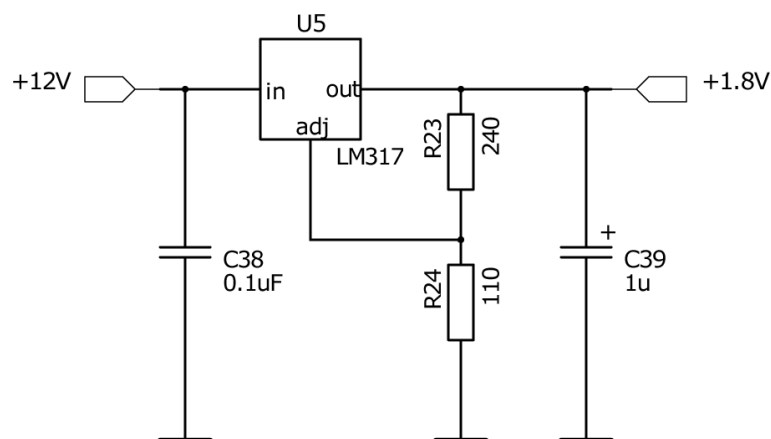
Jednotlivým částem se budu podrobněji věnovat v následujících podkapitolách.

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

B.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 LR232.

Zdroje 1,8 V budou tvořeny základním katalogovým zapojením obvodu LM317 nastaveným pro 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 mW¹. 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.B.2.



Obr. B.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.

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

B.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 SW, pro ovládání obvodu DDS AD9951. Obvod LT232 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 přenášných 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 kabelu. Schéma řídicího obvodu je umístěno v příloze B.

B.3 Obvod číslicové syntézy AD9951

Srdcem celého zařízení bude obvod DDS AD9951 společnosti Analog Devices[3]. Z katalogového listu a blokového schématu (obr.B.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 B.1 pro výpočet výstupního kmitočtu f_{OUT} ve vztahu B.2, vychází na $\Delta f = 0,093$ Hz.

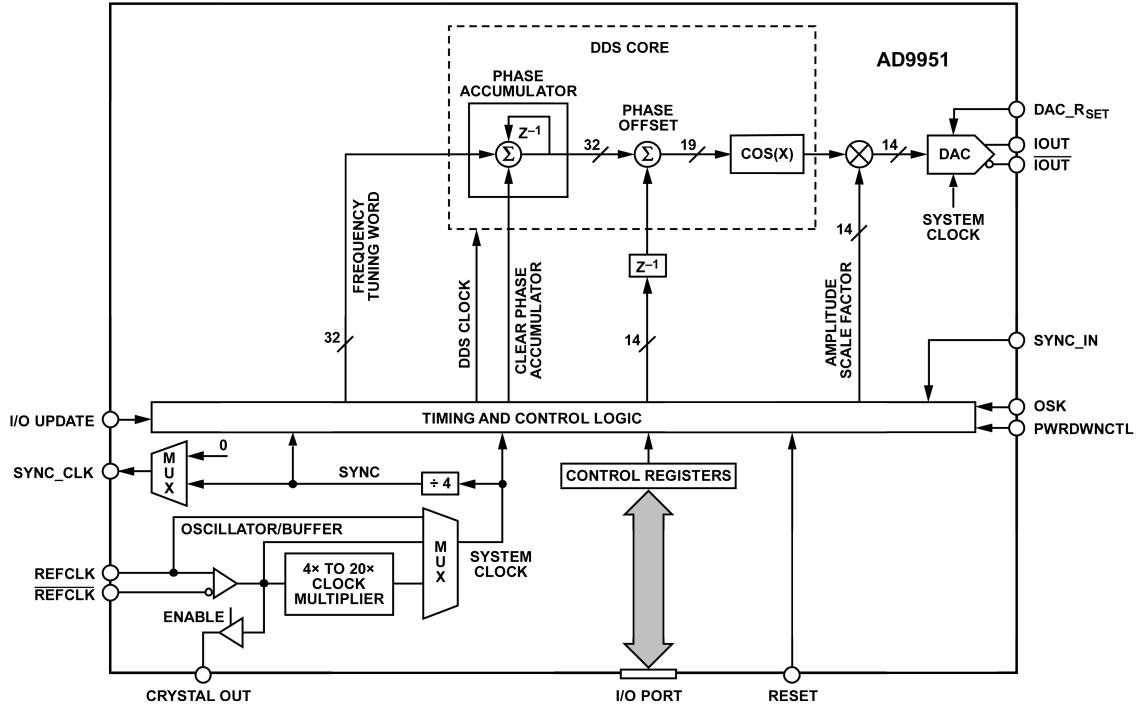
$$f_{OUT} = (FTW) \cdot (f_S)/2^{32} \quad (\text{B.1})$$

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

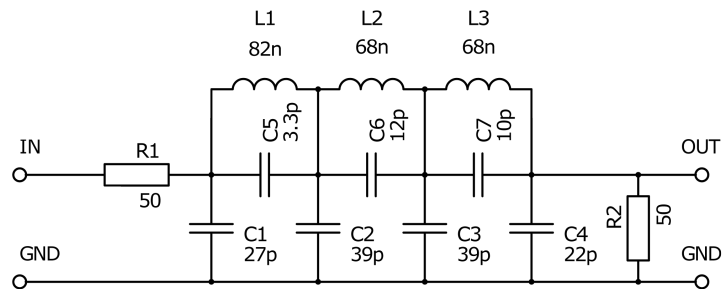
kde $f_S=400$ MHz a FTW je ladící slovo.

$$R_{SET} = 39,19/I_{OUT} \quad (B.3)$$

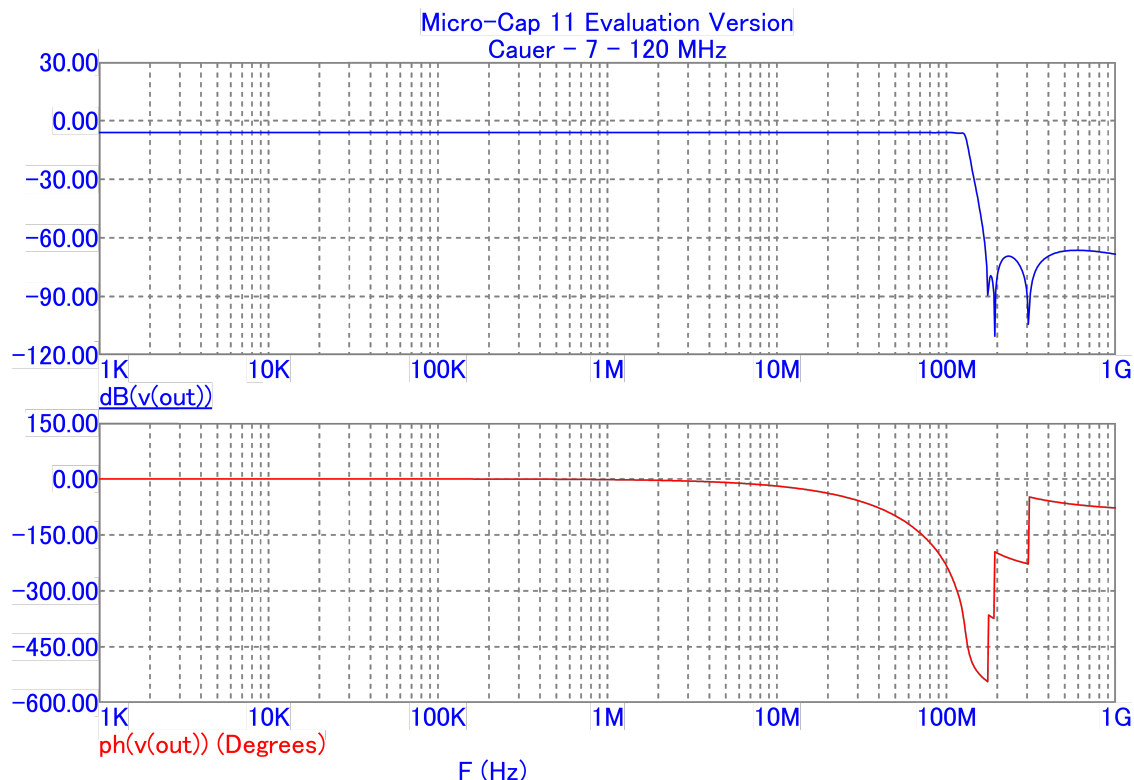
Pro nejlepší SFDR je potřeba omezit výstupní proud I_{OUT} na 10 mA, proto dle vzorce B.3 vypočítáme rezistor, který bude připojen k vývodu DAC_RSET. Jeho hodnota činí dle řady E12 $R_{SET}=3,9k\Omega$.



Obr. B.3: Blokové schéma obvodu AD9951[3]



Obr. B.4: Schéma výstupního filtru



Obr. B.5: Modulová a fázová charakteristika výstupního filtru

B.4 Výstupní filtr

Na výstupu obvodu AD9951 je sice již analogový signál, ovšem schodovitého průběhu, který je způsoben vyčítáním číselných hodnot z tabulky ROM. Přitom platí, čím vyšší výstupní kmitočet, tím více obdélníkový výstupní signál bude. Proto na výstupu obvodu syntézy musí být umístěn výstupní filtr s mezním kmitočtem na nejvyšší nastavitelné hodnotě. V našem případě se jedná o 120 MHz. Při posouzení několika aproximací filtrů dolní propusti jsem se uchýlil k návrhu Cauera přímokového filtru sedmého řádu, který má největší strmost ze všech uvažovaných filtrů. Výstup zařízení je požadován nesymetrický, zatímco výstup obvodu je symetrický. Nesymetrizaci by bylo 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, který provede onu nesymetrizaci. Dále by bylo možné signál znesymetrizovat připojením jednoho výstupního vodiče přes rezistor ke kladnému napájení AVDD (+1,8 V). Tím by ale došlo k úbytku na výstupním napětí.

Pro návrh filtru jsem využil volně dostupný SW s názvem Elsie[9], 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.B.4 a jeho modulovou a fázovou charakteristiku na obr.B.5. Celkové schéma signálových obvodů je v příloze C.

B.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. A také odeslání nastavení mikroprocesoru do PC. 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.

B.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.

C 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.

C.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 B.1.

C.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 mohu 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.

C.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. odvodu AD9951). Pro použití v této aplikaci jsem zvolil procesor PIC16F628A fy Microchip. Tento obyčejný mikroprocesor svým výkonem pro tuto jednoduchou činnost překladu 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.

C.4 Obvod přímé číslicové 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 čemuž mohu na vstup obvodu připojit signál o kmitočtu 20 MHz. I to je však hodně, jelikož požadavek na vstupní kmitočet, pro použití s externím normálem byl 10 MHz. Proto musí být mezi vstupním konektorem a obvodem syntézy zapojen násobič kmitočtu, který byl realizován jako násobič s bipolárním tranzistorem, na jehož nelinearitě vznikají vyšší harmonické, z nichž je následně vyfiltrována pouze jedna na 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.

Další analogovou částí zařízení je výstupní filtr typu dolní propust vypočítaný na kmitočet 120 MHz, který zaručí vyfiltrování výstupního signálu, aby nebyl schodovitý (resp. obdélníkový) průběh, který je způsoben 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 Ω transformátorku v miniaturním provedení od společnosti MiniCircuits. Na výstupu filtru je opět tříbodový zesilovač ERA-3+, který zajistí zesílení výstupního signálu na potřebnou úroveň a zároveň zajistí impedanční přizpůsobení.

C.5 Ovládací SW

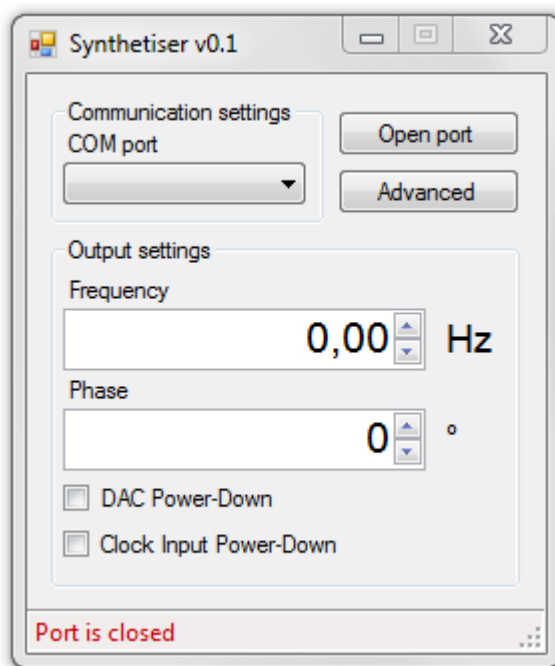
Ovládací SW v počítači je naprogramován v jazyce C# a dovoluje kompletní ovládání obvodu syntezátoru AD9951. Komunikace s HW probíhá skrz USB komunikaci,

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.C.1) umožňuje nastavit nejdůležitější parametry, jako je komunikační COM port, výstupní kmitočet syntezátoru, 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. C.1: Hlavní okno ovládacího programu

Detailní nastavení jednotlivých registrů obvodu AD9951 je v druhém okně ovládacího SW (obr.C.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 jej proto využít v jiných aplikacích. Ať už by se jednalo přímo o toto zapojení nebo o jiné zapojení s možností ovládaní prostřednictvím tohoto SW.

Advanced settings

Control Function Register No. 1 (CFR1)

☐ 1 - SYNC_CLK Out Disable	☐ 8 - LSB first	☐ 22 - Software Manual Sync
☐ 3 - External Power-Down Mode	☐ 9 - SDIO Input Only	☐ 23 - Automatic Sync Enable
☐ 4 - Clock Input Power-Down	☐ 10 - Clear Phase Accum.	☐ 24 - Auto OSK Keying
☐ 5 - DAC Power-Down	☐ 12 - Enable SINE Output	☐ 25 - OSK Enable
☐ 7 - Digital Power-Down	☐ 13 - AutoClr Phase Accum	☐ 26 - Load ARR @I/O UD

Control Function Register No. 2 (CFR2)

Charge Pump Current:

REFCLK Multiplier:

☐ VCO Range

☐ CRYSTAL OUT Pin Active

☐ HW Manual Sync Enable

☐ High Speed Sync Enable

Amplitude Scale Factor (ASF)

Amplitude Scale Factor Register:

Auto Ramp Rate Speed:

Frequency Tuning Word (FTW0)

Frequency Tuning Word: = 0 Hz

Amplitude Ramp Rate (ARR)

Amplitude Ramp Rate Register:

Phase Offset Word (POW0)

Phase Offset Word: = 0°

Load DEFAULT Save to device

Close Read from device

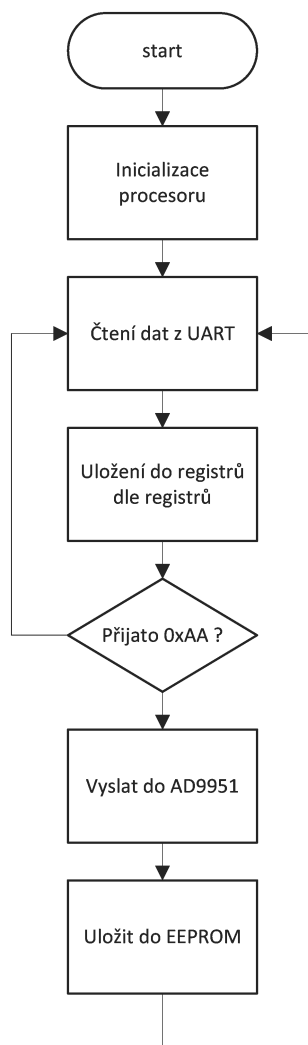
Obr. C.2: Detailní nastavení obvodu AD9951 v ovládacím SW

C.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 potvrzeno vodičem IOUPDATE, po kterém dojde k nahrání zaslaných dat ze vstupních posuvných registrů obvodu do jeho pracovních registrů.

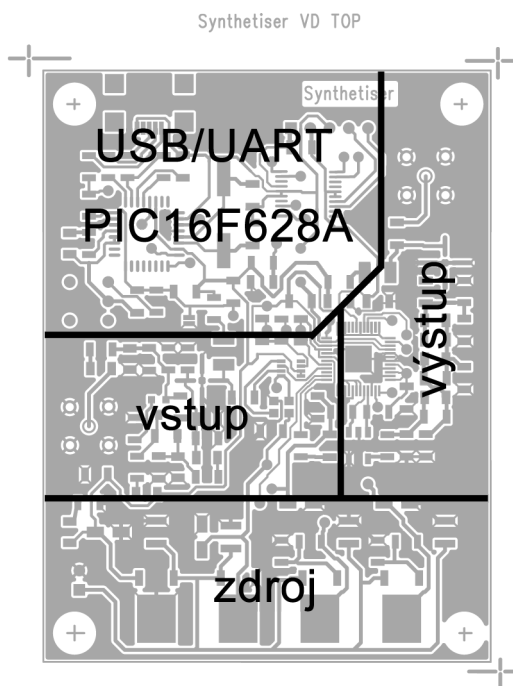
Vývojový diagram FW je na obr.C.3. Program vyčkává na zaslání datového slova z PC, dále dle prvního (adresního) bajtu rozhodne o který registr se jedná a uloží zbylá data do příslušných registrů. Po příjmu potvrzovacího slova (0xAA) odešle tyto informace do syntezátoru a zároveň je uloží do své interní 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. C.3: Vývojový diagram FW mikroprocesoru

C.7 Mechanické provedení

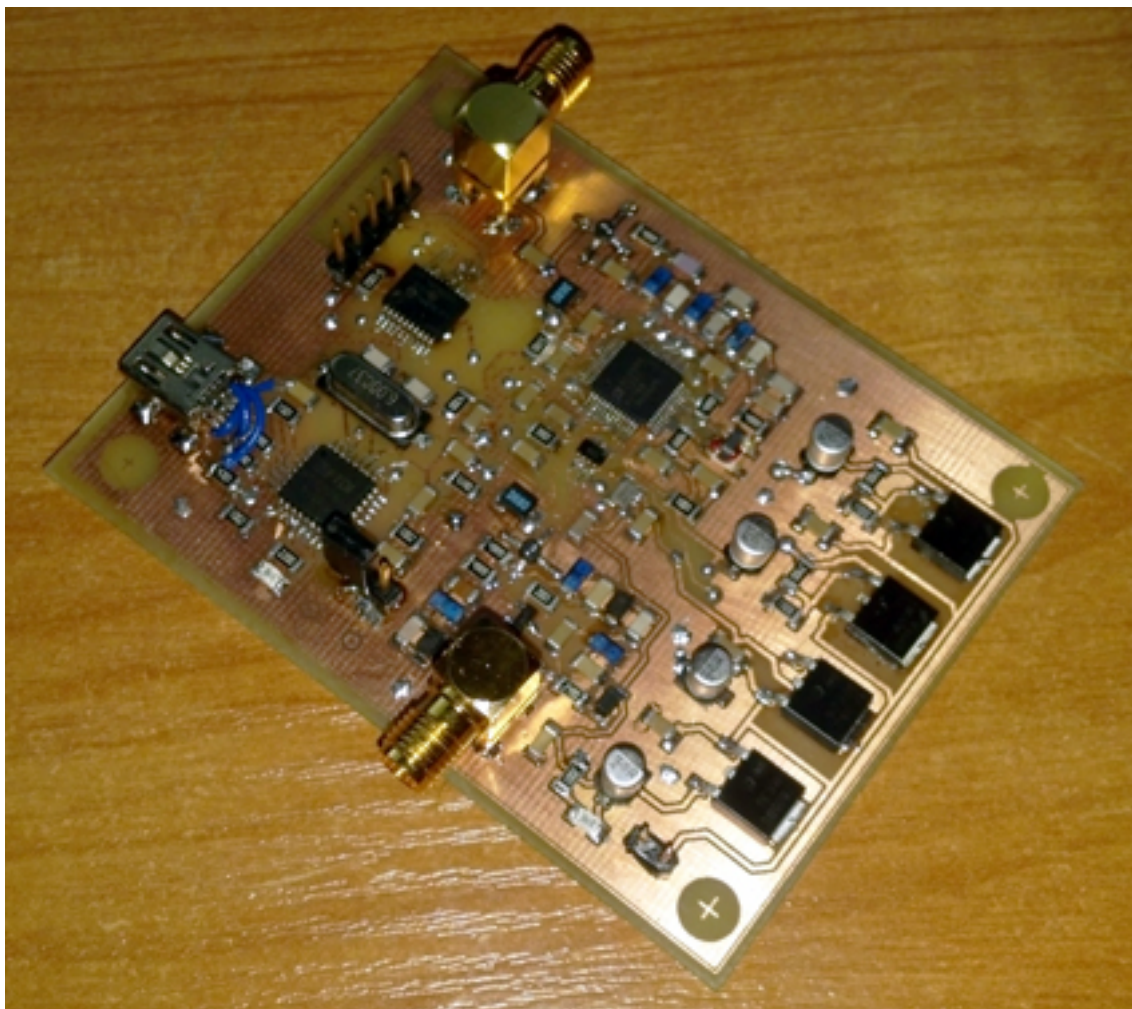
Zařízení je realizováno jako jednodeskové se součástkami na jedné straně. Jednotlivé hlavní části jsou logicky rozvrženy a umístěny na desce aby nebylo nutné vést propojovací cesty napříč celou DPS. Hlavní části tedy jsou: zdroj, digitální část (převodník USB/UART, mikroprocesor), vstupní část syntezátoru a výstupní část syntezátoru. Rozvržení součástek je na obr.C.4.



Obr. C.4: Rozmístění hlavní částí obvodu

Vyrobenou DPS bylo nejprve nutné vyvrtat a ručně prokovit. Prokovy byly provedeny pomocí drátku, který byl z obou stran desky plošných spojů přiletován. Po dokončení prokovů jsem ohmmetrem zkontroloval veškeré propojení na DPS a začal s osazováním. Po osazení a oživení zdrojů jsem osadil zbytek součástek, opět zkontroloval propojení a utvrdil se, že je vše v pořádku. Po připojení zařízení USB kabelem k počítači a zapnutí zařízení došlo v počítači k instalaci ovladačů virtuálního sériového (COM) portu, po čemž jsem se mohl pustit do programování mikroprocesoru.

Motivy oboustranné desky plošných spojů jsou zobrazeny v příloze D. Fotografie osazené desky ze strany součástek je na obr.C.5. Na této prototypové desce je možné vidět chybné zapojení pinů konektorů USB, které musely být pomocí drátků předráťovány do správného pořadí.

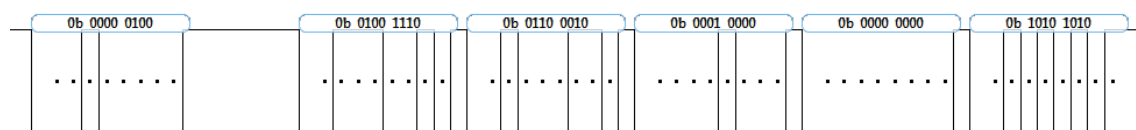


Obr. C.5: Fotografie osazené prototypové desky

C.8 Oživení a programování

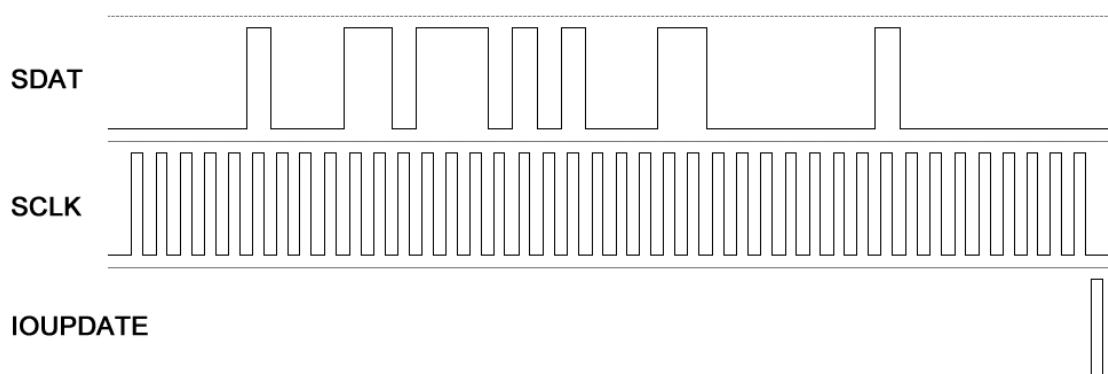
Po osazení, oživení a zkontrolování napětí na všech integrovaných obvodech jsem se mohl pustit do programování mikroprocesoru. Program mikroprocesoru je psán v jazyce C v programovacím prostředí MPLAB IDE. Program je pro svou jednoduchost napsán pouze v jednom souboru zdrojového kódu. Pro správnou funkčnost této aplikace, čili generování výstupního kmitočtu, postačuje nastavit činitel násobení vstupního kmitočtu fázovým závěsem a požadovaný výstupní kmitočet obvodu. 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.C.6. První bajt představuje adresu registru FTW (0x04 Frequency tuning word - Ladící slovo kmitočtu), následující čtyři bajty obsahují slovo představující zadaný kmitočet, přepočítaný dle B.1

a poslední bajt je potvrzovací (0xAA). Příjetím 0xAA mikroprocesor ví, že všechna data byla pro tuto chvíli odeslána a může začít s jejich zpracováním.



Obr. C.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 dvou vodičů. První vodič vede datový signál a druhý vodič vede hodinový signál. Syntezátor reaguje vždy na náběžnou hranu hodinového signálu. Další podmínkou 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.C.7 zobrazena komunikace do obvodu AD9951 se stejnými daty jako příklad komunikace UART z obr.C.6.



Obr. C.7: Časový průběh UART komunikace mezi procesorem a AD9951

Bohužel se vyskytla situace, kdy obvod AD9951 nereaguje na zasláná data, která jsou zcela jistě ve správném tvaru. Při diagnostice problému jsem zjistil, že hodiny obvodem prochází, jelikož na výstupním hodinovém pinu SYNC_ CLK (viz. blokové schéma AD9951 na obr.B.3) byly hodiny o čtvrtinovém kmitočtu oscilátoru. Při pokusu o čtení registrů po zapnutí zařízení, kdy jsou všechny registry defaultně nastaveny do hodnot 0x00, jsou na datovém výstupu hodnoty o vysoké napěťové úrovni. Mohu však s jistotou říct, že požadavek o čtení dat z obvodu AD9951 byl odeslán správně. Vyzkoušel jsem také různá časování hodinového signálu a bez reakce obvodu. Z toho usuzuji, že je tento obvod vadný.

D ZÁVĚR

Frekvenční syntéza je v poslední době jedna z nejpoužívanějších metod tvorby 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 možností velkého přeladění 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 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 Ω 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, které je využito při zapnutí zařízení, v tu chvíli mikroprocesor načte z paměti EEPROM uložená data a vyšle je do obvodu AD9951.

Bohužel, se vyskytl problém s obvodem AD9951, který jinak nereaguje na zasláná data. Neodpovídá na žádost o čtení dat jeho registrů, ani po zápise do jeho registrů neprovede žádné změny se svým výstupem. Řešení se naskýtá ve výměně obvodu a dalším testování.

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>.
- [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>.
- [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>.
- [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>.
- [9] Program pro výpočet filtrů a příčkových struktur. *Elsie*. [cit. 2014-12-10]. Dostupné z: <<http://tonnesoftware.com/elsie.html>>.

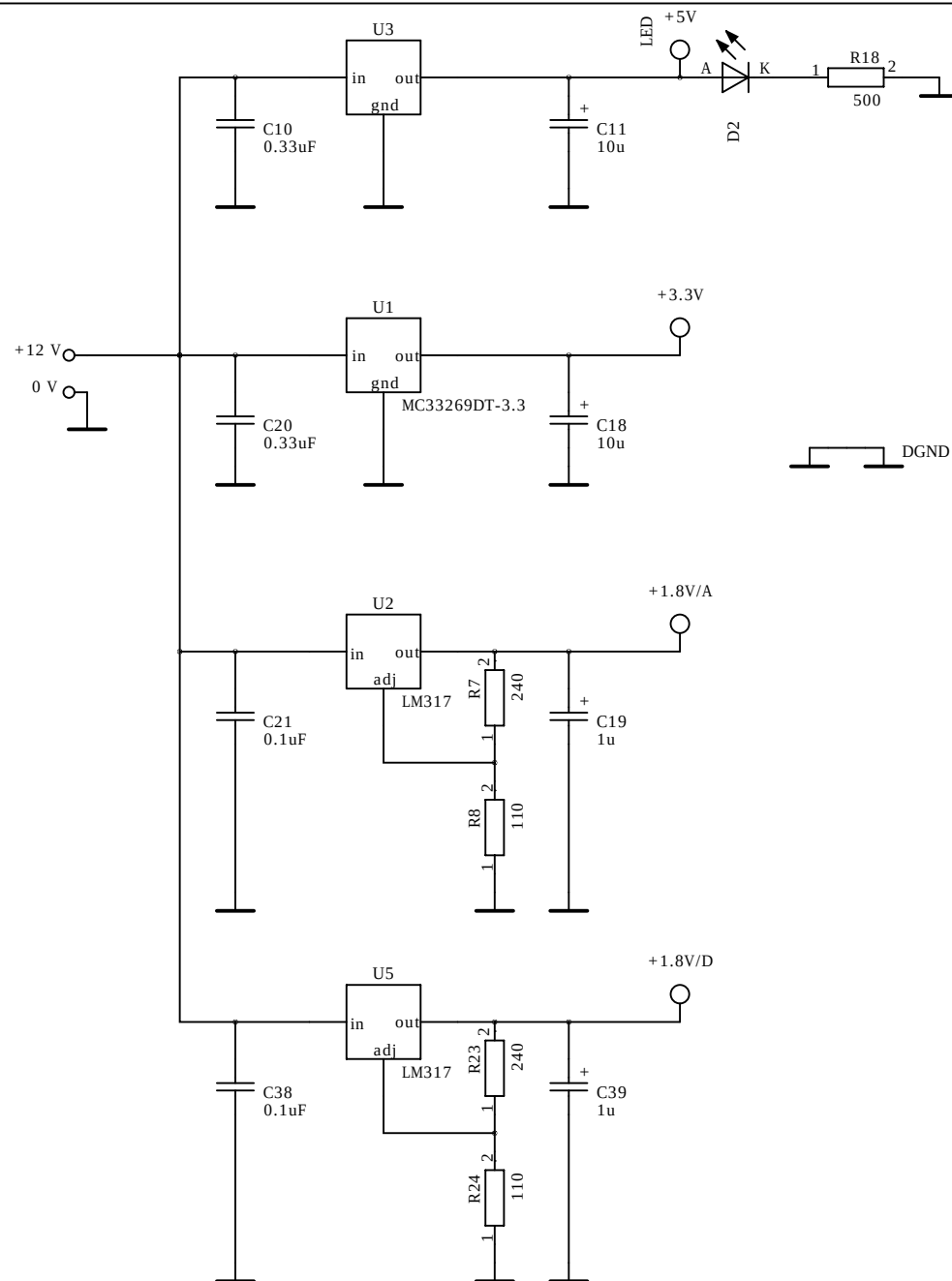
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

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
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
vf	vysokofrekvenční
f_{OUT}	výstupní kmitočet

SEZNAM PŘÍLOH

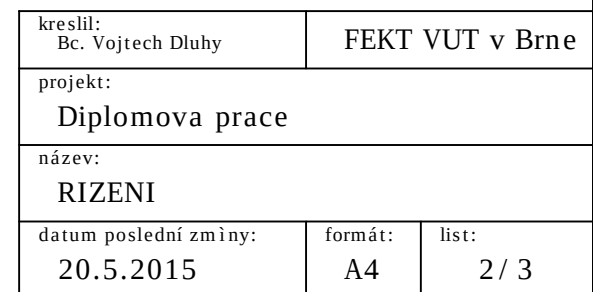
A Schéma zdroje	31
B Schéma řídicích obvodů	33
C Schéma obvodu syntezátoru	35
D Desky plošných spojů	37
E Zdrojový kód FW	39

A SCHÉMA ZDROJE

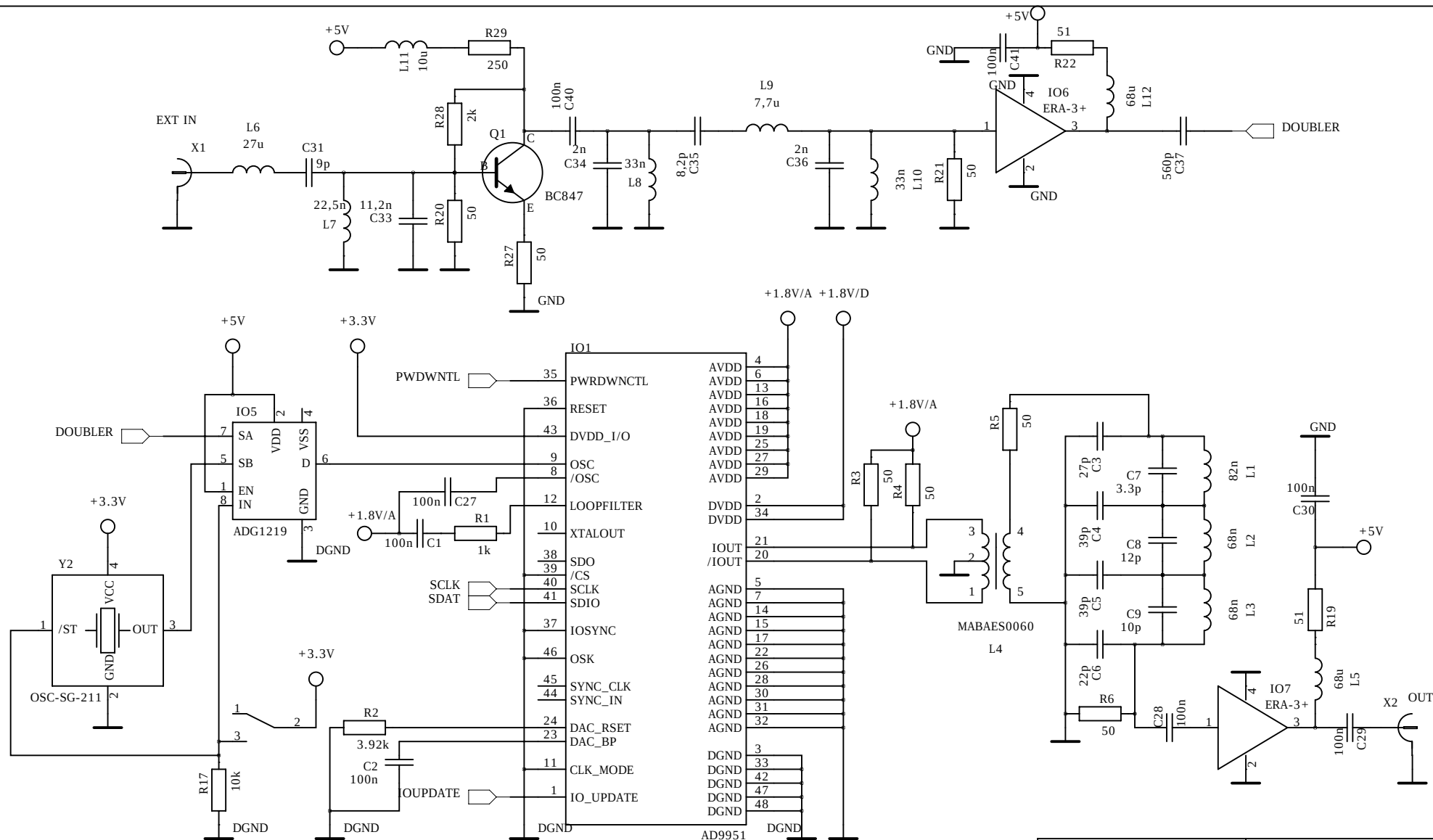


kreslil: Bc. Vojtech Dluhy	FEKT VUT v Brne	
projekt: Diplomova prace		
název: ZDROJ		
datum poslední zmeny: 20.5.2015	formát: A4	list: 1 / 3

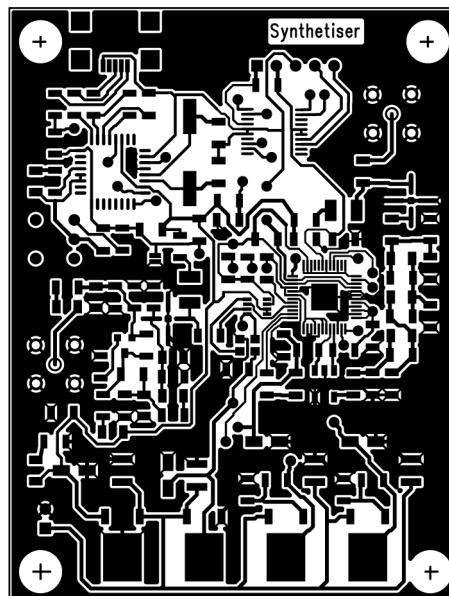
B SCHÉMA ŘÍDÍCÍCH OBVODŮ



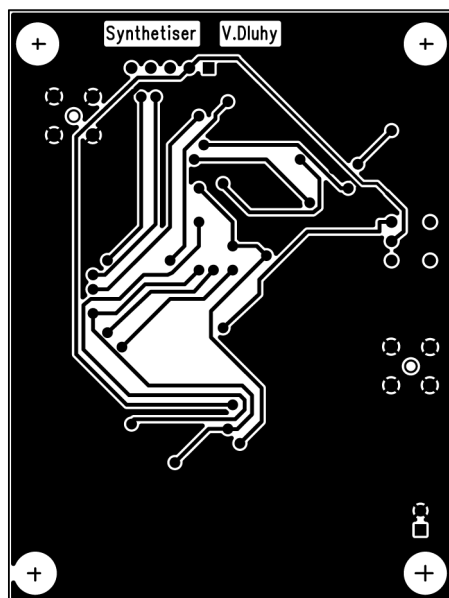
C SCHÉMA OBVODU SYNTEZÁTORU



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 PWDWNIL 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,
    POW1, POW2, POM;
18
19 // Predpisy funkci
20 void putch(unsigned char byte);
21 unsigned char getch();
22 unsigned char getche(void);
23 void datawrite(char data);
24 void EEPROMwrite(char addr, char data);
25 char EEPROMread(char addr);
26 void write();
27
28
29 // Hlavni program
30 void main(){
31     CMCON = 0x07; //nastaveni pinu portu A jako digitalni
32     TRISA = 0x00; //nastaveni portu A jako vystupni
33     TRISB1 = 1; //nastaveni portu B1 jako vstupni —+—> pro spravnou funkci
        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 por příjem
39     SPBRG = 25; //nastaveni hodnoty delicky, kterou se ridi rychlost komunikace
        UART
40
41     SCLK = 0; //nulovani vystupu SCLK (port A0)
42     SDAT = 0; //nulovani vystupu SDAT (port A1)
43     IOUPDATE = 0; //nulovani vystupu IOUPDATE (port A2)
44     PWDWNIL = 0; //nulovani vystupu PWDWNIL (port A3)
45
46     CFR11=CFR12=CFR13=CFR14=CFR21=CFR22=ASF1=ASF2=ARR=FTW1=FTW2=FTW3=FTW4=POW1=POW2=POM=0;
47
48 // Cteni ulozenych registru z pameti EEPROM
49 CFR11 = EEPROMread(0x00);
50 CFR12 = EEPROMread(0x01);
51 CFR13 = EEPROMread(0x02);
52 CFR14 = EEPROMread(0x03);
53 CFR21 = EEPROMread(0x04);
54 CFR22 = EEPROMread(0x05);
55 ASF1 = EEPROMread(0x06);
56 ASF2 = EEPROMread(0x07);
57 ARR = EEPROMread(0x08);
58 FTW1 = EEPROMread(0x09);
59 FTW2 = EEPROMread(0x0A);
60 FTW3 = EEPROMread(0x0B);
61 FTW4 = EEPROMread(0x0C);
62 POW1 = EEPROMread(0x0D);
63 POW2 = EEPROMread(0x0E);
64
65 while(1){
66
67     __delay_ms(2);
68
69     prijate_slovo = getch();
70     zmena=0x00;
71     switch (prijate_slovo){
```



```

72         case 0x00 :
73             CFR11 = getch();
74             CFR12 = getch();
75             CFR13 = getch();
76             CFR14 = getch();
77             break;
78         case 0x01 :
79             CFR21 = getch();
80             CFR22 = getch();
81             break;
82         case 0x02 :
83             ASF1 = getch();
84             ASF2 = getch();
85             break;
86         case 0x03 :
87             ARR = getch();
88             zmena|=0x08;
89             break;
90         case 0x04 :
91             FTW1 = getch();
92             FTW2 = getch();
93             FTW3 = getch();
94             FTW4 = getch();
95             zmena|=0x10;
96             break;
97         case 0x05 :
98             POW1 = getch();
99             POW2 = getch();
100            break;
101         case 0xF0 :
102             POM = 0x00;
103             CFR11 &= 0xCF;
104             break;
105         case 0xF1 :
106             POM |= (1 << 5);
107             CFR11 &= 0xDF;
108             CFR11 |= 0x20;
109             break;
110         case 0xF2 :
111             POM |= (1 << 4);
112             CFR11 &= 0xEF;
113             CFR11 |= 0x10;
114             break;
115         case 0xF3 :
116             POM |= (1 << 4);
117             POM |= (1 << 5);
118             CFR11 |= 0x30;
119             break;
120         case 0xAA :
121             write();
122             break;
123     }
124 }
125 }
126 }
127
128 //Podprogram odeslani znaku UARTem
129 void putch(unsigned char byte)
130 {
131     while(!TXIF)
132         continue;
133     TXREG = byte;
134 }
135
136 //Podprogram prijmu dat z UARTu
137 unsigned char getch() {
138     while(!RCIF)
139         continue;
140     return RCREG;
141 }
142
143 //Podprogram vysilani dat SPI do obvodu AD9951
144 void datawrite(char data){
145     for (int i=7; i>=0; i--)
146     {
147         RA1 = BIT(data, i);
148         __delay_ms(1);
149         RA0 = 1;
150     }

```

```

151         __delay_ms(2);
152         RA0 = 0;
153         __delay_ms(1);
154     }
155 }
156 }
157
158 void EEPROMwrite(char addr, char data)
159 {
160     EEADR = addr;
161     EEDATA = data;
162     WREN = 1;
163     EECON2 = 0x55;
164     EECON2 = 0xAA;
165     WR = 1;
166     __delay_ms(50);
167     WREN = 0;
168 }
169
170 char EEPROMread(char addr)
171 {
172     char data;
173     EEADR = addr;
174     RD = 1;
175     data = EEDATA;
176     return(data);
177 }
178
179 //Podprogram zapisu vseh registru do obvodu AD9951
180 void write(){
181     char INSTRBYTE;
182     INSTRBYTE=0b00000000; //registr Control Function Register No. 1
183     datawrite(INSTRBYTE); //CFR1
184     datawrite(CFR11);
185     datawrite(CFR12);
186     datawrite(CFR13);
187     datawrite(CFR14);
188     IOUPDATE = 1; //provedeni update registru
189     __delay_ms(2);
190     IOUPDATE = 0;
191
192     EEPROMwrite(0x00, CFR11);
193     EEPROMwrite(0x01, CFR12);
194     EEPROMwrite(0x02, CFR13);
195     EEPROMwrite(0x03, CFR14);
196
197     INSTRBYTE=0b00000001; //registr Control Function Register No. 2
198     datawrite(INSTRBYTE); //CFR2
199     datawrite(CFR21);
200     datawrite(CFR22);
201     datawrite(0x18);
202     IOUPDATE = 1;
203     __delay_ms(2);
204     IOUPDATE = 0;
205
206     EEPROMwrite(0x04, CFR21);
207     EEPROMwrite(0x05, CFR22);
208
209     INSTRBYTE=0b00000010; //registr Amplitude Scale Factor
210     datawrite(INSTRBYTE); //ASF
211     datawrite(ASF1);
212     datawrite(ASF2);
213     IOUPDATE = 1;
214     __delay_ms(2);
215     IOUPDATE = 0;
216
217     EEPROMwrite(0x06, ASF1);
218     EEPROMwrite(0x07, ASF2);
219
220     INSTRBYTE=0b00000011; //registr Amplitude Ramp Rate
221     datawrite(INSTRBYTE); //ARR
222     datawrite(ARR);
223     IOUPDATE = 1;
224     __delay_ms(2);
225     IOUPDATE = 0;
226
227     EEPROMwrite(0x08, ARR);
228
229     INSTRBYTE=0b00000100; //registr Frequency Tuning Word

```

```

230     datawrite (INSTRBYTE);    //FTW0
231     datawrite (FTW1);
232     datawrite (FTW2);
233     datawrite (FTW3);
234     datawrite (FTW4);
235     IOUPDATE = 1;
236     __delay_ms(2);
237     IOUPDATE = 0;
238
239     EEPROMwrite(0x09, FTW1);
240     EEPROMwrite(0x0A, FTW2);
241     EEPROMwrite(0x0B, FTW3);
242     EEPROMwrite(0x0C, FTW4);
243
244     INSTRBYTE=0b00000101;    //registr Phase Offset Word
245     datawrite (INSTRBYTE);    //POW0
246     datawrite (POW1);
247     datawrite (POW2);
248     IOUPDATE = 1;
249     __delay_ms(2);
250     IOUPDATE = 0;
251
252     EEPROMwrite(0x0D, POW1);
253     EEPROMwrite(0x0E, POW2);
254 }

```