

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Číslicový převodník efektivní hodnoty pro kmitočtové pásmo 0 až 2 MHz

Michal Špaček

Vedoucí práce: Ing. Radek Sedláček, Ph.D.
Obor: Kybernetika a robotika
Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Špaček** Jméno: **Michal** Osobní číslo: **466107**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Číslíkový převodník efektivní hodnoty pro kmitočtové pásmo 0 až 2 MHz

Název bakalářské práce anglicky:

Digital Convertor of True RMS Value for Frequency Range 0 up to 2 MHz

Pokyny pro vypracování:

1. Navrhněte a realizujte číslicový převodník pro přímé měření efektivní hodnoty napětí, který bude pracovat v kmitočtovém pásmu 0 až několik jednotek MHz s přesností lepší nežli 0,2 %. Obvodové řešení převodníku založte na použití vhodně zvoleného typu FPGA obvodu.
2. Navrhněte vhodné uživatelské ovládání převodníku uživatelem včetně možnosti vizualizace naměřených dat na displeji.
3. Vytvořte programové vybavení pro FPGA obvod v jazyce VHDL realizující výše požadované funkce.
4. Na převodníku implementujte jednoduché komunikační rozhraní pro připojení k PC z důvodu možnosti vzdáleného ovládání a vyčítání naměřených dat (např. sériový port či Ethernet rozhraní).

Seznam doporučené literatury:

- [1] PEDRONI, Volnei A.: Digital electronics and design with VHDL. Amsterdam: Elsevier Morgan Kaufmann Publishers, c2008. ISBN 978-0123742704.
- [2] PROAKIS, John G. a MANOLAKIS, Dimitris G.: Digital signal processing. 4th ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2007. ISBN 978-0131873742.
- [3] SHENOI, Belle A.: Introduction to digital signal processing and filter design. Hoboken, N.J.: Wiley-Interscience, c2006. ISBN 9780471654421.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Radek Sedláček, Ph.D., katedra měření FEL

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

Datum zadání bakalářské práce: **12.02.2019**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2019/2020

Ing. Radek Sedláček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych zde poděkoval za veškerou pomoc při tvorbě této bakalářské práce. Největší dík patří vedoucímu práce Ing. Radku Sedláčkovi, Ph.D. za jeho pomoc, cenné rady a poskytnutí technického zázemí. Dále patří poděkování také mé rodině, která mě podporovala nejen v průběhu této práce.

Prohlášení

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

V Praze, 24. května 2019

Abstrakt

Cílem této práce je vytvořit kompaktní zařízení, které měří efektivní hodnotu napětí v širokém kmitočtovém pásmu. Důraz je kladen především na přesnost měření v celém zadaném kmitočtovém pásmu. Přidanou funkcionalitou je možnost měřit amplitudu jednotlivých frekvenčních složek signálu. Této možnosti je docíleno implementací algoritmu pro rychlou Fourierovu transformaci. Pro správnou funkci je použita i metoda okénkování za účelem potlačení nežádoucího jevu zvaného spektrální prosakování. Zpracování měřeného signálu je prováděno pomocí FPGA obvodu, na němž jsou některé úkony řešeny procesorem NIOS.

Klíčová slova: FPGA, TrueRMS, FFT, NIOS, okénkování, spektrální prosakování

Abstract

The aim of this project is to develop compact device that can measure effective value of voltage in wide frequency range. Main goal is accurate measurement in desired frequency range. Added functionality is possibility measure amplitude each frequency part of the measured signal. This possibility is reached by implemented algorithm for fast Fourier transformation. For properly measure each amplitude by this way is used method named windowing. This method prevents against unwanted spectral leakage. The signal being processed by FPGA with implemented processor NIOS, witch compute some necessary tasks.

Keywords: FPGA, TrueRMS, FFT, NIOS, windowing, spectral leakage

Title translation: Digital Convertor of True RMS Value for Frequency Range 0 up to 2 MHz

Obsah

1 Úvod	1	4 Experimentální výsledky	43
1.1 Motivace	1	4.1 Měření s využitím multimetru	
1.2 Možné použití pro metodu měření pomocí tří voltmetrů	2	Agilent 3458A	43
1.3 Možné metody řešení	3	4.2 Měření s využitím multimetru	
1.3.1 Přímý výpočet	3	Wavetek 1281	45
1.3.2 Lock-in zesilovač	3	4.3 Zhodnocení naměřených dat	46
1.3.3 Rychlá Fourierova transformace	5	5 Závěr	47
1.4 Zvolená metoda řešení	5	A Vysvětlení použitých zkratk	49
2 Hardwarové vybavení	7	B Literatura	51
2.1 Základní struktura	7		
2.2 Rozbor jednotlivých bloků	7		
2.2.1 Driver pro analogově digitální převodník	7		
2.2.2 Analogově digitální převodník	9		
2.2.3 Programovatelné hradlové pole	11		
2.2.4 Komunikace s nadřazeným zařízením	13		
2.2.5 Zobrazovací zařízení	15		
2.2.6 Ovládací rozhraní	17		
2.2.7 Napájecí obvody	19		
2.3 Nevyužitý návrh	21		
2.4 Poznámky k návrhu desky plošných spojů	22		
2.5 Krabice pro vyvíjené zařízení	24		
2.6 Výsledná podoba	25		
3 Implementace	27		
3.1 FPGA čip	27		
3.1.1 VHDL blok pro zpracování vzorků	28		
3.1.2 VHDL blok tvořící hodinový signál pro A/D převodník	29		
3.1.3 Výsledná podoba celkového návrhu na úrovni VHDL	30		
3.2 Procesor NIOS II	30		
3.2.1 Firmware procesoru NIOS II	32		
3.3 Uživatelská aplikace	36		
3.3.1 Tlačítko Connect	37		
3.3.2 Tlačítko Disconnect	37		
3.3.3 Tlačítko Measure	37		
3.3.4 Přepínač Frequency/Time domain	41		
3.3.5 Tlačítko Export data	41		

Obrázky

1.1 Schéma zapojení při měření impedance cívky metodou tří voltmetrů	2	3.1 Blokové schéma VHDL návrhu	27
1.2 Blokové schéma měření efektivní hodnoty napětí pomocí lock-in zesilovače	3	3.2 Vývojový diagram popisující ukládání dat na úrovni VHDL	29
2.1 Blokové schéma True RMS převodníku	7	3.3 Grafická podoba projektu v prostředí Quartus	30
2.2 Vnitřní blokové schéma rozdílového zesilovače [4]	8	3.4 Snímek konfigurace procesoru NIOS v nástroji Platform Designer	32
2.3 Schéma zapojení obvodu LTC6400-8	9	3.5 Vývojový diagram popisující firmware procesoru NIOS II	33
2.4 Vnitřní blokové schéma analogově digitálního převodníku [3]	9	3.6 Frekvenční charakteristika vstupních obvodů	35
2.5 Schéma zapojení obvodu LTC2204	11	3.7 Grafické rozhraní uživatelské aplikace	36
2.6 Vývojový kit MAX10 (10M08S, 144-EQFP) [10]	12	3.8 Vývojový diagram popisující prováděné úkony k zobrazení naměřených dat	38
2.7 Schéma zapojení vývojového kitu MAX10	13	3.9 Ukázka spektrálního prosakování	39
2.8 Bitová posloupnost sériového přenosu pomocí UART	14	3.10 Časový průběh analyzovaného signálu	40
2.9 Schéma zapojení obvodu FT230XS-R	15	3.11 Porovnání vlivu použité okénkové funkce na výsledek frekvenční analýzy	40
2.10 Schéma zapojení displeje EA DOGL128E-6	17	4.1 Porovnání naměřeného napětí pomocí multimetru Agilent 3458A a našeho zařízení	44
2.11 Schéma zapojení spínačů	17	4.2 Procentuální odchylka naměřeného napětí pomocí multimetru Agilent 3458A a našeho zařízení	44
2.12 Frekvenční charakteristika filtru výstupu tlačítek	18	4.3 Porovnání naměřeného napětí multimetrem Wavetek 1281 a naším zařízením při frekvenci 10 kHz	45
2.13 Schéma zapojení ideálních diod	19	4.4 Procentuální odchylka naměřeného napětí multimetrem Wavetek 1281 a naším zařízením při frekvenci 10 kHz	46
2.14 Schéma zapojení obvodu LMZ21701SILT	20		
2.15 Schéma zapojení obvodu LT1763CS8-3.3#PBF	20		
2.16 Schéma zapojení Schmittova klopného obvodu SN74LVC1G17	21		
2.17 Schéma zapojení vstupní desky	22		
2.18 3D model desky plošných spojů	23		
2.19 3D model krabičky	24		
2.20 Skutečná podoba DPS z vrchní strany	25		
2.21 Skutečná podoba DPS ze spodní strany	25		
2.22 Skutečná podoba zařízení ze zadní strany	26		
2.23 Skutečná podoba zařízení z přední strany	26		

Tabulky

1.1 Parametry vybraných měřicích přístrojů	1
3.1 Parametry definované entity	28
3.2 Přehled parametrů jednotlivých výkonových verzí procesoru NIOS II [5]	31
3.3 Parametry testovaného signálu .	39
3.4 Výsledek frekvenční analýzy signálu pro složku o frekvenci 400 kHz	41
4.1 Přehled chyb měření multimetru Agilent 3458A pro napěťový rozsah 100 mV až 10 V [1]	43
4.2 Přehled nejistot měření multimetru Wavetek 1281 pro napěťový rozsah 100 mV [14]	45
4.3 Přehled nejistot měření multimetru Wavetek 1281 pro napěťový rozsah 1 V až 100 V [14]	45
A.1 Význam zkratk použitých v této práci	49

Kapitola 1

Úvod

1.1 Motivace

V některých aplikacích, jako je například metoda měření impedance pomocí tří voltmetrů [12] či v případě měření impedancí pomocí koaxiálního můstku [11], je požadováno přesné měření velikosti napětí v určitém frekvenčním pásmu. Při pohledu na trh s měřicí elektronikou a přístrojovou technikou lze vidět, že je nabízeno nepřeborné množství měřicích přístrojů určujících velikost napětí měřeného signálu. Avšak při detailnějším zkoumání u většiny těchto zařízení narazíme na frekvenční omezení v řádu několika stovek kHz. Při měření signálů o vyšší frekvenci, již výrobce negarantuje chybu měření. Pro názornost je pár vybraných měřicích přístrojů uvedeno v tabulce 1.1.

Výrobce	Typ	Maximální frekvenční rozsah [kHz]	Chyba měření v daném frekvenčním rozsahu	Běžná cena
B&K Precision	5492B	100 - 300	3% + 0.1%	cca 19 000 Kč
Tektronix	DMM6500	100 - 300	4% + 0.5%	cca 25 000 Kč
Keysight	34461A	100 - 300	4% + 0.5%	cca 29 000 Kč
Keysight	34470A	100 - 300	1% + 0.1%	cca 79 000 Kč

Tabulka 1.1: Parametry vybraných měřicích přístrojů

Ve čtvrtém sloupci první člen součtu odpovídá chybě z naměřené hodnoty a druhý člen chybě ze zvoleného rozsahu měření.

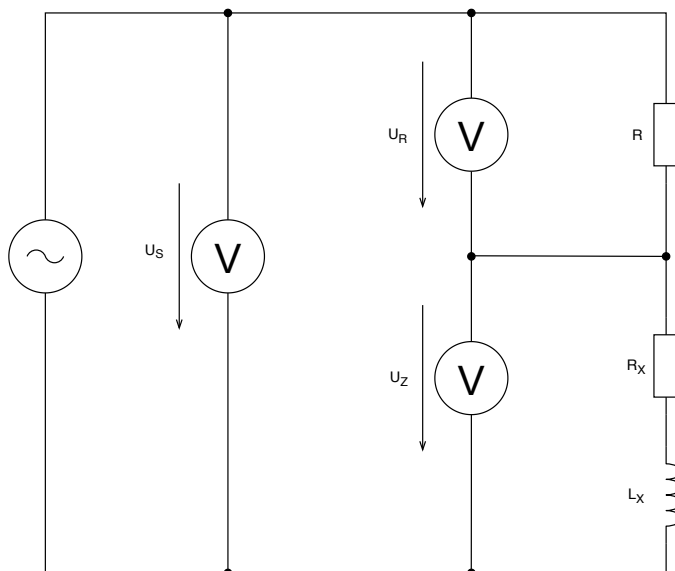
I v případě, že opomeneme nízký frekvenční rozsah měřicích přístrojů, tak měření v rádech stovek kHz je zatíženo poměrně velkou chybou.

Cílem této práce je tedy vytvořit zařízení, které bude vhodnou náhradou za běžně nabízené měřicí přístroje pro měření efektivní hodnoty napětí především na frekvencích v rádech stovek kHz až několik jednotek MHz. Dle zadání by chyba měření měla být lepší jak 0.2%. Pro potlačení rušivých složek signálu, které se v měřeném signálu mohou vyskytovat, je zamýšleno použít algoritmu rychlé Fourierovy transformace s použitím vhodné okénkové funkce pro potlačení spektrálního prosakování. Toho lze využít například v již zmíněné metodě tří voltmetrů, kde nám tento přístup umožní získat

informaci o efektivní hodnotě napětí bez rušivých složek majících jiný než měřený kmitočet.

1.2 Možné použití pro metodu měření pomocí tří voltmetrů

Jak již bylo uvedeno, výsledné zařízení by se s výhodou mohlo využít k měření neznámé impedance metodou tří voltmetrů. Na obrázku 1.1 lze vidět schématické zapojení obvodu pro měření impedance cívky metodou tří voltmetrů. V obvodu se nachází zdroj střídavého napětí o dané frekvenci a amplitudě. Známý rezistor R a cívka s neznámou impedancí, která je ve schématu zakreslena náhradním obvodem. Náhradní obvod tvoří cívka L_x a rezistor R_x . U obou prvků jejich parametry neznáme.



Obrázek 1.1: Schéma zapojení při měření impedance cívky metodou tří voltmetrů

Změříme uvedená napětí U_s , U_R a U_Z . Z napětí U_R a U_Z určíme celkovou impedanci cívky dle vztahu 1.1.

$$Z = R \cdot \frac{U_Z}{U_R}. \quad (1.1)$$

Vztahem 1.2 určíme fázový posun.

$$\varphi = \arccos \left(\frac{U_s^2 - U_R^2 - U_Z^2}{2 \cdot U_R \cdot U_Z} \right). \quad (1.2)$$

Nyní již snadno určíme odpor rezistoru R_x vztahem 1.3 a indukčnost cívky L_x vztahem 1.4.

$$R_X = |Z| \cdot \cos(\varphi). \quad (1.3)$$

$$L_X = \frac{|Z| \cdot \sin(\varphi)}{2 \cdot \pi \cdot f}. \quad (1.4)$$

1.3 Možné metody řešení

1.3.1 Přímý výpočet

Efektivní hodnota střídavého napětí, často označována jako RMS hodnota střídavého napětí, je definována následovně. Efektivní hodnota střídavého napětí je rovna hodnotě stejnosměrného napětí, které by za stejný čas na odporové zátěži vyvolalo stejné výkonové účinky.

Tuto definici lze také vyjádřit následujícím vztahem 1.5.

$$U_{ef} = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt}, \quad (1.5)$$

kde $u(t)$ je hodnota napětí v daném čase t a T je perioda signálu.

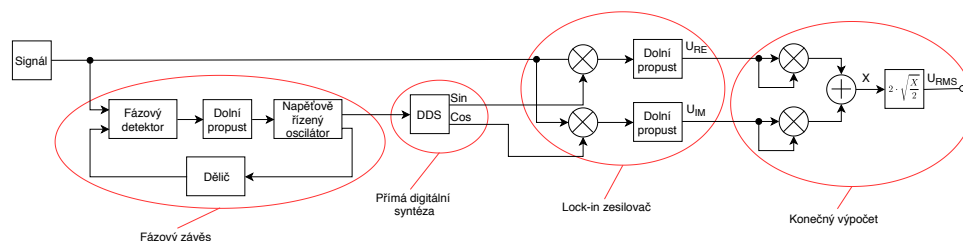
Přepsáním vztahu 1.5 do jeho diskretní podoby 1.6 získáme možný způsob, jak stanovit RMS hodnotu měřeného signálu.

$$U_{ef} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} u^2(n)}, \quad (1.6)$$

kde $u(n)$ je n -tá hodnota napětí navzorkovaného signálu a N je celkový počet vzorků.

1.3.2 Lock-in zesilovač

S využitím lock-in zesilovače bychom byli schopni měřit efektivní hodnotu napětí bez rušivých složek měřeného signálu. Avšak implementace lock-in zesilovače na hradlovém poli není zcela triviální. Na obrázku 1.2 je blokové schéma zamýšlené implementace této metody.



Obrázek 1.2: Blokové schéma měření efektivní hodnoty napětí pomocí lock-in zesilovače

■ Fázový závěs

Fázovým závěsem je označována struktura čtyř bloků, jejichž zapojení je zakroužkované v obrázku 1.2. První z bloků je fázový detektor, který porovnává rozdíl fáze vstupního signálu a signálu přicházející ve zpětné vazbě. Dolní propust upravuje výstupní signál z fázového detektoru. Napětově řízený oscilátor generuje signál o frekvenci odpovídající výstupnímu napětí z dolnofrekvenčního filtru. Tento signál je přes dělič ve zpětné vazbě přiveden zpět na vstup fázového detektoru. Za předpokladu, že fáze vstupního a generovaného signálu si jsou rovny, bude frekvence obou signálů stejná. Toto však platí pouze v případě, že dělič ve zpětné vazbě nijak nezmění frekvenci signálu. Tedy, že jeho dělicí poměr bude roven jedné. Pokud by snížil frekvenci signálu na polovinu, výstupní frekvence bude odpovídat dvojnásobku té vstupní. Tímto způsobem bychom tedy mohli generovat různé násobky vstupní frekvence. Hodnotu výstupní frekvence můžeme vyjádřit vztahem 1.7.

$$f_2 = \frac{f_1}{N}, \quad (1.7)$$

kde f_2 je výstupní frekvence, f_1 je vstupní frekvence a N je koeficient děliče.

■ Přímá digitální syntéza

V obrázku 1.2 blok s označením DDS představuje generování signálu o sinusovém a kosinusovém průběhu. Přičemž kosinusový průběh je vytvořen pouze fázovým posunem toho sinusového o $\frac{\pi}{2}$. Použití přímé digitální syntézy přináší v první řadě větší frekvenční stabilitu, možnost poměrně jednoduchým způsobem měnit frekvenci generovaného signálu a dle implementace lze získat frekvenční rozlišení až v řádu μHz .

■ Synchronní detektor

Slouží ke stanovení velikosti reálné a imaginární složky měřeného signálu. Z toho lze následně velice snadno spočítat amplitudu a fázový posun signálu.

$$U_A = 2 \cdot \sqrt{U_{Re}^2 + U_{Im}^2} \quad (1.8)$$

$$\varphi = \arctan\left(\frac{U_{Im}}{U_{Re}}\right) \quad (1.9)$$

■ Konečný výpočet

V tomto posledním kroku stanovíme efektivní hodnotu měřeného signálu. K tomu využijeme výše zmíněného vztahu 1.8.

$$U_{RMS} = \frac{U_A}{\sqrt{2}} = 2 \cdot \sqrt{\frac{U_{Re}^2 + U_{Im}^2}{2}} \quad (1.10)$$

Vztahem 1.10 získáme efektivní hodnotu signálu odpovídající signálu o stejné frekvenci, jako má referenční signál. Tedy změnou frekvence referenčního signálu můžeme měnit námi zkoumanou frekvenční oblast.

■ 1.3.3 Rychlá Fourierova transformace

■ Fourierova transformace

Fourierova transformace představuje rozložení funkce na součet funkcí sinus a kosinus. Daný zkoumaný signál převádí z časové oblasti do oblasti frekvenční. Tuto transformaci můžeme vyjádřit následujícím vztahem 1.11.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt, \quad (1.11)$$

kde $f(t)$ je funkce v časové oblasti a $F(\omega)$ je obrazem funkce $x(t)$ ve frekvenční oblasti. [8] V případě, že máme funkci popsanou posloupností, tedy se jedná o funkci diskretní, můžeme Fourierovu transformaci vyjádřit vztahem 1.12. V takovém případě již mluvíme o diskretní Fourierově transformaci.

$$F[k] = \sum_{n=0}^{N-1} f[n] \cdot e^{-\frac{2\pi j k n}{N}}, \quad k = 0, 1, \dots, N-1, \quad (1.12)$$

kde $f[n]$ je posloupnost a $F[k]$ rozklad posloupnosti $f[n]$ do Fourierovy řady. [8]

Využitím diskretní Fourierovy transformace můžeme získat informaci o míře zastoupení rušivých signálů o různých frekvencích a amplitudách v našem měřeném signálu. Avšak přímý výpočet spektra dle definice DFT je výpočetně velice náročný. Výpočetní složitost je $O(N^2)$.

■ Rychlá Fourierova transformace

Rychlá Fourierova transformace je efektivnější způsob, jak provést diskretní Fourierovu transformaci. Speciálně algoritmus Cooley-Tukey má výpočetní náročnost $O(N \cdot \log(N))$. Podmínkou pro užití tohoto algoritmu je daná délka vstupní posloupnosti. Ta musí být beze zbytku dělitelná mocninou dvou, tedy $N = 2^m$.

■ 1.4 Zvolená metoda řešení

Po provedení simulací a zvážení výhod či nevýhod výše zmíněných přístupů byla vybrána kombinace přímého výpočtu RMS hodnoty napětí a rychlé Fourierovy transformace. Tato cesta je, jak implementačně, tak časově výhodnější. Funkce, kterou zastává rychlá Fourierova transformace, by mohla být řešena i pomocí lock-in zesilovače. Odstranili bychom tím vliv spektrálního prosakování. Avšak implementace takového systému je podstatně náročnější. I toto je jeden z důvodů, proč v konečné fázi byla zvolena rychlá Fourierova

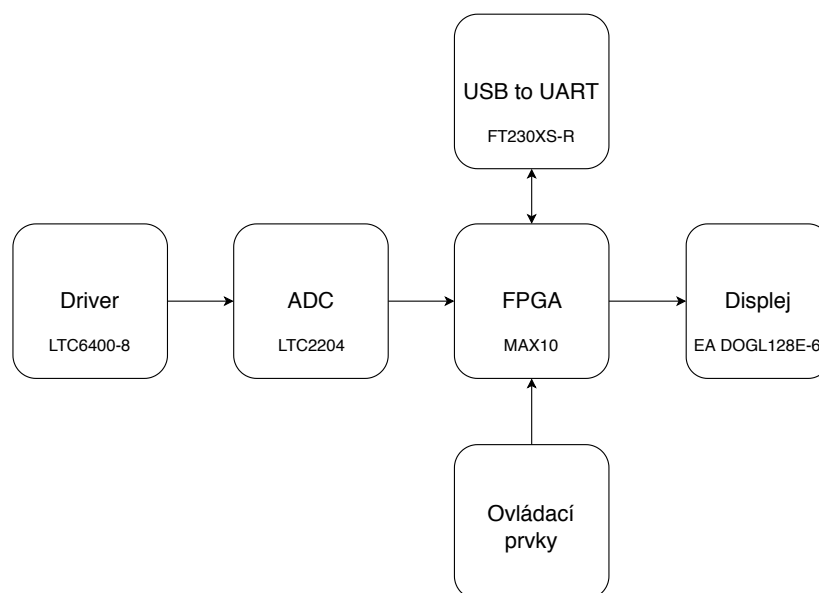
transformace. Nicméně návrh celého zařízení je koncipován s ohledem na případnou modifikaci řešení. Tedy po hardwarové stránce je zařízení připraveno pro implementaci zmíněného lock-in zesilovače.

Kapitola 2

Hardwarové vybavení

2.1 Základní struktura

Pro dosažení uspokojivé spolehlivosti a přesnosti měřicího zařízení je nepochybně klíčový správný hardwarový návrh celého systému. Blokové schéma na obrázku 2.1 představuje základní strukturu našeho zařízení spolu s typovým označením konkrétních prvků, které byly použity k realizaci celého hardwarového řešení.



Obrázek 2.1: Blokové schéma True RMS převodníku

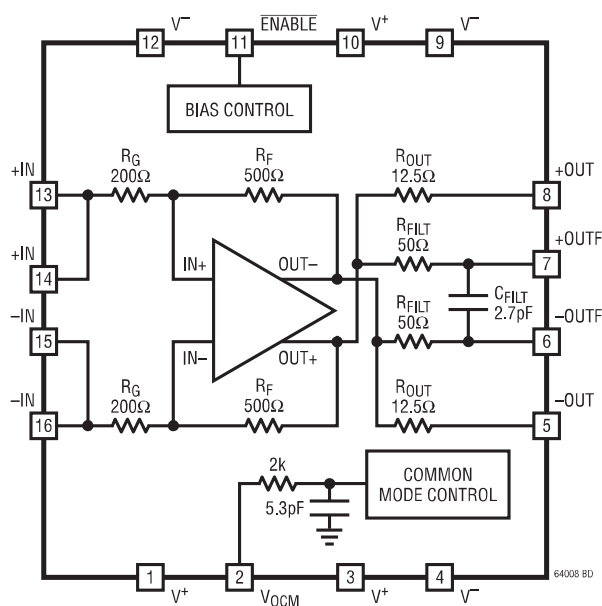
2.2 Rozbor jednotlivých bloků

2.2.1 Driver pro analogově digitální převodník

Tato část celkového řešení je na obrázku 2.1 označena jako Driver. Slouží jako impedanční přizpůsobení pro A/D převodník, upravuje amplitudu vstupního signálu a může sloužit jako jakási ochrana před připojením signálů,

které by mohly mít za následek destrukci A/D převodníku. Konvertuje jednovodičový vstup signálu na diferenční výstup, což je výhodné především pro zvýšení odolnosti vůči rušivým vlivům signálu. V neposlední řadě posouvá signál o úroveň napětí V_{OCM} . Hodnota napětí V_{OCM} by měla být 1.25 V.

V našem případě byl zvolen obvod LTC6400-8. Jedná se o diferenční operační zesilovač s fixně nastaveným zesílením. Na trhu lze nalézt typově stejné obvody s rozdílným fixně nastaveným zesílením. Námi zvolený obvod má zesílení 8 dB, což vstupní signál zesílí zhruba 2.5 krát. Vnitřní strukturu integrovaného obvodu lze vidět na obrázku 2.2.



Obrázek 2.2: Vnitřní blokové schéma rozdílového zesilovače [4]

Na obrázku 2.2 vidíme rezistory R_G a R_F . Těmito rezistory je nastaveno fixní zesílení obvodu operačního zesilovače dle vztahu 2.1.

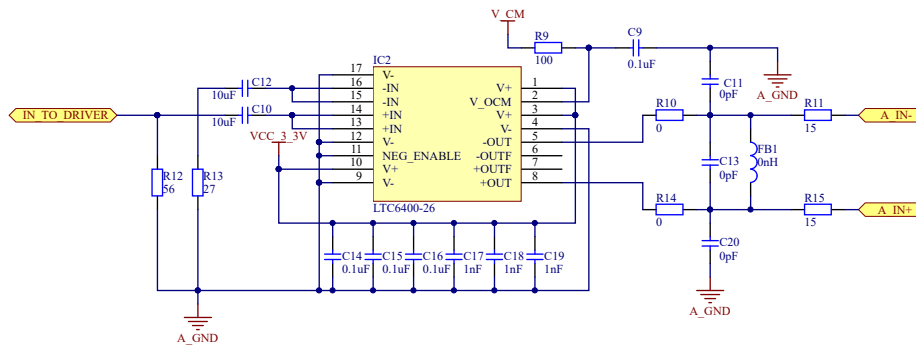
$$A = \frac{R_F}{R_G} \quad (2.1)$$

Dále je u tohoto obvodu možnost využití výstupu s filtrem typu dolní propust. Dle výrobce má tento filtr mezní kmitočet na frekvenci 590 MHz. Tedy, že při této frekvenci bude útlum filtračního článku -3 dB. Se záměrem ponechání určité variability nastavení výstupního filtru, jsme zmíněný integrovaný filtr nevyužili. V případě potřeby lze pomocí pinu 11 uvést obvod do úsporného režimu, ve kterém v nejhorším možném případě odebírá ze zdroje proud o velikosti 3 mA.

Obvod je napájen nesymetricky. Úroveň napájecího napětí je 3.3 V. Při této úrovni napětí by dle výrobce neměl odebíraný proud v extrémním případě přesáhnout úroveň 0.95 mA. Diferenční vstupní impedance obvodu je 400 Ω . Z důvodu této poměrně nízké vstupní impedance není příliš vhodné využívat

tento obvod bez nějakého impedančního oddělovače ve formě, například neinvertujícího operačního zesilovače se zavedenou jednotkovou zápornou vazbou.

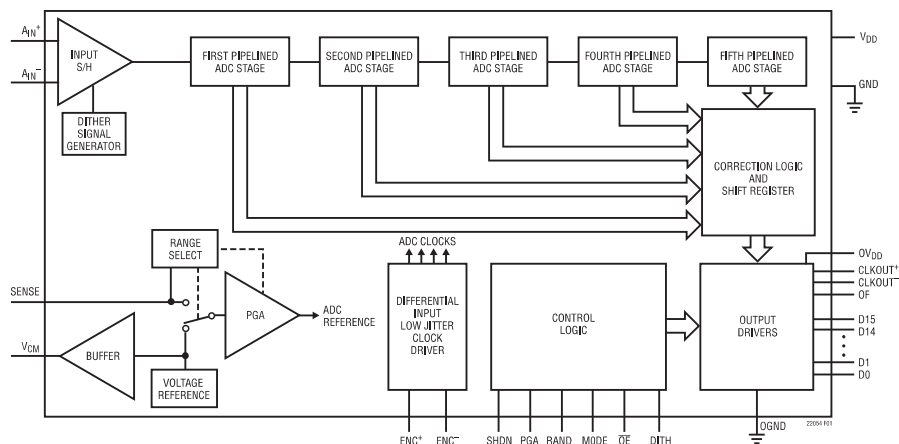
Schéma zapojení použité v návrhu lze vidět na obrázku 2.3. Jedná se o typické zapojení uváděné v dokumentaci k tomuto obvodu. V tomto zapojení by se měla vstupní impedance blížit hodnotě 50Ω . [4]



Obrázek 2.3: Schéma zapojení obvodu LTC6400-8

2.2.2 Analogově digitální převodník

Na obrázku 2.1 je tento komponent zastoupen blokem s názvem ADC. Jedná se o vzorkovací obvod, který digitalizuje analogový signál. Tedy provádí vzorkování v čase a kvantování v amplitudě. U A/D převodníků nás především zajímají dva následující parametry. Počet bitů, pomocí kterých je schopen aproximovat úroveň vstupního signálu a rychlost vzorkování, která udává, na kolik diskretních částí bude signál rozdělen za nějaký časový úsek. S ohledem na dodržení vzorkovacího teorému jsme do našeho návrhu vybrali obvod LTC2204. Blokové zapojení výše zmíněného obvodu je na obrázku 2.4.



Obrázek 2.4: Vnitřní blokové schéma analogově digitálního převodníku [3]

Jedná se o 16-ti bitový převodník se zřetězenou architekturou. Zřetězení je rozděleno do pěti stupňů, které je následně vyhodnoceno vnitřní logikou. V každém stupni je signál digitalizován, část digitálního slova je využita vyhodnocovací logikou. Digitalizovaný signál je opět převeden pomocí D/A převodníku na analogový signál a následně je signál upraven pro další vyhodnocovací stupeň.

Obvod disponuje diferenčním vstupem signálu, který, jak již bylo zmíněno dříve, je výhodný zejména z pohledu odolnosti vůči okolnímu rušení. Pomocí pinu PGA lze měnit zesílení vstupního signálu před samotnou digitalizací. Logická úroveň 1 nastaví zesílení na 0 dB, logická úroveň 0 nastaví zesílení na zhruba 3.5 dB. Pinem SENSE je volen zdroj referenčního napětí pro převodník. Připojením tohoto pinu na napájecí úroveň převodníku 3.3 V zvolíme interní referenční zdroj typu bandgap o úrovni 2.5 V. Na pin označený jako V_{CM} je generováno napětí pro předcházející obvod upravující zpracovávaný signál. Právě o úroveň tohoto napětí V_{CM} je signál posunut.

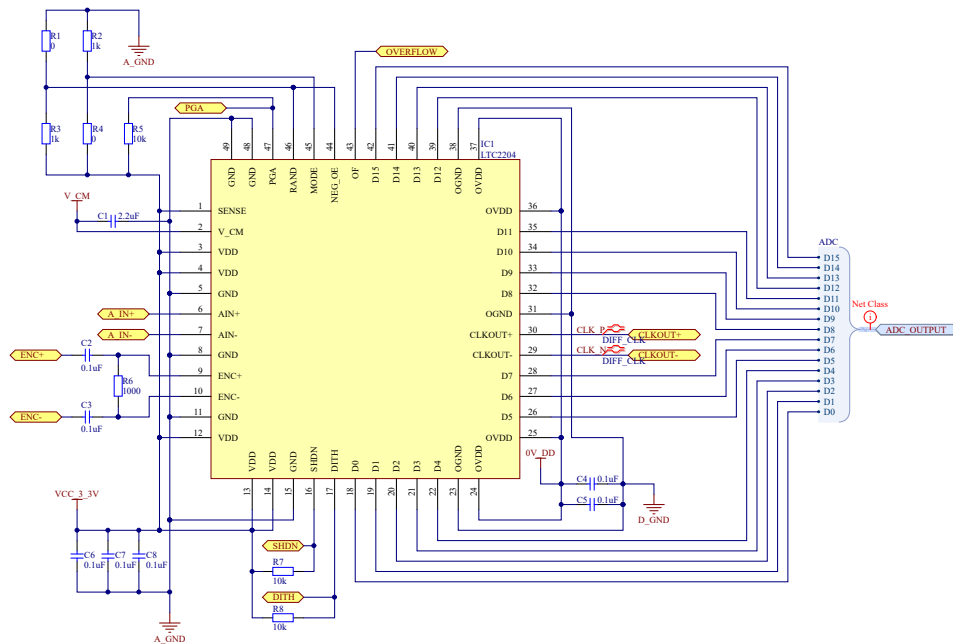
Na převodníku dále nalezneme dva diferenční páry hodinového signálu. Diferenční pár označený ENC je vstup, kterým řídíme vzorkování převodníku. S náběžnou hranou signálu ENC^+ a sestupnou hranou ENC^- je odebrán vzorek vstupního analogového signálu. Naopak diferenční pár CLKOUT představuje indikaci dostupnosti nového binárního slova. Tedy v případě, že na výstupu $CLKOUT^+$ se objeví vzestupná hrana a na $CLKOUT^-$ sestupná hrana, můžeme aktuální stav výstupu D0 až D15 považovat za platný a uložit si tento stav do paměti pro další zpracování. Výrobce uvádí, že by frekvence hodinových signálů měla být v rozmezí 1 MHz až 40 MHz.

Výstupní digitální brána má své vlastní oddělené napájení, které je v našem návrhu realizováno vyhrazeným regulátorem pouze pro tento účel.

Dále lze obvod uvést do úsporného režimu pomocí pinu SHDN. Při logické 0 je obvod v pracovním režimu. Naopak při přivedení logické 1 se odstaví napájení analogových obvodů a digitální výstup se uvede do stavu vysoké impedance. V tomto stavu výrobce deklaruje spotřebu typicky 0.2 mW. Obdobnou funkci zastupuje pin \overline{OE} s tím rozdílem, že stav tohoto pinu neovlivní funkci analogových obvodů, ale pouze digitální výstup.

AD převodník je napájen ze spínaného regulátoru napětí, který generuje na svůj výstup napětí o velikosti 3.3 V. Při tomto napětí výrobce slibuje, že odebíraný proud ze zdroje bude typicky 145 mA. Převodník poskytuje vzorkovací frekvenci až 40 Msps. Pro naše účely je vzorkovací frekvence obvodu LTC2204 dostačující. V případě potřeby je zde možnost záměny za obvod s označením LTC2205, který má vzorkovací frekvenci 65 Msps. [3]

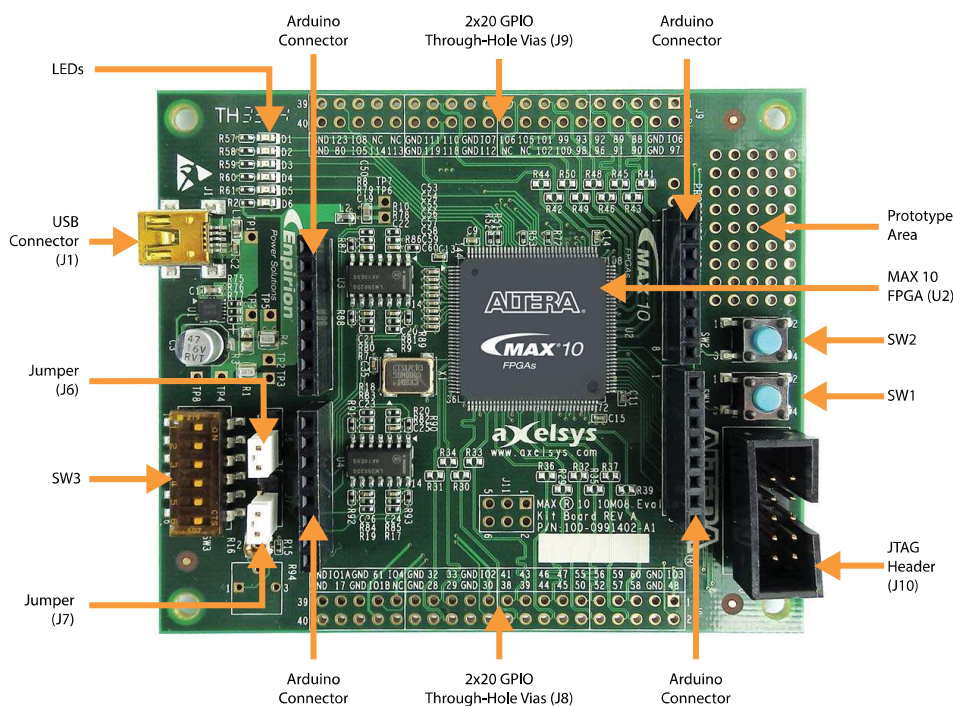
Na obrázku 2.5 nalezneme schéma zapojení výše popisovaného obvodu LTC2204.



Obrázek 2.5: Schéma zapojení obvodu LTC2204

2.2.3 Programovatelné hradlové pole

Této části návrhu náleží na obrázku 2.1 blok s názvem FPGA. Zde jsme se rozhodli jít cestou již hotového vývojového kitu. Samotný kit je vyobrazen na následujícím obrázku 2.6. Tato část je srdcem celého zařízení. Z pohledu těch nejzákladnějších úkonů spravuje ukládání digitalizovaných vzorků signálu do paměti, řídí rychlost vzorkování, zajišťuje komunikaci s případným nadřazeným systémem v podobě například počítače, vyčítá potřebné informace na displej a přijímá pokyny z uživatelského rozhraní v podobě tlačítek.



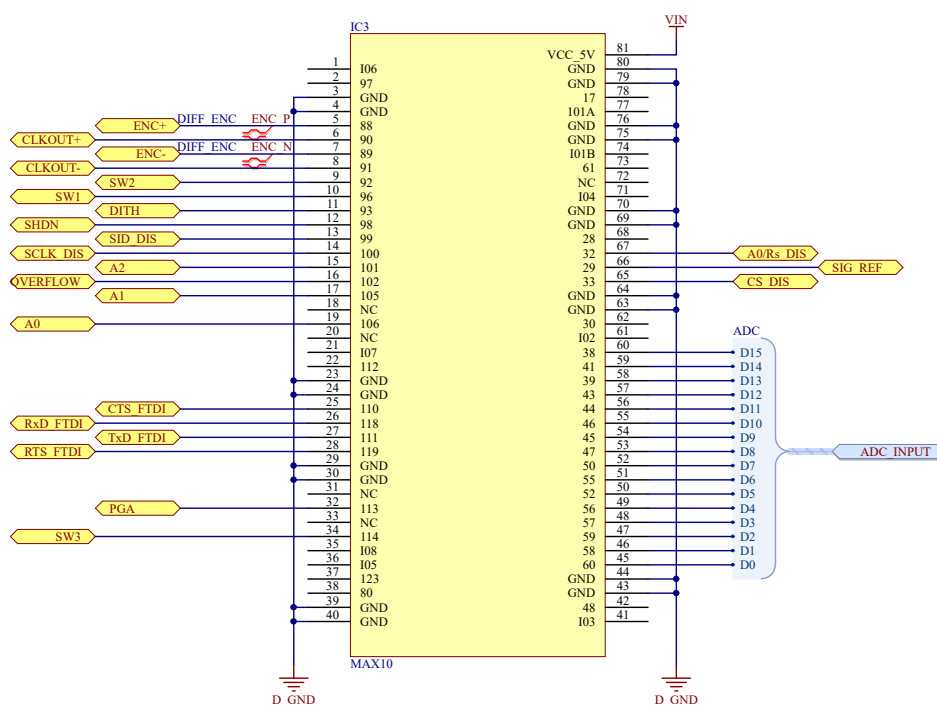
Obrázek 2.6: Vývojový kit MAX10 (10M08S, 144-EQFP) [10]

Výše zmiňovaný vývojový kit má na své desce umístěny indikační, ovládací a různé další prvky, které usnadňují vývoj projektu. Na desce nalezneme dvě tlačítka na obrázku 2.6 označená jako SW1 a SW2. Tlačítko SW2 slouží k resetu zařízení a tlačítko SW1 je ponecháno pro konfiguraci uživatelem. Taktéž přepínače SW3 nemají definovanou žádnou výchozí funkci. Přítomnost arduino pinů zajišťuje kompatibilitu s různými moduly, jako například ethernet modul, Wi-Fi modul, řadič pro řízení motorů a mnoho dalších, které jsou na trhu k dostání. V levém horním rohu se nachází 6 LED diod, přičemž prvních 5 LED diod emitují červené světlo. Jsou volně konfigurovatelné uživatelem. Poslední LED dioda vyzařuje zelené světlo a slouží jako indikace přítomnosti napájení kitu. USB konektor označený J1 slouží pouze k napájení celé desky. Datové piny tohoto konektoru nejsou připojeny k žádnému dalšímu zařízení na kitu. Patice JTAG označena J10 slouží k programování FPGA čipu MAX10. Pro tuto činnost je potřeba použít externí programovací jednotku s obchodním názvem USB-BLASTER. Dále můžeme na desce nalézt dva jumpery, kterými měníme nastavení analogového vstupu 7 a 8. V horní a dolní části vývojového kitu jsou umístěny pájecí piny s označením J8 a J9, na které jsou mimo jiné připojeny také piny FPGA čipu. S využitím těchto pájecích pinů a propojů je kit spojen s navrženou deskou. Na kitu je samozřejmě přítomný i samotný již zmiňovaný čip MAX10 (10M08SAE144C8G).

Pro napájení samotného vývojového kitu je potřeba napětí 5 V. Proudový odběr je velice závislý na implementované vnitřní struktuře FPGA čipu. Pro přívod napájení na desku je využít testovací pin kitu.

Vlastní FPGA obvod MAX10 má logické úrovně 0 a 3.3 V. Tyto logické úrovně jsou zajištěny připojením I/O bran čipu na napájecí napětí 3.3 V. Připojení je fyzicky realizováno pomocí nulových odporů na desce. Tedy v případě potřeby je možné tyto nulové odpory odstranit, tím odpojit stávající napájecí úroveň bran a připojit na piny k tomu určené externí napájení. Tyto piny se nacházejí mezi soustavou již dříve zmíněných pájecích pinů J8 a J9. Čip má ve svém pouzdře integrováno 8000 logických celků. Vnitřní paměť je typu M9K o velikosti 378 kbit. Také zde nalezneme uživatelskou flash paměť o velikosti 172 kbit. Zdrojem hodinového signálu je externí oscilátor o kmitočtu 50 MHz. [10]

Obrázek 2.7 představuje schéma zapojení výše popisovaného vývojového kitu do našeho návrhu.



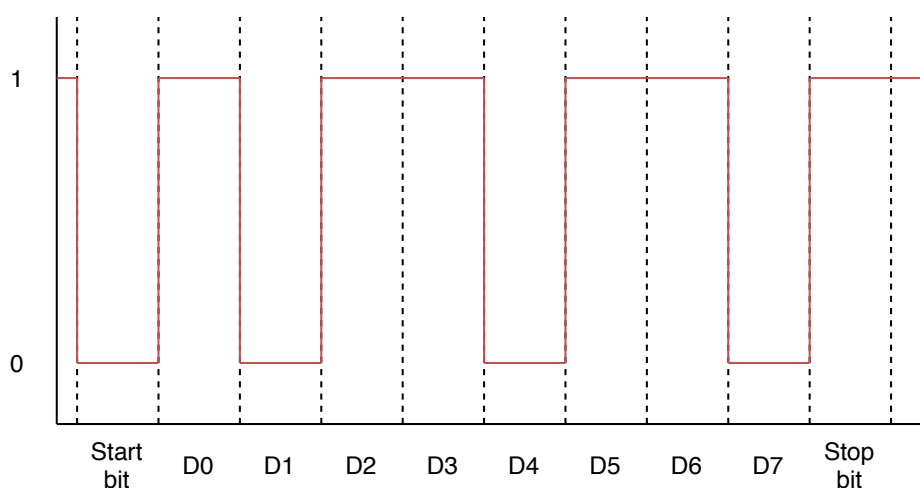
Obrázek 2.7: Schéma zapojení vývojového kitu MAX10

2.2.4 Komunikace s nadřazeným zařízením

Komunikační rozhraní mezi naším zařízením a nadřazeným systémem je na obrázku 2.1 naznačeno blokem s názvem USB to UART. Pro vytvoření komunikačního spojení je fyzicky použit obvod FT230XS-R. Tento obvod, jak již název příslušného bloku napovídá, zajišťuje funkční konverzi mezi rozhraní USB 2.0 a UART. Obvod může být napájen i externě pomocí připojeného USB kabelu. V návrhu je tato možnost využita nejen k napájení samotného obvodu, ale k napájení celého zařízení. Mimo jiné obvod disponuje i možností indikace probíhající komunikace pomocí dvou pinů. Připojením pinů, například k LED diodám, můžeme zajistit optickou indikaci uživateli

o právě probíhající komunikaci. Jedná se o plně duplexní komunikaci. Mluvíme tedy o indikaci přijímaných i vysílaných dat. V našem zapojení jsou do čipu, ze strany rozhraní USB 2.0, připojeny dva datové vodiče. Pozitivní a negativní část datového vedení. Jedná se o poloduplexní komunikaci. Naopak ze strany UART jsou ke komunikaci využity dva vodiče. Jeden vodič je určený pro vysílání datových rámců, druhý pro jejich přijímání. Fyzicky tedy využíváme dva vodiče pro plně duplexní komunikaci. [7]

Na straně našeho zařízení jsou datové rámce přenášeny do komunikačního převodníku pomocí UART. Jedná se o asynchronní sériovou komunikaci.



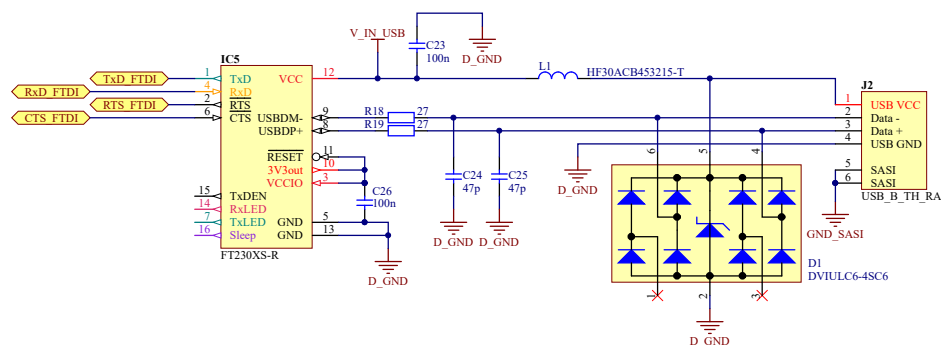
Obrázek 2.8: Bitová posloupnost sériového přenosu pomocí UART

Obrázek 2.8 zachycuje bitovou posloupnost odesílaného datového rámce pomocí UART. Tento konkrétní průběh představuje odesílání znaku „m“. Tomuto znaku v ASCII tabulce náleží dekadická hodnota 109. V binární soustavě toto číslo zapíšeme jako „01101101“ s vědomím, že poslední bit je v obrázku 2.8 reprezentován pod označením D0. Pokud opomeneme datovou část, tak každý odesílaný datový rámec má vždy jasný předem nastavený tvar. Vysílání každého nového bajtu začíná tzv. „Start bitem“. Na vysílacím datovém vodiči dojde ke změně z vysoké logické úrovně na nízkou. Jak již bylo zmíněno, jedná se o asynchronní komunikaci. Tímto způsobem je tedy příjemce informován o začínajícím vysílání jednotlivých bitů. Tento počáteční úkon bychom mohli považovat za synchronizační impuls pro příjemce. Následně přichází přenášení samotných dat ve formě jednotlivých bitů. Nejnižší bit se přenáší jako první. Úroveň každého bitu je na datovém vodiči držena konstantní dobu. Tato doba je dána zvolenou komunikační rychlostí. Rychlost je zde udávána v baudech. Představuje počet bitů, které přeneseme za jednu sekundu. Nejběžněji používané rychlosti jsou 9600 Bd a 115200 Bd. Po datové části následuje tzv. „Stop bit“, který opět příjemci indikuje konec vysílání. Případně je možné ještě před „Stop bit“ vložit tzv. „Parity bit“. Paritní bit může sloužit jako velice jednoduchá kontrola přijatých dat. Sudá či lichá parita indikuje sudý či lichý počet jedniček v datovém slově. Tedy například

v našem výše vyobrazeném datovém slově se nachází pět jedniček. Pokud bychom zvolili sudou paritu, museli bychom paritní bit nastavit do jedničky. Poté by již součet všech jedniček v datovém slově spolu s paritním bitem byl sudý. Příjemce má poté možnost udělat taktéž kontrolní součet. V případě, že by jeho součet vyšel lichý, prohlásil by přijatá data za poškozená. V opačném případě data prohlásí za korektně přijatá. To avšak také nemusí být správně. Může dojít k situaci, kdy vznikne sudý počet chyb. V takovém případě nejsme touto metodou schopni zmíněnou situaci detekovat.

Nastavení komunikace pomocí UART je do jisté míry variabilní, jak bylo popsáno výše. Je však nutné, aby toto nastavení bylo známo nejen na straně odesílatele, nýbrž i na straně příjemce. V opačném případě nebudou přijímaná data reprezentována korektně.

Zapojení obvodu FT230XS-R nalezneme na následujícím obrázku 2.9.



Obrázek 2.9: Schéma zapojení obvodu FT230XS-R

2.2.5 Zobrazovací zařízení

Blok s označením Displej představuje na obrázku 2.1 zobrazovací zařízení určené k vizualizaci důležitých údajů a informací uživateli. Na trhu se zobrazovací technikou lze nalézt nepřeberné množství různých typů displejů. Mohou se lišit technologií zobrazovacího panelu, napájecí úrovní, typem datové sběrnice, případně i zobrazovacími možnostmi. Nejčastěji se setkáváme s těmito kombinacemi:

- Technologie zobrazovacího panelu
 - LCD
 - TFT
 - OLED
- Typ datové sběrnice
 - SPI
 - I2C
 - paralelní sběrnice

- Zobrazovací možnosti
 - matice pixelů
 - segmentový
 - alfanumerický
- Napájecí napětí
 - 5 V
 - 3.3 V

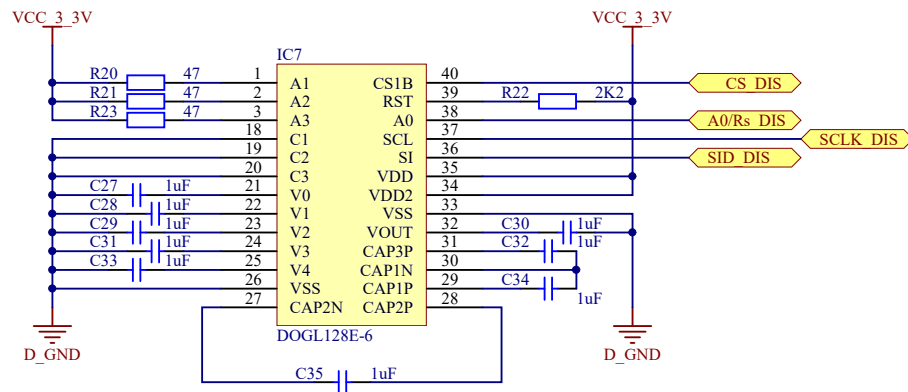
V našem případě byl zvolen displej EA DOGL128E-6. Zobrazovací panel je typu LCD. Funkce této technologie je založena na polarizaci světla. Jedná se o dva na sebe ortogonální polarizační filtry, mezi kterými je vrstva tekutých krystalů. Za touto soustavou se navíc nachází odrazivá plocha či případně zdroj světla. V případě, že na krystaly nepůsobí vnější elektrické pole, zajišťují tekuté krystaly svým spirálovitým uskupením rotaci polarizovaného světla o 90° , a tím umožňují světlu projít skrz oba polarizační filtry. Působením elektrického pole na krystaly výše popsaný děj přestane platit. Ze spirálovitého uskupení přejdou do uspořádané konfigurace. Nedochozí tedy k rotaci polarizovaného světla. Světlo procházející prvním z filtrů již neprojde filtrem druhým. Elektrickým polem lze tedy řídit, zda-li se světlo z nějaké lokální části panelu dostane do vnějšího prostředí či nedostane.

Panel je tvořen pixely s rozlišením 128×64 . Dále je podsvícen zeleně svítícími LED diodami pro lepší viditelnost. Napájen je úrovní 3.3 V. Pro samotné podsvícení je potřeba zajistit nižší napájecí úroveň a to konkrétně 2.1 V.

Pro komunikaci s procesorem je zde použita sběrnice SPI. Standardně má sběrnice SPI čtyři aktivní vodiče nazývané MOSI, MISO, CLK a SS. Vodič MOSI slouží k posílání dat podřízenému zařízení, MISO naopak k příjmu dat od podřízeného zařízení, CLK představuje hodinový signál a vodič SS k selekci podřízeného zařízení. Signálový vodič SS nabývá plného smyslu při připojení více jak jednoho zařízení na jednu SPI sběrnici. V takovém případě je žádoucí vybrat, se kterým z připojených zařízení chceme v danou chvíli komunikovat. V případě využití obou datových vedení se tedy jedná o plně duplexní vedení.

V našem konkrétním případě používáme sběrnici SPI pro komunikaci s displejem. Naším cílem je tedy data pouze zapisovat, nikoliv číst. Z tohoto důvodu u použitého displeje datový vodič MISO ani nenajdeme. V takovém případě se jedná o pouze poloduplexní vedení. Místo něj však nalezneme vodič výrobcem značený jako A0. Jeho logickou úroveň definujeme, zda-li posíláme data či instrukce. Dle výrobce je řadič displeje schopen pracovat s hodinovým signálem o frekvenci až 20 MHz. [6]

Konkrétní zapojení displeje DOGL128E-6 je zachyceno na obrázku 2.10.

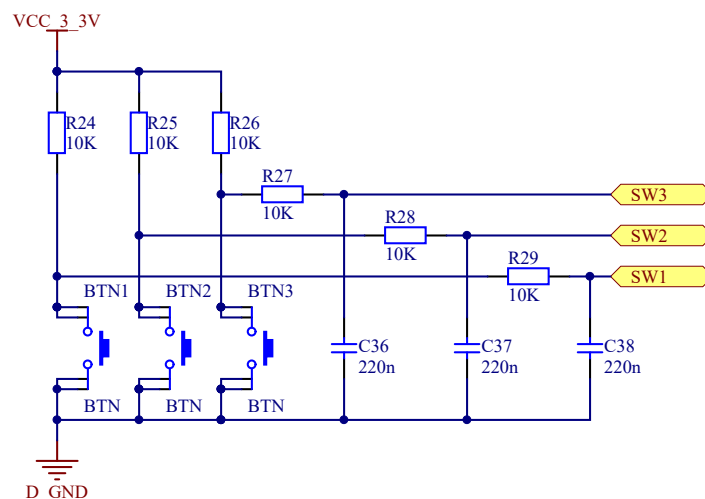


Obrázek 2.10: Schéma zapojení displeje EA DOGL128E-6

2.2.6 Ovládací rozhraní

Poslední částí na obrázku 2.1 je blok s názvem Ovládací prvky. Tato část představuje uživatelské ovládací rozhraní ve formě třech tlačítek umístěných pod displejem. Pomocí nich může uživatel měnit zobrazovaný obsah displeje.

Tlačítka jsou tvořena mechanickými kontakty. S tím však přicházejí určité neduhy, kterými mohou být například zákmity. V zájmu redukce tohoto jevu je na výstupu každého z tlačítek aplikován filtr typu dolní propust. Tento filtr by měl rychle se v čase měnící zákmity tlačítka odstranit nebo je přinejmenším do značné míry potlačit. Zapojení tlačítek spolu s filtry a pull-up rezistory nalezneme na obrázku 2.11.



Obrázek 2.11: Schéma zapojení spínačů

Přenos filtru typu dolní propust má v obecném pojetí tvar 2.2.

$$H(s) = \frac{1}{\tau \cdot s + 1} = \frac{1}{\frac{s}{\omega_0} + 1}, \quad (2.2)$$

kde τ je časová konstanta a ω_0 je mezní kruhová frekvence filtru. Při této frekvenci má filtr útlum právě -3 dB.

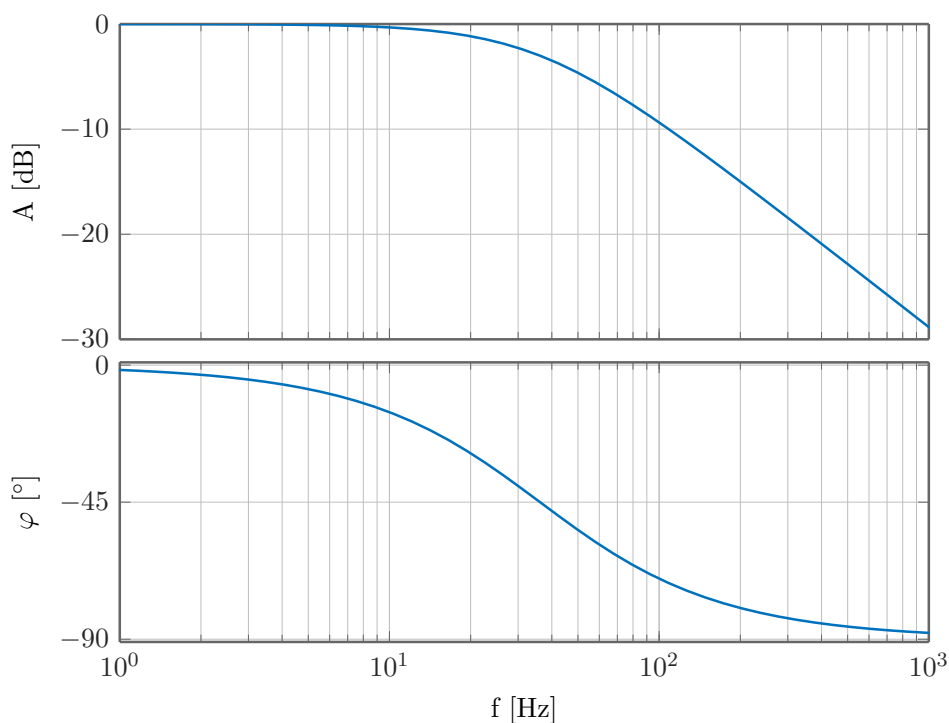
Časovou konstantu vypočítáme ze vztahu 2.3.

$$\tau = R \cdot C. \quad (2.3)$$

Jak je již patrné ze vztahu 2.2, mezní kmitočet vypočteme vztahem 2.4.

$$f_m = \frac{1}{R \cdot C \cdot 2 \cdot \pi}. \quad (2.4)$$

V našem návrhu tvoří sériová kombinace rezistorů odpor $R = 20 \text{ k}\Omega$. Kondenzátor má kapacitu $C = 220 \text{ nF}$. V tomto spojení vychází mezní kmitočet filtru $f_m = 36.17 \text{ Hz}$. Frekvenční charakteristiku v pásmu 1 až 1000 Hz odpovídající navrženému filtru nalezneme na obrázku 2.12.



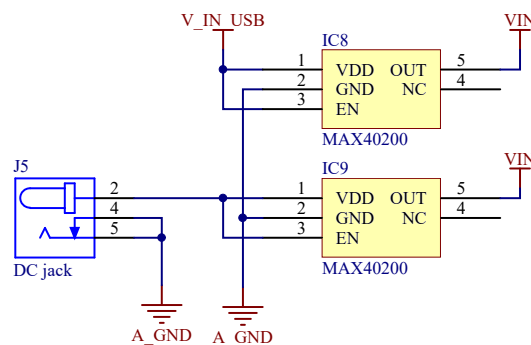
Obrázek 2.12: Frekvenční charakteristika filtru výstupu tlačítek

Dále jsou v návrhu přidány tzv. pull-up rezistory. Ty slouží k tomu, aby vstupy vyhodnocovací logiky, které jsou připojeny na tlačítka, nezůstaly v rozpojeném stavu kontaktů nepřipojeny. V případě rozpojeného kontaktu tlačítka je díky přidanému pull-up rezistoru na výstupu, z pohledu zpracovávajícího obvodu, stav vysoké logické úrovně. Sepnutím kontaktu připojíme výstup přímo na zem a vytvoříme tak na výstupu stav nízké logické úrovně.

Tedy přidáním pull-up rezistoru jsme výstupu umožnili nabývat v ustáleném stavu pouze dvou úrovní.

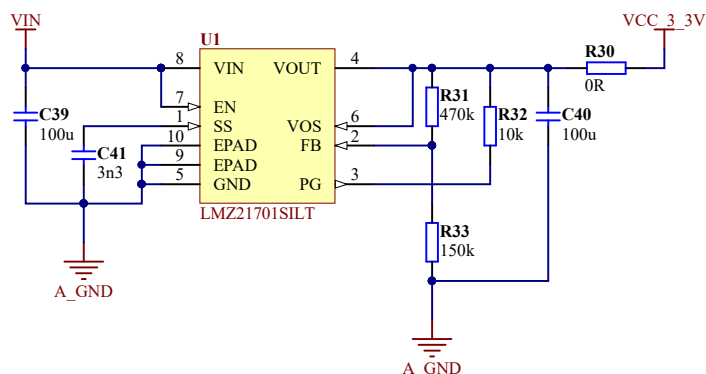
2.2.7 Napájecí obvody

Nedílnou součástí celého návrhu jsou napájecí obvody. Naše zařízení by mělo být napájeno napětovou úrovní 5 V. Z důvodu možnosti napájet zařízení, jak z USB portu, tak z externího adaptéru, je potřeba tyto dva napájecí vstupy ošetřit pro případ, že budou aktivně použity zároveň. Za předpokladu, že oba zdroje nemají absolutně stejné napětí by vznikl vyrovnávací proud, se snahou rozdíl napětí vyrovnat. To by však mohlo mít za následek destruktivní změny buď našeho zařízení nebo jednoho ze zdrojů. Ochranu před výše zmíněným problémem tvoří dvě ideální diody připojené ke každému z napájecích vstupů. Polarizací v propustném směru umožníme napájení zařízení a zároveň zamezíme vzniku zmíněných problémových stavů. Ideální dioda se chová podobně jako klasická dioda s tím rozdílem, že úbytek na ideální diodě v propustném směru je podstatně menší. Zapojení této ochrany napájecích vstupů lze vidět na obrázku 2.13.



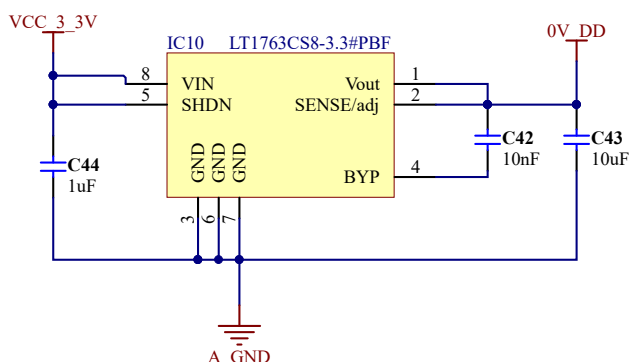
Obrázek 2.13: Schéma zapojení ideálních diod

Napájecí úroveň 5 V využijeme pouze k napájení vývojového kitu s FPGA čipem. Všechny zbylé obvody a zapojení vyžadují 3.3 V. Za účelem napětového přizpůsobení je v návrhu použit spínaný regulátor LMZ21701SILT. Jedná se o velice jednoduchý integrovaný obvod, který pro svojí funkci vyžaduje jen velice malé množství externích součástek. Jeho zapojení nalezneme na obrázku 2.14. Na vstupu a výstupu výkonové části jsou umístěny filtrační kondenzátory. Cívka, nezbytná k funkci spínaného regulátoru, je již integrována v pouzdru. Dále je připojen na vstup SS kondenzátor pro funkci soft-start. Na vstup FB je připojen dělič napětí, který zastává funkci zpětné vazby pro sledování výstupního napětí. Regulátor by měl dle výrobce být schopen dodat do obvodu proud až 1 A. [13]



Obrázek 2.14: Schéma zapojení obvodu LMZ21701SILT

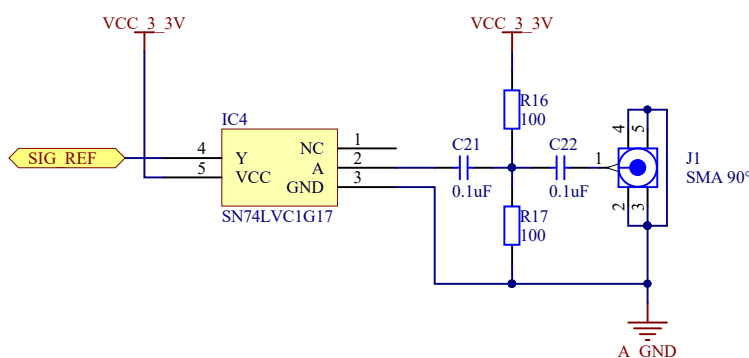
Na obrázku 2.15 také nalezneme obvod LT1763CS8-3.3#PBF. Ten je do návrhu přidán pro napájení digitálních bran A/D převodníku. Motivací tohoto kroku bylo zapojení vývojové desky k převodníku dle výrobce. [3] Napěťový regulátor je typu LDO. To značí, že dokáže pracovat i s malým rozdílem vstupního a cíleného napětí. Nejedná se o spínaný regulátor. Můžeme tedy očekávat, že při větších odběrech bude obvod podstatně více tepelně namáhán než výše uvedený regulátor. To je způsobeno tím, že se jedná o lineární regulátor, u něhož je ztrátový výkon dán zejména rozdílem vstupního a výstupního napětí a proudem tekoucím do zátěže. Tato skutečnost nám však vzhledem k naší aplikaci nevádí. Dle výrobce by měl být tento obvod schopen dodat proud až 500 mA. Ke své funkci opět potřebuje jen velice málo dalších součástek. Dle zapojení na obrázku 2.15 postačí připojit na vstup a výstup filtrační kondenzátor, propojit výstupní pin s pinem SENSE a v sérii s dalším kondenzátorem připojit i vstup BYP k výstupnímu pinu. Propojení pinu BYP s výstupem pomocí kondenzátoru by mělo pomoci obvodu ke snížení šumu. [2]



Obrázek 2.15: Schéma zapojení obvodu LT1763CS8-3.3#PBF

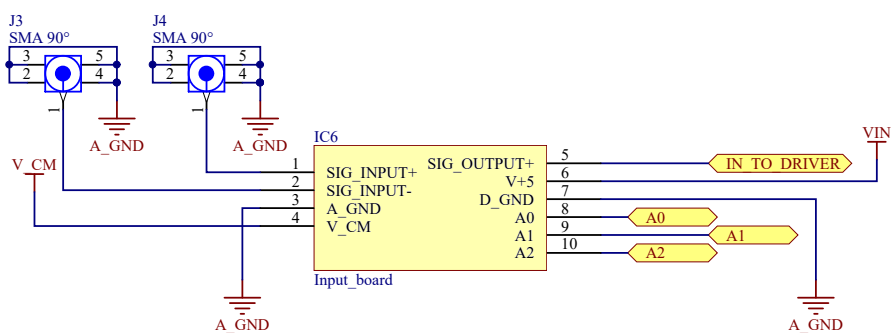
2.3 Nevyužitý návrh

Na desce plošných spojů nalezneme dva ucelené bloky, které nebyly osazeny. Jeden z těchto bloků by měl sloužit ke zpracování referenčního signálu. K tomu je využit Schmittův klopný obvod SN74LVC1G17. Na obrázku 2.16 nalezneme jednoduché zapojení tohoto obvodu. Aktuálně v tomto projektu není se zmíněnou funkcionalitou operováno. Do návrhu byl obvod zařazen z důvodu možného budoucího rozšíření o lock-in zesilovač na hradlovém poli.



Obrázek 2.16: Schéma zapojení Schmittova klopného obvodu SN74LVC1G17

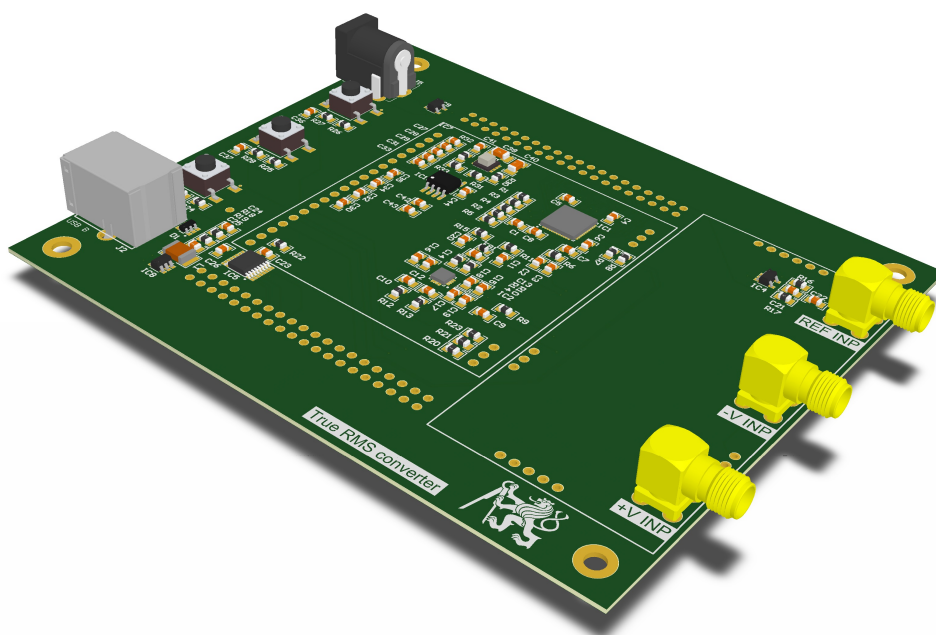
Druhým blokem je připravená patice pro zasunutí desky, která by disponovala obvody pro úpravu vstupního signálu. K tomuto řešení jsme přistoupili po dlouhém a neúspěšném hledání vhodného obvodu či spojení více obvodů pro úpravu vstupujícího měřeného signálu. Požadavkem byla možnost měnit zesílení signálu tak, aby byl zajištěn vstupní napěťový rozsah od desítek mV do několika jednotek V. Zároveň, aby bylo možné měřit, jak stejnosměrnou, tak střídavou složku signálu. To vše ve frekvenčním pásmu od DC do několika jednotek MHz. Po prostudování neuspokojující nabídky obvodů různých výrobců vznikl návrh přípravy pro pozdější řešení daného problému. Na patici je přivedeno napájecí napětí 5 V, digitální a analogová zem, napěťová úroveň V_{CM} generovaná A/D převodníkem, 3 ovládací piny připojeny na FPGA obvod, pozitivní a negativní signálový vstup pro možnost diferenčního měření a výstup signálu na driver pro A/D převodník. Na obrázku 2.17 je schéma zapojení popisované patice. Aktuálně je přímo na patici pomocí propojky spojen pozitivní signálový vstup se signálovým výstupem.



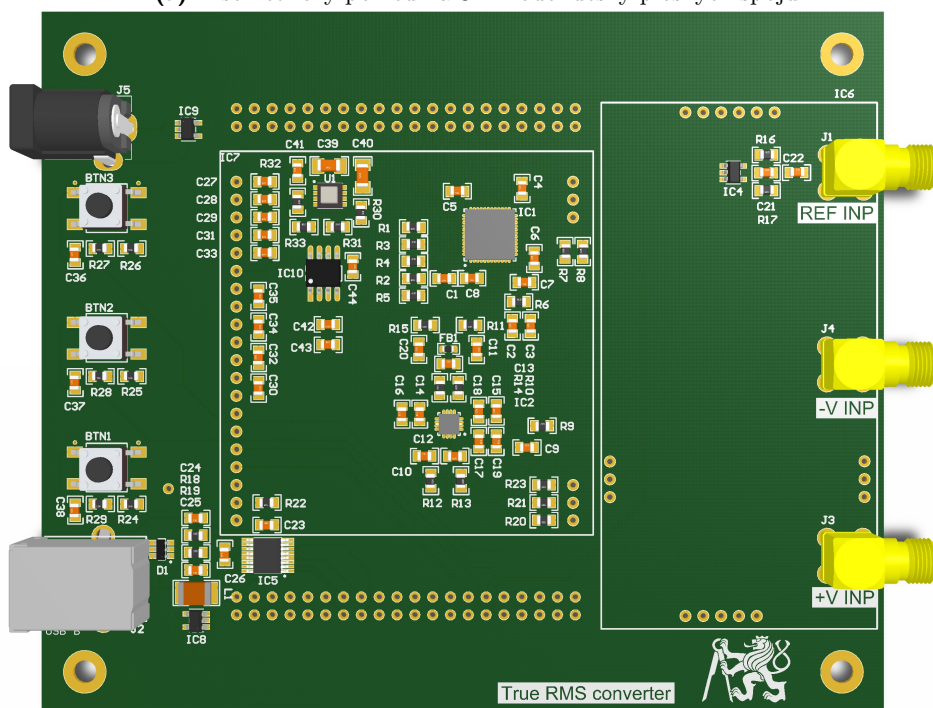
Obrázek 2.17: Schéma zapojení vstupní desky

2.4 Poznámky k návrhu desky plošných spojů

Pro návrh desky plošných spojů jsme použili čtyřvrstvého vedení cest. Vrchní a spodní vrstva slouží k vedení signálových cest. Vnitřní vrstvy jsou využity k rozvodu napájecích úrovní po celé desce. Toto rozložení je velice výhodné z hlediska přehlednosti celého návrhu. Vedení cest na vnitřních vrstvách je inverzní k návrhu cest na vrstvách vnějších. To znamená, že při návrhu neurčujeme, kudy daná cesta povede, ale naopak, kde bude chybět vodivé spojení. Vytvoří se tím tak větší vodivé celky, které mají relativně malý odpor. Z hlediska napájení je tento návrh naprosto ideální. Ve výsledku mají cesty relativně velkou proudovou zatížitelnost. Propojení mezi jednotlivými vrstvami je provedeno pomocí prokůvů. K návrhu byl použit software Altium Designer. Výsledek našeho snažení lze vidět ve formě 3D modelu na obrázku 2.18.



(a) : Isometrický pohled na 3D model desky plošných spojů

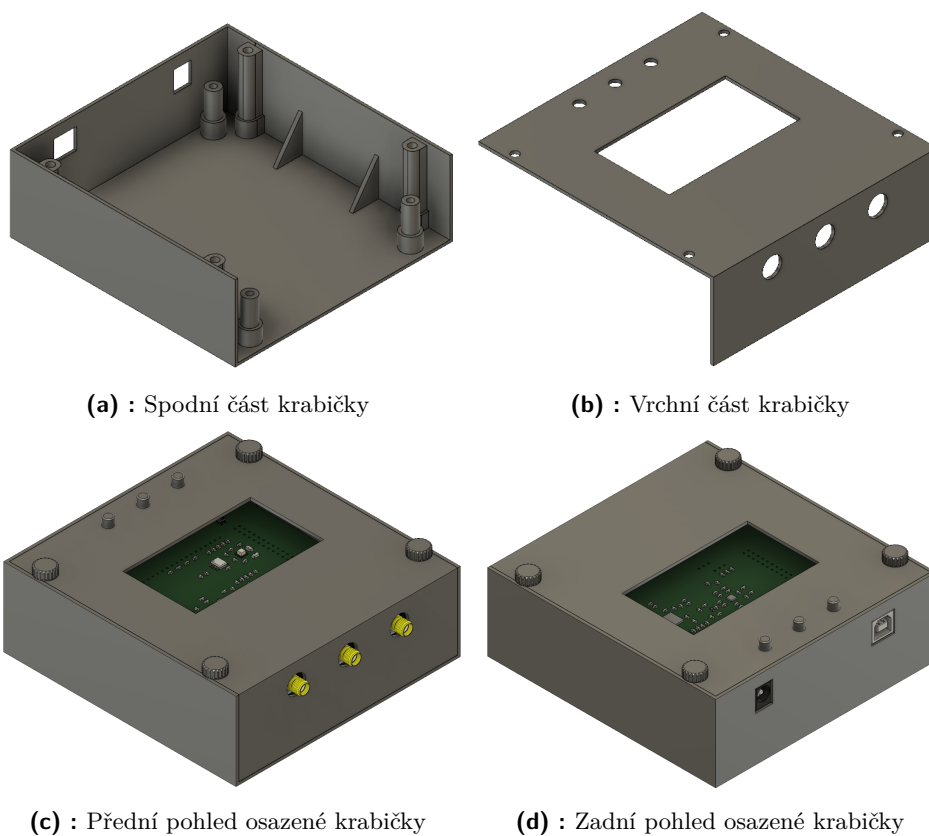


(b) : Pohled shora na 3D model desky plošných spojů

Obrázek 2.18: 3D model desky plošných spojů

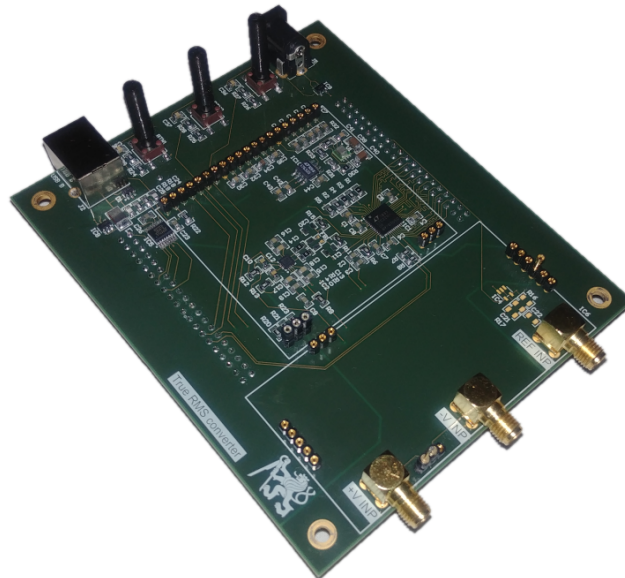
2.5 Krabička pro vyvíjené zařízení

Pro zakrytí choulostivých obvodů a dosažení více uživatelsky přívětivého vzhledu celého zařízení byla vymodelována krabička. Následně byla vytisknuta na 3D tiskárně z materiálu PETG a osazená DPS do ní byla vložena. Modelování krabičky proběhlo v softwaru Autodesk Fusion 360. Obrázek 2.19 vyobrazuje podobu 3D modelu vytvořené krabičky.

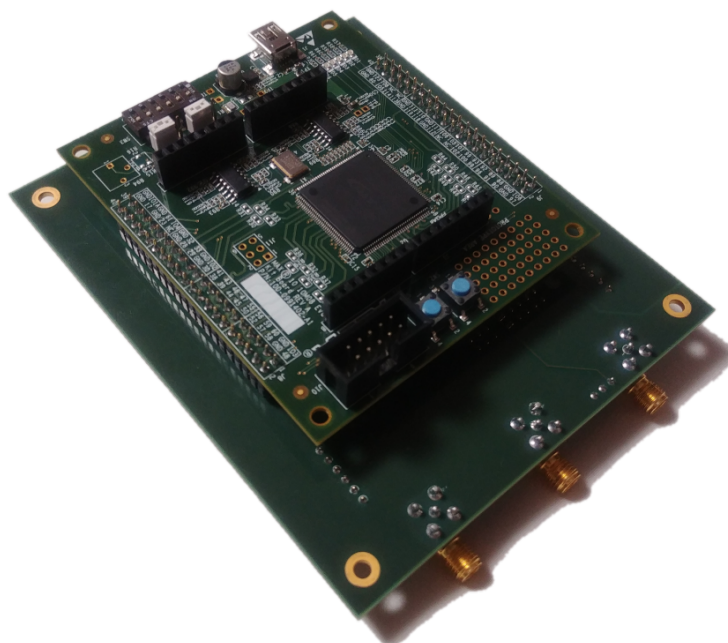


Obrázek 2.19: 3D model krabičky

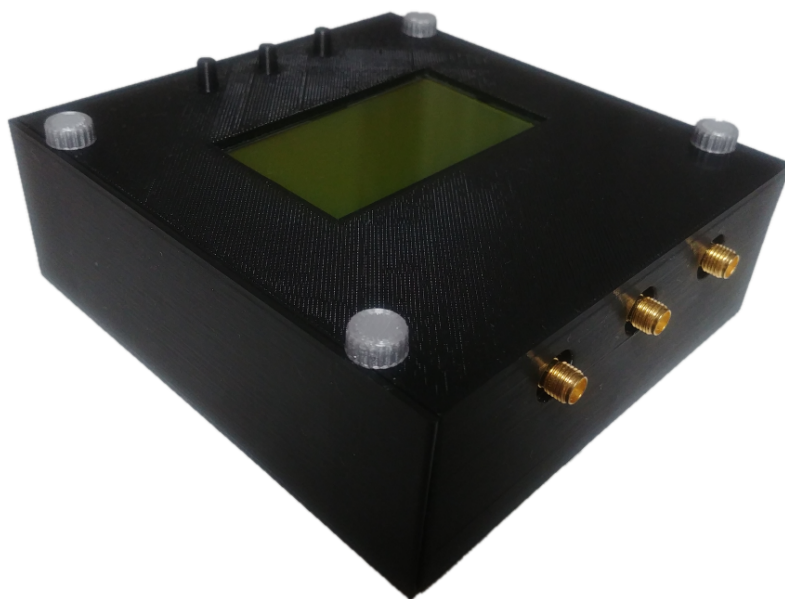
2.6 Výsledná podoba



Obrázek 2.20: Skutečná podoba DPS z vrchní strany



Obrázek 2.21: Skutečná podoba DPS ze spodní strany



Obrázek 2.22: Skutečná podoba zařízení ze zadní strany



Obrázek 2.23: Skutečná podoba zařízení z přední strany

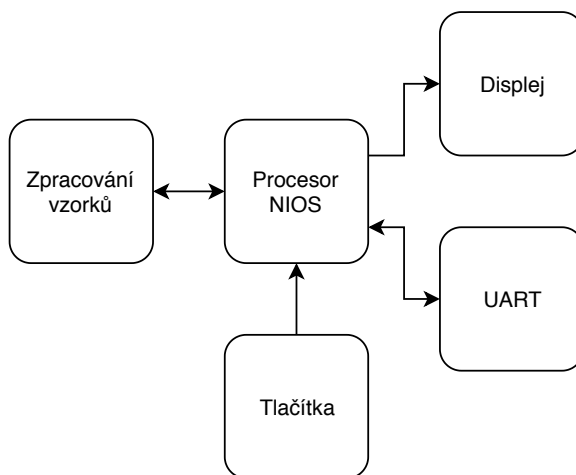
Kapitola 3

Implementace

3.1 FPGA čip

Jak již bylo zmíněno v sekci 2.2.3, je naším centrálním výpočetním prvkem hradlové pole MAX10. Pro jeho programování je použit jazyk VHDL. Prostřednictvím VHDL jazyka jsme schopni sestavovat hardwarové bloky hradlového pole do funkčních celků.

Na úrovni jazyka VHDL je hlavním cílem zajistit obsluhu A/D převodníku včetně ukládání vzorků do paměti. Následné zpracování uložených vzorků by měl přebrat procesor NIOS. Návrh vnitřního zapojení hradlového pole můžeme provádět buď graficky spojováním funkčních bloků nebo jejich zapojení popisovat kódem. Osvědčilo se rozdělit návrh do menších celků, které jsou následně propojeny grafickou formou návrhu. Tento způsob je výhodný především pro přehlednost celého sestavovaného projektu. Obrázek 3.1 popisuje VHDL návrh formou blokového zapojení jednotlivých funkčních částí.



Obrázek 3.1: Blokové schéma VHDL návrhu

3.1.1 VHDL blok pro zpracování vzorků

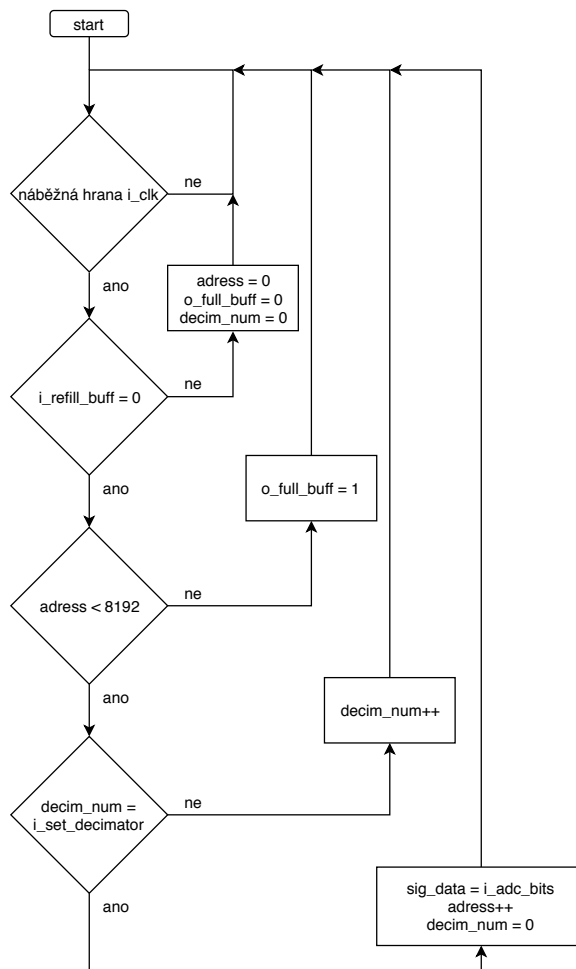
Tato část je implementována formou kódu. Na začátku pomocí tzv. entity zadefinujeme vstupní a výstupní rozhraní dané části návrhu. Tabulka 3.1 ukazuje signály definované v rámci entity.

Název signálu	Typ	Počet bitů	Datový typ
i_global_clk_40	vstup	1	std_logic
i_clk	vstup	1	std_logic
i_adc_bits	vstup	16	std_logic_vector
i_memory_address	vstup	13	std_logic_vector
i_refill_buff	vstup	1	std_logic
i_set_decimator	vstup	12	std_logic_vector
o_memory_output	výstup	16	std_logic_vector
o_clk	výstup	1	std_logic
o_full_buff	výstup	1	std_logic

Tabulka 3.1: Parametry definované entity

- i_global_clk_40 → Hodinový signál pro A/D převodník.
- i_clk → Hodinový signál o kmitočtu 50 MHz.
- i_adc_bits → Bitové slovo nesoucí informaci o daném vzorku.
- i_memory_address → Vstup pro adresaci do paměti procesorem.
- i_refill_buff → Bit, určující, zda-li je možné začít přepisovat stará data.
- i_set_decimator → Vstup, jehož hodnota určuje, jakým faktorem budeme decimovat měřený signál.
- o_memory_output → Výstup dat z paměti pro procesor.
- o_clk → Hodinový signál posílaný na AD převodník.
- o_full_buff → Bit, indikující plnu paměť.

Samotná funkce této části návrhu je velice jednoduchá. Vzorky, které jsou dostupné na výstupní digitální bráně A/D převodníku, budeme ukládat do paměti. NIOS je informován o aktuálním stavu paměti pomocí jednoho bitu. Naopak stavem dalšího bitu, nastavovaného procesorem, určujeme, zda-li můžeme stará data začít přepisovat novými vzorky. Je zde také implementována decimace vzorkovaného signálu, řízená ze strany procesoru. Tedy stavem řídicí sběrnice pro decimaci určíme, kolikátý vzorek bude uložen do paměti. Funkce je popsána vývojovým diagramem na obrázku 3.2.



Obrázek 3.2: Vývojový diagram popisující ukládání dat na úrovni VHDL

- `adress` → Proměnná, uchovávající aktuální adresu do paměti s daty.
- `sig_data` → Vstup do 16-ti bitové paměti.
- `decim_num` → Pomocná proměnná pro decimaci signálu.

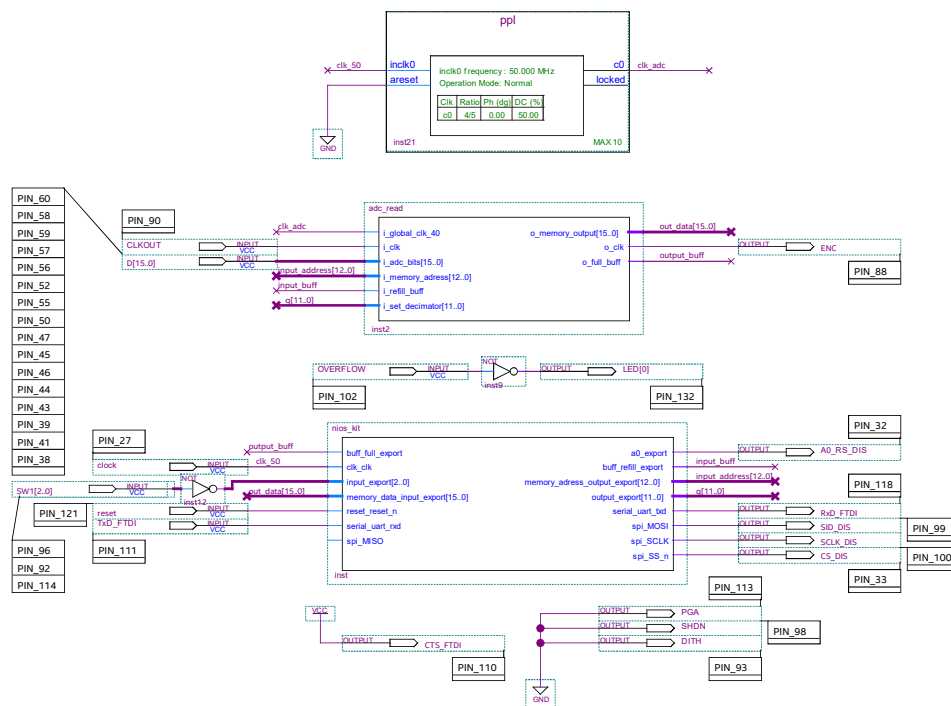
V návrhu je použita dvouportová paměť s šířkou datové sběrnice 16 bitů a šířkou adresní sběrnice 13 bitů. Z toho tedy plyne, že do paměti lze uložit 8192 vzorků.

3.1.2 VHDL blok tvořící hodinový signál pro A/D převodník

Vzorkovací frekvence pro A/D převodník je generována pomocí PLL bloku. Jedná se o fázový závěs, pomocí kterého je tvořena jedna fixní frekvence hodinového signálu. Vstupem PLL je hodinový signál generovaný krystalovým oscilátorem o frekvenci 50 MHz. PLL upraví frekvenci hodinového signálu na 40 MHz.

3.1.3 Výsledná podoba celkového návrhu na úrovni VHDL

Na obrázku 3.3 je k vidění samotné propojení jednotlivých bloků návrhu na grafické úrovni v prostředí Quartus.



Obrázek 3.3: Grafická podoba projektu v prostředí Quartus

3.2 Procesor NIOS II

Výpočetní jednotka NIOS II je proprietární procesor společnosti Altera určený k aplikaci na hradlovém poli. Jeho hardwarová struktura je vystavěna dle zapsaného kódu. Jeho hlavní předností je variabilita, díky které si vývojář může v rámci možností přizpůsobit konečnou podobu procesoru dle svých potřeb. [5] Jeho návrh je možný provést v nástroji Platform Designer, který je k dispozici v rámci vývojového prostředí Quartus.

Jedná se o procesor typu RISC. Rozeznáváme tři výkonnostní verze procesoru. Značí se písmeny e (economy), s (standard) a f (fast). Ve stejném uvedeném pořadí stoupá i vybavenost a výkon daného procesoru. V tabulce 3.2 je uvedený přehled vybraných parametrů jednotlivých procesorů. V projektu je aktuálně použita verze s označením e.

Nios II	e	s	f
Stupeň zřetězení	1 stupeň	5 stupeň	6 stupeň
Predikce skoků	-	statická	dynamická
Násobení	software	3-cyklová násobička	1-cyklová násobička
Posun	software	3-cyklový posouvač	3-cyklový posouvač
Počet logických elementů	540	1030	1600
Maximální frekvence hodin	195 MHz	110 MHz	140 MHz
Výkon	18 MIPS	55 MIPS	145 MIPS

Tabulka 3.2: Přehled parametrů jednotlivých výkonových verzí procesoru NIOS II [5]

Nástroj Platform Designer nám umožňuje graficky propojit jednotlivé funkční bloky, které by měl výsledný procesor obsahovat. Nastavení samotných jednotlivých bloků probíhá formou vyplnění připravených konfiguračních formulářů. Podobu grafické konfigurace v nástroji Platform Designer můžeme vidět na obrázku 3.4. Zmíněný obrázek zároveň ukazuje konfiguraci námi použitého procesoru pro tento projekt.

Z důvodu velkého nedostatku paměti RAM musela být použita i paměť flash. Procesoru tedy byla přiřazena RAM paměť o velikosti 18000 bitů. Zbývající paměťové nároky naší aplikace pokrývá paměť flash. Je použita základní frekvence hodinového signálu 50 MHz. Dále zde nalezneme paralelní vstupní bránu s šířkou 20 bitů a paralelní výstupní bránu o šířce 27 bitů. V případě vstupní brány je 16 bitů použito pro přijímání uložených dat ze vzorkovací paměti, 3 bity pro tlačítka a 1 bit pro indikaci plné vzorkovací paměti. Výstupní brána disponuje 13-ti bity pro adresaci vzorkovací paměti, 12-ti bity pro řízení decimace signálu, 1 bitem pro požadavek o znovu naplnění vzorkovací paměti a 1 bitem pro obsluhu displeje. Pro komunikaci s nadřazeným systémem v návrhu nalezneme blok obsluhující rozhraní UART. Rychlost komunikace je nastavena na 115200 Bd, bez parity a jeden stop bit. Pro obsluhu displeje je použit blok tvořící SPI rozhraní.

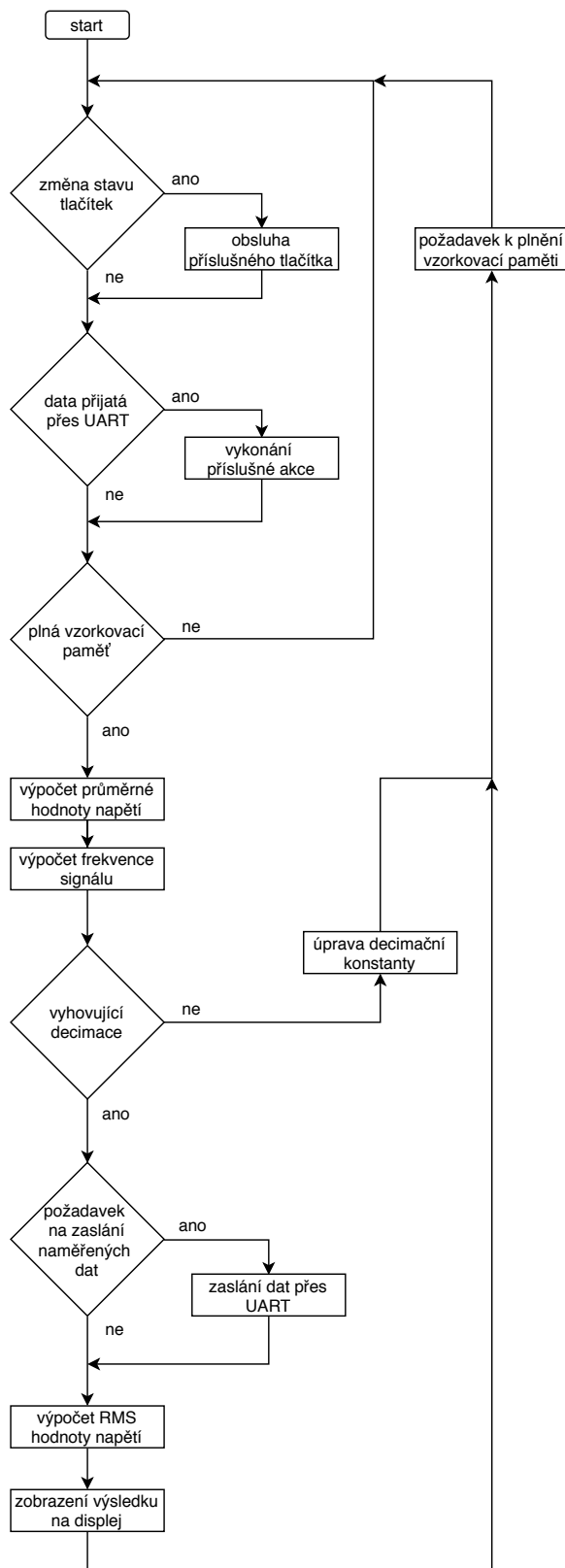
3. Implementace

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		clk	Clock Source	clk	exported			
		clk_in	Clock Input	reset				
		clk_in_reset	Reset Input	<i>Double-click to export</i>	clk			
		clk_reset	Clock Output	<i>Double-click to export</i>				
		reset	Reset Output					
<input checked="" type="checkbox"/>		onchip_ram	On-Chip Memory (RAM or ROM)...	<i>Double-click to export</i>	clk	# 0x0004_8000	0x0004_c61f	
		clk1	Clock Input	<i>Double-click to export</i>	[clk1]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk1]			
		reset1	Reset Input	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		nios2	Nios II Processor	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		data_master	Avalon Memory Mapped Master	<i>Double-click to export</i>	[clk]			
		instruction_master	Avalon Memory Mapped Master	<i>Double-click to export</i>	[clk]			
		irq	Interrupt Receiver	<i>Double-click to export</i>	[clk]			IRQ 0
		debug_reset_requ...	Reset Output	<i>Double-click to export</i>	[clk]			IRQ 31
		debug_mem_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	# 0x0005_0800	0x0005_0fff	
		custom_instructio...	Custom Instruction Master	<i>Double-click to export</i>	[clk]			
<input checked="" type="checkbox"/>		sysid	System ID Peripheral Intel FPGA...	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		control_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	# 0x0005_1118	0x0005_111f	
<input checked="" type="checkbox"/>		timer	Interval Timer Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]			
		reset	Reset Input	<i>Double-click to export</i>	[clk]	# 0x0005_10c0	0x0005_10df	
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		irq	Interrupt Sender	<i>Double-click to export</i>	[clk]			
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]			
		reset	Reset Input	<i>Double-click to export</i>	[clk]	# 0x0005_1120	0x0005_1127	
		avalon_jtag_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		irq	Interrupt Sender	<i>Double-click to export</i>	[clk]			
<input checked="" type="checkbox"/>		pio_out	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]			
		reset	Reset Input	<i>Double-click to export</i>	[clk]	# 0x0005_10a0	0x0005_10bf	
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		pio_in	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1100	0x0005_110f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		onchip_flash_0	On-Chip Flash Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0002_0000	0x0003_c7ff	
		nreset	Reset Input	<i>Double-click to export</i>	[clk]	# 0x0005_1110	0x0005_1117	
		data	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		csr	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
<input checked="" type="checkbox"/>		pio_memory_input	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_10f0	0x0005_10ff	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		pio_memory_out...	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1080	0x0005_109f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		pio_refill_output	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1060	0x0005_107f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		pio_full_input	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_10e0	0x0005_10ef	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		uart_0	UART (RS-232 Serial Port) Intel...	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1040	0x0005_105f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				
		irq	Interrupt Sender	<i>Double-click to export</i>	[clk]			
<input checked="" type="checkbox"/>		spi	SPI (3 Wire Serial) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1020	0x0005_103f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		spi_control_port	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		irq	Interrupt Sender	<i>Double-click to export</i>	[clk]			
		external	Conduit	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		pio_A0	PIO (Parallel I/O) Intel FPGA IP	<i>Double-click to export</i>	clk			
		clk	Clock Input	<i>Double-click to export</i>	[clk]	# 0x0005_1000	0x0005_101f	
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]			
		external_connection	Conduit	<i>Double-click to export</i>				

Obrázek 3.4: Snímek konfigurace procesoru NIOS v nástroji Platform Designer

3.2.1 Firmware procesoru NIOS II

Firmware je psaný v jazyce C. Při prvotním spuštění či po resetu zařízení dojde nejprve k počátečnímu nastavení do základního stavu a následně se cyklicky provádí posloupnost úkonů dle vývojového diagramu na obrázku 3.5



Obrázek 3.5: Vývojový diagram popisující firmware procesoru NIOS II

■ Adaptivní nastavení decimace

Jako první je určen aritmetický průměr signálu. Tím získáme informaci o případném stejnosměrném posunu signálu. V další fázi kontrolujeme průchod signálu hodnotou tohoto stejnosměrného posunu. V případě, že žádný stejnosměrný posun v signálu není obsažen, jedná se o kontrolu průchodu nulou. Po zdetekování průchodu nějakou ze zmíněných hodnot napočítáme, jak velká posloupnost vzorků se nachází nad rozhodovací hodnotou. Pro správnou funkci této metody předpokládáme periodický nejlépe harmonický vstupní signál. Výslednou frekvenci měřeného signálu určíme dle vztahu 3.1.

$$f = \frac{f_s}{2 \cdot N \cdot K}, \quad (3.1)$$

kde f_s je vzorkovací frekvence A/D převodníku, N je počet vzorků jedné půlperrody a K je decimační faktor.

Mimo jiné se provádí i analýza průměrné délky kladné periody. Pomocí této hodnoty je opakován výpočet frekvence dle výše zmíněného vztahu 3.1. Experimentálně bylo ověřeno, že nejpřesnějších měření dosáhneme aritmetickým průměrem těchto dvou určených frekvencí.

Decimační faktor K je stanoven na základě počtu vzorků jedné půlperrody. Postup určení, co možná nejvhodnějšího decimačního faktoru, je v celku přímočarý. Pokud je počet vzorků jedné půlperrody větší jak 400, faktor je vynásoben číslem 10. Naopak v případě, že je počet vzorků menší jak 40, faktor je dělen číslem 10. Pro vstupní signály o frekvenci vyšší jak přibližně 500 kHz již decimace signálu není potřeba. Decimační faktor K nabývá hodnot 1 až 1000.

■ Výpočet RMS hodnoty signálu

Po zajištění, co možná nejvhodnějšího navzorkování signálu, přecházíme k samotnému výpočtu efektivní hodnoty napětí. Pro dosažení velké přesnosti měření RMS hodnoty je využito poznatků nastudovaných v článku zabývajících se právě touto problematikou. [9] RMS hodnota je počítána dle definice 1.6. Výpočet však neprovádíme nad všemi vzorky. V předcházející fázi jsme určili frekvenci analyzovaného signálu. Jednoduchým výpočtem tedy zjistíme počet vzorků tvořících čtvrtinu periody měřeného signálu. Dle uvedeného vztahu 3.2 provedeme výpočet nad všemi vzorky s vynecháním poslední čtvrtperiody. Obdobně provedeme druhý výpočet dle vztahu 3.3, kde naopak začínáme počítat o čtvrt periody déle. Tímto posunem o čtvrt periody simulujeme výpočet RMS hodnoty dvou totožných signálů, které jsou od sebe fázově posunuty o $\frac{\pi}{2}$.

$$U_{ef1} = \sqrt{\frac{1}{N - N_p} \sum_{n=0}^{N-(1+N_p)} u^2(n)}, \quad (3.2)$$

$$U_{ef2} = \sqrt{\frac{1}{N - N_p} \sum_{n=N_p}^{N-1} u^2(n)}, \quad (3.3)$$

kde N je celkový počet vzorků, N_p je počet vzorků jedné čtvrtperrody a $u(n)$ je číselná hodnota jednoho vzorku.

Výslednou hodnotu získáme aritmetickým průměrem dle vztahu 3.4.

$$U_{ef} = \frac{U_{ef1} + U_{ef2}}{2}. \quad (3.4)$$

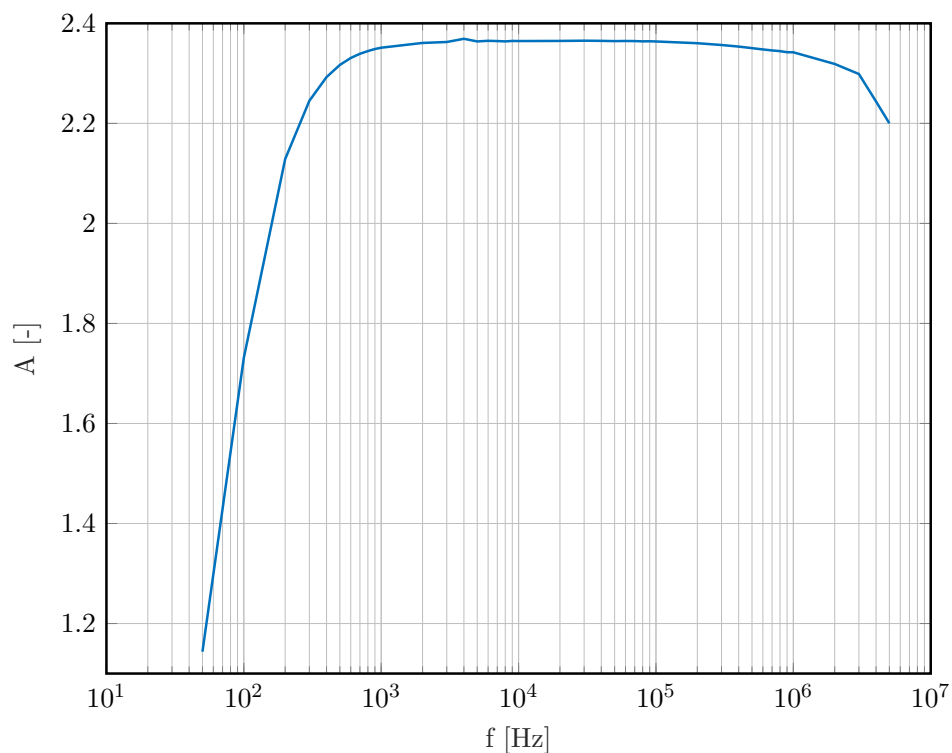
Výsledek, který dostaneme pomocí vztahu 3.4, nyní musíme upravit vztahem 3.5.

$$U_V = \frac{2.25}{2^{16} \cdot G(f)} \cdot U_{ef}, \quad (3.5)$$

kde $G(f)$ je frekvenčně závislé zesílení vstupních obvodů.

■ Korekce frekvenčně závislého zesílení vstupních obvodů

Napříč frekvenčním spektrem se projevuje frekvenčně závislé zesílení obvodů zpracovávajících měřený signál. Tato skutečnost zásadním způsobem ovlivňuje výslednou přesnost měření. Na obrázku 3.6 je vyobrazena závislost zesílení vstupních obvodů na frekvenci signálu. Útlum na frekvenci 0 až 1000 Hz je dán použitou střídavou vazbou vstupního obvodu.



Obrázek 3.6: Frekvenční charakteristika vstupních obvodů

Pro kompenzaci nelinearit výše uvedené frekvenční charakteristiky potřebujeme znát zesílení $G(f)$ ze vztahu 3.5. Lineární interpolace mezi naměřenými body frekvenční charakteristiky se ukázala jako spolehlivý způsob, jak toto zesílení odhadnout. V případě dostatečně hustého měření v bodech zlomů charakteristiky, lze touto metodou dosáhnout relativně přesných odhadů výsledného zesílení. Výpočet odhadu v obecném tvaru je zapsán vztahem 3.6.

$$A_{int} = A_{n-1} + \frac{f_{n-1} - f_{int}}{f_{n-1} - f_n} \cdot (A_n - A_{n-1}), \quad (3.6)$$

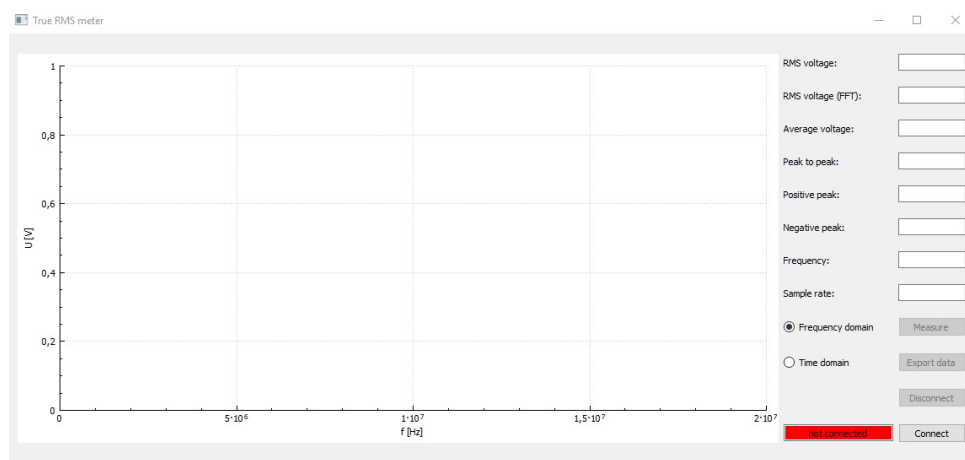
kde A_{int} je odhadnuté zesílení, f_{int} je frekvence signálu, f_{n-1} , f_n , A_{n-1} a A_n jsou naměřené body frekvenční charakteristiky.

■ Odesílání navzorkovaných dat

V případě přijetí znaku „s“ do přijímacího bufferu UARTové komunikace, je zaznamenán požadavek na odeslání uložených dat dotazujícímu se zařízení. Než se tak stane, jsou navzorkovaná data podrobena analýze, zda-li bylo měření provedeno se správnou vzorkovací frekvencí. Pokud není potřeba měnit frekvenci vzorkování, jsou data beze změny odeslána dotazujícímu se zařízení. Odesílá se celý obsah vzorkovací paměti spolu s informací, s jakou vzorkovací frekvencí bylo měření provedeno.

■ 3.3 Uživatelská aplikace

Uživatelská aplikace je vytvořena v jazyce C++ za pomoci knihoven QT, které tvoří multiplatformní framework pro vývoj grafického uživatelského prostředí. Aplikace vznikla ve vývojovém prostředí QT Creator. Prostředí podporuje návrh výsledné podoby aplikace grafickou cestou. Tato možnost výrazně ulehčí návrh samotného grafického rozhraní. Grafická podoba aplikace je zachycena na obrázku 3.7.



Obrázek 3.7: Grafické rozhraní uživatelské aplikace

■ 3.3.1 Tlačítko Connect

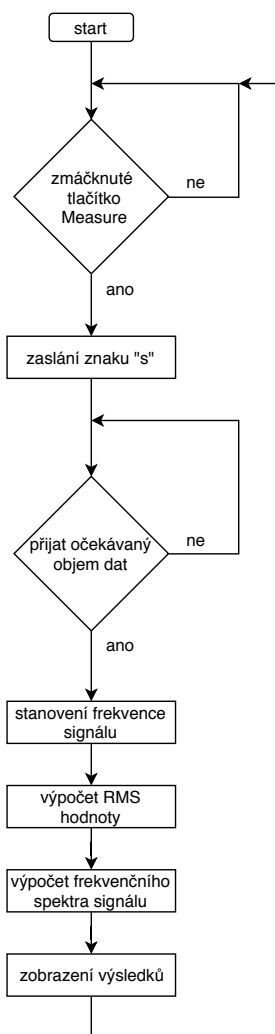
Po zmáčknutí tlačítka Connect dojde k prohledání připojených zařízení, které se v systému hlásí jako zařízení připojená přes sériový port. V případě, že nalezneme zařízení se shodnými charakteristickými identifikátory, jako má FTDI čip našeho měřiče, označíme toto zařízení za nalezené. Indikační pole v pravém dolním rohu aplikace změní barvu pozadí z červené na zelenou a vypíše se název portu, ke kterému je zařízení připojeno. Dále se aktivuje možnost poslat požadavek na měření. Může se stát, že sice hledané zařízení nalezneme, ale nepovede se nám otevřít sériový port pro komunikaci. V takové situaci zůstane indikační pole červené a vypíše se název portu, na kterém je nalezené zařízení připojeno. Třetí možností je, že zařízení nenalezneme. V takovém případě aplikace setrvá v neměnném stavu.

■ 3.3.2 Tlačítko Disconnect

Zmáčknutím tohoto tlačítka vyvoláme okamžité odpojení zařízení a uvolnění rezervovaného sériového portu. Aplikace se navrátí do svého původního stavu. V tomto stavu je připravena na opětovné připojení. Naprosto stejný efekt vyvolá odpojení zařízení od počítače.

■ 3.3.3 Tlačítko Measure

Kliknutím na tlačítko Measure spustíme posloupnost úkonů, které vedou k závěrečnému zobrazení naměřených a vypočítaných dat. Zmíněná posloupnost úkonů je graficky zpracovaná formou vývojového diagramu na obrázku 3.8. Jako první je odeslán startovní znak „s“. Tento znak vytvoří na straně příjemce požadavek na zaslání naměřených dat spolu s informací o vzorkovací frekvenci, při které byla data pořízena. Následuje proces přesouvání samotných dat. Po přijetí celého objemu dat jsou zahájeny výpočetní operace. Jak výpočet frekvence, tak stanovení RMS hodnoty signálu jsou principiálně totožné s popisovanými postupy v sekci 3.2.1. Přidanou funkcionalitou je zde frekvenční analýza. Pro tyto účely je použit iterativní algoritmus známý pod názvem radix-2 FFT. Tento algoritmus byl implementován dle nalezeného pseudokódu. [15] Velikost vstupního slova pro FFT je 2^{13} prvků. Pro získání korektních výsledků frekvenční analýzy je analyzovaný signál násobený korekčním oknem.



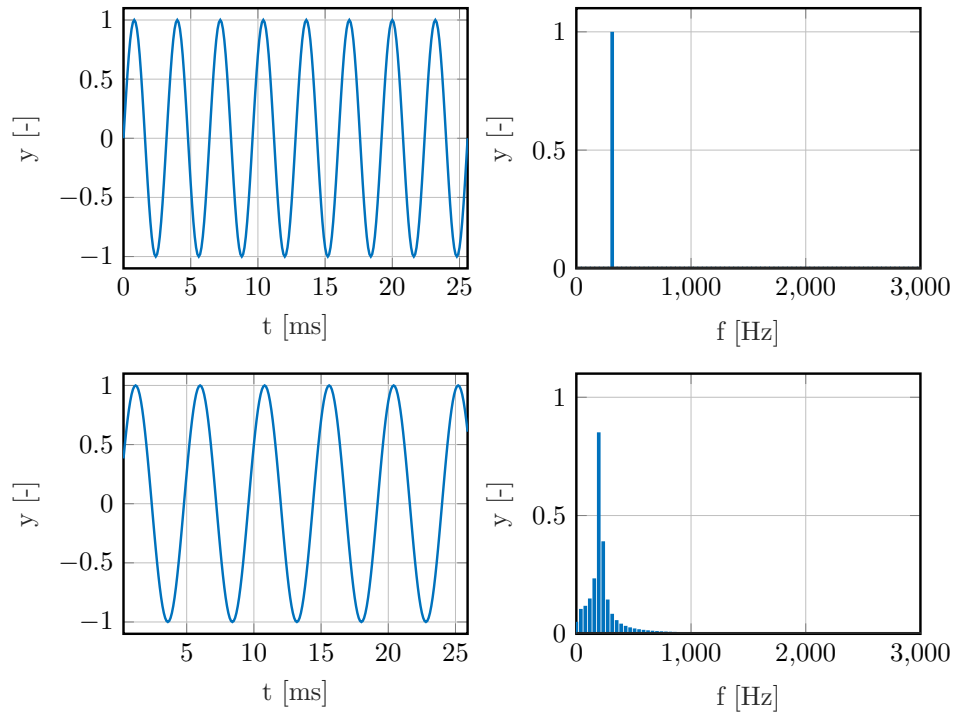
Obrázek 3.8: Vývojový diagram popisující prováděné úkony k zobrazení naměřených dat

■ Vliv volby okénka na výsledek spektrální analýzy

Při spektrální analýze pomocí FFT musí být počet zpracovávaných vzorků signálu roven mocnině dvou. Z tohoto důvodu se ve většině případů dostaneme do situace, kdy počet period signálu není celočíselným násobkem. Důsledkem toho je prosakování ve spektru, které znehodnocuje výsledek frekvenční analýzy. Tento jev lze pozorovat na obrázku 3.9. Tento problém můžeme částečně řešit použitím jiného než obdélníkového okna. Existuje velká spousta okénkových funkcí, přičemž každá z nich má různé uplatnění. V našem případě je potřeba implementovat takovou okénkovou funkci, která bude co možná nejméně deformovat skutečnou velikost amplitudy jednotlivých signálů. Použitím jiného než obdélníkového okna dochází ke zkreslení výsledných amplitud jednotlivých frekvenčních složek signálu. Korekci lze provést jednoduchým vynásobením výsledného spektra dle následujícího vztahu 3.7.

$$A_{error} = \frac{N}{\sum_{n=0}^{N-1} w(n)}, \quad (3.7)$$

kde N je velikost okna a $w(n)$ je okénková funkce.



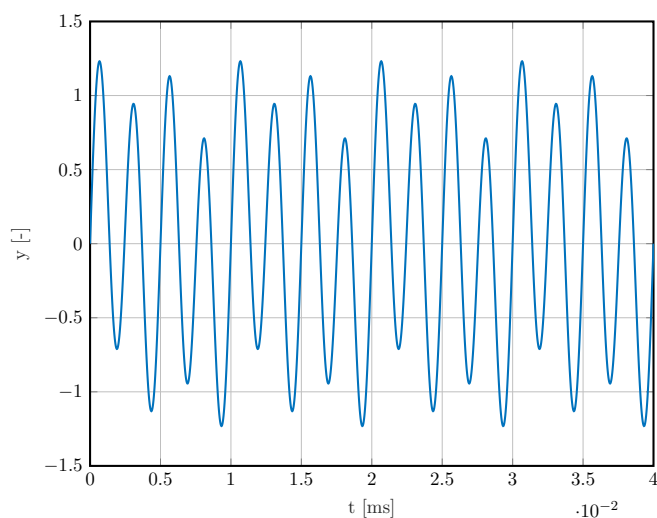
Obrázek 3.9: Ukázka spektrálního prosakování

V prostředí Matlab byla provedena simulace signálu obsahujícího tři různé frekvenční složky o různých amplitudách. Parametry jednotlivých složek jsou uvedeny v tabulce 3.3. Při frekvenční analýze byly aplikovány různé okénkové funkce a pro srovnání jsou výsledná spektra k nahlédnutí na obrázku 3.11. Při simulaci byla vzorkovací frekvence $f_s = 40$ MHz a okénko o velikosti $N = 2^{13}$. Podoba signálu, který byl podroben frekvenční analýze je k vidění na obrázku 3.10,

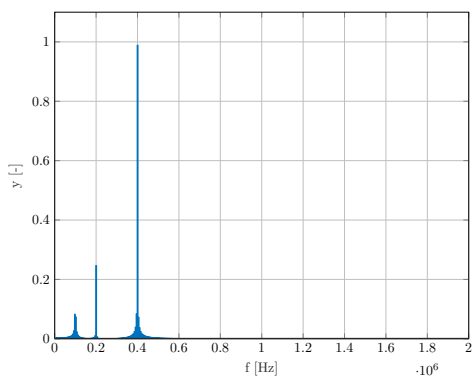
	f_1 [kHz]	f_2 [kHz]	f_3 [kHz]	y_{f_1} [-]	y_{f_2} [-]	y_{f_3} [-]
testovaný signál	400	200	100	1	0.25	0.125

Tabulka 3.3: Parametry testovaného signálu

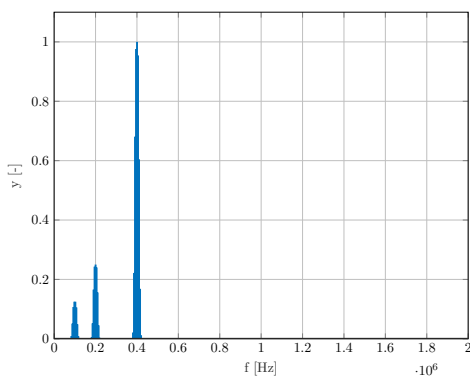
kde f_1 , f_2 a f_3 jsou frekvence jednotlivých dílčích signálů, které jsou následně sečteny, y_{f_1} , y_{f_2} a y_{f_3} jsou amplitudy zmíněných jednotlivých složek signálu.



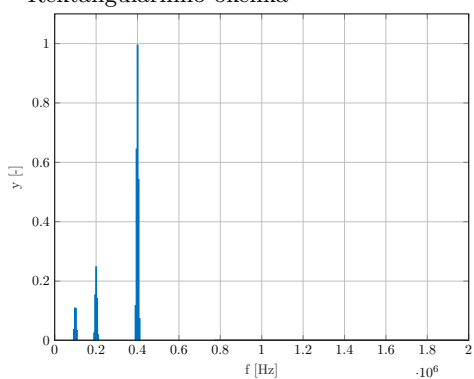
Obrázek 3.10: Časový průběh analyzovaného signálu



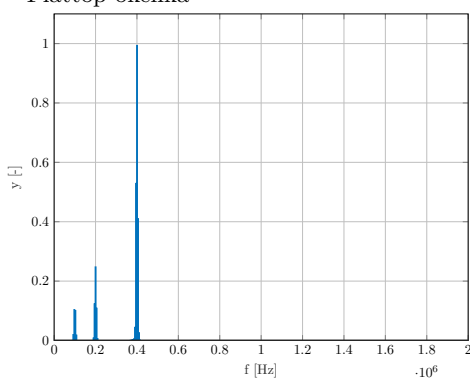
(a) : Frekvenční spektrum s použitím Rektangulárního okénka



(b) : Frekvenční spektrum s použitím Flattop okénka



(c) : Frekvenční spektrum s použitím Blackman okénka



(d) : Frekvenční spektrum s použitím Gaussova okénka

Obrázek 3.11: Porovnání vlivu použité okénkové funkce na výsledek frekvenční analýzy

V tabulce 3.4 nalezneme vyčíslenou chybu určení amplitudy dané frekvenční složky signálu v závislosti na použité okénkové funkci,

Okénková funkce	y_r [-]	y_m [-]	chyba [%]
Rektangulární	1.000	0.990040	-0.9960
Flattop	1.000	1.000043	0.0043
Blackman	1.000	0.996792	-0.3208
Gauss	1.000	0.995451	-0.4549

Tabulka 3.4: Výsledek frekvenční analýzy signálu pro složku o frekvenci 400 kHz

kde y_r je skutečná amplituda dané složky signálu, y_m je amplituda složky signálu určená frekvenční analýzou a chyba je procentuálně vyčíslená odchylka skutečné a vypočítané amplitudy dané frekvenční složky signálu.

Dle výše provedené analýzy je pro přesné měření nejvýhodnější použít okénkovou funkci zvanou Flattop. Při výpočtu spektra měřeného signálu je tedy použita právě tato funkce.

■ 3.3.4 Přepínač Frequency/Time domain

Uživateli je umožněno zobrazit, jak časové, tak frekvenční spektrum signálu. V případě, že je vybrána možnost Frequency domain, ve vykreslovacím poli uživatel uvidí výsledek frekvenční analýzy zkoumaného signálu. V případě opačném se v tomto poli zobrazí časový průběh měřeného signálu.

■ 3.3.5 Tlačítko Export data

Tlačítko Export data vyvolá uložení souboru s názvem „samples.csv“. Jak již přípona napovídá, soubor je uložen ve formátu CSV. Jedná se o hodnoty, které jsou od sebe odděleny čárkou. V souboru nalezneme 8192 přijatých, žádným způsobem neupravených vzorků. Soubor je uložen do stejného adresáře, ve kterém je spuštěna aplikace.

Kapitola 4

Experimentální výsledky

Neopomenutelnou částí tohoto projektu je i závěrečná validace přesnosti měření. Zde narážíme na problém, jaké měřicí zařízení lze brát jako etalon přesného stanovení RMS hodnoty signálu. Již z charakteru motivace této práce je zřejmé, že dostupnost vhodného měřicího přístroje pro ověření přesnosti měření našeho zařízení není příliš velká. Pro tento účel byl použit multimetr Agilent 3458A a Wavetek 1281.

4.1 Měření s využitím multimetru Agilent 3458A

Pomocí tohoto multimetru Agilent 3458A byly stanoveny korekční faktory pro naše měřicí zařízení. Následně bylo provedeno testovací měření pro ověření přesnosti měření. V tabulce 4.1 nalezneme výrobcem udávanou chybu měření. Z tabulky je patrné, že do frekvence 100 kHz má přístroj pro nás ještě relativně přijatelnou chybu. Pro vyšší frekvence je již chyba měření větší, než kterou lze dle zadání tohoto projektu tolerovat. Uvedené nejistoty měření platí pro měření v režimu Synchronous Sub-sampled Mode.

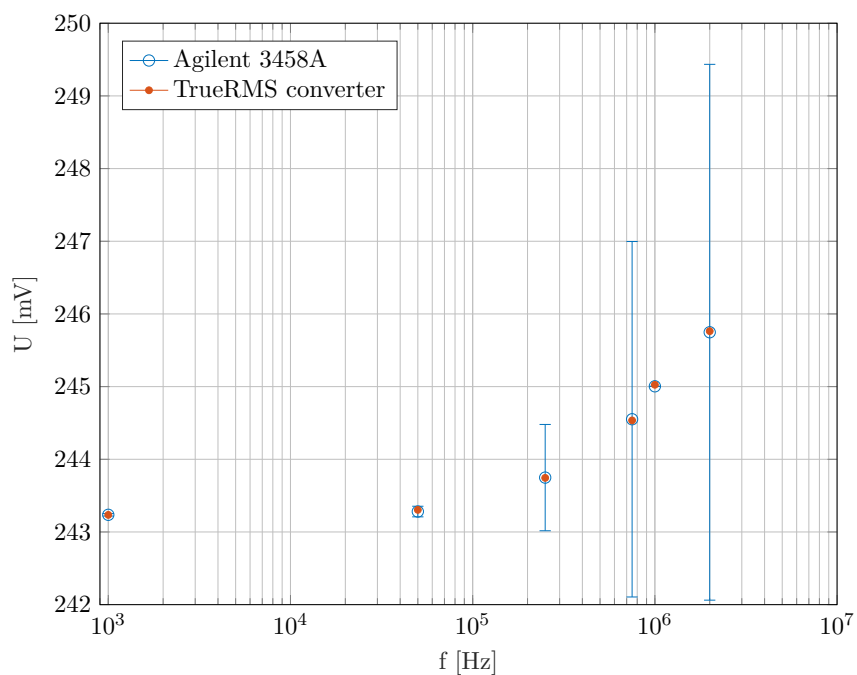
Rozsah	40 Hz až 1 kHz	1 kHz až 20 kHz	20 kHz až 50 kHz	50 kHz až 100 kHz	100 kHz až 300 kHz	300 kHz až 1 MHz	1 MHz až 2 MHz
Chyba měření [%]	0.007 + 0.002	0.014 + 0.002	0.03 + 0.002	0.08 + 0.002	0.3 + 0.01	1 + 0.01	1.5 + 0.01

Tabulka 4.1: Přehled chyb měření multimetru Agilent 3458A pro napěťový rozsah 100 mV až 10 V [1]

První z údajů je chyba z naměřené hodnoty a druhý z údajů je chyba z použitého rozsahu.

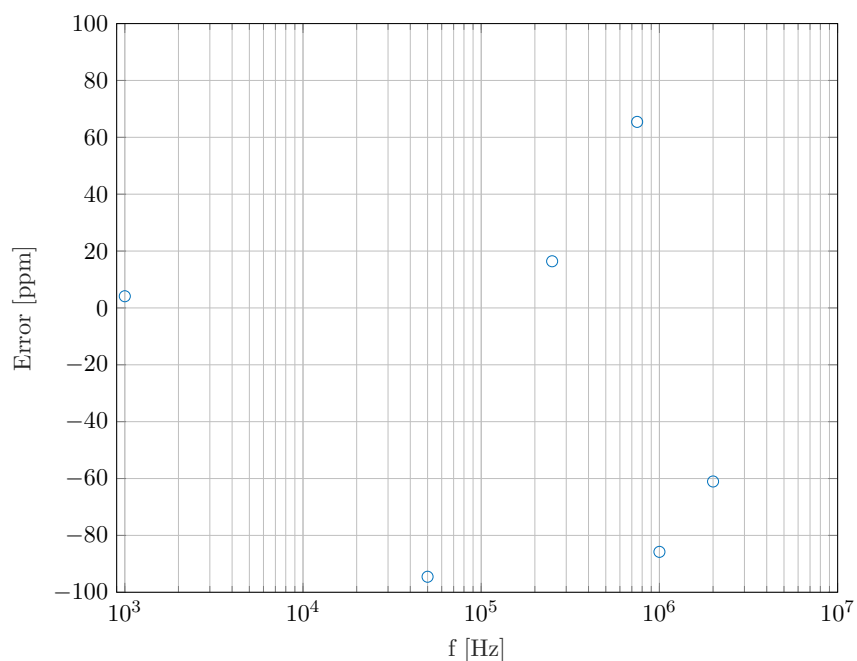
V kalibračním listu pro tento přístroj je uvedeno, že při frekvenci 1 MHz a napětí 10 V přeměřoval multimetr o cca 0.855 %. Tato hodnota představuje relativní chybu měření na frekvenci 1 MHz. Byla ověřena lineární převodní charakteristika multimetru a následně provedena korekce naměřeného napětí o právě tuto relativní chybu. Tím bychom se měli přiblížit skutečné efektivní hodnotě napětí měřeného signálu. Na obrázku 4.1 nalezneme naměřená napětí naším zařízením a multimetrem Agilent 3458A.

4. Experimentální výsledky



Obrázek 4.1: Porovnání naměřeného napětí pomocí multimetru Agilent 3458A a našeho zařízení

Na obrázku 4.2 je graficky zpracována procentuální odchylka mezi hodnotou, kterou ukázalo naše měřicí zařízení a Agilent 3458A.



Obrázek 4.2: Procentuální odchylka naměřeného napětí pomocí multimetru Agilent 3458A a našeho zařízení

4.2 Měření s využitím multimetru Wavetek 1281

Multimetru Wavetek 1281 byl využit pro ověření linearity převodní amplitudové charakteristiky. Tabulky 4.2 a 4.3 ukazují nejistoty měření v daných frekvenčních pásmech pro různé napětové rozsahy.

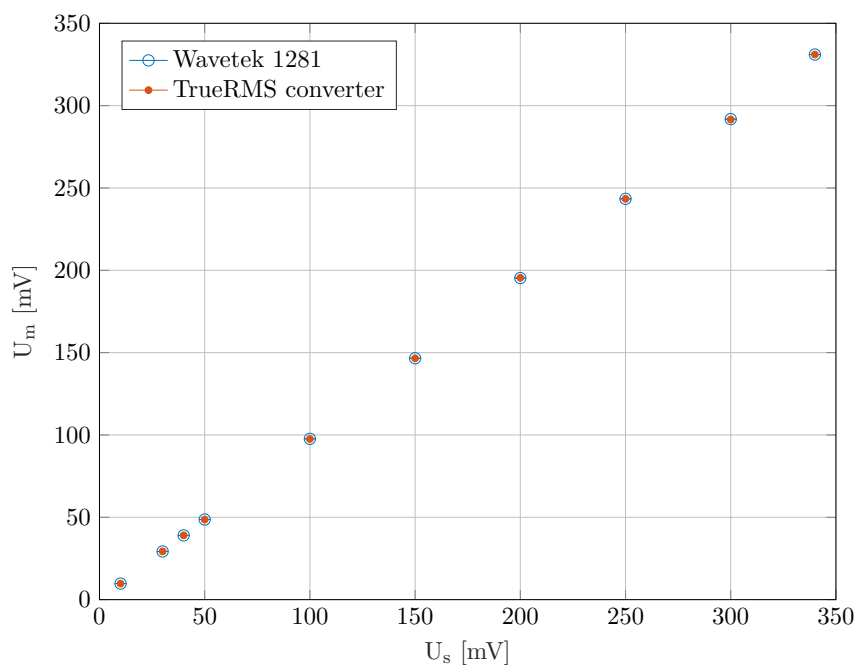
Rozsah	40 Hz až 10 kHz	10 kHz až 30 kHz	30 kHz až 100 kHz
Nejistota měření [%]	0.01 + 0.002	0.03 + 0.04	0.07 + 0.01

Tabulka 4.2: Přehled nejistot měření multimetru Wavetek 1281 pro napětový rozsah 100 mV [14]

Rozsah	2 kHz až 10 kHz	10 kHz až 30 kHz	30 kHz až 100 kHz	100 kHz až 300 kHz	300 kHz až 1 MHz
Nejistota měření [%]	0.008 + 0.001	0.02 + 0.002	0.05 + 0.01	0.3 + 0.1	1 + 1

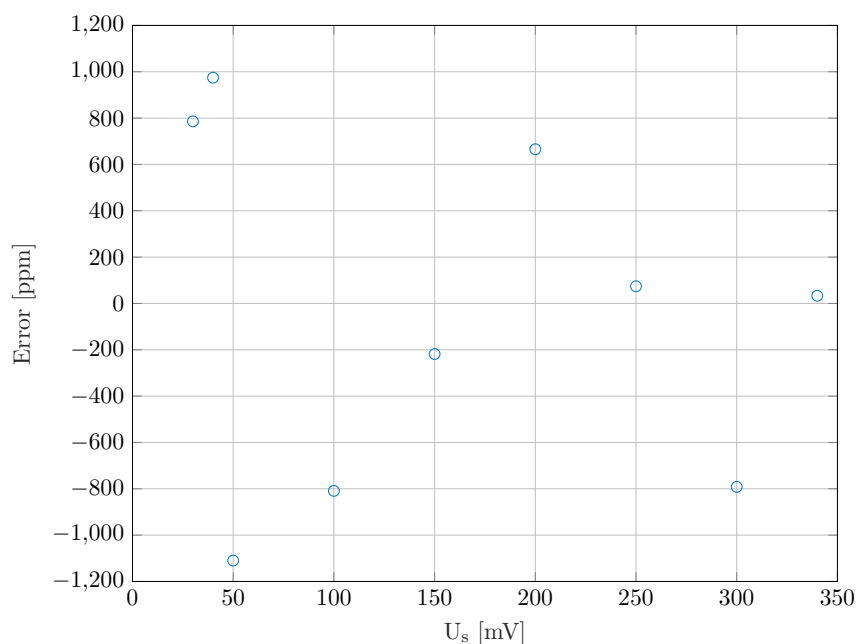
Tabulka 4.3: Přehled nejistot měření multimetru Wavetek 1281 pro napětový rozsah 1 V až 100 V [14]

Na obrázku 4.3 je zachyceno měření našeho zařízení a multimetru Wavetek 1281. Byl měřen signál o frekvenci 10 kHz. Efektivní hodnota napětí tohoto signálu byla měněna v rozsahu 10 mV až 340 mV. Napětí U_s tedy odpovídá nastavenému napětí na generátoru a napětí U_m je napětí naměřené.



Obrázek 4.3: Porovnání naměřeného napětí multimetrem Wavetek 1281 a naším zařízením při frekvenci 10 kHz

Na obrázku 4.4 nalezneme procentuální vyjádření odchylky naměřeného napětí multimetrem Wavetek 1281 a naším zařízením.



Obrázek 4.4: Procentuální odchylka naměřeného napětí multimetrem Wavetek 1281 a naším zařízením při frekvenci 10 kHz

4.3 Zhodnocení naměřených dat

Z měření multimetrem Agilent 3458A a uvedené graficky zpracované odchylky zobrazené hodnoty multimetrem a naším zařízením v obrázku 4.2 lze usoudit, že naše měřicí zařízení je schopné dosáhnout měřicí chyby lepší jak 0.2%. Bohužel má tento multimetr dle výrobce od frekvencí vyšších jak 100 kHz nezanedbatelnou chybu. Z tohoto důvodu lze prohlásit, že naše zařízení má chybu měření menší jak 0.2% pouze do frekvence 100 kHz, a také na frekvenci 1 MHz, kde jsme dle kalibračního listu dopočítali relativní chybu měření vůči kalibrátoru. Pro ostatní frekvence nejistota měření použitého multimetru znevažuje přesnost měření našeho zařízení. Lze však předpokládat, že relativní chyba měření do frekvence 1 MHz nebude přesahovat tu námi vypočtenou. Za takového předpokladu splníme zadanou maximální chybu měření i do frekvence 1 MHz.

I v případě ověřování linearity převodní amplitudové charakteristiky jsme při měření nepřesáhli rozdíl námi a multimetrem naměřené efektivní hodnoty napětí větší jak 0.2 %. Uvedený výrok platí i se započítanými nejistotami měření multimetru Wavetek 1281.

Kapitola 5

Závěr

Cílem této bakalářské práce bylo vytvořit zařízení, kterým by bylo možné měřit efektivní hodnotu napětí signálů o frekvencích v řádech až několik MHz. Umožněno mělo být i měření stejnosměrné složky signálu. Oblast stejnosměrného měření bohužel nebyla zajištěna. Nebylo možné nalézt takový obvod, který by vyhověl všem našim požadavkům v plném rozsahu viz. 2.3. Z toho důvodu byl v konečné fázi proveden ústupek, který znemožňuje spolehlivě zpracovávat stejnosměrný signál. Aktuální hardwarový návrh by sice byl schopný přenášet i stejnosměrnou složku, ale ukázalo se, že bez dalších vstupních obvodů je vhodnější vstup driveru vázat střídavou vazbou.

Pro tento projekt byl nejdříve vytvořen obvodový návrh, který byl následně realizován. Zařízení je řízeno FPGA obvodem. Vznikl tedy návrh na úrovni VHDL jazyka pro použitý čip Intel MAX10. Dále byl vytvořen firmware pro procesor NIOS, který se částečně stará o správný běh celého zařízení. Pro vzdálenou obsluhu zařízení byla vytvořena PC aplikace. V konečné fázi projektu byla pro zařízení vymodelována krabička a následně vytisknuta na 3D tiskárně.

Zařízení lze napájet buď z USB portu nebo z přídatného adaptéru. Na zařízení nalezneme displej, který slouží k zobrazování základních parametrů měřeného signálu. K ovládání měřicího zařízení slouží tři tlačítka umístěná pod zobrazovacím panelem. Vytvořená aplikace zajišťuje obsluhu měřicího zařízení. V případě použití této aplikace uživatel získá možnost zobrazit frekvenční spektrum měřeného signálu a případně naměřená data exportovat do souboru ve formátu CSV. Aplikace také umožňuje vykreslit časový průběh signálu.

Co se týče přesnosti měření jsme dosáhli uspokojujících výsledků. V tomto ohledu se povedlo nepřesáhnout maximální mez chyby měření dle zadání. Přesnost měření je nyní závislá především na vhodné kalibraci.

Do budoucna by bylo potřeba vhodným způsobem vyřešit zpracování signálu tak, aby bylo umožněno měřit i stejnosměrnou složku signálu. Aktuální hardwarový návrh je pro toto rozšíření uzpůsoben. V případě potřeby měřit signály o vyšší frekvenci by mohl být stávající A/D převodník vyměněn za rychlejší verzi se vzorkovací frekvencí 130 MHz. Díky shodnému pouzdru by se tato změna obešla bez větších zásahů do stávajícího návrhu. Citelného zlepšení bychom také dosáhli implementací lock-in zesilovače za účelem nahrazení nyní užívané spektrální analýzy.

Příloha A

Vysvětlení použitých zkratk

Použitá zkratka	anglický význam	český význam
RMS	root mean square	odmocnina součtu druhých mocnin
DDS	direct digital synthesis	přímá digitální syntéza
DFT	discrete Fourier transform	diskrétní Fourierova transformace
FFT	fast Fourier transform	rychlá Fourierova transformace
ADC	analog to digital converter	analogově digitální převodník
FPGA	field programmable gate array	programovatelné hradlové pole
UART	universal asynchronous receiver transmitter	univerzální asynchronní přijímač vysílač
SPI	serial peripheral interface	sériové periferní rozhraní
DPS	-	deska plošných spojů
PLL	phase locked loop	fázový závěs
RISC	reduced instruction set computer	redukováná instrukční sada
CSV	comma separated values	hodnoty oddělené čárkami
LCD	liquid crystal display	displej z tekutých krystalů

Tabulka A.1: Význam zkratk použitých v této práci

Příloha B

Literatura

- [1] Agilent Technologies: Agilent 3458A Multimeter. *Access*.
URL https://d3fdwrtpsindh7j.cloudfront.net/Docs/datasheet/hp_3458a.pdf
- [2] Analog Devices: LT1763: s. 1–20.
URL <http://cds.linear.com/docs/en/datasheet/1763fh.pdf>
- [3] Analog Devices: LTC2204: s. 1–25.
URL <https://www.analog.com/media/en/technical-documentation/data-sheets/22054fc.pdf>
- [4] Analog Devices: LTC6400-8: s. 1–16.
URL <https://www.analog.com/media/en/technical-documentation/data-sheets/64008f.pdf>
- [5] Chu, P. P.: *Embedded SoPC Design with Nios II Processor and VHDL Examples*. US: Wiley, první vydání, 2011, ISBN 9781118008881;111800888X;1118146530;9781118146538;.
- [6] ELECTRONIC ASSEMBLY: EA DOGL128E-6.
URL <https://www.lcd-module.com/eng/pdf/grafik/dogl128-6e.pdf>
- [7] FTDI: FT230XS-R.
URL https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf
- [8] Hansen, E. W.: *Fourier transforms: principles and applications, with an introduction to complex analysis*. Wiley, první vydání, 2014, ISBN 9781118479148;1118479149;9781118901694;111890169X;.
- [9] Hegeduš, H.; Mostarac, P.; Malarić, R.: Comparison of RMS Value Measurement Algorithms of Non-coherent Sampled Signals. *Measurement Science Review*, ročník 11, č. 3, 2011: s. 79–84, ISSN 13358871, doi: 10.2478/v10048-011-0019-9.
URL <http://www.measurement.sk/2011/Hegedus.pdf>

- [10] Intel: MAX 10 FPGA (10M08S , 144-EQFP) Evaluation Kit: User Guide. 2015.
URL https://www.altera.com/en_US/pdfs/literature/ug/ug_max10_eval_10m80.pdf
- [11] Kucera, J.; Sedlacek, R.; Bohacek, J.: An HF coaxial bridge for measuring impedance ratios up to 1MHz. *Measurement Science and Technology*, ročník 23, č. 8, 2012, ISSN 13616501, doi:10.1088/0957-0233/23/8/085004.
- [12] Muciek, A.; Cabiati, F.: Analysis of a three-voltmeter measurement method designed for low-frequency impedance comparisons. *Metrology and Measurement Systems*, ročník 13, č. 1, 2006: s. 19–33, ISSN 0860-8229.
URL http://metrology.pg.gda.pl/full/2006/M&MS_2006_019.pdf
- [13] Texas Instruments: LMZ21701. 2018.
URL <http://www.ti.com/lit/ds/symlink/lmz21701.pdf>
- [14] Wavetek: Wavetek 1281.
URL <https://www.axiomtest.com/documents/models/Wavetek1281Datasheet.pdf>
- [15] Wikipedia contributors: Cooley–Tukey FFT algorithm — Wikipedia, The Free Encyclopedia. 2019, [Online; accessed 4-May-2019].
URL https://en.wikipedia.org/w/index.php?title=Cooley%E2%80%93Tukey_FFT_algorithm&oldid=892373686