

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
FAKULTA ELEKTROTECHNICKÁ
KATEDRA MĚŘENÍ



Diplomová práce

Autor:
Bc. Michal PŘIBIL

Vedoucí:
Ing. Jan FISCHER CSc.

10. května 2013



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Michal Přibíl**

Studijní program: **Kybernetika a robotika**
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Tříkanálová měřicí jednotka s mikrořadičem STM32**

Název tématu anglicky: **3-Channel Measurement Module Based on the Microcontroller STM32F**

Pokyny pro vypracování:

Navrhněte a realizujte měřicí jednotku s mikrořadičem řady STM32xx pro měření parametrů analogových bloků. Orientujte se na využití převodníků D/A a A/D a obsažených na čipu STM32xxx, které budou sloužit pro generování testovacích signálů a digitalizaci výstupních signálů analogových bloků. Ověřte statické i dynamické vlastnosti převodníků A/D a D/A v mikrořadičích STM32xxx a zjistěte limity jejich použití i velikost zkreslení při digitalizaci a generování signálů.

S využitím mikrořadiče STM32F3xx sestavte malý systém, který má dvoukanálově generovat napěťové průběhy pro testování objektu a současně tříkanálově synchronně digitalizovat jeho výstupní signály. Realizujte funkci číslicového synchronního detektoru. Funkci měřicí jednotky ověřte při vyhodnocování vlastností feromagnetických materiálů. Vytvořte potřebné programové vybavení pro vlastní mikrořadič i pro nadřazené PC, které má sloužit pro ovládání experimentu i zpracování výsledků. Posuďte vhodnost využití programu LabView pro ovládání jednotky.

Seznam odborné literatury:

- [1] ARM Limited: Cortex-M4, Revision r0p1, Technical Reference Manual, ARM DDI 0439C, 2010
- [2] ARM Limited: ARMv7-M Architecture Reference Manual, ARM DDI 0403D (ID021310), 2010
- [3] STMicroelectronics: Reference Manual, RM0090, doc ID 018909 Rev3

Vedoucí diplomové práce: doc. Ing. Jan Fischer, CSc.

Datum zadání diplomové práce: 29. listopadu 2012

Platnost zadání do¹: 30. června 2014


Prof. Ing. Vladimír Haasz, CSc.
vedoucí katedry




Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 29. 11. 2012

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Anotace

Tato diplomová práce je zaměřena na kompletní návrh a implementaci systému pro sběr dat s mikrokontrolérem STM32F303 od hardwarové části přes firmware až po uživatelskou aplikaci určenou k jeho řízení. Dále se zabývá testováním vlastností A/D a D/A převodníků použitého mikrokontroléru a okrajově měřením vlastností feromagnetických materiálů.

Abstract

This diploma thesis is aimed at complete design and implementation of a data acquisition system with STM32F303 microcontroller unit from hardware circuitry to firmware and user application intended for its control. It also goes into microcontroller unit's analog-to-digital and digital-to-analog converter testing and peripherally into ferromagnetic materials properties measurement.

Čestné prohlášení autora práce

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 dne

.....
podpis autora práce

Tímto bych chtěl poděkovat svému vedoucímu diplomové práce panu docentovi Janu Fischerovi za to, že si vždy našel čas a energii na to vysvětlit mi věci, které považuje za důležité, a za to, že mi vyjadřoval důvěru i pochvalu tehdy, když jsem si je zasloužil.

Obsah

Seznam obrázků	ix
Seznam tabulek	x
Seznam použitých zkratk	xi
1 Úvod	1
2 Rozbor problematiky	2
2.1 DAQ jednotky	2
2.2 Měření hysterézní smyčky feromagnetických materiálů	3
2.3 Generování signálů	3
2.4 Synchronní detektor – Lock-in zesilovač	4
2.5 Testování D/A a A/D převodníků	4
2.6 Volba programovacích jazyků	4
3 Využití procesorů STM32F v LabVIEW	5
3.1 Komunikace přes UART/COM	5
3.2 Komunikace přes virtuální COM port	5
3.3 Komunikace přes specifický USB protokol	7
4 Návrh hardwaru	9
4.1 Výstupní obvod	9
4.1.1 Oddělovač	9
4.1.2 Filtr	9
4.1.3 Posouvač/zesilovač úrovní výstupního signálu	10
4.1.4 Budič	11
4.2 Vstupní obvod	12
4.2.1 Posouvač/zesilovač pro přizpůsobení vstupního napětí	12
4.2.2 Ochranný obvod vstupu A/D převodníku	14
4.3 Realizace použitá při měření feromagnetik	15
5 USB komunikační protokol pro komunikaci s jednotkou	19
6 Firmware pro STM32F303	22
6.1 Popis používaných hardwarových prostředků	22

6.2	Popis programového vybavení použitého pro programování firmwaru	22
6.3	Popis vlastní implementace	24
6.3.1	Funkce <code>main</code> – inicializace	24
6.3.2	Nastavení přerušení	24
6.3.3	Obsluhy přerušení	25
6.3.4	Konfigurace hodin	26
6.3.5	Popis rozhraní pro komunikaci s PC (USB)	28
6.3.6	Dynamická alokace paměti pro data A/D i D/A převodníků (<code>SetDataBuffers</code>)	30
6.3.7	Programování A/D převodníků procesoru STM32F303	31
6.3.8	Inicializace DMA pro přenos dat A/D převodníků (<code>ADC_DMA_init</code>)	34
6.3.9	Reset DMA (<code>ADC_DMA_reset</code>)	35
6.3.10	Spouštění vzorkovací sekvence	35
6.3.11	Generace signálů	36
7	Obslužný software pro PC	39
7.1	Ovladač jednotky (<i><code>interfacing.c/.h</code></i>)	39
7.1.1	Obecný popis	39
7.1.2	Referenční manuál k ovladači jednotky	39
7.2	Zdrojový soubor s funkcemi zpracování dat (<i><code>utils.c/.h</code></i>)	46
7.2.1	Výpočet frekvence a fáze (<code>UTS_FrequencyPhase</code>)	47
7.2.2	Lock-in zesilovač (<code>UTS_Lockin</code>)	48
7.2.3	Výpočet hysterézní smyčky feromagnetika (<code>UTS_HysteresisLoop</code>)	49
7.3	Zdrojový soubor grafického uživatelského rozhraní (<i><code>gui.c/.h</code></i>)	51
7.3.1	Popis uživatelského rozhraní	51
7.3.2	Poznámky k implementaci GUI	53
8	Lock-in zesilovač a test jeho implementace	55
8.1	Princip funkce	55
8.2	Způsob implementace lock-in zesilovače	56
8.3	Testování implementace lock-in zesilovače	56
9	Měření vlastností A/D převodníků	59
9.1	A/D převodníky s postupnou aproximací a jejich vlastnosti	59
9.2	Měření statických vlastností A/D převodníků	60
9.2.1	Stanovení offsetu a reference	61
9.2.2	Měření nelinearit A/D převodníků	64

9.2.3	Vlastní měření nelinearit A/D převodníku	68
9.3	Měření dynamických vlastností A/D převodníků	73
9.3.1	Přenosová frekvenční charakteristika A/D převodníků	73
9.3.2	Vyhodnocení jitteru A/D převodníků	77
10	D/A převodníky, jejich vlastnosti a měření	82
10.1	D/A převodníky a jejich vlastnosti	82
10.2	Statické parametry	84
10.3	Dynamické parametry	86
11	Vlastnosti feromagnetik a jejich měření	89
11.1	Popis vlastností a odvození vztahů	89
11.2	Měření hysterézních smyček modulem se STM32F303	91
12	Vyhodnocení práce	94
A	Data naměřená při testování lock-in zesilovače	96
B	Záznamy signálů a hysterézní smyčky	98

Seznam obrázků

2.1	Struktura zásuvných měřicích karet (převzato z [16])	2
3.1	Schéma zapojení systému pro měření charakteristik optronů	6
3.2	Čelní panel virtuálního přístroje pro měření charakteristik optronů pomocí STM32F4 implementovaného v NI LabVIEW	7
4.1	Blokové schéma výstupního obvodu měřicí jednotky	9
4.2	Posouvač napětí generovaného D/A převodníkem	10
4.3	Výstupní budič	12
4.4	Blokové schéma výstupního obvodu měřicí jednotky	12
4.5	13
4.6	15
4.7	Schéma zapojení pro měření hysterézní smyčky feromagnetika, ADC1, ADC2 a DAC1 značí příslušné piny STM32F3-Discovery	16
4.8	Zapojení výstupního obvodu pro úpravu signálu použité při měření feromagnetik	17
4.9	Zapojení vstupního obvodu pro úpravu signálu použité při měření feromagnetik	17
6.1	Schéma propojení hardwaru a použitého softwaru při programování	23
6.2	Clock tree procesoru STM32F303 (převzato z [3])	27
6.3	Způsob alokace bufferů pro A/D a D/A převodníky	31
6.4	Registry délky odběru vzorků (převzato z [3], graficky upraveno)	33
6.5	Nastavení délky doby odběru vzorků	33
6.6	Spodní dva byty registru předděliček A/D převodníků a PLL (horní 2 byty rezervované; převzato z [3])	34
6.7	Nastavení předděliček A/D převodníků	34
6.8	Grafické znázornění implementace DDS	37
7.1	Struktura/příklad konfiguračního souboru používaného pro měření hysterézních smyček, hodnoty odpovídají zapojení a vzorku používanému v této práci	51
7.2	Demonstrační grafické uživatelské rozhraní softwaru pro ovládání měřicí jednotky se STM32F303	53
8.1	Schéma pro testování lock-in zesilovače	56
9.1	Blokové schéma A/D převodníku s postupnou aproximací (převzato z [1])	59
9.2	Schéma obvodu SAMPLE/HOLD (převzato z [4])	60
9.3	Demonstrace odečtu hodnot pro zjištění offsetu	62
9.4	63
9.5	64
9.6	Generátor trojúhelníkového průběhu použitý pro testování A/D převodníku metodou histogramu	65

9.7	Schéma pro měření nelinearit A/D převodníků	66
9.8	Grafické znázornění modelu signálu na výstupu RC článku buzeném obdélníkovým signálem	66
9.9	Histogram dat získaných měřením testovacího průběhu při měření nelinearit A/D převodníku procesoru STM32F3, červeně zobrazen klouzavý průměr zvýrazňující trend dat	69
9.10	Detailní zobrazení klouzavého průměru (délka průměrování 401) histogramu z obrázku 9.9	70
9.11	Teoretická hustota pravděpodobnosti výskytu vzorku v kvantu N testovacího signálu pro testování A/D převodníků (normalizována na Volty, zde osa v kvantech)	71
9.12	Diferenciální nelinearita A/D převodníku procesoru STM32F303 (absolutní maximum 0,377 mV – 0,53 LSB – mezi v přechodu mezi kódovými slovy 3391 a 3392)	72
9.13	Detail diferenciální nelinearity A/D převodníku procesoru STM32F303	72
9.14	Integrální nelinearita A/D převodníku procesoru STM32F303 (absolutní maximum 1,33 mV – 1,86 LSB – kódovém slově 1023)	73
9.15	Posouvací obvod pro přidání stejnosměrné složky 1,5 V ke střídavému signálu podle [11], vpravo pohled do dutinek určených k nasazení na STM32F3-Discovery	74
9.16	Schéma propojení generátoru, osciloskopu, posouvacího obvodu a STM32F3-Discovery pro měření frekvenční charakteristiky jeho A/D převodníku	74
9.17	Frekvenční charakteristika A/D převodníku mikrokontroléru STM32F303 při vzorkovací frekvenci 281,25 kHz	76
9.18	Frekvenční charakteristiky A/D převodníků mikrokontrolérů STM32F051, STM32F303 a STM32F407 při různých vzorkovacích frekvencích	77
9.19	Průběh využitý pro stanovení jitteru A/D převodníku s červeně vyznačenou ruční aproximací (frekvence vstupního signálu 281,27 MHz)	78
9.20	Demonstrace vlivu jitteru na kvalitu rekonstrukce měřeného signálu	79
9.21	Detail signálu použitého pro analýzu jitteru s označenými body s největší vzájemnou diferencí, červeně je zobrazena aproximace signálu s významnými argumenty, zeleně odečet argumentu pomocí funkce arcsin	80
10.1	Obrázek převodní charakteristiky 3-bitového D/A převodníku s kótami pro stanovení INL, DNL, offsetu a chyby zesílení (převzato z [7])	83
10.2	Schéma použité pro měření statických parametrů D/A převodníku STM32F3	84
10.3	Průběh INL D/A převodníku procesoru STM32F3 (absolutní maximum 1,84 mV – 2,56 LSB – v kódovém slově 471)	85
10.4	Průběh DNL D/A převodníku procesoru STM32F3 (absolutní maximum 0,8 mV – 1,11 LSB – mezi kódovými slovy 3734 a 3735)	86
10.5	Schéma použité pro měření statických parametrů D/A převodníku STM32F3	87
10.6	Odezva výstupu D/A převodníku na skok z nuly na plný rozsah	87
11.1	Obecná hysterézní smyčka	90
11.2	Hysterézní smyčka zobrazená pomocí aplikace vyvinuté pro ovládání jednotky se STM32F303	92

- B.1 Průběhy magnetizačního proudu a výstupního napětí nahoře a k nim příslušející vypočtené hysterézní smyčky dole. Pro lepší grafickou kvalitu generováno Matlabem. . . . 98
- B.2 Průběhy magnetizačního proudu a výstupního napětí nahoře a k nim příslušející vypočtené hysterézní smyčky dole. Pro lepší grafickou kvalitu generováno Matlabem. . . . 99
- B.3 Průběhy magnetizačního proudu a výstupního napětí nahoře a k nim příslušející vypočtené hysterézní smyčky dole. Pro lepší grafickou kvalitu generováno Matlabem. . . . 100

Seznam tabulek

6.1	Přerušeni nakonfigurovaná pro řízení běhu firmwaru, nižší číslo značí vyšší prioritu	25
6.2	Device descriptor pro enumeraci jednotky	28
6.3	Configuration, interface a endpoint descriptory pro enumeraci jednotky	29
7.1	Tabulka výběru předděličky A/D převodníku	43
7.2	Tabulka výběru délky doby odběru vzorku v počtu cyklů hodin A/D převodníku	44
7.3	Tabulka vzorkovacích frekvencí pro všechny kombinace délek doby odběru vzorku a předděličky A/D převodníků (pro rozlišení 12 bitů)	44
8.1	Zpracovaná data pořízená při testování algoritmu lock-in zesilovače, každý řádek odpovídá jedné kombinaci frekvence měřeného signálu f_{sig} , vzorkovací frekvence f_S a SNR měřeného signálu	58

Seznam použitých zkratk

ADC	Analog-to-Digital Converter
CAN	Controller Area Network
DAC	Digital-to-Analog Converter
DDS	Direct Digital Synthesis
DMA	Direct Memory Access
DNL	Differential Non-Linearity
ESD	Electro-Static Discharge
FFT	Fast Fourier Analysis
GPIO	General Purpose Input Output
GUI	Graphical User Interface
INL	Integral Non-Linearity
ISR	Interrupt Status Register
LSB	Least Significant Bit
MSB	Most Significant Bit
NI	National Instruments
PLL	Phase Lock Loop
SAR	Successive Approximation Register
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
VI	Virtual Instrument

1 Úvod

V této práci je detailně popsán kompletní návrh systému pro sběr dat s mikrokontrolérem (MCU) STM32F303. Multifunkční jednotky pro sběr dat mají několik nesporných výhod. Narozdíl od běžných měřicích přístrojů jsou to nejen snadno přenosné a kompaktní, ale i podstatně levnější varianty pro nenáročná měření. Jejich přirozenou vlastností je nutnost připojení k počítači, který tak nahrazuje ovládací panel přístroje a jeho ovládací prvky. Jelikož je veškerá interakce uživatele s přístrojem založena na softwarové bázi, lze dokonale přizpůsobit uživatelské rozhraní aktuální funkci, kterou zařízení vykonává.

Touto ideou je inspirována většina této práce, která byla zadána v reakci na neustále rostoucí schopnosti jednočipových mikrokontrolérů, zejména pak těch vyvíjených firmou ST Microelectronics nesoucích označení STM32. Tyto MCU jsou vybaveny sadou periférií, která v závislosti na řadě a typu výrobku může obsahovat řadiče k průmyslovým (např. CAN, Ethernet) nebo interním standardizovaným sběrnicím (SPI, S²C atd.) či k rozhraní známému z běžné konzumní elektroniky. Jedná se samozřejmě o USB, které je klíčovým prvkem umožňujícím jednoduché připojení k běžnému osobnímu počítači. Mezi další, pro využití v měřicí technice ještě mnohem důležitější, periferie dostupné v portfoliu STM32 patří A/D a D/A převodníky, časovače a čítače, komparátory a další. Jmenovaná hardwarová vybava těchto mikrokontrolérů jim dává možnost stát se podstatně levnější a otevřenější alternativou ke komerčně dostupným DAQ modulům.

Úkolem této práce je právě vytvoření veškerého softwarového vybavení i elektronických součástí k tomu, aby se z mikrokontroléru STM32F303, osazeného do vývojového kitu STM32F3-Discovery, stala jednotka pro sběr dat, která bude schopna synchronního vzorkování třemi A/D převodníky a zároveň dvoukanálové generace testovacích signálů. Práce však zachází do problematiky hlouběji než vyžaduje zadání diplomové práce a implementuje nepovinné, ale užitečné funkce usnadňující práci s měřicí jednotkou.

Softwarem, který je nutné vyvinout je konkrétně firmware pro MCU, ovladač zprostředkávající přístup k němu přes sběrnici USB a nakonec i uživatelská aplikace s grafickým uživatelským rozhraním. Samotný návrh softwarového vybavení by pro praktickou použitelnost modulu nestačil, proto je dále zapotřebí navrhnout hardwarovou nadstavbu určenou k přizpůsobování napěťových úrovní jak výstupních, tak i vstupních signálů, dále pak k ochraně vstupů. Návrh elektroniky musí být ověřen a otestován, k čemuž je využit praktický úkol: měření hysterézní smyčky předloženého feromagnetika. To s sebou přináší nutnost implementace specializovaného algoritmu pro zpracování dat. Další specializovaný algoritmus musí být navržen pro realizaci číslicového lock-in zesílení, které nemá předepsané v rámci této práce žádné praktické využití, ale jeho funkci je, samozřejmě, nutné ověřit a otestovat.

Značná část práce je také věnována testování vlastností A/D a D/A převodníků použitého MCU, při kterém je využito z velké části už dříve rozvinuté teorie, která je rozšířena o vlastní poznatky, které jsou prakticky využity v případech, kdy není využití původní nezměněné teorie možné nebo vhodné.

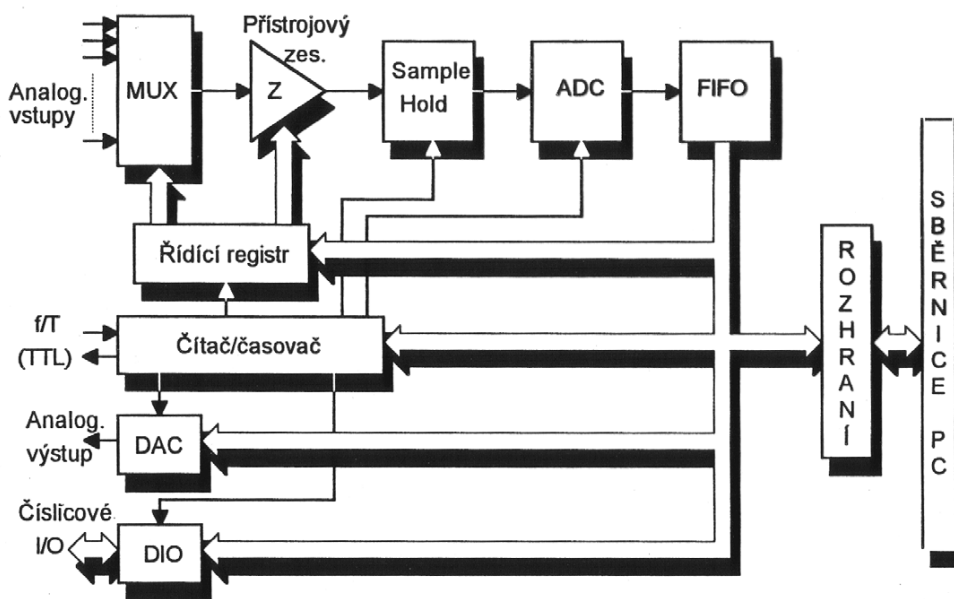
2 Rozbor problematiky

V této kapitole jsou diskutovány body zadání a je zde nastíněn koncept této diplomové práce. Dále se tato část snaží uvést souvislosti mezi jednotlivými body zadání.

2.1 DAQ jednotky

Konstrukci velice jednoduchého měřicího/testovacího pracoviště umožňují tzv. DAQ (Data Acquisition) jednotky, chcete-li moduly. Tato zařízení umožňují koncentrovat schopnosti běžně v laboratoři používaných přístrojů, jako je voltmetr, signální generátor či logický analyzátor, do jediného, obvykle nepoměrně menšího uspořádání. Ačkoliv DAQ moduly nenabízejí natolik dobré parametry (např. po stránce přesnosti, obnovovací frekvence generátoru, malého zkreslení oscilátorem generované sinusoidy) jako specializované přístroje, jejich malé rozměry, relativně nízká cena a jednoduchá konfigurovatelnost je předurčují pro využití v nenáročných aplikacích a nesespecializovaných měřeních, ve kterých je vhodná vyšší flexibilita.

Známým výrobcem DAQ jednotek je například firma National Instruments, která nabízí širokou škálu těchto zařízení. Jejich strukturu a možnosti nejlépe vystihuje obrázek 2.1 převzatý z přednášek předmětu Virtuální přístroje vyučovaného na katedře měření^[16]. Obrázek slouží k popisu zásuvných karet, struktura je však totožná s externími moduly připojovanými např. přes sběrnici USB.



Obrázek 2.1: Struktura zásuvných měřicích karet (převzato z ^[16])

Prolistováním katalogového listu k předmětu této práce, mikrokontroléru firmy ST Microelectronics STM32F303 se lze dopátrat ke zjištění, že tento jediný chip obsahuje všechny součásti, které

jsou zobrazeny na obrázku 2.1. Ve srovnání s produkty jmenované společnosti NI obsahuje procesor STM32F303 standardní počet analogových výstupů, tedy dva. Navzdory nižším řadám těchto produktů však STM32F303 obsahuje 4 A/D převodníky, které jsou schopny simultánního vzorkování a dohromady mají 39 multiplexovaných vstupů^[2]. Digitální vstupy/výstupy jsou u mikrokontroléru samozřejmé.

Shrnuté poznatky naznačují, že využití moderních mikrokontrolérů integrujících v sobě výpočetní jádro spolu s rozsáhlým množstvím nejen číslicových, ale i analogových periférií, je vhodnou alternativou k výrazně nákladnějším komerčně dostupným DAQ modulům. Je však třeba brát na zřetel, že mikrokontrolér sám o sobě není odolný vůči vnějším vlivům jako je přepětí na vstupech a ESD. Při začlenění do měřicího systému či předání neznalé obsluze tak může snadno dojít k jeho poškození, proto je nutné navrhnout dodatečnou elektroniku pro ochranu jeho vstupů.

2.2 Měření hysterézní smyčky feromagnetických materiálů

Demonstrace funkčnosti této jednotky je realizována na feromagnetickém materiálu ve formě toroidu opatřeného budicím a měřicím vinutím. Tento toroid představuje měřený objekt a návrh hardwarové části modulu musí respektovat jeho vlastnosti.

Jelikož D/A převodníky STM32F303 mají velice omezené možnosti po stránce napěťového rozsahu i proudového zatížení, není možné je samy o sobě využít ke generaci testovacího signálu pro měření vlastností feromagnetik. S rozsahem vstupních napětí to samé platí pro A/D převodníky. Pro obě tyto periférie musí být rozšířen napěťový rozsah a umožněno generovat/vzorkovat i napětí záporná vůči jejich napájení.

Pro úspěšné splnění tohoto úkolu bude nutné navrhnout patřičnou hardwarovou nadstavbu a implementovat zpracování dat vyplývající z teorie elektromagnetismu.

2.3 Generování signálů

Základním požadavkem na multifunkční měřicí jednotku je možnost generování testovacích signálů. Tím je umožněno testování obvodů, jejichž vstupy je zapotřebí budit ať už stejnosměrným napětím nebo signálem o určité frekvenci a průběhu.

Předepsaný hardware nenabízí rozsáhlé možnosti generace signálů, v běžné praxi by bylo možné využít externích D/A převodníků. Na převodníky obsažené v procesoru STM32F3 se cílí z toho důvodu, aby byl vytvořen základ pro konstrukci jednoduchých měřicích jednotek využívající minimální množství komponent a umožňující snadnou konfiguraci či změnu účelu jen změnou firmwaru.

Aby bylo možné generovat libovolný signál, je nutné, aby byl tento signál uložen v paměti mikrokontroléru, výjimku tvoří jednoduché signály jako obdélník či trojúhelník, které lze jednoduše počítat za běhu. Otázkou je, jak adresovat tyto vzorky. V zásadě lze uvažovat o těchto možnostech

- Adresovat vzorky jeden za druhým – nejjednodušší varianta, která ovšem umožňuje změnu frekvence signálu pouze změnou obnovovací frekvence generátoru a maximální frekvence signálu je dána délkou jeho obrazu v paměti.
- Rovnoměrnými skoky vynechávat vzorky – tato metoda by vedla k rozšíření možností předchozího

způsobu, bylo by však třeba navrhnout sofistikovaný algoritmus určující obnovovací frekvenci i délku skoku.

- Algoritmus DDS – Direct Digital Synthesis – známá a rozšířená metoda adresování vzorků v paměti, nabízí jednoduchou implementaci s velice citlivým nastavením frekvence při konstantní obnovovací frekvenci.

Poslední jmenovaná možnost bude implementována jako nejvhodnější a nejefektivnější metoda pro generování testovacích průběhů.

2.4 Synchronní detektor – Lock-in zesilovač

Lock-in zesilovač je významným zařízením umožňujícím extrakci užitečného měřonového signálu známé a konstantní frekvence z velice zašuměného průběhu. V této práci je realizována číslicová verze synchronního detektoru. Funkci synchronní detekce jak je známa ze špičkových zařízení nebude zřejmě možné realizovat, bude však implementován její princip. To, jakým způsobem bude realizován, se bude odvíjet od vlastností firmwaru pro STM32F303 vyvíjeného jako jeden z hlavních bodů této práce.

Ačkoliv synchronní detektor nelze použít v měření hysterézních křivek feromagnetik realizovaném v rámci této práce, tato funkce je implementována z toho důvodu, jak již bylo uvedeno v předchozím odstavci, že se jedná o nepostradatelný prvek v některých specifických měřicích aplikacích.

2.5 Testování D/A a A/D převodníků

S uvážením skutečnosti, že se celá práce zabývá generací a vzorkováním signálů, je rozumným požadavkem ověření vlastností použitých A/D i D/A převodníků. Některé důležité vlastnosti převodníků jsou uvedeny v katalogovém listu použitého procesoru^[2], měření jsou určena k bližší charakteristice, doplnění informací chybějících v datasheetu a stanovení limitů, pro které je jejich použití vhodné.

2.6 Volba programovacích jazyků

Důležitým rozhodnutím před započatím vývoje veškerého softwaru je, jaké programovací jazyky se budou pro kterou vrstvu (firmware, ovladač, koncová aplikace) používat.

Ač lze STM32 programovat v assembleru, tato možnost by vedla k velice zdlouhavému vývoji. Možnost použití jazyka C programování podstatně urychluje a v případě potřeby tento jazyk nabízí stejnou kontrolu nad hardwarem a při použití přímých zápisů na konkrétní místa v paměti je rozdíl v rychlosti výsledného programu minimálně diskutabilní.

Programování PC aplikace v dnešní době výkonných počítačů posouvá výkon kódu do ústraní a při výběru programovacího jazyka se uvažuje nad určením (pro jaké platformy) a dobou vývoje. Důležitým kritériem byla při výběru programovacího jazyka možnost vyvíjet software v Linuxu, což téměř automaticky zajišťuje snadnou portovatelnost na Windows, která v opačném směru bývá problematická. Vzhledem ke zkušenostem s jazykem C a jeho předvídatelnosti a možností byl tento jazyk rovněž zvolen pro programování ostatních vrstev softwaru.

3 Využití procesorů STM32F v LabVIEW

Tato kapitola pojednává o prozkoumaných možnostech komunikace NI LabVIEW s procesory STM32F. Uvedené poznatky lze bezesporu aplikovat i na jiné procesory vybavené příslušnými periferiemi, tedy USB řadičem či řadičem UART.

3.1 Komunikace přes UART/COM

Drtivá většina mikrokontrolérů obsahuje hardwarové rozhraní UART, které lze po opatření převodníkem napěťových úrovní CMOS \leftrightarrow RS-232 (například ADM3232) využít pro komunikaci buďto přes standardní sériovou linku počítače (COM port) nebo použít převodník USB \leftrightarrow RS-232 u nových počítačů postrádajících sériové rozhraní.

Tato metoda byla využita při měření frekvenční charakteristiky A/D převodníku STM32F0-Discovery a STM32F3-Discovery v prvotní verzi firmwaru určené pouze pro tento účel.

Sériová komunikace lze v LabVIEW realizovat velice jednoduše použitím funkcí VISA Write a VISA Read z knihovny VISA.

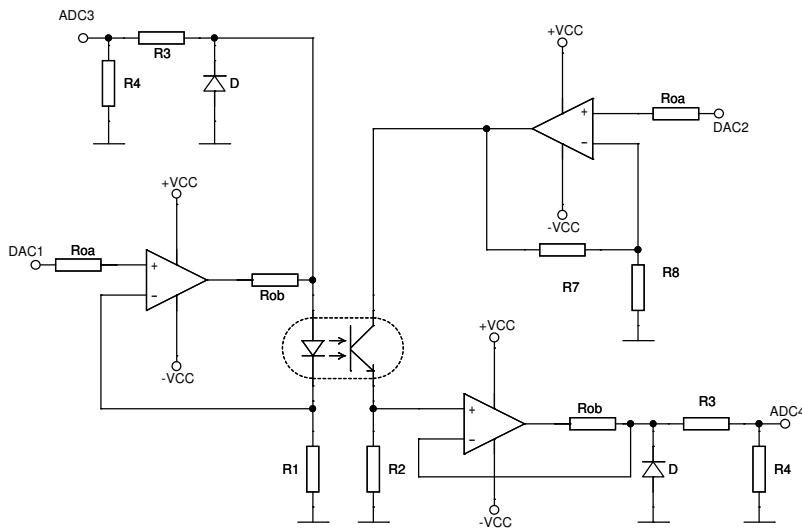
3.2 Komunikace přes virtuální COM port

Tato metoda představuje určitý mezistupeň mezi předchozím zmíněným a následujícím způsobem komunikace a byla použita při realizaci systému pro měření charakteristik optronů se STM32F4-Discovery.

Virtuální COM port není samozřejmostí napříč spektrem dostupných mikrokontrolérů, vyžaduje totiž specifickou konfiguraci řadiče USB, která je ale v dostupných demonstračních projektech implementována. Dále je pak zapotřebí ovladač pro PC od firmy ST Microelectronics, který s jednotkou komunikuje definovaným protokolem a veškerou komunikaci zprostředkovává operačnímu systému (dostupný pouze pod Windows), ve kterém s ní lze pracovat jako s běžnou sériovou linkou.

Z tohoto důvodu lze s jednotkou komunikovat v prostředí LabVIEW způsobem popsáním v předchozí podkapitole.

Pro úplnost je na obrázku 3.1 uvedeno schéma zapojení zmíněného modulu, jsou použity oba kanály D/A převodníku a dva A/D převodníky. Modul pracuje staticky, tzn. že v každém kroku jsou nastavena stejnosměrná napětí D/A převodníky a vzorkovány měřené hodnoty A/D převodníky. Z principu ale nic nebrání tomu generovat dynamické průběhy nebo vzorkovat dlouhé sekvence vzorků, jako tomu bylo při měření frekvenční charakteristiky procesoru STM32F4, která je zobrazena na obrázku 9.18. Firmware pro STM32F407 s nakonfigurovaným virtuálním portem použitý pro všechny aplikace zmíněné v této práci je dostupný na příloženém CD v adresáři *Zdrojove_kody/F4-VCP*.



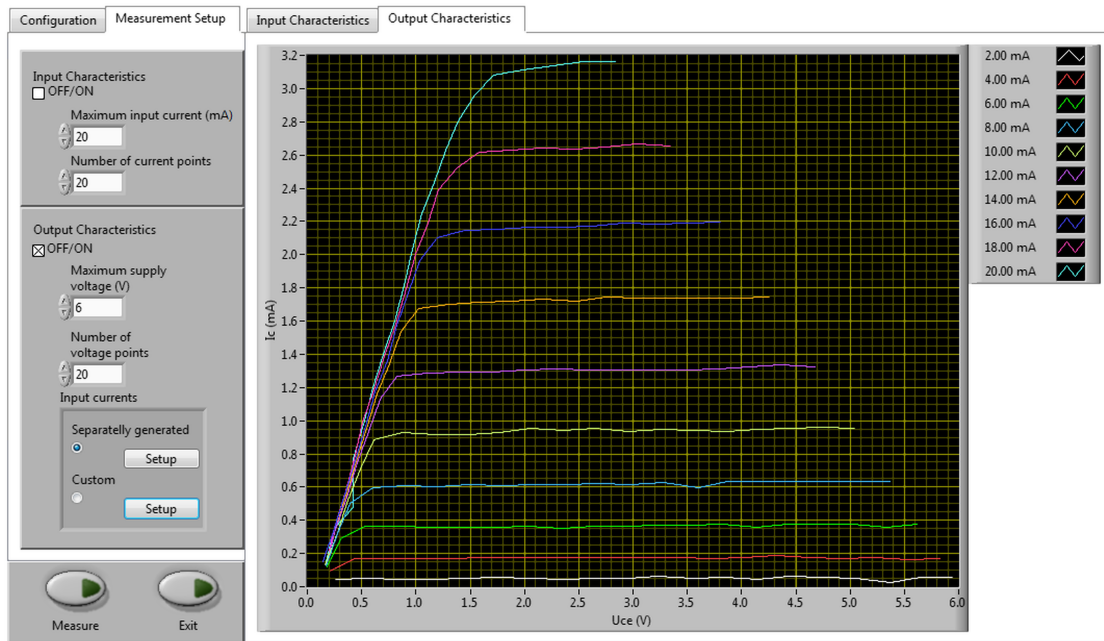
Obrázek 3.1: Schéma zapojení systému pro měření charakteristik optronů

Hodnoty součástek použitých pro ladění a testování:

- OZ – TL084
- Odporů R_{ob} nezapojeny
- $R_{oa} = 1 \text{ k}\Omega$
- $R_4 = 100 \Omega$, $R_3 = 2 \times 150 \Omega$ v sérii – dělič $1/4$
- $R_7, R_8 = 1 \text{ k}\Omega$ – zesílení 2
- $R_1 = 150 \Omega$ – při výstupu DAC1 max. 3 V odpovídá maximálnímu proudu LED 20 mA
- $R_2 = 1 \text{ k}\Omega$ – výstupní proud v mA roven měřenému napětí ve V
- D – usměrňovací diody, $I_{AK,max}$ min. 1 A

Systém byl testován na optronu WK 16412.

Program napsaný v LabVIEW byl realizován pomocí bloků *Source.vi* a *Voltmeter.vi*, které na základě argumentů nesoucích číslo D/A nebo A/D převodníku, případně hodnotu nastavovaného napětí, vykonávají potřebnou komunikaci po virtuální sériové lince a na výstupu zprostředkovávají data snadno zpracovatelná zbytkem programu. Zdrojové soubory pro LabVIEW k systému jsou umístěny na CD ve složce *Zdrojove_kody/LabVIEW/Mereni_optronu*. Vzhled uživatelského rozhraní s charakteristikami změřenými pomocí tohoto systému je na obrázku 3.2.



Obrázek 3.2: Čelní panel virtuálního přístroje pro měření charakteristik optronů pomocí STM32F4 implementovaného v NI LabVIEW

3.3 Komunikace přes specifický USB protokol

Možnosti komunikace LabVIEW přímo s endpointy jsou jmenovány na webu NI v příručce k ovládání přístrojů přes USB^[14]. Jelikož se při návrhu komunikačního protokolu k navrhovanému modulu (viz 5) nepočítalo s využitím v LabVIEW, bylo zapotřebí malého zásahu do ovladače USB řadiče STM32F3. Knihovna VISA totiž nepodporuje přenosy interrupt OUT, které jsou hojně využívány pro řízení jednotky. Je tedy nutné využití control přenosů, které se ale s knihovnou pro USB řadič obtížně zpracovávají.

Proto byl modifikován ovladač USB řadiče, aby reagoval na jinak nepoužitý setup request 0x10. Při přijetí tohoto požadavku přepne číslo D/A převodníku k nastavení dat průběhu na 3, čímž rutina přerušení při přijetí bulk OUT dat zpracovává tyto data jako řídicí data a nikoliv data pro nastavení průběhů pro D/A převodník. V případě, že program psaný v LabVIEW požaduje nastavení průběhu D/A převodníku, vyšle požadavek na bulk endpoint 2 podle komunikačního protokolu a poté vyšle na stejný endpoint nastavovaná data. Poté je ovšem zapotřebí před vysláním dalšího příkazu opět přepnout EP2 do příkazového režimu uvedeným setup requestem.

K přepnutí do režimu pro LabVIEW lze použít virtuální přístroj *Zdrojove_kody/LabVIEW/STM32Driver/AllowLabviewAccess.vi*, toto se doporučuje provést před každým novým příkazem. Poté lze zaslat příkaz podle komunikačního protokolu uvedeného v kapitole 5 s výjimkou, že se nepoužívá endpoint 1, ale endpoint 2. Funkčnost této metody byla ověřena a byly implementovány virtuální přístroje ve stejné složce, v jaké se nachází první zmíněný VI. Vývoj tohoto

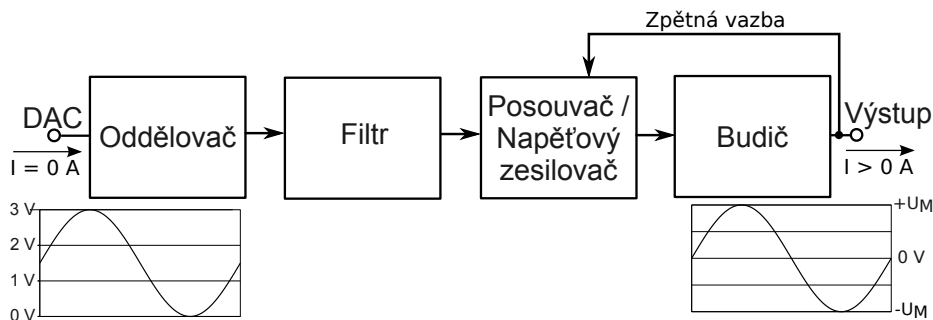
ovladače byl ale zastaven a je možné, že tyto VI neodpovídají aktuální verzi protokolu. S popisem uvedeným v této kapitole však není problém ovladač pro LabVIEW dopracovat, cílem této práce nebylo tento ovladač implementovat, pouze nalézt a ověřit možnosti jeho řešení.

4 Návrh hardwaru

V této části je popsán postup návrhu výstupních a vstupních částí pro měřicí modul se STM32F3. D/A a A/D převodníky obsažené na chipu pracují pouze s napětími od 0 V do referenčního napětí (asi 2,95 V). Měřicí modul však musí být schopen generovat, resp. měřit, střídavá napětí. V obou podkapitolách je vysvětlen obecný postup návrhu a nakonec je uvedena konfigurace, která byla postavena na nepájivém vodivém poli a využita pro měření hysterézní smyčky toroidního feromagnetického vzorku.

4.1 Výstupní obvod

Po tomto obvodu se požaduje, aby odstranil stejnosměrnou složku signálu generovaného D/A převodníkem, signál zesílil a zajistil, že D/A převodník nebude zatěžován. Funkce výstupního obvodu je demonstrována obrázkem 4.1, který je zároveň blokovým schématem.



Obrázek 4.1: Blokové schéma výstupního obvodu měřicí jednotky

4.1.1 Oddělovač

Oddělovač slouží k tomu, aby nebyl zatěžován D/A převodník, který má výstupní odpor 15 k Ω (viz datasheet^[2]) a jedná se tedy o velice měkký zdroj napětí. Oddělovač je tvořen jediným operačním zesilovačem zapojeným jako sledovač napětí.

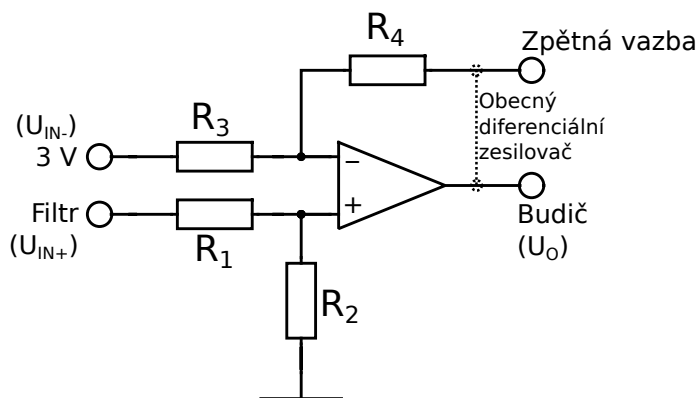
4.1.2 Filtr

Jelikož signál generovaný D/A převodníky obsahuje výrazné nespojitosti, které mohou být v některých aplikacích nežádoucí (mezi ně patří i magnetická měření realizovaná v této práci), je nutné zařadit do řetězce dolnoproustný filtr, který skoky v napětí odstraní. Filtr by měl mít horní mezní frekvenci dostatečně nad maximální frekvencí, kterou se zamýšlí generovat (včetně významných harmonických při generaci nesinusových signálů). Odstup těchto frekvencí závisí na řádu a typu aproximace použitého filtru. Řád filtru dále určuje, jak dobře lépe se potlačí kvantizační šum.

4.1.3 Posouvač/zesilovač úrovní výstupního signálu

Procesory STM32 obsahují nesymetricky napájené D/A převodníky, u prakticky použitelného generátoru průběhů se však požaduje, aby výstup mohl být symetrický. K tomuto účelu byl navržen posouvač napětí. Tímto obvodem se od výstupu D/A převodníku odečte polovina jeho referenčního napětí tak, aby bylo možné generovat napětí obou polarit o stejné velikosti.

Při návrhu posouvače se uvažovalo o použití oddělovacího kondenzátoru, který by stačil pro účel měření hysterézních smyček, dokonce by pro něj byl vhodnější než navržený posouvač. Navrhovaná jednotka má však za úkol být univerzální, tj. umožňovat i stejnosměrný výstup. Nepochybně by bylo možné navrhnout sofistikovanější elektroniku, která by umožňovala dynamicky přepínat mezi různými metodami oddělování stejnosměrné složky a offsetování výstupu. Vzhledem k rozsahu práce však nebylo možné věnovat této části o mnoho více času. Nakonec byl tedy navržen posouvač s operačním zesilovačem (obrázek 4.2). Jedná se v principu o diferenciální zesilovač (DZE) s OZ.



Obrázek 4.2: Posouvač napětí generovaného D/A převodníkem

Pro odvození se uvažuje standardní zapojení DZE (ve schématu naznačeno propojkou čárkovaně). Výstupní napětí U_O je rovno

$$U_O = U_{IN+} \frac{R_2(R_3 + R_4)}{R_3(R_1 + R_2)} - U_{IN-} \frac{R_4}{R_3}. \quad (4.1)$$

Požaduje se, aby diferenciální zesilovač přenášel výstupní napětí D/A převodníku s jednotkovým zesílením. Střed rozsahu A/D převodníku je roven 1,5 V, což je hodnota, kterou je zapotřebí odečíst. Vzhledem k tomu, že na STM32F3-Discovery je přítomen stabilizátor s výstupem 3 V, který je vyveden na header, je výhodné toto napětí využít k posuvu. Zesílení pro vstup 3 V je tedy $-\frac{1}{2}$. Požadavky tohoto odstavce shrnuje rovnice

$$U_O = U_{IN+} - U_{IN-} \frac{1}{2}, \quad (4.2)$$

kde jsou obě napětí zastoupena symboly použitými pro odvození DZE podle obrázku 4.2. Položením pravých stran rovnic 4.1 a 4.2 do rovnosti se získá soustava dvou rovnic o čtyřech neznámých

$$\begin{aligned}\frac{R_2(R_3 + R_4)}{R_3(R_1 + R_2)} &= 1, \\ \frac{R_4}{R_3} &= \frac{1}{2}.\end{aligned}\tag{4.3}$$

Úpravou první rovnice se získá tvar

$$\frac{R_2}{R_1 + R_2} + \frac{R_4}{R_3} \frac{R_2}{R_1 + R_2} = 1,\tag{4.4}$$

kde po dosazení $\frac{1}{2}$ za $\frac{R_4}{R_3}$ a dalších úpravách vyjde jednoduchá rovnost

$$2R_1 = R_2,\tag{4.5}$$

která spolu s rovnicí

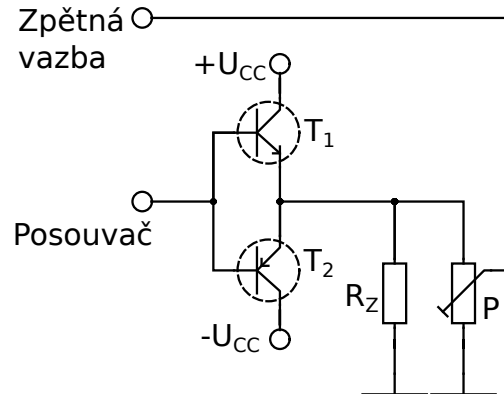
$$R_3 = 2R_4\tag{4.6}$$

dává dva jednoduché a na sobě nezávislé vztahy, kde se jeden odpor zvolí a druhý dopočítá.

V konečném zapojení jsou výstup a zpětná vazba tohoto DZE rozpojeny. Důvodem je, že regulace výstupního napětí se provádí až na výstupu výkonového budiče, který je řízen výstupem právě navrženého DZE.

4.1.4 Budič

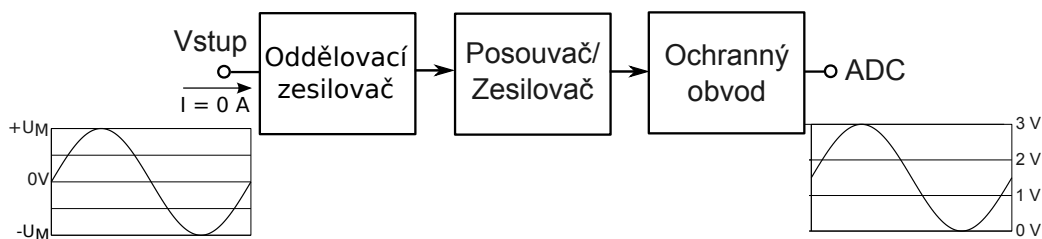
Jelikož operační zesilovač sám o sobě zvládá výstupní proudy do několika desítek mA, je nutné pro buzení řady experimentů tento výstup posílit, k tomuto účelu slouží jednoduchý booster zapojený podle schématu na obrázku 4.3. Tranzistory T_1 a T_2 fungují v režimu sledovače, neposilují tedy signál napěťově ale pouze proudově. O napěťové zesílení se stará předchozí část obvodu, do které je zavedena zpětná vazba z výstupu přes trimr P, který tvoří napěťový dělič umožňující nastavení zesílení od 1 až do neomezené hodnoty. Zátěž je ve schématu reprezentována rezistorem R_Z .



Obrázek 4.3: Výstupní budič

4.2 Vstupní obvod

Funkcí vstupního obvodu je přizpůsobit měřené, obecně střídavé, napětí tak, aby mohlo být vzorkováno A/D převodníkem procesoru STM32F3. Obrázek 4.4 opět ukazuje blokové schéma tohoto obvodu. Na vstupu je opět oddělovací zesilovač s OZ, který může být realizován jako sledovač nebo neinvertující zesilovač. Za ním je zapojen posouvač/zesilovač, který tentokrát posouvá napětí o 1,5 V nahoru, zároveň zeslabuje nebo zesiluje vstupní střídavé napětí a tím určuje rozsah. Poslední částí vstupního obvodu je ochranná část, která zajišťuje, že se za žádných okolností nedostane na vstup A/D převodníku napětí mimo rozmezí $-0,2$ V až $3,5$ V (výrobce povoluje rozmezí $-0,3$ V až 4 V, viz datasheet^[2]).

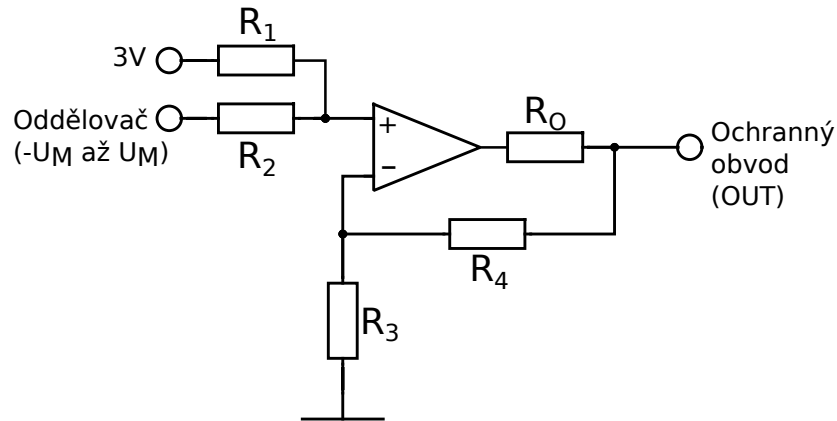


Obrázek 4.4: Blokové schéma vstupního obvodu měřicí jednotky

4.2.1 Posouvač/zesilovač pro přizpůsobení vstupního napětí

Úkolem posouvače/zesilovače je posunout a zesílit nebo zeslabit vstupní, obecně střídavý, signál tak, aby nabýval hodnot 0 až 6 V, tj. dvojnásobné napětí než to, které je A/D převodník schopný měřit. Dvojnásobné napětí oproti možnostem A/D převodníku je zde kvůli návaznému ochrannému

obvodu, který bude vysvětlen v následující podkapitole. Schéma posouvače/zesilovače je na obrázku 4.5.



Obrázek 4.5

Idea tohoto zapojení tkví v tom, že ať přivedeme na vstup jakkoliv velké záporné napětí $-U_M$, vždy lze naváhot odporů R_1 a R_2 tak, aby na neinvertujícím vstupu bylo nulové napětí. Při nárůstu vstupního napětí bude napětí na neinvertujícím vstupu lineárně růst až do horní hranice U_M . Na základě této horní hranice se následně navrhne neinvertující zesilovač tvořený OZ a odpory R_3 a R_4 tak, aby maximální napětí na výstupu tohoto bloku bylo 6 V. Následuje odvození.

Jak bylo řečeno, při vstupu $-U_M$ musí být na neinvertujícím vstupu nulové napětí. Tento požadavek se vyjádří pomocí Théveninova teorému rovnicí

$$3 \text{ V} \frac{R_2}{R_1 + R_2} - U_M \frac{R_1}{R_1 + R_2} = 0, \quad (4.7)$$

která se po několika jednoduchých úpravách převede na tvar

$$R_1 = \frac{3 \text{ V}}{U_M} R_2, \quad (4.8)$$

jež definuje poměr velikostí odporů R_1 a R_2 v závislosti na amplitudě vstupního napětí U_M .

Splnění druhého a posledního požadavku spočívá v zajištění jednotkového zesílení celého bloku pro vstup 3V. Přenos napětí 3V na neinvertující vstup je roven

$$A_{3\text{V,NI}} = \frac{R_2}{R_1 + R_2}, \quad (4.9)$$

přenos z neinvertujícího vstupu na výstup celého bloku je známým vztahem pro neinvertující zesilovač

$$A_{\text{NI,OUT}} = 1 + \frac{R_4}{R_3} \quad (4.10)$$

a celkový přenos mezi vstupem $3V$ a výstupem bloku (OUT) je roven součinu těchto dílčích zesílení, tedy

$$A_{3V,OUT} = A_{3V,NI}A_{NI,OUT} = \frac{R_2}{R_1 + R_2} \left(1 + \frac{R_4}{R_3} \right) = 1. \quad (4.11)$$

Jelikož hodnoty R_1 a R_2 byly určeny v návrhu vstupní části, nic nebrání tomu vyjádřit poměr odporů R_3 a R_4 rovnicí

$$\frac{R_4}{R_3} = \frac{R_1 + R_2}{R_2} - 1. \quad (4.12)$$

Zesílení signálu tohoto bloku je

$$A = \frac{R_1}{R_1 + R_2} \left(1 + \frac{R_4}{R_3} \right) \quad (4.13)$$

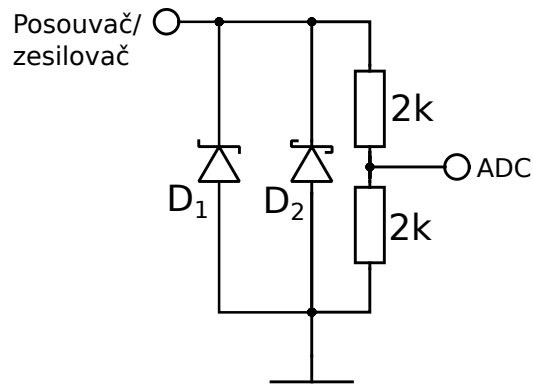
Rezistor R_O hraje důležitou roli ve funkci ochranného obvodu, který je popsán v následující podkapi- tole.

4.2.2 Ochranný obvod vstupu A/D převodníku

Jelikož je STM32F3-Discovery relativně nákladnou součástí celé sestavy, je nutné věnovat zvláštní pozornost ochraně jeho vstupů. Ochranný obvod prezentovaný v následujících odstavcích se uplatní v těchto případech

- pokud se na vstup vstupního obvodu přivede napětí mimo rozsah definovaný oddělovacím zesilovačem a posouvačem/zesilovačem
- pokud dojde k poruše operačního zesilovače posouvače/zesilovače a na jeho výstupu se neočekávaně objeví saturační napětí.

Jeho zapojení je na obrázku 4.6. Zenerova dioda D_1 omezuje seshora napětí na děliči tvořeném oběma rezistory a musí být zvolena tak, aby při nastání nejhorší možné situace (tj. kladné saturační napětí na výstupu OZ posouvače/zesilovače) nepřekročilo napětí v závěrném směru této diody 8 V (tedy 4 V na vstupu ADC). Stejně pravidlo platí u Schottkyho diody D_2 , která převodník chrání proti záporným napětím. Tato musí být opět patřičně dimenzována, aby nebyla přetěžována nadměrným proudem způsobujícím nárůst napětí v propustném směru nad 0,6 V. K omezení proudu diodami slouží rezistor R_O (viz obrázek 4.5).



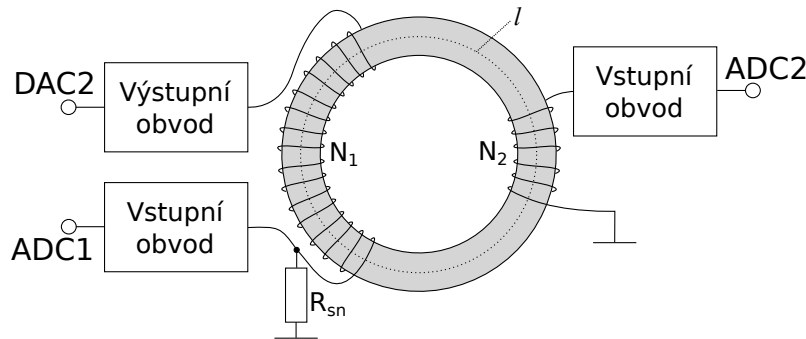
Obrázek 4.6

Zesílení ochranného obvodu je kvůli pevně danému výstupnímu děliči rovno $\frac{1}{2}$.

V této podkapitole nejsou uvedeny žádné konkrétní vztahy. Důvodem je to, že návrh této části závisí na parametrech jednotlivých diod, které má konstruktér zrovna k dispozici. Těmito parametry jsou zejména jmenovitá hodnota propustného, resp. Zenerova napětí, a příslušný diferenciální odpor. Samozřejmostí je pak dodržení mezních parametrů diod.

4.3 Realizace použítá při měření feromagnetik

Vstupní a výstupní obvody popsané v této kapitole byly spolu se STM32F3-Discovery sestaveny na nepájivém vodivém poli a využity pro snímání hysterézní smyčky předloženého feromagnetika. Popsané části byly pospojovány podle schématu na obrázku 4.7. Kanál číslo 2 D/A převodníku je využit pro generování testovacího signálu, A/D převodník číslo 1 vzorkuje napětí na snímacím odporu R_{sn} , které odpovídá magnetovacím proudům, druhý A/D převodník snímá napětí na sekundárním vinutí. Všechny operační zesilovače v tomto zapojení byly napájeny zdrojem symetrického napájení ± 12 V. Napětí 3 V ve schématech bylo získáno z discovery kitu, který obsahuje stabilizátor s výstupním napětím 3 V.



Obrázek 4.7: Schéma zapojení pro měření hysterézní smyčky feromagnetika, ADC1, ADC2 a DAC1 značí příslušné piny STM32F3-Discovery

Výstupní obvod je opatřen na vstupu sledovačem napětí s operačním zesilovačem UA741, jehož výstup je přiveden do dolnoprostupného RC filtru druhého řádu se zlomovou frekvencí

$$f_c = \frac{1}{2\pi R^2 C^2} = 15915 \text{ Hz} . \quad (4.14)$$

Zlomová frekvence a řád filtru jsou plně dostačující pro odstranění nespojitostí v generovaném signálu, který má obnovovací frekvenci více než 100 kHz.

Rozdílový zesilovač určený k odstranění stejnosměrné složky ze signálu byl spočítán přesně podle teorie popsané v 4.1.3 se základním odporem 100 kΩ.

Jako koncové tranzistory byly použity tranzistory BD909 a BD911. Výrobce u těchto tranzistorů uvádí při $I_c = 0,5 \text{ A}$ minimální proudový zisk 40, což při minimálním omezení výstupního proudu UA741 10 mA znamená maximální výstupní proud minimálně 0,4 A, tedy maximální proud, který může dodat laboratorní zdroj, který byl k tomuto měření použit. Maximální výkonová ztráta na tranzistorech tak bude maximálně

$$P_{d,\max} = 12 \cdot 0,4 \text{ W} = 4,8 \text{ W} . \quad (4.15)$$

Je nutné vyšetřit, zda je tranzistory nutné opatřit chladiči. V katalogovém listu BD909/911 (TO220) se neuvádí tepelný odpor mezi přechodem a prostředím, ten je však uveden u jiného tranzistoru s velice podobným pouzdrem TO220AB – IRFZ34N, a to $62 \text{ }^\circ\text{C/W}$. Jelikož maximální povolená teplota přechodu tranzistoru je 150°C , což odpovídá při s rezervou stanovené teplotě v místnosti $30 \text{ }^\circ\text{C}$ oteplení $120 \text{ }^\circ\text{C}$, je maximální povolený ztrátový výkon tranzistorů méně než 2 W, chladič je tedy nutný.

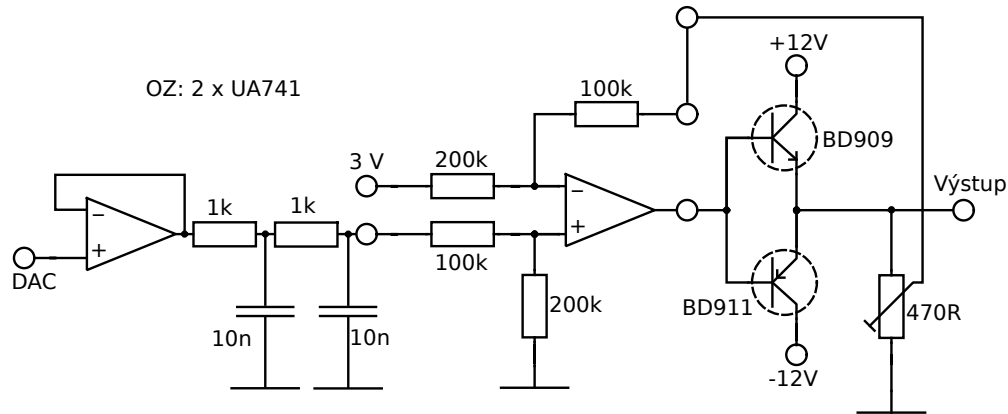
Celkový maximální tepelný odpor se vypočte

$$R_{\theta JC} + R_{\theta CS} + R_{\theta SA,\max} = \frac{\Delta\theta_{\max}}{P_{d,\max}} , \quad (4.16)$$

kde $R_{\theta JC}$ je tepelný odpor mezi přechodem a pouzdrem, který je pro daný tranzistor konstantní (pro BD909/911 $R_{\theta JC} = 1,4 \text{ }^\circ\text{C/W}$), $R_{\theta CS}$ je tepelný odpor mezi pouzdrem a chladičem, který datasheet k IRFZ34N udává typicky $0,5 \text{ }^\circ\text{C/W}$, a $\Delta\theta_{\max}$ je maximální rozdíl teploty přechodu a okolí. Maximální tepelný odpor mezi chladičem a prostředím – $R_{\theta SA,\max}$ – je parametr, podle kterého se vybírá chladič, a ten se vypočte vztahem

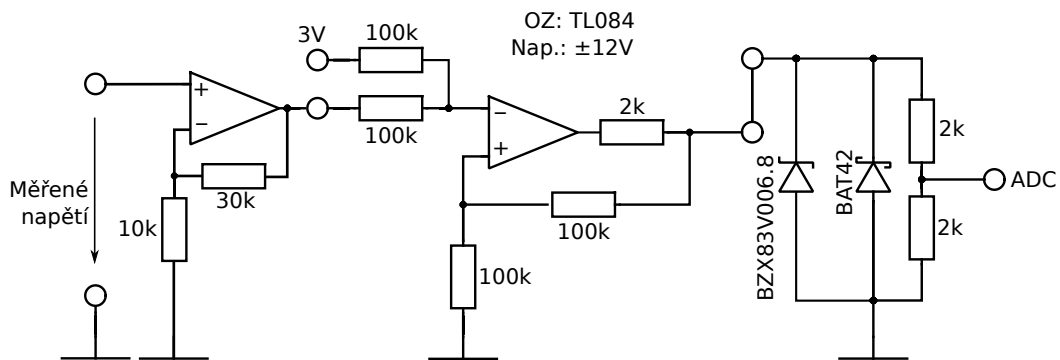
$$R_{\theta SA, \max} = \frac{\Delta\theta_{\max}}{P_{d, \max}} - R_{\theta CS} - R_{\theta JC} = \frac{120}{4,8} - 0,5 - 1,4 \text{ } ^\circ\text{C/W} = 23,1 \text{ } ^\circ\text{C/W}. \quad (4.17)$$

Chladiče, které byly na tranzistory našroubovány mají udávaný tepelný odpor $8,2 \text{ } ^\circ\text{C/W}$, jsou tedy dostačující. Použitím teplovodivé pasty se zajistilo, že bylo dosaženo minimálního možného tepelného odporu mezi pouzdrem a prostředím.



Obrázek 4.8: Zapojení výstupního obvodu pro úpravu signálu použité při měření feromagnetik

Výsledná realizace vstupního obvodu má svoji podobu dánu z velké části výběrem rezistorů, které byly k dispozici v laboratoři, a samozřejmě velikostí napětí snímaných při měření feromagnetika, které bylo k tomuto účelu poskytnuto. Výsledné zapojení vstupní části, se kterým bylo pracováno, je na obrázku 4.9. Na jeho vstupu se nachází neinvertující zesilovač se zesílením 4. Zesílení navázaného posouvacího obvodu lze vypočítat podle vztahu 4.13 a v tomto případě je rovno jedné. Ochranný obvod má pevně dané zesílení $\frac{1}{2}$. Vynásobením všech zesílení se stanoví zesílení celého řetězce $A = 2$. Ochranný obvod byl navrhován podle pravidel uvedených v předchozí podkapitole.



Obrázek 4.9: Zapojení vstupního obvodu pro úpravu signálu použité při měření feromagnetik

Jako Zenerova dioda byla zvolena BZX83V006.8, tedy dioda se Zenerovým napětím 6,8 V. V jejím katalogovém listu lze nalézt její VA charakteristiku v propustém směru, která vykazuje velice malý diferenciální odpor, díky kterému napětí v propustném směru nepřesáhne minimálně do 50 mA hodnotu 7 V, zároveň její prahové napětí má velikost okolo 6,5 V, je tedy zajištěno, že se neuplatní při napětích v požadovaném rozsahu do 6 V a zabrání nárůstu nad 7 V.

V katalogovém listu Schottkyho diody BAT42 není VA charakteristika uvedena, ale lze z něj vyčíst napětí v propustném směru při 10 mA, jež je 400 mV. Vydělením této hodnoty dvěma vyjde bezpečně záporné napětí na vstupu A/D převodníku (max. -300 mV).

Návrhem ochranného rezistoru se musí zajistit, že při kladném nebo záporném saturačním napětí na výstupu OZ posouvacího obvodu nedojde k nárůstu proudu žádnou z diod nad mez, která by způsobila nebezpečný úbytek napětí na těchto diodách. Z toho vyplývá podmínka pro Zenerovu diodu

$$\frac{12 \text{ V} - 6.8 \text{ V}}{R_O} < 50 \text{ mA} \quad (4.18)$$

a pro Schottkyho diodu

$$\frac{12 \text{ V} - 0.4 \text{ V}}{R_O} < 10 \text{ mA} . \quad (4.19)$$

Je jasné, že kritérium dané Schottkyho diodou je silnější, stačí tedy jeho splnění, tedy

$$\begin{aligned} R_O &> \frac{12 \text{ V} - 0.4 \text{ V}}{10 \text{ mA}} \\ R_O &> 1160 \Omega . \end{aligned} \quad (4.20)$$

Zároveň je nutné, aby ochranný odpor nezpůsobil, že operační zesilovač svým saturačním napětím nebude moci na výstupním děliči vybudit napětí 6 V, čímž by se omezil rozsah měření. Bude-li se saturační napětí OZ uvažovat s rezervou 11 V a tato rezerva se uplatní k zanedbání proudu tekoucího odpory ve zpětné vazbě, potom pro maximální hodnotu ochranného odporu platí

$$R_O < 3300 \Omega . \quad (4.21)$$

Oběma podmínkám vyhovuje ochranný odpor velikosti 2 k Ω .

5 USB komunikační protokol pro komunikaci s jednotkou

V této kapitole je popsán způsob komunikace mezi hostitelským počítačem a měřicí jednotkou, formáty zpráv a časové posloupnosti přenosů.

Každá transakce bez výjimky začíná příkazem ze strany hostitelského PC, který je vždy uvozen bytovým identifikátorem určujícím obsah zprávy. V kódu firmwaru a ovladače jednotky jsou tyto číselné identifikátory pro větší přehlednost nahrazeny makry definovanými ve společném header souboru.

Jednotka využívá tyto endpointy:

- EP1 OUT – Interrupt endpoint 1 OUT – řídicí zprávy
- EP2 OUT – Bulk endpoint 2 OUT – přenos dat pro DDS generátor
- EP2 IN – Bulk endpoint 2 IN – přenos navzorkovaných dat

Všechny vícebytové hodnoty se přenášejí ve tvaru little-endian (nejnižší byte se přenáší první, nejvyšší poslední).

Vyžádání a přenos vzorkovaných dat

Tato sekvence začíná bytovou zprávou obsahující pouze identifikátor 0x01 odeslanou na interrupt endpoint 1 jednotky.

Následuje sekvence vzorkování s délkou trvání danou nejdelším součinem délky záznamu a vzorkovací periody ze všech A/D převodníků. Po ní následuje přenos na bulk endpointu 2 ve směru IN, kdy hostitelský počítač vyžádá množství dat v bytech rovnající se součtu délek bufferů všech A/D převodníků. Převodníky jsou 12-tibitové, každý vzorek je tedy reprezentován dvěma byty.

Nastavení velikostí bufferů všech A/D i D/A převodníků

Velikosti bufferů (typu unsigned short, tzn. 16-tibitové celé číslo) se nastavují jak pro A/D, tak i pro D/A převodníky, jediným přenosem 11-tibytové zprávy ve tvaru

EP1 OUT:

0x02	ADC1_LO	ADC1_HI	ADC2_LO	ADC2_HI	ADC3_LO	...
------	---------	---------	---------	---------	---------	-----

... ADC3_HI	DAC1_LO	DAC1_HI	DAC2_LO	DAC2_HI
-------------	---------	---------	---------	---------

,

kde ADCx_LO, resp. DACx_LO značí spodní byte délky bufferu A/D převodníku, resp. D/A převodníku. HI tedy pochopitelně značí horní byte.

Odeslání dat pro generátor signálu

Nejdříve se odešle řídicí zpráva informující jednotku, že bude následovat bulk OUT přenos pro nastavení hodnot pro generaci signálu. Zpráva obsahuje informaci, kterému kanálu D/A převodníku jsou data určena ($\text{DAC_CH} = \{1, 2, 12 - \text{data společná pro oba kanály}\}$), a počet bitů pro adresaci dat ($\text{BITS} = \log_2 N$, N je počet vzorků pro generovaný signál). Tento přenos lze shrnout schématem

EP1 OUT:

0x04	DAC_CH	BITS
------	--------	------

.

Poté následuje přenos dat na bulk endpoint 2 OUT, který obsahuje $2N$ bytů dat (vzorky jsou 12-tibitové, každý obsahuje 2 byty).

EP2 OUT:

[$2N$ bytů dat]

Nastavení frekvence generovaného signálu

Frekvence generovaného signálu lze nastavit s přesností na jednotky Hz zprávou

EP1 OUT:

0x05	DAC_CH	FREQ(0-7)	FREQ(8-15)	FREQ(16-23)	FREQ(24-31)
------	--------	-----------	------------	-------------	-------------

,

kde DAC_CH je číslo kanálu generátoru (1 nebo 2) a FREQ jsou jednotlivé byty 32-bitové celočíselné frekvence.

Nastavení obnovovací frekvence DDS

Obnovovací frekvence DDS se nastaví odesláním jednoduché zprávy na endpoint 1, která obsahuje pouze identifikátor a 32-bitovou hodnotu obnovovací frekvence (UPDT_RT).

EP1 OUT:

0x06	UPDT_RT(0-7)	UPDT_RT(8-15)	UPDT_RT(16-23)	UPDT_RT(24-31)
------	--------------	---------------	----------------	----------------

Nastavení předděliček a délky odběru vzorku A/D převodníků

K nastavení těchto hodnot slouží 6-bytová řídicí zpráva nesoucí indexy předděliček (ADC12_PSC a ADC3_PSC) a indexy délek doby odběru vzorku (ADC1_ST , ADC2_ST a ADC3_ST). Konkrétní hodnoty indexům přiřazují tabulky 7.1 a 7.2. Celá zpráva má podobu

EP1 OUT:

0x07	ADC12_PSC	ADC1_ST	ADC2_ST	ADC3_PSC	ADC3_ST
------	-----------	---------	---------	----------	---------

.

Nastavení režimu funkce D/A převodníku

Vypnutí (MODE = 0), režim DDS (MODE = 1) nebo stejnosměrnou funkci (MODE = 2) lze pro daný kanál nastavit zprávou ve tvaru

EP1 OUT:

0x09	DAC_CH	MODE
------	--------	------

 ,

kde DAC_CH je číslo kanálu, kterému se nastavuje režim (nabývá hodnot 1 nebo 2).

Nastavení stejnosměrné hodnoty

Pomocí zprávy vyobrazené níže lze nastavit pro daný kanál A/D převodníku (DAC_CH) hodnotu stejnosměrného výstupu (DC_VALUE).

EP1 OUT:

0x0A	DAC_CH	DC_VALUE_LO	DC_VALUE_HI
------	--------	-------------	-------------

Nastavení synchronizace počátku vzorkovací sekvence s DDS generátorem

Začátek vzorkování lze synchronizovat s prvním vzorkem následující periody DDS generátoru (D/A převodníku, kanálu DAC_CH = {1, 2, ostatní – synchronizace neaktivní}) dvoubytovou zprávou ve tvaru

EP1 OUT:

0x0D	DAC_CH
------	--------

 .

6 Firmware pro STM32F303

6.1 Popis používaných hardwarových prostředků

Díky strategii řady výrobců integrovaných obvodů jsou pro významné řady jejich produktů dostupné vývojové kity, které výrazně usnadňují práci při vývoji firmwaru pro příslušné čipy. Tyto kity jsou natolik cenově přijatelné a mají natolik kvalitní podporu, že jsou vhodné i pro amatéry. Mezi tyto vývojové prostředky patří mezi jinými i výrobek firmy ST Microelectronics STM32F3-Discovery, který, stejně jako všechny ostatní současné kity této společnosti, obsahuje na jedné desce plošných spojů jak procesor dané řady (v zásadě model opatřený maximálním možným počtem periférií), tak programovací a ladicí procesor STM32F103, jehož firmware je pomocí na desce dostupných rozhraní nemodifikovatelný. Tento procesor realizuje rozhraní mezi USB a JTAG, zvané ST-LINK V2, které lze pořídit i samostatně ve formě malého zapouzdřeného zařízení.

Vstupy a výstupy vývojového kitu

Na kitu jsou dva konektory USB Mini-B, jeden pro připojení k ST-LINK V2, tedy fyzicky spojený s ladicím/programovacím procesorem STM32F103, druhý pro přímé připojení k programovatelnému procesoru STM32F303. Díky použití stejného konektoru pro obě rozhraní lze využívat kabel jednoho typu jak k programování, tak k uživatelským aplikacím. V této práci byly používány většinou kabely dva, jelikož bylo nutné provádět ladění za probíhající komunikace pomocí uživatelského USB. Zapojení obou konektorů umožňuje napájení z kteréhokoliv z nich, aby bylo možné využití jednoho kabelu pro všechny účely. Téměř všechny piny procesoru jsou vyvedeny na dva dvouřadé konektory (rozteč 100 mil/2,54 mm) umožňující zasunutí desky do nepájivého kontaktního pole, což nelze doporučit vzhledem k tomu, že spolu sousedí kontakty, které není vhodné zkratovat. Headery lze však snadno vyvést pomocí postupně ustupujících plochých kabelů používaných zejména pro připojování diskových a optických jednotek v osobních počítačích.

6.2 Popis programového vybavení použitého pro programování firmwaru

Tato kapitola popisuje veškerý software použitý pro programování a ladění firmwaru pro STM32F303. Nejvýstižněji použitou konfiguraci vystihuje obrázek 6.1.

Veškerý vývoj probíhal na osobním notebooku s operačním systémem linux, konkrétně distribucí Ubuntu 12.04. Tento systém byl použit nejen na základě osobních preferencí autora, obecně totiž linux umožňuje lepší kontrolu nad hardwarem a jeho efektivní monitoring. Toho bylo využito při monitorování komunikace přes uživatelské USB rozhraní (mezi vnitřním USB řadičem mikrokontroláru STM32F303 a PC aplikací).

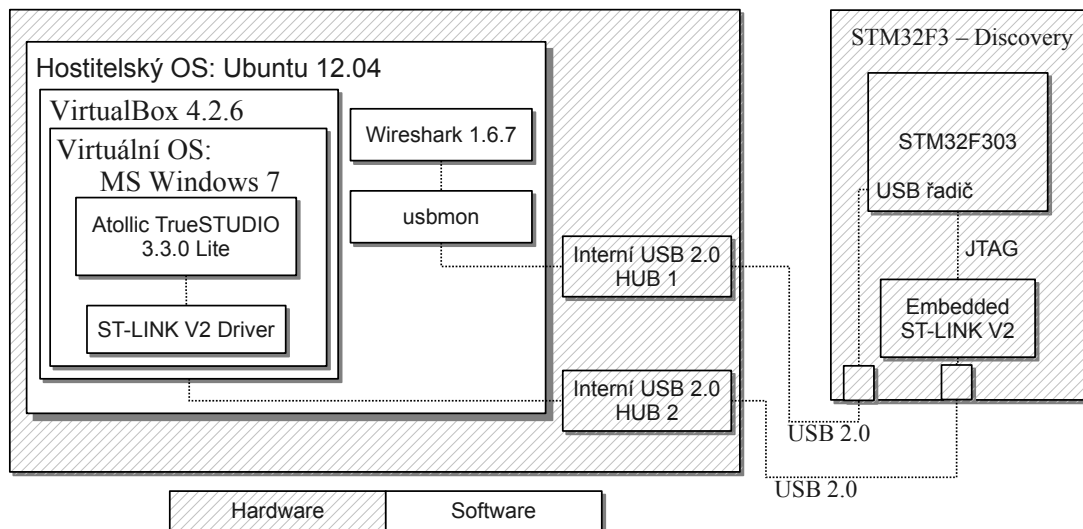
Monitoring komunikace na USB

Monitoring byl možný díky kernel modulu *usbmon*, který je ve výchozím stavu instalace Ubuntu 12.04 nainstalován a automaticky spouštěn při startu systému. Data poskytovaná tímto modulem lze číst přímo z terminálu pomocí příkazu `sudo cat /sys/kernel/debug/usb/usbmon/X`, kde X zastupuje soubor odpovídající monitorované sběrnici (použitý notebook má celkem 4 sběrnice USB, při ladění byla používána vždy sběrnice číslo 1, konkrétně soubor `...usbmon/1t`). Tento přístup

má tu výhodu, že velice přímočaře zobrazuje surová data probíhající na monitorované sběrnici. Dále je možné využít front-end software *Wireshark*, který slouží k monitoringu nejen USB sběrnice, ale i řady dalších rozhraní. V případě použití k monitoringu USB využívá právě modul *usbmon*, ale narozdíl od přímého vyčítání provádí uživatelsky přívětivou interpretaci veškerých dat, je vhodný v případě, kdy se uživatel dostane do situace, ve které nedokáže správně interpretovat surový výpis poskytovaný *usbmonem*. Nevýhodou však je, že je nutné jednotlivé transakce rozbalovat pomocí myši. V průběhu této práce byly tedy využívány oba způsoby.

Vývojové prostředí (IDE) využité k programování a ladění STM32F303

Pro vývoj firmwaru bylo využito freeware licence programu Atollic TrueSTUDIO for ARM Lite (verze 3.3.0). Tento software byl provozován ve virtuálním stroji s Windows 7 zprostředkovaném pomocí virtualizačního nástroje Oracle VM VirtualBox ve verzi 4.2.6. Byl proveden jeden neúspěšný pokus o instalaci IDE přímo v Ubuntu pomocí nástroje Wine. Vzhledem k povaze softwaru a zejména ovladače ST-LINK V2 se s neúspěchem při instalaci počítalo. Podpora USB musí být do VirtualBoxu přidána pomocí rozšíření (VirtualBox Extension Pack), díky němuž lze libovolné USB zařízení doslova předat hostovanému systému, čímž k němu přirozeně hostitelský OS ztrácí přístup. VirtualBox umožňuje zvolit pro vybrané USB zařízení automatické přepojování hostovanému systému. Ve verzi používané při této práci způsobovalo nastavení automatického připojování selhání při spouštění systému, konkrétně byla hlášena chyba při čtení .xml souboru s uloženou konfigurací virtuálního stroje, nebylo zjištěno, zda tato chyba postihuje i jiná zařízení nebo je ošetřena v jiné verzi VirtualBoxu, použitá verze byla v době psaní této práce aktuální.



Obrázek 6.1: Schéma propojení hardwaru a použitého softwaru při programování

6.3 Popis vlastní implementace

Tato kapitola popisuje všechny části firmwaru, které byly implementovány v rámci této práce. Při programování byly využity standardní ovladače periférií STM32 a knihovna pro programování USB poskytovaná pro konkrétní řadu procesorů firmou ST Microelectronics. Dokumentace k těmto knihovnám lze dohledat samostatně a tato práce se jejich implementací nezabývá.

V celé kapitole jsou používána makra definovaná ve zmíněných ovladačích, která jsou psána monospace fontem. Jejich význam je vcelku dobře čitelný, jedná se o adresy periférií či masky pro nastavování registrů. Pro plné pochopení je vhodné alespoň letmé prohlédnutí zdrojového kódu ovladačů spolu s referenčním manuálem k procesoru^[3], kde jsou všechny registry a adresy podrobně popsány a zmapovány.

6.3.1 Funkce `main` – inicializace

Funkce `main` v tomto firmwaru plní pouze inicializační funkci, jelikož je běh programu plně řízen přerušeními. Základní inicializace v této funkci realizovaná se sestává z těchto úkonů

- Zavolání funkce `Set_System` použité z demonstračního projektu k USB knihovně
- Konfigurace přerušení (viz odstavec níže)
- Nastavení hodinových signálů v procesoru (viz 6.3.4)
- Zavolání funkce `SetupEventGenerator` (účel popsán v 6.3.10)
- Zavolání funkce z ovladače `USB-FS-Device_Driver` `USB_Init` pro inicializaci USB

6.3.2 Nastavení přerušení

Jak bylo řečeno v předchozím odstavci, tento firmware je řízen výhradně přerušeními. Ta jsou konfigurována ve vlastní proceduře `SetupInterrupts`. Nejprve se zvolí prioritní skupina 2, která specifikuje, že se bude u přerušení zadávat 2-bitová preempční priorita a 2-bitová subpriorita. Nastavení přerušení a jejich priorit s krátkým popisem jejich významu je uvedeno v tabulce 6.1.

Přerušení musí být nakonfigurována tak, aby bylo dosaženo co nejoptimálnějšího běhu programu. Na pořadí priorit obecně nezáleží u přerušení, která nenastávají najednou, nepanuje tedy obava, že bude obsluha jednoho přerušení obsluhou jiného. Chování programu založeného na přerušeních je někdy obtížně predikovatelné, to platí zejména v případě integrovaného USB řadiče. Aby komunikace byla možná, musí mít přerušení probuzení USB bezpodmínečně nejvyšší prioritu, hned po ní ostatní interrupty USB. Dále už bylo možné priority nastavit s podstatně větší svobodou. Priorita timeru `TIM6` byla nastavena jako nejnižší, jelikož se jedná o obsluhu DDS generátoru, který může mít nastavenou takovou obnovovací frekvenci, že zatíží procesor na 100%, došlo by tak k uvážnutí programu. Ze zbývajících dvou skupin přerušení byla přiřazena vyšší priorita obsluze přerušení A/D převodníků, která nastává pouze nepravidelně při vysokých vzorkovacích frekvencích, kdy dojde k tomu, že je navzorkována další hodnota aniž byla předchozí zpracována. Toto přerušení musí být obslouženo co nejrychleji, aby nedošlo ke ztrátě dat a nehrozí u něj, že by blokovalo obsluhy jiných. Priorita o jednu nižší, tedy 2, byla nastavena pro přerušení vyvolané dokončeným přenosem DMA.

Tabulka 6.1: Přerušeni nakonfigurovaná pro řízení běhu firmwaru, nižší číslo značí vyšší prioritou

Označení v rámci standardního ovladače periférií	Preempční priorita	Subpriorita	Popis
USB_LP_CAN1_RX0_IRQn	1	0	Přerušeni vyvolávané řadičem USB
USBWakeUp_IRQn	0	0	Přerušeni probuzení USB
DMA1_Channel11_IRQn	2	0	Přerušeni vyvolané při dokončení vzorkovací sekvence ADC1
DMA2_Channel11_IRQn	2	1	Přerušeni vyvolané při dokončení vzorkovací sekvence ADC2
DMA2_Channel15_IRQn	2	2	Přerušeni vyvolané při dokončení vzorkovací sekvence ADC3
ADC1_2_IRQn	1	1	Přerušeni při nečekané události převodníku ADC1 nebo ADC2
ADC3_IRQn	1	2	Přerušeni při nečekané události převodníku ADC3
TIM6_DAC_IRQn	3	0	Přerušeni timeru TIM6 realizujícího takt DDS generátoru

6.3.3 Obsluhy přerušeni

Obsluhy všech přerušeni jsou implementovány v souboru *stm32_it.c*. Mezi ně patří obsluhy přerušeni řadičem USB, jejichž implementace je v jednom případě řešena voláním funkce z knihovny pro programování USB řadiče a ve druhém pouhým resetováním flagu externího interruptu na přerušovací lince číslo 18, která je vnitřně spojena s USB řadičem a využita pro probuzení přes USB.

Obsluhy, které byly implementovány v rámci této práce jsou popsány v následujících odstavcích.

Přerušeni vyvolaná A/D převodníky

A/D převodníky jsou nastaveny tak, aby generovaly přerušeni pouze v případě přetečení výstupního bufferu, tzn. že vzorkovaná hodnota nebyla přečtena před příchodem další. Tato skutečnost je ignorována, jelikož neodvratně nastává při vyšších vzorkovacích frekvencích. Jediným úkolem přerušeni je vynulovat příslušný flag (OVR).

Přerušeni vyvolaná řadiči DMA

V případě přerušeni řadičem DMA dostává firmware signál, že byla dokončena nakonfigurovaná vzorkovací sekvence na jednom z A/D převodníků. Obsluha tohoto přerušeni zastavuje převodník, jehož

požadavky na přenosy daný DMA řadič a jeho kanál vykonával, dále ve svém registru přerušení nuluje příznak DMAx_IT_TCy (x – číslo řadiče, y – číslo kanálu), nastavuje jeden konkrétní bit v jednoobytovém registru realizovaném v operační paměti a volá rutinu `EndOfMeasurementRoutine`.

Úkolem jmenované rutiny je porovnat zmíněný registr modifikovaný obsluhami přerušení DMA s registrem stejné struktury, do kterého se před spuštěním vzorkovací sekvence zapíše logické 1 na pozici odpovídající A/D převodníkům zúčastňujícím se aktuálního vzorkování. V případě, že se oba registry rovnají, znamená to, že všechny A/D převodníky, které byly pro tuto sekvenci spuštěny, dokončily sběr dat a všechna data se mohou odeslat po sběrnici USB hostitelskému počítači (procedura `SendData`). Rovněž se zakáže časovač TIM1, aby mohly být A/D převodníky znovu spuštěny (viz 6.3.10).

Přerušení časovače TIM6

Časovač TIM6 generuje přerušení pravidelně v intervalech daných obnovovací frekvencí DDS generátoru. Jako první se v obsluze jako obvykle nuluje příslušný příznakový bit (zde update interrupt flag UIF) a následuje spuštění D/A převodníků, při kterém dojde u obou kanálů k přesunu dat zapsaných při předchozím přerušení do DHR (data holding register) do registru DOR (data output register), čímž se data projeví na výstupu převodníku.

Důvodem, proč se data do DHR ukládají v jednom cyklu a do výstupního registru se přesouvají v dalším, je to, aby se převodníky spouštěly co nejdříve po vyvolání přerušení a minimalizovala se pravděpodobnost, že bude před spuštěním tato obsluha zdržena přerušením s vyšší prioritou, čímž by mohlo docházet k výraznému jitteru (odchylkám okamžiku spouštění od požadovaného).

Po spuštění D/A převodníku může být spuštěna vzorkovací sekvence povolením časovače TIM1 (viz 6.3.10). K tomu dojde tehdy, pokud je vzorkování synchronizováno s nulovou fází některého z kanálů D/A převodníku, je vydán požadavek na spuštění vzorkovací sekvence a adresa předchozího vzorku je větší než adresa aktuálního vzorku (tedy došlo k přetečení fázového registru a na výstupu je první vzorek nové periody). Rozhodnutí, zda bude při příštím přerušení spuštěna vzorkovací sekvence je učiněno na konci obsluhy přerušení, před kterým je ještě realizován algoritmus DDS, který je pro větší tematickou konzistenci dokumentu popsán v 6.3.11.

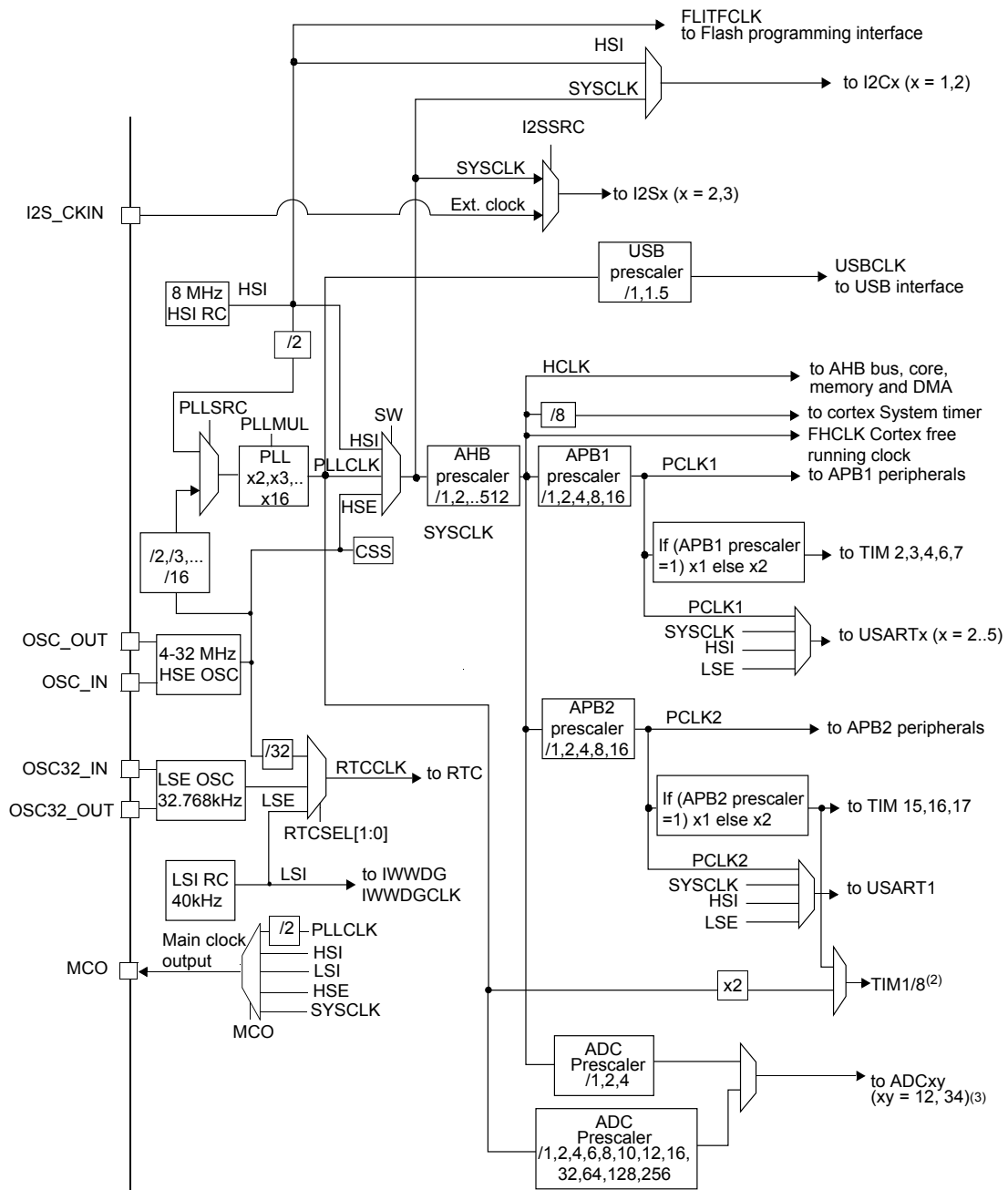
6.3.4 Konfigurace hodin

Základní pomůckou při nastavování hodin procesorů STM32 je tzv. clock tree (pro STM32F303 viz obrázek 6.2), tedy schéma všech hodinových signálů v procesoru, zahrnující oscilátory, fázový závěs (PLL), sloužící k násobení vstupní frekvence, všechny předděličky a prepínače pro volby frekvencí hodinových signálů pro jednotlivé periferie.

Řadič USB vyžaduje ke svojí funkci hodiny o frekvenci 48 MHz, je tedy nutné nastavit předděličku USB (USB prescaler) na 1,5.

Pro dosažení maximálního výkonu zpracovávání instrukcí byl nastaven AHB prescaler na 1.

Předděličky A/D převodníků jsou konfigurovatelné příkazy přicházejícími z PC a tato konfigurace musí předcházet prvnímu měření, při počáteční konfiguraci hodin se tedy neiniciluzují.



Obrázek 6.2: Clock tree procesoru STM32F303 (převzato z [3])

6.3.5 Popis rozhraní pro komunikaci s PC (USB)

K programování USB řadiče byla využita knihovna od ST Microelectronics *STM32_USB-FS-Device_Driver*. Rozhraní bylo vytvořeno postupným přetvořením demonstračního projektu *Custom_HID* dodávaného s ovladačem. Důvodem byl vývoj z funkčního stavu, který zajišťoval, že nebude chybět žádná pro funkci podstatná část. Výsledné chování se navenek původnímu demonstračnímu projektu v žádném ohledu nepodobá.

Descriptorů zařízení (*usb_desc.c*)

Tento soubor obsahuje descriptorů, kterými se zařízení identifikuje operačnímu systému hostitelského PC. Device descriptor byl definován podle tabulky 6.2

Tabulka 6.2: Device descriptor pro enumeraci jednotky

0x12	Délka descriptoru (18)
0x01	Typ descriptoru (1 – device descriptor)
0x00 0x02	Verze USB (2.00)
0x00	Třída zařízení (0 – definováno interface descriptorem)
0x00	Podtřída (pro třídu 0 vždy 0)
0x00	Protokol (pro třídu 0 vždy 0)
0x40	Maximální velikost paketu (64 bytů)
0x83 0x04	ID výrobce (0x0483 – STMicroelectronics)
0x11 0x11	ID zařízení (0x1111 – nepoužité ID)
0x00 0x01	Verze zařízení (1.00)
0x00	Index string descriptoru pro popis výrobce TODO prepsat firmwar
0x01	Index string descriptoru pro popis výrobce TODO prepsat firmwar
0x00	Index string descriptoru pro popis seriového čísla TODO prepsat firmwar
0x01	Počet konfigurací (1)

Dalšími descriptorů jsou configuration descriptor, interface descriptor a čtyři endpoint descriptorů. Všechny jsou uvedeny v jedné tabulce, jelikož i v programu jsou uloženy v jednom poli kvůli tomu, že se při enumeraci zasílají jako jeden blok dat.

Tabulka 6.3: Configuration, interface a endpoint descriptorů pro enumeraci jednotky

0x09	Délka descriptoru
0x02	Typ descriptoru (2 – configuration descriptor)
0x2E 0x00	Součet délek configuration, interface a všech endpoint descriptorů (46)
0x01	Počet rozhraní (interface deskriptorů)
0x01	Číslo konfigurace, kterou popisuje tento descriptor
0x00	Číslo string descriptoru pro popis této konfigurace
0x80	Tento byte značí, že je zařízení napájeno po sběrnici
0x32	Maximální odběr proudu (50×2 mA)
0x09	Délka descriptoru
0x04	Typ descriptoru (4 – interface descriptor)
0x00	Číslo tohoto rozhraní
0x00	Hodnota pro výběr alternativního nastavení
0x04	Počet endpointů (endpoint descriptorů)
0xFF	Třída rozhraní (0xFF – rozhraní definuje výrobce)
0x00	Podtřída rozhraní
0x00	Protokol rozhraní
0x00	Index string descriptoru popisujícího toto rozhraní
0x07	Délka descriptoru
0x05	Typ descriptoru (5 – endpoint descriptor)
0x01	Adresa endpointu (EP1, směr OUT)
0x03	Typ endpointu (0x03 – interrupt endpoint)
0x0b 0x00	Maximální velikost paketu (11 bytů)
0x20	Interval pro polling endpointu (32 ms)
0x07	Délka descriptoru
0x05	Typ descriptoru (5 – endpoint descriptor)
0x81	Adresa endpointu (EP1, směr IN)
0x03	Typ endpointu (0x03 – interrupt endpoint)
0x04 0x00	Maximální velikost paketu (4 byty)
0x20	Interval pro polling endpointu (32 ms)
0x07	Délka descriptoru
0x05	Typ descriptoru (5 – endpoint descriptor)
0x82	Adresa endpointu (EP2, směr IN)
0x02	Typ endpointu (0x02 – bulk endpoint)
0x40 0x00	Maximální velikost paketu (64 bytů)
0x00	Interval pro polling (u bulk EP se ignoruje)
0x07	Délka descriptoru
0x05	Typ descriptoru (5 – endpoint descriptor)
0x02	Adresa endpointu (EP2, směr OUT)
0x02	Typ endpointu (0x02 – bulk endpoint)
0x40 0x00	Maximální velikost paketu (64 bytů)
0x00	Interval pro polling (u bulk EP se ignoruje)

Konfigurace endpointů a činnosti při enumeraci (*usb_prop.c*)

V tomto souboru jsou implementovány callback funkce volané v jednotlivých fázích enumerace. Důležité modifikace v něm provedené zahrnují

- Nastavení počtu použitých endpointů ve struktuře `Device_Table` na 4.
- Nastavení všech použitých endpointů v reset callbacku (funkce `CustomHID_Reset`), nastavení jsou realizována tak, aby respektovala popis endpointů podle tabulky 6.3. Tento bod byl dokončen s výraznou pomocí demonstračních projektů a kapitoly 29.4.2 – *System and power-on reset* v referenčním manuálu k STM32F303^[3], kde je postup inicializace endpointů detailně popsán.
- Modifikace callback funkce nastavení konfigurace přidáním volání funkcí pro inicializaci všech A/D převodníků, k nim příslušejícího DMA, D/A převodníku a timeru TIM6, který časuje DDS generaci signálů.

Obsluha požadavků na přenosy (*usb_endp.c*)

Zde jsou callbacky pro jednotlivé přenosy na nakonfigurovaných endpointech. Názvy callback funkcí jednoznačně určují, o jaký endpoint a směr přenosu se jedná.

- `EP1_OUT_Callback`: Na tomto endpointu jednotka přijímá příkazy z hostitelského PC. Struktura všech povolených příkazů je uvedena v popisu komunikačního protokolu (kapitola 5). Ke zpracování příkazů se volá funkce `ProcessCommand`, kde se program rozvětví podle identifikátoru zprávy, na jehož základě dojde k nastavení příslušných proměnných atd. Zpracování je poměrně přímočaré a snadno pochopitelné ze zdrojového kódu.
- `EP2_OUT_Callback`: Hlavní funkcí tohoto callbacku je přijmout data průběhu DDS generátoru a uložit je do příslušného bufferu. Přenosu na EP2 by měl předcházet příkaz na EP1 OUT (viz 5) udávající, pro který kanál D/A převodníku jsou data určena, případně packet adresovaný EP0 popsán v kapitole 3.3, přepínající `EP2_OUT_Callback` na příkazovou rouru pro řízení pomocí `LabVIEW`.
- `EP2_IN_Callback`: EP2 ve směru IN slouží výhradně k přenosu navzorkovaných dat. Tento callback se vykonává opakovaně dokud nedojde k přenesení všech navzorkovaných dat. Prvních 64 bytů dat se do výstupního bufferu endpointu zapíše po skončení vzorkovací sekvence, všechna zbývající data jsou do výstupního bufferu zapisována v tomto callbacku.

Při každém zápisu se zjišťuje, zda zbývá přenést více nebo rovno než 64 bytů dat, pokud ano, zapíše se dalších 64 bytů dat do výstupního bufferu a do registru délky paketu se zapíše 64. Pokud zbývá dat méně, zapíší se do výstupního registru všechna zbývající data a do registru délky paketu zbývající množství dat. Tak se zajistí, že se odešlou všechna data korektně.

6.3.6 Dynamická alokace paměti pro data A/D i D/A převodníků (`SetDataBuffers`)

Pro zvýšení univerzálnosti firmwaru je paměť pro všechny převodníky alokována dynamicky, mohou tedy být na základě potřeb aplikace některé buffery rozšířeny na úkor jiných, například v případě, že je zapotřebí pořídit dlouhý záznam pouze na jednom kanálu bez nutnosti snižovat vzorkovací kmitočty.

Tohoto je dosaženo tak, že je nejprve staticky alokována paměť pro všechna data na velikost co nejvíce se blížící volné paměti mikrokontroléru. Tato paměť je poté rozdělena na základě požadavku aplikace, která stanoví velikosti všech bufferů (viz komunikační protokol 5). Ukazatel na data převodníku

ADC1 je napevno nastaven na adresu začátku staticky alokovaného bufferu. Část kódu na obrázku 6.3 bez dalšího komentáře objasňuje způsob dynamické alokace. Pro jednoznačnost byly do tohoto výpisu přidány datové typy proměnných.

```

/*      uint8_t*          uint8_t*          uint16_t      */
    ukazatel_na_buffer_ADC2 = ukazatel_na_buffer_ADC1 + velikost_bufferu_ADC1;
    ukazatel_na_buffer_ADC3 = ukazatel_na_buffer_ADC2 + velikost_bufferu_ADC2;
    ukazatel_na_buffer_DAC_kanal1 = ukazatel_na_buffer_ADC3 + velikost_bufferu_ADC3;
    ukazatel_na_buffer_DAC_kanal2 = ukazatel_na_buffer_DAC_kanal1 + velikost_bufferu_DAC1;

```

Obrázek 6.3: Způsob alokace bufferů pro A/D a D/A převodníky

6.3.7 Programování A/D převodníků procesoru STM32F303

Tato část se zabývá inicializací a možnostmi konfigurace příkazy na rozhraní USB. V této části dokumentu se často používají makra definovaná ve standardním ovladači periferií.

Inicializace A/D převodníků (`ADC_init`)

A/D převodníky se inicializují funkcí `ADC_init`, jež se volá pouze jednou při enumeraci a přijímá 4 argumenty:

- Adresu konfiguračních registrů nastavovaného A/D převodníku (`ADC1`, `ADC2` nebo `ADC3`)
- Adresu konfiguračních registrů brány, na které je umístěn vybraný kanál nastavovaného A/D převodníku (`GPIOA`, `GPIOB`, ...)
- Makro pro pin brány, na kterém je umístěn vybraný kanál nastavovaného A/D převodníku (`GPIO_Pin_0` ($1 \ll 0$) až `GPIO_Pin_15` ($1 \ll 15$))
- Číslo kanálu A/D převodníku

Jejím účelem je nastavit kanál brány a konkrétní pin převodníku podle definic v souboru `firmware_defines.h`.

Popis funkce K zajištění podmínek potřebných k nastavování A/D převodníků se jako první příkaz vykoná zastavení případného vzorkujícího převodníku. Následně se pomocí podmínkové struktury vykonají buď příkazy pro nastavení společné předděličky pro `ADC1` a `ADC2` a povolení společných hodin pro tyto dva převodníky nebo obdobné nastavení pro `ADC3`.

V dalším kroku se povolí hodiny pro příslušnou GPIO bránu. Pro zachování univerzálnosti této funkce byla provedena analýza maker `RCC_AHBPeriph_GPIOx`, kde `x` nabývá hodnot `A`, `B` atd. Z této analýzy vyplývá, že hodnotu libovolného makra `RCC_AHBPeriph_GPIOx` lze získat operací `RCC_AHBPeriph_GPIOA << ((GPIOx & 0x0000FF00) >> 10)`.

Poté je nastaven zvolený pin GPIO brány jako analogový I/O bez pull up/down rezistoru, je aktivován napěťový regulátor A/D převodníku a provedena kalibrace převodníku. Další běh programu je blokován dokud není kalibrace dokončena.

Nastavení chování převodníků je shodné pro všechny tři používané a provádí se pomocí k tomu určených funkcí ze standardní knihovny `STM32F37x_StdPeriph_Driver`. Nastavení jsou následující:

- Převodníky jsou nastaveny jako na sobě nezávislé
- Zvolen asynchronní režim (nezávislý na sběrnici AHB) umožňující použití předděličky hodinového signálu
- Zakázán DMA přístup pro režim "multi" (opak zvoleného nezávislého)
- Režim DMA nastaven na jednorázový přenos (je restartován při každém odběru dat)
- Prodleva mezi dvěma vzorkovacími fázemi nastavena na 0
- Nastaven kontinuální režim vzorkování
- Nastaveno rozlišení 12 bitů
- Nastavení spouštění na vzestupnou hranu externího spouštěče (EXT9 – událost TIM1_TRGO)
- Zarovnání dat doprava umožňující jednoduché přetypování přenesených dat PC aplikací
- Nastavení módu při přetečení registru výstupních dat na přepis předchozí hodnoty (v případě, že dojde k digitalizaci další hodnoty aniž by předchozí hodnota byla přečtena DMA)
- Zakázán automatický převod injektovaných kanálů po konverzi regulérních kanálů
- Nastavení počtu regulérních kanálů na 1 (každý ADC převádí pouze 1 kanál)

Po provedení předchozích nastavení je nastaven vybraný kanál ADC: je mu přiděleno první místo v pořadníku převodů jednotlivých kanálů (již bylo zmíněno, že je převáděný kanál rovněž jediný) a je mu nastavena výchozí doba odběru vzorku 19,5 hodinových impulsů.

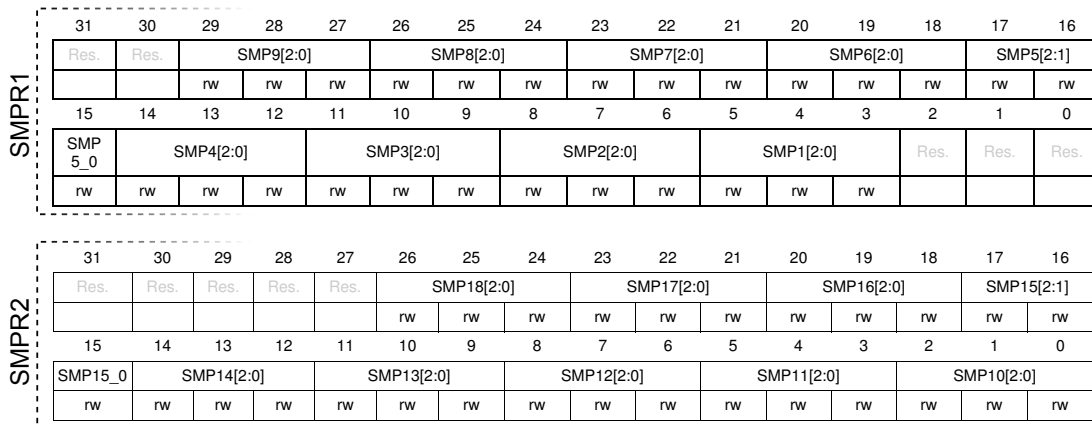
Nakonec je provedeno povolení přerušeni při přetečení registru výstupních dat a povolení převodníku. Funkce před svým ukončením už jen vyčká na nastavení příznaku `ADRDY` (ADC ready) v ISR daného ADC, značícího připravenost převodníku.

Změna doby odběru vzorku a předděličky hodinového signálu (**SetADCTimings**)

Po inicializaci je možné převodníky v omezené míře přenastavit. Konkrétně lze změnit hodnoty předděliček pro ADC1/ADC2 a pro ADC3 a dále doby odběru vzorku pro každý převodník zvlášť. V prvotní verzi firmwaru byla k přenastavování těchto parametrů využívána funkce `ADC_init` popsaná v předchozím odstavci, ale vzhledem k velkému množství prováděných operací byl tento přístup z pochopitelných důvodů nepříliš stabilní a obtížně předvídatelný. Byl proto zvolen minimalistický nízkoúrovňový přístup vyhýbající se funkcím standardní knihovny, které jsou určeny k hromadné konfiguraci celých skupin registrů.

Délka odběru vzorků se nastavuje v registrech `SMPR1` a `SMPR2` pro každý ADC zvlášť. Struktura těchto registrů je zachycena na obrázku 6.4, z něhož je zřejmé, že registr `SMPR1` obsahuje nastavení pro kanály 1–9 a registr `SMPR2` nastavení pro zbývající kanály (10–18). Tři bity definující délku odběru vzorku přesně odpovídají binární formě indexu volby z combo boxu PC aplikace. Jediná operace, kterou je nutné vykonat s daty vrácenými při výběru použitím combo boxu, je jejich posuv tak, aby byla zapsána na správnou pozici pro používaný kanál.

Vlastní nastavení příslušných hodnot nejlépe zachycuje výňatek kódu na obrázku 6.5. Vzhledem k tomu, že se používá pouze jeden kanál, nenastává žádný problém při nulování zbytku registru.



SMPx[2:0]: Channel x sampling time selection

These bits are written by software to select the sampling time individually for each channel. During sample cycles, the channel selection bits must remain unchanged.

- 000: 1.5 ADC clock cycles
- 001: 2.5 ADC clock cycles
- 010: 4.5 ADC clock cycles
- 011: 7.5 ADC clock cycles
- 100: 19.5 ADC clock cycles
- 101: 61.5 ADC clock cycles
- 110: 181.5 ADC clock cycles
- 111: 601.5 ADC clock cycles

Obrázek 6.4: Registry délky odběru vzorků (převzato z [3], graficky upraveno)

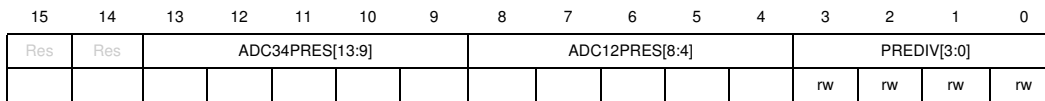
```

if (číslo_kanálu < 10)
    ADCx->SMPR1 = index_doby_odběru_vzorku << 3*číslo_kanálu;
else
    ADCx->SMPR2 = index_doby_odběru_vzorku << 3*(číslo_kanálu-10);

```

Obrázek 6.5: Nastavení délky doby odběru vzorků

Nastavení předděliček probíhá velice podobným způsobem. Pětibitová hodnota, kterou je nutné zapsat na příslušnou pozici registru `RCC_CFGR2` (obrázek 6.6) je složena ze 4 bitů daných indexem zvoleného dělicího poměru na spodních čtyřech bitech a logické 1 na nejvyšším bitu povolující PLL hodiny pro použití k časování ADC. V tomto případě se ale jedná o opětovné nastavování stejného registru, musí se tedy zabránit přepisu ostatních částí registru. Tato část kódu je zachycena na obrázku 6.7.



Obrázek 6.6: Spodní dva byty registru předděliček A/D převodníků a PLL (horní 2 byty rezervované; převzato z [3])

```

if (ADCx == ADC1 || ADCx == ADC2)
{
    RCC->CFGR2 &= 0xFFFFFE0F;          /* Nulování */
    RCC->CFGR2 |= 0x100 + (index_dělicího_poměru << 4); /* Nastavení nové hodnoty */
}
else
{
    RCC->CFGR2 &= 0xFFFFC1FF;          /* Nulování */
    RCC->CFGR2 |= 0x2000 + (index_dělicího_poměru << 9); /* Nastavení nové hodnoty */
}

```

Obrázek 6.7: Nastavení předděliček A/D převodníků

6.3.8 Inicializace DMA pro přenos dat A/D převodníků (**ADC_DMA_init**)

Vzhledem k vzorkovací frekvenci, které mohou převodníky dosahovat (až 5 MHz), nepřipadá v úvahu obsluha každého přerušení při dokončení převodu. Další skutečností, která toto vylučuje, je ta, že v průběhu celé vzorkovací sekvence může (a v obvyklé aplikaci také bude) probíhat algoritmus DDS pro D/A převodník. Vzorkovací frekvence D/A převodníku je tedy limitována zejména volným výkonem jádra procesoru. Vše výše zmíněné si vynucuje použití řadiče DMA, čímž se zajistí ukládání dat do paměti bez zbytečného zatěžování jádra.

Kanály obou DMA řadičů jsou dány pro jednotlivé ADC tabulkami 27, resp. 28 v referenčním manuálu pro STM32F303^[3]. Proces inicializace DMA je přímou sekvencí příkazů:

- Povolení hodin pro DMA řadiče
- Inicializace struktury pro nastavení vlastností kanálu DMA (typu `DMA_InitTypeDef`):
 - Nastavení směru přenosu z periferie do paměti
 - Zakázání režimu přenosu z paměti do paměti
 - Nastavení velikosti dat v paměti na half-word (16 bitů), tak, aby byl ukazatel do paměti inkrementován po dvou bytech
 - Povolení inkrementace ukazatele do paměti
 - Nastavení normálního režimu (tj. ne kruhový režim)
 - Nastavení velikosti dat v periférii na half-word
 - Zakázání inkrementace ukazatele na data v periférii
 - Nastavení priority na "very high" (vzhledem k tomu, že DMA využívají jen ADC, na této hodnotě prakticky nezáleží)

- Poté se pro každý ADC zvlášť provede sekvence příkazů
 - Nastavení adresy dat periferie ve struktuře z předchozího bodu na DR (data register) příslušného ADC
 - Inicializace kanálu DMA funkcí ze standardní knihovny (jejím argumentem je, mimo ukazatele na registr daného kanálu, ukazatel na strukturu vyplněnou v předchozích bodech)
 - Povolení kanálu DMA
 - Povolení přerušení při dokončení přenosu (transfer complete interrupt)
 - Nastavení ADC na jednorázový přenos (nikoliv kruhový)
 - Povolit ADC generaci DMA requestů

6.3.9 Reset DMA (`ADC_DMA_reset`)

Před další vzorkovací sekvencí se provádí reset DMA řadičů. Při dřívější práci s procesory STM32 nastávaly problémy při použití kruhového adresování (automatického resetu adresy paměti a znovunačtení množství dat k přenesení), byl tedy zvolen přístup manuálního resetování. To je realizováno

- deaktivací všech tří kanálů řadičů přiřazených k přenosům ADC vzorků (kanál 1 řadiče č. 1, kanály 1 a 5 řadiče č. 2),
- nastavením registru CMAR (memory address register) příslušného řadiče a kanálu na začátek bufferu A/D převodníku, ke kterému kanál přísluší a
- nastavením registru CNDTR (number of data register) každého kanálu na polovinu velikosti registru příslušného ADC (každý vzorek má velikost 2 byty).

6.3.10 Spouštění vzorkovací sekvence

Jelikož mají A/D převodníky vzorkovat dokonale synchronně, nestačí pouze jeden za druhým je spustit, jelikož mezi spuštěním prvního a posledního A/D převodníku může uběhnout doba, která se bude blížit vzorkovací periodě. Jediným způsobem, jak převodníky spustit synchronně, je nakonfigurovat je na externí spouštění. Zde je možné využít jako externí signály události vyvolávané integrovanými časovači. Zvláště vhodnou je pro softwarové spouštění událost TRGO, kterou lze svázat s povolovacím bitem kteréhokoliv timeru.

Za tímto účelem byla napsána rutina `SetupEventGenerator`, která provede nejzákladnější konfiguraci časovače TIM1 pro účely generování události TRGO. A/D převodníky, nastavené na náběžnou hranu vnějšího spouštěče (viz konfigurace A/D převodníků – 6.3.7), se tak spouští při povolení předtím zakázaného časovače TIM1.

V praxi to funguje tak, že se při přijetí požadavku z hostitelského PC spustí A/D převodníky, jež nemají nulové buffery, jelikož jsou ale nastaveny na externí trigger, přejdou pouze do stavu čekání na spouštěč. Pokud se nečeká na synchronizaci s DDS generátorem tak se následně okamžitě povolí TIM1, čímž se spustí vzorkovací sekvence. V případě, že se na synchronizaci čeká, časovač TIM1 se povolí v obsluze přerušení časovače TIM6 při splnění patřičné podmínky (více v popisu obsluhy – 6.3.3).

6.3.11 Generace signálů

Díky tomu, že je na chipu STM32F303 obsažen dvoukanálový D/A převodník (DAC), je možné generovat dva signály bez použití externích obvodů. Vzhledem k požadavku na snadnou změnu frekvence generovaného signálu byl použit algoritmus DDS k adresování dat pro D/A převodník. Jeho výhodou je jednoduchost a z ní plynoucí rychlost. K jeho operaci stačí jeden fázový registr (v případě mikrokontroléru programovaného pomocí jazyka C se jedná o 32-bitový unsigned integer) a fázový inkrement.

Konfigurace D/A převodníku (`DAC_init`)

STM32F303 nedává v případě D/A převodníku na výběr z mnoha kanálů vyvedených na množství pinů jako je tomu u A/D převodníků. Má pouze 2 kanály a ty jsou pevně vyvedeny na piny PA4 a PA5. Konfigurace převodníku je tedy velice přímočará a sestává se z následujících kroků:

- Povolení hodin pro periférii (DAC) a pro GPIO bránu
- Nastavení pinů 4 a 5 brány GPIOA na analogové I/O bez pull up/down rezistorů
- Dále se opět využije struktura, definovaná standardní knihovnou, uchováující nastavení DAC, ve které se nastaví vlastnosti:
 - Spouštění se zvolí softwarové
 - Zakáže se automatická generace vlny (periferie umožňuje automatickou generaci trojúhelníkového signálu nebo šumu)
 - Zakázání výstupního bufferu
- Zápis společného nastavení do registrů obou kanálů převodníku
- Povolení převodníků
- Zápis hodnoty 0x800 (střed rozsahu) do doprava zarovnaného 12-bitového DHR (data holding register)
- Softwarové spuštění obou kanálů (zajišťující nulovou výstupní hodnotu po inicializaci)

Inicializace časovače TIM6 a konfigurace časové základny DDS generátoru (`TIM6_init`)

V této funkci se inicializuje časovač, který realizuje hodinový signál DDS generátoru. Délka periody tohoto časovače lze nastavit jednak pomocí předděličky hodinového signálu a jednak hodnotou, ve které se čítač nuluje. Oba tyto parametry mohou nabývat hodnot 1 až 65 535, lze tedy nastavováním jednoho z nich regulovat frekvenci přerušení v rozsahu f_{min} až $65\,535f_{min}$. Při testování DDS algoritmu bylo zjištěno, že jeho maximální obnovovací frekvence je asi 400 kHz.

Byla tedy zvolena pevná hodnota pro nulování timeru 7, což umožňuje nastavit teoreticky obnovovací frekvenci až 10,3 MHz, čemuž odpovídá minimální 157 Hz. Ovladač jednotky však omezuje tuto hodnotu na přibližně 412 kHz. Takto velká rezerva je zvolena proto, aby byla i při frekvencích nad 100 kHz dodržena přijatelná citlivost nastavení (nejbližší nižší obnovovací frekvence je přibližně 395 kHz).

Při inicializaci časovače se zvolí předdělicí poměr 103, čímž se stanoví výchozí obnovovací frekvence 100 kHz, čítač se nastaví na čítání směrem nahoru. Inicializace se zakončí povolením vyvolávání přerušení při nulování a aktivaci timeru.

Změna obnovovací frekvence DDS generátoru – `SetDACSamplerate`

Pro změnu obnovovací frekvence se nejdříve spočítá předdělicí poměr hodin, kterým se regulace obnovovací frekvence (UR) realizuje podle vztahu

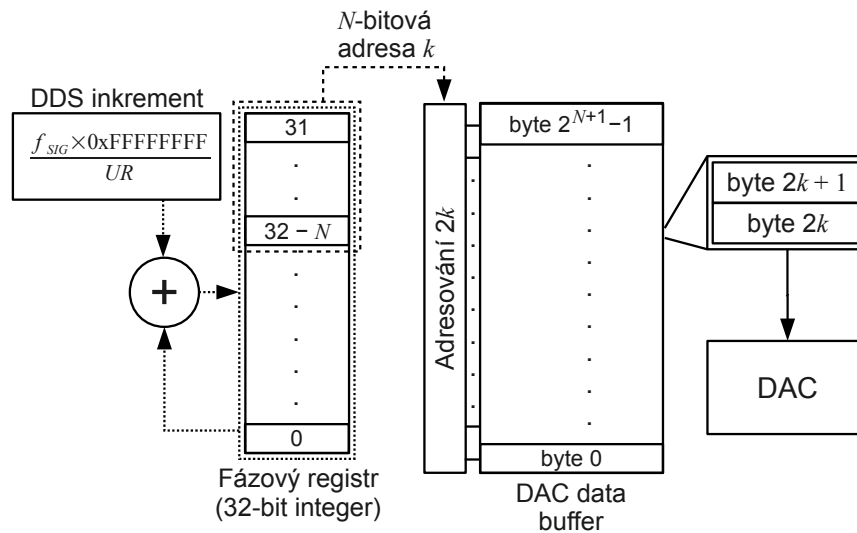
$$PSC = \frac{72\,000\,000}{7UR}, \quad (6.1)$$

který dělí frekvenci hodinového signálu, kterou je časovač synchronizován (72 MHz), resetovací hodnotou 7 násobenou novou obnovovací frekvencí. Jelikož musí být předdělicí poměr kladné celé 16-bitové číslo, je takto vypočtená hodnota případně upravena do rozmezí 1 až 0xFFFF (65 535). Hodnota předdělicího poměru se zapíše do příslušného registru časovače (TIM6_PSC). V průběhu uvedeného výpočtu docházelo k zaokrouhování, je tedy nutné přepočítat skutečnou obnovovací frekvenci podle vztahu 6.2, aby mohl být přepočítán inkrement DDS.

$$UR = \frac{72\,000\,000}{7PSC} \quad (6.2)$$

Implementace DDS

Schéma funkce DDS implementované ve firmwaru je na obrázku 6.8. Při každém přerušení vygenerovaném časovačem TIM6 dojde k přičtení inkrementu k fázovému registru (typu 32-bit integer). Poté se vezme horních N bitů tvořících adresu k , která se využije k adresaci bufferu, ve kterém je uloženo 2^N vzorků (při 12-bitových vzorcích je tedy velikost 2^{N+1} bytů). Do výstupního registru příslušného kanálu D/A převodníku se tedy zapíšou byty na adresách k a $k + 1$.



Obrázek 6.8: Grafické znázornění implementace DDS

DDS se inicializuje při změně frekvence generovaného signálu nebo obnovovací frekvence. To obnáší za prvé výpočet inkrementu podle vztahu na obrázku 6.8, za druhé výpočet velikosti posuvu pro převod

z hodnoty fázového registru na adresu k

$$\text{posuv} = 32 - N . \quad (6.3)$$

Potom se uvnitř cyklu počítá adresa

$$k = \text{fázový registr} \gg \text{posuv} . \quad (6.4)$$

Při inicializaci nedochází k nulování fázového registru, průběh tedy pokračuje od posledního vzorku. Inicializaci DDS inkrementu a posuvu implementuje funkce `InitDDS`, cyklus DDS je implementován v obsluze přerušení časovače `TIM6`.

7 Obslužný software pro PC

Kapitola si klade za cíl objasnit strukturu a principy, na základě kterých byl programován ovládací software. Jeho základem je knihovna pro komunikaci s měřicí jednotkou, zdrojový soubor *interfacing.c* a příslušný header *interfacing.h*. Knihovna dále využívá header file *defines.h*, který obsahuje výhradně makra společná pro firmware i ovladač pro PC. Funkcí těchto maker je přiřadit k jednotlivým bytovým identifikátorům jména pro snadnou orientaci v obou zdrojových kódech.

7.1 Ovladač jednotky (*interfacing.c/.h*)

Zdrojový kód ovladače jednotky obsahuje funkce, které jsou určeny k volání nadřazenou aplikací, ať už je její účel jakýkoliv. Slouží k tomu, aby zajistil jednoduché programování bez nutnosti studování komunikačního protokolu zavedeného pro komunikaci na sběrnici USB. Rovněž zajišťuje ošetření v případě odpojení jednotky a zajišťuje opětovnou inicializaci v případě, že je jednotka znovu připojena. V obou případech vykonává volitelnou callback funkci. Programátor nadřazené aplikace tak může zajistit, aby byla v případě odpojení zobrazena informativní zpráva, deaktivovány ovládací prvky softwaru atd.

7.1.1 Obecný popis

Jak přípona souboru napovídá, knihovna je psána v jazyce C. Využívá API *libusb* a *pthread*.

Knihovna *libusb* se využívá k vlastní komunikaci s jednotkou. Využívá se jednoduššího, blokovacího neboli synchronního, přístupu. To znamená, že při každém požadavku na přenos dochází k zastavení aplikace do jeho vykonání nebo do vypršení timeoutu, který je stanoven na 100 ms (s výjimkou čekání na navzorkovaná data, kde je timeout roven součtu doby trvání sběru dat, maximální doby čekání na synchronizaci a času 100 ms).

API *pthread* je využito pro vytvoření vlákna zajišťujícího monitoring spojení, opětovné připojení a vykonávání callback funkcí při připojení/odpojení jednotky.

7.1.2 Referenční manuál k ovladači jednotky

Tato část slouží jako manuál k ovladači jednotky. Jsou zde popsány funkce, které jsou k dispozici aplikačnímu programátorovi. U každé funkce je popsán její účel a argumenty, případně návratová hodnota. U funkcí, u kterých bylo uznáno za vhodné doplnit implementační detaily, tak bylo učiněno. Informace, které nejsou v této kapitole uvedeny, jsou snadno pochopitelné z přiloženého zdrojového kódu (soubor *Zdrojove_kody/PC_aplikace/Ovladac/interfacing.c*).

Společným aspektem všech "Set" funkcí (včetně funkce *IFC_InitializeADCs*) je naplnění připraveného bytového pole hodnotami podle protokolu (viz kapitola 5) (obvykle s využitím funkce *memcpy*) a jeho odeslání pomocí funkce *libusb_interrupt_transfer* na interrupt endpoint 1 jednotky.

```
void IFC_init(void (* connection_callback)(void),
              void (* disconnection_callback)(void))
```

- Popis:

- Inicializuje callback funkce prováděné při připojení/odpojení jednotky, inicializuje knihovnu *libusb* a spouští vlákno určené ke kontrole připojení a k jeho případnému opětovnému vytvoření

- Argumenty:

- `connection_callback`: ukazatel na funkci, která se má zavolat v případě připojení jednotky, NULL pokud se nemá vykonávat žádná callback funkce
- `disconnection_callback`: ukazatel na funkci, která se má zavolat v případě odpojení jednotky, NULL pokud se nemá vykonávat žádná callback funkce

```
void IFC_CheckConnectionThread(void)
```

- Popis:

- Vlákno pro kontrolu/obnovení připojení běží v nekonečné smyčce s periodou 0,1 s. Pokud je jednotka odpojena, pokouší se připojit pomocí funkce **IFC_ConnectToDevice** (viz níže) v případě úspěšného navázání spojení je vykonána callback funkce po připojení. Stejně tak se zavolá callback funkce při odpojení jednotky.

- Poznámky k implementaci:

- Jediným způsobem k detekci odpojení, který byl nalezen, je požadavek na nastavení konfigurace (požaduje se nastavení jediné používané konfigurace, takže se tím funkce jednotky neovlivňuje). Tak se z knihovny *libusb* získá chybový kód (nebo 0 v případě, že je jednotka připojena) bez realizace "relevantního" přenosu, který je zdržován timeoutem.

```
int IFC_ConnectToDevice(void)
```

- Popis:

- Pokusí se otevřít připojení k jednotce.

- Návrátová hodnota:

- 0: spojení bylo úspěšně navázáno
- `libusb_error`: došlo k chybě (viz dokumentace *libusb*^[9])

```
int IFC_GetADCData(unsigned short* adc_data_pointer,  
                   unsigned int data_length)
```

- Popis:
 - Vyžádá obsah datových bufferů A/D převodníků a uloží je v oblasti definované ukazatelem `adc_data_pointer`.
- Argumenty:
 - `adc_data_pointer`: ukazatel na místo v paměti, kam se mají uložit data pořízená A/D převodníky
 - `data_length`: délka v bytech všech dat, která se mají přenést
- Návrátová hodnota:
 - 0: Vše proběhlo v pořádku
 - `libusb_error`: došlo k chybě (viz dokumentace *libusb*^[9])
- Poznámky k implementaci:
 - Jelikož je využíváno synchronního *libusb* API, jsou požadavky na přenosy dat opatřeny timeouty, zpravidla délky 100 *ms*. Vzhledem k tomu, že může být délka vzorkovací sekvence v řádu několika sekund a navíc může být nastavena synchronizace s DDS generátorem, dochází v této funkci před vydáním požadavku k přepočítání timeoutu. Nejdříve je pro každý kanál spočítána délka vzorkovací sekvence z nastaveného předdělicího poměru, délky doby odběru vzorku a velikosti bufferu. V paměti nadále zůstává pouze největší výsledek. Dále je spočítáno zpoždění, které může nastat čekáním na synchronizaci. Pokud je synchronizace aktivní, vezme se převrácená hodnota frekvence signálu generovaného D/A převodníkem, podle kterého se synchronizuje. Oba uvedené příspěvky se sečtou, přidá se rezerva 100 *ms* a tato suma se využije jako timeout pro přenos navzorkovaných dat.

```
int IFC_SetDACData(char DAC_to_set_data, unsigned short* dac_data_pointer,  
                   unsigned char bit_depth)
```

- Popis:
 - Nastavení průběhů pro D/A převodníky.
- Argumenty:
 - `DAC_to_set_data`: číslo kanálu D/A převodníku, kterému se nastavují data, může být 1, 2 nebo 12 pro nastavení společných dat pro oba kanály
 - `dac_data_pointer`: ukazatel na místo v paměti, kde jsou uložena data k odeslání
 - `bit_depth`: počet bitů potřebných pro adresaci vzorků (např. 10 pro 1024 odesílaných vzorků)
- Návrátová hodnota:
 - 0: Vše proběhlo v pořádku
 - `OUT_OF_MEMORY (-1)`: V mikrokontroléru není dostatek volné paměti pro alokování zvolených velikostí bufferů

```
float IFC_SetDACSamplerate(unsigned int sample_rate)
```

- Popis:
 - Nastavení obnovovací frekvence D/A převodníku.
- Argumenty:
 - `sample_rate`: obnovovací frekvence pro nastavení D/A převodníku, maximum 100 000, při zadání vyšší se omezí na 100 000
- Návrátová hodnota:
 - Přepočítaná skutečně nastavená obnovovací frekvence

```
int IFC_SetADCBuffers(unsigned short* adc_buffer_sizes)
```

- Popis:
 - Nastavení velikostí bufferů pro ukládání vzorkovaných dat.
- Argumenty:
 - `adc_buffer_sizes`: ukazatel na pole `unsigned short` délky 3, každý prvek pole obsahuje jednu délku bufferu v bytech:

- * [0]: ADC 1
- * [1]: ADC 2
- * [2]: ADC 3

- Návrátová hodnota:

- 0: Vše problémlo v pořádku
- OUT_OF_MEMORY (-1): V mikrokontroléru není dostatek volné paměti pro alokování zvolených velikostí bufferů

```
void IFC_SetDACSignalFrequency(unsigned char DAC_channel,
                               unsigned int dac_signal_frequency)
```

- Popis:

- Nastavení frekvence s přesností na jednotky Hz vybraného kanálu DDS generátoru.

- Argumenty:

- DAC_channel: Číslo kanálu k nastavení frekvence
- dac_signal_frequency: Frekvence signálu v Hz (1 - 50 000)

```
void IFC_InitializeADCs(char* ADC_time_settings)
```

- Popis:

- Nastaví předděličky hodinového signálu a doby odběru vzorku A/D převodníků.

- Argumenty:

- ADC_time_settings: ukazatel na pole char[5], kde
 - * [0]: index předdělicího poměru pro ADC1 a ADC2
 - * [1]: index délky doby odběru vzorku pro ADC1
 - * [2]: index délky doby odběru vzorku pro ADC2
 - * [3]: index předdělicího poměru pro ADC3
 - * [4]: index délky doby odběru vzorku pro ADC3

Index pro požadovanou předděličku nebo délku odběru vzorku lze nalézt v tabulkách 7.1 a 7.2.

Tabulka 7.1: Tabulka výběru předděličky A/D převodníku

Index	0	1	2	3	4	5	6	7	8	9	10	11
Předdělička	1	2	4	6	8	10	12	16	32	64	128	256

Tabulka 7.2: Tabulka výběru délky doby odběru vzorku v počtu cyklů hodin A/D převodníku

Index	0	1	2	3	4	5	6	7
Počet cyklů	1,5	2,5	4,5	7,5	19,5	61,5	181,5	601,5

Tabulka 7.3: Tabulka vzorkovacích frekvencí pro všechny kombinace délek doby odběru vzorku a předděličky A/D převodníků (pro rozlišení 12 bitů)

Sampling time (cycles)	ADC Prescaler					
	1	2	4	6	8	10
1,5	5 142,857	2 571,429	1 285,714	857,143	642,857	514,286
2,5	4 800	2 400	1 200	800	600	480
4,5	4 235,294	2 117,647	1 058,824	705,882	529,412	423,529
7,5	3 600	1 800	900	600	450	360
19,5	2 250	1 125	562,5	375	281,25	225
61,5	972,973	486,486	243,243	162,162	121,622	97,297
181,5	371,134	185,567	92,784	61,856	46,392	37,113
601,5	117,264	58,632	29,316	19,544	14,658	11,726

Sampling time (cycles)	ADC Prescaler					
	12	16	32	64	128	256
1,5	428,571	321,429	160,714	80,357	40,179	20,089
2,5	400	300	150	75	37,5	18,75
4,5	352,941	264,706	132,353	66,176	33,088	16,544
7,5	300	225	112,5	56,25	28,125	14,063
19,5	187,5	140,625	70,313	35,156	17,578	8,789
61,5	81,081	60,811	30,405	15,203	7,601	3,801
181,5	30,928	23,196	11,598	5,799	2,899	1,450
601,5	9,772	7,329	3,664	1,832	0,916	0,458

```
void IFC_SetDACMode(unsigned char DAC_num, unsigned char mode)
```

- Popis:
 - Nastavuje režim pro vybraný D/A převodník.
- Argumenty:
 - DAC_num: Výběr kanálu D/A převodníku (1 nebo 2)
 - mode:
 - * DAC_MODE_OFF: převodník generuje 0x8FF – polovinu referenčního napětí

- * `DAC_MODE_AC`: je aktivní algoritmus DDS a generuje se signál uložený v bufferu pro příslušný kanál
- * `DAC_MODE_DC`: na výstupu převodníku se udržuje nastavená stejnosměrná hodnota viz **`IFC_SetDACDCValue`** níže

```
void IFC_SetDACDCValue(unsigned char DAC_num, unsigned short dc_value)
```

- Popis:
 - Nastaví stejnosměrnou hodnotu pro daný kanál. Aby byla hodnota generována, musí být nastaven kanál do stejnosměrného režimu, viz **`IFC_setDACMode`** výše.
- Argumenty:
 - `DAC_num`: výběr kanálu D/A převodníku (1 nebo 2)
 - `dc_value`: hodnota výstupu ve stejnosměrném režimu pro daný kanál

```
void IFC_SyncWithDAC(unsigned char DAC_num)
```

- Popis:
 - Umožňuje synchronizovat začátek vzorkovací sekvence s prvním vzorkem následující periody signálu generovaného vybraným kanálem D/A převodníku. Synchronizace je aktivní pouze, pokud je DDS generace vybraného kanálu D/A převodníku aktivní, tj. má nastavená data průběhu a je v režimu `DAC_MODE_AC` (viz **`IFC_SetDACMode`**).
- Argumenty:
 - `DAC_num`: Kanál D/A převodníku, podle kterého bude synchronizován počátek vzorkovací sekvence. Pro 1, resp. 2, je vybrán příslušný převodník, jakékoliv jiné číslo synchronizaci zakazuje

```
void IFC_GetSamplerate(unsigned char ADC_num, float* samplerate)
```

- Popis:
 - Vrací vzorkovací frekvenci nastavenou pro vybraný kanál.
- Argumenty:
 - `ADC_num`: Číslo A/D převodníku, jehož vzorkovací frekvence se vrací
 - `samplerate`: Adresa pro vrácenou vzorkovací frekvenci

```
void IFC_SetDACAmplitudes(unsigned char DAC_channel, unsigned short dac_amplitude)
```

- Popis:
 - Nastaví amplitudu signálu vybraného kanálu v rozmezí 0 - 4095 LSB.
- Argumenty:
 - `DAC_channel`: Kanál D/A převodníku pro nastavení amplitudy (1 nebo 2)
 - `dac_amplitude`: Amplituda v jednotkách LSB (0 - 4095)
- Poznámky k implementaci:
 - Jelikož je pro přepočítání signálu podle zvolené amplitudy zapotřebí původní signál o plném rozsahu (aby se dala amplituda snižovat a znovu zvyšovat bez ztráty rozlišení), bylo možné vybírat z několika možností
 - * Přepočítávat každý vzorek přímo v algoritmu DDS před jeho zápisem do registru D/A převodníku, čímž by se zvyšovalo zatížení mikrokontroléru a tedy snižovala obnovovací frekvence
 - * Mít v STM32F3 dva buffery, 1 pro původní a druhý pro signál se změněnou amplitudou, což by vyžadovalo dvojnásobnou paměť pro každý kanál
 - * Nakonec zvolený způsob: udržovat původní průběh v PC ovladači a při změně amplitudy odeslat všechna upravená data znovu. Tento způsob sice znamená větší vytěžování USB sběrnice ale vzhledem k maximální délce bufferu 38 kB to nepředstavuje problém
 - Při změně amplitudy se nejprve alokuje nové pole pro uložení upraveného signálu o stejné délce, jakou mají původní data. Dále se spočítá offset, který koriguje změnu stejnosměrné složky podle vztahu

$$offset = 2047 - 2047 \frac{amplituda}{4095} .$$

Poté se každý vzorek ve smyčce přes celou délku dat přepočte vzorcem

$$n_{new} = n_{orig} \cdot \frac{amplituda}{4095} + offset$$

a podle komunikačního protokolu se tato data přenesou do STM32F3.

7.2 Zdrojový soubor s funkcemi zpracování dat (*utils.c/.h*)

V souborech *utils.c/.h* v adresáři *Zdrojove_kody/PC_aplikace/Utils* jsou implementovány funkce, kterými lze jednoduše podrobit základní analýze navzorkovaná data. Mezi tyto patří funkce pro stanovení střední hodnoty, amplitudy, efektivní hodnoty, frekvence a fáze, funkce provádějící číslíkové lock-in zesílení, funkce vracející pole magnetických intenzit a indukci pro vykreslení hysterézní smyčky a dále pomocné funkce implementované podle potřeby ostatních funkcí. Jelikož se jedná, až na poslední jmenované, o z odborného hlediska zajímavé funkce, budou v této kapitole popsány co nejpodrobněji. Přesto se doporučuje pro plné pochopení otevřít si zdrojový kód (je-li k dispozici) a tento text využít ke zorientování se v programu.

7.2.1 Výpočet frekvence a fáze (**UTS_FrequencyPhase**)

Tato funkce byla vytvořena zejména jako podstatná součást algoritmu pro číslicovou lock-in filtraci, kde je nutné přesné určení frekvence a fáze synchronizačního signálu, kterým může být signál vzorkovaný jedním z A/D převodníků. Jedná se však o funkci, kterou lze upotřebit v nepřeberné řadě dalších aplikací. Prototyp funkce je

```
void UTS_FrequencyPhase(int meas_samplerate, int meas_length,
                        unsigned short* meas_data, double* frequency,
                        double* phase)
```

a její argumenty mají následující význam

- `meas_samplerate`: Vzorkovací frekvence, kterou byl záznam pořízen
- `meas_length`: Délka záznamu
- `meas_data`: Ukazatel na naměřená data
- `frequency`: Adresa pro uložení spočítané frekvence
- `phase`: Adresa pro uložení spočítané fáze

Popis implementace

Využívá se zde knihovny pro rychlou Fourierovu transformaci *FFTW*. Po zavolání se mimo jiné alokují pole `fft_in` a `fft_out`, typu `fftw_complex`, což je struktura obsahující reálnou a imaginární složku čísla. K alokaci se využívá funkce `fftw_malloc`. Po alokaci se do pole `fft_in` zkopírují vstupní data a následně se vytvoří tzv. plán zavoláním funkce `fftw_plan_dft_1d`, což je struktura zahrnující veškeré informace k provedení FFT (délka dat, ukazatel na vstupní data, ukazatel na výstupní data, směr FFT).

Vlastní FFT se vykoná zavoláním funkce `fftw_execute(fft_plan)`, čímž se do pole `fft_out` uloží výsledky transformace. Následuje smyčka, ve které se prochází pole s výsledky do jeho poloviny (do Nyquistovy frekvence) a hledá se největší hodnota (z toho plynoucí omezení je, že se frekvence signálu určí správně pouze u průběhů s dominantní první harmonickou). Vzhledem k tomu, že index nejvyšší hodnoty v procházených datech představuje zaokrouhlený počet period v zaznamenaném průběhu, tento krok vede ke zjištění frekvence velmi orientačně. Zjištěný index (`coarse_frequency`) nejvyššího binu diskretního spektra se dokonce ani nepřepočítává na frekvenci. Následuje uvolnění paměti zabírané poli `fft_in` a `fft_out`.

V dalším kroku se vypočítá nahrubo střední hodnota signálu `coarse_mean`. Nahrubo z toho důvodu, že se provádí sumace celého záznamu a nikoliv celistvého násobku period měřeného signálu.

Algoritmus popsáný v tomto odstavci byl navržen, aby byl co možná nejrobustnější po stránce přesnosti stanovení frekvence. Podmínkou však je, aby měl signál minimálně jednu souvislou čtvrtperiodu větší než je nahrubo spočítaná střední hodnota. Nejprve se hledá vzorek, jež má hodnotu menší než je `coarse_mean`. Proč se tato procedura provádí bude jasné po přečtení celého odstavce. Poté se hledá první vzorek s hodnotou větší než `coarse_mean`. Když je nalezen, vezmeme tento vzorek (označme jej n_k) a všechny následující vzorky, jež jsou rovněž větší než `coarse_mean` (tyto označme n_{k+1} , n_{k+2} ... n_{k+l}). Takto se dostanou vzorky tvořící jednu celou a souvislou plochu ohraničenou zezdola hodnotou `coarse_mean` a shora změřeným průběhem. Následuje výpočet těžiště této plochy podle vztahu

$$T = \frac{\sum_{i=k}^{k+l} i(n_i - coarse_mean)}{\sum_{i=k}^{k+l} (n_i - coarse_mean)}. \quad (7.1)$$

Těžiště T se poté uloží do pole s těžišti pokud platí, že $l + 1 > \frac{N}{4coarse_frequency}$, kde N je délka záznamu. Pokud toto neplatí, těžiště se zahodí. Tímto se předejde tomu, aby se považovalo za relevantní těžiště náhodně vzniklé plochy (například vlivem šumu). Po průchodu celým záznamem se poslední výpočet těžiště rovněž zahodí v případě, že je poslední vzorek větší než `coarse_mean`, a to ze stejného důvodu, z jakého se neuvažuje případná sekvence vzorků na začátku záznamu. Tímto důvodem je, aby se zamezilo hledání těžiště nekompletní plochy vystupující nad `coarse_mean`, čímž by došlo k posuvu těžiště a tím pádem k chybnému určení periody.

Perioda se vypočte jako rozdíl největší a nejmenší hodnoty těžiště v jejich seznamu vydělený počtem vyhodnocených těžišť sníženým o jednu. K výpočtu fáze se použije poloha prvního těžiště, která u sinusoidy a trojúhelníku odpovídá maximu, u obdélníku středu kladné půlperiody. Od polohy prvního těžiště se odečte čtvrtina periody, čímž se získá teoretická poloha nuly u výše zmíněných průběhů. Pokud je po tomto odečtu z nějakého důvodu poloha nuly záporná, vrátí se rovnou výsledek

$$\varphi = -2\pi \frac{T_0}{perioda}. \quad (7.2)$$

Pokud je poloha těžiště kladná a menší než je délka periody, provede se výpočet fáze podle vztahu

$$\begin{aligned} \varphi &= 2\pi \left(1 - \frac{T_0}{perioda}\right), T_0 > \frac{perioda}{2}, \\ \varphi &= -2\pi \frac{T_0}{perioda}, T_0 < \frac{perioda}{2}, \end{aligned} \quad (7.3)$$

v opačném případě se sníží o délku periody, aby se dostala do základního intervalu a tento postup se opakuje, dokud není znám výsledek.

Tato metoda z principu funguje na běžné průběhy jako je sinusoida, obdélník, trojúhelník a obecně všechny průběhy s jedním dominantním minimem a maximem v jedné periodě.

7.2.2 Lock-in zesilovač (`UTS_Lockin`)

V této části je popsána implementace číslicového lock-in zesilovače. Teoretický rozbor a vlastnosti tohoto řešení jsou popsány v kapitole 8. Lock-in zesilovač je implementován funkcí s prototypem

```
void UTS_Lockin(int reference_freq, int meas_samplerate,
                int meas_length, unsigned short* meas_data,
                unsigned short* ref_data, float* real,
                float* imag)
```

s argumenty

- `reference_freq`: Frekvence referenčního signálu pro vnitřní výpočet. Pokud je `NULL`, využije se k výpočtu frekvence referenčního signálu externích dat `ref_data`
- `meas_samplerate`: Vzorkovací frekvence, kterou byl pořízen záznam zpracovávaných dat `meas_data` a případného referenčního signálu `ref_data`

- `meas_length`: Délka záznamu (minimum z délek zpracovávaných dat a referenčních dat)
- `meas_data`: Ukazatel na data ke zpracování
- `ref_data`: Ukazatel na data referenčního signálu. V případě, že je `reference_freq` rovno NULL. V opačném případě ignorován.
- `real`: Adresa pro uložení reálné složky výsledku
- `imag`: Adresa pro uložení imaginární složky výsledku

Tato část programu začíná testem, zda byla explicitně zadána frekvence referenčního signálu f_{ref} . Pokud ano, pracuje se dál s touto frekvencí a fází referenčního signálu φ_{ref} se nastaví na nulu. V tomto případě je na uživateli, zda bude fázi zcela ignorovat nebo využije možnosti synchronizovat počátek sběru dat s implementovaným DDS generátorem. Pak je fáze referenčního signálu definovaná zpožděním a fázovým posuvem analogových obvodů zařazených mezi D/A převodníkem a měřeným objektem. Pokud frekvence referenčního signálu není zadána, je zavolána funkce `UTS_FrequencyPhase` pro zjištění frekvence a fáze referenčního signálu.

Dále je spočítán počet period referenčního signálu zaznamenaný ve zpracovávaných datech, aby se zajistila délka sumace přes celistvý počet period. Ten se vypočte podle vztahu

$$M = \left\lfloor N \frac{f_{ref}}{f_S} \right\rfloor, \quad (7.4)$$

kde N je délka dat a f_S je vzorkovací frekvence, kterou byla data pořízena a symbol $\lfloor \rfloor$ značí zaokrouhlení dolů. Následuje výpočet délky integrace

$$N_{int} = M \frac{f_S}{f_{ref}}. \quad (7.5)$$

Zbývající část programu je smyčka a konečné přenásobení realizující vztahy

$$\begin{aligned} Re\{A\} &= \frac{2}{N_{int}} \sum_{i=0}^{N_{int}-1} \left[n_i \sin \left(2\pi i \frac{f_{ref}}{f_S} + \varphi_{ref} \right) \right], \\ Im\{A\} &= \frac{2}{N_{int}} \sum_{i=0}^{N_{int}-1} \left[n_i \cos \left(2\pi i \frac{f_{ref}}{f_S} + \varphi_{ref} \right) \right], \end{aligned} \quad (7.6)$$

kde n_i je i -tý vzorek změřeného průběhu.

7.2.3 Výpočet hysterézní smyčky feromagnetika (`UTS_HysteresisLoop`)

Implementace této funkce vychází z teorie popsané v kapitole 11. Funkce ze vstupních dat naměřených na snímacím odporu pro měření proudu primárním vinutím a dat získaných měřením na výstupu sekundárního vinutí vypočítá vektory magnetické intenzity a indukce. Její prototyp je

```
void UTS_HysteresisLoop(unsigned short* magnetization_current_sensing_data,
                        unsigned short* output_voltage_data,
                        unsigned short length, float* sr)
```

a argumenty

- `magnetization_current_sensing_data`: Ukazatel na data naměřená na snímacím rezistoru pro měření proudu primárním vinutím
- `output_voltage_data`: Ukazatel na data naměřená na sekundárním vinutí vzorku
- `length`: Délka dat (počet vzorků vstupního proudu a výstupního napětí)
- `sr`: Ukazatel na vzorkovací frekvenci

a návratovou hodnotou ukazatelem na pole typu `float` s délkou $2 \times \text{length}$ obsahující vypočtené průběhy magnetické intenzity a indukce.

Na začátku si funkce načte parametry vzorku a hardwaru ze souboru *hist_loop_measurement_configuration* v adresáři projektu. Následuje výpočet frekvence signálu pomocí funkce `UTS_FrequencyPhase`.

Poté se pro návratové pole alokuje paměť pro $2 \times \text{length}$ hodnot typu `float`. Ukazatel na hodnoty H se nastaví na počátek nově alokované paměti a ukazatel na hodnoty B do poloviny. Tím je zaručeno, že jsou oba vektory seřazeny v jednom bloku, který lze vně funkce zpracovat.

Pomocí dříve zjištěné frekvence signálu se vypočte počet zaznamenaných period podle postupu použitého už v `UTS_Lockin`, tedy rovnice 7.4, a délka tohoto počtu period N_{int} vyjádřená v počtu vzorků podle rovnice 7.5.

Jelikož má z principu signál vzorkovaný převodníky přidanou stejnosměrnou složku `–offset` – a navíc může obsahovat přídavnou stejnosměrnou složku způsobenou tolerancemi součástek v přizpůsobovací elektronice, provádí se odstranění offsetu odstraněním průměru prvních N_{int} vzorků. Je nutné, aby se průměrování provádělo z přesného počtu period, aby byla stanovena přesně stejnosměrná složka. Toto platí zejména pro odstranění offsetu výstupních dat, při výpočtu indukce se napětí integruje (diskrétní vzorky sumují), a jak známo integrací konstanty je lineární funkce o sklonu rovném této konstantě, v konečném důsledku to znamená, že je k výslednému správnému průběhu indukce přičtena lineární funkce, což způsobuje posuv hysterézní smyčky s každou periodou po ose B . V případě, že je zaznamenáno např. 10 period signálu, je vykreslená smyčka velice nečitelná. Offset v hodnotách měřených na snímacím rezistoru nehraje tak kritickou roli, hodnoty se neintegrují, tudíž přítomnost offsetu způsobí posuv nezměněné smyčky v ose H , i to je však nežádoucí.

Výpočet vektoru intenzity se opírá o vztah 11.2. V cyklu se vypočtou všechny hodnoty následovně:

- Od každého vzorku se odečte offset, offsetovaný vzorek označme n_{off}
- Vzorek se pře počte na napětí vztahem

$$u_i = n_{\text{off}} \frac{U_{\text{ref}}}{4095A}, \quad (7.7)$$

kde U_{ref} je hodnota referenčního napětí A/D převodníku a A je zesílení řetězce pro úpravu signálu zařazeného mezi snímacím odporem a A/D převodníkem, obě tyto hodnoty se čtou na začátku funkce z konfiguračního souboru.

- Je spočtena hodnota intenzity podle vztahu 11.2, ve kterém je I_1 nahrazeno podílem napětí na snímacím odporu a jeho hodnoty R_{sn} , ve výsledku

$$H_i = \frac{u_i N_1}{R_{sn} l}. \quad (7.8)$$

Hodnoty použité k výpočtu jsou opět definované konfiguračním souborem.

Při výpočtu vektoru indukce se postupuje podle prvních dvou bodů postupu pro výpočet vektoru intenzit. Při výpočtu vektoru indukce se postupuje podle rovnice 11.8, která je modifikována do diskrétní formy, integrace se tedy transformuje na přičítání jednotlivých napětí lomených vzorkovací frekvencí, tím se rovnice upraví do tvaru

$$B_i = B_{i-1} - \frac{u_i}{f_S N_2 S_{Fe}}, \quad (7.9)$$

kde f_S je vzorkovací frekvence, N_2 počet závitů sekundárního vinutí definovaný v konfiguračním souboru spolu s průřezem jádra S_{Fe} . První hodnota indukce se počítá s $B_{i-1} = 0$, jelikož počáteční indukce je neznámá.

Jelikož není možné určit počáteční indukci, je nutné stanovit posun tak, aby byla hysterézní smyčka symetrická podle osy H

Struktura konfiguračního souboru

Struktura souboru s parametry měřeného vzorku a zapojení je patrná z obrázku 7.1. Kvůli použitému způsobu čtení musí být tato struktura přesně dodržena včetně absence mezer okolo symbolu = a odřádkování okamžitě za číslem, změny lze provádět pouze v číslech, přičemž N_1 a N_2 musí být vždy čísla celá, ostatní mohou mít desetinnou část, která se odděluje desetinnou tečkou.

```
l=0.071
N1=30
N2=5
Sfe=0.00002
Uref=2.94
A=2
Rsn=4.8
```

Obrázek 7.1: Struktura/příklad konfiguračního souboru používaného pro měření hysterézních smyček, hodnoty odpovídají zapojení a vzorku používanému v této práci

7.3 Zdrojový soubor grafického uživatelského rozhraní (*gui.c/.h*)

Grafické uživatelské rozhraní bylo implementováno pomocí API GTK+ 2, což je multiplatformní grafická knihovna, která byla zvolena z důvodu možné portovatelnosti softwaru na Windows.

Vzhledem k tomu, že GUI není z odborného hlediska na katedře měření zajímavé, omezí se tato kapitola popis uživatelského rozhraní jako takového, na jeho vlastnosti, chování a způsob ovládání. Co se programování týče, budou popsány pouze nejzajímavější segmenty programu.

7.3.1 Popis uživatelského rozhraní

Navržené uživatelské rozhraní je zobrazeno na obrázku 7.2. Jelikož tento software nemá primárně žádný specifický účel, bylo navrženo tak, aby bylo možné jednotku ovládat na úrovni jednotlivých

bodů uvedených v komunikačním protokolu (kapitola 5). V horní části GUI je combo box umožňující výběr společného předdělicího poměru pro A/D převodníky 1 a 2. Stejný combo box je v rámečku ADC3, jelikož tento převodník má vlastní předdělicí poměr (ve skutečnosti společný s ADC4, který je ale nepoužitý). Změna v kterémkoliv combo boxu nastavení délky odběru vzorku nebo předdělicího poměru automaticky tato nastavení přenáší do mikrokontroléru. Změna v textových polích se projeví až při změně combo boxu nebo po stisku tlačítka "Initialize ADCs".

V rámečku každého ADC lze zvolit velikost bufferu pro vzorkovaná data a délky odběru vzorků v počtu cyklů hodinového signálu. Velikosti bufferů jsou v bytech, tedy počet nashromážděných vzorků je poloviční (12-bitové hodnoty). Vzorkovací frekvence, které lze nastavit předdělicím poměrem a délkou odběru vzorku jsou uspořádány v tabulce 7.3.

Checkbox "Interprete as hysteresis loop" slouží k přepínání zobrazení dat. Když je nezatržený, data se zobrazují ve stylu osciloskopu pro všechny tři kanály. Zatržením se provede výpočet hysterézní smyčky pomocí funkce `UTS_HysteresisLoop` popsané výše v části 7.2.3 a vypočtená smyčka se zobrazí v grafu. Data pro výpočet hysterézní smyčky jsou data vzorkovaná kanály 1 a 2.

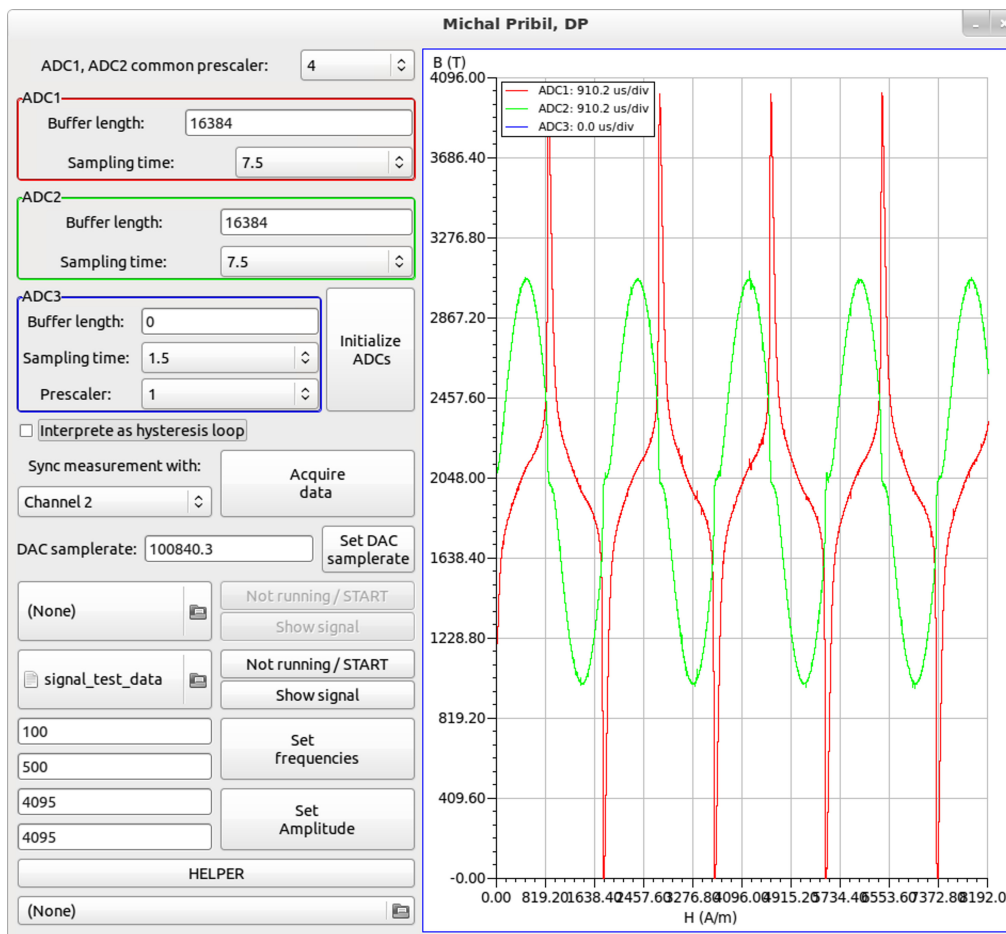
Níže jsou vedle sebe combo box označený "Sync measurement with:" s tlačítkem pro provedení odměru nastaveného v horní části. Combo box umožňuje spouštět vzorkovací sekvenci synchronně s počátkem signálu generovaného jedním z kanálů D/A převodníku.

Pole pro nastavení obnovovací frekvence DDS generátoru se po zadání hodnoty samo přepíše na hodnotu, která je skutečně nastavena.

Dále jsou v GUI dvě stejné skupiny tlačítek. Levé tlačítko slouží k výběru průběhu pro daný kanál generátoru signálů, tlačítko vpravo nahoře aktivuje/deaktivuje generátor. Pokud generátor neběží, je výstupní hodnota daného D/A převodníku v polovině rozsahu, což odpovídá nulovému napětí na výstupu celého systému. Tlačítko "Show signal" otevírá nové okno se zobrazeným průběhem signálu, který byl načten. Pokud je soubor s průběhem platný a jednotka je připojena k PC, data jsou okamžitě zapsána do bufferu STM32F303.

Horní dva textboxy ve skupině níže slouží k nastavení frekvence generovaného signálu, frekvence se do mikrokontroléru zapíše až stiskem tlačítka "Set frequencies", stejným způsobem se chovají i textboxy a tlačítko pro nastavení amplitudy průběhu (ve skutečnosti se jedná o peak-to-peak hodnotu v počtu kvant D/A převodníku).

Tlačítko "HELPER" je pozůstatek vykonávací funkce, která je zrovna zapotřebí při ladění, tlačítko pro výběr souboru bylo určeno pro ukládání vzorkovaných dat, nepodařilo se jej ale uvést do funkčního stavu kvůli nesouladu dokumentace a skutečného stavu knihovny GTK+ 2, další podrobnosti jsou nepodstatné.



Obrázek 7.2: Demonstrační grafické uživatelské rozhraní softwaru pro ovládání měřicí jednotky se STM32F303

7.3.2 Poznámky k implementaci GUI

Ve zdrojovém souboru *gui.c* jsou definovány callback funkce prováděné při připojení/odpojení jednotky, jejichž adresy jsou předávány funkci `IFC_Init`. Při odpojení jsou zakázány ovládací prvky, které přímo interagují s jednotkou, tzn. tlačítko pro sběr dat, tlačítko nastavení obnovovací frekvence, tlačítka spouštění DDS generátorů a tlačítka pro nastavení frekvence a amplitudy. Dál je umožněno nastavovat všechny parametry, které se při připojení jednotky automaticky přenesou včetně průběhů pro DDS generátory. Přírozenou vlastností je, že jsou po připojení generátory vypnuty, je to žádoucí chování předcházející nečekaným událostem. Po odpojení a připojení je tedy nutné generátory zapnout, pokud je jejich běh požadován.

Většina funkcí obsažená v souboru *gui.c* implementuje callback funkce jednotlivých ovládacích prvků, které většinou provádějí pouze přenos dat mezi prvky v grafickém uživatelském rozhraní a ovladačem jednotky implementovaným funkcemi v souboru *interfacing.c*.

Funkce, které obsahují kód specifický pro grafické API GTK+ 2, jsou:

- `GUI_BuildGUI`: Hlavní inicializační
- `GUI_InitializeWidgets`: Definice ovládacích prvků
- `GUI_PlaceWidgets`: Umístění ovládacích prvků
- `GUI_ConnectSignals`: Přiřazení signálů generovaných ovládacími prvky ke callback funkcím

K vykreslování grafu je použita knihovna `gtkgraph`.

8 Lock-in zesilovač a test jeho implementace

Lock-in zesilovač je specializované zařízení určené k měření slabých střídavých napětí známé frekvence. Předními výrobci špičkových lock-in zesilovačů jsou například Stanford Research Systems nebo Zurich Instruments.

V této kapitole je uveden princip, na kterém jsou lock-in zesilovače založeny, a to do hloubky, v jaké byl nakonec implementován ve vyvíjené softwarové sadě. Konkrétní implementace je popsána v kapitole 7.2.2. Na konci této části je popsán způsob testování funkčnosti implementace a jsou zde uvedeny jeho výsledky.

8.1 Princip funkce

Funkce lock-in zesilovače je založena na násobení měřeného průběhu referenčním sinusovým průběhem, což je vyjádřeno rovnicí převzatou z [6]

$$V_{\text{psd}} = V_{\text{sig}} V_L \sin(\omega_r t + \theta_{\text{sig}}) \sin(\omega_L t + \theta_{\text{ref}}) , \quad (8.1)$$

kde V_{psd} je výstupní napětí fázově citlivého detektoru (PSD), což je jednoduše řečeno násobička, V_{sig} je amplituda harmonické zpracovávaná detektorem, V_L je amplituda referenční sinusoidy, kterou je vstupní signál násoben, ω_r a ω_L jsou úhlové frekvence referenčního a měřeného signálu a θ_{sig} , θ_{ref} jsou fáze měřeného signálu a referenčního signálu.

Využitím vztahů pro goniometrické funkce se uvedená rovnice rozloží na

$$V_{\text{psd}} = \frac{1}{2} V_{\text{sig}} V_L \sin([\omega_r - \omega_L]t + \theta_{\text{sig}} - \theta_{\text{ref}}) - \frac{1}{2} V_{\text{sig}} V_L \sin([\omega_r + \omega_L]t + \theta_{\text{sig}} + \theta_{\text{ref}}) , \quad (8.2)$$

kde jedna složka představuje signál součtové frekvence $\omega_r + \omega_L$ a druhá signál na rozdílové frekvenci $\omega_r - \omega_L$. V případě, že jsou obě frekvence ω_r i ω_L stejné, rozdílová frekvence je nulová a výstupní napětí je stejnosměrné.

Jak z této teorie vyplývá, měřenosnou složkou výstupu PSD je stejnosměrné napětí a jelikož všechny ostatní harmonické obsažené v měřeném signálu, včetně šumu, vytváří své vlastní produkty s referenčním signálem, avšak na nenulových frekvencích, je nutné se těchto produktů zbavit. K tomu je v systému použit dolnoproustný filtr, který, v ideálním případě, všechny ostatní složky signálu odfiltruje. Dolnoproustný filtr ale omezuje i změny v užitečné stejnosměrné složce, musí být tedy jeho mezní frekvence zvolena jako kompromis mezi rychlostí odezvy a schopností odfiltrovat nežádoucí složky.

Jelikož měřený signál nemusí být ve fázi s referenčním signálem, což je zpravidla způsobeno fázovým posuvem v měřeném objektu, používají se dva PSD s fází referenčního signálu posunutou o 90° , čímž se získá reálná a imaginární složka měřeného signálu ve vztahu k referenci a je možné spočítat amplitudu měřeného signálu jako absolutní hodnotu komplexního čísla a fázi pomocí funkce \arctg .

Zde uvedená teorie zachycuje jen základní princip lock-in zesilovačů použitý v implementaci v této práci, podrobný výklad je obsažen v aplikačním listu Stanford Research Systems^[6], ze kterého je teorie čerpána.

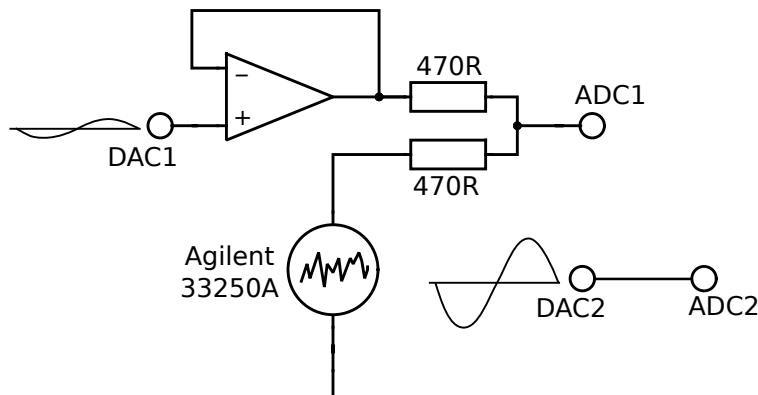
8.2 Způsob implementace lock-in zesilovače

Lock-in zesilovač nebyl hlavní prioritou této práce a proto se implementoval jako poslední hlavní funkce celého systému. Ten byl navrhován tak, aby umožňoval vzorkovat vysokými vzorkovacími kmitočty a je konfigurován tak, aby data vzorkovaná A/D převodníky byla přesouvána do RAM mikrokontroléru pomocí DMA. Algoritmus pro zhodnocení, zda je vzorkovací frekvence dostatečně malá na to, aby bylo možné v reálném čase provádět lock-in zesílení signálu, by byl pravděpodobně komplikovanou záležitostí a každopádně by musel být proveden zásadní zásah do původní filozofie návrhu firmwaru. Implementaci přímo v MCU by mohla vést k nestabilitě funkce firmwaru a vyžadovala by velké množství paměti.

Algoritmus pro lock-in filtraci lze přitom realizovat až v hostitelském PC, tím pádem nezvyšovat zátěž jádra mikrokontroléru a neomezovat jeho parametry (vzorkovací frekvence, simultální generace signálů). Lock-in zesilovač byl tedy realizován ve zdrojovém souboru pomocných funkcí *utils.c*. Podrobný popis implementace je uveden v popisu ovládacího softwaru pro PC (kapitola 7.2.2).

8.3 Testování implementace lock-in zesilovače

Testování algoritmu lock-in zesilovače bylo realizováno za podmínek odpovídajícím typickému nasazení lock-in zesilovačů. Ověřována je jeho schopnost detekovat amplitudu a fázi signálu do značné míry znehodnoceného šumem. Schéma zapojení pro toto měření je zachyceno na obrázku 8.1. Měřený signál je tvořen D/A převodníkem s nastavenou malou výstupní amplitudou odděleným od zbytku obvodu operačním zesilovačem v zapojení sledovače. Superponovaný šum se generuje generátorem Agilent 33250A. Oba signály se sčítají zhruba s polovičními váhami na rezistorrech podle Théveninova teorému a tento signál je přiveden do A/D převodníku. Referenční signál je vytvářen druhým kanálem D/A převodníku a přiváděn přímo do A/D převodníku číslo 2.



Obrázek 8.1: Schéma pro testování lock-in zesilovače

Měření probíhalo v sériích po 5ti odměrech pro tři různé kombinace vzorkovací frekvence a frekvence měřeného/referenčního signálu pro efektivní hodnoty šumu nastaveného na generátoru v řadě 250 mV, 500 mV, 750 mV a 1 V. Pro každou kombinaci vzorkovací frekvence a frekvence signálu byla zaznamenána jedna srovnávací hodnota s šumem nastaveným na 1 V_{RMS} , což je nejmenší možná hodnota.

Pro zachování stále stejné amplitudy signálu generovaného kanálem číslo 1 D/A převodníku musel být totiž generátor připojen a jeho výstup aktivován (při deaktivaci výstupu dochází k fyzickému odpojení pomocí relé). Srovnávací hodnota slouží k vyhodnocení přesnosti lock-in zesilovače při měření signálu s nezanedbatelným šumem. Délka všech záznamů byla 8192 vzorků. Jak je popsáno v 7.2.2, dolnopro-
pustný filtr je realizován jako klouzavý průměr s délkou průměrování rovnou největšímu celistvému počtu period v záznamu.

Zhodnocení výsledků

Všechny výsledky jsou uvedeny v příloze A, statisticky zpracované hodnoty jsou v tabulce 8.1.

SNR signálu bylo stanoveno přímo ze změřeného průběhu analýzou provedenou v Matlabu. Nejprve byla vždy provedena FFT srovnávacího průběhu a byla pro toto spektrum vypočtena efektivní hodnota podle vztahu

$$V_{\text{RMS}} = \sqrt{\sum_{i=1}^N FFT_i^2}, \quad (8.3)$$

kde N je délka dat/FFT. Poté byly vypočteny FFT pro každý průběh, pro který je třeba určit SNR a od tohoto spektra bylo odečteno spektrum spočítané pro srovnávací signál. Z takto upraveného spektra byla vypočtena efektivní hodnota šumu $V_{\text{RMS,noise}}$, potom

$$SNR = 20 \log \frac{V_{\text{RMS,sig}}}{V_{\text{RMS,noise}}}. \quad (8.4)$$

Z tabulky 8.1 není patrná jakákoliv závislost přesnosti měření na počtu period měřeného signálu obsaženém v záznamu. To dává smysl, jelikož současným přeskládáním vzorků se stejným pořadovým číslem referenčního a měřeného signálu lze docílit umělé změny počtu period z pohledu užitečného signálu, zároveň se tím výsledná průměrovaná hodnota nezmění. Při této úvaze lze ignorovat šum, jelikož se jedná o nekorelovaný šum, jeho spektrální charakteristika se tímto přeskládáním tedy nezmění. V konečném důsledku je tedy přesnost určení amplitudy a fáze užitečného signálu dána v zásadě pouze délkou průměrování, která určuje schopnost odfiltrovat nežádoucí složky (uvažuje-li se, že je užitečný signál s referenčním za všech okolností synchronní).

Jak lze vyčíst z tabulky 8.1, nejhorší výsledek ze všech 60ti měření byl zachycen při SNR signálu $-15,6$ dB, a to relativní chyba amplitudy signálu $21,05$ %, což odpovídá $SNR = 20 \log(1/0.2105) = 13,5$ dB, což znamená zlepšení SNR o 29 dB. Nejnepřesnější určení fáze vztahené k velikosti šumu bylo zjištěno $13,66^\circ$ při SNR měřeného signálu $-19,1$ dB.

Lze dokázat, že stejného výsledku, kterého se dosáhne prodloužením délky klouzavého průměrování, lze dosáhnout průměrováním hodnot získaných v oddělených po sobě jdoucích úsecích dat, tedy že se zprůměrováním k hodnot získaných z jednotlivých záznamů docílí stejného SNR jako při pořízení a vyhodnocení k -krát delšího záznamu dat.

Toto tvrzení potvrzují poslední dva sloupce tabulky, ve kterých je výsledná hodnota výstupu brána jako průměr ze všech původních výsledků. Maximální chyba určení amplitudy je rovna $8,32$ %, odpovídající zlepšení SNR o 37 dB, přičemž většina hodnot byla určena s chybou do 3 %, což znamená zlepšení SNR o asi 50 dB. Chyba určení fáze byla maximální v ojedinělém případě, a to necelých 8° , všechny ostatní hodnoty fáze měly chybu max. $2,25^\circ$.

Tabulka 8.1: Zpracovaná data pořízená při testování algoritmu lock-in zesilovače, každý řádek odpovídá jedné kombinaci frekvence měřeného signálu f_{sig} , vzorkovací frekvence f_S a SNR měřeného signálu

	SNR (dB)	$\delta V _{\max}(\%)$	$\overline{\delta V }(\%)$	$\delta\varphi_{\max}(\circ)$	$\overline{\delta\varphi}(\circ)$	$\overline{\delta V }(\%)$	$\overline{\delta\varphi}(\circ)$
$f_{sig} = 100$ Hz	-9,5	-2,60	2,61	-2,60	1,34	1,68	0,30
	-15,6	21,05	8,32	-4,92	2,60	8,32	0,63
$f_S = 243\,243$ Hz	-19,1	18,97	10,85	10,07	6,02	4,85	2,25
	-21,2	12,65	9,72	-13,67	5,71	-0,58	-1,90
$f_{sig} = 100$ Hz	-9,5	-3,35	1,27	-2,96	1,22	-1,12	-1,22
	-15,6	6,46	2,76	6,03	2,48	0,99	0,99
$f_S = 30\,405$ Hz	-19,1	-12,45	6,58	-12,56	7,91	-1,96	-7,91
	-21,2	-20,38	10,37	8,57	3,75	2,21	1,04
$f_{sig} = 1000$ Hz	-9,5	5,60	3,31	-3,29	1,69	2,60	-0,29
	-15,6	3,79	3,27	3,25	1,06	-0,26	0,59
$f_S = 30\,405$ Hz	-19,1	13,45	9,25	13,66	5,41	4,11	2,01
	-21,2	-14,06	8,47	4,96	3,25	-2,42	0,60

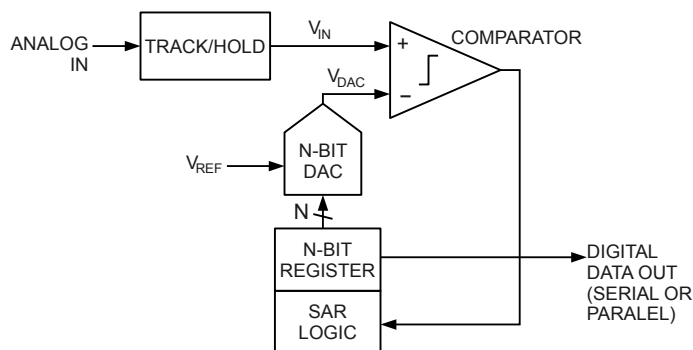
- kde
- SNR – odstup signál-šum měřeného signálu
 - $\delta|V|_{\max}$ – maximální chyba absolutní hodnoty amplitudy výstupního napětí ze všech 5ti měření
 - $\overline{\delta|V|}$ – střední hodnota všech absolutních chyb absolutní hodnoty amplitudy výstupního napětí
 - $\delta\varphi_{\max}$ – maximální chyba fáze výstupního napětí vůči referenčnímu signálu ze všech 5ti měření
 - $\overline{\delta\varphi}$ – střední hodnota všech absolutních chyb fáze výstupního napětí vůči referenčnímu signálu
 - $\overline{\delta|V|}$ – chyba střední hodnoty všech absolutních hodnot amplitudy výstupního napětí
 - $\overline{\delta\varphi}$ – chyba střední hodnoty všech fází výstupního napětí vůči referenčnímu signálu

9 Měření vlastností A/D převodníků

V této kapitole jsou popsány metody měření vlastností A/D převodníků a vyhodnocení nasbíraných dat. Základními vlastnostmi, které se uplatňují při jakémkoliv použití A/D převodníku, ať už při měření pomalu se měnících stejnosměrných napětí nebo při stroboskopickém měření signálů o frekvencích mnohonásobně převyšujících vzorkovací frekvenci převodníku, jsou statické vlastnosti. Těmi se rozumí offset, tedy minimální detekované napětí, referenční napětí určující maximální měřitelné napětí a zejména linearita převodníku. Dynamické vlastnosti budou hrát roli při měření rychle se měnících napětí, ať už napěťových skoků či vysokofrekvenčních signálů. Z dynamických vlastností se tato práce zabývá šířkou pásma převodníku a jeho jitterem, tedy nekoherencí odběru vzorků.

9.1 A/D převodníky s postupnou aproximací a jejich vlastnosti

A/D převodník, který je přítomen v procesoru STM32F303 patří do kategorie nejběžněji používaných A/D převodníků. Tou jsou převodníky s postupnou aproximací (SAR – Successive Approximation Register, blokové schéma je na obrázku 9.2). Ve srovnání s integračními D/A převodníky vyniká zpravidla podstatně vyšší rychlostí za cenu obětování jejich přirozené filtrační vlastnosti umožňující eliminaci střídavého rušení známé frekvence (touto frekvencí je obvykle frekvence síťového napětí, tedy 50 Hz). Na opačné straně spektra stojí paralelní A/D převodníky, které jsou oproti SAR převodníkům výrazně dražší, jelikož vyžadují 2^n (kde n je rozlišení převodníku v bitech) přesných rezistorů a rychlých komparátorů, což v případě 12-bitového převodníku procesoru STM32F303 znamená 4096 výše zmíněných elektronických prvků. Další výhodnou vlastností SAR převodníků je definovaná doba převodu, která je přímo úměrná rozlišení převodníku v bitech. Například převod převodníku procesoru STM32F303 trvá $n + \frac{1}{2}$ cyklu.



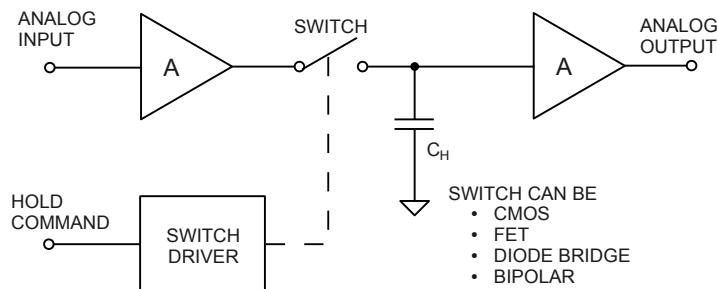
Obrázek 9.1: Blokové schéma A/D převodníku s postupnou aproximací (převzato z [1])

Popis funkce

Popis funkce se opírá o obrázek 9.2. Blok TRACK/HOLD (nebo také SAMPLE/HOLD, ačkoliv literatura [4] poukazuje na drobné rozdíly v obou termínech) zajišťuje neměnnost analogové hodnoty na vstupu A/D převodníku po dobu převodu. Tento blok se skládá ze vstupního zesilovače (obvykle

napětového sledovače).

Blok SAMPLE/HOLD je určující pro dynamické vlastnosti A/D převodníku a má charakter dolní propusti (jedná se o zesilovač s kapacitní zátěží). Jeho přenosová charakteristika (u STM32F303) je změřena v části 9.3.1.



Obrázek 9.2: Schéma obvodu SAMPLE/HOLD (převzato z [4])

Další částí je D/A převodník (N-BIT DAC), jehož rozlišení určuje rozlišení celého D/A převodníku. Do tohoto převodníku vstupují data z příslušného registru (N-BIT REGISTER), jehož obsah je řízen řídicí logikou (SAR LOGIC). Ten v prvním cyklu převodu nastavuje MSB registru na 1 a poté podle výstupu komparátoru (COMPARATOR) zhodnotí, zda je výstupní hodnota D/A převodníku větší než je vstupní hodnota. Pokud ano, hodnota MSB se vynuluje, pokud ne, zůstane nastavena na 1. Stejný postup se opakuje i pro ostatní bity registru směrem k LSB. V okamžiku, kdy je srovnání provedeno po nastavení LSB, je registr zkopírován do výstupního registru převodníku.

Statické vlastnosti převodníku jsou dány jednak linearitou použitého D/A převodníku a jednak statickými vlastnostmi komparátoru, zejména jeho vstupním offsetem, který musí být výrazně menší než je rozlišovací schopnost převodníku.

9.2 Měření statických vlastností A/D převodníků

Tato kapitola se zabývá metodami měření statických vlastností A/D převodníků. Navzdory tomu, že se jedná o statické parametry, měření samotné je do jisté míry dynamické, jelikož testovací napětí jsou zásadně časově proměnná. Samozřejmě se však používají frekvence, resp. strmosti, průběhů takové, aby při nich dynamické jevy odehrávající se v převodníku měly zcela zanedbatelný vliv na prováděná měření.

Vlastnosti A/D převodníků jsou obdobné těm u D/A převodníků, ale důkladnější zamyšlení vede k závěru, že jejich testování je diametrálně odlišné. Zatímco při měření statických vlastností D/A převodníků stačí staticky nastavovat hodnoty a měřit výstupní napětí, u A/D převodníků podobně triviální metoda neexistuje. Při stanovování napětových úrovní rozhraní mezi dvěma kódovými slovy bychom museli přivést na vstup převodníku napětí trojúhelníkového průběhu a poté detekovat okamžik, kdy došlo ke změně výstupu A/D převodníku. V tom okamžiku bychom museli změřit okamžité napětí přivedené na převodník a tak stanovit příslušnou napětovou úroveň. V praxi je ovšem tento přístup velice nevhodný. Jednak by strmost trojúhelníkového průběhu musela být tak malá, aby se neuplatnila ve změně napětí během doby od reakce převodníku na překročení rozhodovací úrovně až po ukončení odměru referenčním voltmetrem. Tento problém by se dal při pevné a známé době zpoždění kom-

penzovat, je tu ale ještě závažnější problém. Tím je šum, který je pozorovatelný ve všech průběžích pořízených vyšetřovaným A/D převodníkem. Díky tomuto šumu může dojít k překlopení úrovně kdykoliv v rozmezí úrovní dalece přesahujícím LSB převodníku.

Alternativní metoda odstraňující výše zmíněné potíže byla nalezena v dokumentu dostupném na webu MIT^[8]. Jedná se o histogramový test linearity. Ač je konečný přístup k měření krajních napětí rozsahu převodníku velice odlišný od toho, který popisuje Lundberg, byl jeho výkladem inspirován. Při měření linearit se jeho postup využívá již poměrně věrně, jen je místo sinusového testovacího signálu využít přechodný děj, okolo kterého musela být vystavěna potřebná teorie tak, aby mohl v této metodě nahradit sinusový průběh.

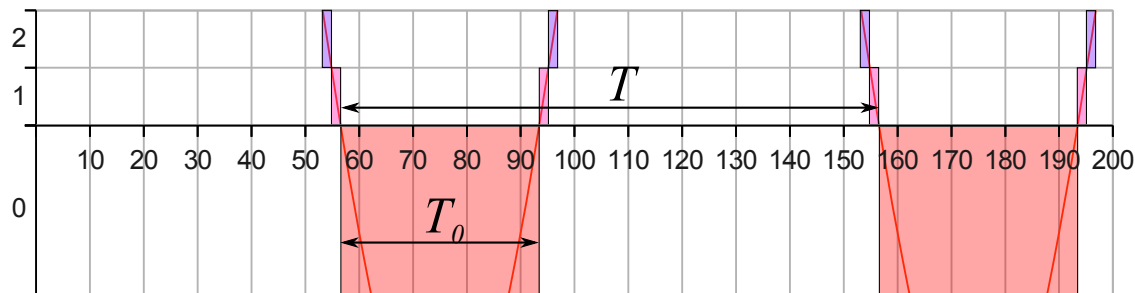
9.2.1 Stanovení offsetu a reference

Podle Lundberga^[8] lze offset a referenci stanovit pomocí testu histogramem. Tento test vyžaduje přesné změření amplitudy a střední hodnoty vstupního signálu. Tento test nemohl být realizován pomocí DDS generátoru Agilent 33250A, jelikož má stejné rozlišení D/A převodníku jako je rozlišení měřeného A/D převodníku, a generátor HP 8647A dovoluje minimální výstupní frekvenci 250 kHz. V datasheetu k použitému voltmetru HP 34401A^[10] se však uvádí v rozmezí frekvencí měřeného signálu 100 kHz až 300 kHz chyba měření True RMS $\pm(4\% \text{ of reading} + 0.50\% \text{ of range})$, což by znamenalo nepřijatelnou chybu určení offsetu a reference (min. $\pm 40 \text{ mV}$).

Byl tedy zvolen jiný přístup. Využít střídavý signál takové frekvence, při níž má použitý voltmetr nejmenší chybu. Pro tento účel byl využit generátor Agilent 33250A. Vzhledem k nedostatečnému rozlišení D/A převodníku generátoru na něm byla nastavena velice malá amplituda (70 mV), čímž se dosáhlo toho, že kvantum D/A převodníku generátoru je v měřicím signálu podstatně menší než kvantum A/D převodníku, při obnovovací frekvenci signálu 200 MHz lze potom tento signál považovat v rámci tohoto měření za velice kvalitní sinusoidu. Malá výstupní amplituda vyžadovala nastavit offset signálu velice blízko k nule, resp. referenci, měřeného A/D převodníku, ale vždy tak, aby byla ještě v rozsahu převodníku a daly se beze změn aplikovat metody popsané dále. Frekvence testovacího signálu byla 1 kHz.

K vysvětlení stanovení offsetu pomůže obrázek 9.3, na kterém je zachycen střídavý testovací signál a barevnými obdélníky je vyznačen výstup převodníku. Posčítáním všech nulových vzorků a vydělením této sumy délkou periody (udané počtem vzorků) se získá rozsah argumentů sinusoidy popisující testovací signál, ve kterém je vstupní signál menší než je offset převodníku, matematicky zapsáno

$$\Delta\alpha = 2\pi \frac{T_0}{T}. \quad (9.1)$$



Obrázek 9.3: Demonstrace odečtu hodnot pro zjištění offsetu

Dále se využije princip dobře pochopitelný z obrázku 9.4 a podle vztahu v něm uvedeném se vypočte hodnota sinu, která odpovídá zjištěnému rozsahu argumentů $\Delta\alpha$. V obrázku je $\Delta\alpha$ přenesena pro snadný výpočet do první půlperrody, což vyhovuje vztahu pro stanovení reference, pro stanovení offsetu se využije známého pravidla

$$\sin \beta = -\sin(\beta + (2k + 1)\pi), k \in \mathbb{Z} \quad (9.2)$$

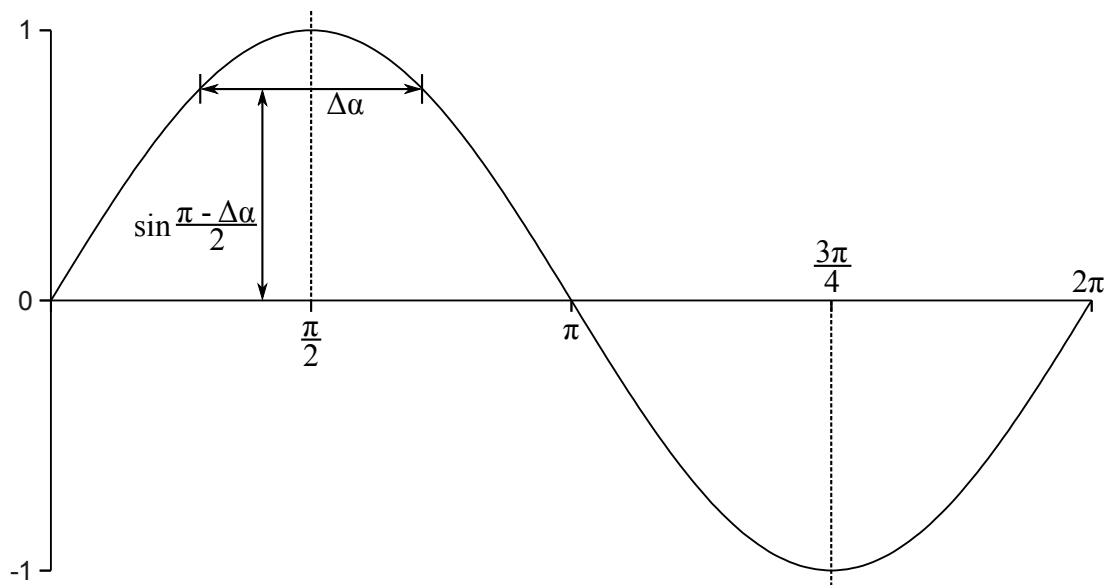
Hodnota offsetu se tedy vypočte podle vztahu

$$U_O = U_{DC} - U_{AC} \sin \frac{\pi - \Delta\alpha}{2} = U_{DC} - U_{AC} \sin \left(\pi \left(\frac{1}{2} - \frac{T_0}{T} \right) \right), \quad (9.3)$$

kde U_{DC} je střední hodnota testovacího průběhu a U_{AC} je jeho amplituda. Pro referenci platí stejný vztah, jen se součtem a nikoliv rozdílem, tedy

$$U_R = U_{DC} + U_{AC} \sin \frac{\pi - \Delta\alpha}{2} = U_{DC} + U_{AC} \sin \left(\pi \left(\frac{1}{2} - \frac{T_{4095}}{T} \right) \right), \quad (9.4)$$

kde T_{4095} je počet vzorků majících hodnotu 4095, tedy odpovídajících napětí nad referencí převodníku.

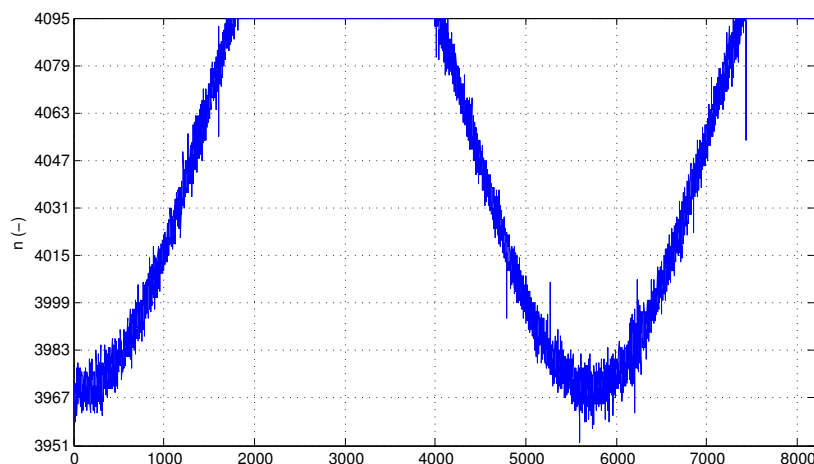


Obrázek 9.4

Data pro vyhodnocení těchto parametrů byla tvořena mnoha záznamy dlouhými 8192 vzorků (pro určení offsetu 115, pro určení reference 86 záznamů). Z každého byl poté vyňat úsek odpovídající jedné periodě a v tomto úseku byl vyhodnocen poměr nulových vzorků, respektive full-scale vzorků. Tyto poměry se poté zprůměrovaly a byly vypočteny hodnoty offsetu a reference podle vztahů 9.3 a 9.4. Jak si lze všimnout na ukázce jednoho takového záznamu na obrázku 9.5, průběh je zatížen výrazným šumem, který ale při použití této metody nehraje zásadní roli. Uvážíme-li totiž, že je průběh v rozmezí hodnot daném amplitudou šumu lineární, z dlouhodobého hlediska se vliv šumu vyruší.

	Offset	Reference
U_{DC} (mV)	20,09	2 929,3
U_{AC} (mV)	70,54	69,96
U_N (mV)	2,675	2934,8

TODO doplnit výsledky a chyby



Obrázek 9.5

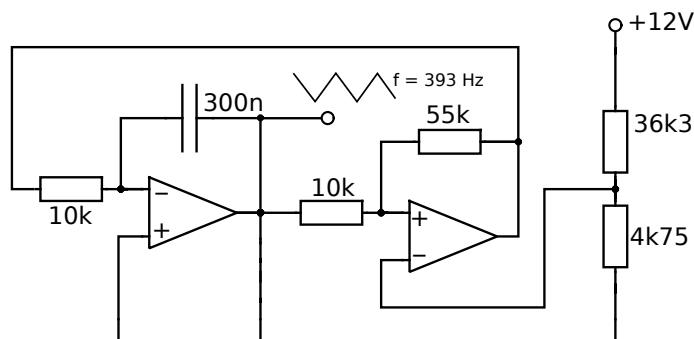
9.2.2 Měření nelinearit A/D převodníků

Lundberg ve svém listu o testování A/D převodníků^[8] popisuje metodu testu histogramem pro vyšetřování jejich nelinearit. Ačkoliv se jedná teoreticky o jednoduše proveditelné měření, ukázalo se, že s vybavením, které bylo v laboratoři k dispozici, nelze tuto metodu aplikovat bez výrazných modifikací.

Při měřeních byly k dispozici 2 generátory: sinusový oscilátorový generátor HP 8647A ($f = < 250 \text{ kHz}, 1 \text{ GHz} >$) a DDS generátor Agilent 33250A ($f_{\max} = 80 \text{ MHz}$).

Generátor HP 8647A nebylo možné použít, protože umožňuje regulaci frekvence až od 250 kHz, při které má voltmetr HP 34401A nepřijatelnou chybu měření. Použití DDS generátoru Agilent 33250A by bylo také silně nevhodné, jelikož má pouze 12-bitový D/A převodník, což na testování 12-bitového D/A převodníku z pochopitelných důvodů nestačí.

Bylo proto zvoleno jiné řešení. Byl navržen generátor trojúhelníkového průběhu, který byl postaven na kontaktním poli velice blízko testovanému A/D převodníku tak, aby nedocházelo k jeho zkreslení vlivem parazitních impedancí. Zapojení generátoru je na obrázku 9.6. Analogový generátor zajistí spojitost testovacího signálu a navíc zjednoduší vyhodnocení nasbíraných dat. Výstup generátoru je navržen tak, aby přesahoval rozsah převodníku v povolených mezích a v jeho rozsahu byly striktně lineární části tohoto signálu.



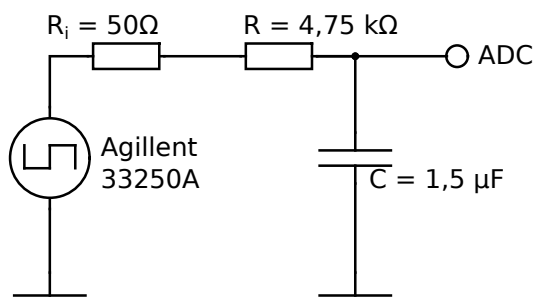
Obrázek 9.6: Generátor trojúhelníkového průběhu použitý pro testování A/D převodníku metodou histogramu

Po zapojení tohoto generátoru byla zkoumána linearita jeho výstupu a bohužel bylo zjištěno, že nelinearita je značná, dalece přesahující LSB měřeného převodníku. Tento problém by byl řešitelný, kdyby šlo nelinearitu přesně popsat a následně kompenzovat. Tomu bránily dva důvody:

- Osciloskop, kterým byla linearita vyšetřována (Tektronix TDS 3034B) umožňuje odečít hodnot v potřebném rozsahu (3 V) pouze v krocích velikosti 10 mV (cca 13 LSB).
- Nelinearita neodpovídala tvaru předpokládanému u tohoto zapojení, tedy exponenciálně.

Toto řešení tedy bylo zavrženo, ale po provedení měření následující metodou bylo zjištěno, že kondenzátor použitý v integrátoru generátoru byl vyroben z nevhodného dielektrika pro tento typ aplikace – keramiky. Při použití svitkového kondenzátoru s plastovým dielektrikem by toto řešení mohlo být vhodnou alternativou k metodě popsané v následujících odstavcích, která vyžadovala velice náročnou teoretickou přípravu.

Řešení, které bylo nakonec navrženo, se opírá o přechodný děj na RC članku, který je matematicky dobře popsatelný a jedná se o tolik jednoduchý systém, že existuje jen málo vlivů, které mohou ovlivnit jeho chování. Použité zapojení je na obrázku 9.7. Kondenzátor zde byl zvolen svitkový, vhodný do lineárních aplikací, jednoduchým testem měřením přechodového děje bylo ověřeno, že je kapacita tohoto kondenzátoru konstantní v celém rozmezí použitých napětí. Toto se nedá říci o keramickém kondenzátoru, jehož rozdíl kapacity činil při napětích 0 V a 3 V rozdíl více než 10 %.



Obrázek 9.7: Schéma pro měření nelinearit A/D převodníků

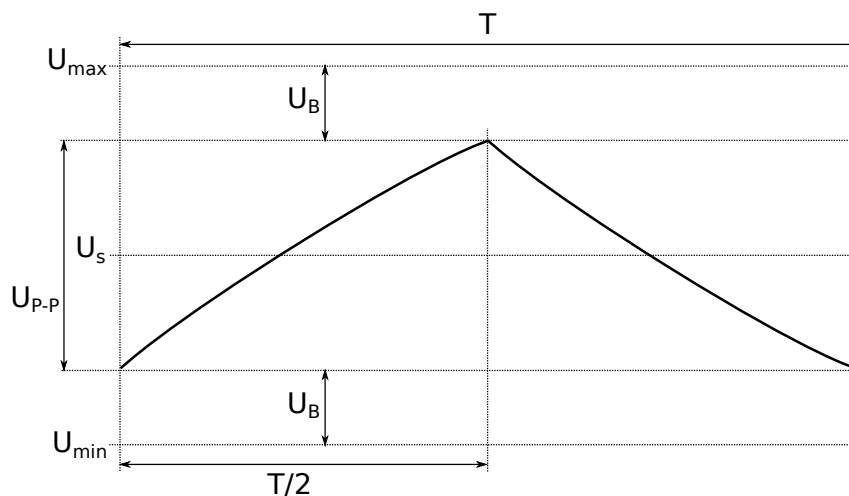
Průběh napětí na kondenzátoru bude mít v ustáleném stavu podobu zachycenou na obrázku 9.8. Tento signál lze matematicky popsat vztahem

$$u(t) = U_{max} - (U_{max} - U_{min} - U_B)e^{-\frac{t}{\tau}}, 0 < t < \frac{T}{2}, \quad (9.5)$$

$$u(t) = U_{min} + (U_{max} - U_{min} - U_B)e^{-\frac{(t-\frac{T}{2})}{\tau}}, \frac{T}{2} < t < T.$$

Dosažením v první rovnici $u(t) = U_{max} - U_B$ a $t = \frac{T}{2}$ získáme rovnici pro maximum průběhu, ze které lze jednoduše vyjádřit

$$U_B = \frac{(U_{max} - U_{min})e^{-\frac{T}{2\tau}}}{1 + e^{-\frac{T}{2\tau}}}. \quad (9.6)$$



Obrázek 9.8: Grafické znázornění modelu signálu na výstupu RC článku buzeném obdélníkovým signálem

Aby mohla být změřena převodní charakteristika A/D převodníku s jeho linearitami metodou

histogramu, je nejprve nutné vyjádřit teoreticky hustotu pravděpodobnosti s jakou se budou vyskytovat vzorky s danými hodnotami na základě testovacího signálu.

Z přirozené úvahy vyplývá, že hustota pravděpodobnosti pro kterékoliv napětí je úměrná derivaci inverzní funkce k $u(t)$ v tomto bodě. Inverzní funkce k $u(t)$ je $t(u)$. Vysvětleno běžnou řečí, velikost derivace funkce $t(u)$ vyjadřuje, jak velké rozmezí napětí daného průběhu odpovídá určitému malému rozmezí času. Jestliže je v některé oblasti signál strmý, trvá velice krátkou dobu než překoná rozdíl jednoho kvanta A/D převodníku a tím pádem je pravděpodobnost vzorku v tomto kvantu malá, v inverzní funkci se toto projeví jako malá změna času při velké změně napětí, tedy jako malá derivace. To přesně odpovídá histogramu, který vyjadřuje pro každé kvantum relativní délku trvání v periodě (relativní vůči délce periody).

Jelikož budící signál není funkce prostá, nelze spočítat celou její inverzi. Lze však spočítat inverzní funkce pro obě půlperiody vyjádřením t z funkcí $u(t)$, čímž se získají dvě inverzní funkce pro obě půlperiody

$$\begin{aligned} t_{1\frac{T}{2}}(u) &= -\tau \ln \frac{U_{max} - u}{U_{max} - U_{min} - U_B}, \\ t_{2\frac{T}{2}}(u) &= -\tau \ln \frac{u - U_{min}}{U_{max} - U_{min} - U_B} + \frac{T}{2}, \end{aligned} \quad (9.7)$$

pro které se dále spočtou derivace, jež jsou úměrné hustotě pravděpodobnosti napětí. Výsledné derivace jsou

$$\begin{aligned} t'_{1\frac{T}{2}}(u) &= -\tau \frac{1}{U_{max} - u}, \\ t'_{2\frac{T}{2}}(u) &= -\tau \frac{1}{u - U_{min}}. \end{aligned} \quad (9.8)$$

Hustotu pravděpodobnosti je nutno spočítat pro rozsah napětí, která se vyšetřují, tedy od offsetu do reference, tedy hodnot, které byly změřeny v předchozí kapitole. Hustota pravděpodobnosti vyjádřená tímto způsobem nezáleží na znaménku ale pouze velikosti vypočtených derivací. Výsledný tvar funkce hustoty pravděpodobnosti $P(u)$ se tedy získá součtem absolutních hodnot těchto derivací, tedy

$$XP(u) = \begin{cases} 0 & , u < U_{off} \\ \left| t'_{1\frac{T}{2}}(u) \right| + \left| t'_{2\frac{T}{2}}(u) \right| & , U_{off} < u < U_{ref} \\ 0 & , U_{ref} < u \end{cases} \quad (9.9)$$

kde X vyjadřuje multiplikativní chybu. Jelikož funkce hustoty pravděpodobnosti musí mít plochu rovnou jedné, je nutné vypočítat X . Jelikož $XP(u)$ je součtem dvou funkcí, jeho plocha je tedy rovněž součtem ploch těchto dvou funkcí, na základě této skutečnosti lze stanovit

$$X = \int_{U_{off}}^{U_{ref}} \left| t'_{1\frac{T}{2}}(u) \right| du + \int_{U_{off}}^{U_{ref}} \left| t'_{2\frac{T}{2}}(u) \right| du, \quad (9.10)$$

kde obě integrované funkce jsou derivacemi už známých funkcí, stačí tedy do těchto primitivních funkcí dosadit integrační meze a spočítat určité integrály, čímž získáme výslednou multiplikativní chybu

$$X = \left| t_{1\frac{T}{2}}(U_{ref}) - t_{1\frac{T}{2}}(U_{off}) \right| + \left| t_{2\frac{T}{2}}(U_{ref}) - t_{2\frac{T}{2}}(U_{off}) \right|. \quad (9.11)$$

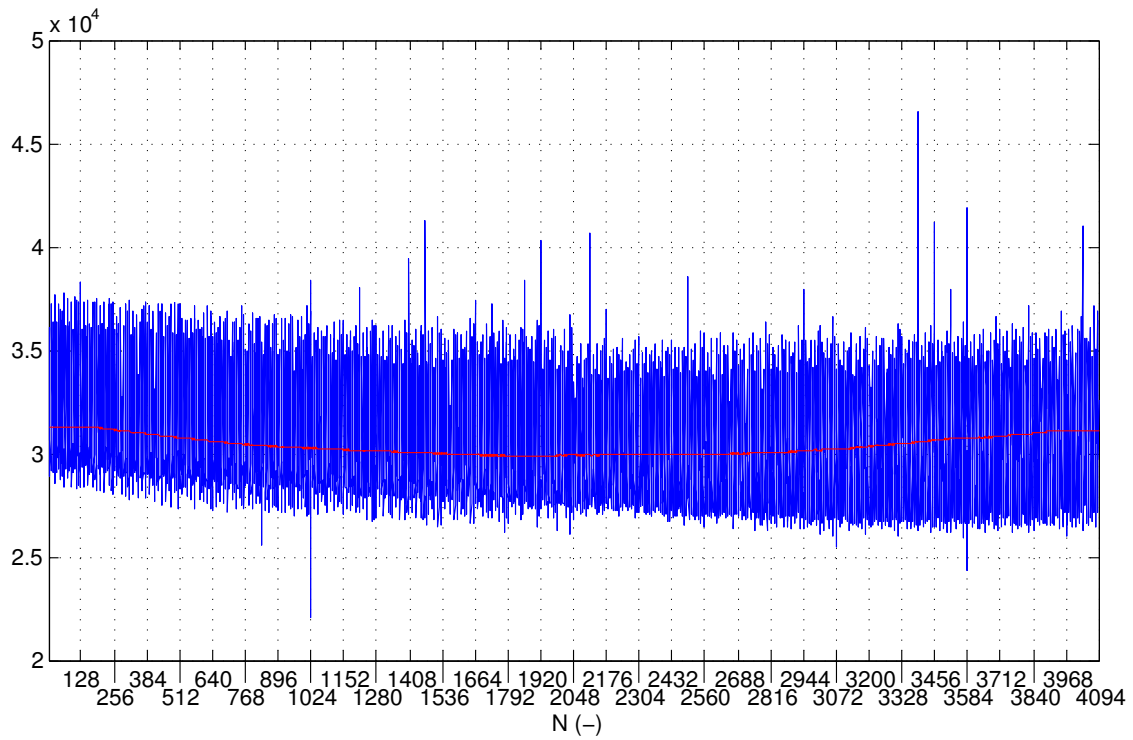
Výslednou funkci hustoty pravděpodobnosti získáme vydělením rovnice 9.9 vypočtenou multiplikativní chybou X .

9.2.3 Vlastní měření nelinearit A/D převodníku

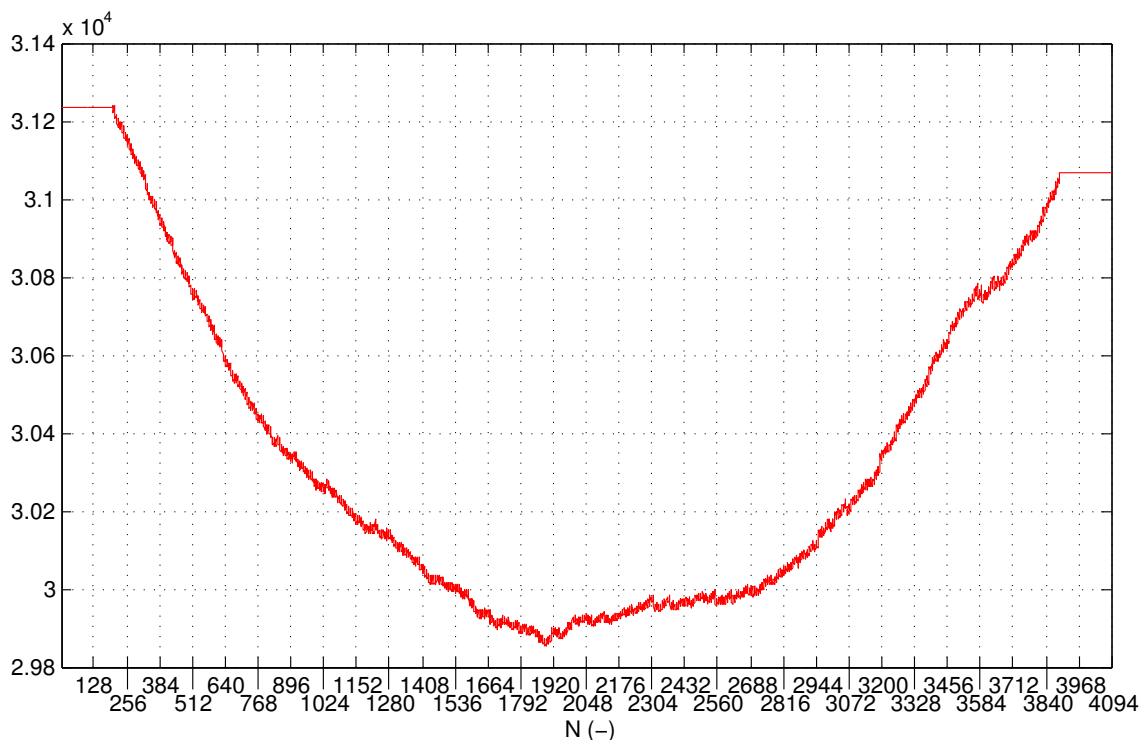
Před vlastním měřením byl pokusně zjištěn rozsah možností nastavení výstupního napětí generátoru Agilent 33250A, který byl v tomto měření použit jako generátor obdélníkového signálu (viz obrázek 9.7). Jelikož RC článek představuje systém s jednoznačným chováním, je známo, že v ustáleném stavu na jeho výstupu bude střední hodnota rovna průměru obou úrovní obdélníkového signálu a jeho rozkmit bude symetrický na obě strany od střední hodnoty.

Pro testování A/D převodníků metodou histogramu je vhodné přivést na jejich vstup napětí, které přesahuje na obou stran v bezpečných mezích jejich rozsah, což v případě STM32F3 představuje napětí 0 až 3 V. Na generátoru se tedy nastavovalo napětí horní a dolní úrovně na $1,5 \pm (1,5 \text{ V} + U_B)$, kde U_B je definováno obrázkem 9.8 a čím je vyšší, tím lineárnější je průběh na vstupu A/D převodníku a tím musí být vyšší frekvence vstupního obdélníkového signálu, aby výstup nepřekročil požadované meze. Konkrétní nastavené hodnoty pro minimální a maximální napětí obdélníku bylo nastaveno na -5 V , resp 8 V (pozn.: na generátoru 33250A je napětí nastavováno pro 50Ω zátěž, zadávají se tedy poloviční hodnoty, tedy $-2,5 \text{ V}$ a 4 V). Skutečné hodnoty byly změřeny voltmetrem HP 34401A na výstupu RC článku při frekvenci nastavené na $0,1 \text{ Hz}$ ($\ll \frac{1}{T}$), frekvence obdélníku pro měření byla nastavena podle osciloskopu aby se dosáhlo rozkmitu specifikovaného výše v tomto odstavci. Výstup RC článku byl následně připojen na vstup měřeného A/D převodníku a byla změřena efektivní hodnota střídavého signálu (pomocí voltmetru 34401A).

Následně bylo naměřeno asi 138 milionů vzorků tohoto signálu, z nichž asi 124 milionů mělo hodnotu 1 až 4094, tedy tolik hodnot bylo použito pro sestavení histogramu. Histogram naměřených hodnot pro všechna kvanta kromě prvního a posledního je zachycen na obázku 9.9. Za povšimnutí stojí červeně zobrazená křivka ukazující trend, kterým se data pohybují. V detailu je zobrazen na obrázku 9.10.



Obrázek 9.9: Histogram dat získaných měřením testovacího průběhu při měření nelinearit A/D převodníku procesoru STM32F3, červeně zobrazen klouzavý průměr zvýrazňující trend dat



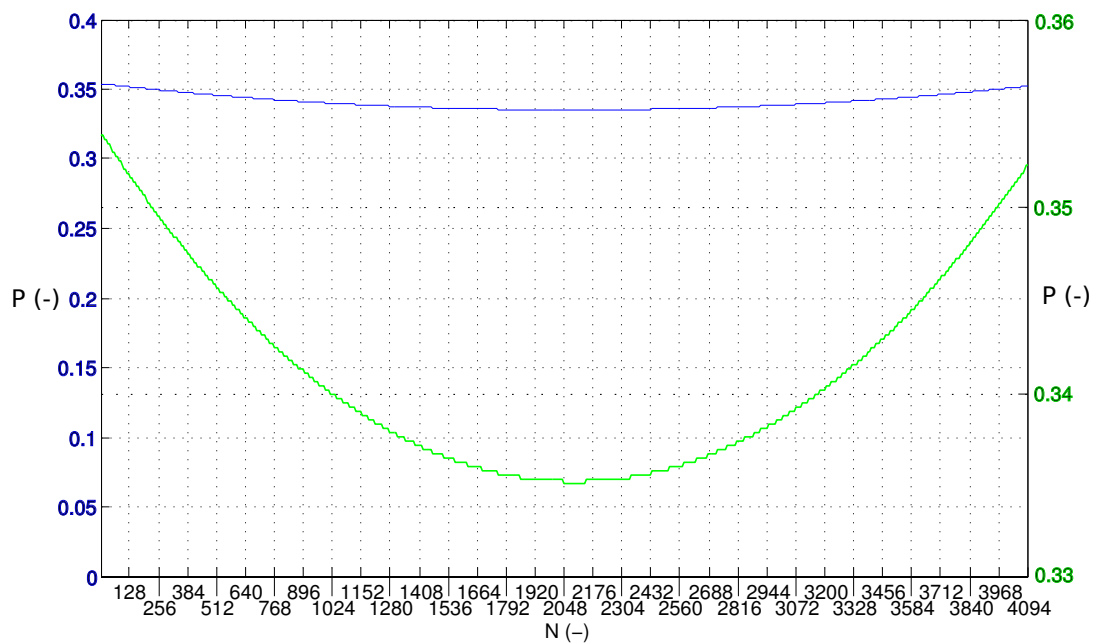
Obrázek 9.10: Detailní zobrazení klouzavého průměru (délka průměrování 401) histogramu z obrázku 9.9

Jak je z obrázku 9.10 patrné, histogram vykazuje výrazné prohnutí. Jistá velice výrazná a pravidelná nelinearita histogramu byla očekávána, z toho důvodu byla provedena důkladná analýza testovacího signálu. Jednak byly změřeny hodnoty napětí po odeznění přechodového děje na RC članku, čímž se zohlednil i případný svodový odpor kondenzátoru. Dále byla změřena efektivní hodnota testovacího průběhu. Tyto hodnoty posloužily k přesnému naladění teoretického modelu vytvořeného v Matlabu. V něm byl pomocí rovnic 9.5 a 9.6 vytvořen teoretický průběh, ve kterém jsou hodnoty U_{max} a U_{min} a T velice přesně známy. Hodnota τ byla nastavena na součin jmenovitých hodnot použitého odporu a kondenzátoru v RC članku. Dále byla vyhodnocena efektivní hodnota střídavé složky teoretického průběhu a porovnána s efektivní hodnotou změřenou voltmetrem přímo při měření (toto je možné jen díky tomu, že voltmetr HP 34401A měří skutečnou efektivní hodnotu – true RMS). Jediná nepřesně známá veličina v teoretickém modelu, časová konstanta τ byla ručně nastavována tak, aby došlo ke shodě efektivní hodnoty teoretického průběhu se skutečným signálem.

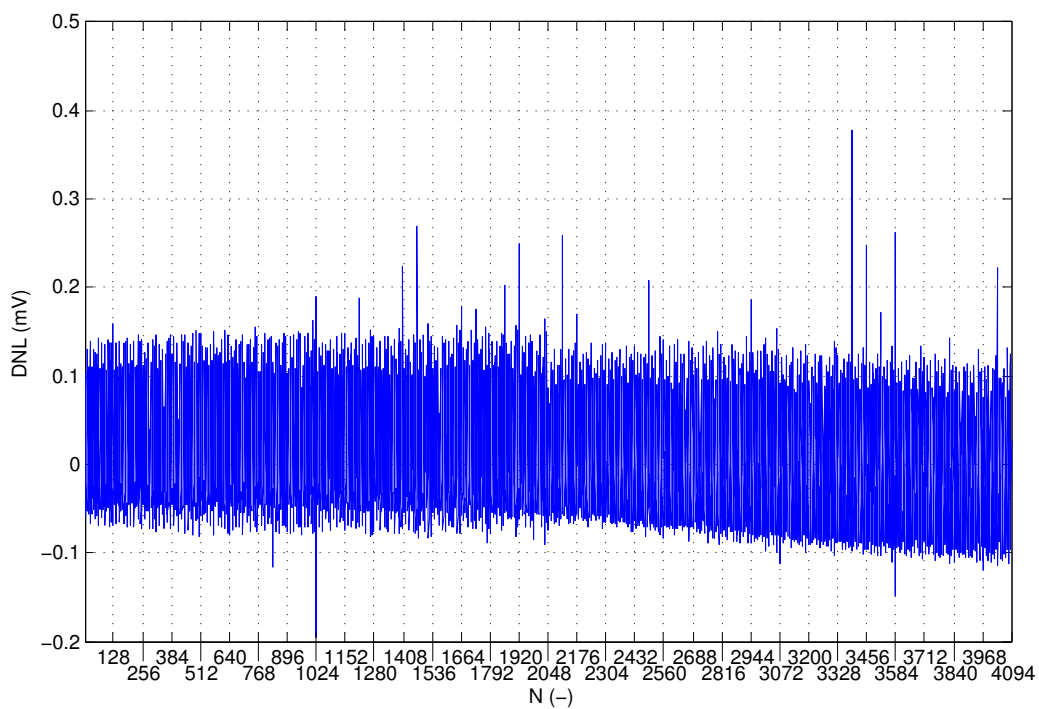
Pomocí parametrů přesně namodelovaného testovacího signálu byla vypočtena hustota pravděpodobnosti výskytu vzorku v jednotlivých kvantech, která je zobrazena na obrázku 9.11. Podobnost s klouzavým průměrem histogramu naměřených dat na obrázku 9.10 je zřejmá. V obou průbězích je patrný charakter paraboly s minimem posunutým mírně za střed rozsahu převodníku. To, zda se opravdu jedná o parabolu, nebylo prokázáno. Teoretickou hustotou pravděpodobnosti pro každý vzorek se vydělí každá čára naměřeného histogramu, čímž se provede jeho korekce. Po vydělení histogramu jeho sumou a vynásobením číslem 4095 se histogram změní v graf šířek slov vyjádřených v LSB převod-

níku. Odečtením jedné od všech hodnot a vynásobením velikosti LSB se získá diferenciální nelinearita převodníku (DNL) vyobrazená na obrázku 9.12.

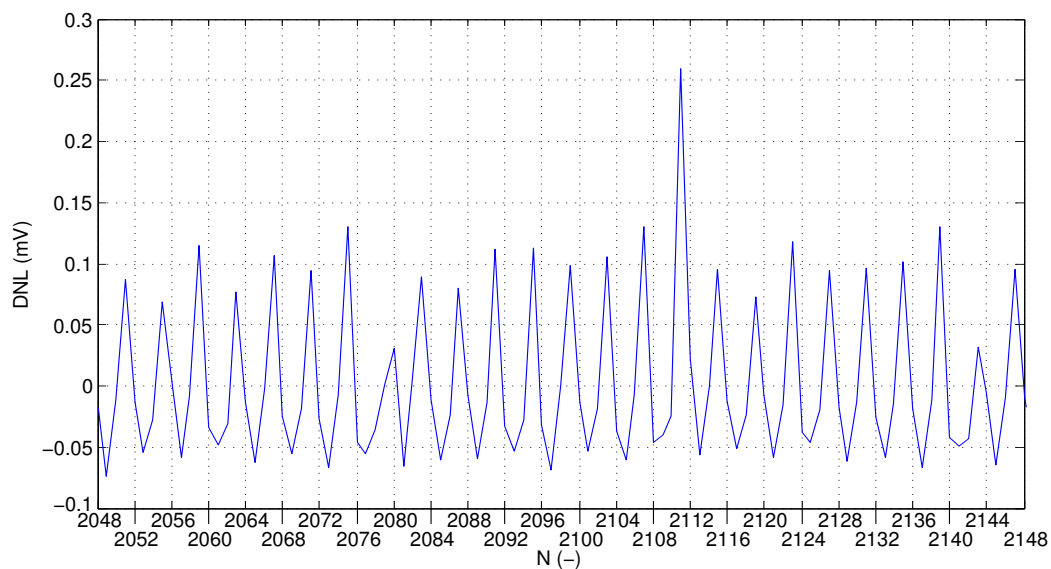
Za povšimnutí stojí detail diferenciální nelinearity na obrázku 9.13, který ukazuje zajímavý jev, kdy je každé čtvrté kódové slovo výrazně širší (běžně o 20 %) než slova okolní. Toto chování poukazuje výraznou nelinearitu výstupu spodních 2 bitů D/A převodníku použitého v architektuře SAR A/D převodníku, jelikož spodní dva bity se opakují vždy po čtyřech kódových slovech.



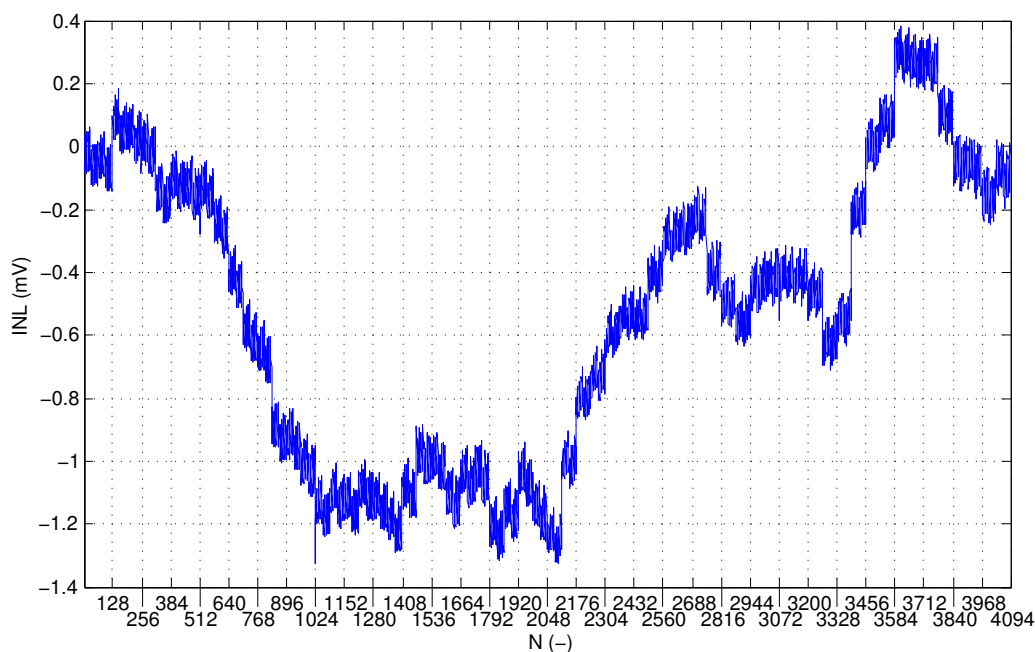
Obrázek 9.11: Teoretická hustota pravděpodobnosti výskytu vzorku v kvantu N testovacího signálu pro testování A/D převodníků (normalizována na Volty, zde osa v kvantech)



Obrázek 9.12: Diferenciální nelinearita A/D převodníku procesoru STM32F303 (absolutní maximum 0,377 mV – 0,53 LSB – mezi v přechodu mezi kódovými slovy 3391 a 3392)



Obrázek 9.13: Detail diferenciální nelinearity A/D převodníku procesoru STM32F303



Obrázek 9.14: Integrovaná nelinearita A/D převodníku procesoru STM32F303 (absolutní maximum 1,33 mV – 1,86 LSB – kódovém slově 1023)

9.3 Měření dynamických vlastností A/D převodníků

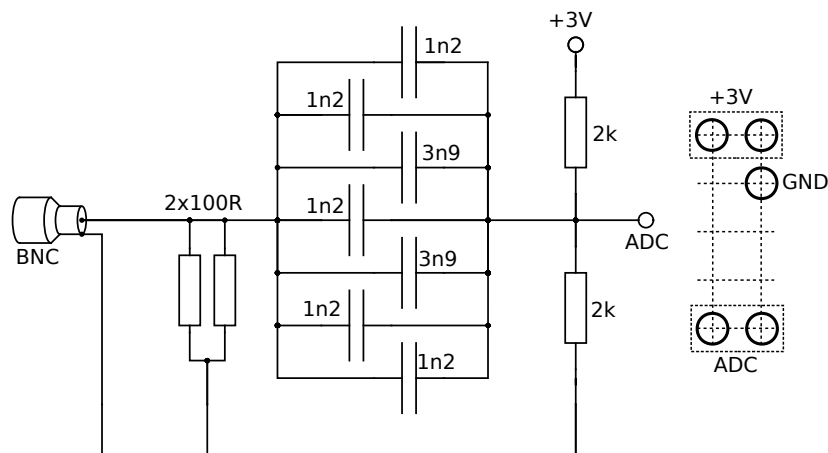
V této části dokumentu jsou vyšetřovány dynamické vlastnosti A/D převodníků, jmenovitě přenosová frekvenční charakteristika a z ní odečtená šířka pásma a dále jitter, tedy nestabilita hodinového signálu převodníku vedoucí ke zkreslení signálů.

9.3.1 Přenosová frekvenční charakteristika A/D převodníků

Měření frekvenční charakteristiky bylo provedeno za pomoci sinusového generátoru HP 8647A. Tento generátor má symetrický výstup bez možnosti offsetování, musel být tedy vytvořen posouvací obvod navržený podle DP pana Dobruského^[11] (viz obrázek 9.15), který na výstup generátoru superponuje stejnosměrnou složku 1,5 V, tedy napětí v polovině rozsahu měřeného A/D převodníku. Posouvací obvod byl postaven ze zakončovacích odporů ($2 \times 100\Omega$), baterie oddělovacích keramických kondenzátorů (z důvodu minimalizace parazitních indukčností přívodů) a děliče tvořeného dvěma odpory 2k. Oddělovací kapacita je přítomna z toho důvodu, aby nebyl dělič pro nastavení offsetu zatěžován zakončovacím odporem, resp. vnitřním odporem generátoru. Dělič však tvoří spolu s oddělovacím kondenzátorem derivační článek, tedy horní propust se zlomovou frekvencí

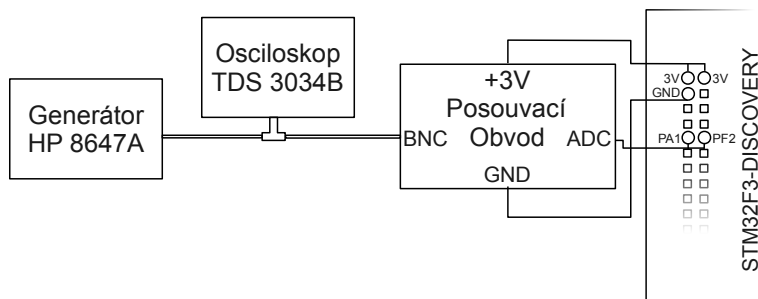
$$f_0 = \frac{1}{2\pi RC}, \quad (9.12)$$

kde R odpor děliče transformovaný pomocí Théveninova teorému na hodnotu $1\text{ k}\Omega$ a C je součet všech kapacit v baterii, tedy $13,8\text{ nF}$. Dosazením dostaneme zlomovou frekvenci horní propusti $11,5\text{ kHz}$. Vzhledem k tomu, že se frekvenční charakteristika měřila od frekvence $281,2\text{ kHz}$, lze prohlásit, že se útlum tohoto článku neuplatňuje.



Obrázek 9.15: Posouvací obvod pro přidání stejnosměrné složky $1,5\text{ V}$ ke střídavému signálu podle [11], vpravo pohled do dutinek určených k nasazení na STM32F3-Discovery

Schéma zapojení pro toto měření je na obrázku 9.16. Posouvač na obrázku 9.15 byl zkonstruován pro nasazování na STM32F3-Discovery zespoda, na schématu je kit zobrazen seshora, proto je rozmístění dutinek posouvače zrcadleně. Pin PA1 byl nastaven jako vstup pro ADC1, pin PF2 byl nepoužitý.



Obrázek 9.16: Schéma propojení generátoru, osciloskopu, posouvacího obvodu a STM32F3-Discovery pro měření frekvenční charakteristiky jeho A/D převodníku

Měření bylo provedeno pro frekvence od $281,19\text{ kHz}$ do $605,864\text{ MHz}$ v rozlišení 9 bodů na dekádu. Nejprve byla frekvence k -tého bodu nahrubo nastavena na hodnotu

$$f(k) = f_s \left\| 10^{\frac{k}{9}} \right\|, \quad (9.13)$$

kde $|||$ značí zaokrouhlení a f_S je vzorkovací frekvence. Tyto frekvence byly zvoleny tak, aby každá nově nastavená frekvence byla poblíž násobku vzorkovací frekvence. Z teorie DFT je známo, že frekvence reprodukováného je rovna rozdílu nejbližšího násobku vzorkovací frekvence a frekvence měřeného signálu. Cílem je zaznamenat průběh, kde bude dobře na první pohled vidět tvar rekonstruovaného signálu. Důvodem je jednak následná analýza jitteru oscilátoru A/D převodníku, dále pak kontrola, zda měření probíhá korektně.

Pro každou frekvenci byla zaznamenána frekvence signálu nastavená na generátoru, rozkmit signálu odečtený z osciloskopu a sekvence 8192 vzorků pořízených testovaným A/D převodníkem. Vzorkovací frekvence byla zvolena konstantní 281,25 kHz (délka doby odběru vzorku 2,17 μ s oproti minimální požadované 94 ns při vstupním odporu 1 k Ω , viz [3], tabulka 34).

Pro každou sekvenci dat byl spočítán poměr

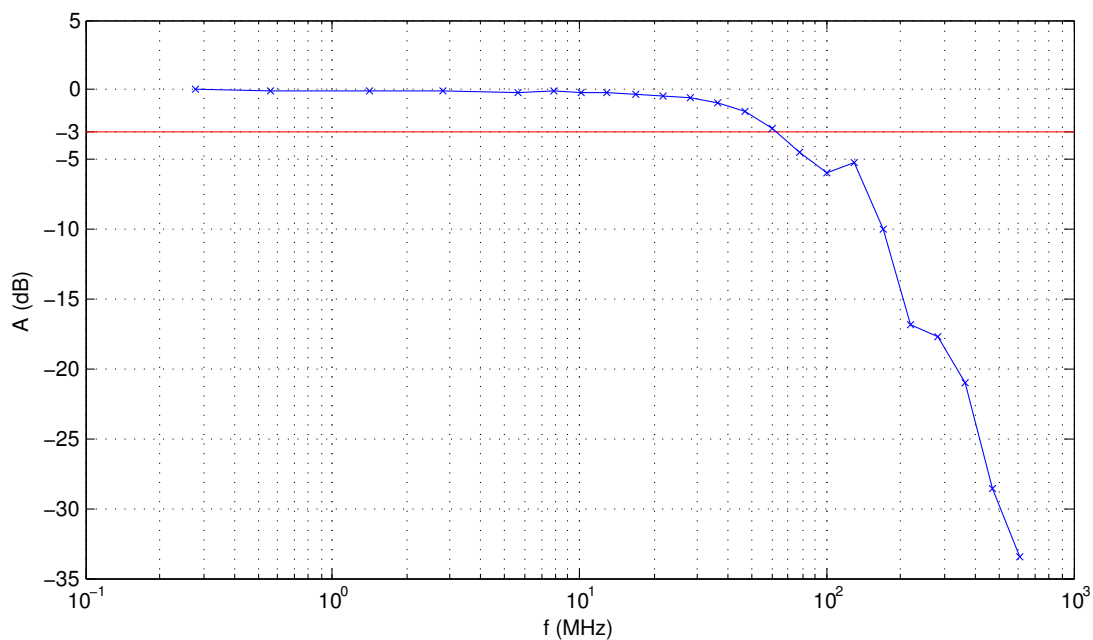
$$A(k) = \frac{\Delta n_{max}(k)}{U_{in,p-p}(k)}, \quad (9.14)$$

kde k je pořadí nastavené frekvence $\Delta n_{max}(k)$ je rozdíl největší a nejmenší hodnoty vzorku v pořízeném průběhu a $U_{in,p-p}$ je rozkmit napětí naměřený na osciloskopu pro k -tou nastavenou frekvenci signálu.

Výsledná frekvenční charakteristika (obrázek 9.17) byla poté spočítána jako

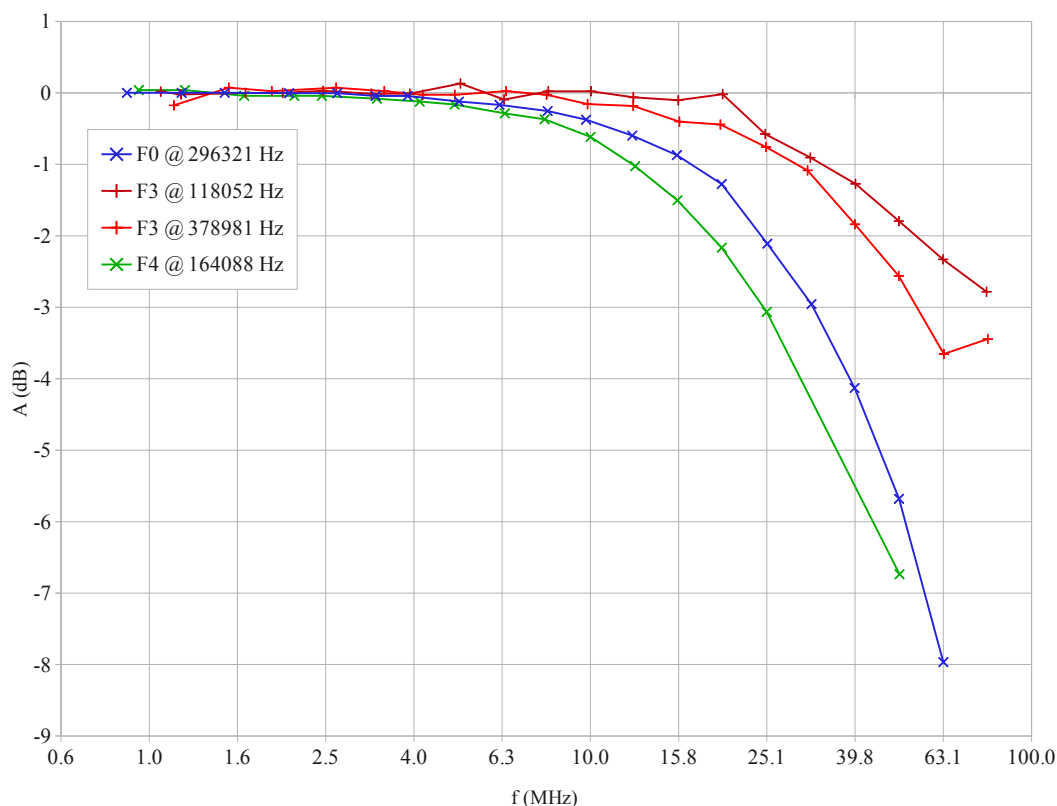
$$A_{dB}(k) = 20 \log \frac{A(k)}{A(0)}. \quad (9.15)$$

Z pohledu do charakteristiky lze prohlásit, že měření probíhalo korektně a definovaně až do frekvence okolo 100 MHz, byla tedy úspěšně změřena šířka pásma A/D převodníku (přesněji řečeno jeho analogové části včetně parazitních impedancí na DPS STM32F3-Discovery). Ta byla stanovena metodou lineární interpolace mezi odměry na frekvencích 60,47 MHz (-2,77 dB) a 78,19 MHz (-4,45 dB) na 62,91 MHz.



Obrázek 9.17: Frekvenční charakteristika A/D převodníku mikrokontroléru STM32F303 při vzorkovací frekvenci 281,25 kHz

Ještě před vytvořením finální verze firmwaru pro STM32F303 byly změřeny frekvenční charakteristiky tohoto mikrokontroléru při vzorkovacích frekvencích 118 kHz a 379 kHz, dále pak mikrokontrolérů STM32F051 při vzorkovací frekvenci 296 kHz a STM32F407 při 164 kHz. Měření bylo realizováno generátorem Agilent 33250A, který byl přes digitální osciloskop Agilent 54622 připojen přímo na headery discovery kitů těchto procesorů. Generátor umožňuje offsetování výstupního průběhu, proto se toto měření obešlo bez posouvacího obvodu. Zakončení bylo provedeno ze tří paralelně spojených rezistorů jmenovité hodnoty $50\ \Omega$ zapojených u pinů headerů vedoucích na zem a vstup A/D převodníku procesoru. Toto měření, jehož výsledky jsou na obrázku 9.18, je uvedeno pro letmé porovnání šířek pásma A/D převodníků obsažených ve zmíněných procesorech. Dále pak dokazuje závislost šířky pásma na vzorkovací frekvenci.

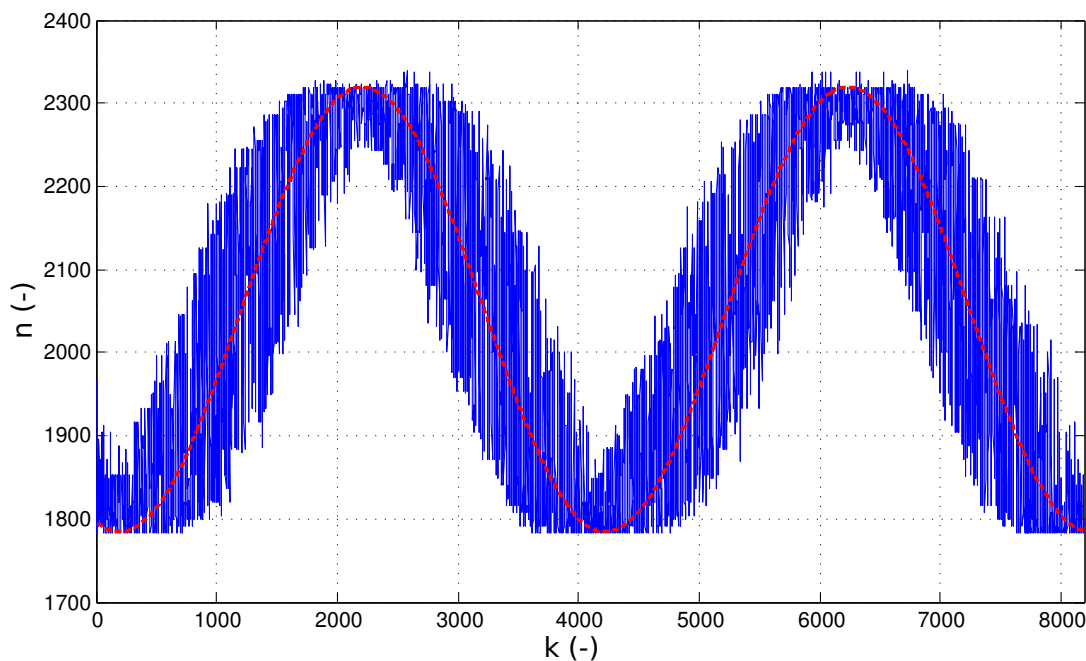


Obrázek 9.18: Frekvenční charakteristiky A/D převodníků mikrokontrolérů STM32F051, STM32F303 a STM32F407 při různých vzorkovacích frekvencích

9.3.2 Vyhodnocení jitteru A/D převodníků

Jitter označuje kolísání hodinového signálu, které způsobuje nekoherenci vzorkování, která omezuje použití A/D převodníku při stroboskopickém vzorkování. Projevuje se jako šum, který degraduje vzorkovaný průběh, zejména v oblastech největších změn vstupního signálu, projevuje se tak "roz-mazným" průběhu ve směru časové osy. Tím jej lze také odlišit od běžného šumu, který znehodnocuje celý průběh rovnoměrně. Na obrázku 9.20 jsou vyobrazeny některé z průběhů pořízených při měření frekvenční charakteristiky, které po polovinách dekad ukazují rostoucí vliv jitteru.

Jitter je definován jako maximální odchylka času mezi odebráním dvou následujících vzorků od času daného vzorkovací frekvencí, tedy $T = 1/f_S$. Ke stanovení jitteru A/D převodníku procesoru STM32F3 byla využita data vzorkovaná při měření frekvenční charakteristiky při frekvenci vstupního signálu 281,27 MHz. Tento průběh je zachycen na obrázku 9.19



Obrázek 9.19: Průběh využitý pro stanovení jitteru A/D převodníku s červeně vyznačenou ruční aproximací (frekvence vstupního signálu 281,27 MHz)

Aproximace

$$f_{approx.} = n_s + A \sin\left(\frac{2\pi k}{T} + \varphi\right) \quad (9.16)$$

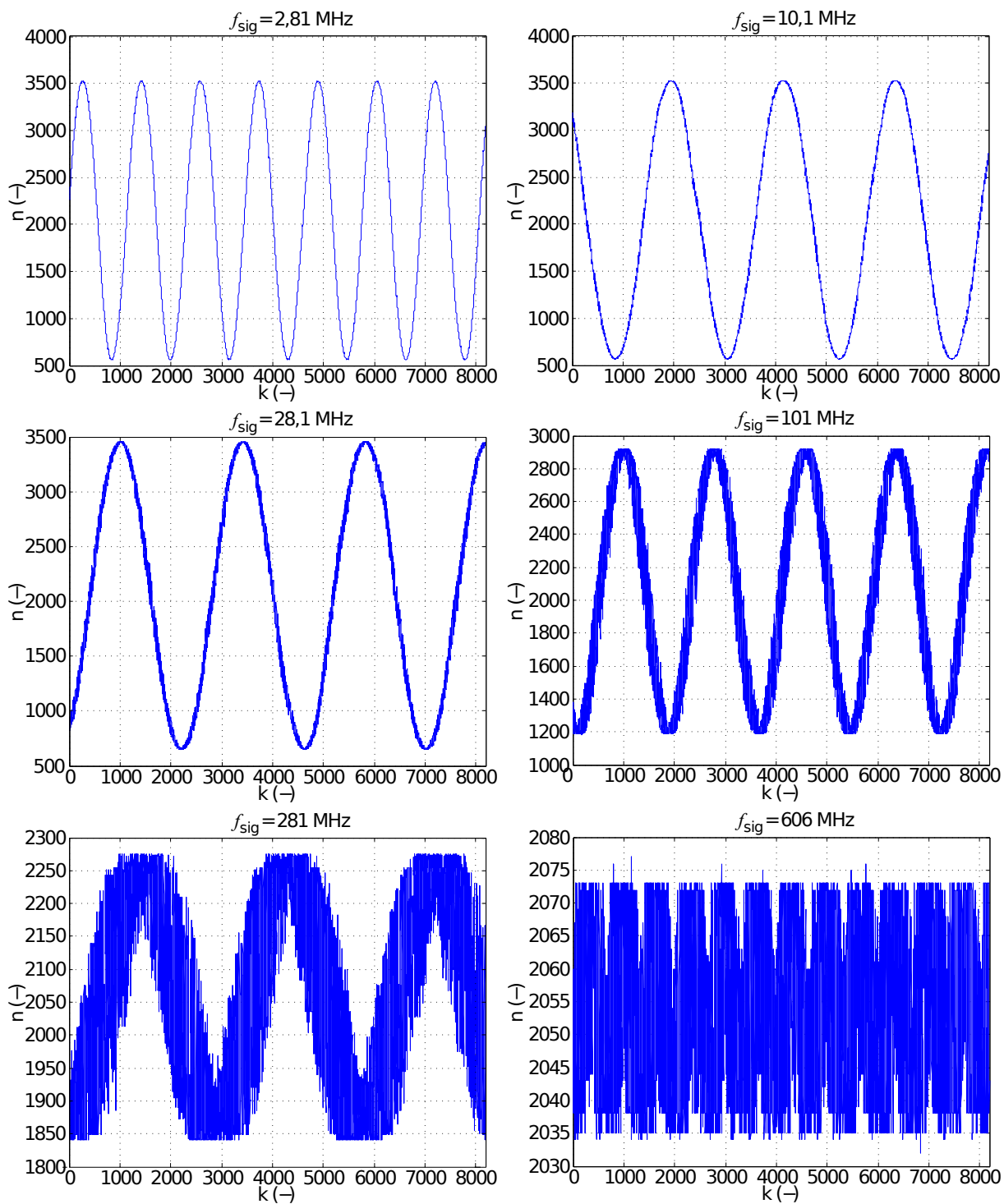
Jitter se určí tak, že se najdou dva po sobě jdoucí vzorky s maximálním rozdílem hodnot a pomocí sinusové aproximační funkce se jim přiřadí argument sinu odpovídající jejich funkční hodnotě (ilustrováno na obrázku 9.21) podle vztahu

$$\alpha_k = \arcsin \frac{n_k - n_s}{A}. \quad (9.17)$$

Odečtem vzorku s nižším pořadovým číslem od vzorku s vyšším se získá rozdíl argumentů $\Delta\alpha$, od kterého se dále odečte rozdíl argumentů ideálního převodníku $\Delta\alpha_{ideal}$, kterému by se rovnala hodnota $\Delta\alpha$ v případě, že převodník nemá jitter. Vzhledem k povaze funkce arcsin vyjde tento rozdíl opačně v případě, že se odečítá na klesající části sinusoidy (ošetřeno signem derivace aproximační funkce). Jitter se poté získá přepočtem na čas pomocí frekvence měřeného signálu. Tento odstavec je shrnut ve vzorci 9.18.

$$t_{jit} = \left| \frac{\text{sign}\left(\cos\left(\frac{2\pi k}{T} + \varphi\right)\right) \Delta\alpha - \Delta\alpha_{ideal}}{2\pi f_{sig}} \right|, \Delta\alpha_{ideal} = \frac{2\pi}{T}. \quad (9.18)$$

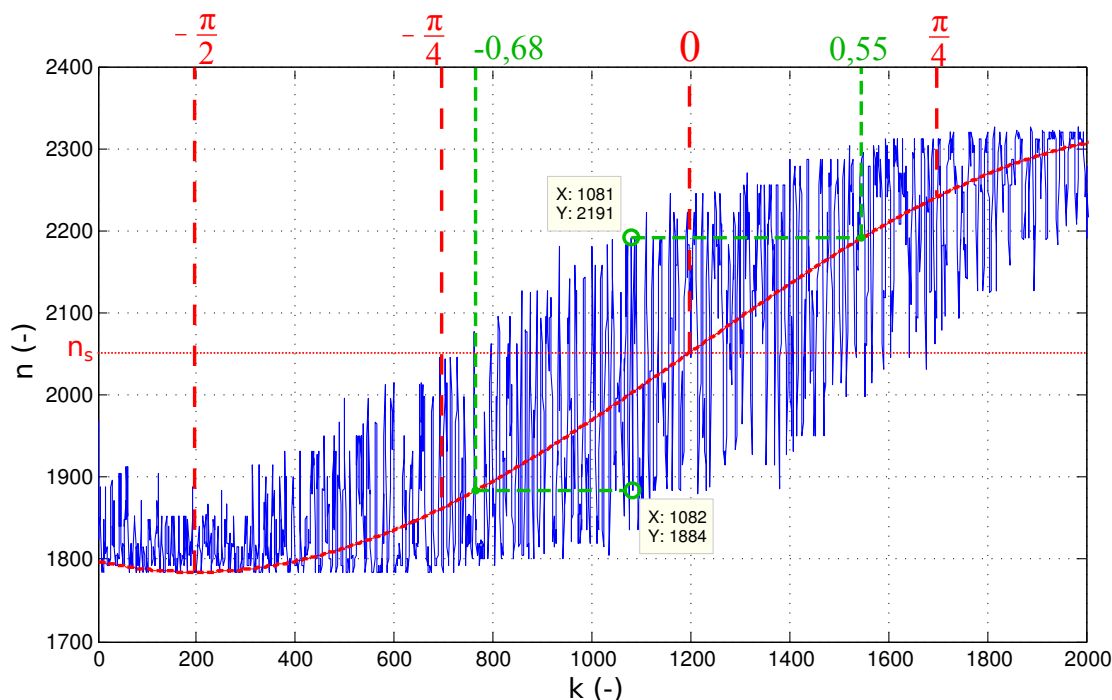
Popsaná metoda funguje nejlépe s frekvencí měřeného signálu nastavenou tak, že hodnota jitteru



Obrázek 9.20: Demonstrace vlivu jitteru na kvalitu rekonstrukce měřeného signálu

je zhruba čtvrtinou periody tohoto signálu. To se pozná tak, že se v malém okolí od maxima, resp. minima, interpolačního signálu právě nevyskytují žádné vzorky s hodnotou menší, resp. větší, než je střední hodnota průběhu. Při této frekvenci je citlivost měření jitteru vysoká a zároveň nemůže dojít k falešným měřením, tzn. že vzorek v oblasti kolem průchodů nulou lze na aproximační funkci promítnout jednoznačně.

To, že se k výpočtu použijí vzorky z oblastí největších změn a tedy nedojde k falešným měřením, není zapotřebí zajišťovat explicitně, jelikož největší difference mezi vzorky nastávají spolehlivě v těchto oblastech.



Obrázek 9.21: Detail signálu použitého pro analýzu jitteru s označenými body s největší vzájemnou diferencí, červeně je zobrazena aproximace signálu s významnými argumenty, zeleně odečet argumentu pomocí funkce arcsin

Jitter měřeného převodníku byl stanoven při frekvenci měřeného signálu 217,7062 MHz. V naměřených datech (detail na obrázku 9.21) byly nalezeny dva vzorky s nejvyšší diferencí (1081 a 1082) a jitter byl vyhodnocen podle výše uvedených vztahů následovně.

- Manuálně byla provedena aproximace funkcí

$$f_{approx.} = 2052 + 267,5 \sin\left(\frac{2\pi k}{4025} - 0,6\pi\right),$$

- byly vypočteny argumenty obou vzorků a jejich rozdíl

$$\alpha_{1081} = \arcsin \frac{2191 - 2052}{267,5},$$

$$\alpha_{1082} = \arcsin \frac{1884 - 2052}{267,5},$$

$$\Delta\alpha = \alpha_{1082} - \alpha_{1081} = -1,23 \text{ rad}$$

- a dosazením do rovnice 9.18 byl vypočten jitter

$$t_{jit} = \left| \frac{\text{sign} \left(\cos \left(\frac{2\pi 1081}{4025} - 0,6\pi \right) \right) (-1,23) - \frac{2\pi}{4025}}{2 \pi 217,7062 \cdot 10^6} \right| = 0,9 \text{ ns.}$$

Kontrolně byl vypočítán jitter z dat pořízených při měření signálu o frekvenci 101 MHz. Výsledek byl 0,86 ns, čímž se ověřila správnost uvedených výpočtů.

10 D/A převodníky, jejich vlastnosti a měření

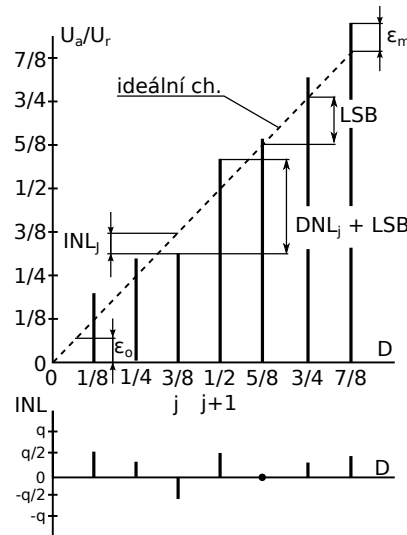
V této kapitole je uvedena základní teorie týkající se vlastností D/A převodníků, v následujících dvou podkapitolách jsou popsány metody jejich měření a prezentovány výsledky. Měření bylo realizováno pomocí firmwaru a PC ovladače, které jsou v této práci popisovány. Z toho plyne skutečnost, že je D/A převodník nakonfigurován na 12-bitový výstup a je deaktivovaný výstupní posilovač, při jehož použití bylo pozorováno výrazné zkreslení výstupního signálu.

10.1 D/A převodníky a jejich vlastnosti

Jak se píše v monografii *Elektronické obvody pro měřicí techniku* (Vedral, Fischer^[7]), statické parametry D/A převodníků jsou dány offsetem (chybou nuly), chybou zesílení a integrální a diferenciální nelinearitou. Na rozdíl od měření statických vlastností A/D převodníku se zde jedná o principiálně velice jednoduchou záležitost. Jediné potřebné vybavení pro provedení tohoto měření je dostatečně přesný voltmetr, pomocí kterého se zaznamenají úrovně pro všechna kódová slova. Dostatečně přesným voltmetrem rozumíme přístroj s rozlišením i maximální chybou výrazně pod úrovní LSB převodníku. Vzhledem k tomu, že se jedná o statické měření, není zapotřebí podnikat žádná zvláštní opatření (např. co se délek vodičů a impedančního přizpůsobení týče). Všechny statické parametry se stanoví z naměřených výstupních hodnot pro všechna kódová slova.

Statické parametry nejlépe objasňuje obrázek 10.1. Z něj je patrné následující:

- offset zjistíme jako výstupní napětí převodníku při nulovém aktivním kódovém slově
- INL je dáno jako rozdíl mezi naměřenou a jmenovitou výstupní hodnotou
- DNL se stanoví odečtením úrovně LSB od rozdílu výstupních napětí pro dvě po sobě následující kódová slova
- chyba zesílení odpovídá relativní odchylce výstupního napětí při maximálním výstupu od jmenovité hodnoty.



Obrázek 10.1: Obrázek převodní charakteristiky 3-bitového D/A převodníku s kótami pro stanovení INL, DNL, offsetu a chyby zesílení (převzato z [7])

Literatura [7] dále jmenuje významné dynamické parametry, a to za prvé dobu ustálení na konečnou hodnotu s chybou $\pm 0,5$ LSB (v monografii se uvádí jmenovitá hodnota, to by ale podle všeho u kódového slova s $INL > 0,5$ LSB znamenalo, že se neustálí nikdy). Datasheet^[2] k mikrokontroléru STM32F3 používaném v této práci definuje dobu ustálení jako dobu, za kterou přeběhne D/A převodník z minimální na maximální hodnotu ± 1 LSB (konkrétně udává maximum $4 \mu s$). Za druhé se jedná o zákmity při přepínání velkého množství proudových zdrojů v D/A převodnících s rezistorovými sítěmi nebo jinými proudovými zdroji. Jak literatura uvádí, k tomuto jevu dochází zejména při přechodu přes polovinu a první a třetí čtvrtinu referenčního napětí. Důvod již byl naznačen. Při přechodu na polovinu referenčního napětí dochází u osmibitového převodníku k přechodu z kódového slova 01111111 na 10000000, tedy se v jednom okamžiku vypíná zdroj přispívající napětím $\frac{U_{ref}}{2} - LSB$ a zapíná zdroj přispívající napětím $\frac{U_{ref}}{2}$. Při extrémní prodlevě mezi oběma ději tedy může být na výstupu převodníku plné napětí nebo naopak nulové napětí (neuvažujeme-li offset).

Předpokládá se, že jsou oba zmíněné dynamické jevy velice rychlé a zajímavá napětí při nich pozorovaná dosahují velmi malých hodnot. Je tedy nutné při jejich pozorování využít osciloskop s velkou šířkou pásma a vzorkovací frekvencí. Výrobce mikrokontroléru STM32F3 udává v příslušném datasheetu^[2] maximální obnovovací frekvenci D/A převodníku $1 MHz$, šířka a velikost zákmity při přepínání mezi úrovněmi není zmíněna, nelze tedy přesně určit požadavky na vlastnosti osciloskopu a bude nutné pokusem s dostupným přístrojem zjistit, zda je pro toto měření vhodný.

10.2 Statické parametry

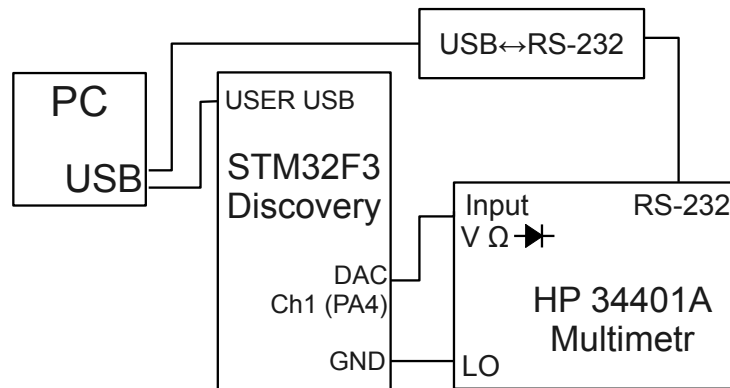
Popis měření

Schéma měření statických vlastností D/A převodníků je zachyceno na obrázku 10.5. K měření bylo využito voltmetru Hewlett-Packard 34401A, který byl připojen k PC pomocí USB \leftrightarrow RS-232 převodníku, discovery kit s STM32F3 byl rovněž připojen k PC pomocí USB. Měřicí svorky voltmetru byly připojeny na výstup D/A převodníku a na zem discovery kitu. RS-232 port voltmetru byl nakonfigurován aby odpovídal nastavení virtuálního portu PC (viz další odstavec) a adresa přístroje byla nastavena na 31 ("Talk only").

Po opakovaných a neúspěšných pokusech o oboustranou komunikaci pomocí SCPI příkazů byl zvolen přístup s voltmetrem nastaveným na adresu "Talk only", tedy nepřetržitě vysílání odměrů na RS-232 bez vyzvání. Důvod, proč byly tyto pokusy neúspěšné nebyl zjištěn, možnostmi jsou: chybějící handshake vodiče nebo nesoulad převodníku USB \leftrightarrow RS-232 se specifikací RS-232. I v režimu "Talk only" přístroj hlásí chyby přenosu při odesílání zhruba poloviny hodnot. Vzhledem k tomu, že toto měření tvoří minoritní část práce, bylo nalezeno alternativní řešení a tyto chyby nebyly dále zkoumány.

K realizaci tohoto měření byl napsán speciální program (projekt *DAC_static_param*) využívající knihovnu pro ovládání jednotky (*interfacing.c/.h*). Ve stručnosti tento program dělá následující:

- Nastavení parametrů seriové linky (8 datových bitů, 4800 Bd, bez parity) pomocí systémového příkazu
- Připojení k jednotce
- Nastavení režimu DAC na stejnosměrný výstup
- Nastavovací a měřicí smyčka:
 - Nastavení hodnoty DAC a její zápis do souboru
 - Provedení 10 odečtů výstupní hodnoty a jejich zápis do souboru



Obrázek 10.2: Schéma použité pro měření statických parametrů D/A převodníku STM32F3

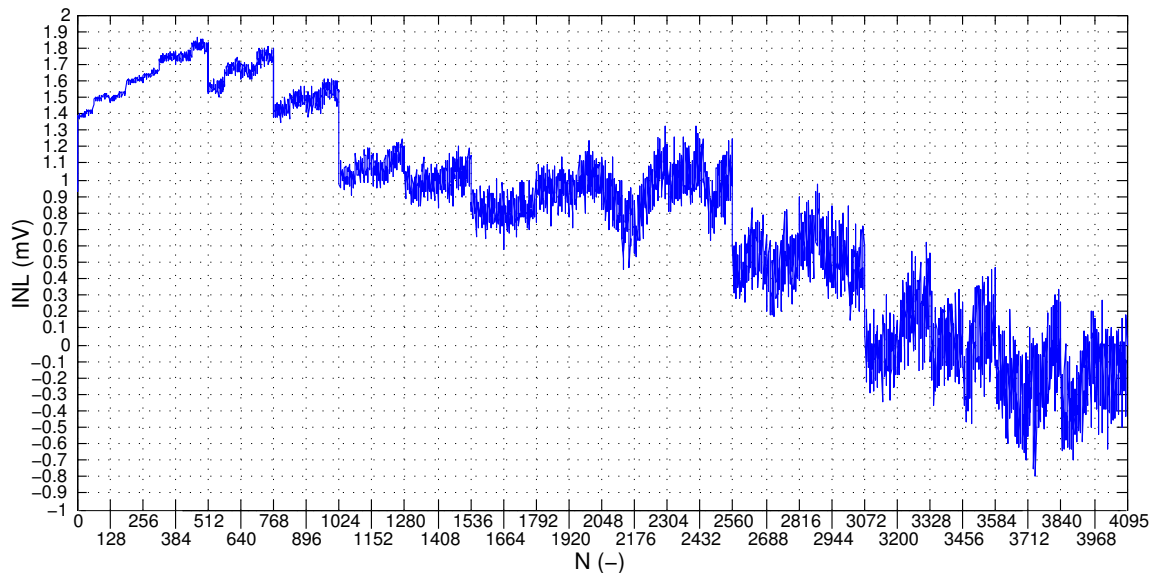
10 odběrů bylo provedeno zejména z toho důvodu, aby bylo zajištěno, že se hodnota na vstupu do voltmetru stihne ustálit. Podle výrobce má D/A převodník výstupní odpor $15\text{ k}\Omega$ a i pozorováním

bylo zjištěno, že doba náběhu při použití delších vodičů výrazně narůstá, proto při zpracování hodnot byl první odměr v každé sérii ignorován.

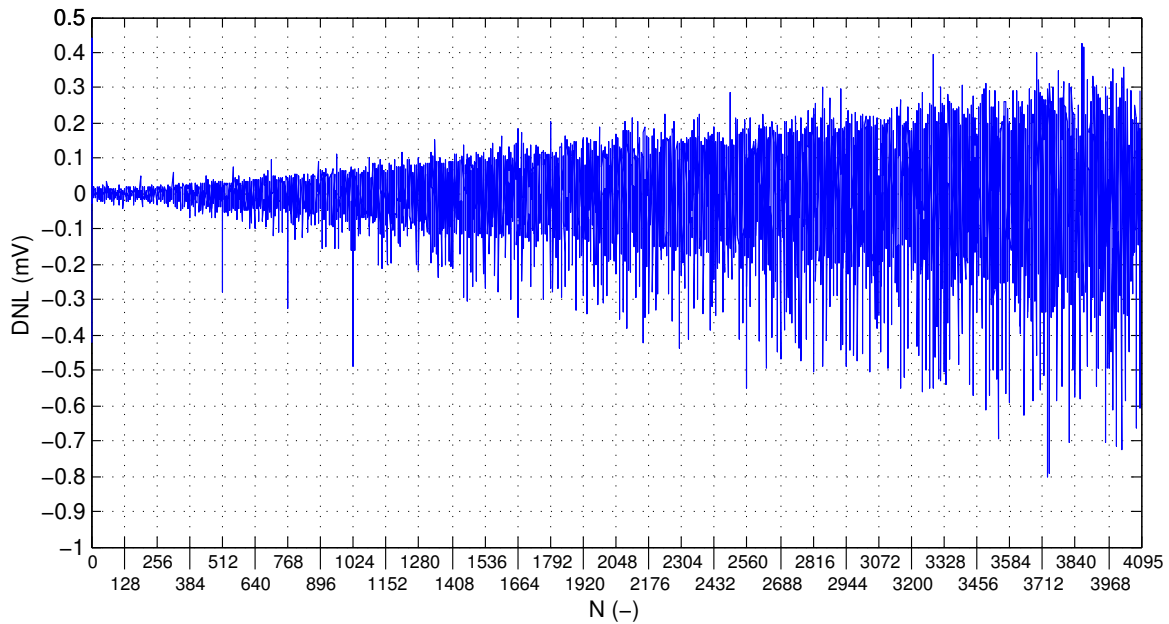
Pro zpracování výstupního souboru byl použit skript v Matlabu (*process_recorded_dac_data.m*), který vytvoří pole velikosti 4096×10 , do něj uloží všechny naměřené hodnoty pro všechna vstupní kódová slova k dalšímu zpracování. Dále vytvoří linearizaci průběhu a srovnáním naměřených dat stanoví INL, porovnáním (s odečtením velikosti LSB) sousedících vzorků DNL převodníku.

Jelikož referenční napětí převodníku není pevně stanoveno a jelikož chyba zesílení lze snadno korigovat zesilovačem/posouvačem, který musí být v dané aplikaci tak jako tak přítomen, vyhodnocení chyby zesílení v tomto měření nebylo uvažováno. Na obrázcích 10.3 a 10.4 jsou zobrazeny popořadě průběhy INL a DNL měřeného převodníku. Za povšimnutí stojí skoky INL v přechodech z některých z kódových slov $256k - 1$ na $256k$ (kde k je celé číslo), což odpovídá deaktivaci elementů převodníku odpovídajících bitům 0 – 7, spojených s aktivací elementu bitu 8. Těmto skokům odpovídají špičky v grafu DNL, které jsou viditelné zejména v přechodech na kódová slova 512, 768 a 1024. Vzhledem k tomu, že za referenční napětí byla vzata maximální hodnota výstupu převodníku, INL s rostoucím kódovým slovem konverguje k nule.

Maximální výstup převodníku je s napájením na použitém discovery kitu roven $2,948V$, čímž je dána velikost LSB $0,720mV$. Maximální INL bylo dosaženo v kódovém slově 471 (1D7 hexadecimálně), a to $1,84mV$ (2,56 LSB). Maximální DNL byla zjištěna mezi kódovými slovy 3734 a 3735, a to $0,8mV$ což převyšuje velikost LSB a projevilo se to mírným poklesem výstupního napětí namísto nárůstu o 1 LSB.



Obrázek 10.3: Průběh INL D/A převodníku procesoru STM32F3 (absolutní maximum $1,84mV$ – 2,56 LSB – v kódovém slově 471)

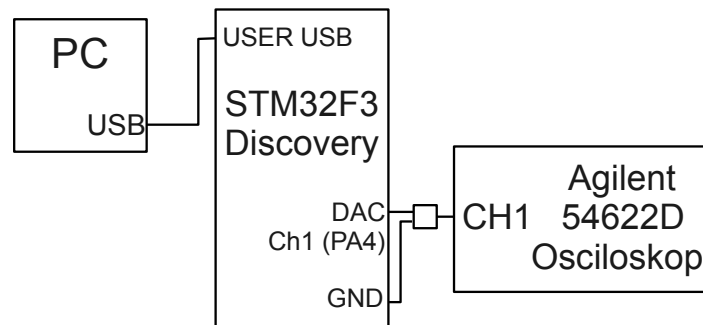


Obrázek 10.4: Průběh DNL D/A převodníku procesoru STM32F3 (absolutní maximum $0,8mV$ – $1,11$ LSB – mezi kódovými slovy 3734 a 3735)

Výsledky měření jsou k dispozici na CD v adresáři `[root]/Mereni/DAC_test`. Soubor `dac_test` obsahuje data, jak byla nashromážděna programem zmíněným výše, `process_recorded_dac_data.m` byl popsán v předchozím odstavci a posledním souborem obsaženým v adresáři je `processed_recorded_data.mat`, jež obsahuje zpracovaná data (výstupní napětí pro každé kódové slovo a průběhy INL a DNL).

10.3 Dynamické parametry

Dynamick Při měření dynamických parametrů D/A převodníku bylo využito obdobné zapojení jako v případě statických parametrů. Pouze byl voltmetr nahrazen osciloskopem, který nebyl nijak spojen s PC, jelikož se při tomto měření zaznamenávaly jednotlivé hodnoty (viz obrázek 10.5).

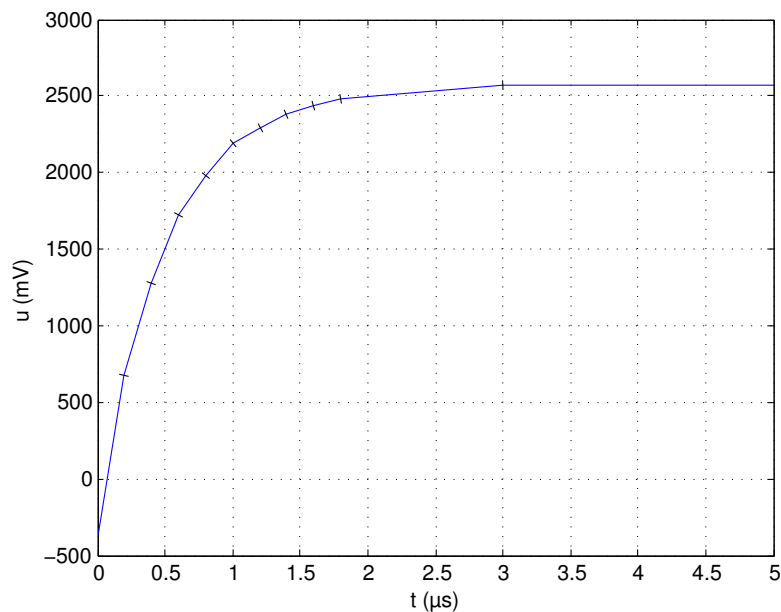


Obrázek 10.5: Schéma použité pro měření statických parametrů D/A převodníku STM32F3

Nejdříve byla proměřena odezva výstupu na skok kódového slova z nulové hodnoty na maximum rozsahu. Odezva je zobrazena na obrázku 10.6 a lze z ní odhadnout časovou konstantu $\tau \doteq 0,57\mu\text{s}$ a mezní frekvenci $f_{-3\text{dB}} \doteq 280\text{ kHz}$. V úvodu kapitoly byla uvedena doba ustálení na $\pm 1\text{ LSB}$ udávaná výrobcem $4\mu\text{s}$. Doba ustálení lze z časové konstanty vypočítat jako

$$t = -\tau \ln \frac{1}{4095} = 4,74\mu\text{s}. \quad (10.1)$$

Vypočtená hodnota zhruba souhlasí s deklarací výrobce, její nárůst může být způsoben parazitními kapacitami na discovery kitu a případně nepřesností odečtu hodnot z osciloskopu.



Obrázek 10.6: Odezva výstupu D/A převodníku na skok z nuly na plný rozsah

Při pokusu o zachycení glitchů D/A převodníku nebyl zaznamenán úspěch. Pro testování byl použit signály přeskakující mezi hodnotami $0x7ff-0x800$, $0x3ff-0x400$ a $0xAff-0xB00$, tedy signály měnící

maximální možný počet bitů pro zobrazení glitchů na osciloskopu. Vzhledem k mezní frekvenci $f_{-3\text{dB}} \doteq 280 \text{ kHz}$ D/A převodníku o málo přesahující čtvrtinu jeho maximální obnovovací frekvence (1 MSps) jsou totiž jakékoliv případné glitche odfiltrovány.

11 Vlastnosti feromagnetik a jejich měření

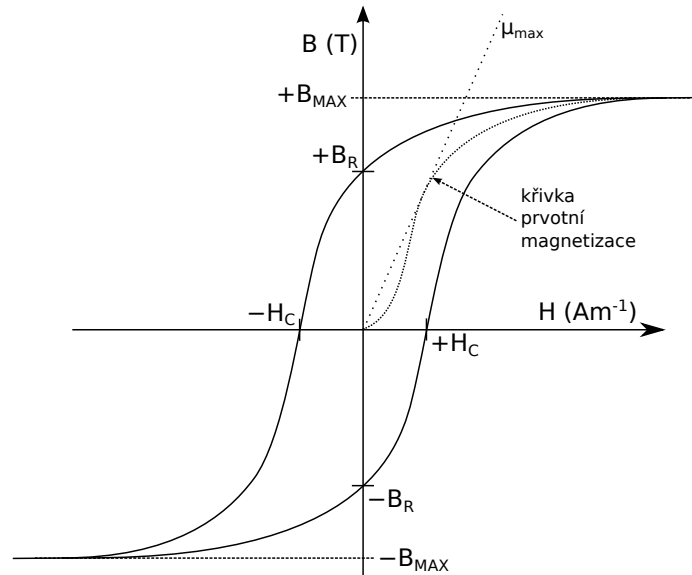
V této kapitole je rozebrána teorie měření hysterézních smyček a měření je poté realizováno modulem navrženým při této diplomové práci.

11.1 Popis vlastností a odvození vztahů

Základními veličinami zavedenými pro popis vlastností feromagnetických materiálů jsou indukce B (T), magnetická intenzita H (Am^{-1}) a permeabilita μ (Hm^{-1}).

Důležitou charakteristikou zachycující všechny důležité parametry feromagnetika je jeho hysterézní smyčka, jež je principiálně zachycena na obrázku 11.1, ve kterém vyznačeny důležité body odpovídající parametrům:

- B_R – remanentní neboli zbytková indukce – je indukce, která zbyde ve feromagnetiku zmagnetizovaném vnějším magnetickým polem po tom, co toto vnější pole zmizí, je způsobena setrávající orientací magnetických domén materiálu
- H_C – koercitivní intenzita – je magnetická intenzita potřebná k tomu, aby byl feromagnetický materiál odmagnetován, tzn. aby se magnetické domény zorientovaly do poloh, kdy se jejich magnetické účinky vyruší a v materiálu nezůstává žádná zbytková indukce po vypnutí vnějšího pole
- B_{max} – saturační indukce – je indukce, při které je materiál již plně nasycen, všechny domény jsou orientovány tak, že maximálně zesilují magnetické pole a při dalším nárůstu magnetické intenzity dochází k nárůstu indukce pouze vlivem permeability vakua
- křivka prvotní magnetizace – je křivka, kterou se řídí závislost magnetické indukce v materiálu na magnetické intenzitě při magnetizaci odmagnetovaného materiálu
- μ_{max} – maximální permeabilita – je podle [15] definována jako sklon tečny ke křivce prvotní magnetizace procházející středem souřadnicového systému



Obrázek 11.1: Obecná hysterézní smyčka

Měření hysterézní smyčky feromagnetika spočívá v jeho buzení střídavým proudem přivedeným do budícího vinutí. Magnetická intenzita je v uzavřeném vzorku definována jako součet všech magnetických intenzit vyvolaných všemi proudy tekoucími ve smyčkách uzavřených kolem vzorku podle Ampérova pravidla pravé ruky. Každý proud v uzavřené smyčce kolem vzorku přispívá k intenzitě hodnotou

$$H_i = \frac{I_i}{l}, \quad (11.1)$$

kde l je střední délka siločáry ve feromagnetiku. Vzhledem k tomu, že se při měření hysterézní smyčky v této práci využívá nejjednoduššího možného způsobu, protéká proud pouze jedním budícím vinutím o počtu závitů N_1 , kde každý závit představuje jednu uzavřenou smyčku. Sekundární vinutí je měřeno přes operační zesilovač s vysokou vstupní impedancí, proud jím tekoucí lze tedy zanedbat. Primární vinutí je tvořeno jedním drátem a celým tedy teče stejný proud I_1 , tím se suma intenzit zredukuje a výsledná intenzita ve vzorku je

$$H = \frac{N_1 I_1}{l}. \quad (11.2)$$

K určení magnetické intenzity ve vzorku tedy stačí znát l , N_1 a měřit proud I_1 .

K odvození vztahu pro magnetickou indukci ve vzorku se využije Maxwellovy rovnice

$$\int_l E dl = -\frac{d\phi}{dt}, \quad (11.3)$$

která říká, že integrál elektrické intenzity přes uzavřenou smyčku je zápornou derivací magnetického toku uvnitř této smyčky. Integrál elektrické intenzity je roven rozdílu potenciálů, tedy napětí. Bude-li se opět jeden závit snímacího (sekundárního) vinutí považovat za jednu uzavřenou smyčku, lze pravou stranu rovnice položit rovnu napětí indukovanému v jednom závitě. Zřetěžením závitů sekundárního vinutí o počtu závitů N_2 do série se všechny tyto elementární zdroje napětí posčítají a tak se získá

upravená rovnice

$$u_2 = -N_2 \frac{d\phi}{dt} , \quad (11.4)$$

kde u_2 je okamžitá hodnota napětí na sekundárním vinutí. K další úpravě vztahu se využije známé rovnice

$$\phi = \int_S B dS , \quad (11.5)$$

která říká, že magnetický tok určitou plochou S je roven integrálu magnetické indukce celou touto plochou. Provede-li se zjednodušení, které bude uvažovat, že veškerý magnetický tok je homogenní a protéká pouze měřeným vzorkem o průřezu S_V , lze integrál nahradit součinem indukce v materiálu a jeho průřezu, čímž se získá vztah

$$\phi = BS_V , \quad (11.6)$$

který se dosadí do rovnice 11.4 a dostane se

$$u_2 = -N_2 S_V \frac{dB}{dt} . \quad (11.7)$$

Zintegrováním rovnice podle času a vyjádřením indukce B vznikne konečný vztah

$$B = - \frac{\int u_2 dt}{N_2 S_V} . \quad (11.8)$$

11.2 Měření hysterézních smyček modulem se STM32F303

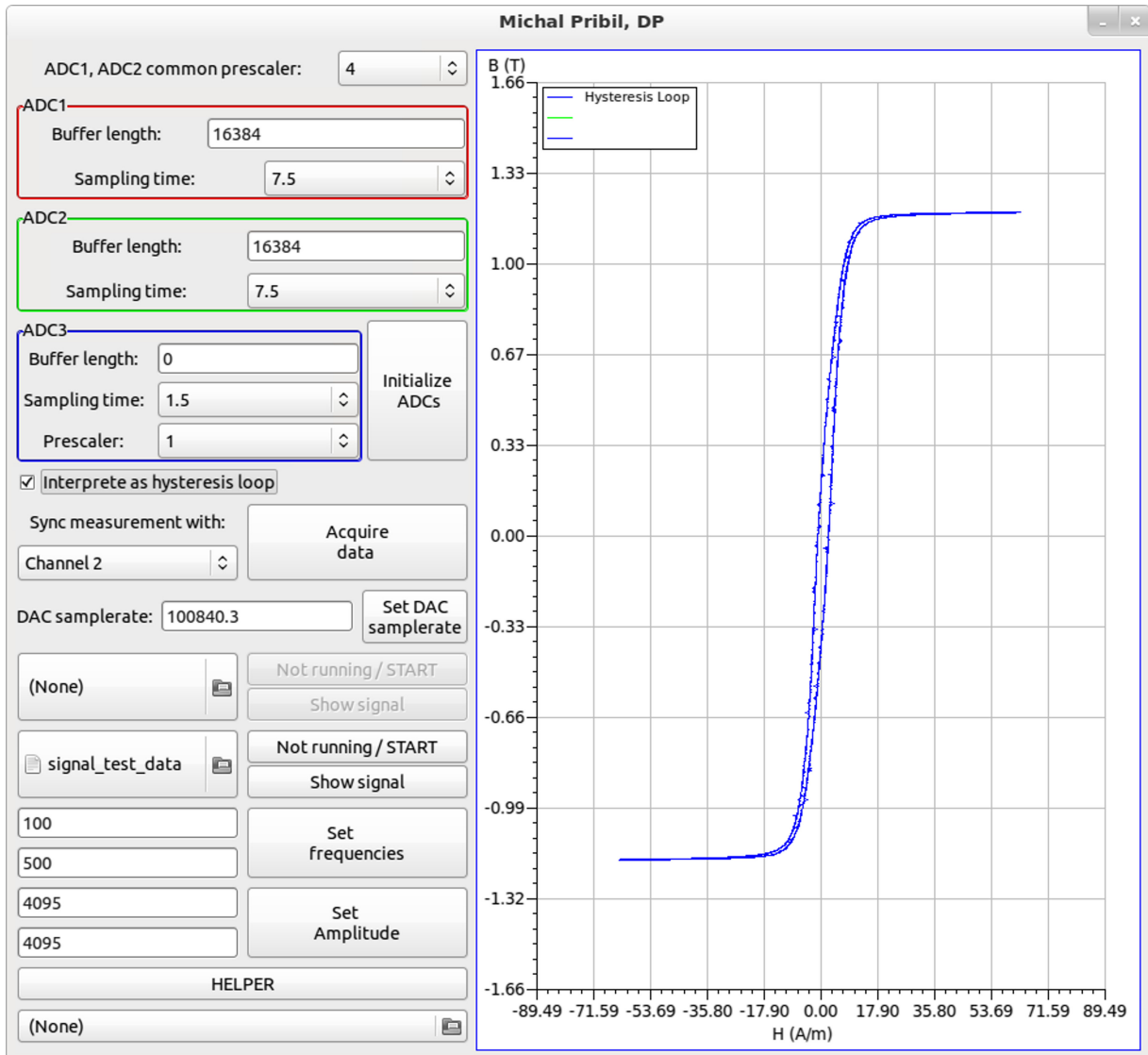
Většina této práce popisovala návrh systému, který prokazuje svoji funkčnost v této, závěrečné, kapitole. Vše okolo tohoto modulu od návrhu hardwaru až po programování aplikace pro jeho řízení bylo podrobně popsáno, proto jsou v této kapitole prezentovány dosažené výsledky.

Parametry vzorku

- střední délka siločáry l : 71 mm
- průřez: 0,2 cm²
- indukce při nasycení: 1,2 T
- saturační intezita: 9 Am⁻¹

Ukázka výstupu programu vyvinutého při této diplomové práci je vyobrazena na obrázku 11.2. Vzorkované průběhy, ze kterých byla tato hysterézní křivka vypočtena byly využity v kapitole 7.3 pro demonstraci vzhledu aplikace.

Jelikož vyvinutá PC aplikace zobrazuje průběhy pochopitelně rastrově a ukládání do vektorového formátu vhodného pro tisknutí by bylo pravděpodobně velice komplikované, ostatní výsledky jsou prezentovány pomocí grafů vytvořených v Matlabu. V průběhu prací bylo mnohokrát zapotřebí uložit surová navzorkovaná data, proto byla tato implementována možnost jejich ukládání. Výsledná hysterézní smyčka se však v žádné fázi vývoje neukládala, proto jsou grafy z Matlabu vytvořeny ze surových vzorkovaných dat algoritmem pro výpočet hysterézní smyčky "řádek po řádku" přepsaným



Obrázek 11.2: Hysterézní smyčka zobrazená pomocí aplikace vyvinuté pro ovládání jednotky se STM32F303

z jazyka C do jazyka Matlabu, výsledky jsou proto ekvivalentní tomu, co bylo zobrazeno v grafu aplikace. Druhým důvodem využití grafů generovaných Matlabem je interpretace vzorkovaných dat ve skutečných hodnotách, tzn. proudu a napětí, což nebylo v PC aplikaci implementováno.

Příklady průběhů změřených na testovaném feromagnetiku pro různé buzení jsou na obrázcích obsažených v příloze B.

Odečtem z naměřených hysterézních smyček byla za prvé potvrzena správnost zpracovávacího algoritmu, jelikož se hodnoty shodovaly s těmi poskytnutými při zapůjčení vzorku, za druhé byly stanoveny hodnoty

- koercitivní intenzita $H_c = 1,7 \text{ Am}^{-1}$
- remanentní indukce $B_r = 0,31 \text{ T}$

12 Vyhodnocení práce

V závěru je uvedeno zhodnocení všeho, čeho bylo v rámci práce dosaženo a jaké přínosy práce přináší.

Implementovaný firmware včetně ovladače pro PC představuje velice dobře fungující základ pro použití např. ve studentských laboratorních úlohách. Jelikož je ovladač psán pro Linux, bylo by nutné jej portovat pro použití ve Windows, což by ale neměl být problém. Firmware lze označit za dobře odladěný. Programování firmwaru mohlo být jistě programátorsky čistší, nebylo využíváno struktur, které by program zpřehlednily a většina funkcí je umístěna v souboru *main*, což dále zhoršuje přehlednost kódu. Důvod, proč toto nebylo změněno, je bezpečnost funkčního kódu a nedostatek času na experimenty. Po stránce efektivity strojového kódu bylo však učiněno maximum pro to, aby byly kritické a často se opakující části rychlé a spolehlivé.

Navržená PC aplikace pro ovládání jednotky určitě není po estetické stránce příliš kvalitní, kvůli nedostatku zkušeností s použitou grafickou knihovnou není roztahování okna patřičně ošetřeno a proto není doporučeno jej neprovádět. Po funkční stránce je tato aplikace na dobré úrovni, názorně demonstruje možnosti firmwaru a mikrokontroléru jako takového a pro kohokoliv, kdo by chtěl přepracovat aplikaci pro jiný, specifický, účel, se jedná o dobře čitelný základ. Aplikace je stabilní při běžném používání, náhodně dojde k jejímu nečekanému ukončení po odpojování/připojování jednotky při jejím spuštění.

Hardwarová část, která byla navržena, splňuje požadavky pro vhodnost začlenění jednotky do široké škály aplikací, jelikož poskytuje dobrý stupeň ochrany A/D převodníků, analogově regulovatelnou amplitudu výstupního napětí DDS generátoru a možnosti jednoduchého nastavení zesílení vstupní části. Větší pozornost mohla být věnována návrhu výstupního budiče, aby bylo zajištěno, že nebude při vyšších frekvencích generovaného signálu docházet k jeho znehodnocování vlivem přechodového zkreslení. Tato práce ukazuje potenciál využití MCU STM32, realizace sofistikované elektroniky by mohla být výborným tématem některé z dalších studentských prací.

Část zabývající se měřením vlastností A/D převodníků představovala nepostradatelnou zkušenost, která vyžadovala velkou dávku technické intuice, přemýšlení a teoretické přípravy, aby bylo měření realizovatelné a relevantní s dostupnými technickými prostředky. Jasně definované limity použití A/D a D/A převodníků byly stanoveny, ostatní omezení jejich aplikace je závislé na jejich použití.

Funkce jednotky byla demonstrována zobrazením hysterézní smyčky předloženého feromagnetika v okně oblužné aplikace a byla ověřena správnost návrhu hardwarové části i algoritmu pro výpočet průběhu hysterézní smyčky.

Tímto lze prohlásit, že byly všechny body zadání úspěšně splněny.

Reference

- [1] H. Yong, W. Wuchen a další. *A 14-bit Successive-Approximation AD Converter with Digital Calibration Algorithm* [online]. 2009
Dostupné z: <http://80.ieeexplore.ieee.org/dialog/cvut.cz/stamp/stamp.jsp?tp=&arnumber=5351481>
- [2] ST Microelectronics. *STM32F302xB STM32F302xC STM32F303xB STM32F303xC Datasheet* [online]. Revize 5, leden 2013
Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/DM00058181.pdf>
- [3] ST Microelectronics. *RM0316 Reference Manual* [online]. Revize 2, březen 2013
Dostupné z: http://www.st.com/web/en/resource/technical/document/reference_manual/DM00043574.pdf
- [4] Analog Devices. *MT-090 TUTORIAL - Sample-and-Hold Amplifiers* [online]. Revize 0, 2009
Dostupné z: <http://www.analog.com/static/imported-files/tutorials/MT-090.pdf>
- [5] GNOME.org. *GTK+ 2 Reference Manual* [online]. Datum poslední změny: 5. března, 2013
Dostupné z: <https://developer.gnome.org/gtk2/>
- [6] Stanford Research Systems. *About Lock-In Amplifiers* [online]. Datum poslední změny: Neznámé, staženo 15. dubna, 2013
Dostupné z: <http://www.thinksrs.com/downloads/PDFs/ApplicationNotes/AboutLIAs.pdf>
- [7] VEDRAL, Josef a FISCHER, Jan. *Elektronické obvody pro měřicí techniku*, monografie ČVUT, Praha: Vydavatelství ČVUT, 2004. 340s. ISBN 80-01-02966-2
- [8] LUNDBERG, Kent H. *Analog-to-Digital Converter Testing* [online]. 2002.
Dostupné z: http://web.mit.edu/klund/www/papers/UNP_A2Dtest.pdf
- [9] libusb.org. *Dokumentace ke knihovně libusb* [online].
Dostupné z: <http://libusb.sourceforge.net/api-1.0/>
- [10] Hewlett-Packard Company. *HP 34401A Multimeter*. 1995. 964-0145 EN
- [11] DOBRUSKÝ, Jiří. *Kompaktní videosenzor*. ČVUT. Diplomová práce. Praha. 2007.
- [12] USB.org. *USB Class Codes* [online].
Dostupné z: http://www.usb.org/developers/defined_class
- [13] Beyond Logic. *USB in a NutShell* [online].
Dostupné z: <http://www.beyondlogic.org/usbnutshell>
- [14] National Instruments. *USB Instrument Control Tutorial* [online]. Publikováno 2. listopadu 2012.
Dostupné z: <http://www.ni.com/white-paper/4478/en>
- [15] Academia.edu. *Ferromagnetic materials* [online].
Dostupné z: http://www.academia.edu/342779/Ferromagnetic_Materials
- [16] PLATIL, Antonín a ROZTOČIL, Jaroslav. Přednášky předmětu Virtuální přístroje [online]. Citováno 8. května 2013. Dostupné z: <http://measure.feld.cvut.cz/vyuka/predmety/A3M38VIP>