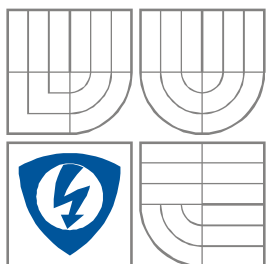


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

MODUL PRO VERIFIKACI ROTAČNÍCH POZIČNÍCH SENZORŮ

MODULE FOR VERIFICATION OF ROTATIONAL POSITION SENSORS

DIPLOMOVÁ PRÁCE
MASTERS'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. DAVID KROLÁK

VEDOUCÍ PRÁCE
SUPERVISORS

Ing. MARIÁN KUBÁŇ, Ing. JIŘÍ DŘÍNOVSKÝ, Ph.D.



Diplomová práce

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

Student: Bc. David Krolák

ID: 146873

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Modul pro verifikaci rotačních pozičních senzorů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi řízení mechanického rotačního systému určeného pro měření pozičních senzorů. Cílem práce bude nalézt vhodnou metodu pro buzení rotačního systému dostupného na pracovišti firmy ON Design Czech, sestávajícího z DC motorku a pozičního indukodéru. Druhým stěžejním bodem práce bude prozkoumání možností přesného snímání a zpracování pozice s přihlédnutím na rychlost otáčení. Výsledný modul by měl být schopen předat informaci o aktuální pozici prostřednictvím vhodně zvoleného komunikačního rozhraní. Navrhněte možnosti časově synchronizovaného měření. Rozeberte a nadefinujte komunikační protokol měřícího modulu spolu s komunikačním rozhraním.

Podle výstupů semestrálního projektu nadefinujte vhodné hardwarové řešení. Zhodnoťte možnost realizace univerzálního měřícího modulu pomocí obvodu programovatelného hradlového pole (FPGA) nebo mikrokontroléru (MCU). Realizujte prototyp měřícího modulu. Vypracujte detailní návrhovou dokumentaci k sestavenému zařízení včetně popisu ovládacího programu. Provedte testovací měření finální verze modulu.

DOPORUČENÁ LITERATURA:

[1] BURKHARD K. Měření, řízení a regulace pomocí PC. BEN-technická literatura: 2003. ISBN 80-7300-089

Termín zadání: 8.2.2016

Termín odevzdání: 19.5.2016

Vedoucí práce: Ing. Jiří Dřínovský, Ph.D.

Konzultant diplomové práce:

doc. Ing. Tomáš Kratochvíl, Ph.D., předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Diplomová práce se věnuje návrhu a realizaci řídicího modulu pro automatizované měření moderních bezkontaktních indukčních snímačů úhlové pozice pomocí mechanického rotačního systému, skládajícího se ze stejnosměrného motoru a inkrementálního kvadrurního kodéru. V práci jsou uvedeny metody buzení stejnosměrného motoru a sběru naměřených dat jednočipovým počítačem. V práci je prezentován komunikační protokol mezi modulem a osobním počítačem prostřednictvím rozhraní USB. Součástí je také návrh ovládacího software a firmware včetně změřených a zhodnocených vlastností řídicího modulu.

KLÍČOVÁ SLOVA

C, C#, inkrementální kvadrurní kodér, jednočipový počítač s architekturou ARM, krystal, rozdílový zesilovač, snižující DC-DC konvertor, servozesilovač, stejnosměrný motor, UART, USB

ABSTRACT

This master's thesis deals with design and realisation of a control module for automated measurements of the modern contactless inductive position sensors by a mechanical rotational system consisting of a DC motor and an incremental quadrature encoder. The thesis presents methods of driving DC motor and recording of measured data by a microcontroller. The thesis presents a communication protocol between the control module and personal computer via USB interface. A part of this thesis is also to develop a control software and firmware, including the measurement and evaluation of properties of the control module.

KEYWORDS

C, C#, crystal, DC motor, differential amplifier, microcontroller with ARM architecture, quadrature encoder, servo-amplifier, step-down DC-DC convertor, UART, USB

KROLÁK, D. *Modul pro verifikaci rotačních pozičních senzorů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2016. 64 s., 9 s. příloh. Diplomová práce. Vedoucí práce: Ing. Marián Kubáň, Ing. Jiří Dřínovský, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Modul pro verifikaci rotačních pozičních senzorů“ jsem vypracoval samostatně pod vedením vedoucích diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucím diplomové práce Ing. Jiřímu Dřínovskému, Ph.D. za administrativní úkony a Ing. Mariánu Kubáňovi za trpělivost, odborné vedení a další cenné rady při zpracování mé diplomové práce. Také bych chtěl poděkovat Ing. Aleši Povalačovi, Ph.D. za zapůjčení vývojového kitu s jednočipovým počítačem při vývoji firmware.

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072

Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

OBSAH

1	Rozbor částí měřicího systému	2
1.1	Stejnoseměrný motor	2
1.2	Tachodynamo.....	3
1.3	Inkrementální rotační kodér.....	4
1.4	Lineární servozesilovač	5
1.5	Synchronní, asynchronní sériové rozhraní USART	6
1.6	Univerzální sériová sběrnice USB	6
1.7	Standard RS-422	7
2	Koncept modulu	8
2.1	Aplikace modulu.....	8
2.2	Blokové schéma.....	10
2.3	Analýza nároků na záznam pozice.....	11
2.4	Komunikační protokol	13
3	Volba součástek a obvodový návrh	16
3.1	DC-DC snižující konvertor.....	16
3.2	LDO napěťový regulátor.....	21
3.3	Rozdílový zesilovač s napěťovou referencí.....	22
3.4	Přijímač RS-422 a volba MCU.....	25
3.5	Krystalový oscilátor MCU.....	26
3.6	Obvodové řešení komunikačního rozhraní	29
4	Firmware	30
4.1	Nastavení periférií MCU a vývojové prostředí.....	30
4.2	Popis programu	35
4.3	Kalibrace	35
4.4	Měření rychlosti otáčení	37
4.5	Krokování pozice.....	38
4.6	Synchronní měření pozice při rotaci.....	40
4.7	Roztočení a zastavení rotoru.....	41

5	Software	43
5.1	Požadavky	43
5.2	Vývojové prostředí	43
5.3	Popis GUI	44
5.4	Popis činnosti.....	47
6	Měření vlastností modulu	51
6.1	Kalibrace	52
6.2	Přesnost nastavené rychlosti	54
6.3	Přesnost krokování.....	56
6.4	Zpoždění záznamu pozice.....	57
7	Závěr	59
A	výrobní podklady desky plošného spoje modulu	65
A.1	Celkové zapojení modulu	65
A.2	Obvodové zapojení napájecí části.....	65
A.3	Obvodové zapojení digitální části.....	66
A.4	Deska plošného spoje – strana součástek	67
A.5	Deska plošného spoje – strana spojů	67
A.6	Osazovací výkres – strana součástek	68
A.7	Osazovací výkres – strana spojů.....	68
A.8	Seznam součástek	69
B	Fotografie finální verze modulu	72
B.1	Pohled ze strany součástek	72
B.2	Pohled ze strany spojů	73

SEZNAM OBRÁZKŮ

Obrázek 1.1	Koncepce a) stejnosměrného a b) krokového motoru (převzato z [30] a [10]).	2
Obrázek 1.2	Ukázka disků inkrementálního a absolutního kodéru (převzato z [29]).	4
Obrázek 1.3	Typické zapojení servozesilovače (převzato z [24]).	5
Obrázek 1.4	Příklad přenosu 8 bitů UARTem (převzato z [37]).	6
Obrázek 1.5	Rozpoznání USB Full Speed zařízení (překresleno z [7]).	7
Obrázek 1.6	Linka RS-422 (převzato a upraveno z [23]).	7
Obrázek 2.1	Blokové schéma měřicího pracoviště.	8
Obrázek 2.2	Mechanický rotační systém.	9
Obrázek 2.3	Blokové zapojení modulu.	10
Obrázek 2.4	Znázornění nekorektního časování záznamu pozice.	11
Obrázek 2.5	Časování zpracování externího přerušení (převzato z [11]).	12
Obrázek 2.6	Graf specifických hodnot pro záznam pozice.	13
Obrázek 3.1	Základní zapojení snižujícího konvertoru.	16
Obrázek 3.2	Fáze spínání: a) nabíjení b) cívka zdrojem proudu c) vybíjení kondenzátoru C1.	16
Obrázek 3.3	Řídící obvod s integrovaným spínačem MC34063 a jeho aplikace (převzato z [16]).	17
Obrázek 3.4	Průběhy napětí před (žlutě, DC vazba) a za cívkou (modře, AC vazba).	20
Obrázek 3.5	Detail jedné periody sepnutí konvertoru.	20
Obrázek 3.6	Rozvržení vývodů a typické zapojení LDO MC33375ST (převzato z [33] a upraveno).	21
Obrázek 3.7	Maximální hodnota ESR výstupního kondenzátoru a převodní charakteristika LDO (převzato z [33]).	21
Obrázek 3.8	Návrh zapojení rozdílového zesilovače.	22
Obrázek 3.9	Frekvenční charakteristiky NCS20074 (převzato z [32]).	23
Obrázek 3.10	Finální zapojení rozdílového zesilovače (analogová část modulu).	24
Obrázek 3.11	Graf stability reference TL431A a typické zapojení (převzato z [22]).	24
Obrázek 3.12	Náhradní zapojení krystalu (převzato z [17]).	26
Obrázek 3.13	Zapojení oscilátoru MCU (převzato z [17]).	27
Obrázek 3.14	Průběh napětí na vstupu a výstupu krystalu.	29

Obrázek 3.15	Ochranné zapojení USB rozhraní.	29
Obrázek 4.1	Konfigurační prostředí STM32CubeMX.....	30
Obrázek 4.2	Rozložení použitých pinů MCU.	31
Obrázek 4.3	Nastavení PLL a děliček hodinových signálů.....	31
Obrázek 4.4	Nastavení čítače/časovače TIM2.	32
Obrázek 4.5	Nastavení vlastností generovaného projektu.	33
Obrázek 4.6	Vývojové prostředí Em::Blocks 2.30.....	34
Obrázek 4.7	Stavový diagram MCU programu.....	35
Obrázek 5.1	Vývojové prostředí Microsoft Visual Studio Ultimate 2013.....	43
Obrázek 5.2	Výchozí vzhled ovládacího programu po spuštění.	44
Obrázek 5.3	GUI ovládacího programu.	44
Obrázek 5.4	Ukázka kontextového menu.....	45
Obrázek 5.5	a) Formulář nastavení vzorkovací periody a b) dialogové okno v případě chybného zadání.	45
Obrázek 5.6	Ideální průběh referenční pozice v a) časové a b) jeho obraz v kmitočtové oblasti.....	46
Obrázek 5.7	Detail ideálního průběhu referenční pozice při rotaci v kmitočtové oblasti.....	46
Obrázek 6.1	Součástí prototypového řešení: a) vývojový kit (převzato z [8]) a b) prototyp rozhraní k mechanickému systému.	51
Obrázek 6.2	Charakteristiky a) výstupního napětí rozdílového zesilovače a b) otáček rotoru při kalibraci.	52
Obrázek 6.3	Průběh výstupního napětí z DA převodníku (modře) a rozdílového zesilovače (žlutě).	53
Obrázek 6.4	Charakteristiky naměřených otáček rotoru.	54
Obrázek 6.5	Relativní chyba nastavení otáček rotoru.	54
Obrázek 6.6	Závislost výstupního napětí rozdílového zesilovače na nastavené rychlosti.	55
Obrázek 6.7	Záznam komunikace s prototypem modulu při měření rychlosti.	55
Obrázek 6.8	Nastavení pro analýzu zpracování signálu SYNCB.	57
Obrázek 6.9	Reakce a doba zpracování signálu SYNCB.....	57
Obrázek 6.10	Záznam komunikace z testování měření pozice prototypem.....	58

SEZNAM TABULEK

Tabulka 1.1	Srovnání stejnosměrného a krokového motoru.	3
Tabulka 1.2	Přehled přenosových rychlostí standardů USB (převzato z [9]).....	6
Tabulka 2.1	Přehled příkazů navrhovaného řídicího modulu.	14
Tabulka 2.2	Rozsah hodnot atributů příkazů řídicího modulu.	15
Tabulka 4.1	Nastavení GPIO pinů MCU.	32
Tabulka 4.2	Výpočty změny úhlu a jeho podmínky.	38
Tabulka 4.3	Výpočty finální úhlové pozice a její podmínky.....	38
Tabulka 6.1	Tabulka naměřených hodnot při krokování pozice.....	56

ÚVOD

V těchto dnech dochází k neustálému zvyšování počtu a vývoji elektroniky kolem nás, obzvláště v automobilovém průmyslu, kde dochází mimo jiné k růstu počtu různých inteligentních tzv. SMART senzorů. Jedním z takových senzorů je bezkontaktní indukční snímač polohy, který se používá v aplikacích náročných na vysokou přesnost měření a linearitu v některých případech i při vysoké rychlosti změny pozice např. pro snímání a řízení otáčivých systémů v automobilu jako je plynový pedál, škrtkací klapka, měření otáček motoru apod.

Tato práce si dává za úkol navrhnout řídicí modul včetně komunikačního rozhraní k ovládání mechanického rotačního systému určeného pro přesné měření rotačních pozičních senzorů, který se nachází na pracovišti firmy ON Design Czech společnosti ON Semiconductor, včetně sběru a zpracování referenční pozice z inkrementálního kvadraturního kodéru v závislosti na synchronizačním signálu s přihlédnutím k maximální možné rychlosti otáčení. Cílem diplomové práce je teoretický rozbor, návrh koncepce modulu, definice komunikačního protokolu, realizace prototypu a změření jeho parametrů.

Následující text je rozdělen do sedmi základních částí. První kapitola se zabývá obecnou definicí a analýzou jednotlivých segmentů měřicího zařízení obsahující porovnání s některými dalšími používanými typy včetně teorie komunikačních standardů použitých v práci. Rozbor řešení a koncept blokového zapojení elektronického systému modulu se nachází v kapitole druhé. Volba součástek a obvodový návrh jednotlivých částí modulu uvádí kapitola třetí. Popis firmware včetně jeho funkcí prezentuje 4. kapitola. V páté kapitole se poté nachází návrh a popis ovládacího software pro počítač. Analýze a měření parametrů realizovaného modulu je věnována kapitola šestá. A na závěr kapitola 7. uvádí shrnutí dosažených výsledků závěrečné práce.

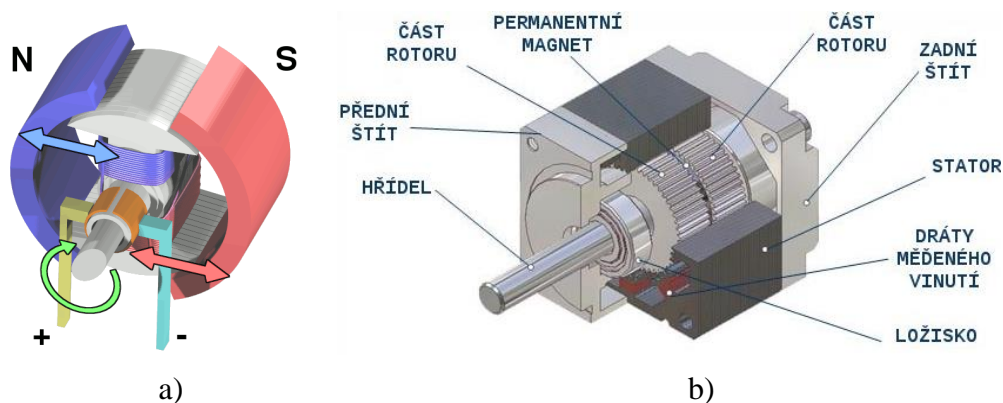
1 ROZBOR ČÁSTÍ MĚŘÍCIHO SYSTÉMU

1.1 Stejnosměrný motor

Primárním úkolem elektrického motoru je přeměna elektrické energie na mechanickou práci. Ve většině případů je činnost elektromotoru založena na principu silových účinků magnetického pole. Existují také typy založené na piezoelektrickém jevu a elektrostatickém poli, ale těmto pohonům nebude nadále věnována pozornost, protože jejich popis není předmětem této práce.

Kartáčový stejnosměrný motor jako jeden z prvních točivých strojů v historii se skládá z permanentního magnetu tvořící stator, ve kterém se pohybuje vinutí rotoru napájené pomocí komutátoru, který mění smysl proudu vinutím pro dosažení vždy stejné orientace indukovaného magnetického pole s vnějším během rotace. Jelikož se souhlasné póly magnetů odpuzují, budou na vinutí jakožto pohyblivý elektromagnet neustále působit síly dle Lorenzova zákona nutící jej k pohybu (viz Obrázek 1.1). Rozdíl od uvedené koncepce a reality je v použití minimálně třípólové kotvy (tři vinutí rotoru), aby nevznikal problém s mrtvým úhlem v důsledku velké mezery mezi póly kotvy zabraňující zkratu napájení.

Krokový motor (Obrázek 1.1) je jistým způsobem stejnosměrný motor tvořený rotorem se stálým magnetem a sudým počtem statorových vinutí (pólové dvojice) buzených právě pulsy stejnosměrného proudu umožňující rotaci hřídele po daných diskretních úhlových krocích nabývajících hodnot od $1,8^\circ$ do 90° [5] v závislosti na počtu pólových dvojic (zubů rotoru u hybridních krokových motorů) a způsobu buzení.



Obrázek 1.1 Koncepte a) stejnosměrného a b) krokového motoru (převzato z [30] a [10]).

Porovnání mezi kartáčovým stejnosměrným a krokovým motorem je uvedeno v tabulce Tabulka 1.1. Výhodou stejnosměrného motoru je snadné řízení otáček a směru s vysokým točivým momentem při nízkých otáčkách, kdežto u krokového motoru převládá výhoda regulace s otevřenou smyčkou zpětné vazby, dlouhodobá životnost a provoz prakticky bez údržby. Při překročení mezního zatížení však dochází ke ztrátě kroku a také má tendenci k mechanickému zakmitávání při běžném provozu.

Tabulka 1.1 Srovnání stejnosměrného a krokového motoru.

Typ motoru	Kartáčový stejnosměrný	Krokový
Řízení otáček	Napětím, pro konstantní otáčky nevyžaduje elektroniku, jednoduché	Spínáním proudů do vinutí vyžadující elektroniku s přesným časováním, složité
Linearita řízení	Lineární závislost otáček na napětí kotvy v širokém rozsahu	Diskrétní polohy, nespojitý pohyb při nízkých otáčkách
Otáčky	Vysoké, omezené komutací	Nízké, asi 500 ot/min dle [5]
Nastavení pozice	Nutná zpětná vazba ve spojení s řídicí elektronikou, pozici nelze zafixovat	Vybuzením definovaným počtem pulzů pomocí elektroniky bez zpětné vazby, možnost zablokování rotoru
Přesnost pozice	Daná zpětnovazebním snímačem a způsobem řízení	Určená počtem pólových dvojic (zubů u hybridního typu) a způsobem buzení
Údržba	Periodická, opotřebením uhlíkových kartáčů díky elektrickým obloukům při rozpojování indukční zátěže	Pouze ložiska
Životnost	Krátká	Dlouhá

1.2 Tachodynamo

Jedná se o elektrodynamický snímač otáček, jehož stejnosměrné výstupní napětí je přímo úměrné úhlové rychlosti a polarita směru otáčení. Skládá se z permanentního magnetu tvořící stator a rotor s vinutím, které se otáčí v magnetickém poli statoru. Indukované výstupní napětí snímané komutátorem skládající se typicky z nízkou odporových uhlíkových sběračů lze dle Faradayova zákona popsat vztahem [3]:

$$u_{\text{dynamo}} = (2 \cdot N \cdot h \cdot r \cdot B) \cdot \omega_c = k \cdot \omega_c, \quad (1.1)$$

kde N , h a $2r$ reprezentuje počet závitů, výšku a průměr vinutí, B je magnetická indukce a ω_c představuje úhlovou rychlost. Konstanta úměrnosti k je známá jako konstanta zpětné elektromotorické síly, anglicky *back-emf constant*, nebo napěťová konstanta.

Výhodou tachodynamu je vysoká citlivost, naopak mezi velkou nevýhodou patří nepřijemnosti spojeny s přechodovými jevy v důsledku komutace projevujícími se jako jiskření způsobující opotřebením kontaktů, zvlnění a šum výstupního napětí.

Tyto negativní vlastnosti odstraňuje tachoalternátor, jehož rotor je tvořen stálým magnetem, který při otáčení indukuje do vinutí statoru elektromotorické střídavé napětí taktéž přímo úměrné rychlosti otáčení. Změna úhlové rychlosti otáčení se projeví jako změna amplitudy indukovaného napětí a jeho kmitočtu, jež závisí také na počtu pólových dvojic rotoru. To umožňuje přesně změřit rychlost otáček pomocí čítače, což je přesnější než měření napětí ovlivněné teplotou a změnou magnetického pole, ale na rozdíl od tachodynamu nelze určit směr otáčení.

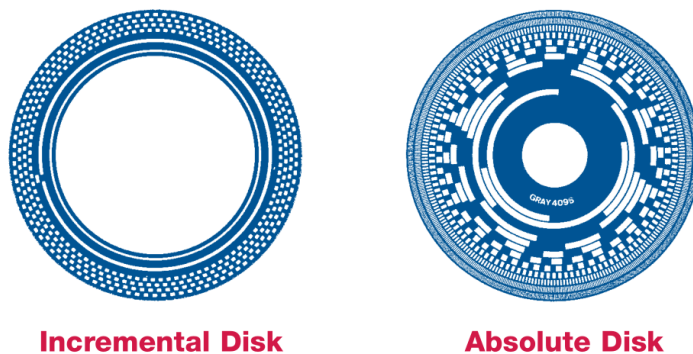
1.3 Inkrementální rotační kodér

Každý převodník, který generuje jistou formu digitálního kódu, může být nazván jako kodér. Rotační kodér se používá pro měření úhlové pozice a rychlosti. Ve většině případů nachází uplatnění v robotických manipulátorech, polohovacích systémech satelitů, rotačních strojích jako jsou motory, pumpy, kompresory, turbíny apod. Rozlišení u všech typů závisí na množství pulsů generovaných za jednu otáčku kodéru a tím také na velikosti výstupního digitálního slova u absolutního kodéru. Mezi pozitivní vlastnosti patří vysoká přesnost (zejména šumová imunita) díky vynikající konstrukci, spolehlivost a relativní jednoduchost implementace v číslicovém řídicím systému v porovnání s jejich analogovými protějšky.

Rotační kodéry lze rozdělit kromě dvou hlavních kategorií absolutních a inkrementálních daných interpretací výstupu převodníku podle fyzikálního principu snímání na optické, magnetické, kontaktní a založené na jevu blízkosti. Poslední tři typy se používají ve speciálních případech např. při extrémních teplotách [1]. Dále se bude práce věnovat optickým rotačním kodérům, které jsou nejrozšířenější.

Výstupem inkrementálního rotačního kodéru je série pulsů v závislosti na rotaci disku. Čítáním pulsů za určitý čas již z principu umožňuje měřit úhlovou změnu a rychlost. Natočení rotoru lze získat vzhledem k referenčnímu bodu čili bázové pozici, jejíž signál bývá označován jako M „Mark“ nebo I „Index“. Nástupná hrana tohoto signálu se používá k nulování čítače pozice a může se také použít k inkrementaci čítače otáček, jehož obsah se nuluje po uplynutí daného časového intervalu. Určení směru je umožněno kvadraturním výstupem, kdy výstupní pulsy jsou navzájem posunuty o 90°. Rozlišení úhlu je dáno počtem pulsů na otáčku (PPR).

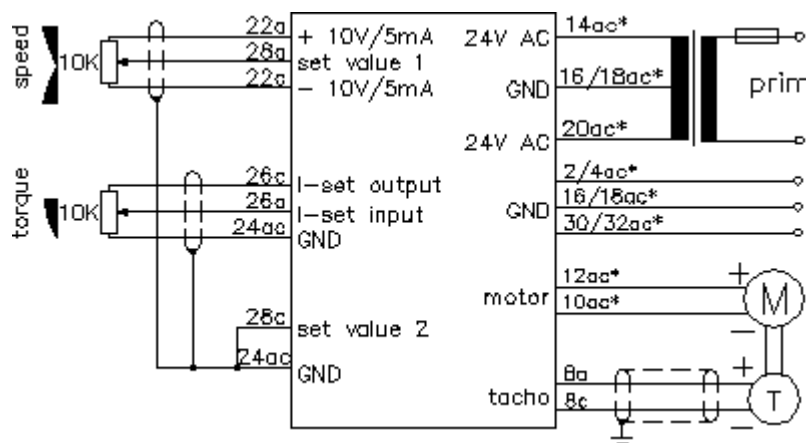
Mezi výhody inkrementálního kodéru patří menší počet signálních vodičů vzhledem k absolutnímu kodéru a menší požadavky na vlastnosti kódového disku včetně snímacího systému což pochopitelně snižuje cenu kodéru. Při každém zapnutí akviziční jednotky je však vždy nutné inkrementální kodér kalibrovat, tedy nalézt nulovou pozici. Na Obrázek 1.2 se nachází typické provedení kódovacích disků (nejčastěji ze skla) inkrementálního a absolutního kodéru. Více informací se lze dočíst v [29].



Obrázek 1.2 Ukázka disků inkrementálního a absolutního kodéru (převzato z [29]).

1.4 Lineární servozsilovač

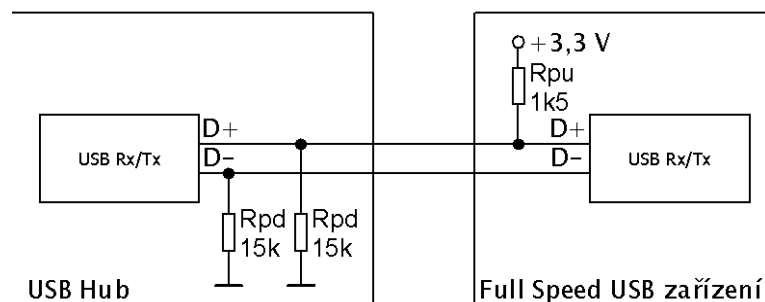
Servozsilovač je zesilovač se zápornou zpětnou vazbou a výkonným výstupním stupněm umožňující řídit a udržovat konstantní otáčky kartáčového stejnosměrného motoru. Základní princip regulace otáček spočívá v nulovém rozdílu napětí z nastavovacího prvku a korigovaného výstupu tachodynama. Koncový stupeň lineárního zesilovače je konstruován tak, aby umožňoval spojitou regulaci, což znamená, že napětí nutné k dosažení určité rychlosti otáčení je trvale k dispozici na svorkách motoru. Čtyřkvadrantové řízení, charakteristické zapojením výstupních tranzistorů do tzv. H-můstku, je využito k pohonu či brzdění v obou směrech v závislosti na polaritě řídicího signálu. Zesilovač MAR 25/2 firmy Mattke Ag [24] umožňuje nastavení rychlosti a momentu pomocí kaskádního řízení zajišťující neměnnost nastavených hodnot při proměnlivé zátěži motoru. Kromě preferované regulace s tachodynamem (viz Obrázek 1.3) lze použít řízení pomocí indukovaného elektromotorického napětí ve vinutí metodou kompenzace $I \cdot R$ na výstupu nevyžadující žádný mechanický snímač na motoru.



Obrázek 1.3 Typické zapojení servozsilovače (převzato z [24]).

Předností je absence elektrických nebo akustických rušení projevující se jako „pískání“ v důsledku kmitání plechů vinutí motoru v porovnání se spínanými servozsilovači, jež jsou založeny na principu nespojitě regulace zapínáním a vypínáním koncových tranzistorů v daných časových intervalech, kdy výstupní napětí odpovídá střední hodnotě výstupních pulsů, a proto jsou lineární servozsilovače doporučovány pro použití v laboratořích a lékařských přístrojích. Na druhou stranu nabývají velkých rozměrů kvůli potřebnému chladiči k ochraně výstupních tranzistorů před přehřátím, kterými trvale protéká proud při buzení motoru.

Ztráty u servozsilovačů s malým výstupním výkonem pohybující se v mezích 30 až 50 W jsou akceptovatelné a při volbě budiče motorku převážila výhoda nízkého rušení spolu s cenou výkonové jednotky oproti celkové účinnosti, tedy ponechání lineárního servozsilovače. Více informací lze nalézt v [24] a [21].



Obrázek 1.5 Rozpoznání USB Full Speed zařízení (překresleno z [7]).

Pro vlastní vývoj USB zařízení je kromě zvládnutí teorie nutné napsat vhodný ovladač pro operační systém a zakoupit 16 bitové identifikační číslo výrobce VID od USB organizace, které spolu s PID tvoří jednoznačnou a nezaměnitelnou identifikaci zařízení. Pro snadnou implementaci je dostupný převodník mezi USB a UARTem jako je např. FT232RL [25] s maximální přenosovou rychlostí 3 MBd od výrobce FTDI s příslušným volně šiřitelným ovladačem (PC program umožňující používat dané zařízení). Podobný převodník bývá také v dnešní době implementován v moderních mikrokontrolérech (jednočipových počítačích), jejichž výrobce taktéž poskytuje volně šiřitelný ovladač.

1.7 Standard RS-422

RS-422 včetně RS-485 patří mezi průmyslové sběrnice. Rozdíl spočívá v tom, že RS-422 se používá pro jednoduchý rychlý sériový přenos dat. Pro určení maximální přenosové rychlosti lze aplikovat pravidlo podle [23], kdy by přenosová rychlost v bitech za sekundu vynásobením délky vedení v metrech neměla překročit hodnotu 10^8 . Podle uvedeného pravidla lze například na vzdálenost 500 m komunikovat s přenosovou rychlostí maximálně 200 kb/s. Odolnost vůči rušení zajišťuje stíněný kroucený pár tvořící diferenciální linku, která musí být na svém konci pro vyšší komunikační rychlost než 200 kb/s a vzdálenost větší než 1 m zakončena rezistorem R_T odpovídající velikosti charakteristické impedanci kabelu Z_0 s akceptovatelnou tolerancí (typ. 100Ω , $\pm 20\%$ dle [23]) kvůli zabránění odrazům od konce vedení a tím zkreslení signálu (viz Obrázek 1.6).



Obrázek 1.6 Linka RS-422 (převzato a upraveno z [23]).

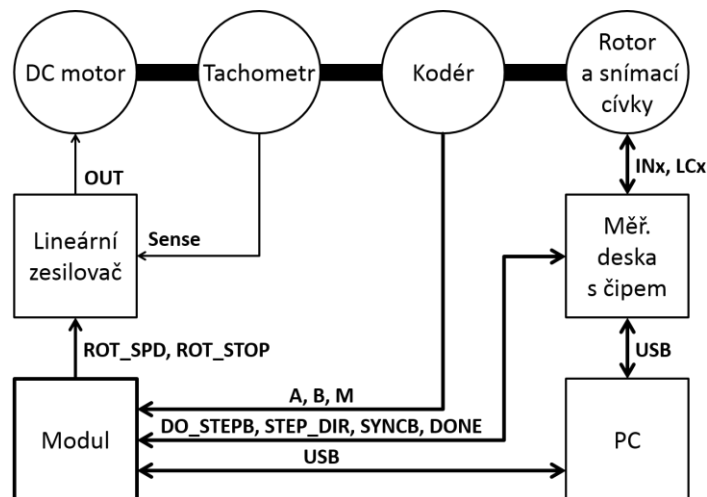
2 KONCEPT MODULU

Návrh řídicího modulu je podřízen použitými zařízeními, se kterými bude modul spolupracovat. Typ a princip součástí, ze kterých se skládá rotační mechanický systém, byly popsány v předchozí kapitole. Jde především o DC motorek s planetovou převodovkou způsobující snížení otáček na 1/14 otáčky motorku, tachodynamo, inkrementální kodér a výkonový lineární servozesilovač sloužící pro buzení motorku (již hotová jednotka - není součástí práce).

Po obeznámení s těmito zařízeními je pak možné zvolit, jaké informace a v jakém formátu bude modul zpracovávat, jestli bude nezbytně nutné dalších speciálních obvodů nebo postačí periferie zvoleného mikrokontroléru.

2.1 Aplikace modulu

Modul je určen pro aplikaci testování při vývoji moderních bezkontaktních indukčních pozičních sensorů, kdy se zjišťuje přesnost měřeného úhlu natočení vůči referenční hodnotě zaznamenané právě touto jednotkou, a proto je kladen velký důraz na synchronizaci záznamu dat se záznamem probíhajícím na měřicí desce s testovaným čipem. Na mechanickém systému se pro zkoušený integrovaný obvod nachází excitační (LC) a snímací (IN) cívký, které se propojují s měřicí deskou pomocí vodičů LCx a INx, včetně rotoru tvořící vinutí nakrátko (viz Obrázek 2.1).



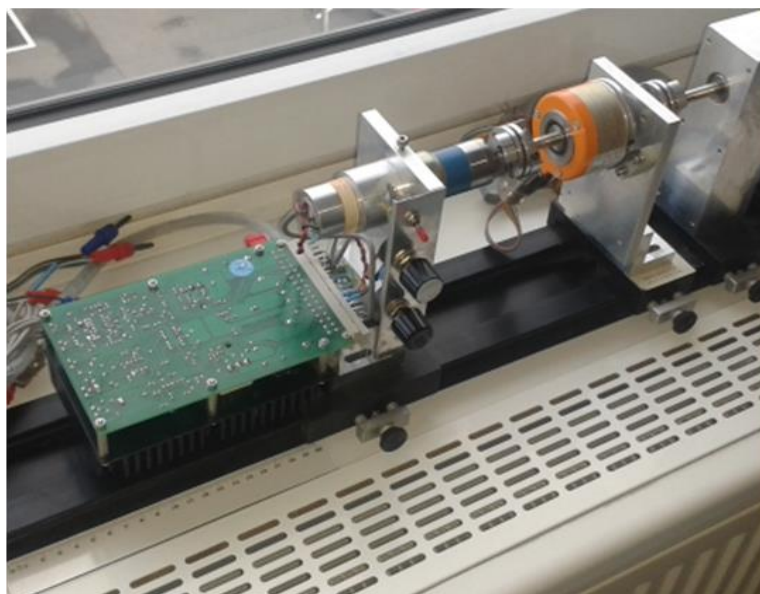
Obrázek 2.1 Blokové schéma měřicího pracoviště.

Úkolem modulu je také řízení stejnosměrného motorku pomocí budícího výkonového lineárního zesilovače firmy Mattke Ag [24], který obsahuje symetrický zdroj referenčního napětí 10 V se zatížitelností 5 mA, zdroj konstantního proudu a napěťové vstupy pro regulaci krouticího momentu a otáček motorku včetně směru otáčení. Řízení rychlosti otáčení umožňuje analogový signál ROT_SPD, anglicky *Rotor Speed*, generovaný upraveným rozdílovým zesilovačem v modulu napájeným z referenčního zdroje zesilovače (viz kapitola 3.3). Nastavení krouticího momentu je provedeno manuálně omezením výstupního proudu koncového stupně zesilovače

pomocí potenciometru napájeného zdrojem konstantního proudu, jehož běžec je připojen na příslušný vstup zesilovače. Motorek se zastavuje pomocí signálu ROT_STOP, který přes sériový ochranný rezistor 12 k Ω (výpočet uveden v kapitole 3.4) spíná NPN bipolární tranzistor zapojený jako spínač se společným emitorem. Sepnutím tranzistoru dojde k nastavení nulového proudu motorkem a tím také k nastavení nulového momentu. Důvodem tohoto řešení je vstupní napěťová nesymetrie servozesilovače způsobující chod motorku na nízkých otáčkách při nulovém napětí signálu ROT_SPD.

Signály DO_STEPB a STEP_DIR, charakterizující povel k provedení kroku a jeho směr, slouží k přímému řízení otočení o předem zadaný úhlový krok v módu „externího“ krokování pozice. Synchronizace záznamu pozice v režimu synchronního měření pozice při rotaci je provedena pomocí signálu SYNCB, anglicky *Synchronization*, přičemž sufix B (*Bar*) znamená signál aktivní v logické nule. Řídicí systém je navrhován tak, aby reagoval na sestupné hrany příkazových signálů DO_STEPB a SYNCB. Dále byl implementován signál DONE, který je v klidovém stavu v logické jedničce a v případě provádění kroku či měření rychlosti otáčení rotoru je vybaven do logické nuly. Po dokončení kroku nebo ukončení měření rychlosti je nastaven zpět do logické jedničky.

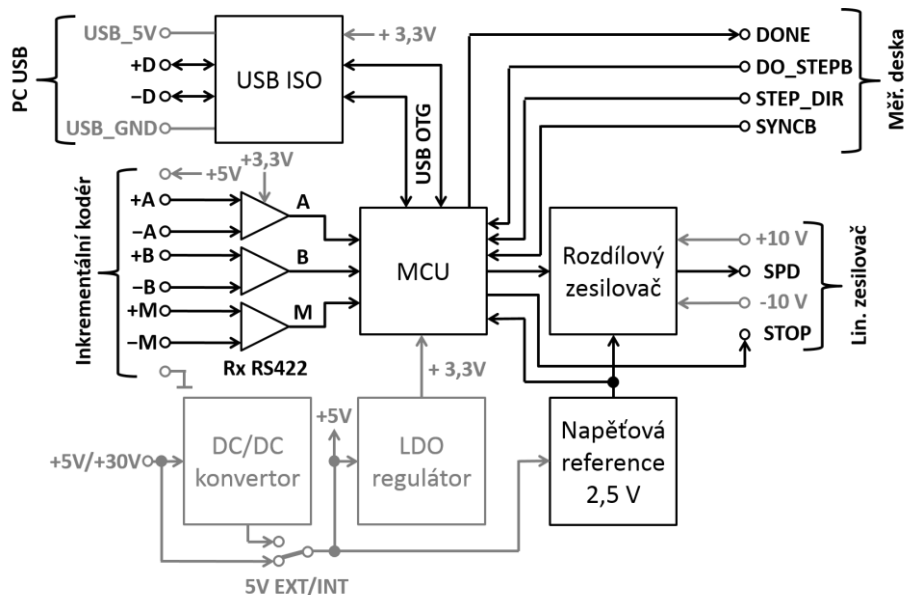
Na Obrázek 2.2 je uveden mechanický rotační systém se servozesilovačem v porovnání s dvacetimetrovým pravítkem, kde je patrné manuální ovládání otáček a momentu. Páčkovým přepínačem se pak volí mezi manuálním ovládáním a externím řízením pomocí modulu.



Obrázek 2.2 Mechanický rotační systém.

2.2 Blokové schéma

Návrh blokového zapojení řídicího modulu se nachází na Obrázek 2.3 a vychází z předchozích požadavků systému. Jádrem celého modulu je MCU, který ovládá externí zařízení vhodně k němu připojená a zprostředkovává komunikaci s osobním počítačem (dále PC), která zahrnuje povely a předávání naměřených dat v diskretním čase. Na rozdíl od programovatelného pole (FPGA), které se používá zejména v aplikacích pro zpracování velmi rychlých signálů v reálném čase s přesným časováním, MCU vykonává program sekvenčně a je tak určen pro nepříliš náročné aplikace na časování. Podmínkou pro použití MCU v aplikaci náročnější na časování je dostatečná velikost kmitočtu hodinového signálu, který určuje čas vykonání jednotlivých instrukcí. Programem se rozumí posloupnost instrukcí uloženou v programové paměti.



Obrázek 2.3 Blokové zapojení modulu.

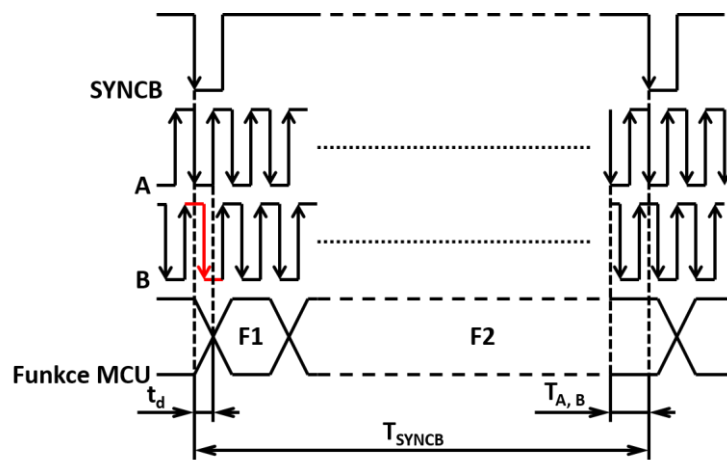
V modulu se také nacházejí přijímače diferenciálních signálů A, B a M standardu RS-422 z inkrementálního kodéru, které jsou přijímačem přeloženy na nesymetrický signál s odpovídajícími napěťovými úrovněmi tak, aby je mohl MCU zpracovat.

Rozdílový zesilovač slouží k rozšíření rozsahu řídicího napětí servozesilovače do záporných hodnot a jeho návrh s napěťovou referencí je popsán v kapitole 3.3. Napěťové posouvače pro oddělení napěťového rozsahu externích a interních vstupních signálů není nutné implementovat do hardwarového návrhu z důvodu výběru vstupně výstupních pinů MCU schopných zpracovávat signály v 5 V napájecí doméně. Samozřejmě je vybavení těchto vstupních linek ochranným sériovým rezistorem o velikosti 100 Ω .

Celý modul včetně kodéru je možné napájet buď z externího 5 V zdroje nebo z důvodu snížení počtu potřebných napájecích zdrojů na pracovišti z pozitivní části napájení servozesilovače, jehož napětí činí 30 V. V případě 30 V napájení je nutné toto napětí snížit na 5 V, kdy maximální proudový odběr činí zhruba 0,5 A. Protože výkonová ztráta lineárního napěťového regulátoru by byla v tomto případě značná, bylo zvoleno řešení se snižujícím spínaným měničem (DC/DC konvertorem), který dosahuje menších výkonových ztrát a tím vyšší účinnosti.

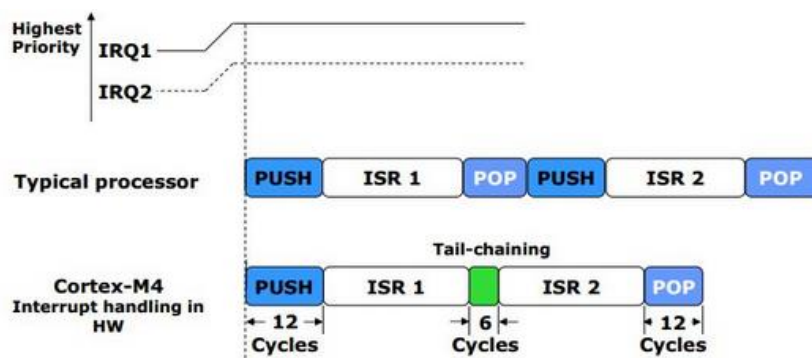
2.3 Analýza nároků na záznam pozice

Požadavky pro akvizici pozičních dat jsou dány maximální frekvencí synchronizačních pulsů a rozlišením inkrementálního kodéru. Rozlišení předloženého kodéru EDH581 s kvadraturním výstupem firmy INDUcoder je 100 000 PPR [12], které s využitím nástupných a sestupných hran výstupních signálů lze zčtyřnásobit. Nejdůležitější je zaručit, že při maximálních otáčkách rotačního systému omezených nejvyšším kmitočtem výstupních budičů linek RS-422 dojde s akceptovatelným zpožděním k vykonání podprogramu obsluhy externího přerušeni a včasnému vyčtení registru čítače pozice, než se inkrementuje nebo dekrementuje. Obrázek 2.4 ukazuje nekorektní zachycení referenční pozice vlivem velkého zpoždění t_d , kdy F1 představuje vykonání podprogramu vyvolaného sestupnou hranou synchronizačního signálu SYNCB a F2 činnost programu provádějící další operace ve volném čase, jako je například testování, zdali není dosaženo zvoleného počtu naměřených vzorků.



Obrázek 2.4 Znázornění nekorektního časování záznamu pozice.

V případě použití MCU s jádrem Cortex-M4 trvá přechod do obslužného programu přerušeni 12 taktů. V případě příchodu dvou externích přerušeni v jeden okamžik se začne vykonávat přerušeni s vyšší prioritou a následný podprogram druhého přerušeni se provede se zpožděním pouze 6 taktů (viz Obrázek 2.5 a [11]). Výchozí taktovací kmitočtet továrně nastaveného interního RC oscilátoru MCU STM32F407 [15] činí 16 MHz s tolerancí 1 % přes pracovní teplotní rozsah bez aktivované PLL kmitočtové násobičky, které umožňuje dosáhnout taktování až 180 MHz.



Obrázek 2.5 Časování zpracování externího přerušení (převzato z [11]).

Při použití kmitočtu RC oscilátoru bez zvýšení implementovanou násobičkou je zpoždění vykonání podprogramu 750 ns. S uvažováním maximálního kmitočtu výstupu z kodéru 300 kHz [12] a reakcím na hrany reaguje čítač každých 833 ns, což je ve srovnání se zpožděním vykonání podprogramu na hranici spolehlivosti vyčítání korektní pozice. Použitý MCU je vybaven externím krystalem s rezonančním kmitočtem 8 MHz pro vnitřní vysokorychlostní oscilátor (HSE) jehož výstup se po zapnutí nebo restartu MCU interně připojí na vstup PLL násobičky pomocí nastavení příslušných registrů. Ta je nastavená tak, aby systémový takt byl 168 MHz. Poté zpoždění provedení podprogramu pro externí přerušení je zcela zanedbatelné.

Z definice rychlosti otáčení n , které se udává v otáčkách za minutu (ot/min), maximálního kmitočtu výstupního signálu kodéru $f_{\max A, B}$, rozlišení kodéru Res a periody synchronizačních pulsů (vzorkování) T_{SYNCB} byly odvozeny následující vztahy popisující maximální možné otáčky pro teoreticky ještě správné měření pozice (2.1), periodu čítače (2.2) a minimální periodu vzorkování (2.3) při daných otáčkách rotoru a požadovaného minimálního počtu vzorků na jednu otáčku N_{\min} . Parametr p udává násobek zvýšení rozlišení inkrementálního kodéru, kdy při reakcích čítače pozice na nástupnou i sestoupnou hranu u kvadrurního signálu nabývá hodnoty 4.

$$n_{\max} = \frac{60 \cdot f_{\max A, B}}{Res} \quad (2.1)$$

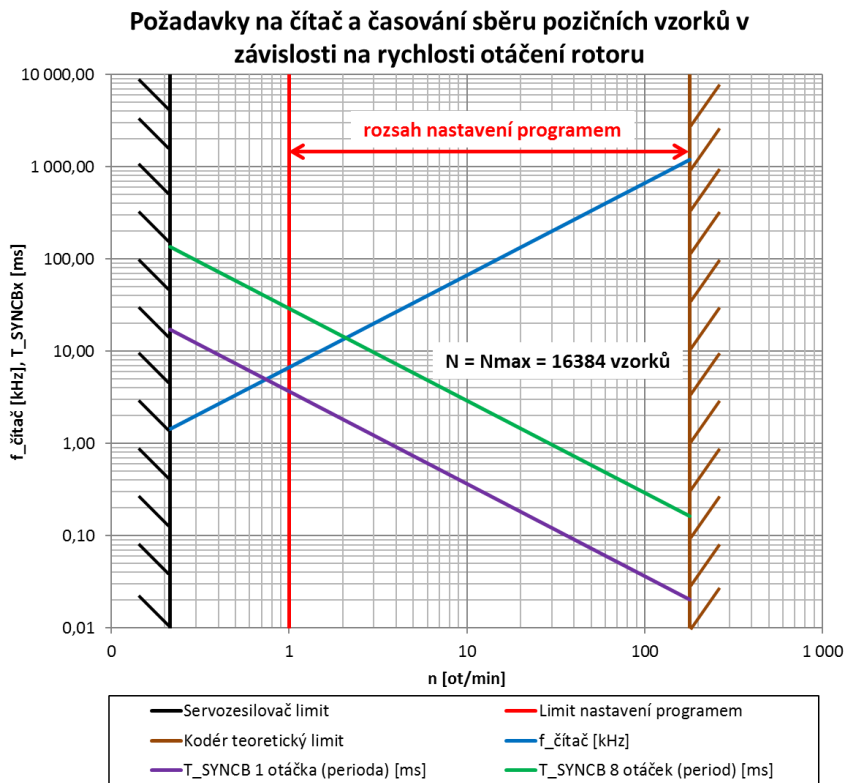
$$T_{\text{inkrement}} = \frac{1}{f_{\text{čítač}}} = \frac{60}{n \cdot p \cdot Res} \quad (2.2)$$

$$T_{\text{SYNCB min}} = \frac{60}{N_{\min} \cdot n}, N_{\min} \geq 2 \quad (2.3)$$

Po dosažení hodnot de facto maximální přenosové rychlosti budičů linky RS-422 a rozlišení inkrementálního kodéru se dochází k závěru, že teoretická nejvyšší rychlost otáčení činí 180 ot/min. Při vyšších otáčkách klesá spolehlivost správného záznamu pozice.

$$n_{\max} = \frac{60 \cdot 300 \cdot 10^3}{100 \cdot 10^3} = 180 \text{ ot/min}$$

Z grafu uvedeného na Obrázek 2.6 lze vyčíst mezní požadavky na rychlost čítání čítače, která v tomto případě činí 1,2 MHz. Dále lze z grafu zjistit synchronizační (vzorkovací) periodu pozičních dat, je-li požadavek zaznamenat jednu otáčku nebo 8 otáček do daného celkového počtu vzorků, jehož maximum je 16384 (mocnina 2) kvůli pozdějšímu zpracování FFT algoritmem.



Obrázek 2.6 Graf specifických hodnot pro záznam pozice.

2.4 Komunikační protokol

Řízení modulu je založeno na sériové komunikaci z PC po USB sběrnici, kdy v koncovém bodu dochází k translaci z USB na UART protokol, který se v počítači tváří jako virtuální sériový port. Komunikační protokol je nadeřinovaný tak, že se přenáší příkazový řetězec v jistých případech doplněný o číselnou hodnotu či hodnoty v ASCII kódu. Poté v závislosti na přijatých datech ze sériového portu řídicí modul zajistí požadovanou funkci v součinnosti s mechanickým rotačním systémem. Seznam implementovaných příkazů je uveden v tabulce Tabulka 2.1 a jejich atributů v Tabulka 2.2.

Každý příkaz odeslaný z počítače k modulu je zakončen znakem CR (carriage return - ASCII 0x0D) a to i pokud příkaz vrací hodnoty tedy pro komunikaci v opačném směru. Atributy jsou odděleny od příkazu a od sebe mezerou (SP, space - ASCII 0x20). Zvolený formát rámce je 8 datových bitů, jeden stop bit a žádný paritní bit. Pro prvotní testování bylo využito spojení přes hyperterminál s přenosovou rychlostí 9600 Bd. V ovládacím software (počítačové aplikaci) je pak přidána příslušná knihovna umožňující práci se sériovým portem (otevření, zápis, čtení a uzavření) na téže přenosové rychlosti.

Tabulka 2.1 Přehled příkazů navrhovaného řídicího modulu.

Funkce	Příkaz	Popis
Zahájení komunikace	Hi ONROT_CTRL!	Po zapnutí řídicí modul očekává tento příkaz, aby mohl sdělit hodnoty kalibrační procedury, která se provádí hned po inicializaci firmware (MCU programu).
Kalibrace	CALIB	Znovu provede kalibrační proceduru.
Měření pozice	MPOS n <i>směr</i> N	Rotor se natočí na báзовou pozici a poté se roztočí rychlostí n , v požadovaném směru <i>směr</i> a zahájí se zápis N vzorků pozice na sestupnou hranu signálu REQБ. Po dosažení N vzorků naměřených dat modul ukončí měření a odešle záznam pozic do počítače.
Krokování	STEP <i>spouštění</i> <i>krok</i>	Rotor se natočí na báзовou pozici a následně bude modul vyčkávat na nástupnou hranu signálu STEP při externím spouštění, kdy dojde k otočení o zadaný <i>krok</i> ve směru závislejícím na signálu DIR. V případě softwarového spouštění modul vyčkávat na příkaz DO STEP. Činnost vyčkávatí se opakuje do přijetí příkazu STOP nebo zmačknutí tlačítka STOP na modulu.
Udělej krok	DO STEP <i>směr</i>	Při softwarovém řízení provede otočení o předem stanovený krok se zadaným směrem <i>směr</i> .
Měření rychlosti	MSPD n <i>směr</i>	Rotor se roztočí zadanou rychlostí n v požadovaném směru <i>směr</i> . Každých 100 ms se vrací údaj o aktuální rychlosti do doby zmačknutí tlačítka STOP na modulu.
Zastavení	STOP	Zastavení krokování či měření rychlosti. Při měření rychlosti se předpokládá rychlé odeslání příkazu aspoň třikrát za sebou s adekvátním časovým odstupem (možná kolize s údajem o aktuální rychlosti).

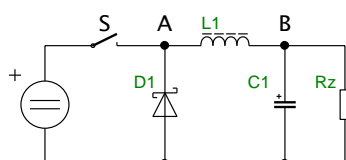
Tabulka 2.2 Rozsah hodnot atributů příkazů řídicího modulu.

Atribut	Název	Rozsah zadání
<i>krok</i>	Poziční krok	od 1 do 200 000 zadáváno v PPR
<i>n</i>	Rychlost otáčení	od 1 do 180 v ot/min
<i>N</i>	Počet vzorků záznamu	od 1 do 16 384 vzorků (dáno kapacitou SRAM a FFT analýzou)
<i>směr</i>	Smysl otáčení	1 pro kladný smysl ve směru hodinových ručiček, cokoliv jiného pak smysl opačný
<i>spouštění</i>	Typ spouštění provedení kroku	EXT – externí, SW – softwarový (PC aplikací)

3 VOLBA SOUČÁSTEK A OBVODOVÝ NÁVRH

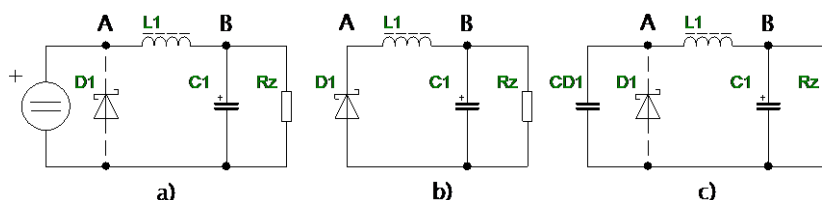
3.1 DC-DC snižující konvertor

Principiální zapojení snižujícího konvertoru též v anglické literatuře nazývaného *step-down* či *Buck converter* si lze prohlédnout na Obrázek 3.1. Principem tohoto zapojení je ve vhodných chvílích spínat spínač S, který je většinou řízen sofistikovaným integrovaným obvodem, a nabíjet tak cívku L1. Ta pak proudem dobíjí kondenzátor C1 a zároveň dodává proud do zátěže Rz. Při činnosti konvertoru se rozeznávají dva režimy. Režim se spojitým proudem tekoucím cívku nastává tehdy, kdy se kondenzátor C1 vybíjí rychleji než cívka. Proud cívku v tomto režimu osciluje kolem své střední hodnoty. Naopak má-li kondenzátor C1 velkou kapacitu, kdy po vybití cívky je schopen dodávat proud do zátěže, se hovoří o režimu nespojitého proudu cívku, protože proud cívku klesá na nulovou hodnotu [34].



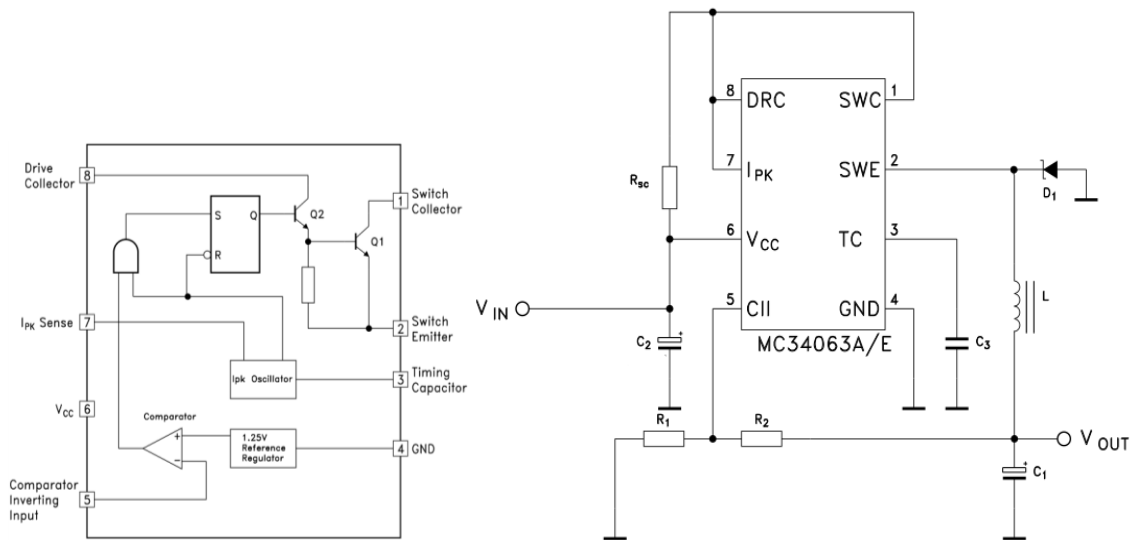
Obrázek 3.1 Základní zapojení snižujícího konvertoru.

Jednotlivé fáze činnosti konvertoru lze podle režimu rozkreslit až do třech náhradních schémát, které se nacházejí na Obrázek 3.2. Vždy jsou však pro oba dva režimy totožné první dvě fáze a) a b), kdy fáze b) je charakteristická u tohoto typu konvertoru zapojením Shottkyho diody, která uzavírá obvod v okamžicích rozepnutí spínače a umožňuje tak průtok proudu cívku. Pro režim nespojitého proudu cívku pak přibývá fáze c), kdy na kondenzátoru C1 je ještě dostatečné napětí, respektive je v něm naakumulovaná dostatečná energie na pokrytí energetické spotřeby zátěže. Kondenzátor CD1 modeluje parazitní kapacitu záporně polarizované diody D1 a způsobuje, že cívka na začátku této fáze kmitá na vysokém kmitočtu, což lze pozorovat osciloskopem v bodě A. Jakmile napětí na kondenzátoru C1 poklesne na stanovenou mez, sepne spínač a nastává fáze a). Celý děj se poté opakuje dokola, dokud není přerušeno napájení obvodu nebo v případě řízení sofistikovaným integrovaným obvodem není vyhodnoceno proudové přetížení.



Obrázek 3.2 Fáze spínání: a) nabíjení b) cívka zdrojem proudu c) vybíjení kondenzátoru C1.

Na trhu existuje mnoho specializovaných integrovaných obvodů pro realizaci DC/DC konvertorů. Pro potřeby v řídicím modulu byl vybrán univerzální řídicí integrovaný obvod MC34063AB od výrobce ST Microelectronics [16] jehož vnitřní blokové zapojení a typické aplikační zapojení jako snižující měnič si lze prohlédnout na následujícím obrázku.



Obrázek 3.3 Řídicí obvod s integrovaným spínačem MC34063 a jeho aplikace (převzato z [16]).

Jak je patrné z obrázku uvedeném výše, výstupní napětí se snímá pomocí napěťového děliče skládajícího se z rezistorů R_1 a R_2 . Výstup z napěťového děliče se přivádí na vstup komparátoru s určitou hystezí, který řídí RS klopný obvod budící integrovaný spínač skládající se z NPN bipolárních tranzistorů v Darlingtonově zapojení. Spínač je zároveň řízen oscilátorem, který se stará o to, aby spínač nebyl dlouho sepnut obzvláště při náběhu po přivedení napájení, což snižuje jeho výkonovou ztrátu. Obvod je ještě vybaven nadproudovou detekcí pomocí snímacího rezistoru R_{sc} zapojeného v sérii se spínačem. Objeví-li se na tomto rezistoru vyšší napětí jak zhruba 0,3 V, tak dojde k zablokování vnitřního oscilátoru, který zároveň zablokuje RS klopný obvod a ten rozpojí spínač.

Samotný návrh začíná výpočtem poměru doby sepnutí k době rozepnutí, jež je podle [16] dána požadovaným napětím na výstupu U_{OUT} , prahovým napětím Shottkyho diody v propustném směru U_{FD1} , minimálním napájecím napětím $U_{IN \min}$ a saturačním napětím spínače U_{sat} , které dle výrobce ve zvolené konfiguraci činí 1,3 V. Minimální napájecí napětí bylo stanoveno 29,3 V při napájení z 30 V zdroje napětí kvůli napěťovému úbytku na klasické křemíkové diodě 1N4007 implementované jako ochrana proti přepólování napájení. Shottkyho dioda byla vybrána MBRS2040LT3G od výrobce ON Semiconductor [35] jejíž maximální parametry jsou maximální závěrné napětí 40 V a maximální stejnosměrný proud v propustném směru 2 A. Střední proud 0,5 A Shottkyho diodou na ní způsobí maximální úbytek napětí $U_{FD1 \max}$ zhruba 0,35 V dle datového listu výrobce.

$$\frac{t_{ON}}{t_{OFF}} = \frac{U_{OUT} + U_{FD1 \max}}{U_{IN \min} - U_{sat \max} - U_{OUT}} = \frac{5 + 0,35}{29,3 - 1,3 - 5} = 0,2326$$

Ze vztahu popisující maximální periodu spínání T_{\max} (minimální spínací kmitočet) byl odvozen vztah (3.1) pro výpočet doby, po kterou je spínač vypnutý, pomocí poměru vypočteného v předchozím kroku. Tento vztah je pak dosazen zpět do rovnice pro výpočet maximální periody spínání, ze které je odvozen vztah (3.2) pro výpočet maximální doby sepnutí spínače $t_{\text{ON max}}$ při zvoleném minimálním spínacím kmitočtu f_{\min} .

$$T_{\max} = \frac{1}{f_{\min}} = t_{\text{ON max}} + t_{\text{OFF}} = t_{\text{OFF}} \cdot \left(\frac{t_{\text{ON}}}{t_{\text{OFF}}} + 1 \right) \Rightarrow t_{\text{OFF}} = \frac{1}{f_{\min} \cdot \left(\frac{t_{\text{ON}}}{t_{\text{OFF}}} + 1 \right)} \quad (3.1)$$

$$t_{\text{ON max}} = T_{\max} - t_{\text{OFF}} = \frac{1}{f_{\min}} - \frac{1}{f_{\min} \cdot \left(\frac{t_{\text{ON}}}{t_{\text{OFF}}} + 1 \right)} \quad (3.2)$$

Minimální spínací kmitočet konvertoru byl zvolen na horní hranici slyšitelného frekvenčního pásma u zdravého člověka a to 20 kHz. Poté maximální doba sepnutí spínače činí:

$$t_{\text{ON max}} = \frac{1}{20 \cdot 10^3} - \frac{1}{20 \cdot 10^3 \cdot (0,2326 + 1)} = 9,4353 \mu\text{s}.$$

Z maximální doby sepnutí se poté určí hodnota externího časovacího kondenzátoru C_3 podle vztahu uvedeném v datovém listu řídicího obvodu [16], který je nabíjen a vybíjen zdrojem konstantního proudu.

$$C_3 = 4,5 \cdot 10^{-5} \cdot t_{\text{ON max}} = 4,5 \cdot 10^{-5} \cdot 9,4353 \cdot 10^{-6} = 424,5885 \text{ pF} \Rightarrow 470 \text{ pF}$$

$$t_{\text{ON max}} = \frac{C_3}{4,5 \cdot 10^{-5}} = \frac{470 \cdot 10^{-12}}{4,5 \cdot 10^{-5}} = 10,4 \mu\text{s}$$

Z dostupných výrobní řad byl vybrán keramický kondenzátor o kapacitě 470 pF, kdy poté stoupne doba sepnutí spínače na cca 10,4 μs . Následuje výpočet minimální indukčnosti cívky L_{\min} , kdy její hodnota závisí na přiloženém napětí, špičkovém proudu $I_{\text{šp}}$ a maximální době sepnutí. Špičkový proud cívkou je volen jako dvojnásobek maximálního výstupního proudu. Na základě dosazení do vztahu (3.3) a výpočtu byla zvolena indukčnost akumulací cívky s feritovým jádrem 330 μH .

$$L_{\min} = \frac{U_{\text{IN min}} - U_{\text{sat}} - U_{\text{OUT}}}{I_{\text{šp}}} \cdot t_{\text{ON max}} = \frac{U_{\text{IN min}} - U_{\text{sat}} - U_{\text{OUT}}}{2 \cdot I_{\text{OUT max}}} \cdot t_{\text{ON max}} \quad (3.3)$$

$$L_{\min} = \frac{29,3 - 1,5 - 5}{2 \cdot 0,5} \cdot 10,4444 \cdot 10^{-6} = 238,1323 \mu\text{H} \Rightarrow 330 \mu\text{H}$$

Výstupní kondenzátor je určen na základě dovoleného zvlnění výstupního napětí způsobeného jednak reaktancí kondenzátoru a jednak jeho ekvivalentním sériovým odporem ESR, který by měl být co nejmenší. V tomto ohledu se proto počítá s tím, že největší vliv má právě reaktance kondenzátoru, kterou popisuje následující vztah pro výpočet minimální kapacity výstupního kondenzátoru dle [16].

$$C_{1\min} = \frac{I_{\text{šp}} \cdot (t_{\text{ON}} + t_{\text{OFF}})_{\text{max}}}{8 \cdot U_{\text{OUT zvlšp-šp}}} = \frac{2 \cdot I_{\text{OUT max}}}{8 \cdot U_{\text{OUT zvlšp-šp}} \cdot f_{\min}} \quad (3.4)$$

$$C_{1\min} = \frac{2 \cdot 0,5}{8 \cdot 0,05 \cdot 20 \cdot 10^3} = 125 \mu\text{F} \Rightarrow 220 \mu\text{F}$$

Dosazením maximálně dovoleného zvlnění 50 mV špička-špička (1 % z výstupního napětí 5 V) při proudovém odběru 0,5 A a předpokládaném minimálním spínacím kmitočtu 20 kHz vychází minimální kapacita výstupního kondenzátoru 125 μF . S jistou rezervou byl vybrán kondenzátor s kapacitou 220 μF na 16 V. Řídící obvod disponuje nadproudovou ochranou, která se nastavuje pomocí velikosti snímacího rezistoru R_{SC} . Výpočet jeho hodnoty je uveden níže, kdy výrobce udává minimální reakční hodnotu napětí na snímacím rezistoru $U_{\text{IPK min}} 0,3 \text{ V}$.

$$R_{\text{SC}} = \frac{U_{\text{IPK min}}}{I_{\text{šp}}} = \frac{U_{\text{IPK min}}}{2 \cdot I_{\text{OUT max}}} = \frac{0,3}{2 \cdot 0,5} = 0,3 \Omega$$

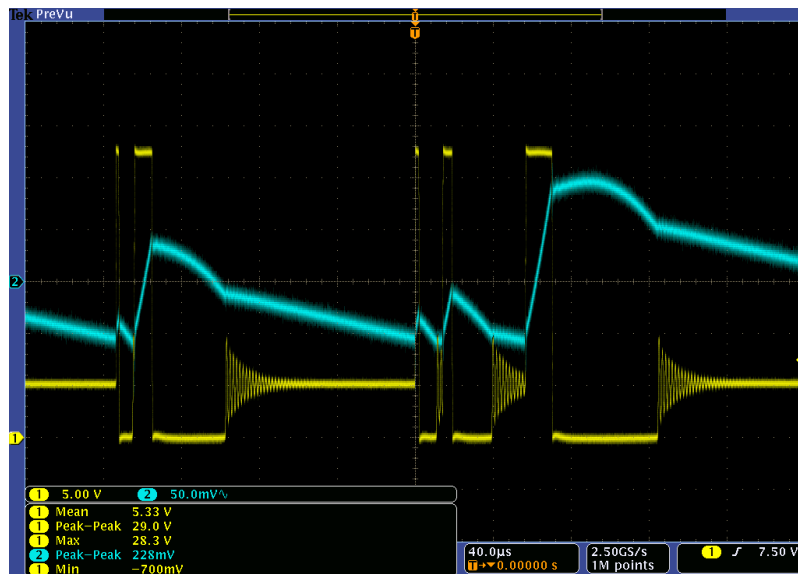
Posledním krokem návrhu DC/DC konvertoru je výpočet hodnot rezistorů napěťového děliče. Příčný proud děličem byl zvolen 1 mA a zanedbáním proudu tekoucím do vstupu komparátoru lze vypočítat hodnotu rezistoru R_1 dle následujícího vztahu:

$$R_1 = \frac{U_{\text{ref}}}{I_{R_{1,2}}} = \frac{1,25}{1 \cdot 10^{-3}} = 1,25 \text{ k}\Omega \Rightarrow 1,2 \text{ k}\Omega.$$

Následně ze známého vztahu popisující výstupní napětí napěťového děliče lze odvodit vztah pro výpočet hodnoty rezistoru R_2 opět za předpokladu nulového výstupního proudu navrhovaného děliče.

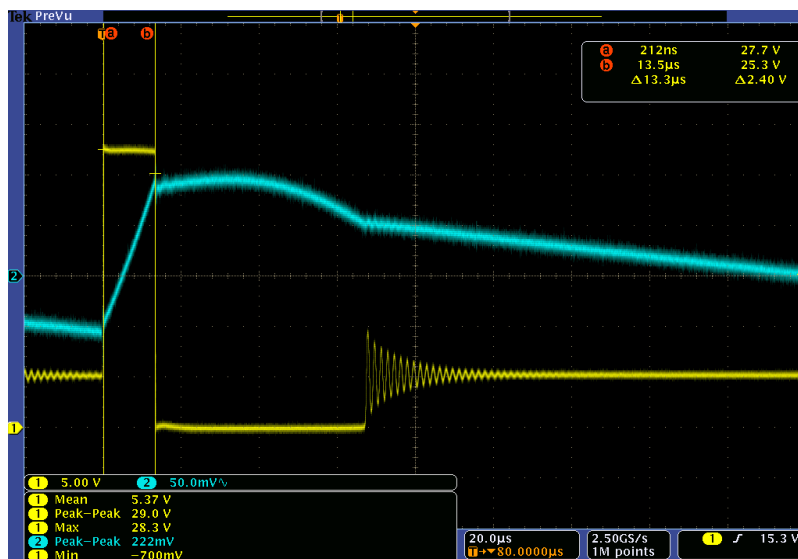
$$U_{\text{ref}} = U_{\text{OUT}} \cdot \frac{R_1}{R_1 + R_2} \Rightarrow R_2 = \left(\frac{U_{\text{OUT}}}{U_{\text{ref}}} - 1 \right) \cdot R_1 = \left(\frac{5}{1,25} - 1 \right) \cdot 1,2 = 3,6 \text{ k}\Omega$$

Tím je výpočet součástek DC/DC konvertoru u konce. Při návrhu plošného spoje se co nejlíže k napájecímu pinu řídicího integrovaného obvodu přidá keramický blokující kondenzátor 100 nF a také je dbáno na dodržení zásady co nejmenší proudové smyčky prvků Schottkyho dioda, cívka a výstupní kondenzátor pro minimalizování elektromagnetického rušení. Na výstup konvertoru je pak ještě do série se zátěží přidána feritová perla v SMD provedení s impedancí 1,5 k Ω při 100 MHz. Po realizaci řídicího modulu bylo provedeno kontrolní měření činnosti konvertoru pomocí osciloskopu, které si lze prohlédnout na následujícím obrázku. Podmínky měření byly vstupní napětí 29,4 V a proud zátěží cca 101 mA. Při tomto napájení spínaný zdroj odebíral proud zhruba 24 mA. Ze změřených hodnot byla poté určena účinnost zdroje, která činí 72 %.



Obrázek 3.4 Průběhy napětí před (žlutě, DC vazba) a za cívkou (modře, AC vazba).

Obrázek 3.5 ukazuje detail jedné spínací periody, jejíž fáze odpovídají režimu nespojitého proudu cívkou, jež byl uveden v úvodu. Maximální čas sepnutí byl změřen 13,5 μ s, který koresponduje s akceptovatelnou odchylkou s navrhovanou hodnotou 10,4 μ s.

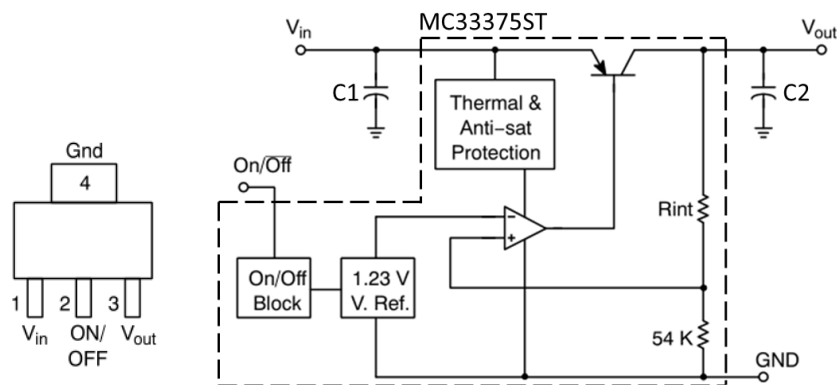


Obrázek 3.5 Detail jedné periody sepnutí konvertoru.

3.2 LDO napěťový regulátor

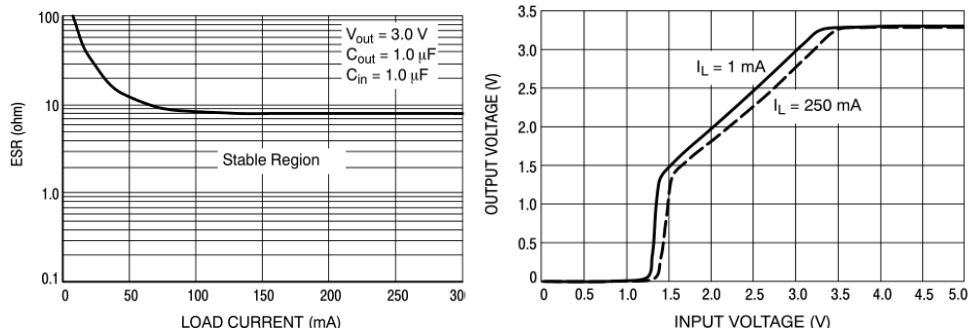
Řešení napájení digitální části řídicího modulu bylo zvoleno pomocí LDO napěťového regulátoru, anglicky *low-dropout regulator*, s výstupním napětím 3,3 V z důvodu nepoužitelnosti klasických napěťových regulátorů (stabilizátorů) jako je např. LM317, které pro svou správnou činnost potřebují minimální rozdíl napětí mezi vstupním a výstupním napětím 2 V. Tento typ se od klasických stabilizátorů liší zapojením regulačního tranzistoru, který zde není zapojen jako sledovač napětí tedy se společným kolektorem ale jako zesilovač čili zapojení se společným emitorem. Toto řešení kromě výhody nižšího rozdílu napětí mezi vstupem a výstupem však přináší problémy se stabilitou.

Ze široké nabídky byl vybrán integrovaný obvod MC33375ST od výrobce ON Semiconductor [33] s vyhovujícími parametry jako jsou maximální výstupní proud 300 mA, potlačení zvlnění vstupního napětí 65 dB a nepříliš velké nároky pro zajištění stability. Popis vývodů a základní zapojení je uvedeno na Obrázek 3.6.



Obrázek 3.6 Rozvržení vývodů a typické zapojení LDO MC33375ST (převzato z [33] a upraveno).

Výrobce doporučuje minimální kapacitu výstupního kondenzátoru 330 nF s nízkým ekvivalentním odporem ESR, který lze vyčíst z grafu závislosti minimálního ESR na proudu zátěži pro zaručení stability obvodu uvedeném na Obrázek 3.7. Zde je také ukázána převodní charakteristika LDO regulátoru, z které je možné vyčíst minimální úbytek na řídicím tranzistoru regulátoru v závislosti na zatěžovacím proudu.

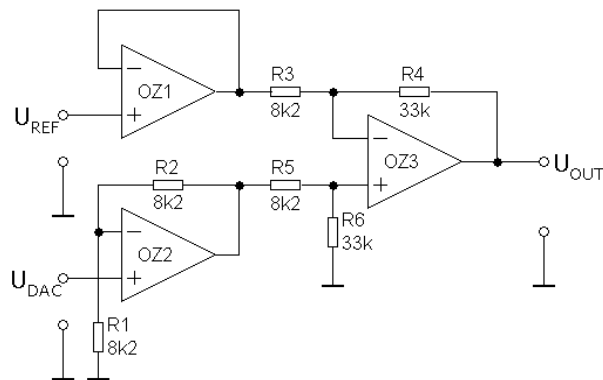


Obrázek 3.7 Maximální hodnota ESR výstupního kondenzátoru a převodní charakteristika LDO (převzato z [33]).

Pro realizaci zapojení byl zvolen vstupní a výstupní kondenzátor o kapacitě 330 nF s keramickým dielektrikem typu X7R. Kontrolním měřením na hotovém zařízení poté bylo ověřeno, že LDO regulátor na svém výstupu nekmitá.

3.3 Rozdílový zesilovač s napěťovou referencí

Z důvodů požadavku otáčení do obou směrů je nutné křížení záporné napětí. Výstup 12 bitového DA převodníku MCU disponuje však pouze kladným napětím v závislosti na vstupním datovém slově a velikosti napěťové reference volené 2,5 V s definovanou přesností. K získání rozsahu výstupního napětí obsahující záporné hodnoty je proto použit známý rozdílový zesilovač uvedený na Obrázek 3.8.



Obrázek 3.8 Návrh zapojení rozdílového zesilovače.

Samotný rozdílový zesilovač s OZ3 je na vstupech vybaven neinvertujícími zesilovači pro vyloučení vlivu vnitřních impedancí vstupních napěťových zdrojů. OZ1 slouží pouze pro zmíněné impedanční oddělení, kdežto OZ2 má navíc za úkol zesílit napětí z převodníku ideálně na dvojnásobek. Protože je cílem získat výstupní napětí v rozsahu ± 10 V musí také OZ3 zesilovat rozdíl vstupních napětí a to 4 krát.

Operační zesilovače byly vybrány s výstupem pracujícím v plném rozsahu napájecího napětí tzv. Rail-to-Rail NCS20074 od výrobce ON Semiconductor [32] s vlastní spotřebou maximálně 650 μ A na zesilovač a 3,2 MHz pásmem jednotkového zesílení při kapacitní zátěži 25 pF. V jednom pouzdře se nacházejí čtyři takovéto operační zesilovače, čímž celkový odběr poté činí 2,6 mA. Volba rezistorů je podmíněna s přihlédnutím na skutečnost, že obvod je napájen symetrickým referenčním napětím z lineárního servozesilovače firmy Mattke Ag [24] s proudovým omezením 5 mA. Proto byl stanoven maximální výstupní proud OZ2 0,5 mA, který určuje celkový zatěžovací odpor 10 k Ω při maximálním výstupním napětí 5 V tohoto operačního zesilovače. Hodnota zatěžovacího odporu odpovídá sérioparalelní kombinaci rezistorů R_1 , R_2 , R_5 a R_6 :

$$R_L = \frac{(R_1 + R_2) \cdot (R_5 + R_6)}{R_1 + R_2 + R_5 + R_6}. \quad (3.5)$$

Pro dostatečné potlačení souhlasné složky se volí stejné dvojice rezistorů R_4 , R_6 a R_3 , R_5 , které se dále pro zjednodušení označí jako R_a a R_b . Rezistory R_1 a R_2 jako R_c

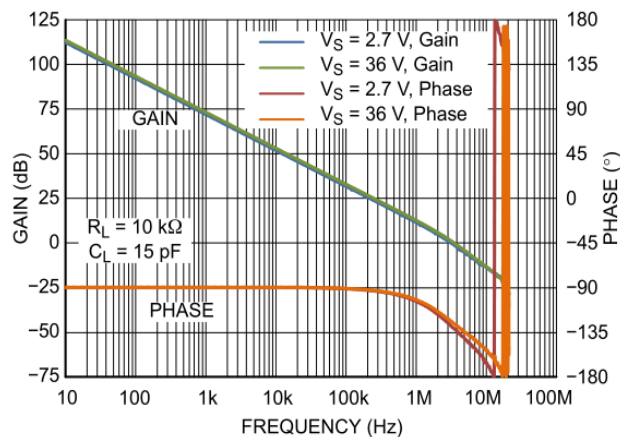
jsou voleny shodné velikosti zajišťující dvojnásobné zesílení zesilovače s OZ2. Tím se vztah (3.5) zjednoduší na tvar:

$$R_L = \frac{2 \cdot R_c \cdot 5 \cdot R_b}{2 \cdot R_c + 5 \cdot R_b} \quad (3.6)$$

Po zavedení další zjednodušující podmínky $R_b = R_c = R$ pro eliminaci dvou proměnných lze vzorec (3.6) upravit na vztah:

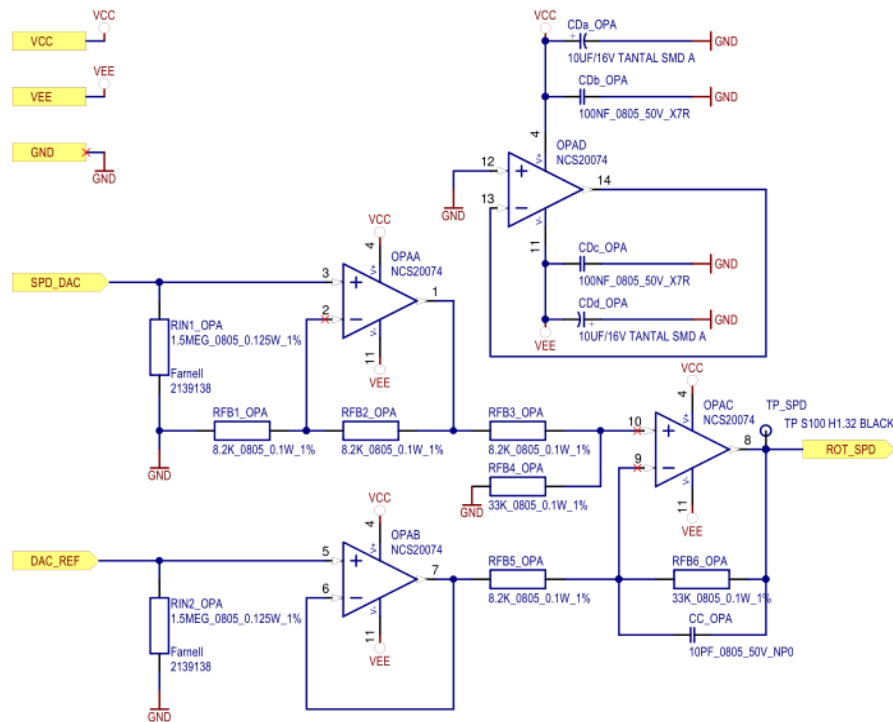
$$R_L = \frac{10}{7} \cdot R \quad (3.7)$$

Dle uvedených vztahů vychází minimální hodnota $R = 7 \text{ k}\Omega$ pro mezní zátěž $10 \text{ k}\Omega$. Rezistory skupin R_b , R_c voleny $8,2 \text{ k}\Omega$ a R_a $33 \text{ k}\Omega$ odpovídající čtyřnásobku R_b zajišťující požadované zesílení rozdílového zesilovače s chybou $0,6 \%$. Impedance napěťového vstupu servozesilovače [21] činí $25 \text{ k}\Omega$, kdy jakožto zátěž navrženého zesilovače způsobí při výstupním napětí 10 V jeho celkový proudový odběr asi $3,43 \text{ mA}$ tedy s dostatečnou rezervou vůči maximu 5 mA . Co se týče stability, není potřeba provádět frekvenční kompenzaci, protože je zesilovač s otevřenou smyčkou zpětné vazby stabilní v celém svém kmitočtovém rozsahu jak uvádí Obrázek 3.9 (fázový posuv menší než 180° při zesílení větším jak 0 dB).



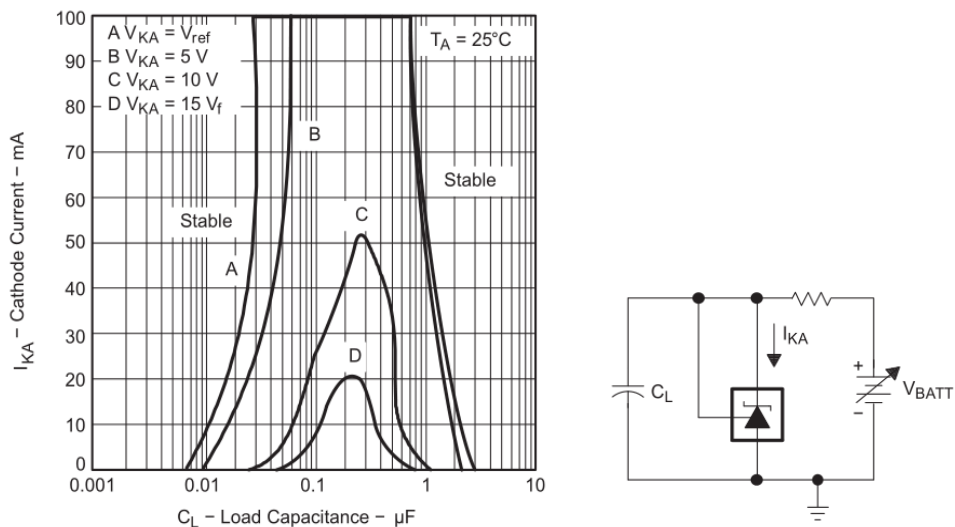
Obrázek 3.9 Frekvenční charakteristiky NCS20074 (převzato z [32]).

Konečné zapojení rozdílového zesilovače je uvedeno na Obrázek 3.10, kdy na vstupy byly přidány rezistory definující vstupní odpor $1,5 \text{ M}\Omega$ z požadavku minimální zátěže DA převodníku integrovaného v MCU pro přesnost výstupního napětí 1% s vypnutým výstupním sledovačem z důvodu nevyhovujících vlastností. Dále byl přidán kompenzační kondenzátor CC_OPA s kapacitou 10 pF paralelně k zpětnovazebnímu odboru rozdílového zesilovače pro preventivní opatření vůči záskmitům. Nevyužitý operační zesilovač integrovaného obvodu byl zapojen jako sledovač napětí a jeho vstup přizemněn, aby co nejméně ovlivňoval ostatní operační zesilovače. Nakonec se do napájení operačních zesilovačů přidaly 100 nF keramické blokující kondenzátory umístěné co nejbliž napájecím pinům integrovaného obvodu a co nejbliž k vstupním napájecím svorkám byly vloženy tantalové kondenzátory s kapacitou $10 \mu\text{F}$ dimenzované na napětí 16 V .



Obrázek 3.10 Finální zapojení rozdílového zesilovače (analogová část modulu).

Pro napěťovou referenci byla vybrána programovatelná Zenerova dioda TL431A firmy Texas Instruments [22] se základním výstupním napětím typicky 2,495 V s tolerancí 2 % při příčném proudu 10 mA. Dále výrobce zaručuje, že v celém pracovním rozsahu teplot se výstupní hodnota nezmění více než o 55 mV. Toto napětí slouží také jako reference 12 bitového DA převodníku integrovaného v MCU. Minimální proud napěťovou referencí je 1 mA a proto bude napájena z 5 V napájecí sběrnice přes sériový rezistor o velikosti 470 Ω (úbytek 2,5 V při proudu 5 mA). Graf uvedený na Obrázek 3.11 ukazuje oblasti stability nad danou křivkou výstupního napětí v závislosti na kapacitní zátěži pro případ dodatečného blokování výstupu, které podle výrobce není nutné.



Obrázek 3.11 Graf stability reference TL431A a typické zapojení (převzato z [22]).

3.4 Přijímač RS-422 a volba MCU

Jako přijímače signálů A, B a M standardu RS-422 z inkrementálního kodéru byl vybrán integrovaný obvod DS34LV86TM obsahující čtyři CMOS přijímače od firmy Texas Instruments [27] s celkovým statickým proudovým odběrem maximálně 15 mA (9 mA typicky) bez zátěže a s napájecím napětím 3,3 V. Doba propagace signálu ze vstupu na výstup přijímačů je nanejvýš 35 ns, což je více než dostačující. Výrobce dále uvádí maximální kmitočet překlápění 32 MHz a kompatibilitu s TTL úrovněmi.

Výběr MCU je dán požadavky aplikace modulu, mezi které patří dostatečná datová paměť, schopnost komunikovat s počítačem přes USB, možnost vysokého taktovacího kmitočtu pro dostatečně rychlou reakci na synchronizační signál, hardwarová podpora pro inkrementální kódér a digitálně analogový převodník (dále DAC) s vyhovujícím rozlišením. Pro tyto účely byl vybrán MCU STM32F407VET6 od výrobce ST Microelectronics [15], jehož parametry jsou shrnuty v následujícím odstavci.

Vybrané parametry STM32F407VET6:

- ✓ 512 KB Flash paměť programu
- ✓ 128 KB SRAM datové paměti rozšířené o 64 KB CCM (sdíleno s jádrem)
- ✓ dva 12 bitové DAC
- ✓ Ze 17 časovačů dva 32 bitové čítače s možností funkce jako pulzní čítač pro kvadrurní inkrementální kódér
- ✓ Maximální interní taktovací kmitočet 180 MHz získaný implementovanou PLL násobičkou z externího krystalu 4 až 26 MHz
- ✓ USB FS integrovaná fyzická vrstva

K pinu 6 výstupního portu PA použitého MCU je připojen přes bázový rezistor NPN bipolární tranzistor Q1 MMBTA05LT1G od výrobce ON Semiconductor s $h_{21EminQ1} = 100$ dle [36], který nastavuje nulový krouticí moment DC motorku. Maximální hodnota bázového rezistoru spínacího tranzistoru Q1 je určena následujícím odvozeným vztahem, kde $U_{MCU IOmax}$ je maximální výstupní napětí portu zvoleného mikrokontroléru dle [15], U_{BEQ1} je napětí mezi bází a emitorem vybraného tranzistoru, $R_{POT MOMENT}$ udává hodnotu odporu dráhy potenciometru sloužícího k nastavení krouticího momentu motorku uvedené v [21] a $U_{CC SERVO ZES}$ je hodnota pozitivního napájecího napětí servozesilovače.

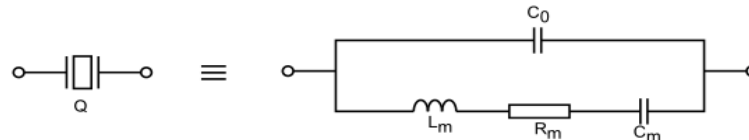
$$R_{BQ1max} = \frac{(U_{MCU IOmax} - U_{BEQ1}) \cdot R_{POT MOMENT}}{U_{CC SERVO ZES}} \cdot h_{21EminQ1} = \frac{(2,4 - 0,6) \cdot 10 \cdot 10^3}{30} \cdot 100$$

$$R_{BQ1max} = 60 \text{ k}\Omega \Rightarrow R_{BQ1} = \frac{R_{BQ1max}}{k} = \frac{60}{5} = 12 \text{ k}\Omega$$

Výsledná velikost bázového rezistoru byla vybrána s ohledem na zaručené vybudení bipolárního tranzistoru do saturace určeného pomocí činitele přebuzení k , který byl v tomto případě volen na horní mezi praktických hodnot, a to 5 [2].

3.5 Krystalový oscilátor MCU

Srdcem každého MCU je generátor hodinového signálu, který určuje rychlost zpracování jednotlivých instrukcí. Ve valné většině se ke generování tohoto signálu používá Piercův oscilátor, jehož rezonanční obvod tvoří křemenný výbrus odborně nazývaný krystalový rezonátor pracující na principu piezoelektrického jevu. Ten lze modelovat při uvažování jedné sériové rezonanční frekvence pomocí sériového ztrátového rezonančního obvodu a k němu paralelně připojeného kapacitoru, jehož kapacita je určena kapacitou elektrod a vývodů (viz Obrázek 3.12).



Obrázek 3.12 Náhradní zapojení krystalu (převzato z [17]).

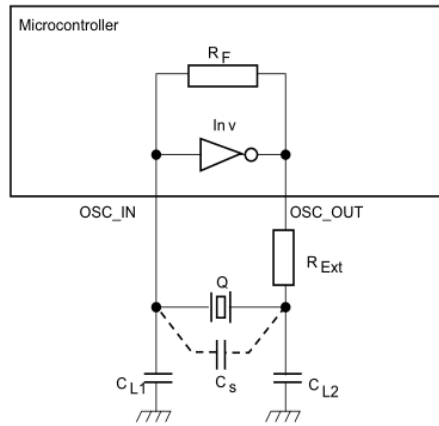
Krystalový rezonátor (dále krystal) rezonuje na několika sériových kmitočtech, kdy se pro případ návrhu oscilátoru zaměřuje na dominantních z nich popsánou následujícím Thompsonovým vztahem:

$$f_s = \frac{1}{2 \cdot \pi \cdot \sqrt{L_m \cdot C_m}}, \quad (3.8)$$

kde je L_m indukčnost charakterizující vibrační hmotnost a C_m kapacita popisující elasticitu krystalu. Dále se v náhradním zapojení vyskytuje pohybový odpor R_m určující ztráty v rezonančním obvodu a kapacita elektrod C_0 , ke které se při zapojení do oscilátoru přidá paralelně zatěžovací kapacita C_L . Díky kapacitám C_0 a C_L krystal kmitá na paralelní rezonanční frekvenci f_p , kterou lze v jistých mezích přeladovat podle vztahu (3.9).

$$f_p = f_s \cdot \left[1 + \frac{C_m}{2 \cdot (C_0 + C_L)} \right] \quad (3.9)$$

Typické zapojení Piercova oscilátoru v MCU si lze prohlédnout na Obrázek 3.13. Tvoří jej invertor zapojený jako zesilovač s definovaným ziskem díky rezistoru R_F zavádějící silnou zápornou zpětnou vazbu, ke kterému se paralelně přidá rezonanční obvod ve formě krystalu se zatěžovací kapacitou tvořenou kondenzátory C_{L1} , C_{L2} a parazitní kapacitou C_S tvořenou kapacitou plošného spoje a pinů oscilátoru MCU. Ke krystalu lze do série připojit rezistor R_{EXT} , který snižuje namáhání krystalu popsáným jeho výkonovou ztrátou v odborné literatuře udávanou jako Driving level DL a zároveň zhoršuje rozběh oscilátoru zmenšením zisku zesilovače což bude probráno v následujících odstavcích.



Obrázek 3.13 Zapojení oscilátoru MCU (převzato z [17]).

Krystal je zatížen zatěžovací kapacitou C_L , jejíž hodnotu a případně rozsah hodnot doporučuje výrobce krystalu. V případě zvoleného krystalu LFX TAL026380 řady 12SMX s paralelním rezonančním kmitočtem 8 MHz výrobce doporučuje hodnotu zatěžovací kapacity 16 pF [31]. Celková hodnota zatěžovací kapacity je dána vztahem (3.10) pomocí kterého se vypočítá kapacita kondenzátorů C_{L1} a C_{L2} . Ty jsou voleny shodné kapacity 18 pF za předpokladu, že celková parazitní kapacita je zhruba 7 pF podle [17].

$$C_L = \frac{C_{L1} \cdot C_{L2}}{C_{L1} + C_{L2}} + C_S \quad (3.10)$$

$$C_L - C_S = \frac{C_{L1} \cdot C_{L2}}{C_{L1} + C_{L2}} \Rightarrow C_{L1} = C_{L2} = 2 \cdot (C_L - C_S) = 2 \cdot (16 - 7) = 18 \text{ pF}$$

Po určení kapacit zatěžovacích kondenzátorů je nutné ověřit, zda-li se oscilátor rozkmitá či nikoliv. K tomu je nutné dodržet dvě oscilační podmínky, z nichž první, fázová, je u krystalových oscilátorů dodržena automaticky. Druhá podmínka říká, že je nutné vykompenzovat ztráty v rezonančním obvodu. To se v tomto případě provádí pomocí zesilovače, který musí mít adekvátní hodnotu zisku G . Dostatečnou hodnotu zisku lze ověřit pomocí následujících vztahů pro výpočet kritické přenosové vodivosti krystalu $g_{m \text{ kritické}}$ (3.11) a tzv. jistoty v zisku G_{jistota} (3.12), která by měla být podle [17] větší jak 5. K tomuto ověření je nutné zjistit sériový ekvivalentní odpor krystalu ESR_{krystal} , který činí 80 Ω dle datového listu výrobce [31].

$$g_{m \text{ kritické}} = 4 \cdot ESR_{\text{krystal}} \cdot (2 \cdot \pi \cdot f_{\text{OSC}})^2 \cdot (C_0 + C_L)^2 \quad (3.11)$$

$$g_{m \text{ kritické}} = 4 \cdot 80 \cdot (2 \cdot \pi \cdot 8 \cdot 10^6)^2 \cdot (7 \cdot 10^{-12} + 16 \cdot 10^{-12})^2 = 427,7060 \mu\text{S}$$

$$G_{\text{jistota}} = \frac{g_{m \text{ MCU OSC min}}}{g_{m \text{ kritické}}} > 5 \quad (3.12)$$

$$G_{\text{jistota}} = \frac{5 \cdot 10^{-3}}{427,706 \cdot 10^{-6}} = 11,69$$

Výpočtem bylo zjištěno, že se s určitou rezervou oscilátor rozkmitá. Na závěr je nutné zkontrolovat, zdali nebude krystal příliš mechanicky namáhán či jinými slovy nedojde k jeho přebuzení. K tomuto účelu slouží parametr DL_{krystal} krystalu určující jeho výkonové zatížení, které lze vypočítat dle odvozeného vztahu (3.13) vycházejícího ze vzorců pro elektrický výkon a reaktanci kondenzátoru. V upraveném vztahu (3.14) vystupuje kromě celkové zatěžovací kapacity, oscilačního kmitočtu a ekvivalentního sériového odporu krystalu také rozkmit výstupního napětí oscilátoru, které bylo odhadnuto na hodnotu 3 V. Výrobce specifikuje mezní výkonové zatížení krystalu 100 μW [31].

$$DL_{\text{krystal}} = ESR_{\text{krystal}} \cdot I_{\text{krystal}}^2 = ESR_{\text{krystal}} \cdot [U_{\text{ef}} \cdot 2 \cdot \pi \cdot f_{\text{OSC}} \cdot (C_L + C_0)]^2 \quad (3.13)$$

$$U_{\text{ef}} = \frac{U_{\text{p-p}}}{2 \cdot \sqrt{2}} \Rightarrow DL_{\text{krystal}} = ESR_{\text{krystal}} \cdot \frac{[U_{\text{p-p}} \cdot \pi \cdot f_{\text{OSC}} \cdot (C_L + C_0)]^2}{2} \quad (3.14)$$

$$DL_{\text{krystal}} = 80 \cdot \frac{[3 \cdot \pi \cdot 8 \cdot 10^6 \cdot (16 \cdot 10^{-12} + 7 \cdot 10^{-12})]^2}{2} = 120,2923 \mu\text{W}$$

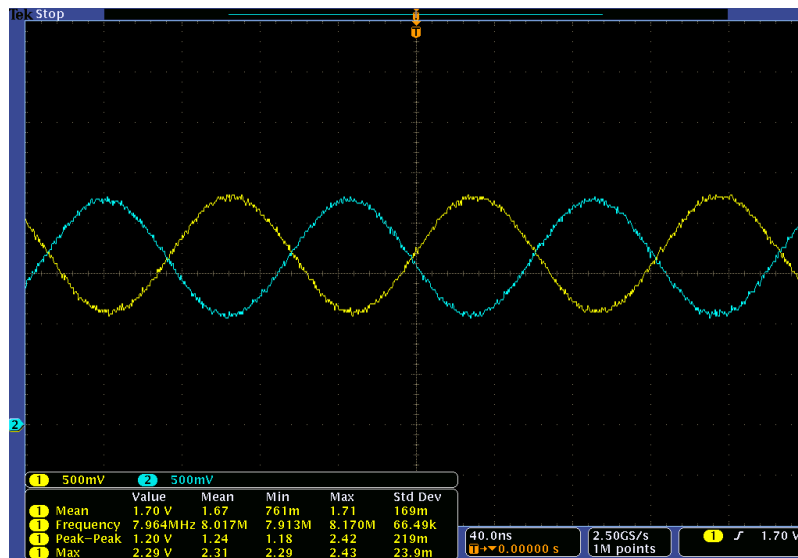
Dle výpočtu se dochází k závěru, že krystal v uvedeném oscilátoru bude lehce přetížen. Pro snížení výkonové zátěže krystalu se do série s krystalem přidal omezující rezistor R_{EXT} . Po této změně se zvětší ekvivalentní sériový odpor krystalu o hodnotu tohoto rezistoru a pro kritickou přenosovou vodivost lze poté psát rovnici (3.15), ze které se odvodí vztah (3.16) pro určení maximální hodnoty omezujícího rezistoru za uvažování maximální kritické přenosové vodivosti $g_{\text{m kritické max}}$ dané výrobcem MCU (1 mA/V) [15]. Hodnota omezujícího rezistoru byla vybrána 82 Ω .

$$g_{\text{m kritické}} = 4 \cdot (ESR_{\text{krystal}} + R_{\text{EXT}}) \cdot (2 \cdot \pi \cdot f_{\text{OSC}})^2 \cdot (C_0 + C_L)^2 \quad (3.15)$$

$$R_{\text{EXT max}} = \frac{g_{\text{m kritické max}}}{4 \cdot (2 \cdot \pi \cdot f_{\text{OSC}})^2 \cdot (C_0 + C_L)^2} - ESR_{\text{krystal}} \quad (3.16)$$

$$R_{\text{EXT max}} = \frac{1 \cdot 10^{-3}}{4 \cdot (2 \cdot \pi \cdot 8 \cdot 10^6)^2 \cdot (7 \cdot 10^{-12} + 16 \cdot 10^{-12})^2} - 80 = 107,0444 \Omega \Rightarrow 82 \Omega$$

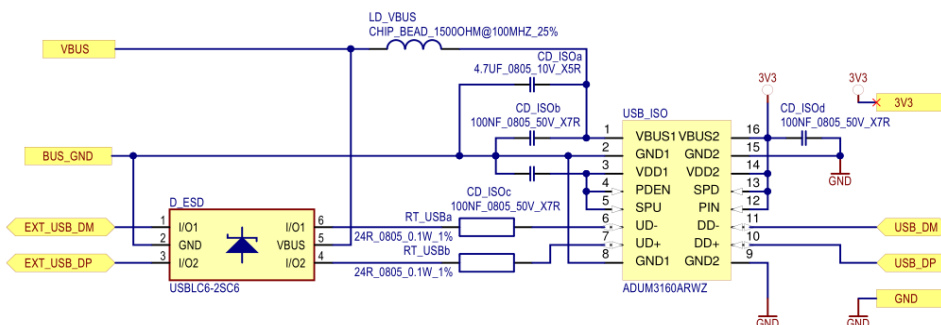
Výsledné průběhy napětí na vstupu a výstupu krystalového rezonátoru zobrazuje Obrázek 3.14. Signály byly změřeny pomocí osciloskopu firmy Tektronix s pasivními sondami o vstupním odporu 10 MΩ s paralelní kapacitou 8 pF což způsobilo menší rozladění rezonátoru a výstupní kmitočet byl tak 7,964 MHz s rozkmitem 1,2 V. Tvarové zkreslení nebylo pozorováno, což odpovídá korektnímu buzení rezonátoru.



Obrázek 3.14 Průběh napětí na vstupu a výstupu krystalu.

3.6 Obvodové řešení komunikačního rozhraní

Přestože se řídicí modul bude používat v laboratorních podmínkách, je od něj očekávána jistá odolnost vůči rušení, které mohou pocházet od okolních pracovišť, kdy je také nutné eliminovat zemní smyčky obzvláště při propojení se stolním PC. Proto byly na USB komunikačním rozhraní s PC implementovány funkce ochrany proti elektrostatickému výboji označovaném jako ESD protekce a galvanické oddělení. Pro ESD protekci byl vybrán integrovaný obvod USBLC6-2 od výrobce ST Microelectronics obsahující 4 ESD diody pro dvě datové linky a jednu Zenerovu diodu pro ochranu napájecí sběrnice proti přepětí [18]. Galvanické oddělení je provedeno pomocí specializovaného integrovaného obvodu ADUM3160 pro USB FS sběrnici od výrobce Analog Devices pracující na mikrovlnném principu oddělení [1]. Finální zapojení protekcí USB sběrnice si lze prohlédnout na následujícím obrázku.



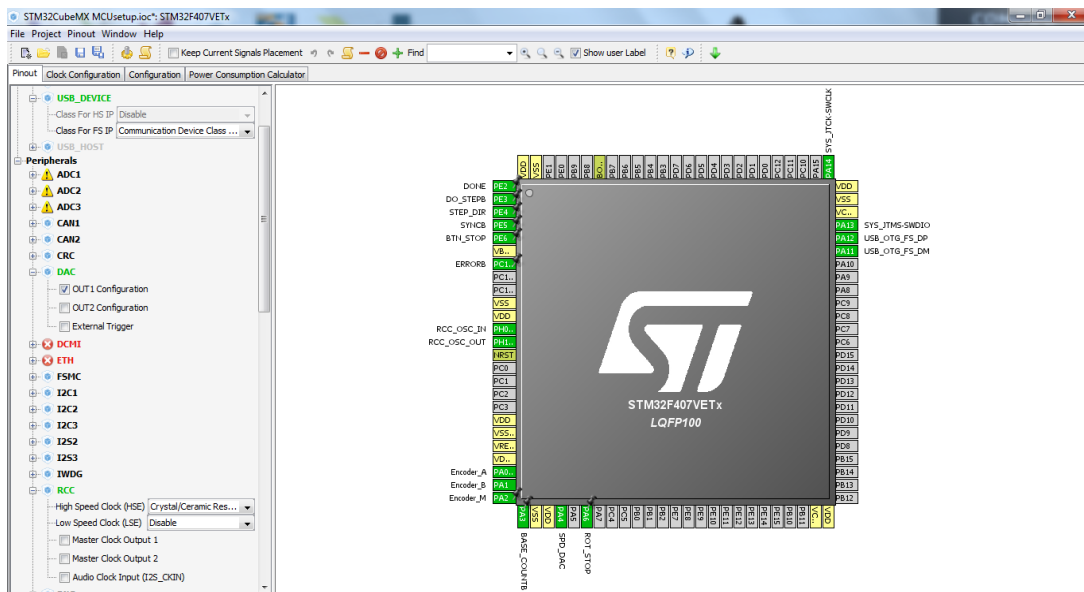
Obrázek 3.15 Ochranné zapojení USB rozhraní.

4 FIRMWARE

Součástí každého MCU je program, který je odborně označován jako firmware a bez kterého by MCU byl jako tělo bez duše. V dnešní moderní době při neustálém vývoji digitální techniky rostou i nároky na její ovládání. Naštěstí se také vyvíjejí užitečné nástroje na zvládnutí této problematiky, pomocí kterých je vývoj firmware o něco jednodušší. Následující kapitoly ukazují použití dvou takových užitečných nástrojů a popisují činnost jednotlivých částí firmware řídicího modulu, který je uveden v příloženém CD.

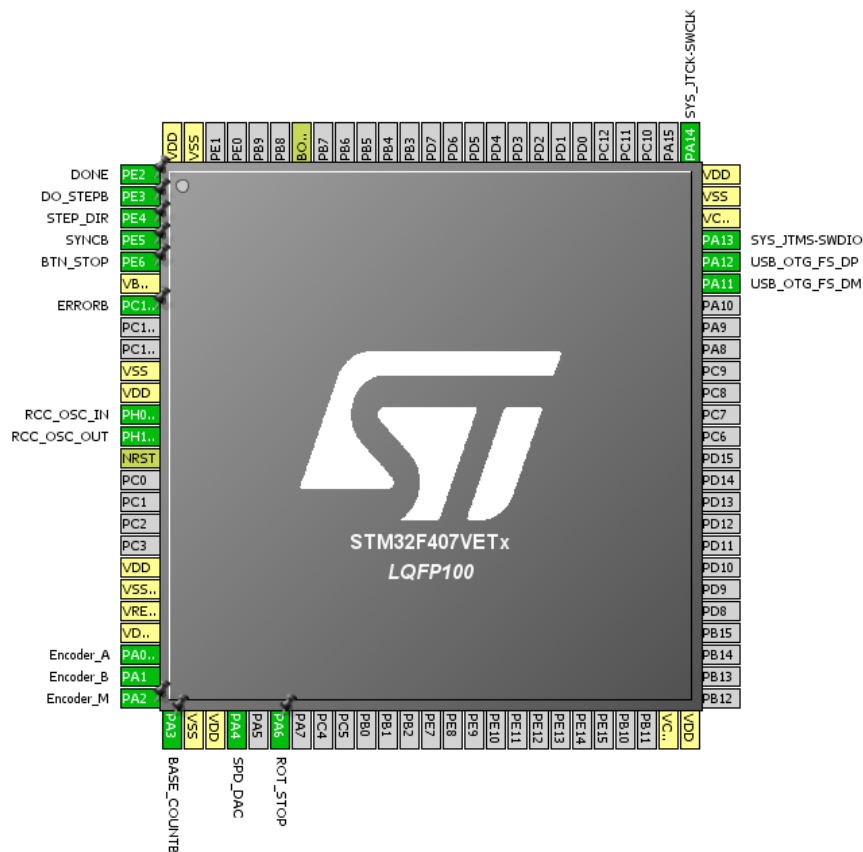
4.1 Nastavení periférií MCU a vývojové prostředí

První z užitečných nástrojů při vývoji firmware pro MCU od výrobce ST Microelectronics je program STM32CubeMX pro konfiguraci MCU, který si je možné po registraci zdarma stáhnout na stránkách výrobce [20]. V tomto programu lze jednoduše vybrat rodinu a typ MCU, pro který má být firmware vyvíjen. Poté si je možné pomocí intuitivního GUI (uživatelské grafické rozhraní), které si lze prohlédnout na Obrázek 4.1, nadefinovat vstupně výstupní piny a nastavit zvolené periférie podle požadavků aplikace.



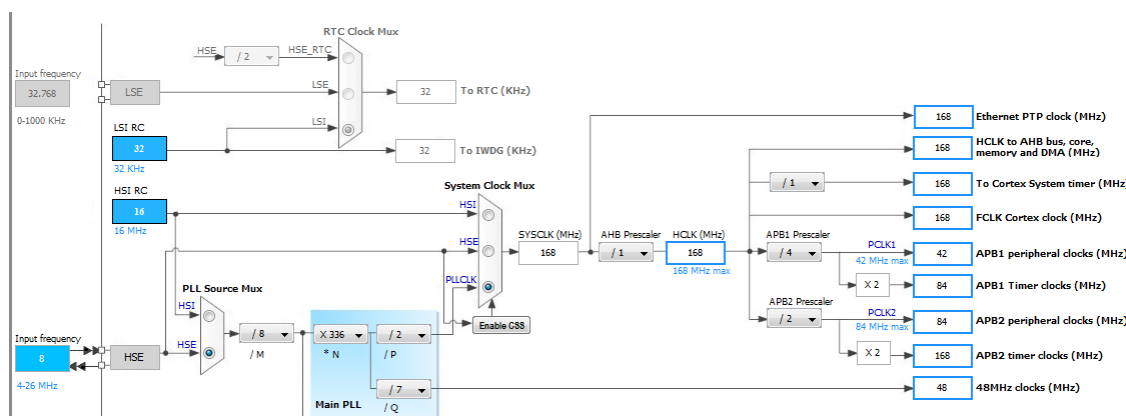
Obrázek 4.1 Konfigurační prostředí STM32CubeMX.

Nastavení vývodů MCU splňující požadavky pro aplikaci v řídicím modulu je uvedeno na Obrázek 4.2, kde je také vidět napájecí piny, resetovací vstup NRST, a signál BOOT0, pomocí kterého se vybírá, z jaké paměťové oblasti má být načten program, respektive z jaké adresy. MCU s architekturou ARM je totiž založen na Harvardské architektuře, čili sdílí adresní prostor s datovou pamětí, programovou pamětí a perifériemi. Uzemní-li se tento pin, je program načítán a nahráván do FLASH paměti. Vstup NRST reaguje na sestupnou hranu a využívá interní zvedací (pull-up) rezistor což zjednodušuje návrh plošného spoje [15].



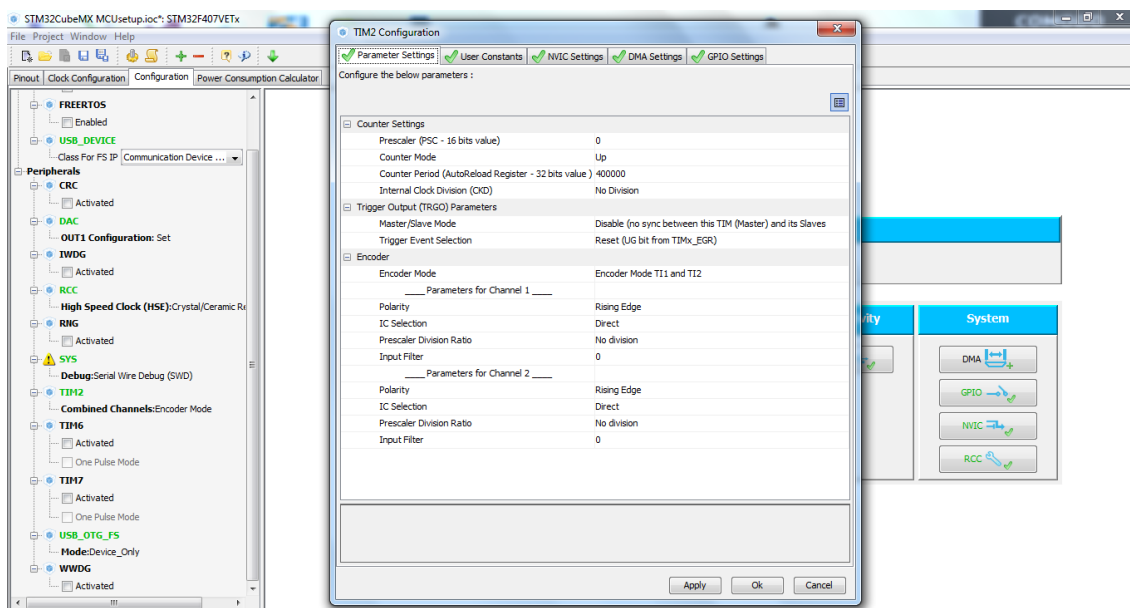
Obrázek 4.2 Rozložení použitých pinů MCU.

Další užitečnou funkcí konfiguračního programu je nastavení vnitřní PLL násobičky a děličky pro generování hodinových signálů periférií a jádra MCU. Nastavení se provádí vybráním vhodného koeficientu násobičky nebo děličky, kdy po zvolení dojde k automatickému přepočítání kmitočtů hodinových signálů a následně ke kontrole, zdali jsou kmitočty ve specifikovaném rozsahu hodnot zvoleného MCU. Pro navrhovaný firmware je nastavení časování ukázáno na Obrázek 4.3.



Obrázek 4.3 Nastavení PLL a děliček hodinových signálů.

Obrázek 4.4 ukazuje další funkci konfiguračního programu a to nastavování jednotlivých periférií, v tomto případě čítače/časovače TIM2 ve funkci čítače pulzů inkrementálního kodéru. Zda-li bude čítač reagovat pouze na nástupné nebo na nástupné i na sestupné hrany kvadrurního signálu se volí v položce Encoder Mode a pro volbu reakce na obě hrany se tento mód jmenuje Encoder Mode TI1 and TI2. V tomto módu je pak matoucí, že musí v nastavení jednotlivých kanálů čítače zůstat volba Polarity na Rising Edge podle referenčního manuálu MCU [14]. V případě zvolení Both Edges nepracuje čítač korektně.



Obrázek 4.4 Nastavení čítače/časovače TIM2.

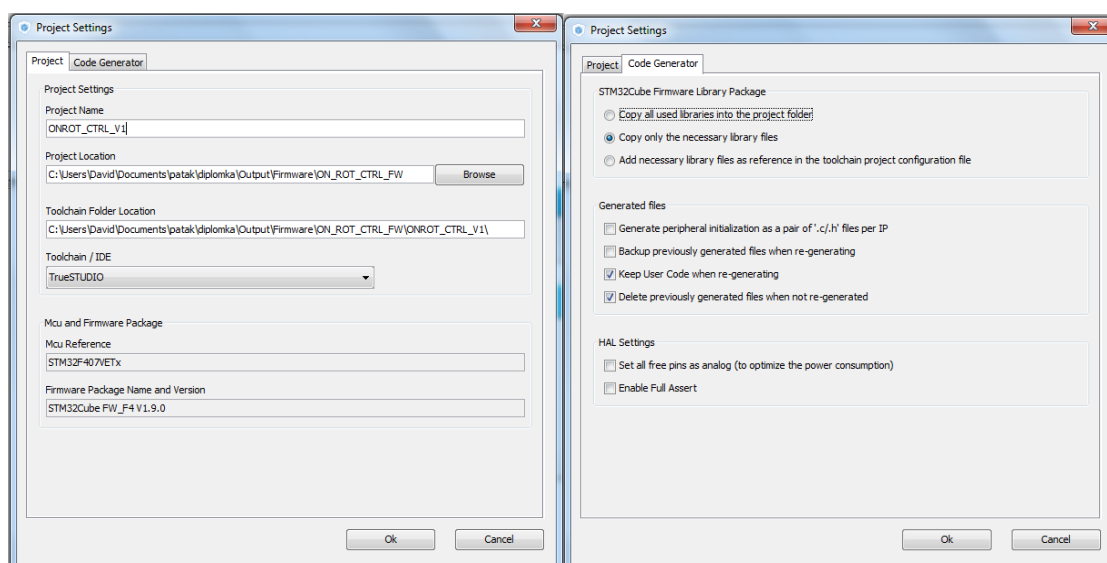
V záložce Configuration se také nalézá detailní nastavení vstupně výstupních pinů (GPIO), které je shrnuto v následující tabulce, kde PU znamená aktivovaný pull-up a PD aktivovaný pull-down interní rezistor.

Tabulka 4.1 Nastavení GPIO pinů MCU.

Signál	MCU pin	Směr	Funkce pinu	PU/PD/Vysoká impedance
A	PA0	vstup	TIMER2 CH1	PD
B	PA1	vstup	TIMER2 CH2	PD
M	PA2	vstup	IRQ2, nástupná hrana	PD
BASE_COUNTB	PA3	výstup	PUSH/PULL	vysoká impedance
SPD_DAC	PA4	výstup	DAC CH1	vysoká impedance
ROT_STOP	PA6	výstup	PUSH/PULL	vysoká impedance
USB_DM	PA11	obousměrný	USB OTG FS	N/A
USB_DP	PA12	obousměrný	USB OTG FS	N/A
SWDIO	PA13	obousměrný	Programování a ladění	PU
SWCLK	PA14	vstup	Hodinový signál SWD	PU

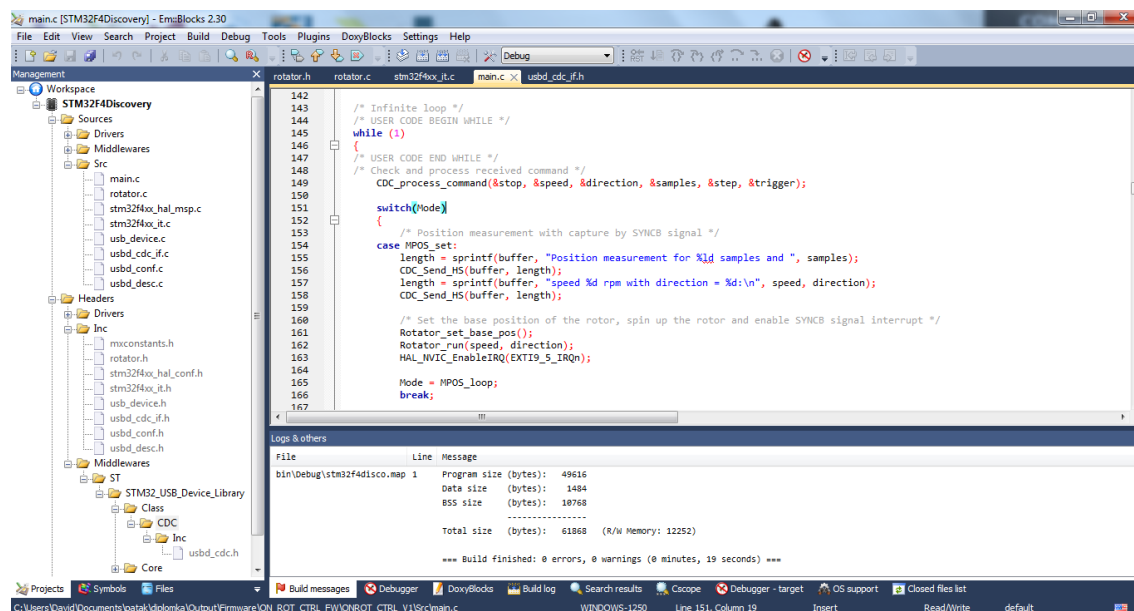
Signál	MCU pin	Směr	Funkce pinu	PU/PD/Vysoká impedance
ERRORB	PC13	výstup	PUSH/PULL	vysoká impedance
DONE	PE2	výstup	PUSH/PULL	vysoká impedance
DO_STEPB	PE3	vstup	IRQ3, sestupná hrana	PU
STEP_DIR	PE4	vstup	statický	vysoká impedance
SYNCB	PE5	vstup	IRQ5, sestupná hrana	PU
BTN_STOP	PE6	vstup	IRQ6, nástupná hrana	PD

Po provedení nastavení periférií program umožňuje vygenerovat projekt, do kterého přidá potřebné knihovny pro práci se zvolenými perifériemi. Tyto knihovny obsahují uživatelské sekce kódu, které se při opětovném generování kódu programem nepřepíší. Program také připraví kostru kódu hlavního programu, jež se skládá z inicializace zvolených periférií a tzv. Middle Wares, v tomto případě bylo vybráno CDC (Communication Device Class) pro virtuální sériový port na USB rozhraní, a nekonečné smyčky. Nastavení generování projektu obsahuje výběr balíku nástrojů pro vývoj (tzv. Toolchain), zde vybráno TrueSTUDIO, zvolení zkopírování do projektu jen nejnútnejších knihoven a zbytek byl ponechán v defaultním nastavení. Celkové nastavení projektu je uvedeno na následujícím obrázku.



Obrázek 4.5 Nastavení vlastností generovaného projektu.

Vygenerované soubory z konfiguračního programu byly poté přidány mezi zdrojové soubory přichystaného prázdného projektu ve vývojovém prostředí Em::Blocks obsahující linker skript, definice symbolů nutné pro HAL knihovny a include adresáře z předmětu MPOA (Mikroprocesory s architekturou ARM) vyučovaném na ústavu radioelektroniky. GUI vývojového prostředí ukazuje Obrázek 4.6.



Obrázek 4.6 Vývojové prostředí Em::Blocks 2.30.

Protože jsou ve firmware použity funkce pro datové typy float a USB enumerace, bylo nutné navýšit velikost zásobníku (stack), do kterého se ukládají návratové adresy při vykonávání podprogramů, a haldy (heap). Tyto velikosti jsou definovány v linker skriptu a jejich nastavení je uvedeno v následující ukázce kódu.

```
...
/* Entry Point */
ENTRY(Reset_Handler)

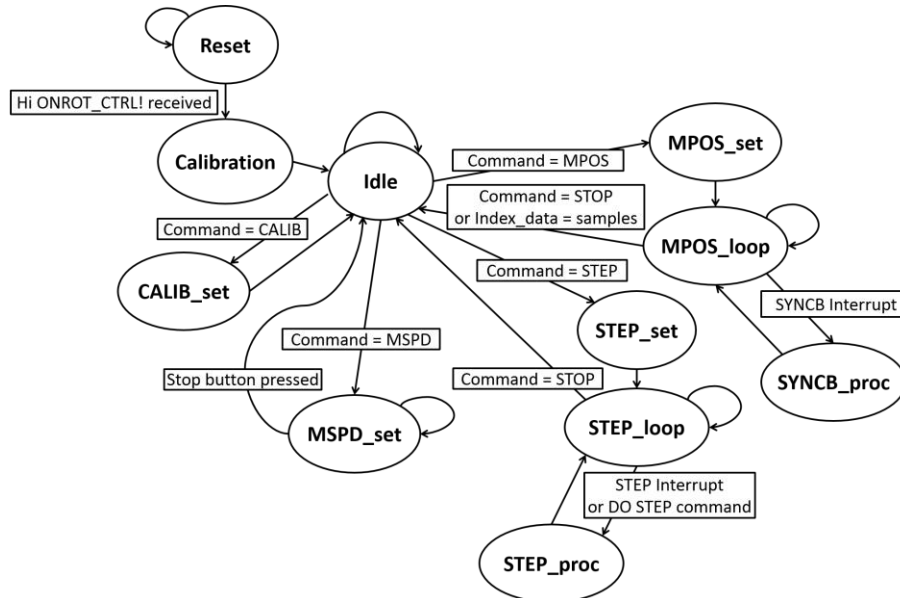
/* Highest address of the user mode stack */
_estack = 0x20020000;          /* end of RAM */
/* Generate a link error if heap and stack don't fit into RAM */
_Min_Heap_Size = 0x1000;     /* required amount of heap */
_Min_Stack_Size = 0x1000;    /* required amount of stack */

/* Specify the memory areas */
MEMORY
{
FLASH (rx)      : ORIGIN = 0x8000000, LENGTH = 512K
RAM (xrw)       : ORIGIN = 0x20000000, LENGTH = 128K
CCMRAM (rw)    : ORIGIN = 0x10000000, LENGTH = 64K
}
...
```

A na závěr pro korektní činnost funkce `printf` s proměnnými typu float bylo nutné přidat do nastavení kompilátoru nacházející se v kontextovém menu Project - Build options... a v okně Project build options zvolit záložku Other options v nastavení kompilátoru (Compiler settings) a připsat direktivu `-u _printf_float`.

4.2 Popis programu

Program pro MCU se skládá z hlavního programu obsahující hlavní nekonečnou smyčku tzv. stavového automatu s konstrukcí přepínače testující změnu stavu. Stavový digram popisující podmínky přechodu z jednoho do druhého stavu je uvedený na Obrázek 4.7. V závislosti na aktuálním stavu se volají příslušné funkce a procedury, které provádějí požadovanou činnost řídicího modulu.



Obrázek 4.7 Stavový diagram MCU programu.

V případě externího přerušení provádění programu se zavolá příslušný podprogram pro obsluhu přerušení, který změní proměnnou stav, jež se vyhodnocuje v hlavním programu. Většina funkcí je volána z knihovny HAL zaručující správné nastavení velkého množství registrů MCU a také čtení čímž je práce s periferiemi MCU o trochu snadnější. Specifičtější funkce pro obsluhu rotačního mechanického systému, jako je roztočení rotoru, krokování pozice apod., jsou napsány v oddělené knihovně s názvem *rotator*. Zdrojový soubor knihovny se poté přidá do hlavního programu pomocí direktivy `#include` pro preprocesor.

4.3 Kalibrace

První z funkcí MCU je zjištění dvou důležitých parametrů pro řízení rotačního systému a to konstanty úměrnosti otáček k napětí zde reprezentovanému jako hodnota digitálně analogového převodníku *DAC* a nulové otáčky n_{offset} způsobené napětíovou nesymetrií. Zde si je nutné uvědomit, že maximální hodnota vstupního slova u 12 bitového převodníku činí 4095. Tato hodnota způsobí na výstupu navrženého rozdílového zesilovače v ideálním případě napětí 10 V. Naopak, při nulové hodnotě se na výstupu objeví -10 V. Výstup DA převodníku je popsán známým vztahem:

$$U_{DAC} = U_{ref} \frac{DAC}{2^m - 1}, \quad (4.1)$$

kde U_{ref} je referenční napětí a m je rozlišení převodníku. Dosazením vztahu (4.1) do vztahu popisujícího výstupní napětí navrženého rozdílového zesilovače v závislosti na vstupním napětí se získává rovnice popisující řídicí napětí servozesilovače:

$$U_{\text{SPD}} = (U_{\text{DAC}} \cdot A_{\text{UOZ2}} - U_{\text{ref}}) \cdot A_{\text{UOZ3}} = \left(A_{\text{UOZ2}} \cdot \frac{\text{DAC}}{2^m - 1} - 1 \right) \cdot A_{\text{UOZ3}} \cdot U_{\text{ref}}, \quad (4.2)$$

kde A_{UOZx} jsou zesílení zesilovačů s příslušnými OZ tvořících rozdílový zesilovač, které v ideálním případě nabývají hodnot $A_{\text{UOZ2}} = 2$, $A_{\text{UOZ3}} = 4$. Pro otáčky rotoru mechanického systému n platí následující vztah (4.3), který bere v potaz jistou vstupní napěťovou nesymetrii U_{offset} . Dosazením rovnice (4.2) do vztahu (4.3) se dostává rovnice (4.4) popisující otáčky v závislosti na nastavené hodnotě digitálně analogového převodníku, kde GR je převodový poměr převodovky DC motorku.

$$n = \frac{kn}{GR} \cdot (U_{\text{SPD}} + U_{\text{offset}}) = \frac{kn}{GR} \cdot \left(\frac{\text{DAC}}{2^m - 1} \cdot A_{\text{UOZ2}} - 1 \right) \cdot A_{\text{UOZ3}} \cdot U_{\text{ref}} + n_{\text{offset}} \quad (4.3)$$

$$\frac{n - n_{\text{offset}}}{A_{\text{UOZ3}} \cdot U_{\text{ref}}} = \frac{kn}{GR} \cdot \left(\frac{\text{DAC}}{2^m - 1} \cdot A_{\text{UOZ2}} - 1 \right) \quad (4.4)$$

Úpravou rovnice (4.4) na vztah (4.5) je účelem vyjádřit hodnotu digitálně analogového převodníku pro požadované otáčky rotoru. Substitucí většiny konstantních hodnot zahrnutých do konstant A a B lze pro hodnotu DA převodníku psát rovnici (4.6), která se upraví na vztah (4.7) zavedením tzv. digitální převodní konstanty kn_{digital} vyjadřující změnu otáček rotoru na změnu hodnoty DA převodníku o 1 kód. Tato hodnota se právě vyhodnocuje při procesu kalibrace a vypočítává se pomocí vztahu (4.8). Konstanta B v tomto případě nabývá poloviční hodnoty rozsahu kódu DA převodníku.

$$\frac{(n - n_{\text{offset}}) \cdot GR}{A_{\text{UOZ2}} \cdot A_{\text{UOZ3}} \cdot U_{\text{ref}} \cdot kn} + \frac{1}{A_{\text{UOZ2}}} = \frac{\text{DAC}}{2^m - 1} \quad (4.5)$$

$$\text{DAC} = (n - n_{\text{offset}}) \cdot \frac{(2^m - 1) \cdot GR}{A_{\text{UOZ2}} \cdot A_{\text{UOZ3}} \cdot U_{\text{ref}} \cdot kn} + \frac{2^m - 1}{A_{\text{UOZ2}}} = (n - n_{\text{offset}}) \cdot \frac{A}{kn} + B \quad (4.6)$$

$$\frac{1}{kn_{\text{digital}}} = \frac{A}{kn} \Rightarrow \text{DAC} = (n - n_{\text{offset}}) \cdot \frac{1}{kn_{\text{digital}}} + B \Rightarrow kn_{\text{digital}} = \frac{n - n_{\text{offset}}}{\text{DAC} - B} \quad (4.7)$$

$$B = \left[\frac{2^m - 1}{A_{\text{UOZ2}}} \right] = \left[\frac{2^{12} - 1}{2} \right] = 2048 \Rightarrow kn_{\text{digital}} = \frac{n - n_{\text{offset}}}{\text{DAC} - 2048} \quad (4.8)$$

Při kalibraci se nejprve desetkrát změní rychlost otáčení rotoru při nulovém výstupním napětí rozdílového zesilovače n_{offset} s časovou prodlevou mezi měřeními 250 ms a výsledná hodnota se na závěr zprůměruje. Poté se ve třech bodech výstupního napětí rozdílového zesilovače (cca 0,5, 1 a 3 V) změní otáčky rotoru a v každém bodu se vypočítá převodní konstanta dle vztahu (4.8), která se kumulativně sčítá a na závěr vydělí počtem měření. Nutno podotknout, že se při kalibraci a pozdějším nastavování rychlosti otáčení uvažuje převážně lineární závislost otáček rotoru na výstupním napětí rozdílového zesilovače. Odvozený vzorec (4.9) z rovnice (4.7) se používá ve firmware k nastavení hodnoty DA převodníku pro dosažení požadované rychlosti.

$$DAC = \frac{n - n_{\text{offset}}}{kn_{\text{digital}}} + 2048 \quad (4.9)$$

Nakonec kalibrace se zavolá procedura nalezení bázové pozice rotoru, která ve spolupráci s jednoduchým podprogramem pro obsluhu přerušení na nástupnou hranu signálu M informující o dosažení nulové pozice nastaví počáteční pozici rotoru.

4.4 Měření rychlosti otáčení

První z měřicích funkcí firmware je měření otáček využívající blokující čekání přesně stanovený čas. Vychází z myšlenky, že 1 otáčka za sekundu je změna pozice odpovídající počtu pulsů na jednu otáčku, tedy rozlišení kodéru, za jednu sekundu, je pak vztah určující počet otáček za minutu následující:

$$n = 60 \cdot \frac{\Delta\varphi}{\Delta t \cdot p \cdot Res} \quad (4.10)$$

kde n jsou otáčky, $\Delta\varphi$ je změna úhlu za zvolené uplynutí doby Δt a p je koeficient zvyšující rozlišení kodéru. Při reakcích jak na nástupné tak i sestupné hrany kvadraturního signálu bude $p = 4$ (při vyhodnocování pouze nástupných hran u obou signálů bude $p = 2$). Rozsah volby časové změny závisí na rozsahu měřené rychlosti a také na rozsahu úhlové změny. Následující vztah (4.11) a výpočet ukazuje hranice volby tohoto parametru:

$$60 \cdot \frac{\Delta\varphi_{\text{min}}}{n_{\text{max}} \cdot p \cdot Res} \leq \Delta t \leq 60 \cdot \frac{\Delta\varphi_{\text{max}}}{n_{\text{min}} \cdot p \cdot Res}, \quad (4.11)$$

$$60 \cdot \frac{1}{180 \cdot 4 \cdot 100 \cdot 10^3} \leq \Delta t \leq 60 \cdot \frac{399999}{1 \cdot 4 \cdot 100 \cdot 10^3} \Rightarrow 833,3 \bar{3} \text{ ns} \leq \Delta t \leq 59,9 \bar{9} \text{ s}.$$

Blokující zpoždění v importované knihovně HAL z CubeMX [20] umožňuje provést zpoždění ve stovkách milisekund. Z tohoto důvodu byl zvolen časový rozdíl 100 ms jako kompromis mezi přesností a rychlostí měření. Výpočet změny úhlu je uveden v následující tabulce s podmínkami ošetřující průchod nulovou pozicí (šedě podbarvené pole v tabulce), kdy dochází k nulování pozičního čítače. Proměnné φ_1 a φ_2 označují počáteční a koncovou úhlovou pozici a φ_{max} je maximální hodnota úhlové pozice inkrementálního kodéru, která činí v tomto případě 400 000 kódů na otáčku.

Po výpočtu následuje neprodleně odeslání výsledku přes sériovou linku do PC. Tento postup se opakuje do doby, než je stlačeno tlačítko BTN_STOP. Pak dojde k plynulému zastavení rotoru a vyčkávání na další příkaz.

Tabulka 4.2 Výpočty změny úhlu a jeho podmínky.

Smysl čítání	φ_1 [kód]	φ_2 [kód]	Výpočet změny úhlu $\Delta\varphi$ [kód]	Podmínka
Inkrementální	500	1500	$\Delta\varphi = \varphi_2 - \varphi_1$ $\Delta\varphi = 1500 - 500 = 1000$	$\varphi_2 \geq \varphi_1$
	399500	500	$\Delta\varphi = \varphi_2 - (\varphi_1 - \varphi_{\max})$ $\Delta\varphi = 500 - (-500) = 1000$	$\varphi_2 < \varphi_1$
Dekrementační	1500	500	$\Delta\varphi = \varphi_2 - \varphi_1$ $\Delta\varphi = 500 - 1500 = -1000$	$\varphi_2 \leq \varphi_1$
	500	399500	$\Delta\varphi = (\varphi_2 - \varphi_{\max}) - \varphi_1$ $\Delta\varphi = (-500) - 500 = -1000$	$\varphi_2 > \varphi_1$

4.5 Krokování pozice

Další z funkcí navrhnutého firmwaru je otočení o stanovený úhel (krok), kdy dojde k zaznamenání počáteční pozice, která se odešle do PC, a koncová pozice se dopočítá dle vztahů uvedených v následující tabulce, ve které jsou také uvedeny podmínky provedení kroku. Nesplní-li se při klasickém výpočtu (nepodbarvená pole) v daném smyslu čítání (otáčení) uvedená podmínka, provede se výpočet pomocí vztahu uvedeného v šedě podbarveném poli tabulky. Nesplnění této podmínky je způsobeno konečnou hodnotou pozice, kdy finální pozice nemůže být větší než tato hodnota, a v této práci je tento jev nazván jako přechod nulovou pozicí.

Tabulka 4.3 Výpočty finální úhlové pozice a její podmínky.

Smysl čítání	Výpočet finální úhlové pozice	Podmínka provedení kroku
Inkrementální	$\varphi_2 = \varphi_1 + \Delta\varphi$	$\varphi_2 < \varphi_{\max}$
	$\varphi_2 = \varphi_1 - \varphi_{\max} + \Delta\varphi$	$\varphi_2 < \varphi_{\max}$
Dekrementační	$\varphi_2 = \varphi_1 - \Delta\varphi$	$\varphi_2 \geq 0$
	$\varphi_2 = \varphi_1 + \varphi_{\max} - \Delta\varphi$	$\varphi_2 \geq 0$

V prvotní fázi této funkce se nastaví poloha rotoru na nulovou (bázovou) pozici a poté se vyčkává na spouštěcí signál provedení kroku. K tomuto účelu lze použít řízení příkazem, čili pomocí ovládacího software, anebo pomocí externího přerušení sestupnou hranou signálu DO_STEPB příslušné IRQ linky. V podprogramu pro obsluhu přerušení se pak do proměnné popisující stav stavového automatu nastaví hodnota STEP_proc pomocí níž se poté zavolá funkce provedení kroku. Při procesu krokování se rotor roztočí na malé otáčky tak, že se přivede odpovídající napětí z DA převodníku

na daný vstup servozesilovače a signál ROT_STOP se nastaví do logické nuly. Během otáčení je neustále snímána aktuální pozice a porovnávána s pozicí finální. Dojde-li k shodě, provede se náhlé zastavení rotoru nastavením signálu ROT_STOP do logické jedničky. Nastavení DA převodníku na neutrální hodnotu (polovina rozsahu) se provede až poté. Po ukončení kroku se ještě vyčte hodnota finální pozice a odešle se do PC. Poté se čeká na další spuštění provedení kroku nebo po obdržení STOP příkazu se mód krokování pozice ukončí a očekává se další volba funkce modulu.

V tomto módu je jisté omezení nastavení kroku dané převážně setrvačností mechanického systému. Prakticky bylo ověřeno, že pokud se krok nastaví příliš malý, rotor nezastaví na očekávané pozici, ale udělá otáčku o 360° plus zadaný krok. Tento jev lze jednoduše nazvat přeskočení pozice a lze jej ovlivnit ze strany firmware hodnotou nastavených otáček při vykonávání kroku. Obecně platí, že čím menší otáčky tím rychleji se rotor zastaví a tím menší lze nastavit úhlový krok. Hodnota minimálních otáček je však omezena rozlišením DA převodníku a celkovou koncepcí buzení DC motorku. Také je tato hodnota ovlivněna především rušením řídicího signálu. Proto musí být hodnota řídicího napětí poměrově vyšší vůči hodnotě rušivých signálů a řídicí signál chráněn vůči rušivým signálům např. stíněním. Z programového hlediska je nastavení malých otáček ovlivněno správností určení rychlostí otáčení při nulovém vstupním napětí servozesilovače tedy prvním krokem procesu kalibrace.

Z důvodu uvedených v předchozím odstavci byla vybrána rychlost otáčení 6 ot/min, což za uplynutí doby jedné sekundy způsobí posun o 40 000 pulsů neboli 36° s využitím maximálního možného rozlišení inkrementálního kodéru. Ukázka zdrojového kódu uvedeného níže ukazuje testování dosažení cílové pozice. Toto řešení lze použít za předpokladu dostatečně rychlého vzorkování hodnoty čítače pozice. V kódu je implementováno blokující čekání po dobu 10 ms z důvodu přechodových jevů regulačního systému což zvyšuje minimální hodnotu úhlového kroku o 0,36°.

```
uint32_t Rotator_step_proc(int32_t step, uint8_t trigger_type, uint8_t
direction)
{
...

/* Set Stop signal to zero for enable rotation */
HAL_GPIO_WritePin(ROT_STOP_GPIO_Port, ROT_STOP_Pin, 0);

/* Avoid oscillation of the shaft - caused increase minimum step */
HAL_Delay(10);

/* Test if the step reached */
do
{
    position = (int32_t)(__HAL_TIM_GET_COUNTER(&htim2));
} while(position != phi2);

/* Set correct DAC output voltage for 0 rpm and set Stop signal for
zero torque */
HAL_GPIO_WritePin(ROT_STOP_GPIO_Port, ROT_STOP_Pin, 1);
HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, __DAC_V2value(
1.25));

return phi1;
}
```

4.6 Synchronní měření pozice při rotaci

Poslední z měřících funkcí modulu je synchronní snímání (vzorkování) úhlové pozice při otáčení rotoru. Funkce je závislá na správném provedení kalibrace převodní konstanty včetně offsetu budícího systému DC motorku. Po jejím spuštění se pozvolna rotor roztočí na požadované otáčky a poté program vyčkává na sestupnou hranu externího signálu SYNCB. Ten je v MCU zpracováván jako zdroj externího přerušení pomocí linky IRQ podprogramem, ve kterém se nastaví adekvátní hodnota do proměnné popisující aktuální stav jednoduchého stavového automatu, jež je součástí hlavního programu.

Po změně stavu na stav SYNCB_proc se provede zaznamenání aktuální hodnoty čítače pozice do datové paměti SRAM a inkrementaci počítadla uložených vzorků. Poté se změní stav na MPOS_loop ve kterém se vyčkává na další sestupnou hranu synchronizačního signálu. Tato činnost je prováděna tak dlouho, dokud hodnota počítadla uložených vzorků nedosáhne požadovaného počtu záznamů pozic nebo dokud není přijat příkaz STOP k zastavení prováděné činnosti. Po zastavení rotoru se zaznamenané hodnoty odešlou přes rozhraní USB do PC. V ukázce zdrojového kódu je uvedeno vyhodnocování stavu stavového automatu a provedení záznamu aktuální pozice rotoru. Dále je zde uveden způsob měření doby mezi přijetím synchronizačního pulzu a jeho zpracováním pomocí vystavení signálu DONE do logické nuly po tuto dobu, což je provedeno v podprogramu pro obsluhu přerušení a v hlavním programu po provedení záznamu se signál nastavuje zpět do logické 1.

```
...
case MPOS_loop:
    while(Index_data < samples && !stop)
    {

        /* Check and process received command (only STOP command change
        the state) */

        CDC_process_command(&stop, &speed, &direction, &samples, &step,
        &trigger);

        /* Check if SYNCB state is required */
        if(Mode == SYNCB_proc)
        {
            /* Capture rotor position to SRAM and change the state */

            Position[Index_data] = __HAL_TIM_GET_COUNTER(&htim2);
            Index_data++;
            Mode = MPOS_loop;

            /* Only for delay measurement */
            HAL_GPIO_WritePin(DONE_GPIO_Port, DONE_Pin, 1);
        }
    }
...

```

4.7 Roztočení a zastavení rotoru

Pro funkce měření rychlosti a synchronního záznamu pozice při rotaci je nutné rotor uvést adekvátním způsobem do pohybu a na konci těchto funkcí jej také zastavit. To je prováděno pomocí procedur `Rotator_run` a `Rotator_stop`. Při procesu roztáčení se vychází z výchozí hodnoty DA převodníku, která je rovna polovině jeho rozsahu. V prvním kroku se vypočítá hodnota převodníku, jež se má dosáhnout pro zadanou rychlost otáčení. Poté se výchozí hodnota v cyklu inkrementuje o jedničku s definovaným zpožděním 2 ms mezi jednotlivými inkrementacemi, dokud není dosaženo finální hodnoty převodníku. Uvedené řešení je použito z důvodu dosažení určité strmosti řídicího napětí. Podobně je realizováno zastavení motorku, kdy dochází k dekrementaci zpět na výchozí hodnotu DA převodníku. Na konec se vystaví signál STOP do příslušné úrovně, aby se rotor neotáčel vlivem napěťové nesymetrie servozesilovače. Následující úryvek zdrojového kódu ukazuje implementovaný algoritmus roztočení rotoru mechanického systému.

```
void Rotator_run(uint8_t rpm, uint8_t direction)
{
    static int16_t DAC_target;

    /* Set Stop signal to zero for enable rotation */
    HAL_GPIO_WritePin(ROT_STOP_GPIO_Port, ROT_STOP_Pin, 0);

    /* Check required direction of the rotation */
    if(!direction)
    {
        /* Calculate the DAC_value for required speed */
        DAC_target = (int16_t)((float)(rpm) - rpm_offset)/kn_digital)
+ __DAC_V2value(1.25);
        DAC_value = __DAC_V2value(1.25);

        while(DAC_value < DAC_target)
        {
            /* Check correct DAC_value, then set DAC output voltage */
            if(DAC_value <= 4095 && DAC_value >= 0)
            {
                HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R,
(uint32_t)(DAC_value));
            }

            /* Increase DAC_value by defined slope */
            DAC_value++;
            HAL_Delay(2);
        }
    }
    else
    {
        /* Calculate the DAC_value for required speed */
        DAC_target = (int16_t)((-1.0*(float)(rpm) -
rpm_offset)/kn_digital) + __DAC_V2value(1.25);
        DAC_value = __DAC_V2value(1.25);

        while(DAC_value > DAC_target)
        {
```

```

        /* Check correct DAC_value, then set DAC output voltage */
        if(DAC_value <= 4095 && DAC_value >= 0)
        {
            HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1,
DAC_ALIGN_12B_R, (uint32_t)(DAC_value));
        }

        /* Decrease DAC_value by defined slope */
        DAC_value--;
        HAL_Delay(2);
    }
}

```

5 SOFTWARE

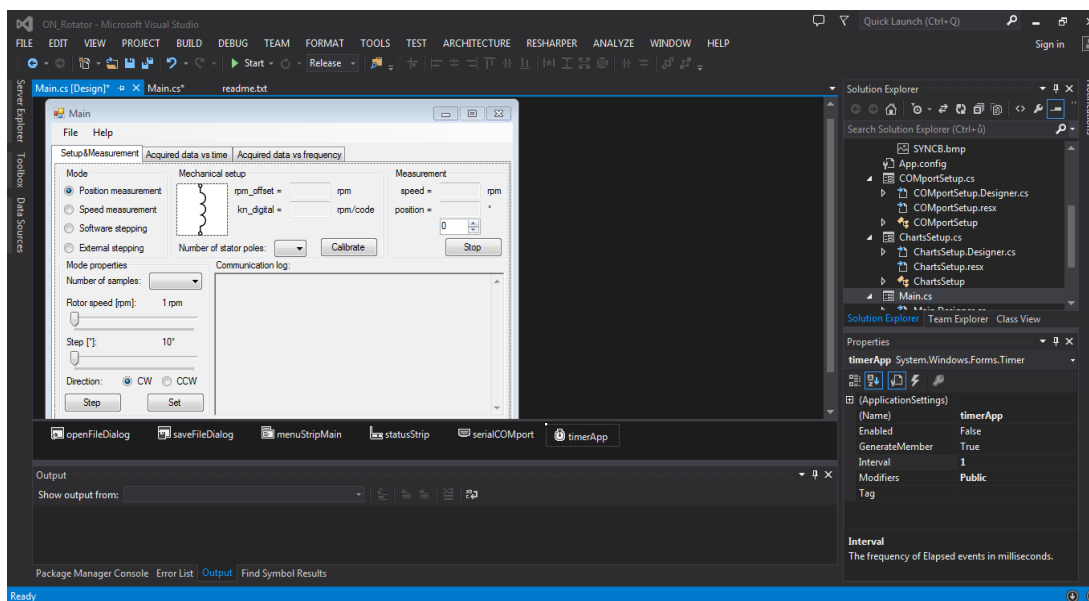
Kvůli jednoduššímu ovládání řídicího modulu, ke kterému by bylo nutné nastudovat podrobnější dokumentaci, slouží software pro počítač, ve kterém jsou implementovány všechny příkazy v uživatelsky přijatelné podobě. Prakticky umožňuje okamžité užívání zařízení. V následujících kapitolách budou zmíněny požadavky na ovládací software, který se nalézá na příloženém CD, a nastíněn jeho vývoj.

5.1 Požadavky

Od ovládacího programu pro počítač se požaduje intuitivní ovládání čili jednoduchost a přehlednost ovládacích prvků. Dále se od něj očekává zpracování naměřených pozičních dat jak z referenčního snímače tak testovaného senzoru pro rychlou analýzu a vyhodnocení verifikačního testu. Také by měl umožňovat naměřená data ukládat na pevný disk počítače pro další zpracování například v dokumentaci o měření. Dalším požadavkem je záznam z komunikace mezi počítačem a řídicím modulem nejvíce využívanou při ladění programu nebo pro pozdější analýzu v případě výskytu závady.

5.2 Vývojové prostředí

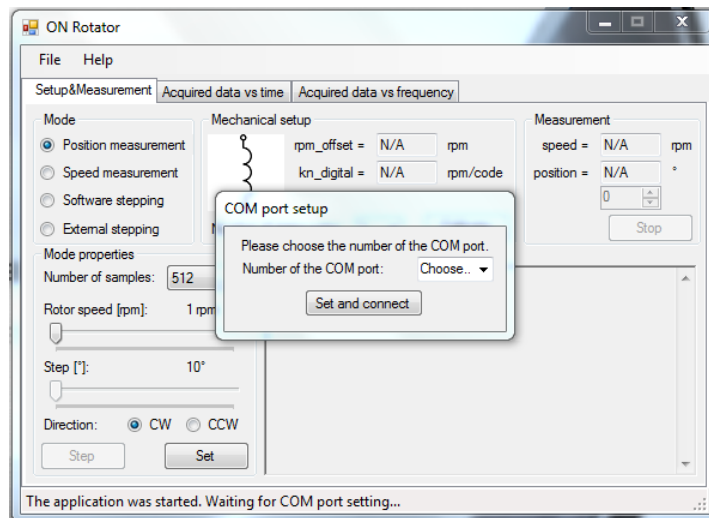
Software pro počítač k ovládání řídicího modulu byl navrhnut ve vývojovém prostředí Visual Studio Ultimate 2013 od společnosti Microsoft, jehož ukázka se nachází na Obrázek 5.1. Program je napsán v jazyce C# platformy .NET Framework jako tzv. formulářová aplikace (Windows Forms) využívající standardní formuláře (okna) jako uživatelské rozhraní. Pro spuštění aplikace je nutné mít v počítači nainstalován .NET Framework verze 4.0, který je podporován od Windows XP po Windows 10.



Obrázek 5.1 Vývojové prostředí Microsoft Visual Studio Ultimate 2013.

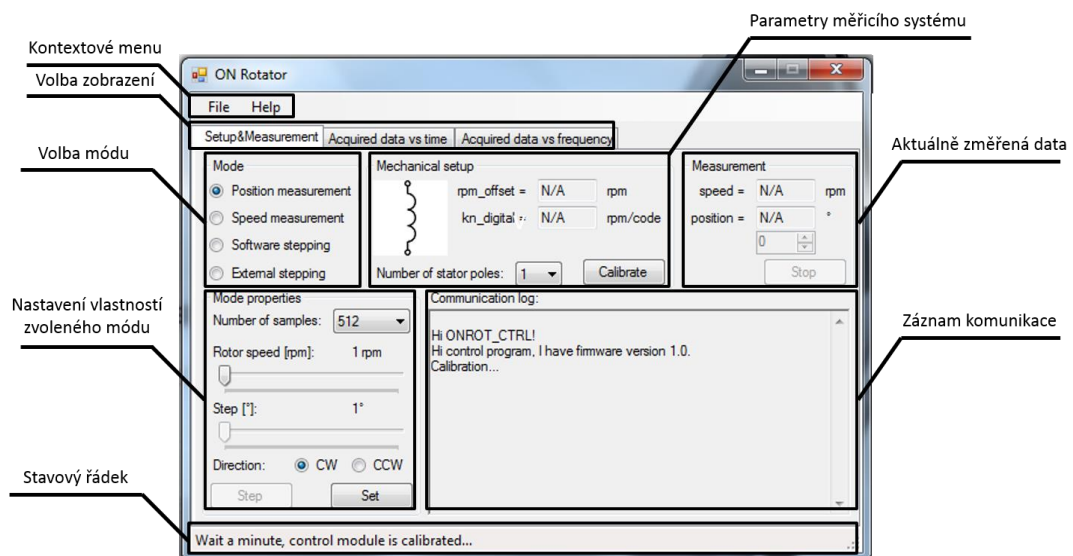
5.3 Popis GUI

Výchozí vzhled ovládacího programu po spuštění je uveden na Obrázek 5.2. Je-li řídicí modul správně napájen a připojen přes USB k PC, objeví se v seznamu patřičný sériový port (COM port), který je nutné vybrat a poté stisknout tlačítko Set and connect. Načež se program pokusí navázat spojení přes sériovou linku s modulem. V případě, že otevření portu selže, je uživatel vyzván dialogovým oknem k překontrolování připojení modulu popřípadě k jeho restartování. Po stisknutí tlačítka OK se aktualizuje seznam portů. Pro umožnění komunikace s modulem je nutné mít v počítači také nainstalován ovladač virtuálního sériového portu, který lze po registraci stáhnout na stránkách výrobce MCU [19].



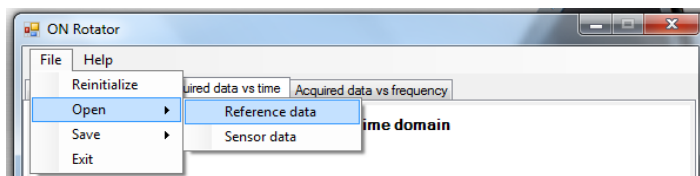
Obrázek 5.2 Výchozí vzhled ovládacího programu po spuštění.

Celkový vzhled uživatelského rozhraní (GUI) je ukazuje Obrázek 5.3, které je rozděleno do třech částí, mezi kterými se lze přepínat pomocí záložek v sekci volba zobrazení. V těchto částech se nacházejí grafy pro zobrazení naměřených pozic v závislosti na čase a také po FFT transformaci zobrazení ve frekvenční oblasti.



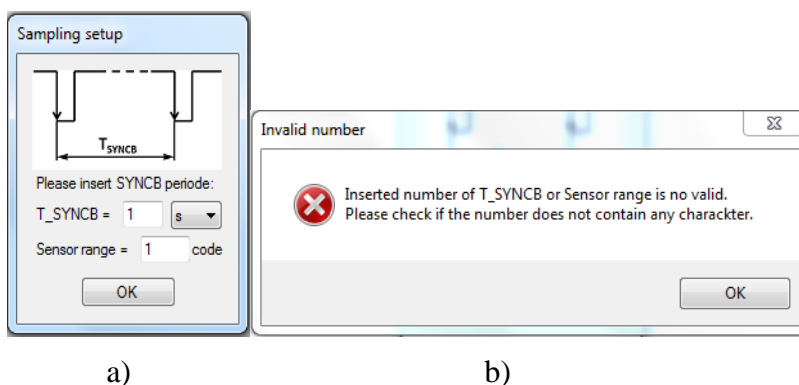
Obrázek 5.3 GUI ovládacího programu.

GUI dále obsahuje sekce parametrů měřicího systému, aktuálně změřených dat, záznam komunikace s modulem a volby módu, kde se nachází přepínač, jehož volbou se aktivují příslušné objekty v sekci nastavení vlastností zvoleného módu. To je řešeno tímto způsobem z důvodu jednak informačního, aby uživatel věděl a měl větší přehled o vlastnostech nastavujících se v daném módu, ale také aby nedocházelo k vysílání neadekvátních příkazů do modulu. Ve stavovém řádku se zobrazují informace o aktuálně provedených či prováděných činnostech programu. Po nastavení módu se ovládací prvky v sekcích volba módu a parametrů měřicího systému zablokují, aby nedošlo k spuštění nepovolené operace během provádění vybraného módu.



Obrázek 5.4 Ukázka kontextového menu.

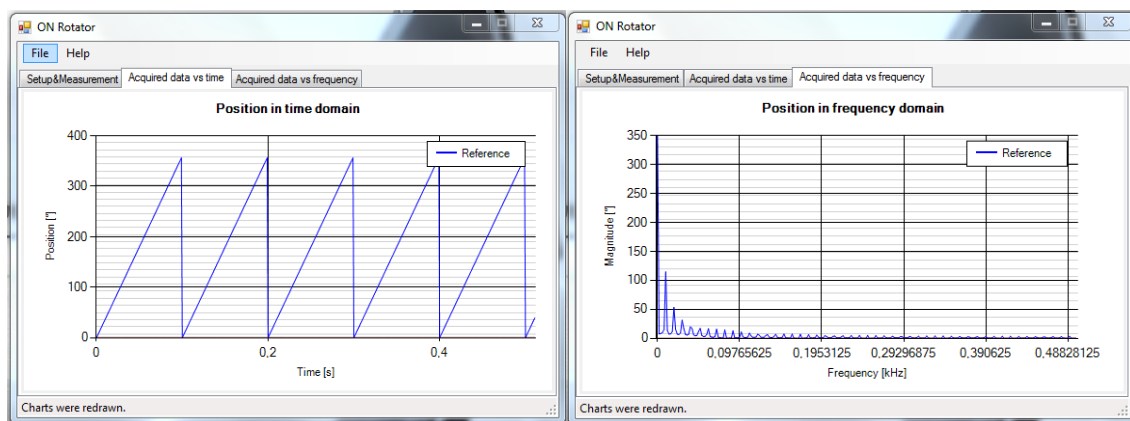
Pro otevření již naměřených a uložených dat ve formátu CSV (hodnoty oddělené čárkou) v počítači slouží volba v kontextovém menu podle Obrázek 5.4. Do tohoto formátu také program umožňuje ukládat naměřená data z referenčního senzoru. Záznam komunikace je ukládán v textovém formátu (*.txt). Po výběru zda-li jsou to data z referenčního nebo testovaného snímače se zobrazí klasické dialogové okno pro výběr souboru. Po zvolení souboru a potvrzení volby se zobrazí formulářové okno dotazující se na hodnotu periody synchronizačního signálu nebo také s jakou periodou byly data vzorkovány. V aplikaci je předpokládáno, že vzorkovací perioda je stejná jak u referenčního tak i testovaného snímače. V případě chybného zadání hodnoty periody nebo rozsahu zkoumaného senzoru, kdy chybou je neúmyslné vepsání znaku místo čísla je tato výjimka ošetřena chybovým hlášením, po jejímž odsouhlasení lze hodnotu zadat znovu a potvrdit kliknutím na tlačítko OK (viz Obrázek 5.5).



Obrázek 5.5 a) Formulář nastavení vzorkovací periody a b) dialogové okno v případě chybného zadání.

Během jakéhokoliv měření jsou objekty v sekci aktuálně změřená data neustále aktualizována odposlechem sériové linky. V této sekci je také aktivováno tlačítko Stop pro umožnění ukončení měření v jakémkoliv stádiu. V případě změny v mechanickém systému nebo v případě chybné kalibrace po zapnutí (např. zapomenutí přepnutí z manuálního ovládání měřicího systému na externí řízení) není nutné modul restartovat ale použít volbu Calibrate pro opětovné provedení kalibrace systému.

Program také dokáže převést naměřené data z časové oblasti do kmitočtové oblasti, kde je kmitočet reprezentován jako otáčky za jednu sekundu. K tomuto účelu je v programu implementován FFT algoritmus, který zpracuje naměřené vzorky a ty se na základě zadané periody SYNCB signálu (vzorkovací periody) zobrazí v grafu v záložce Acquired data vs frequency. Naměřená data v časové oblasti si lze prohlédnout v záložce Acquired data vs time. Obrázek 5.6 zobrazuje ideální průběh referenční pozice v závislosti na čase a jeho obraz v kmitočtové oblasti při vzorkovací periodě 1 ms a rotaci 10 ot/min.

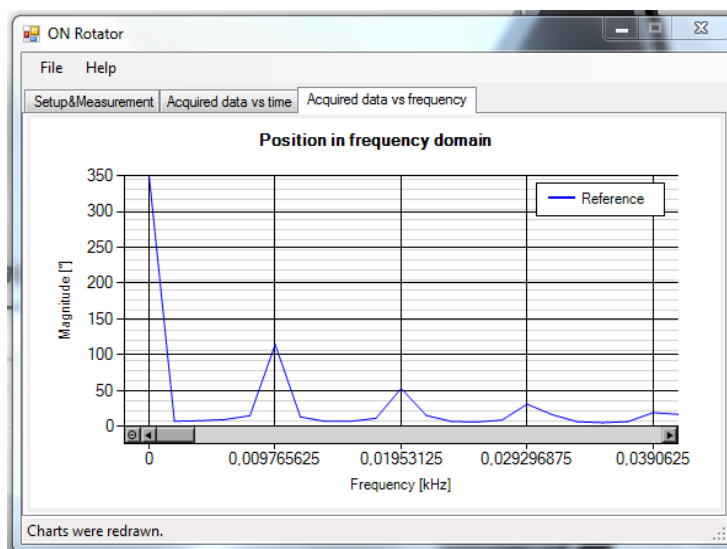


a)

b)

Obrázek 5.6 Ideální průběh referenční pozice v a) časové a b) jeho obraz v kmitočtové oblasti.

V objektu grafu je také zabudovaná funkce přiblížení umožňující prohlédnout si detaily naměřených dat. Ta se aktivuje najetím kurzoru myši do zóny grafu a současným zmáčknutím levého tlačítka tažení myši se vybere požadovaná zóna na zvětšení. Po uvolnění tlačítka myši software provede roztažení vybrané oblasti do celkové plochy grafu. Obrázek 5.7 ukazuje použití této funkce na zmiňovaném referenčním signálu v kmitočtové oblasti.



Obrázek 5.7 Detail ideálního průběhu referenční pozice při rotaci v kmitočtové oblasti.

5.4 Popis činnosti

Software pracuje na základě vyhodnocování událostí, které přicházejí z uživatelské strany kliknutím na některý z grafických ovládacích prvků nebo zmáčknutím příslušné klávesy. Každý ovládací prvek má své vlastnosti a seznam událostí, ke kterým se mohou naprogramovat předem daná posloupnost činností k dosažení požadovaného výsledku. V jazyku C#, který je jazykem objektově orientovaného programování, se používají třídy, které si lze představit jako továrnu na objekty. Ta se skládá z privátních a veřejných datových členů (proměnných), metod, vlastností, událostí, operátorů a delegátů, ale těmi se zabývat práce nebude. Vždy však obsahuje konstruktor, který se volá při vytváření třídy. Následující ukázka zdrojového kódu ukazuje inicializaci a konstruktor třídy hlavního formuláře obsahující vytvoření formulářů pro nastavení sériového portu a vzorkovací periodu, se kterými se dále v aplikaci pracuje. Poté je volaná metoda formuláře nastavení sériového portu pro zobrazení při události načtení hlavního okna aplikace, ve které si lze také prohlédnout inicializaci ostatních prvků uživatelského rozhraní.

```
...
namespace ON_Rotator
{
    public partial class Main : Form
    {
        private int poles, samples, first, last, refDataIndex;
        private int setSpeed = 1; // minimum speed of rotation 1 rpm
        private int setStep = 10; // minimum step 10°

        private const int EncoderRange = 400000;
        private string message, command, temp;
        public double timeStep;

        private int sensorDataIndex = 0;
        public float[] refData = new float[16384];
        private float[] sensorData = new float[16384];

        public int sensorRange = 1;
        private COMportSetup portSetup;
        private ChartsSetup chartsSetup;

        /* Constructor of the Main form */
        public Main()
        {
            InitializeComponent();
            portSetup = new COMportSetup(this);
            chartsSetup = new ChartsSetup(this);
        }

        private void Main_Load(object sender, EventArgs e)
        {
            cmbPoles.SelectedIndex = 0;
            cmbSamples.SelectedIndex = 0;

            cmbSamples.Enabled = true;
            trckBarSetSpeed.Enabled = true;
            trckBarSetStep.Enabled = false;
        }
    }
}
```

```

        btnStep.Enabled = false;
        btnSet.Enabled = true;
        btnStop.Enabled = false;
        numUpDownDataIndex.Enabled = false;

        tbMeasSpeed.Text = @"N/A";
        tbMeasPosition.Text = @"N/A";
        tbKnDigital.Text = @"N/A";
        tbRpmOffset.Text = @"N/A";

        command = "MPOS";
        statusLabel.Text = @"The application was started. Waiting
for COM port setting...";
        portSetup.Show();
    }
    ...
}
}

```

Popsání jednotlivých použitých metod a použitých tříd se vymyká zamýšlenému rozsahu této diplomové práce, a proto se tomuto tématu nebude práce nadále věnovat. Budou pouze povrchně uvedeny použité principy k dosažení požadované funkcionality aplikace. Více se o objektově orientovaném programování lze dočíst v [4].

První z nich je volání metod objektů z třídy hlavního okna, které se provádí nastavením vlastnosti přístupu na veřejnou u vybraných objektů, jako je např. sériový port. Další ze zajímavých funkcí software je algoritmus načítání naměřených dat z CSV souboru, jež je uveden v následující ukázce zdrojového kódu, kde je uvedena metoda s datovým proudem.

```

private int OpenCsvFile(float[] data)
{
    int index = 0;

    openFileDialog.Filter = @"csv files (*.csv)|*.csv|All files
(*.*)|*.*";
    openFileDialog.FilterIndex = 1;
    openFileDialog.RestoreDirectory = true;
    openFileDialog.FileName = "";

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        StreamReader csv = new StreamReader(openFileDialog.FileName);

        while (!csv.EndOfStream)
        {
            string line = csv.ReadLine();
            string[] values = line.Split(',');

            float.TryParse(values[1], NumberStyles.AllowDecimalPoint,
CultureInfo.InvariantCulture, out data[index]);
            index++;
        }

        samples = index;
        csv.Close();
        csv.Dispose();
    }
}

```

```

chartsSetup.Show();
chartsSetup.TopMost = true;
statusLabel.Text = @"File '" + openFileDialog.FileName + @"' was
opened.";
}

return index;
}

```

Podobně se provádí ukládání naměřených pozičních dat do CSV souboru pomocí formátovaného výstupu.

```

/* Save reference data into csv file */
private void measuredDataToolStripMenuItem1_Click(object sender,
EventArgs e)
{
saveFileDialog.Filter = @"csv files (*.csv)|*.csv|All files
(*.*)|*.*";
saveFileDialog.FilterIndex = 1;
saveFileDialog.RestoreDirectory = true;
saveFileDialog.FileName = "";

if (saveFileDialog.ShowDialog() == DialogResult.OK)
{
StreamWriter csv = new StreamWriter(saveFileDialog.FileName);

for (int i = 0; i < samples; i++)
{
string value = refData[i].ToString(CultureInfo.InvariantCulture);
string newLine = string.Format("{0},{1}", i, value);
csv.WriteLine(newLine);
}

csv.Close();
csv.Dispose();
statusLabel.Text = @"File was saved.";
}
}

```

Odposlouchávání sériového portu je prováděno pomocí systémového časovače, jehož perioda je nastavena na 1 ms (menší nastavit nelze). Každou milisekundu je tedy volána událost Tick, ve které se vyčtou neprodleně aktuální data ze sériové linky a uloží se do textového řetězce (string) s názvem message . Ten se pak testuje, zdali neobsahuje některé z očekávaných klíčových slov. Obsahuje-li přijatý řetězec dané klíčové slovo, je postoupen syntaktické analýze pro získání požadovaných hodnot na základě rozpoznání klíčového slova. Ukázkou syntaktické analýzy pro odchycení kalibračních hodnot měřicího systému ukazuje následující výňatek zdrojového kódu. Podobně je provedeno naslouchání ze strany modulu, kdy se v hlavní smyčce stavového automatu volá procedura zpracování příkazu, která testuje, zdali nebyl nějaký příkaz přijat. Byl-li přijat, je postoupen syntaktické analýze pro zjištění potřebných údajů k provedení požadované činnosti.

```

//Detection calibration data
if (message.Contains("rpm/code"))
{
first = message.IndexOf("kn_digital", StringComparison.Ordinal) +
"kn_digital".Length;
last = message.IndexOf(" rpm/code", StringComparison.Ordinal);
temp = message.Substring(first, last - first);
}

```

```

    tbKnDigital.Text = temp;

    first = message.IndexOf("rpm_offset", StringComparison.Ordinal) +
"rpm_offset".Length;
    last = message.LastIndexOf(" rpm", StringComparison.Ordinal);
    temp = message.Substring(first, last - first);

    tbRpmOffset.Text = temp;
    statusLabel.Text = @"Ready";
}

```

V programu je také využít volně šiřitelný NuGet balíček Math.NET Numerics verze 3.11.1 jehož instrukce k nainstalování do vývojového prostředí Visual studia je uveden na domovské webové stránce [38]. Ten se po stažení a nainstalování do zdrojového kódu přidá pomocí direktivy `using MathNet.Numerics.IntegralTransforms;`, ve které již také lze specifikovat podskupinu použité funkce. Z tohoto balíčku je použita FFT transformace pro přeložení naměřených dat z časové oblasti do oblasti frekvenční, ve které se snáze analyzuje linearita testovaných pozičních senzorů. Příklad použití FFT transformace z balíčku je představen v následujícím úryvku zdrojového kódu funkce překreslování grafů.

```

...
Complex[] refFFTcomplex = new Complex[samples];
Complex[] sensorFFTcomplex = new Complex[samples];

for (int i = 0; i < samples; i++)
{
    refFFTcomplex[i] = new Complex(refData[i], 0);
    sensorFFTcomplex[i] = new Complex(sensorData[i], 0);
}

Fourier.Forward(refFFTcomplex, FourierOptions.NoScaling);
Fourier.Forward(sensorFFTcomplex, FourierOptions.NoScaling);

for (int i = 0; i < samples/2; i++)
{
    refFFTseries.Points.AddXY(i / (samples*timeStep*1000.0),
refFFTcomplex[i].Magnitude*2/samples);
    sensorFFTseries.Points.AddXY(i / (samples*timeStep*1000.0),
sensorFFTcomplex[i].Magnitude*2/samples);
}

chartDataVsFrequency.Invalidate();
...

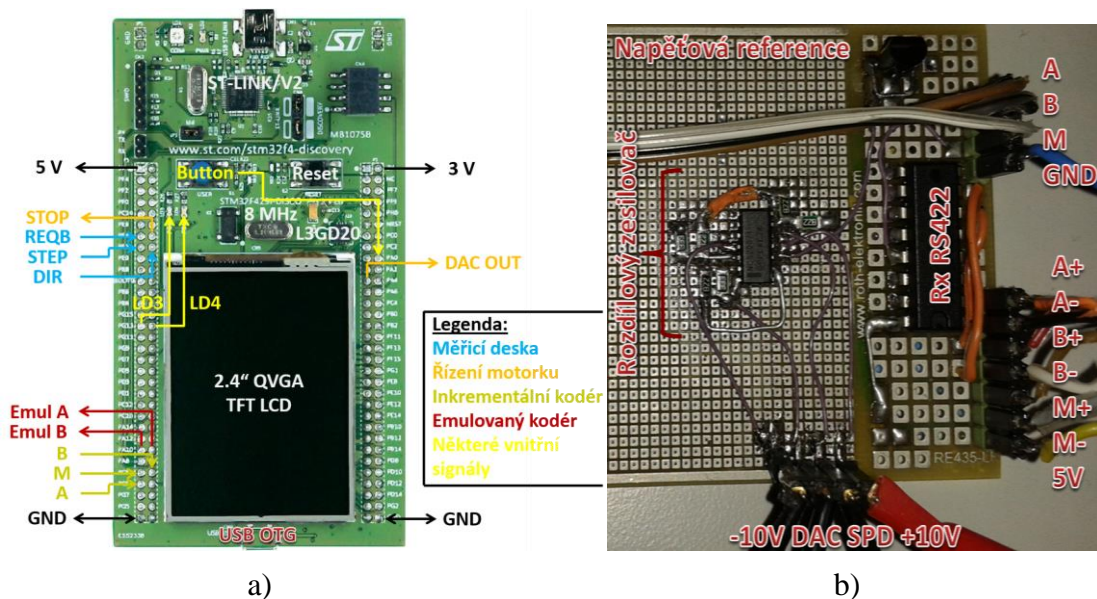
```

Při vývoji .NET aplikace byl také používán velmi užitečný nástroj s názvem ReSharper od společnosti Jet Brains, který se nainstaluje do vývojového prostředí Visual studia jako jeho nadstavba (více informací lze nalézt v [39]). Tento nástroj urychluje psaní kódu pomocí přehledného našeptávání s nabídkou přidání nebo odebrání knihoven v závislosti použitých funkcí v aplikaci a neustále kontroluje syntaxi kódu, na kterou v případě nekorektnosti upozorňuje již při psaní změnou barvy textu s popisem chyby.

6 MĚŘENÍ VLASTNOSTÍ MODULU

Deska plošných spojů řídicího modulu byla navržena v návrhovém prostředí Altium Designer a je uvedena v příloze (viz příloha A) včetně seznamu použitých součástek a podrobných schémat. Podle výkresové dokumentace byla realizovaná finální verze modulu, jehož fotografie si lze prohlédnout v příloze (viz příloha B). Po realizaci následovalo proměření základních parametrů jednotlivých sekcí modulu uvedených v příslušných kapitolách. Ověření činnosti s mechanickým rotačním systémem však nebylo možné z důvodu neočekávané závady na inkrementálním kodéru. Proto jsou v následujících kapitolách prezentovány výsledky měření převážně s prototypem řídicího modulu skládajícího se z vývojového kitu STM32F429 Discovery [13] s MCU STM32F429 ze stejné rodiny jako zvolený a již popsáno rozdílového zesilovače realizovaného na univerzální desce plošného spoje uvedeném na a) b)

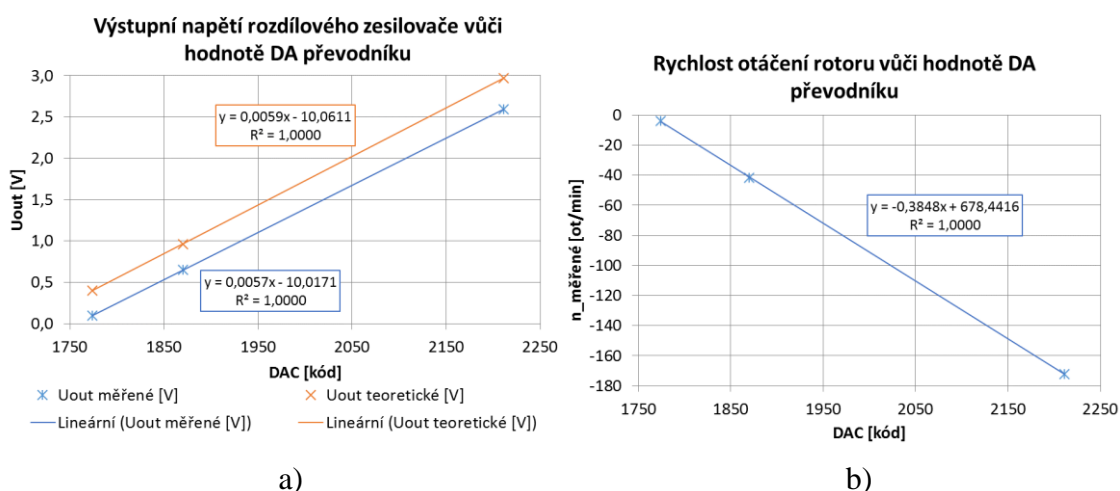
Obrázek 6.1. MCU osazený ve vývojovém kitu obsahuje více periférií a má větší programový a datový paměťový prostor než použitý MCU (viz [28]), což se v průběhu vývoje firmware ukázalo jako zbytečné a proto byl vybrán jeho „sourozenec“ s nižšími parametry.



Obrázek 6.1 Součásti prototypového řešení: a) vývojový kit (převzato z [8]) a b) prototyp rozhraní k mechanickému systému.

6.1 Kalibrace

Ověření kalibrace spočívalo v tom, že se při ladění programu vyčítali postupně dané rychlosti a hodnoty uložené v registru DA převodníku. Také bylo změřeno výstupní napětí rozdílového zesilovače, jehož průběh v závislosti na nastavené hodnotě převodníku uvádí Obrázek 6.2. Porovnáním změřeného průběhu výstupního napětí s očekávaným napětím lze říci, že budicí systém pro servozesilovač obsahuje chybu zesílení - 3,39 % vůči očekávané hodnotě s celkem vysokou absolutní chybou napěťové nesymetrie - 44 mV. Důvodem tak vysoké odchylky napěťové nesymetrie je použití dvou rozdílných referenčních napětí kvůli nemožnosti změnit referenční napětí DA převodníku na vývojovém kitu zapůjčeném v rámci řešení projektu předmětu MPOA vyučovaném Ing. Alešem Povalečem, Ph. D. na ústavu radioelektroniky. Chybu zesílení způsobují výrobní tolerance použitých rezistorů v rozdílovém zesilovači, které nabývají hodnoty až 5 %. Součástí obrázku je také regulační charakteristika otáček rotoru, ze které byla pomocí aproximace lineární rovnicí vyčtena digitální převodní konstanta měřicího systému považovaná za správnou (referenční) hodnotu.

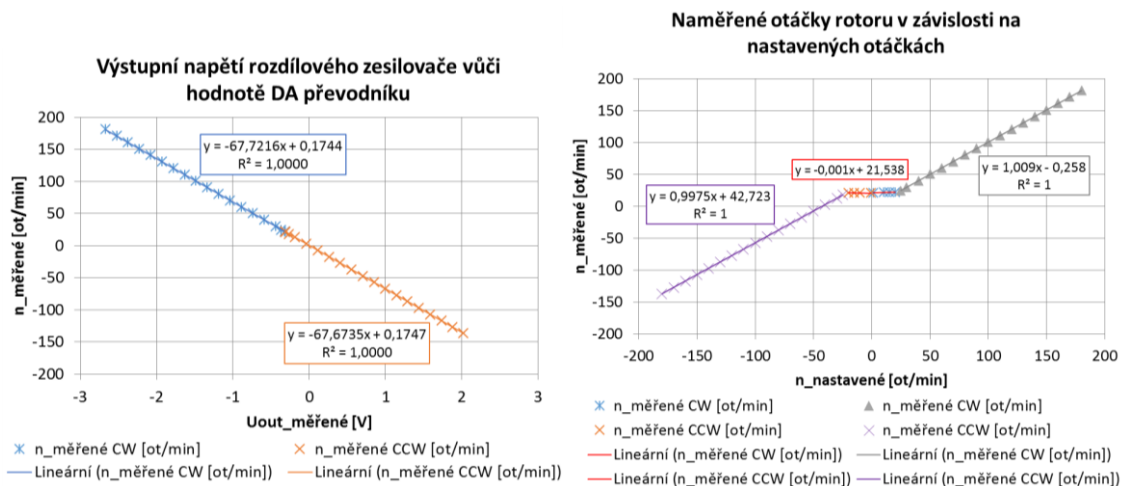


Obrázek 6.2 Charakteristiky a) výstupního napětí rozdílového zesilovače a b) otáček rotoru při kalibraci.

Program MCU určil převodní konstantu v absolutní hodnotě 0,3837 (ot/min)/kód s přesností - 0,3 % při porovnání s analyticky zjištěnou hodnotou 0,3848 (ot/min)/kód. Na začátku kalibrační procedury byla programem změřena rychlost otáčení při teoreticky nulovém výstupním napětí rozdílového zesilovače nabývající hodnoty 22 ot/min s odchylkou 2,14 % vůči analyticky zjištěné hodnotě z aproximační funkce kolem počátku regulační charakteristiky v grafu naměřených otáček v závislosti na nastavených uvedeného na Obrázek 6.4 z následující kapitoly měření přesnosti rychlosti otáčení. Napěťová nesymetrie, jako odchylka od teoretické hodnoty výstupního napětí rozdílového zesilovače činí - 44 mV a je způsobena zejména napěťovou nesymetrií operačních zesilovačů v rozdílovém zesilovači, která se kompenzuje pomocí změřených otáček za teoreticky nulového výstupního napětí zesilovače.

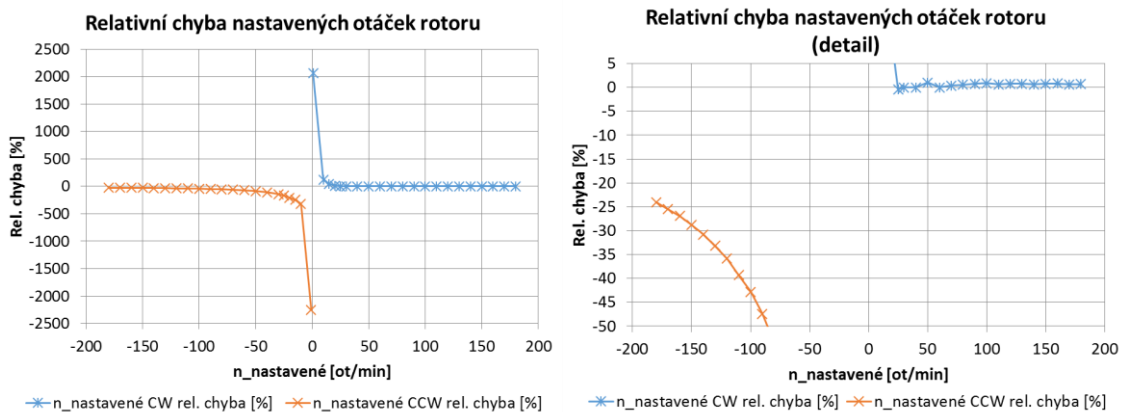
6.2 Přesnost nastavené rychlosti

Po provedení ověření kalibrace byla prozkoumána přesnost nastavované rychlosti otáčení pomocí funkce měření otáček. Výsledné charakteristiky měřených otáček v závislosti na výstupním napětí rozdílového zesilovače (regulační charakteristika servozesilovače) a nastavených otáčkách jsou zobrazeny na následujícím obrázku (viz Obrázek 6.4). V grafech jsou také uvedeny aproximační funkce získané pomocí analýzy naměřených dat. Z nich lze vyčíst např. nesymetrii a směrnici jakožto zesílení při lineárním průběhu.



Obrázek 6.4 Charakteristiky naměřených otáček rotoru.

Při ověřování nastavování otáček bylo zjištěno, že při nízkých otáčkách se nenastavuje na výstupu rozdílového zesilovače předpokládané napětí, ale v rozsahu nastavovaných otáček od 1 do 20 ot/min je na výstupu zesilovače zhruba -0,317 V nezávisle na zvoleném směru otáčení. Tento stav způsobuje extrémní chybu regulace (viz Obrázek 6.5) a je ovlivněn zejména vlastnostmi DA převodníku a výpočtem jeho hodnoty ve firmwaru. Tyto průběhy informují dále o tom, že provedená kalibrace a metoda buzení mechanického systému lze za určitých podmínek použít pro regulaci otáčení rotoru ve směru hodinových ručiček. Pro regulaci v opačném směru otáčení se použít nedá a je nutné ji upravit.



Obrázek 6.5 Relativní chyba nastavení otáček rotoru.

6.3 Přesnost krokování

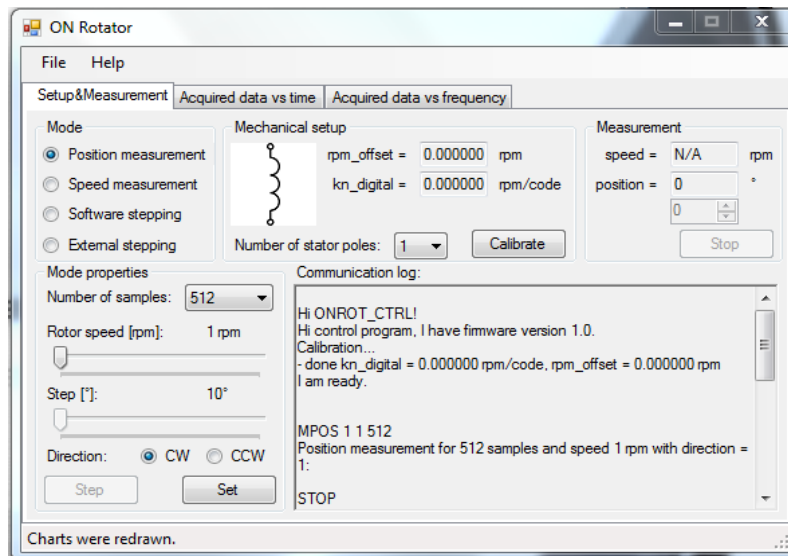
Při vývoji firmware byla také testována funkce krokování pozice, u které se zjišťovala hodnota minimálního kroku, kdy nedojde k přeskočení cílové pozice, a jeho přesnost. Na základě experimentů byl určen minimální úhlový krok o velikosti zhruba 10° (11111 kódů). Poté bylo provedeno 10 měření pro zjištění fluktuace chyby krokování, která byla stanovena na základě výsledků měření pomocí firmware na maximálně 21 kódů čili po přepočtu na stupně při rozlišení inkrementálního kodéru 400 000 PPR zhruba 1 úhlová minuta. Dále si lze všimnout, že konečná pozice se neshoduje s počáteční pozicí v dalším kroku kvůli vysoké citlivosti kodéru na mechanické vibrace. Pro vysoce přesné krokování by bylo lepší použít místo DC motorku krokový, který umožňuje elektrické zablokování mechanické pozice proti pohybu, anebo roztočit rotor a v ekvidistantních časových úsecích vzorkovat pozici což je zahrnuto ve funkci synchronního snímání pozice při rotaci.

Tabulka 6.1 Tabulka naměřených hodnot při krokování pozice.

Měření	Počáteční pozice [kód]	Konečná pozice [kód]	Krok [kód]	Absolutní chyba kroku [kód]
1	21473	32605	11132	21
2	38024	49155	11131	20
3	103844	114976	11132	21
4	170318	181449	11131	20
5	235462	246593	11131	20
6	301226	312358	11132	21
7	365215	376347	11132	21
8	32201	43333	11132	21
9	99455	110587	11132	21
10	166439	177570	11131	20

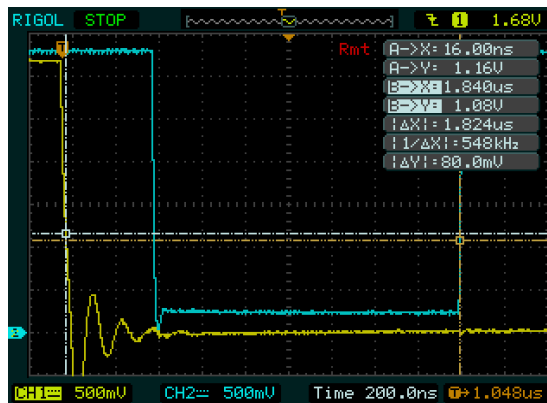
6.4 Zpoždění záznamu pozice

Nejdůležitějším parametrem u synchronního záznamu pozice při rotaci je doba, za kterou firmware zareaguje na synchronizační signál a doba zpracování úhlové pozice zahrnující uložení do datové paměti. Tyto doby byly testovány tak, že se ve firmware implementovali příslušné funkce nastavující signál DONE při zpracování přerušeni do logické nuly a po provedení záznamu do paměti MCU se pak nastaví tento signál opět do logické jedničky. K nastavení finální verze modulu do požadované funkce byl použit vyvinutý ovládací software, ve kterém se nastavili příslušné parametry příkazu MPOS na minimální hodnoty jak ukazuje Obrázek 6.8.



Obrázek 6.8 Nastavení pro analýzu zpracování signálu SYNCB.

Oscilogram uvedený na Obrázek 6.9 zachycuje okamžik zmačknutí mikrospínače simulujícího synchronizační signál SYNCB (žlutě), na který zareagoval MCU stažením signálu DONE (modře) do logické nuly po době 408 ns. Celková doba od vystavení signálu SYNCB do aktivního stavu po konec zpracování aktuální pozice včetně jejího záznamu nabývá hodnoty 1,824 μ s, což je nejmenší perioda (maximální kmitočet cca 548 kHz) signálu SYNCB, kterou je schopen MCU při synchronním měření pozice zpracovat.



Obrázek 6.9 Reakce a doba zpracování signálu SYNCB.

Obrázek 6.10 ukazuje záznam komunikace z testování funkce synchronního měření pozice při vývoji firmware. Mezi prvním a druhým údajem změřené pozice v jednotkách kódu došlo k narušení konstantní rychlosti otáčení okolním rušením kvůli dlouhým přivodním vodičům. Tato nedokonalost byla ošetřena řešením odolnějším na rušení, ale bohužel nemohla být ověřena z důvodu poruchy inkrementálního kodéru.

```
MPOS 10 1 3
Position measurement for 3 samples and speed 10 rpm with direction = 1:

Stopping the rotor...
Transmit collected position data...
Encoder data[0] = 196904
Encoder data[1] = 167899
Encoder data[2] = 167897
Measurement ending - done
```

Obrázek 6.10 Záznam komunikace z testování měření pozice prototypem.

7 ZÁVĚR

Diplomová práce se věnovala rozboru mechanického rotačního systému určeného pro přesné měření moderních bezkontaktních indukčních pozičních senzorů, jeho způsobu buzení včetně možností snímání referenční pozice v závislosti na rychlosti otáčení a realizaci modulu pro jeho ovládání.

Na základě těchto poznatků a požadavků kladených na rotační měřicí systém pak byl navrhnut koncept řídicího modulu, který zároveň plní úlohu digitálního měřicího přístroje, a proveden rozbor jeho dílčích částí včetně rozebrání a definice komunikačního protokolu. Po navržnutí všech hardwarových částí modulu následovala realizace modulu a zároveň vývoj firmwaru použitého MCU.

Ovládací software byl otestován s finální verzí modulu, kdy zahájení a proces komunikace přes emulovanou sériovou linku USB rozhraní probíhal dle očekávání. Software odposlouchává předem určenou sériovou linku, na které je připojen řídicí modul, a ovládá jej podle předem stanovených příkazů. Co však nebylo možné ověřit, jsou měřicí funkce finální verze modulu zejména synchronní záznam pozice při rotaci v součinnosti se softwarem kvůli poruše inkrementálního kodéru, který tak nebyl k dispozici.

Byly, však zjištěny dosažené parametry prototypu řídicího modulu, které jsou regulace otáček v kladném smyslu otáčení v rozsahu od 20 do 180 ot/min s přesností $\pm 1\%$. Přesnost při krokování úhlové pozice $0^{\circ}1'8,04''$ při minimálním nastavitelném kroku rotoru 10° . Dále modul reaguje na synchronizační signál se zpožděním 408 ns, kdy provedení záznamu pozice rotoru trvá nejvýše $1,84\ \mu\text{s}$. Maximální kmitočet vzorkování pozice je tedy 543,5 kHz, což za předpokladu požití synchronizačního signálu s periodou v desítkách μs s rezervou postačuje. Tyto parametry již byly proměřeny na finální verzi modulu, který používá stejný algoritmus reakce na synchronizační signál jako v případě prototypu.

V řídicím firmwaru je ještě stále co zlepšovat. V každém případě je jeho převážná část hotová a je nutné pouze ověřit a popřípadě upravit proceduru kalibrace tak, aby se vykompenzovali případné nelinearity vyskytující se v blízkosti počátku regulační charakteristiky, s kterými se při vývoji firmwaru nepočítalo. Toho si lze všimnout při regulaci otáček ve směru proti hodinovým ručičkám, kdy dochází ke špatné kompenzaci napětového offsetu s působením nelinearity kolem počátku charakteristiky způsobující vysokou chybu nastavených otáček.

Jednou z alternativních řešení by pak bylo použití AD převodníku pro vykompenzování parametrů DA převodníku, jako je chyba zesílení a offsetu, čímž by vznikl měřicí systém se třemi zpětnovazebními smyčkami zaručující vyšší spolehlivost systému, ale také by vstoupili nároky na firmwaru. Také by bylo možné použít digitální potenciometr místo DA převodníku, u kterého je ale problémem přesnost s požadovaným rozsahem napětí mezi jeho přívody, zejména pak schopnost zpracovat záporné napětí.

LITERATURA

- [1] ANALOG DEVICES, INC. *ADuM3160: Data sheet*. Norwood, U.S.A., 2014. Dostupné také z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADuM3160.pdf>
- [2] BASTIAN, Peter. *Praktická elektrotechnika*. 2., dopl. vyd. Praha: Europa-Sobotáles, 2006. ISBN 80-867-0615-X.
- [3] DE SILVA, Clarence W. *Sensors and Actuators: Control Systems Instrumentation*. Boca Raton, FL: CRC Press, 2007. ISBN 1420044834.
- [4] HANÁK, Ján. *Praktické objektové programování v jazyce C# 4.0*. Brno: Artax, 2009. Microsoft (Artax). ISBN 978-80-87017-07-4.
- [5] HUGHES, Austin. *Electric Motors and Drives: Fundamentals, Types and Applications*. 3rd Edition. Amsterdam: Elsevier/Newnes, 2006. ISBN 0750647183.
- [6] JEHLÁŘ, Zbyněk. *Vliv externích elektromagnetických polí na funkci snímačů otáček*. Brno, 2009. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Doc. Ing. Vladislav Singule, CSc.
- [7] KAINKA, Burkhard. *Měření, řízení a regulace pomocí PC: Vývoj hw a sw pro praxi*. Praha: BEN - technická literatura, 2003, 271 s. ISBN 80-7300-089.
- [8] KROLÁK, David. *Modul pro řízení mechanického rotačního systému*. [online]. 2016 [cit. 2016-05-10]. Dostupné z: <http://www.urel.feec.vutbr.cz/MPOA/2015/rot-control>
- [9] MATOUŠEK, David. *USB prakticky s obvody FTDI*. 1. vyd. Praha: BEN - technická literatura, 2003, 270 s. PC. ISBN 80-730-0103-9.
- [10] Speciální krokové motory: Krokové motory na míru. UHER, Petr. *Servo-drive* [online]. 2014 [cit. 2015-12-11]. Dostupné z: http://www.servo-drive.com/specialni_krokov_e_motory_krokov_e_motory_na_miru.php
- [11] VESKE, Sefa. *STM32F4 Discovery Tutorials - Interrupt (EXTI)*. *ARM Programming* [online]. 2015, 30.1.2015 [cit. 2015-12-7]. Dostupné z: <http://armprogramming.com/stm32f4-discovery-tutorials-interruptexti/>
- [12] INDUCODER MESSTECHNIK GMBH. *Standard no shaft encoder EDH581*. Duisburg, Germany, 2003. Dostupné také z: <https://www.yumpu.com/fr/document/view/21901007/standard-no-shaft-encoder-edh-58-edh581-inducoder-messtechnik->
- [13] STMICROELECTRONICS. *UM1670 User manual: Discovery kit for STM32F429/439 lines*. USA, 2013. Dostupné také z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00093903.pdf
- [14] STMICROELECTRONICS. *STM32F4xx: Reference manual*. 2015. Dostupné také z: http://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00030003.pdf
- [15] STMICROELECTRONICS. *STM32F407xx: Datasheet*. 2016. Dostupné také z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/ef/92/76/6d/bb/c2/4f/f7/DM00037051.pdf/files/DM00037051.pdf/jcr:content/translations/en.DM00037051.pdf>

- [16] STMICROELECTRONICS. *MC34063AB: DC-DC converter control circuits: Datasheet*. 2013. Dostupné také z:
<http://www.st.com/content/ccc/resource/technical/document/datasheet/03/f9/c4/3d/7f/eb/4c/5e/CD00001232.pdf/files/CD00001232.pdf/jcr:content/translations/en.CD00001232.pdf>
- [17] STMICROELECTRONICS. *Oscillator design guide for STM8S, STM8A and STM32 microcontrollers: Application note*. 2015. Dostupné také z:
http://www.st.com/content/ccc/resource/technical/document/application_note/c6/eb/5e/11/e3/69/43/eb/CD00221665.pdf/files/CD00221665.pdf/jcr:content/translations/en.CD00221665.pdf
- [18] STMICROELECTRONICS. USBLC6-2: Very low capacitance ESD protection. 2011. Dostupné také z:
<http://www.st.com/content/ccc/resource/technical/document/datasheet/06/1d/48/9c/6c/20/4a/b2/CD00050750.pdf/files/CD00050750.pdf/jcr:content/translations/en.CD00050750.pdf>
- [19] STM32 Virtual COM Port Driver. STMicroelectronics [online]. [cit. 2016-05-10]. Dostupné z: http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stsw-stm32102.html
- [20] STM32CubeMX: STM32Cube initialization code generator. STMicroelectronics [online]. [cit. 2016-05-10]. Dostupné z: http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-configurators-and-code-generators/stm32cubemx.html
- [21] MATTKE AG. *Technische beschreibung für Mattke - analog - regelung: Typ MAR 30/2, MAR 25/2, MAR 12/3*. V 2.0. Freiburg, Germany, 2009. Dostupné také z: https://www.mattke.de/fileadmin/_migrated/content_uploads/MAR30_25_12_V2_0_A1_02_DE_S.pdf
- [22] TEXAS INSTRUMENTS. *TL43xx Precision Programmable Reference*. Dallas, Texas, USA, 2015. Dostupné také z: <http://www.ti.com/lit/ds/symlink/tl431.pdf>
- [23] TEXAS INSTRUMENTS. *RS-422 and RS-485 Standards Overview and System Configurations: Application Report*. Dallas, Texas, USA, 2010. Dostupné také z: <http://www.ti.com/lit/an/slla070d/slla070d.pdf>
- [24] MATTKE AG. *Technical Information For Transistor-Four-Quadrant Servo-Amplifier: Serie MAR 25/2*. Freiburg, Germany, 2005. Dostupné také z: https://www.mattke.de/fileadmin/_migrated/content_uploads/MAR25_2_GB_S_01.pdf
- [25] FTDI CHIP. *FT232R USB UART IC*. 2.13. Glasgow, UK, 2015. Dostupné také z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [26] TEXAS INSTRUMENTS. *SN65LBC176 Differential Bus Transceivers*. Dallas, Texas, USA, 2014. Dostupné také z: <http://www.ti.com/lit/ds/symlink/sn751bc176.pdf>
- [27] TEXAS INSTRUMENTS. *DS34LV86T 3V Enhanced CMOS Quad Differential Line Receiver: Data sheet*. 2013. Dostupné také z: <http://www.ti.com/lit/ds/symlink/ds34lv86t.pdf>
- [28] STMICROELECTRONICS. *STM32F429xx: Datasheet - production data*. 2015. Dostupné také z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00071990.pdf>
- [29] DANAHER CONTROLS. *Encoder Application Handbook: A reference guide and tutorial*

- on encoders for motion control: types, technologies, applications, and installations. Gurnee, Illinois, 2003. Dostupné také z: https://www.dynapar.com/uploadedFiles/Products/Danaher_Encoder_Handbook.pdf
- [30] DC motor. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2007, 5. 12. 2015 [cit. 2015-12-06]. Dostupné z: https://en.wikipedia.org/wiki/DC_motor
- [31] IQD. *12SMX: Quartz Crystal Specification*. 2016. Dostupné také z: <http://www.iqdfrequencyproducts.com/products/details/12smx-13-30.pdf>
- [32] ON SEMICONDUCTOR. *NCS20074: Operational Amplifier, Rail-to-Rail Output, 3MHz BW*. Denver, Colorado, USA, 2015. Dostupné také z: http://www.onsemi.com/pub_link/Collateral/NCS20071-D.PDF
- [33] ON SEMICONDUCTOR. *MC33375: 300 mA, Low Dropout Voltage Regulator with On/Off Control: Datasheet*. 2013. Dostupné také z: http://www.onsemi.com/pub_link/Collateral/MC33375-D.PDF
- [34] ON SEMICONDUCTOR. *Theory and Applications of the MC34063 and uA78S40 Switching Regulator Control Circuits: Application note*. 2013. Dostupné také z: http://www.onsemi.com/pub_link/Collateral/AN920-D.PDF
- [35] ON SEMICONDUCTOR. *MBRS2040LT3G: Surface Mount Schottky Power Rectifier: Data sheet*. 2012. Dostupné také z: http://www.onsemi.com/pub_link/Collateral/MBRS2040LT3-D.PDF
- [36] ON SEMICONDUCTOR. *MMBTA05L: Driver Transistors: Data sheet*. 2014. Dostupné také z: http://www.onsemi.com/pub_link/Collateral/MMBTA05LT1-D.PDF
- [37] USART. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2008, 30.11.2015 [cit. 2015-12-7]. Dostupné z: <https://cs.wikipedia.org/wiki/USART>
- [38] Math.NET Numerics 3.11.1. *NuGet Gallery* [online]. 2016 [cit. 2016-05-11]. Dostupné z: <https://www.nuget.org/packages/MathNet.Numerics/>
- [39] ReSharper: Visual Studio Extension for .NET Developers. *JetBrains* [online]. [cit. 2016-05-15]. Dostupné z: <https://www.jetbrains.com/resharper/>

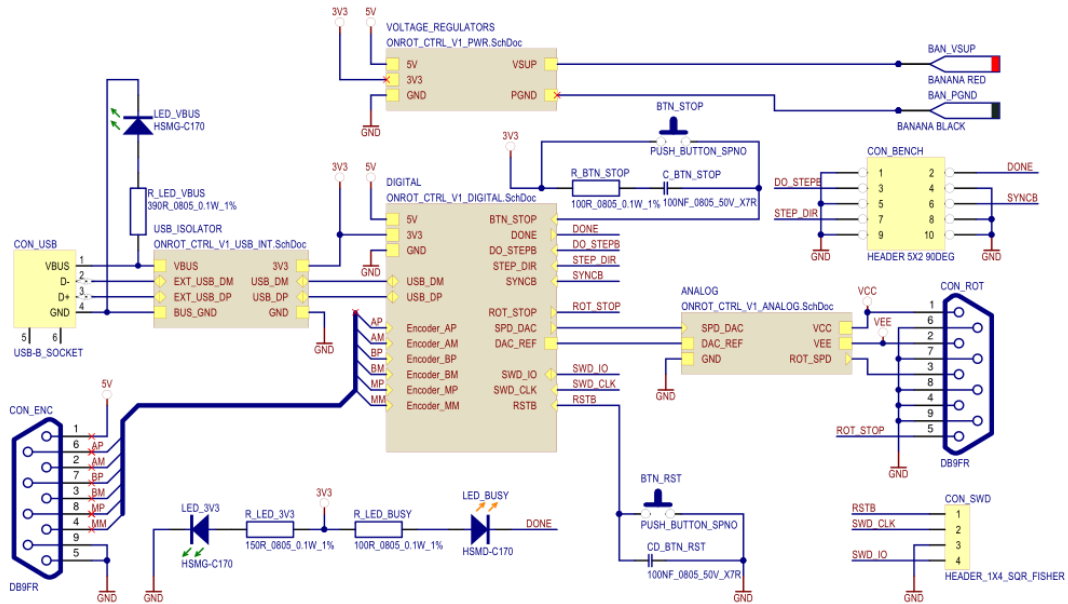
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ARM	Advanced RISC Machine, zdokonalený počítač typu RISC
B	Magnetická indukce [T]
Bd	Baudrate, symbolová rychlost sériové linky (1 b/s)
C	Velikost elektrické kapacity [F]
CCM	Core Coupled Memory, paměť sdílená s jádrem procesoru
DA	Digital/Analog, převodník z digitálního vyjádření na analogový signál
DAC	Datové slovo DA převodníku [-]
DC	Direct Current, stejnosměrný proud
emf	Electromotive Force, elektromotorická síla
ESR	Equivalent Series Resistance, ekvivalentní sériový odpor [Ω]
f	Frekvence, kmitočet [Hz]
FEMC	Flexible External Memory Controller, flexibilní kontrolér externí paměti
FPGA	Field Programmable Gate Array, programovatelné hradlové pole
G	Gain, zisk
GPIO	General Purpose Input Output, vstupně/výstupní rozhraní pro obecné použití
GUI	Graphical User Interface, grafické uživatelské rozhraní
h	Výška [m]
HAL	Hardware Abstraction Layer, abstraktní rozhraní HW nezávislé na výrobci
HW	Hardware, technické vybavení
I	Elektrický proud [A]
IRQ	Interrupt Request, žádost o přerušení programu (procesu)
ISR	Interrupt Service Routine, podprogram pro obsluhu přerušení
JTAG	Joint Test Action Group, rozhraní pro ladění MCU
K	Napěťový přenos [-]
k	Konstanta úměrnosti, napěťová konstanta [V/(rad/s)]
k_n	Konstanta úměrnosti otáček k napětí [(ot/min)/V]
$k_{n_digital}$	Konstanta úměrnosti otáček k digitálnímu kódu [(ot/min)/kód]
LDO	Low Dropout, regulátor napětí s nízkým úbytkem
m	Rozlišení DAC [b]
MC	Memory Capacity, kapacita paměťového média

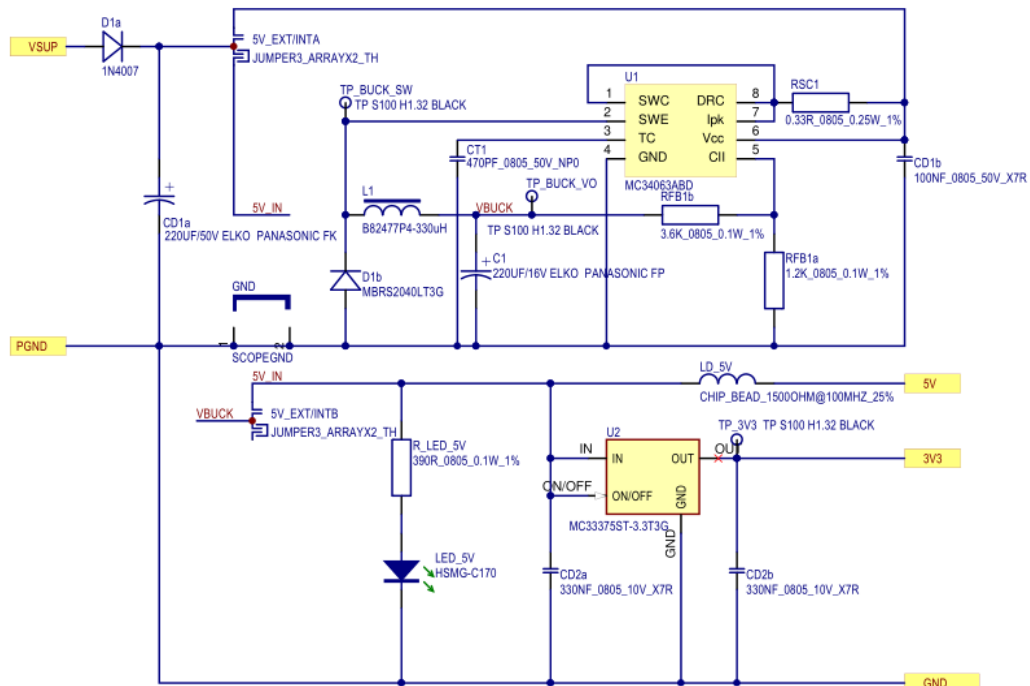
MCU	Micro Controller Unit, mikrokontrolér, mikrořadič, mikropočítač
N	Počet závitů vinutí [-]
n	Rychlost otáčení [ot/min]
NPN	Typ bipolárního tranzistoru duální k PNP
ot/min	Otáčky za minutu, jednotka rychlosti otáčení
OZ	Operační zesilovač
p	Koeficient udávající násobek zvýšení rozlišení převodníku [-]
PC	Personal Computer, osobní počítač
PID	Product Identification, identifikační číslo výrobku
PLL	Phase Locked Loop, smyčka fázového závěsu
PPR	Pulse Per Revolution, pulsy za jednu otáčku, jednotka rozlišení kodéru
r	Poloměr [m]
R	Elektrický odpor [Ω]
RAM	Random Acces Memory, paměť s volným přístupem
Res	Resolution, rozlišení kodéru [PPR]
RGB	Red Green Blue, analogový videosignál (červená, zelená a modrá)
RISC	Reduced Instruction Set Computer, počítač s redukovanou instrukční sadou
RS-232	Recommended Standard 232, doporučený standard č. 232 rozhraní sériového portu (podobně RS-422, RS-485)
SDRAM	Synchronous DRAM, synchronní DRAM, paměť pracující v PC na shodném hodinovém taktu jako procesor.
SMART	Self Monitoring Analysis and Reporting Technology, technologie sebe monitorování, analýzy a oznámení stavu
SRAM	Static RAM, statická paměť RAM, bez periodického obnovování informace jako u DRAM
SWD	Serial Wire Debug, sériové rozhraní pro ladění programu MCU nahrazující JTAG
T	Perioda [s]
TFT	Thin Film Transistor, tenkovrstvý tranzistor
u	Okamžitá hodnota napětí [V]
USART	Universal Synchronous/Asynchronous Receiver and Transmitter, univerzální synchronní a asynchronní přijímač a vysílač
USB	Universal Serial Bus, univerzální sériová sběrnice
VID	Vendor Identification, identifikační číslo výrobce
ω_c	Úhlová rychlost [rad/s]

A VÝROBNÍ PODKLADY DESKY PLOŠNÉHO SPOJE MODULU

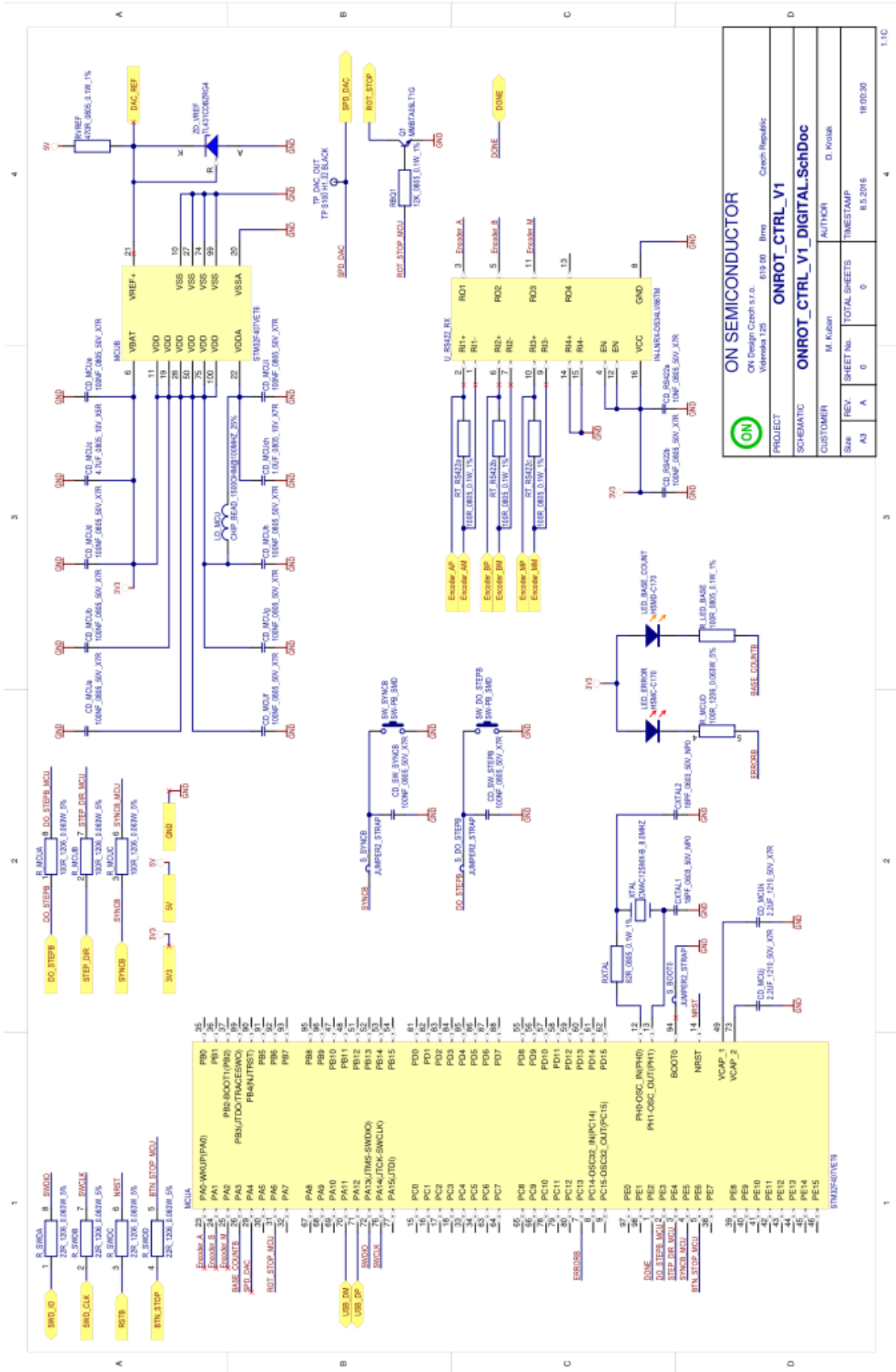
A.1 Celkové zapojení modulu



A.2 Obvodové zapojení napájecí části

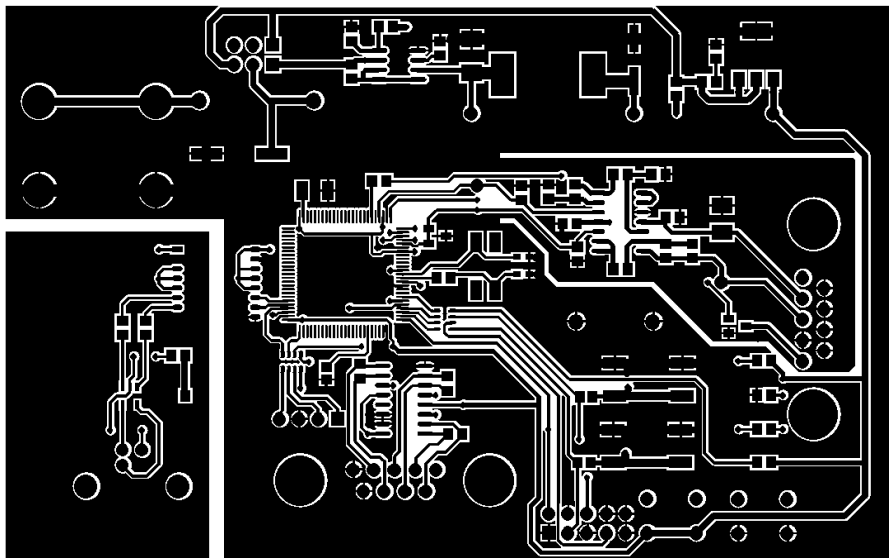


A.3 Obvodové zapojení digitální části



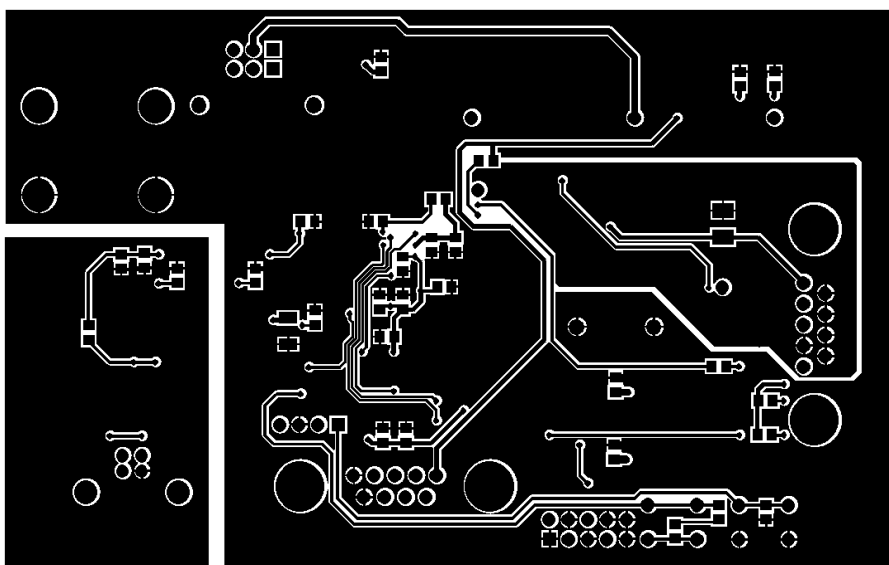
ON SEMICONDUCTOR	
ON Design Czech s.r.o.	619 00 Brno Czech Republic
Valteřská 128	
PROJECT ONROT_CTRL_V1_DIGITAL_SchDoc	
SCHEMATIC	
CUSTOMER M. Kuban	AUTHOR D. Kozák
REV. A	TOTAL SHEETS 0
SHEET No. A3	TIMESTAMP 8.5.2016
18:00:30	

A.4 Deska plošného spoje – strana součástek



Rozměry: 119 x 74 mm, M 1:1

A.5 Deska plošného spoje – strana spojů



Rozměry: 119 x 74 mm, M 1:1

A.8 Seznam součástek

Označení	Hodnota	Pouzdro	Popis
5V_INT/EXT	JUMPER3_ARRAYX2_TH	JUMPERARRAY_2X3	Zkratovací propojka
BAN_PGND	BANANA BLACK	BANANA_BLACK	Zdířka do DPS
BAN_VSUP	BANANA RED	BANANA_RED	Zdířka do DPS
BTN_RST	PUSH_BUTTON_SPNO	PB1720	Mikrospínač
BTN_STOP	PUSH_BUTTON_SPNO	PB1720	Mikrospínač
C1	220UF/16V ELKO PANASONIC FP	PANASONIC_FC_D8	Elektrolytický kondenzátor
CC_OPA	10PF_0805_50V_NP0	CAP0805	Keramický kondenzátor
CD1a	220UF/50V ELKO PANASONIC FK	PANASONIC_FC_G	Elektrolytický kondenzátor
CD1b	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD2a	330NF_0805_10V_X7R	CAP0805	Keramický kondenzátor
CD2b	330NF_0805_10V_X7R	CAP0805	Keramický kondenzátor
CDa_OPA	10UF/16V TANTAL SMD A	AVX_TAJB	Tantalový kondenzátor
CDb_OPA	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CDc_OPA	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CDd_OPA	10UF/16V TANTAL SMD A	AVX_TAJB	Tantalový kondenzátor
CD_BTN_RST	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_ISOa	4.7UF_0805_10V_X5R	CAP0805	Keramický kondenzátor
CD_ISOb	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_ISOc	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_ISOd	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUa	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUb	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUc	4.7UF_0805_10V_X5R	CAP0805	Keramický kondenzátor
CD_MCUch	1.0UF_0805_10V_X7R	CAP0805	Keramický kondenzátor

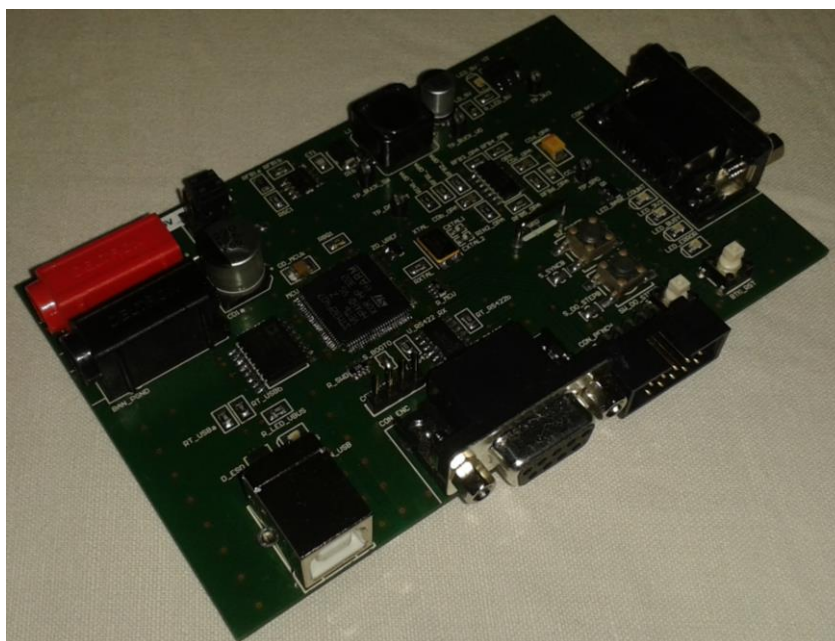
Označení	Hodnota	Pouzdro	Popis
CD_MCUd	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUe	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUf	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUg	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUh	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUi	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_MCUj	2.2UF_1210_50V_X7R	CAP1210	Keramický kondenzátor
CD_MCUk	2.2UF_1210_50V_X7R	CAP1210	Keramický kondenzátor
CD_RS422a	10NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_RS422b	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_SW_STEPB	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CD_SW_SYNCB	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
CON_BENCH	HEADER 5X2 90DEG	X10ABPH_90deg	Konektor
CON_ENC	DB9FR	DB9FR	Konektor D-sub 9
CON_ROT	DB9FR	DB9FR	Konektor D-sub 9
CON_SWD	HEADER_1X4_SQR_FISHER	X4APH	Konektor
CON_USB	USB-B_SOCKET	USB-B_SOCKET	USB konektor
CT1	470PF_0805_50V_NP0	CAP0805	Keramický kondenzátor
CXTAL1	18PF_0603_50V_NP0	CAP0603	Keramický kondenzátor
CXTAL2	18PF_0603_50V_NP0	CAP0603	Keramický kondenzátor
C_BTN_STOP	100NF_0805_50V_X7R	CAP0805	Keramický kondenzátor
D1a	1N4007	DO-41	Polovodičová dioda
D1b	MBRS2040LT3G	SMB_MURS120T3G	Schottkyho dioda
D_ESD	USBLC6-2SC6	SOT23_6L	ESD protekce
GND	SCOPEGND	SCOPEGND	Zemní vývod
L1	B82477P4-330uH	12.5X12.5MM	Cívka s feritovým jádrem
LD_5V	CHIP_BEAD_1500OHM@100MHz	L0805	Feritová perla
LD_MCU	CHIP_BEAD_1500OHM@100MHz	L0805	Feritová perla
LD_VBUS	CHIP_BEAD_1500OHM@100MHz	L0805	Feritová perla

Označení	Hodnota	Pouzdro	Popis
LED_3V3	HSMG-C170	D0805	Zelená LED dioda
LED_5V	HSMG-C170	D0805	Zelená LED dioda
LED_BASE_COUNT	HSMD-C170	D0805	Oranžová LED dioda
LED_BUSY	HSMD-C170	D0805	Oranžová LED dioda
LED_ERROR	HSMC-C170	D0805	Červená LED dioda
LED_VBUS	HSMG-C170	D0805	Zelená LED dioda
MCU	STM32F407VET6	STM-LQFP100_N	MCU
OPA	NCS20074	SOIC14	Operační zesilovač 4x
Q1	MMBTA05LT1G	SOT23-BEC	Bipolární tranzistor
RBQ1	12K_0805_0.1W_1%	R0805	Rezistor
RFB1a	1.2K_0805_0.1W_1%	R0805	Rezistor
RFB1b	3.6K_0805_0.1W_1%	R0805	Rezistor
RFB1_OPA	8.2K_0805_0.1W_1%	R0805	Rezistor
RFB2_OPA	8.2K_0805_0.1W_1%	R0805	Rezistor
RFB3_OPA	8.2K_0805_0.1W_1%	R0805	Rezistor
RFB4_OPA	33K_0805_0.1W_1%	R0805	Rezistor
RFB5_OPA	8.2K_0805_0.1W_1%	R0805	Rezistor
RFB6_OPA	33K_0805_0.1W_1%	R0805	Rezistor
RIN1_OPA	1.5MEG_0805_0.125W_1%	R0805	Rezistor
RIN2_OPA	1.5MEG_0805_0.125W_1%	R0805	Rezistor
RSC1	0.33R_0805_0.25W_1%	R0805	Rezistor
RT_RS422a	100R_0805_0.1W_1%	R0805	Rezistor
RT_RS422b	100R_0805_0.1W_1%	R0805	Rezistor
RT_RS422c	100R_0805_0.1W_1%	R0805	Rezistor
RT_USBa	24R_0805_0.1W_1%	R0805	Rezistor
RT_USBb	24R_0805_0.1W_1%	R0805	Rezistor
RVREF	470R_0805_0.1W_1%	R0805	Rezistor
RXTAL	82R_0805_0.1W_1%	R0805	Rezistor
R_BTN_STOP	100R_0805_0.1W_1%	R0805	Rezistor
R_LED_3V3	150R_0805_0.1W_1%	R0805	Rezistor
R_LED_5V	390R_0805_0.1W_1%	R0805	Rezistor
R_LED_BASE	100R_0805_0.1W_1%	R0805	Rezistor
R_LED_BUSY	100R_0805_0.1W_1%	R0805	Rezistor
R_LED_VBUS	390R_0805_0.1W_1%	R0805	Rezistor
R_MCU	100R_1206_0.063W_5%	CRA3A4E	Rezistorové pole
R_SWD	22R_1206_0.063W_5%	CRA3A4E	Rezistorové pole
SW_DO_STEPB	SW-PB_SMD	PB300	Mikrospínač
SW_SYNCB	SW-PB_SMD	PB300	Mikrospínač
TP_3V3	TP S100 H1.32 BLACK	TESTPIN1	Testovací vývod

Označení	Hodnota	Pouzdro	Popis
TP_BUCK_SW	TP S100 H1.32 BLACK	TESTPIN1	Testovací vývod
TP_BUCK_VO	TP S100 H1.32 BLACK	TESTPIN1	Testovací vývod
TP_DAC_OUT	TP S100 H1.32 BLACK	TESTPIN1	Testovací vývod
TP_SPD	TP S100 H1.32 BLACK	TESTPIN1	Testovací vývod
U1	MC34063ABD	SOIC8	DC/DC konvertor
U2	MC33375ST-3.3T3G	SOT223	LDO regulátor
USB_ISO	ADUM3160ARWZ	SOIC16	USB izolátor
U_RS422_RX	IN-LNRX-DS34LV86TM	SOIC16	RS-422 přijímač 4x
XTAL	CMAC12SMX-B_8.0MHZ	CMAC_12SMX-B	Krystal
ZD_VREF	TL431AIDBZT	SOT23-KRA	Nastavitelná Zenerová dioda

B FOTOGRAFIE FINÁLNÍ VERZE MODULU

B.1 Pohled ze strany součástek



B.2 Pohled ze strany spojů

