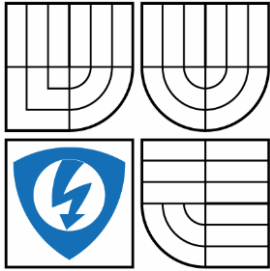


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁVRH ŘÍDICÍ DESKY PRO BLDC MOTOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Zbranek Petr

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Veselý Libor, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Petr Zbranek

ID: 119684

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Návrh řídicí desky pro BLDC motor

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh řídicí desky pro BLDC motor. Zadání lze shrnout do následujících bodů:

1. Zpracujte rešerši týkající se BLDC motorů a jejich možnosti řízení.
2. Navrhněte a odzkoušejte algoritmus řízení BLDC motorů na platformě cRIO se stávajícím hardware.
3. Navrhněte vlastní hardware řídicí desky pro BLDC motor.
4. Implementujte vybraný algoritmus do řídicí jednotky.

DOPORUČENÁ LITERATURA:

Skalický, J., Patočka, M., Feiler, Z.: Elektrické pohony a výkonová elektronika. Brno. Vysoké učení technické v Brně, 2006. s. 1-237. ISBN: 80-214-3286-1.

Termín zadání: 10.2.2014

Termín odevzdání: 19.5.2014

Vedoucí práce: Ing. Libor Veselý, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

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

V rámci této diplomové práce byly řešeny algoritmy a hardware pro řízení BLDC motoru. Jako první byl navržen pro řízení motoru v LabView algoritmus nazývaný „Six-step“. Ten ovšem způsobuje zvlnění momentu, proto bylo ve finálním řešení použito vektorové řízení. Jde o senzorové řízení, proto bylo nezbytné navrhnout měření pozice rotoru. To bylo vyřešeno pomocí magnetického snímače polohy. Po otestování řídicího algoritmu byl navržen vlastní hardware pro řízení BLDC motoru. Hlavní řídicí jednotkou byl zvolen mikroprocesor MC56F82743, který se používá přímo pro řízení BLDC motorů. Mikroprocesor byl naprogramován ve vývojovém prostředí CodeWarrior, pomocí USB-TAP přes JTAG.

Klíčová slova

BLDC motor, Senzorové řízení motorů, Mikroprocesor MC56F82743, CodeWarrior

Abstract

In the context of this master's thesis are solved algorithms and hardware for control BLDC motor. The first was designed algorithm in LabView called "six-step". This algorithm causes a torque ripple, so in the final version was used vector control. It is a sensor control, so was necessary to design a measurement of rotor position. It was solved using a magnetic position sensor. After testing the control algorithm was designed my own hardware for control the BLDC motor. The main control unit was chosen microprocessor MC56F82743, which is designed specifically for BLDC motor control. The microprocessor was programmed in the CodeWarrior development environment, using a JTAG and USB-TAP.

Keywords

BLDC motor, Sensor motor control, Microprocessor MC56F82743, CodeWarrior

Bibliografická citace:

Zbranek, P. *Návrh řídicí desky pro BLDC motor*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 76s. Vedoucí diplomové práce byl Ing. Veselý Libor, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Návrh řídicí desky jsem vypracoval samostatně, pod vedením vedoucího diplomové práce, 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 jsem si 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.

V Brně dne: **15. května 2014**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Liborovi Veselému, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **15. května 2014**

.....
podpis autora

Obsah

1	Úvod.....	9
2	Motor BLDC.....	11
2.1	Základní rozdělení elektrických pohonů.....	11
2.2	Princip BLDC motoru.....	12
2.3	Konstrukční principy.....	13
2.3.1	Stator.....	13
2.3.2	Rotor.....	13
2.4	Charakteristika točivého momentu v závislosti na rychlosti.....	14
2.5	Řízení BLDC motoru.....	15
2.6	Výhody a nevýhody BLDC motoru.....	17
2.7	Srovnání BLDC s jinými motory.....	17
3	Návrh řízení.....	19
3.1	Řízení motoru pomocí lichoběžníkového signálu.....	19
3.1.1	Vytvoření komutační tabulky.....	19
3.1.2	PWM signál.....	20
3.1.3	Ošetření Dead-time.....	21
3.1.4	Proudová a rychlostní regulace.....	22
3.2	Skalární řízení.....	23
3.3	Vektorové řízení.....	24
3.3.1	Clarkové transformace.....	25
3.3.2	Parkova transformace.....	25
3.4	Snímání polohy rotoru BLDC motoru.....	26
3.4.1	Komunikace AS5048.....	28
3.4.2	Výběr Magnetu.....	30
3.4.3	Měření AS5048.....	30
4	Návrh hardwaru.....	32
4.1	Výkonová část.....	32
4.1.1	Návrh měniče.....	32
4.1.2	Návrh budiče pro tranzistory.....	33
4.1.3	Měření proudu.....	36

4.1.4	Stabilizace napětí	38
4.2	Řídicí část - Mikroprocesor 56F82743	40
4.2.1	Základní údaje.....	40
4.2.2	ADC – Analog-to-Digital Converter	42
4.2.3	PIT – Programmable Interval Timer.....	43
4.2.4	PWM.....	44
4.2.5	Zapojení mikroprocesoru MC56F82743	46
5	Návrh softwaru	47
5.1	CodeWarrior.....	47
5.2	Quick Start	48
5.2.1	Freemaster.....	49
5.3	Popis řídicího programu a jeho funkcí	51
6	Realizace	56
6.1	BLDC motoru Multistar.....	56
6.1.1	Parametry udávané výrobcem.....	56
6.1.2	Měřené parametry	56
6.2	Vytvořený Hardware.....	58
6.3	Průběhy a řídicí signály.....	60
7	Závěr	64
8	Seznamy.....	66
8.1	Seznam použitých součástek.....	66
8.2	Seznam obrázků	67
8.3	Seznam tabulek	69
8.4	Seznam použité literatury.....	70
8.5	Seznam příloženého CD.....	72
8.6	Seznam Příloh	72
	Přílohy.....	73

1 ÚVOD

Tato diplomová práce se zabývá návrhem řídicího algoritmu a návrhem hardwaru pro řízení motoru, který bude sloužit k pohánění robotického vozítka pro robotický fotbal. Jako poháněcí motor byl vybrán BLDC zejména kvůli jeho jednoduchosti, cenové dostupnosti a jeho spolehlivosti. Bezkartáčový stejnosměrný motor (BLDC) je navržen pro nízké výkonové řízení, vzhledem k jeho vysoké účinnosti, tichému chodu a nízkým požadavkům na údržbu. Vzhledem k nutnosti použití nákladnějšího elektronického střídače a složitosti ovládání proměnné rychlosti se stále udržují v oběhu rozšířené kartáčové stejnosměrné motory.

Pravidla robotického fotbalu určují maximální velikost robota na 75 x 75 x 75mm, proto je potřeba navrhnout hardware co nejmenší velikosti. Požadavkem také je, aby se robotické vozítko pohybovalo rychlostí 4 m/s, což při námi zvoleném průměru kola 34mm je přibližně 38 otáček za vteřinu.

Diplomová práce je rozdělena do několika částí. V první polovině práce je rozebrán teoretický rozbor o vlastnostech BLDC a jejich řízení.

BLDC motory lze řídit několika způsoby, jako první je aplikováno řízení pomocí lichoběžníkového signálu, toto řízení se nazývá „Six-step“. Zde je první popsána a vytvořena komutační tabulka, podle které se určuje spínání fázi elektronického střídače. K řízení otáček a k řízení proudu byl využit PI regulátor. Jako druhá možnost řízení bylo použito vektorové řízení. Cílem vektorového řízení je dosáhnout nezávislého řízení jak momentu tak buzení a tím dosáhnout chování jako u stejnosměrného motoru s cizím buzením.

Pro správné řízení je potřeba znát polohu rotoru. Techniky pro detekci a určování této polohy jsou popsány v následující kapitole, která se zabývá výhodami a nevýhodami jednotlivých možností. Pro snímání byly vybrány magnetické senzory polohy, které jsme od firmy AMS dostali zdarma jako zkušební vzorky.

Druhá polovina práce je věnována návrhu hardwaru a je rozdělena do dvou částí, na výkonovou a řídicí.

Ve výkonové části je první vysvětleno, jak byl navrhnout třífázový měnič, budiče pro MOSFET tranzistory, stabilizátor a měření proudu. Při výběru MOSFET tranzistoru je důležité zvolit tranzistor s rychlým spínáním, aby docházelo co k nejmenším ztrátám.

Ovšem ještě důležitější je navrhnout správně budiče pro tranzistory, protože horní tranzistor není pinem SOURCE připojen na zem ale na fázi motoru, proto je potřeba pro jeho otvírání dosáhnout napájení větší než je napájecí napětí. K tomu se používá budič s nábojovou pumpou. Pro přesnou regulaci momentu je potřeba přesně znát proud protékající motorem, tento proud je tedy potřeba měřit. V úvahu připadají dvě možnosti, které jsou obě v této práci popsány. První možnost je pomocí měřícího odporu neboli bočnicku, na kterém lze měřit úbytek napětí, který je závislý na proudu. Druhá možnost je použití snímače proudu, založeném na Hallovém efektu. Snímač vrací výstupní napětí v rozsahu přímo pro mikroprocesor 0–3,3V, tedy tento signál už není potřeba nijak upravovat.

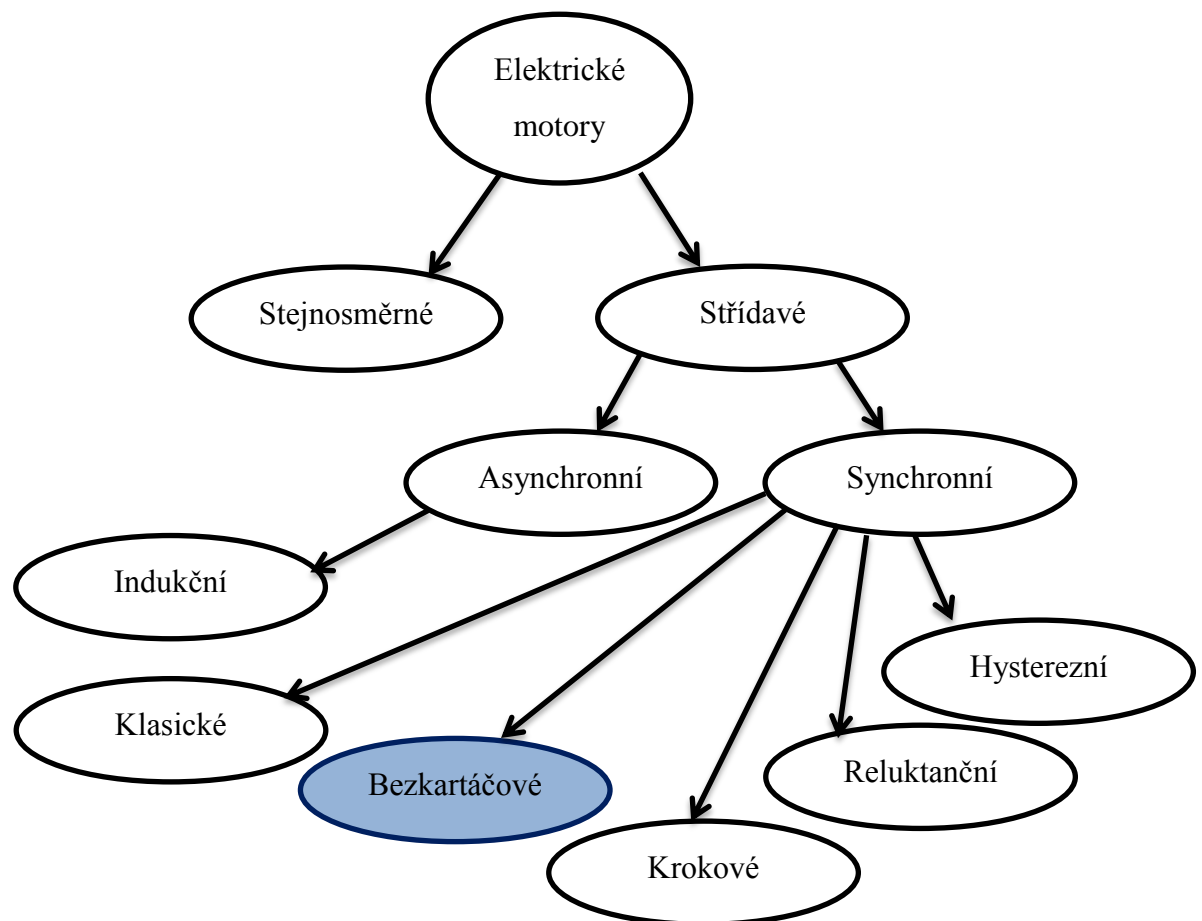
V řídicí části jsou popsány důležité periferie mikroprocesoru a jeho zapojení. Mikroprocesor je programován ve vývojovém prostředí CodeWarrior, který využívá další nástroje k zjednodušení a urychlení práce. Pro nastavení periférií se používá Quick Start, v kterém i bez větších znalostí struktury mikropočítače lze snadno nastavit periferie mikroprocesoru. Pro real-time debuggování se používá Freemaster, kde proměnné lze sledovat ale také jich nastavovat a měnit.

V poslední části je popsána realizace, kde jsou ukázány vlastnosti a obrázky použitého motoru a vytvořeného hardwaru. Jsou zde také popsány a zobrazeny jednotlivé průběhy a řídicí signály, které čtenáři poskytnou větší přehled o dané problematice řízení motorů.

2 MOTOR BLDC

2.1 Základní rozdělení elektrických pohonů

Elektrické pohony využívají elektromagnetické indukce k převodu elektrické energie na energii mechanickou. Elektrické pohony lze dělit podle mnoha kritérií, například jako točivé a netočivé. My se zaměříme hlavně na točivé, které lze dělit podle typu motoru. Základním parametrem je způsob napájení. Podle toho motory rozdělujeme na stejnosměrné a střídavé. Střídavé motory lze rozdělit na synchronní a asynchronní, kde BLDC motor spadá pod synchronní bezkartáčové motory.

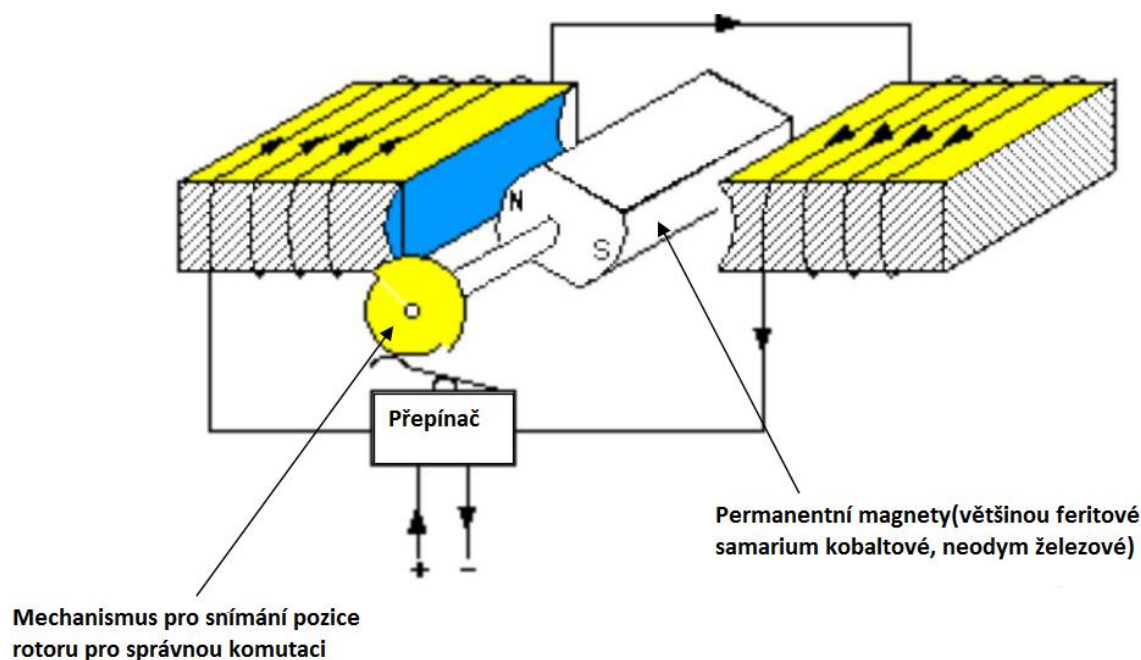


Obrázek 1 - Základní rozdělení elektrických pohonů

BLDC neboli Brushless DC Motor je bezkartáčový elektronicky komutovaný DC motor, který se používá většinou v jednoduchých aplikacích. Jde o motor s jednoduchou konstrukcí a snadným řízením, který je cenově dostupný a jeho spolehlivost vytlačila klasické stejnosměrné motory.

2.2 Princip BLDC motoru

BLDC motor je konstrukčně tvořen státorem, na kterém je navinuto většinou třífázové vinutí. Tyto motory mohou být 2-pólové nebo n-pólové. Rotor tvoří permanentní magnety, které ve vzduchové mezeře vytváří permanentní magnetické pole. BLDC motory neobsahují mechanický komutátor, který je u běžných stejnosměrných motorů zdrojem rušení, poruch a příčinou nižší účinnosti. Místo toho je přepínání jednotlivých vinutí řešeno elektronicky. Komutace musí být prováděna ve správném okamžiku a k tomu je potřeba znát přesnou polohu rotoru. K tomu se většinou využívají Hallovy snímače nebo lze i použít „sensorless řízení“ neboli bezsnímačové řízení. BLDC motory jsou napájeny spínaným zdrojem napětí na satorová vinutí motoru, spínací sekvence se zjišťují pomocí polohy rotoru. [1]
Základní principiální schéma můžeme vidět na obrázku 1 níže.



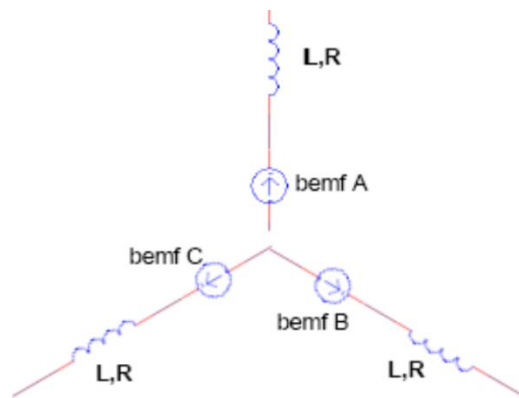
Obrázek 2 - Princip BLDC motoru, [1]

2.3 Konstrukční principy

2.3.1 Stator

Základem BLDC motoru jsou statorové vinutí. Počet vinutí bývá u většiny motorů replikován pro menší zvlnění momentu. Vinutí motoru může být zapojeno do trojúhelníku (Δ) nebo do hvězdy (Y). [2]

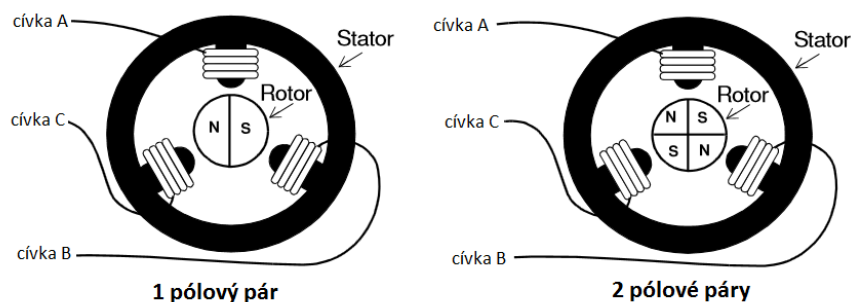
Na obrázku 2 můžeme vidět elektrické schéma statoru zapojené do hvězdy, které se skládá ze tří vinutí. Každé vinutí obsahuje tři prvky v sérii, odpor, indukci a zpětnou elektromotorickou sílu.



Obrázek 3 - Elektrické schéma statoru, [2]

2.3.2 Rotor

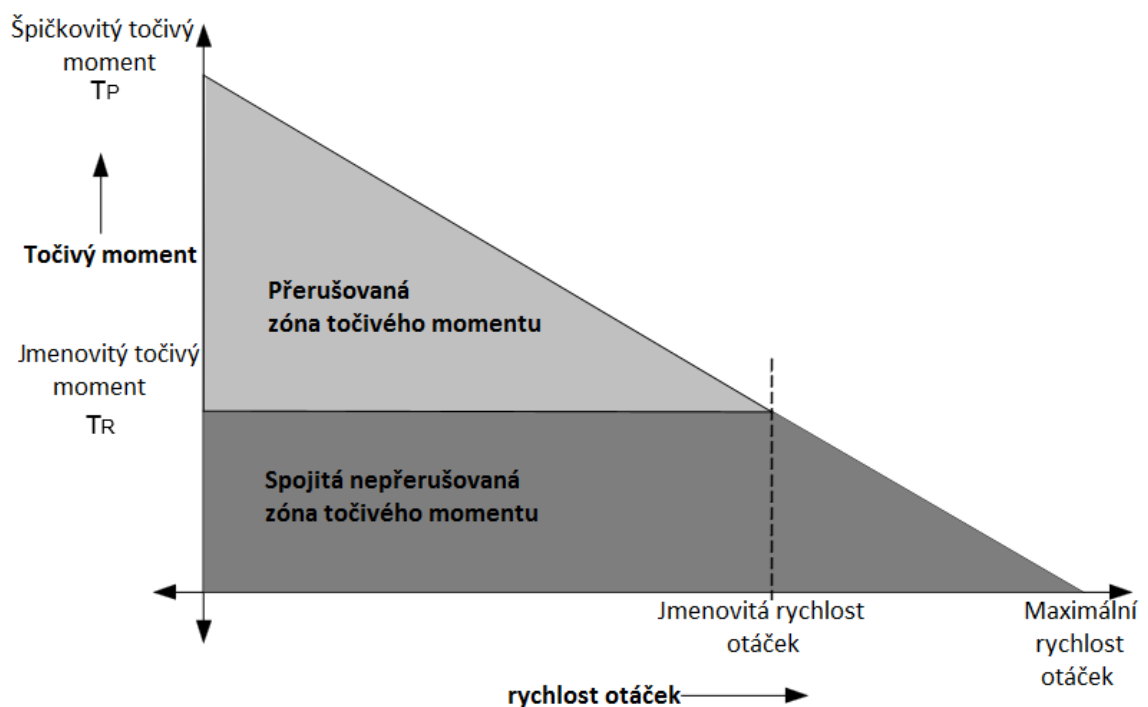
Rotor BLDC motoru se skládá ze sudého počtu permanentních magnetů. Počet magnetických pólů v rotoru ovlivňuje velikost kroku a tím tedy i zvlnění momentu. Čím více máme pólů, tím menší je krok a tím menší zvlnění momentu. BLDC motory většinou mívají 1–5 pólových párů, ovšem v některých případech jich můžou mít až 8. [2]



Obrázek 4 - Rotor a stator BLDC motoru, [2]

2.4 Charakteristika točivého momentu v závislosti na rychlosti

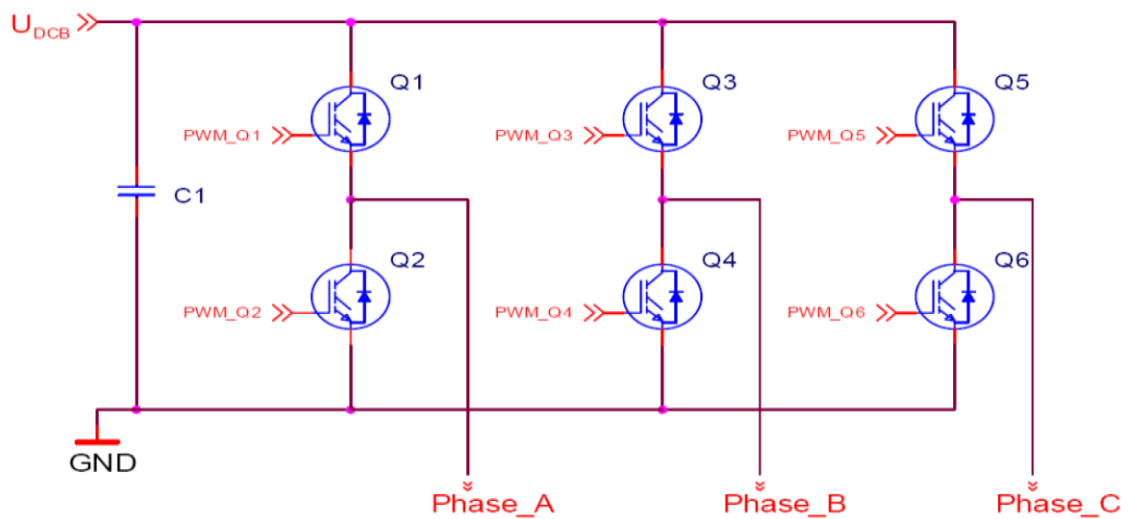
Na obrázku 4 níže, můžeme vidět charakteristiku točivého momentu v závislosti na rychlosti. K definování BLDC motoru slouží dva parametry točivého momentu, špičkový točivý moment (TP- Torque Peak) a jmenovitý točivý moment (TR- Torque rated). Motor můžeme zatížit při trvalém provozu v závislosti na jmenovitém momentu, který je konstantní do určitých otáček. Motor může běžet maximální rychlostí odpovídající až 150% jmenovitých otáček, ovšem točivý moment značně klesne. Některé aplikace vyžadují časté otáčení rotace nebo prudké zrychlování a zpomalování, což požaduje velký točivý moment pro překonání momentu setrvačnosti rotoru nebo zátěže. Tyto motoru mohou poskytnout vysoký točivý moment, ovšem jak vyplývá z charakteristiky pouze při nízkých rychlostech. [3]



Obrázek 5 - Charakteristika točivého momentu v závislosti na rychlosti, [3]

2.5 Řízení BLDC motoru

Základní řídicí algoritmus se nazývá „Six-step“ neboli šestikrokové řízení. Využívá se toho, že k otáčení rotoru BLDC motoru je zapotřebí elektronicky komutovat jednotlivé fáze statorového vinutí a tím dosáhnout ve vzduchové mezeře magnetického pole a elektromagnetického momentu. Jinak řečeno je potřeba ovládat všechny tři fáze motoru, k tomu se využívá 3-fázový střídač. [1]

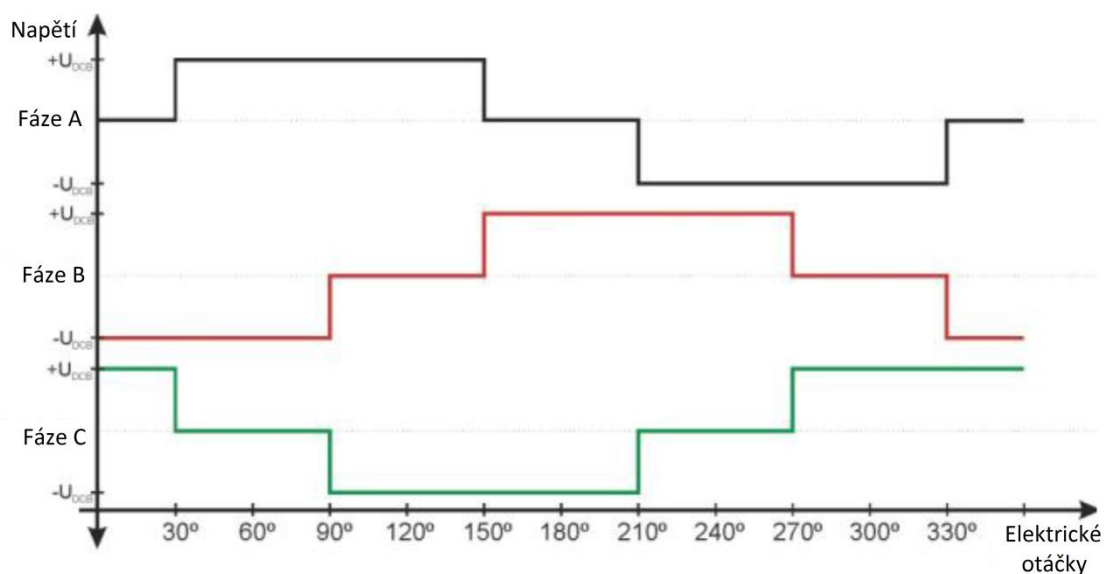


Obrázek 6 - třífázový střídač, [1]

Třífázový střídač se skládá ze tří větví Phase_A, Phase_B a Phase_C, v každé větvi po dvou tranzistorech, tedy celkem ze šesti tranzistorů.

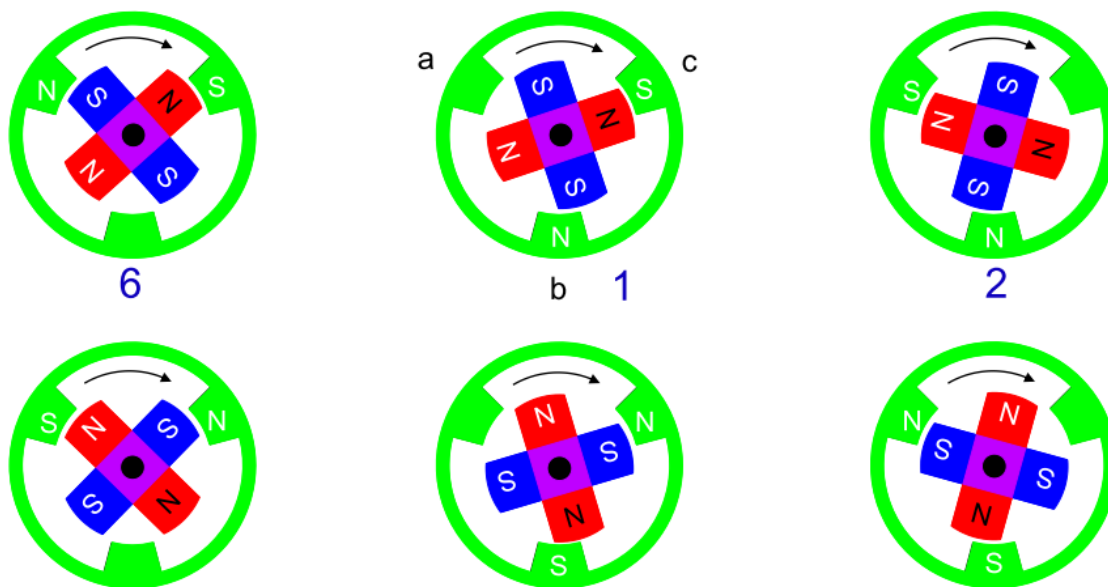
Six step komutace vždy napájí pouze dvě fáze motoru. Fáze jsou napájeny na základě informace o aktuální poloze a požadovaného směru. Nevýhodou tohoto způsobu komutace je, že na elektrickou otáčku generujeme pouze 6 spínacích vzorů, čímž měníme vektor pole po 60° elektrických, což způsobuje zvlnění momentu.

Na obrázku 7 jsou znázorněny potřebné spínací vzory jednotlivých fází.



Obrázek 7 - Spínací vzory pro jednu elektrickou otáčku, [1]

Na obrázku 8 je zobrazeno grafické znázornění polohy rotoru a k tomu magnetické pole statoru pro jednu elektrickou otáčku.



Obrázek 8 - Komutace BLDC motoru pro jednu elektrickou otáčku, [1]

2.6 Výhody a nevýhody BLDC motoru

Výhodou BLDC motoru je vysoká výkonová hustota, tedy poměr výkon/velikost nebo výkon/hmotnost. Výkon BLDC motoru je při stejné velikosti vyšší než u klasického synchronního motoru, přitom cena je nižší. Proti stejnosměrným motorům neobsahuje kartáče a komutátory, čímž dosahujeme dlouhé životnosti bez údržby.

Podstatnou nevýhodou BLDC motoru jak již bylo řečeno je zvlnění momentu. Teoreticky by moment měl být konstantní, ovšem z důvodu nenulové indukčnosti vinutí nedochází při přepínání fází ke skokovému nárůstu a poklesu proudu, tedy změny proudu jsou pozvolné a tím není moment konstantní. [4] [5]

2.7 Srovnání BLDC s jinými motory

Porovnání motorů je zobrazeno v tabulkách. V tabulce 1 je porovnán BLDC motor s kartáčovým DC motorem a v tabulce 2 je porovnán BLDC motor s AC indukčním motorem.

Tabulka 1 - Porovnání vlastností BLDC s kartáčovým DC motorem

Funkce	BLDC Motor	Kartáčový DC motor
Komutace	Elektronická komutace	Kartáčová komutace
Údržba	Málo nutná	Vyžaduje se periodicky
Životnost	Dlouhá	Krátká
Točivý moment/rychlost	Umožňuje provoz ve všech rychlostech	Při vyšších rychlostech snižuje kartáč tření, čímž se snižuje moment
Setrvačnost rotoru	Nízká, protože má permanentní magnety na rotoru	Vyšší setrvačnost, která limituje dynamickou charakteristiku
Rychlostní rozsah	Vysoký	Nízký, omezený kartáči
Generované rušení	Nízké	Vyšší, oblouky a tření na kartáči způsobuje rušení
Náklady	Vysoké, permanentní magnety zvyšují cenu	Nízké
Řízení	Složité a nákladné	Jednoduché a levné
Požadavky na řízení	Vyžaduje se mikroprocesor	Mikroprocesor je požadován pouze pro proměnnou rychlost
Efektivita	Vysoká	Nízká

Tabulka 2 - Porovnání vlastností BLDC a AC indukčního motoru

Funkce	BLDC Motor	AC indukční motor
točivý moment/rychlost	Umožňuje provoz na všech rychlostech při jmenovité zátěži	Nelineární, Má nižší točivý moment při nižších rychlostech
setrvačnost rotoru	Nízká, má lepší dynamickou charakteristiku	Vysoká, má špatnou dynamickou charakteristiku
výstupní výkon/velikost	Vlivem permanentních magnetů rotoru je dosaženo menších rozměrů pro daný výkon	Rotor i stator mají vinutí, výstupní výkon je menší než u BLDC
startovací proud	Není potřeba speciální startér	Používá se trojúhelník, hvězda startér
požadavky na řízení	Vyžaduje se mikroprocesor	Mikroprocesor je požadován pouze pro proměnnou rychlost

3 NÁVRH ŘÍZENÍ

3.1 Řízení motoru pomocí lichoběžníkového signálu

První algoritmus pro řízení motoru byl vytvořen pro školní laboratorní motor. Jednalo se o řízení nazývané „Six-step“ řízení. Poloha rotoru byla určována pomocí tří Hallových senzorů, které byly spojeny s rotorem. Řízení pomocí Hallových sond je jedna z nejjednodušších a nejběžnějších způsobů jak řídit BLDC motor. Z Hallových snímačů lze rozpoznat přesnou polohu rotoru s přesností $\pm 30^\circ$ elektrických. Tento motor obsahoval dvě elektrické otáčky na jednu mechanickou, tedy má 2 pólové dvojice a pomocí Hallových snímačů lze tedy určit polohu rotoru s přesností $\pm 15^\circ$ mechanických. Řídicí algoritmus snímá informace z Hallových snímačů a na jejich základech pak volí odpovídající kombinace pro spínání jednotlivých tranzistorů třífázového měniče. Jako první je pro řízení motoru potřeba si vytvořit komutační tabulku.

3.1.1 Vytvoření komutační tabulky

Pro vytvoření komutační tabulky byla jedna fáze motoru připojena na kladné napájecí napětí, zbylé dvě fáze byly připojeny na záporné napětí. Motor se po připojení napětí natočí a je potřeba zapsat údaje s Hallových snímačů. Potom se provede přehození kladného a záporného napájecího napětí podle tabulky a zapíšu se informace ze snímačů, tak jak ukazuje následná tabulka 3 níže.

Tabulka 3 - Komutační tabulka

Fáze			Hall snímače		
A	B	C	A	B	C
+	-	-	1	0	1
+	+	-	0	0	1
-	+	-	0	1	1
-	+	+	0	1	0
-	-	+	1	1	0
+	-	+	1	0	0

Po vytvoření předešlé tabulky víme, jak se mění poloha rotoru v závislosti na připojení fází. Následující tabulka 4 ukazuje, jak je potřeba spínat jednotlivé fáze pro točení rotoru.

Tabulka 4 - Rozšířená komutační tabulka

Hall snímače			Spínání Fází		
A	B	C	A	B	C
1	0	1	NC	+	-
0	0	1	-	+	NC
0	1	1	-	NC	+
0	1	0	NC	-	+
1	1	0	+	-	NC
1	0	0	+	NC	-

Kde: NC – not connected (není připojeno)

Teď víme jak spínat jednotlivé tranzistory na základě údajů z Hallových snímačů. Například při informaci z Hallových senzorů 1–0–1 víme, že pro otočení motoru doprava je potřeba fázi A nepřipojovat, tedy horní a spodní tranzistor fáze A bude zavřen. Na fázi B je plus, tedy horní tranzistor B bude otevřen a spodní B zavřen. Na poslední fázi C je mínus, tedy horní tranzistor C bude zavřena a spodní tranzistor C bude střídavě otevírán a zavírán pomocí modulovaného signálu PWM.

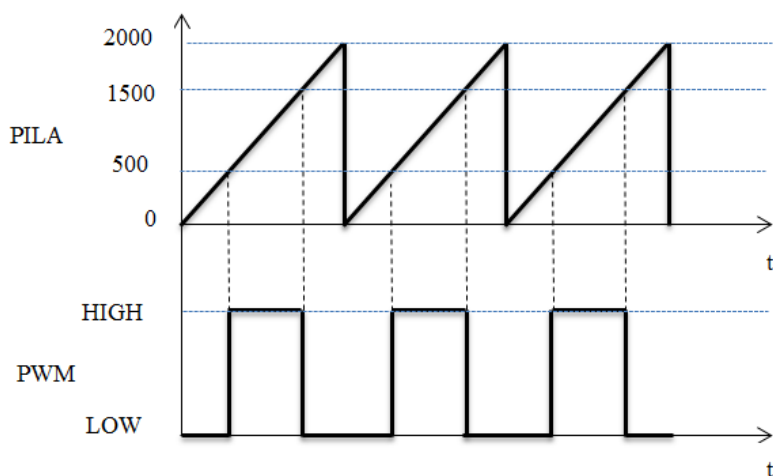
3.1.2 PWM signál

PWM signál neboli pulzně šířková modulace nám umožňuje měnit střední hodnotu signálu pomocí změny jeho střídy. Maximální střední hodnota může maximálně dosáhnout pouze velikost maximálního napájecího napětí.

Střída signálu, anglicky označovaný jako duty cycle, nám označuje poměr času, ve kterých se nám střídají jednotlivé úrovně, tedy logická jednička a nula. Pro střidu 50% bude tedy poměr logické jedničky a logické nuly 1:1 to pro velikost signálu značí půlku napájecího napětí.

V LabView je pro FPGA PWM signál vytvořen pomocí volně běžící pily s rozsahem hodnot 0–2000. Při rychlosti přičítání jedničky každých 40 MHz je tímto vytvořen modulační signál o rychlosti 20 KHz s přesností přibližně 11 bitů.

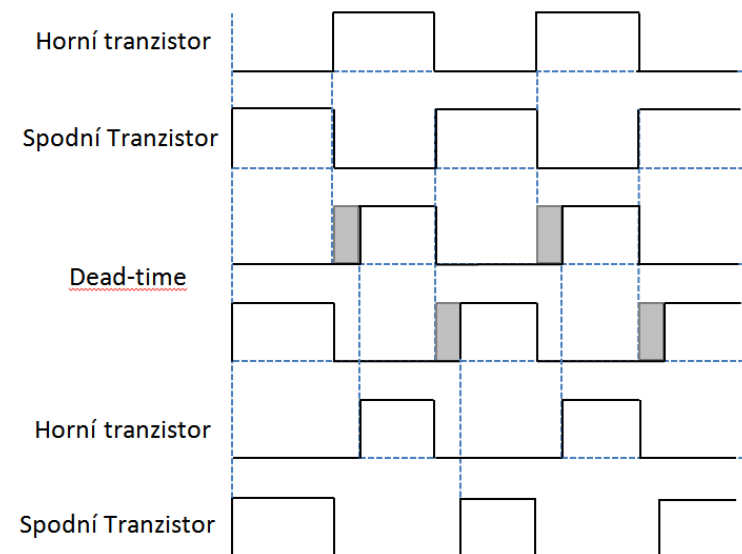
PWM modulaci řídí výstup z proudového regulátoru. Tento výstup nabývá hodnot 0–1. Tento výstup je vynásoben 1000 a je k němu přičtena a odečtena hodnota 1000, tím vznikne spodní a horní hranice. Tyto hranice jsou porovnávány s volně běžící pilou a pomocí jejich komparace je vytvořen PWM signál. Na dalším obrázku 9 můžeme vidět, jak je vytvořen PWM signál pro výstup 0,5. Jde tedy o 50% modulaci, kde spodní hranice bude 500 a horní hranice 1500.



Obrázek 9 - Vytvoření PWM signálu

3.1.3 Ošetření Dead-time

Protože u tranzistoru dochází k přechodovým jevům při jejich otevírání a zavírání, může vlivem pomalejšího zavírání dojít ke krátkodobému zkratu v nějaké větvi měniče. Je potřeba zajistit aby nedošlo v jedné větvi k tomu, že bude spodní i horní výkonový tranzistor otevřen ve stejný okamžik. Tomuto je potřeba zabránit a to pomocí zpoždění naběžné hrany spínacího signálu tranzistoru o určitý čas, tato doba se nazývá dead-time a měla by být větší než přechodový jev tranzistoru. Zpožděním řídicí hrany jednoho tranzistoru je vytvořen dostatečný čas pro zavření druhého tranzistoru, tím je zabráněno zkratu a tedy k zničení tranzistorů. Aplikací dead-timu můžeme vidět na následujícím obrázku 10.

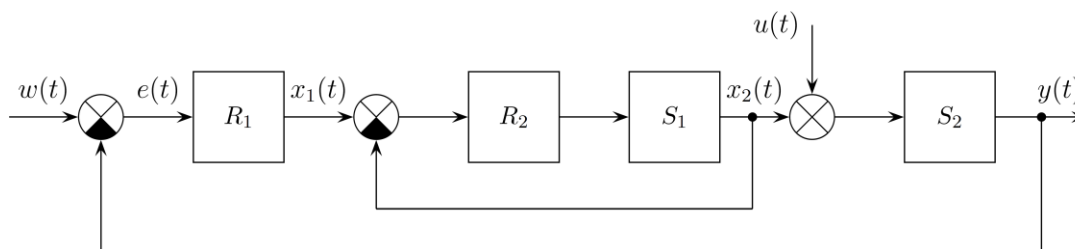


Obrázek 10 - Spínání horního a spodního tranzistoru s dead-time

3.1.4 Proudová a rychlostní regulace

Pro celou řadu regulačních pohonů se pro řízení používá podřazená proudová smyčka, která zlepšuje dynamické chování pohonu při omezení proudu. Regulátor obsahuje taky omezení, které zajistí, že proud nevzroste nad limitní hodnotu při odbuzení motoru nebo při zkratu. [7]

Na obrázku 9 níže můžeme vidět obecné schéma zapojení pro regulaci, kde R_1 označuje regulátor rychlosti a R_2 regulátor proudu. Za regulátory je zapojen motor. S_1 nám popisuje elektrickou část motoru a S_2 mechanickou. Na začátku vstupuje požadovaná veličina $w(t)$, která označuje požadovanou rychlost. Regulátor R_1 nám porovnává požadovanou hodnotu s výstupní rychlostí a určuje akční veličinu $x_1(t)$, která označuje požadovaný proud. Ten potom vstupuje do druhého PI regulátoru R_2 , který slouží k porovnávání požadovaného a výstupního proudu, podle kterých určuje akční veličinu $x_2(t)$. Ta slouží potom k určení střidy PWM signálu. Zbývající označení $u(t)$ je porucha a $y(t)$ je výstupní regulovaná veličina, v našem případě tedy rychlost.



Obrázek 11 - Schéma zapojení rychlostního a proudového regulátoru, [6]

3.2 Skalární řízení

Další možnost jak řídit motor je skalární řízení. Hlavní podstatou skalárního řízení je, že změnou frekvence lze měnit rychlost otáčení motoru. Hlavní podmínkou je udržení konstantního poměru mezi velikostí napájecího napětí a jeho frekvencí. Musí platit, že poměr napětí a jeho frekvence je konstantní tedy $\frac{u}{f} = \text{konstantní}$. Touto podmínkou se zajistí konstantní magnetický tok, který bude roven nominální hodnotě v širokém rozsahu otáček, což zajistí, že motor nebude přebuzován ani odbuzován. Tato podmínka ovšem neplatí pro nízký rozsah frekvencí (obecně platí od 10 Hz), zde je totiž nutno zajistit pomalejší pokles napájecího napětí.

Velkou výhodou skalárního řízení je, že nepotřebuje informaci o úhlové rychlosti motoru ani o natočení jeho rotoru. I přes to, že není požadována znalost úhlové rychlosti, může být určena nepřímo výpočtem z frekvence napájení, pokud zátěž nepřesáhne kritickou hodnotu.

Nevýhodou je závislost rychlosti na zátěži, horší dynamické vlastnosti a špatná regulace momentu.

Ke skalárnímu řízení se nejčastěji využívají dva způsoby řízení:

- **Řízení s konstantním statorovým tokem** – Jde o řízení s konstantním magnetickým tokem, tedy i statorový proud je konstantní. Ke změně rychlosti otáček se mění poměr frekvence a napájecího napětí. Jde o spojitou regulaci, kterou lze použít pro velký rozsah řízení otáček. Jde o řízení, kdy motor dosahuje konstantního momentu.

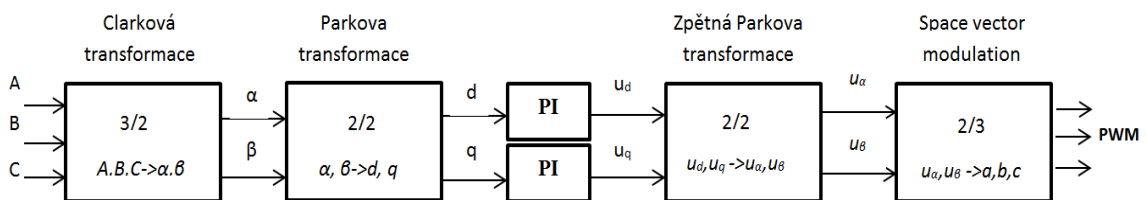
- **Řízení s konstantním napětím** – Při tomto řízení se u motoru dosahuje konstantního výkonu. Používá se v případě, kdy nastane možnost, že nebude možné zvyšovat napájecí napětí, které už dosahuje maximální úrovně. Tímto vlivem dochází u vyšších frekvencí k problému udržení konstantního poměru mezi frekvencí a napájecím napětím. Proto se pro řízení motoru udržuje konstantní napětí a mění se pouze frekvence, čímž se snižuje magnetický tok a tím dojde i ke snížení momentu motoru.

3.3 Vektorové řízení

Vektorové řízení je dobré pro dynamicky náročné aplikace. Vektorové řízení odstraňuje většinu nevýhod skalárního řízení a v porovnání s ním poskytuje mnohem větší výkon. Jde o zpětnovazební řízení, které umožňuje nezávisle řídit jak tok motorem, tak i jeho moment. Poskytuje přesné řízení a regulaci a to i v přechodových stavech jako je rozběh a brždění. Hlavní nevýhodou tohoto řízení je nezbytná znalost o aktuální poloze rotoru a velká náročnost na výpočty složitěho řídicího algoritmu.

Vektorové řízení nenastavuje pouze velikost magnetického toku a proudu ale také velikost jejich fázového posunu, kterým lze určovat velikost momentu. Největšího momentu lze dosáhnout, je-li proud kolmý na tok, tedy proud je před magnetickým tokem fázově posunutý o 90° .

Pro zjednodušení řízení motoru se přechází ze třífázové soustavy do dvoufázové. Tento převod se provádí pomocí Clarkové a Parkovy transformace. Využívá se toho, že ze třífázového souřadného systému A, B, C se pomocí Clarkové transformace převede do dvoufázové soustavy v souřadnicích α, β . Poté dojde k provedení Parkovy transformace, která převede dvoufázový systém se souřadnicemi α, β do dvoufázového systému se souřadnicemi d, q . V tomto systému jsou použity dva PI regulátory, které slouží k řízení. Akční veličiny z regulátoru je potřeba opět převést do třífázové soustavy. K tomu se využije zpětná Parkova transformace a následně SVM (Space Vector Modulation), po kterých dostaneme třífázový výstup s žádanými hodnotami napětí na jednotlivých fázích a, b, c , pomocí těchto výstupních hodnot se nastaví požadovaná velikost PWM. Grafické znázornění neboli schéma můžeme vidět na obrázku 12.



Obrázek 12 - Schéma zapojení pro transformaci a řízení

3.3.1 Clarkové transformace

Jak již bylo řečeno, Clarkové transformace provádí transformaci z třífázového souřadného systému a, b, c do dvoufázového systému α, β .

Definice obecné Clarkové transformace je:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = K * \begin{bmatrix} 1 & \cos(\gamma) & \cos(2\gamma) \\ 0 & \sin(\gamma) & \sin(2\gamma) \end{bmatrix} * \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (1)$$

Definice zpětné Clarkové transformace je:

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\gamma) & \sin(\gamma) \\ \cos(2\gamma) & \sin(2\gamma) \end{bmatrix} * \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (2)$$

Pokud zapojení vinutí motoru odpovídá hvězdě, musí platit, že součet všech proudů odpovídá 0, tedy:

$$i_a(t) + i_b(t) + i_c(t) = 0 \quad (3)$$

Platí-li předchozí podmínka, lze říci, že třetí veličina je lineární kombinací zbylých dvou, tedy v praxi lze měřit pouze dva proudy a třetí proud je dopočítáván.

Pro libovolnou veličinu lze zjednodušenou transformaci zapsat podle následujících vzorců 4 a 5.

$$x_\alpha = x_a \quad (4)$$

$$x_\beta = \frac{1}{\sqrt{3}} x_b - \frac{1}{\sqrt{3}} x_c \quad (5)$$

Zjednodušená zpětná transformace je popsána podle vzorců 6, 7 a 8.

$$x_a = x_\alpha \quad (6)$$

$$x_b = -\frac{1}{2} x_\alpha + \frac{\sqrt{3}}{2} x_\beta \quad (7)$$

$$x_c = -\frac{1}{2} x_\alpha - \frac{\sqrt{3}}{2} x_\beta \quad (8)$$

3.3.2 Parkova transformace

Parkova transformace provádí transformaci ze souřadného systému α, β do souřadného systému d, q , kde jedna souřadnice je kolmá k ose rotoru a druhá je s ní spjata. Rotorové a statorové vinutí mají vůči sobě stejnou polohu, z periodických

koeficientů se stanou konstantní. Úhel mezi statorovým a rotorovým vinutím se označuje θ . Obecný tvar transformace můžeme vidět v rovnici 9 a 10.

$$x_d = x_\alpha * \cos(\theta) + x_\beta * \sin(\theta) \quad (9)$$

$$x_q = x_\beta * \cos(\theta) - x_\alpha * \sin(\theta) \quad (10)$$

Obecný tvar zpětné transformace můžeme vidět v rovnici 11 a 12.

$$x_\alpha = x_d * \cos(\theta) - x_q * \sin(\theta) \quad (11)$$

$$x_\beta = x_q * \cos(\theta) + x_d * \sin(\theta) \quad (12)$$

3.4 Snímání polohy rotoru BLDC motoru

Motor, který měl pohánět robotické vozítko, neobsahoval žádné Hallovy snímače ani jiné snímače polohy. Jak již bylo řečeno, pro správné řízení BLDC motoru je potřeba znát jeho přesnou polohu. Proto bylo potřeba vybrat, jak se bude poloha rotoru určovat.

Při vybírání techniky pro detekci rotoru je potřeba vybírat na základě následujících faktorů, jako jsou výkonové požadavky, dostupné místo na motoru, cena a prostředí, kde bude motor použit.

Vysoko rozlišovací hřídelový enkodér nám nabízí v celém rozsahu rychlosti nejlepší výkon. Hlavní výhodou je, že nám enkodér dává absolutní pozici rotoru pro všechny rychlosti s dostatečně velkým rozlišením. Nevýhodou je, že použití tohoto typu enkodéru by znamenalo zvýšení celkové ceny projektu, proto se používá pouze, je-li k tomu opravdový důvod.

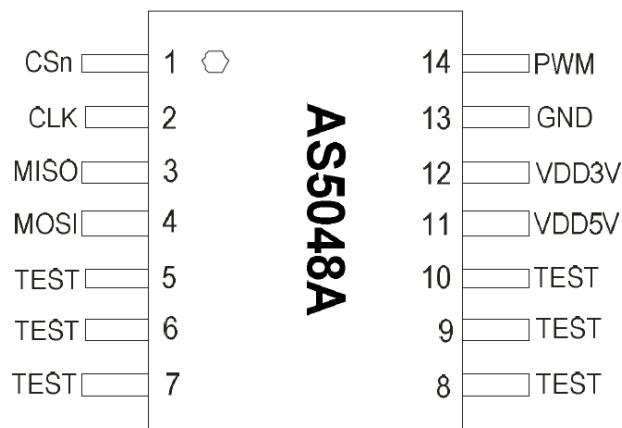
Nízko rozlišovací snímače polohy (například Hallovy sondy) poskytují mezi cenou a výkonem celkem přijatelný kompromis. Tyto snímače nám poskytují přesně komutační okamžiky pro střídač u všech rychlostí. Ovšem velkou nevýhodou je, že nejsme schopni přesně určovat rychlost pro nízké otáčky, kvůli velkému času mezi změnami stavů ze senzorů. Proto se většinou u použití tohoto typu snímání omezuje regulace rychlosti na limitní hodnotu pro nízké otáčky rotoru.

Použití předešlých zmiňovaných senzorů vyžaduje mechanické úpravy motoru k umístění snímače. Další nevýhodou je, že některé sondy mohou být teplotně závislé, proto se pak musí omezovat provoz motoru. Všeobecně použití senzorů zvyšuje celkovou cenu projektu, proto v posledních letech získává větší význam bezsenzorové řízení, kde se využívají metody založené na principu změny indukčnosti v závislosti na poloze rotoru.. [8]

Tabulka 5 - Souhrn typických vlastností pro jednotlivé detekce polohy rotoru, [8]

Typ pro detekci	Výkonnost rychlostí regulace	Spouštěcí výkon	Cena
Vysoko rozlišovací hřídelový enkodér	Velká rozlišovací schopnost v celém rozsahu z klidové do jmenovité rychlosti	Není nutný zarovnaný stav. Zaručená absence zpětné rotace při spuštění. Klidný start	Vysoká Vyžaduje přídavné vodiče a místo
Nízko rozlišovací senzor	Dobrá výkonost Uzavřená smyčka od 5% jmenovité hodnoty rychlosti. Provoz uzavřené smyčky není možný pod touto úrovní.	Není nutný zarovnaný stav. Zaručená absence zpětné rotace při spuštění. Klidný start	Střední Vyžaduje vodiče pro připojení senzorů
Bezsenzorová technika	Dobrá výkonost Uzavřená smyčka od 5% jmenovité hodnoty rychlosti. Provoz uzavřené smyčky není možný pod touto úrovní.	Při spuštění vyžaduje zarovnaný stav. Nemůže garantovat absenci zpětné rotace při spuštění. Zajistit klidný start je obtížné	Nízká

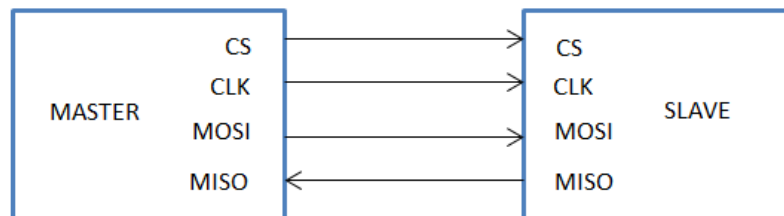
Pro detekci rotoru byl vybrán magnetický absolutní snímač polohy. Snímač je od firmy AMS (austria micro systems) a jeho průmyslové označení je AS5048. Jde o 14-bitový snímač k měření úhlu natočení v rozmezí 0–360°. Snímána je poloha magnetu, který je připevněn na rotor. Velkou výhodou bylo hlavně to, že nám byly snímače nabídnuty jako zkušební vzorek, proto byla jejich cena nulová. Napájecí úroveň snímače je 3–5,5 V. Zapojení jednotlivých pinů je zobrazeno v následujícím obrázku 13.



Obrázek 13 - AS5048, [9]

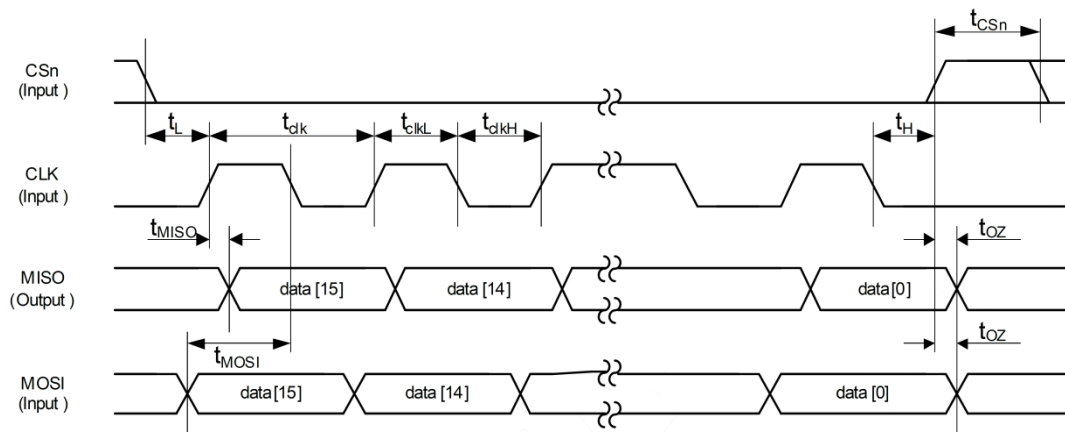
3.4.1 Komunikace AS5048

Snímač komunikuje pomocí SPI (Serial Peripheral Interface) neboli sériového periferního rozhraní. Toto rozhraní se často používá pro komunikaci mezi řídicími mikroprocesory a ostatními obvody. Obecné zapojení SPI komunikace můžeme vidět na obrázku 14.



Obrázek 14 - Obecné zapojení komunikace SPI

Komunikace se spouští nastavením CS (chip selectu) na hodnotu LOW tedy na logickou nulu. Potom je potřeba generovat CLK (clock hodiny), při kterých dochází k čtení informace o poloze. První je poslán MSB neboli nejvíce významný bit. Na MOSI (master output slave input) je potřeba poslat příkaz pro čtení úhlu pomocí 0xFFFF, tedy po celou dobu komunikace musí být MOSI nastaven na HIGH úroveň. Informaci čteme z pinu MISO (master input slave output).



Obrázek 15 - Časový diagram SPI snímače AS5048, [9]

Pro správnou komunikaci je potřeba dodržet časové odstupy, které jsou pojmenovány a zobrazeny v předchozím obrázku 15. Minimální a maximální velikost těchto hodnot můžeme vidět v tabulce 6.

Tabulka 6 - časové parametry SPI

Parametr	Popis	Min	Max	Jednotky
t_L	Čas mezi sestupnou hranou CS a náběžnou hranou CLK	10		ns
T_{CLK}	Sériový periodický hodinový signál	100		ns
t_{CLKL}	Délka clocku LOW	50		ns
t_{CLKH}	Délka clocku HIGH	50		ns
t_H	Čas mezi sestupnou hranou CLK a naběžnou hranou CS	$T_{CLK}/2$		ns
T_{CSnH}	Délka CS na úrovni HIGH	10		ns
t_{MOSI}	Délka platnosti vstupních dat	20		ns
t_{MISO}	Délka platností výstupních dat		20	ns

Po proběhnutí komunikace přijmeme data, které jsou tvořena 16 bity. Obsah jednotlivých bitů, které přijmeme, je znázorněn v tabulce 7.

Tabulka 7 - Popis jednotlivých bitů z přijatých dat

Přijaté data										
Bit	MSB	14	13	12	...	3	2	1	LSB	
	PAR	RWn	Adresa <13:0>							
Definice a popis bitů										
PAR		Paritní bit								
RWn		Bit indikující čtení (1) nebo zápis (0)								
Adresa		14 bitové data								

Z obrázku je vidět, že první dva MSB bity jsou bity kontrolní. První bit je paritní, tedy značí, jestli je počet jedniček sudý. Druhý bit RWn nám indikuje, jestli provádíme zápis nebo čtení. Ostatní bity na adrese 13–0 nás informují o aktuální poloze rotoru. Po převedení z binární soustavy do desítkové dostaneme číslo v rozmezí 0-16384. Toto číslo je pro získání přesného úhlu natočení rotoru potřeba upravit podle vzorce 13.

$$\varphi = A * \frac{360}{16384} \quad (13)$$

Kde: φ (°) - úhel natočení rotoru

A (-) - odnota přečtená ze snímače

3.4.2 Výběr Magnetu

AS5048 umí pracovat s mnoha různými magnety. Magnet by měl ovšem mít minimální velikost, kterou udává výrobce, v našem případě 6-8mm v průměru a 2,5mm v tloušťce. Magnet musí být umístěn kolmo k povrchu snímače. Jako materiál magnetu byla zvolena směs neodymu, železa a boru. Tento typ nabízí nejlepší poměr ve srovnání výkonu a ceny. Má velmi dobré magnetické vlastnosti a jeho průmyslové označení u firmy AMS je AS5000-MD6H-1.

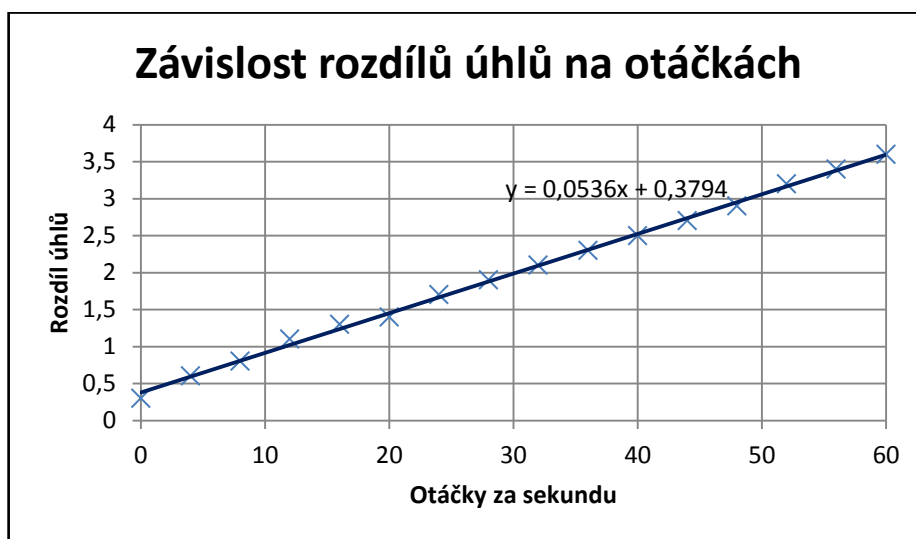
3.4.3 Měření AS5048

Při vyšších otáčkách rotoru dochází vlivem analogové části snímače ke zpoždění přesné aktuální polohy, proto bylo potřeba snímač proměřit a skutečnou polohu následně dopočítávat. Měření bylo provedeno pomocí motoru

s inkrementálním enkodérem, který měřil úhel natočení s přesností 2^{12} tedy 4096 pulzů na jednu otáčku. V LabView byl vytvořen program pro zobrazování rozdílů úhlů a otáček za sekundu z enkodéru. Motor byl roztočen na různé hodnoty rychlosti otáčení, při kterých byl měřen rozdíl z enkodéru a snímače.

Tabulka 8 - Závislost rozdílů úhlů na otáčkách

Otáčky za sekundu	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
Rozdíl úhlů	0,3	0,6	0,8	1,1	1,3	1,4	1,7	1,9	2,1	2,3	2,5	2,7	2,9	3,2	3,4	3,6



Obrázek 16 - Graf závislosti rozdílů úhlů na otáčkách za sekundu

Jak můžeme vidět v grafu na obrázku 16, jde o lineární závislost na otáčkách motoru. V počátku při rychlosti 0 ovšem můžeme vidět rozdíl 0,3 to je způsobeno tím, že se nepodařilo přesně vycentrovat magnet na střed snímače. Tato chyba by ovšem neměla mít vliv na pozdější dopočítávání polohy.

Pro zjištění přesné aktuální polohy je tedy potřeba dopočítat úhel pomocí rovnice 14:

$$\varphi_R = \varphi_S \pm 0,0536 * RPS \quad (14)$$

Kde: φ_R (°) – Skutečná poloha rotoru

φ_S (°) – úhel ze snímače

RPS (-) – velikost otáček za sekundu

Plus nebo minus v rovnici je závislé na směru otáčení, kde při otáčení doprava musíme k úhlu přičítat a při směru otáčení doleva odčítat.

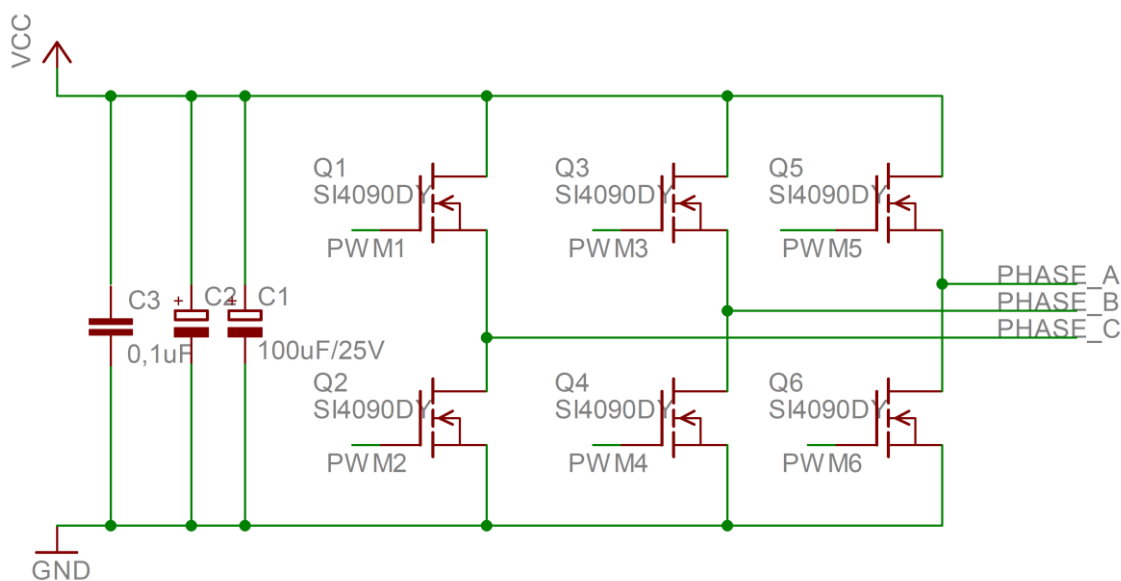
4 NÁVRH HARDWARU

4.1 Výkonová část

Hlavními požadavky jsou, aby hardware pracoval v rozsahu vstupního napětí VCC 6–15V s maximálním protékajícím proudem 15 A. Jako první je potřeba navrhnout třífázový střídač neboli měnič, pracující v dané oblasti. K němu poté navrhnout odpovídající budiče pro MOSFET tranzistory a funkční měření proudu. Pro měření proudu je potřeba napájecí napětí 3,3V, které je nutno stabilizovat ze vstupního napájecího napětí VCC.

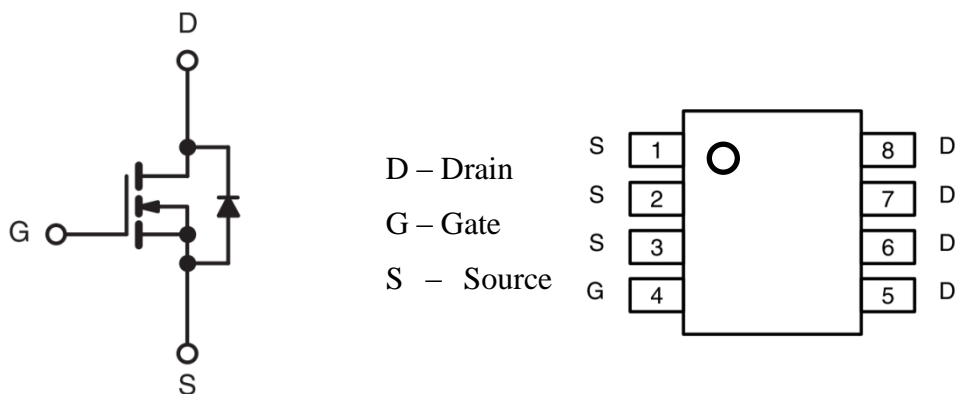
4.1.1 Návrh měniče

Jako první byl navržen třífázový střídač neboli měnič. Jde vlastně o hlavní obvod, který zajišťuje řízení elektrické energie ze zdroje do BLDC motoru. Měnič, jak už bylo řečeno v teoretické části, se skládá ze šesti tranzistorů, po dvou v každé větvi a mění vstupní jednosměrné napájecí napětí na výstupní trojfázové střídavé napětí. Na vstupní napětí je potřeba zvolit velký kondenzátor, který zajistí minimální zvlnění elektrického napětí, a kondenzátor s nízkou hodnotou k filtrování šumu. Jak můžeme vidět na obrázku 17 níže, pro filtraci byl použit kondenzátor C3 o velikosti 0,1 μ F a proti zvlnění napětí dva elektrolytické kondenzátory C1 a C2 s velikostí 100 μ F.



Obrázek 17 - Navržený třífázový střídač

Jako typ tranzistoru byl zvolen MOSFET tranzistor typu N, který při nízkých hodnotách závěrného napětí dosahuje malých hodnot odporu v sepnutém stavě, čímž se dosahuje menších ztrát a vyšší účinnosti. Jeho schématickou značku můžeme vidět na obrázku 18.

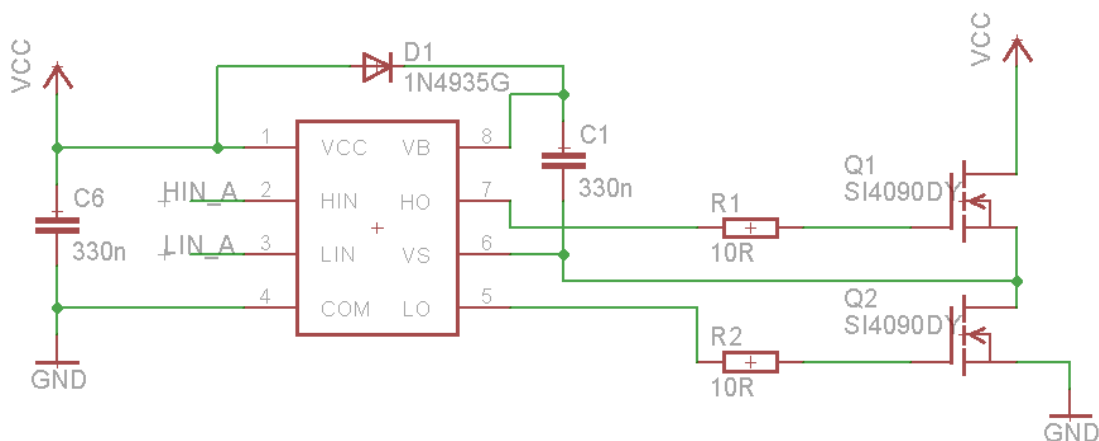


Obrázek 18 - Schématická značka a rozmístění pinů SI4090DY,

Zvolen byl tranzistor s označením SI4090DY, který je schopen spínat do 19,7A se vstupním napětím až 100V a odporem v sepnutém stavě 0,012Ω.

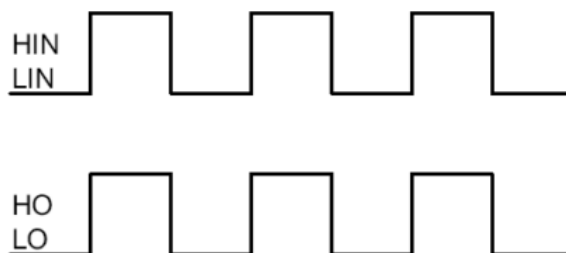
4.1.2 Návrh budiče pro tranzistory

Pro správné otvírání MOSFET tranzistorů je potřeba použít budič. Spínání spodních tranzistorů, které mají připojen Source na zem, není problém. Problém ovšem nastává pro spínání horních tranzistoru, kde potřebujeme pro otevření tranzistoru rozdíl mezi napětí na GATE a na SOURCE minimálně 2 V. A nýbrž pin SOURCE horního tranzistoru je připojen na fázi motoru tedy na zátěž a na pin DRAIN spodního tranzistoru, potřebujeme napájecí napětí vyšší, než je napájení můstku. To získáme použitím budiče s nábojovou pumpou. Jako budič byl vybrán obvod s označením IRS2301S, který pracuje s napájecím napětím 5–20 V. Jde o budič, který spíná jeden horní a jeden spodní tranzistor. Jeho zapojení můžeme vidět na obrázku 19.



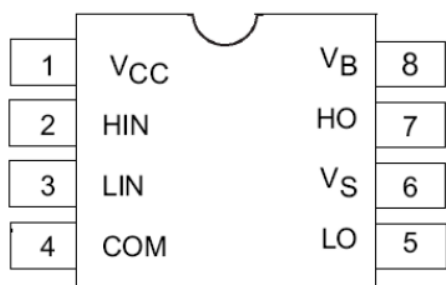
Obrázek 19 - Zapojení budiče IRS2301S

Budič IRS2301S na rozdíl od jemu podobných budičů jako IRS2302 nebo IRS2304 nemá žádný vstupní ani výstupní pin negován ani zpožděn, tedy výstupní signál odpovídá vstupnímu signálu, který je v našem případě 3,3V. Také nijak nešetří DEAT-TIME, který je tedy potřeba ošetřit softwarově. Závislost výstupních signálů na vstupních můžeme vidět na následujícím obrázku.



Obrázek 20 - Vstupní a výstupní signály budiče IRS2301S

Rozmístění jednotlivých pinů budiče IRS2301S můžeme vidět na následujícím obrázku.



Obrázek 21 - Rozmístění pinů IRS2301S

VCC – Vstupní napájecí napětí

HIN – Vstupní řídicí signál pro horní tranzistor

LIN – Vstupní řídicí signál pro dolní tranzistor

COM – Zem

VB – Boostovací napětí

HO – Výstupní signál řídicí spínání horního tranzistoru

LO – Výstupní signál řídicí spínání horního tranzistoru

V zapojení budiče je potřeba zvolit správnou velikost kapacity C1 a C6. Tato minimální velikost se určí s následujícího vzorce číslo 15.

$$C \geq \frac{2Q_{BS}}{U_{cc} - U_f - U_{LS} - U_{min}} \quad (15)$$

Kde: U_{cc} – je vstupní napájecí napětí

U_f – úbytek napětí na diodě, přes kterou je kondenzátor nabíjen

U_{LS} – úbytek napětí na dolním tranzistoru v sepnutém stavě

U_{min} – minimální napětí pro úplné otevření tranzistoru

Q_{BS} – náboj kondenzátoru pro který platí:

$$Q_{BS} = 2Q_g + \frac{I_{GBS(max)}}{f} + Q_{LS} + \frac{I_{CBS(leak)}}{f} \quad (16)$$

Kde: Q_g – je náboj tranzistoru na GATE

I_{GBS} – je maximální závěrný proud

Q_{LS} – je náboj požadovaný koncovým členem vnitřního budícího obvodu

I_{CBS} – je svodový proud kondenzátoru

f – je frekvence spínání

Po dosazení hodnot do vzorce číslo 16 dostaneme požadovaný výsledný náboj kondenzátoru:

$$Q_{BS} = 2 * 50 * 10^9 + \frac{260 * 10^{-6}}{20 * 10^3} + 8,5 * 10^{-9} + 0 = 125 * 10^{-9} C$$

Po dosazení této hodnoty do vzorce 15 získáme minimální požadovanou velikost kapacity kondenzátoru:

$$C \geq \frac{2 * 125 * 10^{-9}}{10 - 1,5 - 0,12 - 2} = 34,2 nF$$

Toto je minimální velikost kapacity, ovšem v praxi se tato hodnota volí 5x – 15x větší. Proto byla výsledná hodnota kapacity u kondenzátoru zvolena 330nF.

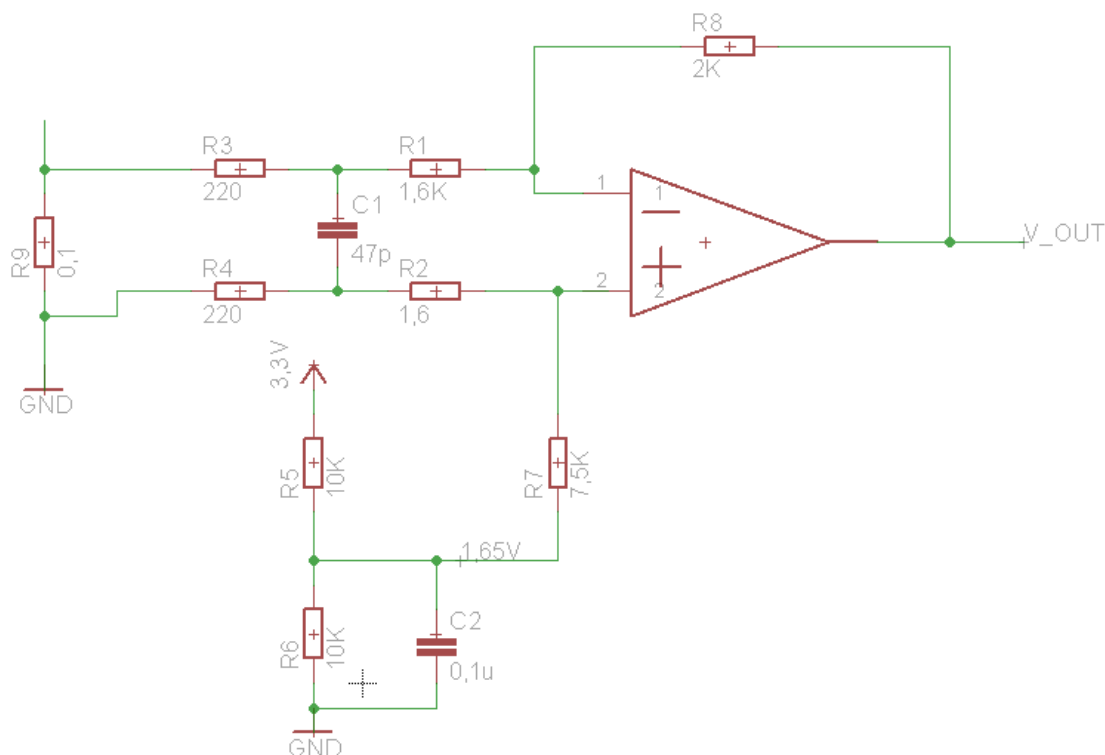
4.1.3 Měření proudu

Pro správnou regulaci a řízení proudu obvodem je potřeba daný proud v obvodu měřit. V úvahu připadali dvě možnosti jak proud měřit.

První možnost byla použít snímací odpor jako bočník a měřit na něm úbytek napětí. Tato možnost se používá hlavně z důvodů nízké ceny. Snímací odpor je zapojen mezi zemí a spodním tranzistorem. Pro měření proudu je proto nezbytné, aby byl spodní tranzistor sepnutý, čímž vzniká hlavní požadavek na řízení a tím je synchronizace měření proudu s PWM. Měřený signál je potřeba ještě upravit, aby výstupní hodnota byla ve vhodném rozsahu pro DSC (Digital Signal Controller), což je 0–3,3 V. Pokud použijeme snímací odpor o velikosti 0,1 Ω, tak při plném proudu 15 A bude tento úbytek 1,5 V. Tento signál se upraví pomocí operačního zesilovače, který je zapojen jako diferenční zesilovač, zesiluje tedy rozdíl mezi invertujícím a neinvertujícím vstupem. Na neinvertující vstup je potřeba přivést referenční napětí 1,65 V, které bude odpovídat proudu 0 A. Pomocí odporů je potřeba nastavit zesílení operačního zesilovače, které je $(3,3-1,65)/1,5$ tedy v našem případě nutné zesílení 1,1. Zesílení se nastavuje pomocí odporů R1, R3 a R8 podle vzorce:

$$A = \frac{R8}{R1 + R3} \quad (17)$$

Výsledné zesílení získáme tedy použitím odporů $R8 = 2 \text{ K}\Omega$, $R3 = 220 \text{ }\Omega$ a $R1 = 1,6 \text{ K}\Omega$. Při tomhle zapojení dostaneme na výstupu operačního zesilovače napětí V_{OUT} , v rozsahu 0–3,3 V, kde 0 V odpovídá maximálnímu zápornému proudu -15 A, hodnota 1,65 V odpovídá nulovému proudu 0 A a 3,3 V značí maximální kladný proud 15 A. Zapojení operačního zesilovače můžeme vidět na následujícím obrázku 22. Měřicí odpor je na obrázku označen jako R9 a jak již bylo řečeno, pro každou fázi je zapojen mezi spodní tranzistor a zem.

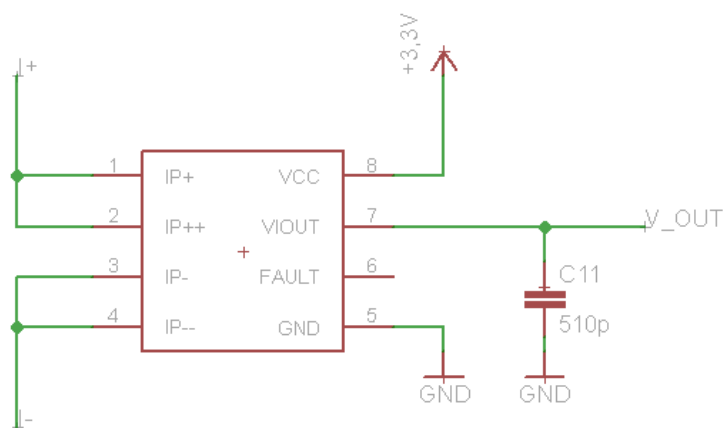


Obrázek 22 - Zapojení operačního zesilovače pro měření proudu

Referenční napětí 1,65 V je získáno z napětí 3,3 V pomocí odporového děliče, kde jsou využity dva odpory o velikosti 10 K Ω . Tento odporový dělič je zobrazen v předchozím obrázku 22 v dolní části.

V našem projektu byla jako finální verze použita pro měření proudu druhá možnost, která je sice nákladnější, ovšem z hlediska množství použitých součástek a tím ušetření místa na DPS (desce plošných spojů) mnohem jednodušší. Jde o použití snímače určeného přímo pro měření proudu od firmy Allegro Microsystems s průmyslovým označením ACS711. Jde o snímač, který je schopen měřit proudy v rozsahu ± 25 A. Snímač obsahuje lineární obvod, který pracuje na Hallově efektu pro měření proudu. Tedy další výhodou je, že ho můžeme zapojit přímo do obvodu s motorem na rozdíl od předchozího zapojení, kde měření muselo být zapojeno mezi spodní tranzistor a zem. Tímto zapojením tedy odpadá nutnost synchronizovat měření s PWM, takže pro měření nemusí být sepnut spodní tranzistor měniče. Zapojením přímo do fáze můžeme proud měřit v každém okamžiku. Snímač jediné co požaduje, je napájení 3–5 V. Výstup snímače je napětí v rozsahu 0–3,3 V, tedy v rozsahu vyhovující přímo mikroprocesoru,

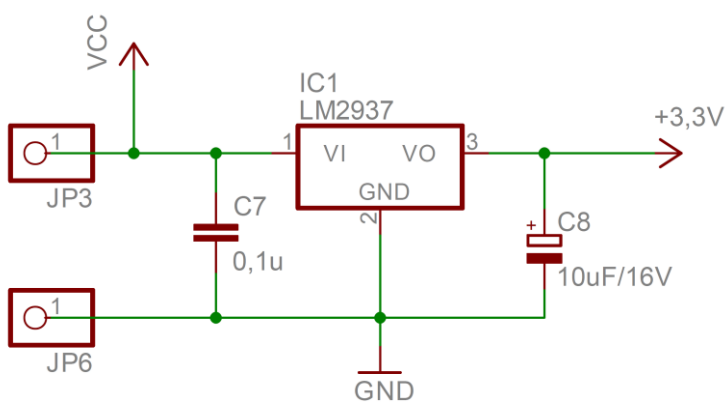
proto není nutné použít operační zesilovač ani nijak tento signál dále upravovat. Velikost výstupního signálu snímače 0 V odpovídá -25 A, napětí 1,65 V odpovídá nulovému proudu 0 A, a 3,3 V odpovídá maximálnímu proudu 25 A. Zapojení proudového snímače můžeme vidět na obrázku 23.



Obrázek 23 - Zapojení ACS711 pro měření proudu

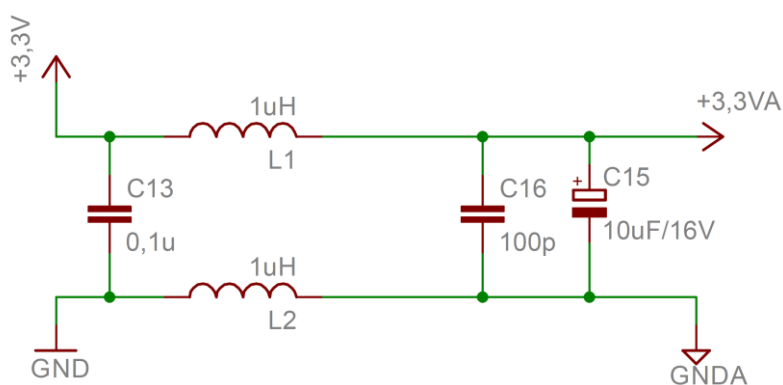
4.1.4 Stabilizace napětí

Celý obvod bude napájen ze dvou nebo tří akumulátorových článků, tedy přibližně vstupní napětí 6–12 V. Pro magnetický senzor polohy, pro mikroprocesor a pro snímač měření proudu je potřeba tohle napětí stabilizovat na 3,3 V. K tomu je využit stabilizátor s označením LM2937-3,3 se vstupním napětím v rozsahu 4,75–26 V. Jde o stabilizátor na 3,3 V, který je schopný dodávat až 500 mA. Na vstup stabilizátoru byl připojen filtrační kondenzátor 0,1 μF proti šumu a na výstup elektrolytický kondenzátor 10 μF proti kolísání výstupního stabilizovaného napětí.



Obrázek 24 - Zapojení stabilizátoru

Pro mikroprocesor je potřeba přivést ještě referenční analogové napětí s analogovou zemí. To získáme pomocí dvou cívek o velikost 1uH, které jsou zapojeny sériově do obvodu, jak na kladný potenciál, tak na zem. Toto zapojení můžeme vidět na obrázku 25 níže. Výstupní analogové napětí je taky potřeba filtrovat proti šumu a kolísání, k tomu byla zvolena dvojice paralelně připojených kondenzátorů o velikosti 100 pF a 10 μ F.



Obrázek 25 - Oddělení analogového napětí 3,3V pro procesor

4.2 Řídicí část - Mikroprocesor 56F82743

Jako hlavní řídicí část obvodu byl vybrán mikroprocesor od firmy Freescale s průmyslovým označením 56F82743VLC, který slouží přímo pro řízení motorů.

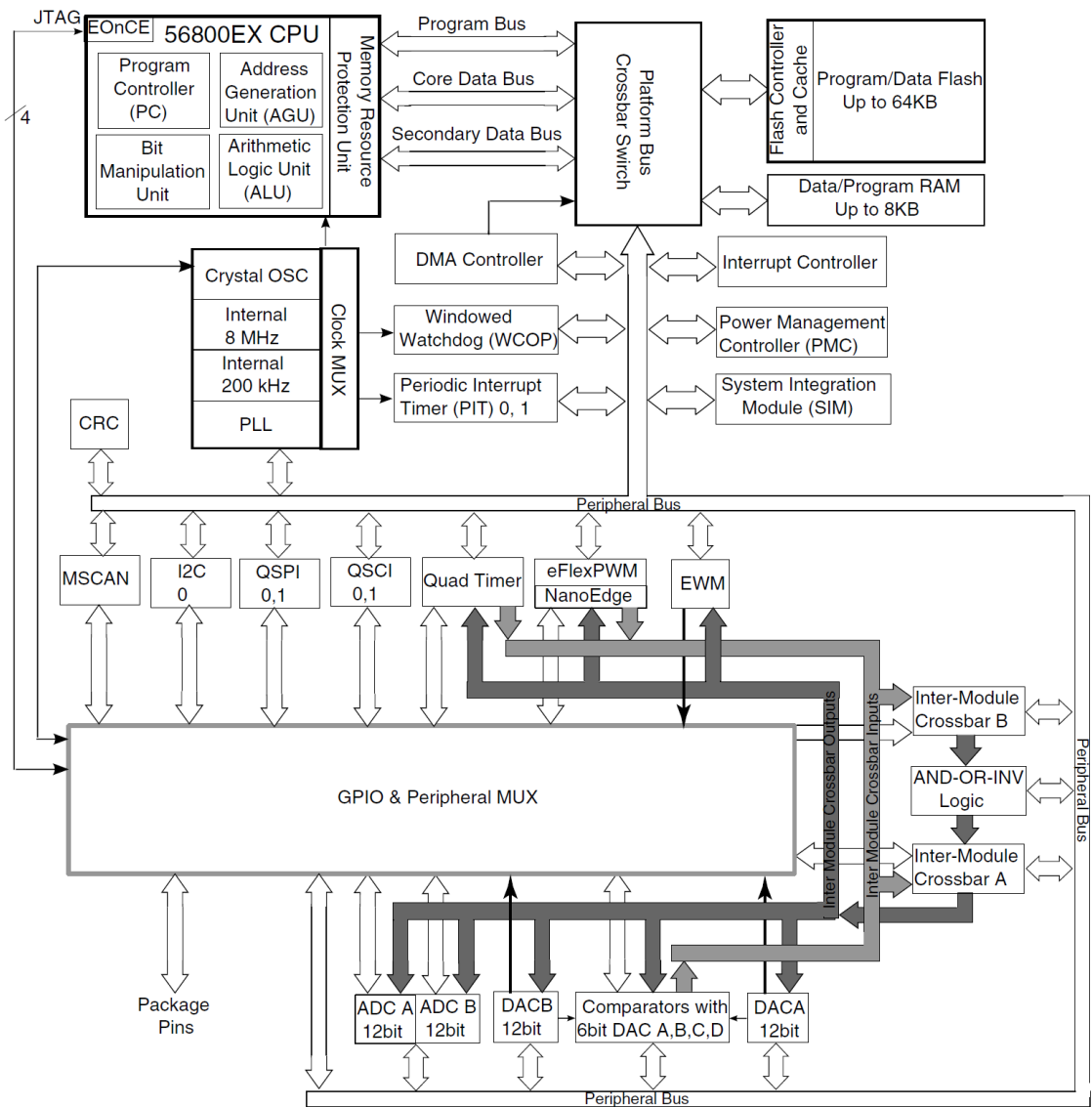
4.2.1 Základní údaje

Mikroprocesor 56F85743 patří do rodiny DSC (Digital Signal Controllers) založených na 32-bitovém jádru DSP56800EX, které kombinuje výpočetní výkon DSC procesoru a výhody klasického mikrokontroléru.

4.2.1.1 Vlastnosti:

- Frekvence mikroprocesoru až 100 MHz, což odpovídá výpočetnímu výkonu až 100 MIPS
- 64 KB flash paměti
- 8 KB RAM paměti – Data i program lze mapovat do obou pamětí flash i RAM
- Dva vysokorychlostní tříkanálové 12-bitové ADC (analog-to-digital converter) převodníky s dynamickým zesilovačem
- Dva 12-bitové DAC (digital-to-analog converter) a tři analogové komparátory
- PWM modul s 6 PWM výstupy
- Komunikační rozhraní SCI (Serial Communication Interface), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit)
- Jeden 16-bit programovací časovač a dva periodické intervalové časovače
- Dále obsahuje externí Watchdog monitor (EWM)
- CRC- Cyklický redundantní kontrolér
- Relaxační oscilátor integrovaný na čipu
- DMA kontrolér
- JTAG/Enhanced On-Chip Emulation
- Napájecí napětí 3–3,6 V
- Až 26 GPIO

4.2.1.2 Blokové schéma mikročítače MC56F82743



Obrázek 26 - Blokové schéma mikročítače MC56F82743, [16]

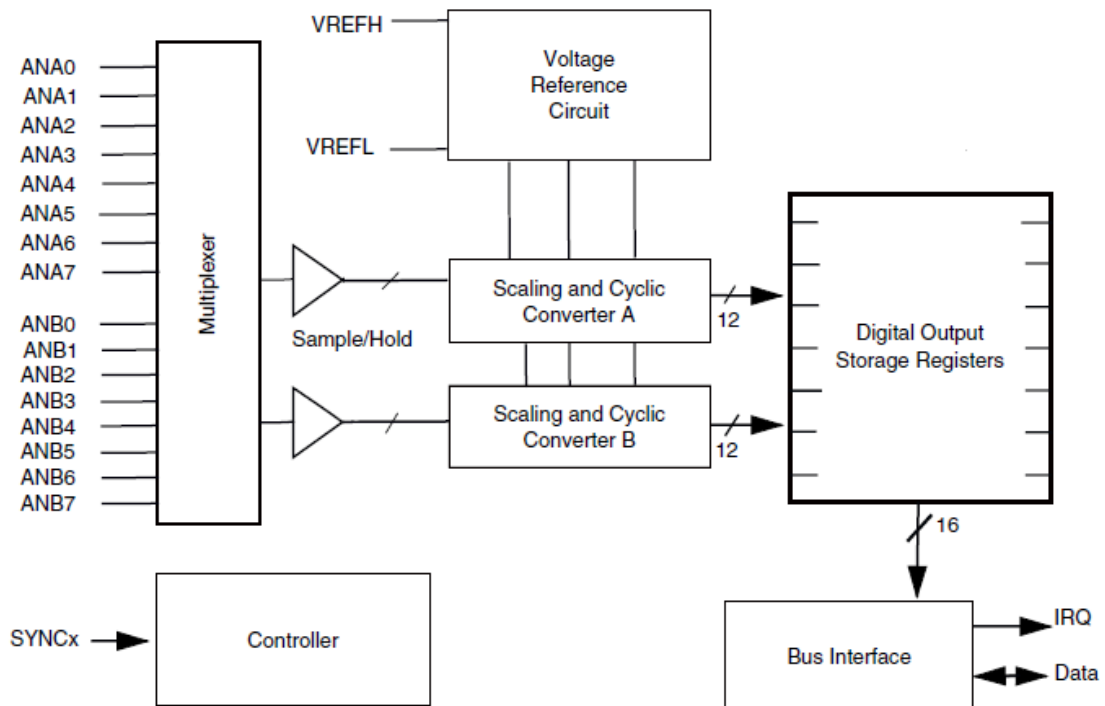
4.2.2 ADC – Analog-to-Digital Converter

Analogově digitální převodník se skládá ze dvou nezávislých převodníků, z nichž každý obsahuje samostatný sample a hold obvod. Převodníky v mikroprocesoru dokáží zpracovat až 6 externích analogových vstupů (2x3). Analogové vstupy jsou díky vstupnímu multiplexeru jednoduše konfigurovatelné.

4.2.2.1 Vlastnosti ADC převodníků:

- 12-bitů rozlišení
- Až 2,5 milionů vzorků za sekundu
- Můžou být synchronizovány s dalšími perifériemi, které jsou připojeny na interní crossbar modul, jako PWM a jiné
- Maximální měřicí frekvence 10 MHz
- zesílení vstupních signálů x1, x2 a x4
- Postupně je měřeno a zapisováno všech 6 měření
- Znaménkový a bezznaménkový výstup
- Možnost korekce výstupu odečtením předem naprogramované hodnoty

4.2.2.2 Blokový diagram ADC modulu



Obrázek 27 - Blokový diagram ADC modulu, [16]

4.2.3 PIT – Programmable Interval Timer

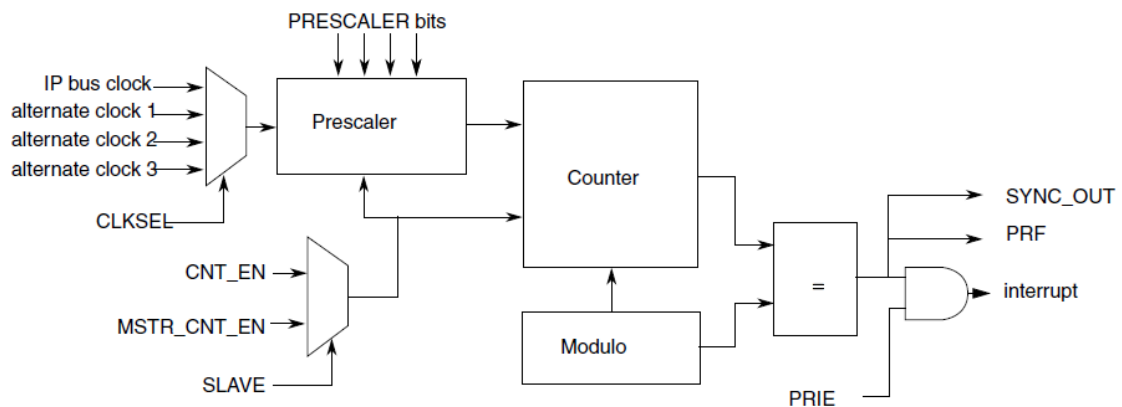
Programovací časový modul neboli PIT slouží pro jednoduché časové aplikace. PIT timer se skládá z 16-bitového čítače, modulo registrů a řídicího registru. Modulo a řídicí registry slouží pro zápis i pro čtení na rozdíl od čítacího registru, který slouží pouze pro zápis.

Modulo registr je naplněn číslem, které slouží jako horní hranice, do které čítač čítá. Pokud čítač dopočítá do dané hodnoty, resetuje se na 0x0000 a začne nové čítání. Při dopočítání se taky nastaví flag, který může generovat přerušení, které ovšem musí být povoleno.

4.2.3.1 Vlastnosti:

- 16-bitový časovač/čítač
- Programovatelné čítání modulo
- Až 4 možnosti volitelných hodinových zdrojů
- Slave mode, který umožňuje synchronizovat povolení několika PIT modulů

4.2.3.2 Blokový diagram PIT modulu



Obrázek 28 - Blokový digram PIT modulu, [16]

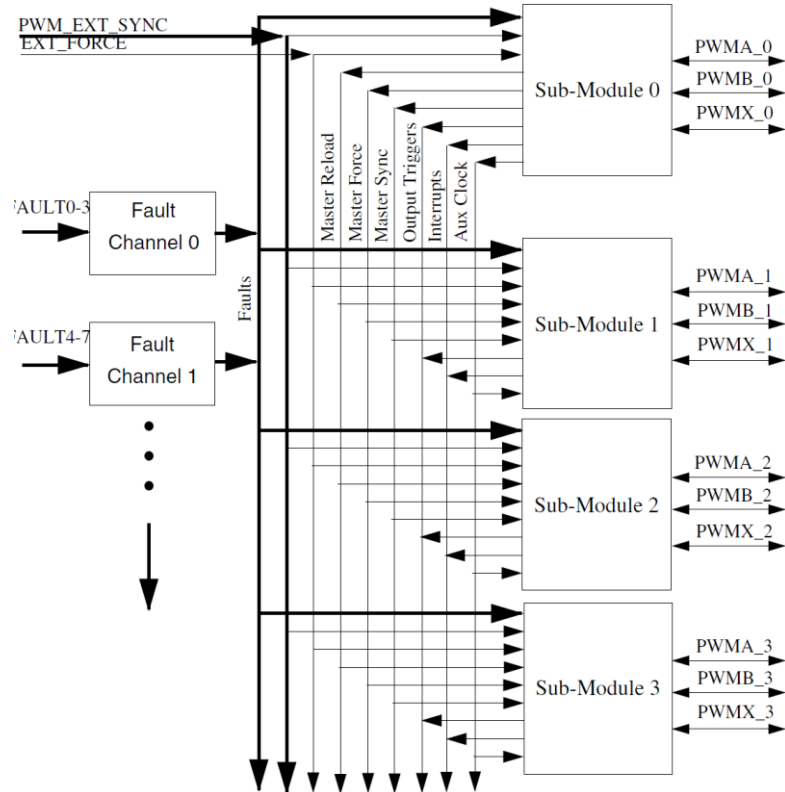
4.2.4 PWM

PWM modul obsahuje PWM submodely, kde každý z nich je nastaven na řízení jedné poloviny mostu výkonového stupně. PWM modul umožňuje generovat buďto tři komplementární výstupy, šest nezávislých výstupů nebo jejich různé kombinace. PWM je centrována na střed ale může být zarovnána i na okraj volně běžící pily, kde oba módy dovolují řídit šířku pulzu od 0 % do 100 %.

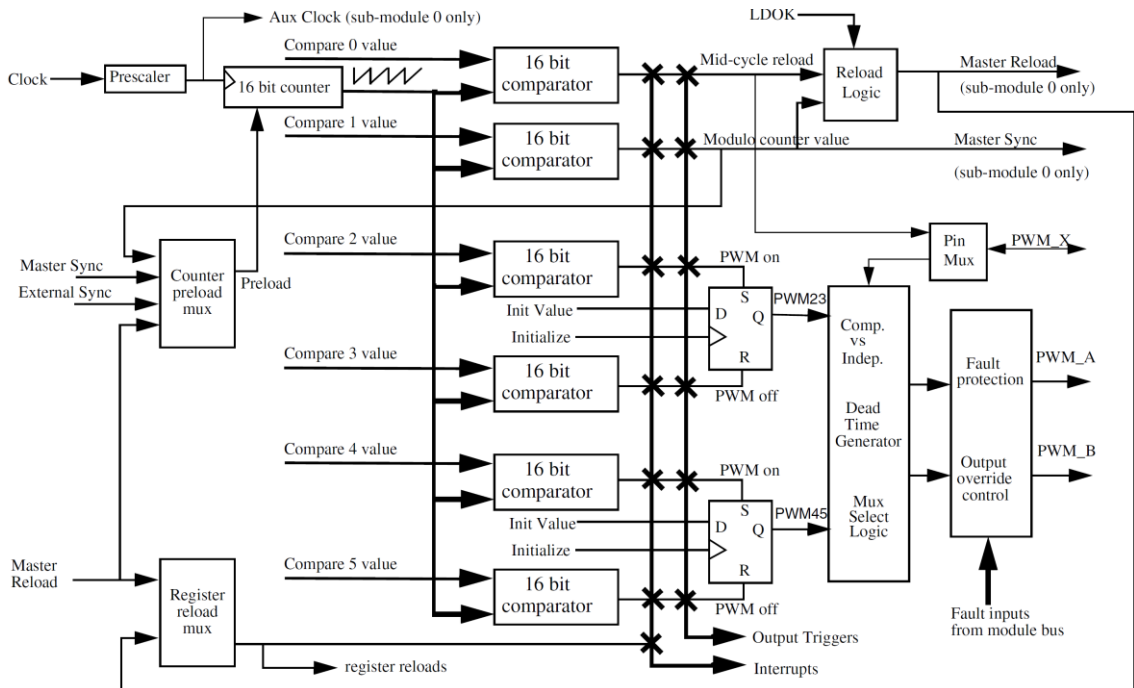
4.2.4.1 Vlastnosti:

- 16-bitové rozlišení
- Výstupy mohou pracovat jako komplementární páry nebo jako jednotlivé nezávislé výstupy
- Podpora pro synchronizaci s externím zařízením nebo s jinou PWM
- Dvojitá vyrovnávací paměť PWM registru
- Integrovaný reload rate od 1 do 16
- Reload schopnost uprostřed PWM cyklu (half-cycle reload)
- Podpora pro dvojitě spínání výstupu PWM
- Programovatelné filtry pro poruchové vstupy
- Generování deadtime nezávisle pro nástupnou a sestupnou hranu
- Individuální ovládání každého PWM výstupu pomocí softwaru
- Každá komplementární dvojice může pracovat s vlastní PWM frekvencí a deadtime hodnotou
- Lze použít asymetrický PWM výstup
- Programovatelná fault ochrana

4.2.4.2 Blokový digram PWM modulu



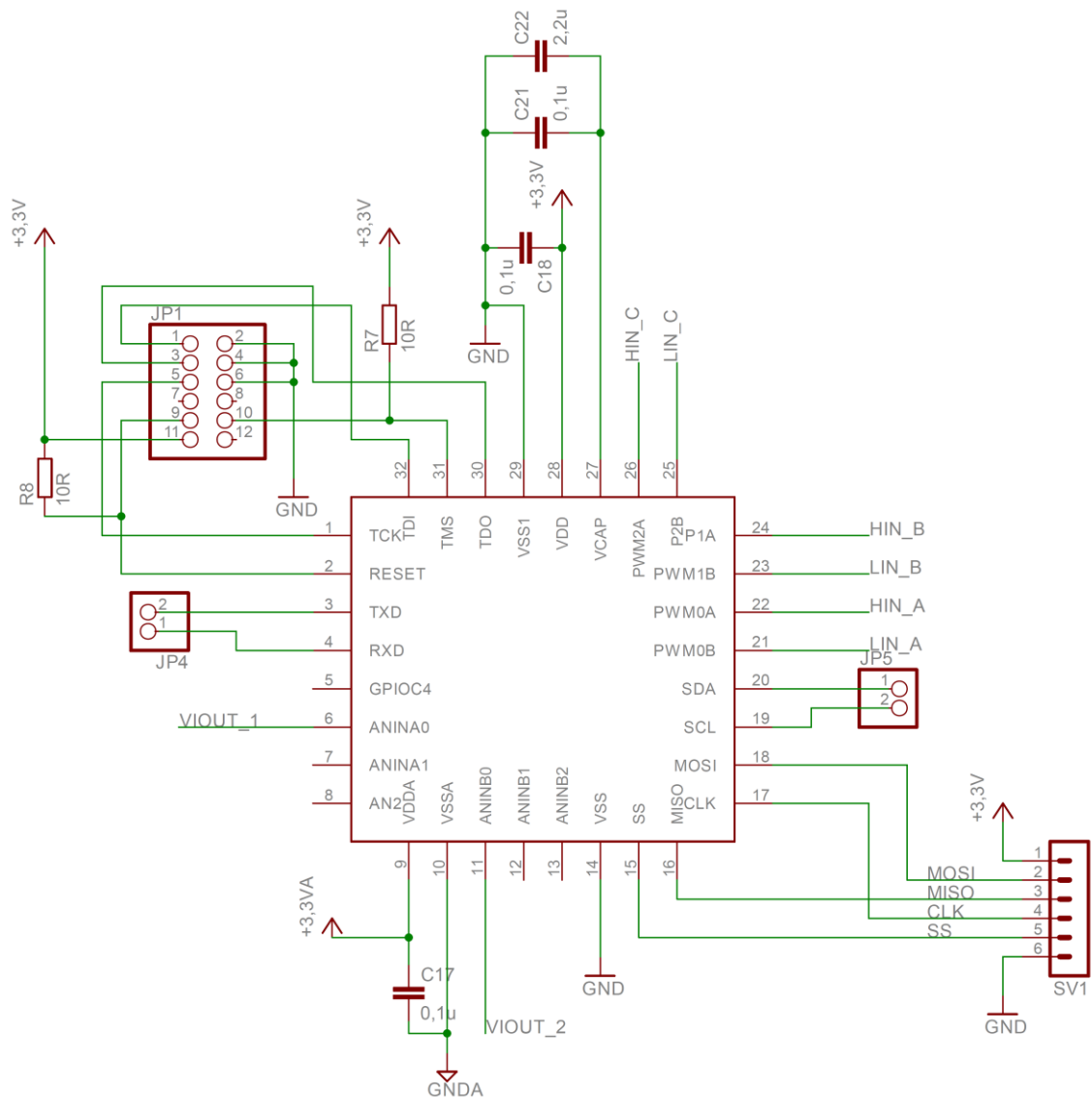
Obrázek 29 - Blokový diagram PWM modulu, [16]



Obrázek 30 - Blokový diagram PWM submodule, [16]

4.2.5 Zapojení mikroprocesoru MC56F82743

Použitý mikroprocesor má package s označením 32-pin LQFP. Jde tedy o mikroprocesor s 32 piny a je programován přes JTAG, u kterého je potřeba zapojit piny TMS a RESET přes 10 ohmový rezistor na napájení 3,3 V. Zapojení jednotlivých programovacích pinů můžeme vidět na obrázku 31, kde JTAG je označen jako JP1. K napájení mikroprocesoru byl přidán filtrovací kondenzátor o velikosti 0,1 μF co nejlíže k napájecím pinům. K pinu 27 na obrázku označeném jako VCAP byly připojeny dva filtrovací kondenzátory 0,1 μF a 2,2 μF . Komunikace SPI je na pinech 15–18. JP4 označuje piny pro SCI komunikaci a konektor JP5 I2C komunikaci.

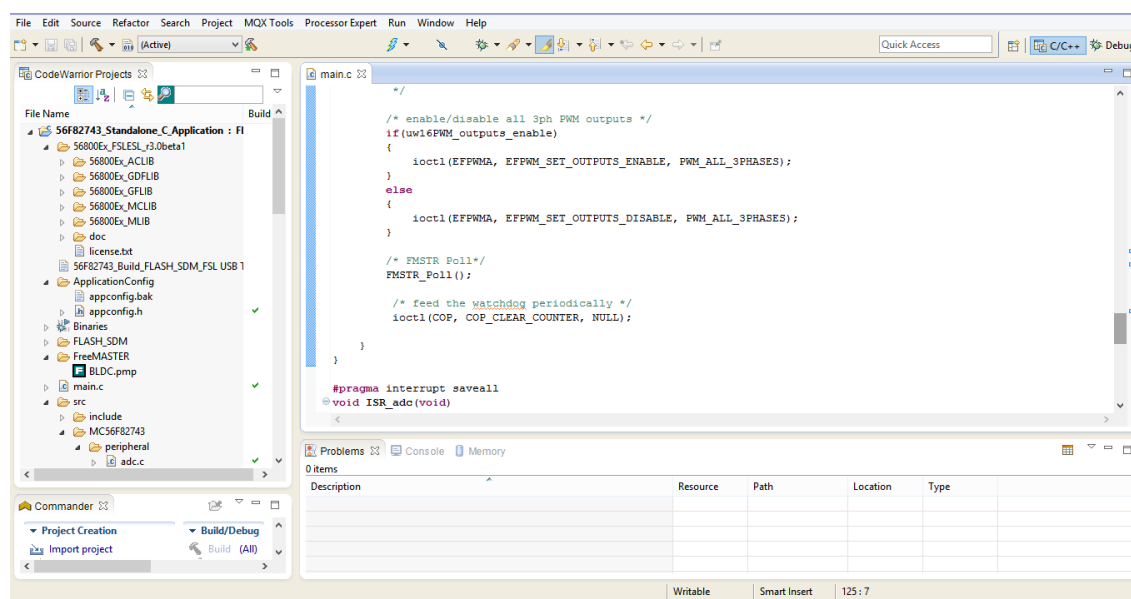


Obrázek 31 - Schéma zapojení mikroprocesoru MC56F82743

5 NÁVRH SOFTWARE

5.1 CodeWarrior

Programování mikroprocesoru se provádí pomocí vývojového prostředí CodeWarrior Development Studio. CodeWarrior je kompletně integrované vývojové prostředí (IDE), které uživateli poskytuje vysoký automatický přehled k urychlení a usnadnění vývoje i u těch nejsložitějších aplikací. Prostředí integruje editor, kompilátor a linker zdrojových kódů a zároveň uživateli poskytuje velké množství knihovných funkcí pro různé platformy. Prostředí CodeWarrior umožňuje programovat v několika jazycích symbolických adres jako assembleru, C nebo C++. CodeWarrior obsahuje také grafický debugger, který umožňuje uživateli krokovat vytvořenou aplikaci, čímž ji lze snadno ladit. Ukázka vývojového prostředí CodeWarrior je na obrázku 32.



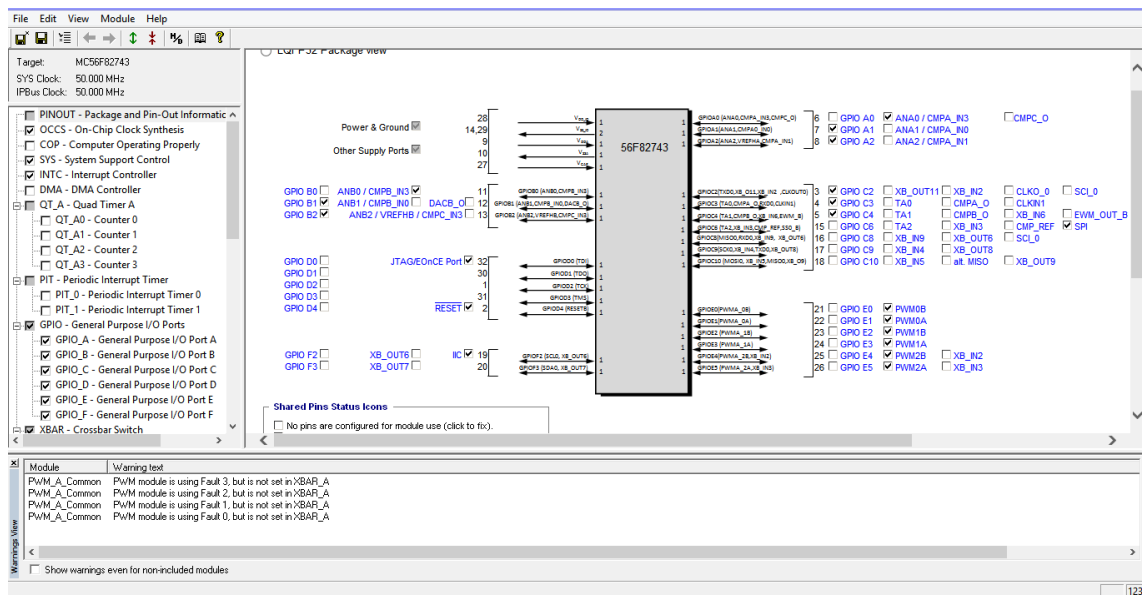
Obrázek 32 - Ukázka vývojového prostředí CodeWarrior

Mikroprocesor MC56F82743 byl naprogramován pomocí JTAG přes USB-TAP.

5.2 Quick Start

Vývojové prostředí Quick Start umožňuje uživateli vytvořit rychle a snadno aplikační zdrojový kód nezávisle na znalosti o architektuře mikropočítače. Součástí Quick Startu je také grafické prostředí GCT (Graphical Configuration Tool), které poskytuje efektivní nastavení chování mikropočítače.

GCT umožňuje uživateli graficky nastavovat čip pomocí tabulkového a grafického výběru, čímž se doba pro nastavení mikropočítače snižuje na minimum. Díky snadnému nastavení se práce výrazně zjednodušuje a nastavení se stává pohodlnější a přehlednější. Ukázkou prostředí GCT můžeme vidět na obrázku 33 níže.



Obrázek 33 - Ukáзка GCT pro nastavení čipu

Při nastavování čipu otevře CodeWarrior z běžícího projektu nástroj GCT. Ten z uloženého souboru s názvem *appconfig.h* vezme aktuální nastavení a zobrazí ho v GCT v grafické podobě. Jestliže konfigurační knihovna neobsahuje žádné nastavení, zobrazí se defaultní nastavení. Po otevření knihovny lze nastavit jakoukoliv periferii, funkci nebo činnost, která je na čipu dostupná. Po změně nastavení nebo uložení dojde k přepsání knihovny *appconfig.h*. Tyto uložená nastavení je možno použít pro nastavení a ovládání čipu pomocí příkazů *ioctl()*.

ioctl() příkazy jsou vlastně low-level drivery, pomocí kterých lze řídit funkčnost mikropočítače za běhu programu.

Ioctl (Input Output Control) nám umožňuje:

- Přístup do registru počítače
- Řídit všechny zdroje mikropočítače
- Aplikační kód nesahá přímo na registry
- Kompilátor ioctl příkazy překládá optimálně
- Zvyšuje čitelnost kódu
- Zjednodušuje přenositelnost aplikace

5.2.1 Freemaster

Freemaster slouží uživateli jako monitorovací a vizualizační nástroj při vývoji a ladění aplikací v reálném čase. Poskytuje uživateli celou řadu vlastností, jak grafické tak textové zobrazení proměnných. Také umožňuje přímé ovlivnění nebo nastavení proměnných. Freemaster může sloužit jako:

- Monitor v reálném čase
- Platforma vhodná k demonstračním účelům
- Grafický řídicí panel

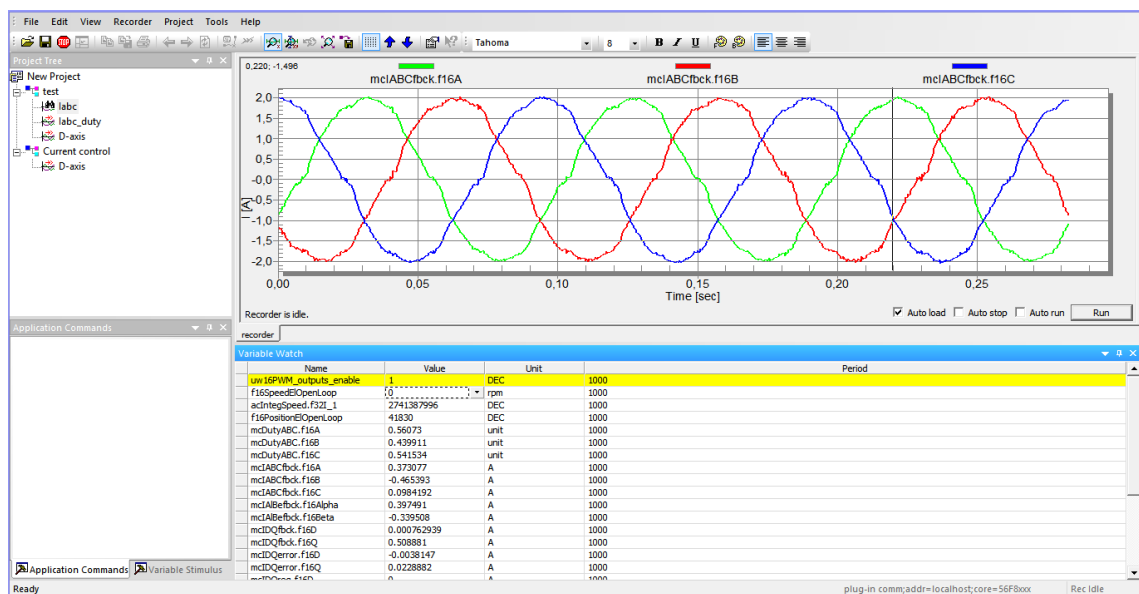
Freemaster tedy umožňuje přístup do paměťového místa s pevnou adresou. Na základě analýzy souborů ELF zjistí adresy všech globálních a statických proměnných. Dokáže vyčíst jejich typ, velikost a přesná jména, čímž umožňuje uživateli k těmto proměnným v kódu snadno přistupovat. K přístupu na běžící program v mikroprocesoru z počítače podporuje Freescale všechny typy komunikací jako:

- SCI, UART
- JTAG/EonCE
- BDM
- CAN

Freemaster je volně dostupný na stránkách firmy Freescale, kde jej lze zdarma stáhnout. Po instalaci je prostředí rozděleno do několika oken. Levá část je vyhrazena pro přesný přehled přidávaných objektů, které jsou seřazeny do stromové struktury. Mezi tyto objekty patří Recorder, Scope a Subblock. Pravá horní část je vyhrazena pro grafické sledování proměnných pomocí osciloskopu nebo záznamu. Sledování pomocí osciloskopu nám umožňuje zobrazit až 8 proměnných zároveň. Grafická podoba odpovídá grafu s časovou osou. Vysokorychlostní záznam neboli recorder nám

umožňuje zobrazit také 8 proměnných zároveň a poskytuje nám přesnou vzorkovací frekvenci. Data se dočasně uloží do bufferu v paměti procesoru, kterou je potřeba vyčíst pro záznam.

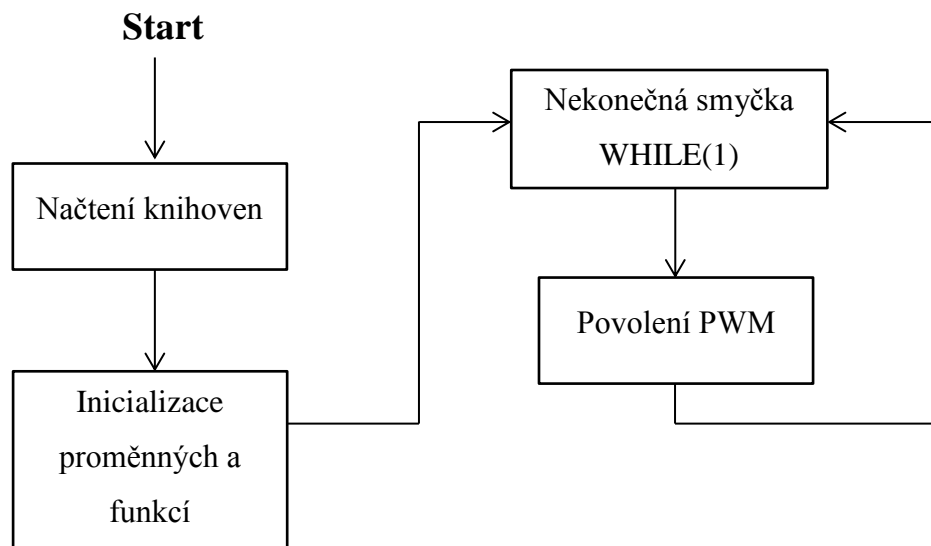
Dolní část okna je vyhrazena pro textové sledování proměnných. U každé proměnné lze sledovat jméno, hodnotu nebo jednotku. Zobrazení jednotlivých proměnných lze snadno upravovat a měnit. Proměnnou lze tedy transformovat do jiného formátu, který uživateli poskytne přijatelnější zobrazení. S proměnnou ve formátu fractional můžeme tedy pracovat přímo jako s napětím nebo proudem.



Obrázek 34 - Freemaster

5.3 Popis řídicího programu a jeho funkcí

V levé horní části vývojového prostředí CodeWarrior je zobrazena adresová struktura projektu, včetně všech souborů a knihoven, které projekt obsahuje. Ve složce se nachází soubor se jménem *main.c*, který obsahuje hlavní zdrojový kód. Běh programu popisuje následující obrázek, který zobrazuje hlavní diagram programu.



Obrázek 35 - Hlavní Diagram programu

Jako první se po spuštění provede načtení všech funkčních knihoven. Jde o knihovny pro řízení jednotlivých funkcí a periférií mikroprocesoru. Zde je vypsán seznam všech knihoven, které jsou v programu načteny:

- `#include "qs.h"`
- `#include "occs.h"`
- `#include "sys.h"`
- `#include "adc.h"`
- `#include "cop.h"`
- `#include "crc.h"`
- `#include "dac.h"`
- `#include "gpio.h"`
- `#include "hscmp.h"`
- `#include "iic.h"`
- `#include "intc.h"`
- `#include "pwm.h"`
- `#include "pit.h"`
- `#include "qtimer.h"`
- `#include "sci.h"`
- `#include "spi.h"`
- `#include "fmc.h"`
- `#include "ftfl.h"`
- `#include "mcm.h"`
- `#include "dma.h"`
- `#include "ewm.h"`
- `#include "pmc.h"`
- `#include "aoi.h"`
- `#include "xbar.h"`
- `#include "freemaster.h"`

Kromě těchto funkčních knihoven je potřeba taky načíst volně dostupné knihovny od firmy Freescale. Tyto knihovny umožňují používat funkce pro řízení motoru.

Knihovny obsahují algoritmy, které usnadňují uživateli psaní zdrojového kódu. Mezi tyto algoritmy patří například námi nejvíce využívané Clark a Park transformace, SVM (Space vector modulation), řada matematických funkcí a spoustu dalších.

Seznam načtených knihoven od firmy Freescale:

- `#include "aclib.h"`
- `#include "mlib.h"`
- `#include "mclib.h"`
- `#include "gflib.h"`
- `#include "gdflib.h"`

Po načtení knihoven je potřeba deklarovat a inicializovat proměnné. Všechny proměnné je potřeba deklarovat jako `static` aby k nim bylo možno přistupovat pomocí Freemastrua `ioctl` příkazů. Ukázka deklarace proměnné:

```
static      UWord16                uw16PWM_outputs_enable;  
static      MCLIB_3_COOR_SYST_T    mcDutyABC;
```

kde `CLIB_3_COOR_SYST_T` je struktura obsahující tři proměnné.

```
typedef struct  
{  
    Fracl6 f16A;  
  
    Fracl6 f16B;  
  
    Fracl6 f16C;  
}  
CLIB_3_COOR_SYST_T;
```

Po deklaraci proměnných je pak nutné inicializovat funkce a periférie mikroprocesoru. Inicializace se provede podle následujících `ioctl` příkazů:

```
ioctl(SYS, SYS_INIT, NULL);  
ioctl(COP, COP_INIT, NULL);  
ioctl(GPIO, GPIO_INIT_ALL, NULL);  
ioctl(INTC, INTC_INIT, NULL);
```

```

ioctl(XBAR_A, XBAR_A_INIT, NULL);

ioctl(EFPWMA, EFPWM_INIT, NULL);

ioctl(ADC, ADC_INIT, NULL);

ioctl(EFPWMA, EFPWM_CLEAR_FAULT_FLAGS, EFPWM_FAULT_0 |
      EFPWM_FAULT_1 | EFPWM_FAULT_2 | EFPWM_FAULT_3);

ioctl(SPI_0, SPI_INIT, NULL);

ioctl(INTC, INTC_INIT, NULL);

FMSTR_Init();

```

Pomocí těchto příkazů dojde k inicializaci a běhu periférií a funkcí jako jsou, vnitřní hodiny mikroprocesoru, vstupně výstupních pinů, pulsně šířkové modulace, měření analogových veličin, SPI komunikace, povolení přerušování a jiné. Poslední příkaz není ioctl ale je nutný pro inicializaci a povolení Freemasteru.

Po deklaraci a inicializaci všech proměnných a funkcí je napsána nekonečná smyčka pomocí WHILE příkazu, kde program neustále běhá dokola, dokud nepřijde přerušování. V nekonečné smyčce dochází pouze k povolení nebo zakázání PWM.

```

if (uw16PWM_outputs_enable)
{
    ioctl(EFPWMA, EFPWM_SET_OUTPUTS_ENABLE, PWM_ALL_3PHASES);
}
else
{
    ioctl(EFPWMA, EFPWM_SET_OUTPUTS_DISABLE, PWM_ALL_3PHASES);
}

```

Celé vektorové řízení probíhá ve funkci, do které program skočí, přijde-li přerušování. Funkce pro přerušování má následující tvar:

```

#pragma interrupt saveall

void ISR_adc(void)
{
    Zde probíhá vektorové řízení
}
#pragma interrupt off

```

Řízení probíhá přesně, jak bylo popsáno v kapitole 3. Nejprve je poslán po SPI příkaz pro čtení polohy, která je hned následně vyčtena a upravena na příslušnou hodnotu, která odpovídá řízení pro daný motoru.

```

ioctl(SPI_0, SPI_WRITE_DATA, 0xFFFF);

uw16SPI_data=ioctl(SPI_0, SPI_READ_DATA, NULL);

uw16SPI_data=uw16SPI_data&0x3FFF;

uw16SPI_data=uw16SPI_data<<2;

fposition=uw16SPI_data%(9362);

fposition=((fposition/(4681))-1);

f16Position=(Frac16)uw16SPI_data;

```

Jako další jsou změřeny dva proudy I_a a I_c . Třetí proud je dopočítán pomocí dvou změřených a to pomocí negace jejich součtu:

```

mcIABCfbck.f16A = sub((Frac16)ioctl(ADC, ADC_READ_SAMPLE, 0),
FRAC16(0.488524));
mcIABCfbck.f16C = sub((Frac16)ioctl(ADC, ADC_READ_SAMPLE, 8),
FRAC16(0.488524));
mcIABCfbck.f16B =MLIB_Neg16(MLIB_Add16(mcIABCfbck.f16A,
mcIABCfbck.f16C));

```

Po změření proudů se provedou transformace. Nejprve je provedena Clark transformace. Poté je upraven údaj o pozici rotoru na příslušnou hodnotu a je provedena Park transformace.

```

MCLIB_ClarkTrf(&mcIAlBefbck, &mcIABCfbck);

mcAngle.f16Sin = GFLIB_Sin12Tlr(fposition);

mcAngle.f16Cos = GFLIB_Cos12Tlr(fposition);

MCLIB_ParkTrf(&mcIDQfbck, &mcIAlBefbck, &mcAngle);

```

Po provedení transformací se provede proudová regulace Q a D složky.

```
mcIDQerror.f16D = MLIB_Sub16(mcIDQreq.f16D, mcIDQfbck.f16D);  
mcUDQreq.f16D   = GFLIB_ControllerPIp(mcIDQerror.f16D,  
                                       &gfParamsD, &i16PiSatFlagD);  
  
mcIDQerror.f16Q = MLIB_Sub16(mcIDQreq.f16Q, mcIDQfbck.f16Q);  
mcUDQreq.f16Q   = GFLIB_ControllerPIp(mcIDQerror.f16Q,  
                                       &gfParamsQ, &i16PiSatFlagQ);
```

Po provedení proudové regulace se použije zpětná Parkova transformace a Space Vector Modulation, ze kterých získáme požadovanou velikost PWM signálu.

```
MCLIB_ParkTrfInv(&mcUAlBeReq, &mcUDQreq, &mcAngle);  
  
uw16SVMSector = MCLIB_SvmStd(&mcUAlBeReq, &mcDutyABC);
```

Při známém požadovaném duty-cyclu jsou nastaveny výstupní PWM signály.

```
ioctl_four_param(EFPWMA_SUB0, EFPWMS_CENTER_ALIGN_UPDATE_MOD_CHAN  
                 NEL_23, EFPWMA_SUB0, mcDutyABC.f16A);  
  
ioctl_four_param(EFPWMA_SUB1, EFPWMS_CENTER_ALIGN_UPDATE_MOD_CHAN  
                 NEL_23, EFPWMA_SUB0, mcDutyABC.f16B);  
  
ioctl_four_param(EFPWMA_SUB2, EFPWMS_CENTER_ALIGN_UPDATE_MOD_CHAN  
                 NEL_23, EFPWMA_SUB0, mcDutyABC.f16C);
```

Posledními příkazy se potvrdí nastavení PWM a zruší se přerušení.

```
ioctl(EFPWMA, EFPWM_SET_LOAD_OK, EFPWM_SUBMODULE_0 |  
        EFPWM_SUBMODULE_1 | EFPWM_SUBMODULE_2);  
  
ioctl(ADC, ADC_CLEAR_STATUS_EOSI, ADC_CONVERTER_0 |  
        ADC_CONVERTER_1);
```

6 REALIZACE

6.1 BLDC motoru Multistar

6.1.1 Parametry udávané výrobcem

- 980 K_v – tedy 980 otáček za minutu na 1 V napájecího napětí
- Maximální proud 9 A
- Maximální výkon 99 W
- Odpor statorového vinutí 0,207 Ω

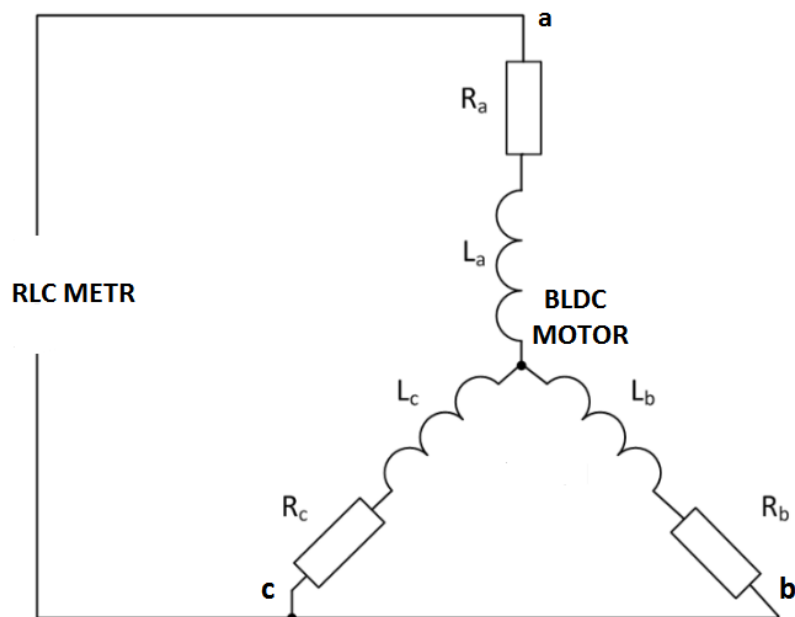


Obrázek 36 - BLDC Motor Multistar

6.1.2 Měření parametry

6.1.2.1 Odpor statoru R_s

Odpor statorového vinutí je definován mezi fázovou svorkou a středem vinutí. Odpor je závislý na teplotě. Pokud není uvedeno, při jaké teplotě byl odpor měřen, bývá obvykle určen při 25°C. Pro měření odporu statorového vinutí lze využít Ohmova zákona, tedy odpor lze změřit jakýmkoliv ohmmetrem, např. multimetrem. Pro měření našeho odporu byl využit LRC metr a odpor byl měřen při teplotě 22°C. Na obrázku 37 můžeme vidět zapojení RLC metru pro měření odporu statorového vinutí.



Obrázek 37 - Zapojení RLC metru pro měření odporu a indukčnosti,

Tabulka 9 - Odporu statorového vinutí

Svorky statorového vinutí	Naměřené hodnoty odporu R	Vypočítaný odpor R_s
[-]	[Ω]	[Ω]
a-b	0,243	0,125
b-c	0,246	
c-a	0,247	
a-bc	0,186	0,123
b-ac	0,185	
c-ab	0,184	

Odpor statorového vinutí je vypočten jako průměr poloviny změřeného odporu při použití dvou fázových svorek tedy

$$R_s = \frac{R}{2} \quad (18)$$

Při zapojení všech fázových svorek plátí vzorec:

$$R_s = \frac{2}{3} * R \quad (19)$$

Odpor statorového vinutí byl změřen přibližně 0,124 Ω , což s hodnotou udávanou výrobcem moc neodpovídá.

6.1.2.2 Indukčnost statoru v souřadnicích L_d a L_q

K zjištění indukčnosti statorového vinutí byl opět použit LRC metr. Pro zjištění indukčnosti je potřeba znát přesně polohu rotoru, která tuto indukčnost ovlivňuje. Zapojení RLC metru pro měření indukčnosti, které je stejné jako pro měření odporu, můžeme vidět na předchozím obrázku.

Indukčnost statoru se mění se změnou polohy. Proto měníme polohu rotoru, dokud nedosáhneme maximální a minimální indukčnost, kde minimální indukčnost odpovídá L_d a maximální L_q . Platí tedy:

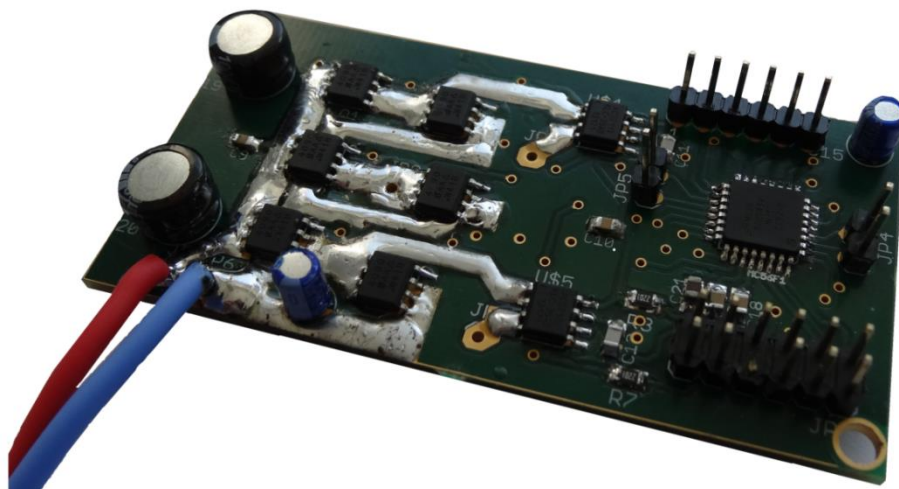
$$L_d = \frac{2}{3} * L_{min} \quad (20)$$

$$L_q = \frac{2}{3} * L_{max} \quad (21)$$

Minimální indukčnost byla změřena 24,8 μH tedy $L_d = 16,53 \mu\text{H}$ a maximální indukčnost 29,2 μH tedy $L_q = 19,47 \mu\text{H}$.

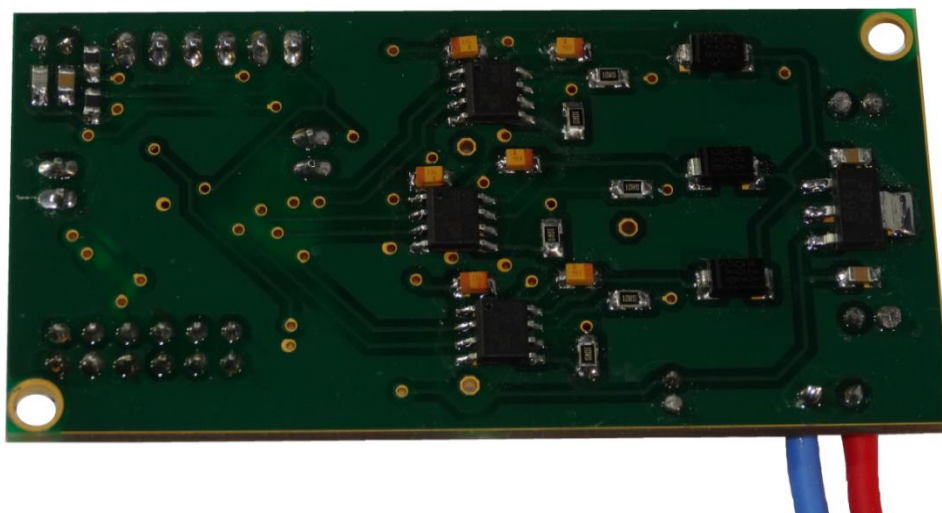
6.2 Vytvořený Hardware

Na následujícím obrázku 38 můžeme vidět DPS (Desku plošných spojů) ze strany označené jako TOP. V levém dolním rohu můžeme vidět červený a modrý přívod pro napájecí napětí. Úplně vlevo dva elektrolytické filtrovací kondenzátory, vedle nich šestici MOSFET tranzistorů a uprostřed snímače proudu. Vpravo uprostřed mikroprocesor MC56F82743 a kolem něj piny pro komunikaci po SPI, I2C, SCI a JTAG. Přesné osazení a názvy součástek můžeme vidět v příloze číslo 3.



Obrázek 38 - Foto DPS strana TOP

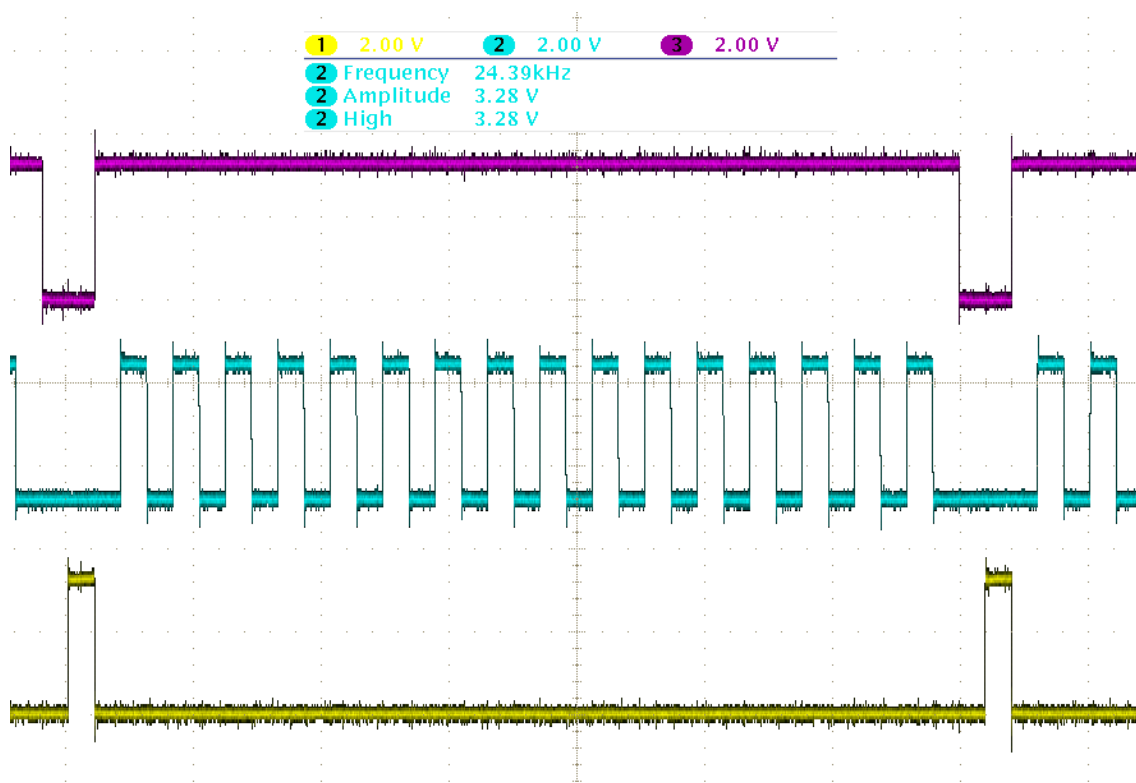
Na obrázku 39 můžeme vidět DPS ze strany BOTTOM. Úplně vpravo je osazen stabilizátor napětí na 3,3 V. Vedle něj vysokorychlostní diody sloužící k napájení boostovacích kondenzátoru uprostřed. Uprostřed DPS můžeme vidět trojici Driveru (budiče tranzistoru). Kompletní rozmístění s názvy použitých součástek můžeme vidět v příloze číslo 4.



Obrázek 39 - DPS strana BOTTOM

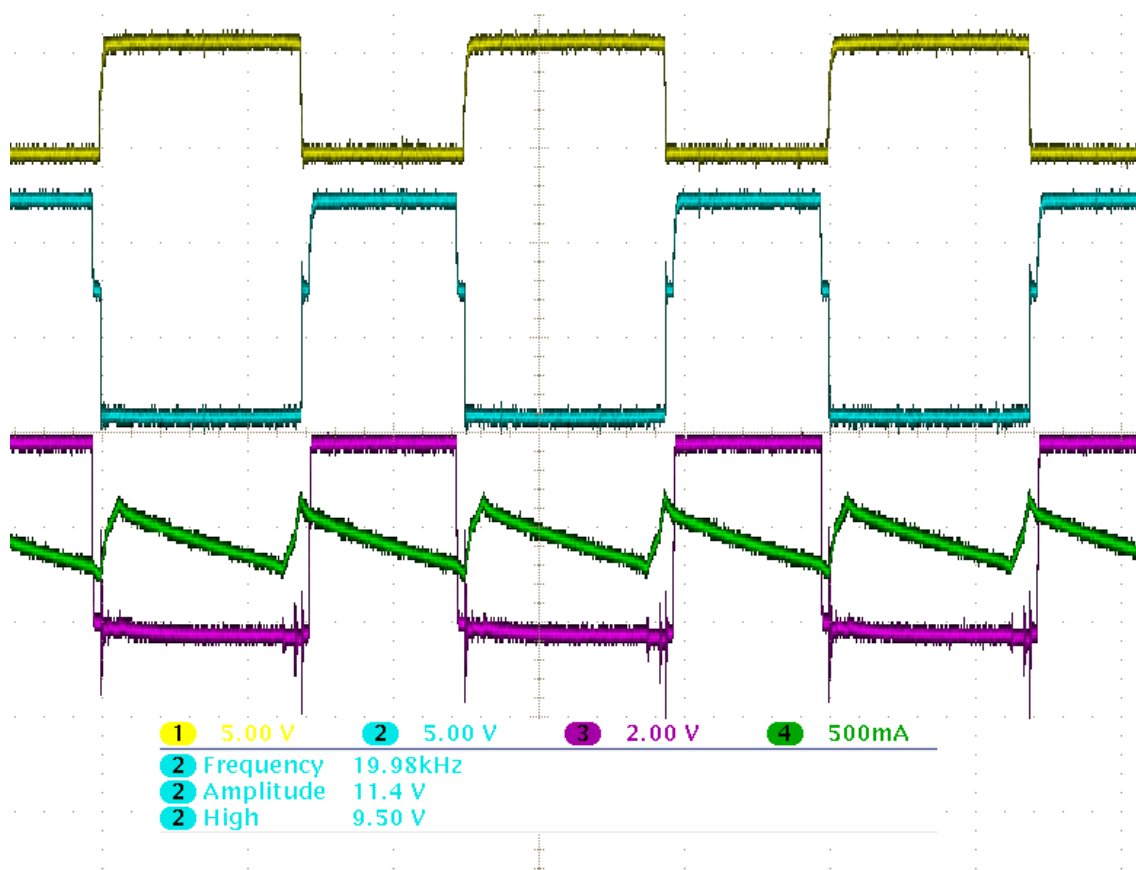
6.3 Průběhy a řídicí signály

Jako první je v programu zjištěna poloha rotoru. Pozice je zjištěna pomocí snímače, který komunikuje s řídicím obvodem po SPI. Princip SPI byl podrobně popsán v kapitole 3.4.1 a jeho reálné průběhy jsou zobrazeny na obrázku 40. Celkem jsou zobrazeny tři průběhy. Signál úplně dole označen žlutou barvou značí CS (Chip select). Celá komunikace začíná nastavením CS na hodnotu LOW. Pro čtení polohy je nutné do snímače vyslat READ příkaz, který je odeslán pomocí 0xFFFF, což značí samé jedničky. Tento signál se značí MOSI a na obrázku je znázorněn fialovou barvou. Synchronizace celé komunikace se provádí pomocí CLK (clocku), který má na obrázku tyrkysovou barvu. Na obrázku můžeme vidět celou funkční komunikaci.



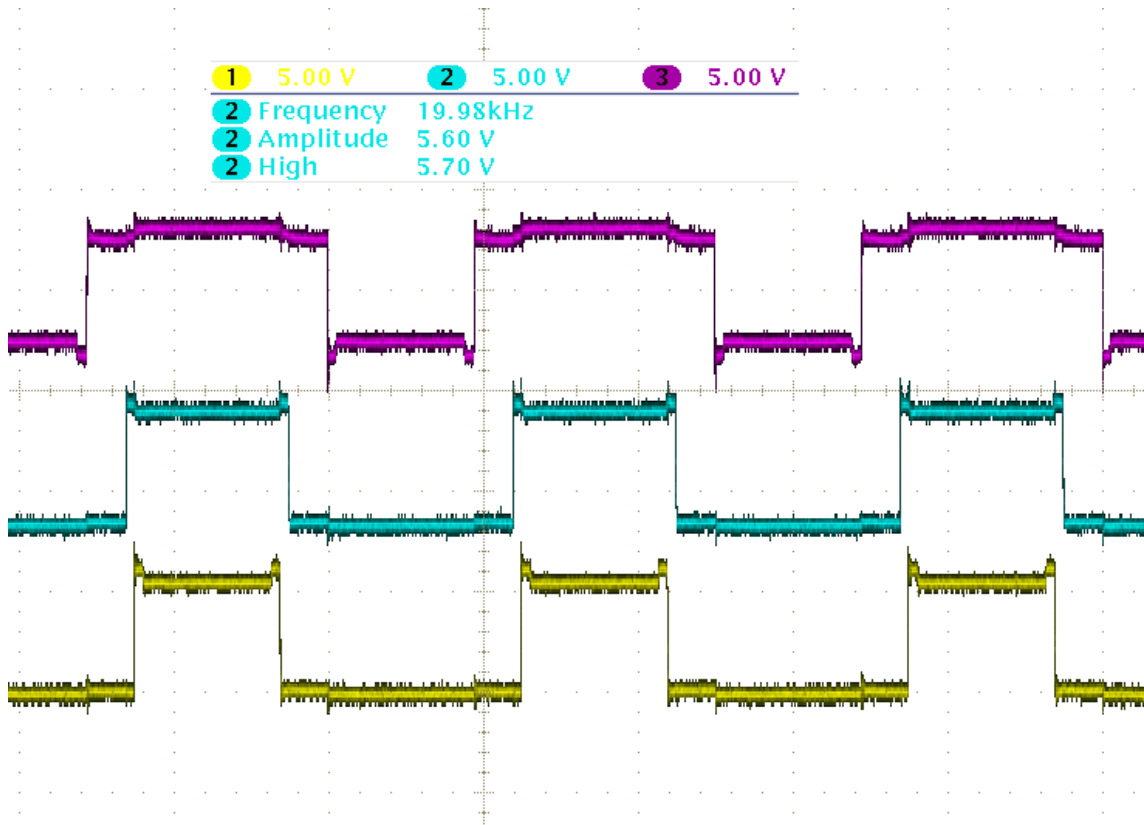
Obrázek 40 - SPI komunikace

Na dalším obrázku 41 můžeme vidět řídicí signály pro jednu větev měniče. Nahoře na obrázku jsou dva řídicí signály z budiče. Žlutý průběh značí řídicí signál pro dolní tranzistor, tyrkysový průběh pro horní tranzistor. U tyrkysového signálu si můžeme všimnout odlišné napěťové úrovně a schodu při náběžné a sestupné hraně. Jak již bylo vysvětleno v kapitole 4.1.2, při zavření spodního tranzistoru se na fázi motoru objeví napájecí napětí. Proto je pro otevření horního tranzistoru nezbytné dosáhnout větší napětí než je napájecí. V okamžiku, kdy přijde řídicí signál pro otevření horního tranzistoru, budič dodá vyšší napětí než je napájecí, čímž dojde k otevření horního tranzistoru. Otevření horního tranzistoru značí fialový signál. Při porovnání žlutého a fialového průběhu si můžeme všimnout taky Dead-timu, který je nastaven na 1 μ s. Poslední zelený signál nám zobrazuje, jak se mění proud v závislosti na spínání tranzistorů.



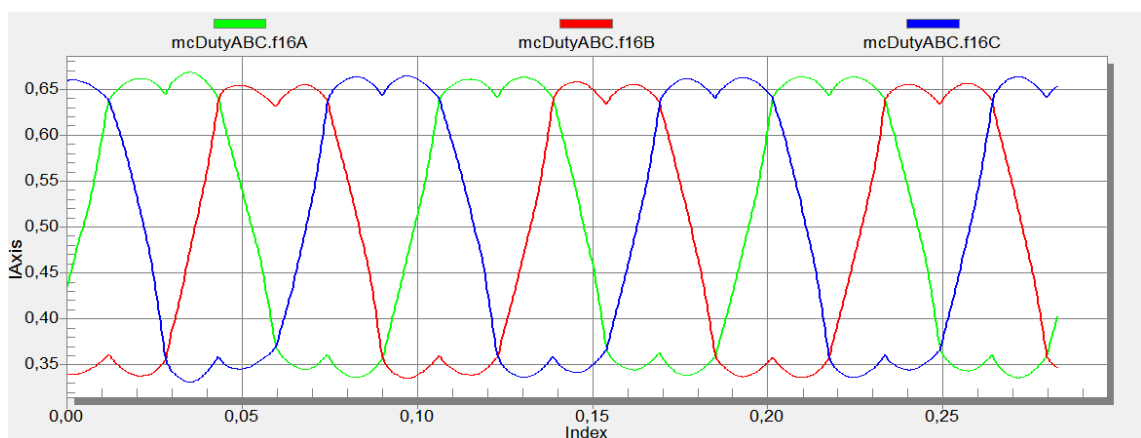
Obrázek 41 - Řídicí signály pro tranzistory

Obrázek 42 nám zobrazuje PWM na všech třech fázích. Přesto že duty-cycle je na každé fázi jiný, všechny tři jsou přesně nad sebou, tedy centrovány na střed. Podle frekvence osciloskopu bylo také ověřeno že PWM běží rychlostí 20 KHz.



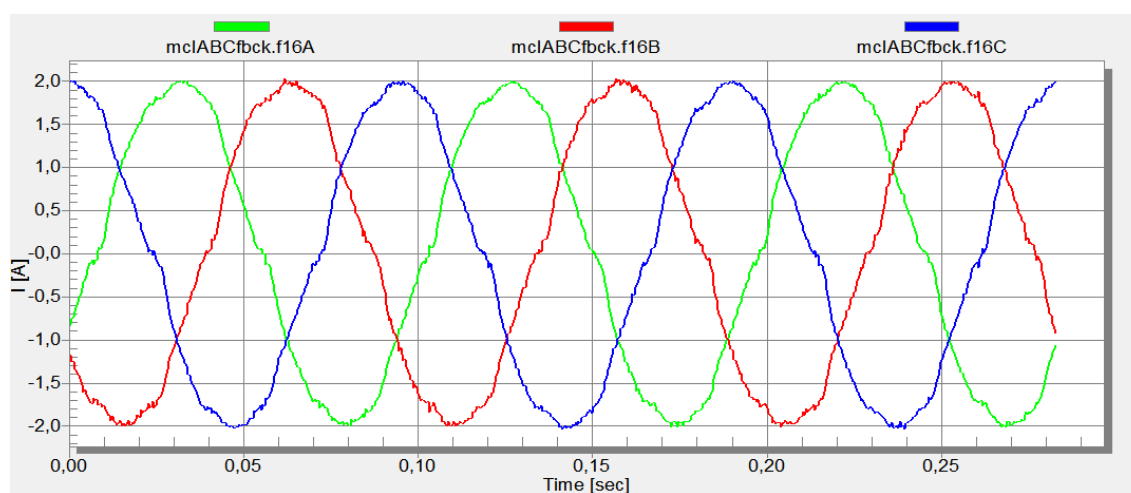
Obrázek 42 - Centrování PWM na střed ve všech třech fázích

Duty-cycle se při točení rotoru na všech třech fázích mění. Tato změna závisí na rychlosti otáčení rotoru. Závislost duty-cyclu ovlivňující PWM, tedy napětí na jednotlivých fázích, můžeme vidět na následujícím obrázku 43, kdy je tato závislost zobrazena při konstantní rychlosti.



Obrázek 43 - Závislost duty-cyclu na pozici rotoru při konstantní rychlosti

Na následujícím obrázku 44 můžeme vidět, jak se mění proud při otáčení rotoru konstantní rychlostí. Můžeme zde vidět, jak každý proud protékající jednotlivou fází motoru dosahuje sinusového tvaru. Také je dobře patrný převod do trojfázové soustavy, kde všechny tři proudy jsou proti sobě posunuty o 120° . Tyhle průběhy také dokazují plně funkční regulaci proudu, kdy každý proud dosahuje, při nastavení požadované Q složky na hodnotu 2, maximální úrovně 2 A. Na obrázku si můžeme všimnout menšího zvlnění proudu kolem hodnot blízko nule, způsobené dead-timem, jehož použití je ovšem nezbytné. Toto zvlnění by bylo možné kompenzovat softwarově, ovšem pro správnou funkčnost řízení to není nezbytné.



Obrázek 44 - Proudů na všech fázích při točení rotoru konstantní rychlostí

7 ZÁVĚR

Hlavním cílem této práce bylo seznámit se s BLDC motory a jejich řízením. Bylo potřeba navrhnout algoritmus pro řízení BLDC motoru a realizovat jej v LabView. Jako první bylo navrženo řízení pro školní motor. Základem tohoto řízení, nazývané „Six-step“, je určit komutační tabulku, pomocí které dochází ke správnému spínání jednotlivých fází elektronického střídače. Po určení komutační tabulky byl vytvořen pulzně modulovaný PWM signál, jehož střidu určuje regulátor. Značnou nevýhodou tohoto řízení je zvlnění momentu, proto bylo ve finále vybráno vektorové řízení, které využívá Clarkové a Parkovy transformace pro převedení z trojfázového systému se souřadnicemi a, b, c do dvoufázového systému se souřadnicemi d, q . Jde o senzorové řízení proto je tedy potřeba znát přesnou polohu rotoru. Pro školní motor byla poloha určována pomocí Hallových snímačů. Detekce rotoru pomocí Hallových snímačů je technika poměrně jednoduchá, protože jsou odolné vůči šumu a nejsou závislé na rychlosti rotoru. Finální BLDC motor MultiSTAR, který bude sloužit k pohánění robotického vozítka, neobsahuje Hallovy snímače, proto je potřeba použít jinou techniku pro detekci rotoru. Poloha rotoru je proto určována pomocí absolutního magnetického snímače polohy, který je umístěn kolmo k hřídeli motoru, na které je umístěn magnet. Snímač komunikuje pomocí SPI. Nevýhodou je, že analogová část snímače způsobuje posunutí úhlu při větších rychlostech, proto je potřeba pro zlepšení vlastnosti řízení tento úhel dopočítávat v závislosti na rychlosti.

Řídicí algoritmus vektorového řízení byl navržen v LabView s použitím funkčních knihoven pro řízení třífázového motoru. Navrhnutý řídicí algoritmus byl, podobně jako první řídicí algoritmus, vyzkoušen a doladěn pomocí CompactRIA v FPGA. Pak se mohlo přejít k návrhu vlastního hardwaru. Nejprve byla navržena a ozkoušena pouze výkonová část, jako je třífázový měnič, budiče tranzistorů, stabilizátor napětí a měření proudu. Měnič se skládá ze šesti tranzistorů, které vydrží spínat až do proudu 19,7 A. Tyto tranzistory je nutné spínat budičem s nábojovou pumpou, zejména kvůli horním tranzistorům, které nejsou zapojeny na zem ale na zátěž do fáze motoru. K měření proudu připadaly v úvahu dvě možnosti. První možnost byla pomocí měřícího odporu zapojeného jako bočník, na kterém lze měřit úbytek napětí. Toto měření vyžadovalo nutnost synchronizace, protože proud lze měřit pouze, je-li otevřen spodní

tranzistor. Proto byla použita druhá možnost, která je sice nákladnější, ovšem z hlediska jednoduchosti a ušetření místa na DPS pro nás lepší. Proto byl pro měření použit přímo snímač proudu. Tento snímač měří proudy v rozsahu do ± 25 A, a funguje na principu Hallového efektu, tedy lze jej zapojit přímo do fáze motoru, čímž lze měřit proud v každém okamžiku. Výstupní signál není potřeba nijak upravovat ani pro mikroprocesor, protože jeho výstup je v rozsahu 0–3,3 V.

Po odzkoušení funkčnosti výkonové části byla připojena i část řídicí. Jako řídicí obvod byl zvolen mikroprocesor s označením MC56F82743, který je určen díky svému výpočetnímu výkonu a rychlosti přímo pro řízení motorů. Tento mikroprocesor byl programován pomocí JTAG přes USB-TAP ve vývojovém prostředí CodeWarrior. Pomocí prostředí Quick Start byli nejprve naprogramovány jednotlivé periferie mikroprocesoru. Potom byl v CodeWarrior napsán řídicí algoritmus v jazyce C. V prostředí Freemaster byly jednotlivé proměnné nastaveny a zobrazeny společně s ostatními průběhy. Všechny možné průběhy, které souvisí s řízením motoru, jsou zobrazeny v poslední kapitole. Těmito průběhy byla taky potvrzena plná funkčnost navrhnutého hardwaru a softwaru, čímž bylo splněno zadání v plném rozsahu.

8 SEZNAMY

8.1 Seznam použitých součástek

Název	Množství	Popis	Označení
MC53F82743	1	32BIT DSC, 256KB FLASH, 100MHZ, LQFP-32	MC56F82743VLC
LM2937	1	LDO REGULATOR, 3.3V, 0.5A, 3SOT223	LM2937IMP- 3.3/NOPB
Q1-Q6	6	MOSFET tranzistor 100V, 19.7A, SOIC-8	SI4090D
IRS2301	3	GATE DRIVER, HI/LOW SIDE, 8SOIC	IRS2301SPBF
D1-D3	3	DIODE, AVALANCHE, 1.5A, 200V	BYG20D-E3/TR
C1-C6	6	CAP, TANT, 330NF, 25V, size 0805	TAJR334K025RNJ
C8,C15	2	CAP, ALU ELEC, 10UF, 16V, RAD	ECA1CAD100X
C19,C20	2	CAP, ALU ELEC, 100UF, 25V, RAD	ECEA1EKS101
C11,C12	2	AP, MLCC, C0G/NP0, 510PF, 50V, size 0805	MC0805N5 11J500CT
C7,C9,C10,C13, C17,C18,C21	7	CAP, MLCC, X7R, 100NF, 50V, size 0805	MC0805B 104K500CT
C16	1	CAP, MLCC, 100PF, 100V, size 0805	CGA4C2C0G2 A101J060AA
C22	1	2,2u/16V X7R MUR 10%, size 0805	CL21B225 KOFNNNG
ICS711	2	CURRENT SENSOR, 25A, 3.3V, 8SOIC	ACS711KLCTR
R1-R8	8	10R, REZISTOR, 10 OHM, 1%, size 0805	CR0805-FX
L1-L2	2	INDUCTOR, SHIELDED, 1UH, 20%, size 0805	MLZ2012A1R0MT
AS5048	1	Absolute position senzor, 14- bit, TSSOP 14	AS5048A-HTSP

8.2 Seznam obrázků

Obrázek 1 - Základní rozdělení elektrických pohonů.....	11
Obrázek 2 - Princip BLDC motoru, [1]	12
Obrázek 3 - Elektrické schéma statoru, [2].....	13
Obrázek 4 - Rotor a stator BLDC motoru, [2]	13
Obrázek 5 - Charakteristika točivého momentu v závislosti na rychlosti, [3].....	14
Obrázek 6 - třífázový střídač, [1].....	15
Obrázek 7 - Spínací vzory pro jednu elektrickou otáčku, [1].....	16
Obrázek 8 - Komutace BLDC motoru pro jednu elektrickou otáčku, [1]	16
Obrázek 9 - Vytvoření PWM signálu	21
Obrázek 10 - Spínání horního a spodního tranzistoru s dead-time.....	22
Obrázek 11 - Schéma zapojení rychlostního a proudového regulátoru, [6]	22
Obrázek 12 - Schéma zapojení pro transformaci a řízení	24
Obrázek 13 - AS5048, [9].....	28
Obrázek 14 - Obecné zapojení komunikace SPI	28
Obrázek 15 - Časový diagram SPI snímače AS5048, [9].....	29
Obrázek 16 - Graf závislosti rozdílů úhlů na otáčkách za sekundu.....	31
Obrázek 17 - Navržený třífázový střídač.....	32
Obrázek 18 - Schématická značka a rozmístění pinů SI4090DY,	33
Obrázek 19 - Zapojení budiče IRS2301S	34
Obrázek 20 - Vstupní a výstupní signály budiče IRS2301S.....	34
Obrázek 21 - Rozmístění pinů IRS2301S.....	34
Obrázek 22 - Zapojení operačního zesilovače pro měření proudu	37
Obrázek 23 - Zapojení ACS711 pro měření proudu.....	38
Obrázek 24 - Zapojení stabilizátoru.....	38
Obrázek 25 - Oddělení analogového napětí 3,3V pro procesor.....	39
Obrázek 26 - Blokové schéma mikroočítače MC56F82743, [16]	41
Obrázek 27 - Blokový diagram ADC modulu, [16]	42
Obrázek 28 - Blokový digram PIT modulu, [16].....	43
Obrázek 29 - Blokový diagram PWM modulu, [16]	45
Obrázek 30 - Blokový diagram PWM submodulu, [16].....	45
Obrázek 31 - Schéma zapojení mikroprocesoru MC56F82743.....	46

Obrázek 32 - Ukázka vývojového prostředí CodeWarrior	47
Obrázek 33 - Ukázka GCT pro nastavení čipu	48
Obrázek 34 - Freemaster	50
Obrázek 35 - Hlavní Diagram programu	51
Obrázek 36 - BLDC Motor Multistar	56
Obrázek 37 - Zapojení RLC metru pro měření odporu a indukčnosti,	57
Obrázek 38 - Foto DPS strana TOP	58
Obrázek 39 - DPS strana BOTTOM.....	59
Obrázek 40 - SPI komunikace	60
Obrázek 41 - Řídicí signály pro tranzistory	61
Obrázek 42 - Centrování PWM na střed ve všech třech fázích	62
Obrázek 43 - Závislost duty-cyclu na pozici rotoru při konstantní rychlosti	62
Obrázek 44 - Proudů na všech fázích při točení rotoru konstantní rychlostí.....	63

8.3 Seznam tabulek

Tabulka 1 - Porovnání vlastností BLDC s kartáčovým DC motorem	18
Tabulka 2 - Porovnání vlastností BLDC a AC indukčního motoru.....	18
Tabulka 3 - Komutační tabulka	19
Tabulka 4 - Rozšířená komutační tabulka	20
Tabulka 5 - Souhrn typických vlastností pro jednotlivé detekce polohy rotoru, [8]	27
Tabulka 6 - časové parametry SPI	29
Tabulka 7 - Popis jednotlivých bitů z přijatých dat	30
Tabulka 8 - Závislost rozdílů úhlů na otáčkách.....	31
Tabulka 9 - Odporu statorového vinutí.....	57

8.4 Seznam použité literatury

- [1] Lepka, J. a Grasblum, P., Použití mikroprocesorů pro řízení pohonů s BLDC motory. Dostupné z WWW:
http://www.crr.vutbr.cz/system/files/brozura_09_1111.pdf
- [2] AVR MICROCONTROLLERS. *AVR493: Sensorless Commutation of Brushless DC Motor (BLDC) using AT90PWM3 and ATAVRMC100*. 12/2006. Dostupné z:
<http://www.atmel.com/Images/doc7658.pdf>
- [3] MICROCHIP TECHNOLOGY INC. *AN885: Brushless DC (BLDC) Motor Fundamentals*. 07/28/2003. Dostupné z WWW:
<http://ww1.microchip.com/downloads/en/AppNotes/00885a.pdf>
- [4] VOREL, Pavel. *Synchronní stroje s permanentními magnety*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2005, 109 s. ISBN 80-720-4417-6.
- [5] SKALICKÝ, Jiří. *Elektrické servopohony*. 2. vyd. Brno: Vysoké učení technické, 2001, 86 s. ISBN 80-214-1978-4
- [6] BLAHA, P. a VAVŘÍN, P.: *Řízení a regulace 1*. Vysoké učení technické v Brně, 2005, 212 s.
- [7] CAHA, Zdeněk. *Elektrické pohony*. 1. vyd. Praha: SNTL, 1990, 359 s. ISBN 80-030-0418-7.
- [8] Sensorless Technique for BLDC Motors, Daniele Gambetta [online].
Dostupné z WWW:
<https://eprints.usq.edu.au/1427/>
- [9] AUTRIA MICRO SYSTEMS. *AS5048A/AS5048B: Magnetic rotary encoder*.
Dostupné z WWW:
<http://www.ams.com/eng/Products/Position-Sensors/Rotary-Magnetic-Position-Sensors>
- [10] Štulrajter, M.; Hrabovcová, V.; Franko, M.: Permanent magnets synchronous motor control theory. *Journal of Electrical Engineering*, ročník 58, c. 2, 2007: s. 79–84.

- [11] FREESCALE. *56800Ex_ACLIB: Advanced Control Library*, User reference manual. 02/2014. Dostupné z WWW:
http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800Ex_ACLIB.pdf
- [12] FREESCALE. *56800Ex_MLIB: Math Library*, User reference manual. 02/2014. Dostupné z WWW:
http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800Ex_MLIB.pdf
- [13] FREESCALE. *56800Ex_MCLIB: Motor control library*, User reference manual. 02/2014. Dostupné z WWW:
http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800Ex_MCLIB.pdf
- [14] FREESCALE. *56800Ex_GFLIB: General Digital Filters Library*, User reference manual. 02/2014. Dostupné z WWW:
http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800Ex_GFLIB.pdf
- [15] FREESCALE. *56800Ex_GDFLIB: General Digital Filters Library*, User reference manual. 02/2014. Dostupné z WWW:
http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800Ex_GDFLIB.pdf
- [16] FREESCALE. *MC56F827xx Data Sheet: Reference Manual*. 10/2013. Dostupné z WWW:
http://cache.freescale.com/files/32bit/doc/ref_manual/MC56F827XXRM.pdf
- [17] FREESCALE. *DSP56800E_Quick_Start_User_manual*, DSC56800EXQSUG, 07/2013

8.5 Seznam příloženého CD

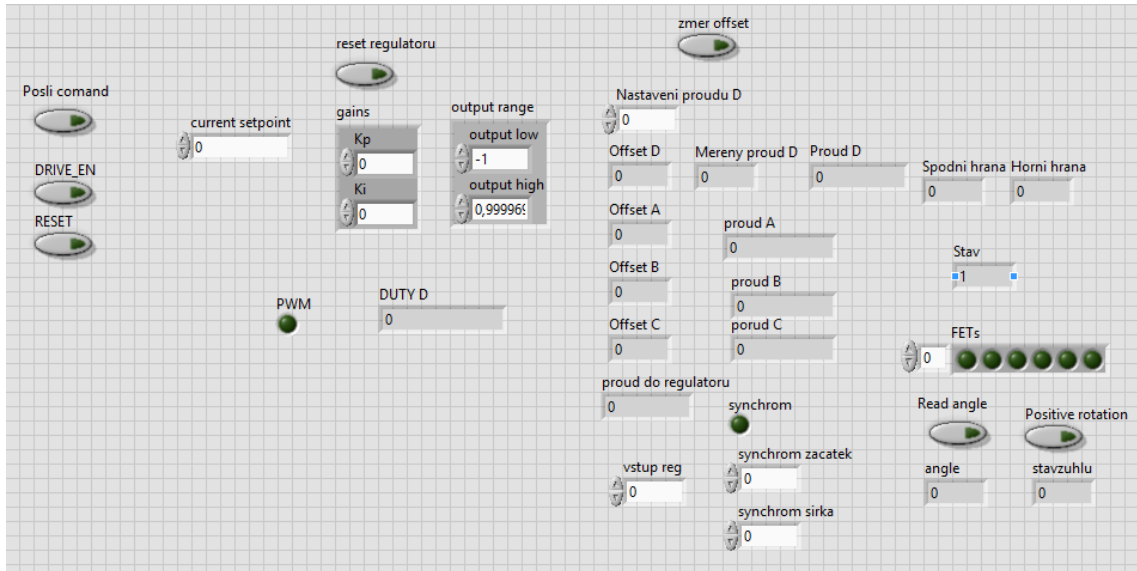
- Hlavní dokument samostatné práce
ZbranekPetr_2014.pdf
- Program pro řízení BLDC motoru v LabView pomocí Six-Step komunikace
FPGA_BLDC_Six-step_řízení.vi
- Program pro řízení BLDC motoru v LabView pomocí vektorového řízení
FPGA_BLDC_Vektorové_řízení.vi
- Schéma a deska v programu EAGL
EAGL_DPS_BLDC.brd
EAGL_DPS_BLDC.sch
- Kompletní program pro řízení BLDC motoru v CodeWarrior včetně všech funkčních a řídicích knihoven
CodeWarrior_řízení_BLDC

8.6 Seznam Příloh

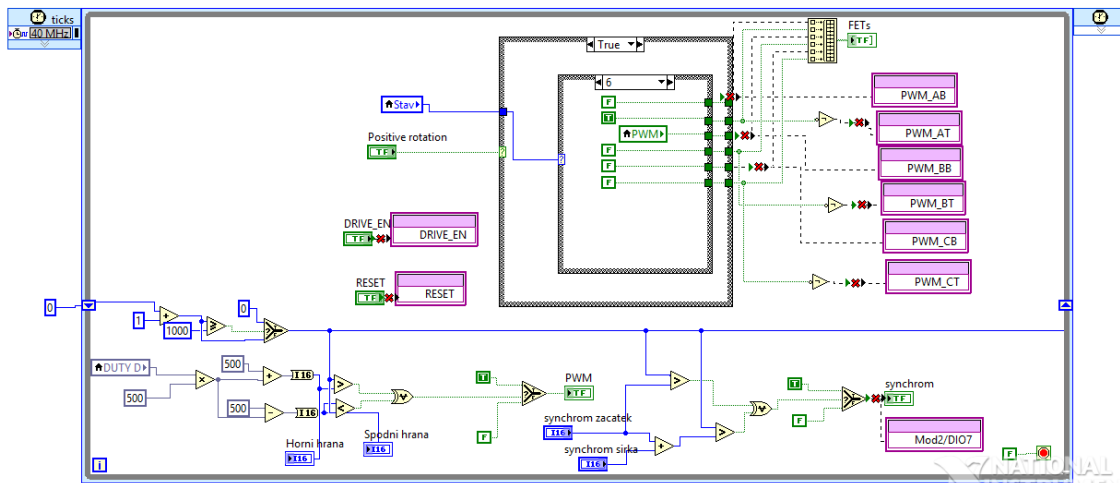
Příloha číslo 1 – Ukázka ovládacího panelu pro řízení BLDC motoru v LabView.....	73
Příloha číslo 2 – Ukázka programu pro řízení motoru BLDC v LabView.....	73
Příloha číslo 3 – DPS strana TOP.....	74
Příloha číslo 4 – DPS strana BOTTOM.....	75
Příloha číslo 5 – Kompletní schéma zapojení.....	76

PŘÍLOHY

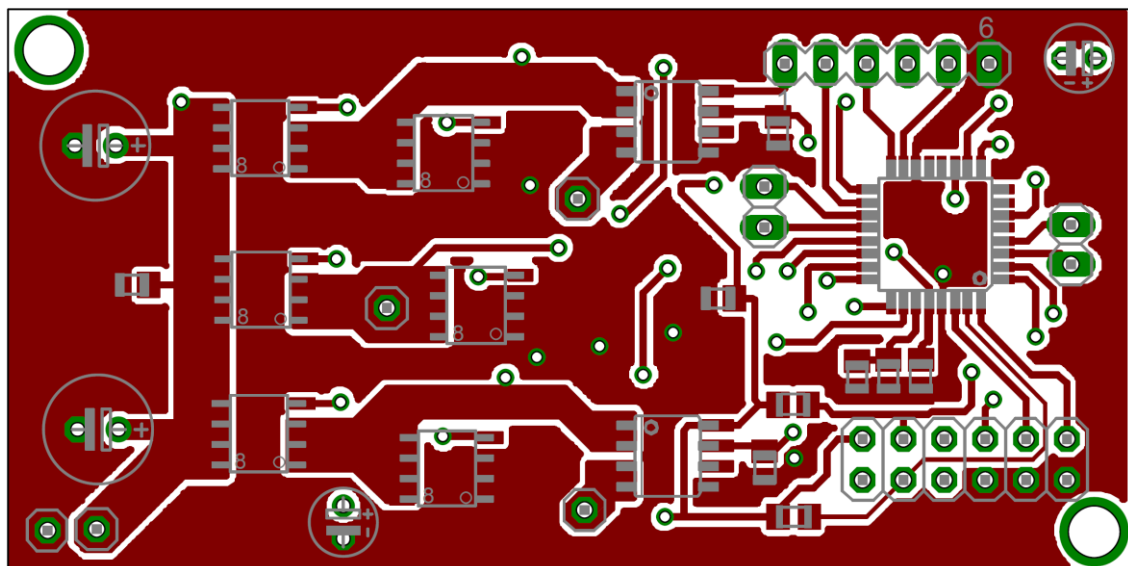
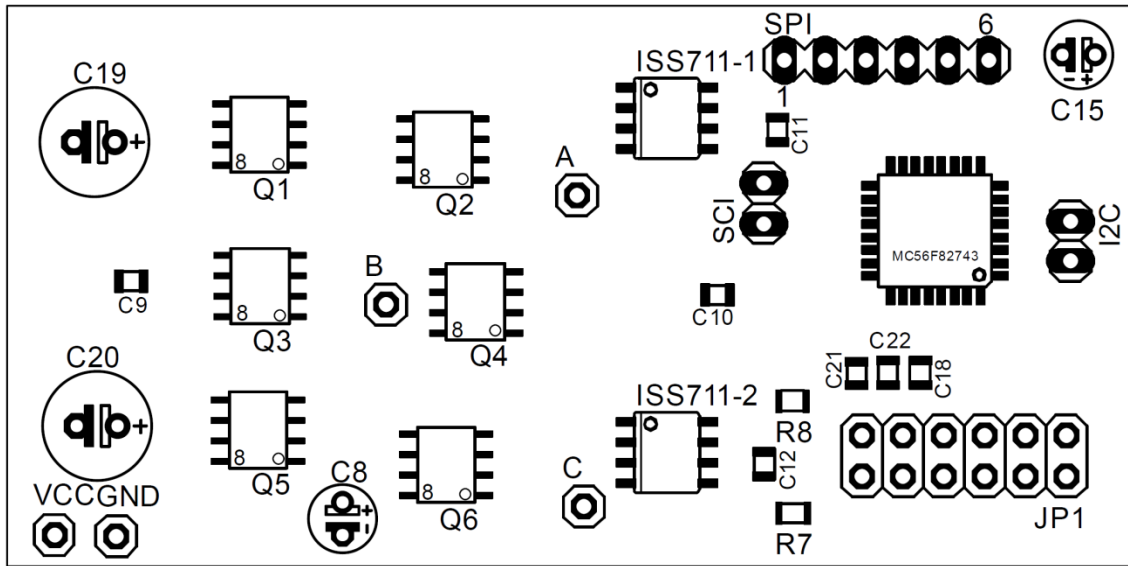
Příloha číslo 1 – Ukázka ovládacího panelu pro řízení BLDC motoru v LabView



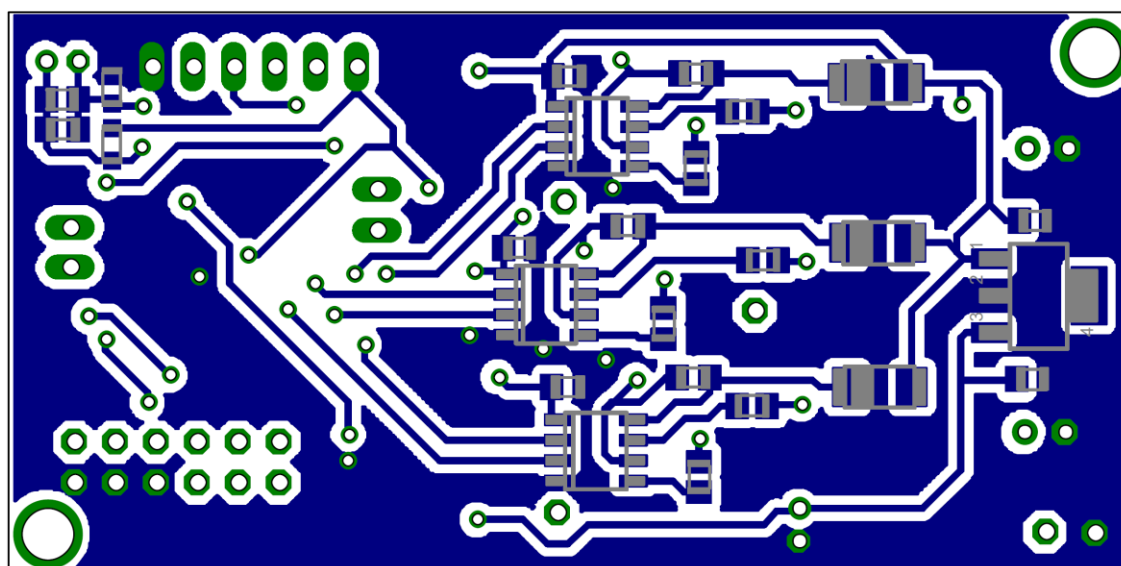
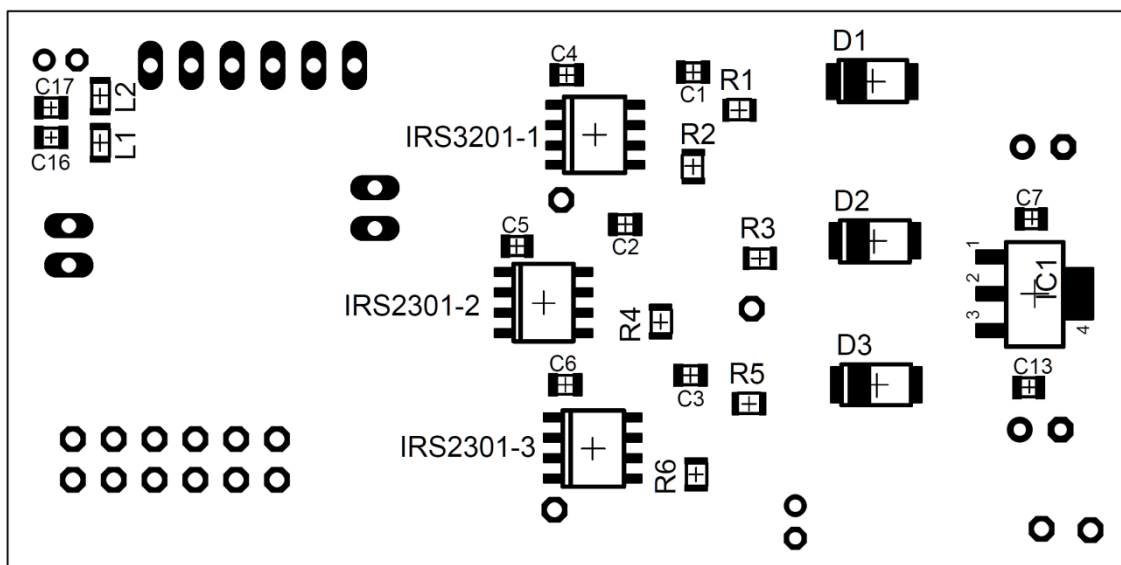
Příloha číslo 2 – Ukázka programu pro řízení motoru BLDC v LabView



Příloha číslo 3 – DPS strana TOP



Příloha číslo 4 – DPS strana BOTTOM



Příloha číslo 5 – Kompletní schéma zapojení

