



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra elektrických pohonů a trakce

Řízení BLDC motoru platformou STM-NUCLEO-IHM001

Control of BLDC Machine by STM-NUCLEO-IHM001

Bakalářská práce

Studijní program: Elektrotechnika, energetika a management

Studijní obor: Aplikovaná elektrotechnika

Vedoucí práce: Ing. Vít Hlinovský, CSc.

Kryštof Kuzma

Praha 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	Kuzma	Jméno: Kryštof	Osobní číslo: 457023
Fakulta/ústav:	Fakulta elektrotechnická		
Zadávací katedra/ústav:	Katedra elektrických pohonů a trakce		
Studijní program:	Elektrotechnika, energetika a management		
Studijní obor:	Aplikovaná elektrotechnika		

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Řízení BLDC motoru platformou STM -NUCLEO-IHM001

Název bakalářské práce anglicky:

Control of BLDC Machine by STM-NUCLEO-IHM001

Pokyny pro vypracování:

- 1) Provedte studii metod řízení BLDC motorů
- 2) Sepište HW a SW požadavky pro realizaci řídicí strategie BLDC motorů
- 3) Realizujte software pro řízení BLDC motoru platformou STM -NUCLEO-IHM001

Seznam doporučené literatury:

- [1] HRABOVCOVÁ Valéria. Moderné elektrické stroje, EDIS ŽU, 2001
- [2] ZHAO, Jian / YU, Yangwei. Brushless DC Motor Fundamentals Application Note. MSP [online]. 2011
- [3] ST Microelectronic. STM32F302x6 STM32D302x8. ST Microelectronic [online]. 2017

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vít Hlinovský, CSc., katedra elektrických pohonů a trakce FEL

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

Datum zadání bakalářské práce: **26.04.2018** Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Vít Hlinovský, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

PROHLÁŠENÍ

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

V Praze dne

.....

podpis

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu práce Ing. Vítu Hlinovskému, CSc. za zapůjčení přípravku P-NUCLEO-IHM001 a vhodné studijní literatury. Děkuji mu také za to, že mi umožnil přístup k osciloskopům. Dále bych chtěl poděkovat své rodině za morální a materiální podporu.

ABSTRAKT

Cílem práce je seznámit čtenáře s problematikou BLDC motorů a jejich řízení. Úvodem budou rozebrány jednotlivé vlastnosti BLDC motorů a jejich konstrukční prvky. Zmíněn zde bude matematický model motoru a jeho charakteristika. Dále bude řeč o způsobu jejich řízení. Důraz bude kladen zejména na způsob vyhodnocování polohy rotoru, a to především na bezsenzorovou metodu. S tím souvisí seznámení se s firmou ST Microelectronic, potažmo jejich platformou STM32F302R8 a příslušným modulem IHM07M1 určeným k řízení BLDC motorů. Bude zde také opodstatněn výběr vývojového prostředí. V praktické části bude podrobně probrán zdrojový kód, řídicí vývojovou platformu, a to včetně vývojových diagramů. V závěru pak budou uvedeny oscilogramy hodnot naměřených při chodu motoru na základě tohoto kódu.

KLÍČOVÁ SLOVA

BLDC, bezkartáčový stejnosměrný motor, komutace, rotor, stator, Six-step komutace, PWM modulace, pulsně šířková modulace, bezsenzorové řízení, Hallovy sondy, BEMF, zpětné elektromotorické napětí driver, ST Microelectronics, STM32F302R8, IHM07M1, IHM001, software

ABSTRACT

The focus of the thesis is to introduce the BLDC motors and their driving problematics. In the beginning, the individual properties of BLDC engines and their structural elements will be analyzed. The mathematic model of the motor will be mentioned there as well. Part of the work will be also dedicated to the driving problematics. Which is asociated with the rotor position detection. Especially important will be the part about rotor position capturing, using sensorless method. Project is related to STM32F302R8 development board and the IHM07M1 modul, designed to control BLDC motors. Development board and modul is produced by ST Microelectronic company. There will be also substantiate choice of workbenches. In the practical part the source code including the flowchart will be closely explained. In the end there will be also enclosed oscilogrames, captured on the base of the mentioned source code driving.

KEYWORDS

BLDC, brushless direct current motor, comutation, rotor, stator, Six-step comutation, PWM modulation, pulse width modulation, sensorless driving, Hall sensors, BEMF, back electromotive force driver, ST Microelectronics, STM32F302R8, IHM07M1, IHM001, software

Obsah

Úvod	11
1 BLDC motor	12
1.1 Konstrukce BLDC motorů.....	12
1.1.1 Magnety	12
1.1.2 Rotor	13
1.1.3 Stator.....	14
1.1.4 Hallový senzory	15
1.1.5 Rotační enkodéry	16
1.2 Srovnání BLDC a jiných typů motorů.....	18
2 Teorie řízení BLDC motorů	20
2.1 Komutace	20
2.2 Driver	23
2.3 Vyhodnocování polohy rotoru	25
2.3.1 Senzorické snímání polohy.....	25
2.3.2 Bez-senzorové určování polohy.....	26
2.4 Řízení rychlosti	27
3 Matematický model BLDC motoru	28
4 Hardware a software pro řízení BLDC motoru	32
4.1 STM32F302R8	32
4.2 X-NUCLEO-IHM07M1	34
4.3 ARM Mbed OS	36
5 Softwarové řízení motoru	37
5.1 Softwarová Six-step komutace	37
5.2 Finální verze programu.....	40

6	Naměřené průběhy	53
6.1	Zpětné elektromotorické napětí (BEMF).....	53
6.2	Proudová zpětná vazba	54
6.3	Pulsně šířková modulace (PWM)	55
6.4	Napájení motoru	56
7	Závěr	57
	Literatura	58
	Zdroje použitých obrázků	60

SEZNAM OBRÁZKŮ

Obrázek 1 - Konstrukce rotoru	14
Obrázek 2 - Náhradní schémata vinutí	15
Obrázek 3 - Konstrukce statoru	15
Obrázek 4 – Hallova sonda	16
Obrázek 5 - Rotační enkodéry	17
Obrázek 6 - Komutace	20
Obrázek 7 – Zjednodušené řízení komutace	21
Obrázek 8 – Vektor magnetického pole v průběhu komutace	22
Obrázek 9 - Zapojení polovodičových součástek řídicích komutací	23
Obrázek 10 - Výstup Hallových sond	25
Obrázek 11 - Detekce průchodu BEMF nulou	26
Obrázek 12 – Typické průběhy PWM signálu	27
Obrázek 13 - Náhradní schéma motoru pro matematický model	28
Obrázek 14 - Závislost momentu na otáčkách	31
Obrázek 15 - Blokové schéma STM32F302R8	33
Obrázek 16 - STM32F302R8	34
Obrázek 17 - P-NUCLEO-IHM001	35
Obrázek 18 - Schéma zapojení čipu L6230	36
Obrázek 19 - Vývojový digram softwarového řízení	41
Obrázek 20 - Vývojový diagram BEMF detekce a komutace	48
Obrázek 21 - BEMF oscilogram	53
Obrázek 22 - Oscilogram proudové zpětné vazby	54
Obrázek 23 - Oscilogram PWM	55
Obrázek 24 - Oscilogram napájení motoru	56

SEZNAM TABULEK

Tabulka 1 - Srovnání BLDC a AC motoru	18
Tabulka 2 - Srovnání BLDC a kartáčového DC motoru	19
Tabulka 3 - Spínání komutace	21
Tabulka 4 - Výstupy Hallových sond	26
Tabulka 5 - Spínání fází při komutaci	38

SEZNAM ROVNIC

Rovnice 1 – Fázové napětí a	29
Rovnice 2 – Fázové napětí b	29
Rovnice 3 – Fázové napětí c	29
Rovnice 4 – Vztah mezi vlastními indukčnostmi symetrického motoru	29
Rovnice 5 – Vztah mezi vzájemnými indukčnostmi symetrického motoru	29
Rovnice 6 - Vztah mezi odpory jednotlivých vinutí symetrického motoru.....	29
Rovnice 7 – Zjednodušený vztah fázového napětí a.....	29
Rovnice 8 – Zjednodušený vztah fázového napětí b	29
Rovnice 9 – Zjednodušený vztah fázového napětí c.....	29
Rovnice 10 – Elektromotorické napětí pro elektrické stupně rotoru θ_e	30
Rovnice 11 – Elektromotorické napětí pro pro elektrické stupně rotoru $\theta_e - \frac{2\pi}{3}$	30
Rovnice 12 – Elektromotorické napětí pro pro elektrické stupně rotoru $\theta_e + \frac{2\pi}{3}$	30
Rovnice 13 – Pohybová rovnice	30
Rovnice 14 – Moment hnací	30
Rovnice 15 – Moment dynamický	30
Rovnice 16 – Vzath mezi mechanickou a elektrickou úhlovou rychlostí.....	31

ÚVOD

V dnešní době moderních technologií jsou na elektrické pohony kladeny čím dál tím větší nároky. Lidstvo se snaží vyvíjet výkonnější motory s delší životností, které jsou navíc jednodušší po konstrukční stránce a kladou menší nároky na údržbu. Právě tyto parametry splňují BLDC motory. Jsou tedy používány čím dál tím více v různých odvětvích průmyslu a pomalu nahrazují zastaralé kartáčové motory.

BLDC motory jsou z konstrukčního hlediska tvořeny rotorem a státorem. Stator tvoří cívky, obvykle navinuté na pólových nástavcích. Pomocí řídicí jednotky je v nich měněna polarita a regulován proud dle potřeb řízení motoru. Rotor je tvořen permanentními magnety, které jsou na něm pevně uchyceny. Ty reagují na změnu elektromagnetického pole cívek a pohybují s rotorem. Motor může být realizován dvěma základními způsoby. První varianta zahrnuje rotor, jehož magnety jsou umístěny tak, že obklopují stator. V druhém případě je rotor obklopen cívkami. Tato specifická konstrukce odstraňuje doposud nejvíce namáhanou část motoru, komutátor. Motor se tak stává odolnějším a spolehlivějším.

Díky své nenáročnosti na údržbu a zpravidla i bezporuchovosti nachází BLDC motory široké spektrum uplatnění. Na jejich aplikaci lze narazit takřka všude. Jako příklad lze uvést pevné disky, CD přehrávače, různé nástroje, větráky a spoustu dalších. Uplatnění nachází také v mnoha průmyslových aplikacích. Příkladem může být průmysl automobilový, letecký, medicínský a v průmyslové automatizaci. Právě zde je klíčová jejich spolehlivost. V poslední době lze tyto motory nalézt také v různé řadě dopravních prostředků.

Řízení BLDC motorů vyžaduje změnu polarity cívek statoru v určitém sledu. Právě pořadí a rychlost spínání cívek je vyhodnocována řídicí jednotkou. Ta může motor řídit několika způsoby lišícími se zapojením daného motoru. Primárně lze druhy řízení rozdělit na řízení s pomocí senzorů a bezsenzorové. Každá možnost řízení má své výhody. Bezsenzorové řízení nevyžaduje další součástku navíc, pro detekci pozice motoru. Lze tak u výsledného výrobku snížit cenu. Preciznost a variabilita řízení je tím však omezena. V případě senzorového řízení jsou nejčastěji používány Hallovy senzory, které dokáží detekovat pozici rotoru s větší přesností. S danou informací pak lze řízení přizpůsobit aktuálním požadavkům na chod motoru. Řízení je pak efektivnější a plynulejší. Obecně se dá říci, že právě zajištění vhodné komutace je nejsložitějším aspektem řízení BLDC motorů.

1 BLDC MOTOR

Z historického hlediska se jedná o jeden z novějších druhů motoru. Název BLDC v sobě ukrývá anglickou zkratku slov brushless direct current motor, tedy přeloženo do češtiny bezkartáčový stejnosměrný motor. Jak z názvu vyplývá, BLDC motory jsou napájeny stejnosměrným proudem.

Tyto motory lze řadit mezi skupinu synchronních strojů, protože magnetické pole rotoru a statoru rotuje stejnou rychlostí, tedy synchronně. Jejich komutace je zajištěna elektronicky. Není tedy nutné používat komutátor, který je nejvíce namáhanou součástí stejnosměrných kartáčových motorů. Zvyšuje se tedy odolnost motoru a jeho náročnost na údržbu klesá. Na druhou stranu je potřeba soustavu s BLDC motorem opatřit řídicí jednotkou, tzv. driverem, která zajišťuje komutaci.

1.1 Konstrukce BLDC motorů

BLDC motory jsou, z konstrukčního hlediska, tvořeny rotorem a statorem. Na statoru jsou umístěny permanentní magnety. Rotor je pak tvořen cívkami, které jsou ovládány řídicí jednotkou. Řídicí jednotka kromě řízení otáček také shromažďuje informace o motoru. Může tak například poskytovat ochranu proti přehřátí, nebo zjišťovat údaje o současné rychlosti a poloze rotoru.

1.1.1 Magnety

Nedílnou součástí BLDC motorů jsou permanentní magnety. Právě ony nahrazují vinutí rotoru. Zjednodušují tedy celkovou konstrukci motoru a redukují produkci tepla rotorovým vinutím. Pro výrobu motorů s permanentními magnety (dále jen PM) se užívá několik druhů magnetů. Lze je rozdělit na PM na bázi slitin, feritové (keramické) PM a PM ze vzácných zemin. Ty zde budou dále porovnávány především na základě maximálního energetického součinu $(BH)_{\max}$.

PM na bázi slitin jsou poměrně starým druhem magnetů. Zprvu se pro výrobu využívaly slitiny železa a uhlíku, nebo ocele a kobaltu s příměsí chrómu a wolframu. Tyto ocele získají po zakalení trvalé magnetické vlastnosti. Jejich maximální energetický součin $(BH)_{\max}$ však dosahuje vcelku malých hodnot ($1,6 \text{ kJ/m}^3$ a $7,5 \text{ kJ/m}^3$). Později začala být používána skupina slitin hliníku, niklu a kobaltu (AL-Ni-Co), která díky chladnutí v magnetickém poli dosahuje výrazně vyššího $(BH)_{\max}$. U Alnico 5 je to například až 40 kJ/m^3 . Mechanické vlastnosti Alnico slitin jsou však velmi špatné. Takovéto magnety jsou tvrdé a křehké, navíc během chladnutí výrazně zmenšují svůj tvar. Je tedy potřeba je po zchladnutí dotvarovat broušením.[14]

Feritové PM jsou vyráběny ze surového materiálu rozdrceného na krystaly. Takováto drť se pak zpracovává způsobem práškové metalurgie. Ta spočívá v lisování prášku a následně v jeho spékání. Jejich chemický vzorec lze popsat jako $MO \cdot 6(Fe_2O_3)$, kde M značí barium, stroncium nebo olovo. Oproti Alnico sloučeninám mají menší $(BH)_{\max}$ (až 30 kJ/m^3), ale lépe odolávají demagnetizačním polím.[14]

U PM ze vzácných zemin rozlišujeme dva nejpoužívanější druhy. Prvním je materiál tvořený samariem a kobaltem (Sm-Co). Tyto magnety dosahují $(BH)_{\max}$ až 160 kJ/m^3 . Jejich cena je však díky použití vzácných zemin poměrně vysoká.[14]

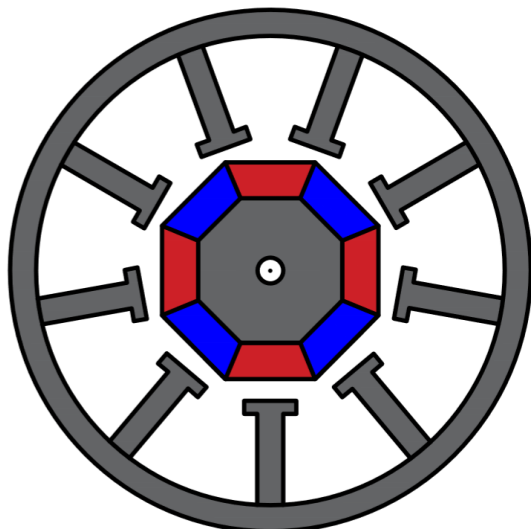
Druhý velmi rozšířený materiál je neodym-železo-bor (NdFeB). Ze všech zmíněných druhů magnetů je tento druhem nejmladším. Také dosahuje největšího energetického součinu $(BH)_{\max}$ (až 250 kJ/m^3). Díky těmto vlastnostem lze redukovat velikost magnetů a oproti Sm-Co jsou také menší náklady na výrobu. Nevýhodou je však změna vlastností magnetu s vysokými teplotami, kdy při práci s teplotami okolo $100 \text{ }^\circ\text{C}$ se musí konstruovat se zvýšenou pečlivostí.[14]

1.1.2 Rotor

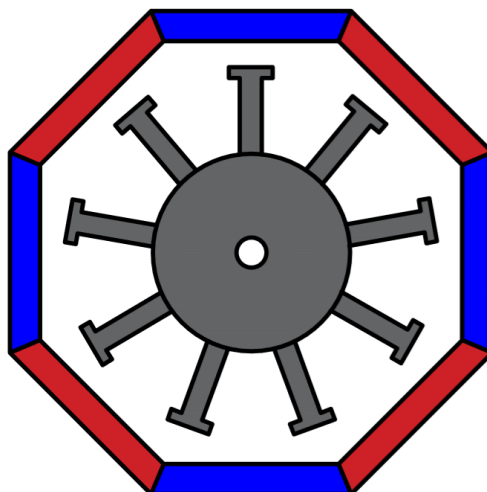
Hlavní součástí rotoru jsou permanentní magnety, které již byly podrobněji probrány v textu výše. Magnety jsou rozmístěny po obvodu rotoru a tvoří tak spolu s cívkami statoru hybnou sílu motoru. Magnetů je zpravidla sudý počet a na rotoru jsou umístěny se střídavou polaritou. Dvojice magnetů s opačnou polaritou ležící v jedné ose se nazývá pólový pár. Typicky bývá v motoru 2 až 8 pólových párů. Počet pólových párů ovlivňuje vlastnosti motorů a je tedy jedním z klíčových parametrů při návrhu motoru. Více pólových párů předznamenává motor k většímu momentu, ale také snižuje maximální možnou dosažitelnou rychlost.

Podle toho, jakým způsobem je rotor zkonstruován rozlišujeme dva druhy motorů. Prvním je motor s vnějším rotorem (outer rotor/ out-runner) a druhým pak motor s vnitřním rotorem (inner rotor/ in-runner). Zatímco vnější rotor obklopuje cívky statoru, vnitřní rotor je těmito cívkami obklopován. To má značný vliv na odvod tepla, vzniklého průchodem proudu cívkami do okolí. Vnější rotor tak působí jako jakási izolace a teplo se v motoru drží mnohem déle. Toto konstrukční řešení je tedy používáno zejména v případě motorů s nízkými otáčkami nebo s nízkým pracovním proudem. Pro výkonnější motory je tak vhodné použít konstrukci s vnitřním rotorem, čímž se značně zefektivňuje chlazení. Konstrukční rozdíl je patrný z Obrázku 1.

vnitřní (inner) rotor



vnější (outer) rotor



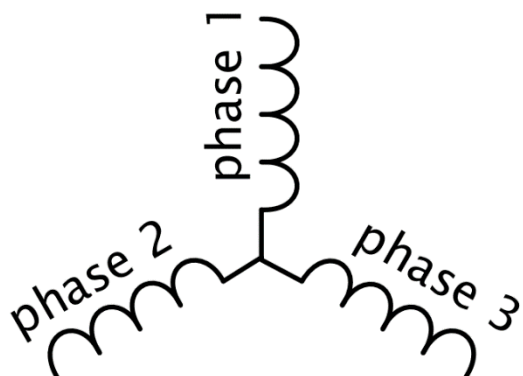
Obrázek 1 - Konstrukce rotoru

1.1.3 Stator

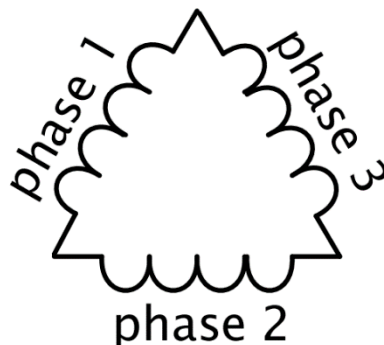
Stator je tvořen cívkami. Je podobný jako stator indukčního motoru s odlišně uspořádaným vinutím. Vinutí je uspořádáno tak, aby cívky tvořily sudý počet pólů korespondující s pólovými páry rotoru. Stator může být navinut dvěma způsoby, které ovlivňují vlastnosti motoru. Prvním je vinutí do trojúhelníku (Δ) a druhé do hvězdy (star/Y). První druh vinutí, tedy do trojúhelníku, předurčuje motor k malému momentu a velkým otáčkám. Druhý druh vinutí, do hvězdy, v podstatě motoru předurčuje opačné vlastnosti. Motory tak poskytují větší moment a menší otáčky. Jednotlivá vinutí lze pak zjednodušit na schémata zobrazena na Obrázku 2.

Samotné cívky jsou nejčastěji navinuty na ocelové kostře. Existují však také konstrukce motoru, která jádro postrádá a stator je tvořen pouze vodiči. Provedení s kostrou se označuje jako slotted a bez kostry jako slotless. Vinutí bez kostry se vyznačuje nízkou indukčností a nízkým kroutícím momentem. Tyto motory pracují dobře při vysokých i malých rychlostech. Jelikož kostra v tomto případě není oporou, k udržení tvaru vinutí se užívá epoxidové pryskyřice. Nevýhodou jsou vyšší náklady na výrobu, neboť vzduchové mezery se musí vyplnit vynutím a také samotné navíjení je komplikovanější. Motor se statorem navinutým na kostře nedosahuje oproti motoru bez kostry tak dobrých vlastností. Pro většinu aplikací je však dostatečný a jeho cena je příznivější. Konstrukční rozdíl je patrný z Obrázku 3

hvězda

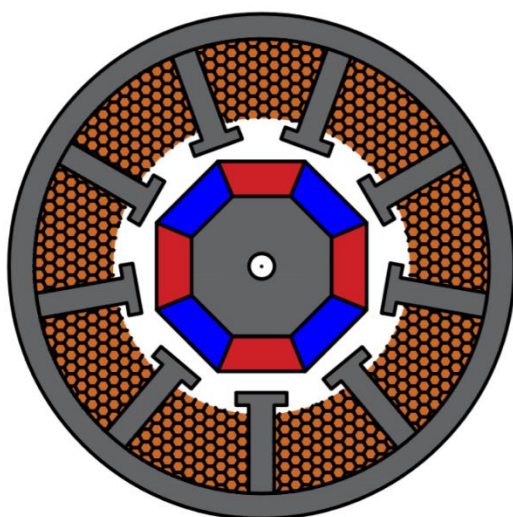


trojúhelník

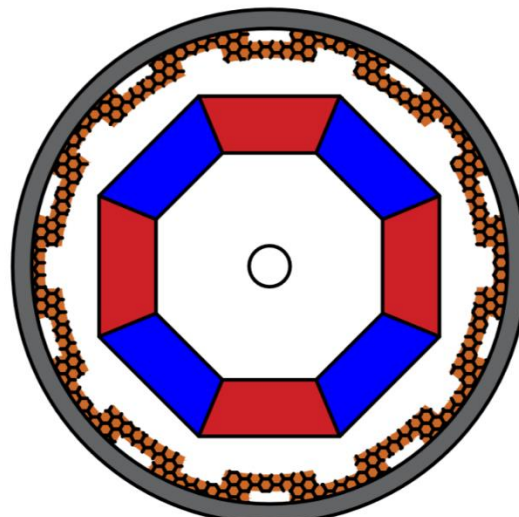


Obrázek 2 - Náhradní schémata vinutí

s kostrou (slotted)



bez kostry (slottless)



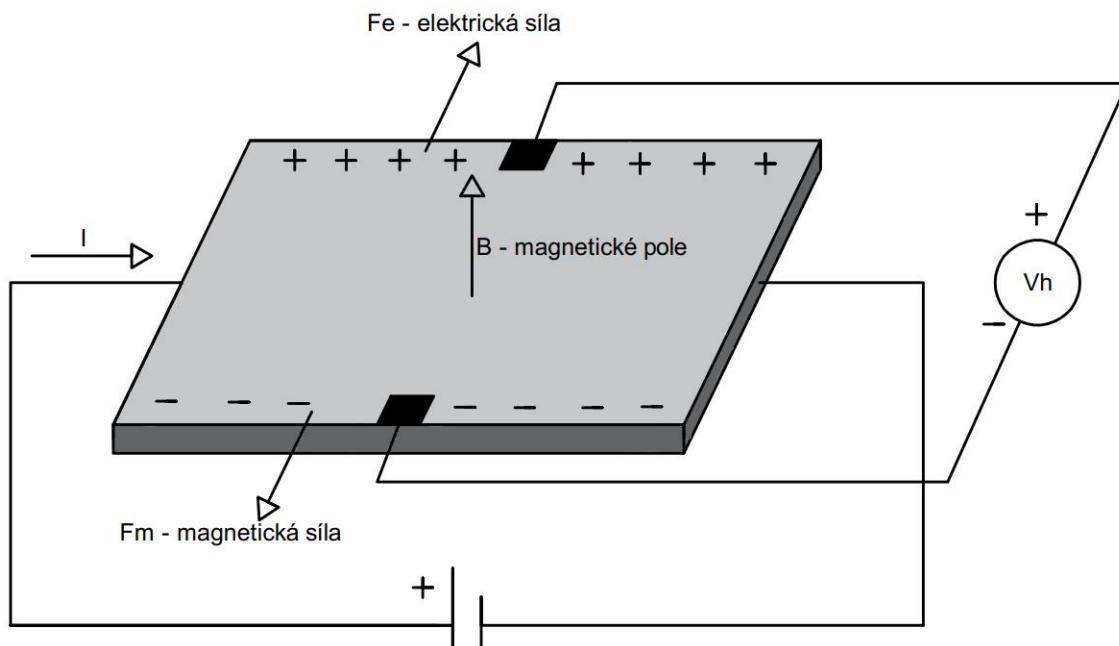
Obrázek 3 - Konstrukce statoru

1.1.4 Hallové senzory

Hallové senzory, neboli polohové senzory jsou komponentou, kterou motor pro svou funkci nutně nepotřebuje. Jejich použití nám však poskytuje datový výstup, který udává aktuální polohu rotoru. Často jsou tedy užívány jako součást řídicího systému. Ve složitějších aplikacích jsou takřka nezbytné. Zajišťují totiž plynulejší komutaci.

Hallové senzory využívají tzv. Hallova jevu. Jedná se v podstatě o polovodičovou destičku, která je umístěna v magnetickém poli cívky. Tato destička je protékána proudem. Změna magnetického pole se projevuje vychylováním směru toku proudu procházejícího destičkou.

Výsledkem tak bude změna napětí mezi jednotlivými konci destičky (v Obrázku 4 značeno jako V_h), která je úměrná velikosti působícího magnetického pole. Grafické zobrazení zapojení Hallových sond je patrné z Obrázku 4.



Obrázek 4 – Hallova sonda

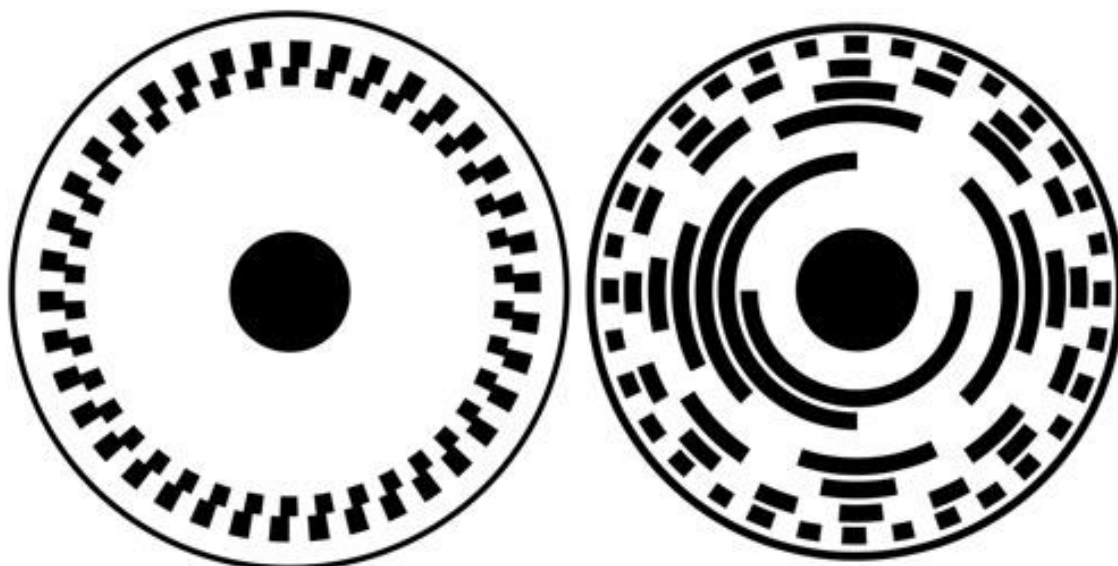
1.1.5 Rotační enkodéry

Rotační enkodéry, stejně jako Hallovy sondy nejsou nezbytnou komponentou motoru. Jsou však jedním z možných řešení senzorského snímání pohybu motoru. V závislosti na svém provedení poskytují poměrně přesné informace o aktuální poloze rotoru, které lze dále vyhodnocovat. Enkodéry lze primárně dělit na absolutní a inkrementální.

Absolutní enkodéry nám poskytují konkrétní hodnotu o úhlu natočení. Jejich výstupem je určitá binární hodnota odpovídající úhlu natočení. Jejich rozlišení závisí na přesnosti a technologii výroby. Díky své komplikované konstrukci jsou zpravidla náročnější na výrobu. Tomu odpovídá i vyšší cena. Poskytují však hodnoty, které jsou jednodušší na zpracování a nevyžadují složitější logiku programu.

Další varianta, tedy inkrementální enkodér, se zpravidla skládá ze dvou různých výstupů. Výstupy při rotaci nabývají hodnot logické 1, nebo 0. V závislosti na sledu změn jednotlivých výstupů lze pak určit směr otáčení, případně i konkrétní polohu. Tu však lze určit pouze za předpokladu, že známe polohu výchozí. Tu pak při změně výstupu zvyšujeme/ snižujeme, což odpovídá poloze aktuální. Lze je také vhodně využít k snímání rychlosti otáčení, kdy lze sledovat a vyhodnocovat rychlost změny výstupů.

Po konstrukční stránce jsou enkodéry typicky tvořeny jakýmsi terčíkem, který na sobě má vodivé segmenty. Tyto segmenty se při rotaci dostávají do kontaktu s výstupními konektory a na výstupu jsou tak generovány proměnné hodnoty, odpovídající rotaci motoru. Typické příklady těchto terčů jsou zobrazeny na Obrázku 5. Vlevo je inkrementální enkodér a vpravo absolutní.



Obrázek 5 - Rotační enkodéry

Enkodéry jsou dostupné v mnoha různých provedeních. Zejména u absolutního enkodéru je klíčovým parametrem počet bitů na výstupu. Od něho se pak odvíjí rozlišení enkodéru. Existují také různé druhy vyhodnocování rotace. Jednou z možností je vyhodnocování pomocí elektrického kontaktu. Další variantou je snímání světla procházejícího terčíkem, nebo světla, které se od terčíku odrazí. Mluvíme pak o tzv. optických enkodérech. Tato varianta je zpravidla přesnější. Komponenta v tomto provedení také dosahuje vyšší životnosti díky eliminaci otěru kontaktů. Z toho však také plyne vyšší cena součástky.

1.2 Srovnání BLDC a jiných typů motorů

Jak již bylo v práci několikrát zmíněno, BLDC motory se svým principem funkčnosti poněkud vymykají starším typům motorů. Je tedy vhodné provést jejich srovnání. Pro názornost je srovnání nejdůležitějších parametrů uvedeno v následujících tabulkách. Srovnání je provedeno jak pro kartáčové stejnosměrné motory v Tabulce 2, tak pro střídavé asynchronní motory v Tabulce 1. BLDC motor totiž vykazuje jistou příbuznost s oběma druhy. Je sice napájen stejnosměrným proudem, avšak průběh komutace značně připomíná komutaci motorů asynchronních.

Tabulka 1 - Srovnání BLDC a AC motoru

parametry	BLDC motor	asynchronní AC motor
komutace	Komutace je řízená na základě senzorů, nebo zpětné elektromotorické síly.	Komutace je zajištěna elektricky, charakteristikou napájecího zdroje.
momentová charakteristika	Charakteristika je lineární. Dovoluje motoru pracovat s danou zátěží na všech rychlostech.	Charakteristika je nelineární. Při nižších rychlostech tak klesá moment.
setrvačnost rotace	Setrvačnost je nízká, kvůli použití permanentních magnetů.	Rotační setrvačnost je vyšší.
řízení	Řízení je složité, vyžaduje driver a řídicí logiku.	Řízení je jednodušší, pro konstantní rychlost vyžaduje pouze vhodný napájecí zdroj.
skluz	Mezi rotorem a státorem není žádný skluz. Motor pracuje synchronně.	Rotor se točí s frekvencí nižší než frekvence elektromagnetického pole. Tento rozdíl je definován skluzem.
záběrný proud	Při rozběhu se proud nemění.	Při rozběhu je vyžadován vysoký záběrný proud, dosahující několikanásobné hodnoty proudu jmenovitého.
poměr váha/výkon	Vyšší poměr daný použitím magnetů na rotoru.	Nižší poměr daný přítomností statorového i rotorového vinutí.

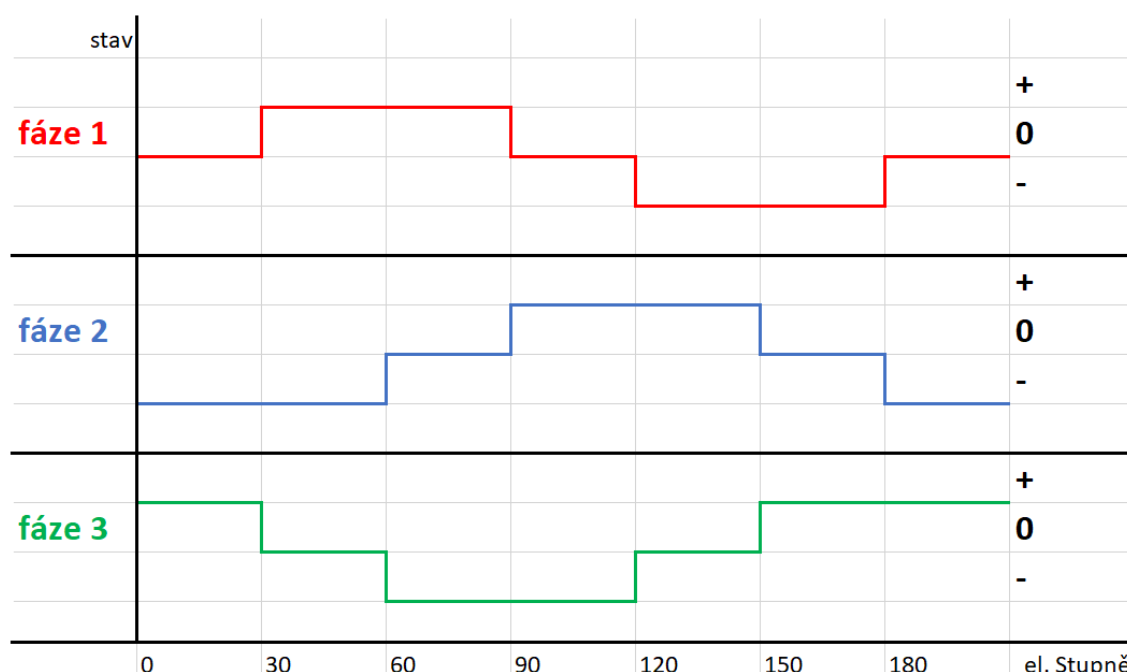
Tabulka 2 - Srovnání BLDC a kartáčového DC motoru

parametry	BLDC motor	Kartáčový DC motor
komutace	Komutace je elektronicky řízená na základě senzorů, nebo zpětné elektromotorické síly.	Komutace je zajištěna mechanicky komutátorem a kartáči.
účinnost	Účinnost je vyšší než u kartáčových motorů.	Účinnost je nižší než u BLDC motorů zapříčiněná ztrátami při komutaci.
životnost	Životnost je delší.	Životnost je kratší kvůli více namáhaným komponentám jako komutátor a kartáče.
údržba	BLDC motory jsou méně náročné na údržbu	Je nutná pravidelná kontrola kartáčů a případná výměna.
momentová charakteristika	Charakteristika je lineární. Dovoluje motoru pracovat s danou zátěží na všech rychlostech.	Na vyšších rychlostech omezuje přítomnost kartáčů dodávaný moment.
řízení	Řízení je složité a vyžaduje driver a řídicí logiku.	Řízení je jednoduché, pro konstantní rychlost vyžaduje pouze vhodný napájecí zdroj.
odvod tepla	Odvod tepla je výrazně lepší, protože vinutí je pouze na statoru.	Odvod tepla je horší, protože teplo je nutné odvádět z vinutí rotoru i statoru.
setrvačnost rotace	Setrvačnost je nízká, kvůli použití permanentních magnetů.	Rotační setrvačnost je vyšší.
elektrický šum	Elektrický šum je velmi malý.	Kartáče při komutaci generují elektrický šum.
poměr váha/výkon	Vyšší poměr daný použitím magnetů na rotoru.	Nižší poměr daný přítomností vinutí statorového i rotorového.
cena	Vyšší cena daná technologií výroby.	Cena je nižší.

2 TEORIE ŘÍZENÍ BLDC MOTORŮ

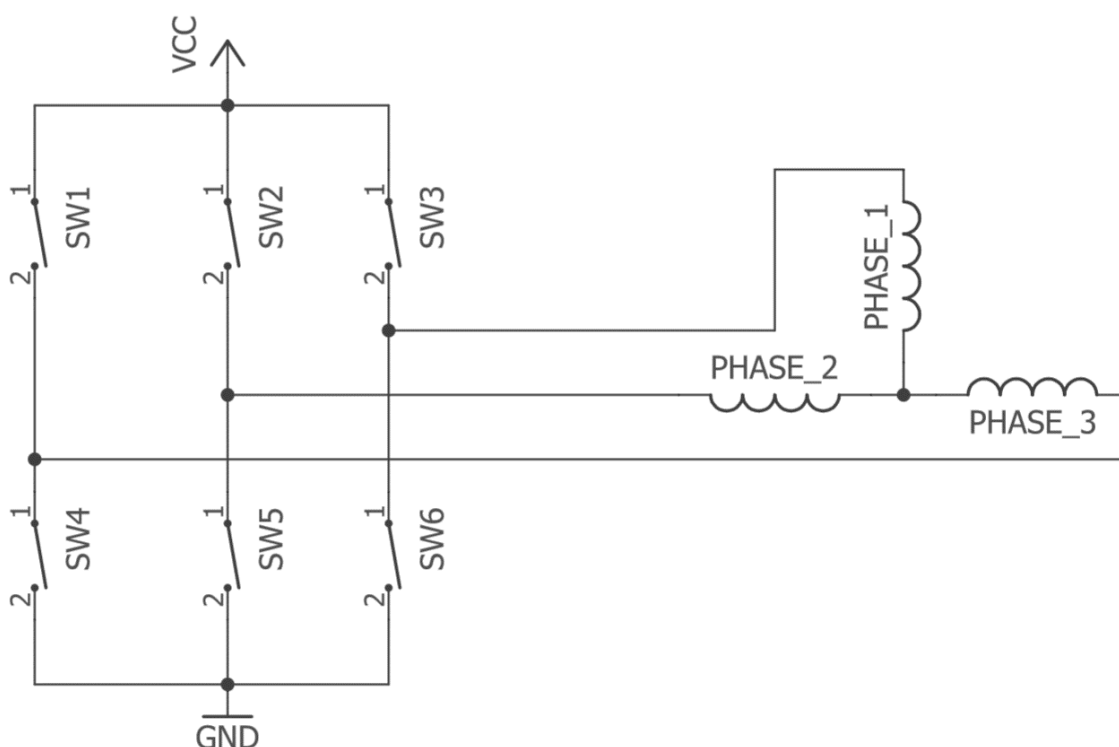
2.1 Komutace

Pro běh BLDC motorů je klíčové řízení směru proudu, nazývané též komutace. U BLDC motorů je komutace prováděna elektronicky, tedy kontrolérem. Kontrolér v požadovaném časovém sledu spíná jednotlivé cívky s určitou polaritou. Vytváří tak točivé magnetické pole, které roztáčí rotor. Pro optimální chod motoru je nutné, aby vektor magnetického pole statoru byl vždy o 90 stupňů posunut oproti vektoru magnetického pole rotoru. Celkový vektor, udávající velikost a směr magnetického pole statoru, je dán součtem vektorů magnetických polí jednotlivých fází. Pro úspěšné řízení celého motoru je tedy nezbytné ovládat směr a velikost proudu jednotlivých fází. V ideálním případě každá fáze vede po dobu 120 stupňů elektrických. Na Obrázku 6 je znázorněno ideální přepínání jednotlivých fází motoru.



Obrázek 6 - Komutace

Kontrolér, který zajišťuje správný průběh komutace, si můžeme pro zjednodušení představit jako soubor spínačů, které připojují v požadovaném sledu dané cívky. Tento zjednodušený návrh je zobrazen na Obrázek 7. Podle tohoto schématu a grafu přepínání fází můžeme určit časový sled sepnutí jednotlivých spínačů Q. Ten bude následující: SW3 SW 5, SW 3 SW 4, SW 2 SW 4, SW 2 SW 6, SW 1 SW 6, SW 1 SW 5.



Obrázek 7 – Zjednodušené řízení komutace

Zmíněnému sledu spínání odpovídá Tabulka 3 Ta zobrazuje pro jednotlivé stavy aktuální polaritu cívek.

Tabulka 3 - Spínání komutace

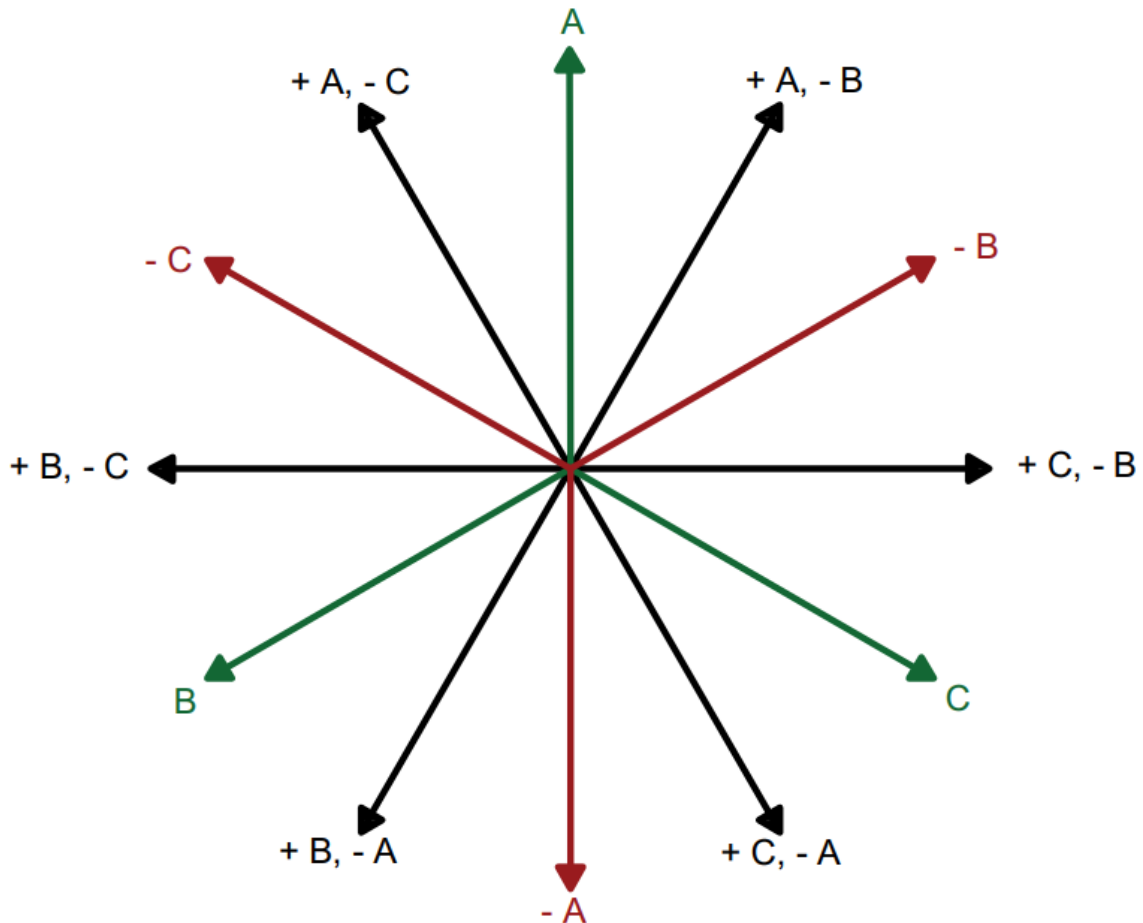
spínače						cívky		
SW1	SW2	SW3	SW4	SW5	SW6	PHASE_1	PHASE_2	PHASE_3
OFF	OFF	ON	OFF	ON	OFF	+	-	float
OFF	OFF	ON	ON	OFF	OFF	+	float	-
OFF	ON	OFF	ON	OFF	OFF	float	+	-
OFF	ON	OFF	OFF	OFF	ON	-	+	float
ON	OFF	OFF	OFF	OFF	ON	-	float	+
ON	OFF	OFF	OFF	ON	OFF	float	-	+

V Tabulce 3 si můžeme všimnout, že cívky, které zrovna nejsou připojeny na kladnou, nebo zápornou polaritu, jsou označeny jako tzv. float. Cívky jsou v tomto stavu odpojeny od napájení a lze je vhodně použít k bezsensorovému řízení motoru. Lze na nich totiž měřit zpětné elektromotorické napětí.

Při přepínání cívek podle Tabulky 3 budou tyto cívky protékány proudem. Tento proud vytvoří ve vinutí statoru magnetomotorické napětí (mmn), jehož poloha je dána vektorovým součtem mmn právě protékáných vinutí.[15]

Vektor mmn bude rotovat v kruhu. Jeho pohyb zobrazuje Obrázek 8. Vektory jednotlivých stavů jsou označeny černě a tvoří je vždy součet vektorů cívek, které jsou právě připojeny. Vektory připojených cívek jsou zobrazeny barevně. Zeleně jsou označeny vektory pouze pro kladnou polaritu a červeně pouze pro polaritu zápornou.

Pro posloupnost, zobrazenou v Tabulce 3, bude vektor magnetického pole rotovat proti směru hodinových ručiček, přičemž tabulkovému označení PHASE_1, PHASE_2 a PHASE_3 odpovídá značení A, B a C. Pro opačný směr otáčení by bylo nutné spínat jednotlivé cívky v opačném pořadí.



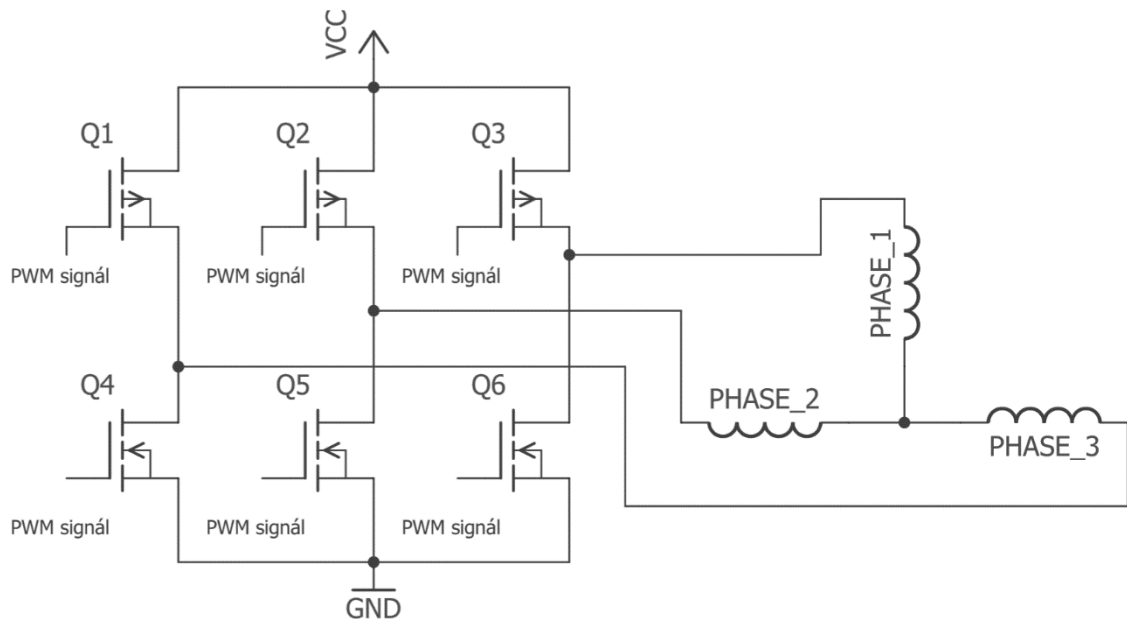
Obrázek 8 – Vektor magnetického pole v průběhu komutace

Můžeme si všimnout, že je v Tabulce 3 pro třífázové vinutí motoru zobrazeno šest kombinací sepnutí. Stejně tak v Obrázku 8 pozorujeme šest poloh vektorů odpovídající stavům sepnutí. Od tohoto počtu se odvíjí označení této komutace, které se říká six-step komutace, tedy komutace obsahující šest kroků.

2.2 Driver

Přepínání fází, znázorněné v části o komutaci, je ve skutečnosti zajišťováno součástkou, které se říká driver. Princip jeho funkce byl již popsán, nyní se však zaměříme na to, čím je tvořen.

Z pravidla se jedná o polovodičový můstek. Ten je typicky tvořen mosfet tranzistory. Ty jsou vhodné pro komutaci motorů nízkých a středních výkonů. Pro motory s nízkými výkony bývají tyto tranzistory integrovány v čipu. Takovýto čip je použit i v této práci a bude zde podrobněji probrán. V případě výkonnějších motorů je nutné zajistit efektivnější odvod tepla. Driver je tak nutno konstruovat z jednotlivých tranzistorů. V případě motorů s velkým výkonem se pak často volí IGBT tranzistory. Druh a parametry použitých tranzistorů se odvíjejí od parametrů motoru, který chceme řídit. Klíčovou je velikost proudu, který přes tranzistory bude protékat. Průtokem proudu vzniká teplo a při nevhodně navrženém chlazení by mohlo dojít k poškození součástky, nebo čipu, ve kterém je integrována. Jejich počet pak závisí na počtu fází daného motoru. Typické zapojení tranzistorů třífázového invertoru je patrné z Obrázku 9.



Obrázek 9 - Zapojení polovodičových součástek řídících komutaci

Takto zkonstruovaný polovodičový můstek můžeme pak nazvat střídačem. Jednotlivé tranzistory jsou spínány, jak je znázorněno v Tabulce 3 a na výstupu je napětí, jehož průběh má střídavý charakter.

Spínání tranzistorů bývá zajišťováno procesorem, nebo mikrokontrolérem. Konkrétně jsou tranzistory spínány signálem pulsně šířkové modulace (PWM) z dané logické jednotky.

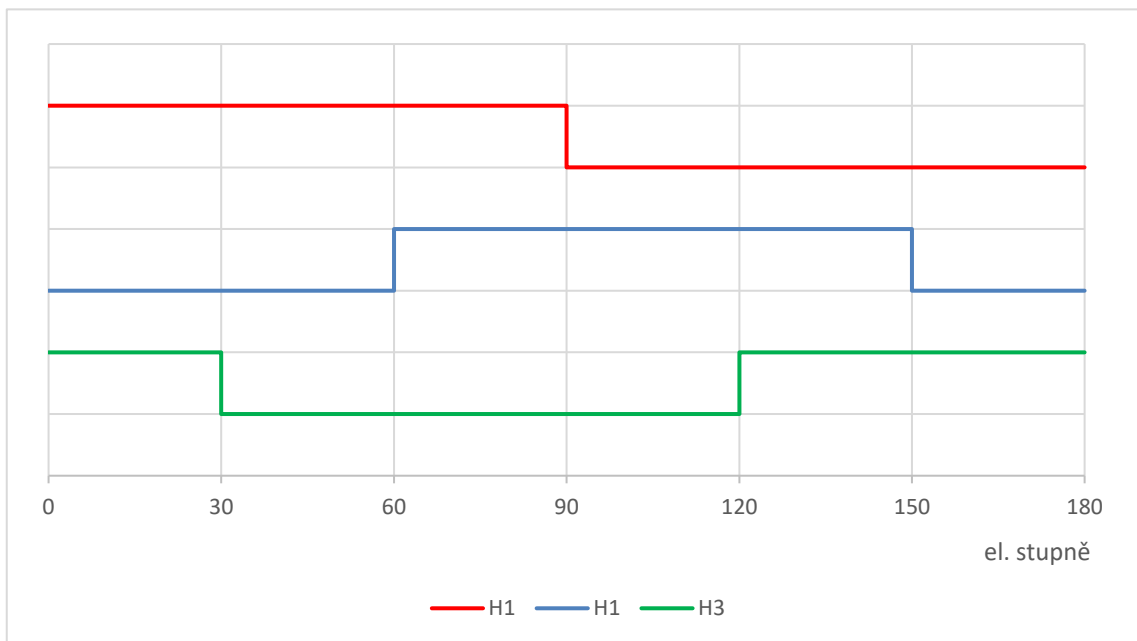
2.3 Vyhodnocování polohy rotoru

Jak již bylo v této práci zmiňováno, optimální chod motoru je zajištěn, je-li vektor magnetického pole statoru o 90 stupňů před vektorem magnetického pole rotoru. Právě z tohoto důvodu je nezbytné znát konkrétní polohu rotoru. Polohu rotoru lze stanovit dvěma způsoby, a to pomocí senzorů, anebo bez nich.

2.3.1 Senzorické snímání polohy

V případě snímání polohy senzory máme dvě možnosti. Můžeme použít enkodér, což je metoda velmi přesná. Poskytuje nám údaje o poloze, úhlu natočení a otáčkách motoru. Enkodér je však poměrně drahá součástka. Z toho důvodu je častěji užíváno Hallových sond.

Hallové sondy se instalují nejčastěji přímo do motoru a jsou rozmístěny po 60 nebo 120 stupních elektrických. Když se do blízkosti senzorů dostanou permanentní magnety, senzory poskytnou výstup, kterým je logická 0 nebo 1. To umožňuje vcelku přesně stanovit aktuální stav motoru. Typický příklad výstupu halových sond je patrný z Obrázku 10.



Obrázek 10 - Výstup Hallových sond

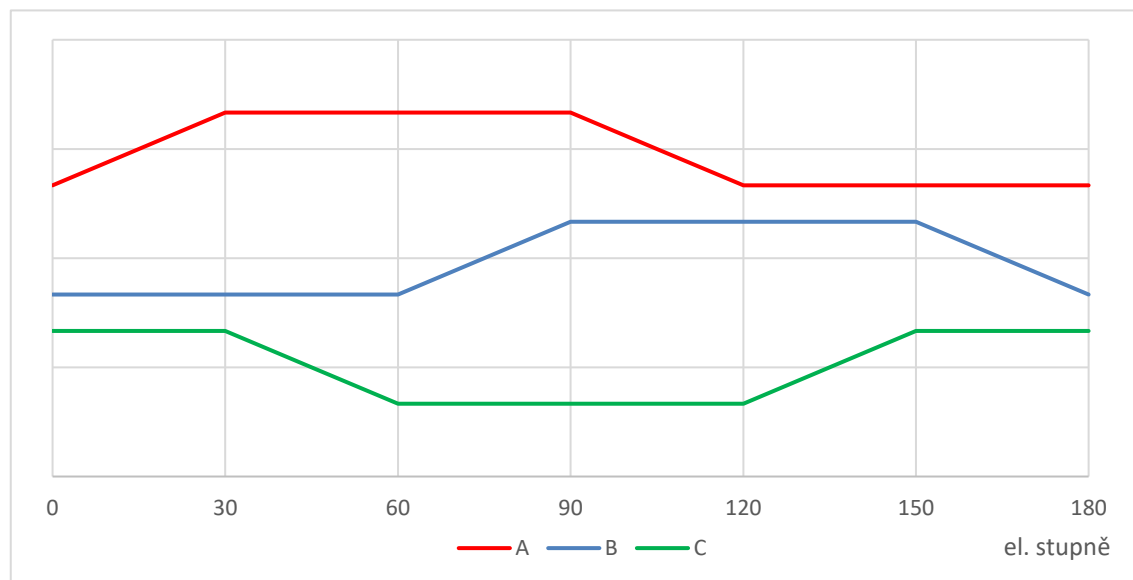
Kýžená six-step komutace probíhá na základě vyhodnocení těchto průběhů. Z Obrázku 10 můžeme odvodit, že k novému přepnutí motoru dojde vždy, při změně některého z výstupů. Tabulka 4 ukazuje jednotlivé kroky komutace odpovídající výstupům Hallových sond.

Tabulka 4 - Výstupy Hallových sond

H1	H2	H3	Krok komutace
1	0	1	0
1	0	0	1
1	1	0	2
0	1	0	3
0	1	1	4
0	0	1	5

2.3.2 Bez-senzorové určování polohy

V případě stanovení polohy bez senzorů se spoléháme na snímání zpětného elektromotorického napětí neboli BEMF. Při chodu motoru jsou vždy připojeny pouze dvě fáze. To nám dává možnost snímat indukované napětí na fázi třetí. Lze pak detekovat průchod BEMF nulou, tedy takzvaný „zero crossing“. Právě čas, při kterém BEMF prochází nulou je vhodný pro přepnutí do dalšího stavu six-step komutace. Snímaný průběh indukovaného napětí má typicky podobu jako na Obrázku 11.



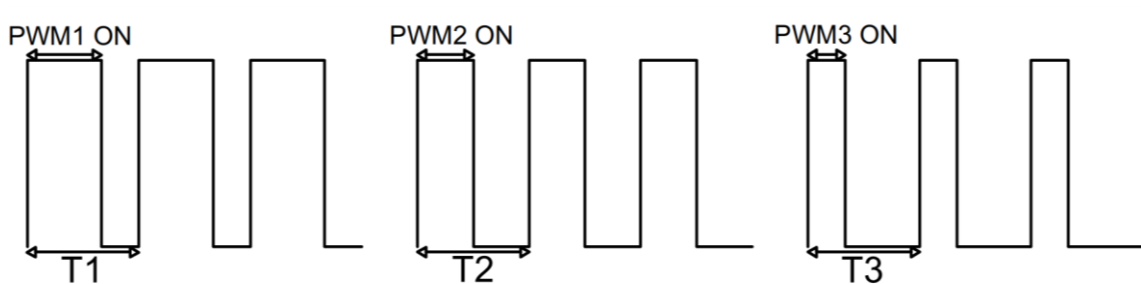
Obrázek 11 - Detekce průchodu BEMF nulou

Snímání vhodného průběhu BEMF je však možné až od určitých otáček. Tyto otáčky jsou dány mechanickými vlastnostmi motoru a pro různé motory se tak liší. Motor je tedy zprvu nutné na tyto otáčky rozeběhnout asynchronně pouhým zrychlujícím se přepínáním cívek bez možnosti snímání charakteristického průběhu BEMF. Tento rozběh bývá zpravidla velmi rychlý a zajišťuje synchronizaci rotoru s magnetickým polem statoru při bezsenzorovém chodu.

Další možností bezsenzorového řízení je stanovení polohy motoru na základě znalosti jeho matematického modelu. Zde potřebujeme znát fázová napětí a proudy. Jejich integrací lze určit magnetický tok a pomocí něho stanovit polohu motoru a vhodnou dobu komutace.

2.4 Řízení rychlosti

Rychlost BLDC motoru lze ovlivnit velikostí střední hodnoty proudu, který přivedeme na jednotlivé fáze. Jednotlivé tranzistory v driveru jsou řízeny logickou jednotkou. Výstupem logické jednotky je signál pulsně šířkové modulace (PWM). Rotor synchronního motoru se otáčí rychlostí, která je přímo úměrná napájecí frekvenci.[15] Běžně se zvolí perioda PWM signálu a mění se poté šířka pulsu. Tři takové průběhy jsou zobrazeny na Obrázku 12. Jedná se o PWM signály se stejnou periodou, ale různými šířkami pulsů.

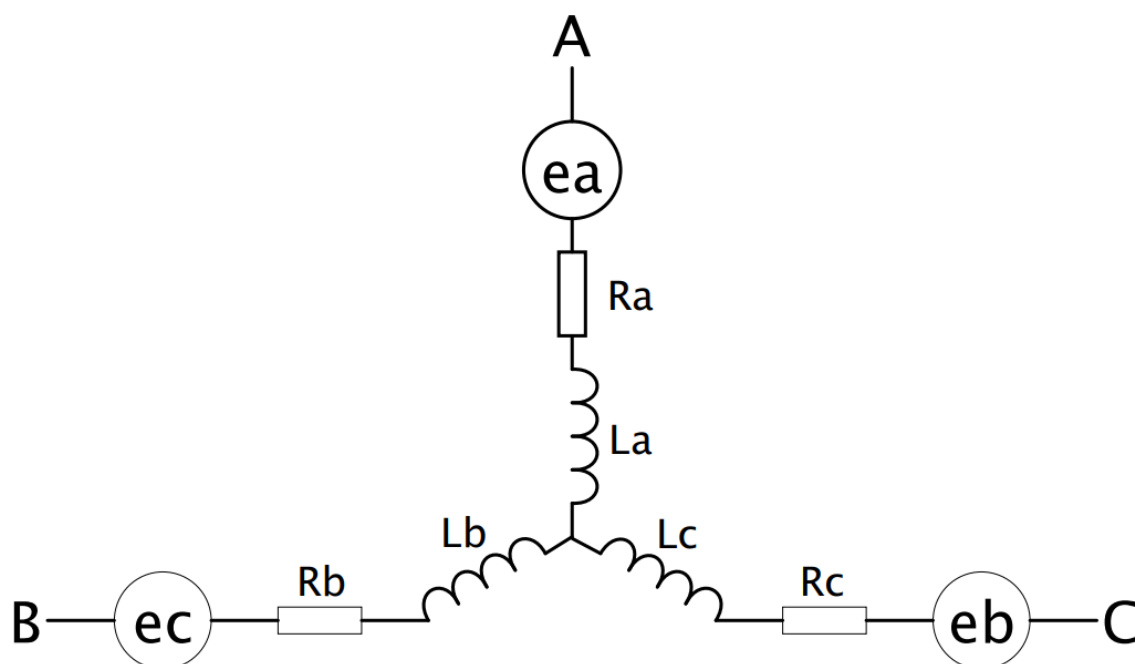


Obrázek 12 – Typické průběhy PWM signálu

Zvýšení střední hodnoty proudu má za následek zvýšení intenzity magnetického pole. Tím pádem jsou v jednotlivých krocích komutace pólové páry rotoru přitahovány k odpovídajícím cívkám rychleji a motor tak zvyšuje svoji rychlost.

3 MATEMATICKÝ MODEL BLDC MOTORU

Pro přesnější popis motorů slouží tzv. matematický model. Je také klíčovým k sestavení simulace motoru v různých simulačních programech a pochopení jeho vnitřních elektromagnetických mechanismů. Pro popis BLDC motoru v této bakalářské práci použijeme náhradní schéma motoru, zobrazené na Obrázku 13.



Obrázek 13 - Náhradní schéma motoru pro matematický model

Můžeme na něm pozorovat motor s vinutím do hvězdy. Rovnice matematického modelu tedy budou platit pro tento druh vinutí.

Pro vinutí na Obrázku 13 můžeme vyjádřit obvodové rovnice ve tvaru:

$$\begin{aligned} u_a &= R_a * i_a + \frac{d(L_a * i_a + L_{ab} * i_b + L_{ac} * i_c)}{dt} + e_a \\ &= R_a * i_a + \psi_a + e_a \end{aligned} \quad (1)$$

$$\begin{aligned} u_b &= R_b * i_b + \frac{d(L_b * i_b + L_{ab} * i_a + L_{bc} * i_c)}{dt} + e_b \\ &= R_b * i_b + \psi_b + e_b \end{aligned} \quad (2)$$

$$\begin{aligned} u_c &= R_c * i_c + \frac{d(L_c * i_c + L_{ac} * i_a + L_{bc} * i_b)}{dt} + e_c \\ &= R_c * i_c + \psi_c + e_c \end{aligned} \quad (3)$$

Jednotlivé veličiny mají následující význam:

u_a, u_b, u_c – fázová napětí [V]

i_a, i_b, i_c – fázové proudy [A]

R_a, R_b, R_c – odpory jednotlivých vinutí [Ω]

L_a, L_b, L_c – vlastní indukčnosti [H]

L_{ab}, L_{ac}, L_{bc} – vzájemné indukčnosti [H]

e_a, e_b, e_c – elektromotorická napětí [V]

ψ_a, ψ_b, ψ_c – spřažené magnetické toky [Wb]

V případě ideálního symetrického BLDC motoru můžeme dále napsat, že:

$$L_a = L_b = L_c = L \quad (4)$$

$$L_{ab} = L_{ac} = L_{bc} = M \quad (5)$$

$$R_a = R_b = R_c = R \quad (6)$$

Z tohoto poznatku plyne, že rovnice (1), (2) a (3) lze poté zjednodušit na tvar:

$$u_a = R_a * i_a + \frac{(L - M) * di_a}{dt} + e_a \quad (7)$$

$$u_b = R_b * i_b + \frac{(L - M) * di_b}{dt} + e_b \quad (8)$$

$$u_c = R_c * i_c + \frac{(L - M) * di_c}{dt} + e_c \quad (9)$$

V předchozích výpočtech jsme do rovnic zahrnovaly elektromotorická napětí e . Tato napětí jsou dána, konstrukčními vlastnostmi motoru. Pro typické trapézové vinutí motoru platí podle zdroje [16] rovnice:

$$e_a = K_e * f(\theta_e) * \omega_m \quad (10)$$

$$e_b = K_e * f\left(\theta_e - \frac{2\pi}{3}\right) * \omega_m \quad (11)$$

$$e_c = K_e * f\left(\theta_e + \frac{2\pi}{3}\right) * \omega_m \quad (12)$$

Jednotlivé veličiny mají následující význam:

ω_m – mechanická úhlová rychlost rotoru [$rad * s^{-1}$]

K_e – konstanta zpětné elektromotorické síly [$\frac{V}{rad * s^{-1}}$]

$f(\theta_e)$ – trapézová funkce motoru

θ_e – elektrické stupně rotoru [rad]

Při svém chodu je motor poháněn elektromagnetickým hnacím momentem. Proti momentu hnacímu působí moment zátěžný, který vyjadřuje moment, jímž na hřídel působí pracovní mechanismus. Rozdíl těchto dvou momentů je roven momentu dynamickému, který vyjadřuje působení sil proti změně rovnovážného stavu motoru.[15] Vztahy mezi momenty vyjadřuje pohybová rovnice (13):

$$M - M_z = M_d \quad (13)$$

Jednotlivé veličiny mají následující význam:

M – hnací moment [Nm]

M_z – zátěžný moment [Nm]

M_d – dynamický moment [Nm]

Moment hnací a dynamický lze podle zdroje [16] vyjádřit rovnicemi (14) a (15).

$$M = \frac{e_a * i_a + e_b * i_b + e_c * i_c}{\omega_m} \quad (14)$$

$$M_d = J * \frac{d\omega_m}{dt} \quad (15)$$

Jednotlivé veličiny mají následující význam:

ω_m – mechanická úhlová rychlost rotoru [$rad * s^{-1}$]

J – moment setrvačnosti [$kg * m^2$]

i_a, i_b, i_c – fázové proudy [A]

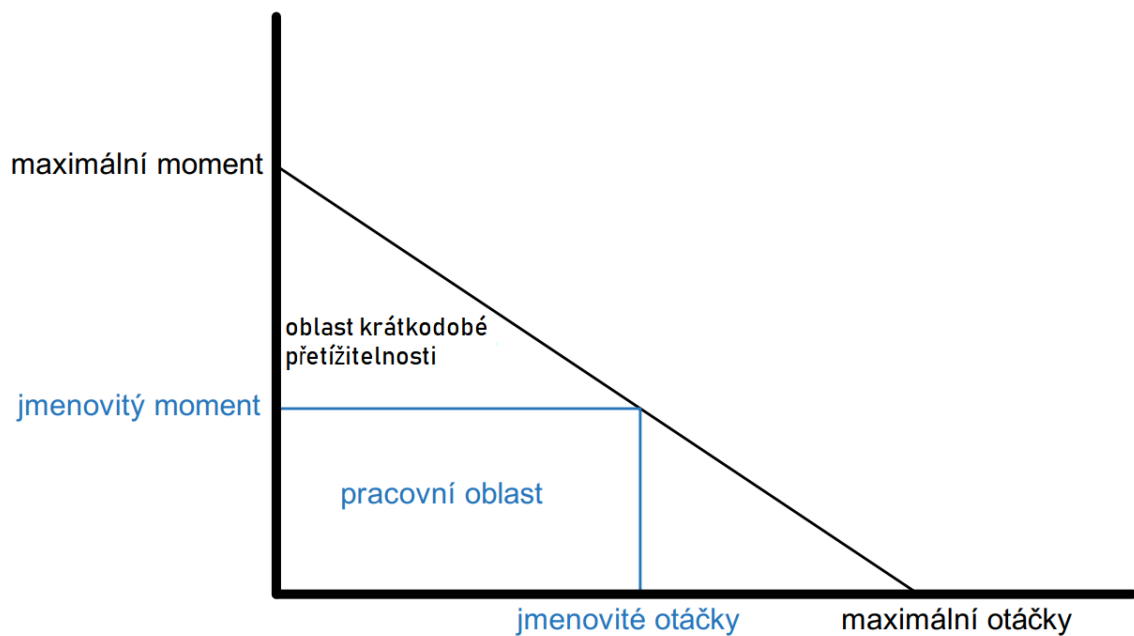
e_a, e_b, e_c – elektromotorická napětí [V]

Vztah mezi elektrickou úhlovou rychlostí ω_{el} [$rad * s^{-1}$] a mechanickou úhlovou rychlostí ω_m lze vyjádřit jako:

$$\omega_{el} = \frac{\omega_m * p}{2} \quad (16)$$

Přičemž p značí počet pólových párů.

Závislost momentu na otáčkách má přibližně lineární charakter. Její průběh lze sledovat na Obrázku 14. Z obrázku je patrné, že s rostoucím momentem otáčky klesají lineárně, a to až do nulové hodnoty. Pracovní oblast se nachází v oblasti od jmenovitých otáček k nule. Motor lze typicky krátkodobě přetěžovat, pro zvýšení momentu. Tento stav však motor nadměrně zatěžuje, nelze ho tedy takto trvale provozovat. Možná délka provozu s krátkodobě zvýšeným momentem se odvíjí od konstrukce motoru. V aplikacích vyžadujících dlouhodobější trvání tohoto stavu je tak nutné k tomuto faktu při návrhu motoru přihlídnout.



Obrázek 14 - Závislost momentu na otáčkách

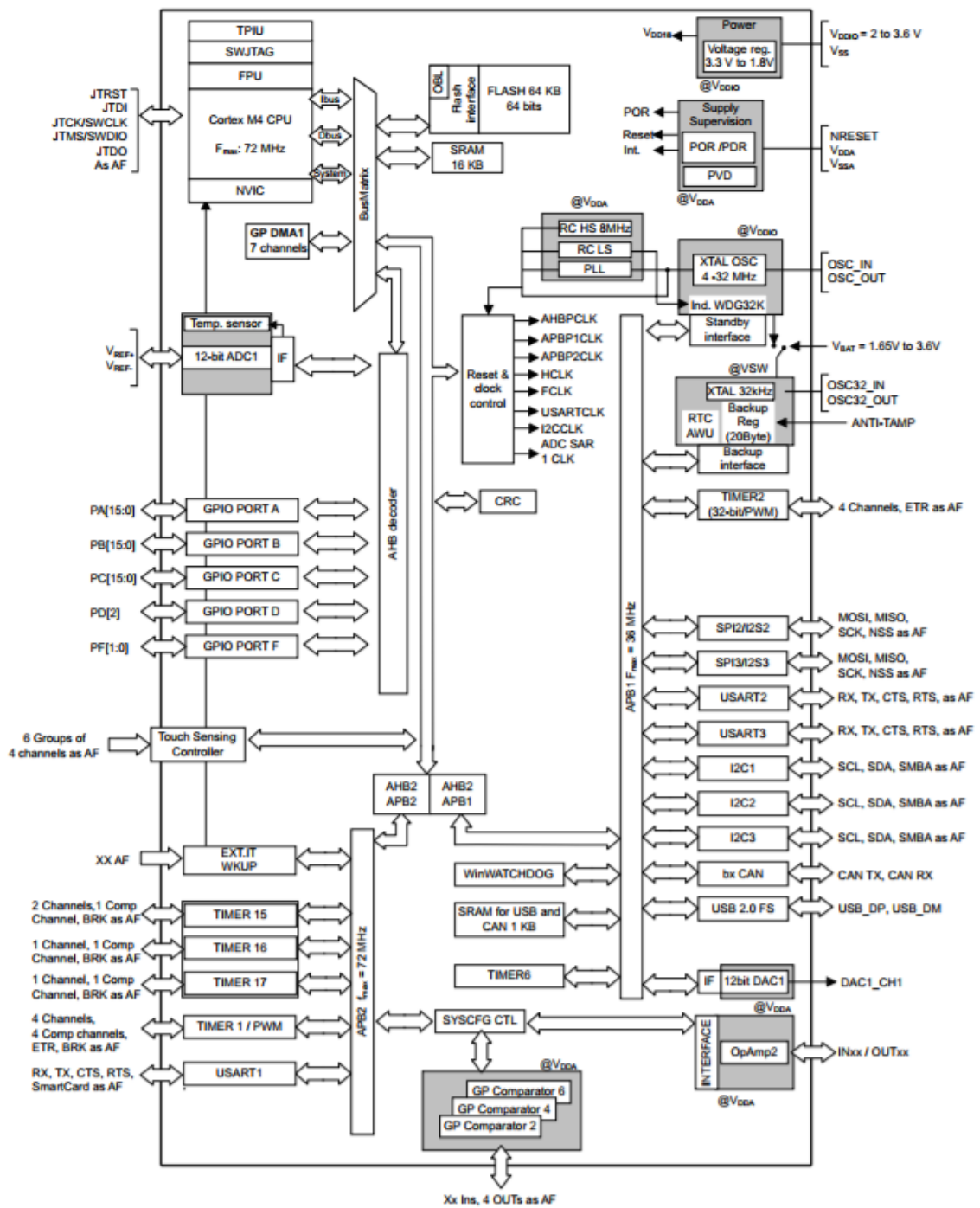
4 HARDWARE A SOFTWARE PRO ŘÍZENÍ BLDC MOTORU

4.1 STM32F302R8

Jedná se o programovatelnou platformu. Jejím základem je výpočetní jednotka se 32 bitovým jádrem ARM Cortex M4, které dovoluje pracovat na frekvenci až 72MHz.[1] Samotná deska má mnoho výstupů a vstupů. Na ně lze připojovat různé periferie. Stavby jednotlivých vstupů a výstupů jsou pak čteny nebo ovládány procesorem. Vývojová platforma také umožňuje připojovat různá rozšíření tzv. shieldy. Jednou z velkých výhod je kompatibilita s platformou Arduino. Díky tomu lze využívat ještě větší množství periférií/ shieldů. Je tedy vhodná pro prototypování a testování nejrůznějších projektů. Blokové schéma vývojové platformy je zobrazeno na Obrázku 15, samotná platforma je pak zobrazena na Obrázku 16.

Díky integrovanému debuggeru a programmeru ST-LINK/V2-1 je také značně usnadněno nahrávání programu přímo do procesoru. Odpadá tak potřeba externího debuggeru.

K vývojové platformě jsou dodávány i knihovny HAL a nejrůznější ukázky programů. Tento fakt přispívá k snadnému učení se tuto platformu ovládat.



Obrázek 15 - Blokové schéma STM32F302R8



Obrázek 16 - STM32F302R8

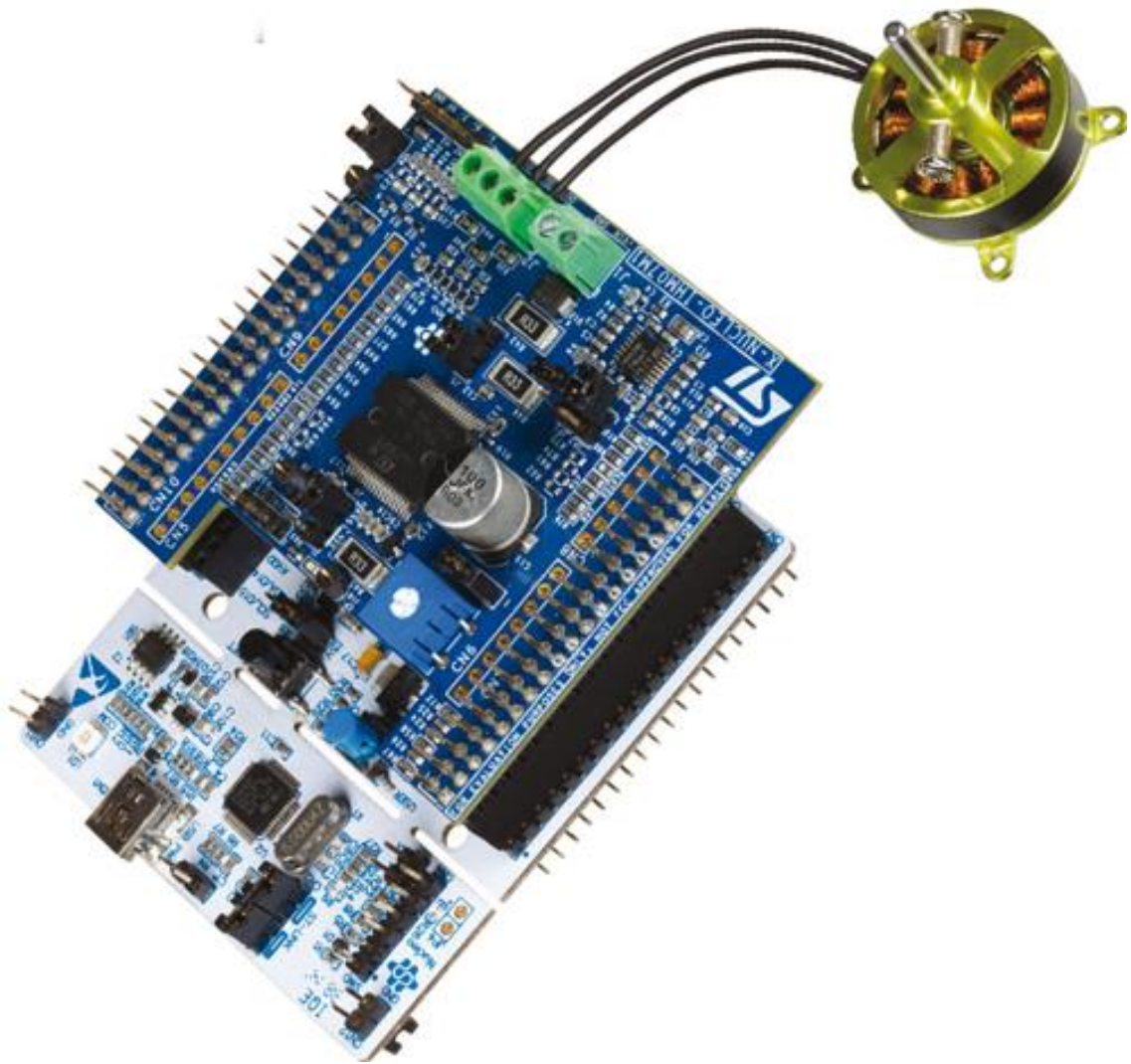
4.2 X-NUCLEO-IHM07M1

Pro snadnější ovládání motoru pomocí vývojové platformy Nucleo-F302R8 na ni lze připojit rozšiřující modul X-Nucleo-IHM07M1. Jedná se o tzv. shield. Modul spolu s platformou STM32F302R8 a motorem nese označení P-NUCLEO-IHM001.

Modul je speciálně zkonstruován pro ovládání třífázového BLDC motoru. Na desce se nacházejí tři výstupy pro připojení jednotlivých fází motoru a vstup pro externí zdroj napájení. Napájet lze stejnosměrným napětím v rozsahu 8 až 48V.[2]

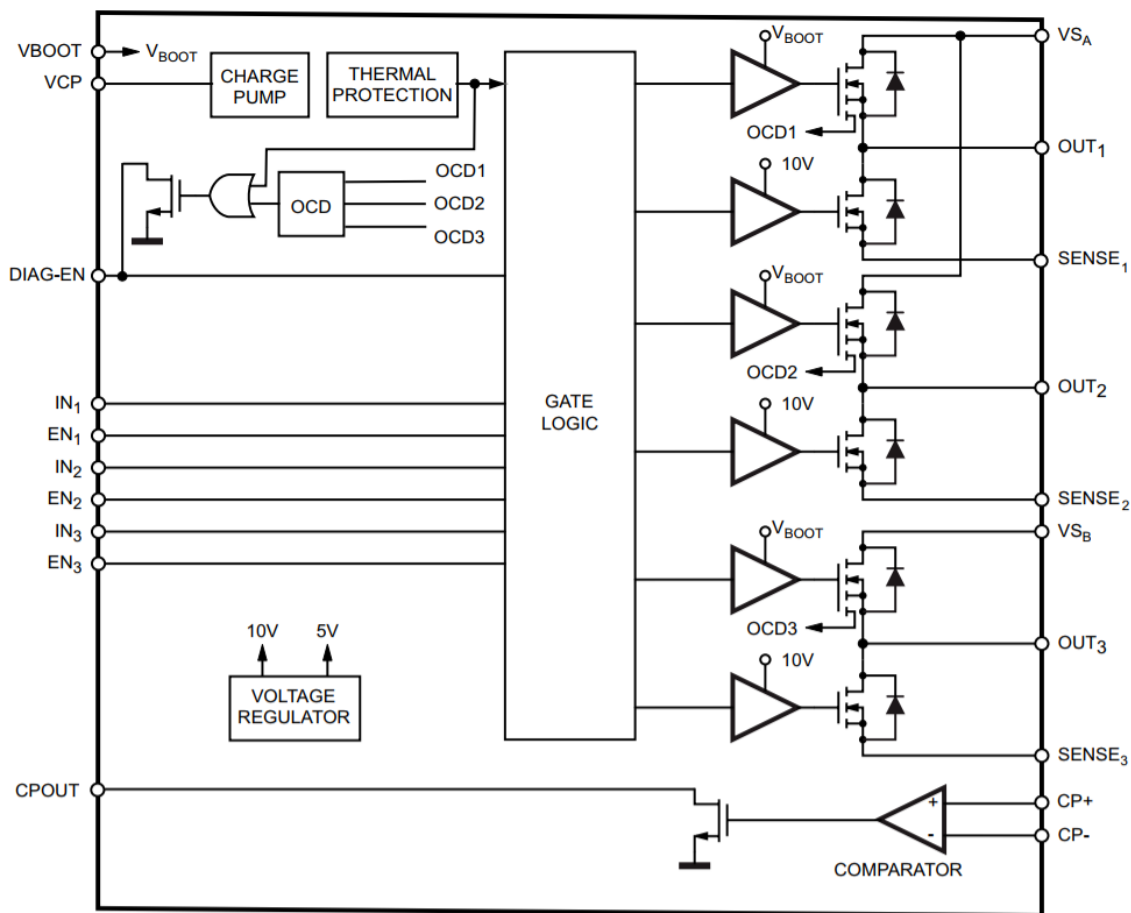
Modul také poskytuje možnost připojit Hallovy sondy, pokud se v motoru nacházejí. V případě bezsenzorového řízení jsou pak vyvedeny tři vstupy s označením Sense, určené pro zero-crossing detekci. Podoba modulu s platformou a motorem je patrná z Obrázku 17.

Stěžejní komponentou modulu je čip L6230. V něm jsou integrovány tranzistory, které umožňují spínání jednotlivých fází motoru. V čipu je také integrovaná nadproudová ochrana, která zabraňuje nechtěnému zničení motoru.



Obrázek 17 - P-NUCLEO-IHM001

Napájet ho lze napětím v rozmezí 8 až 54V a je schopen operovat na frekvenci do 100kHz včetně.[6] Tento čip je pak ovládán samotným procesorem. Dohromady tak tvoří řídicí jednotku, pohánějící motor. Schéma zapojení čipu je znázorněno na Obrázku 18.



Obrázek 18 - Schéma zapojení čipu L6230

4.3 ARM Mbed OS

Pro programování vývojové platformy STM32F302R8 jsem zvolil vývojové prostředí Arm Mbed OS. Jedná se o online vývojové prostředí, je tedy dostupné z jakéhokoli zařízení s internetovým prohlížečem bez nutnosti cokoli instalovat. Programovacím jazykem je zde C/C++. Je zde použito API programovací rozhraní, které zajišťuje funkci kódu na různých vývojových platformách.[17] Na stránkách provozovatele vývojového prostředí je také k nalezení množství různých ukázek programů, které usnadňují vývoj programů vlastních. Je zde možné nalézt také různé návody a vysvětlivky k základním funkcím obsaženým ve vývojovém prostředí. Pro naprosté začátečníky jsou zde k dispozici i výukové stránky vysvětlující nejruznější pravidla programovacího jazyka C/C++. K dispozici je zde také uživatelské fórum, na které přispívá široká komunita. Mbed OS je v současnosti ve verzi 5.

Alternativou k Mbed OS pak mohou být vývojová prostředí Kiel, IAR Ewarm, nebo například AC6. Nejedná se již o online prostředí, ale o prostředí stažitelná, desktopová. Všechna lze ať už s určitým omezením, nebo v plné verzi stáhnout zdarma a podporují široké spektrum produktů od ST Microelectronic.

5 SOFTWAREVÉ ŘÍZENÍ MOTORU

V následující sekci budou postupně probrány jednotlivé části zdrojového kódu s odpovídajícím komentářem. První část bude věnována především Six-step komutaci. Druhá část pak bude zaměřena především na algoritmus bezsenzorového řízení. Probrána zde bude také rozběhová rampa, nezbytná k úspěšnému rozběhu motoru. Podobu zdrojového kódu celého programu je pak možné vidět v příloze na CD.

Jako vývojové prostředí jsem se rozhodl použít online platformu MBed OS umožňující psaní zdrojového kódu v jazyce C/C++. Všechny následující části zdrojového kódu vznikaly na základě knihovny tvořící základ tohoto vývojového prostředí. Tato knihovna se nazývá mbed a zdrojový kód v této práci je založen právě na ní.

5.1 Softwarová Six-step komutace

Podstatou Six-step komutace je vhodné napájení daných cívek. Teorie již v této práci byla probrána. V praxi bylo nutné nadefinovat požadované vstupy cívek, resp. výstupy pulsně šířkové modulace (PWM), kterou budou cívky napájeny. Nutné je také jako výstupy nadefinovat kanály, které povolují přívod napájení ze zdroje na čip L6230. Po bližším prozkoumání datasheetu [2] jsem zjistil, že požadované piny pro povolení napájení jsou značeny jako ENABLE_CH1-L6230, ENABLE_CH2-L6230 a ENABLE_CH3-L6230, kterým odpovídají piny PC10, PC11 a PC12. Vstupy cívek jsou pak značeny jako VH PWM, UH PWM, WH PWM a odpovídají jim piny PA8, PA9 a PA10. Základní konfigurace, tvořící předpoklad pro úspěšné ovládání komutace pak vypadá následovně:

```
//porty pro povoleni napajecich kanalu
DigitalOut phaseAEN(PC_10); //CH1
DigitalOut phaseBEN(PC_11); //CH2
DigitalOut phaseCEN(PC_12); //CH3

//porty pro napajeni jednotlivych civek
PwmOut phaseA(PA_8);    //UH_PWM
PwmOut phaseB(PA_9);    //VH_PWM
PwmOut phaseC(PA_10);   //WH_PWM
```

Dalším krokem je nastavení pulsně šířkové modulace, napájecí cívky. Zvolil jsem periodu spínání 20 μ s. Šířka pulsu, tedy duty cycle bude dále uvažována jako proměnná typu integer nazvaná pwmDuty. Nastavil jsem ji na 10 μ s, tedy 0,5 periody spínání. Softwarový zápis této konfigurace vypadá takto:

```
//nastaveni periody PWM
phaseA.period_us(20);
phaseB.period_us(20);
phaseC.period_us(20);

//nastaveni sirky pulsu
phaseA.write(pwmDuty);
phaseB.write(pwmDuty);
phaseC.write(pwmDuty);
```

Před samotným započítím komutace je žádoucí znát pozici rotoru. Vhodné je tedy jednorázově sepnout cívky odpovídající poslednímu kroku komutace. Při spuštění jsou tak magnety vhodně natočeny vůči působícímu magnetickému poli.

S takto nakonfigurovaným programem již lze úspěšně řídit Six-step komutaci. Pro otáčení ve směru hodinových ručiček jsem zvolil kombinaci spínání odpovídající Tabulce 5.

Tabulka 5 - Spínání fází při komutaci

krok	fáze A	fáze B	fáze C
0		+	-
1	-	+	
2	-		+
3		-	+
4	+	-	
5	+		-

Na fázi s kladnou polaritou je nutné zapisovat PWM signál a fázi se zápornou polaritou je nutno přivést na nulový potenciál. Pro kladnou i zápornou fázi platí, že aby mohly být připojeny na žádaný potenciál je nutné zároveň povolit napájecí kanál. Pro stanovení výchozí pozice rotoru dle tabulky sepne fázi A s kladným a fázi C se záporným potenciálem. Tomu odpovídá kód:

```
phaseAEN = 1;
phaseCEN = 1;
phaseA.write(pwmDuty);
phaseC.write(0);
```

Analogicky můžeme psát kód i pro ostatní stavy sepnutí:

```
//B+, C-
phaseAEN = 0;
phaseA.write(0);
phaseB.write(pwmDuty);
phaseBEN = 1;
```

```
//B+, A-
phaseCEN = 0;
phaseAEN = 1
```

```
//C+, A-
phaseBEN = 0;
phaseB.write(0);
phaseC.write(pwmDuty);
phaseCEN = 1;
```

```
//C+, B-
phaseAEN = 0;
phaseBEN = 1;
```

```
//A+, B-
phaseCEN = 0;
phaseC.write(0);
phaseA.write(pwmDuty);
phaseAEN = 1;
```

```
//A+, C-
phaseBEN = 0;
phaseCEN = 1;
```

Z kódu je patrné, že napájecí kanály nejsou povolovány v příslušných krocích. Je zde totiž vhodně využito faktu, že při přepnutí do dalšího kroku zůstane konfigurace z předchozího kroku zachována až do doby, kdy dojde k její změně. Je tedy zbytečné psát příkaz navíc.

Pro úspěšný chod programu nestačí výše zmíněné kroky komutace pouze opakovat za sebou. Bez vhodného načasování přepínání by se napájení motoru měnilo příliš rychle a rotor by tak nebyl schopen synchronizace s magnetickým polem statoru.

Pro časování jednotlivých kroků lze využít funkce `wait()`, která mezi jednotlivé stavy komutace vnese předem definované zpoždění. Těto možnosti je vhodné využít pro asynchronní rozběh motoru. Pro samotný chod založený na detekci BEMF je však tato metoda nepřijatelná. Jednotlivá zpoždění totiž znemožňují čtení hodnoty BEMF ve vhodný čas a zamezují tak ideálnímu přepnutí následujícího kroku komutace.

Náhradou za funkci `wait()` může být řídicí logika, která je založena na snímání polohy motoru. Jak v této práci již bylo popsáno, lze vyhodnocovat výstup Hallových sond, rotačního enkodéru, nebo snímat BEMF. Programová logika pro bezsenzorové řízení motoru pomocí snímání BEMF bude v této práci dále probrána a vysvětlena.

5.2 Finální verze programu

Pro úspěšnou realizaci jakéhokoli programu je vhodné si navrhnout jeho vývojový diagram. V tomto případě bylo v první řadě nutné motor uvést do výchozí polohy jednorázovým sepnutím daných cívek. V této poloze motor setrvává, dokud nepřijde požadavek na jeho rozběh. Po vyslání požadavku se spustí rozběhová rampa, která motor roztáčí asynchronně se stále se zvyšující rychlostí. Teprve když je dosaženo požadované rychlosti, při které již lze detekovat BEMF cyklus rozběhové rampy, je zastaven. Následně je možné spustit samotný cyklus bezsenzorové rotace motoru. Tento cyklus pokračuje, dokud nepřijde požadavek na ukončení programu. Vývojový diagram je zobrazen na Obrázku 19.



Obrázek 19 - Vývojový digram softwarového řízení

Při psaní programu bylo možné vycházet z kódu, který byl uveden v kapitole Six-step komutace. Bylo ho však nutné značně upravit. Pro úspěšné snímání BEMF bylo nutné nadefinovat porty, k tomu určené, jako vstupy pro analogovou hodnotu. Tyto porty byly v datasheetu [2] značeny jako BEMF1_sensing, BEMF2_sensing a BEMF3_sensing. Tomuto značení odpovídaly piny PC3, PB0 a PB15. Kromě těchto vstupů jsem ještě nadefinoval analogový vstup pro potenciometr, který se nachází na pinu PB1. Také jsem nastavil modré tlačítko na přípravku označené jako USER_BUTTON jako zdroj přerušení. Nakonec jsem nastavil digitální výstup na led diodu umístěnou na pinu LED1, která bude sloužit k signalizaci sepnutí tlačítka. Výsledná konfigurace má potom následující podobu:

```
#include "mbed.h"

//piny pro zero-crossing detekci BEMF
AnalogIn BEMF1(PC_3); //C7_37
AnalogIn BEMF2(PB_0); //C7_34
AnalogIn BEMF3(PA_7); //C10_26 PA_7

//piny pro povoleni kanalu driveru L6230
DigitalOut phaseAEN(PC_10); //CH1
DigitalOut phaseBEN(PC_11); //CH2
DigitalOut phaseCEN(PC_12); //CH3

//vystupy pro napajeni civek PWM modulaci
PwmOut phaseA(PA_8); //UH_PWM
PwmOut phaseB(PA_9); //VH_PWM
PwmOut phaseC(PA_10); //WH_PWM

AnalogIn pot(PB_1); //potenciometr
InterruptIn button(USER_BUTTON); //preruseni od modreho tlacitka
DigitalOut led(LED1); //led dioda pro signalizaci sepnuti tlacitka
```

Pro úspěšný chod programu bylo nutné nadefinovat několik proměnných, které byly dále využity pro různé funkce v programu. Jedná se o proměnné pro pomocné výpočty, případně ukládání programových mezihodnot a různé další účely. Jejich podoba je patrná z následující části kódu. Jejich účel odpovídá popisu v komentářích uvedených za zkratkou „//“.

```

int state = 0; //promenna pro vypocet stavu prepnuti
double pwmDuty = 0.6; //sirka PWM modulace
double potRead; //promenna pro cteni hodnoty potenciometru
int buttonState = 0; //promenna pro ulozeni stavu tlacitka
double pwmSet; //pomocna promenna pro nastaveni sirky PWM pulsu
char doOnce = 0; //zajistuje jednorazove spusteni rozbehove rampy

//promenna pro cteni soucasneho BEMF signalu
float sense1;
float sense2;
float sense3;
int rampX;

//promenna pro zastaveni funkce while pro dane kroky
char stopA = 0;
char stopB = 0;
char stopC = 0;

//ukladani predchoziho BEMF signalu
float sense1Prev = 0;
float sense2Prev = 0;
float sense3Prev = 0;

```

Nastavení výchozí polohy rotoru, stejně jako perioda PWM modulace je umístěno v programové části main(). Výchozí poloha rotoru odpovídá poslední poloze Six-step komutace. Při rozběhu tak motor přepne do první polohy a plynule tak naváže na výchozí stav. Tímto způsobem lze zajistit plynulejší rozběh. Zmíněná část kódu je zobrazena na následující stránce:

```

//nastaveni periody PWM pro jednotlivá vinuti
phaseA.period_us(20);
phaseB.period_us(20);
phaseC.period_us(20);

//vychozi poloha rotoru A+ C- (krok 5)
phaseAEN = 1;
phaseBEN = 0;
phaseCEN = 1;
phaseA.write(1);
phaseC.write(0);

```

Pro počáteční asynchronní rozběh slouží funkce, kterou jsem pojmenoval ramp. Roztáčí motor na stejném principu, jako bylo probráno v kapitole o Six-step komutaci. Rozdíl je v tom, že zpoždění mezi jednotlivými kroky se po každém cyklu zmenšuje a rychlost rotace tak roste. To je zajištěno funkcí for, která zpoždění zmenšuje z hodnoty 7000 μ s na 1000 μ s s krokem 200 μ s. Je také nutné, aby toto navyšování rychlosti proběhlo pouze jednou, proto je spuštění rozběhové rampy ošetřeno podmínkou pro proměnnou doOnce. Tato proměnná po provedení rozběhové rampy změní hodnotu a zaručí tak jednorázový rozběh. Kód pro rozběhovou rampu vypadá následovně:

```

void ramp(){
  if(doOnce == 0){
    for(short rampDelay = 7000; rampDelay >= 1000; rampDelay -= 200){
      phaseAEN = 0;
      phaseA.write(0);
      phaseB.write(pwmDuty);
      phaseBEN = 1;
      wait_us(rampDelay);

      phaseCEN = 0;
      phaseAEN = 1;
      wait_us(rampDelay);

      phaseBEN = 0;
      phaseB.write(0);
      phaseC.write(pwmDuty);
    }
  }
}

```

```

    phaseCEN = 1;
    wait_us(rampDelay);

    phaseAEN = 0;
    phaseBEN = 1;
    wait_us(rampDelay);

    phaseCEN = 0;
    phaseC.write(0);
    phaseA.write(pwmDuty);
    phaseAEN = 1;
    wait_us(rampDelay);

    phaseBEN = 0;
    phaseCEN = 1;
    wait_us(rampDelay);

    doOnce = 1;
  }//for end
} //if doOnce end
} //ramp end

```

Po rozběhu motoru na rychlost, kdy je možné vhodně snímat zpětné elektromotorické napětí (BEMF), je nutné spustit samotnou pracovní sekvenci. V této sekvenci jsou volány jednotlivé kroky komutace. Pro přehlednost zápisu samotné pracovní sekvence jsem si tyto kroky v programu nadefinoval zvlášť a ve vhodnou chvíli se na ně pouze odkazuji. Funkce jsou nazvány step0 až step5 a jejich definice vypadá následovně:

```

void step0(){ //B+, C-
    phaseAEN = 0;
    phaseBEN = 1;
    phaseCEN = 1;
    phaseB.write(pwmDuty);
    phaseC.write(0);
} //step0 end

```

```

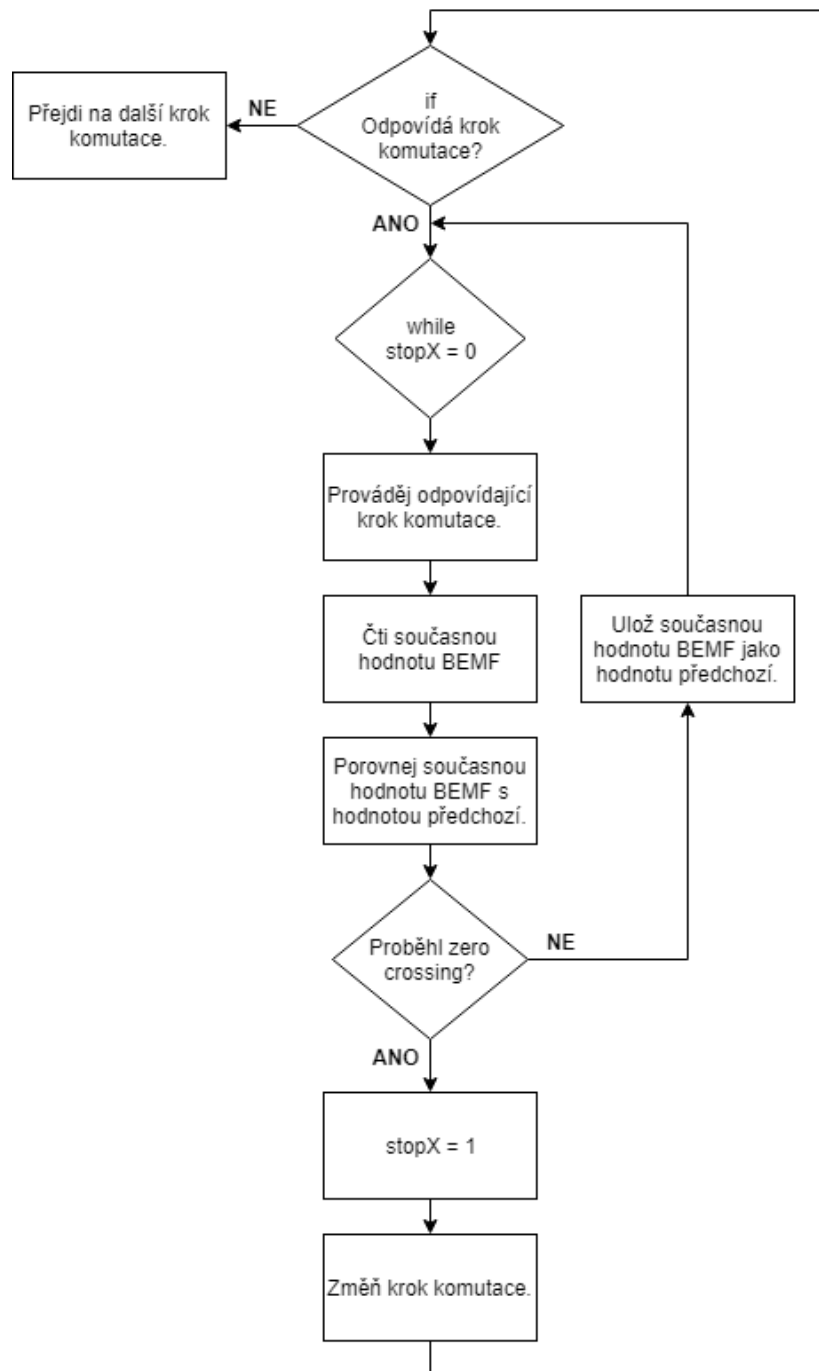
void step1(){ //B+, A-
    phaseAEN = 1;
    phaseBEN = 1;
    phaseCEN = 0;
    phaseA.write(0);
    phaseB.write(pwmDuty);
} //step1 end
void step2(){ //C+, A-
    phaseAEN = 1;
    phaseBEN = 0;
    phaseCEN = 1;
    phaseA.write(0);
    phaseC.write(pwmDuty);
} //step2 end
void step3(){ //C+, B-
    phaseAEN = 0;
    phaseBEN = 1;
    phaseCEN = 1;
    phaseB.write(0);
    phaseC.write(pwmDuty);
} //step3 end
void step4(){ //A+, B-
    phaseAEN = 1;
    phaseBEN = 1;
    phaseCEN = 0;
    phaseA.write(pwmDuty);
    phaseB.write(0);
} //step4 end
void step5(){ //A+, C-
    phaseAEN = 1;
    phaseAEN = 0;
    phaseCEN = 1;
    phaseA.write(pwmDuty);
    phaseC.write(0);
} //step5 end

```

Samotný bezsenzorový běh motoru zajišťuje funkce run. Pro vstup do cyklu, pro jednotlivé kroky komutace, slouží podmínka if, vyhodnocující proměnnou state, která odpovídá kroku komutace. Pokud je podmínka splněna, program se zacyklí v cyklu while podmíněného proměnnou stopX, kde X odpovídá značení fáze pro BEMF detekci. Zde čte analogovou hodnotu z fáze, která v tomto stavu komutace není připojena na zdroj. Tuto hodnotu ukládá do proměnné sense1 a poté sense1Prev. V určitou chvíli tak lze porovnávat hodnotu z předchozího (sense1Prev) a současného (sense1) čtení. Pokud je detekován tzv. zero crossin, tedy pokud průběh hodnot překročí nulovou hodnotu, změní se hodnota proměnné stopX, díky čemuž program opustí cyklus while. Změní se také hodnota proměnné state, takže je možno přejít na další krok komutace. Po vystoupení z cyklu while je opět hodnota stopX vynulována. Díky tomu se zde funkce, po dokončení Six-step komutace, může opět zacyklit. Překročení nuly snímaným průběhem BEMF je detekováno pro překročení ze záporných i kladných hodnot. Na základě teoretických předpokladů je pro určitý krok komutace detekováno překročení nuly pouze v jednom směru. Stejně tak v určitém kroku komutace je nutné měřit BEMF pouze na jedné fázi motoru. Tyto dva faktory zrychlují průběh vyhodnocování programu. Programovou logiku lze popsat stavovým diagramem na Obrázku 20.

Rychlost rotace se v této části programu mění šířkou pulsu PWM modulace, tedy proměnné pwmDuty. Ta odpovídá analogové hodnotě čtené z potenciometru. Vstup potenciometru byl přetypován na hodnotu od 0,4 do 1,0. Nejnížší nastavitelná hodnota byla určena experimentálně s přihlédnutím k úspěšné synchronizaci motoru po rozběhové rampě. Pro hodnoty nižší než 0,4 už úspěšná synchronizace nebyla pravidlem. Kód pro změnu šířky PWM modulace vypadá následovně:

```
void speed(){
    //nastaveni sirky duty cyklu od 0.4 do 1.0
    pwmSet = pot.read();
    pwmSet *= 0.6;
    pwmSet += 0.4;
    pwmDuty = pwmSet;
} //speed end
```



Obrázek 20 - Vývojový diagram BEMF detekce a komutace

Zdrojový kód odpovídající probrané řídicí logice a vývojovému diagramu má následující podobu:


```

void run(){

if(state == 0){ //B+ C-, sense A down
    while (stopA == 0){
        step0();
        sense1 = BEMF1.read();
        if((sense1 <= 0.0f)&&(sense1Prev >= 0.0f)){
            stopA = 1;
            state = 1;
        }//if sense1 end
        sense1Prev = sense1;
    }//while stopA end
    stopA = 0;
} //if count end

if(state == 1){ //B+ A-, sense C up
    while (stopC == 0){
        step1();
        sense3 = BEMF3.read();
        if((sense3 >= 0.0f)&&(sense3Prev <= 0.0f)){
            stopC = 1;
            state = 2;
        }//if sense1 end
        sense3Prev = sense3;
    }//while stopC end
    stopC = 0;
} //if count end

if(state == 2){ //C+ A-, sense B down
    while (stopB == 0){
        step2();
        sense2 = BEMF2.read();
        if((sense2 <= 0.0f)&&(sense2Prev >= 0.0f)){
            stopB = 1;
            state = 3;
        }
    }
}
}

```

```

        }//if sense1 end
        sense2Prev = sense2;
    }//while stopB end
    stopB = 0;
}//if count end

if(state == 3){    //C+ B-, sense A up
    while (stopA == 0){
        step3();
        sense1 = BEMF1.read();
        if((sense1 >= 0.0f)&&(sense1Prev <= 0.0f)){
            stopA = 1;
            state = 4;
        }//if sense1 end
        sense1Prev = sense1;
    }//while stopA end
    stopA = 0;
}//if count end

if(state == 4){    //A+ B-, sense C down
    while (stopC == 0){
        step4();
        sense3 = BEMF3.read();
        if((sense3 <= 0.0f)&&(sense3Prev >= 0.0f)){
            stopC = 1;
            state = 5;
        }//if sense1 end
        sense3Prev = sense3;
    }//while stopC end
    stopC = 0;
}//if count end

if(state == 5){    //A+ C-, sense B up
    while (stopB == 0){
        step5();

```

```

sense2 = BEMF2.read();
if((sense2 >= 0.0f)&&(sense2Prev <= 0.0f)){
    stopB = 1;
    state = 0;
} //if sense1 end
sense2Prev = sense2;
} //while stopB end
stopB = 0;
} //if count end
} //run end

```

Spuštění dosud probraného programu je podmíněno přerušením generovaným tlačítkem. Zmíněný fakt je patrný z vývojového diagramu na Obrázku 19. Program tedy čeká na stlačení tlačítka. Pokud je tlačítko stlačeno, vyvolá se přerušení, které program přesune do funkce, která změní hodnotu proměnné `buttonState`. Proměnná `buttonState` nabývá pouze hodnot 0 a 1. V závislosti na hodnotě této proměnné pak dojde ke spuštění, případně vypnutí programu. Na následující části kódu je vidět logika funkce vyvolané přerušením.

```

void pressed(){
    buttonState = !buttonState;
    led = !led; //signalizace stavu led diodou
} //pressed end

```

Pro `buttonState` s hodnotou 1 se motor rozběhne. V případě nulové hodnoty je provedena funkce `stop`, která vypne napájení cívek a motor se setrvačnými silami zastaví. Tato funkce také změní hodnotu `doOnce`. Zajistí se tak možnost opětovného rozběhu motoru pomocí rampy v případě, že hodnota `buttonState` bude opět rovna jedné. Funkce `stop` má následující podobu:

```

void stop(){
    phaseAEN = 0;
    phaseBEN = 0;
    phaseCEN = 0;
    doOnce = 0; //umožňuje opětovný rozběh po stisknutí
} //stop end

```

Samotné řízení zapínání a vypínání motoru probíhá v programové části main() v nekonečné funkci while(1). Zde jsou volány jednotlivé funkce zajišťující chod programu. Zápis kódu této části vypadá následovně:

```
while(1) {

    button.fall(&pressed); //preruseni od tlacitka

    if(buttonState){ //pokud je buttonState == 1

        ramp(); //asynchronni rozbehova rampa
        run(); //algoritmus pro chod motoru s vyuzitim BEMF
        speed(); //rizeni rychlosti pomoci duty cycle

    }//if buttonState end

    else{
        stop(); //zastaveni motoru
    }//else end

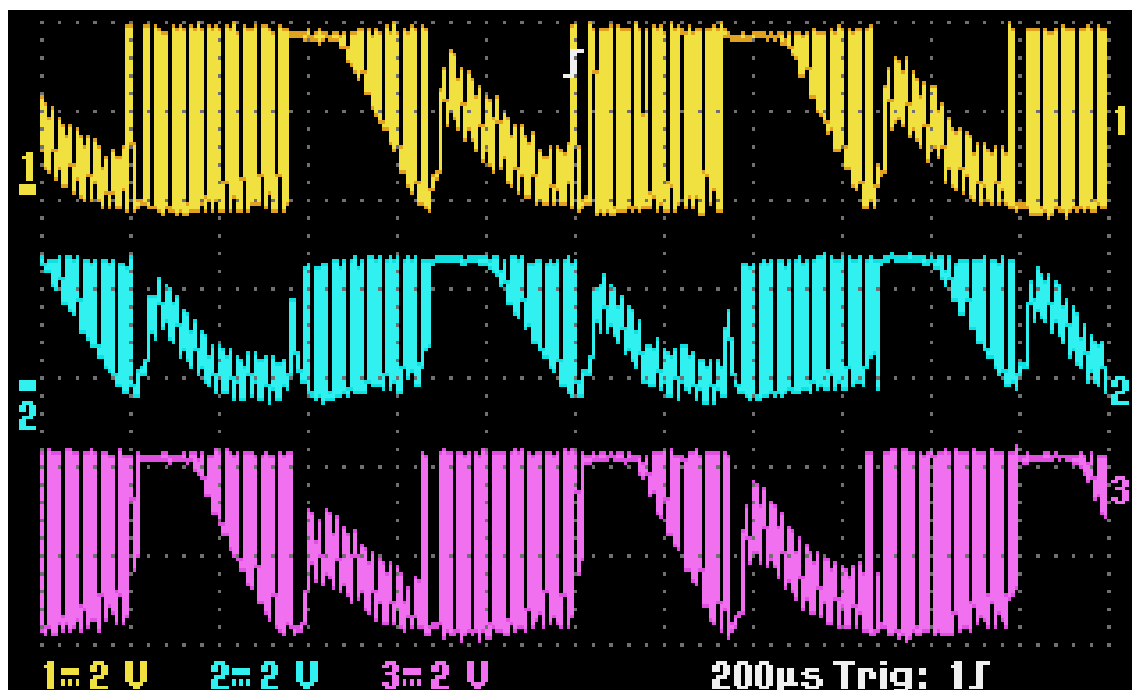
} //while 1 end
```

6 NAMĚŘENÉ PRŮBĚHY

V následující kapitole budou uvedeny oscilogramy průběhů klíčových veličin. Všechny oscilogramy byly naměřeny pomocí bateriového osciloskopu Tektronix THS3024. Průběhy odpovídají veličinám při chodu motoru podle softwarového kódu uvedeného v předešlé kapitole. Všechny veličiny byly měřeny proti potenciálu země. Podoba následujících oscilogramů se v průběhu chodu lišila podle nastavené rychlosti otáčení. Všechny tyto snímky tak byly pořízeny pro fixní polohu potenciometru, který tuto rychlost ovládá.

6.1 Zpětné elektromotorické napětí (BEMF)

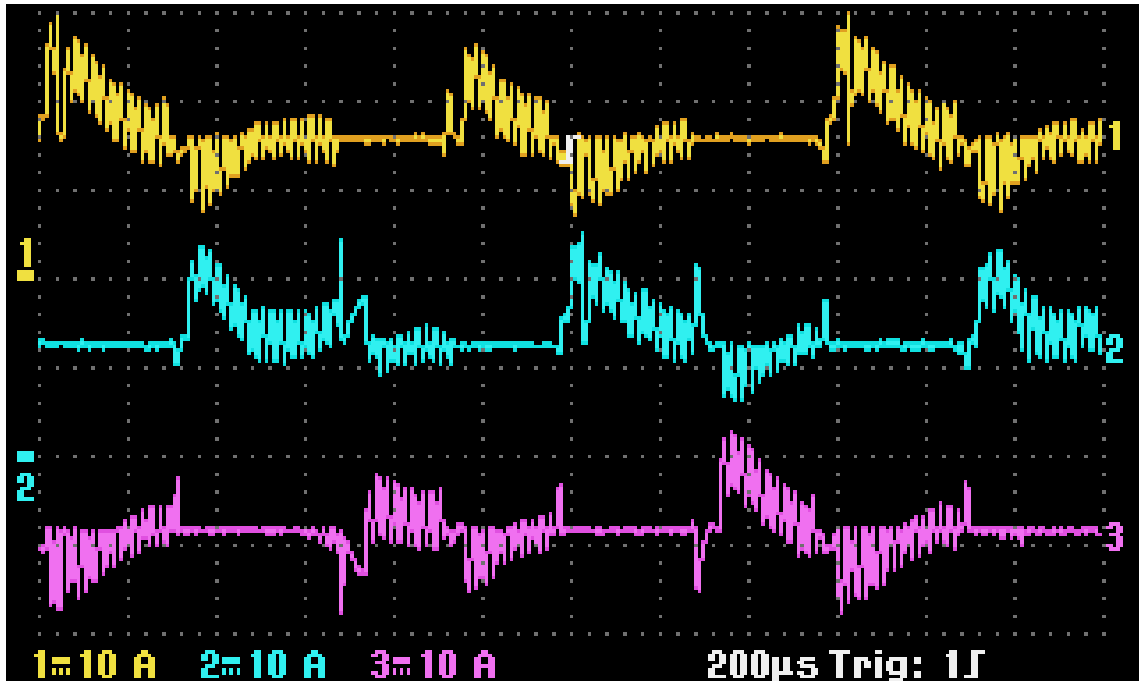
Na následujícím oscilogramu je zobrazeno zpětné elektromotorické napětí. Toto napětí bylo vhodně vzorkováno a vyhodnocováno. Bylo klíčovým pro správné načasování komutace.



Obrázek 21 - BEMF oscilogram

6.2 Proudová zpětná vazba

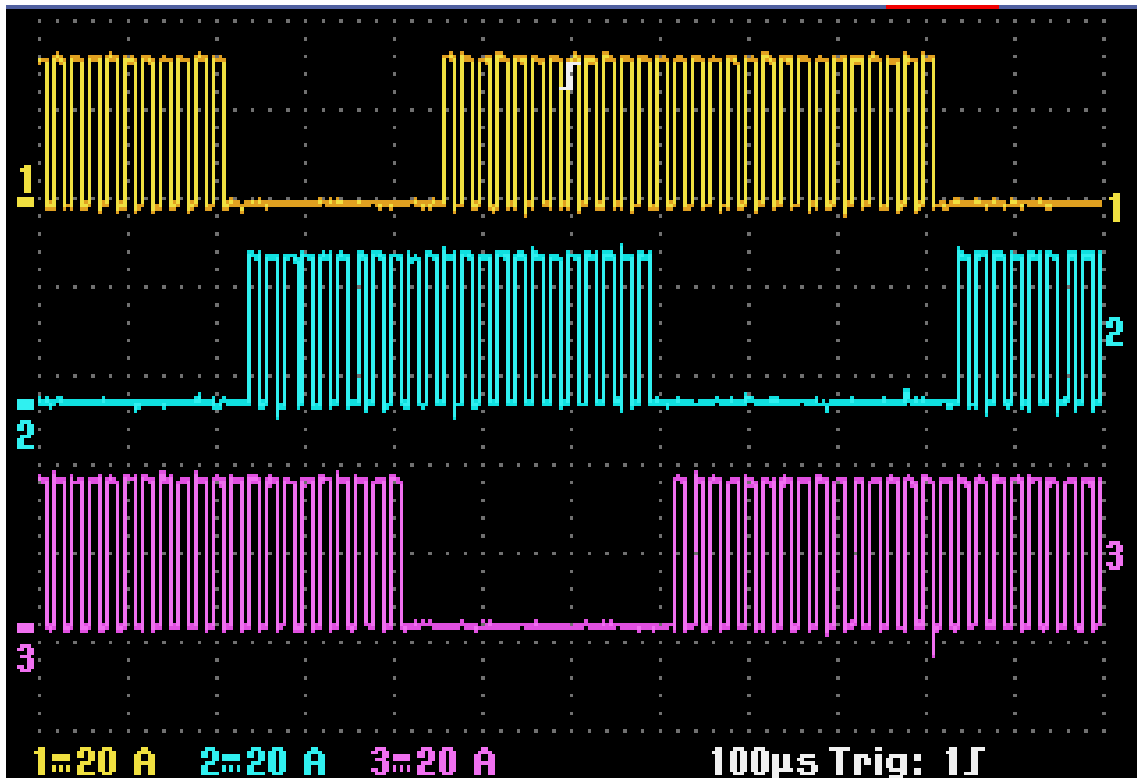
Další průběh, který bylo možno měřit, byl průběh proudové zpětné vazby. Pro řízení motoru nebyl v této práci nijak využit, ale utváří celkovou charakteristiku motoru. Tento průběh je zobrazen na následujícím oscilogramu.



Obrázek 22 - Oscilogram proudové zpětné vazby

6.3 Pulsně šířková modulace (PWM)

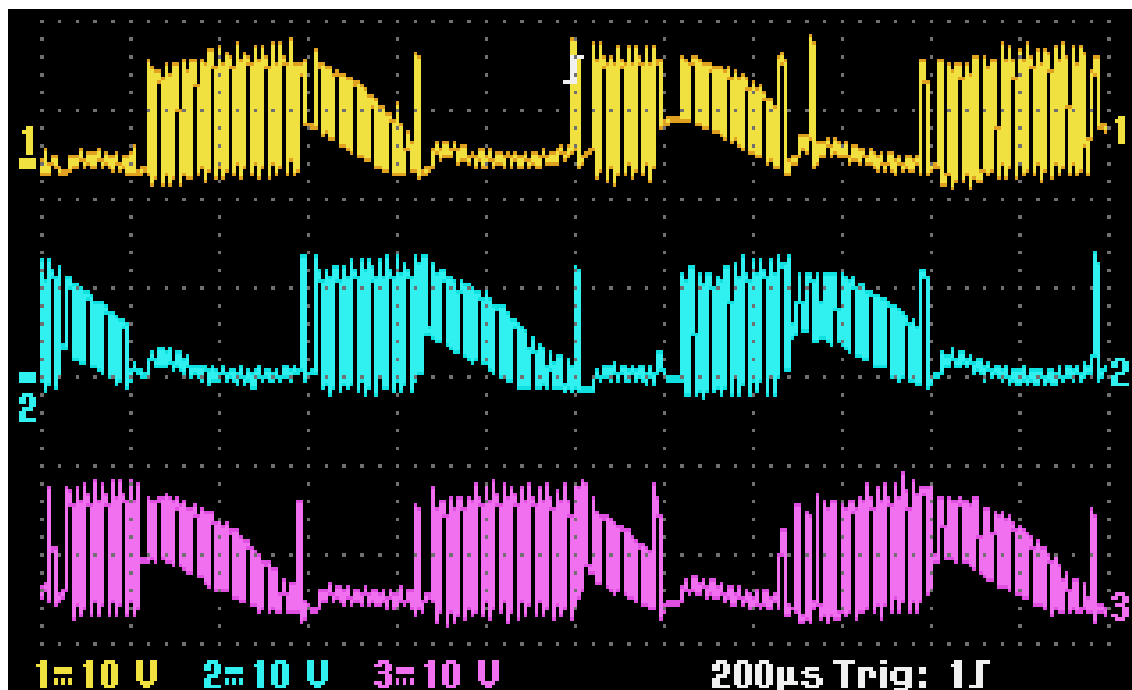
Na základě vyhodnocování průběhu zpětného elektromotorického napětí (BEMF) byly jednotlivé tranzistory a tím i napájení motoru řízeny pomocí signálu pulsně šířkové modulace (PWM). Jeho reálný průběh je patrný z následujícího oscilogramu. Šířka těchto pulsů se v průběhu řízení měnila v závislosti na konkrétní analogové hodnotě potenciometru. Tímto způsobem probíhá ovládání rychlosti.



Obrázek 23 - Oscilogram PWM

6.4 Napájení motoru

Na následujícím snímku je zobrazen výstup z vývojové platformy. Těmito průběhy napětí byl napájen motor při svém chodu.



Obrázek 24 - Oscilogram napájení motoru

7 ZÁVĚR

Cílem této práce bylo seznámení se s problematikou řízení BLDC motorů. V souvislosti s tímto požadavkem bylo nutné blíže se zabírat konstrukcí těchto motorů. V této práci je tak obsažen popis jednotlivých mechanických částí BLDC motorů a vliv jejich konstrukce na funkčnost. Dalším bodem bylo pojednání o různých způsobech řízení motorů. Důležitou částí je popis sensorického snímání polohy rotoru. Pro tuto práci bylo však klíčové bližší zkoumání bezsensorového řízení BLDC motorů na základě snímání zpětného elektromotorického napětí BEMF.

Pro úspěšnou realizaci zadání této bakalářské práce bylo vhodné se blíže seznámit s jednotlivými hardwarovými prvky určenými k řízení BLDC motorů. Z tohoto důvodu jsou zde probrány drivery, a to jak po konstrukční, tak po funkční stránce. Důraz je zde kladen především na konkrétní výrobky od firmy ST Microelectronics, pomocí kterých byl motor v praktické části poháněn. Je zde zmíněn daný přípravek STM32F302R8 a IHM07M1, bez nichž by realizace nebyla možná. Také je zde opodstatněn výběr vývojového prostředí Mbed OS, ve kterém byl vytvořen daný zdrojový kód pro pohon motoru.

Podstatná část bakalářské práce se věnuje praktické části zadání, a sice tvorbě software pro pohon motoru. Jsou zde zobrazeny vývojové diagramy odpovídající programové logice spolu s jednotlivými částmi kódu. Tento kód je zde pak podrobně vysvětlen a obohacen o komentáře usnadňující jeho pochopení. V závěru jsou pak uvedeny oscilogramy pořízené pro motor poháněný přípravkem IHM001 pracujícím podle programu zde probraného.

LITERATURA

- [1] ST Microelectronic. STM32F302x6 STM32D302x8. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/e4/06/d5/04/58/af/41/14/DM00093333.pdf/files/DM00093333.pdf/jcr:content/translations/en.DM00093333.pdf>
- [2] ST Microelectronic. UM1943 User manual. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: http://www.st.com/content/ccc/resource/technical/document/user_manual/group0/83/ea/b9/2d/82/7b/4a/ed/DM00226187/files/DM00226187.pdf/jcr:content/translations/en.DM00226187.pdf
- [3] ST Microelectronic. UM1945 User manual. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: http://www.st.com/content/ccc/resource/technical/document/user_manual/70/a6/a2/95/7a/3d/4d/f8/DM00226315.pdf/files/DM00226315.pdf/jcr:content/translation/en.DM00226315.pdf
- [4] ST Microelectronic. STM32PMSM FOC SDK 4.3 Getting Started. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: http://www.st.com/content/ccc/resource/sales_and_marketing/presentation/application_presentation/group0/83/54/8e/3b/90/89/4c/e7/STM32_PMSM_FOC_SDK_Getting_Started/files/stm32_pmsm_foc_sdk_getting_started.pdf/jcr:content/translations/en.stm32_pmsm_foc_sdk_getting_started.pdf
- [5] ST Microelectronic. UM2124 User manual. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: http://www.st.com/content/ccc/resource/technical/document/user_manual/group0/90/b6/9f/0c/50/c1/47/1d/DM00334922/files/DM00334922.pdf/jcr:content/translations/en.DM00334922.pdf
- [6] ST Microelectronic. L6230. *ST Microelectronic* [online]. 2016 [cit. 2018-06-04]. Dostupné z: <http://www.st.com/resource/en/datasheet/l6230.pdf>
- [7] ST Microelectronic. AN1946 APPLICATION NOTE. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: http://www.st.com/content/ccc/resource/technical/document/application_note/aa/b4/69/3f/75/58/4a/a1/CD00020086.pdf/files/CD00020086.pdf/jcr:content/translations/en.CD00020086.pdf
- [8] MADAAN, Pushek. Brushless DC Motors – Part I: Construction and Operating Principles. *EDN NETWORK* [online]. 2013. [cit. 2018-01-06]. Dostupné z: <https://www.edn.com/design/sensors/4406682/Brushless-DC-Motors---Part-I--Construction-and-Operating-Principles>
- [9] ING. JOSEF, Šimon. BLDC nebo DC motor s nulovými náklady na údržbu. *ELEKTRO časopis pro elektroniku* [online]. 2011. [cit. 2018-01-06]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/bldc-aneb-dc-motor-s-nulovymi-naklady-na-udrzbu--10016>

- [10] ICE Charlie Microchip Technology. Bezsenzorové řízení BLDC motorů pro široké využití. *DPS*, Elektronika od A do Z [online]. 2014. [cit. 2018-01-06]. Dostupné z: <https://www.dps-az.cz/soucastky/id:2700/bezsenzorove-rizeni-bldc-motoru-pro-siroke-vyuziti>
- [11] VOJÁČEK, Antonín. Magnetické senzory s Hallovým efektem – 1. princip. *Automatizace.hw.cz* [online]. 2007. [cit. 2018-01-06]. Dostupné z: <https://automatizace.hw.cz/magneticke-senzory-s-hallovym-efektem-1-princip>
- [12] PROKOP Libor / CHALUPA Michael. 3-Phase BLDC Motor Control with Sensorless Back EMF Zero Crossing Detection Using 56F80x. Freescale Semiconductor [online]. 2005. [cit. 2018-01-06]. Dostupné z: <http://cache.freescale.com/files/product/doc/AN1914.pdf>
- [13] ZHAO, Jian / YU, Yangwei. Brushless DC Motor Fundamentals Application Note. MSP [online]. 2011. [cit. 2018-01-06]. Dostupné z: <https://www.monolithicpower.com/Portals/0/Documents/Products/Documents/appnotes/Brushless%20DC%20Motor%20Fundamentals.pdf>
- [14] Doc. Ing. Hrabovcová Valéria, PhD / Ing. Ladislav Janoušek / Ing. Pavol Rafajdus / Ing. Miroslav Ličko, PhD. *Moderné elektrické stroje*. Žilinská univerzita v Žilíně. 2001. ISBN 80-7100-809-5
- [15] Ing. Koblrl Pavel / Ph.D. prof. Ing. Pavelka Jiří, DrSc. *Elektrické pohony a jejich řízení*. ČVUT v Praze. 2016. ISBN 978-80-01-06007-0
- [16] CHAN Chin-Long / SAMAD Bin Zahurin. Brushless DC Motor Electromagnetic Torque Estimation with Single-Phase Current Sensing. *J Electr Eng Technol Vol. 9*. Strana: 2-3. 2014. ISSN 2093-7423
- [17] ARM Mbed. Introduction. ARM Mbed [online]. 2018 [cit. 2018-06-04]. Dostupné z: <https://os.mbed.com/docs/v5.8/introduction/index.html>

ZDROJE POUŽITÝCH OBRÁZKŮ

- Obrázek 5 Phidgets Inc. Encoder Primer. Phidgets [online]. 2017 [cit. 2018-06-04]. Dostupné z: https://www.phidgets.com/docs/images/b/b1/Absolute_vs_incremental.jpg
- Obrázek 15 ST Microelectronic. STM32F302x6 STM32D302x8. *ST Microelectronic* [online]. 2017 [cit. 2018-01-06]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/e4/06/d5/04/58/af/41/14/DM00093333.pdf/files/DM00093333.pdf/jcr:content/translations/en.DM00093333.pdf>
- Obrázek 16 ST Microelectronic. STM32D302R8. *ST Microelectronic* [online]. [cit. 2018-06-04]. Dostupné z: http://www.st.com/content/ccc/fragment/product_related/rpn_information/board_photo/2e/4a/01/43/4f/fa/4a/3d/nucleo-F1.jpg/files/nucleo-F1.jpg/_jcr_content/translations/en.nucleo-F1.jpg
- Obrázek 17 ST Microelectronic. P-NUCLEO-IHM001. *ST Microelectronic* [online]. 2017 [cit. 2018-06-04]. Dostupné z: <http://www.st.com/en/evaluation-tools/p-nucleo-ihm001.html>
- Obrázek 18 ST Microelectronic. L6230. *ST Microelectronic* [online]. [cit. 2018-06-04]. Dostupné z: http://www.st.com/content/ccc/fragment/product_related/rpn_information/product_circuit_diagram/53/20/5c/2b/63/4e/4e/63/circuit_diagram_18094_thumbnail.png/files/circuit_diagram_18094_thumbnail.png/_jcr_content/translations/en.circuit_diagram_18094_thumbnail.png

Všechny ostatní obrázky byly vytvořeny autorem.