

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

MODUL PRO ŘÍZENÍ STEJNOSMĚRNÝCH MOTORŮ

DC MOTOR CONTROLLER MODULE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Aleksei Duz

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radek Kuchta, Ph.D.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**

Ústav mikroelektroniky

Student: Bc. Aleksei Duz

ID: 174283

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Modul pro řízení stejnosměrných motorů

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je návrh více kanálového modulu pro řízení stejnosměrných motorů. Během řešení práce se seznámte s konstrukcí a možnostmi řízení stejnosměrných motorů. Vyberte vhodné snímače polohy rotoru motoru. Navrhněte a pomocí simulací ověřte vhodný model řízení. Modul musí umožnit komunikaci s nadřazeným systémem, za tímto účelem vyberte vhodnou komunikační sběrnici. Součástí řešení bude návrh desky plošného spoje podle navrženého schématu a implementace programového vybavení do řídicího mikrokontroléru. Ověřte vlastnosti vytvořeného řešení v reálných podmínkách.

DOPORUČENÁ LITERATURA:

[1] Fischer P. High Performance Brushless DC Motor Control. School of Engineering & Technology CQUniversity Australia. May 2014

Termín zadání: 3.2.2020

Termín odevzdání: 19.5.2020

Vedoucí práce: doc. Ing. Radek Kuchta, Ph.D.

doc. Ing. Lukáš Fucík, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá návrhem a vytvořením řídicí jednotky pro řízení stejnosměrných motorů. Práce okrajově popisuje princip, na kterém je založen stejnosměrný motor, způsob jeho řízení a možnostmi získání aktuální informace o poloze. Následně jsou popsány různé typy sběrnic pro zajištění komunikace s nadřazeným systémem a odůvodnění zvolené varianty. V další sekci je kladen zvláštní důraz na vytvoření vhodného modelu řízení, návrh elektrického schématu a desky plošných spojů. V závěru práce je popis oživování realizovaného modulu a testování jeho základních funkcí.

KLÍČOVÁ SLOVA

EC a DC motory, řízení motorů, EtherCAT, regulátory, DPS

ABSTRACT

This thesis focuses on the design and creation of control unit for DC motors drives. A principle on which the DC motor is based, a method of its control and a possibility to obtain current information about the position are briefly described in this work. After that, the different types of buses for communication with a superior system and justification of the chosen one, are presented. In the next section, there is a special emphasis on creation of the suitable driver model and a design of the electrical schematic and PCB. The end of the thesis contains a description of commission a PCB and testing the main functions.

KEYWORDS

EC a DC motors, control motors, EtherCAT, regulators, PCB

DUZ, Aleksi. *Modul pro řízení stejnosměrných motorů*. Brno, 2020, 78 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce: doc. Ing. Radek Kuchta, Ph.D.

PROHLÁŠENÍ

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

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

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Radku Kuchtovi, Ph.D. a majiteli firmy DFC Design s.r.o. Ing. Soběslavu Valachovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	12
1 Teorie	13
1.1 Motory	13
1.1.1 Kartáčové motory	13
1.1.2 Bezkartáčové motory	16
1.2 Řízení stejnosměrného motoru	18
1.2.1 Řízení polohy	18
1.2.2 Kvadrantové řízení	19
1.3 Regulátory	20
1.3.1 Lineární proudový regulátor	20
1.3.2 Pulzní proudový regulátor	21
1.4 Zapojení pro ovládání stejnosměrného motoru	21
1.4.1 Poloviční H-můstek	22
1.5 Senzory polohy	22
1.5.1 Enkodéry	23
1.5.2 Resolver	24
1.6 Komunikace	25
1.6.1 CAN	25
1.6.2 ProfiBUS	26
1.6.3 Průmyslové Ethernety	27
2 Návrh	30
2.1 Matematický model motoru	30
2.2 Regulace	32
2.2.1 Návrh regulátorů	33
2.2.2 Testování řízení	36
2.3 Návrh elektrického schématu	39
2.3.1 Řídící jednotka	39
2.3.2 Můstkový driver	40
2.3.3 EtherCAT	43
2.3.4 Ochrana	44
2.3.5 Konektory	45
2.3.6 Napájení modulu	48
2.4 Návrh DPS	50
2.4.1 Rozmístění součástek	50
2.4.2 Skladba DPS	51

2.4.3	Vedení výkonové části	52
2.4.4	Propojení vrstev	52
2.4.5	Vedení rychlých signálů	52
2.4.6	Fixace modulu	54
2.4.7	Výroba desky plošných spojů a příprava k osazení	54
3	Programové vybavení modulu	56
3.1	EtherCAT	56
3.1.1	Topologie	56
3.1.2	Odolnost	57
3.1.3	Struktura rámce	58
3.1.4	Adresování	58
3.1.5	Synchronizace	59
3.2	Master	59
3.3	Slave	60
3.3.1	Implementace komunikace	61
3.3.2	Implementace řízení	63
3.3.3	Hlavní smyčka firmwaru	64
4	Oživování	66
5	Testování modulu	67
5.1	Test komunikace	67
5.2	Test můstkových driverů	68
5.3	Test polohovací úlohy	68
5.4	Test ochran	70
	Závěr	72
	Literatura	75

Seznam obrázků

1.1	Klasifikace stejnosměrných motorů	13
1.2	Klasický kartáčový motor	14
1.3	Samonosné vinutí MAXON	14
1.4	Princip kartáčového DC motoru	15
1.5	Typická konstrukce EC motoru	17
1.6	Princip BLDC motoru	17
1.7	Uzavřená regulační smyčka řízení polohy	18
1.8	1Q a 4Q řídicí jednotka	19
1.9	Lineární proudový regulátor	20
1.10	Pulzní proudový regulátor	21
1.11	Poloviční H-můstek	22
1.12	Struktura ISO/OSI	26
1.13	Model ProfiBUSu	27
1.14	Model ProfiNETu	28
1.15	Model EtherCATu	29
2.1	Elektrické schéma stejnosměrného motoru	30
2.2	Matematický model stejnosměrného kartáčového motoru	32
2.3	Vytvořená regulace	33
2.4	MATLAB Simulink - Tuning Tool - Návrh regulátoru	35
2.5	Zapojení systému	37
2.6	Odezva systému - PI-regulátor polohy	38
2.7	Odezva systému - P-regulátor polohy	38
2.8	Blokové schéma navržené DPS	39
2.9	Zapojení zesilovače pro měření proudu	42
2.10	Model zapojení PHY	43
2.11	Elektrické zapojení PHY	44
2.12	Zapojení konektoru s Hallovou sondou	46
2.13	Příklad optického enkodéru	47
2.14	Model napájení procesorové části	49
2.15	Příklad sítí prokovů	52
2.16	Hotová DPS - Top strana	55
2.17	Hotová DPS - Bottom strana	55
3.1	Průchod EtherCAT rámce	57
3.2	Průchod EtherCAT rámce při poruše	57
3.3	Průchod EtherCAT rámce v odolném zapojení při poruše	57
3.4	Ethernet rámeček	58
3.5	EtherCAT rámeček	58

3.6	Datagram ethercatového rámce	58
3.7	Plná inicializace mastera se slavem	60
3.8	Tvar EtherCAT řetězce	62
3.9	Změny souboru XMC_ESC.c	63
3.10	Alokace pinů EtherCAT modulu	63
3.11	Hlavní smyčka firmwaru	65
3.12	Funkce process_app	65
5.1	Průběh pohybu o vzdálenost 200mm	69
5.2	Průběh proudu při pohybu o vzdálenost 200mm	69
5.3	Signály z enkodéru	70
5.4	Testování modulu pro řízení stejnosměrných motorů	71
5.5	Modulu pro řízení stejnosměrných motorů - Top	73
5.6	Modulu pro řízení stejnosměrných motorů - Bot	74

Seznam tabulek

1.1	Tabulka enkodérů firmy MAXON	25
2.1	Tabulka výpočtu koeficientu dle Ziegler-Nicholse	34
2.2	3,3 V - proudový odběr	48
2.3	5 V - proudový odběr	49
2.4	12 V - proudový odběr	49
2.5	Původní skladba DPS	51
2.6	Upravená skladba DPS	51
2.7	Délky datových spojů MCU - PHY	53
5.1	Parametry motoru	67
5.2	Parametry enkodéru	67
5.3	Specifikace výsledného modulu	73

Seznam zkratek

BDC - Brushed Direct Current

BLDC - Brushless Direct Current

DC - Direct Current

DPS - Deska Plošných Spojů

EC - Electronic Commutation

ESC - EtherCAT Slave Controller

ESI - EtherCAT Slave Information file

EtherCAT - Ethernet for Control Automation Technology

IDE - Integrated Development Environment

MCU - Microcontroller

PHY - Physical Layer Transceiver

PWM - Pulse Width Modulation

SMD - Surface Mount Device

SSC - Slave Stack Code

Úvod

Stejnoseměrné motory různého typu nachází své použití v mnoha aplikacích díky svým parametrům, rozměrům a možnostem. V průmyslu vznikají požadavky na malé stejnosměrné motory doplněné snímači a převodovkami z toho důvodu, že se dá provést propojení se systémem celého stroje (hlavní řídicí jednotkou), potřebují minimum místa atd. Jen správně vybraný motor může splňovat veškeré dynamické požadavky stroje.

V první části diplomové práce se rozebírají principy funkčnosti stejnosměrných kartáčových a bezkartáčových motorů, způsob jejich elektrického ovládní, možnosti regulace a řízení. Proto se dělá řešerše možných zpětných vazeb (snímačů) a různých typů sběrnic pro komunikaci se systémem stroje.

Druhá část popisuje model motoru, návrh regulace, návrh elektrického schématu a desky plošných spojů. Kapitoly o regulaci a modelu motoru se zabývají simulací chování motoru v reálných podmínkách. Dále je v této části prezentován návrh elektrického schématu a DPS, který je podložen výpočty.

Třetí část se zabývá vývojem programového vybavení řídicí jednotky modulu. V této části práce se hlouběji rozebírá vybraná sběrnice, popisuje se způsob vytvoření nadřazené jednotky (masteru) a firmware pro navržený modul.

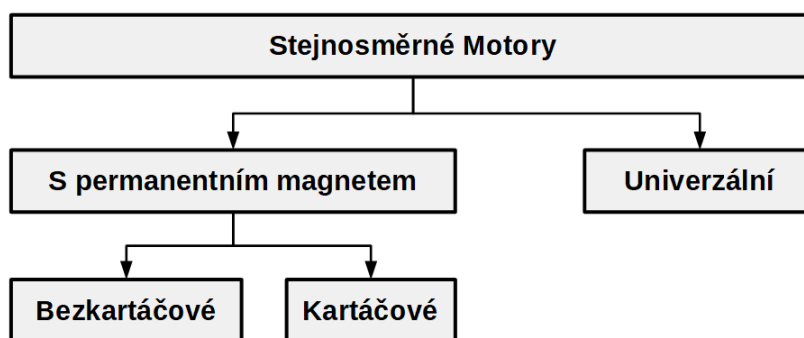
Poslední části se zabývá oživováním již vyrobeného modulu, a prezentují se postupy testování a jejich výsledky. Dále je zde zhodnocena provedená práce, jsou popsána budoucí vylepšení a možnosti využití navrženého modulu.

1 Teorie

Následující kapitoly rozebírají různé stejnosměrné motory a popisují principy činnosti jednotlivých typů. Z toho důvodu se dělá průzkum způsobů jejich řízení a regulace. Z hlediska budoucího modulu se dělá rozbor hardwarového provedení, zpětných vazeb a komunikace.

1.1 Motory

Stejnsměrné motory se rozdělují způsobem znázorněným na obrázku 1.1. Na tomto obrázku nejsou znázorněny krokové motory, které jsou speciálním případem motorů s komutací PWM. Práce se zabývá řízením motorů s permanentními magnety. Budou popsány motory kartáčové a bezkartáčové.



Obr. 1.1: Klasifikace stejnosměrných motorů

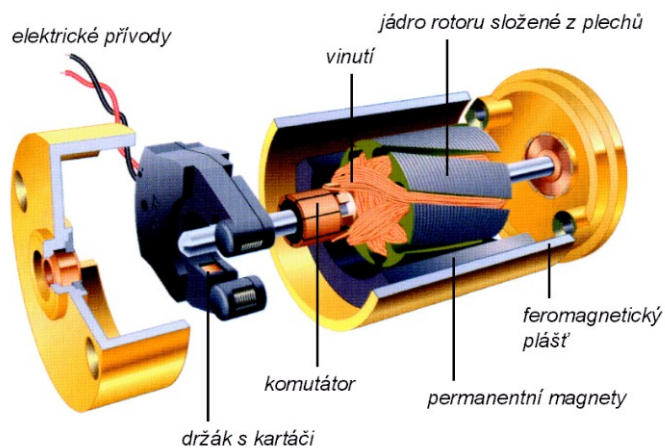
1.1.1 Kartáčové motory

Kartáčové motory mají jednoduché ovládání, stačí jenom přivést stejnosměrné napětí a motor se roztočí. Kvůli lepšímu pochopení těchto motorů je nutno se orientovat v jejich konstrukci a na jakém fyzikálním principu jsou založeny.

Konstrukce

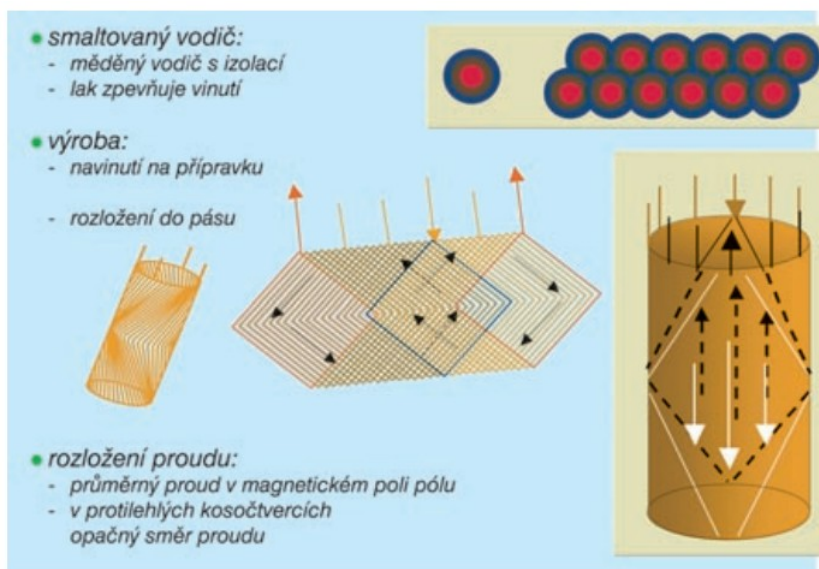
Kartáčové motory (DC motory, Brushed DC motory, BDC motory) se skládají z permanentního magnetu, kartáčů, hřídele s plechovým jádrem, lamel a vinutí rotoru.

Z obrázku 1.2 je vidět, jakým způsobem je sestaven motor a jak vypadá tvar jádra rotoru motoru. Je to standardní provedení rotoru. Takto tvarované plechové jádro má výhodu v tom, že snižuje ztráty vířivými proudy, a tím pádem zvyšuje účinnost motoru. Firma MAXON se v tomto posunula trochu dál a vyvinula samonosné vinutí



Obr. 1.2: Klasický kartáčový motor [1]

rotoru, které je zobrazeno na obrázku 1.3. Toto vinutí má výhodu v tom, že motor s určitým výkonem, který bude mít implementované toto vinutí, bude ve výsledku lehčí a menší oproti motoru se standardním jádrem. Siločáry magnetické indukce se uzavřou přes vinutí a vzduch beze ztrát v železe. Samotné železné jádro je tímto vinutím odstraněno.



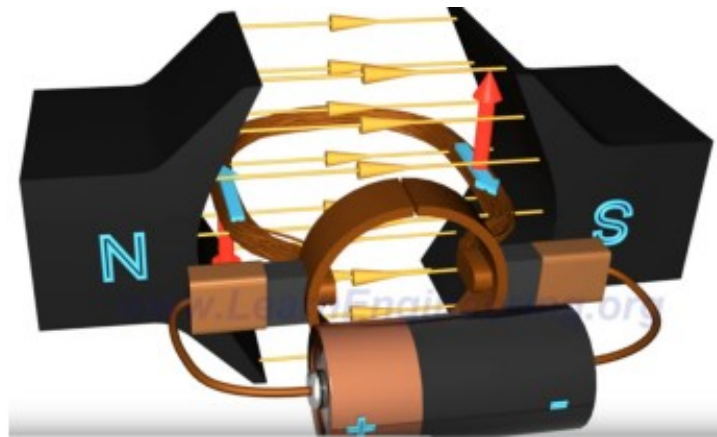
Obr. 1.3: Samonosné vinutí MAXON [3]

Další variabilitou konstrukce je použití kovových nebo grafitových kartáčů. Kovové kartáče se používají v aplikacích s rovnoměrným chodem bez častých rozběhů. Oproti grafitovým mají menší sílu přitlaku k lamelám komutátoru, což vede k men-

šímu mechanickému brzdění. Navíc kovové kartáče nevytváří uvnitř motoru prach. Grafitové kartáče se naopak používají v aplikacích s častými rozběhy a provozech s krátkodobým přetížením. Díky větší ploše jsou schopny dodat do vinutí motoru větší proud, ale potřebují větší přítlak k lamelám. To způsobuje větší brzdění. Jak již bylo poznamenáno, opotřebení těchto kartáčů vytváří grafitový prach, který se může využít jako mazivo komutátoru.

Princip

Kartáčové motory jsou založeny na následujícím fyzikálním principu. Na elektrický přívod motoru se přivede elektrické napětí, a cívkou rotoru začne téct proud přes lamely komutátoru. Díky magnetickému poli tvořenému permanentním magnetem a procházejícímu proudem vznikne Lorentzova síla, která bude působit na vinutí pohyblivé části motoru, a rotor se začne otáčet. Obrázek 1.4 znázorňuje vysvětlený fyzikální princip.



Obr. 1.4: Princip kartáčového DC motoru [4]

Lorentzova síla je daná vztahem

$$F = B * Q * v, \quad (1.1)$$

kde B je indukce magnetického pole, Q je náboj a v je rychlost. Náboj se dá rozložit na proud protékající vodičem za jednotku času, a rychlost na podíl délky vodiče a času. Po rozepsání Q a v dostaneme vztah, který reprezentuje Lorentzovou sílu v motoru:

$$F = B * i * l * N, \quad (1.2)$$

kde N je počet závitů vodiče, i je proud vinutím a l je délka vodiče.

Životnost

Životnost kartáčových stejnosměrných motorů je limitována kartáči a lamely, protože při přechodu kartáčů mezi lamelami komutátoru vzniká jiskření, které je způsobené rozpojením lamely a kartáče. Uvedené jiskření je důsledkem zániku magnetického pole, které se přeměnilo na elektrický proud uzavírající se přes oblouk. Důsledkem toho je opalování kartáčů a lamel. Aby se zamezilo jiskření, dávají se mezi lamely komutátoru kondenzátory. Tím pádem se energie přemění na elektrický proud, který nabijí kondenzátor, což eliminuje vznik oblouku a zvyšuje životnost kartáčového motoru.

Životnost komutátorového motoru se pohybuje od stovek hodin až do několika desítek tisíc hodin. Délka životnosti kartáčového motoru může jít přes desítky tisíc hodin, pokud se motor bude provozovat s nízkým proudovým zatížením a bude mít zapojené kondenzátory mezi lamelami.

1.1.2 Bezkartáčové motory

Bezkartáčové motory mají oproti kartáčovým komplikovanější řízení. Stejnosměrné napětí se má přivést na vinutí motoru ve správnou chvíli a na určitou dobu. Tato specifická vlastnost výrazně komplikuje řídicí elektroniku a firmware.

Konstrukce

U bezkartáčových stejnosměrných motorů (BLDC motory, EC motory) byly odstraněny kartáče a lamely. Uvedená úprava konstrukce zvětšila životnost motoru, rychlost otáčení rotoru, a zmenšila elektromagnetické rušení.

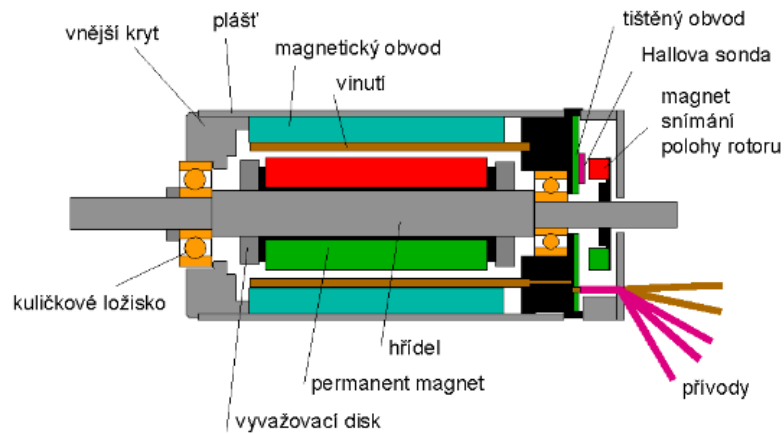
Další změnou v konstrukci motoru bylo prohození vinutí cívk rotoru a permanentního magnetu statoru. Tato změna vedla k tomu, že se cívka statoru rozdělila na tři. Uvedené rozdělení způsobuje komplikaci řízení.

Vnější magnetický obvod motoru slouží k uzavírání magnetického pole, proto materiál, který se pro tento magnetický obvod použije, musí minimalizovat hysterezní ztráty způsobené přemagnetováním železa a ztráty vířivými proudy. Na obrázku 1.5 je představena konstrukce EC motoru.

Díky uvedeným změnám vznikla modifikace motoru, kde rotor je vnější částí a stator je vnitřní částí motoru. Uvedenou modifikaci je možné vidět na obrázku 1.6.

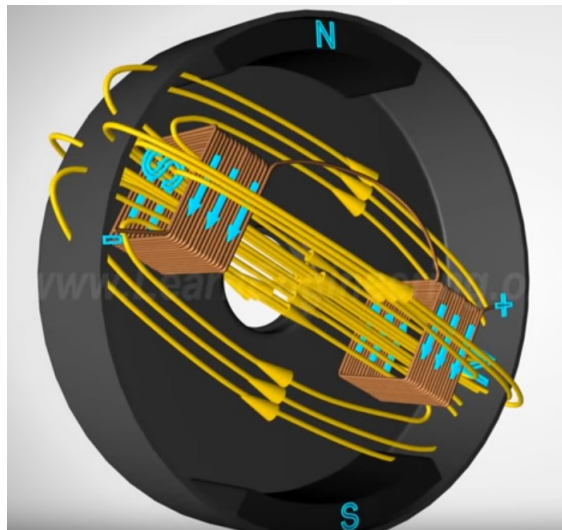
Princip

EC motory využívají stejné fyzikální principy jako DC motory. Jenže na rozdíl od kartáčového motoru tento krát Lorenzova síla působí na permanentní magnet, který je součástí rotoru. Stator má na sobě tři vinutí, která se postupně spínají v kladném



Obr. 1.5: Typická konstrukce EC motoru [2]

a záporném směru. Tím pádem vytvářejí proměnné magnetické pole uzavírající se vnějším magnetickým obvodem. Na obrázku 1.6 je představen uvedený fyzikální princip.



Obr. 1.6: Princip BLDC motoru [6]

Životnost

Životnost bezkartáčových motorů je dána opotřebením kuličkových ložisek. Doba životnosti tohoto elementu se pohybuje kolem několika desítek tisíc hodin, jinak by byla zaručena prakticky nekonečná doba života takových motorů.

1.2 Řízení stejnosměrného motoru

Existují dva způsoby řízení stejnosměrného motoru:

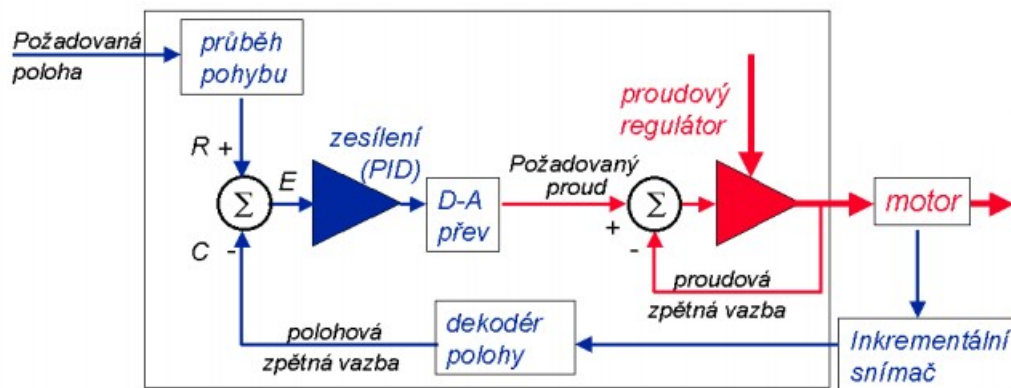
- řízení s otevřenou smyčkou,
- řízení s uzavřenou smyčkou.

Řízení motoru s otevřenou smyčkou znamená, že řídicí modul motoru nepoužívá zpětnou vazbu a při řízení nepočítá s aktuálním stavem motoru. Řízení motoru s uzavřenou smyčkou je řízení s použitím zpětné vazby. Řídicí modul motoru snímá stav motoru, a na základě získané informace a požadavku nadřízeného systému provádí řízení.

Zpětná vazba je realizována snímačem otáček nebo polohy, který se umístí na hřídel motoru. Existují motory, které mají snímače už zabudované uvnitř (často se aplikuje u BLDC motorů). Připojení snímače přímo na hřídel motoru vede ke zjednodušení dynamiky řízení. Řídicí modul motoru nebude muset počítat s mechanickou vůlí převodovky a pružností spojení.

1.2.1 Řízení polohy

Tento způsob řízení byl zvolen z toho důvodu, že navrhovaný modul by měl nejen řídit motory, ale i řešit polohovací úkoly. Řízení polohy je možné jen modulem s možností připojení zpětné vazby. Zpětná vazba může být tvořena např. enkodérem s rozlišením 1024 pulzů na otáčku. Pomocí snímače polohy se dá zjistit rychlost otáčení rotoru.



Obr. 1.7: Uzavřená regulační smyčka řízení polohy [2]

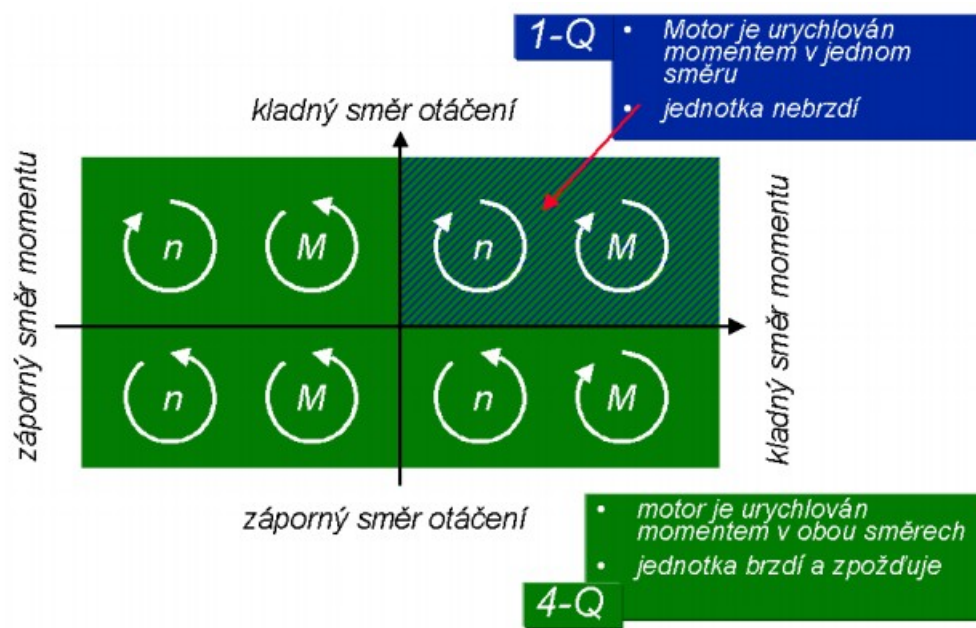
Na obrázku 1.7 je nakreslen model řízení se zpětnou vazbou. Uzavřená regulační smyčka představuje mezi sebou uzavřený kruh sériově zapojených regulátorů, soustavy (motor) a zpětné vazby (inkrementální snímač). Sériově zapojené regulátory mohou být rozdělené na řídicí regulátor (polohový regulátor) a řízený regulátor

(proudový regulátor). Takové zapojení se dělá proto, aby byla regulace zpřesněná především u soustavy s dopravním zpožděním.

Uvedený model řízení funguje následujícím způsobem. Modul dostane požadavek na změnu pozice. Pomocí inkrementálního snímače získá informace o aktuální pozici, a na základě uvedených hodnot vypočítá polohovou regulační odchylku. Tato odchylka se zpracuje polohovým regulátorem, a vznikne požadovaná hodnota proudu. Z požadované a zpětnovazební hodnoty proudu se vypočítá proudová regulační odchylka, která se zpracuje proudovým regulátorem. Vznikne akční veličina proudu, která bude řídit motor, a tím pádem i polohu.

1.2.2 Kvadrantové řízení

Kvadrantové řízení říká, co modul řízení motoru umí z hlediska možnosti řízení. Řídicí moduly se rozdělují na jednokvadrantové (1Q), dvojkvadrantové (2Q) a čtyřkvadrantové (4Q). Význam tohoto parametru se dá pochopit z obrázku 1.8.



Obr. 1.8: 1Q a 4Q řídicí jednotka [2]

1Q řídicí jednotky jsou schopné točit rotorem motoru jen v jednom směru pohybu. Směr se volí zapojením přívodu motoru.

2Q řídicí jednotky točí rotorem motoru ve dvou směrech. Směr se dá nastavovat pomocí přepínače nebo přivedeným signálem. Tento typ jednotek není schopen provádět brzdění. Zpomalení nebo zastavení je zajištěno pouze mechanickým odporem motoru.

Existuje ale možnost rychlého zastavení motoru řízeného modulem 1Q nebo 2Q. To se dělá zkratováním vinutí motoru.

4Q řídicí jednotky jsou schopné točit rotorem motoru ve všech směrech. Uvedený typ modulu tudíž umožňuje provádět zrychlení a brzdění. Ve chvíli, kdy nejsou přivedené řídicí signály, motor bude zůstat v klidu.

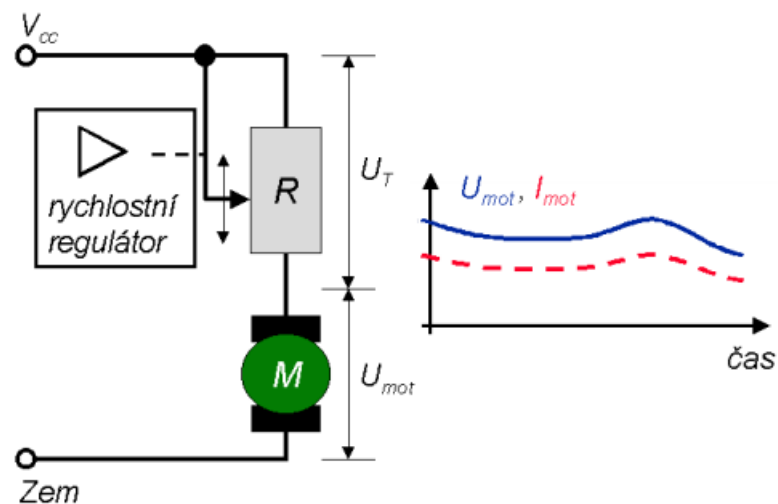
Návrh modulu pro řízení stejnosměrných motorů je orientován na vytvoření jednotky, která bude umět 4Q řízení.

1.3 Regulátory

V této kapitole se bude pojmem *regulátor* zabývat z hlediska hardwaru. Proudový regulátor je koncovým stupněm každého modulu řízení motoru. V kapitole 1.2 regulátory jsou bloky pro zpracování informací, které provádí výpočet akčních veličin. Na základě této veličiny vzniknou řídicí signály, které následně řídí motor. Z hlediska hardwarového provedení existují dva typy regulátorů. Uvedené regulátory v koncovém stupni používají výkonové tranzistory.

1.3.1 Lineární proudový regulátor

Lineární proudový regulátor představuje jeden výkonový tranzistor, zapojený do série s motorem. Napájecí napětí se rozdělí na dva úbytky: úbytek na motoru a úbytek na výkonovém tranzistoru, kde se úbytek napětí na tranzistoru projeví vyzařováním tepla. Lineární proudový regulátor je znázorněn na obrázku 1.9.

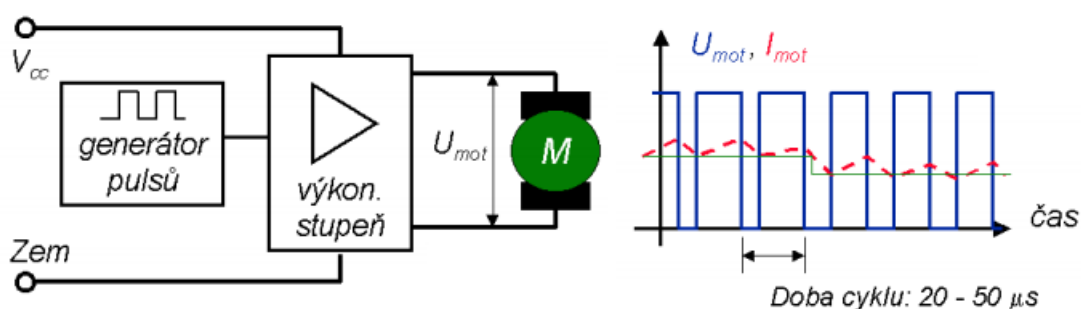


Obr. 1.9: Lineární proudový regulátor [2]

Tento typ regulátoru nevyžaduje složitou elektroniku, má slabé elektromagnetické rušení a neomezuje indukčnost motoru. Nevýhodou tohoto regulátoru jsou ale velké tepelné ztráty na výkonovém tranzistoru. S velkou pravděpodobností se bude muset řešit chlazení tranzistoru.

1.3.2 Pulzní proudový regulátor

Pulzní proudový regulátor používá pro řízení motoru pulzně šířkovou modulaci. Výkonové tranzistory, které ovládají motor, pracují ve dvou stavech: zcela otevřen a zcela zavřen. Díky tomu jsou tepelné ztráty na tranzistorech minimální. Pulzní proudový regulátor je znázorněn na obrázku 1.10.



Obr. 1.10: Pulzní proudový regulátor [2]

Zvlnění proudu se dá zamezit přidáním indukčnosti. Musí se to však dělat opatrně, jelikož nedostatečná indukčnost může způsobit přehřátí vinutí motoru i při běžných proudech. Proto je nutno dávat pozor na hodnoty přidané indukčnosti.

Pulzní proudový regulátor vyžaduje složitou elektroniku, má vysoké energetické ztráty při zastavení motoru, omezuje indukčnost motoru a vyvolává elektromagnetické rušení. Na druhou stranu výhodou tohoto provedení je vysoká účinnost a nízké tepelné ztráty i při vysokých proudech.

Po porovnání uvedených regulátorů se rozhodlo, že koncový stupeň budoucího modulu pro řízení motorů bude tvořen pulzním proudovým regulátorem.

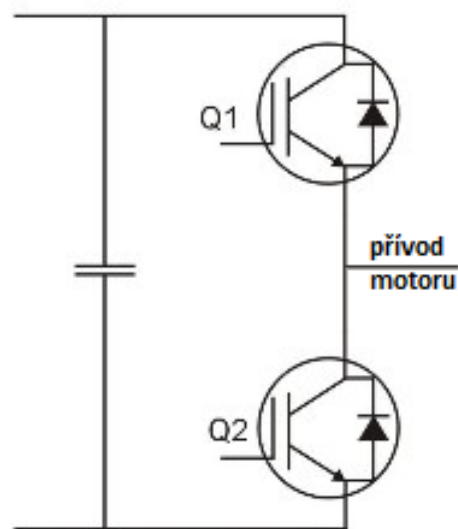
1.4 Zapojení pro ovládání stejnosměrného motoru

Stejnoseměrný motor se ovládá výkonovými tranzistory, protože součástkami můžou téct proudy až do desítek ampér, a napájecí napětí motorů se může pohybovat v rozsahu od 12 V až do 60 V.

Na konci podkapitoly 1.3.2 bylo rozhodnuto založit návrh na použití pulzního proudového regulátoru. Koncový stupeň tohoto regulátoru představuje můstkové zapojení výkonových tranzistorů MOSFET nebo IGBT.

1.4.1 Poloviční H-můstek

Poloviční H-můstek je tvořen zapojením dvou výkonových tranzistorů do série, jak je znázorněno na obrázku 1.11. Uvedený můstek je základním stavebním prvkem pro ovládání stejnosměrných motorů. Kombinace těchto můstků tvoří koncový stupeň pulzního proudového regulátoru pro řízení kartáčového nebo bezkartáčového motoru.



Obr. 1.11: Poloviční H-můstek

Ovládání polovičního H-můstku se provádí pomocí signálů PWM, které budou otevírat a zavírat výkonové tranzistory. Při použití můstkového zapojení se vždy otevírají jenom dva tranzistory. Zbývající tranzistory zapojení musí být uzavřeny. Podle použitého motoru bude volen způsob jeho napájení a počet polovičních H-můstků zapojených v koncovém stupni.

1.5 Senzory polohy

Aby budoucí modul pro řízení motoru mohl řešit polohovací úkoly, musí být zajištěna zpětná vazba, která bude udávat informaci o aktuální poloze hřídele rotoru. Zpětná

vazba představuje senzor, který bude informovat mikroprocesor modulu o tom, jak se mění pozice.

Kartáčový motor může jako zpětnou vazbu použít enkodér nebo resolver. Bezkartáčový motor používá stejné senzory, dále může mít v sobě tento typ motoru zabudované tři Hallové sondy, které můžou být použity pro řešení polohovacích úkolů.

1.5.1 Enkodéry

Enkodéry jsou založeny na různých fyzikálních principech. Rozdělují se na enkodéry absolutní a relativní. Rozdíl mezi nimi je v tom, že absolutní enkodéry při zapnutí informují modul o aktuální pozici a mohou okamžitě začít vykonávat polohovací úkoly. Relativní enkodéry se potřebují nejdříve uvést do definované (referenční) pozice, a pak mohou začít řešit polohovací úkol. Přesnost enkodéru je závislá na hustotě dělení měřicí části. Snímač polohy se jako samotný díl připojuje k hřídeli motoru. Dále budou popsány jednotlivé typy enkodérů.

Digitální inkrementální enkodér

Tento typ enkodéru spadá do relativních kontaktních snímačů polohy. Změna pozice se posílá do řídicí jednotky ve tvaru obdélníkového signálu z několika kanálů, nebo může být odeslána přes sériovou linku. Dva kanály jsou stejné a mají posun pulzů vůči sobě o 90 stupňů. Jeden kanál obsahuje nulový bod představující jenom jeden pulz. Tento pulz se používá pro nastavení referenční pozice. Vyhodnocením signálů se mohou získat informace o rychlosti otáčení rotoru, směru otáčení a poloze.

Digitální absolutní enkodér

Tento enkodér je absolutním snímačem polohy. Poloha rotoru je zakódována pomocí binárních kódovacích kroužků. Uvedené kroužky vytváří *datové slovo*, které nese informaci o poloze během jedné otáčky. Enkodér vrací informace o pozici ve tvaru bitové sekvence odeslané s hodinovým signálem za použití různých protokolů.

Tyto enkodéry se rozdělují na jednosměrné a obousměrné. Jednosměrné digitální absolutní enkodéry jsou schopné určit pozici během jedné otáčky (ihned je informace o poloze). Obousměrné enkodéry potřebují pro určení pozice několik otáček rotoru.

Magnetické enkodéry

Magnetické enkodéry fungují na principu snímání změn magnetického toku permanentního magnetu Hallovo sondou nebo magnetickým odporovým senzorem. Získaná data se zpracují v elektronice enkodéru, a pak se posílají ve tvaru pulzního signálu nebo bitové sekvence.

Existují různé typy magnetických enkodérů, které se dále rozdělují na programovatelné a neprogramovatelné, absolutní a inkrementální enkodéry, dle počtu pulzů na otáčku atd.

Optické enkodéry

Tento typ enkodérů funguje na principu průchodu nebo odrazu světelného paprsku tvořeného LED diodou. Světelný obraz je tvořen kotoučem, který má přesné otvory pro zajištění průchodu světla nebo reflektory pro odraz paprsků. Přijímače jsou tvořené fototranzistory nebo fotorezistory, které převedou signál světlo/tma na elektrický proud. Výsledný proud se zesílí a odešle se ve tvaru pulzů, nebo pomocí sériové komunikační sběrnice.

Induktivní enkodér

Induktivní enkodér představuje diferenciální transformátor, kde primární cívka generuje střídavé pole o frekvenci 1 až 2 MHz. Dle Faradova zákona se v sekundárních cívkách zapojených v diferenciální konfiguraci indukují napětí. Podmínkou indukování napětí v cívkách je, že musí existovat vazba s primární cívkou, což je zajištěno feromagnetickým předmětem.

Porovnání typů enkodérů

Při zvolení enkodéru pro motor se musí vzít v úvahu parametry snímače, okolní prostředí (rušení) a místo instalace (rozměry enkodéru a volný prostor kolem motoru). V tabulce 1.1 jsou vyznačeny výhody a nevýhody různých typů enkodérů. Označení (ano) znamená podmíněnou použitelnost. Zkratky QUAD, MEnc, MR a EASY se týkají magnetických enkodérů, MILE jsou induktivní enkodéry.

1.5.2 Resolver

Resolver představuje rotující transformátor, který se používá pro detekci úhlu natočení, a je vyrovnán s magnetickým polem rotoru motoru. Tento senzor se skládá z jedné primární a dvou sekundárních cívek, které mezi sebou mají posun o 90 stupňů. Primární vinutí je napájeno střídavým signálem a točí se spolu s rotorem. V sekundárních cívkách se indukují napětí. Amplituda sekundárních napětí je dána funkcemi $\sin\varphi$ a $\cos\varphi$, kde φ je úhel natočení rotoru. Na základě amplitud se vyhodnocuje aktuální stav.

Tab. 1.1: Tabulka enkodérů firmy MAXON [7]

	QUAD	MEnc	MR	EASY	MILE	optical
vysoká rychlost	ano	ano	ano	ano	(ano)	
nízká rychlost			(ano)	(ano)	ano	ano
Přesná pozice			(ano)	(ano)	(ano)	ano
možnost sériové linky			ano	ano	ano	ano
možnost indexu kanálu			ano	ano		ano
kompaktní konstrukce	ano	(ano)	ano	ano	ano	
odolnost proti nečistotám	ano	ano	ano	ano	ano	
odolnost proti mag. poli	(ano)	(ano)		(ano)	ano	ano
odolnost proti radiaci		ano				

1.6 Komunikace

Průmyslové sběrnice mají velký význam v průmyslových aplikacích, jsou nástrojem pro sběr, distribuci a vyhodnocení dat různého informačního charakteru. Častým příkladem moderních sběrnic pro komunikaci v průmyslu jsou CAN, ProfiBUS a sběrnice, které jsou založeny na technologii Ethernet. Nejpoužívanějšími z této kategorie jsou ProfiNET a EtherCAT.

1.6.1 CAN

Původním účelem sběrnice CAN bylo nasazení v osobních automobilech. Časem byla sběrnice rozšířena o další aplikační možnosti na jiných technických zařízeních. Sběrnice umožňuje spolehlivé decentralizované řízení v reálném čase s rychlostí komunikace až 1 Mb/s.

Aby byla zajištěna kompatibilita nasazení, používá CAN vrstevný referenční model ISO/OSI a má definované tyto funkční vrstvy: objektovou CAN, transportní CAN a fyzickou vrstvu. Transportní a objektová vrstva odpovídají a obsahují služby a funkce spojové (linkové) vrstvy modelu ISO/OSI, jehož struktura je představena na obrázku 1.12.

Objektová vrstva má za úkol zajistit kompatibilitu s hardwarem obsluhovaným aplikační vrstvou ISO/OSI modelu a zajistit tok prioritních dat s možností vyhledávání v reálném čase. Transportní vrstva zajistí přenos dat s minimalizací a signalizací chyb, zkontroluje naplnění datového řetězce, provede arbitrážní procedury mezi vysílačem a přijímačem, řeší kolizní situace a zajistí adresování cílových zařízení, bitovou a rámcovou synchronizaci atd. Fyzická vrstva má na starosti propojení modulů se sběrnicí CAN.



Obr. 1.12: Struktura ISO/OSI

CAN je schopna rozlišovat prioritu zpráv, zajišťovat minimální prodlevy při režijních činnostech, řešit časovou synchronizaci, detekovat chyby komunikace a provádět automatické opakované odesílání poškozeného rámce.

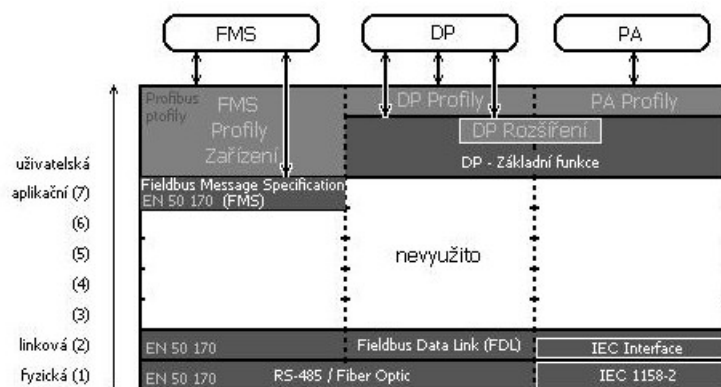
Maximálně může být dle doporučení zapojeno 30 uzlů. Přenos zprávy probíhá tak, že před začátkem komunikace mají být všechny uzly synchronizovány a připraveny vzorkovat ve stejnou dobu. Pak začne vysílání. Každý uzel přijímá zprávu, ale reagovat bude jen ten, kterému tato zpráva patří. Z toho vyplývá, že topologie zapojení uzlů na sběrnici je linková.

1.6.2 ProfiBUS

Sběrnice ProfiBUS se již nenazývá sběrnicí, ale systémem sběrnic. Byla vyvinuta kvůli zajištění komunikace mezi moduly různých výrobců nebo různými automatizačními zařízeními. ProfiBUS, stejně jako CAN, vychází z modelu ISO/OSI a používá fyzickou, linkovou a aplikační vrstvu. Systém sběrnic je v nejnižších vrstvách navázán na sensorový systém AS-Interface, což je vrstva sensorů a akčních členů. V aplikační vrstvě je ProfiBUS navázán na ProfiNET, který umožňuje přenos libovolných dat přes počítačové sítě.

Jak je vidět z obrázku 1.13, fyzická vrstva podporuje několik sběrnic pro přenos dat (RS-485, optické vlákno a proudová smyčka). Linková vrstva definuje mechanismus přístupu různých modulů na komunikační médium (linková topologie sběrnice), a zajišťuje tvorbu zprávy na bitové úrovni včetně vytvoření kontrolních částí. Aplikační vrstva realizuje komunikaci z hlediska uživatele.

Aby byl zajištěn přenos dat, přidávají se k paketu části záhlaví a zápatí definující stavy zařízení, adresy vysílajících a přijímajících stanic, čísla přijatých a odeslaných



Obr. 1.13: Model ProfiBUSu [12]

paketů, a mnoho dalších informací. Data, která řídí kontinuální procesy, jsou přenášena cyklicky v síti. Měřicí data odpovídající asynchronnímu hlášení a nepravidelná konfigurační data jsou přenášena z hlediska nutnosti.

Existují tři varianty standardu ProfiBUS. Každá z nich má stanovenou maximální rychlost komunikace. Nejrychlejší a nejrozšířenější variantou je standard ProfiBUS Decentralized Periphery s maximální rychlostí přenosu dat 12 Mb/s.

1.6.3 Průmyslové Ethernety

Průmyslový Ethernet propojuje různé jednotky a moduly, zajišťuje komunikaci mezi nimi a řídicím systémem. Oproti obyčejné ethernetové síti průmyslová síť musí splňovat následující požadavky:

- splnění požadavku komponent podílejících se na komunikaci dle priority,
- včasné reagování na hlášení,
- minimalizaci kolísání doby odezvy,
- stabilitu hardwaru a softwaru,
- odolnost proti mechanickým vlivům a vlhkosti,
- možnost přístupu uživatele.

Průmyslový Ethernet je rozšířen o protokoly, které dovolují efektivnější výměnu dat mezi více než dvěma moduly. Objemy přenášených dat se pohybují od jednotek bajtů až do stovek megabajtů. Díky tomu, že tyto komunikace jsou založeny na rozhraní Ethernet, podporují rychlosti přenosu dat 10 Mb/s (10BaseT), 100 Mb/s (100BaseTX) a 1 Gb/s (1000BaseTX), a jsou založeny na standardizovaném modelu OSI.

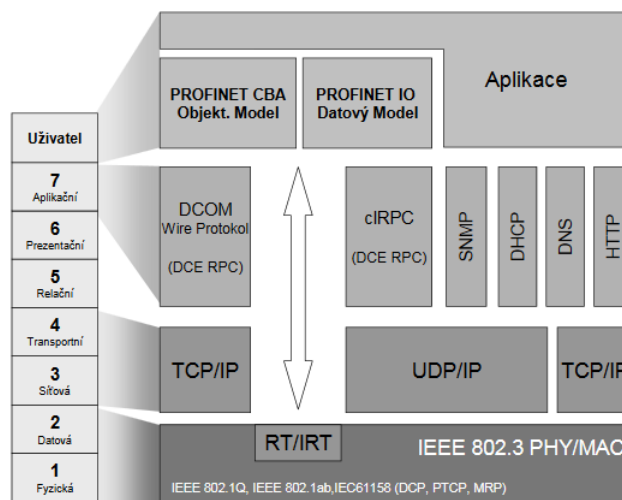
Každá vrstva si vyměňuje informace se sousedními vrstvami prostřednictvím struktury, která se nazývá *protokol data unit* (protokolem datových jednotek). Při

odesílání dat se zpráva poskládá na aplikační vrstvě, a postupně začne procházet každou vrstvou dolů až do nejnižší. Tam vznikne bitový řetězec odesílaný přijímacímu zařízení. Při přijímání dat to funguje naopak.

Uplatnění těchto sběrnic je v oblasti řízení rychlých procesů, jako je řízení polohy a pohybu za použití elektrických pohonů, ovládání elektroniky stroje atd.

ProfiNET

ProfiNET představuje kombinaci výhod standardu ProfiBUS a Ethernet. Je flexibilní síťovou topologií, která je rozšířena o topologií typu hvězda, strom a kruh. ProfiNET pracuje na principu ProfiBUSu. Každá profinetová jednotka funguje jako opakovač (repeater) a pomocí rozbočovače (switch) se dají provádět úpravy na topologii. ProfiNET existuje ve dvou provedeních ProfiNET CBA a ProfiNET IO. Model ProfiNET sběrnice je vidět na obrázku 1.14.

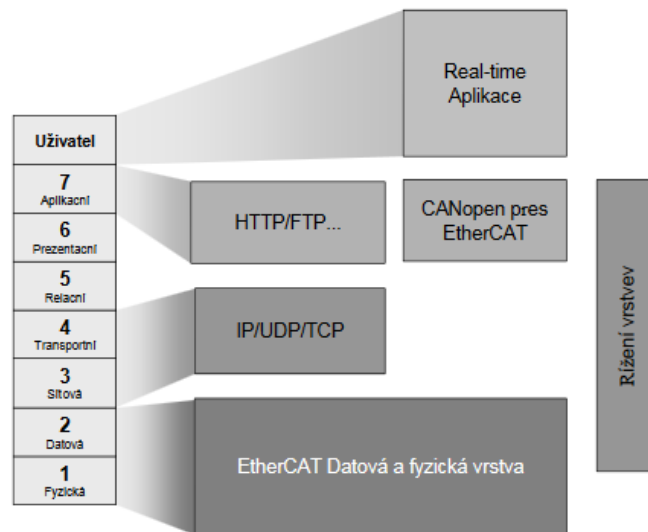


Obr. 1.14: Model ProfiNETu [9]

EtherCAT

EtherCAT sběrnice má unikátní princip funkčnosti založený na technologii zpracování dat za chodu (processing on the fly). Princip spočívá v tom, že EtherCAT zařízení tvoří logický kruh, kde se začíná a končí masterem. Vytvořený datový rámec je rozdělen na datové bloky. Každá jednotka v kruhu má svůj blok. Rámec začne procházet jednotkami a vyměňovat si vstupní/výstupní data za běhu. Výsledkem takového průchodu je, že každá jednotka má nová data, a master má procesní data všech zařízení v síti. Jediné zpoždění, které se objevuje na sběrnici, je dáno zápisem a čtením dat ethercatového řetězce.

EtherCAT využívá technologii distribuovaných hodin. Díky tomu se dá provádět synchronní nastavení výstupních hodnot celé sítě s odchylkou pod $1 \mu\text{s}$, což je ideální pro aplikace s pohybem pohonů nebo integraci měřících úloh.



Obr. 1.15: Model EtherCATu [9]

Sběrnice provádí adresování jednotek automaticky a je schopna rozpoznat topologii navržené sítě.

Díky uvedeným výhodám EtherCAT sběrnice byla vybrána pro komunikace, která zajistí spojení budoucího modulu s nadřazeným systémem. Detailněji bude EtherCAT popsán v sekci 3.

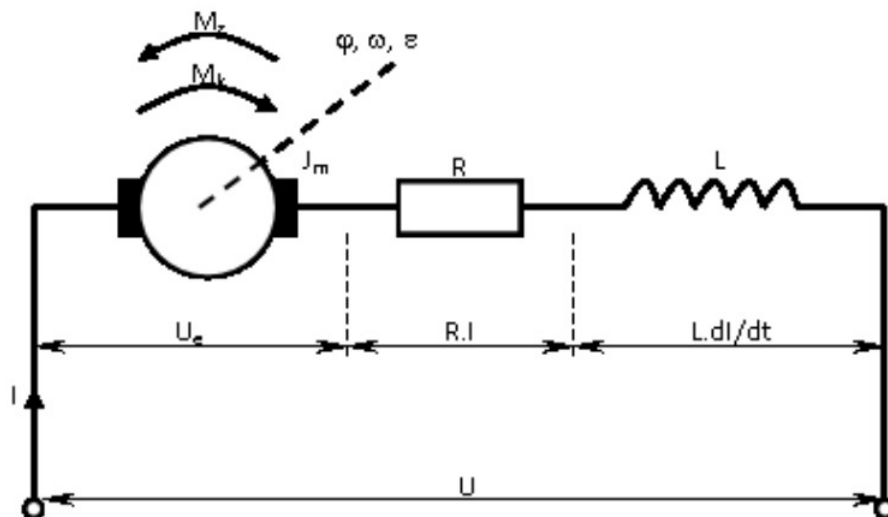
2 Návrh

Následující kapitoly se zabývají návrhem modulu pro řízení stejnosměrných motorů. Návrh byl rozdělen na několik částí:

- tvorba matematického modelu stejnosměrného motoru - základ simulace,
- návrh modelu řízení motoru - simulace chování motoru,
- návrh elektrického schématu - hardware,
- návrh DPS - hardware.

2.1 Matematický model motoru

Aby se dalo zkoumat chování stejnosměrného motoru, musí se udělat jeho matematický model. Primární funkcí modulu pro řízení motorů bude ovládání kartáčových motorů, vytvoří se model kartáčového motoru.



Obr. 2.1: Elektrické schéma stejnosměrného motoru [18]

Na obrázku 2.1 je představeno náhradní schéma motoru. Z uvedeného modelu se odvodí vzorec pro napětí:

$$U = U_e + R * i + L * \frac{di}{dt}, \quad (2.1)$$

kde U_e je indukované napětí, R je odpor kotvy a L je indukčnost motoru.

Indukované napětí se dá rozepsat následujícím způsobem:

$$U_e = c * \Phi * \omega = k_v * \omega, \quad (2.2)$$

kde c je konstrukční konstanta motoru, Φ je magnetický indukční tok, ω je úhlová rychlost a k_v je napěťová konstanta (Back-EMF konstanta).

Napěťovou konstantu výrobce často v technické specifikaci motoru neudává. Uvedená konstanta je významná z důvodu toho, že motor má indukčnost. Při otáčení hřídele se indukují Back-EMF napětí, které působí proti přiloženému napětí. Toto napětí se snaží bránit změnám proudu. Aby model co nejvíce odpovídal realitě, musí se nalézt ekvivalent napěťové konstanty.

Motor vytváří elektrický moment, který je proporcionální proudu:

$$M_e = k_m * i, \quad (2.3)$$

kde k_m je momentová konstanta, kterou výrobce uvádí v technické specifikaci motoru.

Když se bude vycházet z předpokladu, že v motoru nedochází k elektromagnetickým ztrátám. Mechanická síla se může porovnat s elektrickou energií rozptýlenou v armatuře motoru:

$$M_e * \omega = U_e * i, \quad (2.4)$$

$$k_m * i * \omega = k_v * i * \omega. \quad (2.5)$$

Z rovnice 2.5 plyne, že místo hodnoty napěťové konstanty se může použít hodnota momentové konstanty. Úpravou rovnice 2.1 se získá nová napěťová rovnice motoru:

$$U = k_v * \omega + R * i + L * \frac{di}{dt}, \quad (2.6)$$

Výpočet dynamického momentu se dělá na základě rovnice rovnováhy momentů:

$$M = J * \frac{d\omega}{dt} + M_0 + k_f * \omega, \quad (2.7)$$

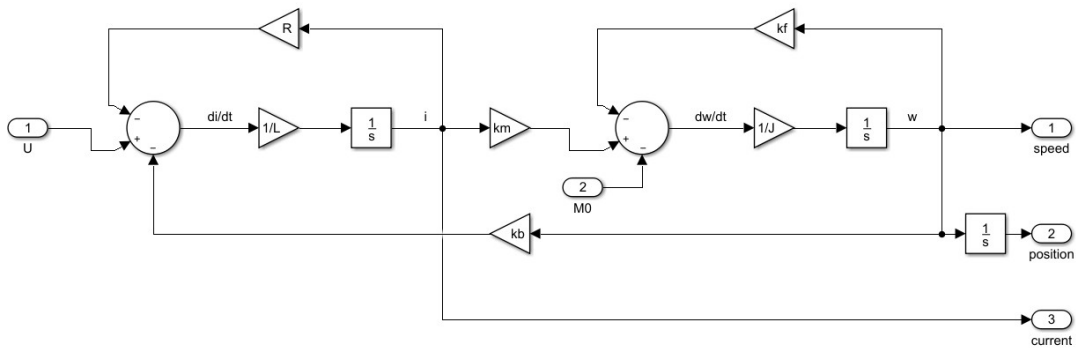
kde J je moment setrvačností motoru, M_0 je vnější zatěžovací moment, M je dynamický moment na hřídeli a k_f je koeficient třecí síly. Z důvodu toho, že po rozběhu motoru třecí síla bude mít konstantní hodnotu, v matematických modelech se zanedbává. Kvůli přiblížení modelu realitě, se s tímto koeficientem bude počítat. Třecí síla bude mít malou přesně definovanou hodnotu.

Dynamický moment na hřídeli motoru je ve své podstatě vnitřní elektromagnetický moment, který vzniká z důvodu průchodu proudu. Proto se dá rovnici 2.7 upravit do tvaru:

$$k_m * i = J * \frac{d\omega}{dt} + M_0 + k_f * \omega, \quad (2.8)$$

Aby se dalo vytvořit model kartáčového motoru v MATLAB Simulink, musí se převést rovnice 2.6 a 2.8 na diferenciální tvar:

$$\frac{di}{dt} = \frac{1}{L} * (U - R * i - k_v * \omega), \quad (2.9)$$



Obr. 2.2: Matematický model stejnosměrného kartáčového motoru

$$\frac{d\omega}{dt} = \frac{1}{J} * (k_m * i - M_0 - k_f * \omega). \quad (2.10)$$

Model popsaný rovnicemi 2.9 a 2.10 je zobrazen na obrázku 2.2.

Aby se dala provádět simulace, musí být motor identifikován. Po zkoumání katalogu firmy MAXON byl zvolen kartáčový motor DCX35L.

2.2 Regulace

Regulace se zabývá řízením soustavy a snaží se, aby se celý systém choval stabilně. Návrh regulace se dělá, aby vznikla představa, jak by měl fungovat budoucí modul.

Regulace se realizuje regulátorem a zpětnou vazbou. Regulátor se skládá z jedné a více složek, na základě kterých se stanoví typ regulátoru. Existují tři různé složky, a každá z nich má určitou funkci:

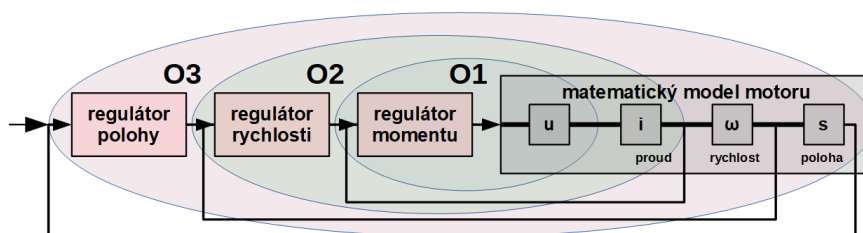
- P - proporcionální složka - zesiluje nebo zeslabuje signál a mění dobu odezvy,
- I - integrační složka - eliminuje chyby odchylky a ustaluje signál na požadované hodnotě,
- D - derivační složka - zrychluje odezvu regulátoru.

Složitost regulátoru a regulace závisí na soustavě, kterou je potřeba regulovat. Protože navrhovaný modul bude řešit polohovací úkoly, bude navrhovat regulace pro řízení polohy. P-regulátor sám o sobě není schopen zcela vyrovnat odchylku mezi požadovanou a aktuální hodnotou polohy. Aby se dalo dosáhnout nulové regulační odchylky, přidá se složka I, která zkoriguje zbylou část odchylky. Tím vznikne PI-regulátor. Jestli je požadována velice rychlá odezva regulátoru a PI-regulátor nebude stíhat regulovat soustavu, přidá se složka D, která zrychlí regulaci. Tím se zvýší složitost systému a pravděpodobnost nestability.

Model motoru z kapitoly 2.1 je soustavou, pro kterou se bude navrhovat řízení. Motor je mechanická soustava s určitými fyzickými omezeními (odezva je v ms),

výsledná regulace nesmí být moc rychlá. Nutnost použití PID-regulátoru odpadá.

Regulace se bude navrhovat s použitím P- a PI-regulátoru. Z modelu motoru se dá získat informace o poloze, rychlosti otáčení a proudu. Každý z těchto parametrů může mít vliv na hodnotu výsledné akční veličiny. Z tohoto důvodu se budou navrhovat tři regulátory, které budou zapojené v sérii. Každý regulátor upraví akční veličinu na základě určitého parametru. Obrázek 2.3 prezentuje strukturu vytvořeného řízení.



Obr. 2.3: Vytvořená regulace

Aby řízení odpovídalo skutečnosti, do regulátorů se mají přidat bloky saturací. Tím se zohlední maximální proud a napětí motoru. Bloky saturace přidávají do řízení nelinearitu typu nasycení akční veličiny (windup jev). Daný problém se vyskytuje u regulátorů se složkou I a projevuje se nechtěným chováním.

Nasycením akční veličiny dojde k rozpojení regulační smyčky. Výstup soustavy nebude závislý na vstupu. Regulátor bude pracovat v otevřené smyčce a nekonečně integrovat odchylku. Akční veličina se bude pořád zvětšovat a bude neustále saturovat. Následně se to projeví změnou znaménka regulační odchylky. Způsoby řešení uvedeného problému budou popsány v příští podkapitole.

2.2.1 Návrh regulátorů

Vytvořená regulace se skládá ze tří regulátorů, které jsou zapojené do série. Tyto regulátory se rozdělují na dvě skupiny:

- řídicí regulátor - regulátor polohy (P- a PI-regulátor),
- řízené regulátory - ostatní regulátory (PI-regulátory).

Návrh každého regulátoru se má provádět za přítomnosti všech regulátorů zapojených v sérii. Prvním krokem návrhu je nastavení proporcionální složky na 1 a integrační na 0 u každého regulátoru. Dále se budou měnit parametry regulátoru, který se navrhuje.

Regulátory se budou navrhovat dvěma způsoby:

- metoda Ziegler-Nichols – druhá metoda, výpočetní,
- grafická metoda - nástroj Tuning Tool v prostředí MATLAB Simulink.

Metoda Ziegler-Nichols

Metoda Ziegler-Nichols se použije jenom pro návrh řídicího regulátoru. Tato metoda se zakládá na následujícím vzorci:

$$F_R(p) = K_r * (1 + \frac{1}{T_i * p} + T_d * p), \quad (2.11)$$

kde K_r je zesílení regulátoru, T_i je hodnota periody integrační složky a T_d je hodnota periody derivační složky.

Před začátkem návrhu je nutno zkontrolovat, jestli regulátory mají vynulované parametry, a proporcionální složku mají nastavenou na 1. Dál se začne postupně zvyšovat hodnota zesílení regulátoru polohy a kontrolovat výstup celého systému. Když bude výstup systému představovat netlumené kmity, zvyšování hodnoty zesílení se musí zastavit. Aktuální hodnota zesílení je parametrem K_{krit} . Tomuto parametru se říká hodnota kritického zesílení regulátoru. Pak se z obrazu netlumených kmitů získá informace o periodě signálu. Tímto se získá parametr T_{krit} . Tomuto parametru se říká hodnota kritické periody. Posledním krokem návrhu je výpočet parametrů P- a PI-regulátoru pomocí tabulky 2.1.

Tab. 2.1: Tabulka výpočtu koeficientu dle Ziegler-Nicholse

Regulátory	K_r	T_i	T_d
P	$0,5 * K_{krit}$	-	-
PI	$0,45 * K_{krit}$	$0,85 * T_{krit}$	-
PD	$0,4 * K_{krit}$	-	$0,12 * T_{krit}$
PID	$0,6 * K_{krit}$	$0,5 * T_{krit}$	$0,12 * T_{krit}$

Ze simulace byly získané hodnoty kritického zesílení $K_{krit} = 83,9675$ a kritické periody $T_{krit} = 7,321$ ms. Následně se vypočítaly parametry regulátorů. Rovnice 2.12 a 2.13 prezentují výsledný tvar P- a PI-regulátoru:

$$F_{R-PI-poloha}(p) = 37,7854 * (1 + \frac{1}{6,2229 * 10^{-3} * p}), \quad (2.12)$$

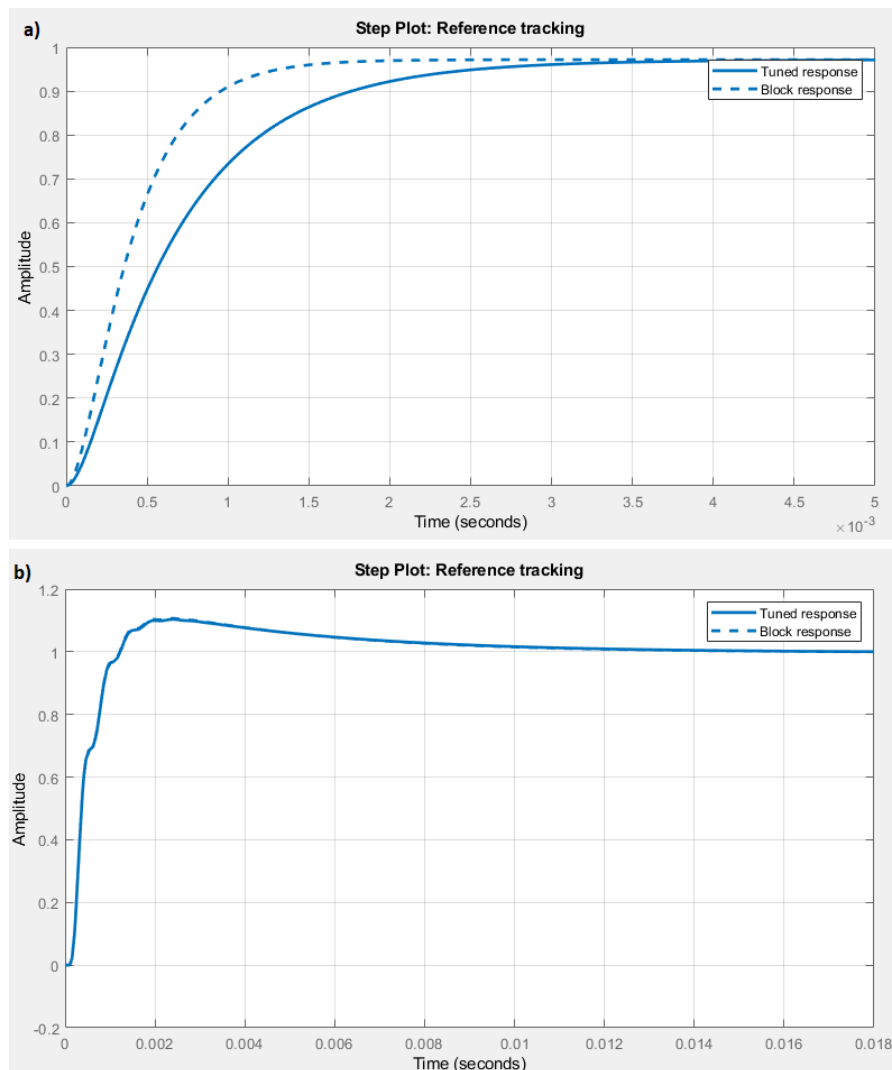
$$F_{R-P-poloha}(p) = 41,98375. \quad (2.13)$$

Tunning tool

Tunning tool je jedním z nástrojů prostředí MATLAB Simulink, který se používá pro doladování regulátoru po návrhu. Někdy se tento nástroj dá použít pro návrh jednoduchého regulátoru.

Řízený regulátor má být jednoduchý a musí dorovnat regulační odchylky. Proto se návrh regulátoru rychlosti a momentu bude provádět nástrojem Tuning Tool.

Tvorba regulátoru se provádí nastavením požadované odezvy. Z obrázku 2.4a je patrné, že regulátor není schopen dosáhnout požadované hodnoty. Použitím jezdců nástroje se začne měnit tvar průběhu charakteristiky, dokud nebude nalezeno vyhovující řešení. Příkladem řešení je obrázek 2.4b. Dál se hodnoty navrženého regulátoru kliknutím tlačítka zkopírují do regulátoru v modelu. Návrh jednoduchého regulátoru je dokončen.



Obr. 2.4: MATLAB Simulink - Tuning Tool - Návrh regulátoru

Doba odezvy řízených regulátorů se musí přiměřeně rovnat době odezvy soustavy. Motor je mechanická soustava. Při vytvoření příliš rychlého regulátoru motor nebude schopen zareagovat na změnu akční veličiny. Může nastat situace, že motor zareaguje, ale systém se bude chovat nestabilně.

Regulátory rychlosti a momentu jsou prezentovány rovnicemi 2.14 a 2.15:

$$F_{R-PI-rychlost}(p) = (0, 6183 + \frac{126,6}{p}), \quad (2.14)$$

$$F_{R-PI-moment}(p) = (1, 293 + \frac{2562,4}{p}). \quad (2.15)$$

Po vytvoření řízených regulátorů se do jejich struktur přidávají bloky saturací. Regulátor rychlosti má zabudovanou proudovou saturaci, protože výstupem tohoto regulátoru je akční veličina proudu. Regulátor momentu má zabudovanou napěťovou saturaci.

Na začátku kapitoly 2.2 byla zmíněna problematika nasycení akční veličiny. Problém nasycení akční veličiny se dá odstranit použitím jedné z následujících metod:

- zpětný přepoččet (back-calculation),
- přerušování integrace (clamping).

Obě metody pracují se strukturou regulátoru. Metoda back-calculation se ve většině případů používá s PID-regulátorem. Tento typ regulátoru se nevyskytuje v regulační smyčce, a proto se použije metoda clamping.

Clamping

Při použití této metody dochází k zastavení integrace za určitých podmínek:

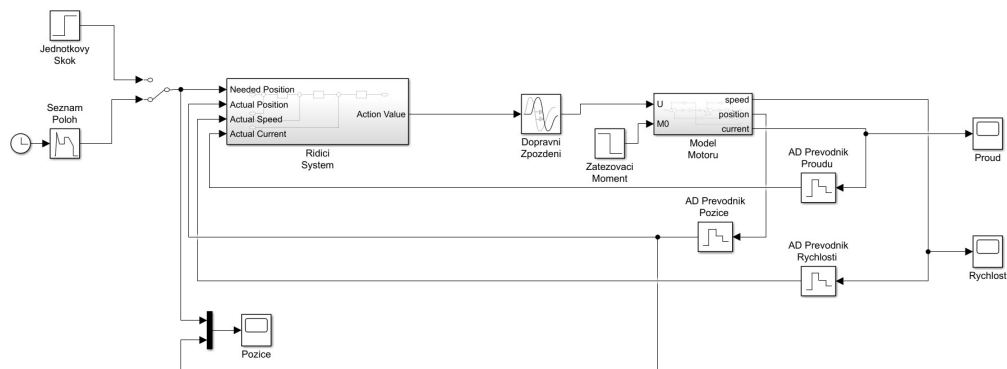
- pokud je hodnota odchylky vysoká, zastaví se integrace (ochrana proti přechodu za hranici saturace),
- pokud je výstup regulátoru za hranicí saturace, integrování se neprovádí.

Takový přístup může způsobit problém, kdy regulace skončí s nenulovou regulační odchylkou. Tomu se dá zabránit zavedením následující podmínky. Jestli výstup regulátoru překročí hranici saturace a znaménko regulační odchylky a výstupu bude stejné, integrování se zastaví. Jestli se znaménko regulační odchylky a výstupu bude lišit, integrovaná hodnota se zmenší.

2.2.2 Testování řízení

V podkapitole 2.2.1 byly vytvořeny dva regulátory polohy. Díky tomu vznikají dvě nezávislé regulace, které se dají porovnat.

Aby se dala regulace testovat, musí se vytvořit testovací zapojení systému s ohledem na reálné podmínky. Motor je spojitá soustava, pro kterou byly vytvořeny spojitě regulátory. V reálných podmínkách se motor ovládá řídicí jednotkou, která pracuje s diskrétními hodnotami. Tudíž je řídicí jednotka regulátorem, který pro svou činnost potřebuje převádět analogové signály na diskrétní a naopak. Proto bude zapojení zatíženo bloky, které budou modelovat různé typy zpoždění.



Obr. 2.5: Zapojení systému

Na obrázku 2.5 je nakresleno testovací zapojení. Bloky uvedeného zapojení mají následující význam:

- model motoru – matematický popis motoru,
- řídicí systém – sériové zapojení regulátorů,
- zatěžovací moment – moment, který začíná působit v určitou dobu,
- převodník AD – simulace převodníků AD s různými periody vzorkování (proud se vzorkuje 4 krát častěji, perioda vzorkování proudu $T_{vz} = 2 \text{ us}$),
- dopravní zpoždění – časové zpoždění, které je způsobeno mikroelektronikou a převodníky DA,
- jednotkový skok – jednotkový skok (používal se pro návrh regulátorů),
- seznam poloh – řetězec požadavků o změnu polohy.

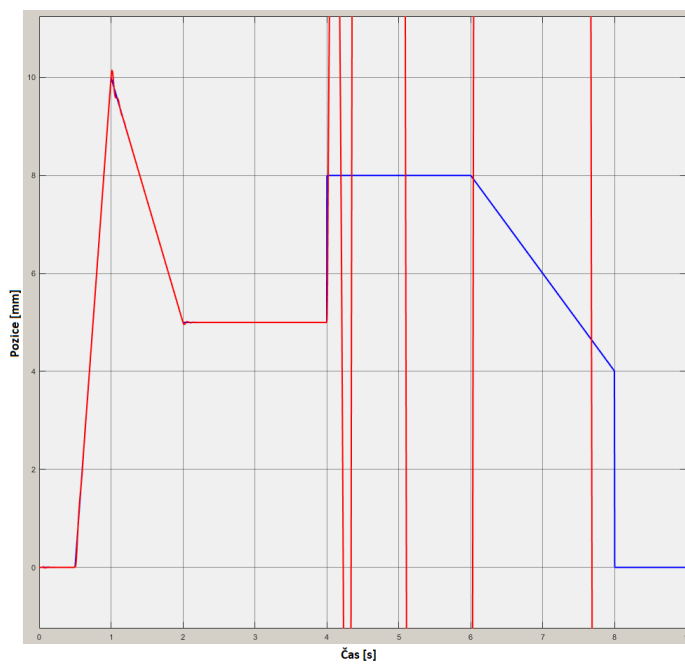
Seznam poloh je tvořen takovým způsobem, aby se otestovalo chování systému v různých situacích:

- plynulá změna polohy,
- skoková změna polohy,
- přechod z plynulé změny polohy na skokovou.

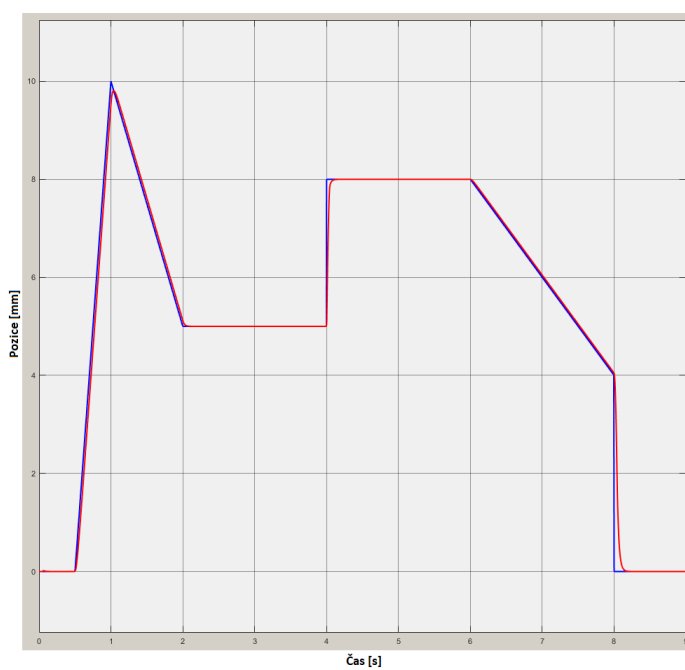
Na obrázku 2.6 je nakresleno chování systému s PI-regulátorem polohy. Regulátor dobře pracuje v situacích plynulé změny polohy. Takže je schopen eliminovat chyby způsobené různými zpožděními. V situaci skokové změny polohy PI-regulátor začíná být nestabilní, protože se regulační odchylka integruje třikrát. PI-regulátor polohy se může použít v polohovacích aplikacích za předpokladu, že řídicí jednotka bude mít složitější řízení. Firmware bude rozšířen o funkci převodu skokové změny polohy na lineární.

Obrázek 2.7 prezentuje chování systému s P-regulátorem polohy. Systém se chová stabilně a vždy splní požadavek změny pozice.

Po porovnání dvou regulací bylo rozhodnuto použít model řízení s P-regulátorem polohy. Ve firmwaru se bude implementovat jenom P-regulátor polohy. Je to z toho



Obr. 2.6: Odezva systému - PI-regulátor polohy



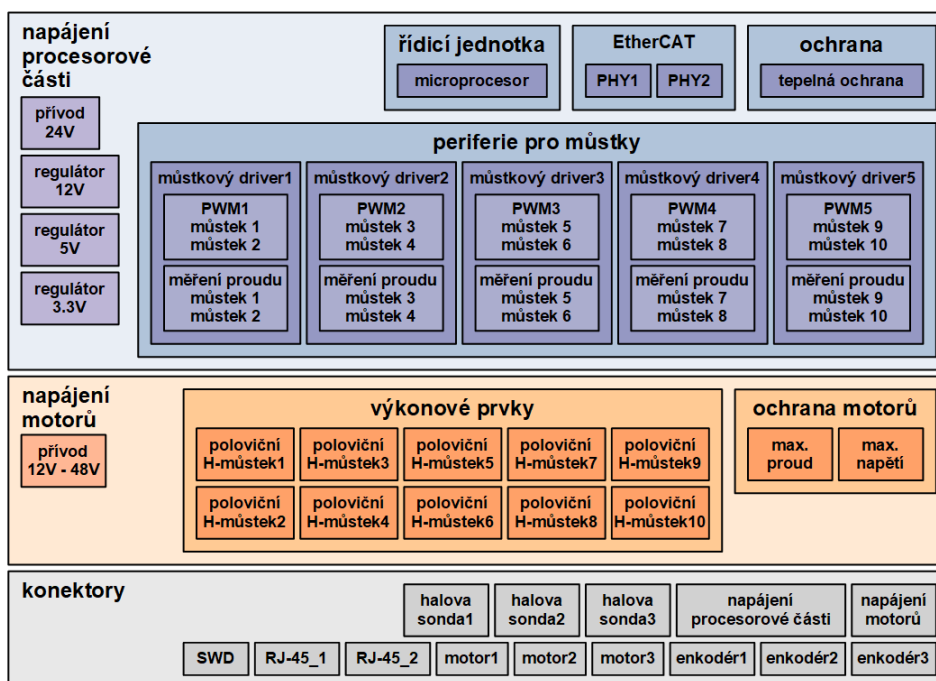
Obr. 2.7: Odezva systému - P-regulátor polohy

důvodu, že hlavním cílem řízených regulátorů bylo pomocí bloků saturace zohlednit napájení motoru v simulaci.

2.3 Návrh elektrického schématu

Návrh je zaměřen na vytvoření modulu s pulzním proudovým regulátorem, možností 4Q řízení motoru a komunikací EtherCAT. Aby se daly řešit polohovací úkoly, DPS musí obsahovat konektory pro připojení senzorů polohy. Jako senzory polohy se používají enkodéry, resolvery, jak je napsáno v kapitole 1.5, a Hallové sondy. Tedy DPS musí mít konektory pro připojení enkodéru každého motoru a obsahovat konektory pro připojení Hallovy sondy.

Na začátku se modul navrhoval pro řízení třech kartáčových motorů. Pak bylo rozhodnuto rozšířit jeho možnosti. Nová verze modulu by měla umožnit ovládání třech kartáčových nebo třech bezkartáčových motorů. Avšak primární funkcí modulu zůstává řízení kartáčových motorů. Na obrázku 2.8 je nakresleno blokové schéma modulu.



Obr. 2.8: Blokové schéma navržené DPS

2.3.1 Řídicí jednotka

Řízení takového modulu je velmi náročné. Proto se na řídicí jednotku kladou určité požadavky:

- možnost použití EtherCAT komunikace,
- dostatečný počet nezávislých PWM modulů,

- dostatečný počet nezávislých převodníků AD.

Výše uvedené požadavky splňuje MCU od firmy Infineon řady XMC4800 s označením XMC4800F144K2048AA. Mikrokontroler je založen na jádře ARM Cortex-M4 a je vyroben pro použití v průmyslových aplikacích.

Z důvodu toho, že piny mikroprocesoru jsou multifunkční, vznikl při tvorbě zapojení MCU seznam priorit:

- EtherCAT komunikace,
- SWD,
- PWM moduly,
- signály s informací o měření proudu,
- signály z Hallových sond,
- signály enkodérů.

Zmíněná řídicí jednotka pracuje na frekvenci 144 MHz. Aby se dosáhlo uvedené hodnoty hodinového signálu a signál byl kvalitní, k speciálnímu pinu se zapojí 12 MHz oscilátor. Signál s uvedenou frekvencí vstoupí do modulu PLL uvnitř součástky, kde se kmitočet vynásobí 12, a výstupní hodinový signál bude mít maximální frekvenci s dobrou kvalitou.

Mikroprocesor bude využívat převodníky AD. Aby převody byly přesné a kvalitní, měřený signál se musí porovnávat se stabilním a přesným referenčním napětím, které je možné vytvořit pomocí např. přesného nízko-šumového regulátoru napětí. Tento způsob získání referenčního bodu pro převodníky AD je zbytečně složitý a drahý. Referenční napětí se udělá z napájecího napětí a filtru kondenzátor-cívka-kondenzátor. To zajistí poměrně stabilní napěťovou referenci pro převodníky v MCU.

XMC4800 má v sobě dva moduly s názvy POSIF (position interface), které se používají pro připojení zpětné vazby (enkodér, Hallova sonda) a jsou propojené přímo s moduly PWM pro optimalizaci procesu ovládání motorů. Bohužel tyto moduly nejde použít z důvodu toho, že piny těchto modulů jsou sdílené s piny modulu EtherCAT, který má při zapojení větší prioritu. Dalším důvodem nepoužití je nedostatečný počet modulů POSIF.

2.3.2 Můstkový driver

Jak již bylo zmíněno na začátku této kapitoly, návrh řízení motoru je směřován k použití pulzního proudového regulátoru s můstkovým zapojením. Pro ovládání tří motorů DC je nutno použít 6 polovičních H-můstků. Pro ovládání tří motorů EC je nutno použít 9 polovičních H-můstků.

V zapojení se budou používat MOSFET tranzistory s kanálem typu N. Z výsledků porovnání zapojení a tranzistorů byly vybrány součástky od firmy Texas

Instruments CSD19534Q5A. Důvodem zvolení tohoto tranzistoru byly následující body:

- $V_{DS} = 100 \text{ V}$ - pokryje celý napěťový rozsah stejnosměrných motorů (kapitola 1.4),
- $I_D = 10 \text{ A}$ - stejnosměrný motor může mít proudový odběr cca až 8 A,
- $V_{GS} = 2.8 \text{ V}$,
- osobní zkušenosti s tímto tranzistorem.

I když je spínací napětí tohoto tranzistoru malé, není zaručeno, že tranzistor se bude otevírat a uzavírat spolehlivě při přímém spojení s MCU. Z důvodu toho, že se navrhuje pulzní regulátor, MCU bude posílat do Gate pinu tranzistoru PWM signál s hodnotou napětí 3,3 V. PWM signál by měl být převeden na vyšší hodnotu napětí, např. 12 V. Tuto úlohu řeší můstkový driver. Přivedené signály PWM z 3,3 V budou přesně okopírovány a vysílány s napětím 12 V. Tím pádem bude zajištěno spolehlivé spínání MOSFET tranzistorů v můstku.

Můstkové drivery mají spoustu různých variací a jsou rozšířené o další funkcionality (měření proudu, ochrany, nouzové vypínání atd.). Hlavními požadavky na driver pro navrhovanou DPS jsou:

- optimální množství ovládaných tranzistorů,
- ovládání externích tranzistorů,
- možnost měření proudu větví,
- možnost odpojení napájení motoru přívodem speciálního signálu.

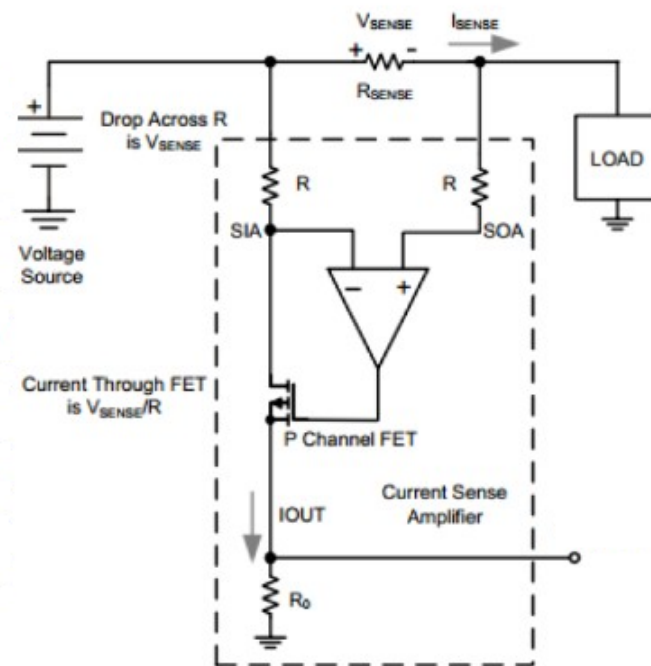
Pro ovládání H-můstků byl vybrán integrovaný obvod od firmy Texas Instruments SM72295MAX/NOPB. Tento můstkový driver splňuje výše uvedené požadavky, a používá se pro řízení plného nebo dvou polovičních H-můstků. Každý přivedený PWM signál řídí jenom jeden výkonový tranzistor. Použití externích tranzistorů umožňuje pouštět do motoru vyšší proudy. SM72295 má jako další funkci informační diody LED, integrované převodníky IV a nouzové vypínání.

Vybraný můstkový driver také obsahuje dva zesilovače pro měření proudu, pomocí kterých se provádí měření proudu větví a informace se odesílá do MCU ve tvaru analogových signálů. Tím se dá zlepšit řízení motoru nebo je použit pro ochranu. Na obrázku 2.9 je nakresleno zapojení zesilovače pro měření proudu.

Rovnice 2.16 ukazuje výpočet výstupního napětí z operačního zesilovače. Hodnoty pro výpočet byly použity ze schématu modulu.

$$V_{IPHASE} = \frac{V_{SENSE} * R}{R_O} = \frac{0.015 * 2200}{10} = 3,3[V], \quad (2.16)$$

kde V_{IPHASE} je výsledná analogová hodnota napětí odpovídající měřenému proudu, V_{SENSE} je napětí na měřicím odporu odpovídající maximálně předpokládanému proudu (15 A), R a R_O jsou rezistory z obrázku 2.9 s hodnotami ze schématu.



Obr. 2.9: Zapojení zesilovače pro měření proudu [15]

Část integrovaného obvodu s napájením zesilovačů pro měření proudu je napájena 3,3 V. Tedy maximální analogová hodnota, která bude odeslána do mikroprocesoru, bude 3,3 V.

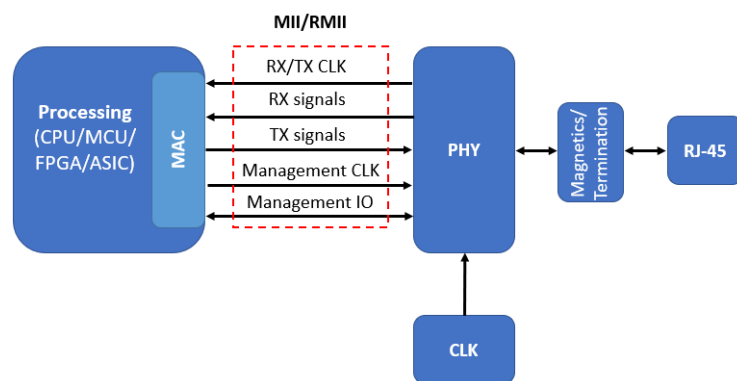
SM72295 umožňuje řídit jenom dva poloviční H-můstky, proto se musí použít 5 stejných čipů. Tím se zaručí řízení 10 polovičních můstek. DC motory potřebují 6 polovičních můstek a EC motory potřebují 9 polovičních můstek. Jeden poloviční můstek zůstává nevyužit. Proto zesilovač pro měření proudu, který by měřil proud touto větví, se používá pro měření proudu všemi motory. Výsledek tohoto měření se používá ochranným zapojením pro reakci na velký proud (nadproudová ochrana) a snímá se MCU. Ve chvíli, kdy dojde k sepnutí nadproudové ochrany nebo jiných ochran (napěťová ochrana a teplotní ochrana), OVS pin (pin nouzového vypnutí) změní stav na High a odpojí driver do té doby, dokud se tento signál nevrátí do neaktivního (Low) stavu.

Uvedený můstkový driver nemá ochranu proti současnému sepnutí všech tranzistorů, což je nežádoucím stavem, který může způsobit poškození integrovaného obvodu nebo můstku. Ochranu proti této situaci zajišťují PWM moduly MCU, které dovolí najednou zapínat pouze dva kanály ze čtyř.

2.3.3 EtherCAT

V kapitole 1.6 je základní popis průmyslových Ethernetů a několika druhů často používaných sběrnic. Aby byla komunikace zaručena, musí existovat fyzická vrstva, která bude řešit předávání bitových řetězců mezi jednotkami.

EtherCAT blok řídicí jednotky se zabývá tím, že přijímá, analyzuje a zpracovává procházející datový tok. Hlavním účelem bloku je povolení a koordinace přístupu k interním registrům a paměťovým prostorům. Uvedená funkcionality patří rozhraní MAC. To znamená, že kontrolér nemá zabudovanou fyzickou vrstvu uvnitř. Je nutno externě přidat PHY blok, jak je zobrazeno na obrázku 2.10, který bude převádět digitální datový tok z MAC na bitový řetězec a odesílat ho po fyzickém síťovém rozhraní, anebo bude provádět opačnou operaci.



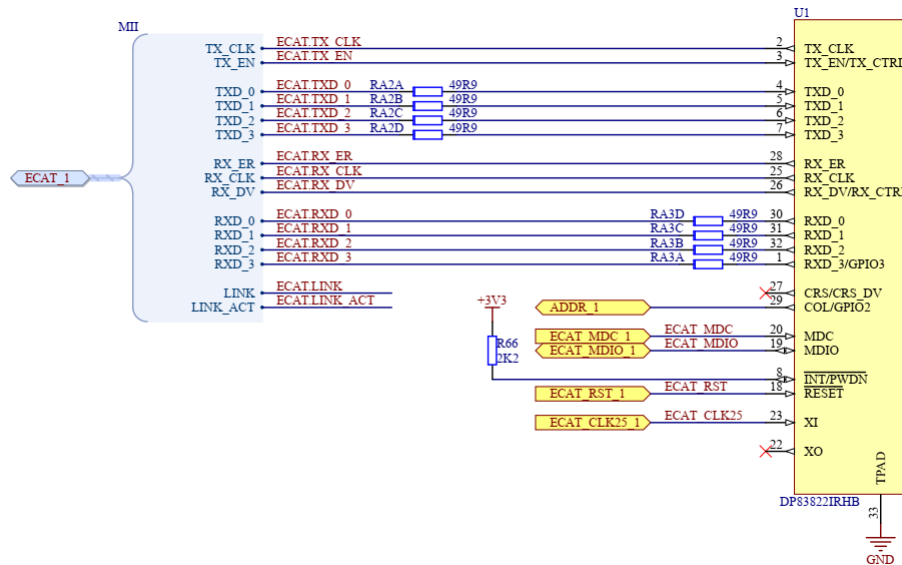
Obr. 2.10: Model zapojení PHY [17]

Čip PHY používá standardy MII a RMI. Pomocí těchto standardů vznikne spojení s řídicí jednotkou. Standard MII nemá obousměrnou komunikaci, přenáší data použitím čtyř spojů TX, čtyř spojů RX a vyžaduje hodiny s určitou frekvencí (25 MHz) pro RX a TX. Odesílání a přijímání se provádí na náběžnou hranu hodinového impulsu. Tím se zaručuje synchronní přenos po čtyřech vodičích a se získanou zprávou se dá správným způsobem pracovat. Standard RMI funguje stejným způsobem, rozdíl je jenom v hodinových signálech (2x rychlejší) a počtu vodičů (2x méně).

V návrhu byl použit PHY čip od firmy Texas Instruments DP83822IRHB. Daný integrovaný obvod pracuje s oběma standardy, které se volí pomocí speciálního pinu. Na této kartě je orientace na použití MII standardu z důvodu použití pomalejších hodin a možnosti inspirace Evaluation Modulem DP83822I EVM.

Datové vodiče mezi MCU a PHY jsou opatřeny odpory 50Ω , jak je nakresleno na obrázku 2.11. Důvod takového zapojení spočívá v tom, že vysílač a přijímač mají různé vnitřní impedance, díky čemuž může ve spoji vzniknout odraz signálu od přijímače, který následně bude zatěžovat vysílač. Aby se tomu zamezilo, na spoj

bude přidán 50 Ω odpor, který s odporem vysílače vytvoří dělič. Tím vznikne signál s poloviční amplitudou, který dorazí k přijímači a odrazí se. Díky tomu odrazu vznikne signál s amplitudou identickou přijatému signálu. Dvě poloviční amplitudy přijatého a odraženého signálu se sečtou a přijímač získá signál s plnou amplitudou.



Obr. 2.11: Elektrické zapojení PHY

2.3.4 Ochrana

Můstkový driver SM72295 nemá žádné ochranné zapojení a ponechání ochrany na MCU není dobrým řešením, protože se řídicí jednotka může zaseknout. V tomto stavu ochrana přestane fungovat nebo nezareaguje okamžitě. Je tedy lepší cestou navrhnout jednoduchou analogovou ochranu. Řešením se stalo použití čtyř operačních zesilovačů, jednoho klopného obvodu typu D, teplotního senzoru a třech potenciometrů.

Pomocí potenciometrů se nastavují referenční napětí pro operační zesilovače.

Operační zesilovače jsou zapojené jako komparátory napětí, a používají se následujícími způsoby:

- porovnání napětí odpovídajícího hodnotě proudu s napětovou referencí (proudová ochrana),
- porovnání napětí odpovídajícího teplotě s napětovou referencí (teplotní ochrana),
- porovnání napájecího napětí motoru s napětovou referencí (napětová ochrana),
- kontrola výstupů výše uvedených operačních zesilovačů a sdílení informace o sepnutí jedné z ochran.

Výstup posledního zesilovače je připojen na hodinový signál D klopného obvodu. Tento obvod má na datovém vstupu připojené napětí 3,3 V (log. 1), které se okopíruje s příchodem náběžné hrany hodinového signálu na výstup. Výstup z počátku má 0 V (připojeno u můstkových driverů na piny OVS) a při resetování obvodu se bude vracet do původního stavu (log. 0). Uvedený D klopný obvod byl použit kvůli tomu, aby můstkové drivery čekaly na pokyny mikrokontroleru při sepnutí ochran. Jinak řečeno MCU může při sepnutí ochran vykonat určitou rutinu, pak následně uvolnit můstkové drivery a pokračovat v činnosti.

2.3.5 Konektory

Na celé desce se používá jenom jeden typ konektorů (MRT9 P3.5/x) s různým počtem pinů z důvodu zachování designu desky. Hlavními požadavky na tento typ součástky byly:

- dostatečná proudová propustnost,
- rozpojitelnost,
- montáž THT - pevné uchycení na DPS.

Z hlediska návrhu větší pozornost vyžadují následující konektory a jejich zapojení:

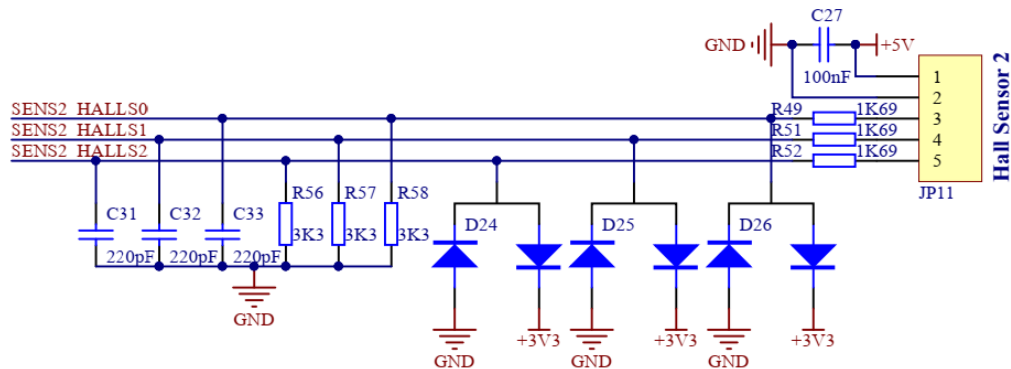
- napájecí konektor pro motory,
- konektory pro připojení motorů,
- konektory pro enkodéry a Hallovy sondy.

Pro napájení motorů se použil čtyřpinový konektor, kde dva piny jsou zapojeny na "+" a dva další jsou na "-". Takové zapojení bylo zvoleno z důvodu proudové propustnosti. Během návrhu se počítalo s tím, že každým motorem poteče maximálně proud 5 A, tedy maximální proud deskou může být 15 A. Jeden pin vybraného konektoru zvládne maximálně 9 A, při zapojení dvou pinů paralelně se hodnota maximálního proudu zvedne do 18 A. Zapojení dvou pinů znamená, že piny konektoru nebudou proudově přetíženy při maximálně předpokládaném proudovém odběru.

Pro připojení motoru by se dalo vystačit s dvojpínovým konektorem, třípínový konektor se použil kvůli rozšíření možností karty. Díky tomuto zapojení je možné řídit motory zapojené na jednom konektoru. Řešení zapojit motory takovým způsobem povede k tomu, že z 4Q řízení stane 1Q. Uvedené zapojení se týká jenom kartáčových motorů, směr otáčení motorů bude záviset na zapojení přívodů motorů.

Co se týče konektorů pro enkodéry a Hallovy sondy, ty jsou spojeny přímo s mikroprocesorem. Snímače pro svou činnost potřebují napájení, hodnota napětí se ve většině případů rovná 5 V. Z tohoto plyne, že signály ze snímačů mohou mít hodnoty napětí 5 V. Přivedení takového napětí na MCU způsobí jeho zničení. Snížení napětí

by se dalo dosáhnout pomocí tzv. level shifter součástky, která převede signál z 5 V logiky na 3,3 V logiku, nebo se může použít obyčejný dělič napětí s ochrannou diodou a filtračním kondenzátorem, jak je znázorněno na obrázku 2.12.



Obr. 2.12: Zapojení konektoru s Hallovou sondou nebo enkodérem

$$V_{\text{HALLS0}} = \frac{V_{\text{ENC}} * R_{49}}{R_{58} + R_{49}} = \frac{5 * 3300}{4990} = 3,3[V], \quad (2.17)$$

kde V_{HALLS0} je hodnota napětí přivedená na pin MCU, V_{ENC} je napětí signálu z enkodéru a R_{49} a R_{58} jsou rezistory zapojené ve schématu.

Při vysokých kmitočtech signálu a velké hodnotě kapacity kondenzátoru může docházet ke ztrátě signálu. Tedy je nutno provést výpočet hodnoty kondenzátoru. Při výpočtu hodnoty se vzaly následující technické charakteristiky:

- motor - 5000RPM,
- enkodér - rozlišení 8192 (4096 pulzů na kanál).

Maximální rychlost motoru je nutno přepočítat z RPM (otáček za minutu) na otáčky za sekundu:

$$v_{\text{SEC}} = \frac{v_{\text{RPM}}}{60} = \frac{5000}{60} = 83,333[\text{ot./sec.}], \quad (2.18)$$

kde v_{SEC} je rychlost otáček za sekundu a v_{RPM} je rychlost otáček za minutu.

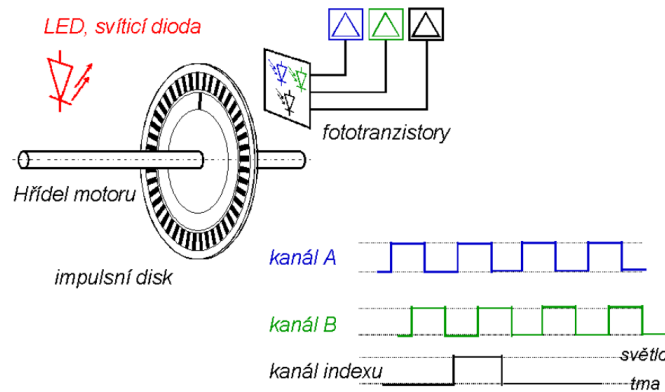
Rozlišení enkodéru je bezrozměrná veličina, ale může se jí přiřadit fiktivní jednotka pulzy za otáčku. Díky této úvaze je možné vypočítat maximální frekvenci signálu, který bude vyslán enkodérem do MCU.

$$f_{\text{ENC}} = \text{rozlišení} * v_{\text{SEC}} = 4096 * 83,333 = 341000[\text{Hz}], \quad (2.19)$$

kde f_{ENC} je maximální frekvence signálu enkodéru.

V danou chvíli je známá teoretická frekvence signálu z enkodéru při maximálních otáčkách. Z tohoto parametru se může získat perioda signálu, která se bude rovnat

2,933 μs . Teoreticky bude signál z enkodéru vypadat jako PWM signál se střídou 50 %, protože kotouče uvnitř enkodéru mají rovnoměrnou strukturu. Příklad takového enkodéru je na obrázku 2.13.



Obr. 2.13: Příklad optického enkodéru [2]

Aby signál neztrácel informaci pod vlivem filtračního kondenzátoru, maximální doba nabíjení a vybíjení musí být nižší než čtvrtina periody signálu. Z toho plyne, že doba nabíjení/vybíjení se rovna 0,733 μs .

Poslední parametr, který je nutno vědět pro výpočet hodnoty kapacity kondenzátoru, je proud tekoucí do této součástky. Proud je dán děličem napětí, a počítá se dle Ohmová zákona:

$$I = \frac{U}{R_{58} + R_{49}} = \frac{5}{4990} = 1,002[mA]. \quad (2.20)$$

Hodnota kapacity se vypočítá dle vzorce 2.21, který se musí upravit:

$$U * C = T * I, \quad (2.21)$$

$$C_{33} = \frac{T * I}{U} = \frac{0,733 * 10^{-6} * 1,002 * 10^{-3}}{3,3} = 2,225 * 10^{-10} = 222,5[pF]. \quad (2.22)$$

Díky výpočtu 2.22 je jasně, jak maximálně velký filtrační kondenzátor se má připojit na signálové cesty enkodérů a Hallových sond. Z obrázku 2.12 je vidět, že se přidali kondenzátory o hodnotě 220 pF. V reálu kapacity mohou být větší, když se přičtou parazitní kapacity diod, padů součástek a spojů. Použití enkodérů s rozlišením 8192 a více není běžná situace. Při použití přesnějších enkodérů se musí hodnoty filtračních kondenzátorů přepočítat.

Vedení enkodérů a Hallových sond se vystavuje různým zdrojům rušení, kde se vlivem elektromagnetických polí může dojít k generování rušivého signálu. Filtrační kondenzátory se používají k omezování elektromagnetického rušení a zvýšení odolnosti "přijímacího" zařízení.

2.3.6 Napájení modulu

Návrh napájení DPS je jedním z klíčových bodů při tvorbě schématu zapojení. Z důvodu toho, že karta má logickou (MCU, komunikace, senzory, ochrany) a výkonovou (napájení motorů) část, je napájení karty rozděleno.

Napájení výkonové části je přivedeno na každý z deseti polovičních H-můstků a na napěťovou ochranu.

Logická část se napájí ze zdroje s výstupním napětím 24 V. Tato hodnota napájení byla zvolena z důvodu, že na strojích je to běžná hodnota napájení. Dále je napájení logiky rozděleno na tři různá napětí:

- 12 V - napájení můstkových driverů,
- 5 V - napájení Hallových sond a enkodérů,
- 3,3 V - napájení zbývající logiky (MCY, PHY atd.).

Před hledáním regulátorů napětí je nutno spočítat proudový odběr pro každou hodnotu napájecího napětí a započítat k tomu rezervy.

Tab. 2.2: 3,3 V - proudový odběr

Součástka	Počet [ks]	Max. proudový odběr [mA]
MCU	1	135
PHY	2	82
D-klop. obvod	1	100
můstkový driver	5	3
komparátor	1	2
celková spotřeba		416

Z tabulky 2.2 je vidět, že odběr proudu pro napětí 3,3 V se bude teoreticky rovnat 416 mA. Aby bylo zaručeno, že při maximálním proudovém zatížení regulátoru nedojde k vypnutí logiky z důvodu nedostatku proudu, musí se k získané hodnotě proudového odběru připočítat rezerva (hodnota rezervy se volí návrhářem). Regulátor napětí s výstupem 3,3 V musí mít výstupní proud minimum 1 A. Velká hodnota rezervy zaručí eliminaci možných chyb v návrhu. Např. se nezapočítal odběr nějaké součástky nebo odběr součástky byl větší, než se očekávalo.

Tabulka 2.3 říká, že maximální proudový odběr senzorů se teoreticky pohybuje kolem 690 mA. Z toho plyne, že 5 V regulátor napětí musí mít výstupní proud minimum 1A.

Tabulka 2.4 prezentuje proudovou spotřebu součástek používajících napětí 12 V. Dané napětí je přivedeno jenom na můstkové drivery s proudovou spotřebou 15 mA. Z tohoto plyne, že regulátor 12 V může mít výstupního proudu minimum 100 mA.

Tab. 2.3: 5 V - proudový odběr

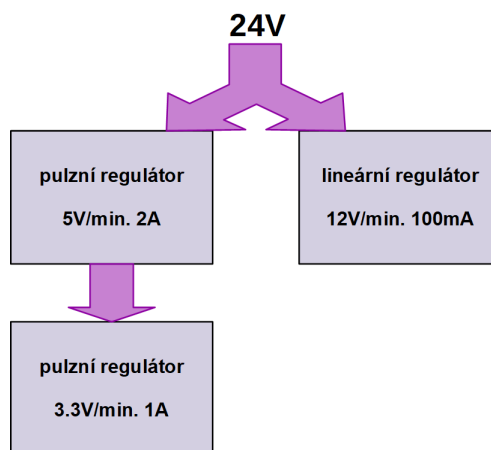
Součástka	Počet [ks]	Max. proudový odběr [mA]
enkodér	3	220
hallová sonda	3	10
celková spotřeba		690

Tab. 2.4: 12 V - proudový odběr

Součástka	Počet [ks]	Max. proudový odběr [mA]
můstkový driver	5	3
celková spotřeba		15

Teď je nutno rozhodnout o tom, jaké typy regulátorů použít, a jak mají být poskládané. Existují dva typy regulátorů:

- pulzní napěťový regulátor (switching) - poměrně složité zapojení, minimální tepelné ztráty,
- lineární napěťový regulátor (LDO) - jednoduché zapojení, tepelné ztráty závisí na maximálním výstupním proudu a rozdílu vstupního a výstupního napětí.



Obr. 2.14: Model napájení procesorové části

Na obrázku 2.14 je nakreslen navržený model napájení procesorové části modulu. Je potřeba věnovat zvýšenou pozornost návrhu lineárního regulátoru, protože na něm může dojít k velkému vyzařování tepla.

$$P = (U_{\text{in}} - U_{\text{out}}) * I = (24 - 12) * 0,015 = 0,18[W], \quad (2.23)$$

kde U_{in} je napětí na vstupu regulátoru, U_{out} je výstupní napětí, I je proud odebíraný z regulátoru a P je ztrátový výkon (vyzařované teplo).

$$\theta_j = \theta_a + R_{\theta JA} * P = 25 + 55 * 0,18 = 34,9[^\circ C], \quad (2.24)$$

kde θ_j je teplota přechodu, θ_a je teplota okolí a $R_{\theta JA}$ je tepelný odpor přechodu bez chladiče.

Z rovnice 2.24 je vidět, že teplota LDO lineárního regulátoru teoreticky se bude pohybovat kolem 34,9 °C. Regulátor nepotřebuje přídavné chlazení. V situaci maximálního proudového zatížení regulátoru bude teplota 91 °C. Teplota LDO regulátoru je v tomto případě značná, bude potřeba přidat chladič. Situace s maximálním proudovým zatížením LDO regulátoru by neměla nastat.

Takže je potřeba věnovat pozornost 5 V switching regulátoru, který musí mít výstupní proud minimálně 2 A. Dříve se uvažovalo, že výstupní proud tohoto regulátoru má být 1 A, ale z toho důvodu, že je k němu připojen regulátor s výstupním napětím 3,3 V, se sečtou výstupní proudy uvedených regulátorů.

2.4 Návrh DPS

Návrh desky plošného spoje modulu je stejně důležitý jako tvorba schématu. Od tohoto kroku závisí, jestli zapojení bude fungovat a bude-li fungovat optimálně. Během návrhu DPS se stanovily následující důležité body:

- rozmístění součástek,
- vedení výkonové části,
- propojení mezi vrstvami (prokovy),
- vedení nejrychlejších signálů,
- skladba DPS,
- fixace modulu.

2.4.1 Rozmístění součástek

Od rozmístění součástek závisí složitost designu DPS a počet vrstev. Jestli se podaří minimalizovat křížení cest, dá se dosáhnout minimálního počtu vrstev.

Po rozmístění součástek vznikl obrys DPS a zjistilo se, že některé křížení se dá odstranit prohozením pinů ve schématu (PWM bloky, kanály enkodérů a Hallových sond). Během provádění změn zapojení jednotlivých pinů řídicí jednotky se dbalo na to, aby těmito zásahy do schématu nedošlo ke vzniku chyb. Některé křížení nešlo odstranit z důvodu zapojení speciálního bloku MCU (EtherCAT blok).

Po rozvodu modulu se dospělo k závěru, že budou stačit jenom 4 vrstvy s rozložením:

- Top vrstva - signály;
- Mid1 vrstva - napájení;
- Mid2 vrstva - GND;
- Bottom vrstva - signály.

2.4.2 Skladba DPS

Po inspiraci různými moduly od firmy Texas Instruments s osazeným integrovaným obvodem SM72295 bylo rozhodnuto použít skladbu uvedenou v tabulce 2.5. Z důvodu toho, že výrobce DPS neměl skladem takové jádro, nabídl skladbu uvedenou v tabulce 2.6.

Tab. 2.5: Původní skladba DPS

Vrstva	Typ	Tloušťka [mm]
top	měď	0,07
diel1	izolace	0,3048
mid1	měď	0,07
diel2	jádro	0,8636
mid2	měď	0,07
diel3	izolace	0,3048
bottom	měď	0,07

Tab. 2.6: Upravená skladba DPS

Vrstva	Typ	Tloušťka [mm]
top	měď	0,05
diel1	izolace	0,296
mid1	měď	0,07
diel2	jádro	0,713
mid2	měď	0,07
diel3	izolace	0,296
bottom	měď	0,05

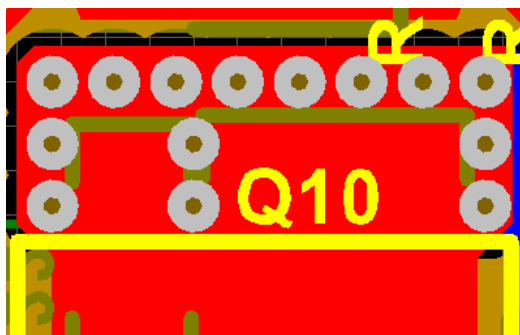
Nežádoucí změnou v nové skladbě je zmenšení tloušťky vnějších vrstev, ale toto ztenčení při procesu pokovení všech otvorů se napraví, a výsledná tloušťka vnějších mědí bude na požadované hodnotě 70 μm (narůst galvanické mědi). Materiálem pro prepregy a jádro je IS 400 ML.

2.4.3 Vedení výkonové části

Pod výkonovou částí DPS se rozumí zapojení polovičních H-můstků a jejich napájení. Napájení motorů má hodně odvětvení do jednotlivých můstků a tečou tam velké proudy. Nejlepším způsobem, jak to rozvést, je použít rozlitou měď (polygony). Když budou polygony velké a široké, bude zaručeno, že DPS se nezničí a proudové ztráty budou minimální. Kvůli zvětšení proudové propustnosti a snížení rizik se použije posílená měď.

2.4.4 Propojení vrstev

Prokovy se používají pro přechod mezi vrstvami. Na celé desce jsou použity prokovy s průměrem díry 0.2 mm a průměrem mědi 0.6 mm. Teoreticky je jeden takový prokov schopen zvládnout proud 1 A (dle elektrotechnické kalkulačky Saturn PCB Toolkit). Pro větší proudy se může udělat síť prokovů, jak je znázorněno na obrázku 2.15. V uvedené síti je 14 prokovů. Toto provedení by teoreticky mělo zvládnout proud 14 A.



Obr. 2.15: Příklad sítě prokovů

2.4.5 Vedení rychlých signálů

U rychlých signálů se ve výsledném návrhu musí kontrolovat délka spojů a rozhodovat o tom, jestli je nutno provádět délkové přizpůsobení. Na DPS jsou nejrychlejšími signály hodiny a datové spoje z MCU do PHY (25 MHz). Nejvíce je nutné dbát na datové vodiče, protože ty paralelně přenášejí informaci.

Tabulka 2.7 udává informace o délce spojů obou PHY. Největší délkový rozdíl spojů je roven 13,073 mm. Aby se dalo rozhodnout o nutnosti délkového přizpůsobení, musí se udělat jednoduchý výpočet. Výsledek výpočtu se pak porovná s

Tab. 2.7: Délky datových spojů MCU - PHY

Spoj	Délka [mm]
ECAT0.TX0	53,166
ECAT0.TX1	55,145
ECAT0.TX2	56,456
ECAT0.TX3	58,059
ECAT0.RX0	54,144
ECAT0.RX1	54,322
ECAT0.RX2	55,117
ECAT0.RX3	59,583
ECAT1.TX0	37,078
ECAT1.TX1	37,531
ECAT1.TX2	50,151
ECAT1.TX3	48,678
ECAT1.RX0	45,552
ECAT1.RX1	44,016
ECAT1.RX2	42,504
ECAT1.RX3	42,510

periodou signálu, na základě kterého se dělal výpočet a rozhodne se o přizpůsobení.

$$T_{\text{signal}} = \frac{1}{f_{\text{signal}}} = \frac{1}{25 * 10^6} = 4 * 10^{-8} = 40[ns], \quad (2.25)$$

kde T_{signal} je perioda signálu a f_{signal} je frekvence signálu.

$$T_{\text{zpozdzeni}} = \frac{\Delta l_{\text{signal}}}{v_{\text{signal}}} = \frac{13,073 * 10^{-3}}{1,5 * 10^8} = 8,715 * 10^{-11} = 87,15[ps], \quad (2.26)$$

kde $T_{\text{zpozdzeni}}$ je doba rozdílu příchodů signálů s nejkratší a nejdelší cestou, Δl_{signal} je rozdíl délky nejkratší a nejdelší cesty a v_{signal} je rychlost šíření signálů.

Signály ve vodičích na DPS se můžou šířit s rychlosti od $1,5 * 10^8$ m/s do $3 * 10^8$ m/s. Ve vzorci 2.26 se použila hodnota rychlosti šíření signálu $1,5 * 10^8$ m/s, aby se počítalo s nejhorší variantou.

$$T_{\text{zpozdzeni-\%}} = \frac{T_{\text{zpozdzeni}} * 100\%}{T_{\text{signal}}} = \frac{87,15 * 10^{-12} * 100}{40 * 10^{-9}} = 0,218[\%], \quad (2.27)$$

kde $T_{\text{zpozdeni-\%}}$ je vyjádření zpoždění v procentech vůči periodě signálu. Díky výsledku rovnice 2.27 může být rozhodnuto o nutnosti provádění délkového přizpůsobení. Délkové přizpůsobení není třeba provádět, protože hodnota zpoždění oproti periodě signálu je velice malá, a dá se považovat za zanedbatelnou. Z toho se může

udělat další závěr, signály, které jsou pomalejší než 25 MHz, nemusí být přizpůsobeny. Délkové přizpůsobení se musí udělat u diferenciálních párů, ty jsou poznamenány ve schématu speciálním způsobem.

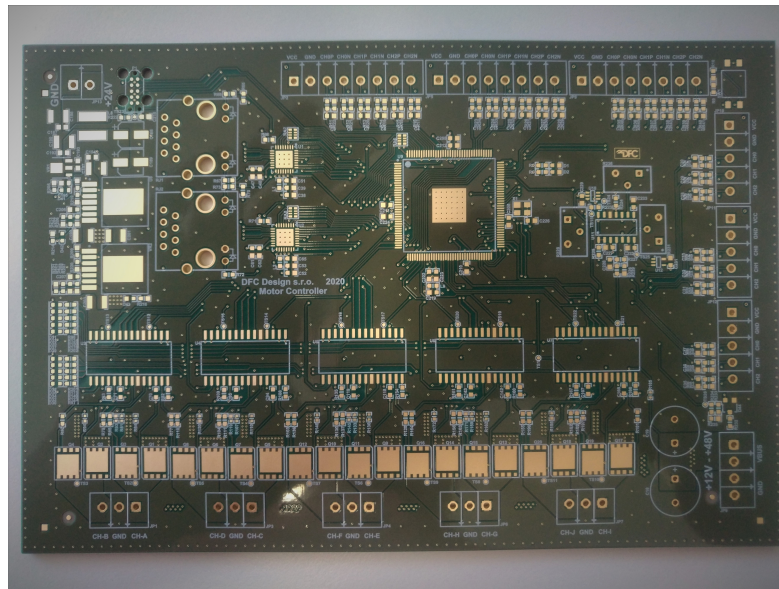
2.4.6 Fixace modulu

Po dokončení elektrického designu modulu jsou jeho rozměry 100 mm x 150 mm x 1,6 mm, a před odesláním DPS na výrobu je nutno řešit problém fixace modulu ve strojích. Fixace pomocí šroubu není praktické řešení z důvodu komplikované instalace a odinstalace. Po prohlédnutí boxů a krabiček pro elektroniku se řešením stalo použití modulárních boxů. Aby tam modul zapadl, musí se zvětšit obrys DPS do rozměrů 107 mm x 160 mm x 1,6 mm.

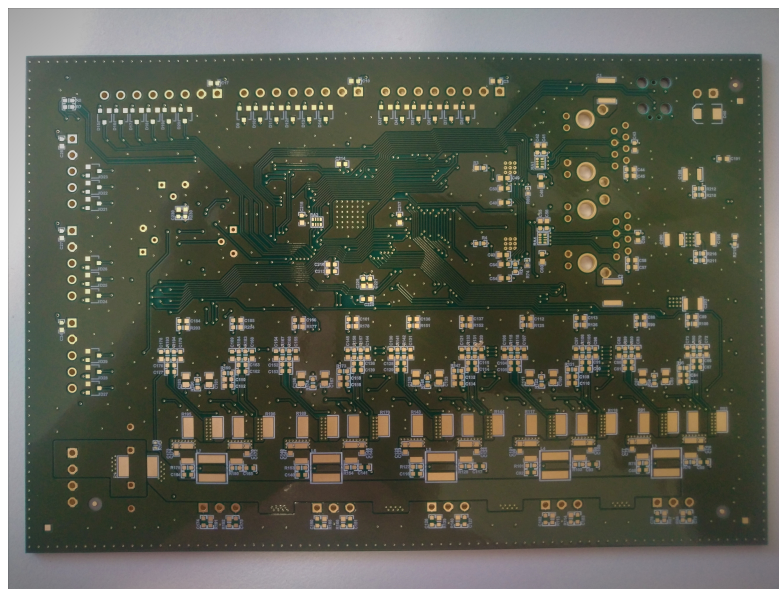
2.4.7 Výroba desky plošných spojů a příprava k osazení

Výroba desky se připravovala v panelu, aby bylo možné provést strojní osazení a pájení. Díky zvětšení obrysu se podařilo panel odstranit a značky pro osazovací stroj zaintegrovat do samotné desky. Osazovací značky se rozmístily takovým způsobem, že nemají ani jednu společnou souřadnici. To se dělá z důvodu eliminování chyby špatného umístění desky ve stroji, a kvůli lepšímu nastavení stroje.

Po úpravě DPS se vygenerovaly "Gerber Data" a objednala se výroba dvou kusů navrženého modulu se skladbou uvedenou v tabulce 2.6. Následovně se udělala a objednala se šablona pro tento modul. Vygenerovaly se soubory "BOM", aby se objednaly součástky, "Pick Place" a "Assembly Drawings", které se budou používat při osazování a pájení modulu.



Obr. 2.16: Hotová DPS - Top strana



Obr. 2.17: Hotová DPS - Bottom strana

3 Programové vybavení modulu

Pro návrh programového vybavení modulu se použilo vývojové prostředí IDE DAVE v4.4.2 od firmy Infineon. Daná platforma umožňuje nejen psát kód, ale i pomocí speciálních nástrojů se dá provést inicializaci potřebných bloků s přiřazením pinů a vygenerovat základní kód bez mazání už napsaného. Následující kapitoly popíší podrobněji EtherCAT komunikaci, způsob jejího programování, napsané funkce pro ovládání motorů a chod programu.

3.1 EtherCAT

EtherCAT byl vyvinut firmou Beckhoff Automation v roce 2003, a v roce 2007 vznikl mezinárodní standard. Funkcionalita, adresování a synchronizace jsou popsány v podkapitole 1.6.3, kde bylo uvedeno, že komunikace je založena na principu master-slave.

Master je jediným modulem v síti, kterému je povoleno vysílat nové ethercatové rámce. Ethercatový rámec je založen na ethernetovém rámci používající standard IEEE 802.3. Jakékoliv zařízení s ethernetovým portem může být mastrem, funkce spočívá jenom v softwaru. Z toho plyne, že není potřeba mít speciální hardware pro jeho vytvoření, je dostatečné použít obyčejnou síťovou kartu nebo ethernetový kontrolér.

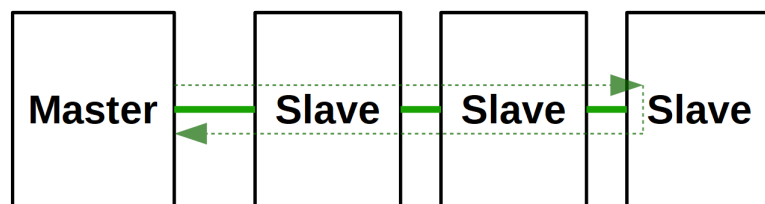
EtherCAT Slave už oproti masteru potřebuje speciální hardware, který musí zvládnout určité požadavky na rychlost průchodu signálů.

3.1.1 Topologie

Z podkapitoly 1.6.3 je známo, že EtherCAT je schopen rozpoznat topologii sítí a funguje na principu zavřeného kruhu s počátkem a koncem v masteru, jak je znázorněno na obrázku 3.1. Maximální počet modulů připojených ke sběrnici může být až 65535. Provedení fyzické vrstvy je závislé na délce kruhu:

- do 100 metrů - fast Ethernet (100BaseTX),
- nad 100 metrů - optické vlákno (100BaseFX).

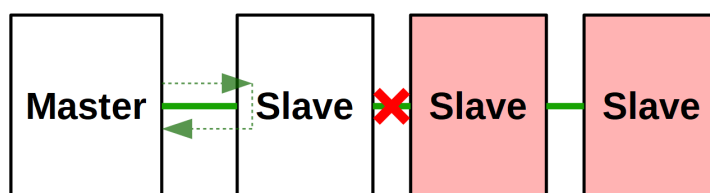
Jak již bylo řečeno, EtherCAT je založen na principu master-slave, tedy by měla být jen komunikace mezi masterem a slavem. To však není úplně pravda, protože se ethercatová sběrnice může používat pro centralizované i decentralizované systémy. Z čehož plyne, že existují komunikace master-master a slave-slave.



Obr. 3.1: Průchod EtherCAT rámce

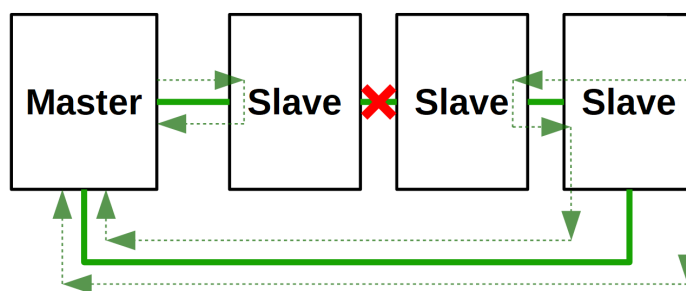
3.1.2 Odolnost

Ve chvíli poruchy vedení sběrnice nebo zařízení další zařízení budou odpojené od sítě, jak je znázorněno na obrázku 3.2.



Obr. 3.2: Průchod EtherCAT rámce při poruše

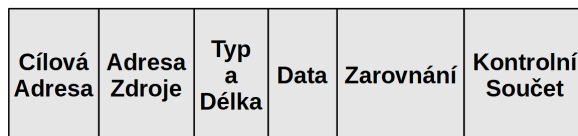
Aby se odstranil tento nedostatek, poslední slave se musí výstupem připojit k masteru, což znamená, že pro použití odolného zapojení master musí mít dva porty (mastery určené pro EtherCAT komunikaci). Jakmile master zjistí, že vznikla porucha, během jednoho komunikačního cyklu uzavře logický kruh jiným způsobem a bude pokračovat ve své činnosti. Odolné zapojení je prezentováno na obrázku 3.3.



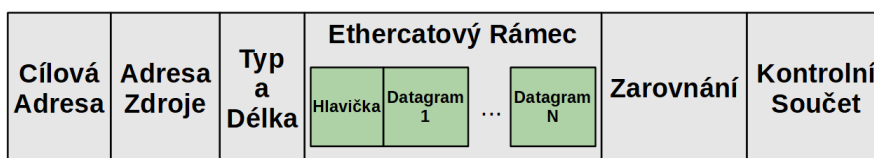
Obr. 3.3: Průchod EtherCAT rámce v odolném zapojení při poruše

3.1.3 Struktura rámce

Na začátku sekce o EtherCATu bylo řečeno, že Ethernet a EtherCAT používají stejný standard a rámec EtherCATu představuje sebou upravený rámec Ethernetu. Tyto rámce jsou znázorněny na obrázcích 3.4 a 3.5 .

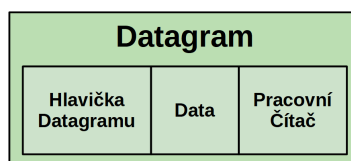


Obr. 3.4: Ethernet rámec



Obr. 3.5: EtherCAT rámec

Rámec EtherCATu nahrazuje datový rámec Ethernetu. Skládá se z hlavičky a řady datagramů. Na obrázku 3.6 je představeno složení datagramu, kde pracovní čítač je kontrola použití datagramu. Každé zařízení, které pracuje s určitým datagramem, inkrementuje hodnotu čítače. Po vrácení rámce do masteru se vyhodnotí, jestli hodnota čítače datagramu je rovna vypočítané hodnotě. Ve chvíli, kdy se hodnota čítače nerovná vypočítané hodnotě, bude vyhodnoceno, že přenos proběhl s chybou.



Obr. 3.6: Datagram ethercatového rámce

3.1.4 Adresování

V podkapitole 1.6.3 je řečeno, že adresování se provádí automaticky, ale může se zvolit způsob, jakým bude síť adresována:

- inkrementální - délka 2 byty,

- pevné - délka 2 byty,
- logické - délka 32 bytů.

Inkrementální adresování funguje způsobem, že slave zapojení po masteru získá adresu 0, uloží ji do paměti, provede inkrementaci a pošle dál.

Při pevném adresování slavu se přidělí adresa neodpovídající jeho pořadí, pak se adresování bude provádět pomocí inkrementace přidělené adresy. Při výpadku napájení se adresa ztratí.

Logické adresování se provádí čtením a zápisem dat z nebo do logického adresového prostoru masteru.

3.1.5 Synchronizace

Podkapitola 1.6.3 uvádí informaci o synchronizaci pomocí distribuovaných hodin, které se berou z prvního slavu schopného k distribuci času. Master pomocí synchronizačního datagramu, do kterého každé slave zařízení uloží čas průchodu, vyrovná zpoždění na sběrnice tak, že vypočte zpoždění každým slavem, a počítá s ním v dalších operacích.

3.2 Master

EtherCAT master nepotřebuje žádný speciální hardware, bude stačit obyčejná ethernetová karta s portem. V této práci je implementace EtherCAT masteru provedena na notebooku s 100 Mbit síťovou kartou.

Pro vytvoření mastera existují různé placené a open source softwary. V tomto případě se použije TwinCAT3 od firmy Beckhoff. TwinCAT3 je nadstavbou pro program Visual Studio od firmy Microsoft.

V této práci je TwinCAT použit kvůli ovládání hotového modulu pro řízení stejnosměrných motorů. Pomocí tohoto nástroje se do karty odesílají hodnoty, o kolik mají motory pohnout, a na kterou stranu mají vykonat pohyb. Zpátky se získávají aktuální hodnoty změny pozic každým motorem.

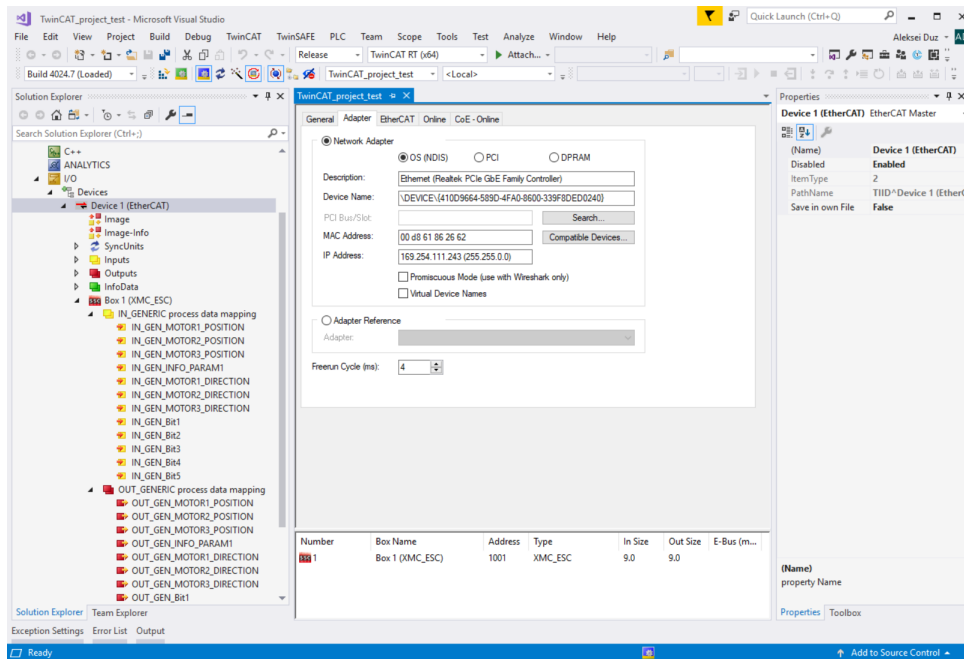
Použití softwaru je velice jednoduché, a vytvoření masteru se provádí následujícím způsobem:

- nainstalovat Visual Studio, TwinCAT3 a drivery pro síťovou kartu,
- po spouštění Visual Studio založit nový projekt typu TwinCAT XAE Project,
- v levém panelu rozkliknout záložku *I/O*, pravým tlačítkem vybrat *Device* a zvolit *Add New Item*,
- vybrat záložku *EtherCAT* a zvolit *EtherCAT Master*,
- v okně *Device Found At* vybrat síťovou kartu notebooku.

Master je vytvořen. Je nutno provést inicializace připojených slave zařízení:

- slave zařízení připojit k notebooku Ethernet kabelem a zapnout napájení,
- pravým tlačítkem kliknout na vytvořené zařízení (*Device1 (EtherCAT)*) a zvolit *Scan*.

Připojené slave zařízení jsou inicializovány. Jestli skenování EtherCAT slave zařízení proběhne bez chyb, objeví se záložka *Box1* popisující ethercatový rámec. Správně inicializovaný master a slave jsou znázorněny na obrázku 3.7.



Obr. 3.7: Plná inicializace mastera se slavem

Záložka *IN_GENERIC* prezentuje data, které byly vyčtena ze slave zařízení. Záložka *OUT_GENERIC* obsahuje data, které se posílají do slavu. Nastavování pohybu motorů se dělá následujícím způsobem:

- rozkliknout potřebnou záložku *OUT_GEN_MOTOR x _DIRECTION* a nastavit směr otáčení - zápis 0 nebo 1;
- rozkliknout příslušnou záložku *OUT_GEN_MOTOR x _POSITION* a nastavit vzdálenost, o kterou se má motor posunout.

3.3 Slave

Navržený modul pro řízení stejnosměrných motorů je slave zařízením, které ke své komunikační činnosti potřebuje speciální hardware. Aby MCU mohlo komunikovat po sběrnici, firmware musí mít zdrojové soubory Slave Stack Code, které budou realizovat EtherCAT slave. Uvedený kód se generuje nástrojem Slave Stack Code Tool

od firmy Beckhoff. Pro získání tohoto nástroje je nutno být právnickou osobou a být registrovaným u skupiny EtherCAT Technology Group. SSC Tool se používá k vygenerování SSC na základě speciálního Excel souboru a EtherCAT Slave Information souboru (XML soubor).

3.3.1 Implementace komunikace

Základem programového vybavení je jeden z příkladů vytvořených výrobcem čipu. Po zprovoznění komunikace se vybraný příklad upraví dle potřeby.

SSC Tool

Firmware je založen na příkladu ETHCAT_SSC_XMC48, který je stažen z oficiálních stránek výrobce (https://www.infineon.com/cms/en/product/promopages/aim-mc/dave_downloads.html). Projekt ve svém adresáři obsahuje složku SSC, ve které jsou následující soubory:

- Infineon_XMC_ECACAT_SSC_Config.xml - konfigurační soubor MCU pro SSC Tool,
- XMC_ESC.xlsx - excel soubor pro nastavení tvaru ethercatového řetězce,
- Src - složka, kam se generují zdrojové kódy (pokud složka chybí, SSC Tool si ji vytvoří).

Při zakládání projektu v softwaru SSC Tool se nejdříve musí naimportovat konfigurační soubor Infineon_XMC_ECACAT_SSC_Config.xml, po importování se vytvoří projekt s nastavením pro mikrokontroler XMC4800/4300.

V levé části nástroje SSC Tool jsou sekce, ve kterých se dá nastavovat informace různého typu, soubory pro vygenerování, distribuované hodiny atd. Uvedené nastavování se může přeskočit. Dalším krokem je práce s procesními daty. Před importováním (*Tool* → *Application* → *Import*) excel souboru s definovaným tvarem EtherCAT řetězce je nutno si promyslet, jak by měl vypadat řetězec a provést změny v XMC_ESC.xlsx. Generování SSC zdrojových kódů se provádí až po nahraní do SSC Tool upraveného excel souboru (*Project* → *Create new Slave Files*).

Zkoušely se různé varianty řetězce, a dospělo se k následujícím závěrům:

- upravovat jenom vstupní a výstupní data (názvy a default hodnoty),
- zachovat datovou strukturu příkladu (jinak vygenerování proběhne úspěšně, ale při komunikaci se bude hlásit chyba).

Výsledný tvar ethercatového řetězce zapsaného do souboru XMC_ESC.xlsx je znázorněn na obrázku 3.8.

Se zdrojovými kódy se generuje i ESI soubor, který obsahuje popis vytvořeného ethercatového řetězce. Tento soubor se musí zkopírovat do složky s jinými konfiguračními soubory (... \TwinCAT\3.1\Config\Io\EtherCAT). Při kopírování Visual

//0x6nnx Input Data of the Module (0x6000 - 0x6FFF)						
0x6000	RECORD			IN_GENERIC		
		0x01	UINT	IN_GEN_MOTOR1_POSITION	0	ro
		0x02	UINT	IN_GEN_MOTOR2_POSITION	0	ro
		0x03	UINT	IN_GEN_MOTOR3_POSITION	0	ro
		0x04	UINT	IN_GEN_INFO_PARAM1	0	ro
		0x05	BOOL	IN_GEN_MOTOR1_DIRECTION	0	ro
		0x06	BOOL	IN_GEN_MOTOR2_DIRECTION	0	ro
		0x07	BOOL	IN_GEN_MOTOR3_DIRECTION	0	ro
		0x08	BOOL	IN_GEN_Bit1	0	ro
		0x09	BOOL	IN_GEN_Bit2	0	ro
		0x0A	BOOL	IN_GEN_Bit3	0	ro
		0x0B	BOOL	IN_GEN_Bit4	0	ro
		0x0C	BOOL	IN_GEN_Bit5	0	ro
//0x7nnx Output Data of the Module (0x7000 - 0x7FFF)						
0x7000	RECORD			OUT_GENERIC		
		0x01	UINT	OUT_GEN_MOTOR1_POSITION	0	rw
		0x02	UINT	OUT_GEN_MOTOR2_POSITION	0	rw
		0x03	UINT	OUT_GEN_MOTOR3_POSITION	0	rw
		0x04	UINT	OUT_GEN_INFO_PARAM1	0	rw
		0x05	BOOL	OUT_GEN_MOTOR1_DIRECTION	0	rw
		0x06	BOOL	OUT_GEN_MOTOR2_DIRECTION	0	rw
		0x07	BOOL	OUT_GEN_MOTOR3_DIRECTION	0	rw
		0x08	BOOL	OUT_GEN_Bit1	0	rw
		0x09	BOOL	OUT_GEN_Bit2	0	rw
		0x0A	BOOL	OUT_GEN_Bit3	0	rw
		0x0B	BOOL	OUT_GEN_Bit4	0	rw
		0x0C	BOOL	OUT_GEN_Bit5	0	rw

Obr. 3.8: Tvar EtherCAT řetězce

Studio musí mít zavřené okno, jinak to nenačte zkopírovaný soubor. Jestli ESI soubor není zkopírován do správné složky, v prostředí Visual Studio dojde k inicializaci slavu s nedefinovanou strukturou řetězce.

Úpravy a inicializace

Po vygenerování souborů softwarem SSC Tool je nutno přepsat několik funkcí ve zdrojovém souboru s názvem XMC_ESC.c. Veškeré přepsané funkce jsou znázorněny na obrázku 3.9.

Funkce *memcpy* provádí následující operace:

- kopírování vstupních dat z lokální paměti do paměti ESC - využití ve funkci *APPL_InputMapping*,
- kopírování výstupních dat z paměti ESC do lokální paměti - využití ve funkci *APPL_OutputMapping*

Funkce *APPL_Application* se cyklicky volá v hlavní smyčce (synchronizace vypnuta) nebo v přerušení (synchronizace zapnuta). Tato funkce pracuje s funkcí *process_app*, která provádí nastavení výstupů a čtení vstupů.

Následujícím důležitým krokem je nastavení pinů MCU. Pomocí nástroje *Manual Pin Allocator* a schématu DPS se provede inicializace pinu modulu EtherCAT. Příklad nastavování pinů je na obrázku 3.10.

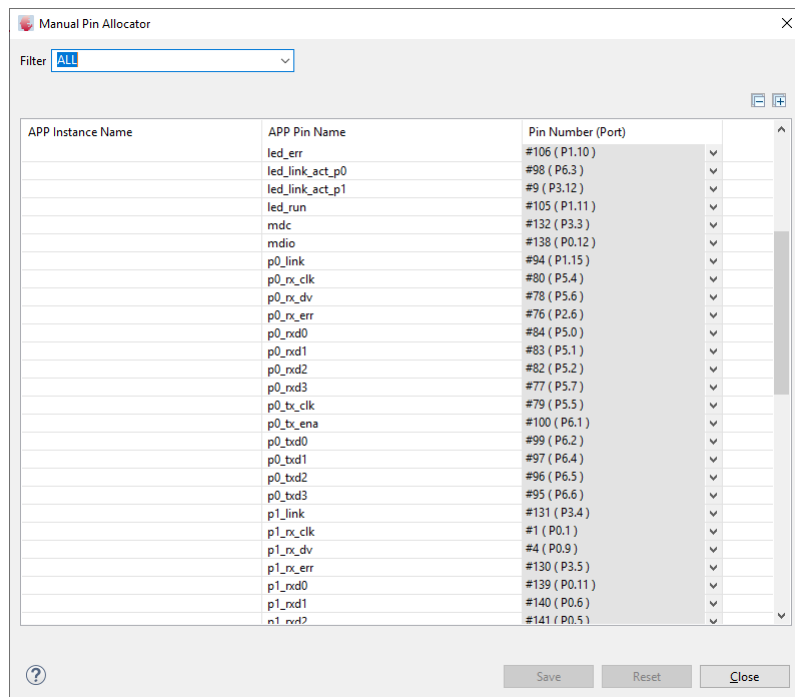
V tomto stádiu vznikl základní firmware, který se bude následovně rozšiřovat.

```

263 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
264 /**
265 \param    pData pointer to input process data
266
267 \brief    This function will copies the inputs from the local memory to the ESC memory
268           to the hardware
269 */////////////////////////////////////////////////////////////////////////////////////////////////////////////////
270 void APPL_InputMapping(UINT16* pData)
271 {
272     memcpy(pData, &(((UINT16 *)&IN_GENERIC0x6000)[1]), sizeof(IN_GENERIC0x6000)-2);
273 }
274
275 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
276 /**
277 \param    pData pointer to output process data
278
279 \brief    This function will copies the outputs from the ESC memory to the local memory
280           to the hardware
281 */////////////////////////////////////////////////////////////////////////////////////////////////////////////////
282 void APPL_OutputMapping(UINT16* pData)
283 {
284     memcpy(&(((UINT16 *)&OUT_GENERIC0x7000)[1]), pData, sizeof(OUT_GENERIC0x7000)-2);
285 }
286
287 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
288 /**
289 \brief    This function will called from the synchronisation ISR
290           or from the mainloop if no synchronisation is supported
291 */////////////////////////////////////////////////////////////////////////////////////////////////////////////////
292 void process_app(TOBI7000 *OUT_GENERIC, TOBI6000 *IN_GENERIC);
293 void APPL_Application(void)
294 {
295     process_app(&OUT_GENERIC0x7000, &IN_GENERIC0x6000);
296 }

```

Obr. 3.9: Změny souboru XMC_ESC.c



Obr. 3.10: Alokace pinů EtherCAT modulu

3.3.2 Implementace řízení

P-regulátor polohy je spojitým regulátorem. MCU pracuje s diskretními hodnotami, tedy je diskretním systémem. Aby se mohl P-regulátor použít ve firmwaru, musí se

ze spojitého regulátoru udělat diskrétní. To se dělá pomocí dvou transformací:

- zpětná Laplaceova transformace - převod ze spojitého systému do časového,
- Z-transformace - převod z časového systému do diskrétního.

$$F_{R-P-polohy}(z) = Z\{L^{-1}\{F_{R-P-polohy}(p)\}\} = 41,98375. \quad (3.1)$$

Rovnice 3.1 vychází ze rovnice 2.13 a prezentuje diskrétní P-regulátor polohy, který se může implementovat do firmwaru. Implementace je udělána ve tvaru funkce (*calc_position_manipulated_variables*), která se volá v hlavní smyčce. Parametrem funkce je regulační odchylka polohy, která se vypočítává z požadované a aktuální pozice. *Calc_position_manipulated_variables* vrací akční veličinu polohového regulátoru, na základě které se nastavují duty cycle PWM.

Dalšími funkcemi pro řízení jsou:

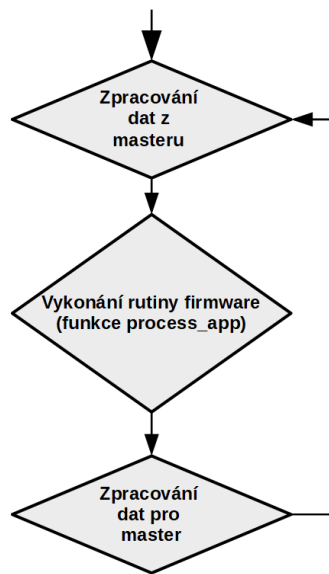
- PWM_Start - spouští všechny PWM kanály najednou kvůli eliminaci zpoždění mezi kanály,
- PWM_SetMotor_DutyCycle - nastavuje *duty cycle* určitých kanálů v závislosti na směru a čísle motoru.

Kvůli získání informace o aktuální poloze rotoru motoru je napsána funkce *calc_motor_current_position*. Parametrem funkce je čítač, který se inkrementuje při příchodu pulzů z enkodéru. Výsledek se vrací do smyčky a odesílá se do masteru jako informace o aktuálním stavu polohy.

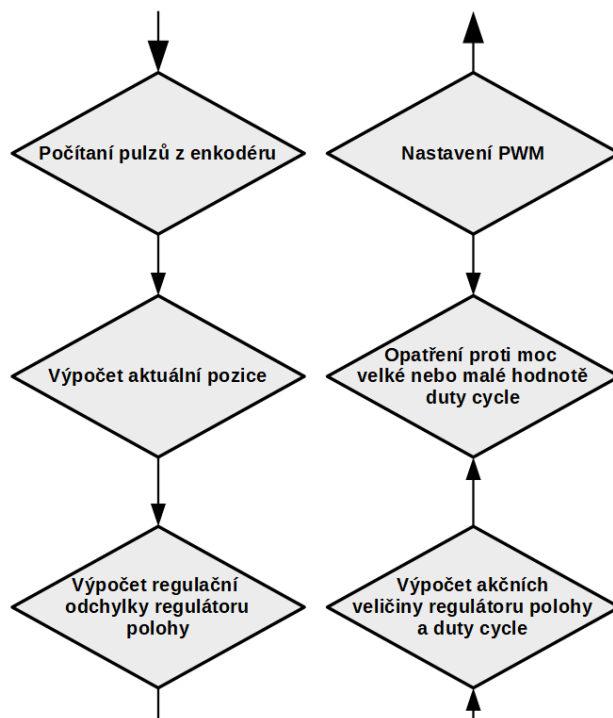
3.3.3 Hlavní smyčka firmwaru

Obecně firmware funguje tak, že se po inicializaci parametrů motorů a periférií spustí nekonečná smyčka, která je znázorněna na obrázku 3.11.

Řízení motoru probíhá ve funkci *process_app*. Na obrázku 3.12 je nakresleno pořadí vykonání určitých kroků ovládání stejnosměrných motorů.



Obr. 3.11: Hlavní smyčka firmwaru



Obr. 3.12: Funkce process_app

4 Oživování

Tato sekce popisuje postup oživování vyrobeného modulu. Uvedená činnost se skládá z několika důležitých kroků:

- kontrola modulu, případná oprava postavených či chybějících součástek,
- kontrola nežádoucích zkratů,
- opatrné zapínání DPS s proměřením důležitých napětí,
- nahrávání napsaného firmwaru,
- kontrola hlavních sekcí DPS, případná oprava firmwaru.

Během prvních dvou kroků probíhá elektrická a vizuální kontrola. Po provedení oprav je možné k navrženému modulu připojit napájecí napětí. Karta má dva přívodní konektory: JP9 (napájení motorů) a JP13 (napájení procesorové části). Napájení má být rozdělené, tedy že se mají použít dva jednobáňové zdroje stejnosměrného napětí nebo jeden dvou a více kanálový zdroj. Při oživování se používá zdroj GW INSTEK GPD-4303S s následujícím nastavením proudového omezení a napětí:

- napájení procesorové části 50 mA / 3 V,
- napájení motorů 1 A / 24 V.

Můžeme si všimnout, že napájecí napětí procesorové části je nastaveno na velice nízkou hodnotu. Je to tak uděláno kvůli tomu, aby bylo možné kontrolovat hodnoty napětí v napájecí části karty. Postupně se začne zvyšovat napětí a průběžně se kontrolovat výstupní napětí a napětí na pinech EN součástek LMZ14203 a LMZ12003. Minimální hodnota napětí, při které se zapnou všechny regulátory napětí, je 18.7 V. Po spouštění regulátorů se objevilo, že modulu nestačí proud (zapnutí proudového omezení). Proudové omezení pak se změnilo na 150 mA.

Teď je nutno naprogramovat osazené MCU. Za použití programátoru z karty *XMC4800 Relax Kit* a softwaru *J – Link Commander* se provede programování následujícím způsobem:

- zadat příkaz *Connect*,
- zkontrolovat název zařízení (jestli použité MCU nesedí s uvedeným, po zadání otazníku zvolit správný typ),
- zvolit interface *SWD*,
- nechat rychlost interface na 4000 kHz,
- zadat příkaz *loadfile* s cestou nahrávaného souboru a jeho formátem (např. *loadfile ...\XMC48_MC_ECACAT\Release\ETHCAT_SSC_XMC48.hex*),
- zresetovat modul např. vypnutím napájení.

V tuto chvíli se oživování modulu může považovat za dokončené, a může se přejít k testování.

5 Testování modulu

Cílem testování je ujistit se v tom, že navržený modul se dá používat pro ovládání motorů s možností řešení polohovacích úloh. K tomu je potřeba mít notebook se 100 Mbit síťovou kartou, alespoň jeden motor a enkodér. Při testování se používal motor MEZ K6A1 a enkodér SCANCON 2REB500-667402.

Tab. 5.1: Parametry motoru

napájecí napětí	24 V
proud	0.19 A
max. otáčky	5000
výkon	2W

Tab. 5.2: Parametry enkodéru

napájecí napětí	4.5 V - 30 V DC
max. proud	36 mA
počet pulzů na otáčku	1024

5.1 Test komunikace

Spuštění komunikace probíhá následujícím způsobem:

- propojit Ethernet kabelem modul a PC - připojení ke konektoru RJ1,
- vytvořit master (viz. kapitola 3.2),
- tlačítkem *Toggle Free Run State* převést komunikace do operačního modu.

Po provedení výše uvedených kroků je nutno se podívat na modul a zkontrolovat LED diody D1 a D2. Trvale svítící zelená dioda znamená, že komunikace běží a dá se provádět nastavení jednotlivých parametrů ethercatového řetězce. Tato část komunikace funguje.

Dále se musí otestovat druhá část komunikace. Ke konektoru RJ2 se připojí další EtherCAT slave zařízení. Znova se provede inicializace sítí. Jestli se oba moduly načtou a odpovídají na příkazy z masteru, komunikace je plně funkční, a testování komunikace se může považovat za dokončené.

5.2 Test můstkových driverů

Původně firmware byl napsán pro zkoušení modulu s kartáčovým motorem. Pro kartáčové motory se používají můstkové drivery U3, U4 a U5. Před tím, než se k modulu připojil motor, se zkontrolovala správnost komutace a to tak, že se spustily PWM kanály pro pohyb na jednu stranu a pak na druhou. Vstupy můstkových driverů po každém nastavení se zkontrolovaly osciloskopem. Při tomto testování se taky otestovaly můstkové drivery U6 a U7. Během testování se zjistilo, že uvedené dva drivery jsou na sobě závislé a některé kanály PWM modulu se inicializují jako invertovaný signál. Uvedené body při ovládání motoru jsou nežádoucími. V následující verzi karty se tyto problémy se odstraní.

Po dokončení testování všech můstkových driverů firmware se upravil pro testování úloh polohování.

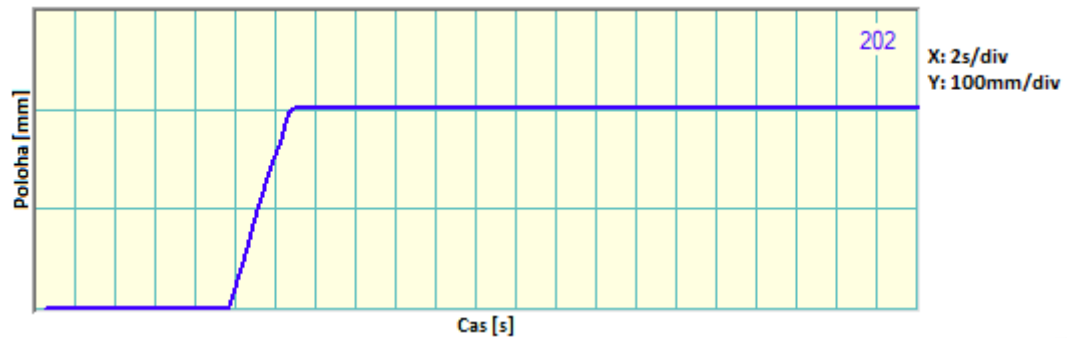
K dispozici jsou dva motory DC a jeden inkrementální enkodér. Enkodér se spojí s jedním z motorů. Finální verze firmwaru je taková, že motor s inkrementálním enkodérem řeší polohovací úkoly a druhý motor přesně kopíruje chování motoru s enkodérem.

Motor bez enkodéru se může připojit k driveru U3 nebo U4. Motor s enkodérem se připojí k můstkovému driveru U5. Uvedený enkodér má 3 kanály, kde kanál je představen jedním drátem. Enkodér se připojí ke konektoru JP2 a tvoří pro motor určitý zatěžovací moment, který je dán tuhostí hřídele. S tímto zatížením firmware počítá, jinak už nepočítá s vnějšími zatěžovacími momenty.

5.3 Test polohovací úlohy

Testování spočívá v tom, že pomocí komunikace EtherCAT se nastaví hodnota posunu, a motor se má pohnout o tuto hodnotu v určitém směru. Během tohoto testování se testuje enkodér, protože motor bez zpětné vazby se nezastaví na požadované pozici. Během testování se uvažuje, že na hřídel motoru je nainstalován fiktivní kotouč s průměrem 80 mm. Komutace motoru se provádí signálem PWM s frekvencí 20 kHz.

Aby se pak dalo porovnat chování reálného motoru s navrženým modelem při skokové změně pozice, musí se počítat s tím, že model za otáčku teoreticky zdolá vzdálenost 1 mm. U reálného motoru se počítá s nasazením kotouče o průměru 80 mm. Aby vzdálenosti posunu byly srovnatelné, reálný motor musí provést posun o 200 mm, jelikož vzdálenost posunu modelu z obrázku 2.7 je 2.5 mm. Do parametru *OUT_GEN_MOTOR1_POSITION* se zadá vzdálenost 200 mm, a motor okamžitě začne vykonávat pohyb. Průběh pohybu se dá pozorovat z parametru *IN_GEN_MOTOR1_POSITION*.

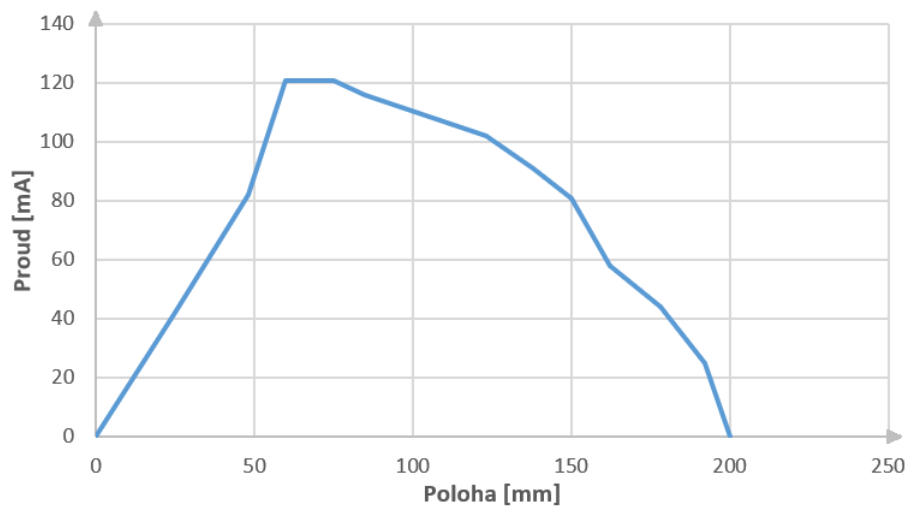


Obr. 5.1: Průběh pohybu o vzdálenost 200mm

Z obrázku 5.1 je vidět, jak probíhal pohyb motoru. Při porovnání pohybu motoru a modelu z obrázku 2.7 je vidět, že vypadají přibližně stejně až na to, že model motoru zdolá stejnou vzdálenost 8x rychleji než reálný motor. Je dáno následujícími důvody:

- model není schopen přesně popsat chování DC motoru,
- model je tvořen na základě podobného motoru, než byl použit při testování,
- firmware limituje duty cycle motoru kvůli zachování stability pohybu a ochraně motoru.

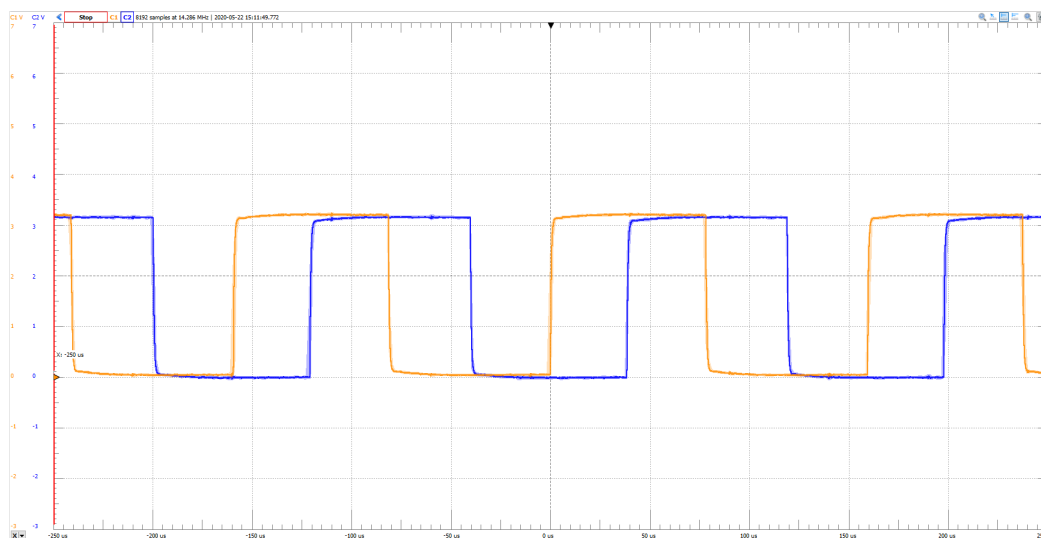
Během kontroly pohybu se provádělo měření proudu, které je zrekonstruováno na obrázku 5.2.



Obr. 5.2: Průběh proudu při pohybu o vzdálenost 200mm

Z důvodu toho, že se motor zastavuje na požadované pozici, se zapojení enkodéru může považovat za funkční. Obrázek 5.3 prezentuje signály dvou kanálů použitého

enkodéru ze strany MCU. Jednotlivé kmity v průběhu jsou dány vibrací stolu.



Obr. 5.3: Signály z enkodéru

Po provedení polohovacího testu se motor a enkodér rozpoj. Jestli není připojen druhý motor, musí se připojit. Po spuštění a rozběhnutí motorů se zkontroluje moment obou motorů. Z výsledků měření vyplývá, že motor, který má vývod nejbliž k napájení, vykazuje největší moment. Je to z důvodu, že tímto motorem proteče praktický veškerý proud dodávaný zdrojem. Tím pádem je při pohybu lepší ovládat motory po jednom, nebo upravit modul a napájet každý motor zvlášť.

5.4 Test ochran

Navržený modul obsahuje tři ochrany:

- tepelná ochrana,
- ochrana proti velkému proudu,
- ochrana proti vysokému napájecímu napětí motorů.

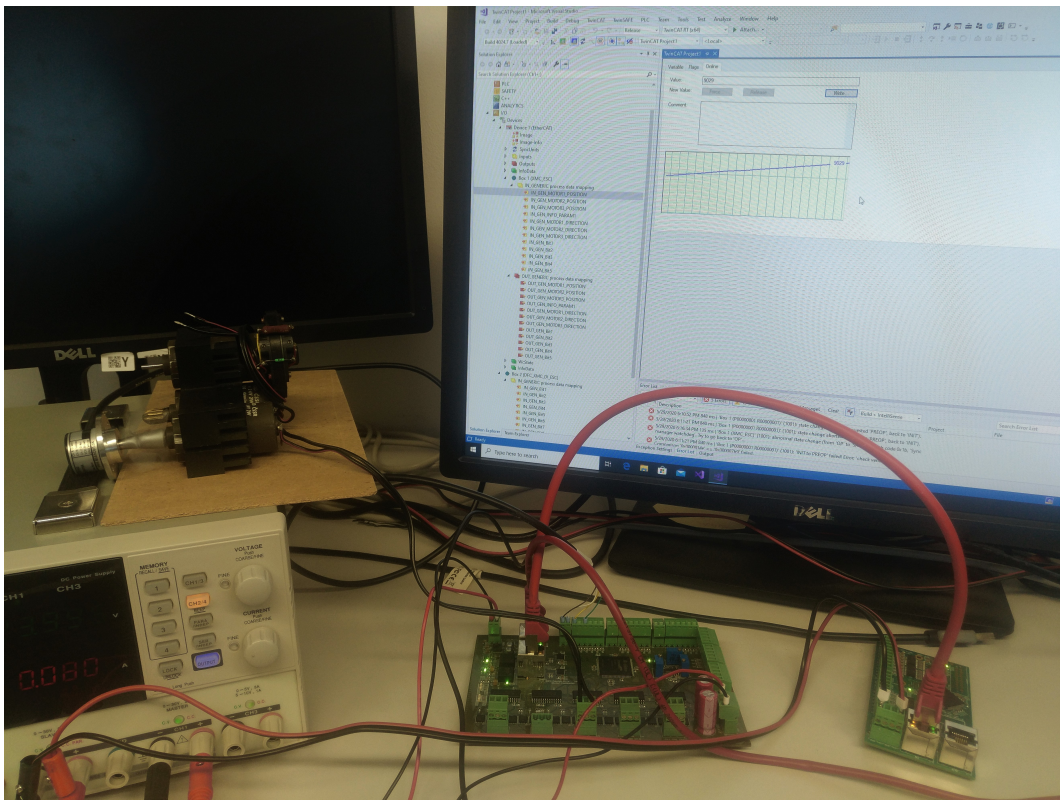
Hodnoty, při kterých se mají ochrany aktivovat, se nastavují pomocí potenciometrů.

Testování tepelné ochrany se provádělo při pokojové teplotě. Teplota kolem modulu se měří snímačem teploty, a naměřená hodnota se vysílá analogovým signálem. Testování se provedlo tak, že pomocí potenciometru R231 se změnila napěťová úroveň, při které by měla ochrana zareagovat. Ochrana zareagovala a odpojila můstkové drivery.

Před testováním ochrany motorů proti velkému napájecímu napětí je nutno odpojit motory, aby nedošlo k jejich poškození. Pak se začne zvedat napětí motorů,

dokud se nesezne ochrana. Ochrana se sepnula při hodnotě napětí 31.6 V. Hodnotu napětí, při které se má zapnout ochrana, se dá nastavit potenciometrem R236.

Proudová ochrana je založena na měření proudu společným měřicím rezistorem. Uvedený rezistor je propojen s měřením proudu nejbližšího polovodičového H-můstku pomocí propojek. Z důvodu toho, že byl modul kompletně osazen kvůli testování všech můstků, nejde použít společný měřicí rezistor pro proudovou ochranu. Aby se dalo použít tento rezistor, měl by se kompletně odpájet nejbližší poloviční H-můstek. Toto je velice problematické, protože SMD součástky nemají termální plošky, a tloušťka mědi je cca $70 \mu\text{m}$. Při pokusu o odpájení může hrozit poškozením modulu. Z tohoto důvodu je testování proudové ochrany obdobné testování tepelné ochrany. Pomocí potenciometru R228 se začne měnit napěťová úroveň, při které by měla ochrana zareagovat. Ochrana zareagovala a odpojila můstkové drivery. V následující verzi modulu se problém sdíleného měření odstraní a proudová ochrana bude nezávislá na můstkovém driveru U7.



Obr. 5.4: Testování modulu pro řízení stejnosměrných motorů

Závěr

Tato diplomová práce se zabývala realizací a následným testováním modulu pro ovládání stejnosměrných motorů s možností komunikace s nadřazeným systémem.

První část práce obsahuje popis kartáčových a bezkartáčových motorů, způsoby jejich řízení, regulace a popis zapojení výkonového stupně pro ovládání uvedených motorů. Modul umožňuje řešit polohovací úkoly, proto jsou v této sekci popsány i snímače polohy a jsou shrnuté do tabulky, usnadňující volbu vhodného typu senzoru. Modul je řízen z nadřazeného systému, proto jsou zde krátce rozepsány často používané průmyslové sběrnice. Pro komunikaci s nadřazeným systémem zvolena sběrnice EtherCAT.

Dále se práce zabývá tvorbou matematického modelu motoru, návrhem regulace a simulací. Následuje návrh elektrického schématu a desky plošných spojů. Simulace se prováděly v programu MATLAB Simulink, s cílem otestovat zvolený model řízení. Z výsledků simulací vznikly jasné požadavky na modul pro zajištění polohovacích úloh. Následně je popsána tvorba schématu s podrobným postupem návrhu elektrického zapojení. Poslední část sekce se zabývá návrhem desky plošného spoje.

Následující část diplomové práce podrobněji rozebírá vybranou komunikaci a popisuje testovací verzi programového vybavení modulu. Po podrobnějším rozboru komunikace po sběrnici EtherCAT se sekce zabývá postupem vytvoření řídicí (master) a podřízené (slave) jednotky. Pro vytvoření podřízené jednotky se využilo programové vybavení SSC Tool. Dále je popis implementace řízení a hlavní nekonečné smyčky programového vybavení. Část programového vybavení pro řešení polohovacích úloh je napsána především pro testování celého modulu, tedy řeší pouze základní pohyby ovládané z připojeného počítače PC pomocí nástroje Visual Studio - TwinCAT3.

Po návrhu desky plošných spojů a programového vybavení následuje popis ožívování a testování modulu. Během ožívování se do modulu pro ovládání stejnosměrných motorů naprogramuje vytvořené programové vybavení a provede se testování primárních funkcí. Z výsledků testů je zřejmé, že navržený modul je schopen řídit stejnosměrné motory, a navíc řešit polohovací úkoly. Zadání diplomové práce bylo ve všech bodech splněno.

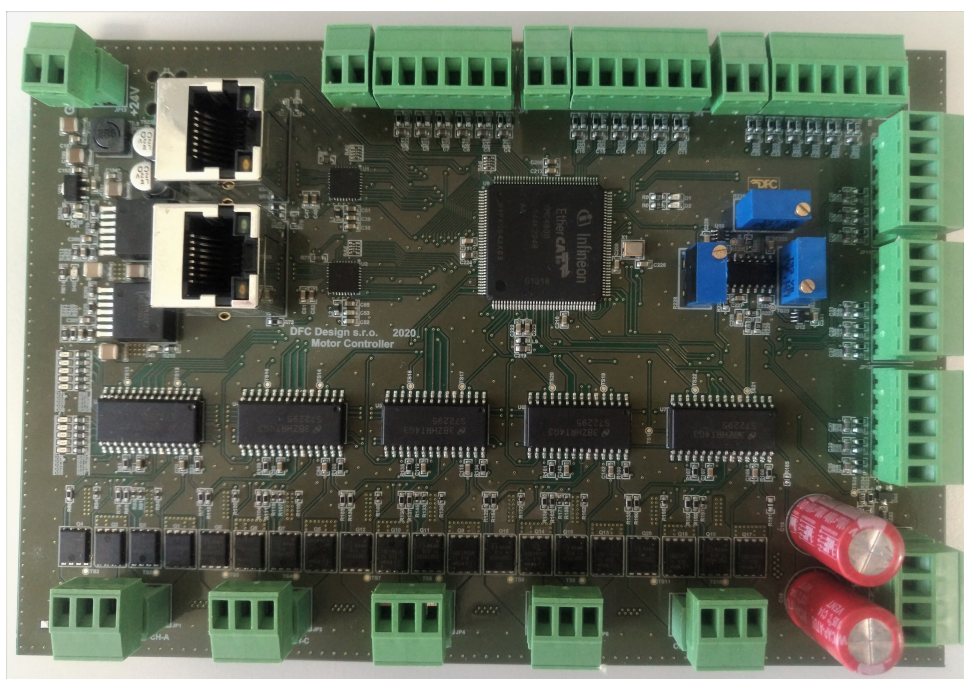
Primárně se modul navrhoval pro ovládání tří kartáčových motorů, ale v průběhu vývoje byl rozšířen o možnost ovládání dvou bezkartáčových motorů.

Návrh celého řešení vycházel z požadavků firmy DFC Design, které poskytla prostředky pro testování komunikace a plánuje její využití při řešení vlastních projektů.

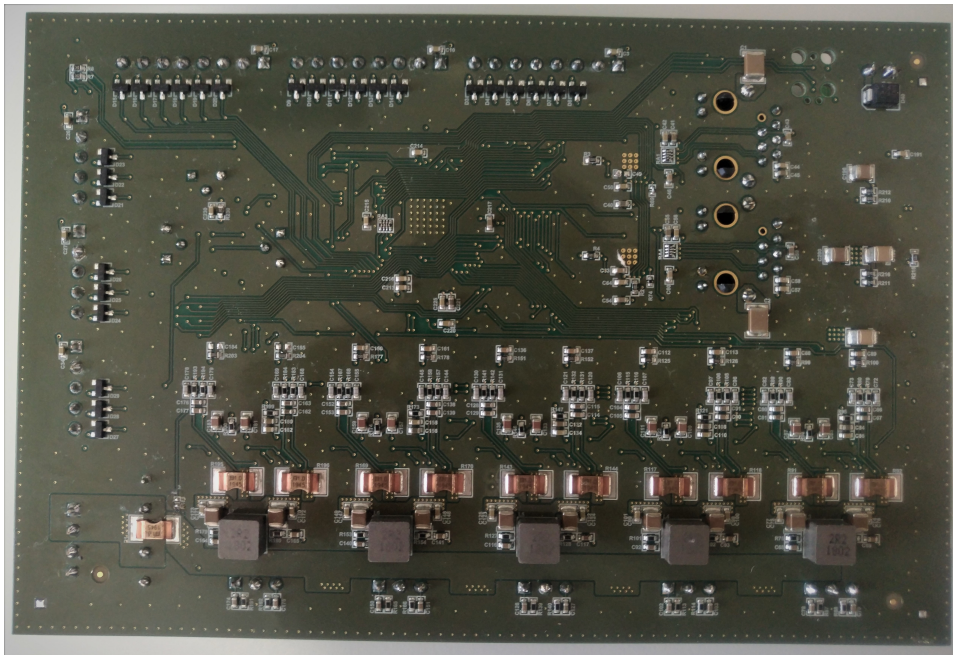
Výsledkem práce je modul pro ovládání stejnosměrných motorů s následující krátkou specifikací.

Tab. 5.3: Specifikace výsledného modulu

napájení procesorové části	18.7 V - 25 V
napájení motorů	12 V - 48 V
max. proud motory	15 A
počet DC motorů	3
počet EC motorů	2
ochrany motorů a DPS	Ano
řízení	2Q (4Q)



Obr. 5.5: Modulu pro řízení stejnosměrných motorů - Top



Obr. 5.6: Modulu pro řízení stejnosměrných motorů - Bot

Literatura

- [1] Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií: *Mikroprocesorová technika v aplikacích řízení elektrických pohonů* [online], [cit. 01. 05. 2020]. Dostupné z URL:
<http://www.crr.vutbr.cz/kurzy-seminare/mikroprocesorova-technika/kurz-mikroprocesorova-technika-v-aplikacich-rizeni-elektric>
- [2] UZIMEX PRAHA, spol. s r.o.: *MALÉ STEJNOSMĚRNÉ MOTORY MAXON* [online], Praha: 2002, poslední aktualizace 18. 07. 2002 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.uzimex.cz/soubory/20070103_maxon_serial.pdf
- [3] BROŽ, V.: *Životnost moderních malých stejnosměrných motorů* [online], poslední aktualizace 09. 2001 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.uzimex.cz/soubory/20011001_mm_2001-09.pdf
- [4] Learn Engineering: *DC Motor, How it works?* [online], poslední aktualizace 22. 09. 2014 [cit. 28. 04. 2020]. Dostupné z URL:
<https://www.youtube.com/watch?v=LAtPHANefQo>
- [5] VOJÁČEK, A.: *Motory a jejich řízení s MCU - 2. část - spínací můstky a jejich připojení k MCU* [online], poslední aktualizace 01. 03. 2008 [cit. 28. 04. 2020]. Dostupné z URL:
<https://automatizace.hw.cz/motory-jejich-rizeni-s-mcu-2-cast-spinaci-mustky-jejich-pripojeni-k-mcu>
- [6] Learn Engineering: *Brushless DC Motor, How it works ?* [online], poslední aktualizace 13. 10. 2014 [cit. 28. 04. 2020]. Dostupné z URL:
<https://www.youtube.com/watch?v=bCEiOnuODac>
- [7] Technology – short and to the point: *Maxon Sensor* [online], poslední aktualizace 11. 2014 [cit. 28. 04. 2020]. Dostupné z URL:
https://www.maxongroup.com/medias/sys_master/root/8815461793822/sensor-Technology-short-and-to-the-point-14-EN-038-039.pdf?attachment=true
- [8] Technology – short and to the point: *Maxon Motor Control* [online], poslední aktualizace 11. 2014 [cit. 28. 04. 2020]. Dostupné z URL:
https://www.maxongroup.com/medias/sys_master/8815459401758.pdf
- [9] Ústav mikroelektroniky, Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně: *MNEN - Průmyslové sběrnice* [online],

- [cit. 01. 05. 2020]. Dostupné z URL:
http://www.umel.feec.vutbr.cz/mnen/Files/kap_4.pdf
- [10] ŠPONAR, R.: *Vlastnosti a užití průmyslových sběrnic* [online], poslední aktualizace 05. 04. 2004 [cit. 01. 05. 2020]. Dostupné z URL:
<http://www.elektrorevue.cz/clanky/04019/index.html>
- [11] ADÁMEK, M.: *Referenční model ISO/OSI* [online], [cit. 01. 05. 2020]. Dostupné z URL:
<http://www.umel.feec.vutbr.cz/adamek/komp/data/iso.htm>
- [12] KRYŠTŮFEK, J.: *Průmyslová komunikační síť Profibus* [online], [cit. 01. 05. 2020]. Dostupné z URL:
<http://www1.fs.cvut.cz/cz/u12110/site/profibus/>
- [13] SIEMENS: *Čas na změnu – přejděte z PROFIBUSu na PROFINET* [online], [cit. 28. 04. 2020]. Dostupné z URL:
<https://new.siemens.com/cz/cs/products/automation/industrial-communication/profinet/pb-2-pn.html>
- [14] ROSTAN, M.: *Jak se ve skutečnosti vybírají průmyslové sběrnice* [online], poslední aktualizace 01. 02. 2018 [cit. 28. 04. 2020]. Dostupné z URL:
<https://automatizace.hw.cz/jak-se-ve-skutecnosti-vybiraji-prumyslove-sbernice.html>
- [15] Texas Instruments: *SM72295EVM User's Guide* [online], poslední aktualizace 05. 2018 [cit. 01. 05. 2020]. Dostupné z URL:
<http://www.ti.com/lit/ug/snvu473/snvu473.pdf>
- [16] Cadence PCB Solution: *MII and RMII Routing Guidelines for Ethernet* [online], poslední aktualizace 24. 07. 2019 [cit. 28. 04. 2020]. Dostupné z URL:
<https://resources.pcb.cadence.com/blog/2019-mii-and-rmii-routing-guidelines-for-ethernet>
- [17] Texas Instruments: *Ethernet System Hardware on AM-Class Devices* [online], [cit. 28. 04. 2020]. Dostupné z URL:
https://training.ti.com/sites/default/files/docs/ETH_System_HW_on_AM_Class_Devices_Slides.pdf
- [18] LACHMAN, M., MENDŘICKÝ, R.: *Modelování polohových servomechanismů v prostředí Matlab/ Simulink* [online], poslední aktualizace 27. 11. 2013 [cit. 28. 04. 2020]. Dostupné z URL:
<http://www.ksa.tul.cz/getFile/id:3431>

- [19] MathWorks: *DC Motor* [online], [cit. 01. 05. 2020]. Dostupné z URL:
<https://www.mathworks.com/help/physmod/sps/ref/dcmotor.html>
- [20] RICHTER, M.: *Ziegler-Nicholsova metoda návrhu regulátorů* [online], poslední aktualizace 24. 03. 2009 [cit. 01. 05. 2020]. Dostupné z URL:
http://www.uamt.feec.vutbr.cz/richter/vyuka/0809_BRR1/labs/zn_reg.html.cs
- [21] KOZUBÍK, M.: *METODY ANTI-WINDUP PŘI ŘÍZENÍ LINEÁRNÍCH-DYNAMICKÝCH SYSTÉMŮ* [online], Brno: 2017, poslední aktualizace 29. 05. 2017 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=148088
- [22] HADÁMEK, J.: *SLAVE MODUL VYUŽÍVAJÍCÍ KOMUNIKAČNÍ PROTOKOL ETHERCAT* [online], Brno: 2018, poslední aktualizace 21. 05. 2018 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=172474
- [23] EtherCAT Technology Group: *The Ethernet Fieldbus* [online], poslední aktualizace 2009 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.ethercat.org/pdf/english/EtherCAT_Introduction_0905.pdf
- [24] Infineon Technologies: *XMC4800 EtherCAT APP SSC Slave Example* [online], poslední aktualizace 2016 [cit. 01. 05. 2020]. Dostupné z URL:
https://www.infineon.com/dgdl/Infineon-Getting_Started_XMC4800_Relax_EtherCat_APP_Slave_SSC_Example-GS-v01_03-EN.pdf?fileId=5546d462636cc8fb01639266b44447a3
- [25] Fakulta Strojní, Vysoká škola báňská - Technická univerzita Ostrava: *Syntéza jednorozměrových diskrétních regulačních obvodů* [online], [cit. 01. 05. 2020]. Dostupné z URL:
<http://books.fs.vsb.cz/cislicovaregulace/data/images/pdf/8kapitola.pdf>

Přílohy

Příloha	kořenový adresář
└─ Altium_Designer	
└─ EtherCAT_MotorController	Pdf a Gerber soubory
└─ EtherCAT_MotorController_Sablona.....	Pdf soubor s šablonou
└─ Firmware	
└─ XMC48_MC_ECAT	
└─ Release.....	Soubor s příponou .hex
└─ Zdrojove_Kody.....	Zdrojové kódy
└─ Matlab.....	Simulační soubory
└─ TwinCAT_master.....	Visual Studio projekt - EtherCAT master