

ZÁPADOČESKÁ UNIVERZITA V PLZNI

---

Fakulta elektrotechnická  
Katedra zadávající téma diplomové práce

# DIPLOMOVÁ PRÁCE

Návrh řídicího algoritmu pro elektrickou motokáru

Autor práce: **Jan Blaško**  
Vedoucí práce: **Ondřej Suchý**

---

2022

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2022/2023

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jan BLAŠKO**  
Osobní číslo: **E20N0049P**  
Studijní program: **N0714A060013 Elektronika a informační technologie**  
Specializace: **Elektronika**  
Téma práce: **Návrh řídicího algoritmu pro elektrickou motokáru**  
Zadávající katedra: **Katedra elektroniky a informačních technologií**

## Zásady pro vypracování

1. Cílem práce je navržení řídicího algoritmu pro ovládání elektrické motokáry. Řídicí algoritmus bude ovládat jak výkonový střídač tak i podpůrné periferie (displej, baterie, světla).
2. Stručně popište hardware řídicí jednotky motokáry.
3. Proveďte výběr vhodné řídicí metody pro pohon elektrické motokáry.
4. Implementujte vybrané řízení do řídicí jednotky a experimentálně ověřte.
5. Proveďte rozchození externích periférií (displej, baterie, pedály).
6. Vytvořte FMEA analýzu možných stavů a zapracujte je do řídicího algoritmu.

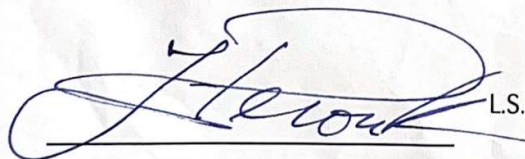
Rozsah diplomové práce: **40 – 60**  
Rozsah grafických prací:  
Forma zpracování diplomové práce: **elektronická**

Seznam doporučené literatury:

1. KERNIGHAN, Brian W. a Dennis M. RITCHIE. Programovací jazyk C. 2. vydání. Přeložil Zbyněk ŠÁVA. Brno: Computer Press, 2019. ISBN 978-80-251-4965-2.

Vedoucí diplomové práce: **Ing. Ondřej Suchý**  
Research and Innovation Centre for Electrical  
Engineering

Datum zadání diplomové práce: **7. října 2022**  
Termín odevzdání diplomové práce: **26. května 2023**



L.S.

**Prof. Ing. Zdeněk Peroutka, Ph.D.**  
děkan



**Doc. Ing. Jiří Hammerbauer, Ph.D.**  
vedoucí katedry

V Plzni dne 7. října 2022

## **Abstrakt**

Tato diplomová práce se zabývá implementací řídicího algoritmu pro elektrickou motokáru. Cílem práce je výběr vhodné řídicí metody, oživení a ověření funkce nezbytných periférií pro vybrané řízení. Zprovoznění komunikací s displejem a s balancéry na bateriích. Dále se zabývá implementací a laděním vybraného řízení. Součástí práce je také FMEA příčin a důsledků jednotlivých stavů, které mohou nastat.

## **Klíčová slova**

PMSM, Elektrická motokára, Vektorové řízení, FMEA, TMS320F28335, PI regulátory, Feedforward, PWM, RS-485, CAN

**Abstract**

This thesis deals with the implementation of a control algorithm for an electric go-kart. The objective of this work is to select a suitable control method, revive and verify the functionality of necessary peripherals for the selected control. It involves setting up communication with the display and the battery balancers. It also involves the implementation and tuning of the selected control. The thesis also includes FMEA (Failure Mode and Effects Analysis) of causes and consequences of individual states that may occur.

**Key Words**

PMSM, Electric go-kart, Field oriented control, FMEA, TMS320F28335, PI controllers, Feedforward, PWM, RS-485, CAN

## **Poděkování**

Rád bych poděkoval vedoucímu diplomové práce Ondřeji Suchému za odbornou přípravu a metodologickou pomoc při zpracování mé práce a všem z katedry výkonové elektroniky a strojů, kteří mi během zpracování poskytli odborné konzultace k problematice.

# Obsah

Úvod.....	- 1 -
1 Hardware řídicí jednotky .....	- 2 -
1.1 Mikrokontrolér TMS320F28335 .....	- 2 -
1.2 Senzory .....	- 3 -
1.2.1 Proudové senzory .....	- 3 -
1.2.2 Napěťový senzor .....	- 3 -
1.2.3 Teplotní senzory .....	- 3 -
1.2.4 Senzor polohy rotoru .....	- 3 -
1.3 Battery management systém .....	- 4 -
1.3.1 Komunikační protokol RS-485 .....	- 4 -
1.3.2 Obsah odesílaného paketu .....	- 5 -
1.3.3 Obsah přijímaného paketu .....	- 5 -
1.4 Displej .....	- 6 -
1.4.1 CAN sběrnice.....	- 7 -
1.5 Pedály .....	- 7 -
1.6 Externí Flash paměť .....	- 8 -
2 Výběr řídicí metody pro PMSM.....	- 9 -
2.1 Synchronní stroj s permanentními magnety .....	- 9 -
2.2 Skalární řízení .....	- 9 -
2.3 Přímé řízení momentu DTC .....	- 10 -
2.4 Vektorové řízení.....	- 10 -
2.5 Popis transformací vektorového řízení.....	- 11 -
2.6 Clarkové transformace .....	- 11 -
2.7 Parkova transformace.....	- 13 -
2.8 PI regulátor.....	- 13 -
2.9 Feedforward .....	- 14 -
3 Konfigurace periférií použitých pro řízení .....	- 15 -
3.1 Generování PWM signálu .....	- 15 -
3.1.1 Konfigurace ePWM modulu.....	- 16 -

3.1.2	Experimentální ověření.....	- 18 -
3.2	Vyčítání analogových hodnot.....	- 19 -
3.2.1	Konfigurace ADC modulu.....	- 20 -
3.3	Měření otáček a polohy rotoru .....	- 21 -
3.3.1	Konfigurace eCAP modulu.....	- 22 -
3.3.2	Vyčtení absolutní polohy rotoru .....	- 23 -
3.3.3	Experimentální ověření eCAP konfigurace .....	- 24 -
3.3.4	Konfigurace eQEP modulu.....	- 25 -
3.3.5	Vyčtení rychlosti.....	- 25 -
4	Komunikace.....	- 26 -
4.1	Komunikace s BMS (RS-485) .....	- 26 -
4.1.1	Konfigurace SCI modulu .....	- 26 -
4.1.2	Rutina zpracovávání zpráv.....	- 27 -
4.1.3	Experimentální ověření.....	- 29 -
4.2	Komunikace s LCD displejem (CAN B).....	- 29 -
4.3	Debugging softwaru (CAN A).....	- 30 -
4.4	Kalibrace konstant.....	- 31 -
5	Oživení řídicí jednotky .....	- 32 -
5.1	Zatížení jednotky s RL zátěží.....	- 32 -
5.2	Testování jednotky s PMSM motorem .....	- 37 -
6	FMEA analýza.....	- 42 -
	Závěr .....	- 43 -
	Literatura.....	- 44 -
	Přílohy.....	- 46 -



## Seznam symbolů a zkratek

<b>Značka</b>	<b>Popisek</b>	<b>Jednotka</b>
<i>PMSM</i>	Synchronní motor s permanentními magnety	
<i>CAN</i>	Controller Area Network	
<i>FMEA</i>	Failure Modes and Effect Analysis	
<i>ePWM</i>	The enhanced Pulse Width Modulator	
<i>eCAP</i>	The enhanced Capture	
<i>eQEP</i>	The enhanced Quadrature Encoder Pulse	
<i>ADC</i>	Analog to Digital Converter	
<i>I</i>	Proud	[A]
<i>LiFeYPO4</i>	Lithium železo fosfátový akumulátor	
<i>U</i>	Napětí	[V]
$\Psi$	Magnetický tok	[Wb]
<i>T</i>	Teplota	[°C]
<i>t</i>	Čas	[s]
<i>NTC</i>	Negative temperature coefficient	
<i>GPIO</i>	General-purpose input/output	
<i>IRC</i>	Inkrementální rotační čidlo	
<i>BMS</i>	Battery Management System	
<i>CRC</i>	Cyclic Redundancy Check	
<i>DTC</i>	Direct torque controll	
<i>SOC</i>	Start Of Conversion	
<i>FIFO</i>	First In First Out	
<i>SCI</i>	Seriál Communications Interface	
<i>UART</i>	Universal Asynchronous receiver transmitter	
<i>JTAG</i>	Joint Test Action Group	

## Úvod

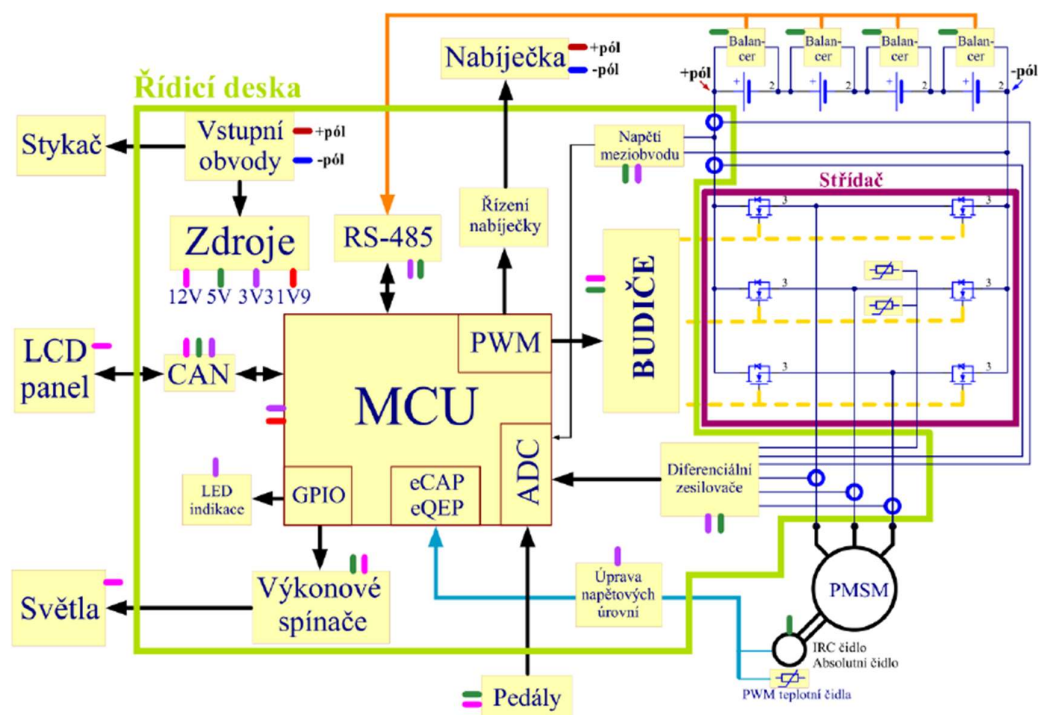
Elektrické motokáry jsou stále populárnější a jejich technologický vývoj pokračuje neustále. Tato práce je věnována vývoji softwaru pro řídicí jednotku elektrické motokáry. Historicky byla tato motokára osazena stejnosměrným motorem s permanentními magnety o výkonu 5kW a snižovacím pulzním měničem, který byl řízen mikrokontrolérem Atmel AVR. Nyní je pohon nahrazen synchronním motorem s permanentními magnety (PMSM) o totožném výkonu 5kW. K řízení se nyní využívá mikrokontrolér TMS320F28335 od firmy Texas Instruments, který řídí 3-fázový střídač složený z MOS-FET tranzistorů. Řídicí jednotka kromě řízení motoru obstarává také další periferie. Mezi tyto periferie patří displej, baterie, pedály, světla a ovládání nabíječky. Za tímto účelem je nutné vytvořit komplexní řídicí systém, který umožní přesné řízení pohonu společně s obsluhou všech periférií a zajistí, že motokára bude pro řidiče dostatečně bezpečná.

Práce je rozdělena na několik kapitol. První kapitola popisuje hardware řídicí jednotky a dalších částí jako je displej, pedály nebo balancovací jednotky. Druhá kapitola obsahuje teoretický základ řídicích metod a výběr vhodné metody řízení. Třetí kapitola se zabývá nastavením všech nezbytných modulů jako například PWM modulaci, vyčítání analogových veličin atd. Čtvrtá kapitola popisuje obsluhu a nastavení komunikace RS-485 a CAN. Pátá kapitola se věnuje oživování řídicí jednotky a odladění vybraného řízení. V poslední kapitole se práce věnuje FMEA analýze.

Práce je zakončena závěrem, ve kterém je souhrn všech dosažených výsledků.

# 1 Hardware řídicí jednotky

Řídicí jednotka byla navržena pro elektrickou motokáru, kterou má katedra výkonové elektroniky a strojů k dispozici pro výukové a propagační účely. Tato jednotka řídí synchronní motor s permanentními magnety (PMSM), komunikaci balancery připojenými na baterie a reakci na jejich chybové stavy, ovládání nabíječky, pedály a osvětlení motokáry viz Obr. 1. [1]



Obr. 1: Blokové schéma řídicí jednotky [1]

## 1.1 Mikrokontrolér TMS320F28335

Jednotka byla navržena s mikroprocesorem TMS320F28335 z řady Delfino C2000 od firmy Texas Instruments, který má požadované periférie pro ovládání veškerých zařízení na motokáře a dokáže řídit PMSM motor [1]. Procesor má periférie, které se využívají ve výkonové elektronice, mezi které patří:

- A/D převodníky – analýza veškerých naměřených veličin, vstupní rozsah 0 – 3 V
- ePWM – klíčová periférie pro řízení výkonových elektronických systémů
- CAN – Robustní sériové komunikační rozhraní
- a další užitečné periférie jako eCAP, eQEP, SPI

## 1.2 Senzory

Řídicí jednotka je osazena mnoha senzory, které jsou zavedeny jako zpětné vazby do řídicích algoritmů.

### 1.2.1 Proudové senzory

Na jednotce jsou osazena čidla HTFS-400P [2], která měří proudy tekoucí do motoru a proud z baterií. Ten funguje na principu Hallova jevu a měříme jím obě polarity proudu. Jelikož výstupem senzoru je napětí v rozsahu 0 až 5 V analogového signálu, což odpovídá proudům -400A až 400Arms, tak musí být upraven na měřicí rozsah A/D převodníku v MCU. Měření proudů na všech fázích motoru je spolu s měřením polohy velmi důležité pro vybrané řízení motoru (viz 2. kapitola). [1]

### 1.2.2 Napětíový senzor

Pro měření napětí v meziobvodu je použit senzor HCPL-7520 [3], a to hlavně proto, že má měřicí vstupy galvanicky oddělené od baterie. Maximální napětí na vstupu, které senzor dokáže změřit, je 200mV. Rozsah čidla je omezen od středu výš. [1]

### 1.2.3 Teplotní senzory

Teplota střídače přimontovaného k chladiči je měřena dvěma NTC senzory. Při zvýšení teploty se na NTC článku sníží velikost odporu. Tato změna vyvolá změnu napětí měřenou na A/D převodníku MCU.

Pro měření motoru byly zvoleny senzory SMT172 [4], které mají na výstupu PWM signál. PWM signály budeme měřit na GPIO pinech, které jsou součástí eCAP periférie MCU. [1]

### 1.2.4 Senzor polohy rotoru

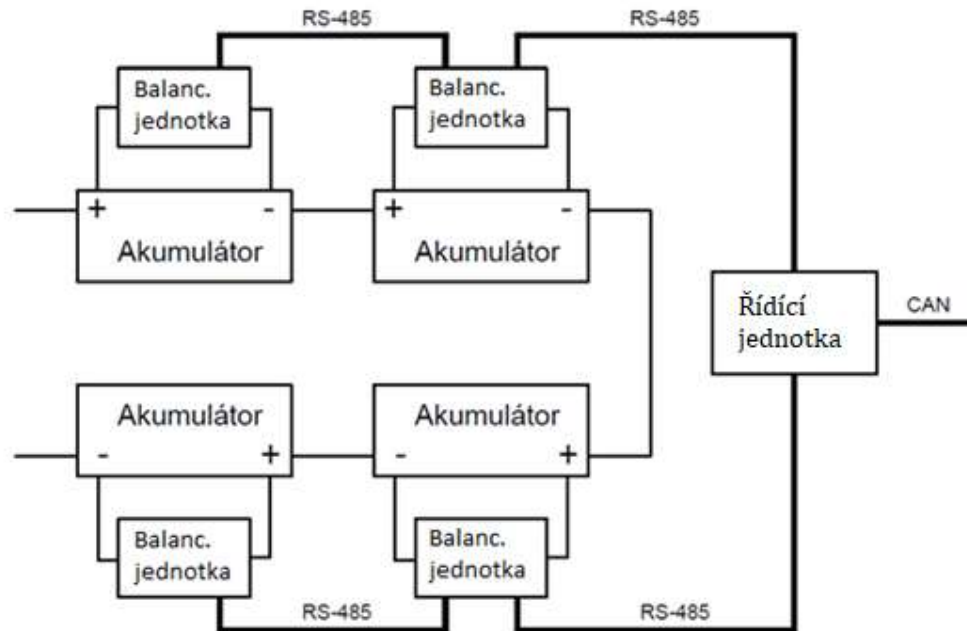
Vektorové řízení, které bude následně popsáno v další kapitole, potřebuje pro ovládání PMSM znát přesnou polohu rotoru. Poloha je použita při přepočtech mezi souřadnicovými systémy. Výstupem senzoru AS5040[5] je absolutní hodnota natočení rotoru a pro další výpočty také IRC.

Toto čidlo obsahuje několik různých režimů. V navrženém řídicím systému bude použit režim s PWM výstupem, který nese informaci o úhlu natočení rotoru. Tato hodnota je

vyhodnocena eCAP periférií v MCU s periodou 1 kHz. Senzor měří úhel natočení rotoru s rozlišením  $0,35^\circ$  a dokáže měřit rychlost až do 30000 otáček za minutu. [1]

### 1.3 Battery managment systém

V elektrické motokáře budou použity balancovací jednotky, které byly navrženy pro snímání důležitých veličin a hlídání velikosti napětí na LiFeYPO4 akumulátorech. Balancovací jednotky hlídají hladinu napětí a balancují jeho velikost při nabíjení. Dále také měří teplotu akumulátoru a chladiče balančního tranzistoru. Na každém akumulátoru je umístěna jedna balancovací jednotka, která měří napětí a teplotu (viz Obr. 2). Díky informaci o napětí a teplotě může být dále zabezpečen nedestruktivní provoz akumulátorů. [6]

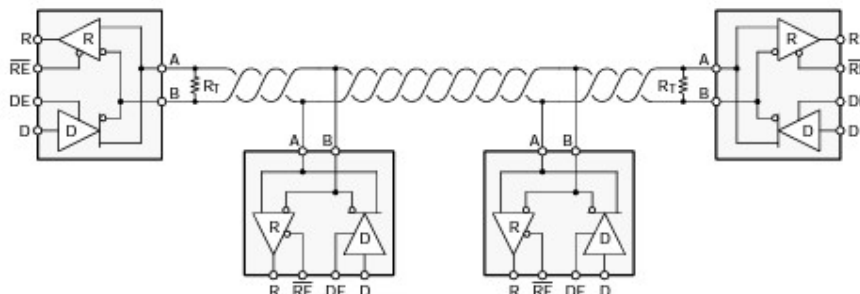


Obr. 2: Blokové schéma propojení jednotek [6]

#### 1.3.1 Komunikační protokol RS-485

Zařízení mezi sebou komunikují po RS-485 standardu asynchronní sériové komunikace s modulační rychlostí 9600 Bd. Pro motokáru je použit budič SN65HVD75[7] od firmy Texas Instruments. Tento budič kombinuje diferenciální vysílač a diferenciální přijímač, které jsou interně propojeny, a tak vytvoří sběrnici vhodnou pro half-duplex komunikaci viz Obr. 3. Half-duplex znamená, že v jednu chvíli dokáže budič data pouze přijímat nebo

odesílat. To řídíme pomocí RE a DE pinů. Vzhledem k tomu, že RE je negované, tak oba piny se můžou připojit na stejný GPIO pin. Samotný RS-485 standart neřeší v komunikačním protokolu kolize vysílaných zařízení. Výhodou diferenciálního vysílání a přijímání je v odolnosti proti rušení. [7]



Obr. 3: Vnitřní zapojení SN65HVD75 budiče [7]

### 1.3.2 Obsah odesílaného paketu

Dotazovací zpráva na BMS jednotku je složena ze tří bytů, viz Tab. 1. První byte obsahuje adresu dotazované měřicí jednotky, druhý byte (viz Tab. 2) v sobě nese informaci s bitem Balancing, který informuje řídicí jednotku, pokud došlo ke snížení nabíjecího proudu při balancování. Na konci každého paketu je zařazen CRC výpočet pro detekci chyb. Byl zvolen polynom generující Dallas/Maxim 8bit CRC viz rovnice (1). [6]

$$g(x) = x^8 + x^5 + x^4 + 1 \quad (1) [6]$$

Adresa	STATUS_BMS	CRC
--------	------------	-----

Tab. 1: Struktura odesílaného paketu [6]

Byte	Bit	Označení	Význam
STATUS_BMS	0	Balancing	Bitová informace o snížení nabíjecího proudu
STATUS_BMS	1	Charge	0: režim jízda, 1: režim nabíjení
STATUS_BMS	2:7	-	reserved

Tab. 2: Význam proměnných v odesílaném paketu [6]

### 1.3.3 Obsah přijímaného paketu

Přijímaný paket je složen z osmi bytů, viz Tab. 3. První je opět adresovací byt. Další 4 byty nesou informaci o napětí a teplotě baterie. V bytu STATUS\_BAT je signalizace požadavku na snížení nabíjecího proudu a zároveň indikuje činnost balančního obvodu.

Adresa	U_bat_H	U_bat_L	TEMP_bat_H	TEMP_bat_L	STATUS_BAT	ERR_BAT	CRC
--------	---------	---------	------------	------------	------------	---------	-----

Tab. 3: Struktura přijímaného paketu [6]

Předposlední byte, označený jako ERR\_BAT, v sobě nese všechny vyhodnocené chyby, jako je podpětí nebo přepětí na baterii, přehřátí baterie či přehřátí balančního MOSFETu, což by mělo jednotku chránit při balancování atd., viz Tab. 4.

Byte	Bit	Označení	Význam
U_bat_L	0:7	-	Napětí baterie – low byte
U_bat_H	0:7	-	Napětí baterie – high byte
TEMP_bat_L	0:7	-	Teplota baterie – low byte
TEMP_bat_H	0:7	-	Teplota baterie – high byte
STATUS_BAT	0	BalancingRequest	1: požadavek na snížení nabíjecího proudu
STATUS_BAT	1:7	-	reserved
ERR_BAT	0	Overvoltage	Napětí větší než 16,5 V
ERR_BAT	1	Undervoltage	Napětí nižší než 10,5 V
ERR_BAT	2	OverTemp	Přehřátí baterie $T > 40$ °C
ERR_BAT	3	OverTempBal	Přehřátí balančního tranzistoru
ERR_BAT	4	BalCurrentErr	Neočekávaný proud balančním tranzistorem
ERR_BAT	5	BalanceErr	Při balancování neklesá U_bat
ERR_BAT	6	TempErr	Chyba měření teploty baterie
ERR_BAT	7	TempBalErr	Chyba měření teploty balančního tranzistoru

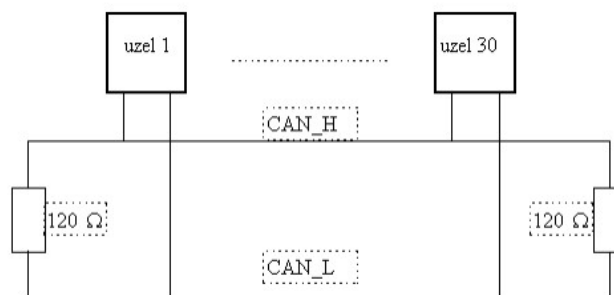
Tab. 4: Bitové rozložení přijímaného paketu [6]

## 1.4 Displej

Dále bude motokára obsahovat navrženou zobrazovací jednotku. Základem je mikrokontrolér STM32F746IGT od firmy STMicroelectronics s frekvencí jádra 216 MHz, s vnitřní flash pamětí 1 MB. Má RAM paměť o velikosti 320 kB, což je nedostatečné pro uchování stránky displeje, proto je k mikrokontroléru přidána externí 64 Mb SDRAM paměť. Mikrokontrolér používá FMC (Flexible memory controller) periférii k zajištění maximální přístupové rychlosti k paměti. Jednotka dále obsahuje sedmipalcový displej, externí EEPROM, GPS přijímač. Zobrazovací jednotka komunikuje s řídicí jednotkou skrze CAN periférii. [8]

### 1.4.1 CAN sběrnice

Controller Area Network je sériová komunikace s vlastním protokolem, která v reálném čase dosahuje velké efektivity a má dobré zabezpečení celé sběrnice.



Obr. 4: Blokové schéma propojení jednotek [9]

CAN zprávy nevyužívají přímé adresování a všechny uzly jsou připojené na jedné fyzické komunikační sběrnici viz Obr. 4. Obsah zprávy tedy může být sdílen s více uzly. Zde by mohlo dojít ke kolizi 2 nebo více uzlů, které budou chtít vysílat zprávu ve stejný čas. To je zajištěno identifikátorem na začátku zprávy, kde dochází k arbitráži zpráv, a tím určuje i pořadí vysílání. [9]

Pro komunikaci s displejem byla použita základní struktura CAN zprávy viz Tab. 5. V prvních dvou bytech je obsaženo ID objektu, kterému je zpráva určena. V dalších dvou bytech je identifikační číslo CAN zprávy. Další dva byty obsahují informace o velikosti odesílaných dat. Pro data je rezervováno 16 bytů.

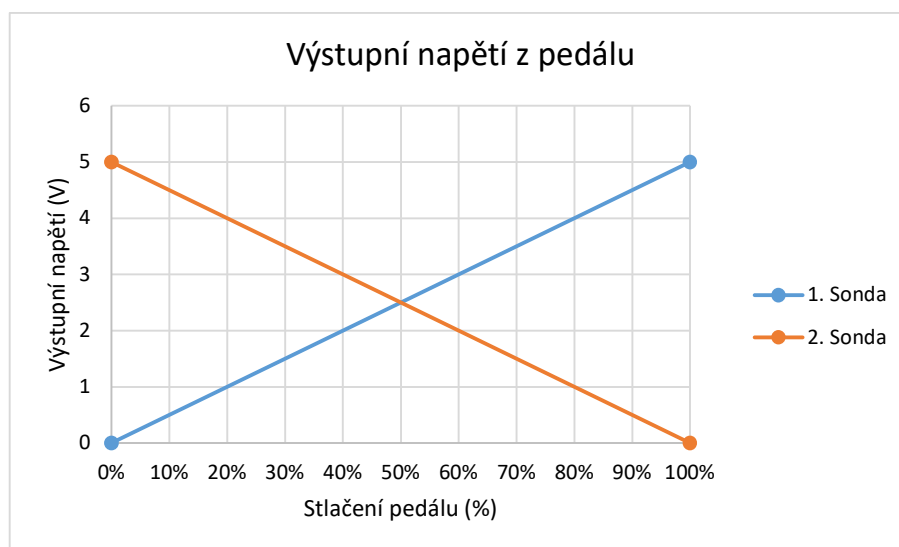
Identifikátor objektu	Identifikátor zprávy	Délka dat	Data[0:7]
-----------------------	----------------------	-----------	-----------

Tab. 5: Struktura CAN zprávy pro displej

### 1.5 Pedály

K vyhodnocení polohy pedálů se využívá Hallova jevu. V pedálu jsou zabudované dvě lineární Hallovy sondy A1302[10]. Výstup z čidla má rozsah od 0 až 5 V. Jedna sonda je vůči té druhé invertovaná. To znamená, že při nestlačeném pedálu bude z jedné sondy na výstupu 0 V a na druhé sondě plné napětí 5 V. Druhá sonda je přidána z důvodů ověření korektnosti prvního výstupního napětí, viz graf 1.

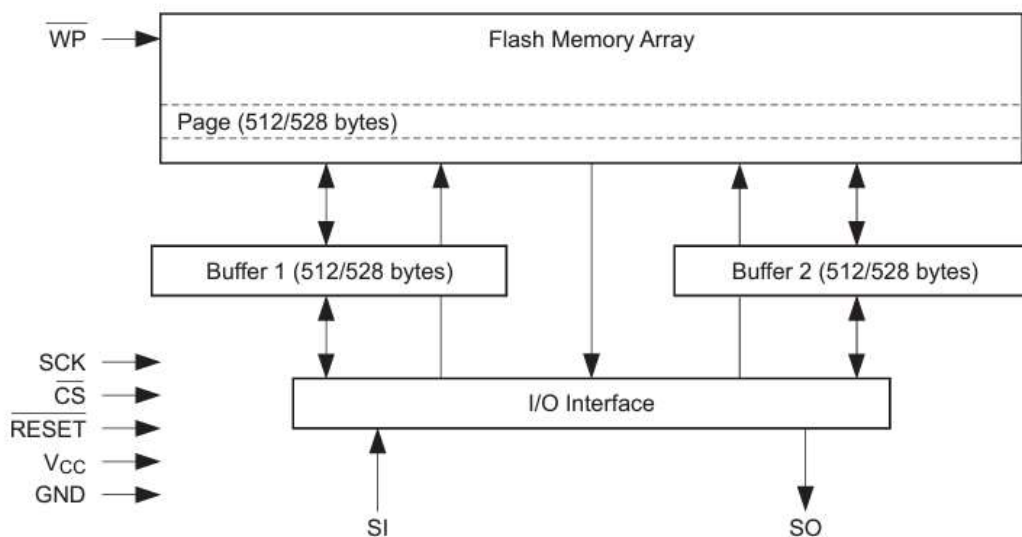




Graf 1: Výstupní napětí z pedálu

## 1.6 Externí Flash paměť

AT45DB321E [11] je flash paměť se sekvenčním přístupem. Jeho 34 603 008 bitů paměti je uspořádáno jako 8 192 stránek po 512 nebo 528 bytech. Kromě hlavní paměti obsahuje AT45DB321E také dvě vyrovnávací paměti SRAM po 512/528 bytech viz Obr. 5. Tyto vyrovnávací paměti umožňují přijímat data v době, kdy je stránka v hlavní paměti přeprogramována. Řídící jednotka komunikuje s flash pamětí skrze SPI rozhraní. [11]



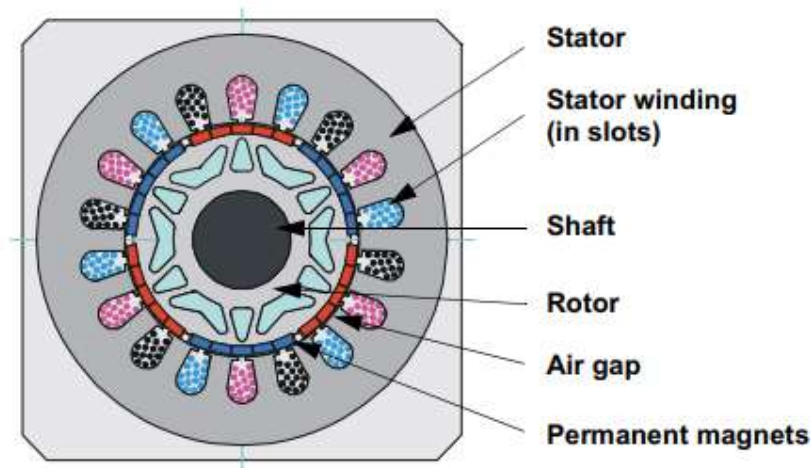
Obr. 5: Blokové schéma flash paměti [11]

## 2 Výběr řídicí metody pro PMSM

V této kapitole jsou popsány tři druhy řízení, z nichž bude zvolen jeden typ řízení a u něj budou následně popsány všechny jeho části.

### 2.1 Synchronní stroj s permanentními magnety

Synchronní stroje se běžně skládají z třífázového statorového vinutí a elektricky buzeným polem na rotoru, kterým protéká stejnosměrný proud. Toto rotorové vinutí lze nahradit permanentními magnety. Jejich použití má mnoho výhod, jako například eliminace kartáčů, kluzných kroužků a ztrát v rotorovém vinutí. Synchronní stroje se vždy točí synchronní rychlostí, která závisí na frekvenci přivedené do statorového vinutí a počtu jejich pólů. Průřez synchronního motoru s permanentními magnety je vidět na Obr. 6. [12]



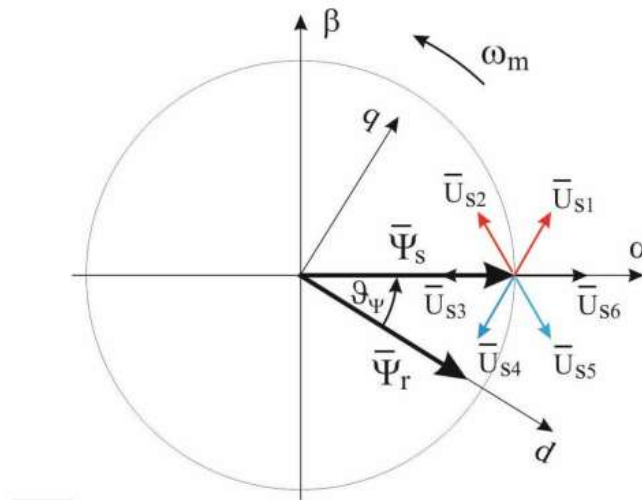
Obr. 6: Průřez synchronního motoru s permanentními magnety [13]

### 2.2 Skalární řízení

Skalární řízení je založeno na řízení konstantního poměru napájecího napětí a napájecí frekvencí. Tím řídíme magnetický tok, který udržujeme konstantní a blízko jeho jmenovité hodnotě. Skalární řízení neumožňuje řízení okamžité hodnoty momentu, což má za následek nedostatečnou dynamiku regulace. Problém nastává při přechodových dějích, protože řízení vychází z rovnic pro ustálený stav asynchronního stroje. Kvůli nedostatečné dynamice regulace není tato metoda pro řízení pohonu v motokáře vhodná. [14]

### 2.3 Přímé řízení momentu DTC

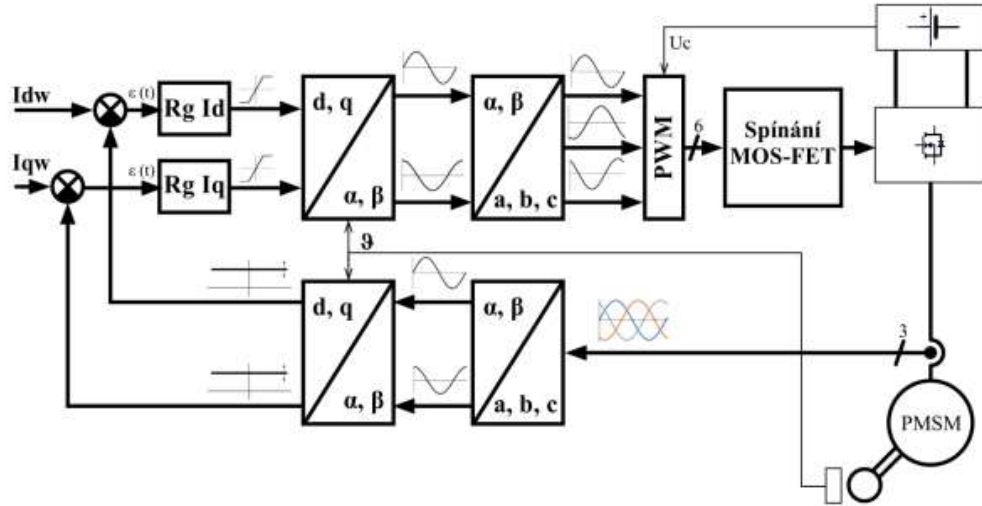
Tato metoda byla původně také navržena pro řízení asynchronních strojů, ale tento koncept lze aplikovat i na synchronní pohony. Vektorové řízení i DTC využívá vektor toku statoru a odhad točivého momentu. DTC využívá principu dvouhodnotové regulace momentu, nevyužívá se zde PWM modulátor, ale přímo se spínají fyzické kombinace střídače. Trajektorie koncového bodu vektoru statorového toku je řízena pomocí napěťových vektorů (viz Obr. 7). Obvykle nelze přímo měřit magnetický tok statoru, a proto je nutné použít matematický model pro výpočet velikosti a polohy vektoru statoru. [14]



Obr. 7: Řízení trajektorie statorového toku [14]

### 2.4 Vektorové řízení

Vektorové řízení je elegantní metoda řízení PMSM, kde se k řízení prostorových vektorů magnetického toku, proudu a napětí používá teorie orientovaná na pole. Je možné nastavit souřadnicový systém tak, aby se vektory rozložily na část generující magnetické pole a část generující točivý moment. Struktura regulátoru motoru (viz Obr. 8) je pak téměř stejná jako u samostatně buzeného stejnosměrného motoru, což zjednodušuje řízení PMSM. Při této metodě je třeba rozdělit části statorového proudu, aby bylo možné odděleně řídit magnetický tok a točivý moment samostatně. K dosažení tohoto cíle je nutné nastavit rotační souřadnicový systém spojený s magnetickým polem rotoru; tento systém se obecně nazývá d-q souřadnicový systém. [13]



Obr. 8: Základní regulační schéma vektorového řízení [1]

DTC oproti vektorovému řízení využívá matematický model a hysterezní komparátory. K tomu je zapotřebí vysoká vzorkovací frekvence. Také u DTC při přechodu mezi sektory dochází ke zkreslení proudu a točivého momentu. Oproti DTC jsou ve vektorovém řízení regulační smyčky proudů v osách d-q, transformace souřadnic a modulátor napětí. Jelikož je díky transformacím možné odděleně řídit magnetický tok a moment, tak je pro řízení PMSM motoru vhodnější vektorové řízení. [14]

## 2.5 Popis transformací vektorového řízení

Model použitý pro návrh vektorového řízení lze pochopit pomocí teorie prostorových vektorů viz rovnice (2). Tyto vektory v komplexním tvaru vyjadřují veličiny třífázového motoru. Takový model je platný pro okamžitou změnu napětí a proudu harmonického signálu a vhodně popisuje výkon stroje v ustáleném i přechodném stavu.

$$\bar{i}_s = k(i_{sa} + a i_{sb} + a^2 i_{sc}) \quad (2) [13]$$

kde  $i_s$  je výsledný vektor z  $i_{sa}$ ,  $i_{sb}$ ,  $i_{sc}$ ,  $k$  je transformační konstanta  $2/3$  a  $a$  je prostorový operátor, který je  $e^{j2\pi/3}$ . [13]

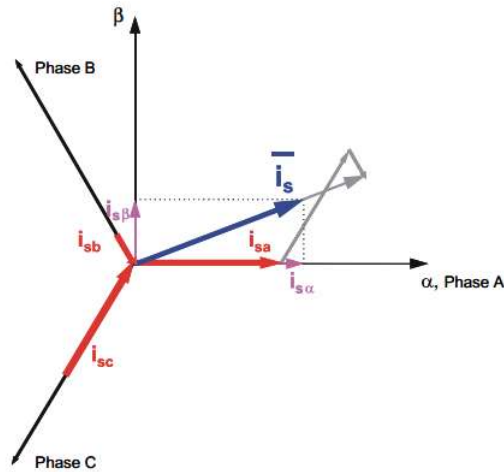
## 2.6 Clarkové transformace

Komplexní prostorové vektory mohou být popsány pouze pomocí dvou ortogonálních os, viz následující rovnice (3). Motor pak lze považovat za dvoufázový stroj a tím se sníží počet

rovnice a zjednoduší se návrh řízení. Pro takovou unifikaci systému řízení se používá Clarkové transformace. [13]

$$\bar{i}_s = i_{s\alpha} + j i_{s\beta} \quad (3) [13]$$

kde  $i_{s\alpha}$  je reálná část prostorového vektoru a je rovna okamžité hodnotě stejnosměrné složky statorového proudu a  $i_{s\beta}$  je imaginární část a je rovna kvadrurní složce statorového proudu viz Obr. 9.



Obr. 9: Prostorový vektor statorového proudu [2]

Vztahy pro Clarkovu transformaci:

$$i_\alpha = k(i_a - \frac{1}{2}i_b - \frac{1}{2}i_c), \quad (4) [13]$$

$$i_\beta = k \frac{\sqrt{3}}{2} (i_b - i_c). \quad (5) [13]$$

Vztahy pro zpětnou Clarkovu transformaci:

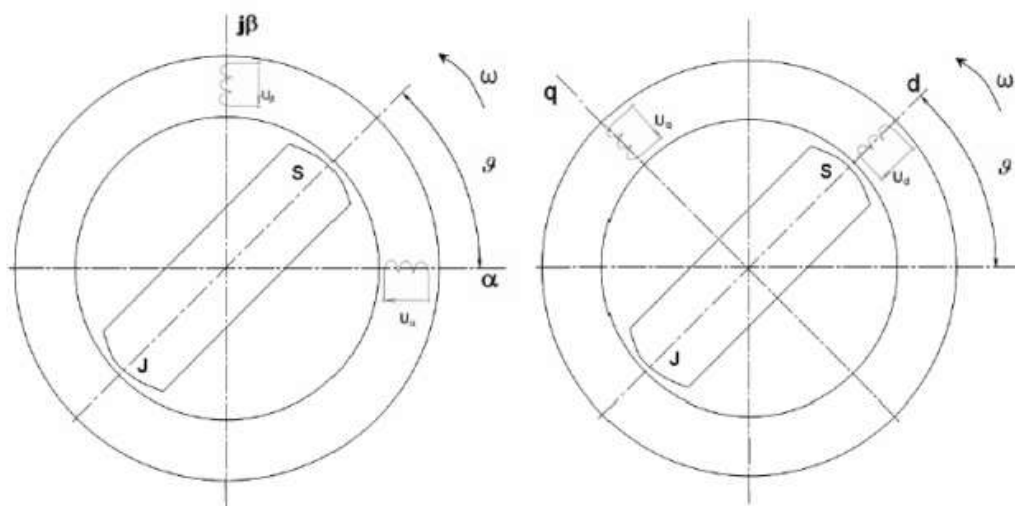
$$i_a = i_\alpha, \quad (6) [13]$$

$$i_b = -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta, \quad (7) [13]$$

$$i_c = -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta. \quad (8) [13]$$

## 2.7 Parkova transformace

Parkova transformace slouží k převedení stacionárního souřadnicového systému  $\alpha, \beta$  na rotující souřadnicový systém  $d, q$ . Jedná se o transformaci statorových souřadnic na rotorové, které jsou závislé na úhlu natočení  $\vartheta$  (viz Obr. 10).



Obr. 10: Znárodnění Parkovy transformace [15]

Vztahy pro Parkovu transformaci:

$$i_d = i_\alpha \cos \vartheta + i_\beta \sin \vartheta, \quad (9) [13]$$

$$i_q = -i_\alpha \sin \vartheta + i_\beta \cos \vartheta. \quad (10) [13]$$

Vztahy pro zpětnou Parkovu transformaci:

$$i_\alpha = i_d \cos \vartheta - i_q \sin \vartheta, \quad (11) [13]$$

$$i_\beta = i_d \sin \vartheta + i_q \cos \vartheta. \quad (12) [13]$$

## 2.8 PI regulátor

PI regulátor je jedním ze základních typů regulátorů, který je využíván pro regulaci systémů v průmyslových a automatizačních aplikacích. Je to kombinace dvou základních prvků - proporcionální (P) a integrální (I) regulace. Proporcionální část regulátoru přispívá k reakci systému na okamžitou změnu vstupního signálu. Integrální část regulátoru přispívá k eliminaci dlouhodobých odchylek systému tím, že sleduje jejich historii a akumuluje je v čase. Výstup z PI regulátoru se bude lišit v závislosti na hodnotách P a I konstant. Obecně

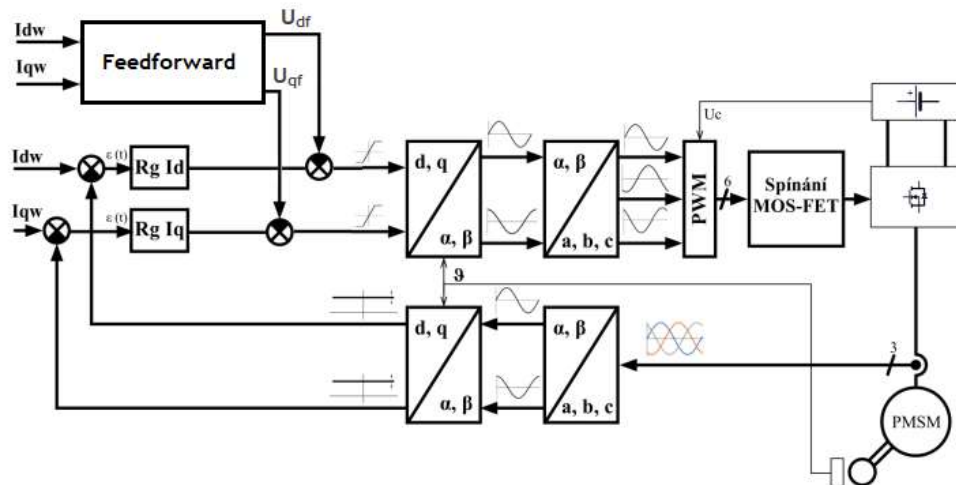
Lze konstatovat, že s rostoucí hodnotou P konstanty se zvyšuje rychlost reakce systému na změny vstupního signálu, ale také se zvyšuje pravděpodobnost oscilací a vlivu rušení. S rostoucí hodnotou I konstanty se zvyšuje schopnost systému eliminovat dlouhodobé odchylky, ale také se zvyšuje doba odezvy systému. [16]

## 2.9 Feedforward

Feedforward využívá vstupní signál a model systému k výpočtu ideálního výstupního signálu, který by měl být dosažen. V kontextu PMSM motoru se feedforward využívá k odstranění kmitání a zlepšení dynamiky motoru. Jeho aplikací snižujeme nároky na kvalitu PI regulátorů. K výpočtu žádaných hodnot statorového napětí ( $U_d$  a  $U_q$ ) využívá základních znalostí o parametrech motoru jako je odpor, indukčnost, magnetický tok a úhlová rychlost. (viz rovnice 13 a 14). Na Obr. 11 je přidán feedforward do regulačního schématu vektorového řízení. [17]

$$\mathbf{u}_q = R_s \mathbf{I}_q + L_s \frac{d\mathbf{i}_q}{dt} + \omega \psi_q \quad (13) [18]$$

$$\mathbf{u}_d = R_s \mathbf{I}_d + L_s \frac{d\mathbf{i}_d}{dt} - \omega \psi_d \quad (14) [18]$$



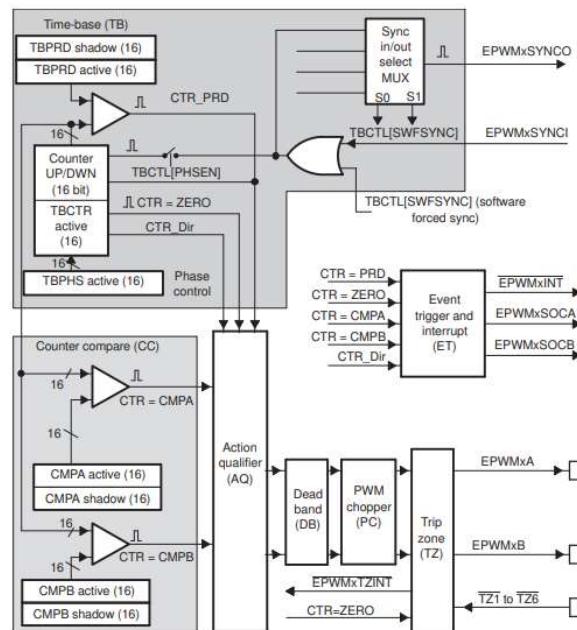
Obr. 11: Regulační schéma vektorového řízení s feedforwardem

### 3 Konfigurace periférií použitých pro řízení

V této kapitole bude popsána konfigurace jednotlivých periférií procesoru, které jsou nezbytné pro vybraný typ řízení jako je ePWM, ADC, eCAP, eQEP

#### 3.1 Generování PWM signálu

ePWM (Enhanced Pulse Width Modulator) je periférie v mikrokontroléru TMS320F28335, která umožňuje generovat pulzně šířkově modulované signály s různými konfiguracemi. Takto modulovaný signál se používá k ovládní výkonových spínačů, které mění napětí a proud dodávaný do motoru. Blokové schéma jednoho ePWM modulu lze vidět na Obr. 12.



Obr. 12: Blokové schéma ePWM modulu [19]

Základními bloky ePWM jsou:

- **Time-base (TB)** – nastavení časové základny
- **Counter compare (CC)** – čítač porovnávaný s časovou základnou
- **Action-qualifier (AQ)** – nastavení reakcí na CC modul
- **Dead-Band (DB)** – *p*Přidání mrtvého časupásmo
- **PWM-Chopper (PC)** – modulace generovaného PWM signálu
- **Trip-Zone (TZ)** – *v*řypnutí PWM výstupu na *chybu*
- **Event-trigger (ET)** – *g*Generování přerušení nebo SOC



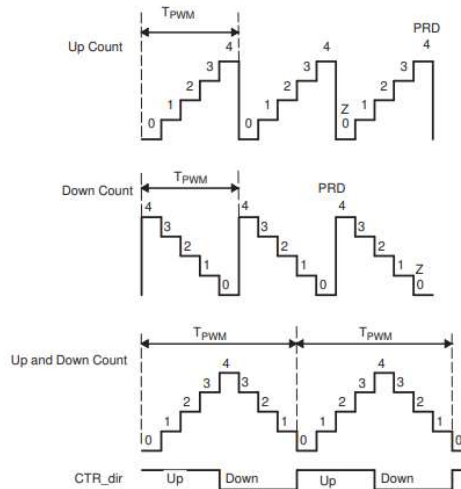
### 3.1.1 Konfigurace ePWM modulu

Řídicí jednotka pracuje se třemi ePWM moduly, které jsou inicializované se stejnou konfigurací. Každý z nich je použit pro řízení jedné fáze motoru.

Na začátku inicializace ePWM modulů je třeba vypnout synchronizaci hodin a na konci inicializace je opět zapnout. Tím docílíme, že všechny povolené hodiny modulů ePWM jsou spuštěny s první náběžnou hranou TBCLK, ale to pouze za předpokladu, že nastavení předděličky musí být pro všechny moduly zcela identické. [19]

- **Konfigurace Time-Base submodulu**

Nejprve je třeba si určit jeden ze tří režimů čítání, protože zde se liší výpočet periody signálu, viz Obr. 13. Nastavení provedeme v 16-ti bitovém registru TBCTL.



Obr. 13: Režimy čítání ePWM modulu [10]

Byl zvolen třetí režim čítání nahoru a dolů. Následující rovnice (15) a (16) popisují výpočet 20 kHz periody signálu pro zvolený režim čítání.

$$T_{PWM} = \frac{1}{F_{PWM}} = \frac{1}{20 * 10^3} = 0,00005 s = 50 \mu s \quad (15) [19]$$

$$TBCLK = \frac{SysClkOut}{(HSPCLKDIV * CLKDIV)} = \frac{150 * 10^6}{(2 * 1)} = 75 MHz \quad (16) [19]$$

Nastavením HSPCLKDIV a CLKDIV bitů vytvoříme předděličku systémových hodin procesoru pro ePWM modul s periodou 75 MHz. S touto periodou pracuje čítač TB submodulu. Výpočet jedné periody PWM je popsán v následujících rovnicích (17) a (18).

$$T_{PWM} = 2 * TBPRD * T_{TBCLK} = 2 * TBPRD * \frac{1}{TBCLK} \quad (17) [19]$$

$$TBPRD = \frac{T_{PWM} * TBCLK}{2} = \frac{50 * 10^{-6} * 75 * 10^6}{2} = 1875 \quad (18) [19]$$

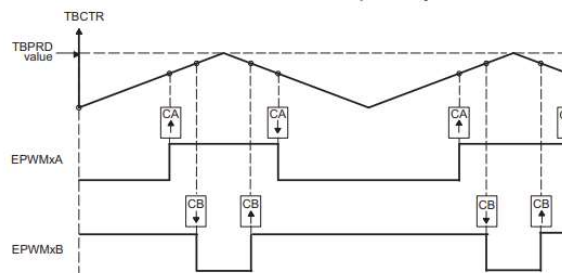
TBPRD je 16-ti bitový registr, ve kterém je uložena hodnota, která se porovnává s hodnotou uloženou v čítači. [19]

- **Konfigurace Counter-Compare submodulu**

Skládá se z dvou registrů (CMPA a CMPB). V tomto případě bude stačit jeden z těchto registrů. Hodnota zadávána do CMPA nesmí být větší než vypočítaná hodnota TBPRD, jelikož je tato hodnota porovnávána s hodnotou čítače z TB submodulu. Při rovnosti hodnoty čítače a CMPA se generují signály pro další submodul. Do CMPA registru se budou při každé periodě zadávat vypočtené hodnoty průběhů napětí ve vybraném řízení. Z toho důvodu se v CMPCTL registru nastavují stínové registry. Ty zapisují nové hodnoty do CMPA registru při události nastavené v LOADAMODE bitech. Tímto zabezpečíme přepis hodnot až na začátku nové periody. [19]

- **Konfigurace Action-Qualifier submodulu**

Tento submodul slouží k řízení akce, která má být provedena, když přijde vygenerovaný signál z CC submodulu. Pokud čítač dosáhne hodnoty CMPA a čítač inkrementuje, tak na výstupu EPWMxA bude 3,3 V a výstup z EPWMxB bude inverzní. Když bude čítač dekrementovat, tak na výstupu EPWMxA bude 0 V a na výstupu EPWMxB bude 3,3 V viz Obr. 14. [19]



Obr. 14: Výstup z Action-Qualifier submodulu [10]

- **Konfigurace Dead-Band submodulu**

Slouží ke zpoždění přepnutí výstupů a pomáhá tak zabránit poškození při přepínání fází. DB submodul má jeden řídicí registr DBCTL. Výstupní mód nastavíme tak, aby

při náběžné hraně přidá zpoždění na EPWMxA a při sestupné hraně přidá zpoždění na EPWMxB. V bitech POLSEL dáváme informaci DB submodulu, že EPWMxB výstup je invertovaný. Obě zpoždění budou nastavena na 1  $\mu$ s. Následující rovnicí (19) dopočítáme hodnotu pro 10-ti bitové registry DBFED (Falling edge delay) a DBRED (Rising edge delay). [19]

$$DBxED = xED * TBCLK = 1 * 10^{-6} * 75 * 10^6 = 75 \quad (19) [19]$$

- ***Konfigurace Trip-zone submodulu***

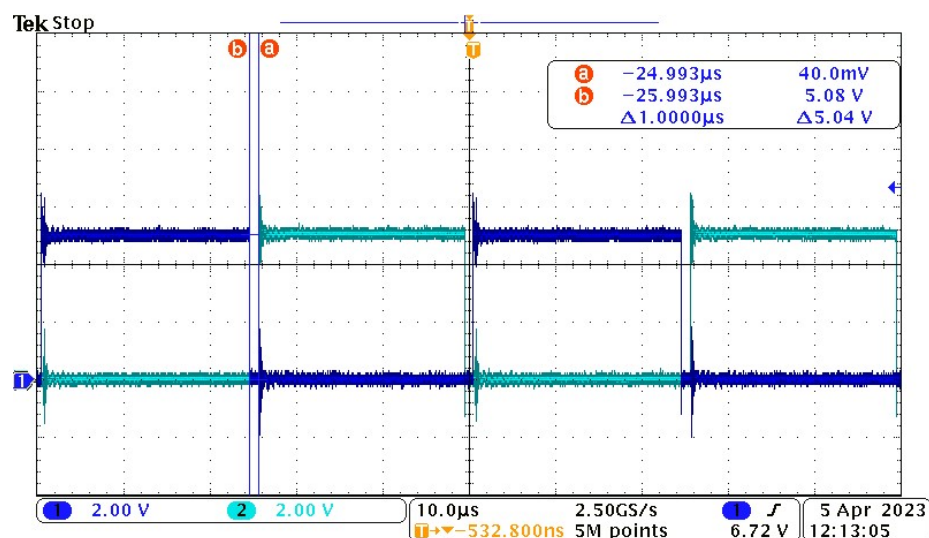
Tento submodul je nakonfigurován tak, aby při zapsání do registru TZFRC byl výstup z ePWM nastaven na 0 V, dokud se nezapíše do registru TZCLR. Registr TZSEL je nastaven na one-shot akci a nastavením TZCTL registru se určí, jak bude výstup reagovat. [19]

- ***Konfigurace Event-Trigger submodulu***

Řídí události generované TB a CC submoduly, aby vytvořil přerušení pro CPU anebo spustil převod signálu pro ADC, když nastane vybraná událost. Pro vybrané řízení je submodul nastaven tak, když se čítač dostane na hodnotu 0, vyvolá přerušení a spustí konverzi ADC. Tento submodul nastavujeme pouze pro první ePWM modul. [19]

### 3.1.2 Experimentální ověření

Po nahrání a spuštění řídicího algoritmu do jednotky je osciloskopem změřen průběh napětí, které je na výstupu PWM. Kanál jedna měří výstup EPWM1A a kanál dva EPWM1B. Na Obr. 15 je změřený průběh výstupního napětí z 1. PWM modulu, kde lze vidět, že jsou proti sobě vstupy inverzní, naměřené mrtvé pásmo o délce 1  $\mu$ s a dále jde vidět i perioda, která je 50  $\mu$ s.

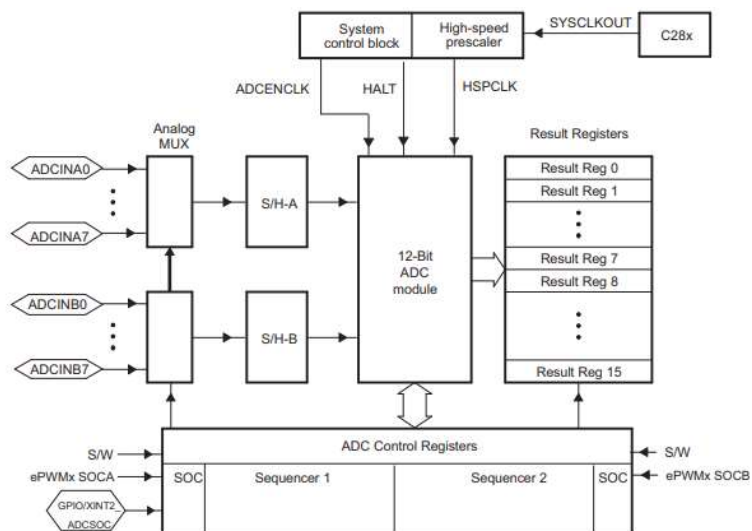


Obr. 15: Průběh napětí výstupu z PWM; Napěťový výstup z EPWM1A CH1 (modrá) 2V/div a Napěťový výstup z EPWM1B, CH2 (Tyrkisová) – 2V/div (50A/div)

### 3.2 Vyčítání analogových hodnot

Pro vyčítání analogových hodnot jsou v MCU interní AD převodníky. ADC modul má rozsah vstupního napětí od 0 do 3 V. Modul ADC má 16 kanálů, které lze konfigurovat jako dva nezávislé 8-kanálové moduly. Tyto dva nezávislé 8-kanálové moduly lze kaskádovat, aby vytvořily modul se 16 kanály. I přesto, že má modul více vstupních kanálů a dva sekvencery, v modulu ADC je pouze jeden 12-ti bitový převodník, který může převést v jednom okamžiku pouze jednu hodnotu, viz

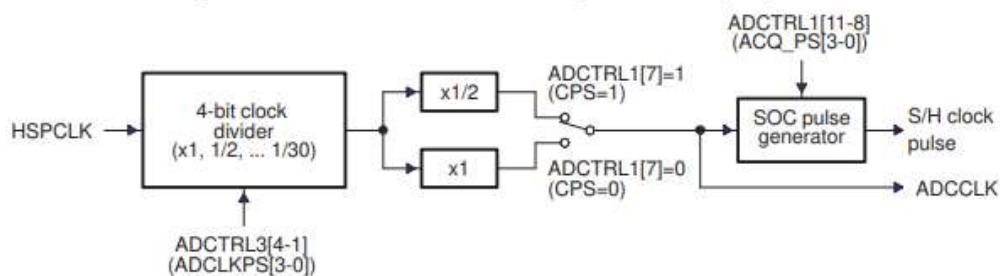
Obr. 16. Po dokončení převodu se hodnota vybraného kanálu uloží v odpovídajícím 16-ti bitovém registru. [19]



Obr. 16: Blokové schéma ADC modulu [19]

### 3.2.1 Konfigurace ADC modulu

Jako první je nutné nastavit časování periferie a dobu vzorkování signálu. Při nastavení vzorkování AD převodníku je třeba najít kompromis mezi rychlostí a přesností vzorku. Krátké vzorkování sice bude dřív provedené, ale je možné, že hodnoty nebudou tak přesné. Perioda PWM signálu, která vyvolává přerušení pro převod analogových hodnot, je 50  $\mu$ s. Do této periody se musí vměstnat i celé řízení, tak i měření všech analogových vstupů.



Obr. 17: Schéma časování ADC modulu [19]

Nastavení hodin ADC modulu se provede následující rovnicí (20), která vychází z Obr. 17.

$$ADCCLK = \frac{HSPCLK}{ADCLKPS * (CPS + 1)} = \frac{75 * 10^6}{10 * (0 + 1)} = 7,5 \text{ MHz} \quad (20) [19]$$

ADCCLKPS[3:0] bity v registru ADCTRL3 dělí periferní hodiny HSPCLK. Ty jsou defaultně nastaveny na polovinu systémových hodin, tzn. 75MHz. V této periférii je možné

dále dělit hodiny pomocí bitu CPS. Pomocí bitů ACQ\_PS[3:0] v registru ADCTRL1 je možné prodlužovat periodu vzorkování. V SMODE\_SEL bitu je nastaven sekvenční režim vzorkování. Viz následující rovnice (21), kde je vypočtena doba vzorkování jednoho kanálu.

$$T_{vz} = \frac{1}{ADCCLK} * (ACQ_{PS} + 1) = \frac{1}{7,5 * 10^6} * (4 + 1) = 0,667 \mu s \quad (21) [19]$$

V registru ADCTRL1 je nastaven kaskádní mód, start-stop mód v CONT\_RUN bitu, kdy při dokončení převodu čeká na další SOC od PWM modulu, a teprve poté začne nový převod. Povolení přerušeni a jeho reakci na SOC od PWM modulu nastavíme v registru ADCTRL2.

Nakonec se přiřadí jednotlivým kanálům sekvence, pořadí a tím bude dán jejich registr s 12-ti bitovou hodnotou. Všechny měřené veličiny jsou uvedené v Tab. 6.

Měřené veličiny	Vstupy do CPU	Result registry
Proud U fáze	ADCINA0	AdcRegs.ADCRESULT1
Proud V fáze	ADCINB0	AdcRegs.ADCRESULT2
Proud W fáze	ADCINB1	AdcRegs.ADCRESULT3
Proud z baterie	ADCINA7	AdcRegs.ADCRESULT4
Napětí na bateriích	ADCINA6	AdcRegs.ADCRESULT5
Pedál brzdy 1	ADCINA4	AdcRegs.ADCRESULT6
Pedál brzdy 2	ADCINA2	AdcRegs.ADCRESULT7
Pedál plynu 1	ADCINA5	AdcRegs.ADCRESULT8
Pedál plynu 2	ADCINA3	AdcRegs.ADCRESULT9
Proud z nabíječky	ADCINA1	AdcRegs.ADCRESULT10
Teplota na motoru 1	ADCINB3	AdcRegs.ADCRESULT11
Teplota na motoru 2	ADCINB6	AdcRegs.ADCRESULT12
Teplota na motoru 3	ADCINB7	AdcRegs.ADCRESULT13
Teplota na chladiči řídicí jednotky 1	ADCINB2	AdcRegs.ADCRESULT14
Teplota na chladiči řídicí jednotky 2	ADCINB5	AdcRegs.ADCRESULT15

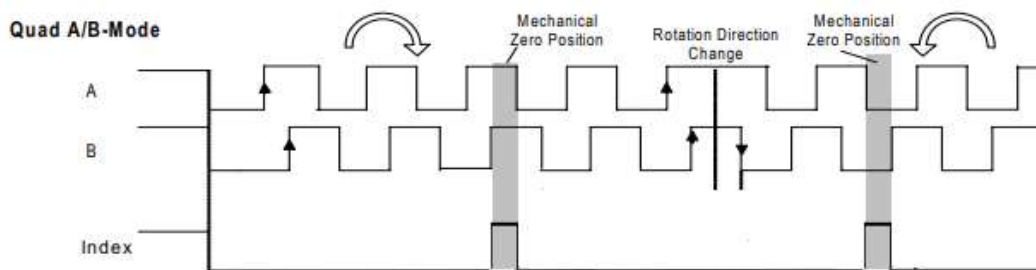
Tab. 6: Tabulka měřených veličin skrze analogové vstupy

### 3.3 Měření otáček a polohy rotoru

V předešlé kapitole 1.2.4 bylo zmíněno, že čidlo polohy generuje PWM signál, který v sobě nese informaci o absolutní poloze rotoru. Mikrokontrolér TMS320F28335 má eCAP periférii. Ta umí detekovat a zpracovávat periodu pulzu vstupního signálu PWM. Pro zpracování šířky pulzu periférie využívá Time-Stamp čítač, který začíná při detekci náběžné nebo doběžné hrany a čítá, dokud nedetekuje další hranu pulzu. Konfigurace eCAP5 modulu pro senzor AS5040 je znázorněna na

Obr. 19 v následující kapitole 3.3.1. [19]

Senzor AS5040 má v sobě také IRC čidlo pro měření rychlosti a má tři výstupy. První dva signály (A a B) jsou vzájemně fázově posunuty o  $90^\circ$ , aby bylo možné určit směr otáčení. Směr je dán tím, který signál generuje impulzy jako první (viz Obr. 18). Tento typ signálu se nazývá kvadrurní signál. Třetí signál generuje pouze jeden impulz za otočku a určuje výchozí polohu. [5]

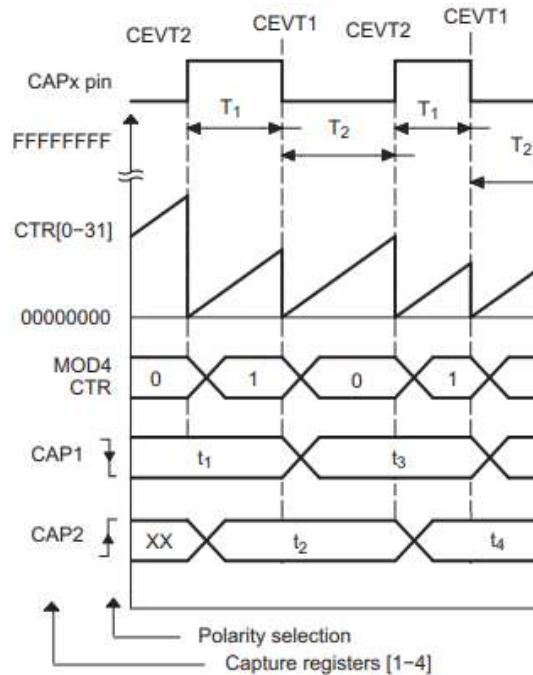


Obr. 18: Výstup z IRC čidla [5]

Modul kvadrurního enkodéru (eQEP) je součástí mikrokontroléru TMS320F28335.

### 3.3.1 Konfigurace eCAP modulu

Pro konfiguraci má eCAP modul dva kontrolní registry ECCTL1 a ECCTL2. CAPLDEN bit řídí načtení hodnot CAP registrů až při události nastavené v ECEINT registru. V bitech CAPxPOL je nastaveno, při jaké hraně dojde k zaznamenání časové hodnoty do CAP registrů. Vzhledem k tomu, že výstupem jsou časové rozdíly mezi hranami, tak je pomocí bitů CTRRSTx nastaven reset čítače po každé zachycené hraně. V druhém kontrolním registru je nastaven operační mód na kontinuální čtení vstupu a synchronizace v SYNCO\_SEL bitech je vypnuta. V registru ECEINT je konfigurace taková, že když dojde k naplnění CAP2 registru, vyvolá se přerušení. [19]



Obr. 19: Měření PWM signálu eCAP modulem [19]

### 3.3.2 Vyčtení absolutní polohy rotoru

eCAP po zachycení hodnoty vyvolá přerušovací rutinu, kde dochází k přepočítání na polohu rotoru a elektrického pole statoru, viz zdrojový kód 1.

```

interrupt void ECAP5_INT_ISR(void)
{
    float DutyOnTime = 0;

    DutyOnTime = ECap5Regs.CAP2;
    Period = DutyOnTime + ECap5Regs.CAP1;

    Angle = (uint16_t)(round((DutyOnTime/Period)*1025)-1); //Calculate duty cycle
    Elec_Angle_uint = ((Angle << 2) - 240) & 0x3FF;
    Elec_Angle = (float)Elec_Angle_uint * 0.35;
#ifdef FILT_POS
    eQEP_position_filter();
#else
    meas_primary.position = -Elec_Angle * 0.01745;
#endif
    var_position.raw_data.float_var = Elec_Angle_uint;

    ECap5Regs.ECCLR.bit.CEVT2 = 1;//clearing the inter. flag and preparing for the next invocation
    ECap5Regs.ECCLR.bit.INT = 1;
    ECap5Regs.ECCTL2.bit.REARM = 1;

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP4;
}

```

Zdrojový kód 1: Přerušovací rutina eCAP modulu

Z registrů CAP2 a CAP1 se uloží zachycené hodnoty, ze kterých se dopočítá střída jednoho cyklu. Výstup čidla je 10ti bitový, takže musí dojít k přepočtu podle následující rovnice. [19]

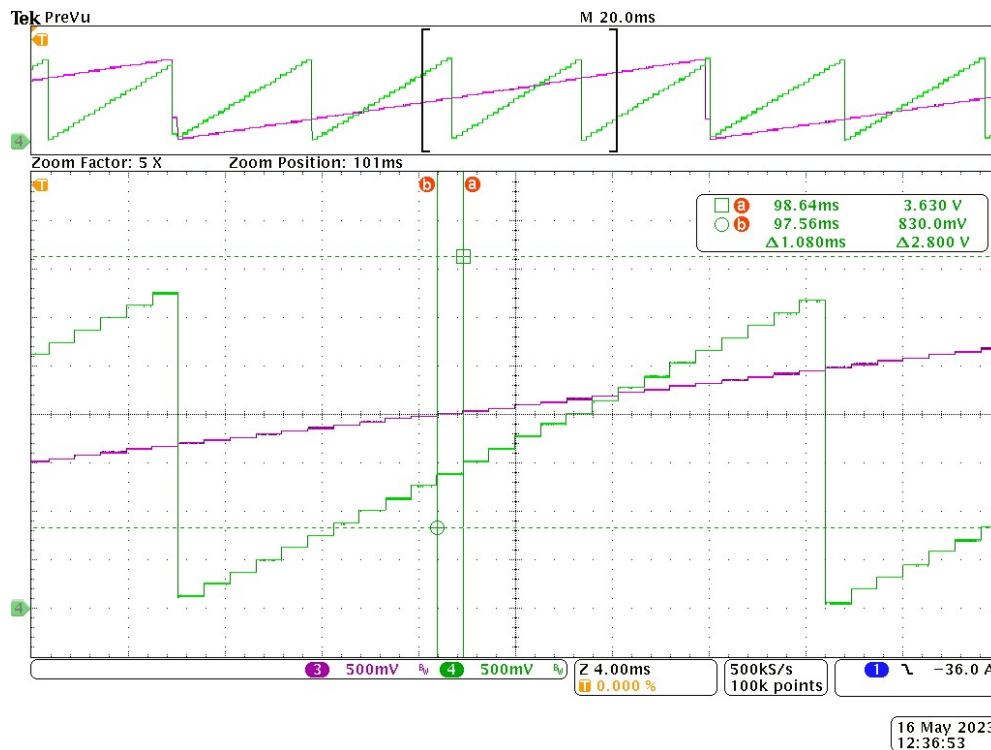


$$position = \frac{DutyOnTime * 1025}{Period} - 1 \quad (22) [5]$$

Pro vektorové řízení je třeba zadávat natočení elektrického pole rotoru. PMSM motor má 4 pólpáry, a tak pro výslednou hodnotu úhlu elektrického pole je potřeba proměnnou Angle vynásobit čtyřmi. V programu se vykoná bitový posun o 2 bity doleva a vymaskuje se spodních 10 bitů. Tyto dva výsledky jsou přenásobeny hodnotou 0,35, což odpovídá změně úhlu o  $0,35^\circ$  při změně jednoho bitu. Ve struktuře meas\_primary bude uloženo natočení elektrického pole statoru v radiánech.

### 3.3.3 Experimentální ověření eCAP konfigurace

Skrze DA převodník se posílá naměřená hodnota z eCAP modulu tak, aby ho bylo možné zachytit osciloskopem na výstupu. Na Obr. 20 je fialový naměřený průběh (kanál 3) polohy rotoru a zelený průběh (kanál 4) polohy elektrického pole s frekvencí otáčení 37 Hz. Je zde vidět, že eCAP pracuje na frekvenci blízké 1 kHz. Pro toto ověření použijeme podmínění překlad bez filtru. Tento filtr je použit a popsán v 5.2



Obr. 20: Měřená absolutní poloha rotoru; úhel natočení rotoru ( $0-2\pi$ ) CH3 (fialová) – 500mV/div a úhel natočení rotoru používaný v transformacích ( $0-2\pi$ ) CH14 (zelená) – 500mV/div při  $f_{el} = 37\text{Hz}$ .

### 3.3.4 Konfigurace eQEP modulu

Periférie eQEP má svůj 32bitový čítač (QUTMR), který je taktován systémovými hodinami a porovnáván s hodnotou nastavenou v QUPRD registru. To udává periodu, s kterou se volá přerušení s vyčítáním rychlosti. V QDECCTL registru je nastaven kvadrurní mód čítání. Registr QEPCTL bude konfigurován tak, že bitem UTE je povolen reset jeho hodnoty na konci jedné periody měření, detekce náběžných hran při čítání. [19]

### 3.3.5 Vyčtení rychlosti

S periodou 100 Hz dochází k vyvolání přerušovací rutiny eQEP, viz zdrojový kód 2. Perioda měření byla zvolena podle následujícího rovnice (23), kde zjistíme, jak velká chyba může nastat při nižší periodě měření úhlové rychlosti  $T_\omega$ .

$$f_m = \frac{1}{4 * IRC * T_\omega} = \frac{1}{4 * 1024 * 0,01} = 0,0976 \quad (23) [20]$$

S ještě nižší periodou vzorkování by tato chyba mohla být větší a pro vyčítání rychlosti je to v této aplikaci dostačující.

```
#define QEP_CONST (0,09765625) // 1/(1024*0.01) - pulses per revolution, 0.01 ms period
interrupt void EQEP_INT_ISR(void)
{
    int32_t qap_read1 = 0;

    qap_read1 = EQep1Regs.QPOSLAT;
    meas_primary.mech_speed = QEP_CONST * qap_read1;
    meas_primary.el_speed = 4 * meas_primary.mech_speed;

    EQep1Regs.QCLR.bit.INT = 1;
    EQep1Regs.QCLR.bit.UTO = 1;
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP5;
}
```

*Zdrojový kód 2: Přerušovací rutina eCAP modulu*

V registru QPOSLAT je zachycena hodnota, kterou je potřeba vynásobit konstantou QEP\_CONST. Ve zdrojovém kódu 2 je v komentáři u QEP\_CONST konstanty uvedeny její výpočet, kde hodnota 0.01 je časový interval jednoho měření v ms a 1024 je počet pulzů na otáčku. Rychlost otáčení statorového pole je čtyřikrát rychlejší kvůli počtu pólů.

## 4 Komunikace

Jak bylo v 1. kapitole zmíněno, motokáru využívá dvou komunikací. RS-485 ke komunikaci s BMS jednotkami a dvě CAN sběrnice. Jednu ke komunikaci s displejem a druhou pro debugging celé řídicí jednotky, potažmo motokáry.

### 4.1 Komunikace s BMS (RS-485)

Modul SCI je dvoudrátový asynchronní sériový port, běžně známý jako UART. Tento modul podporuje digitální komunikaci mezi CPU a dalšími asynchronními periferiemi, které používají standardní formát NRZ (non-return-to-zero). Přijímač i vysílač SCI mají každý svou FIFO paměť s kapacitou 16 bytů. [19]

#### 4.1.1 Konfigurace SCI modulu

Na rozdíl od jiných periferií, zde je třeba nastavit kvalifikaci vstupu jako asynchronní. Na vstup je použit GPIO62 pin a jeho kvalifikaci vstupu nastavíme v GPBQSEL2 registru.

Pro konfiguraci samotného SCI modulu se využívají SCICTL1, SCICTL2 a SCICCR registry. V SCICTL1, kde bitem TXENA povolíme odesílání skrze příslušný pin a bitem RXENA povolíme přijímání zpráv a jejich uložení do přijímacího bufferu.

V registru SCICTL2 se nastaví první dva bity. Toto nastavení nám zajistí informaci o zaplnění přijímacího a odesílacího bufferu. Když bude v odesílacím bufferu připravena zpráva pro přenos, tak v bitu TXRDY bude hodnota 0. Pokud tam bude 1, tak je buffer prázdný a je připraven přijmout další zprávu k odeslání.

Konfigurací SCICCR zajistíme nastavení vlastnosti odesílaných a přijímaných zpráv. Zde se nechá všechno defaultně nastavené kromě prvních tří bitů, které nastavují, kolik bitů se odešle v jedné zprávě. SCI tedy přijímá zprávy o velikosti 8 bitů, s jedním stop bitem a bez parity.

Hodnota v 8-mi bitových registrech SCIHBAUD a SCILBAUD určuje modulační rychlost, která by v tomto případě měla být 9600 Bd. Tuto hodnotu dostaneme následující rovnicí (24).

$$BRR = \frac{LSPCLK}{(BaudRate * 8)} - 1 = \frac{37,5 * 10^6}{(9600 * 8)} - 1 = 487,28 \quad (24) [19]$$

Hodnota se zaokrouhlí na 487. V hexadecimální soustavě je toto číslo reprezentováno jako 1E7. Do SCILBAUD zapíšeme hodnotu E7 a do SCIHBAUD hodnotu 1.

Na závěr se povolí FIFO paměť pro odesílání i pro přijímání a obě FIFO paměti se vyresetují. [19]

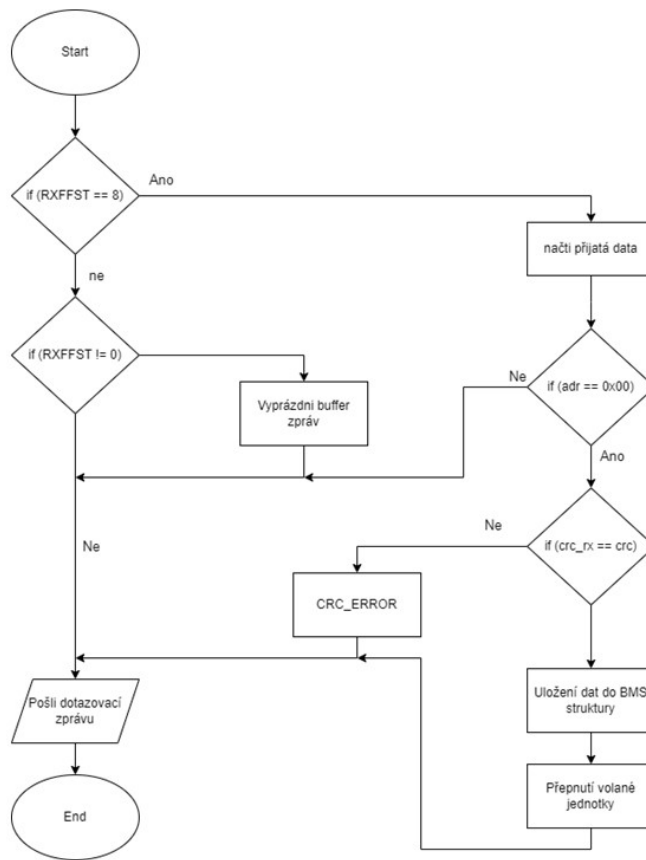
#### 4.1.2 Rutina zpracovávání zpráv

Jak bylo naznačeno v kapitole 1.3, samotný standard RS-485 neřeší v komunikačním protokolu kolize vysílaných zařízení, a tak toto musí být zabezpečené v řídicím softwaru. Řídicí jednotka se opakovaně dotazuje každé měřící jednotky zvlášť a očekává od každé téměř okamžitou odpověď. Zprávu vysílanou na sběrnici řídicí jednotky zachytí každá BMS jednotka. Odpoví pouze jednotka, které jsme se dotázali, zbytek jednotek zprávu ignoruje. Jednotlivé adresy jsou uvedeny v Tab. 7.

Zařízení	Adresa
Řídicí jednotka	0x00
BMS jednotka č. 1	0x11
BMS jednotka č. 2	0x22
BMS jednotka č. 3	0x44
BMS jednotka č. 4	0x88

Tab. 7: Tabulka jednotlivých adres [6]

Protože je to half-duplex komunikace, tak všechny jednotky jsou na začátku v režimu přijímání zpráv a přepnou se do režimu odesílání jen po nezbytně krátkou dobu pro odeslání celého paketu. Řídicí jednotka to řeší periodicky, kdy po 250 ms se postupně doptává všech jednotek a následně přijímá jejich data. Celá rutina je zjednodušeně popsána v následujícím vývojovém diagramu, viz Obr. 21.



Obr. 21: Vývojový diagram obsluhy RS485

Na začátku si řídicí jednotka provede kontrolu, zda nemá data v přijímací FIFO paměti. Pokud nemá žádná data, tak posílá dotazovací zprávu jedné z BMS jednotek a při dalším zavolání funkce by ve FIFO paměti mělo být 8 bytů. Může se stát, že jednotka nepřijme celý paket a pokud přijme jen část paketu, tak všechna data ignoruje a vyprázdní buffer. Pokud má ve FIFO paměti 8 bytů (což je jeden paket jedné zprávy z BMS jednotky), tak načte všechna přijatá data. Dále zkontroluje adresu v prvním bytu, jestli je zpráva řídicí jednotce určena, a následně provede výpočet CRC a porovná s přijatým CRC na konci paketu. Pokud CRC výpočet není totožný s CRC přijatým, jednotka dostane informaci o neshodě a přijatá data zahodí. V opačném případě se data uloží do jedné z BMS struktur pro následné zjednodušení přístupu k jednotlivým proměnným nebo k celé struktuře a dotáže se další jednotka.

Struktura BMS zprávy je totožná s paketem zprávy (viz Tab. 4). V Tab. 8 jsou formáty proměnných, které v sobě nesou informaci o napětí a teplotě baterie a v následujících rovnicích (25) a (26) je příklad přepočtu bitové hodnoty na napětí a teplotu.

Proměnná	Velikost	Typ	Formát	Rozsah	Násobič	Přepočet
UBAT	16 bit	Unsigned	5.11	0; 31,9995	2048	2048 = 1V
TEMP	16 bit	Signed	8.7	-256; 255,992	128	128 = 1 °C

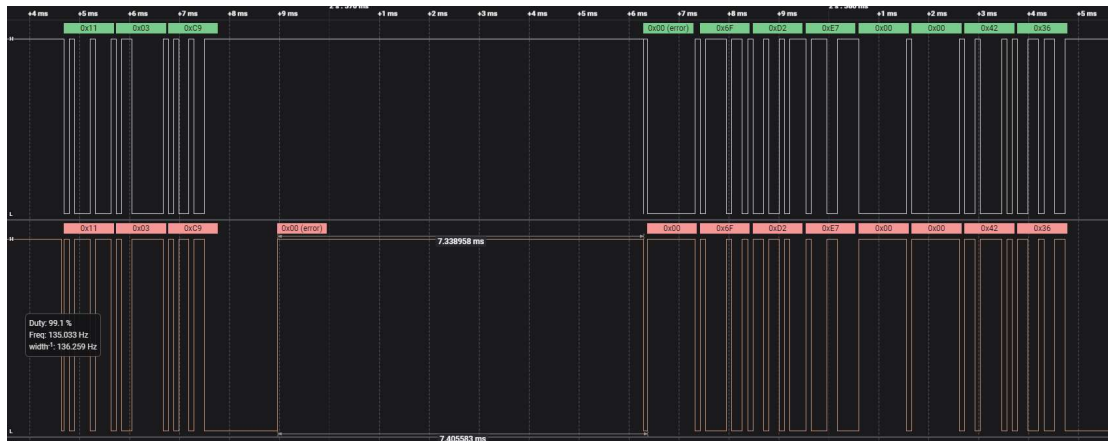
Tab. 8: Formáty přijímaných hodnot [6]

$$U_{bat} = \frac{UBAT}{2048} = \frac{32168}{2048} = 15,707V \quad (25) [6]$$

$$T_{bat} = \frac{TEMP}{128} = \frac{7524}{128} = 58,781^{\circ}C \quad (26) [6]$$

### 4.1.3 Experimentální ověření

K řídicí jednotce byla připojena balancovací jednotka skrz komunikaci RS-485. Logický analyzátor je připojen na data A a B. Na obrázku vidíme 3 bytový paket, který posílá řídicí jednotka na sběrnici. Tuto zprávu přijme balancovací jednotka a během 8ms posílá 8 bytový paket. Na Obr. 22 lze též vidět, že RS-485 komunikace je diferenciální.

Obr. 22: Komunikace řídicí jednotky s balancovací jednotkou; data A *CHI*, data B *CH2*.

## 4.2 Komunikace s LCD displejem (CAN B)

Pro konfiguraci CAN modulu CPU byl použit software z katedry výkonové elektroniky z důvodu kompatibility CAN debuggeru, který byl na této katedře vyvíjen a kterému se věnuje další kapitola 4.3. Ze softwaru byla převzata pouze inicializace CAN B modulu.

Jak již bylo zmíněno v 1. kapitole, CAN sběrnice určená pro komunikaci s displejem používá základní strukturu CAN zprávy, kterou vidíme v následující ukázce kódu.

```
typedef struct
{
    uint16_t obj_id;    ///< CAN message object ID - i.e. index of CAN mailbox.
    uint16_t msg_id;   ///< CAN message ID.
    uint16_t len;      ///< If receive mailbox then it consist of length of data received. If
    transmit then it defines message length during setup and afterwards.
    uint16_t data[8];  ///< Data array
} canMsg_t;
```

Zdrojový kód 4: CAN struktura zprávy

Komunikace s displejem je vyřešena tak, že funguje pouze jednosměrně, kdy řídicí jednotka posílá jednotlivé zprávy do displeje. Displej pouze dostává jednotlivé zprávy jako například efektivní hodnoty proudu jednotlivých fází, stav baterie, rychlost a bude vypisovat veškeré chybové stavy, které řídicí jednotka vyhodnotí. Příklad jedné zprávy je v tab. 9. Celý seznam zpráv odesílaných na displej je uveden v příloze.

Object ID	Message ID	Lenght of data	DATA (16bit)	Popis
0x300	0x200	6	EFF_CURR	RMS hodnota ze všech 3. proudů
			U_INTER_CIR	Napětí meziobvodu
			AKCELERATION	Moment motoru
			MOTOR_TEMP	Teplota motoru
			INVERTER_TEMP	Teplota střídače
			U_BAT_PERC	Procentuální hodnota napětí na baterkách (rs-485)
			BAT_TEMP	Teplota na bateriích

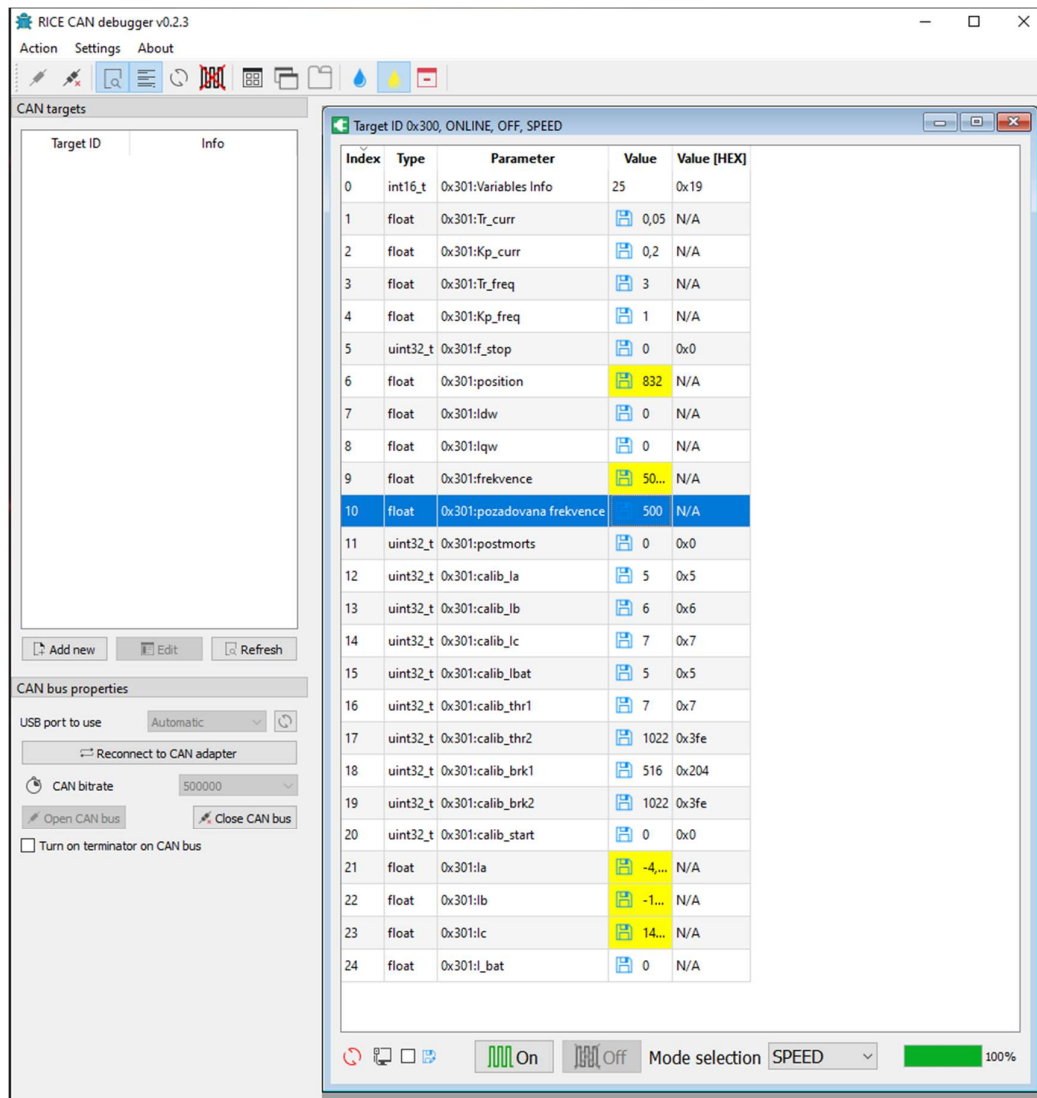
Tab. 9: Příklad CAN zprávy

### 4.3 Debugging softwaru (CAN A)

Debugging je nástroj, který slouží vývojářům k vyhledávání chyb v softwaru ve fázi, kdy se software ladí.

Na katedře výkonové elektroniky byl vyvinut software RICE CAN debugger v0.2.3 a i hardware pro komunikaci mezi řídicí jednotkou a počítačem. Takto je programátor schopný monitorovat různé proměnné a jejich hodnoty, které představují například nějaká měření, indikaci chybových stavů, konstanty pro regulátory a další (viz Obr. 23). Do některých proměnných lze i zapisovat, to je velmi užitečné například při ladění konstant PI regulátorů nebo zadávání požadovaného proudu.

Další výhodou použití tohoto softwaru je ukládání průběhů do post-mortů. To je pole hodnot jedné proměnné, kterou je možné skrze aplikaci stáhnout ve formě .csv souboru o velikosti několika vzorků. Tento průběh se poté dá vykreslit do grafu, viz graf 2.



Obr. 23: RICE CAN debugger v0.2.3

#### 4.4 Kalibrace konstant

CAN debugger má v sobě také funkci ukládat některé proměnné do externí paměti, bylo potřeba upravit pouze softwarovou část komunikace skrze SPI pro použitý MCU TMS320F28335.

Kalibraci lze zpusit pouze v debug režimu motokáry z bezpečnostních důvodů. V předchozím Obr. 23 jsou vidět kalibrační konstanty offsetů AD převodníku (např. calib\_la, calib\_lb, atd,) nebo konstanty regulátorů  $K_P$  a  $T_r$ . Tyto konstanty se při zapnutí řídicí jednotky stáhnou z flash paměti. Pro kalibraci offsetů je vytvořena rutina, kdy si jednotka naměří 1000 vzorků a z nich stanoví jednotlivé offsety. Ty se poté uloží do flash paměti.



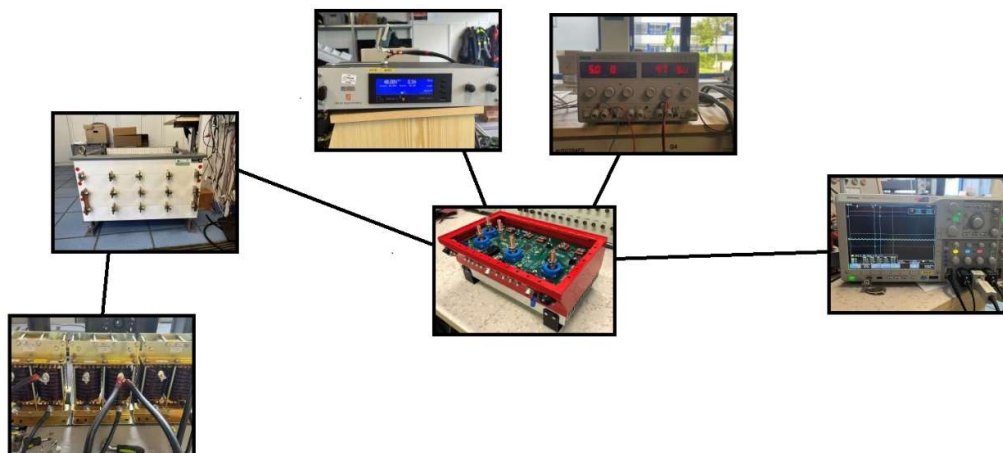
## 5 Oživení řídicí jednotky

Řídicí jednotka s měničem nikdy v minulosti nebyla testována na vyšších výkonech. Proto z důvodu bezpečnosti byla jednotka připojena na RL zátěž a do řídicího algoritmu se implementovalo skalární řízení. Zde bylo třeba opravit menší hardwarové nedostatky, viz další podkapitola.

Dále se jednotka připojila k motoru a do řídicího algoritmu se implementovalo vektorové řízení. Zde bylo třeba odladit správné vyčítání polohy, správné nastavení PI regulátorů, implementace feedforwardu a jednoduché rampy.

### 5.1 Zatížení jednotky s RL zátěží

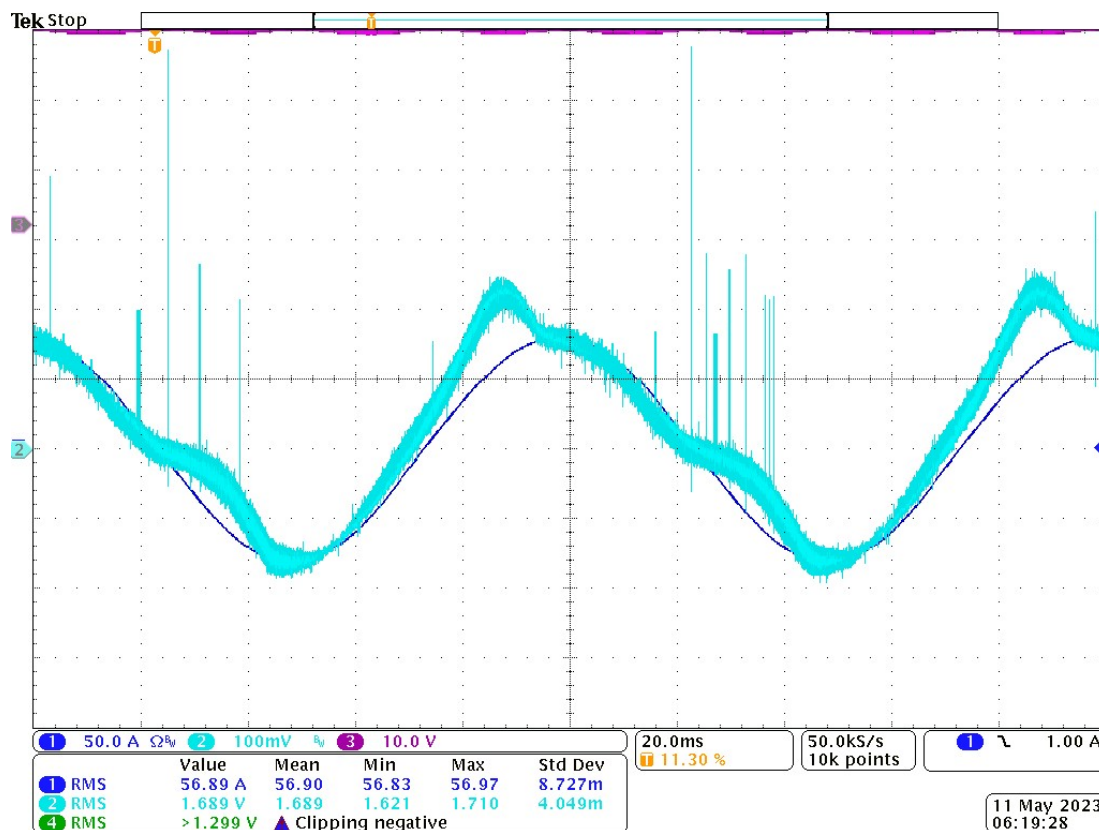
Na Obr. 24 vidět schéma měřicího stanoviště, kde k řídicí jednotce je připojena RL zátěž zapojena do hvězdy. K jednotce jsou připojené dva zdroje, jeden pro řídicí desku (vpravo) a druhý napájí střídač (vlevo DELTA) z důvodu jednoduššího ladění programu, kdy lze vypínat výkonový zdroj ale řídicí jednotka může nadále pracovat. Měření průběhů probíhá před osciloskopem a debugging softwaru skrze laptop, který komunikuje po CAN a JTAG.



Obr. 24: Měřicí stanoviště s RL zátěží

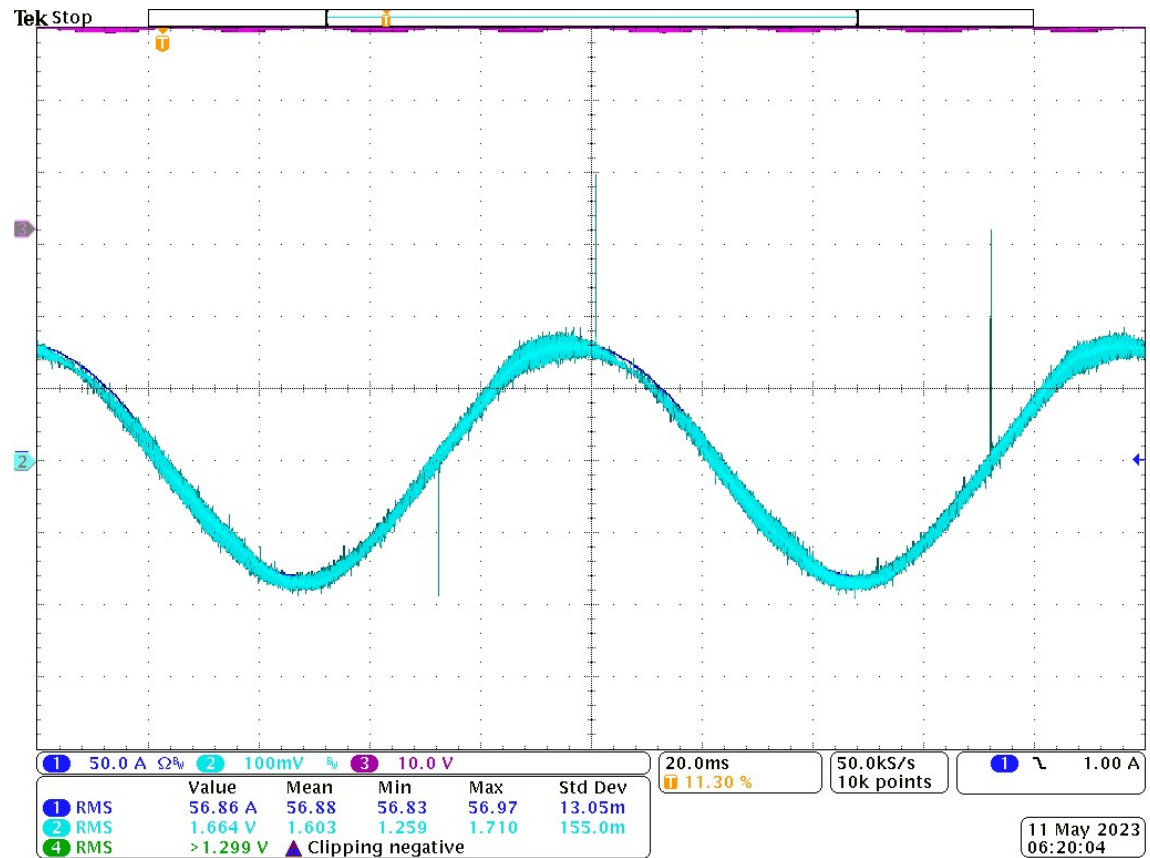
Na Obr. 25 je vidět průběh proudu 1. fáze do RL zátěže (CH1 - modrá) a druhý průběh je naměřený z čidla proudu (CH2 – tyrkysová). Z obrázku je jasně vidět, že naměřený průběh proudu řídicí jednotkou je velmi zdeformovaný. Při nižších hodnotách proudu se tento jev neprojevil, s narůstajícím proudem však narůstá i elektromagnetické rušení, které střídač produkuje. Na oscilogramech se u výstupu z DA převodníku objevují špičky napětí, které

tam ve skutečnosti nejsou. Kontrola proběhla skrze uložená data v procesoru, která byla následně vykreslena a špičky na průběhu měřeného proudu nebyly zaznamenány.



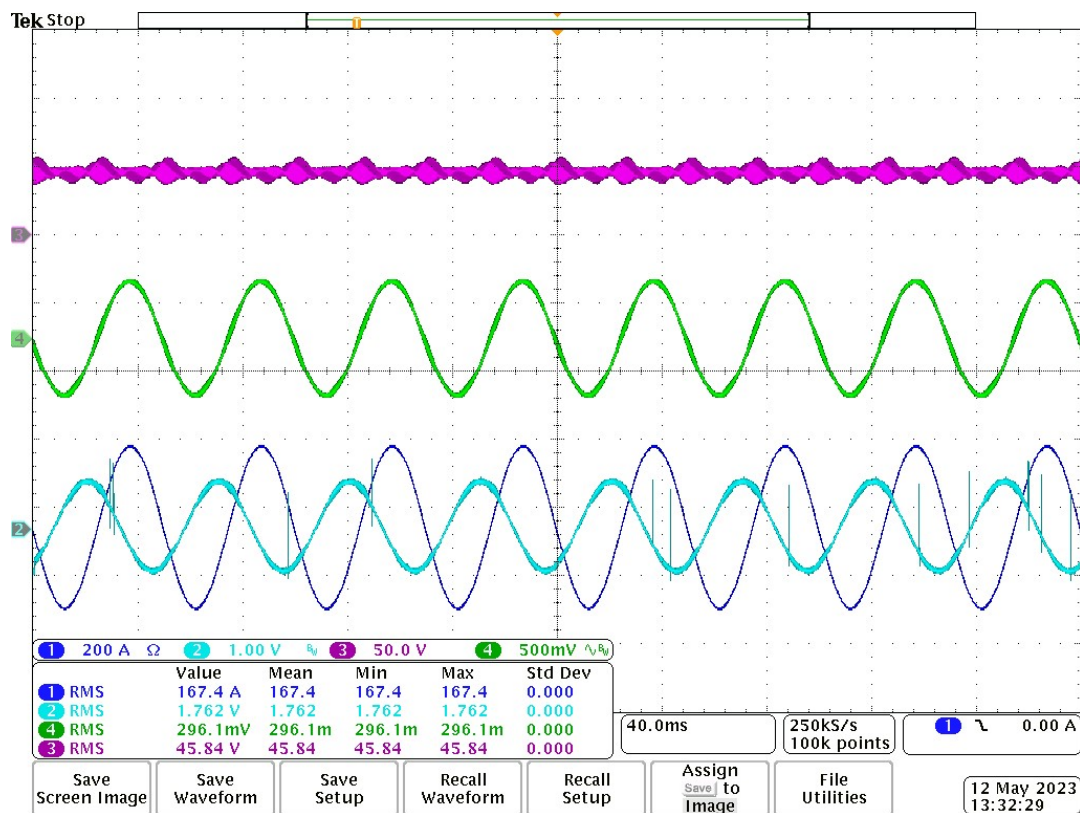
Obr. 25: Průběh zarušeného proudu. Průběh proudu 1. fáze CH1 (modrá) 50A/div a naměřený průběh proudu 1. fáze, CH2 (Tyrkisová) – 100mV/div (50A/div)

Po dalším zkoumání se přistoupilo k úpravě řídicí jednotky. Mezi střídač a řídicí desku se vložil pozinkovaný plech, který odstínil elektromagnetické rušení. Na následujícím Obr. 26 je vidět průběh po hardwarové úpravě.

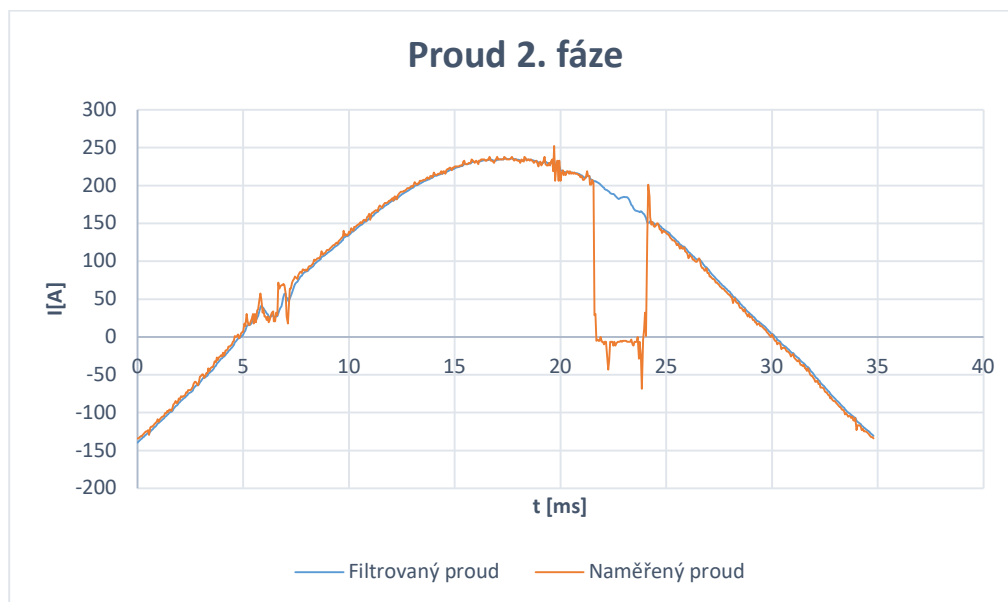


Obr. 26: Průběh proudu po hardwarové úpravě; Průběh proudu 1. fáze *CHI* (modrá) 50A/div a naměřený průběh proudu 1. fáze po hardwarové úpravě, *CH2* (Tyrkisová) – 100mV/div (50A/div)

Na následujícím Obr. 27 jde vidět testování řídicí jednotky na velké proudy kolem 170 Arms s frekvencí 20 Hz.



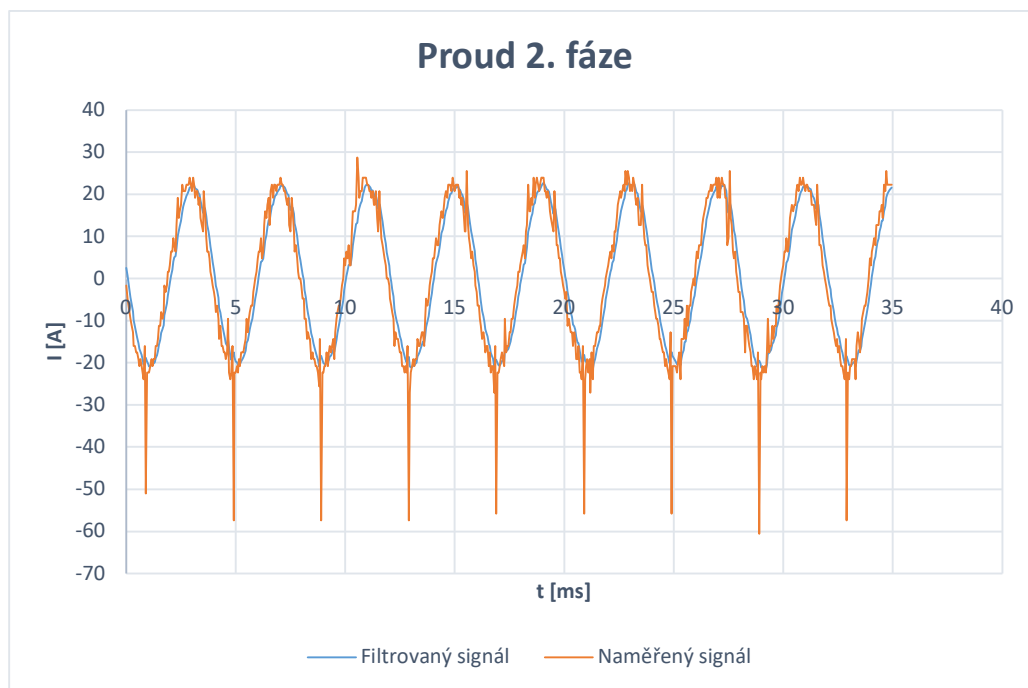
Obr. 27: Měření proudů; Průběh proudu 1. fáze *CH1* (modrá) - 200A/div, filtrovaný průběh proudu 2. fáze *CH2* (Tyrkisová) - 400A/div, napětí v meziobvodu *CH3* (fialová) - 50V/div a napětový výstup z čidla proudu 1. fáze *CH4* (zelená) - 500mV/div při  $I = 167.4 \text{ Arms}$  a  $f = 20 \text{ Hz}$ .



Graf 2: Vyhodnocený průběh proudu 2. fáze procesorem při  $I = 167.4 \text{ Arms}$ ,  $f = 20 \text{ Hz}$

Z Obr. 27 jde vidět, že výstup z proudového čidla je nezarušený, ale v grafu 2 jde vidět, že se stále do procesoru dostává nějaké rušení. Další možnost je špatně vyhodnocený signál AD převodníkem. Nakonec se přistoupilo k řešení vyfiltrování, naměřených dat a to tak, že je použit logaritmický filtr na tři vzorky a každá proudová špička s větší derivací proudu je vyhozena a nahrazena dopočítaným vzorkem z dalších dvou fází. Toto je možné jen tehdy, kdy tyto chyby jsou v jeden okamžik jen na jedné fázi.

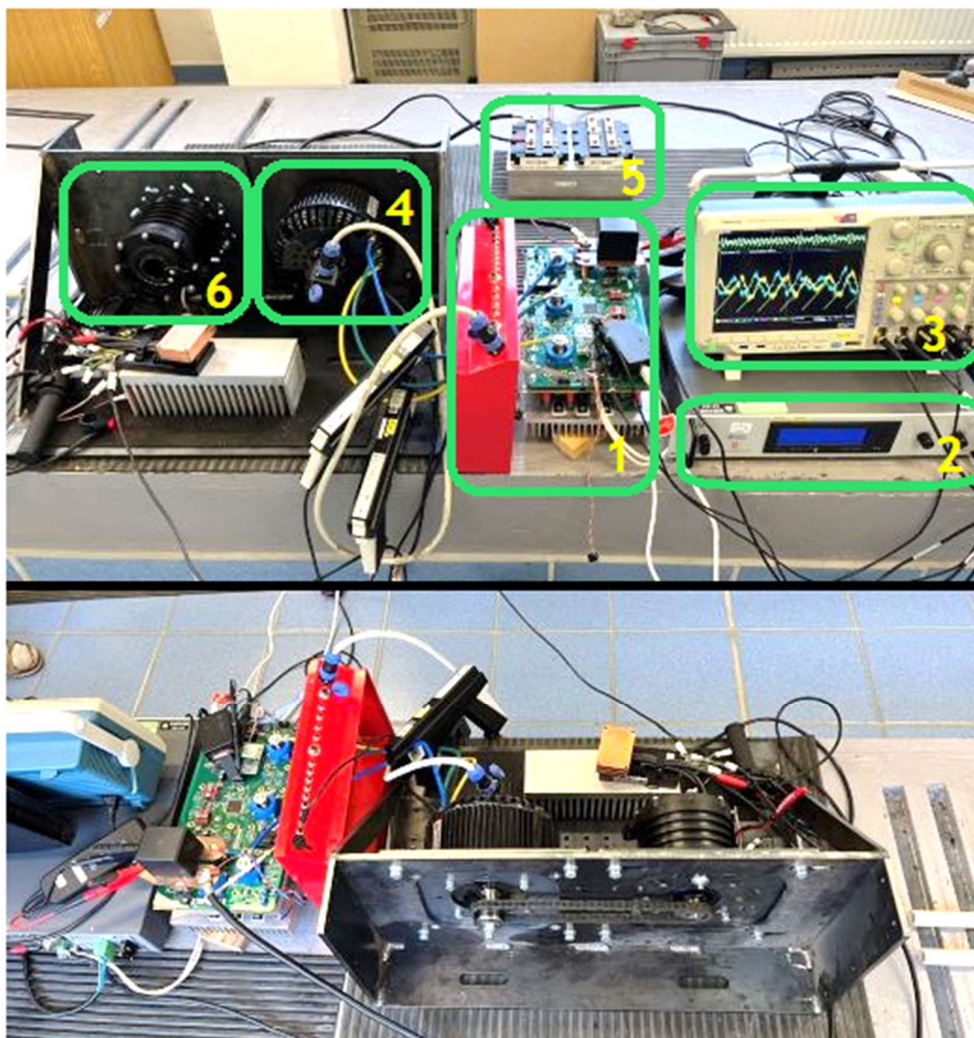
Na následujícím grafu 3 je průběh proudu 2. fáze při frekvenci 250 Hz a velikosti proudu 16 Arms.



Graf 3: Vyhodnocený průběh proudu 2. fáze procesorem při  $I = 16 \text{ Arms}$ ,  $f = 250 \text{ Hz}$

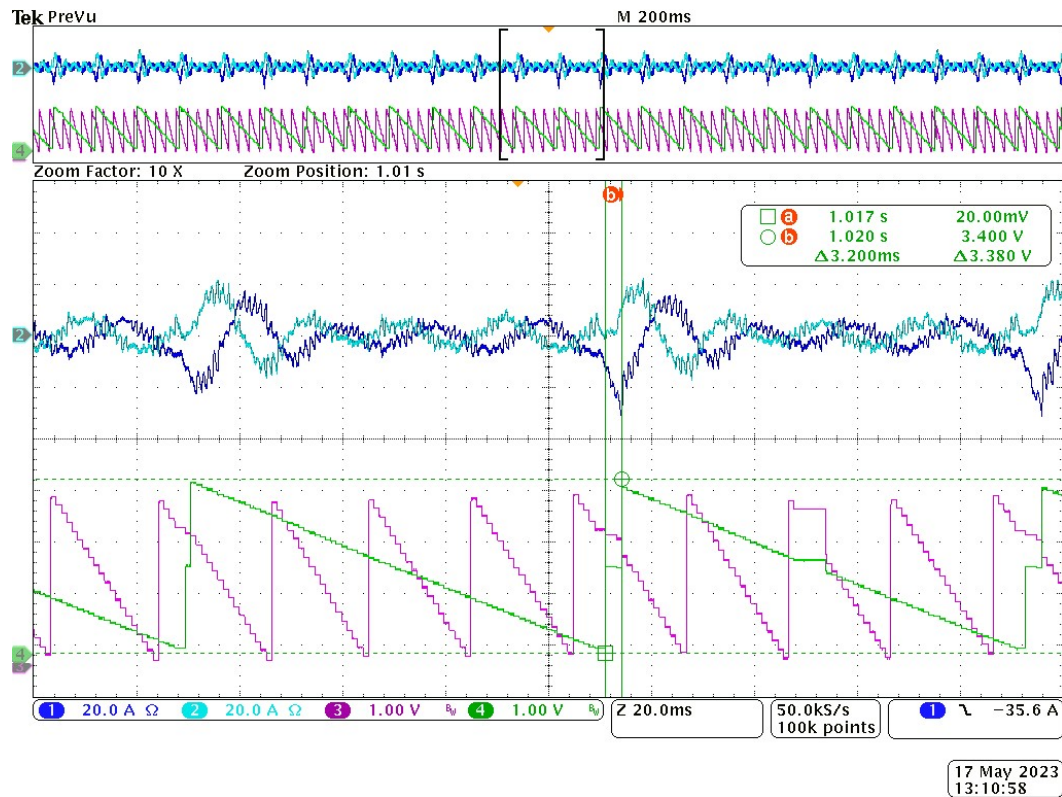
## 5.2 Testování jednotky s PMSM motorem

Na *Obr. 28* je vidět měřicí stanoviště s PMSM motorem (4), který je připojený k druhému motoru (6), který simuluje zátěž. PMSM motor je připojen k řídicí jednotce (1), na jednotlivých fázích měříme osciloskopem (3) průběhy proudů a přes DA převodník výstup z čidla otáček. Zdroj (2) slouží k napájení řídicí jednotky. Mezi zdroj a napájení řídicích jednotek motorů je připojena dioda (5), která zamezuje rekuperaci do zdroje, aby nedošlo k jeho poškození.



*Obr. 28: Měřicí stanoviště s PMSM motorem*

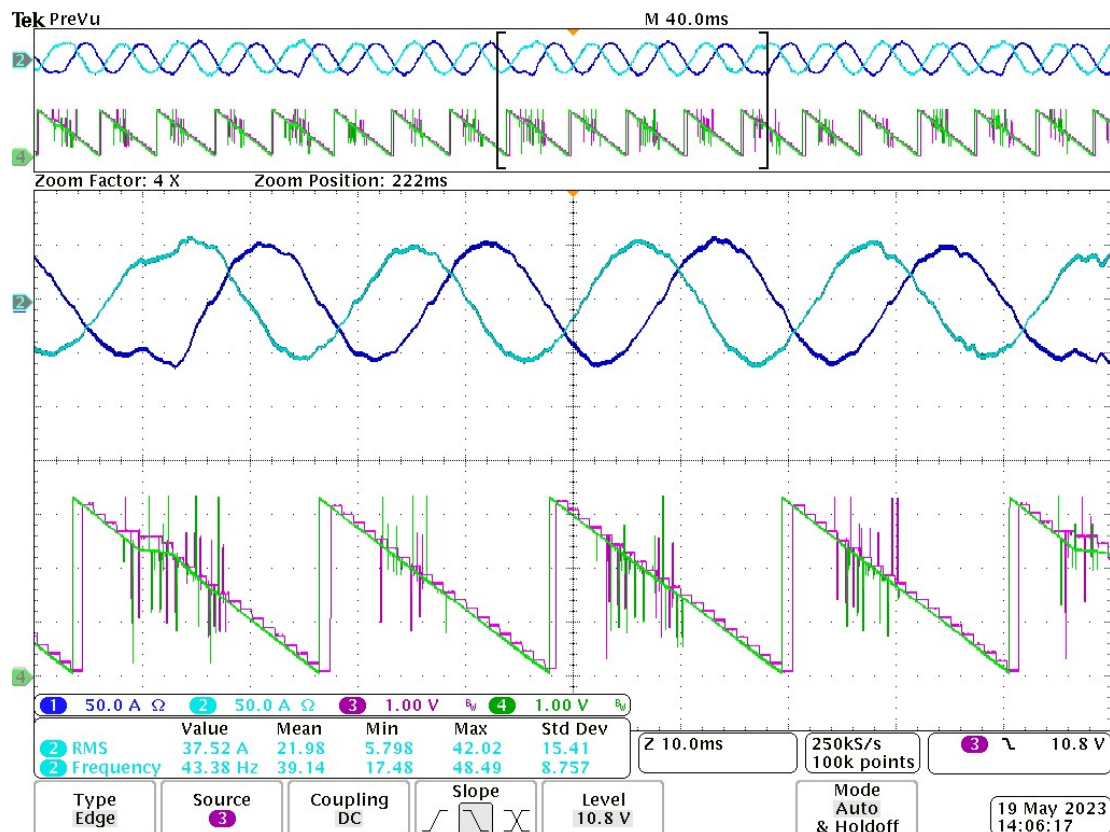
Na Obr. 29 lze vidět, že díky menší periodě měření je jedna hodnota polohy použita pro více cyklů přerušení od AD převodníku, ve kterém dochází k řízení. Kvůli tomu dochází ke zvlnění výstupního proudu do motoru, což je nežádoucí. Dále jde také vidět, že je chyba při přechodu z  $0$  na  $2\pi$ . Tuto chybu může generovat buď čidlo nebo jednotka nemusí v eCAP modulu správně zaznamenat přijatá data od čidla.



Obr. 29: Měření úhlu natočení; Průběh proudu 1. fáze *CH1* (modrá) - 20A/div, průběh proudu 2. fáze *CH2* (Tyrkisová) - 20A/div, úhel natočení rotoru používaný v transformacích ( $0-2\pi$ ) *CH3* (fialová) - 1V/div a úhel natočení rotoru ( $0-2\pi$ ) *CH4* (zelená) - 1V/div při  $f_{el} = 20\text{Hz}$ .

Z toho důvodu byl navržen interpolační filtr, který při každém cyklu regulace přepočítává hodnotu polohy a dochází i k částečnému vyfiltrování chyby při přechodu nulové pozice rotoru. Na

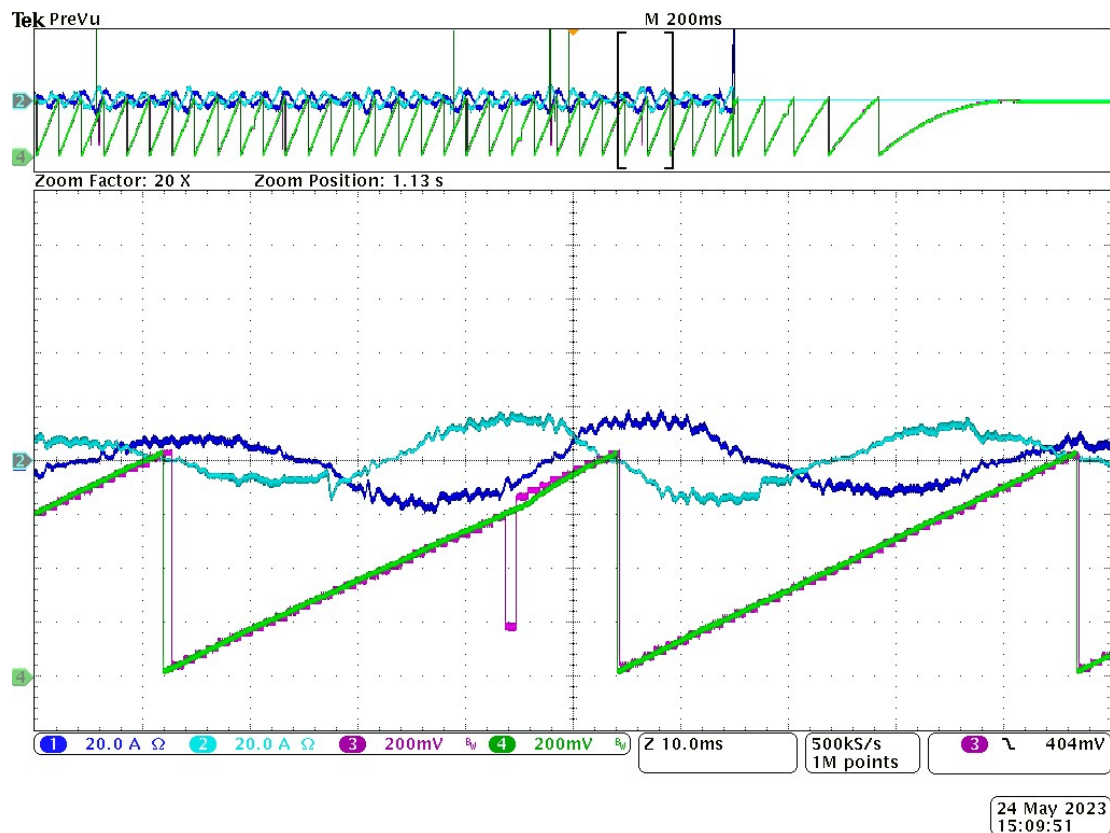
Obr. 30 jsou vidět průběhy po aplikaci interpolačního filtru. Vidíme, že chybu ale nedokáže úplně odstranit, ale průběh proudu se několikanásobně zlepšil. Při tomto experimentu byl motor nezatížený.



Obr. 30: Měření úhlu natočení s interpolačním filtrem; Průběh proudu 1. fáze CH1 (modrá) - 50A/div, průběh proudu 2. fáze CH2 (Tyrkisová) - 50A/div, úhel natočení rotoru používaný v transformacích (0-2 $\pi$ ) CH3 (fialová) - 1V/div a úhel natočení rotoru (0-2 $\pi$ ) CH4 (zelená) - 1V/div při  $f_d = 180\text{Hz}$ .

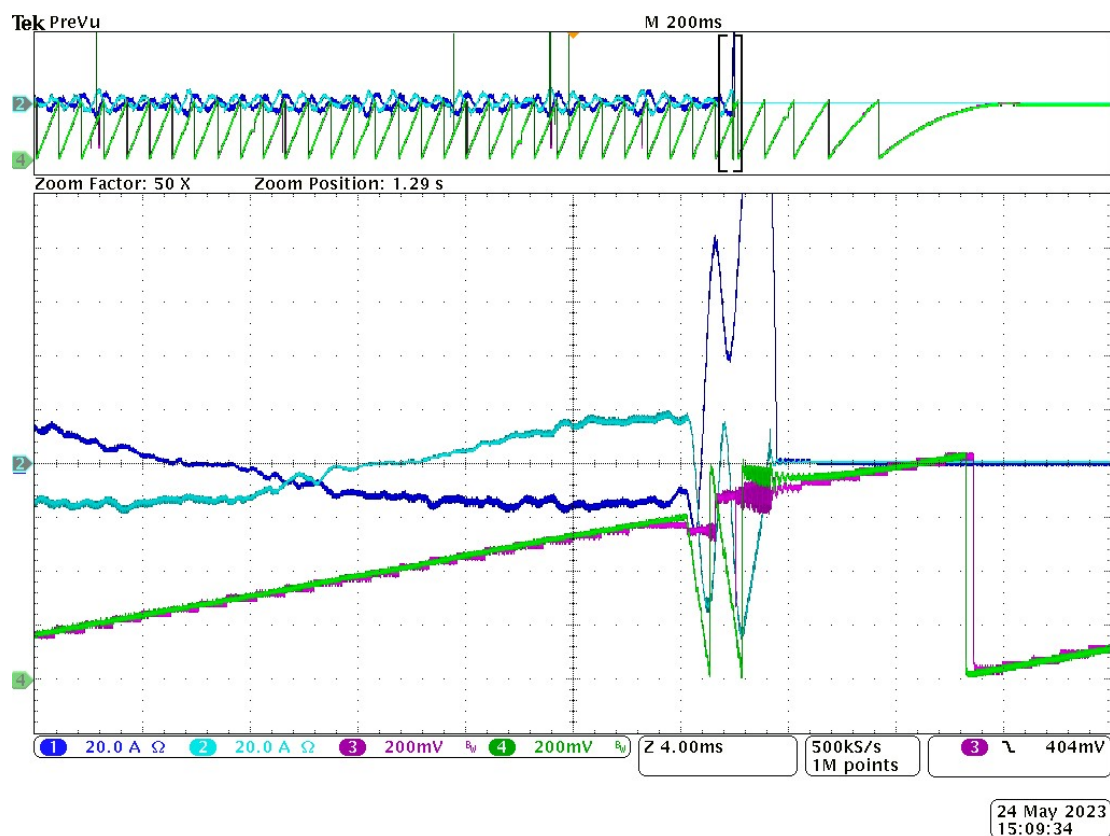


Při dalších testech se připojil druhý motor jako zátěž skrze řetěz viz Obr. 28. Z dalších testů a měření vyplínulo, že námi navržený interpolační filtr, si dokáže poradit s velkými falešnými skoky v poloze, které generuje čidlo viz obr. 31.



Obr. 31: Měření průběhů proudu a polohy při zatížení motoru; Průběh proudu 1. fáze CH1 (modrá) - 20A/div, průběh proudu 2. fáze CH2 (Tyrkisová) - 20A/div, úhel natočení rotoru používaný v transformacích (0-2 $\pi$ ) CH3 (fialová) - 200mV/div a úhel natočení rotoru (0-2 $\pi$ ) CH4 (zelená) - 200mV/div při  $f_{el} = 150\text{Hz}$ .

Z dalšího Obr. 32 je ale vidět, že jakmile, přijde změna, která je menší, ale také falešná, tak si s ní filtr poradit nedokáže a jeho reakcí na změnu, začne přepočítávat špatnou změnu polohy. Na to reagují regulátory proudu, který zapříčiní otočení fáze proudu, který vystoupá, do takové velikosti, že zareagují softwarové proudové ochrany, které jsou v tomto případě nastaveny na 150 A. Po reakci proudových ochran dojde k bezpečnému odpojení.



Obr. 32: Měření průběhů proudu a polohy při zatížení motoru 2; Průběh proudu 1. fáze CH1 (modrá) - 20A/div, průběh proudu 2. fáze CH2 (Tyrkisová) - 20A/div, úhel natočení rotoru používaný v transformacích (0-2 $\pi$ ) CH3 (fialová) - 200mV/div a úhel natočení rotoru (0-2 $\pi$ ) CH4 (zelená) - 200mV/div při  $f_{el} = 150\text{Hz}$ .

Seznam použitých měřicích přístrojů je v tab 10.

Zdroj	Delta Elektronika – SM66-AR-110
Osciloskop	Tektronix MSO 4104B
Proudová Sonda	Tektronix TCPA400
Laboratorní Zdroj	TTI EX354RT
Diferenciální sonda	Tektronix P50205A

Tab. 10: Seznam použitých měřicích přístrojů

## 6 FMEA analýza

FMEA (Failure Mode and Effect Analysis) je jednou z nejpoužívanějších technik analýzy rizik v projektovém inženýrství, která slouží k identifikaci možných způsobů selhání a předvídání jejich následků. Umožňuje identifikovat potenciální problémy výrobku ještě před tím, než se dostanou do konečné fáze nebo k zákazníkovi. Při použití FMEA se zkoumá každá součást, aby se identifikovaly možné poruchy. Zvažují se tři opatření: pravděpodobnost výskytu poruchy, dopad nebo závažnost poruchy a schopnost odhalit poruchu dříve, než dojde k jejímu vzniku.

Přestože FMEA upřednostňuje kritičtější poruchy, vyžaduje také analýzu každé součásti systému, a to může být časově velmi náročné. Také neexistuje žádné přesné pravidlo pro stanovení pravděpodobnosti výskytu a detekce. [21]

Příklad zpracované FMEA analýzy je v tab. 11 a celá analýza se nachází v příloze.

Funkce	Potencionální Chyba	Příčina	Následek	Ošetřeno
Střídač	Nadproudy jednotlivých fází	Mohlo dojít ke zkratu na jedné z fází	Okamžité vypnutí výkonových výstupů + info. na displej	Měření proudu jednotlivých fází
.	.	.	.	.
.	.	.	.	.
CAN	Nefunkční komunikace	Přepálení pojistek	Výměna pojistky	Signalizace diod

Tab. 11: Příklad FMEA analýzy

## Závěr

Cílem této práce bylo navrhnout řídicí algoritmus pro jednotlivé periférie ovládané řídicí jednotkou elektrické motokáry.

V první části diplomové práce byl popsán hardware řídicí jednotky a periférie použité v elektronické motokáře.

V druhé části byli popsány 3 typy řízení PMSM motoru. Zvolilo se vektorové řízení a byli popsány jeho jednotlivé části jako Clarkové a Parkovo transformace, PI regulátory a feedforward.

Ve třetí části byli nastaveny všechny potřebné periférie používané v řízení. Byl ověřen výstup z PWM modulu, vyčítání z AD převodníků. Úspěšně byl nastaven a otestován eCAP k zachycování PWM signálu z čidla polohy. Na konci této části bylo nastaveno měření rychlosti skrze eQEP periférii. Měření na osciloskopu, bylo ověřeno fungování vyčítání polohy a rychlosti.

Ve čtvrté části byli teoreticky popsány komunikace RS-485 a CAN. Došlo k experimentálnímu ověření komunikace RS-485, kdy balancovací jednotka úspěšně odpovídá na dotaz od řídicí jednotky. CAN A byl otestován skrze RICE CAN debugger, přes který se v další fázi software ladil. Inicializace CAN B vychází z podobné struktury jako CAN A. Byli navrženy zprávy, které jsou odesílané na displej. Tyto zprávy byli úspěšně zachyceny CAN analyzátozem.

Předposlední část diplomové práce je věnována ožívování řídicí jednotky, došlo k mnoho hardwarovým změnám, které jsou přiloženy v příloze C, formou tabulky. Došlo k testování na RL zátěži, kdy došlo k odladění měření proudů. V další fázi ožívování byl připojen k řídicí jednotce PMSM motor. V tomto bodě bylo nutné provést úpravy ve vyčítání polohy, které vedli k implementaci interpolačního filtru viz kapitola 5.2. Na motoru bez zátěže úspěšně dopočítával polohu, ale když došlo k jeho zatížení, tak filtr nedokázal správně vyhodnotit polohu. Z tohoto důvodu je toto řešení nevhodné a bude nutné přejít buď na bezsenzorové řízení a nebo dojde k přechodu na jiný typ čidla.

V poslední části diplomové práce byla úspěšně sepsaná FMEA analýza, ve které byli pospány možné chybné stavy, které mohou nastat a byli navrženy způsoby, které tyto chybné stavy řeší a snaží se jim předejít.

## Literatura

- [1] Suchý, Ondřej. "Implementace řídicí jednotky motokáry s elektrickým pohonem." (2019).
- [2] [online] [cit.21-05-23] Dostupné z: <https://www.farnell.com/datasheets/1524365.pdf>
- [3] [online] [cit.21-05-23] Dostupné z: <https://docs.rs-online.com/7d34/0900766b81327f26.pdf>
- [4] [online] [cit.21-05-23] Dostupné z: [https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2010/jps89\\_cdq2\\_mam584/jps89\\_cdq2\\_mam584/HCPL-7520-300E-Avago-datasheet-7803.pdf](https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2010/jps89_cdq2_mam584/jps89_cdq2_mam584/HCPL-7520-300E-Avago-datasheet-7803.pdf)
- [5] [online] [cit.21-05-23] Dostupné z: <https://smartece-sensors.eu/cms/media/Datasheets/Temperature/SMT172-datasheet.pdf>
- [6] Polach, Tomáš. "Realizace měřících obvodů Li-ion baterie." (2013).
- [7] [online] [cit.21-05-23] Dostupné z: <http://www.ti.com/product/SN65HVD232>
- [8] Beran, Martin. "Aplikační SW pro zobrazovací jednotku elektrovozidla." (2019).
- [9] Blaško, Jan. "Návrh hardwaru pro CAN simulátor." (2020).
- [10] [online] [cit.21-05-23] Dostupné z: <https://www.jameco.com/Jameco/Products/ProdDS/2135881.pdf>
- [11] [online] [cit.21-05-23] Dostupné z: <https://docs.rs-online.com/7d34/0900766b81327f26.pdf>
- [12] Essick, John. Hands-on introduction to LabVIEW for scientists and engineers. Fourth edition. New York: Oxford University Press, [2019]. ISBN 9780190853068.
- [13] Prokop, Libor, and Pavel Grasblum. "3-phase PM synchronous motor vector control using a 56F80x, 56F8100, or 56F8300 device." Freescale Semi—conductor Application Note (2005).
- [14] PEROUTKA, ZDENĚK. Výukové materiály z předmětu PVE2.2021[cit.21-05-23]. Dostupné z: <https://courseware.zcu.cz/>
- [15] Petruška, Lubomír. Model synchronního motoru [online]. Brno, 2008 [cit. 2023-05-21]. Dostupné z: <http://hdl.handle.net/11012/2498>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Libor Veselý.
- [16] [online] [cit.21-05-23] Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8554788>
- [17] Haugen, Finn. "Basic dynamics and control." Porsgrunn: TeachTech 1 (2010).

- [18] Belda, Květoslav. "Mathematical modelling and predictive control of permanent magnet synchronous motor drives." *Transactions on Electrical Engineering* 2.4 (2013): 114-120.
- [19] [online] [cit.21-05-23] Dostupné z: <https://www.ti.com/lit/ug/sprui07/sprui07.pdf>
- [20] PEROUTKA, ZDENĚK. Výukové materiály z předmětu MRP2021[cit.21-05-23]. Dostupné z: <https://courseware.zcu.cz/>
- [21] Segismundo, A. and Augusto Cauchick Miguel, P. (2008), "Failure mode and effects analysis (FMEA) in the context of risk management in new product development: A case study in an automotive company", *International Journal of Quality & Reliability Management*, Vol. 25 No. 9, pp. 899-912. <https://doi.org/10.1108/02656710810908061>

## Přílohy

### Příloha A

Object ID	Message ID	Lenght of data	DATA (16bit)	Popis
0x300	0x200	6	EFF_CURR	RMS hodnota ze všech 3. proudů
			U_INTER_CIR	Napětí meziobvodu
			AKCELERATION	Moment motoru
			MOTOR_TEMP	Teplota motoru
			INVERTER_TEMP	Teplota střídače
			U_BAT_PERC	Procentuální hodnota napětí na baterkách (rs-485)
			BAT_TEMP	Teplota na bateriích
			0x300	0x201
BALANCING	Informace jestli balanční jednotky balancují			
CHARGE_ERROR	Neočekávaný proudy skrze balanční tranzistory balančních jednotek			
CHARGE_DONE	Informace o ukončení nabíjení			
U_BAT_PERC	Procentuální hodnota napětí			
BAT_TEMP	Teplota na bateriích			
ERR_BAT	Chyba balancování, vysoké napětí, přehřátí			
0x300	0x202	6		
			COMM_ERROR	Komunikace s balancovacíma jednotkama
			PADS_ERROR	Chyba pedálu brzdy nebo plynu
			DRIVERS_ERROR	větvojvý zkrat
			CURR_ERROR	Nadproudy
			VOLT_ERROR	Nízké napětí meziobvodu

## Příloha B

Funkce	Potencionální chyba	Příčina	Následek	Ošetřeno
Baterie	Přepětí na bateriích	Přebíjení baterie rekuperací nebo nabíječkou	Vypnutí nabíječky nebo zakázání rekuperace	Kontrola napětí na bateriích pomocí balancovacích jednotek
	Malé napětí na bateriích	Některá z baterií je vybitá	Při 15% baterie je uživatel upozorněn skrze displej a omezí se výkon. Při 5% procentech baterie dojde k bezpečnému odpojení jednotky od baterií	Kontrola napětí na bateriích pomocí balancovacích jednotek
	Přehřátí baterií	Dlouhý provoz, rychlé cyklování,	Při určité teplotě by došlo k upozornění na displeji, při dalším nárůstu teploty omezíme výkon motokáry a při kritické teplotě dojde k bezpečnému vypnutí	Kontrola teploty na bateriích pomocí balancovacích jednotek
	Přehřátí balančního tranzistoru	Při nabíjení dojde k přehřátí balančního	Sníží nabíjecí proud	Kontrola teploty na balančním tranzistoru uvnitř balancovacích jednotek
	Neočekávaný proud balančním tranzistorem		Po opakovaném přijetí chyby se zobrazí informace na displeji a je vyhodnocována teplota na tranzistoru a při překročení teploty dojde k bezpečnému vypnutí	Kontrola proudu procházející balančním tranzistorem uvnitř balancovacích jednotek
	Při balancování neklesá napětí na akumulátoru	Poškození balančního tranzistoru	Vypnutí nabíjení a uložení chyby do paměti a rozsvícení signalizační diody	Kontrola napětí na bateriích pomocí balancovacích jednotek



Balanční Jednotka	Chyba měření teploty	Poškození balancovacích jednotek	Omezení na spodní dva režimy s nižším výkonem, zobrazení na displeji a nebo bezpečné vypnutí	Kontrola ostatních teplot balancovacích jednotek a vycházíme z předpokladu, že pokud jsou teploty na funkčních jednotkách podobné, lze očekávat podobnou teplotu i na nefunkční jednotce. Pokud je nefunkční více jak jedna jednotka dojde k bezpečnému vypnutí
	Chyba měření teploty balančního tranzistoru	Poškození balancovacích jednotek	Nouzový režim motokáry	Pomocí Balancovacích jednotek
	Balancovací jednotka nebo více jednotek neodpovídá	Poškození balancovacích jednotek	Pokud se jedná o jednu dotazovanou jednotku, která se několikrát po sobě neozve, nemusí to být ještě problém a napětí na akumulátoru se dá dopočítat ze zbylých tří jednotek, ale chyba by se měl řešit co nejdříve, jelikož nemáme další informace o jednom článku. Pokud jednotek neodpoví víc nebo žádná, dojde ke kontrole napětí meziobvodu, pokud ok, omezí se výkon. Pokud napětí meziobvodu bude malé, po té se motokára bezpečně vypne	Softwarově ošetřeno
	Chyba CRC výpočtu	Chybné data	po několika erorech CRC výpočtu dojde ke stejnému postupu, jako když je balanční jednotka poškozena	Softwarově ošetřeno
Motor	Přehřadí motoru	Dlouhý a výkonově náročný provoz	Omezení výkonu motokáry	Kontrola teplot na motoru
Střídač	Přehřadí chladiče	Dlouhý a výkonově náročný provoz	Omezení výkonu motokáry	Kontrola teplot na střídači

	Nadproudy jednotlivých fází	Mohlo dojít ke zkratu na jedné z fází	Okamžité vypnutí výkonových výstupů a zobrazení informace na displeji	Měření proudů jednotlivých fází
	Malé napětí meziobvodu	Souvisí s vybitím baterií	Souvisí s informací o malém napětí na bateriích	Měření napětí meziobvodu
Drivery	desaturace MOS-FET tranzistorů	Větvový zkrat	Reakce Trip-Zone a vypnutí výkonových výstupů	Driver obsahuje desaturační ochranu
Čidlo polohy	Chybná data polohy při přechodu 0° natočení motoru	Špatně usazené čidlo polohy, chyba čidla, chyba vyhodnocování v řídicí jednotce	Interpolační filtr polohy nebo přejít na bezsenzorové řízení polohy	Ošetřeno Interpolačním filtrem
Řídicí deska	Chybné AD převody	Zarušení řídicí jednotky	možná hardwarová úprava nebo filtr proudů	Ošetřeno filtrem
Pedály	Vyhodnocení chyby na inverzních snímačích	Chyba na čidle v pedálu	Chybová hláška na displeji	Softwarově ošetřeno
Startovací rutina	Malé napětí meziobvodu , sešlápnutý pedál plynu	(vybité baterie), uživatel sešlápně pedál	Motokára se nenastartuje a vypne	
Bezpečné vypnutí	-		Uložení důležitých chybových hlášek	
RS-485	nefunkční komunikace skrze RS-485	Přepálení pojistek	výměna pojistky	Signalizace diod
CAN	nefunkční komunikace skrze CAN	Přepálení pojistek	výměna pojistky	Signalizace diod

## Příloha C

RS-485	U dvou jednotek shořeli pojistky, třetí nefunkční, funkční pouze jedna		
Deska	výměna dolního prvku složený ze 3. tranzistorů na 2. fázi z důvodu průrazu		
	Z důvodu generování falešných pulzů z driveru byli přidány kapacity do blízkosti výstupu driveru o velikosti 1 mikroF		
	Změna Gate odporů z (najít ve schematu) na 100ohm		
	Nalezeno rušení na analogových vstupech do procesoru.	Změna kapacit u filtru analagového měření proudu z 1nF na 100nF	
		Úprava gate odporů na střídači z 0,5 ohm na 6,8ohm za účelem zpomalení přepínání	
		zvětšení vzdálenosti od měniče	
Přidání pozinkovaného plechu mezi měnič a řídicí desku			
Ladění filtru na odstranění špiček na měřených proudech vlivem rušení			
čidlo otáček	Pomalé vyčítání. Jedna hodnota polohy jednou za 1 ms.		
	Ladění interpolačního filtru		

## Příloha D

Software pro řízení motokáry je umístěn online na portále [portal.zcu.cz](http://portal.zcu.cz)