



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ANALÝZA CHOVÁNÍ MĚNIČE PŘI ŘÍZENÍ PMSM MOTORU

ANALYSIS OF PMSM MOTOR INVERTER BEHAVIOUR

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. David Buchal

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Blaha, Ph.D.

BRNO 2020

# Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. David Buchal

**ID:** 186037

**Ročník:** 2

**Akademický rok:** 2019/20

## NÁZEV TÉMATU:

### **Analýza chování měniče při řízení PMSM motoru**

#### **POKYNY PRO VYPRACOVÁNÍ:**

1. Seznamte se s problematikou řízení PMS motorů s permanentními magnety.
2. Seznamte se s poskytnutým přípravkem (motor, měnič, řídicí jednotka). Řídicím procesorem je AURIX od firmy Infineon s otevřeným řídicím programem.
3. Přepište ovladač měření analogových hodnot, aby bylo možné měřit hodnoty fázových proudů v průběhu periody PWM.
4. Naměřená data uložte do paměti a následně přeneste do PC a zobrazte v prostředí MATLAB Simulink.

#### **DOPORUČENÁ LITERATURA:**

[1] Sul, S. K.: Control of Electric Machine Drive Systems, Wiley-IEEE Press, 2011, ISBN: 978-0-470-59079-9

Další dle doporučení vedoucích.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 1.6.2020

**Vedoucí práce:** doc. Ing. Petr Blaha, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**  
předseda oborové rady

#### **UPOZORNĚNÍ:**

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

## **ABSTRAKT**

Tato diplomová práce se zabývá návrhem programového řešení pro mikrokontrolér AURIX TC234 od společnosti Infineon Technologies, které umožní vzorkování signálu s požadavkem na co nejkratší periodu vzorkování a na libovolné nastavení okamžiku zahájení měření. Toto řešení je následně přidáno do ukázkové aplikace vektorového řízení bezkartáčového stejnosměrného motoru a otestováno. Získané hodnoty lze dále využít například pro analýzu průběhu proudů motoru, ze kterých by mohlo být možné posoudit stav spínacích tranzistorů měniče.

## **KLÍČOVÁ SLOVA**

AURIX TC234, eMotor Drive Kit, PMSM, BLDC, FOC, vektorové řízení, HW spouštění A/D převodu, CCU6

## **ABSTRACT**

This master's thesis deals with the design of a software solution for the AURIX TC234 microcontroller by Infineon Technologies which enables a user to start sampling of the signal at any required moment. A sampling time should be set to the lowest possible value. This solution is then added to the application which is used for a field oriented control of the brushless direct current motor. The obtained values can be further used, for example, for analyzing the waveform of motor currents from which it might be possible to determine the condition of the transistor inverter.

## **KEYWORDS**

AURIX TC234, eMotor Drive Kit, PMSM, BLDC, FOC, field oriented control, HW triggering A/D conversion, CCU6

## **Bibliografická citace**

BUCHAL, David. *Analýza chování měniče při řízení PMSM motoru*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126989>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Blaha.

## **Prohlášení**

Prohlašuji, že svou diplomovou práci na téma „Analýza chování měniče při řízení PMSM motoru“ 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **1. června 2020**

.....  
podpis autora

## **Poděkování**

Rád bych poděkoval vedoucímu diplomové práce doc. Ing. Petru Blahovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé závěrečné práce.

V Brně dne: **1. června 2020**

.....  
podpis autora

# Obsah

<b>1</b>	<b>ÚVOD</b> .....	<b>11</b>
<b>2</b>	<b>VYBRANÉ TEORETICKÉ POZNATKY</b> .....	<b>12</b>
2.1	Synchronní motor s permanentními magnety .....	12
2.2	Vektorové řízení PMSM motoru.....	13
2.3	Matematický model PMSM motoru .....	15
<b>3</b>	<b>VÝVOJOVÁ SADA S BLDC MOTOREM</b> .....	<b>17</b>
3.1	Vývojová deska AURIX Application Kit TC234.....	18
3.1.1	Překladač HighTec Free TriCore Entry Toolchain .....	19
3.1.2	Framework BIFACES a ovladače iLLD .....	20
3.1.3	AURIX Development Studio .....	20
3.2	Výkonová deska eMotor Drive Kit [8] .....	20
3.2.1	Můstkový budič TLE9180.....	21
3.3	Ukázková aplikace FOC.....	22
3.3.1	Použité periferie MCU [5] .....	22
3.3.2	Převedení aplikace do BIFACES .....	26
3.3.3	Doporučený postup při zakládání nového projektu.....	29
3.3.4	Vývojový diagram hlavní programové funkce [5].....	30
3.3.5	Implementované řízení [5].....	31
3.3.6	Ovládání aplikace .....	32
<b>4</b>	<b>VYTVOŘENÁ MĚŘICÍ APLIKACE</b> .....	<b>34</b>
4.1	Možnosti řešení zadaného problému.....	35
4.2	Popis funkce vybraných periférií v aplikaci .....	36
4.2.1	Modul GTM .....	36
4.2.2	Jednotka CCU6.....	38
4.2.3	A/D převodník VADC.....	39
4.2.4	Řadič DMA.....	41
4.3	Propojení s ukázkovou aplikací FOC .....	42
4.3.1	Zařazení do projektové struktury.....	42
4.3.2	Zdroj analogového signálu pro VADC .....	43
4.3.3	Analýza signálu PWM z podmodulu TOM .....	44
4.4	Grafické znázornění funkce aplikace .....	45
4.5	Zjednodušené vývojové diagramy .....	46
4.5.1	Zahájení měření a výpis naměřených hodnot.....	46
4.5.2	Hlavní část měřicí aplikace.....	47
4.6	Získávání časových značek.....	47
4.6.1	Přerušování od DMA kanálu .....	48
4.6.2	Vstupní podmodul TIM.....	48

4.7	Zahájení měření a zobrazení naměřených hodnot .....	49
4.8	Vybrané konfigurační parametry .....	51
<b>5</b>	<b>PRAKTICKÉ OVĚŘENÍ MĚŘICÍ APLIKACE .....</b>	<b>52</b>
5.1	Testování harmonickým signálem .....	52
5.1.1	Schéma zapojení .....	53
5.1.2	Získané výsledky .....	53
5.1.3	Zhodnocení .....	55
5.2	Testování na reálném motoru .....	55
5.2.1	Schéma zapojení .....	55
5.2.2	Získané výsledky .....	56
5.2.3	Zhodnocení .....	61
<b>6</b>	<b>MOŽNOSTI DALŠÍHO VYLEPŠENÍ MĚŘICÍ APLIKACE .....</b>	<b>62</b>
6.1	Automatizovaný přenos dat z MCU do PC .....	62
6.2	Vytvoření uživatelského prostředí na LCD displeji .....	62
6.3	Použití vyšší řady MCU .....	62
<b>7</b>	<b>ZÁVĚR .....</b>	<b>64</b>
	<b>Literatura .....</b>	<b>65</b>
	<b>Seznam příloh .....</b>	<b>67</b>



# Seznam obrázků

Obr. 1: Základní dva typy PMSM motorů – vlevo IPMSM, vpravo SPMSM [3] .....	12
Obr. 2: Příklad aplikace s PMSM motorem [4] .....	13
Obr. 3: Řídicí smyčka FOC s PMSM motorem [5] .....	14
Obr. 4: Blokové schéma pro modelování PMSM motoru [2] .....	16
Obr. 5: Zkompletovaná vývojová sada AURIX Motor Application Kit s BLDC motorem .....	17
Obr. 6: Blokové schéma aplikační sady TC234 [6] .....	18
Obr. 7: Blokové schéma MCU TC23x řady AURIX [7] .....	19
Obr. 8: Blokové schéma výkonové desky eMotor Drive Kit [8] .....	21
Obr. 9: Blokové schéma můstkového budiče TLE9180 [11] .....	22
Obr. 10: Třífázový PWM signál s okamžikem zahájení měření proudu [5] .....	23
Obr. 11: Příklad zapojení inkrementálního enkodéru k MCU TC234 [5] .....	25
Obr. 12: Zjednodušená struktura realizovaného plánovače úloh [5] .....	26
Obr. 13: Porovnání původní a nové projektové struktury aplikace .....	28
Obr. 14: Vývojový diagram hlavní programové funkce [5] .....	30
Obr. 15: Vývojový diagram funkce obsluhující přerušování VADC [5] .....	31
Obr. 16: Vývojový diagram funkce TorqueControl_FocStep [5] .....	32
Obr. 17: Hlavní ovládací obrazovka LCD displeje .....	33
Obr. 18: Blokové schéma měřicí aplikace .....	34
Obr. 19: Ilustrace uspořádání kanálů podmodulu TOM pro generování PWM [9] ..	36
Obr. 20: Vnitřní struktura kanálu 8 až 14 podmodulu TOM [7] .....	37
Obr. 21: Princip vytváření PWM signálu pomocí podmodulu TOM [9] .....	38
Obr. 22: Vnitřní struktura časovače T13 jednotky CCU6 [7] .....	39
Obr. 23: Vnitřní struktura A/D převodníku VADC [7] .....	40
Obr. 24: Blokové schéma převodníkového jádra VADC [7] .....	40
Obr. 25: Blokové schéma řadiče DMA [7] .....	42
Obr. 26: Průběh z osciloskopu, výstupy TOM1 – modrá CH3, červená CH7 .....	44
Obr. 27: Grafické znázornění funkce aplikace .....	45
Obr. 28: Zjednodušený vývojový diagram zahájení měření a výpisu naměřených hodnot .....	46
Obr. 29: Zjednodušený vývojový diagram hlavní části měřicí aplikace .....	47
Obr. 30: Vnitřní struktura kanálu podmodulu TIM [7] .....	48
Obr. 31: Příklad výstupu měřicí aplikace v programu PuTTY .....	50
Obr. 32: Multifunkční USB přístroj DIGILENT ANALOG DISCOVERY 2 .....	52
Obr. 33: Schéma zapojení ověřovacího experimentu MCU TC277 .....	53
Obr. 34: Získané grafické závislosti – první a druhé měření .....	54
Obr. 35: Schéma zapojení ověřovacího experimentu MCU TC234 .....	55

Obr. 36: Získaná grafická závislost, měření s časovými značkami – nezatížený motor při nastavené rychlosti 2500 ot/min.....	58
Obr. 37: Porovnání získaných grafických závislostí – testovací signál .....	59
Obr. 38: Získané grafické závislosti, měření bez časových značek – nezatížený/zatížený motor při nastavené rychlosti 2500 ot/min.....	60

## Seznam tabulek

Tab. 1: Vybrané parametry BLDC motoru Nanotec DB42S02 [10].....	21
Tab. 2: Pozměněný atribut uvnitř souboru <i>Config.xml</i> .....	27
Tab. 3: Projektová struktura měřicí aplikace .....	42
Tab. 4: Parametry vstupních kanálů podmodulu TIM pro získávání časových značek.....	49
Tab. 5: Vybrané konfigurační parametry měřicí aplikace.....	51
Tab. 6: Tabulka naměřených hodnot testovacího signálu s časovými značkami vztaženými k začátku PWM.....	71

# 1 ÚVOD

Pro řízení elektrických pohonů se běžně používá MCU (*Mikrokontrolér*), v němž se provádí výpočet řídicího algoritmu na základě naměřených hodnot proudů motoru. Tyto hodnoty jsou ale získávány pouze v omezeném časovém kroku a jejich informace není dále využita.

Cílem této diplomové práce je navrhnout programové řešení pro MCU řady AURIX od společnosti Infineon Technologies, které umožní vzorkování signálu s požadavkem na co nejkratší periodu vzorkování a na libovolné nastavení okamžiku zahájení měření. Toto řešení bude potřeba přidat do ukázkové aplikace vektorového řízení pro vývojovou sadu s BLDC (*Brushless DC*) motorem, aby mohlo být provedeno měření na reálném motoru.

Získané hodnoty lze dále využít například pro diagnostiku motoru, kdy by dalším zpracováním mohla být provedena analýza průběhů fázových proudů motoru, což by umožnilo například posoudit stav spínacích prvků napěťového střídače.

Text diplomové práce je dále rozdělen do pěti kapitol, přičemž první čtyři svým obsahem odpovídají bodům zadání této práce. První kapitola se zabývá vybranými teoretickými poznatky z problematiky řízení synchronních motorů s permanentními magnety. Druhá kapitola čtenáře seznámí s poskytnutou vývojovou sadou obsahující BLDC motor a především ukázkovou aplikací vektorového řízení. Ve třetí kapitole je popsáno vlastní řešení zadaného problému, které uživateli umožní provést vzorkování vstupního signálu s požadavkem na co nejkratší periodu vzorkování a na libovolné nastavení okamžiku zahájení měření. Na tuto kapitolu navazuje praktická část práce, ve které bylo ověřeno navržené řešení testováním na reálném motoru. V poslední, kratší kapitole jsou představeny možnosti, jejichž aplikováním by bylo možné vytvořenou měřicí aplikaci dále vylepšit. Na závěr jsou shrnuty dosažené výsledky.

## 2 VYBRANÉ TEORETICKÉ POZNATKY

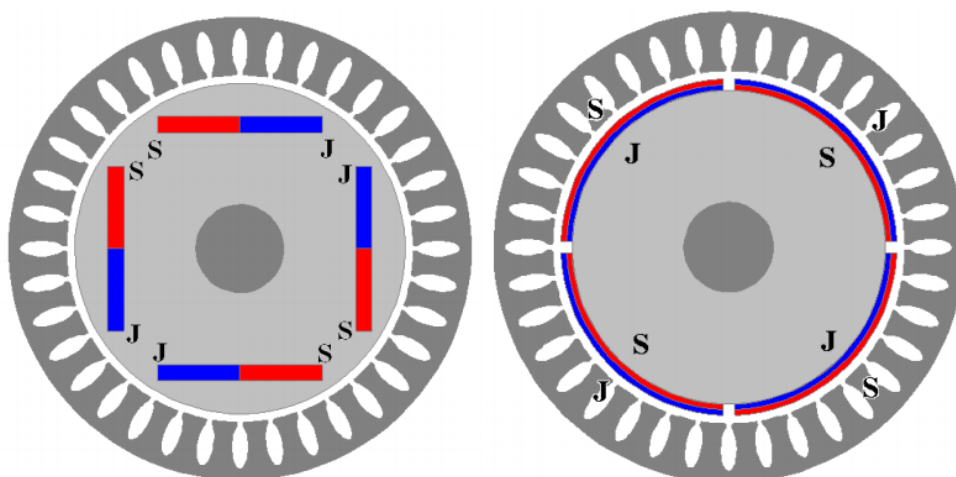
Tato kapitola je věnována prvnímu bodu zadání diplomové práce a obsahuje vybrané teoretické poznatky z problematiky řízení synchronních motorů s permanentními magnety.

### 2.1 Synchronní motor s permanentními magnety

Synchronní motor s permanentními magnety (PMSM – *Permanent Magnet Synchronous Motor*) je střídavý synchronní stroj buzený permanentními magnety. Tento elektrický pohon se hojně používá pro svůj vysoký výkon, účinnost a provozní spolehlivost především v oblasti polohových servomechanismů, tj. například v robotice, průmyslové automatizaci či automobilovém průmyslu. 0

Konstrukce tohoto motoru je podobná BLDC motoru, z čehož vyplývá, že stator motoru je tvořen zpravidla třífázovým vinutím a rotor permanentními magnety. Uspořádání vinutí statoru vytváří ve vzduchové mezeře magnetický tok se sinusovým průběhem, což připomíná indukční elektrický stroj (asynchronní motor). Zpětné elektromotorické napětí PMSM motoru má harmonický, sinusový průběh. 0

Statorové vinutí je zapojeno do hvězdy a k jeho napájení se používá napěťový střídač osazený nejčastěji IGBT tranzistory se zpětnými diodami a řízený signály PWM. Napájení statoru harmonickými proudy má za následek rovnoměrný chod motoru bez momentových pulzací. Pro řídicí algoritmus PMSM motoru je nezbytné znát co nejpřesněji informaci o jeho okamžité poloze a rychlosti. Ta se získává pomocí resolveru nebo inkrementálního snímače. [2]



Obr. 1: Základní dva typy PMSM motorů – vlevo IPMSM, vpravo SPMSM [3]

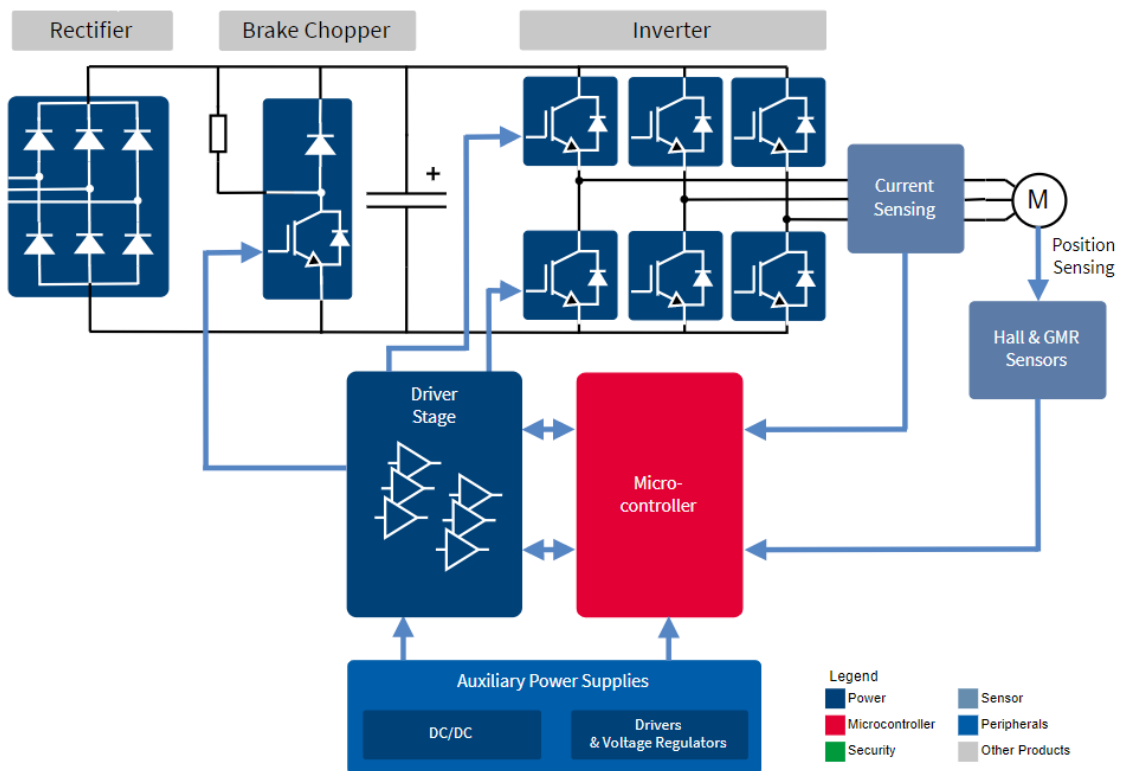
Běžně se používají dva základní typy PMSM motorů, které se liší v umístění permanentních magnetů, viz Obr. 1, což výrazným způsobem ovlivňuje chování

samotného motoru. Permanentní magnety mohou být buď rozprostřeny na povrchu rotoru ve vzduchové mezeře, pak se označují SPMSM (*Surface Permanent Magnet Synchronous Motor*), nebo naopak zabudovány uvnitř rotoru, pak se označují IPMSM (*Interior Permanent Magnet Synchronous Motor*). [3]

Mezi hlavní výhody PMSM motorů patří velký startovací moment a účinnost. Při zachování výkonu jsou jejich rozměry menší než u asynchronních motorů. Dále mají také velmi malé momentové zvlnění. [3]

Naopak mezi hlavní nevýhody lze zahrnout problémy při provozování těchto motorů v prostředí s vysokými teplotami (Curieova teplota), nutnost znalosti okamžité a počáteční polohy rotoru pro řídicí algoritmus, ale i to, že je i po vypnutí motoru stále přítomno magnetické pole. [3]

V porovnání se stejnosměrnými servomotory jsou ty s PMSM motory výhodnější především kvůli jejich malým rozměrům a momentu setrvačnosti, možnosti velkého proudového a momentového přetížení v dynamických stavech, vysoké životnosti a provozní spolehlivosti a minimálních nároků na údržbu. [2]



Obr. 2: Příklad aplikace s PMSM motorem [4]

## 2.2 Vektorové řízení PMSM motoru

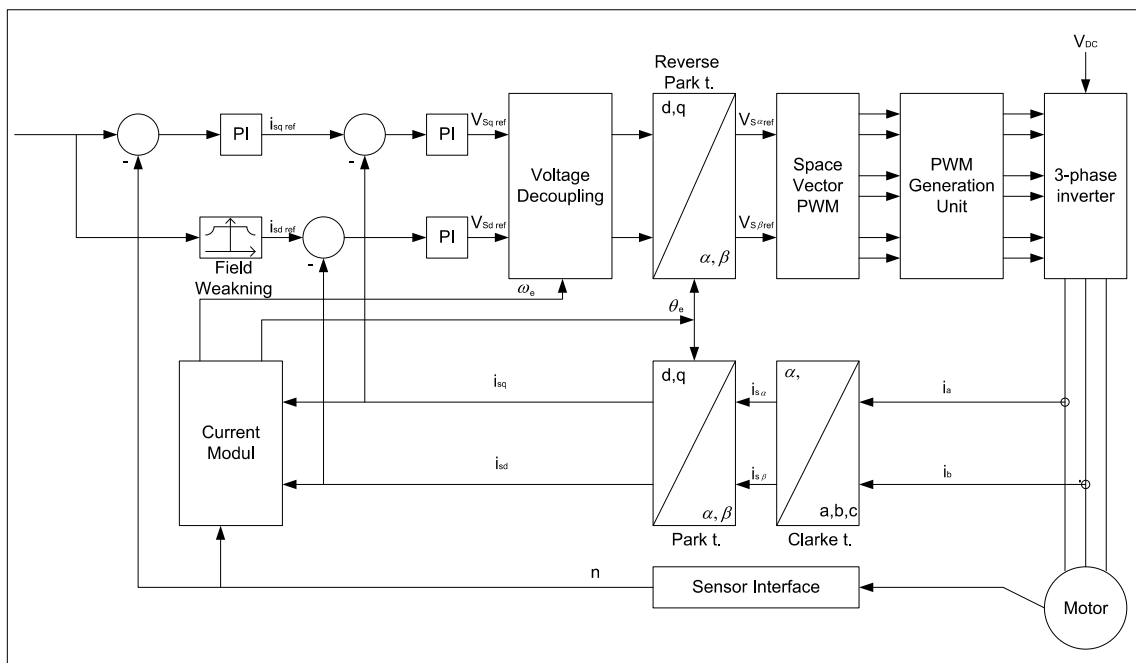
Příklad možné aplikace s PMSM motorem schematicky ilustruje Obr. 2. Jednotlivé bloky představují usměrňovač, stejnosměrný meziobvod s brzdou, třífázový

napěťový střídač, obvody pro měření proudů a polohy rotoru, MCU a můstkový budič, případně mohou být zahrnuty také další doplňkové napájecí zdroje.

Motor je napájen z třífázového tranzistorového měniče a nejčastěji se řídí v uzavřené smyčce vektorově (FOC – *Field-Oriented Control*). Hlavní myšlenkou tohoto typu řízení je rozdělení statorového proudu  $i_s$  na dvě složky: momentotvornou  $i_{sq}$  a tokotvornou  $i_{sd}$ . Každá složka je poté regulována na žádanou hodnotu proudu. [5]

Řídicí algoritmus FOC je realizován uvnitř MCU na základě snímání proudů motoru a aktuální polohy rotoru. Výstupem jsou signály PWM (*Pulse Width Modulation*) pro spínací tranzistory měniče. [5]

Jedna z možných variant zapojení řídicí smyčky FOC je uvedena na Obr. 3. Na hodnoty naměřených proudů jsou nejprve aplikovány dvě matematické transformace – Clarkové a Parkova. [5]



Obr. 3: Řídicí smyčka FOC s PMSM motorem [5]

Převod ze statorového souřadného systému  $a, b, c$  do rotorového  $d, q$  pomocí transformací je pro výpočet algoritmu výhodnější, protože je pevně spjat s rotorem, a proto se otáčí se synchronní rychlostí. [2]

První transformace se nazývá Clarkové a je určena pro přepočítání ze statorového souřadného systému popsaného pomocí tří os  $a, b, c$  do dvouosého  $\alpha, \beta$ . Matice této transformace sice obsahuje funkce *sinus* a *cosinus*, ale lze ji zjednodušit tak, že obsahuje pouze konstanty. Pro opačný přechod mezi souřadnými systémy se používá inverzní Clarkové transformace. [3]

Druhá transformace se nazývá Parkova a používá se pro zjednodušení popisu dějů v elektrických pohonech, například právě při vektorovém řízení. Dvouosý

souřadný systém  $\alpha, \beta$  spojený se státorem se přepočítá do systému  $d, q$  spojeného s rotorem. Rotorový systém je vůči statorovému posunutý ve směru otáčení rotoru o určitou hodnotu úhlu. Transformační matice obsahuje funkce *sinus* a *cosinus* právě tohoto úhlu a již ji nelze zjednodušit. Pro realizaci řídicího algoritmu jsou většinou tyto goniometrické funkce nahrazeny vyhledávací tabulkou, protože je jejich výpočet časově náročný. Pro opačný přechod mezi souřadnými systémy se používá inverzní Parkova transformace. [3]

Po výpočtu akčních zásahů regulátorů obou složek proudu v  $d, q$  souřadnicích může být provedeno odstranění křížových vazeb mezi oběma složkami proudu. Získané hodnoty jsou poté zpětně transformovány do souřadnic  $\alpha, \beta$  pomocí inverzní Parkovy transformace. Nakonec je na ně aplikována prostorově vektorová PWM, která je vstupem do jednotky pro generování PWM signálů pro třífázový měnič. [5]

Jelikož v této práci nebyl přímo realizován řídicí algoritmus FOC, nejsou zde dále rozebírány jeho matematické principy. Další informace o způsobu realizace FOC lze najít například v [3], [5].

Z výše uvedeného plyne, že MCU by měl minimálně obsahovat periferie pro generování PWM signálů, snímání fázových proudů a úhlu natočení hřídele. Dále rovněž dostatečný výpočetní výkon a dostupnou paměť pro realizaci řídicího algoritmu FOC.

## 2.3 Matematický model PMSM motoru

Odvození matematického modelu PMSM motoru v  $d, q$  souřadnicích není obsahem této práce. Dále jsou uvedeny základní rovnice matematického modelu PMSM motoru. Matematické odvození lze dohledat v [2], [3].

Maticový zápis stavových rovnic: [2]

$$p \begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \frac{\omega L_q}{L_d} \\ -\frac{\omega L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} U_d \\ U_q - \omega \Psi_f \end{bmatrix}, \quad (1)$$

kde  $p$  je Laplaceův operátor,

$I_d, I_q$  jsou složky statorového proudu,

$U_d, U_q$  jsou velikosti napětí v rotorovém systému,

$\Psi_f$  je magnetický tok generovaný permanentním magnetem,

$L_d, L_q$  jsou indukčnosti v rotorovém systému,

$R_s$  je odpor statoru,

$\omega$  je elektrická úhlová rychlost rotoru.

Pro derivaci třetí stavové proměnné, tj. mechanické úhlové rychlosti  $\Omega_m$ , platí následující rovnice: [2]

$$p \Omega_m = \frac{1}{J_c} (M_e - M_L) \quad (2)$$

$$\omega = Pp \Omega_m, \quad (3)$$

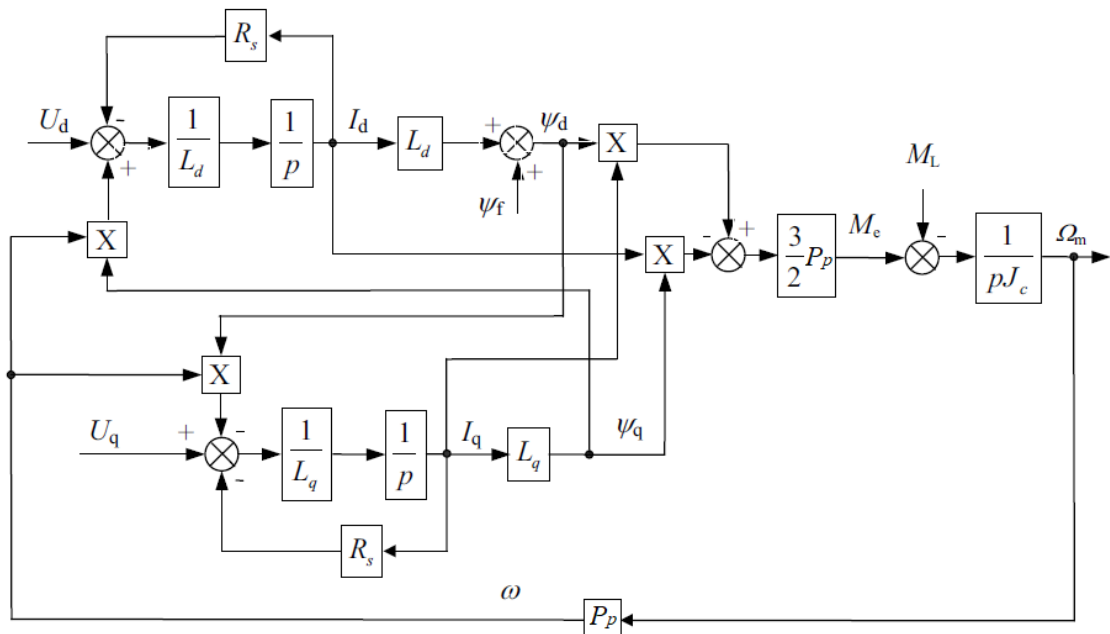
kde  $\Omega_m$  je mechanická úhlová rychlost otáčení rotoru,  
 $J_c$  je moment setrvačnosti rotoru,  
 $M_e$  je točivý moment,  
 $M_L$  je zatěžovací moment,  
 $Pp$  je počet pólových dvojic.

Pro elektromagnetický moment stroje  $M_e$  platí: [2]

$$M_e = \frac{3}{2} Pp (\Psi_d i_q - \Psi_q i_d) = \frac{3}{2} Pp (\Psi_f + (L_d - L_q) i_d) i_q, \quad (4)$$

kde  $\Psi_d, \Psi_q$  jsou spřažené magnetické toky v rotorovém systému.

Blokové schéma PMSM motoru v  $d, q$  souřadnicích vycházející z matematických rovnic (1), (2), (3) a (4) je zakresleno na Obr. 4.



Obr. 4: Blokové schéma pro modelování PMSM motoru [2]

Pozn.:  $p$  bylo upraveno na  $Pp$



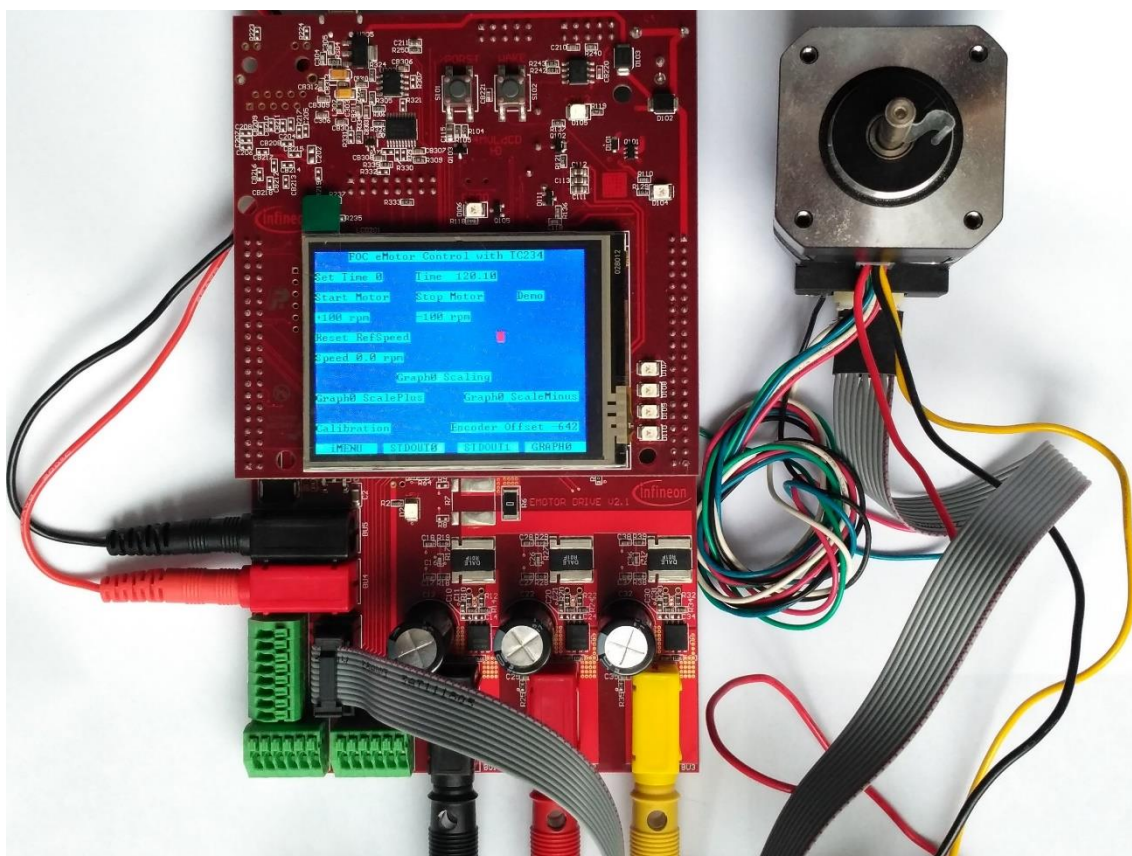
### 3 VÝVOJOVÁ SADA S BLDC MOTOREM

Tato kapitola se věnuje druhému bodu zadání diplomové práce a měla by čtenáře seznámit s použitou vývojovou sadou s BLDC motorem.

Pro účely realizace měřicí aplikace byla použita vývojová sada AURIX Motor Control Application Kit od společnosti Infineon Technologies, která se skládá z vývojové desky TC234 Application Kit, výkonové desky eMotor Drive Kit a BLDC motoru Nanotec DB42S02 s inkrementálním enkodérem Nanotec WEDL5541-B14. Na Obr. 5 je zachycena tato sada ve zkompletovaném stavu.

Jednoduchost zapojení, dostupnost materiálů a především ukázkového kódu FOC pro tuto sadu ušetřily významné množství času, které by jinak bylo potřeba věnovat vývoji podobné aplikace.

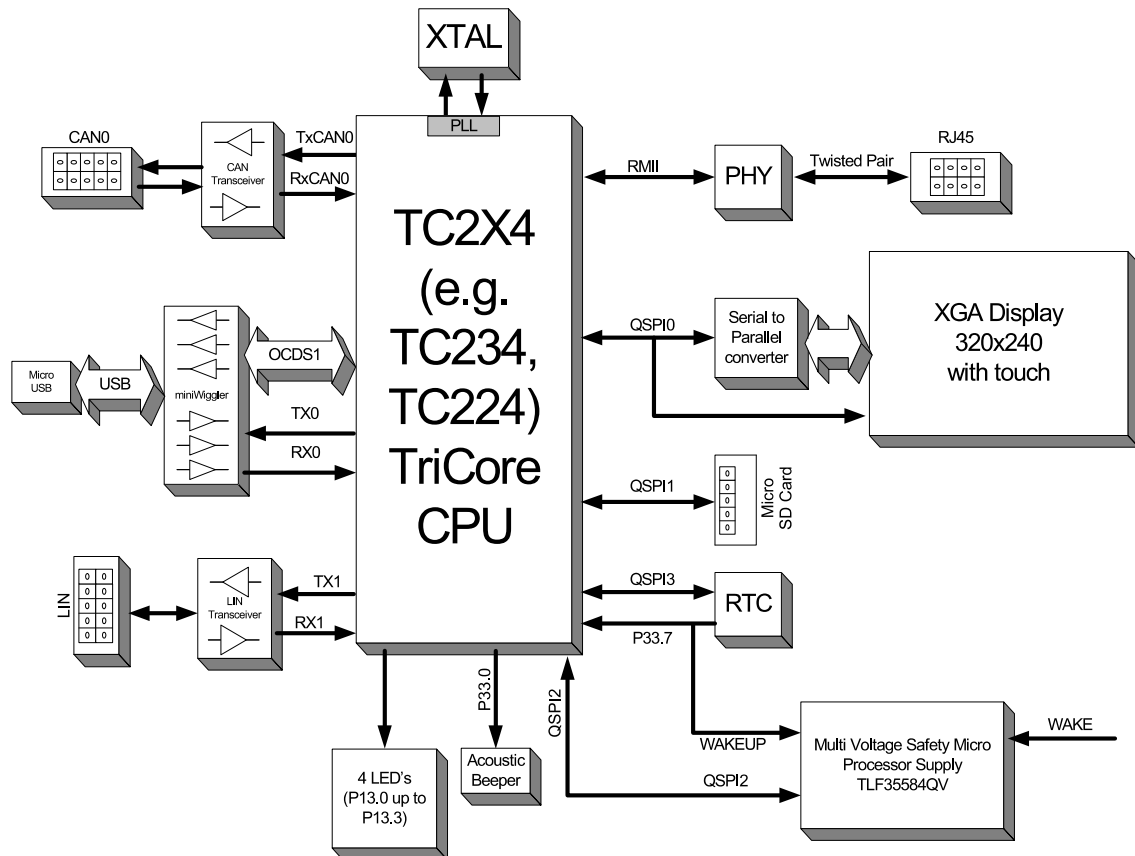
Oproti zadání obsahuje vývojová sada BLDC motor, nikoliv PMSM. Jelikož je možné oba motory řídit pomocí FOC, ve výsledku to ničemu nevádí, poněvadž uplatňované principy zůstávají stejné.



Obr. 5: Zkompletovaná vývojová sada AURIX Motor Application Kit s BLDC motorem

### 3.1 Vývojová deska AURIX Application Kit TC234

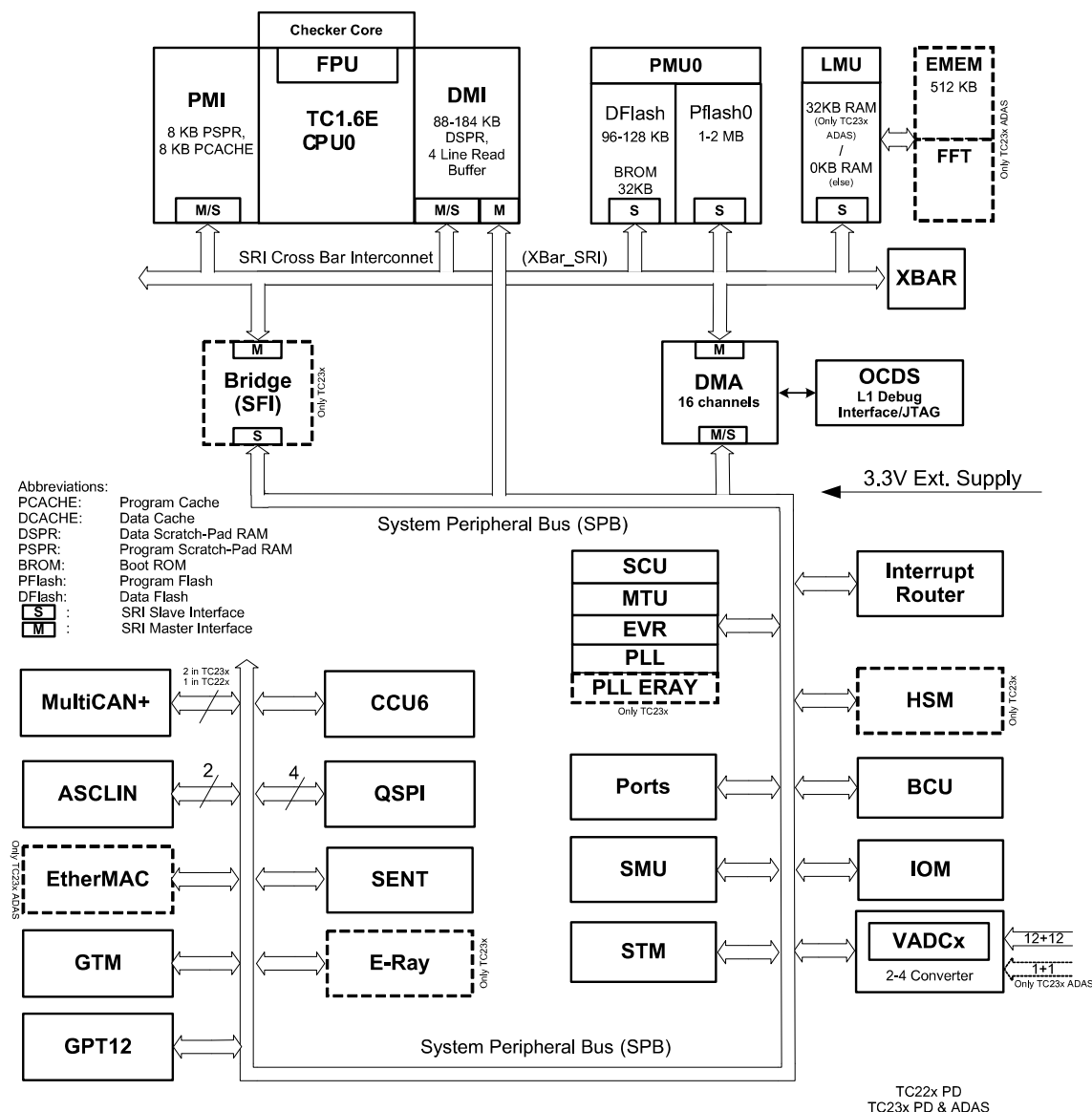
Vývojová deska AURIX Application Kit TC234 disponuje 32bitovým MCU AURIX TriCore TC234 od společnosti Infineon Technologies. Vzhledem k rozličnému množství dostupných periférií a konektorů ji lze použít pro snadný vývoj a testování různých druhů aplikací. Blokové schéma vývojové desky lze vidět na Obr. 6. [6]



Obr. 6: Blokové schéma aplikační sady TC234 [6]

Blokové schéma MCU TC23x řady AURIX zobrazuje Obr. 7. Je patrné, že tento MCU disponuje jedním 32bitovým procesorovým jádrem časovaným maximálně hodinovým signálem o frekvenci 200 MHz, různými druhy pamětí a velkým množstvím dostupných periférií, z nichž mnohé umožňují obsluhu připojených vstupně-výstupních zařízení. [7]

Vývojová deska je s PC (*Personal Computer*) propojena skrze microUSB (*Universal Serial Bus*) kabel. Při malém proudovém zatížení ji lze z USB portu také napájet, takže není potřeba používat externí napájecí zdroj.



Obr. 7: Blokové schéma MCU TC23x řady AURIX [7]

### 3.1.1 Překladač HighTec Free TriCore Entry Toolchain

Pro tvorbu, překlad a ladění C/C++ programu pro MCU řady AURIX s architekturou TriCore je dostupný překladač HighTec Free TriCore Entry Toolchain založený na vývojové platformě Eclipse. Ke své funkci využívá sadu nástrojů GCC (*GNU Compiler Collection*). Vývojové prostředí obsahující tento překladač se obecně nazývá HighTec Development Platform.

Zkušební verzi překladače pro nekomerční účely je možné bezplatně stáhnout po vyplnění formuláře na webových stránkách <https://free-entry-toolchain.hightec-rt.com/>. Na zadanou e-mailovou adresu dorazí licenční soubor, jenž umožní vývojové prostředí s překladačem používat po dobu jednoho roku.

### 3.1.2 Framework BIFACES a ovladače iLLD

Framework BIFACES (*Build and Integration Framework for Automotive Controller Embedded Software*) a na něm založené ovladače iLLD (*Infineon Low Level Drivers*) byly společností Infineon Technologies vyvinuty k usnadnění práce při programování jejich MCU. Ovladače iLLD jsou dostupné pro většinu periférií, jimiž MCU disponuje. Ve většině případů proto není nutné pracovat přímo s registry periférií MCU, nýbrž použít vytvořené funkce, jež tyto registry nastavují. [9]

Pro jednotlivé iLLD ovladače jsou dostupné ukázkové kódy, ve kterých je zprovozněna jednoduchá aplikace s vybranou periférií. Demonstrační kód je vždy tvořen inicializační funkcí pro nastavení periferie a funkcí pro ukázkou její naprogramované činnosti, která je volána z nekonečné smyčky programu.

Framework ani ovladače nejsou bohužel integrovány uvnitř překladače HighTec Free Entry Toolchain. Před jejich použitím je nutné je stáhnout ze stránek výrobce, kde jsou ale k dispozici pouze po registraci a získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.

### 3.1.3 AURIX Development Studio

Skupina vývojářů společnosti Infineon na podzim roku 2019 představila bezplatné vývojové prostředí AURIX Development Studio založené na vývojové platformě Eclipse. Na rozdíl od překladače společnosti HighTec používá sadu nástrojů TASKING a již v sobě zahrnuje také iLLD ovladače. Jsou k němu také přidruženy různé ukázkové kódy, které jsou řádně zdokumentované.

V případě zájmu získat více informací o tomto prostředí je vhodné navštívit webové stránky výrobce věnované tomuto produktu <https://www.infineon.com/cms/en/product/promopages/aurix-development-studio/>.

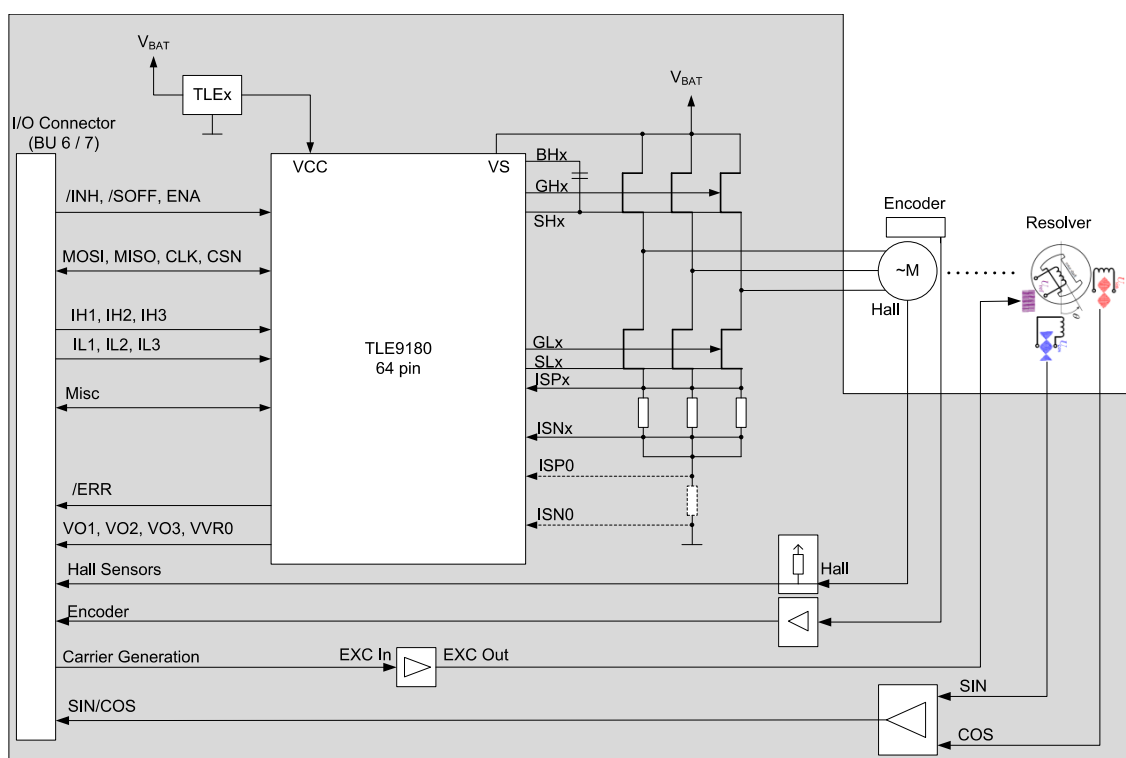
## 3.2 Výkonová deska eMotor Drive Kit [8]

Výkonová deska eMotor Drive Kit, viz blokové schéma na Obr. 8, je napájena ze spínaného napájecího zdroje 12 V a je na ní osazen budič TLE9180 pro třífázový můstek sestavený z trojice dvojitých N-kanálových výkonových tranzistorů MOSFET IPG20N04S4-08A. K tomuto můstku je připojen BLDC motor Nanotec DB42S02 s parametry uvedenými v Tab. 1. Jako snímač úhlu natočení hřídele motoru je použit inkrementální enkodér Nanotec WEDL5541-B14 poskytující rozlišení 1000 pulzů na jednu otáčku rotoru.

Propojení s vývojovou deskou je jednoduché, a to prostým spojením dvou párů kolíkových a dutinkových lišt, jejichž signálový význam lze nalézt v Příloha 2.

Tab. 1: Vybrané parametry BLDC motoru Nanotec DB42S02 [10]

<b>Způsob zapojení</b>	trojúhelník	<b>Indukčnost statoru</b>	0,245 mH
<b>Počet pólových dvojic/fází</b>	4/3	<b>Točivý moment jmenovitý/špičkový</b>	0,05/0,15 N·m
<b>Jmenovité napětí</b>	17 V	<b>Jmenovitý výkon</b>	42 W
<b>Proud bez zátěže/jmenovitý/špičkový</b>	0,6/3,57/10,8 A	<b>Rychlost jmenovitá/bez zátěže</b>	8000/12000 ot/min
<b>Odpor statoru</b>	0,19 Ω	<b>Moment setrvačnosti rotoru</b>	$2,4 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$



Obr. 8: Blokové schéma výkonové desky eMotor Drive Kit [8]

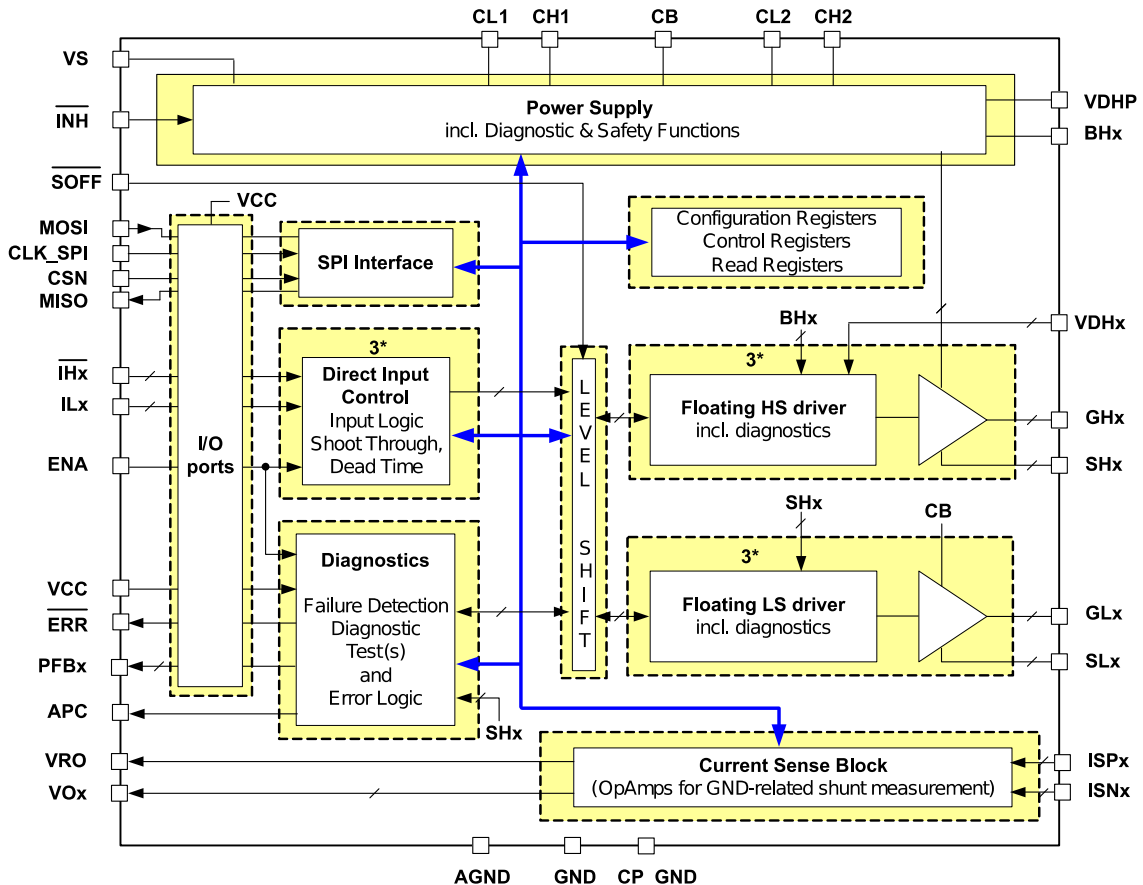
### 3.2.1 Můstkový budič TLE9180

Blokové schéma integrovaného obvodu TLE9180 je znázorněno na Obr. 9. Hlavními bloky jsou budiče hradel pro horní a dolní tranzistory větve můstku, obvody pro vnitřní napájení, napěťové přizpůsobení, sériové komunikační rozhraní QSPI, vstupně-výstupní rozhraní, diagnostiku a měření proudů pomocí nízkoodporových rezistorů, tzv. shunt rezistory, kdy je měřený úbytek napětí na známém odporu převeden na proud.

Můstkový budič je s MCU propojen přes vstupně-výstupní rozhraní. Po zapnutí je nutné jej nejdříve nakonfigurovat, a to ze strany MCU prostřednictvím

sériového komunikačního rozhraní QSPI. V případě potřeby jím lze také vyčíst dostupné diagnostické údaje.

Význam jednotlivých signálů a další informace je možné vyhledat v dokumentaci [11].



Obr. 9: Blokové schéma můstkového budiče TLE9180 [11]

### 3.3 Ukázková aplikace FOC

Na stránkách výrobce lze získat zdrojový kód ukázkové aplikace FOC určenou pro tuto vývojovou sadu. Avšak pouze za předpokladu registrace a následného získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.

Nevýhodou poskytované aplikace je, že byla vytvořena s využitím frameworku SWFramework, který předcházel frameworku BIFACES. S tím souvisí i stáří použitých iLLD ovladačů. Novější verze v době vypracování této práce nebyla k dispozici.

#### 3.3.1 Použité periferie MCU [5]

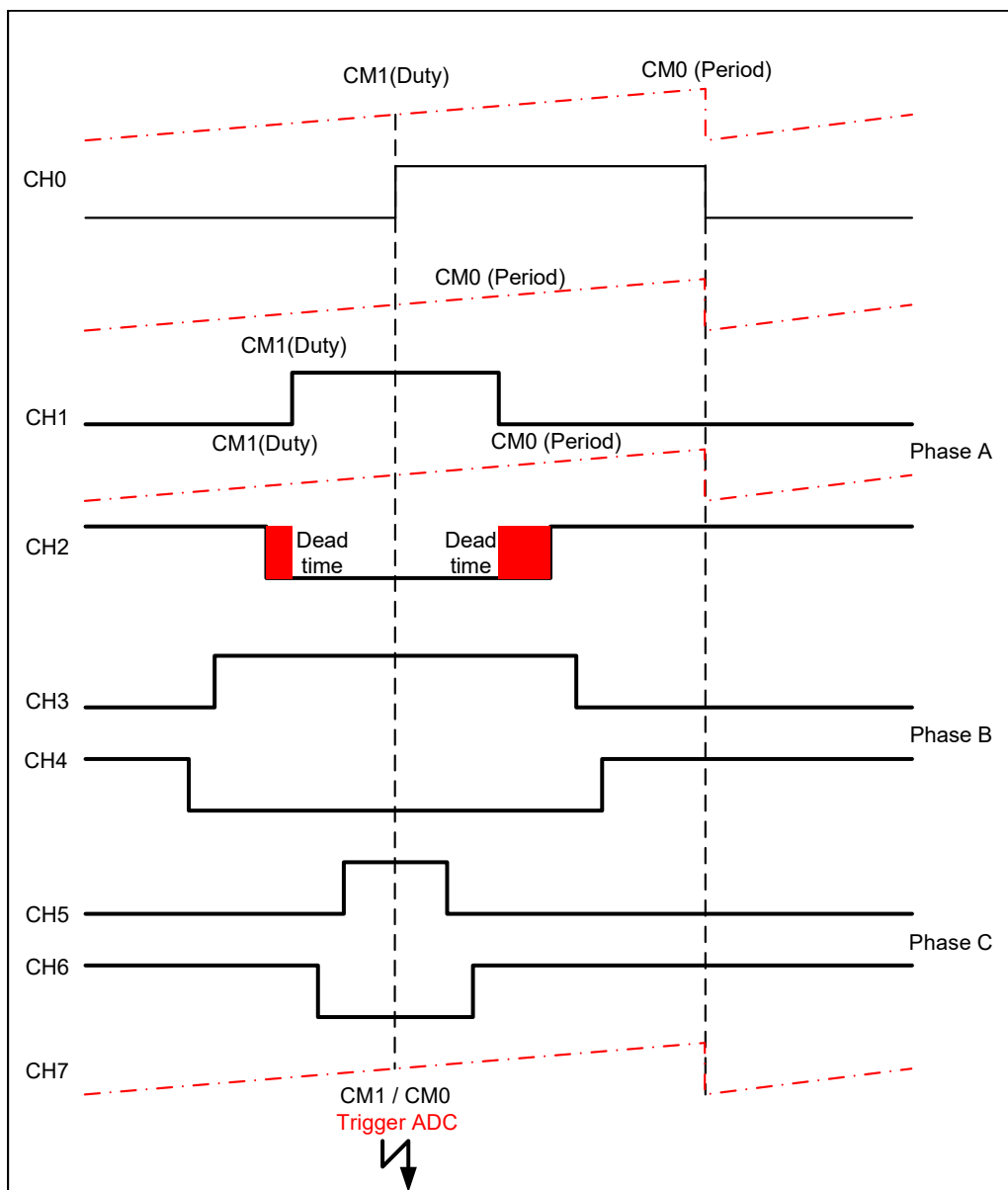
Ukázková aplikace pracuje s následujícími periferiemi MCU TC234:

- modulem GTM (*Generic Timer Module*),

- A/D (analogově-digitální) převodníkem s postupnou aproximací VADC (Versatile Analog-to-Digital Converter),
- jednotkou GPT12 (General Purpose Timer Unit),
- rozhraním QSPI (Queued Synchronous Peripheral Interface),
- systémovým časovačem STM (System Timer).

### 3.3.1.1 Modul GTM

Prostřednictvím výstupního podmodulu TOM (Timer Output Module) periferie GTM se generuje třífázový PWM signál pro jednotlivé spínací tranzistory měniče. MCU TC234 obsahuje celkem dva podmoduly TOM, jež jsou označeny TOM0 a TOM1.



Obr. 10: Třífázový PWM signál s okamžikem zahájení měření proudu [5]

K tomuto účelu je v aplikaci vyhrazeno celkem osm kanálů podmodulu TOM1. První, CH0, je použit jako základní, druhý až šestý, CH1 až CH6, pro každou dvojici, horních a dolních, spínacích tranzistorů a poslední, CH7, určující okamžik zahájení měření proudů pomocí A/D převodníku.

Kmitočet výstupního signálu CH0 je 20 kHz a vytvářené průběhy PWM jsou zarovnané na střed. Parametry dílčích PWM signálů jsou tedy aktualizovány každých 50  $\mu$ s, jakmile jsou k dispozici naměřené hodnoty proudů. Mrtvá doba mezi spínacími cykly tranzistorů je řešena programově.

Příklad možného průběhu třífázového signálu PWM z podmodulu TOM zobrazuje

Obr. 10. Je zde také zakreslen okamžik sepnutí všech dolních tranzistorů, při němž je zahájeno měření proudů.

### **3.3.1.2 A/D převodník s postupnou aproximací VADC**

MCU TC234 disponuje dvěma A/D převodníky s postupnou aproximací VADC, jež jsou rozděleny do skupin Group0 a Group1. Každá skupina obsahuje 12 kanálů připojitelných na vstupní analogové signály. Oba převodníky se používají pro současné měření úbytků napětí na shunt rezistorech zapojených mezi dolní větvi spínacích tranzistorů měniče a zemí. Aplikací Ohmova zákona se z naměřených napětí poté vypočítají fázové proudy.

Převodník skupiny Group0 získává proud první větvi společně s referenční hodnotou napětí rozdílového zesilovače uvnitř můstkového budiče TLE9180. Druhý převodník skupiny Group1 měří proud druhou větvi. Měřené veličiny jsou přivedeny na kanály CH0 (proudy, resp. úbytky napětí na odporech) obou skupin a CH1 (referenční hodnota) pouze skupiny Group0.

Převody analogových veličin jsou navrženy tak, aby se spouštěly současně v okamžiku daném výstupním signálem CH7 podmodulu TOM. Vzhledem k tomu, že jejich zdroje jsou fronty čítající až osm kanálů, lze zajistit opakování měření proudů celkem čtyřikrát. Po dokončení převodu skupiny Group0, tj. čtyři hodnoty proudu a referenční hodnota napětí rozdílového zesilovače, vzniká požadavek na přerušování, ve kterém jsou naměřené hodnoty dále zpracovány. Ze získaných proudů se tak dopočítá proud zbývající větvi, který nelze současně měřit. Platí totiž, že jejich součet je nulový.

### **3.3.1.3 Jednotka GPT12**

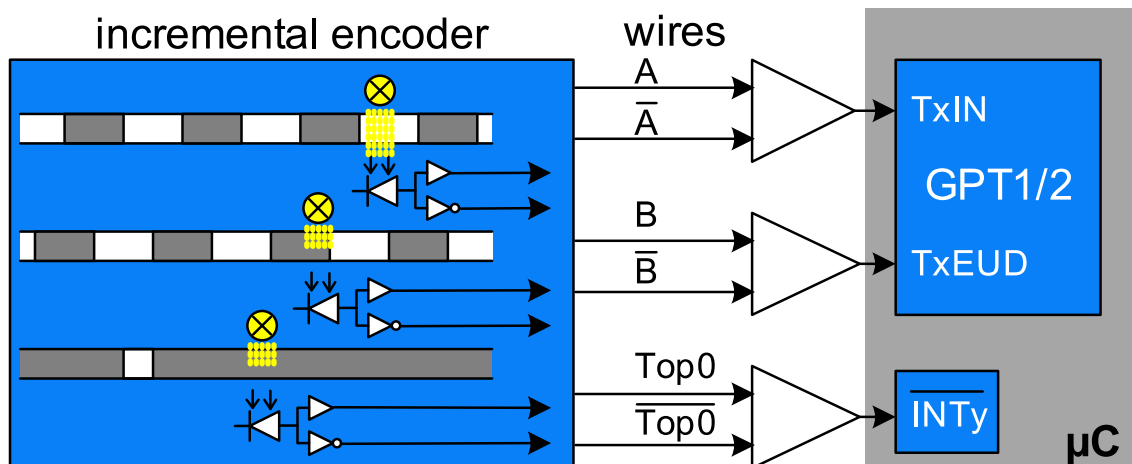
Zpracování dat z inkrementálního enkodéru pomocí jednotky GPT12 je výhodné, protože má připravené rozhraní pro tyto účely. Příklad možného zapojení inkrementálního enkodéru k MCU TC234 zachycuje Obr. 11.

Konfigurace této jednotky uvnitř aplikace určuje časovač T3 jako primární, kdy pracuje v inkrementální režimu, tzn. že čítá obě hrany vstupního signálu TxIN. Rozlišení enkodéru se tím dvakrát zvětší. Průchod nulovou pozicí se vyhodnocuje



pomocí časovače T4. Touto událostí se nuluje časovač T3 a v obslužné funkci přerušení se mohou počítat otáčky.

Rychlost otáčení motoru se počítá dvěma způsoby, jejichž volba je dána nastavitelným prahem. Při nízkých otáčkách se stanoví ze znalosti obsahu volně běžícího časovače T5, který je zachycen do příslušného registru v okamžiku náběžné hrany signálu TxIN či TxEUD. V opačném případě je pro výpočet rozhodující rozdíl hodnot časovače T3.



Obr. 11: Příklad zapojení inkrementálního enkodéru k MCU TC234 [5]

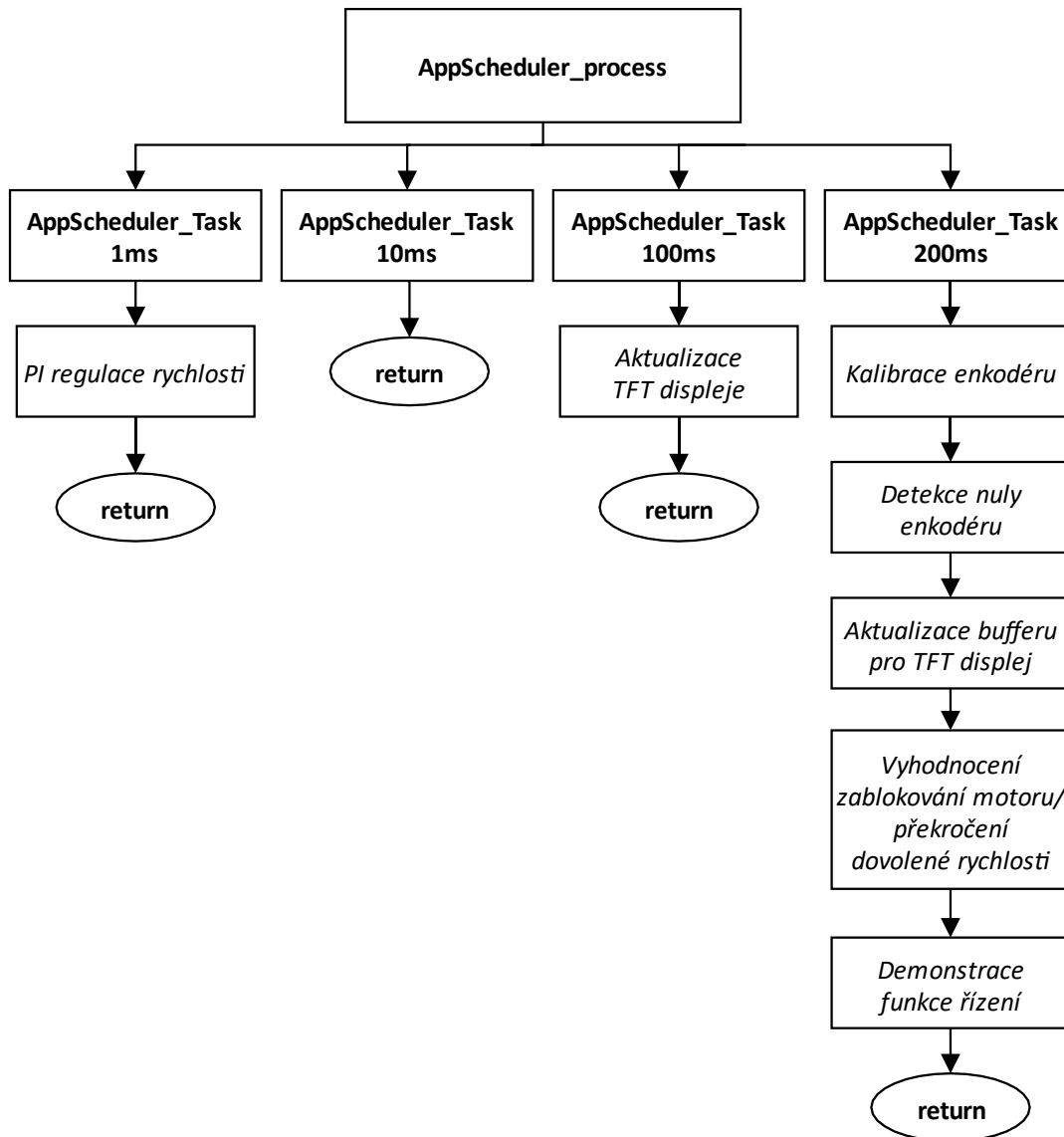
### 3.3.1.4 Rozhraní QSPI

Periferie QSPI MCU TC234 poskytuje celkem čtyři sériová komunikační rozhraní, z nichž aplikace potřebuje právě tři pro komunikaci s těmito integrovanými obvody:

- QSPI0 – řadič LCD displeje na vývojové desce,
- QSPI1 – nepoužívá se,
- QSPI2 – bezpečnostní napájecí mikročip TLF35584 na vývojové desce,
- QSPI3 – můstkový budič TLE9180 na výkonové desce.

### 3.3.1.5 Systémový časovač STM

Na přerušení systémového časovače STM je založen jednoduchý plánovač úloh, kterým jsou spouštěny čtyři úlohy s časovým okamžikem 1 ms, 10 ms, 100 ms a 200 ms. Obsluhované činnosti jednotlivými úlohami jsou uvedeny na Obr. 12.



Obr. 12: Zjednodušená struktura realizovaného plánovače úloh [5]

### 3.3.2 Převedení aplikace do BIFACES

Jak již bylo zmíněno, zdrojový kód aplikace byl vytvořen pomocí staršího frameworku SWFramework. Při používání frameworku BIFACES není v současné verzi problém tento kód přeložit, ale nelze použít novou strukturu projektu a nejnovější verzi ovladačů iLLD. Tyto důvody vedly k nutnosti převedení aplikace do nového frameworku.

Výčet vybraných změn ve zdrojovém kódu, jež byly provedeny:

- rozdíly v celkové struktuře projektu aplikace,
- přizpůsobení nejnovějším iLLD ovladačům:
  - odlišné pojmenování některých funkcí, proměnných, maker, definicí apod.,
  - náhrada rozhraní PosIf (*Position Interface*) iLLD ovladači pro periférii GPT12 – sběr dat z enkodéru,
  - funkce, pomocí nichž je generován PWM signál, nahrazeny iLLD ovladači pro periférii GTM,
  - rozhraní SSC (*Synchronous Serial Communication Interface*) nahrazeno iLLD ovladači pro periférii QSPI,
  - úprava programového kódu periférie VADC – datové struktury a inicializační funkce jsou v nové verzi iLLD ovladačů mírně odlišné,
- odstranění připravených funkcí pro získávání polohy z Hallových snímačů – modul GTM MCU TC234 nedisponuje jednotkou SPE (*Signal Pattern Evaluation*) použitou pro vyhodnocení těchto signálů,
- minimalizace počtu varování vzniklých při překladu zdrojového kódu.

Dříve než se provedl kompletní převod ukázkové aplikace do nového frameworku, předcházelo tomu její zprovoznění za předpokladu zachování projektové struktury původního frameworku. Získal se tím nástroj pro zpětnou kontrolu a ověření jejího chování v obou frameworkcích.

Rozdílnost v projektových strukturách se projevila během překladu programového kódu. Hlavičkové soubory, vkládané do zdrojových souborů přes direktivy *#include*, překladač nemohl nalézt. Řešení vedlo na drobnou úpravou konfiguračního souboru *Config.xml*, který lze nalézt uvnitř adresáře `<projekt>\1_ToolEnv\0_Build\1_Config`. Jednalo se pouze o přidání symbolu hvězdičky do atributu *internalPaths* elementu *specificInclude*, viz Tab. 2, čímž se zajistí úplné procházení projektové struktury. To sice zpomalí výslednou dobu překladu, avšak vyřeší tento problém bez nutnosti náhrady jednotlivých cest k hlavičkovým souborům.

**Tab. 2: Pozměněný atribut uvnitř souboru *Config.xml***

<b>Žlutě zvýrazněná změna v hodnotě atributu <i>internalPaths</i></b>
<pre>&lt;specificInclude internalPaths= „*/Sfr/*, */BaseSw/*/Tricore,                                 */BaseSw/*/Platform,                                 */BaseSw/*/CpuGeneric,*” /&gt;</pre>

Novější verze iLLD ovladačů byla příčinou nefunkčnosti aplikačního kódu pro počáteční nastavení a obsluhu periférií GPT12, GTM a částečně VADC. V prvních

dvou případech byl původní kód přímo nahrazen demonstračními kódy pro iLLD ovladače těchto periférií, jejichž konfigurace byla nastavena podle původního řešení. S tím souvisely i úpravy v ostatních funkcích, kde se s těmito perifériemi pracuje. V posledním případě se jednalo o odlišnosti dané pozměněnou datovou strukturou, ale i inicializační funkcí pro tuto periférii.

Poněvadž komunikace s můstkovým budičem TLE9180 nebyla realizována iLLD ovladači dostupnými pro periférii QSPI, postupovalo se při záměně obdobně, jak bylo popsáno v předchozím odstavci. Cílem bylo sjednocení přístupu k jednotlivým perifériím, tzn. využít nové iLLD ovladače tam, kde to bylo jednoduše proveditelné.

Původní projektová struktura <b>SWFramework</b>	Nová projektová struktura <b>BIFACES</b>
<ul style="list-style-type: none"> <li>▼ BaseFramework_Tc27xB_eMotor           <ul style="list-style-type: none"> <li>▼ 0_Src               <ul style="list-style-type: none"> <li>▼ 0_AppSw                   <ul style="list-style-type: none"> <li>▼ Config                       <ul style="list-style-type: none"> <li>&gt; Common</li> </ul> </li> <li>▼ Tricore                       <ul style="list-style-type: none"> <li>&gt; DemoApp</li> <li>&gt; Display</li> <li>&gt; Main</li> </ul> </li> </ul> </li> <li>▼ 1_SrvSw                   <ul style="list-style-type: none"> <li>&gt; _Utilities</li> <li>&gt; If</li> <li>&gt; SysSe</li> <li>&gt; Tricore                       <ul style="list-style-type: none"> <li>info.dox</li> </ul> </li> <li>2_CDrv</li> <li>&gt; 4_McHal</li> </ul> </li> <li>&gt; 1_ToolEnv</li> <li>&gt; 2_Out</li> <li>&gt; cfg</li> <li>Makefile</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▼ BIFACES_TC234A_eMotor_1v31           <ul style="list-style-type: none"> <li>▼ 0_Src               <ul style="list-style-type: none"> <li>▼ AppSw                   <ul style="list-style-type: none"> <li>▼ Config                       <ul style="list-style-type: none"> <li>&gt; Common</li> </ul> </li> <li>▼ Tricore                       <ul style="list-style-type: none"> <li>&gt; DemoApp</li> <li>&gt; Display</li> <li>▼ FromOldProject                       <ul style="list-style-type: none"> <li>&gt; _Lib</li> <li>&gt; If</li> <li>&gt; SPIDriver</li> <li>&gt; SysSe</li> <li>&gt; Main</li> </ul> </li> </ul> </li> <li>▼ BaseSw                   <ul style="list-style-type: none"> <li>&gt; iLLD</li> <li>&gt; Infra</li> <li>&gt; Service</li> </ul> </li> <li>&gt; 1_ToolEnv</li> <li>&gt; 2_Out</li> <li>Makefile</li> </ul> </li> </ul> </li> </ul> </li></ul>

**Obr. 13: Porovnání původní a nové projektové struktury aplikace**

Na Obr. 13 jsou uvedeny pro porovnání projektové struktury ukázkové aplikace FOC v obou frameworkcích. Je patrné, že jsou jejich struktury mírně odlišné. Liší se především v obsahu uvnitř složek *0\_Src* a *1\_ToolEnv*. Ve podsložce *0\_AppSw*, resp. *AppSw*, se nachází uživatelem vytvořený aplikační kód. Podsložky *1\_SrvSw*, *2\_CDrv* a *4\_McHal* ve starším frameworku korespondují s podsložkou *BaseSw* v novém frameworku. Kvůli zachování funkce aplikace po převedení do

nového frameworku byly chybějící soubory ze staršího frameworku, nakopírovány do složky *FromOldProject*.

### 3.3.3 Doporučený postup při zakládání nového projektu

V případě potřeby založení nového projektu v prostředí HighTec Development Platform, do kterého bude následně vložen programový kód ukázkové aplikace FOC, je vhodné se řídit tímto doporučeným postupem:

1. Spustit vývojové prostředí HighTec Development Platform přes dávkový soubor *StartBifaces.bat* nacházející se v místě instalace frameworku BIFACES, například *C:\Tools\BifacesWin64*. Do prostředí WINDOWS se tímto krokem zavedou příslušné proměnné potřebné pro správnou funkci frameworku BIFACES.

*Pozn.: Před prvním spuštěním je nutné v dávkovém souboru upravit cestu ke spouštěcímu souboru vývojového prostředí HighTec Development Platform, jež je následně nastavena do proměnné MAKECMD\_WITH\_APP, například C:\HighTec\ide\htc-ide-v2.2.4\htc-ide.exe.*

Uvnitř vývojového prostředí následně:

2. Vytvořit nový projekt založený na šabloně projektové struktury pro framework BIFACES odpovídající použitému typu MCU, například MCU TC234 je určena šablona pojmenovaná *BaseFramework\_TC23A*.

*Pozn.: Šablony projektové struktury pro framework BIFACES lze získat pouze po registraci a získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.*

3. Nahradit obsah adresáře *AppSw* programovým kódem ukázkové aplikace FOC umístěným ve složce se shodným názvem. Cesta k tomuto adresáři může být například ve vytvořeném projektu podle šablony *BaseFramework\_TC23A\0\_Src\AppSw*.
4. Nahradit obsah adresáře *BaseSw* soubory s iLLD ovladači použitého typu MCU umístěnými ve složce se shodným názvem. Cesta k tomuto adresáři může být například ve vytvořeném projektu podle šablony *BaseFramework\_TC23A\0\_Src\BaseSw*.
5. Provést úpravu atributu uvnitř souboru *Config.xml*, viz Tab. 2, nacházejícím se například ve vytvořeném projektu podle šablony uvnitř složky *BaseFramework\_TC23A\1\_ToolEnv\0\_Build\1\_Config*.

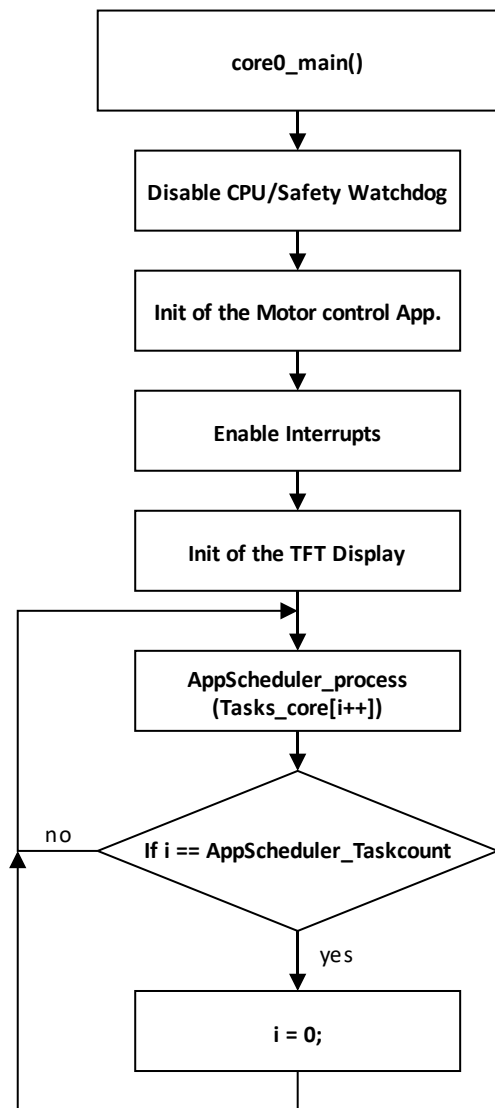
6. Zapsat v konfiguračním souboru pro překlad *Config\_Gnuc.mk* do proměnné *B\_GNUC\_TRICORE\_PATH* správnou cestu k sadě nástrojů překladače HighTec, která může být například *C:\HighTec\toolchains\tricore\v4.9.3.0-infineon-1.0*. Tento soubor je umístěn v podadresáři *Config\_Tricore\_Gnuc* adresáře z předchozího kroku.

Nakonec volitelně:

7. Vytvořit konfiguraci pro spuštění ladicího nástroje.
8. Spustit ladicí nástroj s konfigurací vytvořenou v předchozím kroku. Měl by být zahájen překlad a po jeho úspěšném provedení nabídnuta možnost nahrát přeložený program do paměti MCU.

### 3.3.4 Vývojový diagram hlavní programové funkce [5]

Ukázková aplikace je založena především na přerušeních. Po zavolání funkce *core0\_main*, viz vývojový diagram na Obr. 14, se nejprve vyřadí z činnosti watchdogy. Dále jsou vykonány všechny inicializační funkce potřebné pro realizaci řídicí aplikace (periferie, algoritmus), jsou povolena přerušení a je inicializován LCD displej. Nakonec se program dostane do nekonečné smyčky, v níž jsou postupně zpracovávány úlohy implementované uvnitř jednoduchého plánovače úloh voláním funkce *AppScheduler\_process*. Obsah jednotlivých úloh byl součástí Obr. 12.



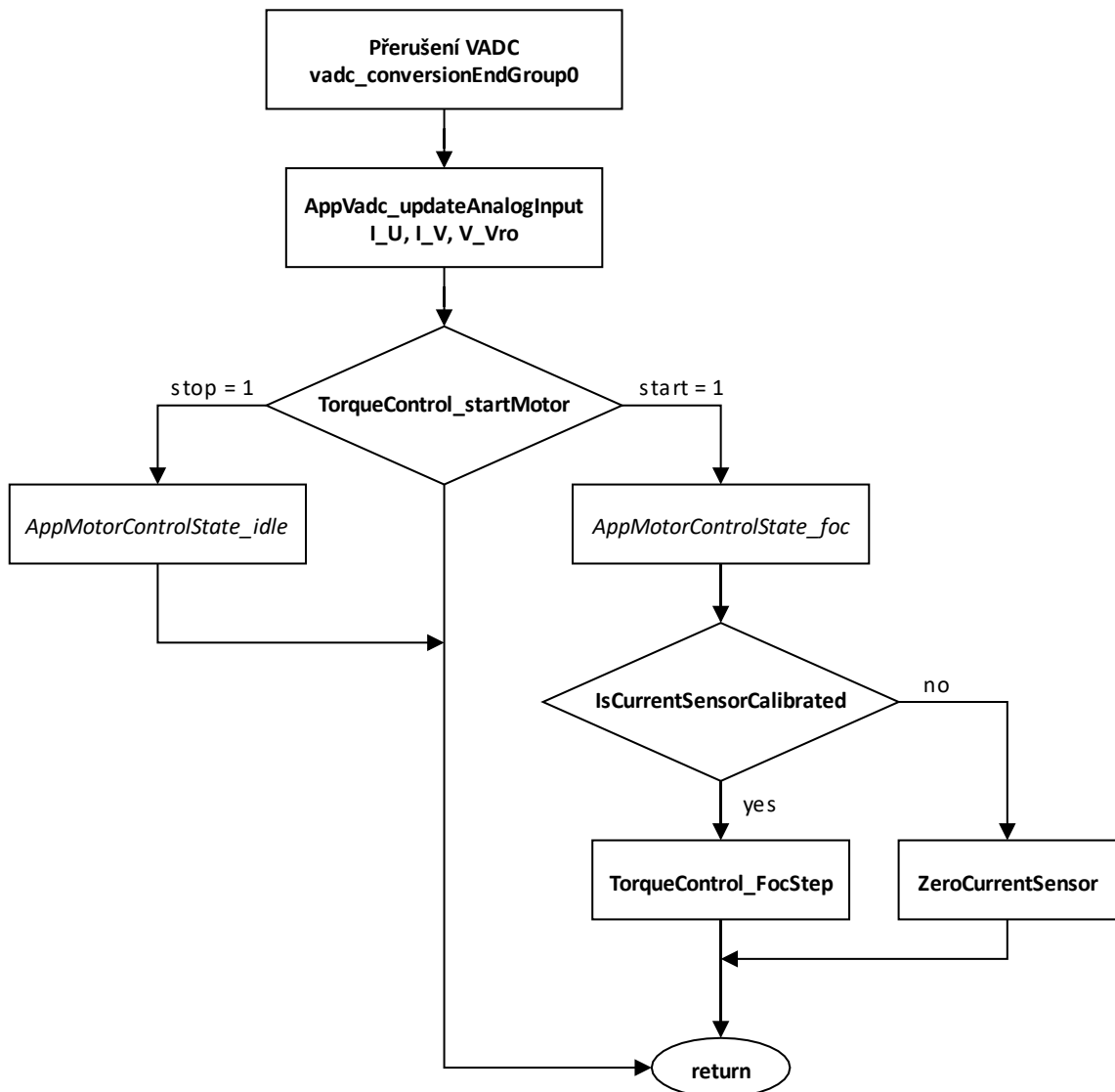
Obr. 14: Vývojový diagram hlavní programové funkce [5]

### 3.3.5 Implementované řízení [5]

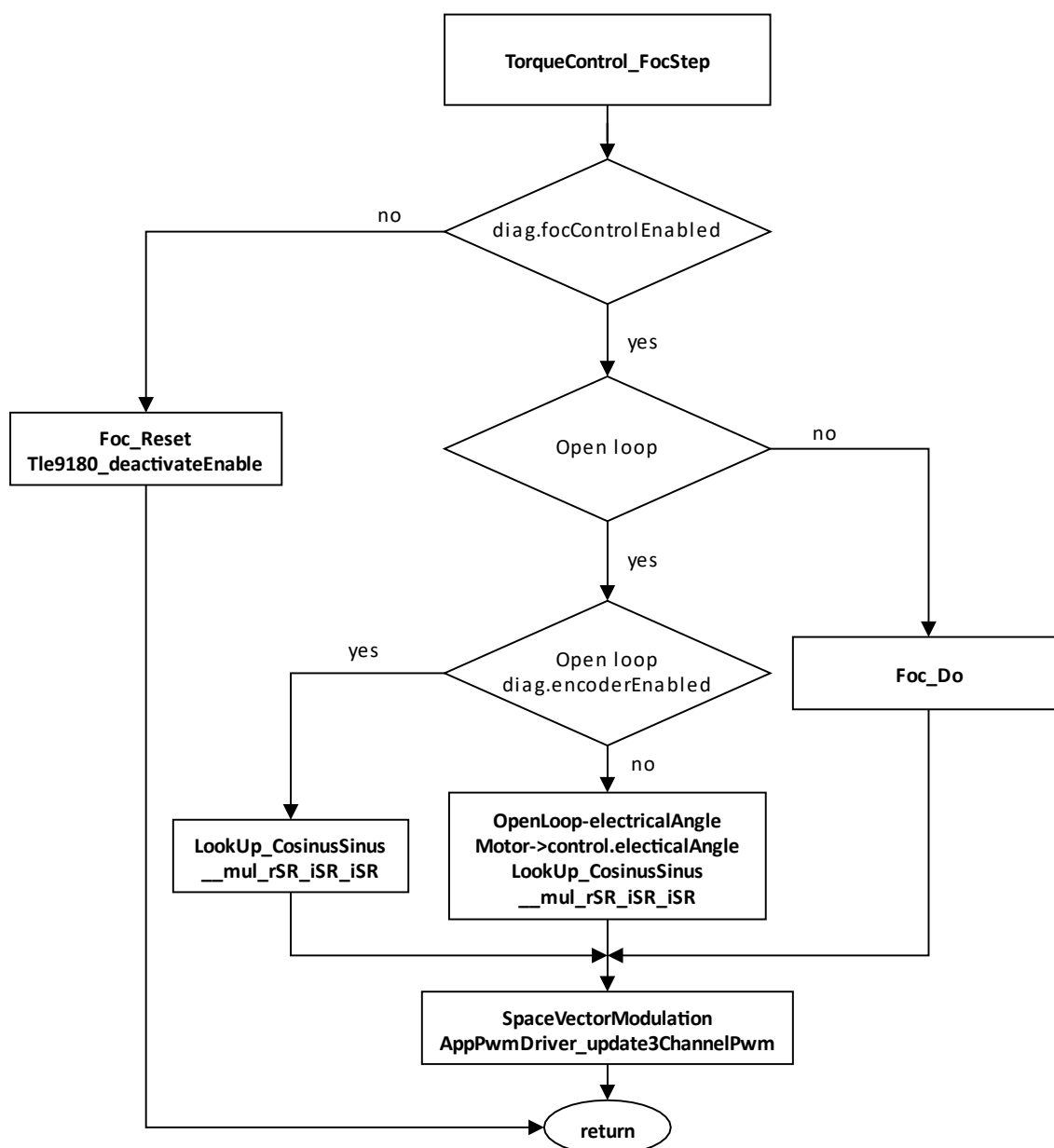
Aplikace obsahuje celkem tři regulátory – otáčkový regulátor a dva regulátory jednotlivých složek statorového proudu při výpočtu FOC. Všechny tři regulátory jsou proporciálně sumáčního (PS) typu.

#### 3.3.5.1 Algoritmus FOC

Jakmile jsou získány hodnoty jednotlivých proudů, což je vždy po uplynutí 50  $\mu$ s, probíhá výpočet algoritmu FOC, viz odpovídající vývojové diagramy zobrazené na Obr. 15 a Obr. 16.



Obr. 15: Vývojový diagram funkce obsluhující přerušení VADC [5]



Obr. 16: Vývojový diagram funkce TorqueControl\_FocStep [5]

### 3.3.5.2 Regulace otáček

Každou 1 ms se získá aktuální rychlost z enkodéru v rad/s a následně se přepočítá do ot/min. Je-li povoleno řízení rychlosti motoru, provede se výpočet akčního zásahu otáčkového PS regulátoru na základě odchylky rychlosti od její žádané hodnoty. Výstupem z regulátoru je tak nová žádaná hodnota momentotvorné složky  $i_q$ .

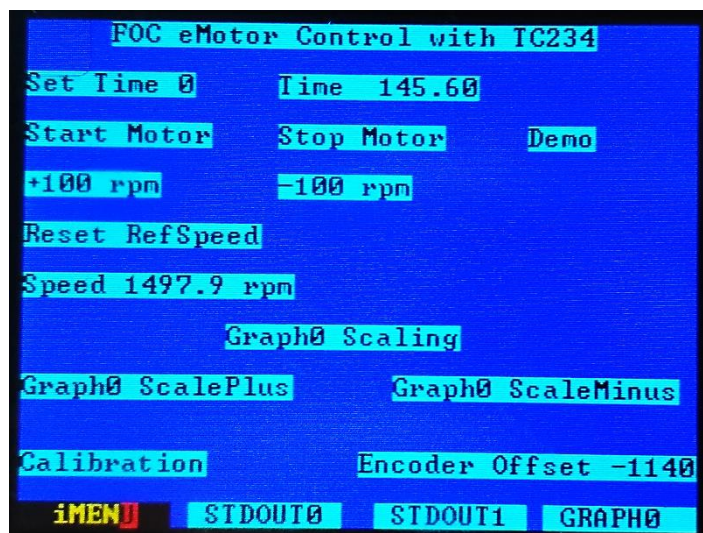
### 3.3.6 Ovládání aplikace

Aplikaci lze ovládat buď přímo z LCD displeje pomocí příslušných tlačítek, viz Obr. 17, anebo v programovacím prostředí prostřednictvím debuggeru nastavením odpovídajících proměnných uvnitř globálních datových struktur *g\_motor.control*



(proměnné *.start/.stop* – spuštění/zastavení motoru, *.calon* – kalibrace enkodéru) a *g\_velocityControl* (proměnná *.ref* – volba žádané rychlosti).

Po připojení napájení je nutné provést kalibraci enkodéru. Kompenzuje se tím poloha mechanické nuly vůči elektrické. Bez provedení této kalibrace nemusí být pokus o roztočení motoru úspěšný.



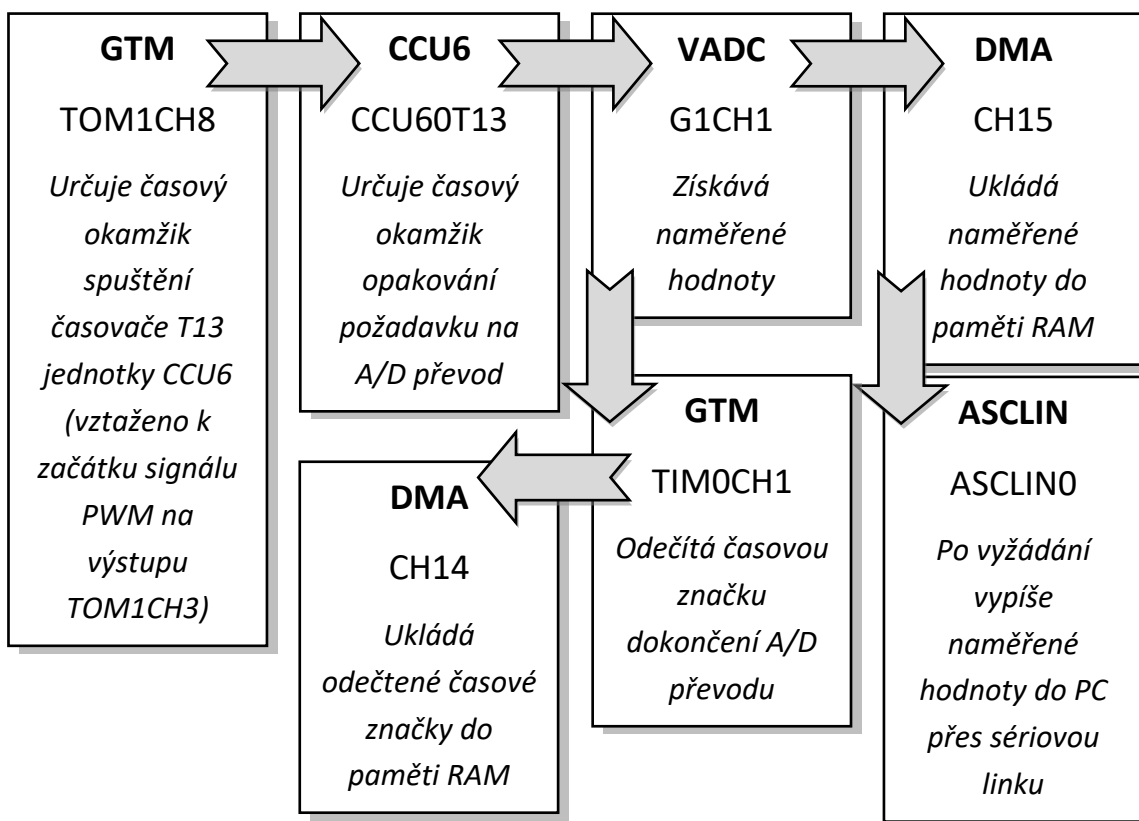
Obr. 17: Hlavní ovládací obrazovka LCD displeje

## 4 VYTVOŘENÁ MĚŘICÍ APLIKACE

Tato kapitola je věnována třetímu bodu zadání diplomové práce. Její obsah tvoří především popis vytvořené měřicí aplikace.

Dle zadání měla vytvořená aplikace umožnit vzorkování signálu s požadavkem na co nejkratší periodu vzorkování a na libovolné nastavení okamžiku zahájení měření. Toho lze následně využít například při diagnostice PMSM motoru, kdy lze provést analýzu průběhu proudů během spínání tranzistorů měniče.

Realizovaná aplikace používá framework BIFACES a demonstrační kódy pro ovladače iLLD verze 1.0.1.11.0. Jejich zdrojové kódy byly upraveny a následně umístěny do jednoho projektu tak, aby bylo dosaženo požadované funkce.



Obr. 18: Blokové schéma měřicí aplikace

Aplikace je naprogramována na jednom procesorovém jádře a využívá tyto periferie MCU:

- jednotku CCU6 (*Capture/Compare Unit 6*),
- řadič DMA (*Direct Memory Access*),
- modul GTM
- A/D převodník VADC,

- rozhraní ASCLIN (*Asynchronous/Synchronous Serial Controller with LIN (Local Interconnect Network)*).

Detailní popis jednotlivých periférií lze vyhledat v uživatelské příručce MCU TC234 [7] či v nápovědě k iLLD ovladačům [9].

## 4.1 Možnosti řešení zadaného problému

Vytváření programového kódu měřicí aplikace předcházelo důkladné seznámení se s jednotlivými perifériemi MCU TC234 v uživatelské příručce [7]. Především se jednalo o periferie GTM a VADC, jež jsou nakonfigurovány pro účely řízení motoru v ukázkové aplikaci FOC, jíž byla věnována kapitola 3.3. Teprve poté bylo možné uvažovat o dostupných možnostech řešení zadaného problému, který spočíval ve vzorkování signálu s požadavkem na libovolné nastavení okamžiku zahájení měření.

Cílem vytvořené aplikace bylo získat možnost spustit měření analogové hodnoty na základě znalosti vygenerovaného PWM signálu. Dále bylo žádoucí, aby se mohlo začít měřit dříve, než se změní logická hodnota PWM. Důraz byl také kladen na použití co nejvíce HW (*Hardware*) prostředků MCU.

Uvažováno bylo nad těmito možnostmi spouštění A/D převodu měřeného kanálu uvnitř periferie VADC:

1. Využít jako zdroj vstupního signálu pro A/D převod sekvenci měřicích kanálů.
  - Velikost měřicí sekvence by ale byla omezena pouze na osm měření, tj. osm vzorků.
  - Měření by mohlo být odstartováno pouze jednou, poté by bylo opakováno s periodou danou přibližně dobou A/D převodu.
2. Spouštět A/D převod pouze výstupním signálem z GTM TOM.
  - Pokud by se k tomuto účelu využíval pouze jeden kanál podmodulu TOM, muselo by se provést vždy jeho přenastavení, a to by se HW obtížně řešilo.
  - V případě použití dvou kanálů nebyl nalezen způsob, jak spustit časovač jednoho kanálu v návaznosti na druhý.
  - Kanál vyhrazený pro měřicí aplikaci by bylo možné HW propojit s ostatními kanály určenými pro generování PWM signálu.
3. Spouštět A/D převod prostřednictvím časovačů CCU6 T12 a T13.
  - Pomocí časovače T12 by se vygeneroval signál, kterým by se HW odstartoval časovač T13. Ten by poté sloužil k opakovanému spouštění A/D převodu v požadovaných okamžicích.
  - Obtížně by se realizovalo HW propojení s modulem GTM, pomocí něhož je generován PWM signál.

4. Spouštět A/D převod pomocí časovačů GTM TOM a CCU6 T13.
  - Tato možnost se zdála jako nejlepší, a proto byla zvolena pro výslednou realizaci měřicí aplikace. Její princip bude nastíněn v následujícím textu.

## 4.2 Popis funkce vybraných periférií v aplikaci

Následující text neobsahuje přímo popis programového řešení měřicí aplikace uvnitř MCU. Pozornost je věnována především prostředkům jednotlivých periférií použitých pro realizaci měřicí aplikace tak, aby byl nastíněn princip jejich funkce.

Signál PWM je generován výstupním podmodulem TOM periferie GTM. Jelikož není možné HW spustit časovač daného kanálu tohoto podmodulu, bylo potřeba najít způsob, který by toto umožnil. Řešení spočívá ve využití periferie CCU6, jejíž časovač T13 je možné HW spustit výstupním signálem z periferie GTM.

Pro A/D převod měřeného signálu byl zvolen převodník s postupnou aproximací (periferie VADC).

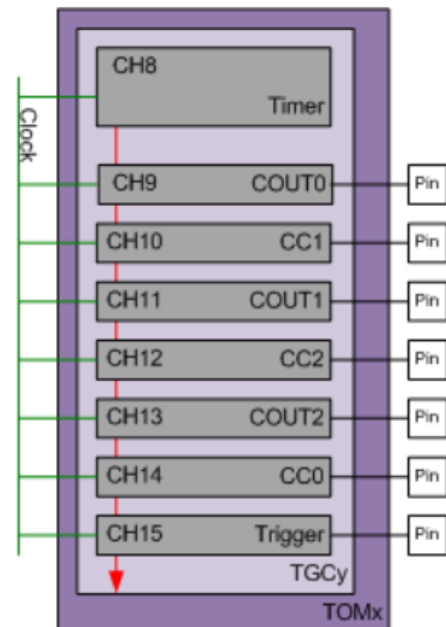
Získané vzorky měřeného signálu jsou poté pomocí řadiče DMA ukládány do paměti RAM bez nutnosti vytěžování procesoru.

### 4.2.1 Modul GTM

Výstupní podmodul TOM1 periferie GTM se při řízení motoru používá pro generování signálů PWM, jak bylo zmíněno dříve v podkapitole 3.3.1.1.

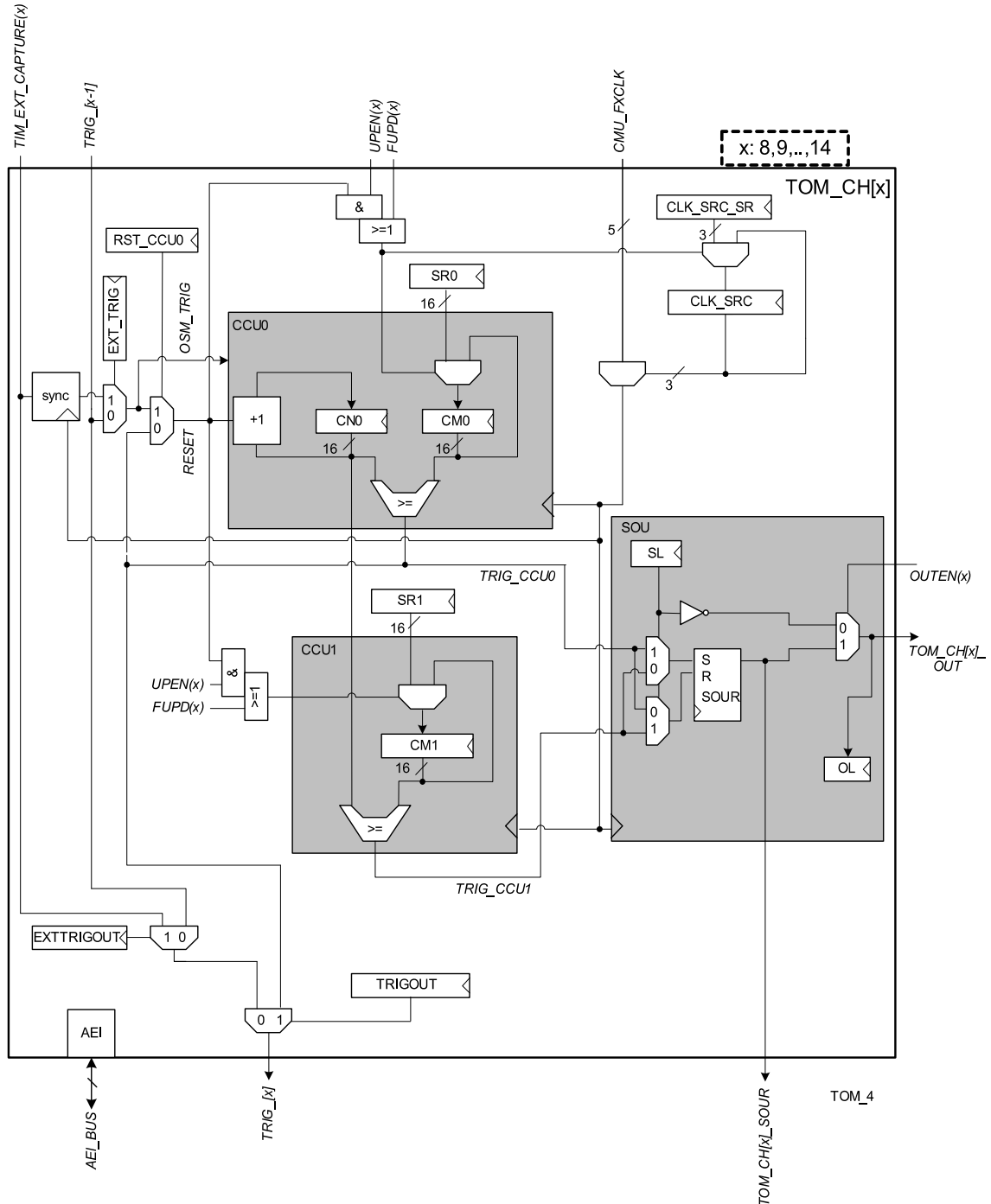
Pro připomenutí, kanál CH0 je určen pro generování referenčního signálu, pomocí něhož jsou vytvářeny signály na kanálech CH1 až 6. Ty jsou používány pro spínání tranzistorů uvnitř měniče motoru, tj. třech horních a třech dolních tranzistorů. Výstupní signál kanálu CH7 slouží pro zahájení měření proudů, podle kterých se následně počítají nové hodnoty PWM. Možné uspořádání kanálů podmodulu TOM pro generování PWM ilustruje Obr. 19, číslování kanálů CH8 až 15 je na něm však odlišné od popsáno.

K tomuto uspořádání byl nově přidán kanál CH8. Demonstrační kód pro iLLD ovladač podmodulu TOM periferie GTM ve funkci jednoduchého časovače, který byl upraven a aplikován, se nazývá *GtmTomTimerDemo*.



Obr. 19: Ilustrace uspořádání kanálů podmodulu TOM pro generování PWM [9]

Vnitřní struktura použitého kanálu, viz Obr. 20, ho umožňuje propojit s předchozím kanálem CH7. Díky tomu lze zaručit, že i tento kanál bude vnitřně svázan s referenčním kanálem CH0, tedy obdobně jako kanály vyhrazené pro generování PWM. Proto se pro něj uplatňuje perioda PWM daná kanálem CH0, a to 50  $\mu$ s.

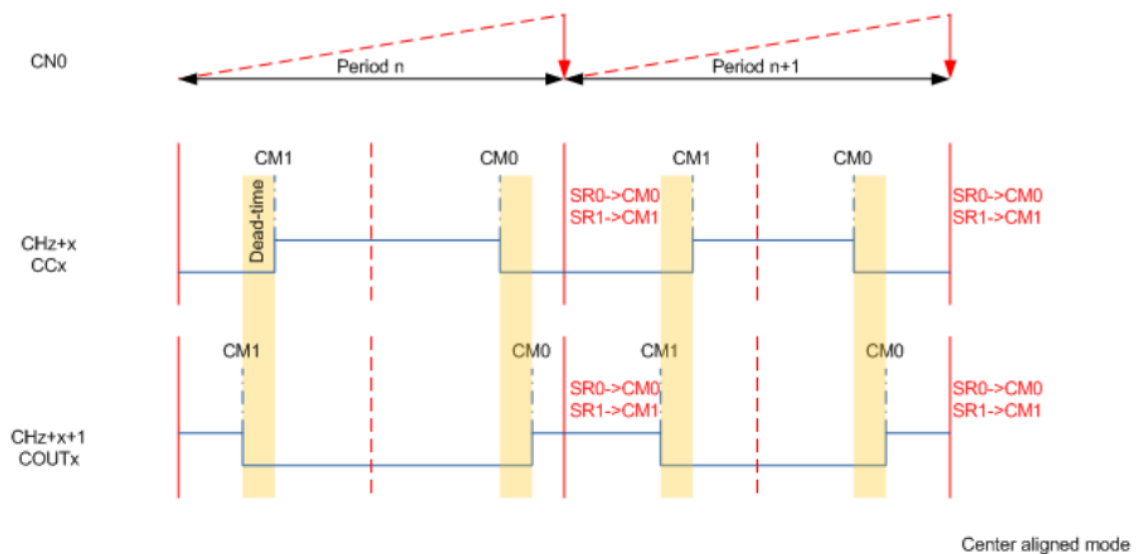


Obr. 20: Vnitřní struktura kanálu 8 až 14 podmodulu TOM [7]

Vnitřní zapojení signálů mezi periferiemi MCU TC234 usnadňuje získání spouštěcího signálu pro periférii CCU6. Výstupní signál TOM1CH8 lze totiž použít pro HW spuštění časovače T13 uvnitř jednotky CCU60 periferie CCU6.

Způsob generování PWM signálu zarovnaného na střed podmodulem TOM bližší zázorňuje Obr. 21. Na konci periody čítače kanálu CH0 se aktualizuje obsah porovnávacích registrů CM0 a CM1, jejichž nové hodnoty byly předtím uloženy ve stínových registrech SR0 a SR1.

Spouštěcí signál na kanálu TOM1CH7, resp. TOM1CH8, se generuje na základě shody aktuální hodnoty registru CN0 s nastavenou hodnotou v registru CM1. Okamžik spuštění je možné nastavit konfiguračním parametrem, viz Tab. 5.



Obr. 21: Princip vytváření PWM signálu pomocí podmodulu TOM [9]

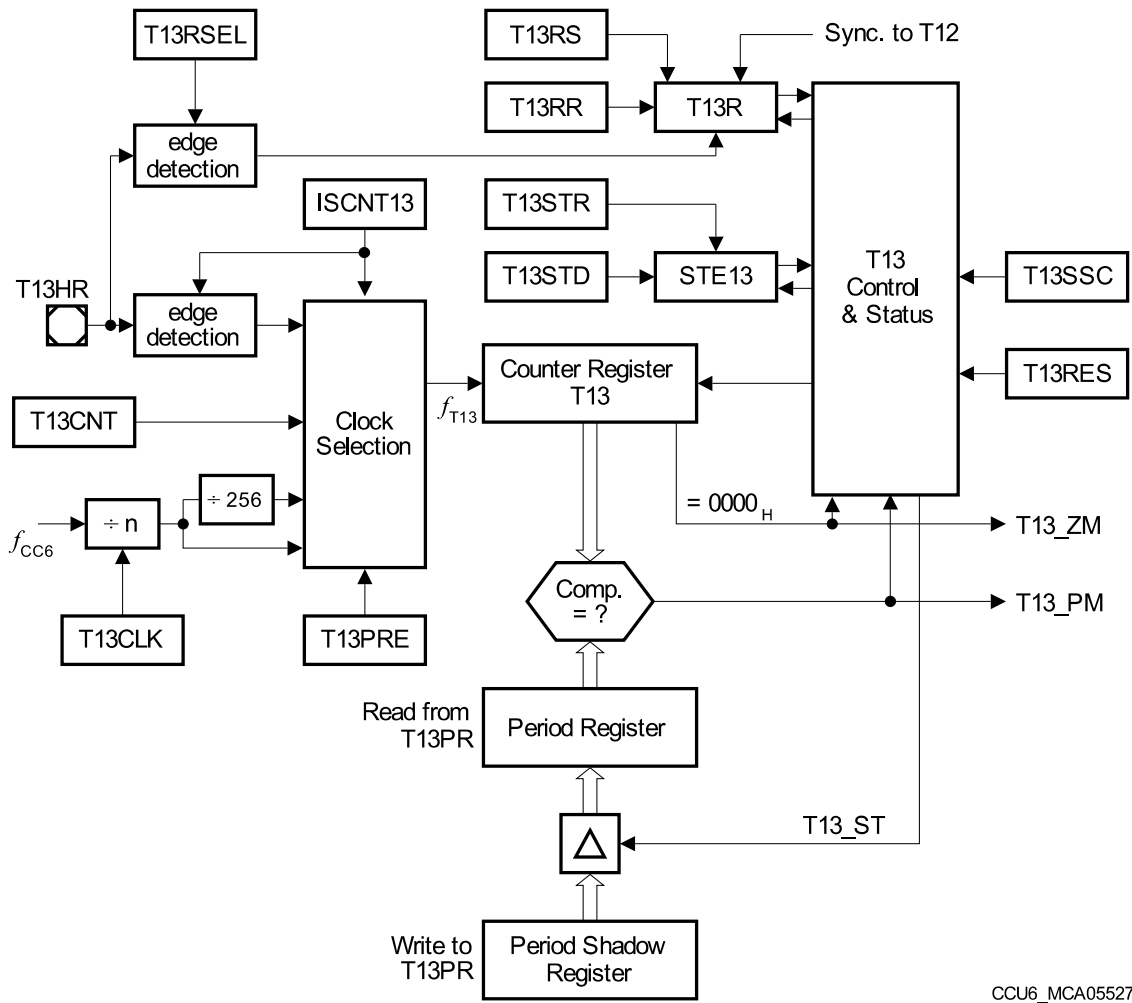
## 4.2.2 Jednotka CCU6

Měřicí aplikace využívá blok CCU60 periferie CCU6, konkrétně její časovač T13, jehož vnitřní struktura je blokově znázorněna na Obr. 22. Demonstrační kód pro iLLD ovladač této periferie ve funkci jednoduchého časovače, který byl upraven a aplikován, se nazývá *Ccu6TimerDemo*.

Frekvence hodinového signálu pro časovač T13 byla zvolena 25 MHz. Perioda tohoto časovače nastavená v registru T13PR určuje okamžik, kdy je opakovaně zahajováno A/D měření, čímž lze zajistit ekvidistantní vzorkování měřeného signálu.

Časovač T13 je odstartován až po příchodu náběžné hrany signálu TOM1CH8 periferie GTM. Při inicializaci se do jeho registru T13PR nastaví hodnota 50, což s ohledem na použitý hodinový signál znamená, že bude časovat s periodou 2  $\mu$ s, tzn. že se měření na VADC spustí vždy po uplynutí tohoto času. Nastavení časovače T13 lze v případě potřeby změnit konfiguračními parametry, viz Tab. 5.

Spouštěcí signál pro VADC může být generován buď výstupním blokem COUT63 obdobným způsobem jako u modulu GTM (nevýhodou je, že je toto vnitřně řešeno klopným obvodem, takže musí proběhnout alespoň jeden cyklus časovače), nebo prostřednictvím žádosti o obsluhu, která nastane při události, kdy časovač T13 dosáhne své periody (v tomto případě lze časovači přednastavit hodnotu tak, aby po spuštění byl ihned vygenerován požadovaný signál). Zvolen byl nakonec druhý způsob.



CCU6\_MCA05527

Obr. 22: Vnitřní struktura časovače T13 jednotky CCU6 [7]

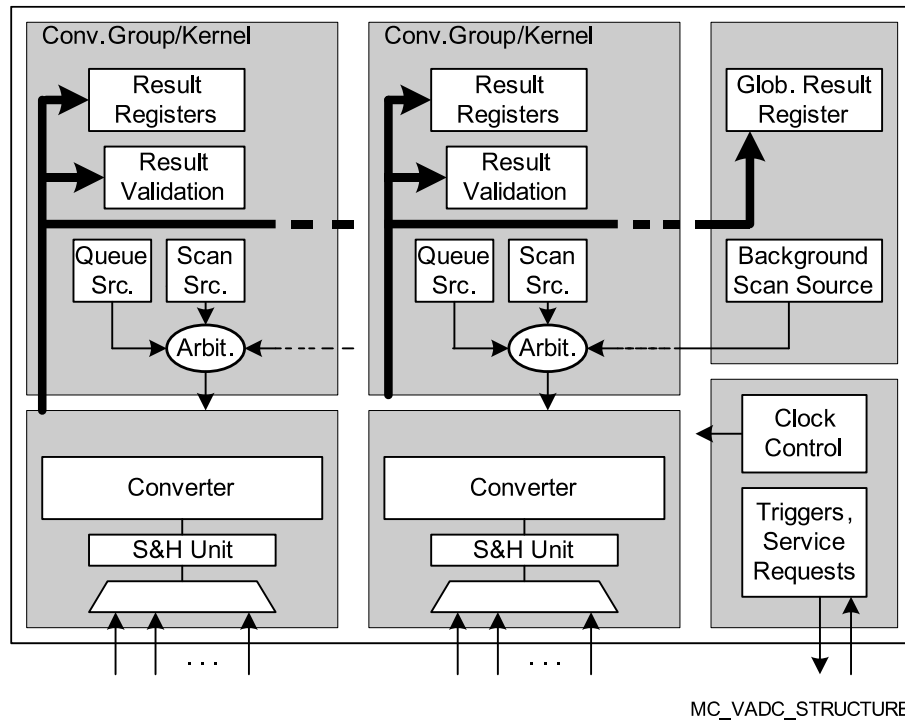
### 4.2.3 A/D převodník VADC

Pro A/D převod měřeného signálu byl zvolen převodník s postupnou aproximací VADC, jehož vnitřní struktura je uvedena na Obr. 23. Detailněji rozkreslené blokové schéma převodníkového jádra, resp. skupiny, ilustruje Obr. 24.

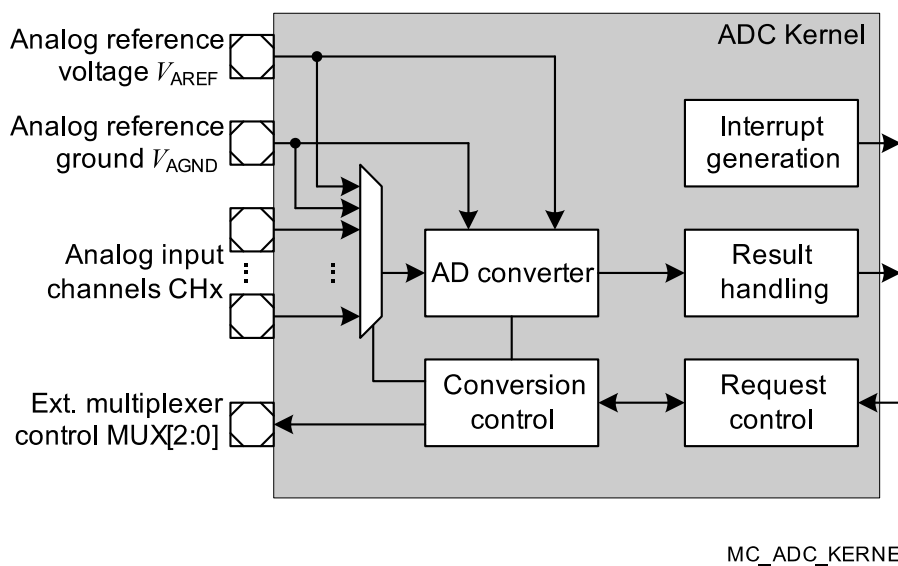
Z textu podkapitoly 3.3.1.2 věnované této periférii v rámci ukázkové aplikace FOC je zřejmé, že jsou obě dostupná převodníková jádra obsazena pro řízení motoru. S tím souvisí i nedostupnost jednoho zdroje pro A/D převod, a to sekvenční typ. Z toho plyne, že bylo nutné realizovat požadované měření

s ohledem na stávající stav, tj. přidat do současného řešení další zdroj pro A/D převod.

V dokumentaci k periférii VADC, která je součástí uživatelské příručky [7], lze nalézt informaci, že lze kromě sekvenčního zdroje pro A/D převod využít například režim skenování. Měření signálu poté neprobíhá v sekvenčním režimu, tj. podle uživatelem zvolené sekvence měřících kanálů, ale v režimu skenování, kdy se provádí převod všech povolených kanálů od nejvyššího čísla po nejnižší.



Obr. 23: Vnitřní struktura A/D převodníku VADC [7]



Obr. 24: Blokové schéma převodníkového jádra VADC [7]



O tom, jaký převod bude právě na převodníkovém jádře probíhat, rozhoduje arbit. Ten v cyklu vyhodnocuje, zdali nepřišel požadavek na zahájení převodu některého z povolených zdrojů. Jednotlivé zdroje mají přidělenou prioritu a lze zvolit, co se má stát v případě, kdy je během převodu zdroj s nižší prioritou (první) požadován převod zdroje s vyšší prioritou (druhý). Buď je okamžitě převod prvního zdroje přerušen, což je následováno zahájením převodu druhého, a teprve po jeho provedení může být zopakován převod prvního, anebo se počká na okamžik, kdy se dokončí převod prvního zdroje, až poté se zahájí převod druhého.

Na základě výše zmíněných informací byl programový kód ukázkové aplikace FOC pro konfiguraci jedné měřicí skupiny periferie VADC doplněn tak, aby zvolený A/D převodník umožnil měření ze dvou samostatných zdrojů. K tomuto účelu byl vybrán kanál CH1 skupiny Group1, který byl přidán do zdroje v režimu skenování. Tomuto zdroji byla nastavena nižší priorita než sekvenčnímu s případným okamžitým ukončením převodu, aby nebyla narušena plynulost řízení motoru.

Další konfigurace tohoto zdroje vyhověla požadavkům na HW spuštění měření na vstupním kanálu G1CH1 signálem z periferie CCU6 a vyvolání žádosti o obsluhu pro řadič DMA po dokončení A/D převodu.

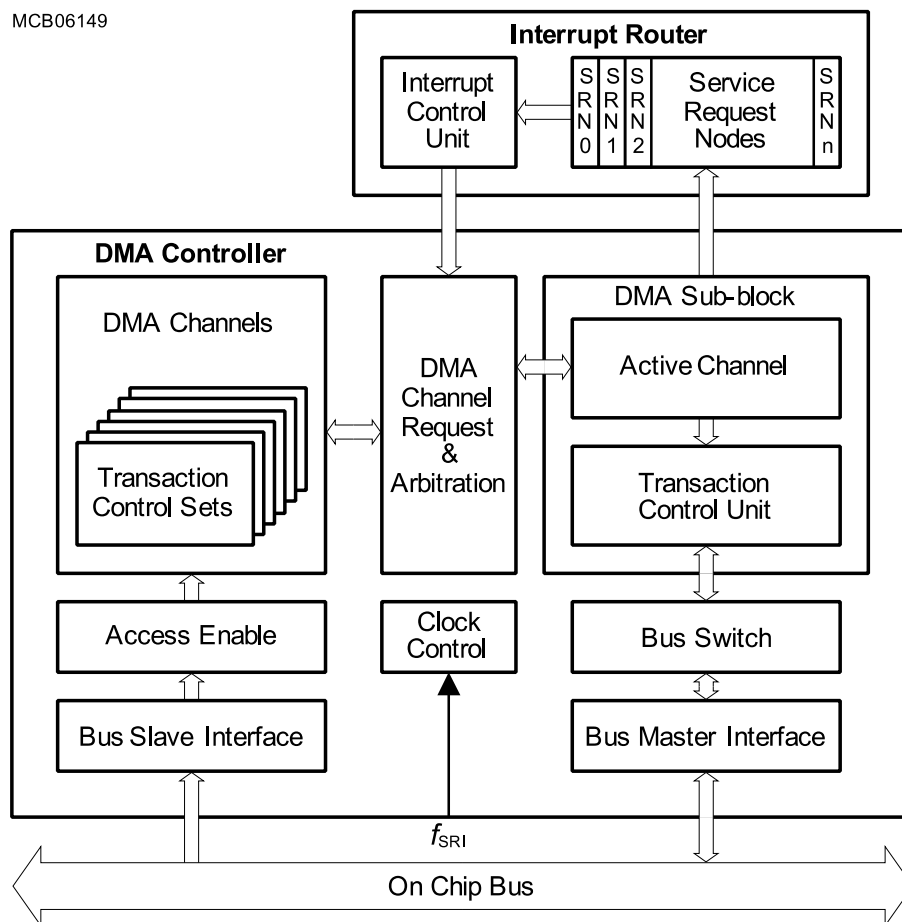
Nastavení A/D převodu tohoto kanálu bylo zvoleno 12bitové, doba vzorkování 0,5  $\mu$ s, s kalibrací po každém převodu, z čehož plyne celková doba převodu cca 1,2  $\mu$ s. Vzorkovací dobu je možné změnit pomocí konfiguračního parametru, viz Tab. 5.

#### 4.2.4 Řadič DMA

Periferie řadiče DMA, viz blokové schéma na Obr. 25, se využívá pro přenos výsledků VADC měření do paměti RAM (*Random Access Memory*) bez nutnosti vytěžování procesoru. Demonstrační kód pro iLLD ovladač této periferie, který byl upraven a aplikován, se nazývá *DmaBasicDemo*.

Výsledky měření jsou do paměti přenášeny prostřednictvím DMA kanálu CH15. Ukládány jsou postupně za sebou počínaje adresou, na níž se nachází první prvek prvního rozměru dvojrozměrného globálního pole *g\_vadcResultBuffer*. Vyvoláním žádosti o obsluhu od periferie VADC se HW spouští přenos v rámci jedné transakce tohoto kanálu. Ta se v případě této aplikace skládá ze 20 přenosů, tj. 20 měření. Počet přenesených vzorků lze nastavit konfiguračním parametrem, viz Tab. 5.

Po dokončení přenosu kompletní transakce je vyvoláno přerušení. Jakmile dojde k jeho zpracování, zjednodušeně lze říci, že se časovač T13 periferie CCU6 zastaví a čeká se na příchod spouštěcího signálu, který odstartuje další sérii měření.



Obr. 25: Blokové schéma řadiče DMA [7]

## 4.3 Propojení s ukázkovou aplikací FOC

Tato kapitola se věnuje propojení měřicí aplikace s ukázkovou aplikací FOC, již byla věnována kapitola 3.3.

### 4.3.1 Zařazení do projektové struktury

Obsah Tab. 3 je věnován projektové struktuře měřicí aplikace, a proto jsou zde uvedeny soubory, kterými je tato aplikace tvořena. Lze je rozdělit do dvou skupin podle toho, zdali patří do ukázkové aplikace FOC, nebo do ní byly nově přidány.

Tab. 3: Projektová struktura měřicí aplikace

<i>Cesta</i>	<i>Soubor</i>	<i>Popis</i>
<b>DOPLNĚNO DO STÁVAJÍCÍHO KÓDU UKÁZKOVÉ APLIKACE FOC</b> <i>Cesta: &lt;projekt&gt;\0_Src\AppSw\</i>		
<i>Config\Common\</i>	<i>ConfigurationIsr.h</i>	<i>Konfigurace priorit a zpracovatelů přerušení</i>
<i>Tricore&gt;Main\</i>	<i>Cpu0_Main.c</i>	<i>Volání inicializačních funkcí periferií</i>

<b>Cesta</b>	<b>Soubor</b>	<b>Popis</b>	
Tricore\DemoApp\	AppTorqueControl.c	Doplnění kódu pro	zahájení měření (uvnitř funkce TorqueControl_FocStep)
	AppTaskFu.c		výpis naměřených hodnot (uvnitř funkce appTaskfu_200ms)
Config\Common\	lfx_Cfg_Vadc.h/.c	Konfigurace a obsluha periferie VADC	
Tricore\DemoApp\	AppVadc.h/.c		
<b>NOVĚ PŘIDÁNO</b>			
<projekt>\0_Src\ AppSw\Tricore\ MeasApp\	AsclinShellInterface.h/.c	Funkce určené k inicializaci periferie pro danou činnost, případně její obsluhu	ASCLIN
	Ccu6TimerDemo.h/.c		CCCU6
	DmaBasicDemo.h/.c		DMA
	GtmTimCaptureTest.c		GTM TIM
	GtmTomTimerDemo.h/.c		GTM TOM

Pozn.: \*.h hlavičkový, \*.c zdrojový soubor

### 4.3.2 Zdroj analogového signálu pro VADC

Kanály VADC vyhrazené pro výpočet algoritmu FOC bylo žádoucí oddělit od kanálů měřicí aplikace kvůli jejich předpokládanému rozdílnému nastavení. Aby se tomuto požadavku vyhovělo, bylo třeba najít způsob, jak připojit analogový signál z můstkového budiče TLE9180 na dva měřicí kanály současně. To ale bylo ztíženo tím, že se vývojová deska s MCU přímo spojuje s výkonovou deskou prostřednictvím dvou párů kolíkových a dutinkových lišt, a proto k nim nelze jednoduše připojit další vodiče bez dodatečného pájení provedeného pravděpodobně na straně konektorů vývojové desky.

Tento problém by se nejlépe vyřešil vyrobením desky plošného spoje, která by se vkládala mezi vývojovou a výkonovou desku, čímž by se zvýšil počet dostupných vstupně-výstupních pinů. Zároveň by byl tento plošný spoj navržen tak, aby byla zajištěna různorodost v možnostech jeho dalšího využití. Touto cestou se však dále nepokračovalo, poněvadž bylo odhadnuto, že by k jeho realizaci nebyl dostatečný časový prostor (návrh a výroba desky plošného spoje, osazení součástkami apod.).

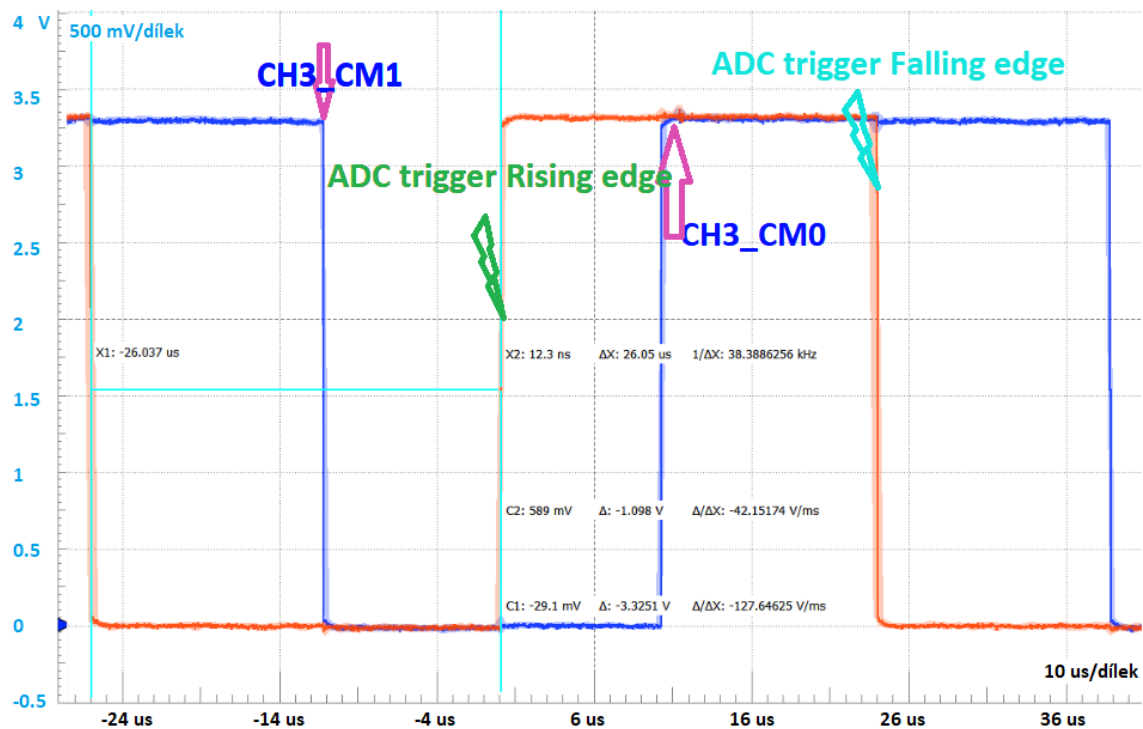
Jako nejjednodušší se zdálo piny s analogovými vstupy na kolíkových lištách mezi sebou propojit, tzn. připájet vodič na spodní stranu tištěného spoje vývojové

desky. Nakonec k tomu nedošlo, neboť periferie VADC poskytuje pro každou skupinu A/D převodníků funkci ALIAS, pomocí níž lze přesměrovat měření na jiný kanál při zachování nastavení původního kanálu. Měření na vstupním kanálu CH1 skupiny Group1 bylo tedy možné změnit na kanál CH0, na němž je zapojen analogový vstup z můstkového budiče TLE9180 pro měření proudu druhou fází.

Pro testování uživatelem vygenerovaným signálem byl vyčleněn kanál CH2, jehož vstup je nezapojen. Měření tedy může probíhat buď na kanálu CH0 díky funkci ALIAS, anebo CH2 v případě potřeby připojení odlišného analogového signálu. Přepínání mezi kanály je řešeno konfiguračním parametrem, viz Tab. 5

### 4.3.3 Analýza signálu PWM z podmodulu TOM

Aby bylo možné libovolně posouvat okamžik začátku měření proudu druhou fází během periody PWM, bylo třeba zjistit, jakým způsobem je generován spínací signál pro dolní tranzistor této fáze. Z tohoto důvodu byly pomocí osciloskopu zobrazeny výstupní signály na kanálech CH3 a CH7 podmodulu TOM1, viz Obr. 26.



Obr. 26: Průběh z osciloskopu, výstupy TOM1 – modrá CH3, červená CH7

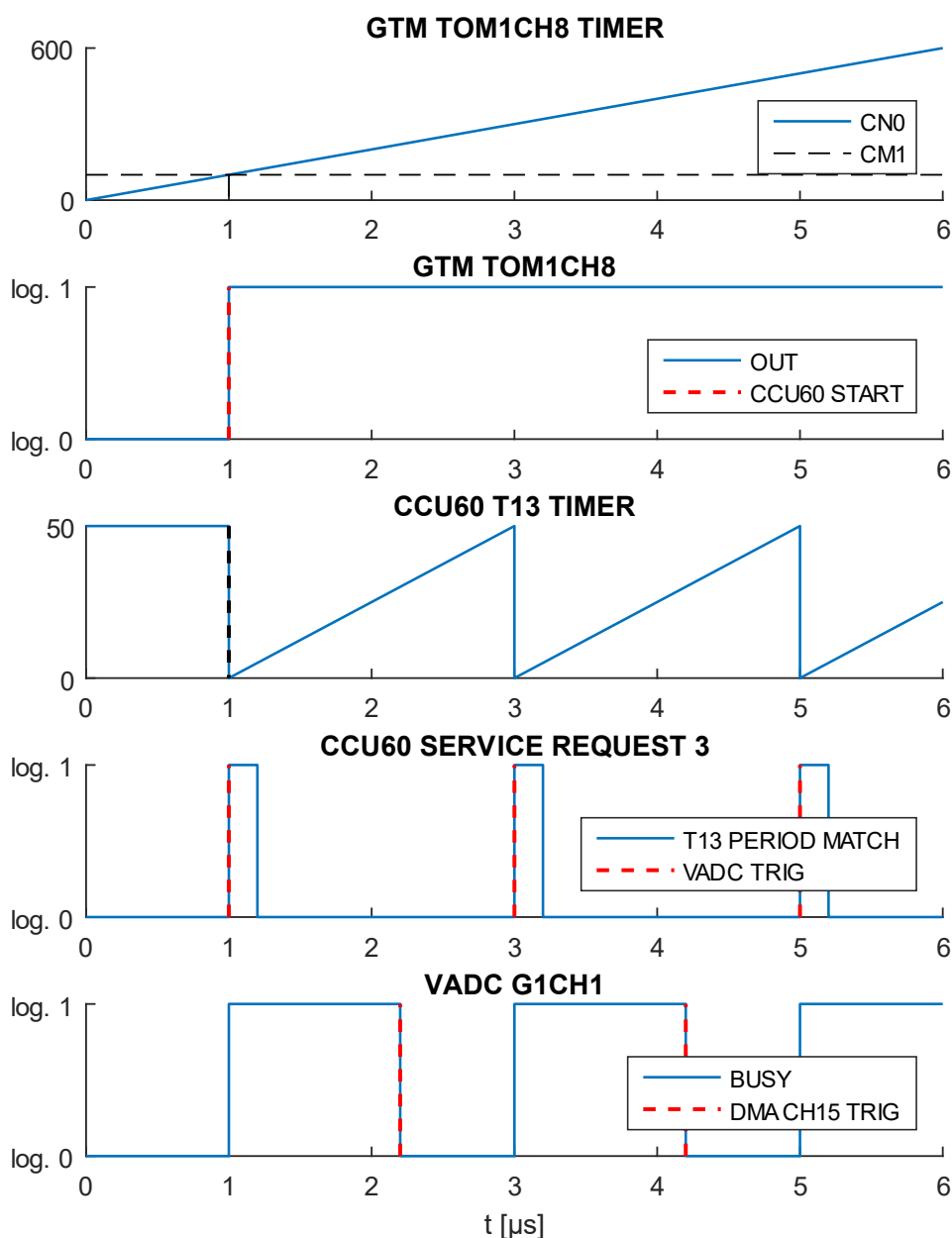
První průběh odpovídá jedné periodě signálu PWM pro spodní tranzistor druhé fáze. Bylo zjištěno, že umístění hran je dáno hodnotami uloženými v registrech CM0 (sestupná) a CM1 (náběžná) kanálu CH3. Tato informace usnadnila realizaci měření v průběhu periody PWM.

Pokud se provede srovnání s Obr. 21, je patrné, že průběh získaný z výstupu kanálu CH3 je zakreslen na spodnímu grafu.

Druhý průběh naopak zobrazuje spouštěcí signál pro převod VADC, po jehož dokončení je proveden výpočet FOC. Protože se dolní tranzistor spíná po přivedení log. 1, je převod VADC spouštěn právě na sestupnou hranu tohoto signálu. Rovněž lze z tohoto průběhu vysledovat periodu PWM, která je dána dobou mezi po sobě jdoucími sestupnými hranami tohoto signálu.

#### 4.4 Grafické znázornění funkce aplikace

Dříve popsané spuštění časovače T13 jednotky CCU6 výstupním kanálem TOM1CH8 periferie GTM a následné zahájení převodu VADC graficky znázorňuje Obr. 27.

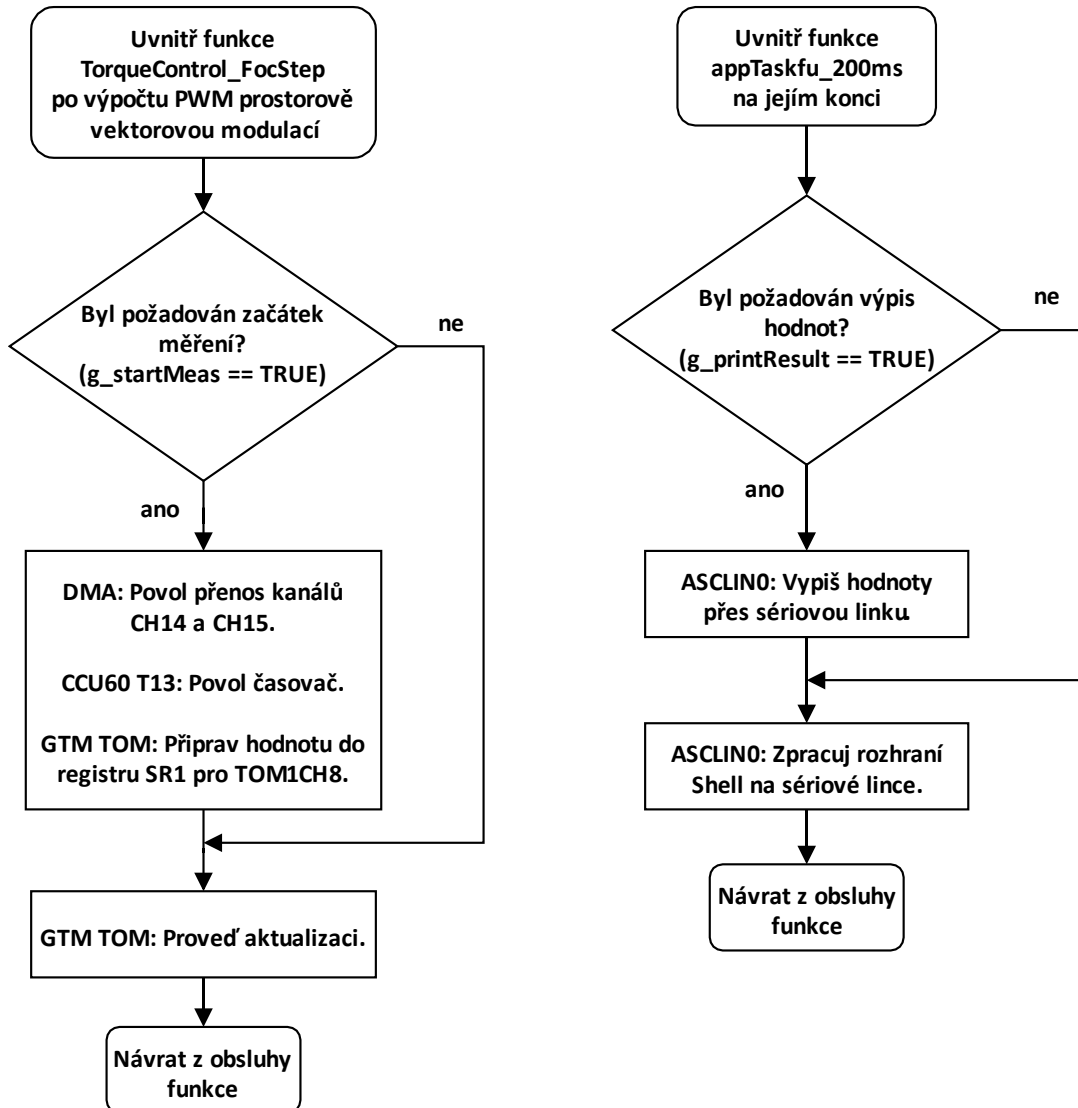


Obr. 27: Grafické znázornění funkce aplikace

## 4.5 Zjednodušené vývojové diagramy

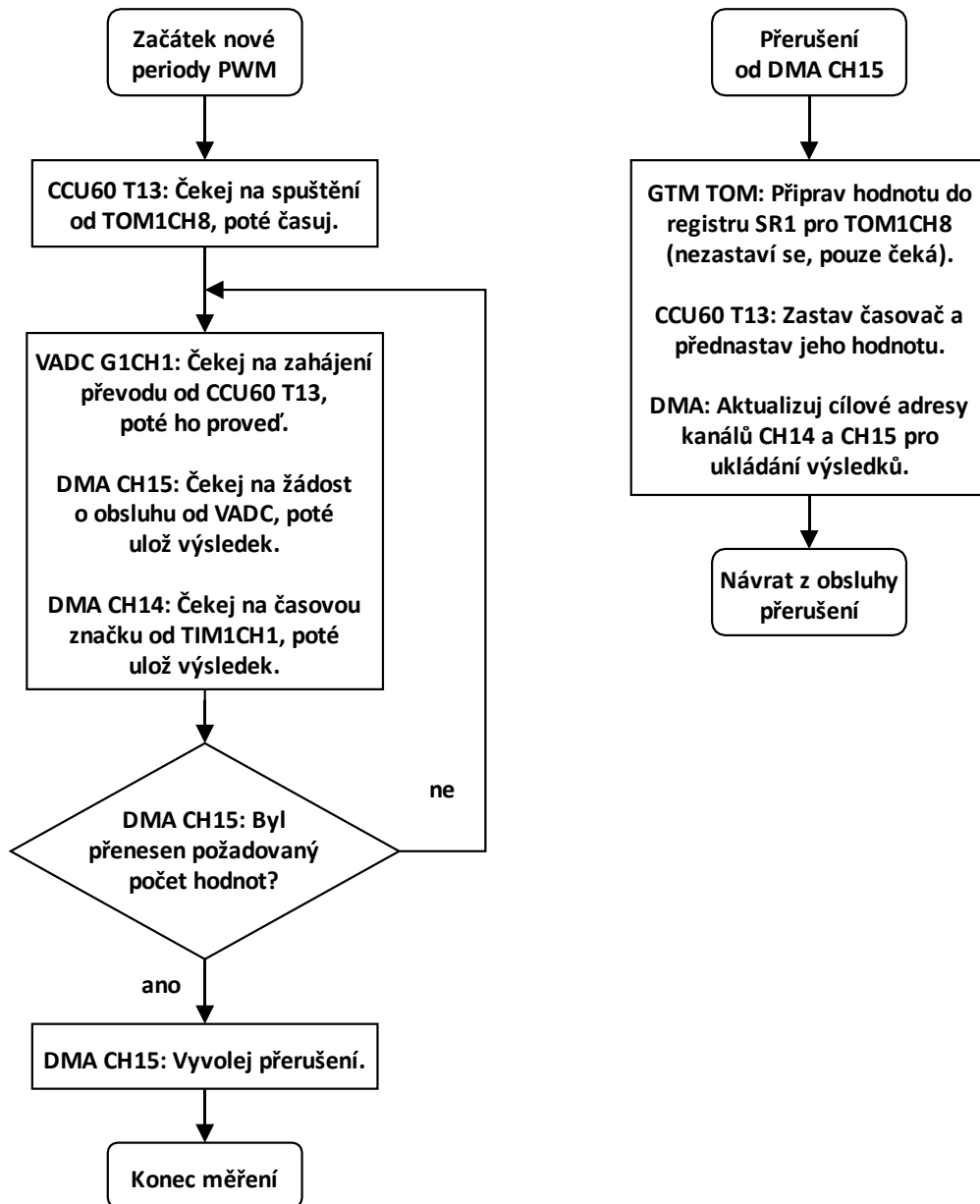
Následující obrázky by měly přiblížit funkci vytvořené měřicí aplikace uvnitř ukázkové aplikace FOC. Jsou na nich zakresleny zjednodušené vývojové diagramy.

### 4.5.1 Zahájení měření a výpis naměřených hodnot



Obr. 28: Zjednodušený vývojový diagram zahájení měření a výpisu naměřených hodnot

## 4.5.2 Hlavní část měřicí aplikace



Obr. 29: Zjednodušený vývojový diagram hlavní části měřicí aplikace

## 4.6 Získávání časových značek

Ukládat do paměti pouze naměřené hodnoty z A/D převodníku se může zdát dostačující, avšak z důvodu překrytí dvou různých zdrojů pro převod je vhodnější se pokusit najít způsob, jak k nim přidat také informaci o časové značce. Lze tak snáze provést kontrolu časového kroku, s nímž jsou jednotlivé výsledky získávány, i porozumět, jak se oba zdroje převodu vzájemně ovlivňují.

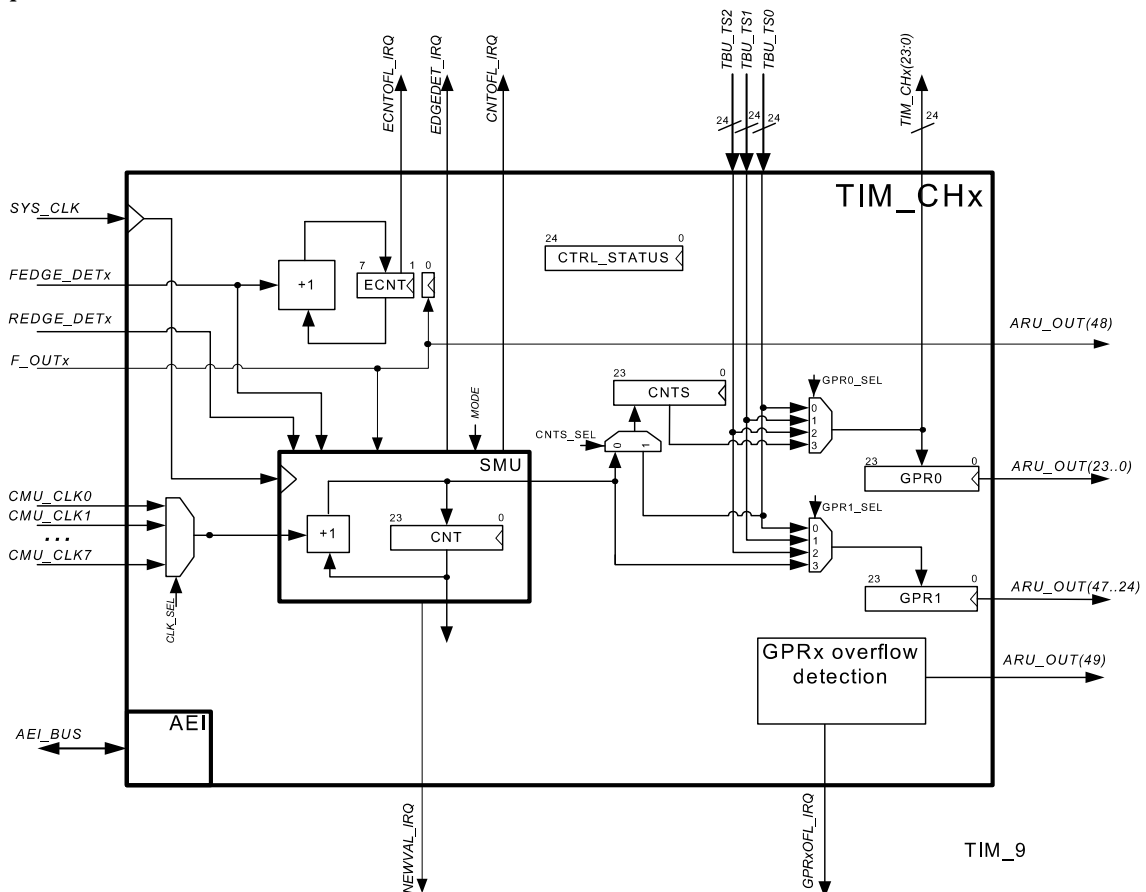
Za tímto účelem byly vyzkoušeny dva možné přístupy, které budou dále představeny.

## 4.6.1 Přerušení od DMA kanálu

První, jednodušší myšlenkou bylo využít přerušení od DMA kanálu. Kanály byly vyhrazeny dva. Prostřednictvím prvního kanálu se ukládaly měřené hodnoty z VADC do paměti a pomocí druhého jim přidružené časové značky.

Princip odečítání časové značky byl následující. Po uložení naměřené hodnoty prvním kanálem se vyžádalo přerušení pro druhý kanál DMA. Jakmile byl tento požadavek řadičem DMA zpracován, uložila se do paměti aktuální hodnota registru CN0 kanálu CH0 podmodulu TOM1, tj. referenčního kanálu pro generování PWM.

Tento způsob odečítání časových značek se nakonec ukázal jako nevhodný, protože získané časové značky obsahovaly zpoždění dané dobou zpracování prvního kanálu DMA.



Obr. 30: Vnitřní struktura kanálu podmodulu TIM [7]

## 4.6.2 Vstupní podmodul TIM

Inspirací k tomu, aby se časové značky získávaly pomocí vstupního podmodulu TIM periferie GTM, bylo řešení představené v iLLD ovladačích MCU TC277 (TC234 tuto periferii neobsahuje) pro periferii sigma-delta A/D převodníku DSDAC (*Delta-*



*Sigma Analog-to-Digital Converter*). Připravené programové rozhraní *IfxDsadc\_Rdc* umožňuje měřit úhlové natočení motoru prostřednictvím resolveru.

Princip odečítání časové značky je v tomto případě jednoduchý. Příslušný kanál vstupního podmodulu TIM, viz Obr. 30, je nakonfigurován v režimu zachycení vstupní události, což znamená, že se na základě vyhodnocení nastavené hrany spouštěcího signálu uloží do registru GPR0 časová značka TBU\_TS0. Hodnota této značky odpovídá výstupu kanálu CH0 podmodulu časové základny TBU (*Time Base Unit*), jenž je časován z podmodulu CMU (*Clock Management Unit*) hodinovým signálem CLK0 o frekvenci 100 MHz.

Hodnota uložená v registru GPR0 tedy odpovídá časové značce vstupní události dané spouštěcím signálem. Navíc lze aktivovat vyvolání přerušení při uložení nové hodnoty.

Pro získávání časových značek byly vyhrazeny dva vstupní kanály CH0 a CH1 podmodulu TIM0, jejichž konfiguraci obsahuje Tab. 4. První vyjmenovaný kanál odečítá časovou značku začátku PWM a druhý okamžik dokončení převodu kanálu z VADC.

Toto řešení se ukázalo jako nejvhodnější.

**Tab. 4: Parametry vstupních kanálů podmodulu TIM pro získávání časových značek**

<b>Časová značka</b>	<b>ZAČÁTKU PWM</b>	<b>MĚŘENÉ HODNOTY</b>
<b>Podmodul TIM, kanál</b>	TIM0, CH0	TIM0, CH1
<b>Zdroj časové značky</b>	TBU_TS0, časováno z CMU_CLK0 (100 MHz)	TBU_TS0, časováno z CMU_CLK0 (100 MHz)
<b>Spouštěcí signál</b>	vnější signál (pin P00.9) – výstup kanálu CH7 podmodulu TOM1 (pin P00.8)	vnitřní signál – žádost o obsluhu C1SR0 po dokončení převodu kanálu VADC
<b>Spouštěcí hrana</b>	sestupná	náběžná
<b>Přerušení</b>	ne, hodnota ukládána programově do globální proměnné <i>g_tsPWM</i>	ano, po uložení nové hodnoty, DMA CH14, obdobný princip jako pro hodnoty z VADC, ukládáno do druhého rozměru globální proměnné <i>g_vadcResultBuffer</i>

## 4.7 Zahájení měření a zobrazení naměřených hodnot

Zahájit měření a zobrazit naměřené hodnoty lze v PC buď přímo uvnitř ladicího nástroje, anebo prostřednictvím programu PuTTY po připojení se k sériové lince otevřené na virtuálním komunikačním portu přes USB s nastavenou rychlostí 115200 bit/s. Přenos dat je realizován pomocí rozhraní UART (*Universal Asynchronous Receiver-Transmitter*). Za tímto účelem byl použit a upraven

ukázkový kód pro iLLD ovladač této periferie s názvem *AsclinShellInterface*. Na otevřeném komunikačním portu se zobrazí rozhraní Shell, které umožňuje posílat příkazy do MCU.

Měření je zahájeno zápisem log. 1 do bitové proměnné *g\_startMeas* uvnitř ladicího nástroje v PC, nebo vysláním příkazu *startMeas* po sériové lince.

Naměřené hodnoty jsou průběžně ukládány do dvourozměrného pole *g\_vadcResultBuffer* v paměti RAM a lze je v PC zobrazit buď přímo v ladicím nástroji, nebo je vypsát pomocí sériové linky. První rozměr pole *g\_vadcResultBuffer* je určen pro hodnoty získané z VADC a druhý pro jim odpovídající časové značky z GTM TIM. Získaná data se vypíší do otevřeného okna sériové linky po zápisu log. 1 do bitové proměnné *g\_printResult*, nebo vysláním příkazu *printRes* po sériové lince.

Alternativně by se mohl přidat další způsob zahájení měření a zobrazení naměřených hodnot, a to doplněním uživatelského rozhraní na LCD (*Liquid Crystal Display*) displeji, se kterým již pracuje ukázková aplikace FOC.

```

Cas konverze CH_I_U: 1.220000 us
Cas konverze CH_I_V: 1.220000 us
Cas konverze CH_I_V_2: 1.220000 us
---RESULTS---
PWM TIMESTAMP-RAW:      0, -usec:   0.00
TRIGGER POINT-RAW: 2688, -usec: 26.88
ADC SAMPLE  TIME-raw    TIME-usec    dT-usec    VALUE-RAW    VALUE-V
    0         61718      0.00         0.00        1850         2.259
    1         61918      2.00         2.00        1819         2.221
    2         62118      4.00         2.00        1800         2.198
    3         62318      6.00         2.00        1787         2.182
    4         62518      8.00         2.00        1780         2.173
    5         62718     10.00        2.00        1772         2.164
    6         62918     12.00        2.00        1762         2.151
    7         63118     14.00        2.00        1763         2.153
    8         63318     16.00        2.00        1765         2.155
    9         63518     18.00        2.00        1765         2.155
   10         63718     20.00        2.00        1766         2.156
   11         64515     27.97        7.97        1765         2.155
   12         64718     30.00        2.03        1766         2.156
   13         64918     32.00        2.00        1765         2.155
   14         65118     34.00        2.00        1765         2.155
   15         65318     36.00        2.00        1775         2.167
   16         65518     38.00        2.00        1772         2.164
   17          182     40.00        2.00        1770         2.161
   18          382     42.00        2.00        1769         2.160
   19          582     44.00        2.00        1767         2.158
---END---
```

**Obr. 31: Příklad výstupu měřicí aplikace v programu PuTTY**

Příklad, jak může vypadat výstup měřicí aplikace v programu PuTTY, je uveden na Obr. 31. Z obrázku je patrné, že jsou nejprve vypsány odpovídající časy konverze jednotlivých měřicích kanálů (proud první a druhou fází a přidané

měření v rámci měřicí aplikace). Dále je to časová značka začátku PWM, časový okamžik, kdy byl vyslán spouštěcí signál pro časovač T13 jednotky CCU6 na výstupu TOM1CH8 a nakonec tabulka s naměřenými hodnotami, tj. jednotlivými získanými vzorky a jim odpovídajícími časovými značkami. Pokud byla hodnota časové značky začátku PWM k dispozici, tj. různá od nuly, sloupec TIME-usec obsahuje časové okamžiky vztažené právě k této hodnotě, jinak je výchozím bodem první naměřená časová značka. Ve sloupci dT-usec je uveden časový rozdíl dvou po sobě jdoucích časových značek. Časová značka naměřené hodnoty odpovídá okamžiku dokončení jejího převodu, tzn. že v sobě zahrnuje i dobu převodu.

## 4.8 Vybrané konfigurační parametry

Pro případnou změnu v konfiguraci měřicí aplikace byly uvnitř Tab. 5 vyjmenovány vybrané konfigurační parametry, které by měly tuto práci usnadnit.

Tab. 5: Vybrané konfigurační parametry měřicí aplikace

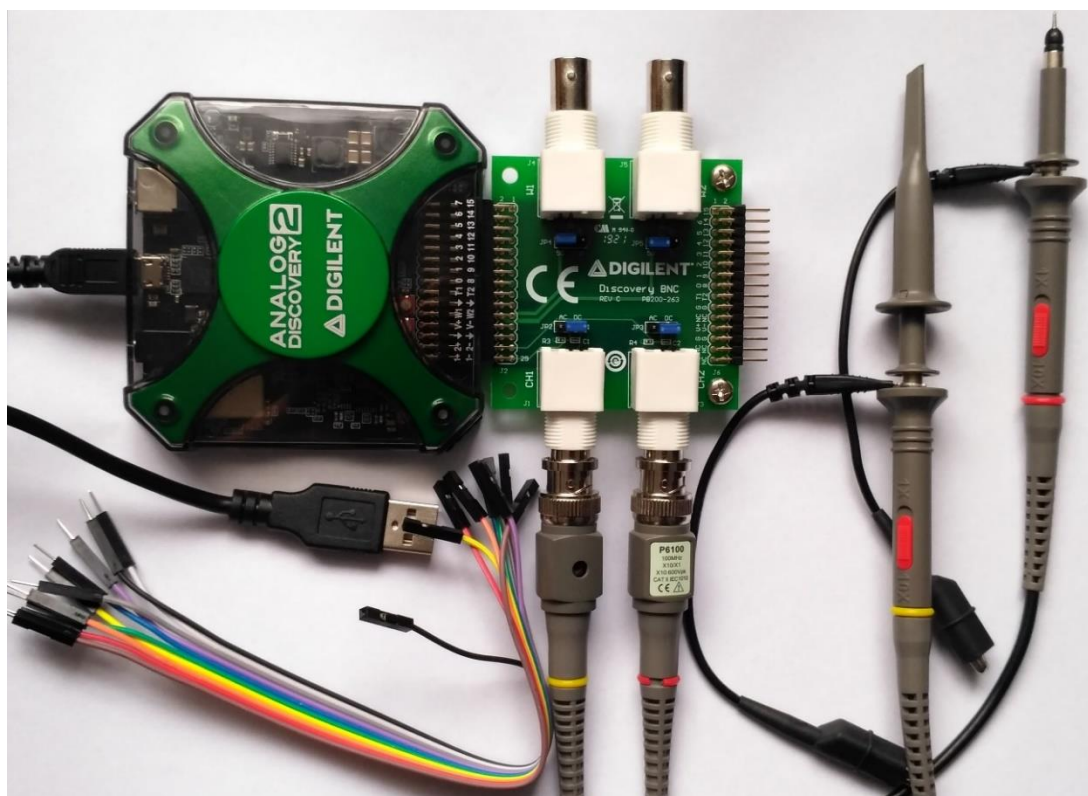
<b>Co</b>		<b>Kde změnit</b>	<b>Popis</b>	
Definice <i>RESULT_NUM</i>		Soubor: <i>DmaBasicDemo.h</i>	Počet měřených hodnot	
Definice <i>TEST</i>		Soubor: <i>Ifx_Cfg_Vadc.h</i>	Použití kanálu CH2 jako vstupní signál pro A/D převod; pokud není definováno, tak se používá CH0 díky funkci ALIAS	
Parametr <i>.inputClass[1].sampleTime</i>		Konfigurace měřicí skupiny Group1, pole <i>IfxVadc_adc_group_config</i>  Soubor: <i>Ifx_Cfg_Vadc.c</i>	Volba doby vzorkování vstupního signálu	
Globální proměnná <i>g_trigPoint</i>		Uvnitř funkce <i>TorqueControl_FocStep</i>  Soubor: <i>AppTorqueControl.c</i>	Nastavení posunu v rámci PWM, okamžik odstartování časovače T13 výstupním signálem TOM1CH8	
Lokální proměnná <i>timerConfig</i> .	<i>base.</i>	<i>t13 Frequency</i>	Uvnitř funkce <i>Ccu6TimerDemo_init</i>  Soubor: <i>Ccu6TimerDemo.c</i>	Volba frekvence, periody a přednastavení časovače T13, který určuje časový okamžik pro spouštění A/D převodu  Pozn.: Přednastavení časovače je nutné změnit i v obsluze přerušení DMA
		<i>t13 Period</i>		
	<i>timer13.co unterValue</i>			

## 5 PRAKTICKÉ OVĚŘENÍ MĚŘICÍ APLIKACE

Tato kapitola se věnuje poslednímu bodu zadání diplomové práce a jsou v ní prezentovány získané výsledky.

Cílem provedených experimentů bylo především ověření správnosti funkce ekvidistantního vzorkování měřeného signálu, jehož realizace byla popsána v předchozí kapitole. Měřicí aplikace byla ale také použita pro měření proudu druhou fází BLDC motoru, čímž se získaly reálné výsledky.

Vstupní signál pro ověřovací měření byl generován prostřednictvím multifunkčního USB přístroje DIGILENT ANALOG DISCOVERY 2, viz Obr. 32. Výhodou tohoto přístroje je, že může současně pracovat i jako osciloskop. Ovládací panely jednotlivých přístrojů se obsluhují z PC prostřednictvím programu WaveForms.



Obr. 32: Multifunkční USB přístroj DIGILENT ANALOG DISCOVERY 2

### 5.1 Testování harmonickým signálem

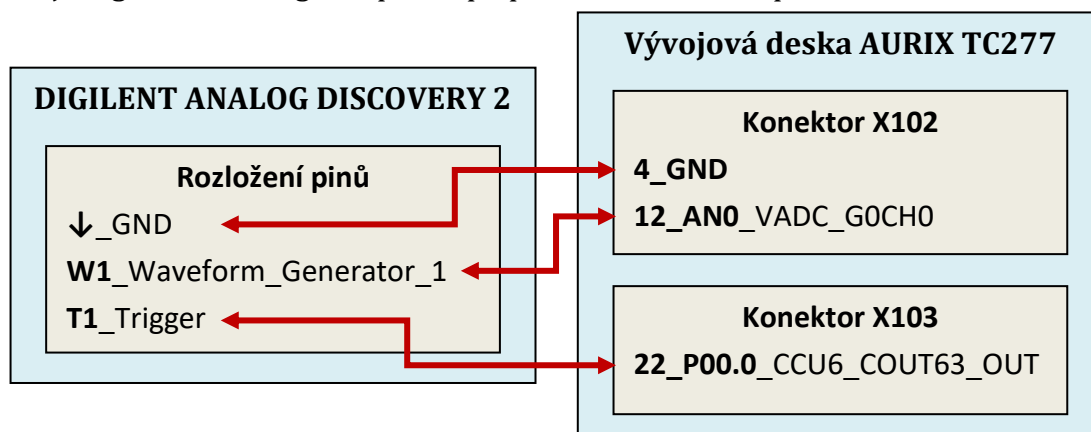
Tato podkapitola prezentuje výsledky měření, které představovaly první výstupy z vytvořené měřicí aplikace a bylo na ně dále navazováno. Cílem provedených měření bylo ověřit její funkčnost, k čemuž byl použit vygenerovaný testovací signál harmonického průběhu. Měřicí aplikace představená v kapitole 4 se během tohoto experimentu lišila tím, že byla naprogramována na výkonnějším MCU TC277 a

nebyla použita vývojová sada s BLDC motorem, tzn. ani ukázková aplikace FOC. Měření bylo sice spouštěno výstupním signálem COUT63\_OUT z periferie CCU6 a probíhalo na jiném vstupním kanálu VADC, ale princip získávání dat zůstal stejný.

Naměřená data jednotlivých měření byla umístěna na CD.

### 5.1.1 Schéma zapojení

Schéma zapojení vývojové desky AURIX TC277 a multifunkčního přístroje DIGILENT ANALOG DISCOVERY 2 zobrazuje Obr. 33. Výstupní signál z prvního generátoru W1 byl přiveden na pin AN0, tedy kanál CH0 skupiny Group0 periferie VADC. Na spouštěcí vstup T1 byl připojen výstupní signál z pinu P00.0, tj. spouštěcí signál pro VADC COUT63\_OUT z periferie CCU6, čehož se dalo využít pro zahájení generování signálu pouze po příchodu tohoto impulsu.



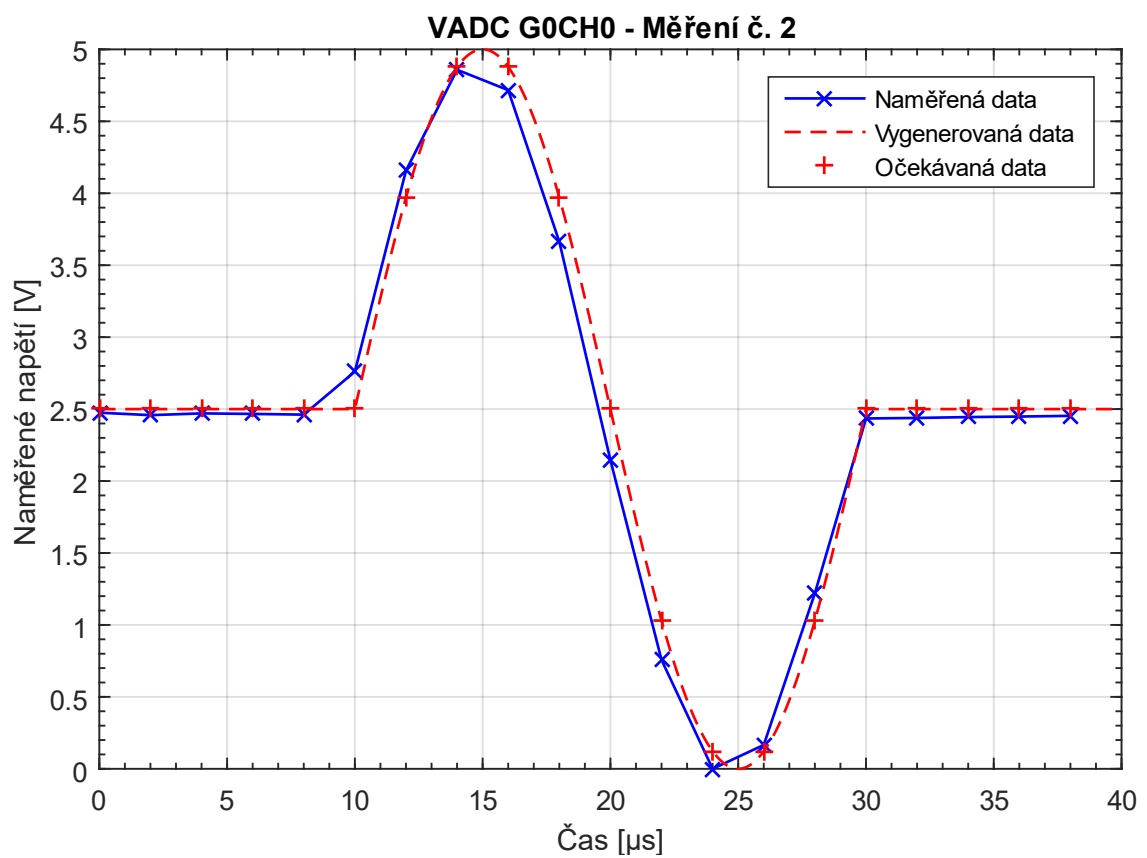
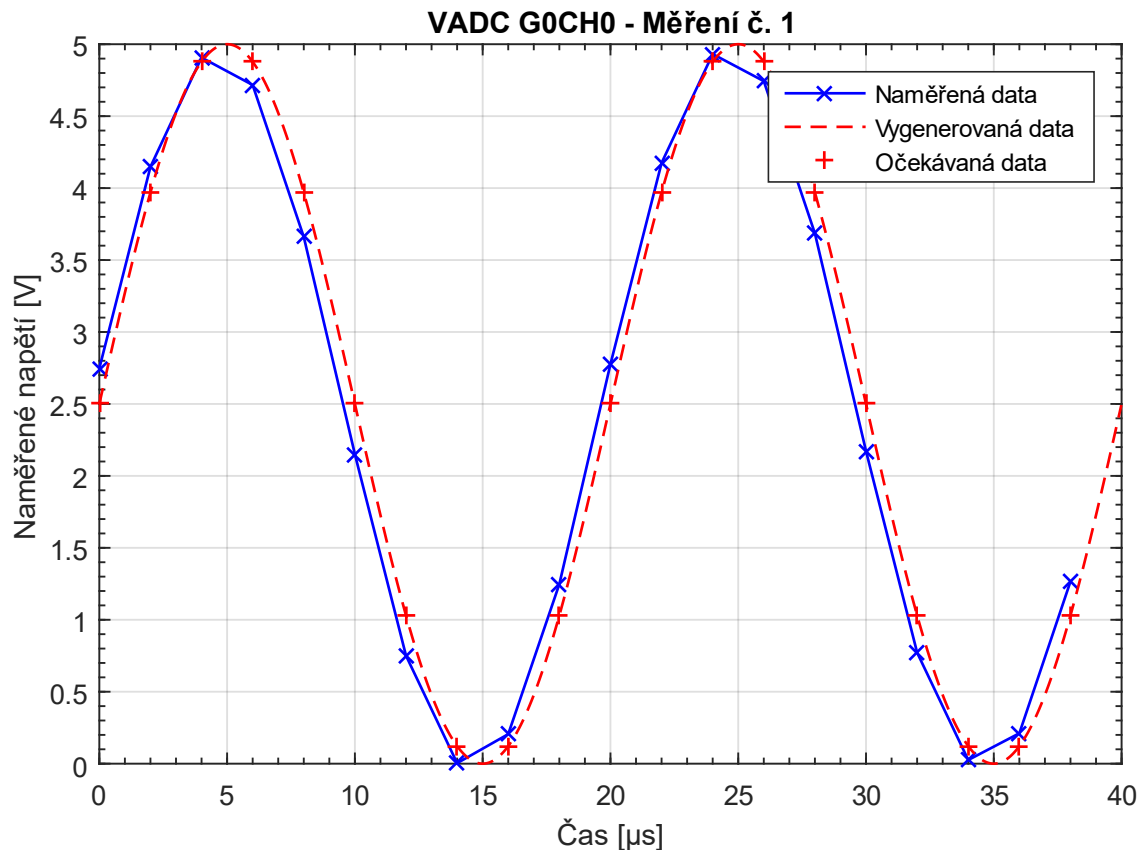
Obr. 33: Schéma zapojení ověřovacího experimentu MCU TC277

### 5.1.2 Získané výsledky

Jako vstupní signál byl pro testování měřicí aplikace použit harmonický signál vygenerovaný multifunkčním přístrojem DIGILENT ANALOG DISCOVERY 2. Jednalo se o signál sinusového průběhu s periodou 20  $\mu$ s o velikosti dané pracovním rozsahem VADC, tj. 0 až 5 V. Vzorkování vstupního signálu bylo nastaveno tak, aby bylo získáno 20 vzorků každé 2  $\mu$ s a doba vzorkování byla 0,5  $\mu$ s. Cílem testovacích měření bylo ověřit, zdali jsou vzorky opravdu získávány s nastaveným časovým krokem, viz Obr. 34.

Během těchto měření byl generátor signálu spouštěn na základě příchodu impulsu z vývojové desky AURIX. Po každém měření byla data manuálně přenesena do PC a následně zpracována výpočetním nástrojem MATLAB.

Přepočítání z digitálního slova na odpovídající hodnotu napětí se provedl na základě znalosti maximálního číslicového rozsahu použitého A/D převodníku (4095, 12bitový) a jeho pracovního rozsahu (5 V). Chyba nuly převodníku nebyla programově kompenzována.



Obr. 34: Získané grafické závislosti – první a druhé měření

### 5.1.3 Zhodnocení

Z vykreslených grafických závislostí pro jednotlivá měření (Obr. 34) je patrné, že vzorkování generovaného signálu probíhalo ekvidistantně každé 2  $\mu\text{s}$ , tak jak bylo očekáváno. Nicméně je zde viditelný časový posun mezi naměřenými a očekávanými hodnotami, jehož hodnota se pohybuje v řádu desetin  $\mu\text{s}$  (průměrně 0,35  $\mu\text{s}$ ). Pro použitý vstupní signál již odpovídající rozdíl ve velikosti vzorků nemusí být zanedbatelný.

Časový posun mohl být způsoben parametry použitých periférií, ale i nedokonalým odstartováním generátoru signálu.

Na výsledky těchto měření bylo při dalším vývoji dále navazováno.

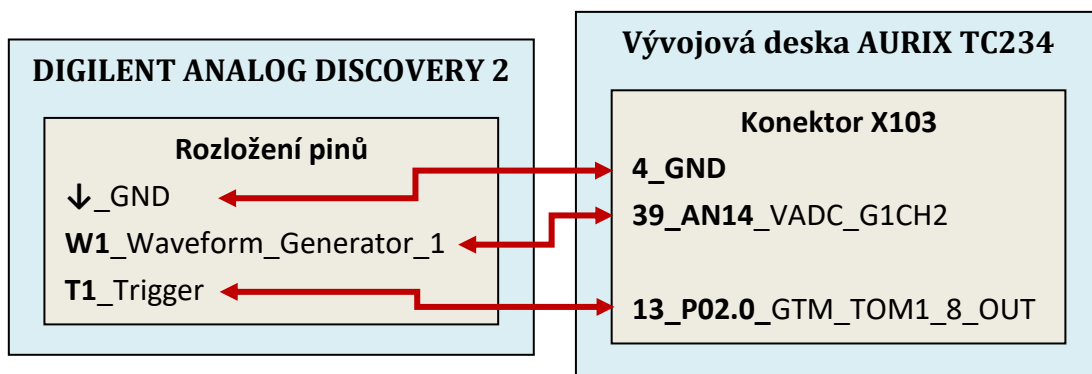
## 5.2 Testování na reálném motoru

Tato podkapitola prezentuje výsledky měření, které byly získány během testování na reálném motoru. Cílem provedených měření bylo ověřit funkčnost vytvořené měřicí aplikace, která byla přidána do ukázkové aplikace FOC. Ověření proběhlo pomocí vygenerovaného testovacího signálu, dokonce bylo vyzkoušeno měření proudu i na reálném BLDC motoru. Představeny jsou dva typy provedených měření – bez znalosti časových značek naměřených hodnot a po jejich zavedení.

Naměřená data jednotlivých měření byla umístěna na CD.

### 5.2.1 Schéma zapojení

Schéma zapojení vývojové desky AURIX TC234 a multifunkčního přístroje DIGILENT ANALOG DISCOVERY 2 zobrazuje Obr. 35. Toto zapojení bylo použito pro ověření funkce aplikace pomocí výstupního signálu z prvního generátoru W1 přivedeného na pin AN14, tedy kanál CH2 skupiny Group1 periferie VADC. Na spouštěcí vstup T1 byl připojen výstupní signál z pinu P02.0, tj. spouštěcí signál pro VADC GTM\_TOM1\_8\_OUT z periferie GTM, čehož se dalo využít pro zahájení generování signálu pouze po příchodu tohoto impulsu.



Obr. 35: Schéma zapojení ověřovacího experimentu MCU TC234

V případě měření proudu na druhé fázi motoru byl vstupním signálem pro A/D převod napěťový signál z výkonové desky eMotor Driver Kit na pinu AN12, tj. kanál CH0 skupiny Group1 periferie VADC. Multifunkční přístroj DIGILENT ANALOG DISCOVERY 2 nebylo nutné pro generování signálu použít.

## 5.2.2 Získané výsledky

Praktické ověření měřicí aplikace bylo učiněno jak na testovacím signálu, tak i na reálném BLDC motoru.

Pro vzorkování signálu se používalo stejné nastavení jako v podkapitole 5.1.2. Počet vzorků byl volen dle potřeby. Po každém měření byla data přenesena do PC přes sériovou linku, kde byla následně vykopírována a zpracována výpočetním nástrojem MATLAB.

Pokud byl vstupem testovací signál, přepočítání z digitálního slova na odpovídající hodnotu napětí se provedl na základě znalosti maximálního číslicového rozsahu použitého A/D převodníku (4095, 12bitový) a jeho pracovního rozsahu (5 V). Chyba nuly převodníku nebyla programově kompenzována.

V případě měření proudu na druhé fázi motoru bylo získané digitální slovo přepočteno na odpovídající hodnotu proudu, přičemž parametry zesílení a posunu pro výpočet byly převzaty z ukázkové aplikace FOC (definice *APPVADC\_CFG\_IMOTOR\_GAIN* a *APPVADC\_CFG\_IMOTOR\_OFFSET*, jejichž počáteční hodnoty jsou nastaveny v hlavičkovém souboru pro konfiguraci periferie VADC *I<sub>fx</sub>\_Cfg\_Vadc.h*). Chyba nuly převodníku je po spuštění aplikace automaticky programově kompenzována.

### 5.2.2.1 Měření a časové značky

Překrytí dvou A/D převodů (pro řídicí algoritmus FOC a měřicí aplikaci) probíhajících na stejném převodníkovém jádře nelze bez časových značek naměřených hodnot spolehlivě rozlišit. Projeví se především ve změně velikosti po sobě jdoucích vzorků a o časovém kroku lze jen těžko rozhodnout, protože nelze předpokládat, že měření probíhalo ve stejných časových okamžicích.

Za účelem předvedení popsaného problému byla provedena dvě různá ověřovací měření, jedno bez znalosti časových značek a druhé po jejich zavedení.

Testovací signály byly vygenerovány multifunkčním přístrojem DIGILENT ANALOG DISCOVERY 2. V prvním případě byl pro vzorkování zvolen lineárně narůstající signál o velikosti dané pracovním rozsahem VADC, tj. 0 až 5 V, s dobou nárustu 50  $\mu$ s. Po uplynutí této doby byla velikost signálu 2,5 V. Ve druhém případě to byl signál pilovitého průběhu s periodou 50  $\mu$ s o velikosti 0,5 až 4,5 V.

Během obou měření byl generátor signálu spouštěn na základě příchodu impulsu z vývojové desky AURIX. Získané grafické závislosti jsou zobrazeny na



Obr. 37, kterému byla z důvodu přehlednosti vyhrazena samostatná stránka v podkapitole 5.2.2.3.

Na prvním grafu je vykresleno naměřené napětí v závislosti na získaných vzorcích. Mezi vzorky 10 a 11 je viditelný skok v naměřené velikosti signálu, díky čemuž se jeho průběh zdá nelineární. Bylo to způsobeno tím, že v tomto časovém okamžiku byl spuštěn A/D převod pro řídicí algoritmus. Tato událost přerušila převod pro měřicí aplikaci po dobu jejího trvání. Nelze tedy spolehlivě vytvořit časovou osu pro toto měření, poněvadž nelze předpokládat, že měření probíhalo ekvidistantně každé 2  $\mu\text{s}$ . To by mohlo být problémové například během dalšího zpracování naměřených dat jiným algoritmem.

Řešení tohoto problému je ukázáno na druhém grafu a spočívá v zavedení časových značek pro naměřené hodnoty. Díky tomu bylo možné vytvořit časovou osu vzhledem k začátku signálu PWM a žádný skok ve velikosti naměřeného signálu již viditelný není. Naměřený průběh odpovídá v lineární části předpokladům.

Dále bylo také ověřeno, zdali naměřený průběh odpovídá vygenerovanému. Ke spuštění generátoru signálu mělo dojít ve stejném okamžiku jako zahájení měření, tj. v čase přibližně 29  $\mu\text{s}$ . Dostupné časové značky nasvědčují tomu, že byl pravděpodobně testovací signál vygenerován dříve asi o 4  $\mu\text{s}$ . Okamžik zahájení vzorkování byl dle časové značky prvního vzorku opožděn vůči spouštěcímu signálu asi o 0,25  $\mu\text{s}$ . Extrapolací naměřených vzorků pomocnou křivkou byl ověřen napěťový rozsah získaného signálu, který vyhovuje vstupnímu signálu.

### 5.2.2.2 Měření na reálném BLDC motoru

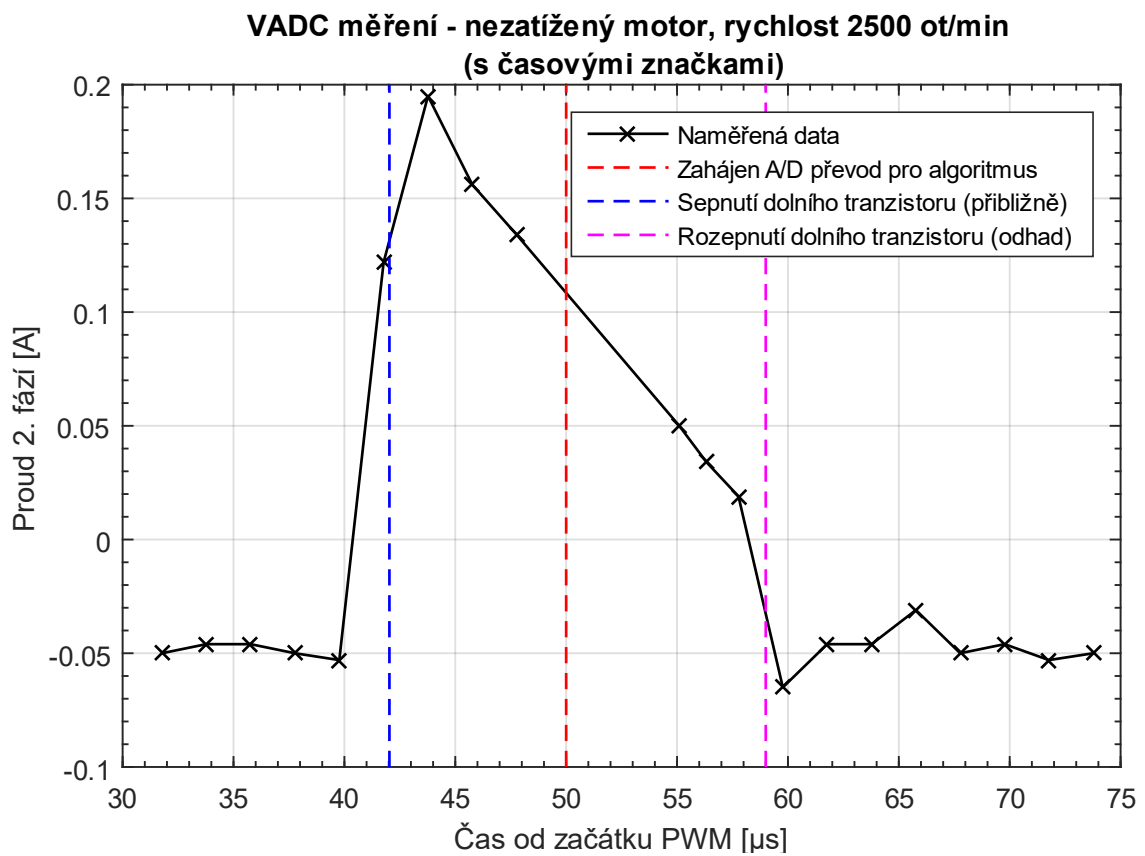
Měřicí aplikace byla také vyzkoušena za účelem získání hodnot proudu druhé fáze reálného BLDC motoru. Provedeny byly dva druhy měření – motor bez zátěže a motor zatížený přidržením hřídelky rukou. Rychlost motoru byla nastavena na 2500 ot/min.

Získané grafické závislosti provedených měření bez časových značek jsou obsahem Obr. 38, kterému byla z důvodu přehlednosti vyhrazena samostatná stránka v podkapitole 5.2.2.3.

Proud druhou fází je na obou grafech vykreslen v závislosti na získaných vzorcích, protože nebyly dostupné jim odpovídající časové značky. Počet vzorků (36) byl zvolen tak, aby se proměřily dva okamžiky, kdy dochází k sepnutí dolního tranzistoru měniče. Okamžik zahájení měření byl zvolen přibližně 10  $\mu\text{s}$  před tímto sepnutím. V oblasti mezi vzorky 1 až 4 a 15 až 28, resp. 14 až 27 (druhý graf), byl dolní tranzistor rozepnut, proto je naměřený proud téměř nulový (průměrně  $-0,05\text{ A}$ ). Ve zbývajících oblastech byl tranzistor sepnut. Přerušování A/D převodu měření pro řídicí algoritmus nastalo v první periodě přibližně mezi vzorky 9 až 11.

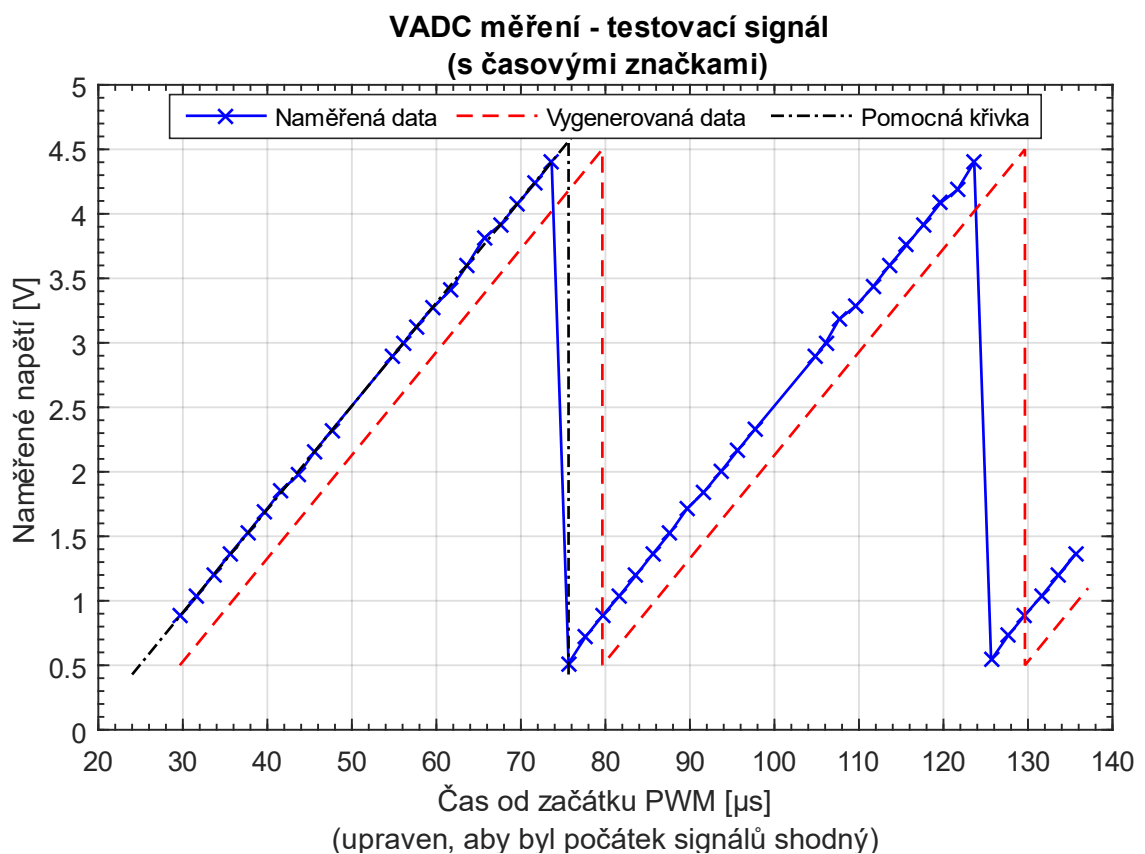
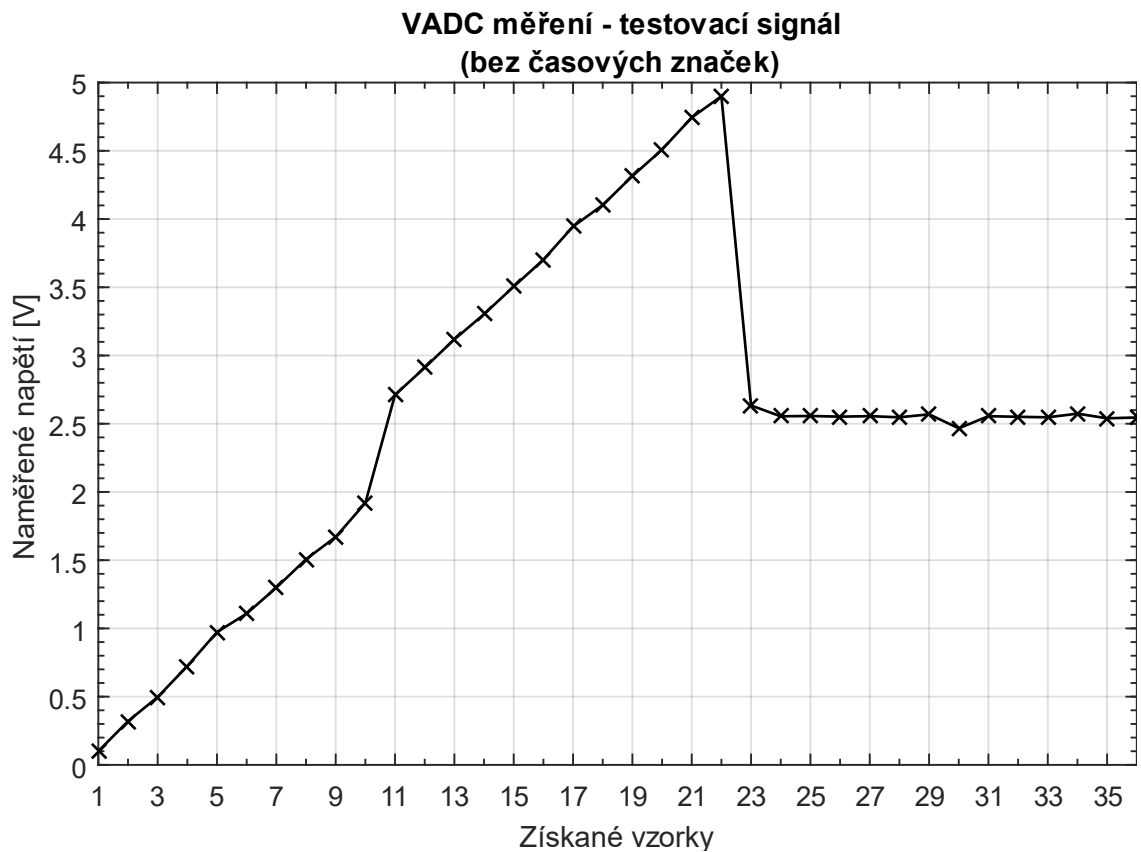
Pokud se porovnájí obě získané grafické závislosti, je patrné, že při zatíženém motoru má naměřený průběh téměř obdélníkový tvar a podstatně vyšší velikost, což není překvapivé.

Více informací o samotném měření poskytlo již zmíněné zavedení časových značek. Příklad průběhu proudu druhou fází v závislosti na čase vzhledem k začátku signálu PWM pro nezatížený motor zobrazuje Obr. 36. V časovém okamžiku přibližně 42  $\mu\text{s}$  došlo k sepnutí dolního tranzistoru, který setrval v tomto stavu odhadem 17  $\mu\text{s}$ . Mezitím byl v čase 50  $\mu\text{s}$  zahájen A/D převod pro řídicí algoritmus. Vykreslený průběh se podobá grafické závislosti uvedené v horní části Obr. 38.

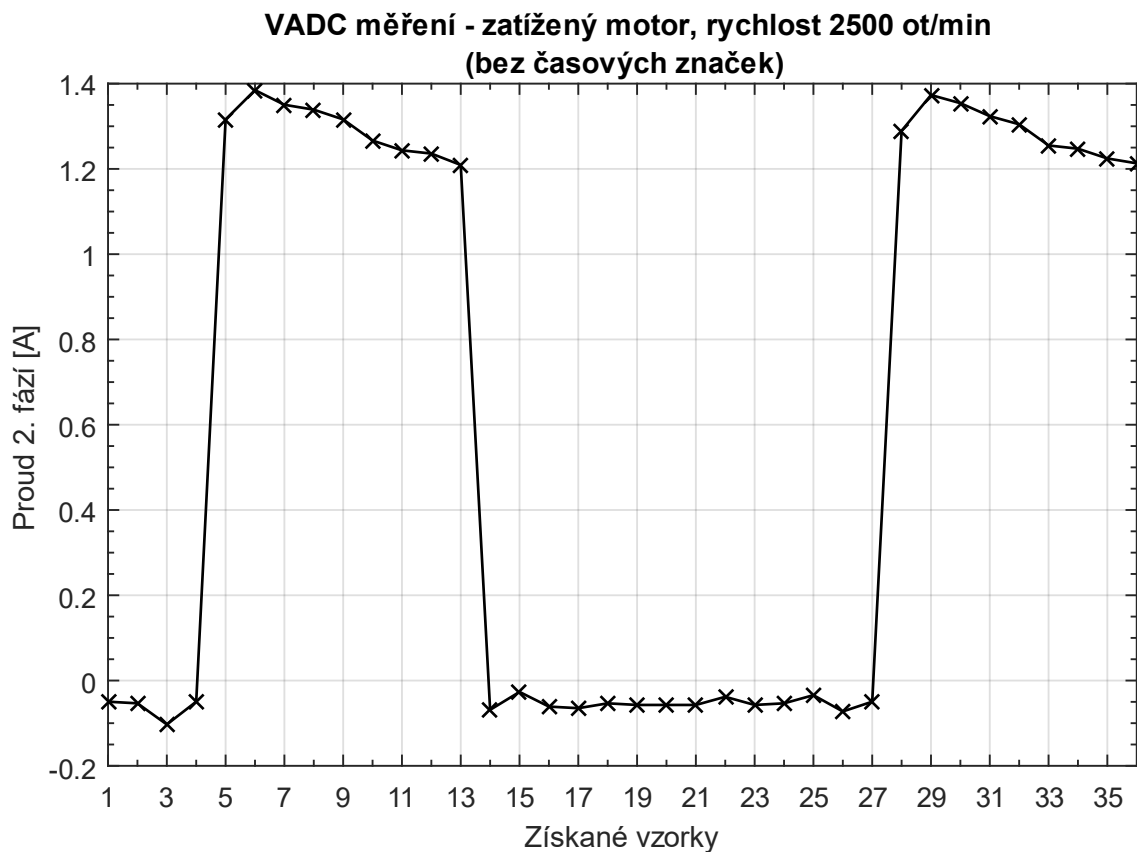
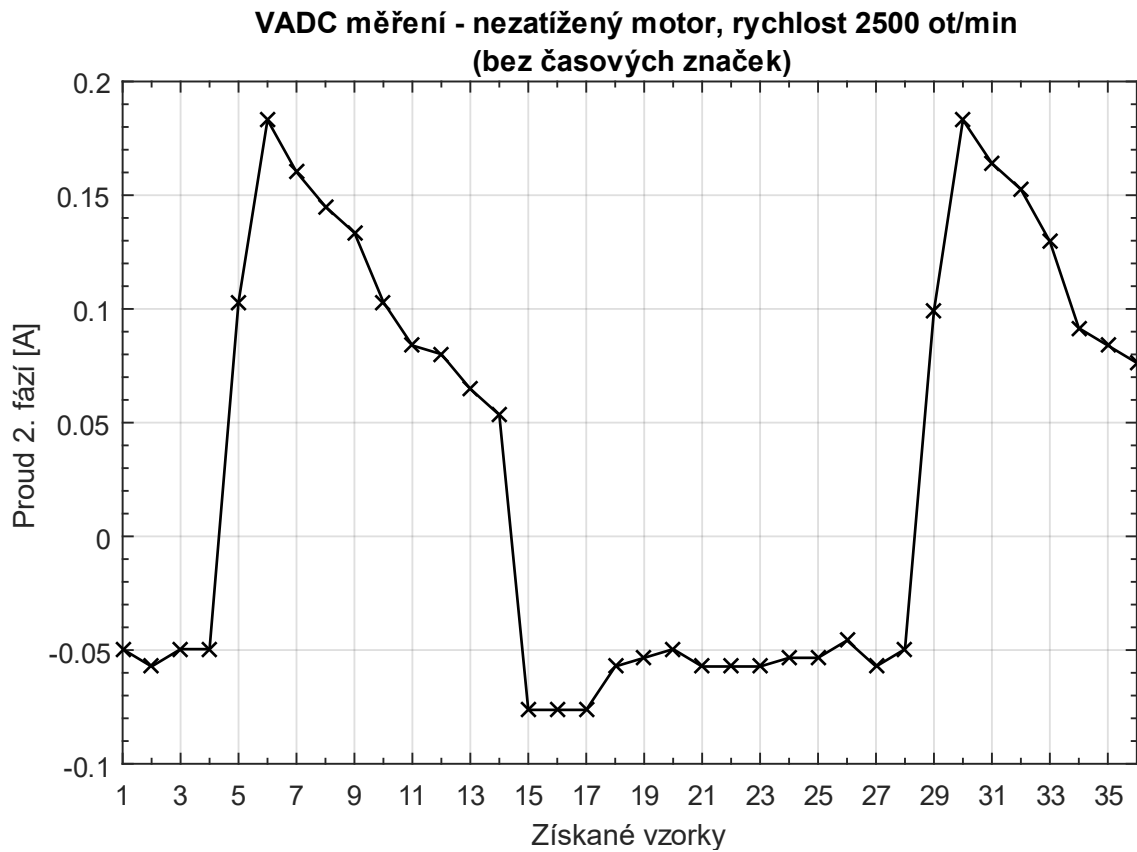


**Obr. 36: Získaná grafická závislost, měření s časovými značkami - nezatížený motor při nastavené rychlosti 2500 ot/min**

### 5.2.2.3 Vykreslené grafické závislosti



Obr. 37: Porovnání získaných grafických závislostí – testovací signál



**Obr. 38: Získané grafické závislosti, měření bez časových značek – nezatížený/zatížený motor při nastavené rychlosti 2500 ot/min**

### 5.2.3 Zhodnocení

Funkčnost měřicí aplikace byla úspěšně ověřena pomocí testovacího signálu. Na vykreslených grafických závislostech (Obr. 37) byla ukázána výhodnost v zavedení časových značek. Z časových důvodů bohužel nebylo možné ověřit, zdali bylo generování signálu uvnitř multifunkčního přístroje DIGILENT ANALOG DISCOVERY 2 opravdu spuštěno dříve. K tomuto účelu by bylo vhodné použít jiný, přesnější generátor testovacího signálu, případně si zobrazit průběhy na jednotlivých výstupech vývojové desky AURIX pomocí osciloskopu.

Provedená testovací měření byla zatížena chybou nuly převodníku, protože nebyla provedena její kompenzace, avšak velikost této chyby se v pracovním rozsahu A/D převodníku 0 až 5 V většinou projevuje v řádu maximálně desetiny V.

Získané časové značky naměřených hodnot poukazují na existenci zpoždění asi 0,25  $\mu\text{s}$  mezi okamžikem spuštění A/D převodu periférií CCU6 a příchodu spouštěcího signálu z periferie GTM.

Časový posun je pravděpodobně způsoben parametry použitých periférií, avšak není vyloučeno, že by se toto zpoždění nedalo odlišnou konfigurací dále zmenšit. Posunutí začátku měření v rámci periody PWM o odpovídající hodnotu ale umožňuje toto zpoždění potlačit.

Měřicí aplikace byla rovněž použita pro reálné měření proudu druhé fáze BLDC motoru. Naměřené průběhy (Obr. 36 a Obr. 38) závisí jednak na okamžiku, kdy je spuštěno měření, ale také na oblasti, ve které je signál PWM právě generován. Měřený proud má totiž sinusový průběh, ale s podstatně delší periodou, než je prováděno měření. Obecně lze říci, že záleží na tom, jak dlouho se dolní tranzistor nachází v sepnutém stavu.

Na obdrženích průbězích nebyly shledány žádné výrazné výkyvy mezi hodnotami jednotlivých vzorků proudu, což nasvědčuje tomu, že použitý BLDC motor pravděpodobně není vhodným kandidátem pro diagnostiku chování tranzistorového měniče.

Zdroje nejistot měření provedených experimentů nebyly vyhodnoceny.

Vhodné by bylo provést další měření i pro jiné počáteční podmínky, než bylo zvoleno v představených experimentech, například s odlišnou dobou vzorkování, časovým krokem mezi jednotlivými vzorky či okamžikem spuštění měření.

## **6 MOŽNOSTI DALŠÍHO VYLEPŠENÍ MĚŘICÍ APLIKACE**

V této kapitole jsou představeny možnosti, jejichž aplikováním by se mohla vytvořená měřicí aplikace dále vylepšit.

### **6.1 Automatizovaný přenos dat z MCU do PC**

Možné vylepšení měřicí aplikace spočívá v rozšíření automatizovaného přenosu naměřených dat z MCU do PC pomocí některé z dostupných komunikačních sběrnic, například využitím již zprovozněné komunikace přes sériovou linku, pomocí níž by se dalo vytvořit propojení s výpočetním nástrojem MATLAB pro automatickou tvorbu získaných grafických závislostí. V současném stavu se musí přenos dat provádět manuálně, což může být pro více hodnot zdlouhavé.

### **6.2 Vytvoření uživatelského prostředí na LCD displeji**

Ovládání měřicí aplikace by se mohlo výrazně zjednodušit vytvořením jednoduchého uživatelského prostředí na dostupném LCD displeji, kde by bylo uživateli umožněno si libovolně nastavit parametry měření, jako jsou například počet naměřených hodnot, okamžik začátku měření atd.

Současné řešení LCD displej nepoužívá. Ten je využíván pouze ukázkovou aplikací FOC. Snaha o vytvoření základního uživatelského prostředí byla realizována prostřednictvím rozhraní Shell, které lze použít pro ovládání MCU ze strany PC pomocí vyslaných příkazů. Toto rozhraní bylo spuštěno na komunikačním portu sériové linky. Implementovány byly ale pouze dva příkazy určené pro zahájení měření a následný výpis hodnot na obrazovku. V tomto řešení by se dalo taktéž pokračovat.

### **6.3 Použití vyšší řady MCU**

Během řešení této práce byl nejprve používán odlišný, výkonnější typ vývojové desky AURIX Application Kit TC277. Vyšší řada MCU by do současného řešení přinesla nové možnosti, jak měřicí aplikaci dále vylepšit.

Propojení vývojové desky osazené MCU TC277 s výkonovou bohužel úspěšné nebylo. Problém se objevil v komunikaci mezi MCU a můstkovým budičem TLE9180, motor se zkrátka nepovedlo roztočit. Dle sdělení vedoucího práce by se musela provést drobná HW změna přímo na vývojové desce. Toto tvrzení bylo ověřeno v loňském akademickém roce, kdy jednu takovou desku upravili a úspěšně zprovoznili. Bohužel se jí však nepodařilo získat k vyzkoušení.

Specifikace MCU TC277 se liší od TC234 především v počtu procesorových jader, který je trojnásobný. Tomu samozřejmě odpovídá i více volné paměti a periférií. Ukázková aplikace FOC i k ní nově přidané měření se spouštějí právě na jednom jádře MCU TC234. Dostupnost druhého jádra by umožnila tyto dvě aplikace od sebe oddělit, a naměřené hodnoty proto například dále, nyní již nezávisle na algoritmu FOC zpracovávat.

Dalším rozdílem je počet A/D převodníků. První je vybaven osmi převodníky, kdežto druhý pouze dvěma. Ukázková aplikace FOC využívá oba dostupné převodníky MCU TC234 k měření proudů, z čehož plyne, že nově přidané měření muselo tento stav respektovat. Pokud by však byly volně k dispozici další převodníky, mohla by se měřicí aplikace realizovat odděleně od měření proudů pro algoritmus FOC. Naměřená data by tak byla konzistentní, protože by byl zaručen ekvidistantní časový krok během jejich získávání. Měření proudu na jedné fázi by se také pravděpodobně mohlo rozšířit i na ostatní fáze, čímž by se získalo mnohem více naměřených dat.

Odečítání časové značky začátku PWM pomocí vstupního podmodulu TIM by se také mohlo zjednodušit. Odstranila by se nutnost spojení odpovídajících pinů vodičem, neboť komponenty modulu GTM MCU TC277 dovolují vnitřně propojit vstupní podmoduly TIM s výstupními TOM. Důvodem je zvýšený počet vstupních podmodulů TIM.

Dále by bylo možné pro získávání časové značky naměřené hodnoty použít událost žádosti o obsluhu od VADC také po uložení výsledku do registru, nikoliv pouze po dokončení převodu kanálu. Vnitřní zapojení signálů MCU TC277 mezi VADC a modulem GTM již toto umožňuje.

## 7 ZÁVĚR

Výsledkem diplomové práce je vytvořená aplikace pro MCU AURIX TC234 umožňující vzorkování signálu s požadavkem na co nejkratší periodu vzorkování a na libovolné nastavení okamžiku zahájení měření. Programové řešení bylo úspěšně přidáno do ukázkové aplikace FOC, díky čemuž mohla být aplikace otestována také na reálném BLDC motoru.

Představené řešení pracuje s perifériemi GTM, CCU6, VADC a DMA. Jeho funkčnost byla ověřena jednak provedením sady měření s testovacím signálem, který byl vygenerován pomocí multifunkčního přístroje DIGILENT ANALOG DISCOVERY 2. Hlavním cílem bylo ověřit správnost ekvidistantního vzorkování měřeného signálu.

Oproti zadání se v diplomové práci pracovalo s BLDC motorem, nikoliv PMSM motorem. S ohledem na to, že se oba motory dají řídit pomocí FOC, v tomto nebyl shledán žádný problém.

Grafické zpracování naměřených dat poukázalo na existenci časového posunu, jehož hodnota se pohybuje v řádu  $\mu\text{s}$ . Toto zpoždění by mohlo být pravděpodobně způsobeno parametry použitých periférií MCU, ale je také možné, že bylo dáno nepřesným generátorem signálu, který nepracoval podle předpokladů.

Navíc byly implementovány časové značky pro naměřené hodnoty, čímž se dokázal potlačit vliv překryvu s A/D převodem pro řídicí algoritmus FOC.

Průběhy naměřených proudů druhé fáze BLDC motoru nepoukázaly na existenci viditelných výkyvů. Tento motor pravděpodobně nebyl vhodným kandidátem pro analýzu chování měniče.

Vzorkovací doba je omezena možnostmi periferie VADC a její minimální hodnota může být  $1 \mu\text{s}$ . Zdroje nejistot ovlivňující měření nebyly vyhodnoceny.

Měřicí aplikaci by bylo vhodné dále otestovat s přesnějším generátorem signálu, což ale nebylo z časových důvodů možné provést. Žádoucí by také bylo provést více měření na reálném motoru s rozšířeným počtem počátečních podmínek pro samotné měření.

V kapitole 6 byly prezentovány možnosti, kterými by bylo možné měřicí aplikaci dále vylepšit. Výrazným vylepšením by bylo použití výkonnějšího typu MCU AURIX TC277, který se bohužel nepovedlo propojit s výkonovou deskou pro řízení BLDC motoru.

Poměrně značnou část času bylo nutné věnovat nastudování funkce ukázkové aplikace FOC, protože ji bylo potřeba převést do nové verze frameworku BIFACES, což se úspěšně podařilo. Díky tomu mohla být měřicí aplikace vytvořena s použitím nových iLLD ovladačů pro periferie MCU.



# Literatura

- [1] FARNELL AN AVNET COMPANY. *Permanent Magnet Synchronous Motor (PMSM)* [online]. [cit. 2019-12-22]. Dostupné z: <https://cz.farnell.com/motor-control-permanent-magnet-sync-motor-pmsm-technology>
- [2] NEBORÁK, Ivo. *Modelování a simulace elektrických regulovaných pohonů*. Ostrava: Vysoká škola báňská – Technická univerzita, 2002. ISBN 80-248-0083-7.
- [3] VESELÝ, Libor. *Algoritmy bezsnímačového řízení synchronního motoru s permanentními magnety* [online]. Brno, 2013 [cit. 2020-05-16]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/41351>. Disertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Pavel Václavek.
- [4] INFINEON TECHNOLOGIES. *Permanent magnet synchronous motor (PMSM)* [online]. [cit. 2019-12-22]. Dostupné z: <https://www.infineon.com/cms/en/applications/solutions/motor-control-and-drives/permanent-magnet-synchronous-motor/#!/products>
- [5] INFINEON TECHNOLOGIES. *Application Note AP32298 – eMotor application example (PMSM with FOC)* [online]. 2019 [cit. 2019-12-22]. K dispozici pouze po registraci a získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.
- [6] INFINEON TECHNOLOGIES. *32-Bit Microcontroller Application Kit TC2X4 User's Manual* [online]. 2014 [cit. 2020-04-12]. Dostupné z: [https://www.infineon.com/dgdl/Infineon-KIT\\_AURIX\\_TC224\\_TFT-UserManual-v01\\_00-EN.pdf?fileId=5546d4626c1f3dc3016c85c5843a7f4c](https://www.infineon.com/dgdl/Infineon-KIT_AURIX_TC224_TFT-UserManual-v01_00-EN.pdf?fileId=5546d4626c1f3dc3016c85c5843a7f4c)
- [7] INFINEON TECHNOLOGIES. *AURIX TC21x/TC22x/TC23x Family 32-Bit Single-Chip Microcontroller User's Manual* [online]. 2014 [cit. 2020-04-12]. Dostupné z: [https://www.infineon.com/dgdl/Infineon-TC21x-TC22x-TC23x-UM-v01\\_01-EN.pdf?fileId=5546d46269bda8df0169ca09bc6023f0](https://www.infineon.com/dgdl/Infineon-TC21x-TC22x-TC23x-UM-v01_01-EN.pdf?fileId=5546d46269bda8df0169ca09bc6023f0)
- [8] INFINEON TECHNOLOGIES. *Application Note AP32376 – eMotor Driver Kit for the AURIX product family* [online]. 2018 [cit. 2020-04-15]. K dispozici pouze po registraci a získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.
- [9] INFINEON TECHNOLOGIES. *iLLD User Documentation, Device: TC23A, Version: 1.0.1.11.0* [online]. 2019 [cit. 2020-04-12]. K dispozici pouze po registraci a získání přístupu k veřejně nepřístupnému obsahu pro MCU řady AURIX.
- [10] NANOTEC. *Katalogový list k BLDC motoru DB42S02* [online]. 2006 [cit. 2020-04-17]. Dostupné z:

<https://en.nanotec.com/fileadmin/files/Datenblaetter/BLDC/DB42/DB42S02.pdf>

- [11] INFINEON TECHNOLOGIES. *TLE9180D-31QK Bridge Driver IC* [online]. 2019 [cit. 2020-04-15]. Dostupné z: [https://www.infineon.com/dgdl/Infineon-TLE9180D-31QK-ProductInformation-v01\\_00-EN.pdf?fileId=5546d4626afcd350016b2d7bab057342](https://www.infineon.com/dgdl/Infineon-TLE9180D-31QK-ProductInformation-v01_00-EN.pdf?fileId=5546d4626afcd350016b2d7bab057342)

# Seznam příloh

<b>Příloha 1</b>	<b>Obsah přiloženého CD .....</b>	<b>67</b>
<b>Příloha 2</b>	<b>Zapojení konektorů .....</b>	<b>68</b>
<b>Příloha 1</b>	<b>Obsah přiloženého CD .....</b>	<b>67</b>
<b>Příloha 2</b>	<b>Zapojení konektorů .....</b>	<b>68</b>

## Příloha 1 Obsah přiloženého CD

Přiložené CD obsahuje:

- elektronickou podobu této práce:
  - složka *DP\_Buchal\_2020\_Text*,
- vytvořený a přeložený projekt pro překladač HighTec Free TriCore Entry Toolchain pro MCU TC234 bez měřicí aplikace, tj. pouze převedenou ukázkovou aplikaci FOC:
  - složka *DP\_Buchal\_2020\_HighTecProjekt\_FOC*,
- vytvořený a přeložený projekt pro překladač HighTec Free TriCore Entry Toolchain pro MCU TC234 s měřicí aplikací přidanou do ukázkové aplikace FOC:
  - složka *DP\_Buchal\_2020\_HighTecProjekt\_FOC\_MeasApp*,
- naměřená data v rámci praktického ověření měřicí aplikace:
  - složka *DP\_Buchal\_2020\_NamerenaData*,
- elektronickou podobu schématu zapojení konektorů vývojové sady s BLDC motorem, viz Příloha 2:
  - složka *DP\_Buchal\_2020\_SchemaKonektory*.

## Příloha 2 Zapojení konektorů

### ZAPOJENÍ KONEKTORŮ X102 A BU6, X103 A BU7

Konektor BU6 připojen na konektor X102

Konektor X102 TC234 Application Kit					
Konektor BU6 eMotor Drive Kit					
VCC_IN	VCC_IN	1	2	VEXTA	+3V3
GND	GND	3	4	GND	GND
AN7	CH2NA	5	6	CH2PA	AN6
AN5	n.c.	7	8	n.c.	AN4
AN3	CH0NA	9	10	CH0PA	AN2
AN1	VRO	11	12	VO1	AN0
P33.5	PFB1	13	14	n.c.	P33.4
P33.3	n.c.	15	16	n.c.	P33.2
P33.1	PFB1_EN	17	18	PFB3	P33.12
P33.8	n.c.	19	20	PFB2	P33.6
P34.0	n.c.	21	22	n.c.	P34.1
P34.2	n.c.	23	24	n.c.	P34.3
P22.4	n.c.	25	26	n.c.	P33.11
P22.0	MOSI	27	28	MISO	P22.1
P22.2	CSN	29	30	CLK_SPI	P22.3
P15.2	n.c.	31	32	n.c.	P15.3
P15.4	/ERR	33	34	n.c.	P15.5
P15.6	n.c.	35	36	n.c.	P15.7
P20.9	/SOFF	37	38	ENA	P20.10
P14.4	n.c.	39	40	n.c.	P14.5

Konektor BU7 připojen na konektor X103

Konektor X103 TC234 Application Kit					
Konektor BU7 eMotor Drive Kit					
VCC_IN	VCC_IN	1	2	VEXTB	+3V3
GND	GND	3	4	GND	GND
P33.10	n.c.	5	6	n.c.	P33.9
P14.8	n.c.	7	8	n.c.	P14.7
P14.6	n.c.	9	10	/INH	P10.6
P20.2	n.c.	11	12	n.c.	P11.8
P02.0	CGPWM_N	13	14	CGPWM_P	P02.1
P02.2	n.c.	15	16	HALLA	P02.3
P02.4	HALLB	17	18	HALLC	P02.5
P02.6	ENC_A	19	20	ENC_B	P02.7
P02.8	ENC_Z	21	22	n.c.	P00.0
P00.1	n.c.	23	24	IL1	P00.2
P00.3	/IH1	25	26	IL2	P00.4
P00.5	/IH2	27	28	IL3	P00.6
P00.7	/IH3	29	30	n.c.	P00.8
P00.9	n.c.	31	32	n.c.	P21.4
P21.5	n.c.	33	34	n.c.	P00.12
AN17	n.c.	35	36	n.c.	AN16
AN15	n.c.	37	38	VO3	AN13
AN14	n.c.	39	40	VO2	AN12

Pozn.: n.c. = nezapojeno

Popis signálů – konektory X102 a BU6:

Číslo pinu	Konektor X102 TC234 Application Kit	Konektor BU6 eMotor Drive Kit
1		Supply Input TriBoard (5,5 V ... 50 V)
2		n.c.
3 až 4		Ground
5	VADC GOCH7	Analog input / SIN Secondary coil – Resolver VCC/2
6	VADC GOCH6	Analog input / SIN Secondary coil – Resolver SIN
7	VADC GOCH5	

<b>Číslo pinu</b>	<b>Konektor X102 TC234 Application Kit</b>	<b>Konektor BU6 eMotor Drive Kit</b>
8	VADC GOCH4	
9	VADC GOCH3	Analog input / COS Secondary coil – Resolver VCC/2
10	VADC GOCH2	Analog input / COS Secondary coil – Resolver COS
11	VADC GOCH1	TLE9180 Output of reference voltage of differential amplifier
12	VADC GOCHO	TLE9180 Output of differential 1 amplifier for shunt signal amplification
13		TLE9180 Phase feedback of motor connection phase 1
14 až 16		
17		Enable PFB1
18		TLE9180 Phase feedback of motor connection phase 3
19		
20		TLE9180 Phase feedback of motor connection phase 2
21 až 26		
27	QSPI3 MTSR	TLE9180 SPI Master Out, Slave In
28	QSPI3 MRST	TLE9180 SPI Master In, Slave Out
29	QSPI3 SLSO12	TLE9180 Chip Select
30	QSPI3 SCLK	TLE9180 SPI clock input
31 až 32		
33		TLE9180 Error signal
34 až 36		
37		TLE9180 Independent safe state switch off
38		TLE9180 Enable pin
39 až 40		

Popis signálů – konektory X103 a BU7:

<b>Číslo pinu</b>	<b>Konektor X103 TC234 Application Kit</b>	<b>Konektor BU7 eMotor Drive Kit</b>
1		Supply Input TriBoard (5,5 V ... 50 V)
2		n.c.
3 až 4		Ground
5 až 9		
10		TLE9180 Inhibit pin
11 až 12		
13		Primary coil
14		Primary coil
15		
16		HALL A
17		HALL B
18		HALL C
19	GPT12 T3INA	Encoder A

<b>Číslo pinu</b>	<b>Konektor X103 TC234 Application Kit</b>	<b>Konektor BU7 eMotor Drive Kit</b>
20	GPT12 T3EUDA	Encoder B
21	GPT12 T4INA	Encoder Top Zero
22 až 23		
24	GTM TOM1CH1	TLE9180 PWM Low-side switch 1
25	GTM TOM1CH2	TLE9180 PWM High-side switch 1
26	GTM TOM1CH3	TLE9180 PWM Low-side switch 2
27	GTM TOM1CH4	TLE9180 PWM High-side switch 2
28	GTM TOM1CH5	TLE9180 PWM Low-side switch 3
29	GTM TOM1CH6	TLE9180 PWM High-side switch 3
30 až 34		
35	VADC G1CH5	
36	VADC G1CH4	
37	VADC G1CH3	
38	VADC G1CH1	TLE9180 Output of differential 3 amplifier for shunt signal amplification
39	VADC G1CH2	
40	VADC G1CH0	TLE9180 Output of differential 2 amplifier for shunt signal amplification

Tab. 6: Tabulka naměřených hodnot testovacího signálu s časovými značkami vztaženými k začátku PWM

<b>PWM TIMESTAMP</b>	<b>-RAW</b>	22576	<b>-usec:</b>	22,58	
<b>TRIGGER POINT</b>	<b>-RAW:</b>	2964	<b>-usec:</b>	29,64	
<b>ADC SAMPLE</b>	<b>TIMESTAMP</b>	<b>TIME-usec</b>	<b>dT-usec</b>	<b>VALUE- RAW</b>	<b>VALUE-V</b>
0	25687	31,11	31,11	721	0,880
1	25887	33,11	2,00	845	1,032
2	26087	35,11	2,00	985	1,203
3	26287	37,11	2,00	1120	1,368
4	26487	39,11	2,00	1249	1,525
5	26687	41,11	2,00	1381	1,686
6	26887	43,11	2,00	1518	1,853
7	27087	45,11	2,00	1618	1,976
8	27287	47,11	2,00	1769	2,160
9	27487	49,11	2,00	1903	2,324
10	28208	56,32	7,21	2373	2,897
11	28331	57,55	1,23	2454	2,996
12	28487	59,11	1,56	2560	3,126
13	28687	61,11	2,00	2684	3,277
14	28887	63,11	2,00	2791	3,408
15	29087	65,11	2,00	2950	3,602
16	29287	67,11	2,00	3122	3,812
17	29487	69,11	2,00	3210	3,919
18	29687	71,11	2,00	3343	4,082
19	29887	73,11	2,00	3468	4,234
20	30087	75,11	2,00	3606	4,403
21	30287	77,11	2,00	422	0,515
22	30487	79,11	2,00	590	0,720
23	30687	81,11	2,00	723	0,883
24	30887	83,11	2,00	849	1,037
25	31087	85,11	2,00	982	1,199

26	31287	87,11	2,00	1119	1,366
27	31487	89,11	2,00	1250	1,526
28	31687	91,11	2,00	1402	1,712
29	31887	93,11	2,00	1509	1,842
30	32087	95,11	2,00	1638	2,000
31	32287	97,11	2,00	1773	2,165
32	32487	99,11	2,00	1904	2,325
33	33208	106,32	7,21	2370	2,894
34	33331	107,55	1,23	2459	3,002
35	33487	109,11	1,56	2606	3,182
36	33687	111,11	2,00	2692	3,287
37	33887	113,11	2,00	2812	3,433
38	34087	115,11	2,00	2950	3,602
39	34287	117,11	2,00	3077	3,757
40	34487	119,11	2,00	3209	3,918
41	34687	121,11	2,00	3345	4,084
42	34887	123,11	2,00	3428	4,186
43	35087	125,11	2,00	3605	4,402
44	35287	127,11	2,00	443	0,541
45	35487	129,11	2,00	597	0,729
46	35687	131,11	2,00	728	0,889
47	35887	133,11	2,00	851	1,039
48	36087	135,11	2,00	983	1,200
49	36287	137,11	2,00	1117	1,364