

Bakalářská práce

ROZHRANÍ PRO MĚŘENÍ A ZPRACOVÁNÍ INFORMACE Z OTÁČKOVÉHO A MOMENTOVÉHO ČIDLA

Petr Janoušek



Praha 2021

Vedoucí práce: Ing. Jan Bauer, Ph.D.

České vysoké učení technické v Praze

Fakulta elektrotechnická, katedra elektrických pohonů a trakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Janoušek** Jméno: **Petr** Osobní číslo: **483627**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra elektrotechnologie**
Studijní program: **Elektrotechnika, energetika a management**
Specializace: **Aplikovaná elektrotechnika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Rozhraní pro měření a zpravování informace z otáčkového a momentového čidla

Název bakalářské práce anglicky:

Interface for Rotational Speed and Motor Torque Measurement

Pokyny pro vypracování:

- 1) Seznamte se se způsoby měření rychlosti otáčení pomocí kvadrurního signálu
- 2) Seznamte se se způsobem měření a vyhodnocení síly pomocí tenzometrického snímače
- 3) Pro kit NUCLEO navrhnete bloky kódu pro zpracování informací z čidla otáček a senzoru síly
- 4) Navrhnete HW pro digitalizaci měření otáček a momentu na pracovištích v laboratoři H26
- 4) Navržené bloky kódu ověřte na navrženém HW

Seznam doporučené literatury:

Dodá vedoucí

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jan Bauer, Ph.D., katedra elektrických pohonů a trakce FEL

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

Datum zadání bakalářské práce: **24.01.2021**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Jan Bauer, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

PODĚKOVÁNÍ

Rád bych tímto poděkoval svému vedoucímu práce Ing. Janu Bauerovi, Ph.D. za velmi kvalitní vedení, cenné tipy, doporučení, rady a ochotu pomoci při psaní této práce. Dále bych rád poděkoval své rodině a přítelkyni za nehasnoucí podporu a pomoc během psaní této práce. V neposlední řadě bych rád poděkoval také doc. Ing. Janu Fischerovi, CSc. za jeho cenné rady, poskytnuté studijní materiály a testovací hardware, které mi značně usnadnily celý proces zpracování této práce.

PROHLÁŠENÍ

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

V Praze dne

.....

ABSTRAKT

Práce se zabývá vývojem jednotky pro zpracování signálů z kvadrurního senzoru a senzoru síly pro měření momentu a otáček motoru. Jednotka nabízí možnost tyto hodnoty zobrazit na sedmsegmentovém displeji nebo v analogové podobě pomocí osciloskopu či voltmetru. Jednotka je modulární a je řízena dvěma procesory z řady STM32 od STMicroelectronics.

Klíčová slova: ČVUT FEL, katedra elektrických pohonů a trakce, bakalářská práce, STM32, kvadrurní enkodér, tenzometr

ABSTRACT

This work shows the development of the control unit for measuring rotating speed and torque of motor. The unit also offers the possibility to display these values on a segment display or in analog form using an oscilloscope or voltmeter. The unit is modular and controlled by two CPUs STM32 manufactured by STMicroelectronics

Keywords: CTU FEE, Department of Electric Drives and Traction, bachelor's thesis, STM32, quadrature encoder, strain load cell

OBSAH

ÚVOD.....	4
KAPITOLA 1: TEORETICKÝ ROZBOR.....	5
KAPITOLA 2: SNÍMAČE SÍLY	7
2.1 HYDRAULICKÝ SNÍMAČ SÍLY	7
2.2 PNEUMATICKÝ SNÍMAČ SÍLY.....	8
2.3 TENZOMETRICKÝ SNÍMAČ SÍLY	8
2.3.1 Princip odporového tenzometru	8
2.3.2 Rozdělení odporových tenzometrů	9
2.3.3 Konstrukce tenzometrického senzoru síly.....	10
KAPITOLA 3: VYHODNOCOVÁNÍ SIGNÁLU ZE SNÍMAČE SÍLY	11
3.1 TENZOMETRICKÝ MŮSTEK.....	11
3.2 MODUL PRO VÁHOVÉ SENZORY S HX711	13
KAPITOLA 4: SNÍMAČE OTÁČIVÉ RYCHLOSTI.....	15
4.1 POPIS FUNKCE INKREMENTÁLNÍHO ENKODÉRU	16
4.2 VZNIK KVADRATURNÍHO SIGNÁLU V INKREMENTÁLNÍM ENKODÉRU	16
4.3 TYPY ZPRACOVÁNÍ KVADRATURNÍHO SIGNÁLU.....	18
4.3.1 Zpracování X1.....	18
4.3.2 Zpracování X2.....	18
4.3.3 Zpracování X4.....	19
4.4 ABSOLUTNÍ ENKODÉR.....	19
KAPITOLA 5: VYHODNOCENÍ SIGNÁLU ZE SENZORU OTÁČEK.....	20
5.1 VYHODNOCENÍ KVADRATURNÍHO SIGNÁLU.....	21
5.1.1 Integrovaný obvod v SLG46140V.....	21
5.1.2 Periferie TIM v procesorech STM32	21
5.1.3 Výpočet otáčivé rychlosti.....	26
KAPITOLA 6: ZOBRAZENÍ DAT	29
6.1 MULTIPLEX SEMISEGMENTOVÝCH LED DISPLEJE	30
6.2 ANALOGOVÉ VÝSTUPY	33
6.3 SPI V PROCESORECH STM32	33
ZÁVĚR	36
KAPITOLA 7: LITERATURA.....	38

SEZNAM OBRÁZKŮ

Obr. 1.1. Blokové schéma navrženého zařízení	5
Obr. 1.2. Vývojový diagram kódu v bloku Master	6
Obr. 1.3. Vývojový diagram kódu v bloku Slave.....	6
Obr. 2.1. Hydraulický snímač síly, [13], vlastní zpracování.....	7
Obr. 2.2. Pneumatický snímač síly, [13], vlastní zpracování	8
Obr. 2.3. Přehled provedení odporových tenzometrů, [2], vlastní zpracování.....	9
Obr. 2.4. Tenzometrický sensor síly užitý v rámci testování	10
Obr. 2.5. Simulace statické analýzy mechanického napětí.....	10
Obr. 3.1. Kompensace vlivu teploty pasivním tenzometrem, [4], vlastní zpracování.....	11
Obr. 3.2. Poloviční tenzometrický můstek, [4], vlastní zpracování	12
Obr. 3.3. Úplný tenzometrický můstek, [4], vlastní zpracování	12
Obr. 3.4. Pinout obvodu HX711, [6], vlastní zpracování	13
Obr. 3.5. Přehled informací o vstupních kanálech HX711, [6], upraveno	13
Obr. 3.6. Reálné provedení modulu s HX711	14
Obr. 4.1. Kvadrurní signál, jednostopý sensor, převzato z [14].....	16
Obr. 4.2. Princip dvoustopého senzoru, převzato z [15], upraveno	16
Obr. 4.3. Kvadrurní enkodér s nulovou stopou, převzato z [20]	17
Obr. 4.4. Kvadrurní signál pro dva různé směry otáčení	17
Obr. 4.5. Zpracování X1.....	18
Obr. 4.6. Zpracování X2.....	18
Obr. 4.7. Zpracování X4.....	19
Obr. 4.8. Provedení optických stop s různými typy kódování, převzato z [19], upraveno	19
Obr. 5.1. Zapojení více SLG46140V obvodů na SPI sběrnici, převzato z [9], upraveno	21
Obr. 5.2. Blokové schéma struktury časovače TIM2, převzato z [11].....	22
Obr. 5.3. Příklad procesu vyhodnocení kvadrurního signálu, převzato z [11]	23
Obr. 5.4. Stavový diagram kvadrurního signálu, převzato z [16].....	24
Obr. 5.5. Interface prostředí STM32CubeIDE	25
Obr. 5.6. Interface pro nastavení parametrů TIM2 v prostředí STM32CubeIDE	25
Obr. 5.7. Předpokládaný průběh kvadrurního signálu	26
Obr. 5.8. Nastavení periferie TIM16 v STM32CubeIDE	27
Obr. 5.9. Podoba záložky Clock Configuration.....	27
Obr. 6.1. Pinout shiftregistru 74HC595N, převzato z [18]	30
Obr. 6.2. Blokové schéma multiplex obvodu v displejové jednotce	30
Obr. 6.3. Část schématického zapojení multiplexu displeje s 74HC595N a tranzistorovými poli	31
Obr. 6.4. Osciloskopem naměřené časování při řízení shift registru 74HC595N	32
Obr. 6.5. Zapojení pro full-duplex SPI, [12], vlastní zpracování.....	34
Obr. 6.6. Zapojení pro Simplex SPI, [12], vlastní zpracování.....	34

Obr. 6.7. Zapojení pro half-duplex SPI, [12], vlastní zpracování	35
Obr. 6.8. Nastavení zařízení Slave	35
Obr. 0.1. Fotografie prototypu	36

ÚVOD

Jedny z nejdůležitějších charakteristik elektrických motorů jsou charakteristika momentová a rychlostní. Právě pro svou důležitost je jejich měření součástí laboratorních úloh v některých předmětech zajištěných katedrou elektrických pohonů a trakce na ČVUT FEL. Jak již z názvu charakteristik plyne, neobejdou se bez přesného měření momentu a otáčivé rychlosti. Cílem této práce je navrhnout jednotku pro měření a indikaci těchto veličin a tím digitalizovat, zpřesnit a zrychlit způsob jejich měření.

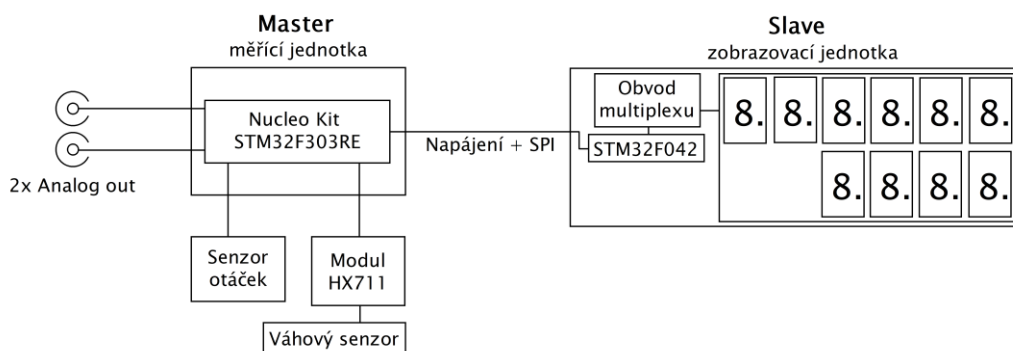
Jednotka nahradí aktuální řešení, které je v případě měření otáček založené na tachodynamu a voltmetru, a na dynamometru a mechanické váze v případě měření momentu. Mechanická váha bude nahrazena elektronickým tenzometrem a tachodynamo kvadraturním enkodérem. Teoretickému úvodu a přehledu do problematiky těchto senzorů jsou věnovány kapitoly s názvy *Snímače síly* a *Snímače otáčivé rychlosti*.

Naměřená a zpracovaná data budou zobrazována pomocí deseti sedmisegmentových LED displejů a dvou analogových výstupů vyvedených na BNC konektory pro připojení například osciloskopu. Jedním z hlavních požadavků je, aby LED displej byl viditelný z velké vzdálenosti i při vysoké ozářenosti laboratoře slunečním svitem. Z toho důvodu bude nutné užít velké displeje s dostatečným jasem – například tedy displeje sedmisegmentové. Velký počet sedmisegmentových displejů si rovněž vyžádá návrh hardwaru, který omezí potřebný počet výstupních pinů mikroprocesoru na minimum – tento navržený hardware je popsán v kapitole 6.1

Vzhledem k prostorovému uspořádání laboratoře elektrických pohonů na ČVUT FEL bude nutné rozdělit celé zařízení na dvě části – LED displej, který bude umístěn na viditelném místě a měřicí část, která bude umístěna níž, a bude obsahovat vstupy pro příslušné senzory, výstupy s BNC konektory a sériový výstup do části s LED displejem. Tyto dvě části budou řízeny dvěma nezávislými mikroprocesory a vzájemně komunikovat po SPI.

KAPITOLA 1: TEORETICKÝ ROZBOR

Tato práce je zaměřena na hardwarový a softwarový vývoj jednotky pro měření a zobrazení otáčivé rychlosti a momentu. S návrhem hardwaru se pojí relativně velké množství rozhodování, myšlenkových postupů a voleb vhodných řešení. V této kapitole budou jednotlivá rozhodnutí a volby komponent odůvodněny. Kapitola také vytvoří přehled bloků užitých pro realizaci. Tyto bloky jsou vidět na blokovém schématu na Obr. 1.1, s jehož pomocí bude jednotka v rámci této kapitoly popsána z hlediska architektury.



Obr. 1.1. Blokové schéma navrženého zařízení

Jako senzor otáček byl v rámci testování užit kvadrurní enkodér magnetického a optického typu. V reálném provedení bude užit enkodér optický, který je v laboratoři k dispozici. Měření momentu je řešeno pomocí váhového senzoru a dynamometru. Testovací váhový senzor se od senzoru dostupného v laboratoři liší pouze mohutností nosné konstrukce, a tedy maximálním možným zatížením. Užití těchto senzorů rovněž plyne ze zadání. Detailnějšímu teoretickému popisu obou senzorů se věnují kapitoly 2 a 4. Informace o zpracování signálů z nich pak jsou v kapitolách 3 a 5.

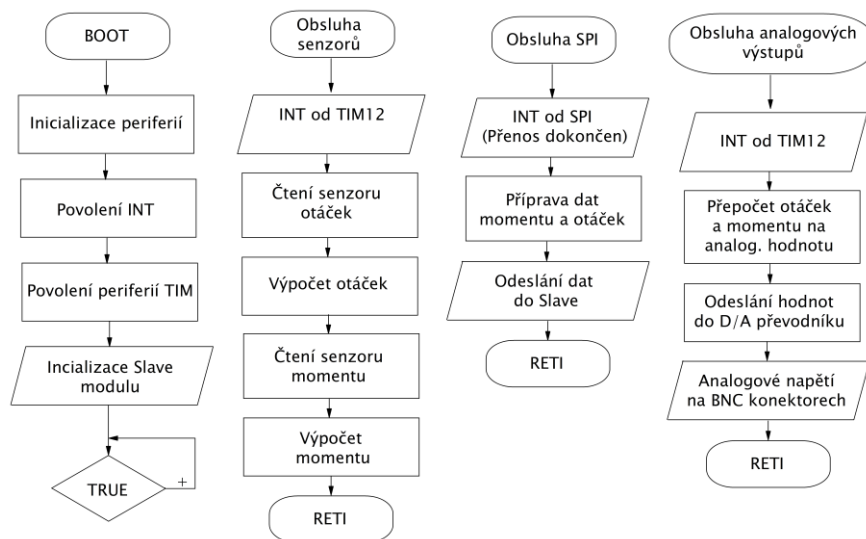
Při volbě displejů byla snaha o dosažení maximální čitelnosti i na velkou vzdálenost a zároveň dosažení co nejnižší ceny. Z toho důvodu byly zvoleny sedmisegmentové displeje, které jsou relativně cenově dostupné i ve velkých rozměrech. Pro řízení těchto displejů byl navržen multiplex obvod řízený procesorem se shiftregistery a tranzistorovými poli s cílem snížit množství řídicích vodičů na minimum. Navržené zapojení je výhodné svou snadnou rozšiřitelností například o další řádky displeje a detailně se mu věnuje kapitola 6.1.

Jelikož bylo nutné displeje umístit na viditelné místo, ale zároveň umožnit snadný přístup ke zbytku jednotky, byla jednotka rozdělena na dvě části (Master a Slave). Toto rozdělení zároveň vedlo k úplnému osamostatnění zobrazovací jednotky (Slave) s vidinou možného užití této jednotky v dalších projektech katedry. Její součástí je tedy i procesor, který bude data přijímat po sériové sběrnici SPI a zajišťovat multiplex displejů. SPI se detailněji věnuje kapitola 6.3.

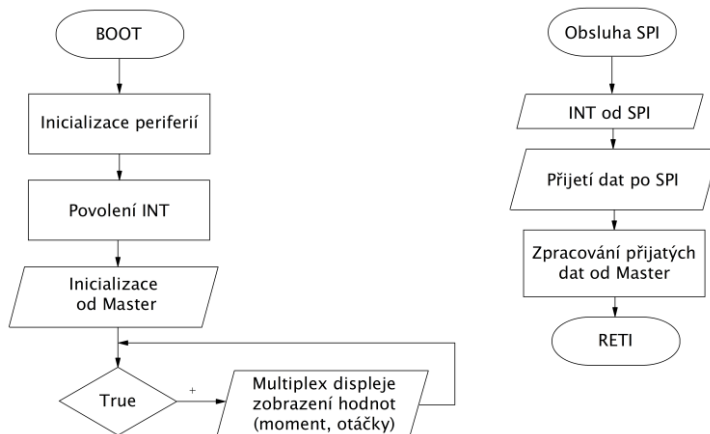
Jako mikroprocesor v bloku Master byl dle zadání bakalářské práce zvolen Nucleo-kit STM32F303RE. Mikroprocesor pro blok Slave byl zvolen ze stejné rodiny – STM32, konkrétně STM32F042. STM32 jsou 32bitové mikroprocesory od STMicroelectronics založené na architektuře ARM. Jsou děleny podle výkonů a užití do čtyř základních kategorií – *High Performance, Mainstream, Ultra-low-power* a *Wireless* [1]. Pro účely této práce byly oba procesory zvoleny z kategorie *Mainstream*. Procesory obsahují velké množství pro tuto práci užitečných periférií. Ty je možné nastavovat buď klasickým způsobem přes registry nebo pomocí programu STM32CubeIDE. V práci jsou detailnějším popisu konkrétních mikroprocesorů a jejich periférií věnovány kapitoly 5.1.2 až 5.1.6.

STM32F303RE v bloku Master bude zpracovávat data z jednotlivých senzorů. Tato data pak bude odesílat po SPI do zobrazovací jednotky (Slave) a přes D/A převodník na analogové výstupy v podobě napětí odpovídající úrovně. Popisu D/A převodníku se věnuje kapitola 6.2.

Hardware je obsluhován bloky kódu jazyka C. Vývojové diagramy těchto kódů jak pro blok Master, tak Slave následují.



Obr. 1.2. Vývojový diagram kódu v bloku Master



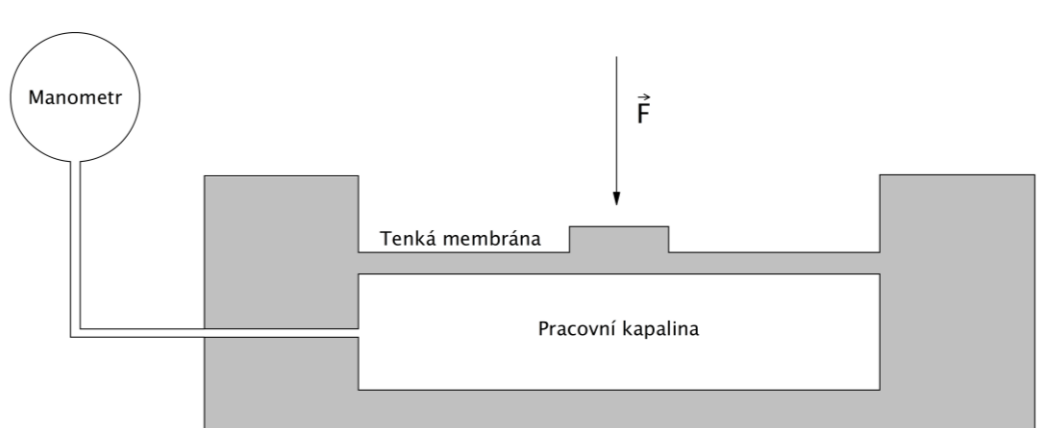
Obr. 1.3. Vývojový diagram kódu v bloku Slave

KAPITOLA 2: SNÍMAČE SÍLY

Snímač síly je zařízení, které má mnoho různých konstrukčních podob v závislosti na konkrétním provedení. Obecně převádí mechanický tlak, tah či moment na analogový elektrický signál, který je dále zpracováván dalším zařízením. Nejčastěji jsou v provedení pneumatickém, hydraulickém a nejběžnějším – tenzometrickém [1]. V této práci je využíván senzor tenzometrického typu, proto bude kladen důraz zejména na něj.

2.1 Hydraulický snímač síly

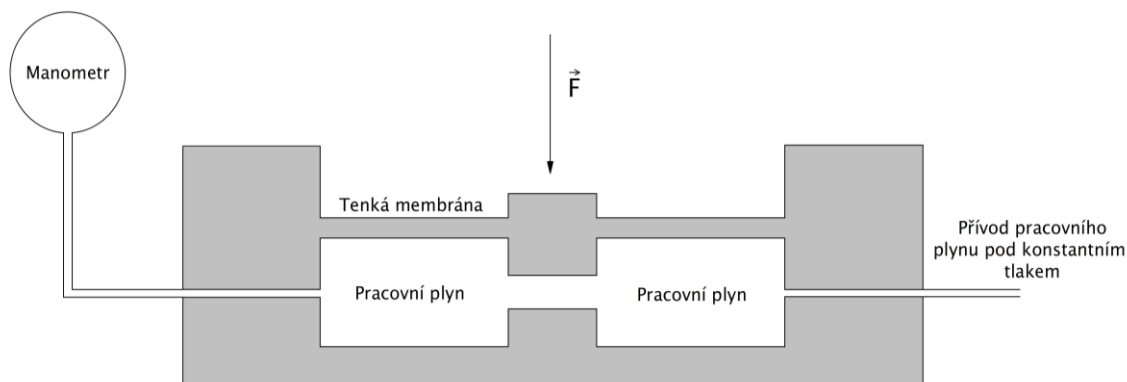
Hydraulický snímač síly ke své funkci využívá nestlačitelnosti kapalin. Ze schematického hlediska se jedná o konvenční soustavu dvou pístů, mezi nimiž je pracovní kapalina. Jeden z pístů je tlačěn měřenou silou a působí na pružnou membránu, která tlačí na kapalinu a tím zvyšuje její tlak. Měřenou sílu určíme z rozdílu tlaků, který získáme pomocí manometru. Vlivem přítomnosti membrány vznikají v závislosti mezi tlakem a silou mírné nelinearity. Díky absenci elektrických součástí najdou hydraulické snímače síly své využití v oblastech se zvýšeným rizikem výbuchu.



Obr. 2.1. Hydraulický snímač síly, [13], vlastní zpracování

2.2 Pneumatický snímač síly

Pneumatický snímač funguje na podobném principu jako snímač hydraulický. Pracovním médiem je plyn, jež je do komor vtláčován pod konstantním tlakem. Neznámá síla tlačí na membránu, která stlačuje pracovní plyn, a tím dojde k vzrůstu tlaku, jehož velikost lze opět odečíst z manometru a sílu dopočítat. Tento snímač rovněž trpí mírnou nelinearitou mezi silou a tlakem kvůli deformaci membrány. U této konstrukce je možné dosáhnout velmi vysoké přesnosti, která se navíc zvyšuje různými konstrukčními úpravami jako je například více komor, jak můžeme vidět na obrázku níže. Oproti hydraulickým mají výhodu, že při poruše nehrozí únik kapaliny a tím možné poškození jiných zařízení.



Obr. 2.2. Pneumatický snímač síly, [13], vlastní zpracování

2.3 Tenzometrický snímač síly

Tenzometry fungují na principu změny elektrické veličiny jako reakce na mechanické namáhání. Touto veličinou může být například kapacita anebo odpor. Nejčastěji se na trhu setkáme s tenzometry odporovými, které bývají zapojené do wheatstoneova můstku, který slouží jako převodník odpor – napětí a zároveň omezuje nepříznivé vlastnosti tohoto typu senzorů. Tenzometrický snímač jako takový vyžaduje mechanické připevnění na podložku různé pevnosti v závislosti na míněném zatížení.

2.3.1 Princip odporového tenzometru

Odporový tenzometr funguje na principu změny rozměrů materiálu jako reakci na působení vnějších sil a s tím spojenou mechanickou deformaci. Změnu odporu na základě změny rozměrů lze vysvětlit poměrně snadno – předpokládejme válcové těleso, které je na jednom konci namáhané silou \vec{F} ve směru osy x a na druhé straně je pevně uchyceno k nehybné zdi. Jelikož je tato situace jednoduchá, můžeme počítat pouze se skaláry a příslušné směry určíme úvahou. Síla, která působí na toto těleso vyvolá jeho prodloužení v podélném směru a u běžných materiálů zúžení ve směru příčném.

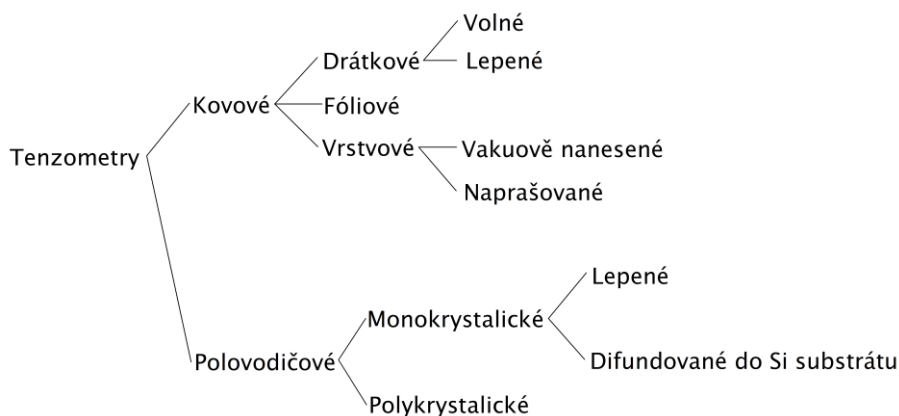
Zapišme elektrický odpor pomocí délky vodiče l , plochy jeho příčného průřezu S a jeho měrného elektrického odporu ρ

$$R = \frac{\rho \cdot l}{S}. \quad (1-1)$$

Z rovnice (1-1) je vidět přímá úměra mezi elektrickým odporem vodiče R a délkou vodiče l a nepřímou úměrou mezi R a plochou jeho průřezu S . Kombinací všech doposud provedených úvah je snadné usoudit, že při mechanickém namáhání dochází k ovlivnění elektrického odporu.

2.3.2 Rozdělení odporových tenzometrů

Na trhu existují různá provedení lišící se výrobním materiálem nosné či funkční části, mechanickým provedením obecně nebo způsobem výroby. Přehled těchto provedení je na následujícím obrázku.



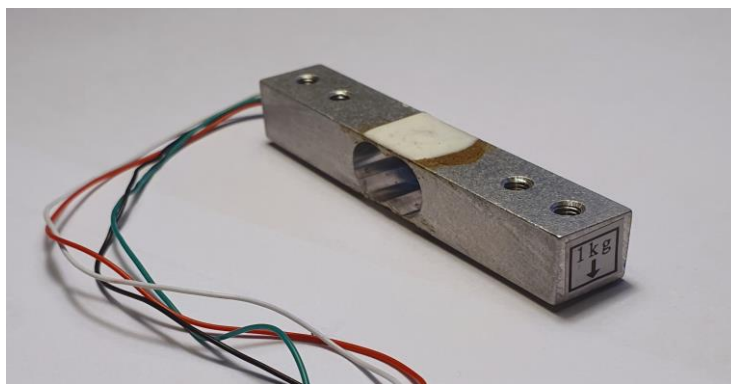
Obr. 2.3. Přehled provedení odporových tenzometrů, [2], vlastní zpracování

Kovové tenzometry se vyrábí ze slitiny mědi (54 %), niklu (45 %) a manganu (1 %) zvané konstantan. Tato sloučenina disponuje velmi nízkým teplotním součinitelem odporu, a to v řádech $10^{-5} [K^{-1}]$ [2]. Tato vlastnost je pro výrobu tenzometrů velkou výhodou. Jejich výhodou oproti tenzometrům polovodičovým je nižší cena, vyšší odolnost vůči změnám okolních teplot a téměř lineární závislost mezi odporem a namáháním. Na druhou stranu kvůli nutnosti většího zesílení výstupního signálu jsou i citlivější na rušení. Polovodičové tenzometry se vyrábějí z křemíku, který se mechanicky připevní k podložce, která je dimenzována s ohledem na maximální zatížení součástky [2]. Disponují až stokrát vyšší přesností v porovnání s tenzometry kovovými [3], oproti nim jsou však mnohem více závislé na teplotě a vyžadují tedy teplotní kompenzaci. Krom toho jsou i cenově méně dostupné.

2.3.3 Konstrukce tenzometrického senzoru síly

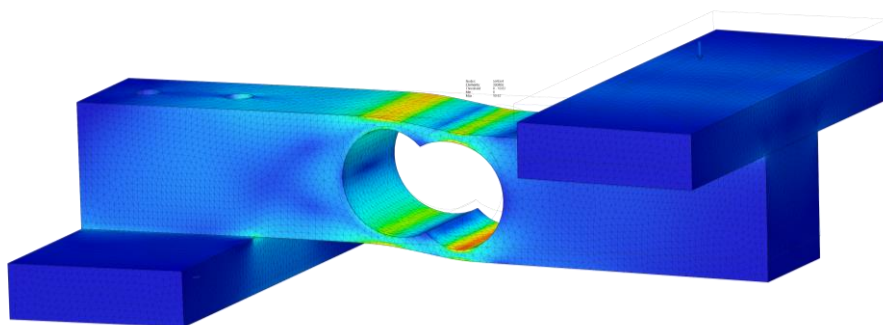
Konstrukce tenzometrických senzorů síly je v principu u všech stejná. Největší rozdíly najdeme v nosné konstrukci, která se liší tvarem a mohutností, v závislosti na plánovaném zatížení daného tenzometru. V rámci testování byl užit senzor dimenzovaný na hmotnost jednoho kilogramu.

Dimenzování senzoru na příslušný rozsah pracovních hmotností spočívá pouze v rozdílné mechanické konstrukci, respektive masivnosti nosné části senzoru. Pevnost materiálu v tahu a tlaku je v tomto ohledu rozhodujícím kritériem. Je jasné, že silové namáhání nesmí nosnou konstrukci nevratně poškodit. U konstrukce je rovněž snahou, aby k mechanické deformaci nedocházelo na celé nosné části, ale aby deformace byla soustředěna na určitou část, na kterou se následně tenzometrické můstky upevňují. Toho je docíleno prostým zeslabením materiálu v požadovaných místech, jak je vidět na Obr. 2.4.



Obr. 2.4. Tenzometrický senzor síly užitý v rámci testování

Pro názornost soustředěnosti mechanického namáhání jako součást této práce vznikl 3D model tenzometrického senzoru, na němž byla simulována statická analýza mechanického napětí. Nižší teplota chromatičnosti barvy značí větší mechanickou deformaci. Z Obr. 2.5 je zjevné, že mechanická deformace je opravdu soustředěna na plochy, na nichž jsou v bílé hmotě na Obr. 2.4 vidět zapouzdřené samotné tenzometry. Na Obr. 2.5 je nosná část za účelem lepší názornosti zatížena daleko větší hmotností, než na který je dimenzována. Při maximálním dovoleném zatížení není mechanická deformace téměř viditelná.



Obr. 2.5. Simulace statické analýzy mechanického napětí

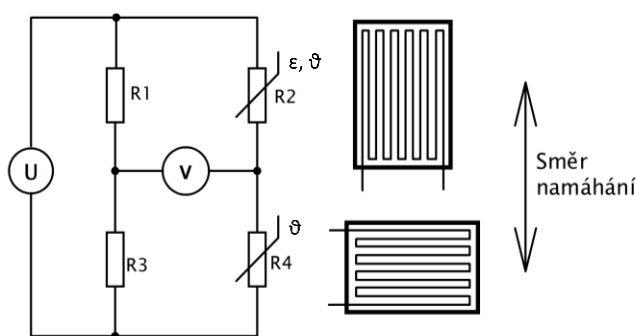
KAPITOLA 3: VYHODNOCOVÁNÍ SIGNÁLU ZE SNÍMAČE SÍLY

Vyhodnocování signálu ze snímače síly je ve své podstatě měření velmi malých změn elektrických odporů, respektive napětí. S vyhodnocováním změn odporu v odporových tenzometrech se pojí několik problémů. Jedním z nich je fakt, že změna odporu je velmi malá – při odporu 350Ω mohou být změny v jednotkách $\mu\Omega$ [4]. Dále nenulová teplotní závislost odporu tenzometru a přívodních vodičů na teplotě. Krom toho i samotná podložka či nosná konstrukce, na níž je tenzometr nalepen, mění s teplotou své rozměry a tím mění i odpor tenzometru. Potýkáme se tedy s obecnou problematikou měření malých změn elektrických napětí, které vyžaduje přesný AD převodník a také vhodné zapojení senzorů, které nepříznivý vliv rušení sníží.

Pro měření velmi malých změn odporů se používá zapojení do nevyváženého wheatstoneova můstku. Tento můstek je pro tuto aplikaci možné zapojit do mnoha možných konfigurací, které přináší různé výhody v ohledu výše zmíněné teplotní kompenzace nebo odstranění vlivu přívodních vodičů.

3.1 Tenzometrický můstek

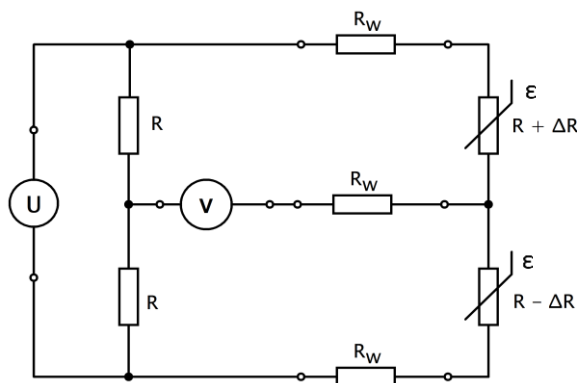
Jednou z možností kompenzace vlivu teploty je pomocí pasivního tenzometru. Schematické zapojení je na Obr. 3.1.



Obr. 3.1. Kompenzace vlivu teploty pasivním tenzometrem, [4], vlastní zpracování

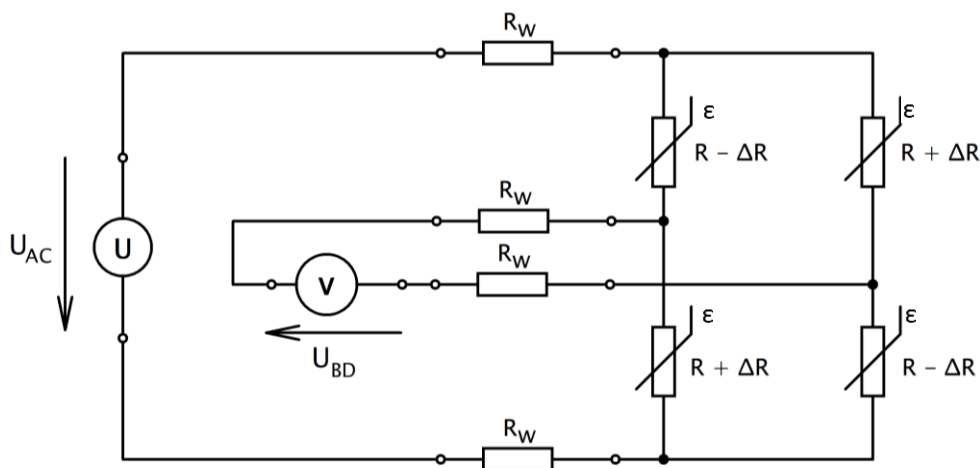
Principem tohoto zapojení je, že hodnota odporu aktivního tenzometru R2 se mění jak s teplotou, tak s mechanickým namáháním. Druhý, pasivní tenzometr R4 svou hodnotu odporu mění pouze vlivem změny teploty a díky tomu se vliv teploty odstraní.

Dalším možností je uspořádání do diferenčního polovičního můstku. Toho je možné užít jen v případě, že tenzometry lze na sledovaný objekt namontovat tak, že jeden je při namáhání prodlužován a druhý stlačován. Takové zapojení zajišťuje nejen zvýšení linearitu, eliminaci vlivu teploty a odporu přívodů R_W , ale také zdvojnásobení přesnosti [2]. Jeho schéma je na Obr. 3.2.



Obr. 3.2. Poloviční tenzometrický můstek, [4], vlastní zpracování

Pro tuto práci nejdůležitější je však zapojení úplného tenzometrického můstku, které využívá čtyř tenzometrů, přičemž dva jsou umístěny tak, aby při mechanickém zatížení docházelo k jejich zkracování a zbylé dva se naopak natahují. Plný můstek má všechny výhody můstku polovičního, nabízí však čtyřnásobnou přesnost oproti užití pouze jednoho tenzometru, a to je důvod, proč je součástí většiny tenzometrických senzorů. Jeho schéma následuje na Obr. 3.3 níže.



Obr. 3.3. Úplný tenzometrický můstek, [4], vlastní zpracování

Výstupní napětí při napájení ze zdroje napětí je pak dáno vztahem

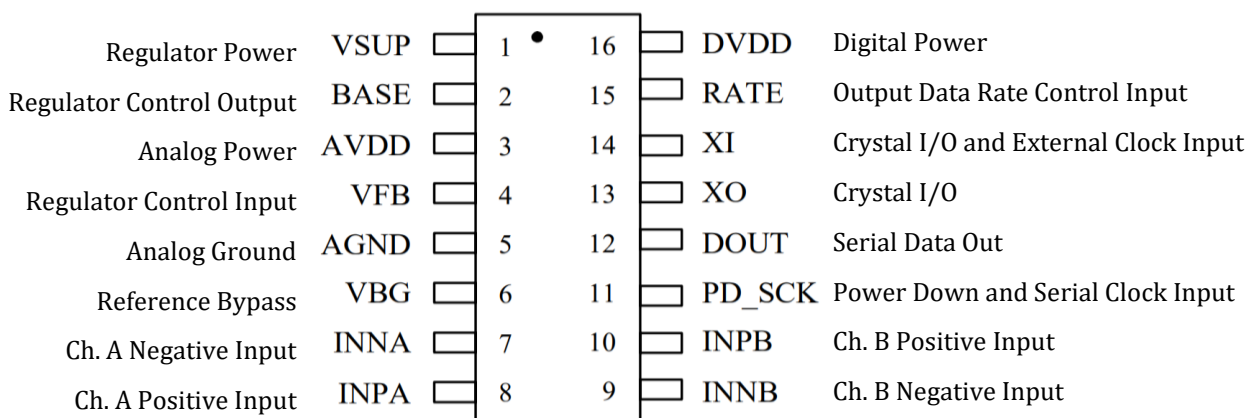
$$U_{BD} = U_{AC} \frac{\Delta R}{R}, \quad (3-1)$$

kde užitá značka veličin jsou zjevné z Obr. 3.3.

Byla tedy vyřešena problematika teplotní kompenzace a zvýšení přesnosti. Při pořizování kompletního senzoru síly jsou jednotlivé tenzometry již vhodně mechanicky namontovány, a jak již bylo zmíněno, jsou zapojeny nejčastěji v úplném tenzometrickém můstku. Jejich mechanickou konfigurací se tedy uživatel nemusí zabývat. Zbývá tedy převedení analogového signálu na signál digitální, a to s maximální možnou přesností a minimální možnou chybou.

3.2 Modul pro váhové senzory s HX711

Obvod HX711 je 24bitový AD převodník vyvinutý společností AVIA Semiconductors právě pro účely zpracování signálů z váhových senzorů. Pinout tohoto obvodu následuje na Obr. 3.4.



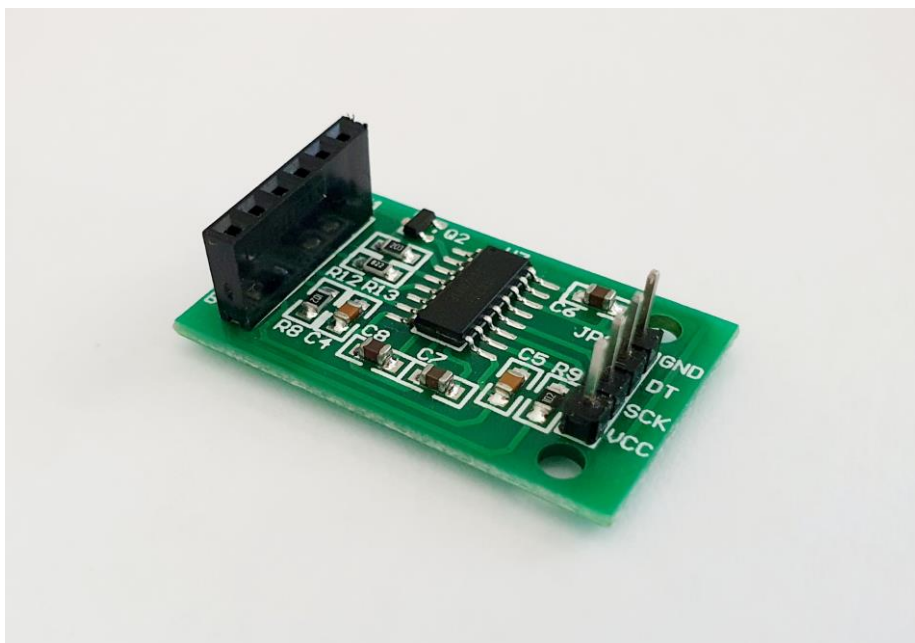
Obr. 3.4. Pinout obvodu HX711, [6], vlastní zpracování

Obvod obsahuje dva vstupní kanály A a B. Oba tyto kanály slouží pro měření diferenciálního napětí – kanál A má zesílení 64 nebo 128, zatímco kanál B 32 [5], [6]. Kanály mají různé rozsahy vstupních diferenciálních napětí a vybírá se mezi nimi rozdílným počtem hodinových pulzů odeslaných na PD_SCK při čtení dat. Přehled rozsahů příslušných kanálů a počtu hodinových pulzů pro jejich výběr následuje na obrázku Obr. 3.5.

Pulsy na PD_SCK	Vstupní kanál	Zesílení	Rozsah vstupního napětí [mV]
25	A	128	±20
26	B	32	±80
27	A	64	±40

Obr. 3.5. Přehled informací o vstupních kanálech HX711, [6], upraveno

AD převod je prováděn metodou $\Sigma\Delta$ (sigma-delta) a rychlost vzorkování je 80 SPS (Samples per second). Výsledné číslo je kódováno v podobě 24bit čísla ve dvojkovém doplňku. V případě, že vstupní napětí přesáhne maximální rozmezí, výstupní data budou odpovídat hodnotám 0x800000 (minimum) nebo 0x7FFFFFF (maximum) [6].



Obr. 3.6. Reálné provedení modulu s HX711

Je nutné si uvědomit, že kvůli kombinaci malých změn odporů, respektive napětí a vysoké přesnosti užitého AD převodníku je pro správná měření nutné zamezit vnějšímu elektromagnetickému rušení nejen přívodních vodičů, ale i celého obvodu. Krom toho je nutné zařídit maximální stabilitu napájecího a tím i referenčního napětí. Stabilizace napájecího napětí je částečně zajištěna modulem, na němž je HX711 dodáván [5]. Kompletní modul pak je plošný spoj obsahující stabilizační obvod, připojovací piny a obvod HX711. Reálné provedení je na Obr. 3.6.

KAPITOLA 4: SNÍMAČE OTÁČIVÉ RYCHLOSTI

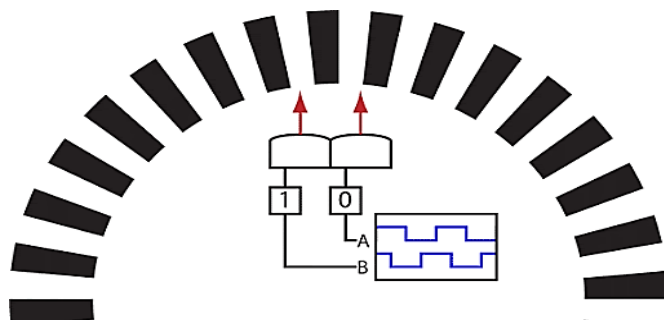
Snímače otáčivé rychlosti jsou dnes nezbytnou součástí moderních pohonů, které pro dosažení žádaných vlastností vyžadují přesnou znalost otáčivé rychlosti spolu se znalostí aktuální polohy rotoru. S takovými pohony se krom robotiky a průmyslové automatizace můžeme setkat například v elektrické trakci.

Rozmanitost provedení těchto snímačů je veliká. Jedním ze starších řešení je tzv. tachodynamo, které využívá lineární závislosti mezi indukovaným napětím v kotvě a otáčkami rotoru u stejnosměrného, nejčastěji cize buzeného motoru. K měření je tedy potřeba jen voltmetr a převodní konstanta respektující magnetický tok a konstrukční uspořádání tachodynama. Zjevnou nevýhodou je nepřesnost určení otáčivé rychlosti, která je ovlivněna nepřesnostmi při měření indukovaného napětí, nelinearitou mezi otáčkami a indukovaným napětím, případně nestabilitou konstrukční konstanty vlivem stárnutí zařízení.

Zařízení, kterých se poslední roky využívá nejvíce pro svou relativní jednoduchost a přesnost, jsou tzv. inkrementální snímače nebo někdy též nazývané inkrementální enkodéry. Senzory tohoto typu typicky neumožňují určení absolutní polohy rotoru – pro získání takové informace je nutné využít senzoru vybaveného vícero stopami, jejichž název pak bývá doplněn o slovo „absolutní“. Počet stop v tomto případě udává maximální přesnost, se kterou je enkodér schopen určit absolutní polohu. Těmto enkodérům, včetně enkodérů absolutních, se věnuje tato kapitola.

4.1 Popis funkce inkrementálního enkodéru

Inkrementální enkodér je typ senzoru pro měření otáčivé rychlosti, který však obecně nedává informaci o aktuální poloze. Výstupem z inkrementálního enkodéru je kvadrurní signál, který se skládá z dvou o čtvrtinu periody fázově posunutých obdélníkových průběhů – viditelný na Obr. 4.1 uprostřed modře. Prvotní signál vznikající z fyzikální podstaty v senzorech však zdaleka nemá obdélníkový tvar a je tedy nutné jej nejčastěji pomocí komparátorů upravit.



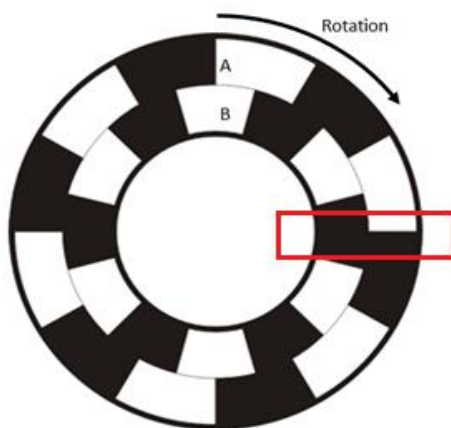
Obr. 4.1. Kvadrurní signál, jednostopý senzor, převzato z [14]

Výhodou kvadrurního signálu je, že udává informaci o rychlosti a díky duplicitě a fázovému posuvu signálů rovněž o směru otáčení. Navíc, díky digitální podobě, je relativně odolný vůči rušení, což rozhodně nelze říci o analogovém výstupním signálu z tachodynamu.

4.2 Vznik kvadrurního signálu v inkrementálním enkodéru

Obecný popis vzniku kvadrurního signálu není vzhledem k množství různých provedení snadný. Na trhu jsou dostupné inkrementální enkodéry fungující například na optoelektrickém, magnetickém, indukčním nebo kapacitním principu [7]. Obecně lze uvažovat dvě základní provedení. Jedním z nich je, že signál vzniká interakcí vhodného rotačního jednostopého disku a dvou senzorů citlivých na příslušnou veličinu. Ukázka takové konfigurace je na Obr. 4.1.

Další možností vzniku signálu je, že rotační kotouč obsahuje dvě stopy, které jsou vůči sobě posunuty o čtvrtinu periody a senzory jsou pak v jedné úrovni. Tato konfigurace je vidět na Obr. 4.2, kde červený obdélník značí umístění senzorů.



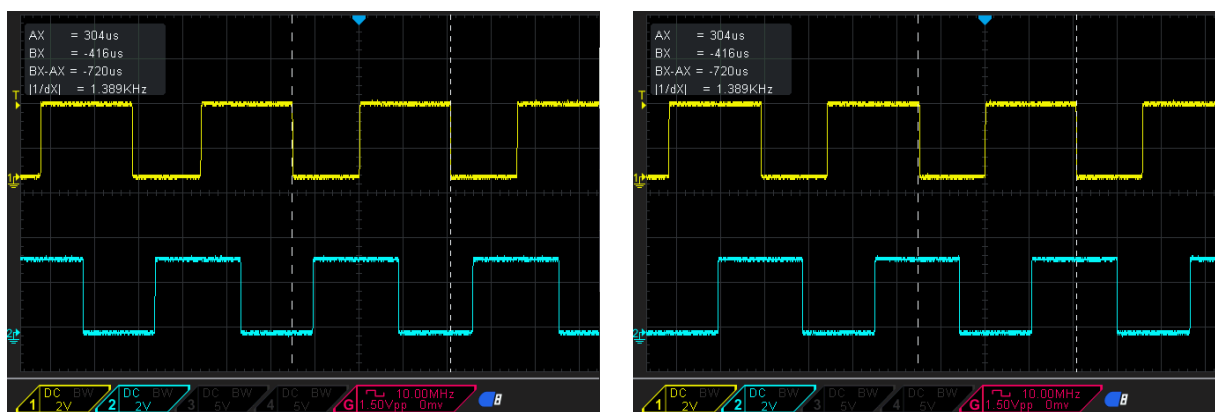
Obr. 4.2. Princip dvoustopého senzoru, převzato z [15], upraveno

Rozlišení takovýchto enkodérů je pak dáno počtem pulzů na jednu otáčku enkodéru. Běžné množství pulzů na otáčku bývá mezi 512 a 180000 (enkodéry s takto vysokým množstvím pulzů na otáčku jsou výhradně optické), což jsou poměrně vysoké hodnoty, které zajišťují vysokou přesnost určování rychlosti otáčení, popřípadě úhlu natočení. Na druhou stranu kladou zvýšené nároky na výpočetní techniku užitou ke zpracovávání takového signálu, a to zejména při vyšších otáčkách. Enkodéry často také obsahují tzv. nulový pulz, respektive nulovou stopu. Tato stopa slouží k synchronizaci a je zdrojem pulzu pouze jednou za otáčku. Některé mikroprocesory mají integrované periferie pro zpracování i tohoto pulzu a s jeho příchodem mohou například vyvolat přerušování, jehož obsluha je pak v rukou programátora. Více o těchto perifériích v podkapitole 5.1.2.



Obr. 4.3. Kvadrurní enkodér s nulovou stopou, převzato z [19]

Reálná ukázka kvadrurního signálu z magnetického enkodéru užitého v rámci testování je vidět na Obr. 4.4. Signál byl osciloskopem zaznamenán pro dva různé směry otáčení.



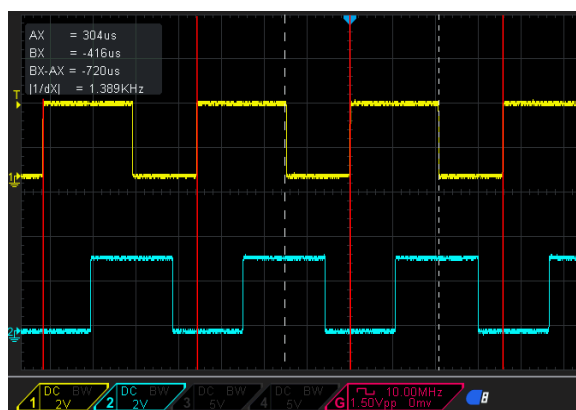
Obr. 4.4. Kvadrurní signál pro dva různé směry otáčení

4.3 Typy zpracování kvadrurního signálu

Kvadrurní signál je možné dekódovat několika způsoby, které se liší zejména hranami, které jsou u signálu detekovány. Toto kritérium má také vliv na přesnost samotného zpracování. S vyšší přesností se pojí vyšší nároky na výpočetní techniku, pomocí níž je signál zpracováván.

4.3.1 Zpracování X1

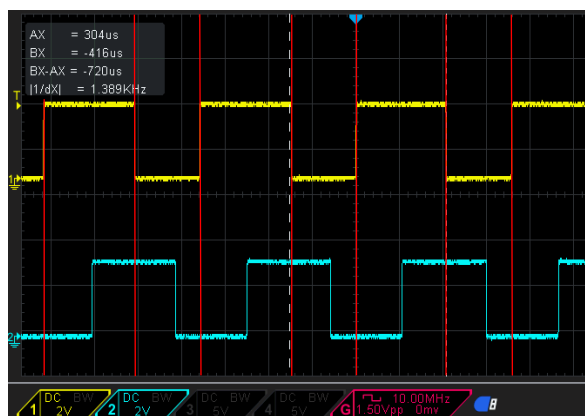
Tento typ zpracování využívá pouze čelo jednoho ze signálů. Druhý signál je nutný pro určení směru otáčení. Situace je vyobrazena na Obr. 4.5, kde každá z červených čar značí inkrementaci případně dekrementaci čítače užitého pro zpracování signálu. Tyto čítače budou podrobněji vysvětleny v kapitole 5.1.



Obr. 4.5. Zpracování X1

4.3.2 Zpracování X2

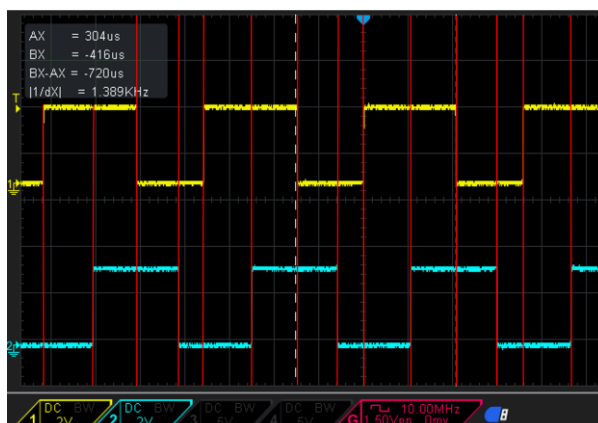
Lepším zpracováním, co se rozlišení týče, je zpracování X2, kde čítač započítává jak čelo, tak tyl jednoho ze signálů. Druhý signál je opět potřebný pro určení směru otáčení. Oproti předchozímu kódování zaručuje vyšší přesnost. Situace je graficky znázorněna na Obr. 4.6 níže.



Obr. 4.6. Zpracování X2

4.3.3 Zpracování X4

Posledním a v praxi nejčastěji užívaným typem zpracování je X4. Čítač v tomto případě zaznamenává jak čela, tak týly u obou signálů. Situace je vyobrazena na Obr. 4.7. Z obrázku je vzhledem k množství červených čar značících inkrementaci případně dekrementaci čítače jasné, že toto kódování zařizuje nejvyšší přesnosti ze všech výše zmíněných způsobů, ale na úkor nutnosti výkonnějšího hardwaru na zpracování. Aktuální mikroprocesorová technika však dosahuje takových výkonů i v nižších cenových třídách, že na hardwarové zatížení v podstatě není nutné brát zřetel a rozhodujícím kritériem je ve většině aplikací přesnost.

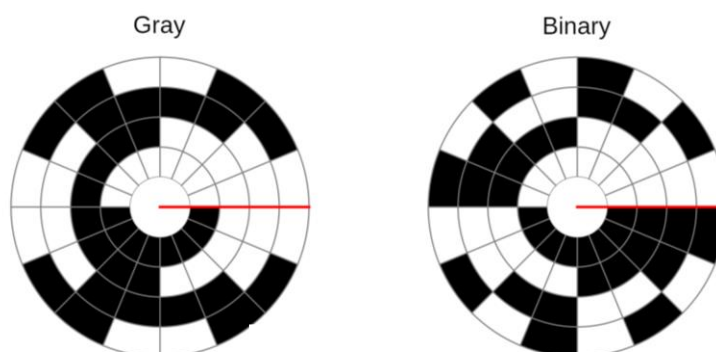


Obr. 4.7. Zpracování X4

4.4 Absolutní enkodér

Absolutní enkodér je speciální a oproti enkodéru inkrementálnímu konstrukčně složitější typ enkodéru, který umožňuje odečít i aktuální polohy motoru v momentě jeho zapnutí.

V tomto případě se užívá vícero optických stop, jejichž množství přímo ovlivňuje maximální přesnost určení polohy. Každá poloha enkodéru má pak konkrétní binární kód. Je-li informace o poloze zakódována v běžné dvojkové podobě, může dojít zejména při vyšších rychlostech k chybným určení polohy. Z tohoto důvodu vznikl Greyův kód, u kterého mezi jednotlivými pozicemi natočené dochází ke změně pouze v jednom bitu, čímž je možné softwarově eliminovat chybné interpretace dat. Ukázka obou typů kódování je na Obr. 4.8.



Obr. 4.8. Provedení optických stop s různými typy kódování, převzato z [18], upraveno

KAPITOLA 5: VYHODNOCENÍ SIGNÁLU ZE SENZORU OTÁČEK

Pro zpracování signálu z kvadrurního senzoru otáček nestačí běžná analogová technika, nýbrž je nutné užití například mikroprocesoru vybaveného vhodnou periferií pro čítání pulzů i při vysokých frekvencích. Jedna z vhodných periferií u mikroprocesorů jsou časovače užitá v režimu čítačů. Někteří výrobci své mikroprocesory navíc vybavily rozšířenými funkcemi, které zpracování takového signálu velmi zjednodušují.

Pulzy kvadrurního signálu jsou pak čítány po známou dobu, jež je přesně vymezena druhým časovačem, který po této periodě vydává požadavek o přerušení a dojde k zaznamenání stavu čítače do paměti. Metoda výpočtu otáčivé rychlosti bude detailněji popsána v podkapitole 5.1.3.

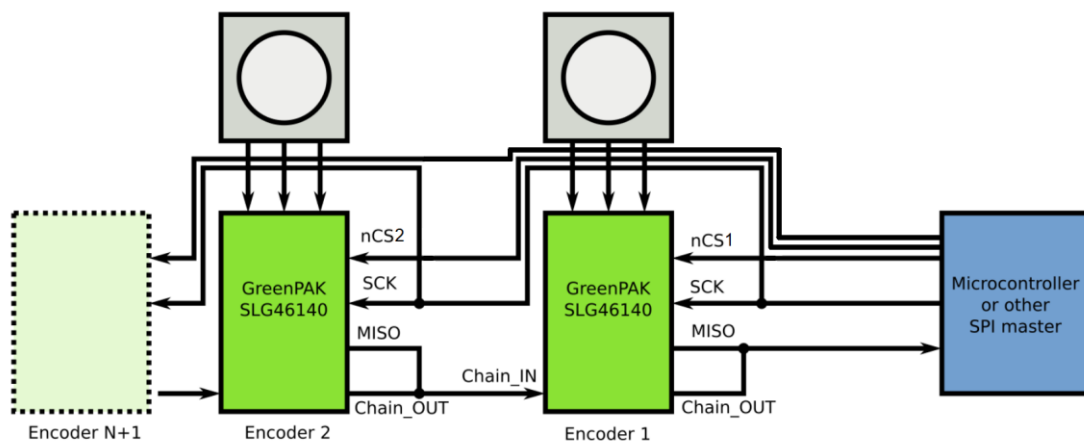
Doba měření musí být dostatečně dlouhá, aby případná chyba v řádu jednoho bitu byla vzhledem k celkovému počtu naměřených pulzů za periodu zanedbatelná. Zároveň nesmí být moc dlouhá, aby bylo měření dynamické a bylo schopné zaznamenávat i prudké změny otáčivé rychlosti. V případě malých otáček může být výhodnější pomocí časovače měřit periodu mezi jednotlivými pulzy namísto jejich frekvence [8].

V minulé kapitole byly objasněny různé typy zpracování kvadrurního signálu v kontextu s jejich výhodami a nevýhodami. Pro účely této práce bude tedy vzhledem k jasným výhodám a zanedbatelným nevýhodám užitó zpracování X4. K vyhodnocování kvadrurního signálu je možné užit krom již zmíněného mikroprocesoru také integrovaných obvodů nebo přímo hradel. Tato možnost však zejména v posledních letech ztrácí vzhledem k cenám mikroprocesorů na své atraktivnosti, a to hlavně v případech, kdy je výrobní série malá nebo obsahuje jen pár kusů, takže je rozdíl v ceně zanedbatelný. Přesto bude v této kapitole pro účely kompletního přehledu možností takový obvod krátce představen.

5.1 Vyhodnocení kvadrurního signálu

5.1.1 Integrovaný obvod v SLG46140V

Jedním z integrovaných obvodů umožňujících zpracování kvadrurního signálu je obvod SLG46140V. Obvod obsahuje 16bitový čítač, blok se vstupními piny pro kvadrurní signál a blok s rozhraním SPI pro připojení k mikroprocesoru. SPI sběrnice rovněž umožňuje připojení vícero takovýchto integrovaných obvodů a jejich rozlišení pomocí vodičů chipselect [9]. Pro tyto vodiče je na Obr. 5.1 užitá zkratka nCS1 respektive nCS2. SPI bude detailněji popsáno v podkapitole 6.3.



Obr. 5.1. Zapojení více SLG46140V obvodů na SPI sběrnici, převzato z [9], upraveno

Jak již bylo zmíněno výše, integrované obvody tohoto typu je však vhodné užít pouze v případech, kdy podobným rozhraním není vybaven mikroprocesor. Pro případ této práce však mikroprocesor vhodnou periferií vybaven je a jejímu důkladnějšímu popisu bude věnována následující podkapitola.

5.1.2 Periferie TIM v procesorech STM32

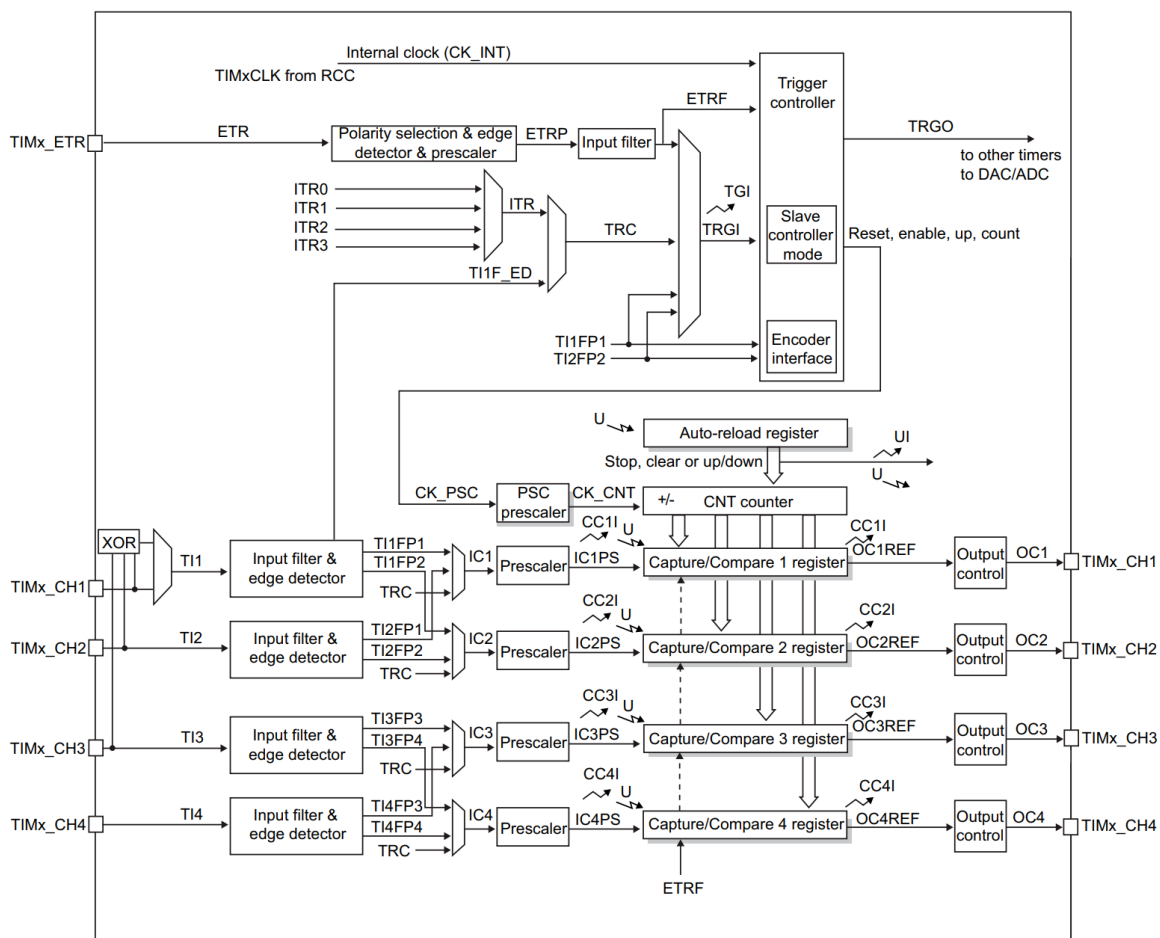
Periferie typu TIM je hardwarový obvod, který je schopen fungovat v módu čítače (anglicky Counter) nebo časovače (anglicky Timer), přičemž rozdíl je v okolnosti, při které mění příslušný čítací registr (anglicky Counter register nebo CNT) svoji hodnotu. Tato periferie slouží ke generování signálů o různé frekvenci, generování pulzně šířkové modulace (PWM), měření času mezi událostmi nebo k čítání událostí. Krom toho je periferií vhodnou pro vyhodnocení právě kvadrurního signálu.

Procesory STM32F30x mají těchto periferií několik, ty se liší se zejména pokročilými funkcemi, zatímco funkce základní najdeme u většiny z nich. Konkrétní periferie typu TIM jsou pak označovány TIMx, kde „x” značí libovolnou číslici – například tedy TIM1, TIM17 atd. Nutno podotknout, že číslice neříká nic o množství TIM periferií v daném procesoru – příkladem je procesor STM32F303RE, který obsahuje TIM20, avšak těchto periferií má méně. Číslice obecně označují periferie vybavené stejnými funkcemi napříč celou rodinou procesorů STM32 – existují však výjimky a je tedy nutné ověřit funkce před užíváním [10].

Podle [10] rozlišujeme čtyři základní úrovně funkčního vybavení a složitostí TIM periferií:

- časovače s pokročilou konfigurovatelností – např. TIM1 a TIM8,
- víceúčelové časovače – např. TIM2 a TIM3,
- časovače s odlehčenou konfigurovatelností – např. TIM9 a TIM10,
- základní časovače – např. TIM6 a TIM7.

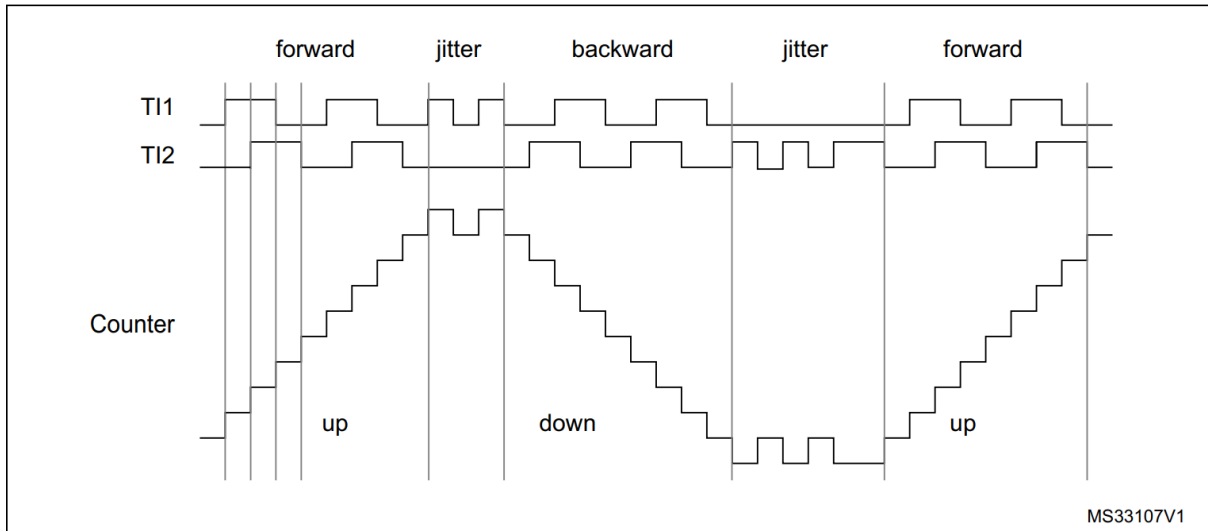
Na Obr. 5.2 můžeme vidět blokové schéma struktury časovače TIM2, který je jedním ze skupiny víceúčelových časovačů (anglicky General-purpose timer). Disponuje 32bitovým counter registrem a integrovanou funkcí *encoder*, která je zásadní pro jednodušší zpracování kvadrurního signálu. Vstupy, které se pro režim *encoder* využívají jsou na Obr. 5.2 označeny TI1FP1 a TI1FP2.



Obr. 5.2. Blokové schéma struktury časovače TIM2, převzato z [11]

5.1.2.1 Periferie TIM v režimu *encoder*

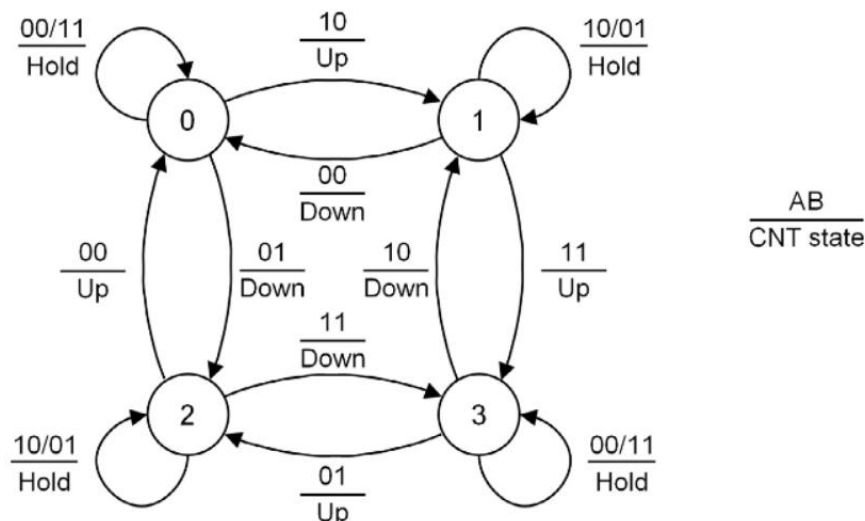
Režim *encoder* nabízí hardwarovou podporu zpracování kvadrurního signálu za účelem čtení rychlosti a směru otáčení u motorů. Výstupy enkodéru se přímo připojí na příslušné kanály timeru tedy TI1FP1 a TIFP2 (pro některé typy enkodérů je nutné zapojit pull-up rezistor) a periferie pak sama vyhodnocuje směr otáčení a podle toho buď inkrementuje či dekrementuje příslušný counter registr. Příklad tohoto vyhodnocujícího procesu je graficky znázorněn na Obr. 5.3.



Obr. 5.3. Příklad procesu vyhodnocení kvadrurního signálu, převzato z [11]

Režim *encoder* nabízí i funkci, která umí pracovat s nulovým pulzem (anglicky Index pulse). Příslušný vývod enkodéru je připojen na vstupní pin TIMx_ETR. Detekce tohoto pulzu je prováděna rozdílnými způsoby při různých směrech otáčení tak, aby se zařízení chovalo symetricky při obou směrech otáčení. Režim *encoder* zastřešuje i další složitější funkce spojené s nulovým pulzem jako například: reset counter registru pouze při jednom směru otáčení nebo reset counter registru pouze jednou. Detekce nulového pulzu může být navíc podmíněna počtem pulzů kvadrurního signálu – je-li nulový pulz jednou za otáčku a enkodér má například 50 pulzů na otáčku, očekává se, že přijde 50 pulzů mezi jednotlivými nulovými pulzy. Není-li tomu tak, může být stav vyhodnocen jako chybový.

Periferie v režimu *encoder* je vybavena i další detekcí chyb, které mohou vznikát vlivem vnějšího rušení, problémy s vodiči nebo znečištěnými senzory. Je-li zpracováván kvadrurní signál, na vstupních pinech je očekávána posloupnost stavů. Pokud není posloupnost dodržena, je stav opět vyhodnocen jako chybový a může dojít k vyvolání přerušení. Na Obr. 5.4 je graficky znázorněn stavový diagram kvadrurního signálu spolu s reakcí čítače na příslušný přechod mezi stavy – *hold* znamená, že čítač svoji hodnotu nezměnil; *up* značí inkrementaci; *down* dekrementaci. V případě, že by zjištěný stav neodpovídal žádné z očekávaných kombinací, mikroprocesor by jej vyhodnotil jako chybový a vykoná nastavené akce, například přerušení.



Obr. 5.4. Stavový diagram kvadrurního signálu, převzato z [16]

5.1.2.2 Nastavení TIM2 do režimu *encoder*

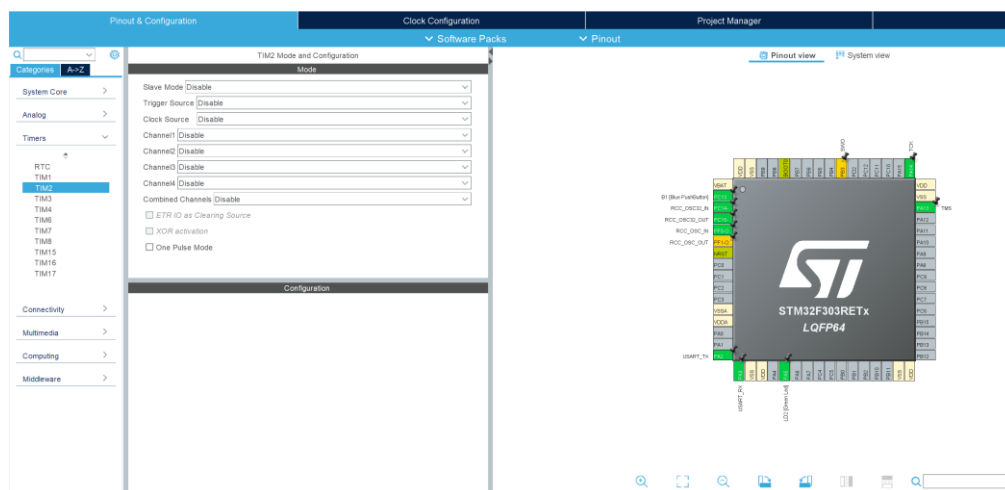
Pro účely této práce bude pro zpracování kvadrurního signálu užít čítač TIM2, jelikož nabízí funkci *encoder* a na rozdíl od jiných TIM disponujících funkcí *encoder* nabízí dvojnásobný rozsah – 32 bitů. Nastavení periferie TIM2 je možné provést dvěma základními způsoby. Jedním z nich je zápis požadovaných hodnot do příslušných registrů. Takový přístup bude s velkou pravděpodobností vyžadovat náhled do referenčního manuálu daného procesoru. Jedno z možných nastavení je dle [11] následující:

- CC1S= 01 (registr TIM2_CCMR1, nastaví kanál jako vstupní a přiřadí TI1 na TI1FP1),
- CC2S= 01 (registr TIM2_CCMR2, nastaví kanál jako vstupní a přiřadí TI2 na TI1FP2),
- CC1P a CC1NP = '0' (registr TIM2_CCER, nastavení neinvertující polarity TI1FP1, resp. TI1),
- CC2P a CC2NP = '0' (registr TIM2_CCER, nastavení neinvertující polarity TI2FP2, resp. TI2),
- SMS= 011 (registr TIM2_SMCR, čítání na hranách signálů TI1 i TI2),
- CEN= 1 (registr TIM2_CR1, povolené čítače).

Takto nastavená periferie by se pak chovala jak je vyobrazeno na Obr. 5.3.

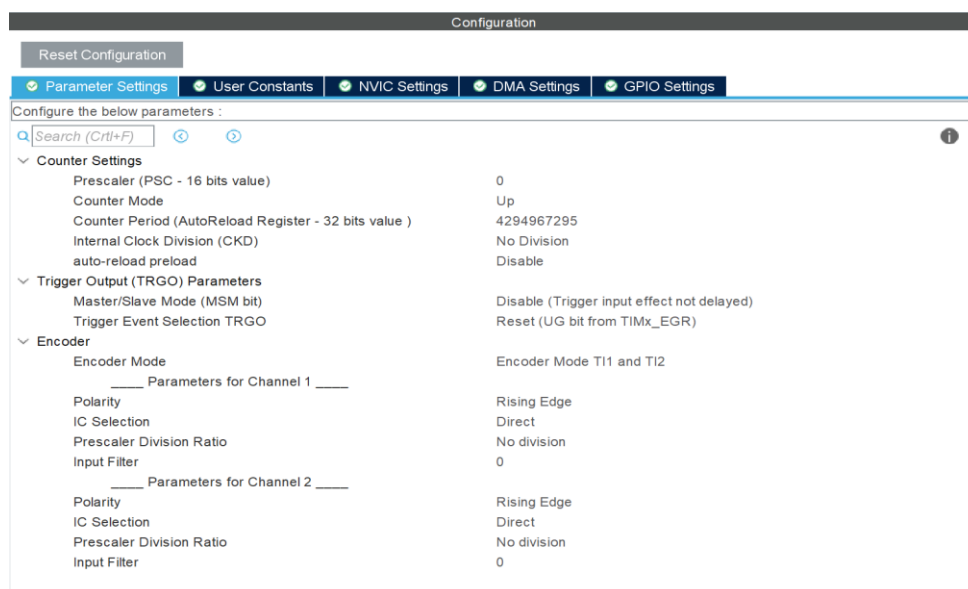
Druhou, snazší možností je nastavení registrů a tím celé periferie pomocí prostředí STM32CubeMX integrovaného do STM32CubeIDE, a to následujícím způsobem:

- v záložce *File*, zvolíme *New* a vybereme STM32 Project
- z nabídky vybereme konkrétní procesor nebo vývojový kit a výběr potvrdíme kliknutím na tlačítko *Next*, projekt pojmenujeme a klikneme na *Finish*
- po dokončení základní inicializace se objeví interface s náhledem na procesor, v levé části vybereme *Timers* a zvolíme TIM2



Obr. 5.5. Interface prostředí STM32CubeIDE

- v řádku *Combined Channels* zvolíme *Encoder mode*, ve spodní části se objeví následující menu, ve kterém nakonfigurujeme konkrétní parametry dle potřeby
- v záložce *GPIO Settings* je pak možné jednotlivě pro oba vstupní piny zapnout pull-up či pull-down rezistory, což některé enkodéry mohou vyžadovat

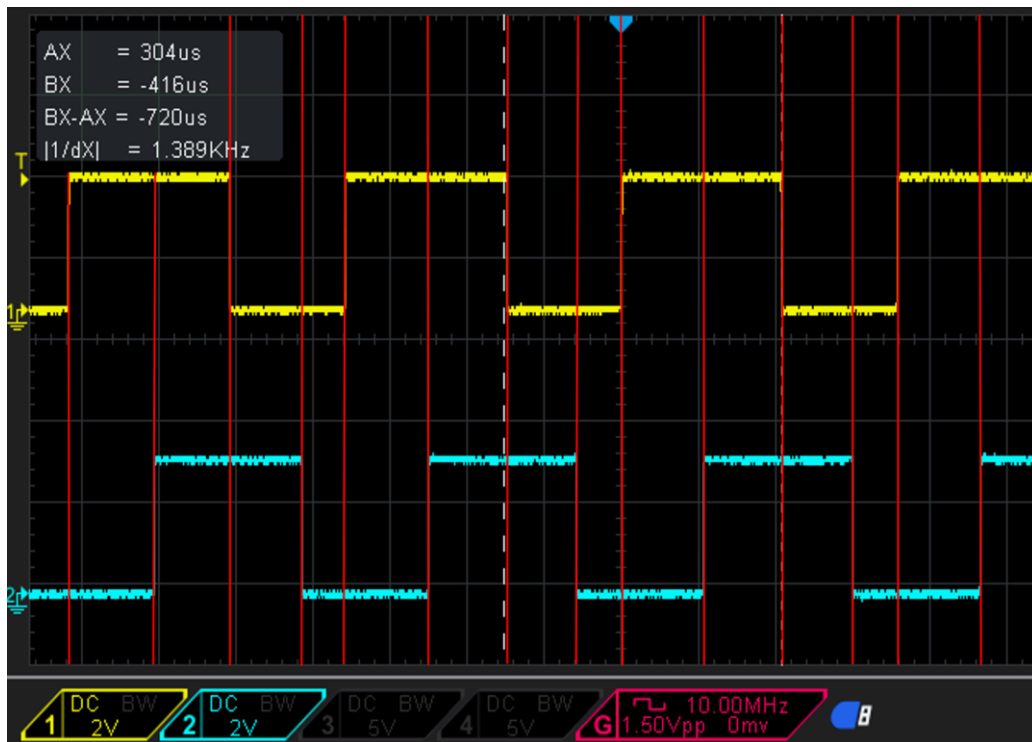


Obr. 5.6. Interface pro nastavení parametrů TIM2 v prostředí STM32CubeIDE

- stisknutím klávesové kombinace *Ctrl + S* dojde k uložení a automatickému vygenerování inicializačního kódu

5.1.3 Výpočet otáčivé rychlosti

Pro výpočet otáčivé rychlosti je kromě čítání pulzů potřeba znát i několik dalších hodnot. Předpokládejme obecný průběh kvadrurního signálu z Obr. 5.7 a že k inkrementaci čítače dochází v každém momentě označeném červenou čarou – tedy zpracování X4.



Obr. 5.7. Předpokládaný průběh kvadrurního signálu

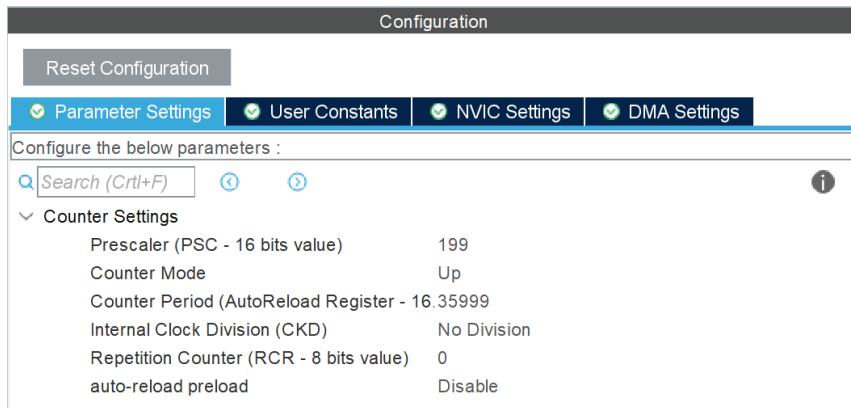
Vzorec pro výpočet otáčivé rychlosti v otáčkách za minutu je pak následující:

$$n = \frac{Q_{puls} \cdot 60}{X \cdot T \cdot Q_{enc}}, \quad (5-1)$$

kde n jsou otáčky za minutu, Q_{puls} je počet načítaných pulzů za dobu čítání $T[s]$, X je konstanta závislá na typu kódování – pro kódování X4 je $X=4$, pro X2 je $X=2$ a pro X1 analogicky a Q_{enc} je konstanta enkodéru, značící počet pulzů na otáčku.

5.1.3.1 Nastavení periferie TIM16

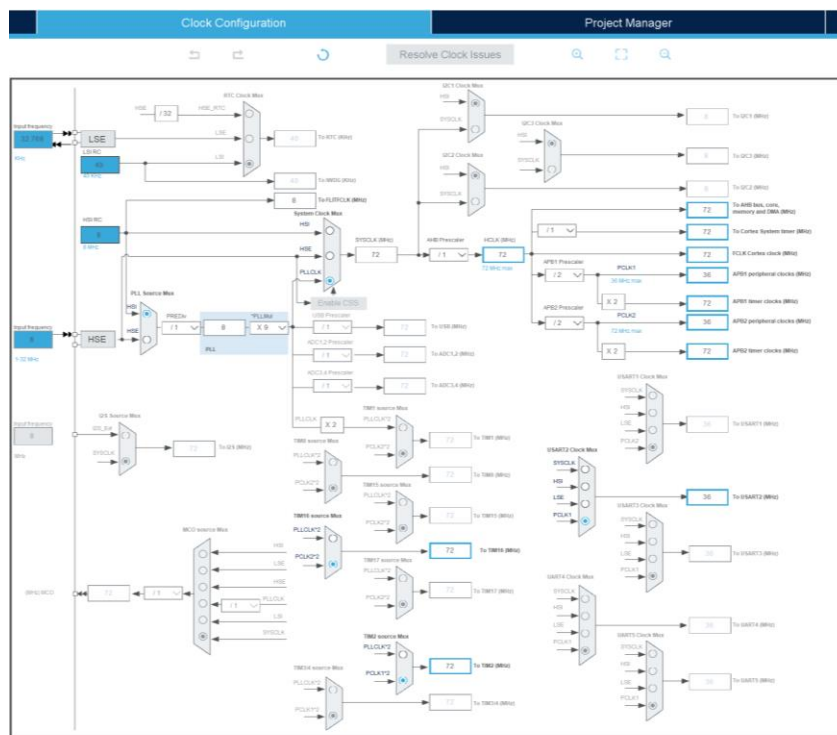
V programu je sekunda přesně odměřována pomocí periferie TIM16. TIM16 je šestnáctibitový čítač, který je nastaven tak, jak je vidět na Obr. 5.8.



Obr. 5.8. Nastavení periferie TIM16 v STM32CubeIDE

Hodnota *Prescaleru*, neboli předděličky, je ve skutečnosti nastavena na 200, protože STM32CubeIDE automaticky přičítá k vepsané hodnotě jedničku, aby tak předešel nedefinovanému výrazu pro *Prescaler* roven nule. Totéž platí i pro *Counter Period*.

V záložce *Clock Configuration* je uživateli zprostředkován pohled na schéma, které umožňuje náhled a konfiguraci zdrojů hodin pro různé periferie – podoba této záložky je na následujícím obrázku. Při bližším pohledu je vidět, že do periferie TIM16 je napojen zdroj hodin o frekvenci 72 MHz.



Obr. 5.9. Podoba záložky Clock Configuration

Z frekvence hodin a znalosti hodnot *Prescaler* a *Counter Period* lze vypočítat periodu přetečení pomocí následujícího vztahu

$$T_{ovr} = \frac{N_{CP} + 1}{\frac{f_{osc}}{N_{PS} + 1}} = \frac{35999 + 1}{\frac{72 \cdot 10^6}{199 + 1}} = 0,1, \quad (5-2)$$

kde T_{ovr} značí periodu přetečení, N_{CP} je *Counter Period*, f_{osc} frekvence hodin a N_{PS} je hodnota *Prescaleru*. V tomto případě je tedy perioda 0,1 s a vzorec pro výpočet otáček lze zkonkretizovat na

$$n = \frac{q \cdot 150}{Q}. \quad (5-3)$$

Po uložení a vygenerování kódu v programu STM32CubeIDE pak stačí povolit běh čítače pomocí:

```
HAL_TIM_Base_Start_IT(&htim16);
```

U periferie TIM16 tedy přesně po 0,1 s dojde k přetečení a k vyvolání přerušení. Přerušení je však nejprve potřeba povolit v rámci nastavování této periferie v záložce NVIC. Přerušení pak zavolá callback funkci s názvem *HAL_TIM_PeriodElapsedCallback*, v jejímž rámci je prováděna obsluha a výpočet otáček následujícím kódem:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    prevVal=Val; //stores previous TIM2 value
    Val=(TIM2->CNT); //captures new TIM2 value
    RPM=(Val-prevVal)*(150/PulsesPerRotation); //calculates RPM
}
```

Doba měření 0,1 s se během testování ukázala jako příliš krátká, protože výše zmíněné chyby v řádu jednoho bitu měly příliš velký vliv.

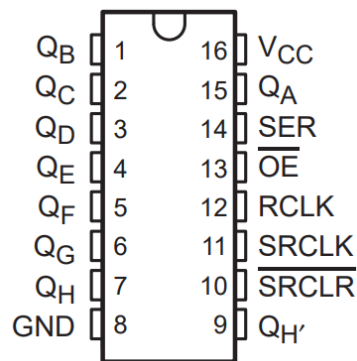
KAPITOLA 6: ZOBRAZENÍ DAT

Zobrazení dat je možné několika způsoby. Primárním a také z hlediska vývoje nejsložitějším způsobem je zobrazení hodnot pomocí velkých sedmsegmentových LED displejů. Tento blok bylo kvůli prostorovému uspořádání laboratoře pohonů na ČVUT FEL vhodné oddělit. Na základě této nutnosti vznikl nápad o vytvoření zcela nezávislé zobrazovací jednotky, která bude přijímat data po SPI a bude obsahovat samostatný mikroprocesor starající se o plynulý chod zobrazování. Tuto jednotku pak bude možné užívat i pro jiné projekty katedry. Pro zajištění zobrazování dat na sedmsegmentových displejích a jejich přijímání po SPI je užit mikroprocesor STM32F042. Tento procesor nedisponuje dostatečným množstvím výstupních pinů, aby mohl segmenty řídit jednotlivě, a proto bylo navrženo multiplex zapojení, které zmenšilo potřebné množství výstupních pinů MCU na minimum a zároveň umožnilo jednoduché řízení. Návrhu hardwaru a řídicího softwaru tohoto zapojení je věnována podkapitola 6.1.

Dalším možným způsobem zobrazení dat je připojením osciloskopu či voltmetru na dva analogové výstupy vyvedené na BNC konektory. Rozsah využitých DA převodníků je pak lineárně rozložen do celkových rozsahů momentů a otáček. Vztah, který je užitečný pro přepočítání mezi napětím a momentem, respektive otáčkami, je uveden a detailněji popsán v podkapitole 6.2. Poslední část této kapitoly je věnována sériové komunikaci SPI spolu s konkrétní ukázkou nastavení této periferie v rozhraní STM32CubeIDE.

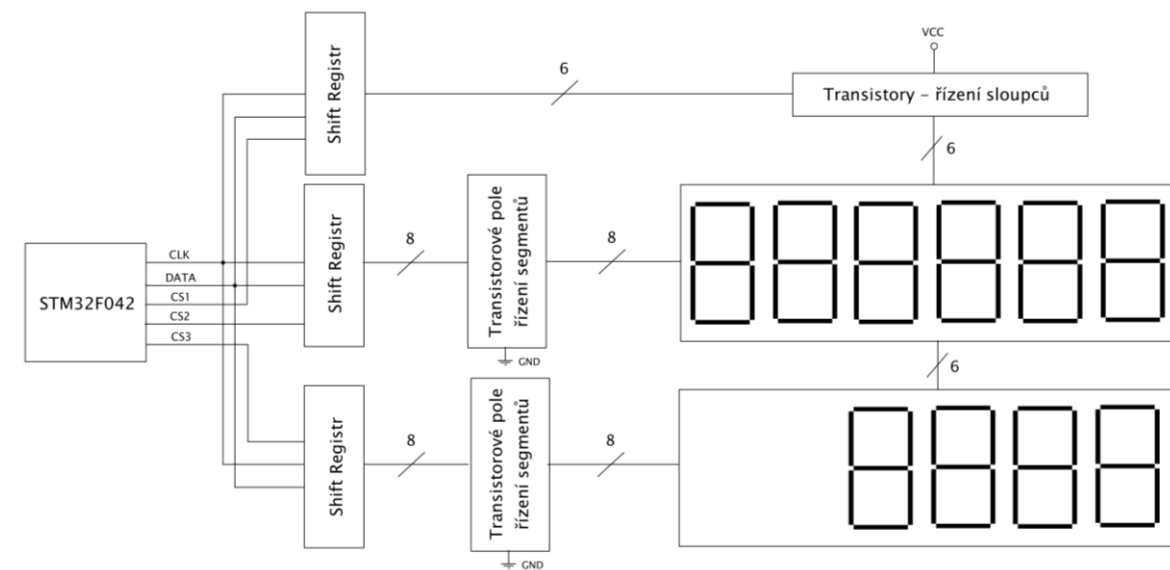
6.1 Multiplex sedmissegmentových LED displeje

V rámci této práce bylo navrženo zapojení využívající tří shiftregistrů 74HC595N. Pinout těchto shift registrů je vidět na Obr. 6.1 níže.

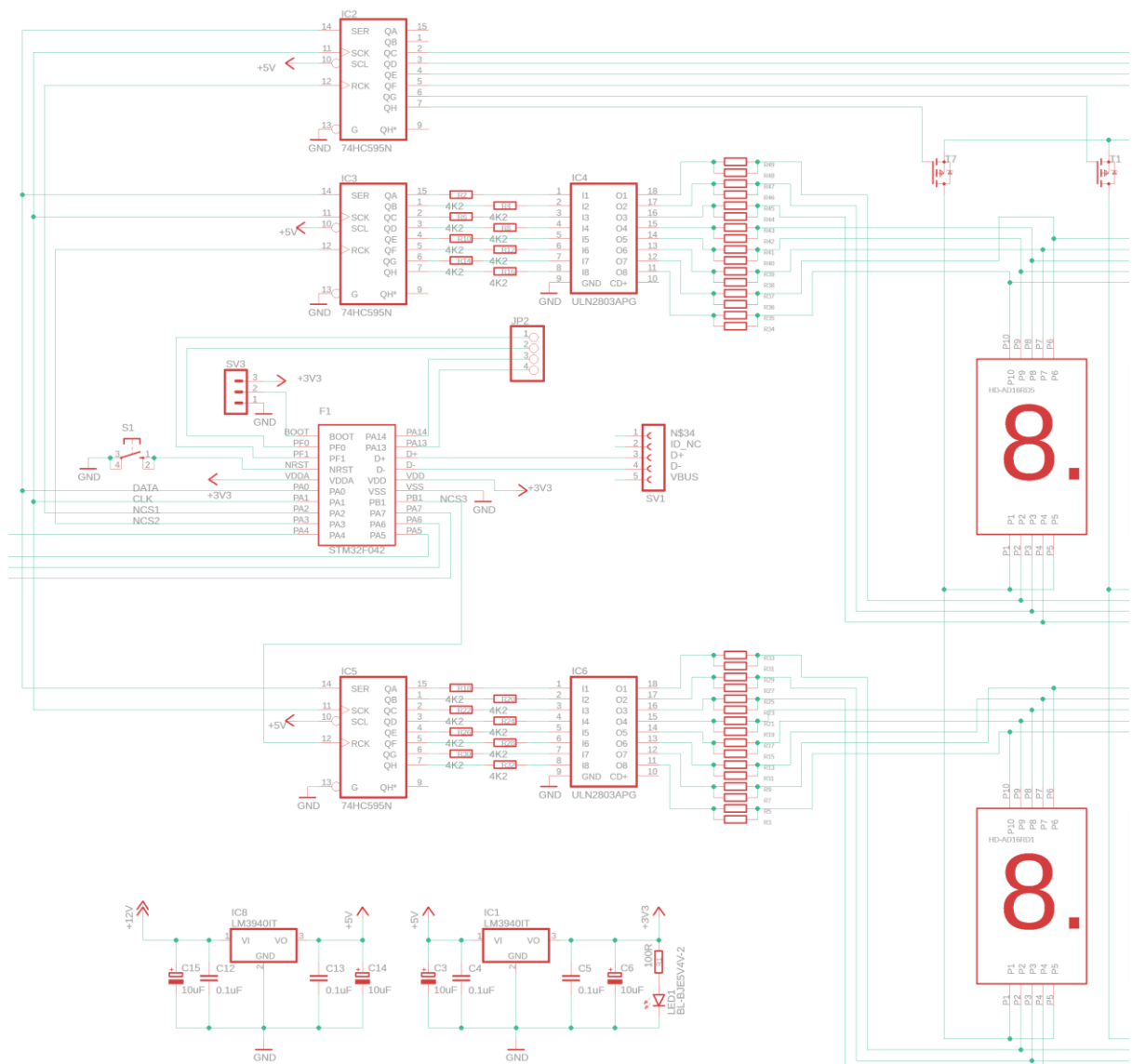


Obr. 6.1. Pinout shiftregistru 74HC595N, převzato z [17]

Tyto shift registry jsou připojené na společné datové sběrnice DATA (pin SER) se společným hodinovým vstupem CLK (pin SRCLK). K výběru požadovaného shift registru jsou využity vodiče CS1, CS2 a CS3 (piny RCLK), které jsou pro jednotlivé shift registry oddělené a každý tedy využívá vlastní port mikroprocesoru. Shift registr do sebe načítá logickou hodnotu, která je v momentě náběžné hrany signálu CLK na sběrnici DATA a provádí automatický shift. Hodnoty uložené v shiftovacím registru jsou však na výstupy propsány až v momentě zaznamenání náběžné hrany na pin RCLK, tedy v momentě, kdy dojde k jejich uložení do paměťového registru. Pinů *output enable* (OE) a *clear register* (SRCLR) není užito – pin OE je uzemněn a pin SRCLR je připojen na napájecí napětí. Obsluha shiftregistrů není zajištěna periferií SPI ale programově, jelikož užitý procesor má pouze jednu tuto periferii a její sdílení pro dva účely by bylo komplikované. Blokové schéma multiplex obvodu následuje na Obr. 6.2. Blokové schéma multiplex obvodu je pak na Obr. 6.3.



Obr. 6.2. Blokové schéma multiplex obvodu v displejové jednotce

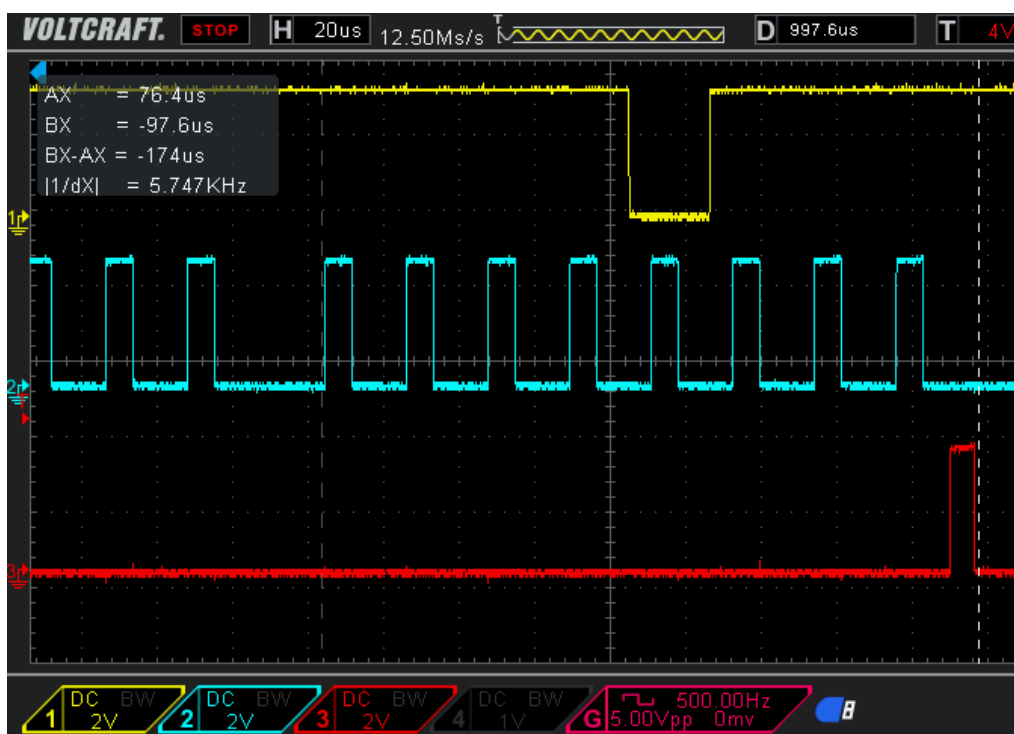


Obr. 6.3. Část schématického zapojení multiplexu displeje s 74HC595N a tranzistorovými poli

Nutno doplnit, že kompletní schéma v pravé části obsahuje celkem 10 sedmissegmentových LED displejů zapojených ve dvou řadách (6+4). Dále obsahuje celkem 3 shift registry a mnoho dalších součástek, které byly ve snaze o lepší čitelnost oříznuty. Některé jsou vidět na předchozím obrázku Obr. 6.2. Kompletní elektrické schéma viz přílohy.

Na Obr. 6.3 jsou kromě výše uvedených shift registrů s označením 74HC595N i dvě bipolární tranzistorová pole typu NPN s označením ULN2803APG. Jednotlivé piny shift registru jsou sice dostatečně proudově dimenzované, aby zvládly vždy jeden konkrétní segment přizemnit, nicméně proud protékající společným pinem GND v případě rozsvícení vícero segmentů by byl příliš velký. Shift registr je tedy užít k řízení bází tranzistorů v tranzistorovém poli přes odpovídající rezistory. Horní shift registr s označením IC2 budí unipolární tranzistory s P kanálem, které vždy připojují napětí na celý sloupec sedmissegmentových displejů.

Posloupnost kroků popisující řízení tohoto hardwaru je pak vidět na Obr. 6.4 sejmutém z osciloskopu. Na prvním kanálu osciloskopu byl zaznamenán průběh ze společné sběrnice DATA (žlutá), na druhém kanálu jsou hodinové pulzy CLK (modrá) a na třetím kanálu je vidět jeden hodinový pulz CS, konkrétně CS1. Na kanálu dva je vidět necelých 11 pulzů, přičemž mezi třetím a čtvrtým pulzem je delší mezera způsobená pulzem na pin CS3, který nebyl osciloskopem snímán. Následujících osm pulzů zajišťuje kompletní naplnění shift registru, a to binárním číslem 11110111. Ze schématu výše lze usoudit, že došlo k rozsvícení čtvrtého sloupce displejů.



Obr. 6.4. Osciloskopem naměřené časování při řízení shift registru 74HC595N

Celkový proces tisku na displej pak probíhá ve funkci *Tisk*, která vznikla v rámci této práce, následovně:

- 1) naplnění shift registru horní řady displejů,
- 2) naplnění shift registru spodní řady displejů,
- 3) naplnění shift registru pro výběr sloupce.

Nutno připomenout, že naplnění shift registru spočívá v nastavení dat na DATA doprovázených pulzy hodin na CLK (celkem tedy 8 pulzů hodin) a následným uložením dat do paměti shiftregistru pomocí pulzu na příslušný CS tak, jak je vidět na Obr. 6.4. Vstupem do funkce je pointer na pole, které bylo přijato od jednotky Master. Funkce rovněž obsahuje pole čísel, které utváří tabulku hodnot, které je nutné odeslat po DATA pro rozsvícení příslušného znaku, případně celého sloupce. Následuje ukázka kódu pro zápis dat do prostředního ze shiftregistru.

```

x=chars[Arr[k]];

for(uint8_t m=0;m<8;m++){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0,x%2);    //data on serial pin
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);    //SRCLK pulse rising edge
    x=x>>1;                                     //bite shift
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);    //SRCLK pulse falling edge
}
//pulse to store data in shiftreg's memory (lower row) (RCLK2)
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 1);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
//pulse to store data in shiftreg's memory (lower row) (RCLK2)

```

6.2 Analogové výstupy

Analogové výstupy jsou řešené pomocí dvoukanálového DA převodníku integrovaném přímo v mikroprocesoru STM32F303RE. Tento DA převodník je 12bitový, což znamená, že jeho napěťový rozsah 0-3,3 V je rozdělen do 4096 úrovní a krok je tedy přibližně 0,806 mV. Funkce, která je pro zápis hodnoty na analogový výstup užita, následuje:

```
HAL_DAC_SetValue(hdac, Channel, Alignment, Data);
```

Vstupem do této funkce je kromě handleru příslušného DA převodníku také kanál, na který chce uživatel zápis provést, zarovnání a v neposlední řadě proměnná *Data*. Tato proměnná je v rozmezí 0-4095 a udává, jaká hodnota napětí bude na výstupním pinu nastavena. Je proto nutné udělat lineární přepočítání mezi rozsahy otáček a momentů na rozsah DA převodníku, tedy 0-4095. Tento přepočítání je v případě otáček zajištěno pomocí následující rovnice:

$$Data_{RPM} = \frac{4095 \cdot (RPM - RPM_{MIN})}{RPM_{MAX} - RPM_{MIN}}$$

V rovnici je cíleně užito parametrů RPM_{MAX} a RPM_{MIN} , aby byla případná změna rozsahu pomocí softwaru snadná a rychlá. V případě výpočtu pro moment je rovnice analogická.

Výsledný kód zajišťující zápis na analogové výstupy mikroprocesoru vypadá následovně:

```

HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_1, DAC_ALIGN_12B_R,Data_RPM );
HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_2, DAC_ALIGN_12B_R,Data_TRQ );

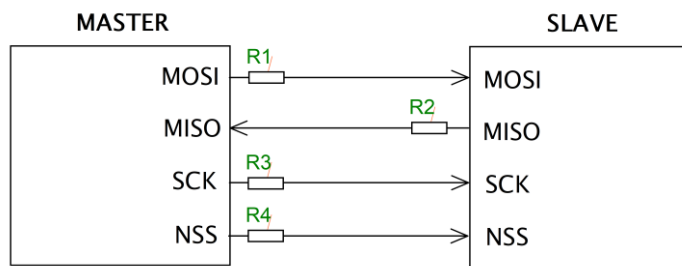
```

6.3 SPI v procesorech STM32

SPI neboli Serial Peripheral Interface je sériová komunikace typu Master – Slave. To znamená, že periferie na této sběrnici nastavená do režimu Master je nadřazená té s nastavením do režimu Slave. Nadřazenost spočívá zejména v tom, že Master typicky pomocí řídicích vodičů vybírá jednoho či více z mnoha Slave periférií, které chce posílat data, případně od které chce data přijímat zpět, a to pomocí vodiče v STM32 procesorech zvaným NSS neboli Slave select [12]. NSS je možné v nastavení v některých případech (například pokud na sběrnici je pouze jedna Slave periferie) zakázat. Krom toho Master řídí také hodinový signál SCK, jímž se celý přenos dat

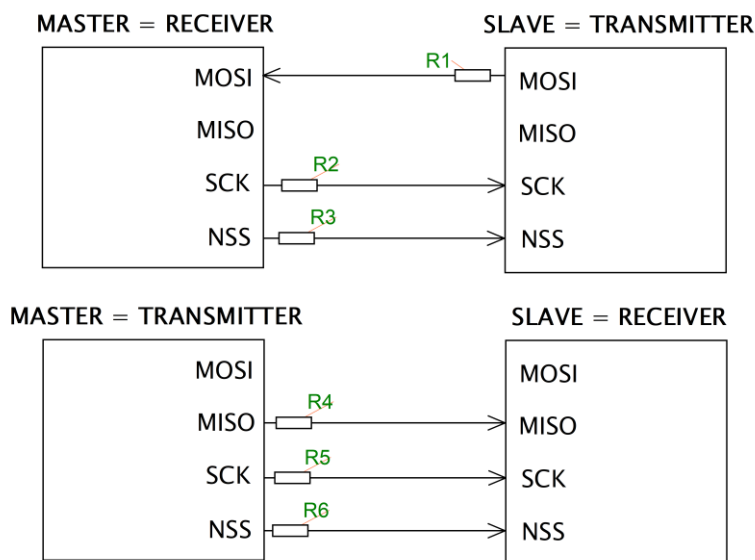
synchronizuje. Typické zapojení obsahuje jednu periférii Master a vícero periférií Slave. To však není pravidlem, protože je možné například na sběrnici připojit vícero periférií Master, takové zapojení však vyžaduje složitější řízení.

U SPI v (nejen) procesorech STM32 existuje několik možných zapojení. Jedním z nich je tzv. full-duplex zapojení, které umožňuje v jeden okamžik přenášet data oběma směry. Zapojení je pak vidět na Obr. 6.5. Zkratka MOSI znamená Master output, Slave input; zkratka MISO je Master input, Slave output a jejich funkce je zjevná ze samotného názvu [12]. Odpory slouží k potlačení vlivu odrazů na vedení a pro samotný princip fungování nejsou třeba.



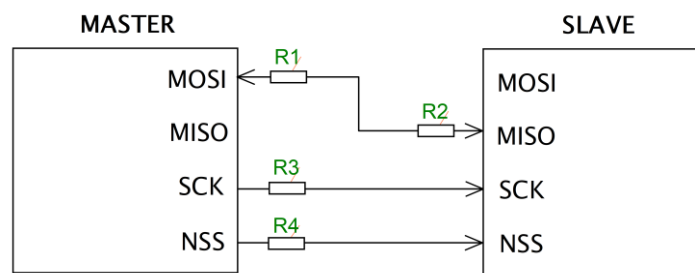
Obr. 6.5. Zapojení pro full-duplex SPI, [12], vlastní zpracování

Dalším zapojením je tzv. simplex. V tomto režimu je jedna periférie na sběrnici statický přijímač a druhá naopak vysílač. Dvě možná zapojení jsou na Obr. 6.6. Směr přenosu dat je v tomto případě pouze jednosměrný a vhodný pouze v případech, kdy není vyžadováno například zpětné posílání dat do senzoru o úspěšném přijetí dat jím odeslaných.



Obr. 6.6. Zapojení pro Simplex SPI, [12], vlastní zpracování

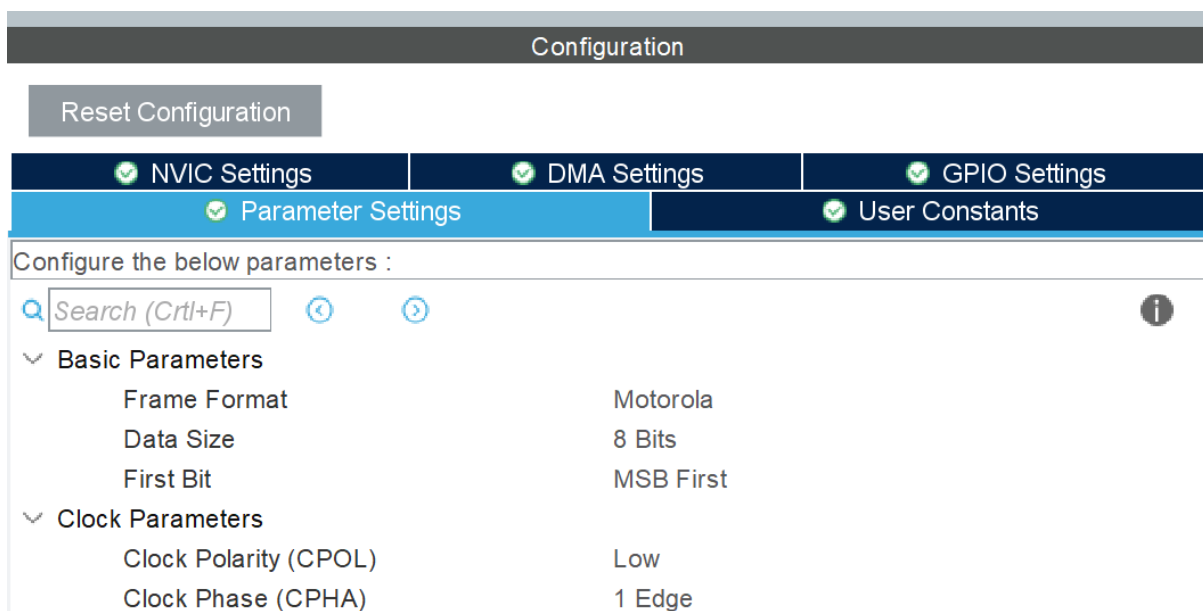
Posledním možným zapojením je zapojení half-duplex, které umožňuje v jeden moment přenášet data pouze jedním směrem, přesto je však linka obousměrná.



Obr. 6.7. Zapojení pro half-duplex SPI, [12], vlastní zpracování

Konkrétní zapojení je pak zvoleno s ohledem na požadovanou přenosovou rychlost, požadované směry komunikace nebo na množství dostupných pinů procesoru. U navržené zobrazovací jednotky byla i přesto, že požadovaný směr komunikace je pouze jednosměrný (Master -> Slave), zvolena komunikace full-duplex s povoleným NSS signálem, a to protože STM32F042 má dostatek volných pinů pro toto nastavení, které může v budoucnu usnadnit rozšíření. Zobrazovací jednotka je na sběrnici tedy připojena v režimu Slave. SPI komunikace je nastavena jako full-duplex s šířkou datového paketu 8 bitů a povoleným hardwarovým NSS signálem – jedná se tedy o stejné zapojení jako je na Obr. 6.5.

Nastavení se pak provádí pomocí STM32CubeIDE podobným způsobem, který byl popsán v podkapitole 5.1.2.2 u periferie TIM. Výše popsané nastavení pak v STM32CubeIDE vypadá pro zařízení Slave tak, jak je vidět na následujícím obrázku.



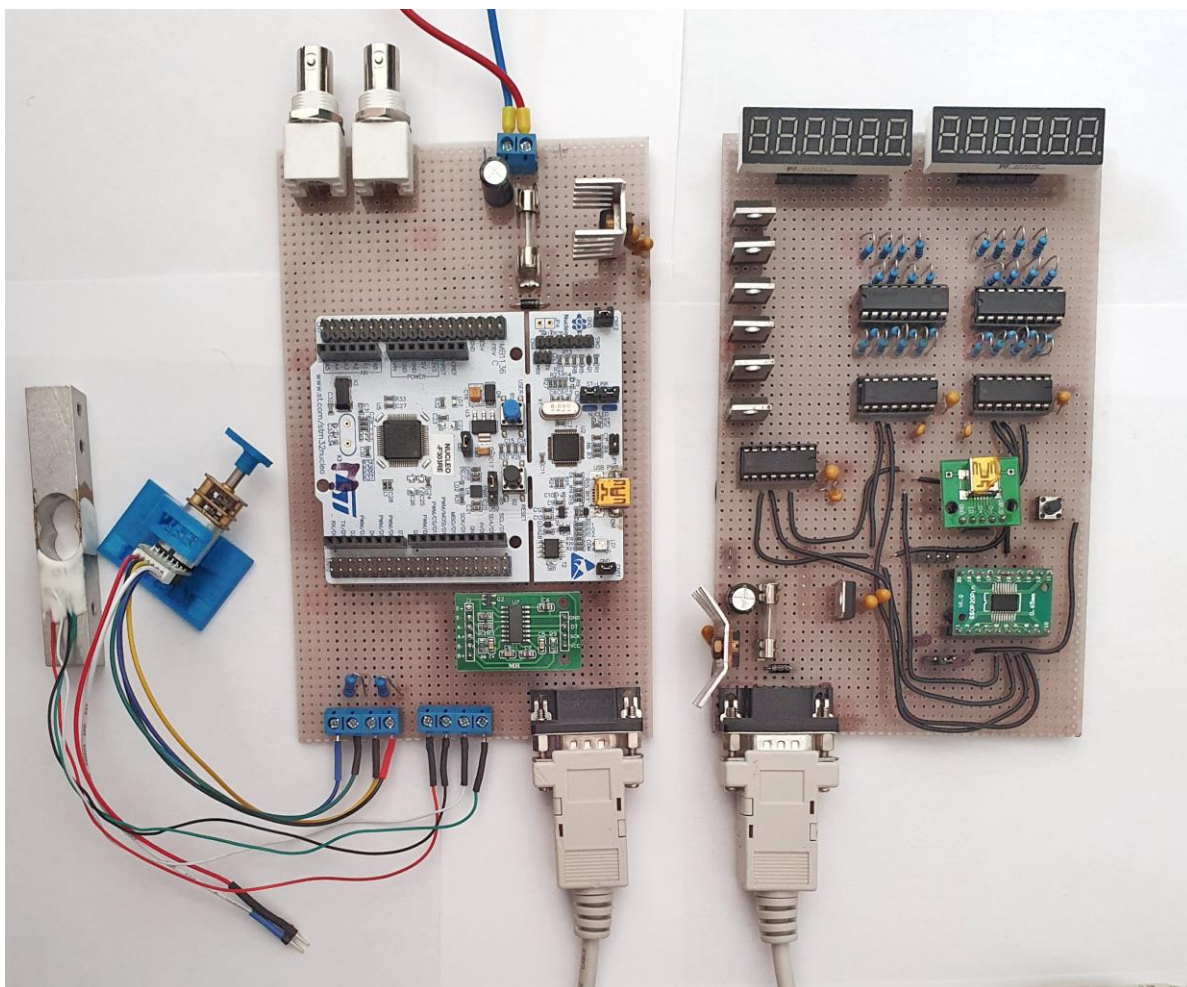
Obr. 6.8. Nastavení zařízení Slave

Samotný přenos dat mezi Masterem a Slave zobrazovací jednotkou je pak zajištěn pomocí konektoru Canon 9, skrz nějž je celá zobrazovací jednotka i napájena.

ZÁVĚR

V rámci této práce vznikl funkční prototyp jednotky pro měření a vyobrazování otáček a momentu motorů, jehož finální podoba bude sloužit v laboratoři elektrických pohonů a usnadní tak práci nejen studentům při laboratorních měřeních. Práce vytvořila ucelený pohled na vývoj hardwaru a softwaru pro tuto jednotku. Zároveň vznikl teoretický popis metod pro měření otáček pomocí kvadrurního enkodéru a měření momentu pomocí tenzometrického senzoru síly. Krom toho byly teoreticky popsány periferie u mikroprocesorů STM32, které byly pro tuto aplikaci užity. Text byl v některých případech obohacen nejen o obrázky z programovacího prostředí STM32CubeIDE s ukázkami konkrétních nastavení, ale také o kusy kódu zajišťující požadovanou funkčnost.

Vyvinutá zobrazovací jednotka obsahuje dva řádky sedmsegmentových displejů umožňující zapojení až šesti takových displejů do jednoho řádku – celkem tedy 12. Nicméně pro účely katedry elektrických pohonů a trakce byla konfigurace 6 displejů v prvním řádku a 4 displeje v řádku spodním ideální a zcela dostačující. Jednotka je navržena tak, že případné rozšíření je snadné a nenáročné. Fotografie vzniklého prototypu následuje.



Obr. 0.1. Fotografie prototypu

Zobrazovací jednotka přijímá data po SPI ve formátu čísla. Pro správnou interpretaci nečíselných znaků (jako jsou například mínus, písmeno F a další) bylo nutné těmto znakům přiřadit příslušné zobrazovací kódy – seznam podporovaných znaků a příslušných kódů se nachází v příloze této práce. Krom toho vznikly funkce *prepData()* a *Tisk()*. Funkce *prepData()* zajišťuje konverzi čísla na podporovaný formát pro zobrazovací jednotku. Funkce pracuje s čísly ve formátu *int32_t* a obsluhu speciálních znaků tedy musí uživatel zajistit sám. Funkce *Tisk()* v displejové jednotce zajišťuje správnou interpretaci přijatých dat po SPI a jejich následné zakódování do binárních čísel, které jsou v rámci této funkce rovněž odesílány na výstupní piny mikroprocesoru a přijímány do shiftregitrů.

Pro návrh DPS a kreslení schémat byl užit program EAGLE od Autodesk. Pro tyto účely byly vytvořeny tři knihovny – pro Nucleo STM32F303RE, STM32F042 upevněném na SSOP 20Pin a LED sedmissegmentový displej HD-AD16RD. Tyto knihovny jsou rovněž k nalezení v příloze této práce společně i s rendery osazené DPS.

Sestavený prototyp na fotografii je plně funkční a architekturou je shodný s provedením konečným. Na konečném provedení bude autor nad rámec zadání nadále usilovně pracovat, aby jej bylo možné co v nejbližší době zapojit do výuky na ČVUT FEL.

KAPITOLA 7: LITERATURA

- [1] WILLIAMS, Bryon. Load Cell Types & How They Work. *Montalvo.com* [online]. [cit. 2021-03-21]. Dostupné z: <https://www.montalvo.com/article-library/load-cell-types/>
- [2] ĎAĎO, Stanislav a Marcel KREIDL. *Senzory a měřicí obvody*. Vyd. 2. Praha: Vydavatelství ČVUT, 1999. ISBN 80-010-2057-6.
- [3] CHELNER, Herbert a Robert A. MUELLER. Load Cell Linearization using Micron Instruments' Semiconductor Strain Gages. *Www.microninstruments.com* [online]. 2018 [cit. 2021-03-21]. Dostupné z: <https://www.microninstruments.com/news/4>
- [4] HAASZ, Vladimír, Jan HOLUB, Michal JANOŠEK, Petr KAŠPAR a Vojtěch PETRUCHA. *Elektrická měření: přístroje a metody*. 3. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2018. ISBN 978-80-01-06412-2.
- [5] Modul pro váhové senzory s HX711. In: *Gme.cz* [online]. Křižíkova 77, 186 00 Praha 8 [cit. 2021-03-22]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.775-019.1.pdf>
- [6] 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales. In: *Sparkfun.com* [online]. P.R. China [cit. 2021-03-22]. Dostupné z: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- [7] QUASDORF, Joachim. *Sine/Cosine-to-Digital Conversion: Methods and Design Challenges* [online]. 2014 [cit. 2021-03-21]. Dostupné z: <https://www.edn.com/sine-cosine-to-digital-conversion-methods-and-design-challenges/>
- [8] BATES, Martin. *Interfacing PIC Microcontrollers: Power Outputs* [online]. 2014, , 7213-259 [cit. 2021-03-22]. ISBN 9780080993638,. Dostupné z: <https://doi.org/10.1016/B978-0-08-099363-8.00007-8>
- [9] Application Note - Quadrature Encoder Counter with SPI Bus Interface. In: *Dialog-semiconductor* [online]. 2019 [cit. 2021-03-06]. Dostupné z: <https://www.dialog-semiconductor.com/>
- [10] General-purpose timer cookbook for STM32 microcontrollers. In: *St.com* [online]. 2019 [cit. 2021-03-22]. Dostupné z: https://www.st.com/resource/en/application_note/dm00236305-general-purpose-timer-cookbook-for-stm32-microcontrollers-stmicroelectronics.pdf
- [11] RM0316 Reference manual. In: *St.com* [online]. 2017 [cit. 2021-03-22]. Dostupné z: https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303x6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf
- [12] STM32 Serial Peripheral Interface. In: *St.com* [online]. [cit. 2021-03-22]. Dostupné z: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/3e/ee/cd/b7/84/4b/45/ee/STM32F7_Peripheral_SPI/files/STM32F7_Peripheral_SPI.pdf/_jcr_content/translations/en.STM32F7_Peripheral_SPI.pdf

- [13] SIRAVANJITH. Electronic and pneumatic Load cells. In: *Automationforum.co* [online]. [cit. 2021-03-21]. Dostupné z: <https://automationforum.co/electronic-and-pneumatic-load-cells/>
- [14] What are Quadrature Encoders. *Creative-robotics.com* [online]. [cit. 2021-03-21]. Dostupné z: <http://www.creative-robotics.com/quadrature-intro>
- [15] Quadrature Encoding. *Www.speedgoat.com* [online]. [cit. 2021-03-21]. Dostupné z: https://www.speedgoat.com/help/slrt/page/io_main/refentry_quadrature_usage_notes
- [16] QUINTÁNS, Camilo, José FARIÑA a Jorge MARCOS-ACEVEDO. *Improving the performance of incremental encoders with conditioning circuits based on FPGA, Measurement* [online]. 2016, , 1-2 [cit. 2021-03-22]. ISSN 0263-2241. Dostupné z: <https://doi.org/10.1016/j.measurement.2016.04.031>
- [17] CD74HC595 8-bit shift registers with 3-state output registers. In: *Gme.cz* [online]. 2004 [cit. 2021-03-22]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.425-118.1.pdf>
- [18] PETERSON, Mark. Gray Code's Error Reduction with Encoders. *Wolfram.com: Wolfram Demonstrations Project* [online]. 2011 [cit. 2021-03-22]. Dostupné z: <https://demonstrations.wolfram.com/GrayCodesErrorReductionWithEncoders/>
- [19] POLLEFLIET, Jean. Power Electronics - Drive Technology and Motion Control. *Academic Press* [online]. 2018, , 181-1834 [cit. 2021-03-22]. ISBN 9780128146415. Dostupné z: <https://doi.org/10.1016/B978-0-12-814641-5.50003-2>