

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Vývoj řídicího systému polohování pro mikrotomografii

Vít Machanec

Vedoucí: Ing. Vojtěch Petrucha, Ph.D.
Studijní program: Kybernetika a robotika
Květen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Machanec** Jméno: **Vít** Osobní číslo: **474382**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Vývoj řídicího systému polohování pro mikrotomografii

Název bakalářské práce anglicky:

Development of a control system for microtomography

Pokyny pro vypracování:

- Seznamte se s funkcí polohovacího systému používaného pro mikrotomografická měření s ohledem na možnosti jeho řízení, popište jej a uveďte požadavky na jeho vylepšení.
- S použitím vývojového kitu s procesorem STM32, s obvody L6470 a vhodných senzorů sestavte platformu demonstrátoru řízení polohování.
- Navrhněte a realizujte firmware a komunikační protokol demonstrátoru.
- Navrhněte a realizujte základní ovládací software demonstrátoru.

Seznam doporučené literatury:

- [1] Steve Heath: Embedded Systems Design, Newnes 2003
- [2] Franklin, Powell, Emami-Naeini: Feedback Control of Dynamics Systems. Prentice Hall, USA
- [3] Jacob Fraden: Handbook of Modern Sensors, Springer 2004
- [4] Mark I. Montrose: EMC and the Printed Circuit Board: Design, Theory, and Layout Made Simple, IEEE 1999
- [5] Vít Záhlava: Návrh a konstrukce DPS, Ben 2010

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vojtěch Petrucha, Ph.D., 13138

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

Datum zadání bakalářské práce: **20.01.2020**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2020/2021

Ing. Vojtěch Petrucha, 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

Tímto děkuji Ing. Vojtěchu Petruchovi, Ph.D. za profesionální vedení mé bakalářské práce. Dále bych chtěl poděkovat kolegům Milanu Malichovi a Janu Žemličkovi za konzultace ohledně vývoje systému a funkce mikrotomografu. Také bych chtěl poděkovat své rodině za podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 22. května 2020

Abstrakt

Cílem této bakalářské práce je návrh řídicího hardware a software pro rozvoj stávajícího polohovacího systému v mikrotomografu v Ústavu technické a experimentální fyziky (ÚTEF) ČVUT v Praze. V rámci práce je popsán stávající systém polohování a možnosti jeho vylepšení. Na jejich základě byl vybrán hardware pro nové řízení a navrhnout a implementován software pro ovládání krokových motorů použitých jako pohonu k polohování. Systém byl s jedním motorem úspěšně otestován na modelu. Vyvinutý systém je možné využít obecně k lineárnímu polohování s přesností na 2 μm . Systém lze rozšířit k řízení až osmi krokových motorů.

Klíčová slova: řízení krokového motoru, zpětná vazba, senzory polohy, mikrotomografie

Vedoucí: Ing. Vojtěch Petrucha, Ph.D.

Abstract

The goal of this bachelor thesis is design of control hardware and software for upgrade of the current positioning system in the Institute of Experimental and Applied Physics (IAEP) CTU in Prague. In the thesis the current system and possibilities for its upgrade are described. With respect to them, hardware for new control was selected and software for control of stepper motor, selected as propulsion for positioning, was developed. System was successfully tested with one motor on a model. Developed system is widely usable for positioning with 2 micron accuracy. System can be further extended for control of up to 8 stepper motors.

Keywords: stepper motor control, feedback, position sensors, microtomography

Title translation: Development of a control system for microtomography

Obsah

1 Úvod a motivace	1	7 Závěr	39
2 Přístroj pro mikrotomografická měření	3	Literatura	41
2.1 Popis přístroje	3	A Příkazy a registry k ovládní motoru	43
2.2 Princip funkce	4		
2.2.1 Detektor záření	4		
2.3 Stávající polohovací systém	5		
2.4 Možnosti polohování	6		
2.4.1 Polohovací mechanismy	6		
2.4.2 Pohony pro polohování	7		
2.4.3 Senzory polohy	8		
2.4.4 Příklady polohovacích mechanismů	8		
3 Návrh nové koncepce	9		
3.1 Komponenty pro nový systém	9		
3.1.1 Motory a jejich řízení	10		
3.1.2 Senzory polohy	16		
3.1.3 Paměti motorů	20		
3.2 Schéma nového systému	21		
4 Firmware	23		
4.1 Komunikační protokol	23		
4.1.1 Příjem zpráv	24		
4.1.2 Části zprávy	24		
4.2 Sledování polohy	25		
4.3 Obsluha EEPROM	27		
4.3.1 Zápis	27		
4.3.2 Čtení	27		
5 Ovládací software	29		
5.1 Připojení ke kitu	29		
5.2 Polohování	30		
5.3 Další příkazy	31		
5.4 Zobrazení přijatých zpráv	32		
6 Testování	35		

Obrázky

2.1 Rozmístění hlavních částí uvnitř mikrotomografu	3	4.1 Struktura zpráv posílaných mezi PC a Nucleo F401RE	23
2.2 Vnější pohled na mikrotomograf v ÚTEF ČVUT	4	5.1 Uživatelské rozhraní pro ovládání demonstrátoru	29
2.3 Struktura Timepix detektoru[1] ..	4	5.2 Panel pro nastavení polohy.....	30
2.4 Pohybové schéma přístroje	5	5.3 Příkazy a parametry k dispozici	31
2.5 Ukázka provedení posuvného pohybu v přístroji	6	5.4 Ukázka výpisu zpráv přijatých z kitu	33
3.1 Princip krokového motoru[6] ...	10	6.1 Model nového polohovacího systému. 1-IHM02A1 připojená na STM32F401RE, 2-krokový motor, 3a,3b-koncové spínače, 4-magnetická páska, 5-vedení signálů z AS5311 kryté řetězem, 6-lineární vedení, 7-EEPROM	35
3.2 Zapojení bipolárního krokového motoru[6]	10	6.2 Uchycení senzoru AS5311 na posuvnou desku modelu	36
3.3 Realizace půl kroku[6]	11	6.3 Výsledky testování přesnosti polohovacího modelu	37
3.4 Řízení s plným krokem a mikrokrokováním[3]	11	6.4 Velikosti odchylek od koncových poloh při měření opakovatelnosti modelu polohování	38
3.5 Blokové schéma L6470[3]	12	A.1 Přehled příkazů k dispozici pro ovládání motoru pomocí L6470[3] .	43
3.6 SPI v řetězovém uspořádání[3] ..	13	A.2 Přehled parametrů k dispozici pro ovládání nebo sledování motoru[3]	44
3.7 Časový diagram SPI sběrnice[3]	13	A.3 Struktura příkazu Run[3]	45
3.8 Příkaz se 3 bajtovým argumentem[3]	13	A.4 Struktura příkazu SoftStop[3] ..	45
3.9 Odpověď s 3 bajtovým argumentem[3]	14	A.5 Struktura příkazu HardStop[3] .	45
3.10 Struktura příkazu Move[3] ...	14	A.6 Struktura příkazu GetParam[3]	45
3.11 Struktura příkazu SetParam[3]	15	A.7 Struktura příkazu GetStatus[3].	45
3.12 Struktura STATUS registru[3] .	16		
3.13 Připojení koncového spínače k pinu nuclea	17		
3.14 Vnitřní stavba AS5311[4]	17		
3.15 Výstupy AS5311 v závislosti na poloze senzoru[4]	18		
3.16 Zprovoznění senzoru AS5311 ..	19		
3.17 Schéma pro desku plošných spojů	20		
3.18 Deska plošných spojů k senzoru AS5304	20		
3.19 Rozložení pinů paměti[5]	21		
3.20 Blokové schéma modelu nově navrženého řídicího systému.....	22		

Tabulky

2.1 Přehled polohování.....	6
3.1 Vybrané nastavitelné parametry	15
3.2 Zohledňované parametry senzoru AS5311	19
3.3 Ukládané parametry motoru ...	21
4.1 Přehled příkazů zpracovávaných firmwarem	26

Kapitola 1

Úvod a motivace

Tato bakalářská práce se zabývá návrhem řídicího hardware a software pro rozvoj stávajícího polohovacího systému pro mikrotomograf v Ústavu technické a experimentální fyziky (ÚTEF) ČVUT v Praze. Stávající systém je řízen elektronikou postavenou na driverech DRV8821 Texas Instruments pro krokové motory s přesností řízení jen na 1/8 kroku. Systému také chybí zpětná vazba o jeho vnitřním uspořádání. Kvůli vnějšímu plášti pro stínění rentgenových paprsků nemá operátor přístroje při nastavování měřící aparatury vizuální kontakt s jeho vnitřními částmi a je pro něj těžké odhadovat jejich polohy. To je pro něj značně nepohodlné a do jisté míry i nebezpečné, protože může z počítače poslat příkaz k posunu, který povede ke srážce částí uvnitř přístroje nebo k vyjetí z koncové polohy a jeho poškození.

V rámci práce byly pro vylepšení systému vybrány hardware součástky od firem ASM a STMicroelectronics, které zajistí operátorovi zpětnou vazbu a umožní řídit motory s přesností na 1/128 kroku a polohovat tak měřící aparaturu s vyšší opakovatelností. Důležitou součástí práce byl vývoj software pro řízení motoru. V závěru práce byl pro ověření funkčnosti řízení sestaven a otestován model polohování s jedním motorem. Nový systém dále implementuje bezpečnostní prvky a jejich obsluhu v koncových polohách dráhy demonstrátoru.

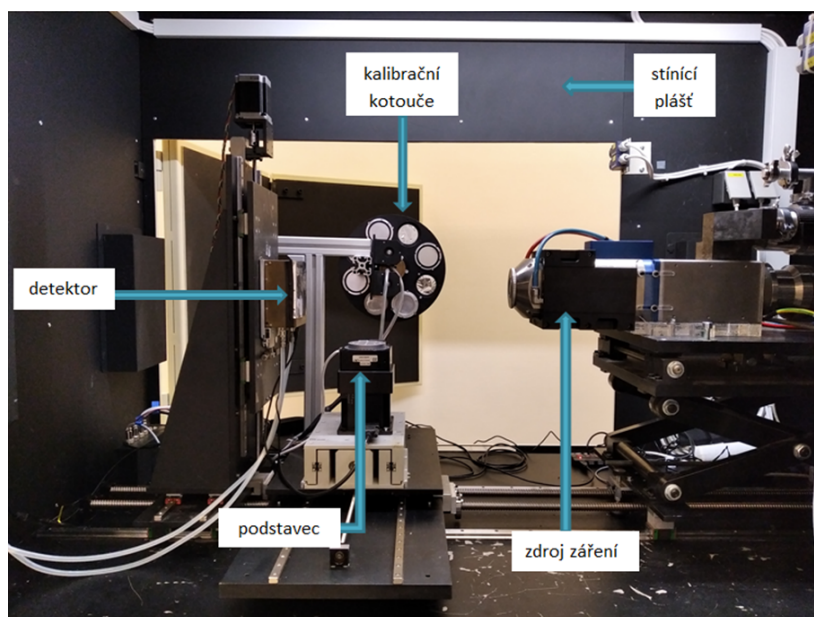
Vyvinutý systém je využitelný nejen k polohování pro účely mikrotomografu, ale lze ho aplikovat na řízení široké škály krokových motorů. Díky zvolené hardware platformě lze systém rozšířit k řízení až 8 krokových motorů.

Kapitola 2

Přístroj pro mikrotomografická měření

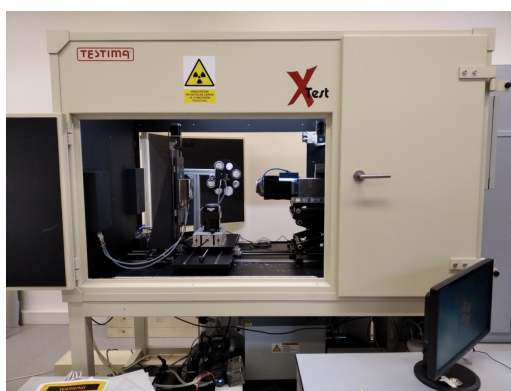
2.1 Popis přístroje

Mikrotomograf v ÚTEF ČVUT se skládá z pěti hlavních částí. Těmito částmi jsou rentgenka, podstavec na zkoumaný objekt, detektor ionizujícího záření, kalibrační kotouče a vnější opláštování. Rozmístění jednotlivých částí v přístroji je zachyceno na obrázku 2.1.



Obrázek 2.1: Rozmístění hlavních částí uvnitř mikrotomografu

Pro přesnější popis přístroje byly zavedeny souřadnicové osy tak, jak jsou zakresleny na schématu 2.4. Výška přístroje je brána jako rozměr ve směru osy z, šířka ve směru osy x a délka ve směru osy y. Na obrázku 2.2 je uveden vnější pohled na přístroj.



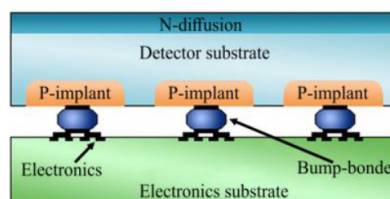
Obrázek 2.2: Vnější pohled na mikrotomograf v ÚTEF ČVUT

2.2 Princip funkce

Princip mikrotomografického měření spočívá v ozařování zkoumaného vzorku před detektorem ionizujícího záření. Jako zdroj záření slouží rentgenka FeinFocus FXE-160.51 s velikostí vyzařovacího bodu menší než $1\ \mu\text{m}$, což umožňuje rozlišení detailů na úrovni mikrometrů. Při měření se na detektoru zaznamenává útlum rentgenového záření v měřeném vzorku. Informace z detektoru se pošle do ovládacího počítače, kde se z ní složí obraz, který se dále zpracovává a zkoumá. Princip funkce detektoru záření je rozveden níže.

2.2.1 Detektor záření

Detektor je složen z matice 4×5 hybridních pixelových detektorů Timepix. Každý Timepix se skládá z matice pixelů 256×256 s roztečí $55\ \mu\text{m}$, citlivá plocha je tedy přibližně $14 \times 14\ \text{mm}$. Na obrázku 2.3 je zobrazena struktura pixelu detektoru. Každý pixel má svou vyhodnocovací elektroniku. Při dopadu



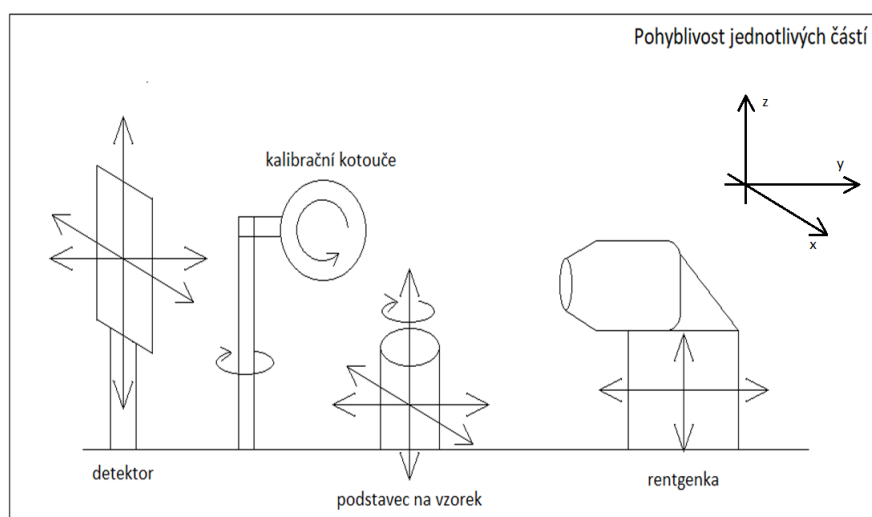
Obrázek 2.3: Struktura Timepix detektoru[1]

záření na povrch pixelu je zaznamenán náboj, který je zesílen zesilovačem, porovnán komparátorem s uživatelem nastavitelnou prahovou hodnotou, a pokud je hodnota překročena, je událost zaznamenána a čítač inkrementován o jedna. Detektor je hybridní, protože senzory pixelů jsou od jejich vyhodnocovacích elektronik odděleny pájecími mikrokuličkami (*bump bonding*), a tak je možné pro povrch detektoru použít i jiný materiál s vyšší detekční účinností pro zvolené záření než křemík (např. CdTe).

Aby bylo možné vytvořit velkou souvislou detekční plochu, byly detektory Timepix ve spolupráci VTT Finland a ÚTEF ČVUT dále vylepšeny tak, že byl ze stran detektoru odstraněn 500 μm široký ochranný prstenec (*guard ring*). Na jedné ze stran je však stále vyvedena elektronika čipu. Jednotlivé detektory jsou tedy do matice vyskládány tak, že jsou lehce nakloněny, aby první řada překrývala elektroniku druhé řady atd.

2.3 Stávající polohovací systém

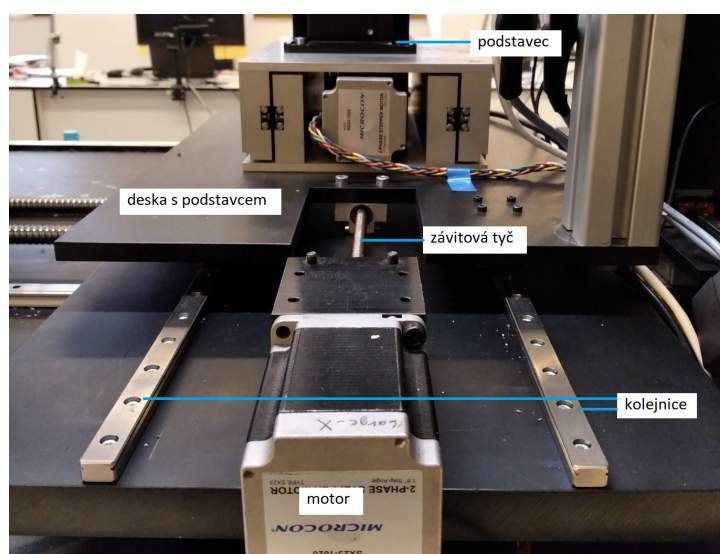
V přístroji se s většinou částí pohybuje posouváním po lineárním vedení. Na schématu 2.4 je naznačeno, v jakých směrech lze pohybovat s hlavními částmi přístroje. Manuálně je v přístroji polohována rentgenka a stojan



Obrázek 2.4: Pohybové schéma přístroje

pro kalibrační kotouče. Ostatní části se posouvají pomocí závitových tyčí otáčených krokovými motory, čímž se rotační pohyb motorů převádí na lineární posuv příslušné části. Na obrázku 2.5 je ukázka provedení posuvného pohybu v přístroji na polohování podstavce ve směru osy x. V přístroji jsou použity bipolární krokové motory od firmy Microcon. K řízení motorů jsou použity drivery DRV8821 od firmy Texas Instruments, které je umožňují řídit s přesností na 1/8 kroku. Krokové motory ovládá operátor pomocí aplikace na PC.

V tabulce 2.1 je uveden přehled polohovacích možností jednotlivých částí přístroje. Rentgenka, detektor a podstavec jsou ve směru osy y na jedné kolejnici a dohromady je jejich polohovací rozsah přibližně 40 cm. V ostatních směrech odpovídá rozsah pohybu zhruba délce lineárního vedení.



Obrázek 2.5: Ukázka provedení posuvného pohybu v přístroji

část přístroje		zdroj zá- ření	podstavec	detektor	kalibrační kotouče
počet stupňů volnosti		2	4	3	3
přibližné délky lin. vedení[cm]	osa x	145	145	145	-
	osa y	-	60	20	-
	osa z	40	5	50	-
rotace		ne	kolem osy z	ne	kolem osy z a vlast- ních os
pohon		manuální	elektrický	elektrický	elektrický, manuální

Tabulka 2.1: Přehled polohování

2.4 Možnosti polohování

V této sekci jsou uvedeny příklady mechanických způsobů polohování, dále pohony využívané pro polohování, senzory polohy a na konci krátce zmíněny příklady dostupných polohovacích systémů na trhu.

2.4.1 Polohovací mechanismy

Jeden ze způsobů polohování, využitý i v mikrotomografu na ÚTEF ČVUT, je využití kombinace elektromotoru, závitové tyče a lineárního vedení. Hřídel se napojí pružnou spojkou na závitovou tyč a ta se na druhém konci upevní v domečku. Tyč může být podle kvality běžná s trojúhelníkovým profilem

závitů, trapézová nebo kuličková. Na tyč je nasazena matice, ke které je přichycena část určená k polohování. Tato část je nasazena na lineární vedení složené ze dvou kolejnic a vozíčků, ke kterým je připevněna.

Další možností je nasadit na hřídel motoru řemenici a použít k polohování řemen. Ten se v jednom místě přichytí na polohovanou část nasazenou na lineární vedení a točením motoru se opět převádí rotační pohyb na posuvný. Řemen lze využít i k převodu otáček a např. pomocí toho zmenšit nejmenší lineární posun.

■ 2.4.2 Pohony pro polohování

■ Krokové motory

Tento druh motorů je použit v mikrotomografu pro polohování jeho částí. Mezi výhody krokových motorů patří opakovatelnost, velký točivý moment při nízkých otáčkách, aretační moment i při absenci napájení, bezúdržbovost a také dostupnost a nízká pořizovací cena. Jsou řízeny v otevřené smyčce, což zjednodušuje jejich ovládání, ale nelze tak sledovat polohu částí uvnitř přístroje. Motory tak mohou ztratit při provozu kroky (např. kvůli mechanickému přetížení), aniž by to bylo detekováno, a polohování se tím stane nepřesným, dokud se systém neresetuje.

Jiné nevýhody jsou hlučnost, klesající točivý moment při vyšší rychlosti, přehřívání při dlouhém provozu a vibrace. Nevýhody lze kompenzovat, např. ztrátu kroků a následné nepřesnosti polohování přidáním zpětné vazby a hlučnost nižší provozní rychlostí. Krokové motory a jejich řízení jsou dále popsány v kapitole 3.1.1.

■ Servomotory

Další přístup k řízení polohování představují servomotory. Ty se obecně kromě motoru skládají navíc ze senzoru, který je na motor napojen a měří jeho pohyb. Senzor zprostředkovává zpětnou vazbu do řídicího obvodu, a ten podle ní upravuje signály, kterými řídí motor. Díky tomu lze velmi přesně nastavovat pozici, rychlost nebo i moment motoru. Proto se servomotory používají tam, kde je vysoká přesnost při polohování zásadní. Jsou například v kloubech průmyslových robotů nebo CNC obráběcích strojích.

Mezi nevýhody servomotorů patří potřeba přesně vyladit regulátor polohování v driveru a tím složitější instalace, menší točivý moment při nízkých otáčkách nebo složitá řídicí elektronika a s tím související vysoká pořizovací cena. Výhodami jsou přesnost, velký točivý moment při vysokých otáčkách, vysoký výkon, spolehlivost, rychlost i zrychlení, hladší běh nebo tichost při provozu.

Řada zmíněných vlastností servomotorů závisí také na použitém motoru. Serva v sobě mohou mít např. kartáčový stejnosměrný motor, který je jedno-

Kapitola 3

Návrh nové koncepce

Při návrhu řídicího systému polohování se uvažovalo s tím, že se použijí bipolární krokové motory, které už byly v ústavu k dispozici. Dále se tedy rozhodovalo o tom, jakými prostředky budou řízeny a jakými senzory se realizuje zpětná vazba.

3.1 Komponenty pro nový systém

Mechanická část systému je zachována tak, jak byla zmíněna v kapitole 2.4.1, využitá závitová tyč je kuličkového typu.

Pro řízení motorů byla vybrána kombinace vývojového kitu Nucleo-F401RE a expanzivní desky X-Nucleo-IHM02A1 s dvěma řídicími obvody L6470. Tento hardware byl zvolen pro svou přesnost a spolehlivost řízení a možnosti napojení až 4 expanzivních desek na jeden kit. Tak je možné rozšířit kapacitu řídicího systému na ovládání až 8 motorů. Detekce koncových poloh byla ošetřena koncovými spínači a pro sledování změny polohy použit inkrementální magnetický senzor AS5311.

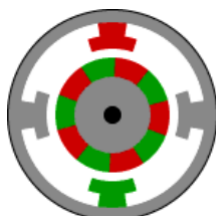
K motoru byla také přiřazena paměť EEPROM, kam je možné uložit jeho vybrané parametry využívané pro řízení. Systém se tak stává méně závislým na konkrétním motoru a uživatelsky jednodušším při zapojování více motorů do systému, protože si systém po zapnutí z pamětí vyčte, na jakých adresách jsou které motory k polohování příslušných částí a nedojde k záměně. Paměti pro parametry motorů obsahuje i stávající řídicí systém a bylo rozhodnuto, že se tato vlastnost ponechá.

Polohy částí přístroje jsou nastavovány ovládacím programem spuštěným v PC. Vývojový kit Nucleo je k PC připojen kabelem USB mini a komunikuje s ním přes virtuální UART. Firmware a ovládací software řídicího systému jsou popsány v kapitolách 4 a 5. Dále jsou popsány komponenty, které byly vybrány pro realizaci systému.

3.1.1 Motory a jejich řízení

Krokový motor

Použitý pohon pro polohování je 2-fázový bipolární krokový motor. Obecně je princip funkce krokového motoru zachycen na obrázku 3.1. Motor se skládá

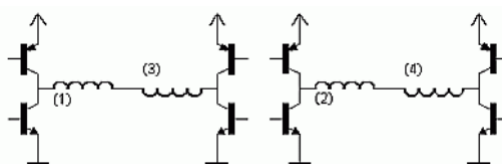


Obrázek 3.1: Princip krokového motoru[6]

z rotoru a statoru. Rotor je tvořen permanentně magnetovaným materiálem, na jehož povrchu se v pruzích podélných s hřídelí motoru střídají severní a jižní póly. Rotor je obklopen státorem, který je tvořen vinutými cívkami, v nichž se při průchodu proudu generuje magnetické pole. To otáčí rotorem silou

$$F = BIL,$$

kteřá je úměrná velikosti magnetického pole, proudu procházejícím cívkou a délkou cívk. Obrázek 3.1 zároveň ukazuje, jak funguje bipolární krokový motor. V takovém motoru prochází proud vždy dvěma protilehlými cívkami tak, aby se na nich generovala opačně orientovaná magnetická pole. Proud cívkami je řízen 2 H-můstkami (obrázek 3.2). Aktivní cívky k sobě přitáhnou



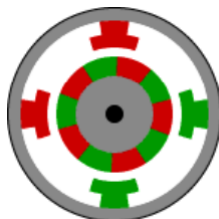
Obrázek 3.2: Zapojení bipolárního krokového motoru[6]

nejbližší k nim opačné póly na rotoru, čímž jím pohnou. Tento pohyb je jedním krokem. Cívkami jsou střídavě aktivovány a deaktivovány a takto lze s motorem postupně otáčet. Počet kroků potřebných pro 1 otáčku rotoru o 360° je roven počtu pólů na rotoru motoru. Např. pro obrázek 3.1 je to 10 kroků na 1 otáčku. Úhlové rozlišení pohybu takového motoru je pak dáno jako

$$\text{úhlové rozlišení } [^\circ] = \frac{360^\circ}{\text{počet pólů rotoru}} = \frac{360^\circ}{10} = 36^\circ.$$

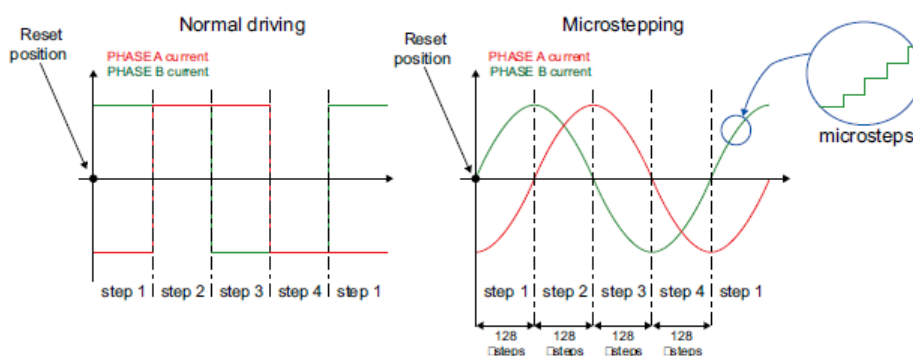
Pokud je pak motor řízen 2-fázově, generují shodně orientované magnetické pole vždy dvě sousední cívkami. Oproti 1-fázovému řízení, kdy generuje magnetické pole současně pouze 1 cívka, dosahuje toto řízení vyššího kroutícího

momentu, ale má dvojnásobnou spotřebu. Střídáním kroků s 1-fázovým a 2-fázovým řízením lze motor řídit s polovičním krokem. Situace je naznačena na obrázku 3.3. Takto lze dosáhnout dvojnásobné přesnosti polohování.



Obrázek 3.3: Realizace půl kroku[6]

Další možností jak zvýšit přesnost polohování krokového motoru je mikrokrokování. Vlevo na obrázku 3.4 je znázorněn průběh proudů vinutí A a B při 2-fázovém řízení. Vinutími střídavě prochází opačně a souhlasně orientované proudy. Při uplatnění mikrokrokování generuje řídicí logika do vinutí motoru dva sinusové signály posunuté vůči sobě o 90° . Situace je znázorněna vpravo na obrázku 3.4. Podle toho, jak jemně jsou sinusovky generovány, lze



Obrázek 3.4: Řízení s plným krokem a mikrokrokováním[3]

nastavovat míru mikrokrokování. Na obrázku jsou čtvrtiny period sinusovek rozdělené na 128 mikrokroků. Motor by tímto signálem byl řízen s přesností $1/128$ kroku. S dosažením vysoké přesnosti a plynulejšího pohybu bohužel motor ztrácí na kroutícím momentu.

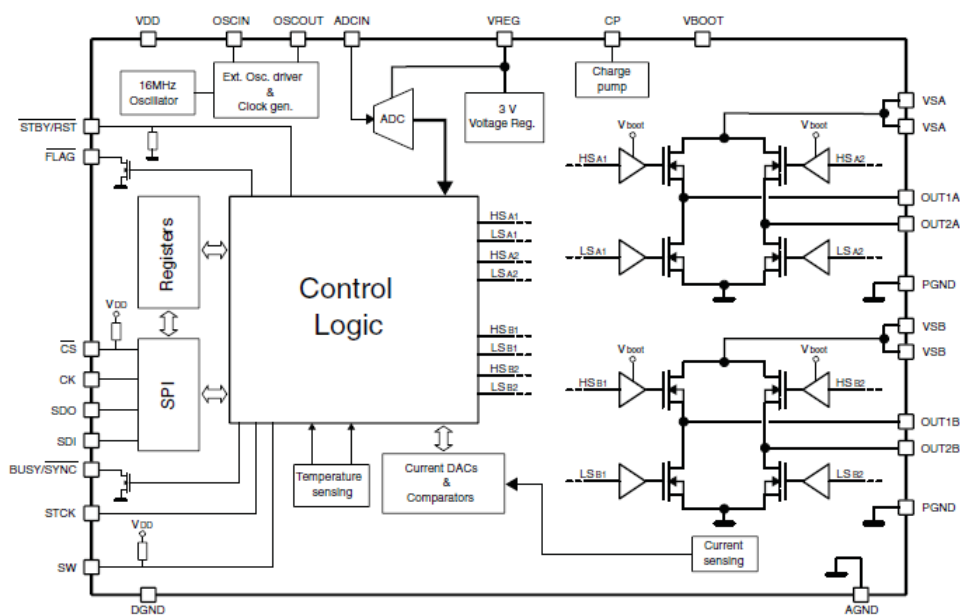
■ Řízení pomocí L6470

Obvod L6470 je motorový driver vhodný pro řízení 2-fázových bipolárních krokových motorů. Níže jsou vybrané základní vlastnosti driveru.

- operační napětí 8-45V, špičkový proud 7A,
- programovatelná rychlost a polohování,

- mikrokrokování s rozlišením až 1/128 kroku,
- implementovaná ochrana proti vysokému proudu a přehřátí,
- SPI rozhraní.

Na obrázku 3.5 je blokové schéma driveru. V pravé části schématu jsou pod

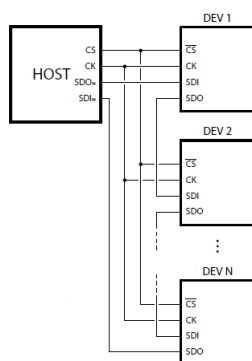


Obrázek 3.5: Blokové schéma L6470[3]

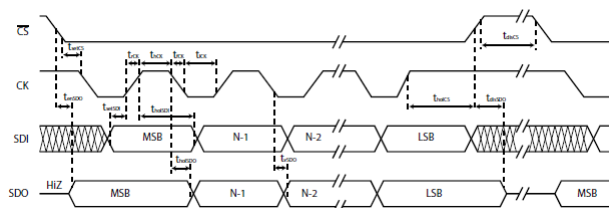
sebou 2 H-můstky, které jsou spínány řídicí logikou pro polohování motoru. Cívky statoru jsou připojeny k vývodům OUT1A, OUT2A a OUT1B, OUT2B. Vlevo od řídicí logiky jsou programovatelné registry, přes něž lze nastavit např. maximální rychlost otáčení nebo akceleraci. Pod registry je rozhraní pro komunikaci mezi nucleo deskou a driverem po SPI sběrnici s rychlostí 5-Mbit/s.

Sběrnice je v řetězovém uspořádání (obrázek 3.6), takže signály SDO a SDI nejsou zavedeny mezi master zařízením a všemi ostatními. Namísto toho je SDO signál každého slave vyveden do SDI následujícího slave. Signál SDI prvního slave je napojen na SDO master. Signál SDO posledního slave vede zpět do master, a tak uzavírá řetězec. Na obrázku 3.7 je časový diagram vysílání po SPI sběrnici.

Obvody L6470 jsou zapojené ve dvojici na expanzivní desce X-Nucleo IMH02A1 připojené k nucleo. To znamená, že v SPI řetězci je vždy 1 nucleo jako master a 2 motor drivery. Pokud se k nucleo připojí více expanzivních desek (lze až 4), je třeba aby na každé byl zkratovaný odlišný pájecí most. Výchozí zkratovaný most je SB23 a tím je určeno, že pro vysílání CS signálu z nucleo musí být použit pin PA4. Další mosty na výběr jsou SB7, SB8 a SB9, které pro CS signál předurčují piny PA10, PB6 a PB4.[7]. Pokud je tedy



Obrázek 3.6: SPI v řetězovém uspořádání[3]



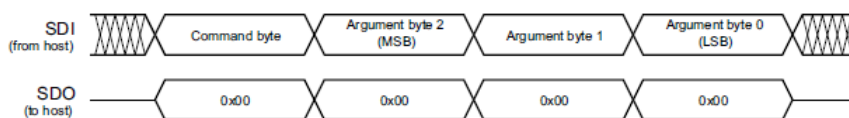
Obrázek 3.7: Časový diagram SPI sběrnice[3]

k nucleu připojeno více než 1 expanzivní deska, je třeba při vysílání příkazu použít správný pin pro CS signál.

■ Příkazy a registry

Celkově má uživatel k ovládání motoru driverem L6470 k dispozici 19 příkazů a 25 registrů, jimiž může měnit parametry řízení. Přehled všech příkazů a parametrů, které lze nastavit nebo sledovat, je v příloze A.

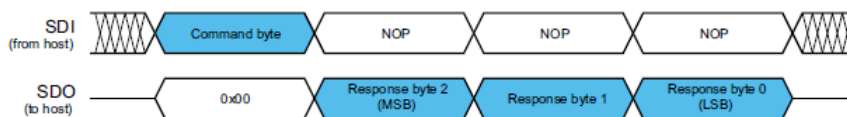
Na obrázku 3.8 je ukázka příkazu s 3 bajtovým argumentem, na kterém lze vidět, jakou strukturu má příkaz posílaný do driveru motoru. Všechny



Obrázek 3.8: Příkaz se 3 bajtovým argumentem[3]

příkazy jsou identifikovány 1 bajtem (*command byte*), který je poslán jako první. Nato mohou podle typu příkazu následovat 1 až 3 bajty, ve kterých je zapsán argument příkazu. Do driveru se posílají vždy 4 bajty. Pokud příkaz nemá argument (např. SoftStop, GetParam) nebo je argument kratší, jsou zbylé bajty posílány jako NOP příkazy. To jsou samé nuly a driver na ně nereaguje. Bajty argumentů i bity všech bajtů jsou posílány od MSB po LSB. Po každém odeslaném bajtu se musí CS signál zvednout aspoň po dobu 800 ns do úrovně *high*, aby stihl driver dekodovat přijatý příkaz a připravit odpověď.

Na obrázku 3.9 je ukázka, jak driver posílá odpovědi. Pokud by např.



Obrázek 3.9: Odpověď s 3 bajtovým argumentem[3]

command byte identifikoval příkaz k vyžádání aktuální rychlosti motoru, byly by hned po přijetí tohoto bajtu z driveru odeslány tři bajty reprezentující hodnotu dotazovaného parametru. Příkaz nemusí vyžadovat odpověď a potom driver posílá jako odpověď nulové bajty (NOP). Pokud přijme driver při vysílání odpovědi nový příkaz, je právě vysílaná odpověď přerušena a driver začíná vysílat odpověď na nově přijatý příkaz.

Z příkazů, které jsou od výrobce k dispozici pro řízení motorů, byly vybrány a do firmware a komunikačního protokolu implementovány tyto:

- Move, Run,
- SoftStop, HardStop,
- SetParam, GetParam,
- GetStatus.

Dále jsou rozvedeny vlastnosti jednotlivých příkazů.

Move

Struktura příkazu Move je na obrázku 3.10. Počet bitů vyhrazených pro hodnotu počtu kroků je 22, maximální hodnota je tedy 4 194 303. Jednotka hodnoty odpovídá nastavenému režimu krokování (plný krok, poloviční krok...). Výchozí režim je mikrokrokování s 1/128 kroku. Směr otáčení je určen posledním bitem prvního bajtu - 0 určuje otáčení ve směru hodinových ručiček, 1 proti směru. Pokud se motor zrovna otáčí, je příchozí příkaz Move ignorován.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	0	0	0	0	DIR	From host
X	X	N_STEP (Byte 2)						From host
N_STEP (Byte 1)								From host
N_STEP (Byte 0)								From host

Obrázek 3.10: Struktura příkazu Move[3]

Run

Struktura příkazu Run je velmi podobná struktuře příkazu Move. Rozdíl je v počtu bitů, které reprezentují hodnotu argumentu a jejich význam. U příkazu Run je to pouze 20 bitů a hodnota argumentu vyjadřuje rychlost, jakou se

má motor otáčet. Maximální hodnota je teoreticky 1 048 575. Prakticky je horní hranice rychlosti otáčení definována parametrem MAX_SPEED. Tento parametr lze nastavit v intervalu od 15.25 do 15610 kroků/s s rozlišením 15.25 kroku/s. Směr otáčení je určován stejně jako u příkazu Move. Hodnota prvního bajtu příkazu je tedy pro otáčení v kladném směru 81 (binárně 0101 0001) a pro otáčení obráceně 80 (binárně 0101 0000).

SoftStop, HardStop

Tyto příkazy nemají žádné argumenty, skládají se tak pouze z 1 bajtu. Hodnota bajtu pro příkaz SoftStop je 176 (binárně 1011 0000) a pro HardStop 184 (binárně 1011 1000). Praktický rozdíl mezi těmito dvěma příkazy je ten, že u SoftStop je rychlost motoru postupně zpomalována, zatímco u HardStop je motor zastaven okamžitě. Rychlost zpomalování lze upravit v registru DEC. U obou příkazů je rotor po zastavení držen ve své poloze pomocí magnetického pole.

SetParam

Obecná struktura tohoto příkazu je na obrázku 3.11. Pět posledních bitů

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	PARAM					From host
VALUE Byte 2 (if needed)								
VALUE Byte 1 (if needed)								
VALUE Byte 0								

Obrázek 3.11: Struktura příkazu SetParam[3]

prvního bajtu (část PARAM) určuje adresu registru, kde je uložen parametr, který chce uživatel změnit. Nová hodnota parametru, kterou chce uživatel nastavit je pak určena dalšími až třemi bajty. Do vyvinutého firmware byla implementována možnost změny tří parametrů uvedených v tabulce 3.1 společně s adresami a délkami registrů, ve kterých jsou parametry uchovávány. Vybrané parametry mají rozsahy 10 a 12 bitů, takže příkazy k jejich nastavení

parametr	adresa registru	délka registru[bit]
Acceleration	5	12
Deceleration	6	12
Maximum speed	7	10

Tabulka 3.1: Vybrané nastavitelné parametry

mají celkem 3 bajty - 1 identifikační a 2 pro hodnotu parametru.

GetParam

U tohoto příkazu funguje adresování parametrů stejně jako u SetParam. Rozdíl je v prvních třech bitech prvního bajtu, které jsou 001. Druhý rozdíl je ten, že příkaz GetParam se skládá z jediného bajtu, který určuje, jaký parametr má driver poslat zpět. Ke čtení byly opět vybrány parametry

Acceleration, *Deceleration* a *Maximum speed*. Driver tedy v odpověď pošle ve všech třech případech 2 bajty s hodnotou parametru.

Struktury příkazů Run, SoftStop, HardStop a GetParam jsou v příloze A.

GetStatus

Příkaz GetStatus žádá o obsah STATUS registru. Příkaz se skládá z jediného bajtu s hodnotou 208 (binárně 1101 0000). Struktura příkazu je v příloze A. Driver v odpověď posílá hodnotu 16 bitového STATUS registru, tedy 2 bajty. Struktura registru je na obrázku 3.12.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SCK_MOD	STEP_LOSS_B	STEP_LOSS_A	OCD	TH_SD	TH_WRN	UVLO	WRONG_CMD
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NOTPERF_CMD	MOT_STATUS		DIR	SW_EVN	SW_F	BUSY	HIZ

Obrázek 3.12: Struktura STATUS registru[3]

Z tohoto registru lze vyčíst informace o stavu driveru a motoru. Např. bity 6 a 5 určují, v jakém stavu se motor nachází (00 - zastavený, 01 - akcelerace, 10 - decelerace, 11 - konstantní rychlost). Bit 4 hlásí směr otáčení motoru (1 - kladný směr, 0 - naopak). Ukazatele informující o ochraně proti ztrátě funkčnosti jsou

- UVLO (under voltage lockout) - zablokování příkazů k pohybu motorem kvůli nízkému napájecímu napětí
- TH_WRN (thermal warning) - varování před přehřátím
- TH_SD (thermal shutdown) - vyřazení z provozu kvůli přehřátí
- OCD (oversurrent detection) - indikuje příliš vysoký proud některým z výkonových MOSFET

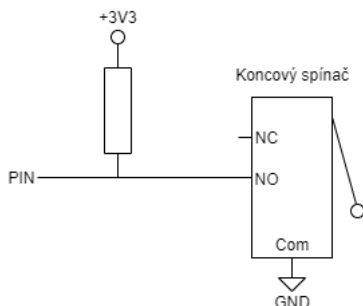
Všechny 4 ukazatele jsou aktivní v 0. Další ukazatel je např. HiZ, který v 1 indikuje, že H-můstky jsou ve stavu vysoké impedance.

3.1.2 Senzory polohy

Koncové spínače

Koncové spínače jsou součástí polohovacího systému jednak pro bezpečnost a potom jako vztažné body. Byly zvoleny, protože jsou jednoduché a dostačují zamýšlenému účelu. Jsou umístěny po jednom na oba konce závitové tyče demonstrátoru, kde jimi jsou detekovány koncové polohy. Na jednom z konců slouží spínač jako referenční nulový bod, podle kterého se určuje pozice posouvané části.

Na demonstrátoru jsou použity páčkové spínače. Ke kitu jsou oba spínače připojeny svými NO (normally opened) vývody, společné vývody COM obou spínačů jsou uzemněny. Schéma propojení s kitem je na obrázku 3.13. Pokud



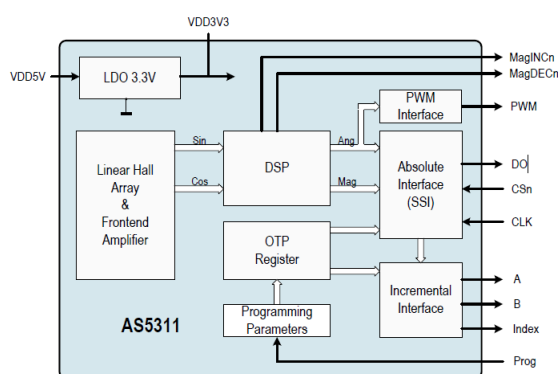
Obrázek 3.13: Připojení koncového spínače k pinu nuclea

se páčka sepne, je NO výstup uzemněn a kit detekuje na příslušném pinu úbytek napětí. Nato kit nejprve vyšle do driveru příkaz HardStop k zastavení motoru a následně pohne s polohovanou částí o několik mikrokroků od spínače, aby ho rozepnul. Na jednom konci demonstrátoru je ještě resetována hodnota čítače výstupu senzoru polohy.

AS5311

Pro sledování polohy je použit inkrementální magnetický lineární senzor polohy AS5311 od firmy ams AG. V provozu funguje senzor v páru s magnetickou páskou, na které se v příčném směru střídají severní a jižní póly o délkách 1mm.

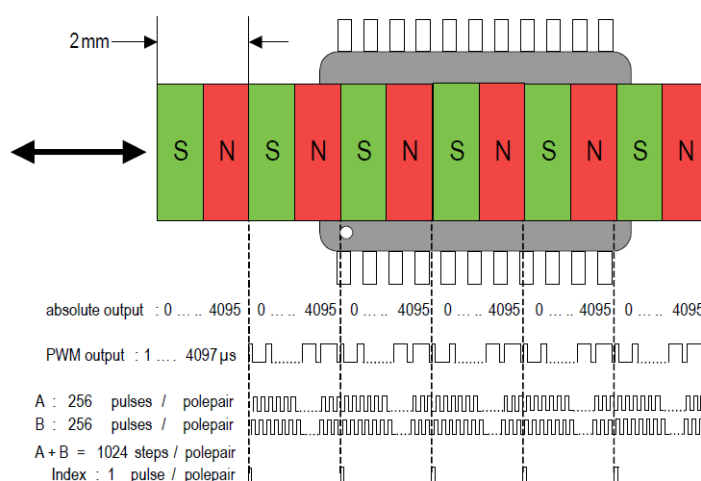
Na obrázku 3.14 je zachycena vnitřní stavba senzoru. V jeho levé části je



Obrázek 3.14: Vnitřní stavba AS5311[4]

lineární Hallovo pole, kterým senzor detekuje změny magnetického pole při projíždění nad páskou. Zaznamenané změny transformuje senzor s pomocí

digitálního signálového procesoru (DPS) do tří různých výstupních signálů. Závislost výstupů na poloze senzoru je na obrázku 3.15.



Obrázek 3.15: Výstupy AS5311 v závislosti na poloze senzoru[4]

Absolutní výstup vysílá informaci o poloze pomocí synchronního sériového rozhraní (SSI). Poloha je relativní v rámci 1 pólového páru a je zakódována do 12 bitového slova. Pro každý pár tedy nabývá hodnot 0 až 4095, takže lze polohu určit s přesností na 488nm. Podobně funguje i PWM výstup, který ale v rámci každého páru pólů generuje PWM signál, jehož střída závisí na pozici nad párem. Na začátku je šířka pulzu 1μs a roste s krokem 0.488μm až na šířku 4097μs na konci páru.

Pro sledování polohy využívá kit třetí inkrementální výstup - kvadrurní A,B výstup se signálem index. Každý z výstupů A,B generuje v rámci pólového páru 256 pulzů, které jsou vůči sobě posunuté o 90 elektrických stupňů. To vytváří 1024 hran na 1 pár pólů, přesnost tohoto výstupu je tedy 1.95μm. Signál index generuje senzor jako 1 pulz v každém pólovém páru.

Senzor umí také např. rozpoznat a dát pomocí pinů MagINCn a MagDECn vědět, zda je ve správné vzdálenosti k magnetické pásce a může správně pracovat. Pokud jsou oba piny v 0, je síla magnetického pole, v němž se senzor nachází, v doporučených mezích (10-40mT). Pokud je MagINCn v 1, je síla pole v rozmezí 3.4-54.5mT. V těchto podmínkách senzor lehce ztrácí na přesnosti. Pokud jsou oba piny v 1, je síla pole mimo rozmezí 3.4-54.5mT a senzor může být stále používán, ale výrobce to nedoporučuje. V tabulce 3.2 jsou uvedeny další zohledňované i zajímavé parametry senzoru AS5311.

Z tabulky je vidět, že senzor je malý, má plně postačující rozlišení i měřicí rozsah a rychlostní limit použití. Protože má možnost napájení na 3.3 V, lze ho bez větších úprav přímo připojit k vývojovému kitu. Senzor má také příznivou cenu, která se pohybuje okolo 13\$. Díky všem výše popsaným kladným vlastnostem byl vybrán právě tento senzor. Kvadrurní výstup zvolený pro sledování polohy však používá i řada dalších senzorů polohy,

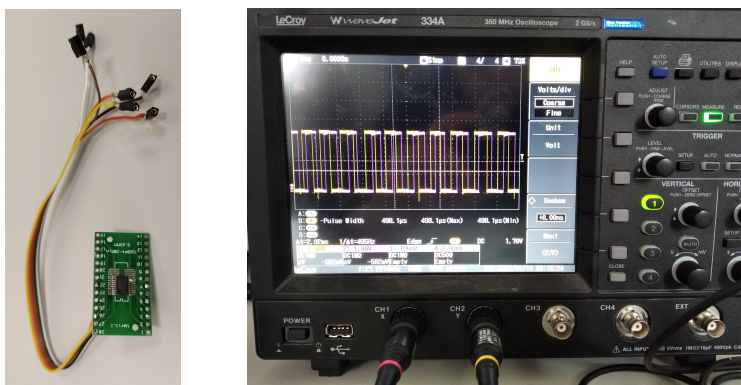
rozměry [mm]	6.5x6.4x1.2
rozdílení [μm]	1.95, 0.488
výstup	ABI, PWM
přesnost [μm]	+/-10
maximální rychlost [m/s]	0.65
vstupní napětí [V]	3.3 nebo 5
napájecí proud [mA]	16
stupnice	mg páska s délkou pólu 1mm
tolerance	halovo pole musí být nad aktivní částí pásky
maximální měřicí délka [m]	50
vzdálenost od pásky [mm]	0.5 (typicky 0.3-0.6)
operační prostředí	atmosférické, teplota -40°C až +125°C

Tabulka 3.2: Zohledňované parametry senzoru AS5311

takže ho lze v případě potřeby vyměnit za jiný.

■ Sestavení senzorů

Senzor AS5311 byl připájen na destičku společně s dráty pro vedení napájení, země a dalších signálů do nuclea. Poté byla testována jeho funkčnost a zkouška proběhla dobře.



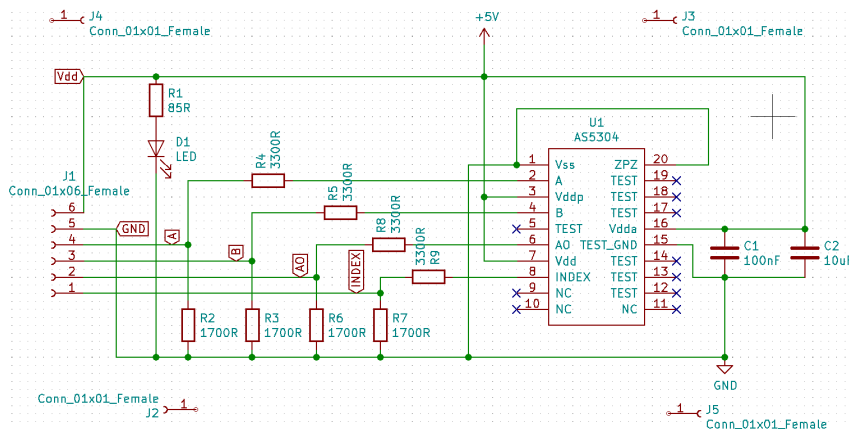
(a) : Zapájený enkodér

(b) : Zkouška funkčnosti

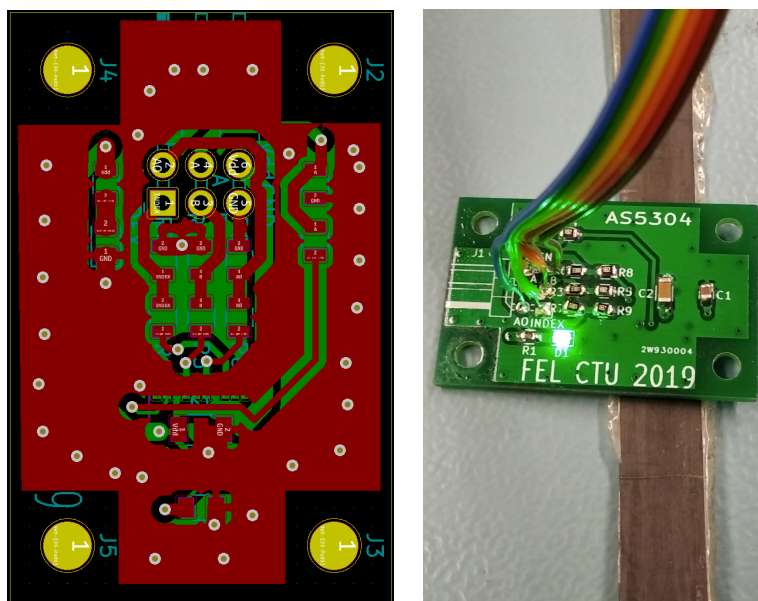
Obrázek 3.16: Zprovoznění senzoru AS5311

Překvapivě se však ukázalo, že magnetická páska s délkou pólu 1 mm je na trhu těžko dostupná. Nakonec se jí podařilo získat, ale souběžně se pro jistotu pracovalo na záložní variantě. Tou byl senzor AS5304 od stejné firmy, který funguje velmi podobně jako AS5311. Liší se v napájení (na 5V) a v přesnosti (na 25 μm). Nicméně i ten se podařilo úspěšně otestovat. Oproti AS5311 bylo potřeba navíc vyřešit napájení, protože piny kitu jsou 3.3V kompatibilní, a tak byla navržena destička, na kterou se pro používané signály přidaly

napěťové děliče. Také se na ni přidaly blokovací kondenzátory a LED pro indikaci napájení. Na obrázcích 3.17 a 3.18 jsou schéma obvodu na destičce, návrh DPS a zapájený senzor.



Obrázek 3.17: Schéma pro desku plošných spojů



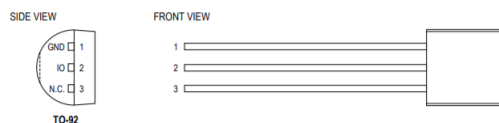
(a) : Návrh desky plošných spojů (b) : Osázená deska plošných spojů

Obrázek 3.18: Deska plošných spojů k senzoru AS5304

3.1.3 Paměti motorů

K uchování parametrů motorů slouží paměť DS2431. Jedná se o 1024 bitovou EEPROM rozdělenou do 4 stránek o 256 bitech. Součástí tvoří pouzdro s rozhraním pro komunikaci po OneWire sběrnici, CRC-16 generátorem, řídicí jednotkou, ROM s unikátním 64 bitovým výrobním číslem, zápisníkem (*Scratchpad*), registrační stránkou (*Register page*) a vlastní EEPROM. Z

pouzdra jsou vyvedeny tři piny (obrázek 3.19). Ke komunikaci s kitem po



Obrázek 3.19: Rozložení pinů paměti[5]

1-Wire sběrnici slouží prostřední IO pin. Zbylé dva piny jsou GND a N.C. (not connected). Při použití 1-Wire protokolu komunikuje EEPROM rychlostí 125 kbps. Rozsah napětí pro zápis i čtení je 2.8 V – 5.25 V, pro teploty od -40°C do +85°C.

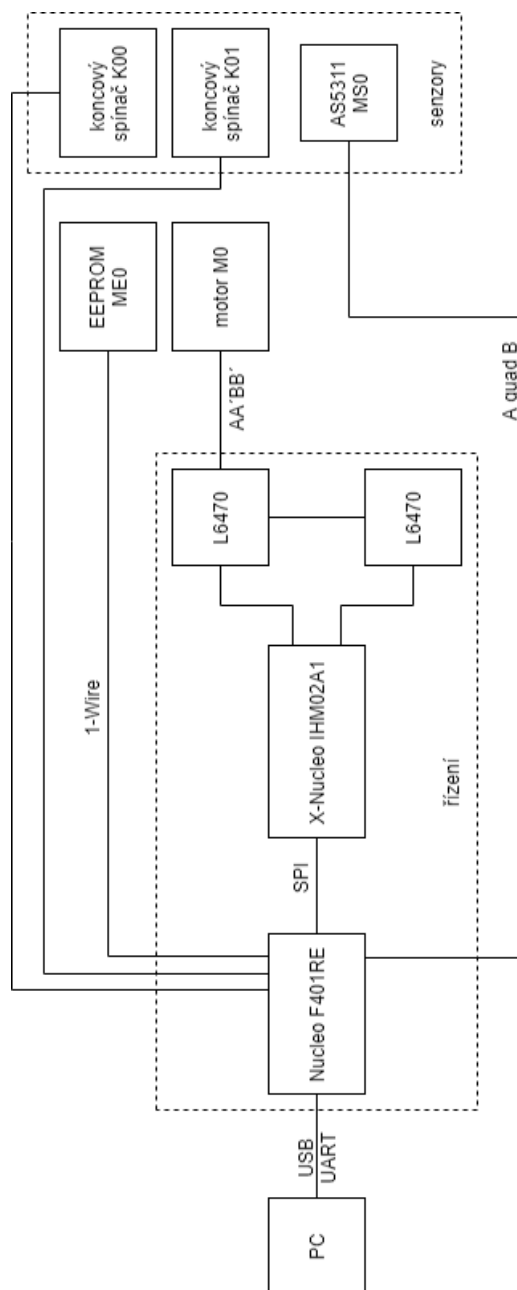
V tabulce 3.3 jsou uvedeny parametry, které jsou ukládány do paměti motoru. Postup pro zápis a vyčtení dat EEPROM je v popsán v sekci 4.3. Paměti EEPROM se využívají např. k uložení kalibračních dat senzorů nebo identifikaci PCB či inkoustových a tonerových kazet pro tiskárny [5].

parametr	velikost[bajt]
absolutní poloha	4
jméno motoru	10
maximální rychlost	2
maximální zrychlení	2
počet plných kroků na otáčku	4

Tabulka 3.3: Ukládané parametry motoru

3.2 Schéma nového systému

Na obrázku 3.20 je znázorněno schéma nově navrženého systému. Schéma je pro demonstrátor, obsahuje tedy pouze jeden připojený motor, jednu EEPROM a jeden blok senzorů. Pro použití v mikrotomografu by se dal systém rozšířit až o tři další expanzivní desky a řídit jednou nucleo deskou 8 motorů. Podle toho by byl přidán i příslušný počet pamětí a senzorů.



Obrázek 3.20: Blokové schéma modelu nově navrženého řídicího systému

Kapitola 4

Firmware

Firmware systému je s ohledem na použití v mikrokontroléru napsán v jazyce C a při jeho tvorbě bylo využito vývojové prostředí STM32CubeIDE. Firmware má na starost přijímání a vykonávání příkazů z PC, zpracovává kvadraturní signál ze senzoru polohy a obsluhuje koncové spínače a EEPROM. Nejprve je popsán způsob komunikace mezi PC a mikrokontrolérem, dále sledování polohy a obsluha paměti motoru.

4.1 Komunikační protokol

Předávání zpráv mezi PC a mikrokontrolérem je definováno komunikačním protokolem, který vychází z ASCII verze Modbus protokolu. Liší se od něj o přidanou část *délka dat* a rozlišením části *funkce* na dvě části: *typ funkce* a *kód funkce*. Níže je znázorněna struktura zpráv.

3A	adresa	typ funkce	kód funkce	délka dat	data	LRC	0D	0A
----	--------	------------	------------	-----------	------	-----	----	----

Obrázek 4.1: Struktura zpráv posílaných mezi PC a Nucleo F401RE

Hodnoty *3A*, *0D* a *0A* indikují začátek a konec zprávy. Jedná se po řadě o ASCII hodnoty symbolů `' :`, `'\r'` a `'\n'`. Každý z těchto symbolů ve zprávě zabírá 1 bajt. Stejně místo zabírá také část *typ funkce* tvořená jedním ASCII znakem. Část *adresa* zabírá 2 bajty a skládá se ze znaku M a číslice 0–9. Ostatní části zprávy mezi začátkem a koncem jsou reprezentovány jednou nebo několika dvojicemi ASCII znaků a jedna dvojice ASCII znaků kóduje 1 bajt. Části *kód funkce*, *délka dat* a *LRC* jsou odesílány jako jedna dvojice ASCII znaků. Část *data* může být v závislosti na typu zprávy velká až 128 bajtů, tedy 256 ASCII znaků.

Ve většině příkazů jsou posílány *data* dlouhá maximálně okolo 10 bajtů. Výjimkou je odpověď na příkaz k vyčtení dat z EEPROM, kdy se podle požadavku vyčte a v celku odešle maximálně 128 bajtů, což odpovídá kapacitě EEPROM.

Pro snazší manipulaci a zpracování jsou navíc části zprávy *kód funkce*, *délka dat*, *data* a *LRC* reprezentovány dvojicemi znaků ze sady 0-9 a A-F nebo a-f. Díky zápisu v šestnáctkové soustavě lze relativně snadno získat původní bajt tak, že se znaky převedou na čísla, která mají binárně rozsah 4 bitů, a tyto 2 čtveřice se následně složí do jednoho bajtu. Původní bajt lze takto i snadno zakódovat.

■ 4.1.1 Příjem zpráv

Mikrokontrolér přijímá v přerušení jednotlivé znaky přes UART z PC a ukládá je do zásobníku. Při přijetí terminálního znaku '\n' je nastavena signální proměnná značící příjem zprávy, která se kontroluje v hlavní smyčce programu. Ta v pseudokódu vypadá takto:

```
while (1):
    actualize position
    if received message:
        process content of buffer
    if limit switch 1 closed:
        stop motor
        move from switch 1 to open it
    if limit switch 2 closed:
        stop motor
        move from switch 2 to open it
```

V hlavní smyčce je tedy aktualizována poloha a kontrolováno pomocí signálních proměnných, zda byla přijata nová zpráva nebo sepnut některý z koncových spínačů. Přijímané zprávy jsou střídavě ukládány do dvou zásobníků, aby mohl být obsah jednoho po přijetí zprávy zpracován a druhý zásobník k dispozici pro přijímání další zprávy. Po přijetí je zpráva nejprve zkontrolována. Jako kontrolní kód byl použit LRC - Longitudinal redundancy check, jehož výpočet lze v pseudokódu zapsat takto:

```
lrc := 0
for each byte in buffer do:
    lrc := (lrc + byte) AND 0xFF
lrc := (((lrc XOR 0xFF) + 1) AND 0xFF)
```

Pokud se vypočtený LRC od přijatého liší, kit příkaz zahodí a do PC je posláno hlášení, že příkaz nebyl přijat v pořádku. Jinak jsou ze zásobníku vytaženy ostatní části zprávy, mikrokontrolér je dekóduje a vykoná požadovaný příkaz.

■ 4.1.2 Části zprávy

Níže jsou rozvedeny jednotlivé části zprávy a jejich účel.

- Adresa - určuje, pro který motor, příp. přidruženou EEPROM, je příkaz určen. Pro aplikaci na demonstrátor nabývá pouze podoby M0.

- Typ funkce - určuje, zda se data zapisují (znak *W*), anebo čtou (znak *R*).
- Kód funkce - identifikuje příkaz pro motor driver anebo příkaz týkající se EEPROM příp. polohy. V případě příkazu pro driver motoru je jako kód funkce použit identifikační bajt příkazu. Pro příkazy týkající se EEPROM a polohy byly určeny kódy zvlášť popsané v dalších dvou sekcích.
- Délka dat - určuje délku samotných dat zprávy, může být i 0, potom pole dat zcela chybí (např. u příkazu *GetParam*).
- LRC - kontrolní kód.

Struktura pole *data* se liší v závislosti na typu příkazu.

- *Move, Run* - data mají délku 3 bajty, to pokrývá délky registrů pro parametry příkazů (počet kroků 22 bitů, rychlost 20 bitů). Společně s přijatým kódem funkce v čele je toto pole přes SPI sběrnici s využitím knihovny HAL posláno do driveru motoru pro vykonání pohybu.
- *SoftStop, HardStop, GetAbsPos, SetAbsPos, WriteAbsPosToEeprom* - pole je prázdné. Kódy funkce příkazů *SoftStop* a *HarfStop* jsou poslány přes SPI do driveru motoru. Význam zbylých tři zmiňovaných příkazů je vysvětlen v sekci 4.2.
- *SetParam* - data mají délku 2 bajty, což pokrývá číselný rozsah vybraných parametrů (maximální rychlost 10 bitů, zrychlení/zpomalení 12 bitů). Pole je v čele s kódem funkce posláno do driveru motoru jako v případě příkazů *Move* a *Run*.
- *GetParam, GetStatus* - při příjmu kitem je pole prázdné, kit pošle kód funkce do motor driveru přes SPI a přijme v odpověď požadovaný parametr. Naplní jím pole data a následně ho pošle s upravenou délkou dat a přepočteným kontrolním kódem zpět do PC.
- *WriteEeprom, ReadEeprom* - první bajt určuje, od které části EEPROM se má zapisovat/číst, druhý bajt udává, kolik bajtů má být zapsáno/vyčteno, a v případě *WriteEeprom* následují samotná data k zápisu.

Pro přehlednost je dále uvedena tabulka 4.1 uvádějící příkazy, které firmware přijímá a vykonává, společně s jejich dalšími vlastnostmi.

4.2 Sledování polohy

Informace o poloze je primárně ukládána do kitu. Kvadraturní výstup senzoru je napojen na piny PA0 a PA1, na kterých je nastaven čítač v Encoder módu s maximální nastavitelnou periodou $2^{32} - 1$. Jelikož senzor dává 1024 pulzů na 1 pár pólů, tedy na 2mm, vydělil jsem periodu čítače

jméno příkazu	typ funkce	kód(y) funkce(i) (hex)	délka dat[bajt]	data
Move	W	40,41	3	počet kroků
Run	W	50,51	3	rychlost
SoftStop	W	B0	0	-
HardStop	W	B8	0	-
SetParam	W	05, 06, 07	2	maximální rychlost, zrychlení, zpomalení
GetParam	R	25, 26, 27	0	-
GetStatus	R	D0	0	-
GetAbsPos	R	FB	0	-
WriteAbsPosToEeprom	W	FC	0	-
SetAbsPos	W	FD	0	-
WriteEeprom	W	FE	4-12	parametr k zapsání
ReadEeprom	R	FF	2-128	-

Tabulka 4.1: Přehled příkazů zpracovávaných firmwarem

číslem 512 a zjistil, že můžu měřit na rozsahu přibližně 8,387 km, resp. -4,1935 km až 4,1935 km, což je vzhledem k rozměrům přístroje dostačující. Čítač plně dekóduje signály z A a B kanálů senzoru a podle toho, který z nich je předsunut před druhý, rozlišuje i směr posunu a buď pulzy přičítá anebo odčítá. Hodnota čítače je ukládána do proměnné, která je aktualizována v hlavní smyčce programu.

Počet pulzů je při zapnutí nuclea resetován na 0. Uživatel má k dispozici tři funkce, kterými může zjišťovat nebo nastavovat aktuální počet pulzů. Zasláním příkazu GetAbsPos (kód funkce FB) je uživateli nazpět poslán aktuální počet pulzů uložený v kitu. Příkazem WriteAbsPosToEeprom (kód funkce FC) uživatel zapíše aktuální počet pulzů do EEPROM a kit mu nazpět pošle, jaký počet pulzů do paměti zapsal. Třetí funkce je SetAbsPos (kód funkce FD), kterou uživatel vydá příkaz k přepsání polohy uložené v kitu hodnotou uloženou v EEPROM. S jejím vyžitím kit může po zapnutí pokračovat i v absolutním polohování od jakékoliv pozice, pokud nebylo s motorem během vypnutí kitu pohnuto. Uživatel může využít ještě funkci SetZeroPos, která ale pouze vydá příkaz Run směrem k nulové poloze, kde se motor po sepnutí koncového spínače zastaví, o kousek vrátí a kit nastaví polohu jako nulovou, resp. vynuluje čítač pulzů ze senzoru.

4.3 Obsluha EEPROM

Pro obsluhu EEPROM mi byly od kolegy poskytnuty knihovny pro komunikaci mezi kitem a pamětí DS2431 po OneWire sběrnici, které byly převzaty z [8] a lehce upraveny. Komunikace po sběrnici je inicializována po zapnutí kitu a do PC je poslána informace o počtu nalezených pamětí na sběrnici.

4.3.1 Zápis

Velikost dat, která lze do EEPROM najednou zapsat, je omezena na 8 bajtů. Data se nezapisují do paměti přímo, ale přes tzv. *Scratchpad*, který má kapacitu právě 8 bajtů. Nejprve je přes OneWire sběrnici poslán příkaz k zápisu do *Scratchpad* a adresa, od které se mají data zapisovat. Poté jsou data zkopírována do *Scratchpad*. Nato CRC-16 generátor v DS2431 vypočte CRC kód z příkazu, adresy i dat. CRC je vyčteno ze sběrnice a je možné ho porovnat s vlastním vypočteným CRC a ověřit, že komunikace funguje správně. Poté je poslán příkaz ke čtení *Scratchpad*, za kterým se musí vyčíst 3 bajty autorizačního kódu pro zápis do paměti. Nakonec je přes sběrnici zaslán příkaz ke zkopírování obsahu *Scratchpad* do paměti následovaný 3 přijatými autorizačními bajty. Poté se počká několik milisekund na provedení zápisu do paměti.

Kód funkce pro zápis do paměti je FE. Pokud je potřeba zapsat do paměti více dat, než je kapacita *Scratchpad*, rozdělí firmware data po přijetí v kitu na dílčí části po 8 bajtech. Ty jsou postupně ukládány na odpovídající po sobě jdoucí adresy do EEPROM.

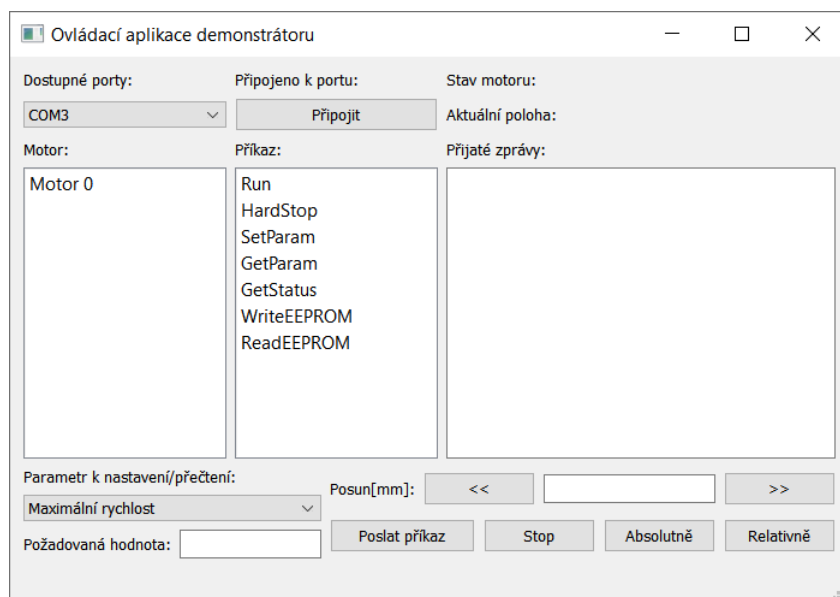
4.3.2 Čtení

V případě čtení lze požadovaná data vyčíst přímo z paměti. Jako první se přes OneWire sběrnici pošle příkaz ke čtení z paměti a za ním adresa, od které se má číst. Nato jsou data ve for cyklu čtena po 1 bajtu až do požadovaného počtu. Dále jsou Modbus protokolem posílána do PC. Maximální velikost vyčítaných dat je omezena velikostí pole *délka dat* ve struktuře příkazu na 255 bajtů. To plně postačuje, protože tím, že je paměť 1024 bitová, lze z ní vyčíst pouze 128 bajtů. Kód funkce pro čtení z paměti je FF.

Kapitola 5

Ovládací software

Aplikace pro ovládání demonstrátoru byla napsána s pomocí aplikačního rámce Qt. Na obrázku 5.1 je ukázáno uživatelské rozhraní aplikace. Dále jsou



Obrázek 5.1: Uživatelské rozhraní pro ovládání demonstrátoru

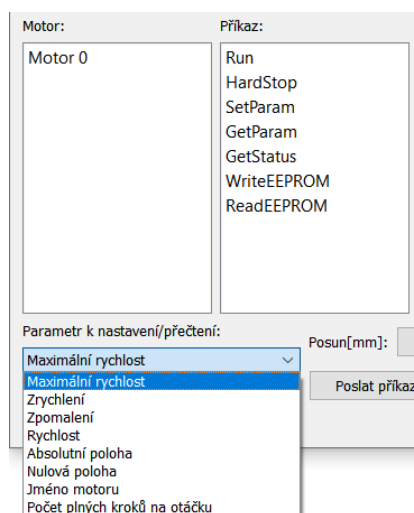
rozebrány jednotlivé funkce rozhraní.

5.1 Připojení ke kitu

Možnost připojení ke kitu je v horní části rozhraní. Aplikace po zapnutí přednastaví výchozí parametry přenosu sériovým portem. Ty jsou: BaudRate 115200, počet datových bitů 8, parita žádná, počet stop bitů 1 a Flow Control NoFlowControl. Poslední parametr udává, zda může jedno z komunikujících zařízení dát druhému signál, k zastavení vysílání (např. z důvodu přehlcení příjemce přijímanými daty). Po kliknutí na tlačítko 'Připojit' otevře aplikace

5.3 Další příkazy

Kromě zadání posunu může uživatel zvolit příkazy z levé části rozhraní, vlevo dole vybrat parametr a pokud to příkaz vyžaduje, specifikovat jeho hodnotu. Na obrázku 5.3 je zobrazena nabídka příkazů a parametrů, které má uživatel k dispozici. Dále jsou rozvedeny jednotlivé příkazy.



Obrázek 5.3: Příkazy a parametry k dispozici

Run

Použití tohoto příkazu vyžaduje zadání rychlosti do pole 'Požadovaná hodnota:' a určit pomocí tlačítek « a » směr točení. Pokud uživatel zadá příliš velkou rychlost, tak je přenastavena na maximální hodnotu 1023. Ta vychází z toho, že driver bude otáčet motorem tak jako tak maximálně rychlostí nastavenou v registru MAX_SPEED. Registr pro ni má velikost 10 bitů, proto byl limit rychlosti Run zvolen 1023.

Hodnota, kterou uživatel udá, je v krocích na tik, kde 1 tik je 250 ns. Vyjádřit rychlost v krocích za sekundu lze podle [3] následujícím vzorcem

$$v_{krok/s} = \frac{v_{krok/tik} \cdot 2^{-18}}{250 \text{ ns}}$$

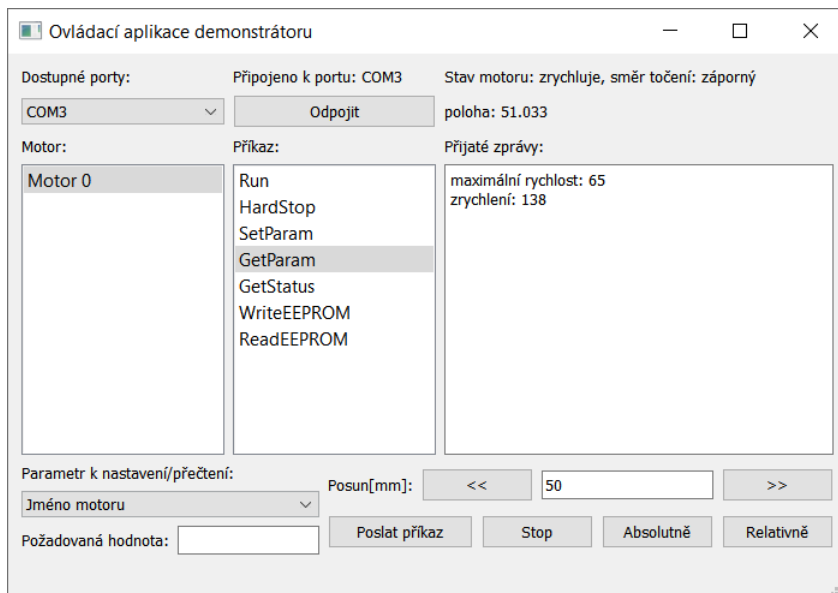
Pokud je tedy výchozí hodnota maximální rychlosti 65 krok/tik, odpovídá to podle výše uvedeného vzorce rychlosti 991,8 krok/s. Se stoupáním závitové tyče 2.5 mm a 200 kroky motoru na jednu otáčku by uživatel pohnul s posuvnou částí o 10 cm (8000 kroků) přibližně za 8.1 s.

HardStop

Jedná se o jiný způsob zastavení motoru, ne se zpomalením jako u SoftStop, ale okamžitě.

SetParam

status registru driveru je dekódován pohybový stav motoru a vypsáno jak se pohybuje (vůbec, konstantní rychlostí, zrychluje, zpomaluje), příp. jakým směrem. Na obrázku 5.4 je ukázka výpisu přijatých zpráv.

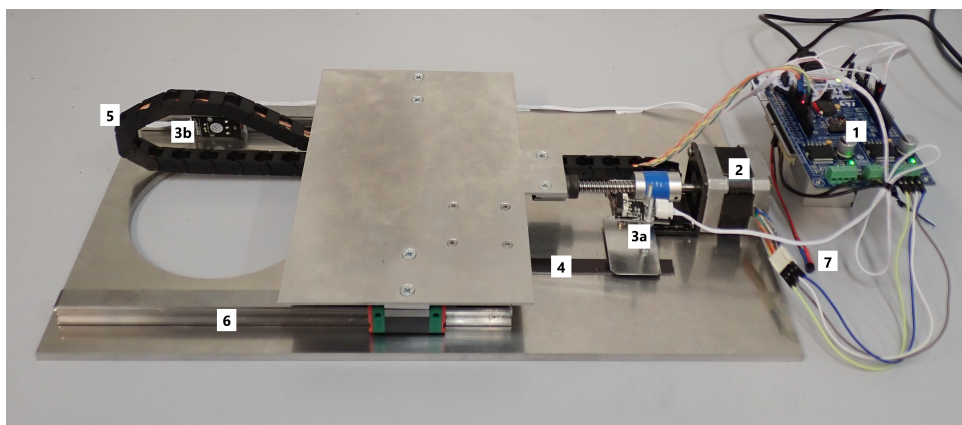


Obrázek 5.4: Ukázka výpisu zpráv přijatých z kitu

Kapitola 6

Testování

Testování systému probíhalo průběžně v rámci práce. Postupně byla s řídicím kitem vyzkoušena zvlášť funkce senzoru polohy, krokového motoru, EEPROM a koncových spínačů. Pro ověření funkčnosti implementovaného firmware a ovládací aplikace byly pak všechny hardware komponenty namontovány na jednoduchý polohovací model. Snímek modelu je na obrázku 6.1.

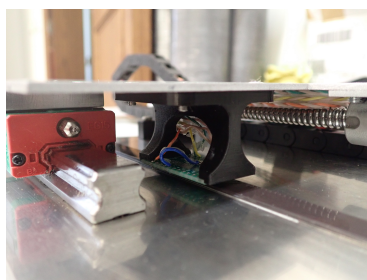


Obrázek 6.1: Model nového polohovacího systému. 1-IHM02A1 připojená na STM32F401RE, 2-krokový motor, 3a,3b-koncové spínače, 4-magnetická páska, 5-vedení signálů z AS5311 kryté řetězem, 6-lineární vedení, 7-EEPROM

Na obrázku 6.2 je vidět uchycení senzoru AS5311 z boku. K uchycení byl vytisknut díl na 3D tiskárně. Dráha modelu měří na délku přibližně 11.5 cm. Použitý krokový motor je Microcon SX17-0804 s 200 krok/ot, závitová tyč má stoupání 2.5 mm.

Zkouška elektronické a programové části systému proběhla dobře. Ukázalo se při ní, že PC i mikrokontrolér od sebe umí navzájem přijímat, kontrolovat a správně dekodovat zprávy a mikrokontrolér zvládá obsluhovat senzor polohy, koncové spínače i paměť EEPROM. Poté se otestovaly ještě některé mechanické vlastnosti systému.

Nejdříve byly vypočteny teoretické možnosti polohování. Maximální rychlost



Obrázek 6.2: Uchytení senzoru AS5311 na posuvnou desku modelu

otáčení udaná výrobcem je 15610 krok/s. S parametry motoru a tyče to dává v přepočtu rychlost posunu přibližně $v_{max} = \frac{15610 \cdot 2.5}{200} \doteq 195$ mm/s. Se započtením maximálního zrychlení a zpomalení, které jsou 59590 krok/s² (v případě modelu odpovídá 744.875 mm/s²), by byla schopna posuvná část modelu urazit dráhu 11.5 cm přibližně za 0.6 s.

Při testování se však ukázalo, že posuvná část je pro motor s nastavenou maximální akcelerací příliš velká zátěž a motor se takto rychle na maximální rychlost ani nedostane a zasekne se. Motor navíc při testu vysoko pískal. Ukázalo se, že vhodná rychlost je v přepočtu 3.8 mm/s, při které je motor velmi tichý. Jedná se sice o nízkou rychlost, ale v praxi není většinou vyžadováno, aby části přístroje urážely velké vzdálenosti v krátkém čase.

Přesnost polohování modelu omezuje dle výpočtů senzor AS5311. Nejmenší posun, který senzor zaznamená, je 1.95 μm. Nejmenší úhel, o který dokáže v režimu 128 mikrokrok/krok řídicí elektronika otočit hřídel motoru, je

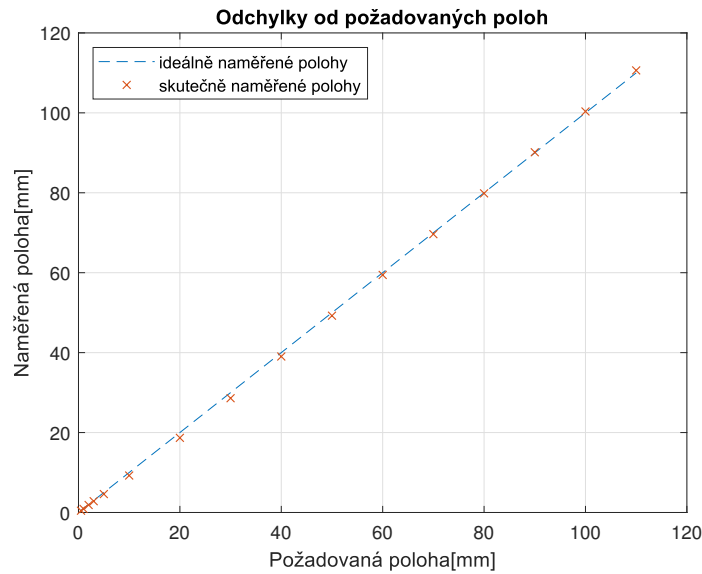
$$\alpha_{min} = \frac{360^\circ}{200 \cdot 128} = 50.625''.$$

Tento úhel odpovídá s použitou závitovou tyčí posunu přibližně

$$posun_{min} = \frac{2.5}{200 \cdot 128} \doteq 100 \text{ nm}.$$

Takovýto posun nebylo možné nijak ověřit. Byla ale testována přesnost polohování na dráze demonstrátoru v rozsahu 0 až 110 mm. Před měřením bylo lehce upraveno stoupání závitové tyče na hodnotu 2.45 mm, aby na krátkých úsecích požadovaná poloha lépe seděla s naměřenou. Vždy se začínalo od nulové polohy a postupně byla posuvná část posílána do vzdáleností 0.5 mm, 1 mm, 2 mm, 3 mm, 5 mm, 10 mm a pak s pravidelným zvětšováním vzdáleností o 10 mm až do vzdálenosti 110 mm. Naměřená data jsou zpracována do grafu 6.3. Na obrázku je bohužel vidět, že naměřené polohy jsou znatelně vychýlené od požadovaných poloh. Největší odchylky byly při požadavku na polohy 20 mm, odchylka 1.32 mm, a 30 mm, odchylka 1.409 mm. V ostatních případech byla odchylka menší než 1 mm.

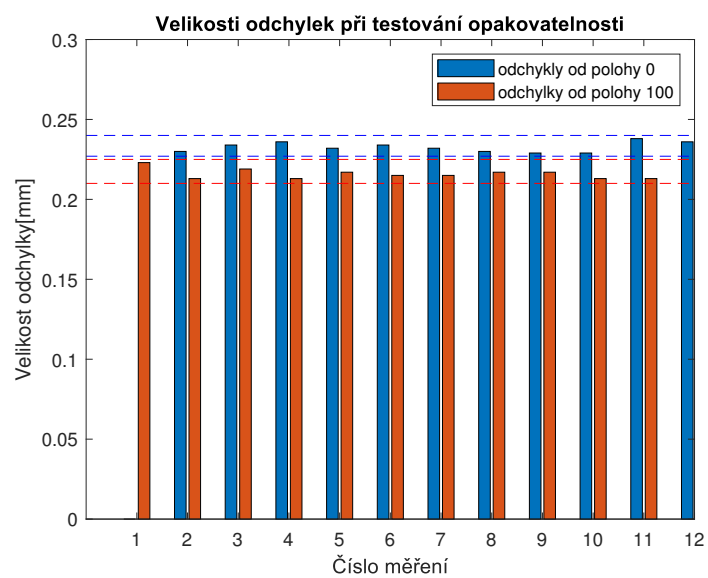
Dále byla zkoušena opakovatelnost polohování na dráze 0 až 110 mm při rychlosti 3.8 mm/s. V grafu 6.4 jsou vyneseny velikosti odchylek od koncových poloh dráhy. Opět se ukazuje nepřesnost polohování, velikosti samotných



Obrázek 6.3: Výsledky testování přesnosti polohovacího modelu

odchylek však nejsou až tak rozdílné (naznačeno modrými a červenými přerušovanými čarami v grafu). Největší rozdíl velikostí odchylek okolo polohy 0 je $9\ \mu\text{m}$, okolo polohy 100 je to $10\ \mu\text{m}$.

Nepřesnosti v polohování byly zřejmě dány z větší části mechanickými nedokonalostmi modelu. Např. závitovou tyč se s hřídelí motoru nepodařilo nastavit přesně do jedné osy, takže při točení viditelně vyhazovala. Pro kompenzaci odchylek by bylo možné v ovládacím software implementovat jednoduchý regulátor, který by odchylky od požadovaných poloh udržoval v rozumném intervalu.



Obrázek 6.4: Velikosti odchylek od koncových poloh při měření opakovatelnosti modelu polohování

Kapitola 7

Závěr

Cílem této bakalářské práce bylo navrhnout řídicí hardware a software pro rozvoj stávajícího polohovacího systému pro mikrotomograf v ÚTEF ČVUT v Praze. Vyvinutý řídicí systém byl s jedním motorem vyzkoušen na polohovacím modelu a úspěšně ověřena jeho funkčnost. Vlastnosti stávajícího systému jsou vylepšeny použitím novějšího driveru pro krokové motory L6470, který umožňuje zvýšit přesnost a opakovatelnost polohování. Zpětná vazba zprostředkovaná senzorem AS5311 a koncovými spínači nabízí zvýšení bezpečnosti i uživatelského komfortu pro nastavování měřicí aparatury. Dále systém obsahuje EEPROM pro uložení parametrů ovládaného motoru, čímž se zjednodušuje používání systému.

Komunikace řídicího systému s PC probíhá pomocí speciálně navrženého protokolu inspirovaného Modbus protokolem. Součástí zpráv vyměňovaných mezi systémem a PC je kontrolní kód pro detekci chyb, čímž se zvyšuje jejich spolehlivost. Na základě v pořádku přijatých příkazů ovládá firmware mikrokontroléru ostatní hardware komponenty systému, tedy krokový motor, senzor polohy, koncové spínače a EEPROM, nebo z nich zprostředkovává uživateli informace nazpět do PC.

Uživatel může pomocí ovládacího software nastavovat úhel natočení použitého motoru s přesností v řádu minut a vyžádat si zpětnou vazbu o lineární poloze na 2 μm přesně. Díky zpětné vazbě lze nastavovat polohu i absolutně, přičemž referenční nulový bod je nastavován sepnutím jednoho z koncových spínačů připojených k mikrokontroléru. Software umožňuje ukládání absolutní polohy i do EEPROM, takže je možné ihned po zapnutí určit aktuální pozici. Přes uživatelské rozhraní lze dále nastavit maximální rychlost, zrychlení a zpomalení motoru a vyčíst jejich aktuální hodnoty, stejně tak jako nastavovat a vyčítat parametry uložené v EEPROM. Příkazem GetStatus si uživatel může vyžádat informaci, v jakém stavu se motor nachází, případně jakým směrem se otáčí. Řídicí systém svými celkově 12 typy příkazů nabízí kompaktní sadu pro řízení krokových motorů driveru L6470.

Vhodným dalším krokem k rozvoji systému by bylo vylepšení uživatelského rozhraní k ovládní motoru na PC, které bylo vytvořeno pouze v základní verzi pro potřeby polohovacího modelu. Rozhraní by mohlo být dále přizpůsobeno

k ovládání většího počtu motorů. Také by mohla být vytvořena programová knihovna funkcí pro ovládání krokových motorů z PC pomocí hardware použitého v této bakalářské práci.



Literatura

- [1] DUDÁK, Jan. Energy Sensitive X-Ray Radiography and Tomography Optimized for Small Animal Imaging. Kladno, 2019. Disertace. ČVUT v Praze, Fakulta biomedicínského inženýrství.
- [2] Newport: Motorized Linear Stage M-IMS500LM-S. 2020. URL <https://www.newport.com/p/M-IMS500LM-S>
- [3] STMicroelectronics: L6470 datasheet. 2015. URL <https://www.st.com/resource/en/datasheet/l6470.pdf>
- [4] ams AG: AS5311 High Resolution Magnetic Linear Encoder datasheet. 2013. URL https://ams.com/documents/20143/36005/AS5311_DS000200_2-00.pdf/575d252f-096b-e5ab-5d8f-d504fd005062
- [5] maxim integrated: DS2431 datasheet. 2015. URL <https://www.maximintegrated.com/en/ds/DS2431.pdf>
- [6] HLINOVSKEÝ, Vít. Krokový motor VII - PowerPoint prezentace. 2015. URL https://moodle.fel.cvut.cz/pluginfile.php/237199/mod_resource/content/1/7%20_%20Krokov%C3%BD%20motor.pdf
- [7] STMicroelectronics: IHM02A1 user manual. 2015. URL https://www.st.com/resource/en/user_manual/dm00237629-getting-started-with-xnucleoihm02a1-twoaxis-stepper-motor-driver-expansion-board-based-on-l6470-for-stm32-nucleo-stmicroelectronics.pdf
- [8] MAJERLE, Tilen. Library 12- OneWire library for STM32F4. 2014. URL <https://stm32f4-discovery.net/2014/05/library-12-onewire-library-for-stm43f4xx/>
- [9] PARTS: 1K 1-WIRE EEPROM (DS2431) [online]. 2008. URL <https://hackaday.com/2008/12/24/parts-1k-1-wire-eprom-ds2431/>
- [10] Qt 5.14 Documentation: All Qt C++ Classes. 2020. URL <https://doc.qt.io/qt-5/classes.html>

Příloha A

Příkazy a registry k ovládání motoru

Command mnemonic	Command binary code					Action
	[7 5]	[4]	[3]	[2 1]	[0]	
NOP	000	0	0	00	0	Nothing
SetParam(PARAM,VALUE)	000	[PARAM]				Writes VALUE in PARAM register
GetParam(PARAM)	001	[PARAM]				Returns the stored value in PARAM register
Run(DIR,SPD)	010	1	0	00	DIR	Sets the target speed and the motor direction
StepClock(DIR)	010	1	1	00	DIR	Puts the device into Step-clock mode and imposes DIR direction
Move(DIR,N_STEP)	010	0	0	00	DIR	Makes N_STEP (micro)steps in DIR direction (Not performable when motor is running)
GoTo(ABS_POS)	011	0	0	00	0	Brings motor into ABS_POS position (minimum path)
GoTo_DIR(DIR,ABS_POS)	011	0	1	00	DIR	Brings motor into ABS_POS position forcing DIR direction
GoUntil(ACT,DIR,SPD)	100	0	ACT	01	DIR	Performs a motion in DIR direction with speed SPD until SW is closed, the ACT action is executed then a SoftStop takes place.
ReleseSW(ACT, DIR)	100	1	ACT	01	DIR	Performs a motion in DIR direction at minimum speed until the SW is released (open), the ACT action is executed then a HardStop takes place.
GoHome	011	1	0	00	0	Brings the motor into HOME position
GoMark	011	1	1	00	0	Brings the motor into MARK position
ResetPos	110	1	1	00	0	Resets the ABS_POS register (set HOME position)
ResetDevice	110	0	0	00	0	Device is reset to power-up conditions.
SoftStop	101	1	0	00	0	Stops motor with a deceleration phase
HardStop	101	1	1	00	0	Stops motor immediately
SoftHIZ	101	0	0	00	0	Puts the bridges into high impedance status after a deceleration phase
HardHIZ	101	0	1	00	0	Puts the bridges into high impedance status immediately
GetStatus	110	1	0	00	0	Returns the STATUS register value
RESERVED	111	0	1	01	1	RESERVED COMMAND
RESERVED	111	1	1	00	0	RESERVED COMMAND

Obrázek A.1: Přehled příkazů k dispozici pro ovládání motoru pomocí L6470[3]

A. Příkazy a registry k ovládní motoru

Address [Hex]	Register name	Register function	Len. [bit]	Reset Hex	Reset value	Remarks ⁽¹⁾
h01	ABS_POS	Current position	22	000000	0	R, WS
h02	EL_POS	Electrical position	9	000	0	R, WS
h03	MARK	Mark position	22	000000	0	R, WR
h04	SPEED	Current speed	20	00000	0 step/tick (0 step/s)	R
h05	ACC	Acceleration	12	08A	125.5e-12 step/tick ² (2008 step/s ²)	R, WS
h06	DEC	Deceleration	12	08A	125.5e-12 step/tick ² (2008 step/s ²)	R, WS
h07	MAX_SPEED	Maximum speed	10	041	248e-6 step/tick (991.8 step/s)	R, WR
h08	MIN_SPEED	Minimum speed	13	000	0 step/tick (0 step/s)	R, WS
h15	FS_SPD	Full-step speed	10	027	150.7e-6 step/tick (602.7 step/s)	R, WR
h09	KVAL_HOLD	Holding K _{VAL}	8	29	0.16-VS	R, WR
h0A	KVAL_RUN	Constant speed K _{VAL}	8	29	0.16-VS	R, WR
h0B	KVAL_ACC	Acceleration starting K _{VAL}	8	29	0.16-VS	R, WR
h0C	KVAL_DEC	Deceleration starting K _{VAL}	8	29	0.16-VS	R, WR
h0D	INT_SPEED	Intersect speed	14	0408	15.4e-6 step/tick (61.5 step/s)	R, WH
h0E	ST_SLP	Start slope	8	19	0.038% s/step	R, WH
h0F	FN_SLP_ACC	Acceleration final slope	8	29	0.063% s/step	R, WH
h10	FN_SLP_DEC	Deceleration final slope	8	29	0.063% s/step	R, WH
h11	K_THERM	Thermal compensation factor	4	0	1.0	R, WR
h12	ADC_OUT	ADC output	5	XX ⁽²⁾		R
h13	OCD_TH	OCD threshold	4	8	3.38A	R, WR
h14	STALL_TH	STALL threshold	7	40	2.03A	R, WR
h16	STEP_MODE	Step mode	8	7	128 microsteps	R, WH
h17	ALARM_EN	Alarm enable	8	FF	All alarms enabled	R, WS
h18	CONFIG	IC configuration	16	2E88	Internal oscillator, 2 MHz OSCOUT clock, supply voltage compensation disabled, overcurrent shutdown enabled, slew rate = 290 V/μs PWM frequency = 15.6 kHz.	R, WH
h19	STATUS	Status	16	XXXX ⁽²⁾	High impedance state, UVLO/Reset flag set.	R
h1A	RESERVED	Reserved address				
h1B	RESERVED	Reserved address				

1. R: readable, WH: writable only when outputs are in high impedance, WS: writable only when motor is stopped, WR: always writable.

2. According to startup conditions.

Obrázek A.2: Přehled parametrů k dispozici pro ovládní nebo sledování motoru^[3]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	0	0	0	DIR	From host
X	X	X	X	SPD (Byte 2)				From host
SPD (Byte 1)								From host
SPD (Byte 0)								From host

Obrázek A.3: Struktura příkazu Run[3]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	1	0	0	0	0	From host

Obrázek A.4: Struktura příkazu SoftStop[3]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	1	1	0	0	0	From host

Obrázek A.5: Struktura příkazu HardStop[3]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
0	0	1	PARAM						From host
ANS Byte 2 (if needed)								To host	
ANS Byte 1 (if needed)								To host	
ANS Byte 0								To host	

Obrázek A.6: Struktura příkazu GetParam[3]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	1	0	1	0	0	0	0	From host
STATUS MSByte								To host
STATUS LSByte								To host

Obrázek A.7: Struktura příkazu GetStatus[3]