

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA ELEKTROTECHNICKÁ**

**BAKALÁŘSKÁ PRÁCE**

**2017**

**Václav Hejný**

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA ELEKTROTECHNICKÁ**

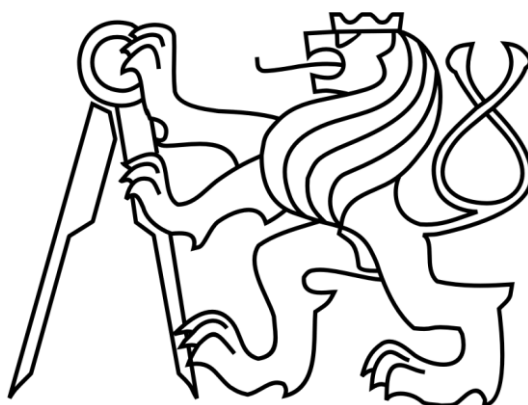


**BAKALÁŘSKÁ PRÁCE**

**2017**

**Václav Hejný**

**České vysoké učení technické v Praze**  
**fakulta elektrotechnická**  
**katedra mikroelektroniky**



**Bakalářská práce**

**Řídicí jednotka pro ovládání tří bipolárních krokových motorů**

**Autor:** Hejný Václav

**Vedoucí práce:** Ing. Teplý Tomáš

**2017**



## Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum:

.....

podpis bakalanta

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hejný** Jméno: **Václav** Osobní číslo: **434849**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra mikroelektroniky**  
Studijní program: **Komunikace, multimédia a elektronika**  
Studijní obor: **Aplikovaná elektronika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Řídicí jednotka pro ovládání tří bipolárních krokových motorů**

Název bakalářské práce anglicky:

**Triple Bipolar Stepper Motor Driver**

Pokyny pro vypracování:

Navrhněte a sestrojte řídicí jednotku pro ovládání tří bipolárních krokových motorů do 2,5 A s funkcí mikrokrokování. Mimo přímé řízení jednotlivých krokových motorů ovládací prvky na řídicí jednotce bude možné řídit motory z PC prostřednictvím rozhraní USB.

Postup vypracování:

- 1) Proveďte rešerši z oblasti řízení krokových motorů a vyberte vhodný způsob pro řízení bipolárních krokových motorů do 2,5 A s funkcí mikrokrokování.
- 2) Navrhněte obvodové zapojení včetně desek plošných spojů a vlastní návrh realizujte.
- 3) Ověřte funkčnost a změřte základní parametry realizovaného systému.
- 4) Zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

- [1] Barnett, R., H. - Cox, S. - O'Cull, L.: Embedded C Programming and the Atmel AVR 2nd Edition, Cengage learning, Delmar 2007
- [2] Závodský, O.: Programujeme AVR v jazyku C, 2012
- [3] Kernighan, B., W. - Ritchie, D., M.: Programovací jazyk C, Computer Press, a.s., Brno 2006
- [4] Záhlava V.: Návrh a konstrukce desek plošných spojů, BEN, Praha 2011

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Tomáš Teplý, katedra mikroelektroniky FEL**

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


Datum zadání bakalářské práce: **10.02.2017**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce: **10.09.2018**

  
Podpis vedoucí(ho) práce

  
Podpis vedoucí(ho) ústavu/katedry

  
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.

20. 3. 2017  
Datum převzetí zadání

Hejný  
Podpis studenta

**Anotace:**

Tato bakalářská práce se zabývá krokovými motory, možnostmi jejich řízení a popisem navržené a realizované řídicí jednotky pro řízení tří bipolárních krokových motorů. Pro řízení motorů jsou využity integrované drivery DRV8825 od výrobce Texas Instruments s možností mikrokrokování, které jsou řízeny jednočipovým mikroprocesorem Microchip Atmel ATMEGA2560. Motory je možné řídit joystickem přímo na řídicí jednotce nebo pomocí aplikace v počítači přes sběrnici USB.

**Summary:**

This bachelor's thesis deals with stepper motors, the possibilities of their driving and describes the developed control unit for driving of three bipolar stepper motors. For driving of motors, the integrated drivers DRV8825 from the producer Texas Instruments are used with the possibility of microstepping mode and these drivers are controlled by the microcontroller Microchip Atmel ATMEGA2560. These motors can be controlled by the joystick on the driving unit or by means of a PC application via USB bus.





## Obsah

1.1. Seznam obrázků .....	III
1.2. Seznam tabulek .....	IV
2. Úvod.....	1
3. Teoretická část .....	2
3.1. Krokový motor .....	2
3.1.1. Princip krokového motoru.....	2
3.2. Krok .....	3
3.3. Typy krokových motorů.....	3
3.3.1. Aktivní krokové motory .....	3
3.3.2. Pasivní krokové motory .....	3
3.3.3. Hybridní krokové motory.....	3
3.3.4. Lineární krokové motory.....	3
3.4. Způsoby řízení.....	4
3.4.1. Jednofázové řízení.....	4
3.4.2. Dvoufázové řízení .....	4
3.4.3. Řízení s celým krokem.....	4
3.4.4. Řízení s polovičním krokem .....	4
3.4.5. Mikrokování.....	4
3.4.6. Unipolární řízení .....	5
3.4.7. Bipolární řízení.....	6
3.5. Zpětná vazba .....	7
3.6. Integrované drivery .....	7
3.6.1. Unipolární řízení .....	7
3.6.2. Bipolární řízení.....	8
3.7. Driver DRV8825.....	8
3.7.1. Vstupní signály driveru .....	8
3.8. Mikrokontrolér .....	9
3.8.1. Microchip PIC.....	9
3.8.2. Microchip Atmel AVR.....	9
4. Praktická část .....	10
4.1. Návrh součástky DRV8825.....	10
4.1.1. Footprint.....	10
4.1.2. Schematická značka .....	10

4.2. Schémata navrženého zapojení .....	11
4.2.1. Napájení obvodu.....	11
4.2.2. Připojení joysticku.....	12
4.2.3. Zapojení driverů .....	12
4.2.4. Displej TFT LCD .....	13
4.2.5. Ovládací tlačítka.....	13
4.2.6. Vstupy pro koncové dorazy.....	14
4.2.7. Zapojení převodníku USB/UART .....	14
4.3. Plošný spoj .....	15
4.4. Zvolený způsob řízení .....	16
4.4.1. Řízení driveru a výpočet pozic motorů .....	16
4.4.2. Řízení motorů joystickem .....	18
4.4.3. Popis nastavení a ovládání řídicí jednotky .....	19
4.4.4. Řízení motorů z počítače.....	20
4.5. Zhotovená řídicí jednotka.....	23
5. Závěr.....	25
6. Literatura .....	26
7. Použitý software .....	27
8. Seznam příloh.....	28
8.1. Pinout Atmega2560.....	29
8.2. Schéma řídicí jednotky – 1. část.....	30
8.3. Schéma řídicí jednotky – 2. část.....	31
8.4. Plošný spoj .....	32

## 1.1. Seznam obrázků

Obr. 1 Princip krokového motoru.....	2
Obr. 2 Lineární krokový motor.....	3
Obr. 3 Schéma unipolárního zapojení.....	5
Obr. 4 Schéma bipolárního zapojení.....	6
Obr. 5 Výstupní signály rotačního enkodéru.....	7
Obr. 6 Footprint pouzdra SOP28 s thermal padem.....	10
Obr. 7 Navržená schematická značka pro obvod DRV8825.....	10
Obr. 8 Schéma napájecího obvodu.....	11
Obr. 9 Schéma zapojení lineárního stabilizátoru 7805.....	11
Obr. 10 Schéma konektorů pro připojení joysticku.....	12
Obr. 11 Schéma zapojení driveru.....	12
Obr. 12 Použitý displej.....	13
Obr. 13 Schéma zapojení tlačítek.....	13
Obr. 14 Schéma zapojení vstupů koncových dorazů posuvů.....	14
Obr. 15 Schéma zapojení převodníku USB/UART.....	14
Obr. 16 3D model osazené desky plošného spoje.....	15
Obr. 17 Graf rozběhové tabulky.....	27
Obr. 18 Tlačítka řídicí jednotky.....	19
Obr. 19 Okno pozic motorů.....	19
Obr. 20 Okno menu.....	19
Obr. 21 Okno pro nastavení velikostí kroků.....	20
Obr. 22 Okno pro nastavení typů závitů posuvů.....	20
Obr. 23 Počítačová aplikace pro ovládání řídicí jednotky.....	22
Obr. 24 Fotografie zapnuté řídicí jednotky s připojeným motorem.....	23
Obr. 25 Zadní strana řídicí jednotky .....	24
Obr. 26 Osazený plošný spoj.....	24

## 1.2. Seznam tabulek

Tab. 1 Jednofázové unipolární řízení s celým krokem.....	5
Tab. 2 Dvoufázové unipolární řízení s celým krokem.....	5
Tab. 3 Unipolární řízení s polovičním krokem.....	5
Tab. 4 Jednofázové bipolární řízení s celým krokem.....	6
Tab. 5 Dvoufázové bipolární řízení s celým krokem.....	6
Tab. 6 Bipolární řízení s polovičním krokem.....	6
Tab. 7 Tabulka kombinací vstupů určujících velikost kroku.....	8

## 2. Úvod

Moje bakalářská práce se zabývá problematikou řízení krokových motorů. Tyto motory jsou jedněmi z nejdůležitějších točivých strojů v oboru automatizace, protože nám umožňují přesné posuvy. Vzhledem k této jejich vlastnosti je proto najdeme téměř v každém elektrickém zařízení, kde jsou nějaké posuvy. Krokové motory můžeme najít například v tiskárnách pro posuv tiskové hlavy a papíru, v optických mechanikách, kde slouží k posuvu optické čtecí hlavy, dále například zajišťují posuvy u CNC obráběcích strojů. A mohli bychom jmenovat i mnoho dalších příkladů aplikací krokových motorů.

K rozhodnutí pro vytvoření této práce mě vedlo to, že na trhu není dostupná žádná levná řídicí jednotka pro řízení tří bipolárních krokových motorů, která by sloužila například k řízení posuvů u tří-osých stolků. Místo řídicí jednotky je ovšem možné použít vývojový kit Arduino se shieldem pro řízení krokových motorů. Pro tuto variantu řízení za pomoci Arduina existuje celá řada programů i mnoho počítačových aplikací. Avšak řízení krokových motorů s Arduinem není vhodné, pokud chceme tyto jednotky prodávat, protože jsou cenově dražší než mnou navržená řídicí jednotka. Mimo jiné tyto jednotky na bázi Arduina jsou obvykle řízeny jen z počítače a nemají možnost připojení joysticku. Moje bakalářská práce popisuje řídicí jednotku, kterou je možné řídit až tři motory, a to dvěma způsoby.

První způsob řízení spočívá v tom, že motory jsou řízeny joystickem přímo na řídicí jednotce, kdy se pozice motorů zobrazuje na TFT LCD displeji, který je také součástí řídicí jednotky. Druhý způsob řízení představuje možnost řídit motory pomocí aplikace v počítači, který komunikuje se zařízením po sběrnici USB. Kromě toho je možné u obou způsobů řízení nastavit mikrokrokování v šesti režimech „celý krok až 1/32 kroku“.

Řídicí jednotka, kterou popisuji zde v mé práci, je navržena pouze pro lineární posuvy se závitovou tyčí. Pro univerzálnost je možnost nastavit typ závitu M2 až M10. Informace, o který závit se jedná, je uložena do paměti EEPROM, která umožňuje permanentní uložení dat, a to z toho důvodu, aby obsluha zařízení nemusela při každém zapnutí nastavovat konkrétní typ závitu pro každý motor, aby výpočty pozic byly korektní.

Cílem práce bylo zvolit součástky vhodné pro řízení bipolárních krokových motorů, navrhnout obvodové řešení, sestavit výše popsanou řídicí jednotku a otestovat funkčnost mého řešení.

## 3. Teoretická část

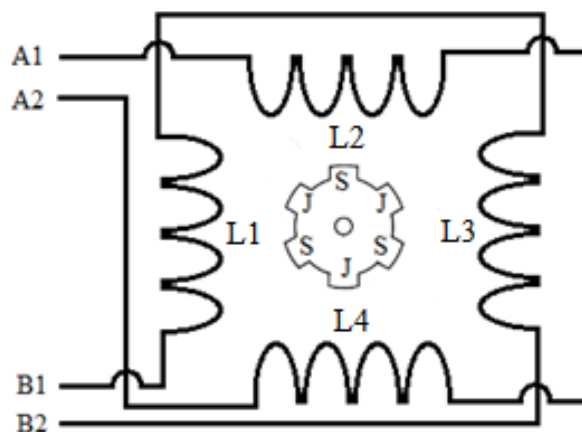
### 3.1. Krokový motor

Krokový motor je speciálním typem synchronního stroje. Na rozdíl od klasického synchronního stroje se nenapájí třemi fázemi střídavého proudu, ale napájí se pomocí impulsů stejnosměrného proudu. Pohyb rotoru je při nízkých otáčkách nespojitý. Pohybuje se po takzvaných krocích. Počet kroků na otáčku je dán typem motoru a závisí na počtu pólových párů na rotoru. V případě, že potřebujeme přesnější posuv, můžeme využít mikrokrokování. K pohybu motoru je vždy potřeba řídicí elektronika, která ovládá rychlost otáčení a směr. Tento typ motorů se vyznačuje velkou mechanickou odolností, dlouhou životností a bezúdržbovým provozem. Nevýhodami těchto strojů jsou ztráty kroků při nadměrném zatížení a rychlém rozběhu, ztráta kroků při mechanickém zakmitávání, kterou jsem vyzoroval pouze při nízkých otáčkách a v režimu celého kroku.

V tomto typu motoru dochází k magnetickým přechodovým jevům, a proto je omezena rychlost otáčení motoru na pouhé stovky kroků za sekundu. Maximální rychlost otáčení závisí na typu motoru a jeho zatížení. Při překročení této rychlosti dochází ke ztrátám kroků a v případě akcelerace se motor nemusí vůbec roztočit.

#### 3.1.1. Princip krokového motoru

Krokový motor se skládá ze stejných částí jako každý jiný motor, a to ze statoru a rotoru. Na statoru jsou umístěny cívky a na rotoru pólové nástavce podle typu motoru – buď z permanentních magnetů, nebo z magneticky měkkého kovu. Na obrázku Obr. 1 je uprostřed vidět rotor se šesti pólovými nástavci, kde se střídají severní a jižní póly. Okolo rotoru je stator se čtyřmi cívkami. Dle propojení protilehlých cívek je poznat, že se jedná o motor pro bipolární řízení. Konstrukce motoru pro unipolární řízení se liší jen způsobem zapojení cívek. Při vedení proudu skrze páru cívek dochází k indukci magnetického pole a rotor se pootočí tak, aby byl v magneticky klidové poloze. Pro další pootočení je nutné zavést proud do druhého páru cívek. Po ustálení rotoru v klidové poloze je rotor pevně držen ve své poloze magnetickým polem generovaným cívkami. Z toho plyne, že motor je neustále napájen a spotřebovává energii.



Obr. 1 Krokový motor

### 3.2. Krok

Krok je reakce motoru na jeden řídicí impuls. Rotor se pootočí z klidové polohy do nejbližší magneticky klidové polohy. Úhel kroku je úhel, o který se pootočí hřídel při pohybu o jeden krok.

$$\alpha = \frac{360^\circ}{N} [^\circ]$$

Vztah 1 Výpočet úhlu jednoho kroku,

kde N je počet kroků na otáčku. Výsledný úhel je počítán ve stupních. Úhel kroku u mnou použitého motoru s 200 kroky na otáčku je 1,8°.

### 3.3. Typy krokových motorů

#### 3.3.1. Aktivní krokové motory

Tento typ krokového motoru je nejběžnější. Rotor tohoto motoru je složen z permanentních magnetů. Motor funguje na principu, kdy se magnety rotoru přitahují, popřípadě odpuzují s vytvořeným magnetickým polem cívek s pólovými nástavci. Princip řízení krokových motorů bude vysvětlen později.

#### 3.3.2. Pasivní krokové motory

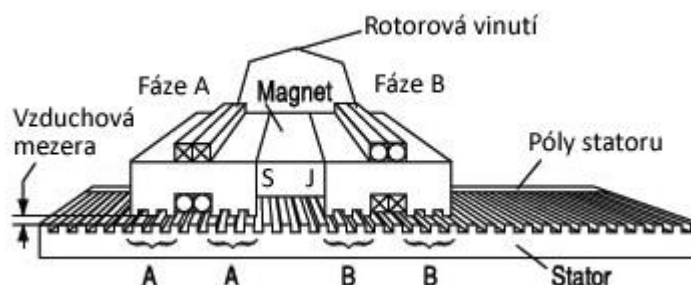
Tyto krokové motory fungují na podobném principu jako aktivní motory. Namísto magnetických sil je zde využita změna reluktance magnetického obvodu při otáčení rotoru. Reluktance je magnetický odpor. V případě, že jsou zuby rotoru v jedné ose s pólovými nástavci vybuzené cívkou, je reluktance nejmenší. V případě, že nejsou zuby rotoru v jedné ose s vybuzenými pólovými nástavci, vzniká moment, který otáčí rotorem motoru. Stav s nejmenší reluktancí se nazývá stabilní magnetická poloha.

#### 3.3.3. Hybridní krokové motory

Tyto krokové motory jsou kombinací aktivních a pasivních motorů.

#### 3.3.4. Lineární krokové motory

Stator je rozvinut do přímky, kdy slouží jako dráha, nad kterou se posouvá rotorové vinutí s dvěma cívkami.



Obr. 2 Lineární krokový motor [6]

### 3.4. Způsoby řízení

Pro řízení krokových motorů existuje několik různých způsobů, které se v mnohém liší. Tyto způsoby dělíme podle velikosti kroku a počtu buzených cívek na následující možnosti.

#### 3.4.1. Jednofázové řízení

Magnetické pole je generováno pouze jednou cívkou pro unipolární zapojení, pro bipolární zapojení je magnetické pole generováno pouze jedním párem cívek.

#### 3.4.2. Dvoufázové řízení

Magnetické pole generují dvě sousední cívky, a tím se zvýší spotřeba elektrické energie a kroutící moment. To znamená, že motor můžeme více zatížit.

#### 3.4.3. Řízení s celým krokem

Tento způsob řízení motoru je nejjednodušší a také nejméně přesný, protože na jednu otáčku motoru je potřeba vykonat pouze tolik kroků, jako má rotor pólových nastavců.

#### 3.4.4. Řízení s polovičním krokem

Při jednom kroku se rotor pootočí o poloviční úhel, než při řízení s celým krokem. Způsob buzení cívek se liší pro unipolární a bipolární řízení. Buzení cívek je popsáno tabulkou Tab. 3 pro unipolární řízení a tabulkou Tab. 6 pro bipolární řízení. Kroky je možné dělit i na menší díly, ale k tomu je potřeba složitější řídicí obvod, který umožňuje mikrokrokování.

#### 3.4.5. Mikrokrokování

Krok se dělí na menší díly tak, že v jedné fázi postupně snižujeme proud a v druhé fázi jej zvyšujeme.

$$I_A = I \cos\left(\frac{k\varphi}{N}\right) \text{ [A]}$$

$$I_B = I \cos\left(\frac{k\varphi}{N} - \frac{\pi}{2}\right) \text{ [A]}$$

Vztahy 2, 3 pro výpočet proudů cívkami motoru,

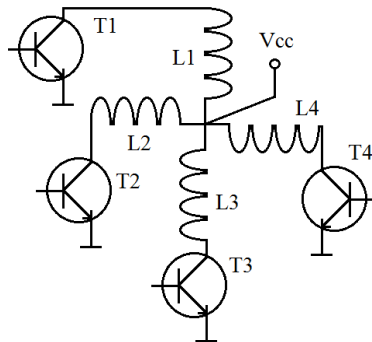
kde N je počet mikrokroků na krok, k je index mikrokroku, kdy  $k = 0, 1, 2, \dots$ , I je maximální budící proud.

Pro regulaci proudu jednotlivých cívek u mikrokrokování se využívá pulzně šířková modulace (PWM) kvůli tomu, že je možné výstupní výkonové tranzistory používat ve spínacím režimu, a tím dosáhnout minimálních tepelných ztrát. Díky tomu mohou být integrované drivery v malých pouzdrech, u kterých není potřeba veliký chladič.



### 3.4.6. Unipolární řízení

Při unipolárním řízení prochází proud cívkami pouze jedním směrem. Motor s tímto buzením má nejmenší spotřebu energie, ale má i nejnižší kroutící moment. Výhodou tohoto řízení je snadné zapojení řídicího obvodu dle obrázku Obr. 3, kdy společnou elektrodu připojíme na napájecí napětí a každou cívku připojíme přes tranzistory k zemi. K buzení cívek dochází sepnutím jednotlivých tranzistorů. Způsoby řízení jsou popsány tabulkami Tab. 1 až Tab. 3.



Obr. 3 Schéma unipolárního zapojení

Tab. 1 Jednofázové unipolární řízení s celým krokem [7]

Cívka 1	I	0	0	0
Cívka 2	0	I	0	0
Cívka 3	0	0	I	0
Cívka 4	0	0	0	I

Tab. 2 Dvoufázové unipolární řízení s celým krokem [7]

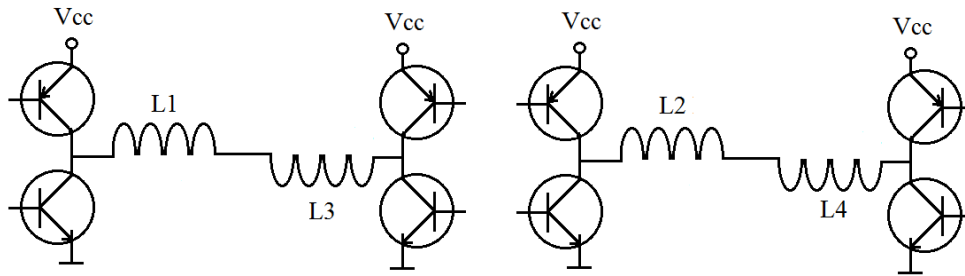
Cívka 1	I	0	0	I
Cívka 2	I	I	0	0
Cívka 3	0	I	I	0
Cívka 4	0	0	I	I

Tab. 3 Unipolární řízení s polovičním krokem [7]

Cívka 1	I	I	0	0	0	0	0	I
Cívka 2	0	I	I	I	0	0	0	0
Cívka 3	0	0	0	I	I	I	0	0
Cívka 4	0	0	0	0	0	I	I	I

### 3.4.7. Bipolární řízení

Při tomto řízení prochází proud vždy protilehlými cívkami. Cívky jsou zapojené tak, že mají navzájem opačné magnetické pole. Při tomto řízení je vyšší spotřeba elektrické energie, ale motor poskytuje vyšší kroutící moment. Pro bipolární řízení je potřeba složitější obvod než pro unipolární řízení. K řízení motoru jsou potřeba dva tranzistorové H-můstky (pro každou dvojici cívek jeden můstek). Zapojení cívek motoru můžete vidět na obrázku Obr. 4. Různé způsoby řízení jsou popsány tabulkami Tab. 4 až Tab. 6.



Obr. 4 Schéma bipolárního zapojení

Tab. 4 Jednofázové bipolární řízení s celým krokem [7]

Cívka 1	-I	0	+I	0
Cívka 2	0	-I	0	+I
Cívka 3	+I	0	-I	0
Cívka 4	0	+I	0	-I

Tab. 5 Dvoufázové bipolární řízení s celým krokem [7]

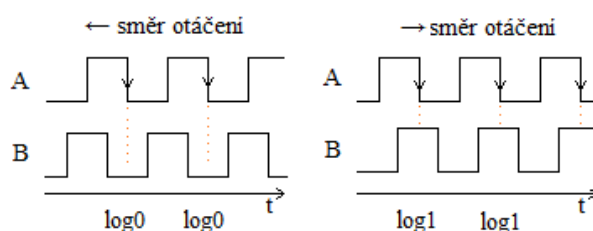
Cívka 1	-I	+I	+I	-I
Cívka 2	-I	-I	+I	+
Cívka 3	+I	-I	-I	+
Cívka 4	+I	+I	-I	-I

Tab. 6 Bipolární řízení s polovičním krokem [7]

Cívka 1	-I	-I	0	+I	+I	+I	0	-I
Cívka 2	0	-I	-I	-I	0	+I	+I	+I
Cívka 3	+I	+I	0	-I	-I	-I	0	+I
Cívka 4	0	+I	+I	+I	0	-I	-I	-I

### 3.5. Zpětná vazba

Pokud krokové motory nadměrně zatěžujeme a zároveň nechceme, aby docházelo k nepřesnostem posuvů vlivem ztráty kroků, je nutné zavést takzvanou zpětnou vazbu. Se zpětnou vazbou se nepočítá počet generovaných budících impulsů pro drivery motorů, ale sčítají se impulsy generované vhodným senzorem, který sleduje otáčení hřídele motoru. Pro tento způsob měření polohy jsou vhodné rotační enkodéry, které nám udávají relativní polohu a mohou být buď mechanické, nebo optické. Pro správné sledování pozic posuvů s motory, které mají 200 kroků na otáčku a řídicí jednotku s možností mikrokrokování v režimu až 1/32 kroku, je nutné, aby měl rotační enkodér 6400 kroků na jednu otáčku, a to proto, aby bylo možné sledovat každý mikrokrok. To platí pro přímé spojení hřídelí motoru a enkodéru. Na obrázku Obr. 5 jsou vyobrazeny průběhy výstupních signálů A a B z rotačního enkodéru. Při otáčení hřídelí enkodéru se na jeho výstupu generují dva obdélníkové signály, které jsou vzájemně posunuty o  $\frac{1}{4}$  periody. Směr posunu signálu je závislý na směru otáčení enkodéru. Pro vyhodnocování těchto signálů je nutné využít u AVR alespoň jeden vstup externího přerušení, který je citlivý na spádovou nebo náběžnou hranu signálu. Při přerušení je možné vyhodnotit směr otáčení podle toho, zdali je druhý signál v log1 nebo log0. Při každém přerušení by se přičítaly vykonané kroky do proměnné v mikrokontroléru. Tímto způsobem je možné dosáhnout přesných posuvů bez ohledu na ztráty kroků. Motor dojede vždy do nastavené polohy.



Obr. 5 Výstupní signály rotačního enkodéru

V řídicí jednotce není implementována funkce pro odečet poloh ze zpětné vazby, protože zvolený mikrokontrolér by nestíhal zpracovávat všechna přerušení. Pokud bych chtěl tuto funkci využívat, bylo by nutné volit nějaký modernější mikrokontrolér, který je taktovatelný vyšší frekvencí, nebo využít nějaké programovatelné hradlové pole.

### 3.6. Integrované drivery

V dnešní době se krokové motory již nebudí pomocí diskretních součástek, protože jednotlivé tranzistory by zabíraly spoustu místa na desce plošného spoje. Z toho důvodu se využívají integrované drivery, které nám umožňují menší rozměry finálního výrobku a dosažení nižšího počtu osazovaných součástek, a z toho vyplývá i nižší cena za použité součástky. Nižší ceny výrobku se dá dosáhnout tím, že se ušetří čas při programování, když se využijí integrované drivery s digitálními vstupy. U těchto driverů se programátor nezabývá principy řízení krokových motorů, ani tím, jaké tranzistory má budit. U těchto driverů obvykle stačí pouze nastavit, jakým směrem a v jakém režimu mikrokrokování se bude motor pohybovat. A pak už jen stačí generovat impulsy pro krokování.

#### 3.6.1. Unipolární řízení

Pro tento způsob řízení existuje několik integrovaných obvodů, které zpravidla neobsahují žádnou složitou řídicí logiku. Obvykle bývají využity jen obvody, ve kterých je integrován větší

počet výkonových tranzistorů na jednom čipu. Například obvod ULN2003 obsahuje sedm výkonových tranzistorů, které vydrží napětí do 50 V a proudové zatížení až 500 mA.

### 3.6.2. Bipolární řízení

U tohoto způsobu řízení existuje mnoho integrovaných obvodů, které lze využít. Tyto obvody obsahují složitější řídicí logický obvod, který je ovládán pomocí několika digitálních vstupů, dále obsahují dva tranzistorové H-můstky nebo integrovaný výstupní budič externích tranzistorových H-můstek. Díky jejich řídicím logickým obvodům je možné mikrokrokování u některých driverů až do velikosti kroku 1/256. Jako zástupce těchto obvodů bych rád uvedl DRV8825 od výrobce Texas Instruments. Dále bych chtěl uvést obvod TEA3718 od STMicroelectronics a obvod TB6560 od Tshiby. Kromě těchto obvodů existuje i mnoho jiných, které se liší z elektrického pohledu různými parametry, např. napájecím napětím a maximálním proudovým zatížením. Pro moji bakalářskou práci jsem zvolil níže popsany integrovaný driver DRV8825, který splňuje všechny požadavky dané zadáním práce. Tento driver jsem zvolil také proto, že se jedná o malý obvod v pouzdře SOP28 a je za příznivou cenu.

### 3.7. Driver DRV8825

Pro praktickou část mé bakalářské práce, která popisuje řízení motorů v praxi, jsem se rozhodl využít, jak jsem se již zmínil, jeden z vyráběných integrovaných bipolárních driverů DRV8825, který se používá pro řízení krokových motorů a je u něj možné maximální proudové zatížení motorem. Tento driver je také vhodný k využití pro moji bakalářskou práci proto, že dokáže motoru dodat proud až 2,5 A a lze jej napájet až 24 V. Driver dále umožňuje mikrokrokování, a to od režimu celého kroku až po 1/32 kroku.

#### 3.7.1. Vstupní signály driveru

K ovládní motoru má driver několik digitálních řídicích vstupů: STEP, DIR, ENABLE, SLEEP, RESET, a trojici vstupů MODE 0–2. Signál DIR slouží k řízení směru otáčení motoru, na vstup STEP se přivádějí řídicí impulsy, které slouží k pootočení motoru o jeden krok, a trojice vstupů MODE0, MODE1, MODE2 slouží k nastavení velikosti kroku. Dalšími řídicími vstupy na driveru jsou vstupy ENABLE, SLEEP a RESET. Tyto vstupy mají inverzní logiku, a proto je potřeba nastavit hodnoty ENABLE = log0, SLEEP = log1, RESET = log1 pro spuštění driveru.

Tab. 7 Tabulka kombinací vstupů určujících velikost kroku [12]

MODE2	MODE1	MODE0	Velikost kroku
0	0	0	celý krok
0	0	1	1/2 kroku
0	1	0	1/4 kroku
0	1	1	1/8 kroku
1	0	0	1/16 kroku
1	0	1	1/32 kroku
1	1	0	
1	1	1	

Tabulka Tab. 7 popisuje nastavení velikosti kroku v závislosti na nastavených logických hodnotách na vstupech MODE.

### 3.8. Mikrokontrolér

Mikrokontroléry jsou jednočipové počítače, které představují v dnešní době důležitou součást automatizační a řídicí techniky. Jsou to integrované obvody s vysokou hustotou integrace, které obsahují kompletní počítač obvykle harvardské koncepce. Kromě samotného procesoru a paměti obsahují také několik doplňujících periferních obvodů. Mezi běžně integrované periferie se řadí například analogově digitální převodníky, analogové komparátory, čítače 8bitové a 16bitové, které podporují i pulzně šířkovou modulaci (PWM). Mimoto čipy obsahují sériové sběrnice pro komunikaci s jinými obvody, jako například asynchronní jednotku UART (universal asynchronous receiver transmitter), synchronní jednotku SPI (serial peripheral interface) a I2C. Použití mikrokontrolérů je vhodné proto, že lze změnit fungování obvodu jen přeprogramováním paměti této jediné součástky bez nutnosti měnit zapojení. Mezi nejznámější výrobce mikrokontrolérů patří firmy Microchip Technology, Texas Instruments nebo ST Microelectronics. Velmi populární jsou 8bitové mikrokontroléry AVR nebo PIC od Microchipu kvůli velkému množství různých typů, a tak si můžeme vybrat nejvhodnější chip, který obsahuje jen potřebné periferie a množství paměti pro danou aplikaci.

#### 3.8.1. Microchip PIC

Mikrokontroléry harvardské architektury s redukovanou instrukční sadou od firmy Microchip se vyrábí 8bitové (PIC10, 12, 16, 18), 16bitové (PIC24, dsPIC30, 33) a 32bitové (PIC32). Rozděleny jsou podle výkonnosti a počtu instrukcí. Nejnižší řady (PIC10–16) mají pouze 35 instrukcí a používají se na nejjednodušší aplikace psané obvykle v assembleru. Vyšší řady mají 75 instrukcí a více, mají vylepšenou podporu vyšších programovacích jazyků. Procesory PIC jsou jedny z nejstarších mikrokontrolérů na trhu a používají se napříč celým spektrem řídicí techniky od domácích spotřebičů po měřicí přístroje.

#### 3.8.2. Microchip Atmel AVR

Mikrokontroléry AVR jsou 8bitové a 32bitové jednočipové mikropočítače harvardské architektury s redukovanou instrukční sadou. Jsou nástupcem řady AT89. Dělí se na čtyři základní skupiny. Asi nejznámější jsou ATtiny a ATmega. ATtiny jsou nejmenší mikrokontroléry mezi AVR, jsou 8bitové a vyrábějí se v provedení 6 až 32 pinů, hodí se pro jednoduché aplikace s krátkým programem. Řada ATmega se vyrábí v pouzdech od 28 do 100 pinů. Slouží pro středně náročné aplikace. Výkonově odpovídají řadě ATtiny, ale mají větší SRAM paměť i paměť programu a více vstupně výstupních pinů. Nejvýkonnějšími zástupci jsou 8bitové ATxmega a 32bitové AT32. Mikrokontroléry AVR vynikají velkou vybaveností periferními obvody, nízkou cenou v poměru s výkonem a univerzálností použití.

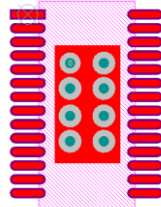
## 4. Praktická část

### 4.1. Návrh součástky DRV8825

Ještě před samotným návrhem schémat je nutné mít vytvořeny všechny potřebné součástky v návrhovém prostředí, ve kterém se vytváří návrh schémat a návrh plošného spoje. Jelikož jsem měl většinu součástek dostupných v knihovnách Altium designeru, nebylo nutné kreslit všechny součástky. Jediná součástka, která v knihovnách není, je driver DRV8825.

#### 4.1.1. Footprint

Při návrhu součástky je vždy potřeba nejprve vytvořit footprint, což jsou kontaktní plošky, na které se letují vývody součástek. Pro driver DRV8825 jsem musel vytvořit footprint pro pouzdro SOP28, a to podle popsanych rozměrů v datasheetu součástky. Dále bylo nutné vytvořit další kontaktní plochu pod chipem, která slouží pro odvod tepla ze součástky. Tato ploška se nazývá thermal pad, ve které bývá buď jedna velká prokovená díra pro ruční pájení, nebo více menších děr pro pájení přetavením. V thermal padu jsem vytvořil několik malých prokovených děr pro lepší odvod tepla do spodní vrstvy plošného spoje.



Obr. 6 Footprint pouzdra SOP28 s thermal padem

Po stranách footprintu na obrázku Obr. 6 jsou plošky, na které se letují vývody součástky. Uprostřed je thermal pad s thermal vias pro odvod tepla do spodní vrstvy plošného spoje, kde je rozlita měď s připojením k zemi. Velký vyšrafovaný fialový obdélník znázorňuje rozměry pouzdra součástky.

#### 4.1.2. Schematická značka

U návrhu schematické značky bylo potřeba nakreslit obdélník, ke kterému se vložil správný počet vývodů. V této schematické značce musí být vloženo 29 vývodů, aby bylo možné připojit thermal pad. Dále jsem musel pojmenovat vývody součástky, aby odpovídaly vývodům popsáním v datasheetu. Po importování footprintu odpovídají čísla padů číslům vývodů ve schematické značce. Výslednou schematickou značku můžete vidět na obrázku Obr. 7.

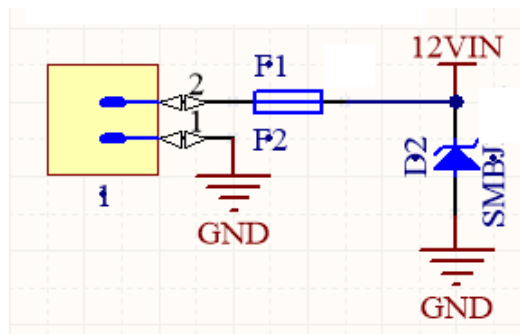
1	CP1	AOUT2	7
2	CP2	AOUT1	5
		VMA	4
6	ISENA		
9	ISENB	BOUT2	8
		BOUT1	10
24	MODE0	VMB	11
25	MODE1		
26	MODE2		
20	DIR	NC	23
22	STEP	BVREF	13
21	nENABLE	AVREF	12
		V3P3OUT	15
16	nRESET		
17	nSLEAP	VCP	3
18	nFAULT	GND	0
27	nHOME	GND	28
19	DACAY	GND	14

Obr. 7 Navržená schematická značka pro obvod DRV8825

## 4.2. Schémata navrženého zapojení

### 4.2.1. Napájení obvodu

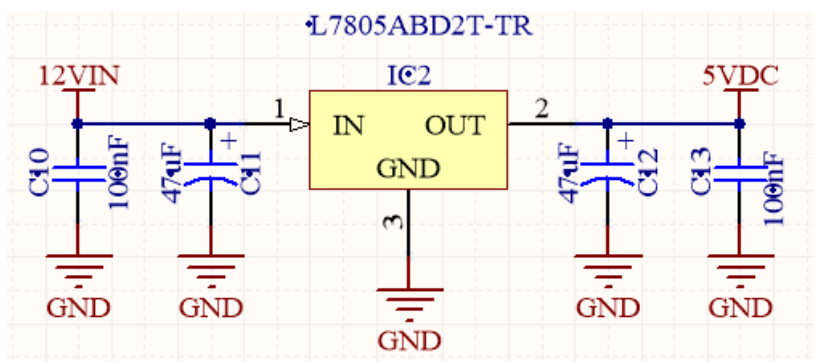
Obvod jsem navrhl tak, že je možné jej napájet stejnosměrným napětím v rozmezí 7–24 V. Tento interval napájecích napětí je určen lineárním stabilizátorem 7805, který v obvodu slouží pro stabilizování napětí pro napájení logiky. Tento stabilizátor začíná správně stabilizovat napětí, pokud je napájen alespoň napětím okolo 7 V. Stabilizátor funguje až do napětí 35 V, ale tohoto napětí nelze v obvodu dosáhnout, protože je ve schématu použit integrovaný obvod DRV8825, který podle dokumentace nesmí být napájen vyšším napětím než 24 V. Jako napájecí konektor jsem zvolil násuvnou svorkovnici od PTR s roztečí 5,08 mm.



Obr. 8 Schéma napájecího obvodu

Vlevo ve schématu na obrázku Obr. 8 je napájecí konektor, kde kladná napájecí elektroda je připojena přes 8ampérovou pojistku na ochranný jednocestný transil SMBJ24A, který funguje jako zenerova dioda a omezuje vyšší napětí na 24 V. Tímto zapojením je obvod chráněn i proti přepólování napájecího napětí, protože se uzavře proudová smyčka přes transil v propustném směru a přes pojistku v sérii. Tím se napájecí zdroj vyzkratuje a dojde k přepálení pojistky.

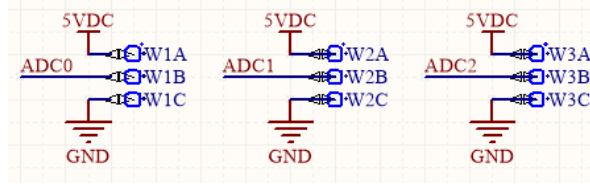
Pro napájení logických obvodů bylo potřeba vytvořit napětí o velikosti 5 V. To bylo možné vytvořit dvěma způsoby, a to lineárním stabilizátorem, nebo spínaným zdrojem. Jako zdroj 5 V jsem využil lineární stabilizátor 7805 vzhledem k jeho nízké ceně.



Obr. 9 Schéma zapojení lineárního stabilizátoru 7805

Jedná se o jednoduchý integrovaný obvod, který má pouze tři vývody. Vstup, kde se připojuje vstupní napětí, zem a výstup, kde je 5 V. Pro správnou činnost je nutné přidat elektrolytické a keramické kondenzátory ke vstupu a výstupu, aby nedocházelo ke kmitání obvodu.

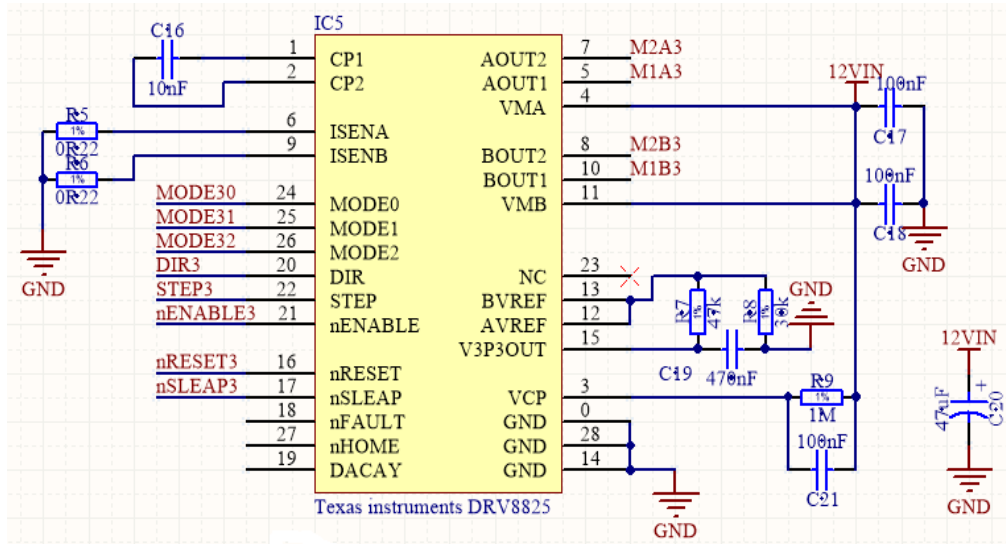
#### 4.2.2. Připojení joysticku



Obr. 10 Schéma konektorů pro připojení joysticku

Ve schématu na obrázku Obr. 10 je vidět, že pro každou osu jsou potřeba tři piny na desce plošného spoje. Jelikož je joystick potenciometrový, je nutné jej napájet, a to stejným napětím jako je reference analogově digitálního převodníku. Proto je napájen 5 V proti zemi. Výstupem každé osy je analogový signál, kde je napětí úměrné poloze páky joysticku. Signálové vodiče ADC0 až ADC2 jsou připojeny na analogové vstupy mikrokontroléru.

#### 4.2.3. Zapojení driverů



Obr. 11 Schéma zapojení driveru

Ve schématu na obrázku Obr. 11 vlevo jsou vidět digitální vstupy a vpravo výstupy pro připojení párů cívek motoru. Tyto výstupy jsou připojeny na čtyřpinové násuvné svorkovnice PTR na zadní straně řídicí jednotky. Hodnoty rezistorů a kapacitorů ve schématu jsou voleny dle doporučeného zapojení v dokumentaci driveru, kdy rezistory R5 a R6 jsou prvky sloužící k omezení maximálního výstupního proudu. Pro výpočet požadovaného proudu fáze lze využít tento vztah:

$$I_f = \frac{V_{ref}}{A_v * R} = \frac{1,25}{5 * R} \quad [V, \Omega, A]$$

Vztah 4 Výpočet proudu fáze v závislosti na omezovacím rezistoru

Ve vztahu je dáno pro tento driver  $V_{ref} = 1,25$  a  $A_v = 5$ . Pro dosažení maximálního fázového proudu  $I_f = 1,25$  A lze spočítat, že  $R = 0,2 \Omega$ . Z toho plyne, že omezovací rezistor by neměl mít nižší hodnotu než  $0,2 \Omega$ , a proto jsem zvolil  $R_5$  a  $R_6 = 0,22 \Omega$ .



#### 4.2.4. Displej TFT LCD

V rámci návrhu praktické části bakalářské práce jsem zvolil barevný grafický displej TFT (thin film transistor) LCD (liquidcrystal display), a to z toho důvodu, že v dnešní době výrobek se znakovým LCD displejem nevypadá moderně. Tento displej je ovládán sériově přes jednotku SPI (serial peripheral interface).

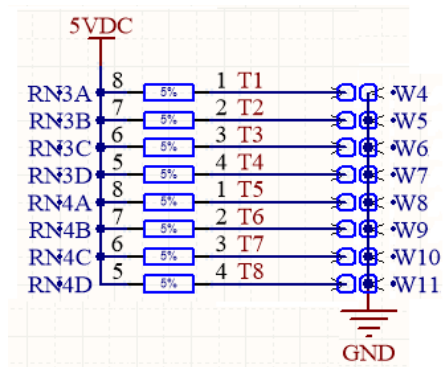


Obr. 12 Použitý displej

Použitý displej má úhlopříčku 1,8 palce s rozlišením 128x160 pixelů, což jsou zobrazované body. V displeji je integrovaný řadič ST7735, který přijímá data od mikrokontroléru a zobrazuje je na správné pixely, kdy dva byty popisují barvu jednoho pixelu. Jas červeného subpixelu je určen 5 bity, u zeleného subpixelu je jas dán 6 bity a jas modrého subpixelu je popsán 5 bity. Tento displej je možné pořídit za rozumnou cenu, například na internetové stránce ebay.com. V programu jsem pro komunikaci s tímto displejem využil již hotovou knihovnu pro AVR, která komunikuje s řadičem displeje přes hardwarovou jednotku SPI.

#### 4.2.5. Ovládací tlačítka

Pro ovládání řídicí jednotky bylo potřeba použít několik tlačítek, která jsou umístěna na ovládacím panelu řídicí jednotky. Tato tlačítka slouží k procházení menu a pomocí nich se nastavují potřebné parametry, jako jsou například velikosti kroků.

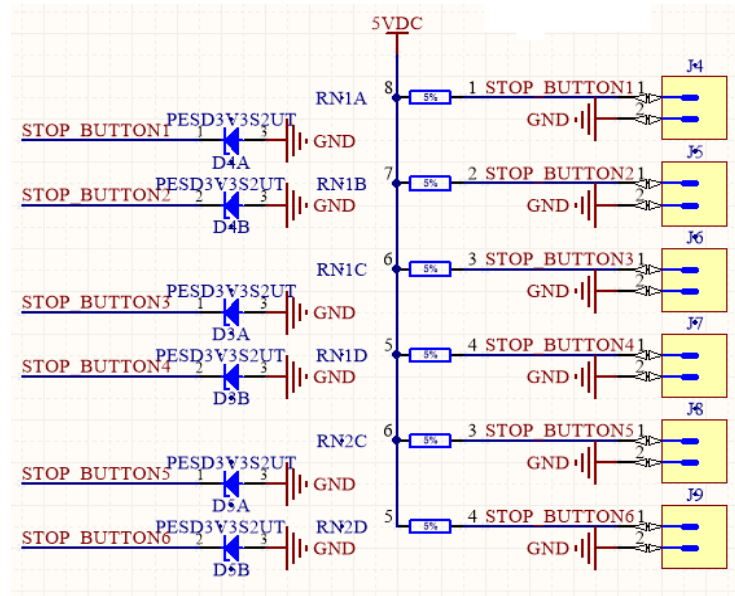


Obr. 13 Schéma zapojení tlačítek

Tlačítka jsou připojena na konektor, který je ve schématu na obrázku Obr. 13 znázorněn vpravo. Každé tlačítko je připojeno jedním vývodem k zemi a druhým přes pull up rezistor k napájení. Pull up rezistory slouží pro dosažení správných logických úrovní, kdy je na vstupech mikrokontroléru T1 až T8 nastaveno 5 V, pokud nejsou stisknuta příslušná tlačítka. Hodnota 5 V odpovídá logické jedničce. Při stisku tlačítka je na těchto vstupech připojena zem, což odpovídá logické nule. Jako pull up rezistory jsem zvolil rezistorové sítě s odporem 1 k $\Omega$ . To jsou čtyři rezistory v jednom pouzdře 1206.

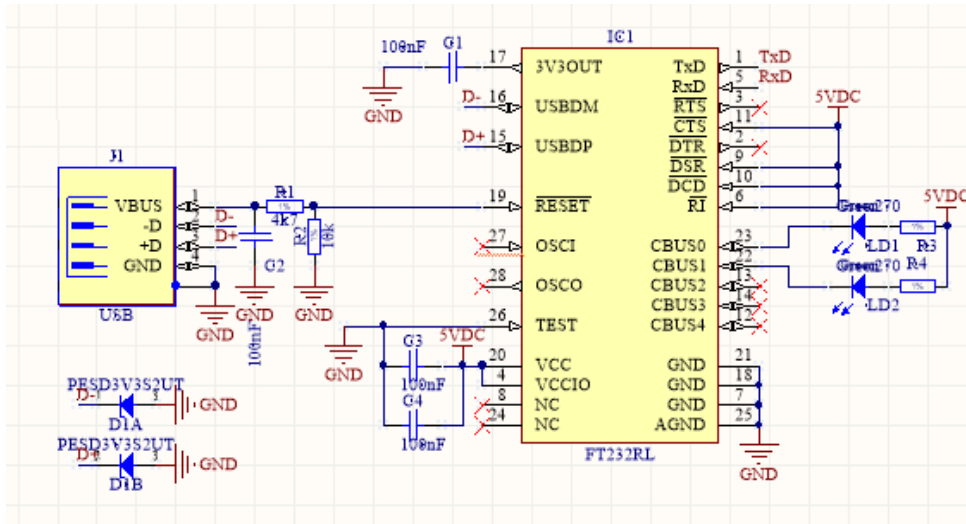
#### 4.2.6. Vstupy pro koncové dorazy

Podobným způsobem jako ovládací tlačítka jsem navrhl vstupy pro koncové dorazy posuvů s tím rozdílem, že tyto vstupy jsou vyvedeny na konektory na zadní straně řídicí jednotky. Vstupy jsou na rozdíl od ovládacích tlačítek připojeny ke vstupům externího přerušování a jsou chráněny proti přepětí ochrannými transily. I v tomto případě je využit externí pull up rezistor s odporem 1 k $\Omega$ . Při výběru vhodných konektorů jsem se rozhodl pro násuvné svorkovnice s roztečí 3,81 mm, vyráběné firmou PTR. Tyto svorkovnice jsem vybral z toho důvodu, že jsou vyráběny jak vícepinové varianty, tak varianty s různou roztečí. To mi umožnilo mít na řídicí jednotce podobné konektory pro napájení výstupů pro motory a vstupů koncových dorazů.



Obr. 14 Schéma zapojení vstupů koncových dorazů posuvů

#### 4.2.7. Zapojení převodníku USB/UART



Obr. 15 Schéma zapojení převodníku USB/UART

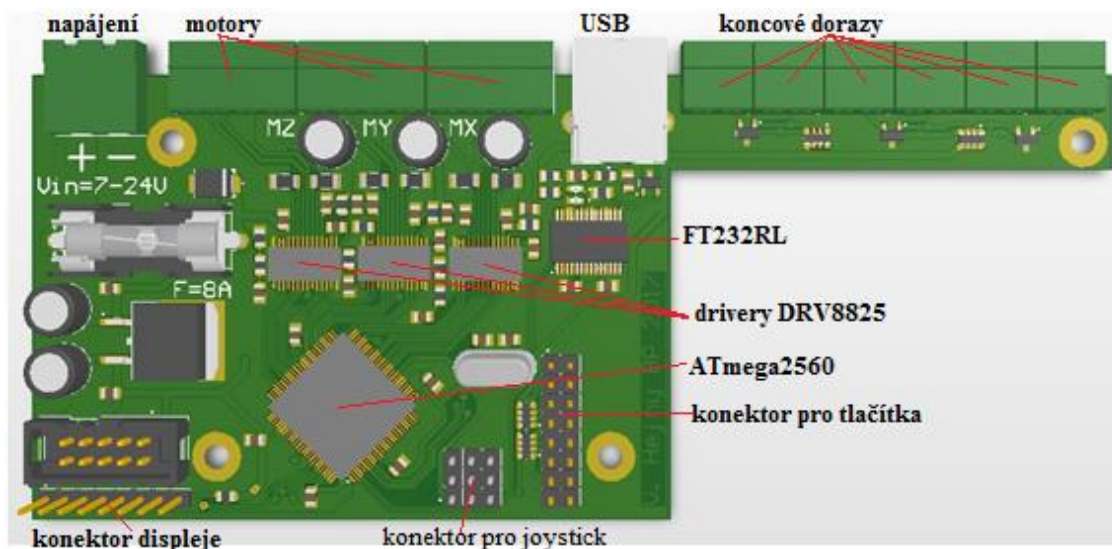
Zapojení obvodu je opět dle doporučení z dokumentace součástky. Vlevo ve schématu na obrázku Obr. 15 můžete vidět USB B konektor a vpravo výstupy TxD a RxD, které jsou připojeny na UART v mikrokontroléru. Zapojení je pro obvod, který není napájen z USB, proto

není +5 V z USB připojeno k napájení, ale jen ke vstupu RESET přes napěťový dělič. Datové signály D- a D+ jsou chráněny přidaným transilem proti přepětí. Dále jsou ve schématu dvě LED diody, které slouží k indikaci přenosu dat. Jedna dioda indikuje přenos z počítače a druhá z mikrokontroléru.

Jako řídicí obvod jsem zvolil mikrokontrolér Microchip Atmel ATMEGA2560, který je taktován krystalem na 16 MHz. Tento jednočip ovládá celou řídicí jednotku. Zajišťuje ovládání motorů, komunikaci s počítačem, obsluhu tlačítek a výpisy na LCD displej.

#### 4.3. Plošný spoj

Pro návrh schémat a plošného spoje jsem využil návrhové prostředí Altium designer, kde bylo nejprve potřeba nakreslit si potřebné součástky, které nebyly v knihovnách součástek. Naštěstí v knihovnách chyběla jen jedna součástka, a to byl driver DRV8825. Po vytvoření součástky DRV8825 bylo možné navrhnout schémata a desku plošného spoje. Když jsem návrh plošného spoje dokončil, vygeneroval jsem gerber soubory, což jsou data pro výrobu plošného spoje, která popisují jednotlivé vrstvy. Plošný spoj jsem nechal vyrobit ve známé pražské firmě PragoBOARD s.r.o.



Obr. 16 3D model osazené desky plošného spoje

Při pohledu na plošný spoj na obrázku Obr. 16 je vlevo nahoře vidět napájecí konektor pro napětí v rozmezí hodnot 7–24 V. Jak již bylo zmíněno, napětí je omezeno na 24 V kvůli driverům krokových motorů. Dle datasheetu by neměly být napájeny vyšším napětím. Na ochranu proti přepětí je použit za pojistkou jednosměrný transil SMBJ24. Při přepólování napájení by mělo dojít k přepálení pojistky. Napájení by mělo být zkratováno transilem za pojistkou. Dále jsou na desce tři konektory pro připojení motorů, USB konektor a 6 vstupů pro tlačítka koncových dorazů. Při stisku tlačítka koncového dorazu dojde v jednočipu k přerušení programu a zastaví se správný motor. Pod programovacím konektorem ISP10 je konektor pro připojení sériového LCD displeje s kontrolérem ST7735. Na desce je dále devítipinový konektor pro připojení tří-osého joysticku. Vybral jsem joystick potenciometrový, a proto na každou osu potřebuje tři piny, a to: 5 V, výstup, GND. Výstupy potenciometru jsou přivedeny na analogové vstupy mikrokontroléru. Poslední zatím nepopsaný konektor slouží k připojení až osmi ovládacích tlačítek. Na řídicí jednotce je použito jen 6 z nich.

#### 4.4. Zvolený způsob řízení

Jako ovládací prvek mé řídicí jednotky jsem zvolil 8bitový mikrokontrolér AVR ATmega2560 od Microchipu. Tento chip jsem zvolil proto, že má dostatečný počet vstupů a výstupů a také dostatek 16bitových čítačů. Během programování a testování řídicí jednotky jsem zjistil, že to nebyla nejlepší volba. Mikrokontroléry AVR nemají možnost nastavení priorit přerušování a jsou také pomalé. Taktovatelné jen do 16 MHz. Proto si myslím, že jsem měl zvolit nějaký jiný modernější 32bitový ARM mikrokontrolér.

Pro komunikaci s počítačem přes sběrnici USB jsem zvolil převodník USB/UART FT232RL od firmy FTDI. Tento převodník pracuje jako virtuální sériová linka. Počítač jej vidí jako port COMx, kterému je potřeba pro můj případ nastavit rychlost komunikace 19200 baudů za sekundu, dále 8bitový datový rámec, a jeden stopbit. Při této komunikaci není použita žádná parita.

##### 4.4.1. Řízení driveru a výpočet pozic motorů

Pro generování řídicích signálů STEP pro integrované drivery jsou využity tři 16bitové čítače v mikrokontroléru. Pro každý driver je využit jeden čítač, aby se mohly všechny motory pohybovat najednou a nezávisle na sobě – pro motor X je využit timer5, pro motor Y timer4 a pro motor Z timer3. Tyto čítače jsou nastaveny do režimu, kdy čítají od nuly do hodnoty v porovnávacím registru OCRxA, kde x je číslo čítače. Při shodě hodnot registru OCRxA s načítanou hodnotou čítače dojde k vynulování čítané hodnoty a dojde k přerušování programu. To znamená, že se zastaví vykonávání programu. Na zásobník se uloží adresa následující instrukce v paměti, aby program mohl po vykonání přerušování pokračovat tam, kde skončil. Dle vektoru přerušování od periferie se najde správná funkce, která se má vykonat. Funkce pro obsluhu přerušování těmito čítači obstarávají generování impulsů STEP, a to tak, že je tento signál nastaven do log 1 a o pár příkazů později zpět do log 0. Dále tyto funkce počítají počet generovaných pulsů do celočíselné proměnné s délkou 64 bitů. Dle směru se kroky přičítají nebo odečítají. Při počítání kroků bylo také potřeba zohlednit možnost mikrokrokování a pro celočíselné počítání kroků bylo potřeba každý počítaný krok vynásobit 32x a pak ještě vynásobit velikostí kroku. Z toho vyplývá, že pro celý krok se přičte nebo odečte od proměnné 32 a pro mikrokrokování s krokem o velikosti 1/32 se přičítá nebo odečítá 1. Z načítaného počtu pulsů se počítají pozice motorů, které jsou platné jen tehdy, pokud nedošlo ke skluzu motoru a ztrátě kroků. Pro správné pozice je potřeba dodržet, aby motory nebyly přetížené, ale zároveň aby nějakou zátěž měly – jinak při nízkých rychlostech dochází k mechanickým kmitům, při kterých dochází ke ztrátě kroků. Také je potřeba, aby motory akcelerovaly a brzdily plynule, neboť v lepším případě motor ztratí několik kroků a v horším případě se nerozběhne vůbec. Polohy motorů jsou počítány pouze pro lineární posuvy se závitovou tyčí a motorem, který má 200 kroků na otáčku. Typ závitové tyče je možné volit v menu nebo v počítači v rozsahu M2 až M10.

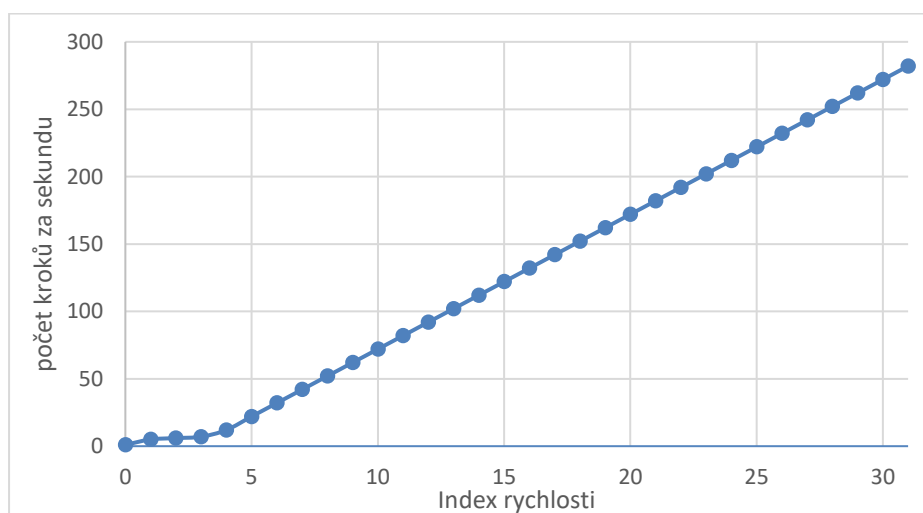
$$pozice = \frac{\text{počet kroků}}{200 \cdot 32} * \text{stoupání závitu} [-, 1/\text{ot}, \text{mm}/\text{ot}, \text{mm}]$$

Vztah 5 Výpočet pozic motorů

Pozice je počítána v milimetrech do neceločíselné proměnné. Ve výpočtu je potřeba vydělit počet kroků hodnotou 32, protože byla touto hodnotou dříve vynásobena z důvodu celočíselného počtu kroků. Dále dělíme hodnotou 200, která odpovídá počtu celých kroků na jednu otáčku motoru. Na konec se tato část vzorce ještě vynásobí stoupáním závitové tyče,

protože různé závity mají různé stoupání, například M2 má stoupání 0,4 mm na otáčku a M6 1 mm na otáčku.

Při nastavené velikosti kroku se motor může pohybovat jednou ze třiceti dvou možných rychlostí, které jsou dány hodnotami v rozběhové tabulce. Rozběhová tabulka slouží pro správnou akceleraci motorů, ke které je potřebná postupná akcelerace na požadovanou rychlost. Tabulka také slouží pro zrychlení programu tím, že není potřeba neustále počítat hodnoty v ní obsažené. Hodnoty z tabulky se dle indexu, kde 0 je nejpomalejší a 31 nejrychlejší, vkládají přímo do porovnávacího registru OCRxA, čímž se ovlivňuje frekvence generovaných pulsů STEP signálů. Ke změně frekvence dochází tak, že čítač stále čítá stejnou rychlostí, ale do jiného čísla, kde se zase vynuluje.



Obr. 17 Graf rozběhové tabulky

Z grafu na obrázku Obr. 17 je vidět, že v nízkých rychlostech motory akcelerují mnohem pomaleji než ve vyšších rychlostech. To je z důvodu, aby bylo možné přesné řízení s joystickem, protože k výpisu poloh na display nedochází s každou změnou pozic, ale jen v určitých okamžicích. Dále motory akcelerují lineárně, aby nedocházelo ke ztrátám kroků.

Pro řízení motorů je přímo na řídicí jednotce umístěn tří-osý joystick. Každá osa joysticku slouží k ovládní jednoho z motorů. U použitého joysticku jako senzory polohy slouží potenciometry. Potenciometry jsou napájeny stejným napětím, kterým je napájena logika a napěťová reference analogově digitálního převodníku, a to 5 V proti zemi. Výstupem joysticku jsou tři analogové signály, u kterých je napětí úměrné úhlu vychýlení řídicí páky. Pokud je páka v neutrální pozici na výstupních signálech, je napětí zhruba 2,5 V. Kvůli vůli páky v neutrální pozici bylo nutné v programu nastavit okolí neutrální polohy tak, aby se motory nerozbíhaly samovolně. Pro měření napětí výstupních analogových signálů joysticku bylo potřeba využít analogově digitální převodník v mikrokontroléru. Tyto signály jsou přivedeny na vstupy ADC0 až ADC2. Při pokusech s AD převodníkem jsem došel k závěru, že obsluhu joysticku je nutné provádět v hlavní smyčce programu, protože při snaze o využití přerušeni docházelo ke špatnému generování STEP signálů pro motory, a to z toho důvodu, že tyto mikrokontroléry nemají možnost nastavení priorit přerušeni. To znamená, že další přerušeni je zakázáno, dokud se nevykoná funkce předešlého přerušeni. Proto je přerušeni využíváno jen tam, kde je to nutné, a to u čítačů, koncových dorazů a UARTu.

#### 4.4.2. Řízení motorů joystickem

Pro ovládání analogově digitálního převodníku existuje několik řídicích registrů. Před zapnutím převodníku je nutné nastavit napěťovou referenci, jež odpovídá maximální hodnotě rozsahu 1023. Je to maximální hodnota, protože převodník je 10bitový a na deseti bitech je možný počet kombinací  $2^{10} = 1024$ . Při této aplikaci analogově digitálního převodníku jsem zvolil externí napěťovou referenci z pinu Aref vůči zemi. Na pin Aref jsem přivedl napětí 5 V, tedy stejné jako u joysticku, aby bylo možné využít celý rozsah od 0 do 1023. Napěťová reference se nastavuje v registru ADMUX. Kombinací bitů REFS1 a REFS0 se nastavují různé možnosti referencí, ať už externí, nebo interní 2,56 V nebo 1,1 V. Pro nastavení externí reference na pinu Aref vůči zemi je nutné nastavit bity REFS1 a REFS0 do log0. Po nastavení reference je možné zapnout převodník. Převodník se zapíná tak, že v registru ADCSRA se nastaví bit ADEN (ADC enable) do stavu log1. Jelikož používám tři analogové vstupy a v mikrokontroléru je jen jeden analogově digitální převodník, je nutné multiplexovat vstupy. Před každým převodem je potřeba zvolit vstup, na kterém chci měřit napětí. To se dělá tak, že v registru ADMUX existují bity MUX4 až MUX0, jejichž kombinací se volí jeden z analogových vstupů. Vzhledem k tomu, že jsem použil jen tři tyto vstupy, stačí mi pro přepínání vstupu jen kombinace bitů MUX1 a MUX0. Po nastavení vstupu je možné spustit převod. Převod se spouští nastavením bitu ADSC (ADC start conversion) v registru ADCSRA. Po převodu se tento bit sám nastaví do úrovně log0 a bit ADIF (ADC conversion complete interrupt flag) se nastaví do log1. Podle toho se dá poznat, že je převod hotový. Po dokončení převodu je možné vyčíst naměřenou hodnotu z 10bitového registru ADC. Naměřená hodnota je číslo v rozsahu 0–1023, kde 512±12 je neutrální poloha. Takže motory jsou zastaveny v intervalu hodnot 500–524. V intervalech naměřených hodnot 0–500 a 524–1023 se zapínají čítače generující signály STEP. Směr motorů je volen podle toho, o jaký interval se jedná. Z naměřených hodnot se počítají celočíselné indexy, které slouží pro adresaci prvku v rozběhové tabulce, a pak do nastavené rychlosti postupně akcelerují motory. Index rychlosti se počítá v každém z intervalů jinak.

$$index = \frac{512-ADC}{16} [-]$$

Vztah 6 Výpočet indexu rychlosti v intervalu naměřených hodnot 0–500

$$index = \frac{ADC-524}{16} [-]$$

Vztah 7 Výpočet indexu rychlosti v intervalu naměřených hodnot 524–1023



#### 4.4.3. Popis nastavení a ovládání řídicí jednotky

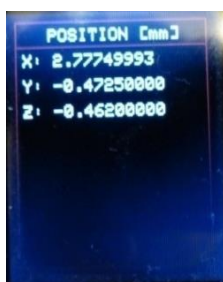
Pro ovládání řídicí jednotky jsem na panel zařízení namontoval šest tlačítek. Tato tlačítka slouží pro vstup do menu a pro všechna nastavení. Rozložení tlačítek je vyobrazeno na obrázku Obr. 18.



Obr. 18 Tlačítka řídicí jednotky

Na řídicí jednotce jsou nejdůležitější tlačítka Enter a Back, která slouží ke vstupu do menu a k potvrzení vybraných prvků. Tlačítko Back slouží také pro návrat zpět na výpis pozic jednotlivých motorů. Dále jsou zde čtyři tlačítka sloužící pro pohyb v menu a pro nastavení velikostí kroků nebo typu posuvu.

Po připojení dostatečného napájení dojde k zapnutí řídicí jednotky. Nejprve se inicializuje řídicí jednotka a displej, a pak se na displeji zobrazí základní okno na obrázku Obr. 19, na kterém se vypisují pozice motorů. V okamžiku zapnutí jsou pozice posuvů nulové. Pro jejich změnu je nutné použít joystick nebo počítač.



Obr. 19 Okno pozic motorů

Každé zobrazované okno má svůj nadpis. V tomto případě se jedná o okno s nadpisem pozice, kde se vypisují za znaky X, Y a Z pozice jednotlivých posuvů. Zobrazované pozice jsou počítány v milimetrech.

Pro nastavení řídicí jednotky bylo nutné naprogramovat menu, kde by bylo možné nastavovat vše potřebné.



Obr. 20 Okno menu

V okně menu na obrázku Obr. 20 je možné vybírat prvky pomocí tlačítek, která jsou označena šípkami nahoru a dolů. V menu je možné volit mezi nastavením velikostí kroků, typem závitů posuvů a také je tu prvek sloužící pro vynulování pozic posuvů.

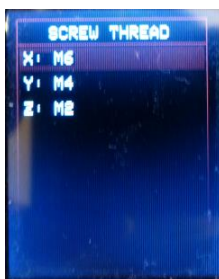
Pokud v menu zvolíme prvek pro nastavení velikostí kroků a potvrdíme jej tlačítkem Enter, zobrazí se nám okno pro nastavení velikostí kroků.



Obr. 21 Okno pro nastavení velikostí kroků

V okně na obrázku Obr. 21 pro výběr motoru, u kterého chceme nastavit velikost kroku, slouží šipky nahoru a dolů. Po výběru motoru můžeme nastavit režim mikrokroku pomocí šipek do stran, kdy šipka doleva zvětšuje krok a šipka doprava ho zmenšuje. Toto nastavení se nepotvrzuje Enterem. Pro návrat zpět do menu je nutné použít tlačítko Back, kdy dojde k uložení nastavených hodnot do paměti EEPROM a nastavené hodnoty se odešlou do počítače.

Posledním oknem je okno pro nastavení typu závitů u jednotlivých posuvů. Do tohoto okna se dostaneme obdobným způsobem jako do nastavení mikrokrokování. Jen zvolíme správný prvek v menu a potvrdíme stiskem tlačítka Enter.



Obr. 22 Okno pro nastavení typů závitů posuvů

V okně na obrázku Obr. 22 je nastavení obdobné jako v případě mikrokrokování s tím rozdílem, že nastavujeme jiné hodnoty. Opět k návratu do menu slouží tlačítko Back, které uloží nastavené hodnoty do paměti EEPROM a odešle je do počítače.

#### 4.4.4. Řízení motorů z počítače

I u tohoto způsobu řízení jsou generovány signály STEP stejným způsobem jako u předchozího způsobu řízení pomocí joysticku, a to 16bitovými čítači. Na rozdíl od řízení joystickem není cílem tohoto řízení nastavovat jen rychlosti generování STEP signálů, ale pootočit motorem o určitý počet kroků do pozice, kterou zadal uživatel.

Řídicí jednotka komunikuje s počítačem přes převodník USB/UART FT232RL. Tento převodník se v počítači obsluhuje jako běžná sériová linka RS232, u které je možné nastavit rychlost komunikace (baud rate), použití parity nebo počet stop bitů.



Při programování řídicí jednotky jsem nejprve musel zvolit, jakým způsobem bude jednotka komunikovat. Zvolil jsem tedy rychlost komunikace 19200 baudů za sekundu bez parity a s jedním stopbitem. Dle toho jsem nastavil příslušné bity kontrolních registrů UARTu.

Z důvodu, že je možné mezi řídicí jednotkou a počítačem přenášet pouze znaky a znakové řetězce, musel jsem si zvolit několik textových příkazů, které mají určitý tvar a ukončovací znak, díky kterému poznám konec příkazu a nemusím odečítat další znaky.

#### 4.4.4.1. Textové příkazy

Přijaté znaky UARTem se ukládají jako text do proměnné typu řetězce znaků, dokud přijatým znakem není “;”, který označuje konec příkazu, a poté se volá funkce, která určí, o jaký příkaz jde a vykoná jej. Pro ovládání jednotky jsem použil příkazy, které slouží k pootočení jednotlivých motorů o daný počet kroků, k nastavení velikosti kroku a typu závitu posuvu. Dále jsem zvolil příkazy pro nastavení směru otáčení, vynulování pozic a příkaz k získání nastavení řídicí jednotky.

Příkaz pro nastavení směru otáčení má zápis ve tvaru “Dxxx;”, kde první znak D určuje, že jde o příkaz pro nastavení směru, a další tři znaky xxx představují znaménko + nebo – podle toho, o jaký směr jde. První z trojice těchto znaků slouží k nastavení směru otáčení motoru X, prostřední popisuje směr motoru Y a poslední určuje směr motoru Z.

Příkazy, které určují, jaký motor se pootočí o kolik kroků, začínají znakem X, Y nebo Z. Tyto znaky určují, o jaký motor jde. První znak je následován patnácti znaky, které je potřeba převést na číslo, a toto číslo pak určuje, o kolik kroků se motor pootočí. Například příkaz pro pootočení motoru Y v režimu celého kroku o 500 kroků má takovýto zápis: “Y000000000016000;”. Tato hodnota se odesílá vynásobená číslem 32 a velikostí kroku z toho důvodu, že v mikrokontroléru se počítají kroky celočíselně. Proto v režimu s velikostí kroku 1/32 pro posuv o 500 mikrokroků má příkaz zápis ve tvaru “Y000000000000500;”.

Další příkazy, které slouží k nastavení velikosti kroku a typu závitu, mají tento zápis: “Mxxx;”, kde M znamená, že jde o příkaz nastavující velikost kroku, a kde znaky xxx jsou tři čísla v rozsahu 0 až 5 pro motory X, Y a Z. Velikost kroku lze určit vztahem:

$$velikost\ kroku = 1/2^x$$

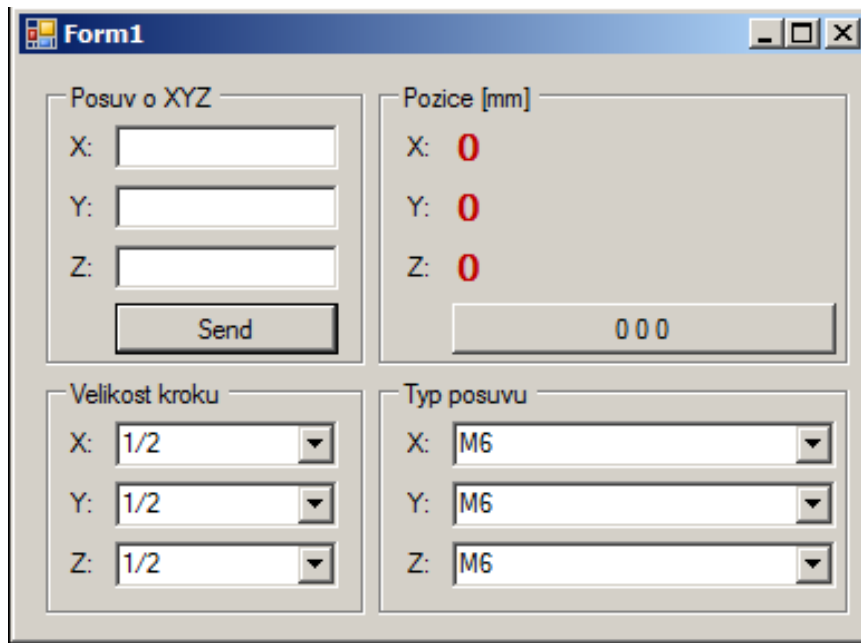
Příkaz “Txxx;” slouží k nastavení typu závitu, kde znaky xxx jsou tři čísla v rozsahu 0 až 6, kdy hodnota 0 odpovídá typu závitu M2 a hodnota 6 odpovídá závitu M10. Hodnoty xxx jsou použity jako indexy do tabulky se stoupáním těchto závitů na otáčku.

Posledními, zatím nepopsanými příkazy, je příkaz “C;” sloužící pro vynulování pozic motorů, a příkaz “G;”, který slouží pro získání hodnot z paměti EEPROM v řídicí jednotce. Jde o hodnoty, které určují nastavení velikosti kroku a typu závitu posuvu.

Podobným způsobem posílá data i řídicí jednotka počítačové aplikaci. I v tomto případě se odesílají data ve formě textu, kde jejich význam určuje první znak. Pozice motorů jsou uvedeny znaky X, Y a Z, které jsou následovány počtem vykonaných kroků. Z tohoto údaje se počítají pozice.

#### 4.4.4.2. Počítačová aplikace pro ovládání řídicí jednotky

Pro naprogramování této aplikace jsem využil programovací jazyk C# a vývojové prostředí Visual studio 2010 vzhledem k jeho jednoduchosti a dostatečnému množství knihoven. Pro komunikaci s řídicí jednotkou jsem využil objekt SerialPort, u kterého jsem nastavil správný způsob komunikace. Nastavený název sériové linky je COM4, a proto je potřeba při připojení řídicí jednotky k počítači zkontrolovat, zda je připojena jako COM4. V případě, že není nastavena správně, je nutné zařízení přejmenovat.



Obr. 23 Počítačová aplikace pro ovládání řídicí jednotky

Při pohledu na obrázek aplikace Obr. 23 je vidět, že je možné nastavit velikosti kroku pouhým výběrem z možností. Stejným způsobem je možné nastavit typ závitů posuvu. Pro rozběh motorů je potřeba nastavit, o kolik milimetrů chceme motory pootočit. Posuv je možné nastavit neceločíselně a se znaménkem, jež označuje směr pootočení. Pro potvrzení zadaných hodnot je nutné kliknout na tlačítko Send. Ze zvolených hodnot se spočítá počet kroků, který je potřeba vykonat v nastaveném režimu, aby motor dojel do nejbližší možné polohy k poloze zvolené. Po výpočtu množství kroků se vygenerují a odešlou textové příkazy. Motory se rozběhnou a během jejich pohybu se zobrazují aktuální pozice motorů. Pro vynulování pozic motorů se používá tlačítko s názvem "0 0 0". Při stisku tohoto tlačítka se odešle do řídicí jednotky příkaz "C;" a vynulují se proměnné, ve kterých je počítán počet kroků. Tím se vynulují i pozice.

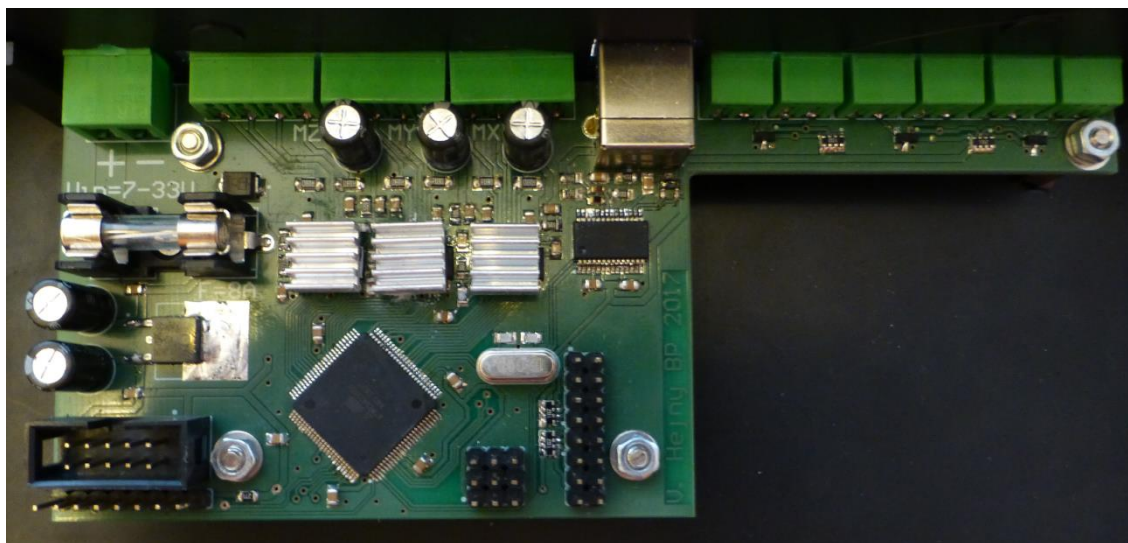
#### 4.5. Zhotovená řídicí jednotka



Obr. 24 Fotografie zapnuté řídicí jednotky s připojeným motorem



Obr. 25 Zadní strana řídicí jednotky



Obr. 26 Osazený plošný spoj

## 5. Závěr

Program řídicí jednotky jsem naprogramoval v programovacím jazyce C ve vývojovém prostředí AtmelStudio, které je navrženo pro psaní kódu pro mikrokontroléry dříve vyráběné firmou Atmel. V tomto prostředí je možné programovat jak AVR, tak i některé ARM procesory v jazycích assembler, C a C++.

Během programování řídicí jednotky jsem narazil na několik problémů. Například jsem dlouho řešil, proč nefungují některá tlačítka. Tato tlačítka byla zapojena zcela stejným způsobem jako ta ostatní, která byla funkční, a i připojení k mikrokontroléru bylo správné. Po dalším bádání, proč tlačítka nefungují, jsem si všiml, že jsou tlačítka připojena k vývodům, na kterých se nachází i JTAG, což jsou vývody pro připojení debuggeru, kterým lze mikrokontrolér programovat i testovat chod programu přímo na chipu. Nakonec řešení tohoto problému bylo snadné, kdy stačilo v konfiguračních bytech zakázat JTAG. Poté již všechna tlačítka fungovala, jak měla.

Dále jsem narazil na problém, kdy nastala chyba během programování a následně mikrokontrolér nešel přeprogramovat. Řešením bylo odpájení krystalu z desky plošného spoje a připojení jiného oscilátoru na vstup XTAL1, což umožnilo přeprogramování mikrokontroléru a následné vrácení krystalu zpět do obvodu.

Během pokusů s řízením motorů v režimu celého kroku jsem také vypořizoval, že rotory motorů se bez zátěže při nízkých rychlostech rozkmitaly natolik, že rotory přeskakovaly kroky. Zatížením motoru se tento efekt eliminoval. Další problém se vyskytl ve stejném režimu kroku, ale při vyšších rychlostech, a tak docházelo ke ztrátě kroků. I v tomto případě zatížení motoru tento problém vyřešilo. V ostatních režimech velikosti kroků se tyto problémy neprojeví. A tak jsem usoudil, že režim celého kroku je nepoužitelný, pokud rotor motoru není zatížen.

Bohužel nebylo s tímto mikrokontrolérem možné, aby měly motory stejnou rychlost otáčení v různých režimech mikrokrokování, proto je režim celého kroku nejrychlejší a rychlost se postupně zpomaluje se zmenšováním velikosti kroku. Rychlosti motorů se tedy zmenšují v závislosti na velikosti kroku, protože pro docílení stejných rychlostí by bylo nutné generovat pro každý menší režim kroku dvakrát více impulsů na STEP signálech. Avšak signál STEP jsem zachoval pro všechny režimy krokování stejný, protože při pokusech, aby rychlosti byly stejné, docházelo ke generování až příliš velkého počtu přerušení a při chodu více motorů najednou přestával mikrokontrolér fungovat. Domnívám se, že docházelo k přetečení zásobníku do datové paměti a k následné nesprávné funkci mikrokontroléru. Pokud bych chtěl, aby se motory pohybovaly ve všech režimech stejně rychle, bylo by nutné použít rychlejší mikrokontrolér nebo nějaké programovatelné hradlové pole, buď například větší CPLD, nebo malé FPGA pro generování STEP signálů.

Sestrojená řídicí jednotka je plně funkční. Funkčnost zařízení je předvedena ve videu na příloženém CD.



## 6. Literatura

- [1] BARNETT, R., H. – COX, S. – O’CULL, L. *Embedded C programming and the Atmel AVR*. 2nd edition 2007, Cengage learning, Delmar
- [2] ZÁVORSKÝ, O. *Programujeme AVR v jazyku C*. 2012
- [3] KERNINGHAM, B., W. – RITCHIE, D., M. *Programovací jazyk C*. Brno: Computer Press, a.s., 2006
- [4] ZÁHLAVA V. *Návrh a konstrukce desek plošných spojů*. Praha: BEN, 2011
- [5] Dostupné z: <<https://www.kvd.zcu.cz/cz/materialy/POS/HTML/66/default.htm>>.
- [6] *Lineární krokový motor* [online].  
Dostupné z: <<http://www.pohonnatechnika.cz/skola/motory/linearni-motor>>.
- [7] ŘEZÁČ, K. *Krokové motory* [online]. 10/2002.  
Dostupné z: <<http://robotika.cz/articles/steppers/cs>>.
- [8] *Wikipedia, the free encyclopedia: Stepper motor* [online]. 1/2003.  
Dostupné z: <[https://en.wikipedia.org/wiki/Stepper\\_motor](https://en.wikipedia.org/wiki/Stepper_motor)>.
- [9] *Rotační encoder* [online].  
Dostupné z: <[http://www.hansinteractive.wz.cz/rotacni\\_ekoder.html](http://www.hansinteractive.wz.cz/rotacni_ekoder.html)>.
- [10] *Datasheet driveru DRV8825* [online]. Texas Instruments, 4/2010.  
Dostupné z:  
<<http://www.ti.com/product/DRV8825/datasheet?keyMatch=drv8825&tisearch=Search-EN-Everything>>
- [11] *Datasheet mikrokontroléru AVR ATmega2560* [online]. Microchip Atmel.  
Dostupné z: <[http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf)>
- [12] *Datasheet převodníku FT232Rl* [online]. Future Technology Devices International Ltd.  
Dostupné z: <[http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)>

## 7. Použitý software

*[p1] - Altium designer 2016 – Návrh schémat a plošného spoje*

*[p2] - Atmel Studio 6.1 – Programování mikrokontroléru AVR*

*[p3] - Visual studio 2010 – Programování počítačové aplikace v jazyce C#*

*[p4] - Microsoft Office 2007*

## 8. Seznam příloh

8.1. Pinout Atmega2560

8.2. Kompletní schémata

8.3. Plošný spoj

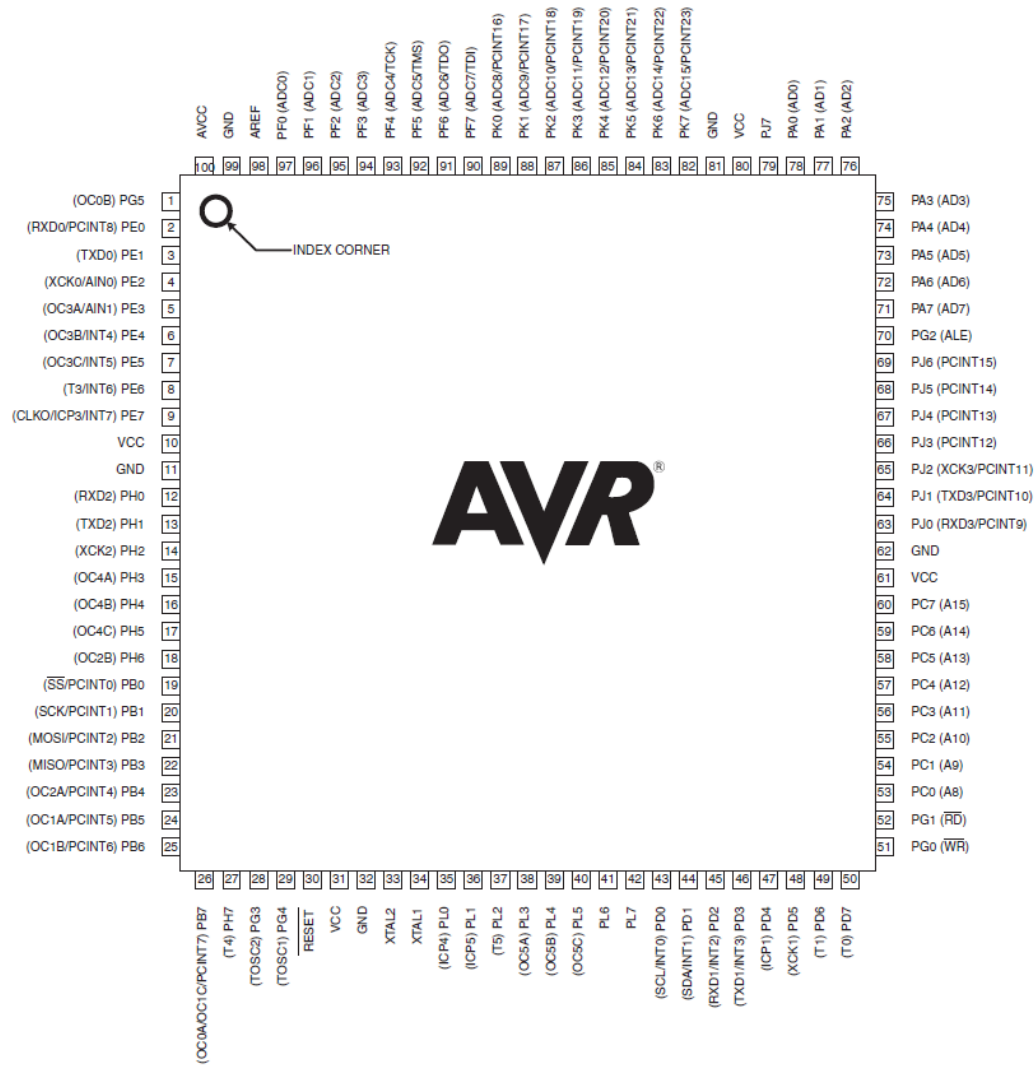
8.4. CD s dokumentací bakalářské práce

Obsah CD

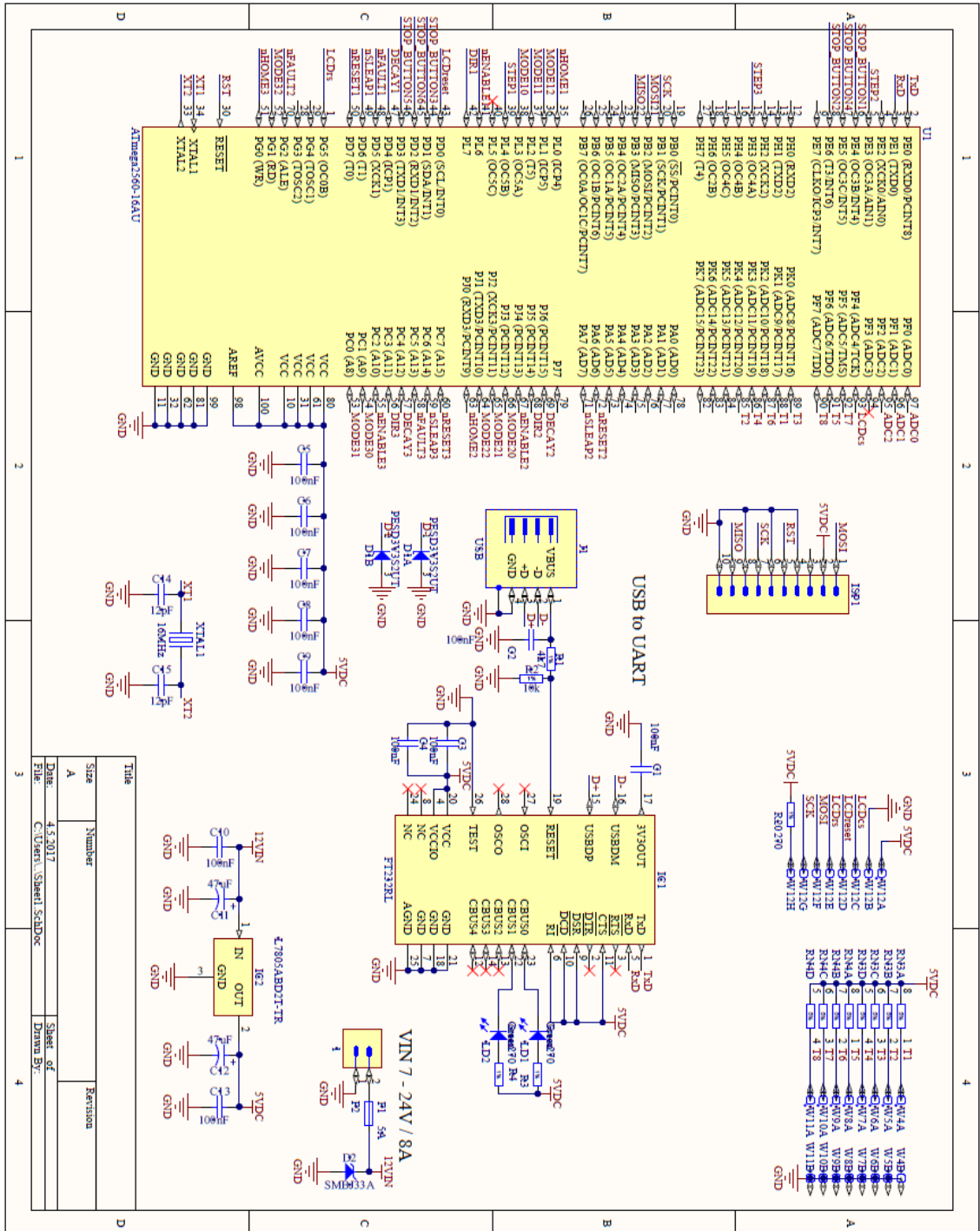
- Text bakalářské práce
- Fotografie řídicí jednotky
- Schémata a DPS řídicí jednotky
- Program řídicí jednotky
- Program počítačové aplikace
- Složka s výrobními daty
  - o Gerber data ve formátu RS274-X
  - o Seznam součástek BOM
  - o HEX program řídicí jednotky



## 8.1. Pinout Atmega2560



## 8.2. Schéma řídicí jednotky – 1. část





## 8.4. Plošný spoj

