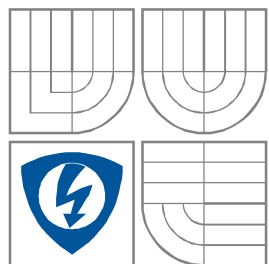


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

# NÁVRH SPECIÁLNÍHO ANTÉNNÍHO ROTÁTORU

DESIGN OF SPECIAL ANTENNA ROTATOR

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

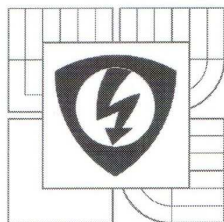
AUTOR PRÁCE  
AUTHOR

Bc. Jan Půhoný

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Jan Kovář

BRNO, 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

## Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Jan Půhoný  
**Ročník:** 2

**ID:** 78406  
**Akademický rok:** 2009/2010

### NÁZEV TÉMATU:

**Návrh speciálního anténního rotátoru**

### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti řízení krokových motorů pro otáčení anténní točny. Navrhněte vhodné budiče a vyberte vhodný mikroprocesor. Navrhněte manuální řízení rotátoru a propojení řídicího mikroprocesoru s PC. Vytvořte program pro mikroprocesor. Dále vytvořte jednoduchý obslužný program pro PC, navrhněte vhodné algoritmy řízení rotátoru pro automatické měření blízkého pole antény. Ověřte funkčnost a zpracujte dokumentaci.

### DOPORUČENÁ LITERATURA:

[1] BURKHARD, M. C pro mikrokontroléry. Praha : BEN - technická literatura, 2003. ISBN: 80-7300-077-6

[2] MÁDR, T. Pracoviště pro automatizované měření antén. Diplomová práce. Brno: FEKT VUT v Brně, 2004

**Termín zadání:** 8.2.2010

**Termín odevzdání:** 21.5.2010

**Vedoucí práce:** Ing. Jan Kovář

**prof. Dr. Ing. Zbyněk Raida**  
Předseda oborové rady

### UPOZORNĚNÍ:

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

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami:

**1. Pan/paní**

Jméno a příjmení:	Bc. Jan Půhoný
Bytem:	Hemže 33, 56501 Choceň
Narozen/a (datum a místo):	27. května 1986 ve Vysokém Mýtě

(dále jen „autor“)

a

**2. Vysoké učení technické v Brně**

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 53, Brno, 602 00  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika  
(dále jen „nabyvatel“)

**Čl. 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☒ diplomová práce
- ☐ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako .....  
(dále jen VŠKP nebo dílo)

Název VŠKP: Návrh speciálního anténního rotátoru

Vedoucí/ školitel VŠKP: Ing. Jan Kovář

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli:

- |  |                      |
|--|----------------------|
| <input checked="" type="checkbox"/> v tištěné formě      | – počet exemplářů: 2 |
| <input checked="" type="checkbox"/> v elektronické formě | – počet exemplářů: 2 |

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

.....  
NABYVATEL

.....  
AUTOR

## **ABSTRAKT**

Diplomová práce se zabývá návrhem elektroniky pro řízení anténního rotátoru ve dvou rovinách. Byly navrženy a realizovány budiče pro dva krokové motory. Dále byla navržena a realizována řídicí deska, která umožňuje jak ruční ovládání rotátoru, tak po připojení k PC vzdálené ovládání přes USB. Zároveň vznikl software pro řídicí desku rotátoru a aplikace pro PC na jednoduché ovládání.

## **KLÍČOVÁ SLOVA**

Anténní rotátor, krokový motor, budič krokového motoru, řízení krokového motoru z PC prostřednictvím USB, A3986, C8051F320, USB HID

## **ABSTRACT**

This master's thesis deals with design electronics for an antenna rotator in two axis. There are designed and implemented drivers for two stepper motors. Was also designed and implemented control panel that allows manual control the rotator and remote control by PC via USB. At the same time created software for control panel and PC application for easy operation.

## **KEYWORDS**

Antenna rotator, stepper motor, stepper motor driver, stepper motor control from a PC via USB, A3986, C8051F320 USB HID

PŮHONÝ, J. *Návrh speciálního anténního rotátoru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 45s. Vedoucí semestrální práce  
Ing. Jan Kovář.

# Prohlášení

Prohlašuji, že diplomovou práci na téma Návrh speciálního anténního rotátoru jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 21.května 2010

.....  
podpis autora

# Poděkování

Děkuji vedoucímu diplomové práce Ing. Janu Kovářovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji prof. Dr. Ing. Zdeňku Kolkovi za pomoc při problémech s nastavením potřebných knihoven pro implementaci USB HID protokolu a za radu ohledně přepěťové ochrany navrženého budiče.

V Brně dne 21.května 2010

.....  
podpis autora

# Obsah

1 Úvod .....	1
2 Teoretická část .....	2
2.1 Krokové motory.....	2
2.1.1 Princip.....	2
2.1.1.1 Krokové motory s pasivním rotorem.....	2
2.1.1.2 Krokové motory s aktivním rotorem.....	2
2.1.1.3 Krokové motory hybridní.....	3
2.1.2 Řízení krokových motorů.....	3
2.1.2.1 Unipolární řízení.....	3
2.1.2.2 Bipolární řízení.....	3
2.1.2.3 Mikrokrokování.....	3
2.1.3 Použité motory.....	4
2.1.4 Budiče pro řízení krokových motorů.....	4
2.2 Řídicí mikrokontrolér.....	6
2.2.1 Vstupně výstupní porty.....	7
2.2.2 Čítače / Časovače.....	7
2.2.3 USB rozhraní.....	7
2.2.4 USB třída HID.....	8
2.2.4.1 Endpointy HID (EP).....	8
2.2.4.2 Reporty HID.....	9
2.2.4.3 Řízení přenosu.....	9
2.2.4.4 Deskriptory HID.....	9
3 Konstrukční část .....	10
3.1 Budič krokových motorů.....	10
3.1.1 Popis zapojení.....	10
3.1.2 Plošný spoj.....	13
3.1.3 Ovládání budiče.....	13
3.2 Řídicí deska.....	17
3.2.1 Popis zapojení.....	17
3.2.1.1 Zdroj.....	17
3.2.1.2 Obvody mikrokontroléru.....	17
3.2.1.3 Připojení budičů.....	18
3.2.1.4 Vstupy od čidel nulové polohy.....	18
3.2.2 Plošný spoj a mechanická konstrukce.....	18
3.3 Napájecí zdroj.....	25
3.3.1 Popis zapojení.....	25
3.3.2 Mechanická konstrukce.....	26
3.4 Propojení dílčích částí.....	28
3.4.1 Přivedení napájecího napětí.....	28
3.4.2 Datové propojení.....	28
4 Programové vybavení .....	29
4.1 Software mikrokontroléru řídicí jednotky.....	29
4.1.1 Ovládání budičů krokových motorů.....	30
4.1.1.1 step_H_R();.....	30
4.1.1.2 step_H_L();.....	30
4.1.1.3 step_V_R();.....	30
4.1.1.4 step_V_L(); .....	31



4.1.1.5 ztrata();.....	32
4.1.1.6 Stop();.....	32
4.1.2 Obsluha klávesnice.....	32
4.1.3 Obsluha LCD displeje.....	32
4.1.4 Obsluha USB.....	34
4.1.5 Časovač pro realizaci jednoduchého multitaskingu a přepínání úloh.....	36
4.2 Software pro PC.....	38
4.2.1 Enumerace a testování připojení HID zařízení.....	38
4.2.2 Zpracování přijatých paketů.....	38
4.2.3 Odesílání paketů řídicí desce.....	40
5 Závěr .....	42
6 Použitá literatura .....	43
7 Přílohy .....	44
7.1 Elektronická příloha.....	45

## Seznam obrázků

Obr. 1: Krokový motor s pasivním rotorem [2].....	2
Obr. 2: Průběhy generované A3986 (mikrokrokování).....	3
Obr. 3: Průběhy na řídicích pinech A3986.....	4
Obr. 4: Typická aplikace A3986 [5].....	5
Obr. 5: IRF540.....	5
Obr. 6: Blokové schéma A3986 [5].....	6
Obr. 7: Periferie C8051F320 .....	6
Obr. 8: USB rozhraní použitého mikrokontroléru [3].....	7
Obr. 9: Registry pro využití USB [3].....	8
Obr. 10: Schéma zapojení budiče krokového motoru.....	14
Obr. 11: Plošný spoj budiče krokového motoru – TOP.....	15
Obr. 12: Plošný spoj budiče krokového motoru - BOTTOM.....	15
Obr. 13: Osazení plošného spoje budiče motoru.....	15
Obr. 14: Osazená deska budiče v kovové skřínce.....	16
Obr. 15: Připojení debug adaptéru Silicon Laboratories k řídicí desce.....	17
Obr. 16: Optické čidlo nulové polohy.....	18
Obr. 17: Schéma zapojení řídicí desky.....	19
Obr. 18: Návrh plošného spoje řídicí desky - TOP.....	20
Obr. 19: Návrh plošného spoje řídicí desky - BOTTOM.....	21
Obr. 20: Řídicí deska - osazovací plán - TOP.....	21
Obr. 21: Řídicí deska - osazovací plán BOTTOM.....	22
Obr. 22: Osazená řídicí deska - TOP.....	24
Obr. 23: Osazená řídicí deska - BOTTOM.....	24
Obr. 24: Řídicí deska vestavěná v přístrojové skřínce Z 91.....	25
Obr. 25: Schéma zapojení napájecího zdroje.....	26
Obr. 26: Propojení sestavy elektroniky rotátoru.....	28
Obr. 27: Vývojový diagram - software pro řídicí desku.....	29
Obr. 28: Vývojový diagram - aplikace pro ovládání rotátoru z PC.....	40
Obr. 29: Vzhled ovládacího programu.....	41

## Seznam tabulek

Tabulka 1: Technické parametry motoru 86HS45.....	4
Tabulka 2: Technické parametry motoru 57HS09.....	4
Tabulka 3: Popis konektoru K1.....	11
Tabulka 4: Popis konektoru K2.....	12
Tabulka 5: Připojení motoru ke konektoru K3.....	12
Tabulka 6: Nastavení rozlišení mikrokrokování.....	12
Tabulka 7: Nastavení rychlosti odezvy na snímaný proud motorem.....	13
Tabulka 8: Seznam součástek pro budič krokového motoru.....	16
Tabulka 9: Význam pinů JP3 a JP4 (jumper zkratován = log1, jumper rozpojen = log0).....	20
Tabulka 10: Význam pinů JP1 a JP2.....	20
Tabulka 11: Seznam součástek pro řídicí desku.....	23
Tabulka 12: Seznam součástek - napájecí zdroj.....	27
Tabulka 13: Vnitřní uspořádání zdroje.....	27
Tabulka 14: Hotový zdroj spolu s budičem, krokovým motorem a řídicí jednotkou.....	27
Tabulka 15: Využití jednotlivých prvků polí na straně PC a $\mu C$ .....	41

## Seznam zkratk

DPS	Deska Plošného Spoje
EP	End Point
HID	Human Interface Device
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
OS	Operační Systém
PC	Osobní počítač
SFR	Special Function Register
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
μC	Mikrokontrolér

## Seznam příloh

Příloha 1	Elektronická příloha práce	CD vloženo v zadních deskách
-----------	----------------------------	------------------------------

# 1 Úvod

Anténní rotátory se používají v aplikacích, kde je třeba měnit úhel natočení antény. V běžných podmínkách se s nimi můžeme setkat u přijímacích antén, kdy uživatel s potřebou příjmu z různých směrů díky rotátoru nepotřebuje více antén. Obvykle se jedná o jednoosé montáže a mění se pouze horizontální úhel natočení antény.

Cílem této diplomové práce bylo vytvořit řídicí elektroniku a software pro dva krokové motory, které ovládají mechaniku rotátoru pro měření směrových vyzařovacích charakteristik antén v bezodrazové komoře. Díky dvěma ovládaným motorům je možno polohu antény měnit jak v horizontální, tak vertikální rovině. Podmínkou byla možnost řízení rotátoru prostřednictvím sběrnice USB, aby mohl být rotátor zařazen do nadřazeného měřicího systému.

Směrová vyzařovací charakteristika patří mezi nejdůležitějších parametry antény. I přes to, že se v posledních letech v této oblasti uplatňuje simulační software, je fyzické měření vyzařovací charakteristiky antén stále používané při určování jejich reálných parametrů. Pro měření vyzařovacích charakteristik je obvykle využívána bezodrazová komora a anténní rotátor na kterém je připevněna měřená anténa připojená k detektoru nebo měřicímu přijímači. Na druhé straně je poté vysílač a pomocná anténa známých parametrů.

V teoretické části diplomové práce jsou nastíněny principy funkce, možnosti řízení krokových motorů a jsou zde popsány zvolené součásti. V konstrukční části je poté představen navržený a realizovaný budič pro řízení krokových motorů použitých v rotátoru včetně napájecího zdroje. Dále je popsán kompletní návrh a realizace řídicí desky pro řízení dvou krokových motorů rotátoru, včetně software pro její mikrokontrolér. Rotátor je možno ovládat kromě ovládacích tlačítek na řídicí desce i pomocí vytvořeného programu na PC.

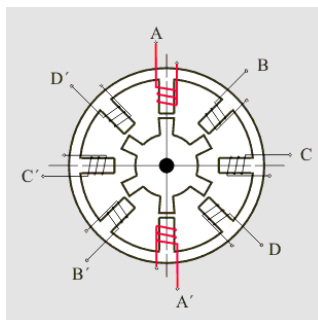
## 2 Teoretická část

### 2.1 Krokové motory

Krokové motory se používají všude tam, kde potřebujeme přesně nastavit polohu pohonu. V navrhovaném rotátoru jsou použity dva motory s příslušnými převody - každý pro nastavení polohy antény v jedné ose. Díky tomu, že točivé magnetické pole v cívkách krokového motoru přímo vytváříme, máme úplnou kontrolu nad provedenými kroky (a tedy i polohou hřídele). Pro snímání nulové referenční polohy je použito optické čidlo. V případě přetížení může dojít ke ztrátě kroku - čidlo nulové polohy slouží i k opravě informace o poloze a zamezení havarijních stavů.

#### 2.1.1 Princip

Základní princip krokového motoru je jednoduchý. Proud procházející cívkou statoru vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru. Vhodným připojováním cívek dosáhneme vytvoření rotujícího magnetického pole, které otáčí rotorem.



Obr. 1: Krokový motor s pasivním rotorem [2]

Podle požadovaného kroutícího momentu, přesnosti nastavení polohy a přípustného odběru volíme některou z variant řízení. Všechny možnosti jsou probrány v další části textu. Kvůli přechodovým magnetickým jevům je omezena rychlost otáčení motoru a to na několik stovek kroků za sekundu (závisí na typu motoru a zatížení). Při překročení této maximální rychlosti (nebo při příliš velké zátěži) motory začínají ztrácet kroky. [1]

##### 2.1.1.1 Krokové motory s pasivním rotorem

Jsou to motory s vyjádřenými póly na statoru i rotoru, využívající výrazně rozdílné magnetické reaktance (vodivosti) v příčné i podélné ose. Podmínkou funkce je rozdílný počet pólů (zubů) na statoru a rotoru.

##### 2.1.1.2 Krokové motory s aktivním rotorem

Jejich rotor je tvořen permanentním magnetem (odtud i název aktivní). Podle uspořádání pólů magnetu odlišujeme dvě skupiny těchto motorů - s radiálně polarizovaným nebo s axiálně polarizovaným permanentním magnetem. Mají rozdílný počet pólů na statoru i rotoru, přičemž rotorové póly jsou tvořeny permanentními magnety.

### 2.1.1.3 Krokové motory hybridní

Slučují konstrukční principy obou předchozích typů. [7]

## 2.1.2 Řízení krokových motorů

### 2.1.2.1 Unipolární řízení

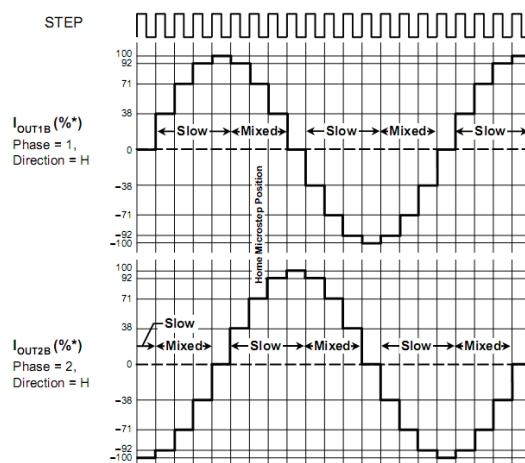
Při unipolárním řízení prochází proud v jednom okamžiku právě jednou cívkou. Motor s tímto buzením má nejmenší odběr, ale také poskytuje nejmenší kroutící moment. Výhodou tohoto řešení je jednoduché zapojení řídicí elektroniky – prakticky stačí jeden tranzistor na každou cívku.

### 2.1.2.2 Bipolární řízení

Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami. Ty jsou zapojeny tak, že mají navzájem opačně orientované magnetické pole. Motor v tomto režimu poskytuje větší kroutící moment, ovšem za cenu vyšší spotřeby. Pro řízení jsou zapotřebí 2 H-můstky: pro každou větev jeden. Zapojení je tedy poněkud složitější. [1] Toto zapojení (bipolární sériové) bylo zvoleno pro konstrukci rotátoru.

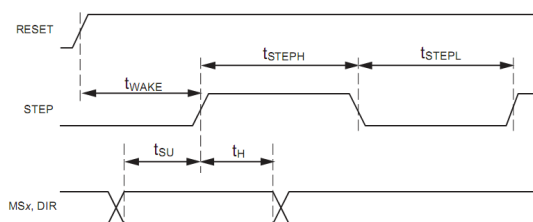
### 2.1.2.3 Mikrokrokování

Plná délka kroku krokového motoru může být rozdělena do menších inkrementů pohybu rotoru, tento režim se nazývá mikrokrokování, respektive „drobení“ kroku (mikrostepping, mini-steps).



Obr. 2: Průběhy generované A3986  
(mikrokrokování)

Budeme-li v jedné fázi postupně snižovat hodnotu budícího proudu a v sousední fázi hodnotu proudu zvyšovat, bude se výsledný vektor magnetické indukce pohybovat mezi krajními polohami, danými vybuzením jednotlivých fází. Pro dvoufázový hybridní motor je pro jeden plný krok motoru požadována rotace magnetického pole o  $90^\circ$  elektrických. Tento princip umožňuje jak přesnější nastavení polohy, tak „hladší“ chod motoru. Níže navržený budič mikrokrokování podporuje až do hodnoty 1/16 kroku.



Obr. 3: Průběhy na řídicích pinech A3986

### 2.1.3 Použité motory

Pro konstrukci rotátoru byly vzhledem k dobré dostupnosti vybrány motory typu 86HS45 a 57HS09. Jejich parametry jsou uvedeny v následujících tabulkách.

86HS45 - 8 vodičů, základní úhel 1,8°, hřídel 12.7x20mm, rozměry 86x86x80mm

Typ	Zapojení	Moment (Nm)	Proud (A)	Odpor vinutí (Ohm)	Indukčnost (mH)	Moment setrvačnosti	Hmotnost Kg
<b>86HS45</b>	Bipolární Paralelní	4.5	6.0	0.375±10%	3.4±20%	1400	2.3
	Bipolární Sériové	4.5	3.0	1.5±10%	14.0±20%		
	Unipolární	3.2	4.2	0.75 ±10%	3.4±20%		

Tabulka 1: Technické parametry motoru 86HS45

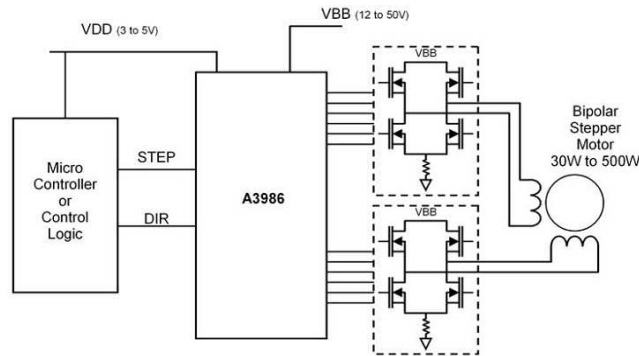
57HS09 - 8 vodičů, základní úhel 1,8°, hřídel 6,35x20mm, rozměry 57x57x56mm

Typ	Zapojení	Moment (Nm)	Proud (A)	Odpor vinutí (Ohm)	Indukčnost (mH)	Moment setrvačnosti g.cm <sup>2</sup>	Hmotnost Kg
<b>57HS09</b>	Bipolární Paralelní	1.3	4.0	0.4±10%	1.2±20%	260	0.6
	Bipolární Sériové	1.3	2.0	1.6±10%	4.8±20%		
	Unipolární	0.9	2.8	0.8±10%	1.2±20%		

Tabulka 2: Technické parametry motoru 57HS09

### 2.1.4 Budiče pro řízení krokových motorů



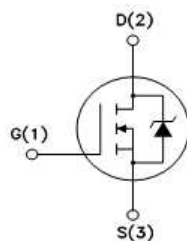


Obr. 4: Typická aplikace A3986 [5]

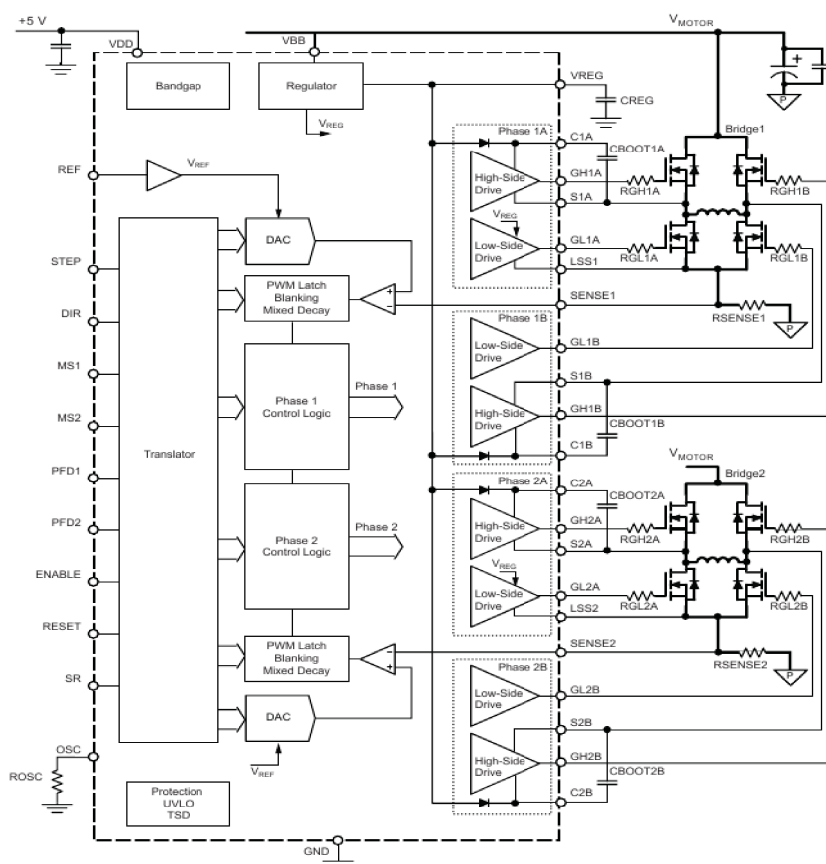
Krokové motory je třeba řídit prostřednictvím budiče. Vzhledem k požadovanému proudu motory byl vybrán budič A3986 společnosti Allegro MicroSystems Inc s externě zapojenými můstky z výkonových MOSFET tranzistorů.

Jedná se o dvojitý driver pro H - můstky s možností mikrokrokování, integrovaným vyhodnocením proudu cívky a paralelním rozhraním (obr. 3) pro připojení k mikrokontroléru. Jako externí součásti jsou požadovány blokovací kondenzátory, 8 x MOSFET tranzistor (pro oba H můstky), rezistory pro snímání proudu cívky a kondenzátory bootstrap. Typické zapojení a blokové schéma obvodu je uvedeno na obr. 4 a 6.

Jako tranzistory pro H můstek byly navrženy IRF540 struktury MOSFET s integrovanou shottky diodou [6] – obr. 5. Vyrábějí se v pouzdru TO220 a jsou dobře dostupné. Maximální spínaný proud je 22A a odpor v sepnutém stavu mají menší než 0,077Ω.  $V_{DS}$  může být až 100V což pro tuto aplikaci vyhovuje.



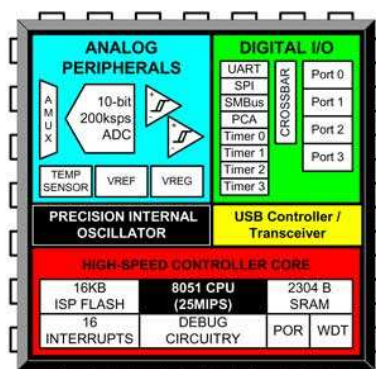
Obr. 5: IRF540



Obr. 6: Blokové schéma A3986 [5]

## 2.2 Řídicí mikrokontrolér

Jako řídicí mikrokontrolér byl s ohledem na požadavek připojení k USB zvolen typ C8051F320 [3] v pouzdře LQFP-32. Rodina mikrokontrolérů C8051F320/1 používá jádro CIP-51 firmy Cygnal, které je plně slučitelné s instrukční sadou MCS-51 a pro vývoj aplikací tak mohou být použity standardní 803x/805x kompilátory. Jádro CIP-51 nabízí stejné periferie jako běžná 8052, včetně čtveřice 16bitových čítačů/časovačů, plně duplexního UART, SPI portu, 2304 byte RAM a 128Byte adresového prostoru pro SFR.



Obr. 7: Periferie C8051F320

CIP-51 dále nabízí lepší výkon a až 70% instrukcí se provádí v jednom nebo dvou hodinových cyklech. Má celkem 109 instrukcí a s maximálním hodinovým taktem 25MHz dosahuje až 25MIPS.

Mikrokontroléry této rodiny jsou určeny pro napájecí napětí v rozmezí 2,7-3,6V (při použití USB rozhraní 3-3,6V) avšak I/O brány i vstup reset podporují 5V logiku. Obvody C8051F320/1 jsou dodávány ve 32pinových LQFP nebo 28 pinových MLP pouzdrech. [4]

Díky integrovanému napěťovému regulátoru, Watchdog časovači a oscilátoru jsou procesory C8051F320/1 zcela soběstačné. Flash paměť je možné přeprogramovat přímo v aplikaci - dovoluje tedy snadný upgrade firmware. Uživatelský software má kompletní kontrolu nad periferiemi a může je individuálně vypínat pro úsporu napájení.

Mikrokontrolér je navíc vybaven vývojovým rozhraním Cygnal 2Wire (C2), které dovoluje snadné ladění aplikací přímo v obvodu, protože nevyžaduje žádné prostředky procesoru. Tato ladicí logika podporuje inspekci a modifikaci paměti a registrů, nastavování bodů přerušení, krokování a příkazy RUN a HALT. Všechny analogové a digitální periferie jsou během ladění pomocí C2 plně funkční. Oba C2 piny rozhraní mohou být sdíleny s uživatelskými funkcemi. Níže budou podrobněji popsány v tomto projektu využití periferie.

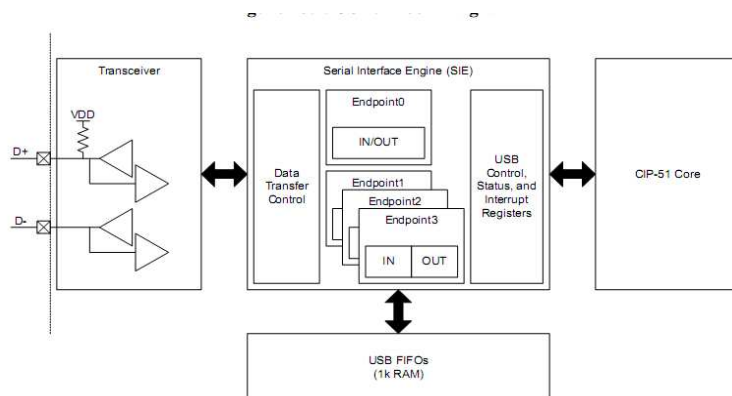
## 2.2.1 Vstupně výstupní porty

C8051F320 disponuje 21 I/O piny, každý vstup může být nastaven jako analogový vstup nebo digitální I/O. Piny nastavené jako digitální I/O mohou být navíc nastaveny jako push-pull nebo open-collector. Rezistory pull-up mohou být pro úsporu energie vypnuty. Zapojení jednotlivých pinů je možno „namapovat“ na jednotlivé porty nebo periferie díky funkci „crossbar“. Lze tak v jisté míře vytvořit ideální rozmístění pinů.

## 2.2.2 Čítače / Časovače

Použitý mikrokontrolér je vybaven čtyřmi čítači / časovači. Obsahuje dva 16-ti bitové čítače / časovače slučitelné se standardní výbavou jádra 8051 a dva „autoreload“ 16-ti bitové čítače / časovače pro použití s AD převodníkem, SMBus a především USB (délka rámce). Tyto časovače mohou být použity k měření časových intervalů, počtu vnějších událostí a vytvářet pravidelné přerušení. V této konstrukci je využit Timer 0 pro generování systémového času a Timer 2 pro obsluhu USB (viz kapitola 4.1.4)

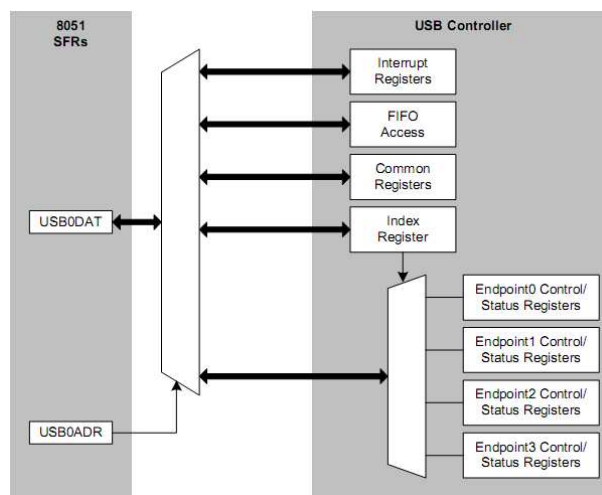
## 2.2.3 USB rozhraní



Obr. 8: USB rozhraní použitého mikrokontroléru [3]

Klíčovou součástí vybraného mikrokontroléru je integrované rozhraní USB 2.0. Díky němu je možno vytvořit zařízení třídy HID (*Human Interface Device*) a mikrokontrolér připojit k počítači. Prostřednictvím USB rozhraní se nastavuje poloha rotátoru a je možné ovládání z PC. Popis komunikace přes USB a vytvořený software pro tento mikrokontrolér je podrobně popsán v kapitole 4.1.

K registrům USB rozhraní v mikrokontroléru lze přistupovat přímo. Jedná se o registry USB 0 Data (USB0DAT) a USB 0 Adres (USB0ADDR).



Obr. 9: Registry pro využití USB [3]

## 2.2.4 USB třída HID

Jednou z tříd USB je Human – Interface – Device (HID). Třída definuje další deskriptory, mechanismy komunikace, formát dat a protokoly. Tato třída byla jedna z prvních tříd USB podporovaných v operačních systémech (OS). Je určena především pro připojování periférií jako je klávesnice nebo myš k PC. Na počítačích s OS počínaje Windows 98 mohou zařízení standardu HID komunikovat s aplikacemi a samotným OS prostřednictvím ovladače který je standardní součástí OS. Vzhledem k tomu, že třída HID podporuje výměnu dat pro specifické aplikace, představuje pro řadu speciálních zařízení vhodnou cestu pro připojení prostřednictvím USB.

HID se nemusí omezovat pouze na výše uvedené případy. Lze ji použít i pro zařízení která potřebují komunikovat prostřednictvím USB a stačí jim rychlost a požadavky specifikované pro třídu HID. Obecně se jedná o zařízení, která nevyžadují velkou přenosovou kapacitu.

Data která chceme přenášet prostřednictvím USB HID mají pevnou strukturu a délku, pokud ne, je třeba je do předepsaného formátu upravit. Data jsou zasílána jako „reporty“.

### 2.2.4.1 Endpointy HID (EP)

Všechny přenosy probíhají pomocí EP. Každé zařízení třídy HID musí mít nejméně jeden EP IN (v 1.00: EP IN (povinný), od verze 1.10: EP IN (povinný) + EP OUT (volitelný) [11]).

### **2.2.4.2 Reporty HID**

Z požadavku na přerušení EP vyplývá, že každý HID musí mít alespoň jeden vstupní report. Výstupní report je nepovinný.

### **2.2.4.3 Řízení přenosu**

Specifikace HID zde definuje šest specifických zpráv pro řízení. Set Report, Get Report upravují směr přenosu mezi zařízením a HOSTem. Set Idle a Get Idle nastavují a čtou data změněná od posledního přenosu. Set Protocol a Get Protocol umožňují pracovat se zjednodušeným protokolem, např. během bootování, když ještě ovladače HID nejsou na straně HOSTu zavedeny.

### **2.2.4.4 Deskriptory HID**

**Configuration descriptor – musí být definován minimálně jeden**

09bLength Descriptor size in bytes

02bDescriptorTypeDescriptor type (Configuration)

0029wTotalLengthTotal length of this and subordinate descriptors

01bNumInterfacesNumber of interfaces in this configuration

01bConfigurationValueIndex of this configuration

00iConfigurationConfiguration string index

A0bmAttributesAttributes (bus powered, remote wakeup supported)

32bMaxPowerMaximum power consumption (100 mA)

**Device Descriptor**

**Interface descriptor – musí být definován minimálně jeden**

**Endpoint descriptor – EP0 nepotřebuje descriptor**

**String descriptor – nepovinné**

**Interrupt IN Endpoint Descriptor**

**Interrupt OUT Endpoint Descriptor**

**HID Descriptor**

Přesná specifikace struktury deskriptorů je uvedena v [10] a [11].

## 3 Konstrukční část

### 3.1 Budič krokových motorů

#### 3.1.1 Popis zapojení

Pro řízení výkonových tranzistorů, které spínají cívky krokového motoru byl vybrán výše popsáný integrovaný obvod A3986. Schéma zapojení vychází z [5] a je na obr. 10. V zapojení je klíčovým prvkem samotný IO1 – A3986, který se stará o buzení dvou plných mostů z tranzistorů T1 – T8. Kondenzátory C3, C4, C6 a C7 slouží k filtraci napájecího napětí– vzhledem k plánovanému využití dostatečně dimenzovaného zdroje nemusí být osazeny. LED dioda D1 indikuje připojení napájecího napětí pro motor. Propojkou JP1 je možno propojit digitální zem (GND) a zem výkonové části (GNDA).

Rezistor R9 nastavuje frekvenci interního oscilátoru A3986. Při navržené hodnotě 10kΩ je  $f_{osc}$  [5]:

$$f_{osc}[MHz] = 100 / (6 + 1,9 \times R_{osc}[k\Omega]) = 100 / (6 + 1,9 \times 10) = \underline{4 MHz}.$$

*Vztah 1: Výpočet frekvence interního oscilátoru A3986*

Kondenzátory C8, C9, C10 a C11 slouží jako „bootstrap“ pro spínání výkonových tranzistorů. Jsou navrženy keramické s hodnotou 10x až 25x větší, než je kapacita elektrolydy gate použitých výkonových tranzistorů (T1 až T8). Dle [6] je  $C_{gate}$  IRF540 870pF, byly tedy zvoleny kondenzátory s hodnotou 22nF.

Rezistory R6 a R12 slouží pro snímání proudů tekoucích cívkami připojeného krokového motoru. Pro zvolený maximální proud jednou cívkou 3,125A a referenční napětí  $V_{REF} = 1,65V$  ( $3,3V / 2$ ) jsou hodnoty rezistorů R6 a R12 následující [5]:

$$R_{6\wedge 12} = \frac{V_{REF}}{8 \times I_{TRIP MAX}} = \frac{1,65}{8 \times 3,125} = \underline{0,066 \Omega}.$$

*Vztah 2: Výpočet hodnot rezistorů pro snímání proudu*

Maximální referenční napětí pro A3986 je 2.00V. Pokud bude na pin REF připojeno vyšší napětí, bude vnitřně omezeno na 2.00V.

Kondenzátor C1 slouží jako blokovací, C2 poté k blokování vnitřního regulátoru (vývod VREG). Hodnota C2 je vypočtena podle [5]:

$$C_{REG}[nF] > C_{BOOT}[nF] \times 3 \times f_{PWM}[kHz] = 22 \times 3 \times 14 = 924nF \approx \underline{1 \mu F}.$$

*Vztah 3: Výpočet  $C_{reg}$*

Řídicí signály pro A3986 jsou vyvedeny na konektor K2. Funkce jednotlivých vývodů včetně detailního popisu je uvedena v tabulce 4.

<b>Pin K1</b>	<b>Přiřazení</b>
1 + 2	+ 12 – 48V DC
3 + 4	GNDA

*Tabulka 3: Popis konektoru K1*

Napájecí napětí pro samotný A3986 bylo zvoleno 3,3V s ohledem na logické úrovně z řídicí desky a je přiváděno přes konektor K2. Motor a výkonovou část desky lze napájet napětím 12 – 48V přes konektor K1. Samotný motor se připojuje ke konektoru K3 v bipolárním sériovém zapojení. Zapojení motoru 86HS35 (podle barev vyvedených vodičů) ke konektoru K3 je uvedeno v tabulce 5.

*Poznámka 1: Kvůli ochraně řídicího obvodu budiče krokových motorů je důrazně doporučeno připojit transil na silovou napájecí větev každého motoru. Tyto transily jsou osazeny přímo na deskách budičů. Hodnotu napětí, při které transil spíná je nutno zvolit s ohledem na zvolené napájecí napětí. V tomto případě byl zvolen transil 15KE33CA (33V) a napájecí napětí motorů 24V.*

*Poznámka 2: Mezi D a S výkonových tranzistorů se doporučuje připojit keramické kondenzátory 100nF pro potlačení přechodových jevů. Tyto kondenzátory nejsou zakresleny ve schematu, ale jsou osazeny přímo na DPS ze strany bottom co nejbližší vývodům zmíněných tranzistorů.*

<b>Pin K2</b>	<b>Přiřazení</b>	<b>Popis</b>
1	PFD2	Nastavení rychlosti reakce na změny proudu motorem
2	ENABLE	0 – výstupní tranzistory aktivní 1 – výstupní tranzistory deaktivovány
3	PFD1	Nastavení rychlosti reakce na změny proudu motorem
4	GND	Napájení A3986, digitální zem
5	DIR	0 – motor se točí po směru hodinových ručiček 1 – motor se točí proti směru hodinových ručiček
6	SR	Synchronous Rectifier – pokud je SR = 0, je aktivní režim ochrany koncových tranzistorů a nemusí být použity ochranné diody. Když je SR = 1 je nutno použít ochranné diody.
7	MS2	Nastavení mikrokrokování
8	+3,3V	Napájení A3986
9	MS1	Nastavení mikrokrokování
10	+3,3V	Napájení A3986
11	STEP	Krok, reaguje na náběžnou hranu
12	GND	Napájení A3986, digitální zem
13	REF	Vstup referenčního napětí
14	RESET	0 – úsporný režim 1 – normální funkce

Tabulka 4: Popis konektoru K2

<b>Pin K3</b>	<b>Barva vodiče (motor 86HS35)</b>
1 + 5	Bílá
2 + 6	Zelená
3 + 7	Červená
4 + 8	Černá
Oranžová + hnědá – propojit u motoru	
Modrá + žlutá – propojit u motoru	

Tabulka 5: Připojení motoru ke konektoru K3

<b>MS2</b>	<b>MS1</b>	<b>Rozlišení mikrokrokování</b>
0	0	Plný krok
0	1	Poloviční krok
1	0	Čtvrtinový krok
1	1	Šestnáctinový krok

Tabulka 6: Nastavení rozlišení mikrokrokování



<b>PFD2</b>	<b>PFD1</b>	<b>Rychlost odezvy</b>
0	0	0% („Slow decay“)
0	1	8% (7 cyklů)
1	0	26% (25 cyklů)
1	1	100% („Fast decay“)

*Tabulka 7: Nastavení rychlosti odezvy na snímaný proud motorem*

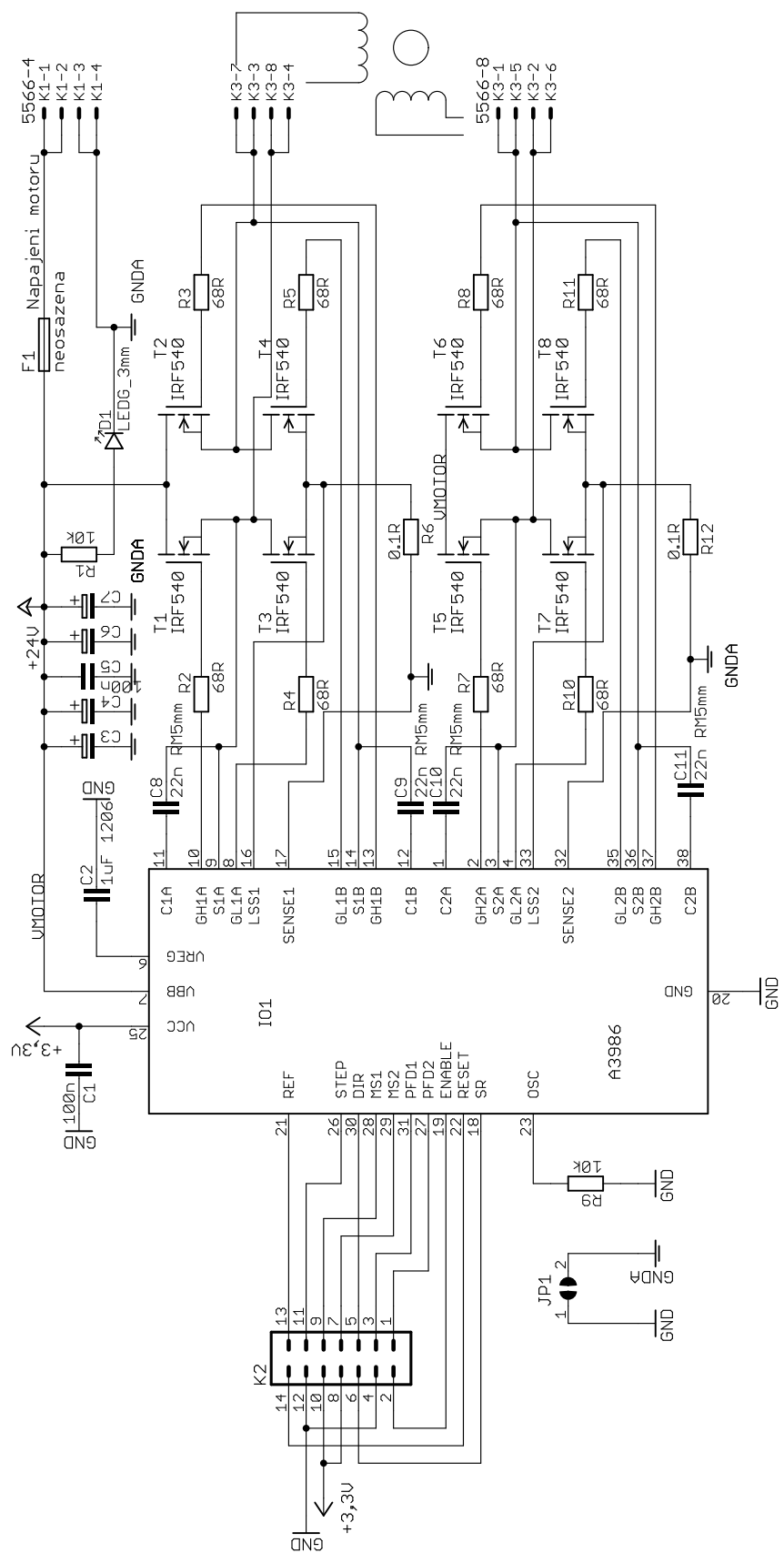
### 3.1.2 Plošný spoj

DPS budiče je navržena jako oboustranná s prokovenými otvory o rozměrech 92 x 65mm. S ohledem na protékající proudy by v případě využití mezních parametrů použitých součástek bylo potřeba silové spoje realizovat na desce se silnějším plátováním (tloušťka mědi 70μm). Pro plánované využití však dimenzování DPS s rezervou postačuje. Návrh plošného spoje je na obr. 11 a 12, osazovací plán poté na obr. 13.

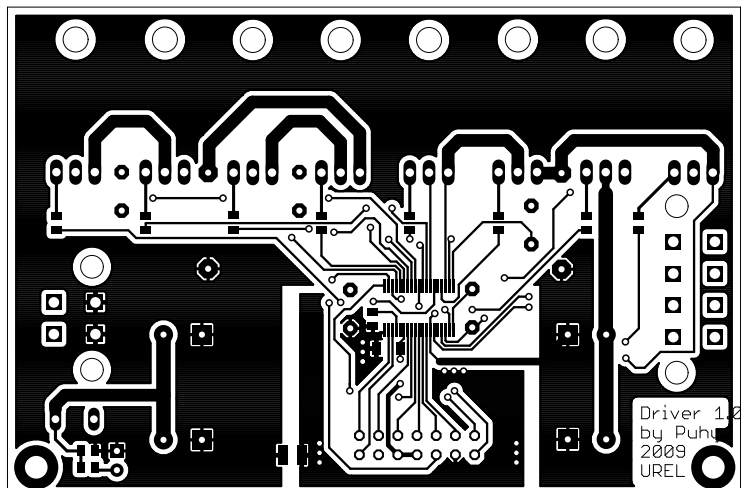
Při osazování DPS je třeba nejprve osadit SMD součástky kvůli prostorové dostupnosti (jako úplně první IO1), nakonec konektory a výkonové tranzistory. Tranzistory jsou umístěny na okraji desky, aby je bylo možno přišroubovat na chladič (musí být montovány přes izolační podložky). V realizovaném vzorku slouží jako chladič kovová skříňka ke které je DPS přišroubována přes „L“ profil a distanční sloupky.

### 3.1.3 Ovládání budiče

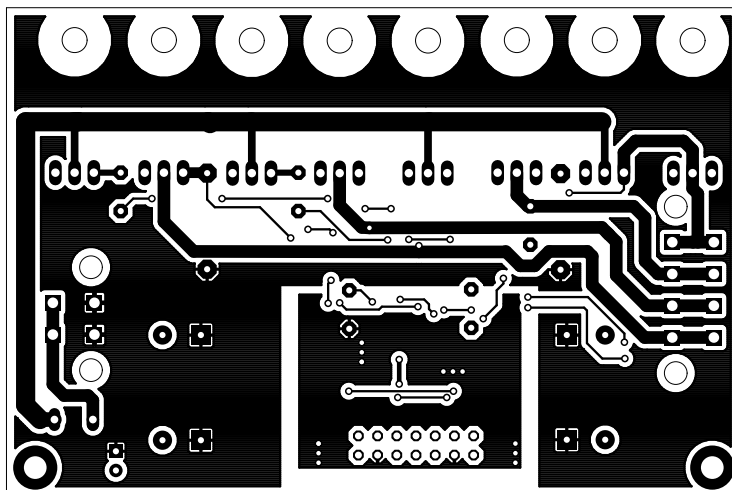
K ovládání slouží signály ENABLE, STEP, DIR, RESET. Vše je podrobně popsáno v [5] a tabulce 4. Pro základní ovládání stačí nastavit RESET do log. 1, ENABLE do log. 0. Náběžnou hranou na STEP se provádí 1 krok. DIR určuje směr otáčení.



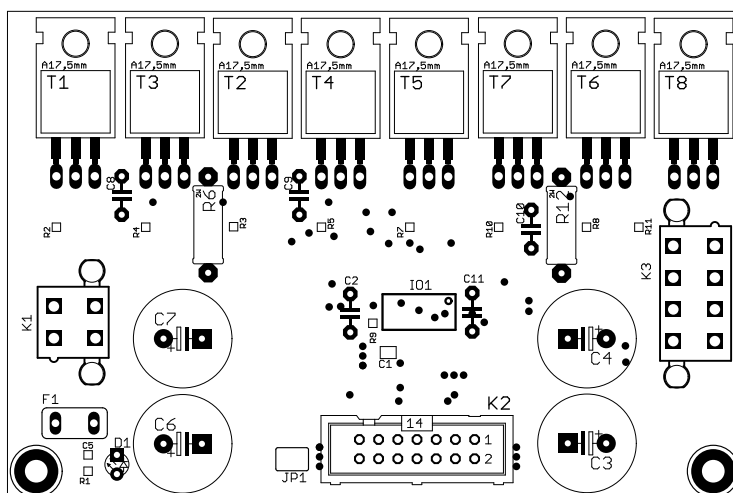
Obr. 10: Schéma zapojení budiče krokového motoru



Obr. 11: Plošný spoj budiče krokového motoru – TOP



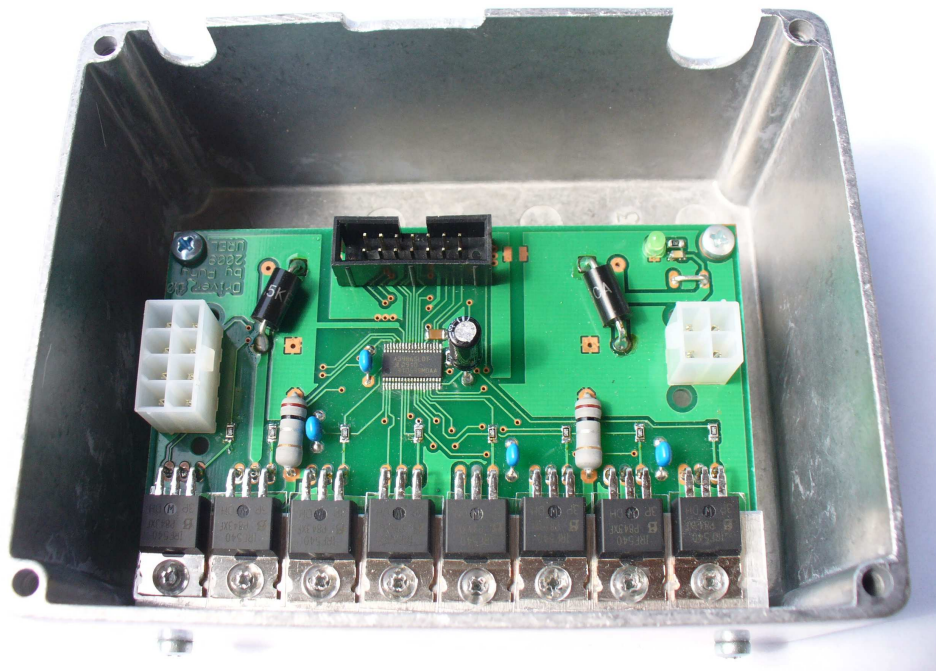
Obr. 12: Plošný spoj budiče krokového motoru - BOTTOM



Obr. 13: Osazení plošného spoje budiče motoru

<b>Označení</b>	<b>Hodnota</b>	<b>Rozměr / pouzdro</b>	<b>Množství</b>
C1 + Cds (viz text)	100n	1206	9
C2	1uF	CK RM5mm	1
C3, C4, C6, C7	viz text		4
C5	100n	0805	1
C8, C9, C10, C11	22n RM5mm	CK RM5mm	4
D1	LEDG 3mm		1
IO1	A3986		1
JP1	neosazeno		1
K1	5566-4	Molex 5566-4	1
K2	MLW14		1
K3	5566-8	Molex 5566-8	1
R1, R9	10k	0805	2
R2, R3, R4, R5, R7, R8, R10, R11	68R	0805	8
R6, R12	0.1R	R 2W	2
T1, T2, T3, T4, T5, T6, T7, T8	IRF540	TO220	8
F1	neosazena	TE5	1

*Tabulka 8: Seznam součástek pro budič krokového motoru*



*Obr. 14: Osazená deska budiče v kovové skřínce*

## 3.2 Řídicí deska

K výše popsanému budiči krokových motorů byla navržena řídicí deska, která zajišťuje definované řízení až dvou budičů, propojení s PC prostřednictvím USB, místní ovládání pomocí tlačítek a zobrazení informací na alfanumerickém LCD displeji.

### 3.2.1 Popis zapojení

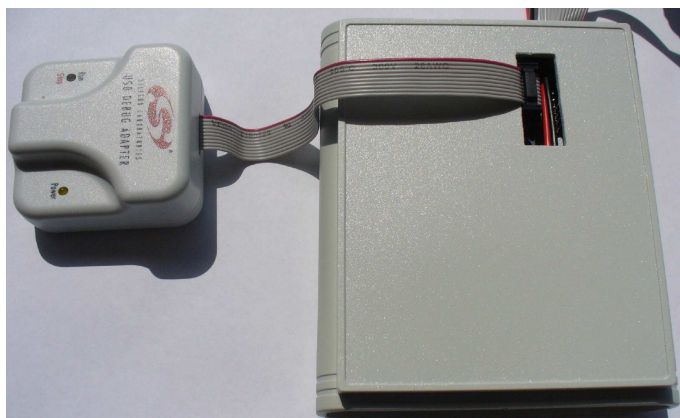
#### 3.2.1.1 Zdroj

Zdroj na řídicí desce zajišťuje usměrnění a filtraci střídavého napětí a jeho stabilizaci na požadovaných 5 a 3,3V.

Napájecí napětí 6 – 12V AC je přivedeno na konektor K9 a po usměrnění můstkem D8 je za D9 vyfiltrováno. Informace o dostupném AC napájení je získávána přes rezistor R27, LED dioda D11 indikuje připojení AC napájení a zároveň jednoduše stabilizuje napětí na signálu „AC\_NAPAJENI“. Signál „AC\_NAPAJENI“ je veden do mikrokontroléru kde poskytuje informaci o dostupnosti napájení. Stabilizátor IO4 vyrábí napětí 5V pro LCD displej a pomocné obvody, IO5 poté 3,3V pro mikrokontrolér. D10 indikuje dostupnost 5V a 3,3V.

#### 3.2.1.2 Obvody mikrokontroléru

Jako řídicí prvek byl zvolen mikrokontrolér s integrovaným USB rozhraním 80C51F320 v 32 pinovém LQFP pouzdře. Schéma zapojení řídicí desky je na obr. 17. Zapojení „debug“ konektoru K5 a obvodu reset vychází z [8] tak, aby byla řídicí deska kompatibilní s vývojovými prostředky Silicon Laboratories (USB debug adapter). Oscilátor byl využit interní, není tedy třeba připojovat krystal. K mikrokontroléru je připojen LCD displej 2x16 znaků se standardním řadičem HD77480. Díky 5V tolerantním vstupům IO1 nemusí být mezi LCD a mikrokontrolérem převáděny logické úrovně (5/3,3V). Dále je zde 5 tlačítek pro komunikaci s uživatelem a ruční ovládání rotátoru. Tlačítka mohou být osazena na DPS nebo může být použita fóliová klávesnice připojitelná ke konektoru K1. Konektor K2 sběrnice USB je připojen podle [8] přímo na piny D+, D-, GND a VBUS mikrokontroléru, je použit typ B podle [10].



Obr. 15: Připojení debug adaptéru Silicon Laboratories k řídicí desce

### 3.2.1.3 Připojení budičů

K připojení budičů krokových motorů k řídicí desce slouží konektory K7 a K8. Signály RESET, DIR, STEP a ENABLE obou konektorů jsou vedeny přímo na piny mikrokontroléru. Ostatní signály jsou vedeny k jumperům JP1 – JP4 kde je možno ručně nastavit požadované hodnoty jak podrobně ukazuje tabulka 9. Referenční napětí pro desky budičů je získáváno děličem R1 a R2 (resp. R3 a R4) a jumperem JP1 (resp. JP2) je možno zvolit zda se bude získávat z 5V nebo 3,3V napájecí větve. Nastavení jumperů ukazuje tabulka 10.

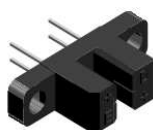
Pro ovládání externího výkonového zdroje pro budiče je v zapojení navržen optočlen IO10. Povolení externího napájení je indikováno diodou D9. Toto řešení umožňuje softwarově kontrolovat napájení budičů a při odpojování napájení nejprve přepnout budič do úsporného režimu, poté odpojit jeho napájení a až následně vypnout řídicí desku. Díky tomuto řešení jsou omezeny přechodové jevy při vypínání a chráněn vlastní budič. Po testování v praxi nebyla tato alternativa využita. Pokud jsou na silové napájecí větvi zapojeny transily, není nutno dodržovat sled vypínání a výstup optočlenu IO10 je možno použít pro případné rozšíření zařízení v budoucnu.

### 3.2.1.4 Vstupy od čidel nulové polohy

V rotátoru je pro každý motor navrženo optické čidlo nulové polohy. Toto řešení umožňuje získat informaci o nulové poloze pohonu a od té počítat jeho polohu relativní.

Vstupy pro dvě čidla jsou řešeny přes konektory RJ11 K6, K7. Napájení pro optická čidla je získáváno z 5V napájecí větve přes ochranné rezistory R10, R11. Ochranu vstupu mikrokontroléru zajišťují ochranné rezistory R14, R15 spolu s diodami D2, D3. Indikace aktivního vstupu zajišťují D4 a D5. Přes D6 a D7 je stav vstupu veden na vstup INT0 mikrokontroléru a při změně je vyvoláno přerušení, které zajistí vynulování čítače kroků, případně opravu hodnoty proměnné s relativní hodnotou polohy pohonu.

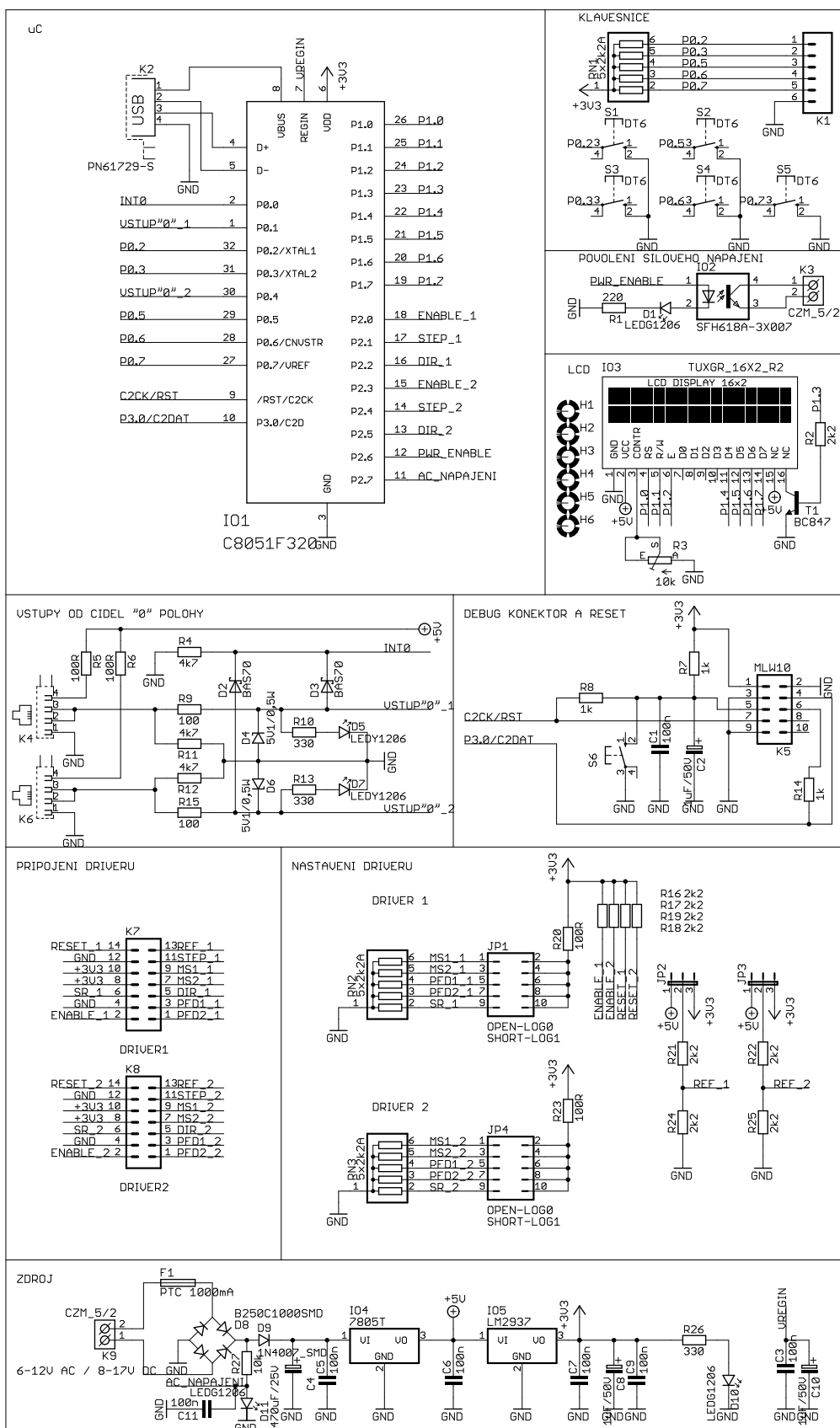
Zvolena byla optická čidla Vishay TCST2103 s rozlišením 0,6mm a šířkou svazku 1mm. Čidlo obsahuje IR LED a fototranzistor s filtrem na denní světlo [12].



Obr. 16:  
Optické  
čidlo nulové  
polohy

## 3.2.2 Plošný spoj a mechanická konstrukce

Pro řídicí desku byla navržena dvouvrstvá DPS s prokovenými otvory o rozměrech 88 x 83mm. Navržená DPS je na obr. 18, 19, osazovací plán poté na obr. 20, 21. Ze strany TOP je osazen LCD modul, mikrokontrolér, tlačítka a některé nízké pasivní SMD součásti. Vyšší součástky jako např. konektory jsou na straně BOTTOM. Tato konstrukce umožňuje desku montovat přímo pod čelní panel bez nutnosti propojování LCD a tlačítek dalším vodičem.



Obr. 17: Schéma zapojení řídicí desky

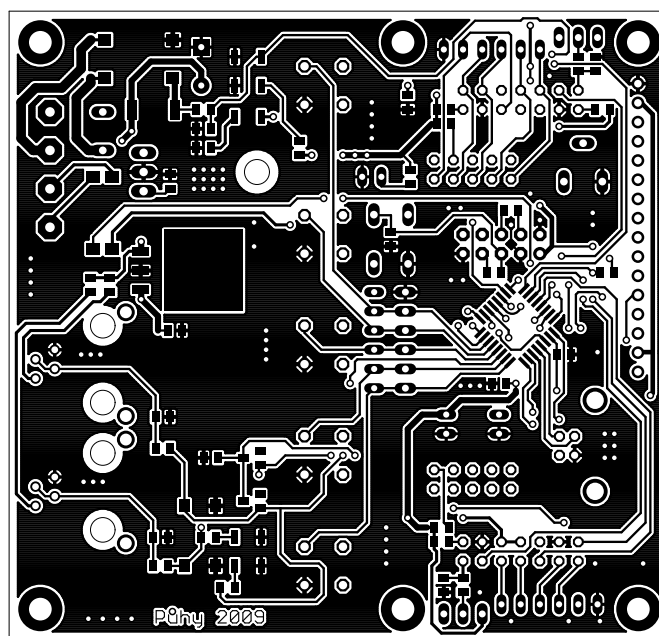
Celek je vestavěn v plastové skříňce Z91 o rozměrech 36x100x130 mm. Skříňka je vyrobena z materiálu ABS. Díky rozměrům řídicí desky nejsou k upevnění třeba žádné šrouby.

<i>Pin</i>	<i>Význam</i>
1 + 2	MS1
3 + 4	MS2
4 + 6	PFD1
7 + 8	PFD2
9 + 10	SR

*Tabulka 9: Význam pinů JP3 a JP4 (jumper zkratován = log1, jumper rozpojen = log0)*

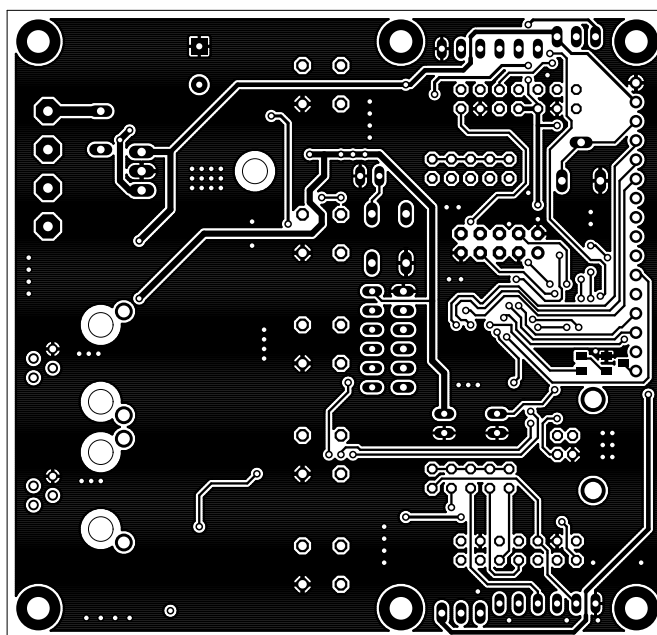
<i>Pin</i>	<i>Význam</i>
1 + 2	Referenční napětí vytvářeno z 5V větve
2 + 3	Referenční napětí vytvářeno z 3,3V větve

*Tabulka 10: Význam pinů JP1 a JP2*

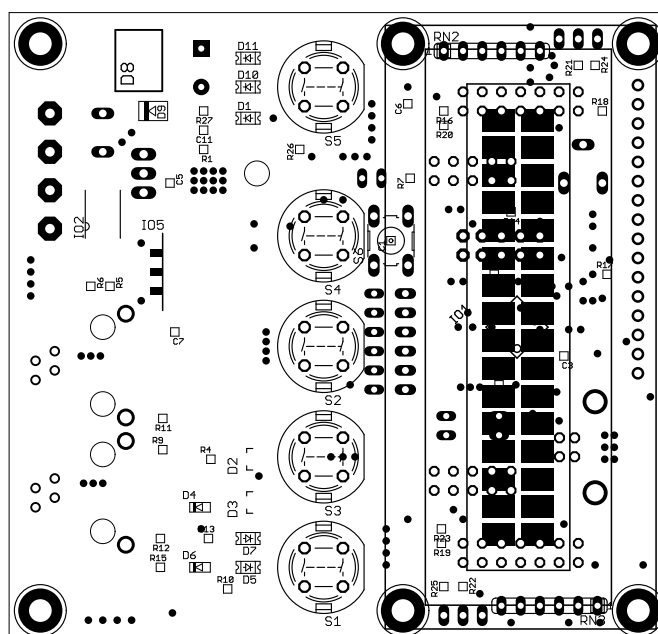


*Obr. 18: Návrh plošného spoje řídicí desky - TOP*

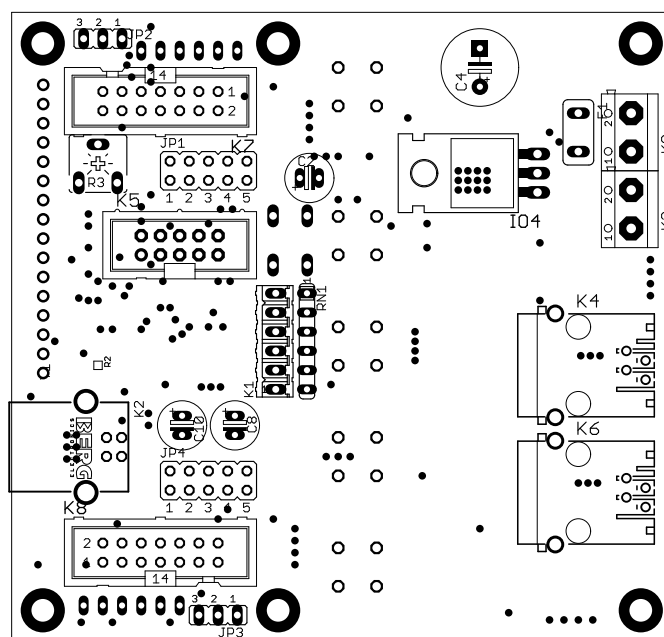




Obr. 19: Návrh plošného spoje řídicí desky -  
BOTTOM



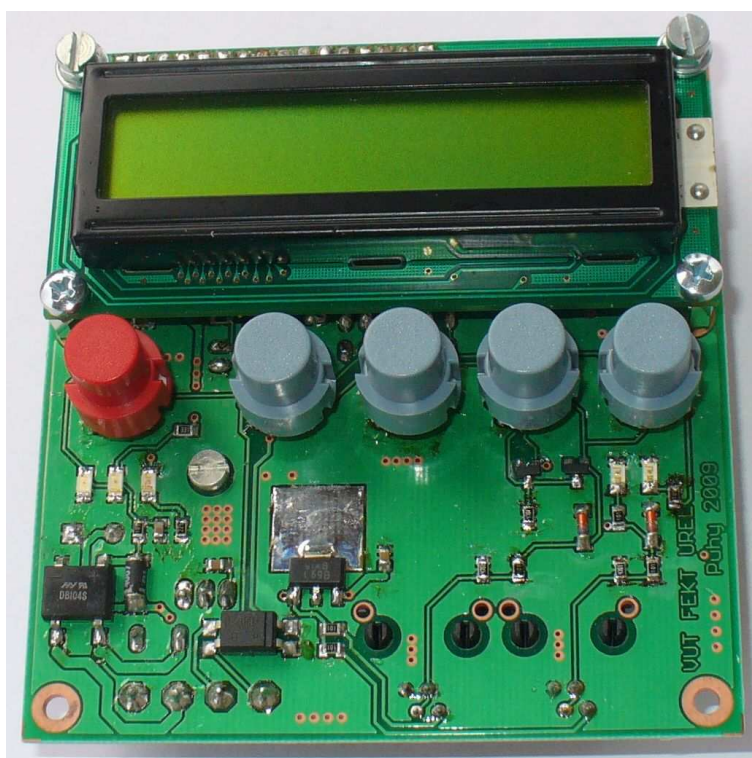
Obr. 20: Řídicí deska - osazovací plán - TOP



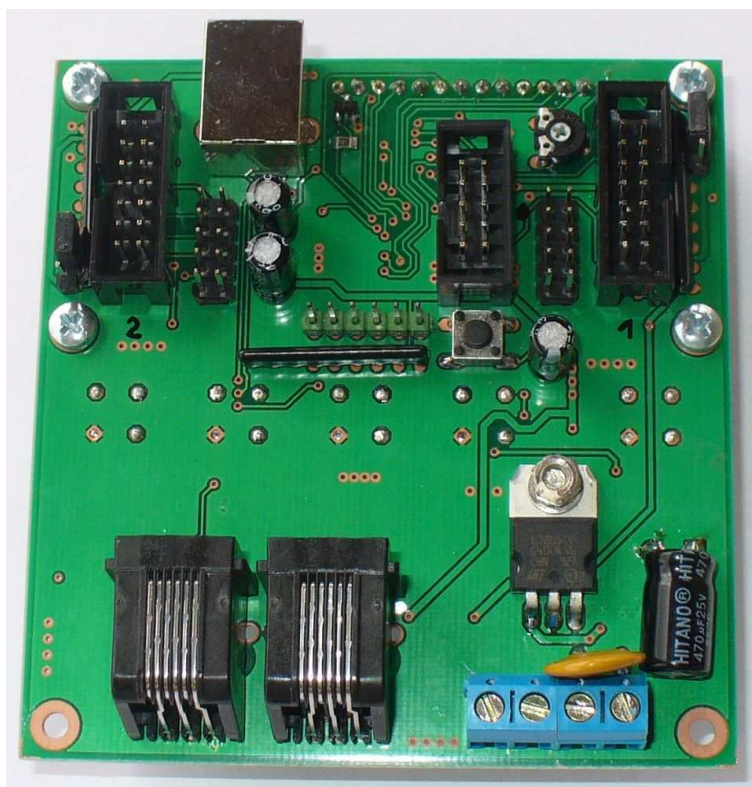
Obr. 21: Řídicí deska - osazovací plán BOTTOM

<b>Označení</b>	<b>Hodnota</b>	<b>Rozměr / pouzdro</b>	<b>Množství</b>
C1, C3, C5, C6, C7, C9, C11	100n	0805	7
C2, C8, C10	1uF/50V		3
C4	470uF/25V		1
D1, D10, D11	LEDG1206	1206	3
D2, D3	BAS70	BAS70	2
D4, D6	5V1/0,5W	SOD-80	2
D5, D7	LEDY1206	1206	2
D8	B250C1000SMD		1
D9	1N4007_SMD	SM-1	1
IO1	C8051F320	LQFP32	1
IO2	PC452		1
IO3	LCD16x2		1
IO4	7805T	7805T	1
IO5	LM2937	LM2937	1
T1	BC847	SOT23BEC	1
JP1, JP4	Lišta 2x5 pinů	RM 2,54mm	2
JP2, JP3	Lišta 1x3 piny	RM 2,54mm	2
K1	ARK500/2		1
K2	USB B konektor	Berg PN61729-S	1
K3, K9	ARK500/2		2
K4, K6	RJ11	Molex 555154-2	2
K5	MLW10		1
K7, K8	MLW14		2
R1	220R	0805	1
R10, R13, R26	330R	0806	3
R2, R16, R17, R18, R19, R21, R22, R24, R25	2k2	0807	9
R27	10k	0808	1
R3	10k	CA6V	1
R4, R11, R12	4k7	0805	3
R5, R6, R20, R23	100R	0806	4
R7, R8, R14	1k	0807	3
R9, R15		100 0808	2
RN1, RN2, RN3	5x2k2A		3
S1, S2, S3, S4, S5	DT6		5
S6		10-XX	1
F1	PTC 1000mA	TE5	1

*Tabulka 11: Seznam součástek pro řídicí desku*



Obr. 22: Osazená řídicí deska - TOP



Obr. 23: Osazená řídicí deska - BOTTOM



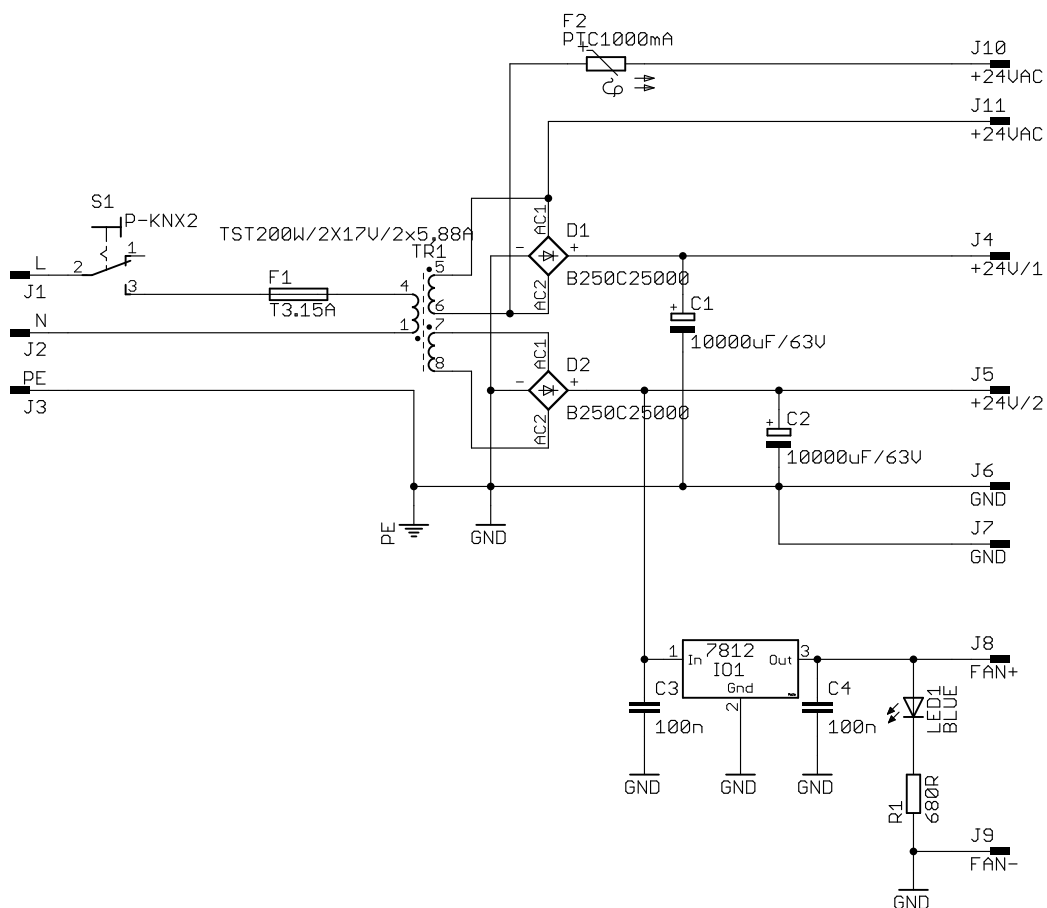
*Obr. 24: Řídicí deska vestavěná v přístrojové skřínce Z 91*

### **3.3 Napájecí zdroj**

#### **3.3.1 Popis zapojení**

Pro napájení vlastních motorů a řídicí elektroniky ze sítě 230V AC byl navržen jeden napájecí zdroj 2 x 24V DC / 2 x 5,88A. Na jedno sekundární vinutí použitého transformátoru je ještě připojen zdroj řídicí elektroniky. Schéma zapojení napájecího zdroje je na obr. 25. Sítové napětí je přes pojistku a vypínač vedeno na primární vnutí transformátoru TR1. Usměrnění a filtraci zajišťují můstky D1 a D2 spolu s kondenzátory C1 a C2.

Země usměrněných napětí jsou navzájem spojeny a připojeny na kostru zdroje a vodič PE. Přes pojistku PTC F2 je vyvedeno napájení pro řídicí desku. Stabilizátor IO1 zajišťuje napětí 12V pro ventilátor a indikační LED.



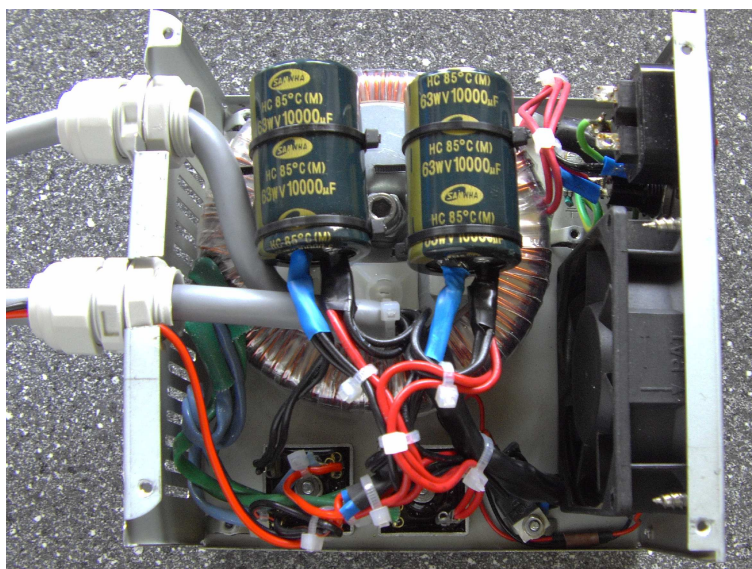
Obr. 25: Schéma zapojení napájecího zdroje

### 3.3.2 Mechanická konstrukce

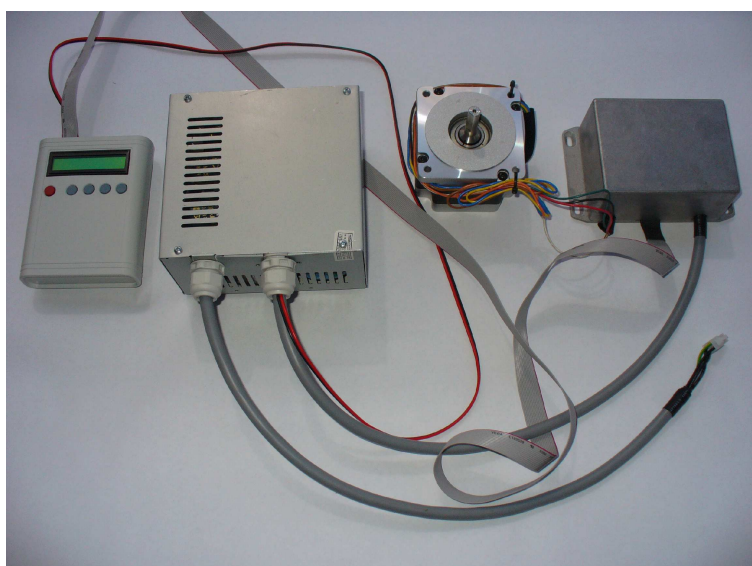
Obvod zdroje byl realizován v kovové skřínce od vyřazeného AT zdroje. Byly využity původní vestavěné konektory a ventilátor. Propojení elektronických součástek je zde provedeno pomocí vodičů a konektorů faston, případně pájených spojů. Využití plošného spoje se zde jevilo jako zbytečné. Usměrnovací můstky jsou upevněny na kovové šasi skříňky, čímž je zajištěno jejich chlazení. Filtrační kondenzátory mechanicky upevňují platové stahovací pásy na kovové platli transformátoru pomocí samolepících přichytek. Vývody pro napájení dvou budičů jsou provedeny pomocí stíněných silových kabelů a opatřeny konektory pro přímé připojení k budičům. Vývody jsou provedeny plastovými průchodkami PG9 a PG13,5. Mechanická konstrukce je zřejmá z uvedených fotografií.

<b>Označení</b>	<b>Hodnota</b>	<b>Rozměr / pouzdro</b>	<b>Množství</b>
C3, C4	100n	RM 5mm	2
R1	680R	0207	1
IO1	7812	TO220	1
C1, C2	10000uF/63V		2
D1, D2	B250C25000		2
LED1	BLUE		1
S1	P-KNX2		1
F2	PTC1000mA		1
F1	T3.15A	5x20mm	1
TR1	TST200W/2X17V/2x5,88A		1
Skříňka AT zdroje včetně ventilátoru a konektorů			1
J1 – J11	FASTON 6,3mm		11

*Tabulka 12: Seznam součástek - napájecí zdroj*



*Tabulka 13: Vnitřní uspořádání zdroje*



*Tabulka 14: Hotový zdroj spolu s budičem, krokovým motorem a řídicí jednotkou*



## 3.4 Propojení dílčích částí

### 3.4.1 Přivedení napájecího napětí

Přivedení silového napájecího napětí od zdroje pro budiče motorů je realizováno pomocí silových kabelů  $4 \times 1,5 \text{ mm}^2$  Cu. Vždy dva vodiče jsou spojeny a tvoří kladný a záporný pól. Záporný pól je ve zdroji spojen s kostrou a vodičem PE.

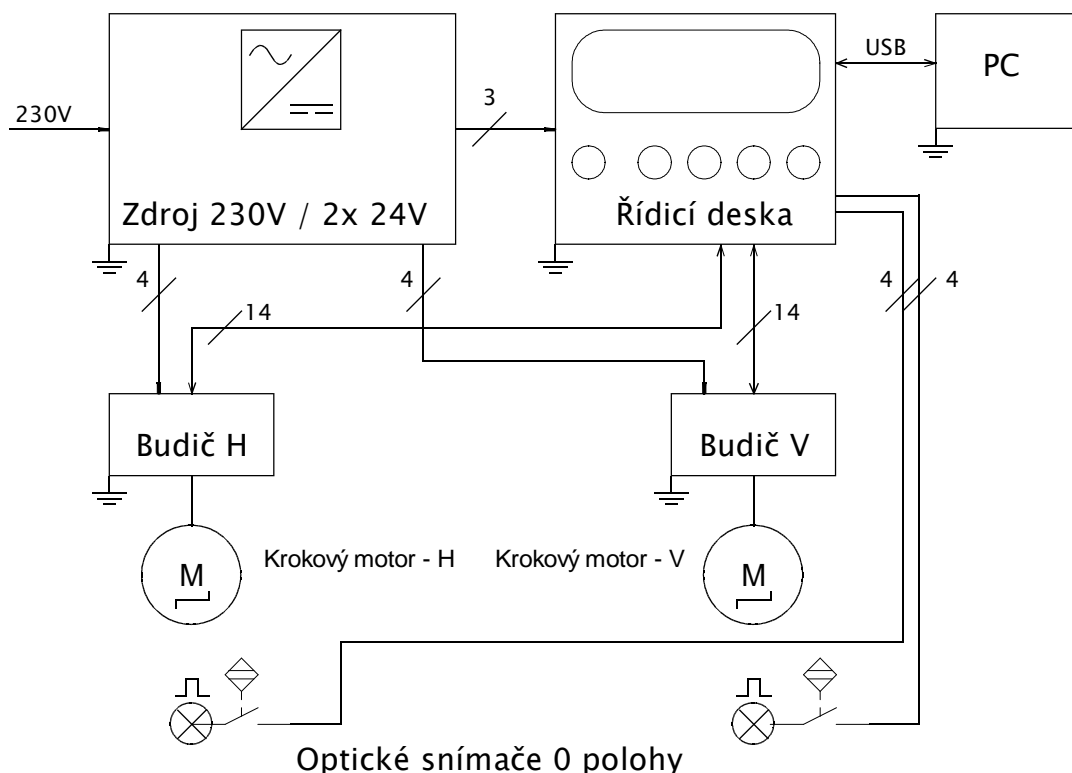
K řídicí desce je přivedeno napájecí napětí pomocí stíněného dvoužilového vodiče, kde stínění je připojeno na záporný pól zdroje a vodič PE. Na řídicí desce poté na záporný pól za usměrňovačem. Zbývající dva vodiče vedou střídavé napětí (řídicí deska je vybavena vlastním usměrňovačem a stabilizátorem). Propojení záporných pólů je zde nutné, aby měla řídicí deska stejnou zem jako budiče motorů. Připojení na vodič PE je důležité z důvodu vyrovnaní potenciálu na konektoru USB řídicí desky a připojovaného PC.

### 3.4.2 Datové propojení

Mezi řídicí deskou a budiči krokových motorů je vedena datová sběrnice pomocí plochých 14 – ti žilových vodičů AWG28 se zářezovými konektory.

Optická čidla nulové polohy jsou připojena pomocí plochých vodičů a zářezových konektorů RJ-11, které jsou zapojeny do řídicí desky.

Propojení celé sestavy je zřejmé z blokového schématu na obr 26.



Obr. 26: Propojení sestavy elektroniky rotátoru





mikrokontroléru.

#### 4.1.1 Ovládání budičů krokových motorů

Pro ovládání budičů krokových motorů A3986 byly vytvořeny následující funkce v souladu s [5]. Budiče krokových motorů jsou označeny podle montáže na mechanickou část rotátoru jako horizontální (H) a vertikální (V).

##### 4.1.1.1 *step\_H\_R();*

Provede 1 krok doprava horizontálním pohonem včetně potlačení pohybu vertikálního posuvu vlivem horizontálního otáčení.

```
void step_H_R()          //horizontální motor - krok doprava
{
    if(vstup_H==1)       //pokud není horizontální pohon na doraze
    {
        Stop();          //pozastaví krokování v případě požadavku od
                           červeného tlačítka
        DIR_1=1;          //provedení jednoho kroku
        DelayUs1(zpozdeni);
        STEP_1=1;
        DelayUs1(zpozdeni);
        STEP_1=0;
        if(pozice_H==(kroku_H-1)){pozice_H=0;}else{pozice_H++;}
        Send_USB_H();     //odeslání aktuální pozice přes USB
        dispX();           //zobrazení na LCD
        DIR_2=1;          //korekce polohy vertikálního pohonu aby nešel
                           do spirály
        DelayUs1(zpozdeni_V);
        STEP_2=1;
        DelayUs1(zpozdeni_V);
        STEP_2=0;
    }
    else
    {
        ztrata();          //ztráta polohy
    }
}
```

##### 4.1.1.2 *step\_H\_L();*

Provede 1 krok doleva horizontálním pohonem včetně potlačení pohybu vertikálního posuvu vlivem horizontálního otáčení. Zdrojový kód je stejný jako u *step\_H\_R()*, s tím rozdílem, že je prováděn posun na druhou stranu ( *DIR\_1=1* a *pozice\_H--;*).

##### 4.1.1.3 *step\_V\_R();*

Provede 10 kroků doprava vertikálním pohonem. Vzhledem k rozlišení šroubu pro posun ve vertikálním směru na použité mechanické montáži (600kroků / 5mm) jsou kroky prováděny po 10.

```
void step_V_R()          //vertikální motor - 10 kroků doprava
{
    if(vstup_V==1)
```

```

    {
    Stop();
    for(i=0;i<10;i++)          //požadavek na 10 kroků
    {
        DIR_2=1;
        DelayUs1(zpozdeni_V);
        STEP_2=1;
        DelayUs1(zpozdeni_V);
        STEP_2=0;
    }
    if(pozice_V==kroku_V-10){pozice_V=0;}else{pozice_V+=10;}
    Send_USB_V();              //odeslání aktuální polohy na USB
    dispX();                    //zobrazení na LCD
    }
else
    {
        ztrata();              //ztráta polohy
    }
}

```

#### 4.1.1.4 step\_V\_L();

Provede 1 krok doprava vertikálním pohonem. Funkce je stejná jako u step\_V\_R(), s tím rozdílem, že je prováděn posun na druhou stranu (DIR\_2=0 a pozice\_V=-10 ).

**kalibrace();**

Krokuje oběma pohony dokud nedosáhne nulové polohy, poté vynuluje proměnnou polohy.

```

void kalibrace()                //kalibrace pohonu - {
                                //čekání na impuls od nulových čidel
                                //otázka na uživatele

    set_cursor(0,0);
    printf("Kalibrovat?      ");
    set_cursor(0,1);
    printf("Ano              Ne");
    set_cursor(0,1);

    if(kbd_press(1)==1) {kalib=1; DelayMs(10000); return;} //ne
    if(kbd_press(5)==1) //ano
    {
        printf(">>> Horizontalni");
        while(SW5==1); //spustit červeným tlačítkem
        DelayMs(100000);
        while(vstup_H==1){step_H_R();} //dosažení nulové polohy H
        pozice_H=0; //vynulování skutečné polohy H
        Send_USB_H(); //odeslání polohy H do PC
        printf("Nutno kalibrovat");
        set_cursor(0,1);
        printf(">>> Vertikalni");
        while(SW5==1);
        DelayMs(10000);
        while(vstup_V==1){step_V_L();} //dosažení nulové polohy V
        pozice_V=0; //vynulování skutečné polohy V
        Send_USB_V(); //odeslání polohy V do PC
        set_cursor(0,0);
        printf("              ");
        set_cursor(0,1);
    }
}

```

```

printf("Opakovat   Hotovo");
DelayMs(100000);
while(1)
{
if(kbd_press(1)==1) {kalib=1; break; }    //hotovo (kalib=1)
if(kbd_press(5)==1) {kalib=0; break; }    //opakovat (kalib=0)
}
DelayMs(10000);
}
}

```

#### 4.1.1.5 ztrata();

Tato funkce je volána v případě ztráty polohy pohonů – indikuje uživateli problém a zajistí zastavení veškerého pohybu.

```

void ztrata()          //indikace ztraty kroku
{
set_cursor(0,0);
printf("Ztrata polohy V ");
set_cursor(0,1);
printf("Rekalibrovat   ");
DelayMs(10000);
kalibrace();
while(1);             //čekání na obsluhu (reset)
}

```

#### 4.1.1.6 Stop();

Pozastavení pohybu kdykoliv při krokování červeným tlačítkem na řídicí jednotce. Znovuspustění se provádí opět červeným tlačítkem SW5.

```

void Stop()           //pozastaveni pri krokovani cervenym tlacitkem SW5
{
if(kbd_press(5)==1)
{
set_cursor(0,0);
printf("   Pozastaveno   ");
set_cursor(0,1);
printf(">>>                ");
DelayMs(10000);
while(SW5==1);        //čekání na opětovný stisk
DelayMs(10000);
dispX();              //aktualizace LCD (aktuální polohy)
}
}

```

### 4.1.2 Obsluha klávesnice

Stisk tlačítek testuje funkce `kbd_press(unsigned char n)`, jako `n` vrací přímo číslo stisknutého tlačítka.

### 4.1.3 Obsluha LCD displeje

Ovladač pro alfanumerický LCD displej 2 x 16 znaků musel být vytvořen částečně vlastní podle [13], protože připojení datových vodičů displeje není provedeno systematicky na

jeden port mikrokontroléru, jak je obvyklé. Důvodem byl jednodušší návrh DPS a nedostatek volných pinů na jednom portu. Výsledkem je ovladač, který umožňuje ovládat displej jehož datové a řídicí vodiče jsou připojeny na libovolný pin mikrokontroléru s tím, že musí mít pochopitelně povolen potřebný směr komunikace.

Pro tuto funkcionalitu musela být přepracována funkce pro zápis dat do LCD displeje tak, aby pracovala s jednotlivými piny které jsou připojeny na datovou sběrnici displeje. Výsledkem je konstrukce uvedená níže.

```
void lcd_write(unsigned char c)
{
    //PORTB = c >> 4; //původní řešení
    DB4 = (c & 0x10) ? 1 : 0;
    DB5 = (c & 0x20) ? 1 : 0;
    DB6 = (c & 0x40) ? 1 : 0;
    DB7 = (c & 0x80) ? 1 : 0;
    LCD_STROBE;
    DelayUs(80);
    //PORTB = c; //původní řešení
    DB4 = (c & 0x01) ? 1 : 0;
    DB5 = (c & 0x02) ? 1 : 0;
    DB6 = (c & 0x04) ? 1 : 0;
    DB7 = (c & 0x08) ? 1 : 0;
    LCD_STROBE;
    DelayUs(80);
}
```

Dále je ještě upraven výpis znaků tak, aby šla používat funkce printf();, je využito přeměrování pomocí putchar přímo v obsluze LCD.

```
char putchar (char c){ //presmerovani na printf
    if(LCD_RS==0){LCD_RS=1;DelayUs(80);}else{}
    lcd_write(c);
    return(c);
}
```

Samotný výpis hodnot na LCD displej provádí funkce disp(); volaná cyklicky. Zároveň je v této funkci proveden i přepočítání na stupně a mm pro přímé zobrazení polohy antény připevněné na rotátor.

```
void disp(){
    pom=(pozice_H*prevod_H)/10; //prevod kroky -> stupne
    set_cursor(0,0);
    printf("H %5dst %5dk ",pom,pozice_H); //H 120st 600k
    pom=(pozice_V*5)/60; //prevod kroky -> mm
    set_cursor(0,1);
}
```

```

        printf("V %5dmm %5dk      ",pom,pozice_V); //V      120mm   1500k
    }

```

Kromě těchto výše uvedených úprav je použit standardní ovladač LCD displeje podle [13].

#### 4.1.4 Obsluha USB

Mikrokontrolér C8051F320 má implementováno hardwarové USB rozhraní. Pro komunikaci s PC standardem HID byl použit ukázkový příklad od Silicon Laboratories [3], [8], [9], [10], [11].

USB je obsluhováno v přerušení:

```

EIE1 |= 0x02;           // Enable USB0 Interrupts
EA = 1;                 // Global Interrupt enable

```

Dále je využit Timer 2 s nastavením reload hodnoty následovně:

```

sfr16 TMR2RL = 0xca; // Timer2 reload value
sfr16 TMR2    = 0xcc; // Timer2 counter

```

Nastavení descriptorů se provádí v souboru USB\_DESCRIPTOR.c, vlastní komunikaci zajišťuje kód v USB\_STD\_REQ.c a USB\_ISR.c.

Byla nadefinována dvě pole s prvky typu char, pro výměnu dat mezi mikrokontrolérem a PC následovně:

```

unsigned char Out_Packet[6] = {0,0,0,0,0,0}; //Přijatá data přes USB
unsigned char In_Packet[6]  = {0,0,0,0,0,0}; //Odeslaná data na USB

```

Jednotlivé prvky výše uvedených polí jsou využity pro přenos dat o poloze pohonů a dalších pomocných dat takto:

In_Packet[1]	LowByte - požadovaná horizontální poloha do PC
In_Packet[2]	HighByte - požadovaná horizontální poloha do PC
In_Packet[3]	Požadavek od tlačítka horizontální
In_Packet[4]	LowByte - požadovaná vertikální poloha do PC
In_Packet[6]	HighByte - požadovaná vertikální poloha do PC
In_Packet[5]	Požadavek od tlačítka vertikální
Out_Packet[1]	Načtení požadované polohy z PC - horizontální LowByte
Out_Packet[2]	Načtení požadované polohy z PC - horizontální HighByte
Out_Packet[3]	Indikace spuštění programu na PC
Out_Packet[4]	Načtení požadované polohy z PC - vertikální LowByte
Out_Packet[5]	Načtení požadované polohy z PC - vertikální HighByte.

Důležité proměnné pro ukládání polohy pohonů jsou:

```

unsigned int pozice_H, pozice_H1, pozice_V, pozice_V1=0;

```

výše uvedené proměnné s indexem 1 odpovídají požadované poloze od nadřazeného

systému, proměnné bez indexu zde reprezentují skutečnou polohu pohonu v krocích.

Přípravu dat o skutečné a požadované poloze pohonů pro odesílání a příjem přes USB zajišťují níže uvedené funkce. Jde především o rozdělení a následné spojení dvou proměnných typu char do int a naopak:

```
void Read_USB_H() //načtení požadované polohy z PC - horizontální
{
    if(USB_State == DEV_CONFIGURED & ProgramOK)
    {
        LowByte = Out_Packet[1];
        HighByte = Out_Packet[2];
        pozice_H1 = HighByte<<8 | LowByte;
    }
}

void Read_USB_V() //načtení požadované polohy z PC - vertikální
{
    if(USB_State == DEV_CONFIGURED & ProgramOK)
    {
        LowByte = Out_Packet[4];
        HighByte = Out_Packet[5];
        pozice_V1 = HighByte<<8 | LowByte;
    }
}

void Send_USB_H() //odeslání aktuální polohy do PC - horizontální
{
    if(USB_State == DEV_CONFIGURED)
    {
        LowByte = (unsigned char)pozice_H1;
        HighByte = ((unsigned char) (pozice_H1>>8));
        In_Packet[1]=LowByte;
        In_Packet[2]=HighByte;
        In_Packet[3]=tl_h; //odeslání požadavku od tlačítka H >0 -
je požadavek, ==0 není požadavek na změnu požadované polohy v PC
        tl_h=0;
    }
}

void Send_USB_V() //odeslání aktuální polohy do PC - vertikální
{
    if(USB_State == DEV_CONFIGURED)
    {
        LowByte = (unsigned char)pozice_V1;
        HighByte = ((unsigned char) (pozice_V1>>8));
        In_Packet[4]=LowByte;
        In_Packet[6]=HighByte; //odeslání aktuální polohy do PC
        In_Packet[5]=tl_v; //odeslání požadavku od tlačítka V
>0 - je požadavek, ==0 není požadavek na změnu požadované polohy v PC
        tl_v=0;
    }
}
```

Aby bylo možno detekovat zapnutí ovládacího programu na PC a připojení řídicí desky k PC jsou potřebné funkce pro USB komunikaci v podmínce:

```
if(USB_State == DEV_CONFIGURED & ProgramOK){},
```

kde první proměnná indikuje připojení k PC, do ProgramOK se cyklicky kopíruje Out\_Packet[3], který ovládací program na PC posílá jako „1“, program mikrokontroléru ho cyklicky nuluje. Je – li tedy ProgramOK==1, ovládací program na PC je spuštěn. Tato detekce

připojení k PC a spuštění nadřazeného ovládacího programu je důležitá kvůli správné funkci tlačítek na řídicí desce. V případě, že by tato detekce nebyla prováděna, nešlo by pohony ovládat pomocí tlačítek na řídicí desce, protože by se neměnila hodnota proměnné požadované polohy na PC a program na PC by požadoval vrácení pohonu do původní polohy. Takto je zajištěna úplná a oboustranná synchronizace mezi proměnnými v programu mikrokontroléru a programem v PC a možnost ovládání tlačítky na řídicí desce za všech okolností.

#### 4.1.5 Časovač pro realizaci jednoduchého multitaskingu a přepínání úloh

V souboru timer\_drv.c je nastaveno přerušení pro hardwarový časovač Timer 0: `timr0_irq () interrupt 1` a je generován systémový čas, podle kterého se přepínají jednotlivé úlohy, jak je uvedeno výše. Tato funkce běží na pozadí díky přerušení. Hodnota systémového času je k dispozici v proměnné `unsigned long sys_time0`.

Jednoduchý multitasking je realizován pomocí porovnávání nastaveného času „úloh“ v hlavní smyčce `while(1){}` programu následovně:

```
while(1)
{
/*Zhasnutí podsvícení LCD po uplynutí doby blik_time od posledního stisku*/

if(sys_time0()-blik_time>3000)
{

    LED_LCD=0;                //zhasnutí podsvícení LCD
    blik_time=sys_time0();
}

/*Testování stiku klaves + odeslání a příjem dat z PC*/

if(sys_time0()-kbd_time>3)
{
    if(kalib==0){kalibrace();}           //je-li požadavek na kalibraci,
                                         proved' kalibraci

    ProgramOK=Out_Packet[3];             //načtení program na PC 1 spuštěn
                                         0 nespuštěn

    Read_USB_H();
    Read_USB_V();

    if(kbd_press(1)==1)                  //testování tlačítka
    {
        step_H_R();                     //krok
        pozice_H1=pozice_H;             //požadovaná = skutečná
        dispX();                         //zobrazení na LCD
        tl_h=10;                         //indikace stisku HW tlačítka
        blik_time=sys_time0();           //vynulování časovače od
                                         posledního stisku
    }

    if(kbd_press(2)==1)                  //testování tlačítka
    {
        step_H_L();                     //krok
```



```

        pozice_H1=pozice_H;           //požadovaná = skutečná
        dispx();                     //zobrazení na LCD
        tl_h=20;                     //indikace stisku HW tlačítka
        blik_time=sys_time0();       //vynulování časovače od
    }                                //posledního stisku

    if(kbd_press(3)==1)               //testování tlačítka
    {
        step_V_R();
        pozice_V1=pozice_V;
        dispx();
        tl_v=10;
        blik_time=sys_time0();
    }
    if(kbd_press(4)==1)
    {
        step_V_L();
        pozice_V1=pozice_V;
        dispx();
        tl_v=20;
        blik_time=sys_time0();
    }

    ProgramOK=Out_Packet[3];          //načtení program na PC 1 spuštěn
    0 nespuštěn

    if(pozice_V != pozice_V1)         //porovnání skutečné a požadované
    {                                  polohy V + dorovnání
        if(pozice_V1 < pozice_V)
        {
            while(pozice_V != pozice_V1)
            {
                step_V_L();
                dispx();
            }
        }

        if(pozice_V1 > pozice_V)
        {
            while(pozice_V != pozice_V1)
            {
                step_V_R();
                dispx();
            }
        }
    }
    Send_USB_V();                     //odeslání dat o vertikální poloze
    if(pozice_H != pozice_H1)         // porovnání skutečné a požadované
    {                                  polohy H + dorovnání
        if(pozice_H1 < pozice_H)
        {
            while(pozice_H != pozice_H1)
            {

```

```

        step_H_L();
        dispX();
    }
    if(pozice_H1 > pozice_H)
    {
        while(pozice_H != pozice_H1)
        {
            step_H_R();
            dispX();
        }
    }

    Send_USB_H(); //odeslání dat o
                  horizontální poloze přes USB

    if(ProgramOK){ProgramOK=0; Out_Packet[3]=0;} //nulování paketu
                                                    testování spuštěného
                                                    programu na PC

    kbd_time=sys_time0(); //nulování času od posledního
                          provedení if(sys_time0()-
                          kbd_time>3)
    }
} //konec while

```

## 4.2 Software pro PC

Pro ovládání rotátoru z PC byl vytvořen program pomocí vývojového prostředí C++ Builder a knihoven JEDI Visual Component Library [14] pro OS Windows XP, Vista a 7. Pro obsluhu USB HID je použita komponenta `TJvHidDevice` z výše uvedené knihovny. Zjednodušený vývojový diagram je na obr. 28. Program cyklicky testuje připojení řídicí desky k USB portu PC a přijímá data o skutečné poloze pohonů. Porovnává požadovanou a skutečnou polohu pohonů a odesílá požadovanou polohu pohonu řídicí desce. Dále reaguje na stisk hardwarového tlačítka na řídicí desce změnou požadované polohy podle významu stisknutého tlačítka.

Kompletní zdrojové kódy a spustitelný soubor jsou součástí elektronické přílohy práce.

### 4.2.1 Enumerace a testování připojení HID zařízení

Enumeraci a detekci připojení zajišťuje funkce `HIDEnumerate(TJvHidDevice *HidDev, const int Idx)`, která při úspěšné enumeraci změní barvu prvku `Shape3` na zelenou a vypíše „Zařízení připojeno“.

Pro detekci spuštění ovládacího programu na PC řídicí deskou je cyklicky odeslán `Out_Packet[4] = 1`.

### 4.2.2 Zpracování přijatých paketů

Zpracování přijatých paketů od řídicí desky má na starost funkce `OnDeviceData(TJvHidDevice* HidDev, Byte ReportID, const void * Data, Word Size)`, ve které jsou zpracovává přijatá data (`In_Packet`) do proměnných typů `int`

následujícím způsobem:

```
pozice_V1=In_Packet4<<8 | In_Packet3;
```

Vyrovnění požadované a skutečné polohy probíhá např. pro vertikální rovinu následovně:

```
ScrollBar1->Position=pozice_V1;.
```

Dále jsou ve funkci OnDeviceData(TJvHidDevice\* HidDev, Byte ReportID, const void \* Data, Word Size) zpracovávány požadavky od hardwarových tlačítek na řídicí desce na změnu polohy. Když je totiž řídicí deska připojena k PC se spuštěným programem, je třeba synchronizovat ovládání programem PC a hardwarovými tlačítky aby nedocházelo ke kolizím. Při testování se tento problém projevoval tak, že řídicí deska provedla změnu o jeden krok, tu odeslala PC a program na PC vyslal zpět požadavek na vrácení do původní polohy, protože původně nebyl přenášén stav tlačítek na řídicí desce.

Tento problém je ošetřen následovně:

```
if(In_Packet5>=9){ScrollBar2->Position=pozice_V1;} pro vertikální posun a
```

```
if(In_Packet2>=9){ScrollBar4->Position=pozice_H1;} pro horizontální posun.
```

V In\_Packet2, resp 5 je přenášén stav hardwarových tlačítek na řídicí desce, je tedy možno přesně identifikovat, které tlačítko bylo stisknuto.

Hodnota v proměnné pozice\_\*1 obsahuje aktuální polohu pohonu a ta se přenese do požadované polohy pohonu ScrollBar4(2)->Position. Tím je zajištěno, že při ovládání tlačítka na řídicí desce je tato hodnota synchronizována mezi PC a řídicí deskou a je možno ovládat řídicí desku jak programem na PC, tak hardwarovými tlačítky.

Dále je provedena indikace stisku hardwarových tlačítek pomocí objektů Shape1,2,4 a 5. In\_Packet2, resp. 5 obsahuje hodnotu 0, pokud nebylo stisknuto tlačítko, případně hodnotu 10 nebo 20 podle stisku tlačítka na řídicí desce.

```
if (In_Packet2==10)
{
    Shape1->Brush->Color=clGreen;
}
else
{
    Shape1->Brush->Color=clWhite;
}

if (In_Packet2==20)
{
    Shape2->Brush->Color=clBlue;
}
else
{
    Shape2->Brush->Color=clWhite;
}

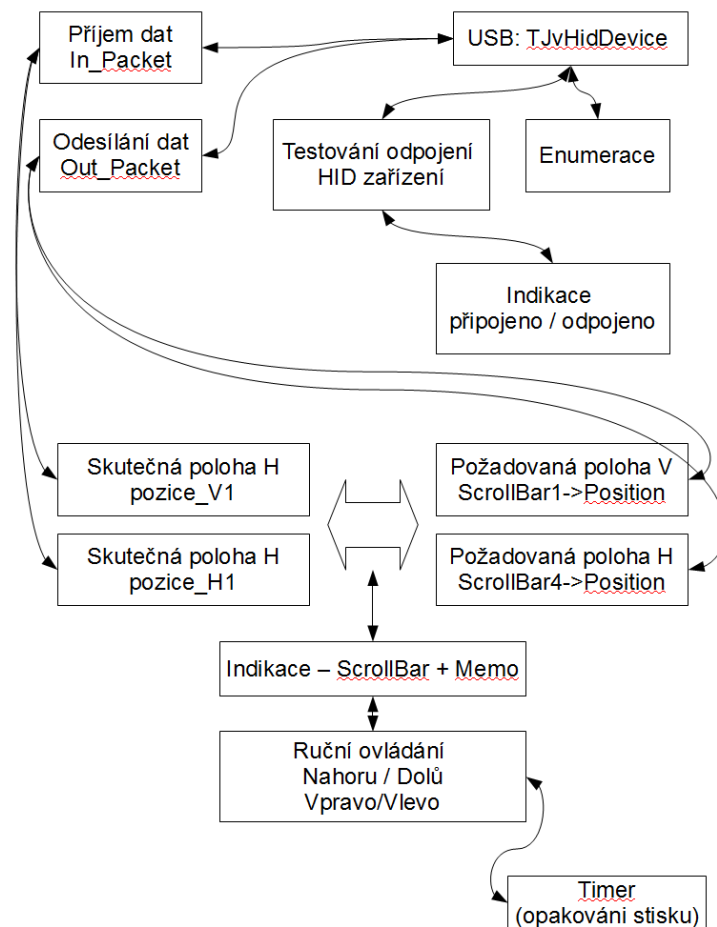
if (In_Packet5==10)
{
    Shape4->Brush->Color=clGreen;
}
else
{
    Shape4->Brush->Color=clWhite;
}
```

```

    }

    if (In_Packet5==20)
    {
        Shape5->Brush->Color=clBlue;
    }
else
    {
        Shape5->Brush->Color=clWhite;
    }

```



Obr. 28: Vývojový diagram - aplikace pro ovládání rotátoru z PC

V tabulce 15 jsou uvedeny odpovídající si indexy In\_Packet a Out\_Packet využitých pro přenos dat mezi PC a řídicí deskou (μC).

#### 4.2.3 Odesílání paketů řídicí desce

Při vyvolané události ScrollBar4Change(TObject \*Sender) dochází k odeslání požadované horizontální polohy pohonů řídicí desce následujícím způsobem:

```
Out_Packet[0] = 0; // ReportID = 0
```

```

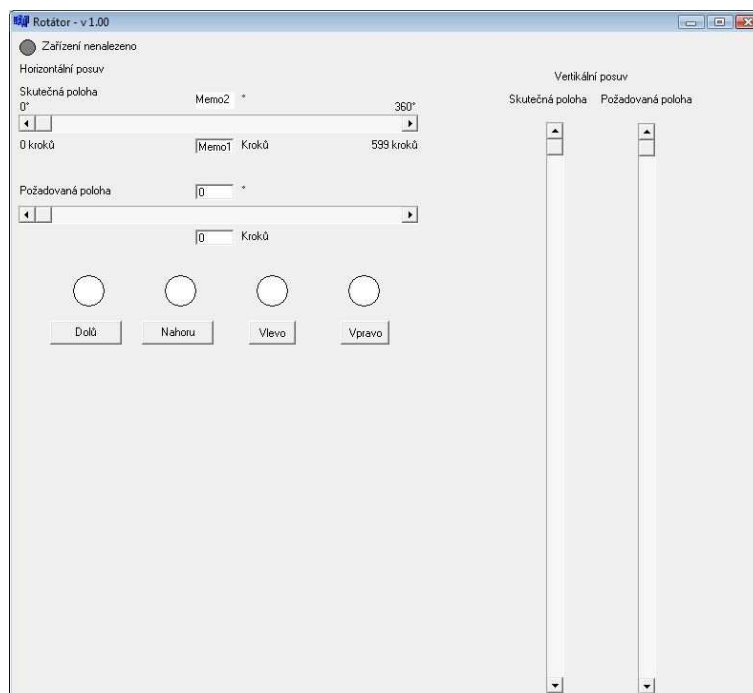
Out_Packet[4] = 1; //posila 1 na detekci zapnuti programu
Out_Packet[2] = (unsigned char)(ScrollBar4->Position); //Low Byte
Out_Packet[3] = (unsigned char)((ScrollBar4->Position)>>8); //HigtByte
ActiveDevice->WriteFile(Out_Packet, 9 , Written);

```

Obdobným způsobem jsou data odeslána při změně na ScrollBaru pro vertikální polohu – ScrollBar2.

PC	uC	Popis
In_Packet0	In_Packet[1]	LowByte - požadovaná H poloha uC->PC
In_Packet1	In_Packet[2]	HighByte - požadovaná H poloha uC->PC
In_Packet2	In_Packet[3]	Požadavek od tlačítka H uC->PC
In_Packet3	In_Packet[4]	LowByte - požadovaná V poloha uC->PC
In_Packet5	In_Packet[6]	HighByte - požadovaná V poloha uC->PC
In_Packet4	In_Packet[5]	Požadavek od tlačítka V uC->PC
Out_Packet[0]	Out_Packet[1]	LowByte - požadovaná poloha H – PC->uC
Out_Packet[1]	Out_Packet[2]	HighByte - požadovaná poloha H – PC->uC
Out_Packet[2]	Out_Packet[3]	Indikace spuštění programu na PC - PC->uC
Out_Packet[3]	Out_Packet[4]	LowByte - požadovaná poloha V - PC->uC
Out_Packet[4]	Out_Packet[5]	HighByte - požadovaná poloha V - PC->uC

Tabulka 15: Využití jednotlivých prvků polí na straně PC a  $\mu C$



Obr. 29: Vzhled ovládacího programu

## 5 Závěr

Byly zkonstruovány a otestovány vhodné budiče pro řízení krokových motorů včetně návrhu a realizace řídicí desky s možností připojení nadřazeného systému pomocí USB. Ovládání je funkční z vytvořeného programu na PC i ovládacích tlačítek přímo na řídicí desce. Elektronika byla otestována na mechanické konstrukci rotátoru, kterou realizoval Ing. Jan Kovář.

V průběhu vývoje byly vyřešeny dílčí problémy s komunikací prostřednictvím USB. Vyskytly se především problémy s instalací knihoven [14] na PC s OS Windows Vista. Pomohlo až vypnutí nástroje „Řízení uživatelských účtů“ ve Windows Vista a spuštění instalace knihoven [14] s administrátorskými právy.

Dále při testování docházelo k destrukci řídicího obvodu budiče krokových motorů A3986 při odpojení silového napájení před napájením připojeného  $\mu$ C. V [5] není bohužel zmínka o potřebné přepětíové ochraně na silové napájecí větvi u H můstků. Transily se spínacím napětím 33V připojené na silové napájení problém vyřešily beze zbytku.

Díky profesionální výrobě plošných spojů nebyly problémy s oživením a jak driver, tak řídicí deska po pečlivém osazení fungovali na první zapojení až na výše uvedený problém s přepětím, které vznikalo na cívkách motoru.

V budoucnu je díky zdokumentovanému software možno vytvořit ovladač pro zahrnutí rotátoru do automatického měřicího pracoviště a směrové charakteristiky měřit automatizovaně s využitím např. Agilent VEE.

## 6 Použitá literatura

- [1] Kamil Řezáč, Krokové motory [Online] [Cit. 25.4.2009], dostupné z URL: <http://robotika.cz/articles/steppers/cs>
- [2] VŠB, Akční členy, [Online] [Cit. 25.4.2009], dostupné z URL: [http://e-automatizace.vsb.cz/ebooks/ridici\\_systemy\\_akcni\\_cleny/Akc\\_el.html](http://e-automatizace.vsb.cz/ebooks/ridici_systemy_akcni_cleny/Akc_el.html)
- [3] C8051F320/1 Datasheet, [Online] [Cit. 25.4.2009], dostupné z URL: <http://www.keil.com/dd/docs/datashts/silabs/c8051f32x.pdf>
- [4] C8051F320 - Mikroprocesory řady x51 s rozhraním USB, [Online] [Cit. 25.4.2009], dostupné z URL: <http://hw.cz/Produkty/Nove-soucastky/ART938-C8051F320---Mikroprocesory-rady-x51-s-rozhranim-USB.html>
- [5] A3986 Dual Full-Bridge MOSFET Driver with Microstepping Translator - Datasheet, [Online] [Cit. 25.4.2009], dostupné z URL: <http://www.allegromicro.com>
- [6] IRF 540 – Datasheet, [Online] [Cit. 25.4.2009], dostupné z URL: [http://www.datasheetcatalog.com/datasheets\\_pdf/I/R/F/5/IRF540.shtml&ei=4TL\\_Sfn4AZKxs\\_gac0fn9Ag&sa=X&oi=spellmeleon\\_result&resnum=1&ct=result&usg=AFQjCNHUshi1eURVnYTp4MzABYFvxPw7gg](http://www.datasheetcatalog.com/datasheets_pdf/I/R/F/5/IRF540.shtml&ei=4TL_Sfn4AZKxs_gac0fn9Ag&sa=X&oi=spellmeleon_result&resnum=1&ct=result&usg=AFQjCNHUshi1eURVnYTp4MzABYFvxPw7gg)
- [7] Pohony s krokovými motory, [Online] [Cit. 30.4.2009], dostupné z URL: [http://fe1.vsb.cz/kat453/www453/soubory/texty/ucebni\\_texty/se/cast\\_C\\_el\\_pohony/se\\_eph\\_c1\\_kroac\\_02\\_teorie.pdf](http://fe1.vsb.cz/kat453/www453/soubory/texty/ucebni_texty/se/cast_C_el_pohony/se_eph_c1_kroac_02_teorie.pdf)
- [8] C8051F32X DEVELOPMENT KIT USER'S GUIDE Rev. 0.8 9/06 Silicon Laboratories, [Online] [Cit. 2.1.2010], dostupné z URL: <http://www.silabs.com>
- [9] AXELSON, Jan, USB COMPLETE FOURTH EDITION, Madison, Lakeview Research LLC, WI 53704, 529s., ISBN13 978-1-931448-08-6
- [10] USB 2.0 specification, [Cit. 2.1.2010] [Online], dostupné z URL: [http://www.usb.org/developers/docs/usb\\_20\\_122909-2.zip](http://www.usb.org/developers/docs/usb_20_122909-2.zip)
- [11] Kolka, Zdeněk Prof. Dr. Ing., MPOA Přednáška č. 5, interní materiál UREL
- [12] Transmissive Optical Sensor with Phototransistor Output, Vishay, [Cit. 17.5.2010] [Online], dostupné z URL: [http://zefiryn.tme.pl/dok/a06/tcst1103\\_1202\\_2103.pdf](http://zefiryn.tme.pl/dok/a06/tcst1103_1202_2103.pdf)
- [13] BURKHARD, M. C pro mikrokontroléry. Praha : BEN - technická literatura, 2003. ISBN: 80-7300-077
- [14] JEDI Visual Component Library, [Cit. 18.5.2010] [Online], dostupné z URL: <http://jvcl.delphi-jedi.org/>

## **7 Přílohy**



## **7.1 Elektronická příloha**

Součástí diplomové práce je elektronická příloha na CD.

Struktura adresáře na CD je následující:

DP_xpuhon00.pdf	vlastní text diplomové práce
software.zip	software pro PC a $\mu$ C