

České vysoké učení technické v Praze

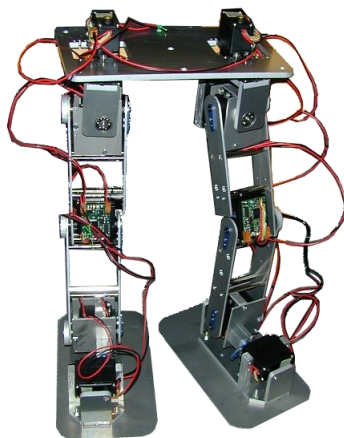
Fakulta elektrotechnická

katedra řídicí techniky



Řízení servomotoru kráčejího robota

bakalářská práce



Vypracoval: Martin Rakovec
Vedoucí práce: ing. Marek Peca

Praha 2009

Prohlášení

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

Dne 27.7.2009

Martin Rakovec

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Rakovec**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Řízení servomotoru krácejícího robota**

Pokyny pro vypracování:

1. Navrhněte novou verzi elektroniky pro řízení servomotorů robota Spejbl.
2. Implementujte regulátor tohoto servomotoru s důrazem na omezení proudu.
3. Vytvořte software pro počítač s OS Linux a řídicí mikropočítač servomotoru komunikující po sběrnici CAN pro ověření činnosti regulátoru.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Marek Peca

Platnost zadání: do konce zimního semestru 2009/10

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



doc. Ing. Boris Šimák
doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Abstrakt

Hlavním cílem této bakalářské práce bylo navrhnout novou verzi výkonové a měřicí elektroniky pro servomotor robotu Spejbla. Během práce byla otestována knihovna implementující programátorské rozhraní VCA na mikropočítači. Pro tento účel vznikl software na testování, který odhalil nedostatky knihovny a software, který ukazuje správné použití knihovny. Zároveň byla knihovna zdokumentována. Byla navržena nová verze výkonové a měřicí elektroniky, která využívá k měření proudu snímací rezistory a přístrojový zesilovač k odečítání úbytků napětí. Měření proudu je díky posunu nuly výstupu zesilovače bipolární. Vinou nesprávné volby přístrojového zesilovače má tato elektronika jisté nedostatky. Nepodařilo se doladit regulátor proudu a příčina špatné regulace proudu nebyla odhalena.

Abstract

Main goal of this bachelor thesis was propose new version of power and measuring electronics for servomotor of the robot Spejbl. Within the work there was tested library which implements programming interface VCA on microcomputer. For this purpose arose software for testing which discovered failings of this library and software which shows correct use of this library. Library was documented as well. New version of power and measuring electronics was proposed. That using sensing resistors and instrumentation amplifier for subtraction of voltage drop. Current measurement is bipolar owing to pseudoground on the amplifier output. This electronics has some flaws due to incorrect choice of instrumentation amplifier. Current controller tuning was unsuccessful and reason of bad current control wasn't discovered.

Poděkování

Chtěl bych poděkovat všem mým blízkým za jejich nevyčerpatelnou podporu. Dále mé poděkování patří všem kolegům studentům a učitelům za jejich věcné rady a připomínky. V neposlední řadě mé poděkování patří Ing. Marku Pecovi za poučení a vedení práce.

Obsah

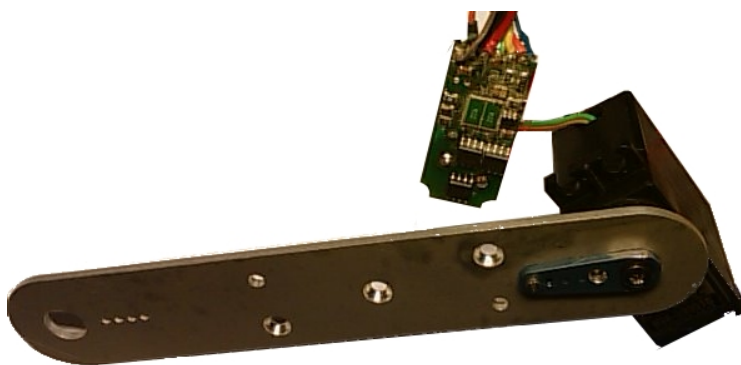
1	Úvod	3
2	Použité desky s mikropočítačem	4
2.1	Deska Spejbl-ARM	4
2.2	Deska lpcEurobot	4
3	CAN	5
3.1	Knihovna lpcanvca.h	6
3.2	Zprovoznění komunikace přes CAN	7
3.3	Testování komunikace přes CAN	8
3.4	Ukázkové použití VCA na mikropočítači	8
3.4.1	Zahrnuté knihovny	9
3.4.2	Definice proměnných	9
3.4.3	Inicializace sběrnice	9
3.4.4	Přijímání a vysílání zpráv	10
3.4.5	Překlad programu pomocí MakeOMK	10
4	Upravená verze softwaru pro mikropočítač	11
4.1	Řídící program servomotoru	11
4.2	Program pro hodiny	12
5	Výkonová a měřicí elektronika	14
5.1	Testování původní elektroniky	14
5.1.1	Testování celé regulační smyčky polohy	14
5.1.2	Testování regulační smyčky proudu motorem	15
5.1.3	Vyhodnocení testů	16
5.2	Nová verze výkonové a měřicí elektroniky	17
5.2.1	Měření proudu	17
5.2.1.1	Nedostatek návrhu	20
5.2.2	Úprava softwaru pro mikropočítač	21
5.2.3	Testování nové verze elektroniky	21
5.2.3.1	Zhodnocení	24
6	Závěr	26
6.1	Dosažené výsledky	26
6.2	Náměty na pokračování	26
A	Nová verze výkonové a měřicí elektroniky	28
A.1	Schéma elektroniky	28
A.2	Schéma upravené elektroniky	29
A.3	Obrazec plošného spoje	30
A.4	Osazovací plán	30

1 Úvod

Tato práce navazuje na diplomovou práci o kráčeícím robotu [1]. Hlavním cílem bylo navrhnout novou verzi výkonové a měřicí elektroniky servomotorů pro dvounohého robota Spejbla. Důraz byl kladen hlavně na zlepšení kvality měření proudu servomotorem. U původní elektroniky způsobovalo nekvalitní měření proudu kmitání kloubů robota v klidovém stavu tj. při nulovém proudu. Před návrhem nové elektroniky bylo nutné nejprve ověřit, zda je tento problém kmitání opravdu způsoben špatným měřením proudu a potvrdit tak závěry uvedené v [1, 6.3.3.2].

Dále si tato práce kladla za cíl úpravu stávajícího softwaru řídicího mikropočítače servomotorů. Jednalo se zejména o přepsání stávající komunikace přes průmyslovou sběrnici CAN¹ s využitím programátorského rozhraní VCA² implementovaného v knihovně *lpcanvca.h*. Hlavním důvodem pro tento krok bylo zjednodušení kódu a zlepšení jeho přehlednosti tak, aby byla případná následná práce na robotu jednodušší. Zároveň bylo nutné prozkoumat hlavní řídicí program pro PC, který si vyžádal malé úpravy pro použití jen s jedním servomotorem.

S touto úpravou přibyla nutnost zprovoznit a otestovat sestavu notebook, převodník USB-CAN a CAN rozhraní mikropočítače. Hlavními problémy, které si vyžádaly mnoho času a úsilí, bylo nastavení rychlosti komunikace na straně mikropočítače a celkově podivuhodné chování celé sestavy. Výsledkem byla implementace testovacích programů na množství ztracených zpráv při přenosu přes průmyslovou sběrnici a implementace ukázkového programu na správné využití programátorského rozhraní VCA na straně mikropočítače pro komunikaci přes průmyslovou sběrnici. Při těchto testech se projevilo několik nedostatků knihovny pro programátorské rozhraní VCA na straně mikropočítače jako jsou ztráty zpráv při plném provozu. Tyto nedostatky jsou pro účely této bakalářské práce nepodstatné.



Obrázek 1: Servomotor s řídicí elektronikou

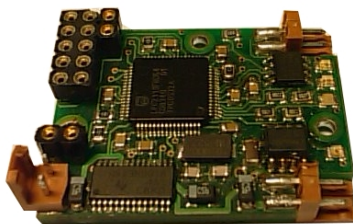
¹CAN - Controller Area Network.

²VCA - Virtual CAN Application Programming Interface (API - rozhraní pro programování aplikací).

2 Použité desky s mikropočítačem

V rámci této práce bylo nutné použít minimálně 2 desky s mikropočítačem a vyvedenou průmyslovou sběrnici CAN a řídicí počítač připojený také na sběrnici CAN. V našem případě zastává úlohu řídicího počítače průměrný notebook (Acer Aspire 3100) s operačním systémem Linux (Ubuntu 8.10). První deska s mikropočítačem je používána jako řídicí deska servomotoru. Tato deska zastává funkci regulátoru proudu a je propojena s měřicí a výkonovou elektronikou. Druhá deska slouží jako zdroj hodin pro sběrnici CAN.

2.1 Deska Spejbl-ARM



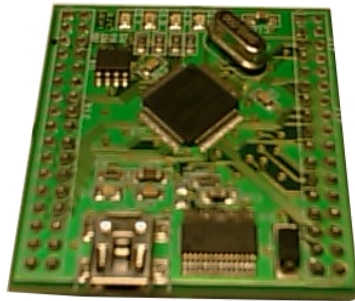
Obrázek 2: Spejbl-ARM deska pro řízení servomotoru.

Tato deska byla navržena přímo pro řízení jednoho servomotoru robotu. Deska je osazena 32-bitovým mikropočítačem s jádrem ARM a s vyvedenou průmyslovou sběrnici CAN. Dále jsou vyvedeny vývody, potřebné na propojení desky *Spejbl-ARM* s výkonovou a řídicí elektronikou. Podrobnější popis desky je uveden v [1, 4.2]. Tato deska byla použita jako řídicí deska servomotoru.

Programování mikropočítače probíhá přes sériové rozhraní, které bylo z důvodu absence sériového portu na notebooku nahrazeno převodníkem USB-UART (obvod FT232R [5]). Samotné nahrávání programu bylo prováděno za pomoci systému *makeOMK*. V repositáři *sysless-lpc21xx* bylo nutné provést několik změn v konfiguračním souboru pro *Spejbl-ARM* desku, protože původní verze nastavení v tomto souboru neumožňovala programování desky *Spejbl-ARM* výše uvedeným způsobem a pro první pokusy byl používán konfigurační soubor pro desku *lpcEurobot*, což nebylo zrovna optimální řešení. Původní konfigurační soubor umožňoval programování mikropočítače jen přes sériové rozhraní počítače.

2.2 Deska lpcEurobot

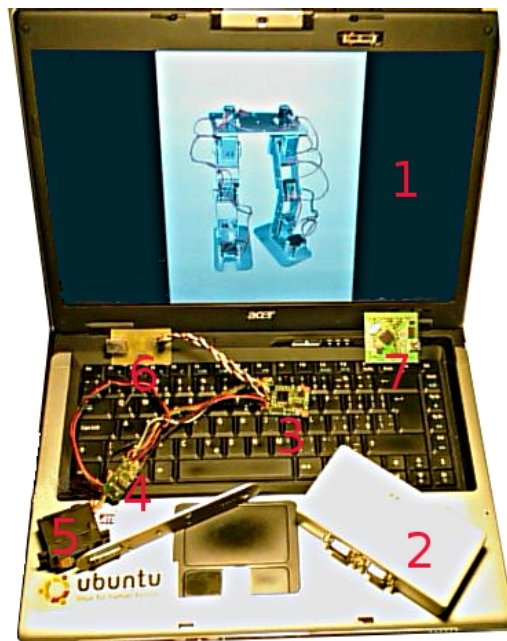
Tato deska je vybavena stejným mikropočítačem jako deska *Spejbl-ARM* popsaná výše. Jedná se v podstatě o univerzální mikropočítačovou desku, která má vyvedeny téměř všechny vývody mikropočítače na kolíkové lišty. Podstatným rozdílem oproti *Spejbl-ARM* desce je, že tato deska obsahuje převodník USB-UART a je napájena přímo z počítačového portu USB. Programování mikropočítače tedy probíhalo stejně jako u desky *Spejbl-ARM*. Tato deska byla použita jako zdroj hodin na sběrnici CAN.



Obrázek 3: Deska lpcEurobot pro CAN hodiny.

3 CAN

Pro veškerou práci bylo nutné nejprve zprovoznit sestavu notebook, USB-CAN převodník a deska s mikropočítačem a vyvedenou sběrnici CAN (viz. obrázek 4). Převodník USB-CAN je podporován ovladačem *OCERA LinCAN³ v0.3.3* a zajišťuje tedy plnou kompatibilitu s původním softwarem pro robota Spejbla. Pro komunikaci přes sběrnici CAN bylo využito programátorské rozhraní VCA. Toto rozhraní je na straně mikropočítače implementováno v knihovně *lpcanvca.h*, která byla hlavním cílem testování.



Obrázek 4: Sestava pro testování řízení servomotoru. 1 - notebook, 2 - převodník USB-CAN, 3 - deska Spejbl-ARM, 4 - deska výkonové a měřicí elektroniky, 5 - servomotor, 6 - převodník USB-UART, 7 - deska lpcEurobot

³LinCAN - Linux CAN driver

3.1 Knihovna `lpcanvca.h`

Tato knihovna je součástí repositáře `sysless-lpc21xx` na webu `rtime`⁴. Knihovna implementuje programátorské rozhraní VCA pro mikropočítač LPC. Jedná se o napodobeninu programátorského rozhraní VCA z varianty pro počítač. Funkce knihovny jsou v podstatě stejné jako u varianty pro počítač (popsáno v [2, 3.1]. Zde je uveden seznam funkcí a krátký popis jejich funkce a parametrů:

- `int lpc_vca_open_handle(vca_handle_t *vcah_p, int dev, int flags, uint32_t btr, unsigned rx_ivect, unsigned tx_ivect, unsigned err_ivect)`
 - Funkce pro inicializaci sběrnice a vytvoření nového handleru.
 - Funkce vrací `VCA_OK` pokud je vše v pořádku.
 - `*vcah_p` – ukazatel na místo, kam bude uložen nový VCA handler.
 - `dev` – číslo CAN kontroléru (0, 1, ...).
 - `flags` – číslo upravující režim otevření CAN (např. `VCA_O_NOBLOCK`).
 - `btr` – registr časování sběrnice (**B**it **T**iming **R**egister).
 - `rx_ivect` – číslo vektoru přerušení pro příjem zpráv.
 - `tx_ivect` – číslo vektoru přerušení pro vysílání zpráv.
 - `err_ivect` – číslo vektoru přerušení pro chyby.
- `int vca_close_handle(vca_handle_t vcah)`
 - Funkce pro uzavření handleru pro sběrnici CAN.
 - Funkce vrací `VCA_OK` pokud vše proběhlo v pořádku.
 - `vcah` – handler pro sběrnici CAN.
- `int vca_send_msg_seq(vca_handle_t vcah, canmsg_t *msg, int count)`
 - Funkce pro vyslání zpráv na sběrnici.
 - Vrací počet úspěšně odeslaných zpráv.
 - `vcah` – VCA handler.
 - `*msg` – ukazatel na zprávu pro odeslání.
 - `count` – délka vysílané zprávy (počet bytů).

⁴`git://rtime.felk.cvut.cz/sysless-lpc21xx.git` .

- `int vca_rec_msg_seq(vca_handle_t vcah, canmsg_t *msg, int count)`
 - Funkce pro příjem zpráv ze sběrnice.
 - Vrací počet přijatých zpráv.
 - `vcah` – VCA handler.
 - `*msg` – ukazatel na přijatou zprávu.
 - `count` – délka přijaté zprávy (počet bytů).

- `vca_wait(vca_handle_t vcah, int wait_msec, int what)`
 - Funkce pro čekání (blokující) na příjem/odeslání zprávy.
 - Vrací kladné číslo jestliže je vše v pořádku.
 - `vcah` – handler pro sběrnici CAN.
 - `wait_msec` – čekání v milisekundách (0 = nekonečno).
 - `what` – 0,1 = čekat na odeslání, 2 = čekat na příjem, 3 = čekání na obojí.

Využití funkcí je patrné v programu, který ukazuje správné použití *lpcanvca.h*. Tento program je popsán v kapitole 3.4.

3.2 Zprovoznění komunikace přes CAN

Pro spojení převodníku s notebookem bylo nutné nainstalovat ovladač *OCERA LinCAN v0.3.3*. Po připojení převodníku USB-CAN k notebooku je vždy nutné zavést ovladač (*lincan.ko*) do jádra systému pomocí *insmod*. Před odpojením převodníku je nutné vždy nejprve ovladač odebrat z jádra pomocí *rmmmod*, jinak dojde k „zamrznutí“ systému. Dále byla nainstalována komponenta CAN *OCERA CANVCA v1.0*.

Ověření základní funkčnosti převodníku bylo provedeno vysláním zpráv na sběrnici CAN pomocí jednoduchého programu *rtx*, který je součástí *LinCAN*. Při vysílání resp. přijímání zpráv svítí na převodníku příslušná LED dioda. Při zavádění ovladače do jádra je možné nastavit parametr *processlocal* na hodnotu 2, což aktivuje tzv. *echo*. Převodník poté přijímá i zprávy, které vyslal počítač. Je tedy možné ověřit funkčnost převodníku pomocí vysílání a zpětného přijímání zpráv.

Na straně mikropočítače vznikl problém s nastavením komunikační rychlosti sběrnice. Nastavení rychlosti sběrnice na straně mikropočítače se provádí nastavením správného časování sběrnice v registru *CANBTR*⁵. Pro výpočet hodnoty registru *CANBTR* byla použita funkce *lpcan_btr* (součást knihovny *lpcan.h*), která má jako jeden z parametrů hodinovou frekvenci mikropočítače. Při zadání frekvence 10MHz (frekvence krystalu na *Spejbl-ARM* desce) komunikace neprobíhala a signály na sběrnici (vodiče *CAN_H* a *CAN_L*) byly velmi zdeformované.

⁵CANBTR - CAN Bus Timing Registry. Registr pro časování sběrnice CAN.

Pokusně byla nastavena frekvence na hodnotu 20MHz. Poté již komunikace probíhala. Příčinou tohoto problému bylo implicitní nastavení hodnoty registru *VPBDIV*⁶ na 2, což způsobí, že hodinová frekvence mikropočítače je dvojnásobkem frekvence krystalu. Tato chyba byla odhalena výpočtem *CANBTR* pro několik různých frekvencí a porovnáním vypočtených hodnot. Tento problém byl tedy jednoduše vyřešen nastavením *VPBDIV = 1*.

3.3 Testování komunikace přes CAN

Aby bylo možné v této práci využívat knihovnu *lpcanvca.h*, která slouží jako VCA pro mikropočítač, bylo nutné nejprve otestovat její funkčnost. Jako jednoduchý test byl implementován program pro PC a pro mikropočítač, který testuje kvantitativní ztrátu zpráv v obou směrech. Program na PC vyšle jednu zprávu, která stanoví začátek přenosu (dále jen START) a poté vyšle stanovený počet zpráv o délce jednoho bytu po sběrnici k mikropočítači. Tato zpráva obsahuje hodnoty v rozsahu 0x00 až 0xFF. V přijímači (nyní mikropočítač) se zaznamenává počet zpráv a po přijetí zprávy oznamující konec testu (dále jen STOP) vyšle počet přijatých zpráv zpět. Vysílač (nyní PC) vyšle potvrzení o přijetí (dále jen ACK).

Po přijetí ACK se směr testování obrátí. Test probíhá obdobně. Vysílač (nyní LPC) začne vysílat zprávy hned po přijetí ACK a po odeslání stanoveného počtu zpráv odešle STOP. Toto způsobí ukončení načítání zpráv na straně přijímače a vypsání výsledků testů na monitoru počítače.

Při těchto testech bylo zjištěno, že při vysílání zpráv bez prodlevy mezi jednotlivými zprávami je průměrná ztráta ve směru z notebooku do mikropočítače 77% (dorazilo 230 zpráv z 1000) a z mikropočítače do notebooku 56% (dorazilo 440 zpráv z 1000). Vzhledem k provozu na sběrnici CAN v této práci (frekvence hodin 250Hz) není tento problém pro další práci podstatný. Tato chyba je pravděpodobně způsobena chybnou implementací kruhových bufferů na straně mikropočítače. Odstranění této chyby však nebylo náplní této práce.

3.4 Ukázkové použití VCA na mikropočítači

Pro následnou práci s knihovnou *lpcanvca.h* byl implementován jednoduchý program, který ukazuje správné použití této knihovny. Program je nazvaný *CAN_HelloWorld*. Program plní jen jednoduchou funkci snímače napětí. Pokud je přijata zpráva s *ID* 0x00 resp. 0x01, program odešle hodnotu naměřenou na převodníku *A/D0* resp. *A/D1* (vyslaná zpráva má stejné *ID* jako zpráva přijatá). Pokud má přijatá zpráva jiné *ID*, program odešle zprávu s hodnotou a *ID* 0xEE. Zde uvádím jen jednoduchý popis jednotlivých částí programu. Kompletní program je v příloze této práce.

⁶VPBDIV - VPB Divider

3.4.1 Zahrnuté knihovny

Pro správnou funkci programu je nutné do programu zahrnout (část *include*) následující knihovny:

- `#include <can/canmsg.h>`, která definuje strukturu CAN zprávy.
- `#include <can/lpcan.h>`, která obsahuje funkci `lpcan_btr` pro výpočet hodnoty registru *CANBTR*.
- `#include <can/lpcanvca.h>`, která implementuje programátorské rozhraní VCA pro mikropočítač.

3.4.2 Definice proměnných

Zde jsou uvedeny proměnné, které využívá program pro programátorské rozhraní VCA:

- `vca_handle_t can`; – zde bude uložen handler sběrnice CAN.
- `can_msg_t msg`; – struktura pro CANovou zprávu.
- `uint32_t btr`; – zde bude vypočtená hodnota registru *CANBTR*.

3.4.3 Inicializace sběrnice

Než přistoupíme k inicializaci průmyslové sběrnice CAN, potřebujeme nejprve vypočítat správnou hodnotu registru *CANBTR*. Hodnota tohoto registru závisí na hodinové frekvenci procesoru, na požadované rychlosti sběrnice, *SJW*⁷, *Sample Point*⁸ a *SAM*⁹. Tuto hodnotu můžeme vypočítat pomocí funkce:

```
lpcan_btr(&btr, 1000000/*Bd*/, 10000000/*Hz*/, 0/*SJW*/, 70/*Sample Point*/,
         0/*SAM*/);
```

pro hodinovou frekvenci 10MHz a rychlost 1MB. Pokud je použit krystal 10MHz a počítáme s hodinovou frekvencí mikropočítače 10MHz, je nutné nastavit registr *VPBDIV* na hodnotu 1 (standardně nastaven na hodnotu 2). Při znalosti hodnoty registru *CANBTR* můžeme přistoupit k inicializaci sběrnice. Inicializace sběrnice probíhá následujícím voláním funkce:

```
lpc_vca_open_handle(&can, 0/*device*/, VCA_0_NONBLOCK/*flags*/, btr,
                  10/*rx_ivect*/, 11/*tx_ivect*/, 12/*err_ivect*/);
```

pokud vše proběhne v pořádku (funkce vrátí *VCA_OK*), bude v proměnné *can* handler pro přístup ke sběrnici CAN, který využívají ostatní funkce uvedené níže.

⁷SJW - Synchronization Jump Width, viz. [6, str. 285]

⁸Sample Point - viz. *CANESG1* a *TESG2* [6, str. 285]

⁹SAM - viz. [6, str. 285]

3.4.4 Přijímání a vysílání zpráv

Pro přijímání zpráv využijeme funkci `vca_rec_msg_seq` v následujícím tvaru:

```
vca_rec_msg_seq(can, &msg, 1);
```

Tato funkce vrací počet přijatých zpráv, které budou v proměnné `msg`. V ukázkovém programu je tato funkce volána vždy na začátku hlavní smyčky programu v podmínce `if`, kde se vyhodnotí počet přijatých zpráv. Pokud není přijata žádná zpráva, program přistoupí k dalšímu cyklu (`continue;`). V opačném případě se přistoupí k vyhodnocení zprávy (viz. 3.4).

Pro vysílání zpráva využijeme funkci `vca_send_msg_seq` v následujícím tvaru:

```
vca_send_msg_seq(can, &msg, 1);
```

Tato funkce vrací počet úspěšně odeslaných zpráv. Pro nastavení *ID* a samotných dat zprávy využijeme prvky `.id` a `.data[n]` struktury `can_msg_t`. Využití této funkce v programu je popsáno v 3.4.

3.4.5 Překlad programu pomocí MakeOMK

Pro překlad programů ,které využívají výše uvedené knihovny, za pomoci systému *MakeOMK* je nutné přidat do souboru *Makefile.omk* řádek:

```
jmeno_programu_LIBS = lpcanvca lpcan,
```

4 Upravená verze softwaru pro mikropočítač

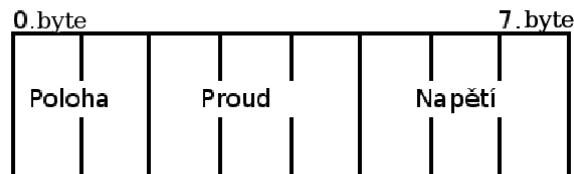
V původním softwaru pro mikropočítač byl přístup k průmyslové sběrnici CAN řešen pomocí knihovny *lpcan.h*, která umožňuje přístup ke sběrnici na nízké úrovni (na úrovni ovladače). To znamená, že bylo nutné obsluhovat přerušení jako je například přerušení při příjmu zpráv. Tento způsob je poměrně složitý a kód je z tohoto důvodu nepřehledný. Pro zjednodušení kódu a zvýšení jeho přehlednosti jsem přistoupil k úpravě původního kódu. Jednalo se tedy konkrétně o přepsání komunikace po průmyslové sběrnici CAN s využitím výše zmíněné knihovny *lpcanvca.h* s programátorským rozhraním VCA.

4.1 Řídící program servomotoru

U tohoto programu (*spejbl_motor/motorx.c*) se jednalo jen o malé úpravy původního kódu (*smotorx/motorx.c*). Protože s použitím programátorské rozhraní VCA už není třeba starat se o přerušení ohledně CAN (stará se o něj přímo programátorské rozhraní VCA), byl program upraven následovně:

- Příjem zpráv se testuje v hlavní programové smyčce.
 - Pokud byla přijata zpráva, proběhne volání funkce `can_rx()`, kde je přijatá zpráva vyhodnocena.
 - Pokud nebyla přijata zpráva, zavolá se funkce regulátoru `control(adc_i, adc_x)` (pokud je to třeba) a funkce pro odeslání zprávy `can_tx()` (pokud je to třeba).
- Funkce pro vyhodnocení přijatých zpráv vyhodnocuje celkem 3 případy:
 1. *ID* přijaté zprávy je rovno *ID* hodinové zprávy. V tomto případě se nastaví požadovaná hodnota proudu (*control_w*) na přijatou požadovanou hodnotu proudu (v proměnné *rx_cmd_value*), pokud tato hodnota nemá hodnotu *CTRL_OFF* tzn. vypnutí motoru. Zároveň je nastaven požadavek na vyslání zprávy řídicímu počítači (*tx_request*).
 2. *ID* přijaté zprávy je rovno *ID* konfigurační zprávy. Tato zpráva slouží k nastavení řídicí desky servomotoru. V této zprávě je uvedeno *ID* nastavované řídicí desky servomotoru, *ID* řídicí zprávy, kterou bude servomotor řízen z počítače a index, který označuje konkrétní část (první byt ze dvou) této řídicí zprávy, která bude určena pro tento servomotor (v jedné řídicí zprávě mohou být hodnoty až pro 4 motory najednou).
 3. *ID* přijaté zprávy je rovno *ID* řídicí zprávy. V tomto případě se ze zprávy vezme jen část určená výše zmíněným indexem a nastaví se do proměnné *rx_cmd_value*. Pokud má tato proměnná hodnotu *CTRL_OFF*, vypne se motor a jeho řízení (proměnná *control_on*).

- Funkce pro vysílání zpráv je zpracována jen v případě že je to požadováno programem ($tx_request = 1$). Pokud je požadavek splněn, je zpráva naplněna požadovanými hodnotami (viz. obrázek 5).



Obrázek 5: Struktura datové části CAN zprávy zasílané servomotorem

Při úpravě softwaru jsem narazil na chybu původního autora s přetypováním proměnných. Konkrétně šlo o přetypování 2 proměnných typu $uint8_t$ (2 byty zprávy CAN) do proměnné typu $uint16_t$. Přetypování tímto způsobem:

```
uint16_t v1 = ((uint16_t*)msg.data)[0]; //hodnota msg.data[0] a msg.data[1]
```

fungovala pouze tak, že v proměnné $v1$ bylo správně nastaveno jen horních 8 bitů a spodních 8 bitů bylo nastaveno na 0. Jedná se o chybu zarovnání tzv. *alignment*. Rozhodl jsem se tento problém řešit následujícím způsobem:

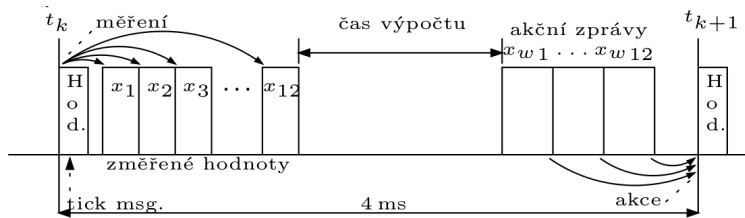
```
uint16_t v1 = msg.data[0] | ((msg.data[1]) << 8);
```

takto už přetypování fungovalo správně. Tato upravená verze softwaru (*spejbl_motor*) je v příloze této práce.

4.2 Program pro hodiny

Program plní jednoduchou funkci hodin pro průmyslovou sběrnici CAN. Funkce hodin je taková, že s frekvencí 250Hz vysílá na sběrnici CAN hodinové zprávy. Zpráva od hodin má délku 1 byte a obsahuje vždy pořadové číslo "tiku". Tento program využívá přerušování časovače mikropočítače. Program tedy při každé obsluze přerušování inkrementuje proměnnou udávající počet tiků a obsah této proměnné načte jako data do zprávy, kterou poté vyšle na sběrnici CAN.

Pro tento program (*spejbl_hodiny/hodiny.c*) bylo také využito programátorské rozhraní VCA s použitím knihovny *lpcanvca.h*. Sběrnice je otevírána v neblokujícím režimu. Tato nová verze softwaru pro hodiny neplní správně svou funkci. Při připojení na sběrnici CAN společně s notebookem (převodník USB-CAN) a řídicí deskou servomotoru se ztrácejí hodinové zprávy. Podle provozu na sběrnici CAN (viz. obrázek 4.2) je správné, když mezi dvěma hodinovými tiky jsou odeslány zprávy od všech motorů a vypočtené akční zásahy pro všechny motory.



Obrázek 6: Provoz na sběrnici CAN během jedné periody (převzato z [1, obr. 6])

S touto verzí softwaru provoz vypadá tak, že zpráva od motoru „překryje“ zprávu od hodin. Například je přijata zpráva hodinová zpráva s pořadovým číslem 1, poté zpráva od motoru a pak je přijata hodinová zpráva s pořadovým číslem 3. Hodinová zpráva s pořadovým číslem 2 se tedy ztratila.

Protože toto nebyl krok nutný pro pokračování v práci, rozhodl jsem se použít původní verzi softwaru pro hodiny robotu Spejbl, který fungoval bez úprav bezproblémově.

5 Výkonová a měřicí elektronika

Hlavním cílem této práce bylo navrhnout novou verzi výkonové a měřicí elektroniky servomotoru. S původní verzí této elektroniky vznikal problém kmitání v kloubech robotu při nulovém proudu motorem. Tento problém je podrobně popsán v [1, 6.3.3.2], kde je vyvozen závěr, že kmitání v kloubech robotu je způsobeno špatným měřením proudu. Z tohoto důvodu bylo nutné nejprve otestovat původní elektroniku na přítomnost těchto kmitů a potvrdit tak tyto záběry.

V původní verzí řídicí elektroniky je měřen proud do H-můstku pomocí snímacího odporu ($0,11\Omega$), který je zařazen mezi kladný pól napájecího zdroje a H-můstek. Na tomto odporu vzniká úbytek napětí úměrný protékajícímu proudu. Toto napětí je snímáno operačním zesilovačem a filtrováno RC-filtrem. Toto vyfiltrované napětí je snímáno A/D převodníkem mikropočítače na desce *Spejbl-ARM*.

Problémem je, že proud (úbytek napětí na odporu) je měřen unipolárně. Dále proud není měřen na takovém místě, že při odpojení motoru od kladného pólu zdroje (např. při vypnutí motoru se oba póly motoru připojí na záporný pól zdroje) zaniká možnost měření proudu motorem.

5.1 Testování původní elektroniky

Testování původní elektroniky probíhalo měřeními odezvy systému na skok požadované polohy resp. požadovaného proudu. Tyto odezvy jsou měřeny pomocí programu *servo_test*, který naměřená data ukládá do textového souboru. Data z tohoto souboru jsou vykreslována pomocí programu *Gnuplot*¹⁰.

5.1.1 Testování celé regulační smyčky polohy

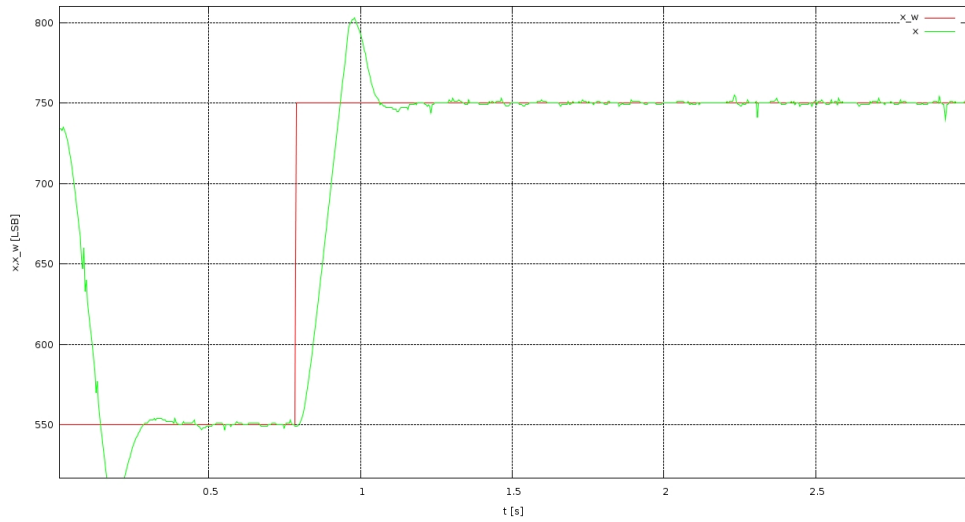
Jako první byla testována celá smyčka řízení polohy tzn. v činnosti je regulátor polohy (PID v notebooku) a regulátor proudu (PI v mikropočítači). Regulátor polohy má jako vstupní veličiny požadovanou polohu x_w a skutečnou (naměřenou) polohu x a jeho akční veličinou je proud motorem i_w . Regulátor proudu má jako vstupní veličiny požadovaný proud i_w a skutečný (naměřený) proud motorem i a jeho akční veličinou je napětí u (PWM¹¹). Toto napětí zapříčiní mimo jiné změnu polohy x a proudu i .

Odezva systému na skok požadované polohy x_w je znázorněna na obrázku 7. Zde nebyla hřídel zatížena vnější silou a projevil se zde kmit v klidové poloze (nulový proud). Tyto kmity se jeví jako náhodný šum.

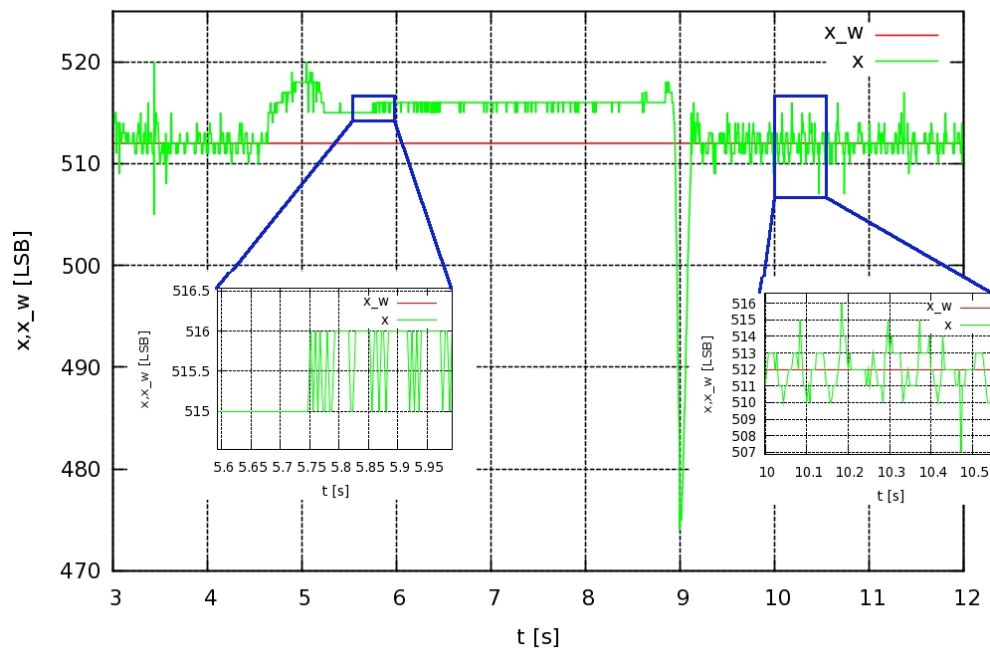
Na obrázku 8 je znázorněna odezva systému na zatížení hřídele motoru. Zatížení hřídele motoru probíhalo přidržetím páky, připevněné na hřídel, rukou tak, aby motorem protékal dostatečně velký proud. Z obrázku je vidět, že při zatížení hřídele (vychýlení od požadované polohy na grafu viz. detail) kmitání ustane.

¹⁰<http://www.gnuplot.info/>

¹¹PWM - Pulse Width Modulation - pulsně šířková modulace



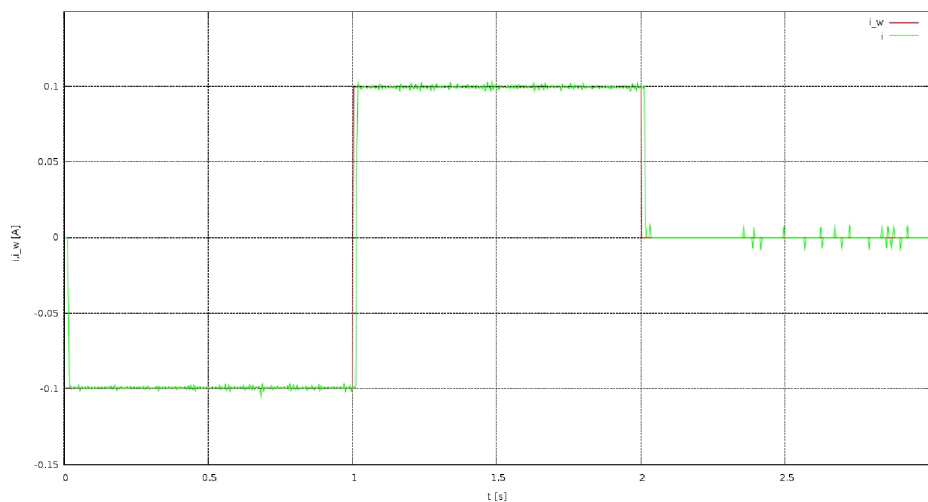
Obrázek 7: Odezva systému na skok požadované polohy - bez zátěže



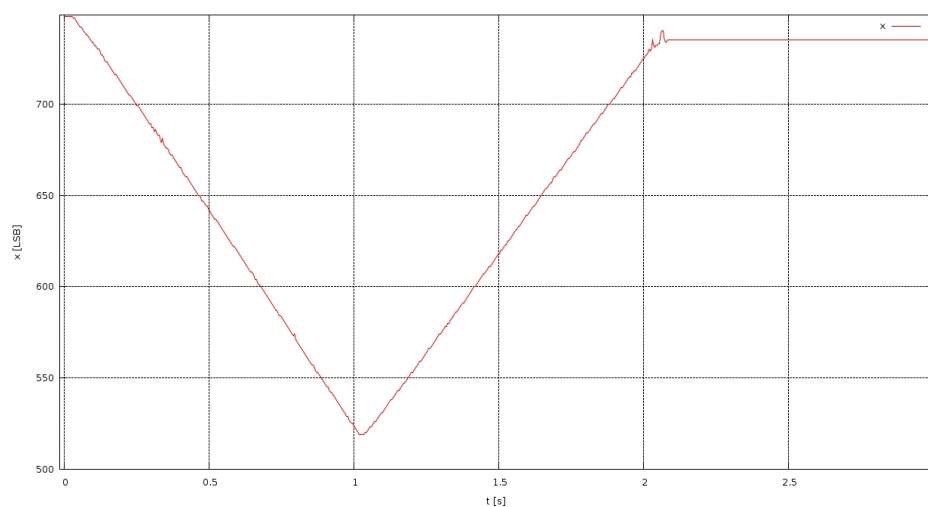
Obrázek 8: Odezva systému na zatížení hřídele motoru

5.1.2 Testování regulační smyčky proudu motorem

V této části testování původní elektroniky byl vyřazen regulátor polohy (PID) v řídicím počítači. Hodnota požadovaného proudu i_w byla zadávána ručně tak, aby bylo dosaženo skoku požadovaného proudu. Změnou hodnoty proudu se měnila rychlost otáčení hřídele motoru. Na obrázku 9 a 10 je odezva systému na skok požadovaného proudu i_w .



Obrázek 9: Odezva systému na skok požadovaného proudu - proud



Obrázek 10: Odezva systému na skok požadovaného proudu - poloha

Z obrázku 9 je vidět, že průběh měřeného proudu obsahuje také jistý šum. Tento šum je také způsoben s největší pravděpodobností špatným měřením proudu. Jak je vidět z obrázku 10, průběh polohy se mění konstantně a bez rušivých kmitů.

5.1.3 Vyhodnocení testů

V prvním testu byla potvrzena přítomnost kmitů kloubu v klidové poloze. Dále bylo ověřeno, že při zatížení hřídele motoru kmity téměř zmizí (viz. obrázek 8). Z výsledků testů, uvedených v kapitolách 5.1.1 a 5.1.2 tohoto dokumentu, a závěrů uvedených v [1, 6.3.3.2] je zřejmé, že problém kmitání v kloubech je opravdu způsoben špatným měřením proudu.

5.2 Nová verze výkonové a měřicí elektroniky elektroniky

Pomocí testů v podkapitole 5.1 byly potvrzeny závěry [1, 6.3.3.2], že kmitání hřídele servomotoru je opravdu způsobeno špatným měřením proudu. Je tedy nutné přistoupit k návrhu nové řídicí elektroniky servomotoru s ohledem na zadané požadavky na měření proudu. Hlavními požadavky bylo:

- zlepšit kvalitu měření proudu
- měřit proud bipolárně (se znaménkem)
- měřit proud na správném místě H-můstku

Z těchto požadavků tedy vyplývá několik řešení. Základem je měřit proud jako úbytek napětí na snímacím rezistoru mezi H-můstkem a minusovým pólem zdroje a to u obou větví H-můstku. Tyto naměřené úbytky napětí je poté nutné odečíst. Zde se nabízí dvě možnosti:

1. odečíst úbytky napětí elektronicky (analogově) a měřit jen jedním A/D převodníkem tento rozdíl napětí nebo
2. měřit dvěma A/D převodníky dva úbytky napětí a rozdíl provést až na softwarové úrovni (digitálně) v mikropočítači.

Já jsem zvolil první variantu. Výkonová část elektroniky zůstane téměř stejná jako v [1, 4.3.2]. Konceptuálně tedy tato část obsahuje výkonové spínací tranzistory (Si4532ADY) a budiče spínacích tranzistorů chráněné hradlem v diodové logice. Polovodičový teploměr již nebude v této verzi elektroniky zahrnut, protože v původní verzi nebyl téměř využíván.

5.2.1 Měření proudu

Proud by měl být měřen v obou větvích H-můstku a to v místě napojení H-můstku na záporný pól zdroje napětí. Zde jsou umístěny snímací rezistory o hodnotě $0,05\Omega$ (paralelní kombinace 2 rezistorů o velikosti $0,1\Omega$), na kterých v důsledku protékajícího proudu vzniká úbytek napětí. Tyto úbytky jsou snímány jedním přístrojovým zesilovačem (dále jen zesilovač), který zesiluje v ideálním případě jen rozdíl těchto úbytků (potlačení souhlasných signálů). Správná volba tohoto zesilovače byla velmi důležitá, protože cílem tohoto návrhu je zvýšit kvalitu měření proudu. Zesilovač by měl mít šířku pásma v jednotkách až desítkách kHz (díky vnitřní smyčce, která jede na 20kHz) při zesílení 10, dále by měl mít malý napěťový offset a teplotní drift a pokud možno co největší CMR¹². Výstupní napětí zesilovače je filtrováno RC článkem (odfiltrování vyšších harmonických) a snímáno A/D převodníkem mikropočítače.

¹²CMR - Common-Mode Rejection = Činitel potlačení souhlasných signálů.

Dalším cílem tohoto návrhu bylo zajistit bipolární měření proudu. To můžeme zajistit cíleným posunem nuly výstupního napětí zesilovače tak, aby nulovému proudu odpovídalo výstupní napětí úměrné polovině rozsahu A/D převodníku. Z tohoto důvodu je tedy nutné zajistit dostatečně přesný zdroj referenčního napětí úměrnému polovině rozsahu A/D převodníku. Toto napětí je přivedeno na referenční vstup zesilovače. U tohoto posunu může nastat problém s kolísáním nuly díky teplotě a podobným vlivům. To by však nemělo působit větší problémy, protože se jedná o chybu podobného rázu jako je například statické zatížení hřídele servomotoru a navíc se jedná o poruchu na vstupu, která by měla být vykompenzována integračními složkami regulátoru.

S ohledem na výše uvedené požadavky jsem zvolil přístrojový zesilovač s vestavěným zdrojem referenčního napětí INA125 od firmy Texas instruments. Výhodou této volby je minimum použitých součástek. Zde uvádím nejdůležitější parametry (podrobněji viz. [8]) tohoto zesilovače:

- Napájecí napětí $2,7V - 36V$ nebo $\pm 1,35V - \pm 18V$
- Zesílení $4 - 10000$
- Napěťový offset maximálně $\pm 500\mu V$
- Teplotní drift maximálně $\pm 5\mu V/^\circ C$
- CMR minimálně $72dB$ pro zesílení $G = 10$
- Klidový proud maximálně $\pm 5nA$
- Šířka pásma (bandwidth) pro zesílení $G = 10$ je $45 kHz$
- Reference napětí $1,24V, 2,5V, 5V, 10V$ s přesností $\pm 1\%$

S ohledem na dostupná referenční napětí jsem zvolil napětí $1,24V$ pro posun nuly výstupního napětí zesilovače na tuto napěťovou úroveň. Tímto posunem tedy získáme výstupní rozsah napětí zesilovače $2,48V$ ($\pm 1,24V$ proti umělé nule), takže se jedná o poměrně velký kompromis, protože vstupní rozsah A/D převodníku je $3,3V$, takže nebude využit celý rozsah A/D převodníku. Výhodou však zůstává, jak jsem již uvedl výše, minimální počet součástek a minimum využitého místa na desce plošného spoje. Při návrhu jsem zvolil rozsah měřeného proudu na $\pm 2,5A$. Protože regulátor proudu implementovaný v mikropočítači omezuje proud motorem na $\pm 1A$, měl by být tento rozsah měření proudu postačující na pokrytí špičkových hodnot měřeného proudu. Protože snímací rezistory mají hodnotu $0,05\Omega$, bude proud $2,5A$ odpovídat úbytek napětí:

$$U_{S_{2,5A}} = I_S \cdot R_S = 2,5 \cdot 0,05 = 0,125V. \quad (1)$$

Při návrhu jsem vycházel ze schématu uvedeného na obrázku 6 v [8, str. 13]. Pro výstupní napětí zesilovače platí rovnost:

$$U_O = (U_{S_1} - U_{S_2})G + U_{REF}, \quad (2)$$

kde

- U_{S_1} a U_{S_2} je napětí přivedené na vstupy zesilovače, tedy úbytky napětí na snímacích rezistorech
- U_O je výstupní napětí zesilovače
- U_{REF} je referenční napětí přivedené na odpovídající vstup zesilovače
- G je zesílení zesilovače (dále jen zesílení)

Pro zesílení platí vztah uvedený v [8, str. 10] jako rovnost (1), tedy:

$$G = 4 + \frac{60k\Omega}{R_G}, \quad (3)$$

kde R_G je externí rezistor, kterým lze nastavit zesílení.

Ze zvoleného rozsahu měření ($\pm 2,5A$), velikosti úbytku napětí na snímacím rezistoru $U_{S_{2,5A}}$ (viz. (1) a rozsahu výstupního napětí zesilovače $U_{O_{2,5A}}$ ($2,48V$ resp. $\pm 1,24V$ proti umělé nule) můžeme určit požadované zesílení takto:

$$G_P = \frac{U_{O_{2,5A}}}{U_{S_{2,5A}}} = \frac{1,24}{0,125} = 9,92 \simeq 10. \quad (4)$$

Zaokrouhlením byl rozsah měření proudu změněn na $\pm \frac{1,24}{10 \cdot 0,05} = \pm 2,48A$. Dále ze vztahu (3) určíme rezistor R_G (potřebný pro nastavení požadovaného zesílení, vypočteného ve (4)) takto:

$$R_G = \frac{60k\Omega}{G_P - 4} = \frac{60k\Omega}{10 - 4} = 10k\Omega. \quad (5)$$

Z naměřeného napětí (zesílený rozdíl úbytků napětí na snímacích rezistorech) v mikroočítači můžeme tedy určit proud servomotorem takto:

$$I = \frac{U_{S_1}}{R_{S_1}} - \frac{U_{S_2}}{R_{S_2}} = \frac{U_O}{G \cdot R_S}, \quad (6)$$

kde předpokládáme stejnou hodnotu snímacích rezistorů R_{S_1} resp. R_{S_2} .

V [8] je uvedeno, že hodnoty vnitřních rezistorů přístrojového zesilovače jsou absolutní. Z tohoto faktu je jasné, že přesnost měření proudu závisí zejména na přesnosti snímacích rezistorů, přesnosti rezistoru, určujícího zesílení přístrojového zesilovače, přesnosti referenčního zdroje napětí a na přesnosti A/D převodníku. Ostatní vlivy jsou zanedbatelné.

Přesnost měření proudu vyjádříme jako neurčitost nepřímého měření. Podle [3, 1.4.2] lze neurčitost nepřímého měření obecné veličiny Y , která je dána funkcí:

$$Y = f(X_1, X_2, \dots, X_n) \quad (7)$$

určit jako geometrický součet absolutních diferenciálů funkce f podle jednotlivých veličin X takto:

$$|u_Y| \leq \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial X_i} \Delta X_i \right)^2}. \quad (8)$$

Pravděpodobnou neurčitost měření proudu můžeme tedy podle vztahu (8) určit takto:

$$|u_I| = \sqrt{\left(\frac{\partial I}{\partial U_O} \Delta U_O \right)^2 + \left(\frac{\partial I}{\partial G} \Delta G \right)^2 + \left(\frac{\partial I}{\partial R_S} \Delta R_S \right)^2}. \quad (9)$$

Nepřesnost použitých rezistorů je 1% a tedy i nepřesnost zesílení (závisí jen na použitém externím rezistoru) bude také 1%. Nepřesnost napětí U_O uvažujeme $\pm 1LSB$, což je $\frac{100}{1024} = 0,098\%$. S využitím vztahu (6) dostáváme vyčíslenou **pravděpodobnou nepřesnost nepřímého měření proudu**:

$$|u_I| = \sqrt{\Delta U_O^2 + \Delta R_S^2 + \Delta G^2} = \sqrt{0,098^2 + 1^2 + 1^2} = \mathbf{1,148\%} \quad (10)$$

Schéma zapojení této verze elektroniky je v příloze A.1. Desku plošného spoje jsem zhotovil jako prototyp tak, že jsem neuvažoval rozměry umožňující osazení desky do krytu servomotoru, jak bylo zadáno vedoucím práce. Obrazec desky plošného spoje (v příloze A.3) byl vytištěn na obyčejný papír na laserové tiskárně a následně byl tepelnou cestou přenesen na cuprexitovou desku.

5.2.1.1 Nedostatek návrhu

Po prvním odzkoušení nové verze výkonové a řídicí elektroniky (viz. schéma v příloze A.1) jsem zjistil, že měření proudu není funkční. Výstup přístrojového zesilovače byl stále nulový. Důvodem bylo to, že jsem při návrhu opomenul omezení souhlasného vstupního napětí zesilovače. Podle [8, str. 12, odstavec *SINGLE SUPPLY OPERATION*] musí být souhlasné vstupní napětí zesilovače $> 150mV$ při jednoduchém napájení (V_{S-} připojeno na zem) a menší než $0,8 \cdot (V_{S+} - 0,8)$. Bylo tedy jasné, že podle mého návrhu se nachází vstup zesilovače mimo tyto meze.

Tento problém jsem se rozhodl dočasně vyřešit zařazením obyčejné usměrňovací diody mezi snímací rezistory a zem. Protéká-li diodou kladný nenulový proud, vzniká na diodě úbytek napětí přibližně $0,7V$ a tím je zajištěn odstup souhlasného napětí vstupujícího do přístrojového zesilovače. Jelikož spínací tranzistory mají v klidovém stavu velmi malý odběr a na diodě by nevznikl dostatečný úbytek napětí, je nutné přidat jeden rezistor, který zajistí dostatečný odběr proudu a tedy i požadovaný úbytek napětí na diodě. Tato úprava elektroniky je znázorněna ve schématu v příloze A.2. Toto řešení je ovšem pouze dočasné a z praktického hlediska nepoužitelné, protože na diodě vzniká velká výkonová ztráta, která se mění na teplo a navíc díky přidanému rezistoru elektronika odebírá zbytečně velký proud.

5.2.2 Úprava softwaru pro mikropočítač

Pro novou verzi výkonové a měřicí elektroniky je potřeba mírně upravit software pro mikropočítač, konkrétně přepočítat hodnoty naměřené A/D převodníkem. Původní verze elektroniky neměřila u proudu znaménko, kdežto u nové verze je proud měřen i se znaménkem posunutím nuly napětí, které je úměrné měřenému proudu. Je tedy nutné tento posun zahrnout do softwaru a získat tak údaj o směru protékajícího proudu.

Posun nuly byl programově realizován tak, že v softwaru mikropočítače je během obsluhy přerušování A/D převodníku (měření) odečítána hodnota odpovídající nule od naměřené hodnoty. Hodnota nuly je naměřena v začátku programu, dokud je motor vypnut. Tato hodnota je však pravdivá pouze pokud motor negeneruje proud (při pohybu).

Dále bylo nutné změnit multiplikační konstantu, která zajišťuje převod naměřené hodnoty na hodnotu proudu v ampérech. Tuto konstantu lze snadno určit z hodnoty snímacího rezistoru, zesílení přístrojového zesilovače a rozsahu A/D převodníku takto:

$$MUL = \frac{3,3}{1024 \cdot R_S \cdot G}. \quad (11)$$

Dále je třeba uvažovat že hodnota proudu je filtrována filtrem typu klouzavý průměr (*Moving Average - MA*), ale výsledná hodnota není vydělena počtem vzorků, aby nedošlo ke ztrátě přesnosti. Je tedy nutné zahrnout do multiplikační konstanty i dělení počtem vzorků v klouzavém průměru tedy (v našem případě $12 \cdot 80$).

5.2.3 Testování nové verze elektroniky

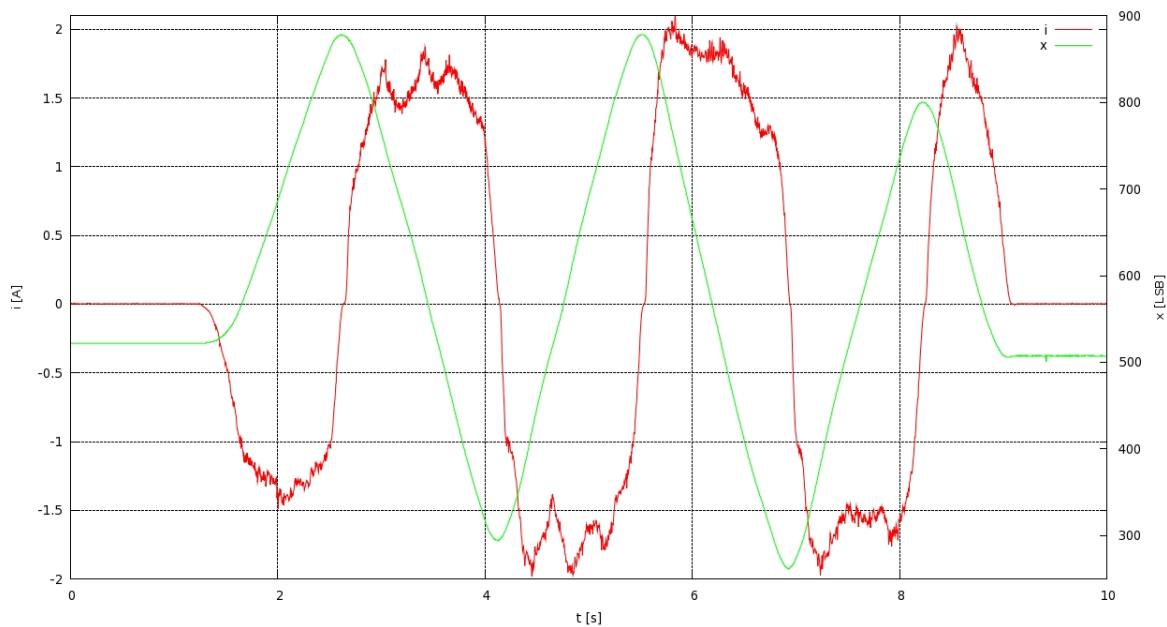
Pro dále uvedené grafy jen připomínám značení použitých veličin:

- i - měřený proud servomotorem
- i_w - požadovaný proud servomotorem
- x - měřená poloha hřídele servomotoru
- u - napětí (PWM) přiváděné na servomotor

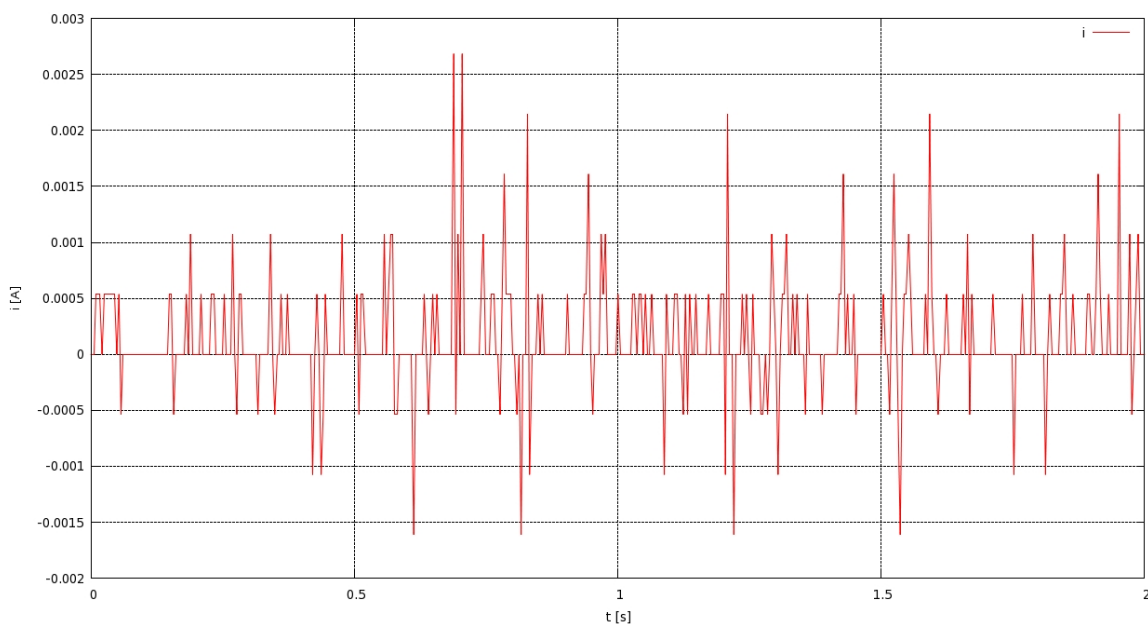
Jako první byla nová verze elektroniky testována v případě, kdy je motor ve zkratu (oba póly servomotoru připojeny na zem) a na hřídel servomotoru působí vnější síla. Tuto vnější sílu představoval pohyb hřídelí servomotoru rukou na obě strany s využitím přiměřené síly. Při tomto pohybu protéká motorem značný zkratový proud. Na obrázku 11 je znázorněn průběh proudu servomotorem i a průběh polohy x .

Obrázek 12 znázorňuje průběh měřeného proudu v nule. Je zde vidět rušivý šum, který je však poměrně malý v poměru k rozsahu ($\pm 2,5A$). Směrodatná odchylka šumu byla vypočtena pomocí funkce matlabu `STD` podle vzorce

$$\sigma = \sqrt{\frac{1}{N} \sum_{j=1}^N (i_j - \bar{i}_j)^2} = 4,494 \cdot 10^{-4} A \quad (12)$$



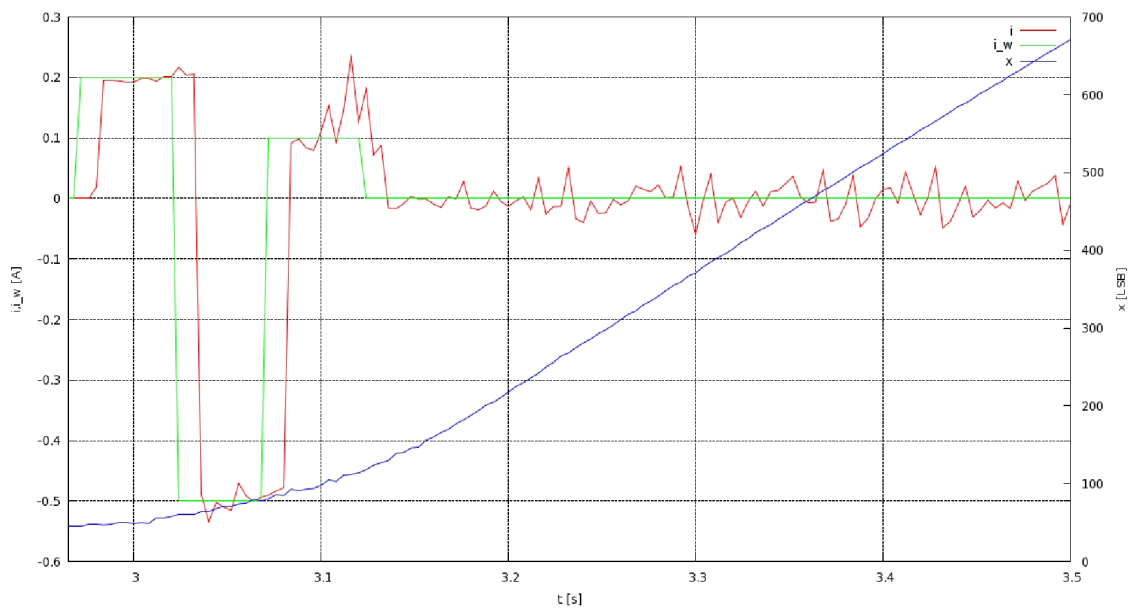
Obrázek 11: Průběh proudu a polohy při zkratovaném servomotoru za působení vnější síly.



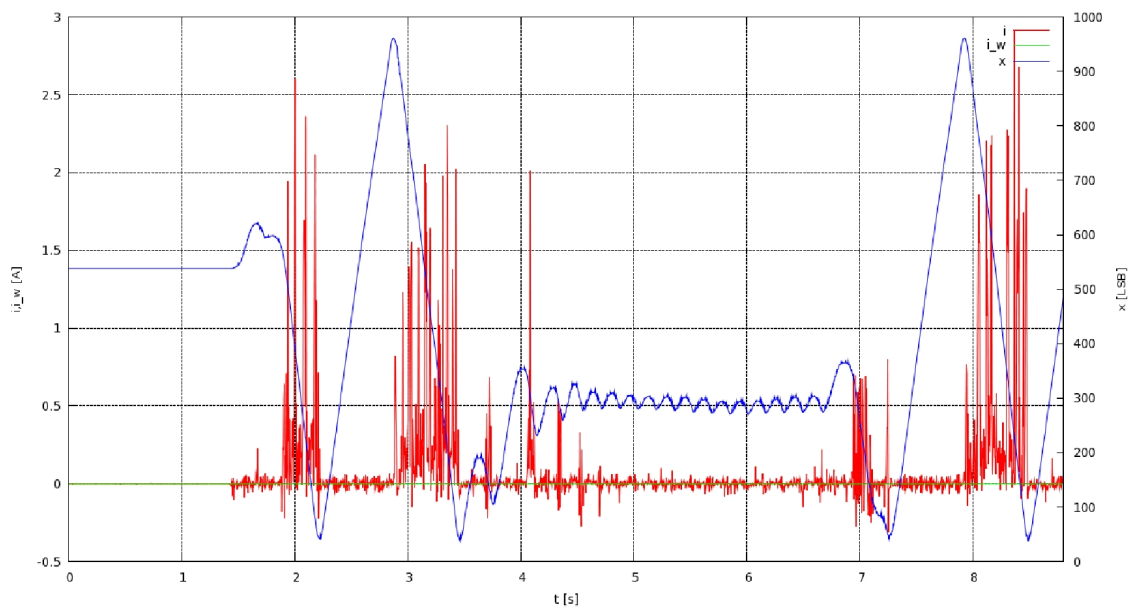
Obrázek 12: Průběh měřeného proudu v nule.

Další testy byly zaměřeny na regulátor proudu. Hlavním cílem těchto testů bylo vyladění regulátoru proudu. Jako první byl testován původní PI regulátor spolu s motorem. Reakce na skokové změny žádaného proudu je znázorněna na obrázku 13 a na obrázku 14 je znázorněn průběh měřeného proudu při regulaci na nulový proud při působení vnější síly.

Zde je zároveň zobrazen průběh polohy hřídele servomotoru. Celý systém se rozkmitá (páka na hřídeli střídá mezní hodnoty - dorazy) již při působení téměř nezatelné síly (dotek prstem ruky). Ve střední části průběhu je znázorněna reakce na pokus o zastavení kmitavého pohybu přidržetím páky na hřídeli servomotoru.

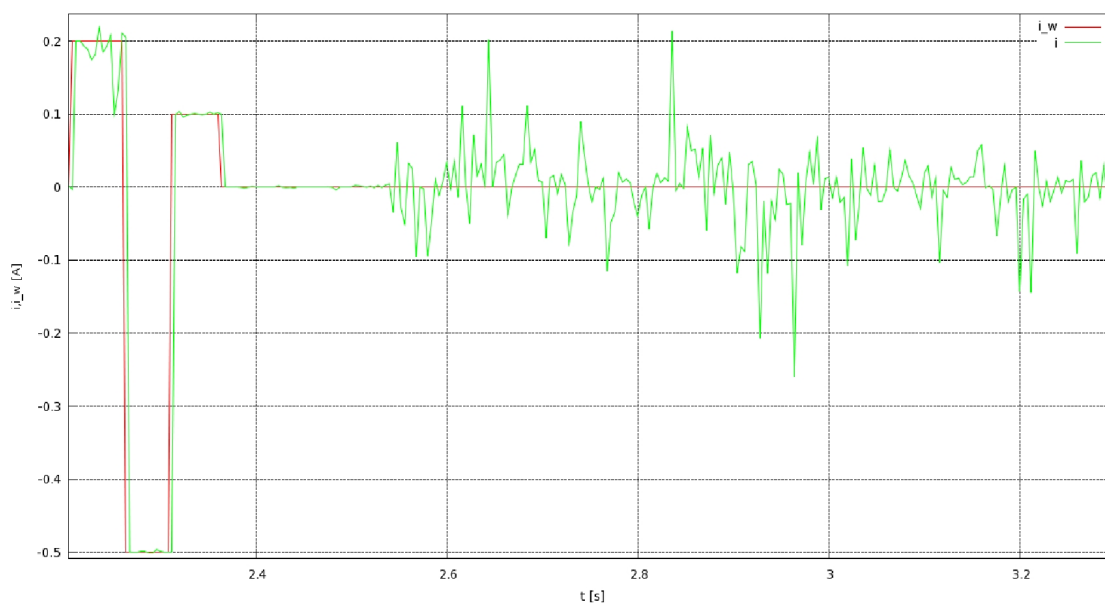


Obrázek 13: Reakce na skokovou změnu žádaného proudu - PI regulátor



Obrázek 14: Regulace na nulový proud při působení vnější síly - PI regulátor

Dále byl motor nahrazen výkonovým rezistorem o hodnotě 1Ω , aby byly odstraněny rušivé vlivy motoru (indukované napětí atd.). Reakce systému na skokovou změnu žádaného proudu je znázorněna na obrázku 15, kde je vidět neschopnost regulátoru udržet proud na nulové hodnotě. Problém bude s největší pravděpodobností způsobem šumem.



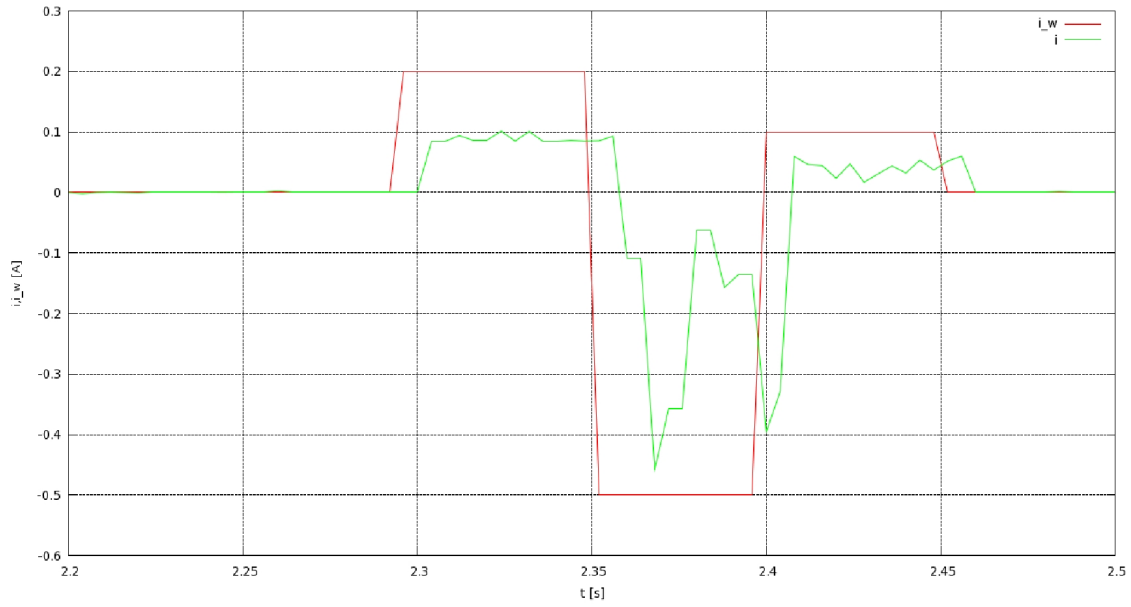
Obrázek 15: Reakce na skokovou změnu žádaného proudu rezistorem - PI regulátor

Další snaha byla vedena k návrhu P regulátoru, který zajistí nulový akční zásah při nulové regulační odchylce. Jako první byl tento regulátor testován na výkonovém rezistoru. Reakce na skokovou změnu proudu je znázorněna na obrázku 16, kde je vidět, že regulátor sice zajistí správnou regulaci na nulový proud, ale bohužel není možné dosáhnout požadovaného proudu při skokové změně. Větší zesílení vede pouze k rozkmitání systému.

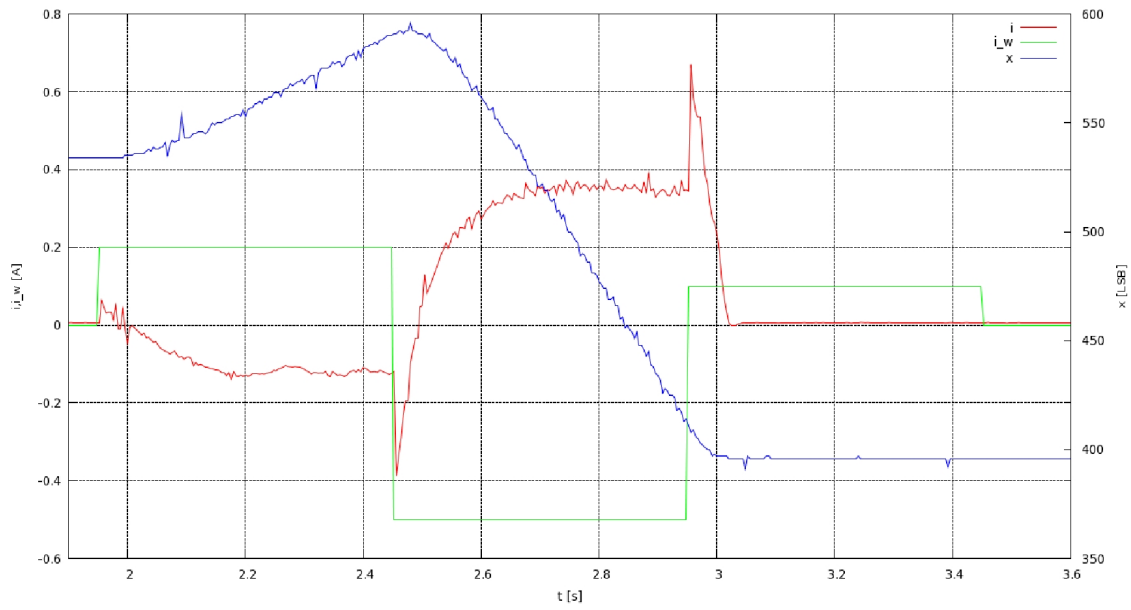
Tento regulátor byl otestován i se servomotorem s tím, že bylo zmenšeno zesílení (na rezistoru zesílení 2, zde 0.5). Zde však regulátor nebyl schopný reagovat správně na skokovou změnu požadovaného proudu, jak je znázorněno na obrázku 17.

5.2.3.1 Zhodnocení

S novou verzí výkonové a měřicí elektroniky se nepodařilo zcela dosáhnout vytyčeného cíle. První problém nastal již při ožívování elektroniky, jak je popsáno v 5.2.1.1, kde je popsána drobná úprava měřicí části elektroniky. Podle všech testů měřicí část elektroniky pracuje správně, ale regulátor proudu se nepodařilo správně „naladit“ a to jak P regulátor, tak i PI regulátor. Nepodařilo se mi zjistit podložený důvod mizerné regulace proudu. Bylo by vhodné doplnit testy o osciloskopická měření, protože proud je měřen A/D převodníkem s frekvencí 20kHz a dále ještě podvzorkován na 250Hz a tedy zde uváděné průběhy proudu neodhalí případné chvilkové jevy.



Obrázek 16: Reakce na skokovou změnu žádaného proudu rezistorem - P regulátor



Obrázek 17: Reakce na skokovou změnu žádaného proudu servomotorem - P regulátor

6 Závěr

6.1 Dosažené výsledky

- V rámci této práce byla otestována knihovna *lpcanvca.h*, která slouží jako programátorské rozhraní VCA pro mikro počítač. Pro tento účel byl implementován testovací software na testování ztráty zpráv při přenosu.
- Pro snadnější využití výše uvedené knihovny byl implementován ukázkový software pro mikro počítač, který ukazuje správné využití této knihovny.
- Software pro mikro počítač řídicí elektroniky servomotoru robotu Spejbla byl upraven s využitím knihovny *lpcanvca.h*. Komunikace přes průmyslovou sběrnici CAN je tedy obsluhována výhradně s využitím programátorského rozhraní VCA.
- Původní verze výkonové a měřicí elektroniky servomotoru byla otestována na přítomnost rušivých kmitů.
- Byla navržena, sestavena a otestována nová verze výkonové a měřicí elektroniky jako prototyp.
- Byl navržen nový software pro mikro počítač s ohledem na novou verzi výkonové a měřicí elektroniky, který vycházel z původního softwaru pro robota Spejbla.
- Nepodařilo se mi zcela doladit regulátor proudu.

6.2 Náměty na pokračování

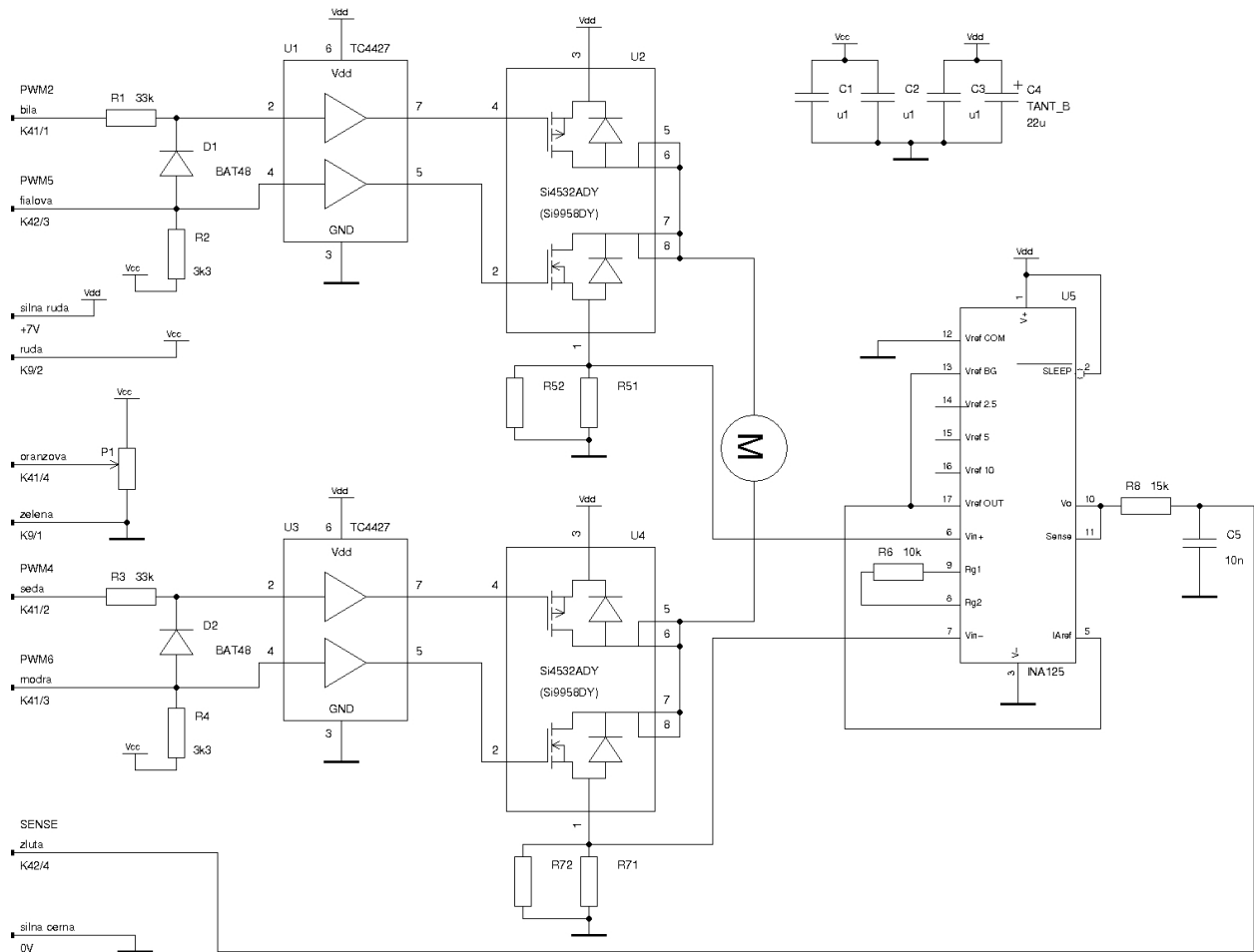
- Doplnit testy nové verze výkonové a měřicí elektroniky o osciloskopická měření a odhalit tak příčinu mizerné regulace proudu.
- Vyměnit přístrojový zesilovač v nové verzi výkonové a měřicí elektroniky za vhodnější typ.

Literatura

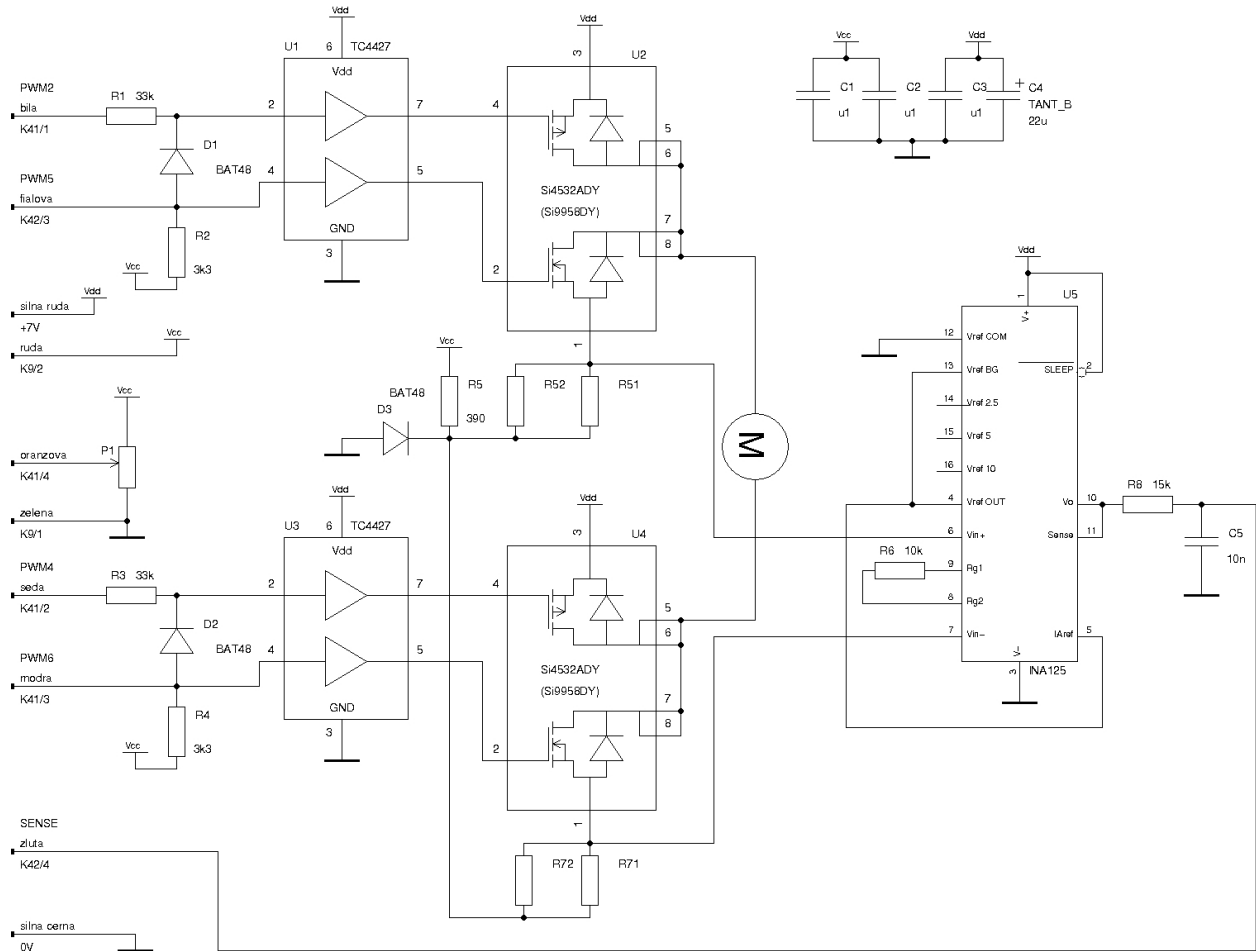
- [1] M. Peca. *Kráčející robot*. [s.l.], 2008. 66 s.
ČVUT - Fakulta elektrotechnická, katedra řídicí techniky.
http://rtime.felk.cvut.cz/~pecam1/spejbl_dp.pdf.
- [2] J. Krakora, P. Pisa, F. Vacek, Z. Sebek, P. Smolik, Z. Hanzalek,
Deliverable D7.4 Communication components V2, OCERA 2004,
<http://www.ocera.org/archive/Deliverables/wp7/D7.4.pdf>
- [3] V. Haasz, M. Sedláček. *Elektrická měření : Přístroje a metody*.
2. vyd. Praha : Vydavatelství ČVUT, 2003. 337 s.
- [4] M. Sojka, P. Piša, *OCERA Make System Manual*,
<http://rtime.felk.cvut.cz/omk/omk-manual.pdf>
- [5] *FT232R USB UART IC Datasheet Version 2.02*,
Future Technology Devices International Ltd.,
http://www.ftdichip.com/Documents/DataSheets/DS_FT232R_V202.pdf
- [6] *LPC21xx/22xx User Manual*, NXP Semiconductors,
<http://www.standardics.nxp.com/support/documents/microcontrollers/>
- [7] *CAN Specification 2.0B*, Robert Bosch GmbH, 1991
<http://www.semiconductors.bosch.de/pdf/can2spec.pdf>
- [8] *INA125 Instrumentation Amplifier Datasheet*, Texas Instruments
<http://www.ti.com/lit/gpn/ina125>

A Nová verze výkonové a měřicí elektroniky

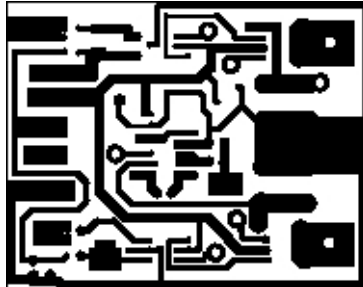
A.1 Schéma elektroniky



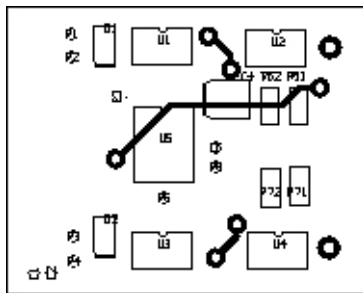
A.2 Schéma upravené elektroniky



A.3 Obrazec plošného spoje



A.4 Osazovací plán



B Obsah příloženého CD

- / `bakalarka.pdf` - elektronická verze této práce
- / `software_LPC` - adresář se softwarem pro mikropočítač
 - / `can>HelloWorld` - ukázkové využití knihovny `lpcanvca.h`
 - / `lpcanvcaTest` - software pro testování komunikace přes průmyslovou sběrnici CAN
 - / `spejbl_hodiny` - software pro řízení servomotoru robotu Spejbla
 - / `spejbl_motor` - software pro hodiny na sběrnici CAN
- / `software_PC` - adresář se softwarem pro počítač
 - / `cerveaux` - původní software pro řízení robotu Spejbla s úpravou pro použití s jedním servomotorem
 - / `el_test` - program pro testování nové verze výkonové a měřicí elektroniky (zasílá ovládací zprávy servomotoru přes sběrnici CAN)
 - / `lpcanvcatest` - program pro testování komunikace přes průmyslovou sběrnici CAN
 - / `servo_tests` - program pro testování původní (i nové) verze výkonové a měřicí elektroniky
 - / `zaznam` - program pro záznam naměřených a vypočtených hodnot přijatých přes průmyslovou sběrnici CAN od mikropočítače
- / `elektro` - obsahuje schémata a plošné spoje navržené v této práci
 - / `gschem-sym` - symboly pro program `gschem`
 - / `pcb-elements` - definice pouzder pro program `pcb`
 - / `sp_hmost1` - schéma a plošný spoj výkonové a měřicí elektroniky