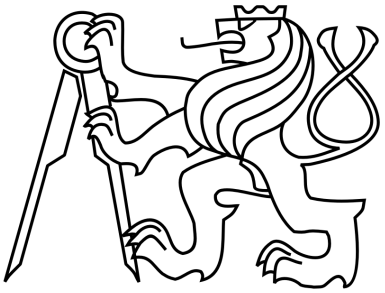
České vysoké učení technické v Praze
Fakulta elektrotechnická



Bakalářská práce

Řídící obvod pro stick-slip pohony

Martin Skorunka

Vedoucí práce: Ing. Martin Špiller

Studijní program: Elektrotechnika a informatika strukturovaný bakalářský

Obor: Kybernetika a měření

leden 2009

Poděkování

Předně bych rád poděkoval vedoucímu práce Ing. Martinovi Špillerovi za jeho pružnost, rady a nesmírnou trpělivost. Dále všem ostatním, kteří mi pomohli svým názorem, či radou při návrhu aplikace a uvést dílčí části výrobku do chodu. V neposlední řadě bych chtěl pak poděkovat své rodině, která mi byla oporou po celou dobu studia.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady(literaturu,projekty,SW atd.) uvedené v přiloženém seznamu.

V Praze dne 22. 1. 2009 .....

Abstract

This bachelor's thesis deals with design of controler for stick-slip linear motor in a way that it could be used for universal purposes and various types of aplications. As a model for this aplication there has been used multifunctional I/O board MF624 from company Humusoft. Main advantage of this aplication should be simple realization of time critical aplications.
In the beginning, this project considers a choise of used components. Then a choice for a proper hardware platform is being made, mentioning certain pros and cons. The last part is a study on the solutions of particular problems that emerged during the hardware and software implementation process.
The result of this project is a tool that supports solutions to projects where strict time criteria apears and needs to be met.

Abstrakt

Tato bakalářská práce se zabývá návrhem řídící jednotky pro pohon lineárního motoru typu stick-slip, aby byla použitelná univerzálně pro více aplikací. Jako vzor k této aplikaci posloužila multifunkční karta MF624 firmy Humusoft. Hlavní výhoda této aplikace by měla být snadná realizace časově kritických aplikací.
Obsahem práce je úvaha nad výběrem vhodných součástek , diskuze nad výběrem vhodné hardwarové platformy a popis řešení dílčích problémů spojených s implementací hardwarového a programového vybavení.
Tato práce vznikla jako pomůcka pro realizování aplikací, ve kterých se objevují problémy s přesnými časovými nároky.

Obsah

Seznam obrázků **xi**

Seznam tabulek **xi**

1 Úvod **1**
1.1 Motivace . . . 1
1.2 Nároky na řídící jednotku . . . 2
1.3 Princip Stick-slip motorů . . . 2
1.4 Použití stick-slip pohonů . . . 3

2 Testování karty MF624 **5**
2.1 Testování přesnosti . . . 5
2.1.1 Výsledky . . . 5
2.2 Testování rychlosti . . . 6
2.2.1 Výsledky . . . 6
2.3 Závěr . . . 7

3 Mikrokontrolér **9**
3.1 Vývojový prostředek . . . 10
3.2 Stručný popis ATMEGA16 . . . 10
3.3 Programování mikrokontrolérů AVR . . . 11
3.4 Paměťový prostor . . . 12
3.5 Popis portů . . . 13
3.6 Sériové rozhraní USART . . . 14
3.7 Sériové rozhraní SPI . . . 15

4 Ostatní obvody **19**
4.1 Sběrnicové obvody . . . 19
4.1.1 Popis standartu RS232 . . . 19
4.1.1.1 Stručný technický popis RS232 . . . 19
4.1.1.2 Asynchronní přenos dat . . . 19
4.1.2 Popis obvodu MAX232 . . . 20
4.1.2.1 Funkce obvodu . . . 20
4.1.2.2 Zapojení obvodu . . . 20
4.1.3 Popis obvodu FT232R . . . 22
4.2 Digitálně-analogový převodník . . . 22
4.2.1 Popis obvobu MCP4922 . . . 23
4.2.1.1 Popis sériového přenosu . . . 23
4.3 Operační zesilovač . . . 25
4.3.1 Obvod TL082 . . . 25
4.4 Napájecí obvody . . . 26
4.4.1 Popis obvodů 78LXX a 79LXX . . . 26
4.4.2 Popis obvodu LT1009 a LM336 . . . 27

5 Realizace řídící jednotky **29**
5.1 Vývojová deska . . . 29
5.2 Produkční deska . . . 29
5.3 Software . . . 30
5.4 Základní testy . . . 32

5.5 Testy na pohonu motoru . . . 33
5.6 Závěr . . . 35

6 Zhodnocení práce **37**
6.1 Hardware . . . 37
6.1.1 Plošný spoj . . . 37
6.1.2 Mikrokontrolér . . . 37
6.1.3 Napájecí obvody . . . 37
6.1.4 Podpůrné obvody . . . 37
6.1.5 DAC převodník . . . 37
6.1.6 Operační zesilovač . . . 38
6.2 Software výbava MCU . . . 38
6.3 Výsledky . . . 38

7 Literatura **40**

Seznam použitých zkratek **40**

A Manuál k desce **41**

B Komunikace s deskou **43**

C Schéma zapojení ostatních obvodů **44**

D Obsah přiloženého CD **45**

Seznam obrázků

1.1 Multifunkční IO deska MF624 . . . 1
1.2 Princip piezoelektrického motoru stick-slip . . . 2
1.3 Prototyp tříosého motoru . . . 3

2.1 Rušení na analogovém vstupu MF624 . . . 5
2.2 Nestabilita generovaných pulzů pomocí MF624 bez zátěže CPU . . . 6
2.3 Nestabilita generovaných pulzů pomocí MF624 se zátěží CPU . . . 6

3.1 Architektura procesoru ATMEGA16 . . . 11
3.2 Rozložení pinů ISP rozhraní . . . 12
3.3 Paměťový prostor procesoru ATMEGA16 . . . 12
3.4 Zjednodušený obvod USART procesoru ATMEGA16 . . . 15
3.5 Obvod SPI procesoru ATMEGA16 . . . 16
3.6 Propojení dvou SPI zařízení . . . 17
3.7 Komunikační slovo dvou SPI zařízení . . . 17

4.1 Princip funkce nábojové pumpy . . . 20
4.2 Sběrnicový obvod MAX232 . . . 21
4.3 Zapojení obvodu MAX232 . . . 21
4.4 Blokové shéma FTDI čipu FT232R . . . 22
4.5 Blokové shéma DAC převodníku MCP4922 . . . 24
4.6 Slovo přenosu dat po SPI . . . 25
4.7 Zapojení operačního zesilovače TL082 . . . 26
4.8 Zapojení napájecích obvodů řady 78L A 79L . . . 27
4.9 Zapojení reference LT1009 . . . 27

5.1 Testovací multifunkční deska . . . 29
5.2 Multifunkční deska . . . 30
5.3 Diagram přenosu dat a výběr jednotlivých funkcí . . . 31
5.4 Diagram podprogramu přenosu dat do DAC . . . 32
5.5 Rušení na analogovém výstupu . . . 33
5.6 Vyfiltrované rušení na analogovém výstupu . . . 33
5.7 Skok -10V až +10V bez filtrace . . . 34
5.8 Skok -10V až +10V s filtrací . . . 34
5.9 Maximální rychlost digitálního výstupu s použitím čítače . . . 34
5.10 Pilový průběh z pohonné jednotky a zkratovací pulz z desky . . . 35
5.11 Pilový průběh z pohonné jednotky a zkratovací pulz z desky . . . 35
5.12 Pilový průběh změřený na pohonné jednotce s piezomotorem . . . 35

Seznam tabulek

3.1 Popis alternativních funkcí portu B . . . 13
3.2 Popis alternativních funkcí portu D . . . 14

B.1 Popis komunikačních slov příkazů . . . 43

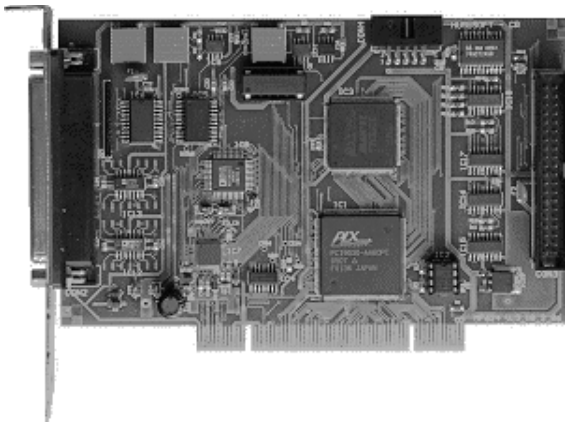
1 Úvod

Výběr řídící jednotky a návrh celého sytému je značně komplikovanou záležitostí. Od teoretických výpočtů následuje ještě složitější cesta k samotnému hardwarovému, praktickému řešení. Jedním z těchto zákulisí je volba správné řídící jednotky, která by byla schopná smysluplně řídit daný proces. Často se setkáváme s jednotkami, které nemají dostatek vstupně-výstupních portů, nebo nenabízejí funkci, která by se nám hodila, ale lepší jednotky umožňují si takové funkce doprogramovat. Velké možství dějů a aplikací nevyžaduje rychlou reakční dobu. U některých jednotek je dokonce čas jednoho cyklu stanoven na dobu jedné sekundy. Taková jednotka je použitelná například pro řízení vytápění domu, ale k řízení rychlých dějů, se kterými se často setkáváme, bude prakticky nepoužitelná.

Tato práce má za úkol nabídnout test možností multifunkční vstupně-výstupní karty MF624,na obrázku 1.1, od firmy Humusoft spol. s r.o., která má být použita k řízení obvodu pohonu stick-slip motoru. Jak čtenář pochopí později karta MF624 byla podle původních předpokladů k řízení takového pohonu příliš pomalá.

Celý zbytek následujícího textu se zabývá hardwarovým a softwarovým návrhem aplikace. Je v něm popsán a zdůvodněn výběr jednotlivých součástek. Po otestování funkcí desky jsou zohledněny její výhody a nevýhody. U jednotlivých nedostatků jsem také nabídl důvod a možnost jeho nápravy v náledujících verzích desky.

Další nemalou součástí práce je detailní popis zařízení, všech přepínačů, konektorů a komunikačního protokolu, kterým zařízení komunikuje s počítačem.



Obrázek 1.1: Multifunkční IO deska MF624

1.1 Motivace

Následuje několik důvodů, které mě přiměly zajímat se o toto téma.

Hlavním důvodem byla zejména vůle pomoci při realizace pohonu, který se jeví jako velice zajímavý pro mikropolohování a jistě bude prakticky využit.

Dalším důvodem byla touha po letech navrhnout skutečný hardware od počátečních úvah, až k finálnímu produktu, který by byl schopný správně řídit pohon motoru a sobecká touha ukojit svou kreativitu. Jelikož jsem neměl přílišné praktické zkušenosti s návrhem takového hardware, ani jsem nedisponoval znalostmi o konkrétním použitém hardware, bylo mou snahou předat čtenáři pokud možno co největší množství informací o problémech, které mě potkaly při návrhu, vývoji a testování výrobku.

1.2 Nároky na řídící jednotku

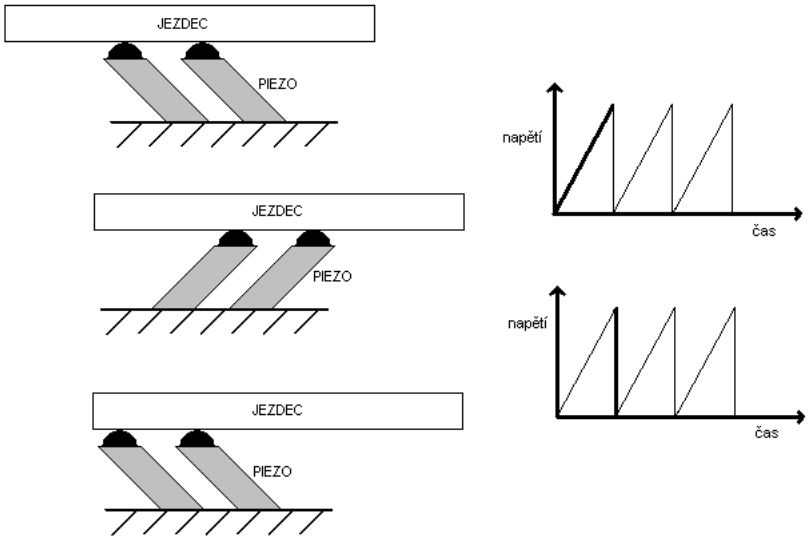
Protože řídící obvod je od počátku tvořen jako nástroj pro ovládání konkrétní aplikace, je třeba vymezit kritéria, která by měla řídící jednotka splňovat.

Řídící jednotka by měla nabídnout základní vstupně-výstupní funkce, jakými jsou analogový výstup s co nejmenší dynamickou chybou, digitální výstup pro každý kanál umožnující co nejplynuleji regulovat výstupní frekvenci až do 50kHz. Řídící jednotka by měla mít možnost snadného a přitom rychlého ovládání těchto výstupů s možností některé funkce upravovat případně zaměnit. Taková jednotka by měla být schopná později popsaný pohon řídit. Jeden z dalších požadavků bylo použití DAC převodníků se sériovým přenosem dat, aby bylo možné aplikaci později použít pro řízení vícekanálového pohonu. Během vývoje prostředku se zrodila myšlenka vytvořit pokud možno levný multifnkční prostředek, který by se dal použít pro velké množství aplikací.

1.3 Princip Stick-slip motorů

Fyzikální princip fungování stick-slip motoru je založen na objevu piezoelektrického jevu bratry Curierovými, kteří zjistili, že některé krystaly materiálů při mechanickém namáhání generují napětí. Později se zjistilo, že po přivedení elektrického napětí na tyto krystaly, dochází k jeho deformaci.(Tento jev nazýváme nepřesně elektrostricke,piezostrikce, nebo také nepřímý piezoelektrický jev).

Pro čtenáře musí být těžko představitelné, jak nepatrný pohyb, který čtenář zná spíše z piezoelektrických snímačů, mikrofonů a krystalů, dokáže pohybovat předmětem na mnohem delší vzdálenosti. Řešením by mohlo být zvětšení krystalů nebo rozložení pohybu do mnoha menších kroků. První řešení je utopické, protože krystaly by musely být extrémně velké. Tudíž byl hledán způsob, jak mechanicky vyřešit možnost rozfázování pohybu. Jedno z možných řešení je princip stick-slip motorů (přilnout-proklouznout), u kterých, jak už možná název napovídá, je využíváno fyzikálního principu tření. Nejlépe bude pochopitelný princip motoru na obrázku 1.2.



Obrázek 1.2: Princip piezoelektrického motoru stick-slip

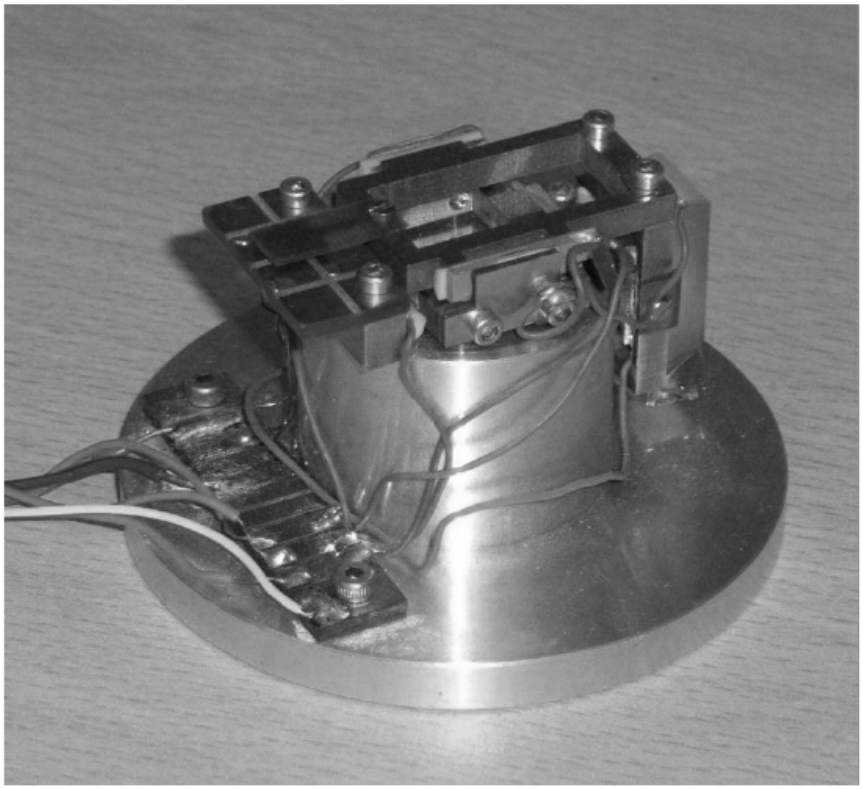
Při pomalém růstu napětí na krystalech se jezdec, který je volně položený na krystalech polarizovaných do střihu, pohybuje společně s krystaly. Při prudkém poklesu napětí se krystaly smykem vrátí do původní polohy, ale jezdec svou setrvačností zůstane stát.

Jak je zřejmé, lze tímto způsobem jezdcem pohybovat na libovolné vzdálenosti. Charakterem pohybu připomíná tento druh motoru motor krokový. Délka jednoho kompletního kroku bývá řádově 10 až 100 nm, lze však krok neprovádět celý, pak je přesnost pohybu určená pouze přesností snímače polohy.

Velikou pomocí mi při pochopení látky piezo-elementů byly přednášky a dokumenty od ing. *Jaroslava Honců* ,CSC [Hon02] a [Hon06] a *Avrida Bergandera* [Ber03], ze kterých jsem čerpal informace a později i obrázky pro tuto práci.

1.4 Použití stick-slip pohonů

V praxi tyto pohony nacházejí své uplatnění zejména v aplikacích kde je nutná velká přesnost pohonu. Stick-slip motory jsou používané při mikropolohování např. v mikroskopech, avšak použití těchto motorů je stále spíše ve fázi vývoje. Na obrázku 1.3 je vyobrazen prototyp tříosého stick-slip motoru.



Obrázek 1.3: Prototyp tříosého motoru

2 Testování karty MF624

V této kapitole se text zaobírá testováním multifunkční zásuvné karty MF624 pro účel řízení pohonné jednotky stick-slip motorů. Všechny zjištěné a změřené průběhy jsou pak v závěru kapitoly co nejobjektivněji zhodnoceny.

V první části se text věnuje přesnosti analogových výstupů karty, zejména v jejich statických parametrech, které jsou pro pohon velmi důležité, protože generátor pily vyžaduje stálou hladinu přiváděného napětí.

V druhé části se text zaobírá rychlostí karty MF624, tj. maximální rychlosti spínání a vypínání digitálních výstupů. Rychlost je v aplikaci ovládání pohonné jednotky velice důležitá, protože generátor pily vyžaduje po dosáhnutí maximální úrovně napětí piezokrystal zkratovat. Maximální frekvence, které je potřeba dosáhnout je 10kHz, při které musí být karta schopná plynule regulovat střídu zkratovacího pulzu.

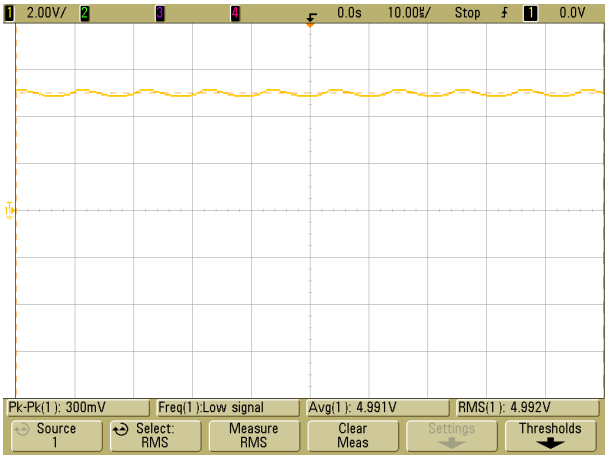
2.1 Testování přesnosti

Testování přesnosti probíhalo za použití přesného voltmetru a osciloskopu. Měřil jsem celou převodní charakteristiku desky (+-10V)za použití multimetru. Abych zjistil, zda není analogový výstup zarušený dynamicky, měřil průběhy napětí na několika úrovních osciloskopem.

Nastavování hodnot probíhalo z prostředí matlab v OS Windows XP, pomocí knihoven zhotovených firmou Humusoft.

2.1.1 Výsledky

Převodní charakteristika měřená voltmetrem by byla v grafu lineární, protože změřené hodnoty se nikdy nelišily o více jak 30mV. Z toho hlediska je karta velice přesná. Situace se proměnila v neprospěch karty, když jsem na výstup připojil osciloskop, protože výstup byl zarušený frekvencí 93kHz a peak-to-peak napětí tohoto rušení dosahovalo 300mV - detail na obrázku 2.1. Zdroj rušení se bohužel nedal přesně určit, ale domnívám se, že se jedná a rušení ze spínaného zdroje počítače.



Obrázek 2.1: Rušení na analogovém vstupu MF624

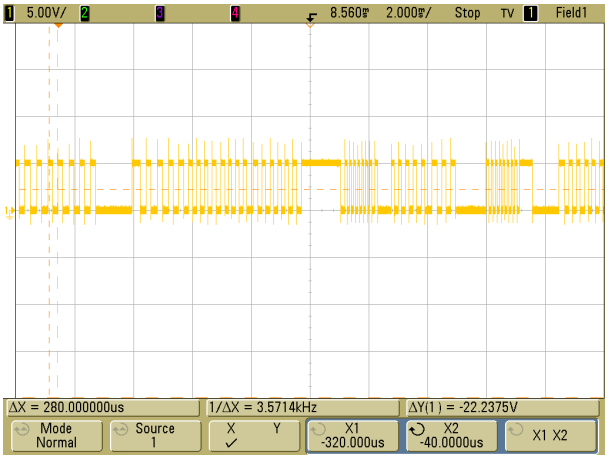
Pro potřeby řízení pohonné jednotky by se toto rušení dalo snadno vyfiltrovat dolnopropustním filtrem, protože pohon není náročný na dynamické vlastnosti výstupu karty.

2.2 Testování rychlosti

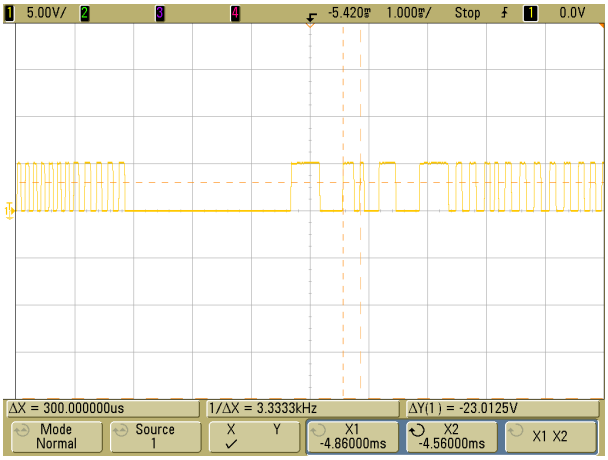
Testování rychlosti desky MF624 probíhalo měřením na digitálním výstupu za pomocí osciloskopu. Nastavování hodnot probíhalo z prostředí matlab v OS Windows XP pomocí knihoven zhotovených firmou Humusoft. Pro maximální rychlost střídání napěťových úrovní jsem naprogramoval jednoduchou smyčku, která opakovaně měnila výstup z HI do LO.

2.2.1 Výsledky

Jak je možné vidět na následujících obrázcích 2.2 a 2.3, karta není schopná generovat pravidelný signál, který by měl dostatečné rozlišení.



Obrázek 2.2: Nestabilita generovaných pulzů pomocí MF624 bez zátěže CPU



Obrázek 2.3: Nestabilita generovaných pulzů pomocí MF624 se zátěží CPU

Stabilita generování obdelníkového signálu by se zřejmě nechala vylepšit použitím RTOS, popř. použitím výkonnějšího počítače, aby nedocházelo ke zpožděním v softwarové vrstvě operačního systému Windows XP, ale rozlišení většího než řádově jednotky kHz by se stejně nepodařilo dosáhnout.

2.3 Závěr

Co se přesnosti týče karta Humusoft by byla pro pohonou jednotku dostatečná a dala by se po umístění filtru dolní propusti bez problémů použít. Bohužel interface, který deska používá, je silně závislý na použitém OS a pro danou aplikaci je velice pomalý, proto je tuto kartu nemožné použít pro ovládání pohonné jednotky.

3 Mikrokontrolér

V této kapitole se budu především zabývat úvahami o možných použitých mikrokontrolérech, konfrontacemi s datasheety a v neposlední řadě simulací složitějších obvodů, které by mohly splnit požadavky zadání.

Jelikož jednotka MF624 nebyla schopná vytvořit dostatečně rychlý signál, bylo nutné navrhnout zvláštní řídící jednotku. Jako nejjednodušší řešení se zdálo být použití jednoduchého mikrokontroléru, který by byl schopný generovat dostatečně rychlý binární výstup a k němu přiřadit přesný DAC, který by generoval budící signál pro zesilovač pohonné jednotky.

V první řadě jsem celkově zvážil nároky na takovou aplikaci a snažil se najít odpovídající součástky, které by dostatečně vyhověly těmto požadavkům. Později se však ukázalo, že prostředek, který vyvíjím, mohl mít mnohem širší uplatnění, protože kolegové z katedry začali vyvíjet další aplikace, ve kterých se deska MF624 hodí spíše pro testování, ale přísným požadavkům na rychlost již nevyhovuje.

Proto se zrodila myšlenka mnohem složitějšího prostředku, který by byl schopen v co nejvíce možných ohledech dostát možnostem MF624 při zachování své výhody rychlosti. To ovšem velice zkomplikovalo samotný hardwarový návrh, protože již nebylo možné zůstat u myšlenky jednoduchého mikrokontroléru s jednoduchým programem a triviálním rozhraním.

Jednotka MF624 obsahuje:

- Osm single-ended 14-bitových analogových vstupů
- Osm 14-bitových analogových výstupů
- 8 digitálních vstupů
- 8 digitálních výstupů
- 4 vstupy inkrementálních snímačů (diferenciální)
- 4 čítače

Jelikož jedním z nároků na aplikaci byla nízká pořizovací cena bylo zřejmé, že žádný levný mikrokontrolér nebude schopen dostát všem těmto nárokům. Do úvahy přišlo také řešení s více mikrokontroléry, které by spolu komunikovaly po sběrnici. Po počátečních úvahách jsem došel k závěru, že takové řešení by nebylo špatné, ale celou aplikaci by to enormně zkomplikovalo, tak bylo od tohoto řešení upuštěno.

Na trhu s elektronikou se dnes nabízí ohromné množství mikrokontrolérů. Při výběru bylo nutné zvážit dostupnost vývojového vybavení, jak hardwarového, tak softwarového. Při mém studiu jsem se setkal pouze s mikrokontroléry PIC, avšak ani jeden z nich nebyl až tak výhodný. Katedra navíc nevlastnila v době vývoje této práce žádný software ani hardware, se kterým by se nechal software simulovat a emulovat chod procesoru PIC.

Po dlouhém bádání, pročítání datasheetů a zvažování jsem došel k závěru, že bude výhodné použít jeden z osmibitových mikrokontrolérů řady ATMEGA od firmy ATMEL, které jsou známé svojí výbornou výbavou. Jako cenově a zároveň funkčně rozumné řešení se jevily mikrokontroléry ATMEGA32 popř. ATMEGA16, které jsou téměř identické, liší se pouze velikostí programové paměti a jiným rozložením adresního prostoru, které se však při využívání knihoven(názvů registrů) od firmy ATMEL neprojeví.

3.1 Vývojový prostředek

Na katedře byl však k dispozici vývojový prostředek firmy ATMEL STK500. Toto zařízení je již velice sofistikované.

Zařízení disponuje:

- hardwarové programování
- rozhraní JTEG
- rozhraní ISP
- řadu tlačítek
- řadu signalizačních diod
- RS232 rozhraní

Vývojový prostředek STK500 se připojuje k počítači pomocí rozhraní RS232, přes které je mikrokontrolér programován i je možné ho ladit. Rozhraní RS232 se však stalo pro mě velkým úskalím při používání tohoto vývojového nástroje. Žádný z mých počítačů, které jsem měl k dispozici doma a na koleji nebyl tímto rozhraním již vůbec vybaven. Na nějaký čas jsem zprovoznil staré PC, ale na tom byla práce doopravdy pomalá. Avšak práce s mikrokontroléry ATMEGA mě díky výborné dokumentaci, vývojovému software(AVR studio) tak zaujala, že jsem se rozhodl koupit si vlastní programátor. Jelikož produkty firmy Atmel jsou známé kvalitou svého zpracování, je i jejich cena značně vysoká. Na doporučení kolegy jsem se začal zajímat o programátor AVR Dragon. Jedná se o levnější produkt, který je určen spíše pro domácí použití a pro *entusiasty*.

AVR Dragon nabízí:

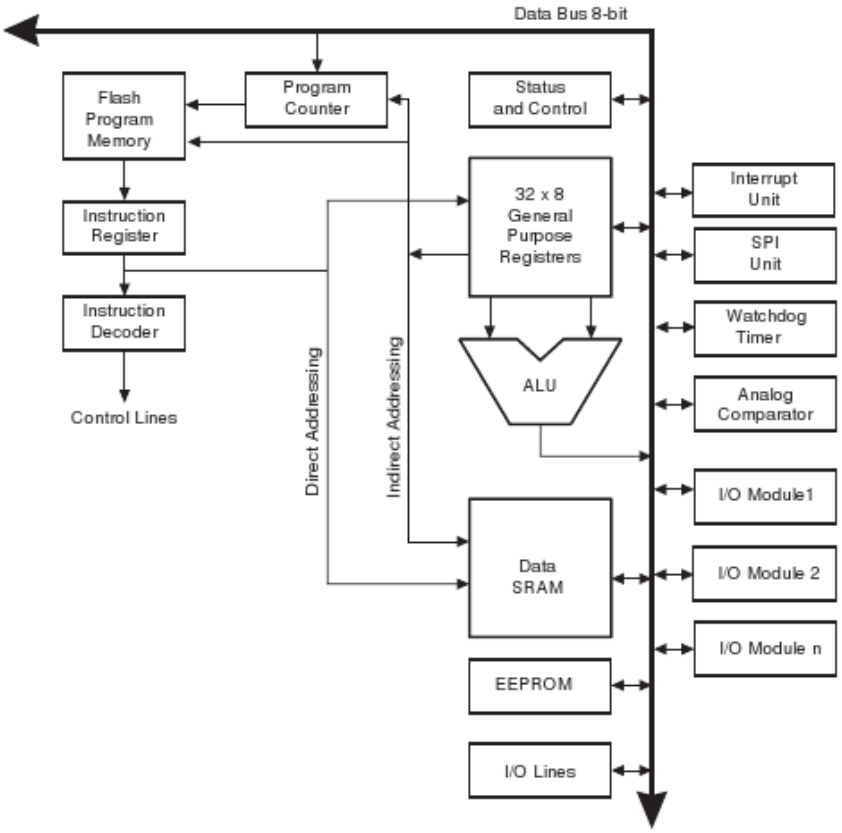
- rozhraní USB
- hardwarové programování
- rozhraní JTEG
- rozhraní ISP

Tento produkt, ale neobsahuje vůbec žádnou vývojovou část, lze si ji však snadno dodělat. Celkové náklady na výrobu kompletního vývojového nástroje ekvivalentního k STK500 jsou pak zhruba třetinové, proto jsem se rozhodl zakoupit tento prostředek. Za dobu, po kterou jsem tento vývojový prostředek používal, jsem zaregistroval pouze jediný problém, a to po updatování firmware samotného programátoru, který znemožnil jeho používání. Avšak problém byl opraven do několika hodin vrácením starého firmware a návratu ke starší verzi AVR Studia.

3.2 Stručný popis ATMEGA16

Mikrokontroléry AVR používají koncepci Harvardské architektury s jádrem RISC. Mikrokontrolér je vybaven 131 instrukcemi, ze kterých je většina jednocyklových. Jeho výkon je 16MIPS v případě taktování na maximálních 16MHz. Jádro AVR se skládá z 32 stejných 8 bitových univerzálních registrů, ve kterých mohou být obsazeny jak data, tak adresy. K přístupu

k těmto registrům stačí pouze jeden hodinový cyklus. Propojení univerzálních registrů s ALU (Aritmetic Logic Unit) zaručuje provedená právě jedné ALU operace během jednoho hodinového cyklu. Při této operaci jsou vstupní operandy uloženy v souboru registrů a výstup z této operace je následně uložen zpět do registru.
Posledních 6 registrů můžeme ve dvojicích použít jako ukazatele na adresy pro nepřímé adresování v paměťovém prostoru dat. Tyto 16 bitové registry označované písmeny X, Y a Z dovolují libovolné operace uložení (load/store). Programátor má tak na výběr, zda-li se ukazatel adresy bude po zpracování dané instrukce inkrementovat, nebo se před zpracováním dané instrukce bude dekrementovat. Pro adresování v 16 bitových registrech Y a Z, je velmi užitečné použít 6 bitového posunu ukazatele adresy.
ALU jednotka dále umožňuje aritmetické a logické operace mezi registry, nebo mezi registry a konstantou a dále také jednoregistrové operace. Ke standardním registrovým operacím se mohou přidat i operace využívající paměťové adresní módy na soubory registrů. To je umožněno tím, že soubor registrů zabírá dolních 32 adres datového prostoru 00 až 1F, a to dovoluje přístup k registrům jako jako k běžným paměťovým buňkám. Mikrokontrolér obsahuje 0.5kB paměti typu EEPROM a 1kB vnitřní SRAM paměti. Zjednodušená architektura mikroprocesoru je uvedena na obrázku 3.1.



Obrázek 3.1: Architektura procesoru ATMEGA16

3.3 Programování mikrokontrolérů AVR

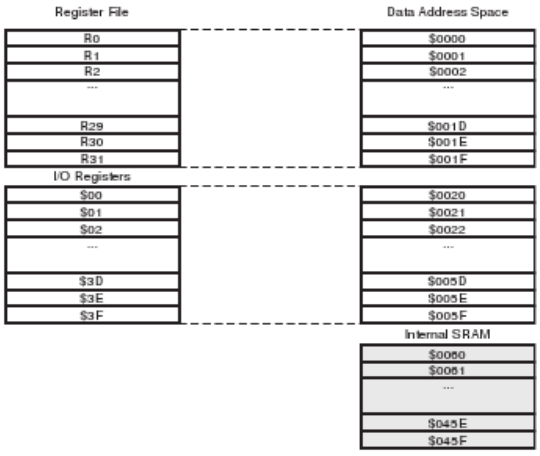
Mikrokontroléry AVR využívají koncepci Harvardské architektury (oddělený paměťový prostor programu a paměťový prostor dat). Program uložený v programové paměti je vyhodno-

cován jednoduchým překrýváním instrukcí (pipeline), tzn. zatímco jedna instrukce se provádí, následující instrukce se přesunuje z programové paměti.Programová paměť je "In System Programmable Flash Memory"s velikostí 32kByte. To znamená, že kromě klasického paralelního programování, které se používá u většiny jednočipových mikrokontrolérů, se dále využívá programování, u kterého dochází k přepnutí I/O portů z periferních obvodů k adresním a datovým vývodům vnitřní programové paměti. Přivedení log. 0 na pin reset dochází k přepnutí I/O portů na programovací. Po odebrání napětí z tohoto pinu, dojde k resetování celého programu a následné inicializaci programu. Druhou metodou je programování mikrokontroléru přes sériové rozhraní ISP. Mikrokontrolér zůstává v aplikaci a pomocí několika signálů (u většiny AVR MCU jsou to signály MOSI,MISO, SCK, RESET a GND) připojených k mikrokontroléru, se do mikrokontroléru nahraje nový program. Jelikož toto rozhraní jsem používal při programování mikrokontroléru a bude později použito i na vyvíjené jednotce, uvádím na obrázku 3.2 rozložení pinů ISP rozhraní.

Obrázek 3.2: Rozložení pinů ISP rozhraní

3.4 Paměťový prostor

Příklad paměťového prostoru ATMEGA16 je zobrazen na obrázku 3.3. Zde vidíme, že prvních 32 adres v datové paměti patří souboru registrů. Třebaže tyto registry nejsou implementovány jako součást SRAM, umožňuje tato organizace paměťového prostoru značnou flexibilitu v systému přístupu k jednotlivým registrům. Paměť programu typu FLASH má velikost 32kB a výrobce zaručuje, že paměť je možné 1000krát přeprogramovat. Při zápisu programu do paměti FLASH lze nastavením řídících bitů LB1 a LB2 zakázat čtení, a tím zabránit případnému zcizení programu.



Obrázek 3.3: Paměťový prostor procesoru ATMEGA16

3.5 Popis portů

Mimo pamětí je mikroprocesor vybaven i několika vstupně-výstupními sběrnicemi, které jsou v manuálu označované jako porty. Mikrokontrolér ATMEGA16 má čtyři konfigurovatelné vstupně-výstupní porty. Jsou označovány písmeny A až D.

Vstupně výstupní porty lze konfigurovat:

- PORTA[PA7..PA0]

  Port lze konfigurovat jako obousměrnou binární vstupně-výstupní sběrnici. Piny portu umožňují použití vnitřního pull-up rezistoru(Pro každý vstup zvlášť). Když se port chová jako vstup může se chovat jako zdroj i jako zem.

  Alternativně může port A sloužit jako 10bit 8-kanálový ADC převodník.

- PORTB[PB7..PB0]

  Port B má stejné vlastnosti jako port A.

  Alternativně jej lze však konfigurovat podle tabulky 3.1.

- PORTC[PC7..PC0]

  Port C má stejné vlastnosti jako port A.

  Alternativně jej lze však konfigurovat především jako rozhraní JTAG a two-wire, čímž se pro aplikaci alternativní funkce portu C stávají nepoužitelné.

- PORTD[PD7..PD0]

  Port D má stejné vlastnosti jako port A.

  Alternativně jej lze však konfigurovat podle tabulky 3.2.

PB7	SCK - hodinový signál SPI
PB6	MISO - master input,slave output
PB5	MOSI - master output,slave input
PB4	!SS - výběr slave zařízení
PB3	AIN1 (neg. vstup analogového komparátoru) OC0 (shoda porovnání časovače/čítače0)
PB2	AIN0 (pos. vstup analogového komparátoru) INT2(vstup pro externí přerušení)
PB1	T1 (externí zdroj pro časovač,čítač1)
PB0	T0 (externí zdroj pro časovač,čítač0) XCK (externí vstup pro zdroj hodin sběrnic USART)

Tabulka 3.1: Popis alternativních funkcí portu B

PD7	OC2 - výstup čítače,časovače 2 pro mód porovnávání
PD6	ICP1 - vstup čítače,časovače1
PD5	OC1A - výstup čítače,časovače 1A pro mód porovnávání
PD4	OC1B - výstup čítače,časovače 1B pro mód porovnávání
PD3	INT1 (externí vstup přerušení 1)
PD2	INT0 (externí vstup přerušení 0)
PD1	TXD (výstupní pin USART)
PD0	RXD (vstupní pin USART)

Tabulka 3.2: Popis alternativních funkcí portu D

3.6 Sériové rozhraní USART

Univerzální synchronní a asynchronní rozhraní je dnes již obvyklou výbavou lepších mikrokontrolérů, ale myslím si, že stojí za to rozebrat možnosti tohoto rozhraní, zvláště proto, že mám v úmyslu toto rozhraní použít pro komunikaci s PC pro zadávání příkazů.

Obvod USART mikroprocesoru umožňuje:

- Plně duplexní přenos
  Umožněn oddělením části přijímače a vysílače
- Synchronní i asynchronní přenos dat vnitřně konfigurovatelný registry obsluhujícími UART.
- Přesné a jemné generování přenosové rychlosti
- Podporuje datové rámce 5 až 9 datových bitů 1 až 2 stopbity
- Podporuje sudou i lichou paritu
- Detekce přetečení dat
- Detekce chyb rámcování
- Filtrování rušení - dolnopropustní filtr
- Tři oddělená přerušení pro dokončení vysílaní, dokončení přijímání a prázdný odesílací buffer.
- Mód multiprocesorové komunikace
- Dvojnásobná rychlost pro asynchronní mód.

Obvod USART se skládá ze tří částí:

- Obvod generování hodinového signálu

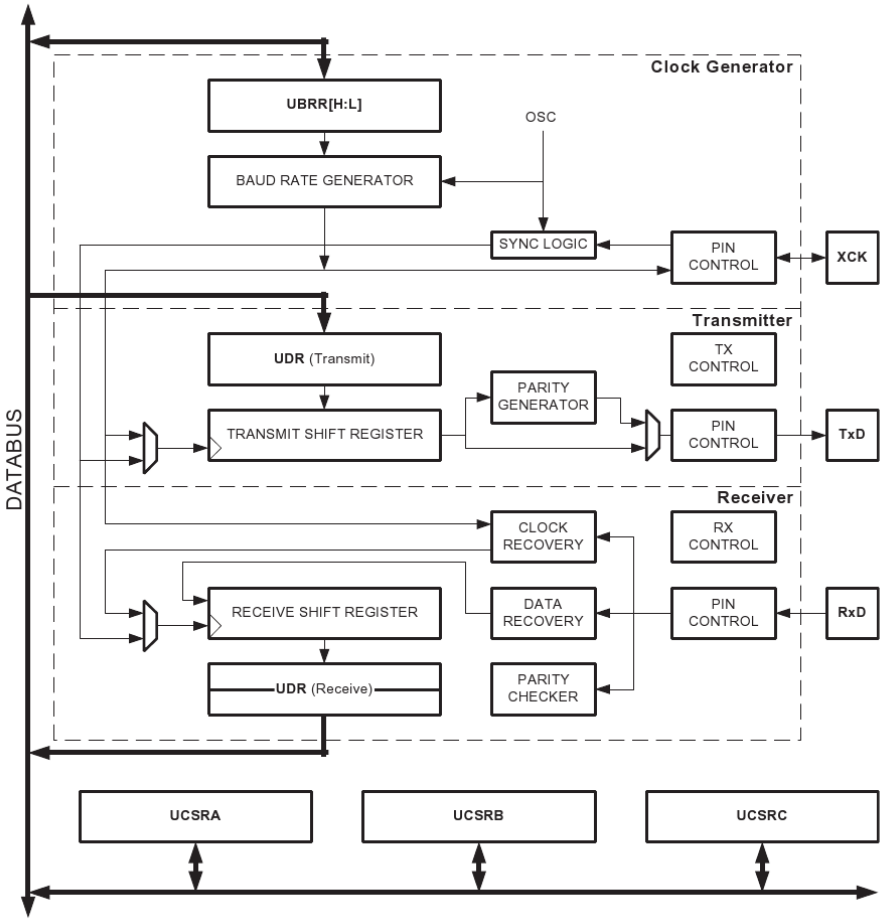
  V první řadě je nutné poznamenat, že při přenosu dat používám asynchronní přenos, u kterého je nutné, aby se hodinový signál u obou zařízení pokud možno rovnal. Dělící frekvence pro obvod USART je samozřejmě závislá na na frekvenci, na které pracuje mikroprocesor. Po detaily a tabulku s chybami převodů doporučuji zhlédnout datasheet od firmy*Atmel Corporation* [Cor08a].

- Obvod odeslání dat

  Samotný přenos dat probíhá zapsáním dat do registru UDR, je však důležité, aby programátor hlídal obsah příznaku UDRE, aby nedošlo k přepsání odesílaných dat. Já jsem však spíše příznivcem použití vektoru přerušení. Při použití přerušení lze snadno naprogramovat obsluhu přerušení tak, aby po odeslání znaku byl odeslán následující.

- Obvod příjmu dat

  U příjmu dat je také velice důležité nevyčítat data předčasně, ovšem s výhodou lze opět použít odděleného přerušení. Protože byl přidán druhý registr a registry jsou zapojené jako kruhová fronta, lze z registru vyčítat pouze jednou. Také je důležité před vyčtením dat z UDR registru zkontrolovat registry FE a DOR a vyčíst devátý bit přenosu, protože tato informace je po vyčtení UDR registru ztracena.



Obrázek 3.4: Zjednodušený obvod USART procesoru ATMEGA16

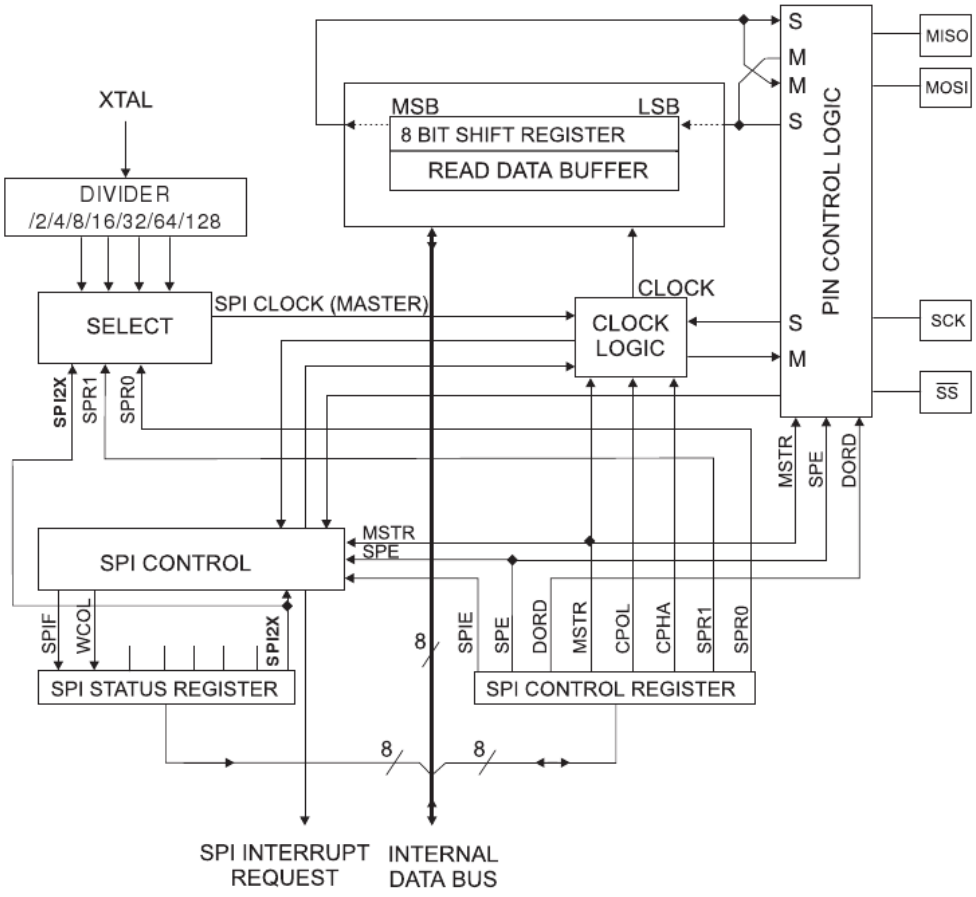
3.7 Sériové rozhraní SPI

Mikrokontrolér Atmega16 umožňuje využít hardwarového 8-bitového rozhraní pro přenos dat. Jeho veliká výhoda spočívá v tom, že se pouze do datového registru zapíší data a lze čekat na přerušení po dokončení přenosu dat.

Jedná se o třívodičové rozhraní, které umožňuje obousměrný přenos dat. Jelikož bylo v úmyslu použití sériových DAC převodníků, toto rozhraní se zdálo jako nejlépe použitelné díky své velké rychlosti.

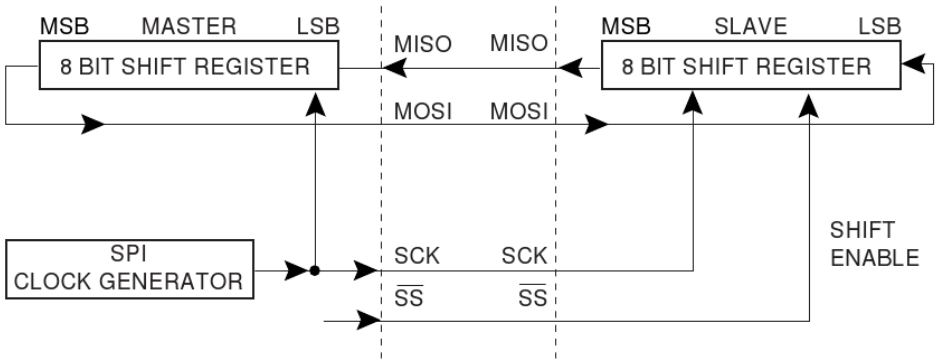
Jeho vnitřní struktura je zobrazena na obrázku 3.5, ze kterého je dobře pochopitelná funkčnost. Činnost celého obvodu je řízena dvěma kontrolními registry, které povolují přerušení(SPIE), volí master zařízení(MSTR), pro další funkce kontrolních registrů konzultujte [Cor08a].

Jako velice důležitou informaci považuji upozornění, že je nutné vyčíst buffer dat, před ukončením následujícího přenosu, protože pak jsou data přepsána. Pro nastavení rychlosti se používá dělič z hodinového signálu procesoru a lze ho nastavit od 2 až do 128 pro pomalá zařízení.

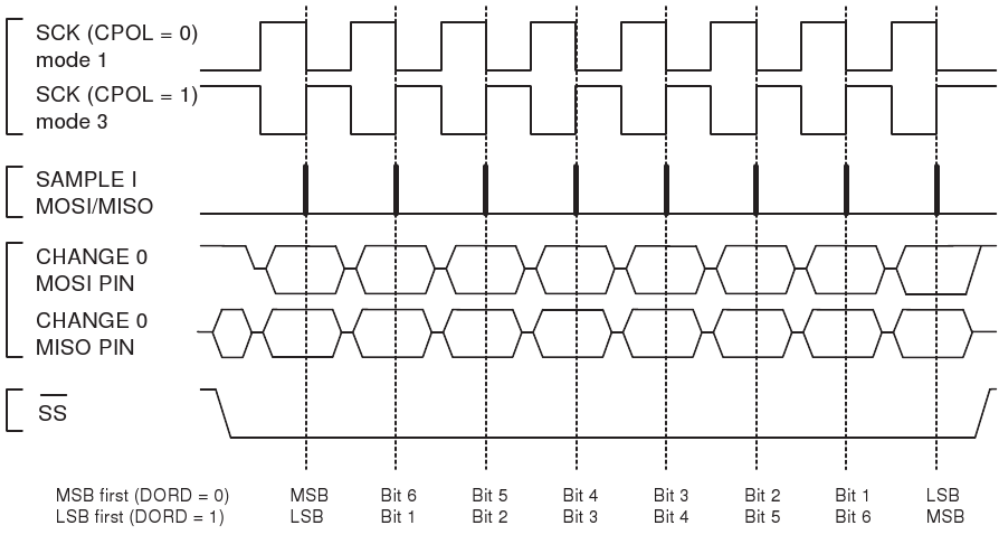


Obrázek 3.5: Obvod SPI procesoru ATMEGA16

Způsob jakým spolu dvě zařízení komunikují je velice zřejmý už z hardwarového pohledu na SPI hardware Atmega16, ale pro větší přehlednost uvádím ještě schéma propojení dvou SPI zařízení 3.5 a záznam ze sběrnice SPI, po které by už mělo být jasné, jak rozhraní funguje.



Obrázek 3.6: Propojení dvou SPI zařízení



Obrázek 3.7: Komunikační slovo dvou SPI zařízení

4 Ostatní obvody

V této kapitole se budu zabývat výběrem a popisem podpůrných obvodů, které jsou nutné pro správnou funkci celé aplikace.

Ačkoliv je mikrokontrolér ATMEGA16 velice samostatným a odolným obvodem, pro naší aplikaci se ukázalo, že jeho možnosti nebudou pro náš obvod dostačující, a tak musel být dovybaven řadou dalších velice důležitých obvodů.

ATMEGA je vybaven pouze TTL sběrnicemi, takže prvním z nutných obvodů bude budič sběrnice RS232.

Deska bude napájena symetrickým napětím +-15V, bude nutné použít řadu stabilizátorů, pro výrobu nižšího napětí.

4.1 Sběrnicové obvody

Jelikož je nutná vnější komunikace s mikrokontrolérem, bylo nutné zvážit jaký druh sběrnice bude použit. V úvahu přišla dvě možná řešení a to použití USB nebo UART rozhraní.

Procesor ATMEGA16 však nedisponuje hardware rozhraním USB, které by bylo velice náročné na implementaci, ale později jsem se dozvěděl o projektu AVR-USB, který umožňuje pod GNU/GPL licencí použít rozmanité druhy kódů jak pro samotný mikrokontrolér tak pro operační systém, ale v té době už byla aplikace v pokročilém stadiu vývoje.

Další možností bylo použití FTDI čipu, nebo nějakého jiného převodníku USB - UART. Avšak s těmito alternativními možnostmi jsem měl velice málo praktických zkušeností a programové úpravy by vydaly na samostatnou bakalářskou práci, tak po konzultaci s vedoucím bakalářské práce jsme se dohodli na ověřeném rozhraní RS232.

4.1.1 Popis standartu RS232

Používání této normy je v dnešní době na ústupu, avšak je stále velice populární u techniků, zejména díky své osvědčenosti v praxi, softwarové výbavě a v neposlední řadě velké odolnosti, které bylo dosaženo robustním hardwarovým řešením.

4.1.1.1 Stručný technický popis RS232

Tato norma popisuje jak hardware tak způsob přenosu jednotlivých dat. U RS232 se používá sériového přenosu dat, tj. data jsou vysílána bit po bitu, u RS232 probíhá komunikace od nejméně významného bitu(LSB) až po nejvýznamnější bit(MSB). Počet přenášených bitů je obvykle 8 (pro jeden byte dat), lze se ale setkat i s přenosem 5 až 9 bitů. Jednotlivé logické úrovně jsou definovány hladinou napětí, která se mohou lišit od +-5V až po +-15V, ale ve většině případů se setkáváme s -12V pro log. 1 a +12V pro log. 0.

4.1.1.2 Asynchronní přenos dat

Samotný přenos dat je pouze dvouvodičový, ale rozhraní definuje další signály pro řízení přenosu dat, avšak mikrokontrolér atmega16 je vybaven pouze datovými linkami, bylo akorát nutné dbát na správné zapojení konektorů.

Nevýhodami tohoto režimu je nutnost synchronizace vysílače a přijímače na počátku přenosu dat, velké přesnosti vnitřních hodin a dobrým vzorkovacím mechanismem propojených zařízení.

Jelikož tato sběrnice neobsahuje ochranu proti rušení, je dobré nepřipojovat zařízení na dlouhé vzdálenosti.

Ochranou, kterou RS232 používá, je synchronizační start bit, který již následují data LSB -> MSB. Slovo může být pro bezpečnější přenos doplněno paritním bitem, který doplní slovo

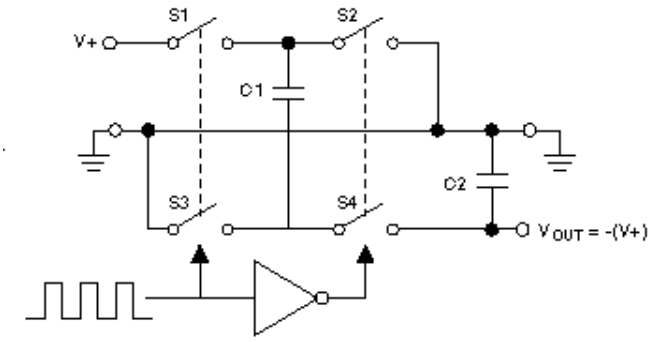
na sudou nebo lichou paritu, která slouží jako detekce chyby přenosu dat. Jako poslední článek slova jsou jeden nebo dva stopbity, které signalizují dokončení přenosu slova.

4.1.2 Popis obvodu MAX232

Jak už bylo zmíněno v předchozí sekci dokumentu RS232 nepracuje na TTL úrovni, proto bylo nutné netypický signál převést do TTL úrovně, protože procesor ATMEGA16 pracuje pouze s TTL úrovněmi. Dnes již existují jednoduché způsoby jak konvertovat úrovně RS232 na TTL a jednou z nich je monolitický obvod MAX232.

4.1.2.1 Funkce obvodu

Nejzajímavější vlastností obvodu je ta, že je napájen pouze pěti volty. Pro dosáhnutí hranice 12V tudíž musí použít obvodu, který by požadovaného napětí dosáhl. Obvody maxim používají dvojici obvodů nábojových pump, první se stará o zdvojnásobení napájecího napětí a druhá nábojová pumpa se stará o invertování zdvojeného napětí, která je znázorněna na obrázku 4.1.

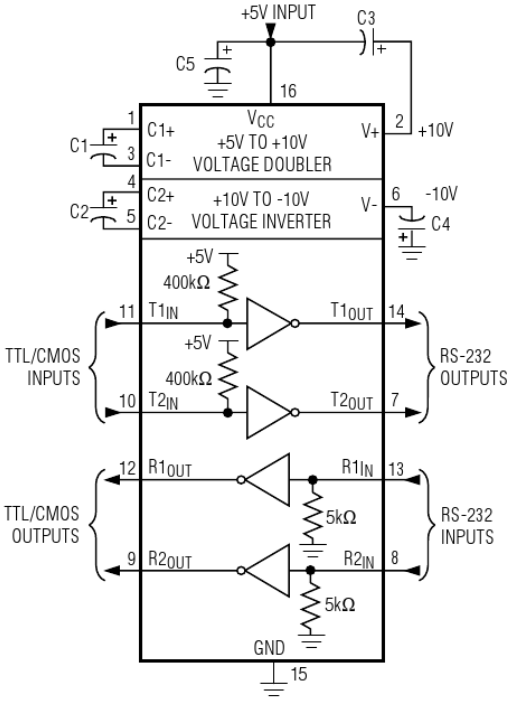


Obrázek 4.1: Princip funkce nábojové pumpy

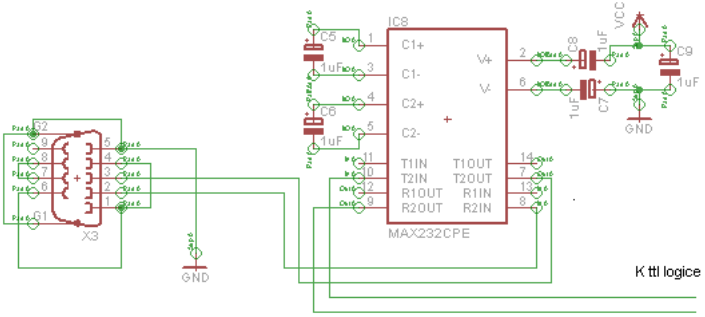
Obvod nefunguje jako jednoduchý budič sběrnice, ale i jako invertor, aby +12V dosáhlo 0V a -12V dosáhlo +5V. Nejnázorněji je struktura obvodu zobrazená na obrázku 4.2 z datasheetu firmy *MAXIM* [MIP08].

4.1.2.2 Zapojení obvodu

Jak už je zřejmé z obrázku 4.2 a z podstaty fungování obvodu, je potřeba k obvodu zapojit sérii kondenzátorů. Pro úplnost však přikládám i část schéma, ve které lze pozorovat kompletní propojení i s konektorem D-SUB. Schéma se nachází na obrázku 4.3.



Obrázek 4.2: Sběrnicový obvod MAX232



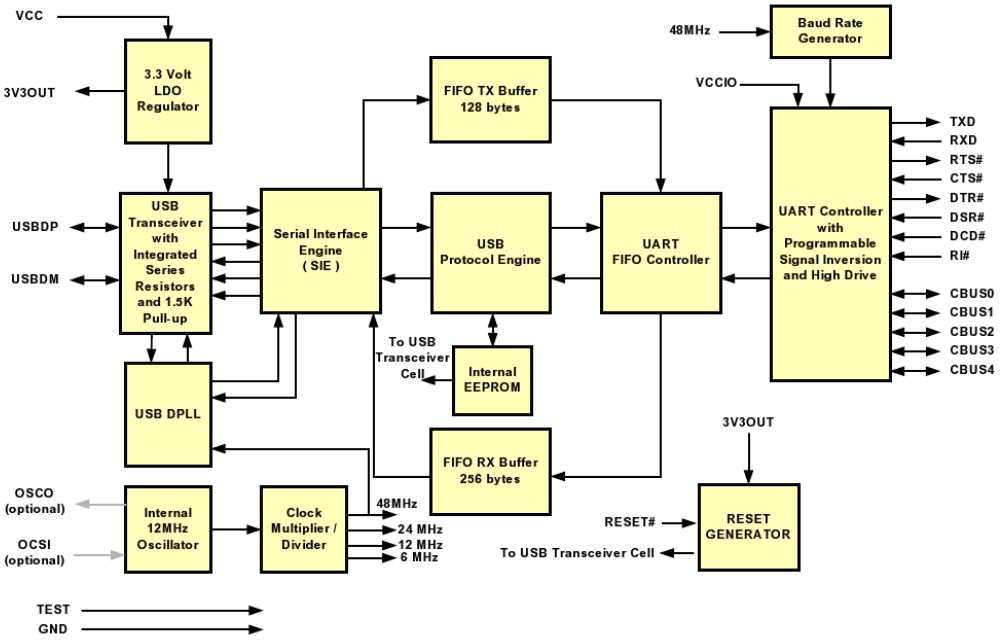
Obrázek 4.3: Zapojení obvodu MAX232

4.1.3 Popis obvodu FT232R

Jedná se o jednočipový převodník sběrnice USB na UART, často označovaný jako FTDI čip. Po připojení k poítači se chová v systému jako standartní COM port. Tento obvod je v dnešní době velice oblíbený díky své jednoduché implementaci, která umožňuje zachování programů a komunikačních protokolů ze starých RS232 rozhraní.

Tento čip oproti svým předchůdcům obsahuje vnitřní EEPROM paměť, která je předprogramovaná, tak ,aby každý z těchto čipů bylo možné jednoznačně odlišit, má zaintegrované odpory a dokonce i dolnopropustní filtr pro analogové napájení. Obvod je podle výrobce dobře odolný proti elektromagnetickému rušení. Obvod lze konfigurovat pro RS232 jako 7 a 8 bitový přenos s jedním nebo dvěma stopbity až do rychlosti jednoho Megabaudu.

USB rozhraní velké množství výhod, jako je možnost hotplug(připojení za běhu), velkou rychlost, spolehlivost a v neposlední řadě velký výskyt USB portu na nových počítačích, které již RS232 mnohdy nemají v základní výbavě, nebo v nedostatečném množství. Velkým a zásadním nedostatkem je nutnost složitého OS, který již podporuje USB zařízení, do kterých velké množství RTOS systémů zatím nepatří. Jeho vnitřní zjednodušená struktura je ke zhlédnutí na obrázku 4.4



Obrázek 4.4: Blokové shéma FTDI čipu FT232R

4.2 Digitálně-analogový převodník

Jelikož deska MF624 umožnuje 8 analogových výstupů a i aplikace stick-slip motoru potřebuje pro každý kanál jeden analogový výstup, bylo nutné výrobku umožnit generovat analogový výstup. První článek takového výstupu spočívá v převodu digitálního signálu na analogový.

V dnes je na trhu velké množství integrovaných obvodů, které tento převod umožňují.

Základní rozdělení DAC z hlediska přenosu dat:

- Paralelní
- Seriové

Zde jsem volil mezi sériovými převodníky, protože jsem měl v úmyslu vytvořit co nejvíce analogových výstupů s použitím co nejmenšího množství vývodů mikrokontroléru. Jelikož jsem chtěl dosáhnout co největšího rozlišení, musel jsem použít převodník s věším rozlišením než je 8 bitů, a tak bych při paralelním přenosu za použití demultiplexoru použil nejméně dva porty mikrokontroléru, což by zničilo záměr vytvoření universálního prostředku.

Dalším kriteriem pro výběr DAC byla cena. Jelikož jsem nikdy v předchozím studiu ani při svém hobby s DAC převodníky nepracoval, byl jsem velice nemile překvapen jejich cenou, která se mnohdy u přesných a rychlých typů blíží hranici tisícikoruny za jeden kus, jednokanálového DAC převodníku.

Zřejmě i díky své ceně jsou v pražských obchodech s elektronikou DAC převodníky špatně dostupné a mnohdy se musí objednávat a dlouho čekat na jejich dodání.

I přes všechny komplikace a konflikty zájmů se mi nakonec přes řadu kompromisů podařilo vybrat DAC převodník se seriovým přenosem, který by byl kdykoliv k mání za relativně nízkou cenu a stal se jím produkt firmy Microchip MCP4922.

4.2.1 Popis obvobu MCP4922

Obvod MCP4922 využívá architekturu odporových sítí, která má výhody v rychlém nastavní, nízké diferenciální nelinearitě(DNL) a malé teplotní závislosti. Převodník je 12bitový, ale má INL +-2LSB.

Obvod nabízí:

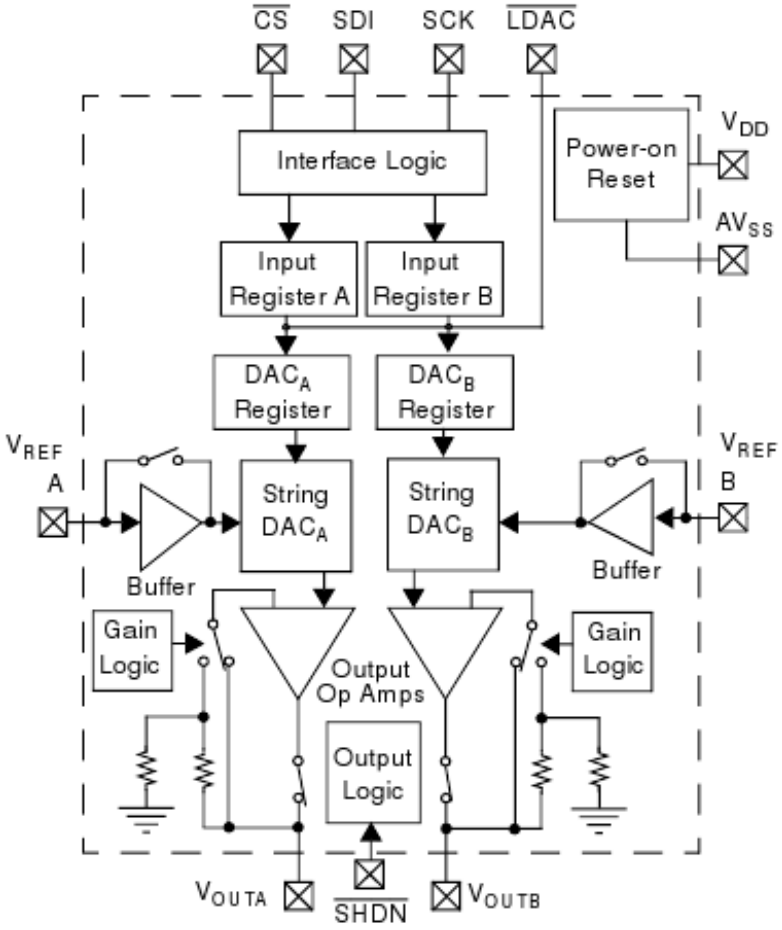
- 12-bitové rozlišení
- 2 nezávislé kanály
- Rail-to-rail výstupní napětí(0..5V)
- Volitelné zdvojené buffrování
- Synchronizované nastavení latch signálem
- SPI rozhraní až do 20MHz
- Doba ustálení 4.5us

Vnitřní struktura obvodu MCP4922 je ke zhlédnutí na obrázku 4.5, který ukazuje dva vstupní buffry s latch signálem. Jistě také stojí za zmínku vstupní SHDN signál připojený na výstup DAC, který přepíná výstup DAC do vysoké impedance.

Bohužel se jedná o jednosměrný převodník, který na každém svém vstupu snese napětí maximálně o 0.3V větší než je napájecí. Bylo tudíž zřejmé, že pro vygenerování bipolárního napětí je tento čip samostatně nepoužitelný, ale oproti jiným DAC snadno dostupným na našem trhu nabízí množství výhod, které mě přesvědčily k jeho použití, stojí za komplikaci v podobě přídavného analogového obvodu.

4.2.1.1 Popis sériového přenosu

Jelikož je převodník vybaven sériovým rozhraním pro přenos dat bylo nutné velice dobře prozkoumat slovo, které slouží pro nastavení převodníku. Původně jsem tento převodník chtěl použít kvůli rychlému SPI rozhraní, kterým je vybaven i mikrokontrolér ATMEGA16, bohužel jsem však udělal chybu a datasheet si nepřečetl dostatečně pozorně. Jelikož jsem neměl dostatečné zkušenosti se čtením datasheetů a byl jsem naivní v tom ohledu, že komerce neprostoupila



Obrázek 4.5: Blokové shéma DAC převodníku MCP4922

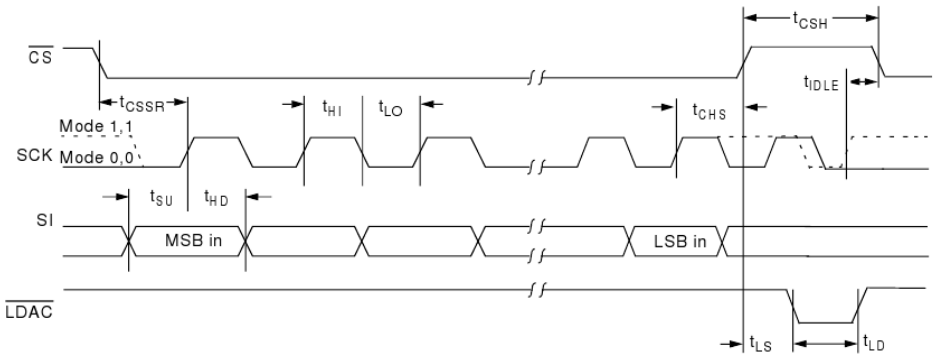
i technické manuály, a tak jsem zjistil až příliš pozdě, že nejdůležitější informace, které datasheet obvykle nabízí, jsou napsané tím nejmenším písmem v poznámkách pod textem.

Z této mé chyby vyplynula nemožnost použití hardwarové výbavy mikrokontroléru pro sériovou komunikaci s převodníkem, protože celá sekvence přenosu 16-ti bitů nesmí být po dobu komunikace přerušena signálem CS. Když totiž tato situace nastane vstupní buffer převodníku se kompletně zresetuje a čeká se na nový kompletní přenos dat. Mikrokontrolér však používá osmibitový přenos a k přerušení přenosu dojde po každém bytu přenesených dat. Bylo proto nutné sériový SPI přenos naprogramovat softwarově, takže jsem přenos byl nucen prozkoumat detailně.

Na obrázku 4.6 je zobrazeno jedno přenosové slovo. Jak je z datasheetu zřejmé nemusel jsem příliš řešit časování přenosu dat, protože softwarový přenos dat z mikrokontroléru bude mnohem pomalejší, protože přenos dat jsem musel realizovat ve smyčce, ve které se bude vždy první bit registru přenášet na výstupní port, s tím, že registr se bude muset posouvat doleva. Samotná instrukce pro nastavení výstupního portu trvá 2 cykly procesoru, přičemž musí procesor čekat na softwarový signál hodin (SCK), takže za předpokladu, že procesor poběží na maximální frekvenci v ideálním případě poběží přenos na 1MHz, ale díky podmíněným skokům a instrukci posuvu to bude zřejmě přenos ještě pomalejší. Další detaily jsou k nalezení v *datasheet* [Cor08b]

Popis bitů slova:

- 15 (MSB) - výběr kanálu A/B
- 14 - buffer zapnut když je true
- 13 - zesílení 1x když je true
- 12 - když je false tak se daný kanál přepne do vysoké impedance
- 11..0 - dvanáct datových bitů



Obrázek 4.6: Slovo přenosu dat po SPI

4.3 Operační zesilovač

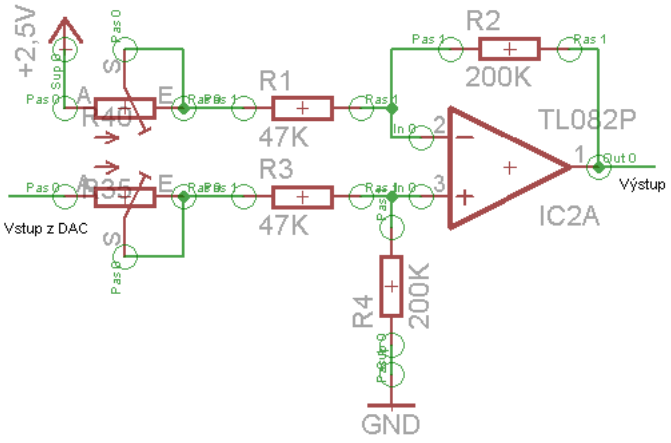
Jelikož jsem jako DAC převodník zvolil unipolární čip, bylo nutné převést unipolární napětí na rozsah +-10V. Podle manuálu k MCP4922 má převodník nejmenší DNL i INL při napájení reference 5V. Finální rozsah z převodníku bude tedy 0 až 5V. Nejjednodušším způsobem jak toto napětí převést na bipolární (+-10V), je použitím rozdílového zapojení operačního zesilovače.

4.3.1 Obvod TL082

Protože je DAC převodník pouze 12-ti bitový, nebylo nutné používat extrémně přesný operační zesilovač. Klasickým zástupcem operačních zesilovačů je obvod TL082. Jedná se o dvojitý JFET, vysokorychlostní, nízkonákladový operační zesilovač s velkou šířkou zesilovacího pásma. Další info v *datasheet* [STM08].

Pro mojí bakalářskou práci se tento zesilovač zdál jako ideální a po konzultaci s ing. Ondřejem Holubem se má domněnka potvrdila a mohl jsem přejít do fáze návrhu.

Jednoduchou úvahou dospějeme k závěru, že bude nutná 2.5V reference pro (-) vstup a zesílení rovno 4. V úvahu přišlo ještě použití přístrojového zapojení OZ, ale bylo odloženo do případné další verze, kdyby nebylo rozdílové zapojení OZ dostatečně přesné. Na obrázku 4.7 uvádím použité zapojení obvodu TL082. Vnitřní struktura OZ je víceméně klasická a odpovídá výukovým schématům z kurzu Elektronika na ČVUT, proto jí zde neuvádím. Trimry na vstupu v každé větvi slouží k jemnému doladění zesílení, které jsem do schématu umístil na doporučení vedoucího práce.



Obrázek 4.7: Zapojení operačního zesilovače TL082

4.4 Napájecí obvody

Celý obvod bude potřebovat několik napájecích úrovní a tak bylo nutné nastudovat a vybrat vhodné stabilizátory a reference pro napájení součástek. Musel jsem zde opět zohlednit snadnou dostupnost a cenu jednotlivých referencí, což mi nedalo příliš mnoho voleb, ale alespoň jsem jednotlivé součástky porovnal a pokusil se je otestovat, avšak jsem byl nucen dělat kompromisy, které přinesly další řadu omezení, které rozeberu u popisu jednotlivých součástek.

4.4.1 Popis obvodů 78LXX a 79LXX

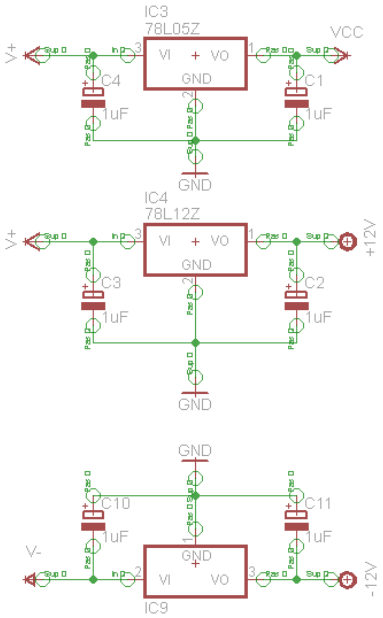
Obvody řady 78LXX i 79LXX ve své struktuře používají zpětnovazební smyčku, která přes obvod zesilování chyby, který zesiluje rozdíl napětí vnitřní reference a napětí výstupu (zpětná vazba), a přivádí jej zpět na vstup vstupního zesilovače. Obvody 78LXX A 79LXX jsou přesné a vyznačují se velkou stabilitou.

Vyrábí se v pouzdře TO92, protože nevyzařují za normálních provozních podmínek téměř žádné teplo. Aby bylo možné dosáhnout takových parametrů při zachování ceny mají obvody z této řady omezení na maximální výstupní proud 0.1A , ale oproti řadě 78XX A 79XX mají typickou chybu určenou výrobcem na 0.1 procenta oproti 1 procentu u řad 78XX.

V obvodu budou potřeba tyto úrovně napájení:

- 5V - obvod 78L05 Napájení TTL logiky, tj. MCU, DAC, MAX232.
- 12V - obvod 78L12 Napájení vyvedené přímo na výstup z desky, podle MF624
- -12V - obvod 79L12 Napájení vyvedené přímo na výstup z desky, podle MF624
- +-15V - přímo napájené ze zdroje Napájení symetrickým zdrojem pro obvody TL082

Zapojení všech obvodů jsem provedl přesně podle doporučení výrobce na obrázku 4.8, ale zřejmě by bylo vhodné zapojení doplnit blokovacími diodami a dolnopropustními filtry pro vyšší harmonické, protože by do obvodu mohlo pronikat rušení ze zdroje a při nešikovné manipulaci z deskou by mohlo dojít ke zničení napájecího obvodu. Detaily k nalezení v *datasheet* [Sem08].

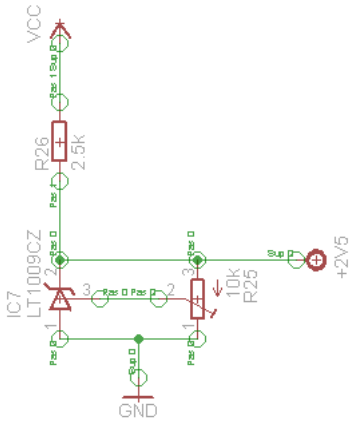


Obrázek 4.8: Zapojení napájecích obvodů řady 78L A 79L

4.4.2 Popis obvodu LT1009 a LM336

Jak jsem zmínil v kapitole o operačním zesilovači na vstup (-) OZ, bylo nutné přivést co nejpřesnější refereci 2.5V . Potom, co jsem přečetl manuály ke klasickým 2.5V referencím z řady LM, jsem usoudil, že bude vhodné navrhnout zapojení tak, aby bylo možné použít i přesnější referenci LT1009, protože jsem již nestihl tuto referenci otestovat.

Obvody mají tu vlastnost, že když je zapojen ve zpětném směru, chová se podobně jako zenerova dioda, tj jinak má velice strmou převodní charakteristiku kolem 2.5V. To znamená, že i když napájecí napětí kolísá, výstupní napětí je stále rovno 2.5V . Obvod LT1009 je oproti LM336 přesnější, ovšem jeho cena je téměř desetinásobná. Zapojení LT1009 i LM336 (4.9) je téměř identické, jen u LM336 je při použití součástky jako reference podle výrobce nevhodné používat ladící trimr.



Obrázek 4.9: Zapojení reference LT1009

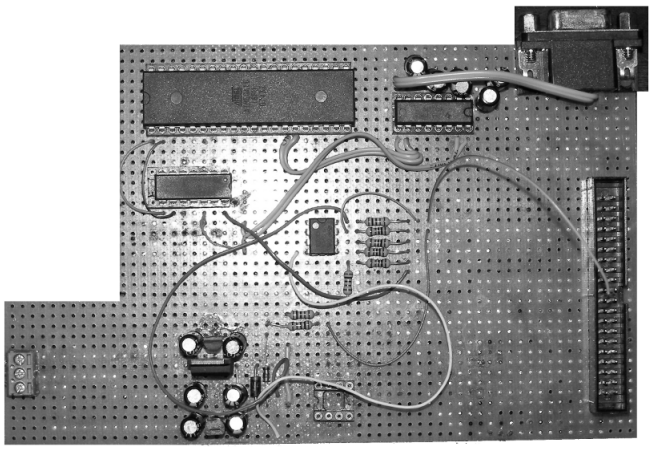
5 Realizace řídící jednotky

V této kapitole se budu zabývat popisem samotného vývoje aplikace. V první kapitole ve zkratce popíšu testovací obvod, na kterém jsem zkoušel první hardware i software výbavu, než jsem provedl samotnou realizaci. V dalších částech kapitoly se text zaobírá vývojem a testováním již konečného produktu. Velkou pomocí a zdrojem informací při návrhu mi byly články a fóra na *online stránkách* [ss09].

5.1 Vývojová deska

Pro ozkoušení základních zapojení součástek a software jsem vyvíjel testovací desku. Pro konstrukci jsem použil universální pájivé pole, které jsem volil kvůli větší spolehlivosti. Práce s ním je však zdlouhavá a neefektivní, proto bych nezkušeným vývojářům, kteří dělají více chyb, tuto variantu nedoporučoval.

Deska nakonec byla osazena v zjednodušeném zapojení finálního produktu, ale kromě 2.5V reference obsahovala všechny aktivní členy. Je jí možno zhlédnout na obrázku 5.1



Obrázek 5.1: Testovací multifunkční deska

Zrealizování tohoto testovacího obvodu hodnotím velice pozitivně, protože se podařilo identifikovat nevhodné součástky (zejména obvody 78XX) a nemožnost použití hardware rozhraní SPI, které je ve výbavě MCU. Dále jsem ozkoušel digitální vstupy, výstupy a analogový výstup z OZ. Bohužel jsem opomenul měřit analogový výstup osciloskopem a tak jsem neidentifikoval rušení, které se přenášelo do desky z počítače.

5.2 Produkční deska

Ve této kapitole popíšu již ostrou realizaci multifunkční desky, zmíním použitý software a v neposlední řadě popíšu chyby, závady a nedokonalosti v návrhu.

Schéma a výrobní podklady, které bylo nutné vytvořit pro výrobu desky, jsem vytvářel v softwaru *Eagle* [CC08] a mimo ostatní jsou poskytnuty na přiloženém CD. Hlavní výhodu v použití tohoto software jsem spatřoval v jeho jednoduchosti a předchozích zkušenostech. Pro-

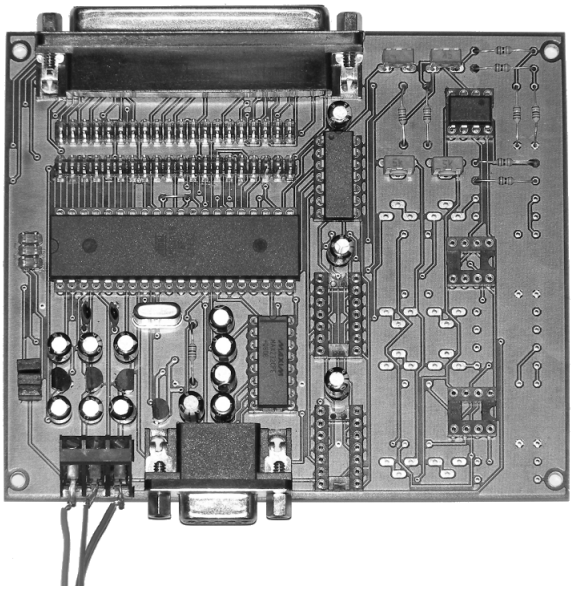
gram používal školou poskytnutou licenci, která měla ve verzi 4 ještě stále dosti omezení, zejména omezení velikosti desky. Jediný problém, který jsem při vývoji desky v tomto software zaznamenal, spočíval ve zpětné nekompatibilitě vytvořených dat. To znamenalo, že jsem byl nucen schéma překreslovat, protože jediná data, která jsem mohl použít byla mnou vytvořená knihovna obvodu MCP4922, proto bych všem, kteří vyvíjí pomocí tohoto software doporučil od začátku používat pouze jednu verzi programu.

Při vývoji desky jsem zohlednil identifikované problémy z vývojové desky, ale bohužel jsem se nevyhnul chybám, které plynuly z mé malé zkušenosti při vývoji plošných spojů.

Při návrhu desky jsem zohledňoval snadnou použitelnost v praxi. Deska je vybavena rozhraním pro programování procesoru, který umožňuje programovat MCU přímo na desce plošného spoje, bez odpojování napájení, nebo jakýchkoliv součástek. Pro konfigurovatelnost výstupních pinů D-SUB konektorů jsem na desku umístil dva 40-ti pinové konektory, které umožňují libovolné přepojování. V základní konfiguraci stačí přímo propojit tyto konektory a výstupní D-SUB konektor bude mít stejné rozložení jako deska MF624. Dále je na desce umístěn konektor, který umožňuje odpojit napájecí obvody VCC, 5V reference pro MCU a 2.5V reference pro OZ, a na místo nich připojit externí zdroj.

Jako nepříjemnost v mém návrhu považuji špatně vyvedený bod pro připojení země, který je vhodný požít při měření. Další závadu spatřuji v neoddělení digitální a analogové země, protože TLL obvody můžou ovlivňovat analogovou zem, přes kterou se do obvodu dostane vysokofrekvenční rušení. Tuto skutečnost jsem znal již při návrhu, ale protože DAC MCP4922 nemá oddělenou digitální a analogovou zem tuto skutečnost jsem zanedbal.

Na fotce 5.2 je zobrazena již téměř doosazená deska. Doosadit se mi jí bohužel nepodařilo kvůli nedostatku trimrů, které se mi nepodařilo včas sehnat.



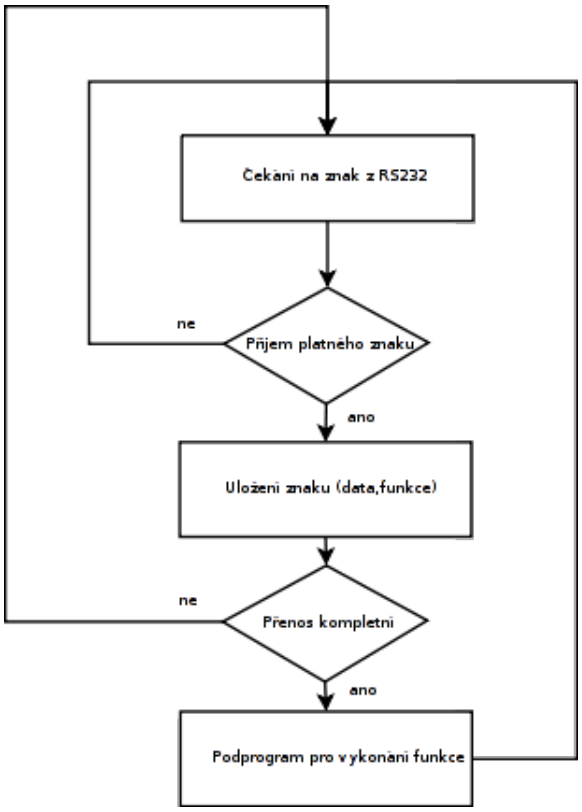
Obrázek 5.2: Multifunkční deska

5.3 Software

Samotný software, kterým je mikroprocesor vybaven zabral nemalou část času, který jsem do vývoje karty investoval, protože program byl vyvíjen v assembleru, který jsem do značné míry studoval, zejména pak instrukční sadu mikroprocesoru. Ačkoliv je kód, který jsem psal

assembleru pro jeho lepší optimalizaci, vydatně komentován a měl by být snadno pochopitelný v následujícím textu se jeho architekturu pokusím stručně popsat.

Pro komunikaci s počítačem deska využívá rozhraní RS232. Do TTL logiky je signál převáděn obvodem MAX232CPE. Mikroprocesor ATMEGA16 je vybaven hardwarovým rozhraním RS232, které se stará o přenos jednotlivých znaků. Jelikož obvod mikrokontroléru umožňuje generovat přerušení po dokončení přenosu, s výhodou ho používám, aby mikrokontrolér neplýtval výpočetní čas neustálým dotazováním na stav registrů. Pro většinu funkcí stačí přijmout čtyři znaky. Po přečtení všech potřebných znaků se daný příkaz vykoná. Univerzální diagram pro přenos všech znaků je na obrázku 5.3.



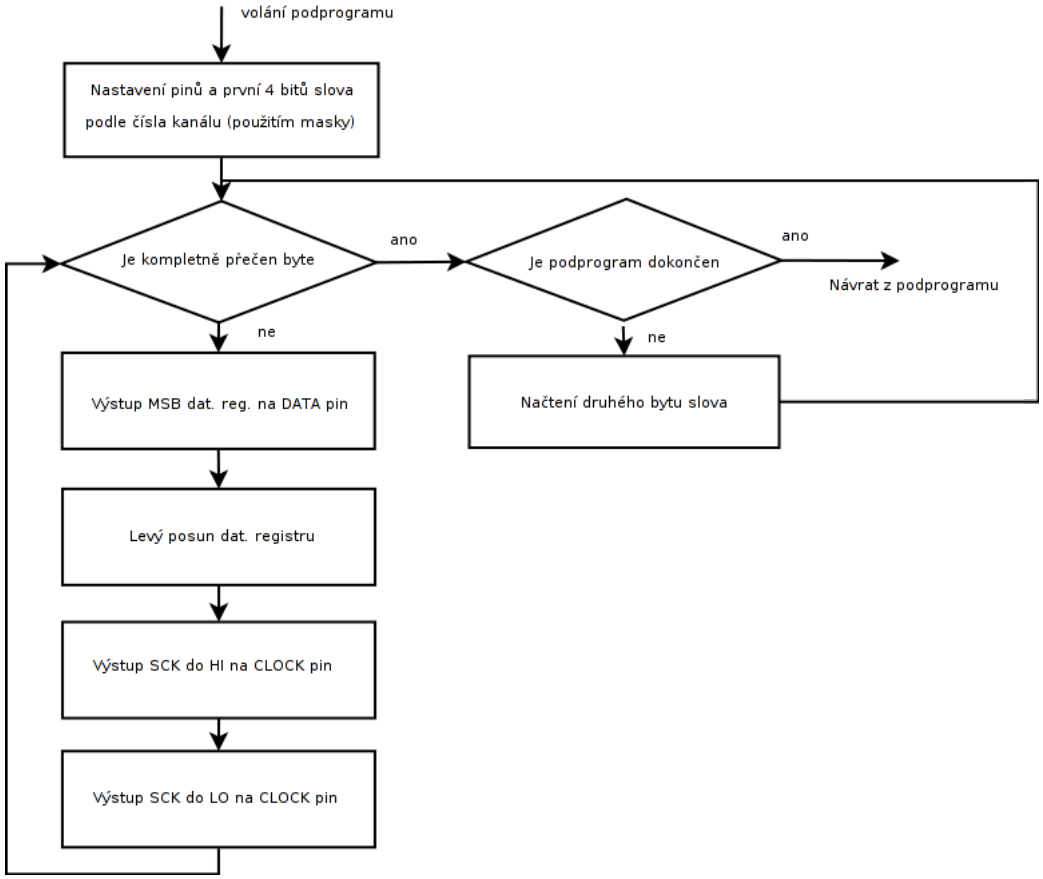
Obrázek 5.3: Diagram přenosu dat a výběr jednotlivých funkcí

Výhoda použití přerušení pro příjem znaků se v tomto kódu nijak neprojeví, protože mikroprocesor zatím kromě příjmu příkazů a jejich vykonávání nic nevykonává. Měl jsem v úmyslu umožnit nahrávání příkazů do SRAM, které by se vykonávaly ve smyčce, aby se odstranila nevýhoda pomalého RS232 rozhraní. Software jsem navrhoval tak, aby k tomuto účelu byl použitelný. Proto jsou i funkce mikroprocesoru kódovány jako podprogramy, aby je bylo možné volat z více míst. Ačkoliv jsem na kódu, který by možnost vykonávání příkazů ve smyčce, vydatně pracoval, nepodařilo se mi ho kvůli značné komplikovanosti včas odladit.

Funkce generování zkratovacího pulzu běží mimo tuto smyčku a je opět založená na mechanizmu přerušení, který generuje 16-ti bitový čítač, který se ještě v přerušení zresetuje. Prodloužení střídy signálu jsem dosáhl nastavitelným čekáním mezi přepnutím výstupního portu.

Další komplikovanou funkcí je nastavení výstupního napětí na DAC převodníků, respektive princip, jakým podprogram funguje, který je vyobrazen na diagramu 5.4

Pro funkci přenosu dat je potřeba přenést tři znaky. V prvním znaku je obsaženo číslo analogového výstupu. Toto číslo se přepočítá na nastavení prvního bitu slova DAC převodníku a na byte výstupních dat pro piny, které je potřeba aktivovat pro výběr správného DAC.



Obrázek 5.4: Diagram podprogramu přenosu dat do DAC

Druhým bytem dat je posledních osm datových bitů převodníku. Tato hodnota v programu není nijak kontrolována a v přenosu, nebo vypočtu by nemělo dojít k chybě.

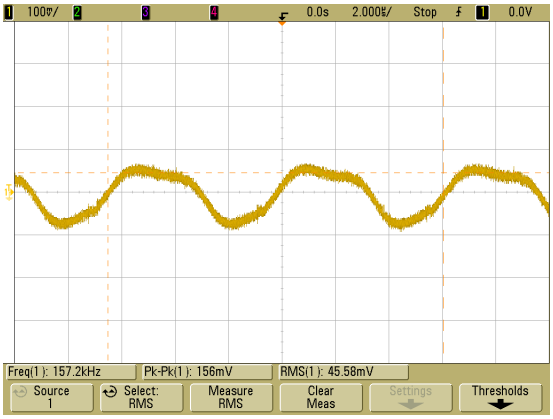
Třetím bytem přenesených dat jsou poslední čtyři znaky horního datového bytu. Pokud jsou některé bity horního slova nastavené do log.1 dojde k jejich vymaskování.

V prvním kroku podprogram vymaskuje horní 4 bity prvního bytu dat a sestaví kompletní první byte. Přenos obou bytů je prováděn identicky ve stejné smyčce. Na výstupní datový pin se vždy nastaví MSB registru. Pak se provede povinný obdélníkový signál na pin hodinového signálu. Poté dojde k levému posuvu dat v registru, když není přenos všech osmi bitů dokončen smyčka se opakuje po dokončení se do registru nahraje další byte dat a smyčka se opakuje. Po dokončení přenosu dolního bytu dat se vykonají potřebné signály pro projevení přenesených dat na výstupu DAC, kterými jsou signál LDAC, SHDN přepnutí signálů CS (chip-select) o původní hodnoty. Jako konečná instrukce se volá návrat z podprogramu, která pomocí dat uložených v zásobníku vrátí registr PC (program counter) na další instrukci která následuje volání podprogramu.

5.4 Základní testy

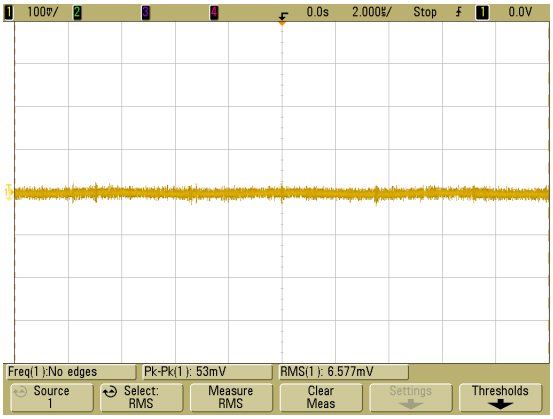
Jelikož jsem základní digitální funkce obvodu odzkoušel již na testovací desce zbývalo otestovat pouze přesnost a rychlost produkční desky.

Při prvních měřeních přesnosti jsem objevil, že do analogových výstupů se projevuje znatelné rušení, zejména když je na výstupu OZ nastaveno nulové napětí a výstup není zatížen, které dokumentuji obrázkem 5.5.



Obrázek 5.5: Rušení na analogovém výstupu

Zdroj rušení přichází do obvodu ze sběrnice RS232, které zmizí po odpojení této sběrnice, proto bych pro další verzi tohoto obvodu doporučil galvanické odpojení sběrnice RS232 od multifunkční desky. Pokusil jsem se toto rušení vyfiltrovat dolní propustí(5.6), ale bohužel došlo tím k degradaci dynamických vlastností výstupu(5.8 a 5.7). Dalším testem, kterým jsem desku



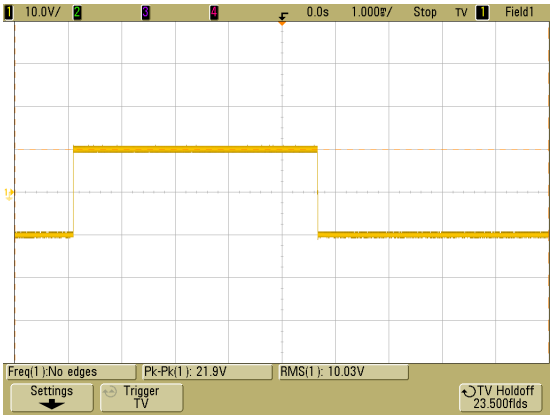
Obrázek 5.6: Vyfiltrované rušení na analogovém výstupu

podrobil byla rychlost digitálního výstupu. Výsledek si lze prohlédnout na obrázku 5.9. Maximální rychlost, který zachová požadované rozlišení frekvence, dosáhla deska 1MHz za použití 16-ti bitového čítače.

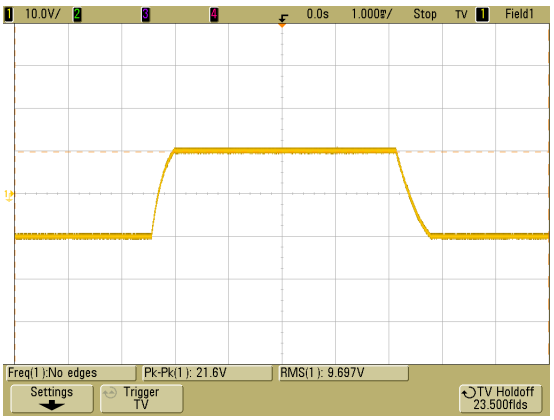
Po virtuálním motoru jsem otestoval pohon na opravdovém piezomotoru. Oproti kondenzátoru je piezomotor mnohem složitější systém, tak bylo nutné doprogramovat možnost prodloužení zkratovacího pulzu, aby se odstranily zákmity piezomotoru. Výsledek dopadl velice dobře a je možné si ho prohlédnout na obrázku 5.12.

5.5 Testy na pohonu motoru

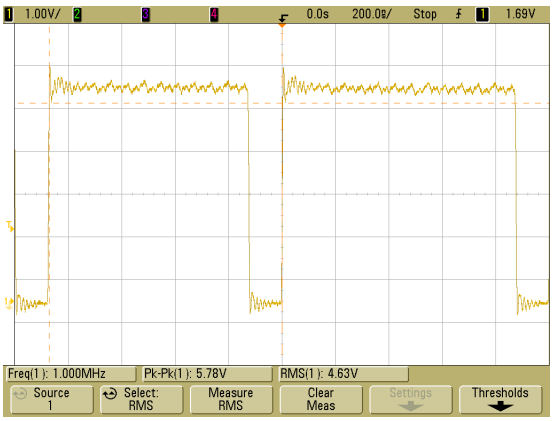
Po prvních testech, jsem desku použil jako generátor vstupních signálů pro pohon stick-slip motoru. První test na pohonu probíhal na virtuálním motoru (kondenzátor) a výsledek byl oproti desce MF624 výborný. Výsledek je ke zhlédnutí 5.10 a detail 5.11.



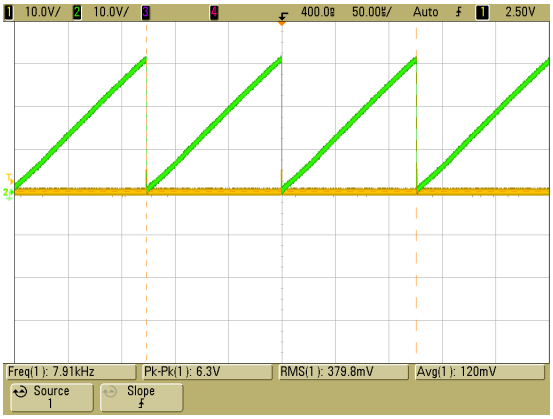
Obrázek 5.7: Skok -10V až +10V bez filtrace



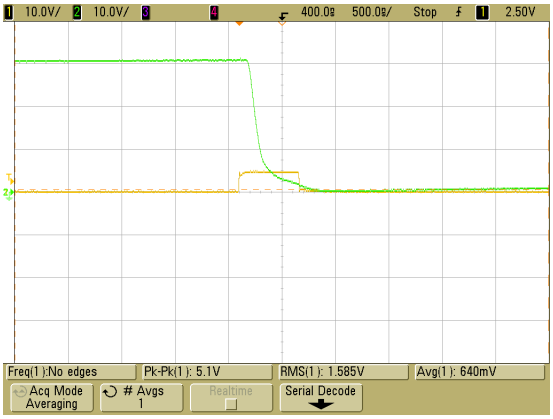
Obrázek 5.8: Skok -10V až +10V s filtrací



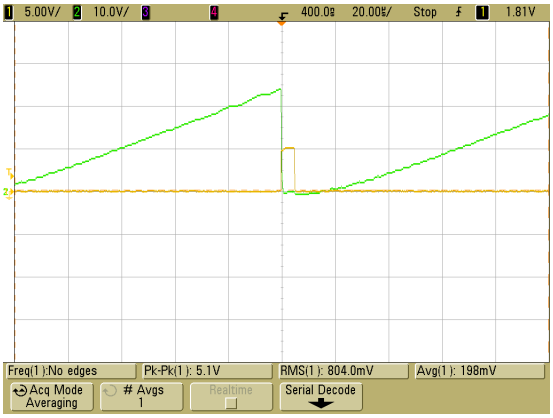
Obrázek 5.9: Maximální rychlost digitálního výstupu s použitím čítače



Obrázek 5.10: Pilový průběh z pohonné jednotky a zkratovací pulz z desky



Obrázek 5.11: Pilový průběh z pohonné jednotky a zkratovací pulz z desky



Obrázek 5.12: Pilový průběh změřený na pohonné jednotce s piezomotorem

5.6 Závěr

Ukázalo se, že pro požadavky generování řídících signálů pro pohonou jednotku stick-slip motoru je řídící jednotka vhodná a ve všech parametrech splňuje požadavky i pro maximální rychlost motoru.

6 Zhodnocení práce

V této kapitole text pojednává zejména o výsledcích celé práce. Celou kapitolu jsem rozdělil do dvou hlavních částí. V první části se pokusím probrat detaily a postřehy, které jsem nasbíral o použitých součástkách. V druhé části se pokusím detailněji probrat software, který jsem pro aplikaci vyvinul.

6.1 Hardware

6.1.1 Plošný spoj

Domnívám se, že provedení plošného spoje bylo úspěšné a žádné závažné chyby v návrhu jsem neobjevil. Výborným vylepšením by však bylo optimalizovat propojovací cesty ručně, aby se dosáhlo menšího množství prokovů. Dalším zlepšením by bylo rozdělením digitální a analogové země. Tuto změnu bych zkombinoval s výměnou DAC převodníků za typ který má rozdělené napájení digitálních a analogových obvodů.

6.1.2 Mikrokontrolér

Mikrokontrolér ATMEGA16 při testování svých parametrů úspěšně splnil veškeré požadavky, které by měly pokrýt větší množství aplikací, než deska MF624. Jelikož se jedná a nízkonákladový mikrokontrolér, lze o něm prohlásit, že je jedním z nejlepších v dané kategorii. Kdybych přemýšlel o změně procesoru musel bych již vkročit do kategorie signálových procesorů s jádrem ARM, které by byly samozřejmě ve všech parametrech lepší, ale jejich pořizovací cena je mnohonásobně vyšší.

6.1.3 Napájecí obvody

Po počátečních testech na vývojové desce jsem pochopil, že pro napájení OZ budu muset použít externí laboratorní zdroj a pro další napájecí obvody použít stabilizátory 78LXX a 79LXX, protože jsou mnohem lepší ve statických i dynamických parametrech. Dokonce je i reference LM336 dostatečně přesná a nebylo jí nutné měnit, takže obvody napájení kompletně splnily požadavky.

6.1.4 Podpůrné obvody

Jediným podpůrným obvodem byl budič sběrnice RS232. Obvod MAX232CPE se stal nakonec velkým zklamáním, protože přes něj do celého obvodu pronikalo rušení o maximální frekvenci 100kHz a amplitudy 100mV. Jeho zdrojem je pravděpodobně zdroj počítače a pro další verze desky bych rozhodně doporučil tento obvod vyměnit, nebo celou sběrnici galvanicky oddělit optočleny.

6.1.5 DAC převodník

Během testování jsem nenarazil na důvod proč by bylo vhodné DAC MCP4922 měnit, ale kdyby v budoucích aplikacích jeho přesnost nebo rychlost nevyhovovala, byl by zřejmě předělat celou architekturu prostředku, protože samotný MCU ATMEGA16 by byl při 16MHz příliš pomalý. Na výstup se dokonce ani výrazně neprojevilo, že má napájenou digitální i analogovou část ze stejné větve(společná digitální a analogová zem)

6.1.6 Operační zesilovač

S obvodem TL082 se také nevyskytl žádný velký problém ani komplikace, která by stála za zmínku. Je však velice důležité obvod napájet přesným napětím a zkontrolovat linearitu zesílení, popřípadě jí za použití trimrů korigovat.

6.2 Software výbava MCU

Celý program pro MCU ATMEGA16 byl vyvíjen v assembleru, aby jsem byl schopný optimalizovat program na co nejnižší úrovni. Jelikož MCU neprovádí téměř žádné výpočty bylo by neefektivní použít programovací jazyk C. Mimo jiné mám i dojem, že assembler nutí programátora pochopit samotný hardware MCU, což je při práci na tak nízké úrovni jako je programování digitálních přenosů velice důležité.

Bohužel se mi nepodařilo včas naprogramovat všechny funkce, které jsem chtěl. Obvod nyní umí nastavit všech šest analogových výstupů, nastavit osm digitálních výstupů(jednotlivě i najednou) a přečíst všechny čtyři digitální vstupy.

Program, který by uměl číst analogové vstupy nahrávat jednotlivé příkazy do paměti a vykonávat je ve smyčce, byl však značně komplikovaný a nepodařilo se mi jej včas odladit, proto tyto funkce MCU zatím neumí a pro jejich využití je bude nutné doprogramovat.

6.3 Výsledky

Domnívám se, že vzhledem k tomu že se jedná o první verzi multifunkční desky tak její schopnosti jsou výborné. Menší nedostatky, které se v aplikaci vyskytly jsou v dalších verzích snadno odstranitelné.

Funkčnost, kterou jsem nestihl doprogramovat, bude pro člověka zběhlého v programování assembleru otázkou několika dní a jednotka bude vybavená pro mnohem náročnější aplikace než je řízení stick-slip motoru.

Hlavní výhodou této práce je právě její modularita a možnost program MCU přizpůsobit přesně dané aplikaci. Avšak v dalších verzích by bylo dobré dokončit obvod analogových vstupů tak aby byl bipolární +-10V, protože tímto deska ztrácí svojí výhodu univerzality.

Jako shrnutí práce uvádím seznam věcí, které se podařilo úspěšně zprovoznit.

Hardware:

- 8 digitálních TTL výstupů
- 4 digitální TTL vstupy
- 6 analogových výstupů(+-10V)
- 8 analogových vstupů (0..5V)

Software:

- 16 digitálních TTL výstupů
- 4 digitální TTL vstupy
- 6 analogových výstupů(+-10V)
- rychlý generátor pulzů založený na 16-ti bitovém čítači

7 Literatura

[Ber03] Arvid Bergander. Control, wear testing integration of stick-slip micropositioning., 2003.

[CC08] inc. CasSoft Computer. Eagle v4 a v5 manual, 2008. http://www.cadsoft.de/.

[Cor08a] Atmel Corporation. Avr atmega32 – datasheet, 2008. http://www.atmel.com/avr.

[Cor08b] Microchip Corporation. Mcp4922 – datasheet, 2008. http://www.microchip.com/.

[Hon02] Jaroslav Honcu. Usage of piezoelements for damping of mechanical vibrations., 2002.

[Hon06] Jaroslav Honcu. Přednášky předmětu ipr., 2006.

[MIP08] inc. Maxim Integrated Products. Tl082 – datasheet, 2008. http://www.maxim-ic.com/.

[Sem08] Taiwan Semiconductor. 78lxx a 79lxx – datasheet, 2008. www.taiwansemi.com/.

[ss09] HW server s.r.o. Hardwarovy portal, 2009. http://hw.cz/.

[STM08] inc. STMelectronics. Tl082 – datasheet, 2008. http://www.st.com/.

Seznam použitých zkratek

MCU Micro Controler Unit

DAC Digital-Analog Converter

ISP In-System Progr Backus-Naur Form

OS Operační systém

RTOS Real-Time Operační Systém

TTL Tranzistorově-Tranzistorová Logika

DAC Digital-Anlog Converter

OZ Operační Zesilovač

JFET Junction gate Field-Effect Transistor

INL Integral Non-Linearity

DNL Diferential Non-Linearity

USB Universal Serial Bus

UART Universal Asynchronous Receiver/Transmitter

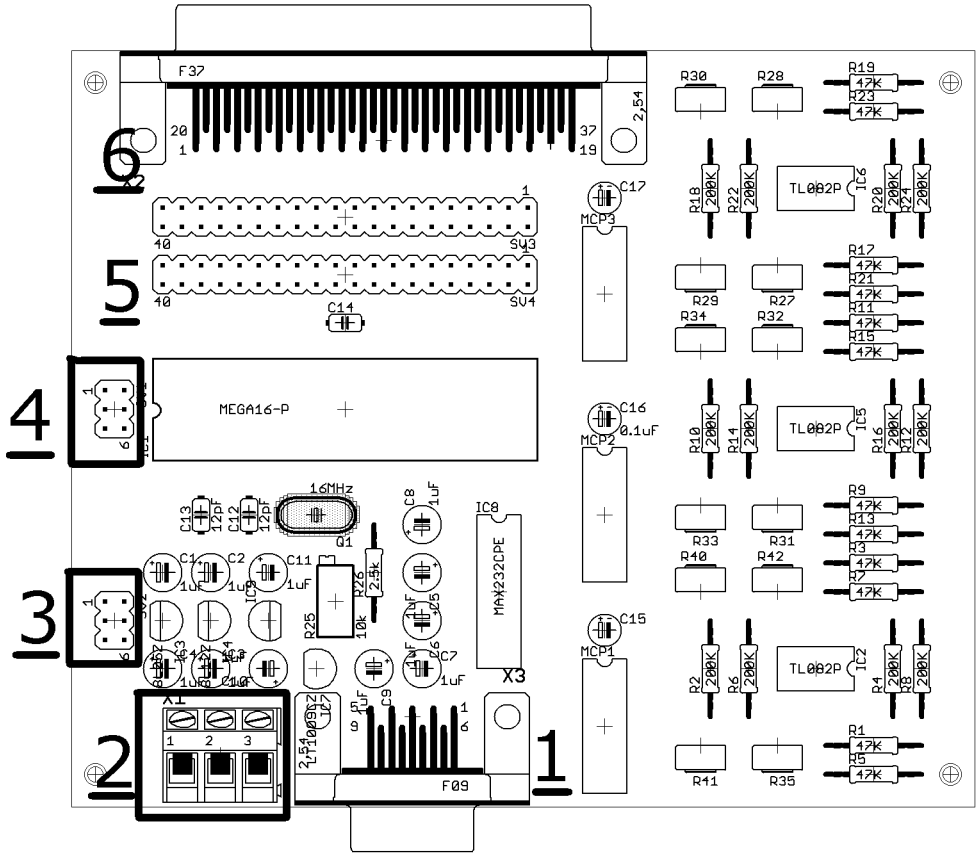
SRAM Static Random Access Memory

MIPS Million Instructions Per Second

ALU Arithmetic Logic Unit

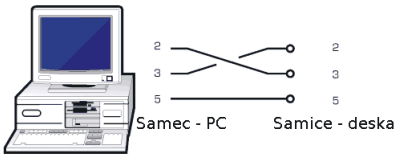
A Manuál k desce

Rozložení konektorů



Seznam konektorů:

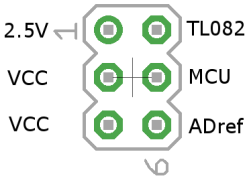
- 1 - konektor pro RS232 - samice



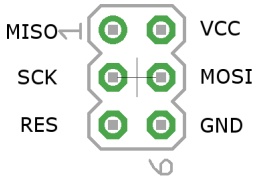
  Slouží k propojení pc s deskou - zbývající piny jsou propojené přímo

- 2 - konektor napájení
  Vstup č.1 - +15V
  Vstup c.2 - 0V
  Vstup č.3 - -15V

- 3 - propojovací konektor napájecích obvodů



- 4 - ISP konektor pro propojení s programátorem



- 5 - 40-ti pinová výstupní sběrnice
  Umožňují přepojení výstupů, při přímém propojení mají stejný pinout jako 6.
- 6 - 37 pin samice - výstup shodný s MF624

AD0	1	20	DA0
AD1	2	21	DA1
AD2	3	22	DA2
AD3	4	23	DA3
AD4	5	24	DA4
AD5	6	25	DA5
AD6	7	26	-12V
AD7	8	27	+12V
AGND	9	28	+5V
	10	29	GND
	11	30	DOUT0
DIN0	12	31	DOUT1
DIN1	13	32	DOUT2
DIN2	14	33	DOUT3
DIN3	15	34	DOUT4
	16	35	DOUT5
	17	36	DOUT6
	18	37	DOUT7
	19		

Návod k použití a důležitá upozornění

Konektor 3 při nezměněném napájení musí být vždy kompletně propojen.

U konektorů 3 a 4 je velice důležité, aby nedošlo k jejich záměně, protože by pravděpodobně došlo k pokození MCU, v horším případě i ke zničení programátoru.

U napájecího konektoru 2 nesmí dojít k omylu v zapojení ke zdroji, jinak se zničí napájecí obvody a OZ TL082.

Při manipulaci s přepojováním 40-ti pin konektorů vždy důkadně překontrolujte zapojení.

Programování MCU probíhá přímo na desce plošného spoje. MCU je napájen přímo z aplikace a není nutné přivádět napětí z programátoru. Kdyby bylo nutné programovat MCU s napájením z programátoru je nutné rozpojit konektor 3 před připojením programátou k desce.

Při inicializaci zařízení dochází ke krátkému 10us skoku do -10V na analogových výstupech zařízení. Proto desku nejprve zapněte a až poté připojujte Váš výrobek, protože by vlivem napěťového skoku mohlo dojít ke zničení vašeho zařízení.

Digitální výstupy nikdy nezatěžujte více než celkově 100mA a 20mA na jednotlivý výstup, mohlo by dojít k poškození MCU, nebo zresetování MCU. Napájecí +12V a -12V je stabilní do odběru 0.1A, díky použití přesných stabilizátorů 78L12 a 79L12.

B Komunikace s deskou

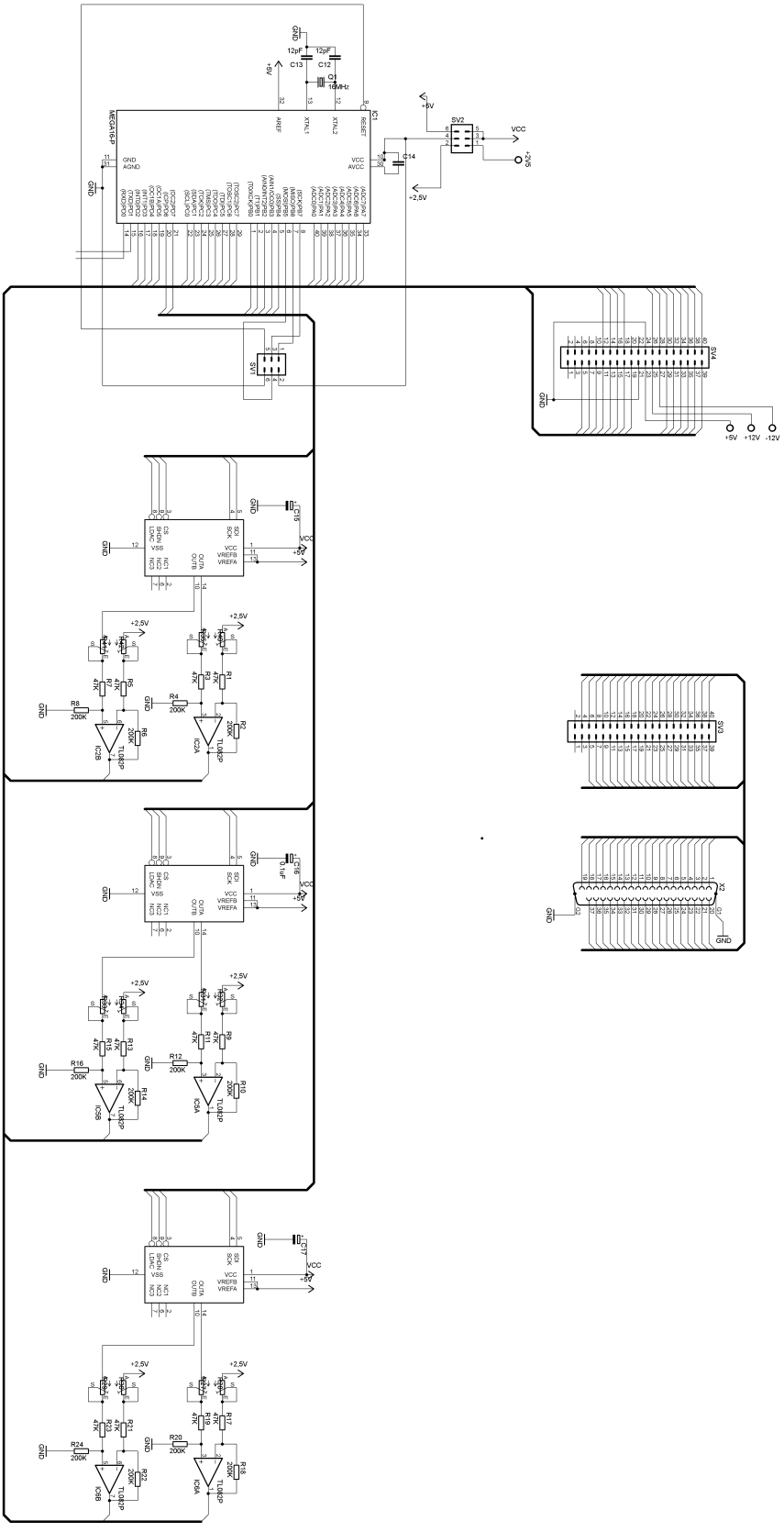
Pro komunikaci s deskou se používá RS232 rozranní. Program je odlazen pro přenosovou rychlost 9600baud 8 datbitů a 2 stopbity. Jelikož jsem nestihl naprogramovat knihovny pro prostředí matlab desku jsem testoval za použití terminálů v OS windows i linux. V následující tabulce je uveden přesný rozpis příkazů a formát dat, který má MCU naprogramován.

Funkce	ASCII ZNAK	1.byte	2.byte	3.byte
Analogový výstup	'V'	dec. číslo kanálu	spodní byte	horní byte
Init čítače	'C'	-	-	-
Frekvence čítače	'F'	spodní byte	horní byte	-
Digitální výstup	'S'	byte	-	-
Digitální výstup (místo ADC)	'W'	byte	-	-
Digitální výstup s maskou	'M'	maska	byte	-
Digitální vstup	'I'	-	-	-

Tabulka B.1: Popis komunikačních slov příkazů

Při volání fce. digitální vstup vyšle deska byte vstupních dat do PC. Digitální výstup (místo ADC) nastavuje výstupní sběrnici, která měla být původně vstupní, ale z důvodu nedostatku času nebyla dokončena.

C Schéma zapojení ostatních obvodů



D Obsah přiloženého CD

Na přiloženém CD se nacházejí tyto adresáře:

- resources - výrobní podklady pro desku (Eagle)
- documentation - dokumentační část bakalářské práce
- datasheet - obsahuje datasheety použitých součástek
- sources - zdrojové soubory programu pro MCU (AVR studio)