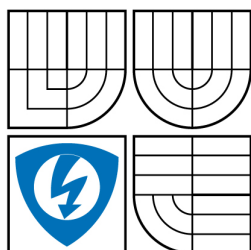


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# **MALÝ ŘÍDICÍ SYSTÉM S ROZHRANÍM ETHERNET**

SMALL CONTROL SYSTEM WITH ETHERNET BUS

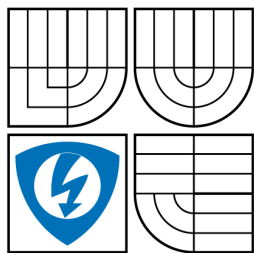
**BAKALÁRSKA PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**RADOVAN LIŠKA**

**VEDÚCI PRÁCE**  
SUPERVISOR

doc. Ing. ZDENĚK BRADÁČ, Ph.D



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Radovan Liška

**ID:** 73010

**Ročník:** 3

**Akademický rok:** 2008/2009

**NÁZEV TÉMATU:**

**Malý řídicí systém s rozhraním Ethernet**

## POKYNY PRO VYPRACOVÁNÍ:

Navrhněte malý řídicí systém s rozhraním Ethernet. Vybavte jej sadou digitálních a analogových V/V s průmyslovými standardními rozsahy. Systém založte na mikrokontroléru Rabbit. Elektroniku osadte, oživte a otestujte její funkčnost. Vytvořte plně komentované programové vybavení v C.

## DOPORUČENÁ LITERATURA:

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 1.6.2009

**Vedoucí práce:** doc. Ing. Zdeněk Bradáč, Ph.D.

**prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

### Abstrakt

Práca sa zaoberá komplexným návrhom a riešením (DPS) dosky plošných spojov pre malý riadiaci systém s rozhraním ethernet. V práci je rozobratý návrh napájacej časti obvodu, teda galvanické oddelenie, usmernenie a stabilizácia napätia. Pre komunikáciu snímačov, prípadne staníc s regulátorom zabezpečuje priemyselná komunikácia s rozhraním RS232 a RS485. Pre získanie obrazu o sústave je regulátor vybavený digitálnymi a analógovými vstupmi. Akčné zásahy regulátora realizujú analógové a digitálne výstupy. Zobrazovanie dát a konfigurácia regulátora sa rieši pomocou webového rozhrania cez http server.

**Kľúčové slová:** regulátor, ethernet, digitálne a analógové vstupy a výstupy, priemyselná komunikácia, web server

### Abstract

The thesis deals with complex design and realization of a PCB (printed circuit board) for a small control system with an ethernet connection. There is a section dealing with the design of a supply circuit which covers galvanic separation circuit, regulation and stabilization of voltage in this thesis. For communication with sensing units or other units with automatic controllers used in industrial communication is used RS232 and RS485 interface. Automatic controller acquires information from the driven system by analog and digital inputs. Actuators are controlled by analog or digital outputs from microcontroller. Data representation and configuration of controller is realized using an http server. Data is displayed in the appropriate web browser.

**Keywords:** regulator, ethernet, digital and analog inputs and outputs, industrial communication, web server

## Bibliografická citace

LIŠKA, R. *Malý řídicí systém s rozhraním ethernet*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 44 strán., 9 s. příloh. Vedoucí bakalářské práce doc. Ing. Zdeněk Bradáč, Ph.D.

## Prehla s e n i e

„Prehlasujem, že svoju bakalársku prácu na tému "Malý riadiaci systém s rozhraním ethernet" som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia § 152 trestného zákona č. 140/1961 Zb.“

V Brne dňa : 1 júna 2009

Podpis autora :

## Pod'a k o v a n i e

Ďakujem týmto svojmu vedúcemu bakalárskej práce doc. Ing Zdeňkovi Bradáčovi Ph.D rodičom a kamarátom za cenné pripomienky a rady pri vypracovávaní bakalárskej práce.

V Brne dňa : 1.júna 2009

Podpis autora:

## **OBSAH**

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| <b>1. ÚVOD .....</b>                                                  | <b>10</b> |
| <b>2. HARDWARE .....</b>                                              | <b>12</b> |
| 2.1 ÚVOD .....                                                        | 12        |
| 2.2 Modul rcm3000 rabbit core [1] .....                               | 12        |
| 2.2.1 Obsah modulu :.....                                             | 12        |
| 2.2.2 5V tolerancia pre vstupy .....                                  | 13        |
| 2.2.3 Sériové porty .....                                             | 13        |
| 2.2.4 Systémové hodiny(clock).....                                    | 14        |
| 2.2.5 Paralelné vstupy / výstupy (I/O).....                           | 14        |
| 2.2.6 Slave port .....                                                | 15        |
| 2.3 D/A A A/D PREVODNÍK .....                                         | 16        |
| 2.3.1 Úvod.....                                                       | 16        |
| 2.3.2 Vlastnosti prevodníků a kritéria zvažované pri ich výbere ..... | 16        |
| 2.3.3 Vstupné zapojenie.....                                          | 19        |
| 2.3.4 Prepočty súčiastok .....                                        | 21        |
| 2.4 RS-232.....                                                       | 23        |
| 2.4.1 Úvod.....                                                       | 23        |
| 2.4.2 Analýza a realizácia hardvéru .....                             | 23        |
| 2.5 RS – 485.....                                                     | 24        |
| 2.5.1 Popis a rozdiel oproti RS-232 .....                             | 24        |
| 2.5.2 Dizajn a návrh obvodu .....                                     | 24        |
| 2.6 DIGITÁLNE VSTUPY.....                                             | 25        |
| 2.6.1 Prepočty súčiastok digitálnych vstupov.....                     | 25        |
| 2.6.2 Rozšírenie o spínače .....                                      | 26        |
| 2.7 DIGITÁLNE VÝSTUPY .....                                           | 26        |
| 2.8 NAPÁJANIE .....                                                   | 28        |
| <b>3. SOFTWARE.....</b>                                               | <b>30</b> |
| 3.1 web prístup k mikroprocesoru .....                                | 31        |
| 3.1.1 TCP/IP protokol.....                                            | 31        |
| 3.1.2 Web stránka.....                                                | 37        |

|                                                              |           |
|--------------------------------------------------------------|-----------|
| 3.1.3 Popis aplikácie – Užívateľský manuál.....              | 39        |
| 3.2 Algoritmus PSD regulátora .....                          | 42        |
| 3.3 Komunikácia mikroprocesora a DAC.....                    | 44        |
| 3.3.1 Popis činnosti a kódu programu pre D/A prevodník ..... | 44        |
| 3.3.2 Postupnosť krokov vo funkcii .....                     | 46        |
| <b>4. ZÁVER .....</b>                                        | <b>48</b> |
| <b>5. ZOZNAM POUŽITEJ LITERATÚRY.....</b>                    | <b>50</b> |
| <b>6. ZOZNAM PRÍLOH .....</b>                                | <b>51</b> |

## ZOZNAM OBRÁZKOV

|                                                                                                                                           |    |
|-------------------------------------------------------------------------------------------------------------------------------------------|----|
| Obrázok 2-1 Kaskádne výstupné registre paralelných portov D a E.....                                                                      | 15 |
| Obrázok 2-2 Digitálna filtrácia vstupných pinov .....                                                                                     | 15 |
| Obrázok 2-3 schéma zapojenia A/D prevodníka [10] .....                                                                                    | 19 |
| Obrázok 2-4 Neinvertujúci zosilovač so spätnoväzbovou dolnou priepusťou 1. rádu<br>a) zapojenie b) Amplitudová charakteristika [11] ..... | 20 |
| Obrázok 2-5 Neinvertujúci zosilovač s dolnou priepusťou 1. rádu na vstupe a)<br>zapojenie b)amplitudová charakteristika [12] .....        | 22 |
| Obrázok 2-6 Schéma zapojenia digitálneho výstupu .....                                                                                    | 26 |
| Obrázok 3-1 Dátový tok medzi dvoma počítačmi prostredníctvom TCP/IP .....                                                                 | 31 |
| Obrázok 3-2 TCP/IP protokol [12] .....                                                                                                    | 32 |
| Obrázok 3-3 Minimálny obsah komponentov slúžiaci pre spustenie web<br>aplikácie [13].....                                                 | 36 |
| Obrázok 3-4 Úvod a prihlasovanie sa na stránku .....                                                                                      | 39 |
| Obrázok 3-5 Spojitá regulácia .....                                                                                                       | 40 |
| Obrázok 3-6 Diskrétna regulácia .....                                                                                                     | 41 |
| Obrázok 3-7 Nastavovanie Parametrov .....                                                                                                 | 41 |
| Obrázok 3-8 Simulinkovské schéma PSD regulátora s filtráciou derivačnej zložky<br>[14].....                                               | 42 |
| Obrázok 3-9 Blokové schéma D/A prevodníka [15] .....                                                                                      | 45 |
| Obrázok 3-10 Postupnosť zápisu dát [16].....                                                                                              | 46 |

## ZOZNAM SKRATIEK

| Skratka/Symbol | Popis                                        |
|----------------|----------------------------------------------|
| SPI            | Serial Peripheral Interface                  |
| MB/s           | Megabyte/sekundu                             |
| MBps           | Megabyte per second                          |
| mA             | miliampér                                    |
| mV             | milivolt                                     |
| tcp/ip         | transfer control protocol/ internet protocol |
| vs             | versus                                       |
| sd/mmc         | secure digital/multi media card              |
| PWM            | Pulse Width Modulation                       |
| CMOS           | complementary metal oxide semiconductor      |
| PSD            | proporciálno-sumačne-derivačný               |
| IRDA           | Infrared Data Association                    |
| EMI            | Electromagnetic Interference                 |
| ADC            | Analog to Digital converter                  |
| DAC            | Digital to Analog converter                  |
| ref            | Referenčné                                   |
| TUV            | Teplá užitková voda                          |
| OZ             | Operačný zosilňovač                          |
| ksps           | kilosamples per second                       |
| CSS            | Cascading style sheets                       |
| HTML           | Hypertext Markup Language                    |
| MIME           | Multipurpose Internet Mail Extension         |
| HTTP           | Hypertext transfer protokol                  |
| SCADA          | Supervisory control and Data acquisition     |
| SPI            | Serial peripheral interface bus              |

## 1. ÚVOD

Každým rokom rastie na svetovom trhu záujem o systémy s komponentami pre automatizáciu priemyselných procesov a neustáva ani v dobe prudkého rastu cien energií a širších obáv z inflácií. Priemyselná automatizácia je jeden z mojich záujmov a dôvod, prečo som si vybral danú tému. Môj produkt by mal pracovať samostatne z hľadiska regulácie, má teda obsahovať obvody a komponenty určené pre správu procesu. Jedná sa o akýsi embedded systém, postavený na riadiacom module RCM3000 RabbitCore. K tomuto modulu je treba vyrobiť základnú dosku, na ktorej bude umiestnený a na ktorej budú implementované komponenty pre komunikáciu cez RS232 alebo priemyselnú RS485. Na to, aby sme získali obraz meranej veličiny sú potrebné vstupy - či už analógové, alebo digitálne. Moje zariadenie bude obsahovať 8 digitálnych a 8 analógových vstupov. Na vygenerovanie odozvy, respektíve akčného zásahu regulátora, sústave som navrhol 8 digitálnych a 2 analógové výstupy.

Modul RCM3000 obsahuje už zabudované obvody pre správu pamäti a komunikáciu cez ethernet na rýchlostnej úrovni 10MB/s. Modul obsahuje procesor taktiež od firmy Rabbit.

Podobné zariadenia vyrábajú firmy ako Siemens, Schneider Electric a iné. Firma Siemens ponúka na trhu výrobok s označením „LOGO“. Zariadenie, mikrosystém, disponuje štyrmi analógovými vstupmi v základnej výbave, čo je možné neskôr modulárne rozšíriť o ďalšie. Tieto analógové vstupy/výstupy sú nastavené na priemyselný štandard 0/4 – 20 mA . Ja budem používať priemyselné analógové výstupy napäťové, na úrovni 0 - 10V.

Programovanie mikrosystému LOGO je umožnené pomocou blokových diagramov a tento software je vyvinutý pre rôzne platformy ako napríklad Microsoft Windows, Linux a Mac OS X. Ja sa budem snažiť vyvinúť plnohodnotný software pre platformu Microsoft Windows, pretože firma Rabbit ponúka množstvo knižníc a utilít, ktoré sú programované práve pod ňou. Moja predstava zo strany softwaru je, že systém bude môcť byť konfigurovaný cez ethernet port a bude sa naň dať prístupit' aj z internetu. To bude vyžadovať znalosť TCP/IP komunikácie a prenosu

dát a paketov po sieti. Webové rozhranie by som chcel naprogramovať v PHP, kvôli dynamickým odozvám systému a grafickému hľadisku tak, aby boli užívateľovi ľahko ovládateľné a zrozumiteľné. Webové rozhranie by som chcel rozdeliť na dve časti z hľadiska prístupu a možnosti konfigurácie a to na Užívateľskú časť s obmedzenými právami a Časť pre administrátora s plnými právami. Vzhľadom na to, že zariadenie má komunikovať cez internet by som chcel navrhovanú dosku rozšíriť o slot pre SD/MMC pamäťové karty, ktorého prostredníctvom by mohol byť spustený http server zariadenia, prípadne by mohol slúžiť ako úložisko dát. Bude treba dbať na správu reálneho času mikrokontroléru, koncept webového rozhrania a algoritmizáciu regulácie.

Po naprogramovaní tohto riadiaceho systému by malo byť možné toto zariadenie použiť v širokej škále priemyselných odvetví. Ako príklad môžeme spomenúť otváranie ventilov, vrát, ovládanie čerpadiel a kompresorov, zavlažovanie rastlín. Taktiež implementácia analógových funkcií je veľmi dôležitá pri udržovaní konštantnej teploty v miestnosti, výšky hladiny v nádrži a podobne.

Prácu na malom riadiacom systéme som rozdelil na dve časti. V semestrálnom projekte som sa zaoberal návrhom hardwaru. V tejto sekcii je rozobraná táto problematika spolu s výpočtami pre jednotlivé obvody a návrhmi. Pre pochopenie činnosti jednotlivých podblokov som uviedol stručné popísanie ich činnosti a ich možnosti. Ďalší semester je vyhradený pre oživenie a naprogramovanie tohto mikrosystému. Tento popis bude v sekcii software.

## **2. HARDWARE**

### **2.1 ÚVOD**

Pre realizáciu zariadenia je potrebné sa dôkladne oboznámiť s požiadavkami a nárokmi na zariadenia, a to z hľadiska presnosti, rýchlosti, a kvality, ktoré sú dominujúce v každej oblasti. V poslednom rade však treba zohľadniť aj pomer cena - výkon. Je zbytočné a nerozumné implementovať veci, ktoré sa nebudú využívať. Na matičnú dosku sa bude implementovať modul RCM3000, AD a DA prevodník pre zabezpečenie komunikácie procesora s IO zariadeniami, obvody pre priemyselnú komunikáciu RS232 a RS485, ďalej to budú obvody pre digitálne vstupy a výstupy, a slot na sd/mmc karty a napájanie ako rozšírenie. Všetky tieto subbloky budú navrhnuté s ohľadom na priemyselný štandard. Blokové schéma zariadenia je v prílohe 1.

### **2.2 MODUL RCM3000 RABBIT CORE [1]**

#### **2.2.1 Obsah modulu :**

- procesor Rabbit 3000, ktorý je taktovaný na frekvencii 29.4MHz
- Dva 34 pinové konektory poskytujú 52 digitálnych vstupov/výstupov z toho 44 je voľne konfigurovateľných, 4 pevné vstupy a 4 pevné výstupy. Zbernica označená ako Alternate, môže byť konfigurovaná pre 8 dátových a 6 adresových liniek(zdieľaných s paralelnými I/O linkami) a vstupy pre čítanie a zápis.
- Desať 8-bitových časovačov a jeden 10-bitový časovač s dvoma match registrami.
- 512 kilobajtová Flash pamäť a 512kB SRAM pamäť
- Hodinový clock
- Watchdog
- Možnosť zabezpečenia zariadenia voči výpadku napájania záložnou batériou.
- 10/100 kompatibilný RJ-45 ethernet konektor s 10MB/s rozhraním
- 10-bitový PWM čítač

- Šesť CMOS kompatibilných sériových portov : maximálna asynchrónna rýchlosť je 1,84Mbps , maximálna synchrónna rýchlosť 7,35Mbps. Štyri porty sú konfigurované ako SPI a dva ako SDLC/HDLC sériové porty.
- Podpora 1,15Mbps IRDA vysielача.

#### Výhody RCM300 :

Modul sa programuje pomocou jazyka C a je vybavený množstvom implementovaných funkcií pre pohodlnejšiu spoluprácu z hľadiska procesor – užívateľ. Vďaka vysokej rýchlosti hodinového signálu, (ďalej len clock), je modul schopný vykonávať matematické, logické a vstupno- výstupné operácie spoľahlivo a rýchlo. Veľká veľkosť pamäte umožňuje písať rozsiahle programy a zároveň slúžiť ako úložisko pre dáta. Integrovaný ethernet port je nevyhnutný pre sieťovú komunikáciu. Využitie TCP/IP programov.

Rabbit 3000 je dostupný v škole, preto ho budem používať.

Detailnejší popis On-Chip perifériei, ktoré budem využívať a ktoré sú v mojej aplikácii dôležité :

#### 2.2.2 5V tolerancia pre vstupy

Rabbit 3000 pracuje v rozsahu napätí od 1,8V do 3,6V, ale väčšina vstupných pinov je dimenzovaná až do 5V. Výnimkou sú napájacie piny a piny oscilátorového bufera. Keď privedieme 5V na tieto piny, tak sa začnú javiť ako piny s vysokou impedanciou hoci je Rabbit vypnutý. Táto 5V tolerancia umožňuje priamo pripojiť zariadenia pracujúce na 5 voltoch.

#### 2.2.3 Sériové porty

Sériové porty sú : A, B, C, D, E, F. Všetky môžu pracovať v asynchrónnom móde. Tieto asynchrónne porty využívajú 7, alebo 8-bitové dátové formáty, na základe toho, či sa parita používa, alebo nie. Deviaty bit je prídavný. Softwarový ovládač pre sériový port umožňuje podať informáciu o tom, kedy bol posledný bit zo shift registra poslaný. Toto je dôležité pre RS-485 komunikáciu, pretože half duplexový linkový ovládač nedokáže zistiť smer opačného posielania, pokiaľ nebol poslaný posledný bit. Vo väčšine UART zariadení je ťažké generovať prerušenie po

tom, ako je poslaný posledný bit. Adresový bit, taktiež nazývaný A bit, môže byť poslaný buď na vysokej, alebo nízkej úrovni, po tom, čo bol poslaný posledný dátový bit. Za adresovacím bitom nasleduje stop bit, ktorý je poslaný vysokou úrovňou. Porty A, B, C, D môžu byť konfigurované v sériovom, hodinovo riadenom režime. V tomto móde clock linky je synchronizovaný s clock vstupných, alebo výstupných dát. Clock môže zabezpečovať buď pripojené zariadenie, alebo môže byť sprostredkované Rabbitom. Keď clock poskytuje Rabbit, prenosová rýchlosť môže byť viac ako polovica systémovej clock frekvencie. Pokiaľ je však clock poskytovaný vonkajším zariadením, tak je maximálna prenosová rýchlosť systémovej clock / 6, kvôli potrebe synchronizácii externého a interného clock. Sériový mód riadený clock môže byť použitý pre zariadenia, ktoré využívajú zbernicu SPI. Port A má špeciálnu vlastnosť a tou je, že môže byť použitý pre cold-boot systému po reštarte. Taktiež sa môže využívať pre vývoj softwaru pod Dynamic C. Všetky porty používajú časovací mód, podporujúci komunikáciu cez infraport.

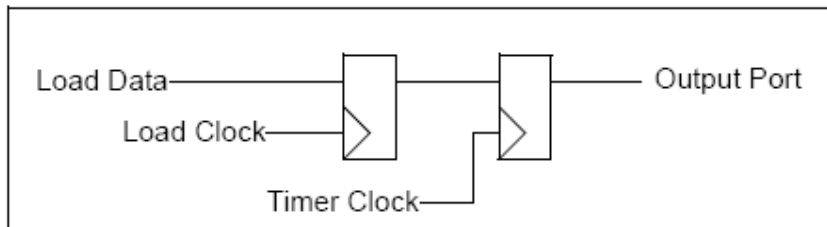
#### 2.2.4 Systémové hodiny(clock)

Hlavný oscilátor používa externý kryštál s frekvenciou v rozsahu od 1,8MHz do 26MHz. Procesorové hodiny sú odvodené od výstupu oscilátora, môžu byť teda rovné frekvencii oscilátora, alebo dvojnásobne veľkej frekvencii, prípadne delené frekvencia oscilátora / 2, 4, 6, 8. Procesorový clock môže tiež bežať na frekvencii 32,768Khz, čo je vlastne reálny čas clock oscilátora, pre nízko výkonové operácie, v ktorých hlavný oscilátor môže byť softwarovo vypnutý.

#### 2.2.5 Paralelné vstupy / výstupy (I/O)

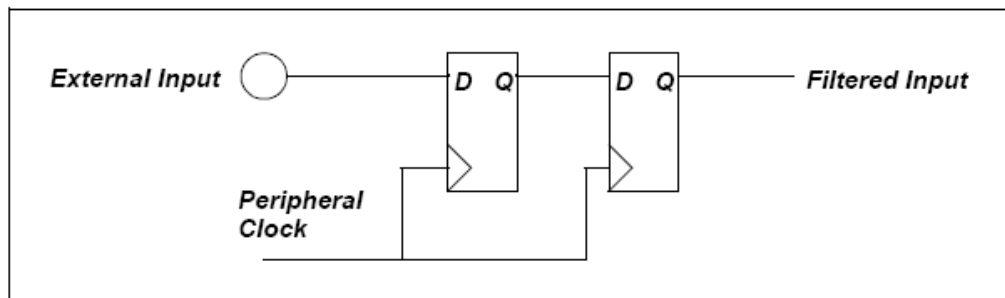
Modul obsahuje 56 paralelných I/O liniek rozdelené medzi sedem 8-bitových portov od A po G. Väčšina portov má alternatívne funkcie, ako napríklad sériové dátové impulzy, alebo čipovo výberové impulzy. Paralelné porty D, E, F, G majú možnosť časovo synchronných výstupov.

Výstupné registre sú kaskádne, ako je ukázané na obrázku 2-1 nižšie.



**Obrázok 2-1 Kaskádne výstupné registre paralelných portov D a E**

Ukladanie dát do portu sa vykonáva v prvej úrovni registra. Postupne sa tento obsah prevádza do výstupného registra po privedení signálu časovača. Clock môže byť vybraný ako výstup jedného z časovačov A1, B1, alebo B2. Časovací signál môže tiež vyvolať prerušenie, ktoré sa môže použiť na nastavenie ďalšieho bitu na výstup pri ďalšom pulze časovača. Táto vlastnosť môže byť použitá na generovanie presne kontrolovaných pulzov, ktorých hrany sú umiestnené s veľkou časovou presnosťou. Pomocou týchto I/O liniek sme schopní vytvárať signálovú komunikáciu, pulznú šírkovú moduláciu a riadiť krokové motory.



**Obrázok 2-2 Digitálna filtrácia vstupných pinov**

Vstupné piny do paralelných portov sú filtrované D flip flop registrami, ako je ukázané na obrázku 2.2. Toto zamedzuje pulzom kratším než periférny clock, aby boli rozpoznávané a následne synchronizujúce z externého do interného clocku a zamedzuje problému s meta stabilitou(dočasné nedefinované stavy).

### 2.2.6 Slave port

Slave port je navrhnutý tak, aby Rabbit mohol byť konfigurovaný ako slave druhým procesorom. Port je zdieľaný paralelným portom A, ktorý patrí medzi porty s

obojsmernou komunikáciou. V mojom prípade táto možnosť nebude využitá, pretože môj systém bude nastavený ako master.

## 2.3 D/A A A/D PREVODNÍK

### 2.3.1 Úvod

Pri výbere A/D a D/A prevodníkov je veľa vecí, ktoré treba pri návrhu hardwaru zvážiť. Ja som samostatne tieto prevodníky nevyrábal, avšak aj pri kúpe pre môj mikrosystém som musel vhodne vybrať najvhodnejší pre moju aplikáciu, vzhľadom k tomu že trh ponúka viacero druhov rôznych kvalít a rýchlostí. Výrobcov je niekoľko, ja som sa zameral na Microchip a Maxim. Nakoniec som sa rozhodol pre A/D a D/A prevodníky od firmy Microchip s označením MCP3208 pre A/D a MCP4922 pre D/A.

### 2.3.2 Vlastnosti prevodníkov a kritéria zvažované pri ich výbere

Pri vysvetľovaní sa budem zameriavať na ADC. DAC je na podobnom princípe a sú zvažované tie isté pravidlá. Čo to vlastne ADC je, čo to znamená? ADC je z anglického „Analog to digital converter“, a znamená to prevod jedného typu signálu prevádzaného, na druhý typ signálu. Môžeme ho považovať teda za zariadenie, v ktorom sú aj digitálne aj analógové signály. Vstup analógového signálu je reprezentovaný digitálnym výstupom.

Rovnica výstupu je 
$$výstup = \frac{2^n * G * A_{in}}{V_{ref}} \quad (2.1)$$

Kde- n je počet bitov prevodníka

G je zosilovací faktor

$A_{in}$  je analogový napät'ový, alebo prúdový vstup

$V_{ref}$  je referenčné napätie, alebo referenčný prúd

Referenčné napätie/prúd slúži k na porovnávanie oproti vstupnému napätiu/prúdu. Na základe toho výstupné slovo povie, ktorá frakcia referenčného napätia/prúdu

najlepšie vystihuje vstup.  $G$  je zvyčajne jedna. Avšak táto premenná je pomerne nová a prišla do tejto rovnice s vyvinutím nových AD prevodníkov od firmy National Semiconductor.

Táto rovnica bola jedným z kritérií pri výbere A/D a D/A prevodníka. O čo vlastne ide? Je známe, že viac bitov dáva väčšie rozlíšenie a menšie kroky, čím sa výstupné slovo viac podobá vstupnému signálu a signál je presnejší.

$$\text{Rovnica rozlíšenia jednotlivých krokov prevodníkov : rozlíšenie} = \frac{V_{ref}}{2^n} \quad (2.2)$$

Toto rozlíšenie je tiež rovné LSB bitu. Nižšie referenčné napätie dáva menšie kroky pre komparáciu signálov, ale na druhej strane to môže byť na úkor šumu. Nastáva tu teda problém pri voľbe ref napätia, pretože jeho znížením strácame dynamický rozsah vstupného signálu. Môže sa nám teda stať že pri malých úrovniach signálu sa nám daný signál úplne stratí v šume.

Prevodníky sú napájané 3.3V s dc/dc meniča traco a majú 12 bitové rozlíšenia. Referenčné napätie som volil menšie s hodnotou 2.5V. Táto hodnota sa bude môcť dať nastaviť pomocou obvodu TL431 na zdroji referenčného napätia, v závislosti od výskytu problémových stavov pri realizácii hw. Na vstup prevodníka privádzam 8 analógových signálov. Tieto vstupy môžu byť konfigurované ako diferenciálne – teda vstupy sa budú párovo porovnávať. Ja využívam zapojenie individuálnych vstupov. Maximálna vzorkovacia frekvencia bude pri 3V okolo 70 ksp/s. Výhoda prevodníka, je že je postavený na CMOS technológií, čo sa odzrkadľuje aj pri odbere prúdu, len niekoľko mikroampér pri v aktívnom režime. V standby režime sú to rádovo nanoampére. Keďže mikrosystém má byť čo najúspornejší, tak táto vlastnosť je vítaná. Ďalšie parametre, ktoré som zvažoval pri výbere : Veľkosť jednotlivých chýb, nelinearít a odchýliek. Niektoré z nich vysvetlím.

- Offset error : Je to konštanta, ktorá sa dá ľahko skalibrovať. V ideálnych prípadoch je to priamka, každému „kvantu“ (kvantum = rozsah vstupných hodnôt, ktoré reprezentuje výstupný binárny kód,) prísuší určitý kód. Táto

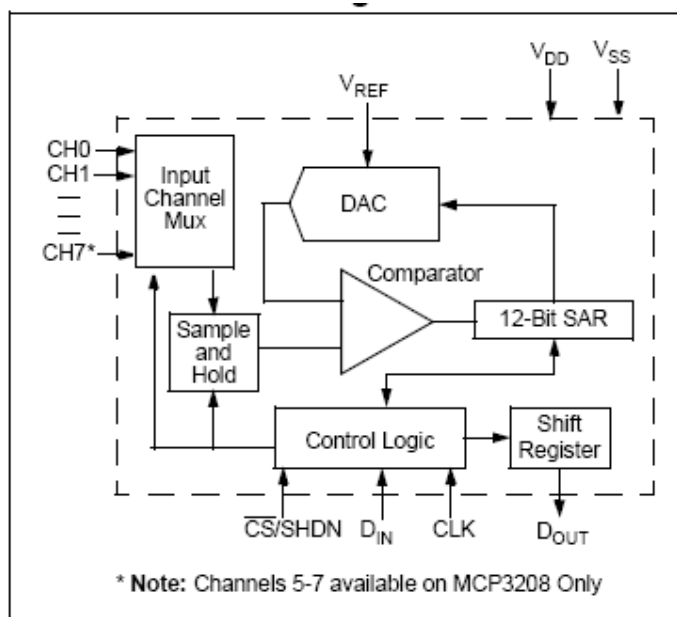
závislost' je lineárna priamka . a offset tuto znamená, o koľko je posunutá od začiatku , teda od nuly. U môjho A/D prevodníka sú to 3 najmenej významné bity.

- Gain Error : odchýlka od ideálneho sklonu prenosovej funkcie. Na vysvetlenie- : Ak posunieme našu aktuálnu prenosovú funkciu do nulovej ofsetovej odchýlky, rozdiel medzi aktuálnou a ideálnou prenosovou funkciou vzhľadom k celkovému rozsahu stupnice je Gain Error.
- DNL (differential Non-Linearity) : parameter, ktorý opisuje chybu v šírke kroku. Indikuje odchýlku od ideálneho 1 LSB šírky kroku analógového vstupného signálu vzhľadom na odpovedajúci binárny kód. Niekedy sa stane, že nejaký kód v prenosovej funkcii chýba. Toto som tiež zohľadňoval pri výbere. Na to, aby sa zamedzilo takejto situácii je treba dbať na to, aby daný prevodník mal DNL väčšie (viac pozitívne) ako -1.0 LSB. Môj vybraný prevodník má DNL  $\pm 0.5$  LSB.
- INL (Integrated Non-Linearity) : Vyjadruje vzdialenosť od ideálnej prenosovej funkcie A/D alebo D/A prevodníka. Nezahŕňa kvantizačné chyby, ofsetové chyby, gain error. INL môže byť väčšia než DNL.
- Ďalším parametrom je SINAD (signal to noise and distortion) čo je vlastne kombinácia SNR a THD, SNR(signal to noise ratio) : jedná sa o pomer výstupnej amplitudy signálu ku výstupnej úrovni šumu, pričom SNR nezahŕňa vyššie harmonické alebo jednosmernú zložku signálu.
- Parameter SINAD nám pomôže pri kalkulovaní efektívneho počtu bitov.

$$ENOB = \frac{(SINAD - 1.76)}{6.02} \quad (2.3)$$

V mojom prípade je to 12.66 bitov. Pri výbere prevodníkov existuje ešte mnoho iných pojmov, ktoré treba zvážiť. Ja som uviedol len zopár do mojej písomnej práce, avšak pri návrhu boli zohľadňované. Detailnejší popis možno získať na stránkach výrobcov. Môj zdroj bol microchip datasheet.

Prevodník pracuje na princípe postupnej aproximácie a patrí medzi sériové prevodníky. Pomocou rozhrania SPI je ho možné konfigurovať. Usporiadanie jednotlivých blokov v AD prevodníku je na obrázku č. 2.4



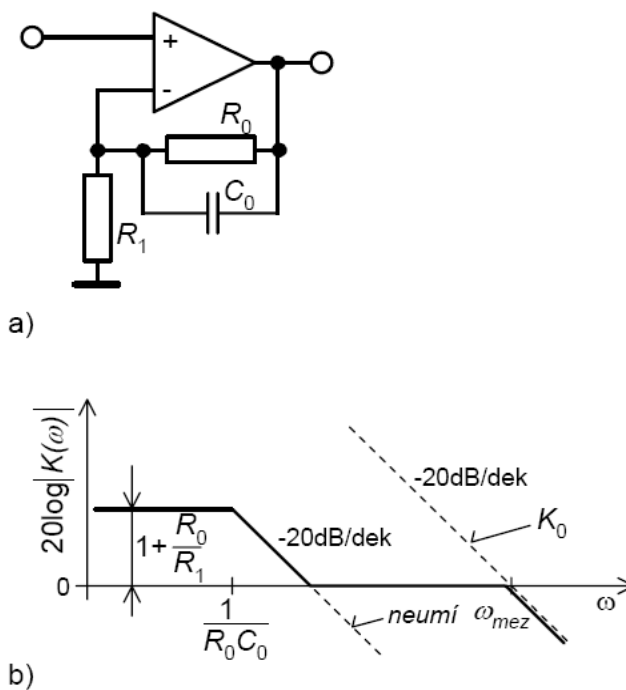
Obrázok 2-3 schéma zapojenia A/D prevodníka [10]

### 2.3.3 Vstupné zapojenie

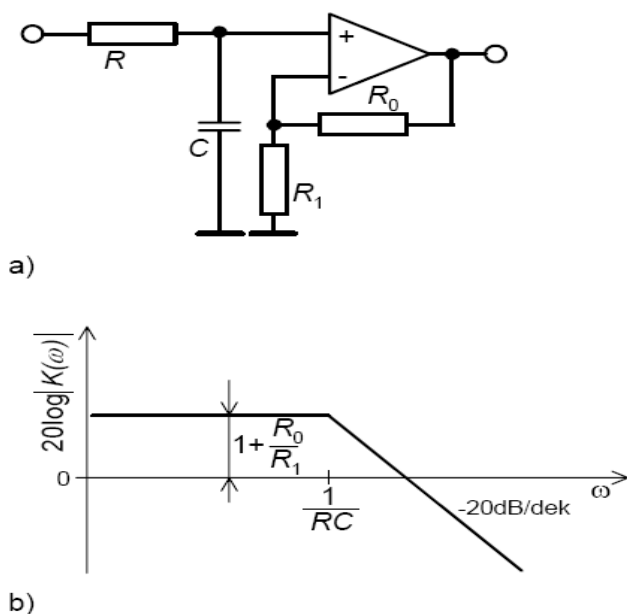
Na vstupe privádzam signál cez svorkovnice wago, jednotlivé vstupy sú chránené proti vyšším napätiam ako je povolené, pomocou transil. V tomto mikrosystéme používam na vstup aj výstup jednotlivých prevodníkov napäťové slučky v rozsahu 0-10V. Vstupný signál ďalej privádzam do operačného zosilňovača, ktorý je zapojený v neinvertujúcom zapojení zo vstupnou dolnou priepusťou na obmedzenie frekvencií nad 2Khz, ktorá stačí pre moje aplikácie. Zapojenie operačného zosilňovača v neinvertujúcom smere bolo vybrané kvôli tomu, že neobracia polaritu signálu. Taktiež výhodnejšie sa ukázalo zapojiť dolnú priepusť nie do spätnej väzby, ale pred vstup signálu do oz. Priepusť je nezaťažená, pretože vstupná impedancia OZ je prakticky nekonečná. Výhody tohto zapojenia oproti zapojeniu dolnej priepusti v spätnej väzbe sú :

- Pasívny RC článok nemôže zakmitávať a odozva na jednotkový skok je omnoho čistejšia
- Amplitúdová charakteristika nemá zlom na ose 0 dB, ale plynulo pokračuje so strmosťou -20 dB i na vysokých frekvenciách

Porovnanie je na obrázkoch 2.4 a 2.5.



**Obrázok 2-4** Neinvertujúci zosilovač so spätnoväzbovou dolnou priepusťou 1. rádu a) zapojenie b) Amplitudová charakteristika [11]



**Obrázok 2-5** Neinvertujúci zosilovač s dolnou priepust'ou 1. rádu na vstupe a)  
zapojenie b) amplitudová charakteristika [12]

Ako poslednou časťou pred vstupom do prevodníka prechádza signál trímrovaným napäťovým deličom. Bolo treba previesť napäťovú úroveň vonkajšieho obvodu, ktorý pracuje v rozsahu 0-10V na odpovedajúci rozsah A/D prevodníka, v mojom prípade 0-2.5V.

#### 2.3.4 Prepočty súčiastok

Vstupné transili sú nastavené na 12V ako ochrana proti prepätiam, nakoľko vstup do A/D prevodníka je prepočítavaný v rozsahu 0-10V.

Operačné zosilňovače sú ako komponenty štyroch OZ v jednom puzdre. Jedná sa o LM324N. OZ som vyberal na základne malej spotreby a na rail-to-rail prevádzke. Sú napájané napätím 12V. Sú využité ako som už spomínal spolu s odpormi a kondenzátormi ako dolná priepusť.

- Návrh filtra dolnej priepusti : Shannonov vzorkovací teorém vraví, že pokiaľ chceme rekonštruovať spojité signál, tak ho musíme navzorkovať s frekvenciou minimálne dvakrát vyššou, ako je

maximálna frekvencia rekonštruovaného signálu. To znamená- pozrel som si na akej vzorkovacej frekvencii bude približne pracovať môj ADC (max 70ksps odvodené z datasheetu) a z toho mi vyšla maximálna frekvencia rekonštruovaného signálu (35kHz). Pre moje aplikácie stačí pracovať s frekvenciou do 2kHz. Z rovnice 4 si vyjadrím R a kondenzátor vhodne zvolím a vypočítam R104.

$$f = \frac{1}{2\pi RC} \quad (2.4)$$

$$R = \frac{1}{2\pi f C} = \frac{1}{2\pi * 2000 * 1 * 10^{-9}} = 79k\Omega$$

Podľa normy a tried presnosti som tento odpor prispôbil na 82kΩ. Odpory R17 a R32 som volil tak, aby zlomok z rovnice 5 dal čo najmenšie číslo, ideálne 0, pretože ja nepotrebujem signál zosilňovať, ale zoslabovať. R17 som teda pridal 100Ω a R32 100kΩ.

$$K = \frac{U_0}{U_1} = 1 + \frac{R_0}{R_1} \quad (2.5)$$

Kde K = napäťový prenos

$$R_0 = R17$$

$$R_1 = R32$$

- Napäťový delič pozostáva z odporov R15, R16 a trímru PT10S2.

Výpočet bol prevedený tak, aby delič delil napätie v pomere 1:4.

Napätie do A/D prevodníka je prepočítané nasledovne :

$$U_2 = \frac{R_{16} + R_{tr}}{R_{16} + R_{tr} + R_{15}} \quad (2.6)$$

Hodnoty boli volené R16 = 2k40, R15 = 9k10, Rtr = 1k. U D/A prevodníka som vychádzal z obdobných poznatkov.

## 2.4 RS-232

### 2.4.1 Úvod

Štandard definuje asynchrónnu sériovú komunikáciu pre prenos dát. Poradie prenosu dátových bitov je od najmenej významného bitu LSB, po najvýznamnejší bit MSB. Počet dátových bitov je voliteľný, obyčajne sa používa 8. Logický stav 0/1 je prenášaný pomocou dvoch napäťových bipolárnych úrovní a môžu nadobúdať úrovne  $\pm 5V$ ,  $\pm 10V$ ,  $\pm 12V$ ,  $\pm 15V$ . Najčastejšie sa používa varianta, pri ktorej logickej hodnote 1 odpovedá  $-12V$  a logickej hodnote 0  $+12V$ . Základné tri vodiče rozhrania (príjem RxD, vysielanie TxD a spoločná zem) sú doplnené ešte ďalšími slúžiacimi k prenosu (vstupy DCD, DSR, CTS, RI, výstupy DTR, RTS). Tieto nemusia, ale môžu byť používané. Výstupná elektronika je vybavená ochranou proti skratu, keby prúd po prekročení 20mA ďalej narastal.

Sériový port je známy pod názvami ako sériová linka, či štandard RS-232. Patrí medzi komunikačné rozhranie, ktoré som implementoval do môjho mikrosystému. Sériový port je už zastaraná záležitosť, avšak sériová komunikácia sa často krát využíva i dnes avšak v priemysle majú väčší význam jej modifikácie ako napr. štandardy RS422, alebo RS485. Pre jednoduchú komunikáciu medzi dvoma zariadeniami typu - Odoslať a prijať dáta, je sériový port jednoduché a lacné riešenie.

### 2.4.2 Analýza a realizácia hardvéru

K realizácii sériovej komunikácie je potreba okrem SPI rozhrania na danom procesore aj budič spolu s prijímačom. Na doske som použil integrovaný obvod MAX3232, ktorý obsahuje budič spolu s prijímačom a ďalšími potrebnými súčiastkami. Tento zabudovaný vysielateľ - prijímač tzv. transceiver (prevodník) pracuje ako prevodník TTL (CMOS) /RS-232 podľa normy EIA/TIA-232. Je napájaný 3.3V. Vo vypnutom režime má odber len 1 $\mu$ A. Výrobca garantuje prenosovú rýchlosť 120kbps pre tento typ prevodníka. Obsahuje 2x vysielateľ a 2x prijímač. Ja využívam len jeden pár transceivera. Je pravda, že by stačil len jeden transceiver, ale v dobe návrhu a pri kúpe súčiastok, dané komponenty neboli na sklade a na objednávku by sa dlho čakalo. Na druhej strane firma Maxim vyrába

kvalitné komponenty jednak z hľadiska spotreby, i z pomeru cena – výkon. Prevodník používa 4x externé kondenzátory, ako nábojovú pumpu, ktorá slúži na zvýšenie výstupných napäťových úrovní RS232. Kondenzátory som nadizajnoval podľa odporúčania výrobcu.

## **2.5 RS – 485**

### **2.5.1 Popis a rozdiel oproti RS-232**

Keď potrebujeme poslať malé bloky informácii na väčšiu vzdialenosť, RS-485 je určite dobrý výber. Ako jednotlivé uzly pre túto sieť slúžia či už počítače, mikrokontroléry, alebo hocijaké zariadenia schopné asynchrónnej komunikácie. V porovnaní s ethernetom je RS-485 sú nároky na protokol jednoduchšie a lacnejšie.

RS-485 je vlastne elektrická špecifikácia pre viacbodové systémy, ktoré používajú symetrické vedenia. To je jedna z výhod oproti klasickej RS-232, pretože v symetrickom vedení sa šum odčíta a amplitúda signálu sa zväčší. V dokumente špecifikácii pre štandard RS-485 (TIA/EIA-485-A) sú definované elektrické charakteristiky z jednotlivých liniek, i z prijímačov a vysielačov vrátane odporúčaných zapojení. Vzdialenosť jednotlivých uzlov v sieti môže byť viac ako 1200m a prenosová rýchlosť môže dosiahnuť viac ako 10Mbps. Samozrejme, že čím väčšia vzdialenosť, tým menšiu rýchlosť dostaneme.

### **2.5.2 Dizajn a návrh obvodu**

Na dosku by som chcel implementovať nízko príkonové transceiver s napájacím napätím max 3.3V. Čipy jednotlivých výrobcov majú označený neinvertujúci signál ako linka A a invertujúci signál ako linka B. Medzi týmito linkami je napäťový rozdiel 200mV. 200mV je vlastne hladina, kde nie je výstupný signál definovaný. Pokiaľ je signál A väčší o viac ako 200mV od signálu B, výstupom je logická 1 a naopak. Výstupný obvod z čipu RS-485 je veľmi jednoduchý. Signály zo svorkovnice A a zo svorkovnice B vedú priamo do konektora RJ11, odkiaľ sa bude viesť, najlepšie kríženou dvojlinkou, (kvôli rušeniu,) k inému zariadeniu. Paralelne k vodičom A a B som zapojil odpor s hodnotou 120Ω. Tento odpor je nazývaný tzv. ukončovací a slúži k redukcii napäťových odrazov od konca

vedenia. Hodnota je vypočítaná nasledujúcou úvahou : Pre digitálne signály, ktoré majú frekvenciu väčšiu než 100kHz sa impedancia vodiča správa viac ako rezistor, indukčné a kapacitné zložky sú malé. Typická hodnota odporu kríženej dvojlinky je 120Ω. Tento odpor je však v sérii s prepojkou. Pokiaľ by bol tento transceiver použitý ako koncový tak by sa odpor uplatnil. V opačnom prípade bude prepojka rozopnutá a odpor sa nebude uplatňovať.

## 2.6 DIGITÁLNE VSTUPY

### 2.6.1 Prepočty súčiastok digitálnych vstupov

Základom digitálnych vstupov je optočlen, ktorý slúži ako galvanické oddelenie pre vonkajší vstupný obvod a pre obvod mikroprocesoru. Optočlen som vybral od firmy Vishay, pretože ponúkajú v jednom puzdre 4 optočleny, ktoré obsahujú fototranzistor opticky spravený s gálium-arsenidovou diódou. Vstupná časť obvodu obsahuje transil ako ochranu proti prepätiam, tu som zvolil hodnotu transilu na max 30V, vzhľadom k tomu, že vstupy sú nastavené na prácu v priemyselnom rozsahu 0-24V.

Výpočet pre odpor

$$R_{90} = \frac{U}{I} = \frac{U_{nap} - U_{diody}}{I_{diody}} = \frac{24V - 1.25V}{5 \cdot 10^{-3} A} = 4550\Omega \quad (2.7)$$

Vzhľadom na triedy presnosti je odpor zvolený  $R_{90} = 4k70$ .

Vnútorň obvod tvorí nízkopríkonová svietiaci led dióda a dva odpory. R86 pre nastavenie prúdu diódou a R88 pre „procesor“. Tento odpor je volený veľký, kvôli tomu aby na procesore nevznikali nedefinované stavy. Vstup procesora pri nečinnosti digitálneho vstupného obvodu je držaný na log. jednotke, čo v tomto prípade predstavuje 3.3V. Toto napätie je teda aj na odpore R88 . Keďže procesor má veľmi veľký vstupný odpor tak sa do neho neprivádzajú veľké prúdy. Sú to rádovo mikroampére. R88 je navrhnutý na hodnotu 100kΩ.

$$\text{Odpor pred diódou : } R_{86} = \frac{U_{nap} - U_{diody} - U_{ce}}{I_{diody}} = \frac{3.3V - 1.65V - 0.6V}{2 \cdot 10^{-3} A} = 525\Omega \quad (2.8)$$

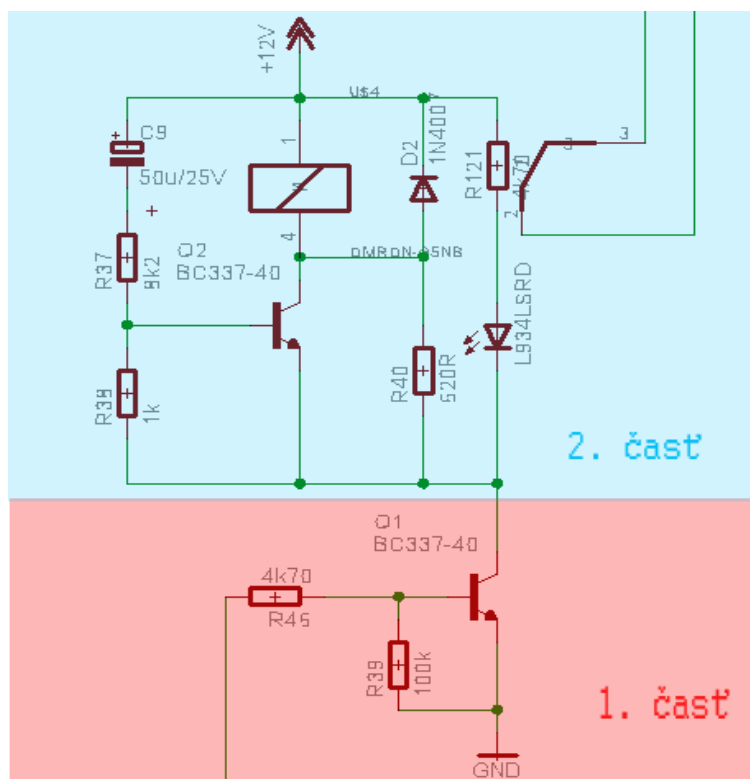
Podľa triedy presnosti R86 = 470R

### 2.6.2 Rozšírenie o spínače

Vstupy procesora som využil na pripojenie spínačov. Pokiaľ nebude daný výstup práve používaný regulátorom v čase regulácie bude aktivovaný vstup pre spínač pomocou, ktorého sa bude dať spustiť príslušný výstup manuálne.

## 2.7 DIGITÁLNE VÝSTUPY

Obvod tvoria dve hlavné časti. Prvá časť obsahuje tranzistor Q1, ktorý pracuje v spínacom režime a má za úlohu spínať nadväzujúci pripojený obvod. Druhá časť je spínaná časť a jej úlohou je spínanie reléového výstupu, ďalej len relé, a rozsvietenie led diódy.



Obrázok 2-6 Schéma zapojenia digitálneho výstupu

Toto zapojenie som vybral z dôvodu spotreby elektrického prúdu pri spínaní relé. Zvažujme situáciu : Pri dlhom zapnutí digitálneho výstupu, ktoré môže nastať napríklad pri filtrácii bazéna, alebo obehu čerpadla pri kúrení na TUV je pri klasickom zapojení relé spotreba energie závislá hlavne na jeho odbere. Toto zapojenie umožňuje po uplynutí časovej konštanty  $\tau = R37 \cdot C9$  znížiť spotrebu systému.

**[9]** Popis činnosti 2.časti obvodu : Obvod pracuje tak, že po privedení napájacieho napätia sa začne nabíjať kondenzátor C9 prúdom, ktorý preteká obvodom bázy tranzistora Q2. Týmto prúdom sa tranzistor otvorí a relé pritiahne. Behom krátkej doby, ktorá je daná časovou konštantou  $\tau = R37 \cdot C9$ , sa prestane kondenzátor nabíjať a tranzistor sa uzavrie. Od tohto okamžiku je cievka relé napájaná iba cez odpor R40, ktorý obmedzí prúd.

$$\text{Odpor, ktorý obmedzí prúd : } R40 = \frac{U_{nap} - (U_{od} + 1)}{I_{od}} = \frac{12V - (5 + 1)}{10 \cdot 10^{-3} A} = 600\Omega \quad (2.9)$$

$U_{nap}$  = napájacie napätie

$U_{od}$ ,  $I_{od}$  = napätie, prúd pri odpade kotvy.

Odpor podľa normy  $R40 = 620R$  . Časovú konštantu  $\tau = R37 \cdot C9$  som nastavil ako v návode, pretože stačí na dobu odpadu môjho relé.  $R37 = 8k2$  ,  $C9 = 50\mu F$ . Cievka relé je chránená proti odpáleniu pomocou ochrannej diódy D2.

Na signalizáciu stavu činnosti som opäť použil nízkopríkonové diódy. Odpor pre nastavenie prúdu diódou :

$$R121 = \frac{U}{I} = \frac{U_{nap} - U_{diody} - U_{ce}}{I_{diody}} = \frac{12V - 1.65V - 0.6V}{2 \cdot 10^{-3} A} = 4875\Omega \quad (2.10)$$

Kolektorový prúd tranzistoru Q1 :

$$I_c = I_d + I_{relé} = 2 \cdot 10^{-3} A + 16.7 \cdot 10^{-3} A = 18.7 \cdot 10^{-3} A = 18,7mA \quad (2.11)$$

$\beta$  (prúdový zosilňovací činiteľ) som volil 30.

$$\text{Bázový prúd } I_{\beta 1} = \frac{I_c}{\beta} = \frac{18.7 \cdot 10^{-3} A}{30} = 6.2 \cdot 10^{-4} A \quad (2.12)$$

Procesor dokáže na výstupe výstupných pinov dodávať prúd max 6mA. Pri počítaní odporu  $R_{be}$  budem jeho prúd dopočítavať do 3mA.

$$\text{Prúd } I_{be} = 3\text{mA} - 0.6\text{mA} = 2.4\text{mA}$$

$$I_{\beta} = I_{\beta 1} + I_{be} = 3\text{mA}$$

$$\text{Odpor } R_{45} = \frac{U_{proc} - U_{be}}{I_b} = \frac{0.7V * 3.3V - 0.7V}{3 * 10^{-3} A} = 537\Omega \quad (2.13)$$

$$R_{39} = \frac{U_{be}}{I_{be}} = \frac{0.7V}{2.4 * 10^{-3} A} = 292\Omega \quad (2.14)$$

Podľa normy  $R_{45} = 680\Omega$  ,  $R_{39} = 330\Omega$

## 2.8 NAPÁJANIE

Hlavnou súčasťou každého systému je napájanie. Môj mikrosystém využíva celkovo tri napájacie zdroje. Prvý je traco realizovaný pomocou dc/dc meniča traco . Tento stabilizovaný zdroj som vybral na základe dobrej účinnosti a z hľadiska galvanického oddelenia. Na svojom výstupe dáva napätie 3.3V s maximálnym prúdovým odberom 500 mA. Toto napätie sa využíva na napájanie väčšiny integrovaných obvodov , vrátane modulu Rabbit 3000. Z tohto napätia je ďalej napätie stabilizované na 2.5V, ktoré slúži ako referenčné napätie pre A/D, D/A prevodníky. Napätie sa nastavuje pomocou obvodu TL431. Ako druhý zdroj je obvod 7812, ktorý poskytuje napätie pre jednotlivé relé. Tretí zdroj je tiež traco dc/dc menič s výstupným napätím 12V a slúži na napájanie operačných zosilňovačov.



### 3. SOFTWARE

V každom procese je dôležitá jak akvizícia dát tak ich spracovanie, vyhodnotenie a vizualizácia tzv. SCADA systém. Preto pre môj systém som implementoval nasledovné algoritmy, pomocou ktorých dokážem predchádzajúce požiadavky splniť.

Jedná sa o :

- Webové rozhranie pre kontrolu a riadenie
- Algoritmus diskrétného PSD regulátora
- Algoritmus pre jednoduché riadenie PLC

Pomocou týchto algoritmov som schopný moje zariadenie využiť pre riadenie spojitých procesov ako napríklad meranie teploty, riadenie rýchlosti jednosmerného motora a iné. Spojitá veličina sa najskôr navzorkuje, prevedie na binárne číslo s rozlíšením A/D prevodníka, cez SPI komunikáciu sa prevedie do mikroprocesora, kde sa PSD algoritmom prepočíta akčný zásah, ktorý sa posiela opäť cez synchrónnu SPI do D/A prevodníka, ktorý prepočíta a nastaví hodnotu napätia pre výstup do procesu. Ďalšou výhodou tohto systému je voľba PLC režimu, kedy som schopný riadiť a spracovávať diskkrétne udalosti. Na základe impulzov z diskrétnych snímačov regulátor reaguje príslušným povelením, akciou do procesu tým, že ovláda až 8 reléových vstupov či už zaraz, alebo jednotlivo.

Ovládanie regulácie je možné z webového rozhrania. Web je tvorený na spôsob demonštrácie systému regulácie a teda nie všetky vstupy a výstupy sú použité.

Veľkým prínosom v programovaní je, že firma Rabbit ponúka množstvo knižníc už s preddefinovanými makrami a funkciami a taktiež dosť praktických príkladov, na ktorých sa dá overiť a vyskúšať si prácu a činnosť jednotlivých knižníc. V mojom prípade sa jedná o :

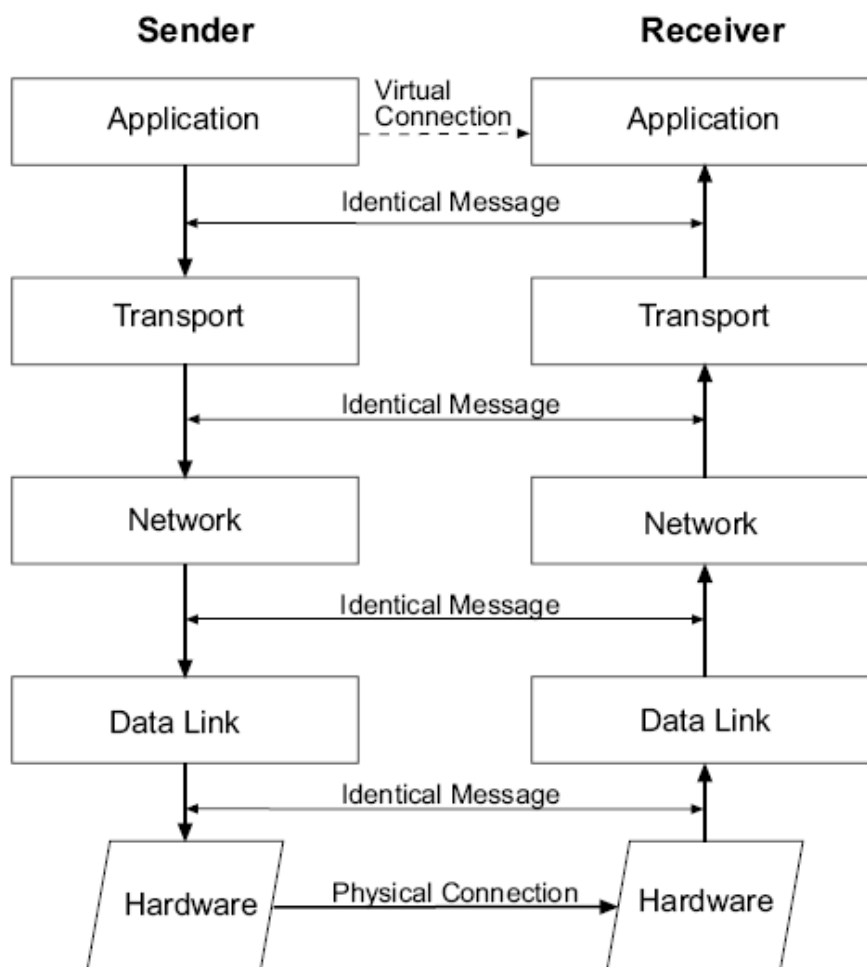
- TCP/IP komunikáciu pre web rozhranie
- Využitie Zservra spolu s http
- Funkcie `WrPortI` a `RdPortI`, `BitWrPortI` a `BitRdPortI` pre zápis a čítanie z registrov

### 3.1 WEB PRÍSTUP K MIKROPROCESORU

Kapitola sa zaoberá zobrazovaním nameraných a spracovaných dát na webové rozhranie a tiež ovládanie výstupov systému. Základným prvkom pre komunikáciu je tcp/ip.

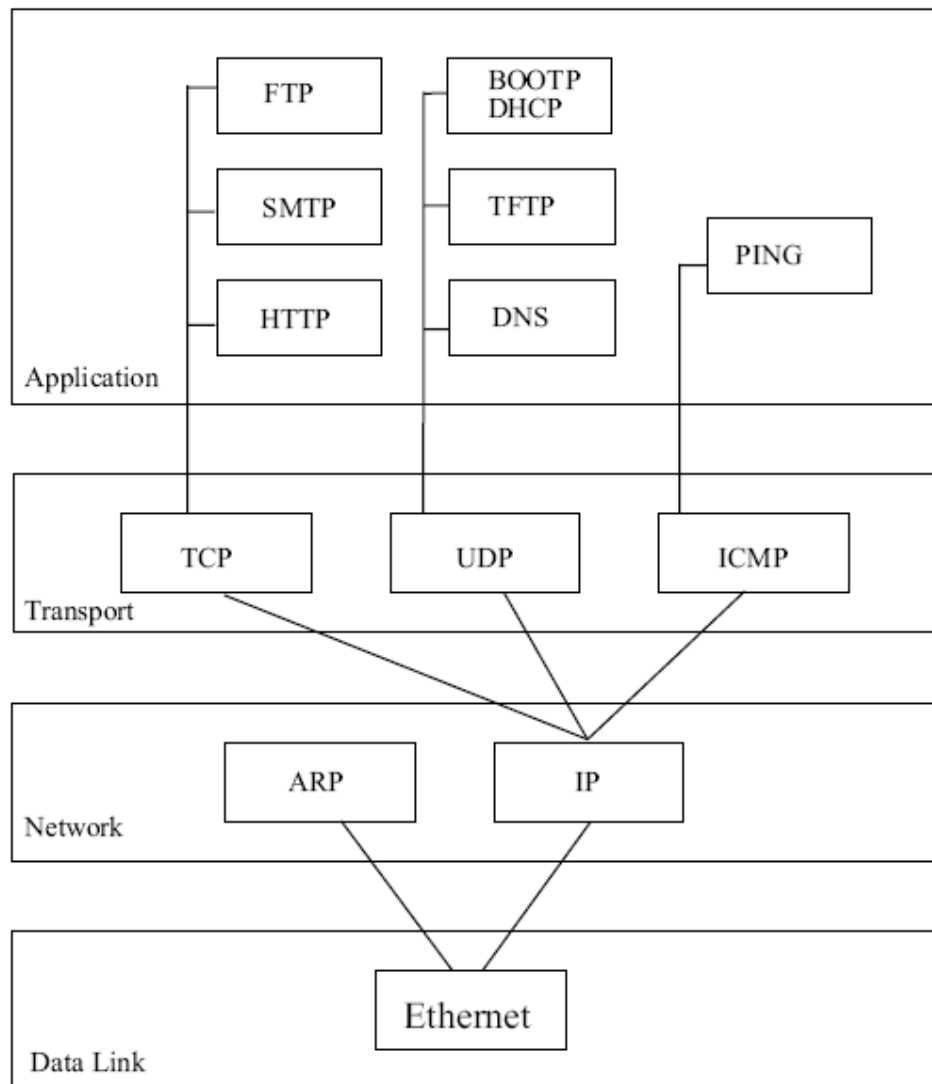
#### 3.1.1 TCP/IP protokol

Na tomto protokole je založená internetová komunikácia. Tok dát medzi dvoma počítačmi používajúcich TCP/IP je znázornený na obrázku 3.1 .



Obrázok 3-1 Dátový tok medzi dvoma počítačmi prostredníctvom TCP/IP

Pokiaľ chcú dva a viac počítačov založiť komunikáciu medzi sebou je potrebné, aby na každom z nich bol TCP/IP protokol implementovaný. Keďže používam komunikáciu cez internet využívam všetkých 5 vrstiev tohto implementačného protokolu. Aké služby a čo sa na ktorej vrstve robí je znázornené na obrázku 3.2.



**Obrázok 3-2 TCP/IP protokol [12]**

V programe je nadefinovaná vlastná ip adresa zariadenia pomocu makra `_PRIMARY_STATIC_IP` na adresu 192.168.1.1

IP adresa poskytuje komunikáciu zariadeniam na rôznych druhoch sieťach. Je zviazaná s maskou siete prípadne podsiete a ak sa nachádzame za smerovačom tak i s východnou bránou siete. Na základe mojej zvolenej ip adresy, ktorá patrí do tzv. C kategórii siete som volil masku siete 255.255.255.0 a ip východnej brány 192.168.1.8. Pre používanie makier a funkcií TCP/IP protokolu je treba vložiť do programu knižnicu s nadefinovanými funkciami pre ich spracovanie. Ide o `dcrtcp.lib`, v ktorej sú nadefinované hlavičky ďalších knižníc ako napríklad mnou používaná knižnica `tcp.lib`, v ktorej prebieha overovanie a znovu zasielanie paketov po sieti. V piatej aplikačnej vrstve využívam knižnicu `http.lib`, ktorá slúži na zaobstaranie komunikácie medzi stranou servera, ktorý je spustený na rabbit procesore a strany klienta, teda webového prehliadača.

Spustenie web servera a správna konfigurácia je základ úspešného peer to peer (klient klient) spojenia pre môj systém. Pri každom návrhu je treba dbať na dôraz čo od danej vizualizácie očakávame, teda ako bude prebiehať komunikácia, či budú dáta šifrované, prípadne nejaký obmedzený prístup pre užívateľov a administrátora, čo všetko sa bude ovládať a zobrazovať a v neposlednom rade ako bude stránka vyzeráť. Komunikácia medzi jednotlivými aplikáciami v mikroprocesore prebieha nasledovným spôsobom :

V aplikačnom bloku sa nachádza kód, ktorý som naprogramoval. O ostatné služby a obslužné rutiny sa postarajú vložené knižnice. V tomto bloku prebiehajú:

1. kompilačno inicializačné procesy, ktoré slúžia na vybratie príslušných knižnicových modulov – `dcrt.lib`, `http.lib`, `spi.lib`, inicializáciu konštánt a premenných potrebných v programe, štruktúr – tu patrí štruktúra `IO_data`, ktorá slúži pre vstupno/výstupné dáta systému (vstupné napätie `AIx`, výstupné napätie `AOx` a iné), externých súborov, ktoré vkladám pomocou makra `#ximport`. Makro `#ximport` hovorí prekladaču aby vložil súbor v úvodzovkách do flash pamäti a bol v programe prístupný pod danou konštantou, ktorá sa nachádza za úvodzovkami. Príklad : `#ximport "D:\temporal_school\6.semester\BBPA\Rabbit_prog\08\index.html"`  
`index_html` Pomocou tohto makra sú povkladané všetky súbory pre

chod web stránky. V tejto sekcii sú aj makrá pre prehľadnejšiu stránku kódu a tiež pre jeho jednoduchšie programovanie.

2. Počiatočná inicializácia za behu programu, ktorá sa vykoná iba raz vo funkcii *main()*. Tu inicializujem TCP/IP stack pomocou funkcie *sock\_init()*, http server pomocou funkcie *http\_init()*, *SPIinit()* na inicializovanie SPI komunikácie, *MCP3208Init()* pre inicializáciu A/D prevodníka. Ako rozšírenie mojej práce tu mám tiež zakomponované overovanie prihlásenia užívateľa, ktorý sa prihlasuje pod svojím menom a heslom. Sú volané funkcie *sspec\_addrule()* a *sauth\_setusermask()*. V inicializačnej časti pre premenné vo funkcii *main()* mám ponastavované počiatočné stavy premenných pre algoritmus PSD regulátora, premenné pre reprezentáciu stavových veličín procesu pre web rozhranie. Pre obsluhu relé a periférnych zariadení – prevodníkov bolo treba ponastavovať porty mikroprocesora. Konfigurácia zahŕňala nastavenie portov buď pre vstup alebo výstup a u sériových portov pre SPI komunikáciu kontrolné registre pre nastavenie clock a registrov pre dátovú zbernicu. Využívali sa funkcie : *WrPortI()*, *RdPortI()*, *BitWrPortI()*, *BitRdPortI()*.
3. Hlavná funkcia *main()*, v nej je cyklicky volaná funkcia *http\_handler()*, ktorá zabezpečuje proces komunikácie medzi serverom a klientom na sieti. Okrem nej sú tu tiež funkcie *plc()* a *psd()* pre riadenie spojitej a diskkrétnej regulácie.
4. Komunikácia klienta s http serverom pomocou *#web* premenných, ktoré sa pravidelne aktualizujú pri zmene hodnoty, či už na strane servera alebo klienta.

V HTTP bloku sa server stará o spracovávanie požiadavkov z okolitého sveta, teda od klientov z internetu. Požiadavky sa spracovávajú a vyhodnocujú a na základe stanovených pravidiel sa jednotlivým úkonom priradujú prislúchajúce akcie, prístupy do siete a manipulácie s dátami, povolenia. Výsledok spracovania sa posiela späť na prehliadač.

Zserver blok. Úlohou tohto bloku je kontrolovať prístup k ďalším blokom v diagrame. V mojom prípade sa okrem iného stará aj o mapovanie premenných do pamäti.

Riadky začínajúce `SSPEC_MIMETABLE_START` sa starajú o inicializáciu mime tabuľky. Príklad :

```
SSPEC_MIMETABLE_START
SSPEC_MIME_FUNC(".zhtml", "text/html", zhtml_handler),
...
SPEC_MIMETABLE_END
```

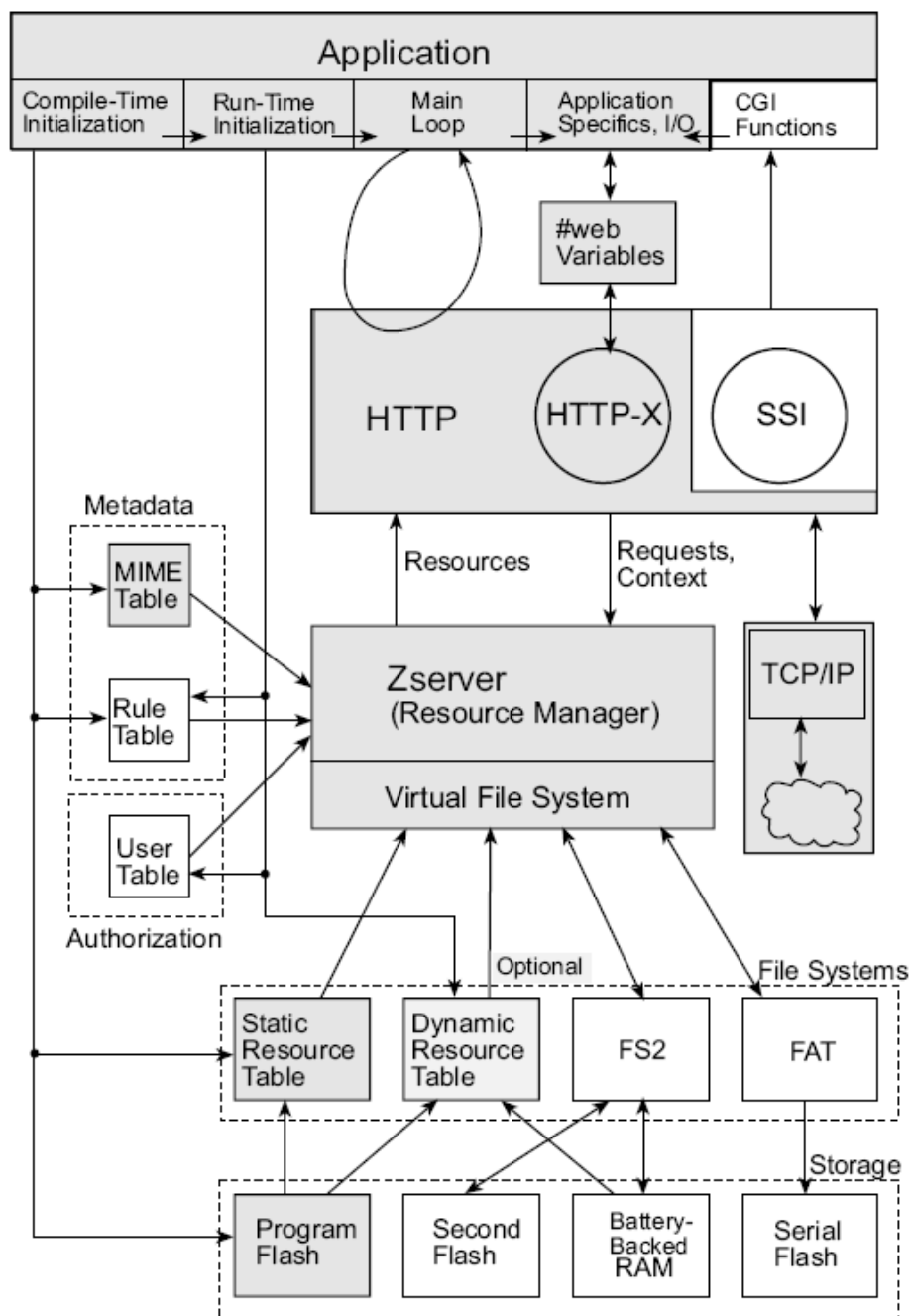
Kód vraví aby sa do MIME tabuľky namapovali súbory končiace s `zhtml` príponou a boli typu `text/zhtml`. `Zhtml_handler` je súčasť zservera používaná pre manipuláciu s dynamickými premennými na stránke. Pomocou nich som schopný obnovovať len meniace sa premenné namiesto načítavania celej stránky. Časť kódu používajúca dynamické premenné je oddelená a zapísaná špeciálnymi znakmi v časti html stránky. Štýl zápisu je podobným php skriptom. V tejto časti programu (mapping variables) vyhradím miesto v pamäti pre súbory potrebné pre zobrazenie stránky na prehliadači.

Blok kódu

```
SSPEC_RESOURCETABLE_START
SSPEC_RESOURCE_XMEMFILE("/", index_html),
SSPEC_RESOURCE_XMEMFILE("/index.html", index_html),
...
SSPEC_RESOURCETABLE_END
```

nastavuje tabuľku zo zdrojovými adresami. To znamená, že po zavolaní `index_html` sa zavolá z pamäte odpovedajúci zdrojový súbor.

Nie všetky možnosti Zservera som používal vo svojej aplikácii preto niektoré z nich nie sú spomenuté pri opise. Princíp činnosti aplikácie je znázornený na obrázku 3.3.



Obrázok 3-3 Minimálny obsah komponentov slúžiaci pre spustenie web aplikácie [13]

### 3.1.2 Web stránka

Pri rozhodovaní pre zobrazovanie a ovládanie premenných som sa rozhodoval medzi dvoma možnosťami realizácie. Prvá bola napísať stránku do pola char , ktoré by bolo priamo súčasťou zdrojového kódu programu. Pri posielaní pola by sa v prehliadači zobrazovala naprogramovaná stránka. Išlo by však len o statickú stránku s množstvom problémov, ktoré by sa prejavili neskôr. Druhá, už realizovaná je vytvorenie web stránky ako samostatných súborov. Firma Rabbit Semiconductor pripravila pre túto časť úlohy potrebné skripty a knižnice.

Obsah stránky tvoria :

- Statická stránka index.html, ktorá je úvodná a obsahuje odkazy na podstránky. Obsah stránky tvorí popis a možnosti ovládania spojitej a diskretnej regulácie a nastavovanie PSD regulátora.
- Dynamické stránky DiskReg.zhtml, setup.zhtml, Spojreg.zhtml Tieto súbory majú príponu .zhtml, ktorá je potrebná pre web server, aby vedel že stránka obsahuje meniaci sa obsah.
- V DiskReg.zhtml je naprogramovaná stránka pre diskretnú reguláciu. Je tu možnosť plc regulácie.
- Setup.zhtml je konfiguračná stránka pre PSD regulátor.
- SpojReg.zhtml je stránka ktorá slúži na zadanie a zobrazenie aktuálnej teploty, prípadne inej spojitej veličiny v procese. Pre správnu reguláciu je potrebná konfigurácia PSD regulátora v záložke Nastavenie (Setup.zhtml).
- Formátovací súbor style.css Tento štylizovací súbor obsahuje potrebné štýly pre krajší vzhľad stránky. Využívaný jazyk je css (Cascading Style Sheets).

Stránka je naprogramovaná v html jazyku, s využitím kaskádnych štýlov a skriptov zhtml. Jazyk html bol zvolený preto, lebo je súčasťou príkladov v rabbit zariadeniach. Navyše je jednoduchý a nenáročný . Pre jeho grafickú úpravu som ho zoštylizoval s css jazykom. Stránka sa tak stáva užívateľovi viac prívetivá. Vo všetkých dynamických súboroch sú obsiahnuté zhtml skripty. Navyše štylovanie

v css je zaberie menej pamäte ako načítavanie obrázkov do webovskej stránky. Takto sa zbytočne nezaťažuje činnosť mikroprocesora.

Príklad použitia skriptu v zhtml súbore diskkrétnej regulácie:

```
<INPUT TYPE="text" NAME="Request" SIZE=1 VALUE="
<?z print($Request) ?>">
<?z if (error($Request)){ ?>(ERROR!)<?z } ?>
```

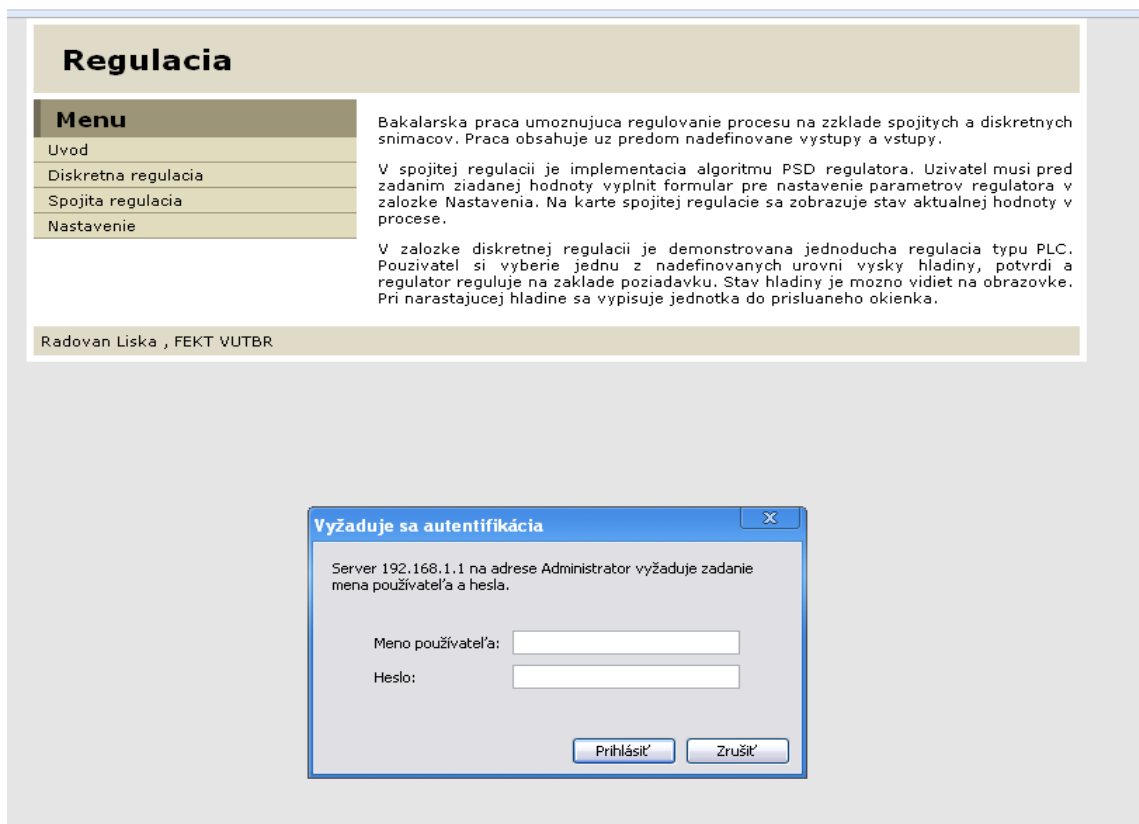
Táto časť kódu zobrazuje a vypisuje stav premennej Request. Je to premenná ktorá symbolizuje žiadanú hodnotu pre výšku hladiny v nádrži. Zároveň je táto premenná limitovaná podmienkou, ktorá sa kontroluje na strane servera. V prípade že dôjde k porušeniu podmienky, výpis ERROR sa vypíše na obrazovku vedľa premennej.

Veľkou výhodou takto riešenej stránky je, že jej obsah sa načíta iba raz a počas doby trvania celej regulácie sa aktualizujú len jednotlivé premenné.

Ako doplnok stránky slúži autentifikácia užívateľa pre nastavovanie parametrov na záložkách spojitaj a diskkrétnej regulácie.

### 3.1.3 Popis aplikácie – Uživatelský manuál

Po napísaní adresy 192.168.1.1 sa na obrazovke objaví úvodná stránka. Jej obsah je tvorený základným popisom a ovládaním zvolených regulácií. V ľavom okraji sa nachádzajú záložky, ktoré po kliknutí na ne zobrazia ich obsah. Pre zobrazenie ich obsahu je potrebné, aby sa užívateľ prihlásil do systému. V dialógovom okne je potrebné vyplniť prihlasovacie meno (login) a heslo (password). Dialógové okno je znázornené na obr. 3.4.



Obrázok 3-4 Úvod a prihlasovanie sa na stránku

V záložce spojitá regulácia je má uživatel' možnost' zadat' požadovanú teplotu v miestnosti. Aktuálna teplota sa nachádza v okienku pod požadovanou teplotou(Obrázok. 3.5).

**Regulacia**

| Nazov parametra  | Hodnota parametra | popis                               |
|------------------|-------------------|-------------------------------------|
| Ziadana teplota  | 30.000000         | Zadajte teplotu od 0 do 100 stupnov |
| Aktualna teplota | 25                | teplota v miestnosti                |

Submit Reset

Radovan Liska , FEKT VUTBR

**Obrázok 3-5 Spojitá regulácia**

V záložke Diskrétna regulácia je regulovanie stavu výšky hladiny v nádrži. Z výberového políčka si môže užívateľ vybrať predom definovanú výšku. Aktuálna výška sa zobrazuje v políčkach pod výberom.

Zobrazovanie prebieha spôsobom :

Ak nie je v nádrži žiadna voda, tak v žiadnom z políčok nie je 1. Jednotka znázorňuje dosiahnutú výšku. V prípade, že v nádrži nie je žiadna voda je spustený digitálny výstup na 8mom relé. Na toto relé je možné pripojiť záťaž s odberom do 2A alebo prepojiť so stykačom a spúšťať zariadenia s väčšou záťažou (napríklad čerpadlo). Ak snímač vody indikuje hladinu, privedie na digitálny vstup napätie. Na príslušnom porte je snímaný stav, a v prípade že je v stave log. 1 je zobrazený na stránke. V prípade, že dosiahne požadovaná výška hladiny žiadanú výšku, digitálny výstup sa vypína. (obrázok 3.6)

## Regulacia

### Menu

- Uvod
- Diskretna regulacia
- Spojita regulacia
- Nastavenie

| Nazov parametra  | Hodnota parametra                                             | popis                                                 |
|------------------|---------------------------------------------------------------|-------------------------------------------------------|
| Ziadana hodnota  | <input type="text" value="4"/> <input type="text" value="6"/> | Zadajte hodnotu pre vysku vody v nadrzi (4m alebo 6m) |
| MIN              | 4                                                             | minimalna hodnota hladiny                             |
| MAX              | 6                                                             | maximalna hodnota hladiny                             |
| Aktualna vyska 1 | 0                                                             | vyska hladiny dosiahla 2 metre                        |
| Aktualna vyska 2 | 0                                                             | vyska hladiny dosiahla 4 metre                        |
| Aktualna vyska 3 | 0                                                             | vyska hladiny dosiahla 6 metrov                       |

Radovan Liska , FEKT VUTBR

Obrázok 3-6 Diskrétna regulácia

Záložka Nastavenia je čisto nastavovacieho charakteru. Je potrebné tento list vyplniť pred tým, než sa spustí spojitá regulácia. Pri nenastavení parametrov PSD regulátora regulátor algoritmus bude hlásiť chybu nenastavenia a nebude regulovať. Stránka je zobrazená na obázku Obrázok 3.7.

## Regulacia

### Menu

- Uvod
- Diskretna regulacia
- Spojita regulacia
- Nastavenie

| Nazov parametra | Hodnota parametra                      | popis                                      |
|-----------------|----------------------------------------|--------------------------------------------|
| P zlozka        | <input type="text" value="20.000000"/> | Proporcionalna zlozka PSD reg              |
| I zlozka        | <input type="text" value="12.000000"/> | Sumacna zlozka PSD reg                     |
| D zlozka        | <input type="text" value="0.500000"/>  | Derivacna zlozka PSD reg                   |
| N               | <input type="text" value="3"/>         | N - filtracny koeficient derivacnej zlozky |

Radovan Liska , FEKT VUTBR

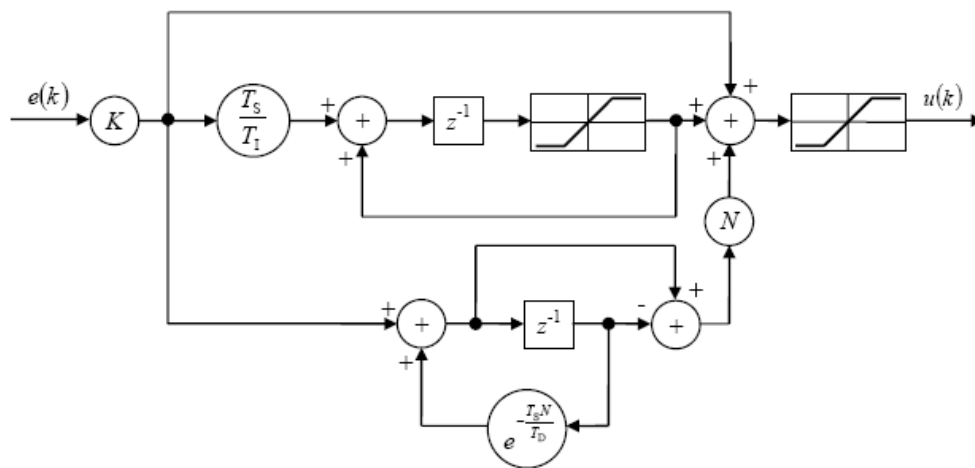
Obrázok 3-7 Nastavovanie Parametrov

### 3.2 ALGORITMUS PSD REGULÁTORA

PSD regulátor vychádza zo spojitého regulátora PID. Na to aby sme priebeh výstupného signálu z PSD regulátoru priblížili spojitému PID regulátoru je potrebné splniť množstvo podmienok počnúc od realizácie A/D prevodníka, ktorý je hlavným zdrojom nepresnosti, pretože v ňom dochádza ku kvantovaniu. Pre zrekonštruovanie signálu je potrebná frekvencia signálu, ktorá je daná Shannonovým teorémom. Rozbor z hľadiska hardware bol urobený v časti hardware. Na to, aby sa daný signál čo najviac podobal pôvodnému signálu je treba zabezpečiť, aby boli analógovým filtrom dostatočne odstránené všetky rušivé frekvencie a signály, ktoré presahujú frekvenciu vzorkovania. Dôležitý parameter v algoritme bude teda vzorkovacia frekvencia. Skracovaním periódy vzorkovania však narastá hodnota derivačnej zložky. Je preto výhodné aplikovať psd algoritmus s filtráciou derivačnej zložky. Pre odvodenie algoritmu vyjdem s rovnice :

$$Fr(z) = K \left( 1 + \frac{T_s}{T_i} \frac{z^{-1}}{1 - z^{-1}} + N \frac{1 - z^{-1}}{1 - e^{-\frac{NT_s}{T_d}} z^{-1}} \right) \quad (2.15)$$

Simulinkovské schéma riešenia algoritmu je na obrázku Obr 3.8



Obrázok 3-8 Simulinkovské schéma PSD regulátora s filtráciou derivačnej zložky [14]

Algoritmus PSD regulátora som zrealizoval do funkcie nasledovným spôsobom :

```
void psd(int ZiadanaHodnota)
{
    MCP3208Init;
    IO_data.AI_1 = MCP3208Read(0);
    E = ZiadanaHodnota-IO_data.AI_1;
    expo = pow(2.718, (-Ts*(N/Td)));
    ds = K*E+(ds*expo);
    Sum = Sum+K*(Ts/Ti)*E;
    Akcni_Zasah = K*E+Sum+N*(1-expo)*(K*E-ds+(ds*expo));
    if (Sum>MaxAZ)
        Sum = MaxAZ;
    if(Sum<MinAZ)
        Sum = MinAZ;
    if(Akcni_Zasah>MaxAZ)
        Akcni_Zasah = MaxAZ;
    if(Akcni_Zasah<MinAZ)
        Akcni_Zasah = MinAZ;
    MCP4922Write (1,1,1,Akcni_Zasah);
}
```

Algoritmus funguje tak, že sa snaží dorovnať odchylku E na nulu. Funkcia má vstupný parameter žiadaná hodnotu, teda hodnotu na ktorú bude regulátor regulovať. V prvej polovici je realizovaný čisto algoritmus pre PSD s filtráciou derivačnej zložky a v druhej polovici je ešte obmedzenie akčného zásahu a antiwind up (obmedzenie sumačnej zložky). Akčný zásah sa posiela do D/A prevodníka.

Nastavovanie jednotlivých parametrov si nastavuje užívateľ sám. Nastavované parametre regulátora sú :

Ts – perioda vzorkovania

Td – derivačná zložka

Ti – integračná zložka

P – proporciálna zložka

K – zosílenie

N – filtračný koeficient derivačnej zložky

### 3.3 KOMUNIKÁCIA MIKROPROCESORA A DAC

Pre komunikáciu mikroprocesora a D/A prevodníka je použitá SPI zbernica. Ide o synchrónnu sériovú dátovú linku pracujúcu vo full duplex režime(režim podporujúci prenos dát oboma smermi súčasne). Zariadenia komunikujú v master/slave režime. Pokiaľ je pripojených viac zariadení na zbernici, tak master vyberá pomocou CS(chip select), ktoré zariadenie bude v tej chvíli komunikovať.

Označenie vodičov na zariadení :

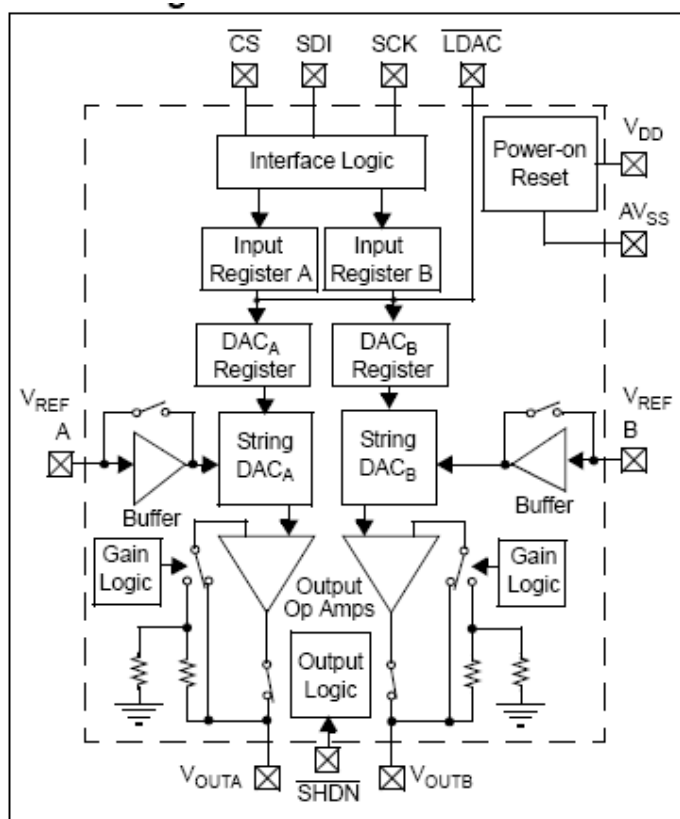
- CS - chip select
- SCK - serial clock , je poskytovaný zariadením typu master
- SDI - serial data in
- SDO - serial data out

#### 3.3.1 Popis činnosti a kódu programu pre D/A prevodník

Príkazy a dáta sú do prevodníka posielané na nábežnú hranu clock signálu cez SDI pin. Komunikácia je len jednosmerná, preto môžeme do prevodníka dáta iba zapisovať. Vytvoril som si funkciu *int MCP4922Write ( int ShutDown, int Channel, int Gain, float Volts )*, ktorá mi zabezpečuje posielanie dát z mikroprocesora do prevodníka. Funkcia má vstupné parametre :

- *shutdown* : inidikačný bit pre vypnutý alebo zapnutý prevodník. Pokiaľ je v logickej 1 tak je prevodník zapnutý.
- *Channel* : slúži pre výber výstupného kanálu prevodníka. Pre 0 je to kanál A a pre 1 kanál B.
- *Gain* : výstupné zosílenie signálu : pre 1 je to zosílenie 1(tento parameter je využitý v mojej aplikácii), pre 0 je to zosílenie 2.
- *Volts* : udáva hodnotu napätia, ktoré sa nastaví na prevodníku a výstupné napätie.

Blokové schéma převodníku je na obrázku 3.9. Zápis do registrov převodníka prebieha nasledovne: Zápis začne nastavením signálu CS do log 0. Převodník očekává 16 bitov, ktoré sa zapíšu do registrov. Ako prvý sa prijme najvýznamnejší bit dát (MSB). 16 bitov je rozdelených na 4 bity konfiguračné a 12 bitov dátových, v ktorých je zakódovaná hodnota pre výstupné napätie. Dáta sú vkladane nábežnou hranou. Na konci tohto zápisu, teda po poslaní posledného bitu(LSB), sa signál CS nastaví na logickú 1, aby sa dáta mohli presunúť do príslušného záchytného vstupného registru v převodníku. Aby sa dáta dostali na výstupných registrov je potrebné, aby sa signál LDAC nastavil na 0 na čas  $t_{LD}$

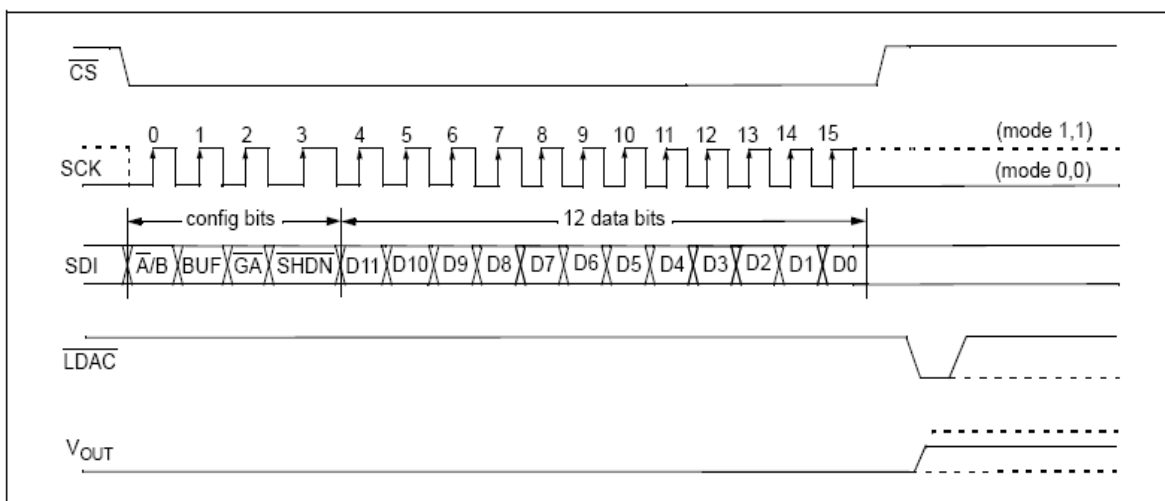


Obrázok 3-9 Blokové schéma D/A převodníka [15]

SPI umožňuje posielat' maximálne 8 bitov v jednom pakete. Preto bolo treba vytvoriť paket o veľkosti 16 bitov, ktorý sa posielal na 2 krát. Zakódované informácie obsahovali (čísľuje sa od MSB):

- 1.bit : bit pre ovládanie výstupu signálu, teda ktorý kanál sa bude ovládať, či kanál A, alebo kanál B.
- 2.bit : Vstupný kontrolný bit pre nastavovanie referencie napätia prevodníka, kde sa nastavuje mód so zásobníkom, alebo bez.
- 3.bit : bit pre nastavenie zosílenia, či už 1 alebo 2
- 4.bit : výstupný „vypínací“ bit , ktorý indikuje v akom režime je prevodník, teda či je aktívny, alebo vypnutý.
- 5ty až 16ty bit : dátové bity, v ktorých je zakódovaná informácia o hodnote napätia.

Zápis bitov je znázornený na obrázku 3.10.



Obrázok 3-10 Postupnosť zápisu dát [16]

### 3.3.2 Postupnosť krokov vo funkcii

- Nastavil som si podmienky pre obsluhovanie jednotlivých signálov.
- Prepočítal som si vstupný parameter napätia na číslo, ktoré sa bude kódovať do paketu podľa vzorca (16)
- Vytvoril som posielací paket.

- Pomocou funkcie *SPIWrite()* som zapísal tento paket do registrov v prevodníku. Táto funkcia je naprogramová firmou Rabbit Semiconductor a nachádza sa v knižnici SPI.LIB. Pred použitím funkcie *SPIWrite()* je potrebné inicializovať SPI rozhranie pomocou funkcie *SPIInit()*.

Prepočet napätia do prevodníka :

$$Data = \frac{2^n * V_{out}}{V_{ref} * G} \quad (2.16)$$

kde

- n     -- počet bitov prevodníka  
Vout -- požadované napätie  
Vref -- hodnota referenčného napätia  
G     -- zesílenie

Podrobný popis zariadenia sa dá nájsť v datasheete od prevodníka. Pre algoritmus A/D prevodníka MCP3208 som použil obdobný postup ako pre D/A prevodník.

## 4. ZÁVER

Práca sa zaoberala vytvorením riadiaceho systému, počnúc návrhom hardwaru až po jeho realizáciu spolu s implementovaním softwarovej možnosti kontroly a ovládania akčných členov v procese. Práca je rozdelená na dve časti. Prvá časť sa zaoberá teoretickým návrhom hardwaru a v druhej sa venujem jeho realizácii a doplnením o software. Z hľadiska kvalitnej regulácie, teda aby regulátor reguloval na nulovú ustálenú odchýlku, bolo treba dôkladne premyslieť návrh hardwarovej časti obvodu. Veľký dôraz bolo treba dať na periódu vzorkovania, od ktorej sa odvíjal aj návrh dolno frekvenčnej priepusti spolu s vhodným referenčným napätím pre A/D prevodník. Z počiatku som mal zapojený tento filter len v spätnej väzbe operačného zosilňovača, čo sa po hlbšom preštudovaní problematiky vlastností filtrov ukázalo ako nedostačujúce a obmedzujúce. Nakoniec som zvolil dolnú priepusť ešte pred vstupom signálu do operačného zosilňovača, čím som výrazne zlepšil dynamické vlastnosti tohto zapojenia. Okrem toho som dbal aj na to, aby som pri danom počte bitov prevodníka čo najviac z nich využil. Pri vysokých frekvenciách sa totižto výrazne znižuje rozlíšenie prevodníka, čo sa týka efektívneho využitia počtu bitov. Pri referenčnom napätí bol zas problém pri jeho voľbe z toho dôvodu, že bolo treba zosúladiť výstupné napätie senzorov po úprave filtrom a zosilňovačom so vstupnou citlivosťou prevodníka. Pri zvolení malého referenčného napätia by som riskoval „utopenie sa“ signálu v šume a pri veľkom by zase bol prevod prevodníka príliš nepresný. Nakoniec som nastavil hodnotu napätia na 2,5V.

V časti hardware som popísal princípy jednotlivých komponentov a ich riešenie v celkovom systéme. V tejto časti sa nachádzajú aj ťažiskové výpočty súčtiastok.

Modul mikrosystému Rabbit 3000 ponúka viacero možností na jeho využitie. Výrobca ponúka možnosti rozšíriť modul o zobrazovaciu časť (display), o tlačítka pre nastavovanie hodnôt a mnohé ďalšie. Ako zaujímavý príklad a realizáciu by som videl použitie joysticku na listovanie v menu a kvadratický enkodér na nastavovanie hodnoty pričom krok zmeny tejto hodnoty by bol závislý na rýchlosti otáčania enkodérom.

Softwarová časť rozoberá tvorbu zdrojového kódu pre riadenie regulátora, web servera, vnútornú komunikáciu medzi mikroprocesorom a periférnymi zariadeniami a tvorbu web stránky. Do mikroprocesora som programoval diskretný PSD regulátor. Naprogramovaný je algoritmus výpočtu a parametre regulátora sa volia na základe konkrétnych požiadavok na reguláciu dodatočne. Program spracováva číselné hodnoty z A/D prevodníka. Po dokončení výpočtu sa výsledok posielal do D/A prevodníka. Na jeho výstupe sú potom zodpovedajúce napäťové úrovne. Číslicová komunikácia je riešená prostredníctvom SPI zbernice.

Práca zahŕňa osadenú a funkčnú dosku. Otestované sú digitálne vstupy a výstupy používané pre diskretnú reguláciu typu PLC, a tieto je možné nastavovať cez priloženú webovú stránku na záložke Diskretná regulácia.

Pre analógové vstupy a výstupy sú vytvorené obslužné rutiny na ich ovládanie. Je ešte treba odladiť komunikáciu analógových vstupov a výstupov k čomu nedošlo z dôvodu zlyhania viacerých dodávateľov špecifických komponentov.

Zmeny hardwaru oproti pôvodnému návrhu boli v prepojení zemí do jednej. Prepojenie som sa snažil sústrediť do jedného bodu, a to pri hlavnom napájaní dosky. Pri optočlenoch bolo nutné zmeniť polaritu ochranných diód, ktoré v pôvodnom zapojení signál skratovali a optočlen preto nespínal. Úpravy neboli tak závažného charakteru, že by bolo nutné navrhnuť a vyrobiť novú dosku plošných spojov. Popísané úpravy sú opravené v priloženej schéme a sú realizované pomocou drôtových spojov na pôvodnej doske.

Ako rozšírenie tohto systému je možné realizovať pridanie slotu na SD/MMC karty, ktoré by slúžilo ako úložisko dát pre údaje z procesu. Softwarová časť by sa dala dokončiť pre využívanie všetkých vstupov a výstupov, ktorými zariadenie disponuje. Pre univerzálnejšie využitie regulátora by bolo vhodné dorobiť obslužné programy pre RS232, RS485 a pre komunikáciu s SD/MMC kartou.

Pri návrhu a konštrukcii jednotlivých blokov na mikrosystéme som získal cenné poznatky a prehľad v danej oblasti. Podrobne som sa oboznámil s komunikáciou prostredníctvom RS-232, RS-485 a získal prehľad v zostavovaní jednotlivých komponentov do ucelených blokov ako i výbere vhodných súčiastok a preštudoval problematiku prevodu analógového signálu na digitálny a naopak.

## 5. ZOZNAM POUŽITEJ LITERATÚRY

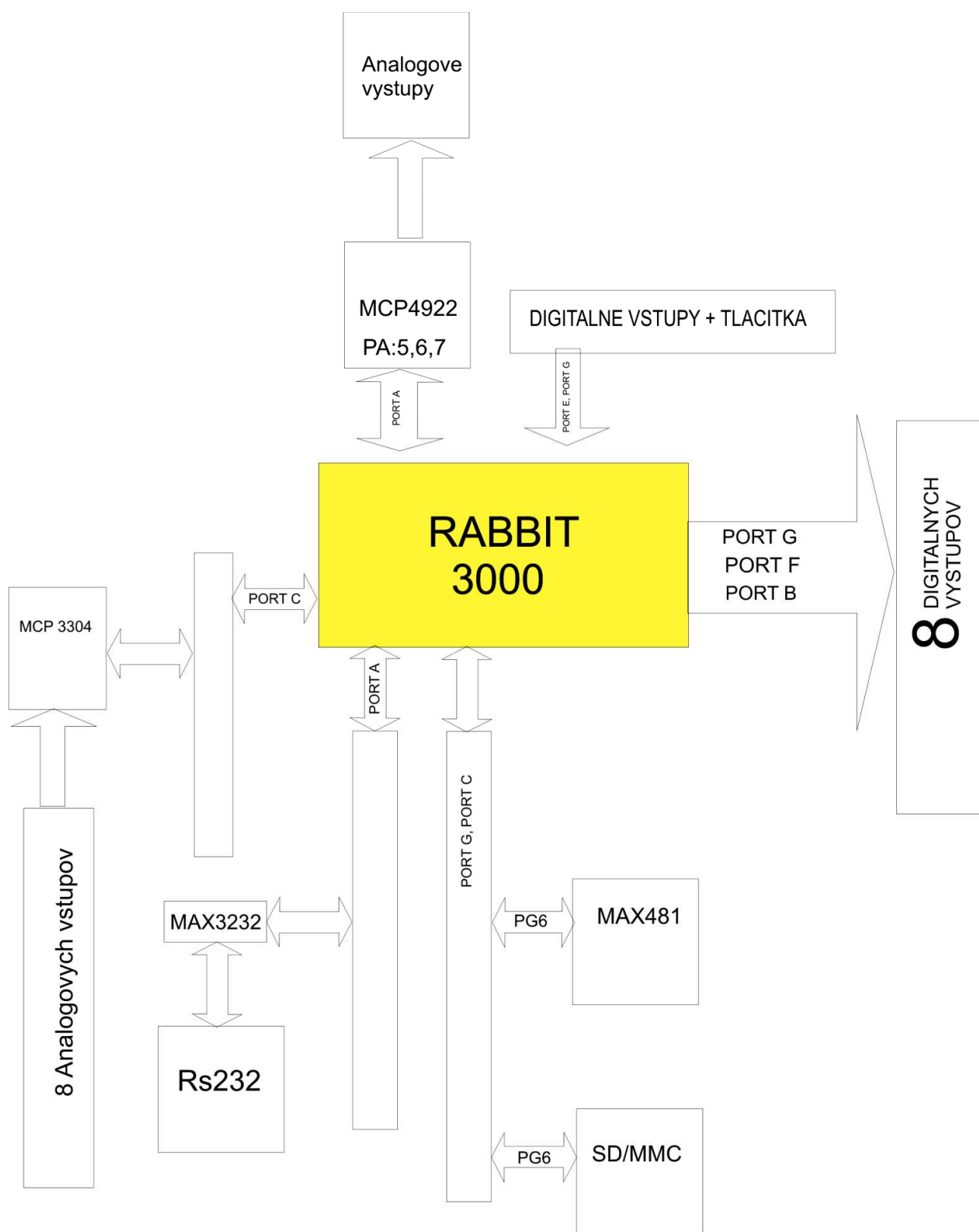
- [1] *Rabbit® 3000 Microprocessor User's Manual* [online]. 2007 [cit. 2008-12-23]. Dostupné z:  
<<http://www.rabbit.com/documentation/docs/manuals/RCM3000/UsersManual/index.htm>>.
- [2] *A/D and D/A Conversion/Sampling Circuits* [online]. 2008 [cit. 2008-11-10]. Dostupné z: <[www.maxim-ic.com](http://www.maxim-ic.com)>.
- [3] *HW server představuje - Sériová linka RS-232* [online]. 2005 [cit. 2008-10-03]. Dostupné z: <<http://hw.cz/rs-232>>.
- [4] *Data Converters* [online]. 2006 [cit. 2008-12-23]. Dostupné z:  
<[http://www.national.com/appinfo/adc/files/ABCs\\_of\\_ADCs.pdf](http://www.national.com/appinfo/adc/files/ABCs_of_ADCs.pdf)>
- [5] *Data Converters* [online]. 2005 [cit. 2008-10-11]. Dostupné z:  
<[http://focus.ti.com/analog/docs/analogtechdoc\\_hh.tsp?familyId=82&documentCategoryId=1&techDoc=1&rootFamilyId=82](http://focus.ti.com/analog/docs/analogtechdoc_hh.tsp?familyId=82&documentCategoryId=1&techDoc=1&rootFamilyId=82)>.
- [6] TIŠNOVSKÝ, P. *Sběrnice RS-422, RS-423 a RS-485*. [online]. , 18.12.2008 [cit. 23-11-08]. Dostupné z:  
<<http://www.root.cz/clanky/sbernice-rs-422-rs-423-a-rs-485/>>
- [7] *Dc/dc converters* [online]. 2007 [cit. 2008-11-18]. Dostupný z:  
<<http://www.tracopower.com/>>.
- [8,10,11] PATOČKA, Miroslav, VOREL, Pavel. *Řídicí elektronika-Aktivní obvody*. [s.l.] : [s.n.], 2004. 154 s.
- [8] *Relé s ekonomickým provozem* [online]. 2004 [cit. 2008-12-25]. Dostupný z: <<http://hw.cz/Teorie-a-praxe/Navrhy-vyvojare/ART1029-Rele-s-ekonomickym-provozem.html>>.
- [10] Microchip. MCP3208 [online]. 2005, 1-1 , 7.6.2005 [cit. 2009-28-05]. Dostupný z:  
<<http://www.chipsinfo.com/Datasheet/CEE93E4E202F60E400B2E963EEFFEA9F.htm>>

- [12] Rabbit Semiconductor. An Introduction to TCP/IP [online]. 19-19 , 24.7.2007 [cit. 2009-28-05]. Dostupné z :  
<http://www.rabbit.com/documentation/docs/manuals/TCPIP/Introduction/index.htm>
- [13] Rabbit Semiconductor. Dynamic C TCP/IP User's Manual Vol. 2 [online]. 2007, 19 , 24.7.2007 [cit. 2009-28-05]. Dostupné z www:  
<http://www.rabbit.com/documentation/docs/manuals/TCPIP/UsersManualV2/index.html>
- [14] PIVOŇKA, P.; VELEBA, V. *Číslicová řídicí technika Počítačová cvičení*. Sekcia 3.2, ALGORITMUS PSD REGULÁTORA, strana. 40.
- [15] Microchip. MCP4921/4922 [online]. 2005, 43 , 7.6.2005 [cit. 2009-28-05]. Dostupné z:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/21897B.pdf>
- [16] Microchip. MCP4921/4922 [online]. 2005, 44 , 7.6.2005 [cit. 2009-28-05]. Dostupné z :  
<http://ww1.microchip.com/downloads/en/DeviceDoc/21897B.pdf>.

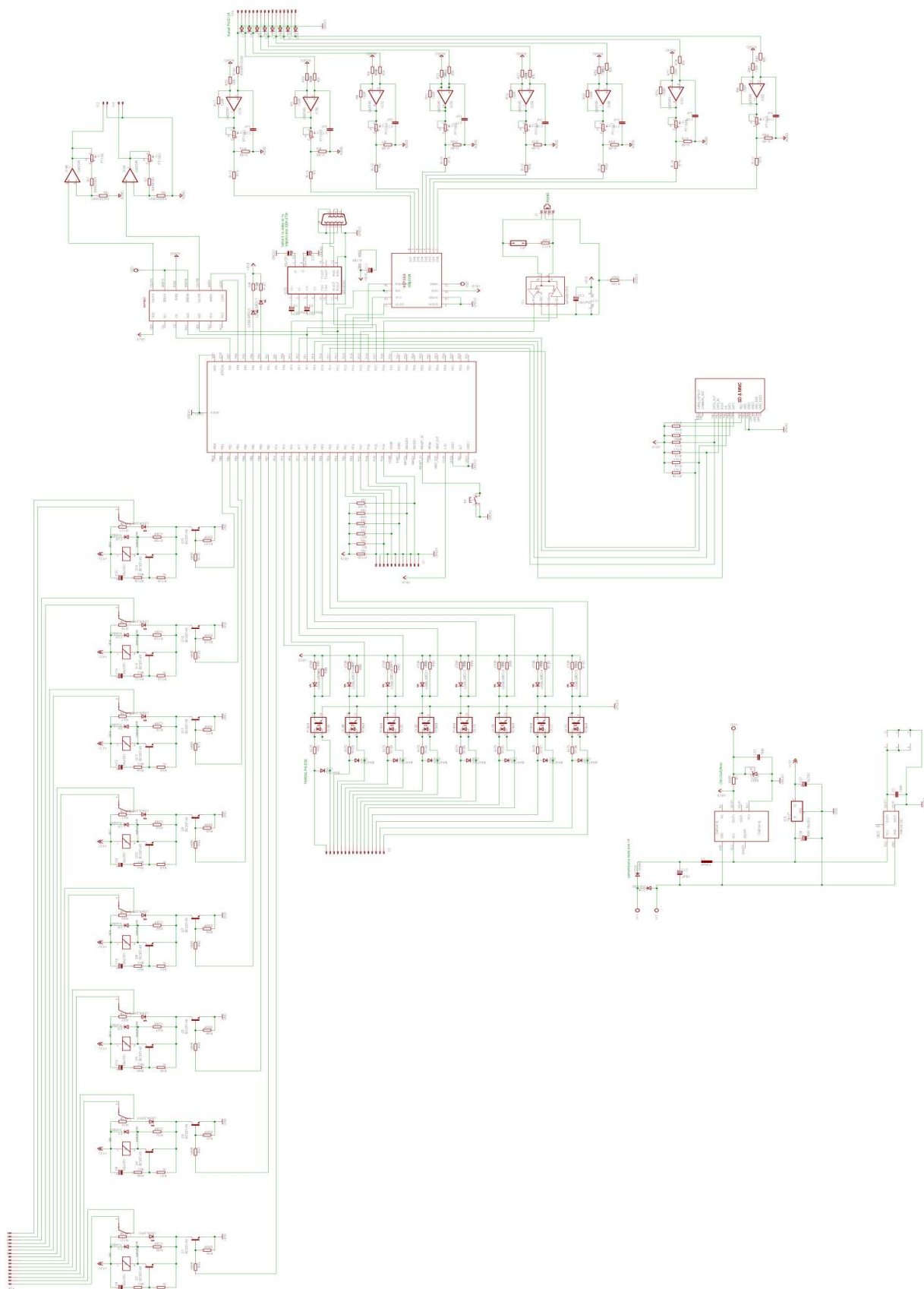
## **6. ZOZNAM PRÍLOH**

|           |                                 |
|-----------|---------------------------------|
| Príloha 1 | Blokové schéma mikrosystému     |
| Príloha 2 | Elektrická schéma mikrosystému  |
| Príloha 3 | Spodná strana DPS               |
| Príloha 4 | Vrchná strana DPS               |
| Príloha 5 | DPS celkový pohľad na spoje DPS |
| Príloha 6 | Rozmiestnenie súčiastok na DPS  |
| Príloha 7 | Pohľad na produkt z vonku       |
| Príloha 8 | Pohľad na produkt z vnútra      |
| Príloha 9 | Zoznam súčiastok                |

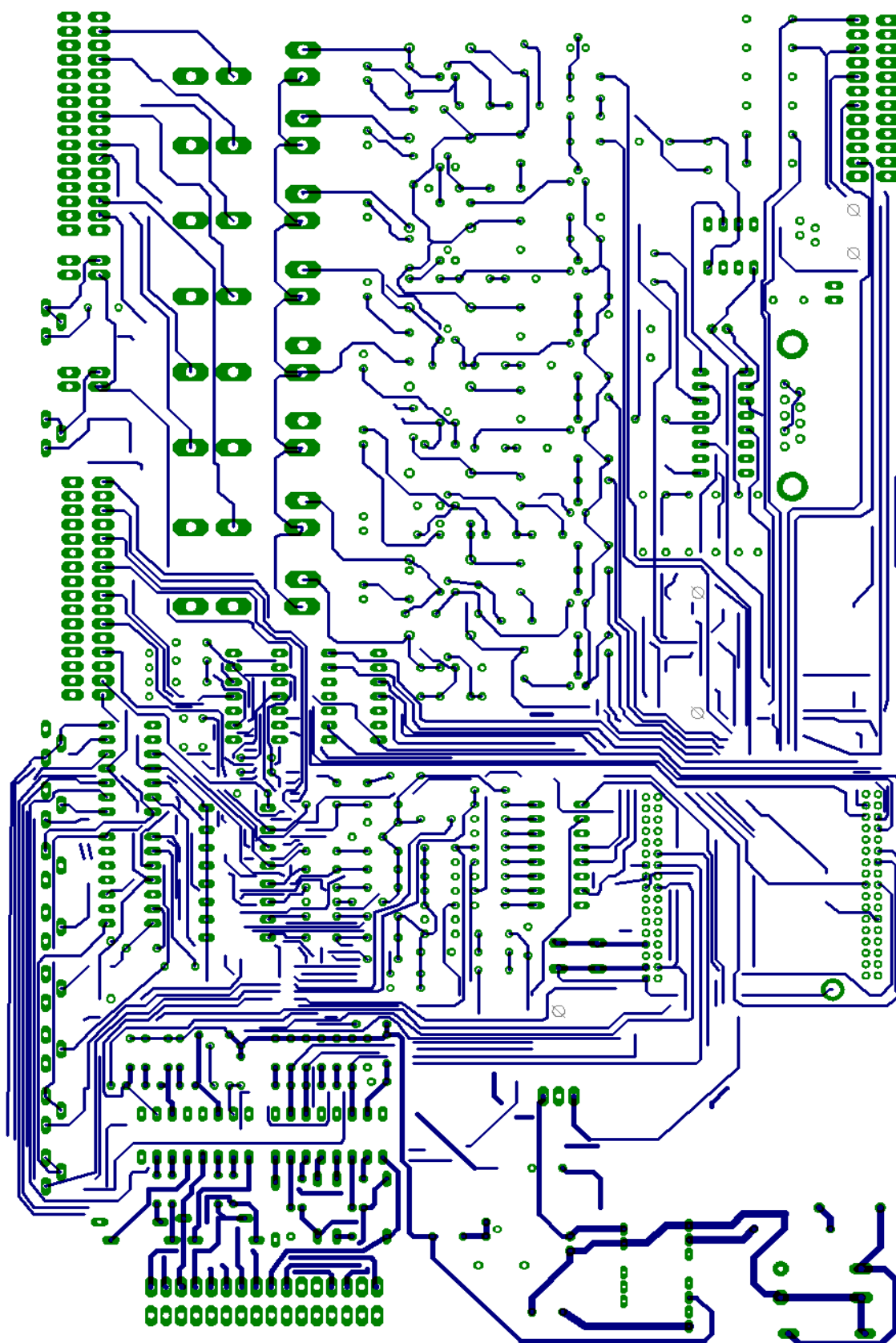
## Príloha 1 : Blokové schéma mikrosystému



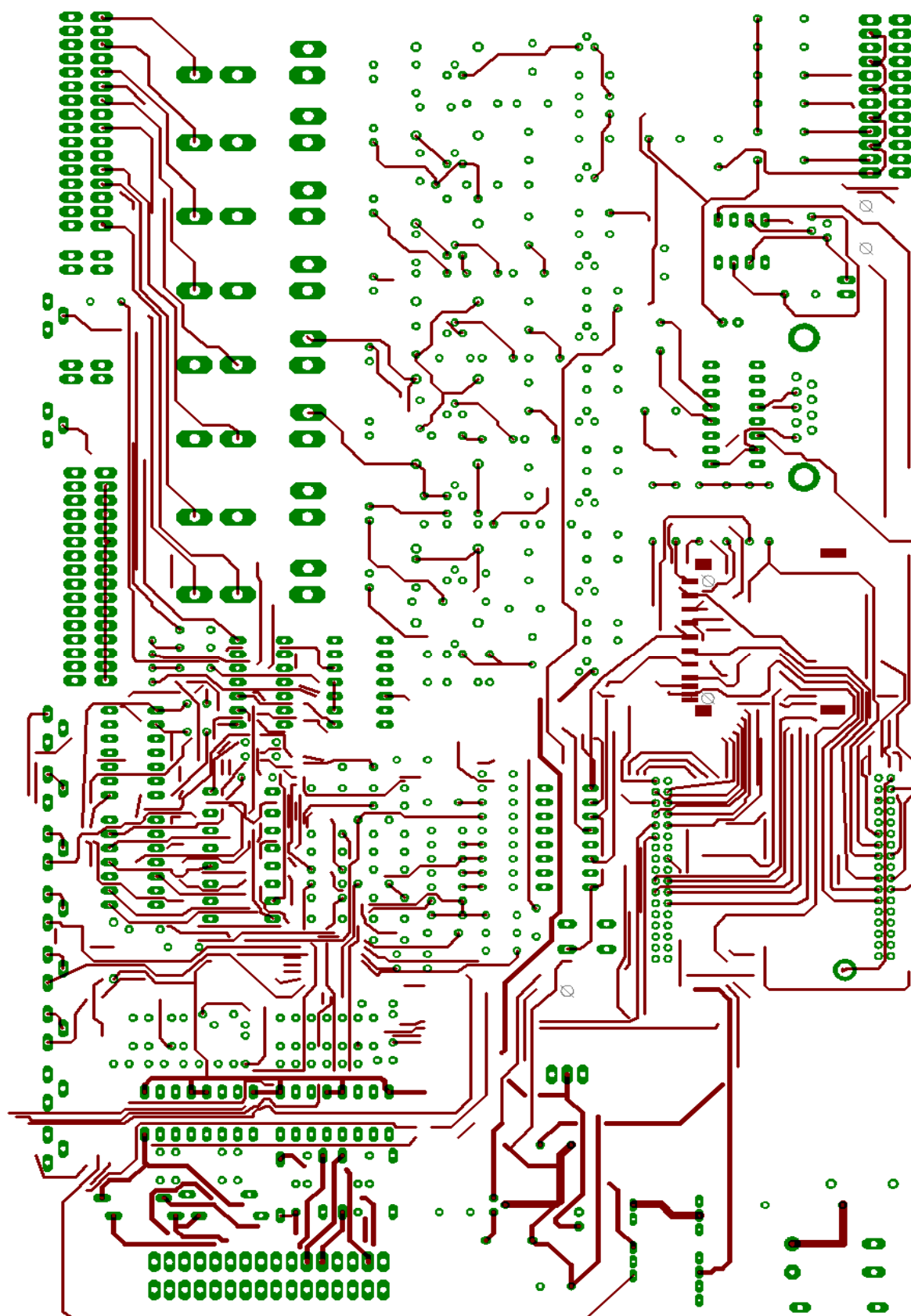
## Príloha 2 : Elektrická schéma mikrosystému



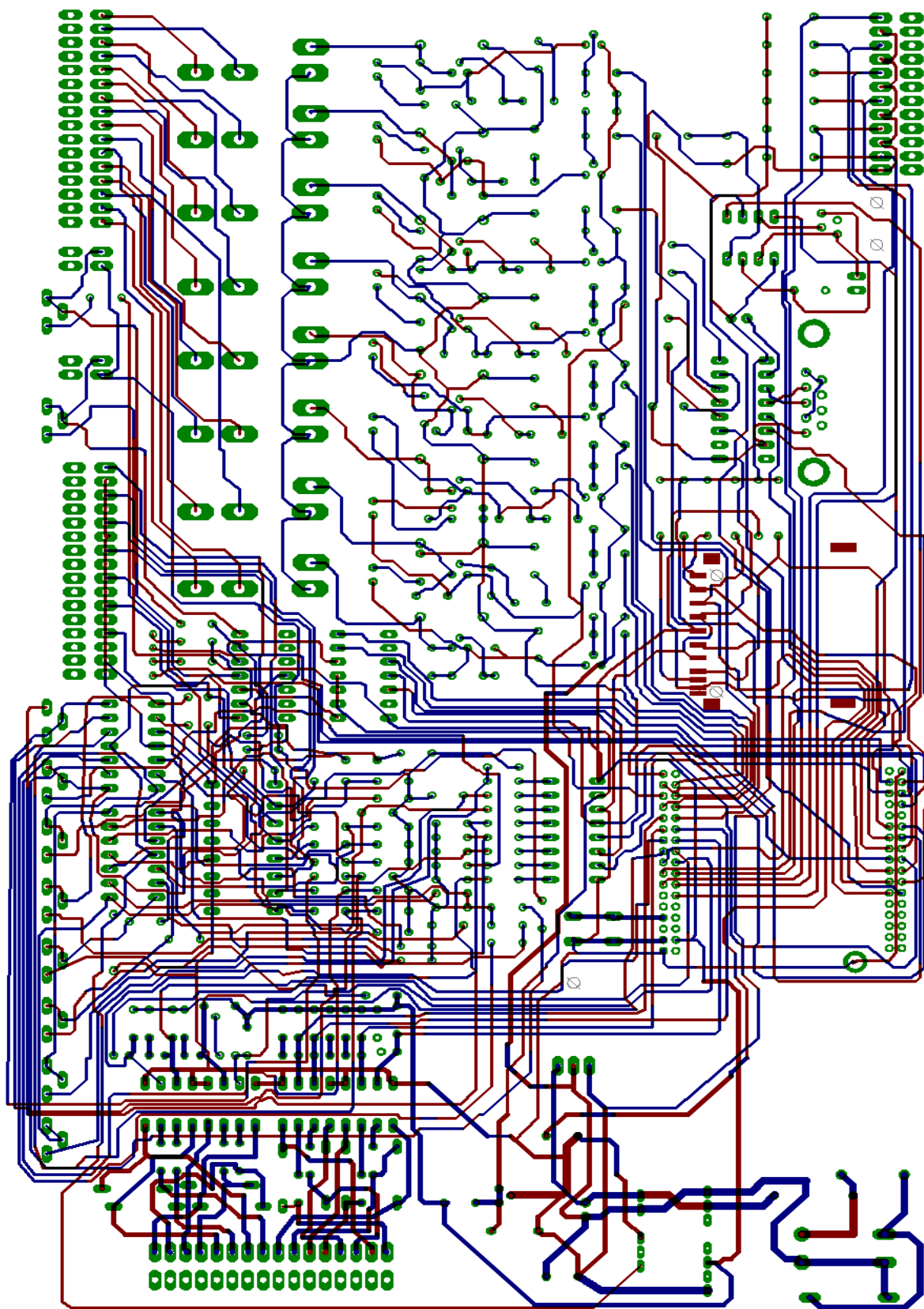
### Príloha 3 : Spodná strana DPS mikrosystému



#### Príloha 4 : Vrchná strana DPS mikrosystému



## Príloha 5 : celkový pohľad na spoje DPS



## Príloha 6 : Rozmiestnenie súčiastok na DPS

