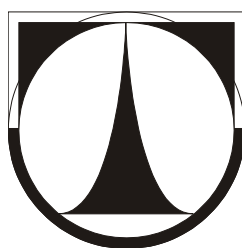


Technická univerzita v Liberci

Fakulta strojní



DIPLOMOVÁ PRÁCE

Elektronická univerzální řídicí jednotka

LIBEREC 2007/2008

Jiří Jindrák

ANOTACE

Technická univerzita v Liberci

Fakulta strojní

Katedra aplikované kybernetiky

Jméno a příjmení: Jiří JINDRÁK
Studijní program: M2301 Strojní inženýrství
Obor: 3902T021 Automatizované systémy řízení ve strojírenství
Zaměření: Automatizace inženýrských prací
Vedoucí DP: Ing. Michal Moučka, Ph.D.
Konzultant DP: Ing. Slavomír Němeček

Abstrakt:

Cílem této diplomové práce je navrhnutí, realizace a ověření funkce elektronické počítačem řízené jednotky. Tato jednotka je osazena několika analogovými, číslicovými vstupními a výstupními kanály. Analogové vstupní i výstupní kanály mají přepínatelný rozsah (0-5V, 0-10V). Jednotka je řízena jednočipovým mikropočítačem, který komunikuje s osobním počítačem pomocí rozhraní RS232. Rozhraní USB je pouze hardwarově podporováno. Součástí vybavení řídicí jednotky jsou i dva PSD regulátory.

Abstract:

The aim of this diploma work is to design, realize and check the functions of the electronic device controlled by the computer. This device includes some analog and digital entering and outgoing channels. Analog channels have switchable range (from 0V to 5V or 10V). The device is controlled by the microcontroller that communicates with personal computer by the RS232 interface. The USB interface is supported only by the device hardware. Two PSD regulators are also included in the device.

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu práce panu Ing. Michalovi Moučkovi, Ph.D. za odborné vedení, poskytnuté informace a za pomoc při zpracování diplomové práce.

Obsah

1 Úvod.....	1
2 Regulace.....	2
2.1 Číslicový PID regulátor.....	3
2.2 Filtrace měřených signálů.....	4
3 Jednočipové mikropočítače (mikrokontroléry).....	5
3.1 Architektury počítačů.....	6
3.2 Vlastnosti jednočipových mikropočítačů.....	7
3.2.1 Přerušovací systémy.....	8
3.3 Periferní obvody.....	9
3.3.1 Časovače procesoru PIC18F4553.....	10
3.4 Registry speciálních funkcí mikropočítačů PIC18Fxx.....	10
3.4.1 Mikrokontrolér PIC18F4553.....	12
3.5 Komunikace mikrokontroléru s okolím.....	16
3.5.1 Asynchronní komunikace.....	16
3.5.2 Komunikace RS232.....	17
3.5.3 Synchronní komunikace.....	19
4 Hardwarové řešení.....	20
4.1 Výběr vhodného mikropočítače.....	21
4.2 Digitální vstupy.....	22
4.3 Digitální výstupy.....	23
4.4 Analogové vstupy.....	24
4.5 Analogové výstupy.....	25
4.6 Zdroje referencí.....	27
4.7 Postup kalibrace.....	28
4.8 Výroba desky plošných spojů.....	30
5 Softwarová část řešení.....	31
5.1 Rozložení programových bloků.....	32
5.2 Nastavení portů.....	33
5.3 Nastavení komunikace RS232.....	33
5.3.1 Komunikační protokol.....	34
5.3.2 Přehled funkcí přístupných komunikačním protokolem.....	35
5.4 Časování regulátorů a TIMEOUT.....	38
5.5 Popis specifických proměnných.....	39
5.6 Knihovní funkce jednotky.....	41
5.6.1 Funkce související s regulátory.....	41
5.6.2 Funkce analogových kanálů.....	42
5.6.3 Ostatní funkce.....	43
6 Ověření správnosti a funkčnosti řešení.....	44
6.1 Elektrická specifikace.....	46
7 Závěr.....	47
Použitá literatura :	48
Seznam příloh:	49

1 Úvod

Cílem této diplomové práce je navrhnutí, realizace a ověření funkce elektronické počítačem řízené jednotky. Tato jednotka bude osazena několika analogovými, číslicovými vstupními a výstupními kanály. Analogové kanály budou mít přepínatelný rozsah (0-5V, 0-10V). Jednotka bude realizována na desce plošných spojů a řízena jednočipovým mikropočítačem, jehož softwarové vybavení bude umožňovat komunikaci s osobním počítačem. Součástí budou rovněž dva PSD regulátory, které budou moci samostatně (bez účasti PC) plnit svoji funkci.

Toto zařízení bude moci demonstrovat studentům možnosti jednočipových mikropočítačů při regulaci či řízení a monitorování technologických procesů pomocí komunikace mikrokontroléru s osobním počítačem. Každý, kdo s tímto zařízením bude pracovat, by měl mít základní znalosti z oblasti elektroniky a programování mikropočítačů.

2 Regulace

V současné době se stává řízení a automatická regulace součástí strojírenských výrobků, zařízení a technologických celků. V průmyslové praxi jsou zcela běžné regulace polohy, tlaku, teploty, otáček, výkonu apod. Automatická regulace ušetří obsluhu rutinních a opakovaných činností, zajišťuje kvalitu produkce a komfort pro obslužný personál. Kvalitní řízení a regulace zvyšuje konkurenceschopnost výrobků na domácích i zahraničních trzích. V posledních letech se v řízení a regulaci prosazují číslicové řídicí a regulační systémy. Tyto systémy jsou v anglické literatuře označovány jako "Process control system". Řídicí a regulační systémy jsou přirozeně využívány pro řízení a regulaci rozsáhlých technologických celků s mnoha regulačními a řídicími smyčkami. PID regulátory jsou u těchto systémů realizovány softwarově jako funkční bloky[2].

Teorie řízení poskytuje matematický aparát pro analýzu a syntézu systémů automatické regulace. V současné době z hlediska zpracování signálů a odpovídajícího hardware je možno teorii řízení rozdělit na

- 1) Teorii řízení spojitých systémů
- 2) Teorii řízení diskrétních systémů

Teorie řízení spojitých systémů zahrnuje analýzu a syntézu spojitých lineárních a nelineárních systémů. Základy analýzy spojitých systémů jsou studovány v základních předmětech automatické regulace[2].

Teorie řízení diskrétních systémů se zabývá analýzou a syntézou diskrétních systémů, které zpracovávají signály vzorkováním. Tyto systémy využívají zpravidla pro realizaci řídicích zákonů mikroprocesorů nebo mikropočítačů, které jsou opatřeny vstupními A/D a výstupními D/A převodníky[2].

2.1 Číslicový PID regulátor

Číslicový PID regulátor je alternativou ke spojitému regulátoru pro číslicovou techniku. Tyto číslicové regulátory se často označují jako PSD regulátory. Kde P - je proporcionální, S - je sumační a D - je diferenční složka regulátoru. Stejně jako u spojitých regulátorů lze analogicky vytvořit regulátory typu: PS, PD, S a P. Při realizaci řídicí jednotky byl použit následující přírůstkový (rekursivní) tvar PSD regulátoru (1)[1].

$$\Delta u(kT) = A \cdot e(kT) + B \cdot e[(k-1)T] + C \cdot e[(k-2)T], \quad (1)$$

kde

$$\begin{aligned} A &= k_R \left(1 + \frac{T}{T_I} + \frac{T_D}{T} \right) \\ B &= -k_R \left(1 + 2 \frac{T_D}{T} \right) \\ C &= k_R \left(\frac{T_D}{T} \right). \end{aligned} \quad (2)$$

Seřiditelnými parametry jsou: k_R, T_I, T_D za které je možno dosadit přímo hodnoty získané podle zvoleného kritéria kterýmkoliv optimalizačním postupem pro spojitý regulátor PID. Diskrétní regulátor PSD (1) generuje akční zásahy $\Delta u(kT)$ na základě změřené aktuální regulační odchylky $e(kT)$, její pamatované minulé hodnoty $e[(k-1)T]$ a předminulé hodnoty $e[(k-2)T]$. Kde T - je perioda vzorkování a k - je krok.

2.2 Filtrace měřených signálů

Šumy nepříznivě ovlivňují měřené signály. Vzorkování, u číslicového zpracování signálů, způsobuje převádění šumů vysoké frekvence na šumy nízké frekvence. Tyto šumy již mohou nepříznivě ovlivnit chování číslicového regulačního obvodu a proto bývají tyto obvody vybaveny i algoritmem filtrace zašumělých signálů. Lze použít spojitý filtr nebo diskretní.

Uvažujeme lineární spojitý filtr 1.řádu [1]:

$$G_f(p) = \frac{(E_f(s))}{(E(s))} = \frac{1}{(T_f s + 1)}, \quad (3)$$

kde $E_f(s)$ je L-obraz filtrované regulační odchylky $e_f(t)$, tj. výstupní veličiny filtru. Přenosu (3) odpovídá diferenciální rovnice

$$T_f \frac{(d e_f(t))}{dt} + e_f(t) = e(t), \quad (4)$$

která po diskretizaci (derivaci zastoupíme zpětnou diferencí) bude mít tvar

$$T_f \frac{(e_f(kT) - e_f[(k-1)T])}{T} + e_f(kT) = e(kT). \quad (5)$$

Úpravou získáme rekurentní diferenční tvar rovnice lineárního diskretního filtru 1.řádu

$$e_f(k) = (1 - F) e_f[(k-1)T] + F \cdot e(kT), \quad (6)$$

kde

$$F = \frac{1}{(1 + \frac{T_f}{T})} \quad (7)$$

je koeficient filtrace a musí vyhovovat nerovnosti $0 < F \leq 1$ [1]. Algoritmus filtrace (6) byl implementován v řídicí jednotce.

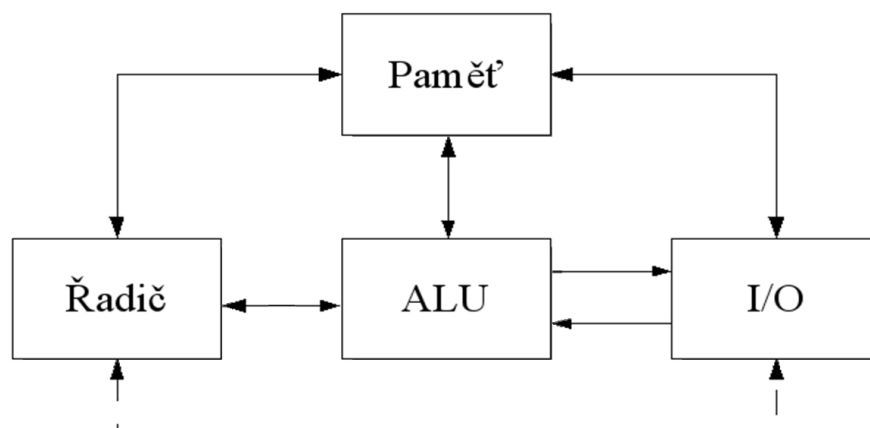
3 Jednočipové mikropočítače (mikrokontroléry)

Jednočipový mikropočítač je určitá forma centrální jednotky, umístěná v jediném pouzdře integrovaného obvodu. Obsahuje mikroprocesor s přiřazenými podpůrnými obvody a pamětmi. Je základem většiny řídicích a měřicích systémů se schopností pracovat podle předem daného programu. Jeho využití je prakticky neomezené. Pracuje v běžných elektronických přístrojích používaných v domácnosti jako je automatická pračka, myčka apod. až po nejnáročnější aplikace v leteckém či automobilovém průmyslu. Jednotlivé konfigurace mikrokontrolérů se liší i drobnými změnami v kapacitě pamětí (EEPROM, FLASH, RAM), počtu vstupů/výstupů, pouzdry, výpočetním výkonem, ale i vlastní spotřebou, která je velmi důležitá u bateriových systémů. Volbou vhodného „jednočipu“ lze v aplikaci ušetřit mnoho externích součástek a obvodů a tím navrhnout co nejjednodušší sestavu řídicího hardware. Takováto jednotka bude snadněji vyrobitelná, spolehlivější a samozřejmě i levnější. Z tohoto důvodu je na trhu mnoho modifikací těchto mikropočítačů od mnoha výrobců jako je Atmel, Microchip, Motorola, Texas Instruments aj. Je také snaha o integrování různých specializovaných podpůrných obvodů pro řízení periférií, které pak minimálně vytěží samotný procesor. Mezi takovéto obvody patří například různé komunikační prostředky s okolím, jako je (RS232, I²C, USB, CAN...), které lze většinou do mikropočítače naprogramovat, ale je nutné v takovém to případě počítat s velkou vyčerpáním procesoru touto komunikací. Dále můžeme vybírat podle počtu čítačů/časovačů, přítomností A/D převodníků a jejich rozlišením, komparátorů, vstupů s možností generování přerušování apod.

3.1 Architektury počítačů

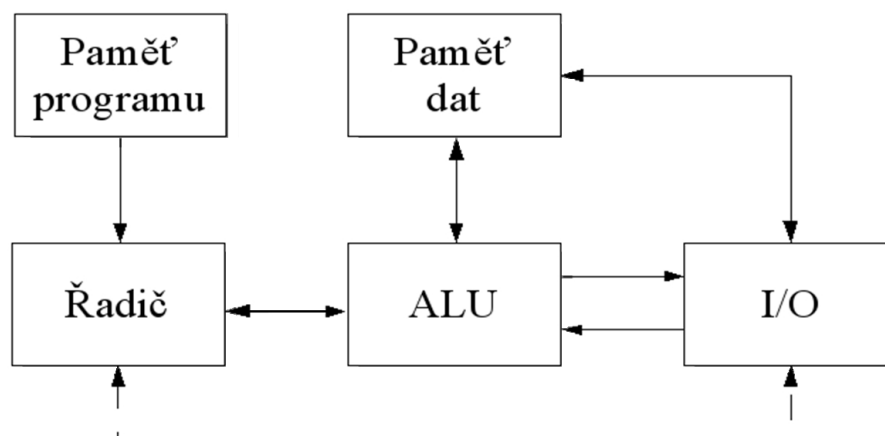
Existují dvě základní struktury číslicových počítačů:

1. Von Neumanova architektura. Aplikační data a program jsou uloženy ve společné paměti. Rozdíl mezi programem a daty je dán pouze jejich interpretací. Schéma Von Neumanovy architektury je na obr.1.



Obr.1: Von Neumanova koncepce

2. Harvardská architektura. Blokové schéma je na obr.2. Z hlediska paměti se jedná o dva oddělené prostory. Jeden prostor je pro program, druhý pro data. Tato architektura má omezený rozsah použitelnosti a je typická pro jednočipové mikropočítače či jednoúčelové obvody jako jsou například v jednoduchých kalkulačkách.



Obr.2: Harvardská koncepce

3.2 Vlastnosti jednočipových mikropočítačů

Jednočipové mikropočítače, označované také jako mikrokontroléry, obsahují všechny důležité obvody v jednom pouzdru. Vytvářejí tak samostatně fungující centrální jednotku. Důležité části mikropočítače jsou:

- **ŘADIČ** řídí činnosti spojené s interpretací instrukcí. Vykonává posloupnost elementárních operací v závislosti na vykonávané instrukci. V architektuře RISC (Reduced Instruction Set Computer) je řešen jako sekvenční obvod s neměnnou logikou a instrukce vykonává přímo. Jako programový řadič pracuje v architektuře CISC (Complex Instruction Set Computer). Tento řadič používá mikrokód pro komplexní vykonání instrukce, vykonání trvá delší čas než u architektury RISC.
- **(ALU)** Aritmetickologická jednotka provádí aritmetické a logické operace při kterých používá vlastní pomocné a příznakové registry.
- **REGISTR** je velmi rychlá paměť. Celková kapacita registrů je vzhledem k operační paměti malá. Operace s údaji v registrech však probíhají velmi rychle. Slouží k uchování operandů, nepřímému adresování do paměti atd.
- **SBĚRNICE** realizují propojení jednotlivých modulů počítače. Má několik vstupů a výstupů. Rozděluje se na adresové, datové a řídicí.
- **PAMĚŤ PROGRAMU** dnes většinou paměť typu Flash. U moderních mikropočítačů bývá tato paměť zpřístupněna pro zápis i z aplikačního programu. Má schopnost uchovat obsah déle jak 40let a lze ji asi 100 000krát smazat nebo přepsat.
- **PAMĚŤ** typu Read/Write občas doplněná o paměť typu EEPROM která uchovává obsah i po odpojení napájení. Paměť EEPROM lze až milionkrát přepsat.
- **PERIFERNÍ OBVODY** pro vstup a výstup

Dále mikropočítače obsahují různé technické prostředky jako je: Analogový komparátor, A/D a D/A převodník, obvody pro pulzní šířkovou modulaci (PWM), obvody hlídající činnost mikrokontroléru tzv. Watchdog či správné funkce hodinového kmitočtu, aj. Na obr.4 je blokové schéma mikropočítače firmy Microchip PIC18F4553, kde jsou vidět všechny obsažené subsystémy.

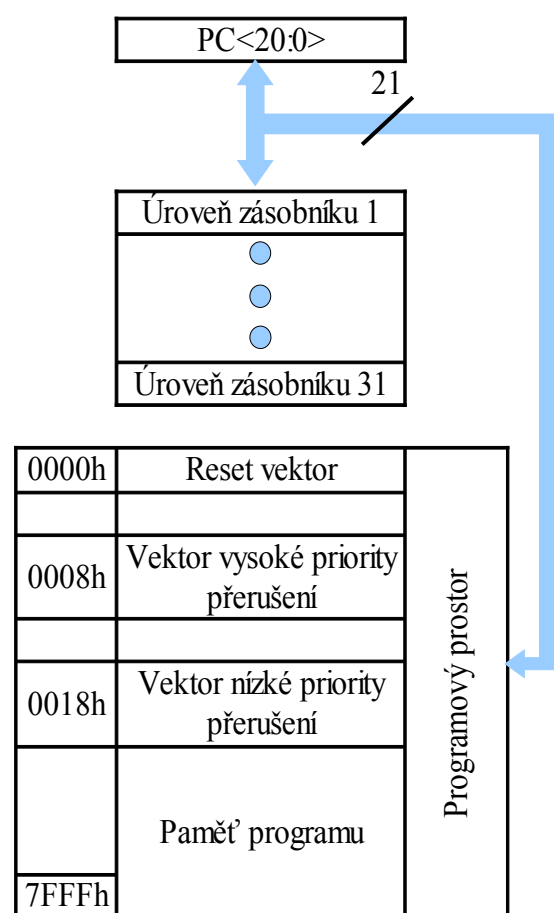
3.2.1 Přerušovací systémy

Přerušovací signál je signál vysílaný hardwarovým zařízením k procesoru. Tímto signálem se snaží zařízení zabrat procesor pro sebe. Procesor musí přerušit svou činnost a zpracovat požadavek zařízení. Příkladem může být přetečení čítače/časovače (změna všech logických nul na jedničky a naopak), detekci změny stavu na pinu procesoru (vzestupná hrana, sestupná hrana či logická úroveň) apod.

Moderní procesory mají vektorový systém přerušování. Každé přerušování má svůj vektor, který ukazuje na adresu v paměti, kde je uložen obslužný program přerušování. Před skokem

na vektor přerušování uloží procesor svůj momentální stav do speciálního registru – zásobníku. Po skončení podprogramu přerušování se tak procesor může vrátit k naposled vykonávané činnosti. Procesor musí zároveň obsahovat mechanismus, kterým se přerušování dočasně zakáže. Tyto funkce se nastavují v SFR (registry speciálních funkcí).

Většina mikropočítačů obsahuje prioritní přerušovací systém, který umožňuje volitelně nastavit některým zdrojům přerušování vyšší prioritu. Na obr.3 je mapa paměti s vyobrazením vektorů přerušování jednotlivých priorit u procesoru PIC18F4553. Pokud nastane událost s vyšší prioritou přerušování, dojde k přerušování i právě prováděné obsluhy přerušování s nižší prioritou.



Obr.3: Mapa paměti PIC18F4553

3.3 Periferní obvody

Využití mikrokontrolérů je stejně tak různorodé jako jejich vybavení periferními obvody. Je to z důvodu přizpůsobení se výbavou mikrokontroléru konkrétní aplikaci a tím snížit její cenu, zvýšit spolehlivost zařízení a zjednodušit ho. Jednotlivé varianty, určené pro jistou konkrétní aplikaci, se liší počtem a druhem komunikačních portů, podporou pulzní šířkové modulace, počtem A/D případně D/A převodníků, čítačů/časovačů a podobně.

Základní nepostradatelnou periferií jsou **Paralelní I/O** (Input/Output) porty sloužící pro vstupy a výstupy mikrokontroléru. Funkce těchto portů se obvykle nastavuje speciálními registry. S těmito porty lze obvykle pracovat najednou, nebo po jednotlivých bitech (pinech). Na jeden vývod jednočipu bývá vyvedeno hned několik funkcí. To je z důvodu nedostatku počtu pinů. V daném programu se pak musí pro daný časový okamžik použít pouze jedna z nich. Vícenásobné vyvedení funkcí na pin je dobře patrné na obr.4.

Mezi další obvyklé obvody patří **čítače/časovače**, jejich počet v mikrokontroléru je individuální. U 8bitových počítaců bývají 8-mi nebo 16-ti bitové. Jejich funkce (čítače nebo časovače) se nastavují opět speciálními SFR registry. Základní funkce časovače vychází z přičítání (nebo odečítání) impulzů hodinového signálu do pracovního registru časovače. Při jeho přetečení (překlopení všech binárních jedniček do nul nebo obráceně) se nastaví příznak přetečení. Tento příznak se buď vyhodnocuje uživatelským softwarem, nebo je navázán na přerušovací systém. Režim čítače pracuje obdobně. Zdrojem jeho čítání jsou vnější události, které bývá možno více specifikovat v SFR registru jako např. náběžná nebo sestupná hrana.

Sériové rozhraní slouží pro komunikaci mikrokontroléru s okolím. Jejich počet a druh je různý podle typu mikropočítače a to od 0 až po několik. Pokud takovou periferii mikrokontrolér má, většinou podporuje více komunikačních protokolů. Nejčastější bývá standardní asynchronní komunikace (RS232), nebo sériová synchronní komunikace. Podporovány bývají také protokoly jako CAN, I²C, SPI, USB aj. I²C a SPI se nejčastěji používají pro komunikaci na kratší vzdálenost, většinou v rámci jedné desky plošných spojů, mezi pomocnými integrovanými obvody jako jsou A/D převodníky, zdroje reálného času apod. CAN komunikace se například hodně využívá v automobilovém průmyslu, USB pak pro komunikaci s PC.

3.3.1 Časovače procesoru PIC18F4553

Princip časovače spočívá v čítači, jehož plnění je zajištěno kmitočtem odvozeným od frekvence oscilátoru či z externího zdroje. Jeho přetečení (překlopení všech binárních jedniček na nuly) pak vygeneruje přerušení.

Modul 0 lze nastavit jako čítač/časovač 16-ti nebo 8-mi bitový. Zdroj kmitočtu externí (s volbou hrany) nebo interní. Dále obsahuje nastavitelnou 8-mi bitovou děličku kmitočtu. U interního zdroje je na vstup děličky přiváděn kmitočet $F_{osc}/4$.

Modul 1 lze nastavit jako 16-ti bitový časovač/čítač s volbou zdroje interní/externí s podporou LP oscilátoru například pro zdroj reálného času. Dále je použit pro PWM funkce.

Modul 2 je 8-mi bitový čítač/časovač s programovatelnou 2bitovou a 4bitovou děličkou. Umožňuje čítání period. Dále využit pro funkce PWM (CCP) či pro sériovou synchronní komunikaci.

Modul 3 je 16-ti bitový čítač/časovač s nastavitelným zdrojem signálu. Dále využit pro CCP funkce.

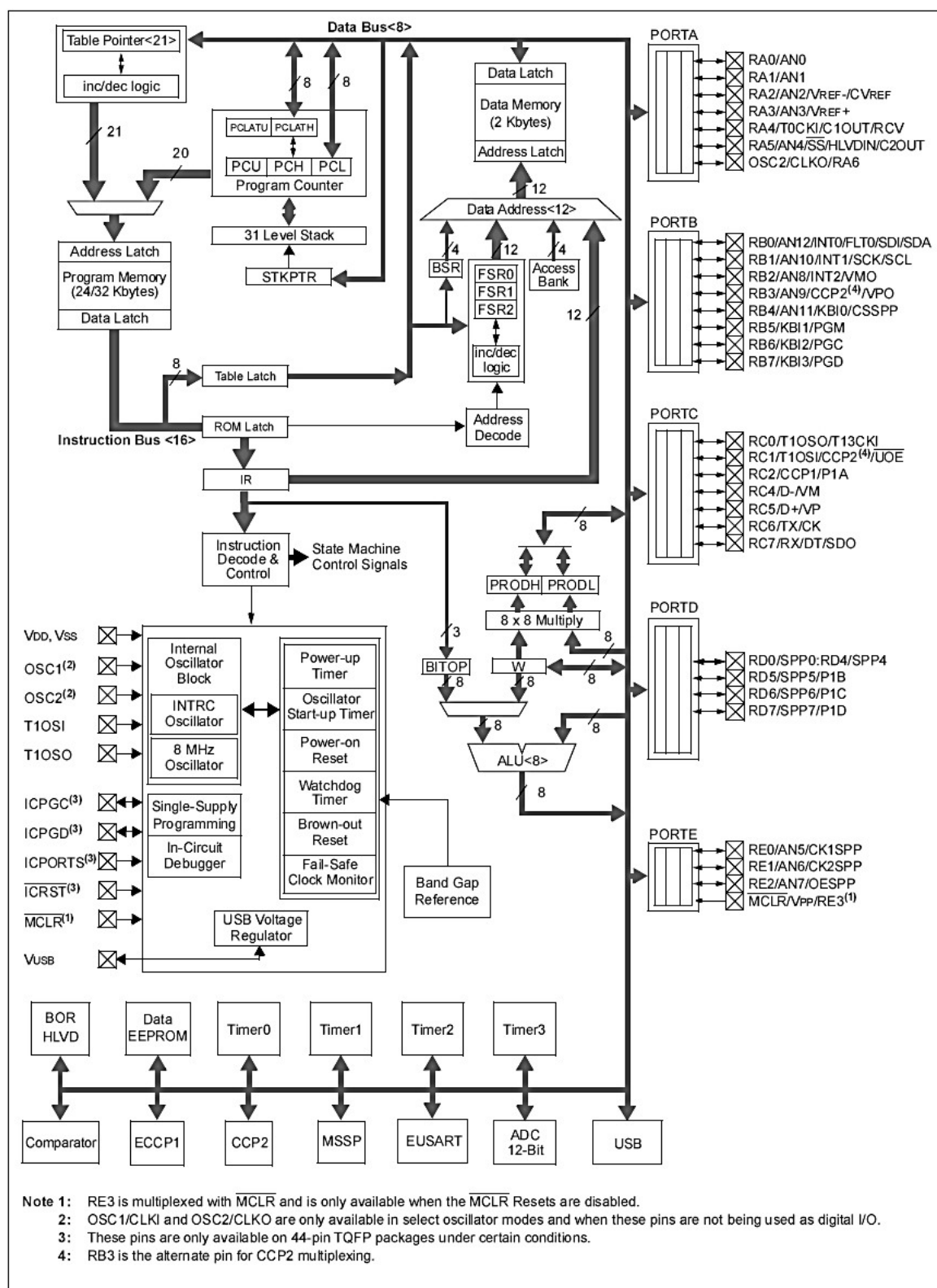
3.4 Registry speciálních funkcí mikropočítačů PIC18Fxx

Podobné registry speciálních funkcí mají i jednočipové mikropočítače jiných firem. Tyto registry SFR (Special Function Registers) bývají mapovány do vyhrazeného prostoru paměti dat. Po startu mikropočítače se důležité SFR registry nastaví do předem definovaného stavu, který je popsán v katalogovém listu, ostatní buď na nulu nebo náhodně. SFR registry také slouží k odečítání speciálních stavů. U mikrokontrolérů firmy Microchip se nejběžněji používají tyto registry speciálních funkcí. (Malé písmeno x nahrazuje číslo nebo písmeno).

- TRISx: Registr portů kde se jednotlivé piny nastavují jako vstupy či výstupy. Pro vstupy se využívá stav vysoké impedance označován jako třetí logický stav. Pro vstup definujeme konkrétnímu bitu logickou 1, pro výstup pak logickou 0.
- LATx: Registr výstupů. Do těchto registrů zapisujeme logické hodnoty, které mají být na výstupních pinech mikropočítače.
- PORTx: Z těchto registrů můžeme číst aktuální logické hodnoty na odpovídajících pinech mikrokontroléru.
- STATUS: Obsahuje stavové výsledky aritmetickologických operací.

- INTCONx: Nastavení povolení přerušení pro časovače, komunikační jednotky, jejich příznakové bity a priority, povolení externích přerušení a nastavení pul UP odporů na porty.
- TMRxH a TMRxL: Jsou pracovní registry časovačů/čítačů. H (High) byte, L (Low) byte u 16-ti bitových čítačů/časovačů.
- TxCON: Jsou registry nastavení funkcí příslušných čítačů/časovač (zdroj signálu, nastavení děličky apod.).
- ADRESH a ADRESL: Obsahuje výsledek A/D převodníku.
- ADCONx: Nastavení funkce A/D převodníku, které piny mikropočítače mají být nastaveny jako analogový vstup, rychlost konverze, doba nabíjení, a stavové bity spuštění a příznak dokončení operace.
- BAUDCON: Slouží pro nastavení komunikace, její polarity, způsob generování přenosové rychlosti apod.
- SPBRGH: Pracovní registr generování přenosové rychlosti horní byte.
- SPBRG: Pracovní registr generování přenosové rychlosti dolní byte.
- RCREG: Obsahuje právě přijatý byte. Jeho přečtením se vynuluje příznak přerušení od příjmu dat (plného přijímacího zásobníku).
- TXREG: Do tohoto registru se ukládají data, která mají být odeslána. Zápisem do tohoto registru se automaticky vynuluje příznak přerušení odesílání dat (prázdného odesílacího zásobníku).
- TXSTA: Nastavení druhu komunikace, zjišťování jejího stavu a devátý vysílaný datový bit, který lze využívat jako paritní bit či pro určení adresáta nebo pro jiné účely.
- RCSTA: Podobně jako u TXSTA, v případě využití devátého bytu k určení adresáta, lze v tomto registru nastavit ignoraci přijatých dat v případě že nejsou určeny pro tento mikrokontrolér.
- EEADR: Nastavení adresy v paměti EEPROM.
- EEDATA: Zapisovaná případně přečtená data z paměti EEPROM.
- EECONx: Nastavení čtení/zápisu z/do paměti EEPROM a nastavení přístupu do paměti FLASH.
- IPRx: Registry definování priorit přerušení.
- PIRx: Registry příznaků přerušení od periférií.
- PIEx: Registry povolení přerušení od periférií.

3.4.1 Mikrokontrolér PIC18F4553



Obr.4: Blokové schéma mikrokontroléru PIC18F4553

Na obr.4 je blokové schéma mikropočítače firmy Microchip PIC18F4553, který byl použit v elektronické řídicí jednotce. Jsou na něm znázorněny všechny podpůrné subsystémy, které tento mikrokontrolér má. Jedná se o 8-mi bitový mikropočítač s nanoWatt Technology firmy Microchip, kterou je dosažena velmi nízká spotřeba. Je založen na architektuře RISC (Reduced Instruction Set Computer) s oddělenou pamětí pro program a pamětí dat (Harvardská architektura). Mnoho vnitřních subsystémů, implementovaných v jediném pouzdře mikrokontroléru, zmenšuje, zjednodušuje a zlevňuje danou aplikaci tím, že není nutno připojovat mnoho dalších externích součástí. Rozložení vývodů je patrné z Obr.6 nebo z blokového schéma na obr.4.

Charakteristiky mikrokontroléru PIC18F4553:

- Komunikační možnosti
 - Podpora USB V2,0 s pamětí 1-kbytů, včetně zdroje napětí pro buzení sběrnice USB.
 - Rozšířený modul EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter) s podporou třívodičové komunikace SPI (Serial Peripheral Interface) ve všech čtyřech módech a komunikací I²C.
- Periferie
 - Vysoká zatížitelnost vývodu 25mA.
 - Tři zdroje externího přerušení.
 - Čtyři časovací moduly.
 - Podpora pulzně šířkové modulace (PWM) s podporou polovičního i plného můstku pro řízení např. stejnosměrných motorů.
 - 13 Analogových převodníků s vysokým rozlišením 12bitů a programově nastavitelným nabíjecím časem.
 - Dva analogové komparátory.
- Vnitřní flexibilní oscilátory nastavitelné od 31kHz do 8Mhz. Dva módy připojení externího oscilátoru. Monitorování správnosti funkce hodinového taktu.

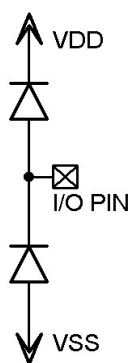
- Další vlastnosti

- Rozšířená a optimalizovaná instrukční sada pro programování v jazyce C.
- 32kBytů paměti Flash pro program, 256Bytů paměti EEPROM.
- 1024Bytů paměť dat a 1024bytů sdílená paměť pro USB.
- Prioritní přerušovací systém.
- Systém kontroly běhu programu (Watchdog) s programovatelnou periodou.
- Systém pro programování v aplikaci (ICSP) včetně debugingu po dvou vodičích.
- Rozsah pracovního napájení 2,0 až 5,5V.

Mikrokontrolér PIC18F4553 má k dispozici 30 I/O pinů rozložených do pěti portů. PORTA, PORTB, PORTC, PORTD a PORTE, které lze nezávisle na sobě nastavit jako vstupní

nebo výstupní v registrech TRISx. (Popsáno v kapitole 3.4). Všechny piny chipu jsou uvnitř chráněny diodami proti potenciálu mimo napájecí napětí.

Princip této ochrany je patrný na obr.5.



Obr.5: Diodová ochrana

TTL vstupy (kompatibilní s TTL logikou) znamená, že logická hodnota 0, pro napájení tohoto mikropočítače 4,5 až 5,5V, je definována od 0V do 0,8V. Logická hodnota 1 pak od 2V do napájecího napětí. Pro jiné napájecí napětí integrovaných obvodů definuje norma TTL logické hodnoty proporcionálně vůči napájecímu napětí. Přejít mezi jednotlivými logickými úrovněmi je poměrně ostrý (s malou hysterezí).

ST (Schmitt Trigger) u tohoto volitelného nastavení se vyhodnocuje logická hodnota vstupu následovně. Logická 0 je definována v oblasti VSS až 0,2VDD (pro napájení 5V je tato hodnota 0V až 1V) a logická 1 v napěťové oblasti 0,8VDD až VDD (pro napájení 5V je to 4V až 5V). Oproti TTL má vstup ST větší hysterezi. Ta se využívá k potlačení změny logické úrovně při drobných zákmitech, neboť v pásmu hystereze zůstává předchozí logická úroveň nezměněna.

PORTA je 7-bitový. Všechny piny tohoto portu lze použít jako digitální výstupy nebo vstupy. Jako analogový vstup lze použít všechny piny PORTuA až na pin RA4, na který je možno přepnout pro přívod hodinového signálu pro TMR0. Nastavení analogového převodníku se provádí v registrech ADCONx. RA4 rovněž lze využít pro přívod signálu z externího budiče sběrnice USB či nastavit jako výstup z komparátoru1. RA2 a RA3 mohou být nastaveny pro přívod analogových referencí, přičemž RA2 může být spojen se vstupem + komparátoru2.

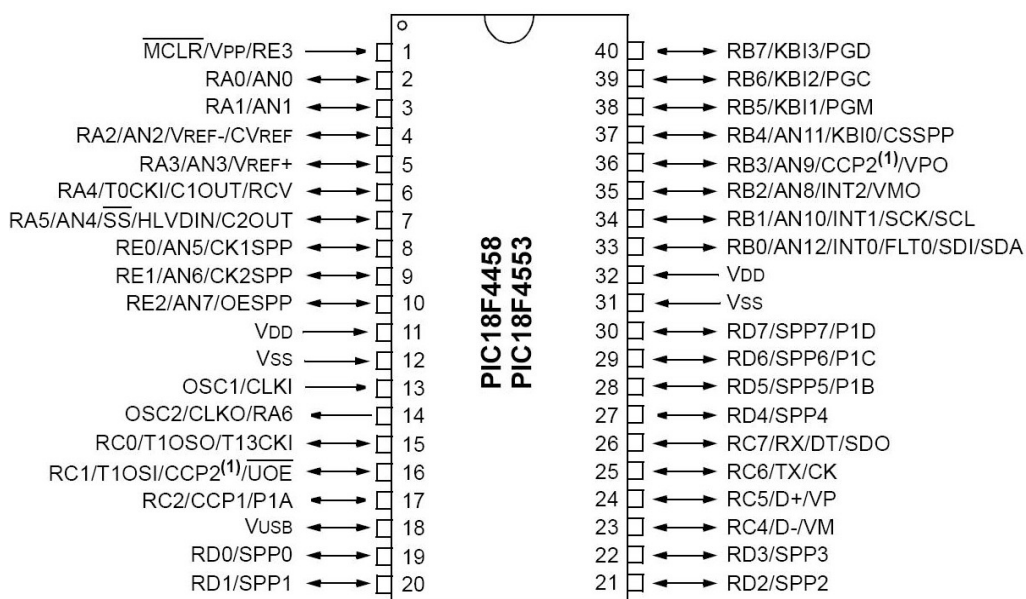
PORTB je 8-bitový. Nastavení pro digitální vstupy/výstupy nabízejí všechny piny tohoto portu. RB0 až RB4 nabízí i analogový vstup. Další speciální funkce jsou na RB0 a RB1 na kterých může být nastavena komunikace I²C, SPI či externí přerušování. Na tomto portu jsou dále podporovány funkce PWM (pulzně šířkové modulace) a přerušování při změně stavu. Zapnutí pull-up rezistoru (cca 10k Ω) na PORTB se projeví u pinů programovaných jako digitální vstup. Pull-up rezistory na odpojeném pinu nastaví potenciál VDD (kladného pólu napájení mikropočítače), což jednoznačně definuje logickou hodnotu. Takový pin se pak spojuje s potenciálem VSS pro dosažení změny logické hodnoty na logickou 0.

PORTC je 7-bitový. RC4 a RC5 lze využít pro komunikaci USB nebo jako digitální vstupy. Na RC6 a RC7 je vyveden EUSART (universální synchronní/asynchronní přijímač/vysílač). Na PORTC jsou rovněž přítomny analogové vstupy, PWM módy a samozřejmě digitální vstupy/výstupy.

PORTD je 8-bitový. Slouží převážně pro digitální vstupy/výstupy. Na RD5 je PWM výstup. PORTD může být nakonfigurován jako paralelní port.

PORTE je 4-bitový. Pin RE3 se většinou konfiguruje jako MCLR (Master Clear). Celý port má na vstupech Schmittův klopný obvod. Piny lze konfigurovat jako digitální vstupy/výstupy či analogové vstupy. Pin RE3 může být pouze vstupní.

Další informace o PIC18F4453 lze nalézt v katalogovém listu na stránkách výrobce: www.microchip.com nebo na příloženém CD.



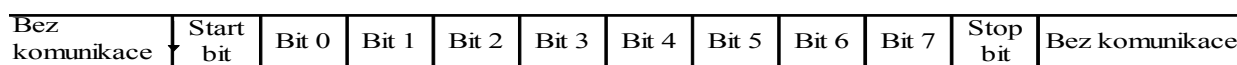
Obr.6: Vývody PIC18F4453

3.5 Komunikace mikrokontroléru s okolím

Pro komunikaci systému s jinými systémy (mikrokontroléry, PC, inteligentními integrovanými obvody apod.) je nejčastěji použito nějakého standardizovaného přenosu dat. Nejčastěji bývá použito sériové komunikace synchronní nebo asynchronní. Dále se hodně používá komunikace I²C. Mezi méně známé patří unikátní obousměrná komunikace probíhající pouze po jednom vodiči, kterou vyvinula firma Dallas Semikonduktor.

3.5.1 Asynchronní komunikace

Asynchronní komunikace přenáší data v určitých sekvencích. Začíná odesláním start bitu, na jehož hranu se synchronizují všechna přijímací zařízení. Dále probíhá přenos vlastních datových bitů, které jsou generovány přesnou přenosovou rychlostí. Proto všechna komunikující zařízení musí obsahovat svůj vlastní zdroj dostatečně přesného oscilátoru. Přenos je ukončen stop bitem. Další přenos je opět odstartován start bitem.



Obr.7: Přenos znaku asynchronní sériovou komunikací

Nejčastější tvar této komunikace začíná start bitem, po něm následuje 8 datových bitů, dále volitelný 9-tý bit (paritní nebo adresa) a nakonec stop bit s volitelnou délkou 1, 1,5 nebo 2 bity. Časová délka bitu se určí z nastavené přenosové rychlosti, udávané v Bd „baudech“ (počet bitů za sekundu) následovně: délka bitu $[s] = 1/Bd$. Standardně se používají rychlosti 2,4; 4,8; 9,6; 19,2; až po 115,2 kBd i více. Přenosová rychlost se volí podle maximální délky vedení a podle požadavků na přenos dat. Například rychlost 19,2kBd se doporučuje pro délku vedení do 15metrů.

Pro ověření správnosti přenášených dat je možno přenášet v „devátém“ datovém bitu paritní bit.

- Sudá parita – počet jedničkových bitů + paritní bit = sudý počet
- Lichá parita – počet jedničkových bitů + paritní bit = lichý počet

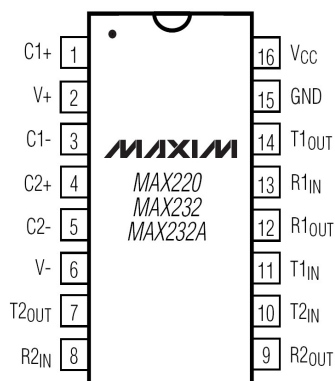
3.5.2 Komunikace RS232

RS232 je rozhraní pro komunikaci vytvořené původně do vzdálenosti 20m. Větší odolnost proti rušení zajišťují logické signály většího napětí, než je standardních 5V (viz. tab.1). Přenos informací probíhá asynchronně, pomocí pevně nastavené přenosové rychlosti. Synchronizace probíhá vždy na začátku přenosu sestupnou hranou.

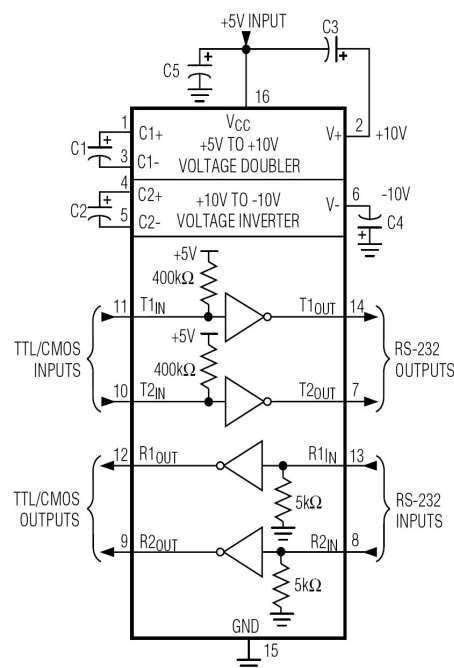
Datové signály		
Úrovně	Vysílač	Přijímač
Log. 0 (L)	+5V až +15V	+3V až +25V
Log. 1 (H)	-5V až -15V	-3V až -25V
Nedefinovaný	-3V až +3V	

Tab.1: Napěťové úrovně datových signálů RS232

Pro připojení sériové asynchronní komunikace mikrokontroléru přímo k rozhraní RS232 osobního počítače slouží integrovaný obvod firmy Maxim MAX232. Tento obvod obsahuje jednu dvojici převodníků úrovní TTL/RS232 a jednu dvojici převodníků RS232/TTL. Napětí pro buzení RS232 si obvod sám vyrobí pomocí nábojové pumpy, která pro svoji funkci používá externí kondenzátory. Aplikační zapojení obvodu včetně externích kondenzátorů je na obr.10. Zapojení vývodů pouzdra DIP/SO 16 na obr.9. Díky tomuto obvodu lze propojit navzájem tyto jinak neslučitelné napěťové úrovně.



Obr.9: Zapojení vývodů



Obr.10: Aplikační schéma obvodu MAX232

Při komunikaci mikrokontroléru s okolím probíhá přenos jedním směrem po jednom vodiči. Vysílací strana je označována TXD a přijímací strana RXD. Signálová zem GND je společná pro všechny signály. Klasické rozhraní RS232 je určeno pro připojení modemu a obsahuje ještě další pomocné řídicí signály. Přehled signálů RS232 v konektoru CANNON9 je v tab.2.

PIN	OZNAČENÍ	NÁZEV	POPIS
1	CD	Carrier Detect	Modem detekoval nosný signál
2	RXD	Receive Data	Tok dat z modemu do terminálu
3	TXD	Transmit Data	Tok dat z terminálu do modemu
4	DTR	Data Terminal Ready	Terminál připraven komunikovat
5	GND	System Ground	Signálová zem
6	DSR	Data Set Ready	Modem připraven komunikovat
7	RTS	Request to Send	Terminál oznamuje modemu, že komunikační cesta je volná
8	CTS	Clear To Send	Modem oznamuje terminálu, že komunikační cesta je volná

Tab.2: Zapojení konektoru CANNON 9 pro rozhraní RS232

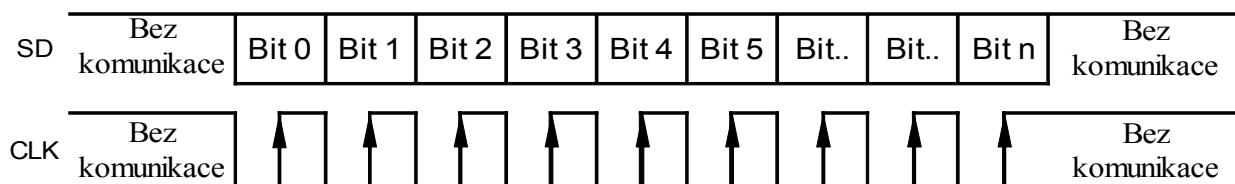
Hodnoty napětí řídicích signálů jsou v tab.3 Tyto signály lze využít i jako zdroj napájecího napětí pro nízko spotřebové aplikace např. v kombinaci s obvodem MAX232.

Řídicí signály		
Signál	Driver	Terminátor
OFF	-5V až -15V	-3V až -25V
ON	+5V až +15V	+3V až +25V

Tab.3: Hodnoty napětí řídicích signálů

3.5.3 Synchronní komunikace

Synchronní přenos informací znamená, že se nastaví na nějakém vodiči nebo vodičích určitá úroveň vyjadřující informaci, která se pak potvrzuje změnou úrovně synchronizačního signálu. Potvrzení platnosti bitu se tedy provede buď sestupnou nebo vzestupnou hranou na hodinovém signálu. Jedná se tedy o jednosměrnou komunikaci vyžadující dva vodiče. Tento přenos dat je podporován mnoha integrovanými obvody jako jsou posuvné registry, inteligentní IO apod. Komunikace probíhá mezi nadřazeným obvodem (MASTER) a podřazeným obvodem (SLAVE). MASTER je většinou inteligentní IO případně mikrokontrolér, zatímco SLAVE může být i sekvenční obvod s pevnou logikou. MASTER ovládá synchronizační signál označovaný jako „CLK“ (Clock), kterým se data kvantují, a po datovém vodiči „SD“ (Serial Data) přijímá nebo vysílá data. Výhodou této komunikace je, že není závislá na čase. Hodinový signál si tedy může MASTER generovat „jak se mu hodí“ ale ne rychleji než je stihne SLAVE zpracovat či než délka vodiče umožní. U obvodů řady 74HC může rychlost generování hodinového signálu dosahovat i 100MHz což odpovídá přenosové rychlosti 100Mbps (megabitů za sekundu). Princip přenosu dat je patrný z obr.11.



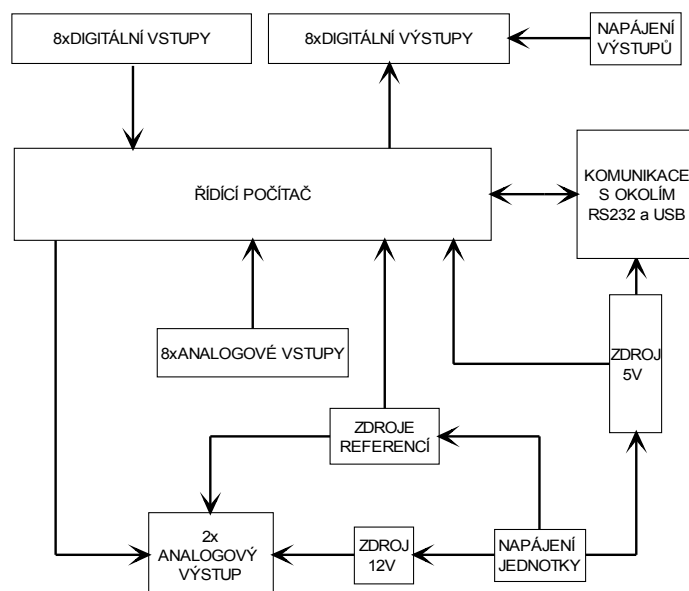
Obr.11: Průběh synchronního přenosu dat

Komunikace SPI (Serial Peripheral Interface) vychází z výše popsané synchronní komunikace. Obsahuje několik dalších pomocných signálů jako např. CS (Chip Select), kterým se vybírá zařízení se kterým se má komunikovat. Maximální rychlost generování hodinového signálu se u SPI běžně pohybuje u 20MHz.

4 Hardwarové řešení

Deska řídicí jednotky je napájena z externího zdroje napětím minimálně 12V, maximálně 30V. Digitální vstupy a výstupy byly galvanicky odděleny od řídicí jednotky optoelektrickými prvky. Digitální výstupy mají vyvedenou napájecí svorkovnici. Jednotka komunikuje s PC pomocí sériové asynchronní komunikace RS232. Rozhraní USB je hardwarově připraveno pro budoucí rozšíření komunikačních prostředků řídicí jednotky s okolím. Použité dvoupatrové svorkovnice usnadňují připojení akčních i kontrolních členů. V nižším patře jsou země jednotlivých periférií, ve vyšším pak jednotlivé vstupy/výstupy. Informace o elektrických parametrech jsou v kapitole 6.1. Na obr.12 je blokové schéma jednotky.

Elektrické schéma, stejně jako návrh plošného spoje, byli navrženy v programu Eagle 4.14. Tento program je firmami velmi často používaný z důvodů nízké ceny. Zdrojové soubory navrženého schéma a plošného spoje, stejně tak demoverze Eagle 4.16r2, jsou na přiloženém CD. Výhoda tohoto programu spočívá v širokém výběru knihovních součástek, snadnou tvorbu schemat i spojů. Formát Eagle podporuje většina firem zabývajících se výrobou plošných spojů. Navržené schéma i plošný spoj lze programem



Obr.12: Blokové schéma řídicí jednotky

zkontrolovat z mnoha hledisek elektrických i mechanických, jako například dodržení minimálních mezer na plošném spoji, zkraty, či nezapojené spoje. Program také obsahuje Autorouter, který automaticky propojí součástky na desce plošných spojů. Ten se však hodí hlavně u vícevrstvých spojů. Navržený plošný spoj je jednostranný, což velmi snižuje jeho cenu v případě výroby. Plošný spoj jsem vyrobil v domácím prostředí, tzv. „tonerovou“ metodou popsanou v kapitole 4.8.

4.1 Výběr vhodného mikropočítače

Pro danou aplikaci je zapotřebí takový mikropočítač, který bude splňovat následující parametry:

- 1) Pro použití mikrokontroléru musí být k dispozici potřebné softwarové i hardwarové vybavení jako je **programátor**, prostředí pro psaní zdrojového textu programu a hlavně **kompilátor** nějakého programovacího jazyka (ASM, jazyk C), který přeloží program do strojového kódu.
- 2) Pro splnění úkolu musí mít mikropočítač: 8 A/D převodníků, 16 I/O pinů pro digitální vstupy a výstupy, několik I/O pinů pro řízení pomocných integrovaných obvodů, podporu komunikace RS232, podporu rozhraní USB pro budoucí připojení a dostatek paměti pro aplikaci.

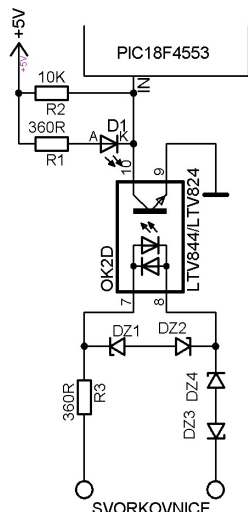
Tyto požadavky splňují mikropočítače firmy Microchip řady PIC18Fxx.

označení	paměť programu	EEPROM	RAM	I/O Pinů	A/D ch	komunikace	časovače	pouzdro
PIC18F4450	16kB	0B	768B	34	13 10-bit	1 -A/E/USART	1-8bit 2-16bit	40/PDIP 44/QFN 44/TQFP
PIC18F4455	24kB	256B	2048B	35	13 10-bit	1 -A/E/USART 1 -MSSP (SPI/I2C)	1-8bit 3-16bit	40/PDIP 44/QFN 600mil 44/TQFP
PIC18F4458	24kB	256B	2048B	35	13 10-bit	1 -A/E/USART 1 -MSSP (SPI/I2C)	1-8bit 3-16bit	40/PDIP 44/QFN 44/TQFP
PIC18F4550	32kB	256B	2048B	35	13 10-bit	1 -A/E/USART 1 -MSSP (SPI/I2C)	1-8bit 3-16bit	40/PDIP 44/QFN 600mil 44/TQFP
PIC18F4553	32kB	256B	2048B	35	13 12-bit	1 -A/E/USART 1 -MSSP (SPI/I2C)	1-8bit 3-16bit	40/PDIP 44/QFN 44/TQF

Tab.4: Výběr vyhovujících mikrokontrolérů

Parametrickým vyhledáváním na stránkách www.microchip.com vznikla tab.4 vyhovujících modelů ze které volím PIC18F4553 v pouzdru 40/PDIP, který má větší rozlišení A/D převodníků (12 bitů) a zároveň dostatek paměti pro budoucí rozšiřování funkcí jednotky.

4.2 Digitální vstupy



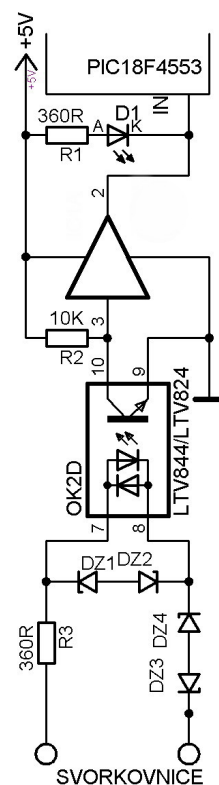
Obr.13: Schéma digitálního vstupu

proudové a napěťové poměry vyhodnocování logického vstupu. Na obr.13 je schéma zapojení digitálních vstupů. Rozletováním pájecích jumperů (SJ1 až SJ7) dostaneme samostatné galvanicky oddělené vstupy.

Bezpečná strana digitálního vstupu je tvořena fototranzistorem s otevřeným kolektorem uvnitř optočlenu. Tento tranzistor připojuje zem ke vstupnímu pinu mikrokontroléru, který je napájen pull-up rezistorem R2. Rezistor R2 zajišťuje logickou hodnotu 1 v případě, že optočlenem neprochází žádný proud. Zelená Led D1, napájena rezistorem R1, indikuje průchod proudu vstupem optočlenu, nikoli logický stav vstupu! Logický stav vstupu zjišťuje mikrokontrolér který má na vstupu Schmittův klopný obvod či TTL. Vzhledem k tomu, že digitální signály by měly být ostré a jasné definované, neměl by být s touto indikací v praxi problém.

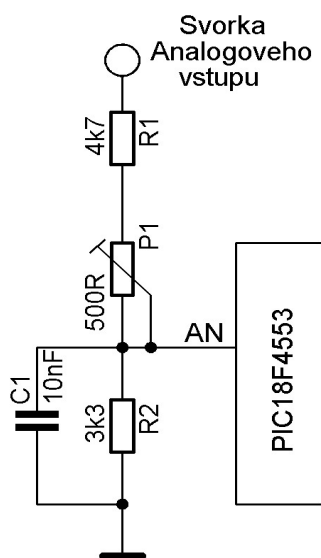
Pokud-bychom chtěli aby zelená LED D1 indikovala skutečný logický stav vstupu, museli-bychom vně mikrokontroléru vytvořit podobný subsystém jako obsahuje mikrokontrolér uvnitř. Návrh řešení je na obr.14.

Digitální vstupy řídicí jednotky byly galvanicky odděleny od řídicího počítače optoelektrickou součástkou LTV844, katalogový list je na přiloženém CD. Jedná se o součástku typu AC-Input. To znamená, že nezáleží na směru průchodu proudu součástkou. Toto je zajištěno antiparalelním zapojením dvou led uvnitř optočlenu na straně vstupu. Antisériové zapojení zenerových diod DZ1 a DZ2 má ochranný charakter, který zachovává funkci AC input. Lze na jejich místo osadit rezistory, které upraví proudové a napěťové poměry vyhodnocení logického vstupu. Odpor R3 je navržen pro vstup 5V a omezuje proud procházející optočlenem. Antisériové zapojení zenerových diod DZ3 a DZ4 slouží ke snížení napětí do vstupu optočlenu o hodnotu zenerova napětí. Diody DZ1 až DZ4 nejsou na desce osazeny. Lze je rovněž osadit rezistory, které upraví



Obr.14: Případné vylepšení digitálního vstupu

4.4 Analogové vstupy



Obr.17: Schéma napěťového děliče analogových vstupů

Řídící jednotka byla vybavena osmi vstupními analogovými kanály. Schéma jednoho takového kanálu je na obr.17. Pro vstup jsem navrhl impedanci cca 10kΩ a rozsah přiváděného napětí od 0V do 10,24V a to i v módu nastaveného rozsahu 0V až 5,12V. Tyto hodnoty vychází z použití základní reference 4,096V. Princip přepínání rozsahů je uveden v kapitole 4.7 Zdroje referencí. Ochrana proti přepětí spočívá jednak v dostatečné vstupní impedanci a jednak v integrované diodové ochraně všech portů mikrokontroléru PIC18F4553, která je na Obr.5.

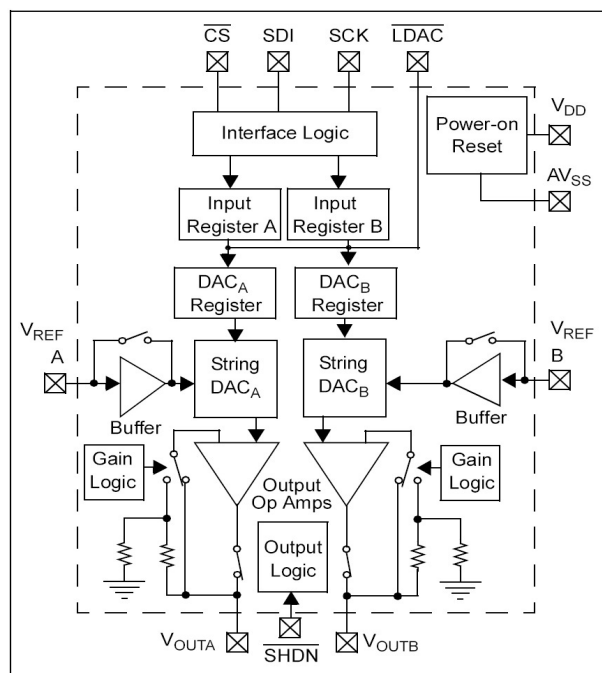
Pro základní rozsah do 10,24V bylo třeba stanovit dělicí poměr napěťového děliče na obr.17. aby výsledná hodnota napětí na analogovém vstupu mikrokontroléru byla právě 4,096V, přičemž je nutno počítat s možností kompenzace nepřesnosti použitých rezistorů použitím trimru (potenciometru) P1. Kondenzátor C1 blokuje měřené napětí.

$$\frac{U_{VS}}{U_{AN}} = \frac{10,24}{4,096} = \frac{(R1 + \frac{P1}{2} + R2)}{R2} = 2,5 \quad (8)$$

Aby byl dělič co nejpřesnější byly použity rezistory s přesností ±1%. Co se týče teplotní stability napěťového děliče, je třeba aby použité prvky byly, pokud možno, ze stejného materiálu (se shodným teplotním koeficientem). Protože uhlíkové rezistory jsou časově nestabilní a vykazují velký šum, jsou pro konstrukce měřicích zařízení nevhodné. Proto jsou v řídicí jednotce použity výhradně metalické rezistory a v měřicích částech přesnější metalizované rezistory firmy ROYAL OHM. Potenciometr pak od firmy VYSHAY, jehož metalická dráha je na keramickém tělesu. Tento trimr se označuje jako cermetový.

Při předpokladu dělicího poměru 2,5, celkové rezistance děliče cca 10kΩ a tolerancí součástek ±1% je v případě zapojení podle obr.17 potřeba kalibrační trimr o velikosti 240Ω. Nejbližší vyšší velikost trimru je 500Ω. Navržené hodnoty R1=4,7kΩ, R2=3,3kΩ a P1=500Ω pak přesně splňují rovnici (8). Ke kalibraci jednoho vstupu by pak mělo postačit pootočit trimrem ze střední hodnoty maximálně o ±45°.

4.5 Analogové výstupy



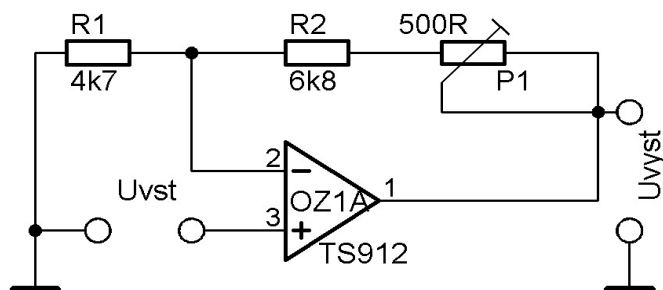
Obr.18: Blokové schéma MCP4922

K realizaci dvou analogových výstupních kanálů bylo použito integrovaného obvodu firmy Microchip MCP4922, který má dva analogové výstupy s vysokým rozlišením 12bitů. S řídicím počítačem komunikuje po sběrnici SPI. Oba kanály (A,B) mají vyvedeny vstupy referencí samostatně na piny VrefA, VrefB. V řídicí jednotce tato vlastnost nebyla využita. Na oba tyto piny se přivádí shodná reference 2,048V.

Důležitou vlastností MCP4922 je programově volitelné zesílení a to v režimu 1x nebo 2x. Princip zesílení je patrný v zapojení operačních zesilovačů na blokovém schématu na obr.18. Tento princip byl využit k přepínání

rozsahů 0 až 5,12V a 0 až 10,24V.

Protože tento obvod neleze napájet pouze napětím v rozsahu 2,7V až 5,5V bude třeba výstupy VoutA a VoutB dále zesílit. V řídicí jednotce nebude k dispozici symetrické napájení, proto bylo třeba použít operační zesilovač typu Rail-to-Rail, jehož výstup může dosahovat hodnot napětí až po hodnotu napájecího napětí. K tomuto účelu byl použit integrovaný obvod firmy ST Microelectronics TS912, který obsahuje dva operační zesilovače v jednom pouzdře DIP-8.



Obr.19: Zapojení operačního zesilovače

Realizované zapojení operačního zesilovače v řídicí jednotce je na obr.19. Jedná se o základní zapojení operačního zesilovače jako neinvertující zesilovač, který zesiluje (násobí konstantou vždy větší než 1) vstupní napětí. Oproti invertujícímu zesilovači, který má vstupní impedanci danou velikostí odporu $R1$, se u tohoto zapojení vstupní impedance blíží nekonečnu a nezávisí na hodnotách odporů $R1$ a $R2$. Pro výstupní napětí platí vztah (9).

$$U_{vyst} = U_{vst} \cdot \left(1 + \frac{(R2 + P1)}{R1}\right) \quad (9)$$

Při přivedené referenci 2,048V na vstup V_{REF} integrovaného obvodu, nastavení výstupu na dekadickou hodnotu 4095 (všech 12bitů na 1) a nastaveném zesílení uvnitř integrovaného obvodu na 1x, musí být napětí na výstupní svorkovnici analogového výstupu příslušného kanálu řídicí jednotky napětí 5,11875V. Podle vztahu (10) bude na výstupu z V_{OUT} napětí 2,0475V.

$$V_{OUT} = \frac{(V_{REF} * G * D_N)}{2^n}, \quad (10)$$

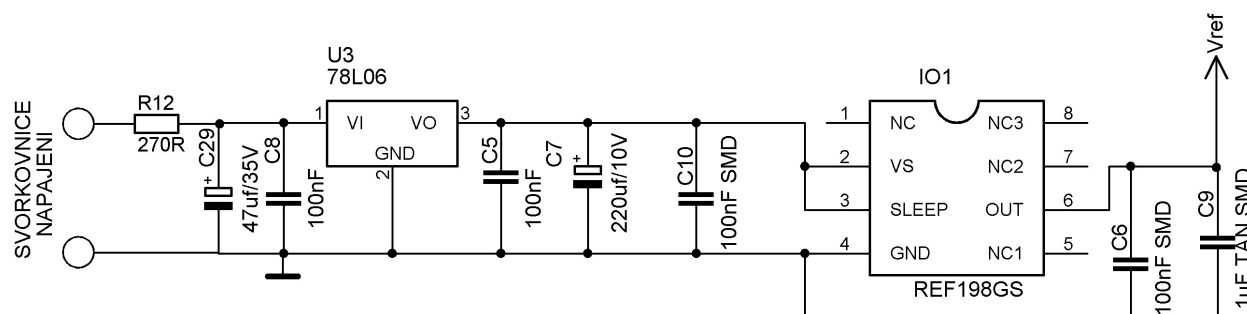
kde G je volitelné zesílení (1x nebo 2x), D_N je zadaná dekadická hodnota výstupu, n je počet bitů (v našem případě 12). Ze získaných hodnot můžeme nyní určit potřebnou hodnotu zesílení K , operačního zesilovače následovně:

$$K = \frac{5,11875}{2,0475} = 2,5 \quad (11)$$

Nyní zbývá určit hodnoty rezistorů $R1$, $R2$ a trimru $P1$, který slouží ke kalibraci podobně jako u analogového vstupu. Impedance zpětné vazby by neměla být příliš velká, ani příliš malá, protože by zbytečně zatěžovala výstup. Rezistory jsem zvolil takto: $R1=4700\Omega$, $R2=6800\Omega$ a trimr $P1=500\Omega$ tak jak je na obr.19. Jedná se o metalizované rezistory s přesností $\pm 1\%$ a cermetový trimr (metalická odporová dráha na keramickém jádru). Tyto hodnoty (při středně nastaveném trimru na 250Ω) dají přesně požadované zesílení 2,5x.

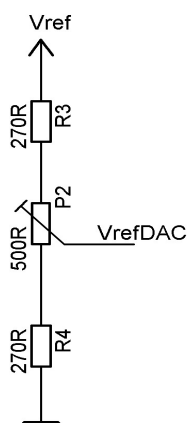
4.6 Zdroje referencí

Pro základní napěťovou referenci byl v řídicí jednotce použit integrovaný obvod firmy Analog Devices REF198. Tento obvod přesné napěťové reference se povedl zakoupit pouze v pouzdru SMD konkrétně SOIC_8. Zapojení obvodu je na obr.20.



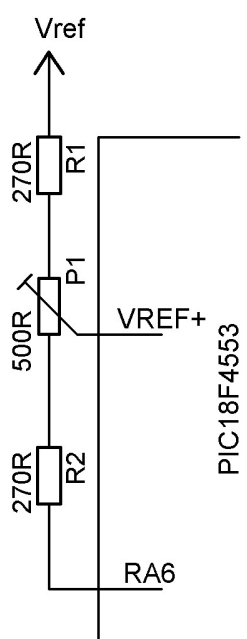
Obr.20: Zapojení reference REF198

Zapojení na obr.20 se skládá ze dvou hlavních částí. Napájení obvodu reference a zapojení vlastního obvodu reference. Jelikož se předpokládá napájení řídicí jednotky 24V, je toto napětí v první fázi sníženo na hodnotu 6V. (Použití napětí 5V, které je v řídicí jednotce k dispozici, by nemuselo za všech okolností zajistit správnou funkci obvodu REF198). O toto se postará kladný stabilizátor 78L06 v pouzdru TO92. Odpor R12 má za úkol snížit jeho tepelnou ztrátu. Napětí 6V je dále přivedeno na obvod stabilizátoru reference, na jehož výstupu OUT (dále Vref) je napětí 4,096V s přesností $\pm 2\text{mV}$. Kondenzátory, které jsou v zapojení na obr.20, zabráňují kmitání stabilizátorů a filtrují šумы.



Obr.21: Zdroj reference pro DAC převodník

Napětí 2,048V, které se přivádí na vstupy DAC převodníku (V_{REFA} a V_{REFB}), se vytváří napěťovým děličem jehož zapojení je na obr.21. Ze zapojení je patrná snaha o dělení napětí na polovinu (dělicí poměr 2), což by mělo být splněno při nastavení trimru P2 na střední hodnotu. Tento trimr slouží ke kompenzaci výrobních nepřesností metalizovaných rezistorů (R3 a R4) a k vlastní kalibraci analogového výstupu. Trimr je cermetový (metalická odporová dráha na keramickém jádru). Tato kombinace rezistorů a trimru zajišťuje dostatečnou teplotní i časovou stabilitu napěťového děliče. Celková impedance děliče představuje přiměřenou zátěž pro zdroj reference.



Obr.22: Princip
přepínání rozsahů

Princip přepínání rozsahů je naznačen na obr.22. Ve své podstatě se jedná o napěťový dělič popsáný v předchozím odstavci na obr.21. Jeho funkce je však podmíněna pinem RA6 řídicího mikrokontroléru. Pokud bude pin RA6 nastaven jako výstupní ($TRISA6=0$) a logická hodnota výstupu na tomto pinu bude L ($LATA6=0$), bude schéma na obr.22 fungovat jako napěťový dělič. Trimr P1 slouží ke kompenzaci výrobních nepřesností metalizovaných rezistorů ($R3$ a $R4$) a k vlastní kalibraci reference (přivedené na pin $VREF+$ řídicího mikropočítače) rozsahu 0V až 5,12V analogového vstupu. Postup kalibrace je popsán v následující kapitole. Trimr P1 je cermetový (metalická odporová dráha na keramickém jádru). Tato kombinace rezistorů a trimru zajišťuje dostatečnou teplotní i časovou stabilitu napěťového děliče. V případě že bude pin mikrokontroléru RA6 nastaven jako vstupní ($TRISA6=1$) bude se nacházet ve stavu vysoké impedance. V tomto případě bude napěťový dělič na obr.22 vyřazen z provozu. Na pin $VREF+$ bude přiváděno napětí přímo ze zdroje reference $Vref$.

4.7 Postup kalibrace

Ke kalibraci analogových vstupů a výstupů řídicí jednotky potřebujeme přesný kalibrační voltmetr. Při kalibraci je nutné proměřit více úrovní (v celém rozsahu) a pro případ nelinearity provést vhodnou linearizaci tak, aby přesnost byla dostatečná v celém rozsahu. U analogových výstupů je třeba pamatovat na závislost výstupu na jeho zatížení. Analogové vstupy zatěžují měřený signál zátěží asi $8,5k\Omega$.

1. Nastavit rozsah na 0V až 10,24V a změřit hodnotu napětí na pinu 5 řídicího mikropočítače. Tento pin je označen jako RA3/AN3/ $VREF+$ a hodnota napětí jako $Vref$.

KALIBRACE ANALOGOVÉHO VÝSTUPU

1. Trimrem P9 nastavit na pinu 14 nebo 11 integrovaného obvodu MCP4922 (piny označeny jako VrefA nebo VrefB) hodnotu napětí $\frac{V_{ref}}{2}$.
2. Pro kanál 0 analogového výstupu proměřit charakteristiku výstupního napětí v závislosti na zadané hodnotě výstupu (v rozsahu 0V až 10,24V). Provést linearizaci a kalibraci pomocí zpětnovazebního trimru P11. I přesto, že použitý operační zesilovač je typu Rail-to-Rail nemusí být dosaženy krajní hodnoty analogového výstupu, zejména pak hodnota 0V!
3. Bod 2. provést pro kanál 1 analogového výstupu za pomoci trimru P12
4. Rozsah 0V až 5,12V nelze nezávisle kalibrovat. Přepínání rozsahů je docíleno volitelným zesílením (1x nebo 2x) v obvodu MCP4922.

KALIBRACE ANALOGOVÉHO VSTUPU

1. Pro nastavený rozsah (0V až 10,24V) a pro kanály 0 až 7 provést linearizaci závislosti skutečné hodnoty napětí na analogovém vstupu a hodnoty udávané řídicí jednotkou. Provést kalibraci pomocí trimrů P1 až P8 (každý trimr je pro kalibraci pouze jednoho kanálu).
2. Pro nastavený rozsah analogového vstupu 0V až 5,12V nastavíme trimrem P10 na pinu 5 řídicího mikropočítače (tento pin je označen jako RA3/AN4/VREF+)

hodnotu napětí $\frac{V_{ref}}{2}$.

4.8 Výroba desky plošných spojů

Elektrická schémata, stejně jako návrhy plošných spojů, byly navrženy v návrhovém programu Eagle 4.14.

Při vlastní výrobě desky plošných spojů, jsem použil již osvědčenou metodu. Jedná se o jednoduchou, levnou a rychlou metodu výroby DPS vycházející z možnosti přenosu motivu na desku nažehlením toneru laserové tiskárny. Tato metoda je známá v nepřeberném množství modifikací, technologických postupů a různých specializovaných médií (papír, fólie..). S níže popsanou metodou mám zatím nejlepší zkušenosti.

Jako médium pro přenos toneru z laserové tiskárny používám barevné lepicí papíry A4, které se dají zakoupit v běžném papírnictví. Listy jsou tenké a lesklé. Důležitá je strana s lepidlem, které je rozpustné ve vodě. Tato strana samozřejmě za sucha nelepí a při průchodu tiskárnou se nenatavuje. Pokud lepidlo při průchodu laserovou tiskárnou zkrystalizuje, jedná se o nevhodný papír.

Nejprve vytiskneme motiv na běžný kancelářský papír. Ustříhneme dostatečně velký kus barevného lepicího papíru. Ústřížek barevného papíru připevníme na běžný list kancelářského papíru tak, aby překrýval místo kam byl předtím proveden zkušební výtisk a aby lepidlo byla navrchu. Připevnění provedeme např. papírovou samolepkou a to na straně, která jako první vjíždí do tiskárny. Provedeme výtisk motivu na lepidlo stranu papíru, je vhodné zvýšit sytost tisku na nejvyšší. Tato metoda jednak snižuje spotřebu barevného papíru a snižuje riziko zpříčení barevného papíru v tiskárně na minimum. Většina tiskáren umožňuje tisk přímo na samotný tento papír. Zásadní výhodou použitého papíru je, že natištěný motiv na něm pevně drží a nestírá se, což se mi při použití originálního papíru stávalo již při průchodu papíru tiskárnou.

Před samotnou fází nažehlování, je třeba destičku cuprexitu zbavit všech oxidů a odmastit. Běžně postačí prášek na nádobí (takový aby nepoškrábal měděnou vrstvu) a poté asi na 5 minut ponoření do roztoku sodného louhu odpovídajícího asi koncentraci vyvolávacího roztoku při použití fotocesty (4 pecičky na cca 50 ml vody) pro úplné odmaštění. Já jsem použil speciální brusnou pastu na kovy od firmy WÜRTH, kterou jsem poté omyl pouze vodou.

Fáze nažehlování je nejdůležitější operací, teplotu a samotný postup je někdy potřeba "natrénovat". Nejdříve se provede „před nažehlením“. Jednotlivé vrstvy od nejspodnější při nažehlování jsou: Měkká podložka (několik listů papíru apod.), motiv, destička mědi směrem dolů, ochranný list papíru, žehlička. Přítlak na žehličku je třeba také vyzkoušet. Po prohřátí je již

papír s motivem přilepen k destičce, což je v několika případech již dostačující a další postup se provádí po vychladnutí destičky. Já ještě provádím doplňující operaci při níž je žehlička plotnou nahoru. Na ní ochranný papír a destička nalepeným papírem nahoru. Papír přejíždím silikonovým nebo gumovým válečkem s přtlakem asi 2kg, který ještě zkvalitní výslednou nažehlenou vrstvu toneru na destičce. Na žehličce byla nastavena teplota “dva puntíky“, celková doba nažehlení cca 2 min.

Po vychladnutí destičky se uplatní další výborná vlastnost použitého papíru a to vrstva lepidla, která se ve vodě rozpustí. Stačí tak ponořit destičku do vody na cca 5min a po té se papír buď sám odlepí od destičky, nebo ho lze snadno sundat bez poškození motivu. Pokud je motiv mírně poškozen lze ho opravit lihovou fixou nebo toner odstranit a pokusit se ho nažehlit znovu.

Toner z tiskárny drží na měděné fólii velmi dobře a před leptáním není třeba s ním cokoli dalšího provádět. Pro leptání lze použít všechny běžné metody.

Po vyleptání odstraníme toner nitro-ředidlem, acetonem či opět práškem na nádobí. Fotky z postupu výroby řídicí jednotky jsou na přiloženém CD.

5 Softwarová část řešení

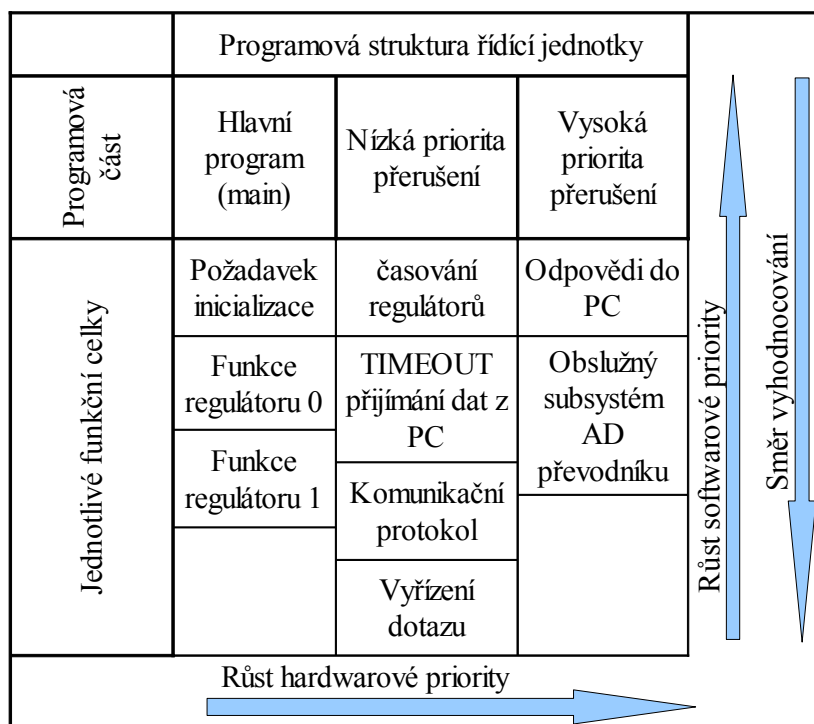
Pro vývoj programu pro mikrokontrolér bylo použito vývojové prostředí HI-TIDE verze 3.11 od firmy HI-TECH Software. Tento program je určený pro programování mikrokontrolérů Microchip PIC18Fxx v jazyce C. Program obsahuje i několik základních knihoven, které se vkládají pomocí direktivy „#include“. Dále umožňuje rozdělení programu do několika celků napsaných v několika zdrojových textech, čímž se zvyšuje přehlednost zejména u rozsáhlých programů. Do zdrojového textu lze vkládat příkazy assembleru pomocí direktivy „asm“ a „#endasm“. Vývojové prostředí také umožňuje simulaci programového kódu, čímž se stává opravdu velmi užitečným nástrojem při vývoji aplikací s jednočipovými mikropočítači.

Pro nahrávání programu do chipu, jsem použil programátor PRESTO od firmy ASIX. Tento rychlý programátor umožňuje programování velkého množství mikrokontrolérů, zejména pak mikrokontroléry firmy Microchip a to přímo v aplikaci. Což umožňuje velmi rychle a pohodlně testovat již napsané části programu bez neustálého přendávání mikrokontroléru z programátoru do aplikace. Programátor se připojuje k osobnímu počítači pomocí USB.

Knihovna obsahující sadu funkcí pro komunikaci osobního počítače a řídicí jednotky byla napsána v prostředí programu Microsoft Visual Studio C++.

Po každém resetu mikropočítače se provádí tzv. inicializace, ve které se provádí nastavení všech potřebných funkcí mikrokontroléru. Při inicializaci se např. určí který pin bude vstupní či výstupní a to pomocí SFR registrů (speciální funkční registry), které lze měnit i při běhu vlastního programu.

5.1 Rozložení programových bloků



Obr.23: Programová struktura řídicí jednotky

Mikrokontrolér PIC18F4553 má dvě úrovně priority přerušení, které jsou v řídicí jednotce využity. Na obr.23 je patrné rozložení programových bloků. Priorita je daná jednak úrovní přerušení a jednak posloupností zápisu funkcí v tomto přerušení. Ta je označena jako softwarová priorita.

5.2 Nastavení portů

PORTA: Celý tento port je byl nastaven jako vstupní.

- RA3: Vstup reference pro AD převodník.
- RA4: slouží pro digitální vstup.
- RA6: Využit pro přepínání rozsahů, vstup či výstup je měněn během programu.
- ostatní piny portu A byly využity jako analogové vstupy.

PORTB: Celý tento port byl nastaven jako vstupní.

- RB2: Použit jako analogový vstup.
- ostatní piny portu B využity pro digitální vstupy.

PORTC: RC0 až RC2 nastaveny jako výstupní. Ostatní piny jako vstupní.

- RC6: Vývod slouží jako TxD(vysílací pin) pro asynchronní komunikaci.
- RC7: Vývod slouží jako RxD(přijímací pin) pro asynchronní komunikaci.
- RC0: slouží pro synchronní komunikaci SPI (SD) serial data.
- RC1: slouží pro synchronní komunikaci SPI (SCK) serial clock.
- RC2: slouží pro synchronní komunikaci SPI (CS) chip select.
- RC4 a RC5: Pro komunikaci USB.

PORTD: celý tento port nastaven jako výstupní a je celý využit pro digitální výstupy.

PORTE: RE0 až RE2 nastaveny jako vstupní a využity pro analogové vstupy.

- RE3: ponecháno pro funkci MCLR (Master clear).

5.3 Nastavení komunikace RS232

Pro komunikaci s osobním počítačem byla použita asynchronní komunikace RS232. Ta probíhá na pinech RC6 a RC7, které musí být nastaveny jako vstupní (TRISC6=TRISC7=1). Komunikace byla nastavena na rychlost 19200 bps, 8datových bitů, bez paritního bitu, 1stop bit. Priorita přerušení od příjmu dat byla nastavena na nízkou a od odeslání na vysokou.

5.3.1 Komunikační protokol

Komunikace PC s jednotkou je založena na principu dotaz-odpověď. Komunikace probíhá podle následujícího komunikačního protokolu. Komunikaci zahajuje PC posláním dotazu tvaru:

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte n	byte (PB+4)	byte (PB+5)
'S'	'K'	CF	PB	DATA[0]	DATA[1]	DATA[n-5]	DATA[PB-1]	'K'

komunikace probíhá po bytech, v případě posílání větších datových typů se posílá nejnižší byte jako první. Komunikace se zahájí odesláním znakového řetězce 'SK' po té následuje CF-číslo funkce jednotky, PB-počet přenášených bytů do ukončovacího znaku 'K', následuje přenos dat (maximálně 20bytů), komunikaci ukončuje znak 'K'. Po doručení dotazu pošle jednotka do PC odpověď která může být trojího tvaru:

- 1) Jednotka pošle znakový řetězec 'OK' který znamená úspěšné vykonání příkazu.
- 2) Jednotka pošle řetězec o velikosti 3bytů z niž první dva znaky jsou 'ER', třetí byte je datový a označuje číslo chyby.
- 3) Jednotka pošle odpověď tvaru:

byte 1	byte 2	byte 3	byte 4	byte 5	byte n	byte (PB+3)
'S'	'K'	PB-1	DATA[0]	DATA[1]	DATA[n-4]	DATA[PB-1]

Datový byte PB-1 obsahuje počet přenášených bytů snížený o 1. Číly maximální počet bytů je 256 (hodnota PB-1=255).

5.3.2 Přehled funkcí přístupných komunikačním protokolem

Řídící jednotka komunikuje podle protokolu uvedeném v kapitole 5.3.1 a poskytuje následující funkce, které jsou očíslovány. U vícebytových datových typů je nejnižší byte poslán/přijímán jako první. Specifické názvy proměnných jsou popsány v kapitole 5.5.

- Č.0: čtení analogových kanálů.
 - Vstup: unsigned char kanal (č. analogového kanálu 0 až 7).
 - Výstup: unsigned short int value (hodnota).
- Č.1: čtení z eeprom od zadané adresy dále.
 - Vstup: unsigned char adresa, unsigned char počet-1 (počet čtených bytů snížený o 1).
 - Výstup: unsigned char data[počet].
- Č.2: zápis do eeprom od zadané adresy dále.
 - Vstup: unsigned char adresa, unsigned char data[PB-1] (pole dat o velikosti PB-1).
 - Výstup: pouze zpětné hlášení.
- Č.3: přímý zápis na digitální výstupy.
 - Vstup: unsigned char hodnota (logické 1 = výstup zapnout, logické 0 = vypnout).
 - Výstup: pouze zpětné hlášení.
- Č.4: zápis na digitální výstupy s funkcí AND (port=port AND hodnota).
 - Vstup: unsigned char hodnota (logické 1 = bez změny, logické 0 = vypnout).
 - Výstup: pouze zpětné hlášení.
- Č.5: zápis na digitální výstupy s funkcí OR (port=port OR hodnota).
 - Vstup: unsigned char hodnota (logické 1 = výstup zapnout, logické 0 = bez změny).
 - Výstup: pouze zpětné hlášení.
- Č.6: přečtení stavu digitálních vstupů.
 - Vstup: žádný byte.
 - Výstup: unsigned char value (negativní logika: logické 0 = aktivní, 1=neaktivní).
- Č.7: nastavení hodnoty analogového výstupu.
 - Vstup: unsigned char kanal (č. kanálu 0,1), unsigned int value (hodnota na výstup).
 - Výstup: pouze zpětné hlášení.

- Č.8: nastavení konfiguračního slova analogového výstupu.
 - Vstup: unsigned char kanal (č. kanálu 0,1), unsigned char DAC_CONF (konfigurační slovo).
 - Výstup: pouze zpětné hlášení.
- Č.9: zjištění všeho o analogových výstupech.
 - Vstup: žádný byte.
 - Výstup: unsigned char DAC_CONF[2] (konfigurační slovo pro kanál 0 a 1), unsigned int value[2] (hodnota analogového výstupu pro kanál 0 a 1).
- Č.10: čtení z paměti RAM, včetně SFR registrů. Od zadané adresy dále.
 - Vstup: unsigned int adresa (adresa v paměti ram), unsigned char počet-1 (počet čtených bytů snížený o 1).
 - Výstup: unsigned char data[počet].
- Č.11: zápis do paměti RAM, včetně SFR registrů. Od zadané adresy dále. (nebezpečné!).
 - Vstup: unsigned int adresa (adresa v paměti ram), unsigned char data[PB-2] (pole dat o velikosti PB-2).
 - Výstup: pouze zpětné hlášení.
- Č.12: Zápis proměnných typu float4 (4bytový float) do paměti EEPROM ve formátu float3(3bytový float).
 - Vstup: unsigned char adresa (adresa v eeprom), float[(PB-1/4)] (velikost pole).
 - Výstup: pouze zpětné hlášení.
- Č.13: Čtení proměnných float3(3bytový float) v paměti EEPROM ve formátu float4(4 bytový float).
 - Vstup: unsigned char adresa (adresa v eeprom), unsigned char počet (velikost pole float4).
 - Výstup: float4[počet] (pole 4bytových float o velikosti počet).
- Č.14: Test komunikace.
 - Vstup: žádný byte.
 - Výstup: pouze zpětné hlášení.
- Č.15: Nastavení povolených vlastností analogových vstupů včetně rozsahu.
 - Vstup: unsigned char Vref_ADC (konfigurační slovo).
 - Výstup: pouze zpětné hlášení.

- Č.16: Zjištění nastavení analogových vstupů.
 - Vstup: žádný byte.
 - Výstup: unsigned char Vref_ADC (konfigurační slovo).
- Č.17: Provede načtení nastavených parametrů v EEPROM a jejich aktivaci. (jako při startu)
 - Vstup: žádný byte.
 - Výstup: pouze zpětné hlášení.
- Č.18: Čtení všech nastavených hodnot týkající se regulátoru 0 nebo 1.
 - Vstup: unsigned char kanal (číslo regulátoru 0,1).
 - Výstup: unsigned char PSD?CON[4] (konfigurační byty), unsigned short int PSD?EMAX, PSD?EAMAX, T, PSD?DUMAX, PSD?UMAX, PSD?UMIN, PSD?UPOS, float4 a?,b?,c?,F? (otazník nahrazuje zadané číslo regulátoru).
- Č.19: Zápis parametrů regulátoru 0 nebo 1.
 - Vstup: unsigned char kanal (číslo regulátoru 0,1), float4 a?,b?,c?,F? (otazník nahrazuje zadané číslo regulátoru).
 - Výstup: pouze zpětné hlášení.
- Č.20: Zápis konfigurace regulátoru 0 nebo 1.
 - Vstup: unsigned char kanal (číslo regulátoru 0,1), PSD?CON[4], unsigned short int PSD?EMAX, PSD?EAMAX, PSD?DUMAX, PSD?UMAX, PSD?UMIN, PSD?UPOS.
 - Výstup: pouze zpětné hlášení.
- Č.21: Zápis žádané hodnoty regulátoru 0 nebo 1.
 - Vstup: unsigned char kanal (číslo regulátoru 0,1), insigned short int value (žádaná hodnota).
 - Výstup: pouze zpětné hlášení.

5.4 Časování regulátorů a TIMEOUT

V řídicí jednotce byly využity dva časovače. Časovač 0 je používán pro generování časů, pro regulátor 0 a 1. Časovače 2 bylo použito pro generování timeoutu při přijímání dat z PC. Priorita obou časovačů byla nastavena jako nízká.

Kmitočet oscilátoru byl nastaven na 8MHz, na vstup děličky je tedy přiváděn kmitočet 2MHz. Přehled generování časů, při různém nastavení děličky, je v tab.5.

NASTAVENÍ ČASOVAČE 0 PRO PERIODY VÝPOČTŮ REGULÁTORŮ					
TOPS2: TOPS0	dělička	vstupní kmitočet do časovače[Hz]	Perioda [us]	nastav.krok. 16bit mód [ms]	maximální čas [ms]
		2 000 000	před naplnění	63 536	
000	2	1 000 000	1	2,000	512
001	4	500 000	2	4,000	1 024
010	8	250 000	4	8,000	2 048
011	16	125 000	8	16,000	4 096
100	32	62 500	16	32,000	8 192
101	64	31 250	32	64,000	16 384
110	128	15 625	64	128,000	32 768
111	256	7 813	128	256,000	65 536

Tab.5: Přehled generování časů

Od periody časovače 0 jsou odvozeny časy vzorkování obou regulátorů. Pomocí komunikace s PC lze měnit nastavení děličky. Pro konkrétní regulátor pak vzorkovací periodu v počtu násobků času přerušení časovače 0 v rozmezí od 1x do 255x. (Zadání nuly znamená čas odpovídající 256-ti násobku času přerušení časovače 0). Časovač byl nastaven jako 16-ti bitový.

Časovač 2 byl nastaven na čas cca 32ms, timeout příjmu dat pak na 10-ti násobek, tedy 320ms.

5.5 Popis specifických proměnných

V řídicí jednotce bylo použito následující symbolické pojmenování proměnných, které jsou specifické pro funkce jednotky. Znak „?“ zde bude nahrazovat číslici analogového vstupního/výstupního kanálu či číslici regulátorů.

- Proměnné s plovoucí řádovou čárkou typu float:
Kr? - zesílení, **Ti?** - integrační složka, **Td?** - derivační složka, **F?** - filtrační složka, **a?**, **b?**, **c?**, bližší informace o těchto složkách v kapitole 2.1.
- Celočíselné bez znaménkové proměnné o velikosti 2byťů(unsigned short int) :

PSD?EMAX – maximální povolená regulační odchylka. Pokud zadána hodnota 0, bude funkce podmíněné regulace vypnuta a hodnota **PSD?EMAX** ignorována.

PSD?EAMAX – hodnota regulační odchylky při které dojde k aktivaci podmíněné regulace.

Regulátor přestane generovat akční zásahy, dokud bude regulační odchylka v pásmu povolené reg. odchylky **PSD?EMAX**.

PSD?DUMAX – omezuje maximální velikost přírůstku akční veličiny (0 = funkce vypnuta).

PSD?UMAX – omezuje maximální velikost akční veličiny (vždy omezena na max 4095, 0=neomezovat).

PSD?UMIN – omezuje minimální velikost akční veličiny (vždy omezena na max 4095, 65535=0 – neomezovat).

PSD?UPOS – posun akční veličiny o konstantu, tato konstanta je odečtena od aktuální akční veličiny a výsledná hodnota je opravena tak, aby splňovala výše uvedené omezující podmínky.

T – základní perioda časovače 0 udávaná v ms.

- Celočíselné bez znaménkové proměnné o velikosti 1byťů(unsigned char) :

PSD?CON0 – zdroj skutečné hodnoty (č. analogového vstupu 0 až 7, 255=regulátor vypnut).

PSD?CON1 – zdroj žádané hodnoty (č. analogového vstupu 0 až 7, 255= po rozhraní RS232).

PSD?CON2 – kanál akční veličiny (č. analogového výstupu 0,1) pozn.: 1 až 7 bit ignorován.

PSD?CON3 – perioda vzorkování vyjádřena v násobku základní periody časovače 0. (0 až 255).
 (V případě zadání 0 bude výsledný čas 256-ti násobek časovače 0!).

T0CON – nastavení časovače 0. Bity odpovídají katalogovému listu PIC18F4553, bit PSA bude

T0CON:							bit 0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
1	0	0	0	-	X	X	X

ignorován nastavením na 0. Horní 4bity musí odpovídat tabulce výše, jinak bude nastavení ignorováno a provedeno implicitně. Hodnoty „X“ lze volit a nastavují děličku, viz tab.5 16-ti bitový mód.

Vref_ADC – nastavení konfigurace AD převodníku. Bity odpovídají katalogovému listu

Vref_ADC							bit 0
-	Vref	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
X	X	X	X	0	1	1	0

PIC18F4553 registru ADCON1. Bit **Vref** je přidán. Vref=1 – reference 4,096V, Vref=0 – reference 2,048V. Dolní 4bity musí odpovídat tabulce výše, jinak bude nastavení ignorováno a provedeno implicitně. Hodnoty „X“ lze libovolně měnit a vyjadřují přiřazení referencí analogovému převodníku, který může být nastaven i jako diferenční. Standardně se používá **VCFG1=0, VCFG0=1**, pro které je proveden přepočít na napětí ve zkušebním programu.

DAC_CONF? - nastavení konfigurace analogového výstupu (DA převodníku). Horní 4bity

DAC_CONF?							bit 0
A/B	BUF	GA	SHDN	-	-	-	-
X	X	X	X	0	1	1	0

odpovídají katalogovému listu MCP4922. Dolní 4bity musí odpovídat tabulce výše, jinak bude nastavení v EEPROM provedeno implicitně. V případě požadavku z PC budou vždy dolní 4bity ignorovány. Pro bit **GA=0** bude hodnota zesílení 2x (rozsah 0-10,24V), **GA=1** odpovídá hodnotě zesílení 1x (rozsah 0-5,12V). Bit **A/B** v bytu **DAC_CONF?** vyjadřuje přiřazení čísla analogového výstupu k fyzickému kanálu. Pro byte **DAC_CONF0** bit **A/B=0** a pro byte **DAC_CONF1** bit **A/B=1**.

5.6 Knihovní funkce jednotky

Všechny knihovní funkce pro komunikaci s jednotkou vrací jeden bez znaménkový byte vyjadřující číslo chyby. Pokud je tato hodnota rovna 1, funkce proběhla úspěšně. Mapa uložení proměnných v paměti EEPROM, které se načítají při inicializaci, je uvedena v příloze.

5.6.1 Funkce související s regulátory

Pole dat s pojmenováním jsou pod textem v tab.6 (otazník nahrazuje zadané číslo regulátoru).

- *unsigned char* **GET_AKUAL_PSD_ALL**(*unsigned char* kanal, *unsigned char* *chardata4x, *unsigned short int* *shortdata7x, *float* *fldata4x)

Funkce získá z jednotky a uloží na zvolené pozice aktuální nastavení a parametry pro zvolený regulátor. *kanal* – číslo regulátoru 0 nebo 1.

- *unsigned char* **SET_PSD_CONF**(*unsigned char* kanal, *unsigned char* *chardata4x, *unsigned short int* *shortdata6x)

Funkce zapíše zvolené nastavení do jednotky. *Kanal* – číslo regulátoru 0 nebo 1,

- *unsigned char* **SET_PSD_PAR**(*unsigned char* kanal, *float* *fldata4x)

Tato funkce zapíše zvolené parametry regulátoru (a,b,c,f). *Kanal* – číslo regulátoru 0 nebo 1,

- *unsigned char* **SET_PSD_PAR2**(*unsigned char* kanal, *float* *fl2data4x)

Funkce přepočítá parametry regulátoru (Kr,Ti,Td,F) podle vztahu (2) na (a,b,c,F) a zapíše do jednotky. *Kanal* – číslo regulátoru 0 nebo 1.

- *unsigned char* **SET_FLOAT4_EEPROM**(*unsigned char* adresa_eeprom, *float* *hodnoty, *unsigned char* pocet)

Funkce uloží do EEPROM (od zadané adresy) předané hodnoty typu float velikosti 4byty ve formátu float velikosti 3byty se kterými počítá řídicí jednotka. *Pocet* – počet hodnot (max 4).

- *unsigned char* **GET_FLOAT4_EEPROM**(*unsigned char* adresa_eeprom, *float* *hodnoty, *unsigned char* pocet)

Funkce přečte z EEPROM (od zadané adresy) hodnoty typu float velikosti 3 byty a uloží do na předanou pozici ve formátu float 4byty. *Pocet* – počet hodnot (max 10).

- *unsigned char* **SET_PSD_V**(*unsigned char* kanal, *unsigned short int* *shortdata1x)

Zapíše předanou hodnotu jako žádanou hodnotu. Kanal – číslo regulátoru 0 nebo 1.

char data4x[4]	unsigned short int shortdata7x[7]	unsigned short int shortdata6x[6]	float fldata4x[4]	float fl2data4x[4]
PSD?CON0	PSD?EMAX	PSD?EMAX	a?	Kr?
PSD?CON1	PSD?EAMAX	PSD?EAMAX	b?	Ti?
PSD?CON2	T	PSD?DUMAX	c?	Td?
PSD?CON3	PSD?DUMAX	PSD?UMAX	f?	f?
	PSD?UMAX	PSD?UMIN		
	PSD?UMIN	PSD?UPOS		
	PSD?UPOS			

Tab.6: Použitá pole dat s názvy proměnných

5.6.2 Funkce analogových kanálů

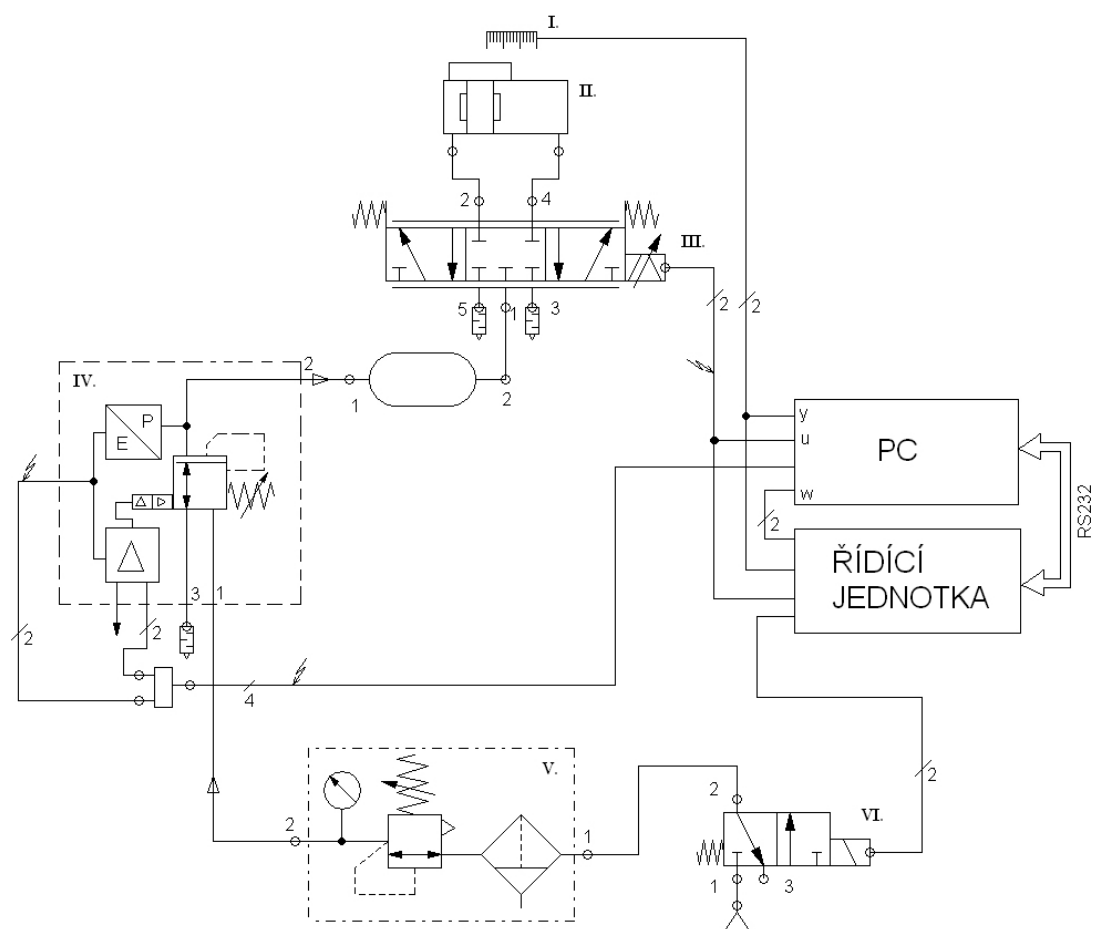
- *unsigned char* **GET_ADC_CH**(*unsigned short int* *hodnota, *unsigned char* kanal)
Získá dekadickou hodnotu představující napětí analogového kanálu. *kanal* (0 až 7)
- *unsigned char* **SET_REF**(*unsigned char* adc_conf)
Nastaví hodnotu *adc_conf* v řídicí jednotce.
- *unsigned char* **GET_REF**(*unsigned char* *adc_config)
Zjistí aktuální hodnotu *adc_conf* používanou v řídicí jednotce.
- *unsigned char* **GET_DAC_ALL**(*unsigned char* *DAC_CONF0, *unsigned char* *DAC_CONF1, *unsigned short int* *value0, *unsigned short int* *value1)
Zjistí a uloží na předané pozice vše o analogových výstupech. *DAC_CONF?* je konfigurace a *value?* je nastavená hodnota výstupu představující napětí.
- *unsigned char* **SET_DAC_CONF**(*unsigned char* config)
Zapíše konfigurační slovo pro analogový výstup. Kanal si funkce určí s obsahu *config*. Bližší popis u proměnné **DAC_CONF?**.
- *unsigned char* **SET_DAC_VALUE**(*unsigned char* kanal, *unsigned short int* value)
Nastaví hodnotu *value* (představující napětí výstupu) na kanal analog.výstupu (0 nebo 1).

5.6.3 Ostatní funkce

- *unsigned char* **READ_EEPROM**(*unsigned char* *adresa_eeprom*, *unsigned char* **data*, *unsigned short int* *pocet*)
Přečte paměť EEPROM od adresy *adresa_eeprom* a data uloží do pole *data*. *pocet* – počet čtených bytů (max 256).
- *unsigned char* **WRITE_EEPROM**(*unsigned char* *adresa_eeprom*, *unsigned char* **data*, *unsigned char* *pocet*)
Zapíše *data* do paměti EEPROM od adresy *adresa_eeprom*. *pocet* – počet zapisovaných bytů (max 19).
- *unsigned char* **READ_RAM**(*unsigned short* *adresa_ram*, *unsigned char* **data*, *unsigned short int* *pocet*)
Přečte hodnoty z paměti RAM od adresy *adresa_ram* a uloží do pole *data*. *pocet* – počet čtených byte (max 256). (Čte i hodnoty SFR registrů jejich adresy jsou uvedeny v katalogovém listu mikrokontroléru na přiloženém CD.)
- *unsigned char* **WRITE_RAM**(*unsigned short* *adresa_ram*, *unsigned char* **data*, *unsigned char* *pocet*)
Zapíše hodnoty z pole *data* do paměti RAM od adresy *adresa_ram*. *pocet* – počet zapisovaných byte (max 18). (zapisuje i hodnoty SFR registrů) Neodborné použití této funkce může způsobit nestabilitu řídicí jednotky, kterou je pak třeba restartovat!
- *unsigned char* **SET_PORT**(*unsigned char* *hodnota*, *unsigned char* *metoda*)
Zapíše hodnotu *hodnota* na port digitálních výstupů zvolenou metodou *metoda*.
metoda = 1 - přímá metoda (logické 1 = výstup zapnout, logické 0 = výstup vypnout)
metoda = 2 - s funkcí AND (logické 1 = bez změny, logické 0 = výstup vypnout)
metoda = 3 - s funkcí OR (logické 1 = výstup zapnout, logické 0 = bez změny)
- *unsigned char* **GET_PORT**(*unsigned char* **hodnota*)
Uloží do *hodnota* aktuální stav digitálních vstupů. Logické 1 = vstup aktivní
- *unsigned char* **Init_EEPROM**(*void*)
Provede inicializaci parametrů uložených v EEPROM. (Provádí se i při startu jednotky)
- *unsigned char* **TestKomunikace**(*void*)
Provede pokus komunikace s jednotkou. Vrací číslo chyby jako všechny ostatní funkce. Pokud je vrácená hodnota = 1 komunikace proběhla v pořádku.

6 Ověření správnosti a funkčnosti řešení

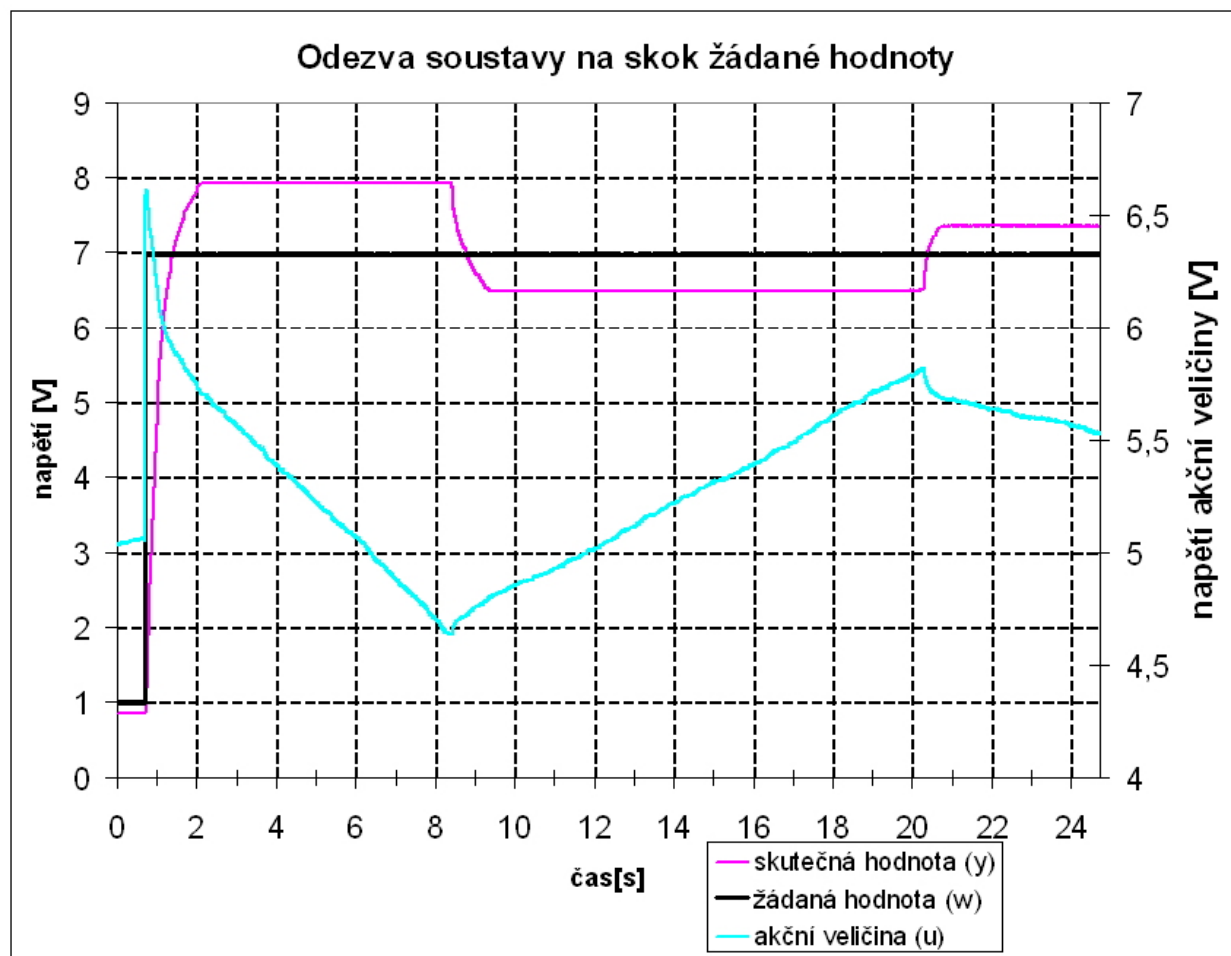
Funkčnost regulátorů, stejně tak analogových i digitálních vstupních a výstupních kanálů, byla ověřena v laboratoři pneumatických systémů na katedře aplikované kybernetiky TUL. Na obr.24 je zapojení pneumatického obvodu, na kterém se testovala funkčnost regulátorů řídicí jednotky[9].



Obr.24: Pneumatické schéma

Popis členů pneumatického obvodu (obr.24):

- I. Odporový snímač polohy (potenciometr) FESTO MLO POT-225-TLF.
- II. Pneumatický válec DGPL 25-225-PPV-A-KF-B.
- III. Elektro-pneumatický proporcionální průtokový ventil FESTO MPYE 5-1/8-LF-010B.
- IV. Elektro-pneumatický proporcionální tlakový ventil FESTO MPPE 3-1/4-10-010B.
- V. Redukční ventil s filtrem FESTO LFR 1/4-5M-MINI.
- VI. Ventil FESTO HEE 1/4-MINI-24.



Graf 1: Odezva systému na skok žádané hodnoty

Na grafu 1 je znázorněna reakce soustavy polohového pneumatického obvodu (obr.24) na skok žádané hodnoty w (polohy) z 1V na 7V. Parametry regulátoru, pro vyzkoušení jeho funkčnosti, byly nastaveny takto: $K_R = 0,24$; $T_I = 0,3$; $T_D = 0$ a vzorkovací čas 10ms.

6.1 Elektrická specifikace

Absolutně maximální parametry, při překročení může dojít k trvalému poškození.

Parametr		Hodnota	Jednotka
Napájecí napětí jednotky		35	V
Napětí analogového vstupu ^(*2)		±25	V
Maximální zátěž analogového výstupu		±130	mA
Digitální výstupy	Napájení	35	V
	špičkový proud ^(*1)	6	A
	zpětný proud ^(*1)	-1	A
digitální vstup ^(*1)		±50	mA

Ostatní elektrické parametry:

Parametr		Min	Typ	Max	Jednotka
Napájecí napětí jednotky		12	-	30	V
odběr jednotky ^(*3)		20	130	300	mA
Analogové vstupy	napětí	0	-	10,24	V
	rozlišení (0-5,12V)	-	0,5	-	mV
	rozlišení (0-10,24V)	-	1	-	mV
	vstupní zátěž	7,9	-	-	kΩ
Analogové výstupy	napětí 0V	0	-	45	mV
	proud nakrátko	-	-	80	mA
	vnitřní odpor	-	90	115	Ω
Digitální výstupy	napájení	2,5	-	35	V
	stálý proud ^(*4)	-	-	1	A
Digitální vstupy (5V) ^(*5)	Logická 0	0	-	±1,7	V
	Logická 1	±3,2	±5	±10	V
Digitální vstupy ^(*5)	Logická 0	0	-	±2,8	mA
	Logická 1	±5,2	±10	±25	mA

(*1) vztaženo na jeden kanál.

(*2) Napětí přivedené na analogový vstup větší než zvolený rozsah, zejména pak záporné, může způsobit po dobu působení tohoto napětí značnou nepřesnost všech analogových kanálů.

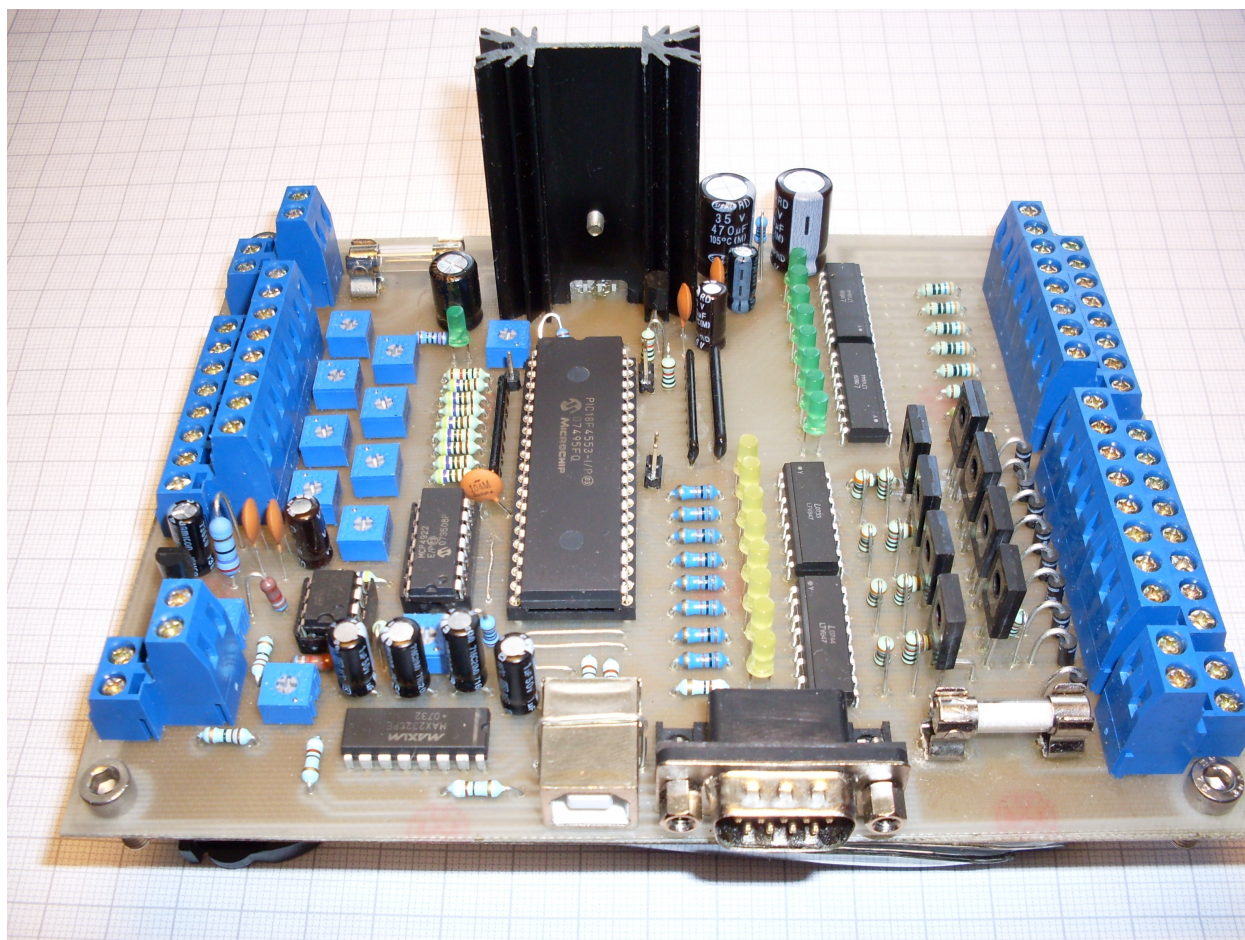
(*3) V závislosti na aktuálních aktivních vstupech/výstupech.

(*4) Záleží na chlazení tranzistorů a jejich tepelné ztrátě.

(*5) Hodnoty lze v širokém rozsahu přizpůsobit osazením rezistorů nebo zen.diod. viz obr.13.

7 Závěr

Cílem této diplomové práce bylo vytvořit elektronickou řídicí jednotku, která by ukázala praktické využití jednočipových mikropočítačů v oblasti řízení a regulace technologických procesů a její možnosti komunikace s nadřazenými periferiemi. Funkčnost regulátorů, stejně tak analogových i digitálních vstupních a výstupních kanálů, byla ověřena v laboratoři pneumatických systémů na katedře aplikované kybernetiky TUL. Fotografie realizované řídicí jednotky je na následujícím obrázku.



Samostatnému návrhu a sestavení jednotky se věnují kapitoly čtyři a pět. Kapitola čtyři se věnuje hardwaru. Je zde popsáno řešení dílčích funkčních částí jednotky a elektrické zapojení. Kapitola pět se věnuje softwarovému vybavení jednotky a to jak mikrokontroléru tak i osobního počítače. Je zde přiblíženo uspořádání programových celků a procesů v mikrokontroléru z hlediska jejich členění a priorit. Druhá část kapitoly pět se věnuje funkcím elektronické řídicí jednotky a komunikačnímu protokolu mezi jednotkou a osobním počítačem.

Dále jsou zde popsány funkce řídicí jednotky, které poskytují jak knihovní funkce, tak samotný komunikační protokol.

Celý systém řídicí jednotky nelze považovat za konečný a vyčerpaný. V paměti mikrokontroléru zbylo dostatek prostoru pro přidání mnoha dalších funkcí jako např. režim smplování analogových kanálů, možnosti rozšíření komunikace s osobním počítačem prostřednictvím rozhraní USB či naprogramování autonomního řízení nějakého procesu s využitím nejen již implementovaných dvou PSD regulátorů ale i dvou portů digitálních vstupních a výstupních kanálů. Hardware jednotky je téměř vyčerpán. Nezbyly zde žádné volné piny řídicího procesoru mimo pinu MCLR. Deska plošného spoje umožňuje modifikaci digitálních vstupních kanálů ve smyslu interpretace logické úrovně z hlediska velikosti napětí či proudu procházejícím digitálním vstupem.

Použitá literatura :

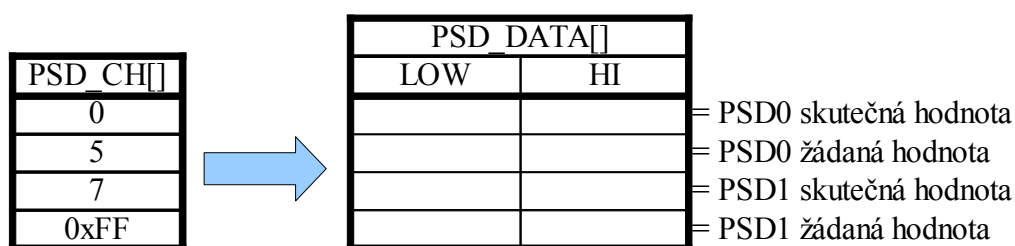
- [1] BALÁTEĚ, J.: Vybrané statě z automatického řízení. Brno: Vysoké učení technické v Brně, 1996.
- [2] MODRLÁK, O.: Analýza dynamických systémů. Liberec: Technická univerzita v Liberci, 2004.
- [3] HI-TEC PICC-18 Compiler. Acacia Ridge: HI-Tech Software Pty. Ltd., 2006.
- [4] Katalogové listy firmy MICROCHIP, www.microchip.com.
- [5] www.hw.cz
- [6] www.cmail.cz/doveda
- [7] www.cs.vsb.cz/licev
- [8] VACEK, V.: Sériová komunikace ve WIN32, Praha, BEN-Technická literatura 2003.
- [9] MOUČKA, M.: Identification of the pneumatic positional system, Acta Mechanica Slovaca, 1-A/2008.

Seznam příloh:

A – Obslužný program AD převodníku.....	50
B – Proměnné uložené v EEPROM.....	51
C – Implementace komunikačního protokolu.....	52
D – Schéma zapojení řídicí jednotky.....	55
E – Tištěný spoj.....	56
F – Osazovací schéma.....	57
G – Seznam součástek.....	58
H – Zdrojový text hlavičkového souboru knihovny funkcí funkce.h.....	59

A - Obslužný program AD převodníku

Obslužný subsystém AD převodníku pracuje v přerušení vysoké úrovně, jak je znázorněno na obr.23. Ten prioritně zpracovává žádosti čtení hodnoty analogových kanálů, které jsou určeny pro oba regulátory. Tyto žádosti jsou v bytovém poli velikosti 4 „PSD_CH“ ve kterém jsou uloženy čísla kanálů, nebo značka 0xFF znamenající již provedený úkol nebo žádný požadavek. Tento systém funguje autonomně dokud jsou v poli žádostí obsaženy položky různé od 0xFF. Jinak ho je třeba inicializovat vyvoláním přerušení od analogového převodníku. Příklad chování tohoto subsystému je na obr.P1.



Obr.P1: Příklad chování subsystému řídicí AD převodník

V situaci jako je na obr.P1 provede subsystém vložení hodnoty analogového kanálu č.0 na pozici skutečné hodnoty pro regulátor 0. Dále kanál č.5 na pozici žádané hodnoty pro regulátor 0 apod. Značka 0xFF (hodnota 255) znamená buď že subsystém danou operaci již provedl, nebo ji provádět nemá. V tomto případě je žádaná hodnota zadávána z PC.

B - Proměnné uložené v EEPROM

Po startu jednotky se načítají nastavení z paměti EEPROM. Ta je využita od adresy 0 do adresy 59. Změna těchto hodnot se projeví po restartu jednotky, nebo při spuštění funkce č.17.(načtení EEPROM parametrů). Zbylé místo (196 bytů) je možno jinak uživatelsky využít. Seznam proměnných a jejich jednotlivé lokace jsou k dispozici v tab.P1. význam proměnných pak v kapitole 5.5.

PŘEHLED PROMĚNNÝCH ULOŽENÝCH V EEPROM A VOLNÉ PAMĚTI					
Adresa EEPROM	Název proměnné	Adresa EEPROM	Název proměnné	Adresa EEPROM	Název proměnné
0	Kr0	21	F1	42	PSD1UMAX
1		22		43	
2		23		44	PSD1UMIN
3	Ti0	24	PSD0EMAX	45	
4		25		46	
5		26	PSD0EAMAX	47	PSD1UPOS
6	Td0	27		48	
7		28	PSD0DUMAX	49	PSD0CON0
8		29		50	PSD0CON1
9	F0	30		51	PSD0CON2
10		31	PSD0UMAX	52	PSD0CON3
11		32		53	PSD1CON0
12	Kr1	33		54	PSD1CON1
13		34	PSD0UPOS	55	PSD1CON2
14		35		56	PSD1CON3
15	Ti1	36		57	T0CON
16		37	PSD1EMAX	58	Vref_ADC
17		38		59	DAC_CONF0
18	Td1	39		60	DAC_CONF1
19		40	PSD1DUMAX	...	VOLNÁ PAMĚŤ
20		41		255	

Tab.P1: Přehled využití paměti EEPROM

C - Implementace komunikačního protokolu

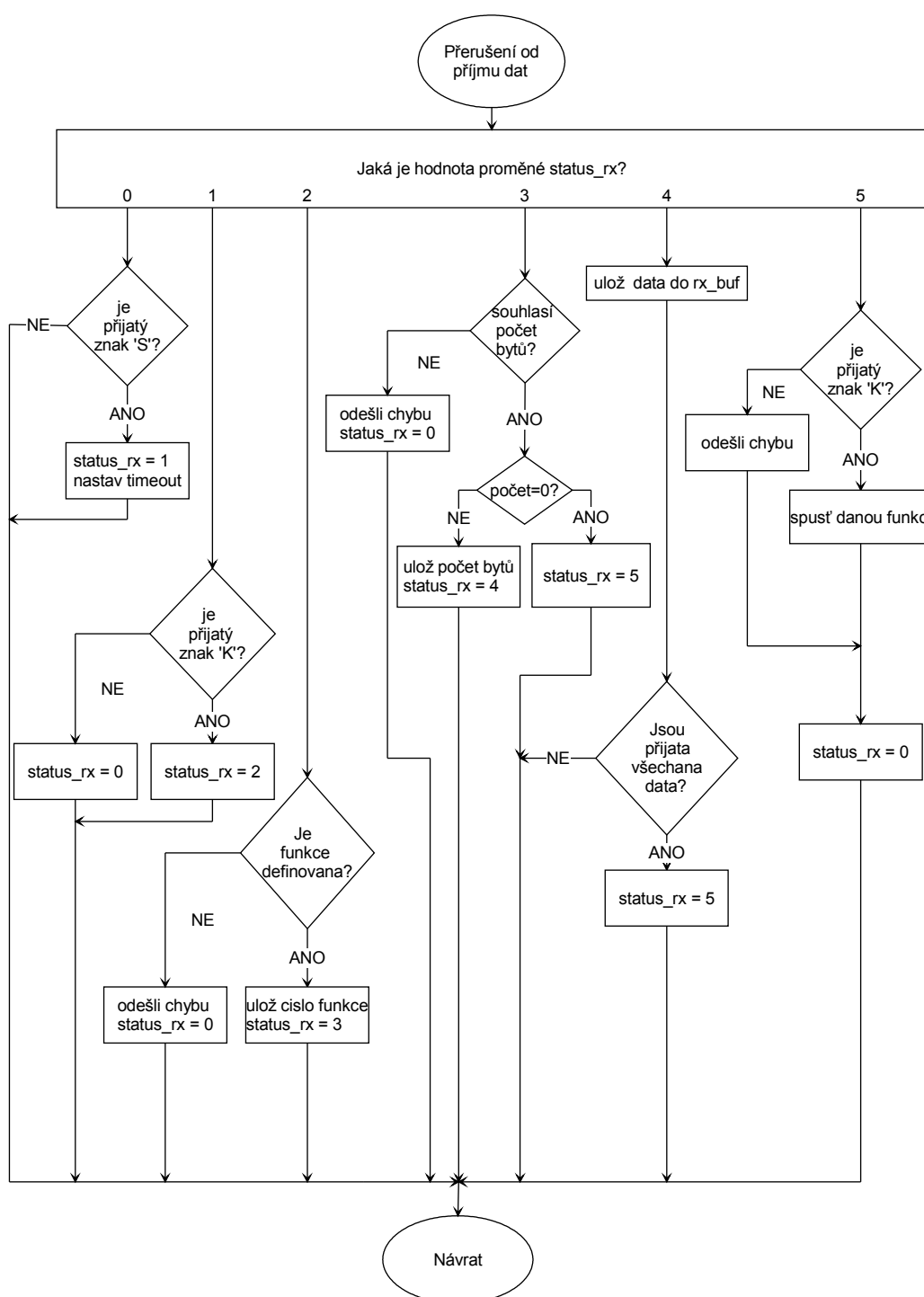
Počítač komunikuje s PC pomocí komunikačního protokolu popsaného od kapitoly 5.3 a následujících. Pokud hardware řídicího mikrokontroléru obdrží jakýkoli byte, vygeneruje příznak přerušení nízké úrovně a zároveň nastaví příznak přerušení od příjmu dat. V tuto chvíli mohou nastat pouze tři případy:

1. Systém se nachází v hlavní funkci main, jejíž vykonávání se přeruší a začne se vykonávat obsluha přerušení nízké úrovně od začátku. V té se zjistí která událost nastala a provede se její vyhodnocení.
2. Systém již vykonává některou z úloh nízké priority přerušení. V tomto případě kompletně dokončí předchozí úlohu a pokračuje dál v průchodu přerušením nízké priority. V tomto průchodu novou událost buď vyhodnotí nebo se po výstupu spustí průchod tímto přerušením znovu.
3. Systém se nachází ve vysoké úrovni přerušení. V tomto případě bude průchod obsluhy přerušení s nízkou prioritou spuštěn po dokončení všech úloh s vysokou úrovní priority.

Průchod obsluhou přerušení nízké úrovně je patrný na obr.23 ve sloupci nízká priorita přerušení. (Obsluha přerušení nízké priority může být přerušena obsluhou přerušení s vysokou prioritou!).

Aktuální stav vyhodnocování příchozích dat vyjadřuje bytová proměnná „status_rx“, která nabývá hodnot 0 až 5. Pokud vyprší čas určený na vyhodnocení přijímaných dat, nastaví se „status_rx“ na hodnotu 0 čímž vyhodnocování začne od začátku. Na obr.P2 je vývojový diagram vyhodnocování přijatých dat.

Přijatá informace o číslu funkce a přenášeném počtu bytů se porovná s hodnotou statické tabulky „funkce“, která obsahuje kolik jaká funkce vyžaduje bytů. Pokud funkce může mít proměnlivou délku požadovaných bytů, je v tabulce „funkce“ uložena značka hodnoty 0xFF. Přijatá data se ukládají do pole „RX_BUF“ velikosti 20 bytů. Pokud vše proběhne podle protokolu, zavolá se funkce „exe_func“, která vykoná příslušnou operaci.

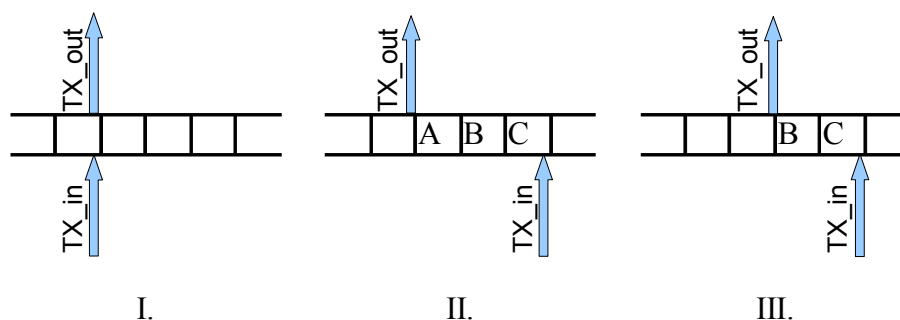


Obr.P2: Vývojový diagram implementace komunikačního protokolu

Pro odpověď se využívá funkcí:

- `add_txbuf(byte)`: Tato funkce přidá předaný byte do fronty pro odesílání. Zároveň kontroluje stav fronty a zajišťuje start jejího odesílání. Je to nejnižší forma přístupu a ostatní vyšší funkce volají tuto funkci.
- `send(adresa,pocet)`: Této funkci se předá adresa prvního bytu a počet bytů, ta tyto byty přidá do fronty pro odesílání.
- `add_OK()`: přidá do fronty znakový řetězec 'OK'.
- `add_ER(byte)`:přidá do fronty znakový řetězec 'ER' a hodnotu „byte“
- `add_SK(byte)`:přidá do fronty znakový řetězec 'SK' a hodnotu „byte“

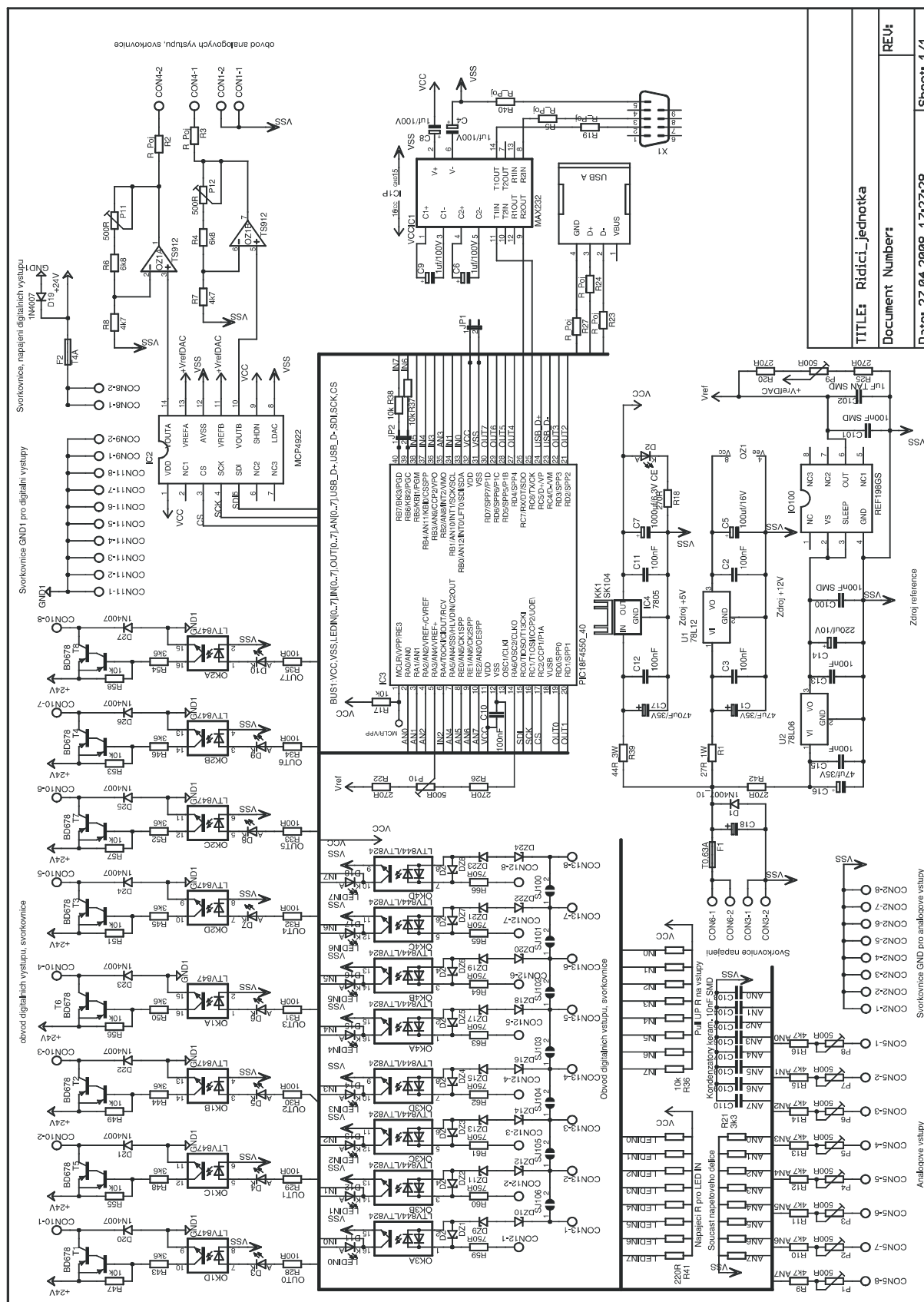
Fronta, kterou plní výše uvedené funkce, je implementována na poli „TXBUF“ o velikosti 255bytů. Její „vyprazdňování“ provádí po inicializaci blok odpovědi do PC na obr.23. Tento subsystém pracuje v nejvyšší prioritě přerušování.



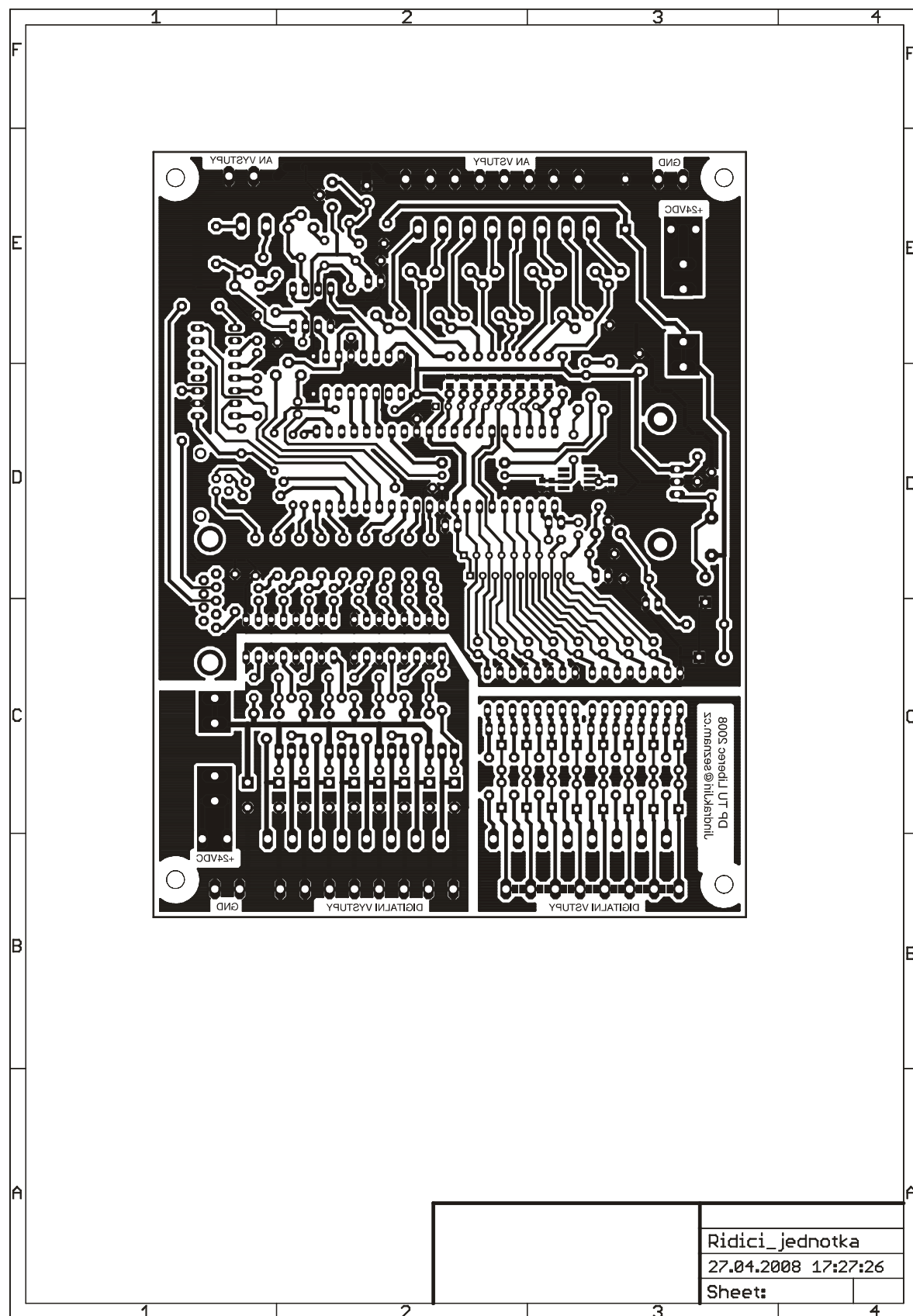
Obr.P3: Odesílací fronta I.) prázdná, II.) po přidání prvků, III.) po odebrání prvku

Na obr.P3 je vyobrazena implementovaná fronta odesílání dat do PC. I) fronta neobsahuje žádná data k odeslání, II) stav fronty po přidání prvků A,B,C, III) stav fronty po odeslání prvku A. Proměnná „TX_in“ obsahuje index, který v poli „TXBUF“ označuje konec fronty. Index v proměnné „TX_out“ ukazuje prvek fronty, který byl odeslán v minulém kroku.

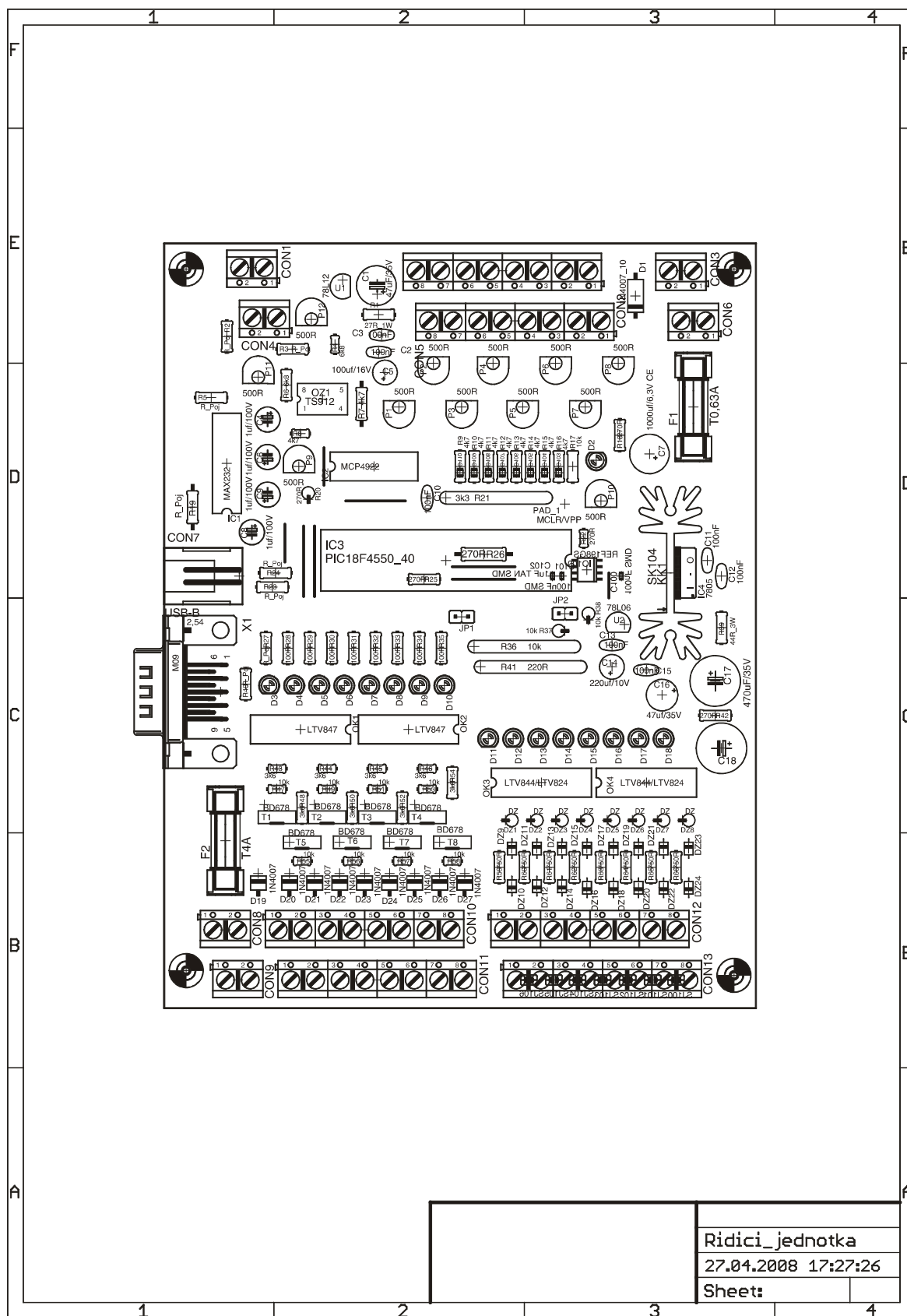
D - Schéma zapojení řídicí jednotky



E - Tištěný spoj



F - Osazovací schéma



G - Seznam součástek

Počet	Název	Popis	Součástka na DPS
1	CE 47/35PHT	Elektrolytický kondenzátor 47uF 35V 105°C 5x11	C1, C16
2	CSMD100N0603X	Keramický kondenzátor SMD 100nF 0603 16V X7R 10%	C100, C101
1	TC 1/16-SMDA	Tantalový kondenzátor SMD 1uF 16V velikost-A 20%	C102
8	CSMD10N0805X	Keramický kondenzátor SMD 10nF 0805 50V X7R 10%	C103..C110
1	CE 220/10PHT	Capacitor electrolytic 220uF 10V 105°C 5x11 r=2mm	C14
1	CE 470/35P	Elektrolytický kondenzátor 470uF 35V 10x16	C17
1	CE 470/35P	Elektrolytický kondenzátor 470uF 35V 10x16	C18
7	CCK 100N	Keramický kondenzátor 100nF 100V rozteč 5 mm	C2, C3, C10, C11, C12, C13, C15
4	CE 1/100PHT-Y	Elektrolytický kondenzátor 1uF 100V 105°C 5x11	C4, C6, C8, C9
1	CE 100/16PHT	Elektrolytický kondenzátor 100uF 16V 105°C 5x11 mm	C5
1	CE 1000/10PHT	Elektrolytický kondenzátor 1000uF 10V 105°C 10x12,5 mm	C7
6	TB-5.0-P-2X2F	Svorkovnice do DPS patrová rozteč 5,0 mm 2x2 piny	CON1..13
6	TB-5.0-P-2X3F	Svorkovnice do DPS patrová rozteč 5,0 mm 2x3 piny	CON1..13
1	USBB-G	Zásuvka USB-B úhlová do plošného spoje	CON7
10	1N4007	Usměrňovací dioda univerzální 1A 1000V DO41	D1,D19..D27
9	LL-304GDF	Dioda LED 3mm s plochým čelem zelená 125° 10mcd	D2,D11.. D18
8	LL-304YDF	Dioda LED 3mm s plochým čelem žlutá 125° 15mcd	D3..D10
24	Neosazeno	Možno osadit rezistory či zen.diodami dle potřeby	DZ1..DZ24
1	ZKM 0.63A	Pojistka středně zpožděná skleněná 5x20mm 0,63A, možno i méně A	F1
1	ZKM 4A	Pojistka středně zpožděná skleněná 5x20mm 4A+zháš.náplň, možno i více A	F2
1	MAX232CPE+	Integrovaný obvod RS232 Receiver/Tran.Maxin DIP16	IC1
1	MCP4922	12-Bit DAC with SPI™ Interface	IC2
1	PIC18F4550-I/P	Integrovaný obvod 32kB EnhFlash 2k RAM FS-USB 2.0 DIP40, možno i PIC18F4553	IC3
1	L7805ABV	Integrovaný obvod stabilizátor+5V 1,5A Wer.Ind. TO220	IC4
1	REF198GS	Integrovaný obvod napěťová reference 4,096V SO8	IO100
1	ZL201-10G	Kolíková lišta do DPS přímá rozteč 2,54 mm 1x10 pinů	JP1, JP2,MCLR/VPP
1	SK104	chladič	KK1
2	LTV847	Optočlen 4násobný 5kV 35V 0,05A DIP16	OK1, OK2
2	LTV844	Optočlen 4násobný AC-IN 70V 5kV DIP16	OK3, OK4
1	TS912IN	Integrovaný obvod 2x operační zesilovač RailToRail DIP8	OZ1
12	3362P-500R	Potenciometr cermetový ležatý 0,25W 1/4" 500 Ohm	P1..P12
1	1W 27R	Metalizovaný rezistor 1W 27R 5%	R1
12	M0.6W 10K	Metalizovaný rezistor 0,6W 10K 1%	R17, R36..R38, R47, R49, R51, R53, R55..R58
6	M0.6W 270R	Metalizovaný rezistor 0,6W 270R 1%	R18, R20, R22, R25, R26, R42
2	R_poj	Pojistný odpor 11R, možno proklemovat	R2, R3
1	DR3.3K-8/9	Rezistorová síť 8x3,3K 9 pinů 2%	R21
8	M0.6W 100R	Metalizovaný rezistor 0,6W 100R 1%	R28..R35
2	KNP03WS-47R	Drátový rezistor výkonový 3W 47R 5% Royal Ohm	R39
2	M0.6W 6K8	Metalizovaný rezistor 0,6W 6K8 1%	R4,R6
1	DR220R-8/9	Rezistorová síť 8x220R 9 pinů 2%	R41
8	M0.6W 3K6	Metalizovaný rezistor 0,6W 3K6 1%	R43..R46, R48, R50, R52, R54
6	R_poj	Pojistný odpor 91R, možno proklemovat	R5, R19, R23, R24, R27, R40
8	M0.6W750R	Metalizovaný rezistor 0,6W 750R 1%	R59..R66
10	M0.6W 4K7	Metalizovaný rezistor 0,6W 4K7 1%	R7.. R16
8	BD678	Tranzistor PNP Darlington 60V 4A 40W TO126	T1..T8
1	78L12	Integrovaný obvod stabilizátor 12V 0,1A TO92	U1
1	78L06	Integrovaný obvod stabilizátor napětí 6V 0,1A Pos. TO92	U2
1	DHP8-09M	Vidlice úhlová D-SUB standard 7,2 mm 9 pinů plast.kr.	X1

H - Zdrojový text hlavičkového souboru knihovny funkcí funkce.h

```

/*****
// Hlavičkový soubor funkcí pro komunikaci s řídicí jednotkou
/*****
#include "funkce.c"          // zdrojový text funkcí

//-----
// OpenPort
// otevře předané číslo COM portu, číslo se předává jako znak.
//-----
unsigned char OpenPort(char cislo);

//-----
// Close Port
// zavře otevřený port,
//-----
unsigned char ClosePort();

//-----
// ReadData
// přijímání dat z COM, vstupuje pointer na buffer a počet bytu, zapisuje do pole buffer
//-----
unsigned char ReadData(char *buffer,int size);

//-----
// WriteData
// odeslání na COM, vstupuje pointer na buffer a počet bytu
//-----
unsigned char WriteData(char *buffer,int size);

//-----
// READ_EEPROM
// čtení z paměti EEPROM, vstupuje adresa v EEPROM, pointer na pole do kterého se zapisuje,
// a počet čtených bytů
//-----
unsigned char READ_EEPROM(unsigned char adresa_eeprom,unsigned char *data,
                           unsigned short int pocet_bytu);

//-----
// WRITE_EEPROM
// zápis do paměti EEPROM, vstupuje adresa v EEPROM, pointer na pole dat k zápisu do eeprom,
// a počet zapisovaných bytů
//-----
unsigned char WRITE_EEPROM(unsigned char adresa_eeprom,unsigned char *data,
                           unsigned char pocet_bytu);

//-----
// READ_RAM
// čtení z paměti RAM, vstupuje adresa v RAM, pointer na pole do kterého se zapisuje,
// a počet čtených bytů
//-----
unsigned char READ_RAM(unsigned short int adresa_ram,unsigned char *data,
                       unsigned short int pocet_bytu);

//-----
// WRITE_RAM
// zápis do paměti RAM, vstupuje adresa v RAM, pointer na pole dat k zápisu do RAM,
// a počet zapisovaných bytů
//-----
unsigned char WRITE_RAM(unsigned short adresa_ram,unsigned char *data,
                        unsigned char pocet_bytu);

//-----
// TestKomunikace
// vrací pouze číslo chyby
//-----
unsigned char TestKomunikace(void);

```

```
//-----
// Init_EEPROM
// načte data z EEPROM do aktivních registru (jako po resetu jednotky)
// vrácí pouze číslo chyby
//-----
unsigned char Init_EEPROM(void);

//-----
// GET_ADC_CH
// vstupuje pointer kam se zapíše získaná dekadická hodnota, číslo kanálu (0 až 7)
//-----
unsigned char GET_ADC_CH(unsigned short int *hodnota, unsigned char kanál);

//-----
// GET_DAC_ALL
// získá nastavení a hodnoty analog.výstupu
// vstupují pointer kam se zapisou získaná data
//-----
unsigned char GET_DAC_ALL(unsigned char *DAC_CONF0, unsigned char *DAC_CONF1,
                          unsigned short int *value0, unsigned short int *value1);

//-----
// GET_ADC_CONFIG
// zjistí konfigurační byte analog.vstupu
// vstupuje pointer kam se zapíše získaná hodnota
//-----
unsigned char GET_ADC_CONFIG(unsigned char *adc_config);

//-----
// GET_PORT
// zjistí stav digitálních vstupů, bin 1 = aktivní
// vstupuje pointer kam se zapíše získaná hodnota
//-----
unsigned char GET_PORT(unsigned char *hodnota);

//-----
// GET_FLOAT4_EEPROM
// načte z EEPROM datový typ float, který je uložený ve 3bytové velikosti a uloží ve 4bytovém formátu do pole hodnoty
// vstupuje počáteční adresa v EEPROM, pointer kam se zapisou získané hodnoty typu float, počet hodnoty typu float
//-----
unsigned char GET_FLOAT4_EEPROM(unsigned char adresa_eeprom, float *hodnoty,
                                unsigned char pocet_floatu);

//-----
// GET_ACTUAL_PSD_ALL
// zjistí vše o příslušném regulátoru
// vstupuje číslo reg (0 nebo 1), pointer kam se zapisou příslušné získané hodnoty
//-----
unsigned char GET_ACTUAL_PSD_ALL(unsigned char kanál, unsigned char *chardata4x,
                                unsigned short int *shortdata7x, float *fldata4x);

//-----
// SET_FLOAT4_EEPROM
// zapíše do EEPROM datový typ float ve formátu 3bytové velikosti (pocet_floatu max 4)
// vstupuje počáteční adresa v EEPROM, pointer na hodnoty typu float, počet hodnoty typu float
//-----
unsigned char SET_FLOAT4_EEPROM(unsigned char adresa_eeprom, float *hodnoty,
                                unsigned char pocet_floatu);

//-----
// SET_ADC_CONFIG
// nastaví konfigurační byte analog.vstupu
// vstupuje konfigurační byte
//-----
unsigned char SET_ADC_CONFIG(unsigned char adc_conf);

//-----
// SET_DAC_VALUE
// nastaví hodnotu analog.výstupu
// vstupuje číslo kanálu (0 nebo 1) a hodnota
//-----
unsigned char SET_DAC_VALUE(unsigned char kanál, unsigned short int value);
```

```

//-----
// SET_DAC_CONF
// nastaví konfigurační byte analog.vystupu
// vstupuje konfig byte, číslo kanálu se určí z obsahu config bytu
//-----
unsigned char SET_DAC_CONF(unsigned char config);

//-----
// SET_PORT
// nastaví digitální výstupy
// vstupuje hodnota, metoda 1-primo, 2-port AND hodnota 3-port OR hodnota
//-----
unsigned char SET_PORT(unsigned char hodnota,unsigned char metoda);

//-----
// SET_PSD_PAR
// nastaví parametry regulátoru primo
// vstupuje číslo reg. (0 nebo 1), pointer na pole 4x float obsahující koeficienty a,b,c,F
//-----
unsigned char SET_PSD_PAR(unsigned char kanal,float *fldata4x);          //a,b,c,F

//-----
// SET_PSD_PAR
// nastaví parametry regulátoru Kr,Ti,Td,F a přepočítá na hodnoty které lze přímo zapsat a zapíše je
// vstupuje číslo reg. (0 nebo 1), pointer na pole 4x float obsahující koeficienty Kr,Ti,Td,F
//-----
unsigned char SET_PSD_PAR2(unsigned char kanal,float *fldata4x);//Kr,Ti,Td,F

//-----
// SET_PSD_CONF
// nastaví konfiguraci regulátoru (přirazení kanálu časování apod.)
// vstupuje číslo reg. (0 nebo 1), pointer na pole 4x char,pointer na pole 6x unsigned short int,
//-----
unsigned char SET_PSD_CONF(unsigned char kanal,unsigned char *chardata4x, unsigned short int *shortdata6x);

//-----
// SET_PSD_V
// nastaví zadanou hodnotu
// vstupuje číslo reg. (0 nebo 1), pointer na zadanou hodnotu typu unsigned short int,
//-----
unsigned char SET_PSD_V(unsigned char kanal,unsigned short int *shortdata1x);

//-----
// DAC_REF
// nastaví do globálních proměnných typu float
// aktuální hodnotu velikosti 1bitu ve voltech V_BIT_DAC0, V_BIT_DAC1 pro anal.výstup
//-----
unsigned char DAC_REF(void);

//-----
// ADC_REF
// nastaví do globální proměnné typu float
// aktuální hodnotu velikosti 1bitu ve voltech V_BIT_ADC pro anal. vstup
//-----
unsigned char ADC_REF(void);

//-----
// DAC_CONVERT
// převede zadanou hodnotu ve voltech na celocíslnou hodnotu podle aktuálního nastavení
// vstupuje pointer na float hodnotu ve voltech která se zadá, a pointer na celocíslnou hodnotu která se získá
// funce upraví i hodnotu float která odpovídá celocíselné hodnotě value
//-----
unsigned char DAC_CONVERT(unsigned char kanal,float *hodnota,unsigned short *value);

//-----
// ADC_CONVERT
// převede zadanou hodnotu ve voltech na celocíslnou hodnotu podle aktuálního nastavení
// vstupuje pointer na float hodnotu ve voltech která se zadá, a pointer na celocíslnou hodnotu která se získá
// funce upraví i hodnotu float která odpovídá celocíselné hodnotě value
// - slouží pro zadávání zadaných hodnot regulátoru
//-----
unsigned char ADC_CONVERT(float *hodnota,unsigned short *value);

```

Místopřísežné prohlášení

„Místopřísežně prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s použitím uvedené literatury.“

V Liberci dne 22.5.2008

.....
Jiří Jindrák