

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE



KATEDRA INFORMATIZÁCIE A RIADENIA PROCESOV

NÁVRH ELEKTRONICKÉHO MODELU
NELINEÁRNEHO PROCESU

Dňa: 20.5.2005

Miesto: Bratislava

Vedúci diplomovej práce: Prof. Ing. Ján Mikleš, DrSc.

Vypracoval: Bc. Marek Gese

Pod'akovanie:

Týmto si dovoľujem poďakovať vedúcemu diplomovej práce Prof. Ing. Jánovi Miklešovi, DrSc. za všestrannú pomoc, ochotu a odborné vedenie a Doc. Ing. Jánovi Dankovi, CSc. za rady z oblasti elektroniky. Zároveň by som chcel poďakovať svojim rodičom.

Abstrakt

Diplomová práca sa zaoberá návrhom a konštrukciou elektronického zariadenia, ktoré na základe matematického modelu objektu tento objekt simuluje. Interakcie skutočného objektu s okolím nahrádzajú elektrické analógové vstupy a výstupy. Súčasťou je návrh programu pre zariadenie simulujúci vzorový chemický prietokový reaktor.

Abstract

This diploma work deals with the design and the construction of the electronic device, which simulates the process based on its mathematical model. The interaction of the real object with surround replaces electronic analog inputs and outputs. Part of the diploma work is also design of the program for a device simulating representative chemical flow reactor.

Obsah

Zoznam obrázkov	7
Úvod	8
1. Prietokový reaktor s miešaním	9
1.1 Popis prietokového reaktora s miešaním	9
1.2 Matematický model chemického reaktora	10
1.3 Vzorový model reaktora určený na simuláciu	13
2. Návrh zariadenia	15
2.1 Požiadavky na parametre zariadenia	15
2.2 Rozbor možností na zostrojenia zariadenia	15
3. Konštrukčná časť	17
3.1 Mikrokontroléry Atmel AVR	17
3.1.1 Pamäťový priestor mikrokontrolérov	18
3.1.2 Paralelné porty	19
3.1.3 Zabudované AD prevodníky	20
3.2 Popis častí zariadenia	21
3.2.1 Prevodník DA	22
3.2.2 Prevodník prúd napätie	23
3.2.3 Prevodník napätie prúd	24
3.2.4 Obvod referenčného napätia pre AD prevodníky	25
3.2.5 Predný panel	26
3.2.6 Riadenie a budenie LED displejov	28
3.2.7 Schéma zariadenia	29
3.2.8 Zoznam súčiastok	32
4. Programovanie mikrokontrolérov	17
4.1 Programovací jazyk basic FastAVR	18
4.2 Nahrávanie programu do mikrokontrolérov	19

5. Overenie činnosti zariadenia	37
5.1 Statická charakteristika prevodníkov AD	37
5.2 Statická charakteristika prevodníkov DA	38
5.3 Simulácia chemického prietokového reaktora v zariadení	39
Záver	41
Použitá literatúra	42
Príloha	43
Výpis programu pre ATmega16	43
Výpis programu ATmega8	51

Zoznam obrázkov

1.1	Schéma prietokového reaktora s duplikátorovým plášťom a ohrevným chladiacim hadom	10
3.1	Zjednodušený výklad AVR architektúry	17
3.2	Rozdelenie vnútorného pamäťového priestoru	18
3.3	Stavy vývodu pinu mikrokontroléru pri DA prevode	21
3.4	Schéma prevodníka prúdu na napätie	23
3.5	Schéma prevodníka napätia na prúd	24
3.6	Schéma referenčného obvodu napätia s použitím TL431	25
3.7	Schéma zapojenia zobrazovacích prvkov a klávesnice	27
3.8	Plošný spoj pre zobrazovacie prvky a klávesnicu	27
3.9	Usporiadanie segmentov v číslicovke	28
3.10	Schéma zariadenia	30
3.11	Plošný spoj zariadenia	31
4.1	Prostredie kompilátora FastAVR	34
4.2	Prostredie programu PonyProg2000	35
5.1	Statická charakteristika vstupných kanálov	37
5.2	Statická charakteristika výstupných prúdových smyčiek	38
5.3	Porovnanie odozvy zariadenia so simuláciou získanou v Matlabe - sledovanie koncentrácie -	39
5.4	Porovnanie odozvy zariadenia so simuláciou získanou v Matlabe - sledovanie koncentrácie -	40

Úvod

V posledných rokoch sa neoddeliteľnou súčasťou výučby stáva používanie moderných technológií, čo vedie k rapídnemu rozvoju elektronických vyučovacích pomôcok. Študenti navrhujú regulátory na počítačoch, riadený objekt je simulovaný jeho matematickým modelom, overenie toho či navrhnutý regulátor spĺňa zadané požiadavky je už len otázkou času za ktorý počítač vykoná výpočty. Študent tak rýchlo získava predstavu o tom ako sa bude riadený systém správať s daným regulátorom. V praxi však riadený systém predstavuje skutočný objekt, ktorý treba k riadiacemu systému pripojiť cez jeho vstupno-výstupné rozhranie. Predkladaná diplomová práca si kladie za cieľ vytvorenie elektronického prístroja, ktoré dokáže simulovať rôzne zariadenia pripojené k riadiacemu systému a tak umožní overenie správneho nastavenia vstupno-výstupného rozhrania riadiaceho systému.

1 Prietokový reaktor s miešaním

1.1 Popis prietokového reaktora s miešaním

Prietokové reaktory sa uplatňujú vo vysokotonážnych výrobách, najmä pre ich :

- nižšie prevádzkové náklady na jednotku množstva vyrobeného produktu ako pri vsádzkových reaktoroch,
- menšie nároky na obsluhu v porovnaní so vsádzkovými reaktormi, keďže sa neopakujú operácie spojené s naplňaním, vyprázdňovaním a čistením reaktora (najmä pri veľmi rýchlych reakciách),
- možnosť automatickej regulácie,
- väčšiu stabilitu reakčných podmienok, a tým aj stabilnejšiu kvalitu produktu.

Schéma tohto reaktora je na obr.1.1. V laboratóriu sa čoraz častejšie využíva pri štúdiu kinetiky reakcií najmä pre tieto výhody:

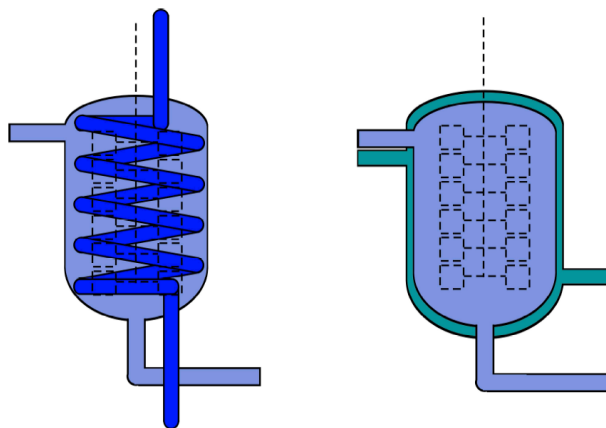
- konštrukčná jednoduchosť
- jednoduché meranie a regulácia vstupných a výstupných prúdov
- ak reaktor pracuje izotermicky v ustálenom stave, zloženie reakčnej zmesi a tým aj reakčná rýchlosť pri intenzívnom miešaní sa počas reakcie nemení, čo umožňuje veľmi jednoducho určiť reakčnú rýchlosť (Ilavský a kol., 1990).

1.2 Matematický model chemického reaktora

Najjednoduchším modelom prietokového reaktora s miešaním je model pri predpoklade dokonalého miešania. Z tohoto predpokladu vyplýva, že:

- Zloženia a teplota zmesi v celom objeme reaktora sú rovnaké.
- Zloženie a teplota zmesi na výstupe z reaktora sú také isté ako v reaktore.

Z týchto predpokladov vyplýva, že rýchlosť reakcie je v celom objeme reakčnej zmesi rovnaká, a teda bilanciu možno vzťahovať na celý reaktor.



Obr. 1.1 Schéma prietokového reaktora s duplikátorovým plášťom a ohrevným chladiacim hadom

Zavedenie symbolov pre fyzikálno-chemické veličiny:

a	-koeficient $\text{J} \cdot \text{min}^{-1} \cdot \text{K}$
A	-teplovýmenná plocha výmenníka tepla m^2
b	-koeficient
c_A	-molárna koncentrácia látky A na výstupe z reaktora $\text{kmol} \cdot \text{m}^{-3}$
c_{Af}	-molárna koncentrácia látky A na vstupe do reaktora $\text{kmol} \cdot \text{m}^{-3}$
c_p	-špecifická tepelná kapacita reakčnej zmesi $\text{J} \cdot \text{g}^{-1} \cdot \text{K}^{-1}$
c_{pc}	-špecifická tepelná kapacita chladiaceho média $\text{J} \cdot \text{g}^{-1} \cdot \text{K}^{-1}$
E	-aktivačná energia $\text{J} \cdot \text{kmol}^{-1}$
ΔH	-reakčná entalpia $\text{J} \cdot \text{kmol}^{-1}$
k_0	-predexponenciálny faktor min^{-1}
n_A	-tok látkového množstva látky A z reaktora kmol
n_{Af}	-tok látkového množstva látky A do reaktora kmol
q	-objemový prietok reakčnej zmesi $\text{m}^3 \cdot \text{min}^{-1}$
q_f	-vstupný objemový prietok reaktantov $\text{m}^3 \cdot \text{min}^{-1}$
q_{ch}	-objemový prietok chladiaceho média $\text{m}^3 \cdot \text{min}^{-1}$
r	-reakčná rýchlosť $\text{kmol} \cdot \text{min}^{-1} \cdot \text{m}^{-3}$
R	-univerzálna plynová konštanta $\text{J} \cdot \text{kmol}^{-1} \cdot \text{K}^{-1}$
t	-čas min
V	-reakčný objem reaktora m^3
α	-úhrnný koeficient prechodu tepla $\text{J} \cdot \text{min}^{-1} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$
ϑ	-teplota reakčnej zmesi K
ϑ_{ch}	-teplota chladiaceho média K
ϑ_{chf}	-teplota chladiaceho média na vstupe K
ρ	-stredná hodnota hustoty reakčnej zmesi $\text{g} \cdot \text{m}^{-3}$
ρ_c	-stredná hodnota hustoty chladiaceho média $\text{g} \cdot \text{m}^{-3}$
v_A	-stechiometrický koeficient látky A

Materiálova bilancia vychádza zo zákona zachovania hmotnosti, ktorý tvrdí, že súčet hmotností všetkých látkových tokov do systému je rovný súčtu hmotností všetkých látkových tokov vystupujúcich zo systému a hmotností akumulovaných častí tokov prúdov v systéme. Po prevode hmotností na látkové množstvá dostávame nasledovnú rovnicu:

$$\dot{n}_{Af} dt + v_A \dot{\xi}_V dt = \dot{n}_A dt + d(Vc_A) \quad (1.1)$$

$$q_f c_{Af} + v_A \dot{\xi}_V V = q c_A + \frac{d}{dt}(Vc_A) \quad (1.2)$$

Riešenie rovnice (1.2) závisí od tvaru rýchlostnej rovnice, zmeny hustoty a objemovej rýchlosti na výstupe a vstupe do reaktora.

Energetická bilancia je matematická interpretácia všeobecne platného princípu zachovania energie. Pri bilancovaní chemických reaktorov sú členy reprezentujúce mechanickú energiu v porovnaní s ostatnými členmi zanedbateľné a z bilancie sa vypúšťajú. Pri všetkých typoch reaktorov okrem elektrochemických sa energia vymieňa s okolím iba vo forme tepla a objemovej práce. Pre izobarické systémy energetická bilancia potom prechádza na entalpickú bilanciu, ktorej slovné vyjadrenie je:

entalpia		teplo		entalpia		zmena
privedených	+	privedené	=	odvedených	+	entalpie
látok		z okolia		látok		systému

Druhý člen pravej časti rovnice zahŕňa teplo privedené do objemového elementu inak ako prúdením. Prakticky ide o teplo vymenené s okolím cez steny reaktora, alebo v zabudovanom výmenníku tepla. Člen je kladný, ak sa teplo z okolia privádza, záporný, ak sa teplo odvádza.

Posledný člen na pravej strane rovnice je akumulčný člen a je kladný, ak sa entalpia systému zväčšuje, záporný, ak sa znižuje.

Rovnako ako pri odvodzovaní látkovej bilancie, aj pri zostavovaní entalpickej bilancie

je bilančným systémom celý obsah reaktora. Pri odvodzovaní matematického modelu zanedbáme tepelnú kapacitu steny reaktora, ktorá ohraničuje reakčnú zmes od chladiacej kvapaliny a jej tepelný odpor. Predpokladáme konštantnú hustotu a špecifickú tepelnú kapacitu reakčnej zmesi. Ďalej predpokladáme konštantný objem reakčnej zmesi v reaktore a konštantný úhrnný koeficient prechodu tepla. Objemový prietok reakčnej zmesi na vstupe do reaktora a na výstupe z reaktora sú rovnaké a predpokladáme ich konštantné.

$$V\rho c_p \frac{d\vartheta}{dt} = q\rho c_p \vartheta_f - q\rho c_p \vartheta - \alpha F(\vartheta - \vartheta_{ch}) + V(-\Delta H)r(c_p \vartheta) \quad (1.3)$$

Tretia rovnica, popisujúca reaktor je entalpická bilancia chladiaceho média

$$\frac{d\vartheta_{ch}}{dt} = \frac{q_{ch}}{V_{ch}}(\vartheta_{chv} - \vartheta_{ch}) + \frac{A\alpha}{c_{pch}V_{ch}\rho_{ch}}(\vartheta - \vartheta_{ch}) \quad (1.4)$$

(Ilavský a kol., 1990)

1.3 Vzorový model reaktora určený na simuláciu

Veľmi často sa však na popis reaktora používa model pozostávajúci z dvoch rovníc. Takisto v rovnici (1.3) sa zvykne člen vyjadrujúci prestup tepla z chladiaceho média rôzne modifikovať a nahrádzať rôznymi polynómami a pribúda aj člen charakterizujúci odvod tepla do okolia. Uvediem prípad kde je modifikovaný člen charakterizujúci výmenu tepla medzi reakčnou zmesou a chladiacim médiom pre reakciu prvého poriadku $A \rightarrow B$

$$V \frac{dc_A}{dt} = q(c_{Af} - c_A) - V k_0 e^{\frac{E}{RT}} c_A$$

$$V\rho c_p \frac{d\vartheta}{dt} = q\rho c_p (\vartheta_f - \vartheta) - \frac{aq_c^{b+1}}{q_c + \frac{aq_c}{2\rho_c c_{pc}}} (\vartheta - \vartheta_{cf}) + (-\Delta H)V k_0 e^{\frac{E}{RT}} c_A$$

V tabulke 1.1 sú uvedené parametre reaktora a chemickej reakcie s ktorými sa bude overovať zostrojené zariadenie so simuláciou získanou výpočtom.

(Marlin, 1995)

Tab.1.1 Parametre reaktora a reakcie

V	1	m^3
c_{A0}	2.0	kmol.m^{-3}
ϑ_f	323	K
c_p	4.1868	$\text{J.g}^{-1}.\text{K}^{-1}$
ρ	10^6	g.m^{-3}
k_0	10^{10}	min^{-1}
E/R	8330.1	K
$-\Delta H$	544.284×10^6	J.kmol^{-1}
ϑ_{cf}	365	K
q_c	15	$\text{m}^3.\text{min}^{-1}$
c_{pc}	4.1868	$\text{J.g}^{-1}.\text{K}^{-1}$
ρ_c	10^6	g.m^{-3}
a	7.02545×10^6	$\text{J.min}^{-1}.\text{K}^{-1}$
b	0.5	

2 Návrh zariadenia

2.1 Požiadavky na parametre zariadenia

Požiadavky na zariadenie sú:

1. Zariadenie musí dávať výsledné hodnoty prúdov a stavových veličín v reálnom čase
2. Vstupnými veličinami sú:
 - vstupná koncentrácia reaktantu A
 - teplota vstupujúcej reakčnej zmesi
 - prietok chladiaceho média
 - teplota chladiaceho média
 - relatívna chyba prevodu vstupných a výstupných veličín na elektrické menšia ako 2 %.
3. Výstupnými veličinami sú:
 - výstupná koncentrácia reaktantu A
 - teplota vystupujúcej reakčnej zmesi
4. Možnosť zmeny parametrov, prípadne matematického modelu
5. Zobrazovanie hodnôt vstupných a výstupných veličín priamo na zariadení

2.2 Rozbor možností na zostrojenie zariadenia

Kým ľubovoľný lineárny systém je možné previesť na ekvivalentný elektrický obvod s použitím kondenzátorov, odporov a operačných zosilňovačov, pri nelineárnych systémoch je pri požiadavke veľkej presnosti a možnosti zmeny parametrov jedinou možnosťou na vytvorenie elektrického modelu systému využitie mikroprocesorovej techniky. Najjednoduchším riešením by bolo použitie osobného počítača so zabudovanou AD/DA kartou na komunikáciu s okolím a naprogramovať model. Inou možnosťou je použitie jednočipových mikrokontrolérov. Nevýhody riešenia s PC je vo veľká spotreba elektrickej energie a veľké rozmery zariadenia. Ak by sa mal eliminovať monitor a použiť LED displeje na zobrazovanie údajov, treba navrhnuť a

vyrobiť elektroniku pre ich budenie, rovnako ako pri použití mikrokontrolérov. Výhodou riešenia s PC je veľký výkon procesora, existencia matematického kooprocesora a veľká univerzálnosť. Najdôležitejšou otázkou pri rozhodovaní či použiť PC alebo mikrokontrolér je otázka či mikrokontroléry dokážu v reálnom čase riešiť zadaný systém diferenciálnych rovníc. Na trhu je veľký počet rôznych druhov mikrokontrolérov. Jednými z najrozšírenejších sú 8 bitové mikrokontroléry PIC firmy Microchip a mikrokontroléry s jadrom 51 a mikrokontroléry AVR. Použitie 16 bitových mikrokontrolérov pre ich vysokú cenu som zamietol. Procesory AVR ATMEGA majú v porovnaní s procesormi PIC výkonnejšiu sadu inštrukcií pre matematické operácie, tak som sa rozhodol pre použitie AVR ATMEGA.

Zadaný systém diferenciálnych rovníc som naprogramoval do procesoru AVR ATMEGA8 taktovaného na frekvencii 16 MHz v programovacom jazyku Basic FastAVR.

Bola simulovaná zmena prietoku reakčnej zmesi z hodnoty 0 na hodnotu $1 \text{ m}^3 \cdot \text{min}^{-1}$.

Pri pevnom časovom kroku 0.01 sekundy bola chyba v jednotlivých bodoch simulácie najviac 0,12 % v porovnaní so simuláciou robenou na PC s krokom $1\text{E}-5$ sekundy. Simulácia pre 10 minút reakčného času trvala v skutočnosti 5 sekúnd. Pri teste rýchlosti bolo vypnuté zobrazovanie vypočítaných výsledkov na pripojenom LCD module. Mikrokontrolér dokáže simulovať s dostatočnou presnosťou zadaný chemický reaktor približne 100 krát rýchlejšie ako prebieha dej v skutočnosti. Jeho výkon je plne postačujúci a z dôvodu vyrobiť malé kompaktné zariadenie som sa rozhodol pre stavbu zariadenia s použitím mikrokontroléra miesto PC.

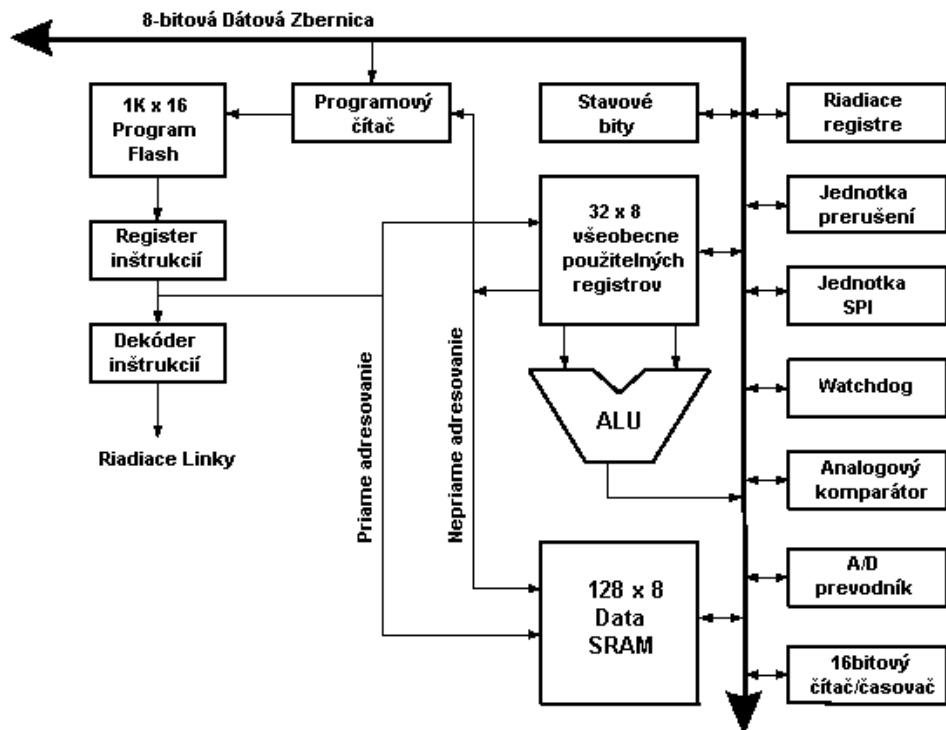
Vzhľadom k tomu že zobrazovanie vypočítaných hodnôt na LED displejoch si vyžaduje veľa strojového času, rozhodol som sa pre použitie dvoch mikrokontrolérov. Úlohou prvého je načítavanie vstupných hodnôt z AD prevodníkov a výpočet výstupných veličín. Druhý mikrokontrolér zobrazuje vstupné a výstupné veličiny na LED a LCD displejoch a zabezpečuje prevod výstupných veličín na elektrický signál v DA prevodníkoch. Mikrokontroléry medzi sebou komunikujú UART rozhraním.

3. Konštrukčná časť

3.1 Mikrokontroléry Atmel AVR

V tejto časti sú uvedené základné informácie o mikrokontroléroch AVR a podrobnejšie informácie, ktorých znalosť je nevyhnutná k vysvetleniu činnosti zariadenia a jeho programovaniu.

Mikrokontroléry ATMEL AVR sú nízkopríkonové 8 bitové mikrokontroléry založené na AVR RISC architektúre. Pretože vykonávajú inštrukcie v jedinom hodinovom cykle, dosahujú tieto mikrokontroléry výpočtový výkon 1 MIPS na 1 MHz. Mikrokontroléry s označením ATMEGA patria tiež do triedy AVR, ale majú rozšírený inštrukčný súbor a nové integrované rozhrania.



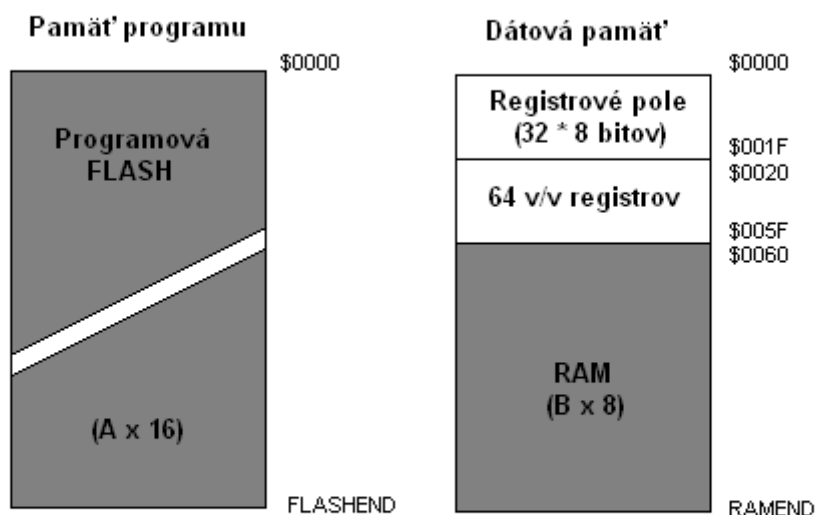
Obr. 3.1 Zjednodušený výklad AVR architektúry

3.1.1 Pamäťový priestor mikrokontrolérov

Mikrokontroléry rady AVR používajú harvardskú architektúru, ktorá sa vyznačuje oddeleným pamäťovým priestorom pre dáta a program. Zjednodušená schéma architektúry AVR je na obrázku 3.1.

Vnútrotná programová pamäť

Pamäť programu je tvorená zabudovanou FLASH pamäťou, ktorej kapacita sa pohybuje od 1 KB do 256 KB. Táto pamäť je z hľadiska prebiehajúceho programu adresovaná po slovách. Veľkou výhodou procesorov AVR je možnosť programovať ich priamo v zapojení, ISP (In System Programming), cez rozhranie SPI (Serial programming Interface).



Obr.3.2 Rozdelenie vnútorného pamäťového priestoru

RAM

Dátová pamäť RAM mikrokontrolérov AVR sa pohybuje v rozsahu od 0 B do 8 KB. Mimo túto pamäť každý procesor obsahuje 32 8-bitových všeobecne použiteľných registrov, ktoré u jednoduchších typov bez RAM ju v obmedzenej miere nahrádzajú.

Vnútná EEPROM

Okrem vyššie uvedenej dátovej a programovej pamäte majú všetky mikrokontroléry AVR zabudovanú dátovú EEPROM veľkosti 64 B až 4 KB. Táto pamäť si udržiava svoj obsah aj pri vypnutí napájania, pričom je ju možné programovať zvonku programátorom, alebo do nej môže zapisovať/čítať dáta bežiaci program v mikrokontroléri. Nehodí sa na časté ukladanie a čítanie dát pre je pomalosť a obmedzený počet zápisov. Výrobca udáva počet preprogramovaní na 100000 krát.

3.1.2 Paralelné porty

Paralelným portom rozumieme skupinu vývodov, ktorá je z hľadiska mikrokontroléru chápaná ako celok. Zvyčajne sa jedná o 8 vývodov.

Všetky porty mikrokontrolérov AVR majú rovnaké základné vlastnosti, môžu sa vzájomne odlišovať napojením rôznych zabudovaných periférií.

Každý vývod portu môže meniť svoj smer nezávisle na ostatných vývodoch, môže sa teda jednať o vstup alebo výstup. Porty sa ovládajú trojicou vstupno/výstupných registrov.

Mikrokontroléry AVR môžu disponovať až štvoricou portov, procesory ATmega až šiestimi portami, pomenovanými PA, PB, PC, PD, PE, PF.

Porty sú ovládané trojicou registrov:

- Register DDRX určuje smer toku dát
- Register PORTX je dátovým registrom portu, jeho obsah zodpovedá hodnote zapísanej do vyrovnávacieho registra portu

Register PINX je určený len na čítanie a obsahuje hodnotu prečítanú z jednotlivých vývodov.

Písmeno X v názvoch registrov označuje názov portu A, B, C... .

Registre DDRX a PORTX určujú jednak smer vývodov, ale tiež ich špecifické vlastnosti uvedené v tabuľke XR. Základné správanie vývodov je zhrnuté v tabuľke 3.1.

Tab. 3.1

DDBXn	PORTXn	vstup/výstup	pull-up	poznámka
0	0	vstup	odpojený	trojstavový vstup
0	1	vstup	pripojený	vstup v úrovni log. 1
1	0	výstup	odpojený	log. 0
1	1	výstup	odpojený	log. 1

3.1.3 Zabudované AD prevodníky:

Vačšina mikrokontrolérov rady AVR má zabudovaný 10 bitový AD prevodník pracujúci algoritmom postupnej aproximácie. Tento algoritmus má výhodu v pomernej veľkej rýchlosti, (15000 až 150000 vzoriek za sekundu), má však malú odolnosť voči šumu na vstupe. Táto nevýhoda je čiastočne odstránená interným vzorkovačom, ktorý udržiava merané napätie počas merania na konštantnej úrovni. Vzhľadom na veľkú rýchlosť merania je možné šum potlačiť tak, že sa použije aritmetický priemer z väčšieho počtu po sebe idúcich meraní. Šum generovaný činnosťou jadra mikrokontroléru je možné znížiť zastavením jadra počas prevodu, ukončený prevod generuje prerušenie ktoré prepne jadro mikrokontroléru z režimu spánku do pracovného módu.

Vstup AD prevodníku je pripojený na 6-8 kanálový multiplexer, takže je možné snímať 6 až osem vstupov, nie však súčasne.

Rozsah spracovávaného napätia je daný napätím na vývodoch AGND a AREF.

AD prevodník môže pracovať v dvoch režimoch:

- ◆ Jednoduchý prevod – každý prevod musí byť vyvolaný užívateľom (programom)
- ◆ Volný beh – prevody sú periodicky opakované bez nutnosti opätovného programového spúšťania

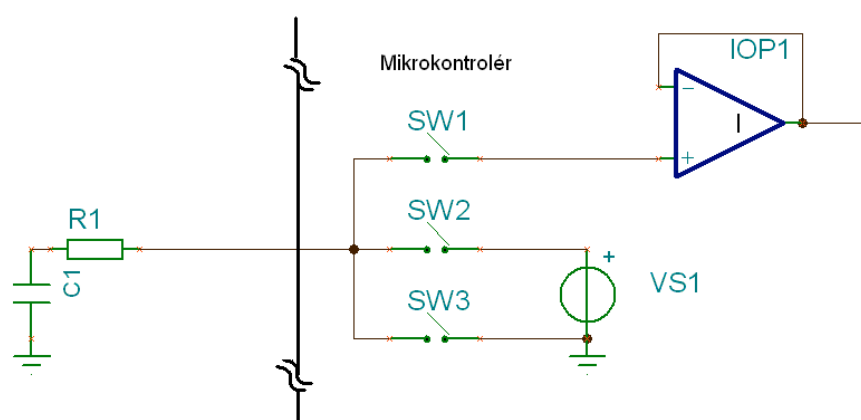
Ukončenie prevodu môže vyvolať prerušenie, takže počas samotného prevodu môže mikrokontrolér vykonávať inú činnosť, pri vzniku prerušenia vyzdvihne nameraný údaj, spustí nové meranie a pokračuje v predtým vykonávanej časti programu.

3.2 Popis častí zariadenia

3.2.1 DA prevodník

Mikrokontroléry AVR nemajú zabudované DA prevodníky. Na realizáciu DA prevodu je možné použiť externý prevodník riadený mikrokontrolérom, ale vzhľadom na cenu týchto prevodníkov som sa rozhodol použiť integrovaný AD prevodník v mikrokontroléri na realizáciu DA prevodu.

Pin mikrokontroléru na ktorý je pripojený vstup AD prevodníka je možné programovo nastaviť aj ako výstupný pin s požadovanou logickou úrovňou. Tak je možné nabíjať, respektíve vybíjať kondenzátor pripojeného RC člena. Po prepnutí pinu do režimu AD prevodníka a zmeraním napätia na RC člene sa vyhodnotí či nastavené napätie zodpovedá žiadanému, ak nie, vypočíta sa čas počas ktorého treba nabíjať kondenzátor aby dosiahol žiadané napätie. Pin sa následne prepne do režimu výstupu, nastaví na logickú 1 alebo 0, podľa toho či sa napätie na RC člene má zvyšovať alebo znižovať. Po uplynutí vypočítaného času sa overí hodnota prepnutím pinu do režimu AD prevodníka, v prípade rozdielu sa proces opakuje kým nie je na RC člene požadované napätie.



Obr. 3.3 Stavy vývodu pinu mikrokontroléru pri DA prevode

Konštantu RC člena treba voliť tak, aby sa počas najmenšieho možného časového úseku na ktorý možno prepnúť pin z režimu vysokej imedancie (vstup AD prevodníku) do režimu nabíjania/vybíjania RC člena bola zmena napätia na kondenzátore maximálne o napätie ktoré zodpovedá rozlišovacej schopnosti AD prevodníka.

Ak by bol kondenzátor napájaný zdrojom konštantného prúdu, prírastok napätia na kondenzátore by bol lineárnou funkciou času. Pri napájaní RC člena konštantným napätím je napätie na kondenzátore dané vzťahom (3.1) :

$$U_c(t) = U(1 - e^{-\frac{t}{RC}}) \quad U_c(0) = 0 \quad (3.1)$$

Napätie na kondenzátore ktorého napätie nebolo v čase 0 nulové možno zistiť podľa vzťahu (3.2):

$$U_c(t) = [U_R - U_c(0)](1 - e^{-\frac{t}{RC}}) + U_c(0) \quad (3.2)$$

Čas potrebný na dosiahnutie novej hodnoty napätia na kondenzátore sa vyjadrí z rovnice (3.3):

$$t = -\ln \left[\frac{U_R - U_c(t)}{U_R - U_c(0)} \right] RC \quad (3.3)$$

Výpočet času potrebného na požadovanú zmenu podľa rovnice (3.3) je pre mikrokontrolér časovo náročnou operáciou, preto je čas na ktorý sa má pin prepnúť do požadovanej úrovne nutné vypočítať čo najpresnejšie, aby skutočná hodnota konvergovala ku žiadanej, podľa algoritmu popísaného vyššie pri minimálnom počte opakovaní výpočtu.

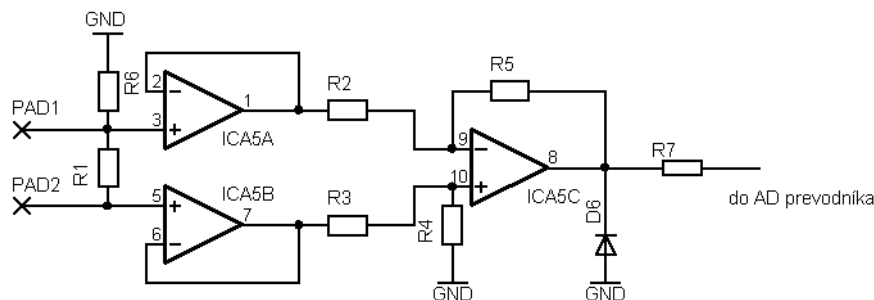
3.2.2 Prevodník prúd napätie

Pre prepojenie zariadenia s okolím sú použité prúdové smyčky, ktoré majú väčšiu odolnosť proti rušeniu oproti prenosu signálu napätím.

Jedným z najčastejšie používaných štandardov je prúdová smyčka 4-20 mA.

Ideálnym riešením by bolo galvanické oddelenie zariadenia od riadiaceho systému, ktoré by bolo možné realizovať použitím optočlenov, ale vzhľadom k veľkej nelinearite v prenose jednoduchých optočlenov by bolo potrebné použiť optočleny so sigma-delta moduláciou, ich cena je však vysoká.

Na prevod prúdu na napätie je použité zapojenie podľa obrázku 3.4.



Obr. 3.4 Schéma prevodníka prúdu na napätie

Na rezistore R1 s hodnotou 250 ohmov vzniká prechodom prúdu 4 –20 mA napätie 1-5 V, ktoré sa privádza cez dva sledovače OP1, OP2 s vysokou impedanciou na diferenciálny zosilovač tvorený operačným zosilňovačom OP3 a rezistormi R2 až R5, so zosilnením 1. Je dôležité aby rezistory R2, R3, R4, R5 mali rovnaký odpor. Boli zvlášť vybrané z väčšieho počtu rezistorov. Hodnota rezistoru R1 je kritická a treba ju presne dodržať. Vstup prevodníka je chránený pred záporným napätím zenerovou diódou, ktorá zároveň zabraňuje tomu, aby sa na vstup AD prevodníku dostalo napätie väčšie ako 6 V.

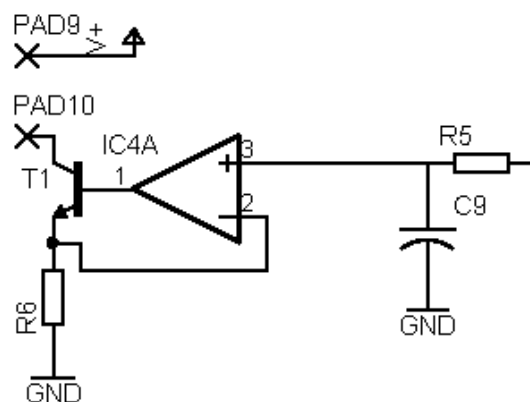
Vzhľadom na to že spodné referenčné napätie AD prevodníka na pine AGND je 0 V, napätie ktoré je na výstupe prevodníku prúd-napätie pri vstupnom prúde 4 mA je 1 V, nevyužíva sa celý merací rozsah AD prevodníka. Jednoduchým riešením by bolo priviesť na referenčný signál

AGND napätie 1 V, tým by však bolo znemožnené používať zariadenia so staršími systémami využívajúce prúdovú smyčku 0-20 mA. Napriek tomuto obmedzeniu je presnosť AD prevodníku dostatočná.

3.2.3 Prevodník napätie prúd

Je realizovaný zapojením na obrázku 3.5. Vstupné napätie sa privádza z RC členu ktorý je súčasťou DA prevodníka na neinverujúci vstup operačného zosilňovača. Na referenčnom odpore R6 vzniká prechodom prúdu napätie, ktoré je privádzané na invertujúci vstup. Napätie na výstupe operačného zosilňovača sa ustáli na hodnote pri ktorej prúd tečúci referenčným odporom vyvolá rovnaký úbytok napätia ako je napätie na neinverujúcom vstupe.

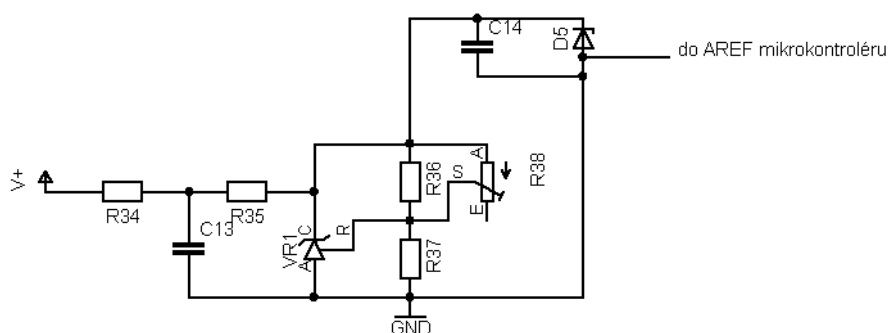
Použité tranzistory mali zosilňovací činiteľ väčší ako 300, aby prúd prechádzajúci cez bázu tranzistora do referenčného odporu bol zanedbateľný voči prúdu prechádzajúcemu prúdovou smyčkou.



Obr. 3.5 Schéma prevodníka napätia na prúd

3.2.4 Obvod referenčného napätia pre AD prevodníky

Referenčné napätie 5 V sa získava z precízneho napäťového regulátora TL431, ktorý je napájaný stabilizovanými 12 V. Presné napätie možno nastaviť viacotáčkovým trimrom R5. Nízkošumová Zenerova dióda D5 zabráňuje tomu, aby sa na vstup mikrokontroléra dostalo napätie väčšie ako 6V a chráni ho pred zničením vysokým napätím, ktoré by sa mohlo dostať na pin mikrokontroléra pri poruche činnosti regulačného obvodu.



Obr. 3.6 Schéma referenčného obvodu napätia s použitím TL431

Hodnoty rezistorov v odporovom deliči R36,R37 boli vypočítané tak, aby bol využitý celý rozsah trimru pre rozsah nastaviteľného referenčného napätia 2,5 – 5,6 V.

Teplotná závislosť referenčného napätia v intervale 20 - 50 °C je menej ako $50 \mu\text{V} \cdot \text{K}^{-1}$.

3.2.5 Predný panel

Schéma zapojenia samostatnej zobrazovacej časti zariadenia je na obrázku 3.7. V zapojení sú použité štvormiestne červené LED displeje HD-M514RD so spoločnou anódou. Výhodou použitých displejov je vzájomné prepojenie rovnakých segmentov jednotlivých číslíc vnútri púzdra displeja, čím sa zmenší počet potrebných vodivých ciest na plošnom spoji a to umožní použitie jednostranného plošného spoja.

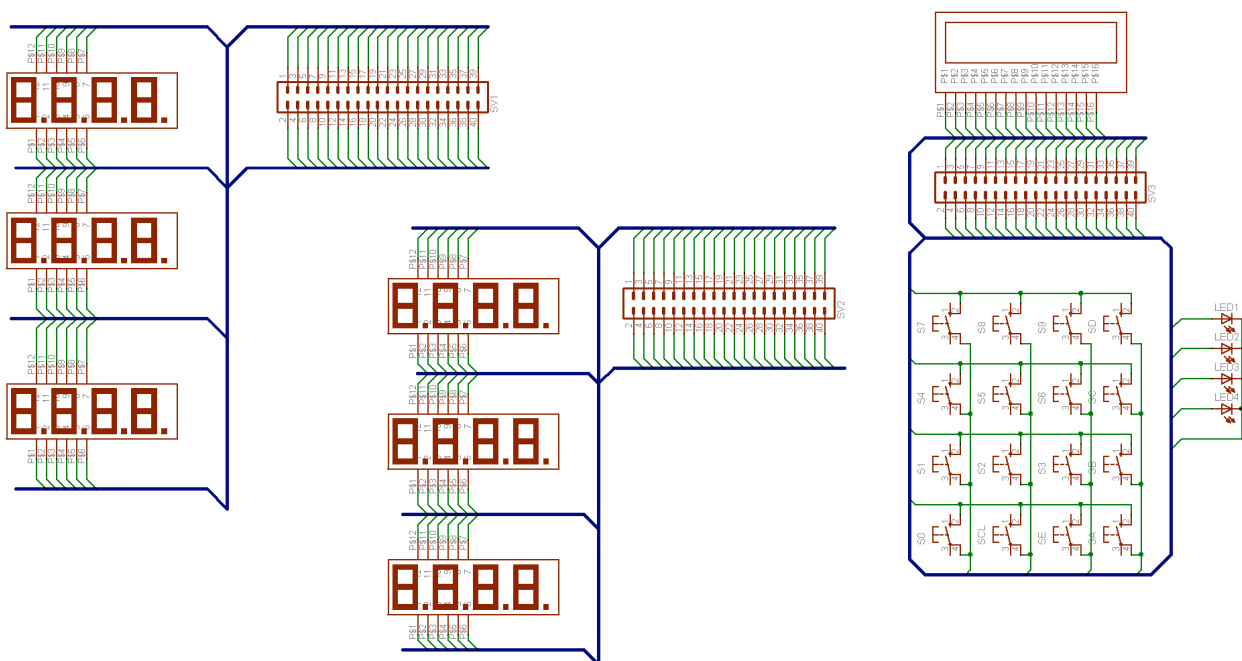
Maximálny trvalý prúd na jeden segment je 30 mA, v impulznom režime 150 mA, priemerný prúd pri impulznom režime môže byť najviac 30 mA.

Svietivosť jedného segmentu pri 10 mA je 2,5 mcd.

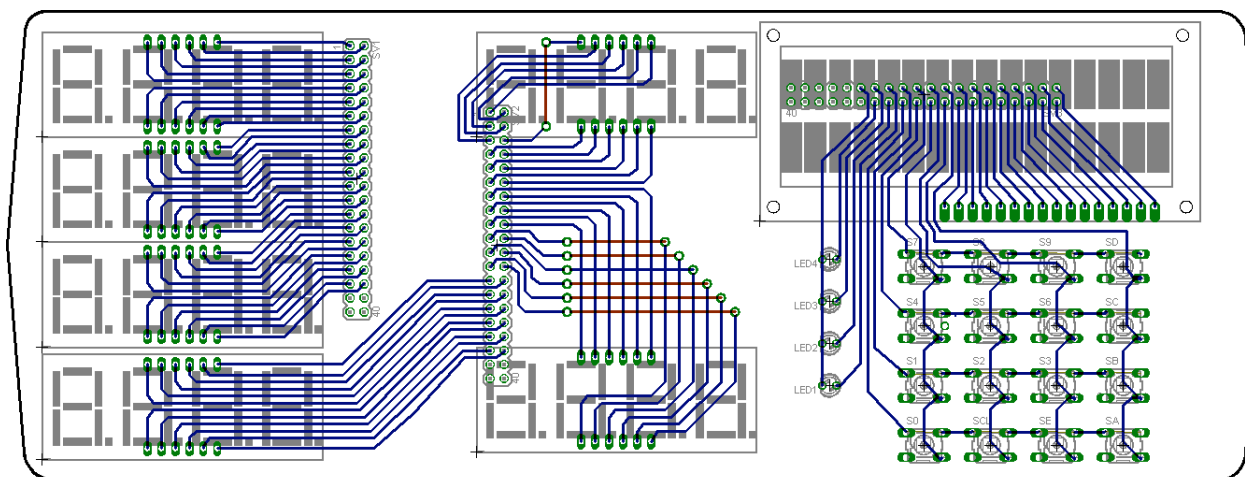
Použitý LCD modul obsahuje štandardný radič HD44780 podporovaný vyššími programovacími jazykmi. Z LCD modulu sú na prepojovací konektor vyvedené všetky kontakty, ale na strane dosky s mikrokontrolérmi sú použité iba vodiče pre štvorbitovú komunikáciu procesoru s LCD modulom z dôvodu šetrenia počtu pinov mikrokontroléru.

Mikrospínače sú zapojené do matice 4*4 a takto vzniknutá klávesnica je pripojená na zbernicu ôsmimi spojmami.

Plošný spoj predného panela zobrazený v zmenšenej veľkosti spolu s rozmiestnením súčiastok je na obrázku 3.8.



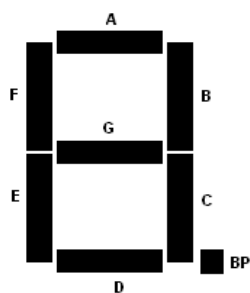
Obr. 3.7 Schéma zapojenia zobrazovacích prvkov a klávesnice



Obr. 3.8 Plošný spoj pre zobrazovacie prvky a klávesnicu

3.2.6 Riadenie a budenie LED displejov

Šestica štvormiestnych LED displejov je budená multiplexne. Na krátky časový úsek sú rozsvietené všetky číslicovky zodpovedajúce tisícim. Zhasnú sa a rozsvietia sa číslicovky zodpovedajúce stovkám. Pokračuje sa cez desiatkové k jednotkovým. Potom sa celý cyklus opakuje. Segменты sú budené šiestimi obvodymi 74HC573. V tabulke 3.2 sú kódy jednotlivých zobrazovaných čísiel a znakov, ktoré treba posielat' na port C procesoru ATmega16.



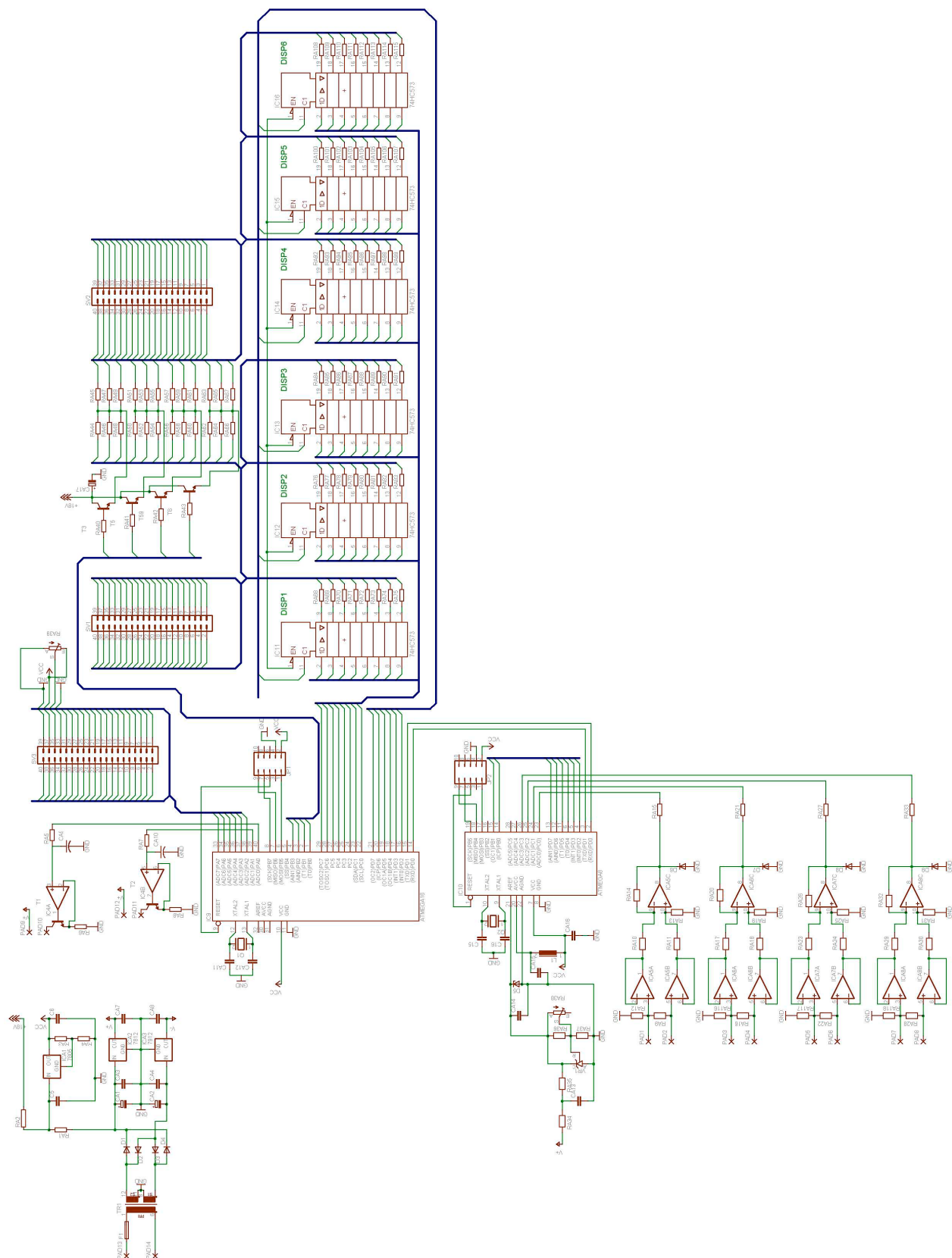
Obr. 3.9 Usporiadanie segmentov v číslicovke

Tab. 3.2 Ovládanie segmentov číslicovky a tvorba znakov

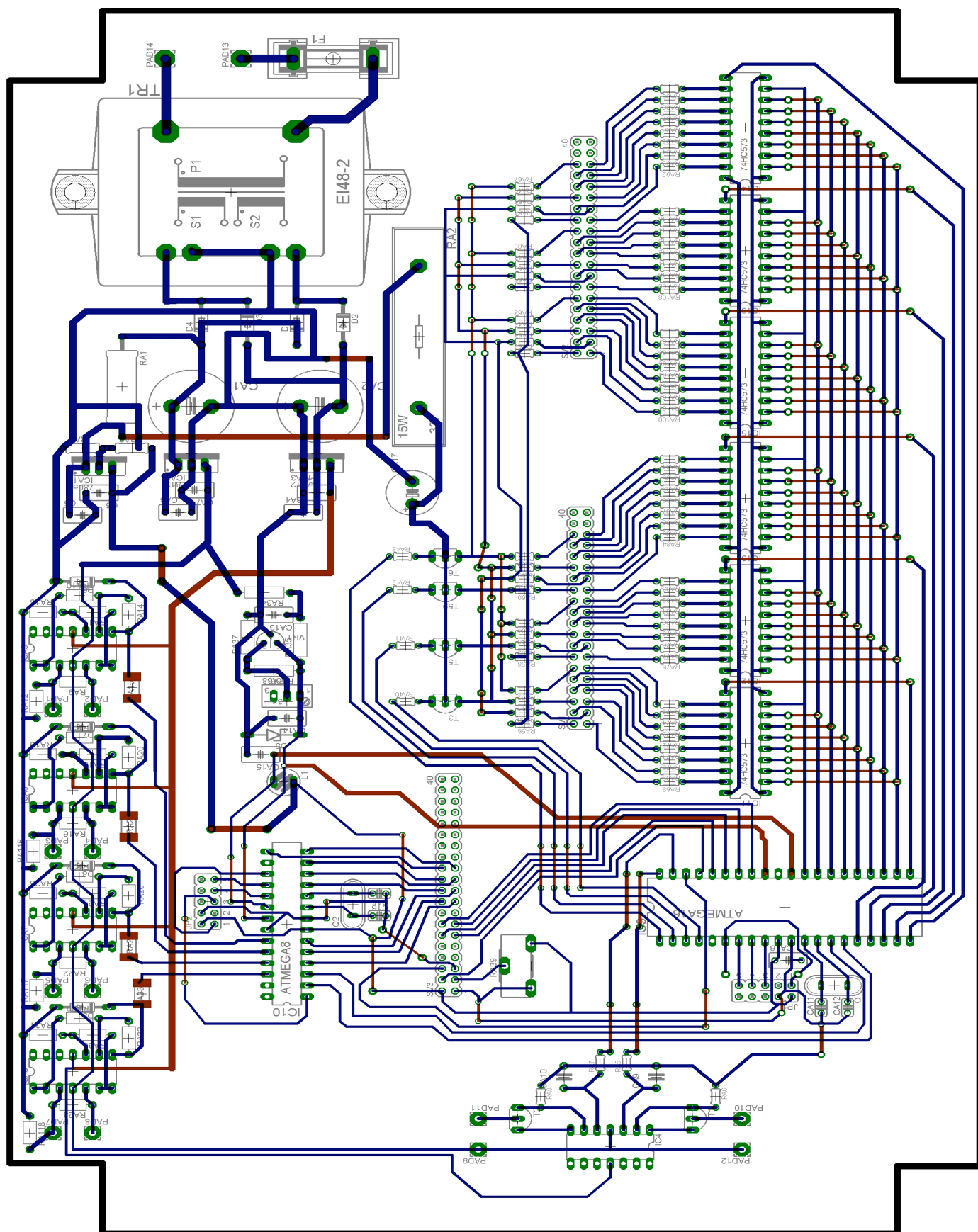
Pin	PC.7	PC.6	PC.5	PC.4	PC.3	PC.2	PC.1	PC.0	DEC	HEX
Znak	G	C	BP	D	E	A	F	B		
0	1	0	1	0	0	0	0	0	160	A0
1	1	0	1	1	1	1	1	0	190	BE
2	0	1	1	0	0	0	1	0	98	62
3	0	0	1	0	1	0	1	0	42	2A
4	0	0	1	1	1	1	0	0	60	3C
5	0	0	1	0	1	0	0	1	41	29
6	0	0	1	0	0	0	0	1	33	21
7	1	0	1	1	1	0	1	0	186	BA
8	0	0	1	0	0	0	0	0	32	20
9	0	0	1	0	1	0	0	0	40	28
E	0	1	1	0	0	0	0	1	97	61
r	0	1	1	1	0	1	1	1	119	77

3.2.7 Schéma zariadenia

Na obrázku 3.10 je schéma zapojenia bez bloku zobrazovacích prvkov a klávesnice, ktorý sa pripája k zariadeniu pomocou troch skrátených ATA káblov používaných k pripojeniu hardiskov v PC. Schéma a plošný spoj je navrhnutý v programe Eagle v4.13, súbory sú uložené na doprovodnom CD.



Obr. 3.10 Schéma zariadenia



Obr. 3.10 Schéma zariadenia

3.2.8 Zoznam súčiastok:

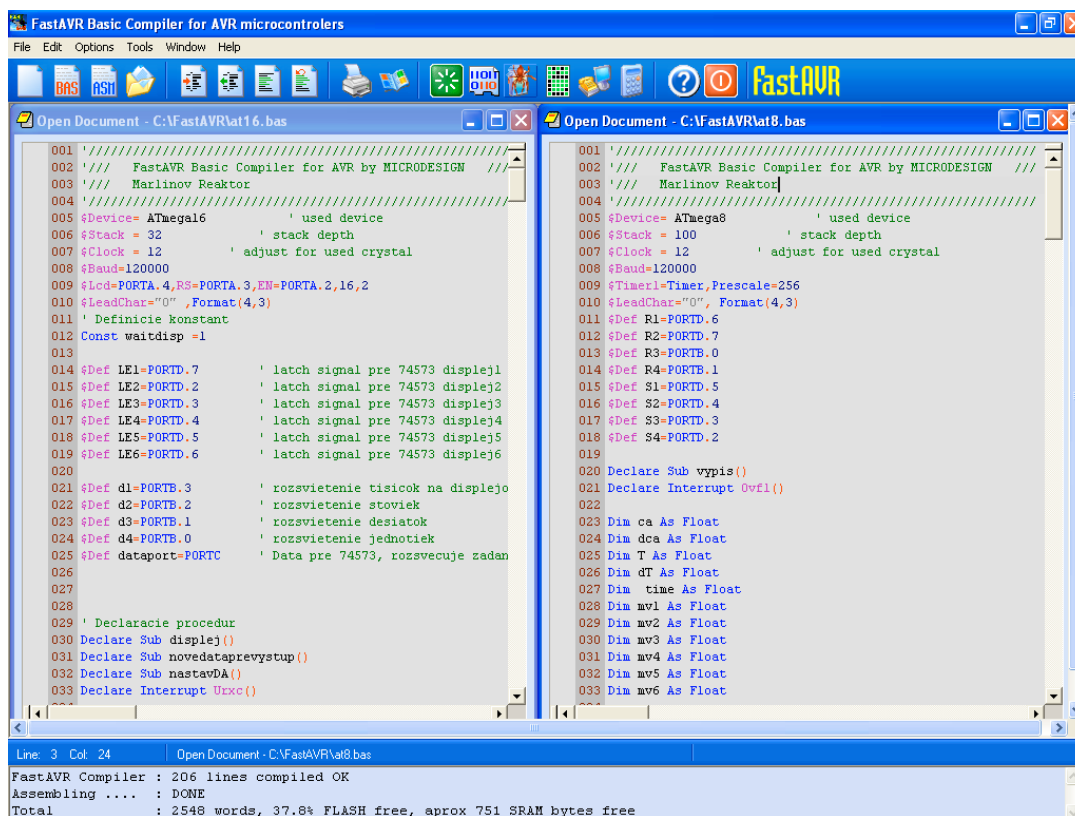
R1	39 Ω 5W
R2	47 Ω 15W
R3	120 Ω
R4	22 Ω
R5,R7	4k7 Ω
R6,R8, R9,R16,R22,R28	250 Ω
R10,R11,R13,R14,R17,R18,R19,R20, R23,R24,R25,R26,R29,R30,R31,R32	8k2
R15,R21,R27,R33	4k7 SMD
R34, R35	330 Ω
R36	620 Ω
R37	470 Ω
R38	10k viacotáčkový trimer
R39	1k trimer
R40,R41,R42,R43	22 Ω
R44...R67	0 Ω
R68...R115	33 Ω
R12,R116,R,R117,R118	47k
C1,C2	3300uF/35 V
C3,C5,C7,C8	100nF
C4	2,2 uF/50V
C8	1 uF/50V
C9,C10	330nF
C11,C12,C18,C19	22 pF
C13,C14,C15,C16	100 nF
C17	470uF/25V
Q1,Q2	kryštál 12 MHz
D1...D4	1N4007
D5,D6,D7,D8,D9	BZX83V005.6

VR1	TL431
T1...T6	2SC2500
IC1	L7805
IC2	L7812
IC3	L7912
IC4...IC8	LM324N
IC9	mikrokontrolér ATMEGA16
IC10	mikrokontrolér ATMEGA8
IC11...IC16	M74HC573
SV1...SV6	lámací konektor S2G40
DISP1...DISP6	M514RD
LCD1	EL1602B

4. Programovanie mikrokontrolérov

4.1 Programovací jazyk FastAVR

Použitý programovací jazyk spája jednoduchosť jazyka Basic a približuje sa rýchlosti programov písaných v jazyku C. Použitie programovacieho prostredia FastAVR 4.0.1 je prehľadné a rýchlo sa v ňom užívateľ zorientuje. Toto prostredie bolo zvolené zámerne preto, aby budúci užívateľ zariadenia mohol rýchlo zmeniť naprogramovaný model a nemusel strácať čas pri spoznávaní zložitých programátorských prostredí pre jazyk C.

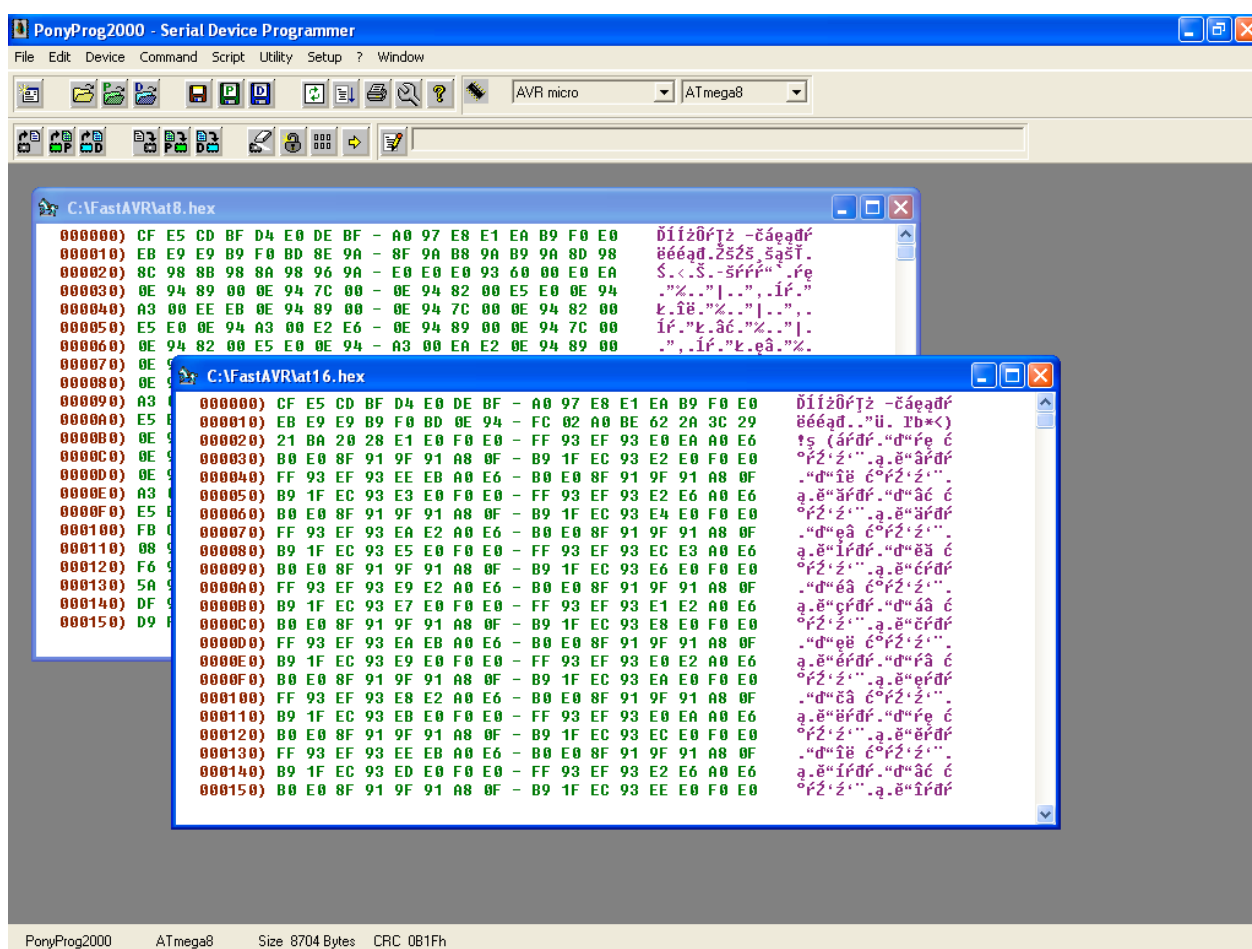


Obr. 4.1 Prostredie kompilátora FastAVR

Po nahraní zdrojového súboru a jeho úprave sa program skompiluje klávesovou skratkou F5. Výsledný kód sa automaticky uloží do súboru s rovnakým názvom ako zdrojový program s príponou hex.

4.2 Nahrávanie programov do mikrokontrolérov

Mikrokontroléry je možné programovať vďaka rozhraniu SPI priamo v zapojení, čo značne urýchľuje programovanie a ladenie aplikácií. Ako programátor bola zvolená jednoduchá konštrukcia podľa obrázku 4.2, známa pod menom PonyProg programátor, ktorá sa zmestí do krytu pre konektor Cannon. PonyProg programátor je ovládaný programom rovnakého mena, PonyProg2000, program je k dispozícii pre systémy Windows aj Linux.



Obr. 4.2 Prostredie programu PonyProg2000

Postup pri programovaní je nasledovný:

Po spustení programu sa nahrá do editora programu súbor pre mikrokontrolér v hex formáte, v menu „DEVICE“ sa vyberie typ programovaného obvodu, v našom prípade Atmega8 alebo Atmega16. V menu „SETUP“ → „Interface Setup“ sa zvolí číslo paralelného portu na ktorý je programátor pripojený a typ programátora, v našom prípade „AVR ISP I/O“. Ak by nebolo možné zvoliť paralelný port, je treba v biose počítača nastaviť LPT port do režimu normal alebo standard LPT port. Režimy EPP, ECP programátor nepodporuje. Kontrolu spojenia PC s programovaným obvodom v zapnutom zariadení možno overiť tlačidlom „PROBE“, pri vypnutom zariadení vyhlási chybu.

V menu „Program“ → „Program Option“ sa zaškrtnú políčka „Reload Files“, „Read Osc.Calibration Byte“, „Erase“, „Write Program Memory Flash“, „Write Security and Config bits“.

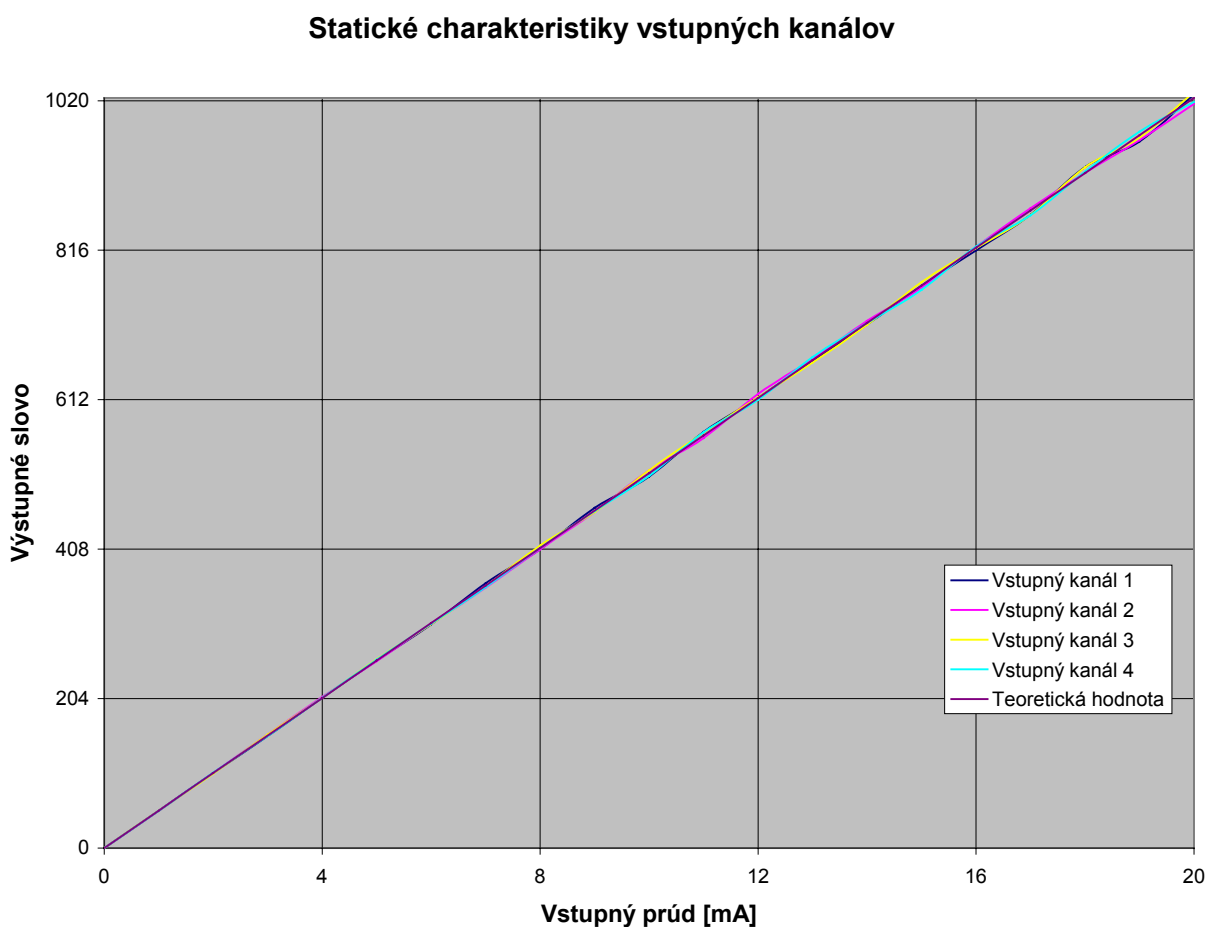
Programovanie mikrokontroléru sa spustí príkazom „Program“ z menu „Command“ alebo klávesovou skratkou Ctrl+P. Samotné programovanie trvá niekoľko sekúnd a po nahratí programu do mikrokontroléru sa nový program automaticky spustí.

5.1 Overenie činnosti zariadenia

Na zmeranie charakteristík vstupno výstupných kanálov zariadenia bol použitý merací prístroj UNIT-T UT70B, s chybou na použitom meracom prúdovom rozsahu 400mA najviac 1% z nameraného údaju.

5.2 Statická charakteristika AD prevodníkov

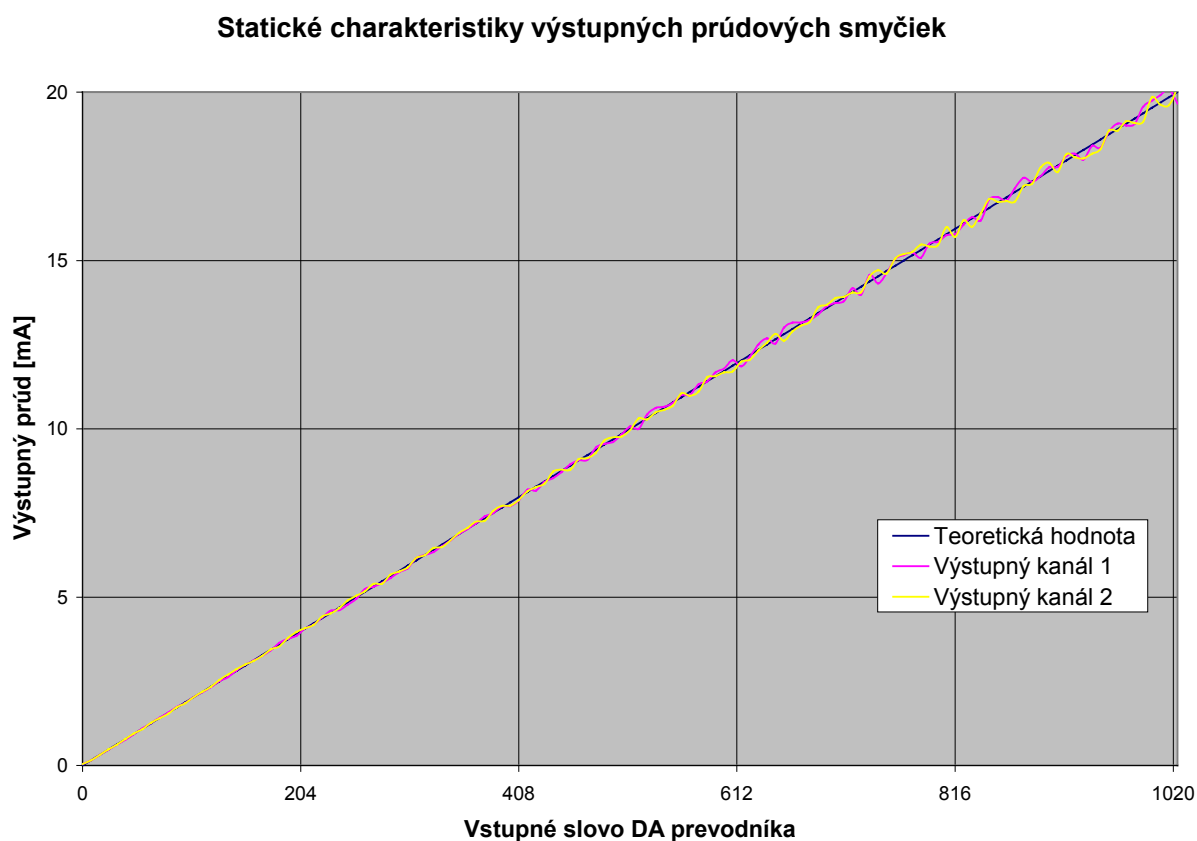
Na obrázku 5.1 je grafické závislosť výstupného slova prevodníka od vstupného prúdu. Výsledná hodnota slova bola vypočítaná z aritmetického priemeru 50 vzoriek idúcich bezprostredne po sebe.



Obr. 5.1 Prevod prúdová smyčka-vstupné dáta

5.2 Statická charakteristika DA prevodníkov

DA prevodník popísaný v kapitole 3.2.1 pri porovnávaní výstupnej hodnoty bral priemer z 10 hodnôt AD prevodu pred uskutočnením nastavenia napätia na RC člene.

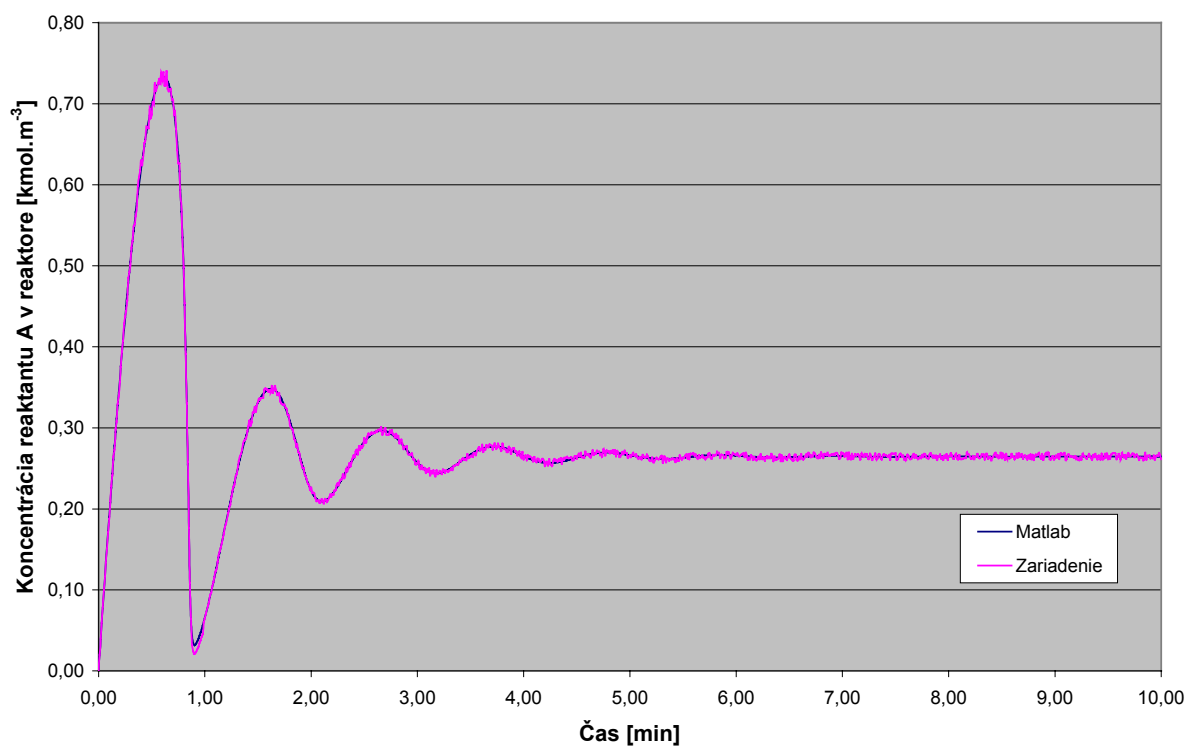


Obr. 5.2 Prevod digitálnej formy čísla na prúd tečúci prúdovou smyčkou.

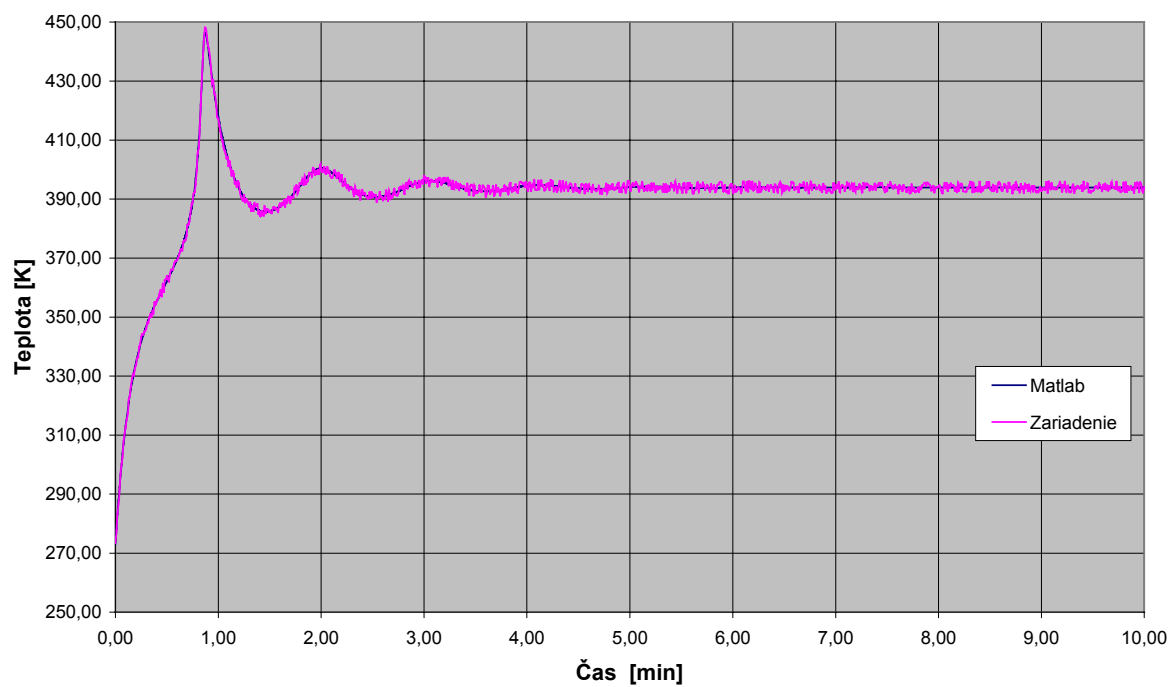
5.3 Simulácia chemického prietokového reaktora v zariadení

Na obrázku 5.3 a 5.4 je odozva zariadenia s naprogramovaným modelom prietokového reaktora pri zmene prietoku reaktantu z 0 na $1\text{m}^3\cdot\text{min}^{-1}$, so simuláciou získanou v Matlabe.

**Obr.5.3 Porovnanie odozvy zariadenia so simuláciou získanou v Matlabe
- sledovanie koncentrácie -**



Obr. 5.4 Porovnanie odozvy zariadenia so simuláciou získanou v Matlabe
- sledovanie teploty -



Záver

Úlohou tejto diplomovej práce bolo vytvorenie zariadenia ktoré by umožnilo overovanie správnej činnosti nadriadeného regulačného systému. Zhotovené zariadenie spĺňa zadané podmienky na počet vstupov, výstupov a presnosť AD a DA prevodníkov.

Zariadenie bolo naprogramované pre jeho správnu činnosť a vyskúšané pre zadaný model chemického reaktora. Správanie bolo vyhodnotené porovnaním prechodových charakteristík reaktora meraných na navrhnutom zariadení a charakteristík reaktora, získaných simuláciou na PC.

Obslužný program je navrhnutý tak, aby budúci užívateľ zariadenia bez znalosti programovania a architektúry použitých mikrokontrolérov ľahko zmenil parametre alebo celý model simulovaného procesu. Pri použití zariadenia pre iné procesy je treba zvážiť presnosť zariadenia, jeho rýchlosť a nepresnosť výpočtov v porovnaní s PC, keďže aritmetické výpočty v zariadení prebiehajú s presnosťou na 7 desatinných miest.

Použitá literatúra

1. **Matoušek, D.:** Práce s mikrokontroléry Atmel AVR, BEN – technická literatúra Praha, 2003
2. **Hrbáček, J.:** Komunikace mikrokontroléru s okolím, I. díl, BEN – technická literatúra Praha, 1999
3. **Jedlička, P.:** Přehled obvodů řady TTL 7400 I . a II. Díl, BEN – technická literatúra Praha, 2000
4. **Rusek, M., Cwirko, R., Marciniak, W.,** ABC elektroniky, ALFA Bratislava, 1992
5. **Bafrnec, M., Bálež, V., Langfelder, I., Longauer, J.:**Chemické Inžinierstvo, I. díl, MALE CENTRUM Bratislava, 1999
6. www.atmel.com, www stránky firmy ATMEL
7. www.fastavr.com, www stránky firmy MicroDesign
8. www.datasheet.com, katalogové listy súčiastok

Príloha

Výpis programu AT16.bas

```
'////////////////////////////////////
'/// FastAVR Basic Compiler for AVR by MICRODESIGN ///
'/// Marlinov Reaktor
'////////////////////////////////////
$Device= ATmega16          ' used device
$Stack = 32                ' stack depth
$Clock = 12                ' adjust for used crystal
$Baud=120000
$Lcd=PORTA.4,RS=PORTA.3,EN=PORTA.2,16,2
$LeadChar="0" ,Format(4,3)
' Definície konstant
Const waitdisp =100

$Def LE1=PORTD.7           ' latch signal pre 74573 displej1
$Def LE2=PORTD.2           ' latch signal pre 74573 displej2
$Def LE3=PORTD.3           ' latch signal pre 74573 displej3
$Def LE4=PORTD.4           ' latch signal pre 74573 displej4
$Def LE5=PORTD.5           ' latch signal pre 74573 displej5
$Def LE6=PORTD.6           ' latch signal pre 74573 displej6

$Def d1=PORTB.3            ' rozsvietenie tisícok na displejoch
$Def d2=PORTB.2            ' rozsvietenie stoviek
$Def d3=PORTB.1            ' rozsvietenie desiatok
$Def d4=PORTB.0            ' rozsvietenie jednotiek
$Def dataport=PORTC        ' Data pre 74573, rozsvetuje zadane segmenty cislicovky

' Declaracie procedur
Declare Sub displej()
Declare Sub novedataprevystup()
Declare Sub nastavDA()
Declare Interrupt Urxc()

Dim realne As Float
Dim timeRC As Float
Dim TabulkaZnakov As Flash Byte

Dim textdisp1 As String * 10
Dim textdisp2 As String * 10
Dim textdisp3 As String * 10
Dim textdisp4 As String * 10
Dim textdisp5 As String * 10
Dim textdisp6 As String * 10
Dim pocitadlow As Word
Dim outDA1 As Word
Dim outDA2 As Word
Dim outDAp1 As Word
Dim outDAp2 As Word
Dim rx As Bit
Dim tmpbit1 As Bit
Dim tmpb1 As Byte
Dim tmpb2 As Byte
```

```

Dim tmpb3 As Byte
Dim tmpb4 As Byte
Dim tmpb5 As Byte
Dim tmpb6 As Byte
Dim tmpw1 As Word
Dim tmpw2 As Word
Dim prijata As Byte
Dim Vystup(24) As Byte
vystup=(160,190,98,42,60,41,33,186,32,40,160,190,98,42,60,41,33,186,32,40,160,190,98,42)
realne=5.234
rx=0
Set PORTA.0
Set PORTA.1
DDRA=255
DDRB=255
DDRC=255
DDRD=&b11111100
' PORTA=0
PORTB=0
PORTC=255
PORTD=0
Enable Interrupts
Enable Urxc
InitLcd()
rx=1
Do
Displej()
If rx=1 Then
    outdap1=outda1
    outdap2=outda2
    nastavDA()
    novedataprevystup()
    rx=0
End If
Loop

Interrupt Urxc(), Save All '4 stacilo
Disable Urxc
InputBin tmpb5
InputBin textdisp1
InputBin textdisp2
InputBin textdisp3
InputBin textdisp4
InputBin textdisp5
InputBin textdisp6
InputBin outDA1,2
InputBin outDA2,2
rx=1
Enable Urxc
End Interrupt

Sub displej()
dataport=vystup(0)
Set LE1
Reset LE1

```

dataport=vystup(4)
Set LE2
Reset LE2
dataport=vystup(8)
Set LE3
Reset LE3
dataport=vystup(12)
Set LE4
Reset LE4
dataport=vystup(16)
Set LE5
Reset LE5
dataport=vystup(20)
Set LE6
Reset le6
Set D1
WaitUs waitdisp
Reset D1

dataport=vystup(1)
Set LE1
Reset LE1
dataport=vystup(5)
Set LE2
Reset LE2
dataport=vystup(9)
Set LE3
Reset LE3
dataport=vystup(13)
Set LE4
Reset LE4
dataport=vystup(17)
Set LE5
Reset LE5
dataport=vystup(21)
Set LE6
Reset le6
Set D2
WaitUs waitdisp
Reset D2

dataport=vystup(2)
Set LE1
Reset LE1
dataport=vystup(6)
Set LE2
Reset LE2
dataport=vystup(10)
Set LE3
Reset LE3
dataport=vystup(14)
Set LE4
Reset LE4
dataport=vystup(18)
Set LE5
Reset LE5

```

dataport=vystup(22)
Set LE6
Reset LE6
Set D3
WaitUs waitdisp
Reset D3

```

```

dataport=vystup(3)
Set LE1
Reset LE1
dataport=vystup(7)
Set LE2
Reset LE2
dataport=vystup(11)
Set LE3
Reset LE3
dataport=vystup(15)
Set LE4
Reset LE4
dataport=vystup(19)
Set LE5
Reset LE5
dataport=vystup(23)
Set LE6
Reset LE6
Set D4
WaitUs waitdisp
Reset D4
Exit Sub
End Sub

```

```

Sub novodataprevystup()
'displej1
tmpb3=0
tmpbit1=0
If Asc(textdisp1,1)=45 Then
    vystup(tmpb3)=127
    Incr tmpb3
End If
For tmpb1 = 2 To 9
    tmpb2= Asc(textdisp1,tmpb1)-48
    If tmpb2=0 And Asc(textdisp1,tmpb1+1)=46 Then GoTo c2
    If tmpb2=0 And tmpbit1=0 Then GoTo c1
c2:    tmpb4=TabulkaZnakov(tmpb2)
    If Asc(textdisp1,tmpb1+1)=46 Then
        tmpb4=tmpb4-32
        Incr tmpb1
    End If
    vystup(tmpb3)=tmpb4
    If tmpb3=3 Then Exit For
    tmpbit1=1
    Incr tmpb3
c1: Next
'displej2
tmpb3=4

```

```

tmpbit1=0
If Asc(textdisp2,1)=45 Then
    vystup(tmpb3)=127
    Incr tmpb3
End If
For tmpb1 = 2 To 9
    tmpb2= Asc(textdisp2,tmpb1)-48
    If tmpb2=0 And Asc(textdisp2,tmpb1+1)=46 Then GoTo c4
    If tmpb2=0 And tmpbit1=0 Then GoTo c3
c4:    tmpb4=TabulkaZnakov(tmpb2)
    If Asc(textdisp2,tmpb1+1)=46 Then
        tmpb4=tmpb4-32
        Incr tmpb1
    End If
    vystup(tmpb3)=tmpb4
    If tmpb3=7 Then Exit For
    tmpbit1=1
    Incr tmpb3
c3:
Next

'displej3
tmpb3=8
tmpbit1=0
If Asc(textdisp3,1)=45 Then
    vystup(tmpb3)=127
    Incr tmpb3
End If
For tmpb1 = 2 To 9
    tmpb2= Asc(textdisp3,tmpb1)-48
    If tmpb2=0 And Asc(textdisp3,tmpb1+1)=46 Then GoTo c6
    If tmpb2=0 And tmpbit1=0 Then GoTo c5
c6:    tmpb4=TabulkaZnakov(tmpb2)
    If Asc(textdisp3,tmpb1+1)=46 Then
        tmpb4=tmpb4-32
        Incr tmpb1
    End If
    vystup(tmpb3)=tmpb4
    If tmpb3=11 Then Exit For
    tmpbit1=1
    Incr tmpb3
c5:
Next

'displej4

tmpb3=12
tmpbit1=0
If Asc(textdisp4,1)=45 Then
    vystup(tmpb3)=127
    Incr tmpb3
End If
For tmpb1 = 2 To 9
    tmpb2= Asc(textdisp4,tmpb1)-48
    If tmpb2=0 And Asc(textdisp4,tmpb1+1)=46 Then GoTo c8
    If tmpb2=0 And tmpbit1=0 Then GoTo c7

```

```

c8:    tmpb4=TabulkaZnakov(tmpb2)
      If Asc(textdisp4,tmpb1+1)=46 Then
            tmpb4=tmpb4-32
            Incr tmpb1
      End If
      vystup(tmpb3)=tmpb4
      If tmpb3=15 Then Exit For
      tmpbit1=1
      Incr tmpb3
c7:
Next

'displej5
tmpb3=16
tmpbit1=0
If Asc(textdisp5,1)=45 Then
      vystup(tmpb3)=127
      Incr tmpb3
End If
For tmpb1 = 2 To 9
      tmpb2= Asc(textdisp5,tmpb1)-48
      If tmpb2=0 And Asc(textdisp5,tmpb1+1)=46 Then GoTo c10
      If tmpb2=0 And tmpbit1=0 Then GoTo c9
c10:    tmpb4=TabulkaZnakov(tmpb2)
      If Asc(textdisp5,tmpb1+1)=46 Then
            tmpb4=tmpb4-32
            Incr tmpb1
      End If
      vystup(tmpb3)=tmpb4
      If tmpb3=19 Then Exit For
      tmpbit1=1
      Incr tmpb3
c9:
Next)'

'displej6
tmpb3=20
tmpbit1=0
If Asc(textdisp6,1)=45 Then
      vystup(tmpb3)=127
      Incr tmpb3
End If
For tmpb1 = 2 To 9
      tmpb2= Asc(textdisp6,tmpb1)-48
      If tmpb2=0 And Asc(textdisp6,tmpb1+1)=46 Then GoTo c12
      If tmpb2=0 And tmpbit1=0 Then GoTo c11
c12:    tmpb4=TabulkaZnakov(tmpb2)
      If Asc(textdisp6,tmpb1+1)=46 Then
            tmpb4=tmpb4-32
            Incr tmpb1
      End If
      vystup(tmpb3)=tmpb4
      If tmpb3=23 Then Exit For
      tmpbit1=1
      Incr tmpb3

```



```

c11:
Next
End Sub

Sub nastavDA()
Do
Start Adc
tmpw2=0
For tmpb1=1 To 50
tmpw1=Adc(0)
tmpw2=tmpw2+tmpw1
Next
tmpw1=tmpw2/50

'If (tmpw1+3)>outDAp1 And (tmpw1-3)<outdap1 Then exit loop
If tmpw1=outDAp1 Then Exit Do
If tmpw1 > outDAp1 Then
    timeRC=Abs(Log((0-outdap1)/(0-tmpw1)))*1.55E-3/2.7E-7
    tmpw2=timeRC
    Set DDRA.0
    Reset PORTA.0
    Pulse PORTA.0,0,tmpw2
    Reset DDRA.0
    Reset PORTA.0

ElseIf tmpw1 < outDAp1 Then
    timeRC=Abs(Log((1170-outdap1)/(1170-tmpw1)))*1.55E-3/2.7E-7
    tmpw2=timeRC
    Set PORTA.0
    Set DDRA.0
    Pulse PORTA.0,1,tmpw2
    Reset DDRA.0
    Reset PORTA.0

End If
tmpw2=0
For tmpb1=1 To 50
tmpw1=Adc(1)
tmpw2=tmpw2+tmpw1
Next
tmpw1=tmpw2/50

'If (tmpw1+3)>outDAp2 And (tmpw1-3)<outdap2 Then exit loop
If tmpw1=outDAp2 Then Exit Do
If tmpw1 > outDAp2 Then
    timeRC=Abs(Log((0-outdap2)/(0-tmpw1)))*1.55E-3/2.7E-7
    tmpw2=timeRC
    Set DDRA.1
    Reset PORTA.1
    Pulse PORTA.1,0,tmpw2
    Reset DDRA.1
    Reset PORTA.1

ElseIf tmpw1 < outDAp2 Then
    timeRC=Abs(Log((1170-outdap2)/(1170-tmpw1)))*1.55E-3/2.7E-7
    tmpw2=timeRC
    Set PORTA.1

```

```
        Set DDRA.1
        Pulse PORTA.1,1,tmpw2
        Reset DDRA.1
        Reset PORTA.1
End If

Loop

End Sub

TabulkaZnakov = 160,190,98,42,60,41,33,186,32,40

End
```

Výpis programu AT8.bas

```
'////////////////////////////////////
'/// FastAVR Basic Compiler for AVR by MICRODESIGN ///
'/// Marlinov reaktor
'////////////////////////////////////
$Device= ATmega8                ' used device
$Stack = 100                    ' stack depth
$Clock = 12                     ' adjust for used crystal
$Baud=120000
$Timer1=Timer,Prescale=256
$LeadChar="0", Format(4,3)
$Def R1=PORTD.6
$Def R2=PORTD.7
$Def R3=PORTB.0
$Def R4=PORTB.1
$Def S1=PORTD.5
$Def S2=PORTD.4
$Def S3=PORTD.3
$Def S4=PORTD.2

Declare Sub vypis()
Declare Interrupt Ovfl()
Declare Interrupt Adc()
Dim ca As Float
Dim dca As Float
Dim T As Float
Dim dT As Float
Dim time As Float
Dim mv1 As Float
Dim mv2 As Float
Dim mv3 As Float
Dim mv4 As Float
Dim mv5 As Float
Dim mv6 As Float
Dim ad1 As Float
Dim ad2 As Float
Dim ad3 As Float
Dim ad4 As Float

Const V = 1.0E0                ' Konstanty reakcie a reaktora
Const cp = 1.0
Const ro = 1.0E6
Const k0 = 1.0E10
Const ER= 8330.1
Const dh= -1.3E8
Const cpc= 1.
Const roc = 1.0E6
Const a= 1678450.
Const b= 5.0E-1
```

Const dtime = 1.0E-3

Const ca0dolna= 0.0 ' definície intervalov, dolna= 4 mA, horna 20mA

Const ca0horna= 2.0

Const Tfdolna=300.0

Const Tfhorna=500.0

Const qdolna=0.0

Const qhorna=2.0

Const qcdolna=5.0

Const qchorna=30.0

Const Tcfdolna=250.0

Const Tcfhorna=400.0

Const cadolna=0

Const cahorna=1

Const Tdolna=250

Const Thorna=450

Dim textdisp1 As String * 10

Dim textdisp2 As String * 10

Dim textdisp3 As String * 10

Dim textdisp4 As String * 10

Dim textdisp5 As String * 10

Dim textdisp6 As String * 10

Dim outDA1 As Word

Dim outDA2 As Word

Dim odosli As Bit

Dim real As Float

Dim tmpf1 As Float

Dim tmpf2 As Float

Dim b As Byte

Dim tmpb1 As Byte

Dim tmpb2 As Byte

Dim tmpw1 As Word

Dim tmpw2 As Word

Dim cakaj As Byte

DDRD.6=1

DDRD.7=1

DDRB.0=1

DDRB.1=1

DDRD.5=0

DDRD.4=0

DDRD.3=0

DDRD.2=0

odosli=0

Enable Interrupts

Enable Ovfl

Enable Adc

Start Timer1

Wait 1

```
Dim ca0 As Float
Dim Tf As Float
Dim q As Float
Dim qc As Float
Dim Tcf As Float
```

```
ca0=2
Tf=323
q=0
qc=15
Tcf=365
ca=0.3
T=273.15
time=0
```

```
Do
tmpw2=0
Start Adc
For tmpb1=1 To 50
tmpw1=Adc(0)
tmpw2=tmpw2+tmpw1
Next
Stop Adc
ad1=tmpw2/50
```

```
tmpw2=0
Start Adc
For tmpb1=1 To 50
tmpw1=Adc(1)
tmpw2=tmpw2+tmpw1
Next
Stop Adc
ad2=tmpw2/50
```

```
tmpw2=0
Start Adc
For tmpb1=1 To 50
tmpw1=Adc(2)
tmpw2=tmpw2+tmpw1
Next
Stop Adc
ad3=tmpw2/50
```

```
tmpw2=0
Start Adc
For tmpb1=1 To 50
tmpw1=Adc(3)
tmpw2=tmpw2+tmpw1
Next
Stop Adc
ad4=tmpw2/50
```

```

'ca0=(ca0horna-ca0dolna)/819.2*(AD1-204.8)
prevodnika
q=(qhorna-qdolna)/819.2*(AD2-204.8)
'Tf=(Tfhorna-Tfdolna)/819.2*(AD3-204.8)
'Tcf=(Tcfhorna-Tcfdolna)/819.2*(AD4-204.8)
'qc=(qchorna-qcdolna)/819.2*(ADX-204.8)

```

'Priradenie hodnot premennym podľa kanálov AD

```

dca = (q / V * (ca0 - ca) - k0 * Exp(-ER / T) * ca) * dtime
mv1=q / V * (Tf - T)
mv2=a * Pow(qc , (b + 1)) / (V * ro * cp)
mv3=qc + a * Pow(qc , b) / (2 * roc * cpc)
mv4=(-dH) * k0 * Exp(-ER / T) * ca / (ro * cp)
dT = (mv1 - mv2/mv3 * (T - Tcf) +mv4 ) * dtime

```

```

ca = ca + dca
T = T + dT
time = time + dtime

```

```

If odosli=1 Then
    tmpf1=204.8+(ca-cadolna)/(cahorna-cadolna)*819.2
    tmpf2=204.8+(T-Tdolna)/(Thorna-Tdolna)*819.2
    textdisp1=Str(ca0)
    textdisp2=Str(q)
    textdisp3=Str(qc)
    textdisp4=Str(time)
    textdisp5=Str(ca)
    textdisp6=Str(T)
    tmpw1=tmpf1
    tmpw2=tmpf2
    outDA1=tmpw1
    outDA2=tmpw2
    vypis()
    odosli=0
End If
Loop

```

```

Interrupt Ovfl()
odosli=1
End Interrupt

```

```

Interrupt acd()
End Interrupt

```

```

Sub vypis()

```

```

PrintBin 1
WaitMs 1
For tmpb1=1 To 10
PrintBin Asc(textdisp1,tmpb1)
Next

```

```
For tmpb1=1 To 10
PrintBin Asc(textdisp2,tmpb1)
Next
```

```
For tmpb1=1 To 10
PrintBin Asc(textdisp3,tmpb1)
Next
```

```
For tmpb1=1 To 10
PrintBin Asc(textdisp5,tmpb1)
Next
```

```
For tmpb1=1 To 10
PrintBin Asc(textdisp6,tmpb1)
Next
```

```
For tmpb1=1 To 10
PrintBin Asc(textdisp4,tmpb1)
Next
PrintBin outDA1,2
PrintBin outDA2,2
```

```
End Sub
```

```
End
```