

Univerza v Ljubljani
Fakulteta za Elektrotehniko

Andraž Korenč

Elektronsko breme (Electronic Load)

Predstavitev projekta
pri predmetu Seminar

V Ljubljani, Junij 2012

Uvod

Kaj je elektronsko breme?

Elektronsko breme je naprava, ki simulira ohmsko breme drugi elektronski napravi. Lahko bi rekli, da je to naprava, ki deluje kot neke vrste spremenljivi upor oz. potenciometer. Ko generator bremenimo s uporom fiksne vrednosti lahko dosežemo samo eno (fiksno) vrednost izhodnega toka, medtem ko lahko z elektronskim bremenom izhodni tok spreminjamo v različnih območjih. Breme pretvarja električno energijo generatorja v toploto, ki se spošča na hladilnem rebro. Hlajenje je lahko vodno ali pa ga realiziramo z ventilatorjem in hladilnim rebrom primerne velikosti.

V breme lahko torej teče le predhodno nastavljena vrednost toka, katera se minimalno spreminja oz. se ne spreminja gleda na temperaturo bremena in ostalih zunanjih dejavnikov.

Ideja o izdelavi elektronskega bremena je nastala med študentskim delom v enem od slovenskih podjetij. Potrebovali smo breme za testiranje LED napajalnikov z konstantnim izhodnim tokom v območju med 2,5 in 3,5 A (50-70W). Želeli smo, da bi bilo breme kar se da enostavno za izdelavo in čim cenejše.

Prvotna ideja je bila, da bi izdelal analogno vezje z maksimalnima vhodnima vrednostima toka in napetost:

$$\begin{aligned} I_{in_{max}} &= 4ADC \\ U_{in_{max}} &= 30VDC \end{aligned}$$

Breme bi delovalo v načinu konstanten tok oz. constan current mode (CC mode) z maksimalno vhodno močjo 120W.

Vežje sem se odločil nadgraditi z mikrokrmilnikom in ga uporabiti kot projekt pri predmetu Seminar.

Zaradi bojazni prekomernega segrevanja hladilnega rebra (MOSFET tranzistor) in problema njegovega hlajenja sem se odločil nekoliko spremeniti vrednosti vhodnih podatkov bremena. Vhodni tok in napetost sem zmanjšal na vrednost med 2,5 in 3,5 A (50-70W).

Za srce vezja sem uporabil Atmel-ov čip Atmega32 (40 pin). Potreben programator sem izdelal sam po predlogi, ki sem jo našel internetu - USBasp programator za AVR mikrokrmilnike (<http://www.fischl.de/usbasp/>).

Končen izdelek sem zapakiral v enostavno ohišje in ga opremil z hladilnim rebrom in ventilatorjem. Podrobnejši opis izdelka najdete v naslednjih poglavjih tega dokumenta.

Specifikacija naprave

Elektronsko breme je narejeno tako, da bo vhodni tok v breme konstanten in enak predhodno nastavljeni vrednosti. Za nastavljanje vrednosti toka so na čelni strani naprave pritrjene štiri tipke (slika 1).

Specifikacija :

- **Napajanje:** za napajanje vezja sem uporabil polnilec odsluženega prenosnega računalnika- $U_{out}= 19V VDC$; $P_{max}=90W$
- **Vhodni podatki bremena:**
 $I_{in} = 2.5 - 3.5 ADC$
 $U_{in_{max}} = 30VDC$
 $P_{in_{max}} = 105W$
- **Dimenzije ohišja:**
dolžina: 130 mm
širina : 160 mm
višina: 90 mm
- **Teža naprave:** 1.3 kg
- **Slika naprave:**



Časovni in finančni plan

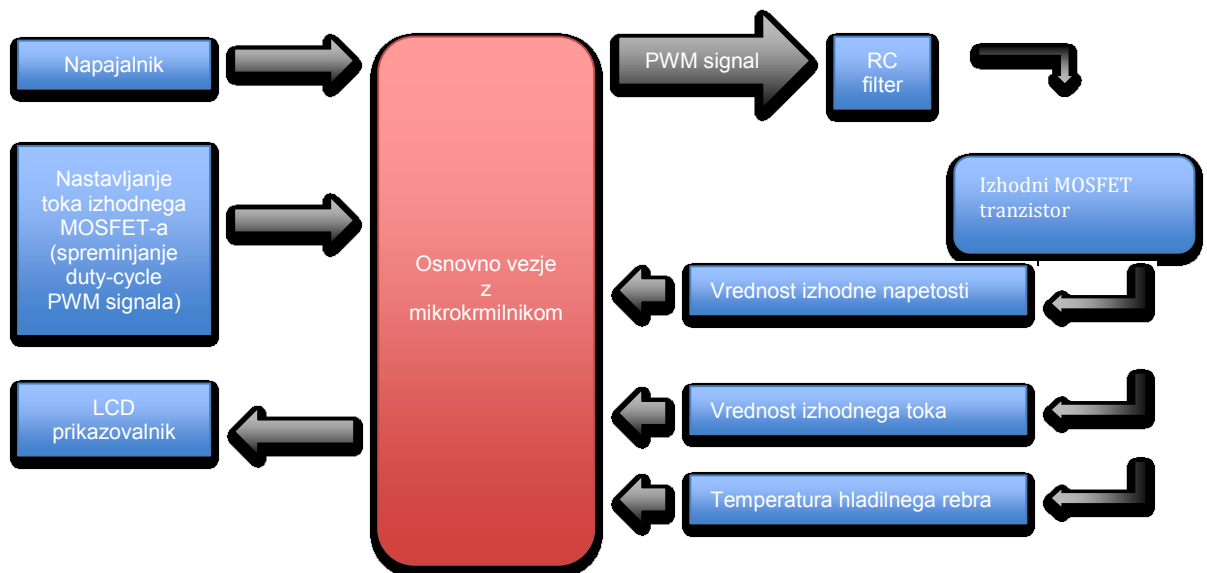
Sam projekt (brez testiranja in pisanja poročila) sem planiral izdelati v roku 14 dni (100h). V tem času bom spisal program za mikrokontroler, izdelal shemo vezja, ploščico vezja, nakupil potreben material (elemente + ohišje + hladilno rebro) in vse naštetost sestavljal v končno podobo.

Samemu testiranju naprave sem nameraval nameniti en dan in še en dodaten dan za pisanje poročila.

Kar se tiče finančnega plana, sem idejo nameraval realizirati z proračunom cca. 70€. S tem denarjem bi nakupil potrebne elemente, ohišje in hladilno rebro.

Načrti

1. Bločna shema

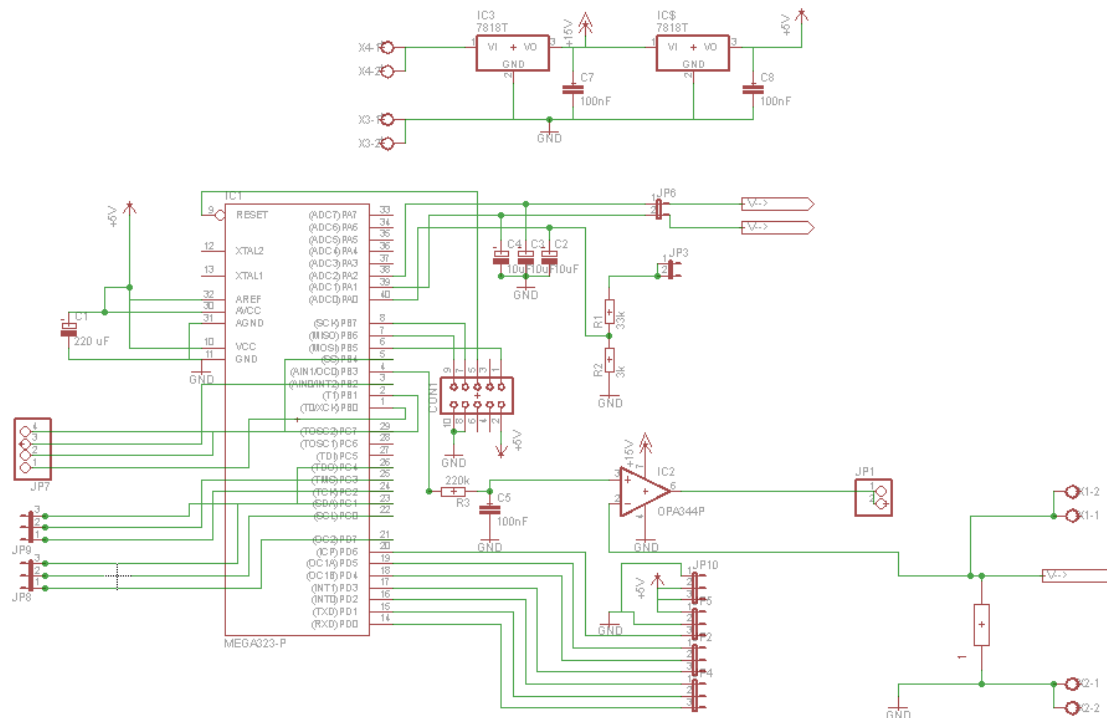


Z vrednostjo pritisnjene tipke na vhodu nastavljam duty-cycle PWM signala s katerim preko RC-filtra krmilimo izhodni MOSFET tranzistor.

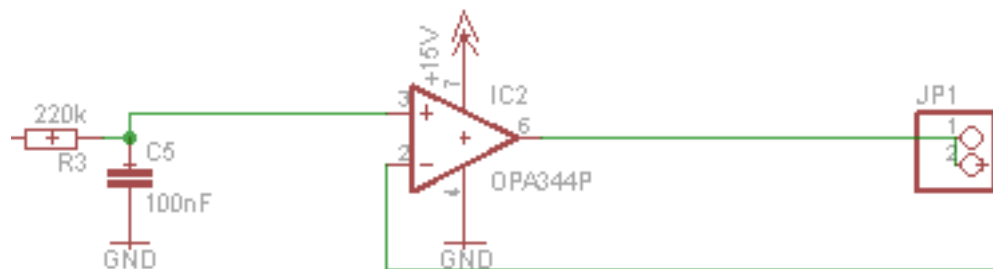
Prvotna ideja je bila, da bi preko treh kanalov digitalno-analognega pretvornika brali trenutne vrednosti napetosti, toka in temperature hladilnega rebra in jih izpisujemo na LCD prikazovalniku. V ta namen sem priredil PCB. Ob testiranju sem naletel na težavo z brenjem vrednosti toka in napetosti (prisotnost šuma). Zaradi opanega problema sem se odločil, da to verzijo bremena realiziram brez omenjenega prikazovanja trenutnih vrednosti. Več o tem lahko preberete v poglavju merilni rezultati.

2. Vežje

Celotno vezje:

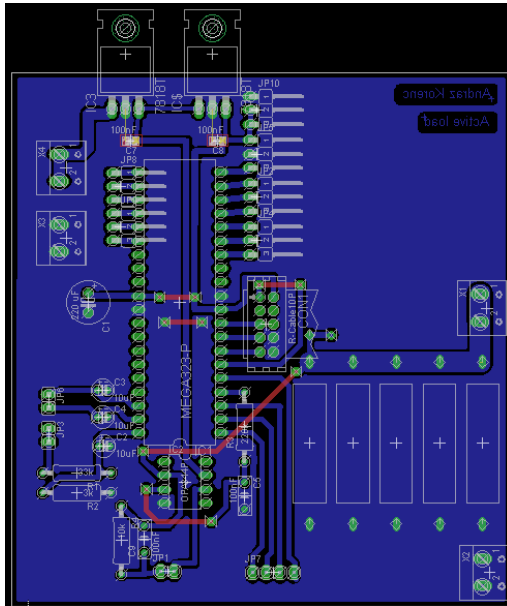


Zgornje celotno vezje je sestavljeno iz večim podvezij. Opis najpomembnejšega podvezja je prikazan na spodnji slika.



Na zgornji sliki je prikazana realizacija krmiljenja izhodnega MOSFET tranzistorja. Izhodi pin mikrokrmilnika je vezan na RC filter, ki ga sestavljata upor R3 in kondenzator C5. Vhod filtra je vezan na pin PB3- izhodni PWM signal. Izhod filtra je vezan na neinvertrajoč vhod operacijskega ojačevalnika IC2. Invertrajoča sponka operacijskega ojačevalnika je vezana na upore (zgornja slika), preko katerih teče celoten vhodni tok v breme. Napetost na teh uporih se primerja z napetostjo izhoda RC filtra in posledično krmili izhodni MOSFET tranzistor (JP1).

3. Tiskanina



Tiskanino sem realiziral kot enostransko tiskanino. Pri realizaciji sem uporabil zalito maso (Ratsnest možnost v programu Eagle). Velikost tiskanine je 90x100 mm.

Končni izgled tiskanine prikazuje leva slika.

Rdeče povezave nisem uspel realizirati na enostranski tiskanini, zato sem jih označil kot povezave na zgornji plasti in jih kasneje ročno povezal.

4. Program

Program sem spisal v programskih paketu WinAVR, AVR Studio in ga prenesek na mikroprocesor s programom ExtremeBurner. WinAVR je skupek odprtokodnih programov nemenjenih delu z mikroprocesorji Atmel AVR.

AVR Studio je programski paket - razvojno orodje podjetja ATMEL za razvoj aplikacij, namenjenim 8-bitnim AVR mikrokontrolerjem, ter je brezplačno za uporabo . Vključuje orodje za vodenje projektov, urejevalnik, simulator in prevajalnik za programsko kodo v zbirnem jeziku (assembler), ter t.i. debugger – razhroščevalnik delovanja. Podpira ATMEL-ova razvojna orodja in programatorje, uporabo RTOS (real-time operating system) ter uporabo vtičnika za prevajalnik GCC.

GCC predstavlja odprto-kodno zbirko prevajalnikov za programske jezike C, C++, objektni C, Fortran, Java, Ada, po licenci GPL, in je del projekta GNU. Ta prevajalnik bomo v primeru uporabe jezika C srečali znotraj paketa WinAVR.

WinAVR je množica programskih orodij, namenjena delu z mikrokontrolerji AVR. Vsebuje GCC prevajalnik, ustrezne knjižnice v jeziku C (AVR Libc [2]), orodje za zbirni jezik (avr-as), programirni programski vmesnik (avrdude), JTAG ICE programski vmesnik (avarice), debugger (avr-gdb), itd.

Na naslednjih straneh sledi koda mikroprocesorja z kratkimi komentarji:

```

#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"

void InitPWM() //Inicijalizacija PWM-ja
{
  /*
  TCCR0 - Timer Counter Control Register (TIMER0)
  -----
  Opis bitov registra

  NO: NAME DESCRIPTION
  -----
  BIT 7 : FOCO Force Output Compare -ni uporabljen
  BIT 6 : WGM00 Wave form generartion mode [SET to 1]
  BIT 5 : COM01 Compare Output Mode [SET to 1]
  BIT 4 : COM00 Compare Output Mode [SET to 0]

  BIT 3 : WGM01 Wave form generation mode [SET to 1]
  BIT 2 : CS02 Clock Select [SET to 0]
  BIT 1 : CS01 Clock Select [SET to 0]
  BIT 0 : CS00 Clock Select [SET to 1]

  -----
  Zgornje nastavitve veljajo za :

  Timer Clock = CPU Clock (No Prescalling)
  Mode = Fast PWM
  PWM Output = Non Inverted

  */
  TCCR0=(1<<WGM00)[(1<<WGM01)][(1<<COM01)][(1<<CS00);

}

/*****
Nastavitev duty cycle na izhodu.
-----
duty: med 0 - 255

0= 0%
255= 100%
*****/

void SetPWMOutput(uint8_t duty)
{
  OCR0=duty; // vredost Output compare register za Timer0
}

//inicijalizacija ADC
void InitADC()
{
  ADMUX=(1<<REFS0); // za nastavitev Aref=AVcc;
  ADCSRA=(1<<ADEN)[(1<<ADPS2)][(1<<ADPS1)][(1<<ADPS0); //Rrescalar div factor =128 in ADEN -enable ADC
}

uint16_t ReadADC(uint8_t ch)
{
  //Izberemo ADC Channel ch med 0-7
  ch=ch&0b00000000;
  ADMUX|=ch;

  //Startamo enkratno pretvorbo- Single conversion
  ADCSRA|=(1<<ADSC);

  //Pocakamo da se pretvorba konca
  while(!((ADCSRA & (1<<ADIF))));

  //Pobrisemo Adif tako da na njegovo mesto postavimo 1
  //V kodo napisemo '1' toda rezultat je da se bit postavi na '0' !!!

  ADCSRA|=(1<<ADIF);

  return(ADC);
}

void main()
{
  uint16_t adc_result;
  uint8_t vrednost=0;
  int pritisk =0;
  int on_level=0;

```



```
DDRB|=(1<<PB3); //data direction register port B nastavimo kot izhod PWM
DDRB &= ~(1 << PINB0 ); //data direction register port B nastavimo kot vhod
DDRB &= ~(1 << PINB1 ); //data direction register port B nastavimo kot vhod
DDRB &= ~(1 << PINB2 ); //data direction register port B nastavimo kot vhod
DDRB &= ~(1 << PINB4 ); //data direction register port B nastavimo kot vhod
PORTB |= 1 << PINB0; // vrednost vhodnega pina je 1 (5V)
PORTB |= 1 << PINB1; // vrednost vhodnega pina je 1 (5V)
PORTB |= 1 << PINB2; // vrednost vhodnega pina je 1 (5V)
PORTB |= 1 << PINB4; // vrednost vhodnega pina je 1 (5V)

//inicijalizacija PWM
InitPWM();
//inicijalizacija LCD
LCDInit(LS_NONE); // nastavev kurzorja v nacin brez kurzorja
LCDClear();
LCDWriteString("Dummy Load v1.0");
LCDWriteStringXY(0,1,"By Andraz Korenc");

InitADC();

while(1)
{
    if (bit_is_clear(PINB,0))
    {
        on_level ++;
        if(on_level>1500)
        {
            if(vrednost<255)
            {
                vrednost=vrednost+1;
                SetPWMOutput(vrednost);
            }
            on_level=0;
        }
    }

    if (bit_is_clear(PINB,1))
    {
        on_level ++;
        if(on_level>1500)
        {
            if(vrednost < 245)
            {
                vrednost=vrednost+10;
                SetPWMOutput(vrednost);
            }
            on_level=0;
        }
    }

    if (bit_is_clear(PINB,2))
    {
        on_level ++;
        if(on_level>1500)
        {
            if(vrednost>1)
            {
                vrednost=vrednost-1;
                SetPWMOutput(vrednost);
            }
            on_level=0;
        }
    }

    if (bit_is_clear(PINB,4))
    {
        on_level ++;
        if(on_level>10)
        {
            if(vrednost>10)
            {
                vrednost=vrednost-10;
                SetPWMOutput(vrednost);
            }
            on_level=0;
        }
    }

    adc_result=ReadADC(0);
    LCDWriteIntXY(5,0,adc_result,4); //nastavev prikaza
}
}
```


Kosovnica

V spodnji tabeli so zbrani uporabljeni elementi:

Naziv	Vrednost elementa	Ohisje	Št. kosov	Proizvajalec	Cena (€)
N-ch MOSFET STW120NF10	U _{ds} =100V I _{ds} =120 A	T0-247	1	STMICROELECTRONIC	2,9
OP AMP LM741	Bandwidth:1MHz; Slew Rate :0.5V/μs;	DIP-8	1	NATIONAL SEMICONDUCTOR	0,6
ER581R0JT	7W, 1Ω	Wirewound	5	TE CONNECTIVITY	4,4
L7815CV	+15 V voltage regulator	T0-220	1	STMICROELECTRONIC	0,6
LM7805	+5 V voltage regulator	T0-220	1	STMICROELECTRONIC	0,5
LC Power	19V ; 4,7A max		1	LC power	//
LCD prikazovalnik	alfanumerični , 2x16 znakov		1	RAYSTAR	13,0
Potenciometr	100 kΩ		1		2,2
Tipka			4		5,2
ATmega 32	ATmega 32, 8bit	DIP ,40 pin	1	Atmel	5,8
Hladilnik	HR 51		1	Mali	7,2
Ohisje		130x160x90	1	Elektro S	14,4
Sponke	Sauro MSM	MSM	4	Wago	0,8
Kondenzator	100 nF / 65V		2	Jamicon	0,6
Kondenzator	10 uF / 25V		3	Jamicon	0,5
Ventilator	12V ; 1,7W	80x80x20	1	Sunon	4,4
Priklopna sponka	Max. 15 A		1		1,5
SKUPAJ					64,6 €

Merilni rezultati

Kot sem omenil v predhodnih točkah lahko elektronsko breme prejme 2,5 - 3.5 A vhodnega toka (105 W vhodne moči) .

Pri zgornjih predpostavkah in vhodni napetosti 25V sem tudi začel testirati breme .

Napetost napajalnika sem nastavil na 25V in preiskusil delovanje vseh štirih vhodnih tipk. Pojavil se mi je problem "debouncing-a" tipk, ki sem ga odpravil s spremembo programske kode. Z novo nastavitvijo tipk sem lahko desegal korak nastavljanja toka cca. 40 mA.

Naslednji problem mi je predstavljal branje analognih vrednost toka. Na vhodu samega AD pretvornika je bilo prisotno toliko šuma , da so bile dobljene meritve povsem neuporabne. Odločil sem se opustiti idejo prikazovanju trenutnih vrednosti na LCD prikazovalniku. Vrednost na vhodu pretvornika bom ob priliki pomeril z osciloskopom in vezje predelal do te mere, da bom delovala tudi funkcija prikazovanja toka.

Za prikaz trenutne vrednosti vhodnega toka je tako potrebno uporabiti zunanji ampermeter.

Pri meritvah se je izkazalo, da transistor začne delovati pri vhodni napetosti vsaj 4V. Tega sem se zavedal že pred samo izdelavo, ker pa je za delovanje bremena namenjeno območje med 20 in 30V vhodne napetosti, s tem ne bomo imeli težav.

Vidimo lahko, da je maksimalen možen tok, ki ga lahko breme prejme enak 4.90 A, vendar pa je ta vrednost že izven uporabnih mej iz specifikacije, tako da pri normalni uporabi ne pride v poštev. Maksimalen tok bi lahko tudi programsko omejil z zgornjo vrednostjo duty-cycle PWM signala.

Pri večjih vhodnih močeh je prišlo do velike porabe moči na samem MOSFET tranzistorju. V kolikor se bo hladilno rebro prekomerno grelo, bom moral obstoječe rebro zamenjati z večjim in tako zmanjšati temperaturo tranzistorja.

Navodila za uporabo

Elektronsko breme ja sila enostavno za uporabo. Na čelni plošči ohišja so pritrjene štiri tipke s katerimi reguliramo vhodni tok v breme oz. tok iz naprave, ki jo testiramo. S tipko 1 povečujemo vhodni tok za majhen korak, kateri se poveča za 10-krat, če pritisnemo tipko 2. Obrato pa za zmanjševanje vhodnega toka uporabljamo tipki 3 in 4 (glej spodnjo sliko).



Za napajanje elektronskega bremena je uporabljen AC adapter računalniškega polnilnika baterije.

Na čelni plošči napreve se nahaj rotacijski potenciometer s katerim nastavljamo contrast LCD prikazovalnika. V primer nejasnega prikaza vrednosti na zaslonu, prikaz le-teh popravimo z potenciometrom.

Maksimalne vrednosti vhodnih podatkov bremena so sledeče:

$$\begin{aligned} I_{in_{max}} &= 3,5 \text{ ADC} \\ U_{in_{max}} &= 30 \text{ VDC} \\ P_{in_{max}} &= 105 \text{ W} \end{aligned}$$

Opozorilo:

Med delovanjem se lahko hladilno telo segreje do 100°C ali več. V kolikor se ohisje naprave odpira po končani uporabi in se hladilno rebro še ni ohladilo je potrebno biti pozoren da ne pride do opeklina ali drugih poškodb.

Na zadnji strani naprave se nahaja ventilator, ki je namenjen odvajanju toplote hladilnega rebra. V normalnem delovanju mora biti naprava postavljena tako, da je pretok zraka nemoten.

Časovna in finančna rekapitulacija

Zastavljen cilj realizacije projekta v štirinajstih dneh (100 h) sem uresničil. Programiranje mikrokrmilnika mi je velo kar tri dni časa (20 h). Problem mi je povročalo branje večim vrednosti AD pretvornikov in prisoten velik šum. V čas izdelave tega projekta ne štejem časa, ki sem ga porabil za izdelavo serijskega programatorja za AVR mikokrmilnike (cca 40 ur).

Z izdelavo same sheme vezja in PCB-ja nisem imel težav. Porabil se 8 ur, da sem tiskanino narisal in dodatnih 60 ur, da sem ploščico izdelal, pritr dil elemente in jo vgradil v ohisje. Zadnjih pet ur sem porabil za preiskus delovanja, pisanje poročila in končno prezentacijo.

Samo izdelavo končnega vezja mi je nekoliko podaljšalo čakanje na elemente, katerih v Sloveniji nisem mogel kupiti in sem jih naročil preko internet.

Finančno sem projekt končal z prokoračitvijo prvotnega proračuna. V kosovnici so zajeti samo elementi napreve, ki sem jih kupil, ne pa tudi elementi ,ki sem jih že

imel doma. Prav tako nisem upošteval cene napajalne naprave, ki sem jo prav tako že imel doma.

Z upoštevanje omenjenih elementov bi strošek izdelka sigurno presegel 90 €.

V samo ceno izdelka nisem vštél stroškov dela. Glede na to, da sem izdelek izdelal v času študentskega dela v enem od slovenskih podjetij, bi za ceno ure dela postavil urno postavko v tem podjetju- 6€/h. Ob upoštevanju tega stroška se cena projekta dvigne na 660€. V kolikor bi se projekta lotil kot zaposlena oseba bi bili stroški še veliko večji.

Tekom izdelave sem imel kar nekaj problemov. Največjega mi je predstavlo branje vrednosti AD pretvornika.

Zato sem se odločil da bom obstoječi projekt nadgradil z popravljenim programom in vezjem, ki bo omogočal hkratno branje večih AD pretvornikov in izpisovanje vrednosti le-teh na LCD prikazovalniku. Polek možnosti konstantnega toka (CC mode) bom dodal tudi možnost konstantne napetosti (CV mode) in konstantne upornosti (CR mode).

V kolikor se bo pojavil problem pregrevanja tranzistorja bom moral razmisliti o novem načinu boljšega hlajenja.

Reference

<http://www.avrfreaks.net/index.php>

<http://www.atmel.com/>

<http://si.farnell.com/>

<http://winavr.sourceforge.net/>

<http://www.fischl.de/usbasp/>

<http://extremeelectronics.co.in/avr-tutorials/>

http://en.wikipedia.org/wiki/RC_circuit

<http://www.cadsoftusa.com/>

<http://www.newbiehack.com/>

<http://www.raystar-optronics.com/index.php?lang=en>

<http://fides.fe.uni-lj.si/~lrnv/racunalnistvo2/zapiskipredavanj.pdf>

<http://lpvo.fe.uni-lj.si/izobrazevanje/2-stopnja-un/seminar-sem/>

http://en.wikipedia.org/wiki/Dummy_load

Priložene datoteke

- koda mikrokrmilnika
- datoteke potrebne za izdelavo tiskanine
- prezentacija