

ŠC PTUJ, ELEKTRO IN RAČUNALNŠKA ŠOLA

VOLKMERJEVA C. 19, 2250 PTUJ

TEMPERATURNI REGULATOR ZA KRMILJENJE

230V ELEKTRIČNEGA GRELCA

4. izpitna enota

Mentor: Slavko Murko Avtor: David Korošec, 4.B

Srednje strokovno izobraževanje,

elektrotehnik

Ptuj, april 2012

Vsebina

[Kazalo slik: 4](#_Toc322375010)

[1. POVZETEK 5](#_Toc322375011)

[1.1 Povzetek 5](#_Toc322375012)

[Summary 5](#_Toc322375013)

[1.2 Ključne besede 5](#_Toc322375014)

[2. UVOD 6](#_Toc322375015)

[2.1 Predstavitev raziskovalnega problema 6](#_Toc322375016)

[2.2 Postopki dela 6](#_Toc322375017)

[2.3 Uporaba izdelka 6](#_Toc322375018)

[3. ELEKTROTEHNIKA 7](#_Toc322375019)

[3.1 Kondenzatorji 7](#_Toc322375020)

[3.2 Upori 8](#_Toc322375021)

[3.3 OPTOKOPLER 9](#_Toc322375022)

[3.4 LCD 10](#_Toc322375023)

[3.5 ČIP (PIC16F876-04/P) 11](#_Toc322375024)

[3.6 Kvarčni kristal 13](#_Toc322375025)

[3.7 Triak 13](#_Toc322375026)

[4. PRAKTIČNI DEL 14](#_Toc322375027)

[4.1 Potek dela 14](#_Toc322375028)

[Nabava materiala 15](#_Toc322375029)

[Izdelava vezja v Sprint-Layout 5.0 16](#_Toc322375030)

[Jedkanje 16](#_Toc322375031)

[Spajkanje elementov na vitro ploščo 17](#_Toc322375032)

[Meritev napetosti 17](#_Toc322375033)

[Izdelava ohišja za nalogo 18](#_Toc322375034)

[4.2 Delovno mesto 20](#_Toc322375035)

[5. ZAKLJUČEK 21](#_Toc322375036)

[6.VIRI 22](#_Toc322375037)

[7.PRILOGA 23](#_Toc322375038)

[7.1 Program naložen na PIC16F876-04/P 23](#_Toc322375039)

# Kazalo slik:

[Slika 1: Simboli različnih izvedb kondenzatorjev: fiksni, elektrolitski, spremenljivi, napetostno spremenljivi 7](#_Toc322373830)

[Slika 2: Upori 7](#_Toc322373831)

[Slika 3: Optocopler MOC3020 8](#_Toc322373832)

[Slika 4: LCD 1x16 8](#_Toc322373833)

[Slika 5: Čip 9](file:///C:\Users\Ervin\Desktop\Poročilo_matura.docx#_Toc322373834)

[Slika 6: Čip-diagram priključkov 9](#_Toc322373835)

[Slika 7: Kvarčni kristal 10](#_Toc322373836)

[Slika 8: Triac 10](#_Toc322373837)

[Slika 9: Shema vezja 11](#_Toc322373838)

[Slika 10: Vezja v Sprint-Layout 13](#_Toc322373839)

[Slika 11: Jedkanje 13](#_Toc322373840)

[Slika 12: Spajkanje 14](#_Toc322373841)

[Slika 13: Meritve 14](#_Toc322373842)

[Slika 14, 15: Izdelava ohišja 15](#_Toc322373843)

# 1. POVZETEK

## 1.1 Povzetek

V nalogi je predstavljena regulacija temperature za električni grelec. Za to nalogo me je vzpodbudilo dejstvo, da bom nalogo lahko uporabil tudi doma.

Naloga kot sama ni bila pretirano težka razen programa, ki je naložen na čip, ki mi je delal precej težav. Podatke sem pridobil z pomočjo spleta ter mentorja. Za to nalogo sem se odločil, ker v današnjem času vse temelji na avtomatsko vodenih procesih. Naloga ima namen, da zagotovi samodejno vzdrževanje temperature v prostoru z možnostjo enostavnega nastavljanja določene temperature, ki jo želimo. Izdelek je vsestransko uporaben in praktično prikazan. Naloga se lahko uporablja predvsem v kmetijskih obratih z vzrejo malih živali oz. valilnicah, ki potrebujejo konstantno temperaturo za uspešno rast.

Z poročilom sem hotel predvsem pokazati delovanje izdelka ter elemente, ki so vključeni v izdelek ter, da bi o tej nalogi lahko razmišljali tudi dijaki, ki bodo nalogo delali za mano. V poročilu je prav tako opisan tudi postopek izdelave ploščice z vezjem torej jedkanje in še prej načrtovanje vezja.

V prvem delu naloge sem prikazal elektrotehniko torej na kratko opisal vse elemente, ki jih vsebuje naloga. V drugem je prikazan praktični del to je nabava materiala, izdelava vezja, izdelava ploščice ter na koncu še meritve, ki sem jih opravil. Opisan je tudi postopek po katerem sem delal. Na koncu naloge sem pa priložil še program katerega sem zapisal v čip.

## Summary

In the report I'm presenting an electrical heater with a heat regulator. I've been inspired for this assignment because it could be used at home.

The assignment itself wans't as difficult as the program that was on the chip, which was causing me a lot of troubles. I got all the information from the internet and my mentor. I've decided for this assignment because today everything is based on automaticly runned processes. The purpose of the assignment was to sustain a constant temperature in a room with an easy regulator to set the temperature that we want. The assignment (heater) can be used anywhere and it's practacly demonstrated. The assignment can mostly be used in farming facilities that are breeding small animals or in hatcheries, that need aconstant temperature for successful growth.

With the report I wanted to show how the product work and to describe all the elements, that the assignment involves so later generatios of students would consider building or making one themselves. In the report there also whole procedure how the circuit pad was created, etching and the planing of the circuit.

In the first part of the assignment I've described all of the containing elements. In the second part the practical part is shown, this means getting the material, making the circuit, circuit pad and at the end all of the measurements that I did. There is also the whole procedure, the way everything was done. In the end of the assignment I've put the program that is on the chip.

**1.2 Ključne besede**

**- Čip -** PIC16F876-04/P

**-Temp. Sonda -** pt100

**-LCD –** 1x16 z osvetlitvijo

**Key words**

**- chip -** PIC16F876-04/P

**-** **temperature probe -** pt100

**-** **LCD - 1x16 with backlight**

# 2. UVOD

## 2.1 Predstavitev raziskovalnega problema

Naloge sem se lotil, da bi izdelal sistem ogrevanja kjer bi se temperatura konstantno ohranjevala. Za izvedbo sem uporabil čip PIC16F876-04/P, ker so cenejši in ravno dovolj zmogljivi za mojo nalogo. Z malce težjim zapisom programa na njega sem imel malo težav, vendar sem jih s pomočjo mentorja hitro rešil in celotno nalogo usposobil. Za spremljanje temperature je v nalogo dodan tudi LCD – zaslon, iz katerega lahko izberemo želeno temperaturo in trenutno temperaturo. Izdelek lahko uporabljamo predvsem v valilnicah, kjer se mora dovajati konstantna temperatura in nimamo možnosti, da bi bili vedno zraven, saj izdelek samo priključimo na napetost in že nam naloga dela po želenem, prej napisanem, programu. Ta naloga me je pritegnila predvsem zaradi že prej napisanega razloga, da lahko samo sedimo doma oz. delamo drugo delo in se ne obremenjujemo z ohranjanjem temperature tam, kjer je to najbolj potrebno.

## 2.2 Postopki dela

Komentiranje sheme.

Nabava materiala.

Izdelava vezja v Sprint-Layout 5.0.

Jedkanje.

Spajkanje elementov na vitro ploščo.

Meritev napetosti.

Izdelava ohišja za nalogo.

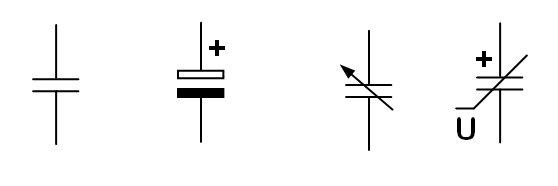
## 2.3 Uporaba izdelka

Izdelek lahko uporabljamo povsod, kjer bi radi, da imamo konstantno temperaturo brez, da bi bil potreben stalni nadzor ljudi saj nam naloga sama uravnava želeno temperaturo.

# 3. ELEKTROTEHNIKA

## 3.1 Kondenzatorji

Električni kondenzator je pasivni element, ki lahko sprejme, shrani in odda elektrino. Njegova osnovna lastnost je kapacitivnost. Podobno kot izdelujemo elemente električne upornosti, izdelujemo tudi elemente električne kapacitivnosti – kondenzatorje. Poleg uspešnosti shranjevanja energije je pomembno tudi to, kako hitro lahko to energijo kondenzator sprazni. Elektrolitske baterije na primer zelo uspešno shranjujejo veliko količino energije, ne morejo pa je hitro izkoristiti. Na drugi strani so elektrolitski kondenzatorji, ki so majhnih dimenzij in so nekoliko manj učinkoviti v smislu gostote shranjene energije, se pa lahko njihov shranjen naboj zelo hitro razelektri. Kondenzator je v osnovi sestavljen iz dveh elektrod – vzporednih prevodnih plošč, ki sta blizu skupaj, vendar se ne stikata. Med njima je izolator ali dielektrik. Poznamo več simbolov za kondenzatorje. Vsi imajo dve vzporedni črti, ki ponazarjata osnovno sestavo: dve vzporedni prevodni plošči. Ti sta lahko izvedeni na veliko različnih načinov. Kondenzatorji se med sabo razlikujejo po več kriterijih. Pomembnejši so njegova »uporabnost« (nespremenljiv, spremenljiv), odvisnost od drugih veličin (napetostno odvisen) in pravilna priključitev (elektrolitski). Prav te kriterije izražajo različni simboli za kondenzatorje, ki jih uporabljamo v električnih shemah.



Slika 1: Simboli različnih izvedb kondenzatorjev: fiksni, elektrolitski, spremenljivi, napetostno spremenljivi

## 3.2 Upori

Upor je eden najpomembnejših in najbolj uporabljanih (tako rekoč nepogrešljivih) [elektrotehničnih](http://sl.wikipedia.org/wiki/Elektrotehnika) in [elektronskih](http://sl.wikipedia.org/wiki/Elektronika) elementov, čigar glavna [veličina](http://sl.wikipedia.org/wiki/Veli%C4%8Dina) je [upornost](http://sl.wikipedia.org/wiki/Upornost) oz. njena [obratna vrednost](http://sl.wikipedia.org/wiki/Obratna_vrednost) - [prevodnost](http://sl.wikipedia.org/wiki/Prevodnost). Idealen upor ima, kar v praksi sicer ne drži povsem, konstantno upornost R, za katero velja [Ohmov zakon](http://sl.wikipedia.org/wiki/Ohmov_zakon):

U(t)=R\cdot I(t),

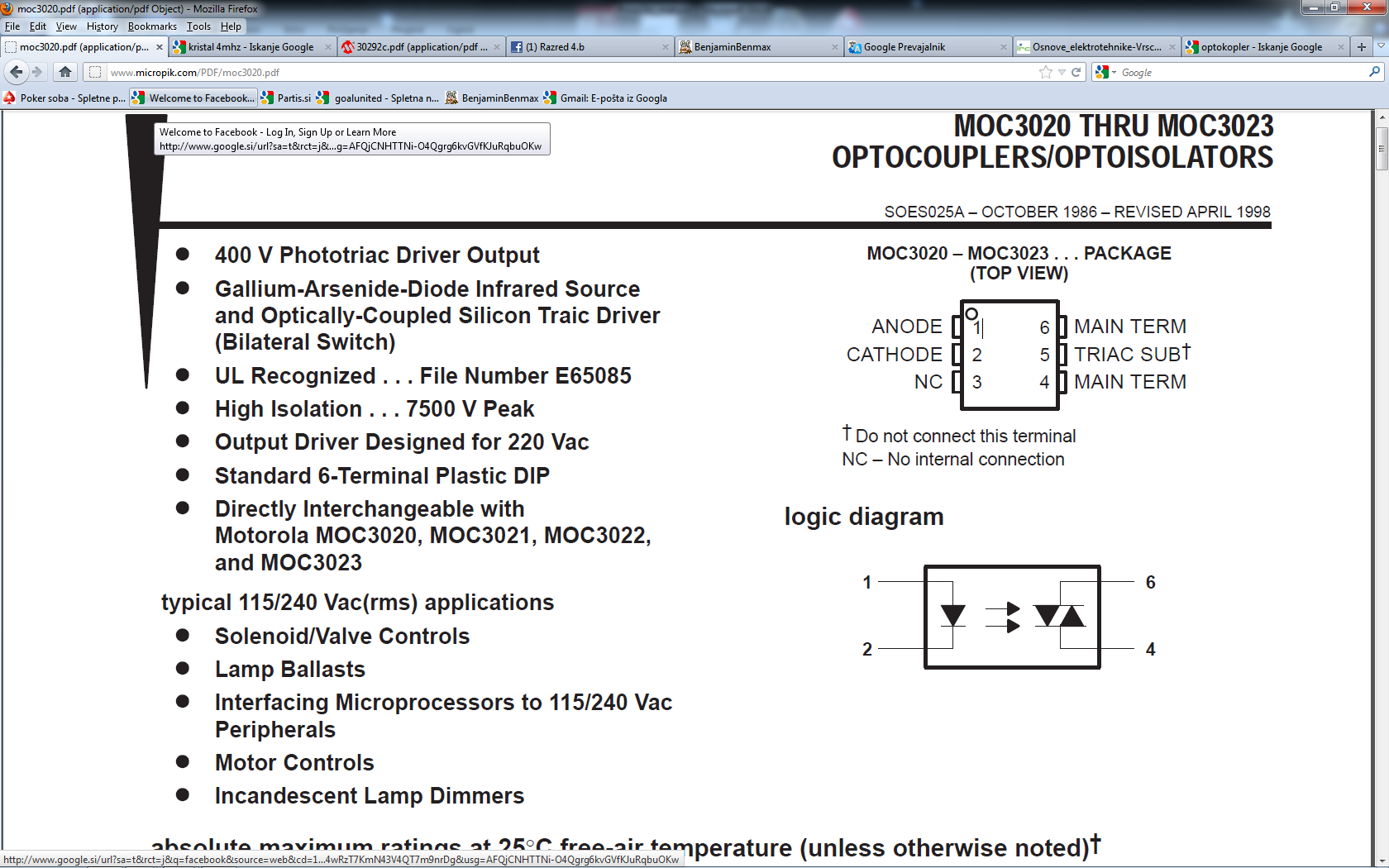
kjer U(t)pomeni trenutni padec [napetosti](http://sl.wikipedia.org/wiki/Elektri%C4%8Dna_napetost) na uporu, I(t)pa trenutni [električni tok](http://sl.wikipedia.org/wiki/Elektri%C4%8Dni_tok) skozenj. Zaradi tega je upor podobno kot [kondenzator](http://sl.wikipedia.org/wiki/Kondenzator) in idealna [dušilka](http://sl.wikipedia.org/wiki/Du%C5%A1ilka) linearen element.



Slika 2: Upori

# 3.3 OPTOKOPLER

Optokopler je integrirano vezje za galvansko ločitev električnih tokokrogov. Glavni namen je zaščita notranjega vezja pred nepričakovanimi zunanjimi vplivi (npr. prenapetost).  
Znotraj IC - ja je LED ali IR - dioda in nasproti nje sprejemni tranzistor, ki *svetlobni* signal pretvori nazaj v električnega.  
Seveda tudi ta IC za svoje delovanje potrebuje električno energijo, čeprav v shemah to zelo verjetno ni označeno.



Slika 3: Optocopler MOC3020

## 3.4 LCD - zaslon

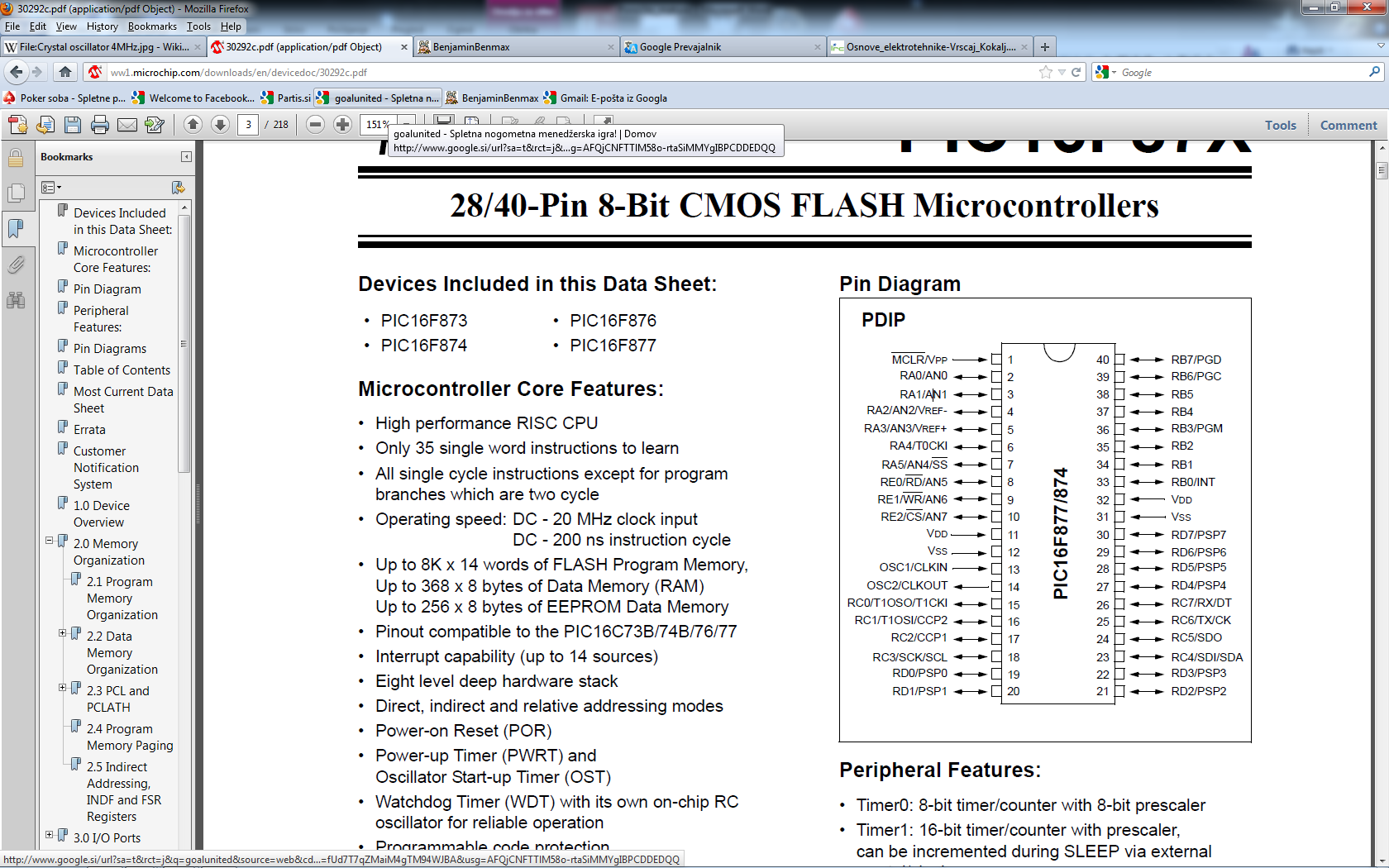
Ne uporabljamo jih samo pri računalnikih in televizijskih sprejemnikih, temveč tudi pri digitalnih fotoaparatih, urah, kamerah, gostinskih aparatih, CD-predvajalnikih in še kje. LCD-zaslon prepoznamo že na daljavo, saj ima svojo značilno tanko obliko. Slika je sestavljena iz mnogih slikovnih pik ali pikslov. V primerjavi s CRT-zasloni porablja bistveno manj energije, zaseda tudi manj prostora in prikaže razločnejšo ter bolj kontrastno sliko. LCD-zaslon za prikaz slike izkorišča lastnost tekočih kristalov, ki ob prisotnosti električnega polja spremenijo prepustnost svetlobe iz ozadja. Vsak LCD-zaslon v osnovi sestavlja šest plasti. V ozadju je vir svetlobe, ki je lahko ogledalo, fluorescentna žarnica ali svetleča dioda. Sledi plošča, ki ima vlogo polarizacijskega filtra. Nad ploščo je množica elektrod s tekočimi kristali. Kristali so vrtljiv polarizacijski filter. Kadar je tekoči kristal obrnjen enako kot polarizacijski filter v plošči, prepušča svetlobo (svetel piksel). Bolj kot je tekoči kristal zasukan, manj svetlobe prepusti (temen piksel).



Slika 4: LCD 1x16

## 3.5 ČIP (PIC16F876-04/P)

Integrirano vezje (tudi **čip**) je mikrovezje, sestavljeno iz množice elektronskih elementov, ki so na skupnem substratu iz polprevodniškega materiala med seboj povezani v električno vezje. Vezje vsebuje poleg aktivnih (tranzistor, dioda,...) tudi pasivne elemente (upor, kondenzator...). Integrirano vezje se po izdelavi vgradi v ohišje, tej celoti pa pravimo čip. Beseda čip izhaja iz ameriškega računalniškega slenga in pomeni silicijevo rezino. Izdelava integriranih vezij je postala mogoča, ko so odkrili, da lahko polprevodniki opravlajo enako nalogo kot vakumske cevi. Na integrirano vezje vplivata dva zelo pomembna dejavnika. To sta cena in učinkovitost. Cena je glede na količino izdelanih vezij izredno nizka, saj je možno v relativno kratkem času izdelati veliko količino IV (**I**ntegrirano **V**ezje). Drug pomemben dejavnik je učinkovitost, ki se z razvojem tehnologije povečuje. Tako postajajo tranzistorji vedno hitrejši njihova poraba se zmanjšuje samo vezje pa postaja vedno manjše. Med najbolj napredna integrirana vezja spadajo mikroprocesorji, ki jih danes najdemo v računalniku pa vse do mikrovalovne pečice.



Slika 5: Čip

Slika 6: Čip-diagram priključkov

## 

## 3.6 Kvarčni kristal



Slika 7: Kvarčni kristal

## 3.7 Triak

Triak si lahko predstavljamo kot dva vzporedno vezana tiristorja, obrnjena vsak v svojo stran. Na ta način lahko triak vključimo (ali vžgemo) v obeh smereh. Osnovna priključka sta ponekod označena kot anoda A in katoda K, drugod kot prva anoda A1in druga anoda A2. Vhodni priključek je skupen in ga imenujemo vrata G. Vključimo ga lahko tako s pozitivnim kot z negativnim napetostnim impulzom na vhodnem priključku. Ko se anodni tok IA spusti pod vrednost držalnega toka IH, se triak sam izključi. Triakov je več vrst, med njimi je npr. samo prožilni triak, ki ima na vratih vgrajen diac. Triak je po delovanju enak tiristorju, le da prevaja električni tok v obeh smereh.

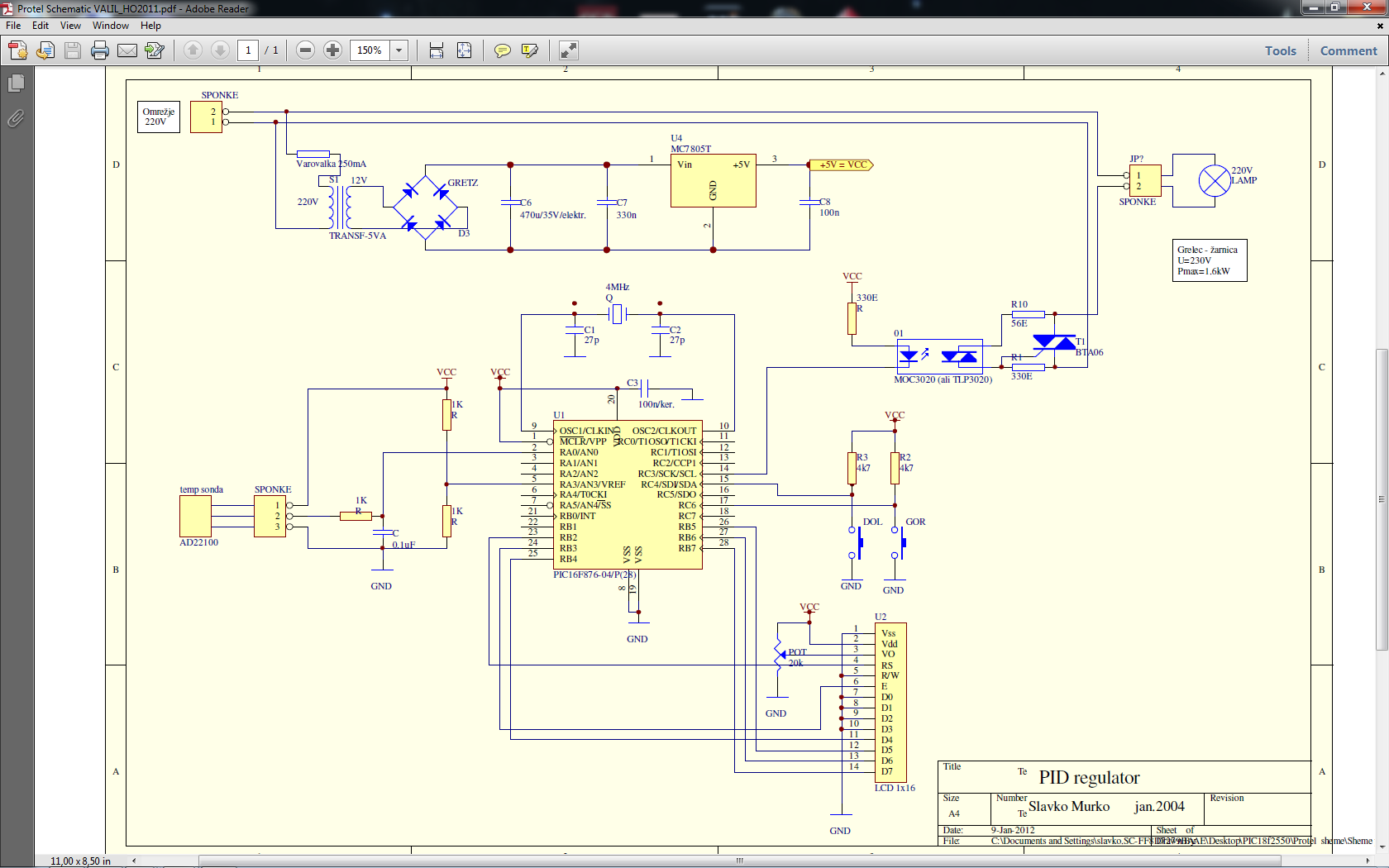


Slika 8: Triac

# 4. PRAKTIČNI DEL

## 4.1 Potek dela

Komentiranje sheme z mentorjem in nabava materiala.

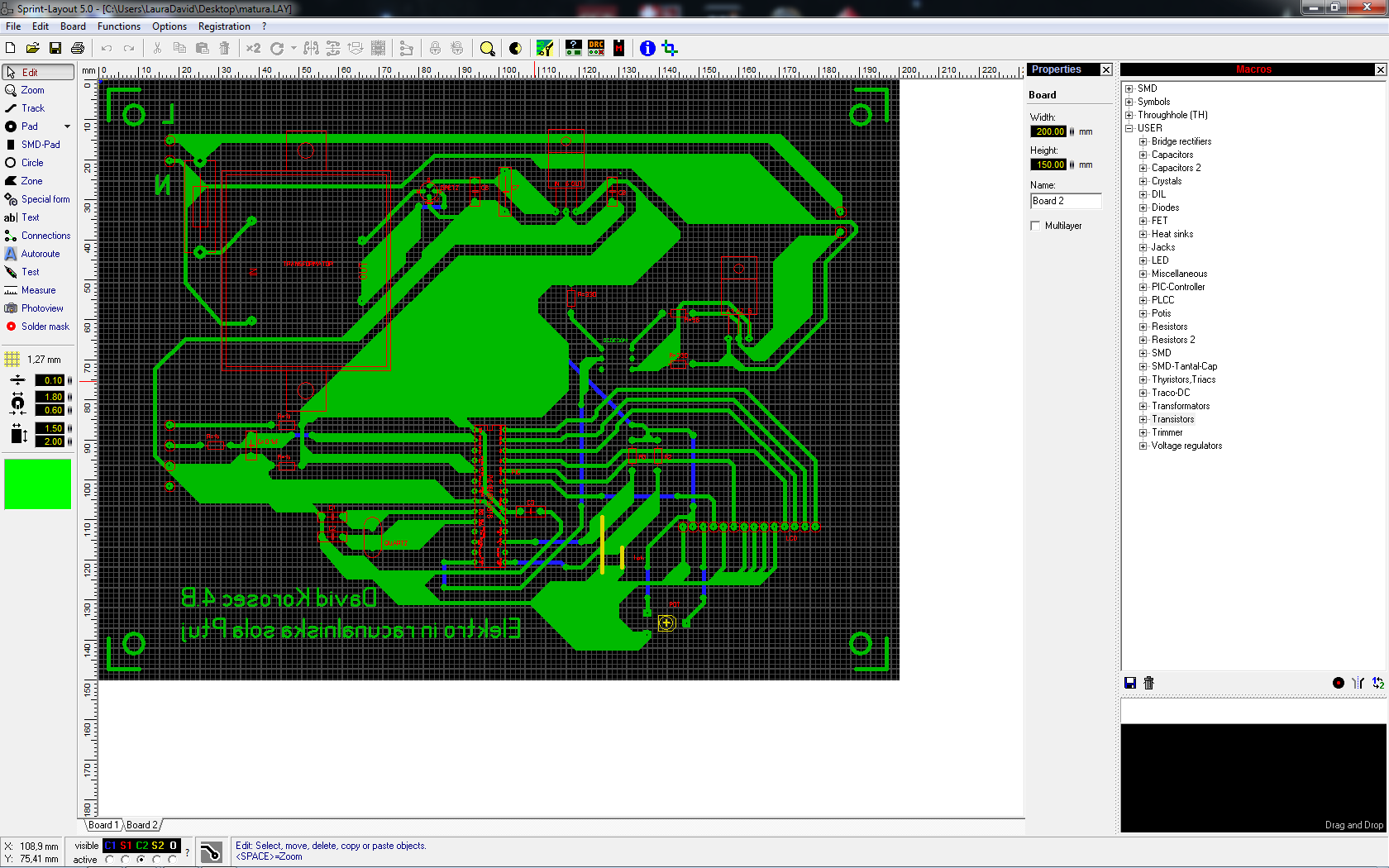


Slika 9: Shema vezja

### Nabava materiala

|  |
| --- |
| **SPONKE**  7 x MSBO2001 |
| **VAROVALKA**  1 x 250mA |
| **TRANSFORMATOR**  1 x 220/12V 5VA |
| **GRETZ**  1xW10 |
| **KONDENZATORJI**  C1=470µ/50V/elektrolitski  C2=330n/400V  C3=100n  C4,5=27p  C6=100n/keramični  C7=0,1µ |
| **OPTO**  1 x MOC3020 |
| **TRIAK**  1 x BTA06 |
| **TIPKI**  2 x |
| **POTENCIOMETER**  1 x PT10 ležeč 22k |
| **QUARTZ**  1 x 4MHz |
| **LCD**  1 x LCD 1x16 z lučko |
| **UPORI**  3 x 1k  2 x 4k7  2 x 330  1 x 56 |
| **TEMP. SONDA**  1 x PT100 |
| **ČIP**  1 x PIC16F876-04/P |
| **VITRO PLOŠČA**  1 x 150x200mm enostranska |
| **PODNOŽJE**  1 x DIL28 ozko |

### Izdelava vezja v Sprint-Layout 5.0



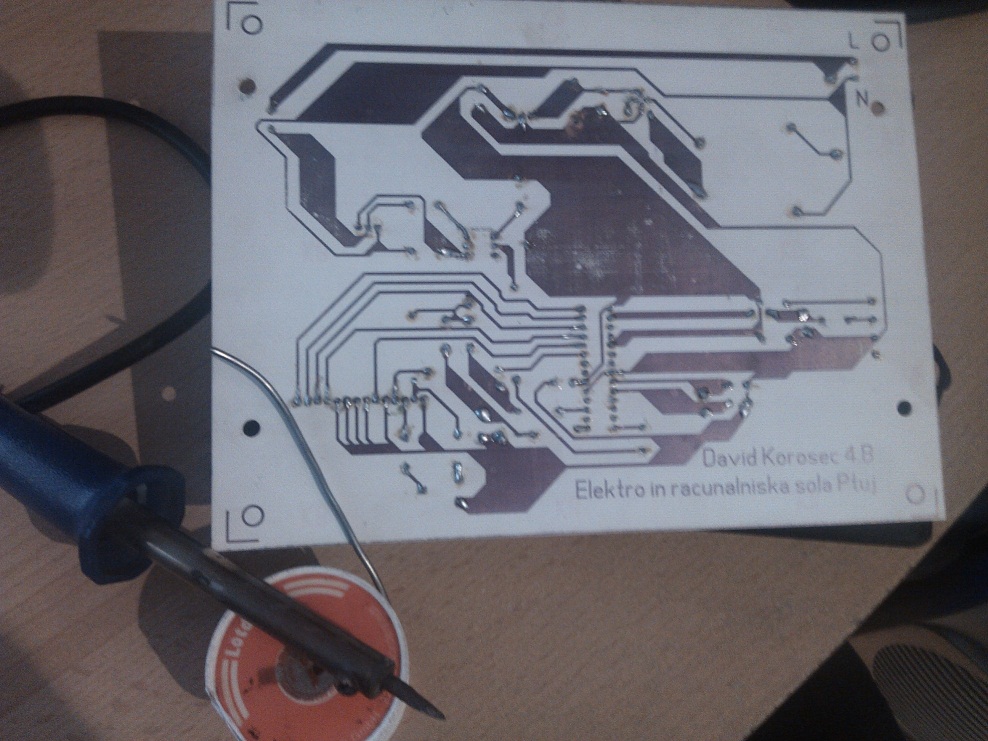
Slika 10: Vezja v Sprint-Layout

### Jedkanje



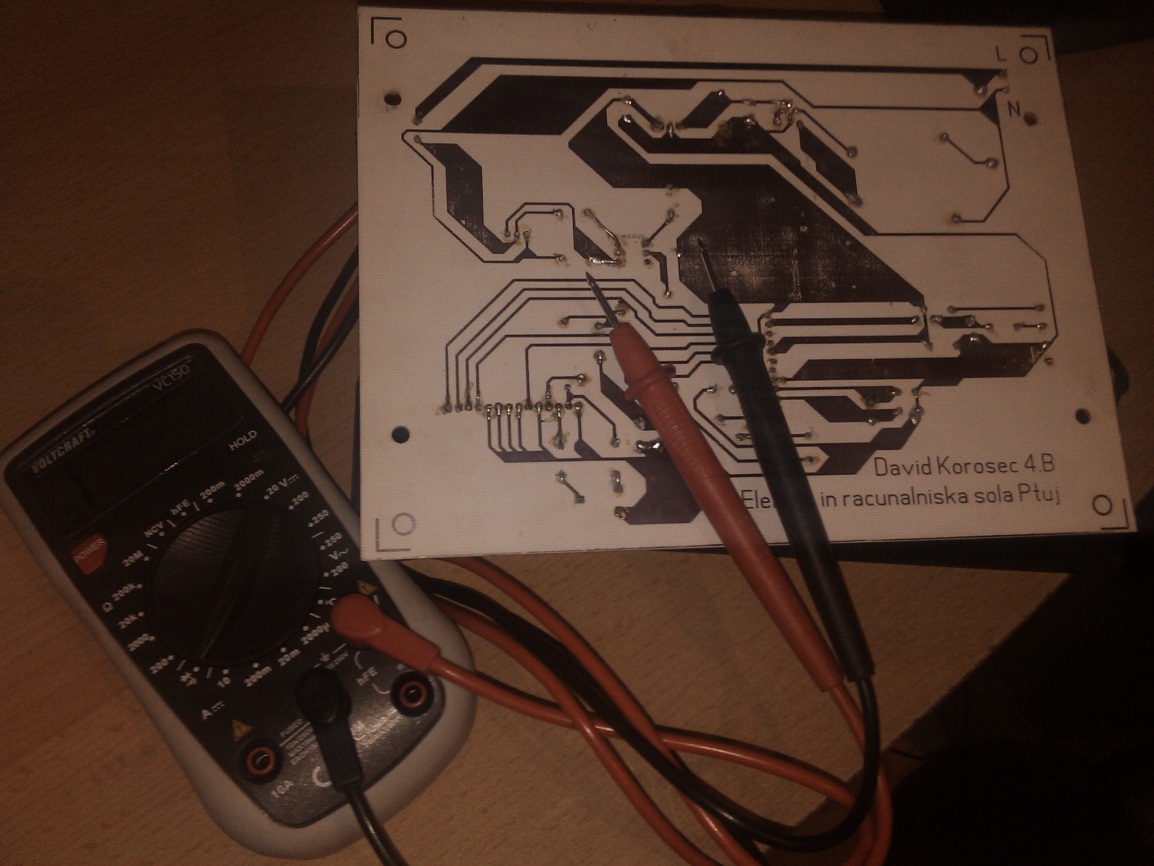
Slika 11: Jedkanje

### Spajkanje elementov na vitro ploščo



Slika 12: Spajkanje

### Meritev napetosti



Slika 13: Meritve

### Izdelava ohišja za nalogo





Slika 14, 15: Izdelava ohišja

Najprej sem prejel shemo vezja od mentorja, s katerim sva se malo pogovorila o nalogi. Če mi ni bilo kaj jasno sem ga vprašal in mi je vse razložil tako, da kasneje ni bilo problema nabaviti materiala in sheme iz papirnate oblike zrisat v digitalno. Ker ima naloga svoje 12V napajanje sem moral v vezju upoštevat še transformator, ki mi je pretvoril el. napetost iz 230V na 12V. Vezje v Sprint-Layout bi lahko naredil v malce pomanjšani obliki kot sem jo vendar sem zaradi velikega prostora na vitro plošči vse skupaj malo povečal in kasneje nisem imel problema z spajkanjem na malem prostoru.

Vezje, ki smo ga izdelali na računalniku in ga nato natisnili na folijo smo morali najprej prenesti na ploščico. Postopek imenujemo FOTOPOSTOPEK. Na voljo imamo 2 različni ploščici, lahko kupimo takšno na kateri je že fotolak lahko pa tudi brez in si ga nanesemo sami. Ko fotolak nanesemo na ploščico ga pustimo, da se posuši ali ga pa damo v pečico in vse skupaj pospešimo tako, da traja nekje 15minut. Ko se nam lak posuši ga moramo osvetlit z našim vezjem tako, da na ploščico prilepimo folijo z vezjem in jo osvetljujemo z UV svetlobo približno 2 minuti. Po osvetljevanju moramo vezje razviti in ga zato potopiti v razvijalec, ki ga kupimo. Odstranimo lak, da se nam prikaže vezje. Če se nam vezje lepo vidi lahko nadaljujemo na najnevarnejše delo pri izdelavi vezja, jedkanje. Tekočino za jedkanje sestavljajo 3 snovi: solna kislina, vodikov peroksid in voda. Najbolj moramo paziti na vodikov peroksid saj je zelo jedek in pušča madeže. Ploščico potopimo v to raztopino in počakamo da nam zjedka ves ne porisan baker. Ploščico še potem samo speremo pod vodo, da ustavimo jedkanje in jo lahko primemo.

Po jedkanju mi je ostalo samo še to, da zvrtam luknjice za elemente. Te so bile nekje Ø1mm. Elemente sem zaspajkal na ploščico in jo preizkusil, ko je bila naloga priključena na napetost sem zmeril ali so povsod takšne napetosti kot morajo biti sicer bi uničil čip. Vajo sem prvič priključil brez čipa. Po ugotovitvi, da so napetosti v redu sem vstavil še čip in na njega naložil program, ki sem dobil od profesorja. Ker je vaja delovala sem kasneje izdelal še ohišje in vse skup sestavil v obliko, ki sem si jo zamislil.

## 4.2 Delovno mesto

Večino časa za nalogo sem porabil v domači delavnici, nekaj pa tudi v šoli, ker doma nisem imel potrebnih elementov za jedkanje in za prenos programa na čip. Za jedkanje sem se dogovoril, da sem ga lahko opravil pri praksi prav tako pa tudi za zapis programa na čip, katerega sem opravil skupaj z mentorjem.

# 5. ZAKLJUČEK

Cilj, ki sem si ga zadal na začetku, sem uspešno opravil vendar sem na koncu ugotovil, da bi lahko kaj dodatno nadgradil ali boljše izdelal, kot sem. Pri vaji kakšnih velikih meritev ni bilo, da bi jih bilo potrebno zapisati. Probleme, ki sem jih odkrival tekom vaje sem reševal sproti tako, da se mi na koncu ni bilo potrebno vračati nazaj na začetek. Edino kar pa upam, da se bo spremenilo z časom, da bodo elementi na širšem trgu postali malo cenejši in bi s tem lahko nabavili tudi boljše (zdaj dražje elemente) in ne cenovno najnižjih, ki imajo slabše karakteristike kot pa seveda boljši, dražji elementi.

# 6.VIRI

[**http://www.impletum.zavod-irc.si/docs/Skriti\_dokumenti/Osnove\_elektrotehnike-Vrscaj\_Kokalj.pdf**](http://www.impletum.zavod-irc.si/docs/Skriti_dokumenti/Osnove_elektrotehnike-Vrscaj_Kokalj.pdf)

[**http://en.wikipedia.org/wiki/File:Crystal\_oscillator\_4MHz.jpg**](http://en.wikipedia.org/wiki/File:Crystal_oscillator_4MHz.jpg)

[**http://ww1.microchip.com/downloads/en/devicedoc/30292c.pdf**](http://ww1.microchip.com/downloads/en/devicedoc/30292c.pdf)

[**http://www.hiwtc.com/products/bt137-triac-590089-2576.htm**](http://www.hiwtc.com/products/bt137-triac-590089-2576.htm)

[**http://sl.wikipedia.org/wiki/Upor\_%28elektrotehnika%29**](http://sl.wikipedia.org/wiki/Upor_%28elektrotehnika%29)

[**http://sl.wikipedia.org/wiki/LCD-zaslon**](http://sl.wikipedia.org/wiki/LCD-zaslon)

[**http://sl.wikipedia.org/wiki/Integrirano\_vezje**](http://sl.wikipedia.org/wiki/Integrirano_vezje)

# 7.PRILOGA

## 7.1 Program naložen na PIC16F876-04/P

#include <16f876.H>

#device ADC=10

#use delay(clock=4000000)

#fuses XT,NOWDT,PUT,BROWNOUT,PROTECT,NOLVP

#zero\_ram

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// RUTINE ZA LCD

//-----------------------------------

struct lcd\_pin\_map { // This structure is overlayed

int unused : 2; // be pin B0.

BOOLEAN rs; //RS na B2

BOOLEAN enable; //E na B3

int data : 4; // zgornji çtirje biti so podatkovni za LCD

} lcd;

#byte lcd = 6 //PORTB

#define lcd\_type 2 // 0=5x7, 1=5x10, 2=2 lines

#define lcd\_line\_two 0x40 // LCD RAM address for the second line

byte const LCD\_INIT\_STRING[4] = {0x20 | (lcd\_type << 2), 0xc, 1, 6};

// These bytes need to be sent to the LCD

// to start it up.

//BYTE const LCD\_INIT\_STRING[4] = {0x20 | (lcd\_type << 2), 0xf, 1, 6};

//utripa in je viden

void lcd\_send\_nibble( BYTE n ) {

lcd.data = n;

delay\_cycles(1);

lcd.enable = 1;

delay\_us(2);

lcd.enable = 0;

}

void lcd\_send\_byte( BYTE address, BYTE n ) {

lcd.rs = 0;

delay\_ms(1);

lcd.rs = address;

delay\_cycles(1);

delay\_cycles(1);

lcd.enable = 0;

lcd\_send\_nibble(n >> 4);

lcd\_send\_nibble(n & 0xf);

delay\_ms(1); /\* poźakaj da se znak izpiçe \*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* L C D \_ I N I T \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

BYTE const Ini\_LCD\_znaki[8\*8] = {

0x0A, 0x04, 0x0E, 0x10, 0x10, 0x11, 0x0E, 0x00, /\*Znak 'ź',č \*/

0x0A, 0x04, 0x0F, 0x10, 0x0E, 0x01, 0x1E, 0x00, /\* Znak 'ç'š \*/

0x0A, 0x04, 0x1F, 0x02, 0x04, 0x08, 0x1F, 0x00 /\* Znak '§'ž \*/

0x02, 0x04, 0x0E, 0x10, 0x10, 0x11, 0x0E, 0x00, /\* Znak ''c' \*/

0x02, 0x07, 0x02, 0x0E, 0x12, 0x12, 0x0E, 0x00, /\* Znak 'dj' \*/

0x04, 0x0E, 0x1F, 0x04, 0x04, 0x04, 0x0E, 0x00, /\* Znak '\_' \*/

0x0E, 0x04, 0x04, 0x04, 0x1F, 0x0E, 0x04, 0x00 /\* Znak '\_' \*/

0x11, 0x1F, 0x1F, 0x1F, 0x0E, 0x04, 0x04, 0x0E /\* Kozarec (poln - źrno)\*/

};

void lcd\_init() {

BYTE i;

lcd.rs = 0;

lcd.enable = 0;

delay\_ms(100); /\* po vklopu, da se lcd pripravi \*/

for(i=1;i<=3;++i) {

lcd\_send\_nibble(3);

delay\_ms(5);

}

lcd\_send\_nibble(2);

for(i=0;i<=3;++i)

lcd\_send\_byte(0,LCD\_INIT\_STRING[i]);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* znaki źç§ -------------- \*/

lcd\_send\_byte(0,0x40); /\* set - vpisovanje uporabnisko definiranih znakov \*/

for (i=0;i<8\*8;i++) {

lcd\_send\_byte(1,Ini\_LCD\_znaki[i]); /\* poçlje definicije po vrsti \*/

}

lcd\_send\_byte(0,0x01); delay\_ms(5); /\* brisanje displeja \*/

}

void lcd\_gotoxy( int x, int y) {

int address;

if(y!=1)

address=lcd\_line\_two;

else

address=0;

address+=x-1;

lcd\_send\_byte(0,0x80|address);

}

void lcd\_putc( char c) {

switch (c) {

case '\f' : lcd\_send\_byte(0,1); /\* brisanje displeja \*/

delay\_ms(2); /\* 2ms \*/

break;

case '\n' : lcd\_gotoxy(1,2); break; //v novo vrstico

case '\r' : lcd\_gotoxy(1,1); break; //v novo vrstico

case '\b' : lcd\_send\_byte(0,0x10); break; //pomik kurzorjaza 1 levo

default : //izpis znaka

switch (c) {

/\* Slovenska + hrvaçka verzija tekstov\*/

case 'Č':

case 'č':

c = 0;

break;

case 'Š':

case 'š':

c = 1;

break;

case 'Ž':

case 'ž':

c = 2;

break;

case 'Ć':

case 'ć':

c = 3;

break;

case 'Đ':

case 'đ':

c = 4;

break;

}

lcd\_send\_byte(1,c);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// KONEC RUTIN ZA LCD

//-----------------------------------

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// RUTINE ZA EEPROM

//-----------------------------------

//--------------

//lokacije v EE

#define SPOMIN1 0 //ciljna temp

#define SPOMIN2 4 //offset sonde

#define SPOMIN3 8 //interval obračanja

void Write\_2bajta\_eeprom(int ee\_lokacija,long stevilo\_tmp) {

write\_eeprom(ee\_lokacija,(BYTE) (stevilo\_tmp));

write\_eeprom(++ee\_lokacija,(BYTE) (stevilo\_tmp >> 8));

}

long Read\_2bajta\_eeprom(int ee\_lokacija) {

long stevilo\_tmp;

stevilo\_tmp = read\_eeprom(ee\_lokacija+1);

stevilo\_tmp<<=8; //šifta levo za osem mest

stevilo\_tmp += read\_eeprom(ee\_lokacija);

return (stevilo\_tmp);

}

void Write\_float\_eeprom(int ee\_lokacija,float stevilo\_tmp) {

write\_eeprom(ee\_lokacija,\*(int)(&stevilo\_tmp));

write\_eeprom(ee\_lokacija+1,\*(int)(&stevilo\_tmp+1));

write\_eeprom(ee\_lokacija+2,\*(int)(&stevilo\_tmp+2));

write\_eeprom(ee\_lokacija+3,\*(int)(&stevilo\_tmp+3));

}

float Read\_float\_eeprom(int ee\_lokacija) {

float stevilo\_tmp;

\*(int)(&stevilo\_tmp) = read\_eeprom(ee\_lokacija);

\*(int)(&stevilo\_tmp+1) = read\_eeprom(ee\_lokacija+1);

\*(int)(&stevilo\_tmp+2) = read\_eeprom(ee\_lokacija+2);

\*(int)(&stevilo\_tmp+3) = read\_eeprom(ee\_lokacija+3);

return (stevilo\_tmp);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// KONEC RUTIN ZA EEPROM

//-----------------------------------

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// RUTINE ZA TIPKE

//-----------------------------------

#define TIPKA\_GOR PIN\_C4

#define TIPKA\_DOL PIN\_C6

int Ti\_sp\_gor,Ti\_sp\_dol;

int Timer\_tipka\_spuscena,Timer\_tipka\_pritisnjena;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* S C A N K E Y \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define ZAPISI\_V\_EE 30 /\*desetink sekunde\*/

#define ZACNI\_HITRI\_INKREMENT 10 /\*desetink sekunde\*/

#define \_PLUS\_ 28

#define \_MINUS\_ 8

#define \_ESC\_ 27

BYTE Scankey(void) {

BYTE tmp = 0;

if (!Ti\_sp\_gor && Ti\_sp\_dol)

tmp =\_PLUS\_;

else if (Ti\_sp\_gor && !Ti\_sp\_dol)

tmp =\_MINUS\_;

else if (!Ti\_sp\_gor && !Ti\_sp\_dol)

tmp =\_ESC\_;

if (tmp) Timer\_tipka\_spuscena = ZAPISI\_V\_EE;

else Timer\_tipka\_pritisnjena = ZACNI\_HITRI\_INKREMENT;

return(tmp);

}

BYTE Newkey(void) {

static BYTE old\_scan;

BYTE new\_scan;

new\_scan = Scankey();

if (old\_scan == new\_scan) new\_scan=0;

else old\_scan = new\_scan;

return(new\_scan);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// KONEC RUTIN ZA TIPKE

//-----------------------------------

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// RUTINE ZA AD

//-----------------------------------

float ad\_offset;

long read\_adc\_30x() {

long vsota,povprecje;

int i;

vsota=0;

for(i=0;i<30;++i) {

delay\_ms(6);

vsota += read\_adc();

}

povprecje =vsota/30;

return(povprecje);

}

//za temp.sondo ad22100

float iz\_ad\_v\_stopinje\_celzija(long ad\_vrednost) {

float stopinje\_c;

stopinje\_c=((((float)ad\_vrednost+ad\_offset)\*100.0)/1024-50.0);

//printf(lcd\_putc,"\n%2.2f;%ld;%f; ",stopinje\_c,ad\_vrednost,ad\_offset);

return(stopinje\_c);

}

float meri\_stopinje\_c() {

return(iz\_ad\_v\_stopinje\_celzija(read\_adc\_30x()));

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// KONEC RUTIN ZA AD

//-----------------------------------

int moc\_grelci;

int Timer\_regulator;

//int timer\_50ms;

//GLOBALNE SPREMENLJIVKE

int Timer\_desetink\_sekunde; //globalna spremenljivka

int32 T2ure\_dessek; //šteje dve uri

#define C2URE 10\*60\*60\*2 //dve uri v desetinkah sekunde

int32 T24ur\_dessek; //šteje 24 ur

#define C24UR 10\*60\*60\*24 //24 ur v desetinkah sekunde T15min\_dessek

int16 T15min\_dessek; //šteje 15 min

#define C15MIN 10\*60\*15 //15 min v desetinkah sekunde

int16 T3min\_dessek; //šteje 3 min

#define C3MIN 10\*60\*3 //3 min v desetinkah sekunde

/\*\*\*\*\*\*\*\*\*\*\*/

/\* M A I N \*/

/\*\*\*\*\*\*\*\*\*\*\*/

main(void) { /\* začetek programa \*/

int i;

int tipka,tipka\_last;

// int a,b,c,d;

int flag\_st\_cilj\_spremenjena;

int flag\_po\_resetu=1,flag\_nastavljam\_offset=0,flag\_offset\_spremenjen;

int flag\_obracam\_rocno=0,flag\_motor\_levo=0, flag\_15min\_hladim\_jajce=0;

float stopinje\_c\_merjeno,stopinje\_c\_cilj,epsilon;

float izhod\_regulatorja;

float izhod\_proporcionalnega\_reg;

float izhod\_integralnega\_reg;

int flag\_za\_utripanje;

int flag\_sonda\_ni\_ok;

//INICIALIZACIJA

//ANALOGNO DIGITALNI PRETVORNIK

set\_tris\_a(0b11111111);

setup\_adc\_ports( RA0\_RA1\_ANALOG\_RA3\_REF );

setup\_adc( ADC\_CLOCK\_INTERNAL );

set\_adc\_channel( 0 );

//DISPLEJ

set\_tris\_b(0b00000011);

lcd\_init();

//PORT C

set\_tris\_c(0b11010000);

#define MOTOR\_DESNO PIN\_C0

#define MOTOR\_LEVO PIN\_C1

setup\_timer\_0 (RTCC\_INTERNAL);

setup\_wdt(WDT\_2304MS);

enable\_interrupts(INT\_TIMER0);

enable\_interrupts(GLOBAL);

printf(lcd\_putc,"\rREGULACI\nJA TEMP.");

delay\_ms(700);//da preberem napis

stopinje\_c\_cilj = Read\_float\_eeprom(SPOMIN1);

if (stopinje\_c\_cilj > 140.0 || stopinje\_c\_cilj < -40) stopinje\_c\_cilj=25.0;

ad\_offset = Read\_float\_eeprom(SPOMIN2);

if (ad\_offset > 100.0 || ad\_offset < -100) ad\_offset=0;

T24ur\_dessek=C24UR;

while (1) { /\* v neskončni zanki \*/

restart\_wdt();

tipka = newkey(); //ko prvič pritisnem

if (tipka) tipka\_last=tipka; //zapomni si jo za avtorepeat

//ce ob resetu drzim obe tipki skocim v nastavljanje offseta

if (flag\_po\_resetu && tipka==\_ESC\_) flag\_nastavljam\_offset=1;

//ce kdaj koli pritisnem obe tipki skocim v obračanje pladnov ročno s tipkami +,-

else if (tipka==\_ESC\_ && flag\_obracam\_rocno==0) {flag\_obracam\_rocno=1;tipka=0;} /\*tipka=0, da ne prime ESC se enkrat in takoj zaključi obracanje\*/

flag\_po\_resetu=0;

//VRINJENI DEL-----------------------------------------za nast. offseta

if (flag\_nastavljam\_offset==1) { //prime nastavlanje offseta

if (tipka==\_PLUS\_) {

if (ad\_offset<=100) { //max +100 integer vrednosti AD

ad\_offset=ad\_offset+1;

flag\_offset\_spremenjen=1;

}

}

if (tipka==\_MINUS\_) {

if (ad\_offset>=-100) { //min -100 integ. iz ad

ad\_offset=ad\_offset-1;

flag\_offset\_spremenjen=1;

}

}

if (Timer\_tipka\_spuscena==0) {

flag\_nastavljam\_offset=0;

if(flag\_offset\_spremenjen == 1) {

Write\_float\_eeprom(SPOMIN2,ad\_offset);

flag\_offset\_spremenjen=0;

}

}

}

else if (flag\_obracam\_rocno==1) { //prime obračanje pladnov ročno

if (scankey()==\_PLUS\_) {

// if (Timer\_tipka\_pritisnjena == 0) { //po 1 s

output\_high(MOTOR\_DESNO); //vklopi motor desno

// }

} else {output\_low(MOTOR\_DESNO);} //izklopi motor desno

if (scankey()==\_MINUS\_) {

// if (Timer\_tipka\_pritisnjena == 0) { //po 1 s

output\_high(MOTOR\_LEVO);//vklopi motor levo

// }

} else {output\_low(MOTOR\_LEVO);} //izklopi motor levo

// if (Timer\_tipka\_spuscena==0) { //po 3 sekundah izklopi to akcijo

// flag\_obracam\_rocno=0; //SE SEDAJ VRNE REGULATRO V NORMALNO DELOVANJE, PRAVILNO PA MORA IZI V NOVO AKCIJO "STOP" IZ KATER LAHKO NADALJUJEŠ Z OBRAČANJE (-) ALI NADALJUJEŠ Z REGULACIJO(+)

// }

if (tipka == \_ESC\_) { //po ponovnem pritisku obeh tipk se regulator spet vklopi

output\_low(MOTOR\_DESNO);

output\_low(MOTOR\_LEVO);

flag\_obracam\_rocno=0;

}

// goto izklopi\_regulator;

}

else { //prime nastavljanje ciljne temp

//konec VRINJENI DEL-----------------------------------------

if (Timer\_tipka\_pritisnjena == 0 && flag\_st\_cilj\_spremenjena == 1 && scankey()) { //ko držim več kot 1s

tipka = tipka\_last; //tipko že držim eno sekundo-newkey vrača 0

//inkrement pa vseeno prime in to avtorepeat

}

if (tipka==\_PLUS\_) {

if (stopinje\_c\_cilj<=49.0) { //max 50 stopinj C

stopinje\_c\_cilj=stopinje\_c\_cilj+0.1;

flag\_st\_cilj\_spremenjena=1;

}

}

if (tipka==\_MINUS\_) {

if (stopinje\_c\_cilj>=-19.0) { //min -20 st C

stopinje\_c\_cilj=stopinje\_c\_cilj-0.1;

flag\_st\_cilj\_spremenjena=1;

}

}

if (Timer\_tipka\_spuscena==0 && flag\_st\_cilj\_spremenjena == 1) {

// delay\_ms(500);

Write\_float\_eeprom(SPOMIN1,stopinje\_c\_cilj);

flag\_st\_cilj\_spremenjena=0;

}

/\* if (Timer\_tipka\_pritisnjena ==0 && timer\_50ms == 0) { //ko držim več kot 1s in je od prejšnjega inkr minilo več kot 50ms

tipka = scankey();

timer\_50ms = 25; //po 2ms

}

\*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//regulator PI

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define CAS\_ITERACIJE 10 /\* 10\*0.1s=1s\*/

if (Timer\_regulator == 0) {

//EPSILON

stopinje\_c\_merjeno = meri\_stopinje\_c();

epsilon=stopinje\_c\_cilj - stopinje\_c\_merjeno;

#define Kp 40.0

izhod\_proporcionalnega\_reg=epsilon\*Kp;

#define Ki 0.1

izhod\_integralnega\_reg = izhod\_integralnega\_reg + Ki\*epsilon;

//stopinje izven območja -20C do +50C - SONDA NI OK - regulator na 0

flag\_sonda\_ni\_ok = 0;

if (stopinje\_c\_merjeno>49.5 || stopinje\_c\_merjeno < -20.0) {

izhod\_proporcionalnega\_reg = 0;

izhod\_integralnega\_reg = 0;

flag\_sonda\_ni\_ok=1;

}

if (flag\_obracam\_rocno==1 || flag\_15min\_hladim\_jajce ==1) { //izklopi regulator, tudi grelci na 0

izhod\_proporcionalnega\_reg = 0;

izhod\_integralnega\_reg = 0;

}

//omejitev izhoda integralnega 0-100

if( izhod\_integralnega\_reg >100.0) {

izhod\_integralnega\_reg=100.0; //0-100%

}

else if (izhod\_integralnega\_reg < 0.0) {

izhod\_integralnega\_reg=0.0; //0-100%

}

izhod\_regulatorja=izhod\_proporcionalnega\_reg + izhod\_integralnega\_reg; //skupni izhod

//omejitev izhoda 0-100

if (izhod\_regulatorja > 100.0) {

izhod\_regulatorja=100.0; //0-100%

}

else if (izhod\_regulatorja < 0) {

izhod\_regulatorja=0.0; //0-100%

}

moc\_grelci=(int)izhod\_regulatorja;

Timer\_regulator=CAS\_ITERACIJE;

}

prvo\_vrsto\_pisi\_nornalno: printf(lcd\_putc,"\rT=%2.1fC;\n X=%2.1fC ",stopinje\_c\_merjeno,stopinje\_c\_cilj);

// }

// }

} /\* konec while zanke \*/

} /\* konec funkcije main \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* P R E K I N I T E V \_ T I M E R 0 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define PWM\_TRIAC PIN\_C3

long interrupt\_counter;

int pwm\_counter;

int hec,deli\_irq=3;

#INT\_RTCC

void Prekinitev\_timer0(void) {

// KVARC == 4 vsakih 1us\*256 = 256us se izvaja rutina

if (--deli\_irq==0) { //da deluje pri 12Mhz kvarzu enkako hitro kot je pri 4Mhz

deli\_irq=3;

//PREBERI TIPKE

if(--hec==0) { // vsakih 12\*256us=3ms

shift\_left(&Ti\_sp\_gor,1,input(TIPKA\_GOR));

shift\_left(&Ti\_sp\_dol,1,input(TIPKA\_DOL));

hec=12;

}

//TIMERJI ZA PROCESE

if (interrupt\_counter==0) {// vsakih 100ms

if(Timer\_tipka\_spuscena!=0) Timer\_tipka\_spuscena--;// 0.1 sek

if(Timer\_tipka\_pritisnjena!=0) Timer\_tipka\_pritisnjena--;// 0.1 sek

if(Timer\_regulator!=0) Timer\_regulator--;// 0.1 sek

if(Timer\_desetink\_sekunde!=0) Timer\_desetink\_sekunde--;// 0.1 sek

if(T2ure\_dessek!=0) T2ure\_dessek--;// 0.1 sek

if(T24ur\_dessek!=0) T24ur\_dessek--;// 0.1 sek

if(T15min\_dessek!=0) T15min\_dessek--;// 0.1 sek

if(T3min\_dessek!=0) T3min\_dessek--;// 0.1 sek

interrupt\_counter=390; //390\*256us je 99.84ms = 1/10 sek

}

else interrupt\_counter--;

//PWM

if (++pwm\_counter == 101) pwm\_counter=1; //preskoči 0 ,da pri moc\_grelci==0 sploh ne vklopi

if (pwm\_counter <= moc\_grelci) { //generiraj impulz

output\_low(PWM\_TRIAC); //negiran izhod

}

else { //generiraj pavzo

output\_high(PWM\_TRIAC); //negiran izhod

}

} //konec if(deli)

}