



KAZALO

1 UVOD.....	6
2 OPIS GOSPODARSKE DRUŽBE IN PRODUKTOV	7
3 OPIS PRAKTIČNEGA IZOBRAŽEVANJA – PROJEKTNEGA DELA ŠTUDENTA.....	9
4 MIKROPROCESOR, MIKRORAČUNALNIK, MIKROKONTROLER	10
4.1 Vrste mikrokontrolerjev	10
4.2 Delovanje mikrokontrolerjev	11
4.3 Izvajanje programa	11
4.4 Priključevanje vhodno/izhodnih enot	12
5 ATMEL AVR ATMEGA8.....	13
5.1 Nekaj pomembnejših tehničnih podatkov mikrokontrolerja AVR ATmega8	13
5.2 Razvrstitev nogic »pinov« na mikrokontrolerju ki je v PDIP ohišju.....	14
5.3 Blok shema mikrokontrolerja	15
6 ORODJA ZA PROGRAMIRANJE MIKROKONTROLERJEV	16
6.1 Programatorji	16
6.2 ISP programator	17
6.3 Izdelava ISP programatorja	19
7 PROGRAM ZA MERJENJE IN URAVNAVANJE TEMPERATURE	24
7.1 Merilna sonda DS1820	25
7.2 LCD Prikazovalnik	27
7.3 Delovanje programa in program	29
8 SKLEP	34
9 PRILOGE.....	35
9.1 Celoten program	35



KAZALO SLIK

Slika 1: PDIP Ohišje mikrokontrolerja AVR ATmega8	13
Slika 2: Razvrstitev pinov na mikrokontrolerju	14
Slika 3: Blok shema mikrokontrolerja.....	15
Slika 4: Povezava med programatorjem in mikrokrmlnikom v vezju.....	17
Slika 5: Sponke in signali konektorja za ISP programator.....	19
Slika 6: Načrt AVR-ISP programatorja.....	20
Slika 7: Shema za tiskanino AVR-ISP Programatorja	20
Slika 8: material, potreben za AVR-ISP programator.....	21
Slika 9: Izdelan programator (sprednja stran)	22
Slika 10: Izdelan programator (spodnja stran)	22
Slika 11: Adapter za ATmega8 mikrokontroler	23
Slika 12: Vezalna shema za merjenje in uravnavanje temperature	24
Slika 13: Ohišje TO-92 in SO meritve sonde DS1820.....	26
Slika 14: Moja sonda DS1820	26
Slika 15: Format zapisa 1-Wire protokola.....	27
Slika 16: Moj LCD prikazovalnik (sprednja stran)	28
Slika 17: Moj LCD Prikazovalnik (zadnja stran)	28
Slika 18: Prikaz delovanja programa.....	29
Slika 19: Kratek program za testiranje mikrokontrolerja	30
Slika 20: Inicializacija in definiranje spremenljivk	31
Slika 21: Definiranje Ledic, LCD zaslona in temperaturnega senzorja	31
Slika 22: Delovanje tipk z omejitvijo spodnje in zgornje meje.....	32
Slika 23: Branje temperature in izpis	33
Slika 24: Delovanje izhoda. (LED diodic)	33



1 UVOD

Po končanem predavanju na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru sem začel izvajati obvezno praktično usposabljanje pri podjetju Audiologs, Milenko Glavica, s.p. Praktično usposabljanje sem izvajal 6 mesecev in sicer od 2.11.2009 do 2.5.2010. Podjetje Audiologs se ukvarja s popravili elektronskih glasbil, studijske opreme, opreme za ozvočenje, DJ opreme in ozvočenjem pa tudi z razvojem in izdelavo elektronskih naprav. To so predvsem elektronske naprave, kot so razni efekti, MIDI playerji, (z možnostjo priklopa na inštrumente), razne razvojne plošče itd. Pri praktičnem izobraževanju sem bolje spoznal mikrokontrolerje AVR družine (delovanje, integriranje v sisteme in navsezadnje tudi programiranje v zbirniku BASCOM AVR s programskim jezikom BASCOM).

2 OPIS GOSPODARSKE DRUŽBE IN PRODUKTOV

Podjetje Audiologs s.p. je bilo ustanovljeno leta 2001 in se nahaja na Jadranski cesti 27 v Mariboru. Ustanovitelj podjetja Audiologs je gospod Milenko Glavica.

Milenko Glavica:

Začetki našega podjetja segajo v leto 2001, z osnovno usmeritvijo v profesionalno avdio tehniko. Sprva je večino dejavnosti predstavljal vzdrževanje profesionalne avdio opreme ter načrtovanje sistemov za ozvočenje, snemanje in predvajanje zvoka.

Po letih uspešnega delovanja smo združili svoja strokovna znanja in bogate izkušnje ter začeli z razvojem lastnih produktov oz. polizdelkov tako na področju profesionalne avdio tehnike kot tudi drugih področjih digitalne tehnologije. Prepričali smo se, da lahko, predvsem na področju visokotehnoloških znanj ter izvirnih rešitev, mala podjetja enakovredno stopajo v korak z večjimi.

Prvi večji projekt, s katerim smo prestopili na področje digitalnih komunikacij, predstavljajo multikanalni ojačevalci s stikalnim usmernikom. Uporabni so za ozvočevanje predavanj, konferenčne sisteme in sestavljeni sistemi ambientalnega zvoka, skratka povsod tam, kjer je kvaliteta zvočne in slikovne komunikacije vitalnega pomena.

V sodelovanju s podjetji, ki so za naše mobilne aplikacije razvili stikalne avdio usmernike, ki jih odlikujejo nizka teža, visok izkoristek (95%), PFC in ničelni šum. Večina končnih stopenj sodi v visoki D oz. digital D razred, vezja pa uravnavajo mirovni tok, kot odpiranja tranzistorjev, moč..itd.

Na ta način smo dosegli visoko efektivnost ozvočevalnih sistemov, ki jih z lahkoto ustrezno prilagodimo glede na želje in potrebe individualnih naročnikov.

DSP-je kreiramo za dodatek profesionalni snemalni tehniki. S prodorom mikrokontrolerjev in FPGA vezij je postal razvoj teh komponent zelo hiter in enostaven. Zelo dodelan softwear nam omogoča razvoj komponent na modelih, v grafičnem okolju ali na posebnih platformah.

Naši multipredvajalniki so projektirani za zahtevnejšo avdio produkcijo. To so predvajalniki za reprodukcijo vseh znanih formatov mid, wav, mp3 (mpeg kodiranje). Primerni so za poljubno vgradnjo, za samostojne predvajalnike in za arhiviranje avdio datotek, podatkov, snemanje in editiranje.

Digitalni reverb, Revecon (DRE tehnologija), je naš najnovejši izdelek s področja avdio tehnologije. Uporabljamo ga kot dodatek osnovnemu signalu za vokalni efekt in kot efekt za instrumente. Dodan je optični in koaksialni vmesnik za digitalno produkcijo S/PDIF standard, v višjih serijah še USB 2.0 serijski prenos. 24 bitov/192KHz in 32 bitno procesiranje zagotavlja perfekcijo zvoka, ki se je do nedavnega zdela nedosegljiva.

Na področju informacijske tehnologije razvijamo projekt zaslon na dotik, ki je zamišljen tako, da bi podpiral določene funkcije z RFID karticami in omogočal dostop do zaupnih informacij, polnjenje računov, internetni dostop ipd.

Poleg naštetih pa v zakladnici naših idej zori še mnogo projektov in čaka na primeren trenutek oz. ustrezne pogoje za njihovo realizacijo.

3 OPIS PRAKTIČNEGA IZOBRAŽEVANJA – PROJEKTNEGA DELA ŠTUDENTA

Pri praktičnem izobraževanju sem se najprej seznanil z mikrokontrolerji, mikroračunalniki in mikroprocesorji. Bolj podrobno sem spoznal vrste mikrokontrolerjev in delovanje mikrokontrolerjev. Podrobneje sem obdelal Atmelov mikrokontroler AVR ATmega8, saj sem ga tudi pozneje uporabil za izvedbo projekta. Nato sem naredil izdelek in sicer programator za programiranje Atmel AVR mikrokontrolerjev AVR ISP programmer (STK 200). Ko sem končal s programatorjem, pa sem se lotil pisanja programa za merjenje in uravnavanje temperature v programu BASCOM.

4 MIKROPROCESOR, MIKRORAČUNALNIK, MIKROKONTROLER

Mikroprocesor je centralna procesna enota (CPU) računalnika, zgrajena kot integrirano vezje. Če mikroprocesorju dodamo še V/I (vhodno izhodne) enote, pomnilnik in nekaj integriranih vezij, ki povezuje omenjene dele v sistem, dobimo mikroračunalnik. Mikroračunalnik je torej računalnik, čigar CPU je mikroprocesor. Mikroprocesor je kot samostojni element neuporaben. Moramo mu dodati se druge elemente, ki skupaj tvorijo mikroračunalnik.

Dejstvo, da dobimo mikroračunalnik, če mikroprocesorju dodamo pomnilnik in V/I enote, velja tako tudi na nivoju čipa. Mnogi proizvajalci realizirajo na istem silicijevem substratu poleg CPE (mikroprocesorja) še druge enote, ki tvorijo mikroračunalnik - takšnemu elementu pravimo mikrokontroler. Mikrokontroler lahko torej definiramo kot mikroračunalnik, ki je zgrajen kot eno integrirano vezje (single chip microcomputer) in deluje samostojno.

4.1 Vrste mikrokontrolerjev

4-bitni mikrokontrolerji se uporabljajo v velikoserijskih proizvodih, kjer mora biti cena mikrokontrolerja nizka. Namenjeni so predvsem uporabi manj zahtevnim aplikacijam.

8-bitni mikrokontrolerji so danes zelo razširjeni, saj nam nudijo široke uporabne možnosti ob relativno nizki ceni. Še posebej so primerni za uporabo v maloserijskih proizvodih in prototipih. Tudi razvojna orodja so zelo razširjena in poceni, kot osnova za razvoj zahtevajo osebni računalnik.

16- in 32-bitni mikrokontrolerji v zadnjem času proizvajalci pospešeno razvijajo 16-bitne družine mikrokontrolerjev in izjemno zmogljive 32-bitne mikrokontrolerje. Oboji so namenjeni zahtevnejšim aplikacijam in so tudi dražji.

4.2 Delovanje mikrokontrolerjev

Delovanje računalnika je zaporedno izvajanje dveh faz: prevzema ukaza iz pomnilnika in izvršitev prevzetega ukaza. Pri prevzemu ukaza CPE(centralna procesna enota) najprej prebere operacijsko kodo (iz lokacije na katero kaže v tistem trenutku PC (programski števec)) v ukazni register krmilne enote. V naslovni register se vpiše vrednost PC in s tem se pojavi naslov ukaza na naslovnih linijah. CPE nato pričakuje vsebino ukaza na podatkovnih linijah. Preko podatkovnega registra se nato operacijska koda prenese v ukazni register krmilne enote.

Od tega trenutka dalje preide računalnik v fazo izvršitve ukaza (execute). Krmilna enota dekodira prevzeti ukaz in sproži krmilne signale, ki omogočijo izvršitev ukaza. V času izvajanja ukaza se tudi poveča vsebina PC. CPE izvrši operacijo in poskrbi, da je po končanem izvrševanju v PC naslov naslednjega ukaza. Pri tem upošteva pravilo, da so ukazi v pomnilniku shranjeni po naraščajočih naslovih - to pravilo se običajno ponazarja z izrazom $PC < PC + 1$. Izjema so tako imenovani skočni ukazi, s katerimi lahko v PC zapišemo poljuben naslov.

4.3 Izvajanje programa

Vsaka CPE ob zagonu (reset signal) začne na enak način. Prvi ukaz lahko prevzame v točno določeni lokaciji oziroma v točno določeni lokaciji prebere naslov prvega ukaza. Vrednost, ki se ob zagonu vpiše v PC, je ob vsakem zagonu enaka; od tod naprej se prevzemajo ukazi zaporedno, kot to narekuje vrednost PC - ja.

4.4 Priključevanje vhodno/izhodnih enot

Pri današnjih mikrokontrolerjih sta v uporabi 3. načini priključevanja V/I enot:

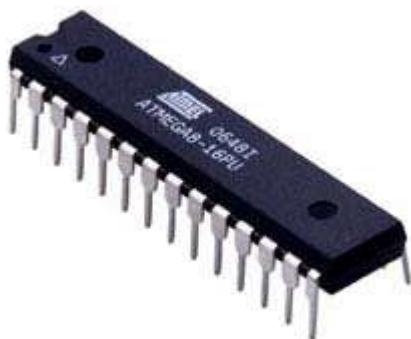
1.način: Pomnilniško preslikan vhod-izhod (memory mapped I/O). Pri tem načinu so registri krmilnikov v pomnilniškem naslovnem prostoru. Gledano iz CPE so videti enako kot pomnilniške besede. Za branje in pisanje lahko uporabimo vse ukaze za dostop do pomnilnika. Posebni V/I ukazi niso potrebni.

2.način: Ločen V/I prostor. Pri tem načinu so registri krmilnikov v posebnem naslovnem prostoru, ki je ločen od pomnilniškega. Za dostop do registrov so potrebni posebni V/I ukazi. Med izvajanjem teh ukazov cpe aktivira signal (ali signale), ki pove, da se naslavlja V/I naslovni prostor.

3.način: Posredno prek V/I procesorjev. Te rešitve pri mikroprocesorjih skoraj ne srečamo, pogosta pa je pri večjih računalnikih. Pri tem CPE nima dostopa do registrov krmilnika. Namesto tega sporoča svoje zahteve V/I procesorjem, ki imajo dosotop do registrov in ki poskrbijo za podrobnosti pri izvrševanju zahtev.

5 ATMEL AVR ATMEGA8

Pri praktičnem izobraževanju sem uporabljal AVR-jev mikrokontroler AVR ATmega8. Ta mikrokontroler je bil izdelan v treh različnih ohišjih. Najbolj pogosto uporabljeno je ohišje PDIP, obstaja pa še TQFP in MLF izvedba ki sta namenjeni bolj za uporabo v integriranih vezjih.



Slika 1: PDIP Ohišje mikrokontrolerja AVR ATmega8

5.1 Nekaj pomembnejših tehničnih podatkov mikrokontrolerja AVR ATmega8

- 8 kilobyтов »in-system« programirljivega FLASH pomnilnika
- 512 bytov EEPROM pomnilnika
- 1 kilobyt internega SRAM-a
- 23 V/I programirljivih linij
- 10000 možnih vpisov/izbrisov FLASH pomnilnika in EEPROM pomnilnika
- dva 8 bitna števca
- en 16 bitni števec
- napajanje od 4,5 do 5,5V
- interni kalibriran RC oscillator



5.2 Razvrstitev nogic »pinov« na mikrokontrolerju ki je v PDIP ohišju

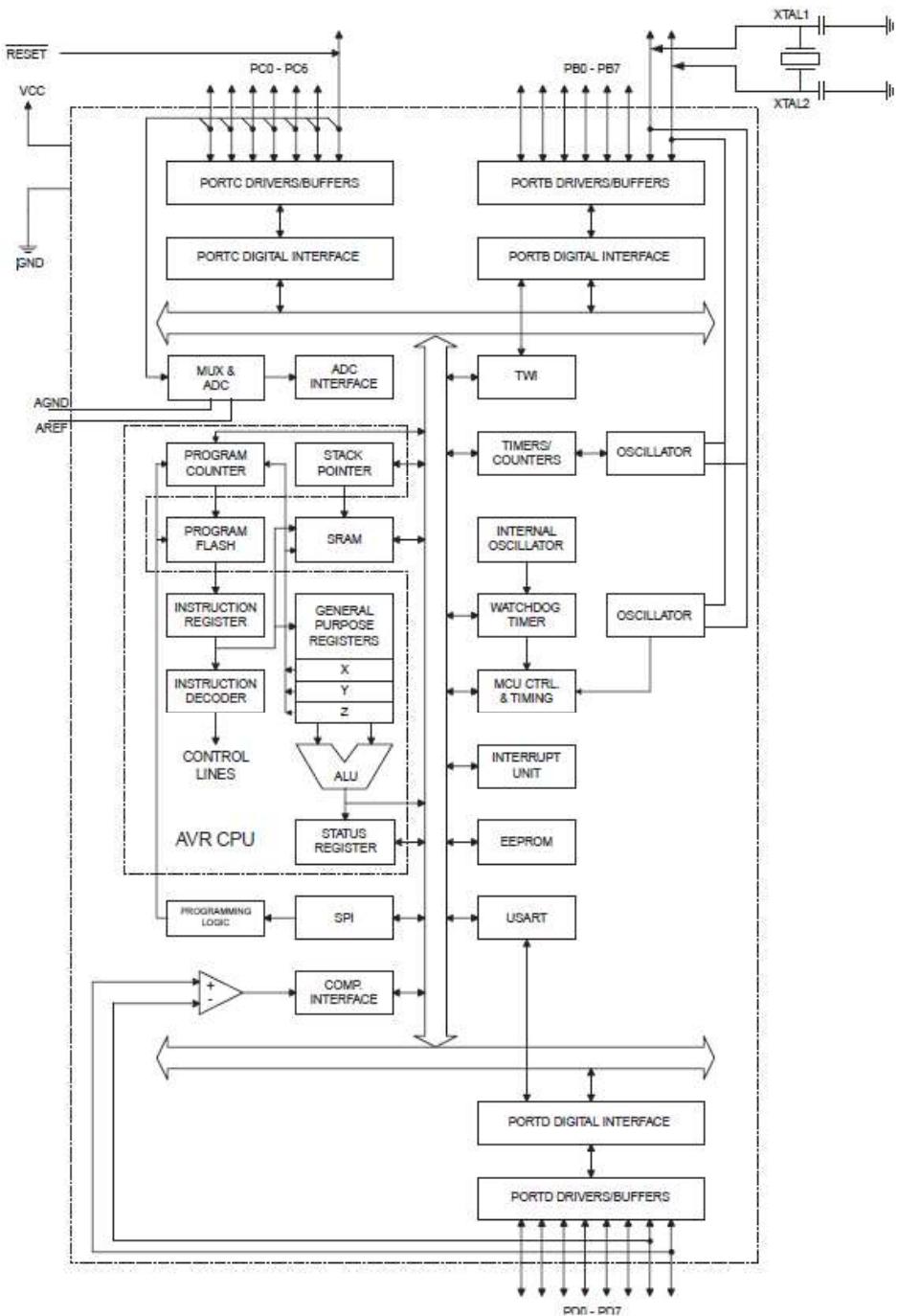
PDIP	
(RESET) PC6	1
(RXD) PD0	2
(TXD) PD1	3
(INT0) PD2	4
(INT1) PD3	5
(XCK/T0) PD4	6
VCC	7
GND	8
(XTAL1/TOSC1) PB6	9
(XTAL2/TOSC2) PB7	10
(T1) PD5	11
(AIN0) PD6	12
(AIN1) PD7	13
(ICP1) PB0	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
PC5 (ADC5/SCL)	
PC4 (ADC4/SDA)	
PC3 (ADC3)	
PC2 (ADC2)	
PC1 (ADC1)	
PC0 (ADC0)	
GND	
AREF	
AVCC	
PB5 (SCK)	
PB4 (MISO)	
PB3 (MOSI/OC2)	
PB2 (SS/OC1B)	
PB1 (OC1A)	

Slika 2: Razvrstitev pinov na mikrokontrolerju

- PORT B (PB7..PB0) - 8 bitna V/I vrata opremljena z internimi dvižnimi upori.
- PORT C (PC5..PC0) - 7 bitna V/I vrata opremljena z internimi dvižnimi upori.
- PORT D (PD7..PD0) - 8 bitna V/I vrata opremljena z internimi dvižnimi upori.
- VCC - napajanje
- GND - masa
- RESET - pin za resetiranje mikrokontrolerja



5.3 Blok shema mikrokontrolerja



Slika 3: Blok shema mikrokontrolerja



6 ORODJA ZA PROGRAMIRANJE MIKROKONTROLERJEV

Pri programiraju mikrokontrolerjev uporabljamo različna orodja. Od programov za pisanje kode, prevajalnikov, do simulatorjev. Za pisanje kode uporabljamo različne programske jezike. Najpopularnejši so »C«, »C++« in BASIC. Za zahtevnejše programiranje in takrat, ko potrebujemo hitro in kratko kodo pa se uporablja zbirni jezik Assembler.

V nadaljevanju si bomo ogledali programiranje AVR mikrokontrolerjev s programskega jezika BASCOM. Beseda BASCOM izhaja iz angleščine, in sicer iz besed basic compiler. Zakaj BASCOM? Ker je zelo enostaven in za začetnike najbolj primeren programski jezik. Je zelo podoben Microsoftovemu QBASIC-u, saj sta 99 odstotno kompatibilna.

BASCOM poleg ostalih ukazov podpira s posebnimi ukazi tudi standardne LCD-je, grafične LCD-je, serijski protokol za komunikacijo s PC-jem RS232, Dallasov 1W protokol poznan po uporabi pri temperaturnih senzorjih (katerega sem uporabil tudi jaz v programu) in A/D pretvornikih, Phillipsov I2C serijski protokol, ki ga razumejo najrazličnejša zanimiva integrirana vezja, kot so: A/D pretvorniki, ura realnega časa in različna vezja, ki se uporabljajo v televizijskih aparatih. Phillips ga je s tem namenom pravzaprav tudi razvil. Dobra lastnost BASCOM-a je tudi to, da lahko rutino za izvedbo določene operacije relativno lahko doda kar uporabnik sam.

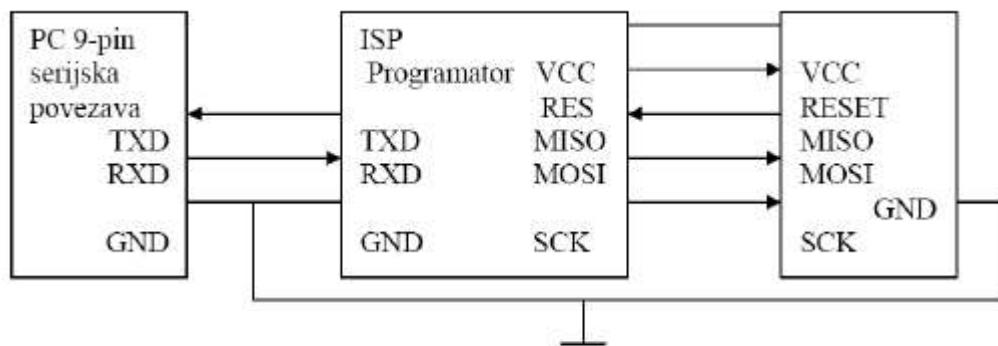
6.1 Programatorji

Mikrokontrolnik lahko sprogramiramo na dva načina. Lahko ga vstavimo v podnožje programatorja ali pa ga sprogramiramo direktno v ciljnem vezju, kar nam ponuja neomejene možnosti pri takozvanem razhroščevanju programa, saj nam mikrokontrolnika ni potrebno vedno vstavljati v programator, da se reprogramira in potem spet v vezje. Programiranje mikrokontrolnika v vezju imenujemo tudi ISP programiranje (in system

programming). Programator se priključi na serijski ali paralelni priključek računalnika odvisno od tipa programatorja. Eden takšnih je tudi PG-302, ki pa zna seveda poleg Atmelovih mikrokrmlnikov programirati tudi vezja proizvajalcev, kot so Phillips, AMD, Dallas in Intel. Programator je pravzaprav enostavna naprava, ki nam strojno kodo vpiše v mikrokrmlnik na enega od dveh možnih načinov.

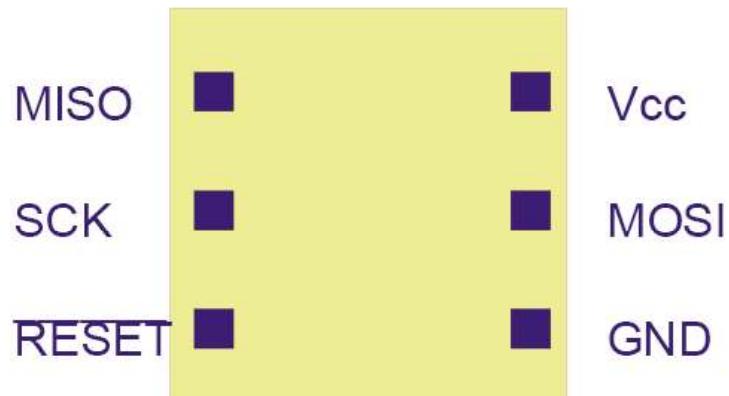
6.2 ISP programator

Pri ISP programiranju lahko programiramo mikrokrmlnike direktno v ciljnem sistemu. Praviloma imajo vsi AVR mikrokrmlniki to možnost. Pomembno je, da smo pri načrtovanju tiskane ploščice predvidili prostor za 6-pinski konektor za programiranje. Pri tem načinu se izognemo odstranjevanju mikrokrmlnika iz vezja in vstavljanju v programator, kar nam lahko prihrani veliko časa in truda še posebej pri razhroščevanju programa, veliko lažje pa je tudi pri popravkih programskeh verzij.



Slika 4: Povezava med programatorjem in mikrokrmlnikom v vezju

Kot je razvidno iz zgornje slike je serijski periferni vmesnik (SPI) sestavljen iz 3-žilne povezave za serijski clock (SCK), master in – slave out (MISO) in master out – slave in (MOSI). Pri programiranju AVR-jev v ciljnem sistemu je programator vedno master, ciljni mikrokrmlnik pa slave. Master omogoči clock za komunikacijo na SCK liniji. Vsak impulz na SCK liniji nam pošlje en bit iz programatorja v ciljni sistem po MOSI liniji. Hkrati pa se isto zgodi na MISO liniji. Za pravilno komunikacijo moramo zagotoviti tudi ozemljitveno linijo med programatorjem in ciljnim sistemom. Da pripravimo mikrokrmlnik v stanje programiranja, postavimo RESET priključek v aktivno stanje. Pri AVR-jih je to negativni potencial. Na negativnem potencialu ostane tudi v času programiranja in v času brisanja le da pri brisanju reset impulzira. Pri tem postopku programator s četrto kontrolno linijo imenovano RESET prevzame kontrolo nad mikrokrmlnikom. Pri programiranju se programator lahko napaja direktno iz vezja in ne porbuje lastnega napajanja, kar eliminira potrebo po dodatnem napajalniku za programator. Lahko pa je tudi obratno. Programator lahko namreč v času programiranja napaja mikrokrmlnik. Spodnja slika prikazuje razpored priključkov, ki jih pri programiranju povežemo med mikrokrmlnikom in programatorjem.



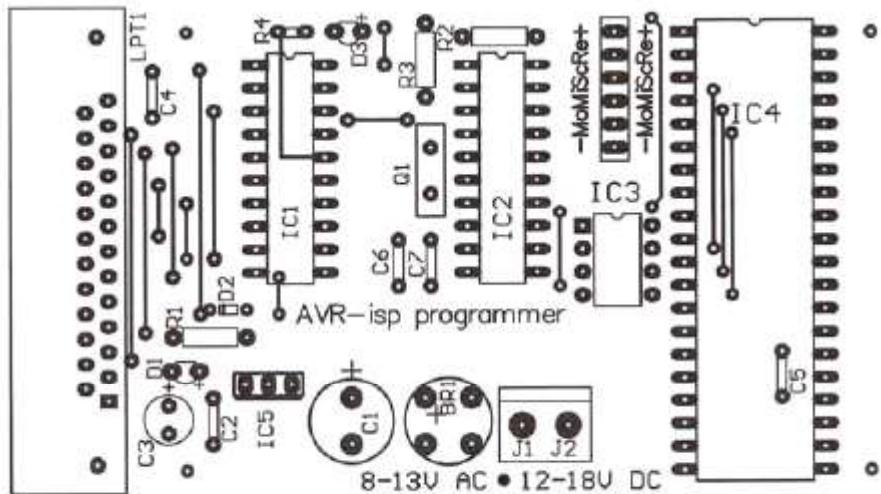
Slika 5: Sponke in signali konektorja za ISP programator

PRIKLJUČEK	IME	KOMENTAR
VCC	Napajanje	Za programiranje lahko programator napajamo direktno iz ciljnega vezja ali obratno.
GND	Masa	Skupna masa za ciljno vezje in programator
RESET	Reset	Za programiranje mora biti reset na nizkem potencialu. Krmili ga programator
SCK	Serial Clock	Sinhronizacijski signal generiran iz programatorja
MISO	Master in – Slave out	Komunikacijska linija iz vezja v programator
MOSI	Master out – Slave in	Komunikacijska linija it programatorja v vezje

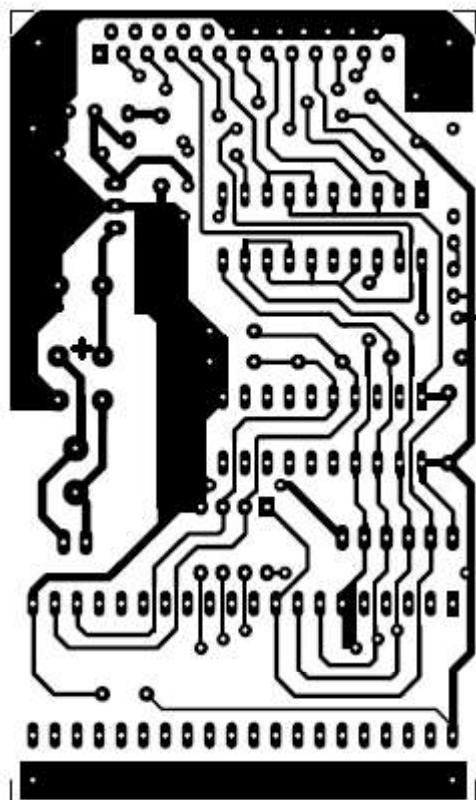
Tabela 1: Opis priključkov ISP konektorja

6.3 Izdelava ISP programatorja

Ko sem podrobneje obdelal mikrokontrolerje in programatorje, sem naredil izdelek in sicer programator AVR ISP Programmer (STK 200). Načrt in film programatorja sem dobil pri podjetju.



Slika 6: Načrt AVR-ISP programatorja



Slika 7: Shema za tiskanino AVR-ISP Programatorja



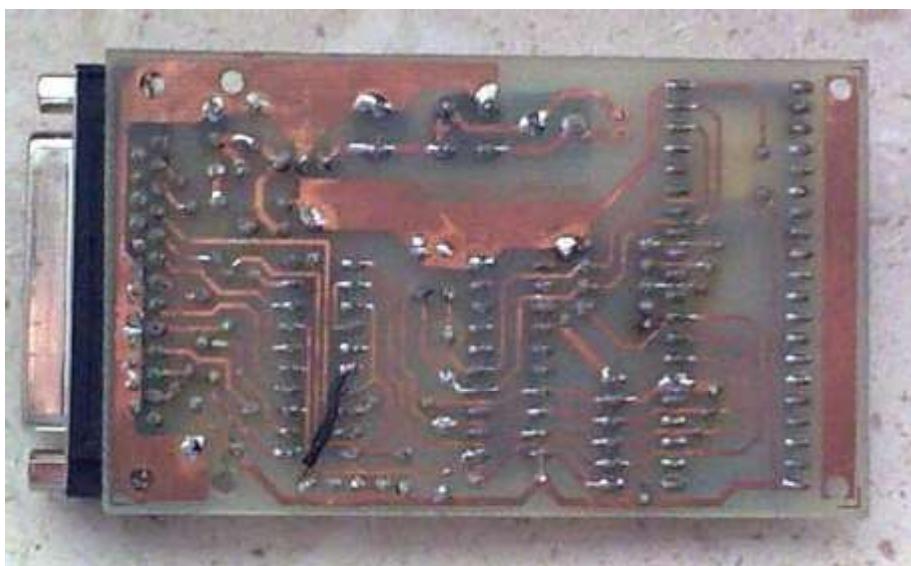
Material za AVR-ISP programmer

Kos	Tip	Vrednost	Referenca
2	upor	1,8 k	R1, R4
1	upor	10 k	R2
1	upor	100 k	R3
1	kvarc	4 MHz	Q1
2	ker.kon.	33 pF RM 5	C6, C7
3	ML kon.	100 nF RM 5	C2, C4, C5
1	el. kon.	22 uF / 50V	C3
1	el. kon.	220 uF / 35 V	C1
1	greatz	B80 C1000	BR1
1	IC	78L05	IC5
1	podnožje	DIL8	IC3
1	podnožje	DIL20	IC2
1	podnožje	DIL40	IC4
1	IC	74HC244	IC1
2	LED 3 mm	RD	D1, D3
1	dioda	1N4148	D2
1	sponke TIV	dvojne	K1
1	pcb		www.avr.4mg.com
6	vrstne špice	RM 2,54	- ,Mo,Mi,SC,Re,+
1	LPT1	konektor za pcb	DB25 ženski

Slika 8: material, potreben za AVR-ISP programator



Slika 9: Izdelan programator (sprednja stran)



Slika 10: Izdelan programator (spodnja stran)



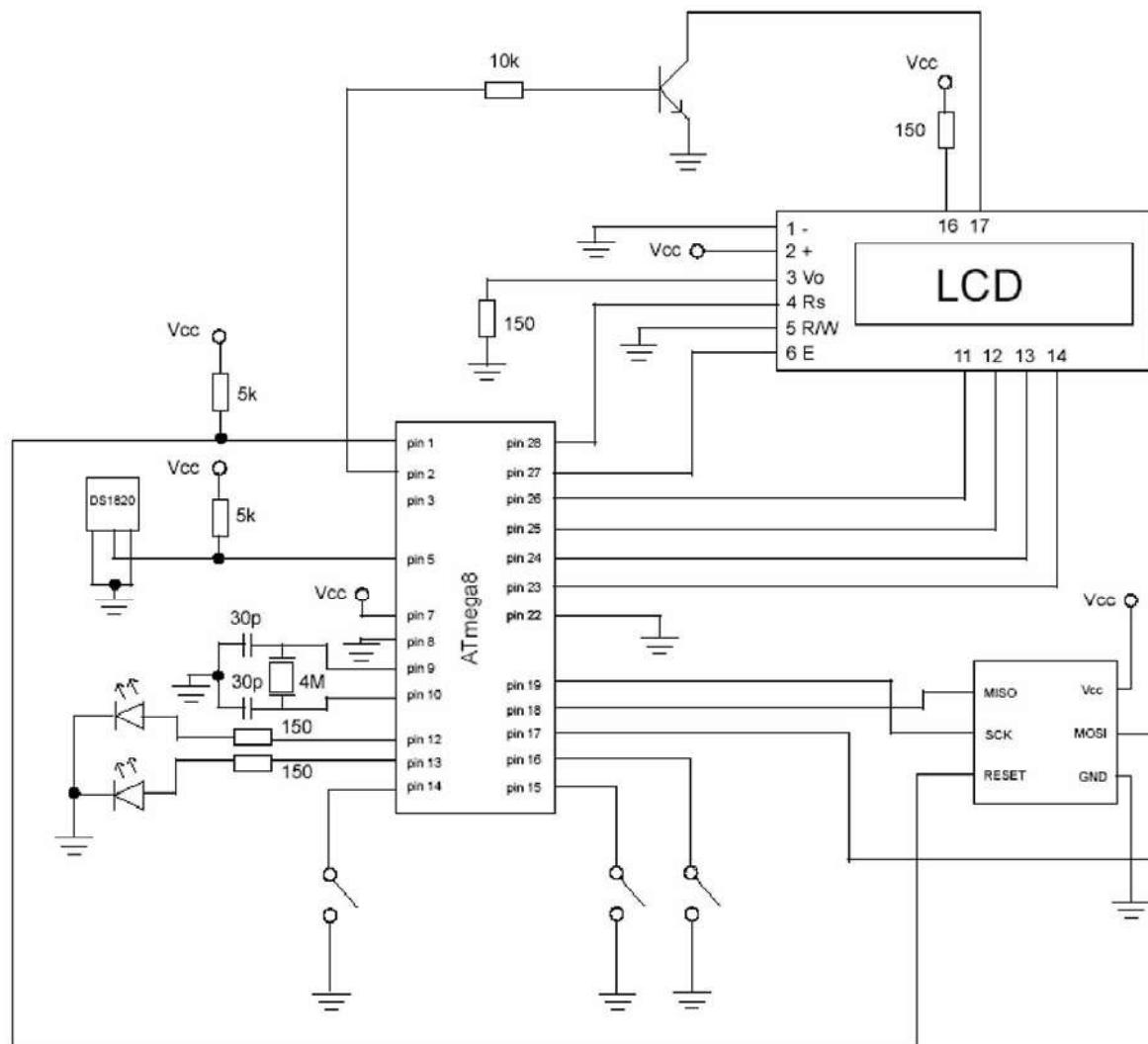
Ta programator ima možnost priključitve mikrokontrolerjev z osmimi, dvajsetimi in z štiridesetimi pini, če hočemo programirati mikrokontroler vstavljen v programator. Mikrokontroler, ki sem ga uporabil jaz (ATmega8), ima 28 pinov, zato sem uporabil še dodaten adapter, ki se vstavi v 40 pinsko podnožje, ko sem želel sprogramirati mikrokontroler izven vezja.



Slika 11: Adapter za ATmega8 mikrokontroler

7 PROGRAM ZA MERJENJE IN URAVNAVANJE TEMPERATURE

Ko sem končal z delom programatorja, sem pa začel pisati program za merjenje in uravnavanje temperature. Sestavil sem vezje, ki je prikazano na spodnji sliki. Poleg mikrokontrolerja ATmega8 sem uporabil še temperaturni senzor DS1820 proizvajalca Dallas Semiconductor in LCD prikazovalnik DEM 16216 SYH-LY/V velikosti 16*2 proizvajalca Display Elektronik GmbH. Za izhod sem uporabil dve LED diodi in sicer eno za prikaz hlajenja in eno za prikaz segrevanja.



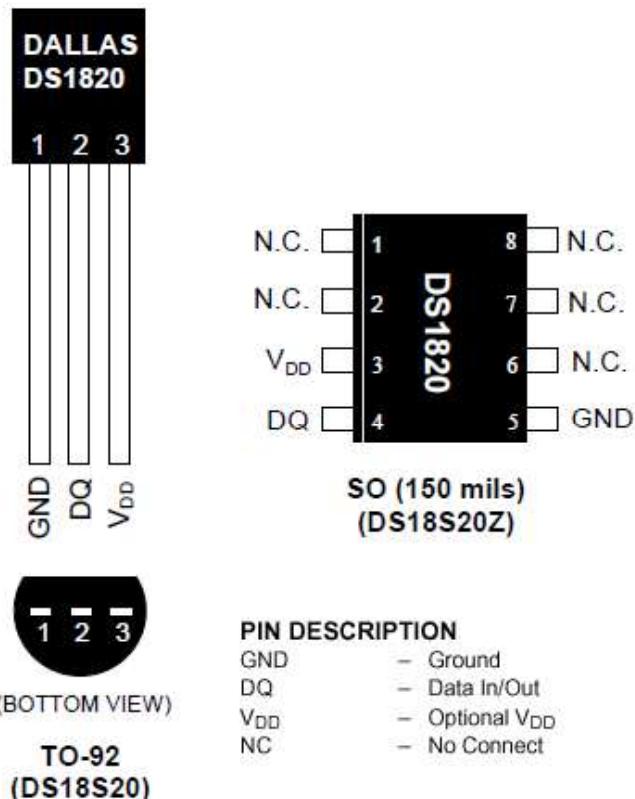
Slika 12: Vezalna shema za merjenje in uravnavanje temperature



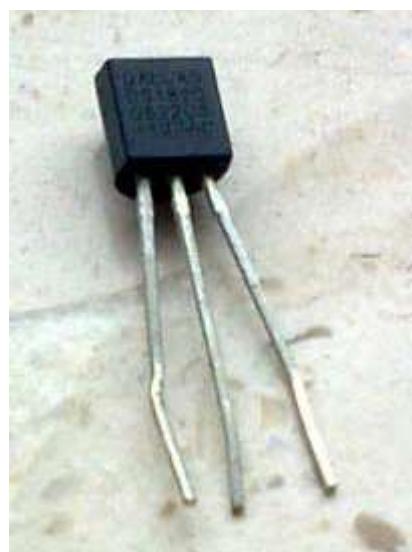
7.1 Merilna sonda DS1820

Za merjenje temperature skrbi senzor DS1820 proizvajalca Dallas Semiconductor. Deluje po unikatnem 1-wire protokolu, ki rabi samo eno linijo za komunikacijo. Senzor DS1820 ne potrebuje nobenih zunanjih komponent, lahko se napaja neposredno iz podatkovne linije (preko pull-up upora $4,7\text{ k}\Omega$). merilno območje senzorja je od -55 pa do 125 °C. Med -10 in +85°C ima napako maksimalno $\pm 0,5^\circ\text{C}$. Temperatura je prebrana kot 9-bitna digitalna vrednost. Vsaka komponenta posebej ima sebi lastno serijsko številko. Zaradi te lastnosti lahko na isti port mikrokontrolerja brez težav priključimo več temperturnih senzorjev. V mojem primeru sem uporabil samo enega, tako da ta lastnost ni bila uporabljena. Pri Dallasu navajajo podatek, da je merilna linija lahko dolga tja do 700 m, vendar se pri taki dolžini linije že pojavijo parazitne induktivnosti in kapacitivnosti , ki pa naj bi se jih dalo odpraviti. Zagotavlja pa, da pri dolžini 100 m deluje brez težav.

DS1820 obstaja v dveh različnih ohišjih, TO-92 in SO, kar prikazuje spodnja slika.



Slika 13: Ohišje TO-92 in SO merilne sonde DS1820



Slika 14: Moja sonda DS1820



1-Wire protokol

Ta protokol je razvilo podjetje Dallas Semiconductor prvenstveno za komunikacijo z nekaterimi svojimi komponentami. Na začetku so bili to temperaturni senzorji, sedaj pa najdemo na tržišču pestro paletu najrazličnejših vezij, ki komunicirajo po 1-Wire protokolu. Za priklop perifernih komponent uporabljam samo eno podatkovno linijo in dve napajalni. Od tod tudi oznaka 1-Wire. Na mikrokrmlilnik lahko spet priključimo več 1-Wire komponent, saj imajo tudi tu komponente različne naslove, preko katerih lahko do njih dostopamo in z njimi komuniciramo. Vsaka izmed 1-Wire komponent ima vpisan unikaten 64-bitni naslov formata, ki ga prikazuje spodnja slika.

8-bitna CRC koda	48- bitna serijska številka	8-bitna koda družine vezij
------------------	-----------------------------	----------------------------

Slika 15: Format zapisa 1-Wire protokola

7.2 LCD Prikazovalnik

Za prikaz meritev služi 16*2 LCD prikazovalnik DEM 16216 SYH-LY/V. Proizvajalec je Display Elektronik GmbH. Priklop LCD-ja je prikazan na vezalni shemi. Osvetlitev sem krmilil preko tranzistorja z mikrokrmlilnikom. Pri osvetlitvi je treba paziti, da zaporedno vežemo predpor, ker osvetlitev deluje na LED. Kontrast prikazovalnika pa krmilimo s potenciometrom. Svoj primer sem rešil z uporom proti masi.



Slika 16: Moj LCD prikazovalnik (sprednja stran)



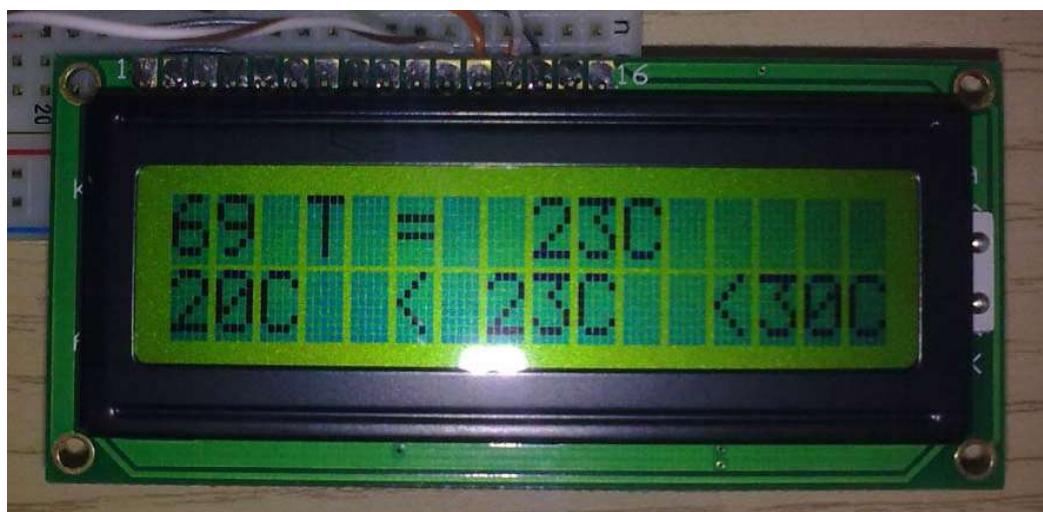
Slika 17: Moj LCD Prikazovalnik (zadnja stran)



7.3 Delovanje programa in program

Program deluje tako, da prebere temperaturo s senzorjem in jo primerja z zgornjo dovoljeno temperaturo in spodnjo dovoljeno temperaturo. Ko je temperatura med zgornjo dovoljeno in spodnjo dovoljeno temperaturo, ne vklopi nobenega od izhodov. Ko pa je temperatura večja od zgornje meje, se vklopi en izhod (recimo ventilator), če pa je temperatura manjša od spodnje dovoljene meje, se pa vklopi drug izhod (recimo grelec). S tipkami pa lahko nastavljamo spodnjo in zgornjo dovoljeno mejo. Vse skupaj se pa prikazuje na LCD prikazovalniku. Naredil sem tudi histerezo. Če se vklopi prvi izhod (ventilator), ostane vkopljen tako dolgo, dokler temperatura ne pade za 3 stopinje od zgornje meje in prav tako za spodnjo mejo, če se vklopi grelec, ostane vkopljen tako dolgo, dokler se temperatura ne dvigne za 3 stopinje od spodnje meje. S tem sem rešil problem nenehnega prižiganja in ugašanja izhodov.

Dodal sem še zaščito, da spodnja meja ne more biti večja kot zgornja meja in obratno.



Slika 18: Prikaz delovanja programa



Do končnega programa sem prišel po sledečih korakih:

- Izdelava vezja na testni plošči
- Priklop vezja na napetost brez mikrokrmlnika in meritev napetosti
- Priklop vezja z mikrokrmlnikom in polnjenje mikrokrmlnika s preprostim programom
- Pisanje definicij za LCD in pisanje sekund števca na LCD
- Programiranje in priklop termometra in izpis temperatur na LCD
- Programiranje tipk
- Končni program ki dela tisto kar želimo

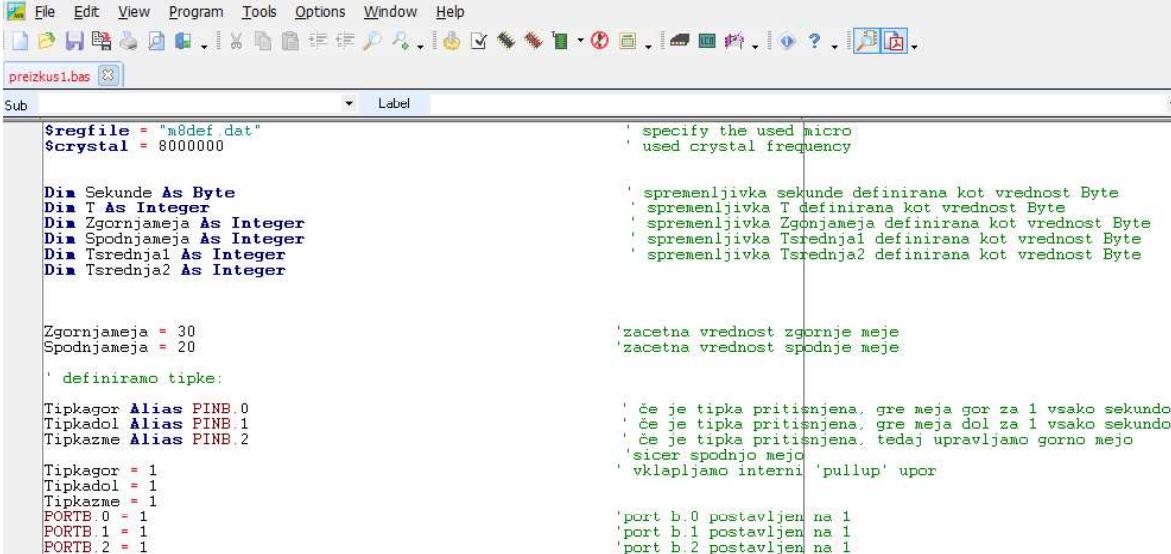
Ko sem izdelal vezje na testni plošči in izmeril napetosti brez mikrokrmlnika, sem napisal zelo kratek program in ga testiral. Program preklaplja diodo vsako sekundo in sicer prižiga in ugaša. Dioda je priključena na port d.6. Ko nam deluje ta program, vemo da znamo pretočiti program iz PC-ja v mikrokrmlnik, da mikrokrmlnik dela in da dela s pravo frekvenco.

```
ub Label
$regfile = "m8def.dat"
$crystal = 8000000
' specify the used micro
' frekvenca kristala

Config PORTD.6 = Output
|           |
Do
|   Toggle PORTD.6
|   Wait 1
Loop
End
' port D.6 konfiguriran kot output
' zacetek do-loop zanke
' preklopi
' čakaj sekundo
' konec do loop zanke
' konec programa
```

Slika 19: Kratek program za testiranje mikrokontrolerja

Ko mi je deloval kratek program za testiranje sem začel z rednim delom programa in sicer najprej sem definiral spremenljivke, in naredil inicializacijo.



```

File Edit View Program Tools Options Window Help
Sub Label
preizkus1.bas
$regfile = "m8def.dat"
$crystal = 8000000
' specify the used micro
' used crystal frequency

Dim Sekunde As Byte
Dim T As Integer
Dim Zgornjameja As Integer
Dim Spodnjameja As Integer
Dim Tsrednja1 As Integer
Dim Tsrednja2 As Integer
' spremenljivka sekunde definirana kot vrednost Byte
' spremenljivka T definirana kot vrednost Byte
' spremenljivka Zgornjameja definirana kot vrednost Byte
' spremenljivka Tsrednja1 definirana kot vrednost Byte
' spremenljivka Tsrednja2 definirana kot vrednost Byte

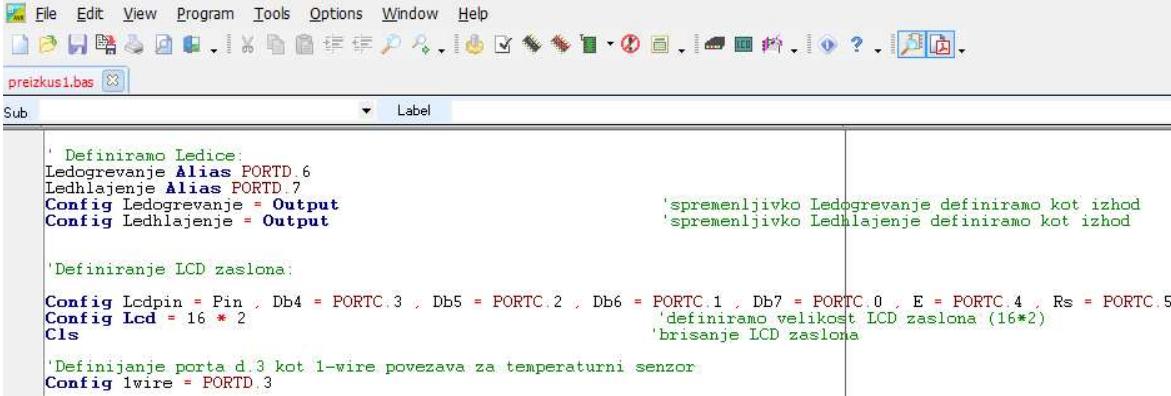
Zgornjameja = 30
Spodnjameja = 20
' zacetna vrednost zgornje meje
' zacetna vrednost spodnje meje

'definiramo tipke:
Tipkagor Alias PINB.0
Tipkadol Alias PINB.1
Tipkazme Alias PINB.2
' če je tipka pritisnjena, gre meja gor za 1 vsako sekundo
' če je tipka pritisnjena, gre meja dol za 1 vsako sekundo
' če je tipka pritisnjena, tedaj upravljamo gorno mejo
'sicer spodnjo mejo
' vklapljam interni 'pullup' upor

Tipkagor = 1
Tipkadol = 1
Tipkazme = 1
PORTB.0 = 1
PORTB.1 = 1
PORTB.2 = 1
'port b.0 postavljen na 1
'port b.1 postavljen na 1
'port b.2 postavljen na 1

```

Slika 20: Inicializacija in definiranje spremenljivk



```

File Edit View Program Tools Options Window Help
Sub Label
preizkus1.bas
'Definiramo Ledice:
Ledogrevanje Alias PORTD.6
Ledhlajenje Alias PORTD.7
Config Ledogrevanje = Output
Config Ledhlajenje = Output
'spremenljivko Ledogrevanje definiramo kot izhod
'spremenljivko Ledhlajenje definiramo kot izhod

'Definiranje LCD zaslona:
Config Lcdpin = Pin , Db4 = PORTC.3 , Db5 = PORTC.2 , Db6 = PORTC.1 , Db7 = PORTC.0 , E = PORTC.4 , Rs = PORTC.5
Config Lcd = 16 * 2
'definiramo velikost LCD zaslona (16*2)
Cls
'brisanje LCD zaslona

'Definiranje porta d.3 kot 1-wire povezava za temperaturnega senzor
Config Iwire = PORTD.3

```

Slika 21: Definiranje Ledic, LCD zaslona in temperaturnega senzorja



Po inicializaciji in definiranju sem program ločil na tri dele, na delovanje tipk z omejitvijo spodnje in zgornje meje, delovanje temperaturnega senzorja in izpis ter upravljanje izhodov (LED).

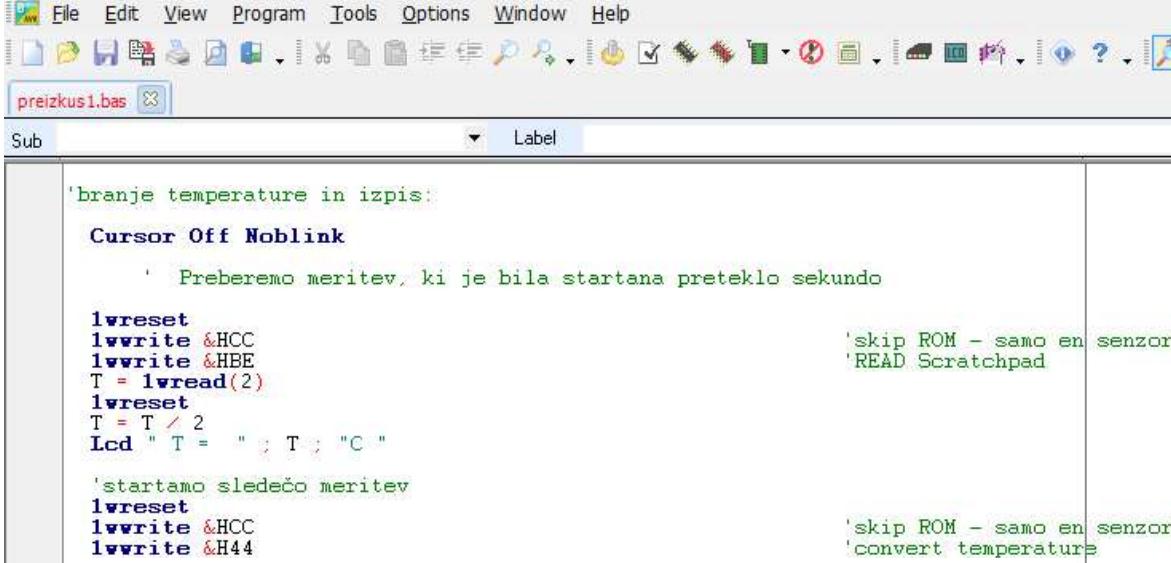
```
Sub Label
    'tipke:

    If Tipkagor = 0 Then
        If Tipkazme = 1 Then
            Incr Zgornjameja
        Else
            Incr Spodnjameja
        End If
    End If
    If Tipkadol = 0 Then
        If Tipkazme = 1 Then
            Decr Zgornjameja
        Else
            Decr Spodnjameja
        End If
    End If

    'omejitev spodnje in zgornje meje:
    'Če je zgornja meja manjša do spodnje:
    If Zgornjameja < Spodnjameja Then Zgornjameja = Spodnjameja

    'Prikažemo na LCD:
    Locate 2, 1
    Lcd Spodnjameja ; "C < " ; T ; "C <" ; Zgornjameja ; "C"
```

Slika 22: Delovanje tipk z omejitvijo spodnje in zgornje meje

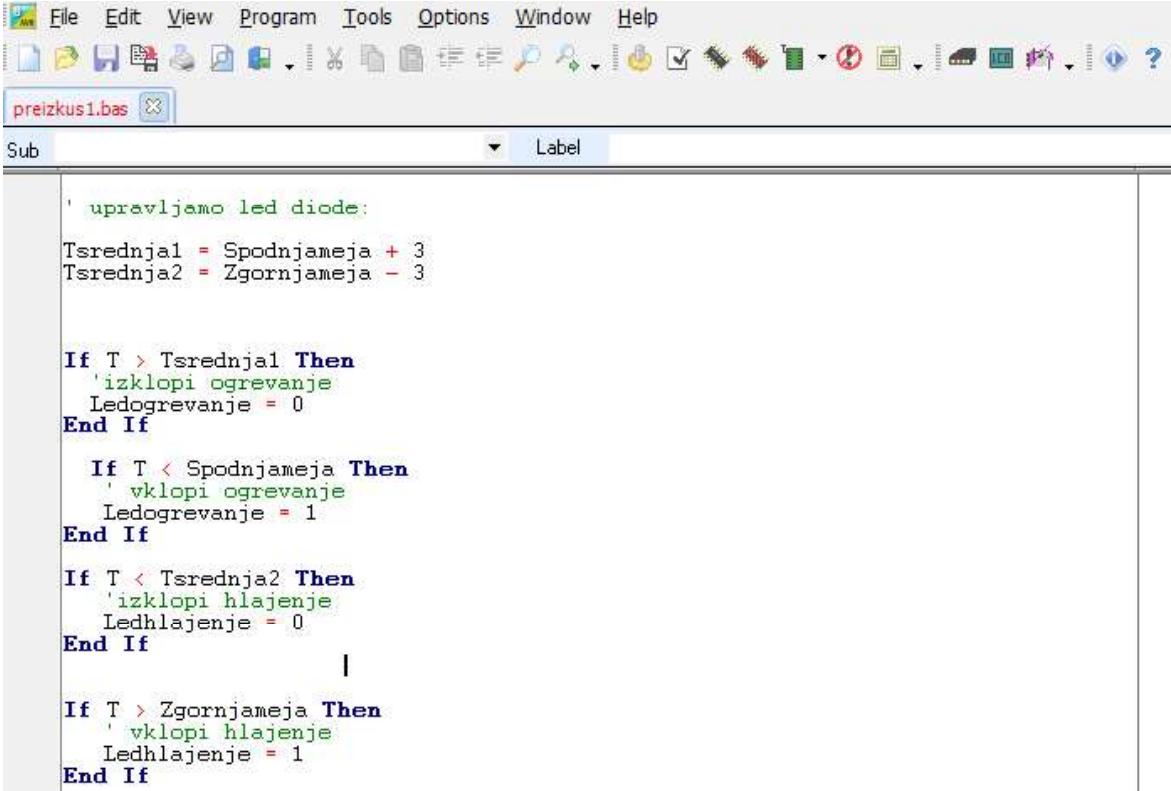


```

File Edit View Program Tools Options Window Help
preizkus1.bas X
Sub Label
'branje temperature in izpis:
Cursor Off Noblink
    ' Preberemo meritev, ki je bila startana preteklo sekundo
1vreset
1vwrite &HCC
1vwrite &HBE
T = 1vread(2)
1vreset
T = T / 2
Lcd " T = " ; T ; "C"
    'startamo sledečo meritev
1vreset
1vwrite &HCC
1vwrite &H44
    'skip ROM - samo en senzor
    'READ Scratchpad
    'skip ROM - samo en senzor
    'convert temperature

```

Slika 23: Branje temperature in izpis



```

File Edit View Program Tools Options Window Help
preizkus1.bas X
Sub Label
' upravljamo led diode:
Tsrednja1 = Spodnjameja + 3
Tsrednja2 = Zgornjameja - 3

If T > Tsrednja1 Then
    'izklopi ogrevanje
    Ledogrevanje = 0
End If

If T < Spodnjameja Then
    'vklopi ogrevanje
    Ledogrevanje = 1
End If

If T < Tsrednja2 Then
    'izklopi hlajenje
    Ledhlajenje = 0
End If

If T > Zgornjameja Then
    'vklopi hlajenje
    Ledhlajenje = 1
End If

```

Slika 24: Delovanje izhoda. (LED diodic)

8 SKLEP

Po opravljenem praktičnem izobraževanju sem dobil nov pogled na mikrokontrolerje in delovanje le teh. Soočil sem se z mnogimi težavami pri programiranju. Mikrokontrolerji so v današnjem svetu zelo razširjeni in bodo v prihodnosti vedno bolj. Najdemo jih povsod. Njihova največja prednost je, da vse vsebujejo vse komponente v enem samem integriranem vezju.

Po opravljeni praksi sem prišel do zaključka, da so mikrokontrolerji zelo uporabni, ter da mi bo novo pridobljeno znanje na nadaljni poklicni poti zelo koristilo.

Ob izteku prakse se zahvaljujem podjetju Audiologs s. p. oziroma gospodu Milenku Glavici, ker mi je omogočil opravljanje praktičnega izobraževanja, na katerem sem pridobil mnogo znanja, ki mi bo koristilo tudi pri izdelavi diplomske naloge in v prihodnosti.

9 PRILOGE

9.1 Celoten program

```

File Edit View Program Tools Options Window Help
File Open Save Save As Print Properties Find Replace Run Stop Exit
preizkus1.bas [ ] Sub Label
$regfile = "m8def.dat"
$crystal = 8000000
' specify the used micro
' used crystal frequency

Dim Sekunde As Byte
Dim I As Integer
Dim Zgornjameja As Integer
Dim Spodnjameja As Integer
Dim Tsrednjai As Integer
Dim Tsrednja2 As Integer
' spremenljivka sekunde definirana kot vrednost Byte
' spremenljivka I definirana kot vrednost Byte
' spremenljivka Zgornjameja definirana kot vrednost Byte
' spremenljivka Spodnjameja definirana kot vrednost Byte
' spremenljivka Tsrednjai definirana kot vrednost Byte
' spremenljivka Tsrednja2 definirana kot vrednost Byte

Zgornjameja = 30
Spodnjameja = 20
' zacetna vrednost zgornje meje
' zacetna vrednost spodnje meje

' definiramo tipke:
Tipkagor Alias PINB.0
Tipkadol Alias PINB.1
Tipkazme Alias PINB.2
' če je tipka pritišnjena, gre meja gor za 1 vsako sekundo
' če je tipka pritišnjena, gre meja dol za 1 vsako sekundo
' če je tipka pritišnjena, tedaj upravljamo gorno mejo
' sicer spodnjo mejo
' vklapljam interni 'pullup' upor

Tipkagor = 1
Tipkadol = 1
Tipkazme = 1
PORTB.0 = 1
PORTB.1 = 1
PORTB.2 = 1
'port b.0 postavljen na 1
'port b.1 postavljen na 1
'port b.2 postavljen na 1

' Definiramo Ledice:
Ledogrevanje Alias PORTD.6
Ledhlajenje Alias PORTD.7
Config Ledogrevanje = Output
Config Ledhlajenje = Output
' spremenljivko Ledogrevanje definiramo kot izhod
' spremenljivko Ledhlajenje definiramo kot izhod

'Definiranje LCD zaslona:
Config Lcdpin = Pin , Db4 = PORTC.3 , Db5 = PORTC.2 , Db6 = PORTC.1 , Db7 = PORTC.0 , E = PORTC.4 , Rs = PORTC.5
Config Lcd = 16 * 2
' definiramo velikost LCD zaslona (16*2)
Cls
'brisanje LCD zaslona

'Definiranje porta d.3 kot 1-wire povezava za temperaturni senzor
Config 1wire = PORTD.3

```



File Edit View Program Tools Options Window Help

preizkus1.bas

Sub Label

```
Do           Select SUB routine          'zacetek Do - Lopp zanke
    'stetje sekund in prikaz na LCD-ju
    T = 0
    Incr Sekunde
    Locate 1, 1
    Lcd Sekunde

    'branje temperature in izpis:
    Cursor Off Noblink
        ' Preberemo meritev, ki je bila startana preteklo sekundo
        lvreset
        lwwrite &HCC
        lwwrite &HBE
        T = lhread(2)
        lvreset
        T = T / 2
        Lcd " T = " ; T ; "C"

        'startamo sledečo meritev
        lvreset
        lwwrite &HCC
        lwwrite &H44
                    'skip ROM - samo en senzor
                    'READ Scratchpad
                    'skip ROM - samo en senzor
                    'convert temperature

    'tipke:
    If Tipkagor = 0 Then
        If Tipkazme = 1 Then
            Incr Zgornjameja
        Else
            Incr Spodnjameja
        End If
    End If
    If Tipkadol = 0 Then
        If Tipkazme = 1 Then
            Decr Zgornjameja
        Else
            Decr Spodnjameja
        End If
    End If

    'omejitev spodnje in zgornje meje:
    'Če je zgornja meja manjša do spodnje:
    If Zgornjameja < Spodnjameja Then Zgornjameja = Spodnjameja

    'Prikažemo na LCD:
    Locate 2, 1
    Lcd Spodnjameja ; "C <" ; T ; "C <" ; Zgornjameja ; "C"
```



File Edit View Program Tools Options Window Help

preizkus1.bas

Sub Label

```
' upravljamo led diode
Tsrednjal = Spodnjameja + 3
Tsrednja2 = Zgornjameja - 3

If T > Tsrednjal Then
    'izklopi ogrevanje
    Ledogrevanje = 0
End If

If T < Spodnjameja Then
    'vklopi ogrevanje
    Ledogrevanje = 1
End If

If T < Tsrednja2 Then
    'izklopi hlajenje
    Ledhlajenje = 0
End If
    |

If T > Zgornjameja Then
    'vklopi hlajenje
    Ledhlajenje = 1
End If

Wait 1
Loop
End
```

Select Label

'čakaj 1s
'konec Do - Loop zanke
'konec programa