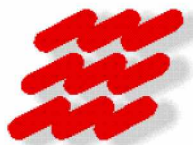


UNIVERZA V LJUBLJANI



FAKULTETA ZA ELEKTROTEHNIKO

Tržaška 25, 1000 Ljubljana



JANEZ KADIVEC

**MERJENJE IN KRMILJENJE
TEMPERATURE V RASTLINJAKU Z
DS1820**

SEMINARSKA NALOGA

pri predmetu

ELEKTRONSKA VEZJA

V Kranju, Marec 2006

KAZALO

1 UVOD	3
1.1 MOTIVACIJA	3
1.2 MIKROPROCESOR, MIKRORAČUNALNIK IN MIKROKONTROLER	3
1.3 DELOVANJE MIKROKONTROLERJEV	4
1.4 ATMEL AVR ATmega8	4
1.5 ORODJA ZA PROGRAMIRANJE AVR MIKROPROCESORJEV	5
1.6 PROGRAMATORJI	5
1.7 ISP PROGRAMATOR	5
1.8 MERILNA SONDA DS1820	7
1.9 LCD-PRIKAZOVALNIK	8
2. DELOVANJE	8
2.1 SHEMA VEZJA	9
2.2 MASKA ZA TISKANINO	9
2.2 IZVEDBA	10
2.3 PROGRAM	10
3. ZAKLJUČEK	13
4. VIRI IN LITERATURA	13

1 UVOD

1.1 MOTIVACIJA

Elektronika že dolgo posega v različne panoge gospodarstva. Danes si človeštvo ne more več predstavljati življenja brez nje. Ravno tako ne brez računalnika in mobitela, pa čeprav ni dolgo tega, ko teh aparatov ni bilo na naših pisalnih mizah oziroma v naših žepih. Še bolj pa si tega ne da predstavljati v industriji kjer je elektronika tako zasidrana, da gospodarstvo tako, kakršno je, ne bi več delovalo.

Kakor v drugih panogah gospodarstva se elektronika že v veliki meri pojavlja tudi v kmetijstvu. Ker mi je ta panoga domača, vidim veliko možnosti za aplikacije elektronskih vezij. Doma se ukvarjamo tudi z vrtnarjenjem, kjer je rastlinjak nepogrešljivi element. Sama temperatura zraka v rastlinjaku je poleg vode in svetlobe ena od glavnih komponent za dobro rast rastlin. Optimalna temperatura za hitro rast rastline je 18–25 °C. V zgodnjih pomladanskih mesecih, ko je temperatura zraka zunaj nizka, se znotraj segreva z grelnikom na kurilno olje. Pozneje, ko se zunanja temperatura zraka dvigne in ob močnem soncu zelo hitro naraste temperatura v rastlinjaku, je potrebno zračenje skozi okna, ki jih ima rastlinjak na stranskih stenah. Ob večernih urah, ko temperatura zraka pade pod optimalno, je potrebno rastlinjak zopet zapreti.

Tukaj sem videl eno od možnosti, da si lahko pomagamo z elektroniko. Zamislil sem si vezje, ki bi na vhodu merilo temperaturo v rastlinjaku in bi na izhodu glede na nastavljeno temperaturo dajalo signal v katero smer bi se vrtela gred motorja. S tem bi se zapirali in odpirali okni rastlinjaka. Nastavitev temperature bi bila mogoča s tipkami. Nastavljena temperatura in dejanska temperatura zraka v rastlinjaku bi bili prikazani na LCD-zaslону.

Temperaturo zraka bi merila sonda DS1820, glavni del vezja pa bi bil Atmelov mikrokontroler AVR ATmega8.

1.2 MIKROPROCESOR, MIKRORAČUNALNIK IN MIKROKONTROLER

Mikroprocesor je centralna procesna enota (CPU) računalnika, zgrajena kot eno integrirano vezje. Če mikroprocesorju dodamo še pomnilnik, V/I enote in nekaj integriranih vezij, ki povezujejo omenjene dele v sistem, dobimo mikroračunalnik. Mikroračunalnik je torej računalnik, čigar CPU je mikroprocesor. Mikroprocesor je kot samostojni element neuporaben. Vedno mu moramo dodajati še druge elemente, ki skupaj tvorijo mikroračunalnik. To dejstvo, da dobimo mikroračunalnik, če mikroprocesorju dodamo pomnilnik in V/I enote, velja tudi za čip. 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.

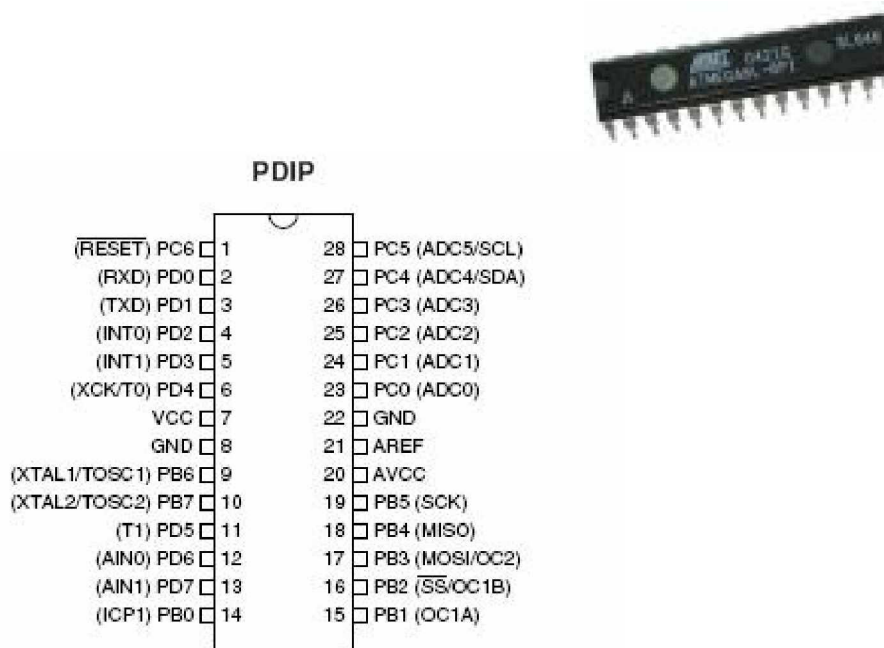
1.3 DELOVANJE MIKROKONTROLERJEV

Delovanje računalnika je zaporedno izvajanje dveh faz: prevzema ukaza od pomnilnika in izvršitve prevzetega ukaza. Pri prevzemu ukaza CPE (centralna procesna enota) najprej prebere operacijsko kodo (z lokacije, na katero kaže v tistem trenutku PC (programski števec)) in jo zapiše 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. Prek podatkovnega registra se nato operacijska koda prenese v ukazni register krmilne enote.

V tem trenutku 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 poveča tudi 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.

1.4 ATMEL AVR ATmega8

V vezju sem uporabil 28-pinski ATmega8, torej AVR v PDIP ohišju. Ima 8 kilobytov »in-sytem« programrljivega flash pomnilnika z možnostjo 10000 vpisov/izbrisov. Ima 512 bytov EEPROMA ter pa 1kilobyta internega SRAMA. Vsebuje 23 programrljivih I/O linij.



Slika 1: ATMEL AVR ATmega8

1.5 ORODJA ZA PROGRAMIRANJE AVR MIKROPROCESORJEV

Pri programiranju mikrokrmilnikov uporabljamo različna orodja, od programov za pisanje kode, prevajalnikov, do simulatorjev. Za pisanje kode uporabljamo različne programske jezike. Najpopularnejši je »C« ali »C++« in BASIC. Za zahtevnejše programiranje in takrat, ko potrebujemo hitro in kratko kodo, pa se uporablja zbirni jezik (angleško Assembler).

V nadaljevanju si bomo ogledali programiranje AVR mikrokrmilnikov s programskim jezikom BASCOM. Beseda BASCOM izhaja iz angleščine, in sicer iz besed basic compiler. Zakaj BASCOM? Ker je zelo enostaven in za začetnike najprimernejši programski jezik. Je zelo podoben Microsoftovemu QBASIC-u, saj sta 99–odstotno kompatibilna. Bascom poleg ostalih ukazov s posebnimi ukazi podpira standardne LCD-je, grafične LCD-je, serijski protokol za komunikacijo s PC-jem RS232, Dallasov 1W protokol, poznan po uporabi pri temperaturnih senzorjih, in A/D pretvornikih, Phillipsov I2C serijski protokol, za katerega smo že omenili, da 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. Bascom trenutno pozna 163 ukazov, vendar se ti hitro dopolnjujejo z novimi, saj je ukaze za izvedbo določene operacije relativno lahko dodati tudi s strani uporabnika.

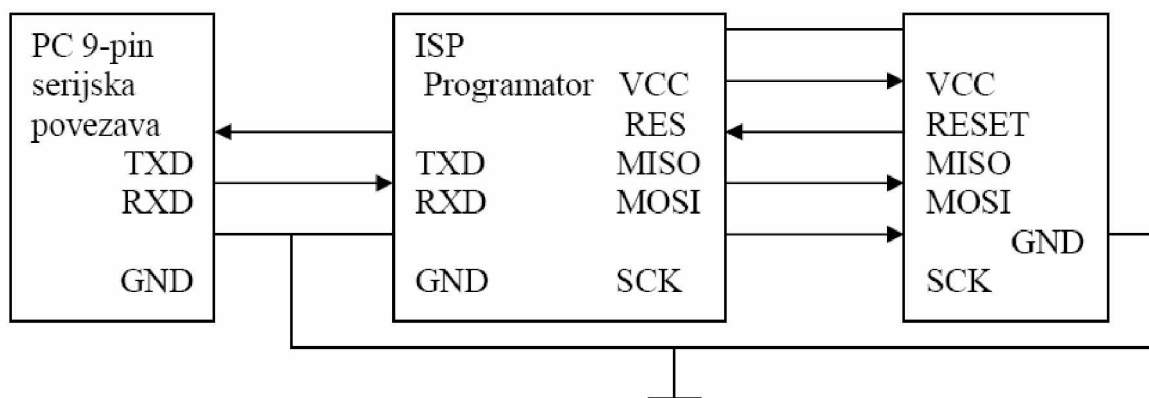
1.6 PROGRAMATORJI

Mikrokrmilnik lahko sprogramiramo na dva načina. Lahko ga vstavimo v podnožje programatorja ali pa ga sprogramiramo neposredno v ciljnem vezju, kar nam ponuja neomejene možnosti pri tako imenovanem razhroščevanju programa, saj nam mikrokrmilnika ni treba vedno vstavljati v programator, da se reprogramira, in potem spet v vezje. Programiranje mikrokrmilnika 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 mikrokrmilnikov programirati tudi vezja proizvajalcev, kot so Phillips, AMD, Dallas in Intel. Programator je pravzaprav enostavna naprava, ki nam strojno kodo vpiše v mikrokrmilnik na enega od dveh možnih načinov.

1.7 ISP PROGRAMATOR

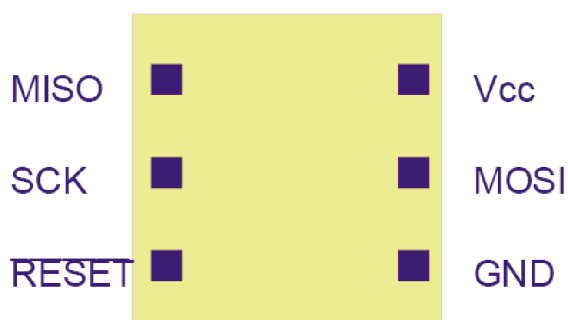
Pri ISP programiranju lahko sprogramiramo mikrokrmilnike neposredno v ciljnem sistemu. Praviloma imajo to možnost vsi AVR mikrokrmilniki. Pomembno je, da smo pri načrtovanju tiskane ploščice predvideli prostor za 5-pinski konektor za programiranje. Pri tem načinu se izognemo odstranjevanju mikrokrmilnika iz vezja in vstavljanju v programator, kar nam lahko prihrani

veliko časa in truda, zlasti pri razhroščevanju programa, veliko lažje pa je tudi pri popravkih programskih verzij.



Slika 2: Programator

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 mikrokrmilnik 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 mikrokrmilnik v stanje programiranja, postavimo priključek RESET 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 nadzor nad mikrokrmilnikom. Pri programiranju se programator lahko napaja neposredno iz vezja in ne potrebuje lastnega napajanja, zato ni potrebe po dodatnem napajalniku za programator. Lahko pa je tudi obratno. Programator lahko namreč v času programiranja napaja mikrokrmilnik. Spodnja slika prikazuje razpored priključkov, ki jih pri programiranju povežemo med mikrokrmilnikom in programatorjem.



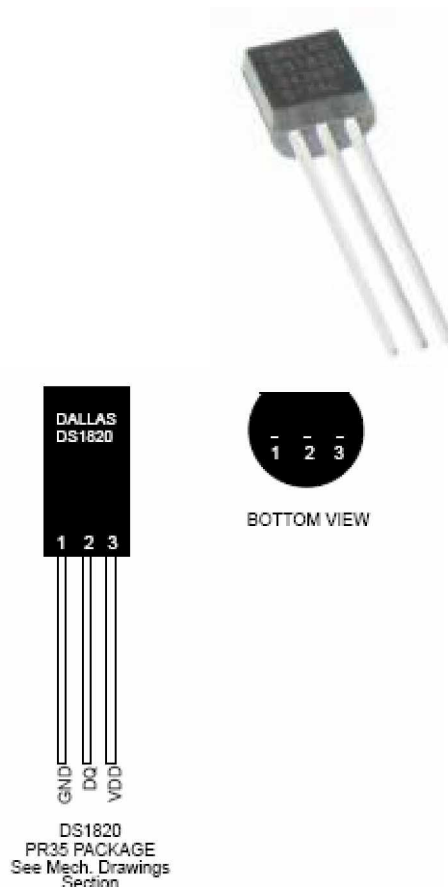
Slika 3: Sponke in signali konektorja za ISP

1.8 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 (prek pull-up upora 4.7 k Ω). Merilno območje senzorja je od -55 pa tja do 125°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 temperaturnih senzorjev. V mojem primeru sem uporabil samo enega, tako da ta lastnost ni bila uporabljena. Bi prišla prav, če bi uporabil več senzorjev. Kot zanimivost naj še omenim, da 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. Zagotavljajo pa, da pri dolžini 100 m deluje brez težav.

Vsaka izmed 1-wire komponent ima vpisan unikatni 64-bitni naslov naslednjega formata:

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



Slika 4: MERILNA SONDA DS1820

1.9 LCD-PRIKAZOVALNIK

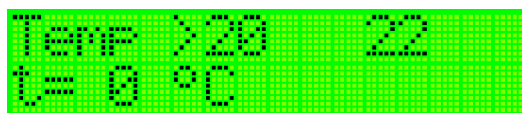
Za prikaz meritev služi 16*2 LCD z osvetlitvijo. LCD PC 1602-C je last podjetja POWER TIP. Priklop LCD-ja je prikazan na načrtu vezja. Osvetlitev sem krmilil prek tranzistorja z mikrokrmilnikom. Pri osvetlitvi je treba paziti, da zaporedno vežemo predupor, ker osvetlitev deluje na LED. Za kontrast prikazovalnika krmilimo s potenciometrom. Svoj primer sem rešil z uporabo proti masi.



Slika 5: LCD-PRIKAZOVALNIK

2. DELOVANJE

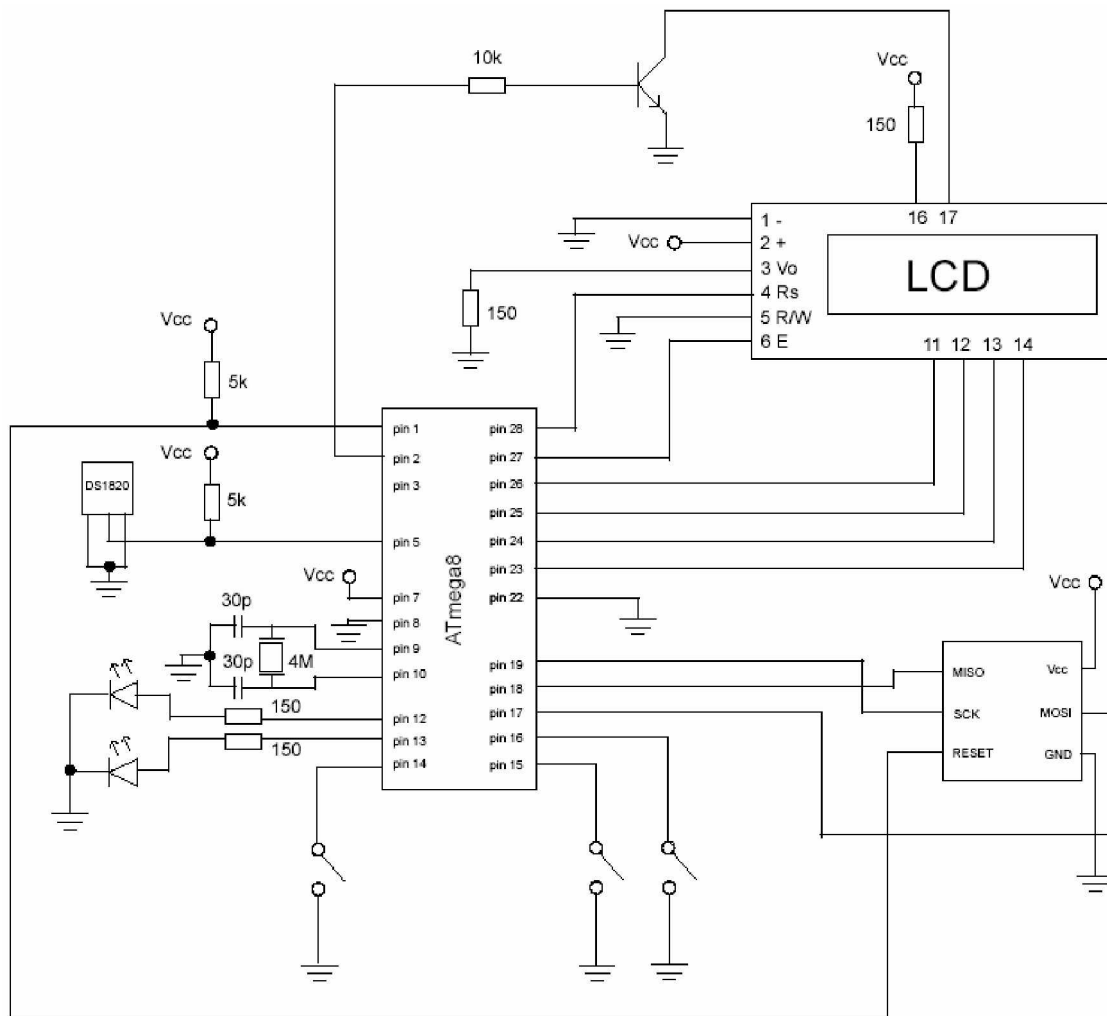
Vezje meri temperaturo v rastlinjaku, jo primerja z nastavljenimi vrednostmi in glede na to na izhod da signal, v katero smer naj zavrti motor (v mojem vezju je tipalo izhoda LED), s tem pa zapre oziroma odpre okni rastlinjaka. Temperaturo je možno nastaviti na spodnjo in zgornjo vrednost. Spodnja vrednost temperature je tista, pri kateri naj se okni zapre, zgornja vrednost pa tista, pri kateri naj se okni odpre. Ni pa možno spodnjo temperaturo nastaviti višje kot zgornjo in obratno, s čimer sem se izognil, da bi bila oba izhoda aktivna hkrati, kar se ne sme zgoditi. Vse te nastavitve pa lahko nastavljamo s tipkami, kar se izpisuje na zaslon. Napajanje je s 4,5 V baterijo.



Slika 6: Prikaz na zaslonu

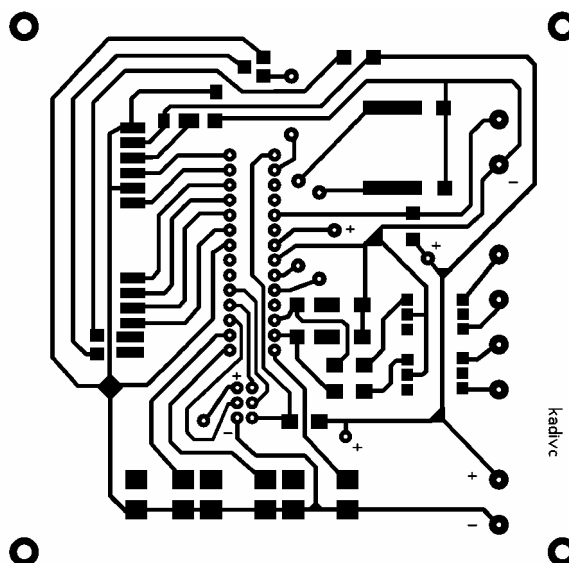
Opomba: V vezalni shemi ni predvidenega optomosa, ki sem ga uporabil v vezju in je viden na tiskanini. Z njim sem omogočil, da bi lahko na izhod vezal močnostni del.

2.1 SHEMA VEZJA



Slika 7: Vezalna shema vezja

2.2 MASKA ZA TISKANINO



Slika 8: Shema za tiskanino

2.2 IZVEDBA

Po vezalni shemi sem v programu Sprint Layout 4.0 naredil masko za tiskanino, ki sem jo natisnil na folijo in jo za likalnikom natisnil na očiščeni pertinaks. Pertinaks sem jedkal in na nastalo tiskanino spajkal nazivne elemente.

Programiral sem v programu za programiranje BASCOM-AVR 1.11.8.1.

2.3 PROGRAM

Temelj vsake sistematične obdelave večjega števila časovno vzporednih dogodkov je časovno rezinjenje. Pri tem se časovna os razreže na posamezne intervale, v katerih se procesor posveča različnim opravilom. Tudi sam sem se odločil za natančno časovno opredelitev nalog. Za take programe rečemo, da potekajo v »realnem času«. Operacijski sistemi za delo v realnem času vsebujejo v svojem najožjem jedru razvrščevalni programski sklop, imenovan »scheduler« ali urnik, ki je odgovoren za logistiko časovnega rezinjenja. Še posebej pri manjših sistemih je treba dobro poskrbeti za učinkovito časovno rezinjenje.

Sistem je zgrajen iz glavnega programa, v katerem se v neskončni zanki izvaja izpis izmerjene in nastavljene temperature na 16*2 LCD-zaslonu, ter iz dveh taskov. Torej program deluje v »realnem času«. Prekinitev generira timer1 vsakih 500 ms. Torej je vsak izmed dveh opravil (taskov) na vrsti po 1 s.

```
$crystal = 4000000
Config Timer1 = Timer , Prescale = 8
On Oc1a Urnik
Compare1a = 25000
Enable Oc1a
Enable Interrupts
```

Definiral sem frekvenco kristala in jo delili z 8, tako da je frekvenca ure 500 kHz. Omogočili smo prekinitve, ki se izvajajo na 25000 strojnih ciklov (500 ms). Ob prekinitvah skoči program na urnik, kjer se nahajajo opravila (taski).

Task 1

V tem opravilu program meri temperaturo. Stevcek_senzor sem dodal, ker potrebujemo nekaj časa, da nam senzor po pretvorbi vrne podatke.

```
Stevcek_senzor = Stevcek_senzor + 1
If Stevcek_senzor = 3 Then
    1wreset
    1wwrite &HCC
    1wwrite &H44
Elseif Stevcek_senzor > 11 Then
```

```

1wreset
1wwrite &HCC
1wwrite &HBE
For I = 1 To 2
    Ar(i) = 1wread()
Stevcek_senzor = 0
Next

Temperatur = Ar(1)
If Ar(2) > 0 Then Temperatur = 0 Else Shift Temperatur , Right

End If

```

Task 2

V drugo opravilo sem združil krmiljenje izhoda, funkcijo tipk in nadzor osvetlitve zaslona.

Pri krmiljenju izhoda s programom krmilimo, kateri izhod da visoko stanje glede na izmerjeno temperaturo.

```

If Temperatur > Temperatura_set(1) Then
    Reset Portd.6
Else
    Set Portd.6
End If

```

```

If Temperatur > Temperatura_set(2) Then
    Set Portd.7
Else
    Reset Portd.7
End If

```

Na Portd.6 in Portd.7 imam vezane LED tako, da mam indikator izhoda.

Pri funkciji tipk imam najprej detekcijo tipk. Torej se glede na pritisnjeno tipko poveča oziroma zmanjša vrednost določene spremenljivke. Vsakič ko je zaznan pritisk katerekoli tipke, pa se spremeni tudi vrednost spremenljivke, ki sproži osvetlitev zaslona.

```

If Pinb.0 = 0 Then
    Tipke_gor = Tipke_gor + 1
    Stevcek_backlite = 1
Else
    Tipke_gor = 0
End If

```

```

If Pinb.1 = 0 Then

```

```

    Tipke_dol = Tipke_dol + 1
    Stevcek_backlite = 1
Else
    Tipke_dol = 0
End If

```

```

If Pinb.2 = 0 Then
    Tipke_ok = Tipke_ok + 1
    Stevcek_backlite = 1
Else
    Tipke_ok = 0
End If

```

Naslednji del kode skrbi za spreminjanje nastavljenе temperature in določanje, katera nastavljenа temperatura naj se spreminja.

```

If Tipke_gor = 1 Then
    Temperatura_set(ts_point) = Temperatura_set(ts_point) + 1
    If Temperatura_set(1) > Temperatura_set(2) Then
        Temperatura_set(ts_point) = Temperatura_set(ts_point) - 1
    End If
End If
If Tipke_dol = 1 Then
    Temperatura_set(ts_point) = Temperatura_set(ts_point) - 1
    If Temperatura_set(2) < Temperatura_set(1) Then
        Temperatura_set(ts_point) = Temperatura_set(ts_point) + 1
    End If
End If
If Tipke_ok = 2 Then
    If Ts_point = 1 Then
        Ts_point = 2
    Else
        Ts_point = 1
    End If
End If

```

V zadnjem delu kode v opravilu se nahaja program za osvetlitev zaslona. Kot sem že omenil, se zaslon osvetli ob pritisku katerekoli tipke. Takrat se spremenljivka Stevcek_backlite postavi na ena. Izhod za osvetlitev je na visokem nivoju, dokler spremenljivka ne doseže vrednosti 250.

```

If Stevcek_backlite > 0 Then
    Set Portd.0
    Stevcek_backlite = Stevcek_backlite + 1
End If
If Stevcek_backlite > 250 Then
    Reset Portd.0
    Stevcek_backlite = 0
End If

```

3. ZAKLJUČEK

Ob izvajanju projekta sem se seznanjal z različnimi orodji za risanje in izdelavo vezja. Pri programiranju sem pridobil veliko izkušenj in preizkusil kar nekaj programerskih prijemov, s katerimi pridemo do ugodnih rešitev problemov, ki jih programerji zagotovo pogosto srečujejo ob svojem delu.

Idej za podobne aplikacije v samem kmetijstvu oziroma vrtnarstvu je nešteto, zato mi bo ta izkušnja zagotovo še kdaj koristila. Mogoče bom s tem projektom kaj kmalu dobil naročilo za izdelavo sistema še za zalivanje, ogrevanje, osvetljevanje...

4. VIRI IN LITERATURA

Tuma, T.: Mikrokrmilniški sistemi, Programiranje za družino HC11, Univerza v Ljubljani: Fakulteta za elektrotehniko, Ljubljana, 2004.

<http://praksa.uni-mb.si/porocila/93527852.pdf>