

»Mladi za napredek Maribora 2013«

30. srečanje

Elektronsko krmiljenje krmilnice

Raziskovalno področje: Elektrotehnika, elektronika

Inovacijski predlog

0Eçđ !KÉ 0E ÁÜWV P S ÉÜŠROZ ÁSOSŸ ÒS

T ^} đ !KÁ ŠCP ÁXQ

Ÿ[|æÁÜÜÒÖP ROZÒŠÒSVÜU ÈÜCE W P OS P Q SCA UŠCÁT OEÜÓUÜ

Maribor, januar 2013

Kazalo

1. POVZETEK	4
2. ZAHVALA.....	4
3. VSEBINSKI DEL	5
3.1. Postopek razvijanja inovacijskega predloga.....	5
3.2. Opis izdelanega modela-inovacijskega predloga	6
3.3. Tehnična in tehnološka dokumentacija	7
3.3.1. Električni načrt budilke z mikrokontrolerjem	7
3.3.2. Električni načrt krmilja z mikrokontrolerjem.....	8
3.4. PIC Mikrokontroler PIC16F84.....	9
3.5. Mikro Basi	11
3.5.1. Resitiranje alarma v zbirniku MicroBasic.....	12
3.6. Zbirni programski jezik Assembler	12
3.6.1. Preverjanje vhodnega signala.....	13
3.7. Sedem-segmentni prikazovalniki	14
4. SKLEP.....	16
5. VIRI.....	17

Kazalo slik

Slika 1: Odprto krmišče (vir: avtor naloge)	6
Slika 2: Zaprto krmišče (vir: avtor naloge)	6
Slika 3: Električna shema alarma (vir: avtor naloge)	7
Slika 4: Postavitev elementov alarma (vir: avtor naloge)	8
Slika 5: Električna shema krmilnega dela vezja (vir: avtor naloge).....	8
Slika 6: Postavitev elementov krmilnega dela vezja (vir: avtor naloge).....	9
Slika 7: Simbol mikrokontrolerja PIC16f84 (vir: avtor naloge)	10
Slika 8: Zgradba PIC16F84 (vir: picstopin).....	11
Slika 9: Primer programa Mikro Basic (vir: avtor naloge)	12
Slika 10: Primer programa v zbirniku Assembler (vir: avtor naloge).....	13
Slika 11: Shema sedem-segmentnega led prikazovalnika (vir: TheLearningPit)	15
Slika 12: Slika sedem-segmentnega prikazovalnika (vir: avtor naloge)	15

1. POVZETEK

Inovacijski predlog je zamisel lažjega načina kmetovanja, pa tudi prihranek pri času, saj pripomore uporabniku, torej kmetu, k lažjemu delu. Zaradi časovno naravnanih loput, ki so vgrajene v krmilnico, uporabnik ne zapravlja časa za krmljenje živine, saj to za njega naredi programirana krmilnica.

Na slovenskem podeželju je kar nekaj prosto rejniških kmetij, katerim bi s to inovacijo veliko pripomogli pri krmljenju živine, saj bi uporabnik oz. kmetovalec lahko napravil svoj urnik oz. si krmljenje prilagodil glede na njegovo zasedenost. Pomagala bi mu tudi v času njegove zasedenosti ali odsotnosti. Ko bo uporabnik ustvaril urnik, bo krmilnica sama prepoznala in razbrala čas krmljenja ter pričela svoj proces, ki bo trajal toliko časa, kakor ga bo uporabnik nastavil.

2. ZAHVALA

Za pomoč, nasvete ter zaupanje se zahvaljujema najinemu mentorju, ki nama je pomagal, da sva izdelala nalogo. K tej ideji ter izboljšavi so veliko pripomogli tudi najini prijatelji. Še posebej pa se zahvaljujema najinim staršem, ki so nama stali ob strani, ko je šlo kaj narobe.

3. VSEBINSKI DEL

3.1. Postopek razvijanja inovacijskega predloga

Ko sva prišla do ideje, sva se morala odločiti, kako idejo uresničiti oziroma jo izdelati. V mislih sva imela predvsem učinkovito ter enostavno delovanje, saj ga mora znati uporabljati vsak. V mislih sva imela računalniško krmiljenje krmilnice, vendar bi bila izdelava takšnega krmilja zelo zapletena, zato sva to opustila. Seveda že obstajajo kmetije z avtomatskim krmiljenjem krmišč vendar so zadeve zelo drage. Tako sva midva hotela stvar poenostaviti ter izdelati napravo čim cenejšo. Vendar kako to narediti?

Izmed možnosti, ki sva jih imela v mislih, sva se odločila, da napravo izdelava s pomočjo PIC mikrokontrolerja, izdelava budilko, na kateri bi lahko nastavili čas ob katerem se bo sprožil signal za odpiranje vrat na krmišču. Vendar je bilo potrebno dodati še mikrokontroler, kateri bo krmilil odpiranje vrat na krmišču. Budilka je seveda morala prikazovati čas, zato so bili potrebni sedem-segmentni prikazovalniki.

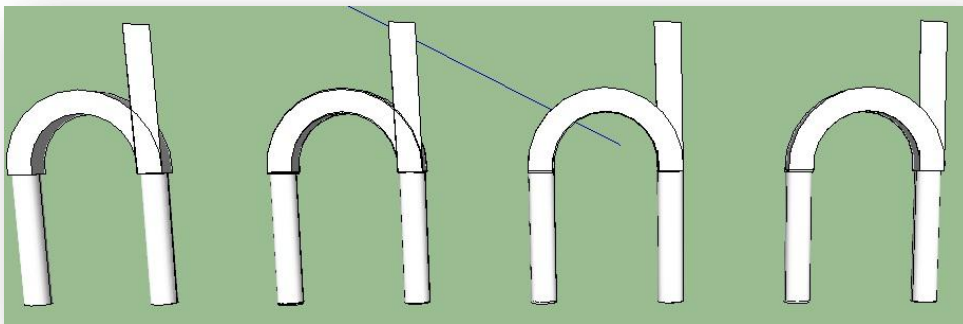
Pri izdelavi naloge sva seveda naletela na probleme kot je krmiljenje. Kako ga izvesti? Z znanjem, ki sva ga že osvojila, sva se odločila, da bi bilo najbolje krmilje izvesti z dvema krmilnikoma, saj nama samo eden ni zadostoval potrebam. Tako sva se odločila, da bova na enem mikrokrmilniku izdelala budilko, na drugem pa ostalo krmilje. Zaradi želje po znanju sva se odločila, da bova program za budilko napisala v zbirniku Mikro Basic, ki sva se ga morala naučiti, saj še nobeden od naju ni nikoli programiral v tem programskem jeziku. Program za drugi mikrokontroler, ki bo krmilil krmiljenje pa v zbirniku, ki sva ga že znala od prej.

Mikrokontroler, ki bo krmilil odpiranje in zapiranje loput, potrebuje za to nalogo dva izhoda, ki bosta to seveda krmilila, saj mora en izhod skrbeti za odpiranje loput, drugi pa za zapiranje loput. Na prvem mikrokontrolerju sva uporabila priključek RA4 kot izhodni priključek, kateri bo poslal signal drugemu mikrokrmilniku. Drugi mikrokrmilnik ima kot vhodni priključek RA0, kot izhodna priključka pa sva uporabila RB0 ter RB1. Program budilke sva napisala tako, da je budilko potrebno izklopiti, ko dobi drugi mikrokontroler signal. Zato sva uporabila še en priključek in to je RB2.

3.2. Opis izdelanega modela-inovacijskega predloga

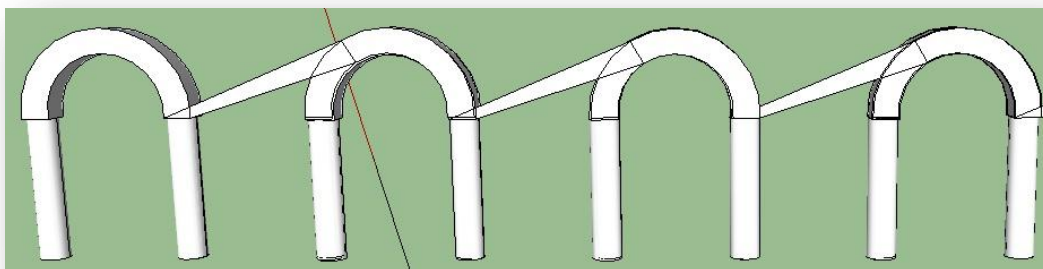
Model je sestavljen iz krmišč, ki so na spodnji sliki, vendar nama je za dokončno realizacijo zmanjkalo časa, zato bova to delovanje prikazala slikovno. S pomočjo programa Google SketchUp sva narisala krmišča v določenih položajih, v položaju odprto in položaju zaprto.

Ko je krmišče odprto, torej dostopno za živino, se lopute dvignejo in živina lahko prosto dostopa do hrane. To jim je seveda omogočeno samo določen čas, saj so drugače na prosti paši. Te lopute bi se seveda odpirale s pomočjo motorjev.



Slika 1: Odprto krmišče (vir: avtor naloge)

Ko poteče čas krmljenja v hlevu se lopute zaprejo ter krmišče ostane zaprto. Živali takrat nimajo več prostega dostopa do krme.

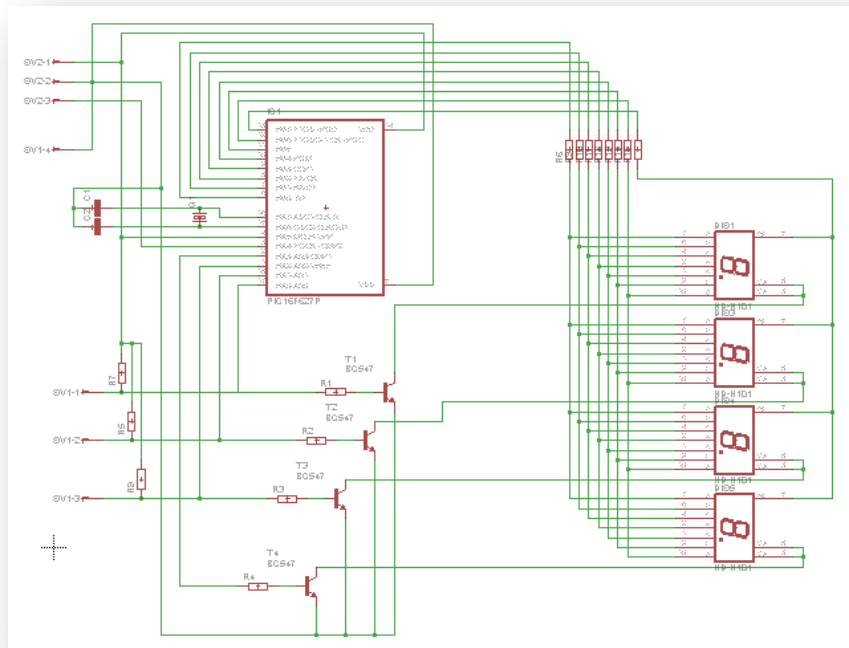


Slika 2: Zaprto krmišče (vir: avtor naloge)

3.3. Tehnična in tehnološka dokumentacija

3.3.1. Električni načrt budilke z mikrokontrolerjem

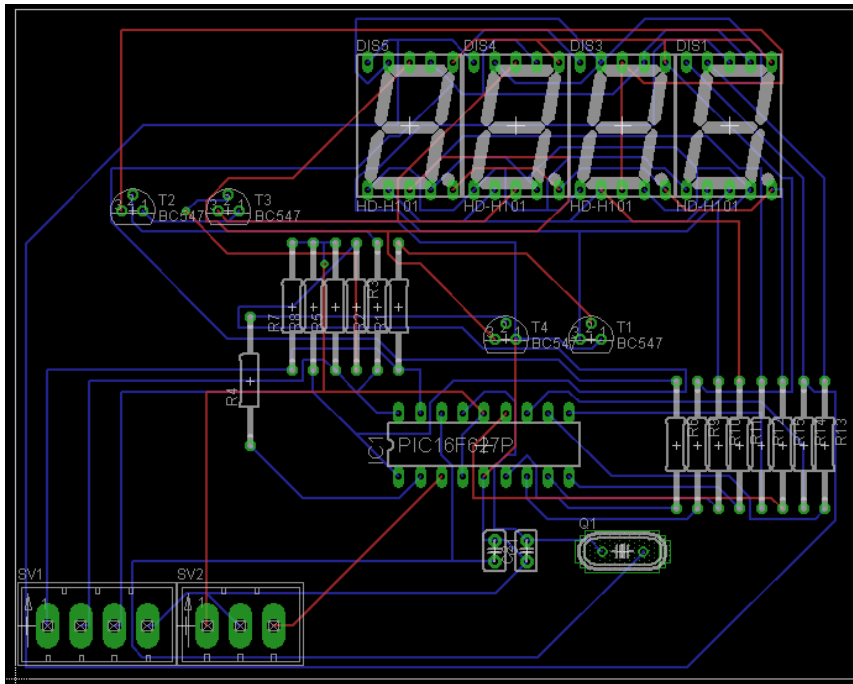
Električni načrt vezja za prvi mikrokontroler, izdelan v programskem okolju Eagle, prikazuje spodnja slika.



Slika 3: Električna shema alarma (vir: avtor naloge)

Električno vezje je izvedeno z PIC mikrokontrolerjem, ki ga bova pozneje opisala ter z sedem-segmentnimi prikazovalniki, ki jih bom tudi opisal v nadaljevanju.

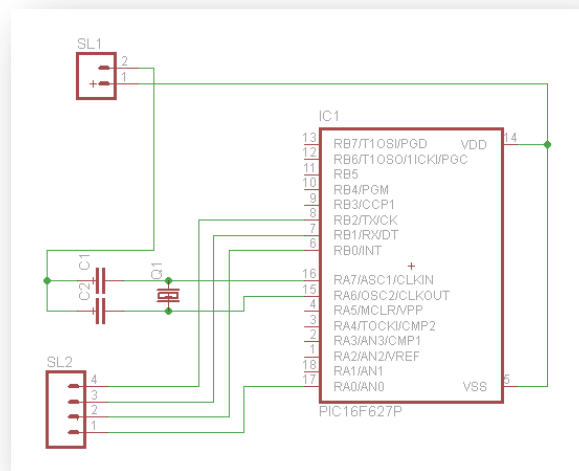
Program Eagle nam pomaga pri izdelavi vezij. Ko si vezje zamislimo in ga načrtujemo, ga izrišemo s pomočjo tega programa, ki nam nato sam izdelava povezave tako, kot jih potrebujemo pri izdelavi tiskanega vezja. Ta program že dobro pozna, zato izdelava vezja ni povzročala posebnih težav. Potrebno je bilo izdelati dvostransko tiskanino, saj nam demo verzija programa ne omogoča izdelati tiskanih vezij večjih dimenzij.



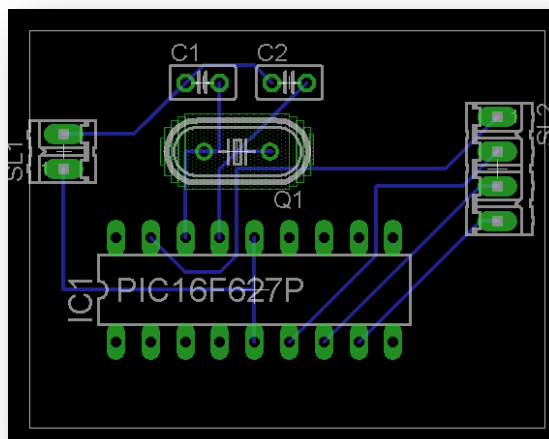
Slika 4: Postavitev elementov alarma (vir: avtor naloge)

3.3.2. Električni načrt krmilja z mikrokontrolerjem

Zaradi že prej omenjenih težav s programom, je bilo potrebno izdelati dve tiskani ploščici. Na spodnjih dveh slikah je prikazan električni načrt za krmilni del.



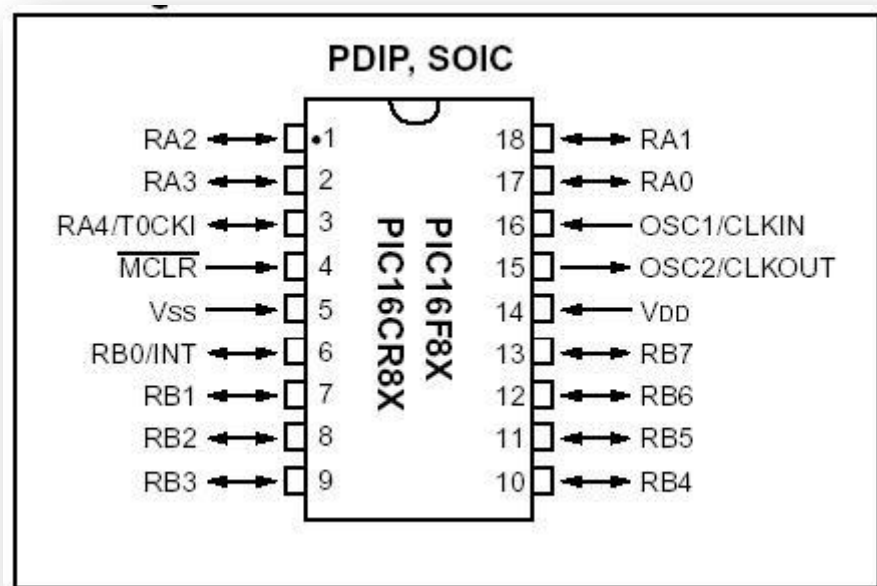
Slika 5: Električna shema krmilnega dela vezja (vir: avtor naloge)



Slika 6: Postavitev elementov krmilnega dela vezja (vir: avtor naloge)

3.4. PIC Mikrokontroler PIC16F84

- Od vseh pinov je 13 vhodno/izhodnih, ki jih lahko programsko krmilimo.
- Napetost napajanja Vdd je 5 V.
- Vsebuje 1 k flash programskega pomnilnika, 68 bajtov RAM pomnilnika in 64 bajtov podatkovnega EEPROM pomnilnika.
- Vsebuje timer in pozna štiri tipe prekinitev.
- CPU je 8 bitna.
- pozna 35 instrukcij za programiranje v zbirnem jeziku (assembler).
- CPU izvaja instrukcije (ukaze). Ima vgrajen 8-bitni delovni register (work). Vanj se shranjujejo podatki po izvedbi posameznih instrukcij.
- Programski pomnilnik je sestavljen iz 1024 lokacij. V vsako lokacijo lahko vpišemo 14-bitni podatek. Instrukcija zavzame eno lokacijo. Posebno vlogo imata prva in peta lokacija. V prvi se nahaja ukaz, ki so bo izvedel takoj po vklopu mikrokontrolerja. V peti se začne del programa, ki se izvede ob prekinitvi.
- Podatkovni RAM pomnilnik je razdeljen na dve banki, banko 0 in banko 1. Prvih 12 lokacij je namenjenih SFR registrom (**S**pecial **F**unction **R**egisters), ki služijo za komuniciranje z ostalimi enotami na čipu. Na naslednjih lokacijah je 68 GPR registrov (**G**eneral **P**urpose **R**egisters).



Slika 7: Simbol mikrokontrolerja PIC16f84 (vir: avtor naloge)

OSC1 - Vhod zunanjega kristalnega oscilatorja (določanje takta).

OSC2 - Izhod zunanjega kristalnega oscilatorja.

MCLR - Master clear, reset vhod. Resetiramo z "0", delovanje programa "1".

RA0 do RA3 - vhodno / izhodni pini.

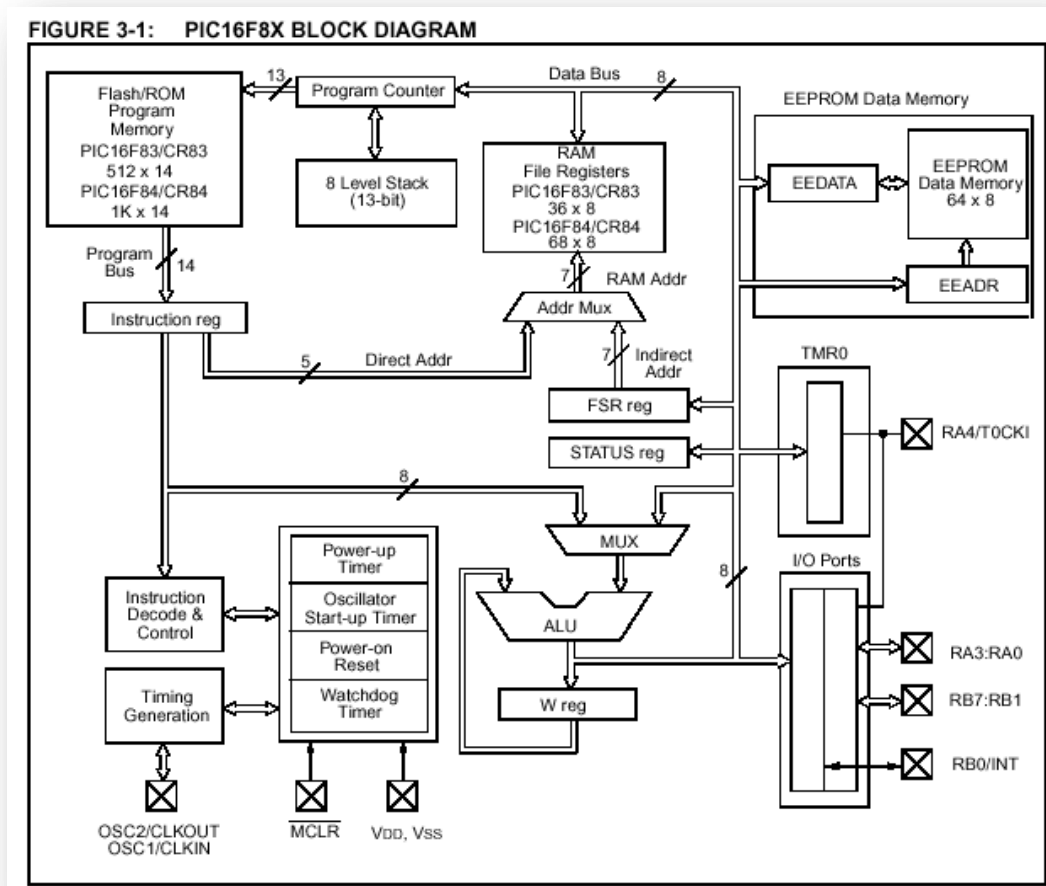
RA4/T0CKI - Pin je open drain tipa. Pozna dve stanji: maso, kadar je pin na stanju "1" in veliko impedanco, ko je v stanju "0". (Če vežemo Led diodo med +5 V in RA4, se bo vklopila, če bomo postavili ta pin na "0". Če pa vežemo Led diodo med pin RA4 in maso, se ne bo nikoli vklopila.)

RB0/INT - Pin RB0 je lahko izbran za zunanji vir prekinitvev. Lahko ga določimo kot vhodni ali izhodni pin mikrokontrolerja.

RB1 do RB7 - vhodno / izhodni pini. Pini RB4 do RB7 so lahko izbrani za zunanji vir prekinitvev.

VSS - masa.

VDD - napetost napajanja +5 V.



Slika 8: Zgradba PIC16F84 (vir: picstopin)

3.5. MikroBasic

MikroBasic je polno-izrazit Basic prevajalnik, ki je za razvoj mikrokrmilniškega projekta primeren za vsakogar. Na voljo je v petih okusih za pet različnih mikroprocesorskih arhitektur. Popularni osnovni programski jezik je najboljša izbira za začetnike, ker ima preprosto sintakso in jasno kodo. Udobno IDE s celovito datoteko pomoči in prosto tehnično podporo pomeni veliko, za prvi korak v svet mikrokrmilnikov. Da ne omenjamo več kot 500 pripravljen knjižic funkcij in primerov, ki nam pomagajo pri razvoju.

3.5.1. Resitiranje alarma v zbirniku MicroBasic

Na spodnji sliki je prikazana funkcija napisana v zbirniku mikroBasic, ki prebere vhodni signal ter nam izključi alarm.

```
'  
' CLEAR alarm  
'  
                                dp.3 = 0  
                                PORTA.4 = 1  
                                end if  
                                wend  
end.
```

Slika 9: Primer programa mikroBasic (vir: avtor naloge)

3.6. Zbirni programski jezik Assembler

Zbirni jezik assembler je najnižji programski jezik, na katerem temeljijo vsi programski jeziki. Opisal bom instrukcije, kaj katera pomeni in kaj v programu naredi.

Ukazi za logične operacije

andwf - logična operacija AND med registrom w in f.

andlw - Sešteje registra l in w.

iorwf - Med w in f se izvede logična operacija OR.

iorlw - Izvede se logična operacija OR med l in w.

xorwf - Izvede se logična operacija XOR med w in f.

xorlw - Izvede se logična operacija XOR med l in w.

clrf - Vsebino registra f izbriše.

clr w - Clear w - izbriše vsebino registra w.

comf - Complement - vrednost registra w prenese v register f.

bcf - Bit clear file - izbriše bit b v registru f.

bsf - Bit set file - postavi bit b v registru f.

rlf - Rotate left f through carry - vsebina registra f se premakne za eno mesto v levo skozi zastavico statusnega registra Carry.

rnf - Rotate right f through carry - vsebina registra f se premakne za eno mesto v desno skozi zastavico statusnega registra carry.

movlw - Move literal to w - konstanta literal se premakne v w.

movwf - Move work to file - vrednost work premakne v f.
movf - Move f - vsebino registra f premakne v ciljni register.
swapf - Swap nibbles in f - zgornje in spodnje štiri bite zamenja med seboj.

Ukazi za aritmetične operacije

addwf - Add w and f - vsebini registra w se prišteje vsebina registra f.
addlw - Add literal and w - vsebini registra w se prišteje l.
subwf - Subtract w from f - od vsebine registra w odšteje vsebino registra f.
sublw - Subtract w from l - od vsebine registra w se odšteje l.
incf - Increment f - poveča f za ena
decf - Decrement f - zmanjša f za ena
goto - Unconditional Branch - brez pogojni skok
call - Call Subroutine - klic podprograma
return - Return from subroutine - vrnitev iz podprograma
incfsz - Increment f, skip if zero - povečaj f za ena preskoči naslednji ukaz, če je vrednost registra f 0.
decfzs - Decrement f, skip if zero - zmanjšaj register f za ena preskoči naslednji ukaz, če je njegova vrednost 0.
btfss - Bit test file skip if set - testiraj bit v registru f, prekoči naslednji ukaz, če je bit v registru f setiran - 1.
btfsc - Bit test file skip if clear - testiraj bit v registru f, preskoči naslednji ukaz, če je bit v registru f enak 0.

3.6.1. Preverjanje vhodnega signala

Če hočemo da nam program preverja stanje na vhodu mikrokontrolerja, moramo preverjat vhodni pin za instrukcijo BTFSS, (Bit test file skip if set - testiraj bit v registru f, prekoči naslednji ukaz, če je bit v registru f setiran - 1.

```
Preverjanje_vhodnega_signala  
  
    btfss    PORTA,0           ;Ali je vhodni logična "1"  
    goto    Preverjanje_vhodnega_signala ;Ni vhodnega signala, preverjaj znova  
    goto    Resetiraj_alarm    ;Program nadaljuje v podprogramu za resetiranje
```

Slika 10: Primer programa v zbirniku Assembler (vir: avtor naloge)

3.7. Sedem-segmentni prikazovalniki

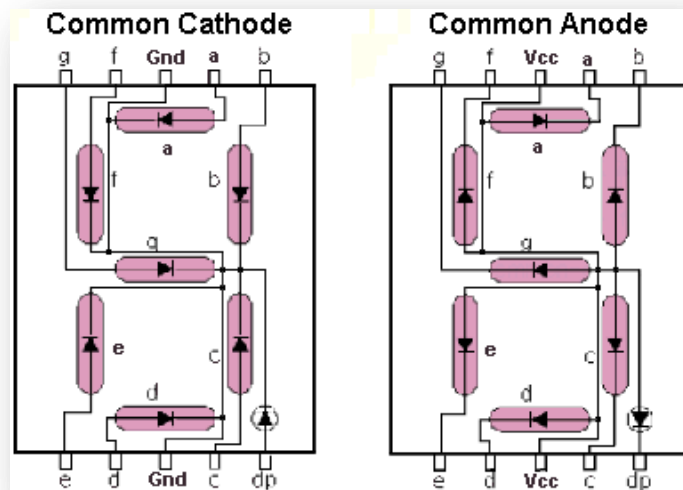
7-segmentni prikazovalnik z omogočitvijo nam omogoča prikaz izhodnih signalov v obliki simbolov na pravokotnem polju, ki ga tvori 7 svetlečih LED diod in LED dioda za piko. Prikaz na celotnem prikazovalniku lahko kadarkoli izklopimo oziroma skrijemo z logično enico na posebnem vhodu za izklop prikaza.

7-segmentni prikazovalnik z omogočitvijo ima 9 vhodnih linij z oznakami a, b, c, d, e, f, g, p in off. 7 vhodov z oznakami od a do g ustreza enako poimenovanim segmentom. Vhod p ustreza dodatnemu segmentu za prikaz decimalne pike. Vhod off omogoča hkraten izklop vseh segmentov.

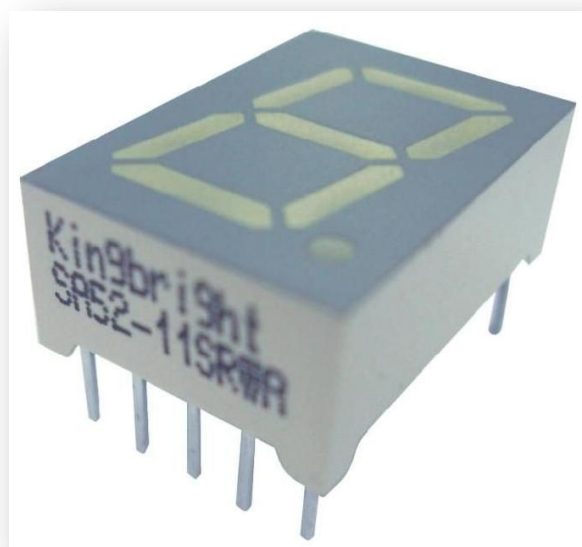
Enica na posamezni liniji pomeni, da bo pripadajoči segment svetil, ničla na posamezni liniji pomeni, da je segment s to oznako temen, ker je izključen. Če vhod off ni priključen ali če je na njem ničla, je delovanje prikazovalnika omogočeno, enica na tem vhodu pa pomeni izklop vseh segmentov.

Opomba: element nam omogoča uporabo skupne linije, ki je običajna pri pravih elementih. Pravi 7-segmentni LED prikazovalniki imajo skupno anodo, ki jo priklopimo na plus ali skupno katodo, ki jo priklopimo na maso. V prvem primeru vključimo posamezen segment tako, da njegov priključek, ki predstavlja katodo LED diode "potegnemo dol" oziroma nanj priključimo logično ničlo, običajno v obliki izhoda s prostim kolektorjem. Pri skupni katodi pa na priključke pripeljemo vir toka, torej logično enko.

7-segmentni prikazovalnik z omogočitvijo izklopa ustreza varianti 7-segmentnega LED prikazovalnika s skupno katodo. V takšnem primeru si lahko predstavljamo, da je vhod off skupna katoda. Če hočemo, pa lahko dobimo tudi ekvivalent za 7-segmentnega LED prikazovalnika s skupno anodo. Za tak tip prikazovalnika moramo vse vhode 7-segmentnega prikazovalnika z omogočitvijo negirati. Simulator nam omogoča, da to naredimo kar na samem prikazovalniku.



Slika 11: Shema sedem-segmentnega led prikazovalnika (vir: TheLearningPit)



Slika 12: Slika sedem-segmentnega prikazovalnika (vir: avtor naloge)

4. SKLEP

Zelo sva zadovoljna in ponosna, da sva uspešno končala inovacijsko nalogo. Ko pomisliva na začetek tega inovacijskega predloga, ko sva imela samo zamisel, kako bi naj to delovalo, naju je čakalo še veliko dela. Vendar z dobro razdeljenim delo sva si zelo olajšala najino raziskovanje.

Nama je na začetku predstavljal največji izziv programski jezik mikroBasic, saj ga nobeden od naju do sedaj še ni poznal. Vendar sva se s pozitivnimi misli na uspeh spopadla s tem programskim jezikom in sva se ga naučila. Več kot sva znala, bolj sva bila željna znanja. Vendar je bilo potrebno obnoviti tudi znanje programskega jezika assembler, saj sva določene stvari že pozabila od zadnjega programiranja.

Zelo bi bila zadovoljna, da bi nama uspelo že narediti maketo, ampak kljub temu bova to zadevo realizirala ter jo izboljšala z dodatnimi senzorji za boljše in varnejše delovanje. To bova naredila tudi praktično, da bova videla, kako se obnese ter ali je cenovno dostopno.

5. VIRI

1. Spletni vir, dostopno na URL: <https://sites.google.com/site/programiranjepic/> (4.2.2013)
2. Mikro Basic (spletni vir). Dostop na URL: <http://www.mikroe.com/mikrobasic/> (4.2.2013)
3. picstopin (spletni vir). Dostop na URL: <http://img.webme.com/pic/m/mohoahmed/a4.jpg> (4.2.2013)
4. TheLearningPit (spletni vir). Dostop na URL:
<http://www.thelearningpit.com/lp/doc/7seg/7segLed.gif> (4.2.2013)