UNIVERZA V LJUBLJANI
FAKULTETA ZA ELEKTROTEHNIKO

Matej Anžin

STROJNA OPREMA ZA IZVEDBO MERILNIKA SRČNEGA UTRIPA

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Ljubljana, december 2005

Zahvala

Doc. dr. Boštjanu Murovcu se kot prvemu zahvaljujem, da me je z veseljem sprejel pod svoje mentorstvo in me spretno vodil v pravo smer. Bil mi je v veliko podporo, saj mi je pomagal pri izbiri tematike. S tem mi je to diplomsko delo pisano na kožo in sem ga z veseljem sestavil in napisal. Posebna zahvala gre doc. dr. Slavku Kocijančiču za predlagano tematiko ter izvirne prispevke z medicinskega stališča. Rad bi se mu zahvalil tudi za mehansko izdelavo senzorja, ki sem ga uporabljal pri mojem diplomskem delu.

Zahvalil bi se rad mojima staršema in sestri Jeri za vso podporo pri študiju.

Posebna zahvala gre moji Nini za vzpodbude in toleriranje zasedenosti z delom v času izdelave diplomskega dela.

Matej Anžin

Kazalo

Kazalo slik

Kazalo tabel

Kazalo grafov

Kazalo enačb

Kazalo izsekov iz kode

Seznam uporabljenih simbolov

ASCII	American Standard Code For Information Exchange
AD	Analogno digitalni pretvornik
DA	Digitalno analogni pretvornik
SPI	Serial Peripheral Interface
BAUD	Enota hitrosti sprememb električnega signala
LED	Light emitting diode
RISC	Reduced instruction set computer

Povzetek

Diplomsko delo je del raziskovalnega projekta, katerega cilj je izgradnja poligrafa za namene raziskovanja.

Poligraf bo vključeval merjenje frekvence srca, prevodnosti kože ter nihanje telesne temperature. S pomočjo teh pokazateljev se bo raziskoval telesni odziv na določena duševna stanja raziskovanega osebka.

Del projekta, katerega to diplomsko delo predstavlja, je merjenje frekvence utripa srca. Namen diplomskega dela je razviti sistem za merjenje frekvence utripa s pomočjo zaznavanja sprememb krvnega tlaka v kapilarah. Sistem mora zagotavljati robustno merjenje spremembe frekvence, tudi med dvema ali temi periodami utripa srca.

Diplomsko delo obsega razvoj elektronskega vezja za zajem photoplethysmografa s pomočjo svetlobnega senzorja ter analognega procesiranja signala. Obdelane signale mora nato vzročiti ter prenesti na računalnik. Za senzor smo se odločili uporabiti svetlečo diodo, katera sveti v infrardečem spektru, ter foto diodo, ki ima največjo občutljivost v istem sprektralnem območju. Prva osvetljuje tkivo s svetlobo določene valovne dolžine, foto dioda, pa zaznava majhne spremembe osvetljensoti, katere so sorazmerne s spremembami svetlobne prevodnosti tkiva. Svetlobna prevodnost tkiva je odvisna od razširjenosti kapilar ter koncentracij dveh vrst hemoglobina, kar pa je odvisno od krvnega tlaka. Dobljeni signal s foto diode smo nato spustili skozi pasovno prepustno analogno sito, ter ga ojačali s posebnim ojačevalnikom, kateremu lahko nastavljamo ojačenje in odmik preko digitalno nastavljive upornosti in digitalno analognega pretvornika. Dobljeni signal smo nato pretvorili s pomočjo analogno digitalnega pretvornika v obliko, primerno za prenos v osebni računalnik. V obseg diplomskega dela spada tudi razvoj sistema za komunikacijo mikrokrmilniškega sistema z osebnim računalnikom. Zaradi velikega števila dejavnikov, kateri vplivajo na delovanje, mora biti mikrokrmilniški sistem sposoben nadzorovati delovanje senzorja ter obdelave signala.

S pomočjo zgoraj navedenih ukrepov je nastalo robustno in prilagodljivo elektronsko vezje, katero omogoča nadaljno obdelavo in meritve.

Ključne besede:

frekvenca srčnega utripa, Infrardeč svetlobni senzor, aktivna analogna sita, digitalno nastavljivo ojačenje, mikrokrmilniški sistem, prenos podatkov na osebni računalnik.

Abstract

This thesis is a part of research project, whose aim is to construct a polygraph for performing psychological research. Polygraph consists of heart beat rate measurements, skin conductance and oscillation of body temperature. With aid of these indicators body responses to different frames of mind of person are intended to be studied.

This thesis covers only heart beat rate measurements. Within our work we developed an electronic circuit for acquiring photoplethysmograph through an optical sensor with an aid of simple analog signal processing. Processed signals are sampled and transferred to personal computer.

Optical sensor consists of infra red light emitting diode and photo diode whose spectral density match. Light emitting diode exposes tissue with light of defined wavelength. Photo diode senses slight changes of illumination, which depends on light conductance changes of a tissue. This phenomena depends further on distension of capillaries and concentration of two types of hemoglobin, which again depends on blood pressure.

Retrieved signal from photo diode is processed with band pass filter and special amplifier, whose gain and offset can be digitally set with variable resistor and digital to analog converter. The resulted signal is digitalized in order to be transmitted to a personal computer. The scope of this diploma extends to the development of a suitable communication subsystem for efficient transfer of sampled data to a host PC. Further, microcontroller system must be able to supervise sensor activity and results of analog signal processing in order to detect and possibly eliminate the influence of parasitic influences from the environment. Adaptive electrical circuit has been developed based on the stated guidelines.

Keywords:

heart beat rate, Infrared light sensor, active analog filter, programmable gain, microcontroller system, data communications

1. Uvod

Avomatsko zbiranje podatkov, tako biometričnih kakor tudi drugih vrst, v zadnjih časih pridobiva na pomembnosti. S pomočjo zbranih podatkov se postavljajo nove hipoteze in teorije, stare pa se nadgrajujejo in dopolnujejo.

Merjenje in zbiranje biometričnih podakov je bilo že od nekdaj zahtevno opravilo, saj se merjene veličine med posameznimi osebami precej razlikujeje. Prav tako so tu vedno prisotni tudi spremenljivi zunanji vplivi, kateri meritve dodatno otežijo. Poleg merjene veličine dobimo ponavadi na senzorju tudi precej šuma, katerega moramo v čim večji meri odstraniti. Sistem za zajem biometričnih podatkov mora biti prilagodljiv tako, da se ti vplivi v čim večji meri izločijo. Prilagodljiv mora biti med samo meritvijo, ker se mora prilagajati razmeram v realnem času.

Zajem podatkov mora delovati robustno in zanesljivo, saj izvajamo meritve na osebah, katerih čas je večinoma omejen in dragocen.

Pri opravljanju meritev na osebah moramo biti poleg naštetega biti pozorni tudi na to, kakšno vrsto senzorja uporabljamo, saj je pomembno, da je njegova uporaba enostavna. Poleg tega se mora merjena oseba dobro počuti med opravljanjem meritev. Iz tega sledi, da mora biti za raziskovalne namene senzor temu prirejen, saj drugače težko prepričamo testne osebke v sodelovanje. Neudobje, povzročeno zaradi neprimernega senzorja, lahko povzroči tudi popačenje meritve, saj nanje neposredno vpliva.

Iz zgoraj naštega sledi, da je za uspešno zajemanje biometričnih podatkov odločilnih mnogo faktorjev. Navsezadnje je odločilna tudi cena ter dostopnost takšnega sistema, saj je pri raziskovalnih projektih pomembno, da izberemo podatke čim širše množice ljudi, s katerimi kasneje izvajamo analize.

Pri raziskovalnem primeru, katerega del je to diplomsko delo, bomo uporabljali te podatke za raziskovanje telesnega odziva na določena duševna stanja. Kadar smo razburjeni nam začne srce biti hitreje in močneje, kar lahko zaznamo s posebnim v ta namem izdelanim sistemom, kateri nam poda kot glavno merjeno veličino frekvenco bitja srca ter potek krvnega tlaka, katera nam bo služila kot pomožna veličina. Naprava se lahko uporablja kot poligraf (detektor laži), kjer ugotavljamo, kako se oseba odzove na postavljena vprašanja.

Tematika tega diplomskega dela obsega izgradnjo takšnega elektronskega vezja, ki bo upošteval naslednje zahteve:

- robustno in zanesljivo delovanje,
- prilagodljivost na različne dejavnike,
- udobnost senzorja in njegova enostavna uporaba.

Cilj je narediti sistem, ki bo omogočal zaznavanje spremembr frekvence utripa srca znotraj dveh ali treh period.

Naloge diplomskega dela so:

- izbira ustreznega senzorja,
- elektronsko vezje za krmiljenje senzorja,
- elektronsko vezje ze ojačenje in filtriranje signala, dobljenega iz senzorja,
- izdelati mikrokrmilniški sistem za nadzor delovanja vezij iz prejšnjih dveh točk, vzorčenje podatkov ter prenos podatkov na osebni računalnik in
- izdelati knjižnico za komunikacijo s mikrokrmilniškim sistemom.

Na strani osebnega računalnika se bodo zbrani podatki digitalno obdelali, izračunala se bo frekvenca in zaznala sprememba frekvence utripa srca. Ti koraki niso predmet te diplomske naloge.

2. Fotoplethysmografija

2.1. Splošno

Plethysmografija je veda, ki se ukvarja s preučevanjem tehnik merjenja prostorninskih sprememb različnih delov telesa. Najbolj pogosto se meri spremembe pri dihanju, se pravi prostorninske spremembe pljuč in prsnega koša. Sledijo spremembe prostornine arterijskih in venskih žil, kapilar ter srca. Fotoplethysmografija počne to na neinvaziven način z uporabo optičnih senzorjev. Temelji na dejstvu, da sprememba volumna povzroči spremembo absorptivnosti svetlobe Spremembo lahko merimo s pomočjo odbojnosti ali prepustnosti. S pomočjo odbojnosti se meri spremembe predvsem na rokah in licih, s prepustnostjo pa na prstih, ušesih in prstih na nogah. Kot svetlobni izvor uporabljamo LED diode različnih valovnih dolžin, kot detektor pa silicijevo fotodiodo. Sprejeti signal vsebuje informacijo o spremembi velikosti organa, kot posledico krvnega pretoka.

2.2. Kako deluje

Fotoplethymografija deluje na dejstvu, da so rdeče krvničke ali eritrociti sestavljene iz približno 15 % hemoglobina, kateri se uporablja za prenos kisika ostalim celicam[18]. Poznamo več različnih tipov hemoglobina, kot je oksihemoglobin (HbO2) , deoksihemoglobin (HbR) ter karboksihemoglobin (HbCO)[18]. Velika večina je oksihemoglobina ter deoksihemoglobina, ki predstavljata več kot 99 % celotnega hemoglobina[18].

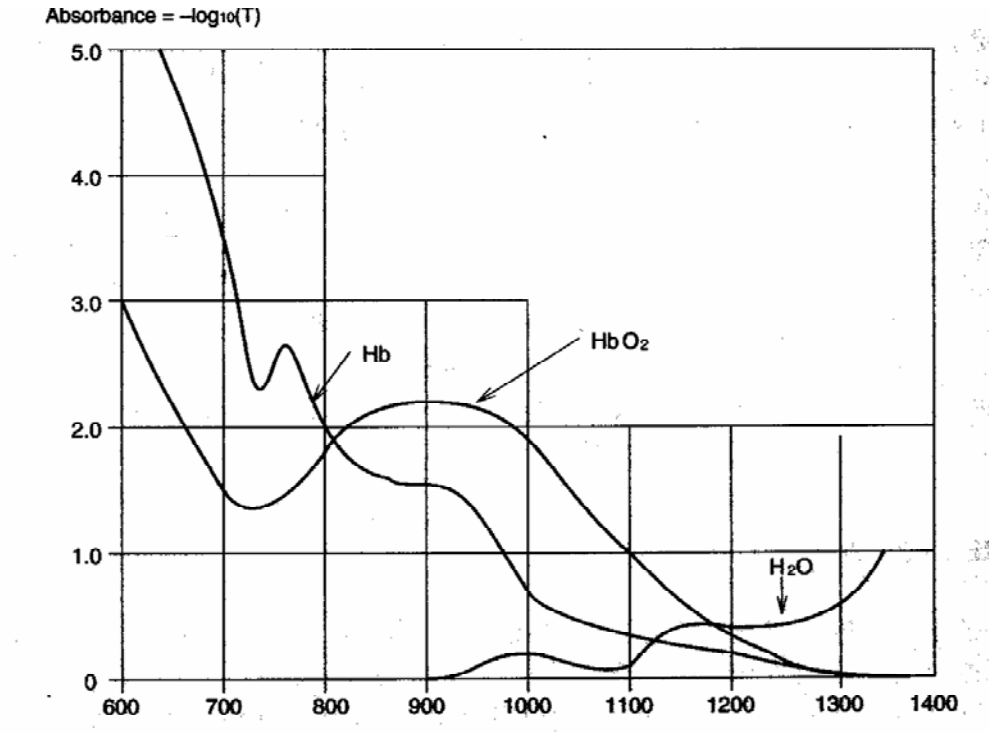
Prevodnost svetlobe je podana kot T

$$T = \frac{I}{I_0}$$

Kjer I_0 predstavlja vhodni svetlobni tok , I pa izhodni svetlobni tok[18]. Absorptivnost je podana kot

$$A = -\log_{10} T$$

Absorptivnost je za različne vrste ter različne valovne dolžine različna. Slika 1. ponazarja absorptivnost oksihemoglobina, deoksihemoglobina in vode v odvisnosti od valovne dolžine.

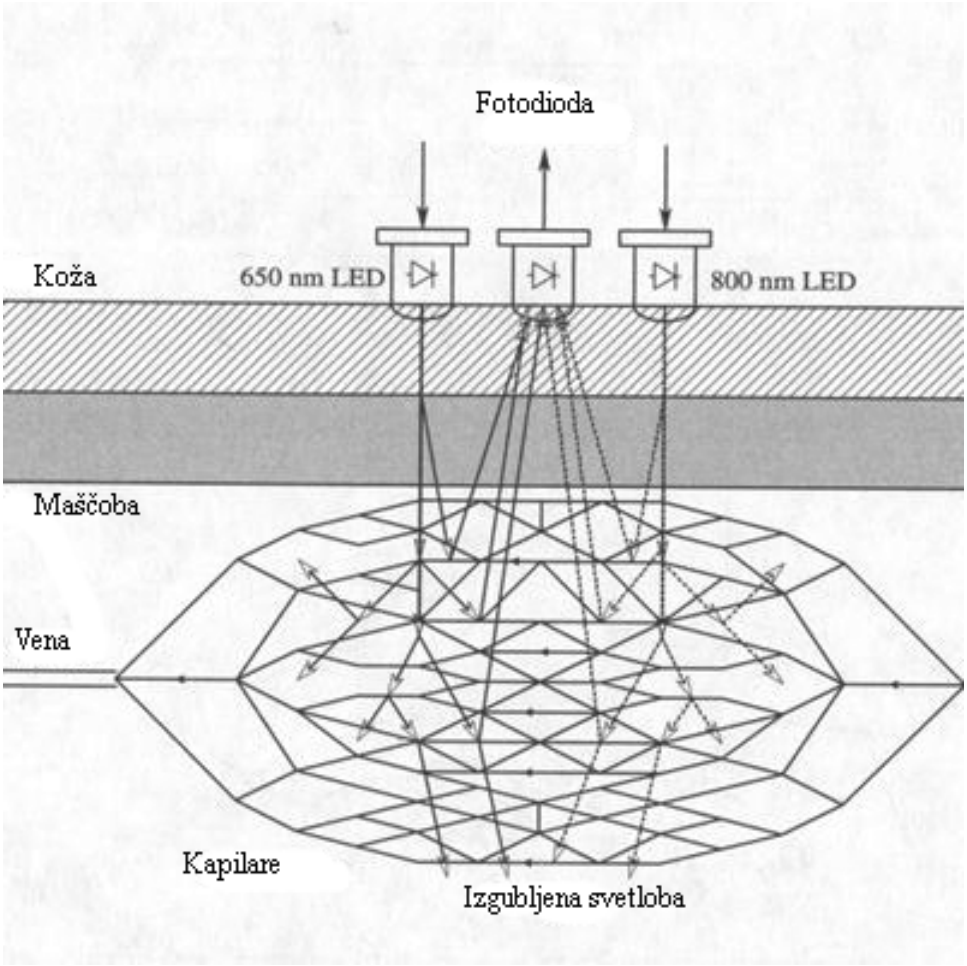


Slika 1: Absorptivnost hemoglobina in vode

Iz grafa št. 1 vidimo, da ob primerni izbiri valovne dolžine, lahko zaznavamo razliko med koncentracijama oksihemoglobina in deoksihemoglobina s pomočjo absorpcije svetlobnega toka skozi tkivo. Razlika med koncentracijama je precej odvisna od krvnega tlaka, torej se spreminja glede na srčni utrip.

Na absorptivnost prav tako vpliva tudi sprememba krvnega tlaka, saj se pod povišanim tlakom poveča količina krvi v kapilarah in tkivu[18]. Iz opisanega vidimo, da je svetlobna prevodnost tkiva precej odvisna od poteka krvnega tlaka.

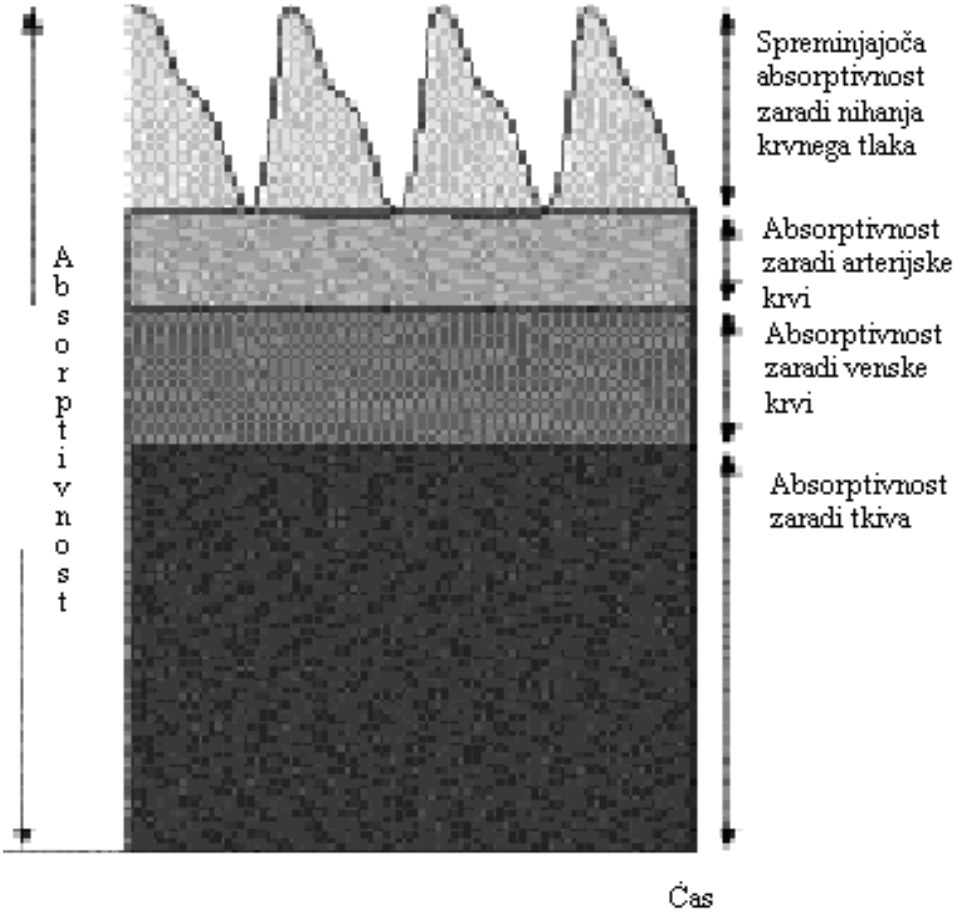
Slika št. 2 prikazuje princip delovanja odbojnega senzorja, kjer z dvema diodama različnih valovnih dolžin ocenimo koncentracijo tako oksihemoglobina kot deoksihemoglobina[18].



Slika 2: Zasnova senzorja

Vendar nas ne zanimajo koncentracije hemoglobina, ampak le frekvenca utripa srca, zato lahko senzor poenostavimo toliko, da uporabimo le en svetlobni vir. Absorptivnost je odvisna tudi od nekaterih drugih dejavnikov, vendar je ta veličina časovno nespremenljiva, tako da jo lahko s pomočjo ustreznega sita izločimo.

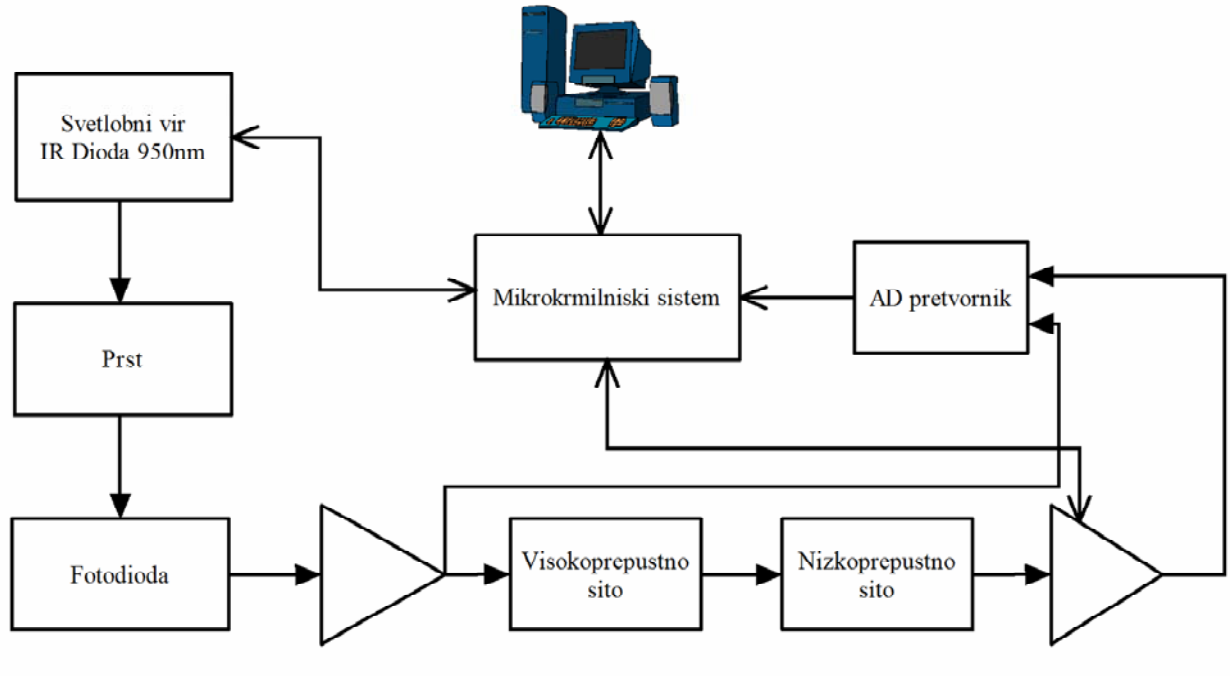
Slika št. 3 prikazuje absorptivnost v odvisnosti od spremembe krvnega tlaka, absorptivnost arterijske in venske krvi ter tkiva.



Slika 3: Absorptivnost hemoglobina

Iz slike št. 3 je razvidno, da je le manjši del absorptivnosti odvisen od spremembe krvnega tlaka.

2.3. Zasnova sistema za merjenje srčnega utripa



Slika 4: Shema sistema za zajem srčnega utripa

Sistem smo zasnovali na optičnem merjenju s pomočjo oddajne diode ter sprejemne fotodiode. Predhodno smo opisali, da zadostuje za ocenitev koncentracij hemoglobina absorptivnost dveh različnih valovnih dolžin. V našen primeru nas zanima le frekvenca utripa, zato lahko sistem poenostavimo in uporabimo eno valovno dolžino. Odločili smo se za infrardeč spekter, saj smo s tem pridobili nekaj na robustnosti, ker so foto diode, namenjene infradrečemu spektru, občutljive na precej ožji spekter svetlobe, kot foto diode za zaznavanje vidne svetlobe. Diodi smo postavili eno ob drugo, ter tako dobili reflektivni senzor. Oddajna dioda je krmiljena s pomočjo nastavljivega tokovnega vira, katerega tok nastavljamo s pomočjo mikrokrmilnika. Foto dioda je priključena na tokovno napetostni pretvornik, saj uporabljamo foto diodo kot tokovni vir. Iz slike št. 3. je razvidno, da je precejšnji del svetlobne prevodnosti konstanten, zato moramo signalu izločiti konstantno napetost s pomočjo visokoprepustnega sita. Na izhodu visokoprepustnega sita vsebuje signal precej šuma ter višjefrekvenčnih motenj, zato speljemo signal skozi nizkoprepustno sito z mejno frekvenco okoli 30Hz. Signal smo nato speljali skozi

ojačevalnik z nastavljivim ojačenjem ter nastavljivim premikom. Signal iz ojačevalnika s spremenljivim ojačenjem in signal iz tokovno napetostnega pretvornika smo pripeljali na analogno digitalni pretvornik. Po končani analogno digitalni pretvorbi se podatek shrani v izravnalnik, kjer čaka, da ga prenesemo na osebni računalnik.

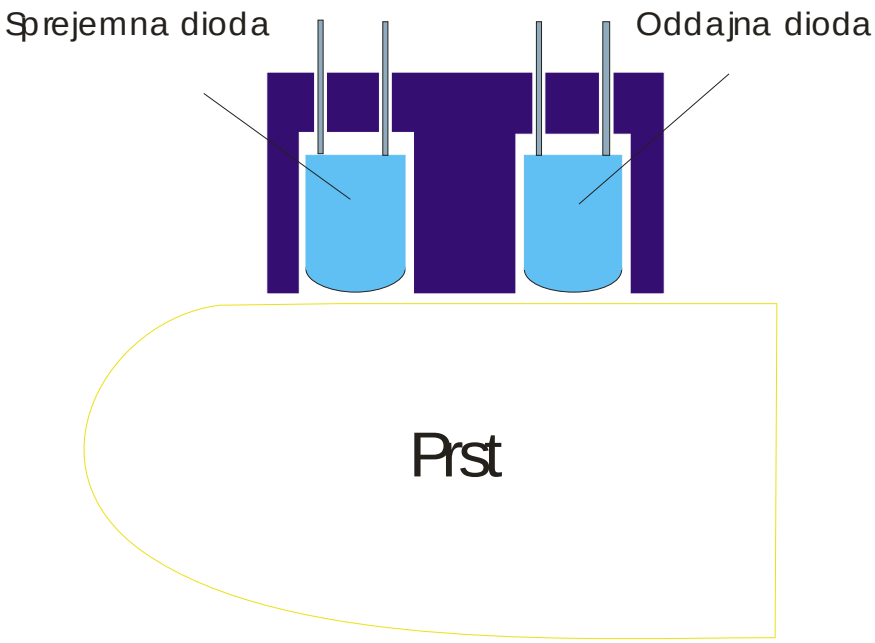
3. Senzor, krmilno vezje in analogno filtriranje signala

3.1. Senzor

3.1.1. Mehanska sestava senzorja

Sezor predstavlja stik med biološkim sistemom in elektronskim vezjem. Čim bolj točno in zanesljivo mora izmeriti merjeno veličino in čim manj vplivati na merjeni sistem. Zaželjeno je, da motnje ne vplivajo preveč na merjeno veličino.

Senzor je sestavljen iz infradreče LED diode, ki sveti v infrardečem spektru in sprejemne foto diode. Postavljeni sta vstran druga od druge, med njima je dobra optična izolacija.

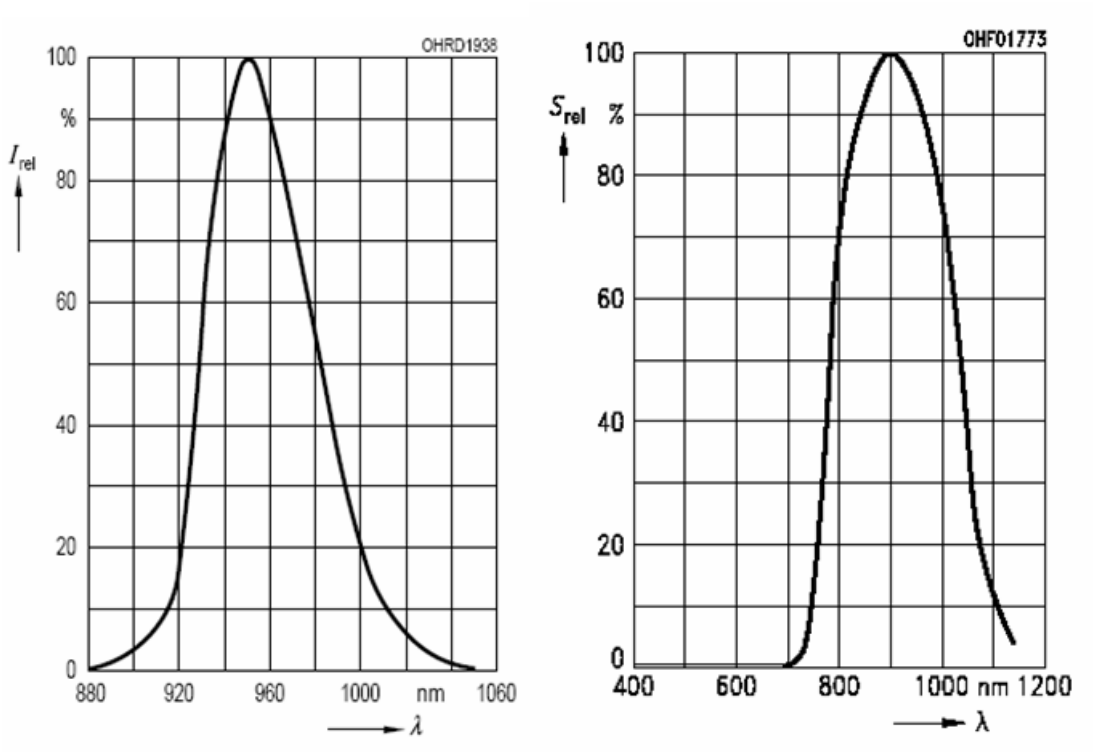


Slika 5: Skica senzorja

Ohišje senzorja smo naredili kot prikazuje slika št. 5 iz plastike, v katero smo zavrtali dve luknji, v katerih sta nameščeni diodi. Za dobro delovanje morata biti oddajna in sprejemna dioda dobro optično ločeni, poleg tega ne sme na sprejemno diodo preveč vplivati zunanja svetloba. Senzor se na prst pritrdi s pritrdilnim trakom.

3.1.2. Izbira ustreznih diod

Pri izbiri diod moramo biti posebej pazljivi, da se spektralna občutljivost foto diode pokriva s spektrom, katerega oddaja oddajna dioda. To nam prikazuje slika št. 6. Poleg tega mora biti ta spekter čim ožji. S tem ukrepom v precejšnji meri izločimo vpliv zunanje svetlobe.

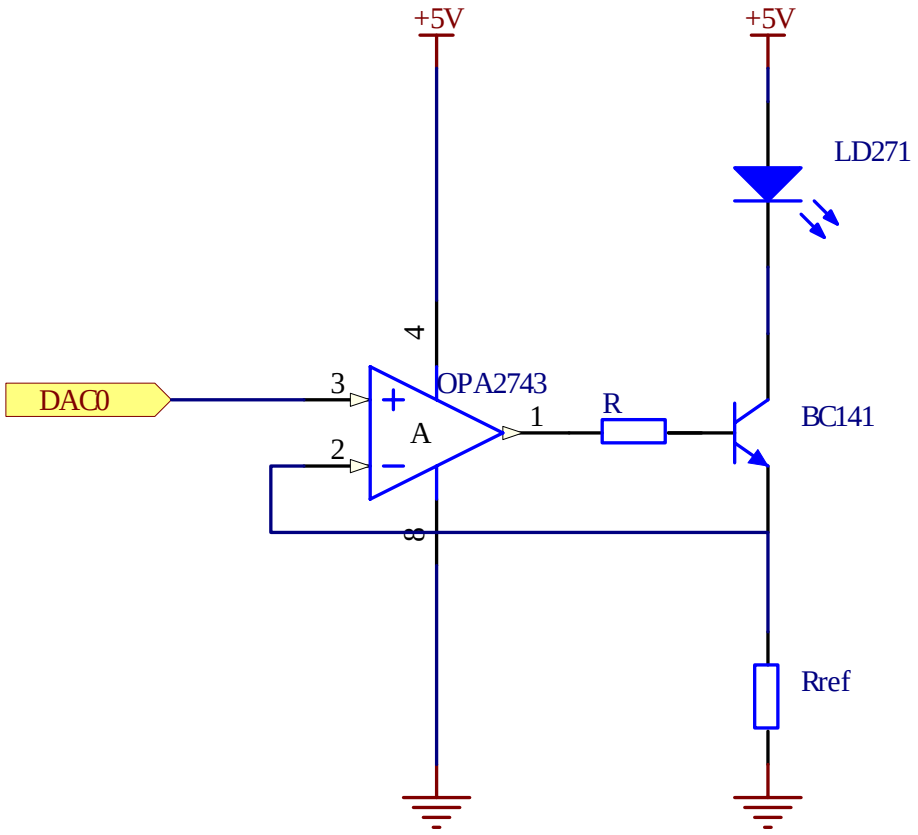


Slika 6: Spektralna občutljivost uporabljene oddajne in sprejemne diode

Uporabili smo oddajno diodo tipa LD271 [13], katera seva v območju med valovnima dolžinama 930 nm in 970 nm, največji svetlobni tok pa proizvede pri valovni dolžini 950 nm. Maksimalno sevalno moč 26 mW doseže pri toku 100 mA.

Sprejemna dioda je bila tipa SFH 203 [12], ki ima ravno tako najvišjo občutljivost pri 950 nm.

3.1.3. Krmilno vezje za oddajno diodo



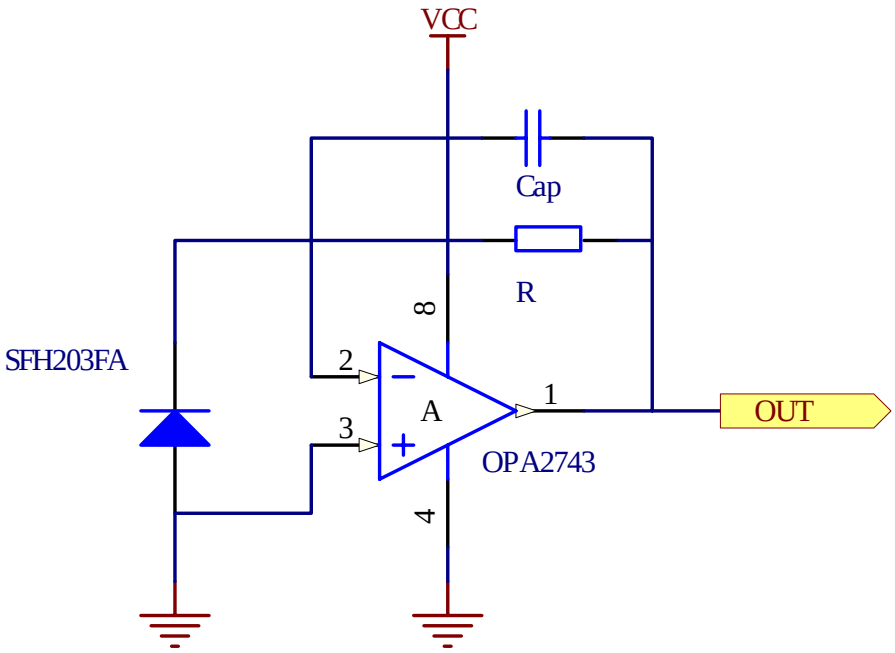
Slika 7: Shema električnega vezja za krmiljenje oddajne diode

Vezje na sliki št. 7 nam prikazuje krmilno vezje za krmiljenje oddajne diode.

Krmilno vezje oddajne diode opravlja nalogo tokovnega vira, kateremu lahko nastavimo tok preko digitalno analognega pretvornika. Nastavitev krmilnega toka je pomembna zato, da lahko dobimo na sprejemni diodi enako delavno točko ne glede na to, kakšna je svetlobna prevodnost tkiva merjene osebe.

V celotnem električnem vezju so uporabljani operacijski ojačevalniki tipa OPA2743 [14] proizvajalca Texas Instruments. Ta operacijski ojačevalnik ima *»Rail to Rail«* vhode in izhode tako, da lahko celotno vezje napajamo z eno napetostjo +5V.

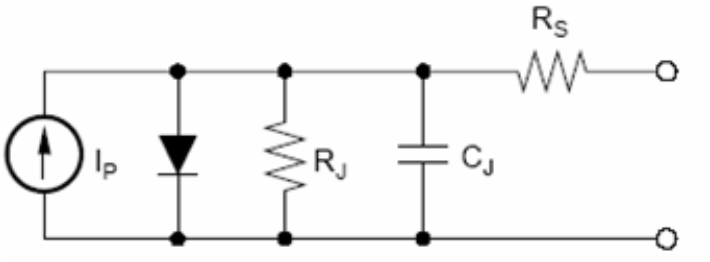
3.1.4. Električno vezje sprejemne diode



Slika 8: Shema električnega vezja za sprejemno diodo

Na sliki št. 8 je prikazano električno vezje sprejemne diode. Naloga vezja je pretvarjati tok v napetost. Foto dioda deluje v področju brez napetosti. Dobro delovanje tega vezja je ključnega pomena pri naši nalogi, saj signal iz tega pretvornika ojačimo za približno 50 dB. Zato je bilo potrebno izvesti analizo povratne zanke.

Na sliki št. 9 je prikazano nadomestno vezje foto diode, kjer I_P predstavlja svetlobno generirani tok, R_J vzporedno upornost, C_J spojno kapacitivnost , Rs pa zaporedno upornost.



Slika 9: Nadomestno vezje foto diode

Fotodioda se obnaša kot tokovni vir, katerega tok je odvisen od svetlobnega toka, prejetega na fotodiodi.

Vezje na sliki št. 8 je podobno invertirajočemu ojačevalniku brez vhodnega upora. Če predpostavimo, da je diferenčna napetost na vhodu operacijskega ojačevalnika $u_d = 0$, lahko po Kirchoffovem zakonu zapišemo enačbo, katera predstavlja prenosno funkcijo med vhodnim tokom ter izhodno napetostjo.

$$u_{izh} = -Ri_{vh}$$

Enačba 1: Prenosna funkcija med tokom in napetostjo

Izhodna napetost je torej odvisna od svetlobnega toka in od upornosti povratne zanke. Z namenom, da dosežemo veliko transrezistanco tokovno napetostnega pretvornika mora biti upornost povratne zanke čim večja, v kolikor to dopuščajo drugi dejavniki. Pri veliki upornosti v povratni vezavi se pojavijo težave, kot je temperaturno lezenje tokovnega premika in s tem lezenje premika izhodne napetosti. Vendar v našem primeru to ni kritično, saj smo kasneje signal speljali skozi visokoprepustno sito, zato te napake ni bilo potrebno kompenzirati.

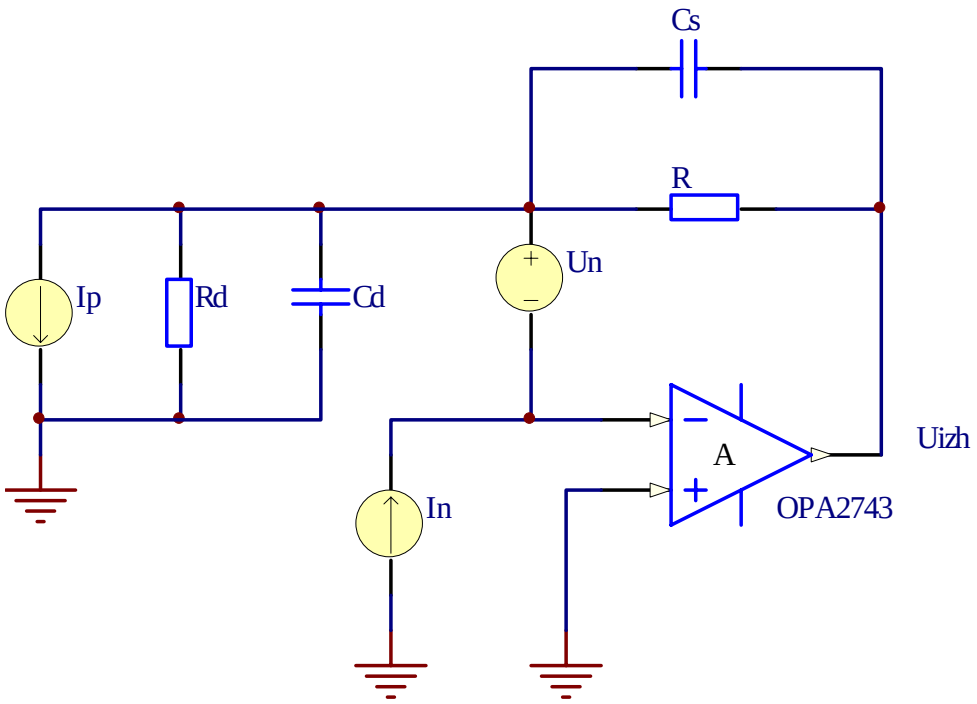
Upornost povratne zveze določa pasovno širino ter šum pretvornika. Johnsonov šum, generiran na uporu, je definiran z izrazom $u = \sqrt{KTR}$, ojačenje pa je sorazmerno uporu R. Iz tega vidimo, da z višanjem upornosti res povečujemo šumno napetost, vendar s tem tudi povečujemo ojačenje [24]. Z velikim uporom v povratni zanki pridobimo na razmerju signal šum z razmerjem $\sqrt{R}$ [24]. Poleg

termičnega šuma v uporu ima tudi operacijski ojačevalnik svoje vire šuma. To sta vhodna šumna napetost ter šumni tok, ki sta prikazana na sliki št. 10.

$$u_{izh} = I_p R + I_n R + \frac{1 + j\omega C_D}{1 + j\omega C_S}$$

Enačba 2: Izhodna napetost tokovno napetostnega pretvornika

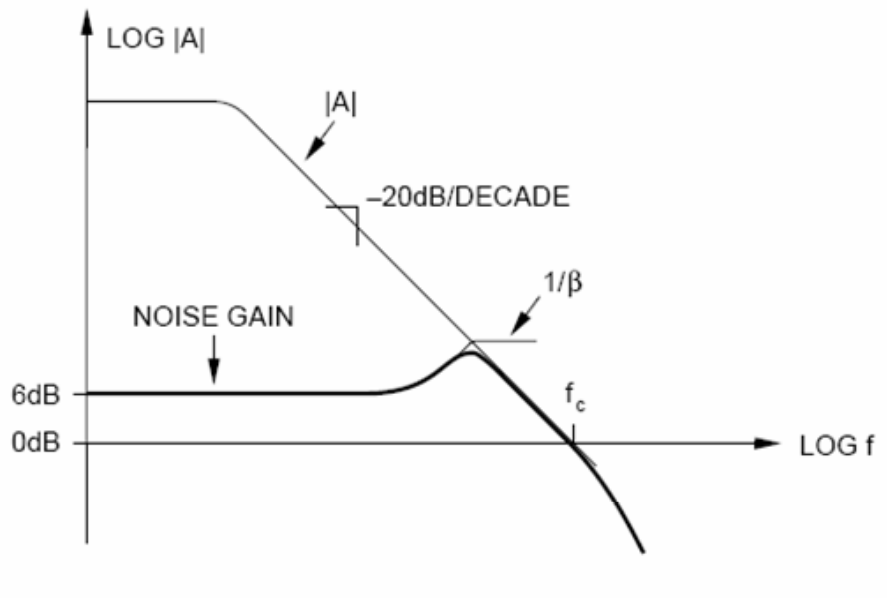
Šumna napetost je pri nizkih frekvencah ojačena s faktorjem $1 + \frac{R_1}{R_D}$, kar je pri veliki upornosti diode zanemarljivo.



Slika 10: Šumni viri operacijkega ojačevalnika

Pri visokih frekvencah se ojačenje šuma spremeni, takrat velja izraz $1 + \frac{C_D}{C_S}$, kjer C_D predstavlja fotodiodno kapacitivnost, C_S pa stresano kapacitivnost in kapacitivnost povratnega upora. Iz tega vidimo, da je šumna napetost precej

ojačena pri visokih frekvencah, saj je diodna kapacitivnost precej večja od stresane kapacitivnosti. Šumna pasovna širina je tako omejena samo s pasovno širino operacijskega ojačevalnika, katera je precej visoka [24]. Razmere prikazuje graf št. 1.



Graf 1: Ojačenje šuma zaradi kapacitivnega delilnika

Iz grafa 1 lahko vidimo, da se ojačenje šuma veča z naraščujočo frekvenco, dokler ni ojačanje odrezano zaradi zmanjšanega ojačenja operacijskega ojačevalnika. Za našo aplikacijo potrebujemo precej manjšo pasovno širino od pasovne širine operacijskega ojačevalnika, zato smo z dodanim kondenzatorjem v povratni zanki zmanjšali celotno pasovno širino tokovno napetostnega pretvornika. S tem se spremeni tudi kapacitivnost povratne vezave ter kapacitivni delinik. Mejna frekvenca takšnega tokovno-napetostnega pretvornika je $f = \frac{1}{2\pi RC}$ [24].

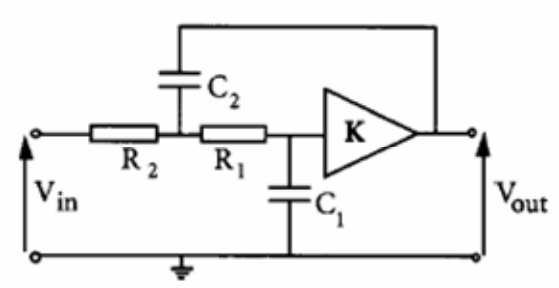
Šum je v precejšnji meri izločen s pomočjo nizkoprepustnega sita 8. reda z mejno frekvenco 30Hz, skozi katerega je speljan signal iz pretvornika.

Za optimalno delovanje pretvornika, lahko svetlobni tok, ki je sprejet na fotodiodo, nastavljamo preko toka skozi oddajno diodo.

3.1.5. Visokoprepustno sito

Signal iz vezja fotodiode nam daje podatek o skupnem svetlobnem toku. Podatek o srčnem utripu predstavlja zelo majhna nihanja svetlobega toka v primerjavi s skupnim svetlobnim tokom. Z visokoprepustnim sitom izločimo enosmerno komponento napetosti, tako da nam ostane le spremenljiva napetost, katera predstavlja spremembe svetlobnega toka.

Visokoprepustni filter smo naredili s pomočjo Sallen-Key člena [4], katerega sestavljajo pasivni elementi in operacijski ojačevalnik. Osnovno vezje takšnega člena je prikazano na sliki št. 11, enačba št. 3 pa nam prikazuje prenosno funkcijo tega člena.



Slika 11: Sallen-Key člen

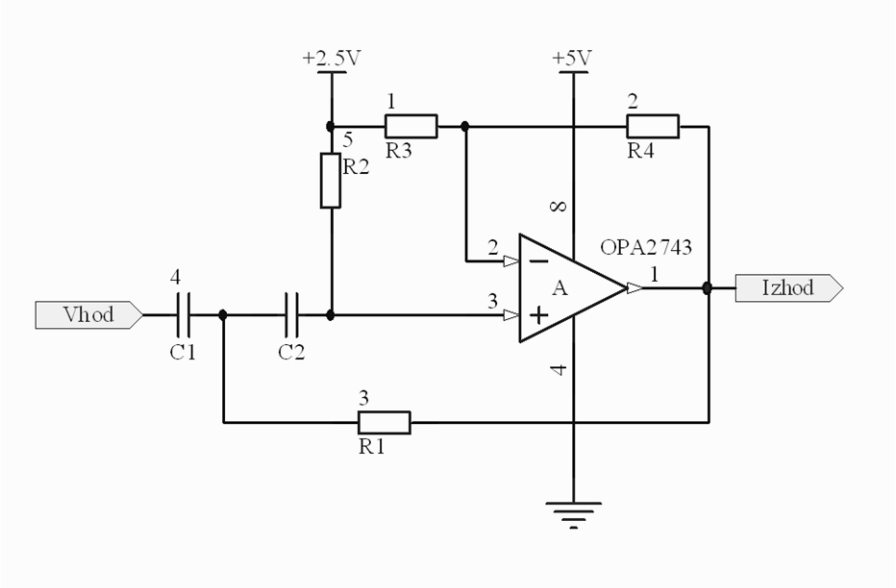
$$H(s) = \frac{K\dfrac{1}{C_1 C_2 R_1 R_2}}{s^2 + s\left[\dfrac{1}{C_2 R_2} + \dfrac{1}{C_2 R_1} + \dfrac{1-K}{C_1 R_1}\right] + \dfrac{1}{C_1 C_2 R_1 R_2}}$$

Enačba 3: Prenosna funkcija Sallen-Key člena

$$H(s) = \frac{k}{s^2 + a_1 s + a_2}$$

Enačba 4: Poljubna prenosna funkcija člena 2. reda brez ničel

Če primerjamo enačbi št. 3 in št. 4 vidimo, da lahko s primerno izbiro pasivnih elementov realiziramo prenosno funkcijo sita s končnim odzivom na enotin impulz 2. reda. Pri projektu smo uporabili visokoprepustno sito namesto nizkopasovnega, torej smo zamenjali kondenzatorje z upori in obratno. Sallen-key vezje lahko modificiramo tako, da dodamo dva upora in s tem dobimo elektrometrski ojačevalnik, s katerim lahko nastavljamo izhodno ojačenje. Tako smo dobili vezje visokoprepustnega sita, katerega prikazuje slika št. 12.



Slika 12: Vezje visokoprepustnega sita

Izbira tipa sita zahteva premislek, ter racionalno odločitev med različnimi lastnostmi sit. Lastnosti sit lahko razdelimo v dve skupini, glede na to ali se lastnosti odrazijo v časovnem ali frekvenčnem prostoru. Glavni frekvenčni lastnosti sit sta amplitudna in fazna karakteristika, lastnosti v časovnem prostoru pa so čas vzpona, maksimalni prevzpon in čas iznihavanja.

V našem primeru je pomembno, da sito ne vpliva na obliko signala, se pravi, da ne sme popačiti signala v časovnem prostoru. Zaradi tega smo se odločili za Besselovo sito, ker najmanj popači obliko signala, medtem ko Čebiševo in Butterworthovo sito povzročita prevelik prevzpon na strmih bokih signala [8]. Zaradi tega ima Besselovo sito slabše lastnosti v frekvenčnem prostoru, kar pa za našo aplikacijo ni odločilnega pomena.

Vrednosti elementov izračunamo iz enačb št. 4 in št. 5, koeficienta k_1 in k_2 pa preberemo iz tabele št. 1.

# poles		Bessel		Butterworth		Chebyshev	
		k_1	k_2	k_1	k_2	k_1	k_2
2	stage 1	0.1251	0.268	0.1592	0.586	0.1293	0.842
4	stage 1	0.1111	0.084	0.1592	0.152	0.2666	0.582
	stage 2	0.0991	0.759	0.1592	1.235	0.1544	1.660
6	stage 1	0.0990	0.040	0.1592	0.068	0.4019	0.537
	stage 2	0.0941	0.364	0.1592	0.586	0.2072	1.448
	stage 3	0.0834	1.023	0.1592	1.483	0.1574	1.846
8	stage 1	0.0894	0.024	0.1592	0.038	0.5359	0.522
	stage 2	0.0867	0.213	0.1592	0.337	0.2657	1.379
	stage 3	0.0814	0.593	0.1592	0.889	0.1848	1.711
	stage 4	0.0726	1.184	0.1592	1.610	0.1582	1.913

Tabela 1: Parametra k_1 in k_2 visokoprepustnih sit [8]

Za Besselovo sito 2. reda sta konstanti k_1=0.1250 k_2=0.268.

$$R_1 = R_2 = R$$
$$C_1 = C_2 = C$$
$$R = \frac{k_1}{C \cdot f_c} \ , \ R_f = R_1 k_2$$

Enačba 5: Prenosna funkcija Sallen-Key člena

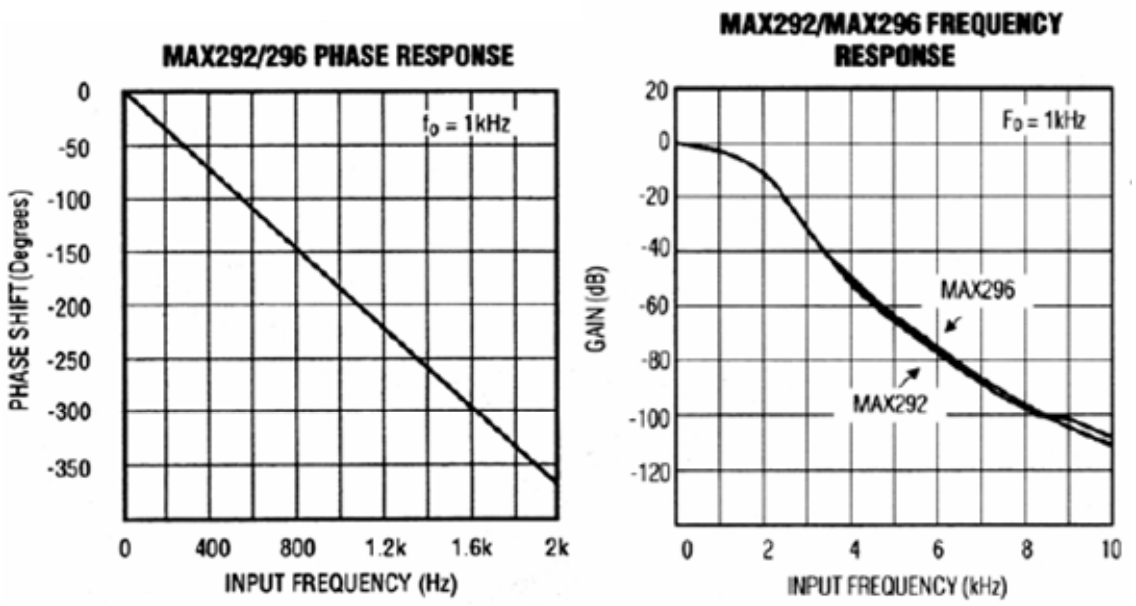
Mejno frekvenco smo določili na 0,5 Hz, kar približno ustreza frekvenci 30 utripov na minuto.

3.1.6. Nizkoprepustno sito

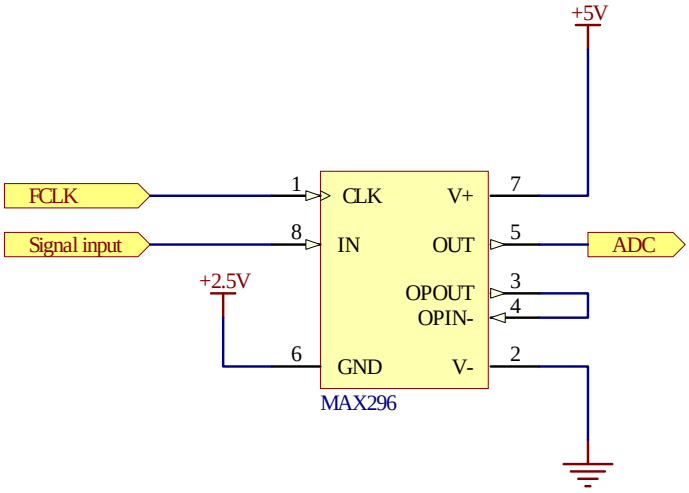
Signal na izhodu iz visokoprepustnega sita vsebuje precej motenj in šuma. Glavni vir motenj je nekonstantni zunanji izvor svetlobe, kot so neonske žarnice, katere utripajo s frekvenco 100 Hz. Z namenom, da bi motnje in šum v čim večji meri

izločili, smo se odločili signal speljati skozi nizkoprepustno sito. Mejno frekvenco nizkoprepustnega sita smo določili nai 30 Hz, kar je zadosti visoko, da se pomembne komponente našega signala ne izgubijo. Nizkoprepustno sito smo zgradili s pomočjo integriranega vezja MAX296 [15]. To vezje vsebuje Besselovo sito 8. reda, mejno frekvenco pa se nastavlja s pomočjo urinega takta. Slika št. 13 prikazuje amplitudni in fazni odziv filtra pri mejni frekvenci f_0 = 1 kHz. Zaradi izgradnje sistema z eno napajalno napetostjo smo morali priključitev nekoliko prirediti, kakor to prikazuje slika št. 14. Kontakt za pozitivno napetost smo priključili na napajalno napetost, kontakt za negativno napetost smo priključili na ozemljitev, ozemljitveni kontakt pa na potencial 2.5V . Pri nizkih napajalnih napetostih predstavlja problem zmožnost sledenja izhoda vhodu, kadar se vhodna napetost približuje napajalni napetosti. Uporabljano integrirano vezje ima razpon izhodne napetosti pri uporabljeni napajalni napetosti med 1V in 4V. Analogno digitalni pretvornik imamo priključen na pozitivno referenčo napetost 5V, negativno pa na 0V. Iz zgoraj navedena je razvidno, da moramo signal napetostno ojačiti, s pomočjo ojačevalnika, kateri ima razpon izhodne napetosti enak razponu med napajalno napetostjo in ozemljitvijo.

Urin takt za izbiro mejne frekvence smo sintetizirali s pomočjo mikrokrmilnika, kar bo opisano v poglavju o mikrokrmilniškem sistemu.



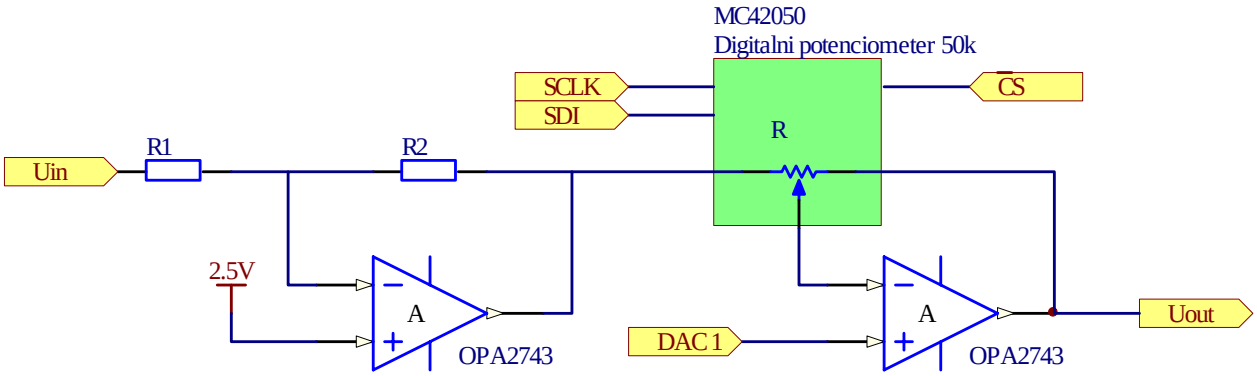
Slika 13: Amplitudni in fazni odziv nizkoprepustnega filtra



Slika 14: Priključitev integrianega vezja MAX296

3.1.7. Digitalno nastavljivo ojačenje

Na izhodu nizkoprepustnega sita dobimo signal, katerega amplituda je delno nastavljiva preko toka skozi oddajno IR diodo. Z namenom, da bi signal lahko kar najbolje zajeli, smo se odločili narediti digitalno nastavljivo ojačenje, katerega lahko uravnava program na osebnem računalniku. Nastavitve vodi osebni računalnik z namenom, da bo lahko algoritem, kateri bo vršil analizo, optimalno nastavil območje signala.



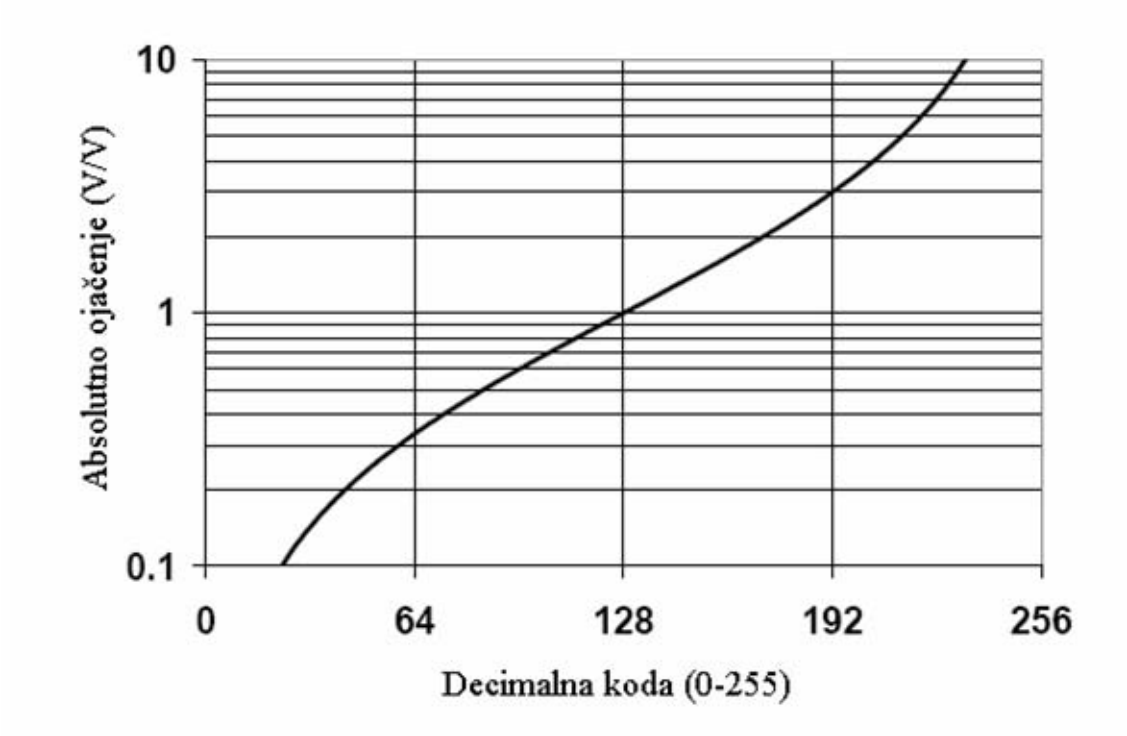
Slika 15: Izvedba nastavljivega ojačenja

Vezje, katero se obnaša kot ojačevalnik z nastavljivim ojačenjem in odmikom, je prikazano na sliki št. 8. Levi del vezja, predstavlja enostaven izmenični invertirajoči ojačevalnik s fiksnim ojačanjem, desni del vezja pa invertirajoči ojačevalnik z nastavljivim ojačenjem. Spremenljivo ojačenje smo naredili s pomočjo digitalno nastavljivega potenciometra MC42050 proizvajalca Microchip [11]. Upornost lahko nastavljamo v 256 korakih med 0 kΩ in 50 kΩ. Položaj drsnika nastavljamo preko serijskega vmesnika SPI. Ojačevalnik s spremenljivo napetostjo je invertirajoči, kjer ena stran delilnega razmerja predstavlja vhodni upor, druga stran delilnega razmerja pa upor v povratni zanki. Ojačanje se izračuna po enačbi št. 6, kjer U_{AB} predstavlja celotno upornost potenciometra, D_N pa trenutni položaj drsnika.

$$u_{izh} = \frac{R_{AB}\left(256 - D_n\right)}{256} u_{vh} + \frac{R_{AB} D_N}{256} u_{ref}$$

Enačba 6: Prenosna funkcija ojačevalnika s spremenljivim ojačanjem

Graf na sliki št. 16 prikazuje ojačanje v odvisnosti od položaja drsnika.



Slika 16: Ojačenje glede na nastavitev drsnika

4. Mikrokrmilniško vezje

4.1. Naloga mikrokrmilniškega vezja

V prejšnjem poglavju smo opisali senzor ter krmilno vezje, katerega naloga je pretvorba krvnega tlaka v električno napetost. Glavna naloga mikrokrmilniškega sistema je vzorčenje te napetosti, ter prenos vzorcev na osebni računalnik. Poleg tega mikrokrmilniški sistem upravlja in nadzoruje delovanje krmilnega vezja senzorja.

4.2. Mikrokrmilnik PIC18F4520

Jedro mikrokrmilniškega sistema predstavlja mikrokrmilnik PIC 18F4520 proizvajalca Microchip [10]. Za ta mikrokrmilnik smo se odločili na podlagi zahtev po velikem številu izhodno vhodnih enot, ceni ter dostopnosti razvojnih orodij.

Mikrokrmilnik je zgrajen na osnovi zmogljivega 8 bitnega mikroprocesorja, kateri je zgrajen v RISC [17] arhitekturi. Pomnilnik je zasnovan po načelu Harvard [17] arhitekture torej ima notranje vodilo ločeno na podatkovno in programsko. Podatkovno vodilo je široko 8 bitov, programsko pa 16 bitov. Podatkovni pomnilnik ima zgrajen v obliki registrov in obsega 1536 zlogov. To omogoča hitro delovanje saj se ukazi naloži/shrani ponavadi počasneje izvajajo kot ukazi, ki delujejo neposredno z registri. Programski pomnilnik je velik 32 K zlogov, se pravi, da lahko vanj shranimo 16 K ukazov. Posebnost arhitekture PIC mikrokrmilnika je strojni programski sklad, kateri omogoča hitrejše preklapljanje med glavnim in prekinitvenim izvajanjem.

Takt lahko določimo na več možnih načinov. Najvišja frekvenca pri kateri proizvajalec zagotavlja stabilno delovanje je 40 MHz. Pri tem taktu zmore mikrokrmilnik izvesti približno 10 milijonov operacij na sekundo. Večina ukazov se izvrši v enem ukaznem ciklu, ki traja štiri urine cikle.

Prekinitve so grupirane v dve skupine glede na prioriteto prekinitve. Vsem izvorom prekinitev lahko določimo katera skupina naj se proži, bodisi prekinitev z visoko ali nizko prioriteto.

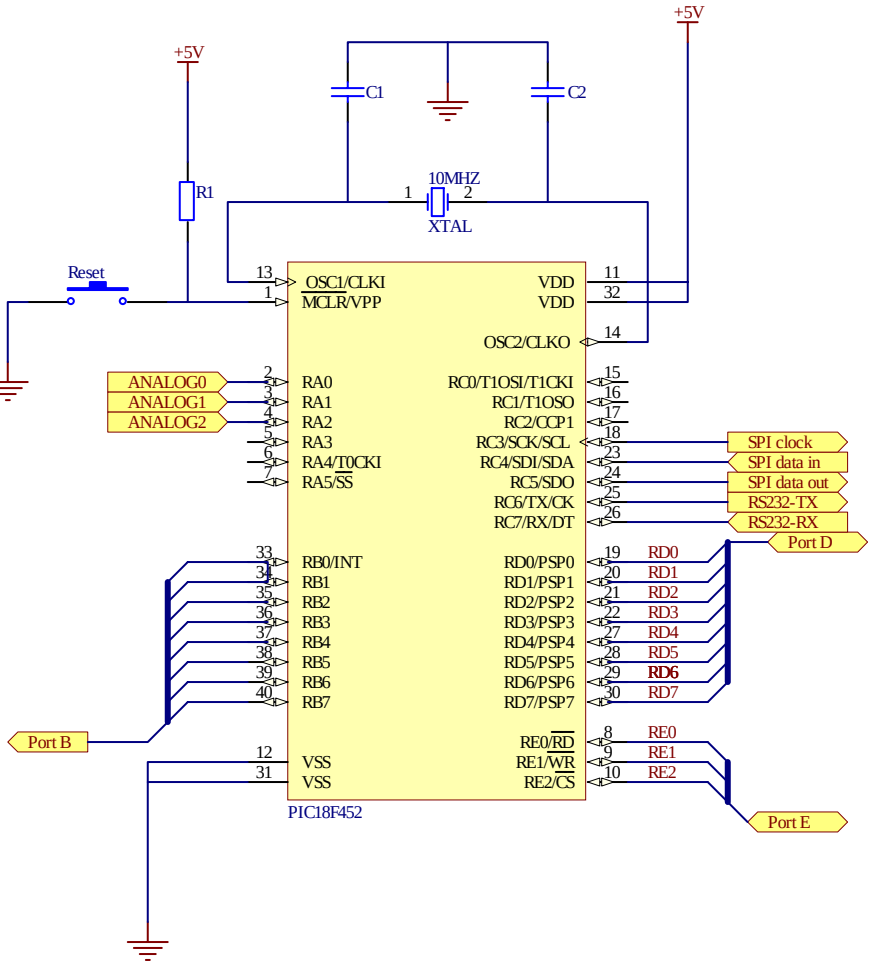
Izbrani mikrokrmilnik je bogato opremljen s perifernimi napravami. Ključne za zastavljeno nalogo so:

- 10 bitni analogno digitalni pretvornik,
- štirje različni časovniki,
- generiranje pulzno širinsko moduliranega signala (PWM),
- asinhroni komunikacijski vmesnik,
- sinhroni komunikacijski vmesnik z možnostjo povezovanja SPI in IIC naprav,
- trije vhodi za prekinitve,
- paralelni vmesnik.

Poleg tega mikrokrmilnik odlikuje majhna poraba električne energije in robustno delovanje. Slaba stran izbranega mikrokrmilnika je razmeroma majhen podatkovni pomnilnik in težaven priklop dodatnega večjega zunanjega pomnilnika.

4.3. Priključitev mikrokrmilnika

Mikrokrmilnik potrebuje za delovanje pripadajoče vezje za napajanje in generiranje urinega signala. Vezje za generiranje urinega signala bi lahko izpustili in s tem izkoristili notranje RC vezje za generiranje urinega signala. Vendar na takšen način ne moremo popolnoma izkoristiti zmogljivosti mikrokrmilnika. Na podlagi tega smo se odločili narediti vezje za generiranje urinega signala s pomočjo kristala. Na mikrokrmilnik moramo priključiti še vezje za reinicializacijo, kakor prikazuje slika št. 17.

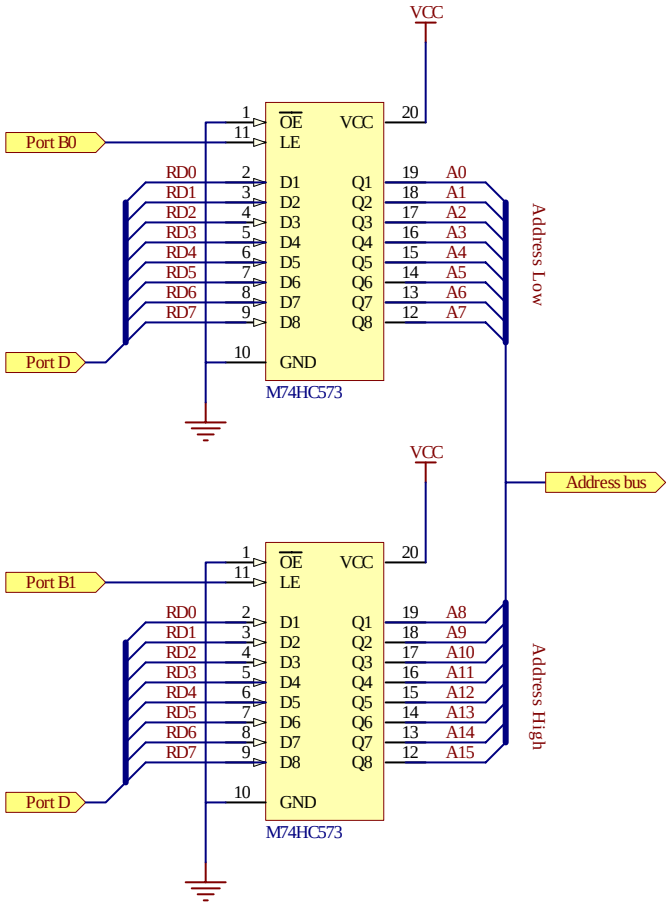


Slika 17: Mikrokrmilnik in priključitev v vezje

4.4. Priklop zunanjega pomnilnika

Prenos podatkov med mikrokrmilniškim sistemom in osebnim računalnikom poteka v paketni obliki. Velikost paketa je odvisna od frekvence odjemanja vzorcev. Signal iz senzorja vzorčimo v določenih časovnih razmakih, zato moramo te podatke shraniti v medpomnilnik. Ta medpomnilnik mora biti dovolj velik, da ne izgubimo kakšnega vzorca, če osebni računalnik dlje časa ne pošlje ukaza za prenos paketa vzorcev. V mikrokrmilniku imamo na voljo približno 1 Kb prostega pomnilnika, kar ne zadošča za našo aplikacijo. Na podlagi dobavljivosti pomnilniških modulov, enostavnosti uporabe ter cenenosti smo se odločili uporabiti statični ram,Hitachi HM 62256 [16] velikosti 32 kilo zlogov. Z namenom, da bi privarčevali z

mikrokrmilnikovimi izhodno vhodnimi kontakti, smo se odločili priključiti zunanji pomnilnik preko zatičev tipa 74HC573 [20]. Za vpisovanje in branje iz pomnilniškega modula potrebujemo 8 podatkovnih kontaktov, 15 naslovnih kontaktov, ter 3 kontakte za nadzor branja in vpisovanja. Potrebujemo 26 kontaktov, od tega 11 vhodno izhodnih, ter 15 izhodnih. Z zatiči smo zaradi enostavnosti realizirali samo izhodne kontakte, kateri predstavljajo naslovno vodilo, kakor je prikazano na sliki št. 18.

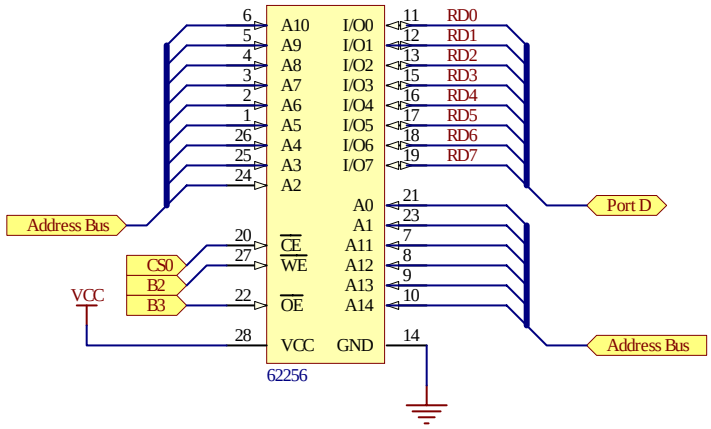


Slika 18: Shema električnega vezja za razširitev pomnilnika

Vpis in branje iz zunanjega pomnilnika poteka tako, da se najprej prenese v zatič zgornjih 7 bitov naslova in potem še spodnjih 8 bitov.

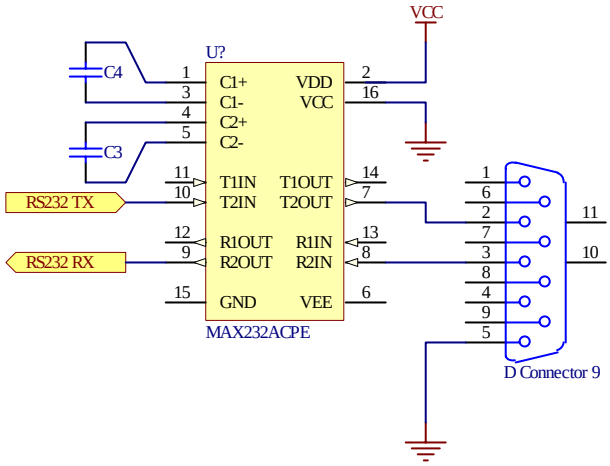
Zatič nadziramo s pomočjo dveh kontaktov LE(»latch enable«) ter OE (»output enable«). Kadar je kontakt LE (»latch enable«) zatiča v visokem stanju, postavimo željeni podatek na vhodne kontakte zatiča, kateri so priključeni na mikrokrmilniško

vodilo. Ob prehodu kontakta LE v nizko stanje, si zatič zapomni predhodno stanje vodila. Ta sistem nam omogoča, da nastavljamo izmenoma izhode enega in drugega zatiča. Podatke lahko vpisujemo in beremo iz pomnilnika, kadar izhoda zatičov predstavljata pravilen naslov. Branje in pisanje nadzorujemo s kontakti OE (»output enable«) ter WE (»write enable«). Podatki se berejo in pišejo preko vodila, kateri je priključen na kontakte D mikrokrmilnika. Kontakti so priključeni na mikrokrmilnik, tako kot prikazujeta sliki št. 8 in št. 19.



Slika 19: Kontakti in priključitev zunanjega pomnilnika

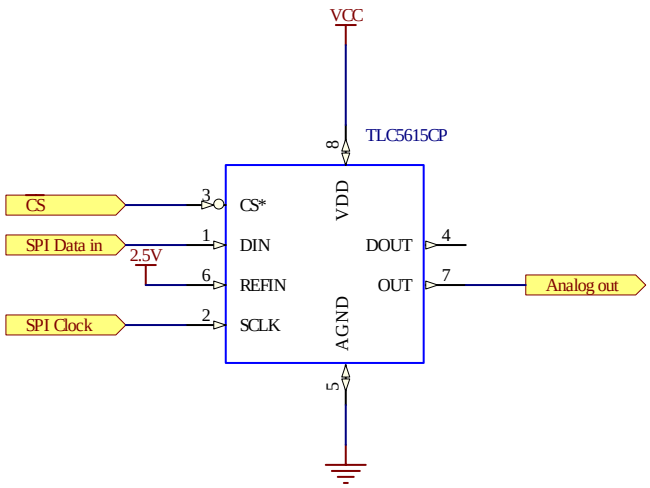
4.5. Vezje za komunikacijo z osebnim računalnikom



Slika 20: Vezje za komuniciranje z osebnim računalnikom

Mikrokrmilnik ima vgrajen vmesnik za asinhrono serijsko komuniciranje z drugimi napravami. Če želimo, da mikrokrmilnik komunicira z osebnim računalnikom, moramo prirediti napetostne nivoje, saj uporablja mikrokrmilnik CMOS nivoje. Standard RS232 predpisuje napetostne nivoje za logično 0 med +3V in +25V, za logično 1 pa med -3V in -25V. Za pretvorbo nivojev smo izbrali integrirano vezje MAX232 [21]. Priključitev na mikrokrmilniške kontakte in na DB9 konektor prikazuje slika št. 20.

4.6. Priključitev Digitalno analognih pretvornikov



Slika 21: Vezje za komuniciranje z osebnim računalnikom

Za nadzor delovanja senzorja potrebujemo mikrokrmilniško nastavljive spremenljive napetostne vire. Prvi napetostni vir potrebujemo za nastavitev toka skozi oddajno diodo, kjer z napetostjo nadziramo tokovni vir.

Zaradi enostavne uporabe smo se odločili uporabiti digitalno analogni pretvornik TLC5615 [22], kateri se na mikrokrmilnik priključi preko zaporednega SPI vmesnika. SPI vmesnik uporablja za povezovanje 3 kontakte, eden je za komunikacijo proti periferiji, drugi je za komunikacijo od periferije proti mikrokrmilniku in tretji je za urin signal. Za priključitev digitalno analognega pretvornika potrebujemo samo 2 kontakta, in sicer zato, ker prenašamo podatke samo v smeri od mikrokrmilnika proti pretvorniku. V našem primeru imamo v sistemu dva digitalno analogna pretvornika, zato potrebujemo tudi logiko za izbiro

integriranega vezja (CS – »chip select«), katera je izvedena z integriranim vezjem 74HC138 [23].

5. Strojno-programska oprema mikrokrmilniškega sistema

5.1. Naloga mikrokrmilniškega programa in razvojno okolje

Naloga mikrokrmilnika je shranjevanje in prenos podatkov ter nazdiranje delovanja analognega dela vezja. Zato moramo na mikrokrmilnik naložiti ustrezen program, kateri opravlja zadane naloge. Programska oprema za mikrokrmilnik je zaradi enostavnosti napisana v jeziku C. Za prevajanje kode smo uporabili prevajalnik C18 proizvajalca Microchip. [19] Razlogi za uporabo tega okolja izvirajo predvsem iz dejstva, da ima ta prevajalnik najboljšo podporo s strani proizvajalca. Poleg tega je prevajalnik vgrajen v razvojno okolje MPLAB[19], kateri se uporablja za razhroščevanje in nalaganje programa na mikrokrmilnik. Tako imamo vsa orodja, katera potrebujemo za razvoj mikrokrmilniške aplikacije na enem mestu. Prevajalniku je priložena tudi posebna knjižnica, katera vsebuje vse potrebne funkcije za uporabo mikrokrmilniških perifernih naprav in vmesnikov. Pisanje aplikacije v programskem jeziku C sicer pomeni, počasnejše izvajanje programa v primerjavi z zbirniškim jezikom, vendar je uporabljeni mikrokrmilnik dovolj hiter, da to ne predstavlja ovire**.**

5.2. Zahteva po izvajanje v dejanskem času

Predno smo pričeli razvijati strojno programsko opremo smo morali dobro premisliti, kako zgraditi mehanizme za zagotavljanje obdelave podatkov in nadzora v dejanskem času.

Mikrokrmilnik mora biti sposoben hkrati izvajati več nalog, katere so:

- vzorčenje in shranjevanje podatkov o krvnem tlaku,
- nadzorovanje delovanja senzorja in

- obdelovati zahteve osebnega računalnika.

Vse naloge se morajo izvrševati v dejanskem času, saj bi v nasprotnem primeru prišlo do nezveznosti v vzorčenem signalu, poleg tega bi bil sistem neodziven na zahteve. Večino težav je bilo odpravljenih s pomočjo krožnih izravnalnikov različnih velikosti.

5.3. Uporaba zunanjega pomnilnika

Kot smo že omenili, uporabljamo zunanji pomnilnik za shranjevanje vzorcev o poteku krvnega tlaka. V zunanji pomnilnik lahko zapišemo 32768 podatkov velikosti 8 bitov. Zunanji pomnilnik je priključen na vrata B in D.

5.3.1. Branje in pisanje iz zunanjega pomnilnika

Pri branju in zapisovanju smo uporabili vezje z zapahi, zato moramo predno preberemo ali zapišemo podatek ustrezno nastaviti oba zapaha. Branje iz zunanjega spomina poteka v večih fazah. Ker uporablja pomilniški čip 15 bitov dolgo naslovno vodilo, moramo najprej napolniti ustrezna zapaha z željenim naslovom. Vrata D so široka 8 bitov in se uporabljajo za pisanje v zapahe, pisanje in branje iz pomnilniškega čipa. Najprej se napolni prvi zapah, kateri si zapomni spodnjih 8 bitov naslovnega vodila, zatem pa še drugi zapah, kateri predstavlja zgornjih 7 bitov naslovnega vodila. Ko izhodi obeh zapahov predstavljata pravilen naslov, lahko iz spominskga čipa preberemo podatek. To storimo tako, da nastavimo vrata D kot vhod, za tem pa postavimo kontakt (OE) v nizko stanje. Podatek lahko preberemo preko vrat D. Izsek iz kode številka 1 prikazuje funkcijo za branje podatkov iz pomnilniškega čipa.

```
unsigned char ReadMemory(unsigned int Address)
//funkcija prebere podatek iz zunanjega pomnilnika
{
  char res;
  TRISB = 0x00;         //nastavimo vrata B kot izhodna
  TRISD = 0x00;         //nastavimo vrata D kot izhodna
  PORTB |= 0x0C;        //nastavimo OE ter WE na 1
  PORTB &= 0xCC;
  PORTBbits.RB0 = 1;    //postavimo LE zapaha 1 na 1
  PORTD = Address;      //zapisemo spodnji del naslova v zapah 1
  PORTBbits.RB0 = 0;    //postavimo LE zapaha 1 na 0
     //od sedaj naprej drzi zapah 1 spodnji
     //del naslova
     PORTD = 0x00;
     PORTBbits.RB1 = 1;    //postavimo LE zapaha 2 na 1
     PORTD = (Address >> 8); //zapisemo zgornih 8 bitov naslova
     //v zapah 2
     PORTBbits.RB1 = 0;    //postavimo LE zapaha 1 na 0
     //od sedaj naprej drzi zapah 2 spodnji
     //del naslova
     TRISD = 0xff;         //nastavimo vrata D kot vhodna
     PORTBbits.RB3 = 0;    //postavimo OE pomnilniskega cipa na 0
     res  = PORTD;         //spravimo podatek v spremenljivko "res"
     PORTBbits.RB3 = 1;    //postavimo OE pomnilniskega cipa na 1
     return res;           //funkcija vrne prebrani podatek
   }
```

Izsek iz kode 1: Branje podatkov iz zunanjega pomnilnika

Funkcija zapisovanja je precej podbna funkciji za branje, saj obe uporabljata ista vrata in iste zatiče.

5.3.2. Problem sočasnega dostopa do zunanjega pomnilnika

Zunanji pomnilnik uporabljata proces vzorčenja in komunikacijski proces hkrati. V pomnilnik zapisuje proces vzorčenja v enakomernih časovnih razmakih podatke o poteku krvnega tlaka. Podatki se ob neenakomernih časovnih razmakih berejo iz pomnilnika in nato v paketih pošiljajo na osebni računalnik.

Dostop do pomnilnika lahko uporablja le en proces na enkrat, saj je iz zgornje funkcije razvidno, da bi v nasprotnem primeru delovala napačno. Proces vzorčenja se odvija v prekinitvenem izvajanju, komunikacijski proces pa v glavnem izvajanju. Do sočasne uporabe zunanjega pomnilnika bi lahko prišlo v primeru, kadar osebni računalnik zahteva prenos podatkov, torej branje, prekinitveno izvajanje pa bi

hotelo zapisati podatek. Temu smo se izognili na ta način, da prekinitveno izvajanje ne piše direktno v pomnilnik, ampak v izravnalnik.

Zaradi enostavnosti uporabe je v programu zunanji pomnilnik uporabljan kot krožni izravnalnik. V izravnalnik piše proces vzorčenja, iz njega pa podatke pobira proces za komunikacijo z osebnim računalnikom. Ob vsakem zapisu vzorca v pomnilnik se kazalec na naslov za branje poveča za eno, ob branju pa se poveča kazalec na naslov za pisanje. Izsek iz kode številka 2 prikazuje povečevanje kazalca na naslov za branje.

```
res = ReadMemory(adcr++); //preberemo vzorec iz zunanjega pomnilnika,
povecamo
                          //bralni kazalec
adcr &= 0x7fff;           //ter ga zaokrozimo
```

Izsek iz kode 2: Povečevanje kazalca zunanjega pomnilnika

Trenutna velikost podatkov se izračuna kot razlika med kazalcema.

```
unsigned int ADCGetSize(void)
{
  //izracunamo kolicino vzorcev v zunanjem pomnilniku
  if (adcw >= adcr)
    return (adcw - adcr);
  else
    return (adcw + 0x7fff - adcr);
}
```

Izsek iz kode 3: Izračun velikosti podatkov v zunanjem pomnilniku

Ko katerikoli od kazalev doseže maksimalno vredost ga zaokrožimo na 0.

5.4. Vzorčenje ter shranjevanje podatkov o krvnem tlaku

Pretvorba iz analognega signala v digitalni zapis se mora izvajati v enakih časovnih razmakih. Stalnost časovnih razmakov je zagotovljena z uporabo časovnika, kateri proži prekinitve z določeno periodo.

5.4.1. Proženje zahteve po analogno digitalni pretvorbi

Znotraj prekinitve se sproži zahteva po analogno digitalni pretvorbi. Zaradi potrebe po spremljanju nekaterih napetostnih nivojev v analognem delu vezja moramo sekvenčno vzorčiti več kanalov analogno digitalnega pretvornika. To storimo tako, da vsakokrat, kadar obdelujemo časovniško prekinitev, nastavimo drugi kanal za vzorčenje. Funkcijo za proženje analogno digitalne pretvorbe prikazuje izsek iz kode številka 4.

```
if ((ADCState == 0x00) & ((state & 0x04) > 0))   //pogledamo ce imamo
vzorcenje vkljuceno, ter

//preverimo status prejsnje pretvorbe
    {
      ADCON0 &= 0x03;            //zbrisemo nastavitve AD pretvornika
      if(ADCChannel == 0x00)     //nastavimo ustrezen kanal
      {
        ADCont= ADCH0;
      }
      if(ADCChannel == 0x01)
      {
        ADCont= ADCH1;
      }
      ADCState = (0x01 << (ADCChannel));  //nastavimo stanje AD pretvornika
      if (ADCON0bits.GO == 0) //pogledamo ce se je prejsnja pretvorba
koncala
      {
        ADCON0 |= ADCont;            //nastavimo kanal AD pretvornika
        ADCON0bits.GO = 1;           //zazenemo AD pretvorbo
      }
    }
    INTCONbits.TMR0IF = 0;        //zbrisemo zastavico prekinitve
  }
```

Izsek iz kode 4: Proženje analogno digitalne pretvorbe

5.4.2. Obdelava prekinitve končane AD pretvorbe

Po končani analogno digitalni pretvorbi se ponovno sproži prekinitev. Znotraj te prekinitve se rezultat pretvorbe shrani v krožni izravnalnik. Tukaj smo uporabili krožni izravnalik iz dveh razlogov. Prvi razlog je, da shranjevanje v zunanji pomnilnik prestavimo iz prekinitvenega izvajanja v glavno. Drugi razlog pa, da ne vemo točno, kdaj bo izvajanje iz glavnega programa prišlo na odsek kode za shranjevanje vzorca v zunanji pomnilnik. Lahko bi se zgodilo, da bi glavna zanka

obtičala nekje v programu dlje časa, od potrebnega za zajem novega vzorca in bi bil s tem stari vzorec prepisan z novim, star vzorec pa za vedno izgubljen. V krožnem izravnalniku je prostora za 16 vzorcev. S tem posegom smo precej izboljšali robustnost delovanja.

```
if (PIR1bits.ADIF)                  //pogledamo zastavico koncane AD
pretvorbe
{
  ADCResultMsgs[adcmw++] = (ADRESH << 8)|(ADRESLOW); //rezultat shranimo v
                                    //krozni izravnalnik
  adcmw &= 0x0f;                    //zaokrozimo krozni izravnalnik
  ADCChannel++;
  ADCChannel &= 0x03;               //ohranimo spodnja 2 bita
  ADCState = 0x00;                  //nastavimo status na
  PIR1bits.ADIF = 0;                //zbrisemo zastavico AD pretvornika
}
```

Izsek iz kode 5: Shranjevanje rezultata AD pretvorbe v izravnalnik

5.4.3. Shranjevanje vzorcev v glavnem izvajanju programa

Glavna zanka programa bere podatke iz izravnalnika in jih shranjuje v glavni pomnilnik, kjer čakajo, da jih komunikacijski proces prenese na osebni računalnik. S tem smo zagotovili, da zunanjega pomnilnika nikoli ne uporabljata dva procesa hkrati.

```
void ProcessADC(void)   //obdelava podatkov iz AD izravnalnika
{
  INTCONbits.GIE = 0;                    //onemogocimo prekinitve
  while (GetADCMsgLength() > 0)
  {
    if ((ADCResultMsgs[adcmr] & 0x1000) > 0)
    {
      //shranimo vzorec v zunanji pomnilnik
      ADCNewValue((ADCResultMsgs[adcmr] >> 2) & 0x00FF);
    }
    else if ((ADCResultMsgs[adcmr] & 0x2000) > 0)
    {
      ADCResult[0] = ADCResultMsgs[adcmr] & 0x03FF; //osvezimo podatek
    }
    else if ((ADCResultMsgs[adcmr] & 0x4000) > 0)
    {
      //osvezimo podatek v spominu
      ADCResult[1] = ADCResultMsgs[adcmr] & 0x03FF;
    }
    adcmr++;            //povecamo bralni kazalec izravnalnika
    adcmr &= 0x0f;      //zaokrozimo izravnalnik
  }
  INTCONbits.GIE = 1;   //omogocimo prekinitve
 }
```

Izsek iz kode 6: Shranjevanje vzorcev v zunanji pomnilnik

Medtem ko glavna zanka pobira podatke iz krožnega izravnalnika, onemogočimo prekinitve, zaradi tega, ker uporabljamo pri tem kazalca krožnega izravnalnika.

5.5. Obdelava zahtev osebnega računalnika ter prenos podatkov

Zajemanje vzorcev in shranjevanje bi bilo precej neuporabno, če teh vzorcev ne bi kasneje uporabili za analiziranje. Zaradi precej enostavnejše obelave podatkov na osebnem računalniku, smo naredili komunikacijski vmesnik, katerega naloga je prenašanje podatkov na računalnik. Računalniški program, ki analizira podatke, lahko ukaže mikrokrmilniškemu sistemu spremembo nekaterih parametrov, kot sta ojačenje signala ter jakost osvetlitve tkiva. Komunikacija poteka preko asinhronega serijskega vmesnika RS232. Prenos poteka preko dveh žic RX (receive) ter TX (transmit) in ne uporablja strojnega nadzora prenosa. Hitrost prenosa je prednastavljena na 115200 baudov, saj je ta hitrost standardno sprejeta ter dovolj hitra za prenos večje količine podatkov. Vsa komunikacija poteka po načelu zahteva

- odziv. Osebni računalnik vedno pošilja zahteve, mikrokrmilniški sistem pa izvršuje ukaze ter odgovarja na poslane zahteve. Mikrokrmilk pošilja podatke le takrat, ko je to od njega zahtevano. Pošiljanje ter sprejemanje je izvedeno preko krožnih izravnalnikov in prekinitev, zaradi razbremenitve izvajanja glavnega programa. Posledično se zaradi tega dvigne hitrost celotnega sistema, saj ni potrebe po dolgotrajnem čakanju pošiljanja celotnega niza hkrati, kar utegne biti precej zamudno. Zahteva po izravnalnikih izhaja tudi iz dejstva, da mora sistem v vsakem primeru delovati v dejanskem času. Proces vzorčenja se mora izvrševati tudi takrat, kadar prenašamo velike pakete podatkov na osebni računalnik.

5.5.1. Sprejemanje podatkov

Pri sprejemu podatkov uporabljamo način, kjer se sproži prekinitev ob vsakem prispelem podatku. Podatek shranimo v krožni izravnalnik, kjer počaka na kasnejšo obdelavo glavne zanke. Uporaba izravnalnika je tukaj potrebna zaradi tega, ker uporabljena serijska povezava ne uporablja nadzora pretoka. Velikost izravnalnika mora biti dovolj velika, da v primeru hitrega pošiljanja ukazov s strani osebnega računalnika ne povzroči preliva sprejemnega modula**.**

```
if (PIR1bits.RCIF == 1)     //pogledamo, ce smo sprejeli kaksen podatek
{
  rxBuff[rxBuffWritePtr++] = RCREG; //shranimo podatek v vhodni izravnalnik
  rxBuffWritePtr &= 0x07;           //ohranimo le spodnje 4 bite kazalca
  PIR1bits.RCIF = 0;                //zbrisemo zastavico
}
```

Izsek iz kode 7: Sprejemanje podatkov

Velikost vhodno krožnega izravnalnika smo določili na 8 zlogov, saj pošilja osebni računalnik precej manj podatkov mikrokrmilniškemu sistemu, kakor v obratni smeri.

5.5.2. Pošiljanje podatkov

Pošiljanje podatkov je ena izmed bolj zahtevnih opravil, kar se tiče procesorskega časa, potrebnega za prenos večje količine podatkov. Zato smo tudi tu uvedli uporabo izravnalnikov. Podatki se pošiljajo proti osebnenu računalniku s pomočjo

prekinitev. Ob vsakem končanem prenosu se sproži prekinitev, katera pogleda, če ima v izravnalniku še kakšen podatek za pošiljanje.

```
if (PIR1bits.TXIF == 1)
  {
    //pogledamo ce se je prejsnji prenos zakljucil
   if(TXSTAbits.TRMT == 1)
   {
      //izracunamo dolzino za posiljanje
     if (txBuffWritePtr >= txBuffReadPtr)
        txBuffLen = (txBuffWritePtr - txBuffReadPtr);
      else
        txBuffLen = (txBuffWritePtr  + 64 - txBuffReadPtr);
      if(txBuffLen != 0)
     {
       //Omogocimo delovanje oddajnega elementa
         TXSTAbits.TXEN = 1;
       //Nalozimo podatek v oddajni register
         TXREG = txBuff[txBuffReadPtr++];
       //ohranimo spodnjih 6 bitov bralnega kazalca
         //oddajnega izravnalnika
         txBuffReadPtr &= 0x3f;
      }
      else
      //ce nimamo vec podatkov, onemogocimo oddajno prekinitev
           PIE1bits.TXIE = 0;
   }
   PIR1bits.TXIF = 0;  //zbrisemo prekinitveno zastavico}
```

Izsek iz kode 8: Pošiljanje podatkov

Prenos podatkov se prične, ko glavna zanka kliče funkcijo za prenos podatkov. Kadar izhodni izravnalnik ni napolnjen do konca, lahko vanj naloži funkcija podatke, kateri bodo kasneje s pomočjo prekinitvene procedure poslali proti osebnemu računalniku. Če je izhodni izravnalnik napolnjen do konca, pa moramo počakati, da prekinitvena procedura pošlje ven vsaj en podatek in tako sprosti prosto mesto za novi podatek.

```
void putcRS232(char data)
{
  char SendOK = 0;
  while (SendOK != 1)    //ponavljamo dokler ne shranimo podatkov
                        //v izravnalnik
  {
    if(GetTxBuffLength() < 64) //ce izravnalnik ni do konca napoljnjen
                               //lahko vanj nalozimo podatke
    {
      PIE1bits.TXIE = 0;    //onemogocimo posiljanje dokler
                           //dodajamo elemente v izravnalnik
      txBuff[txBuffWritePtr++] = data;
      txBuffWritePtr &= 0x3f;
      SendOK = 1;          //nastavimo zastavico uspesnega dodajanja
      PIE1bits.TXIE = 1;   //omogocimo prekinitve oddajnega modula
    }
   }
```

Izsek iz kode 9: Funkcija za pošiljanje znakov

Prenos se prične, kadar oddajni modul izda zahtevo za prekinitev. V prekinitvi se preveri, če je oddajni register prazen. V primeru, da je oddajni register prazen in v izhodnem izravnalniku čaka podatek, se ta podatek naloži v oddajni register. Po končanem prenosu se znova kliče prekinitvena procedura, katera ponavlja opisani postopek, dokler ne zmanjka podatkov v oddajnem izravnalniku. S tem postopkom smo pridobili na hitrosti, kadar pošiljamo podatke krajše od dolžine izhodnega izravnalnika, kateri je velik 64 zlogov. Pri daljših prenosih se izravnalnik kmalu napolni do konca in je funkcija obsojena na čakanje, da se izravnalnik sprazne. Čas, katerega porabi funkcija za pošiljanje podatkov lahko izračunamo iz enačbe št. 7.

$$t = \frac{n \cdot (8\text{bit} + 2\text{bit})}{\text{BAUD}}$$

Enačba 7: Izračun časa potrebnega za pošiljanje zloga

Kjer *t* pomeni čas potreben za izvršitev, BAUD pomeni hitrost prenosa v bitih na sekundo, *n* pa pomeni koliko zlogov želimo poslati. Vedno pošiljamo poleg vsakega zloga tudi začetni ter končni bit, katera zmanjšujeta celotno hitrost prenosa. Za prenos bloka velikosti 2000 zlogov, bi pri hitrosti 115200 bitov na sekundo funkcija potrebovala približno 175 milisekund. Če upoštevamo, da je pri vzorčni frekvenci 2

kHz analogno digitalnega pretvornika čas med vzorci dolg 0,5 ms, na izravnalnik lahko naložimo 16 vzorcev, moramo obdelati podatke vsaj vsakih 8 ms, da ne pride do preliva. Iz zgoraj navedenega ugotovimo, da ne smemo nikoli pošiljati večjih paketov na enkrat. V aplikaciji smo to rešili s pomočjo statusa ter števca poslanih podatkov.

```
transmitCounter = ADCGetSize();        //preberemo stevilo vzorcev
if (ADCGetSize() > 0) state |= 0x02;   //postavimo zastavico posiljanja
podatkov
putcRS232(transmitCounter & 0xff);     //posljemo spodnjih 8 bitov stevila
podatkov
putcRS232((transmitCounter >> 8) & 0xff); //posljemo zgornjih 8 bitov
stevila podatkov
```

Izsek iz kode 10: Proces pošiljanja podatkov na osebni računalnik

Kadar pride zahteva za prenos podatkov, preberemo najprej število podatkov v izravnalniku za vzorce. Če imamo kakšen podatek, nastavimo zastavico, s katero tudi v naslednjem ciklu vemo, da imamo še nekaj podatkov za poslati. Potem pošljemo dva znaka katera predstavljata koliko vzorcev bo poslano v tem paketu.
V vsakem ciklu preverjamo zastavico za pošiljanje in če je nastavljena pošljemo ven en znak.

```
putcRS232(ADCGetValue());    //vsak cikel spravimo en podadtek v izhodni
izravnalnik
transmitCounter--;           //zmanjsamo stevec prenosa
if (transmitCounter == 0) state &= 0xfd;  //ce smo zakljucili prenos
zbrisemo zastavico
```

Izsek iz kode 11: Ciklično pošiljanje vzorcev

Velikost paketa je odvisna od tega, kakšno frekvenco vzorčenja uporabljamo ter od časa med posameznimi zahtevami za prenos. S tem smo dosegli to, da je dolžina prilagodljiva potrebam. Če osebni računalnik potrebuje sveže podatke, izda zahtevo za prenos bolj pogosto kot takrat, kadar že obdeluje podatke in nima časa za obdelavo novih.

5.5.3. Glavna zanka

Vsak program potrebuje funkcijo, katera povezuje skupaj vse module. V programskem jeziku C je to v večini primerov funkcija main(). Naloga te funkcije v našem programu je inicializacija uporabljenih programskih in mikrokrmilniških modulov. Moduli, kateri potrebujejo inicializacijo so:

- dostop do zunanjega pomnilnika,
- analogno digitalni pretvornik,
- serijski komunikacijski vmesnik,
- časovnik namenjen enakomernemu vzorčenju in
- prekinitve.

Po uspešni inicalizaciji modulov lahko pričnemo s cikličnim klicanjem funkcij, katere poskrbijo za:

- prejemanje in izvrševanje ukazov,
- shranjevanje vzorcev v glavni izravnalnik in
- pošiljanje vzorcev.

5.5.4. Sprejemanje in dekodiranje ukazov

Mikrokrmilniški sistem čaka na zahtevo osebnega računalnika. Kadar dobi ukaz, ga mora najprej dekodirati in se odzvati na podano zahtevo. Zahteva je lahko, kot smo že omenili prenos podatkov, nastavitev ojačenja ali kaj drugega. Ukaz je sestavljen iz črke ali niza, konec ukaza predstavlja ASCII znak #13. Ta znak predstavlja »Carriage return« ali postavitev kurzorja na začetek vrstice. Program shranjuje sprejete podatke v ukazno vrsto, kadar pa prejme znak #13 ga dekodira in izvrši ukaz.

```
if (GetRxBuffLength() > 0)  //če imamo kakšen znak v vhodnem izravnalniku
  {
    inChar = rxBuff[rxBuffReadPtr++];   //ga preberemo ter povečamo kazalec
                                        //izhodnega izravnalnika
    rxBuffReadPtr &= 0x07;      //zaokrožimo kazalec izhodnega izravnalnika
    if(state & 0x08)           //preverimo zastavico statusa sprejema ukaza
    {
      if(inChar != 0xd)        //če vhodni znak ni znak za konec ukaza (#13)
      {
        inBuff[inBuffPos++] = inChar;   //spravimo znak v ukazno vrsto
      }
      else
      {
        ProcessLine();        //klicemo funcijo za dekodiranje ukaza
       state &= 0xf7;        //zbrišemo zastavico statusa sprejema ukaza
      }
    }
    else                        //kadar pričnemo prejemati nov ukaz
    {
      inBuffPos = 0;            //postavimo kazalec vhodne vrste na začetek
      inBuff[inBuffPos++] = inChar;   //spravimo znak v vrsto
      state |= 0x08;            //postavimo zastavico statusa sprejema ukaza
    }
  }
```

Izsek iz kode 12: Sprejemanje ukazov

Izsek iz kode št 12 prikazuje ciklično dekodiranje podatkov in izvrševanje ukazov. Zaradi cikličnega izvajanja je potrebno voditi status sprejema ukaza, da vemo kdaj se pričenja nov ukaz. Spodnji izsek iz kode prikazuje dekodiranje ukaza za nastavitev digitalno analognih pretvornikov, kateri so na mikrokrmilnik priključeni preko serijskega vodila SPI. Čeprav sta digitalno analogna pretvornika in digitalno nastavljiv potenciometer priključena preko vodila SPI, nastavljamo naprave različno zaradi različnih komunikacijskih načinov in protokolov. Dekodiranje ukaza za nastavitev ojačenja in toka skozi oddajno diodo prikazuje izsek iz kode št. 13.

```
if (inBuff[0] == 'w') //ukaz za nastavitev analogno digitalnih
                      //pretvornikov ter ojačenja
{
  data = inBuff[2];
  if (inBuff[1] < 0x3)  SetAuxPort(inBuff[1], data << 3);
                    //na vratih 1 in 2
                    //se nahajata digitalno analogna
                    //pretvornika
  else if(inBuff[1] == 0x3) SetRes(3, data);
                    //na vratih 3 se nahaja digitalno
                    //nastavljiv potenciometer za
                    //za nastavitev ojačenja
}
```

Izsek iz kode 13: Dekodiranje ukazov za nastavljanje ojačenja, odmika ojačenja ter toka oddajne diode

V povezavi z nastavljanjem ojačenja in odmika vhodnega ojačevalnika ter toka skozi oddajno diodo, uporabljamo funkcijo za branje napetosti s tokovno napetostnega pretvornika. S pomočjo tega podatka, lahko optimalno nastavimo tok skozi oddajno diodo. Ukaz ter kodo za dekodiranje in izvrševanje prikazuje izsek iz kode št. 14.

```
if (inBuff[0] == '1')   //ukaz za branje vrednosti analogne napetosti
                        //številka pomeni kanal AD pretvornika
 {
   putcRS232(ADCResult[0] & 0x00ff);  //najprej pošljemo spodnjih 8 bitov
   putcRS232((ADCResult[0] >> 8) & 0x00ff);  //za tem še zgornjih 8 bitov
 }
```

Izsek iz kode 14: Dekodiranje ukazov za nastavljanje ojačenja, odmika ojačenja ter toka oddajne diode

Poleg opisanih ukazov, vsebuje program tudi ukaze za resetiranje mikrokrmilnika, testiranje zunanjega pomnilnika ter ukaze za nadzor vzorčenja.

5.6. Knjižnica za komunikacijo z mikrokrmilniškim sistemom

5.6.1. O knjižnici

Z namenom, da bi poenostavili nadaljni razvoj aplikacije za merjenje frekvence utripa srca na osebnem računalniku, smo naredili knjižnico. Knjižnica predstavlja vmesni sloj med merilnikom in programom za obdelavo podatkov. Kadar program za obdelavo podatkov želi nove podatke, kliče funkcijo v knjižnici, katera mu vrne polje podatkov. Knjižnica je napisana za operacijski sistem Windows XP. Knjižnica je dinamična, kar pomeni, da kadar program potrebuje vmesnik do merilnika, pokliče sistemsko funkcijo za ustvaritev knjižnice, ta pa mu vrne objekt preko katerega lahko kliče funkcije iz knjižnice. Knjižnica se lahko uporabi v kateremkoli programskem jeziku ali razvojnem okolju, kateri podpira dinamične knjižnice. Knjižnico smo naredili s pomočjo razvojnega okolja Delphi 6 in komponente za delo s serijskim vmesnikom Async Pro.

5.6.2. Zgradba knjižnice

Pri izdelavi programov, kateri uporabljajo serijski vmesnik moramo biti pozorni, da delujejo asinhrono. To pomeni, da podatkov ne čakamo v neskončni zanki, temveč izkoriščamo sistem sprožil- kateri je vgrajen v operacijski sistem. Operacijski sistem pošlje sporočilo, kadar prispe kakšen podatek preko serijskega vmesnika vsem programom, kateri uporabljajo ta vmesnik. Tako obdelujemo podatke v vhodnem izravnalniku samo, kadar prispe kakšen nov podatek. Tako razbremenimo centralno procesorsko enoto, in s tem dosežemo, da ostali programi, kateri tečejo sočasno delujejo normalno. To je za naš projekt pomembno, saj potrebuje program za obdelavo podakov precej procesorskega časa. Da bi razvoj knjižnice poenostavili, smo uporabili odprtokodno komponento za delo s serijskim vmesnikom Async Pro. Knjižnica vsebuje vse funkcije, katere so podprte s strani mikrokrmilniškega sistema. Poleg teh funkcij vsebuje še funkciji za odpiranje in zapiranje serijskega vmesnika.

5.6.3. Periodično prenašanje vzorcev iz mikrokrmilniškega sistema

Velikost pomnilnika, katerega proces vzorčenja uporablja za začasno odlaganje vzorcev, je precej omejena. Zato mora knjižnica poskrbeti, da pogosto prenaša podatke in s tem prepreči zapolnitev zunanjega pomnilnika mikrokrmilnika preko roba. Zato smo uporabili časovnik, kateri pošlje zahtevo mikrokrmilniku za prenos podatkov vsakih 100 ms. V tem času se pri vzorčni frekvenci 2 Khz nabere v zunanjem pomnilniku mikrokrmilnika približno 200 vzorcev, kar predstavlja tudi velikost paketa. Če bi ta čas podvojili, bi se s tem podvojila velikost paketa. Izsek iz kode št.15 prikazuje periodično pošiljanje zahteve za prenos podakov. Funkcija OnStateTimer se izvede vsakih 100 ms.

```
procedure TSerialAq.OnStateTimer(ASender: TObject);
begin
  if FAqStatus = aqIdle then begin    //ce je trenutni status nezaposleno
    FAdch := 0;                       //nastavimo status na prvi kanal AD pretvornika
    FAqStatus := aqReceive;           //postavimo status na prejemanje
    FRecStat := rcFirstByte;          //postavimo status sprejema na prvi zlog
    FApdComPort.PutChar('T');         //posljemo ukaz za prenos podatkov
    FApdComPort.PutChar(#13);         //posljemo zakljucni znak
  end;
end;
```

Izsek iz kode 15: Periodično pošiljanje ukaza za sprejem vzorcev

Zahteva se pošlje mikrokrmilniškemu sistemu. Mikrokrmilnik ukaz dekodira in pošlje najprej podatek o velikosti paketa, katerega bo poslal. Ta podatek si moramo zapomniti, da bomo po prenosu celotnega paketa znali postaviti status na nezaposleno.

V knjižnici se moramo zavedati statusa, s katerim določimo trenutno stanje prenosa, da ne poskusimo pošiljati novih ukazov, predno stari ne izvršijo do konca. Poleg tega se moramo zavedati tudi, da se vzorci pošiljajo v paketih, zato moramo

imeti status, s katerim določimo trenutno stanje napredka prenosa. Implementacijo teh dveh statusov prikazuje izsek iz kode št. 16

```
TAqStatus = (aqUnInitialized,        //vmesnik ni inicializiran
             aqClosed,               //vmesnik je zaprt
             aqWaitingToBeOpened, //vmesniku smo poslali zahtevo za
                                     //odpiranje in cakamo potrditev
             aqIdle,                 //vmesnik je nezaseden
             aqReceive);             //trenutno sprejemamo podatke preko
                                     //vmesnika

TReceiveStat = (rcNone,           //trenutno ne sprejemamo nicesar
                rcFirstByte,      //trenutno prejemamo prvi zlog dolzine paketa
                rcSecondByte,     //trenutno prejemamo drugi zlog dolzine paketa
                rcBuffer);        //trenutno prejemamo vzorce
```

Izsek iz kode 16: Definicija statusov

Kadar serijski vmesnik prejme podatek, se nam proži dogodek, kateri kliče ustrezno funkcijo, katera obdela prejete podatke. Izsek iz kode št. 17 nam prikazuje mehanizem prejemanja vzorcev. Paket se prenese tako, da se najprej pošlje spodnjih 8 bitov dolžine paketa, potem zgornjih 8 bitov dolžine paketa. Mikrokrmilnik pošlje nato tolikšno število vzorcev, kolikor narekuje prejeta dolžina.

```
procedure TSerialAq.ApdComPortTriggerAvail(CP: TObject; Count: Word);
var
  i: Integer;
  b: byte;
begin
  for i:=0 to Count - 1 do begin          //obdelamo vse podatke
    if (FAqStatus = aqReceive) then       //ce trenutno prejemamo podatke
      b := byte(FApdComport.getChar);     //preberemo znak iz serijskega vmesnika
    if FRecStat = rcFirstByte then begin  //ce trenutno prejemamo prvi zlog dolzine
      FTMPValue := b;
      FRecStat := rcSecondByte;           //naslednji znak bo drugi zlog dolzine
    end
    else if FRecStat = rcSecondByte then begin  //ce prejemamo drugi znak dolzine
      if FADChannel = adCH0 then begin
        FBufflength := FTMPValue + b*256;       //izracunamo dolzino paketa
        if FBuffLength > 0 then                 //ce je dolzina paketa vecja od 0
        FRecStat := rcBuffer                    //naslednji prejeti znaki bodo vzorci
      end
    end
    else if FRecStat = rcBuffer then begin   //ce trenutno prejemamo paket
      FBuffLength := FBuffLength-1;          //zmanjsamo stevilo cakajocih znakov paketa
      FBuffer.Add(TObject(b));               //dodamo vzorec v izravnalnik knjiznice
    end;
  end;
end;
```

Izsek iz kode 17: Asinhrono prejemanje podatkov

6. Sklepne ugotovitve

Zadane cilje smo dosegli brez večjih kompromisov. Izbrana metoda merjenja se je izkazala kot ustrezna in zadovoljiva za namen uporabe. Naredili smo sistem, kateri avtomatizira zajemanje podatkov. S pomočjo spremenljivega toka skozi oddajno diodo, ter nastavljivih parametrov ojačenja smo naredili merilnik prilagodljiv na spremembe parametrov merjene osebe. Metoda se je izkazala za neinvazivno merjeni osebi, zaradi prisotnosti senzorja ne povzročamo psihofizičnih sprememb, saj je sezor prijeten za nošenje. Infrardeč spekter se je izkazal kot primeren za uporabo iz večih pogledov. Merilnik je praktično neobčutljiv na spremembo zunanje svetlobe, težave mu povzročajo le premiki prsta. Premik prsta povzroči spremembo tlaka v kapilarah, zaradi tega, ker je senzor občutljiv tudi na spremebe tlaka. Na to ne moremo vplivati, niti ne moremo v tem pogledu merilnika izboljšati. Mikrokrmilniški sistem se je izkazal kot ustrezen, saj zmore izvajati zadane naloge v dejanskem času.

Možnosti za izboljšave so se pričele kopičiti tekom izdelave senzorja. Najbolj zaželena izboljšava bi lahko bila optimalno prilagajanje parametrov merilnika glede na merjeno osebo. To bi lahko naredili s pomočjo regulacijske zanke, kjer bi regulirali tok skozi oddajno diodo. Seveda se pa regulator ne bi smel hipoma odzvati na motnjo, saj nam le ta predstavlja pomemben podatek. Precej bi lahko naredili še na končni ceni izdelka, kar pa že presega okvire tega dela.

Viri

1. P. Šuhel, B.Murovec, Računalniška integracija proizvodnje, Fakulteta za elektrotehniko, Ljubljana; Gorenje, Velenje, 2003
2. P. Šuhel, Industrijska elektronika: Operacijski ojačevalnik v sistemih, Mengeš, D design, 1995
3. B. Murovec, Laboratorijske vaje pri industrijski elektroniki, Založba FE in FRI, Ljubljana, 2001
4. Robert E. Massara, J.W. Steadman, B.M. Wilamowski, James A. Svoboda, Active filters, The Electrical Engineering Handbook, CRC Press LCC, 2000
5. J. Furlan, Osnove nelinearnih elemento, Založba FER, Ljubljana, 1994
6. R. Kladnik, Osnove fizike, Državna založba Slovenija, 1988
7. S. Winder, Analog and Digital Filter Design, Second Edition, Elsevier Science, Voburn, 1997
8. S.W.Smith, The Scientist and Engineer's Guide to Digital Signal Processing, Second edition, California Technical Publishing, San Diego, 1999
9. M. Gams, P. Jakopin, I. Kanič, D.Kodek, B. Mohar, B. Vilfan, Računalniški slovarček, Cankarjeva založba, Ljubljana 1985
10. PIC18F4520 Data Sheet, http://ww1.microchip.com/downloads/en/DeviceDoc/39631a.pdf, 25. oktober 2005
11. MCP42050 Digital Potentiometer with SPI Interface Data Sheet, http://ww1.microchip.com/downloads/en/DeviceDoc/11195c.pdf, 25. oktober 2005

12. SFH2030 Silicon PIN Photodiode with daylight filter, http://www.ortodoxism.ro/datasheets/siemens/Q62702-P955.pdf, 25. oktober 2005
13. Infrared Emitter LD 271, http://wwww.ges.cz/sheet/l/ld_271, 25. oktober 2005
14. 12V, 7 Mhz, CMOS, Rail to Rail I/O Operational Amplifier, http://www-s.ti.com/sc/ds/opa2743.pdf, 25.oktober 2005
15. 8th-Order, Low Pass, Switched Capacitor Filters, http://pdfserv.maxim-ic.com/en/ds/MAX291-MAX296.pdf, 25.okbober 2005
16. 32,768 word x 8 bit High Speed CMOS Static RAM, http://web.mit.edu/6.s28/www/datasheets/62256.pdf, 25.oktober 2005
17. Mark Balch, Complete Digital Degisn, McGrawHill, New York, 2003
18. PPG Seminar, http://www.eeng.may.ie/~tward/documents/PPG_Seminar.ppt, 18.november 2005
19. http://www.microchip.com, 20. november 2005
20. 74HC573 Octal D-Type Transparent Latch http://www.alldatasheet.com/datasheet-pdf/pdf/15640/PHILIPS/74HC573.html, 20. november 2005
21. MAX 232 Dual EIA 232 Driver / Receiver, http://pdf.alldatasheet.com/datasheet-pdf/view/27224/TI/MAX232.html, 20 november 2005
22. TLC 5615 10 Bit Digital To Analog Converters, http://www.alldatasheet.com/datasheet-pdf/pdf/28916/TI/TLC5615.html, 20. november 2005
23. 3-to-8 line decoder/demultiplexer, http://www.alldatasheet.com/datasheet-pdf/pdf/15535/PHILIPS/74HC138.html, 20.november 2005
24. Feedback plots define op amp AC performance http://focus.ti.com/lit/an/sboa015/sboa015.pdf, 30. november 2005

7. Izjava

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Boštjana Murovca, univ. dipl. inž. el. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Matej Anžin