

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET



Diplomski rad

**Sistem za detektovanje pokreta pomoću
Infracrveno emitujućih dioda u L konfiguraciji**

Mentor:

dr Vladimir Rajović, van. prof.

Kandidat:

Uroš Cvjetinović, 2016/0093

Beograd, Septembar 2021.

SADRŽAJ

SADRŽAJ.....	1
1. UVOD.....	2
2. DETEKCIJA POKRETA.....	4
2.1. INDUKTIVNI DETEKTOR POKRETA	4
2.2. KAPACITIVNI DETEKTOR POKRETA	5
2.3. ULTRAZVUČNI DETEKTOR POKRETA	6
2.4. PIROELEKTRIČNI DETEKTOR POKRETA	7
2.5. DETEKTOR POKRETA POMOĆU INFRACRvene SVETLOSTI	8
3. ARHITEKTURA SOFTVERA MIKROKONTROLERA.....	9
3.1. APLIKATIVNI PROGRAM	9
3.2. API.....	10
3.3. RTOS	10
3.4. MIDDLEWARE	10
3.5. HAL	10
4. ORGANIZACIJA REALIZOVANOG SISTEMA	11
4.1. HARDVER U SISTEMU	11
4.1.1. MINI-M4.....	12
4.1.2. Proximity 13 click	13
4.1.3. USB UART click	15
4.2. SOFTVER SISTEMA	16
4.2.1. CMSIS.....	17
4.2.2. TivaC drajveri	18
4.2.3. Drajveri za senzor.....	18
5. REALIZACIJA APLIKACIJE	19
5.1. ORGANIZACIJA NITI	20
5.1.1. Nit za komunikaciju sa korisnikom.....	22
5.1.2. Nit za čitanje podataka iz senzora	23
5.1.3. Nit za računanje pozicije i detektovanje pokreta.....	24
5.2. DETEKTOVANJE POKRETA.....	25
6. REZULTATI TESTIRANJA APLIKACIJE	26
6.1. PRVI EKSPERIMENT	27
6.2. DRUGI EKSPERIMENT	28
6.3. TREĆI EKSPERIMENT	29
6.4. ČETVRTI EKSPERIMENT.....	30
7. ZAKLJUČAK	31
8. LITERATURA.....	32
SPISAK SKRAĆENICA.....	33
SPISAK SLIKA	34

1. UVOD

Detekcija pokreta ima važnu ulogu u sigurnosnim sistemima, industriji i komercijalnim proizvodima. Sistemi koji detektuju pokret nazivaju se detektori pokreta. Detektori pokreta se mogu nadograditi na postojeće sisteme i proširiti njihove funkcionalnosti.

Postoje različiti načini da se postigne detekcija pokreta, pomoću softverske ili hardverske detekcije. Primer softverske detekcije je obrada slike korišćenjem snimka sa kamere, dok primeri za detekciju pomoću hardvera uglavnom sistemi sa senzorima direktno povezanim na posmatrani predmet ili ne.

Detektor pokreta realizovan pomoću infracrvene svetlosti nije direktno povezan na predmet čije se pokret detektuje. Ovakav tip detekcije zahteva neopstruisan prostor između detektora i predmeta, sa izuzetkom stakla koje propušta infracrvenu svetlost.

Analizom sistema za detekciju pokreta mogu se predstaviti svi nivoi namenskog računarskog sistema. Predstavljanjem realizacije sistema opisuju se rad hardvera koji meri fizičku veličinu, komunikacija između hardvera i svi slojevi softvera na jednom elektronskom uređaju.

Hardverski deo sistema za detekciju pokreta je uspostavljen pomoću dostupnog i pristupačnog hardvera, sa ciljem da se može lako replikovati. Hardver se sastoji od mikrokontrolera i periferije sa senzorom infracrvene svetlosti.

Softverski deo sistema za detekciju pokreta je implementiran tako da sistem bude sistem u realnom vremenu, pa se realizovani sistem zapravo može nazvati sistem detekcije pokreta u realnom vremenu. Implementacija je ostvarena korišćenjem operativnog sistema realnog vremena u softveru sistema. Pojedine funkcionalnosti koje se mogu odvojiti jedna od druge raspoređene su u odgovarajuće programske niti.

Podela delova aplikativnog programa u programske niti je implementirana kako bi se izbegla kolizija protoka informacija. Uvođenjem programskih niti razdvojeni su delovi programa koji obavljaju komunikaciju sa korisnikom i detektovanje pokreta. Razdvajanjem delova programa i korišćenjem operativnog sistema smanjuje se procesorska opterećenost u slučaju da se ne odvijaju nikakva komunikacija niti merenje.

Rad je organizovan u osam glava, sadrži spisak korišćenih skraćenica, naziva slika i spisak korišćene literature.

U drugoj glavi definišu se osnovni pojmovi iz oblasti detekcije pokreta i terminologija koja se koristi u narednim glavama. Opisuju se svi delovi sistema od hardverskog do softverskog. Objašnjavaju se različiti tipovi detektora pokreta i njihove primene.

U trećoj glavi definišu se osnovni pojmovi iz oblasti softvera mikrokontrolera kroz primer jedne uobičajne strukture softvera namenskog sistema. Navode se karakteristike programskog jezika i svih nivoa apstrakcije za pouzdan, modularan i lako razumljiv softver.

U četvrtoj glavi se navode konkretne hardverske i softverske komponente koje su korišćene realizacije sistema za detekciju pokreta. Daju se glavne karakteristike hardverskih komponenti i korišćeni slojevi apstrakcije softvera.

Peta glava govori o načinu realizacije aplikativnog programa korišćenjem svih nivoa apstrakcije.

U šestoj glavi su prezentovani međurezultati aplikacije četiri eksperimenta sa ostavljenim prostorom za tumačenje.

U sedmoj glavi iznet je zaključak o razvoju detektora pokreta i postignutim rezultatima sistema kao i mogućim primenama u industriji i bezbednosnim sistemima.

U osmoj glavi dat je pregled literature.

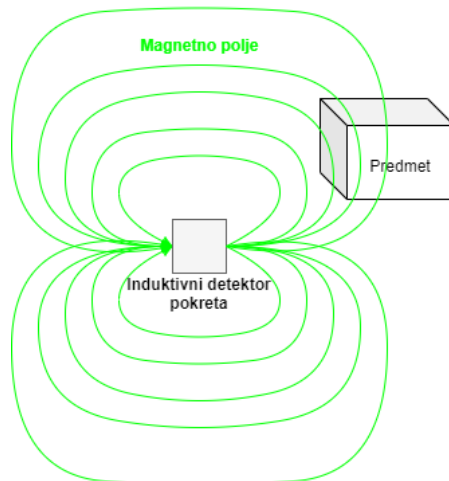
2. DETEKCIJA POKRETA

Senzori za detektovanje pokreta su detektori koji bezkontaktno ili kontaktno detektuju prisustvo i pomeraj objekta u blizini senzora i konvertuju tu informaciju u električni signal koji pamte kao digitalnu informaciju. Ovakvi senzori su korisni u industriji za potrebe praćenja proizvoda, bezbednosti i upravljanja skladištenjem. Osobine detektora pokreta su dug vek trajanja, mala potrošnja i velika brzina rada.

Postoje različiti tipovi detektora pokreta, neki od njih su induktivni, kapacitivni, ultrazvučni, radarski, piroelektrični i optički detektori pokreta. Tipovi detektora se značajno razlikuju po svojim karakteristikama zavisno od sistema u kojim se nalaze.

2.1. Induktivni detektor pokreta

Induktivni detektor pokreta detektuje pokret pomoću promene magnetnog polja koje on generiše. Promena magnetnog polja dovodi do pojave elektromagnetne indukcije koja generiše elektromotornu silu unutar detektora [8]. Pojavom elektromotorne sile dolazi do promene struje u električnom kolu sistema za detekciju. Merenjem indukovane struje sistem ima informaciju o kretanju predmeta unutar svog magnetnog polja. Ilustracija principa rada induktivnog detektora pokreta se može videti na slici 2.1.1.



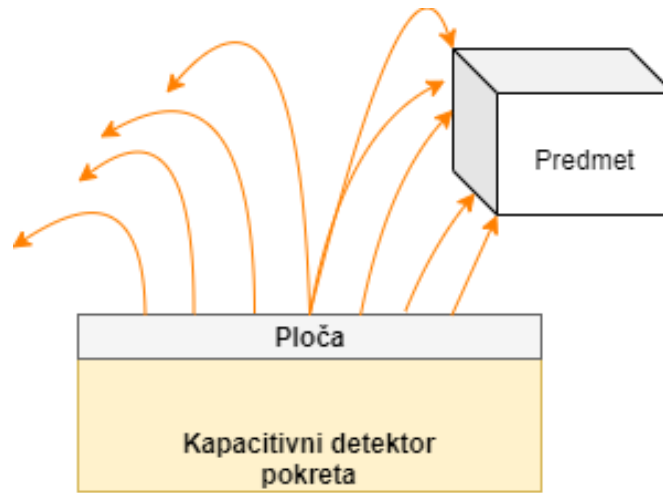
Slika 2.1.1 Princip rada induktivnog detektora pokreta

Sistemi sa induktivnim detektorom pokreta najčešće se primenjuju za metalne predmete. Induktivni detektor pokreta je najbolji detektor pri detektovanju metalnih objekata.

Različiti modeli induktivnog detektora pokreta imaju različite karakteristike. Najčešće detektori ovog tipa mogu detektovati pokret objekata na udaljenostima ne većim od 60 milimetara [10].

2.2. Kapacitivni detektor pokreta

Kapacitivni detektor pokreta detektuje pokret pomoću detektovanja promene kapacitivnosti usled prolaska objekta kroz električno polje koje detektor generiše. Detektor je realizovan pomoću provodne ploče koja pravi električno polje. Usled prolaska objekta kroz električno polje, objekat i ploča realizuju kondenzator sa određenom kapacitivnošću. Merenjem ove kapacitivnosti se dobija informacija o prolasku objekta. Ilustracija principa rada kapacitivnog detektora pokreta se može videti na slici 2.2.1.



Slika 2.2.1 Princip rada kapacitivnog detektora pokreta

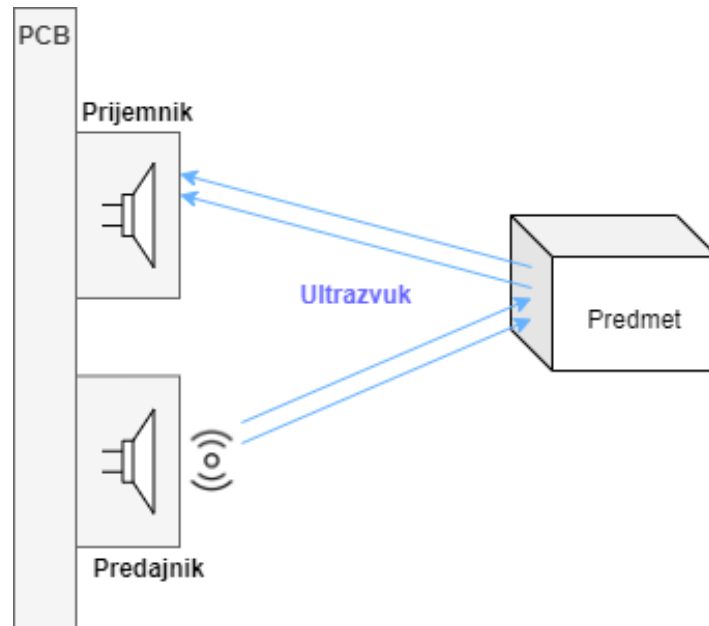
Sistemi sa kapacitivnim detektorom pokreta najčešće se primenjuju za detektovanje objekata, najčešće metalnih, kroz nekakav neprozirni materijal.

Kapacitivni detektor pokreta može da detektuje objekte različitih materijala, što u određenim sistemima je problem. Za detektovanje objekata od gvožđa bolju su induktivni detektori.

Različiti modeli kapacitivnog detektora pokreta imaju različite karakteristike. Najčešće detektori ovog tipa mogu detektovati pokret objekta na udaljenostima ne većim od 60 milimetara [10].

2.3. Ultrazvučni detektor pokreta

Ultrazvučni detektor pokreta detektuju pokret merenjem reflektovanog zvuka, gde reflektovani zvuk potiče od refleksije zvuka koji detektor generiše. Ultrazvuk je zvučni signal sa frekvencijama većim od 32kHz . Detektor je realizovan pomoću predajnog i prijemnog audio elektronike sposobne za generisanje i merenje ultrazvučnog zvuka. Ilustracija principa rada kapacitivnog detektora pokreta se može videti na slici 2.3.1.



Slika 2.3.1 Princip rada ultrazvučnog detektora pokreta

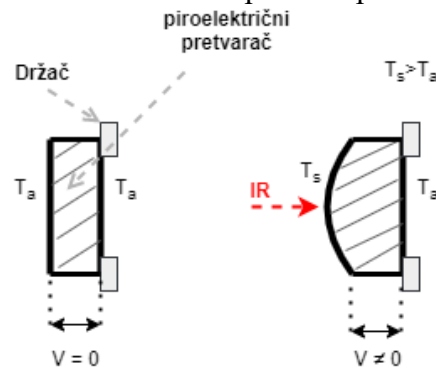
Sistemi sa ultrazvučnim detektorom pokreta najčešće se primenjuju za detektovanje čvrstih predmeta, zbog veće refleksije zvuka. Senzori ovog tipa detektuju promenu unutar određene zapremine oko sebe, što dovodi do toga da imaju bolje karakteristike unutar zatvorenih prostorija.

Ultrazvučnom detektoru pokreta nije neohodan neopstruisan put do objekta koji detektuje, zbog čega je ovaj detektor pogodan za korišćenje u skladištima sa niskim plafonima.

Različiti modeli ultrazvučnog detektora pokreta imaju različite karakteristike. Najčešće detektori ovog tipa mogu detektovati pokret osobe na udaljenostima ne većim od 7 metara [9]. Detektovanje pokreta ruke na udaljenostima manjim od 5 metara.

2.4. Piroelektrični detektor pokreta

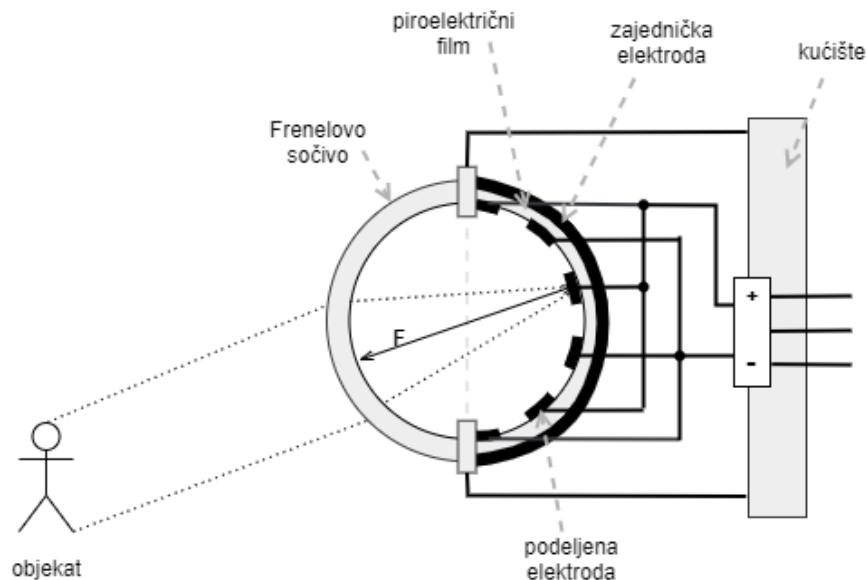
Piroelektrični detektor pokreta detektuje pokret merenjem napona na elektrodama između kojih se nalazi piroelektrični sloj. Toplotno zračenje izaziva širenje ozračene elektrode, što izaziva naprezanje i rezultuje pojavom viška naelektrisanje na elektrodama (piezoelektrični efekat). Princip rada piroelektričnog senzora se može videti na slici 2.4.1. Piroelektrični detektori pokreta su realizovani pomoću piroelektričnih senzora.



Slika 2.4.1 Princip rada piroelektričnog detektora pokreta

Piroelektrični detektori pokreta se koriste za detekciju objekata koji emituju infracrvenu svetlost. Moguće je realizovati aktivni detektor koji emituje infracrvenu svetlost i detektuje reflektovanu svetlost.

Česta realizacija PIR detektora je ilustrovana na slici 2.4.2. Ona se sastoji od frenelovog sočiva, jedne zajedničke elektrode i podeljene elektrode čiji se segmenti nalaze u žižama Frenelovog sočiva. Segmenti podeljene elektrode se dovode naizmenično na diferencijalni pojačavač. Kretanje se detektuje pomoću promene napona na zajedničkoj i podeljenoj elektrodi. Udaljenost detektovanja ne prelazi 8 metara [11].

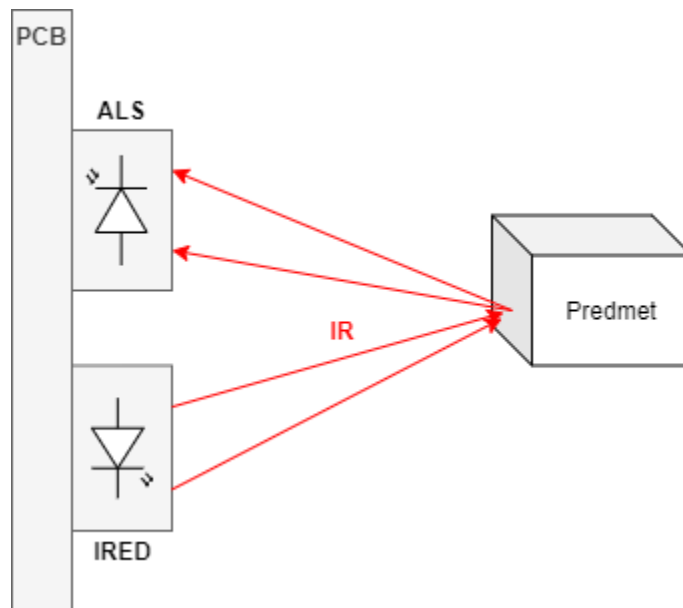


Slika 2.4.2 Ilustracija česte realizacije piroelektričnog detektora pokreta[11]

2.5. Detektor pokreta pomoću infracrvene svetlosti

Detektor pokreta realizovan pomoću infracrvenih dioda (*Infrared Emitting Diodes* – IRED), nadalje IR (*Infrared*) detektor, ima bolje karakteristike od drugih u zatvorenom prostoru. Na IR detektor ne utiče materijal i čvrstoća predmeta, dok kod ostalih tipova detektora može uticati na detekciju.

IR detektor funkcioniše na principu detektovanja poslatog signala, gde je poslati signal zapravo infracrveni zrak. IR detektor poseduje ugrađene diode koje emituju infracrvenu svetlost i detektor svetlosti koji detektuje reflektovanu svetlost. Detektor svetlosti je realizovan tako da se može odrediti ugao upada svetlosti i pozicije izvora svetlosti u odnosu na detektor, što bi u slučaju IR detektora bila reflektovana svetlost od IRED. Detektor svetlosti je najčešće realizovan pomću ALS (*Ambient light sensor*) fotodioda. Ilustracija principa rada IR detektora pokreta se može videti na slici 2.5.1.

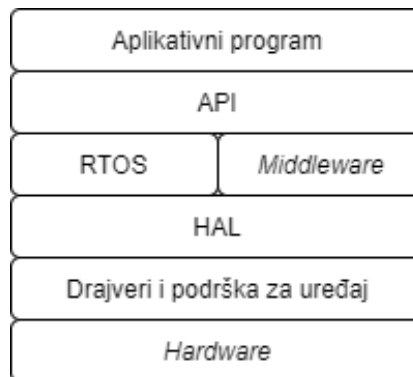


Slika 2.5.1. Princip rada IR detektora pokreta

Različiti modeli infracrvenog detektora pokreta imaju različite karakteristike. Najčešće detektori ovog tipa mogu detektovati pokret osobe na udaljenostima ne većim od 12 metara [9], gde je put između detektora i objekta koji se kreće neopstruisan. Detektovanje pokreta ruke na udaljenostima manjim od 4.5 metara.

3. ARHITEKTURA SOFTVERA MIKROKONTROLERA

Dugi niz godina programiranja uređaja doveo je do pojave da se često programirane stvari standardizuju i podigne nivo apstrakcije pri programiranju. Programiranje jednog uređaja na najnižem nivou apstrakcije je direktno programiranje registara i niza instrukcija u aplikativnom programu. Programiranje uređaja na visokom nivou apstrakcije se postiže pomoću interfejsa i slojeva za apstrakciju. Primer jedne ovakve arhitekture se može videti na slici 3.1.



Slika 3.1 Arhitektura softvera jednog mikrokontrolera

3.1. Aplikativni program

Komplikovaniji uređaji se mogu programirati ili konfigurisati po potrebi programera. Kontroleri se programiraju sa aplikativnim programom koji najčešće konfiguriše i obrađuje podatke periferije u sistemu. Aplikativni program programer može da piše pomoću željenog programskog jezika koji se prevodi u kod nižeg nivoa da bi kontroler mogao da protumači kod. Prevođenje koda se naziva kompajliranje i za to je zadužen kompajler.

Najčešće korišćeni programski jezici za programiranje uređaja za embedded sisteme su C i Asemblerski jezik. Ovi programski jezici su česti jer uređaji u embedded sistemima imaju mali kapacitet memorije i kompajliran kod treba da zauzme mali prostor. C jezik je višeg nivoa apstrakcije od assemblera i često se koriste zajedno u programu. Koriste se zajedno jer assemblerski kod može da bude značajno efikasniji i brži nego neka C biblioteka. Za assemblerski kod potrebno je više vremena za programiranje i tumačenje zbog nižeg nivoa apstrakcije. Sa povećanjem memorijskog prostora u novijim uređajima postaje zastupljeno i korišćenje objektno orijentisano programiranje.

Kompajler je softver na računarima koji prevodi programski jezik višeg nivoa apstrakcije u objektni kod kojim uređaj može da protumači niz instrukcija. Uređaji imaju različite skupove instrukcija i zbog toga kompajleru mora da se naglasi kom uređaju je namenjen program.

Aplikacija koja omogućuje brz i efikasan način programiranja i kompajliranja naziva se IDE (*Integrated Development Environment*). Ove aplikacije pružaju okruženje programeru da ima bolji pregled strukture svog programa, da konfiguriše koji kompajler

će se koristiti za kompajliranje i da uveže potrebne biblioteke. Pojedini IDE-ovi pružaju mogućnost debugovanja u realnom vremenu, sa dodatnim hardverom.

3.2. API

Proizvođači uređaja najčešće uz sam hardver omogućuju programeru i već gotov skup biblioteka za efikasnije softversko programiranje, biblioteke napisane u tu svrhu su aplikacioni programski interfejs, API (*Application Programming Interface*). Svrha interfejsa je da podigne nivo apstrakcije pri programiranju uređaja, ovim se smanjuje potrebno znanje za programiranje neke funkcionalnosti. Interfejs stvara vezu između dva softverska nivoa, gde je jedan višeg nivoa apstrakcije od drugog.

3.3. RTOS

Namenski računarski sistemi se mogu koristiti za sisteme sa strogo definisanim vremenskim ograničenjima da se izvrši deo programa u toku izvršavanja sistema [7]. Sistemi koji moraju da poštuju vremenska ograničenja za pravilno funkcionisanje se nazivaju sistemi u realnom vremenu. Poštovanje strogih vremenskih ograničenja i bolje organizacije sistema u vremenu se postiže operativnim sistemom realnog vremena, RTOS (*Real-Time Operating System*).

RTOS je softver koji dopunjuje nedostatak hardverske računarske moći. Pomoću RTOS-a se emulira više procesorskih niti, iako procesor ne poseduje više jezgara, što dovodi do iluzije izvršavanja nekoliko programskih celina uporedo. Iluzija uporednog izvršavanja RTOS-ovih programskih niti se postiže čestim menjanjem niti koja se trenutno izvršava u procesoru. Programske niti se razlikuju i po visini prioriteta izvršavanja, čime se najvažnije niti izvršavaju prve, a niti istih prioriteta se smenjuju nakon određenog intervala izvršavanja. RTOS, za razliku od operativnih sistema na personalnim računarima, je najefikasniji u sistemima u kojima su niti sa kratkim vremenom izvršavanja.

FreeRTOS je otvoreni softver RTOS kernela za uređaje u namenskim sistemima.

3.4. Middleware

Posredni softver, na engleskom *Middleware*, je softver koji implementira neki API operativnog sistema ili drajvera hardvera [2]. On može da implementira komunikacione, sigurnosne, informacione ili upravljačke protokole. Korišćenjem posrednog softvera se podiže nivo apstrakcije programiranja mikrokontrolera.

3.5. HAL

Hardverski nivo apstrakcije HAL (*Hardware abstraction layer*) je sloj softvera koji je jedinstven za svaki uređaj i definiše skup rutina, protokola i interfejs za hardver uređaja. Ovim slojem je moguće pokrenuti isti aplikativni program na uređajima sa različitim arhitekturama hardvera. Korišćenje HAL biblioteka smanjuje potrebno znanje programera o arhitekturi uređaja.

Pojedini IDE-i omogućuju automatsko uvezivanje HAL biblioteka za odgovarajući uređaj.

4. ORGANIZACIJA REALIZOVANOG SISTEMA

Sistem za detekciju pokreta čine kontroler i senzor pokreta. Kontroler ima ulogu da obavlja komunikaciju sa korisnikom, da konfiguriše senzor na zahtev korisnika i da čita i obrađuje merenja senzora. Aplikativni program na kontroleru je realizovan korišćenjem operativnog sistema i prethodno pomenute funkcionalnosti su podeljene u programske niti.

4.1. Hardver u sistemu

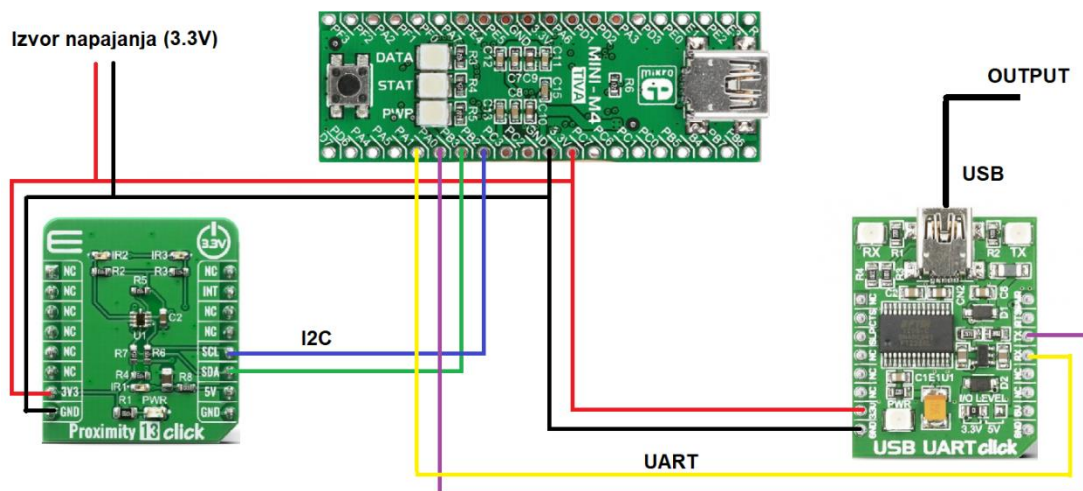
Hardver u sistemu čine mikrokontroler, senzor pokreta, periferija za prevođenje poruka iz UART protokola u USB protokol i napajanja.

Kontroler u sistemu je mikrokontroler na razvojnoj ploči i ima ulogu kontrolisanja sistema i obrađivanje merenja. Mikrokontroler čita komande koje korisnik upisuje i zavisno od njih izvršava odgovarajuće funkcionalnosti. Funkcionalnosti uključuju komunikaciju sa senzorom i informisanje korisnika o stanju sistema i rezultatima detekcije.

Razlog korišćenja posebne komponente za prevođenje u drugi interfejs a ne korišćenje tog interfejsa direktno sa razvojne ploče je značajna razlika u dužini linije UART i USB protokola. Korišćenjem UART protokola za komunikaciju sa korisnikom je ostavljena mogućnost da se sistem za detekciju odvija na većoj udaljenosti od računara na kom čitamo rezultate detekcije.

Senzor pokreta obavlja merenja i prosleđuje ih mikrokontroleru na obradu. Senzor komunicira sa mikrokontrolerom preko I2C protokola. Razlog korišćenja ovog protokola je brzina kojom se odvija ova komunikacija.

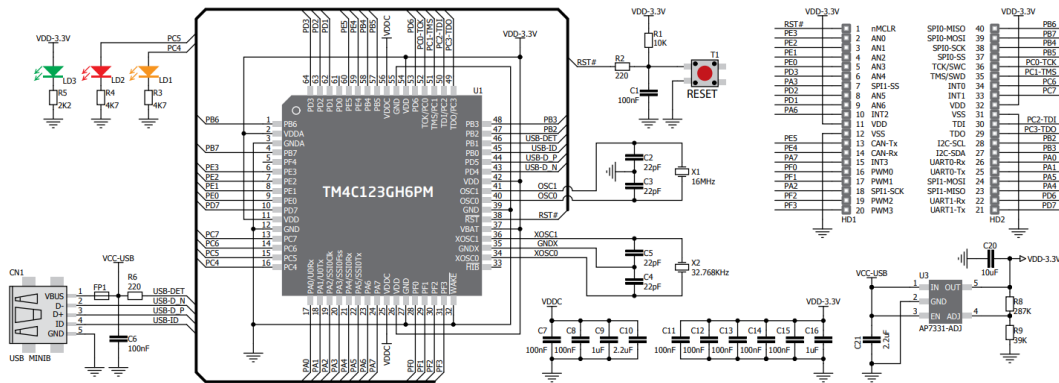
Šema realizovanog sistema se može videti na slici 4.1.



Slika 4.1 Šema sistema detektora pokreta

4.1.1.1. MINI-M4

MINI-M4 razvojna pločica za TivaC seriju sadrži mikrokontroler TM4C123GH6PM. Razvojnu pločicu proizvodi MikroElektronika [6]. Šemu razvojne pločice se može videti na slici 4.1.1.1.



Slika 4.1.1.1 Šema razvojne pločice MINI-M4 za TivaC [6]

Mikrokontroler TM4C123GH6PM iz TivaC serije ima 32-bitni ARM procesor Cortex-M4F i radi na učestanosti od 80MHz. Posедуje 256KB flash memorije i 32KB SRAM memorije.

Mikrokontroler kao i svaki drugi ima veoma komplikovanu arhitekturu da bi se objasnila detaljno, ali treba izdvojiti glavne karakteristike komunikacionih interfejsa.

Razvojna ploča MINI-M4 ima dva interfejsa za UART modul mikrokontrolera TM4C123GH6PM koji ima sledeće glavne osobine:

- FIFO 16 8-bitnih registara za slanje i primanje podataka
- Trigger signale za određene nivoe popunjenosti FIFO registra
- Generator i detektor karaktera za novi red
- Veličine podatka 5, 6, 7 i 8
- Prekidne signale za prazne ili pune FIFO registre

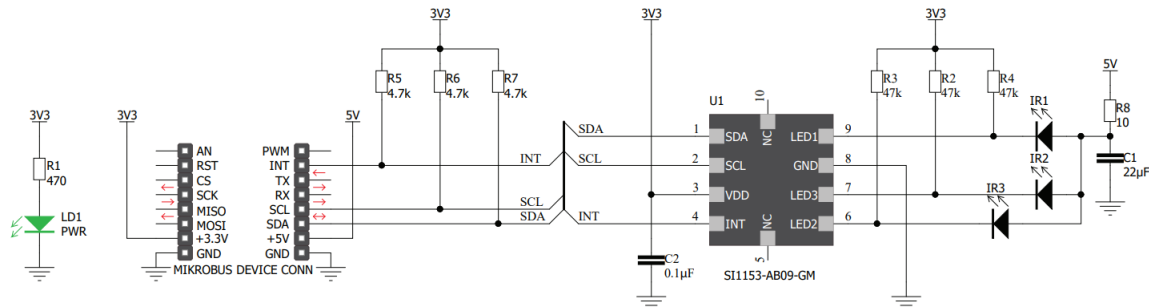
Razvojna ploča MINI-M4 ima interfejs za I2C modul mikrokontrolera TM4C123GH6PM koji ima sledeće glavne osobine:

- Modul može imati ulogu i master-a i slave-a
- Četiri brzine rada: 100 Kbps, 400 Kbps, 1Mbps, 3.33 Mbps
- Uklanjanje gliča
- 7-bitno adresiranje

Mikrokontroler se programira pomoću heksadecimalnog fajla u programu mikroBootloader USB HID v2.30. Zbog ovog ograničenja pri programiranju korišćeno okruženje Keil koje može da generiše heksadecimalni fajl.

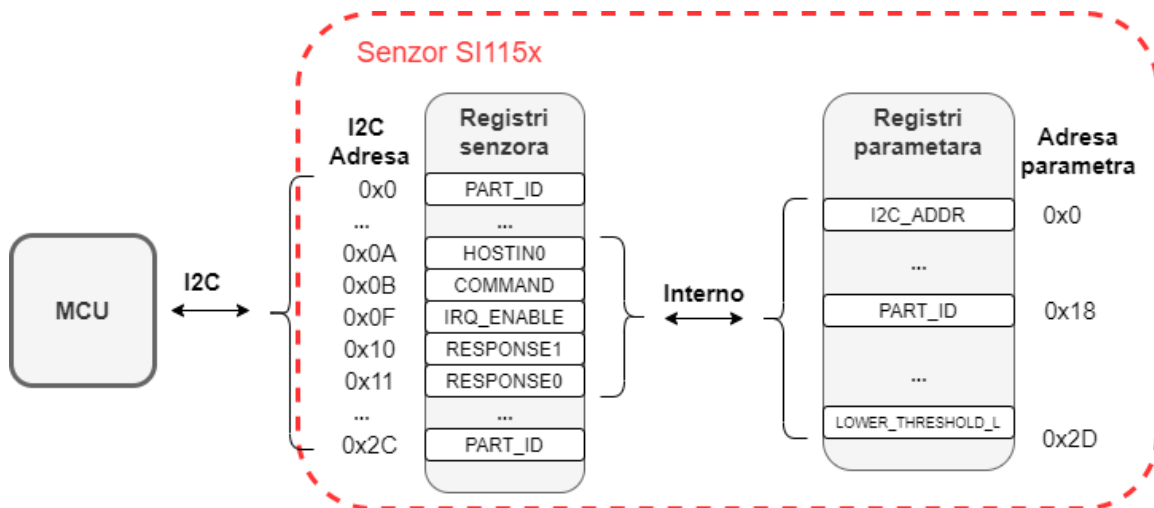
4.1.2. Proximity 13 click

Proximity 13 click je pločica sastavljena od senzora SI1153-AB09-GMR. Pločicu proizvodi MikroElektronika [6]. Šema senzora se može videti na slici 4.1.2.1.



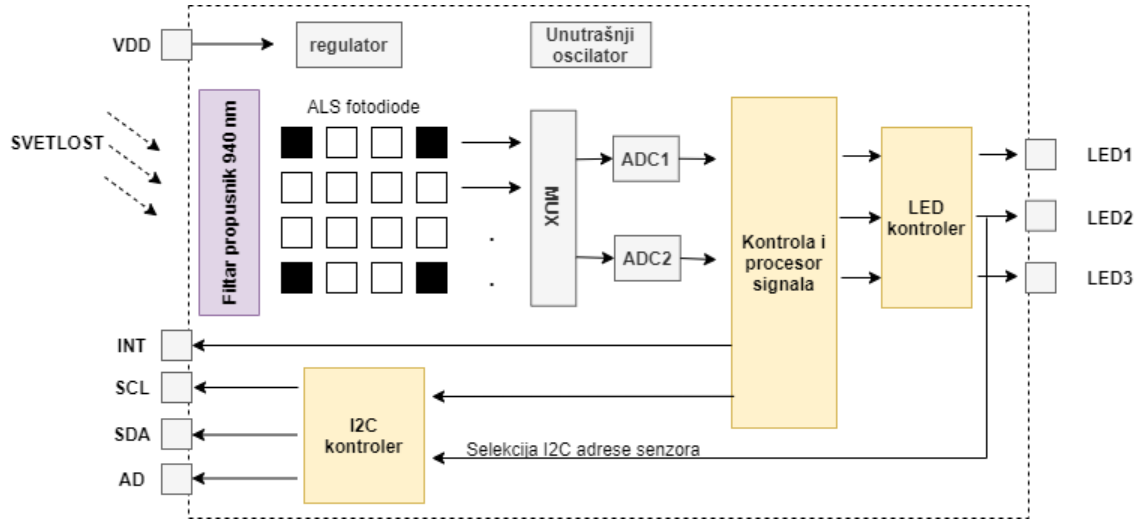
Slika 4.1.2.1 Šema senzora Proximity 13 click [6]

Ovaj senzor može da se konfiguriše da detektuje nivo osvetljenosti, udaljenost ili gestikulaciju. Komunikacija za konfigurisanje ili čitanje merenja se odvija preko I2C protokola. Senzoru je moguće zadati komandu da izvrši jedno merenje ili ga podesiti u režim autonomnog merenja. U toku autonomnog merenja senzor obavlja merenja sa konfigurabilnom periodom. Senzor može da se konfiguriše tako da nakon svakog izvršenog merenja pošalje prekid preko INT pina. Konfigurisanje senzora se postiže upisivanjem u kontrolne registre senzora i u registre parametara. Registrima parametara se ne može direktno pristupiti, već se upis vrši preko kontrolnih registra. Ilustracija čitanja i upisa u registre kontrolne i parametara se može videti na slici 4.1.2.2



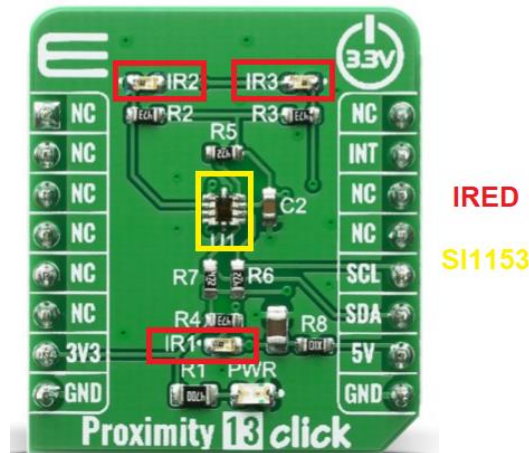
Slika 4.1.2.2 Ilustracija upisa i čitanja registara i parametra senzora

Senzor SI1153-AB09-GMR radi na principu IR detektora pokreta, gde su na senzor spojene tri IRED koje se mogu konfigurisati u različite režime rada pomoću senzora. IRED su raspoređene u L konfiguraciju, koja omogućuje detektovanje pokreta. Reflektovani zrak infracrvenog zraka koje emituju IRED se detektuju pomoću ALS fotodiode i filtra za propuštanje infracrvenog zraka. Blok šema senzora se može videti na slici 4.1.2.3



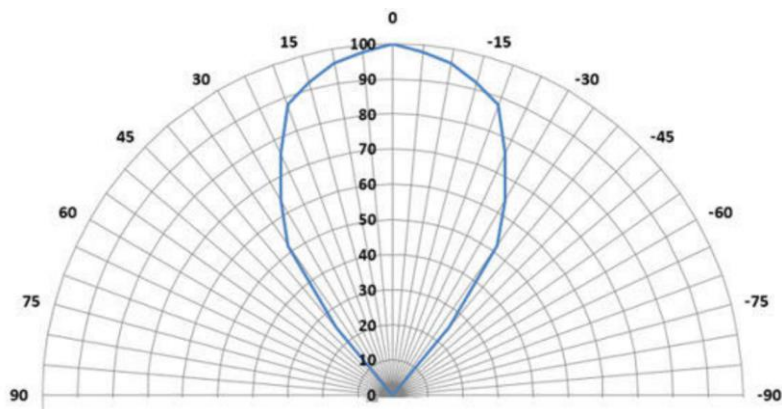
Slika 4.1.2.3 Blok šema senzora SI1153-AB09-GMR

Konfiguracija infracrvenih dioda koja omogućuje detektovanje pokreta je L konfiguracija. Naziv je dobila po rasporedu dioda koji se može videti i na slici 4.1.2.4.



Slika 4.1.2.4 Raspored senzora i dioda na proximity 13 click-u [6]

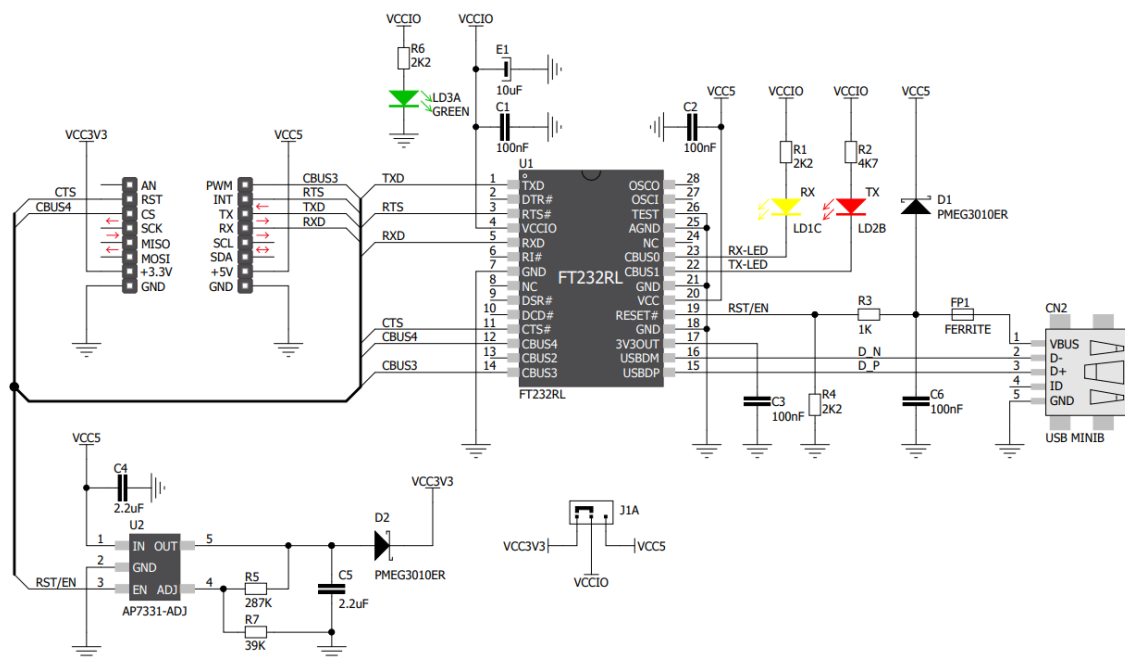
Vrednost koju detektuju fotodiode zavisi od ugla pod kojim reflektovani zrak upada. Osetljivost fotodiode od ugla reflektovanog zraka se može videti na slici 4.1.2.5.



Slika 4.1.2.5 Tipična ugaona osetljivost fotodioda (%)

4.1.3. USB UART click

USB UART click proizvodi MikroElektronika [6] i njen glavni deo je integrisano kolo FT232RL koje je sposobno da konvertuje serijske podatke iz USB protokola u UART protokol i obrnuto. Šema periferije se može videti na slici 4.1.3.1.



Slika 4.1.3.1 Šema USB UART click-a [6]

Komponenta USB UART click je pločica koja pruža interfejs između USB protokola i asinhronog serijskog prenosa podataka (UART). Pomoću ove komponente je moguće čitati merenja i detekcije pokreta koju je senzor detektovao na personalizovanom računaru.

Sistemu za detektovanje pokreta služi za čitanje stanja u kom se nalazi mikrokontroler, čitanje rezultata i zadavanje komandi sistemu.

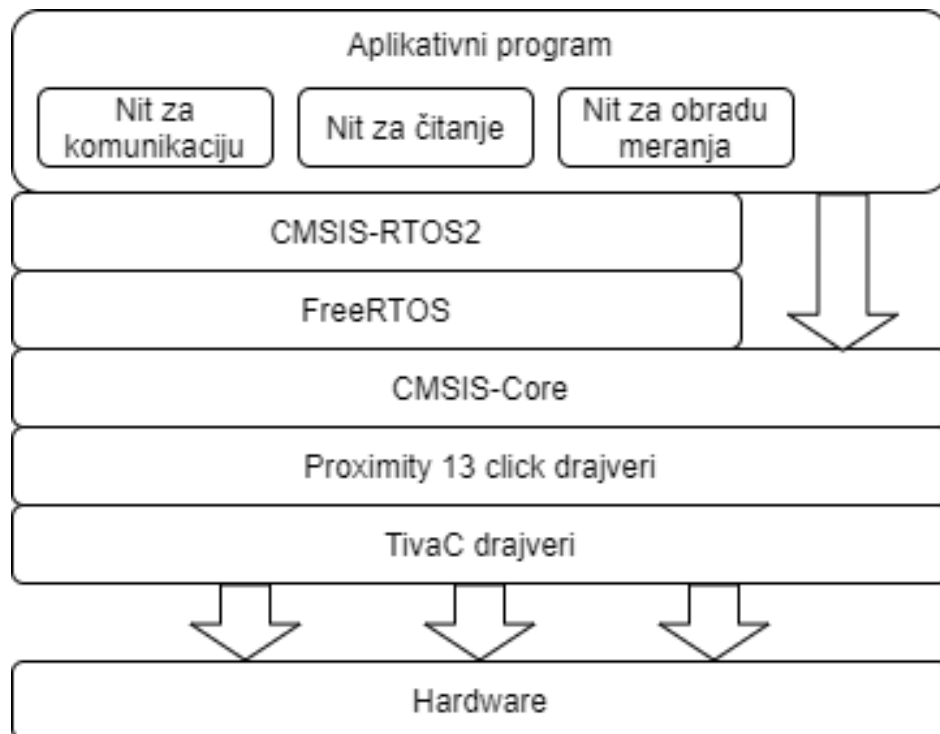
4.2. Softver sistema

Struktura softvera u sistemu se sastoji od aplikativnog programa, API za periferije i RTOS i podrške za mikrokontroler. Aplikacija koristi slojeve apstrakcije i operativni sistem u realnom vremenu da ostvari funkcionalnosti potrebne za detekciju pokreta i komunikaciju sa korisnikom.

Sistem za detekciju pokreta je realizovan kao sistem u realnom vremenu. Kao takav sistem on mora da zadovolji eksplicitan zahtev za vremenom odziva. Kontrola sistema da zadovolji adekvatne vremenske rokove se postiže uvođenjem operativnog sistema realnog vremena, tj. RTOS-a, u strukturu softvera. Sistem za detekciju pokreta koristi operativni sistem realnog vremena FreeRTOS. Ilustracija arhitekture sistema se može videti na slici 4.2.1.

Softver je realizovan korišćenjem Keilovog μ Vision IDE-a [3]. Keil je kompanija koja proizvodi različite alate za razvoj softvera.

Kompajliranje softvera je postignuto pomoću ARM kompajlera verzije 5. Ovaj kompajler je napravljen sa ciljem da optimizuje softver za ARM procesore. Glavna osobina mu je da smanji memorijski prostor aplikacije i to postiže odmotavanjem petlji, redne ekspanzije, prepoznavanje idioma i organizovanjem poziva instrukcija.



Slika 4.2.1 Arhitektura softvera u sistemu

4.2.1. CMSIS

CMSIS (*Common Microcontroller Software Interface Standard*) [3] je interfejs za softver Cortex mikrokontrolera. CMSIS je kreiran sa namerom da standardizuje softver na uređajima. Standardizacijom se postiže da se softver sa CMSIS-om može koristiti na različitim uređajima sa minimalnim izmenama. Ovim je moguće koristiti različite verzije i dodavati nove slojeve apstrakcije, a da se aplikativni program ne menja. Slojeve apstrakcije koji pruža CMSIS se dele na pakete kao što su paket za podršku RTOS-a CMSIS-RTOS, paket za podršku mašinskog učenja CMSIS-NN ili paket za podršku obradu signala CMSIS-DSP. Najniži nivo apstrakcije i drajvere za periferije mikrokontrolera CMSIS interfejs čini CMSIS-Core.

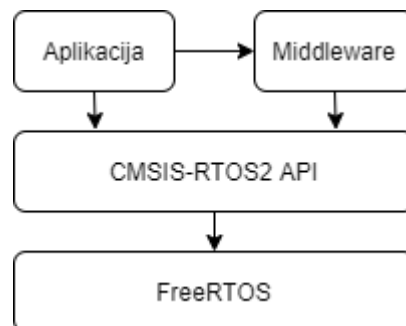
Softver sistema za detekciju pokreta koristi pakete CMSIS-Core i CMSIS-RTOS. Ilustracija ovoga se može videti na slici 4.2.1.1.

CMSIS-Core je API za Cortex procesor i periferije mikrokontrolera. CMSIS razlikuje dve verzije Cortex procesora, M i A. Sistem za detekciju pokreta koristi CMSIS-Core (M) jer mikrokontroler TM4C123GH6PM koristi Cortex-M4 procesor. Ovim paketom su realizovani i omogućeni:

- Sloj za apstrakciju hardvera (HAL) za Cortex-M procesorske registre
- Interfejs sa sistemskim greškama
- Metode za organizovanje heder fajlova
- Metode za inicijalizaciju sistema
- Interne funkcije za generisanje instrukcija za procesor
- Promenljiva sa vrednošću učestanosti sistemskog takta

CMSIS-RTOS2 je interfejs za RTOS softver na elektronskim uređajima sa Cortex procesorima. Interfejs je realizovan tako da promenom tipa RTOS-a koji se koristi aplikativni kod se ne menja. Od tipa RTOS-a podržava FreeRTOS i Keil RTX5, sistem za detekciju pokreta koristi FreeRTOS.

CMSIS-RTOS2 organizuje resurse mikrokontrolera i omogućuje programu da se izvršava pomoću paralelnih niti istovremeno. Aplikacije realizovane pomoću više programa, koji se izvršavaju konkurentno, mogu se lako implementirati u CMSIS-RTOS-u. Ovaj interfejs omogućuje odvajanje različitih funkcionalnosti aplikacija u odvojene niti i komunikaciju između njih.



Slika 4.2.1.1 Ilustracija interfejsa CMSIS-RTOS2

4.2.2. TivaC drajveri

Drajveri za korišćenje perifrija mikrokontrolera iz Tiva serije se nalaze u bibliotekama *The Texas Instruments® TivaWare™ Peripheral Driver Library*. Biblioteke podržavaju dva modela programiranja: direktan registarski i softverski apstraktni pristup. Po želji programera modeli se mogu koristiti pojedinačno ili u kombinaciji.

Model direktnog programiranja registra podrazumeva API za upisivanje vrednosti u registre unutrašnjih perifrija mikrokontrolera. Kompajliran kod direktnog programiranja registara je efikasniji i zauzima manju memoriju. Za uspešno programiranje ovim modelom potrebno je detaljno znanje funkcionalnosti registara u mikokontroleru. Neophodno je znati značenja bitskih polja registra, njegovu vezu sa drugim registrima i redosled kojim treba da se promene za ispravan rad. Adrese registara su predefinisane makroima u fajlovima određenim prema verziji mikrokontrolera.

Model softverskog programiranja izoluje programere od kompleksnosti arhitekture internih perifrija mikrokontrolera. Ovim modelom biblioteka pruža API za korišćenje I programiranje internih perifrija bez poznavanja njihove realizacije. Upis vrednosti u bitsko polje registara da bi se konfigurisao režim rada perifrije se postiže pozivom funkcije iz biblioteke koja konfiguriše režim rada perifrije. Značajno smanjenje programerskog vremena je prednost ovog modela u odnosu na model direktnog registarskog programiranja. Mana softverskog modela je smanjene performanse u odnosu na pravilno korišćenje modela direktnog registarskog.

Kombinovanim korišćenjem oba modela se postiže optimalno programiranje. Za pojedine perifrije koje nemaju strog vremenski rok za komunikaciju sa procesorom ili spoljnim perifrijama se može koristiti model softverskog programiranja, dok za vremenski važne funkcionalnosti je najbolje koristiti model direktnog programiranja registara.

U sistemu za detekciju pokreta korišćena je kombinacija oba modela za postizanje optimalne realizacije.

4.2.3. Drajveri za senzor

Drajveri za senzor pokreta su implementirani u toku implementiranja softvera za detekciju. Ovom implementacijom je kreirana biblioteka koja koristi drajvere za uređaje iz TivaC serije da se definiše interfejs za kontrolisanje senzora. Koristi drajver za I2C komunikaciju. Interfejs biblioteke za drajvere senzora je realizovan skupom funkcija koje upisuju ili čitaju vrednosti registara senzora. Funkcije poštuju vremenske intervale potrebne da senzor reaguje na primljenu poruku.

Od implementiranog interfejsa bitno je izdvojiti funkcije:

- **Si115xReadFromRegister** i **Si115xWriteToRegister**, koje upisuju ili čitaju iz registara Si115 senzora
- **Si115xSendCommand**, **Si115xWaitUntilSleep**, **Si115xPause** i **Si115xReset**, koje kontrolišu ili prate stanje senzora
- **Si115xParamSet** i **Si115xParamRead**, koje upisuju ili čitaju vrednosti parametara senzora

5. REALIZACIJA APLIKACIJE

Nakon povezivanja hardvera i definisanja strukture softvera jedini preostali deo sistema, da bi sistem funkcionisao, je definisanje aplikativnog programa softvera sistema. Zadatak aplikativnog programa je da konfiguriše i obavlja komunikaciju između kontrolera i periferija, da konfiguriše senzor u odgovarajući režim rada, da obrađuje merenja senzora i obaveštava o detektovanom pokretu.

Po pokretanju aplikacije prvo se konfiguriše hardver mikrokontrolera, da bi se omogućio rad mikrokontrolera i komunikacionih modula.

Režim rada sistema se može videti i u LED didodama MINI-M4, koje se osposobljavaju pri konfiguraciji hardvera u programu.

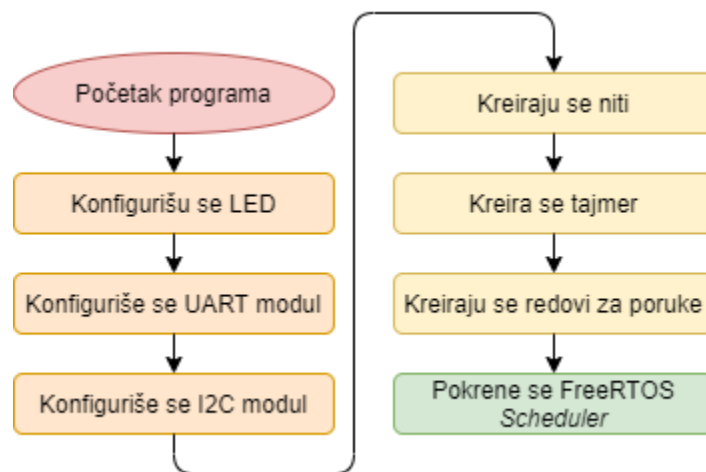
Program korišćenjem API-a podešava UART protokol da radi na nultom portu sa brzinom prenosa podataka 9600 *bps* i dozvoli pojavu prekida ukoliko se određena količina podataka nalazi u FIFO registrima. Prag pojave prekida u slučaju popunjenosti FIFO registra je stavljena na minimalnu vrednost.

Komunikacioni modul I2C se konfiguriše na mikrokontroleru kao *master* modul i podešava se I2C adresa senzora kao adresa *slave*-a.

Dijagram opisanog toka aplikativnog programa se može videti na slici 5.1.

Nakon konfigurisanja hardvera kreiraju se objekti operativnog sistema realnog vremena. Objekti koji se kreiraju su tri niti, tajmer i tri FIFO reda za poruke.

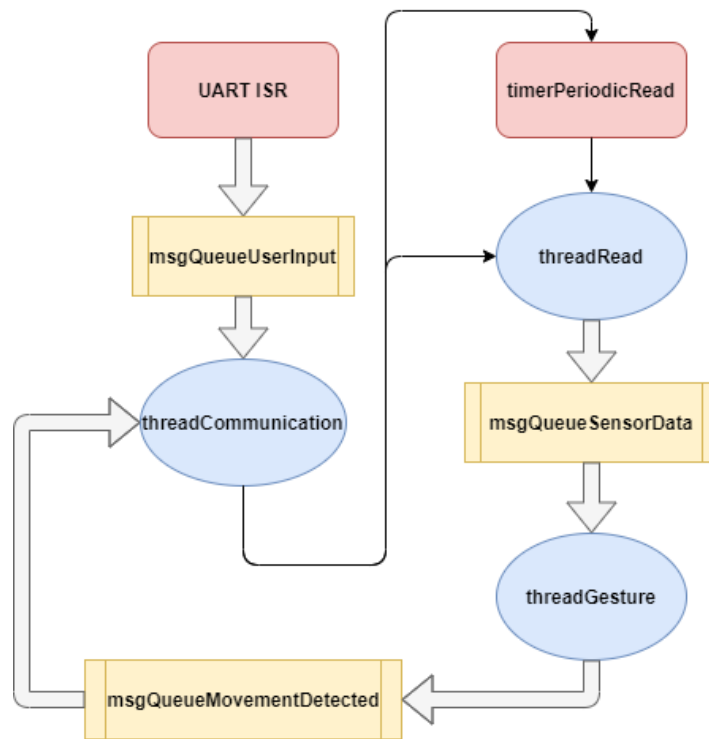
Nakon svih konfiguracija koje je aplikativni program uradio, aplikacija se prepušta operativnom sistemu realnog vremena. U početku rada operativnog sistema pozivaju se niti koje su kreirane da izvrše svoj kod. Niti su kreirane takve da nikad ne završavaju program, već se vraćaju na početak toka programa, i zbog toga nikad neće biti uništene.



Slika 5.1 Dijagram toka aplikativnog programa

5.1. Organizacija niti

Aplikativni program je realizovan pomoću tri niti čije izvršavanje obavlja operativni sistem realnog vremena FreeRTOS. Niti su osmišljene sa idejom da izvršavaju jednu manju celinu funkcionalnosti sistema. Aplikativni program poseduje nit za interfejsnu komunikaciju, za čitanje merenja sa senzora i za obradu izmerenih vrednosti. Organizacija niti se može videti na slici 5.1.1.



Slika 5.1.1 Blok šema organizacije niti

Nit za interfejsnu komunikaciju se u programu naziva **threadCommunication** i ima ulogu da održava komunikaciju sa korisnikom i pokreće aplikaciju za detektovanje sistema. Komunikacija se obavlja preko UART protokola, zbog čega se u prekidnoj rutini UART-a šalju karakteri preko reda sa podacima niti za komunikaciju. U slučaju zahteva startovanja aplikacije, nit dozvoli rad tajmeru koji periodično odblokira nit za čitanje merenja.

Nit za komunikaciju **threadCommunication** je nit najvišeg prioriteta u sistemu.

Nit zadužena za očitavanje merenja sa senzora u programu se naziva **threadRead** i pokreće se sa određenom periodom ako je korisnik pokrenuo merenja. Startovanjem tajmera počinje periodično odblokiranje niti, čime nit izvršava svoj program iznova. Nit je zadužena da pokrene merenje na senzoru, očita izmerene vrednosti i prosledi ih na obradu.

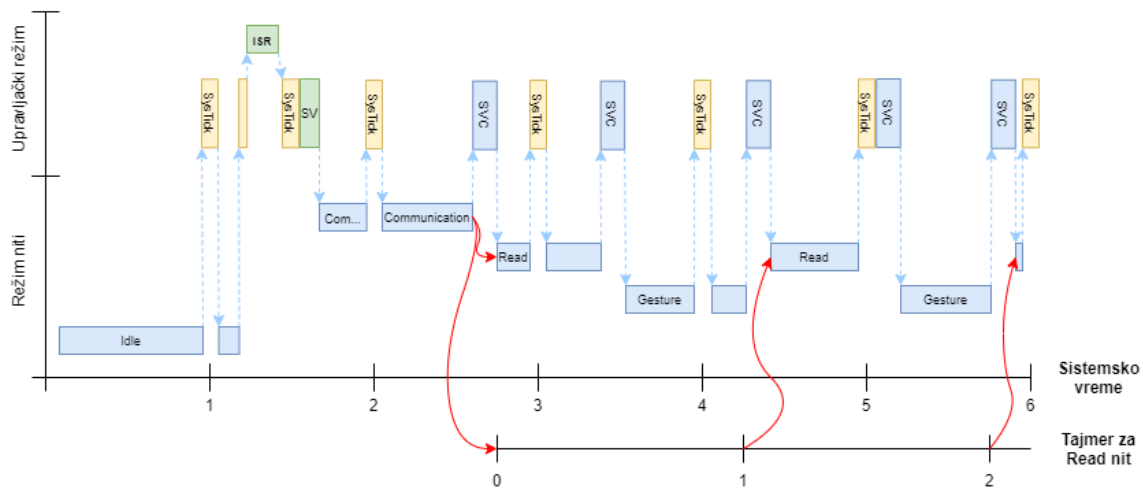
Nit za čitanje merenja **threadRead** je srednjeg prioriteta u sistemu.

Nit zadužena za obradu izmerenih vrednosti se u programu naziva **threadGesture** i ima ulogu u sistemu da pomoću izmerenih vrednosti sa senzora detektuje pokret. Vrednosti merenja se prosleđuju od niti **threadRead** i informacije o detektovanju pokreta se prosleđuju **threadCommunication** preko odgovarajućih redova sa porukama.

Nit za obradu izmerenih vrednosti **threadGesture** je najnižeg prioriteta u sistemu.

Sve dok korisnik ne zada sistemu komandu da započne rad, aktivna nit u sistemu je **Idle** nit. Idle nit je nit koju operativni sistem implementira i ona je aktivna ako su ostale niti u sistemu blokirane i neaktivne. Kada korisnik zada komandu preko UART protokola, desi se UART prekid. Tokom opsluživanja prekida primljeni karakter se prosleđuje niti za komunikaciju. Nit za komunikaciju zavisno od komande odradi određeni deo programa. U slučaju da je zadata komanda za početak rada sistema, nit za komunikaciju pokreće periodičan tajmer. Tajmer na kraju svakog odbrojavanja informiše i odblokira nit za čitanje merenja senzora. Nakon dohvaćanja vrednosti merenja, vrednosti se prosleđuju niti za obradu podataka. Vremenski prikaz opisanog ponašanja sistema se može videti na slici 5.1.2.

Promena trenutno aktivne niti se postiže sa upravljačkim režimom operativnog sistema.



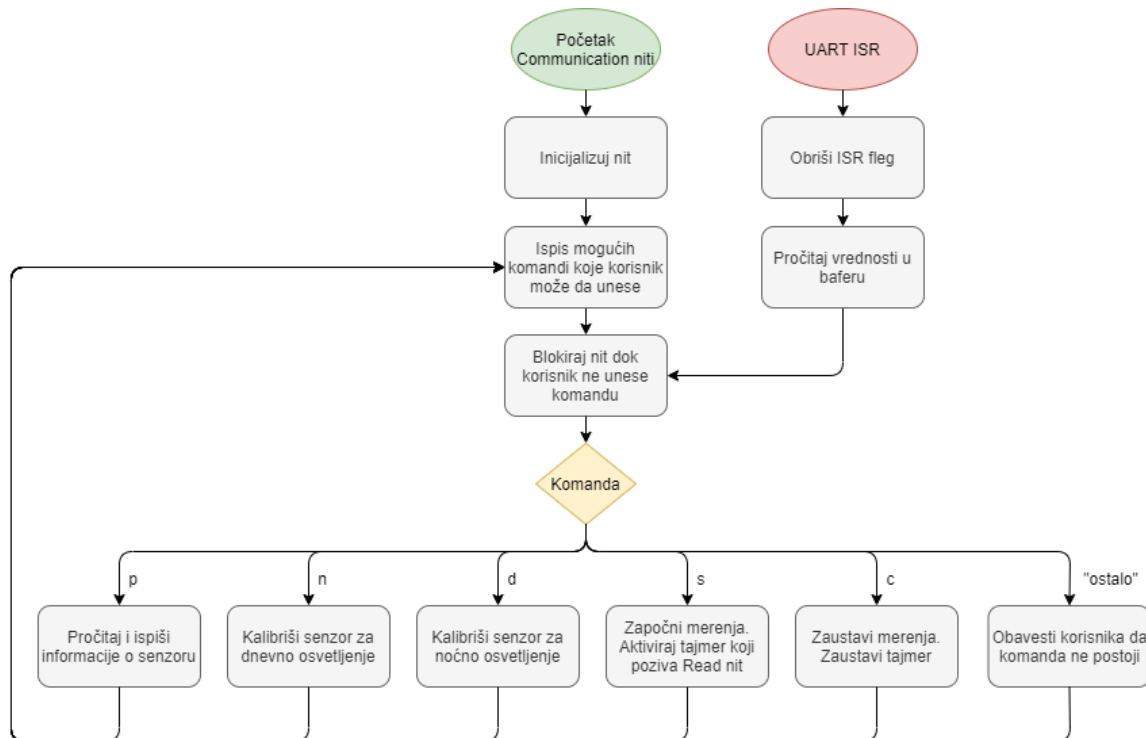
Slika 5.1.2 Ilustracija vremenskog prikaza izvršavanja niti

5.1.1. Nit za komunikaciju sa korisnikom

Nit za komunikaciju sa korisnikom, u programu nazvana **threadCommunication**, obavlja komunikaciju sa korisnikom. Ova nit je najvišeg prioriteta u sistemu i uvek ima prednost u odnosu na druge niti pri izvršavanju. Tok programa niti je takav da će se nakon kreiranja inicijalizovati i odmah krenuti u neprestano izvršavanje drugog dela programa. Dijagram toka niti se može videti na slici 5.1.1.1.

Prvim delom programa niti se smatra inicijalizacija niti. Inicijalizacija niti namešta prioritet niti.

Drugi deo programa je deo programa koji se neprestano izvršava. Prvi korak ovog dela je ispisivanje korisniku dostupne komande koje može da zada sistemu. Komande se zadaju u vidu karaktera, dostupne komande se mogu videti na slici 5.1.1.1. Nit se nalazi u blokiranom stanju sve dok joj se ne prosledi vrednost iz prekidne rutine UART-a.



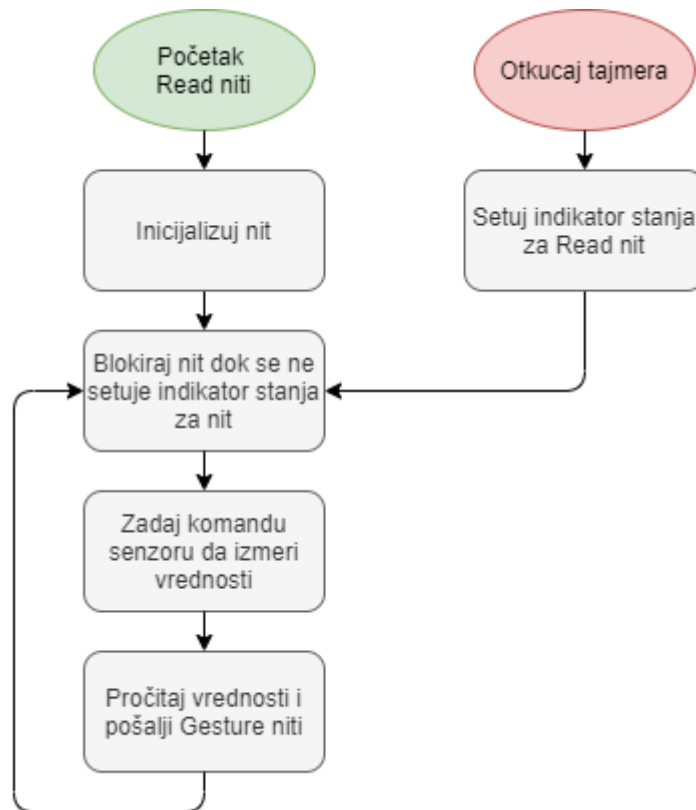
Slika 5.1.1.1 Dijagram toka niti za komunikaciju

5.1.2. Nit za čitanje podataka iz senzora

Nit za čitanje vrednosti merenja iz senzora, u programu nazvan **threadRead**, pokreće senzor da obavi merenje i očitava vrednosti izmerene. Ova nit je srednjeg prioriteta u sistemu i ima prednost u odnosu na niti nižeg prioriteta. Tok programa niti je takav da će se nakon kreiranja inicijalizovati i odmah krenuti u neprestano izvršavanje drugog dela programa. Dijagram toka niti se može videti na slici 5.1.2.1.

Prvim delom programa niti se smatra inicijalizacija niti. Inicijalizacija niti namešta prioritet niti.

Drugi deo programa je deo programa koji se neprestano izvršava. Nit po ulasku u ovaj deo programa prelazi u blokirano stanje. Jedini način da se odblokira ova nit je da korisnik unese komandu da se započne merenje. Po unošenju ove komande nit za komunikaciju odblokira nit za merenje i aktivira rad tajmera u sistemu. Tajmer periodično odblokira nit za merenje sve dok se ne deaktivira komandom korisnika. Izmerene vrednosti merenja senzora se prosleđuju niti za obradu podataka.



Slika 5.1.2.1 Dijagram toka Read niti

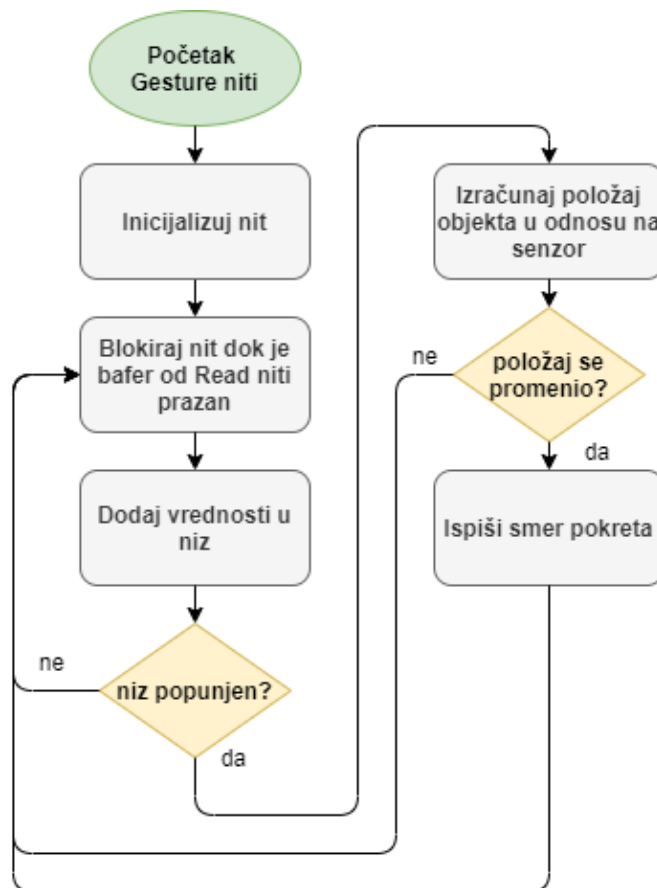
5.1.3. Nit za računanje pozicije i detektovanje pokreta

Nit za obradu izmerenih vrednosti, u programu nazvan **threadGesture**, računa poziciju i detektuje pokrete objekta ispred senzora. Ova nit je najnižeg prioriteta u sistemu, druge niti imaju prednost u odnosu na nit za obradu podataka. Tok programa niti je takav da će se nakon kreiranja inicijalizovati i odmah krenuti u neprestano izvršavanje druga dva dela programa. Dijagram toka niti se može videti na slici 5.1.3.1.

Prvim delom programa niti se smatra inicijalizacija niti. Inicijalizacija niti namešta prioritet niti.

Drugi deo programa je deo programa koji se neprestano izvršava. Nit po ulasku u ovaj deo programa prelazi u blokirano stanje. Jedini način da se odblokira ova nit je da joj se proslede izmerene vrednosti. Vrednosti se upisuju u niz sve dok se ne popuni određen broj odabiraka.

Treći deo programa izračunava poziciju objekta u odnosu na senzor pomoću niza sa podacima. Ako je pozicija promenjena u odnosu na prethodnu poziciju, računa se smer kretanja i ispisuje se korisniku.

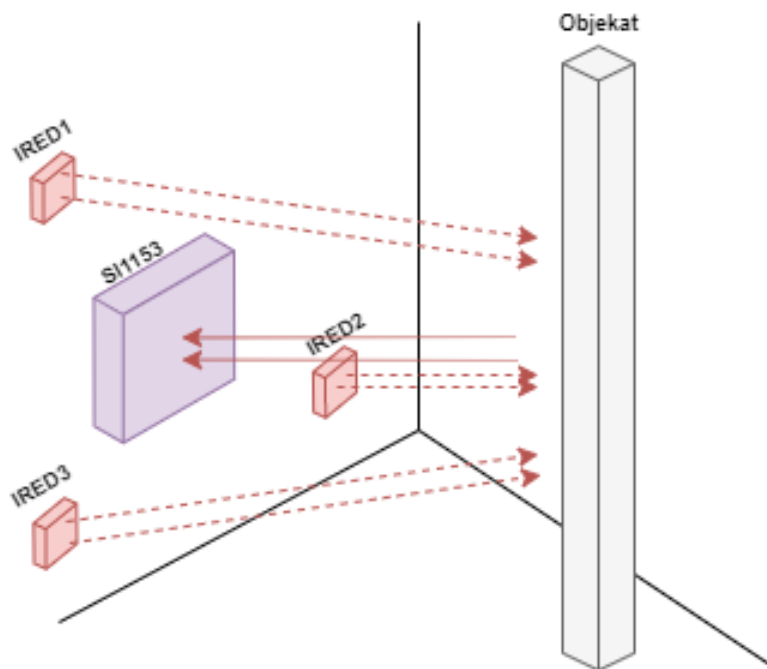


Slika 5.1.3.1 Dijagram toka Gesture niti

5.2. Detektovanje pokreta

Detektovanje pokreta objekta ispred detektora pokreta se postiže upoređivanjem vrednosti reflektovane svetlosti koju je detektor emitovao prema objektu. Emitovanje infracrvenih dioda se može konfigurisati. U sistemu za detektovanje pokreta diode su konfigurisane da rade naizmenično po pokretanju merenja.

Konfiguracija L infracrvenih dioda dovodi do pojave da emitovani zraci infracrvene diode, koja se nalazi najbliže objektu, budu reflektovani sa većom energijom. Merenjem energije reflektovanih zraka se određuje udaljenost objekta od izvora svetlosti. Upoređivanjem vrednosti koje senzor SI1153 izmeri može se odrediti položaj predmeta u odnosu na senzor. Pamćenjem položaja i upoređivanjem sa novim položajima nakon određenog vremenskog intervala se može odrediti smer kretanja objekta u odnosu na senzor. Ilustracija L konfiguracije se može videti na slici 5.2.1.



Slika 5.2.1 Ilustracija L konfiguracije detektora pokreta

6. REZULTATI TESTIRANJA APLIKACIJE

Uspešnost detektovanja pokreta pomoću realizovanog sistema je prezentovana kroz rezultate četiri eksperimenta. Eksperimenti su osmišljeni da testiraju realizovan sistem da detektuje različite vrste pokreta, što ujedno testira različite primene realizovanog sistema.

Rezultati detektovanja se bolje uočavaju preko međurezultata. Kod sistema za detekciju pokreta najpogodniji međurezultat su struje fotodioda tokom merenja. Učitavanje ovih podataka je postignuto slanjem vrednosti preko UART-a personalnom računar, memorisanje tih vrednosti i iscrtavanjem grafika pomoću njih.

Memorisanje vrednosti je odrađeno preko programa *RealTerm:Serial Capture Program* i iscrtavanje grafika preko Excel-a.

Pre svakog eksperimenta sistem je adekvatno kalibrisan da bi bio funkcionalan sa osvetljenjem u prostoriji.

Prvi eksperiment testira mogućnost realizovanog sistema da detektuje osobu koja se kreće ispred senzora. Detekcija kretanja osobe ispred senzora je glavna funkcionalnost sistema za bezbednost. Osim sistema za bezbednost, primena ove detekcije se nalazi i u pametnim kućama.

Drugi eksperiment testira mogućnost realizovanog sistema da detektuje gestikulaciju na malom odstojanju. Detekcija gestikulacije ima primenu u komercijalnim proizvodima sa funkcionalnošću zadavanja komande pomoću gestikulacije.

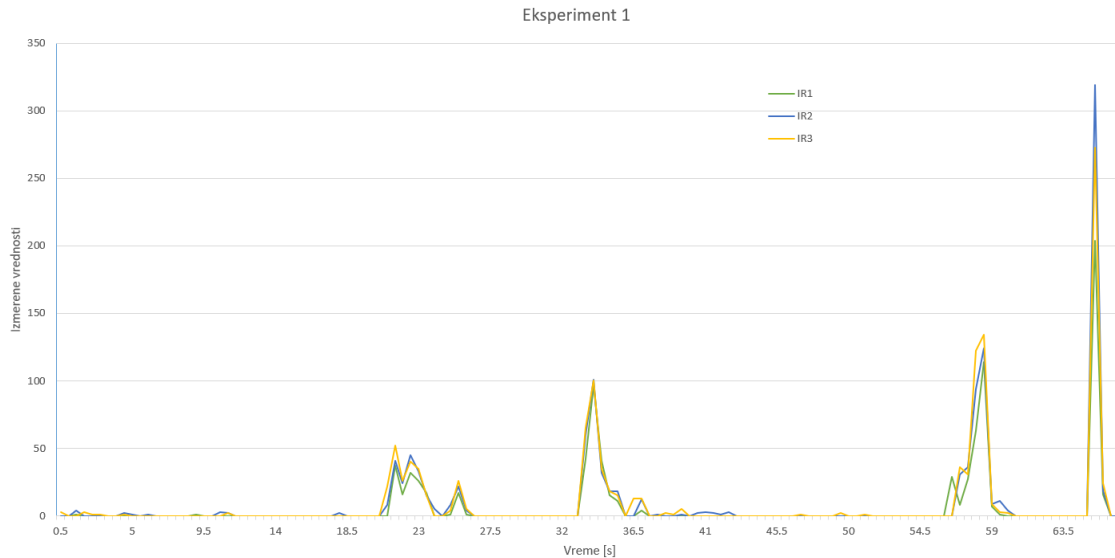
Treći eksperiment testira mogućnost realizovanog sistema da detektuje predmet koji pada ispred njega. Detekcija kretanja predmeta dok pada ispred sistema ima primenu u industrijskim ili sigurnosnim sistemima.

Četvrti eksperiment testira mogućnost realizovanog sistema da detektuje prolazak predmeta većih dimenzija u odnosu na drugi eksperiment. Detekcija kretanja većih dimenzija neposredno ispred senzora ima primenu u industrijskim sistemima sa funkcionalnošću da prate kretanje proizvoda.

Eksperimenti se nisu odvojili u potpuno izolovanom prostoru niti su ponavljani više od par puta. Njih nije izvodio stručan tim i treba ih tumačiti odgovarajuće.

6.1. Prvi eksperiment

Eksperiment je odvijen kretanjem osobe ispred senzora u jednom pa zatim u drugom smeru dvaput, na odstojanju većem od pola metra. Udaljenost i brzina kretanja nisu održavane na konstantne vrednosti tokom merenja. Na slici 6.1.1. su prikazani rezultati merenja tokom prvog eksperimenta.



Slika 6.1.1 Izmerene vrednosti tokom eksperimenta 1

Rezultati eksperimenta pokazuju uspešno detektovanje pokreta u sva četiri prolaska.

U toku prvog prolaska ispred senzora, kretanje ima smer od diode IR3 do diode IR2. Algoritam uspešno prepoznaje smer kretanja.

U drugom prolasku ispred senzora, kretanje ima smer od diode IR2 do diode IR3. Algoritam ne prepoznaje smer kretanja, jer su izmerene vrednosti skoro pa jednake.

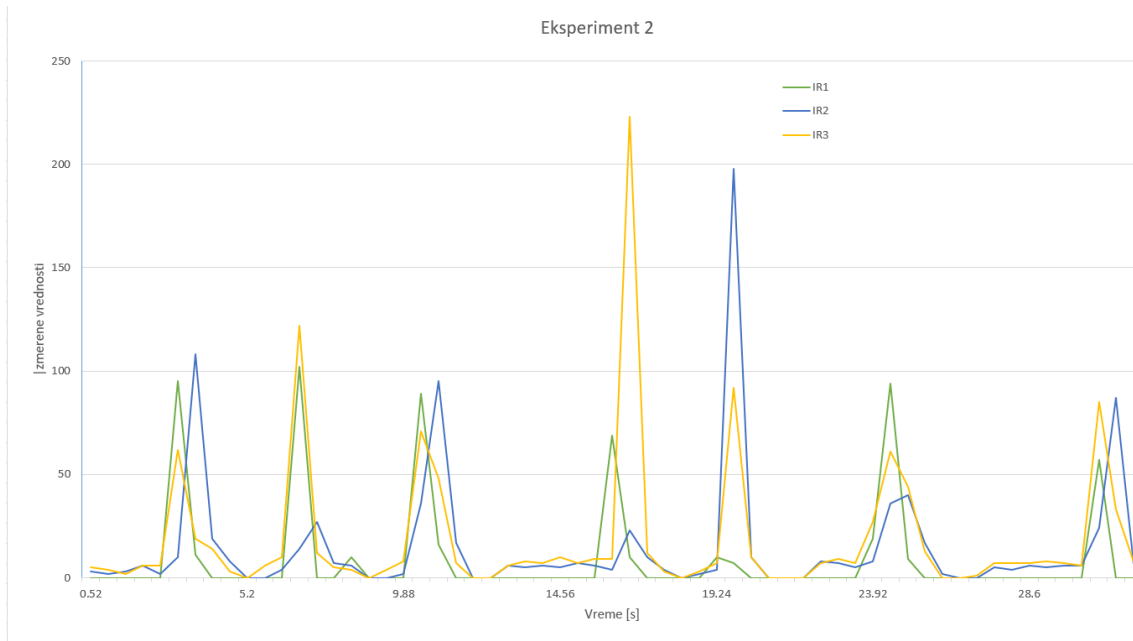
U trećem prolasku ispred senzora, kretanje ima smer od diode IR3 do diode IR2. Algoritam uspešno prepoznaje smer kretanja.

U četvrtom prolasku ispred senzora, kretanje ima smer od diode IR2 do diode IR3. Algoritam uspešno prepoznaje smer kretanja.

Izmerene vrednosti od sva tri kanala su približne jer se objekat kreće na većem odstojanju od senzora.

6.2. Drugi eksperiment

Drugi eksperiment je odvijen na značajno manjoj udaljenosti od prvog eksperimenta. Eksperimentom je testirana mogućnost detektovanja gestikulacije. Gestikulaciju čini pokret šake iznad senzora. Na slici 6.2.1. su prikazani rezultati merenja tokom drugog eksperimenta.



Slika 6.2.1 Izmerene vrednosti tokom eksperimenta 2

U toku prvog prolaska ispred senzora, kretanje ima smer od diode IR2 do diode IR3. Algoritam uspešno prepoznaje smer kretanja.

U drugom prolasku ispred senzora, kretanje ima smer od diode IR3 do diode IR2. Algoritam uspešno prepoznaje smer kretanja.

U trećem prolasku ispred senzora, kretanje ima smer od diode IR2 do diode IR3. Algoritam uspešno prepoznaje smer kretanja.

U četvrtom prolasku ispred senzora, kretanje ima smer od diode IR3 do diode IR2. Algoritam ne prepoznaje smer kretanja, jer je velika razlika u vrednostima.

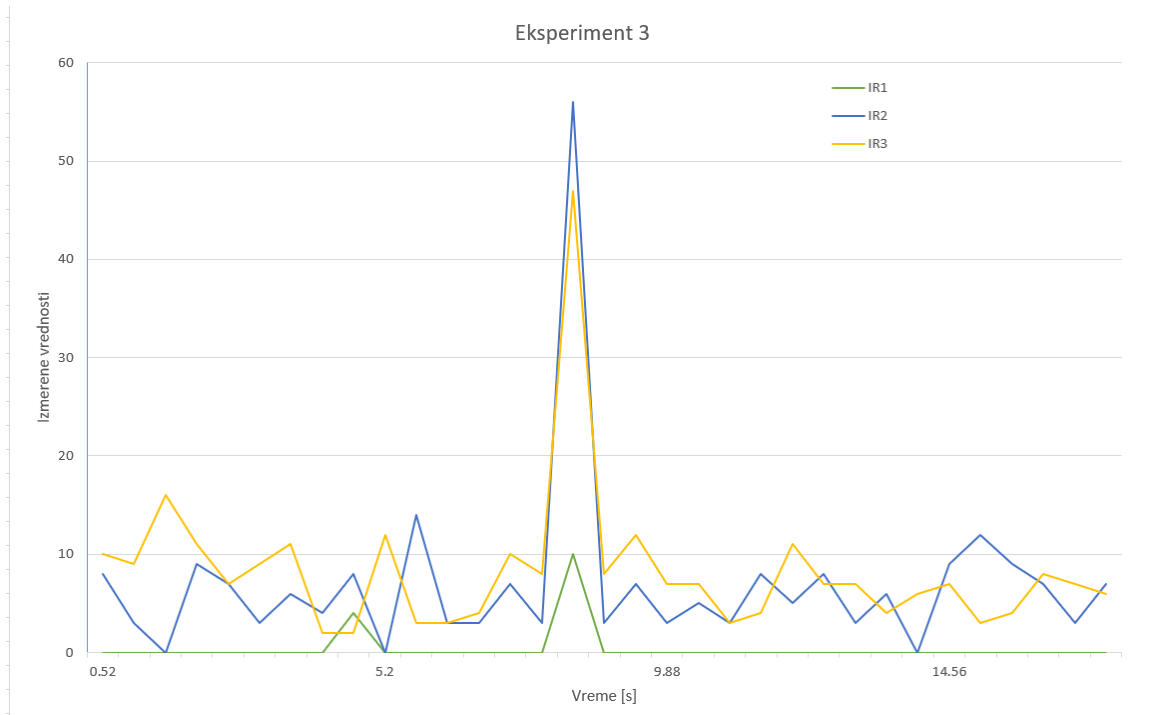
U petom prolasku ispred senzora, kretanje ima smer od dioda IR2 i IR3 do diode IR1. Algoritam uspešno prepoznaje smer kretanja.

U šestom prolasku ispred senzora, kretanje ima smer od diode IR1 do dioda IR2 i IR3. Algoritam uspešno prepoznaje smer kretanja.

U sedmom prolasku ispred senzora, kretanje ima smer od dioda IR2 i IR3 do diode IR1. Algoritam uspešno prepoznaje smer kretanja.

6.3. Treći eksperiment

Treći eksperiment u nizu testiranja aplikacije testira mogućnost detektovanja predmeta koje pada ispred senzora. Predmet je pušten sa visine od jednog metra u odnosu na senzor. Izabran predmet za ovaj eksperiment je kocka sunđera. Na slici 6.3.1. su prikazani rezultati merenja tokom trećeg eksperimenta.

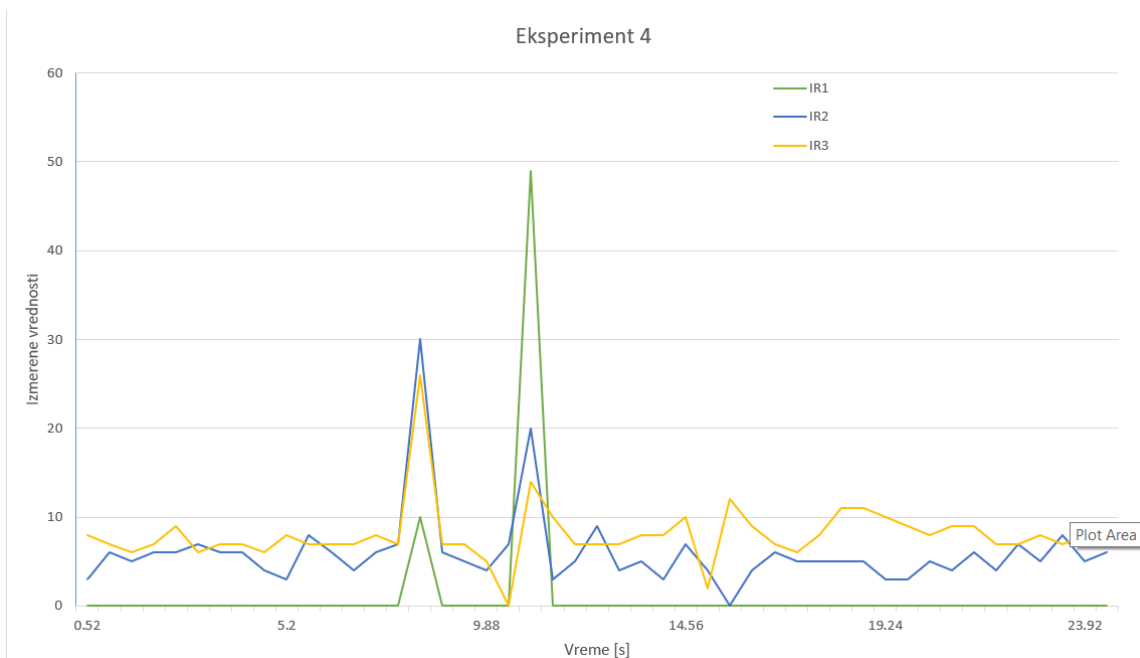


Slika 6.3.1 Izmerene vrednosti tokom eksperimenta 3

U toku prolaska ispred senzora, kretanje ima smer od dioda IR2 i IR3 do diode IR1. Algoritam uspešno prepoznaje smer kretanja.

6.4. Četvrti eksperiment

Četvrtim eksperimentom se testira mogućnost detektovanja ravnog i simetričnog predmeta na malom odstojanju. Za ovaj eksperiment je korišćena sveska formata A5. Na slici 6.4.1. su prikazani rezultati merenja tokom četvrtog eksperimenta.



Slika 6.4.1 Izmerene vrednosti tokom eksperimenta 4

U toku prvog prolaska ispred senzora, kretanje ima smer od dioda IR2 i IR3 do diode IR1. Algoritam uspešno prepoznaje smer kretanja.

U drugom prolasku ispred senzora, kretanje ima smer od diode IR1 do dioda IR2 i IR3. Algoritam uspešno prepoznaje smer kretanja.

7. ZAKLJUČAK

U praksi ovakav sistem ne bi morao da bude realizovan sa operativnim sistemom realnog vremena. Softver bi se brže izvršavao i manji memorijski prostor bi zauze, ali ipak ne bi mogao da bude tako lako nadogradiv kao što je sa trenutnom realizacijom. Realizacija u realnom vremenu omogućuje proširanje funkcionalnosti i uvođenje novih senzora i aktuatora, čime bi se mogao realizovati ceo jedan industrijski proces.

U radu je analiziran senzor sa lošijim karakteristikama, što je dovelo do poteškoća pri kalibrisanju tokom eksperimentisanja. Pri realizaciji ovakvog sistema sa ciljem da tačnost detekcije bude veća, potrebno je izabrati kvalitetniji senzor.

Sistem za detekciju je realizovan sa IR detektorom, dok u praksi ovakav detektor ne bi bio najadekvatniji, što zavisi od okolnosti u kojim se očekuje da sistem radi. Zanimljiva tema za istraživanje je testiranje sistema za detekciju sa IR detektorom sa više dioda u konfiguraciji.

Jedan od budućih pravaca nadogradnje sistema za detekciju pokreta bio bi uvođenje automatske kalibracije sistema. Ovim bi se značajno poboljšao sistem i njegova portabilnost i primenljivost. U slučaju da se sistem adaptira na jednoj štampanoj ploči, dioda koja pokazuje da je uređaj uključen bi trebalo da se udalji od fotodioda koje detektuju zrake.

Drugi pravac razvoja je nadogradnja sistema da se samostalno konfiguriše i krene u rad, bez komandi koje korisnik zadaje. Ovim bi se realizovao sistem koji je odmah spreman za rad, a korisnik bi na ovaj način imao samo informaciju o detekciji pokreta i smeru.

8. LITERATURA

- [1] Lazar Saranovac, Ivan Popović, Namenski računarski sistemi, 2016
- [2] Peter Marwedel, Embedded System Desing: Embedded Systems Foundations of Cyber-Physical Systems, 2011.
- [3] Keil, CMSIS & μ Vision, <https://www.keil.com>
- [4] FreeRTOS, <https://www.freertos.org>
- [5] RealTerm, RealTerm:Serial Capture Program, <https://realterm.sourceforge.io>
- [6] MikroElektronika, <https://www.mikroe.com>
- [7] Ivan Popović, Osnovni pojmovi i koncepti sistema u realnom vremenu, 2021
- [8] Antonije Đorđević, Osnovi elektrotehnike 3. deo – Elektromagnetizam, 2016
- [9] LightSearch, *Light Guide*, <https://www.lightsearch.com/resources/lightguides/sensors.html>
- [10] MachineDesign, *Proximity Sensors Compared: Inductive, Capacitive, Photoelectric, and Ultrasonic*, <https://www.machinedesign.com/automation-iiot/sensors/article/21831577/proximity-sensors-compared-inductive-capacitive-photoelectric-and-ultrasonic>
- [11] Marko Barjaktarović, Fizičko-tehnička merenja, *Senzori prisustva i kretanja*, 2012

SPISAK SKRAĆENICA

IREDA	<i>Infrared Emitting Diodes</i>
IR	<i>Infrared</i>
ALS	<i>Ambient light sensor</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
TXD	<i>UART transmitter line</i>
RXD	<i>UART receiver line</i>
FIFO	<i>First in First Out</i>
bps	<i>Bits per second</i>
IIC / I2C	<i>Inter-IC</i>
SCL	<i>Serial Clock Line</i>
SDA	<i>Serial Data Line</i>
USB	<i>Universal Serial Bus</i>
IDE	<i>Integrated Development Environment</i>
API	<i>Application Programming Interface</i>
RTOS	<i>Real-Time Operating System</i>
HAL	<i>Hardware abstraction layer</i>
CMSIS	<i>Common Microcontroller Software Interface Standard</i>

SPISAK SLIKA

SLIKA 2.1.1 PRINCIP RADA INDUKTIVNOG DETEKTORA POKRETA.....	4
SLIKA 2.2.1 PRINCIP RADA KAPACITIVNOG DETEKTORA POKRETA.....	5
SLIKA 2.3.1 PRINCIP RADA ULTRAZVUČNOG DETEKTORA POKRETA.....	6
SLIKA 2.4.1 PRINCIP RADA PIROELEKTRIČNOG DETEKTORA POKRETA	7
SLIKA 2.4.2 ILUSTRACIJA ČESTE REALIZACIJE PIROELEKTRIČNOG DETEKTORA POKRETA[11]	7
SLIKA 2.5.1. PRINCIP RADA IR DETEKTORA POKRETA	8
SLIKA 3.1 ARHITEKTURA SOFTVERA JEDNOG MIKROKONTROLERA	9
SLIKA 4.1 ŠEMA SISTEMA DETEKTORA POKRETA	11
SLIKA 4.1.1.1 ŠEMA RAZVOJNE PLOČICE MINI-M4 ZA TIVAC [6].....	12
SLIKA 4.1.2.1 ŠEMA SENZORA PROXIMITY 13 CLICK [6].....	13
SLIKA 4.1.2.2 ILUSTRACIJA UPISA I ČITANJA REGISTARA I PARAMETRA SENZORA.....	13
SLIKA 4.1.2.3 BLOK ŠEMA SENZORA SI1153-AB09-GMR	14
SLIKA 4.1.2.4 RASPORED SENZORA I DIODA NA PROXIMITY 13 CLICK-U [6]	14
SLIKA 4.1.2.5 TIPIČNA UGAONA OSETLJIVOST FOTODIODA (%).....	15
SLIKA 4.1.3.1 ŠEMA USB UART CLICK-A [6]	15
SLIKA 4.2.1 ARHITEKTURA SOFTVERA U SISTEMU	16
SLIKA 4.2.1.1 ILUSTRACIJA INTERFEJSA CMSIS-RTOS2.....	17
SLIKA 5.1 DIJAGRAM TOKA APLIKATIVNOG PROGRAMA.....	19
SLIKA 5.1.1 BLOK ŠEMA ORGANIZACIJE NITI	20
SLIKA 5.1.2 ILUSTRACIJA VREMENSKOG PRIKAZA IZVRŠAVANJA NITI.....	21
SLIKA 5.1.1.1 DIJAGRAM TOKA NITI ZA KOMUNIKACIJU	22
SLIKA 5.1.2.1 DIJAGRAM TOKA READ NITI.....	23
SLIKA 5.1.3.1 DIJAGRAM TOKA GESTURE NITI	24
SLIKA 5.2.1 ILUSTRACIJA L KONFIGURACIJE DETEKTORA POKRETA	25
SLIKA 6.1.1 IZMERENE VREDNOSTI TOKOM EKSPERIMENTA 1.....	27
SLIKA 6.2.1 IZMERENE VREDNOSTI TOKOM EKSPERIMENTA 2.....	28
SLIKA 6.3.1 IZMERENE VREDNOSTI TOKOM EKSPERIMENTA 3.....	29
SLIKA 6.4.1 IZMERENE VREDNOSTI TOKOM EKSPERIMENTA 4.....	30