Detekcija prolaska korištenjem mikroupravljačkog sustava

Ivić, Daniel

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:198058

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Detekcija prolaska korištenjem mikroupravljačkog sustava

Završni rad

Daniel Ivić

Osijek, 2024.

Sadržaj

1. Uvod 1

2. Primjene detekcije prolaska u praksi 2

3. Mikroupravljački sustavi u detekciji prolaska 3

3.1. Harvard arhitektura 4

3.2. Mikroupravljački sustav Arduino Uno 5

4. Senzori za detekciju prolaska 7

4.1. PIR senzori 8

4.2. Senzor infracrvenog zračenja 9

4.3. Senzor pokreta APDS9960 11

4.3.1. Detekcija gesti 12

4.3.2. Detekcija udaljenosti 12

4.3.3. Detekcija boje i razine osvjetljenja okoline 12

5. Razvoj i implementacija mikroupravljačkog sustava 13

6. Softverski kod implementacije 15

6.1. Funkcija „setup“ 16

6.2. Funkcija „loop“ 17

6.3. Softverski kod implementacije za sustav detekcije prolaska 17

7. Rezultati i analiza 21

8. Zaključak 24

9. Literatura 25

1. Uvod

U suvremenom dobu, brza evolucija tehnologije nastavlja preoblikovati naše svakodnevne živote, nudeći bezbroj elektroničkih uređaja koji olakšavaju različite zadatke s neviđenom lakoćom i učinkovitošću. Usred ovog tehnološkog krajolika, sustavi mikroupravljača pojavljuju se kao ključne komponente u realizaciji brojnih automatiziranih sustava, povećavajući učinkovitost, preciznost i pouzdanost različitih procesa. Među ovim primjenama, otkrivanje prolaska ističe se kao kritična funkcija sa širokim područjem primjene u industrijskim, sigurnosnim i nadzornim okruženjima.

Ovaj se rad fokusira na implementaciju detekcije prolaska korištenjem mikroupravljačkih sustava, s posebnim naglaskom na Arduino platformu. Detekcija prolaska ima iznimnu važnost u raznim sektorima, od brojanja proizvoda na montažnim trakama do praćenja kretanja vozila u prometu, praćenja ulaska i izlaska pojedinaca iz prostorija i brojnih drugih aplikacija.

Sustavi mikroupravljača služe kao kompaktne računalne jedinice koje integriraju procesore, memoriju, ulazno/izlazne portove i druge komponente na jednom čipu. Ova integracija omogućuje izvršavanje koda za kontrolu raznih elektroničkih komponenti, kao što su senzori, aktuatori i komunikacijska sučelja. Platforma Arduino ističe se kao jedan od najpopularnijih izbora za razvoj mikroupravljačkih sustava, poznata po svom korisničkom sučelju, živahnoj korisničkoj zajednici i opsežnoj podršci za široku skupinu senzora i modula.

Detekcija prolaska uključuje proces identificiranja promjena u stanju ili prisutnosti objekata unutar definiranog prostora ili zone. Ovaj se proces često oslanja na niz senzora koji otkrivaju fizičke, svjetlosne, magnetske ili akustične promjene u okolnom okruženju. Primjena detekcije prolaska obuhvaća široki spektar, od industrijskog praćenja i nadzora do upravljanja sigurnosnim sustavima u svakodnevnom životu.

1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi detektor prolaska korištenjem mikroupravljačkog sustava. Osim detekcije smjera prolaska, uređaj mora moći odrediti broj osoba u prostoriji te ga prikazati na zaslonu. Izrađeni uređaj vrednovati kroz laboratorijska mjerenja.

2. Primjene detekcije prolaska u praksi

Detekcija prolaska putem mikroupravljača je tehnologija koja se sve češće koristi u praktičnom svijetu pri čemu pruža efikasno i sigurno rješenje za brojne izazove u različitim područjima. Ova se tehnologija oslanja na mikroupravljače, male računalne sustave koji su sposobni za obradu signala u realnom vremenu i donošenje brzih odluka na osnovu ulaznih podataka.

Primjena detekcije prolaska u praksi je široka, a neka od područja u kojima se upotrjebljuje su :

- Sigurnost – u sigurnosnim sustavima detekcija prolaska omogućava praćenje kretanja ljudi ili predmeta kroz određeni prostor. Jedan primjer ovakve primjene bi bio sustav video nadzora gdje mikroupravljači mogu analizirati video snimke i detektirati neovlašteni prolazak određenim područjem. Još jedan primjer bi bila sigurnosna vrata ili kontrolne točke za identifikaciju osoba koje ulaze i izlaze iz određenog prostora.
- Industrija – koristi se u proizvodnim linijama kako bi se nadgledalo kretanje proizvoda kroz različite faze procesa proizvodnje. Ovo nam omogućuje praćenje protoka materijala i identifikaciju mogućih problema ili zastoja u proizvodnom procesu. Još jedan primjer primjene detekcije prolaska u industriji je primjena u logistici za praćenje kretanja robe kroz skladišta ili distribucijske centre.
- Automobilska industrija – koristi se u svrhu nadzora prometa na cestama. Sustavi za upravljanje prometom mogu koristiti ovu tehnologiju za praćenje kretanja vozila kroz raskrižja ili za detekciju prometnih prekršaja kao što su prolazak kroz crveno svjetlo ili prekoračenje brzine.
- Medicina – može se koristiti u medicinskim uređajima za praćenje kretanja pacijenata kroz bolnice ili za nadzor korištenja medicinske opreme. Primjer ovakve uporabe je praćenje kretanja kirurga i medicinskog osoblja kako bi se osigurala pravilna upotreba opreme i poštivanje sterilnih procedura.
- Pametni domovi – korištenje za praćenje aktivnosti poput otvaranja i zatvaranja vrata ili prozora te automatsko upravljanje osvjetljenjem ili klimatizacijom prema potrebi korisnika

Ova tehnologija se koristi radi velike brzine i preciznosti u svojem radu, ali također i zbog niskih cijena i jednostavnosti sklopova koji se koriste.

3. Mikroupravljački sustavi u detekciji prolaska

Postoje različiti mikroupravljački sustavi koji se mogu koristiti za detekciju prolaska, ali radi jednostavnosti i pristupačnosti u ovome završnom radu fokus će biti na mikroupravljačkim sustavima baziranim na Arduinu.

Mikroupravljač je procesor opremljen memorijom, tajmerima, I/O pinovima i drugim perifernim uređajima na čipu. Glavni razlog raširenog korištenja mikroupravljača je trošak : integracija svih elemenata na jednom čipu štedi prostor, dovodi do nižih proizvodnih troškova i kraćeg vremenskog razvoja. To štedi vrijeme i novac što su ključni faktori u ugrađenim sustavima. Dodatne prednosti integracije su laka nadogradnja, manja potrošnja energije i veća pouzdanost, što su također vrlo važni aspekti u ugrađenim sustavima. Međutim, korištenje mikroupravljača za rješavanje zadataka softverskim putem koji se može riješiti i hardverskim rješenjem neće pružiti istu brzinu koju bi hardversko rješenje moglo postići. Stoga aplikacije koje zahtijevaju vrlo kratke vremenske reakcije još uvijek mogu zahtijevati hardversko rješenje. Većina aplikacija, a posebno one koje zahtijevaju neku vrstu ljudske interakcije kao što je detekcija prolaska, ne zahtijevaju takve brze vremenske reakcije pa su za takve aplikacije mikroupravljači dobar izbor.

Postoje različite arhitekture mikroupravljačkih sustava koje se mogu koristiti za primjenu detekcije prolaska :

- Von Neumann arhitektura – program i podatci pohranjeni su zajedno i pristupa im se preko iste sabirnice. Nažalost, to dovodi do toga da pristupi programu i podatcima mogu dovesti do sukoba (ovo rezultira poznatom von Neumannovom uskom grlu), što uzrokuje nepoželjna kašnjenja. Prednosti ove arhitekture su jednostavnost pri programiranju i razumijevanju što omogućuje fleksibilnost u izradi programa, a ograničenje je to što je podložna ograničenju brzine izvođenja radi toga što se podatci i naredbe dijele istim putem.
- Harvard arhitektura - program i podaci su u odvojenim memorijama koje se pristupaju putem odvojenih sabirnica. Kao posljedica toga, pristupi kodu ne sukobljavaju se s pristupima podacima, što poboljšava performanse sustava. Kao blagi nedostatak, ova arhitektura zahtijeva više hardvera, jer treba dvije sabirnice i ili dva memorijska čipa ili dvostruko pristupačnu memoriju (memorijski čip koji omogućuje dva neovisna pristupa istovremeno). Odvojeni putovi za podatke i naredbe omogućuju paralelno izvršavanje, što može poboljšati brzinu izvođenja. Osnovni nedostatak ove arhitekture je što je kompleksnija za programiranje u usporedbi s Von Neumann arhitekturom zbog odvojenih memorija za podatke i naredbe.

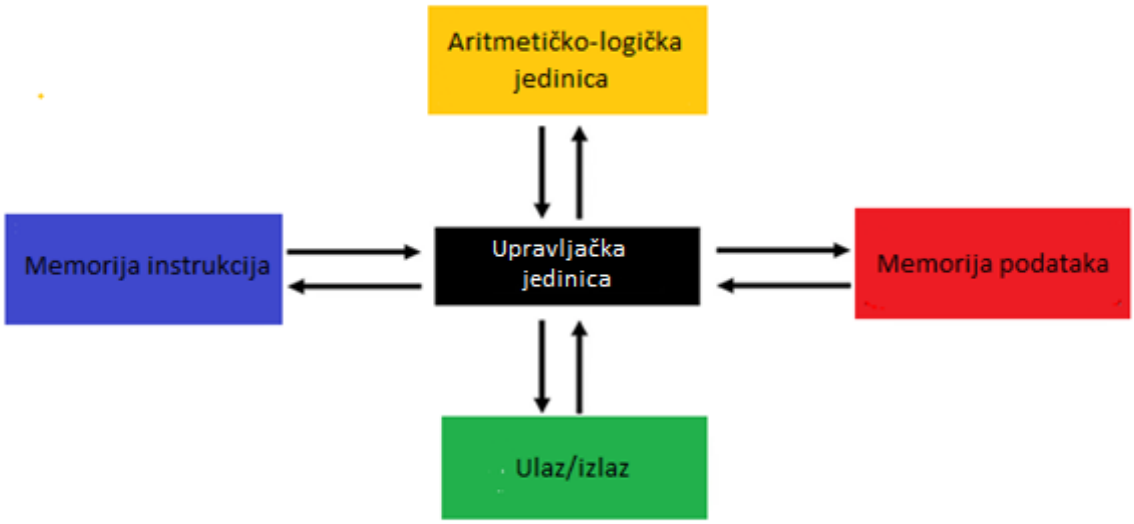
- RISC (eng. Reduced Instruction Set Computer) arhitektura - ima jednostavne, hardverske upute koje često zahtijevaju samo jedan ili nekoliko taktnih ciklusa za izvršenje. RISC strojevi imaju malu i fiksnu veličinu koda s relativno malo uputa i malo načina adresiranja. Kao rezultat, izvršenje uputa je vrlo brzo, ali je skup uputa prilično jednostavan. Prednosti su jednostavne i brze instrukcije, što omogućuje brže izvršavanje programa i smanjuje složenost mikroupravljačkog sustava. Nedostatak je taj što je potrebno više instrukcija za izvršavanje složenih operacija što može povećati veličinu programa.
- CISC (eng. Complex Instruction Set Computer) arhitektura - karakterizira se složenim mikroprogramiranim uputama koje za izvršenje zahtijevaju mnogo taktnih ciklusa. Arhitektura često ima veliku i varijabilnu veličinu koda te nudi mnogo moćnih uputa i načina adresiranja. U usporedbi s RISC-om, CISC duže izvršava svoje upute, ali je skup uputa moćniji. Prednosti su što može obaviti složene operacije u jednoj instrukciji, što može smanjiti i veličinu programa. Najveće ograničenje ove arhitekture je što složene instrukcije mogu dovesti do sporijeg izvršavanja programa i povećane složenosti programa.

U kontekstu detekcije prolaska, arhitektura mikroupravljačkog sustava treba biti odabrana uzimajući u obzir zahtjeve za brzinom, pouzdanošću i resursima. Na primjer, za brzu detekciju prolaska može biti korisna RISC arhitektura zbog bržih instrukcija izvođenja, dok se za aplikacije koje zahtijevaju manje kompleksne operacije može odabrati CISC arhitektura radi smanjenja veličine programa. Važno je također uzeti u obzir i druge faktore poput potrošnje energije i troškova implementacije. Radi prethodno navedenih razloga te jednostavnosti i pristupačnosti za ovaj završni rad odabran je Arduino Uno koji koristi AVR mikroupravljač temeljen na Harvard arhitekturi. Konkretno, koristi mikroupravljač ATMega328P, koji je dio AVR obitelji mikroupravljača proizvedene od strane tvrtke Atmel (sada dio Microchip Technology). Ova arhitektura omogućuje razdvajanje putova za podatke i naredbe, što olakšava programiranje i omogućuje brže izvršavanje programa.[1]

3.1. Harvard arhitektura

Harvard arhitektura je računalna arhitektura s odvojenim pohranama i signalnim putanjama za upute i podatke. Često se uspoređuje s von Neumannovom arhitekturom, gdje programske naredbe i podatci dijele istu memoriju i putanje.

Izraz se često navodi kao da potječe od računala na relejima Harvard Mark I, koje je pohranjivalo upute na probušenu traku (širine 24 bita) i podatke u elektro-mehaničke brojače. Ovi rani strojevi imali su pohranu podataka potpuno sadržanu unutar središnje procesorske jedinice i nisu omogućavali pristup pohrani uputa kao podataka. Programe je morao učitati operator; procesor se nije mogao sam inicijalizirati. Međutim, izraz "Harvardska arhitektura" skovan je desetljećima kasnije, u kontekstu dizajna mikroupravljača, te je retroaktivno primijenjen na Harvardske strojeve i kasnije na RISC mikroprocesore s odvojenim predmemorijama. Suvremeni procesori korisnicima se čine kao sustavi s von Neumannovom arhitekturom, s programskim kodom pohranjenim u istoj glavnoj memoriji kao i podaci. Izvedbeni razlozi, interni i uglavnom nevidljivi korisniku, većinom imaju odvojene predmemorije procesora za upute i podatke, s odvojenim putanjama u procesor za svaku. To je jedan oblik onoga što se naziva modificiranom Harvard arhitekturom.[2]



Sl. 3.1. Harvard arhitektura [3]

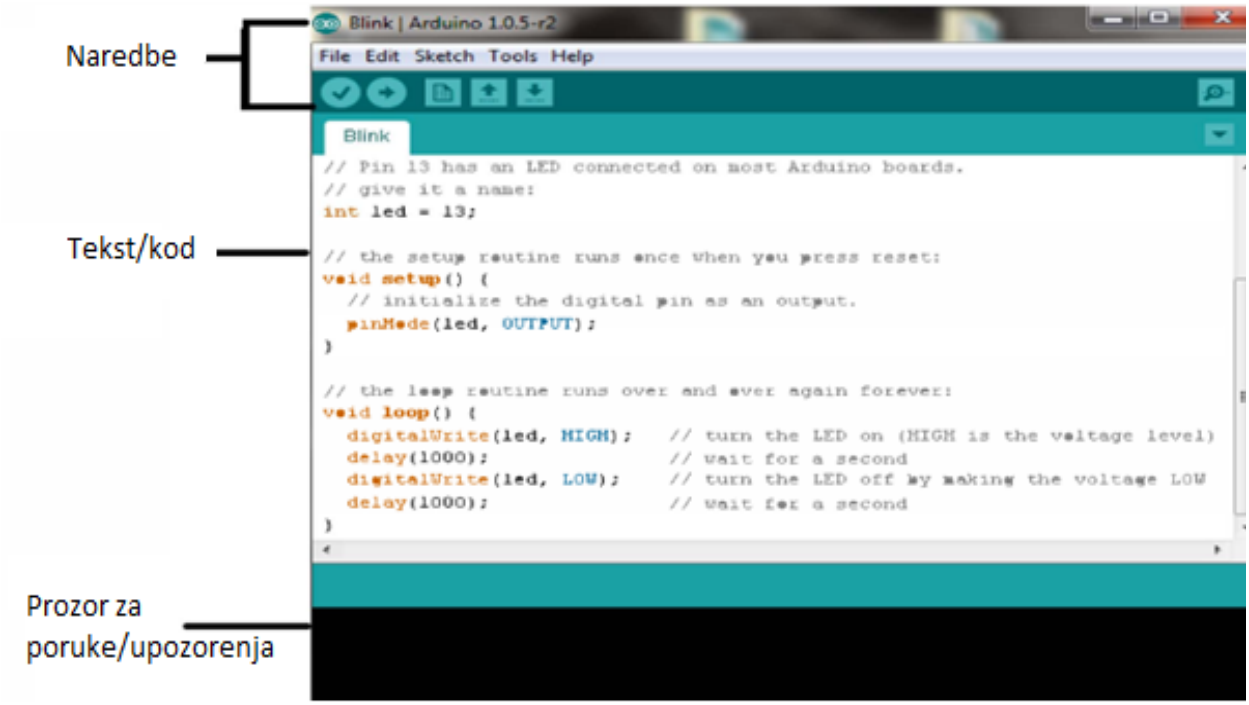
3.2. Mikroupravljački sustav Arduino Uno

Arduino je *Open-Source* platforma koja se koristi za izradu i programiranje elektronike. Može primati i slati informacije većini uređaja, čak i preko interneta za upravljanje određenim elektroničkim uređajem. Koristi hardver nazvan Arduino Uno i softverski program (koristi jednostavni C++ programski jezik) za programiranje ploče. U današnje vrijeme, Arduino se često koristi u programiranju mikroupravljača i stvaranju interaktivnih elektroničkih objekata zbog svojih korisničkih postavki koje su jednostavne za korištenje. Arduino je ploča s čipom koja se,

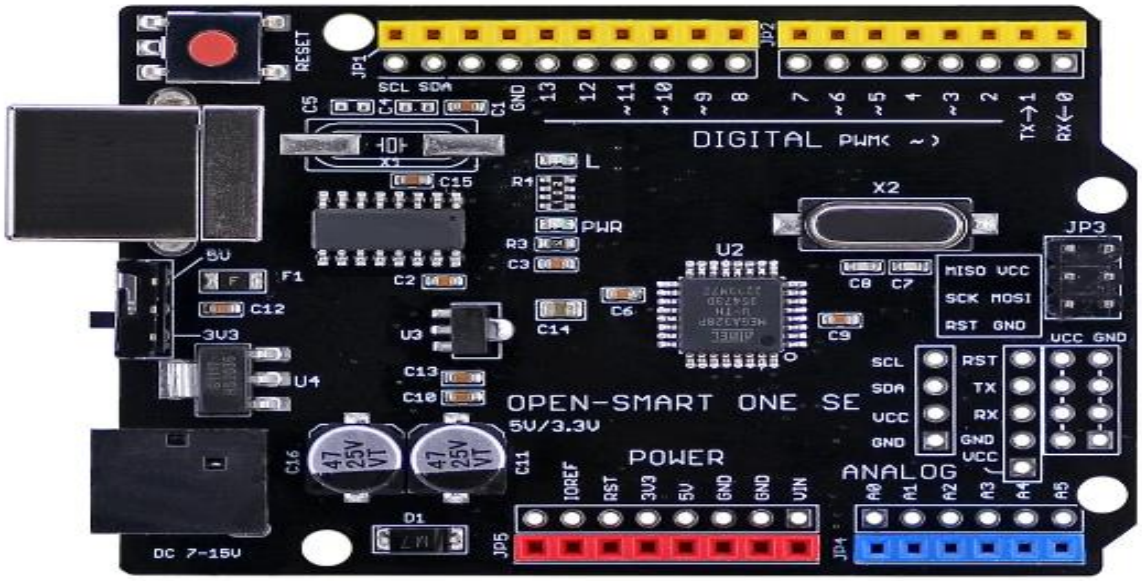
kao i bilo koji mikroupravljač, može programirati za obavljanje različitih zadataka. Šalje informacije iz računalnog programa mikroupravljaču Arduina, a zatim na određenu ploču s više krugova kako bi izvršio određenu naredbu. Arduino može pomoći u čitanju informacija s ulaznih uređaja kao što su senzori, antene, potenciometar itd., i također može slati informacije izlaznim uređajima kao što su LED, zvučnici, LCD zaslon, DC motor itd.

Arduino platforma postala je dobro poznata ljudima koji se bave elektronikom. Za razliku od većine prethodnih programibilnih sklopovskih ploča, Arduino nema zaseban hardverski dio za učitavanje novog koda na ploču; jednostavno se može koristiti USB kabel za učitavanje, a softver Arduina koristi pojednostavljenu verziju C ++, što olakšava učenje programiranja i pruža jednostavnije okruženje koje zaobilazi funkcije mikroupravljača u pristupačniji paket.

Mikroupravljački sustav Arduino može se podijeliti na 2 temeljna dijela : hardver i softver. U hardver spadaju sve komponente pločice kao što su USB priključak, pinovi (analogni, digitalni, za napajanje, …),LED diode itd. Pod softver se podrazumijeva Arduino IDE (eng. Integrated Development Environment). Izgled softvera za Arduino prikazan je na slici 3.2. Primjer Arduino pločice koja je korištena u ovome završnom radu prikazan je na slici 3.3. [4]



Sl. 3.2. Arduino IDE s naznačenim dijelovima programa



Sl. 3.3. Razvojna pločica OPEN-SMART R3 temeljena na Arduino Uno

Ova razvojna pločica temeljena je na Arduino Uno razvojnom sustavu te ATMEGA328P mikroupravljaču. Postoji prekidač za odabir napona napajanja sustava, 3.3V ili 5V, što je vrlo korisno ako se želi postaviti sustav na 3.3V kako bi se uštedjela energija. Također može biti korisno i za komponente koje zahtijevaju rad na 3.3V kao što je senzor pokreta koji je korišten u ovome završnom radu. Pločica sadrži : programibilnu LED diodu za testiranje, čip upravljačkog sučelja USB-a CH340G, kompatibilan s Windowsom 7, Windowsom 8, Linuxom i MAC OS-om, ATMEGA328 mikroupravljač, 14 digitalnih I/O (input / output) pinova (od čega 6 njih sadržava PWM izvor), 32 KB *flash* memorije od čega je 2 KB iskorišteno za *bootloader*, 2 KB SRAM-a, 1KB EEPROM-a te brzinu takta od 16 MHz.

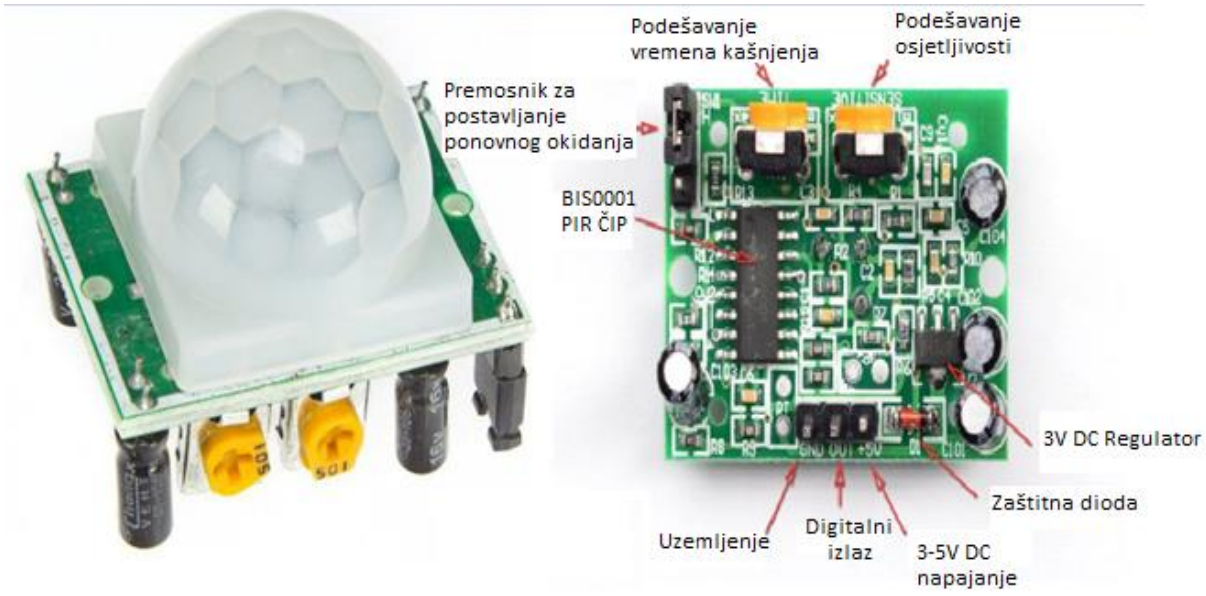
4. Senzori za detekciju prolaska

Detekcija prolaska bazirana je na detekciji pokreta. Detektiranje pokreta u određenom smjeru odgovarajućim senzorima i softverskim kodom rezultira mikroupravljačkim sustavom za detekciju prolaska. Postoje razne vrste senzora koji imaju mogućnost detekcije pokreta, pa samim time postoje i razni načini izrade mikroupravljačkog sustava za detekciju prolaska. Neke vrste senzora koje se mogu koristit za detekciju prolaska i koje su testirane pri izradi ovoga završnog su PIR (eng. *Passive infrared*) senzor, IR infracrveni senzor te APDS9960 senzor pokreta. Radi jednostavnosti, preciznosti i pristupačnosti u ovome završnom radu konačan odabir bio je APDS9960.

4.1. PIR senzori

Pasivni infracrveni (PIR) senzori koriste se u mnogim područjima, uključujući sigurnost. PIR senzori detektiraju prisutnost ljudi i životinja na temelju zračenja toplinske energije njihovog tijela. To se može koristiti za pokretanje događaja, npr. otvaranje vrata, snimanje videa, itd. PIR senzori se široko koriste zbog svoje niske potrošnje energije.

Glavna komponenta PIR senzora je piroelektrični senzor koji se nalazi iza plastičnog poklopca. PIR senzor zapravo je podijeljen na dva dijela. Kada nema pokreta, oba dijela primaju istu količinu IR zračenja iz okoline. Međutim, kada se meta približi senzoru, razina IR zračenja primljenog na jednom dijelu veća je od zračenja na drugom dijelu. PIR reagira na tu promjenu i izlaz postavlja na VISOKO. Raspon detekcije doseže do 7 metara, a potrebno je neko vrijeme da se stabilizira prema okolini kada je uključen. PIR senzor, kao i njegovi dijelovi, testiran u ovome radu prikazan je na slici 4.1.



Sl. 4.1. PIR senzor sa naznačenim dijelovima

Uz piroelektrični senzor nalazi se skup potpornih sklopova, otpornika i kondenzatora. Većina malih senzora za hobiste koristi BISS0001 ("Micro Power PIR Motion Detector IC") nedvojbeno vrlo jeftin čip. Ovaj čip uzima izlaz senzora i provodi neku manju obradu kako bi iz analognog senzora emitirao digitalni izlazni impuls.

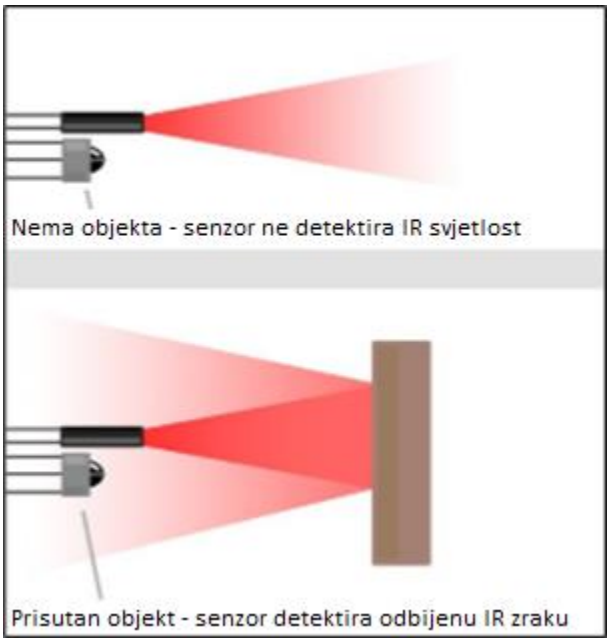
Za mnoge jednostavne projekte ili proizvode koji trebaju detektirati kada je osoba izašla ili ušla u područje ili se približila, PIR senzori su odlični. Oni su niskog napona i niskih troškova, prilično izdržljivi, imaju širok raspon leće i lako se povezuju. Važno je napomenuti da PIR senzori neće

reći koliko ljudi se nalazi u blizini ili koliko su blizu senzora, leća je često fiksirana na određeni domet i udaljenost (iako se može modificirati izradom leća drugih oblika) i ponekad ih aktiviraju kućni ljubimci.

Iz navedenih osobina senzora dolazi se do zaključka da nije najbolji za izradu sustava za detekciju prolaska prvobitno radi velikog broja problema koji nastaju. Prvi problem je što bi za detekciju prolaska u oba smjera bilo potrebno koristiti 2 senzora, jedan postavljen na ulaznoj strani dok je drugi na izlaznoj strani. Drugi problem bi bio udaljenost na koju se moraju postaviti senzori kako ne bi međusobno smetali jedan drugom i vraćali krivu povratnu informaciju sustavu. Iz podataka o senzoru, koje proizvođač daje na raspolaganje, vidljivo je da u horizontalnoj ravnini senzor ima kut detekcije od 110° dok u vertikalnoj ima kut detekcije 70° pri čemu minimalna udaljenost na koju može detektirati nije definirana. Radi ovih problema dolazi se do zaključka da PIR senzor, iako je jeftin i precizan, nije idealan za izradu sustava za detekciju prolaska.[5]

4.2. Senzor infracrvenog zračenja

Infracrveni senzori detektiraju udaljenost objekta pomoću infracrvenog zračenja. Kada zraka detektira objekt, svjetlosna zraka se reflektira i vraća se na prijemnik pod određenim kutom. Metoda triangulacije prikazana je na slici 4.2. [6]



Sl. 4.2. Princip rada IR senzora

Modul senzora prilagodljiv je ambijentalnom svjetlu i sadrži par LED dioda za emitiranje i primanje infracrvenog svjetla. Emitirajuće cijevi šalju infracrveno svjetlo određene frekvencije, a

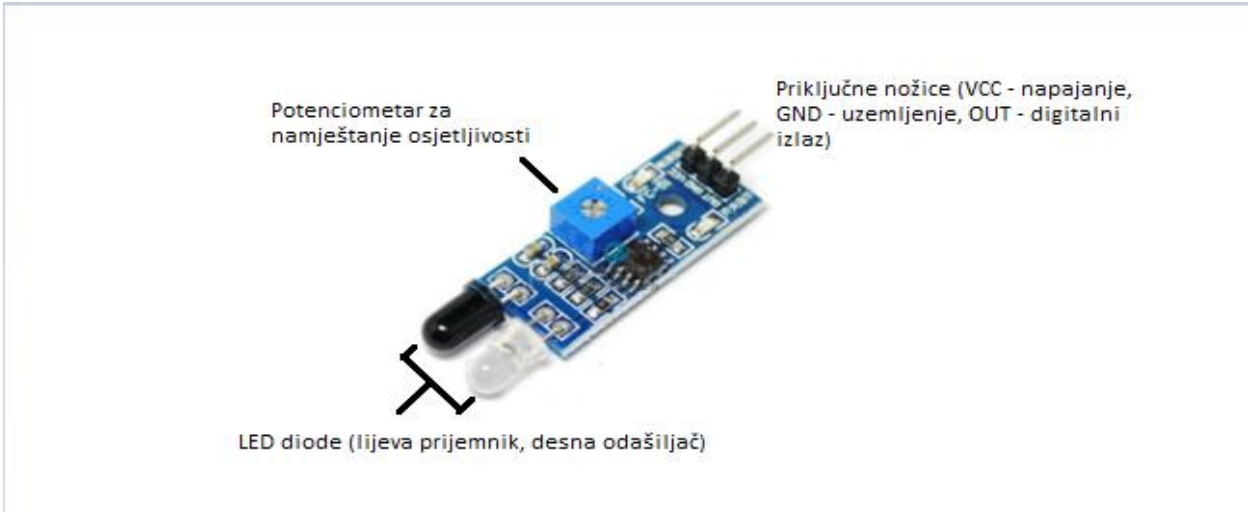
kada se detektira smjer prepreke (reflektirajuća površina), infracrveno svjetlo koje se reflektira prima se putem cijevi za primanje. Nakon obrade pomoću komparatora, zelena svjetlost se upali te izlazno sučelje signala vraća digitalni signal (niske razine). Može se podesiti potenciometar za regulaciju udaljenosti detekcije, s efektivnim rasponom udaljenosti od 2 do 30 cm, i radnim naponom od 3.3V do 5V. Raspon detekcije senzora može se dobiti podešavanjem potenciometra, s malo smetnji, lako se sastavlja, ima jednostavne značajke korištenja, te se može široko koristiti u izbjegavanju prepreka robota, automobila, brojanju linija i praćenju crno-bijelih linija te mnogim drugim primjenama.[6]

Korištenje IR senzora pri izradi sustava za detekciju prolaska može biti dobar odabir, međutim potrebno je razmotriti nekoliko mana koje ti senzori imaju te prepreke koje mogu nastati pri implementaciji :

- Ograničena preciznost: IR senzori ponekad možda neće uvijek precizno detektirati posjetitelje, posebno ako postoje okolišni čimbenici poput prašine, dima ili drugih čestica koje mogu ometati očitanja senzora. To može rezultirati pogrešnim brojanjem ili netočnim podacima.
- Smetnje: IR senzori mogu biti podložni utjecaju okolnih izvora svjetlosti, uključujući sunčevu svjetlost i umjetno osvjetljenje. Ta smetnja može uzrokovati lažna očitanja i utjecati na točnost brojanja posjetitelja.
- Održavanje: IR senzori mogu zahtijevati redovito održavanje kako bi osigurali ispravno funkcioniranje. To može uključivati čišćenje radi uklanjanja prašine i novu kalibraciju kako bi se održala točnost tijekom vremena.
- Ograničen domet: Domet IR senzora može biti ograničen, pogotovo ako nisu visokih performansi ili ako postoje prepreke između senzora i posjetitelja. To bi moglo rezultirati propuštenim brojevima ako posjetitelji nisu unutar dosega senzora.
- Vrijeme odziva: IR senzori mogu imati kašnjenje u detektiranju posjetitelja, pogotovo ako postoji visoka razina prometa i veliki broj prepreka. To kašnjenje može utjecati na stvarnu točnost broja posjetitelja u stvarnom vremenu.

Osim ovih čimbenika, kao i kod PIR senzora postoji i potreba za korištenjem minimalno 2 ovakva senzora. Ponovno nastaje problem međusobne interferencije oba senzora i potrebe za vođenjem računa o razmaku između senzora. Unatoč svemu tome , IR senzor bi bio bolji izbor za rješavanje problema detekcije prolaska od PIR senzora iz razloga što mu je kut detekcije (prema podatcima

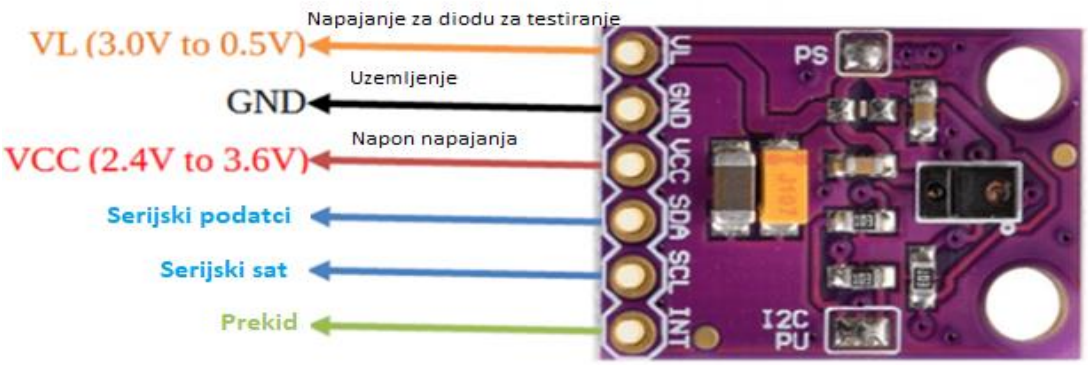
od proizvođača) samo 35° što znači da udaljenost između senzora ne mora biti prevelika. Modul IR senzora prikazan je na slici 4.3.



Sl. 4.3. Modul IR senzora

4.3. Senzor pokreta APDS9960

Za detekciju prolaska može se koristiti i senzor pokreta, konkretno SparkFun RGB i senzor pokreta, mali breakout modul s ugrađenim APDS-9960 senzorom koji nudi mjerenje ambijentalne svjetlosti i boje, detekciju blizine te senzor za beskontaktno prepoznavanje pokreta. S ovim RGB i Gesture senzorom može se kontrolirati računalo, mikroupravljač, robot i još mnogo toga jednostavnim pokretom ruke. APDS-9960 je ozbiljan mali komad hardvera s ugrađenim UV i IR filtrom, četiri odvojene diode osjetljive na različite smjerove te sučelje kompatibilno s I2C protokolom. Radi praktičnosti, izdvojeni su sljedeći pinovi: VL (opcionalno napajanje za IR LED), GND (uzemljenje), VCC (napajanje za APDS-9960 senzor), SDA (I2C podatci), SCL (I2C sat), i INT (prekid). Svaki APDS-9960 također ima raspon detekcije od 4 do 8 inča (10 do 20 cm) što je



Sl. 4.4. Modul senzora pokreta

i više nego dovoljno za detekciju prolaska kod jednokrilnih vrata. Na slici 4.4. prikazan je modul senzora.

4.3.1. Detekcija gesti

Detekcija gesti koristi četiri usmjerene foto-diode za detektiranje reflektirane IR energije (koju generira integrirana LED dioda) kako bi pretvorila informacije o fizičkom kretanju (npr. brzina, smjer i udaljenost) u digitalne informacije. Arhitektura sustava za geste sadrži automatsku aktivaciju (na temelju rezultata sustava za blizinu), oduzimanje ambijentalne svjetlosti, poništavanje međusobnih smetnji, dvostruke 8-bitne pretvarače podataka, kašnjenje međupretvorbe za uštedu energije, FIFO spremnik od 32 skupa podataka te I2C komunikaciju koja se pokreće prekidima. Gestacijski sustav zadovoljava različite zahtjeve mobilnih uređaja u pogledu gesta: jednostavne geste GORE-DOLJE-LIJEVO-DESNO ili kompleksnije geste mogu se precizno detektirati. Potrošnja energije i buka smanjuju se pomoću podesivog vremena IR LED-a.

Putem detekcije gesti moguće je detektirati prolazak te jedina značajka o kojoj ovisi način detekcije je orijentacija samog senzora. Dodatno je moguće softverski mijenjati način detekcije, osjetljivost senzora, ulazne i izlazne pragove gesti i sl. Za način rada senzora za detekciju gesti potrebno je napajanje od 3.3 V što je postignuto korištenjem razvojne pločice za ugrađenom sklopkom za promjenu napona navedenu u prethodnom poglavlju o Arduino Uno mikroupravljaču.

4.3.2. Detekcija udaljenosti

Mogućnost detekcije udaljenosti omogućuje mjerenje udaljenosti (npr. zaslona mobilnog uređaja do korisnikovog uha) putem detekcije reflektirane IR energije (koju generira integrirani LED) foto-diodom. Detekcija objekta vrši se prekidima i događa se svaki put kada rezultat udaljenosti prijeđe gornje i/ili donje postavke praga. Sustav za detekciju udaljenosti sadrži registre za prilagodbu odstupanja kako bi kompenzirao sustavno odstupanje uzrokovano neželjenim refleksijama IR energije koje se pojavljuju na senzoru. Intenzitet IR LED-a tvornički je podešen kako bi se eliminirala potreba za kalibracijom krajnje opreme zbog varijacija komponenti. Rezultati udaljenosti dodatno se poboljšavaju automatskim oduzimanjem ambijentalne svjetlosti.

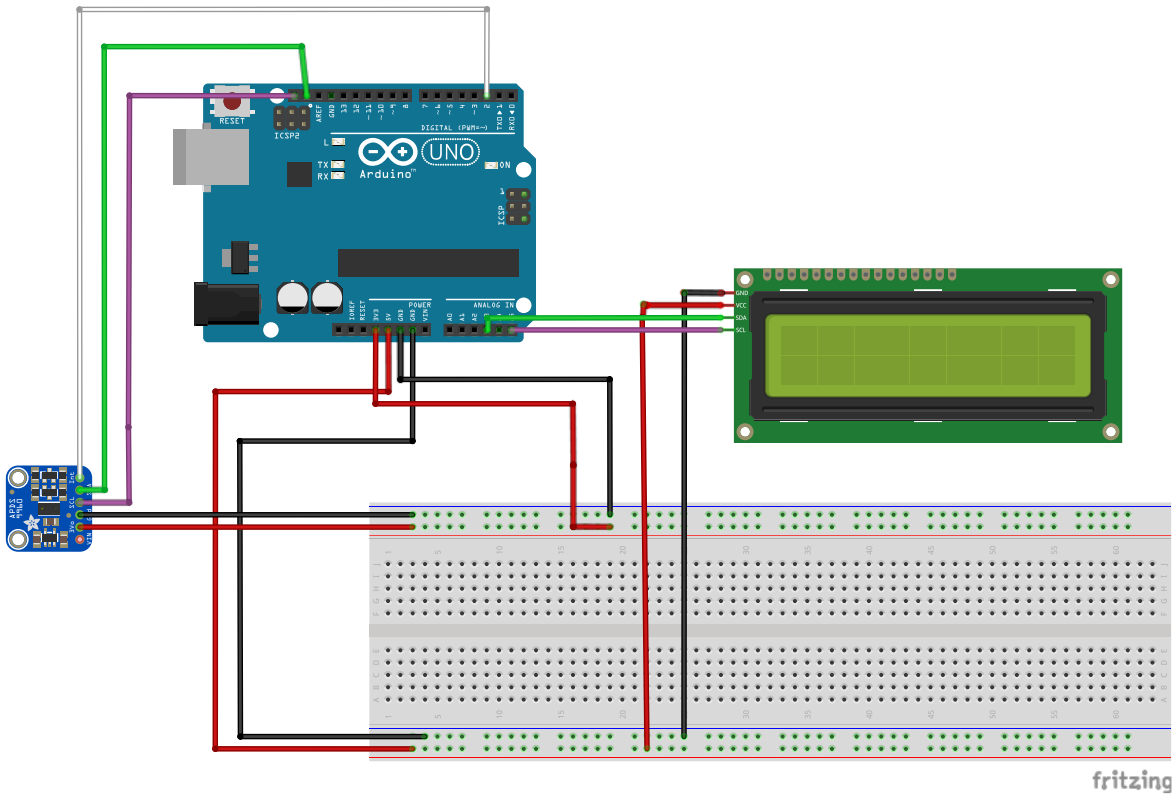
4.3.3. Detekcija boje i razine osvjetljenja okoline

Mogućnost detekcije boje i razine osvjetljenja okoline pruža podatke o intenzitetu crvene, zelene, plave i bijele svjetlosti. Svaki od kanala (R, G, B, C) ima UV i IR blokirajući filter te namjenski pretvornik podataka koji istovremeno proizvodi 16-bitne podatke. Ova arhitektura omogućuje

aplikacijama precizno mjerenje ambijentalne svjetlosti i osjet boje, što uređajima omogućuje izračun temperature boje i kontrolu pozadinskog osvjetljenja zaslona.

5. Razvoj i implementacija mikroupravljačkog sustava

Zadatak završnog rada je izraditi sustav za detekciju prolaska u svrhu brojanja posjetitelja, odnosno kako bi znali koliko se ljudi trenutno nalazi u prostoriji. Sustav je izrađen za korištenje kod prostorija sa jednokrilnim vratima radi maksimalne aktivacije senzora od 20 cm. Predviđeno je da sustav bude što kompaktniji, tj. da je korišteno što manje komponenti. To je još jedan od razloga zašto je odabran APDS9960 senzor jer na taj način je potreban samo jedan senzor te on omogućuje brojanje prolaznika u oba smjera koja može razlikovati. Komponente koje su korištene za izradu sustava su : OpenSmart One R3 razvojna pločica, breadboard, 16x2 LCD sa pretvaračem za I2C komunikaciju te senzor SparkFun APDS9960. Shema sustava prikazana je na slici 5.1.



Sl. 5.1. Shema sustava za detekciju pokreta

Na slici je vidljiva sama jednostavnost sklopa, te uz korištenje manjeg breadboarda moguće je ostvariti kompaktan sustav koji je jednostavan za korištenje. Na shemi je korišten Arduino Uno na kojemu je bazirana razvojna pločica korištena u ovome radu radi nedostatka pločice u biblioteci programa za izradu shema. Za izradu sheme korišten je Fritzing. Radi mogućnosti smanjenja

napajanja svih pinova sa 5V na 3.3V, koju ima razvojna pločica, nije potrebno koristiti dodatne sklopove i komponente za smanjenje razine napona napajanja senzora. Pločica se napaja putem USB-a naponom od 5V. LCD zaslon također ima ugrađen pretvarač za I2C komunikaciju tako da nije bilo potrebno koristiti dodatne sklopove za pretvaranje.

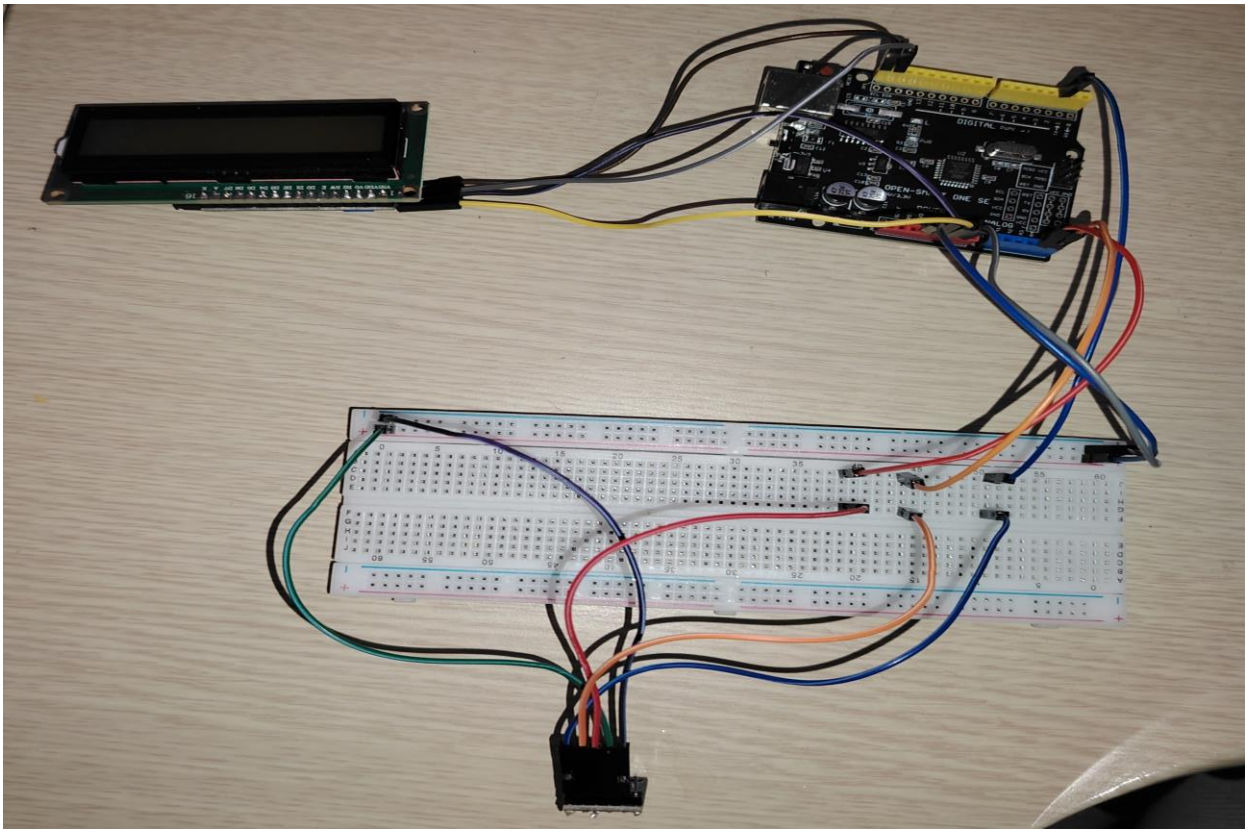
Način spajanja te pinovi koji su korišteni je sljedeći :

- APDS9960 – priključnica INT (interrupt) spojena je na digitalni pin D2 te služi za ostvarivanje prekida, priključnica VCC spojena je na napon napajanja od 3.3V, priključnica GND spojena je na istoimeno uzemljenje pločice, priključnice SDA i SCL spojene su na istoimene priključnice razvojne pločice te one služe za I2C komunikaciju preko koje senzor radi, priključnica VIN nije priključena jer ona služi samo za testiranje senzora putem LED diode.
- LCD – displej ima mogućnost prikazivanja 16 simbola po redu te može prikazati 2 reda, tj. 32 simbola sveukupno, display ima 16 priključnica koje su zalemljene na adapter za I2C komunikaciju te on smanjuje broj priključnica na 4. Priključnica VCC spojena je na napajanje od 5V, međutim, radi korištenja senzora koji radi na 3.3V, cijela razvojna pločica radi na 3.3V što ne predstavlja problem u korištenju displeja radi toga što je isti predviđen za rad od 3-5V. Priključnica GND spojena je na istoimeno uzemljenje razvojne pločice. Priključnice SDA i SCL za I2C komunikaciju spojene su na pinove A4 i A5 razvojne pločice koji imaju mogućnost funkcionalnosti I2C komunikacije. Pri tome A4 predstavlja SDA, a A5 predstavlja SCL.

I2C komunikacija koristi poseban protokol koji joj je dodijeljen, a to je I2C protokol. I2C sabirnica fizički se sastoji od dvije aktivne žice i uzemljenja. Te dvije aktivne žice nazivaju se Serijski sat (SCL) i Serijski podaci (SDA). Ove žice su koriste tehnologiju polu dvosmjernog prijenosa koje prenose informacije između uređaja spojenih na sabirnicu. Svaki uređaj prepoznaje se prema jedinstvenoj adresi, bilo da je to mikroupravljač, upravljač LCD zaslona, memorija ili sučelje tipkovnice, i može djelovati kao predajnik ili prijamnik, ovisno o funkciji uređaja. Na I2C sabirnici uređaji se mogu lako dodavati ili uklanjati, što je vrlo korisno za primjene s niskim održavanjem i kontrolu u ugradbenim sustavima.[7]

Za mogućnost korištenja LCD-a i APDS9960 senzora, prije samoga testiranja funkcionalnosti bilo je potrebno softverskim putem provjeriti ili odrediti I2C adresu tih komponenti. Provjera se vršila putem softverskog koda za pronalazak I2C adresa, koji će detaljnije biti opisan u idućem poglavlju, a pronađene adrese su 0x27 (za LCD displej) te 0x39 (za APDS9960).

Fizička izvedba sustava je vrlo kompaktna i prati shemu iz slike 5.1. te je prikazana na slici 5.2.



Sl. 5.2. Fizička izvedba sustava

6. Softverski kod implementacije

Softverski kod korišten za implementaciju ovoga sustava pisan je u Arduino IDE (eng. *Integrated Development Environment*) razvojnom okruženju posebnim jezikom kojeg koristi Arduino.

Programski kod napisan za Arduino definiran je kao skica. U ovome radu korišten je Arduino IDE 2.3.2. Ovaj IDE sadrži sljedeće dijelove :

- Tekstni urednik : dio sučelja koji služi za pisanje pojednostavljenog koda koristeći pojednostavljenu verziju C++ programskog jezika
- Područje za poruke : prikazuje pogreške i daje povratne informacije o spremanju i izvozu koda
- Tekst : konzola koja prikazuje tekstualni izlaz iz Arduino okruženja, uključujući potpune poruke o pogreškama i druge informacije

- Alatna traka konzole : sadrži razne gumbe kao što su Verify (Provjeri), Upload (Prenesi), New (Novi), Open (Otvori), Save (Spremi) i Serial Monitor (Serijski monitor). U donjem desnom kutu prozora prikazani su razvojna ploča i serijski port koji se koriste.

Datoteka projekta ili skice za projekt spremaju se s ekstenzijom datoteke .ino. Značajke kao što su izrezivanje, kopiranje i lijepljenje podržane su u ovom IDE-u. Također postoji mogućnost pronalaženja određene riječi i zamjene iste drugom pritiskom na tipke Ctrl + F na tipkovnici. Najosnovniji dio ili kostur svih Arduino kodova ima dvije funkcije : void setup() i void loop().

6.1. Funkcija „setup“

Ovo je prva rutina koja počinje kada se Arduino spoji na napajanje. Ova funkcija se izvršava samo jednom tijekom cijelog rada programa. Funkcija setup sadrži inicijalizaciju svakog pina koji namjeravamo koristiti u našem projektu za ulaz ili izlaz. Način na koji bi to trebalo biti napisano je sljedeći :

Linija *Kod*

```
1:      void setup()
2:      {
3:       pinMode(pin, INPUT);
4:       pinMode(pin, OUTPUT);
5:      }
```

Ovdje je pin broj pina koji treba definirati. INPUT / OUTPUT odgovaraju načinu na koji će pin biti korišten.

Također sadrži inicijalizaciju serijskog monitora. Serijski monitor se koristi za praćenje podataka koji se serijski šalju na bilo koji periferni uređaj. Primjer inicijalizacije je sljedeći :

Linija *Kod*

```
1:      void setup()
2:      {
3:       Serial.begin(9600);
4:      }
```

Prije korištenja bilo koje varijable za programiranje, potrebno ih je definirati iznad funkcije "void setup()".

6.2. Funkcija „loop“

Ova funkcija je sljedeća važna funkcija u skici. Sastoji se od onog dijela koda koji treba kontinuirano izvršavati, za razliku od dijela koda napisanog u funkciji setup. Primjer void loop funkcije je sljedeći:

Linija *Kod*

```
1:      void loop()
2:      {
3:       digitalWrite(pin, HIGH);
4:      }
```

Ovdje je digitalWrite funkcija koja zapisuje visoku ili nisku vrijednost na digitalni pin. Ako je pin konfiguriran kao OUTPUT s pinMode(), njegov napon će biti postavljen na odgovarajuću vrijednost: 5V (ili 3.3V na pločama s 3.3V) za HIGH, 0V (uzemljenje) za LOW.

Slično tome, ako postoji potreba za odgodom u skici, postoji još jedna funkcija (*delay*) koja stvara odgodu u izvršavanju koda. Npr. ukoliko želimo odgoditi događaj za 1 sekundu pišemo delay(1000) jer funkcija delay prima vremensku vrijednost u milisekundama.[8]

6.3. Softverski kod implementacije za sustav detekcije prolaska

Prije samog pisanja koda potrebno je definirati što će sve biti potrebno za pisanje i implementaciju koda. Softverski kod implementacije koristi 3 dodatne biblioteke za rad komponenti. Te 3 biblioteke su : LiquidCrystal_I2C.h (za korištenje LCD-a), Wire.h (za korištenje I2C uređaja te za pronalaženje I2C adresa uređaja) te SparkFun_APDS9960.h (za korištenje senzora pokreta).

Na slici 6.1. prikazan je softverski kod za pronalaženje I2C adresa uređaja. Ovaj kod koristi biblioteku Wire.h. Programski kod za ovaj program dostupan je u samom Arduino razvojnom okruženju.

Program na temelju povratne vrijednosti naredbe WIRE.endTransmission provjerava postoji li uređaj koji je pronašao tu adresu, odnosno odgovorio na nju. Provjerava raspon adresa od 0x1 do 0x127. U ovome slučaju biti će pronađen LCD na adresi 0x27 te senzor gesti na adresi 0x39. Na temelju brojača definiran sa nDevices određuje je li pronađen bilo kakav uređaj ili ne. Ukoliko funkcija WIRE.endTransmission daje povratnu vrijednost 0 pronađen je uređaj na trenutnoj adresi u petlji, ukoliko daje povratnu vrijednost 4 došlo je do nepoznate greške na trenutnoj adresi. Ta greška može značiti odspajanje uređaja, kratki spoj, prekid u krugu i sl. Rezultate skeniranja

program ispisuje u serijski monitor. Program izvršava skeniranje svakih 5 sekundi pomoću funkcije delay(5000) napisane na kraju loop funkcije.

```
#include <Wire.h>

// Set I2C bus to use: Wire, Wire1, etc.
#define WIRE Wire

void setup() {
  WIRE.begin();

  Serial.begin(9600);
  while (!Serial)
     delay(10);
  Serial.println("\nI2C Scanner");
}


void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    WIRE.beginTransmission(address);
    error = WIRE.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);           // wait 5 seconds for next scan
}
```

Sl. 6.1. Softverski kod za I2C skener

```ino
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SparkFun_APDS9960.h>
// Global Variables
SparkFun_APDS9960 apds;
LiquidCrystal_I2C lcd(0x27, 16, 2);
int visitors=0;
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.print("Visitors: ");
  lcd.setCursor(0, 1);
  lcd.print(visitors);
  Serial.println("APDS-9960 - Gesture test");
  if (apds.init()) {
    Serial.println("APDS-9960 initialization complete");
  } else {
    Serial.println("Something went wrong during APDS-9960 init!");
  }
  // Adjust gesture parameters for improved consistency
  apds.setGestureEnterThresh(90);
  apds.setGestureExitThresh(75);
  apds.setGestureGain(GGAIN_4X);
  if (apds.enableGestureSensor(true)) {
    Serial.println("Gesture sensor is now running with adjusted parameters");
  } else {
    Serial.println("Something went wrong during gesture sensor init!");
  }
}
void loop() {
  if (apds.isGestureAvailable()) {
    Serial.println("Gesture interrupt triggered");
    uint8_t gesture_data = apds.readGesture();
    Serial.print("Raw Gesture Data: 0x");
    Serial.println(gesture_data, HEX);
    switch (gesture_data) {
      case DIR_UP:
        Serial.println("Gesture: UP");
        break;
      case DIR_DOWN:
        Serial.println("Gesture: DOWN");
        break;
      case DIR_LEFT:
        Serial.println("Gesture: LEFT");
        visitors++;
        lcd.setCursor(0, 1);
        lcd.print(visitors);
        break;
      case DIR_RIGHT:
        Serial.println("Gesture: RIGHT");
        if(visitors>0)visitors--;
        lcd.setCursor(0, 1);
        lcd.print(visitors);
        break;
      case DIR_NEAR:
        Serial.println("Gesture: NEAR");
        break;
      case DIR_FAR:
        Serial.println("Gesture: FAR");
        break;
      default:
        Serial.println("Unknown Gesture");
    }
  }
}
```

Sl. 6.2. Softverski kod implementacije detekcije prolaska

Prilikom objašnjavanja softverskog koda za kompletnu implementaciju, kod će biti podijeljen u dijelove. Osim funkcionalnosti, kod sadrži i određene dijelove i naredbe za testiranje koda putem serijskog monitora. Osim što prepoznaje geste za lijevo i desno, također prepoznaje i geste za gore, dolje, približavanje i udaljavanje od senzora. Također sadrži i funkciju za provjeravanje rada senzora gesti te ispisuje njegovo stanje nakon svake prepoznate geste, ili ako gesta nije prepoznata to će i ispisati. Osim toga ispisuje i kada je okinut senzor, odnosno kada je detektiran prekid (interrupt). Slika 6.2. prikazuje čitavi softverski kod implementacije.

Prvi dio koda, od 1. do 7. linije, služi za definiranje biblioteka koje su korištene te za definiranje globalnih varijabli. Globalne varijable u ovome kodu su „apds“ (koristi se za rad sa senzorom), „lcd“ (koristi se za rad sa displejom, 0x27 predstavlja njegovu adresu, 16 predstavlja broj simbola po retku, 2 predstavlja broj redaka prikaza teksta) te „visitors“ što predstavlja brojač osoba koje su ušle ili izašle iz prostorije. Od 10. do 14. linije koda izvršena je inicijalizacija LCD-a (naredba lcd.init()). Naredbom lcd.backlight() pali se osvjetljenje na displeju, naredbom lcd.print() vrši se ispis teksta na LCD, naredba lcd.setCursor postavlja redak i stupac na koji se ispisuje tekst na LCD-u (u ovome slučaju, nakon ispisa „Visitors:“ postavlja se na početak drugog retka te će se tu ispisivati broj posjetitelja (globalna varijabla „visitors“ definirana ranije). U nastavku, od 15. do 30. linije koda vrši se inicijalizacija (naredba apds.init()) te podešavanje senzora gesti. Naredbe apds.SetGestureEnterThresh() te apds.SetGestureExitThresh služe za podešavanje osjetljivosti i preciznosti detekcije gesti. Razlozi korištenja ovih naredbi biti će detaljnije opisani u idućem poglavlju. Naredba apds.setGestureGain() predstavlja osjetljivost detekcije, odnosno udaljenost na koju senzor može detektirati pokrete. U ovome slučaju postavljena je na zadanu vrijednost iz biblioteke. Naredba apds.enableGestureSensor(true) pali senzor gesti i provjerava je li isti upaljen.

Preostale linije ispisuju povratne informacije o radu senzora na serijski monitor.

U loop funkciji provjeravaju se geste senzora i vrši se detekcija gesti. Prije samog provjeravanja gesti postoje 2 dodatne funkcije koje testiraju rad senzora. Funkcija apds.IsGestureAvailable() provjerava postoji je li detektirana bilo kakva gesta pred senzorom, te ukoliko je vraća povratnu vrijednost 1 odnosno TRUE, a ukoliko nije vraća 0 odnosno FALSE. Ukoliko je detektirana gesta na serijski monitor se ispisuje povratna informacija. Nadalje linije 34, 35 i 36 služe za dohvaćanje informacija, odnosno vrijednosti gesti u heksadekadskom obliku te ih ispisuje na serijski monitor, u svrhu testiranja. Ostatak koda je SWITCH-CASE funkcija. Ova funkcija ima nekoliko mogućih različitih slučajeva od kojih se samo jedan može izvršiti tokom jednog izvršavanja loop funkcije. Postoje slučajevi za svaku detektiranu gestu posebno (gore, dolje, lijevo, desno, udaljavanje,

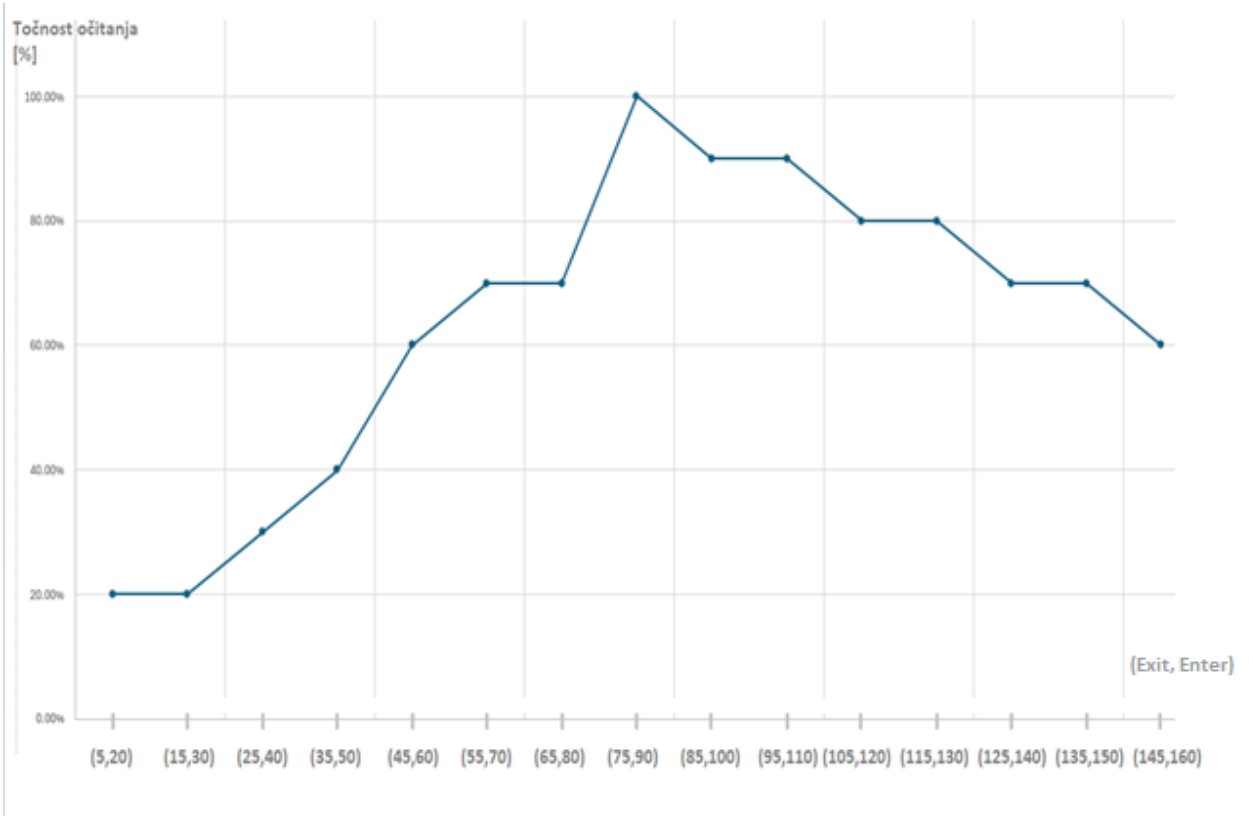
približavanje), međutim samo su 2 slučaja potrebna u funkcionalnosti sustava detekcije prolaska a to su, ovisno o orijentaciji i postavljanju senzora, lijevo i desno (gore i dolje). U ova 2 slučaja dodan je kod koji prilikom detekcije jedne od ove dvije gesti povećava broj posjetitelja te ga ispisuje na displej, a u slučaju detekcije druge geste ga smanjuje i ponovo ispisuje na displej. U kompletnom kodu implementacije nije niti jednom korištena naredba delay. Razlog tome je što je korišteni senzor jako brz i jako precizan te ima mogućnost detektiranja gesti sa minimalnim razmakom od samo 2.8 milisekundi. Osim slučajeva za sve moguće geste, postoji i slučaj za nedefiniranu gestu, koji radi na principu toga što senzor u slučaju nedefinirane geste vraća povratnu informaciju default. Ovaj slučaj također nema svrhu u samoj implementaciji potpune funkcionalnosti sustava, nego služi samo za testiranje sustava i pronalazak pogrešaka. Kod loop funkcije se ponavlja i detektira svaki prolazak. U slučaju korištenja nekog od prijašnjih senzora, odnosno u slučaju korištenja 2 ili više senzora, bilo bi potrebno dodati funkciju koja bi provjeravala koji je od navedenih senzora prvi aktiviran. Također postoji mogućnost dodavanja komponenti u sustav kao što su zujalica koja bi se aktivirala prilikom aktivacije senzora ili tipkala za promjenu smjera rada senzora ili za resetiranje brojača na LCD-u na 0. Dodavanje koda za navedene komponente bi bilo poprilično jednostavno za izvršiti i ne bi zahtijevalo nikakvu izmjenu postojećeg koda, već samo dodavanje koda na određena mjesta. Prilikom testiranja koda još je bitno naglasiti da je potrebno uskladiti baud rate serijskog monitora sa baud rate-om određenim u inicijalizaciji samog serijskog monitora (linija 9). U slučaju da ove vrijednosti nisu iste, serijski monitor će u većini slučajeva davati loš ispis, odnosno ispisivat će nasumične simbole koji nemaju veze sa napisanim softverskim kodom.

7. Rezultati i analiza

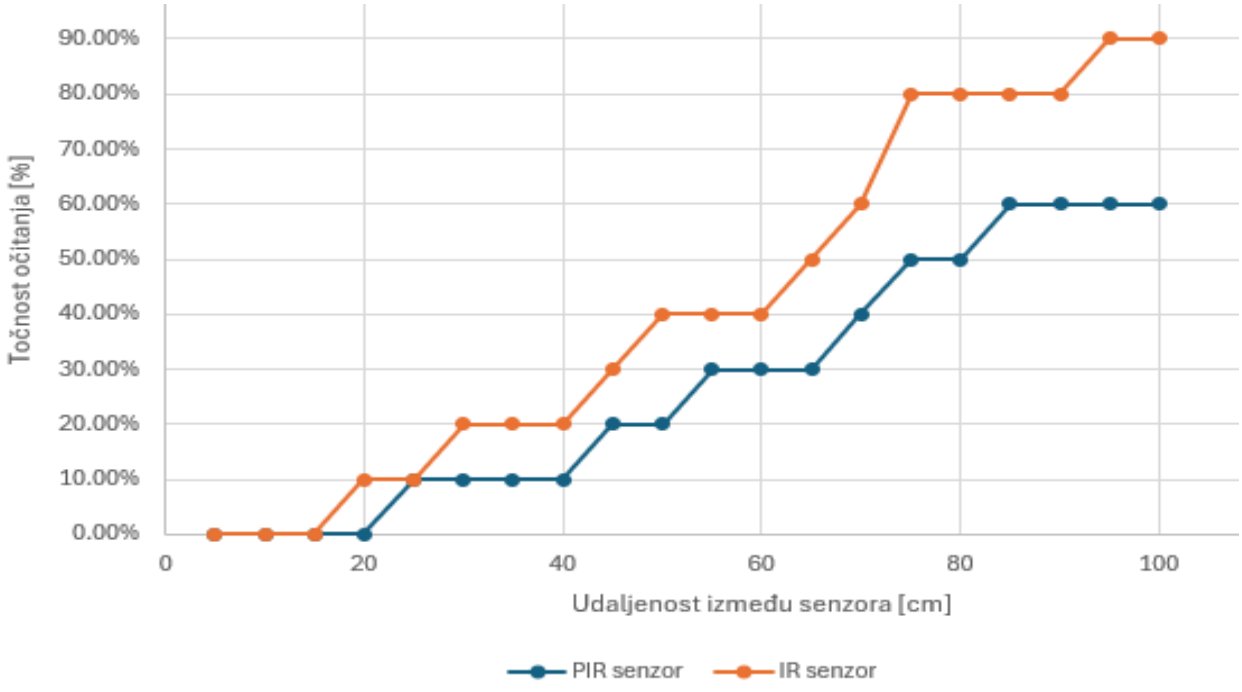
Prilikom samog testiranja sustava došlo je do mnogo izazova i zamjena. Sustav je prvobitno bio izrađen metodom korištenja 2 PIR senzora, međutim taj sustav je bio jako netočan te je bilo potrebno udaljiti senzora na udaljenost veću od skoro 2 metra kako ne bi utjecali jedan na drugog i kako bi detekcija bila točna. Nadalje, nakon prijelaza na APDS9960 nastao je problem sa razvojnom pločicom. Prvobitna pločica nije imala način rada na 3.3V koji zahtjeva senzor, iz toga razloga senzor je jako često očitavao krive vrijednosti i loše je očitavao pokrete. Došlo je do prestanka rada jednog od senzora iz razloga što je prilikom rada bio priključen na 5V. Također, prilikom korištenja prvobitno pločice, dolazilo je do isključivanja različitih komponenti prilikom rada sustava radi neke greške u napajanju koja je nastajala na pločici. Ti su problemi riješeni koristeći razvojnu pločicu sa prekidačem za rad na 3.3V. Senzor je u tom slučaju puno bolje i

preciznije radio, međutim sada je postojao problem što je senzor pri prevelikoj brzini pokreta krivo očitavao pokrete, npr. prilikom brzog pokreta udesno očitavao je lijevo, dok je prilikom sporog pokreta u desno očitavao desno kako bi i trebao. Ovaj problem je riješen podešavanjem parametara apds.SetGestureEnterThresh() te apds.SetGestureExitThresh(). Izvorne vrijednosti za ove parametre su, redom kako su navedeni, 40 i 30. Pri podešavanju postupno su se povećavali ovi parametri u koracima od 10 do vrijednosti 100 za oba parametra. Najbolje i najpreciznije rezultate senzor je davao pri vrijednostima 90 za prvi te 75 za drugi parametar. Daljnjim povećavanjem ovih parametara došlo je do smanjenja preciznosti očitanja pokreta. Nakon izvršenog testiranja, rezultati su bili poprilično dobri te bi senzor u 9/10 puta dobro očitao prolazak, no sada dolazimo do problema visine prolaznike. Ovisno o tome na koju visinu postavimo senzor mijenja se točnost očitanja. Prilikom postavljanja senzora na visinu veću od 1.5 metara senzor je jako precizno radio i očitavao pokrete, međutim kada se postavlja na nižu visinu u nekim slučajevima ne prepoznaje gestu odnosno ne mijenja vrijednost na LCD-u. Ovaj problem bi bilo dosta jednostavno riješiti dodavanjem još jednoga senzora u spoj na nižoj visini, te povećavanjem broja ukoliko je bilo koji od 2 senzora detektirao prolaz. Također ne bi bilo loše izraditi isti ovakav sustav koristeći infracrvene senzore. Sustav bi bio malo kompliciraniji što se tiče pitanja softverskog koda, no vrlo vjerojatno bi davao točnije rezultate. Još jedan od problema ovoga sustava je taj što je prilagodan za korištenje samo kod jednokrilnih vrata radi toga što senzor ne može očitati geste na većoj udaljenosti od 20cm. U ovome slučaju, ukoliko se radi o širim vratima, ponovo bi bilo potrebno dodavati još senzora u spoj kako bi radio točno i precizno.

Prilikom završnog testiranja i izvođenja mjerenja za APSD9960 izvršeno je 10 mjerenja za svaki par vrijednosti parametara EnterThresh i ExitThresh. Na preporuku proizvođača, korištene su vrijednosti 5 za ExitThersh te 20 za EnterThresh. Nakon 10 mjerenja za taj par vrijednosti, oba parametra povećana su za 10. Izvršeno je po 10 mjerenja za 15 parova vrijednosti, odnosno zadnje vrijednosti koje su testirane su 145 za ExitThresh te 160 za EnterThresh. Slika 7.1. prikazuje graf ovisnosti točnosti mjerenja o ovim parametrima senzora. Što se tiče PIR senzora i senzora infracrvenog zračenja, kod njih točnost mjerenja ovisi o udaljenosti između senzora. Izvedena su mjerenja na odaljenosti od 5cm do 1m između senzora, u koracima od 5cm te je točnost iskazana u postotcima odnosu na 10 mjerenja. Ova ovisnost je prikazana na slici 7.2.



Sl. 7.1. Ovisnosti točnosti mjerenja o parametrima APDS9960 senzora.



Sl. 7.2. Ovisnosti točnosti mjerenja o udaljenosti između 2 senzora.

8. Zaključak

Projekt sustava za detekciju prolaska, koji koristi APDS-9960 senzor pokreta i Arduino Uno, nudi praktično i učinkovito rješenje za praćenje broja osoba koje ulaze i izlaze iz određenog prostora. Ovaj sustav rješava potrebu za preciznim nadzorom posjetitelja u različitim okruženjima, kao što su maloprodajne trgovine, knjižnice, muzeji i drugi javni prostori gdje je razumijevanje kretanja posjetitelja ključno za operativnu učinkovitost i upravljanje resursima.

Senzor APDS-9960 je ključan za ovaj projekt, jer omogućuje pouzdano otkrivanje pokreta i smjera kretanja. Integriranjem ovog senzora s Arduinom Uno, stvorili smo sustav sposoban razlikovati dolazni i odlazni promet. Korištenje infracrvene svjetlosti i osjeta blizine omogućava precizno mjerenje, osiguravajući da je točnost brojača održana čak i u različitim uvjetima osvjetljenja i udaljenosti.

Tijekom razvoja, pojavilo se s nekoliko izazova, kao što su kalibracija senzora kako bi se smanjili lažni pozitivni rezultati i osiguravanje da sustav može kontinuirano raditi bez značajne potrošnje energije. Ovi izazove su riješeni kroz iterativno testiranje i optimizaciju hardverskih i softverskih komponenti.

Osim toga, ovaj projekt naglašava potencijal kombiniranja komponenata dostupnih na tržištu s open-source platformama poput Arduina za stvaranje prilagođenih rješenja usmjerenih na specifične potrebe. Troškovna učinkovitost ovog pristupa čini ga dostupnim malim tvrtkama i organizacijama s ograničenim proračunima.

Zaključno, sustav za detekciju prolaska, koji koristi APDS-9960 senzor pokreta i Arduino Uno, ne samo da postiže svoj primarni cilj preciznog brojanja i razlikovanja posjetitelja koji ulaze i izlaze, već također pokazuje šire implikacije automatizacije temeljene na senzorima u raznim primjenama. Uspješna provedba ovog projekta naglašava važnost korištenja tehnologije za prikupljanje korisnih podataka, što u konačnici doprinosi informiranijem donošenju odluka i poboljšanoj operativnoj učinkovitosti. Ovaj projekt služi kao temelj za buduće inovacije i poboljšanja u sustavima praćenja posjetitelja, otvarajući put za pametnija okruženja.

9. Literatura

[1] G.Gridling, B.Weiss, Introduction to Microcontrollers, Version 1.4 , str. 6-15, veljača 2007.

[2] IEEE Annals of the History of Computing , Volume: 44, Issue: 3, str. 59 – 69, srpanj-rujan. 2022,

[3] Medium, O. Gangangana, Von -Neumann Architecture vs Harvard Architecture , ožujak 2021., dostupno na : https://oshadeegangangana.medium.com/von-neumann-architecture-vs-harvard-architecture-fdb20d6f2a70 [12.7.2024.]

[4] Y. A. Badamasi, The working principle of an Arduino, 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), rujan-listopad 2014.

[5] Sahoo, K. C., & Pati, U. C. IoT based intrusion detection system using PIR sensor. 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). str. 1641.

[6] R.Ismail et al IOP Conf. Ser.: Mater. Sci. Eng. 152 012064, Obstacle-avoiding robot with IR and PIR motion sensors, str.3, 2016.

[7] J.Mankar, C.Darode, K.Trivedi, M.Kanoje, P.Shahare, Review of I2C protocol, International Journal of Research in Advent Technology (IJRAT), Volume 2, Issue 1, str. 475,siječanj 2014.

[8]L.Louis, Working principle of Arduino and using it as a tool for study and research, International Journal of Control, Automation, Communication and Systems (IJCACS), Vol.1, No.2, str. 25-26, travanj 2016.

Sažetak

Cilj ovoga završnog rada je izraditi sustav za detekciju prolaska koristeći mikroupravljački sustav. Ovaj cilj ostvaren je korištenjem Arduino Uno razvojnog sustava te APDS9960 senzora gesti, udaljenosti i jačine svjetlosti. Sustav je izrađen na način da senzor gesti prepoznaje geste u 2 različita smjera i na temelju toga određuje je li osoba ušla ili izašla iz prostorije. Rezultati se ispisuju na LCD displej putem I2C komunikacije. Sustav prepoznaje ulazak i izlazak osoba iz prostorije, različitim brzinama i različitih visina. Sustav je radi ograničenja senzora namijenjen za objekte i prostorije sa jednokrilnim vratima. Rezultat ovog projekta je sustav koji ekonomično, brzo i precizno broji posjetitelje koji ulaze i/ili izlaze iz objekta.

Ključne riječi : Arduino, mikroupravljač, detekcija prolaska, senzor pokreta, infracrveni senzor, dvosmjerna detekcija

Summary

The objective of this final project is to create a passage detection system using a microcontroller system. This goal is achieved by utilizing the Arduino Uno development system and the APDS9960 sensor for gestures, proximity, and light intensity. The system is designed so that the gesture sensor recognizes gestures in two different directions and, based on this, determines whether a person has entered or exited the room. The results are displayed on an LCD screen via I2C communication. The system detects the entry and exit of individuals from the room at different speeds and heights. Due to sensor limitations, the system is intended for buildings and rooms with single-leaf doors. The result of this project is a system that economically, quickly, and accurately counts the number of visitors who are entering and/or exiting the premise that the sensor is installed in.

Keywords : Arduino, microcontroller, passage detection, motion sensor, infrared sensor, bidirectional detection