**UNIVERZITET U NOVOM SADU**

**FAKULTET TEHNIČKIH NAUKA**

**NOVI SAD**

**Departman za računarstvo i automatiku**

**Odsek za računarsku tehniku i računarske komunikacije**

**ISPITNI RAD**

**Kandidati: Milan Popđurđev, Jovan Pilčević i Bojan Gostojić**

**Brojevi indeksa: RA 131/2018, RA 15/2016, RA 102/2018**

**Predmet: Sistemska programska podrška u realnom vremenu II**

**Tema rada: Merač obrtaja**

**Mentor rada: prof. dr. Miroslav Popović**

**Novi Sad, januar, 2021.**

**Sadržaj**

strana

1. Opis projektnog zadatka 3
2. Analiza problema i koncept rešenja 4
3. Opis rešenja 5
   1. Povezivanje lasera i senzora 5
   2. Rukovalac 5
   3. Korisnička aplikacija 6
4. Testiranje 7
5. Zaključak 7
6. **Opis projektnog zadatka**

Potrebno je realizovati Linux rukovalac za Raspberry PI 2 uređaj koji treba da omogući komunikaciju korisničkog programa, lasera i fotosenzora, koji povezani na GPIO prolaze dostupne na uređaju predstavljaju uređaj za merenje broja obrtaja. Digitalni izlaz sa fotosenzora označava prisustvo/odsustvo svetlosti u njegovom spektru (prag se može podešavati potenciometrom na senzorskoj pločici), dok digitani ulaz lasera označava kada je laser uključen/isključen.

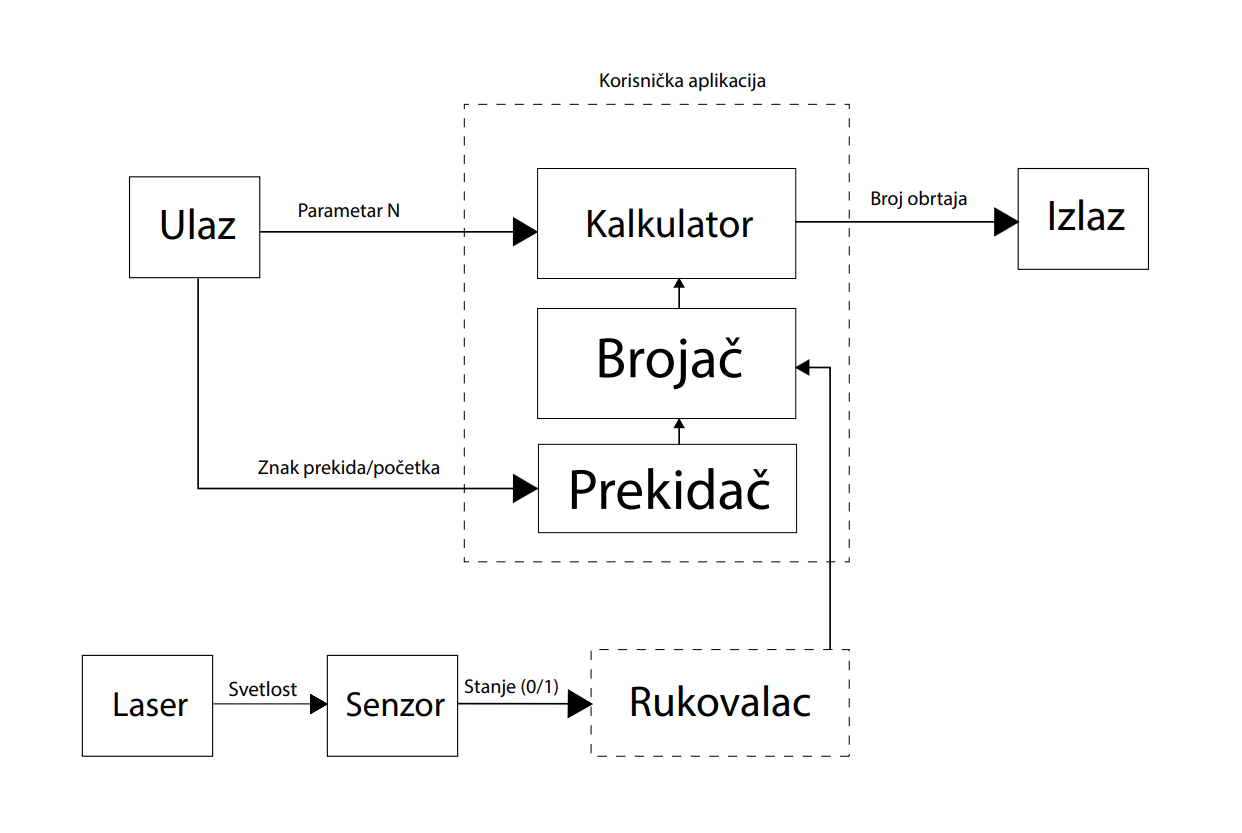
Rukovalac treba da omogući sledeću funkcionalnost:

* Uključivanje/isključivanje obrtometra (lasera i programske podrške) na zahtev korisničke aplikacije i postavljanje parametra N (broj presecanja zraka u jednom obrtaju, npr. kod ventilatora bi to bio broj peraca ventilatora)
* Preuzimanje stanja fotosenzora (prisustvo/odsustvo svetlosti)
* Prosledjivanje informacije sa fotosenzora korisničkoj aplikaciji na zahtev ove vredosti kroz odgovarajuću operaciju čitanja
* Određivanje broja obrtaja na osnovu periode prekida i zadatog parametra N
* Prosleđivanje informacije o broju obrtaja korisničkoj aplikaciji na zahtev ove vrednosti kroz odgovarajuću operaciju čitanja

U svrhu provere funkcionalnosti razvijenog rukovaoca potrebno je razviti korisničku aplikaciju za određivanje broja obrtaja koja posredstvom implementiranog rukovaoca može da dobavi potrebne informacije koje izvorno potiču sa senzora. Aplikacija treba da omogući rad sa parom fotosenzora i lasera i da obrađuje i ispisuje informacije pristigle sa senzora.

1. **Analiza problema i koncept rešenja**

Pre početka izvršavanja programa, potrebno je podesiti osetljivost senzora tako da reaguje samo na zrak lasera. Na početku izvršavanja programa korisnik pomoću operacije pisanja prosleđuje parametar N koji predstavlja broj prekida zraka lasera po obrtaju. Svaki put kada se promeni stanje senzora sa 1 (nema svetlosti) na 0 (ima svetlosti), brojač se uvećava za 1. Kada korisnik pošalje signal za prekid brojanja, vrednost brojača u tom trenutku se deli sa parametrom N i dobija se traženi broj obrtaja koji predstavlja izlaz iz programa. Po završetku brojanja korisniku se omogućuje novi početak brojanja obrtaja pritiskom tastera ’’y’’ na tastaturi.



Slika 1. – Šematski prikaz rešenja zadatka

1. **Opis rešenja**
   1. **Povezivanje lasera i fotosenzora**

Kako bi se ostvarila funkcionalnost lasera i fotosenzora, potrebno ih je pravilno povezati na Raspberry PI uređaj. Pozitivne (+) strane obe komponente treba povezati na pinove 2 i 4 koji predstavljaju napon od 5V. Kod senzora, GND je potrebno povezati na pinove 6 ili 9 koji predstvljaju uzemljenje, a DO se, konkretno u našem slučaju, povezuje na pin 7 (GPIO4). DO je moguće povezati na bilo koji pin opšte namene, mi smo za zadatak izabrali GPIO4.

Nakon povezivanja lasera i fotosenzora na Raspberry PI, treba usmeriti snop lasera ka fotosenzoru i podesiti osetljivost senzora tako da reaguje samo na snop lasera. Osetljivost se podešava mehaničkim okretanjem potenciometra na senzoru.

* 1. **Rukovalac**

Rukovalac je deo programa koji prikuplja informacije koje šalje senzor preko GPIO pina (kao što je prethodno navedeno, u našem slučaju je to GPIO4). U rukovaocu se nalaze razne operacije poput operacija čitanja i pisanja. Mi ćemo koristiti operaciju čitanja.

Kada smo izabrali pin na koji ćemo povezati senzor, u funkciji inicijalizacije se inicijalizuje taj pin tako što se pozivaju metode SetGpioPinDirection i SetInternalPullUpDown. U ovom zadatku smer GPIO pina će biti IN, zato što ne šaljemo informacije iz rukovaoca nego primamo informacije preko tog pina. U funkciji exit pozivamo metodu ClearGpioPin kako bi se deinicijalizovao pin.

Pomoću operacije čitanja rukovalac prima stanje pina i šalje stanje pina korisničkoj aplikaciji. Pozivanjem metode GetGpioPinValue preuzimamo stanje GPIO pina (senzor registruje/ne registruje svetlost, tj. 0 ili 1). Vrednost pina se ubacuje u bafer, koji se kasnije pozivanjem metode copy\_to\_user šalje korisničkoj aplikaciji.

* 1. **Korisnička aplikacija**

Korisnička aplikacija je deo koda koji se sastoji iz dva dela: ’’glavnog’’ dela i dodatne niti koja služi za čitanje vrednosti sa tastature ukoliko korisnik želi da prekine brojanje.

U trenutku pokretanja programa od korisnika se traži da unese broj peraca, tj. broj prekida snopa lasera koji čini jedan obrtaj. Pozivom metode open se otvara modul a pozivom metode read se čita vrednost GPIO pina i postavlja se u ’’nova’’ bafer u koji se čuva svaka nova vrednost očitana sa GPIO pina. Nakon toga se promenljivoj ’’stara’’ dodeljuje vrednost iz ’’nova’’ bafera. Ovaj korak je bitan zbog toga što se time izbegava povećanje brojača na početku programa iako nije došlo do prekida snopa lasera (u trenutku pokretanja programa laser ne mora biti uperen u senzor i bez ovog koraka preusmerenjem lasera na senzor bi se povećao brojač).

Po završetku čitanja vrednosti pina i postavljanja adekvatnih vrednosti u bafer i promenljivu, metodom close se zatvara modul.

Sledeći deo korisničke aplikacije započinje ponovnim otvaranjem, čitanjem iz modula i smeštanjem nove vrednosti GPIO pina u ’’nova’’ bafer. Ukoliko je vrednost u ’’nova’’ baferu 0, a u ’’stara’’ promenljivi 1, desio se prekid snopa lasera i brojač se inkrementira. Promenljiva ’’stara’’ preuzima vrednost iz ’’nova’’ bafera i zatvara se modul. Ovaj proces se ponavlja sve dok se ne unese terminalni simbol ’’q’’ sa tastature.

Ukoliko je unesen terminalni simbol ’’q’’, vrednost brojača se deli sa brojem peraca koji je unesen na početku programa i dobija se konačni broj obrtaja koji se ispisuje na ekranu. Nakon ispisa broja obrtaja, pritiskom tastera ’’y’’ korisniku je omogućeno ponovno brojanje obrtaja.

Dodatna nit se izvršava paralelno sa ostatkom koda i služi za proveru unosa tastera ’’q’’. Ukoliko je unet taster ’’q’’, promenljiva ’’kraj’’ se postavlja na vrednost 1 i označava izlazak iz petlje koja broji prekide snopa lasera.

1. **Testiranje**

Implementaciju rešenja je moguće testirati na više načina, ali nama najjednostavniji način testiranja jeste pomoću klatna. Dva prekida snopa lasera označavaju jednu iteraciju (obrtaj) klatna. Prilikom implementacije rešenja dešavalo se da se za jedan prekid snopa lasera brojač inkrementira više puta. Ovaj problem je rešen dodatnim podešavanjem osetljivosti fotosenzora i dodavanjem usleep funkcije unutar petlje koja broji prekide. Kolegi Bojanu nije radio laser i zbog toga nije bio u mogućnosti da testira program na ovaj način, pa je podešavanjem osetljivosti senzora i jednostavnim prekrivanjem senzora šakom simulirao prekid snopa lasera.

1. **Zaključak**

Napravljen je jedan koncept za realizaciju datog problema. Testiranjem je potvrđena funkcionalnost koda. Poređenjem broja iteracija klatna i ispisa na ekranu potvrđeno je njihovo podudaranje, odnosno ispravnost programa.