

Meritev hitrosti padanja

Strokovno poročilo

Mentor: mag. Milan Setničar, prof.

Avtor: Rok Studen Levstek, R 4. D

Ljubljana, april 2024

Zahvala

Zahvaljujem se profesorju Milanu Setničarju za dragocene nasvete, podporo in spodbudo pri pripravi tega projekta. Brez vaše strokovne usmeritve in podpore ne bi mogel uspešno zaključiti tega projekta. Zahvaljujem se tudi Marcu Calistru in Gašperju Glavini za pomoč pri teoretičnem načrtovanju in Timu Kerncu ter Lovru Ozmeću za pomoč pri fizični implementaciji projekta.

Povzetek

V svoji nalogi sem se posvetil merjenju hitrosti padajoče frnikole s pomočjo laserjev KY-008 in senzorjev za svetlobo LDR s čipom LM393. Cilj projekta je natančno merjenje začetne višine spuščene frnikole iz katere koli višine, pri čemer je osnovna zahteva, da je pot padajoče frnikole pravokotna na svetlobna snopa obeh laserjev. Za doseganje cilja sem razvil vezje, ki je sposobno natančno meriti razliko v času do 3 milisekund med aktivacijo prvega in drugega senzorja. Laserja KY-008 sta bila postavljena drug nad drugim in usmerjena vodoravno, kar je omogočalo vzporedno projekcijo svetlobnih snopov. Nasproti laserjev sem na njima enakih višinah namestil senzor LDR s čipom LM393, ki analogni signal ob prisotnosti svetlobe pretvarja v binarni signal, glede na predhodno določeno mejo svetlosti, ki jo je bilo mogoče prilagoditi s potenciometrom. To zagotavlja selektivno zaznavanje intenzivne svetlobe laserjev in filtrira vplive okolice. Nato sem te komponente povezal z Arduino Uno in ustvaril program, ki z uporabo prekinitve in vgrajenega časovnika v Arduinu, začne šteti ob aktivaciji prvega senzorja (ko frnikola prekine prvi svetlobni snop) ter preneha šteti ob aktivaciji drugega senzorja (ko frnikola prekine drugi svetlobni snop). Izmerjeni čas sem uporabil v matematični formuli, ki omogoča izračun višine, iz katere smo spustili frnikolo.

Ključne besede: Meritev hitrosti, prosti pad, Arduino Uno, ATmega328P, prekinitve, časovniki

Abstract

In my assignment, I focused on measuring the speed of a falling marble using KY-008 lasers and LDR light sensors with an LM393 chip. The goal of the project was to accurately measure any initial height of a dropped marble, with the basic requirement that the path of the falling marble is perpendicular to the light beams of both lasers. To achieve this goal, I have developed a circuit capable of accurately measuring the time difference of up to 3 milliseconds between the activation of the first and second sensors. The two KY-008 lasers were placed one above the other and directed horizontally, which enabled the parallel projection of the light beams. Opposite to the lasers, at the same heights, I installed an LDR sensor with an LM393 chip, which converts analogue signals about the presence of light into binary signal, according to a predetermined brightness limit that can be adjusted with a potentiometer. This ensures selective detection of intense light from lasers and filters out ambient influences. I connected these components to Arduino Uno and created a program that, using interrupts and the built-in timer in Arduino, starts counting when the first sensor is activated (when the marble interrupts the first light beam) and stops counting when the second sensor is activated (when the marble interrupts the second light beam). I used the measured time in a mathematical formula that enables the calculation of the height from which the marble was dropped.

Key words: Speed measurement, free fall, Arduino Uno, ATmega328P, interrupts, timers

Kazalo vsebine

1. Uvod	5
2. Metodologija	6
2.1 Uporabljena oprema in material	6
2.2 Delo in potek vaje	7
2.3 Razvijalno okolje in programski jezik	8
2.4 Oprema.....	9
2.4.1 Arduino Uno	9
2.4.2 KY-008.....	10
2.4.3 LDR	11
2.4.4 LM393	12
2.4.5 Senzorni modul.....	13
2.5 Razvoj vezja.....	14
2.5.1 Postavitev laserjev in senzorjev	14
2.5.2 Povezava komponent.....	15
2.6 Programiranje.....	16
2.6.1 Inicializacija spremenljivk in pinov	16
2.6.2 Inicializacija mikrokontrolnika	16
2.6.3 Funkcija izpis	17
2.6.4 Prekinitvene rutine	17
2.6.4 Glavna zanka	18
3. Meritve in ugotovitve	19
3.1 Formula in merjenje.....	19
3.2 Tabela rezultatov in grafikon	20
3.3 Ugotovitve.....	21
4. Zaključek.....	21
5. Viri in literatura.....	22
6. Izjava o avtorstvu	23

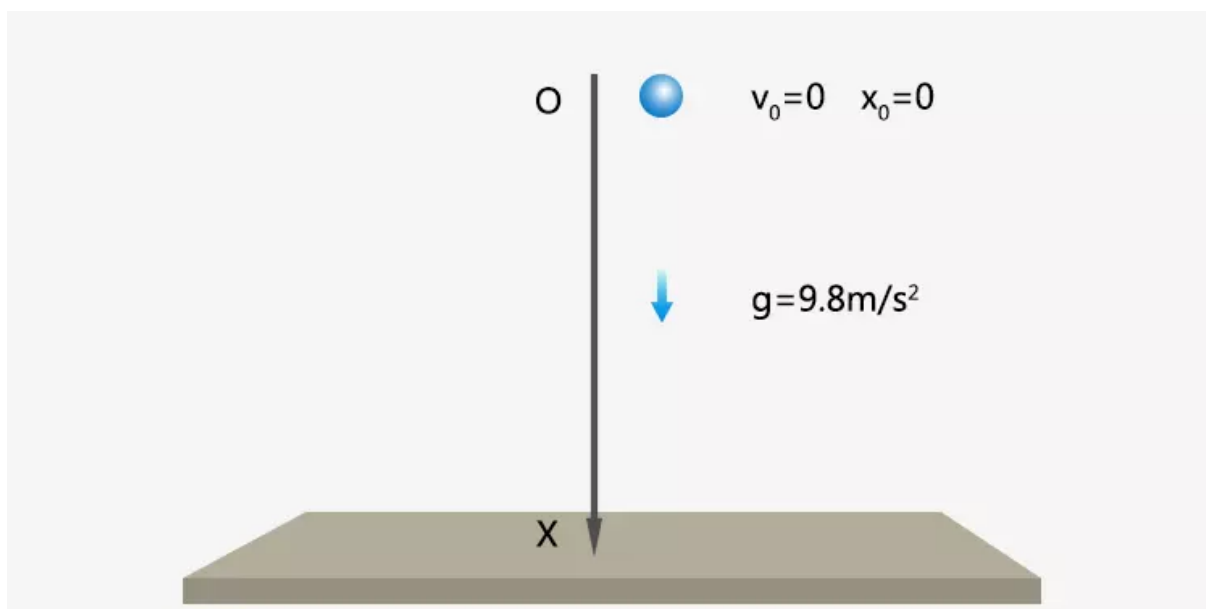
Kazalo slik

Slika 1: Prosti pad	5
Slika 2: Arduino Uno.....	9
Slika 3: Izgled KY-008	10
Slika 4: Zgradba KY-008.....	10
Slika 5: Fotoupor.....	11
Slika 6: Čip LM393.....	12
Slika 7: Senzorni modul	13
Slika 8: Skica postavitve	14
Slika 9: Protoboard.....	15
Slika 10: Inicializacija spremenljivk in pinov.....	16
Slika 11: Inicializacija mikrokrmilnika.....	16
Slika 12: Funkcija izpis	17
Slika 13: Druga prekinitvena rutina	17
Slika 14: Prva prekinitvena rutina.....	17
Slika 15: Glavna loop funkcija.....	18

1. Uvod

V sodobnem času se tehnologija nenehno razvija, kar prinaša nove možnosti za inovativne rešitve v različnih področjih. Eno od teh področij je tudi fizika, kjer se s pomočjo naprednih tehnologij izvajajo različne meritve in analize. V tej maturitetni nalogi sem se posvetil merjenju hitrosti padajoče frnikole s pomočjo laserjev KY-008 in senzorjev za svetlobo LDR s čipom LM393. Cilj projekta je bilo natančno merjenje začetne višine spuščene frnikole iz katere koli višine, pri čemer je osnovna zahteva, da je pot padajoče frnikole pravokotna na svetlobna snopa obeh laserjev.

Namen te naloge je predstaviti postopek izdelave in delovanje razvitega vezja ter analizirati pridobljene rezultate. Skozi naloge bomo spoznali uporabo različnih komponent, kot so laserji, senzorji in mikrokrmilnik Arduino, ter kako jih povezati v funkcionalno celoto za izvajanje natančnih meritev premikanja telesa v prostem padu, kot je prikazano na spodnji sliki.



Slika 1: Prosti pad (vir: <https://www.anec.org/en/physics/free-fall.htm>)

2. Metodologija

2.1 Uporabljena oprema in material

Pri izdelavi sistema za merjenje hitrosti padajoče frnikole smo uporabili več komponent. Prva in najpomembnejša sta bila **laserja KY-008**, ki sta služila kot vir svetlobe za detekcijo gibanja padajoče frnikole. Laserji KY-008 so majhni in zmogljivi laserji z rdečo svetlobo, ki so primerni za različne elektronske projekte.

Za detekcijo prekinitve laserskega snopa sem uporabil **senzorje za svetlobo LDR**. Ti senzorji, spremenijo upornost glede na jakost svetlobe, kar omogoča zanesljivo zaznavanje prekinitve laserskega snopa, ko skozi njih pade frnikola.

Glavna kontrolna enota sistema je bil **mikrokontrolnik Arduino Uno**. Ta priljubljen mikrokontrolnik s širokim izborom digitalnih in analognih vhodov/izhodov je bil odgovoren za obdelavo podatkov, nadzor delovanja sistema ter upravljanje z drugimi komponentami.

Poleg glavnih komponent sem uporabil tudi dodatne elektronske komponente, kot so **čip LM393 in potenciometer**. Te komponente so bile potrebne za zagotavljanje stabilnega delovanja vezja, filtriranje signala ter prilagajanje različnih nastavitev za optimalno delovanje sistema.

Za material ohišja sem izbral les, saj ga je dokaj lahko obdelovati po meri in je dovolj trd za stabilno ohišje, ki se ne maje ali pa hitro zlomi. Poleg tega je bil zelo poceni, zato je bil najboljša izbira za dijaka z omejeno štipendijo.

2.2 Delo in potek vaje

Pri izdelavi sistema za merjenje hitrosti padajoče frnikole sem se srečal z izzivi, ki so zahtevali sistematičen pristop in skrbno načrtovanje. Začel sem z načrtovanjem vezja in določitvijo potrebnih komponent. Skiciral sem osnovno strukturo vezja, kjer sem predvidel postavitev laserjev, senzorjev in Arduino Uno. Pomembno je bilo zagotoviti, da je pot padajoče frnikole pravokotna na svetlobna snopa obeh laserjev za natančno merjenje.

Nato sem izvedel osnovno povezavo komponent, da bi preveril njihovo delovanje. Skrbno sem povezal laserje, senzorje in Arduino Uno, pri čemer sem pazil na pravilno postavitev in kvaliteto povezav. Po montaži sem preveril delovanje sistema, da sem se prepričal o pravilni povezavi in funkcionalnosti. Nato sem nadaljeval s programiranjem na Arduino Uno. Razvil sem program, ki je uporabljal prekinitve in vgrajeni časovnik za merjenje časa prekinitve laserskega snopa, ko frnikola pade skozi. Po končani kodi, sem vezje postavil vodoravno na mizo in z roko testiral njegovo funkcionalnost ter odkril, da senzorji zaznavajo zunanjo svetlobo.

Po tem sem opravil kalibracijo senzorjev za izolirano zaznavanje prekinitve laserskega snopa. Uporabil sem potenciometre in čip LM393 na senzorjih LDR za prilagoditev občutljivosti senzorjev, glede na svetlost laserskih snopov. Vezje sem nato ponovno testiral v vodoravni konfiguraciji in odkril, da je najmanjši čas med laserjema, ki ga je to vezje zmožno zaznati 2.7 milisekund.

Opremljen s to informacijo sem se podal v načrtovanje vertikalne postavitve senzorjev. Paziti sem moral, da frnikola ni previsoko, saj bi to pomenilo, da je čas med laserjema manj kot prej odkrita meja zaznavanja. Imel sem težave s pridobitvijo formule za računanje začetne višine, saj je edini podani podatek čas, ki ga frnikola potrebuje za še nedoločeno razdaljo med laserjema. S pomočjo mentorja in drugih kontaktov, sem razvil formulo in napisal kratek program za hitro računanje začetnih višin. Tako sem določil optimalno razdaljo med laserjema in podal tri različne začetne višine in čas, ki ga je frnikola porabila za opravljanje poti med laserjema.

Zadnji korak je bila fizična implementacija, pri kateri sem digitalno narisal natančen načrt prototipa in pridobil materiale za gradnjo. Po uspešni gradnji sem izvedel ponovno kalibracijo komponent in potrdil moje izračunane meritve, ki sem jih do tedaj imel le na papirju. Z različnih začetnih višin sem spuščal frnikolo in poskušal zadeti snopa obeh laserjev. Zbrane podatke sem nato analiziral ter tako potrdil natančnosti in zanesljivosti sistema.

2.3 Razvijalno okolje in programski jezik

Za razvoj merilnega sistema sem se odločil uporabiti **Arduino IDE**, integrirano razvojno okolje, ki je posebej zasnovano za programiranje Arduino mikrokrmilnikov. Arduino IDE je uporabniku prijazno okolje, ki ponuja intuitiven vmesnik, ki omogoča enostavno pisanje in urejanje programske kode.

Ena od ključnih prednosti Arduino IDE je njegova obsežna knjižnica funkcij in primerov, ki pomagajo pri razvoju in testiranju različnih projektov. Knjižnica omogoča hitro in učinkovito integracijo različnih senzorjev, aktuatorjev in drugih komponent, kar omogoča hiter razvoj. Hitrost nalaganja kode na mikrokrmilnik je še ena izmed odlik Arduino IDE. Z enostavnim klikom na gumb "Upload" lahko uporabnik hitro naloži svojo kodo na mikrokrmilnik, kar omogoča hitro iterativno testiranje in prilagajanje kode.

Programski jezik **C++** se je izkazal za optimalno izbiro zaradi svoje zmogljivosti in fleksibilnosti. Jezik omogoča učinkovito upravljanje z digitalnimi in analognimi vhodi/izhodi, kar je bilo bistvenega pomena za razvoj sistema, ki je zaznaval prekinitve laserskega snopa in beležil čas med njimi. Uporaba prekinitvenih rutin v C++ je bila ključna za zagotavljanje hitrega in natančnega odziva sistema. To je omogočalo takojšnjo zaznavo prekinitve laserskega snopa in beleženje časovnika, kar je bilo osnovno za izračun hitrosti padajoče frnikole.

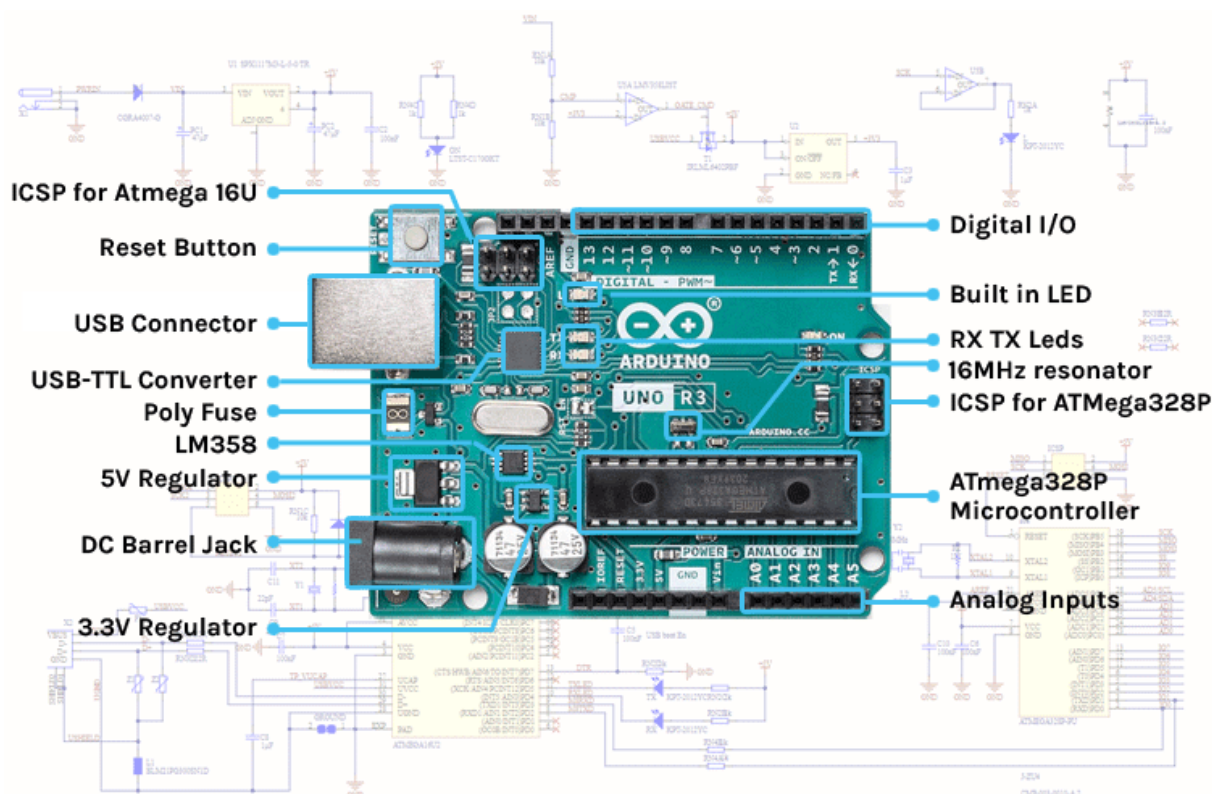
V razvoju programske kode sem se osredotočil na optimizacijo delovanja sistema, zagotavljanje zanesljivosti in natančnosti meritev. C++ jezik je omogočil implementacijo kompleksnih algoritmov za obdelavo podatkov in sprotno preverjanje napak, kar je prispevalo k natančnosti merilnega sistema.

2.4 Oprema

2.4.1 Arduino Uno

Mikrokrmilnik Arduino Uno je ena najbolj priljubljenih platform za razvoj elektronskih projektov in eksperimentiranje v svetu hobistov, študentov in inženirjev. Zasnovan je za enostavno uporabo in programiranje, hkrati pa ponuja dovolj zmogljivosti za izvajanje različnih nalog. Temelji na mikrokrmilniškem čipu ATmega328P, ki ga proizvaja podjetje Atmel. Ima 14 digitalnih vhodno/izhodnih pinov (od tega jih je 6 sposobnih generiranja pulziranih signalov), 6 analogno-digitalnih pretvornikov, 16 MHz kristalni oscilator, USB priključek za programiranje in napajanje ter vmesnik za serijsko komunikacijo. Zaradi svoje razširljivosti in prilagodljivosti je Arduino Uno pogosto uporabljen v različnih projektih, kot so avtomatizacija doma, robotika, merilne naprave, umetna inteligenca, internet stvari (IoT) in še veliko več. Poleg tega je na voljo veliko knjižnic in razvojnih orodij, ki olajšajo razvoj in programiranje za Arduino Uno, kar ga dela privlačno izbiro za začetnike in izkušene razvijalce.

Za moj projekt je Arduino Uno igral ključno vlogo pri zajemanju podatkov iz senzorjev, nadzoru eksperimentalnega sistema ter izvajanju algoritmov za analizo in obdelavo podatkov. Z njegovo pomočjo sem lahko učinkovito upravljal z elektronskimi komponentami ter izvajal merjenje hitrosti padanja. Za lažjo predstavbo, sem na sliki spodaj prikazal mikrokrmilnik z označenimi deli.



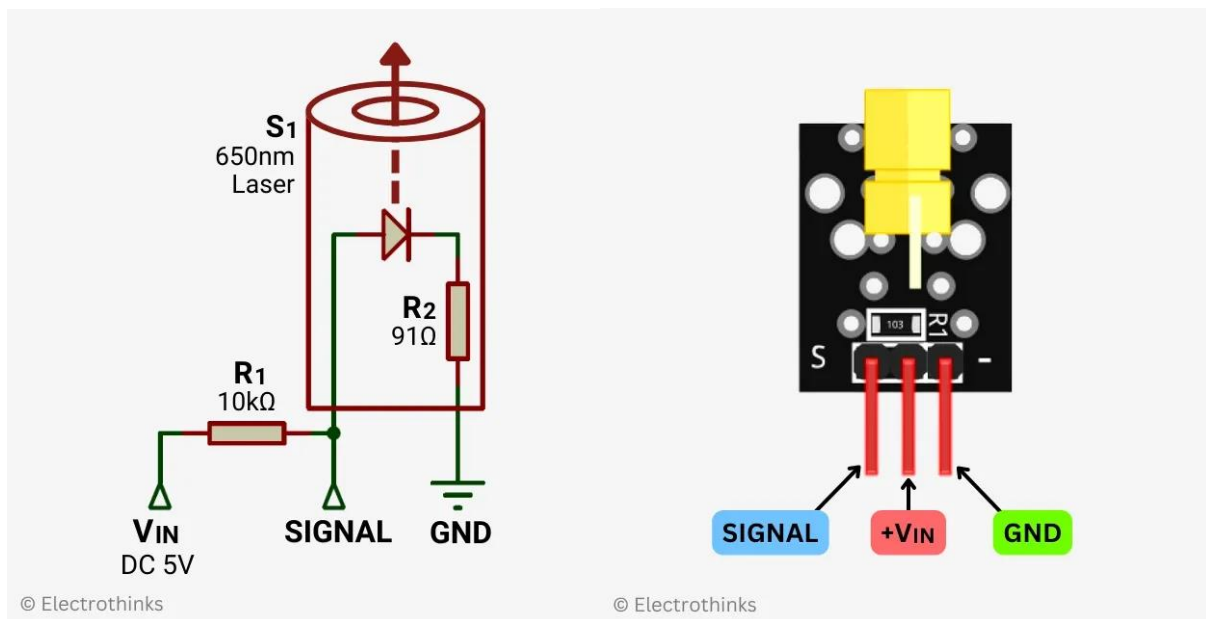
Slika 2: Arduino Uno (vir: <https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware>)

2.4.2 KY-008

Laser KY-008 predstavlja majhen, a izjemno zmogljiv modul, ki je v elektronskih krogih zelo priljubljen zaradi svoje enostavne uporabe in ugodne cene. Ta modul vsebuje infrardeči (IR) laser s 5 mW izhodno močjo, ki omogoča jasen in svetel laserski žarek s 650 nm valovno dolžino. Kljub svoji velikosti pa laser KY-008 ponuja visoko svetilnost, ki je dobro vidna tudi v svetlejših okoljih. Ena izmed ključnih značilnosti laserja KY-008 je njegova nizka poraba energije, kar omogoča dolgo življenjsko dobo baterije ali napajalnika. V nekaterih primerih so lahko vključeni tudi zaščitni elementi, kot so diode za prenapetostno zaščito.

Njegov uporaba je izjemno preprosta - potrebno je le povezati pin za signal na ustrezno napetost (običajno 5V) in ozemljitveni pin z ozemljitvijo. Laser lahko tudi vklopimo in izklopimo. To storimo tako, da uporabimo sredinski pin za napajanje in levega za nastavljane aktivnosti laserja. Spodnji sliki nam prikazujeta kako je laser zgrajen in njegov fizičen izgled.

Zaradi svoje majhnosti, zmogljivosti in enostavne uporabe se laser KY-008 uporablja v številnih aplikacijah, kot so merilne naprave, senzorji za zaznavanje ovir, projekcija vzorcev in še več. V mojem projektu sem uporabil laser KY-008 za ustvarjanje dveh laserskih snopov, med katerima je frnikola prehajala. Zaznavali smo prekinitve laserskih snopov s senzorji za svetlobo, kar nam je omogočilo natančno merjenje časa potrebnega za padanje frnikole.



Slika 4: Zgradba KY-008 (vir: <https://www.electrothinks.com/2023/08/laser-transmitter-module.html>)

Slika 3: Izgled KY-008 (vir: <https://www.electrothinks.com/2023/08/laser-transmitter-module.html>)

2.4.3 LDR

Fotoupor, bolj znano kot Light Dependent Resistor (LDR), je pasivna elektronska komponenta, ki deluje na osnovi fotovoltaičnega efekta. Njegova osnovna struktura je sestavljena iz dveh polprevodniških plošč, običajno sestavljenih iz materialov, kot sta kadmijev sulfid (CdS) ali kadmijev selenid (CdSe), ki sta občutljiva na svetlobo. Med tema dvema ploščama je prisoten tanki sloj izolacijskega materiala, kar je razvidno na spodnji sliki fotoupora.

V neosvetljenem stanju je upornost LDR-ja visoka, kar je posledica minimalne proizvodnje prostih nosilcev naboja v polprevodniških materialih. Ko pa svetloba incidenčno deluje na LDR, se fotoni absorbirajo s strani polprevodniških materialov, kar povzroči nastanek elektronsko-donorskih parov. Ti pari elektronov in lukenj dodatno prispevajo k prevodnosti materiala, zaradi česar se upornost LDR-ja zmanjša.

Dinamika te spremembe upornosti je odvisna od distribucije svetlobe, valovne dolžine in intenzivnosti. LDR je najbolj občutljiv na svetlobo v vidnem delu spektra, vendar lahko deluje tudi v bližnjem infrardečem in ultravijoličnem delu spektra, odvisno od materiala iz katerega je izdelan. Pogosto se uporablja v aplikacijah, ki zahtevajo preprosto in poceni zaznavanje svetlobe, kot so fotografske celice, avtomatske svetilke, optični detektorji in številne druge aplikacije, kjer je potrebno zaznati spremembe v svetlobnih pogojih.



Slika 5: Fotoupor (vir: <https://robocraze.com/blogs/post/what-is-the-ldr-sensor>)

2.4.4 LM393

LM393 je dvojni operacijski ojačevalnik s napetostnim primerjalnikom, ki vsebuje dve neodvisni vhodni stopnji za primerjanje napetosti. Vsaka stopnja vključuje napetostni delilnik in referenčni napetostni vir za primerjavo z vhodno napetostjo. Operacijski ojačevalniki znotraj čipa delujejo kot primerjalniki napetosti, ki primerjajo vhodno napetost z referenčno napetostjo. Ko je vhodna napetost večja od referenčne napetosti, izhod operacijskega ojačevalnika postane visok (logična '1'), v nasprotnem primeru pa postane nizek (logična '0').

Ta logični izhod se lahko uporablja za nadzor ali sprožitev drugih komponent v elektronskem vezju, na primer stikala, releji ali mikrokrmilniki. LM393 je priljubljen zaradi svoje zanesljivosti, enostavnosti uporabe in široke uporabnosti v različnih aplikacijah, kot so avtomatizacija, nadzor, senzorji in druge elektronske sisteme. Na spodnji sliki je primer izgleda čipa.

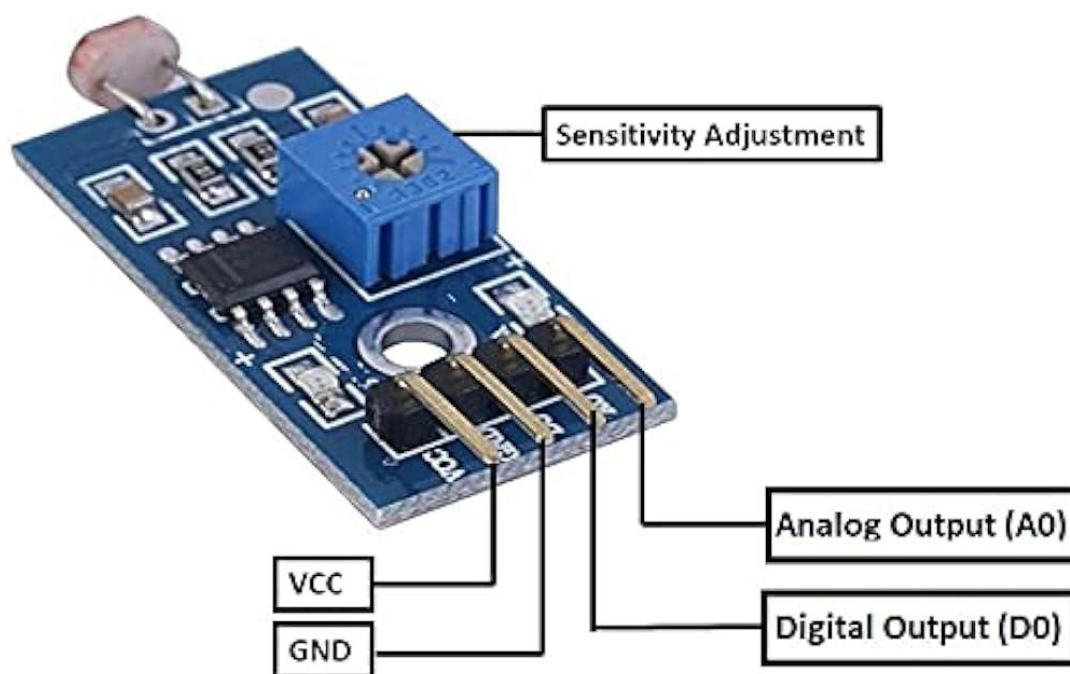


Slika 6: Čip LM393 (vir: <https://lampatronics.com/product/lm393-dual-comparator-dip8/>)

2.4.5 Senzorni modul

Celoten senzorni modul je združek LDR senzorja s čipom LM393 in potenciometrom. LDR senzor konstantno prejema svetlobo in pošilja analogno vrednost čipu. Čip nato primerja dobljeno napetost z referenčno napetostjo, ki jo lahko poljubno prilagodimo s potenciometrom. S tem da nastavimo visoko referenčno napetost, filtriramo navadno svetlobo od intenzivne svetlobe laserja KY-008, saj čip nastavi logični izhod na visoko napetost (digitalno '1') le ob prisotnosti dovolj intenzivne svetlobe in ne navadne svetlobe okolice.

Modul ima 4 vhode; eden za napajanje, drugi za ozemljitev, tretji je digitalni izhod, ki predstavlja logični izhod čipa in četrti izhod je nedotaknjena analogna vrednost LDR senzorja. Za lažjo predstavo senzorja sem spodaj podal sliko, kjer so označeni pini in potenciometer. Za projekt sem uporabil digitalni izhod, saj je implementacija digitalne vrednosti lažja in zakasnitev čipa LM393 je minimalna.



Slika 7: Senzorni modul (vir: <https://imaketoys.com/index.php/product/ldr-light-sensor-module/>)

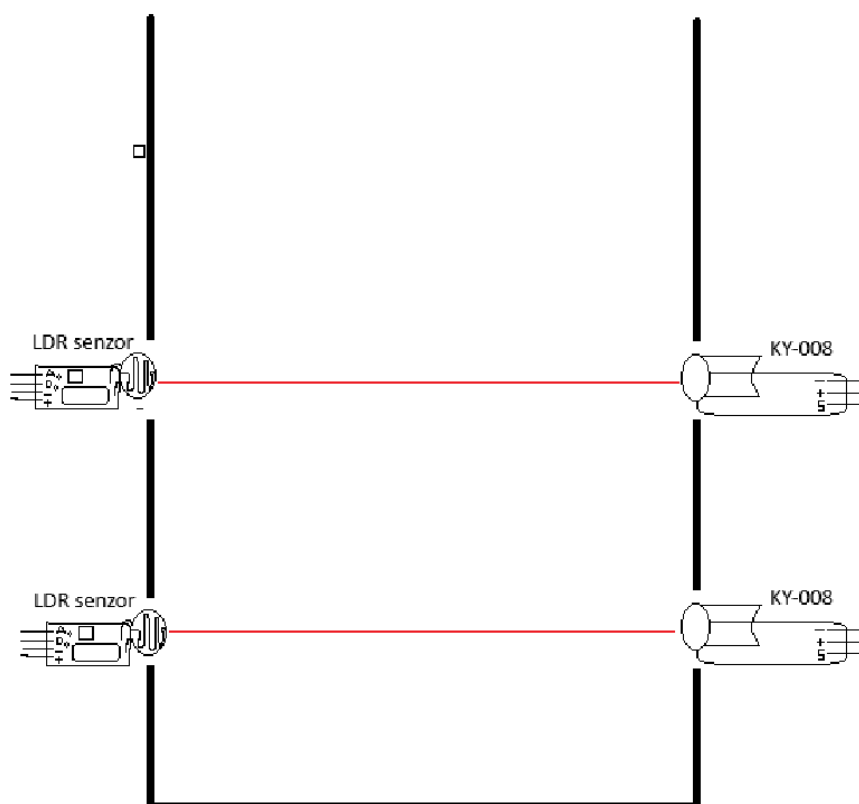
2.5 Razvoj vezja

2.5.1 Postavitev laserjev in senzorjev

Prva postavitev laserjev KY-008 in senzornih modulov je bila namenjena testiranju opreme in koncepta, zato vezje ni bilo fiksno ali pa imelo predpisanih mer. Laserja sem postavil na mizo, na njiju naravnih senzorja in testiral delovanje. Šele po drugi konzultaciji, ko sva s profesorjem potrdila pravilno delovanje vezja, sem začel načrtovati postavitev primeren merjenju frnikole med prostim padom.

Pri postavitvi laserjev KY-008 sem postavil enega nad drugim in ju usmeril vodoravno, kar je omogočilo vzporedno projekcijo svetlobnih snopov. To sem naredil zaradi potrebe po merjenju dveh različnih točk med padanjem frnikole. Natančna postavitev je bila ključna za pravilno delovanje sistema in zagotavljanje natančnih meritev. Senzorje sem namestil nasproti laserjem, na enakih višinah, da sem zagotovil vzpostavitev vodoravnega laserskega snopa. S tem sem zagotovil zanesljivost in natančnost meritev, saj so bili senzorji pravilno usmerjeni glede na pot prostega pada frnikole. Spodaj na sliki je skica, ki sem jo narisal med načrtovanjem.

Pred gradbenim postopkom sem tudi pazljivo izbral tudi material za gradnjo. Z izbiro lesa kot uporabljen material se zagotovil primerno razmerje med ceno, težo in čvrstostjo ohišja. Primerna izbira materiala je bila ključna za ohranjanje konstantne usmerjenosti svetlobe laserja in zaznavanje prekinitve snopa.



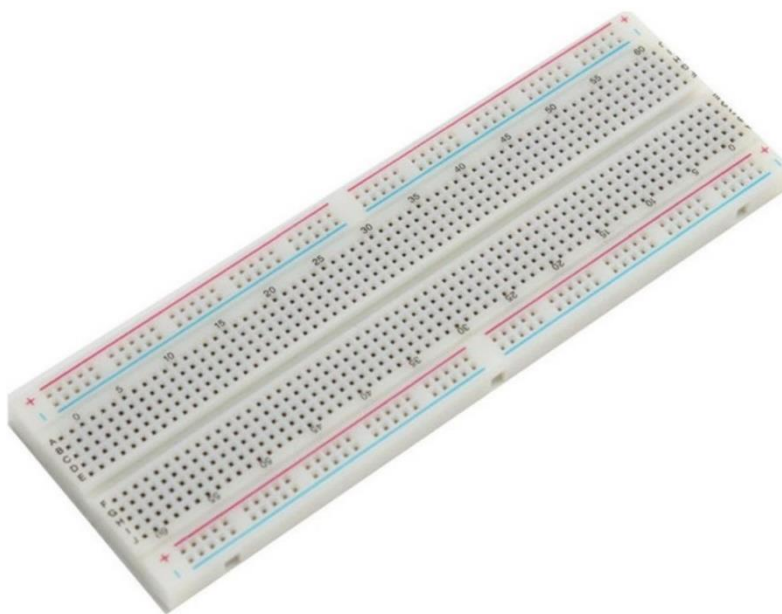
Slika 8: Skica postavitve (vir: lasten posnetek zaslona)

2.5.2 Povezava komponent

Za vzpostavitev povezave med komponentami sem uporabil žične povezave. Arduino Uno, ki deluje kot osrednja kontrolna enota, je nameščen v spodnjem delu ohišja in nanj so neposredno povezani le signalni pini senzorjev, potrebni signal za prvi laser ter protoboard, ki ga Arduino oskrbuje z napajanjem in ozemljitvijo. Za povezavo Arduina z računalnikom, sem uporabil USB vmesnik, ki je omogočal prenos podatkov med mikrokrmilnikom in računalnikom v realnem času. Uporaba protoboarda je omogočila lažjo povezavo napajanja in ozemljitve vseh komponent v sistemu. Njegov izgled je ponazorjen s spodnjo sliko.

Pri laserjih, sem na spodnjem uporabil le pin za signal in za ozemljitev. Spodnji laser je neprestano vklopljen, zato lahko signal obravnavam kot napajanje, brez da bi se laser pri tem poškodoval. Zgornji laser ima dodatno funkcijo vklapljanja in izklapljanja ob napakah pri merjenju, zato je pri njem napajanje in signal za vklop razdeljen in tako porabi vse tri pine.

Za oba senzorja sem uporabil isto število pinov. Prvega sem pustil praznega, saj ne potrebujem analogne vrednosti. Drugi je uporabljen za izhajanje digitalne vrednosti senzorja. Tretji pin je uporabljen za napajanje in zadnji je namenjen ozemljitvi.



Slika 9: Protoboard (vir: <https://www.panthera.si/elektronika/breadboard-protoboard-testna-plosca/>)

2.6 Programiranje

2.6.1 Inicializacija spremenljivk in pinov

V začetku kode definiramo več spremenljivk za upravljanje z merjenjem in komunikacijo., Ključna beseda *Volatile* pomeni, da se spremenljivke lahko spremenijo iz različnih delov kode, vključno z prekinitvenimi rutinami. Spremenljivki *prvi* in *drugi* se uporabljata za sledenje aktivacijam laserjev, *zacetek* in *konec* beležita čase, *startTime* začetni čas merjenja, *errorPrinted* pa preverja ali je napaka že bila izpisana. Konstanta *pin_prvega_laserja* določa digitalni pin, na katerem je priključen prvi laser. Na 10. sliki je prikazan ta odsek kode.

```

1  volatile bool prvi = false;
2  volatile bool drugi = false;
3  volatile unsigned long zacetek = 0, konec = 0;
4  volatile unsigned long startTime = 0;
5  volatile bool errorPrinted = false;
6
7  const int PIN_PRVEGA_LASERJA = 8;

```

Slika 10: Inicializacija spremenljivk in pinov (vir: lasten posnetek zaslona)

2.6.2 Inicializacija mikrokrmilnika

Serijsko komunikacijo z računalnikom vzpostavimo s funkcijo `Serial.begin(115200)`, ki nastavi hitrost komunikacije na 115200 bitov na sekundo. To omogoča zanesljivo izmenjavo podatkov med mikrokrmilnikom Arduino in računalnikom preko serijskega vmesnika. Funkcija `sei()` omogoča globalne prekinitve, kar je ključno za uporabo prekinitvenih rutin v programu. Te prekinitve so potrebne za takojšnjo zaznavo in obdelavo dogodkov, kot so spremembe stanja na vhodnih pinih. Na sliki 11 je prikazan odsek kode za lažjo nadaljnjo razlago, saj drugače ta ni čitljiva.

Registri `DDRB` in `PORTB` nadzorujejo smer in stanje pinov na portu B mikrokrmilnika. S kodo na 13. vrstici nastavimo pin `PB0` kot izhod, medtem ko s kodo na 14. vrstici postavim ta pin v visoko stanje. Enako velja za registre `DDRD` in `PORTD`, ki upravljajo z vhodnimi in izhodnimi pini na portu D. Z uporabo kode na 16. vrstici, pin `PD2` nastavimo kot vhod, medtem ko koda na 17. vrstici postavi ta pin v visoko stanje. Podobno postopamo za pin `PD3`.

Registri `EICRA` in `EIMSK` določajo nastavitve za zunanje prekinitvene pina `INT0` in `INT1`. S kodo na 22. in 23. vrstici določimo, da se prekinitve sprožijo ob spremembi stanja pinov `INT0` in `INT1`. Vrstica 25 in 26 pa omogočata uporabo teh prekinitvenih pinov.

```

9  void setup() {
10     Serial.begin(115200);
11     sei();
12
13     DDRB |= (1 << PB0);
14     PORTB |= (1 << PB0);
15
16     DDRD &= ~(1 << PD2);
17     PORTD |= (1 << PD2);
18
19     DDRD &= ~(1 << PD3);
20     PORTD |= (1 << PD3);
21
22     EICRA |= (1 << ISC00);
23     EICRA |= (1 << ISC10);
24
25     EIMSK |= (1 << INT0);
26     EIMSK |= (1 << INT1);
27 }

```

Slika 11: Inicializacija mikrokrmilnika (vir: lasten posnetek zaslona)

2.6.3 Funkcija izpis

Funkcija izpis je enostavna funkcija za izpis dobljenega časa med aktivacijo prvega in drugega senzorja. V funkciji se izračuna razlika v času, ki je nato deljena, da dobimo rezultat v standardni enoti (sekundah). Nato izpiše čas z natančnostjo šestih decimalk. Na sliki 12 je razviden odsek kode, kjer je napisana funkcija za izpis.

```

53 void izpis() {
54     float Razlika = (konec - zacetek) / 1000000.0;
55     Serial.print("Cas med laserjema: ");
56     Serial.print(Razlika, 6);
57     Serial.println(" sekund");
58 }
    
```

Slika 12: Funkcija izpis (vir: lasten posnetek zaslona)

2.6.4 Prekinitvene rutine

Funkcija ISR(INT0_vect) je prekinitvena servisna rutina, ki se izvaja, ko se sproži prekinitveni dogodek za pinu INT0. Da lažje sledimo razlagi kode je ta prikazana na sliki 13. Na začetku rutine izklopimo prekinitveno zastavico za INT0, da preprečimo ponovno sprožitev prekinitve med izvajanjem servisne rutine. V naslednjem delu kode preverimo, ali je prvi senzor do tedaj že bil aktiviran. Če še ni bil, je spremenljivka *prvi* nastavljena na *false* in ko jo s »!« negiramo, nas to povede v *if* stavek.

V *if* stavku prvo shranimo trenutni čas v spremenljivko *zacetek*. Nato nastavimo spremenljivke *prvi* in *drugi* na *true* in *false*. Tako shranimo, da je bil prvi senzor že sprožen. Za drugi senzor pa to pomeni, da mu nastavimo stanje na ne sprožen in tako ponastavimo kakršnokoli pomotoma sprožanje senzorja.

Sledi drugo shranjevanje trenutnega časa v milisekundah v spremenljivko *startTime*. Ta časovna oznaka se uporablja za preverjanje, ali je padec frnikole trajal več kot 2 sekundi. Na koncu rutine ponovno omogočimo prekinitveni pin INT0, kar ponovno omogoča zaznavanje naslednjih prekinitvenih dogodkov za ta pin.

Pri drugi prekinitveni rutini ISR(INT1_vect), ki je na sliki 14, se odvije večinoma isto, le v obratnem kontekstu. Najprej preprečimo ponovno sprožitev prekinitve, nato zabeležimo končni čas, ter nastavimo prvi senzor kot ne sprožen in drugi kot sprožen, kar je pomembno za preverjanje napak. Po tem izvedemo funkcijo za izpis in ponovno omogočimo zaznavanje naslednjih prekinitven na temu pinu.

```

29 ISR(INT0_vect) {
30     EIMSK &= ~(1 << INT0);
31     if (!prvi) {
32         zacetek = micros();
33         prvi = true;
34         drugi = false;
35         startTime = millis();
36     }
37     EIFR |= (1 << INTF0);
38     EIMSK |= (1 << INT0);
39 }
    
```

Slika 14: Prva prekinitvena rutina (vir: lasten posnetek zaslona)

```

41 ISR(INT1_vect) {
42     EIMSK &= ~(1 << INT1);
43     if (prvi) {
44         konec = micros();
45         prvi = false;
46         drugi = true;
47         izpis();
48     }
49     EIFR |= (1 << INTF1);
50     EIMSK |= (1 << INT1);
51 }
    
```

Slika 13: Druga prekinitvena rutina (vir: lasten posnetek zaslona)

2.6.4 Glavna zanka

Loop() funkcija naj bi bila glavna zanka programa, v kateri se sklicuje funkcije ter izvaja različna opravila a sem glavno računanje in merjenje delegiral prekinitvam zaradi boljše odzivnosti. Funkcija se konstanto izvaja tudi če nima nobenega poziva, zato sem jo uporabil za pasivno preverjanje napak pri merjenju. Za lažje sledenje razlagi, je na isti strani spodaj priložena slika funkcije.

Ob sprožitvi prvega senzorja, se zabeleži čas štetja. Ta čas je nato ob vsaki ponovitvi primerjan k trenutnemu času sistema in če je razlika dve sekundi ali več in drugi senzor še ni bil sprožen, se izvede ponastavitev celotne meritve. Ponastavijo se začetni čas in stanje prvega senzorja se nastavi kot »ne sprožen«. Potem zanka preveri, če nas je že obvestila o napaki. Če ni, se ugasne prvi laser, izpiše se, da se je zgodila napaka pri merjenju in nato se po polovici sekunde zgornji laser ponovno prižge.

Namen tega je takojšna povratna informacija o spodleteli meritvi. Nato se shrani, da smo že bili obveščeni o napaki in zanka začne nov cikel. Stranski učinek vklopljanja in izklopljanja prvega laserja je, da se dvakrat izpiše napaka v merjenju in zato je dodano preverjanje, če smo že bili obveščeni o napaki.

```

60 void loop() {
61     if (prvi) {
62         if (millis() - startTime >= 2000 && !drugi) {
63             zacetek = 0;
64             prvi = false;
65             if (!errorPrinted) {
66                 PORTB &= ~(1 << PB0);
67                 Serial.println("Napaka v merjenju: Padec je trajal več kot 2 sekundi!");
68                 delay(500);
69                 PORTB |= (1 << PB0);
70                 errorPrinted = true;
71             }
72         }
73     } else {
74         errorPrinted = false;
75     }
76 }
    
```

Slika 15: Glavna loop funkcija (vir: lasten posnetek zaslona)

3. Meritve in ugotovitve

3.1 Formula in merjenje

Meritveni inštrument je sposoben natančno meriti na katerikoli dani višini a sem za boljše razumevanje in analizo podatkov izvedel meritve na treh različnih višinah nad tlemi. Za določanje višin laserjev in treh meritvenih točk sem potreboval formulo za računanje začetne višine. Uporabil sem sledečo formulo:

$$t_2 = \frac{\Delta s}{\Delta t \cdot g} + \frac{\Delta t}{2}$$

Enačba 1: Izračun t_2

T_2 predstavlja skupni čas, ki ga je frnikola potrebovala, da opravi pot od prvega do drugega laserja. Delta s je razlika med laserjema. Za to razdaljo sem si zbral 0,40 m, saj večja kot je razdalja, več časa poteče med sprožanjem prvega in drugega laserja. To je pomembno, saj ima vezje omejitev minimalnega možnega časa 0,027 sekund. Delta T predstavlja čas, ki ga frnikola potrebuje, da opravi pot od prvega do drugega laserja.

Z nadaljnjo formulo sem še izračunal začetno višino pri različnih T_2 :

$$s_0 = g \cdot \frac{t_2^2}{2}$$

Enačba 2: Izračun začetne višine

Po ugotovitvi formule sem spisal programa v jeziku PHP, ki mi je omogočal hitro vnašanje različnih časovnih razlik med laserjema in mi izračunal potrebne začetne višine za njihovo realizacijo. Tako sem prišel do sledečih mer:

Spodnji laser sem postavil 0,05 m nad tlemi, medtem ko sem zgornji laser postavil višje, natančneje 0,45 m nad tlemi.

Pri prvem nivoju, ki je bil 0,503 m nad tlemi, sem zaznal časovno razliko (delta t) med prekinitvama laserskega snopa 0,20 sekunde. Celoten čas prostega pada je 0,32 sekunde.

Na drugem nivoju, ki je bil 0,64 m nad tlemi, sem zabeležil krajši časovni interval delta t 0,15 sekunde. Tukaj je celoten čas padanja 0,36 sekunde.

Najvišji nivo, ki je bil 1,08 m nad tlemi, je pokazal še krajšo časovno razliko delta t 0,10 sekunde. Skupni časovni interval pri tej višini je znašal 0,47 sekunde.

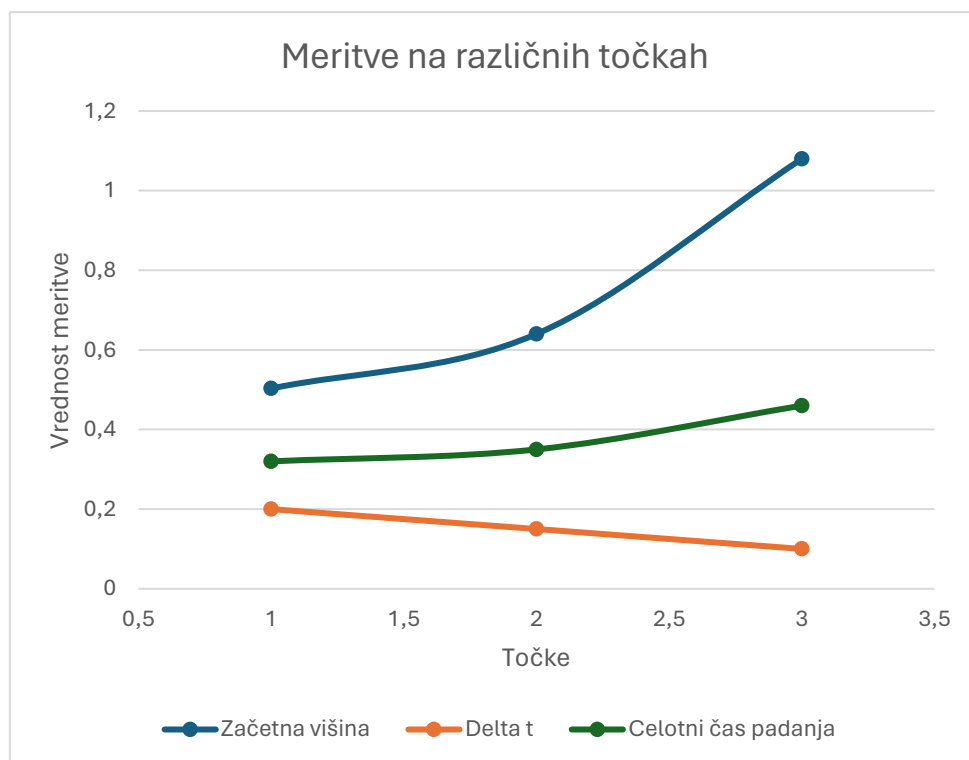
Te meritve so mi omogočile boljše razumevanje, kako razdalja med laserji in razdalja od tal vplivata na hitrost padajoče frnikole. Z rezultati sem lahko bolj ocenil natančnost in zanesljivost mojega sistema za merjenje hitrosti.

3.2 Tabela rezultatov in grafikon

Za lažjo predstavo rezultatov sem naredil tabelo in grafikon iz izmerjenih rezultatov. Tabela vključuje tri ključne točke merjenja, kjer sem zabeležil začetne višine frnikole, časovne razlike med aktivacijo spodnjega in zgornjega laserja (delta t) ter celotni čas padanja. Grafikon, ki je priložen, vizualno predstavlja te podatke, kar omogoča lažje razumevanje razmerja med začetno višino frnikole, hitrostjo padanja in časom med aktivacijo laserjev.

TABELA REZULTATOV			
Točke merjenja	Začetna višina (m)	Delta t (s)	Celotni čas padanja (s)
Prva točka	0,503	0,20	0,32
Druga točka	0,64	0,15	0,35
Tretja točka	1,08	0,10	0,46

Tabela 1: Rezultati meritev na točkah merjenja



Grafikon 1: Vizualni prikaz meritev

3.3 Ugotovitve

Na podlagi izvedenih meritev in analiz lahko izpostavim nekaj ključnih ugotovitev o delovanju in zmogljivosti sistema za merjenje hitrosti padajoče frnikole.

Razdalja med laserji je ključnega pomena za natančnost meritev hitrosti frnikole. Sprva sem predvideval krajšo razdaljo med laserjema, približno 0,10 m, vendar sem med izdelavo teoretičnega dela ugotovil, da za zagotovitev večjega časovnega okna za različne delta t potrebujem daljšo razdaljo. Če bi ohranil prvotno predvideno razdaljo, bi bil časovni okvir za zaznavanje sprememb zelo omejen.

Druga pomembna ugotovitev se nanaša na vpliv višine frnikole nad tlemi na hitrost padanja. Ugotovil sem, da višja kot je začetna višina padanja, hitreje se frnikola giblje. To sem potrdil z meritvami na različnih višinah in opazil, da se s povečanjem začetne višine sprememba v času med aktivacijo prvega in drugega senzorja niža, saj frnikola hitreje opravi pot med njima. Presenetilo me je, kako hitro se zmanjšuje vpliv višine na razlike v delta t , kar je bilo jasno prikazano tudi na grafikonu. Predvideval sem, da bom za opazne razlike v delta t potreboval manjše spremembe višine.

Tretjič, uporaba senzorjev za svetlobo LDR s čipom LM393 in mikrokontrolerja Arduino Uno se je izkazala za zanesljivo metodo za detekcijo gibanja in merjenje časa med prekinitvama laserskega snopa. Kljub nekaterim tehničnim izzivom je sistem deloval stabilno in zanesljivo.

Zadnja ugotovitev je, da je sistem primeren za nadaljnje optimizacije in prilagoditve. Z dodatnimi izboljšavami in finim prilagajanjem parametrov bi lahko dosegli še večjo natančnost in zanesljivost meritev. Skozi projekt sem pridobil dragocene izkušnje v programiranju in izdelavi merilnih sistemov, ki jih bom lahko uporabil v prihodnjih projektih.

4. Zaključek

V okviru te maturitetne naloge sem uspešno razvil in testiral sistem za merjenje hitrosti padanja frnikole z uporabo laserskih senzorjev in mikrokontrolerja Arduino. Skozi proces načrtovanja, izdelave in testiranja sem pridobil pomembne uvide v delovanje tehnologije in fizikalne lastnosti padanja predmeta.

Glavne ugotovitve kažejo, da je natančnost meritev močno odvisna od razdalje med laserskimi senzorji in začetne višine frnikole nad tlemi. Daljša razdalja med laserji omogoča večji časovni "prostor" za različne časovne razlike, kar izboljšuje natančnost meritev. Poleg tega sem opazil, da višja začetna višina frnikole vodi do hitrejšega padanja, vendar se vpliv te spremenljivke na časovne meritve med laserjema zmanjšuje s povečanjem začetne višine.

Med izdelavo sistema sem se srečal s številnimi izzivi, predvsem v programiranju in optimizaciji kode, vendar sem s trdim delom in reševanjem problemov uspešno premagal te ovire. Delo na tem projektu mi je omogočilo pridobitev dragocenih izkušenj z mikrokontrolerjem Arduino, laserskimi senzorji in programiranjem, kar bo v prihodnosti koristno pri razvoju in izvajanju podobnih projektov.

Glede na dosežene rezultate obstaja potencial za nadaljnje raziskave in izboljšave sistema, kot je integracija dodatnih senzorjev za pridobitev širšega nabora podatkov ali optimizacija algoritmov za še večjo natančnost meritev. Ta projekt je služil kot odlična učna izkušnja in osnova za nadaljnje inovacije v tej smeri.

5. Viri in literatura

1. <https://arduinomodules.info/ky-008-laser-transmitter-module/> (26. 12. 2023)
2. <https://forum.arduino.cc/t/exiting-an-interrupt/71067/13> (8. 1. 2024)
3. <https://osoyoo.com/2017/09/13/photoresistor-sensor-module/> (26. 12. 2023)
4. <https://robocraze.com/blogs/post/what-is-the-ldr-sensor> (13. 4. 2024)
5. <https://stackoverflow.com/questions/5111393/do-interrupts-interrupt-other-interrupts-on-arduino> (8.1.2024)
6. <https://www.electroduino.com/ldr-sensor-module-how-ldr-sensor-works/> (7. 1. 2024)
7. <https://www.easybom.com/blog/a/analysis-of-the-working-principle-of-lm393> (14. 5. 2024)
8. <https://www.instructables.com/LDR-Sensor-Module-Users-Manual-V10/> (13. 4. 2024)
9. <https://www.rajguruelectronics.com/ProductView?tokDatRef=ODc0OA==&tokenId=NjE=> (14. 5. 2024)
10. https://www.youtube.com/watch?v=h_y1y6eUvIY (25. 12. 2023)
11. https://www.youtube.com/watch?v=iN6MgaUf_Sg (25. 12. 2023)
12. <https://www.youtube.com/watch?v=iYcXjmJk7dQ&t=214s> (24. 1. 2024)

6. Izjava o avtorstvu

Izjavljam, da je strokovno poročilo *Merjenje hitrosti padanja* v celoti moje avtorsko delo, ki sem ga izdelal/-a samostojno s pomočjo navedene literature in pod vodstvom mentorja/-ice.

Ljubljana, 17. 4. 2024

Rok Studen Levstek