

Dinamičko mjerenje potrošnje energije električnog vozila s pomoću Arduino razvojne platforme

Tišljarić, Leo

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:924600>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -
Institutional Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Leo Tišljarić

**DINAMIČKO MJERENJE POTROŠNJE ENERGIJE
ELEKTRIČNOG VOZILA S POMOĆU ARDUINO RAZVOJNE
PLATFORME**

ZAVRŠNI RAD

Mentor: izv. prof. dr. sc. Tonči Carić

Zagreb, 2016.

Zagreb, 20. travnja 2016.

Zavod: **Zavod za inteligentne transportne sustave**
Predmet: **Algoritmi i programiranje**

ZAVRŠNI ZADATAK br. 3779

Pristupnik: **Leo Tišljarić**
Studij: **Inteligentni transportni sustavi i logistika**
Smjer: **Inteligentni transportni sustavi**

Zadatak **Dinamičko mjerenje potrošnje energije električnog vozila s pomoću Arduino razvojne platforme**

Opis zadatka:

U radu je potrebno izraditi elektronički sklop koji bi omogućio mjerenje potrošnje energije za električno vozilo Pioneer 3-AT uz pomoć hallove sonde ACS715. Za prikupljanje podataka treba izraditi program za razvojno mikroračunalo Arduino koje će uz pomoć izrađenog sklopa prikupljati podatke tijekom vožnje električnog vozila s hallove sonde. Potrebno je izvesti simuliranje vožnje na kosini pomoću vožnje na ravnoj podlozi uz dodatno opterećenje. Za provjeru simulacije potrebno je izvesti stvarna mjerenja na terenu.

Zadatak uručen pristupniku: 23. ožujka
2016.

Mentor:

Predsjednik
povjerenstva za završni
ispit:

izv. prof dr.sc. Tonči Carić

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Leo Tišljarić

**DINAMIČKO MJERENJE POTROŠNJE ENERGIJE
ELEKTRIČNOG VOZILA S POMOĆU ARDUINO RAZVOJNE
PLATFORME**

**DYNAMIC MEASURING THE ENERGY CONSUMPTION OF
ELECTRIC VEHICLE WITH ARDUINO PROTOTYPING
PLATFORM**

ZAVRŠNI RAD

Mentor: izv. prof. dr.sc. Tonči Carić

Student: Leo Tišljarić
JMBAG: 0135235019

Zagreb, rujan 2016.

DINAMIČKO MJERENJE POTROŠNJE ENERGIJE ELEKTRIČNOG VOZILA POMOĆU ARDUINO RAZVOJNE PLATFORME

U ovom radu prikazan je način prikupljanja podataka o potrošnji energije električnog vozila. Prikazan je način izrade elektroničkog sklopa za mjerenje potrebnih parametara pomoću Arduino platforme. Mobilni robot Pioneer 3-AT koristio se kao prikaz električnog vozila. Prikupljeni su podaci o potrošnji robota s obzirom na konfiguraciju terena. Predložen je način simulacije kosine dodavanjem simulacijskih kolica s dodatnom težinom.

Ključne riječi: potrošnja energije, mobilni robot, Arduino, Pioneer 3-AT, simulacija kosine

Summary

DYNAMIC MEASURING THE ENERGY CONSUMPTION OF ELECTRIC VEHICLE WITH ARDUINO PROTOTYPING PLATFORM

In this paper, the method of collecting data on the energy consumption of electric vehicles is proposed. Electronic circuit for measuring necessary parameters was made using the Arduino prototyping platform. Mobile robot Pioneer 3-AT was used as a display of electric vehicle. Data of energy consumption of the robot with regard to the configuration of the terrain was collected. This research carries out the simulation of the gradient of terrain adding simulation carts with added weight.

Keywords: energy consumption, mobile robot, Arduino, Pioneer 3-AT, simulation of gradient

1. UVOD.....	1
2. POTROŠNJA ENERGIJE MALOG ELEKTRIČNOG VOZILA	4
2.1 POTROŠNJA ENERGIJE S OBZIROM NA KONFIGURACIJU TERENA.....	5
2.1.1 Sile na kosini	6
2.1.2 Simuliranje vožnje na kosini	10
2.2 AGM AKUMULATORI.....	12
2.3 OSVRT NA DOSADAŠNJA ISTRAŽIVANJA	14
3. ARDUINO RAZVOJNA PLATFORMA	15
3.1 ARDUINO RAZVOJNA PLOČICA	15
3.1.1 Mikrokontroler.....	16
3.1.2 Ulazno izlazni pinovi platforme	17
3.2 ARDUINO PROGRAMSKA PODRŠKA	18
4. KORIŠTENJA OPREMA I METODE MJERENJA ENERGIJE	20
4.1 PIONEER 3-AT UMANJENI PRIKAZ ELEKTRIČNOG VOZILA	20
4.1.1 Fizičke karakteristike	20
4.1.2 Programska podrška.....	22
4.2 MJERENJE POTROŠNJE ENERGIJE ARDUINO PLATFORMOM.....	22
4.2.1 Hallova sonda	22
4.2.1.1 Hallov efekt.....	23
4.2.1.2 Princip rada Hallove sonde ACS 715	23
4.2.2 Elektronički sklop za mjerenje energije	24
4.2.2.1 Sklopovlje	25
4.2.2.2 Programska podrška za prikupljanje, obradu i ispis podataka	26
5. POSTUPAK MJERENJA I REZULTATI.....	28
5.1 MJERENJA NA KOSINAMA NA TERENU	29
5.2 MJERENJE POTROŠNJE SA SIMULACIJSKIM KOLICIMA	30
5.3 KARAKTERISTIKA PRAŽNENJA AGM AKUMULATORA	33
6. ZAKLJUČAK	35
POPIS LITERATURE	36
POPIS ILUSTRACIJA.....	39
POPIS PRILOGA	40

Razvoj tehnologije uvelike doprinosi ljudskom načinu života. Jedan od primjera su električna vozila. U sve aktualnijim raspravama o zaštiti okoliša na raznim skupovima, posebna pozornost usmjerena je prema problemu zagađenja automobila koji koriste fosilna goriva u odnosu na električne automobile koji ne emitiraju štetne plinove. Upravo iz tog razloga razvoj električnih automobila dobio je posebnu pozornost. Ekonomska potpora koju pružaju mnoge države pri kupnji takvih vozila jasno pokazuje da je električni automobil prepoznat kao jedno od kvalitetnih rješenja pri rješavanju problema zagađenja okoliša ili bar pozitivan korak naprijed pri smanjenju samog i pretjerane upotrebe fosilnih goriva.

Električni automobil pokreće električni motor koji se pogoni iz akumulatora (baterija) ugrađenih u vozilo. Potrošnjom pogonskog goriva (struje) električni automobil ne generira nikakvo zagađenje okoliša, odnosno ne emitira štetne plinove (engl. Zero Emission Vehicle). Zagađenje električnih vozila je indirektno, u smislu da se zagađenje prenosi na izvore električne energije kao što su elektrane te na odlaganje potrošenih baterija, [1]. Akumulatorske baterije ugrađene u električna vozila se nakon relativno kratkog roka (5-10 godina) moraju zamijeniti novima. Uz to baterije koje se trenutno koriste na tržištu električnih automobila često ne ispunjavaju zahtjeve korisnika u smislu ne dovoljnog dometa, odnosno ne dovoljne autonomije koje može postići električno vozilo.

Autonomija u literaturi [2], [3] ima dvojako značenje. Pojam autonomije se kod mobilnih robota definira kao mogućnost samostalnog rada u smislu da čovjek svojim postupcima ne utječe na odluke robota odnosno ne upravlja njime. S druge strane autonomija se kod električnih automobila spominje kao izraz dometa. Odnosno povećanjem autonomije se povećava domet vozila, a time i vrijeme između dva punjenja vozila.

Zbog iznesenih tvrdnji akumulatorska baterija smatra se najvećim nedostatkom električnih vozila. Autonomija električnih vozila se može povećati optimizacijom potrošnje kapaciteta akumulatora kojeg koristi. Optimizacija potrošnje energije akumulatora provodi se, sa strane prometnog inženjera, primjenom algoritama za kombinatornu optimizaciju. Kombinatorna optimizacija se definira kao matematička disciplina koja proučava probleme nalaženja ekstremnih vrijednosti funkcija definiranih na konačnom skupu, [4]. Primjenom algoritama kombinatorne optimizacije moguće je naći energetski optimalnu rutu električnog vozila, a time optimizirati potrošnju energije akumulatora ugrađenih u vozilo.

Jedan od mnogih algoritama koji se primjenjuju za optimizaciju je Dijkstrin algoritam. Dijkstrin algoritam je primarno namijenjen za pronalazak minimalnog puta između dvije točke na težinskom grafu, [5]. Gdje težine koje su pridodane svakom bridu grafa predstavljaju duljinu puta između dva vrha grafa koji predstavljaju gradove.

Modifikacijom osnovnog Dijkstrinog algoritma može se dobiti algoritam za izračun energetske optimalne rute vozila. Ako se težinama bridova pridoda potrošnja energije između dva vrha na grafu i postotak potrošnje kapaciteta akumulatora te se nad tako definiranim grafom provede Dijkstrin algoritam, moguće je dobiti energetske optimalnu rutu vozila. Ako se vozilo kreće energetske optimalnom rutom troši manje energije i time produžuje vijek akumulatora, povećava autonomiju vozila između dva punjenja i smanjuje broj ciklusa punjenja i pražnjenja akumulatora, [6]. Time se povećava udobnost putovanja električnim vozilom i doprinosi se očuvanju okoliša.

Za primjenu Dijkstrinog algoritma potrebno je prikupiti podatke o potrošnji energije električnog vozila, kako bi se mogli pridodati bridovima na težinskom grafu. U radu je provedeno istraživanje u svrhu prikupljanja podataka o potrošnji malog električnog vozila Pioneer 3-AT. Prikupljeni su podatci o potrošnji malog električnog vozila koje predstavlja mobilni robot Pioneer 3-AT.

Opći cilj ovog rada je izraditi sklopovlje i programsku podršku za prikupljanje podataka o potrošnji malog električnog vozila. Specifični cilj je izraditi sklop za mjerenje energije mobilnog robota Pioneer 3-AT, pomoću Arduino razvojne platforme te prikazati način simulacije kosine. Naslov završnog rada je: Dinamičko mjerenje energije električnog vozila s pomoću Arduino platforme. Rad je podijeljen u šest cjelina:

1. Uvod
2. Potrošnja energije električnog vozila
3. Arduino razvojna platforma
4. Korištena oprema i metode mjerenje energije
5. Postupak mjerenja i rezultati
6. Zaključak

U prvom poglavlju, definiran je problem i predmet istraživanja, određena je svrha i ciljevi mjerenja potrošnje energije malog električnog vozila. Dan je osvrt na dosadašnja istraživanja u ovome području, te je objašnjena struktura završnog rada.

U dugom poglavlju, dana je teorijska podloga vezana za potrošnju energije električnog vozila. Prikazane su sile koje djeluju na vozilo kada se kreće kosinom i terenom bez nagiba u svrhu definiranja potrošnje vozila koja se mijenja zbog utjecaja konfiguracije terena. Opisani su AGM akumulatori koji su ugrađeni u mobilnog robota koji je služio kao umanjeni prikaz električnog vozila.

U trećem poglavlju, opisana je Arduino razvojna platforma koja se koristila za mjerenje potrošnje energije vozila. Opisana je razvojna pločica, Arduino programska podrška te način spajanja i programiranja mikrokontrolera na Arduino platformi.

U četvrtom poglavlju, opisana je oprema koja je korištena pri izradi rada. Prikazan je mobilni robot Pioneer 3-AT, njegove fizičke karakteristike i programska podrška. Također, pojašnjen je način izrade sklopa za mjerenje energije, sklopovlje i programska podrška.

U petom poglavlju, prikazan je postupak i rezultati mjerenja. Mjerenjem su prikupljeni podatci o potrošnji mobilnog robota, simulaciji kosine te praznjenju AGM akumulatora.

U posljednjem poglavlju, dan je zaključak rada te mogućnosti budućih istraživanja u ovome području.

Potrošnja energije malog električnog vozila

Potrošnja energije električnog vozila definira se kao ukupna izlazna snaga sa pogonskih akumulatora po jedinici vremena ili puta. Odnosno koliko snage vozilo troši po jedinici vremena ili puta, [7].

Za mjerenje potrošnje energije potrebno je poznavati utjecaj sila na vozilo, snagu koju vozilo troši za kretanje, brzinu kojom se kretalo vozilo i vrijeme koje je potrebno vozilu da savlada određeni put. Suma sila koje savladava vozilo u pokretu:

$$\sum F = F_{kot} + F_{kos} + F_{zrak} + F_a \quad (1)$$

gdje je:

$F_{kot}[N]$ – sila otpora koja se javlja zbog kotrljanja pneumatika po podlozi

$F_{kos}[N]$ – sila otpora koja se javlja zbog utjecaja kosine

$F_{zrak}[N]$ – sila otpora koja se javlja zbog otpora zraka

$F_a[N]$ – inercijska sila

Snaga (u vatima [W]) potrebna za kretanje vozila određenom brzinom v se izražava kao:

$$P = v \cdot \sum F \quad (2)$$

Ukupna energija (u džulima [J]) potrebna za savladavanje puta između dvije točke u prostoru se izražava kao:

$$E_{uk} = t \cdot P \quad (3)$$

gdje t predstavlja vrijeme potrebno za savladavanje puta, odnosno vrijeme za koje se troši snaga iz izraza (2). Kako bi izraz (3) za ukupnu energiju bio valjan mjerenje potrošnje energije mora se vršiti sa konstantnom brzinom.

2.1 Potrošnja energije s obzirom na konfiguraciju terena

Zbog utjecaja sila koje djeluju na vozilo na kosini, povećava se potrebna snaga za savladavanje kosine, a time raste i potrošnja energije.

Energija se definira kao sposobnost tijela da izvrši rad. Izvršeni rad je promjena energije, pa se rad i energija izražavaju istom mjernom jedinicom, džul (J).

Simulacija je zamišljena tako da se uspoređi izvršeni rad vozila kada vozi uz kosinu i kada vozi po terenu bez nagiba ali sa priključenim simulacijskim kolicima koja se vuku po asfaltu. Rad koji vozilo vrši kada se vozilo kreće po terenu bez nagiba, jednak je promjeni kinetičke energije:

$$E_{k_2} - E_{k_1} = W_{r_1} - W_{tr_{kot_1}} - W_{zrak} \quad (4)$$

i kada se kreće po kosini:

$$E_{k_2} - E_{k_1} = W_{r_2} - W_{tr_{kot_2}} - W_{zrak} - \Delta W_p \quad (5)$$

gdje je:

W_{r_1}, W_{r_2} [W] – rad koji ulaže vozilo

ΔW_p [J] – rad potencijalne energije (rad koji se ulaže za svladavanje visine)

W_{zrak} [J] – rad koji se ulaže zbog djelovanja otpora zraka

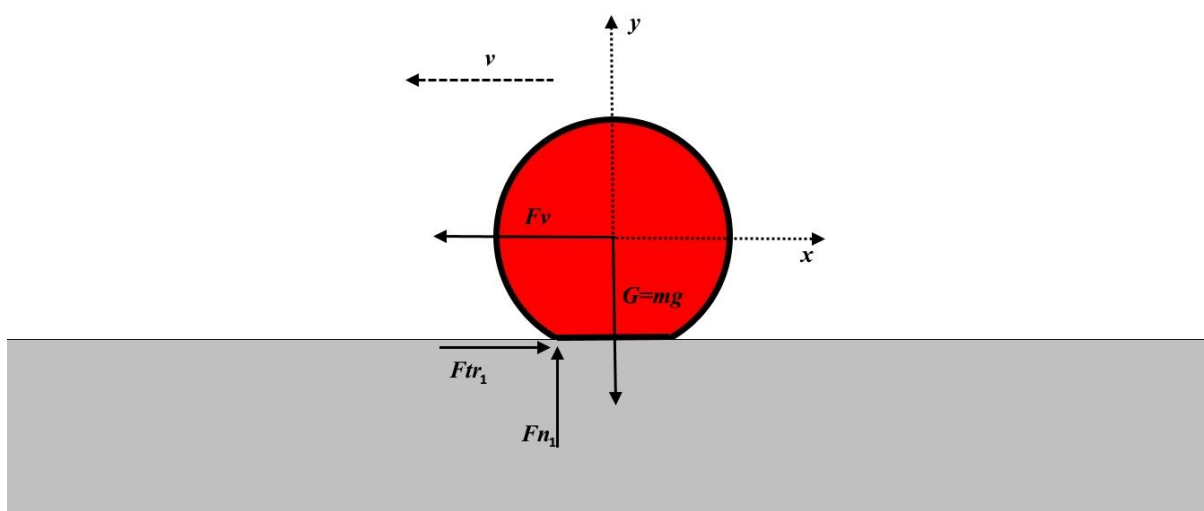
W_{kot_1}, W_{kot_2} [J] – rad koji se ulaže zbog djelovanja trenja kotrljanja

E_{k_1} [J] – kinetička energija s početnom brzinom

E_{k_2} [J] – kinetička energija s krajnjom brzinom

2.1.1 Sile na kosini

Kako bi se mogla mjeriti potrošnja energije vozila, uzimajući u obzir konfiguraciju terena, potrebno je poznavati djelovanje sila koje djeluju na vozilo kada se kreće uz kosinu, [8]. Slike 1. i 2. prikazuju sile koje djeluju na vozilo kada se kreće uz kosinu i po ravnom terenu. Na svako tijelo koje se nalazi na Zemlji djeluje gravitacijska sila G . Kako bi se vozilo kretalo mora postojati neka pogonska sila F_v , kojoj se opiru sile trenja F_{tr} , sila otpora zraka te, ako se kreće po kosini, komponenta gravitacijske sile G .



Slika 1. Sile koje djeluju na tijelo koje se kreće konstantnom brzinom po ravnom terenu

Kako bi se dobio izraz za silu trenja kotrljanja F_{kot} za kretanje po terenu bez nagiba potrebno je postaviti jednadžbu momentnog pravila za moment oko ishodišta koordinatnog sustava sa Slike 1. Moment rezultante s obzirom na neku točku jednak je sumi momenata oko te točke, [8]. S obzirom da se postavljaju analitički uvjeti ravnoteže, momentno pravilo glasi:

$$\vec{M}_O^R = \vec{r} \times \vec{R} = \vec{r} \times \sum_{i=1}^n \vec{F}_i = \vec{M}_O^{F_1} + \dots + \vec{M}_O^{F_n} = \sum \vec{M}_O^{F_i} = 0 \quad (6)$$

gdje je:

$\vec{M}_O^R$ [N/m] – moment resultantne sile R oko točke O

$\vec{r}$ [m] – udaljenost sile od točke O

$\vec{F}_i$ [N] – sila s hvatištem u točki O

Sve sile se nalaze u istoj ravnini pa vrijedi skalarna jednačba:

$$M_O^R = \sum M_O^{F_i} = 0 \quad (7)$$

Momentno pravilo za sile sa Slike 1., po izrazu (7) tada glasi:

$$\sum M_O^F = F_{kot_1} \cdot r - F_{n_1} \cdot e = 0 \quad (8)$$

gdje je:

$r [m]$ – polumjer kotača

$e [m]$ – deformacija kotača

Tada izraz za silu trenja kotrljanja F_{kot} glasi:

$$F_{kot_1} = f_k \cdot F_{n_1} \quad (9)$$

Gdje je f_k koeficijent trenja kotrljanja i iznosi $f_k = e/r$.

Postavljanjem analitičkih uvjeta ravnoteže za sile sa Slike 1. dobiti će se izraz za silu reakcije podloge F_n . Analitički uvjet ravnotežeavnog sustava sila koje se sijeku u jednoj točki, to jest algebarske sume komponenata svih sila u pravcima dviju koordinatnih osi moraju biti jednake nuli, [8]. Uvjet se prikazuje izrazima:

$$\sum_{i=1}^n F_{x_i} = 0 \quad (10)$$

$$\sum_{i=1}^n F_{y_i} = 0 \quad (11)$$

gdje izrazi $\sum_{i=1}^n F_{x_i} = 0$ i $\sum_{i=1}^n F_{y_i} = 0$ predstavljaju analitički uvjet za ostvarivanje ravnoteže, da suma svih sila koje djeluju u smjeru osi x i y mora biti jednaka nuli.

Iz uvjeta ravnoteže (11) dolazi se do izraza za silu reakcije podloge:

$$F_{n_1} = m \cdot g \quad (12)$$

gdje je:

F_n [N] – sila reakcije podloge

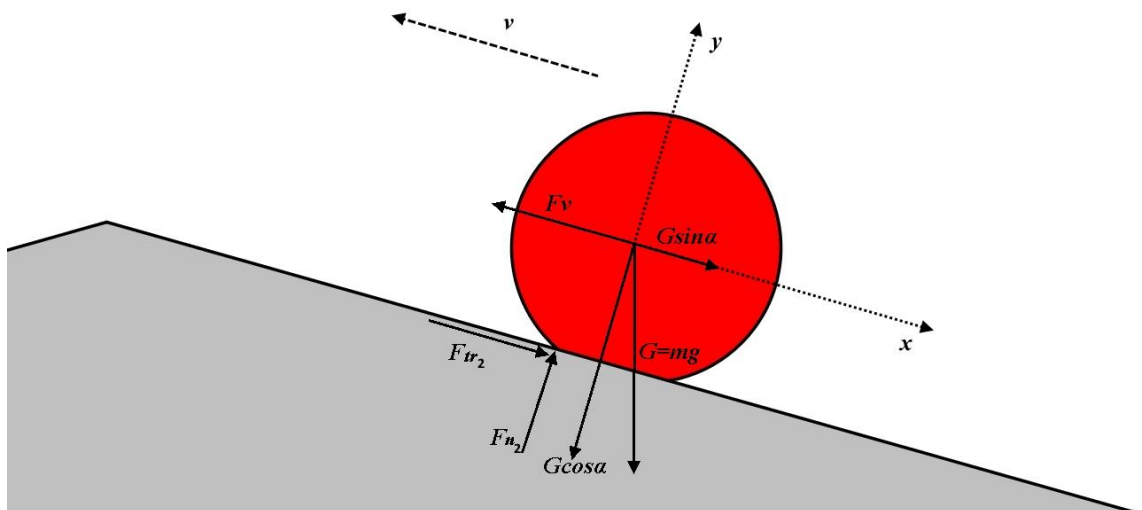
m [kg] – masa tijela

g [m/s²] – ubrzanje sile teže

Kada se u izraz (9) uvrsti izraz za silu reakcije podloge (12) dolazi se do konačne jednadžbe za silu trenja kotrljanja na terenu bez nagiba:

$$F_{kot_1} = f_k \cdot m \cdot g \quad (13)$$

Također, za izračun potrošnje energije, potrebno je poznavati i silu trenja kotrljanja na kosini. Slika 2. prikazuje sile koje djeluju na tijelo kada se kreće konstantnom brzinom uz kosinu.



Slika 2. Sile koje djeluju na tijelo koje se kreće konstantnom brzinom uz kosinu

Izraz za silu trenja kotrljanja po kosini izvodi se također iz momentnog pravila (7), sa silama prikazanim Slikom 2:

$$F_{kot_2} = f_k \cdot F_{n_2} \quad (14)$$

No, komponenta gravitacijske sile G koja se javlja zbog kretanja po kosini, mijenja silu reakcije podloge F_{n_2} . Nakon postavljanja analitičkih uvjeta ravnoteže (11) i (12) izraz za reakciju podloge, kada se vozilo kreće uz kosinu glasi:

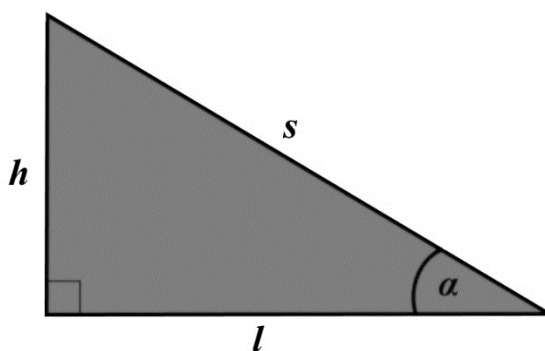
$$F_{n_2} = \cos \alpha \cdot m \cdot g \quad (15)$$

gdje je α kut nagiba terena izražen u stupnjevima ($^\circ$).

Tada izraz za trenje kotrljanja, kada se vozilo kreće uz kosinu, glasi:

$$F_{kot_2} = f_k \cdot \cos \alpha \cdot m \cdot g \quad (16)$$

Nagib terena se u literaturi [9] iskazuje kao postotak nagiba terena, a ne kao kut nagiba terena u stupnjevima. Postotak nagiba terena se računa kao tangens kuta α pravokutnog trokuta, sa Slike 3., odnosno omjer dvije katete pravokutnog trokuta.



Slika 3. Pravokutni trokut za izračun postotka nagiba terena
Izvor: [9]

Tangens kuta α iznosi:

$$\tan \alpha = \frac{h}{l} \quad (17)$$

gdje je:

h [m] – visina pravokutnog trokuta

l [m] – duljina katete pravokutnog trokuta

Izraz za postotak nagiba terena tada glasi:

$$p = \tan \alpha \quad (18)$$

2.1.2 Simuliranje vožnje na kosini

Mjerenje potrošnje mobilnog robota na kosini na terenu zahtjevan je proces. Moguće poteškoće su nabava rampe za simulaciju kosine u laboratoriju, pronalazak adekvatne uzbrdice na terenu i slično. Sukladno navedenom provedena je simulacija kosine sa simulacijskim kolicima. Na mobilni robot priključena su kolica pri kretanju terenom bez nagiba. Kolica s dodatnom težinom klizila su po tlu i svojim radom otpora, prouzrokovanim trenjem klizanja, zamijenila rad koji robot ulaže kako bi svladao potencijalnu energiju (W_p).

Nakon dodavanja težine simulacijskih kolica zakon kinetičke energije za vožnju po terenu bez nagiba glasi:

$$E_{k_2} - E_{k_1} = W_{r_1} - W_{kot_1} - W_{zrak} - W_{kol} \quad (19)$$

gdje W_{kol} predstavlja rad koji vozilo ulaže za savladavanje trenja klizanja kolica po površini. Izjednačavanjem izraza (5) i (20), uzimajući u obzir se simulacija provodi konstantom brzinom, dolazi se do izraza:

$$W_{r_1} - W_{kot_1} - W_{zrak} - W_{kol} = W_{r_2} - W_{kot_2} - W_{zrak} - \Delta W_p \quad (20)$$

Ako se pri mjerenju ostvari uvjet jednakog rada kada se vozilo kreće po kosini i po ravnom terenu, te se zanemari otpor zraka (zbog male površine mobilnog robota):

$$W_{r_1} = W_{r_2} \quad (21)$$

$$W_{zrak} = 0 \quad (22)$$

dolazi se do izraza:

$$W_{kot_1} + W_{kol} = W_{kot_2} + W_p \quad (23)$$

Rad (W) se definira kao fizikalna veličina koja opisuje djelovanje sile, određena kao umnožak sile i prijeđenog puta. U općem slučaju rad se izražava kao:

$$W = \vec{F} \times \vec{s} \quad (24)$$

Ako je sila konstantna i kut između pravca djelovanja sile i pravca puta je 0° onda vrijedi:

$$W = F \cdot s \quad (25)$$

Nakon što se u izraz (23) uvrste izrazi za rad (25) i silu (13) i (16) dobije se sljedeća jednadžba:

$$f_k \cdot m_v \cdot g \cdot s + \mu \cdot m_{kol} \cdot g \cdot s = f_k \cdot g \cdot m_v \cdot \cos \alpha \cdot s + m_v \cdot g \cdot \sin \alpha \cdot s \quad (26)$$

gdje je:

m_v [kg] – masa vozila

m_{kol} [kg] – masa simulacijskih kolica

s [m] – prijeđeni put

α [°] – nagib terena

g [m/s^2] – ubrzanje sile teže

f_k – koeficijent trenja kotrljanja

μ – koeficijent trenja klizanja

Uvrštavanjem izraza za postotak nagiba terena (18) u izraz (26), dobije se izraz za težinu simulacijskih kolica u ovisnosti o postotku nagiba terena:

$$m_{kol} = \frac{m_v \cdot (f_k \cdot \cos(\tan^{-1} p) + \sin(\tan^{-1} p) - f_k)}{f_k + \mu} \quad (27)$$

2.2 AGM akumulatori

Elektrokemijski akumulatori su uređaji koji pretvaraju električnu energiju u kemijsku za vrijeme punjenja i kemijsku energiju u električnu za vrijeme pražnjenja, putem elektrokemijske reakcije. Primjeri elektrokemijskih akumulatora su olovni akumulatori, akumulatori zasnovani na niklu te akumulatori zasnovani na litiju, [10], [11]. Akumulator se sastoji od nekoliko ćelija koje su međusobno povezane. Akumulatorska (baterijska) ćelija sastoji se od dvije elektrode (anode i katode) i elektrolita u kojega su uronjene i u pravilu proizvodi napon od 2 V.

Akumulator je određen kapacitetom (Ah) te nazivnim naponom (V). Kapacitet se definira kao broj amper sati koji se dobije kod pražnjenja akumulatora od stanja napunjenosti do stanja potpune ispražnjenosti, [12]. Pri početku pražnjenja ima veći napon od nazivnog napona koji je definirao proizvođač. Početni napon ubrzo padne na približnu vrijednost definiranog napona koji lagano pada do vrijednost na kojoj uređaj prepoznaje da je akumulator u potpunosti ispražnjen. Raspon napona akumulatora, nazivnog napona 12 V, od potpune napunjenosti do potpune ispražnjenosti je približno 2 V.

Suhoolovni akumulatori imaju dvije olovne ploče kao i start akumulatori u automobilima, samo što između ploča nije tekuća kiselina već je elektrolit u obliku gela. Postoje dva tipa suhoolovnih akumulatora- gel tip i AGM (engl. Absorbent Glass Mat). Suhoolovni akumulatori imaju široko područje primjene za pospremanje električne energije. Koriste se u fotonaponskim sustavima, električnim invalidskim kolicima, sustavima za zaštitu objekata, neprekidnim sustavima napajanja i slično. Litij ionski akumulatori su noviji akumulatori i našli su primjenu posebno u elektronici (mobiteli) i u električnim automobilima zbog svojih dobrih karakteristika. Ovaj tip akumulatora ima barem jednu elektrodu izvedenu iz litija. Robot posjeduje tri suhoolovna AGM akumulatora kapaciteta 9 Ah tvrtke Werker nazivnog napona 12 V.

AGM akumulatori su olovni akumulatori. Za razliku od klasičnih olovnih imaju separatore od staklene vune koji u potpunosti upiju elektrolit. Tako upijen elektrolit zbog kapilarnog efekta nije u mogućnosti istjecati niti ako se akumulator okrene na neku drugu stranu. Predviđen vijek trajanja im je do 10 godina, [13]. Prednosti i nedostaci AGM akumulatora u odnosu na klasične mokre akumulatore prikazane su Tablicom 1.

Tablica 1. Prednosti i nedostaci AGM akumulatora u odnosu na mokri akumulator
Izvor: [14]

PREDNOSTI :	NEDOSTATCI:
– Mogu raditi neovisno u kojem su položaju se nalaze	– Veća cijena
– Povećana otpornost na vibracije	– Ne trpe prepunjenje
– Veća specifična snaga	– Za produljivanje vijeka trajanja ne smije ih se prazniti iznad 50%
– Nije potrebno nadolijevanje vode	
– Elektrolit je upijen u separatoru	
– Mogućnost rada ispod -18°C	
– Mali unutarnji otpor dopušta im brzo punjenje i pražnjenje	

Ciklusi punjenja i pražnjenja, starenje, temperatura te dugo skladištenje imaju velik utjecaj na ukupan kapacitet akumulatora. Poznavanje akumulatorskih parametara može pomoći pri produljenju vijeka trajanja ili optimizaciji rute vozila, [15].

Stanje napunjenosti akumulatora (engl. State of Charge - SOC) definira se kao dostupni naboj pohranjen u akumulatoru u odnosu na naboj potpuno ispražnjenog akumulatora. Procjena ovog parametra važna je zbog optimalnog upravljanja sustavom napajanja električnog vozila. SOC poprima vrijednosti u intervalu (0,1), [11]. SOC je definiran kao:

$$SOC(\tau) = SOC(t_0) - \frac{1}{K_{p_{nom}}} \int_{t_0}^{\tau} i(\tau) d(\tau) \quad (28)$$

gdje je:

$SOC(t_0)$ - početno stanje napunjenosti akumulatora

$K_{p_{nom}}$ [Ah] - normalni kapacitet potpuno napunjenog akumulatora

$i(\tau)$ [A] - struja pražnjenja

2.3 Osvrt na dosadašnja istraživanja

Istraživanja [16], [3], [6], [17] i [18] bave se tematikom optimiziranja potrošnje energije električnih i hibridnih vozila. Istraživanje [16] predlaže optimizaciju izradom profila brzina koji bi smanjili potrošnju električne energije. U [18] prikazuju se energetske optimalne rute s dodatnim uvjetom da vozilo neće stići na odredište, već da mora stati na punionicu. Istraživanja [3], [6] i [17] predlažu primjenu Dijkstrinog algoritma za rješavanje problema energetske optimalne rute, no navode kako se problem optimizacije potrošnje energije ne može riješiti klasičnim Dijkstrinim algoritmom. Razlog tomu je što za funkcioniranje Dijkstrinog algoritma energetske težine pridodane grafu ne smiju biti negativne vrijednosti, koje bi se pojavile zbog regenerativnog kočenja. U ovome radu optimizirala se potrošnja mobilnog robota koji nema opciju regenerativnog kočenja pa se to ograničenje nije uzelo u obzir. Također u istraživanju [3] se u proračun potrošnje energije uzelo u obzir i zagušenje prometne mreže, energetske se optimizirala ruta i po pitanju koliko vozilo troši zbog stani-kreni vožnje. U istraživanju [6] takvo ograničenje zanemaruje se i algoritam se primjenjuje na težinskom grafu koji posjeduje samo zračne udaljenosti. Na takav način postavljen je i graf u ovome radu.

Arduino razvojna platforma

Arduino je razvojna platforma otvorenog koda (engl. Open-source) koja se koristi za programiranje mikrokontrolera i testiranje elektroničkih sklopova, [19].

Ovakva razvojna platforma omogućuje: obradu podataka sa senzora, komunikaciju stroj prema stroju (engl. Machine to Machine Communication), upravljanje malim lokalnim sustavom, dijeljenje podataka na internetu. Spomenute funkcije ovise o vrsti razvojne platforme, procesorskoj snazi, dostupnoj memoriji, potrebnoj radnoj snazi platforme i sličnim ograničenjima.

Postoje više vrsta razvojnih platformi koje se razlikuju po veličini i po namjeni. Osnovne verzije platformi su Arduino Uno, Arduino 101, Arduino Pro. Poboljšane verzije posjeduju sve funkcije osnovnih verzija uz neke dodatke u funkcionalnosti (Arduino Due, Arduino Mega). Također postoji i posebna verzija platforme koja se koristi za 3D ispis, Materia 101.

Osim same razvojne platforme Arduino nudi i cijeli niz dodataka koji služe za nadogradnju osnovne platforme, a nazivaju se *shield*. *Shield* služi za proširenje funkcionalnosti platforme u vidu mogućnosti spajanja na WiFi mrežu, Ethernet ili GSM komunikaciju, korištenje dodatne memorije sa SD karticom, kontrolu električnih motora i slično.

Arduino razvojna platforma može se podijeliti u dva dijela: Arduino razvojna pločica (engl. board) i Arduino programska podrška IDE (engl. Integrated Development Environment).

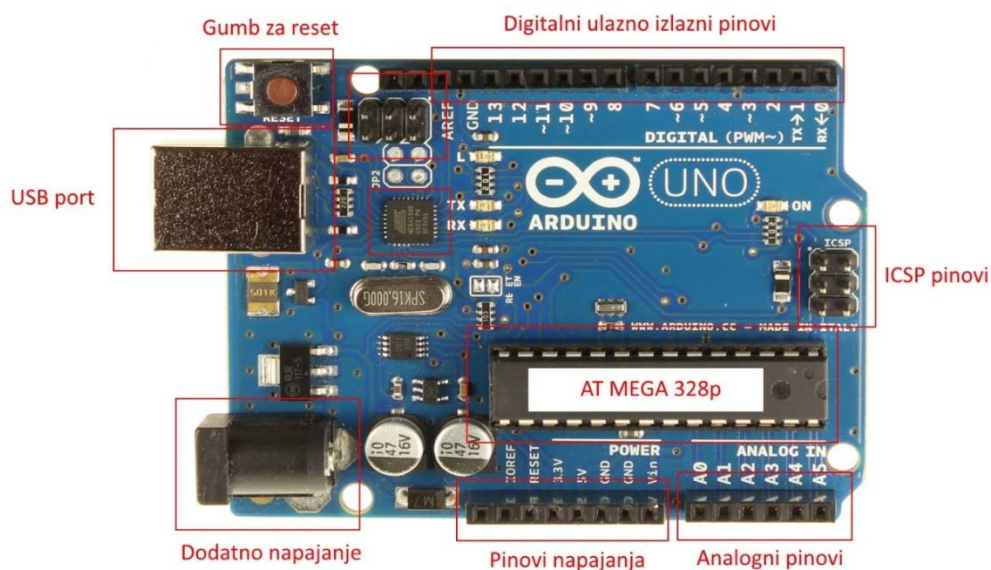
3.1 Arduino razvojna pločica

Razvojna pločica se sastoji od:

- USB priključka – za komunikaciju sa računalom, omogućuje programiranje i napajanje ugrađenog mikrokontrolera,
- Utor za napajanje – služi za napajanje pločice ako USB priključak nije dovoljan za napajanje (kontrola motora ili sl.)
- Dugme za reset – vraća razvojnu pločicu u početno stanje
- Mikrokontroler – programabilni uređaj koji prima i obrađuje signale sa analognih i digitalnih pinova
- Analogni pinovi – pinovi na koje se spaja signal u analognom obliku
- Digitalni pinovi – ulazno/izlazni pinovi koji rade sa digitalnim signalima
- ICP (engl. In Circuit Programmer) – pinovi koji omogućuju programiranje mikrokontrolera
- GND pinovi – pinovi za uzemljenje

- Pinovi snage – pinovi od 3,3 V i 5 V

Slika 4. prikazuje osnovne dijelove Arduino platforme na Arduino Uno platformi.



Slika 4. Osnovni dijelovi i razmještaj pinova Arduino platforme
Izvor: [20]

3.1.1 Mikrokontroler

Mikrokontroler je programabilni uređaj koji prima i obrađuje signale sa analognih i digitalnih pinova. Osnovna razlika između mikrokontrolera i mikroprocesora je funkcionalnost. Mikroprocesor postaje funkcionalan uz dodavanje ostalih komponenti kao što su memorija ili komponente za prijem i slanje podataka. Mikrokontroler je dizajniran tako da su te komponente već integrirane u sam uređaj, odnosno sve se periferne jedinice nalaze u jednom uređaju.

Arduino razvojne platforme koriste mikrokontrolere ATMEGA arhitekture. Platforme Uno, Nano, Mini 05, Mega 2560, Leonardo, Micro, Robot i Esplora koriste ATMEGA mikrokontrolere sa AVR arhitekturom.

AVR arhitektura je uvedena na tržište tokom 1997. godine od strane ATMEL-a. To su 8-bitni RISC (engl. Reduced Instruction Set Computer) [21] mikrokontroleri koji sadrže 32 registra opće namjene te bogat set instrukcija. AVR mikrokontrolere odlikuje brzina izvršavanja do 16 MIPS (engl. Millions of Instructions Per Second) odnosno rade na taktu od 16 MHz, [22].

Vrlo važna stavka kod programiranja mikrokontrolera su njegova ograničenja u vidu dostupne SRAM (engl. Static Random Access Memory) i EEPROM (engl. Electrically Erasable Programmable Read-Only Memory) memorije. SRAM se koristi za izvršavanje programa, a EEPROM predstavlja memoriju za trajno spremanje podataka.

Zbog ograničenja opisanih razvojnih pločica, proizvedene su i pločice sa proširenim funkcionalnostima i većim performansama. Arduino Due razvojna pločica koristi Atmel-ov mikroprocesor SAM3X8E sa 32-bitnim ARM Cortex-M3 procesorom koji posjeduje radni takt od 84 MHz. Također Due posjeduje mnogo više SRAM memorije. Usporedba razvojnih platformi Uno i Due prikazana je Tablicom 2.

Tablica 2. Karakteristike Arduino Uno i Due platformi
Izvor: [19]

	ARDUINO UNO	ARDUINO DUE
ANALOGNI ULAZ	6, 10-bitni	12, 10-bitni
ANALOGNI IZLAZ	6, PWM, 8-bitni	2, DAC, 10-bitni
DIGITALNI ULAZ/IZLAZ	14, 6 PWM, 8-bitni	54, 12 PWM 8-bitni
MIKROKONTROLER	ATMega 328	AT91SAM3X8E
RADNI TAKT	16 MHz	84 MHz
FLASH MEMORIJA	32 KB	512 KB
SRAM	2 KB	96 KB
EEPROM	1 KB	-

3.1.2 Ulazno izlazni pinovi platforme

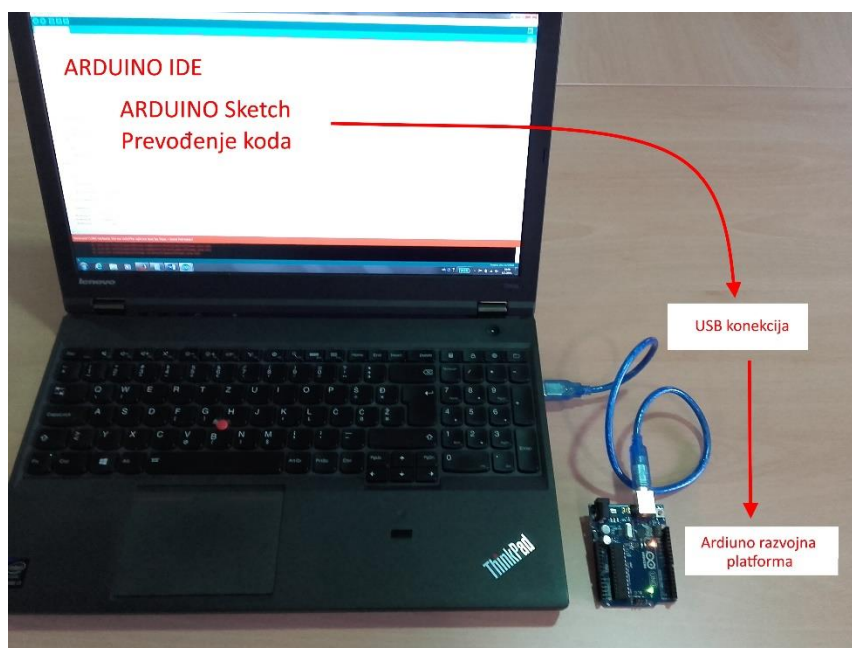
Sve Arduino razvojne platforme posjeduju ulazno/izlazne pinove opće namjene (engl. General Purpose Inputs and Outputs - GPIO). GPIO se sastoji od dva osnovna tipa ulazno izlaznih pinova, digitalni i analogni. Analogni ulazni pinovi služe za mjerenje napona, a broj pinova i raspon reprezentiranja napona ovisi o tipu pločice koji se koristi. Analogni izlaz može se implementirati preko PWM (engl. Pulse Width Modulation) digitalnih pinova ili preko dodatnog analognog- digitalnog pretvornika. Digitalni pinovi mogu se koristiti kao ulazni ili izlazni pinovi. Za programiranje mikrokontrolera koristi se Arduino IDE razvojna okolina koja je instalirana na računalo. Računalo se povezuje sa razvojnom pločicom preko USB protokola. Ostali podržani načini komunikacije su UART (engl. Universal Asynchronous Receiver Transmitter), TTL (engl. Transistor Transistor Logic), I²C (engl. Internal IC)/TWI (engl. Two Wire Interface) i SPI (engl. Serial Peripheral Interface). SPI se koristi za spajanje raznih *shield* pločica za nadogradnju i omogućuje vrlo brzu komunikaciju. UART je hardverski implementiran u razvojnu platformu na digitalnim pinovima 0 i 1 za USB protokol. TWI se

također koristi za komunikaciju između *shield* pločice i Arduino platforme te sa ostalim uređajima. Za korištenje tog protokola koristi se ugrađena biblioteka naredbi pod nazivom *Wire*, [23].

3.2 Arduino programska podrška

Mikrokontroler na Arduino platformi može se programirati na više načina. U ovome radu opisati će se programiranje u Arduino IDE okruženju koje je slično C/C++ jeziku. Programirati se vrši i preko *JavaScript* i *Johnny Five* programske podrške (engl. framework) te korištenjem *Sysfs* alata na pločicama sa Linux operacijskim sustavom, [23].

IDE programiran je u programskom jeziku Java, kompatibilan je sa svim Arduino pločicama. IDE se koristi za pisanje, provjeru, kompajliranje (prevođenje) i prijenos koda, preko USB protokola, na razvojnu pločicu. Način spajanja Arduino platforme i računala prikazuje Slika 5. Kreirani kod naziva se *sketch* i sprema se u .ino formatu.

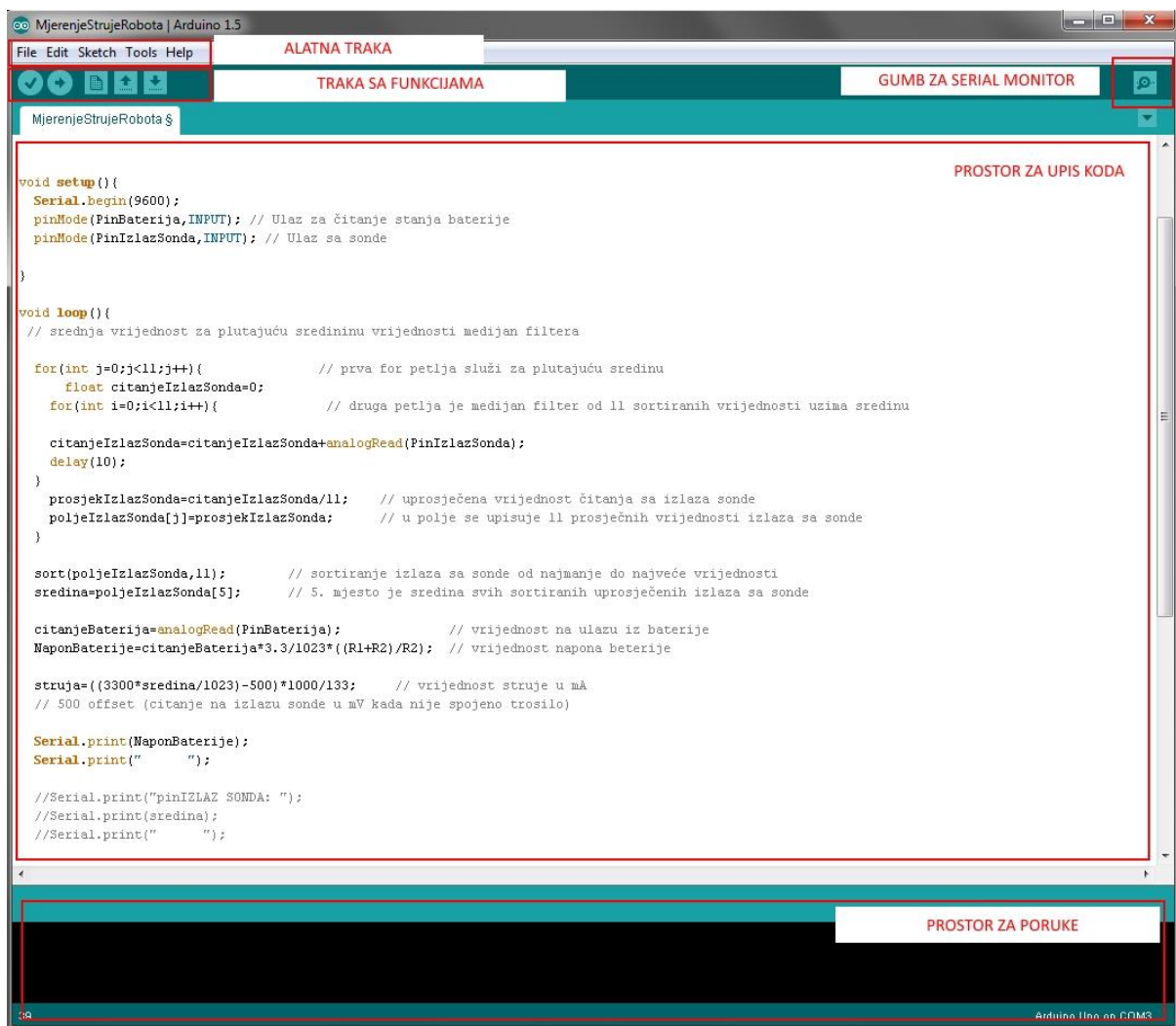


Slika 5. Spajanje Arduino platforme sa računalom sa USB protokolom

Uz IDE je pripremljeno mnogo primjera za učenje i pokazne vježbe rada na Arduino platformi. IDE se sastoji od nekoliko dijelova:

- Alatna traka – sadrži osnovne padajuće izbornike *File*, *Edit*, *Sketch*, *Tools* i *Help*
- Traka sa funkcijama – sadrži funkcije kako slijede *Verify* (provjera ispravnosti koda), *Upload* (prijenos koda na mikrokontroler), *New*, *Open* i *Save (sketch)*
- Gumb za *Serial Monitor* – otvara sučelje za prikaz serijske komunikacije između računala i Arduino razvojne platforme
- Prostor za unos koda
- Prostor poruke – prikazuje eventualne sintaktičke pogreške u kodu, prikazuje process kompajliranja (prevođenja koda za platformu)

Slika 6. prikazuje osnovne dijelove Arduino IDE.



Slika 6. Arduino programska podrška

Korištena oprema i metode mjerenja energije

Za terenska i laboratorijska mjerenja koristila se oprema Laboratorija za inteligentne transportne sustave Fakulteta prometnih znanosti Sveučilišta u Zagrebu. Mobilni robot *Pioneer 3 – AT*¹ korišten je kao umanjeni prikaz električnog vozila. Arduino Due razvojnom platformom se mjerila potrošnja energije mobilnog robota. Računalo korišteno za upravljanje mobilnim robotom, prikupljanje i obradu podataka je Lenovo TinkPad *T540* sa 8 GB RAM memorije, 64-bitnim *i7* procesorom četvrte generacije, grafičkom karticom GeForce *GT730M* te Windows 7 operativnim sustavom².

4.1 Pioneer 3-AT umanjeni prikaz električnog vozila

U ovom radu umanjeni prikaz autonomnog električnog vozila mobilni je robot Pioneer 3-AT. Mobilni robot Pioneer 3-AT, ili skraćeno P3-AT, potječe iz porodice malih i inteligentnih robota, koji imaju pogon na dva ili četiri kotača. Arhitekturu Pioneer skupine robota razvio je dr.sc. Kurt Konolige sa sveučilišta Stanford. Pioneer platforma pogodna je kao umanjeni prikaz električnog vozila zbog toga što posjeduje osnovne značajke svakog električnog vozila: za pogon koristi elektromotore, za napajanje koristi akumulatore koji su izvor istosmjerne struje.

4.1.1 Fizičke karakteristike

Glavni dijelovi P3-AT mobilnog robota su: kućište, nos, konzola, paluba, pogonski kotači, mreža sonara, razni priključci te kontrolne lampice. Konzola i paluba čine dio robota, u koji se mogu ugraditi različiti uređaji ili osjetila robota. Unutar aluminijskog kućišta nalaze se akumulator, pogonski motori, elektronika i druge standardne komponente poput prednjih i stražnjih sonara i odbojnika, [24]. Na Slici 7. prikazane su osnovne fizičke komponente. Za kontrolu mobilnog robota zadužen je 32-bitni mikrokontroler tvrtke *Renesas SH2 – 7144 RISC* s ARCOS (engl. Advanced Robot Control and Operations Software) programskim paketom kao i Pioneer SDK (engl. Software Development Kit) napredni razvojni programski paket, [25]. Akumulatori/baterije mogu pogoniti robota 2 do 4 h bez dodatnih pribora i osjetila.

¹ Mobilni robot Pioneer 3-AT nabavljen je iz tri projekta Sveučilišta u Zagrebu: Optimizacija ruta malih električnih vozila uz kriterij najmanje potrošnje energije, Optimalna navigacija malih električnih vozila u unutarnjim prostorima, Optimiranje ruta flote vozila primjenom više-agentskih optimizacijskih algoritama i stvarno-vremenskih prometnih podataka, 2013-ZUID-2

² Prijenosno računalo nabavljeno je u projektu SORDITO - Sustav za optimizaciju ruta u dinamičkom transportnom okruženju RC.2.2.08-0022, koji je sufinanciran sredstvima Europske unije iz Europskog fonda za regionalni razvoj

8 prednjih i 8 stražnjih sonara ugrađenih na robota moguće je koristiti za detekciju objekata koji su udaljeni od 15 cm do 7 m. Maksimalna brzina robota iznosi 0.75 m/s , a nosivosti je do 40 kg ovisno o terenu na kojem se nalazi, [25]. Za mjerenje brzine razvijene na kotačima robot koristi dva enkodera, jedan za lijevi i jedan za desni kotač. Kako bi se kompenzirale pogreške u mjerenju uzrokovane proklizavanjem kotača moguće je korigirati enkodere s obzirom na krajnju točku.



Slika 7. Fizičke komponente Pioneer 3-AT mobilnog robota

Za pokretanje koristi četiri elektromotora koji zupčastim remenom prenose snagu na kotače. S obzirom na os simetrije robota dva elektromotora pogone lijevu stranu robota, a dva desnu. Od dva elektromotora jedan je glavni i na sebi ima enkoder, a drugi je sporedni. Tablica 3. prikazuje ostale specifikacije Pioneer platforme.

Tablica 3. Specifikacije mobilnog robota Pioneer 3-AT
Izvor: [25]

TEŽINA (BEZ AKUMULATORA I DODATNE OPREME)	14 kg
TEŽINA (S ALUMINIJSKIM STALKOM I PRIJENOSNIM RAČUNALOM)	33 kg
NOSIVOST	40 kg
MAKSIMALNA BRZINA	0,75 m/s
MAKSIMALNI IZNOS NAGIBA TERENA	40%
VRIJEME TRAJANJA RADA	2-3 h
VRIJEME PUNJENJA BATERIJA	6 h
BROJ BATERIJA	3
KAPACITET SVAKE BATERIJE	9 Ah
NAZIVNI NAPON BATERIJE	12 V
VRSTA BATERIJE	Olovne baterije

4.1.2 Programska podrška

Programska podrška sastoji se od niza programa koji omogućuju komunikaciju s robotom putem napisane biblioteke naredbi. Predstavlja svojevrsan jezik koji omogućuje davanje naredbi kojima se upravlja robotom te dolazi s kompletnom programskom podrškom. MobileRobots programska platforma uključuje Pioneer SDK, kompletni set programa i datoteka koji omogućavaju razvoj projekata.

4.2 Mjerenje potrošnje energije Arduino platformom

Iz izraza (2) i (3) vidljivo je da je mjerenje energije mobilnog robota potrebno zapisivati struju, napon i vrijeme. Tim parametrima može se izračunati snaga i energija koju troši mobilni robot. Potreba za korištenjem Arduino platforme nastala je iz razloga što model mobilnog robota koji se koristio nema funkciju zapisivanja struje koju troši.

4.2.1 Hallova sonda

Hallova sonda je elektronički sklop, izveden kao integrirani krug, koji radi na principu Hallovog efekta. U nastavku se opisuje Hallov efekt i senzor korišten za izradu sklopa za mjerenje energije.

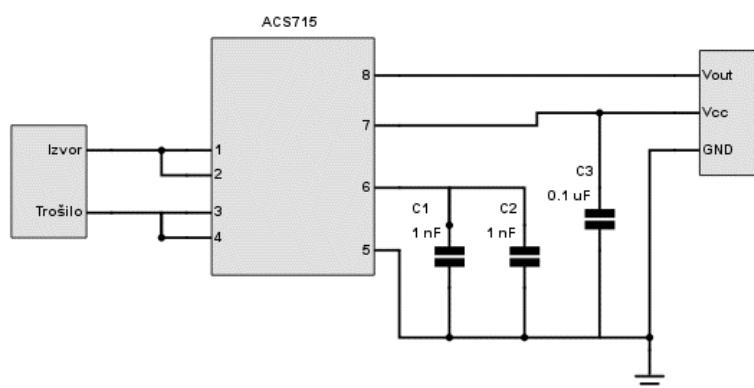
4.2.1.1 Hallov efekt

Hall-ov efekt otkriven je 1879. godine od strane američkog fizičara Edwina Halla. Hallov-efekt se opisuje kao pojava prouzročena Lorentzovom silom pri kojoj se u magnetskome polju u tankoj metalnoj ili poluvodičkoj pločici priključenoj u krug istosmjerne struje pojavljuju karakteristični napon i struja, [26], [27]. Predznak Hall-ovog napona može odrediti predznak nosilaca naboja te je na taj način otkriveno da u metalima struju prenose negativno nabijeni elektroni. U poluvodičima Hallov napon može biti ili pozitivan ili negativan, ovisno o tome jesu li poluvodiči p ili n tipa. Hallov napon se u nekim materijalima mijenja predznak zbog utjecaja promjene temperature, [28].

Primjena Hallovog efekta porasla je razvojem poluvodičke tehnologije. Senzori koji rade na principu Hallovog efekta nalaze široku primjenu u mnogim uređajima. Takvi senzori koriste se za mjerenje struje, brzine rotacije, kao senzori težine itd.

4.2.1.2 Princip rada Hallove sonde ACS 715

Sonda ACS 715 radi na principu Hallovog efekta te posjeduje linearnu strujno-naponsku karakteristiku. Struja koja prolazi kroz sondu stvara inducirano magnetsko polje čija gustoća određuje iznos napona na izlazu, [29]. Sonda je izvedena kao integrirani sklop od osam pinova. Slika 8. prikazuje senzor koji se koristio u radu, raspored i funkciju pinova Hallove sonde i cijelog senzora.



Slika 8. Shema ožičavanja Hallove sonde ACS715
Izvor: [30]

Pinovi 1 i 2 integriranog sklopa ACS715 kratko spojeni su i na njih se spaja pozitivan pol sa izvora napajanja. Pinovi 3 i 4 također su kratkospojeni i na njih se spaja pozitivni pol sa trošila. Pin 5 predstavlja uzemljenje. Pin 6 predstavlja filter koji smanjuje šum. Pin 7 je izlazni pin koji daje vrijednost Hallovog napona. Pin 8 služi za napajanje sklopa od 5 V.

Struja se mjeri na način da se od očitano Hallovog napona sa pina 7 oduzme napon praznog hoda senzora (500 mV) i dobiveni iznos napona podijeli sa konstantom α . α je konstanta definirana od proizvođača senzora, a predstavlja iznos s kojim se ostvaruje pretvorba mili volti u ampere (133 mV = 1 A), [30].

$$I_{IZL} = \frac{U_{IZ} - 500}{\alpha} \quad (29)$$

gdje je:

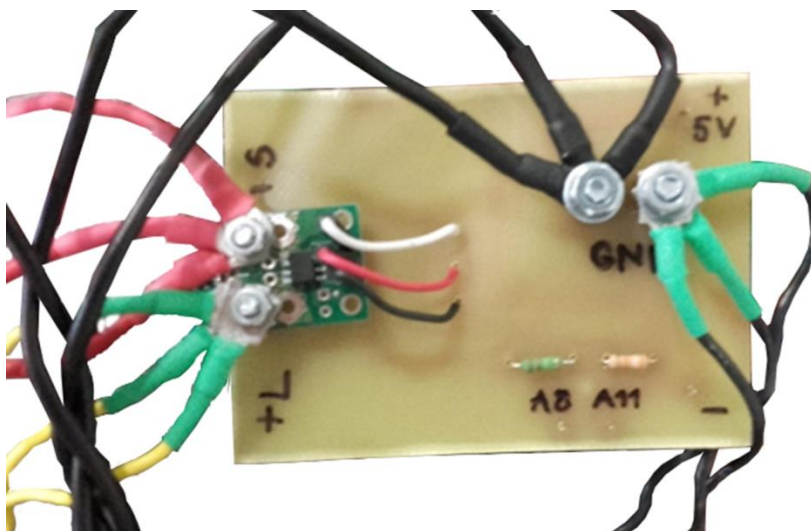
I_{IZL} [A] – mjerena struja

U_{IZ} [mV] – Hallov napon

α – konstanta pretvorbe (133)

4.2.2 Elektronički sklop za mjerenje energije

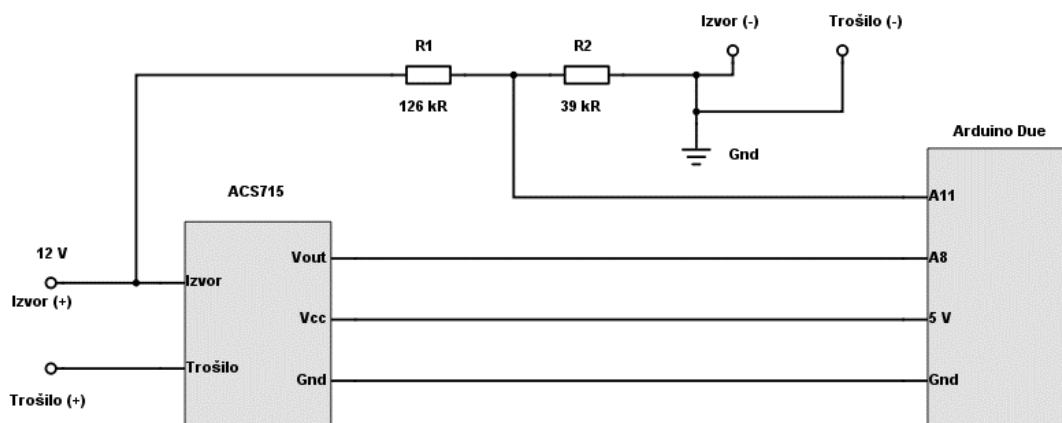
Kako bi se mjerila potrošnja električnog vozila potrebno je posjedovati neku vrtu sklopa za mjerenje snage, iz čega proizlazi mjerenje energije. Sklop je izveden u obliku *Shielda* radi lakšeg ugrađivanja u mobilnog robota. Slika 9. prikazuje izrađeni sklop. U nastavku se opisuje izrađeno sklopovlje (engl. hardware) i program za prikupljanje, obradu i ispis podataka (engl. software).



Slika 9. Elektronički sklop za mjerenje energije

4.2.2.1 Sklopovlje

Sklopovlje se sastoji od tri dijela: senzora sa Hallovom sondom ACS715, naponskog dijelila i Arduino Due razvojne platforme. Senzor je korišten za detekciju struje koju troši mobilni robot, naponsko dijelilo za mjerenje napona baterije i Arduino Due razvojna platforma za prikupljanje podataka te izradu programa pomoću IDE. Shema sklopa prikazana je slikom 10.



Slika 10. Shema sklopa za mjerenje energije

Na pin *Izvor* (engl. Source) integriranog sklopa ACS715 dovodi se pozitivni (+) pol s izvora napajanja. Na pin *Trošilo* (engl. Load) dovodi se pozitivni (+) pol koji predstavlja trošilo. U ovom slučaju trošilo je mobilni robot. Na pin *Gnd* dovode se negativni (-) polovi s trošila, izvora napajanja, sonde i Arduino platforme. Sonda se napaja preko pina *Vcc* s 5 V koje crpi iz Arduino mikrokontrolera. Sklop je izveden tako da mjeri promjenu napona i struje s vremenom uzorkovanja od jedne milisekunde.

Naponsko dijelilo je potrebno iz razloga što je nazivni napon baterija mobilnog robota 12 V, a operativni napon Arduino Due platforme 3,3 V. Ono služi kako bi smanjilo nazivni napon baterija na operativni napon platforme. Iznosi otpornika potrebnih za implementaciju naponskog dijelila računa se iz odnosa struja u naponskom djelilu:

$$\frac{U_{ARD}}{R_2} = \frac{U_{BAT}}{R_1 + R_2} \quad (30)$$

gdje je:

U_{ARD} [V] – operativni napon Arduino Due platforme

U_{BAT} [V] – nazivni napon akumulatora

R_1, R_2 [Ω] – otpornici naponskog dijelila

4.2.2.2 Programska podrška za prikupljanje, obradu i ispis podataka

Programska podrška za prikupljanje, obradu i ispis podataka se sastoji od dva dijela: program pisan u Arduino IDE okruženju i Windows Form aplikacija pisana u Microsoft Visual Studio okruženju C# jezikom.

Prvi program definira funkcionalnosti mikrokontrolera, služi za prikupljanje podataka o struji koju robot troši za pokretanje i napona akumulatora te ih ispisuje na COM port. Struja se računa po izrazu (29), a napon po izrazu (30). Zbog točnosti mjerenja struje, sukcesivno se računa aritmetička sredina od 10 očitanih vrijednosti sa analognog pina A11 u proteklih 10 milisekundi te se takvih 11 aritmetičkih sredina upisuje u polje. Nakon toga medijan filtrom izvlači se srednja vrijednost sortiranih aritmetičkih sredina. Dio algoritma koji izvršava navedeno prikazan je Slikom 11. Arduino program je u cijelosti prikazan u prilogu 1.

```
for(int j=0;j<10;j++){
    // prva for petlja služi za računanje prosjeka 10 učitanih vrijednosti i spremanje u polje

    float citanjeIzlazSonda=0;
    // postavljanje varijable u 0 kako bi se za svaku iteraciju zbrajala slijedeća očitavanja

    for(int i=0;i<11;i++)
    {
        // druga petnja služi za zbrajanje očitavanja sa sonde

        citanjeIzlazSonda=citanjeIzlazSonda+analogRead(PinIzlazSonda);
        delay(9);
    }
    prosjekIzlazSonda=citanjeIzlazSonda/10;    // prosječna vrijednost 10 čitanja sa izlaza sonde
    poljeIzlazSonda[j]=prosjekIzlazSonda;    // u polje se upisuje 11 prosječnih vrijednosti izlaza sa sonde
}
sort(poljeIzlazSonda,11);    // sortiranje izlaza sa sonde od najmanje do najveće vrijednosti
sredina=poljeIzlazSonda[5];    // 5. mjesto u polju je sredina svih sortiranih prosječnih izlaza sa sonde

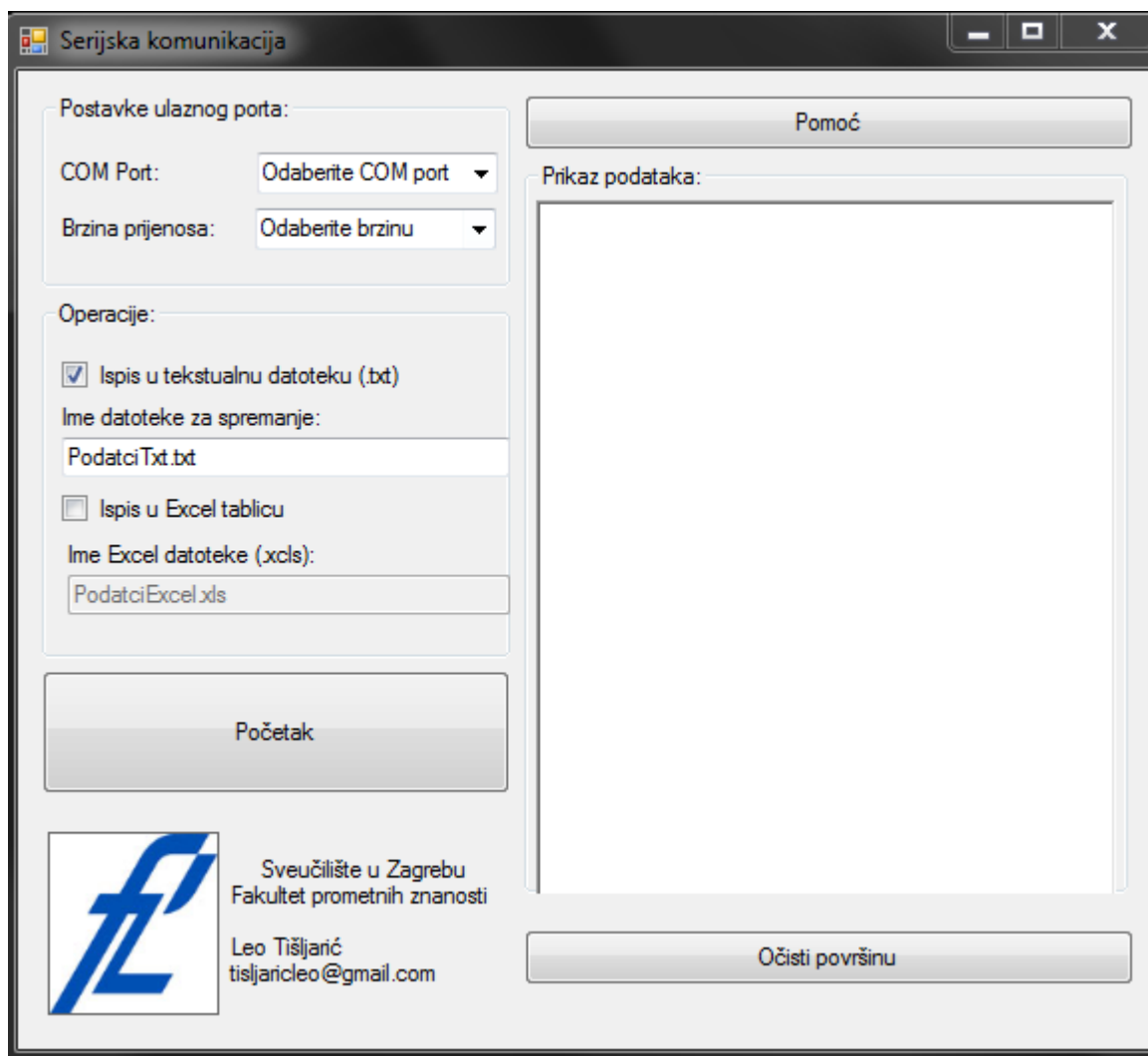
citanjeBaterija=analogRead(PinBaterija);
naponBaterije=citanjeBaterija*3.3/1024*((R1+R2)/R2);    // vrijednost napona baterije u voltima

struja=(500-(3300*sredina/1024))/133;    // vrijednost struje u amperima

Serial.print(struja);
Serial.print(" ");
Serial.println(naponBaterije);
```

Slika 11. Dio Arduino koda koji služi za obradu podataka

Program pisan u C# primarno je napisan zbog razloga što Arduino IDE ne može zapisivati rezultate u tekstualnu datoteku. Aplikacija ima mogućnost ispisa podataka koji se prikupljaju na tri načina: u prozor za prikaz podataka na aplikaciji, tekstualnu datoteku ili Microsoft Excel datoteku. Slika 12. prikazuje izgled aplikacije, a u prilogu 2 se nalazi izvorni kod aplikacije.

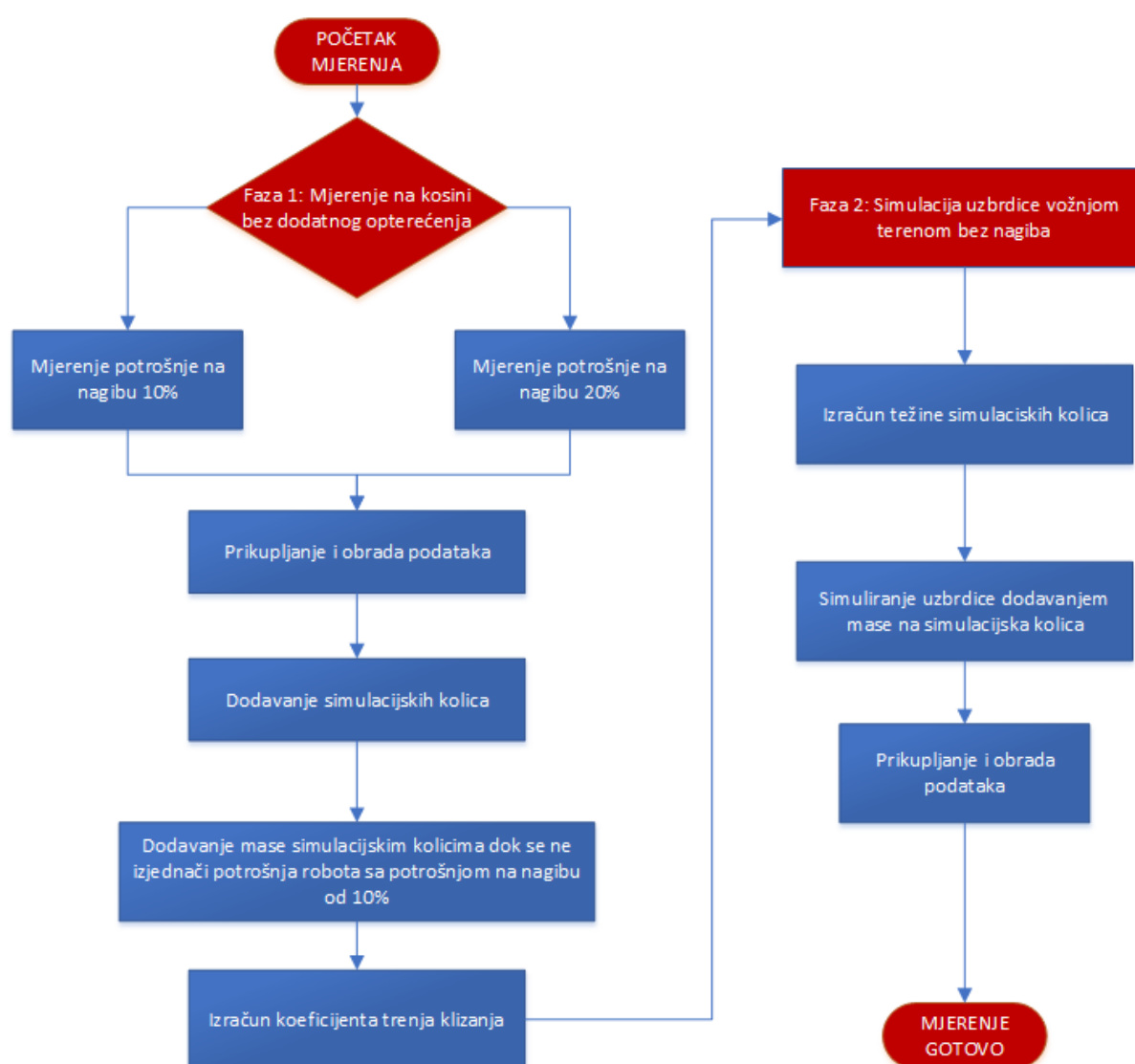


Slika 12. Aplikacija za prikupljanje podataka sa COM porta

U padajućem izborniku *COM Port* odabire se port s kojega se žele čitati podatci. U padajućem izborniku *Brzina prijenosa* odabire se brzina prijenosa podataka (engl. Baud Rate) za serijsku komunikaciju. Označavanjem kvadratića *Ispis u tekstualnu datoteku* omogućava se upis imena tekstualne datoteke u koju će se upisati podatci, uz napomenu da ime datoteke mora sadržavati ekstenziju .txt. Označavanjem kvadratića *Ispis u Excel tablicu* omogućava se upis imena Excel datoteke u koju će se upisati podatci, uz napomenu da ime datoteke mora sadržavati ekstenziju .xls. Klikom na gumb *Početak* pokreće se upis podataka u označenu datoteku i na zaslon aplikacije. Klikom na gumb *Očisti površinu*, sadržaj površine za prikaz podataka na aplikaciji se briše.

Postupak mjerenja i rezultati

Terenska i laboratorijska mjerenja vršena su zbog potrebe prikupljanja podataka o potrošnji energije malog električnog vozila u vidu Pioneer platforme. Mjerenja su izvođena na različitim nagibima kako bi se utvrdila potrošnja energije u ovisnosti o nagibu terena. Mjerenjima je prikazano koliko vozilo troši više energije kada vozi pri kretanju po kosini.

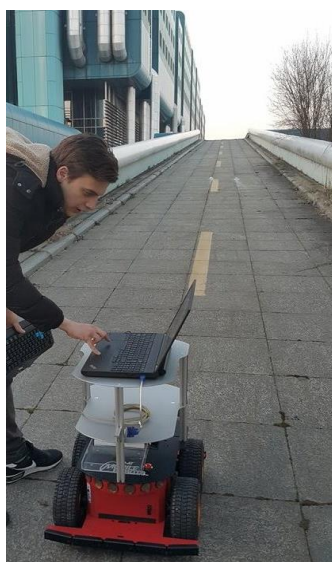


Slika 13. Dijagram toka izvođenja mjerenja

Izvedeno mjerenje može se podijeliti u dvije faze. U prvoj fazi izvođena su mjerenja na kosinama od 10 % i 20 %. Pomoću rezultata iz prve faze mjerenja, izračunat je koeficijent trenja klizanja simulacijskih kolica (μ). Druga faza sastoji se od dodavanja dodatne mase na robota pomoću simulacijskih kolica te zapisivanjem potrošnje. Masa simulacijskih kolica za simulaciju pojedinog nagiba izračunata je izrazom (32). Postupak mjerenja prikazan je dijagramom toka na Slici 13.

5.1 Mjerenja na kosinama na terenu

Terenska mjerenja na kosini izvođena su na dvije lokacije: Klinička bolnica Dubrava i Trg Dragutina Domjanića u Sesvetama. Slike 14. i 15. prikazuju mikrolokacije mjerenja.



Slika 14. Lokacija mjerenja,
nagib 10%



Slika 15. Lokacija mjerenja,
nagib 20%

Mjerenjima su prikupljeni podatci o potrošnji energije mobilnog robota na nagibu od 10 % i 20%. Mjerenja su izvođena sa različitim brzinama, 0,1 m/s, 0,4 m/s i 0,7 m/s. Zapisivala se struja i napon iz kojih je izračunata snaga. Energija se izračunala za put od 100 m. Svako mjerenje izvedeno je četiri puta kako bi se utvrdila odstupanja od mjerenja. Rezultati i odstupanja prikazani su Tablicom 4.

Tablica 4. *Potrošnja mobilnog robota na kosinama nagiba 10% i 20%*

POSTOTAK NAGIBA	BRZINA [m/s]	STRUJA [A]	SNAGA [W]	ENERGIJA [kJ]
10 %	0,1	$1,36 \pm 0,3\%$	$17,29 \pm 0,3\%$	17,290
	0,4	$2,51 \pm 0,5\%$	$31,86 \pm 0,6\%$	7,965
	0,7	$3,69 \pm 0,3\%$	$46,65 \pm 0,3\%$	6,664
20 %	0,1	$2,81 \pm 1,2\%$	$35,41 \pm 1,2\%$	35,41
	0,4	$4,88 \pm 1,2\%$	$60,94 \pm 0,9\%$	15,235
	0,7	$7,20 \pm 1,3\%$	$88,47 \pm 1,3\%$	12,638

Vozeći brzinom 0,7 m/s robot je trošio najmanje energije, iz tog razloga brzina od 0,7 m/s predstavlja energetske optimalnu brzinu mobilnog robota. Energetski optimalnom brzinom izvodila se simulacija uzbrdice.

5.2 Mjerenje potrošnje sa simulacijskim kolicima

Simulacija nagiba terena sa simulacijskim kolicima izvođena je na području Sveučilišnog kampusa Borongaj. Slika 16. prikazuje postupak mjerenja.



Slika 16. *Simulacija uzbrdice simulacijskim kolicima*

Zbog ne mogućnosti mjerenja koeficijenta trenja kotrljanja, koji se razlikuje za kretanje po ravnom terenu i po kosini, postavljen je omjer veličina radova koji se vrše pri kretanju po ravnom terenu i radova koji se vrše pri kretanju po kosini:

$$\frac{W_{kol}}{W_{kot_1}} = \frac{W_p}{W_{kot_2}} \quad (31)$$

Iz izraza (31) dolazi se do pojednostavljene formule za izračun potrebne težine simulacijskih kolica:

$$m_{kol} = \frac{m_v \cdot \sin \alpha}{\mu \cdot \cos \alpha} = \frac{m_v \cdot p}{\mu} \quad (32)$$

Koeficijent trenja klizanja μ u literaturi [31], [32], [33] poprima vrijednosti od 0,6 – 0,8. Koeficijent trenja klizanja izračunat je po formuli:

$$\mu = \frac{m_v \cdot p}{m_{kol}} \quad (33)$$

Koeficijent se mogao izračunati po navedenoj formuli iz razloga što je bila poznata potrošnja mobilnog robota na kosinama od 10% i 20%. Težina simulacijski kolica se dobila postupkom dodavanja težine na kolica sve dok se ne postigne potrošnja koja se ostvarila na kosini od 10% i 20%. Sa poznatom težinom kolica, težinom robota i postotkom nagiba terena izračunat je koeficijent trenja klizanja i iznosi 0,62.

Nakon izračuna koeficijenta trenja klizanja, provela se simulacija kosine od 10% i 20% kako bi se usporedili rezultati simulacije sa poznatim rezultatima sa stvarne uzbrdice. Usporedba rezultata simulacije i mjerenja snage na kosini prikazan je Tablicom 5.

Tablica 5. Usporedba simulacije i stvarnog mjerenja na kosini

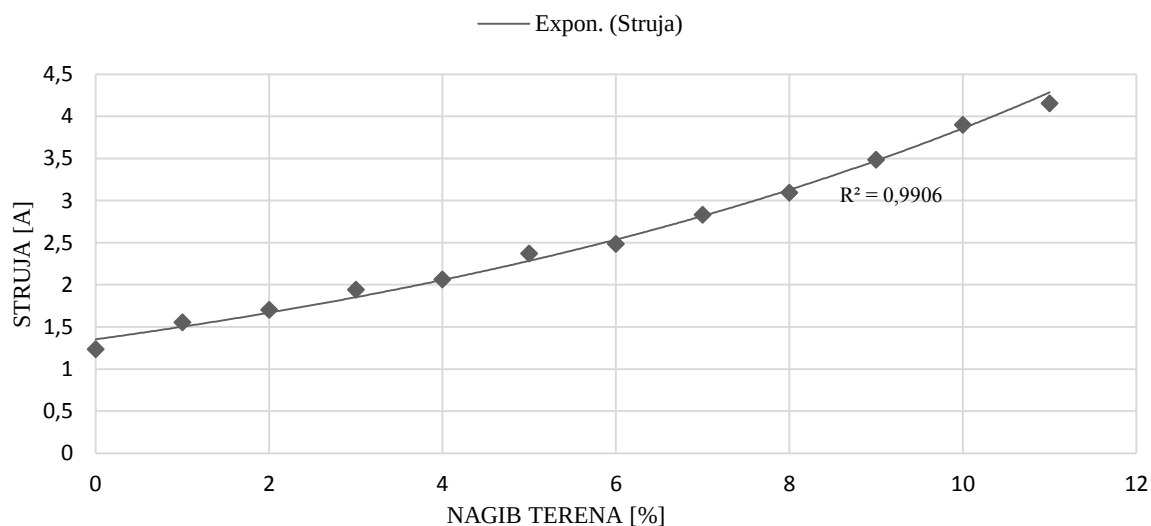
NAGIB TERENA	PARAMETRI	MJERENO NA STVARNOM NAGIBU BEZ TERETA KOLICA	SIMULIRANO S KOLICIMA NA TERENU BEZ NAGIBA	ODSTUPANJA SIMULIRANOG U ODNOSU NA STVARNO
10%	Struja [A]:	3,69 ± 0,3%	3,89 ± 1,3%	5,5 %
	Snaga [W]:	46,65 ± 0,3%	45,56 ± 1,6%	2,34 %
20%	Struja [A]:	7,2 ± 1,1%	7,4 ± 5%	2,78 %
	Snaga [W]:	88,47 ± 1,3%	91,36 ± 5,8%	3,27 %

Simulacijom je potvrđen izraz (32). Prikupljeni su podatci o potrošnji energije mobilnog robota na uzbrdicama od 1 % do 10 % sa korakom od 1 %. Svako mjerenje se također vršilo četiri puta kako bi se utvrdila odstupanja od mjerenja. Tablica 6. prikazuje rezultate simulacije i odstupanja, a Grafikon 1. promjenu struje u ovisnosti o nagibu terena.

Tablica 6. Rezultati simulacije kosine 1% do 10% i 20%

NAGIB TERENA	BRZINA [m/s]	STRUJA [A]	SNAGA [W]	ENERGIJA [kJ]
1%	0,7	$1,55 \pm 4,1\%$	$18,89 \pm 3,92\%$	2,70
2%		$1,70 \pm 4,8\%$	$21,95 \pm 4,50\%$	3,14
3%		$1,94 \pm 3,40\%$	$24,75 \pm 2,40\%$	3,54
4%		$2,06 \pm 2,81\%$	$26,41 \pm 2,83\%$	3,77
5%		$2,37 \pm 2,40\%$	$29,33 \pm 0,78\%$	4,19
6%		$2,48 \pm 4,43\%$	$31,08 \pm 4,40\%$	4,44
7%		$2,83 \pm 3,50\%$	$36,39 \pm 4,60\%$	5,19
8%		$3,09 \pm 1,46\%$	$39,17 \pm 1,19\%$	5,59
9%		$3,48 \pm 3,1\%$	$43 \pm 1,09\%$	6,14
10%		$3,89 \pm 1,2\%$	$47,73 \pm 2,07\%$	6,82
20%		$7,4 \pm 5\%$	$91,36 \pm 5,8\%$	13,05

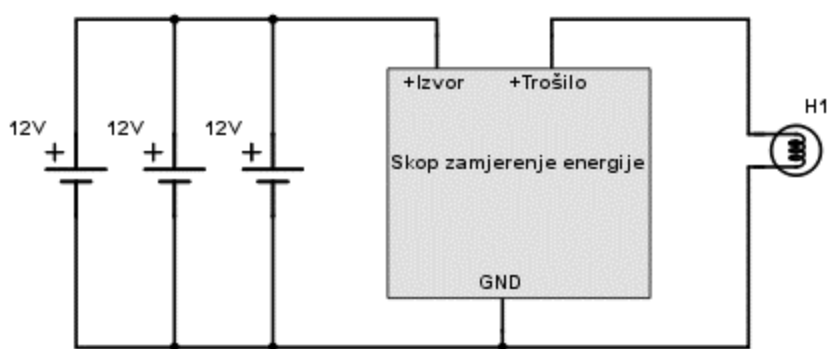
PROMJENA STRUJE U ODNOSU NA POSTOTAK SIMULIRANOG NAGIBA TERENA



Grafikon 1. Odnos struje i postotka nagiba terena

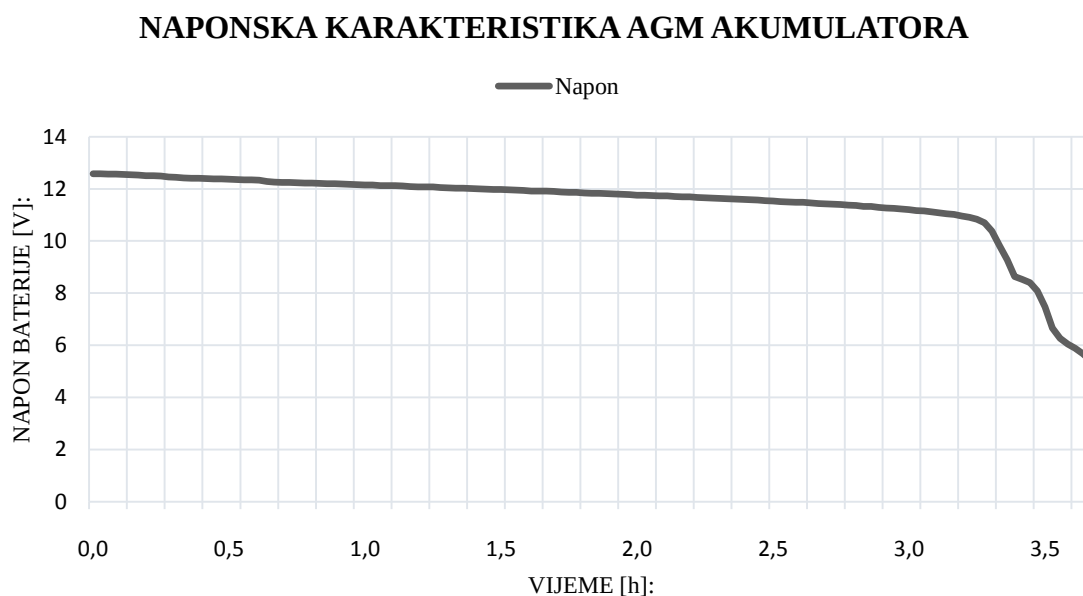
5.3 Karakteristika pražnjenja AGM akumulatora

Karakteristika pražnjenja važna je zbog određivanja stanja napunjenosti akumulatora. Parametar stanja napunjenosti važan je zbog toga što se iz njega iščitava koliki postotak kapaciteta akumulatora je ostao na raspolaganju mobilnom robotu za korištenje. Pokus se izvodio s tri akumulatora koji pogone mobilnog robota. Spojeni su na halogenu žarulju snage 50 W. Pokus je vršen sve dok se akumulatori nisu ispraznili. Shema spoja prikazana je Slikom 17.



Slika 17. Shema spajanja

Bilježenje struje i napona izvodilo se mjernim sklopom opisanim u prethodnim poglavljima. Grafikon 2. prikazuje naponsku karakteristiku dobivenu ovim mjerenjem.



Grafikon 2. Karakteristika pražnjenja AGM akumulatora

Po izrazu (28) za računanje stanja baterije, utvrđeno je da je maksimalno stanje napunjenosti akumulatora ugrađenih u robota 0,55 odnosno 55 %. To znači da je na raspolaganju za korištenje 55% ukupnog kapaciteta akumulatora. Prikupljeni podatci o stanju napunjenosti akumulatora važni su za algoritam optimizacije. Stanje napunjenosti akumulatora određuje hoće li vozilo krenuti nekom rutom ili ne, odnosno pokazuje je li vozilo sa trenutnom energijom koju ima na raspolaganju iz akumulatora u mogućnosti savladati rutu.

Zaključak

U ovom radu obrađena je problematika prikupljanja podataka o potrošnji malog električnog vozila u svrhu optimizacije potrošnje dostupne energije iz akumulatora u vozilu. Ovaj problem postaje sve aktualniji sve većom proizvodnjom i upotrebom električnih automobila. Optimizacija ruta električnog vozila jedan je od načina optimizacije potrošnje energije. S obzirom na relativno spor razvoj novih tehnologija baterijskih akumulatora, optimizacija ruta se nameće kao dobro rješenje ovakvog problema.

Algoritmi za pronalazak najkraće rute, kao Dijkstrin algoritam, za svoje izvršavanje zahtijevaju podatke o cestovnim segmentima nad kojima se vrši optimizacija rute vozila. S obzirom na potrebu prikupljanja podataka, izrađen je sklop za mjerenje energije. Sklop je izveden sa Arduino razvojnom platformom i u mogućnosti je bilježiti struju i napon koje vozilo troši pri kretanju. Struja i napon koji se bilježe služe za izračun snage i energije koje vozilo troši prilikom kretanja po različitim nagibima terena.

Izrađen je algoritam za prikupljanje podataka sa senzora pomoću Arduino platforme. Algoritam je pisan u Arduino IDE okruženju i služi za programiranje mikrokontrolera ugrađenog u platformu. Za prikaz i obradu podataka o struji i naponu koji se zapisuju, izrađena je Windows Form aplikacija pisana u programskom jeziku C#. Aplikacija služi za prikupljanje podataka koje na COM port šalje Arduino platforma. Postoji mogućnost ispisa rezultata u tekstualnu datoteku i Excel tablicu.

Kako bi se mogli prikupiti podatci o potrošnji mobilnog robota na različitim kosinama, predložen je način simuliranja uzbrdice dodavanjem dodatne težine na simulacijska kolica koja se priključuju na robota. Izraz za težinu simulacijskih kolica u ovisnosti o nagibu terena dokazan je u praksi mjerenjima.

- [1] Golubić, J.: Promet i okoliš, Fakultet prometnih znanosti Sveučilišta u Zagrebu, Zagreb, 1999.
- [2] Bekey, G.: Autonomus Robots-From Biological Inspiration to Implementation and Control, The MIT Press Cambridge, London, 2005..
- [3] Jurik, T., Cela, A., Hamouche, R., Natowicz, R., Reama, A., Niculescu, S., Julien, J.: Energy Optimal Real-Time Navigation System, *IEEE Intelligent transport system magazine*, pp. 66-79, 2014.
- [4] Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization Algorithms and Complexity, Dover publications, New York, 1998.
- [5] Dijkstra, E.W.: A note on two problems in connexion with graphs, *Numerische matematik*, svez. 1, pp. 269-271, 1959..
- [6] Sachenbacher, M., Leucker, M., Artmeier, A.: Efficient Energy-Optimal Routing for Electric Vehicles, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, San Francisco, 2011.
- [7] Abousleiman, R., Rawashdeh, O.: Energy Consumption Model of an Electric Vehicle, IEE 2015.
- [8] Jurum-Kipke, J.: Mehanika u prometnom inženjerstvu, Fakultet prometnih znanosti, Zagreb, 2001.
- [9] Pisačić, K.: Osnove inženjerskog proračuna, Skripta, Sveučilište Sjever, Koprivnica, 2014.
- [10] Guzzella, L., Sciarretta, A.: Vehicle Propulsion Systems Introduction to Modeling and Optimization, Springer, Berlin, 2007.
- [11] Kujundžić, G.: Estimacija stanja i parametara VRLA baterije nelinearnim Kalmanovim filtrom, Rad za KDI, Fakultet elektrotehnike i računarstva, Zagreb, 2015.
- [12] Eshani, M., Gao, Y., Gay, S. E., Emadi, A.: Modern Electric, Hybrid Electric and Fuel Cell Vehicles, CRC Press, Boca Raton, 2005.
- [13] ABSORBED GLASS MAT BATTERIES, <http://www.solar-electric.com/agm-battery-technology.html>, pristupljeno: 28.4.2016.

- [14] Absorbent Glass Mat (AGM),
http://batteryuniversity.com/learn/article/absorbent_glass_mat_agm, pristupljeno: 28.4.2016.
- [15] Husain, I.: *Electric and hybrid vehicles- Design Fundamentals*, CRC Press, Boca Raton, 2003.
- [16] Carlos, F., Vicente, M., Joshue, P., David, G., Fawzi, N.: Optimal energy consumption algorithm based on speed reference generation for urban electric vehicles, *IEEE Intelligent Vehicles Symposium*, Seoul, 2016.
- [17] Eisner, J., Funke, S., Storandt, S.: Optimal Route Planning for Electric Vehicles in Large Networks, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, San Francisco, 2011.
- [18] Sweda, T. M., Kalbjan, D.: Finding minimum-cost paths for electric vehicles, *Electric Vehicle Conference (IEVC), 2012 IEEE International*, Greenville, SC, 2012.
- [19] <https://www.arduino.cc/>, pristupljeno: 10.4.2016..
- [20] Badamasi, Y. A.: *The Working Principle Of An Arduino*, 978-1-4799-4106-3/14, IEE, 2014.
- [21] Kreso, S.: *Projektiranje mikroprocesorskih sustava*, Beograd, 2005.
- [22] Balević, A., Marinković, G.: *Atmel AVR mikrokontroleri*, Elektronički fakultet Beograd, Beograd.
- [23] Cvjetković, V., Matijević, M.: Overview of architectures with Arduino boards as building blocks for data acquisition and control systems, 978-1-4673-8246-5/16, 13th International Conference on Remote Engineering and Virtual Instrumentation IEEE, 2016..
- [24] Erdelić, T.: *Lokalizacija mobilnog robota Pioneer 3-DX pomiču kinect senzora*, Završni rad, Fakultet elektrotehnike i računarstva, Zagreb, 2010.
- [25] *Operativni priručnik za Pioneer robote*, Mobile Robots Inc., Amherst, 2006.
- [26] Popčević, P.: Istraživanje Hallovoh efekta kod Taylorovih faza T-Al3Mn kompleksnih metalnih slitina, eksperimentalno istraživanje, Prirodoslovno-matematički fakultet Sveučilišta u Zagrebu, Zagreb, 2011..
- [27] Hrvatsko strukovno nazivlje,
<http://struna.ihjj.hr/searchdo/?q=Hallova+pojava#container>, pristupljeno: 27.5.2016.
- [28] Karuza, D., Deković, M.: *Fizički praktikum IV, Odijel za fiziku Sveučilišta u Rijeci*, Rijeka, 2013.

- [29] Ramsden, E.: Hall-effect sensors: theory and applications, Elsevier, Oxford, 2006..
- [30] Tehnički podatci ACS715, www.allegromicro.com/~media/Files/.../ACS715-Datasheet.ashx, 15.4.2016.
- [31] Alvarez, L., Jingang, Y., Horowitz, R., Olmos, L.: Dynamic Friction Model-Based Tire-Road Friction Estimation and Emergency Braking Control, *ASME*, svez. 127, pp. 22-32, 2005.
- [32] Novosel, D.: Analiza materijala valjka za mjerenje kočne sile, Diplomski rad, Fakultet strojarstva i brodogradnje, Zagreb, 2015.
- [33] Persson, B. N. J. , Tartaglino, U., Albohr, O., Tosatti, E.: Rubber friction on wet and dry road surfaces: the sealing effect, *Phys. Rev. B* 71, 035428, 2005.

Popis slika:

Slika 1. Sile koje djeluju na tijelo koje se kreće konstantnom brzinom po ravnom terenu	6
Slika 2. Sile koje djeluju na tijelo koje se kreće konstantnom brzinom uz kosinu	8
Slika 3. Pravokutni trokut za izračun postotka nagiba terena	9
Slika 4. Osnovni dijelovi i razmještaj pinova Arduino platforme	16
Slika 5. Spajanje Arduino platforme sa računalom sa USB protokolom	18
Slika 6. Arduino programska podrška	19
Slika 7. Fizičke komponente Pioneer 3-AT mobilnog robota	21
Slika 8. Shema ožičavanja Hallove sonde ACS715	23
Slika 9. Elektronički sklop za mjerenje energije	24
Slika 10. Shema sklopa za mjerenje energije	25
Slika 11. Dio Arduino koda koji služi za obradu podataka	26
Slika 12. Metoda IspisCOM()	27
Slika 13. Dijagram toka izvođenja mjerenja	28
Slika 14. Lokacija mjerenja, nagib 10%	29
Slika 15. Lokacija mjerenja, nagib 20%	29
Slika 16. Simulacija uzbrdice simulacijskim kolicima	30
Slika 17. Shema spajanja	33

Popis tablica:

Tablica 1. Prednosti i nedostaci AGM akumulatora u odnosu na mokri akumulator	13
Tablica 2. Karakteristike Arduino Uno i Due platformi	17
Tablica 3. Specifikacije mobilnog robota Pioneer 3-AT	22
Tablica 4. Potrošnja mobilnog robota na kosinama nagiba 10% i 20%	30
Tablica 5. Usporedba simulacije i stvarnog mjerenja na kosini	31
Tablica 6. Rezultati simulacije kosine 1% do 10% i 20%	32

Popis grafikona:

Grafikon 1. Odnos struje i postotka nagiba terena	32
Grafikon 2. Karakteristika pražnjenja AGM akumulatora	33

Prilog 1. Arduino kod za mjerenje snage

```
float naponBaterije,citanjeBaterija;
float R1=118000,R2=38700;
float PinBaterija=0,PinIzlazSonda=9;
float struja,citanje,prosjeckIzlazSonda;
float poljeIzlazSonda[11],sredina;
float citanjeIzlazSonda=0;

void setup(){
  Serial.begin(9600);
}
void loop(){

  for(int j=0;j<10;j++){
    // prva for petlja služi za računanje prosjeka 10 učitanih vrijednosti i spremanje u polje

    float citanjeIzlazSonda=0;
    // postavljanje varijable u 0 kako bi se za svaku iteraciju zbrajala slijedeća očitavanja

    for(int i=0;i<11;i++){
      { // druga petnja služi za zbrajanje očitavanja sa sonde

        citanjeIzlazSonda=citanjeIzlazSonda+analogRead(PinIzlazSonda);
        delay(9);
      }
      prosjeckIzlazSonda=citanjeIzlazSonda/10; // prosječna vrijednost 10 čitanja sa izlaza
      sonde
      poljeIzlazSonda[j]=prosjeckIzlazSonda; // u polje se upisuje 11 prosječnih vrijednosti
      izlaza sa sonde
    }
    sort(poljeIzlazSonda,11); // sortiranje izlaza sa sonde od najmanje do najveće vrijednosti
    sredina=poljeIzlazSonda[5]; // 5. mjesto u polju je sredina svih sortiranih prosječnih
    izlaza sa sonde

    citanjeBaterija=analogRead(PinBaterija);
    naponBaterije=citanjeBaterija*3.3/1024*((R1+R2)/R2); // vrijednost napona baterije u
    voltima

    struja=(500-(3300*sredina/1024))/133; // vrijednost struje u amperima

    Serial.print(struja);
    Serial.println(naponBaterije);
  }

  void sort(float a[], int size) {
    for(int i=0; i<(size-1); i++) {
```

```

for(int o=0; o<(size-(i+1)); o++) {
    if(a[o] > a[o+1]) {
        int t = a[o];
        a[o] = a[o+1];
        a[o+1] = t;
    }
}
}
}
}

```

Prilog 2. Izvorni kod aplikacije za čitanje i zapisivanje podataka sa COM porta

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;
using Excel = Microsoft.Office.Interop.Excel;

namespace SerijskaKomunikacija
{
    public partial class Form1 : Form
    {
        SerialPort port;    // za serijsku komunikaciju
        StreamWriter pisac; // za ispis u .txt datoteku
        string datoteka;    // za ime datoteke
        int i = 0;          // brojac za upis u excel tablicu

        Excel.Application excelAplikacija;
        Excel.Workbook excelRadnaKnjiga;
        Excel.Worksheet excelRadniList;
        object misValue = System.Reflection.Missing.Value; // ovom varijablom se
        definira vrijednost koja nedostaje //npr. ako nije potreban

        pasword za excel datoteku postavi se u missValue
        public Form1()
        {
            InitializeComponent();
            port = new SerialPort();

            excelAplikacija = new Microsoft.Office.Interop.Excel.Application();
            // najava korištenja excel aplikacije
            excelRadnaKnjiga = excelAplikacija.Workbooks.Add(misValue);
            // izrada workbook-a
            excelRadniList =
            (Excel.Worksheet)excelRadnaKnjiga.Worksheets.get_Item(1); // korištenje radnog lista
            1

            txtImeDatoteke.Enabled = false; // postavljeno u false kako bi se
            onemogućio unos prije odabira ispisa
            txtImeExcel.Enabled = false;

        }
    }
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    provjeriPortove(); // funkcija se poziva svaku puta kada se pokrene
    aplikacija
    // provjerava imena COM portova koji su spojeni na racunalo
}

private void provjeriPortove()
{
    try
    {
        // funkcija za provjeru koristenih COM portova
        string[] imenaPortova = SerialPort.GetPortNames();
        foreach (string p in imenaPortova)
        {
            cmbImePorta.Items.Add(p);
        }
    }
    catch (Exception greska)
    {
        MessageBox.Show(greska.Message);
    }
}

private void serialPort_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    // funkcija se izvršava sve dok je uspostavljena komunikacija sa zadanog
    COM porta
    string ulazniPodatak = serialPort.ReadExisting();
    string [] podatci;
    podatci=ulazniPodatak.Split(' ');

    if (port.IsOpen)
    {
        if (checkIspisTxt.Checked)
        {
            // ispis u tekstualnu datoteku
            datoteka = txtImeDatoteke.Text;
            pisac = new StreamWriter(datoteka);
            pisac.WriteLine(ulazniPodatak);
        }
        if (checkExcel.Checked)
        {
            // ispis u excel datoteku
            excelRadniList.Cells[i, 1] = podatci[0];
            excelRadniList.Cells[i, 2] = podatci[1];
            i++;
        }
        // ispis na ekran aplikacije
        richTextBox1.AppendText(ulazniPodatak + "\n");
    }
}

private void Spoji()
{
    try
    {
        // otvaranje novog serijskog porta
        port = new SerialPort(cmbImePorta.Text,
Convert.ToInt32(cmbBrzinaPrijenosa.Text));
        port.Open();
        // definiranje parametara porta
        serialPort.PortName = cmbImePorta.Text;
    }
}

```

```

        serialPort.BaudRate = Convert.ToInt32(cmbBrzinaPrijenosa.Text);

        if (excelAplikacija == null)
            MessageBox.Show("Nemoguće pristupiti Excel aplikaciji!");

        // ispis datuma i vremena na ekran aplikacije
        richTextBox1.AppendText(DateTime.Now+ "\n");

        // kada se pritisne na početak tekst u gumbu se prepiše u Kraj kako
        bi se isti gumb mogao koristiti za dvije stvari
        if (port.IsOpen)
            btnPocetak.Text = "Kraj";
    }
    catch (Exception greska)
    {
        MessageBox.Show(greska.Message);
    }
}

private void KrajKomunikacije()
{
    btnPocetak.Text = "Početak";

    if (checkExcel.Checked)
    {
        try
        {
            // spremanje podataka u excel datoteku
            excelRadnaKnjiga.SaveAs(txtImeExcel.Text,
Excel.XlFileFormat.xlWorkbookNormal, misValue, misValue, misValue, misValue,
Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue, misValue, misValue,
misValue);

            excelRadnaKnjiga.Close(true, misValue, misValue);
            excelAplikacija.Quit();

            // adekvatno zatvaranje excel datoteke
            releaseObject(excelRadniList);
            releaseObject(excelRadnaKnjiga);
            releaseObject(excelAplikacija);
        }
        catch (Exception greska)
        {
            MessageBox.Show(greska.Message);
        }
    }

    i = 0; // postavljanje brojača za upis u excel datoteku u nulu

    // zatvaranje otvorenog porta i toka za upis u tekstualnu datoteku
    if (port.IsOpen)
        port.Close();
    if (checkIspisTxt.Checked)
    {
        pisac.Flush();
        pisac.Close();
    }
}

// metoda definira događaje koji slijede nakon klika na gumb Pocetak
private void btnPocetak_Click(object sender, EventArgs e)
{
    if (port.IsOpen)
        KrajKomunikacije();
}

```

```

        else
            Spoji();
    }

    // metoda definira događaje koji slijede nakon klika na gumb Očisti površinu
    private void btnOčisti_Click(object sender, EventArgs e)
    {
        // ciscenje površine
        richTextBox1.Clear();
    }
    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        // ako port nije zatvoren, zatvoriti ce se kada se ugasi aplikacija
        if (port.IsOpen)
            port.Close();
    }
    // metoda za zatvaranje excel aplikacije nakon korištenja
    private void releaseObject(object obj)
    {
        try
        {
            System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
            obj = null;
        }
        catch (Exception greska)
        {
            obj = null;
            MessageBox.Show("Greška! " + greska.Message);
        }
        finally
        {
            GC.Collect();
        }
    }
    // metode koje onemogućuju/omogućuju upis imena datoteka za upis podataka
    private void checkIspistxt_CheckedChanged(object sender, EventArgs e)
    {
        txtImeDatoteke.Enabled = false;
        if (checkIspistxt.Checked)
            txtImeDatoteke.Enabled = true;
    }
    private void checkExcel_CheckedChanged(object sender, EventArgs e)
    {
        txtImeExcel.Enabled = false;
        if (checkExcel.Checked)
            txtImeExcel.Enabled = true;
    }
    private void btnPomoc_Click(object sender, EventArgs e)
    {
        MessageBox.Show("UPUTE ZA KORIŠTENJE APLIKACIJE\n\n- U padajućem
izborniku COM Port odabire se port s kojega se žele čitati podatci\n"
            + "- U padajućem izborniku Brzina prijenosa odabire se baud rate za
serijsku komunikaciju\n - Označavanjem kvadratića Ispis u tekstualnu"
            + "datoteku omogućava se upis imena tekstualne datoteke u koju će se
upisati podatci NAPOMENA: ime datoteke mora sadržavati ekstenziju .txt\n"
            + "- Označavanjem kvadratića Ispis u Excel tablicu omogućava se upis
imena Excel datoteke u koju će se upisati podatci NAPOMENA: ime "
            + "datoteke mora sadržavati ekstenziju .xls\n- Klikom na gumb Početak
pokreće se upis podataka u označenu datoteku i na zaslon aplikacije\n"
            + "- Klikom na gumb Očisti površinu, sadržaj površine na aplikaciji se
briše ");}
    }
}

```

METAPODACI

Naslov rada: Dinamičko mjerenje potrošnje energije električnog vozila s pomoću Arduino razvojne platforme

Student: Leo Tišljarić

Mentor: izv. Prof. dr.sc Tonči Carić

Naslov na drugom jeziku (engleski):

Dynamic Measuring the Energy Consumption of Electric Vehicle with Arduino Prototyping Platform

Povjerenstvo za obranu:

- | | |
|--------------------------------|-------------|
| • doc. dr.sc Mario Muštra | predsjednik |
| • izv. prof. dr.sc Tonči Carić | Mentor |
| • dr.sc. Pero Škorput | Član |
| • doc. dr.sc. Edouard Ivanjko | zamjena |

Ustanova koja je dodijelila akademski stupanj: Fakultet prometnih znanosti
Sveučilišta u Zagrebu

Zavod: Zavod za Inteligentne transportne sustave

Vrsta studija: Preddiplomski

Studij: Inteligentni transportni sustavi i logistika

Datum obrane završnog

rada: 13. rujna 2016.

Napomena: pod datum obrane završnog rada navodi se prvi definirani datum roka obrane.

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ završni rad
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.
Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj
drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____ završnog rada
pod naslovom: **Dinamičko mjerenje potrošnje energije električnog vozila s
pomoću Arduino razvojne platforme**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom
akademsom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu

.

Student/ica:

U Zagrebu, 13. rujna 2016.

(potpis)