

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

ELEKTRIČNO KOLO

Tematsko področje: ELEKTROTEHNIKA, ELEKTRONIKA IN ROBOTIKA

Avtorja:

Aleksander Pešec, 3. letnik

Aron Koca, 3. letnik

Mentor:

Peter Vrčkovnik, dipl. inž. elektrotehnike

Somentor:

Matjaž Žerak, univ. dipl. inž. elektrotehnike

Velenje, 2015

Raziskovalna naloga je bila opravljena na ŠČ Velenje, Elektro in računalniška šola, 2015.

Mentor:

Peter Vrčkovnik, dipl. inž. elektrotehnike

Somentor:

Matjaž Žerak, univ. dipl. inž.

Datum predavitve: marec, 2015

Aron Koca, Aleksander Pešec

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD ŠC Velenje, šolsko leto 2014/2015

KG električno kolo/izdelava aplikacije/izdelava krmiljenja/povezava gsm

AV PEŠEC, Aleksander/ARON, Koca

SA VRČKOVNIK, Peter/ŽERAK, Matjaž

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2015

LI 2015

IN Električno kolo

TD Raziskovalna naloga

IJ SL

JI sl

AI Namen te raziskovalne naloge je izdelati električno kolo, ki se poveže z mobilnim aparatom porabnika ter na njem prikaže stanje ter informacije kolesa. Že narejeni električni kolesi so tehnično naprednejši, cenovno dražji ter manj modularni kot doma narejen. Glavni sistem je mikrokrmilnik Arduino, na čimer se tudi programira. Celoten sistem poteka tako, da je vozilo povezano na postajo preko wifija, s telefonsko aplikacijo odkleneš ter se povežeš s kolesom preko bluetootha. Modeli ter senzorji podatke pošiljajo na mikrokrmilnik in nato posredno na mobilni telefon. Na zaslonu se prikažejo dobljene informacije, to so hitrost vozila, kapaciteta baterije ter pozicija na gpsu.

KEY WORDS DOCUMENTATION

ND ŠC Velenje 2014/2015

CX electric bike/ application/ circuit control/ gsm connection

AU PEŠEC, Aleksander/KOCA, Aron

AA VRČKOVNIK, Peter/ŽERAK, Matjaž

PP 3320 Velenje, SLO, Trg mladosti 3

PB ŠC Velenje Elektro in računalniška šola, 2015

PY 2015

TI Electric Bicycle

DT REASEARCH WORK

LA SL

AL sl/en

AB The purpose of this research paper is to build an electric bicycle, which connects to a mobile device of the person using the bicycle, and displays on it the information of the data of the vehicle. Already built electric bicycles are more technical advanced and, more expensive and less modular than a homemade version. The core of the automatics, hardware and software is a microcontroller Arduino. The bicycle is connected to the stations through wifi, the whole process starts with the connection of the mobile application to the bicycle through Bluetooth. Then on the screen of the smartphone, the collected data will be shown, such as the vehicle's speed, the battery capacity and the position of the vehicle on the gps.

Kazalo vsebine

UVOD	7
1.1 Hipoteze.....	7
2 PREGLED OBJAV	8
2.1 Doma izdelani električni kolesi	8
2.2 Komercialni električni kolesi.....	8
2.3 Aplikacija	8
3 PREGLED STANJA TEHNIKE.....	8
3.1 Haibike XDURO FS RX 27.5”.....	8
3.2 Organic Transit ELF.....	9
3.3 Arduino.....	10
4 MATERIALI IN METODE DELA.....	12
4.1 Strojni del	12
4.1.1 Elektro motor.....	12
4.2 Elektro del	13
4.1.1 Izvor napetosti	13
4.1.1 Razsvetljava.....	14
4.1.1 Regulator hitrosti	14
4.3 Računalniški del	15
4.3.1 Arduino Mega 2560.....	15
4.3.2 Wifi modul.....	18
4.3.3 Bluetooth modul	18
4.3.4 Hall senzor.....	19
4.3.5 Aplikacija	20

5	RAZPRAVA	24
6	ZAKLJUČEK.....	25
7	ZAHVALA.....	26
8	LITERATURA.....	27
9	AVTORJA RAZISKOVALNE NALOGE.....	28
10	PRILOGE	29

Kazalo slik

1.	HaiBike XDURO FS RX	9
2.	Organic Transit ELF.....	10
3.	Elektromotor.....	13
4.	Akumulatorja.....	13
5.	Razsvetljava kolesa	14
6.	Regulator hitrosti.....	15
7.	Arduino Mega 2560.....	17
8.	Wifi modul	18
9.	Bluetooth modul	18
10.	Hall Senzor	19
11.	Testno vezje.....	19
12.	Shema	20
13.	Interface aplikacije	21
14.	Koda aplikacije.....	21
15.	Zasloni aplikacije	22
16.	Avtorja.....	2

1 UVOD

Električna vozila so odlična alternativa prevoza kot druga vozila, ki jih poganjajo fosilna goriva. Namen te raziskovalne naloge je uspešno izdelati, ne samo električno kolo ampak uporabiti tudi mobilni telefon – aplikacijo, katera nam olajša in izračuna optimalno vožnjo z električnim kolesom. V raziskovalni nalogi smo izdelali električno kolo. Električno kolo vsebuje 24V elektromotor, dvojne 12V akumulatorje ter regulator hitrosti. Ker v Velenju že imamo kolesa za izposajo, smo razmišljali, da bi uporabnikom ponudila na izposajo tudi električna kolesa. V prihodnosti pa bi bila lahko na izposajo samo električna kolesa, če se ponudnik, v našem primeru MO Velenje, za to odloči. Mobilna aplikacija, ki smo jo izdelali, temelji na zajemu vseh mest za izposajo, pri katerem lahko vidimo, koliko koles in kje se trenutno nahajajo. Prav tako lahko vidimo njihovo stanje baterij.

1.1 Hipoteze

Na začetku raziskovanja smo si zastavili naslednje hipoteze:

- Kolo je možno predelati doma, s splošnim znanjem s področja tehnike in tehnologije.
- Stanje parametrov na kolesu je s pomočjo krmilnega vezja in preprostega znanja programiranja možno povezati z mobilnim telefonom.
- Mobilna aplikacija bo omogočala pregled izbranih podatkov.
- Mobilno aplikacijo je možno sprogramirati s splošnim znanjem programiranja Arduina.
- Mobilna aplikacija bo zaradi merjenja in določanja parametrov vožnje olajšala vožnjo s kolesom.

2 PREGLED OBJAV

2.1 Doma izdelana električna kolesa

Na spletu smo pregledali veliko doma narejenih električnih koles. Prišli smo do ugotovitve, da je v Sloveniji zanimanja za električno kolo malo. Veliko bolj je priljubljena izdelava domačih električnih koles v Ameriki ter Avstraliji. Pri velikih primerih uporabijo samo manjši elektromotor ter akumulator.

2.2 Komercialna električna kolesa

Pred izdelavo našega kolesa smo temeljito pregledali električna kolesa, ki jih trenutno ponuja tržišče. Na kratko smo opisali tudi nekaj različnih modelov različnih proizvajalcev.

2.3 Aplikacija

Trenutno je na voljo veliko programov, v katerih si lahko oseba sama sestavi svojo poljubno aplikacijo. Večina aplikacij se piše v programskem jeziku C++, a so tudi drugi, pri katerih se uporabljajo t.i. »blocks«, kjer so na voljo že določeni ukazi in jih nato nadgrajujemo z drugimi ukazi. Aplikacijo, ki smo jo uporabili, je tudi opisana.

3 PREGLED STANJA TEHNIKE

3.1.1 HaiBike XDURO FS RX

Električno kolo višjega razreda firme HaiBike, specializirano za gorske proge ter težek teren, je prvo električno kolo, ki ponuja izredno moč s komaj 350 ali 550 watt motorjem, kar je glede na teren, za katerega je specializiran, fenomenalno. Motorje poganja 36 v litijsko-ionska

baterija, kar je zadnje čase pogostejše kot navaden akumulator, začetna cena je približno 4.000 €.



Slika 1: HaiBike XDURO FS RX

Lahko doseže hitrosti 32 km/h na težkem terenu, ob polni porabi lahko baterija zdrži do maksimalno 3 ur.

3.1.2 Organic Transit ELF

Najbolj popularen in prepoznaven električni trike firme OrganicTransit. Kolo uporablja 600 wat motor ter je težak kot povprečen moški. Lahko doseže hitrosti 32 km/h ter ima opcijo polnjenja s kablom v vtičnico ali preko sončnih celic, ki so locirane na strehi vozila. Za razliko kot druga električna kolesa, je to eno izmed redkih, ki podpira maksimalno dve osebi, začetna cena pa se giblje okoli 5.000 €.



Slika 2: Organic Transit ELF

Za navadno električno kolo ni potrebno vključiti krmilnega vezja arduino, ker pa smo hoteli kolo nadgraditi, smo ga morali dodati v projekt.

3.1.3 Arduino

Zgodovina Arduina sega v leto 2005, kjer se je pričel kot projekt za študente na Internacionalnem inštitutu za Dizajn Ivrea v mestu Ivrea v Italiji.

Arduino je mikrokrmilnik na matični plošči, ki je zasnovan tako, da bi bil postopek z uporabo elektronike v multidisciplinarnih projektih bolj dostopen. Strojno opremo sestavljajo odprtokodna oblika plošče in 8-bitni mikrokontroler Atmel AVR ali 32-bitni Atmel ARM. Programska oprema je sestavljena iz standardnega programskega jezika, prevajalnika in zagonskega nalagalnika, ki se izvaja na mikrokrmilniku. Razvojne plošče Arduino so naprodaj že sestavljene ali pa v »sestavi sam« izvedbi.

Arduino ploščica vsebuje Atmelov 8-bitni AVR mikro krmilnik z dopolnilnimi komponentami za lažje vključevanje in načrtovanje. Pomemben vidik pri Arduinu je, da so priključki standardno izpostavljeni in omogočajo CPU plošči, da se priključi na različne dodatne module, poznane kot ščit.

Nekateri ščiti komunicirajo z Arduinom direktno preko različnih pinov, vendar je veliko teh ščitov posamezno naslovljenih preko serijskega vodila I2C, kar omogoča, da je mnogo teh ščitov uporabljenih vzporedno.

Uradne Arduino ploščice:

- Arduino Diecimila
- Arduino Duemilanove
- Arduino UNO
- Arduino Leonardo
- Arduino Mega
- Arduino NANO

Arduino programi so napisani v C, C++ ali v uradnem Arduinovem programu. Za programiranje si moramo to razvojno okolje tudi prenesti z interneta. Nato moramo preko USB kabla povezati na računalnik Arduina. Pri tem moramo počakati, da se namestijo gonilniki. Če se le ti ne namestijo sami, oz. se ne namestijo pravilno, moramo z interneta oz. uradne Arduino strani prenesti tudi te in jih namestiti ročno. Ob primeru, da ti gonilniki niso nameščeni, Arduino ploščica ne bo prepoznana in tako ne bomo mogli nalagati projektov nanj.

Ko imamo nameščeno razvojno okolje in gonilnike Arduina, lahko pričnemo s pisanjem kode. Najprej definiramo knjižnice, ki jih potrebujemo. Pomembno pri programiranju je, da definiramo 2 glavni funkciji, ki se izvajata:

Setup (): Funkcija se zažene samo enkrat in sicer takrat, ko se Arduino zažene. Tukaj ponavadi nastavimo nastavitve.

Loop (): Ta funkcija se izvaja neprekinjeno, dokler ne izključimo Arduina. To je tudi glavna funkcija in bi ji lahko rekli void Main () v C++.

Za lažje programiranje in razumevanje programske kode ima Arduino IDE na voljo tudi veliko primerov, kako se z neko knjižnico sprogramira osnova s to knjižnico.

Ko imamo program napisan, imamo možnost, da ga preverimo. Če je s programom vse v redu, ga lahko kasneje naložimo na Arduina, v nasprotnem primeru nam bo Arduino IDE sporočil, kaj bi lahko bilo narobe. Če je s programom vse v redu, moramo paziti, da najprej nastavimo, na katero ploščo bomo nalagali program. To storimo s klikom na Orodja → Plošča in nato izberemo ploščo, ki jo uporabljamo. Prav tako moramo nastaviti Serijski vmesnik, na katerem se nahaja Arduino ploščica. To storimo s klikom na Orodja → Serijski vmesnik in izberemo tistega, ki ga ploščica uporablja. Ko imamo to nastavljeno, lahko program naložimo na Arduino ploščico.

4 MATERIALI IN METODE DE LA

4.1 Strojni del

Edini strojni del, ki spada k tej raziskovalni nalogi, je elektromotor.

4.1.1 Elektromotor

Za projekt smo rabili elektromotor, ki ni preveč močen, a hkrati tudi ne prešibek. Zato smo se odločili za motor, ki ima sledeče podatke:

- 350W (Watt)
- 24V/DC (izmenična napetost)
- 19A (tok)
- 2400 RPM (obratov na minuto)



Slika 3: Elektromotor

4.2 Elektro del

Pod električni del našega projekta spada regulator hitrosti, razsvetljava, ter izvor napetosti.

4.2.1 Izvor napetosti

Za izvor napetosti sta bila uporabljena dva 12 V/9 AH akumulatorja, ki sta povezana na elektromotor ter na regulator hitrosti.



Slika 4: Akumulatorja

4.2.2 Razsvetljava

Kot razsvetljava so bili uporabljeni trije elementi: rdeči led trak, beli led trak in luč. Na voljo sta dva stikala, eno stikalo prižge ali ugasne samo sprednjo luč, drugo stikalo pa prižge ali ugasne sprednji ter zadnji led trak.



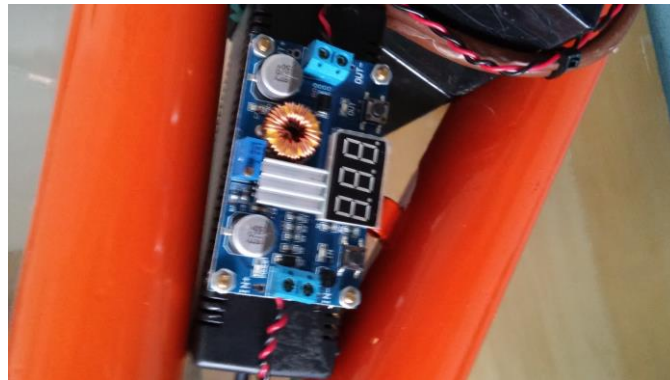
Slika 5: Razsvetljava Kolesa

4.2.3 Regulator hitrosti

Za regulator hitrosti, oziroma regulator napetosti, je bil uporabljen že izdelan komplet.

Regulator, nadzor hitrostim je elektronsko vezje z namenom, da spreminja hitrost ter smer elektromotorja. Regulatorji se pogosto uporabljajo za nadzor električno radijskih modelov, najbolj pogosto pa za regulacijo motorjev.

Ne glede na vrsto uporabljenega regulatorja, nadzoruje spreminjanje hitrosti preklapljanja mreže tranzistorjev s poljskim pojavom. Hitro preklapljanje tranzistorjev povzroča motor, da oddajo svojo značilno visokoglasovno cviljenje, še posebej opzno pri nižjih hitrostih. Enako funkcijo omogoča tudi veliko lažje in bolj natančno spreminjanje vrtljajev motorja in je bolj učinkovit način kot mehnskega tipa.



Slika 6: Regulator Hitrosti

4.3 Računalniški del

Pod računalniški del spada avtomatika najinega projekta. V tem poglavju bomo opisali mikrokrmilnik, ki smo ga uporabili, program ter module.

4.3.1 Arduino Mega 2560

Arduino Mega2560 je mikrokrmilna plošča, ki temelji na ATmega 2560. Vsebuje 54 digitalnih vhodno/izhodnih pinov (od katerih je lahko 15 uporabljenih kot PWM izhod), 16 analognih vhodov, 4 strojne serialne vhode, 16 MHz kristalni oscilator, USB vhod, power jack, ICSP glavo in gumb za resetiranje. Vsebuje vse za podporo mikrokontrolerja. Arduino Mega 2560 lahko enostavno povežemo na računalnik preko USB kabla ali pa ga napajamo preko AC/DC adapterja ali baterije. Arduino Mega je kompatibilen z večino ploščic, ki so dizajnirane za Arduino. Arduino Mega 2560 je nadgradnja Arduino Mega, ki ga tudi nadomešča.

Arduino Mega 2560 se razlikuje od vseh drugih ploščic po tem, da ne uporablja več FTDI USB serialnega čipa. Namesto tega ima Atmega16U2 (v reviziji 1 in 2) programiran USB serialni pretvornik.

Arduino Mega lahko napajamo preko USB priključka ali preko dodatnega napajanja. Izvor napajanja se določa avtomatsko. Dodatno (torej brez USB) napajanje lahko izvira ali iz AC/DC napajalnika ali pa iz baterije. Adapter je lahko povezan s 2.1 mm središčno-

pozitivnim priključkom na ploščo, s tako imenovanim power jack priključkom. Vodi iz baterije se lahko vstavijo v Gnd (ozemljitev) in Vin pin za napajanje plošče.

Plošča lahko upravlja z dodatnim zunanjim napajanjem v območju od 6 do 12 voltov. Če je to napajanje manjše kot 7 V, se lahko zgodi, da bo 5 V pin proizvajal manj kot 5 voltov in takrat je plošča nestabilna. Če pa je to napajanje več kot 12 V, se lahko regulator napetosti pregreje in tako lahko pride do poškodb na plošči. Zato je priporočeno območje med 7 in 12 volti.

Napajalni vtiči so naslednji:

- VIN: Vhodna napetost na Arduino ploščico, kadar se uporablja izvor zunanje napetosti. Skozi ta pin lahko torej oskrbujemo napajanje Arduina.
- 5 V: Ta pin se uporablja za izhodno napetost, ki je regulirana na max. 5 V. Ploščica je oskrbovana z napetostjo preko DC power jacka (7-12 V), USB vhoda (5 V) ali pa VIN pina (7-12 V). Napajanje preko 5 V ali 3.3 V pina gre mimo regulatorja napetosti in lahko ob primeru, da priključimo napajanje na to, uniči ploščico.
- 3V3: 3.3 V napajanje ustvari regulator na ploščici. Maksimalni tok, ki teče, je 50 mA.
- GND: Ozemljitveni pini
- IOREF: Ta pin je na Arduino ploščici, predvideva sklic napetosti, s katero mikrokontroler obratuje. Pravilno nastavljen ščit se bere iz IOREF pina in tako izbira ustrezen vir energije ali pa omogoča napetost izhodnih pinov za delo 5 V ali 3.3 V.

ATmega 2560 ima 256 KB flash pomnilnika za shranjevanje kode (od tega se 8KB uporablja za zagonsko nalaganje), 8 KB SRAM in 4 KB EEPROM-a.

Arduino Mega 2560 ima številne objekte za komuniciranje z računalnikom, drugimi Arduino ploščicami ali ostalimi mikrokontrolerji. ATmega16U2 na kanalih ploščice preko USB vhoda zagotavlja virtualni COM vhod do programske opreme na računalniku (Windows operacijski sistemi za to potrebujejo .inf datoteko, vendar OSX in Linux operacijski sistemi bodo ploščico kot COM vhod prepoznali avtomatsko). Arduino programska oprema IDE vsebuje tudi serialni monitor, kateri omogoča enostavne tekstovne podatke, ki jih pošiljamo na ploščico. RX in TX LED diode bodo na ploščici utripale, kadar se bodo podatki prenašali preko

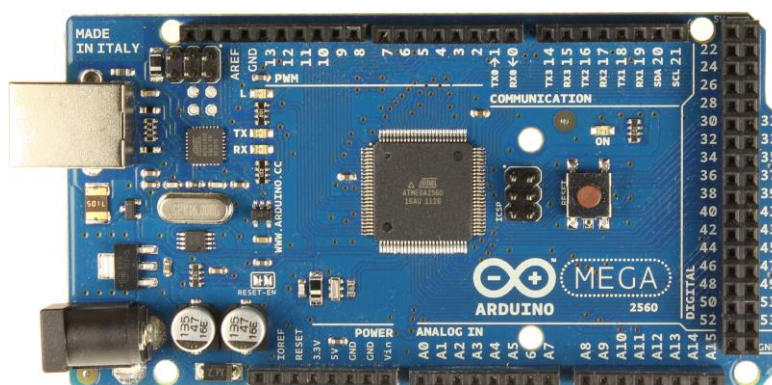
ATmega 8U2/ATmega 16U2 čipa in USB kabla, ki je povezan z računalnikom. SoftWareSerial knjižnica omogoča serialno komunikacijo, na katerikoli digitalni pin Arduino Mega ploščice. ATmega 2560 prav tako podpira TWI in SPI komunikacijo. Arduino programska oprema vključuje Wire knjižnjico za enostavnejšo uporabo TWI vodila.

Arduino Mega ploščico programiramo tako, kot vse ostale ploščice z uporabo Arduino programske opreme IDE.

ATmega 2560 omogoča zagonskemu nalagalniku, da naložimo na ploščico novo kodo brez uporabe dodatne strojne opreme programerja. Komunicira z izvirnim STK500 protokolom.

Namesto tega, da bi morali vedno fizično pritisniti reset gumb, preden bi ponovno naložili program na ploščico, je Arduino Mega 2560 zasnovan tako, da ko naložimo program Arduino, avtomatsko preko programske opreme resetira Arduina. Eden od strojnih nadzorov poteka preko DTR linij od ATmega 8U2, ki so povezane v linije na ATmega 2560 preko 100 nanofarad kondenzatorjev. Kadar so te linije uporabljene, bo ta reset linija onemogočena dovolj dolgo, da bo resetirala čip.

Arduino programska oprema uporablja to možnost, da nam omogoča nalaganje kode enostavno s pritiskom na naloži gumb v Arduino IDE okolju. Arduino Mega 2560 ima poly varovalko, ki ščiti računiška USB vrata pred kratkimi stiki in prevelikemu toku. Čeprav večina računalnikov zagotavlja svojo notranjo zaščito, ta varovalka zagotavlja dodatno plast zaščite. Če pride na USB vhodu do uporabe več kot 500 mA, bo varovalka samodejno prekinila povezavo, dokler ne bomo odstranili naprave, ki proizvaja kratek stik ali pa prevelik tok.



Slika 7: Arduino Mega 2560

4.3.2 Wifi modul

Wifi modul je dodatek za Arduino, s katerim lahko komuniciramo in opravljamo Arduino preko brezžične internetne povezave, ki smo ga uporabili pri našem projektu.

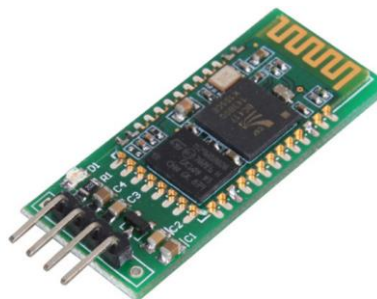
Vsebuje 8 vhodno/izhodnih pinov, s katerimi se poveže z Arduino, ki so napajanje (Vcc), masa (Gnd), ter pini, po katerih se prenašajo podatki (CSN, CE, SCK, MUSI, IRQ, MISI). To omogoča mikročip NRF T 24L01.



Slika 8: Wifi modul

4.3.3 Bluetooth modul

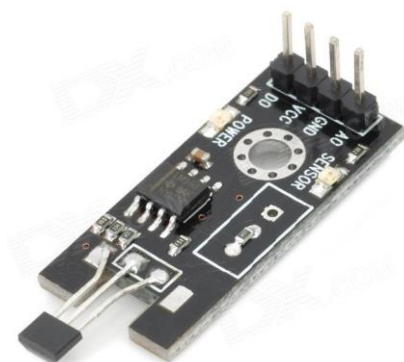
Bluetooth modul uporabljamo za brezžično povezavo med Arduino in neko napravo, v tem primeru med Arduino in GSM-om. Ima 2 vhodno/izhodna prenos podatkov (Rx, Tx) in 2 za napajanje (PWR, GND). Kadar se naprava poveže z modulom to potrди dioda.



Slika 9: Bluetooth modul

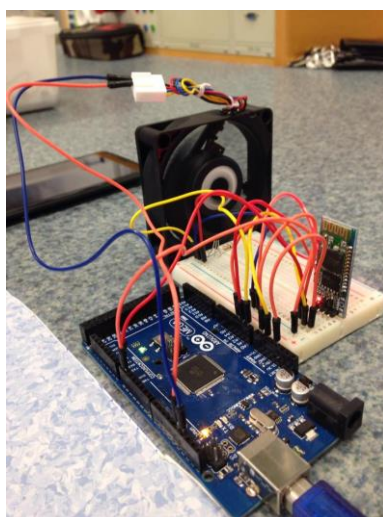
4.3.4 Hall senzor

Hallov senzor je pretvornik, ki spreminja svojo izhodno napetost glede na magnetni pretok skozi hallovo ploščo. Uporabljajo se za pozicioniranje, zaznavanje hitrosti ter še druge podatke. Senzor deluje kot analogni pretvornik, ki neposredno vrača napetost.

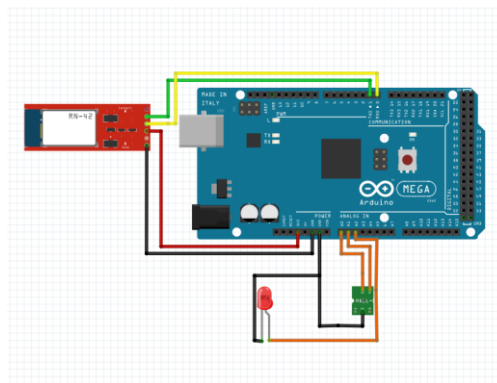


Slika 10: Hall senzor

V našem končnem vezju so bili izbrani moduli: Bluetooth ter Hall senzor. Vezje smo naprej testirali na računalniškem ventilatorju, kasneje pa ga prenesli na kolo.



Slika 11: Testno vezje



Slika 12: Shema

4.3.5 Aplikacija

Po nekaj raziskavah smo odkrili aplikacijo MIT App Inventor 2. App Inventor za Android je open-source spletna aplikacija, zasnovana od kompanije Google, in jo sedaj vzdržuje Massachusetts Institute of Technology (MIT).

Omogoča novincem računalniškega programiranja, da ustvarjajo programske aplikacije za operacijski sistem (OS) Android. To uporablja grafični vmesnik, ki je zelo podobna Scratch in uporabniški vmesnik StarLogo TNG, ki omogoča uporabnikom, da povlečejo in spustijo vizualne predmete ter ustvarijo aplikacijo, ki lahko deluje na napravah Android.

App Inventor in projekti, na katerih temeljijo, so poudarili programiranje, da se lahko uporabi kot vozilo za opravljanje močnih idej skozi aktivno učenje. Kot taka je del tekočega gibanja, računalnikov in izobraževanja, ki se je začelo z delom Seymour Papertove in logotip skupine MIT in 1960 se je manifestiral tudi z delom Mitchel Resnick o Lego Mindstorms in StarLogo.

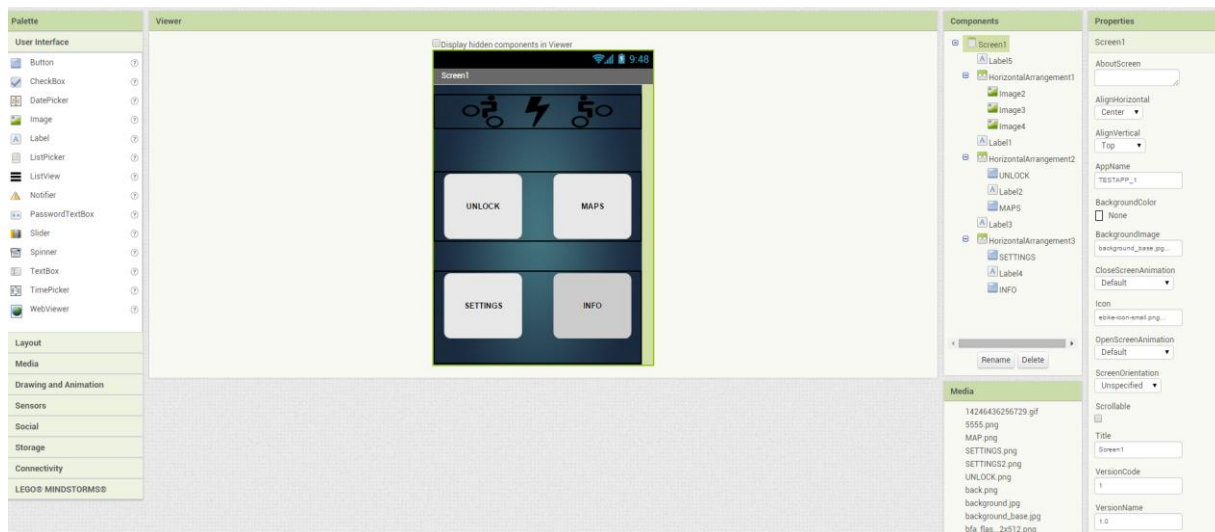
App Inventor vključuje:

oblikovalca, v katerem so komponente programa določeni.

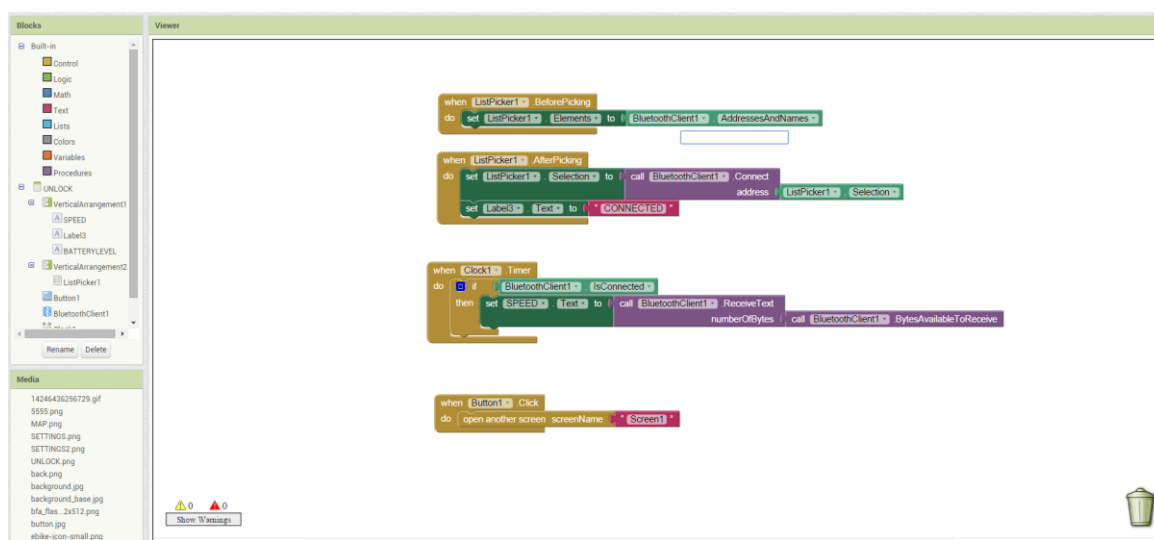
Vključuje vidne komponente, kot so gumbi in slike, ki so dani na simuliranem zaslonu, in ne-vidne komponente, kot so senzorji in spletne povezave.

Blok urednik, v kateri se ustvari logika programa:

Prevajalnik, ki temelji na okvirju programskega jezika Kawa, Scheme in Java ima za vsako določeno kodo svoj t.i. »blok«. Uporabnik nato te bloke sestavlja po svoji želji, da nastane logična povezava med njimi.



Slika 13: Vmesnik aplikacije



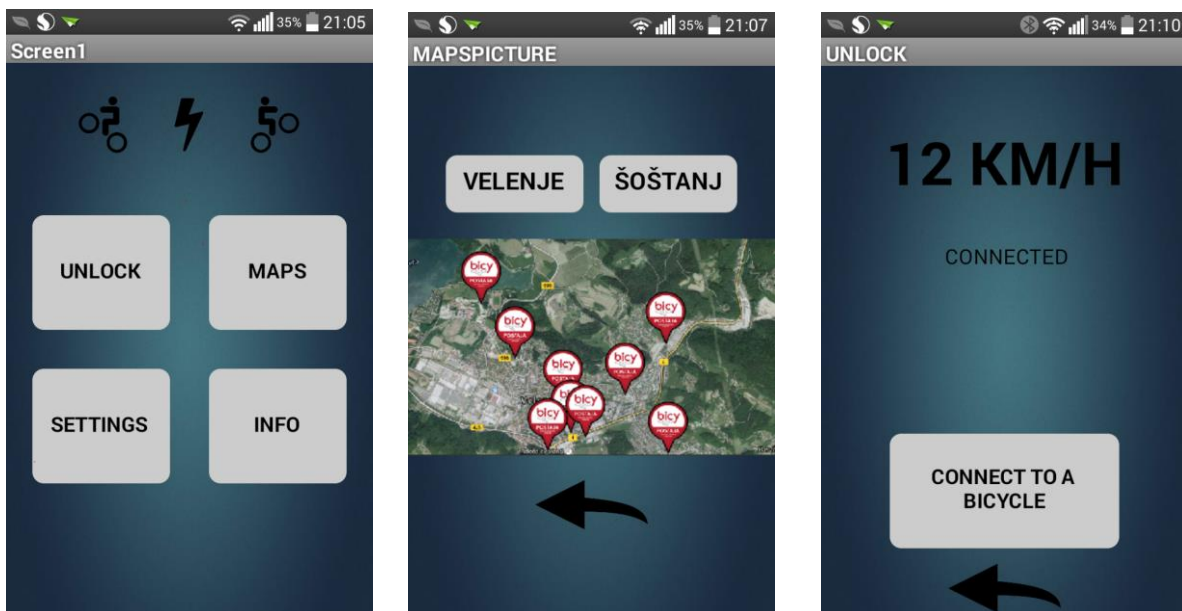
Slika 14: Koda aplikacije

Program smo želeli narediti prijazen uporabnikom. Ko se aplikacija zažene, se odpre domače okno t.i. »homescreen«. Na tem oknu ima uporabnik izbiro, kaj želi narediti naprej. V domačem oknu so gumbi za informacije, nastavitve, mape ter gumb za odklepanje kolesa.

V oknu mape si lahko uporabnik pogleda, kje vse v Velenju in Šoštanju so postaje za kolesa. Na voljo ima slikovno izbiro. Uporabnik lahko s klikom izbere aktualno postajo in vidi, koliko koles je prisotnih in kakšno je njihovo stanje. V novem oknu se prikaže stanje te

postaje; kje je locirana, naslov, koliko koles je na voljo, kapaciteta baterij koles ter možna rezervacija koles.

Kadar se želi uporabnik povezati s kolesom, mora odpreti okno Odpiranje - *Unlock*, ki mu le to omogoči. V novem oknu izbere povezavo s kolesom. To stori s pritiskom na gumb Povezava s kolesom – *Connect to a bicycle*. V podoknu se prikažejo kolesa, ki sna voljo. Ko uporabnik izbere svoje kolo, se na aplikaciji prikaže napis, ali se je mobilni telefon uspešno povezal. Po uspešni povezavi se bodo začeli prikazovati podatki (hitrost ter kapaciteta baterije).



Slika 15: Zasloni aplikacije: od leve proti desni (domači zaslon, prikaz postaj, prikaz podatkov)

Nadgradnja - rezervacija kolesa

Kolo bi uporabnik rezerviral tako, da bi na dodatnem oknu v določeni postaji izbral določeno kolo, ki bi si ga rad rezerviral. Kolo mora biti na voljo, da je lahko rezervacija upoštevana. Pri tem bi se uporabnik predhodno mogel registrirati pri ponudniku koles. Le ta bi v prijavnem obrazcu zapisal kodo mobilnega telefona. Koda bi v nadaljevanju služila kot prepoznavanje uporabnika pri povezavi s kolesom.

Ko uporabnik izbere prosto kolo, vpiše posebno kodo, ki si jo določil pri registraciji, ter s to kodo odklene kolo. Če uporabnik kolesa ne odklene v določenem času, se rezervacija izniči ter je kolo prosto za uporabo ostalim uporabnikom.

Kot drugi način za registracijo bi lahko uporabnik namesto kode imel QR kodo. Če uporabnik želi uporabljati navigacijo do izbrane točke, katero bo prepotoval s kolesom, bi izbral gumb Navigacija – *Navigation*. Podobno kot v programih za izračun poti bi vpisal začetno in končno točko oz lahko v zemljevidu izbere pot od točke A do točke B. Aplikacija nato izračuna približen čas potovanja ter kapaciteto baterije potrebno za pot. Če je kapaciteta baterije prenizka, aplikacija opozori uporabnika glede tega.

5 RAZPRAVA

Na začetku raziskovanja smo si zastavili naslednje hipoteze:

Kolo je možno predelati doma, s splošnim znanjem s področja tehnike in tehnologije.

Z našim znanjem smo uspešno predelali kolo, zato je ta hipoteza potrjena.

Stanje parametrov na kolesu je s pomočjo krmilnega vezja in preprostega znanja programiranja možno povezati z mobilnim telefonom. Hipoteza je potrjena.

Krmilno vezje Arduino je preko povezave bluetooth modula pošiljalo željene podatke na mobilni telefon. Hipoteza je potrjena.

Mobilna aplikacija bo omogočala pregled izbranih podatkov.

Mobilna aplikacija odlično omogoča pregled podatkov, kje so aktualne postaje, koliko je koles, kakšno je njihovo stanje. Mobilna aplikacija omogoča rezervacijo, hipoteza je potrjena.

Mobilno aplikacijo je možno sprogramirati s splošnim znanjem programiranja Arduina.

Z znanjem programskega jezika za krmilno vezje Arduino nismo uspeli uspešno narediti mobilne aplikacije. Zato smo morali poiskati programski jezik za programiranje mobilnih aplikacij MIT App Inventor 2, ki je za razliko od drugih ponudnikov za izdelovanje mobilnih aplikacij, unikatna, saj je programiranje z njo drugačno. Zato je ta hipoteza delno potrjena.

Mobilno aplikacijo bo zaradi merjenja in določanja parametrov vožnje olajšala vožnjo s kolesom.

Aplikacija ne ponuja vseh podatkov, da bi maksimalno olajšala vožnjo s kolesom. Ponuja pa minimalne potrebe, s katerimi se vidi napredek, ter zato delno olajša vožnjo. Aplikacija ter kolo trenutno ne ponudita maksimalno olajšanje vožnje, saj to omejuje najino znanje. Zato je ta hipoteza delno potrjena.

6 ZAKLJUČEK

Najpogostejši način prevoza, ki ga ne poganja kakršenkoli motor so kolesa. A to se že spreminja, saj ljudje želijo imeti dodatno pomoč pri vožnji. Zato se vsak mesec povečuje procent uporabe kolesa z elektropogonom.

Glede na izbrano vožnjo, lahko si jo že predhodno nastavimo, nam aplikacija izračuna, ali so baterije dovolj polne za izbrano vožnjo in koliko časa bomo potrebovali za doseg končnega cilja. Med vožnjo lahko opazujemo, kje se trenutno nahajamo in stanje baterij. Prav tako nam aplikacija sporoča, kje so prazna aktivna mesta glede na našo trenutno lokacijo.

Že narejena kupljena električna kolesa so tehnično naprednejša, cenovno dražja ter manj modularna kot doma narejena. Naše kolo poganja elektromotor, ki je izbran glede na premer zadnjega kolesa in zobnikov, ter dva akumulatorja. Krmilni sistem sestavlja krmilno vezje Arduino. Na krmilno vezje lahko povežemo že kupljene module, ali pa samogradnje, kar nama omogoča dodaten razvoj.

Namen raziskovalne naloge je izdelati električno kolo in mobilno aplikacijo. Aplikacija omogoča pregled aktivnih točk za izposajo kolesa, stanje baterije na posameznem kolesu in izračun parametrov za izbrano pot. Ko pa na aktivni točki lahko z mobilnim telefonom odklenemo izbrano kolo, mobilni telefon odložimo v za to pripravljen nastavek in spremljamo parametre na poti.

Menimo, da bi tovrsten sistem vplival na dober izkoristek in prihranil čas ljudem pri izbiri aktivnih točk poti.

Izkušnje, ki smo jih pridobili pri izdelovanju raziskovalne naloge, še posebej zaradi izvedenih eksperimentov, so dragocene.

Za nas je raziskovanje pomenilo tudi druženje in sodelovanje. Bili so trenutki, ko nam ni šlo dobro in smo se jezili, toda večino časa smo preživeli prijetno, se veselili in celo zabavali.

7 ZAHVALA

Za pomoč pri raziskovalni nalogi bi se radi zahvalili:

- Mentorju učitelju Petru Vrčkovniku za konstantno podporo ter pomoč pri celotnem projektu;
- Somentorju učitelju Matjažu Žeraku za podporo ter nasvete glede strojnega dela;
- Profesorju Urošu Remenihu za pomoč in nasvete glede Arduina ter modulov;
- Staršem za podporo.

8 VIRI IN LITERATURA

1. <http://arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth>
2. <http://www.instructables.com/id/Cheap-2-Way-Bluetooth-Connection-Between-Arduino-a/>
3. <http://arduino.cc/en/Reference/WiFi>
4. http://en.wikipedia.org/wiki/Hall_effect_sensor
5. <http://playground.arduino.cc/Code/HallEffect>
6. <http://diyhacking.com/arduino-hall-effect-sensor-tutorial/>
7. <http://playground.arduino.cc/Tutorials/GPS>
8. <http://www.instructables.com/id/Connecting-GPS-module-to-Arduino/>
9. http://en.wikipedia.org/wiki/Electric_motor
10. <http://www.electric-bikes.com/>
11. http://en.wikipedia.org/wiki/Voltage_regulator
12. <http://electrical4u.com/speed-control-of-dc-motor/>
13. http://www.pyroelectro.com/tutorials/tachometer_rpm_arduino/
14. <http://forum.arduino.cc/index.php?topic=39669.0>

9 AVTORJI RAZISKOVALNE NALOGE

Aron Koca

Aron Koca je dijak 3. Letnika Elektro in računalniške šole (ERŠ) v Velenju. Za raziskovalno nalogo se je odločil ker ga zanimajo električna kolesa in je hotel narediti svojega. V prostem času se ukvarja z igranjem kitare in s kolesarjenjem. Zanima ga tudi avtomatizacija predmetov.

Aleksander Pešec

Aleksander Pešec je dijak 3. Letnika Elektro in računalniške šole (ERŠ) v Velenju. K raziskovalni nalogi se je pridružil, ker ga zanima programiranje, elektronika ter raziskovanje k novi tehnologiji. V svojem prostem času se ukvarja z igranjem klavirature, brenkal ter rad preživi čas za svojim računalnikom.



Slika 16: Avtorja (Levo Aleksander, desno Aron)

10 PRILOGE

PROGRAMSKA KODA ARDUINO:

/**

Avtor: Aleksander Pešec, Aron Koca

Februar 2015

Program šteje hitrost kolesa

Led je povezana na APIN2

Led je prižgana ko senzor zbira podatke

Senzor

VCC - Analog Pin 1

GND - GND

O/P - Analog Pin 0

10k upor med O/P in VCC

LED

Katoda - povezana na zemljo preko doljenega upora (Ni po potrebi)

Anoda - Analog Pin 2

**/

```
long startTime = 0;
```

```
int state, prevState = 0;
```

```
int sensor = 0;
```

```
long counter = 0;
```

```
int ledPin = 12;
```

```
int state1 = 0;
```

```
int flag = 0;
```

```
void setup()
{
  pinMode(A2, OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  // Definirani pini
  Serial.begin(9600);
  // serijska s katero najin bluetooth modul upravlja povezave
}

void loop()
{
  if(Serial.available() > 0){
    state1 = Serial.read();
    flag=0;
    // če so podatki poslani, jih prebere ter shrani
  }

  digitalWrite(13, HIGH);
  sensor = 0;
  state = 0;
  prevState = 0;
  counter = 0;
  // Reset spremeljivk

  startTime = millis();
```

```
digitalWrite(A1, HIGH);  
digitalWrite(A2,HIGH);  
  
// Senzor ter LED se vključi  
  
while((millis() - startTime) < 600)  
  
// Loop za čas 600  
  
{  
  
  sensor = analogRead(0);  
  
  if (sensor > 750)  
  
    state = 1;  
  
  else  
  
    state = 0;  
  
  // Za spremembo stanja števca  
  
  
  if(state != prevState)  
  
    counter++;  
  
    prevState = state;  
  
// števec za hitrost – primerjava novo z staro hitrostjo  
  
  }  
  
}  
  
  counter = counter / 2;  
  
  ; Serial.print(counter);  
  
Serial.print(" KM/H");  
  
// Prikaži KM/H  
  
  
delay(425);  
  
// zakasnitev
```

}