

Univerzitet u Beogradu
Elektrotehnički fakultet
Katedra za računarsku tehniku i informatiku



diplomski rad

Razvoj sistema za udaljeno upravljanje robotskom platformom

Mentor
dr Đorđe Đurđević

Student
Nemanja Stojoski 0041/2013

Beograd, 2017.

Apstrakt

U ovom radu predstavljen je razvoj sistema za udaljeno upravljanje robotskom platformom preko bežične lokalne mreže. Robotska platoftma poseduje kameru koja omogućava prenos slike u realnom vremenu, kao i senzor za merenje udaljenosti koji pomaže platformi pri izbegavanju potencijalnih prepreka. Takođe, platforma se može lako proširiti dodavanjem brojnih senzora i aktuatora koji bi bili potrebni za specifične slučajeve korišćenja ovakvog sistema. Pored kontrole platforme tastaturom, kao jedan metod upravljanja korišćena je i *Myo* narukvica za prepoznavanje pokreta. Ovaj sistem se tokom testiranja pokazao kao prilično pouzdan, a kontrola pomoću narukvice za prepoznavanje pokreta vrlo precizna, što je bio jedan od glavnih kriterijuma uspešnosti projekta.

Sadržaj

1	Uvod	3
2	Problem	5
3	Pregled postojećih rešenja	7
3.1	Kretanje platforme	7
3.2	Komunikacija korisnika sa platformom	9
3.3	Načini upravljanja	10
4	Projekat	11
4.1	Funkcionalna specifikacija	11
4.2	Arhitektura sistema	14
4.2.1	Arhitektura hardvera	14
4.2.2	Arhitektura softvera	17
4.3	Detalji implementacije	21
4.4	Tehničke karakteristike	23
5	Zaključak	24
6	Literatura	25

1. Uvod

Svakodnevni tehnološki napreci i istraživanja u proteklih par decenija, rezultirali su razvijanjem velikog broja uređaja na daljinsko upravljanje. Ovakvi uređaji, bez obzira da li su autonomni ili ih kontroliše udaljeni operater, koriste se za niz kopnenih, vazdušnih, podvodnih i svemirskih operacija u kojima bi direktno prisustvo ljudi bilo ili nemoguće ili veoma opasno. Cena hardverskih komponenti više ne predstavlja barijeru za kreiranje sistema, a okruženja za razvoj softvera postaju sve prilagođenija stručnjacima. Međutim sistemi su najčešće fokusirani na vrlo usko određene oblasti, tj. stručnjaci se pretežno bave kreiranjem specifičnih zatvorenih platformi za potrebe veoma malog broja korisnika.

U periodu od samo jednog veka od svoje prve primene, udaljeno upravljanje uređajima postalo je neizbežni deo komunikacije sa velikim brojem elektronskih aparata. Nikola Tesla je kreirao jedan od prvih bežičnih uređaja na daljinsko upravljanje, čiju je primenu premijerno demonstrirao u Njujorku 1898. godine [1]. Svoj uređaj Tesla je koristio da bi radio talasima upravljaо malim brodom sa metalnom antenom koja je primala tačno određenu radio frekvenцију. Signali su slati brodu uz pomoć kutije koja je posedovala polugu i telegrafski taster koji je prethodno služio za slanje Morzeovog koda. Navedeni signali su pomerali električne kontakte na brodu, i na taj način omogućavali da operater podesi postavku kormila i propelera, i tako upravlja kretanjem broda. Rad na uređajima za udaljeno upravljanje nastavio je španski inženjer Leonardo Torres-Quevedo koji je koristio bežične telegrafske transmitere za kontrolu tricikala, brodova i podmornica [2].

Dalji razvoj tehnologija za daljinsko upravljanje bio je pretežno vezan za vojnu industriju čiji je napredak u dvadesetom veku bio posebno izražen. Tokom Prvog svetskog rata, nemačka mornarica je daljinski upravljala brodovima napunjenim eksplozivom, da bi napadala protivničke brodove [3]. Daljinsko upravljanje podrazumevalo je nastanak potpuno nove vrste oružja koja su omogućavala napad na protivnika koji se nalazi na velikoj udaljenosti, bez direktnog ugrožavanja sopstvenih snaga. U Drugom svetskom ratu, nemačke i američke trupe su nastavile sa eksperimentima u kojima su pokušavale da kreiraju niz daljinski navođenih torpeda i projektila.

Tokom 40-ih godina dvadesetog veka daljinsko upravljanje je fokusirano na olakšavanje procesa korišćenja elektronskih uređaja za široku potrošnju, na primer uređaja za otvaranje vrata garaže, za navođenje modela aviona ili za upravljanje radiom. Veliki pomak je nastao 1950. godine, kada je proizvođač elektronskih aparata „Zenith“ kreirao prvi uređaj za daljinsko upravljanje televizorima, što je značajno povećalo interesovanje za dalji napredak i usavršavanje ove tehnologije.

U današnje vreme, daljinsko upravljanje je napredovalo toliko da je postalo jedna od podrazumevanih karakteristika brojnih tehničkih aparata, koji variraju od igračaka i različitih konzola za video igre pa sve do uređaja za rashlađivanje i grejanje prostorija. Poseban pomak u pogledu dostupnosti ovih tehnologija predstavlja činjenica da je sada, uz jednostavnu instalaciju nekoliko aplikacija, moguće pametni telefon pretvoriti u univerzalni uređaj za upravljanje nizom digitalnih proizvoda.

Udaljeno upravljanje tehnološkim uređajima primenu je pronašlo i u mnogim naučnim projektima. Vojne tehnologije za usmeravanje municije, eksplozivnih naprava i projektila su doveđene do veoma visokih nivoa preciznosti, kao i bespilotni dronovi i oklopna vozila. U američkoj svemirskoj agenciji NASA razvijane su i tehnologije za upravljanje roverima i sondama na udaljenostima koje su do sada smatrane nedostižnim. Podvodna vozila sa udaljenim upravljanjem postala su posebno primenljiva u poslovima poput istraživanja rezervi fosilnih goriva na velikim dubinama, zbog svoje pokretljivosti, dobrog podnošenja ekstremnih uslova i mogućnosti dodavanja velikog broja alata za snimanje i manipulisanje okruženjem.

Interakcija između korisnika i tehničkih aparata postala je posebna oblast koja obuhvata izučavanje dizajna, psihologije i softverskog inženjerstva. Radi poboljšanja ove interakcije u proteklih par decenija razvijen je niz tehnologija koje omogućavaju povećanje fizičke aktivnosti korisnika prilikom korišćenja digitalnih uređaja. Noviji uređaji za daljinsko upravljanje su raznovrsni i mogu se nositi, prepoznavati pokrete, primenjivati koncepte virtualne ili proširene stvarnosti. Iako svi korisnički interfejsi podrazumevaju neki nivo fizičke aktivnosti, čak i ako je to samo pritisak dugmeta, noviji tehnološki principi proširuju repertoar dostupnih akcija tako da uključe pokrete celog tela, pomeranje objekata po površinama ili pomeranje celog uređaja kroz prostor. Cilj napora koji se ulaže u istraživanje najboljih načina za integraciju mimike i kretanja korisnika u tehnološka dostignuća je načiniti njihovu interakciju prirodnjom, eksprezivnijom i sveprisutnjom.

Ostatak ovog rada organizovan je na sledeći način. U drugom poglavlju izložen je konkretan problem koji može biti rešen upotrebom realizovane platforme. U trećem poglavlju je dat opšti pregled postojećih rešenja izgradnje robotske platforme. U četvrtom poglavlju je opisana izrada projekta, sa zasebnim odeljcima koji detaljnije objašnjavaju: način korišćenja platforme od strane korisnika, arhitekturu hardverskih i softverskih elemenata sistema, detalje implementacije kao i tehničke karakteristike sistema. Peto poglavlje sumira rezultate rada i opisuje moguće pravce nadogradnje sistema i daljeg istraživanja ove oblasti.

2. Problem

Cilj ovog rada bilo je osmišljavanje i izgradnja jednostavnog sistema za udaljeno upravljanje robotskom platformom. Pri kreiranju navedenog sistema fokus je bio na sledećim oblastima:

- istraživanje trendova za izradu robotskih platformi,
- testiranje dostupnih tehnologija i komponenata i njihovih performansi pri kreiranju prototipa sistema,
- istraživanje mogućnosti primene udaljenog upravljanja pri izradi robotskih platformi,
- primena kreirane robotske platforme u promenljivim uslovima u okruženju.

Već više od 50 godina robotske platforme se koriste sa ciljem da ljudima olakšaju teške, jednolične i opasne poslove. U tom periodu je uočen značajan napredak u korišćenim tehnologijama u smislu izrade, održavanja, sigurnosti korišćenja i upotrebljivosti. Osim navedenih dostignuća, razvijeni su i napredni sistemi za snimanje, komunikaciju, prikazivanje i interakciju sa okruženjem. U današnje vreme, neki vidovi robotskih platformi primenjuju se u gotovo svim industrijskim i naučnim oblastima, između ostalih: proizvodnji produkata za široku potrošnju, transportu, 3D štampanju, obezbeđenju objekata, poljoprivredi i zdravstvenoj zaštiti. Prilikom izrade ovog rada želja je bila da se kreira prototip jednostavne i prilagodljive platforme, koja ne bi bila dostupna samo velikim korporacijama, već bi je mogli koristiti i nadograđivati i svi zainteresovani pojedinci.

Predviđeno je i uspostavljanje pouzdane komunikacije između različitih celina sistema, bez obzira da li će se ona obavljati žično ili preko mreže. Tokovi komunikacije koje je bilo potrebno realizovati su: komunikacija između klijentskog interfejsa i glavnog računara platforme, komunikacija između glavnog računara platforme i mikrokontrolera zaduženog za interakciju sa hardverom, kao i upravljanje hardverom od strane mikrokontrolera. Postojanje ovogliko zasebnih tokova komunikacije u velikoj meri doprinosi složenosti sistema i povećava mogućnost otkaza u bilo kojoj tački komunikacije, o čemu se moralo voditi računa prilikom implementacije sistema.

Jedan od naknadno uočenih ciljeva prilikom izrade rada bilo je i istraživanje i testiranje mogućnosti kontrolisanja platforme nestandardnim metodama. Daljinsko upravljanje putem različitih uređaja koji omogućavaju prenos naredbi koje korisnik sprovodi efikasno i tačno postaje podrazumevana karakteristika velikog broja elektronskih uređaja. Prilikom istraživanja novijih metoda udaljenog upravljanja fokus je najčešće na njihovoj kompatibilnosti sa različitim softverskim rešenjima, a trebalo bi se posvetiti i istraživanju njihove interakcije sa hardverskim komponentama. Osnovna testiranja interakcije sa hardverom dalje su istraživana u kasnijim poglavljima.

Robotske platforme su često implementirane tako da je upravljanje njima moguće isključivo dok ih operater direktno vidi i dok sam procenjuje njihovu udaljenost od ciljeva ili prepreka. Prilikom izrade ovog rada korišćeni su kamera i ultrazvučni senzor koji služe za procenu udaljenosti od okolnih objekata. Na taj način se operateru daje mogućnost da u realnom vremenu, čak i u uslovima smanjene vidljivosti, pokreće robotsku platformu bez gubljenja informacija o njenoj tačnoj poziciji i opasnosti da je ošteti nekim objektom van sopstvenog vidnog polja.

Na kraju, potrebno je još jednom napomenuti da robotska platforma kreirana prilikom izrade ovog rada nije gotovo rešenje opremljeno za rad u svim situacijama koje se u praksi mogu javiti. Zbog toga fokus neće biti na njenoj zaštiti i održivosti pri različitim nepogodama i nepredviđenim situacijama, već na kreiranju osnovne, prototipske verzije, čiji razvoj može biti

nastavljen u budućnosti. Ovaj sistem bi trebalo da služi isključivo kao demonstracija da je i sa ograničenim materijalnim i vremenskim resursima moguće kreirati funkcionalno rešenje koje se može dalje unapređivati.

3. Pregled postojećih rešenja

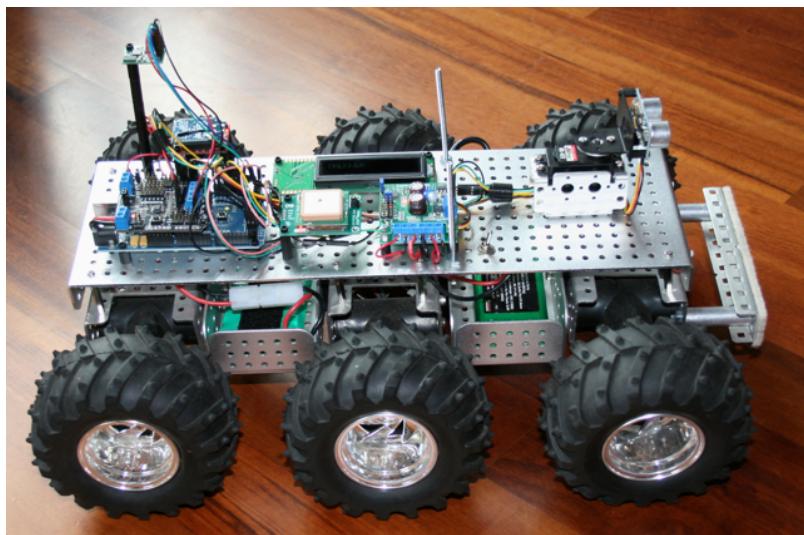
Pri izgradnji sistema za udaljeno upravljanje robotskom platformom potrebno je doneti veliki broj odluka u vezi sa projektom, koje u velikoj meri utiču na njegov dalji razvoj. Ključne odluke pri kreiranju ovakvih robota su: odabir odgovarajućeg načina kretanja platforme, komunikacija korisničkog podsistema za upravljanje samom platformom i način kontrole platforme.

3.1 Kretanje platforme

U ovom odeljku su opisani glavni načini prostornog pokretanja robota koji se kreću po tlu [4]. Naravno, postoje i roboti koji se mogu kretati u vazduhu ili pod vodom, ali oni nisu predmet razmatranja ovog rada. Za svaki pojedinačni način kretanja, u okviru dela u kome su izložene njegove karakteristike, biće prikazana i slika primera platforme koja primenjuje navedeni način kretanja.

Točkovi

Pomeranje platforme pomoću točkova je najpopularniji vid kretanja robota. Obično se se koriste tri točka, gde su dva povezana sa motorima, a treći pruža stabilnost platformi. Roboti sa četiri ili šest točkova mogu koristiti više od dva motora, čime platforma postaje sposobnija za manevriranje na mekanim ili vlažnim terenima. Primer takvog robota prikazan je na slici 3.1. Prednosti platforme sa točkovima su mala složenost i cena u odnosu na ostale načine pokretanja. Takođe postoji mnogo različitih izbora motora i točkova, pa se platforma može prilagoditi različitim korisničkim zahtevima. Nedostatak ovakve platforme je mogućnost proklizavanja zbog male kontaktne površine sa podlogom na kojoj se nalazi.



Slika 3.1: Primer platforme sa šest točkova

Gusenice

Kretanje pomoću gusenica rešava neke od problema sa kojima se suočavaju platforme pokretane pomoću točkova. Konstantni kontakt gusenica sa tlom sprečava proklizavanje platforme. Pored ovoga, postojanje gusenica bolje raspoređuje težinu platforme, pa omogućava kretanje po površinama koje bi bile nepristupačne platformama zasnovanim na točkovima (npr. kretanje

po snegu, pesku ili blatu), što je ilustrovano na slici 3.2. Nedostaci ovakvog tipa kretanja su povećanje mehaničke kompleksnosti sistema i mali broj dostupnih gotovih gusenica, što ograničava prilagođavanje platforme korisniku. Takođe, ukoliko je robot težak, gusenice mogu oštetiti platformu po kojoj se kreće, što može predstavljati problem ako se robot kreće u zatvorenom prostoru.



Slika 3.2: Primer robota sa gusenicama, koji se kreće po pesku

Noge

Roboti sa nogama predstavljaju najsloženiji način kretanja zemljanih platformi. Postojanje nogu ne podrazumeva samo čovekoliko, bipedalno kretanje, već uključuje i kretanje sa četiri ili šest nogu. Roboti sa tri ili više nogu su mnogo jednostavniji za kontrolu od bipedalnih, budući da lakše održavaju ravnotežu. Ovakve platforme mogu zaobići veliki broj objekata koji im se mogu naći na putu, koje bi prethodno navedenim platformama predstavljale nepremostivu prepreku, pa su dobar izbor za vrlo grube terene. Uprkos ovome, roboti sa nogama zahtevaju veliku mehaničku, elektroničku i programsку složenost. Takođe cena izgradnje ovakvih robota je viša zbog velikog broja potrebnih delova. Primer robota sa četiri noge prikazan je na slici 3.3.



Slika 3.3: Primer napredne platforme sa četiri noge

3.2 Komunikacija korisnika sa platformom

U ovom odeljku su opisani najzastupljeniji vidovi komunikacije između platforme i udaljenog operatera [5].

Žičana

Kontrola pomoću žice predstavlja najjednostavniji vid udaljene kontrole platforme. Jedna od prednosti žičane komunikacije je mogućnost napajanja robota preko žice, čime se eliminiše potreba za baterijama. Takođe, ovaj vid komunikacije je vrlo jednostavan, a prenos informacija preko žice je pouzdaniji od bilo kakvog bežičnog prenosa. Sa druge strane, maksimalna udaljenost platforme od operatera je ograničena dužinom žice, a žica se može zapetljati ili čak i prekinuti, čime bi se onemogućila dalja komunikacija sa robotom.

Radio komunikacija

Radio komunikacija predstavlja najjednostavniji tip bežične kontrole platforme. Ovakav tip komunikacije se koristi u maketama automobila i aviona na daljinsko upravljanje. Radio komunikacija obezbeđuje kontrolu platformi na veoma velikim udaljenostima, što je čini vrlo pogodnom za korišćenje. Glavni nedostatak ovog tipa bežične komunikacije jeste vrlo mala brzina prenosa podataka u odnosu na druge vidove bežične komunikacije. Takođe, može doći do interferencije signala ukoliko više korisnika komunicira sa svojim uređajima na istoj frekvenciji.

Bluetooth

Komunikacija Bluetooth protokolom rešava neke od problema koji se javljaju kod razmene informacija putem jednostavne radio komunikacije. Bluetooth protokol zahteva direktno uprivanje dva uređaja, čime se izbegava interferencija sa drugim korisnicima. Takođe, Bluetooth omogućava veću brzinu prenosa podataka, ali po ceni drastično smanjene udaljenosti komunikacije, koja ovim protokolom iznosi maksimalno 10 metara.

Wi-Fi

Komunikacija putem Wi-Fi tehnologije zahteva postojanje bežične lokalne mreže na koji su povezani korisnik i platforma. Ovaj vid bežične komunikacije je nešto složeniji od prethodnih, budući da pri podešavanju zahteva dodatni programerski napor. Maksimalna udaljenost platforme i korisnika zavisi od jačine odašiljača bežične mreže, a ukoliko se iskoristi već postojeća mrežna infrastruktura korisnik može sa bilo koje tačke na svetu putem Internet protokola kontrolisati platformu koja je u dometu nekog odašiljača. Takođe, uz izbor odgovarajućeg odašiljača signala, Wi-Fi pruža najveće brzine prenosa podataka u poređenju sa ostalim vidovima bežične komunikacije.

3.3 Načini upravljanja

Ručni kontroleri

Pod terminom „ručni kontroler” podrazumeva se uređaj koji sadrži tastere, prekidače ili palice, koji služe za upravljanje udaljenom robotskom platformom. Ovo predstavlja najstariji način udaljenog upravljanja, i uključuje uređaje kao što su radio transmiteri i džoystici. Takođe ručni kontroleri mogu biti i upravljači iz konzola za video igre koji se pored svoje osnovne namene, mogu iskoristiti za jednostavnu kontrolu robotske platforme. Svi ovi kontroleri omogućavaju vrlo preciznu i intuitivnu kontrolu platforme, ali se pri dizajniranju same platforme i odabiru kontrolera mora paziti da se sve komande koje platforma može primiti mapiraju na kontroler, kako bi udaljena kontrola bila moguća.

Miš i tastatura

Kontrola pomoću miša i tastature, za razliku od ručnih kontrolera, zahteva postojanje personalnog računara kao posrednika u komunikaciji. Na računaru obično postoji interfejs koji tumači unos od strane miša i tastature i prosleđuje ih platformi. Uz dobro realizovan interfejs, ovakav vid kontrole omogućava izdavanje mnogo većeg broja komandi od ručnih kontrolera, ali je preciznost kontrole obično smanjena.

Ostali načini upravljanja

Prethodni načini upravljanja postoje već decenijama. Međutim, razvojem tehnologije mogući vidovi interakcije čoveka sa računarima su se drastično povećale. Samim tim, metode upravljanja robotskim platformama su takođe mnogobrojnije. U nastavku će biti dato nekoliko primera novijih načina kontrole robotskih platformi.

Leap Motion je uređaj koji uz pomoć niza senzora omogućava prepoznavanje položaja šake i pokreta prstiju bez fizičkog kontakta [6]. Uređaj se sastoји od dve monohromatske infracrvene kamere i tri infracrvene svetleće diode, koje osvetljavaju prostor ispred uređaja. Uređaj prepoznaće svetlost odbijenu o šake i šalje te informacije na dalju obradu. Pomoću *Leap Motion*-a se može pratiti položaj ruke i šake korisnika, koji robotska ruka potom imitira [7].

Kinect uređaj je razvila kompanija Microsoft kao dodatak za njihove *Xbox* konzole za video igre [8]. Ovaj uređaj pomoću kamera i infracrvenih senzora prepoznaće pokrete korisnika u prostoru, čime se omogućuje korisniku da pomeranjem ruku ili celog tela interaguje sa računaram. Kompanije i istraživači koji razvijaju humanoidne robote obično koriste *Kinect* za preslikavanje pokreta upravljača na pokrete robota [9].

Jedan od novih načina interakcije sa računarima predstavlja tumačenje električnih impulsa koji proizvode mišići ruke. Jedan uređaj koji je zasnovan na tom principu je i *Myo*, razvijen u kompaniji Thalmic Labs [10], koji će biti detaljnije opisan u narednom poglavljju.

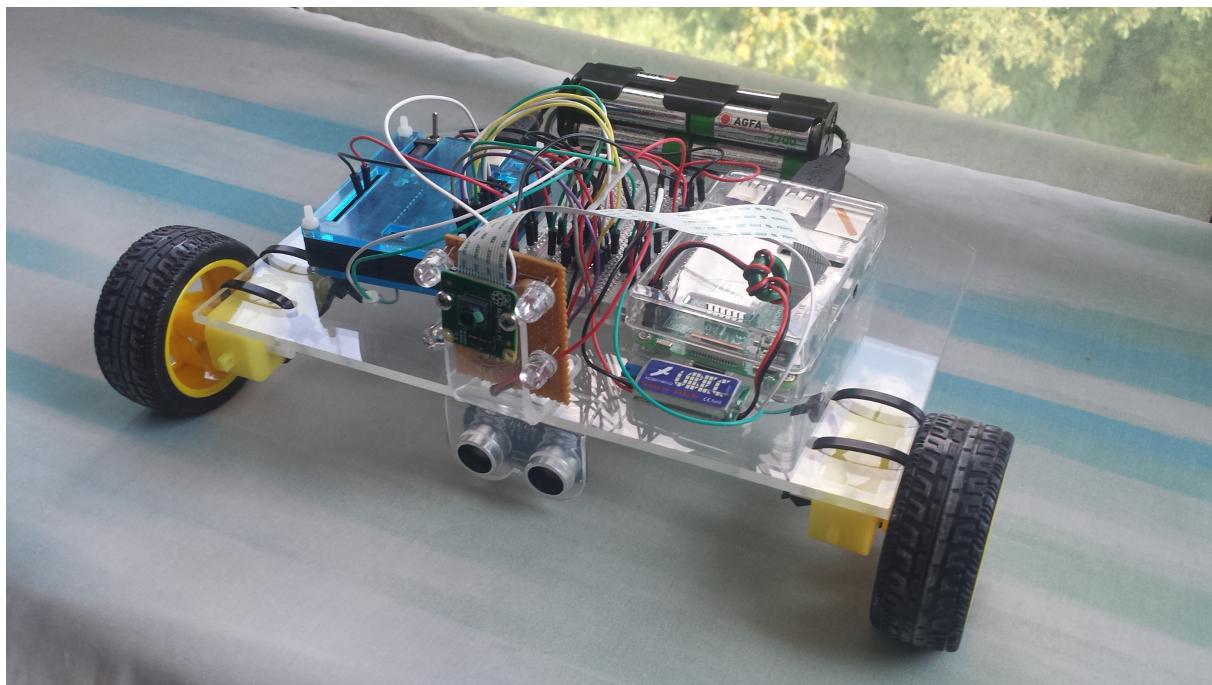
4. Projekat

4.1 Funkcionalna specifikacija

U ovom odeljku je opisan način korišćenja sistema za udaljeno upravljanje robotskom platformom. Prvo je navedeno od kojih komponenti se sastoji ceo sistem, a zatim je opisano pokretanje celog sistema kao i interakcija korisnika sa prethodno pokrenutim sistemom.

Sistem se, na najvišem nivou, sastoji iz robotske platforme i veb prezentacije preko koje se korisniku omogućava kontrola ove platforme. Platformu, pored mehaničkih komponenti sačinjavaju i dva procesirajuća elementa: Raspberry Pi računar [11], koji tumači komande primljene od korisnika i izdaje naredbe drugom procesirajućem elementu - mikrokontroleru koji se nalazi na Arduino platformi [12]. Ovaj mikrokontroler interaguje sa svim mehaničkim komponentama platforme. Detaljnija arhitektura sistema biće data u odeljku 4.2.

Pomeranjem prekidača struja iz baterije počinje da napaja Raspberry Pi, čime sistem postaje spreman za rad. Raspberry Pi će pri pokretanju pokušati da se poveže na neku od prethodno zapamćenih bežičnih mreža. Ukoliko nijedna bežična mreža nije dostupna, Raspberry Pi se mora povezati Ethernet kablom na neku žičanu mrežu i tako konfigurisati za povezivanje na bežičnu. Drugi način za povezivanje je pristupanje grafičkom korisničkom interfejsu putem HDMI kabla, i konfigurisanje parametara za konekciju sa bežičnom mrežom. Na slici 4.1 je prikazan izgled konstruisane robotske platforme. Na prednjem delu platforme se nalaze kamera, četri svetleće diode i senzor udaljenosti. Na levoj i desnoj strani prednjeg dela platforme se nalaze motori i točkovi. U centralnom delu platforme, na levoj strani slike se nalazi Arduino platforma (u plavom kućištu), a sa desne Raspberry Pi.

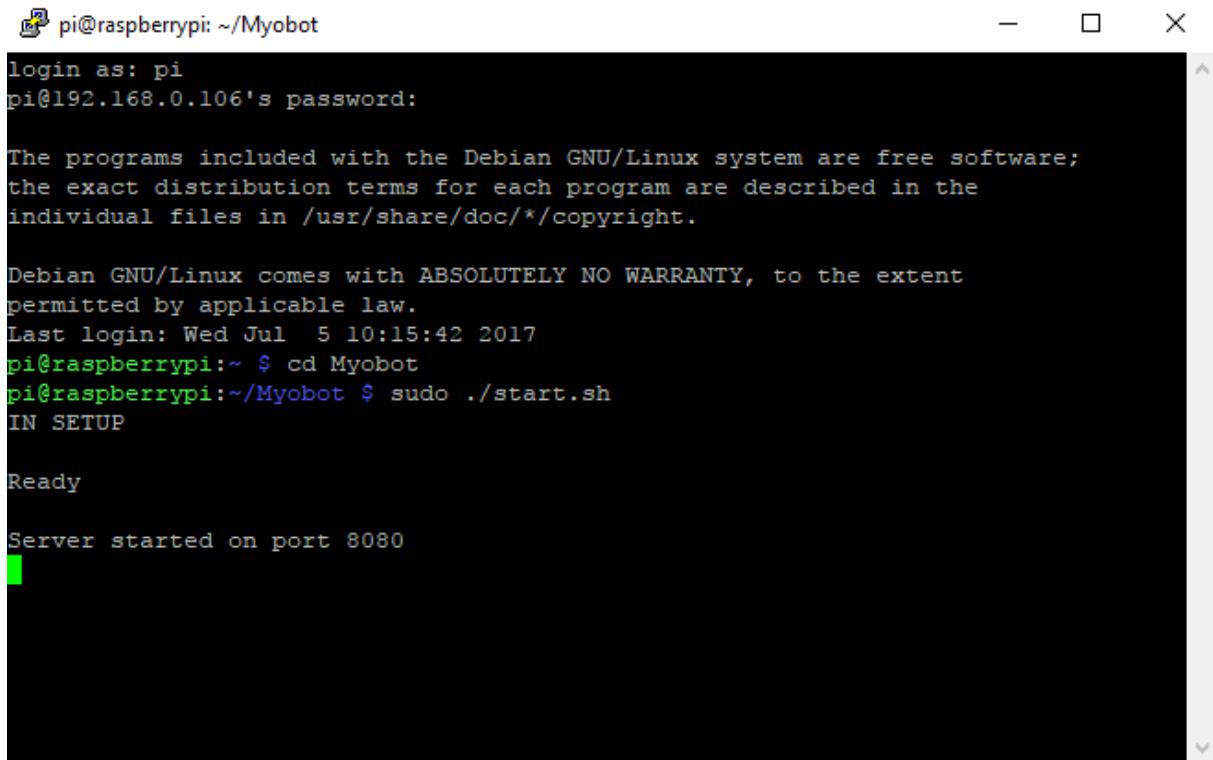


Slika 4.1: Prikaz ugašene platforme konstruisane za potrebe ovog rada

Nakon povezivanja na bežičnu mrežu, sistemu se mora pristupiti putem SSH ili VNC protokola, čime se dobija pristup konzoli ili grafičkom korisničkom interfejsu Raspberry Pi-a.

Zatim je potrebno doći do home direktorijuma ulogovanog korisnika, gde se nalazi Myobot direktorijum, koji predstavlja koren direktorijum sistema za upravljanje platformom. U

korenom direktorijumu se nalazi skripta `start.sh`, čijim izvršavanjem se pokreće server koji omogućava pristup sistemu za udaljeno upravljanje ovom platformom. Slika 4.2 prikazuje logovanje na Raspberry Pi računar putem SSH protokola, korišćenjem PuTTY alata, i naknadno pokretanje servera.



```
pi@raspberrypi: ~/Myobot
login as: pi
pi@192.168.0.106's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

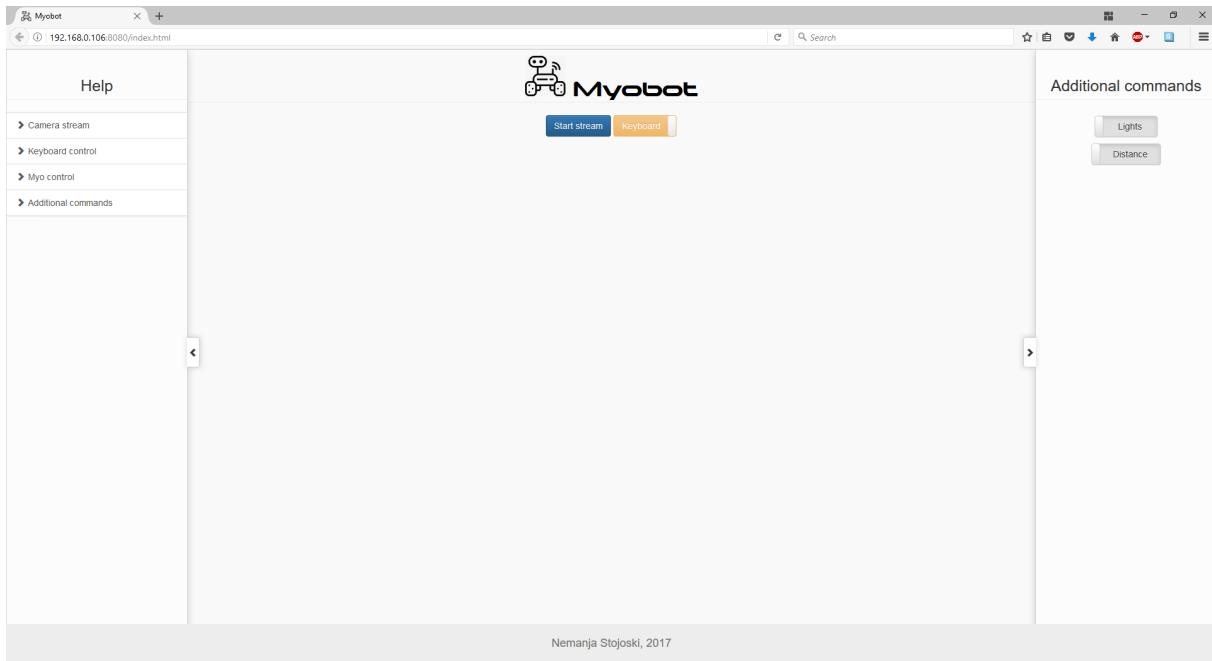
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul  5 10:15:42 2017
pi@raspberrypi:~ $ cd Myobot
pi@raspberrypi:~/Myobot $ sudo ./start.sh
IN SETUP

Ready

Server started on port 8080
```

Slika 4.2: Prikaz logovanja na Raspberry Pi i pokretanja sistema

U okviru korisnikovog veb pregledača potrebno je uneti adresu povezanog Raspberry Pi-a, kao i port na kome je pokrenut server. Dolaskom na ovu adresu korisniku se prikazuje veb prezentacija sistema. Na slici 4.3 je prikazana početna stranica korisničkog interfejsa za udaljenu kontrolu platforme. Prikaz slike je isključen, a pomoći meni i meni sa dodatnim komandama su otvoreni.

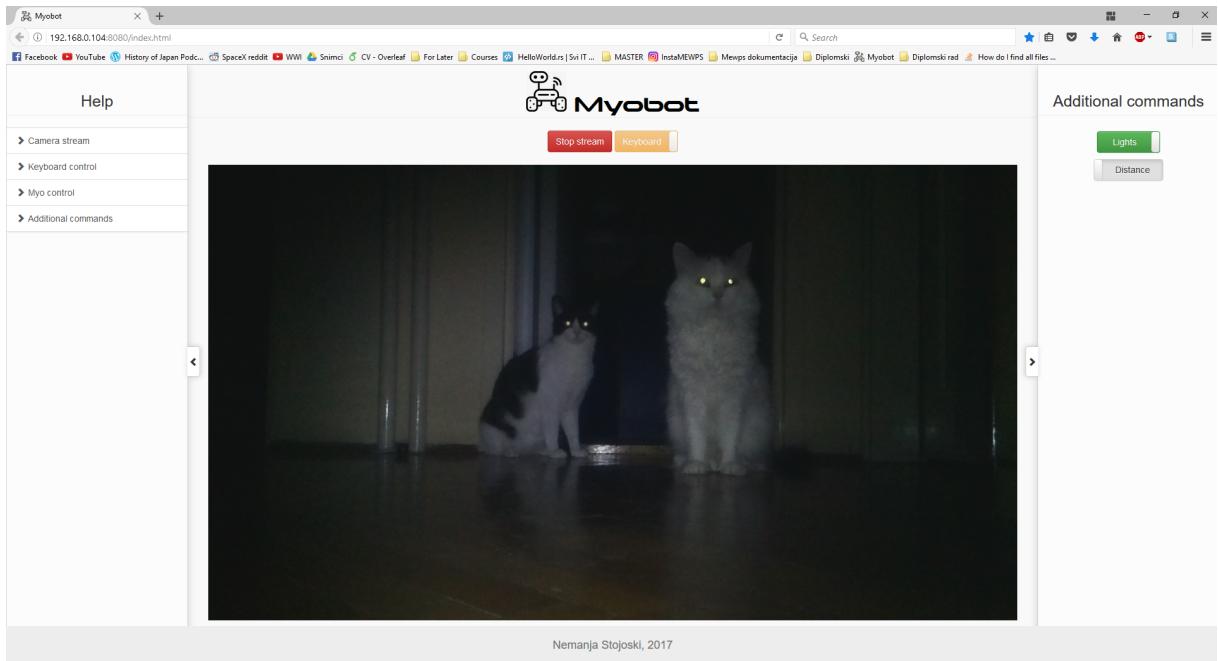


Slika 4.3: Početna stranica korisničkog interfejsa

Navedena veb prezentacija korisniku omogućava pristup kontrolama za upravljanje platformom. U bilo kom trenutku, korisnik može pokrenuti ili zaustaviti prenos slike sa kamere. Korisniku se takođe daje mogućnost odabira načina kontrole platforme. Podrazumevan način kontrole je pomoću tastature. Ukoliko korisnik poseduje *Myo* narukvicu i na svom računaru ima instaliran *Myo Connect* program za povezivanje sa narukvicom, nakon stavljanja narukvice i njene sinhronizacije sa *Myo Connect* programom korisniku se omogućava kontrola platforme pomoću narukvice. Ukoliko korisnik odabere ovaj način kontrole platforme, zahtevi za pokretanjem platforme šalju se pravljjenjem predefinisanih pokreta šake.

Pored prenosa slike i kontrole platforme, korisniku je dostupan meni sa dodatnim opcijama, koji se nalazi na desnoj strani prezentacije. U tom meniju korisnik ima mogućnost da od platforme zatraži trenutno izmerenu udaljenost od prepreke koja se može nalaziti ispred platforme. Merenje je podešeno tako da vraća udaljenosti do 1 m. Takođe, korisnik može upaliti ili ugasiti svetla na platformi, čime se omogućava rad platforme u mračnim područjima.

Na levoj strani prezentacije korisniku je dostupan pomoćni meni koji objašnjava mogućnosti platforme i služi kao uputstvo za korišćenje sistema. Slika 4.4 prikazuje korisnički interfejs sa uključenim prikazom slike sa kamere i aktiviranim svetlećim diodama, za rad u slabo osvetljenom okruženju.



Slika 4.4: Prikaz rada u slabo osvetljenom okruženju, na primeru Baleriona i Garfilda

4.2 Arhitektura sistema

U ovom odeljku je opisana detaljna arhitektura hardverske i softverske celine projekta.

4.2.1 Arhitektura hardvera

Raspberry Pi

Raspberry Pi je „single-board”¹ računar zasnovan na 64-bitnom ARMv8 procesoru [11]. Za potrebe ovog rada, na njega je instaliran Raspbian operativni sistem. Raspbian je operativni sistem zasnovan na Linux Debian operativnom sistemu i optimizovan za hardver Raspberry Pi-a [13]. Korišćena verzija ovog računara je Raspberry Pi 3 Model B, koja je izabrana zato što posedebe bežičnu lokalnu mrežnu konekciju (Wi-Fi), koja je potrebna radi udaljenog povezivanja sa robotskom platformom. U ovom projektu, Raspberry Pi ima dve svrhe: funkcioniše kao server koji opslužuje korisnika povezanog preko lokalne mreže, i na osnovu zahteva primljenih od korisnika izdaje komande Arduino mikrokontrolerskoj platformi.

Arduino

Arduino je mikrokontrolerska platforma otvorenog hardvera i otvorenog koda koja služi za razvoj elektronskih projekata. Korišćena verzija ove platforme je Arduino Uno Rev3, zasnovana na Atmel ATmega328P mikrokontroleru [12]. Odabrana je za izradu ovog projekta zbog jednostavnosti programiranja, kompatibilnosti sa velikim brojem senzora kao i dostupnosti obimne dokumentacije. Sadrži 14 digitalnih input/output pinova, od kojih 6 mogu da se koriste kao PWM² output pinovi, kao i 6 analognih input pinova. Ovi pinovi se koriste u projektu za

¹„single-board” računar: Računar malih dimenzija kod kojeg su sve komponente potrebne za rad računara zaledljene za jednu ploču.

²PWM: Pulse Width Modulation - Modulacija širine pulsa; u ovom projektu PWM pinovi se koriste kako bi se mogle podešiti različite brzine obrtaja motora: brzom aktivacijom i gašenjem kvadratnih talasa različitog trajanja, stiče se privid menjanja napona koji izlazi iz PWM pina, od 0V do 5V.

povezivanje sa senzorima i drugim komponentama. Arduino i Raspberry Pi su povezani USB kablom koji omogućava njihovu međusobnu komunikaciju, kao i napajanje Arduino-a strujom napona od 5V. Kod za Arduino platformu može da se piše u jezicima C i C++, uz poštovanje određenih pravila za strukturiranje koda.

Senzor udaljenosti

U ovom projektu je korišćen HC-SR04 ultrazvučni senzor udaljenosti [14]. Pored standardnih pinova za napajanje, ovaj ultrazvučni senzor poseduje *Trigger* pin za slanje ultrazvučnog talasa, kao i *Echo* pin koji se gasi kada senzor primi odbijeni talas. Princip funkcionisanja je sledeći: *Trigger* pin se drži aktivnim najmanje 10 mikrosekundi nakon čega senzor emituje 8 kratkih ultrazvučnih pulseva frekvencije 40 kHz i aktivira *Echo* pin. Ovaj pin se gasi kada senzor registruje odbijeni ultrazvučni talas, pa se, poznajući brzinu prostiranja mehaničkih talasa u vazduhu na određenoj temperaturi kao i dužinom trajanja aktivne vrednosti na *Echo* pinu može proceniti udaljenost objekta o koji su se odbili ultrazvučni talasi. Preciznost ovog senzora je do 3 mm, a može detektovati objekte udaljene od 2 cm do 400 cm.

Ostale komponente na platformi

Za pokretanje robotske platforme korišćena su dva DC motora sa reduktorom obrtaja, čiji je preporučeni radni napod od 3 V do 12 V. Za pokretanje motora je korišćen L293D čip koji se sastoji od dva H-mosta³, koji omogućavaju istovremeno kontrolisanje oba motora od strane Arduino-a [15].

Motore platforme i Raspberry Pi napaja 8 redno vezanih punjivih Ni-Mh AA baterija kapaciteta 2300 mAh. Svaka baterija je napona 1.2 V, pa je ukupni napon 9.6 V. Budući da Raspberry Pi zahteva napon od 5 V, mora se koristiti neka vrsta reduktora napona da bi sistem ispravno funkcionišao. Za to je iskorišćen UBEC (Universal Battery Elimination Circuit): uređaj koji smanjuje napon sa baterija (sa maksimalno 26 V) na potrebnih 5V [16]. Maksimalna jačina struje koju pruža je 3 A, što je dovoljno za rad celokupne platforme.

Za snimanje videa u realnom vremenu, korišćena je Raspberry Pi Camera, povezana na CSI (Camera Serial Interface) port Raspberry Pi-a [17]. Senzor ima rezoluciju od 8 megapiksela, i omogućava snimanje videa u 1920x1080 rezoluciji brzinom do 30 slika u sekundi.

Na platformi se takođe nalaze i četiri svetleće diode koje mogu služiti kao noćno svetlo, čime se platforma osposobljava za funkcionisanje u slabo osvetljenim područjima.

Myo narukvica

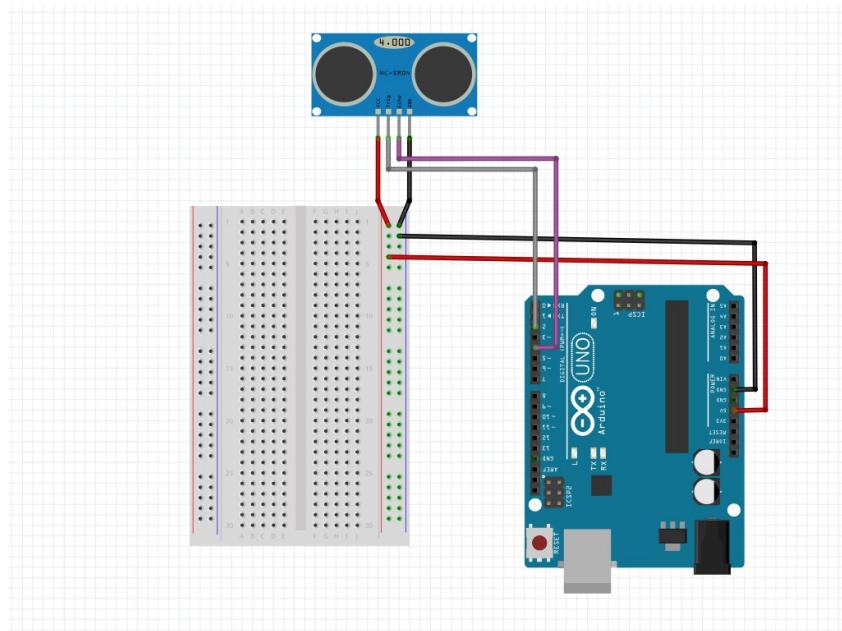
Kao jedan vid udaljene kontrole platforme korišćena je *Myo* narukvica za prepoznavanje pokreta. Prepoznavanje pokreta na ovoj narukvici se zasniva na dva principa [18]. Prvi je elektromiografija tj. merenje električnog potencijala ćelija u mišićima ruke. Budući da svaki položaj šake proizvodi jedinstven električni „potpis” u mišićima podlaktice, on se može prepoznati i zabeležiti određeni položaj šake. Drugi princip prepoznavanja pokreta zasniva se na određivanju orijentacije narukvice u prostoru, kao i njenog ubrzanja tokom vremena, čime se mogu prepoznati karakteristični pokreti cele ruke kao što je mahanje ili kruženje. Za potrebe ovog projekta korišćeno je samo prepoznavanje pokreta zasnovano na merenju električne aktivnosti mišića, uz prepoznavanje predefinisanih položaja šake.

³H-most - Električno kolo koje omogućava primenu napona na potrošača u različitim smerovima; u ovom projektu, H-mostovi u L293D čipu se koriste da bi motori mogli da se kreću unapred ili unazad. Postavljanjem napona na jedan pin L293D čipa motor će se okretati u jednom smeru, dok će se postavljanjem napona na drugi pin čipa motor okretati u suprotnom smeru.

Šeme povezivanja

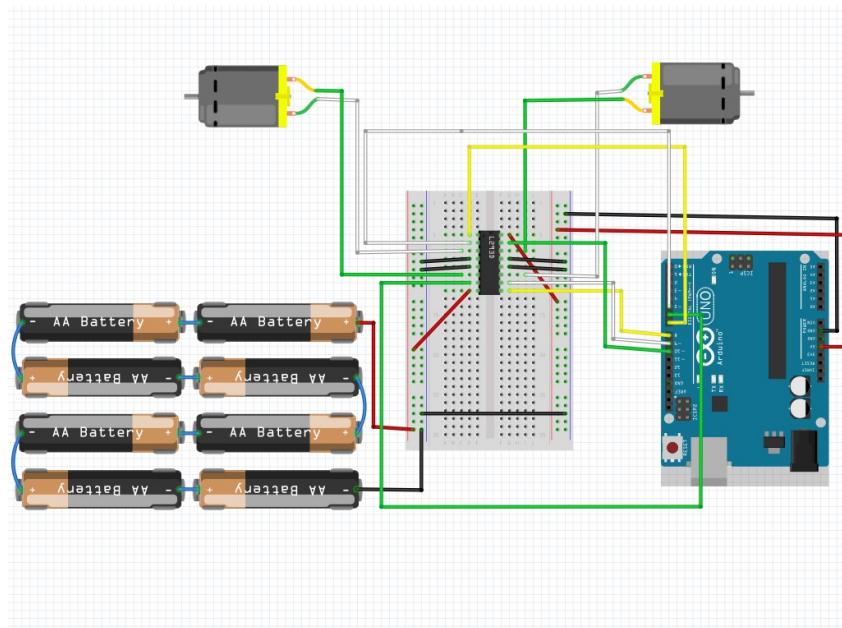
Ovde su date šeme povezivanja glavnih podsistema platforme: povezivanje Arduino-a sa senzorom udaljenosti, povezivanje sa motorima i baterijama, kao i podsistem za paljenje i gašenje svetlećih dioda.

Slika 4.5 prikazuje povezivanje Arduino-a sa HC-SR04 ultrazvučnim senzorom. Crvene žice predstavljaju napon, crne uzemljenje, siva žica je povezana sa *Trigger* pinom, a ljubičasta sa *Echo* pinom.



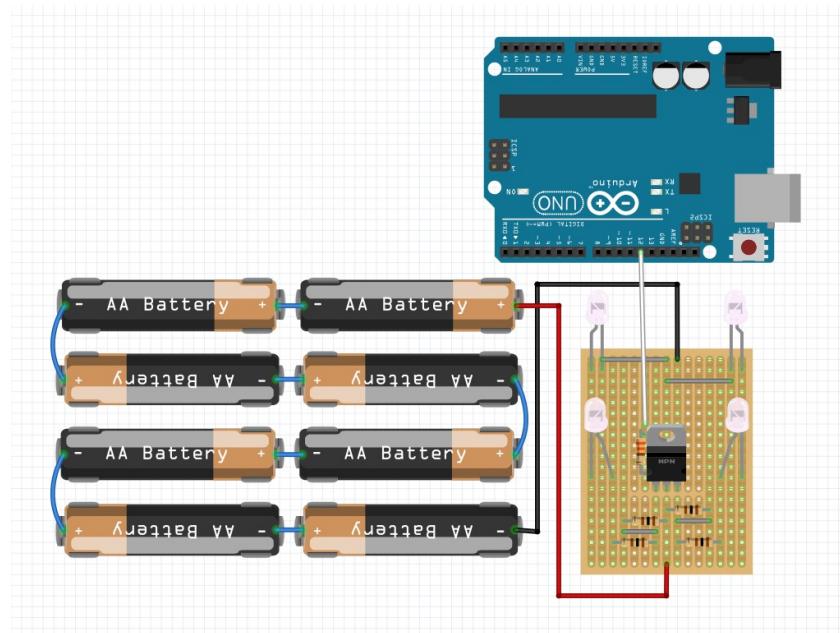
Slika 4.5: Povezivanje Arduino-a sa HC-SR04 ultrazvučnim senzorom.

Na slici 4.6 je prikazano povezivanje Arduino-a sa L293D čipom i njegovo povezivanje sa dva DC motora. Takođe je prikazano i dovođenje napona sa baterija i sa Arduino-a na odgovarajuće pinove L293D čipa. Crvene žice predstavljaju napon, crne uzemljenje, a ostale služe za povezivanje Arduino-a i čipa radi ispravne kontrole motora.



Slika 4.6: Podsistem za kretanje platforme.

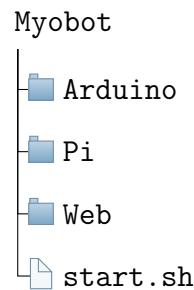
Na slici 4.7 je prikazano povezivanje baterija i Arduino-a sa četiri svetleće diode. Paljenje i gašenje dioda kontroliše FQP30N06L MOSFET tranzistor [19], koji u ovom kolu funkcioniše kao prekidač. Kada Arduino postavi napon sa svog pina 12 na *Gate* pin tranzistora, on dozvoljava protok struje iz baterija kroz diode, čime se one aktiviraju.



Slika 4.7: Povezivanje Arduino-a sa svetlećim diodama.

4.2.2 Arhitektura softvera

Softver ovog sistema se sastoji iz tri celine: koda koji se izvršava na Arduino platformi, koda koji se izvršava na Raspberry Pi računaru i koda koji se izvršava na klijentskom uređaju. Nakon svakog odeljka priložena je struktura direktorijuma koja odgovara toj celini, sa ključnim fajlovima i poddirektorijumima. Slika 4.8 prikazuje strukturu korenog direktorijuma projekta. U korenom direktorijumu se nalazi Bash skripta `start.sh` koja pokreće `main.py` fajl iz Pi direktorijuma. Ukoliko je ovoj skripti prenet opcioni argument „-u”, skripta će prevesti kod iz Arduino direktorijuma i učitati prevedeni kod na Arduino mikrokontroler. Skripta se mora pokrenuti od strane `root` korisnika komandom `sudo`, da bi se obezbedilo uspešno izvršavanje svih komandi u ovoj skripti, budući da neke od njih zahtevaju privilegije `root` korisnika.



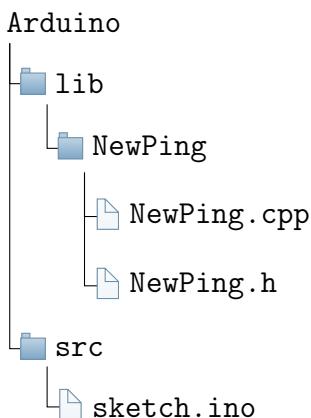
Slika 4.8: Struktura korenog direktorijuma projekta

Arduino

Kod na Arduino platformi se nalazi u jednom fajlu (`sketch.ino`). Pored pomoćnih funkcija, sadrži dve glavne funkcije koje moraju postojati u svakom programu koji se izvršava na Arduino platformi: `setup()` i `loop()`. Funkcija `setup()` se poziva jednom pri pokretanju Arduino uređaja, a zatim se funkcija `loop()` poziva neprekidno dok se Arduino ne ugasi ili se ne učita novi kod u njega. Slika 4.9 prikazuje strukturu Arduino poddirektorijuma.

U `setup()` funkciji se inicijalizuju pinovi koji se koriste za interakciju sa senzorom udaljenosti, pinovi za kontrolu motora, kao i pin za aktivaciju svetlećih dioda. Takođe, ovde se inicijalizuje serijska komunikacija putem USB kabla sa Raspberry Pi-em, podešavanjem brzine prenosa podataka koja mora biti postavljena na istu vrednost na oba kraja komunikacije.

U `loop()` funkciji se ponavljaju tri glavne celine: Prva je periodična provera da li je potrebno aktivirati senzor udaljenosti. On se aktivira ukoliko je Arduino primio START komandu za početak rada, ukoliko se platforma kreće napred ili korisnik zahteva merenje udaljenosti, i ukoliko je prošlo dovoljno vremena od prethodnog merenja udaljenosti (nekoliko desetina milisekundi). Ukoliko su ovi uslovi ispunjeni, Arduino aktivira senzor i postavlja funkciju koja će se pozivati prekidom internog tajmera i proveravati da li je izmerena udaljenost od senzora. Zatim se proverava poslednje vreme sticanja *heartbeat* signala od Raspberry Pi-a. Ukoliko novi signal ne stigne u određenom vremenskom roku to može značiti da je komunikacija između Raspberry Pi-a i Arduino-a na neki način kompromitovana. Motori u tom trenutku mogu biti uključeni, i ne menjaju svoje stanje dok ne stigne zahtev za njihovo gašenje. Ukoliko je komunikacija kompromitovana, zahtev možda nikad ne stigne, pa Arduino preventivno gasi motore da ne bi došlo do štete na platformi. Na kraju se proverava da li je stigla nova komanda od Raspberry Pi-a. O ovim komandama će biti više reči u delu rada posvećenom detaljima implementacije.



Slika 4.9: Struktura Arduino poddirektorijuma

Raspberry Pi

Na Raspberry Pi-u se izvršava najveći deo koda potreban za funkcionisanje ovog sistema. Pi poddirektorijum se sastoji se od 8 fajlova napisanih u Python programskom jeziku i jednog konfiguracionog fajla, što je prikazano na slici 4.10. U daljem tekstu sledi kratak opis funkcije svakog fajla.

- `config.ini`

Ovaj fajl sadrži konfiguracione parametre za Python kod koji se izvršava na Raspberry Pi-u. Ovde se nalaze parametri neophodni za povezivanje sa Arduinom (port za USB serijsku komunikaciju i brzina prenosa), parametri potrebni za pokretanje servera (port i dimenzije slike koja će prikazivati prenos sa kamere), kao i podešavanja za Raspberry Pi kameru (rezolucija, horizontalni i vertikalni obrt kamere).

- **shared.py**

Ovde se nalazi funkcija `init()` koja deklariše i definiše sve promenljive kojima se pristupa iz više fajlova. U ovoj funkciji se takođe vrši i učitavanje parametara iz konfiguracionog fajla. Svi fajlovi kojima je potreban pristup globalnim promenljivama uključuju ovaj fajl kako bi mogli da pristupe tim promenljivama.

- **comm_protocol.py**

U ovom fajlu se nalazi klasa `Communicator` koja služi za komunikaciju sa Arduino uređajem. Sem konstruktora, ova klasa sadrži metode `send()` i `receive()` koje se koriste za slanje i prijem podataka, respektivno. Metodu `send()` može pozivati više niti pa je potrebno zaštитiti njen kod jednim `Lock` objektom.

- **serial_receiver.py**

Ovde se nalazi klasa `SerialReceiver` čiji objekti predstavljaju zasebne niti izvršavanja programa. U njenoj `run()` metodi se tokom celog trajanja programa poziva `receive()` metoda iz prethodne klase, čime se čeka pristizanje vrednosti iz Arduino-a. Ukoliko ova metoda prepozna pristizanje informacije o izmerenoj udaljenosti, pomoću objekta klase `Event` obaveštava nit koja potencijalno čeka da pročita ovu informaciju da je ona stigla.

- **heartbeat.py**

Ovaj fajl sadrži klasu `Heartbeat` čiji objekti predstavljaju zasebne niti izvršavanja programa. U njenoj `run()` metodi se tokom celog trajanja programa šalje `heartbeat` signal nakon isteka određenog vremenskog intervala.

- **hardware.py**

U ovom fajlu se nalaze tri klase, `Distance`, `Motors` i `Lights`, koje predstavljaju interfejs za slanje zahteva Arduino platformi, koja potom interaguje sa hardverom na željeni način.

Klasa `Distance` sadrži metode `distance_on()` i `distance_off()` koje signaliziraju Arduino-u da započne ili zaustavi merenje udaljenosti na korisnički zahtev. Ovde se nalazi i metoda `get_distance()` koja traži od Arduino-a poslednju udaljenost izmerenu ultrazvučnim senzorom. Ova metoda se blokira na `Event` objektu, kojem `SerialReceiver` nit signalizira da se može nastaviti sa radom, kada pristigne izmerena udaljenost.

Klasa `Motors` sadrži metode za pokretanje motora platforme. Ključna metoda u ovoj klasi je `set_motor_powers()` koja prima dva parametra koji predstavljaju procentualnu jačinu željene aktivacije motora. Ukoliko su prosleđeni parametri u odgovarajućem opsegu ona šalje Arduino-u komandu za pokretanje motora, zajedno sa zahtevanom jačinom aktivacije motora. Pored ove metode, ovde se nalazi i metoda `go()` koja pored istih parametara kao i prethodna metoda prima i opcione parametre `stop` i `duration` (koji su podrazumevano postavljeni na `False` i `1`, respektivno). Metoda `go()` prvo poziva metodu `set_motor_powers()` i prosleđuje joj svoje parametre za aktivaciju motora. Dodatno, ukoliko je `stop` parametar postavljen na `True`, ova metoda uspavljuje pozivajuću nit onolikoj vremenu koliko je prosleđeno `duration` parametrom i nakon toga zaustavlja motore. Ostale metode u ovoj klasi pozivaju metodu `go()` prosleđujući joj potrebne vrednosti radi zaustavljanja platforme, kretanja napred, nazad, skretanja levo ili skretanja desno.

Klasa `Lights` sadrži metode `lights_on()` i `lights_off()` koje šalju odgovarajuće zahteve Arduino-u za paljenje i gašenje svetlećih dioda na platformi.

- **server.py**

U ovom fajlu se nalaze sve klase potrebne za uspostavljanje servera na lokalnoj mreži. Objekti Klase `StreamingOutput` sadrže bafer za primanje nove slike sa kamere, najskoriju

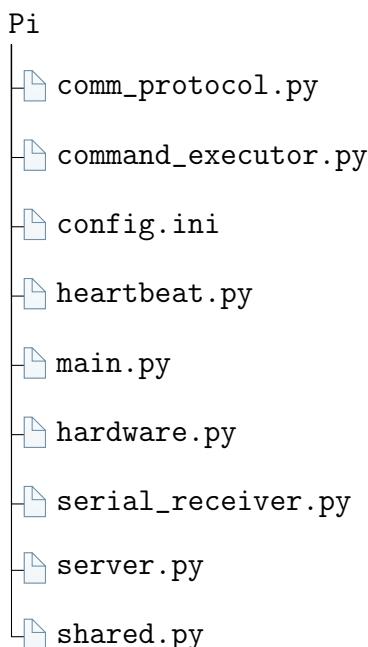
primljenu sliku, kao i objekat klase `Condition` koji služi za sinhronizovanje „proizvođača“ slike - kamere i „potrošača“ slike - klijenta koji šalje zahtev za novom slikom. Klasa `StreamingHandler` predstavlja osluškivač HTTP POST i GET zahteva. POST zahtevi predstavljaju zahteve poslate od strane klijenta AJAX protokolom, a GET zahtevi predstavljaju zahteve za novim HTML, CSS ili JavaScript stranicama, kao i zahteve za novom slikom sa kamere. Klasa `StreamingServer` je izvedena iz klase `server.HTTPServer` uz podešavanje da obrađuje svaki pristigli zahtev u zasebnoj niti. Pri konstrukciji, njoj se prosleđuje prethodno napravljeni `StreamingHandler` kao rukovalac zahtevima.

- `command_executor.py`

U ovom fajlu se nalazi klasa `CommandExecutor` sa metodom `interpret()` koja izvršava zahteve tumačeći poruku primljenu od klijenta AJAX protokolom. To mogu biti zahtevi za pokretanje ili zaustavljanje platforme, pokretanje ili zaustavljanje prenosa sa kamere, zahtev za dohvatanje najnovije vrednosti izmerene senzorom udaljenosti kao i zahtevi za paljenje ili gašenje svetla na platformi. Ovde se takođe nalazi i asocijativni niz parova ključeva i vrednosti, gde ključevima odgovaraju poruke dobijene od klijenta a vrednostima imena metoda koje trebaju da se izvrše primanjem odgovarajuće poruke. Ovo je urađeno da bi se ubrzalo izvršavanje komandi i uprostio kod izbegavanjem dugačkih `if/elif` konstrukcija.

- `main.py`

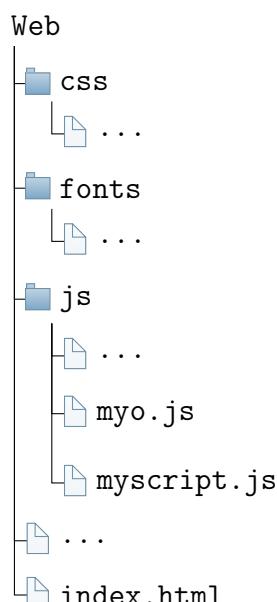
Ovde se nalazi glavna funkcija, `main()` koja se izvršava pri pozivu skripte za pokretanje sistema. U njoj se prvo poziva funkcija `init()` iz `shared.py` fajla, koja inicijalizuje sve deljene globalne promenljive. Nakon ovoga se pokreće nit za prihvatanje vrednosti sa Arduino-a (`SerialReceiver`) kao i nit za održavanje *heartbeat* signala (`Heartbeat`). Zatim se Arduino-u šalje komanda za početak komunikacije, i pokreće server (`StreamingServer`) koji osluškuje zahteve klijentata. Kada korisnik prekine izvršavanje programa šalje se signal prethodno pokrenutim nitima da prestanu sa izvršavanjem, čime se zaustavlja ceo program.



Slika 4.10: Struktura Pi poddirektorijuma

Web

Ovde se nalazi kod koji server šalje klijentu i koji se potom izvršava na klijentskom uređaju. Pored početne HTML stranice, `index.html`, i fajlova koji služe za stilizovanje veb prezenta-cije, ovde se nalazi i JavaScript fajl `myscript.js` koji sadrži funkcije potrebne za komunikaciju klijenta i servera AJAX protokolom kao i osluškivače događaja sa miša, tastature ili *Myo* narukvice, koji generišu AJAX zahteve. Pri kontroli tastaturom, pritiskom određenog slova generiše se AJAX zahtev za pokretanje motora. Pri otpuštanju tog slova generiše se zahtev za zaustavljanjem motora. Slično, pri kontroli pomoću *Myo* narukvice, kada se prepozna određeni pokret ruke šalje se AJAX zahtev za pokretanje motora, dok se prepoznavanjem prekida određenog pokreta šalje zahtev za zaustavljanjem motora. Takođe, pritiskom na odgovarajuću dugmad ili pravljenjem odgovarajućih pokreta ruke se šalju zahtevi za najskorijom vrednošću izmerenom senzorom udaljenosti, kao i zahtevi za paljenjem i gašenjem svetla na platformi. Slika 4.11 prikazuje strukturu Web poddirektorijuma.



Slika 4.11: Struktura Web poddirektorijuma

4.3 Detalji implementacije

U ovom odeljku su detaljnije opisani neki karakteristični delovi sistema.

Komande za Arduino

Komande koje Arduino može primiti od Raspberry Pi-a su:

- **START** - Ovu komandu Raspberry Pi šalje kada započinje sa radom, i čeka na odgovor od Arduino-a, čime je siguran da je Arduino aktivran i da je spremjan da prima ostale komande.
- **DISTANCE_ON** - Ovu komandu šalje Raspberry Pi kada klijent želi da prikaže vrednost izmerenu senzorom udaljenosti. **DISTANCE_ON** komandom se ne šalje sama udaljenost nazad na Raspberry Pi, već se postavlja fleg koji govori Arduino-u da u sledećem pozivanju `loop()` funkcije treba da aktivira senzor udaljenosti i zapamti izmerenu vrednost.
- **GET_DISTANCE** - Kada primi ovu komandu, Arduino šalje Raspberry Pi-u najskoriju vrednost izmerenu senzorom udaljenosti.

- **DISTANCE_OFF** - Ovom komandom se Arduino-u signalizira da klijent više ne želi da prima izmerene udaljenosti, pa merenje nastavlja da se vrši samo dok se platforma kreće napred, da bi se mogao sprečiti potencijalni sudar.
- **SET_MOTORS** - Ova komanda ima dva parametra koji predstavljaju procenat aktivacije levog i desnog motora, respektivno. Nakon čitanja ove dve vrednosti, Arduino poziva funkciju koja pokreće motore tako što aktivira PWM pinove intenzitetom prosleđenih vrednosti. Ukoliko oba prosleđena parametra imaju vrednost 0 motori će se zaustaviti, a ukoliko su vrednosti negativne motori će se kretati unazad.
- **LIGHTS_ON** - Ovom komandom Arduino aktivira pin koji pali svetleće diode na platformi.
- **LIGHTS_OFF** - Ovom komandom Arduino deaktivira pin prethodno aktiviran **LIGHTS_ON** komandom, čime se svetleće diode gase.
- **HEARTBEAT** - Ovom komandom se osvežava vreme najskorijeg prijema *heartbeat* signala, koje se proverava u sledećem pozivanju **loop()** funkcije.
- **END** - Ovom komandom se Arduino-u signalizira da je Raspberry Pi prestao sa radom. Primanjem END komande se gase motori i zaustavljaju se naredne aktivacije senzora udaljenosti.

Prenos slike

Podsistem za prenos slike je zasnovan na primeru iz dokumentacije **picamera** biblioteke, koja sadrži potreban kod za povezivanje sa Raspberry Pi kamerom [20]. Princip funkcionisanja prenosa slike je sledeći: pri učitavanju korisničke stranice šalje se zahtev za **stream.mjpg** elementom. Server ovaj zahtev obrađuje u posebnoj niti i nakon slanja zaglavla nit ostaje u beskonačnoj petlji gde se pri svakoj iteraciji petlje blokira na objektu klase **Condition** iz **threading** biblioteke. Ovaj objekat se nalazi unutar objekta klase **StreamingOutput**. Kamera je pri pokretanju snimanja konfigurisana da snimljeni sadžaj upisuje u objekat klase **StreamingOutput**, automatski pozivajući njegovu **write()** metodu. Kada kamera napravi novu sliku pozvaće se **write()** metoda ove klase, koja će upisati u svoj bafer sliku i signalizirati nitima koje čekaju na njenom **Condition** objektu da treba da ponovo provere uslov za blokiranje. Nakon ovoga, nit koja je čekala novu sliku se budi, uzima sliku iz **StreamingOutput** objekta i prosleđuje je korisniku. Ova nit nastavlja da radi tokom čitavog trajanja programa.

Na klijentskoj strani, kada korisnik pritisne dugme za početak snimanja poslaće se AJAX zahtev koji će **CommandExecutor** protumačiti i započeti sa snimanjem. Započinjanjem snimanja, kamera će dobijati nove slike i pozivati **write()** metodu objekta klase **StreamingOutput**, čime će serverska nit za opsluživanje slike moći da nastavi sa radom. Korisnikovim pritiskom na dugme za prekid snimanja, kamera će prestati sa snimanjem, čime će se serverska nit za opsluživanje slike blokirati.

Nezavisnost zasebnih celina koda

Tokom izrade projekta se težilo ka potpunom razdvajanju koda koji se izvršava na različitim platformama. U tom cilju je, na primer, odlučeno da vrednosti koje Arduino prima od Raspberry Pi-a predstavljaju procentualne, a ne konkretne vrednosti aktivacije motora, potrebne PWM pinovima. Raspberry Pi ne bi trebalo da zna kako je hardverski realizovana aktivacija motora, pa on samo šalje vrednosti u opsegu od -100%, za pokretanje levog ili desnog motora potpunom jačinom unazad, do 100%, za pokretanje motora potpunom jačinom unapred. Pri tome, pokretanje jednog motora unapred zahteva dovođenje napona na jedan pin L293D čipa, dok pokretanje istog tog motora unazad zahteva dovođenje napona na neki drugi pin. Takođe,

primenjivanje PWM tehnike aktivacije motora može varirati od jednog do drugog mikrokontrolera pa Raspberry Pi ne bi trebalo da zavisi ni od toga.

Slično, kod za pokretanje platforme na Raspberry Pi-u ne bi trebalo da zavisi od konkretnog načina kontrole platforme (tastaturom ili *Myo* narukvicom), pa je klijentski deo sistema realizovan tako da platformi šalje informacije samo o tome da li je potrebno kretati se napred, nazad, skretati levo ili skretati desno, a ne, na primer, koji je taster pritisnut na tastaturi ili koji položaj šake je prepoznaša narukvica. Na ovaj način se olakšava dodavanje novih načina kontrole robota (na primer mišem ili džojstikom).

4.4 Tehničke karakteristike

Sav kod je pisan u editorma vim [21] i Notepad++ [22], korišćen je PuTTY [23] alat za povezivanje sa Raspberry Pi-em preko SSH protokola, kao i WinSCP [24] za pristup fajlovima na Raspberry Pi-u. U nastavku odeljka su opisane, zasebno za svaku celinu, tehničke karakteristike sistema: korišćeni jezici, biblioteke i alati.

Arduino

Kod za Arduino je pisan u jeziku C++, uz poštovanje određenih pravila za strukturiranje koda, potrebnih za pisanje programa na ovoj platformi (npr. postojanje `setup()` i `loop()` funkcija). Pored standardnih Arduino biblioteka korišćena je biblioteka *NewPing* verzije 1.8 za aktivaciju senzora udaljenosti, uz povezivanje funkcije koja će se pozivati na prekid od internog tajmera i proveravati da li je stigao povratni ultrazvučni talas senzora udaljenosti [25]. Za prevodenje ovog koda je korišćen *Ino*, alat koji se pokreće sa komandne linije i automatizuje prevodenje koda pomoću *avr-g++* kompjajlera, a zatim pomoću *avrdude* alata učitava prevedeni kod na Arduino mikrokontroler [26].

Raspberry Pi

Sav kod koji se izvršava na Raspberry Pi računaru je pisan na programskom jeziku Python. Korišćene standardne Python biblioteke su: *io*, *os*, *sys*, *threading*, *configparser*, *string*, *logging*, *socketserver*, *http*. Pored ovih, korišćene su i dve nestandardne biblioteke: *serial* i *picamera*. *serial* biblioteka je potrebna za uspostavljanje serijske komunikacije između Raspberry Pi-a i Arduino-a [27], a *picamera* biblioteka je korišćena za povezivanje sa Raspberry Pi kamerom [20].

Web

Jezici korišćeni u ovom delu sistema su HTML i JavaScript, uz CSS za stilizovanje prezentacije. Od biblioteka su korišćeni *Bootstrap* [28] radni okvir, kao i *Bootstrap Toggle* [29] i *BootSideMenu* [30] plugin-ovi. Pored ovoga, korišćen je i JavaScript fajl *myo.js* koji služi za povezivanje *Myo* narukvice sa JavaScript-om [31]. Takođe, klijent mora na svom uređaju da poseduje instaliran Myo Connect program, kako bi povezivanje narukvice sa JavaScript-om funkcionalo.

5. Zaključak

U procesu kreiranja ovog rada razvijen je sistem za udaljeno upravljanje robotskom platformom putem lokalne mreže. Omogućen je rad platforme u uslovima promenljive vidljivosti i tokom noći, uz pomoć kamere postavljene na platformu koja prenosi sliku svoje okoline u realnom vremenu, ultrazvučnog senzora za merenje udaljenosti i svetlećih dioda. Uspostavljena je pouzdana komunikacija između komponenti sistema kroz nekoliko odvojenih tokova. Pouzdanost komunikacije je vrlo bitna karakteristika sistema, pogotovo onih koji u sebi uključuju hardverske komponente. Pored standardnog upravljanja putem tastature, eksperimentisalo se i sa upravljanjem pomoću *Myo* narukvice. Ovaj način upravljanja se pokazao kao intuitivan i jednostavan za korišćenje. Pokreti šakom su se vrlo precizno prevodili u kretanje platforme, što je predstavljalo glavni preduslov za dalje korišćenje ovakvog načina upravljanja.

Dalji razvoj ovakvog sistema može da se kreće u više pravaca.

Platforma se može unaprediti dodavanjem novih hardverskih komponenata. Kamera se može fiksirati unutar kućišta koje se pomoću dva servo motora može pomerati gore-dole i levo-desno. Time bi se omogućilo proširivanje korisnikovog vidnog polja, bez potrebe za pomeranjem same platforme. Takođe, na platformu se može dodati i efektor u vidu robotske ruke koji bi omogućio korisniku da interaguje sa okolinom platforme.

Tokom rada na implementaciji sistema, pretpostavljeno je da će samo jedan korisnik u jednom trenutku pristupati veb interfejsu za kontrolu platforme. Ovo ne mora važiti u opštem slučaju, pogotovo ukoliko platforma bude kontrolisana putem Interneta, budući da tada uređaj mora posedovati javno dostupnu IP adresu. Radi sprečavanja nekonistentnog ponašanja, u navedenom slučaju bi bilo neophodno uvođenje nekog principa kontrole pristupa korisnika. Korisnik bi mogao da izvrši autentikaciju pomoću svog korisničkog imena i lozinke. Ukoliko bi postojalo više korisnika koji bi želeli istovremeno da pristupaju platformi, samo bi jedan korisnik smeо da ima pristup kontrolama motora platforme, a ostali korisnici bi imali pristup samo slici sa kamere. U slučaju postojanja više korisnika koji žele pristup slici sa kamere, morao bi se osmislti efikasniji način slanja slike korisnicima. Budući da hardver Raspberry Pi-a nije dovoljno snažan za opsluživanje više korisnika video snimcima zadovoljavajućeg kvaliteta morao bi se koristiti još neki server kao posrednik. Na taj server bi Raspberry Pi slao video snimak, a zatim bi server distribuirao snimak većem broju korisnika.

Treći pravac razvoja ovakvog sistema može biti ospozobljavanje platforme za autonomno donošenje odluka dodavanjem sposobnosti za prepoznavanje slike sa kamere. Softver za obradu slike (poput *OpenCV* biblioteke) bi mogao da se upotrebi tako da na slici prepoznaće karakteristične oblike od interesa za konkretnog korisnika. Potom bi se ceo sistem mogao konfigurisati tako da izvršava određene akcije u zavisnosti od prepoznatih oblika - upozorava korisnika o potencijalno opasnom predmetu, kreće se ka ili od prepoznatog predmeta ili na neki način interaguje sa njim.

6. Literatura

- [1] Weber, Austin. „Nikola Tesla: Father of Unmanned Vehicle Technology.” Assembly Magazine. 04.26.2010.
<http://www.assemblymag.com/articles/87689-nikola-tesla-father-of-unmanned-vehicle-technology> (06.07.2017)
- [2] Engineering and Technology History Wiki. „Milestones:Early Developments in Remote-Control, 1901.” 15.03.2007.
http://ethw.org/Milestones:Early_Developments_in_Remote-Control,_1901 (08.07.2017)
- [3] Chandler, Nathan. „What is the history of the remote control?” HowStuffWorks.com. 03.03.2011.
<http://science.howstuffworks.com/innovation/repurposed-inventions/history-of-remote-control.htm> (06.07.2017)
- [4] Benson, Coleman. „How to Make a Robot – Lesson 2: Choosing a Robotic Platform.” RobotShop. 12.10.2010.
<http://www.robotshop.com/blog/en/how-to-make-a-robot-lesson-2-choosing-a-robotic-platform-2-3706> (06.07.2017)
- [5] Benson, Coleman. „How to Make a Robot – Lesson 6: Controlling Your Robot.” RobotShop. 17.01.2011.
<http://www.robotshop.com/blog/en/how-to-make-a-robot-lesson-6-controlling-your-robot-2-3688> (06.07.2017)
- [6] Colgan, Alex. „How Does the Leap Motion Controller Work?” 09.08.2014.
<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/> (08.07.2017)
- [7] Y. Pititeeraphab, P. Choikunnan, N. Thongpance, K. Kullathum, Ch. Pintavirooj. „Robot-arm control system using LEAP motion controller.” International Conference on Biomedical Engineering (BME-HUST). 5.10.2016.
<http://ieeexplore.ieee.org/document/7782091> (06.07.2017)
- [8] Carmody, Kim. „How Motion Detection Works in Xbox Kinect.” Wired. 03.11.2010.
<https://www.wired.com/2010/11/tonights-release-xbox-kinect-how-does-it-work/> (08.07.2017)
- [9] Eaton, Kit. „Kinect’s Greatest Hack: A Gesture-Control Robot.” Fastcompany. 23.11.2010.
<https://www.fastcompany.com/1704830/kinects-greatest-hack-gesture-control-robot> (06.07.2017)
- [10] Thalmic Labs. „Myo Tech Specs.”
<https://www.myo.com/techspecs> (08.07.2017)
- [11] Raspberry Pi Foundation. „Raspberry Pi 3 Model B Specifications.”
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (06.07.2017)
- [12] Arduino. „Arduino Uno Rev3 Overview.”
<https://store.arduino.cc/arduino-uno-rev3> (06.07.2017)

- [13] Raspbian. „Raspbian Operating System.”
<https://www.raspbian.org/> (08.07.2017)
- [14] ElecFreaks. „Ultrasonic Ranging Module HC - SR04.”
<http://www.micropik.com/PDF/HCSR04.pdf> (06.07.2017)
- [15] Texas Instruments. „L293x Quadruple Half-H Drivers.” 01.2016.
<http://www.ti.com/lit/ds/symlink/l293.pdf> (06.07.2017)
- [16] ModMyPi. „Hobbywing 5V 3A UBEC Step-Down Converter Description.”
<https://www.modmypi.com/electronics/power-and-batteries/step-updown-modules/hobbywing-5v-3a-ubec-step-down-converter> (06.07.2017)
- [17] Raspberry Pi Foundation. „Camera Module v2 Specifications.”
<https://www.raspberrypi.org/products/camera-module-v2/> (06.07.2017)
- [18] Thalmic Labs. „How does the Myo armband work?”
<https://support.getmyo.com/hc/en-us/articles/202532376-How-does-the-Myo-armband-work-> (06.07.2017)
- [19] Fairchild Semiconductor. „FQP30N06L Datasheet.” 11.2013.
<http://www.onsemi.com/pub/Collateral/FQP30N06L.pdf.pdf> (08.07.2017)
- [20] Jones, Dave. „picamera Documentation.”
<https://picamera.readthedocs.io/en/latest/> (06.07.2017)
- [21] Moolenaar, Bram. „Vim Editor.”
<http://www.vim.org/> (08.07.2017)
- [22] Ho, Don. „Notepad++ Editor.”
<https://notepad-plus-plus.org/> (06.07.2017)
- [23] Tatham, Simon. „PuTTY.”
<http://www.putty.org/> (06.07.2017)
- [24] Prikryl, Martin. „WinSCP.”
<https://winscp.net/eng/index.php> (08.07.2017)
- [25] Eckel, Tim. „NewPing Library for Arduino.” 24.02.2017.
<http://playground.arduino.cc/Code/NewPing> (06.07.2017)
- [26] Amperka. „Ino Documentation.” 2011.
<http://inotool.org/> (06.07.2017)
- [27] Liechti, Chris. „pySerial Documentation.”
<https://pyserial.readthedocs.io/en/latest/> (06.07.2017)
- [28] Bootstrap. „Bootstrap Framework.”
<https://getbootstrap.com/> (06.07.2017)
- [29] Hur, Min. „Bootstrap Toggle.”
<http://www.bootstraptoggle.com/> (06.07.2017)
- [30] Lombardo, Andrea. „BootSideMenu.”
<http://www.lombardoandrea.com/demo/BootSideMenu/index.html> (06.07.2017)
- [31] Thalmic Labs. „myo.js Github Repository.”
<https://github.com/thalmiclabs/myo.js> (08.07.2017)