

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Mobiliojo roboto valdymo programėlės kūrimas

Multimedijos technologijos mobiliems įrenginiams (T120M159)

Kursinis darbas

Atliko: IFM-2/1 stud. Tomas Uktveris

Dėstytojas: *doc. dr. Tomas Blažauskas*

Kaunas, 2013

Turinys

1. Projektas	3
1.2. Tikslinė auditorija	3
1.3. Projekto licencija.....	3
1.4. Projekto kodas.....	4
1.5. Situacija rinkoje	4
1.6. Atlikti darbai	4
1.7. Tolesni plėtojimo darbai	4
2. Techninė įranga	5
2.1. Sistemos komponentai	5
2.2. Testavimo įrenginiai	6
2.3. Komunikacija.....	6
3. Programinės įrangos architektūra	8
3.1. Reikalavimai Android programėlei	8
3.2. Panaudos atvejai.....	8
3.3. Klasių diagrama	9
3.4. Realizavimo ypatumai	10
4. Vartotojo vadovas	11
4.1. Trumpas aprašymas	11
4.2. Programėlės naudojimas	11
4.2.1. Prisijungimas/atsijungimas	11
4.2.2. Sistemos išjungimas	12
4.2.3. Statistikos peržiūra	12
4.2.4. Vairavimo režimas (fiksotas)	12
4.2.5. Vairavimo režimas (giroskopo).....	13
4.3. Naudojami įrenginiai	14

1. Projektas

1.1. Paskirtis

Šio darbo tikslas yra sukurti mobilaus roboto valdymo programėlę Android OS, kuri leistų per atstumą, panaudojant WiFi ryšio ir kitas mobiliojo telefono galimybes, keisti roboto važiavimo kryptį, greitį bei kitus parametrus, stebėti valdymo sistemos statistiką.

Darbas susideda iš projekto analizės, programinės įrangos projektavimo ir jos realizacijos. Dokumente aprašoma projektavimo eiga bei realizacijos detalės, taip pat pateikiamas vartotojo vadovas, skirtas supažindinti vartotoją su sukurtos programinės įrangos valdymu.

1.2. Tikslinė auditorija

Programėlė skirta asmeniniam projektui ir buvo realizuojama siekiant pagilinti Android OS programų kūrimo žinias. Šis dokumentas skirtas visiems, kurie nori susipažinti su šios programėlės kūrimo ypatybėmis bei galbūt pasisemti žinių ar idėjų realizuojant savo projektą.

1.3. Projekto licencija

Darbo dokumentacijai ir kodui suteikiama MIT licencija: <http://opensource.org/licenses/MIT>

Copyright (c) 2013 Tomas Uktveris

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.4. Projekto kodas

Projekto kodas patalpintas GitHub DVCS: <https://github.com/tomazas/robotUI>

1.5. Situacija rinkoje

Panašių projektų-programėlių apžvalga yra vienas iš pirminių etapų prieš pradedant projektavimo ir programavimo darbus. Išanalizavus Google Play sistemoje pateikiamas panašias alternatyvas surasti tokie egzistuojantys sprendimai:

1. Lego Minstorms NXT robotų kontrolės programėlės naudojančios Bluetooth ryšį (NXT Remote Control, NXT Robotic Arm, Arduino Controlled Robot)
2. NAO roboto kontrolės programėlė (NAO Robot Controller)
3. AIBO roboto valdymo programėlė (Aibo Control)
4. Kitų specializuotų robotų valdymas (Bluetooth Control, BT Bot Control Pro)
5. Video stream funkciją turintys sprendimai (Robot RC)
6. 9 mygtukų universalus perprogramuojamas robotų valdiklis (Robot Remote Control)

Tinkamiausias ir universaliausias iš visų surastų projektų yra paskutinis variantas – Robot Remote Control. Valdymas programėlėje realizuojamas per UDP serverį, naudoja paprastą tekstinį protokolą, lengvai perprogramuojami mygtukai, tačiau trūksta giroskopo valdymo galimybių.

Kadangi projekto techninė įranga yra unikali (t.y. analogų neturi), naudojami nestandartiniai komunikavimo su technine įranga protokolai bei reikalinga naudoti giroskopą roboto vairavimui, tai pritaikyti esamus sprendimus konkrečiam atvejui prireiktų daugiau laiko nei sukurti naują sprendimą, todėl buvo nutarta realizuoti nuosavą programėlės variantą darbui su turima technine įranga.

1.6. Atlikti darbai

Šio kursinio darbo kontekste buvo atlikti tokie darbai:

1. Suprojektuota roboto valdymo programinė įranga
2. PĮ realizuota Android OS ir išbandyta su Samsung Galaxy SII I9005 telefonu
3. Dokumentuotas projektas

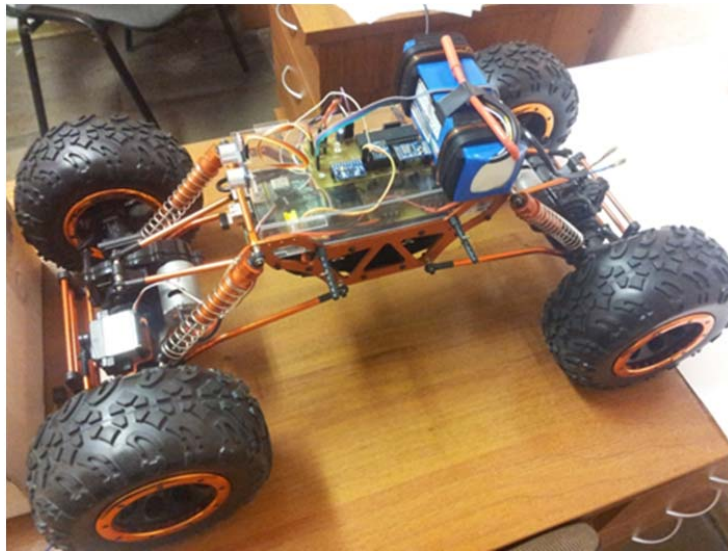
1.7. Tolesni plėtojimo darbai

Programėlė bus naudojama tolesniems roboto testavimo tikslams, pagal poreikį atitinkamai keičiama. Norint pritaikyti sprendimą plačiai auditorijai ar kitai techninei įrangai, reikia suvienodinti (unifikuoti) valdymo sąsają arba naudoti kito komunikacijos protokolo bibliotekas.

2. Techninė įranga

Šiame skyriuje trumpai aprašoma naudota techninė įranga programėlės kūrimui ir testavimui. Apžvelgiami komponentai, jų tarpusavio sąsajos ir bendras sistemos vaizdas.

2.1. Sistemos komponentai



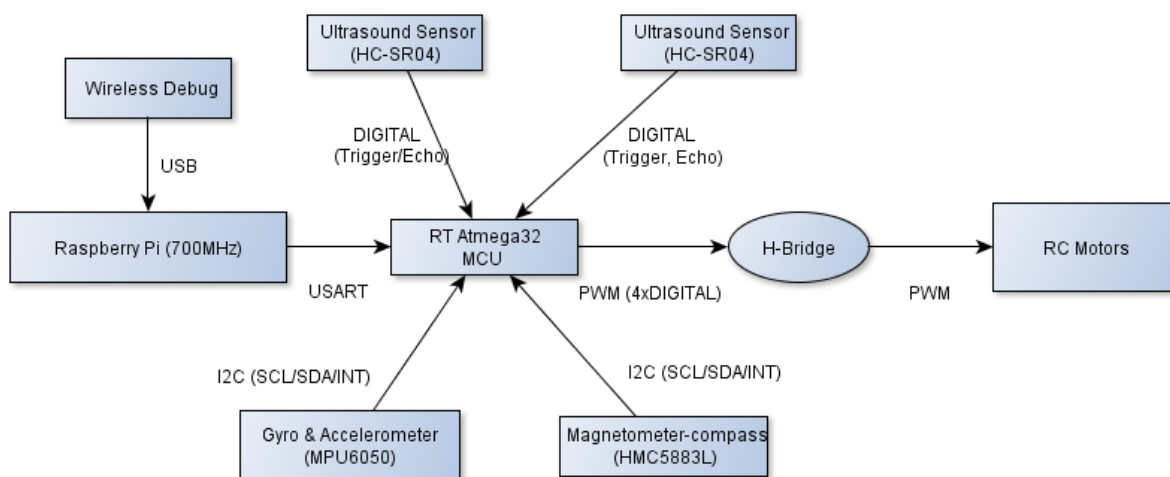
1 pav. Mobilaus roboto vaizdas

Roboto techninės įrangos sistema (1 pav.) susideda iš įvairių komponentų. Roboto važiuoklė/bazė pasirinkta HSP Rock Crawler 1:8 gamyklinė šasi (645x460x225 mm), susidedanti iš aliuminio karkaso, 4 alyva pripildytų amortizatorių, priekinio bei galinio RC motorų bei vairavimui skirtą 1 servo motorą. Energijos tiekimo sistemą sudaro 2000mAh 7.2V NiHM akumuliatorius RC motorams bei 5000mAh 22.2V LiPo akumuliatorius energijai į jutiklius ir procesorių tiekti. Ant roboto viršutinės dalies sumontuoti tokie jutikliai - akselerometras, giroskopas, magnetometras ir 2 ultragarso davikliai.



2 pav. Raspberry Pi ir kitos panaudotos technologijos

Skaiciavimus ir valdymo veiksmus atlieka du CPU – ATmel ATmega32 (48 išvadų) mikrovaldiklis bei Raspberry Pi 700 MHz mikrokompiuteris (2 pav.). Mikrovaldiklis naudoja modifikuotą Arduino branduolį ir bibliotekos funkcijas. Visi jutikliai sujungti su mikrovaldikliu (3 pav.), kuris realiu laiku iš jų surenka visą reikalingą informaciją. Mikrokontrolerio valdymo programa užima 23kB Flash atminties iš 32kB galimų. Motorų valdymas atliekamas per specialiai tam sukonstruotą H-tiltelio plokštę, kuri leidžia keisti kiekvieno motoro sukimosi greitį atskirai.



3 pav. Roboto sistemos schema

2.2. Testavimo įrenginiai

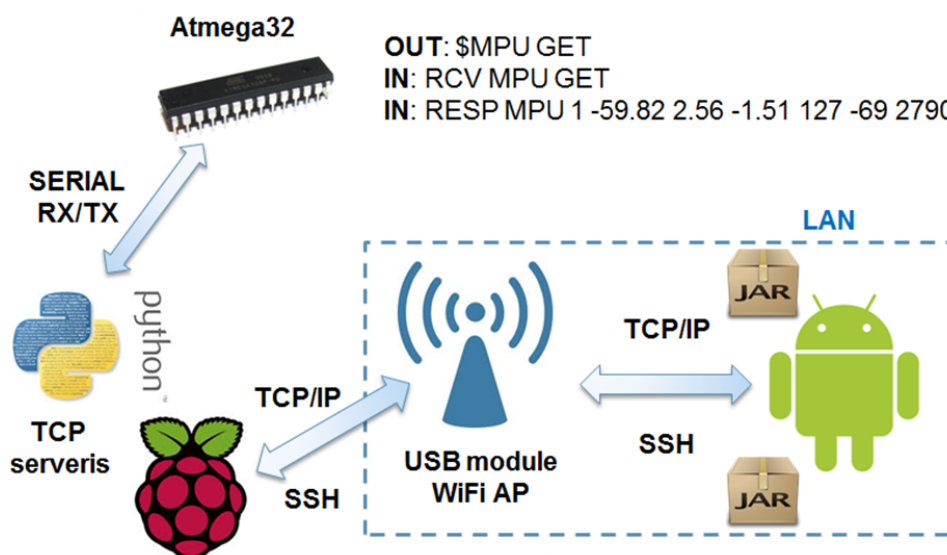
Android programėlė buvo testuota su Samsung Galaxy SII telefonu ir kurta Windows OS Eclipse IDE aplinkoje.

2.3. Komunikacija

Realizacijos testavimui naudojamas WiFi ryšys pasitelkiant telefono galimybes bei Raspberry Pi USB bevielio ryšio (angl. wireless) modulį. Robotas schemeje (4 pav.) yra WiFi

prisijungimo stotelė, o telefonas – klientas. Prisijungęs telefonas prie stotelės tampa roboto tinklo dalimi ir gali bendrauti su robotu kaip su įprastu tinklo įrenginiu per TCP/IP ar kitą transportą.

Raspberry Pi sistemoje veikia Python TCP/IP serveris besiklausantis iš nustatyto prievado (angl. port) valdymo komandų ir persiunčiantis jas ATmega32 mikrovaldikliui per nuosekliają jungtį RX/TX. Gauti rezultatai iš mikrovaldiklio persiunčiami atvirkščia seka.



4 pav. Komunikavimo schema

Taip pat yra galimybė prisijungti prie Raspberry Pi sistemos per SSH protokolą tam, kad būtų galima iškviesti Linux OS funkcijas ir tvarkingai išjungti visą sistemą. Tam telefone naudojama atskira Java *sshj* biblioteka.

Telefonas siunčia vienos ar kelių eilučių tekstinio tipo komandas (GET arba SET) mikrovaldikliui (kaip parodyta diagramoje) nurodydamas iš kurio jutiklio nori gauti informaciją. Atsakymas grąžinamas taip pat tekstinėje formoje.

3. Programinės įrangos architektūra

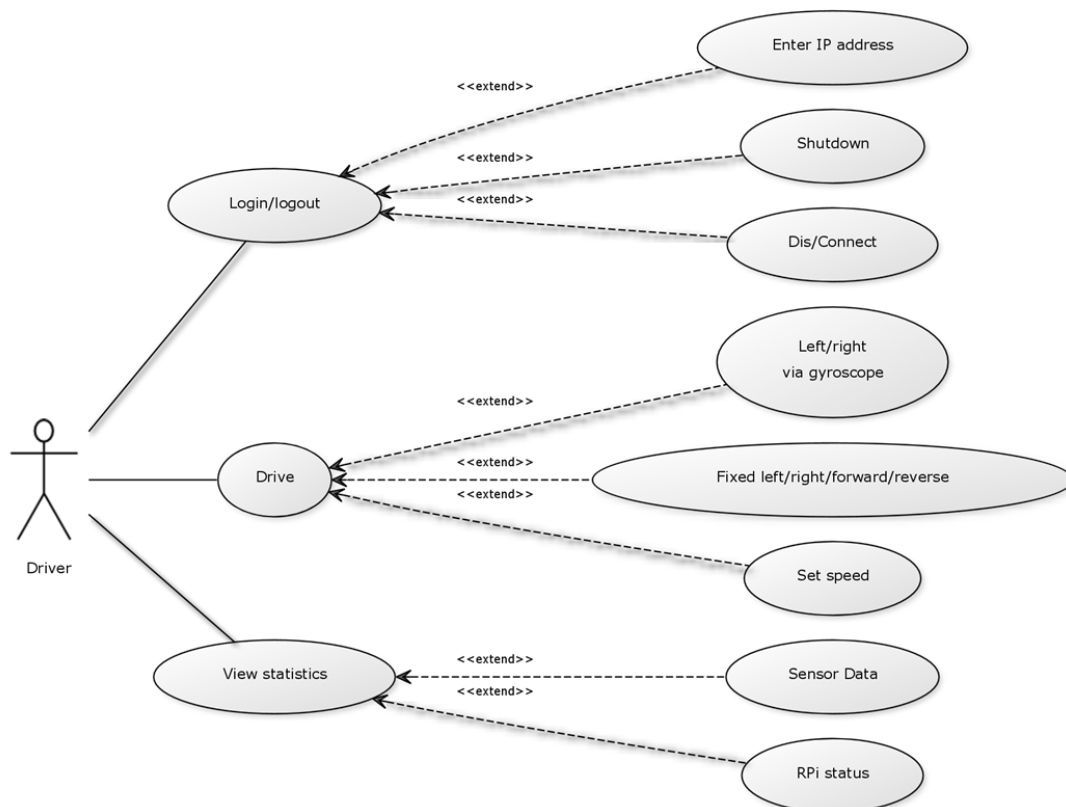
3.1. Reikalavimai Android programėlei

Funkciniai reikalavimai kuriamai programėlei išvardinti toliau:

1. Galimybė prisijungti/atsijungti, išjungti robotą
2. Galimybė įjungti/išjungti jutiklius
3. Rinkti statistiką:
 - a. jutiklių reikšmės
 - b. bendra RaspberryPi sistemos būklė (SD, RAM)
 - c. atsako laikas – RTT (round trip time), ATmega veikimo sparta
4. Vairavimas:
 - a. Fiksuotas valdymas (left/right/forward/reverse mygtukai)
 - b. Naudojant giroskopą

3.2. Panaudos atvejai

Anksčiau iškelti funkciniai reikalavimai atsispindi programėlės panaudos atvejų diagramoje (5 pav.) žemiau. Diagramos generuotos su <http://yuml.me> įrankiu.



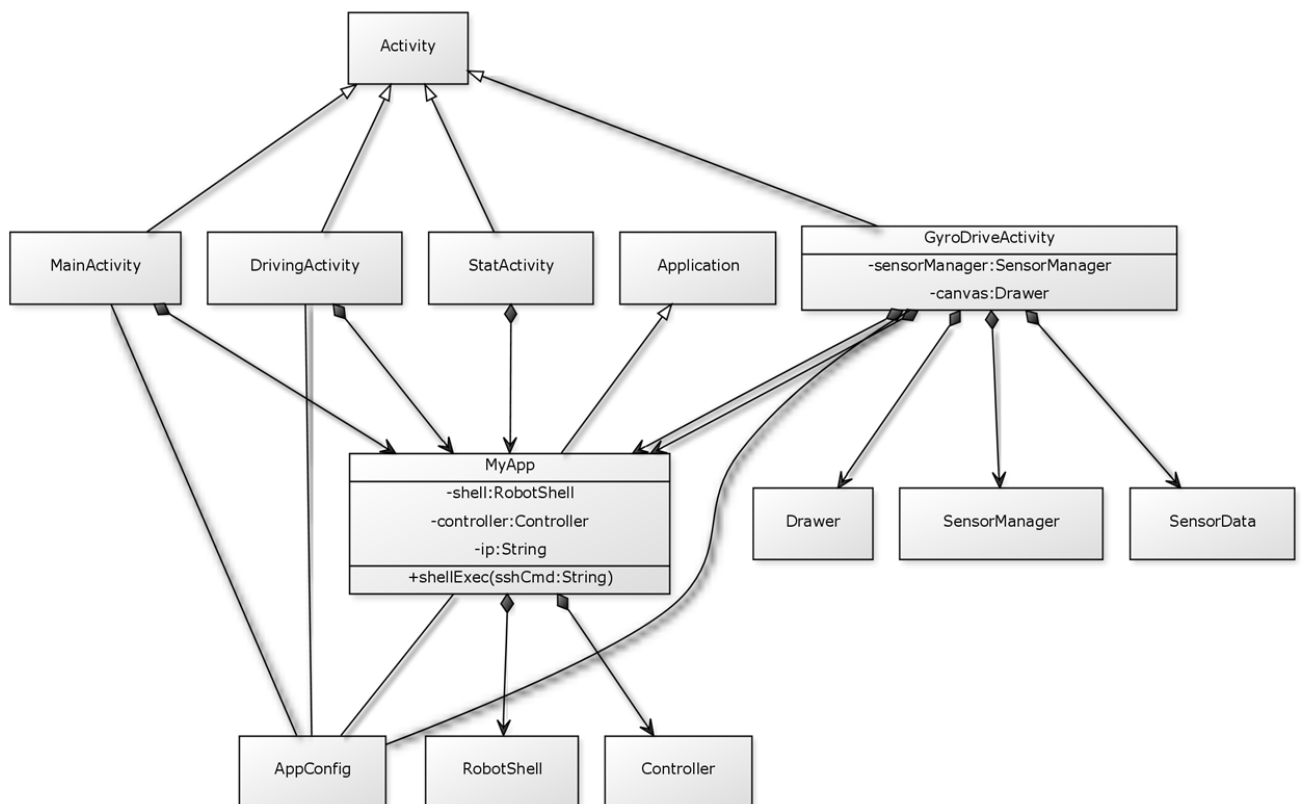
5 pav. Panaudos atvejų diagrama

3.3. Klasių diagrama

Programėlė sudaryta iš 4 pagrindinių Android Activity klasių (6 pav.):

1. MainActivity - programėlės pradinio lango vaizdavimo klasė
2. DrivingActivity - fiksuoto vairavimo lango vaizdavimo klasė
3. StatActivity - statistikos lango vaizdavimo klasė
4. GyroDriveActivity - giroskopo valdymo lango vaizdavimo klasė

Programėlėje naudojama statinė konfigūracija saugoma AppConfig klasėje. Komunikacija su robotu vyksta per RobotShell bei Controller klases. Pirmoji nustato komunikacijai reikalingą transportą, antroji – atsakinga už komunikacijos protokolą. Per visą programėlės gyvavimo laikotarpį egzistuojantys kintamieji saugomi MyApp klasėje.



6 pav. Klasių diagrama

GyroDriveActivity klasėje realizuotas giroskopo duomenų filtravimas naudoja SensorData klasę ir Drawer klasę – duomenų pateikimui bei piešimui ekrane.

3.4. Realizavimo ypatumai

Realizuojant programėlę buvo susidurta su problemomis išvardintomis žemiau:

- Socket'ų negalima naudoti iš GUI gijos – turi būti kuriama atskira gija tinklo komunikacijoms.
- Dėl TCP protokolo buferizavimo ir retransmisijos sukiamas roboto komandų uždelsimas, alternatyva – UDP protokolas ar TCP Nagle algoritmo išjungimas.
- Giroskopo nestabilumas – giroskopo reikšmės bėgant laikui išsiderina nuo pradinės reikšmės (angl. zero drift), todėl sudėtingiau nustatyti tikslią posūkio reikšmę atlikus posūkį ir grįžus vėl atgal. Papildomai reikia naudoti kompensacijos algoritmus ir akselerometro informaciją, kas nerealizuota programėlėje.

4. Vartotojo vadovas

4.1. Trumpas aprašymas

Android programėlė skirta mobilaus roboto valdymui per bevielį WiFi ryšį. Programėlė leidžia per atstumą keisti roboto važiavimo kryptį, greitį bei kitus parametrus, stebėti kitą valdymo sistemos statistiką. Programėlės naudojimui reikalinga techninė įranga, kurioje veiktų TCP serveris priimančias ir persiunčiantis tekstines komandas ATmega ar kitokiems mikrovaldikliams.

4.2. Programėlės naudojimas

Šiame skyriuje trumpai apžvelgiami darbo su programėle etapai.

4.2.1. Prisijungimas/atsijungimas

Atsidarius programėlę pirmame lange (7 pav.) matome bendrą prisijungimo statusą. Norint prisijungti prie roboto techninės įrangos pirmame lange įvedame IP adresą ir prievado numerį bei spaudžiame „Connect“ mygtuką.



7 pav. Pradinis programėlės vaizdas

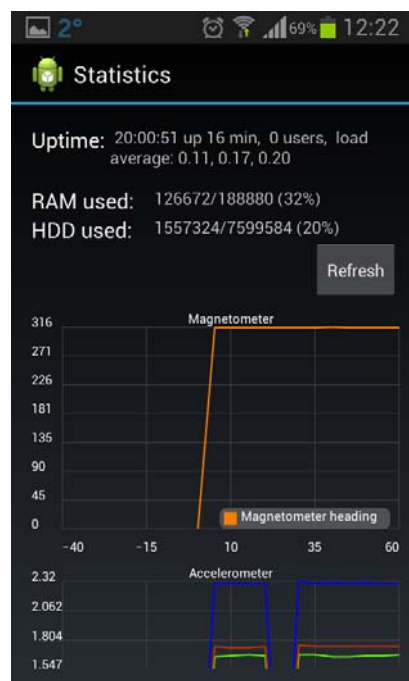
Sėkmingai prisijungus atsiranda pranešimas “Connected” bei matomi sistemos veikimo bei atsako laikai. Pažymimi varnelėmis kokie jutikliai yra šiuo metu įjungti. Norint išjungti kurį nors iš jutiklių, atitinkamai nuimama varnelė prie jo. Norint atsijungti spaudžiamas “Disconnect” mygtukas.

4.2.2. Sistemos išjungimas

Norint išjungti roboto sistemą, pradiname lange pažymime varnelę “Shutdown” ir spaudžiame mygtuką “Execute”, iššokus pranešimui patvirtiname dar kartą.

4.2.3. Statistikos peržiūra

Pereiti į statistikos peržiūros langą (8 pav.) galima iš pradinio lango paspaudus apačioje esantį “Stats” mygtuką. Naujame vaizde matomas Raspberry Pi veikimo laikas, kiek operatyviosios atminties RAM bei kortelės vietos HDD užimta. Norint atnaujinti informaciją, reikia paspausti “Refresh” mygtuką.

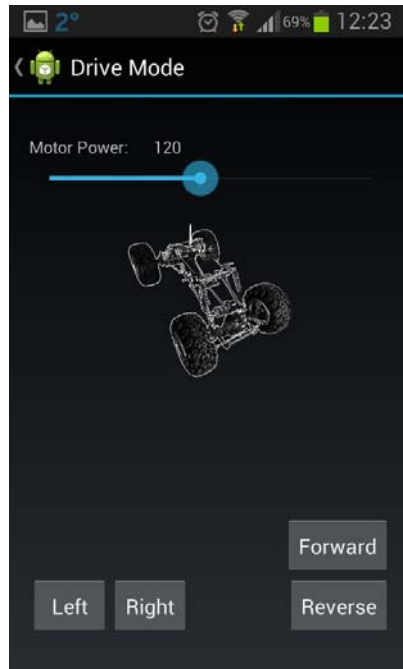


8 pav. Statistikos vaizdas

Grafikuose rodomos reikšmės atsinaujina automatiškai laikui einant. Visų jutiklių parodymai vaizduojami grafikuose.

4.2.4. Vairavimo režimas (fiksotas)

Pereiti į vairavimo režimą (9 pav.) galima iš pradinio lango paspaudus apačioje esantį “Drive Mode” mygtuką. Naujame vaizde matomi keturi mygtukai, kuriais valdomas robotas bei slinkties juosta – skirta nustatyti, kiek galios bus perduodama į motorus (0 - 0%, 255 - 100%).

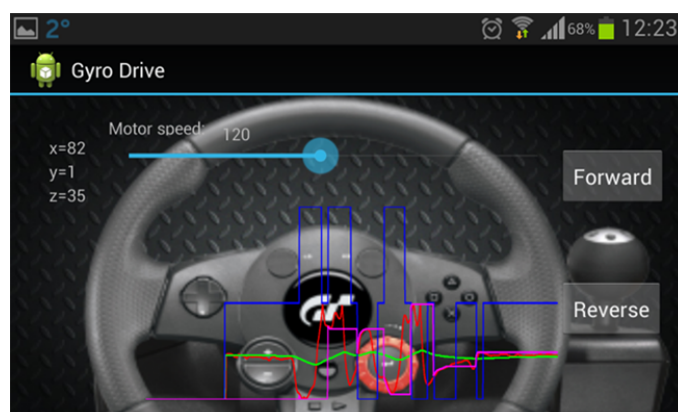


9 pav. Vairavimo režimo vaizdas

Robotas važiuos pirmyn (“Forward”), atgal (“Reverse”), į kairę (“Left”) arba į dešinę (“Right”), jei paspausti atitinkami mygtukai.

4.2.5. Vairavimo režimas (giroskopo)

Pereiti į giroskopo vairavimo režimą (10 pav.) galima iš pradinio lango paspaudus apačioje esantį “Gyro” mygtuką. Naujame vaizde matomi du mygtukai, iš kurių paspaudus vieną robotas važiuoja pirmyn (“Forward”) arba atgal (“Reverse”). Slinkties juosta skirta nustatyti, kiek galios bus perduodama į motorus (0 - 0%, 255 - 100%).



10 pav. Vairavimo giroskopu režimo vaizdas

Realiu laiku gaunami giroskopo duomenys braižomi grafike. Raudona kreivė - nefiltruoti giroskopo duomenys, žalia kreivė - filtruoti duomenys (eksponentinis slenkantis vidurkis),

mėlyna kreivė turi tris būsenas: būseną nesikeičia - centras, sukama į dešinę - aukštas/viršuje, sukama į kairę - apačioje. Violetine spalva žymima naujausia giroskopo X ašies piko reikšmė. Giroskopo pasisukimo ašių reikšmės laipsniais rodomos kairiajame viršutiniame kampe.

4.3. Naudojami įrenginiai

Programėlės realizacijoje naudojamas mobilus įrenginio WiFi ryšys ir giroskopo funkcijos.