

SULAUTETUT järjestelmät PLa-32310

RAKENNUSPROJEKTI

ARDUINO JA RASPBERRY PI 3 OHJATTU ROVER PROTOTYYPPI

|  |  |
| --- | --- |
| Tarkastaja: Mika Saari |  |
|  |  |

SISÄLLYSLUETTELO

[1. Johdanto 1](#_Toc466730837)

[2. ROVERIN ESITTELY 2](#_Toc466730838)

[3. Arduino ohjelma 6](#_Toc466730839)

[3.1 Käytetyt kirjastot 6](#_Toc466730840)

[3.2 Omat metodit 7](#_Toc466730841)

[3.3 Ohjelman rakenne 8](#_Toc466730842)

[4. Raspberry pi ohjelmat 9](#_Toc466730843)

[4.1 Asennetut ohjelmat 9](#_Toc466730844)

[4.2 Skriptit 9](#_Toc466730845)

[5. Yhteydet raspberryyn 10](#_Toc466730846)

[6. Yhteenveto 11](#_Toc466730847)

[Lähteet 12](#_Toc466730848)

LIITTEET: KAIKKI KOODIT SAATAVANA OSOITTEESSA:

<https://github.com/Stenu/robot>

# Johdanto

Tämä harjoitustyö on tehty Tampereen Teknillisen Yliopiston PLA-32310 Sulautetut järjestelmät kurssin syksyn 2016 toteutuksen harjoitustyönä. Harjoitustyössä pyrittiin hyödyntämään kurssilla opetettuja asioita.

Kaikki harjoitustyön koodit on tallennettu GitHub verkkosivustolle, jossa ne ovat julkisesti kaikkien saatavilla. Harjoitustyön GitHub osoite on <https://github.com/Stenu/robot>.

Harjoitustyönä rakennettiin webkameralla varustettu ohjelmoitava pienoismönkijä Rover. Webkamera on asennettu kiikkupäähän, jota voidaan kallistaa ja kääntää. Rover havaitsee esteet liikkuessaan eteenpäin.

Roverissa Arduino on yhdistetty Raspberry Pi 3:een sarjayhteydellä. Raspberryn kautta Roveria voi hallita Android älypuhelimella tai kannettavalla tietokoneella.

# ROVERIN ESITTELY



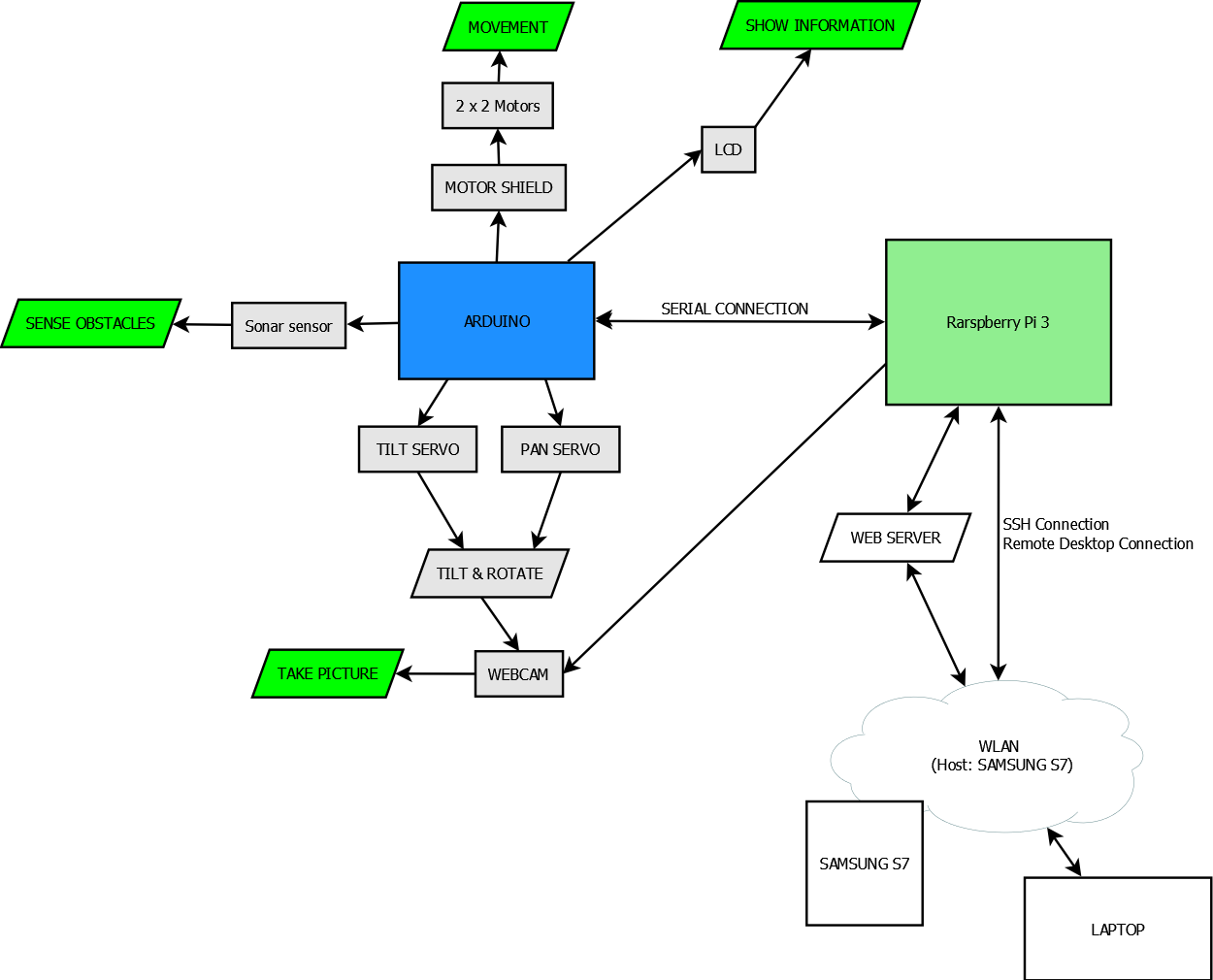


1. Harjoitusprojektina rakennettu Rover ”mönkijä”

Harjoitusprojektina rakennettiin Rover mönkijä. Rover liikkuu neljän DC -moottorin voimin, havaitsee esteet ultraäänianturilla ja voi ottaa kuvia kierto- ja taittopäähän asennetulla webkameralla. Lisäksi Roveriin on asennettu LCD-näyttö, josta voidaan seurata ultraäänianturin mittaustulosta ja ohjelman tilaa.

Lista roverissa käytetyistä osista:

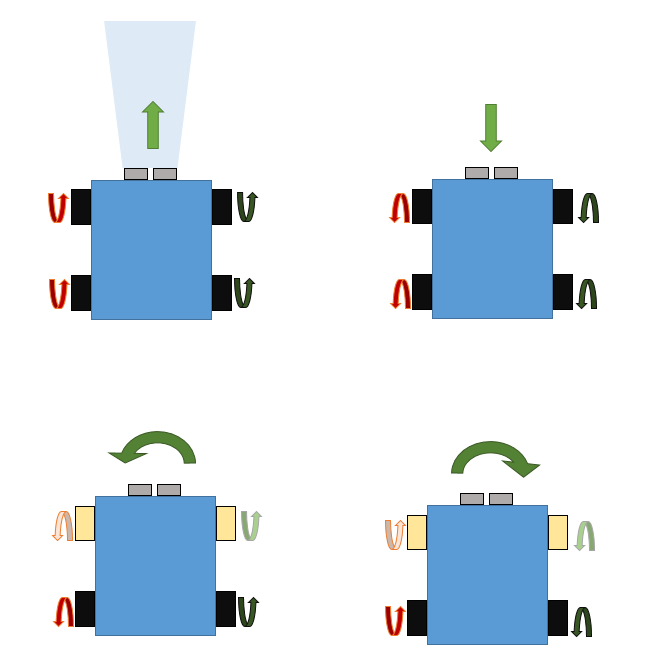
* Raspberry pi 3
* Arduino Mega 2560
* Anduino LCD shield
* Arduino Motor shield
* HC-SR04 Ultrasonic sensor
* Lynxmotion Pan and Tilt Kit
* 7800 mAh usb virtalähde
* DFRobot 4WD Arduino mobile platform
  + 4 DC moottoria
  + Runko ja renkaat
  + paristopidin



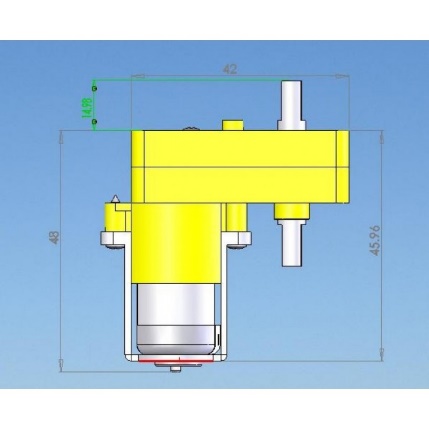
1. Roverin toimintakaavio

Roverin toiminnot suoritetaan Arduinon ja Raspberry Pin yhteistyönä. Arduinoon on kytketty DC-moottorit, servot, ultraäänianturi ja LCD-näyttö. Laitteista vain webbikamera on kytketty Raspberryyn. Raspberry antaa Arduinolle käskyjä sarjayhteydellä, joiden perusteella Arduinossa ohjataan siihen kytkettyjä laitteita.

Raspberry on WLAN yhteydessä Android matkapuhelimeen ja Windows 10 kannettavaan tietokoneeseen, joista käyttäjä pääsee ohjaamaan Roveria.

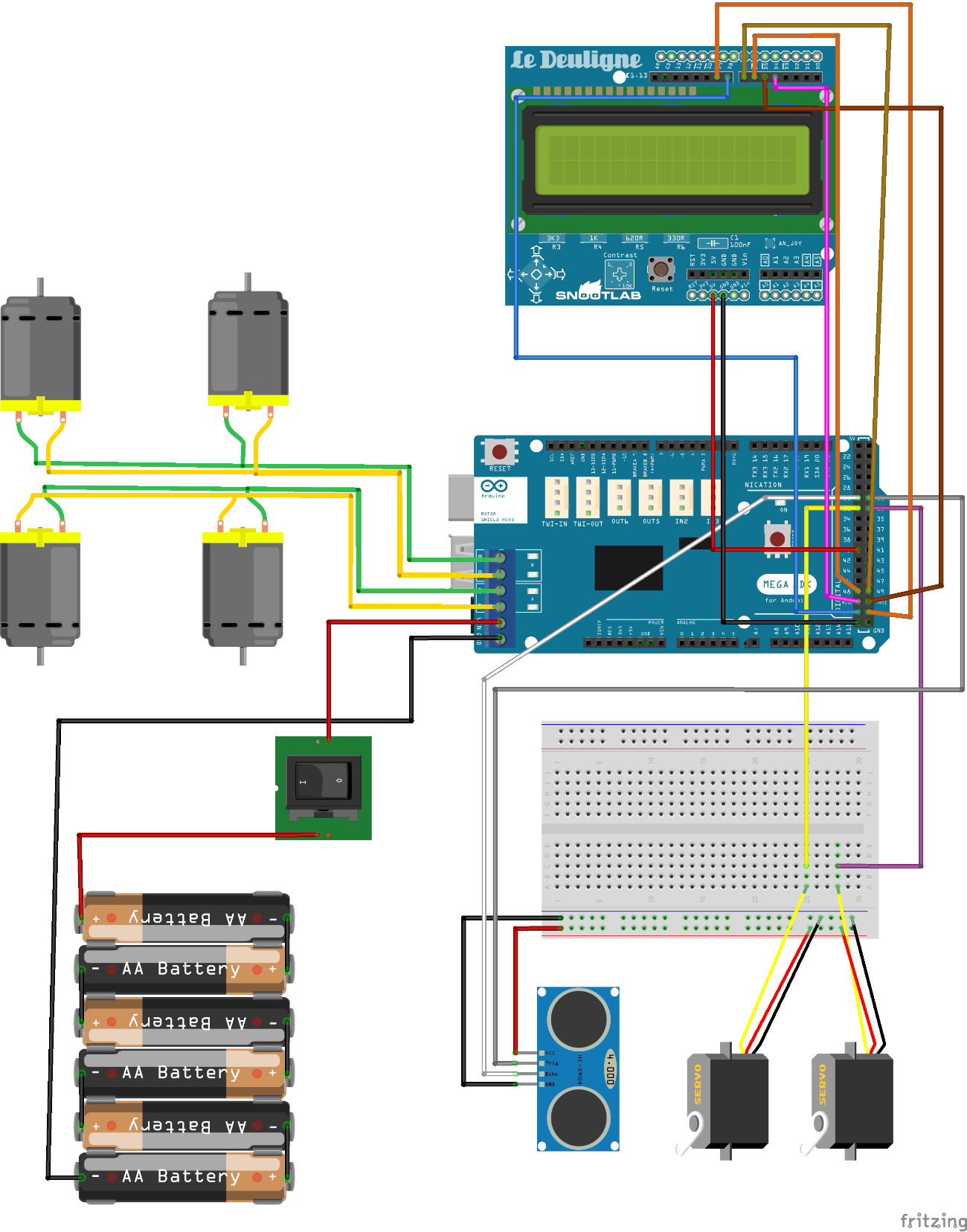


1. Roverin liikkeiden toteuttaminen



1. Roverin DC-moottori ja vaihteisto

Rover liikkuu neljän DC-moottorin voimin. Moottorit on kytketty vasempaan ja oikeaan kanavaan. Liikkuminen eteenpäin ja taaksepäin toteutetaan pyörittämällä renkaita samaan suuntaan. Kääntyminen on toteutettu pyörittämällä renkaita erisuuntiin. Koska renkaat eivät käänny, kitkavoimien johdosta ei kääntyminen onnistu yllä kuvatulla tavalla. Ongelma on kierretty vähentämällä eturenkaiden pitoa poistamalla renkaista kumit. Nyt Rover kääntyy taka-akselin keskipisteen ympäri, eturenkaiden liukuessa pintaa pitkin.



1. Arduinon kytkennät. Piirretty Frizingillä [1]

# Arduino ohjelma

## Käytetyt kirjastot

Arduino-ohjelma laadittiin Arduinon omalla sovelluskehittimellä [2], joka sisältää Arduino-ohjelmissa tarvittavat tavanomaiset kirjastot. Esimerkiksi ohjelmassa sarjayhteys Raspberry Pi 3:seen muodostettiin Serial -kirjastolla, joka sisältyy Arduinon sovelluskehittimen valmiisiin kirjastoihin.

Arduinon sovelluskehittimen kirjastojen lisäksi käytettiin LiquidCrystal [3], NewPing [4] ja Servo [5] -kirjastoja.

LiquidCrystal -kirjastolla luotiin olio lcd, joka alustettiin LCD-näyttöön kytkettyjen pinnien numeroilla. Olion metodi clear() tyhjentää näytön, setCursor(paikka,rivi) asettaa kursorin haluttuun paikkaan ja print(”viesti”) kirjoittaa viestin kursorin alkupaikasta lähtien.

NewPing -kirjastolla luotiin olio sonar, joka alustettiin HC-SR04 ultraäänisensoriin kytketyillä trigger pinnillä, echo pinnillä ja maksimi etäisyydellä. NewPing olio sonar sisältää metodin ping\_cm(), joka antaa palauttaa sensorin havaitseman esteen etäisyyden senttimetreissä.

Metodi toimii lähettämällä signaalin ultraäänisensorin trigger -pinniin, jolloin ultraäänisensori lähettää ultraääniä. Ultraäänianturi antaa signaalin echo -pinniin äänien kaiun palautuessa sensoriin. Olkoon *t* aika, joka kuluu ultraäänien lähettämisestä niiden kaiun palautumiseen sensoriin. Äänen nopeus on ilmassa 34 400 cm/µs [6]. Esteen etäisyys *s* anturista saadaan seuraavasti:

Servo -kirjastolla luotiin oliot panservo ja tiltservo. Oliot kytkettiin signaali -pinneihin attach(pin) -metodilla. Servo olioiden write(position) -metodilla servo asemoidaan haluttuun asemaan.

## Omat metodit

Ohjelman selkeyttämiseksi ja koodin kopioimisen välttämiseksi ohjelmaan kirjoitettiin omia metodeja.

listenCommand -metodi tutkii sarjayhteyden puskuria. Jokaisella pääsilmukan kierroksella katsotaan puskuria. Jos puskurissa on saatavilla merkki, niin merkki luetaan. [ -merkki aloittaa uuden käskyn. ] -merkki lopettaa uuden käskyn ja kutsuu evaluateCommand -metodia.

evaluateCommand – metodi tarkistaa uuden käskyn muodon. Jos käsky on annettu oikeassa muodossa, suoritetaan käskyä vastaava metodi.

tellState -metodi palauttaa state -muuttujan arvon sarjayhteydellä.

writeLcd -metodi tulostaa kaksi kertaa sekunnissa LCD-näytölle viestin ja etäisyysanturin lukeman.

readDistance -metodi lukee etäisyyden ultraäänianturilta ja sijoittaa mittaustuloksen distance -muuttujaan.

panAndTilt -metodi asemoi kierto- ja kulmapään haluttuun asentoon.

turnOnLcd -metodi kytkee LCD-näytön päälle.

turnOffLcd -metodi sammuttaa LCD-näytön.

moveForward -metodi sammuttaa LCD-näytön ja asettaa Arduinon Motor shieldin moodiin jossa rover liikkuu eteenpäin.

moveBackward -metodi sammuttaa LCD-näytön ja asettaa Arduinon Motor shieldin moodiin jossa rover liikkuu taakseppäin.

turnRight ja turnLeft -metodit sammuttavat LCD-näytön ja asettavat Arduinon Motor shieldin moodiin jossa rover kääntyy.

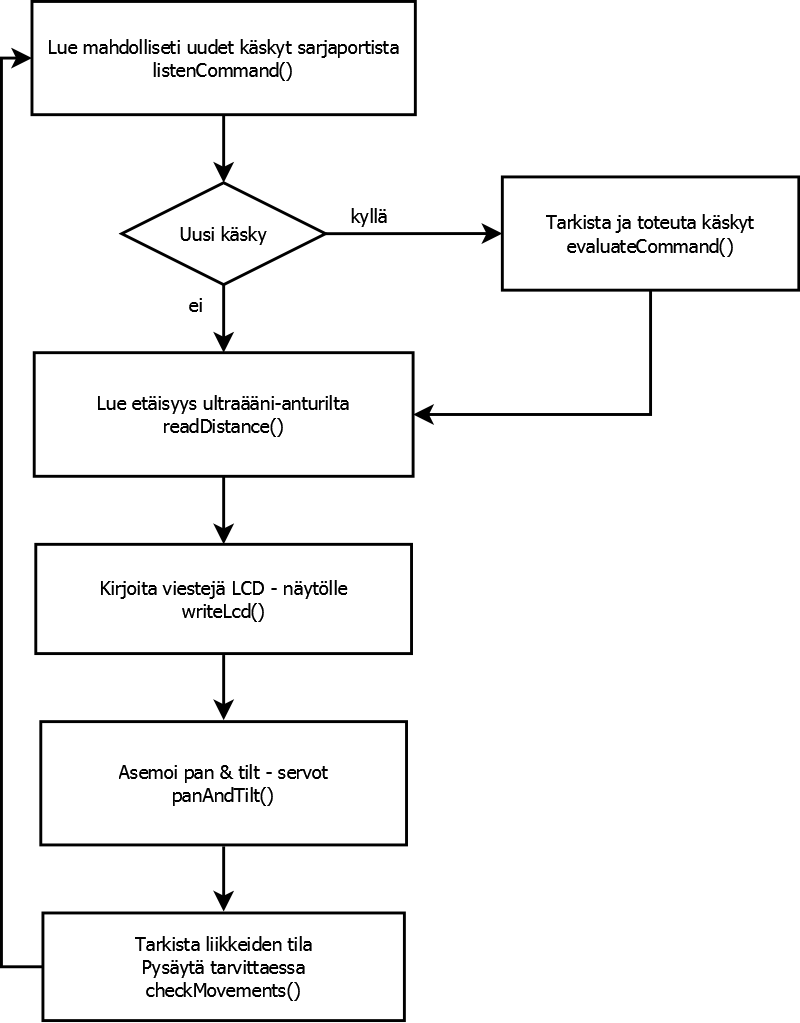
moveStop -metodi käynnistää LCD-näytön ja asettaa Arduinon Motor shieldin moodiin jossa kaikki moottorit ovat pysähtyneet.

moveTime -metodi laskee ”kellon ajan”, jonka jälkeen kaikki liikkeet pysäytetään ja sijoittaa ajan movementStopTime -muuttujaan.

checkMovements -metodi huolehtii liikkeiden pysäyttämisestä. Eteenpäin liikkuessa liike pysäytetään, jos este on alle 10cm etäisyydellä. Muutoin kaikki liikkeet pysäytetään jos nykyinen aika on movementStopTime -muuttujan aikaa suurempi.

## Ohjelman rakenne

Ohjelman alussa määritetään käytettävät kirjastot ja määritellään ohjelmassa käytettävät vakiot ja muuttujat. Tämän jälkeen Arduino-ohjelmissa suoritetaan yhden kerran erityinen setup -metodi, jossa asetetaan olioiden instansseille parametrit ja määritellään käytettävien pinnien asetukset.



1. Arduino-ohjelman pääsilmukka

Seuraavaksi ohjelman suoritus siirtyy pääsilmukkaan, jota toistetaan koko ohjelman toiminnan ajan. Ohjelmoinnissa pääajatus oli pitää ohjelman suoritus jatkuvana. Toisin sanoen ohjelman ei sallittu pysähtyvän odottamaan mitään. Esimerkiksi käskyt luetaan merkki kerrallaan yhden silmukan kierroksen aikana, eikä pysäytetä ohjelman suoritusta käskyn lukemisen ajaksi. Tämä mahdollistaa samanaikaisen käskyjen lukemisen, liikkumisen ja etäisyyden seuraamisen.

# Raspberry pi ohjelmat

## Asennetut ohjelmat

Raspberry Pin käyttöjärjestelmänä toimii Rasbian Jessie with pixel käyttöjärjestelmä. Rasbian pohjautuu Debian linux distribuutioon [7].

Käyttöjärjestelmään on asennettu kuvien jakamiseksi Apache2 Web palvelin [8], kuvien ottamiseksi fswebcam ohjelma [9] ja etätyöpöytäyhteyttä varten xdrp remote desktrop protokolla [10]. Arduino ohjelmointia varten on asennettu Arduino IDE.

## Skriptit

Skripti takePicture.sh ottaa kuvan webkameralla ja tallentaa kuvan webpalvelimelle. Skripti on ainut shelli skrpiti, sillä muut skriptit on koodattu Pythonilla. Skriptit tiltUp.py, tiltDown.py, panRight.py, panLeft.py ja centerCamera.py kääntelevät webkameraa. Skriptit move.py, reverse.py, turnLeft.py, turnRight.py ja stop.py ohjaavat Roverin liikkeitä. Kaikki Python skriptit välittävät käskyn Arduinolle sarjayhteydellä.



1. move.py Python skriptin koodi. Kaikki Python skriptit ovat tämän kaltaisia.

# Yhteydet raspberryyn

Android puhelimesta laitetaan mobiilitukiasema päälle. Raspberry Pi on asetettu yhdistymään puhelimesta tarjottuun WLAN -verkkoon automaattisesti. Samaan WLAN -verkkoon voi liittyä myös Windows 10 kannettavalla.

Puhelimella voidaan ottaa SSH-yhteys Raspberry Pin SSH-palvelimelle. SSH:n kautta voidaan suorittaa Shell ja Python skriptejä. Puhelimeen asennetulla JuiceSSH -clientillä [11] voidaan avata pääteyhteys. Lisäksi puhelimeen on asennettu Rasberry SSH -ohjelma [12]. Rasberry SSH -ohjelmaan luodaan nappeja, jotka suorittavat skriptejä SSH yhteyden kautta.



1. Raspberry SSH

Kannettavalla tietokoneella Raspberryyn voidaan ottaa SSH -yhteys PuTTY ohjelmalla [13] ja Windowsin etätyöpöytäyhteydellä.

# Yhteenveto

Harjoitustyön lopputuloksena syntynyt Rover täyttää alussa sille asetetut vaatimukset. Harjoitustyön tekemiseen ja raportin kirjoittamiseen kului aikaa noin 40 tuntia.

Rakennusvaiheen aikana ilmeni muutama ennakoimaton ongelma. Liikkumisen aikana LCD-näyttö sekosi. Häiriö aiheutui moottoreiden käytöstä. Ongelma ratkaistiin sammuttamalla LCD-näyttö liikkeiden ajaksi. Toinen ongelma oli Roverin kääntyminen. Telaketjuilla varustetut laitteet saadaan kääntymään pyörittämällä teloja vastakkaisiin suuntiin, mutta neljällä pyörällä syntyy kaksi erillistä voimaparia ja kääntyminen ei onnistu. Ongelma ratkaistiin vähentämällä merkittävästi eturenkaiden pitoa, jolloin Rover kääntyy nätisti toisen voimaparin keskipisteen ympäri.

Ultraäänianturin tarkkuus ja luotettavuus jättävät toivomisen varaa. Etäisyyden mittausta voisi parantaa vaihtamalla anturityyppiä.

Kaikki käytetyt Anturit olivat digitaalisia, joten Rover olisi ollut mahdollista toteuttaa ilman Arduinoa. Arduino kuitenkin tuntui laitteistoläheisemmältä ja koska kurssilla käsiteltiin Linuxia, Raspberry Pi:tä ja Arduinoa haluttiin harjoitustyössä yhdistää Arduino ja Raspberry.

Kurssi harjoitustöineen on ollut mielenkiintoinen ja antoisa. Kurssin parasta antia on ollut ohjelmoinnin ulottaminen ”tietokonemaailmasta” reaalimaailmaan.

Lähteet

1. Frizing ohjelma, Saatavissa: <http://fritzing.org/home/>
2. Arduino IDE, Saatavissa: <https://www.arduino.cc/en/Main/Software>
3. LiquidCrystal Arduino kirjasto, saatavissa: <https://www.arduino.cc/en/Reference/LiquidCrystal>
4. NewPing Arduino kirjasto, Tim Eckel, Saatavissa: <http://playground.arduino.cc/Code/NewPing>
5. Servo Arduino kirjasto, Saatavissa: <https://www.arduino.cc/en/Reference/Servo>
6. Tekniikan Taulukkokirja, 2007, Esko Valtanen, sivu 238
7. Raspbian käyttöjärjestelmä, Saatavissa : <http://www.raspbian.org/>
8. Apache2 asennusohje Raspberry Pille, Saatavissa: <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>
9. Fswebcam asennus- ja käyttöohje Raspberry Pille, Saatavissa: <https://www.raspberrypi.org/documentation/usage/webcams/>
10. Etätyöpöytäyhteyden asennusohje Raspberry Pille, Saatavissa: <https://www.maketecheasier.com/enabling-remote-desktop-access-on-raspberry-pi/>
11. JuiceSSH android sovellus, Saatavissa: <https://juicessh.com/>
12. Raspberry SSH android sovellus, Saatavissa: <https://play.google.com/store/apps/details?id=uk.co.knowles_online.raspberryssh&hl=fi>
13. Putty, SSH sovellus Windows käyttöjärjestelmälle, Saatavissa: [http://www.putty.org/](http://www.putty.org/%20)