Sakarias Ahola

ITSEOHJAUTUVA RASPBERRY PI -ROBOTTIAUTO

ITSEOHJAUTUVA RASPBERRY PI -ROBOTTIAUTO

Sakarias Ahola
Opinnäytetyö
Kevät 2019
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Sakarias Ahola
Opinnäytetyön nimi suomeksi: Itseohjautuva Raspberry Pi -robottiauto
Työn ohjaaja: Pekka Alaluukas
Työn valmistumislukukausi ja -vuosi: kevät 2019
Sivumäärä: 27 + 0 liitettä

Tämän opinnäytetyön tarkoituksena oli tutustua Python-ohjelmointikieleen sekä sen ominaisuuksiin Raspberry Pi -tietokoneeseen perustuvassa robotiikassa. Aihe valittiin henkilökohtaisesta mielenkiinnosta.

Työn kohteena oli Raspberry Pin ympärille rakennettu robottiauto. Tavoitteena oli ohjelmoida robotti itseohjautuvaksi sekä kykeneväksi palaamaan ajamaansa reittiä takaisin. Näihin toimintoihin liittyen robotille annettaisiin käskyjä infrapunakaukosäätimen avulla.

Raspberry Pin käyttöjärjestelmänä toimi Raspbian for Robots. Esteentunnistus ja kääntymissuunnan valitseminen toteutettiin servon ja etäisyyssensorin yhdistelmällä. Kaukosäätimen infrapunasignaalit tunnistettiin infrapunasensorin avulla.

Työn tuloksena tehty Raspberry Pin ja robotin sensorit yhdistävä Python-ohjelmakoodi mahdollisti robottiauton liikkumisen itseohjautuvasti ja esteitä väistäen. Ajamisreitin seuraamisessa lähtöpisteeseen ei täysin onnistuttu käännösten pienten käytännön epätarkkuuksien vuoksi. Robotin liikkeet olivat kuitenkin mahdollisia suorittaa käänteisesti ohjelmakoodin avulla.

Opinnäytetyöstä saatiin hyvä pohja Python-ohjelmointikielen yleisemmällekin käytölle. Raspberry Pi -robottiauton jatkokehityksessä sen sensoreiden joukkoon voisi lisätä gyroskoopin tarkentamaan robotin tekemiä käännöksiä.

Asiasanat: Python, Raspberry Pi, robottiauto

ABSTRACT

Oulu University of Applied Sciences
Information Technology, software development

Author(s): Sakarias Ahola
Title of thesis: A self-driving Raspberry Pi robot car
Supervisor: Pekka Alaluukas
Term and year when the thesis was submitted: Spring 2019
Pages: 27 + 0 appendices

The purpose of this thesis was to get acquainted with the Python programming language as well as its usage in robotics utilizing a Raspberry Pi single board computer. This goal was chosen out of personal interest.

The object of the programming was a Raspberry Pi robot car. The goal regarding the robot car was to program it to receive and recognize signals sent from a remote control and to execute functions in accordance with them. The functions would be comprised of an automatic driving mode and a reverse mode for the robot to go back the route it has driven.

Raspbian for Robots was used as the operating system for the Raspberry Pi single-board computer. The obstacle detection and pathfinding abilities were implemented with the help of a distance sensor rotated by a servo. An infrared sensor was used to catch the different remote infrared signals.

As the result of this work, the Python program running on the Raspberry Pi and controlling the robot's sensors allowed the robot car to drive around automatically while turning away from obstacles. The backtracking wasn't fully achieved due to practical inaccuracies, but on the part of the program code the drives and turns of the robot were executable in reverse.

The experience got from this thesis gave a good basis for further Python programming. To improve the robot car itself, a gyroscope could be added in future to increase the accuracy of turns and the reverse driving.

Keywords: Python, Raspberry Pi, robot car

SISÄLLYS

TIIVISTELMÄ 3
ABSTRACT 4
SISÄLLYS 5
1 JOHDANTO 6
2 RASPBERRY PI JA KÄYTTÖJÄRJESTELMÄN ASENNUS 8
2.1 Raspberry Pi 8
2.2 Raspbian 9
2.3 Raspbian for Robots -käyttöjärjestelmän asennus Raspberry Pi 3 -tietokoneeseen 9
3 GOPIGO-PIIRILEVY JA SENSORIT 11
4 GOPIGO-ROBOTIN OHJELMAKOODI 14
4.1 Python 14
4.2 Ohjelmakoodin moduulit ja objektit 14
4.3 Ohjelmakoodin funktiot 15
4.4 Robotin liikkeiden tallennus ja paluutoiminto 17
5 TULOKSET 22
6 YHTEENVETO 25
LÄHTEET 26

1 JOHDANTO

GoPiGo on Dexter Industries -nimisen yhtiön kehittämä robottiauto Raspberry Pi -tietokoneelle. Laaja sensorivalikoima sekä Raspberry Pin omat ominaisuudet mahdollistavat monenlaiset projektit, ja GoPiGo-robotti on suosittu sekä harrastelijoiden keskuudessa että opetusmaailmassa.

Raspberry Pi on pieni yhden piirilevyn tietokone, joka on varustettu kaikilla käyttöjärjestelmän pyörittämiseen sekä GoPiGo-robotin ohjelmoimiseen tarvittavilla komponenteilla ja ominaisuuksilla. Raspberry Pin porttien joukosta löytyvät mm. USB-portit näppäimistölle ja hiirelle sekä HDMI-portti monitorin liittämiseen ja käyttämiseen. GoPiGo-robotissa Raspberry Pi toimii liitettynä GoPiGo-piirilevyyn.

GoPiGo-piirilevy on GoPiGo-robotin ydinosa. Siihen kytketään robotin moottorit ja sensorit, ja niitä ohjataan piirilevyn kautta. Piirilevy yhdistetään Raspberry Pihin liittämällä viimeksi mainitun pinnit GoPiGo-piirilevyssä oleviin soveltuviin pinnivastaanottimiin. Piirilevyt kommunikoivat tämän pinniliitännän kautta. GoPiGo-robotille annetaan virtaa syöttämällä se suoraan GoPiGo-piirilevyyn, joka robotin ajaessa jatkaa virtaa myös liitettyyn Raspberry Pihin. Raspberry Pi kykenee näin pyörittämään robotin ohjelmakoodia.

GoPiGo-robotissa käytetään tässä työssä infrapunasensoria, etäisyyssensoria sekä servoa. Nämä kaikki kytketään GoPiGo-piirilevyyn niille soveltuviin portteihin. Etäisyyssensori kiinnitetään servoon, joka infrapunasensorin tavoin kiinnitetään robotin runkoon.

Tässä työssä tavoitteena on tutustua Python-ohjelmointikieleen ja oppia sen käyttöä Raspberry Pi -tietokonetta käyttävässä robotiikassa. Raspberry Pi -piirilevy, GoPiGo-piirilevy sekä siihen liitettävät sensorit tulee yhdistää Python-ohjelmointikielen avulla toimivaksi kokonaisuudeksi.

Itse GoPiGo-robottiin liittyvä tavoite on ohjelmoida se vastaanottamaan infrapunasensorin kautta erilaisia signaaleja kaukosäätimestä ja suorittamaan toimintoja niiden mukaan. Robotin tulee pystyä liikkumaan itseohjautuvasti ympäriinsä, tun-

nistaen edessään olevia esteitä keulaan liitetyn etäisyyssensorin tekemien mittausten avulla. Sensorin havaitessa esteen tietyn matkan päässä ja robotin pysähtyessä etäisyyssensorin tulee servon liikuttamana skannata robotin ympäristöä uuden kääntymissuunnan valitsemista varten. Ohjelmakoodin täytyy näin valita robotille uusi kääntymissuunta etäisyyssensorin tekemien mittauksien perusteella. GoPiGo-robotin tulee muistaa ajamansa matkat ja tekemänsä käännökset sekä infrapunakaukosäätimestä signaalin saatuaan kyetä suorittamaan niitä käänteisesti ja näin palaamaan ajamaansa reittiä takaisin robottiauton lähtöpisteeseen asti.

Robotti tulee voida siirtää kaukosäädintä ja infrapunasensoria käyttäen itseohjautuvasta tilasta palaamistilaan milloin vain, ja myös toisin päin. Näistä kahdesta toimintotilasta pitää myös olla mahdollista siirtyä lepo- tai odotustilaan, jossa robotti on pysähdyksissä ja odottaa signaalia siirtyä jompaankumpaan edellä mainituista tiloista. Tallennettujen ajomatkojen ja käännösten tulee poistua muistista ainoastaan silloin, kun robotti aloittaa itseohjautuvan, automaattisen ajamisen. Jos paluutoiminto ja robotti pysäytetään eikä uutta automaattista ajoa aloiteta, robotin tulee pystyä jatkamaan paluutoimintoa ja ajoreitin seuraamista takaisin.

2 RASPBERRY PI JA KÄYTTÖJÄRJESTELMÄN ASENNUS

2.1 Raspberry Pi

Raspberry Pi on yhden piirilevyn tietokone, jossa on sen pienestä koosta huolimatta kaikki perustietokoneen käyttöominaisuudet. Tämän mahdollistavat sen sisältämät prosessori, keskusmuisti, grafiikkapiiri ja erilaiset portit. Raspberry Pi -tietokoneita on julkaistu vuosien varrella useita. (1.) Tässä työssä käytettiin Raspberry Pin malliversiota 3 (kuva 1).

KUVA 1. Raspberry Pi 3 (2)

Raspberry Pi 3 -tietokoneessa prosessorina toimii 64-bittinen, 1,2 gigahertsin Cortex-A53-neliydinsuoritin, joka on 50 % tehokkaampi kuin Raspberry Pi 2:n prosessori. Monitorin käytön mahdollistaa 400 megahertsin VideoCore IV -grafiikkaprosessori. Toisin kuin aiemmissa malleissa, Raspberry Pi 3:ssa on Ethernetin lisäksi sisäänrakennettu Wi-Fi. (1.)

Raspberry Pin kiintolevynä toimii SD-muistikortti. SD-kortti on liitettävissä Raspberry Pi -piirilevyyn, joka voi näin käyttää korttiin ajettua käyttöjärjestelmää sekä jäljellä olevaa muistitilaa. Raspberry Pihin on myös mahdollista liittää USB-muistia USB-porttien kautta. Piirilevyssä itsessään on 1 gigatavu keskusmuistia, jota myös grafiikkapiiri käyttää.

Raspberry Pi 3 käyttää kohtuullisen keveässä käytössä virtaa noin saman verran kuin edelliset Raspberry Pi -mallit. Raskas kuormitus voi kuitenkin kaksinkertaistaa virrankäytön. (1.)

2.2 Raspbian

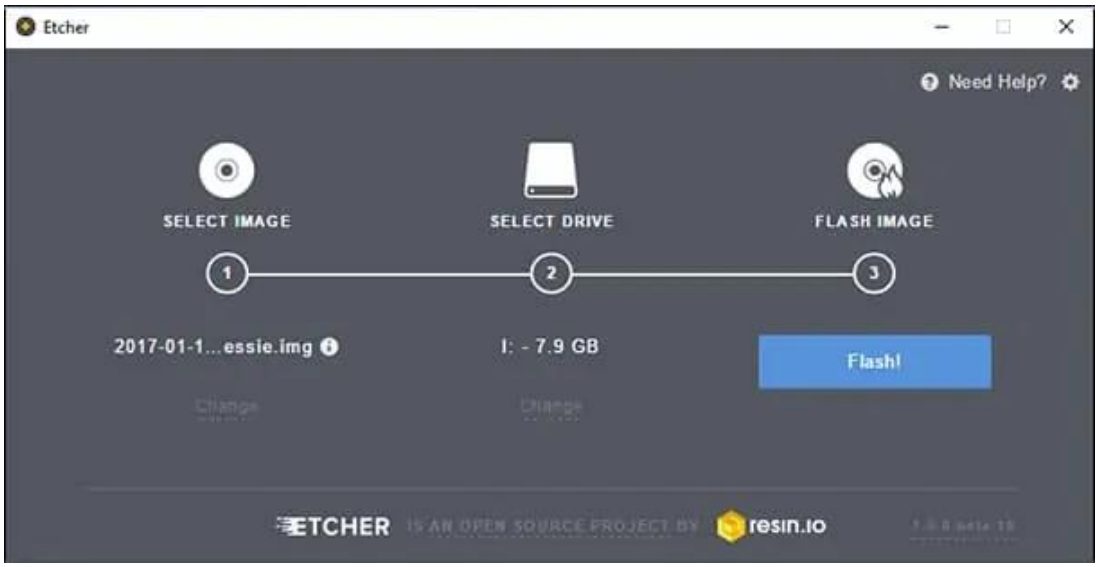
Raspberry Pi 3 -tietokoneella voi käyttää monia käyttöjärjestelmiä, sekä Linux-pohjaisia että muunlaisia. Raspberry Pin virallinen käyttöjärjestelmä on kuitenkin Debian-käyttöjärjestelmään perustuva Raspbian. (3.)

GoPiGo-robotin valmistaja, Dexter Industries, on kehittänyt tuotettaan varten Raspbian-käyttöjärjestelmästä oman version, Raspbian for Robotsin, joka nimensä mukaisesti soveltuu hyvin GoPiGo-robotin ohjelmointiin (4). Tässä työssä Raspberry Pin käyttöjärjestelmäksi valittiin juuri Raspbian for Robots.

2.3 Raspbian for Robots -käyttöjärjestelmän asennus Raspberry Pi 3 -tietokoneeseen

Dexter Industries -nettisivu tarjoaa linkin Raspbian for Robots -käyttöjärjestelmätiedoston lataamiseen. Lataaminen suoritettiin tavalliselle kannettavalle tietokoneelle. Tiedosto oli pakattuna, joten lataamisen jälkeen se täytyi purkaa asiaankuuluvalla ohjelmalla. Purkamiseen käytettiin ilmaista WinRAR-nimistä tiedostojen purkamiseen ja pakkaamiseen tarkoitettua ohjelmaa. Tämän jälkeen Raspbian for Robots oli noin 4 gigatavun suuruinen img-tiedosto. (3.)

Raspbian for Robots -käyttöjärjestelmän asentaminen Raspberry Pihin tapahtui microSD-kortin sekä Etcher-nimisen tiedostonpolttamisohjelman avulla. Etcher on ilmainen ohjelma, johon Dexter Industries -nettisivu tarjoaa linkin sekä ohjelman käyttöohjeet. (Kuva 2.)



KUVA 2. Etcher-ohjelma (5.)

MicroSD-kortti kytkettiin adapterin avulla kannettavan tietokoneen microSD-porttiin, ja käyttöjärjestelmätiedosto poltettiin microSD-korttiin Etcher-ohjelmalla. Muistikortin koko oli 8 gigatavua, mikä on riittävä muistitila Raspbian for Robots -käyttöjärjestelmälle (1).

Muistikortti irrotettiin kannettavasta tietokoneesta ja kytkettiin GoPiGo-robotissa Raspberry Pin SD-porttiin. Kyseinen SD-portti on tyyppiä micro, joten Raspberry Pin kanssa adapteria ei tarvinnut käyttää.

Sekä Raspberry Pi että muistikortti toimivat: virtaa saadessaan Raspberry Pi käynnistyi, ja HDMI-porttiin kytketty monitori näytti käyttöjärjestelmän käynnistysprosessin ja sitä seuraavan työpöytänäkymän ongelmitta. Myös Raspberry Pin USB-portteihin liitetyt näppäimistö ja hiiri olivat toimivia käyttöjärjestelmän kanssa.

Raspbian for Robots -käyttöjärjestelmän käyttöönoton lopuksi suoritettiin sen ohjelmistopäivitys. Tämä vaati toimivan internetyhteyden, jonka mahdollisti Raspberry Pissä oleva langaton lähiverkko-ominaisuus. Tavallisesta älypuhelimesta jaettiin sen oma internetyhteys, jonka Raspberry Pi tunnisti ja otti käyttöön. Tämän jälkeen Raspbian for Robots -päivitysten lataus ja asennus suoritettiin käyttöjärjestelmän mukana tulleella päivitysohjelmalla.

3 GOPIGO-PIIRILEVY JA SENSORIT

GoPiGo-piirilevy on GoPiGo-robotin avainosa, ja muodostaa yhdessä Raspberry Pin kanssa robotin keskuksen. GoPiGo-piirilevyssä oleva ATmega328-mikro-kontrolleri kommunikoi Raspberry Pin kanssa pinniliitännän kautta ja ohjaa näin robotin moottoreita ja käytettäviä sensoreita, joiden liitäntää varten piirilevyssä on portit. (6.) (Kuva 3.)

KUVA 3. GoPiGo-piirilevy (7)

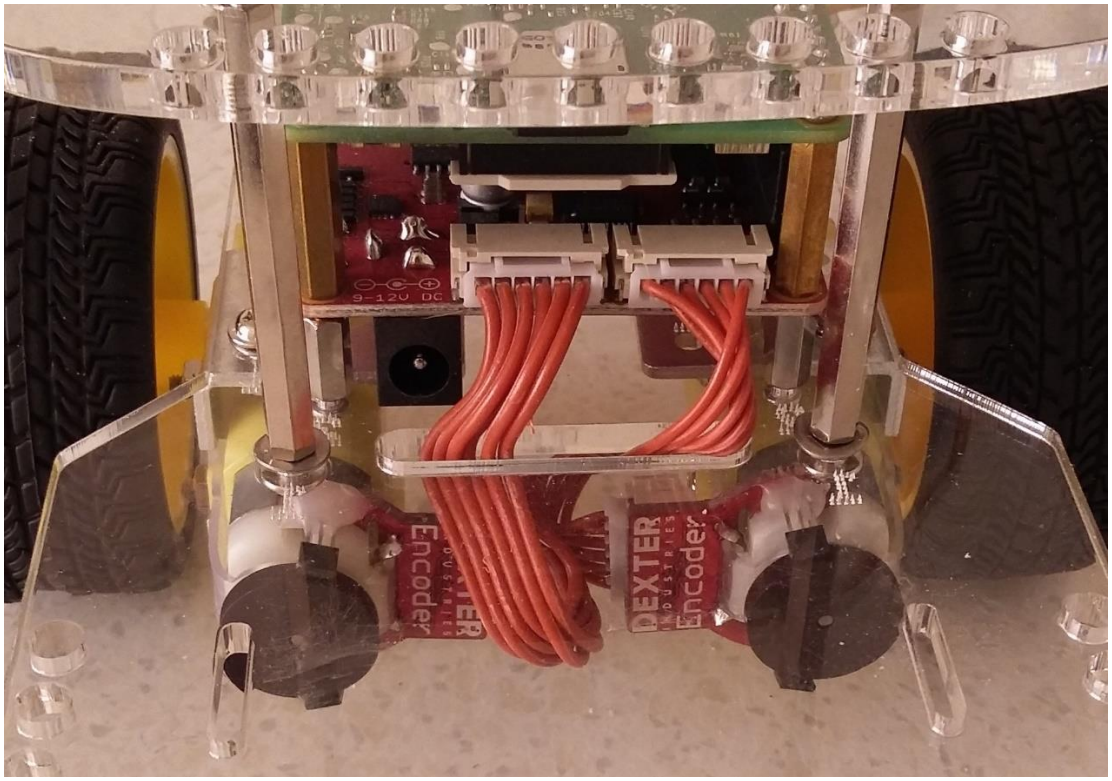
Piirilevyyn kytkettävä ja GoPiGo-robotin runkoon kiinnitettävä uudelleenladattava akku tarjoaa virransyötön moottoreille ja sensoreille. Samalla myös Raspberry Pi saa akusta virtaa piirilevyjen pinniliitännän ansiosta.

Tässä työssä käytettiin GoPiGo-piirilevyn liitännäisinä (akun sekä oikean ja vasemman moottorin ohella) servoa, etäisyyssensoria sekä infrapunasensoria. Servo on pieni teline, joka kiinnitetään moottoristaan GoPiGo-robotin etuosaan ja johon puolestaan voi kiinnittää erilaisia sensoreita. Servomoottori kääntää servo-telinettä ja samalla sensoreita ohjelmakoodin ohjaamana. Etäisyyssensori mittaa etäisyyttä pienten laserpulssien ja niiden heijastusten vastaanottamisen välillä

kuluneen ajan perusteella (8). Infrapunasensori tunnistaa ja vastaanottaa infrapunasäteilyn muodossa olevia signaaleja (9).

Servoon liitettiin tässä työssä etäisyyssensori, joka pystyy näin mittaamaan etäisyyksiä useasta suunnasta servon kääntyillessä. Infrapunasensorin parina toimii infrapunakaukosäädin, jolla voi lähettää infrapunasensorin välityksellä robotille käskyjä ja näin ohjata sen toimintaa.

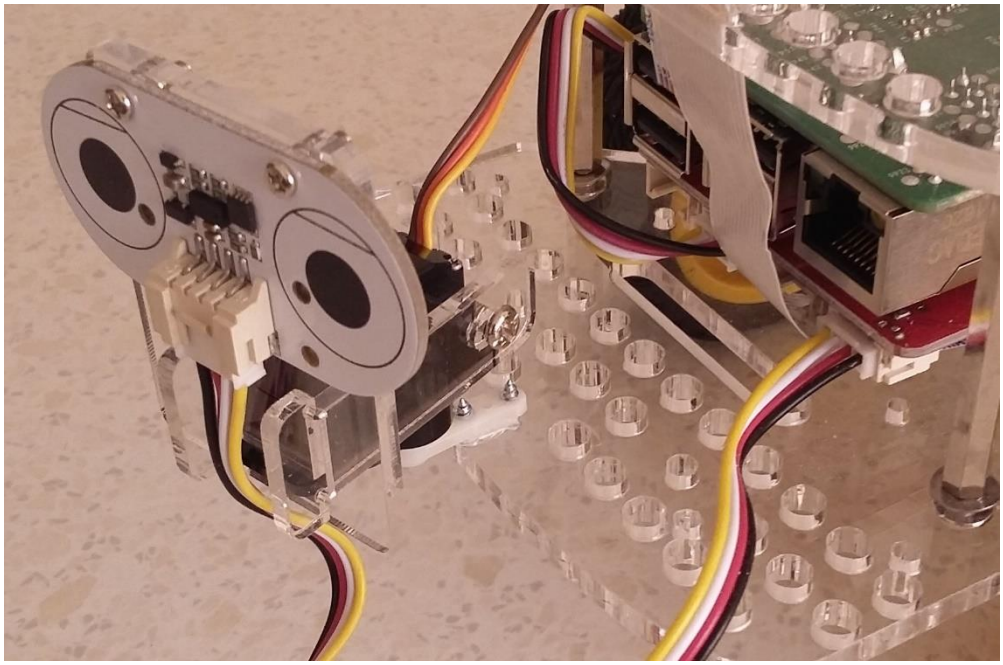
Moottoreita on GoPiGo-robotissa kaksi, yksi vasenta ja yksi oikeata rengasta varten. Moottorit kytkettiin GoPiGo-piirilevyyn, ja Raspberry Pi -piirilevyllä ajettavalla ohjelmakoodilla saattoi näin lähettää GoPiGo-piirilevyn kautta käskyjä kummallekin moottorille renkaiden pyörittämiseen – joko eteenpäin tai taaksepäin tehtävän mukaan. (Kuva 4.)



KUVA 4. GoPiGo-robotin moottorit ja liitäntä

Robottiauton servon ja etäisyyssensorin johdot kytkettiin myös kummatkin GoPiGo-piirilevyn portteihin. Servo kiinnitettiin robotin keulaan, missä se kykeni kääntämään pienen moottorin avulla itseään sekä mitä tahansa liitännäissensoria

robotin sivuilta toiselle. Servoon kiinnitettiin etäisyyssensori, joka kykeni näin ottamaan arvoja eri puolilta ilman itse robotin liikkumista. (Kuva 5.)



KUVA 5. GoPiGo-robotin runkoon kiinnitetty servomoottori, servo sekä etäisyyssensori

4 GOPIGO-ROBOTIN OHJELMAKOODI

4.1 Python

Python on monipuolinen korkean tason ohjelmointikieli. Sen avainominaisuuksiin kuuluu mm. koodirakenteen selkeys ja helppo luettavuus sekä laaja oheiskirjasto ohjelmoinnin tukemiseen. (10.)

Raspbian for Robots -käyttöjärjestelmään oli valmiiksi sisällettynä ohjelmointiympäristö Python-ohjelmointia varten. Ohjelmointiympäristö taas sisälsi GoPiGo-robotin ohjelmointia varten tehdyn *easygopigo3*-moduulin, joten Pythonin valinta robotin ohjelmointikieleksi oli helppo. Moduulit ovat ohjelmakoodia sisältäviä tiedostoja, jotka tarjoavat erinäisiä toimintoja ulkopuoliseen ohjelmakoodiin (11).

4.2 Ohjelmakoodin moduulit ja objektit

GoPiGo-robotin Python-ohjelmakoodin laatimisessa otettiin aluksi käyttöön *easygopigo3*-moduuli sekä *sleep*-toiminto moduulista *time* (kuva 6). *Easygopigo3* oli välttämätön, sillä se mahdollisti GoPiGo-robotin moottorien hallinnan eli liikkumisen sekä siihen liitettyjen sensoreiden käytön ja datan vastaanottamisen niistä. *Time*-moduuli tarjoaa nimensä mukaisesti aikaan liittyviä toimintoja, ja sen sisältämää *sleep*-toimintoa käytettiin tahdistamaan ohjelmakoodin ja robotin toimintaa.

```
import easygopigo3 as easy
from time import sleep
```

KUVA 6. Easygopigo3-*moduulin ja* sleep-*funktion käyttöönotto*

Robotin servo, etäisyyssensori, infrapunasensori sekä itse GoPiGo-robotti täytyi myös alustaa koodissa ennen niiden varsinaista ohjelmointia. Tämä tapahtui luomalla niistä objektit, joiden kautta niiden hallinta koodissa oli mahdollista. *Easygopigo3*-moduulin kautta luotiin *gopigo*-objekti, johon viittaamalla taas alustettiin servo ja sensorit. (Kuva 7.)

```
gopigo = easy.EasyGoPiGo3()
servo = gopigo.init_servo()
distance_sensor = gopigo.init_distance_sensor()
remote = gopigo.init_remote("AD1")
```

KUVA 7. Python-koodissa käytettävien objektien alustus

4.3 Ohjelmakoodin funktiot

Ohjelmakoodi sisältää kolme itse määriteltyä ja kirjoitettua toimintoa eli funktiota. Ajaessaan ympäriinsä ja kääntyessään samalla pois esteistä robotti käyttää vuoron perään *Drive*- ja *Turn*-funktioita, kun taas palaaminen ajettua reittiä takaisin tapahtuu *Reverse*-funktion avulla. *Drive*, *Turn* ja *Reverse* tarkoittavat siis niimiensä mukaisesti ajamistoimintoa, kääntymistoimintoa ja paluutoimintoa.

Ajamistoiminnossa robotti ajaa eteenpäin, kunnes keulan servoon liitetty etäisyyssensori tunnistaa esteen 25 senttimetrin päässä. Robotin moottoreissa olevat enkooderit mittaavat renkaiden pyörimisliikkeitä ja robotin ajomatkaa, ja kun robotti pysähtyy, funktio ottaa enkoodereista mittausarvon ja tallentaa sen ajomatkamittauksia varten luotuun *drives*-listaan. Renkaat pysäyttävän *easygopigo3*-funktion jälkeen robotti odottaa pienen hetken (sekunnin neljännesosan) ennen mittausarvon ottamista ja tallentamista, jotta varmistettaisiin sen tarkkuus (kuva 8). Ajamistoiminto loppuu tähän, ja sitä seuraa aina välittömästi toinen funktio eli kääntyminen.

```
45          gopigo.stop()
46          sleep(0.25)
47          drives.append(gopigo.read_encoders_average(average))
```

KUVA 8. Renkaiden pyörimismittausarvon tallentaminen drives*-nimiseen listaan*

Kääntymistoiminto alkaa servon suorittamalla kääntymisliikkeellä vasemmalta oikealle. Etäisyyssensori mittaa samalla etäisyyksiä eri kulmista robotin kummaltakin puolelta. Mittausarvojen perusteella robotti päättelee, kummalla puolella on esteettömämpää ajaa, ja kääntyy sinne. Ajomatkamittausten tapaan kääntymisten suunnat tallennetaan listaan – tässä tapauksessa kääntymisille tarkoitettuun *turns*-listaan. Käännösten arvot tallennetaan käänteisinä, sillä paluutoiminnossa

robotti on kääntynyt ja ajamisreitin seuraamisessa takaisinpäin käännökset tulee suorittaa vastakkaisina. Käännöksen tallentamisen jälkeen ohjelmakoodi ja robotti palaavat ajamistoimintoon.

Robotin automaattinen ajaminen ja suunnistus voidaan keskeyttää sekä ajamisettä kääntymisfunktiossa. Keskeytyksen voi tehdä milloin tahansa robotin eteenpäin liikkumisen aikana tai heti kääntymisen jälkeen. Infrapunasensori vastaanottaa kaukosäätimestä tulevan keskeytyssignaalin, joka pysäyttää robotin. Keskeytystapauksissa kuljettu matka tai tehty käännös tallennetaan normaalisti. Ohjelmakoodi kuitenkin poistuu *Drive*- ja *Turn*-funktioiden silmukasta, ja infrapunasignaalista eli painetusta kaukosäätimen painikkeesta riippuen robotti joko odottaa paikoillaan uutta signaalia tai siirtyy suoraan palaamaan ajamisreittiä takaisin. Koodissa signaalin arvo *7* vastaa painiketta nro 2, ja *8* painiketta nro 3. Kummatkin muuttavat muuttujan *interrupted* arvoksi *1*, mikä aiheuttaa keskeytyksen, mutta arvo *7* käynnistää lisäksi paluutoiminnon (kuva 9).

```
78      value = remote.read()
79      if (value == 7) or (value == 8):
80          interrupted = 1
81          if value == 7:
82              sleep(0.25)
83              Reverse()
```

KUVA 9. Keskeytys Turn*-funktiossa*

Paluutoiminnossa robotti kääntyy ensin paikoillaan 180 astetta ja alkaa sitten suorittamaan tallennettuja ajomatkoja ja käännöksiä käänteisessä järjestyksessä. Käännökset tehdään vastakkaisina kulkusuunnan muutoksen vuoksi. Robotti jatkaa liikkumista, kunnes ohjelmakoodi on päässyt listojen alkupisteeseen ja takaisin ajaminen on suoritettu. Kuten *Drive*-funktiossa, infrapunasensori seuraa kuitenkin kaukosäädinsignaalia, ja ajamisen voi keskeyttää milloin vain. Tällöin robotti tallentaa toiminnon etenemisen ja siirtyy lepotilaan, missä se odottaa signaalia joko jatkaakseen keskeytettyä paluutoimintoa tai aloittaakseen uuden ajamistilan signaalin arvolla *6* eli painikkeella nro 1. Jälkimmäisessä tapauksessa jo tallennetut ajot ja käännökset poistetaan listoista.

4.4 Robotin liikkeiden tallennus ja paluutoiminto

Robotin automaattisessa ajamistilassa sekä ajofunktio että kääntymisfunktio sisältävät robotin liikkeen tallentamisen. Ensimmäinen tallentaa nimensä mukaisesti suoritetut ajomatkat ja jälkimmäinen tehdyt käännökset. Ajomatkat ja käännökset tallennetaan erillisiin listoihin. *Drive*-funktio käyttää *drives*-listaa ajomatkojen tallentamiseen, ja *Turn*-funktio käyttää *turns*-listaa käännösten tallentamiseen. (Kuva 10.)

```
23  def Drive():
24      global interrupted
25      global stuck
26      gopigo.reset_encoders()
27      if distance_sensor.read_mm() <= 250:
28          stuck += 1
29      else:
30          stuck = 0
31      while distance_sensor.read_mm() > 250:
32          gopigo.backward()
33          value = remote.read()
34          if (value == 7) or (value == 8):
35              gopigo.stop()
36              sleep(0.25)
37              drives.append(gopigo.read_encoders_average(average))
38              turns.append(0)
39              interrupted = 1
40              if value == 7:
41                  Reverse()
42              break
43      if interrupted == 0:
44          gopigo.stop()
45          sleep(0.25)
46          drives.append(gopigo.read_encoders_average(average))
```

KUVA 10. Robotin ajamistoiminto ja ajomatkojen tallennus

Drive-funktion suorituksessa robotti ajaa eteenpäin, kunnes etäisyyssensorin mittaama etäisyys ei ole suurempi kuin 25 senttimetriä (ohjelmakoodissa etäisyyden mittayksikkö on millimetri). Robotin ajaessa eteenpäin *while*-silmukassa infrapunasensorin vastaanottamaa signaalin arvoa tarkastetaan koko ajan, ja arvo tallentuu aina *value*-muuttujaan. Jos signaalin arvo on *7* (hyppy suoraan paluutoimintoon) tai *8* (robotin siirtyminen lepotilaan), robotti pysähtyy ja tallentaa ajetun matkan *drives*-listaan *drives.append* -komennolla. Moottorien johdot kytkettiin tässä työssä GoPiGo-piirilevyyn toisinpäin, joten eteenpäin ajamisen komento on

ohjelmakoodissa *backward* eikä *forward*. Keskeytystapauksessa ajomatkan tallentamisen jälkeen käännöksille tarkoitettuun *turns*-listaan lisätään vielä tyhjä kääntymisarvo, jotta matka- ja käännöslistojen jäsenmäärä pysyy samana. (Kuva 10.)

Globaalin *interrupted*-muuttujan arvoksi asetetaan lopuksi *1*, jotta ohjelmakoodi tietää olla siirtymättä kuluvan funktion suorittamisen jälkeen enää kääntymisfunktioon. Jos keskeytystä ei tapahdu, ohjelmakoodi ainoastaan tallentaa ajetun matkan ja jatkaa sitten käännösfunktioon *interrupted*-muuttujan arvon muuttumatta. Globaali muuttuja *stuck* on merkittävä ainoastaan silloin, jos robotti epäonnistuu esteettömän kääntymissuunnan löytämisessä useamman kerran peräkkäin. Kun robotti havaitsee esteettömän suunnan kääntymisen jälkeen, *stuck* nollautuu. (Kuva 10.)

Kääntymisfunktiossa robotti suorittaa ympäristön skannauksen etäisyyssensorilla ja robotin kääntymisen joko vasemmalle tai oikealle. Toisin kuin *drive*-funktiossa, mahdollinen keskeytyssignaali kaukosäätimestä tarkastetaan ainoastaan funktion lopussa. Tällöin muuttuja *interrupted* arvoksi, joka on funktion alussa aina väistämättä *0*, tulee *1*, mikä lopettaa *drive*- ja *turn*-funktioiden silmukan funktion lopussa. (Kuva 11.)

```
def Turn():
    global interrupted
    leftside = 0
    rightside = 0
    i = 180
    servo.rotate_servo(i)
    while i > 109:
        if (i >= 140) and (i % 10 == 0):
            sleep(0.10)
            leftside += distance_sensor.read_mm()
        i -= 1
        servo.rotate_servo(i)
        sleep(0.01)
    while i > 38:
        if (i <= 80) and (i % 10 == 0):
            sleep(0.10)
            rightside += distance_sensor.read_mm()
        i -= 1
        servo.rotate_servo(i)
        sleep(0.01)
    servo.rotate_servo(109)
    if leftside > rightside:
        gopigo.turn_degrees(-90)
        turns.append(90)
    else:
        gopigo.turn_degrees(90)
        turns.append(-90)
    gopigo.stop()
    sleep(0.10)
    value = remote.read()
    if (value == 7) or (value == 8):
        interrupted = 1
        if value == 7:
            sleep(0.25)
            Reverse()
```

KUVA 11. Robotin kääntyminen

Turn-funktiossa etäisyysarvot jaetaan robotin vasemmalta puolelta ja oikealta puolelta otettuihin. Muuttujaan *leftside* tallennetaan vasemman puolen etäisyydet ja muuttujaan *rightside* oikean puolen etäisyydet. Etäisyyssensori ottaa arvoja 10 kääntymisasteen välein. Mittaamisen sallivat ehdot täyttyvät ohjelmakoodissa viisi kertaa robotin kummallakin puolella. (Kuva 11.)

Ohjelmakoodissa servon kääntymisarvo *109* vastaa käytännössä servon keskipistettä robottiin nähden. Funktion alussa servo, joka on ajaessa aina keskipisteessä eli osoittaa aina suoraan eteenpäin, kääntyy ensin vasemmalle ja sitten mahdollistaa etäisyyssensorin tekemän skannauksen kääntymällä kääntymisalueen oikeanpuoleiseen pisteeseen. (Kuva 11.)

Funktion lopussa ohjelmakoodi valitsee robotille kääntymissuunnan sen perusteella, kummalla *leftside*- ja *rightside*-funktioista on suurempi arvo eli kummalla puolella on suuremmat kokonaisetäisyydet ja suurempi kokonaistila. Kääntymisen yhteydessä käännös tallennetaan käännöslistaan vastakkaisena sen käänteistä suorittamista varten. (Kuva 11.)

Jos robotin edellinen toiminto oli automaattinen ajaminen, se tekee *Reverse*-funktion alussa 180 asteen käännöksen. Paluutoiminnosta lepotilaan ja takaisin siirto ei aiheuta käännöstä, sillä muuttuja *reverse* on silloin jo arvoltaan *1*, eli robotin alkukääntyminen tehtyjen liikkeiden käänteistä suorittamista varten on jo suoritettu. (Kuva 12.)

```
def Reverse():
    global reverse
    if reverse == 0:
        gopigo.turn_degrees(180)
        reverse = 1
    value = 0

    while len(drives) > 0:
        if turns[len(drives)-1] != 0:
            gopigo.turn_degrees(turns[len(drives)-1])
        turns.pop()
        sleep(0.25)
        gopigo.reset_encoders()
        while gopigo.read_encoders_average(average) > drives[len(drives)-1]:
            gopigo.backward()
            value = remote.read()
            if value == 8:
                gopigo.stop()
                sleep(0.25)
                remnant = drives[len(drives)-1] - gopigo.read_encoders_average(average)
                drives.pop()
                drives.append(remnant)
                turns.append(0)
                break
        if value != 8:
            gopigo.stop()
            drives.pop()
            sleep(0.10)
        else:
            break
```

KUVA 12. Ajetun reitin takaisinpäin seuraamisesta vastaava funktio

Funktion pääosiossa ohjelmakoodi suorittaa ja samalla poistaa *drives*- ja *turns*-listoissa olevia ajomatkoja ja käännöksiä. Ellei keskeytystä tapahdu, robotti jatkaa käänteistä suorittamista, kunnes listat ovat tyhjiä ja robotti on saavuttanut lähtöpisteen. Jos paluutoiminto keskeytetään, robotti vähentää suoritettavana

olevasta ajomatkasta jo suoritetun osan ja tallentaa jäljelle jääneen matkan alkuperäisen tilalle. Paluutoimintoa voidaan näin jatkaa, jos keskeytys vie robotin ainoastaan lepotilaan uuden automaattisen ajamisen sijaan. (Kuva 12.)

Robotin automaattisen ajon keskeytyessä käännöksiä ja ajomatka-arvoja on listoissa aina saman verran johtuen mahdollisesta tyhjästä käännösarvosta. Tästä syystä ajojen ja käännösten vuorotteleva suorittaminen aloitetaan paluutoiminnossa käännöksestä. Robotti ei varsinaisesti suorita käännöstä, jos se on tyhjä, mutta käy silti sen läpi ja poistaa sen listasta. (Kuva 12.)

Lepotilassa infrapunasignaalin arvo luetaan pienin väliajoin. Signaalin arvolla *6*, joka vastaa kaukosäätimen painiketta nro 1, robotti aloittaa ajamis- ja kääntymisfunktioita vuorottelevan silmukan. Painike nro 2 aloittaa paluutoiminnon eli *Reverse*-funktion. (Kuva 13.)

```
while True:
    sleep(0.10)

    if remote.read() == 6:
        drives = []
        turns = []
        interrupted = 0
        reverse = 0
        while interrupted == 0:
            Drive()
            if interrupted == 0:
                if stuck < 4:
                    Turn()
                else:
                    stuck = 0
                    gopigo.turn_degrees(180)
                    turns.append(-180)

    if (remote.read() == 7) and (len(drives) > 0):
        Reverse()
```

KUVA 13. GoPiGo-robotin lepotilassa ohjelmakoodi tarkkailee infrapunasignaalia

Muuttujan *stuck* arvo tarkastetaan aina kääntymisen yhteydessä. Arvo *4* viittaa jumiutumiseen käännössuunnan etsimisessä, ja robotin seuraava käännös on tällöin 180 astetta.

Drive- ja *Turn*-funktioiden silmukassa muuttujan *interrupted* arvo tarkastetaan aina ennen kummankin funktion suorittamista. Silmukka pyörii vain, jos muuttujan arvo on *0*. Jos robotin automaattinen ajo keskeytetään ja arvo muuttuu, tämä ehto ei enää täyty ja ohjelmakoodi poistuu silmukasta. (Kuva 13.)

5 TULOKSET

Työn aiheena ollut GoPiGo-robotti saatiin ohjelmoitua Python-ohjelmointikielellä kykeneväksi automaattiseen, esteitä väistävään ajamiseen sekä myös palaamaan ajettua reittiä takaisin. Robotti suorittaa näitä toimintoja kaukosäätimestä robotille lähetettyjen infrapunasignaalien ohjaamana.

Kun GoPiGo-robotissa oleva infrapunasensori vastaanottaa automaattiseen ajamistilaan yhdistetyn infrapunasignaalin, robotti alkaa liikkua itseohjautuvasti ja pysähtyy välillä esteiden edessä. Etäisyyssensori osaa ottaa servon liikuttamana etäisyysmittausarvoja esteiden molemmilta puolilta. Robotti kääntyy esteistä pois, valitsee mittausarvojen perusteella kääntymissuunnaksi joko robotin vasemman tai oikean puolen ja jatkaa sitten ajamista, toistaen ajamis- ja kääntymistoimintoja vuoron perään.

Robotin tehdessä käännöksen tai pysähtyessä tehty käännös tai ajettu matka tallennetaan niille varattuihin listoihin, joita Raspberry Pi -tietokoneessa pyörivä Python-ohjelmakoodi ylläpitää. Käännökset tallennetaan lisäksi vastakkaisina – käännökset oikealle tallennetaan käännöksinä vasemmalle, sekä toisinpäin – koska robotin palatessa reittiä takaisin keula osoittaa tulosuuntaan. Kaukosäätimellä robotille voidaan antaa automaattisen ajamisen keskeyttävä signaali milloin vain. Keskeytystilanteessa robotti ei jatka enää esteeseen asti, vaan pysähtyy heti, tallentaa jo ajetun matkan ja poistuu sitten ohjelmakoodissa itseohjautuvasta tilasta. Jos robotti on kääntymässä signaalin saatuaan, käännös tehdään loppuun ja tallennetaan, minkä jälkeen tilasta poistutaan.

Infrapunasensori vastaanottaa kahdenlaista keskeytyssignaalia. Ensimmäinen signaali siirtää robotin automaattisesta ajamistilasta suoraan käänteiseen ajamistilaan eli suorittamaan tehtyjä käännöksiä ja ajomatkoja käänteisesti. Toisen signaalin saatuaan robotti ainoastaan pysähtyy ja odottaa uutta signaalia joko aloittaakseen tallennetun ajoreitin käänteisen suorittamisen tai aloittaakseen automaattisen ajamisen uudelleen. Jälkimmäisessä tapauksessa edelliset ajomatkat ja käännökset pyyhkiytyvät ohjelmakoodista ja robotin muistista.

Käänteisessä ajamistilassa robotti käy läpi ajomatkat ja käännökset sisältäviä listoja käänteisessä järjestyksessä. Automaattisen ajamisen tilassa ajamiset ja kääntymiset seurasivat aina toisiaan, joten palaamistoiminnossa ohjelmakoodi vuorottelee myös listojen välillä. Robotti suorittaa tallennetut ajomatkat saman pituisina ja käännökset vastakkaisina, seuraten näin ajettua reittiä tulosuuntaan, kunnes kaikki liikkeet ovat suoritettu. Samoin kuin itseohjautuva tila, palaamistoiminto voidaan keskeyttää milloin vain. Keskeytyksessä vielä suorittamattomat ajot ja käännökset poistuvat muistista vain, jos robotille annetaan signaali uuden automaattisen ajamistilan aloittamiseen, joten toimintoa voidaan haluttaessa jatkaa.

Huolimatta servon teoreettisesta ja dokumentaation mukaisesta 180 asteen kääntymisasteikosta, käytännössä kääntymisalue on vain 142 astetta, mikä rajoittaa hieman etäisyyssensorin tekemiä mittauksia robotin suorittaessa kääntymistoimintoa. Ympäristön skannaus toteutettiin lopulta niin, että sensori tekee viisi etäisyysmittausta robotin kummaltakin puolelta 10 asteen välein, aloittaen niin vasemmalta kuin servo kykenee kääntymään ja ottaen viimeisen arvon niin oikealta kuin on symmetristä ensimmäiseen vasempaan arvoon nähden. Mittausvälistä ei tehty tiheämpää, koska jokainen mittaus viivyttää hieman servon ja sensorin liikkumista, ja ympäristön skannauksesta olisi kokonaisuudessaan tullut liian pitkäkestoinen. Kääntymissuunnan valitseminen toimii kuitenkin luotettavasti.

Robotin tekemissä käännöksissä huomattiin, että renkaat kääntyvät pääsääntöisesti hieman vähemmän kuin ohjelmakoodissa niille syötettyjen kääntymisasteiden verran. Tämän pohjalta ohjelmakoodin astemääriä suurennettiin niin, että säännöllinen vajaus asteissa poistui käännösten toteuttamisessa. Käännöksiin jäi kuitenkin vielä pieni epäsäännöllinen vääristymä, ja lisäksi ne tallentuvat robotin renkaille ohjelmakoodissa annettujen kääntymisasteiden perusteella, joten niiden käänteisessä suorittamisessa ja näin myös ajetun reitin seuraamisessa takaisin voi esiintyä poikkeamia.

GoPiGon liikkumisessa ilmeni myös toinen ohjelmakoodista riippumaton epätarkkuus: ajaessa suoraan robotti viettää toisinaan hieman joko vasemmalle tai oikealle. Tämä saattaa johtua fyysisestä viasta renkaassa tai renkaan ja moottorin kiinnityskohdassa. Robotin ajonopeus tuntuu vaikuttavan vaihtelevasti viettävyyden suuruuteen.

Työn aihe toimi hyvänä alustana Python-ohjelmointikieleen tutustumiseen ja ohjelmoinnissa kehittymiseen. Pythonin soveltaminen GoPiGon toimintaperiaatteisiin antoi kokemusta Raspberry Pi -tietokoneen ympärille rakennetusta robotista. Kokonaisuudessaan työ antoi pohjaa GoPiGon kaltaisen robotin jatkokehitykselle sekä Python-ohjelmointikielen käytölle yleisellä tasolla.

GoPiGo-robotin tavoitteelliset toimintaominaisuudet saatiin toteutettua suurimmilta osin. GoPiGo-robotin sensorit ja komponentit ovat toimivia. Python-ohjelmakoodilla sensoreita ja robotin liikkumista voi ohjelmoida ja hallita onnistuneesti. GoPiGo-piirilevyyn kytketty infrapunasensori osaa vastaanottaa ja eritellä erilaisia infrapunasignaaleja, ja ohjelmakoodi ja robotti osaavat suorittaa erilaisia signaaleihin liitettyjä tavoitteina olleita toimintoja. GoPiGo-robotti osaa liikkua automaattisesti, käyttäen servoon liitetyn etäisyyssensorin tekemiä mittauksia esteiden tunnistamisessa ja käännösten tekemisissä. Robotti muistaa tekemänsä liikkeet ja ajamansa reitin ja kaukosäätimestä signaalin saatuaan osaa keskeyttää itseohjautuvan toimintansa milloin vain sekä palata kulkemaansa reittiä takaisin suorittamalla tallennettuja käännöksiä ja liikkeitä käänteisessä järjestyksessä toiseen suuntaan. Robotin liikkeissä – sekä automaattisessa ajamisessa että paluutoiminnossa – on käytännössä epätarkkuuksia, mutta ohjelmakoodi on toimiva ja kaukosäätimen avulla robotin tilaa voi vaihdella onnistuneesti toiminnosta toiseen. Mahdollisessa jatkokehityksessä robotin käännöksiä ja paluureitinkulkua voisi tarkentaa gyroskoopin avulla.

6 YHTEENVETO

Tässä työssä ohjelmoitiin GoPiGo-nimistä Raspberry Pi -tietokoneen ympärille rakennettua robottiautoa. Ohjelmointikielenä toimi Python, ja Raspberry Pin kanssa käytettiin Raspbian for Robots -käyttöjärjestelmää. Työn tavoitteena oli Python-ohjelmointikielen oppiminen erityisesti GoPiGon kaltaisen robotin ja sen sensoreiden ohjelmoinnissa. Robottiauton ohjelmoinnissa tavoite oli, että robotti kykenee automaattiseen ajamiseen, jossa se tunnistaa edessään esteet sekä valitsee kääntymissuunnan. Robotin tuli myös kyetä palaamaan ajamaansa reittiä takaisin.

Robotin renkaiden toiminnassa ilmeni epätarkkuuksia, mikä johti epätarkkuuksiin myös robotin palatessa ajamaansa reittiä takaisin, joten robotin paluutoiminnon toteutuksessa ei onnistuttu täysin. Lisäksi servon ja samalla etäisyyssensorin kääntymisalue oli oletettua suppeampi.

Tavoitteisiin kuitenkin päästiin pääosin. Sekä Python-ohjelmointikieli että Raspberry Pin käyttö robotin ja sen sensoreiden ohjelmoinnissa tulivat tutuiksi. GoPiGo-robotti, sen käyttämät sensorit ja kaukosäädin sekä Raspberry Pi -tietokone ja Raspbian for Robots -käyttöjärjestelmä ovat toimiva ja ohjelmoitava kokonaisuus. Robotti tunnistaa infrapunasignaalit ja tavoitteena olleet toiminnot toimivat Python-ohjelmakoodissa. Servon ja etäisyyssensorin yhdistelmä sekä kääntymissuunnan valinta toimivat hyvin, vaikka etäisyysmittausten ulottuvuus ei olekaan optimaalinen. Ohjelmakoodi tallentaa ajetut matkat ja tehdyt käännökset, ja robotti osaa suorittaa ne käänteisesti, vaikka poikkeaakin hieman ajetusta reitistä.

LÄHTEET

1. Heath, Nick 2017. An Introduction to the Raspberry Pi 3 computer, from how to set it up, to what you can do with it. ZDNet. Saatavissa: https://www.zdnet.com/article/what-is-the-raspberry-pi-3-everything-you-need-to-know-about-the-tiny-low-cost-computer. Hakupäivä 21.1.2019.

2. Halfacree, Gareth. Raspberry Pi 3. Wired. Saatavissa: https://www.wired.co.uk/article/raspberry-pi-three-wifi-bluetooth-release-price-cost. Hakupäivä 22.1. 2019.

3. Nuttall, Ben 2016. Getting started with Raspberry Pi. Opensource. Saatavissa: https://opensource.com/article/16/12/getting-started-raspberry-pi. Hakupäivä 30.1.2019.

4. Our Software: Raspbian for Robots. 2019. Dexter Industries. Saatavissa: https://www.dexterindustries.com/raspberry-pi-robot-software. Hakupäivä 31.1.2019.

5. Gus 2019. How to Install Raspbian: A Simple Guide for Beginners. PiMyLifeUp. Saatavissa: https://pimylifeup.com/how-to-install-raspbian. Hakupäivä 13.2.2019.

6. Hardware and Port Description. 2019. Dexter Industries. Saatavissa: https://www.dexterindustries.com/GoPiGo/learning/hardware-port-description. Hakupäivä 6.2.2019.

7. GoPiGo3 Electronic Board. 2019. Dexter Industries. Saatavissa: https://www.dexterindustries.com/product/gopigo-3-electronic-board. Hakupäivä 7.2.2019.

8. Burnett, Roderick 2017. Ultrasonic Vs Infrared (IR) Sensors. MaxBotix. Saatavissa: https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm. Hakupäivä 19.2.2019.

9. Woodford, Chris 2018. Remote control. Explain that Stuff. Saatavissa: https://www.explainthatstuff.com/remotecontrol.html. Hakupäivä 19.2.2019.

10. Sturtz, John 2018. Introduction to Python 3. Real Python. Saatavissa: https://realpython.com/python-introduction. Hakupäivä 2.3.2019.

11. Sturtz, John 2018. Python Modules and Packages – An Introduction. Real Python. Saatavissa: https://realpython.com/python-modules-packages. Hakupäivä 2.3.2019.