



10. TAMMIKUUTA 2015

# ROBOTTIOHJELMOINNIN HARJOITUSTYÖ

Explorer -robotti

SARI RAUTANEN

011811699

[sari.rautanen@cs.helsinki.fi](mailto:sari.rautanen@cs.helsinki.fi)

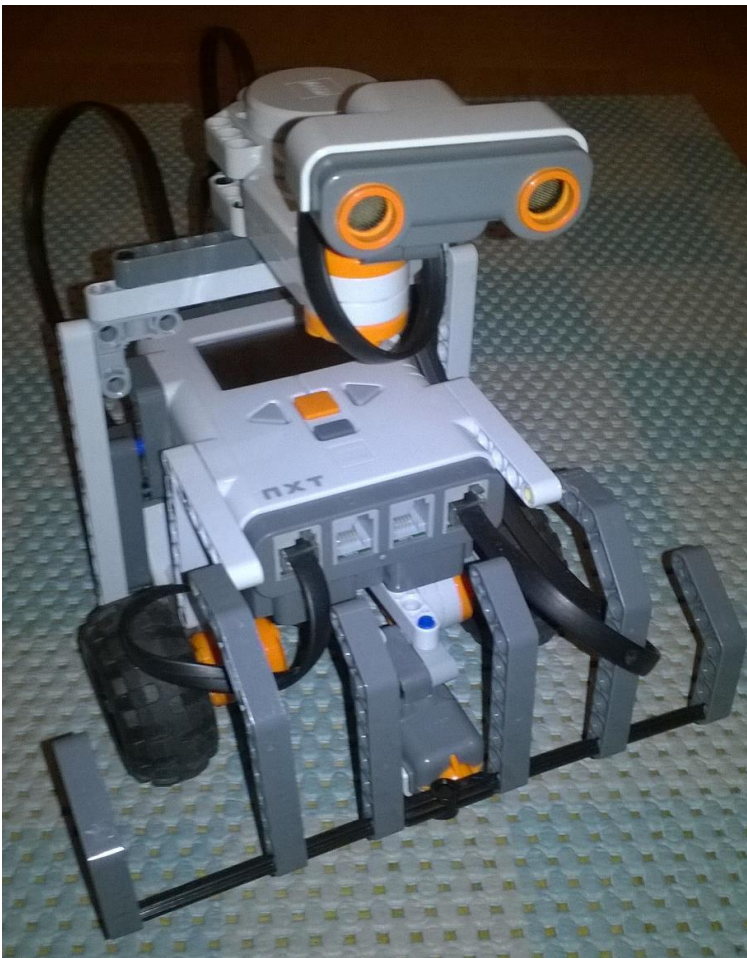


## Explorer-robotin kuvaus

Explorer on tutkijarobotti, joka esteen kohdattuaan osaa vaihtaa suuntaa. Esteen se tunnistaa joko törmäämällä siihen painesensorilla varustetulla puskurillaan, tai ultraäänisensorin havaittua korkeudellaan olevan, alle 25 cm päässä olevan esteen.

Esteen kohdattuaan ennen suunnan vaihtamista robotti havainnoi ympäristöään pyörittämällä moottoroitua ultraäänisensoria. Sensorin arvot välittyvät Bluetooth -yhteyden kautta tietokoneen näytölle, tai jos yhteydenotto tietokoneesta robottiin ei onnistu, niin robotin omalle LCD-näyttörudulle. Etäisyysarvot tulostuvat tähtirivinä/mittaus, yksi tähti vastaa tietokoneen konsoli-ikkunassa yhtä senttiä ja robotin näytöllä näytön pienuuden takia 10 senttiä.

## Robotin rakenne



Explorer on pyörillä kulkeva robotti, johon on kytketty ultraäänisensori sekä painesensori. Ajaminen tapahtuu kahdella moottorilla, jotka voivat pyöriä eri tahtiin kääntymistä varten. Kolmas pyörä on tukipyörä, joka mahdollistaa sulavan liikkumisen. Ultraäänisensori on robotin päällä osoittaen alkutilanteessa menosuuntaan. Se kääntyy havaintoja tehdessään 180 astetta vuorotellen vasemmalle tai oikealle. Painesensori sijaitsee robotin etuosassa parin sentin korkeudella ja sen edessä on leveä puskuri, jonka ansiosta painesensori havaitsee esteet koko robotin leveydeltä sekä myös hieman vinossa kulkusuuntaan sijaitsevat esteet.



Robotin yksityiskohtaiset kokoamisohjeet kuvineen löytyvät täältä:  
<http://www.nxtprograms.com/explorer/steps.html>.

## Koodin rakenne

Robotin koodi koostuu varsinaisesta robotin ohjauskoodista, tiedostossa RoboExplorer.java sekä käyttöliittymästä RobotUI.java, jonka kautta otetaan Bluetooth-yhteys robottiin ultraäänisensorin arvojen vastaanottamista varten.

Robotin ohjauskoodissa noudatetaan lejos.org:n suosittelemaa Behaviors -suunnittelumallia

(<http://www.lejos.org/nxt/nxj/tutorial/Behaviors/BehaviorProgramming.htm>). Siinä robotin toiminnot ovat omia luokkia, jotka toteuttavat Behavior -rajapinnan. Niitä hallinnoidaan Arbitrator-luokan avulla. Arbitrator-luokka määrittelee, mikä toiminto minäkin hetkenä saa suurimman prioriteetin eli pääsee suoritukseen. Behavior-rajapinnan toteuttavia luokkia on tässä toteutuksessa kolme: DriveForward, DetectWall ja Stop.

DriveForward -luokan oliolla on suurin prioriteetti. Se liikuttaa robotin kahta pyörää tasaisella nopeudella eteenpäin.

DetectWall saa suoritusoikeuden vasta kun jompikumpi sensoreista on havainnut esteen. Silloin robotin moottorit pysähtyvät, ultraäänisensori aloittaa etäisyyksien mittauksen ja välittää mittaustulokset joko Bluetoothin kautta RobotUI-luokalle tai jos yhteyttä ei ole, kirjoittaa mittaustulokset tähtirivinä omalle näytölle. Sen jälkeen se jää odottamaan minkä tahansa nappulan painallusta, jotta näytön tutkiminen on mahdollista robotin karkaamatta. Lopuksi robotti peruuttaa ja kääntyy 180 astetta. Sen jälkeen DriveForward -luokan olio saa taas suoritusvuoron ja robotti jatkaa eteenpäin kohti uusia seikkailuja.

Stop -luokan olio aktivoituu vain, jos robotin Escape-nappulaa on painettu. Silloin robotin moottorit sammutetaan ja ohjelman suoritus lopetetaan.

Ohjelma käynnistetään RoboExplorer-luokan Main-metodista, jolloin luodaan Arbitrator-olio sekä välitetään sille Behavior-rajapinnan toteuttavat oliot.

Käyttöliittymäluokka RobotUI käynnistetään omasta Main-metodistaan. Se yrittää ottaa Bluetooth-yhteyden robottiin toistaen yhteydenottopyyntöä silmukassa 100 ms välein 60 kertaa. Jos yhteydenotto onnistuu, UI vastaanottaa robotin lähettämät sensorilukemat ja piirtää tähtirivin/lukema tietokoneen konsolille. Jos Bluetooth-yhteyttä ei saatu, ohjelman suoritus lopetetaan.

## Testaus

### Testitapaus1:

Heilutin kättäni liikkuvan robotin ultraäänisensorin edessä, noin 10 cm päässä. Robotti pysähtyy, jos käsi on tarpeeksi kauan paikallaan. Nopeaa kädenheilautusta sensoria ei ehdi havaita, vaan robotti jatkaa kulkuaan eteenpäin.

### Testitapaus 2:

Kokeilin moottorin eri pyörimisnopeuksilla ultraäänisensorin mittausskykyä. Sensorin getDistances() metodi palauttaa (korkeintaan) kahdeksan viimeisintä mittaustulosta. Nopeudella 100 (astetta sekunnissa) onnistuin saamaan korkeintaan 3 tulosta/mittaus. Nopeudella 50 tulostui parhaimmillaan viisi tulosta. Eli sensoria ei havainnoi kovin nopeasti. Huomasin myös, että havainnoitavan kohteen on oltava paikallaan, jotta mittaus onnistuu. Heiluvaa kättä pyörivä sensoria ei ehdi rekisteröidä. Myöskään ohuita johtoja sensoria ei aina havaitse.

### Testitapaus 3:

Kokeilin pysäyttää robottia asettamalla puskurin eteen erilaisia esteitä. Robotti havaitsi hyvin myös vinossa kulmassa olevat esteet. Painesensorin aktivointiin riitti, että puskurin toinen reuna kosketti estettä. Puskuria matalampia esteitä kuten pyykkilinieneen tukijalkoja robotti ei luonnollisesti huomannut vaan jäi niihin kiinni. Samoin pehmeitä esteitä kuten tyynyjä robotti ei havainnut, koska painesensoriin ei kohdistunut äkillistä töytäisyä.

### Testitapaus4:

Kokeilin pysäyttää robottia eri kohdissa ohjelman suoritusta Escape-nappulaa painamalla. Pysäytys ei toiminut, jos ohjelman suoritus oli kohdassa, jossa odotetaan yhteydenottoa käyttöliittymältä. Muuten pysäytys toimi oikein.

## Testitapaus5:

Yritin ottaa käyttöliittymällä Bluetooth-yhteyden robottiin eri kohdissa robotin ohjelman suoritusta. Käyttöliittymä sai yhteyden, jos yhteydenottoyritys osui kohtaan, jossa robotti kuunteli yhteydenottopyyntöä. Jos robotti vaelsi pitkään kohtaamatta estettä, käyttöliittymän yhteydenottoluuppi ehti pyöriä loppuun ja ohjelman suoritus päättyi ennen yhteyden saamista.

## Rajoitukset ja tulevaisuus

Explorer-robotin ultraäänimittausta voisi mahdollisesti parantaa luopumalla tasaisesti pyörivästä mittausalustasta. Moottorin voisi pysäyttää mittauksen ajaksi esim. 10 asteen välein. Silloin sensori ehkä saisi enemmän mittauksia suoritettua.

Mittaustulosten esittämisgrafiikkaa voisi parantaa. Jos tuloksia per mittaus saisi enemmän, voisi mittauksia esittää esim. "plottina", jonka keskellä piste kuvaa roboa ja ympäröivän kaaren etäisyys kertoisi etäisyyden esteeseen.

Mittaustulosten kysymiseen voisi käyttää koko ajan samaa Bluetooth-yhteyttä. Nyt se yritetään luoda joka mittaukselta aina uudelleen.

Puskurin ja ultraäänisensorin väliin jää "sokea kaista". Sillä korkeudella olevia esteitä robotti ei havaitse. Puskuria voisi tehdä vielä korkeammaksi, jotta se havaitsisi myös tuon korkuiset esteet.

## Käyttöohje

Robottia kootessasi kiinnitä erityistä huomiota kaapeleiden sijoitteluun. Ultraäänisensorin kaapeli ei saa kiertyä sen "kaulan" ympärille sensorin pyöriessä ja painesensorin kaapeli ei saa haitata pyörien tai puskurin liikkumista. Kun olet koonnut robotin ohjeiden perusteella, aja build.xml kun robotti on kiinnitetty USB-piuhalla koneeseen tai Bluetooth-yhteys robotin ja tietokoneen välillä on muodostettu. Jos haluat käyttää RobotUI-käyttöliittymää, sinun on joka tapauksessa aktivoitava tietokoneesi Bluetooth -yhteys. Myös robotin Bluetooth -yhteys pitää silloin olla aktivoituna ja näkyvissä (Visible). Saat aktivoinnin tehtyä robotin omasta ohjauspaneelistä, kohdasta Bluetooth.

1. Ohjelma käynnistyy -tiedostosta RoboExplorer.
2. Tietokoneelta ajettava RobotUI käyttöliittymä käynnistyy suorittamalla tiedosto RobotUI.java.
3. Kun robotti kohtaa esteen, ultraäänisensori aloittaa mittausessionsa. Jos käyttöliittymän Bluetooth-yhteydenotto on onnistunut, mittauksien tulokset näkyvät käyttöliittymä suorittavan tietokoneen Java-konsolilla.

Jos robotti jää paikalleen sensoria pyörittävän moottorin pysähtyttyä tietokoneen käyttöliittymä ei ole saanut siihen yhteyttä. Silloin robotin näytöllä pitäisi näkyä mittauksien tulokset. Paina mitä tahansa robotin nappulaa, ja robotti jatkaa menoaan.

4. Robotin ohjelman suorituksen voi lopettaa sen Escape-nappulaa painamalla. Huomaa kuitenkin, että lopetus ei ole mahdollista, kun robotti odottaa Bluetooth-yhteydenottoa, eli juuri esteen havaitsemisen jälkeen.

Käyttöliittymän suoritus loppuu, kun robotti on pysäytetty ja käyttöliittymä on saanut yhteydenottosilmukkansa suoritettua saamatta yhteyttä robottiin, eli noin 30 s päästä robotin pysäyttämisestä.