

# Robottiohjelmoinnin harjoitustyö

## Tutkijarobotti

Henri Immonen

014289817

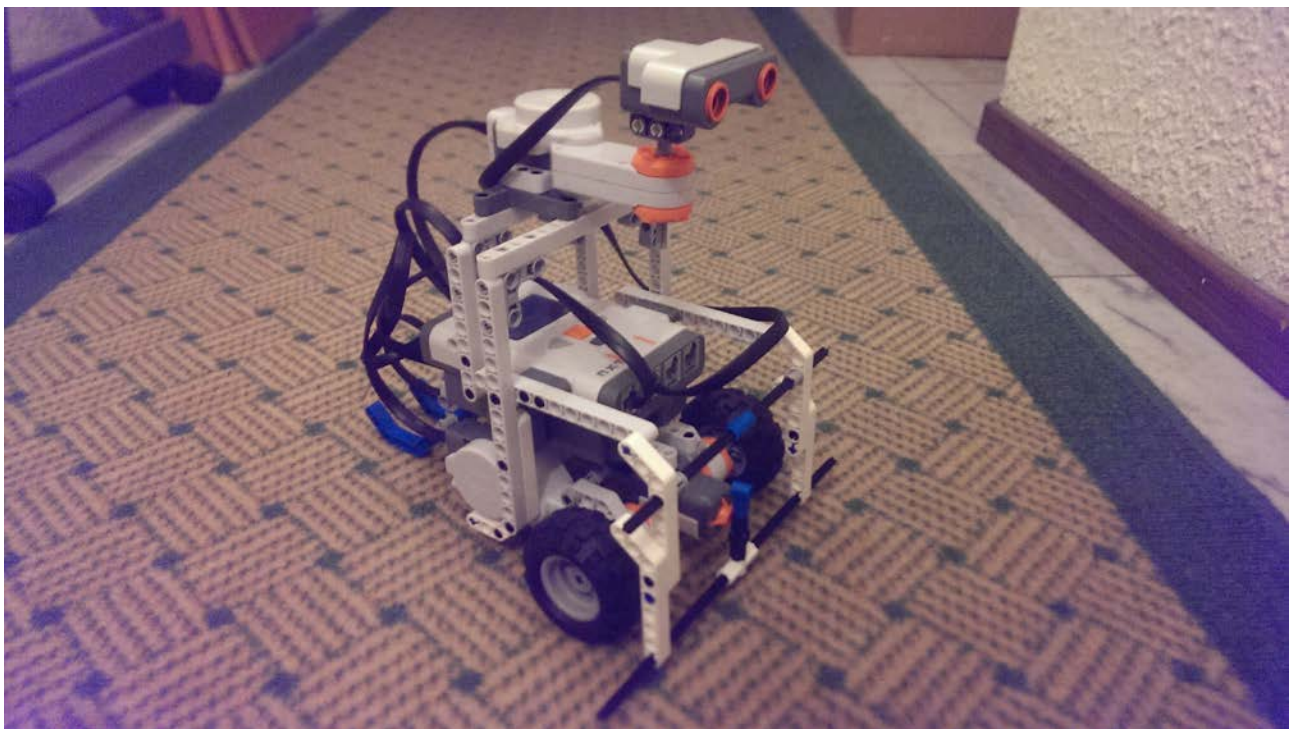
[henrimmo@cs.helsinki.fi](mailto:henrimmo@cs.helsinki.fi)

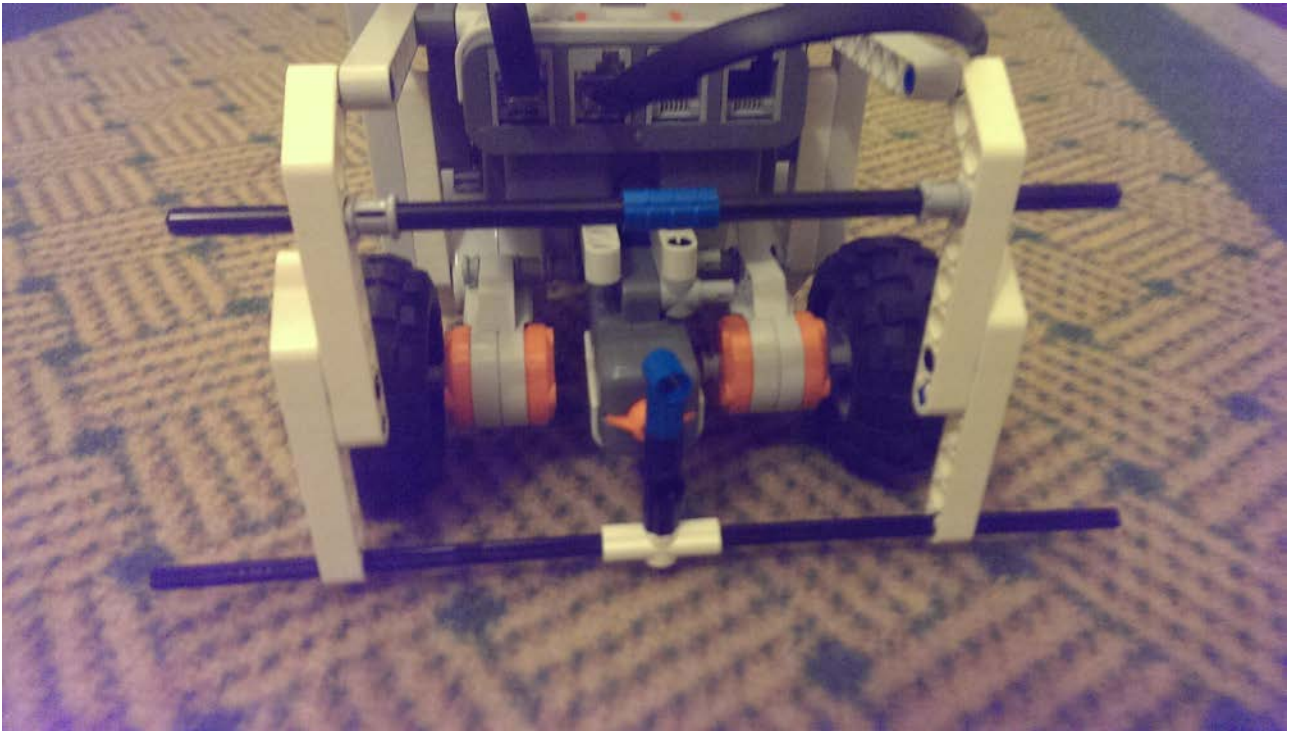
## Tutkijarobotin kuvaus

Tutkijarobotti tunnistaa edessään olevan esteen ja muuttaa kulkusuuntansa. Esteen kohdatessaan robotti peruuttaa hieman ja katsoo vasemmalle ja oikealle. Saatuaan etäisyydet kummastakin suunnasta, robotti kääntyy 90 astetta suuntaan jossa sen mielestä ei ollut esteitä ja jatkaa matkaansa. Mikäli robotti ei kykyenee määrittämään etäisyyksiä kumpaankaan suuntaan, se peruuttaa hieman ja yrittää uudelleen. Robotti jatkaa rullailuaan kunnes se sammutetaan. Robotti toimii parhaiten tasaisella, kovalla alustalla ja kun edessäolevat esteet ovat yli 20 senttiä korkeita.

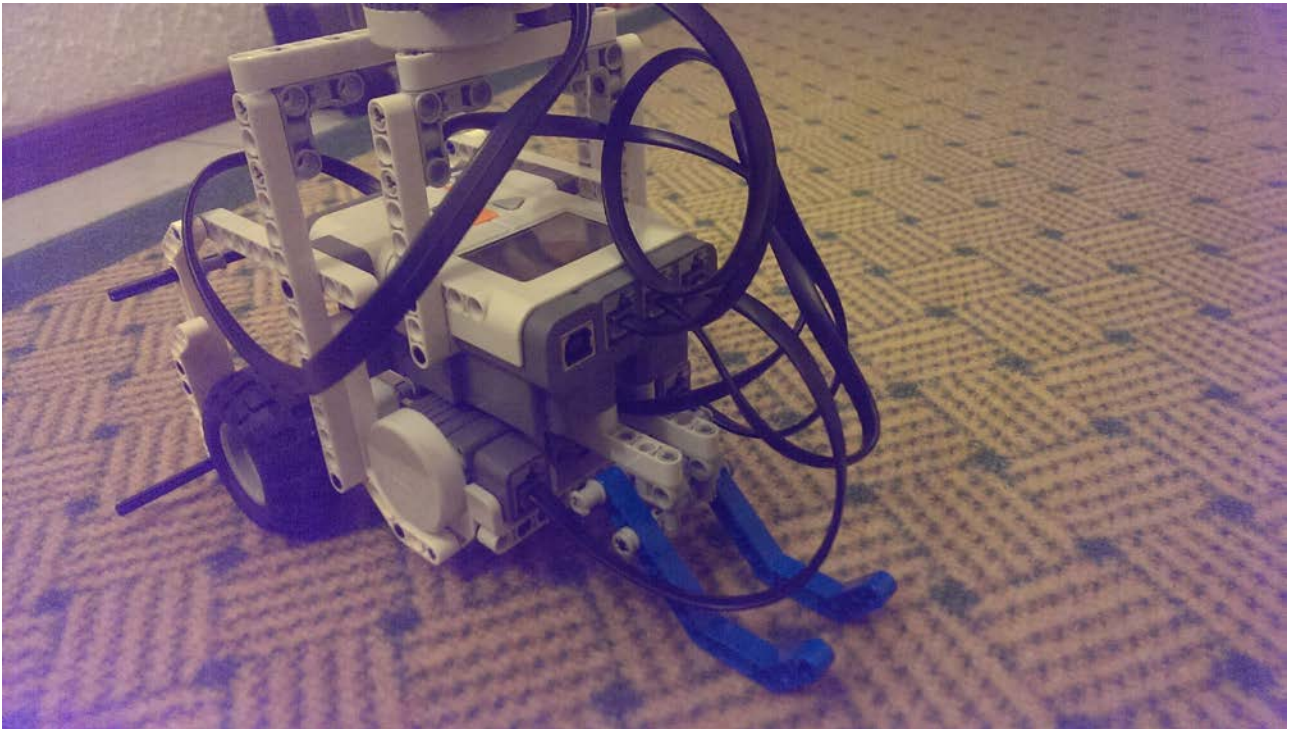
## Robotin rakenne

Robottiin on kytketty kosketussensori, ultraäänisensori ja kolme moottoria. Esteiden havaitseminen tapahtuu ultraäänisensorilla ja kosketussensorilla. Liikkumiseen robotti käyttää kahta moottoria, kolmas moottori kääntää ultraäänisensoria. Ultraäänisensori on robotin päällä, noin 22 sentin korkeudessa. Kosketussensorin edessä on rima, joka osuessaan esteeseen painaa kosketussensoria.









## Koodin rakenne

Koodi koostuu viidestä eri luokasta. DriveForward, ObstacleFound, NewDirection ja Stop toteuttavat leJOS:in Behavior-rajapinnan. Kaikilla näillä luokilla on rajapinnan määrittelemät metodit: `action()`, `suppress()` ja `takeControl()`.

Main-luokka luo nämä luokat ja antaa ne Arbitrator-oliolle joka päättää mikä luokka tulee aktiiviseksi milloinkin.

DriveForward vastaa robotin liikkumisesta eteenpäin. Se on aktiivinen kun robotti ei tarvitse uutta suuntaa.

ObstacleFound tulee aktiiviseksi kun robotti havaitsee ultraäänisensorilla esteen alle 30 sentin päässä edessään tai kun kosketussensoria painetaan. Tällöin robotti peruuttaa 5 senttiä ja luokan boolean `needDirection` arvoksi tulee `true`.

Tällöin NewDirection luokka vaihtuu aktiiviseksi. `Action()`-metodi kutsuu metodia `execute()`, joka kääntää ultraäänisensoria ja mittaa etäisyydet vasemmalta ja oikealta. Tämän jälkeen `decideDirection()` päättää, kääntyykö robotti vasemmalle vai oikealle. Lopuksi kutsutaan ObstacleFound luokan `noNeed()` metodia, joka asettaa `needDirection` arvoksi `false`. Tällöin robotti jatkaa kulkuaan eteenpäin.

Stop-luokka sammuttaa robotin kun exit-nappia painetaan.

## Tesataus

### Tapaus 1

Asetin robotin lähelle huoneen nurkkaa, noin 1 metrin ja 50 sentin päähän seinistä. Robotti lähti liikkeelle ja pysähtyi noin 30 senttiä ennen seinää. Robotti havaitsi että sen oikealla puolella on seinä ja vasemmalla ei mitään. Robotti kääntyi vasemmalle ja jatkoi matkaansa. Testi onnistui!

### Tapaus 2

Halusin testata miten robotti toimii matalan esteen kohdatessaan. Ensiksi asetin robotin ajamaan päin noin 8 senttiä korkeaa esinettä. Ultraäänisensori ei havainnut esinettä ja robotti pysähtyi vasta kun kosketussensori osui esineeseen. Se jatkoi matkaansa vasemmalle ja kohtasi maton, joka myös pysäytti robotin. Robotti jäi eräänlaiseen looppiin jossa se koitti jatkaa matkaansa joko kohti mattoa tai esinettä.

Testi epäonnistui, koska ultraäänisensori ei uutta suuntaa valitessaan havainnut matalia esteitä.

### Tapaus 3

Testasin miten robotti toimii erilaisilla pinnoilla.

Laatta- ja parkettilattialla robotti toimii juuri niinkuin pitääkin, kulkien suorassa linjassa ja tehden 90 asteen käännöksiä.

Matolla robotin käännökset olivat sattumanvaraisempia ja korkeakarvaisella matolla robotin kosketussensori aktivoitui jatkuvasti.

Johtopäätöksenä, että robotti tarvitsee tasaisen , kovan alustan toimiakseen oikein.

## Rajoitukset ja tulevaisuus

Robotin pahin puute on matalien esteiden tunnistaminen. Pahimmassa tapauksessa robotti jää loputtomaan looppiin kun se tulkitsee tien olevan vapaa.

Tulevaisuudessa robotti voisi kyetä vaikkapa kompassisensoria käyttäen pitämään tietyn kurssin tai kartoittamaan reittinsä ja piirtämään sen LCD-näytölle. Ajatukseni oli toteuttaa tämä toiminnallisuus, mutta loma- ja työkiireitten takia en sitä ehtinyt tekemään.

## Käyttöohjeet

1. Aseta robotti lattialle. Käynnistä robotti ajamalla Main.nxj.
2. Robotti lähtee liikkeelle ja väistelee esteitä.
3. Kun haluat pysäyttää robotin, paina NXT Brickin exit-painiketta.