

Robottiohjelmoinnin harjoitustyö

Taskuparkkeeraava-auto
Sulevi Sihvola
sulevi@cs.helsinki.fi

Auton kuvaus

Taskuparkkeeraava auto on Lego Mindstorm osista koottu auto, joka osaa parkkeerata ns. taskuun. Liikkuessaan suoraan eteenpäin auto tarkkaillee etäisyyttä oikealle puolelleen ja löytäessään mahdollisen parkkipaikan, se mittaa parkkipaikan koon. Jos parkkipaikka on tarpeeksi iso, niin auto parkkeeraa itsensä taskuun, jos parkkipaikka ei ollut tarpeeksi iso, niin auto jatkaa matkaa eteenpäin etsien seuraavaa parkkipaikkaa. Parkkeerattuaan auto odottaa napin painallusta, jonka jälkeen auto lähtee parkista ja aloittaa taas mahdollisen parkkipaikan metsästämissä.

Robotin rakenne

Autossa on neljä rengasta, edessä olevat renkaat ovat vastuussa kääntymisestä ja takana olevat etenemisestä. Autossa on kaksi moottoria asennettu taka-akselille niin, että kumpikin moottori on vastuussa vain yhden renkaan pyörittämisestä, eli takana on periaatteessa differentiaali rakenne, jonka ansiosta auto ei vastusta kääntymistä. Moottorit ovat kytketty renkasiin hammasrattaita hyväksi käyttäen. Vaihtamalla näiden hammasrattaiden kokoa on mahdollista saada eri hyötysuhteita irti moottoreista. Yksi moottori on vastuussa eturenkaiden kääntämisestä vasemmalle ja oikealle. Tämä moottori suoraan kiinni L-muotoisessa Lego palikassa, jota kääntämällä eturenkaat kääntyvät mukana. Autossa on ultraäänisensori, joka mittaa etäisyyttä autosta oikealle. Tämä sensorin on asennettu oikean etupyörän taakse noin 8cm korkeuteen, jonka pitäisi olla optimikorkeus tälle sensorityypille. Autossa on myöskin kosketussensori, joka on asennettu auton taakse huomioimaan mahdollinen törmäys peruuttaessa.

Auto on rakenteeltaan erittäin tukeva. Tämä on saavutettu rakentamalla auton perusrunko lähtemään takana olevien moottorien yläpuolelta, ja pitkittäisrakennetta vahvistaa takamoottoreiden alapuolelta menevä tuki järjestelmä. Takamoottorit ovat tuettu toisiinsa yhdellä poikittaisella palalla, ja tämä pala on myös osa koko auton alla olevaa tuentaa. Auton suunnittelussa on otettu huomioon mahdollisuus, että auto voisi mennä kovempiakin nopeuksia, riippuen minkä kokoiset hammasrattaat ovat käytössä autoa liikuttavien moottoreiden ja pyörien välillä.

Auton mitat ovat (kaikki noin liki arvoja):

Pituus 28cm

Leveys 15cm

Korkeus 11.5cm

Paino 0.9kg.



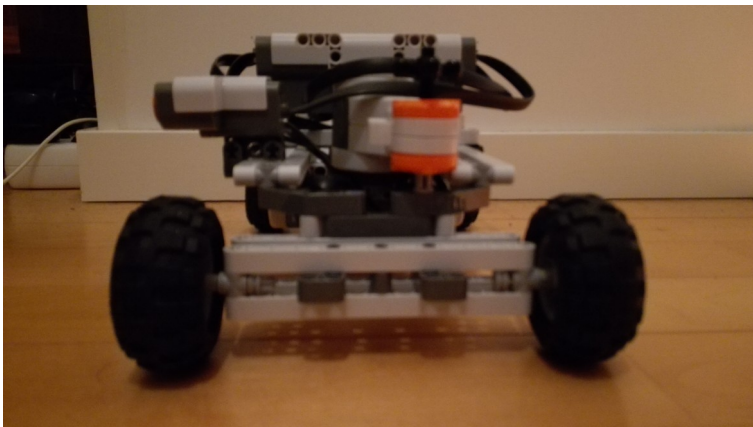
Oikealta



Vasemmalta



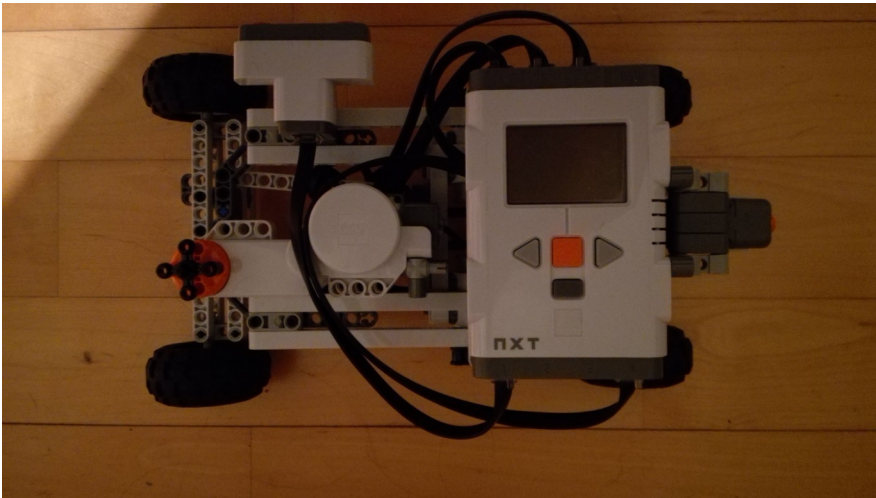
Edestä



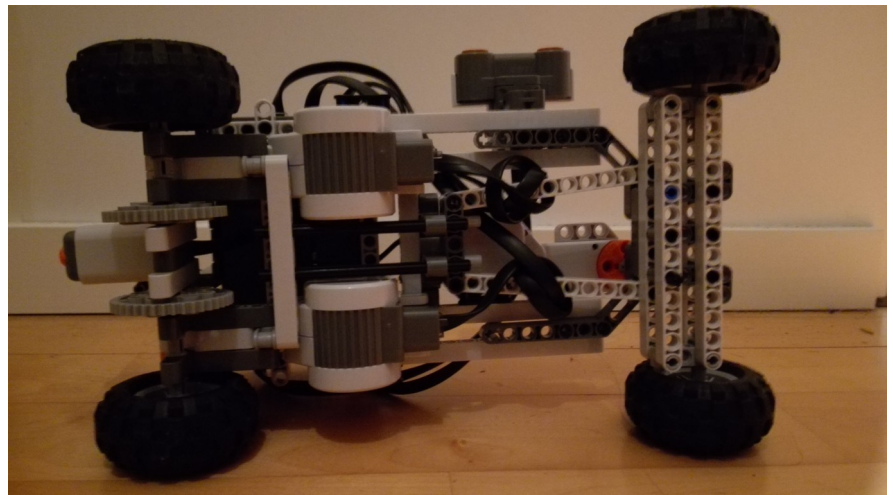
Takaa



Ylhäältä



Alhaalta



Koodin rakenne

Taskuparkkeeraava-auton koodi rakentuu seitsemästä (7) luokasta ja yhdestä wav-äänitiedostosta. Auton pääluokka on Auto, jossa luodaan auton käyttömallit (Behavior) ja annetaan ne Arbitratorille (nimetty ”parkkeeraus”) sekä käynnistetään Arbitrator. Auto-luokassa myöskin asennetaan Esc-nappulaan kuuntelija, jonka tehtävänä on toteuttaa hätäpysäytys nappulan vapautuessa, sekä luodaan OmaPilootti ”pilootti” ja UltrasonicSensor ”sonar”, joita annetaan tarpeen mukaan Behavior rajapintaa toteuttaville luokille.

OmaPilootti-luokka on vastuussa moottorien pyörittämisestä eli auton liikkumisesta eri suuntiin. Tämä luokka ei toteuta mitään rajapintaa eikä peri mitään luokkaa. OmaPiloottin luomisen yhteydessä luokka itse kalibroi oman ohjauksensa.

Luokat EtsiParkkipaikka, ScannaaParkkipaikka, PakitaParkkiin, LahdeParkista ja Tormays toteuttavat kaikki rajapinnan Behavior. Nämä luokat annetaan edellä mainitussa järjestyksessä (EtsiParkkiin ollen alin pinossa ja Tormays ollen ylin pinossa) parkkeeraus-Arbitratorille.

EtsiParkkipaikka tehtävä on ajaa autoa eteenpäin, kunnes joku toinen luokka haluaa ottaa kontrollin. Tällä luokalla on boolean arvo ”parkkipaikka”, jonka ollessa epätosi tämä luokka haluaa ottaa kontrollin.

ScannaaParkkipaikka ottaa kontrollin, kun ultraäänisensori palauttaa lukeman joka on isompi kuin 20 eikä ole 255 (eli virhelukema). Tämä luokka mittaa huomaamansa parkkipaikan ja jos se on tarpeeksi iso (noin 40 cm pitkä), niin ScannaaParkkipaikka-luokka muuttaa EtsiParkkipaikka-luokan boolean arvon ”parkkipaikka” todeksi. Tällöin lähtee myöskin auton etäisyys seinästä PakitaParkkiin oliolle, ja ScannaaParkkipaikka siirtää autoa tarvittavan määrän eteenpäin parkkipaikkaan peruuttamista varten. Jos kävi niin, että parkkipaikka ei ollut tarpeeksi iso, palauttaa parkkeeraus-Arbitrator kontrollin autosta takaisin EtsiParkkipaikka luokalle.

PakitaParkkiin-luokka ottaa kontrollin autosta, kun EtsiParkkipaikka-luokan boolean arvo ”parkkipaikka” vaihtuu epätodesta todeksi. Pakitaparkkiin-luokka peruuttaa auton, joko läheltä seinää suoraan parkkiin (reitti on kovakoodattuna -_-;) tai jos auto on kauempana seinästä auto lähestyy viimeisimmän etäisyyden pohjalta lasketun matkan verran parkkipaikkaa (mutta kaarrokset ovat tässäkin kovakoodatut). Suoritettuaan peruutuksen asettaa tämä luokka luokan LahdeParkista boolean ”parkissa” arvon todeksi.

LahdeParkista-luokka ottaa kontrollin autosta, kun luokan boolean ”parkissa” arvo vaihtuu todeksi. Tämän luokan tehtävä on lähteä parkista pois (kovakoodattu reitti), kunhan käyttäjä painaa nappulaa. Lähdettyään parkista LahdeParkista-luokka vaihtaa oman boolean ”parkissa” arvoksi epätosi ja myöskin vaihtaa EtsiParkkipaikka-luokan boolean ”parkkipaikka” arvon epätodeksi.

Tormays-luokka ottaa kontrollin aina, kun auton takana oleva kosketussensori on painettuna, tällöin luokka pysäyttää moottorit ja soittaa wav-tiedosto (autosta kuuluu ”PERKELE!”), jonka jälkeen auto odottaa, että jotakin nappulaa painettaisiin. Tällä hetkellä Tormays-luokka ei ota kontrollia, kun PakitaParkkiin tai LahdeParkista suorittaa koodia eli liikuttaa autoa, tämä johtuu luokkien kovakoodatuista reiteistä.

Testaus

Testitapaus 1:

Parkkipaikka on noin 24 cm leveä ja noin 40 cm pitkä ja auto on noin 4-5 cm päässä seinästä ennen parkkeerausta. Alue on rajattu joko pahvista tai muovista tehdyillä kappaleilla.

Auto lähtee liikkeelle tehtyään ohjauksen kalibroinnin. Havaitsee mahdollisen parkkipaikan ja alkaa mitata sitä. Auton eteneminen ei ole aivan suoraa. Mitattuaan parkkipaikan auto etenee noin 15cm ja aloittaa peruuttamaan parkkipaikkaan, auto kaartaa hyvin läheltä etureunaa mutta ei osu siihen. Parkkeerattuaan auto odottaa nappulan painallusta. Nappulan painalluksen jälkeen auto kuroaa pois parkista ja jatkaa seinämän skannaamista etsien seuraavaa paikkaa.

Testi ajettiin kolme (3) kertaa saaden suurin piirtein samanlaisen tuloksen aina, auton suunnassa ja kääntymisessä oli pieniä heittoja johtuen auton puskemisesta, mitä tapahtui käännösten aikana. Auto toimi toivotulla tavalla.

Testitapaus 2:

Parkkipaikka on noin 24 cm leveä ja noin 40 cm pitkä, ja auto on noin 13-15 cm päässä seinästä ennen parkkeerausta. Alue on rajattu joko pahvista tai muovista tehdyillä kappaleilla.

Testitapaus 2 etenee samalla tavalla kuin testitapaus 1, ainoana erona on, että auto on kauempana parkkipaikasta sivusuunnassa. Tällöin auto laskee etäisyyden seinästä ja laskee kuinka paljon tarvitsee peruuttaa ensimmäisen kaarroksen jälkeen. Tämän jälkeen auto menee sig-sag-maisesti kunnolla ”ruutuun”.

Testi ajettiin kolme (3) kertaa, millään kertaa auto ei ollut lähelläkään seinää ja aina päätyi erinomaisesti parkkipaikan sisälle. Samanlaisia havaintoja kuin testitapaus 1:ssä oli auton puskemisesta. Auto toimi toivotulla tavalla.

Testitapaus 3:

Parkkipaikka on noin 24 cm leveä ja noin 40 cm pitkä. Alue on rajattu käyttäen paperipusseja.

Auto lähtee liikkeelle mutta töksähtelee. Ilmeisesti ultraäänisensori ei pysty saamaan tarkkaa lukemaa paperipusseista ja tästä syystä auto vaihtelee etsiparkkipaikka ja scannaaparkkipaikkaa toimintojen välillä.

Testi ajettiin kaksi (2) kertaa saaden samanlainen tulos molemmilla kerroilla. Auto ei toiminut toivotulla tavalla.

Testitapaus 4:

Parkkipaikka on noin 24 cm leveä ja noin 30 cm pitkä. Alue on rajattu joko pahvista tai muovista tehdyillä kappaleilla.

Auto lähtee liikkeelle, aloittaa skannaamaan mahdollista parkkipaikkaa, ohitettuaan parkkipaikan auto palaa etsimään seuraavaa mahdollista parkkipaikkaa.

Testi ajettiin kolme (3) kertaa. Auto toimi toivotulla tavalla.

Testitapaus 5:

Parkkipaikka on noin 24 cm leveä ja noin 30 cm pitkä, ja heti perään on toinen parkkipaikka joka on noin 24 cm leveä ja noin 40 cm pitkä. Alue on rajattu joko pahvista tai muovista tehdyillä kappaleilla.

Auto lähtee liikkeelle, ohittaa ensimmäisen mahdollisen parkkipaikan, ja parkkeeraa seuraavaan paikkaan.

Testi ajettiin kerran (1).

Auto toimi toivotulla tavalla.

Testitapaus 6:

Kaksi parkkipaikkaa (noin 24 cm leveä ja noin 40 cm pitkä) peräkkäin. Alue on rajattu joko pahvista tai muovista tehdyillä kappaleilla.

Auto lähtee liikkeelle, käy parkissa ensimmäisessä parkkipaikassa sekä toisessa parkkipaikassa.

Testi ajettiin kerran (1).

Auto toimi toivotulla tavalla.

Testitapaus 7:

Auton takana olevan kosketus sensorin toiminnan testausta:

Tapahtumat kun auton eri toiminnot olivat menossa:

EtsiParkkipaikka:

Auto pysähtyi ja kuului ”perkele!”. Toimi toivotulla tavalla.

ScannaaParkkipaikka:

Auto pysähtyi ja kuului ”perkele!”. Toimi toivotulla tavalla.

PakitaParkkiin:

Auto jatkoi PakitaParkkiin toimintaa reagoimatta millään tavalla. Ei toimi toivotulla tavalla.

LahdeParkista:

Auto jatkoi LahdeParkista toimintaa reagoimatta millään tavalla. Ei toimi toivotulla tavalla.

Testi ajettiin jokaiselle tilanteelle kerran (1). Kovakoodatuista luokista PakitaParkkiin ja LahdeParkista johtuen parkkiin peruuttaessa tai parkista lähtiessä auto ei reagoinut kosketussensorin painallukseen.

Auto ei toiminut joka tilanteessa toivotulla tavalla

Rajoitukset ja tulevaisuus

Auto tunnistaa vain kiinteät esteet ja joskus nekin jäävät ultraäänisensorilta huomaamatta, esimerkiksi paperipussit.

Tällä hetkellä auto puskee käännoksissä, johtuen hammasrattaista jotka ovat tällä hetkellä asennettu takamoottorien ja takarenkaiden väliin. Hammasrattaiden kokoa vaihtamalla on mahdollista saavuttaa ns. differentiaaliakselin hyöty, eli tällöin auto ei vastustaisi käännöstä. Nykyisellä koko yhdistelmällä (isoin hammaspyörä mitä löytyy kiinni renkaassa ja pienin kiinni moottorissa) rengasta ei pysty pyörittämään vapaasti, eli rengas pystyy pyörimään vain kun moottoripyörä.

Ohjaus ei ole tarkka keskittämään, tai keskialueella on ”löysää”. Tämä johtuu suorasta moottori/ohjaus ratkaisusta. Ohjauksen ratkaiseminen jollain toisella tavalla voisi tuoda lisää tarkkuutta.

Auto tällä hetkellä parkkeeraa ja lähtee parkista valmiiksi määritellyjä reittejä pitkin. Autoon voisi lisätä etäisyysensorit eteen ja taakse, tai laittaa nykyisen sensorin kääntymään, jolloin ei tarvitsisi olla ns. kovakoodattuja reittejä käytössä. Tällöin saisi takana olevan kosketus sensorin softan toimimaan halutulla tavalla.

Autossa ei ole erikseen määriteltynä parkkipaikan kokoja, paitsi että auto ohittaa liian pienet parkkipaikat. Tällä hetkellä auto ei välttämättä parkkeeraa järkevästi, jos parkkipaikka on isompi, vaan mahdollisesti vie useamman autonpaikan (ja tosi elämässä tämä olisi todella ärsyttävää). Tästäkin ongelmasta pääsisi eroon käyttämällä sensoreita mittaamaan matkaa auton edestä ja takaa, tai paremmalla koodilla ;-).

Yhtenä ajatuksena autoa rakentaessa oli mahdollisuus android-pohjaisesta kauko-ohjauksesta autolle...

Käyttöohje

Auton kasaamisen jälkeen voi olla, että joudut muuttamaan jakajaa mikä on OmaPilootti-luokassa metodissa matkattuMatka(). Tällä hetkellä jakaja laskee matkan mitä auto on tehnyt oikein, jos takamoottorien ja renkaiden välissä on isoin hammasratas renkaassa ja pienin hammasratas moottorissa. Eli jos käytät eri hammasrattaita niin joudut vähän säätämään ja mittailemaan matkattua matkaa, tai jopa vaihtamaan jakajan kertoimeksi (tämä silloin jos moottorissa on isompi ratas kuin renkaassa)!?!)

Jos ja kun olet edellisen saanut tehtyä, tai kasannut auton suoraan kuvien mukaan, niin rakentele kiinteistä kappaleista mahdollinen parkkipaikka (noin 24 cm leveä ja noin 40 cm pitkä). Aja build.xml tai muuten vetäise tiedosto nxt palikkaan (usbllä tai bluethootilla). Laita auto seinän viereen niin, että mahdollinen parkkipaikka tulee vastaan ja aja Auto.nxj nxt:ltä.

Katsele miten ihmeellisesti auto joko huomaa parkkipaikan tarpeeksi isoksi tai liian pieneksi ja sitten parkkeeraa. Auto odottelee jonkun nappulan painallusta että lähtisi parkista, ja sitten jatkaa seuraavan parkkipaikan metsästystä. Kun olet kyllästynyt missä tahansa vaiheessa auton etenemiseen tai peruuttamiseen niin paina esc-nappulaa (sen oranssin alapuolella, tumman harmaa), niin auto pysähtyy.