

Lappeenrannan teknillinen yliopisto  
Tuotantotalouden tiedekunta  
Tietotekniikan koulutusohjelma

Kandidaatintyö

**Jani Vehviläinen**

## **AVR-POHJAINEN MOBIILIROBOTTI**

Työn tarkastaja:      Professori Lasse Lensu

Työn ohjaaja:        Professori Lasse Lensu

# **TIIVISTELMÄ**

Lappeenrannan teknillinen yliopisto  
Tuotantotalouden tiedekunta  
Tietotekniikan koulutusohjelma

Jani Vehviläinen

## **AVR-POHJAINEN MOBIILIROBOTTI**

Kandidaatintyö

2014

38 sivua, 16 kuvaa, 1 liite

Työn tarkastaja: Professori Lasse Lensu

Hakusanat: AVR, mobiilirobotti, ultraääni-sensori, servomoottori, navigaatio

Kandidaatintyön aineena oli rakentaa autonomisesti pyörillä liikkuva, esteitä väistelevä mobiilirobotti käyttäen Arduino-kehitysympäristöä prototyypin valmistamiseen. Rakensin robotin alusta alkaen itse eli työhön sisältyi robotin mekaniikan kokoaminen, elektroniikan suunnittelu ja rakentaminen sekä toimintaälyn ohjelmointi mikroprosessorille eteenpäin kulkemiseen ja esteiden väistämiseen. Arduinon ytimenä on Atmelin AVR-sarjaan kuuluva ATmega328-mikroprosessori. Robotin ympäristön havainnointi tapahtuu käyttämällä ultraäänisensoria. Robotti oli mielenkiintoinen projekti toteuttaa ja toimi kokonaisuutena katsottuna odotetulla tavalla.

## **ABSTRACT**

Lappeenranta University of Technology  
School of Industrial Engineering and Management  
Degree Program in Computer Science

Jani Vehviläinen

### **AVR BASED MOBILE ROBOT**

Bachelor's Thesis

38 pages, 16 figures, 1 appendices

Examiner: Professor Lasse Lensu

Keywords: AVR, mobile robot, ultrasound probe, servomotor, navigation

The topic of my Bachelor's thesis was to build obstacles avoiding, mobile robot using Arduino developing environment and board. I built the robot from the scratch which included constructing the mechanics, designing and assembling the electronic parts and coding the operating logic for the microprocessor. The core of the Arduino is Atmel ATmega328 microprocessor. The robot uses ultrasound range-detector to detect any obstacles in front of it. The Robot project was quite interesting and to sum up, it worked as designed.

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO</b>	<b>4</b>
1.1	TAUSTA	4
1.2	TAVOITTEET JA RAJAUKSET	4
1.3	TYÖN RAKENNE	5
<b>2</b>	<b>OLENNAISET KÄSITTEET</b>	<b>6</b>
2.1	KIRJALLISUUSKATSAUS	6
2.2	ARDUINO	7
2.2.1	<i>Historia</i>	7
2.2.2	<i>Ominaisuudet</i>	7
2.2.3	<i>DUEMILANOVE-alusta</i>	8
2.3	AVR	10
2.4	SERVOMOOTTORI	10
2.5	ULTRAÄÄNISENSORI	11
<b>3</b>	<b>RATKAISUMENETELMÄT</b>	<b>11</b>
3.1	ROBOTIN MEKANIikka	12
3.1.1	<i>Tasavirtamoottorit (DC)</i>	12
3.1.2	<i>Askelmoottorit</i>	13
3.1.3	<i>Servomoottorit</i>	14
3.2	HAVAINNOINTI	15
3.2.1	<i>Mikrokytkimet</i>	16
3.2.2	<i>Infrapuna-sensori</i>	17
3.2.3	<i>Ultraäänisensori</i>	18
3.3	NAVIGAATIOALGORITMI	19
<b>4</b>	<b>KÄYTÄNNÖN TOTEUTUS</b>	<b>20</b>
4.1	MEKAANINEN JA SÄHKÖINEN KASAUS	20
4.2	TOIMINTAÄLYN OHJELMOINTI	24
4.2.1	<i>Ultraäänisensorin toiminta</i>	25

4.2.2	<i>Liikkuminen</i> .....	26
4.2.3	<i>Testaus</i> .....	26
<b>5</b>	<b>POHDINTA JA TULEVAISUUS</b> .....	<b>27</b>
<b>6</b>	<b>YHTEENVETO</b> .....	<b>29</b>
	<b>LÄHTEET</b> .....	<b>31</b>
	<b>LIITTEET</b>	

## **SYMBOLI- JA LYHENNELUETTELO**

DC	Direct Current
EEPROM	Electrically Erasable Programmable Read-Only Memory
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MHz	Megahertz
PWM	Pulse Width Modulation
RISC	Reduced Instruction Set Computer
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

# 1 JOHDANTO

Roboteiksi kutsutaan mekaanisia laitteita, jotka osaavat toimia jollain tavalla itsenäisesti. Mobiilirobotiksi kutsutaan robottia, joka kykenee liikkumaan pyörillä, telaketjuilla tai esimerkiksi jaloilla eteenpäin. [1] Työssäni keskityin rakentamaan pyörien avulla liikkuvaa robottia.

## 1.1 Tausta

Mikroprosessorit ilmestyivät kuluttajamarkkinoille 1980-luvun vaihteen jälkeen. Yhteen koteloon paketoitua ”mikrosiruta” olivat suuri edistysaskel entiseen verrattuna. Harrastajapiirien suosikiksi muodostui 90-luvun alussa PIC-mikroprosessorit ja myöhemmin Atmelin useasti ohjelmoitavat AVR-mikroprosessorit. [2] Vuosikymmenen loppupuolella harrastajapiireissä mielenkiinto alkoi suuntautumaan Arduino kehitysalustaan, joka tarjoaa monipuoliset mahdollisuudet rakenteluun useiden vapaasti ohjelmoitavien IO-nastojen sekä valmiiden servomoottorikirjastojen ansiosta. Niinpä päätin toteuttaa robotin rakennuksen Arduino-alustalla kandidaatintyönäni.

## 1.2 Tavoitteet ja rajaukset

Työn tavoitteena oli rakentaa autonomisesti kulkeva ja esteitä väistelevä mobiilirobotti. Rakensin robotin alusta alkaen itse eli työhön sisältyi robotin mekaniikan kokoaminen, elektroniikan suunnittelu ja rakentaminen sekä toimintaälyn ohjelmointi mikroprosessorille eteenpäin kulkemiseen ja esteiden väistämiseen.

Valmis robotti kulkee suoraviivaisesti eteenpäin, tehden mittauksia ultraäänellä edessä oleviin kohteisiin. Mittauksen jälkeen tehdään uusi päätös suunnasta ja tarvittava käännösliike. Tämän jälkeen matka jatkuu jälleen eteenpäin.

Mobiilirobotin ytimenä käytin Arduino Duemilanove nimistä mikroprosessorin

kehitysalustaa sen tarjoamien ominaisuuksien vuoksi. Valintaa puoltaa myös se, että minulla on ennestään kyseinen kehitysalusta omistuksessani.

Servomootoreiksi mobiilirobotin liikutteluun olen valinnut yleisesti harrastajien suosiman Futaban S3001-moottorin. Servomootorit täytyi muokata, jotta ne pyörivät vapaasti etenemiseen tarvittavat 360 astetta ympäri. Ultraäänisensorin liikutteluun käytin Tower Pro SG90 micro-servomoottoria.

Esteiden havainnointi tapahtuu käyttämällä ultraäänisensoria, jonka avulla etäisyys saadaan mitattua riittäväällä tarkkuudella.

Robotin ohjaamiseen tarvittavan ohjelmakoodin tuottamiseen käytin Arduinon omaa kehitysympäristöä Macbook tietokoneella, jossa käyttöjärjestelmänä on Mac OS X.

### **1.3 Työn rakenne**

Toisessa luvussa käydään lyhyt kirjallisuuskatsaus aiheeseen, selvitetään mitä aiheeseen liittyvät käsitteet tarkoittavat sekä kerron tarkemmin Arduinosta sekä työssä käytettävästä Duemilanove kehitysalustasta.

Kolmannessa luvussa esittelen ratkaisumenetelmät työn ongelmiin ja käyn läpi muutamia vaihtoehtoja valituille ratkaisuille.

Neljännessä luvussa käydään läpi käytännön toteutus eli robotin rakentaminen.

Viidennessä luvussa pohdin projektia ja esitän näkymät sen tulevaisuudesta.

Kuudennessa luvussa on kootusti yhteenveto työstä.



## 2 OLENNAISET KÄSITTEET

Tässä kappaleessa esittelen työn kannalta olennaiset käsitteet sekä kirjallisuuskatsauksen aiheeseen.

### 2.1 Kirjallisuuskatsaus

Robottiikan aiheesta on kirjoitettu aikojen saatossa lukemattomat määrät kirjoja, koskien aiheen monia eri ala-aiheita. Tässä katsauksessa käsittelen pintapuolisesti muutamaa työssäni käyttämää kirjaa.

Principles of Robot Motion – Theory, Algorithms, and Implementations on teoreettinen kirja. Se esittelee hyvin kattavasti erilaisia algoritmeja robotin navigaatioon, asennon ja sijainnin tunnistamiseen sekä ympäristön havainnointiin. [3]

Introduction to Autonomous Mobile Robots -kirja esittelee suunnistusalgoritmeja hieman tivistetympin, mutta kirjassa on myös esitelty kattavasti erilaisten sensorien toimintaa. Käyttämästäni ultraääni-sensorista on kattava toiminnankuvaus sekä teoriaa, kuinka sitä pystyy hyödyntämään tehokkaasti. [4]

Mobile Robots – Inspiration to Implementation on enemmän harrastajalähtöinen kirja. Siinä käydään kaikki robotin rakentamisen perusasiat läpi sekä teorian, että käytännön kannalta. Kirjassa on selkeitä esimerkkejä ja ohjeita teorioiden soveltamiseen käytännössä. [1]

Sulautetut – Opi rakentamaan robotteja ja sulautettuja on varsin uusi (2009), ensimmäinen suomalainen kirja joka käsittelee Arduino-alustaa. Kirjassa on selkein kuvin esitettyjä ohjeita erilaisten laitteiden rakentamiseen. Kirjassa on mm. ohjeet servomoottoreiden muokkaamiseen, joita käytän työssäni. [5]

## **2.2 Arduino**

Arduino on open source - pohjainen kehitysympäristö Atmelin AVR-sarjan 8-bittiselle mikroprosessorille sekä siihen liittyvä fyysinen kehitysalusta. Kehitysalustalla on mikroprosessorin lisäksi sen tarvitsema elektroniikka, usb-kommunikaatioon tarvittava piiri sekä virransyöttö. [6]

Arduinon tarkoituksena on tarjota opiskelijoille, tutkijoille ja harrastajille edullinen ja helppo tapa päästä käsiksi mikrokontrollerimaailmaan. Perinteisesti mikroprosessorien ohjelmointi on vaatinut joko konekielen osaamista tai vähintäänkin vahvaa perehtymistä mikroprosessorin datalehtiin, syvällisempää tietotaitoa elektroniikasta sekä matalan tason ohjelmointitaitoa C-kielellä.

### **2.2.1 Historia**

Arduino-projekti lähti liikkeelle vuonna 2005 Ivreassa, Italiassa, kun joukko opiskelijoita ja koulun henkilökuntaa halusivat kehittää helpon ja halvan alustan, jolla vähemmän tekniset henkilöt voivat hyödyntää mikroprosessorien mahdollisuuksia erilaisissa projekteissaan. [2] Arduino sai valtavan suosion heti ensimmäisen kahden vuoden aikana ja sitä myytiinkin 50000 kappaletta. Suuri vaikutus sen nopeaan suosioon oli halpa hinta, helppo käyttöisyys ja USB-liitännäisyys.

### **2.2.2 Ominaisuudet**

Arduino-kehitysalustalla on integroituna myös jännitteensyöttö mikroprosessorille, sen tarvitsemat lisäkomponentit ja käyntitaajuuden määrittelevä kide sekä USB-datan mikroprosessorin ymmärtämään sarjamuotoon muuntava puskuripiiri. [6]

Arduinon ohjelmointi tapahtuu sen omassa kehitysympäristössä, joka on Java-pohjainen. Tämä mahdollistaa käyttöjärjestelmäriippumattoman kehityksen. Ohjelmointikielenä toimii erityisesti Arduinoa varten kehitetty kieli, joka vastaa hyvin pitkälti C/C++ -

ohjelmointikieltä. Käyttäjältä jää siis kokonaan pois bittitason rekistereiden ohjelmointi. Tämän mahdollistaa piirillä valmiiksi oleva bootloader-ohjelma, joka tulkkaa Arduinon oman koodin mikroprosessorin ymmärtämäksi konekieleksi. Tarvittaessa rekisterien kirjoitukset onnistuvat kuitenkin mahdollistaen AVR-prosessorin kaikkien ominaisuuksien käyttämisen.

Koska käyttäjän koodi tulkataan, hidastaa se suoritusnopeutta hieman, mutta ATmega328 prosessorin käydessä 16MHz (MegaHertz) taajuudella, ei nopeuden menetys ole merkittävää.

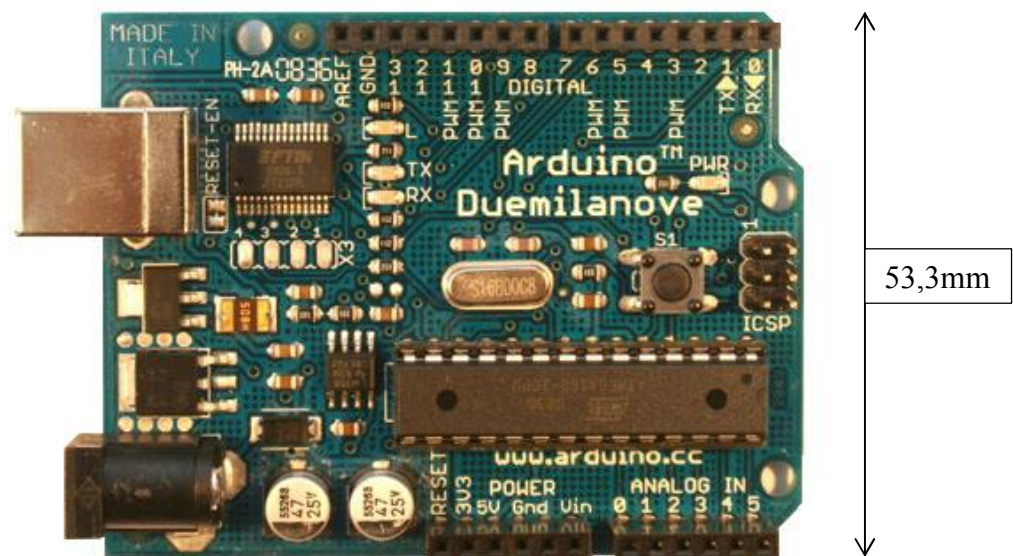
Kehitysympäristössä löytyy suoraan toiminnot koodin polttamiseksi mikroprosessorin flash-muistiin.

Ominaisuus, joka helpottaa ja nopeuttaa kehitysalustan käyttöä, on sen valmiit kirjastot erilaisten toimintojen käyttöön. Siitä löytyy valmiit ominaisuudet IO-porttien lukemiseen sekä kirjoittamiseen, PWM-signaalien luomiseen, servomoottorien ohjaukseen, LCD-näyttöjen (Liquid Crystal Display) ohjaukseen sekä sarjamuotoisen datan tietokoneelle lähettämiseen. Arduino soveltuukin erityisen hyvin interaktiivisiin sovelluksiin, joissa luetaan tietoja sensoreilta ja niiden tuottaman informaation perusteella ohjataan oheislaitteita.

### **2.2.3 DUEMILANOVE-alusta**

Arduino Duemilanove (Duemilanove tarkoittaa italiaksi ”2009”) on Arduino sarjan kolmanneksi uusin kehitysalusta, jossa ytimenä on Atmelin ATmega328-mikroprosessori. Mikroprosessorissa on 32 kilotavua vapaata flash-muistia, josta bootloadin jälkeen vapaaksi jää 30 kilotavua ohjelmakoodia varten. Käytössä on myös 1 kilotavu EEPROM-muistia, jota voidaan käyttää ajonaikaisten tietojen pitkäaikaiseen tallentamiseen. Tällöin virran katkaisu ei hukkaa muistin sisältöä.

Duemilanovessa vapaasti käytettäviä IO-portteja on 14 kpl, joista kuusi kykenee tuottamaan PWM-signaalia. Kuusi porttia toimii analogisena sisääntulona, jollaisella tässä työssä luen mm. ultraäänisensorin lähettämää dataa. Prosessori käy vakiona 16 MHz taajuudella, joka riittää erinomaisesti nopeaankin tiedonkäsittelyyn. Kuvassa 1 näkyy kuinka Arduino on kasattu pieneksi piirilevyksi, jossa kaikki oleellinen on kätevästi saatavilla. [7]



Kuva 1: Arduino Duamilanove [8]

Käyttäjännitteensä Arduino voi ottaa joko USB-portista tai ulkoisesta jännitteensyötöstä. Mahdollisuus käyttää USB-porttia virtalähteenä on erityisen kätevä varsinkin vähän virtaa käyttäviä prototyyppejä valmistaessa. Pitää ottaa huomioon kuitenkin, että USB-portin virranantokyky on vakiona vain 500 mA (milliAmpeeria).

Yksittäiset IO-portit pystyvät antamaan n. 40 mA virran. Monissa sovelluksissa tarvitaan tästä huomattavasti suurempia virtoja, jolloin portin virtaa käytetään ohjaamaan transistoria, joka puolestaan ohjaa suurempia kuormia.

Arduinossa on UART TTL-puskuripiiri (Universal Asynchronous Receiver/Transmitter,

Transistor-Transistor Logic), joka mahdollistaa kaksisuuntaisen sarjamuotoisen tietoliikenteen mikroprosessorin ja usb-portin välillä.

## **2.3 AVR**

AVR on mikroprosessorivalmistaja Atmelin suunnittelema mikrokontrolleriperhe, joka pitää sisällään kolmentyyppisiä mikroprosessoreja: ATTiny-sarja, AT90x-sarja ja ATmega-sarja, jota käytän tämän työn tekemisessä. AVR-sarjan mikrokontrolleihin sisältyy RISC-arkkitehtuuria (Reduced Instruction Set Computing) käyttävä ydin, uudelleenohjelmoitavaa FLASH-muistia, ohjelmoitavia IO-portteja (Input Output) sekä pidempiaikaiseen tallennukseen soveltuvaa EEPROM-muistia (Electrically Erasable Programmable Read-Only Memory [9]. Atmelin AVR- mikropiirit kehitettiin vuonna 1996 ja niistä tuli hyvin nopeasti erittäin suosittuja niiden sisältämän uudelleenohjelmoitavan FLASH-muistin ansiosta. Aikaisemmin suosiossa olleet Microchip Technologyn PIC-mikropiirit (Peripheral Interface Controller) olivat pitkälti kertaohjelmoitavia eli ohjelmointivirheen sattuessa piiri oli käyttökelvoton.

## **2.4 Servomoottori**

Servomoottori on usein varsinkin radio-ohjattavissa laitteissa käytetty moottorityyppi. Servomoottori pitää sisällään DC-moottorin (Direct Current), ohjaimen moottorille, moottorin asentoa tarkkailevan asento-anturin sekä ratasvälitykset moottorin kierrosten säätämiseksi. Servomoottoreissa on 3-napainen liitäntä, jossa on 2 pinniä käyttöjännitteelle sekä 1 pinni ohjaussignaali, joka on PWM-muotoista (Pulse Width Modulation) jännitettä.

Servomoottorin ohjauspiiri on jatkuvasti tietoinen moottorin asennosta potentiometrin välityksellä. Moottori ei voi pyöriä täyttä kierrosta eli 360 astetta ympäri, koska asennon tunnistus menetettäisiin. Analogisen potentiometrin fyysiset rajoitukset tulevat vastaan. Tämän vuoksi servomoottorien rattaistoissa on mekaaninen osa, joka estää moottoria pyörittämästä täyttä kierrosta ympäri.

Rajoittimen vuoksi joudun muokkaamaan servomootoreita työtäni varten, jotta saan poistettua mekaaniset stopparit rattaistosta. Tällä tavalla servomootoreista saadaan täydet 360 astetta ympäri pyörivät ja pyörimisen nopeutta voidaan säätää.

## **2.5 Ultraäänisensori**

Ultraäänisensori on moduuli, jossa on integroituna ultraäänipulssia lähettävä- sekä vastaanottava osa. Koodattu signaalipulssi lähetetään eteenpäin ja pulssin heijastuttua kohteesta se vastaanotetaan. Signaalin vastaanottamisen ja lähetyksen ajan erotuksesta, viipymästä, voidaan laskea etäisyys kohteeseen. [10]

Sensori mahdollistaa edessä olevien esteiden tunnistamisen käyttäen samaa metodia, kuin lepakot käyttävät luonnossa suunnistaakseen. Lepakot pystyvät signaalin perusteella päättämään myös kohteen suunnan ja koon, mutta robotille riittää pelkkä tieto onko jotain edessä vai ei. [10]

## **3 RATKAISUMENETELMÄT**

Mobiilirobotin rakentamiseen ja toimintaan liittyy monia ongelmia ja niiden vaihtoehtoisia ratkaisuja. Alla on esitelty muutama pääongelmaa sekä toteutetut ratkaisumenetelmät.

Ensimmäinen ongelma on, millä tavalla robotin eteneminen toteutetaan. Olen päättänyt ratkaista tämän käyttämällä servomootoria, joka on 360 astetta ympäri pyörimiseen muokattu. Servomootoreihin kiinnitän kumiset pyörät, jolloin eteneminen tapahtuu niiden avulla rullaamalla.

Toinen ongelma on esteiden havainnointiin ja väistämiseen käytettävä teknologia. Toteutan ratkaisun työssäni käyttämällä ultraäänisensoria havainnoimaan mahdolliset esteet. Ultraäänisensoria liikutellaan servomootorin varassa, jolloin saadaan laajempi tilannekuva siitä, kumpaa kautta este kannattaa mahdollisesti lähteä väistämään.

Kolmas ongelma on navigaatioon käytettävä logiikka. Pysin ratkaisemaan ongelman käyttämällä hyvin yksinkertaista itse kehitettyä algoritmiä, jossa otetaan huomioon ultraääni-sensorilta saatava informaatio ja tämän perusteella tehdään päätös kumpaa kautta estettä väistetään. Algoritmi ei pidä sisällään mitään kehittyneempää tietoa robotin laskennallisesta sijainnista.

### **3.1 Robotin mekaniikka**

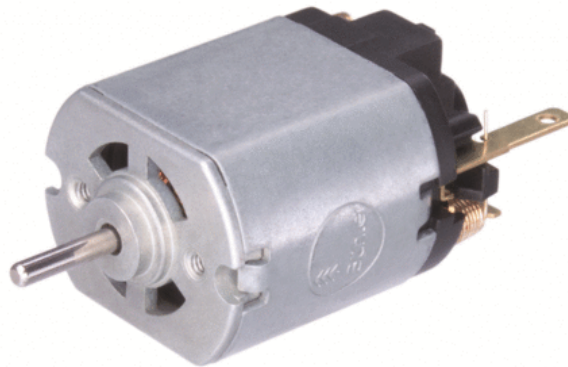
Robotiksi luokiteltavilla laitteilla on lukematon määrä mahdollisuuksia miten ne liikkuvat. Ne voivat rullata, ryömiä, kiemurrella, kävellä, hyppiä, lentää, uida, tai edetä, melkein mielikuvituksen ollessa rajana, lähes millä tavalla tahansa. [1]

Tätä työtä varten olen valinnut liikkumismenetelmäksi pyörillä rullaamisen, koska se on yksinkertaisin ja helpoin tapa toteuttaa mobiilirobotti.

Kun etenemismenetelmänä on rullaaminen, määrittää se heti ehdon käytettävälle moottorityypille. Sen on oltava vapaasti liikkuva, 360 astetta ympäri pyörivä ja jonka suuntaa voidaan kontrolloida. Käytännössä tämä rajaa käytettävät moottorityypit kolmeen tyyppiin: DC-moottorit, askelmoottorit sekä servo-moottorit.

#### **3.1.1 Tasavirtamoottorit (DC)**

Tasavirtamoottorit (DC) ovat perinteisimpiä sähkömoottoreita jotka keksittiin jo 1800-luvulla, joiden vahvuuksiin kuuluu vahva vääntöteho sekä edullinen tuottokustannus. [11] Huonoina puolina voidaan pitää suurta virrankulutusta sekä epätarkkaa ohjausta. Robotissa voidaan käyttää tasavirtamoottoria, mutta tällöin tarvitaan erillinen ohjauspiiri, joka on kykenevä syöttämään moottorin tarvitsevan virran, huolehtimaan mahdollisista jarrutuksista ja kääntämään moottoriin syötettävän jännitteen polariteetin pyörintasuunnan vaihtamiseksi. Kuvassa 2 on tyypillinen esimerkki tasavirtamoottorista.



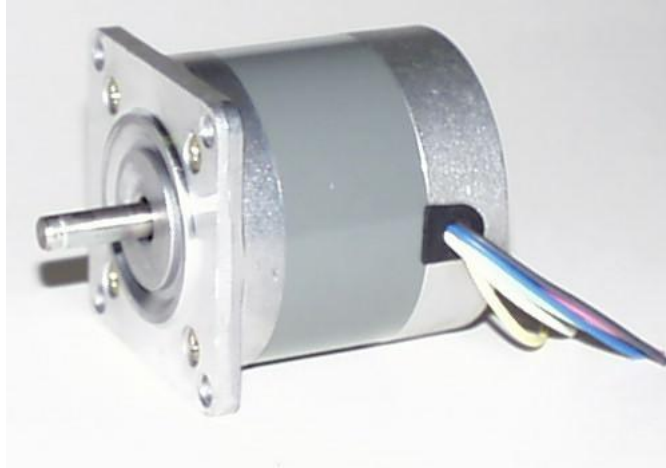
**Kuva 2: Tasavirtamoottori [12]**

Koska tasavirtamoottori ei ole kovin tarkka ohjaukseltaan, asettaa se omat haasteensa robotiikan sovelluksiin, koska niissä useimmiten tarvitaan tarkkaa ohjausta ja tietoa moottorin asennosta. Tasavirtamoottori ei sellaisenaan tarjoa minkäänlaista asentohavainnointia.

### **3.1.2 Askelmoottorit**

Askelmoottorit, kutsutaan myös steppereiksi, ovat myös eräänlaisia tasavirtamoottoreita, mutta niiden toimintaperiaate on monimutkaisempi kuin tavallisen moottorin. Niissä on useimmiten kaksi kela (unipolar) tai neljä kela (bipolar), joihin virtaa syöttämällä moottori voi ottaa tarkasti määritettyjä askelia ja täten tarjota hyvinkin tarkkaa askellusta. [13] Steppereitä käytetään juuri siksi paljon tarkkuutta vaativissa sovelluksissa kuten erilaisissa tulostimissa, skannereissa sekä kopiokoneissa. Askelmoottorit ovat suosittuja myös robottien rakentamisessa, koska ne tarkkuutensa ansiosta tarjoavat hyvin tarkan ohjaustavan erilaisiin robotiikan sovelluksiin, joissa ei tarvita suurta vääntömomenttia. Kuvassa 3 on unipolaarinen askelmoottori jonka ohjaukseen tarvitaan kuusi kanavaa.





**Kuva 3: Askelmoottori [14]**

Askelmoottoreiden huonona puolena voidaan pitää hankalampaa ohjausta, joka vaatii useita lähtöjä ja erityisesti ohjausta varten suunnitellun ohjelman, jolla stepperin eri keloja ohjataan oikeassa järjestyksessä. Jos käytetään bipolar-tyyppistä askelmoottoria tarvitaan lisäksi erillinen H-bridge ohjaus, jolla käämille syötettävän jännitteen polariteetti käännetään. [13]

Askelmoottori saattaa osoittautua hankalaksi myös tilanteessa, jossa vaaditaan suurta vääntömomenttia. Tällöin se saattaa antaa yhden tai useamman askeleen periksi, mikäli vääntöä ei ole tarpeeksi. Tämä puolestaan aiheuttaa ongelmia, sillä moottorin asennon seuraaminen täytyy toteuttaa ohjelman puolella otettuja askeleita laskemalla. Syntyy siis tilanne, jossa ohjelma luulee, että askelia on otettu, kun todellisuudessa pari askelta jääkin väliin.

### **3.1.3 Servomoottorit**

Servomoottorit ovat toiminnallisuudeltaan tasavirtamoottoreita, mutta niihin on yleensä integroituna samaan pakettiin myös servo-ohjauspiiri. Kuvassa 4 nähdään työssä käytetty servomoottori. Käytännössä tämä tarkoittaa ohjauspiiriä, joka tarkkailee asema-anturin perusteella moottorin asentoa ja pyrkii ohjaamaan sen haluttuun asentoon, vastaamaan ohjaussignaalia. Mitä kauempana servo on halutusta asennosta, sitä suuremmalla voimalla ohjauspiiri ohjaa moottoria kohti haluttua asentoa. Kun moottori saavuttaa halutun asennon,

se ei kuluta virtaa ennen kuin asento pyrkii muuttumaan.[15]



Kuva 4: Futaba s3001 - servomoottori [16]

Servojen etuna on niiden erittäin suuri vääntökyky ja tarkka ohjaus. Moottoreita ohjataan PWM-signaalilla, jolloin ohjauspiirille syötetyn signaalin pulssinleveys määrittää halutun asennon. Ohjauspiirille menevä jännite ei ohjaa suoraan moottoria, joten signaaliksi riittää TTL-tasoinen signaali. Näin ollen se on erityisen kätevä rakentaessa robottia, koska tarvittava ohjaussignaali voidaan syöttää suoraan mikrokontrollerin IO-pinnistä.

Servomoottoreita on useita eri tyyppisiä ja osa niistä on ympäriryöriä, mutta suurin osa liikkeeltään rajoitettuja. Servoja voi olla sekä analogisia sekä digitaalisia.

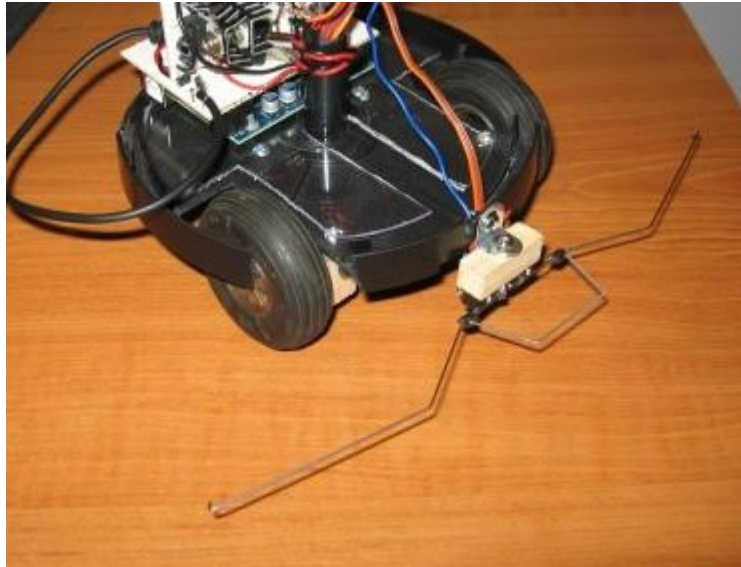
### 3.2 Havainnointi

Oleellinen osa mobiilirobottien toimintaa on niiden jonkinlainen kyky havainnoida tai hahmottaa ympäristöä sekä tehdä päätöksiä siitä saadun informaation perusteella eteenpäin navigoidessa. Tapoja havaita ympäristöä on lukematon määrä ja monet niistä perustuvat sensoreihin. [3]

Alkeellisia mobiilirobotteja rakentaessa käytössä on voi olla yksinkertaisiin on-off-kytkimiin perustuvia kosketusantureita, valoa havainnoivia sensoreita kuten infrapunasensoreita tai äänen havainnointiin perustuvia sensoreita, kuten ultraäänisensoreita. [1]

### 3.2.1 Mikrokytkimet

Mikrokytkimet ovat tyypiltään hyvin yksinkertaisia, yleensä varsin herkkiä on-off-kytkimiä. Niitä käytetään robottien rakentamisessa erilaisten puskureiden tai eläinmaailman tuntosarvien tapaan kertomaan esteen kohtaamisesta. Mikrokytkimeen kiinnitetty jatke kohtaa esteen, jolloin kytkimen tila muuttuu avoimesta suljetuksi. Kuvassa 5 nähdään esimerkki mikrokytkimestä yhdistettynä puskuriin.



Kuva 5: Mikrokytkimellä toteutettu törmäysanturi [17]

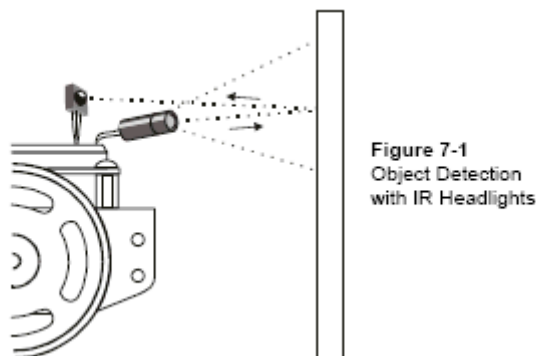
Robotti saa näin tiedon kiinteän etäisyyden päässä olevasta esteestä ja näin voidaan suorittaa haluttu toimenpide. Tyypillinen tapa käyttää mikrokytkimiä on laittaa robotin oikeaan ja vasempaan etukulmaan havainnoimaan törmäystä esteeseen.

Mikrokytkimet ovat yksinkertainen ja edullinen tapa havainnoida ympäristöä, mutta niiden huonoihin puoliin robotiikassa kuuluu se, että tieto kohteesta saadaan vasta, kun siihen törmätään. Tämä puolestaan antaa paljon vähemmän vaihtoehtoja siihen, kuinka esteen kanssa toimitaan.

### 3.2.2 Infrapuna-sensori

Infrapuna-sensorit ovat yksi sovellus valoa lähettävistä ja vastaanottavista sensoreista. Niissä on kaksi osaa, lähettävä ja vastaanottava. Tyypillisesti valodiodilla eli LED:illä (Light Emitting Diode) lähetetään tietylle taajuudelle moduloitua valoa ja sitä vastaanotetaan samalle taajuudelle suotimella rajatulla vastaanottimella. Kun vastaanotetun valon määrä vastaa määritettyä kynnyksiarvoa, mikrokontrolleri tulkitsee, että edessä on kohde.

Useimmiten valolla toimivat läheisyssensorit käyttävät silmälle näkymätöntä infrapunavaloa, koska se on vähemmän häiriöaltis ulkoiselle valolle [17]. Infrapuna-LED lähettää valoa eteenpäin kapealle sektorille, josta se kohteeseen törmätessään heijastuu takaisin vastaanottimelle. Kuvassa 6 esitetään toimintaperiaate.



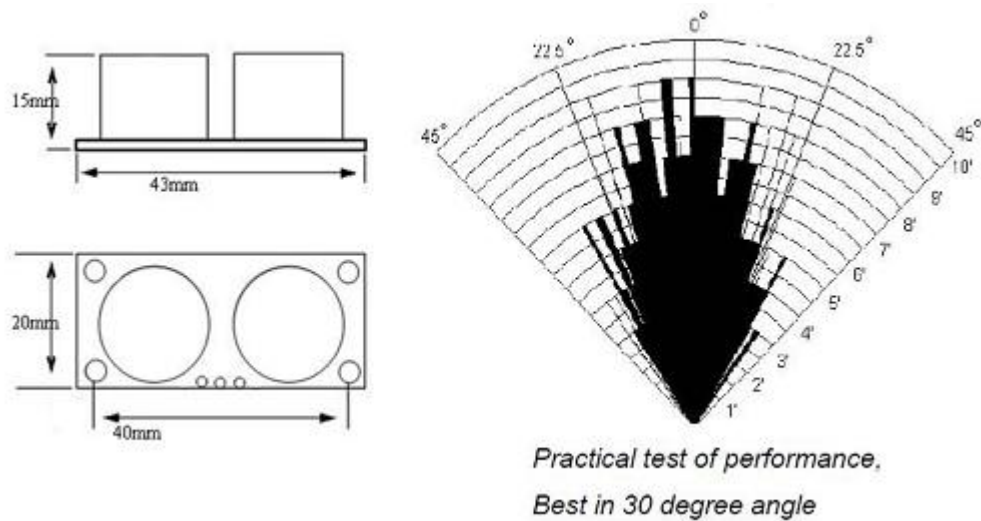
Kuva 6: IR-sensorin toimintaperiaate [21]

Infrapunasensorit sellaisenaan eivät kovin helposti pysty ilmoittamaan etäisyyttä kohteesta, vaan ainoastaan 0- tai 1-tyyppisen raja-arvon. Mikäli infrapunaa halutaan käyttää etäisyyden tarkempaan mittaamiseen, täytyy signaaliin moduloida erillisiä pulsseja, joiden lähettämisen ja vastaanottamisen välistä aikaa tarkkaillaan mikrokontrollereilla. Tämä on kuitenkin osittain epäluotettavaa, koska esim. loisteputkivalot generoivat paljon infrapuna-taajuisia säteitä, jotka häiritsevät signaalia.

### 3.2.3 Ultraäänisensori

Ultraäänisensorit käyttävät nimensä mukaisesti korkeataajuisia ääntä esteiden havainnointiin edessään. Aikasemmin käsittelemästäni infrapunasensorista poiketen ultraäänisensori kykenee myös ilmoittamaan etäisyyden kohteesta [1]. Sensori lähettää lyhyitä pulsseja, jotka esteen kohdatessaan heijastuvat takaisin ja sensorissa oleva vastaanotin havaitsee ne. Kun tiedetään äänennopeus ilmassa voidaan yksinkertaisella yhtälöllä laskea lähetetyn signaalin ja sen vastaanoton välisestä viiveestä etäisyys kohteeseen. Mekanismi on sama mitä luonnossa esimerkiksi lepakot käyttävät liikkeessaan pimeässä.

Ultraäänisensorit voivat olla mallista riippuen hyvinkin tarkkoja. Tietämällä käytettävän sensorin havainnointialueen segmentti voidaan sensoria liikuttelemalla skannata ympäristöä ja sopivilla ohjelmilla jopa piirtää kartta siitä. Kuvassa 7 näkyy työssä käyttämäni HC-SR04 ultraäänisensorin havainnointialue ja fyysiset mitat.



Kuva 7: HC-SR04 ultraäänisensorin havainnointialue [18]

### 3.3 Navigaatioalgoritmi

Aivan alkeellisimpia robotteja lukuunottamatta ne koostuvat fyysisen rakenteensa lisäksi ohjelmakoodista, joka toimii laitteen ”sähköisinä aivoina”. Koodia suoritetaan mikroprosessorissa, joka ohjelmoinnin pohjalta toimii sensoreilta saamiensa informaation avulla. Toimiakseen robotit tarvitsevat siis toimintaohjeen eli algoritmin - kuinka se toimii saadun informaation perusteella.

Käytettävää algoritmia suunniteltaessa ensimmäisenä vaikuttaa se, mihin käyttöön robotti on suunniteltu [3]. Valmiita vaihtoehtoja löytyy laidasta laitaan, lattialla liikkuvasta lelusta tehtaalla keksejä paketoivaan robottiin sekä avarusalusta Marssiin luotsaavaan tekoälyyn. Yksinkertaisia toimintoja suorittavaan robottiin löytyy suuri määrä valmiita algoritmeja, kun taas spesifisempään tehtävään algoritmi on suunniteltava ja toteutettava itse.

Pelkästään mobiiliroboteille, joiden tehtävä on liikkua tasaisella alustalla eteenpäin ja vältellä mahdollisia esteitä, löytyy lukematon määrä erilaisia valmiita algoritmeja. Algoritmien kompleksisuus vaihtelee yksinkertaisesta ”esteen havaittua, käänny vasemmalle” -tyyppisestä algoritmista aina virtuaaliseen ympäristön kartoitukseen ja sen perusteella älykkäillä metodeilla reitin optimaalisesti valitseviin algoritmeihin [3].

Tässä työssä käytän yksinkertaista itse laatimaani algoritmia, jonka avulla robotin on mahdollista suunnistaa tasaisella pinnalla eteenpäin väistellen esteitä törmäämättä niihin. Lisänä algoritmissa on ominaisuus, että robotti väistää estettä sille puolelle missä on ultraääni-sensorin tekemän mittauksen mukaan enemmän vapaata tilaa.

Algoritmin toiminta:

1. Kulje eteenpäin mitaten etäisyyttä kohteisiin.
2. Jos ultraäänisensorilta saatu mittaustulos on lähempänä kuin asetettu raja-arvo, pysähdy.
3. Käänä ultraäänisensoria vasemmalta äärilaidasta oikealle äärilaitaan. Mittaa etäisyys ja palauta suunta, jossa on suurempi etäisyys kohteeseen.

4. Suorita peruutus, jossa käännetään tyhjän puolen suuntaan.
5. Aloita algoritmi alusta.

## 4 KÄYTÄNNÖN TOTEUTUS

Robotin rakentaminen alkoi suunnittelulla, jossa kävin läpi mahdollisia komponentteja, vertailin niiden ominaisuuksia ja suunnittelin, millaisia toimintoja robotilta haluan.

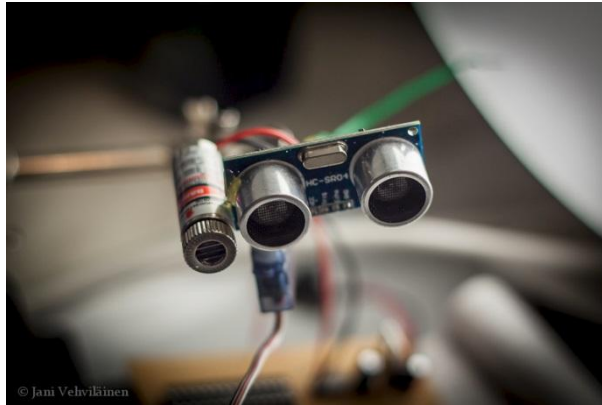
Fyysinen puoli alkoi mekaanisen rungon rakentamisella, komponenttien muokkaamisella [5] ja paikalleen kiinnittämisellä. Rakensin robottia moduuli kerrallaan ja aloin testailla niitä yksitellen tekemällä toiminnan mahdollistavan ohjelmakoodin. Tämän lisäksi jouduin tekemään koodia, jota käytin avuksi servon alueita määrittäessä.

### 4.1 Mekaaninen ja sähköinen kasaus

Ensimmäisenä valmistin robotille rungon, jonka päälle rupeaisin asentamaan osia. Materiaaliksi hankin edullisen akryylisen, 4mm paksun leikkuulaudan, josta oli helppo leikata halutun muotoinen runko.

Robotin on tarkoitus liikkua kahden vetävän pyörän avulla ja takana olla kolmas vapaasti pyörivä rengas tasapainottamassa alustaa. Runkoon leikkasin paikat pyörille ja suunnittelin alustan hieman takapainotteiseksi, jotta robotti ei keikkuisi edestakaisin liikkeellelähdessä. Alustan eteen tein reiän, johon ultraäänisensoria liikutteleva servo asennetaan.

Alustan asentamisen jälkeen kasasin ympäristön hahmottamiseen käytettävän moduulin eli ultraäänisensorin ja sitä liikuttelevan servomoottorin. Lisäsin samaan yksikköön vielä viivalasermoduulin, joka havainnollistaa ultraäänisensorin mittauskohdan keskikohtaa sekä antaa paremman kuvan mittausalueesta. HC-SR04 moduli on kiinnitetty liitinrimaan, johon puolestaan juotin kiinni tarvittavat johdot. Liitinrima on kiinnitetty servomoottorin vipuun kuumaliimalla, kuten sivussa oleva lasermodulikin. Kuvassa 8 näkyy servomoottoriin kiinnitetty kokonaisuus.



**Kuva 8: Ultraäänisensori**

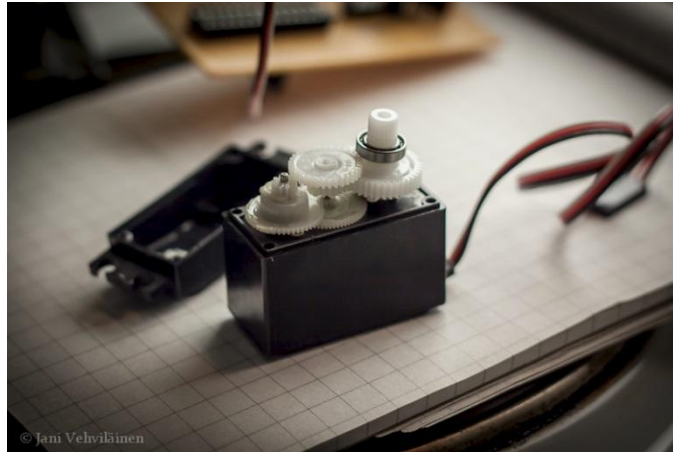
Projektissa käyttämäni Futaba S3001 servomoottori ei ole vakiona ympäripyörivä, joten siihen täytyi tehdä muokkaus, joka mahdollistaa em. toiminnan. Muokkauksen yhteydessä menetetään moottorin asennontunnistus, mutta vastineeksi saadaan pyörimisnopeuden kontrollointi. Servomoottorin muokkaaminen tapahtui avaamalla se ja poistamalla täyden kierroksen estävät mekaaniset esteet rattaista. Kuvassa 9 näkyy rattaan akselin oikealla puolella oleva mekaaninen este.



**Kuva 9: Servomoottorin liikettä rajoittava mekaanen pala**

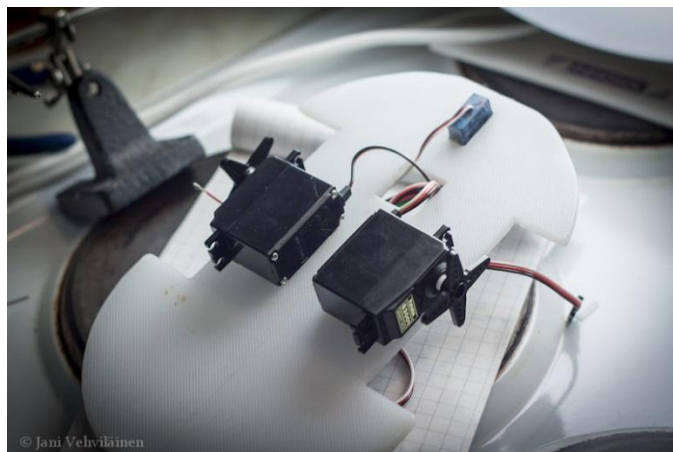
Samalla asentotietoa välittävän potentiometrin kiinnitys porattiin irti. Tällä tavalla muokattu servo ei siis tiedä asentoaan ja jatkaa pyörimistään. Kuvassa 10 on esitetty avattu servomoottori, jossa näkyvät sisällä olevat rattaistot.





**Kuva 10: Futaba S3001 –servomoottori avattuna**

Muokkauksen jälkeen kasasin servot takaisin kokoon ja kiinnitin ne kuumaliimalla alustaan. Servojen mukana tulleet ympyränmalliset vipulevyt puolestaan kiinnitin kuumaliimalla vanhoihin rullaluistimen renkaisiin, joista on laakerit poistettu. Rullaluistimen renkaat toimivat kokonsa ja luistamattomuutensa vuoksi erinomaisesti tämänkaltaisessa projektissa. Kuvassa 11 on robotin alusta, johon kaikki kolme servomoottoria ovat kiinnitettyinä.



**Kuva 11: Robotin runko kiinnitettyjen servomoottoreiden kanssa**

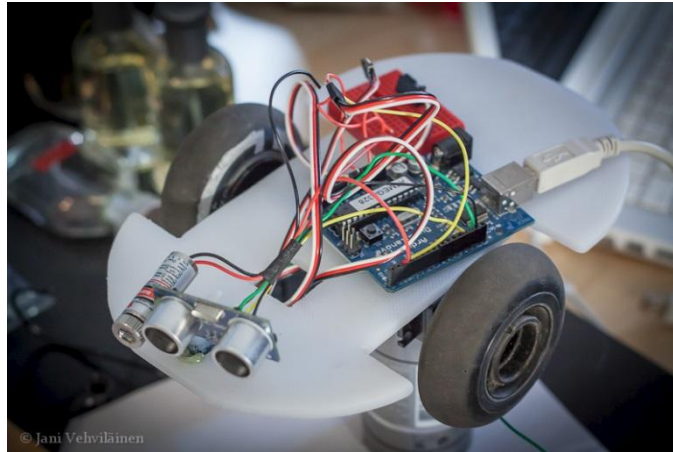
Arduino-kehitysalusta kiinnitettynä alustan päälle kaksipuoleisella teipillä kiinnitettynä, koska haluan, että se on helposti irroitettavissa muihinkin projekteihin tarvittaessa. Kuvassa 12 näkyy robotti, jossa on oleelliset komponentit kiinnitettyinä.



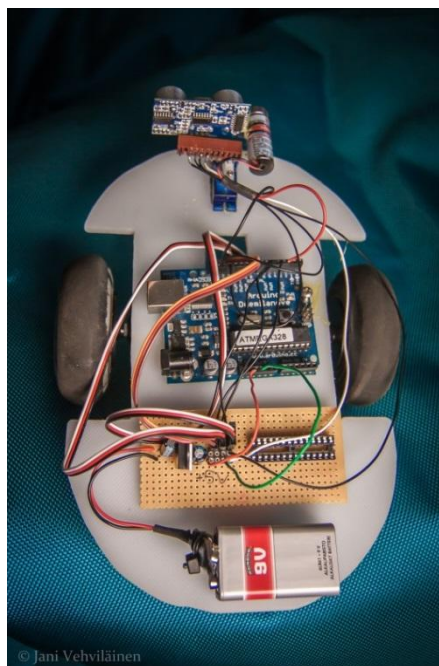
**Kuva 12: Robotti pääkomponentit paikallaan**

Kehitysalusta on kykenevä ottamaan käyttöjännitteensä USB-portista ja kykenee antamaan n. 40mA virtaa jokaisesta IO-portistaan, joten moduulien toiminnallisuutta ohjelmiston kehittämissvaiheessa pystyy testaamaan kätevästi ilman erillistä virtalähdettä.

Lopulliseen robottiin rakensin erillisen virransyötön, joka regulaattorin avulla muuntaa paristolta saamansa käyttöjännitteen robotin käyttämäksi 5 voltin tasajännitteeksi. Erillinen virransyöttö mahdollistaa servomoottoreiden saada tarvitsemansa suuremman virran käyttöön sekä mahdollistaa robotin toimimisen ilman häiritseviä virtakaapeleita. Käyttämäni regulaattori antaa tarvittaessa jopa 1,5 Ampeeria virtaa. Kuvassa 13 robotti on kytketty USB-kaapelilla tietokoneeseen verrattuna kuvaan 14, jossa on mukana erillinen virransyöttö joka ottaa käyttöjännitteen 9v patterista.



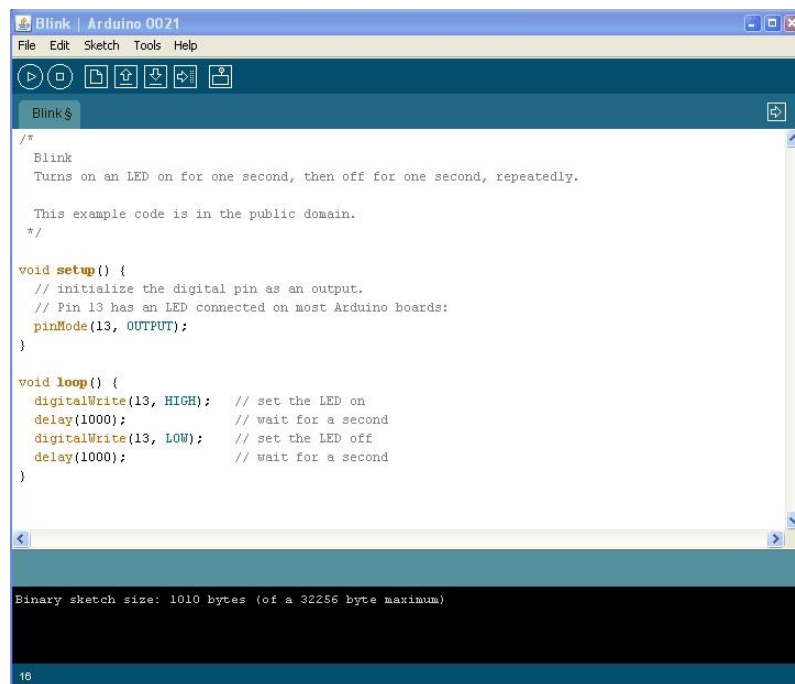
**Kuva 13: Robotti kytkettynä USB-kaapelilla tietokoneeseen**



**Kuva 14: Valmis robotti**

## 4.2 Toimintaälyn ohjelmointi

Arduino tarjoaa ilmaisen java-pohjaisen IDE:n (Integrated development environment) käytettäväksi mikroprosessorissa ajettavien ohjelmistojen kehittämiseen. Arduino IDE:ssä on integroituna koodieditori, työkalut ohjelmakoodin mikroprosessorille lataamiseksi ja sarjaliikenteen monitorointiin. Kuvassa 15 on yksi kehitysympäristön mukana tulevista esimerkkiohjelmista mikroprosessorille.



Kuva 15: Arduino IDE [19]

Robotin toimintaäly koostuu useista aliohjelmista, joilla on omat spesifiset tehtävänsä. Ohjelman suoritus pyörii silmukassa, josta tarvittavia aliohjelmiä kutsutaan vuorollaan suorittamaan niiden toimintoa.

#### 4.2.1 Ultraäänisensorin toiminta

Käyttämäni Micropik HR-SR04 sensori saadaan lähettämään ultraäänipulssi, kun sen data-in pin nostetaan high (jännitteiseen) tilaan 10  $\mu$ S (mikrosekuntin) ajaksi. Vastaavasti sensorin data-out -pinnistä saada signaali, kun sensori on vastaanottanut heijasteen. Mittaamalla tätä lähettämisen ja vastaanottamisen välistä aikaa saadaan laskettua etäisyys kohteesta, koska äänennopeus ilmassa tunnetaan. Koodissa etäisyys muunnetaan vielä senttimetreiksi sillä oletuksena se antaa pituuden jalkoina.

Etäisyysmittausta suoritetaan jatkuvassa loopissa, jolloin jokaisella kierroksella etäisyysarvo tallennetaan muuttujaan. Arvolle tehdään vertailua ja kun etäisyys alittaa määritetyn raja-arvon, käynnistetään toinen aliohjelma, jossa ultraäänisensoria liikutellaan puolelta toiselle. Saadun mittaustuloksen perusteella tehdään päätös ja palautetaan tulos

kummalta puolelta estettä lähdetään väistämään. Samassa aliohjelmassa kytketään myös virta lasermoduliin, joka mittauksen ajan näyttää kohtaa, josta mittaus suoritetaan.

Servomoottorin aisan sijaintia ohjataan (ultraäänen suuntaa) PWM-signaalilla, jonka leveyttä vaihdellaan halutun asennon saavuttamiseksi. Jokaisella servomoottorilla on oma alueensa, jolla se toimii. Useimmiten tämä alue ilmoitetaan moottorin datalehdessä, mutta joissakin tapauksissa se pitää selvittää erillisellä ohjelmalla, joka suorittaa taajuuspyyhkäisyn, ja katsoa mikä on toiminta-alue. Tein tätä käyttöä varten erillisen ohjelman, joka yksinkertaisella loopilla lähti kasvattamaan PWM-pulssin pituutta. Samaan aikaan tarkkailin tietokoneen sarjadata-konsolista mitä arvoja milloinkin syötettiin ja kirjasin ylös leveydet jolla moottori alkoi pyöriä, pysähtyi, ja pyöri toiseen suuntaan.

#### **4.2.2 Liikkuminen**

Edellisessä kappaleessa kuvattiin servomoottorin toiminta-alue. Liikkumista varten muokatuille servomoottoreille pitää hakea myös absoluuttinen keskikohta, jolloin moottorit eivät pyöri. Koodissa on tätä varten erityinen kalibrointiosuus, jolla arvoja voidaan hienosäätää.

Robotin liikkeessä eteenpäin, kumpikin servomoottori pyörii samalla vauhdilla eteenpäin. Ohjelmakoodissa on kuitenkin otettava huomioon, että moottorit ovat asennettu robotin alle akselit eri suuntiin. joten toisen moottorin täytyy pyöriä eri suuntaan, jotta renkaiden pyörimissuunta on kummallakin puolella eteenpäin. Vastaava on otettava huomioon myös kääntymisissä, jolloin robotin toinen pyörä pyörii eteenpäin ja toinen taakse, jonka seurauksena robotti voi kääntyä ympäri lähes paikoillaan.

#### **4.2.3 Testaus**

Robotin rakennuksen aikana testasin ohjelmakoodia jatkuvasti mahdollisten virheiden osalta. Koodasin toimintaa komponentti kerrallaan joten aina, kun sain uuden

toiminnallisuuden tehtyä, otin sen välittömästi käytäntöön. Ultraäänisensorin mittauksen lukemia seurasin sarjaportin kautta tietokoneelta ja valitsin käytännön perusteella toimivimmat luvut. Halusin, että robotti ei lähde mittaamaan etäisyyttä skannaamalla liian aikaisin.

Servomoottoreiden ohjausta testasin alkuun siten, että robotti oli ilmassa ja renkaat pyörivät tyhjänpäällä. Laskin paljonko robotti etenee yhdellä renkaan kierroksella ja lähdin säätämään käännoksissä tarvittavia arvoja tämän mukaan. Renkaan kehän pituuden  $p$  sain laskukaavasta  $p = 2\pi r$  jossa  $r$  on renkaan säde.

Valmista robottia testasin asuntoni lattialla laittamalla sen kulkemaan eteenpäin ja tehden erinäisistä tavaroista esteitä sen eteen.

## **5 POHDINTA JA TULEVAISUUS**

Työ oli omalta osaltani hyvin antoisa ja robotiikan maailmaa käytännön tasolla avaava kokemus. Vastaan tuli asioita, joita en ollut koskaan edes miettinyt ennen kuin tässä yhteydessä törmäsin niihin. Yksi oli toimintalogiikan merkitys toimintaan ja lukemattomat ratkaisumallit - miten samat asiat voi tehdä eri tavalla toimintalogiikan koodissa.

Työn ratkaisumallit osoittautuivat hyvin toimiviksi valittuun ongelmaan, joka oli tässä työssä esteitä väistelevän robotin rakentaminen. Valituilla ratkaisutavoilla robotti saatiin kulkemaan eteenpäin, mittaamaan etäisyyttä kohteeseen ja tekemään väistötoimenpiteet oletetulla tavalla. Projektin aikana suurimmat haasteet syntyivät käytettävän toimintaälyn koodaamisessa eli ohjelmistotasolla.

Vaikka servomoottorit olivat samaa mallia, on niissä tehtaalta tullessakin pieniä eroja. Näin ollen kaikkiin moottoreihin täytyi tehdä erillisellä koodilla kalibrointi ja haarukoida, minkä mittaisilla pulseilla moottorit pyörivät haluttuun suuntaan. Yksi suuri haaste tässä oli saada kummankin renkaan moottori pyörimään samaan suuntaan samalla nopeudella.

Kun käytettävän moottorin, Futaba S3001, pyörimisnopeus nopeimmillaan on käytetyllä TTL-tasoisella 5 V jännitteellä 0,28 sekuntia / 60 astetta [20] eli täysi kierros n. 1,68 sekunnissa, on löydettävä lisäksi pulssinleveys, jolla moottori pyörii huomattavasti hitaammin. Toiminnan varmistamiseksi jouduin tekemään useita testejä ja hakemaan sopivat pyörimisnopeudet tekemällä kokeiluja eri nopeuksilla.

Ultraäänisensorin toiminnassa tuli vastaan ominaisuus mitä en ollut ottanut aikaisemmin huomioon: mitä kauempana mahdolliset kohteet olivat sensorista, sitä kauemmin yhden ohjelmakierroksen ajaminen kesti, koska jokaisella ohjelmakierroksella luettiin etäisyys. Äänennopeus on varsin suuri, mutta kun kyseessä on mikroprosessori, joka käy 16 MHz taajuudella, erosta 2 cm ja 200 cm välillä syntyy jo huomattava ero nopeuteen.

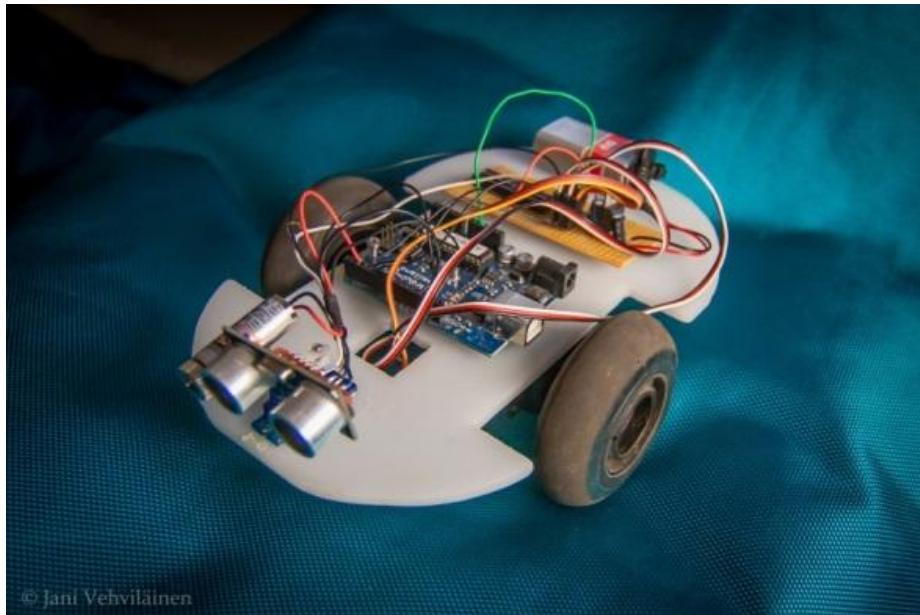
Robotin toimintalogiikka on tällä hetkellä niin yksinkertainen, kuin se vain voi olla, jotta se toteuttaa tehtävänsä eli esteiden väistelyn. Halutessaan robottiin voisi lisätä enemmän älyä esimerkiksi eräänlaisen reitinhahmotuksen suhteen. Tässä tapauksessa robotti käyttäisi erillistä algoritmia suunnistamiseen eli sillä olisi selkeä päämäärä mihin se pyrkisi. Yksinkertaisimpia suunnistusalgoritmeja ovat BUG1 ja BUG2. [3]

Tilan hahmottamista voisi parantaa yhdistämällä ultraäänisensorin liikealueen robotin omaan liikkeeseen eli ennen suunnan valintaa tehtäisiin useampia skannauksia ympäristöön ja sen perusteella tehtäisiin lopullinen suuntapäätös.

Ultraäänisensorin toimintaa voisi tarkentaa tekemällä mittaukselle eräänlaista virheentunnistusta eli absoluuttisten arvojen sijaan mitattaisiin keskiarvoa peräkkäisistä mittauksista ja pyrittäisiin näin eliminoimaan mahdolliset mittausvirheet. Välillä ultraäänisensoriin tuli vikaheijasteita jotka antoivat arvoksi huomattavasti poikkeavampia lukemia.

Loppuyhteenvedona voisinkin todeta, että robotin toimintalogiikalla on hyvin merkittävä osuus lopulliseen suorituskykyyn. Suurin vaikutus siihen oli aikaisemmin mainitsemani etäisyyden mittaamisen viive.

Kuvassa 16 on valmis robotti edustavimmillaan.



Kuva 16: AVR-pohjainen mobiilirobotti

## 6 YHTEENVETO

Kokonaisuutena AVR-pohjainen mobiilirobotti oli erittäin mielenkiintoinen projekti. Siinä tuli käsiteltyä niin robotiikkaa, elektroniikkaa kuin ohjelmistokehitystäkin.

Robotin liikutteluun valitsemani muokatut servomootorit toimivat hyvin niin kuin niiltä alunperin odotinkin. Moottorille syötettävällä pulssilla nopeutta ja suuntaa pystyi kontrolloimaan tarkasti. Moottoreissa oli tarpeeksi vääntöä, että robotti pystyi liikkumaan vaivattomasti ja säilyttämään tarkkuuden käännoksissä. Tämän takasi valmistajan ilmoittama 2,38kg-cm vääntö-luku suhteessa robotin kevyeen massaan [20]. Robotin ohjaukseen vaadittavan logiikan kanssa oli aluksi haasteita saada kummankin moottorin pyörimisnopeudet samaksi.

Ultraäänisensori toimi tämänkaltaiseen tarkoitukseen hyvin sillä tarkan mittauksen perusteella oli helppo todella milloin este on edessä. Valmistajan ilmoittama



mittaustarkkuus, 3mm erottelu [18], olisi riittänyt paljon tarkempaankin ympäristön havainnointiin, kuin tässä projektissa käytettiin. Ultraäänisensorin pystysuuntauksella oli huomattavasti suurempi merkitys, kuin olin alunperin kuvitellut. Aluksi olin kääntänyt sensorin liian ylös, jolloin se ei havainnut renkaiden korkeudella olevia esteitä.

Esteiden väistelemistä varten kehittämin algoritmi oli yksinkertainen ja hyvin pelkistetty. Testaamisen perusteella, se teki mitä algoritmin oli tarkoitus, mutta ei mitään enempää. Samalla robottirungolla olisi mahdollisuus huomattavasti kehittyneempään navigaatioon, kun siihen kirjoittaisi kehittyneemmän ohjelman, joka hyödyntäisi valmiiksi asennettua sensoria.

## LÄHTEET

[1] Joseph L. Jones, Bruce A. Seiger, Anita M. Flynn (1999)

Mobile Robots – Inspiration to Implementation, A K Peters/CRC Press

[2] Kenneth Leung, A History of the Arduino Microcontroller (Viimeksi vierailtu 26.7.2014)

Saatavissa:

[http://www.kenleung.ca/\\_portfolioassets/PDF/HistoryOfArduino\\_KenLeung.pdf](http://www.kenleung.ca/_portfolioassets/PDF/HistoryOfArduino_KenLeung.pdf)

[3] Howie Choset, Kevin Lynch (2005)

Principles of Robot Motion – Theory, Algorithms, and Implementation, A Bradford Book

[4] Roland Siegwart (2011)

Introduction to Autonomous Mobile Robots, The MIT Press; second edition

[5] Tero Karvinen, Kimmo Karvinen (2009)

Sulautetut – Opi rakentamaan robotteja ja muita sulautettuja, readme.fi

[6] Arduino official web page (Viimeksi vierailtu 11.8.2014)

Saatavissa: [www.arduino.cc](http://www.arduino.cc)

[7] Arduino Duemilanove (Viimeksi vierailtu 26.7.2014)

Saatavissa <http://arduino.cc/en/Main/arduinoBoardDuemilanove>

[8] ArduinoDuemilanove.jpg (verkkokuva) (Viimeksi vierailtu 15.9.2014)

Saatavissa: <http://arduino.cc/en/pmwiki.php?n=Main/arduinoBoardDuemilanove>

[9] Official AVR web page (Viimeksi vierailtu 28.7.2014)

Saatavissa: <http://www.atmel.com/products/microcontrollers/avr/default.aspx>

[10] Lesson: How Does an Ultrasonic Sensor Work? (Viimeksi vierailtu 25.7.2014)

Saatavissa:

[http://www.teachengineering.org/view\\_lesson.php?url=collection/umo\\_/lessons/umo\\_sensorswork/umo\\_sensorswork\\_lesson06.xml](http://www.teachengineering.org/view_lesson.php?url=collection/umo_/lessons/umo_sensorswork/umo_sensorswork_lesson06.xml)

[11] DC Motors – How they work, in 4 parts (Viimeksi vierailtu 25.7.2014)

Saatavissa: [http://www.solarbotics.net/starting/200111\\_dcmotor/200111\\_dcmotor.html](http://www.solarbotics.net/starting/200111_dcmotor/200111_dcmotor.html)

[12] 10-15022011-099920L.gif (verkkokuva) (Viimeksi vierailtu 15.9.2014)

Saatavissa:

<http://www.tracepartsonline.net/%28S%28yresv255ji1vyhirpatauh55%29%29/partdetails.aspx?Class=BUHLER&ClsID=/Buhler/Buhler.010/&PartID=10-15022011-099920>

[13] Partco verkkokauppa – Arduino ja askelmoottorit (Viimeksi vierailtu 25.7.2014)

Saatavissa: [http://www.partco.biz/verkkokauppa/article\\_info.php?articles\\_id=6](http://www.partco.biz/verkkokauppa/article_info.php?articles_id=6)

[14] Askelmoottori, stepper.jpg (verkkokuva) (Viimeksi vierailtu 15.09.2014)

Saatavissa: [http://probyte.fi/oscom/index.php?cPath=1\\_23&language=en](http://probyte.fi/oscom/index.php?cPath=1_23&language=en)

[15] Hobby servo fundamendals, Darren Sawich (Vierailtu viimeksi 27.7.2014)

Saatavissa: <http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>

[16] Futaba s3001, 3001-4.jpg (verkkokuva) (Viimeksi vierailtu 15.09.2014)

Saatavissa: <http://www.targethobby.com/images/3001-4.jpg>

[17] Mikrokytkin, Robot\_001.jpg (verkkokuva) (Viimeksi vierailtu 15.9.2014)

Saatavissa <http://letsmakerobots.com/robot/project/explorer-1>

[18] Micropik HC-SR04 Datasheet (Viimeksi vierailtu 11.7.2014)

Saatavissa: <http://www.micropik.com/PDF/HCSR04.pdf>

[19] Arduino IDE, install\_6.jpg (verkkokuva) (Viimeksi vierailtu 15.9.2014)

Saatavissa: <http://www.hobbytronics.co.uk/arduino-installation>

[20] Servo Database (Viimeksi vierailtu 11.8.2014)

<http://www.servodatabase.com/servo/futaba/s3001>

[21] IR-sensor, sensor4.gif (verkkokuva) (Viimeksi vierailtu 15.9.2014)

Saatavissa <http://www.ceruleanplains.com/sensors.htm>

## LIITE 1. Lähdekoodi

```
// AVR-pohjainen mobiilirobotti  
// Kandityö, Jani Vehviläinen  
// janive@gmail.com
```

```
//määrittelyt  
#define trigPin 13 //ultraääni  
#define echoPin 12 //ultraääni  
#define servoUAPin 3 //ultraäänen servomoottori  
#define servoUAMin 500 //oikea laita  
#define servoUAMax 1950 //vasen laita  
#define servoUAKeski 1250  
#define servoVasenKeski 1356 // vasen stop  
#define servoOikeaKeski 1356 // oikea stop  
#define laserPin 2 //laser  
#define servoVasenPin 4 //vasen servomoottori  
#define servoOikeaPin 5 //oikea servomoottori  
#define EsteEtäisyys 10 //ultraäänen treshold
```

```
long etäisyys; // etäisyys esteeseen
```

```
void pulseServo(int servoPin, int pulssinpituusUs) // servon ohjaus aliohjelma  
{  
    digitalWrite(servoPin, HIGH);  
    delayMicroseconds(pulssinpituusUs);  
    digitalWrite(servoPin, LOW);  
    delay(20);  
}
```

(jatkuu)

```

long mittaaEtaisyys() // etäisyyden mittaus aliohjelma
{
    long viive, etaisyys;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    viive = pulseIn(echoPin, HIGH);
    etaisyys = (viive/2) / 29.1; //konversio senttimetreiksi
    return etaisyys;
}

```

```

char skannaaEste()
{
    // keskitaServot();
    long oikeaMax = 0;
    long vasenMax = 0;
    digitalWrite(laserPin, HIGH);

```

```

for (int i=servoUAKeski; i<=servoUAMax; i=i+20) //skannaus vasemmalle
{
    pulseServo(servoUAPin, i);
    long matka = mittaaEtaisyys();
    if (matka > vasenMax)
    {
        vasenMax = matka;
    }
    // Serial.print(matka);

```

(jatkuu)

```

// Serial.print(" oikealla ");
    // Serial.println(i);
}

for (int i=servoUAMax; i>=servoUAKeski; i=i-20) //takaisin keskelle
{
    pulseServo(servoUAPin, i);
}

for (int i=servoUAKeski; i>=servoUAMin; i=i-20) //skannaus oikealle
{
    pulseServo(servoUAPin, i);
    long matka = mittaaEtaisyyss();
    if (matka > oikeaMax)
    {
        oikeaMax = matka;
    }
    // Serial.print(matka);
    // Serial.print(" vasemmalle ");
    // Serial.println(i);
}

for (int i=servoUAMin; i<=servoUAKeski; i=i+20) //takaisin keskelle
{
    pulseServo(servoUAPin, i);
}

if (oikeaMax > vasenMax)
{
    return (1);
}

```

(jatkuu)

```

else
{
return (0);
}

}

void Eteenpain() //Aliohjelma eteenpäin menemiseksi
{
pulseServo(servoVasenPin, 1478); //vasen servo eteenpäin
pulseServo(servoOikeaPin, 1112); //oikea servo eteenpäin
delay(20);
}

void KaannosOikealle() //Aliohjelma oikealla kääntymiseksi
{
for (int o=0;o<20;o++)
{
pulseServo(servoVasenPin, 1378); //vasen servo eteenpäin
pulseServo(servoOikeaPin, 1326); //oikea servo taaksepäin
}
}

void KaannosVasemmalle() //Aliohjelma vasemmalle kääntymiseksi
{
for (int v=0;v<20;v++)
{
pulseServo(servoVasenPin, 678); //vasen servo taaksepäin
pulseServo(servoOikeaPin, 400); //oikea servo eteenpäin
}
}

```

(jatkuu)

```
void keskitaServot()
{
  for (int i=0; i<=25; i++)
  {
    pulseServo(servoUAPin, servoUAKeski);
  }
}
```

```
void setup() //alkumäärittelyt
{

  Serial.begin(19200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(servoUAPin, OUTPUT);
  pinMode(laserPin, OUTPUT);
  pinMode(servoVasenPin, OUTPUT);
  pinMode(servoOikeaPin, OUTPUT);
  keskitaServot(); //keskitetään servo

  //alku stoppi
  digitalWrite(laserPin, HIGH);
  delay(3000);
  digitalWrite(laserPin, LOW);

}
```

```
void loop() //itse ohjelman suoritus
{
  digitalWrite(laserPin, LOW);
```

(jatkuu)



```
etaisyys = mittaaEtaisyys();

if (etaisyys <= EsteEtaisyys) // jos matka on alle tresholdin
{
    char ratkaisu = skannaaEste(); //skannataan ympäristöä
    // Serial.println(ratkaisu);
    if (ratkaisu == 1)
    {
        KaannosOikealle();
    }
    else
    {
        KaannosVasemmalle();
    }
}

Eteenpain(); //edetään jatkuvasti jos ei esteitä edessä

//Serial.print(etaisyys); //sarjadataa tietokoneelle debuggausta varten
//Serial.println(" cm");

}
```