

OHJELMOITAVA LAKAISIN

Opinnäytetyö

Koulutuskeskus Salpaus, Nastola

Sähköalan perustutkinto

Elektroniikan ja tietoliikennetekniikan koulutusohjelma

Elektroniikka-asentaja

Kevät 2011

Tekijä: Joni Pieviläinen

Ohjaaja: Arto *****

Työn nimi	
Ohjelmoitava lakaisin	
Tekijä	Ohjaaja
Joni Pieviläinen	Arto *****
Aika	Sivumäärä
Kevät 2011	13
Tiivistelmä	
Rakensin opinnäytetyönä ohjelmoitavan lakaisimen. Halusin työssäni olevan ohjelmointia, rakentelua ja erilaisten kytkentöjen läpikäyntiä. Laitteen koodissa käytin eri paikoista löydettyjä pätkiä ja sovelsin niitä omaan työhöni. Robotti osaa liikkua itsenäisesti ja koodia vaihtamalla robottia pystytään ohjaamaan puhelimella tai tietokoneella. Lakaisimen rungon rakensin muovista ja metallista.	
Asiasanat	
Arduino, ohjelmointi, servo, vaihde, ROS	
Säilytyspaikka	
Koulutuskeskus Salpaus, Nastola	
Muut tiedot	

SISÄLLYSLUETTELO

1. JOHDANTO

2. OPINNÄYTETYÖN LÄHTÖKOHDAT

2.1. Arduino

2.2. Servo

2.3. ROS- KytKentä

2.4. SRF08

3. OPINNÄYTETYÖN SUUNNITELU

3.1. Koodin rakenne

3.2. KytKentöjen selitys

4. OPINNÄYTETYÖN TOTEUTUS

5. PÄÄTÄNTÖ

6. KODIT

1. JOHDANTO

Opinnäytetyönä rakensin lakaisin robotin. Robotti toimii ohjelmoitavalla Arduino piirillä ja ultraäänianturilla (srf08). Halusin työkseni haasteellisen ja monimutkaisen työn, joten valinta oli ohjelmointi ja robotiikka. Lähtökohtaisesti minulla oli mielessäni suunnitelma, mitä ohjelmakoodia tarvitsisin.

2. OPINNÄYTETYÖN LÄHTÖKOHDAT

Robottilla yleensä tarkoitetaan mekanista laitetta, joka kykenee liikkumaan fyysisessä maailmassa. Robotteja on alusta asti haluttu tekemään ihmisten töitä ”orjamaisesti”.

Tämän johdosta kaikkia mekaanisia laitteita ei luokitella roboteiksi, vaan robotilla pitää olla jotain ihmiseen viittaavia piirteitä.

Robottilla pitää olla aivot eli jonkin tyylinen prosessori, joka ohjaa tämän liikkeitä. Työssäni aivoina toimi Arduino.

2.1. Arduino

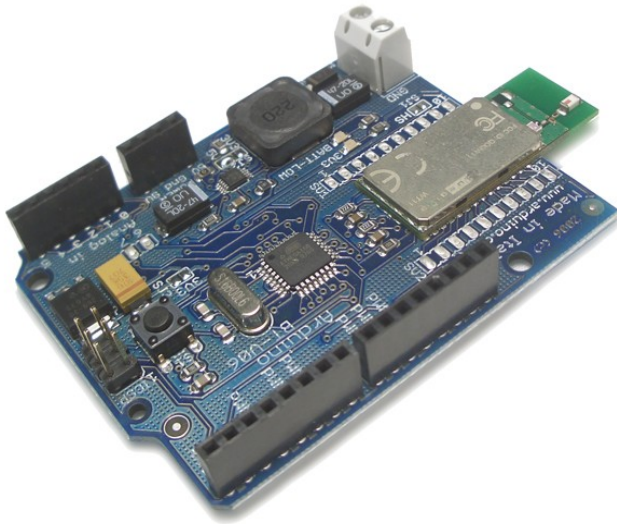
Arduino on italialainen ohjelmoitava piiri, joka on tarkoitettu harrastelijoille ja protomallien kehittelijöille. Arduinoa alettiin tuottamaan 2005 vuonna.

Luettelo ensimmäiset Arduino piireistä.

Arduino	Processor
Diecimila	ATmegaPPP
	ATmega168/3
Duemilanove	28P
Uno	ATmegaPPPP
Mega	ATmegaPPPP
Mega2560	ATmegaPPPP
Fio	ATmegaPPPP
	ATmega168 or
Nano	ATmega328
	ATmega168V
	or
LilyPad	ATmega328V

Työssäni käytin uutta Arduino BT:tä. (Kuva1)

Arduino BT on nimensä mukaisesti bluetoothilla toimiva. Tämä tarkoittaa sitä, että piiriin pitää ladata ohjelmat bluetoothin kautta ja piiri tarvitsee erillisen virtalähteen.



Kuva1. Arduni BT

Arduino BT:ssä 13 digitaalista lähtöä / tuloa ja 5 analogista lähtöä / tuloa. Arduinon virtalähteen jännitteen tulee olla alle 5,5V ja tasajännitettä.

2.2. Servo

Servo on RC laiteissa oleva voimakkaasti vääntävä moottori.(Kuva2) Servon voimakas vääntö tulee servossa olevista vaihte rattaista. Servon moottori on todella nopeakierroksinen ja vähävääntöinen, joten vaihteistolla saadaan asia muutettua päinvastoin.



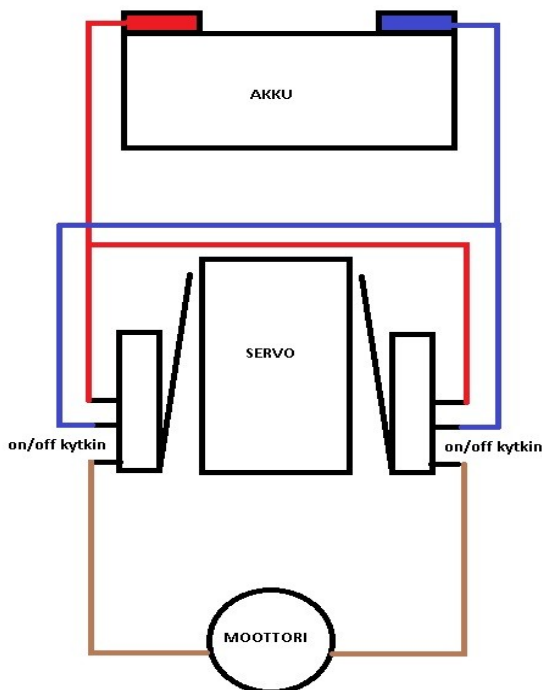
Kuva2. Perinteinen servo.

Servoja yleensä käytetään kaikissa RC harrastusajoneuvoissa. RC harrastuksessa servoa käytetään ohjauksessa, esim. autoissa renkaiden ohjaamisessa ja lentokoneissa siipien asentojen hakemiseen.

Servo ei pysty kääntymään 360°. Servon sisällä on laitettu tietynlaisia estoja. Näiden estojen poistamisella saadaan servo pyörimään rajattomasti ja sitä voidaan käyttää ajomoottorina.

2.3. ROS-KytKentä

ROS-kytkentä on tuttu Robosodan harrastajille. Perus ROS-kytkentä toteutetaan yhdellä servolla ja kahdella kytkimellä. Kytkennän tarkoitus Robosodassa on saada vastaanottimesta tuleva käsky servolle ja servon painaessa toista kytkintä saada sähkö kulkemaan ajomoottorille.



Kuva3. Robosodan versio ROS-kytkennästä.

Tarkemmin selitettynä

Kytkimillä on kaksi asentoa (1ja2).

Kun servo ei paina kumpaakaan kytkintä, tulee kummastakin linjasta silloin miinusta, joten moottori ei voi pyöriä. Kun servo kääntyy vasemmalle tai oikealle alkaa painetusta kytkimestä kulkea plus-jännitettä, joka aiheuttaa moottorin pyörimisen.

Kumpikaan kytkin ei voi samaan aikaan olla pohjassa johtuen servon perustoiminnasta.

2.4. SRF08

SRF08 on ultraäänianturi. Ultraäänianturi lähettää ihmiselle kuulematonta ääntä. Äänen osuessa esteeseen se kimpoaa takaisin anturiin ja anturi pystyy määrittämään niitä asioita, joita halutaan tietää. Työssäni käytin anturia etäisyyden mittaamiseen.



3. OPINNÄYTETYÖN SUUNNITTELU

Suunnittelun aloitin edellisessä jaksossa, jossa ensimmäistä kertaa käytimme Arduinoa. Tunneilta sain ohjelmointiin hyvän perusteen, jota käytin työssäni. Aloin suunnitella robottia, joka liikkuisi servoilla. Halusin työhöni kauko-ohjauksen, joten valintani oli Arduino BT. Halusin robotin tarvittaessa osaavan itse väistää esinettä, joten anturin lisääminen oli pakollinen. Haastavia kytkentäkaavoja ei ollut, käytännössä vain johtojen vetoja eri robotin osilta toiselle.

Aikataulu oli seuraava:

Viikko1	Tiedonhakua ja kytkentöjen selvittämistä
Viikko2	Tavaroiden tilaaminen ja hankkiminen
Viikko3	Koodin kirjoittaminen ja testaaminen protomalleilla
Viikko4	Robotin kokoaminen
Viikko5	Koodin sovittaminen robottiin ja raportin tekeminen

Jotkut työvaiheet veivät muita enemmän aikaa, mutta pysyin hyvin aikataulussa. Laitteen muoto vaihtui useasti robotin ominaisuuksista johtuen.

3.1. Koodien rakenteet

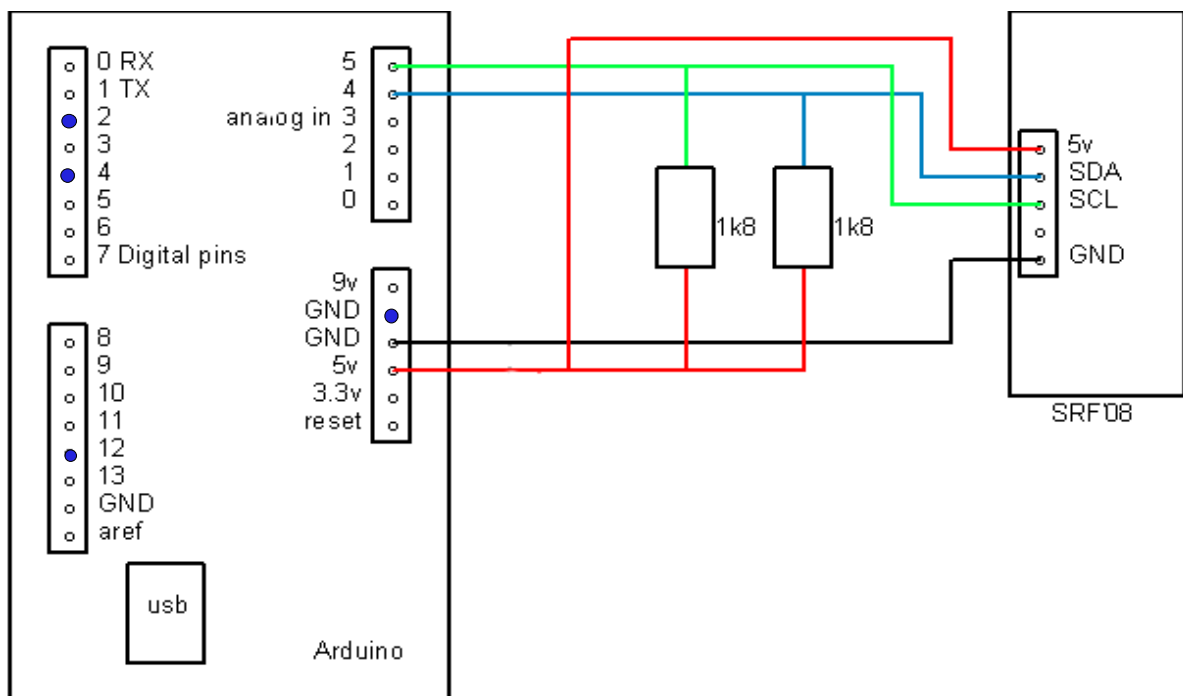
Ohjelmat ovat kirjoitettu Arduinon omalla ohjelmalla ja toiminta tarkastettu samalla ohjelmalla.

Tein kolme koodia. Kaksi robotille ja yhden puhelimelle.

Ensimmäisellä koodilla määritin robotin itsenäisen liikkumisen. Toisella koodilla bluetooth-ohjauksen. Kolmannella koodilla puhelimella ohjaamisen. Puhelimeen tarvisi hankkia Python-ohjelmisto, joka oli ilmainen. Puhelimelle tarkoitettua koodia piti muuttaa puhelimen Pythonille sopivaan muotoon, eli .py- loppuiseksi tiedostoksi.

Koodit liitteenä raportin loppuosaan.

3.2. Kytcentöjen selitys



SRF08 tarvitsee kaksi vastusta, jotta laite antaisi oikeita arvoja. Piiri kytketään Arduinon analogin pinneihin.

Arduinon pinnistä 5 → SRF08 SCL

Arduinon pinnistä 4 → SRF08 SDA

Arduinon pinnistä GND → SRF08 GND

Arduinon pinnistä 5v → SRF08 5v

Työssäni olevat kolme servoa kytkin digitaaliselle puolelle. Servot saavat näistä pinneistä ohjauksen. Siniset pallot osoittavat paikat.

Arduinon pinni 2 → vasen servo

Arduinon pinni 4 → oikea servo

Arduinon pinni 12 → ROS servo

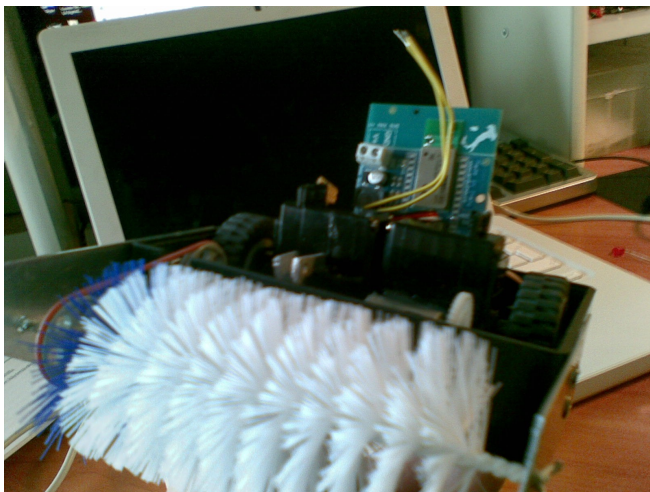
Servot ja Arduino saavat virtansa neljästä AA-paristosta.

Paristot ovat sarjaan kytkennässä, joista saadaan 6V. Tämä jännite ohjataan suoraan moottorille ja servoille. Arduinoon ei saanut laittaa näin suurta jännitettä, joten otin samasta paristo paketin puolesta välistä 3V eli kahdesta paristosta virran.

Moottorin kytkentä on muuten sama kuin ROS-kytkennässä, mutta yhdellä kytkimellä.

4. OPINNÄYTETYÖN TOTEUTUS

Aloitin robotin rakentamisen rungosta. Rungoksi valitsin mustan kovan muovikotelon. Koteloon piti tehdä pohjalle kaksi isoa aukkoa renkaita varten. Tämä tapahtui helposti poralla ja viilalla. Kun aukot olivat valmiit, aloitin servojen asentamisen paikoilleen. Servoille piti tehdä alusta muovista ja parista mutterista, jotta ne pysyisivät suorassa. Ennen servojen asentamista täytyi kiinnittää renkaat. Renkaat kiinnitin parilla pienellä ruuvilla kiinni. Kun servot ja renkaat olivat valmiina, laitettiin ne runkoon kiinni kuumaliimalla.



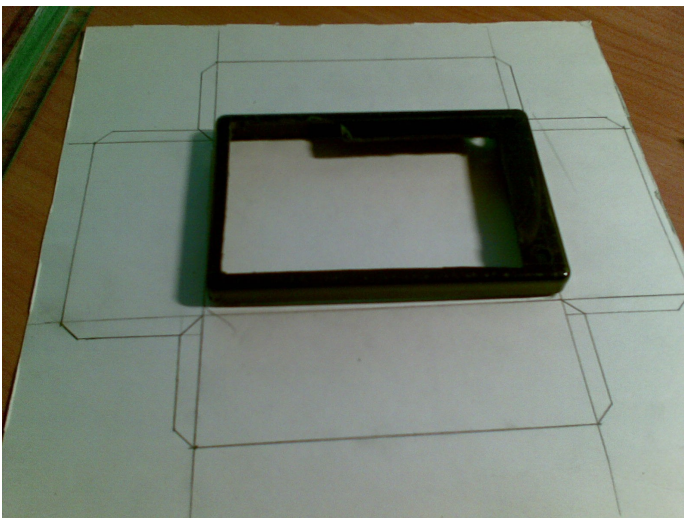
Lakaisimen vaihteen rakentaminen oli ongelmallinen. Ongelmaa tuotti, etten meinannut löytää sopivia rattaita ja moottoria. Löysin sopivat osat vanhasta radio-ohjattavasta autostani.

Isommassa rattaassa oli liian pieni reikä verrattuna minun robotissa olevaan vaihdeakseliin, joten jouduin suurentamaan reikää. Kiinnitin rattaan vaihdeakseliin.

Moottorin sovitin vaihderattaan mukaan, joka tässä tapauksessa oli vasempi laita. Moottorin johdotin ja liimasin kiinni.

Harjalle tuleva vaihde oli ongelmallinen. Mietin, millä saisin moottorin liikkeen tuotua harjalle. Ratkaisin ongelman parilla muovisella hihnapyörällä ja kuminauhalla. Jouduin tekemään suuremman reiän akselille ja kiinnitin pyörän liimalla.

Harjaksi valitsin pulloharjan, jota jouduin purkamaan saadakseni käytettyä harjan omaa vartta akselina. Harja meinasi lähteä purkautumaan liikaa, mutta sain sen estettyä pienellä kuumaliimalla tipalla. Sahasin harjasta kummankin pään tasaisiksi ja toiseen laitaan laitoin hihnapyörän, jotta sain harjan kiinni robottiin. Tarvitsin sivuille harjan pidittimet. Nämä sain helposti kahdesta alumiinin palasesta leikattua ja porattua. Palat kiinnitin robotin kylkeen ja laitoin harjan niiden väliin. Hihnapyörät pitivät laittoa kohdilleen ja kuminauha pyöryttää päälle. Parilla pyörytyksellä testasin toimivuuden. Testasin harjaa myös moottorilla, jotta sain tietää, pysyikö kuminauha moottorin tahdissa. Kuminauha pysyi paikoillaan.



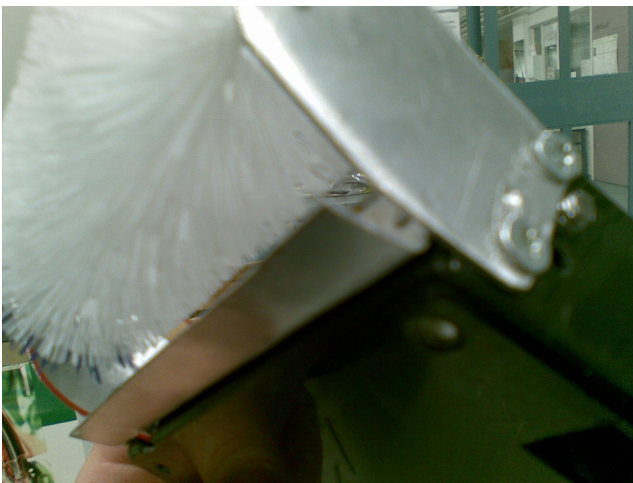
Kuva4. Kannen suunnittelu.



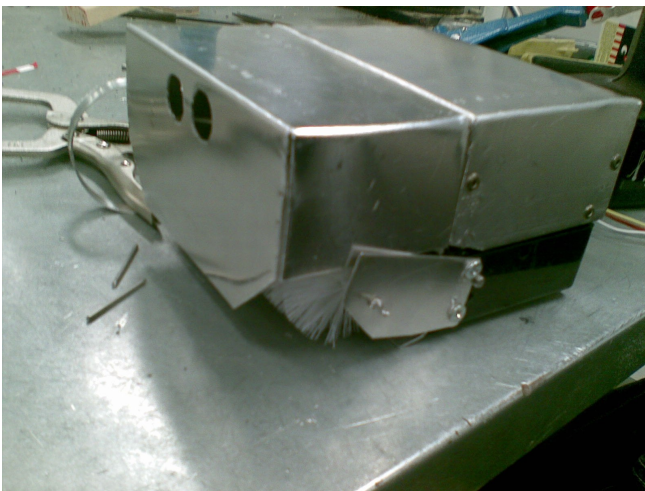
Kuva5. Pahvi mallin sovitus.



Kuva6. Mallin sovitus roboottiin.



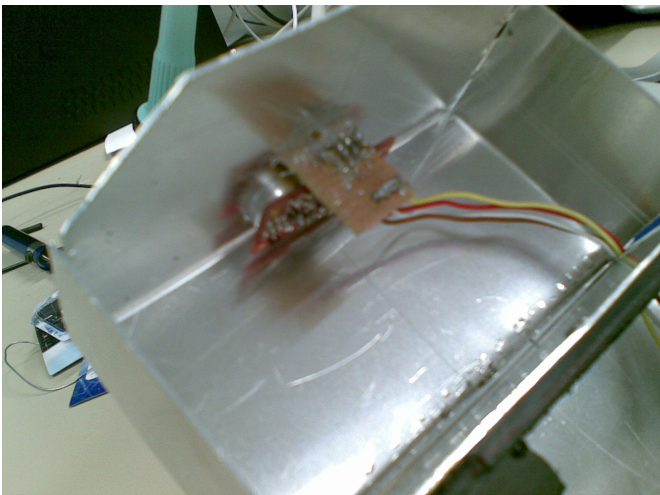
Kuva7. Auran kiinnitys.



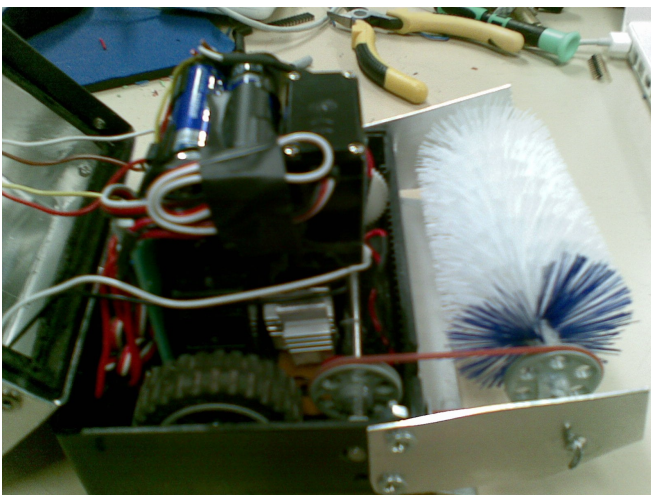
Kuva8. Kannen sovitus robottiin.



Kuva9. SRF08 varten tehdyt reiät.



Kuva10. Anturin kiinnitys.



Kuva11. Valmis robotin sisusta.



Kuva12. Virtakytkimien sommittelu.



Kuva13. Valmiin robotin kuva.

Harjan kiinnilaiton jälkeen täytyi saada robotille uusi kansi, koska muovikotelon kansi ei enää mahtunut sen päälle, joten suunnittelin uuden kannen. (Kuva 4,5,6)
 Leikkasin vanhasta kannesta keskiosan irti, jotta laitteisto pääsisi tulemaan siitä ylöspäin. Kannen reunat tarvitsin, jotta kansi menisi runkoon tukevasti kiinni. (Kuva5.)
 Jotta roskat tulisivat mukaan, tein pienen auran robotin ja harjan väliin. (Kuva7.)
 Kotelon sain tehtyä suunnitelmieni mukaisesti. SRF08 anturille tarvitsi tehdä reiät. (Kuva9.)
 Anturin liimasin kiinni ja johdot vedin Arduinolle. (Kuva10.)
 Laitoin Arduinon robotin takaosaan. Paristot päälle ja ROS-kytkennällä varustetun servon etuosaan. (Kuva11.)
 Virtakytkimille sommittelin paikat taakse ja robotin oikeaan takakulmaan laitoin harjan moottorin kytkimen, jotta harjaa voitaisiin käyttää toisessa koodissa. (Kuva12.)
 Lopuksi kansi kiinni ja työ arvioitavaksi. (Kuva13.)

5. PÄÄTÄNTÖ

Työ oli haasteellinen ja hain sitä tällä työllä. Robotti osasi toimia kummallakin koodilla ja liikkui sulavasti.

Koodeissa piti vähän muuttaa arvoja, jotta laitteen hienosäädöt osuivat kohdilleen.

Opin, että ei kannata olla liian ahne haasteellisuutta etsiessä. Tämä työ meinasi olla jopa minullekin liian haasteellinen.

6. KOODIT

SRF08 anturille

```
#include <Wire.h>
```

```
#include <Sonar_srf08.h>
```

```
Sonar_srf08 MySonar;
```

```
#define CommandRegister 0x00
```

```
int New_Address=248; //0xF8
```

```
#define ResultRegister 0x02
```

```
char unit='c';
```

```
float sensorReading=0;
```

```
float limitCm=10;
```

```
float distanceFront=0;
```

```
int servoRightPin=2;
```

```
int servoLeftPin=3;
```

```
void pulseServo(int servoPin, int pulseLenUs)
```

```
{
```

```
    digitalWrite(servoPin, HIGH);
```

```
    delayMicroseconds(pulseLenUs);
```

```
    digitalWrite(servoPin, LOW);
```

```
    delay(10);
```

```
}
```

```
void moveForward()
```

```
{
```

```
    for (int i=0;i<15;i++){
```

```
        pulseServo(servoLeftPin,1800);
```

```
        pulseServo(servoRightPin,500);
```

```
        Serial.println("eteen");
```

```
}  
}
```

```
void moveBack()  
{  
  for (int i=0;i<15;i++){  
    pulseServo(servoLeftPin,500);  
    pulseServo(servoRightPin,1800);  
    Serial.println("taakke");  
  }  
}
```

```
void moveLeft()  
{  
  for (int i=0;i<15;i++){  
    pulseServo(servoLeftPin,500);  
    pulseServo(servoRightPin,500);  
    Serial.println("vasen");  
  }  
}
```

```
void moveRight()  
{  
  for (int i=0;i<15;i++){  
    pulseServo(servoLeftPin,1800);  
    pulseServo(servoRightPin,1800);  
    Serial.println("oikee");  
  }  
}
```

```
void setup()  
{  
  MySonar.connect();  
  MySonar.changeAddress(CommandRegister, New_Address);  
  New_Address += 4;
```

```

pinMode(servoLeftPin, OUTPUT);
pinMode(servoRightPin, OUTPUT);
Serial.begin(115200);
}

float distanceCm(){
    MySonar.setUnit(CommandRegister, New_Address, unit);
    //pause
    delay(70);
    // set register for reading
    MySonar.setRegister(New_Address, ResultRegister);
    // read data from result register
    sensorReading = MySonar.readData(New_Address, 2);
    //print out distance
    // Serial.print("Distance: ");
    // Serial.print(sensorReading);
    // Serial.print(" inches");
    // Serial.println();
    //return (float) sensorReading;
}

void loop()
{
    distanceFront=distanceCm();
    if(distanceFront<limitCm){
        for (int i=0;i<=3;i++){
            moveBack();
        }
        for (int i=0;i<=3;i++){
            moveLeft();
        }
    }
    else{
        moveForward();
    }
}

```

```
if(Serial.available()){  
    char ch=Serial.read();  
    if('8'==ch){  
        moveForward();  
    }  
    else if('2'==ch){  
        moveBack();  
    }  
    else if('6'==ch){  
        moveRight();  
    }  
    else if ('4'==ch){  
        moveLeft();  
    }  
}  
}
```

Bluetoothilla ohjaaminen puhelimella

```
import appuifw, e32, key_codes, socket
```

```
btDeviceAddress='00:12:34:56:F0:0D' # Arduino BT. 'hcitool scan'
```

```
def quit():
```

```
    sock.close()
```

```
    appuifw.note(u"Bye! See you at www.sulautetu.fi!")
```

```
    appLock.signal()
```

```
##Main
```

```
canvas = appuifw.Canvas()
```

```
appuifw.app.body = canvas
```

```
canvas.bind(key_codes.EKey4, lambda: sock.send("4"))
```

```
canvas.bind(key_codes.EKey5, lambda: sock.send("5"))
```

```
canvas.bind(key_codes.EKey6, lambda: sock.send("6"))
```

```
#mobile phone numper pad isupsode down computer numpad:
```

```
canvas.bind(key_codes.EKey8, lambda: sock.send("2")) #back 8-2
```

```
canvas.bind(key_codes.EKey2, lambda: sock.send("8")) #forward 2-8
```

```
sock=socket.socket(socket.AT_BT, socket.SOCK_STREAM)
```

```
sock.connect((btDeviceAddress, 1)) # extra parenthesis for tuple
```

```
appuifw.note(u"Use 2468 to move, 5 to kick")
```

```
apuifw.app.exit_key_handler = quit
```

```
appLock = e32.Ao_lock()
```

```
appLock.wait()
```

Bluetooth osi Arduinossa

```
int servoOikeaPin=2;
int servoVasenPin=4;
int servoPotkuPin=12;
int kickNeutral=1800;
int kickKick=2500;
```

```
void pulseServo(int servoPin, int pulseLenUs)
{
  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulseLenUs);
  digitalWrite(servoPin, LOW);
  delay(20);
}
```

```
void calibratePotku()
{
  for (int i=0; i<=25; i++){
    pulseServo(servoPotkuPin, kickNeutral);
  }
}
```

```
void potku()
{
  for (int i=0; i<=10; i++){
    pulseServo(servoPotkuPin, kickKick);
  }
  delay(500);
  calibratePotku();
}
```

```
void moveEteen()
{
  for (int i=0; i<=25; i++){
```

```
    pulseServo(servoVasenPin, 1200);  
    pulseServo(servoOikeaPin, 1800);  
}  
}
```

```
void moveTaakse()  
{  
    for (int i=0; i<=25; i++){  
        pulseServo(servoVasenPin, 1800);  
        pulseServo(servoOikeaPin, 1200);  
    }  
}
```

```
void turnVasen()  
{  
    for (int i=0; i<=25; i++){  
        pulseServo(servoVasenPin, 1200);  
        pulseServo(servoOikeaPin, 1200);  
    }  
}
```

```
void turnOikealle()  
{  
    for (int i=0; i<=25; i++){  
        pulseServo(servoVasenPin, 1800);  
        pulseServo(servoOikeaPin, 1800);  
    }  
}
```

```
void setup()  
{  
    pinMode(servoVasenPin, OUTPUT);  
    pinMode(servoOikeaPin, OUTPUT);  
    pinMode(servoPotkuPin, OUTPUT);  
    calibratePotku();  
}
```

```
Serial.begin(115200);  
Serial.println("jalkapallo robootti");  
}
```

```
void loop()  
{  
  if (Serial.available()){  
    char ch=Serial.read();  
    if('8'==ch){  
      moveEteen();  
    } else if ('2'==ch){  
      moveTaakse();  
    } else if ('6'==ch){  
      turnOikealle();  
    } else if ('4'==ch){  
      turnVasen();  
    } else if ('5'==ch){  
      potku();  
    }  
  }  
}
```