



OULUN YLIOPISTO
UNIVERSITY of OULU

KONETEKNIIKAN KOULUTUSOHJELMA

ARDUINO -KONSEPTIN SOVELLUSESIMERKKEJÄ

Aappo Mustakangas

KANDIDAATINTYÖ

2016

Ohjaaja: Yrjö Louhisalmi

TIIVISTELMÄ

Arduino-konseptin sovellusesimerkkejä

Aappo Mustakangas

Oulun yliopisto, Konetekniikan koulutusohjelma

Kandidaatintyö 2016, 29 s.

Työn ohjaaja: Yrjö Louhisalmi

Arduino on avoimeen lähdekoodiin ja laitteistoon perustuva kehitysalusta. Arduino -konsepti koostuu erilaisista kontrollerikorteista ja suuresta kehittäjäyhteisöstä. Arduino-kortti on helposti ohjelmitava sille tarkoitettulla ohjelmalla ja sille löytyy runsaasti ohjeita, esimerkkejä ja projekteja. Tässä tekstissä kerrotaan joitakin esimerkkejä näistä töistä ja pyritään havainnollistamaan Arduinin mahdollisuuksia.

Asiasanat: Arduino, sovellusesimerkki, mekatroniikka

ABSTRACT

Application examples of Arduino concept

Aappo Mustakangas

University of Oulu, Degree Programme of Mechanical Engineering

Bachelor's thesis 2016, 29 p.

Supervisor: Yrjö Louhisalmi

Arduino is a development board based on open source code and hardware. Arduino concept is made up of different kinds of microcontrollers and large Internet community. Arduino board is easily programmable with computer program made for it. Lot of guides, examples and projects can be found from Internet and literature. This work shows some examples and attempts to give understanding of Arduino's possibilities.

Keywords: Arduino, Examples. Mechatronics

ALKUSANAT

Tähän työhön olen saanut opastusta kirjoittamiseen sisällön valitsimiseen Yrjö Louhisalmelta ja Juha Alatalolta. Yrjö Louhisalmi auttoi myös oikolukemisessa.

Oulu, 11.4.2016

™

Aappo Mustakangas

SISÄLLYSLUETTELO

JOHDANTO	6
Mikä on arduino?	7
1.1 Arduinon historiaa.....	9
Esimerkkisovelluksia	10
2.1 Blink.....	10
2.2 Ultimate Machine	10
2.3 Verhojen aukaisin.....	13
2.4 Arduino PID-säädin.....	17
2.5 Kivi-Paperi-Sakset-robotti.....	19
1.7 Arduino-ohjattu Segway-kopio	25
YHTEENVETO	28
LÄHDELUETTELO.....	29
LIITE/LIITTEET	

JOHDANTO

Tässä työssä esitellään Arduino ja kerrotaan erilaisia sovellusesimerkkejä sen käyttömahdollisuuksista. Tarkoituksena on lisätä tekniikan ja mekatroniikan alan opiskelijoiden tietoa Arduinon mahdollisuuksista ja rajoituksista havainnollistavien esimerkkien avulla.

Arduino on loistava kehitysalusta elektronisille ja mekatronisille sovelluksille. Siihen voidaan liittää antureita ja aktuaattoreita helposti ja ohjelmointi tehdään helposti käytettävällä ohjelmistolla.

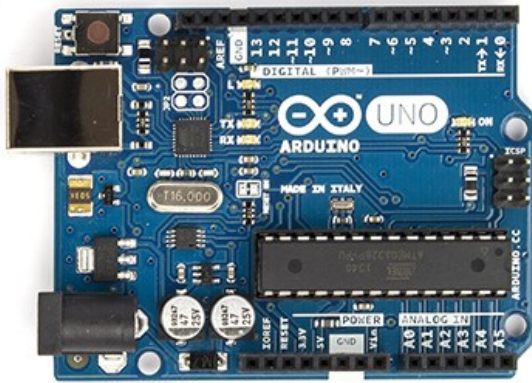
Esimerkeissä on pyritty käyttämään ominaisuuksia, jotka ovat yleisiä monissa mekatronisissa sovelluksissa, kuten sensoreiden lukeminen ja aktuaattoreiden ohjaaminen. Esimerkiksi DC-moottorin ohjaaminen onnistuu Arduinolla usealla eri tavalla.

MIKÄ ON ARDUINO?

Arduino on avoimeen lähdekoodiin ja laitteistoon perustuva mikrokontrolleri sekä kehitysalusta, jota käytetään kun halutaan saada tietojärjestelmä kommunikoimaan ympäristönsä kanssa. Laitteeseen voidaan varsin helposti liittää erilaisia sensoreita ja aktuaattoreita ja se pystyy suorittamaan yksinkertaista koodia. Toisin kuin esimerkiksi Rasperry Pi-mikrotietokone, Arduino on helpompi liittää erilaisiin elektronisiin komponentteihin, mutta ei pysty suorittamaan yhtä monimutkaista koodia. Ohjelmoitavaan logiikkaan verraten Arduino on paljon helpompi ohjelmoida monimutkaisiinkin toimiin ja pystyy käyttämään monimutkaisempia laitteita ja antureita. Laite voidaan ohjelmoida yleisimmillä käyttöjärjestelmillä ja laite itsessään pystyy kommunikoimaan hyvin erilaisten ohjelmien kanssa USB:n välityksellä. Ohjelmointiin käytetään C++ pohjaista Arduinon omaa yksinkertaista ohjelmistoa ja koodi syötetään USB:n avulla tietokoneesta.

Arduinossa on merkittävää sen yhteisö. Internetistä löytyy suuri määrä harrastelijoitten ja ammattilaisten kirjoittamaa koodia sekä projekteja, jotka ovat vapaasti käytettävissä. Vapaan lähdekoodin ansiosta parannuksia vanhaan koodiin on helppo tehdä ja levittää eteenpäin. Esimerkiksi pienelle LCD-näytölle on olemassa valmiit kirjastot ohjelmoimista varten. Arduinon helppo käyttöönotettavuus ja monipuolisuus antavat laitteelle suuren ja laaja-alaisen käyttäjäkunnan, jolta voi saada apua monenlaisiin ongelmiin

Tässä työssä on esitelty Arduino Unoa, jossa on 14 digitaalista pinniä ja 6 analogista pinniä. Sen prosessorina toimii ATmega328P. Arduino-alustoja löytyy monenlaisia ja uusia malleja ilmestyy tasaiseen tahtiin. Kaikki voidaan kuitenkin ohjelmoida samalla ohjelmistolla ja koodilla. Alustoja on esimerkiksi Arduino Mega jossa on yhteensä 70 pinniä ja tehokkaampi prosessori tai Arduino Yün jossa on Ethernet ja WiFi -tuki ja Atheros prosessori, joka tukee Linuxia. Arduinolla voidaan tehdä projekteja ilman erityistä osaamista. Kuitenkin jonkinlainen osaaminen elektroniikasta tai ohjelmoinnista on avuksi, kun halutaan kokeilla uusia ideoita.



Kuva 1: Arduino Uno. (<https://www.arduino.cc/en/Main/ArduinoBoardUno>)

Arduino soveltuu hyvin kehitysluontoisiin projekteihin. Sen monipuolisuuden ansiosta sitä pystytään käyttämään ohjaavana piirinä monenlaisille aktuaattoreille, se voi välittää tietoa sensoreilta tietokoneelle, toimia ohjaimena tai se voi toimia itsenäisesti lukien sensoridataa ja sen mukaisesti ohjata aktuaattoreita. Tiettyä käyttötarkoitusta varten Arduinolle löytyy halvempia vaihtoehtoja tai siitä voidaan todennäköisesti rakentaa sellaisia. Arduinon mikrokontrollerikortteja ei ole rakennettu erityisen kestäväksi ja niiden kunnollinen suojaaminen ympäristöltä voi olla työlästä. Myös Arduinon analoginen resoluutio 10 bittiä, on vaatimaton. Ideoiden testaamiseen tai kehittämiseen Arduino on kuitenkin hyvä valinta monipuolisuutensa ansiosta. Arduino-kontrollerikortti voidaan yleensä siirtää yhdestä käyttötarkoituksesta toiseen ilman esteitä. Arduinon löytyy myös kilviksi (engl. shield) nimettyjä moduuleja, jotka kytetään Arduinon päälle. Moduulit antavat Arduinolle helposti uusia ominaisuuksia. Esimerkiksi WiFi tai Bluetooth kommunikaatio Arduinon kanssa voidaan luoda kilpiä käyttämällä.

Laitteiston avoimuus mahdollistaa järjestelmien muokkauksen helposti ja verkosta löytyy suuri määrä valmiita koodeja ja kirjastoja erilaisiin tehtäviin. Erityisesti kirjastot ovat kätevä työkalu laajentamaan Arduinon käyttömahdollisuuksia. Kirjastoilla tarkoitetaan tässä yhteydessä valmiiksi kirjoitettuja ohjelmia, joita voidaan ladata koodiin tarvittaessa. Usein kirjastoja tarvitaan, kun halutaan ohjata jotain monimutkaisempaa laitetta, kuten esimerkiksi LCD-näyttöä. Arduinon omilta Internet-sivuilta löytyy suuri määrä ohjeita ja esimerkkejä Arduinon ohjelmoimiseen, mikä auttaa kokematonkin käyttäjä pääsemään alkuun. (<http://www.arduino.cc/>)

1.1 Arduinon historiaa

Arduinon kehitysryhmässä oli alkuperin viisi henkilöä Massimo Banzi, David A. Mellis, David Cuartielles, Gianluca Martino ja Tom Igoe. Arduino julkaistiin 2005 halpana oppityökaluna Massimo Banzin oppilaille Italialaisessa Interaction Design Institute Irveassa (IDII). David A. Mellis on saman oppilaitoksen entinen oppilas ja Arduino ohjelmiston pääsuunnittelija. Gianluca Martino toimi elektroniikkainsinöörinä ja David Cuartielles tietoliikenneinsinöörinä. Tom Igoe on tietojenkäsittelytieteiden professori ja tuli hiukan myöhemmin mukaan ryhmään. Arduino-projekti sai paljon vaikutteita Processing-nimisestä ohjelmointikielestä sekä Wiring-nimisestä prototyyppialustasta. IDII lopetti toimintansa varojen puutteeseen ja se sai tekijät päättämään, että projektin lähdekoodi ja laitteisto annetaan vapaaseen levitykseen. He käyttivät Creative Commons -tyyppistä lisenssiä ja aloittivat piirilevyjen massavalmistuksen Irvean lähellä sijaitsevassa laitoksessa. Arduino sai nimensä paikalliselta baarilta, joka puolestaan sai nimensä Italian entiseltä kuninkaalta. (<http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>)

Arduinon nimestä aiheutui lakikiista piirilevyjä valmistaneen yhtiön Arduino SRL ja arduino.cc sivustoa pitäneen yhdysvaltalaisen yhtiön Arduino LLC välillä. Arduino LLC on alkuperäisen 5 henkisen tutkimustiimin yhtiö. Koska laitteisto on vapaata, kuka tahansa voi valmistaa kyseessä olevia controllerikortteja. Arduino nimi kuitenkin rekisteröitiin, jotta saatiin paremmin valvottua piirilevyjen laatua omassa tuotannossa. Yhtiöillä oli erimielisyyksiä Arduinon kehittämissuunnasta ja kumpikin halusi säilyttää tunnetun nimen. Tämän työn kirjoitushetkellä Arduino LLC myy tuotteitaan Arduino nimellä Yhdysvalloissa ja Genuino nimellä muualla maailmassa. Arduino LCC omistaa myös arduino.cc sivuston, joka on tunnetumpi harrastajien keskuudessa. A Arduino SRL myy tuotteitaan Arduino nimellä ja omistaa Arduino.org sivuston. Laitteet ja ohjelmistot ovat toistensa kanssa yhteensopivia mutta tulevaisuudessa tämä voi muuttua. (<http://hackaday.com/2015/02/25/arduino-v-arduino/> ja <http://hackaday.com/2015/03/12/arduino-v-arduino-part-ii/>)

ESIMERKKISOVELLUKSIA

2.1 Blink

Blink on yksinkertainen ohjelma, joka vilkuttaa Arduinon piirilevyn lediä. Ohjelma toimii hyvin tietokoneen ja Arduinon välisen yhteyden kokeilemiseen. Blink koodi on yksi Arduinon ohjelmointiympäristön esimerkkikoodista. Se löytyy valikosta tiedosto - esimerkit – Blink. Muuttamalla koodissa olevia delay -arvoja voidaan muuttaa vilkutuksen tahtia.

Koodi:

```

/*

Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second

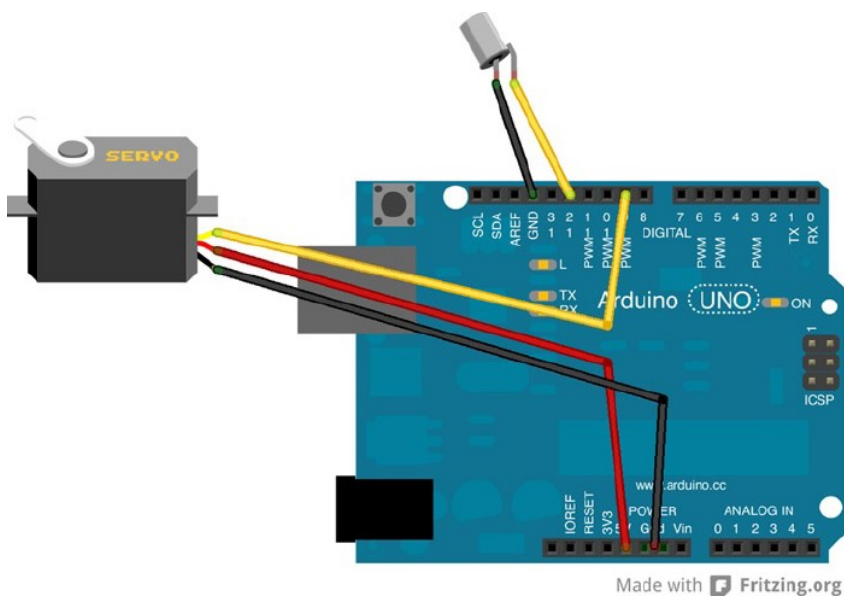
```

2.2 Ultimate Machine

Ultimate machine eli ultimaattisen koneen ainoa tarkoitus on kytkeä käyttäjän ON asentoon kytkemä vipukytkin OFF-asentoon mekaanisella varrella. Laite tarvitsee siis vipukytkimen,

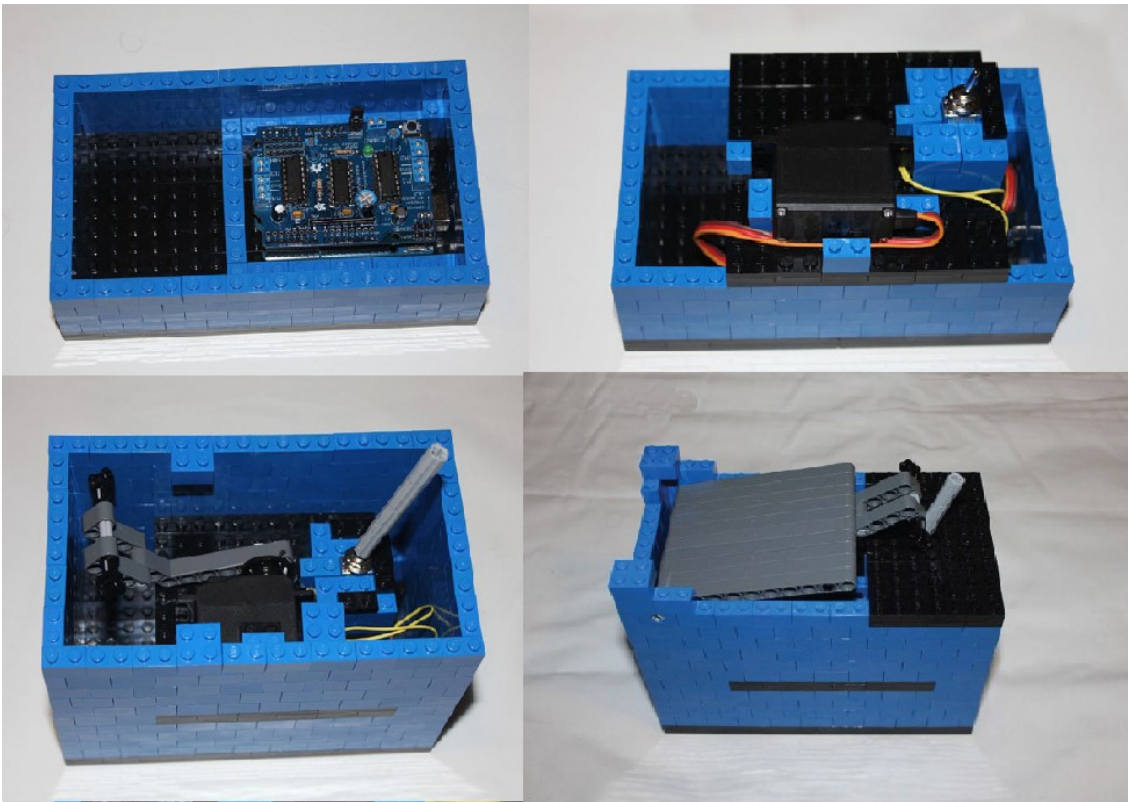
servomoottorin ja rungon. Runko ja vivun jatkovarsi rakennetaan legoista. Esimerkki löytyy kirjasta *Arduino and LEGO projects* (Jon Lazar 2013).

Servomoottorin käyttövirta voidaan ottaa suoraan Arduinosta, mikäli Arduino on kytketty virtalähteeseen eikä moottori vedä yli 20 mA virtaa. Moottorin ohjaus kytketään porttiin, josta saadaan PWM-signaalia. Koodissa tähän on määrätty pinni 9. Vipukytkin voidaan lukea digitaalipinnin arvona. Kun vipukytkin kytketään päälle, se maadoittaa digitaalipinnan ja pinni saa luettaessa arvon LOW. LOW-arvo tarkoittaa, että koodi antaa ajokäskyn servomoottorille.



Kuva 2: Kytkenät. (Lazar J, (2013) Arduino and LEGO project, New York, Apress.)

Rakennettava runko tarvitsee tukevan paikan servomoottorille ja kytkimelle. Servomoottoriin kiinnitetään varsi, joka painaa vipukytkintä kääntyessään. Vipukytkimelle on myös hyvä lisätä jatkovarsi, jotta se ei ole liian jäykkä moottorille. Runkoon laitetaan kansi, josta servomoottorin työntövarsi ilmaantuu. Kuvassa 3 on yksi mahdollinen tapa rakentaa runko. Käytettävissä olevista resursseista riippuen voidaan mallia muokata varsin vapaasti, kunhan saadaan aiemmin mainitut vaatimukset täytettyä.



Kuva 3: Lego runko. (Lazar J, (2013) *Arduino and LEGO project*, New York, Apress.)

Arduino tarvitsee servomootorin ohjaamiseen ohjelmakirjaston. Se löytyy Arduinon valmiista kirjastoista ja se otetaan käyttöön koodin alussa.

```
#include <Servo.h>

// DC hobby servo
Servo servol;

// Switch connected to digital pin 2
int SwitchPin = 2;

void setup() {
  // turn on servo
  servol.attach(9);

  // sets the digital pin 2 as input
  // and enables pullup resistor
  pinMode(SwitchPin, INPUT_PULLUP);
}

void loop() {
  // read the input pin
  int val = digitalRead(SwitchPin);

  // test if switch has been triggered
  if (val == LOW) {
    servol.write(115);
    delay(250);
    servol.write(0);
  }
  delay(100);}
```

2.3 Verhojen aukaisin

Tämä on esimerkkisovellus, jossa käytetään Arduinoa aukaisemaan huoneen verhot. Ohjaimena toimii kaksi nappia, joista toisella saadaan moottori päälle ja pois. Toisella vaihdetaan suuntaa. Työssä tarvitaan H-silta, sähkömoottori, 9V paristo, 2 mikrokytkintä sekä kaksi 10 k Ω vastusta. Tarvitaan myös erinäisiä tavaroita laitteen liittämiseksi verhoihin. Toteutustavasta riippuen tarvittavat esineet voivat vaihdella paljonkin. Työn pohjana on käytetty Arduino projects book:in zeotrope esimerkkiä. (Fitzgerald S & Shiloh M (2013) Arduino Project Book, Toinen uusintapainos, Italia. sivut 102–112.)

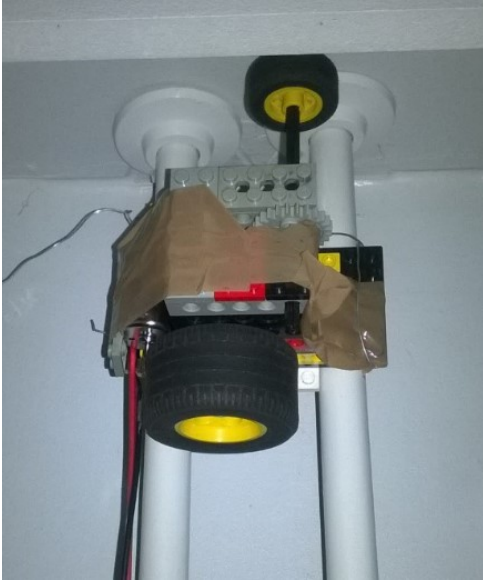
Liikuteltava verho kiinnitetään raiteeseen. Kiinnitetään narulenkki raiteen päistä niin, että se liikkuu vapaasti. Käytetään tässä paperiliittimiä kuvan 4 mukaan. Paperiliittinten pitoa raiteeseen voidaan lisätä esimerkiksi asettamalla kuminauha raiteen ja paperiliittimen leukojen väliin. Kiinnitetään narujen lenkit keskimmäisiin verhoja piteleviin leukoihin. Näin saadaan narua vetämällä verhot aukenemaan ja sulkeutumaan



Kuva 4: Narun liittäminen.

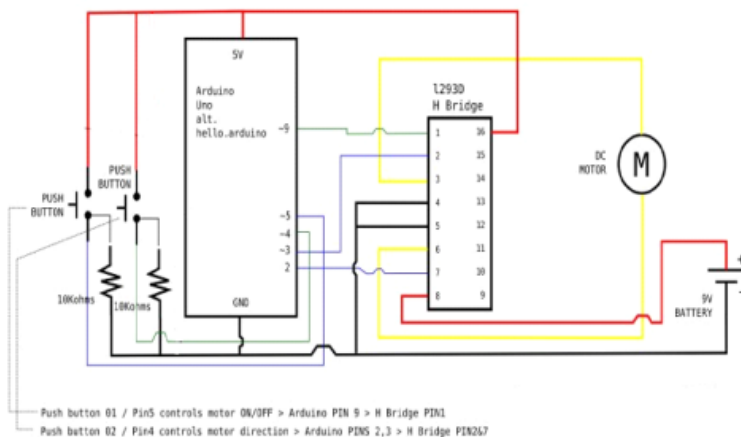
Sijoitetaan voimalähde toiseen päähän raidetta. Tässä käytetään Lego-rengasta kitkapyöränä. Käytössä oleva sähkömoottori on liian nopea ja siitä saadaan liian vähän vääntömomenttia tähän tarkoitukseen. Tämän takia rakennetaan Legoista myös vaihde, jossa sähkömoottorilla välitetään voima toisen kitkapyörän kautta narua vetävälle kitkapyörälle. Vaihteen suhde on 1:4,5. Näin sähkömoottorista saadaan riittävä vääntömomentti verhon vetämiseen ja lopullinen nopeus saadaan

säädettyä ohjelmakoodilla. Lego systeemin kiinnittämisessä seinään joudutaan käyttämään luovuutta kulloisenkin tilanteen mukaan. Tässä vaihde on kiinnitetty lämpöpatterien putkiin niin, että ylempi legopyörä on raiteen tuntumassa ja vetolangan tasalla.



Kuva 5: Legovaihde ja DC-moottori.

Työssä käytetään H-siltaa, joka toteutetaan mikrosirulla. Sen avulla moottori saadaan pyörimään molempiin suuntiin Arduinon PWM-signaaliilla. Silta huolehtii myös siitä, että moottori ottaa virtansa paristosta eikä Arduinon USB liitännästä. Jotkin DC-moottorit käyttävät niin suuria virtoja, että Arduinon elektroniikka rikkoontuu. Kytkennässä on 2 mikrokytkintä, joista toinen huolehtii moottorin käynnistämisestä ja toinen moottorin suunnasta.



Kuva 6: Kytentäkaavio. (Fitzgerald S & Shiloh M (2013) *Arduino Project Book, Toinen uusintapainos, Italia*)(Muokattu)

```

const int controlPin1 = 2; // connected to pin 7 on the H-bridge
const int controlPin2 = 3; // connected to pin 2 on the H-bridge
const int enablePin = 9; // connected to pin 1 on the H-bridge
const int directionSwitchPin = 4; // connected to the switch for direction
const int onOffSwitchStateSwitchPin = 5; // connected to the switch for turning the motor on and off

// create some variables to hold values from your inputs
int onOffSwitchState = 0; // current state of the On/Off switch
int previousOnOffSwitchState = 0; // previous position of the on/off switch
int directionSwitchState = 0; // current state of the direction switch
int previousDirectionSwitchState = 0; // previous state of the direction switch

int motorEnabled = 0; // Turns the motor on/off
int motorSpeed = 200; // speed of the motor
int motorDirection = 1; // current direction of the motor

void setup() {
  // initialize the inputs and outputs
  pinMode(directionSwitchPin, INPUT);
  pinMode(onOffSwitchStateSwitchPin, INPUT);
  pinMode(controlPin1, OUTPUT);
  pinMode(controlPin2, OUTPUT);
  pinMode(enablePin, OUTPUT);

  // pull the enable pin LOW to start
  digitalWrite(enablePin, LOW);
}

void loop() {
  // read the value of the on/off switch
  onOffSwitchState = digitalRead(onOffSwitchStateSwitchPin);
  delay(1);

  // read the value of the direction switch
  directionSwitchState = digitalRead(directionSwitchPin);

  // read the value of the pot and divide by 4 to get
  // a value that can be used for PWM

  // if the on/off button changed state since the last loop()
  if (onOffSwitchState != previousOnOffSwitchState) {
    // change the value of motorEnabled if pressed
    if (onOffSwitchState == HIGH) {

```

```
motorEnabled = !motorEnabled;
}
}

// if the direction button changed state since the last loop()
if (directionSwitchState != previousDirectionSwitchState) {
// change the value of motorDirection if pressed
if (directionSwitchState == HIGH) {
motorDirection = !motorDirection;
}
}

// change the direction the motor spins by talking
// to the control pins on the H-Bridge
if (motorDirection == 1) {
digitalWrite(controlPin1, HIGH);
digitalWrite(controlPin2, LOW);
}
else {
digitalWrite(controlPin1, LOW);
digitalWrite(controlPin2, HIGH);
}

// if the motor is supposed to be on
if (motorEnabled == 1) {
// PWM the enable pin to vary the speed
analogWrite(enablePin, motorSpeed);
}
else { // if the motor is not supposed to be on
//turn the motor off
analogWrite(enablePin, 0);
}
// save the current On/Offswitch state as the previous
previousDirectionSwitchState = directionSwitchState;
// save the current switch state as the previous
previousOnOffSwitchState = onOffSwitchState;
```


2.4 Arduino PID-säädin

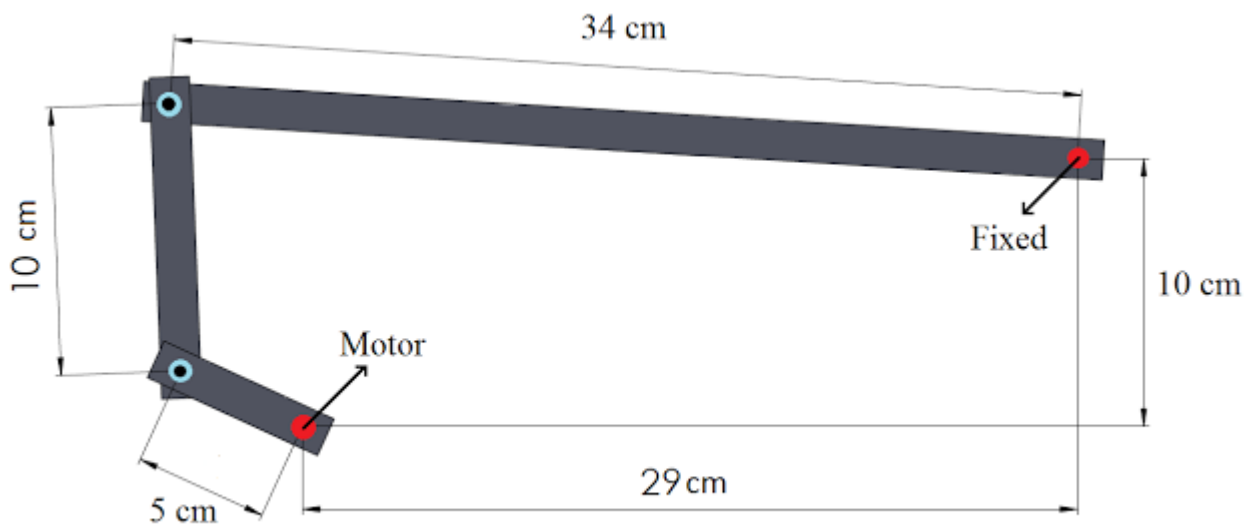
Tämän Esimerkin on tehnyt Ali M. Ibrahim blogiinsa ja se löytyy osoitteesta <http://mechatronicstutorials.blogspot.fi/2014/07/balancing-of-ball-on-beam-using-arduino.html>.

Työssä käytetään kirjastoa servomoottorin ohjaamiseen ja toista kirjastoa PID- säädön suorittamiseen. Kirjastot ja koodi löytyvät blogista ja sieltä löytyy myös työssä hyödyllinen Matlab-scripti.

Työssä tasapainotellaan palloa kourussa. Arduino toimii PID ohjaimena, joka ohjaa servomoottoria, joka puolestaan kallistaa kourua. Arduino saa pallon etäisyystiedot ultraäänianturilta. Arduino voi myös syöttää etäisyystiedot Matlab:iin. Blogista löytyy Matlab tiedosto, joka rekisteröi Arduinon antaman etäisyystiedon ja piirtää siitä kaavion reaaliajassa.

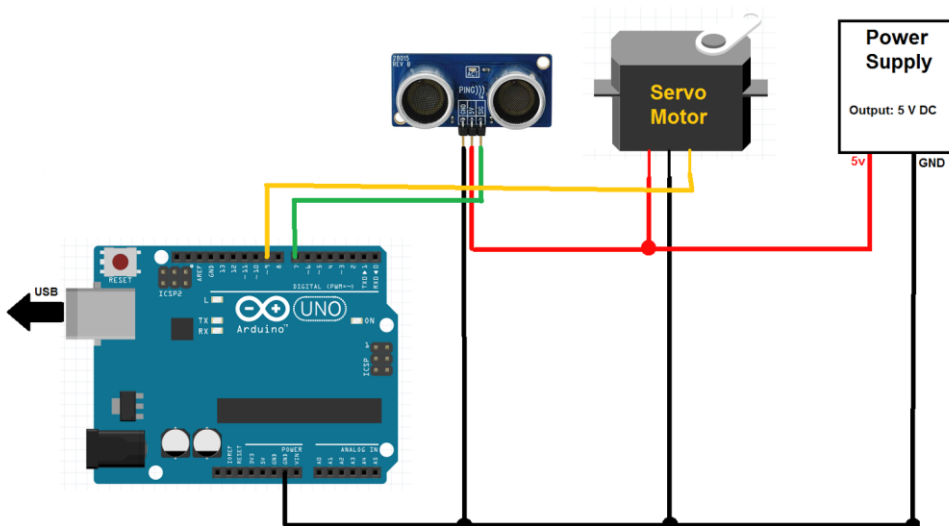
Työssä tarvitaan ultraäänianturi ja servomoottori. Käytössä on PING))) ultraäänianturi, jossa saadaan ultraäänipulssi ja äänen kaiun rekisteröinti saman pinnan kautta. Molemmissa tapauksissa annetaan ultraäänipulssi, jonka jälkeen kuunnellaan kaiku. Pulssin ja kaiun välinen aika voidaan muuttaa koodissa etäisyystiedoksi.

Palkin tukirakenteiden on esitetty kuvassa 7. Osien kokoa voidaan muuttaa tarvittaessa, mutta samalla pitää muuttaa PID säätimen K_p , K_d ja K_i arvoja koodissa. Kouru liitetään toisesta päästä tukevasti pisteeseen, jossa se voi heilua. Toinen pää on kiinni servomoottorin ohjaamassa vipusysteemissä. Kourun on toimiva, jos voidaan laittaa pallo sen päälle niin, että se ei pääse karkaamaan sivusuunnassa, mutta pyörii vapaasti pituussuunnassa.



Kuva 7: Palkkirakenne. (<http://mechatronicstutorials.blogspot.fi/2014/07/balancing-of-ball-on-beam-using-arduino.html>)

Arduinoon kytketään servomoottori ja ultraäänianturi (Kuva 8). Käytettävä servomoottori määrää tarvitaanko ulkoinen jännitelähde. Jos virrantarve ylittää 20 mA, on syytä käyttää ulkoista jännitelähdettä.



Kuva 8: Kytkentäkaavio. (<http://mechatronicstutorials.blogspot.fi/2014/07/balancing-of-ball-on-beam-using-arduino.html>)

2.5 Kivi-Paperi-Sakset-robotti

Arduinolla ohjataan käsiproteesia pelaamaan kivi-paperi-sakset peliä. Nappia painamalla robotti heiluttaa kättään kolmesti ja asettaa sitten käden johonkin kolmesta asennosta. Kätenä käytetään 3D-tulostettua Raptor Hand nimistä käsiproteesia ja tämä malli on vapaasti ladattavissa joko Thingiverse Internet-sivustolta tai Raptor Hand:in omilta Internet-sivuilta. Mallia skaalatiin ennen tulostusta pienemmäksi.



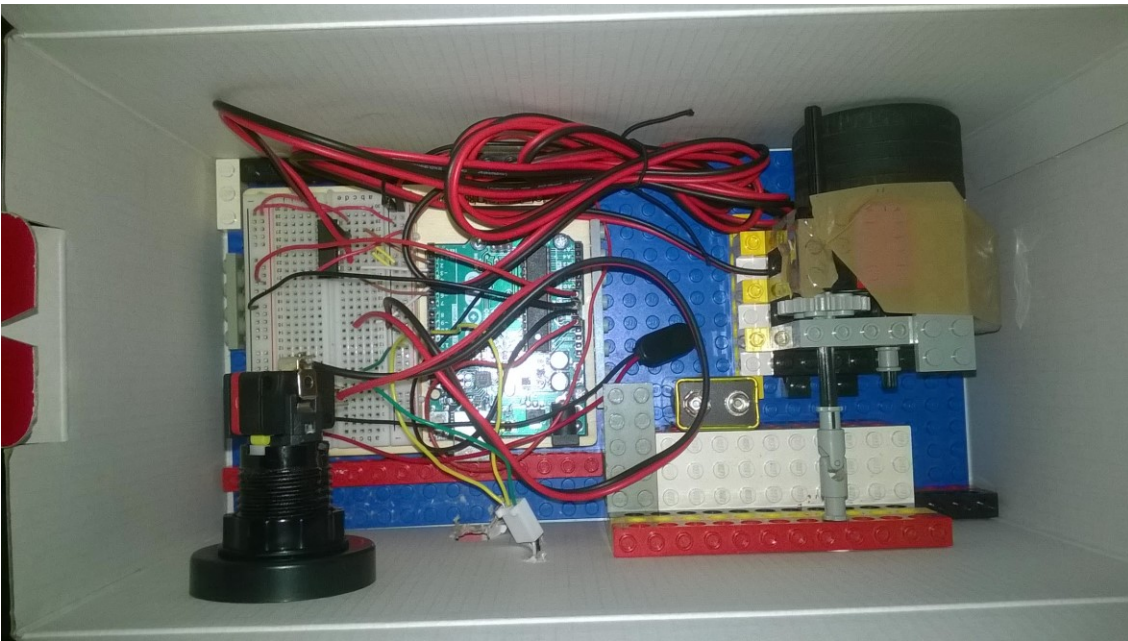
Kuva 9: 3D-tulostettu ja kasattu käsi.

Käsi kootaan tulostetuista osista ja siihen liitetään narut sekä kuminauhut Raptor Hand sivulta löytyvien ohjeiden mukaan. Naruista vetämällä sormet koukistuvat ja kuminauhut palauttavat sormet takaisin ojennettuun asentoon. Sormia ohjaavat narut sidotaan servomoottorin lapaan. Servomoottori puolestaan laitetaan pahvista tehtyyn ”käsivarteen”. Kivi-paperi-sakset peliin riittää, että neljä sormista liikkuu. Sormien narut kiinnitetään siten, että yhteen suuntaan pyöriessään servomoottori vetää vain pikkusormen ja nimettömän naruista. Toiseen suuntaan liikkeessään servomoottori vetää kaikista sormien naruista. Kuvassa 10 näkyy välivaihe, jossa servomoottori vetää kahta narua kumpaakin suuntaan pyöriessään.



Kuva 10: Somien narujen kiinnittäminen servomoottoriin.

Käsivarsi kiinnitetään keskeltä akseliin ja tasapainotetaan siihen. Työssä käytetään verhojen avaajassakin käytettyä Legoista tehtyä vaihdetta tämän akselin heiluttamiseen. Lisäksi tarvitaan riittävän tukeva rakennelma käsivarren ja käden yhdistelmän kannattelemiseen. Legoista saadaan hyvä tukirakennelma ja akseli käsivarrelle. Tämä rakennelma koteloidaan vielä pahvilaatikkoon jolloin saadaan helposti siirreltävä ja kestävä rakennelma.



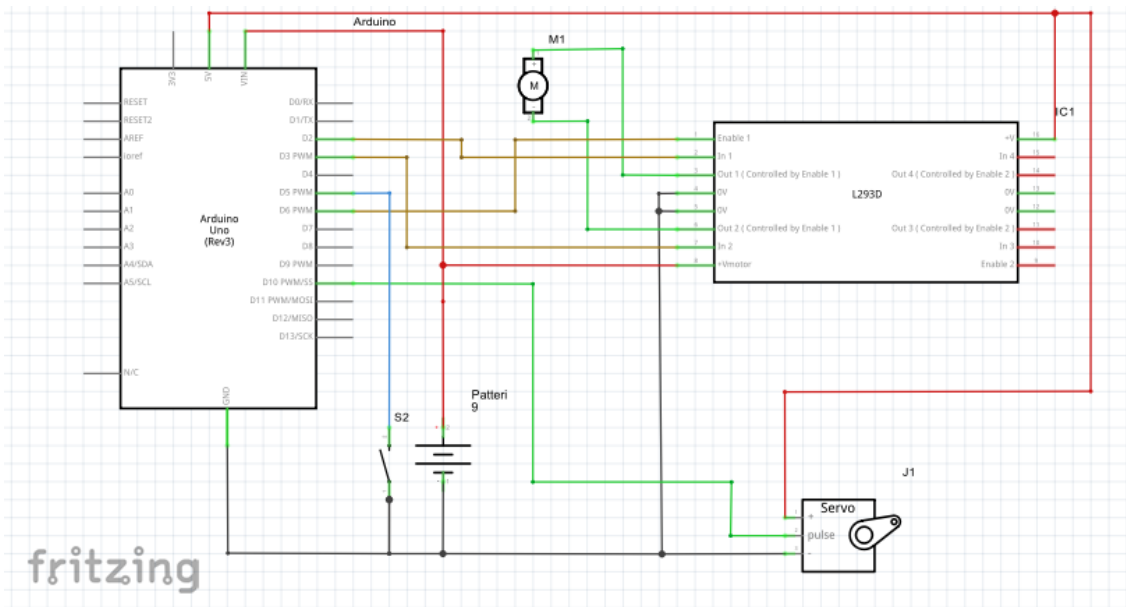
Kuva 11: Kotelon sisältö.



Kuva 12: Robotti kotelon ulkopuolelta.

Legovaihteessa on DC-moottori, jota ohjataan H-sillan kautta toimimaan halutulla tavalla. Virta otetaan sekä moottorille, että Arduinolle 9V paristosta. Laitetta ohjataan nappia painamalla ja servomoottori liikuttaa sormia.

Tässä työssä nappi toimii Arduinon sisäisen pull-up vastuksen avulla active-low-kytkimenä. Nappi kytketään maahan ja pinniin 5. Servomoottorin ohjaus kiinnitetään pinniin 10. H-silta kiinnitetään Arduinon Pinneihin 2, 3 ja 6 kuvan osoittamalla tavalla.



Kuva 13: Kiv-Paperi-Sakset robotin kytkennät.

Koodissa pinni muuttuu aktiiviseksi, kun se kytketään maahan. Koodi suorittaa up ja down funktioita, jotka saavat käsivarren heilahtamaan kolmesti ylös ja alas. Delay arvoja muuttamalla saadaan käden liikkeen nykivyyttä muutettua. Sen jälkeen ajetaan funktio, joka arpoo yhden numeron väliltä 0-2 ja saadun numeron mukaan asettaa jonkin määrätystä asennoista servomoottoriin. Koodissa on myös kommento, joka palauttaa arvoitun numeron USB:n kautta takaisin tietokoneelle. Sitä voidaan käyttää testaamaan toimiiko laite oikein, mutta ei ole välttämätön toiminnan kannalta.

Koodi

```

#include <Servo.h> //Servo kirjasto
const int controlPin1 = 2; // connected to pin 7 on the H-bridge
const int controlPin2 = 3; // connected to pin 2 on the H-bridge
const int enablePin = 6; // connected to pin 1 on the H-bridge
const int nappiPin = 5; //ohjausnappi

// muuttujat
int nappiTila = 0; // napin on/off tila
int motorEnabled = 0; // moottori päälle/pois
int motorSpeed = 200; // moottorin nopeus
int tila = 0; //arvottava numero
//Servo
Servo sormet; // servon nimi
int pos; // servon alkuasento

void setup() {
  // Pinnien roolit
  pinMode(nappiPin, INPUT_PULLUP);
  pinMode(controlPin1, OUTPUT);
  pinMode(controlPin2, OUTPUT);
  pinMode(enablePin, OUTPUT);
  sormet.attach(10); // Servo pinniin 10
  Serial.begin(9600); //Seurataan arpomista. Voi poistaa

  // Moottorin ohjauspinni pois päältä
  digitalWrite(enablePin, LOW);
}

void loop() {
  // Luetaan nappi
  nappiTila = digitalRead(nappiPin);
  delay(1);

  if (nappiTila == LOW) { //Heilutus
    up();
    delay(500);
    down();
    delay(500);
    up();
    delay(500);
    down();
    delay(500);
    up();
    delay(500);
    down();
    kps(); //Kivi Paperi Sakset
  }
  //funktio moottorin liikuttamiseen vetosuunnassa.
  void up(){
    digitalWrite(controlPin1, LOW);
    digitalWrite(controlPin2, HIGH);
    analogWrite(enablePin, motorSpeed);
    delay(500);
    analogWrite(enablePin, 0);
  }
  //funktio moottorin liikuttamiseen päästösuunnassa
  void down(){
    digitalWrite(controlPin1, HIGH);
    digitalWrite(controlPin2, LOW);
    analogWrite(enablePin, motorSpeed);
    delay(500);
    analogWrite(enablePin, 0);
  }
  //sormien tilan arvonta ja liikutus
  void kps(){
    tila=random(0,3); //Arvotaan K,P tai S
    Serial.print(tila);

    delay(1000);
  }
}

```

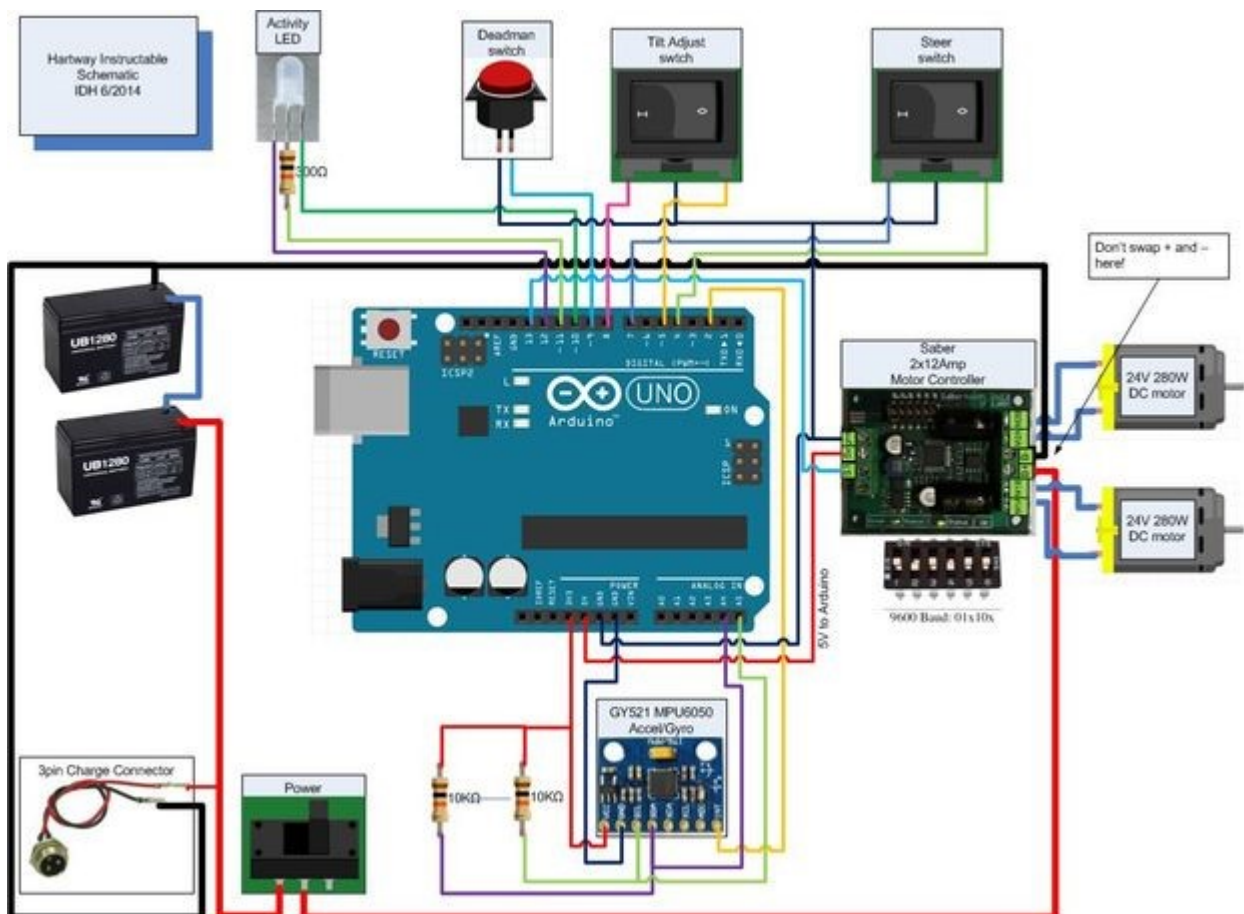
```
//Liikutetaan sormia
  if (tila == 0){          //Sakset
    pos =160;
    sormet.write(pos);

    delay(2000);
  }
  if (tila == 1){ //kivi
    pos =0;
    sormet.write(pos);
    delay(2000);
  }
  if (tila == 2){ //paperi
    delay(2000);
  }

  else { //palautetaan sormet alkuasentoon
    for(pos = sormet.read(); pos <=90; pos += 1) { // goes from 20 degrees to 90 degrees in 1
step
      sormet.write(pos);
      delay(5);
    }
    for(pos = sormet.read(); pos >=90; pos -= 1) { // goes from 20 degrees to 90 degrees in 1
step
      sormet.write(pos);
      delay(5);
    }
  }
}
```


1.7 Arduino-ohjattu Segway-kopio

Arduinoa voidaan käyttää ohjaamaan Segwayn kaltaista laitetta. Segwaytä ohjataan kallistelemalla ja Arduino havaitsee kallistukset Segwayssä olevaa gyroskooppia käyttäen. Arduino ohjaa sitten gyroskoopilta saadun datan mukaan laitteen pyöriä. Työ on lainattu osoitteesta <http://www.instructables.com/id/Rideable-Segway-Clone-Low-Cost-and-Easy-Build/> Tässä työssä gyroskooppina toimii MPU6050 IMU board. Pyörien moottoreina on kaksi 24 V jännitteellä ja 280 W teholla toimivaa DC-moottoria ja niiden virtalähteenä on kaksi 12 V 7 Ah akkua. Arduino ei voi suoraan hallita näin suuria virtoja ja siksi käytössä on myös moottorinohjain Saber 2x12 Amp Motor Controller. Tarvitaan myös 2 keinukytkintä, painokytkin, virtakytkin, LED ja kolme 10k Ω :in vastusta sekä akkujen laturille menevään johtoon sopiva liitin. Kytkenät esitetään kuvassa 14.



Kuva 14: Segway kopion kytkentäkaavio. (<http://www.instructables.com/id/Rideable-Segway-Clone-Low-Cost-and-Easy-Build/>)

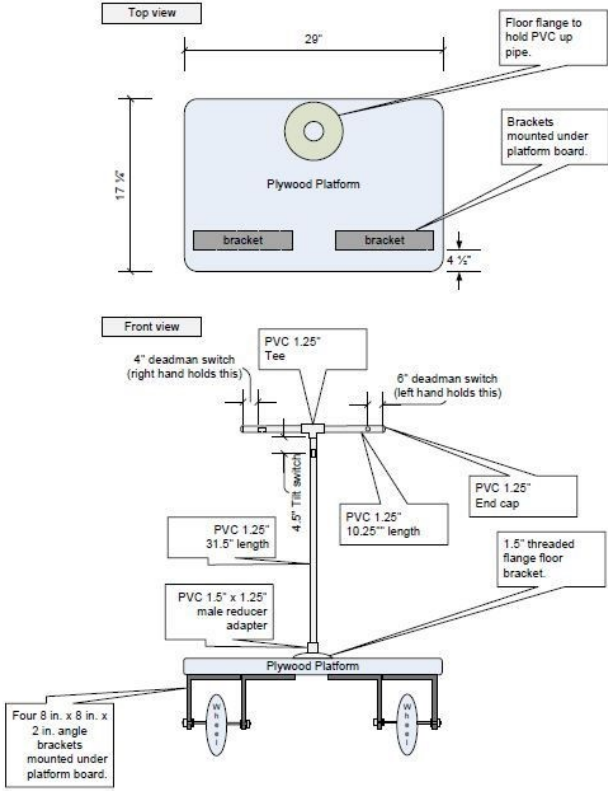
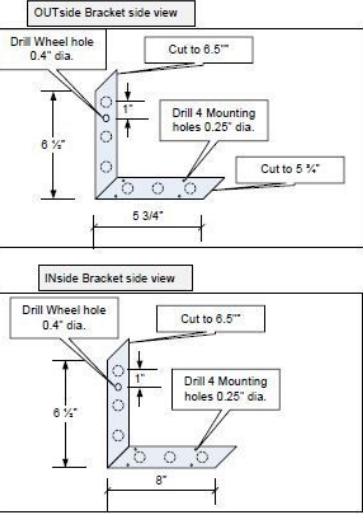
Työssä käytetään monimutkaisempia laitteita. Arduino pystyy kommunikoimaan näitten laitteitten kanssa sopivilla kirjastoilla. Tässä työssä käytetään neljää kirjastoa. I2Cdev.h-kirjastoa käytetään sopivan sarjaviestinnän aikaansaamiseksi. Wire-kirjasto on Arduinon R3 mallin laitteille tarvittava kirjasto, jotta I2Cdev.h-kirjasto toimii oikein. Gyroskoopin tiedot saadaan luettua kirjastolla MPU6050_6Axis_MotionApps20.h. SoftwareSerial-kirjastolla saadaan sarjaviestintä tehtyä pinnien 0 ja 1 kautta ja niitä voidaan käyttää vianhaussa. Wire ja SoftwareSerial kirjastot löytyvät valmiina Arduinon ohjelmointiympäristöstä mutta kaksi muuta joudutaan hankkimaan Internetistä. Kirjastot ovat varsin paljon käytettyjä ja niiden jakajiakin on monia.

Sammuksissa Segway-kopio on pyörien ja etulaitansa varassa. Laite laitetaan päälle virtakytkimellä, jolloin LED muuttuu punaiseksi ja IMU board alkaa kalibroida itseään noin 8 sekunnin ajan. Seuraavaksi laitetta nostetaan niin, että sen astinlauta on vaakasuorassa maan kanssa. Kun kuolleen miehen kytkimenä toimivaa painokytkintä painetaan, LED muuttuu vihreäksi ja Segwayn päälle voidaan astua. Segway saadaan nyt liikkumaan eteen- ja taaksepäin kallistamalla eteen ja taakse. Kääntyminen toteutetaan käyttämällä keinukytkintä. Toisella keinukytkimellä voidaan säätää laitteen neutraalia kallistusta maan suhteen.

Tässä mallissa sähkömoottorien voima viedään ketjuvälityksillä laitteen alla oleville pyörille. Pyörät pyritään laittamaan lähelle laitteen painopistettä ja voidaan sijoittaa myös esimerkiksi laitteen ulkoreunoille. Laitteessa suunta muutetaan painamalla keinukytkimestä. Monessa Segway:tä imitoivissa laitteissa suunnan muuttaminen on tehty ohjaustankoa kallistelemalla. Tämä onnistuu esimerkiksi kytkemällä potentiometri kallistuvan tangon tyveen.

Hartway Instructable
Mechanical plans
IDH 3/2014

Cable Lengths.
 Battery cables #1, extension cord wire = 3'
 Battery cables #2, extension cord wire = 2'
 Motor cables #1, extension cord wire = 2'
 Motor cables #2, extension cord wire = 1.5'
 Steer switch, 3 conductor phone wire = 6'
 Tilt Switch, 3 conductor phone wire = 6'
 Deadman switch, 2 conductor phone wire = 6'



Kuva 15: Malli Segway kopiolle. (<http://www.instructables.com/id/Rideable-Segway-Clone-Low-Cost-and-Easy-Build/>)

YHTEENVETO

Arduino on monipuolinen alusta prototypointiin. Suurena etuna on laaja kehittäjäkunta, jonka ansiosta löytyy valmiita koodeja, joita voidaan hyödyntää omissa töissä. Tämän kirjoituksen esimerkkisovelluksissakin käytettiin paljon valmiita koodeja ja esimerkiksi verhon aukaisimeen saatiin koodi tekemällä pieniä muutoksia toiseen tarkoitukseen tehtyyn koodiin. Arduinolle on paljon muitakin käyttömahdollisuuksia mekatroniikassa. Esimerkiksi laitteiden ohjaaminen Internetin välityksellä tai radiosignaaleilla onnistuu Arduinolla kohtalaisen helposti. Arduinon analogiset mahdollisuudet ovat varsin vaatimattomat, mutta antavat kuitenkin hyvän kokeilupohjan uusille ideoille. Lisämoduulit laajentavat Arduinon käyttöominaisuuksia entisestään. Vapaa laitteisto mahdollistaa myös sen, että muut voivat vapaasti kehittää omia mikrokortteja tai lisämoduuleja Arduinon pohjalta. Markkinoilla on jo paljon Arduinon kopioita muilta valmistajilta. Tulevaisuudessa Arduinolle voi siis olla saatavana entistä laajempi kirjo ominaisuuksia ja käyttömahdollisuuksia.

LÄHDELUETTELO

Fitzgerald S & Shiloh M (2013) Arduino Project Book, Toinen uusintapainos, Italia.

Lazar J, (2013) Arduino and LEGO project, New York, Apress.

Arduino, <http://www.arduino.cc/> [1.4.2015]

Arduino, Arduino UNO <https://www.arduino.cc/en/Main/ArduinoBoardUno> [2.4.2015]

Kushner D, IEEE Spectrum. The Making of Arduino <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino> [15.10.2015]

Williams E, Hackaday, Arduino V. Arduino <http://hackaday.com/2015/02/25/arduino-v-arduino/> [20.10.2015]

Williams E, Hackaday, Arduino V. Arduino Part II <http://hackaday.com/2015/03/12/arduino-v-arduino-part-ii/> [21.10.2015]

Ibrahim AM Mechatronics Tutorials <http://mechatronicstutorials.blogspot.fi/2014/07/balancing-of-ball-on-beam-using-arduino.html> [10.11.2015]

ihart instructables, Rideable Segway Clone - Low Cost and Easy Build

<http://www.instructables.com/id/Rideable-Segway-Clone-Low-Cost-and-Easy-Build/> [20.1.2016]

