



Høgskulen
på Vestlandet

BACHELOROPPGAVE:
BO19E-28 BACHELORMAL

Didrik Aas
Marius Haver Fjotland

30. mai. 2019

Dokumentkontroll

<i>Rapportens tittel:</i> Laserpekerbasert gangeassistent for pasienter med Parkinsons sykdom	<i>Dato/Versjon</i> 30.05.2019/0.20
<i>Forfatter(e):</i> Didrik Aas Marius Haver Fjotland	<i>Rapportnummer:</i> B019E-28
	<i>Studieretning:</i> HEEL16
<i>Høgskolens veileder:</i> Inge Vivås	<i>Antall sider m/vedlegg</i> 45
	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> HVL Elektro intern	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson(er) (inkludert kontaktinformasjon):</i> Mathias Chr. Mathiesen (elektrofaglig)	

Revisjon	Dato	Status	Utført av
0.11	30.01.19	Første utkast	Didrik Aas
0.12	20.02.19	Revidert utkast	Marius Haver Fjotland
0.20	30.05.19	Ferdig utarbeidet	Marius Haver Fjotland og Didrik Aas

Forord

Denne bacheloroppgaven er skrevet i forbindelse med avsluttende utdanning for studieretningen Elektronikk ved Høgskulen på Vestlandet. Vi har fått i oppgave å ta for oss et prosjekt som omhandler og konstruere en laserpeker som skal være et verktøy i hverdagen til folk med Parkinson for å hjelpe dem med å bevege seg rundt. Det har vært en spennende oppgave med mange utfordringer og den har vært godt knyttet opp til vår fagbakgrunn i fra elektronikk studiet.

Vi vil først og fremst takke vår veileder Inge Vivås, for gode retningslinjer, veiledning og tilgjengelighet under oppgaveskrivingen. Vi vil også takke Inta Tove G. Taranrød for å ha tatt seg tid til å gi oss et innblikk i hvordan løsningen kunne bli best mulig og dermed ga oss et godt utgangspunkt å jobbe ut ifra. Deretter vil vi i tillegg takke våre medstudenter Kristian Auestad og Sverre Fjæra, som har vært hjelpelige når det trengtes og kommet med deres egne synspunkter på prosjektet. Denne oppgaven har vært svært lærerik å utforme og håper den blir like lærerik for den som leser.

Takk!

Sammendrag

En kort samlet oppsummering av hva arbeidet går ut på, og hva resultatet av arbeidet er. Diskuter i hvilken grad den leverte løsningen svarer til gitt kravspesifikasjon (er det knyttet begrensninger til løsningen?). Vis om nødvendig til kapitlene i hovedteksten for mer detaljer.

Oppgaven vår går ut på å lage en gangeassistent i form av en gyrostabilisert laserpeker. Gangeassistenten skal gjøre det mulig å gå normalt for personer som lever med Parkinsons sykdom. Produktet skal i tillegg gi mulighet til å justere laseren via blåttann kommunikasjon.

Vi har forholdt oss til et budsjett på 10 000 NOK.

Vi har ikke sett på deksel og vanntetthet men produktet vårt er lett, lite og består ikke av noen spesielt dyre komponenter.

1 Innholdsliste

Dokumentkontroll	2
Forord	3
Sammendrag	4
1 Innledning.....	7
1.1 Oppdragsgiver	7
1.2 Problemstilling	7
1.3 Hovedidé for løsningsforslag	7
2 Kravspesifikasjon	8
3 Analyse av problemet	9
3.1 Utforming av mulige løsninger	9
3.1.1 Løsningsalternativ 1	9
3.1.2 Løsningsalternativ 2	9
3.1.3 Vurderinger i forhold til verktøy og HW/SW komponenter	9
3.2 Konklusjon	12
4 Realisering av valgt løsning	14
5 Testing	21
6 Diskusjon.....	29
7 Konklusjon	30
Referanser.....	31

Figurer

Figur 1.....	11
Figur 2.....	12
Figur 3.....	14
Figur 4.....	17
Figur 5.....	21
Figur 6.....	22
Figur 7.....	25
Figur 8.....	26
Figur 9.....	27
Figur 10.....	28
Figur 11.....	34
Figur 12.....	36
Figur 13.....	37
Figur 14.....	38
Figur 15.....	39
Figur 16.....	40
Figur 17.....	40
Figur 18.....	42
Figur 19.....	43

1 Innledning

1.1 Oppdragsgiver

Oppdragsgiver er HVL Elektro

Ekstern kontaktperson er Inta Tove G. Taranrød, telefon: 55 58 56 83, e-post: itgt@hvl.no

1.2 Problemstilling

Oppgaven vår går ut på å lage en litt mer avansert laserpekerbasert gangeassistent enn den som er på markedet nå. Enkelte pasienter med Parkinsons sykdom sliter med normal gange, da de plages av problemer med å starte bevegelse i bena. En midlertidig løsning på dette kan være en visuell indikator i form av en laserpeker. Produktet skal med laser markere et punkt på bakken, en linje å gå etter, som vil hjelpe pasientene med å starte bevegelse av føttene.

For å løse dette skal vi ta i bruk en mikrokontroller av typen Arduino nano. Vi skal lære hvordan man tar i bruk akselerometer og gyroskop for å så sette hardware og software sammen til et brukervennlig produkt.

1.3 Hovedidé for løsningsforslag

Per nå vet vi at det er mulig å ta i bruk Blåtann kommunikasjon og gyrostabilisering i et Arduinoprojekt. Det burde være fullt mulig å kombinere disse funksjonene for å styre en laserpeker.

Vi ser for oss å bruke et akselerometer og gyroskop for å få til stabiliseringen av pekeren og dermed klare å ha en rett linje som indikator for pasienten.

2 Kravspesifikasjon

For å løse oppgaven har to alternativer, Arduino mikrokontroller eller PLS (programmerbar logisk styring). Vi har valgt å for en mikrokontroller løsning ettersom dette er noe vi har erfaring med og det er et veldig godt fagmiljø rundt Arduino og programmering av mikrokontrollere.

I skrivende stund vet vi at det er mulig å ta i bruk Blåtann kommunikasjon og gyrostabilisering i et Arduinoprojekt. Det burde være fullt mulig å kombinere disse funksjonene for å styre en laserpeker.

Vi ser for oss å bruke ett akselerometer og gyroskop for å få til stabiliseringen.

Kravspesifikasjon

	Krav
1.	Pekeren bør tåle regn
2.	Være relativt liten og ha en overkommelig kostnad
3.	Styringen bør ideelt sett være trådløs (Blåtann?)
4.	Enheten bør kunne benyttes uavbrutt i minst to timer uten batteriskifte eller opplading.

Tidskrav

- Prosjektet skal følge tidskrav i henhold til milepælene for fremdriftsplanen.
- Forstudiet skal være ferdig innen 31. Januar 2019.
- Midtveispresentasjon være gjort i begynnelsen av april 2017.
- Rapporten skal leveres inn 31. mai 2019.
- EXPO skal gjennomføres 13. Juni 2019

Hovedmålet med dette prosjektet er at produktet skal være en bedre versjon enn tidligere produkter av denne typen. Det vil si at det skal være fleksibelt i forhold til hvilken situasjon brukeren er i, for eksempel hvis det er nødvendig med mye klær på. Det er også en fordel om produktet er kledelig.

3 Analyse av problemet

For at pekeren skal tåle vann trenger vi et deksel som tåler vann, vi tenker oss at vi bruker hardplast. Ut ifra de komponentene vi har tenkt å bruke vil størrelse og kostnad ikke være noe stort problem. For den trådløse styringen ser vi fortsatt på ulike løsninger, men det er meget mulig at vi faller ned på blåttann. Vi får ikke testet batteritid enda, men det skal også være overkommelig med det utstyret vi har.

To-akse styring kan vi implementere ved å ta i bruk to mikroserver. Servo 1 skal styre rotasjon rundt x-aksen og servo 2 skal styre rotasjon rundt y-aksen. En mikrokontroller-krets med et Motiontracking device kan brukes til å styre disse to servoene, gitt riktig programkode. Denne to-akse-styringen vil fungere som gyrostabilisering fordi servoenes rotasjon vil kompensere for bevegelser som kan endre laserstrålens retning.

3.1 Utforming av mulige løsninger

3.1.1 Løsningsalternativ 1

Systemet vil da bestå av en hc05, en bluetooth SSP (Serial Port module) som er designet for serielle trådløse tilkoblinger, en fjernkontroll og to laserpekere som vil gi signal til hver sin fot. I dette tilfellet ville vi brukt en app som fjernkontroll for å styre hc05en slik at den bare vil fungere som en mottaker.

3.1.2 Løsningsalternativ 2

I prinsippet den samme løsningen som over utenom at vi bruker en laserpeker som dekker hele bredden, altså fungerer for begge beina. Det vi forenkler systemet og kunne gjøre det mer stabilt.

3.1.3 Vurderinger i forhold til verktøy og HW/SW komponenter

Generelt om Arduino

Arduino er en programmerbar mikrokontroller som brukes for å styre og lese elektroniske komponenter. Arduino-systemet består av et kretskort og programvare som man bruker til å kompilere og laste opp programmet til kretskortet. Kretskortet vil da kommunisere med og styre tilkoblede komponenter i henhold til programmet. Ett eksempel på dette er styring av servoer.

Arduino er veldig populært og brukes av mange til å lage hobbyprosjekter og lignende. Det finnes et stort utvalg av komponenter man kan bruke til å lage arduino-prosjekter, samtidig kan man finne mye god informasjon på internett om hvordan man kan sette sammen et eget prosjekt.

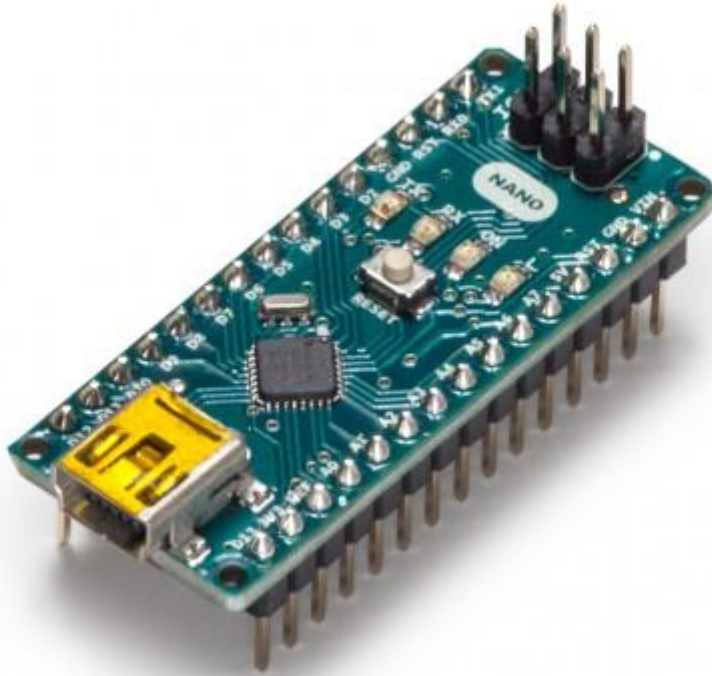
Arduino er lett å sette seg inn i og det var i hovedsak derfor vi valgte å bruke arduino som grunnlag for produktet vårt.

Arduino IDE

Arduino mikrokontrollere programmeres med Arduino Integrated Development Environment (IDE). Mikrokontrolleren kobles til datamaskin via USB før man laster opp programvaren til utviklingskortet. Programmeringsspråket til Arduino er basert på C/C++ inneholder en rekke biblioteker med funksjoner som hentes opp fra koden.

Bibliotek for MPU6050 funksjoner er ikke inkludert i Arduino IDE. Dette biblioteket måtte vi laste ned og importere til Arduino IDE.

Arduino Nano

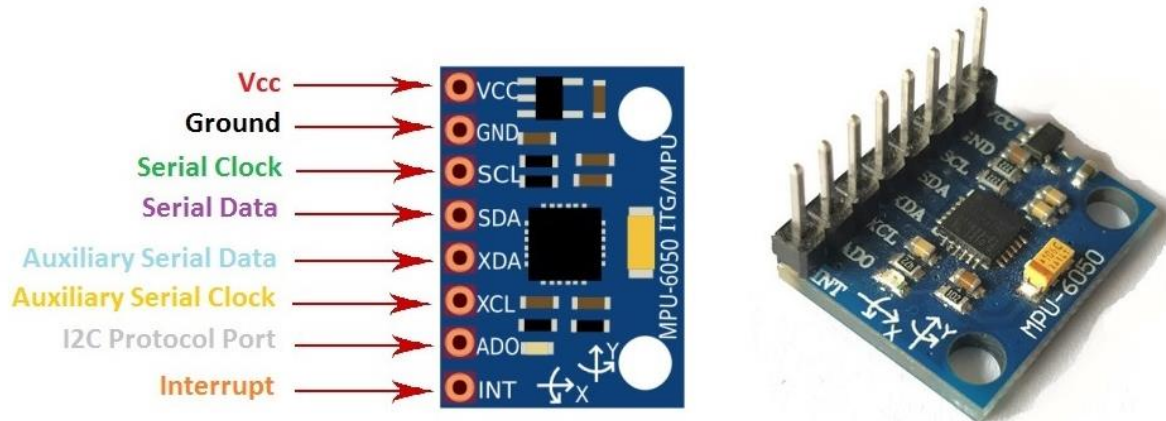


Figur 1

Arduino Nano er en av de mindre versjonene av et Arduino kretskort (18x45 mm). Selv om kretskortet er mye mindre enn andre versjoner vil Arduino Nano ha like god funksjonalitet i dette prosjektet. Kretskortet har 22 digitale input/output pins og 8 analoge input pins, to av disse vil brukes til å lese data fra gyroskopet vårt. Kortet har en input spenning på 7-12 Volt DC spenning som må mates inn på en Mini-B USB Port. Arduino Nano har en operasjonsspenning på 5V og kan brukes til å forsyne alle de tilkoblede elektroniske komponentene med 5V DC spenning.

Kretskortet må settes på et breadboard for å brukes med andre komponenter. Dette er praktisk ettersom vi kan feste alle komponenter på ett enkelt breadboard.

InvenSense MPU-6050



Figur 2

InvenSense MPU-6050 sensoren inneholder et MEMS akselerometer og en MEMS gyro i en enkeltbrikke. Den er veldig nøyaktig, da den inneholder 16-bits analog til digital konvertering maskinvare for hver kanal. Derfor fanger den x-, y- og z-kanalen på samme tid. Sensoren bruker I2C-bussen til grensesnitt med Arduino. MPU-6050 fungerer alltid som en slave til Arduino med SDA- og SCL-pinnene koblet til I2C-bussen.

Sensoren har en "Digital Motion Processor" (DMP), også kalt en "Digital Motion Processing Unit". DMP kan gjøre raske beregninger direkte på brikken. Dette reduserer belastningen for mikrokontrolleren. DMP kan til og med gjøre beregninger med sensorverdiene til en annen brikke, for eksempel et magnetometer som er koblet til den andre -I2C-bussen.

MPU-6050 er ikke dyrt, spesielt med tanke på at den kombinerer både et akselerometer og en gyro.

3.2 Konklusjon

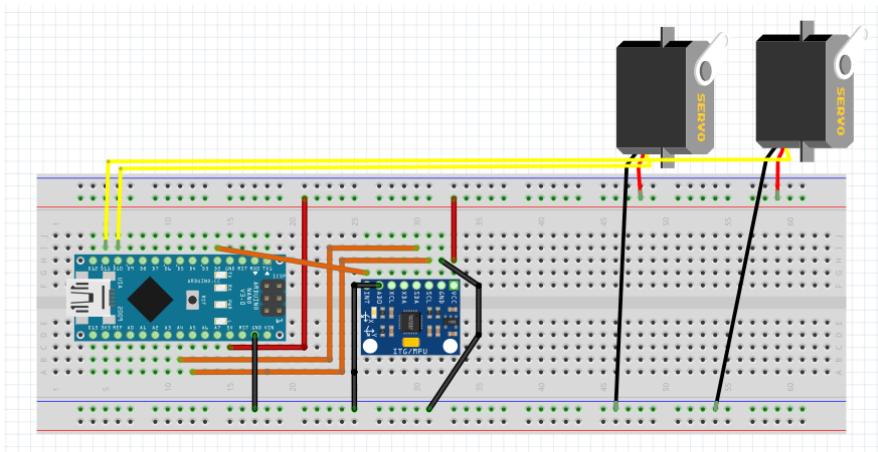
Vi valgte å løse oppgaven ved å ta i bruk Arduino Nano kombinert med en MPU6050 akselerometer-&-gyroskop modul. Vi valgte denne løsningen for å fremstille et produkt som er så lite som mulig. I tillegg er delene billige og lett tilgjengelig, samtidig finnes det mye bra dokumentasjon og prosjekter som kan brukes til inspirasjon.

4 Realisering av valgt løsning

3.1 Design

3.1.1 Kretskort

Laserpekeren er koblet til to servoer som styres av mikrokontrolleren. Mikrokontrolleren kommuniserer med MPU6050, som registrerer endringer i hastighet og retning. Mikrokontrolleren vil bruke denne informasjonen til å rotere servoarmene, disse rotasjonene skal kompensere for kroppsbevegelser som ville fått laseren til å peke i feil retning.



Figur 3

3.1.2 Setup kode

```
1 #include "Wire.h"
2 #include "I2Cdev.h"
3 #include "MPU6050.h"
4 #include "Servo.h"
5
6 MPU6050 mpu;
7 int16_t ax, ay, az;
8 int16_t gx, gy, gz;
9 Servo servol;
10 Servo servo2;
11
12 int val1;
13 int val2;
14 int prevVal1;
15 int prevVal2;
16
17 void setup()
18 {
19   Wire.begin();
20   Serial.begin(38400);
21   Serial.println("Initialize MPU");
22   mpu.initialize();
23   Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");
24   servol.attach(10);
25   servo2.attach(11);
26 }
```

I setup-delen av programmet må vi initialisere MPU6050, programmet skal så teste tilkobling til MPU6050 og skrive til konsoll om komponenten er tilkoblet eller ei. Servoene kobles til digital I/O pin 10 og 11.

3.1.2.1 Loop kode

```
28 void loop()
29 {
30
31 mpu.getMotion6(&sax, &say, &sz, &gx, &gy, &gz); //read 6 values into 6 variables
32 val1 = map(ax, -17000, 17000, 0, 179); //change the range of the ax value from -17000/17000 to 0/179
33 if (val1 != prevVal1)
34 {
35 servo1.write(val1); //if the value of ax changes, the servo rotates
36 prevVal1 = val1;
37 }
38 val2 = map(ay, -17000, 17000, 0, 179); //change the range of the ay value from -17000/17000 to 0/179
39 if (val2 != prevVal2)
40 {
41 servo2.write(val2); //if the value of ay changes, the servo rotates
42 prevVal2 = val2;
43 }
44 delay(50);
45
46 }
```

I loop-delen av programmet skal arduino lese endringer i bevegelse på x-aksen og y-aksen. Mikrokontrolleren skal deretter rotere servoene slik at laseren fortsatt peker i riktig retning.

3.1.3 Bluetooth styring

Som vår løsning for fjernstyring valgte vi å bruke en app som fjernkontroll. Den kan lastes ned på mobilen og kan brukes til å justere laserpekeren og tilpasse den til brukeren. Den skal være lett og bruke og ha store taster. For å sette den opp kreves det en nettverksforbindelse men selve kommunikasjonen foregår over bluetooth.

Til høyre ser vi hvordan skjermen vil se ut på mobiltelefonen. Den er ganske enkel og har tre funksjoner. Av og på knapper, sjekke om den har forbindelse med laserpekeren og å kunne stille selve strålen opp og ned. Dette er greit å kunne for å justere etter for eksempel hvor høy man er eller evt. om terrenget skulle endre seg.



Figur 4

For å lage appen har vi brukt «MIT app inventor». De har en oversiktlig og grei måte å gjøre det på når man ikke har så mye erfaring med å lage apper. Man bygger «blokker» som vist i figur 2.

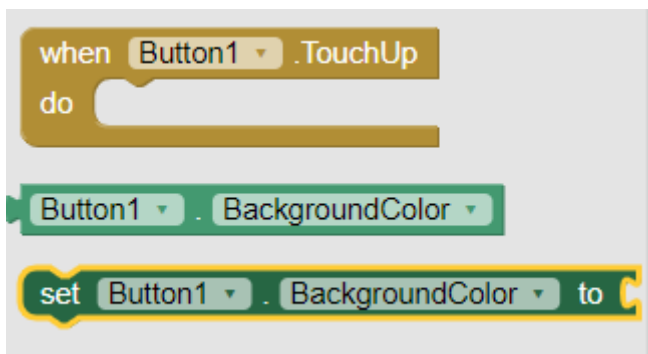


Figure 7.

3.1.3.1. Kildekode utgangspunkt

Til å begynne med tok vi utgangspunkt i denne kildekoden fra et prosjekt med en «little arm»-robot.

```
//////Other Intiations here</p><p>void setup(){  
  
  Serial.begin(9600);  
  
  baseServo.attach(5);      // attaches the servo on pin 5 to the servo object  
  
  shoulderServo.attach(4);  
  
  elbowServo.attach(3);  
  
  gripperServo.attach(2);  
  
  
  Serial.setTimeout(50);    //ensures the the arduino does not read serial for too long  
  
  Serial.println("started");  
  
  baseServo.write(90);      //intial positions of servos  
  
  shoulderServo.write(100);  
  
  elbowServo.write(110);  
  
  ready = 0;  
  
} //primary arduino loop<br>void loop()  
  
{  
  
  if (Serial.available()){  
  
    ready = 1;  
  
    desiredAngle.base = Serial.parseInt();  
  
    desiredAngle.shoulder = Serial.parseInt();
```

```
desiredAngle.elbow = Serial.parseInt();

desiredGrip = Serial.parseInt();

desiredDelay = Serial.parseInt();</p><p> if(Serial.read() == '\n'){           // if the
last byte is 'd' then stop reading and execute command 'd' stands for 'done'

Serial.flush();           //clear all other commands piled in the buffer

Serial.print('d');       //send completion of the command

}

}
```

Den leser verdier fra appen for de forskjellige servoene som beveger de ulike leddene.

3.1.3.2. Kildekode adaptert

```
#include "Wire.h"
//#include "I2Cdev.h"
// #include "MPU6050.h"
#include "Servo.h"

//MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
Servo baseServo;
Servo secondaryServo;
int ready;
int desiredAngle;
```

```
void setup(){
  Serial.begin(9600);
  baseServo.attach(3);    // attaches the servo on pin 5 to the servo object
  secondaryServo.attach(2);

  Serial.setTimeout(50); //ensures the the arduino does not read serial for too long
  Serial.println("started");
  baseServo.write(90);    //intial positions of servos
  secondaryServo.write(100);

  ready = 0;
} //primary arduino loop
void loop()
{
  if (Serial.available()){
    ready = 1;
    desiredAngle.baseServo = Serial.parseInt();
    desiredAngle.secondaryServo = Serial.parseInt();
    desiredDelay = Serial.parseInt();</p><p>    if(Serial.read() == '\n'){
// if the last byte is 'd' then stop reading and execute command 'd' stands for 'done'
      Serial.flush();          //clear all other commands piled in the buffer
      Serial.print('d');       //send completion of the command
    }
  }
}
```

Dette er tilpasset vårt prosjekt, med antall servoer og hvor vi vil ha dem til å virke.

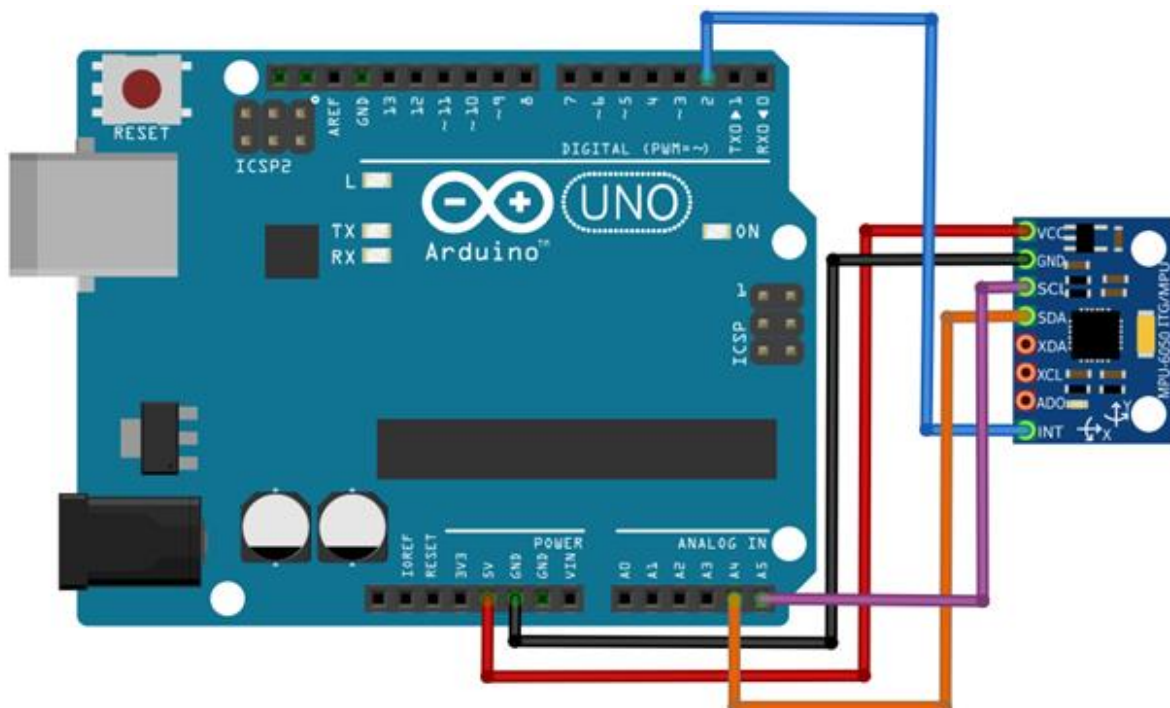
5 Testing

Dette kapitlet handler om tester vi utførte før design og etter ferdigstilt produkt.

4.1 Funksjonstesting av MPU6050

Før vi satt sammen kretsen mente vi det var viktig å teste at gyroskop-modulen fungerte skikkelig. For å gjøre dette måtte vi laste ned og importere bibliotekfiler så kunne bruke funksjonene i testprogrammet. Testprogrammet fulgte med bibliotekfilene.

MPU 6050 modulen kommuniserer med Arduinoen ved å bruke I2C protokollen. MPU 6050 blir koblet til Arduinoen som vist i Figur 1. Testprogrammet man kjører tar i bruk Arduinoens interrupt pin. Arduinoens interrupt pin 0 er satt til digital pin 2, denne kobles til punktet merket INT på MPU 6050.



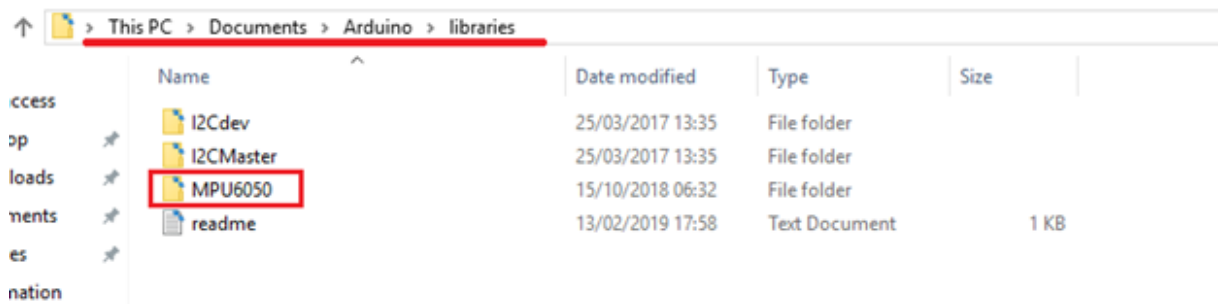
Figur 5

Oppkobling av MPU6050

Neste steg blir å koble opp I2C linjene. For å gjøre dette kobles pin merket SDA på MPU 6050 til Arduinoens analog pin 4 (SDA), og pin merket SCL på MPU 6050 kobles til Arduinoens analog pin 5 (SCL). VCC kobles til Arduinoens 5V power supply pin, GND kobles til ground.

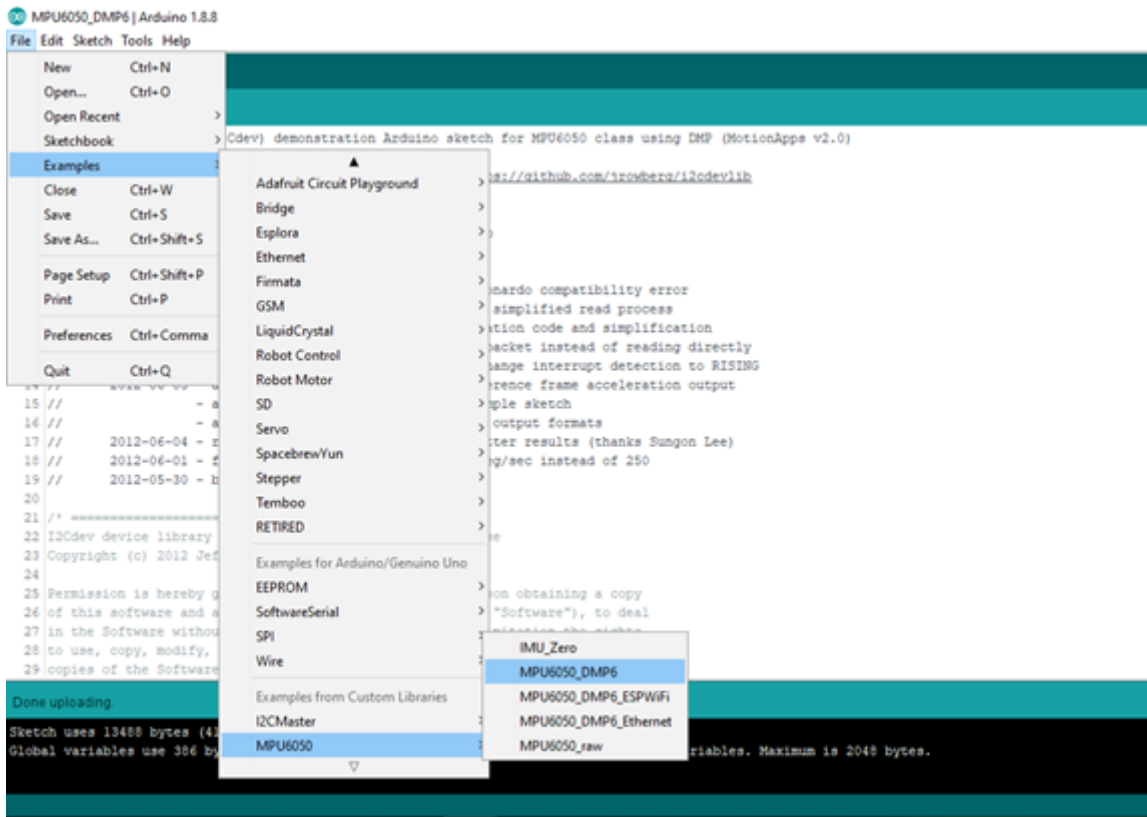
Testkode for MPU 6050

For å få til en funksjonstest MPU6050 med Arduino må man først laste ned Arduino library for MPU 6050, utviklet av Jeff Rowberg. Mappen med navn «MPU6050» kopieres og limes inn i Arduinoens «library» mappe. Denne mappen finner man under Arduinoens installasjonsmappe.

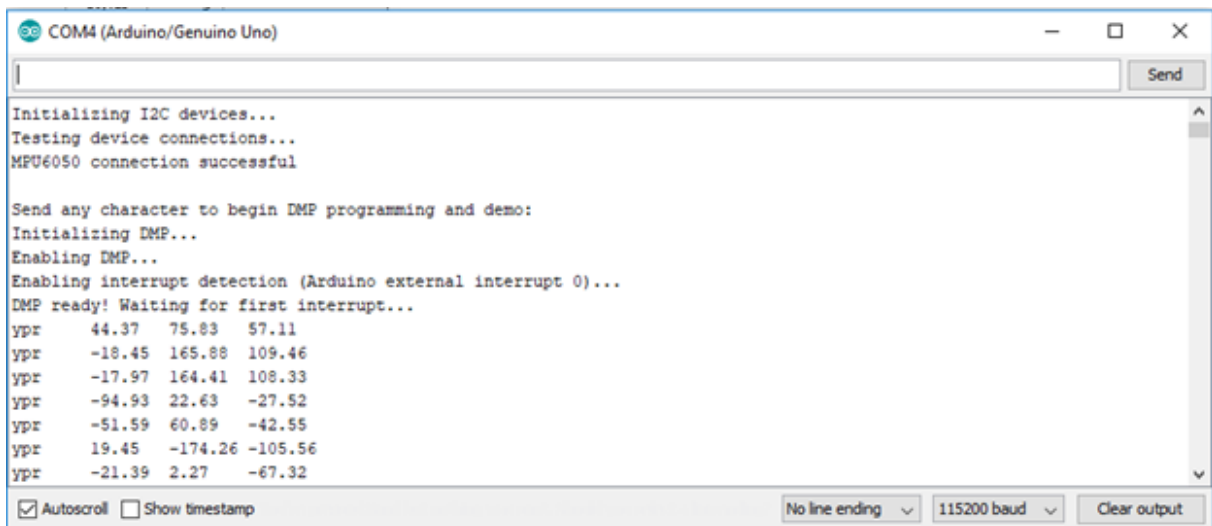


Figur 6

Når man installert biblioteket kjører man Arduino IDE. Testprogrammet skal da ligge inne i Arduino IDE. For å åpne testprogrammet, trykk på fanen «File» og naviger Examples à MPU6050 à MPU6050_DMP6.

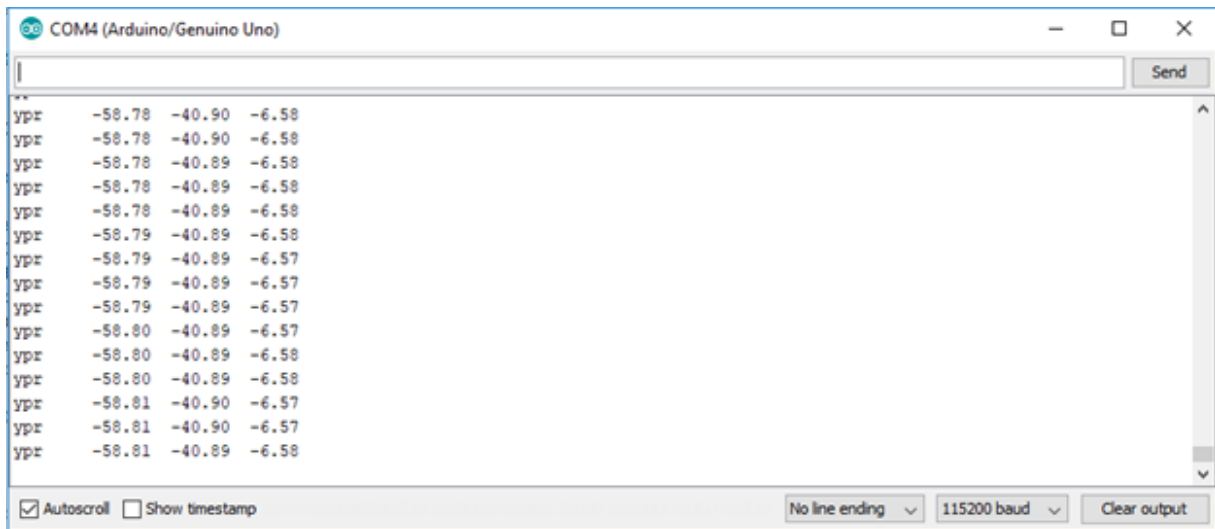


Etter å ha åpnet programmet trykk «Verify», deretter kobles Arduino Uno til PC via USB kabel. Trykk «Upload» og gå inn på Tools à Serial monitor.



Serial monitor

Målingene fra MPU 6050 (i Yaw-Pitch-Roll format) vil så printes ut i serial monitor. Målingene vil være ustabile i starten og det tar noen sekunder før de stabiliseres. Hvis man løfter opp kretsen og roterer den rundt i luften vil måleresultatene endres momentant.



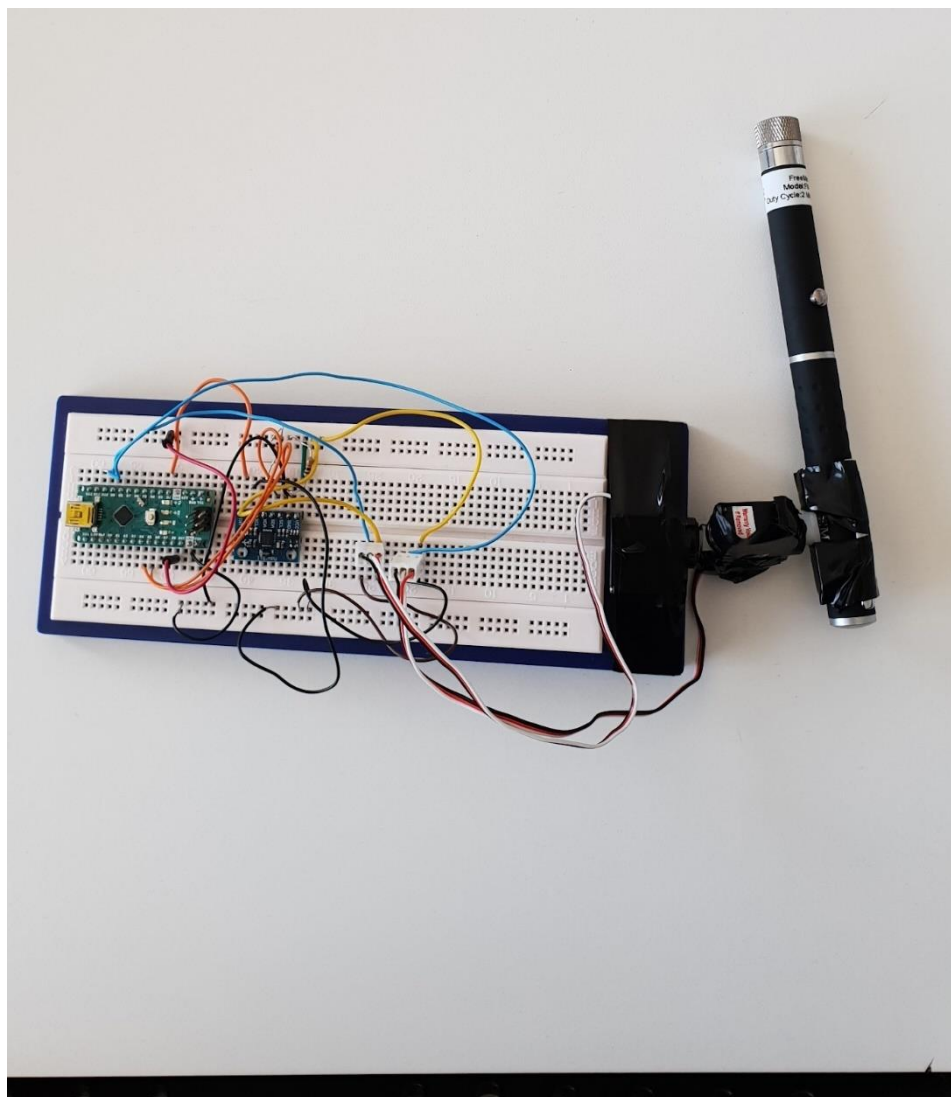
The screenshot shows the Serial Monitor window for a COM4 (Arduino/Genuino Uno) connection. The window displays a list of 16 lines of data, each starting with 'ypr' followed by three numerical values. The values are: -58.78, -40.90, -6.58; -58.78, -40.90, -6.58; -58.78, -40.89, -6.58; -58.78, -40.89, -6.58; -58.78, -40.89, -6.58; -58.79, -40.89, -6.58; -58.79, -40.89, -6.57; -58.79, -40.89, -6.57; -58.79, -40.89, -6.57; -58.80, -40.89, -6.57; -58.80, -40.89, -6.58; -58.80, -40.89, -6.58; -58.81, -40.90, -6.57; -58.81, -40.90, -6.57; -58.81, -40.89, -6.58. The window also features a 'Send' button, a 'Clear output' button, and settings for 'No line ending' and '115200 baud'. The 'Autoscroll' checkbox is checked, and the 'Show timestamp' checkbox is unchecked.

```
ypr -58.78 -40.90 -6.58
ypr -58.78 -40.90 -6.58
ypr -58.78 -40.89 -6.58
ypr -58.78 -40.89 -6.58
ypr -58.78 -40.89 -6.58
ypr -58.79 -40.89 -6.58
ypr -58.79 -40.89 -6.57
ypr -58.79 -40.89 -6.57
ypr -58.79 -40.89 -6.57
ypr -58.80 -40.89 -6.57
ypr -58.80 -40.89 -6.58
ypr -58.80 -40.89 -6.58
ypr -58.81 -40.90 -6.57
ypr -58.81 -40.90 -6.57
ypr -58.81 -40.89 -6.58
```

Serial Monitor, stabiliserte malinge.

4.2 Testing av produkt

Produktet har en veldig enkel funksjon så testingen var forsåvidt enkel, bare å feste kretsen i hoftehøyde og skru på laseren. Denne testen ble gjort på første utkast av prototypen.



Figur 7



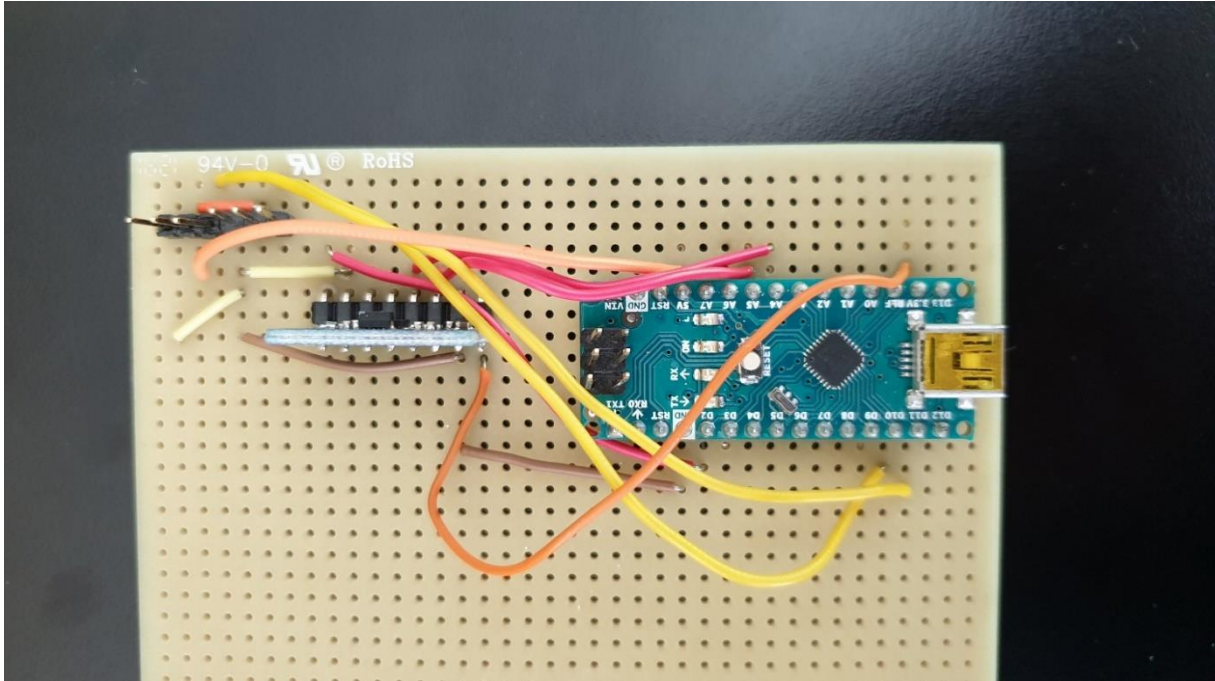
Figur 8

Når servoene er riktig innstilt vil laseren lyse en strek på gulvet noen centimeter foran brukeren. Streken vil ikke bevege seg når brukeren beveger det ene benet fremover. Vi møtte heller ikke på noen problemer når det kom til å “svinge” mens gange assistenten er i bruk.



Figur 9

Vi hadde heller ingen problemer med at servoene låste seg eller at MPU6050 skrudde seg av og på grunnet strømmangel. Vi ser ut ifra denne enkle testen at produktet fungerer som en gang assistent.



Figur 10

Andre utkast av prototype. Ledningene og komponentene er loddet fast på et breadboard. Servoene festes til breadboardet. Videre ønsker vi å få alt av maskinvare inn i en boks. Inne i boksen skal det også ligge et 9 volts batteri med ledere som loddes til Vin og Ground pinnene. Boksen eller dekselet skrues fast til et feste (med en «clip» som kan festes på innsiden av belte eller eventuelt direkte i buksen. Den vil ha åpningen sin på nedre halvdel slik at den blir beskyttet fra vær og vind, men at laseren fortsatt har god bevegelighet.

6 Diskusjon

I forhold til den opprinnelige fremdriftsplanen kom vi litt sent i gang. Dette var en blanding av litt dårlig planlegging og at noen komponenter hadde lang leveringstid. Ellers har vi overholdt de milepælene som var satt opp.

Når vi ser på risikoanalysen i etterkant ser vi at den traff veldig godt. Av de tingene vi hadde satt som lav sannsynlighet så var det bare delvis bestillingsproblem som inntraff, vi fikk alt som vi skulle, men enkelte komponenter tok litt for lang tid. De andre tingene vi hadde oppført som lav risiko var sykdom, økonomiske problemer og annet fravær inntraff ikke.

Problemer med kode var noe vi var bekymret for på forhånd fordi vi ikke følte oss helt trygge i arduino programvaren. Det skjedde også og satte oss et langt stykke tilbake da vi endte opp med å ikke klare å få ut en løsning i fra den dataen vi opprinnelig hadde sett oss ut som et godt utgangspunkt. Koden fungerte ikke, uten noen åpenbar forklaring. Vi trodde først at MPU6050 komponenten ble ødelagt under loddeprosessen, men da vi tok i bruk reserve komponenten og funksjonstestet begge MPU6050 brikkene på nytt, så vi at funksjonene for servo rotasjon førte til at MPU sluttet å kommunisere med Arduino Nano mikrokontrolleren. Det endte opp med at vi måtte forkaste det hele og ta i bruk en ny programkode for servostyring.

I forhold til komponentskade som vi hadde satt til middels i risikoanalysen så støtte vi på problemer med servoene som vi fikk tildelt da noen rett og slett ikke virket.

Vi bestemte oss også for å bruke standard servoer istedenfor mindre hardføre mikro servoer, men vi fant fort ut at denne typen servo trekker mer strøm enn det en Arduino Nano kan forsyne. Dette førte til regelmessige “brown-outs”, som regel etter 5-10 sekunder med drift. Derfor ble vi enige om å heller bruke mikro servoer av typen SG90 i løsningen vår.

Et annet problem vi traff på var integreringen av koden til fjernstyringsappen sammen med laserpekerens kode. Vi fikk ikke dette til å fungere. Da måtte vi sette oss ned å diskutere nødvendigheten av den trådløse styringen. Ettersom det er veldig lite en faktisk kan styre på laserpekeren og at den egentlig bare trengs å skruses på for å fungere som den skal, fant vi ut at vi ville fokusere på å få selve laserpekeren og få den best mulig til fristen. Dette gjorde at vi skjøv blåtann-prosjektet over til videre prosjektarbeid, slik at hvis vi jobber mer med prosjektet i ettertid kan vi plukke opp den tråden igjen for å optimalisere produktet, men vi konkluderte med at det egentlig er mer overfladisk og kanskje mer hensiktsmessig med denne

fjernkontrollen hvis vårt hjelpemiddel skal fungere sammen med andre hjelpemidler. Da ville det vært flott å ha noe som kunne styre alle produktene fra en kontroll.

7 Konklusjon

Da vi kom i gang med denne oppgaven fikk vi bruk for kunnskaper fra elektronikk, automasjon og programmering. det gjorde at arbeidet ble omfattende og vi fikk en bratt læringskurve. Oppgavens kjerne var å lage et hjelpemiddel for å forbedre gange evnen til parkinsonpasienter. Den skulle være robust og gjerne kunne styres trådløst. Vi har laget en app for å kontrollere den, men ikke klart å integrere den inn i systemet. Dette begynte vi å nedprioritere fordi vi så at det ikke var mye som trengte trådløs styring etter at laserpekeren var satt opp. Den kan bare skrues på så er enheten mer eller mindre selvgående.

Hvis vi skulle videreutvikle prosjektet (gitt mer tid) ville vi ha fått det testet av pasienter og fått tilbakemeldinger til forbedringer der. Det vi vet derimot er at vi ville ha tilstrebet og funnet mindre komponenter, i hovedsak laseren som er for stor i forhold til vårt ønske. Det skaper problemer med tanke på bevegelighet og fordi vi vil at vår løsning skal være så lite sjenerende som mulig. Streklasere som var små var vanskelige å få tak i, men måten de fungerer på er at det bare er et filter som splitter en vanlig laser. Derfor tror vi det kunne vært mulig å bruke en annen laser og sammen med filteret fått ett mer gunstig resultat.

Referanser

Her tar dere med litteratur som er brukt under arbeidet. Det kan være lærebøker fra aktuelle fag på skolen, annen støttelitteratur, håndbøker for maskinutstyr eller programvarer, nettsteder dere har benyttet etc. Når det gjelder websider, så bør dette dateres. Litteraturlisten skal være ryddig og systematisk satt opp, og alle referanser skal også refereres til i teksten.

Eksempelet viser bruk av den interne litteraturlisten til Word, og det anbefales å benytte denne. Andre gode verktøy sammen med Word er f. eks. Endnote.

Tips til hvordan du setter opp litteraturliste finner du på [1]

[1] HiB, UiB, NHH, UiO og Nasjonalbiblioteket, «Søk og Skriv,» 12 12 2014. [Internett]. Available: <http://sokogskriv.no/>. [Funnet 12 12 2014].

[1] HC-05 Bluetooth Module. Brukermanual,

Tilgjengelig på: <https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>

[2] Arduino, "Introduction to arduino nano",

Tilgjengelig på: www.theengineeringprojects.com

[3] MIT appinventor,

Tilgjengelig på: <http://appinventor.mit.edu/explore/>

[4] "Using MIT App Inventor to Control Arduino - the Basics"

Tilgjengelig på: <https://www.instructables.com>

[5] "Little arm robot, "An Advanced Arduino Robot Arm",

Tilgjengelig på: <https://www.littlearmrobot.com/>

[6] ”How to interface Arduino and the MPU 6050 sensor”

Tilgjengelig på: <https://maker.pro/arduino>

[7] MPU-6050, “MEMS MotionTracking Devices”,

Tilgjengelig på <https://www.invensense.com>

[8] Arduino, “Gyro Stabilizer W/ Arduino and Servo”

Tilgjengelig på <https://www.instructables.com/>

[9]” Parkninsons sykdom”

Tilgjengelig på: https://sml.snl.no/Parkinsons_sykdom

[10] “Parkninsons sykdom, en oversikt”

Tilgjengelig på: <https://nhi.no/>

[11] “introduction to MEMS gyroscopes”,

Tilgjengelig på: <https://electroiq.com>

Appendiks A Forkortelser og ordforklaringer

DMP	Digital Motion Processor
MEMS	Microelectromechanical systems
MIPS	Microprocessor without Interlocked Pipelined Stages
MPU	Motion Processing Unit
SRAM	Static Random Access Memory

Appendiks B Prosjektledelse og styring

B.1 Prosjektorganisasjon

Vi har delt ansvaret mellom oss, der Didrik har størst ansvar for det praktiske og tekniske med oppgaven, mens Marius har ansvar for det mer administrative.

B.2 Fremdriftsplan

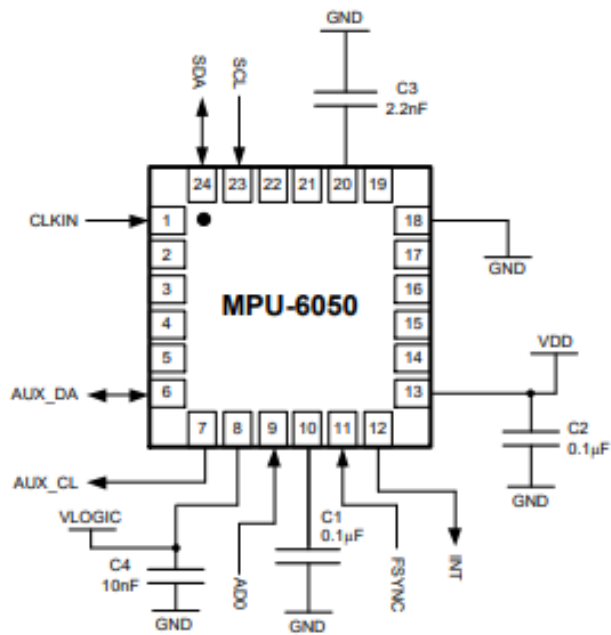
Oversikt over oppgaver



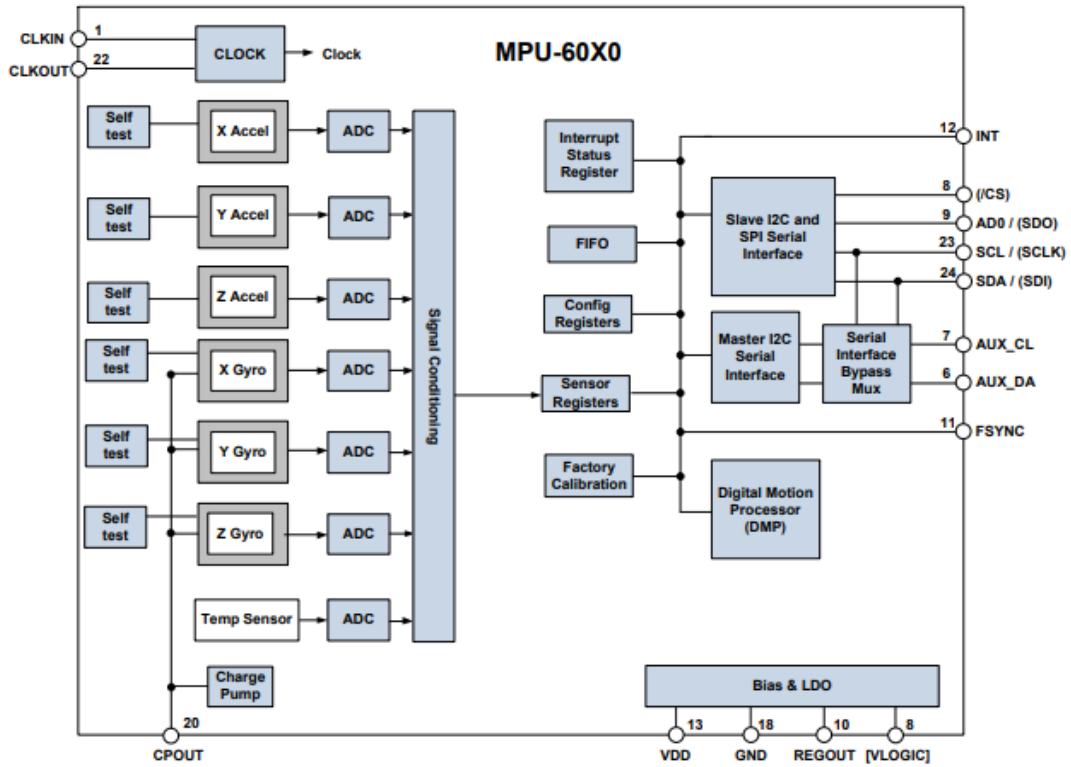
Figur 11

B.3 Risikoliste

Risiko #	Hendelse	Samsynlighet	Konsekvens	Mottiltak
1.	Bestillingsproblem	Lav	Forsinker utviklingen av produkt	Få tak i det meste gjennom HVL
2.	Problemer med kode	Høy	Forsinkelse i fremdriftsplan	Være nøye og øke arbeidsmengde dersom det trengs
3.	Fravær	Lav	Forsinkelse	Godt samarbeidsmiljø og avtalte møtepunkt
4.	Komponent skade	Middels	Ødelegger produktet	Kjøpe inn «backup» komponenter
5.	Problemer med programvare	Middels	Vi må finne nye løsninger	Sette seg godt inn i den programvare som brukes
6.	Økonomisk problem	Lav	Overskriding av budsjett	God planlegging
7.	Tap av data	Lav	Måtte gjøre ting på ny	Lagre arbeid i google docs, gjøre det oversiktlig
8.	Unøyaktig kalibrering	Middels	Tidkrevende recalibrasjoner	
9.	Sykdom	Lav	Kan skape forsinkelser	Være ute i god tid slik at en kan takle uforutsett fravær

Appendiks C Brukerdokumentasjon**C.1 Brukerdokumentasjon****MPU6050***Figur 12***Skjemategning MPU6050**

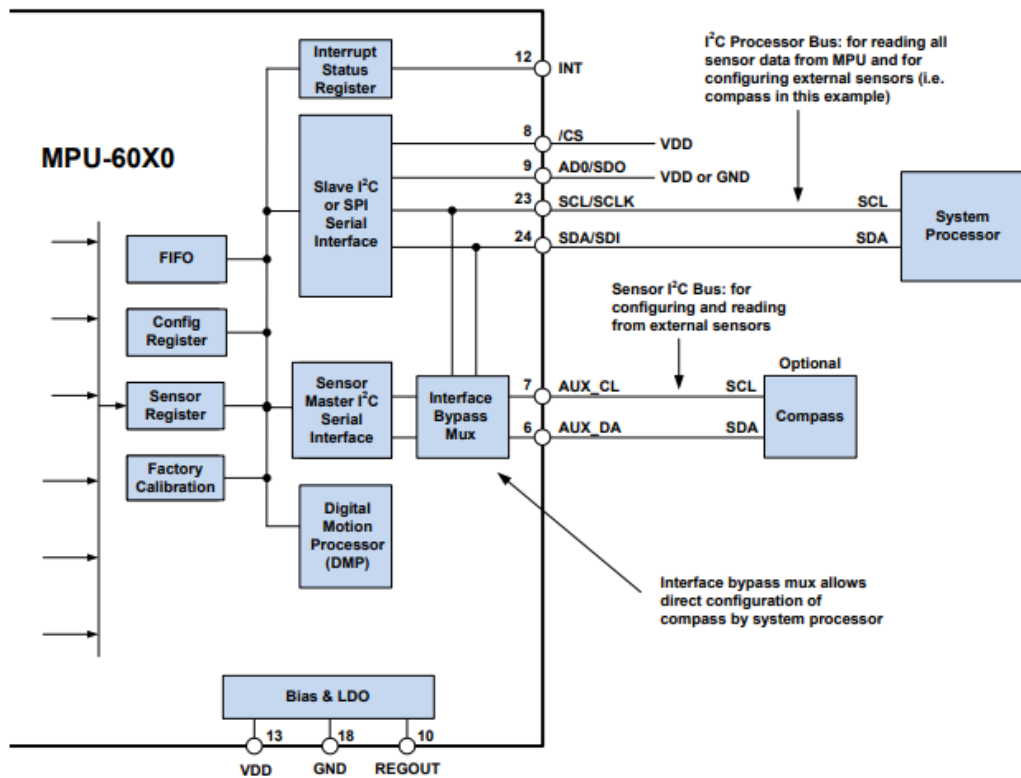
7.5 Block Diagram



Note: Pin names in round brackets () apply only to MPU-6000
 Pin names in square brackets [] apply only to MPU-6050

Figur 13

Blokkdiagram for MPU-60X0 moduler



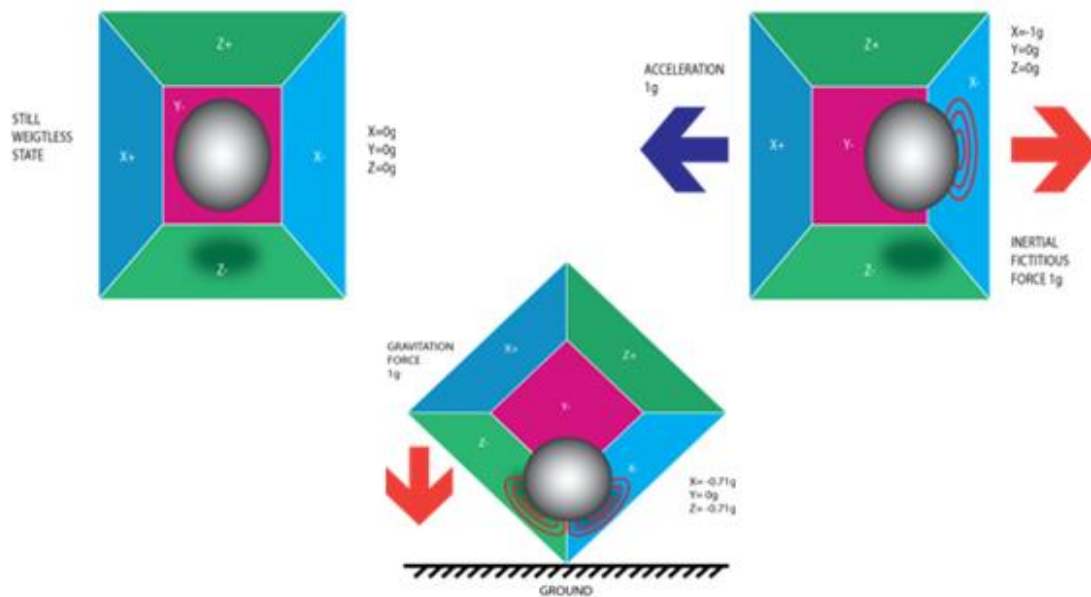
Figur 14

Blokkdiagram for tilkobling til eksternt system

Akselerometer

Et akselerometer er en innretning som registrerer eller reagerer på endringer i fart. Prinsippet for målingene er at én del beveger seg fritt, mens en annen del er fast mot underlaget som skal måles, og avstanden eller kraften mellom disse måles.

Dette akselerometeret virker på prinsippet om piezoelektrisk effekt. Se for deg en kube men en liten kule inni. Veggene til denne kuben består av piezoelektriske krystaller. Når man stiller kuben på skrå vil kulen bevege seg etter hellingen på grunn av tyngdekraften. Når kulen kolliderer med en av veggene vil det induseres små elektriske strømmer. Det er tre par med motstående vegger inne i kuben. Hvert par korresponderer med en akse i 3D-rom: X-, Y- og Z-aksen. Avhengig av strømmene produsert av de piezoelektriske veggene kan vi bestemme retningen på helling og dens magnitudo.



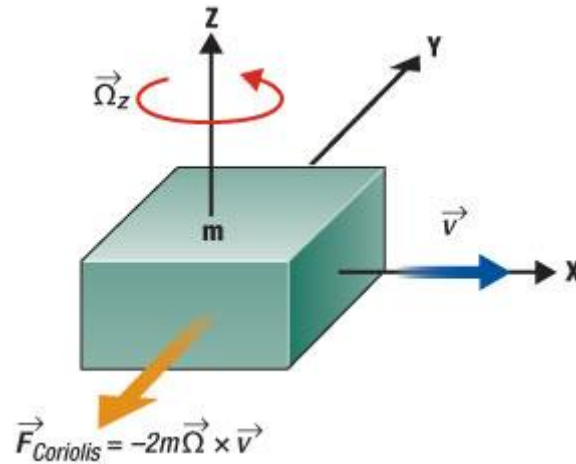
Figur 15

Prinsippskjema for piezoelektrisk akselerometer.

Gyroskop

MEMS gyroskop gjør betydelige fremskritt mot høy ytelse og lavt strømforbruk. De er masseprodusert til lav pris med liten formfaktor som passer til forbrukerelektronikk markedet.

MEMS gyroskop bruker Coriolis-effekten til å måle vinkelhastigheten, som vist på bildet under.

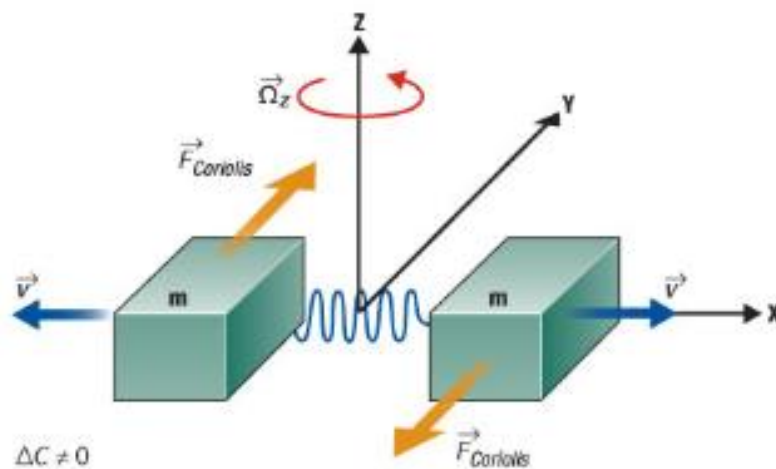


Figur 16

Coriolis-kraften.

De fleste tilgjengelige MEMS-gyroskop bruker en tuning fork-konfigurasjon. To masser svinger og beveger seg konstant i motsatte retninger. Når vinkelhastigheten påføres, virker Coriolis-kraften på hver masse også i motsatt retning, noe som resulterer i kapasitansendring. Denne differensialverdien i kapasitans er proporsjonal med vinkelhastigheten Ω og blir deretter omgjort til utgangsspenning for analoge gyroskop eller LSBer (Least Significant Bit) for digitale gyroskop.

Når lineær akselerasjon påføres to masser, beveger de seg i samme retning. Derfor vil det ikke oppdages noen kapasitansforskjell. Gyroskopet vil utgjøre nullhastighetsnivå for spenning eller LSB, noe som viser at MEMS-gyroskop ikke er følsomme for lineær akselerasjon, slik som tilt, sjokk eller vibrasjon.



Figur 17

Når vinkelhastigheten påføres.

MEMS gyroskop kan måle vinkelhastighet. Digitalkameraer bruker gyroskop til å oppdage håndrotasjon for bildestabilisering. Et gyroskop med yaw rate kan brukes i biler for å aktivere bremsesystemet for elektronisk stabilitetsregulering (ESC) for å unngå at ulykker oppstår når bilen har en skarp sving. Og et rullegyroskop kan brukes til å aktivere kollisjonsputer når en overgangstilstand skjer.

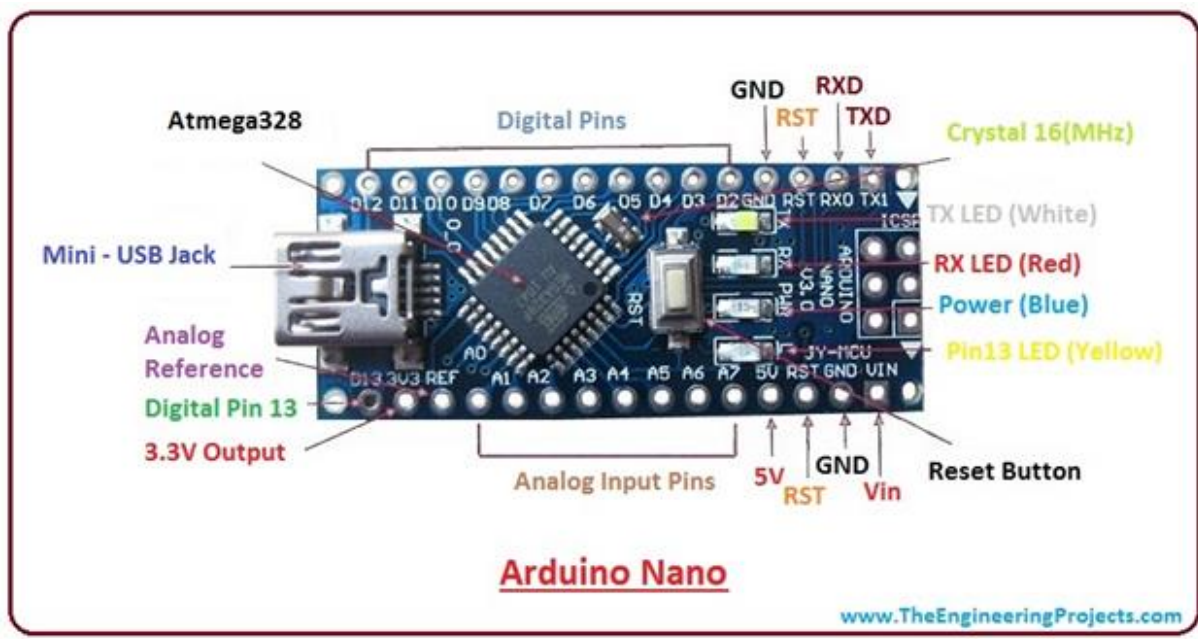
Digital Motion Processor

Den innebygde Digital Motion Processor (DMP) befinner seg innenfor MPU-60X0 og offloader beregninger av bevegelsesprosesseringsalgoritmer fra vertsprosessoren. DMP henter data fra akselerometre, gyroskop og ytterligere tredjeparts sensorer som magnetometre, og behandler dataene.

De resulterende dataene kan leses fra DMPs register, eller kan “buffes” i en FIFO. DMP har tilgang til en av MPU’ens eksterne pinner, som kan brukes til å generere interrupts. Formålet med DMP er å avlaste både tidskrav og prosessorkraft fra vertsprosessoren. Vanligvis bør bevegelsesprosesseringsalgoritmer kjøres med høy hastighet, ofte rundt 200 Hz, for å gi nøyaktige resultater med lav ventetid. Dette kreves selv om applikasjonen oppdateres med en mye lavere rate; For eksempel kan et lavtstrøms brukergrensesnitt oppdatere så sakte som 5Hz, men bevegelsesprosesseringsen skal fortsatt kjøre ved 200Hz. DMP kan brukes som et verktøy for å minimere strøm, forenkle timing, forenkle programvarearkitekturen og lagre verdifulle MIPS på vertsprosessoren for bruk i applikasjonen.

Arduino nano

Arduino nano er et kompakt arduino-kort som er let å bruke på en bunnplate(breadboard). Det består av 14 digitale pins og 8 analoge pins, de kan brukes til forskjellige ting, men er i hovedsak brukt som innganger og utganger. Den har en klokkehastighet på 16 MHz og 16 KB flash-minne. Den anvendes mye i automasjon, robotikk, kontrollsystemer og instrumentasjon. De skal være lette å bruke og gode å bruke i introduksjonsfasen til ett av disse feltene. Nanoen har samme funksjonalitet som UNO men er mindre som gjør den mer allsidig, i elektroniske komponenter er størrelse vanligvis ett viktig tema. Funksjoner som `pinMode()` og `digitalWrite()` brukes til å kontrollere de digitale operasjonene mens de analoge blir styrt av `analogRead()`. Alt dette blir programmert i arduino IDE som står for «Integrated Development Environment» som kan kjøre både online og offline. Som vist på figuren over bruker den mini-USB for tilkoblinger.



Figur 18

I tabellen under vises spesifikasjonene til Arduino Nano, der skiller de mellom Atmega328p og Atmega 168. Vi benytter oss av Atmega328p og har dermed for eksempel 1KB SRAM

Microcontroller	Atmega328p/Atmega 168
Operating Voltage	5V
Input Voltage	7 – 12 V
Digital I/O Pins	14
PWM	6 out of 14 digital pins
Max. Current Rating	40mA
USB	Mini
Analog Pins	8
Flash Memory	16KB or 32KB
SRAM	1KB or 2KB
Crystal Oscillator	16 MHz
EEPROM	512bytes or 1KB
USART	Yes

Arduino Nano Specifications

www.TheEngineeringProjects.com

Figur 19