NTNU
Det skapende universitet

Robotisert vedlikehold

Petter Aspunvik

Master i teknisk kybernetikk
Innlevert: juni 2013
Hovedveileder: Tor Engebret Onshus, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Oppgavetekst

Oppgaveteksten til denne oppgaven er som følger:

> Plassere systemet som allerede er der på en mobil enhet som så kan styres fra samme datamaskin som brukes til å styre armen, samtidig som det legges til rette for å gjøre den autonom.
>
> Bygge opp vognen med kontrollerkort slik at den enkelt kan styres fra den medfølgende datamaskinen. Velge passende hardware til nevnte datamaskin. Lage et program som står separat fra manual move programmet som styrer robotarmen, Det skal også kunne ta input fra ande kilder enn klienten, slik at roboten kan kjøre autonomt rund med input fra kartlegging Software.
>
> For å gjøre den autonom vil det treges flere sensorer som kan bli brukt opp mot det samme kontrollerkortet som sender PWM signaler til motorkontrolleren, disse må også komunisere med en datamaskin

Denne må sees i sammenheng med oppgaveteksten til prosjektoppgaven denne masteroppgaven bygger på:

> Målet med oppgaven er å få systemet til å fungere slik det gjorde under forrige master oppgave, klargjøre systemet til bruk på mobil enhet og finne en passende plattform for dette.
>
> Da det ikke er praktisk med kabler koblet til vogner som kjører rundt og utfører handlinger er det klart at overføringen av data må være trådløs og enheten batteridrevet. Den trådløse overføringen må være sikker og det er ønskelig at enheten skal kunne brukes i noe tid før batteriet må lades.

Systemet som allerede er der henviser her til systemet utviklet i prosjektoppgaven, denne ligger vedlagt på DVD under mappen DVD_prosjekt. Systemet som omtales er en robotmanipulator av type Intelitek Scorebot er 4u. Vognen er den samme vognen som ble funnet og bestilt i prosjektoppgaven, dette er i form av et byggesett som består av en aluminiums chassis med fire likestrømsmotorer og motorkontrollere for disse fire motorene.

Sammendrag og konklusjon

Robotmanipulatoren har en vogn som kan kjøre den rundt slik at den når flere plasser enn hvis den hadde vært fastmontert. Dette var hovedmålet med oppgaven og det anses som oppfylt. Programmet som styrer vognen er fullstendig separat fra programmet som styrer robotarmen. De kunne ha vært på hver sin datamaskin men siden begge datamaskinene må være med på vogna er det enkleste å ha det på samme datamaskin. Videre er grensesnittet for å styre motorene til vogna en enkel COM port i Windows som styrer motorene og en annen som leser av rotasjons enkoderene for å måle avstanden roboten har beveget seg. Alt i alt funger det som er satt på ekstra, det finnes noen små bugs, men de er ikke verre en å handtere for å få et godt resultat for brukeren.

Det er dessverre problemer med både pan-tilt enheten og Internal measurement uniten (IMU) som skal styre denne, manual move programmet som skal ta inn signalene fungerer ikke hvis tråden som tar disse signalene inn startes opp, og pan-tilt enheten klarer ikke lenger å «home» seg inn riktig. Dette er dog mindre problematisk da IMUens oppgave var å styre pan-tilt enheten, så de resterende undersystemene fungerer som ønsket. Men alt i alt fungerer det resterende systemet tilfredsstillende.

Summary and conclusion

The robot manipulator now has a wagon that gives it greater reach, which was the point of this thesis. The program that controls the movement of the wagon is completely separated from the program that controls the robot manipulator. This modularity makes it possible to control the robot manipulator and the wagon form separate computers. But due to space constrictions they are controlled by the same computers. There are also fitted rotary encoders to register movement and distance traveled for programs that use this. Apart from some small bugs with the firmware with the motor controller cards all the functions works.

The pan- tilt- unit already mounted on the robot manipulator does not home any more, this is a mild inconvenience as the tread to read the Internal measurement unit (IMU) that controls the pan- tilt- unit no linger functions. But as a whole the system works satisfactory.

Forord

Denne rapporten er skrevet i 10. og avsluttende semester av masterutdanningen i Teknisk Kybernetikk ved Norges Teknisk Naturvitenskapelige Universitet. Den teller 30 studiepoeng og er utført ved Institutt for Teknisk Kybernetikk i løpet av vårsemesteret 2013
Rapporten presenter en fungerende mobil robot med mulighet for styring av en robotmanipulator og kontroll av vognen over trådløst nettverk styrt

Jeg vil rette en takk til min veileder, Professor Tor Onshus for god veiledning gjennom semesteret, jeg ønsker også å takke instituttets it avdeling og mekaniske verksted.

Petter Aspunvik

Trondheim 25. juni 2013

Innholdsfortegnelse

Figurliste

1 Introduksjon

1.1 Motivasjon

Prosjektoppgaven utført i 9. semester (1) danner grunnlaget for denne masteroppgaven. Den fjernstyrte robotmanipulatoren benyttet i min prosjektoppgave trengte en fungerende transportvogn. Robotsystemet som dette blir er tiltenkt å kunne videreutvikles av andre til å kunne gjøre den fullstendig autonom. Resultatet fra denne oppgaven blir et nytt grunnlag, et fjernstyrt bevegelig robotsystem som andre kan bygge videre på.

1.2 Tidligere arbeider og utgangspunkt

Denne oppgaven bygger videre på prosjektoppgaven skrevet av Petter Aspunvik høsten 2012 (1). Prosjektoppgaven bygger igjen på masteroppgaven skrevet av Kristian Saxrud Bekken våren 2010 ved NTNU (2), som igjen bygger på tildligere oppgaver hvor den ene søkte å lage en løsning for å styre robotarmen som er brukt i dag fra en datamaskin. En annen oppgave var å lage et «teleprecence» system hvor man kunne sitte med en virtual reality hjelm og få sterioskopisk bilde fra en annen datamaskin sendt over nettverket. Dette systemet hadde også en IMU på VR-hjelmen slik at det kunne detekteres om brukeren snudde på hodet, dette ble så brukt til å styre en pan-tilt-enhet som kunne pane og tilte kameraene, noe som gir bruerken en følelse av å være til stede. Disse to sytemene; robotmanupulatoren kontrollert av en datamaskin og «teleprecence» systmet, ble så slått sammen til en enhet.

1.3 Teori

1.3.1 Teori om PWM

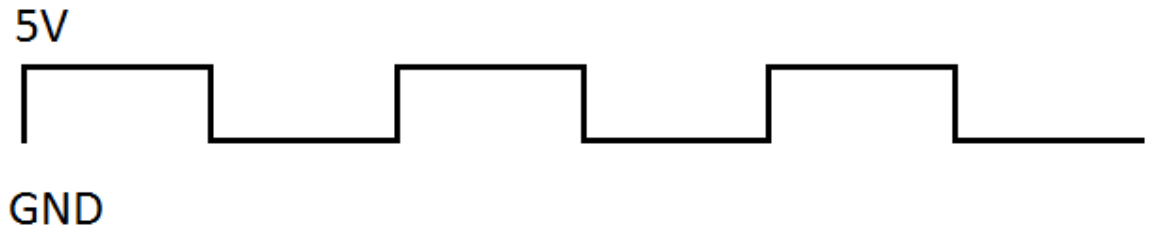
PWM står for «Pulse With Modulation», dette kan oversettes til puls bredde modulasjon. Det er en metode å modulere et spenningsnivå for enten å overføre informasjon eller endre mengden energi som overførers i en tidsperiode.

PWM gjøres mellom to spenningsnivåer, typisk jord og maks spenning. I tillegg til disse nivåene trengs det en bærefrekvens som holdes konstant, en mulighet er 1 kHz. Man bestemmer så «duty cycle» som sier noe om hvor lenge spenningen er maks. Oppgitte verdier for «duty cycle» er enten prosent «%» eller nøyaktig tid «s». Når det oppgis % er dette hvor mange % av tiden innen en puls fra bærebølgen hvor spenningen er maksimal. Oppgis den nøyaktige tiden har man ikke nødvendigvis noe mer informasjon om bærebølgen da man ikke har noen informasjon om hvor mye signalet befinner seg på den lave spenningen før det settes opp til maks for neste puls.

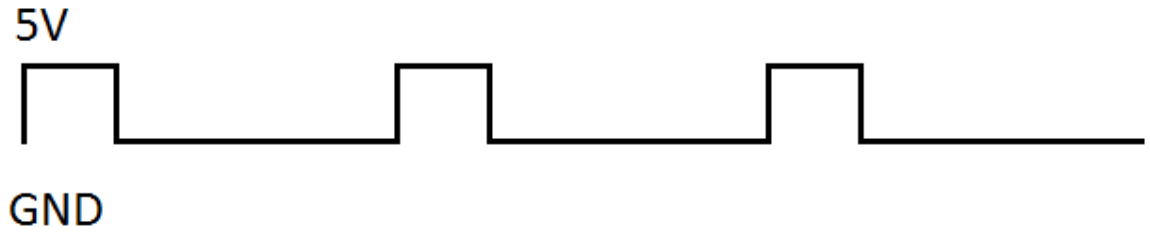
Sammenhengen mellom %, absolutt tid og bærebølgefrekvens er som følgende:

$$\% = s * Hz$$

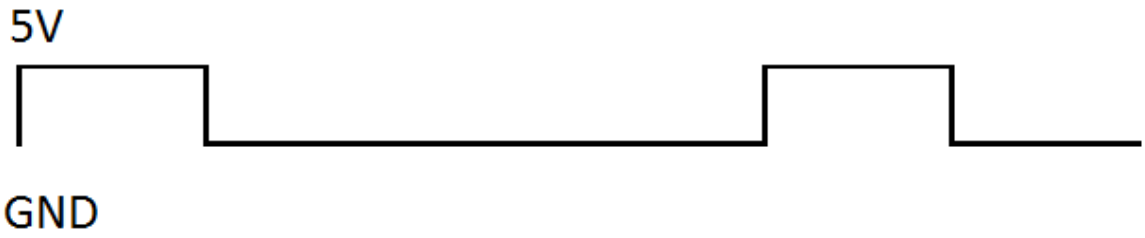
Dette hvor:
% = prosent av tiden hvor signalet er maks, enhetsløs
s = nøyaktig mengde tid signalet er høyt, enhet sekunder
Hz = bærefrekvens, enhet 1/sekunder



Figur 1 PWM signal med 50 % duty cycle



Figur 2 PWM signal med 25 % duty cycle



Figur 3 PWM signal med 25 % duty cycle

Hvis figur 1 tilsvarer et PWM signal med en bærefrekvens på 1 kHz og har 50 % duty cycle tilsvarer det at signalet er høyt i 0,5/1000 [1/s] = 0,0005 s = 0,5 ms.

I figur 3 har frekvensen blitt halvert, til 500 Hz, noe som sammen med en duty cycle på 25 % gir 0,25 / 500 [1/s] = 0,0005 s = 0,5 ms Som periode tid, noe som er det samme som i fig 1. Forskjellen ligger i hvor lenge signalet er lavt etter perioden det har vært høyt.

Hvis PWM brukes til å overføre energi kan det i praksis gjøres på to måter, man kan fore firkantpulsen rett til det som skal drives, eller koble på et lavpassfilter. Brukes et lavpassfilter på knekkfrekvensen til dette filteret settes vesentlig lavere enn bærefrekvensen. Spenningen som da gis fra filteret vil være nærmere en likespenning gitt av denne formelen (3):

$$V_{ut} = V_{inn} * \%$$

Hvor:
V_{inn} = maks spenningen inn, «høy» verdi for PWM signalet
V_{ut} = spenningen filteret gir ut
% = duty cycle til PWM signalet

Det er alltid noe tap i den virkelige verden. Dette fordi spenningen ikke endres momentant men som en meget bratt endrings. Dette fører til at V_{ut} aldri være fult så høy som i regnestykket over. Akkurat hvor mye lavere kommer an på «switshe tapet» som kommer av transienten som skapes når spenningen går fra høy til lav eller lav til høy. Dette taper er avhengig av slewe raten til elektronikken, impedansen og kapasitansen/induktansen.

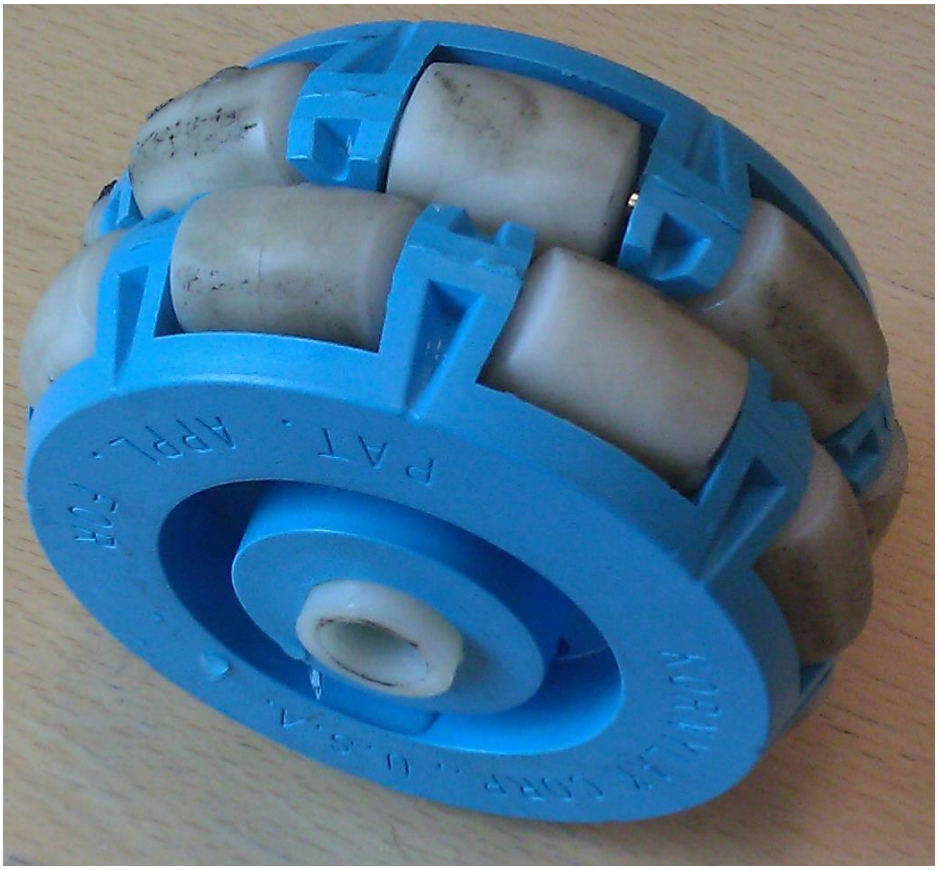
Om det som skal drives har en induktiv del trengs dog ikke filteret, da den induktive delen vil fungere som et eget filter, det er da bare viktig at bærefrekvensen blir så høy at filtereffekten blir den sterkeste, dette kan skape problemer med det overnevnte «switshe tapet».

Ett annet moment med bærefrekvenser er den menneskelige hørselen. Våre ører oppdager lyd mellom 20 Hz og 20k Hz. Utstyr som bruker PWM får ofte en «ratling» på samme frekvensen som bærefrekvensen er på. Så hvis denne ligger i det hørbare området kan dette skape irritasjon for menneskene som skal være rundt dette. Denne «ratlingen» kommer av deler som beveger seg og det er da viktig at utstyret tåler denne ekstra påkjenningen, hvis ikke vil det slites ut raskere enn ved konstant spenning.

1.3.2 Teori om omnihjul

Omnihjul er hjul som kan rulle i flere retninger på en gang. Dette gjøres ved at det plasseres rullere langs hjulets kontaktflate hvor aksene som hjulene ruller rundt IKKE er parallell med aksen som

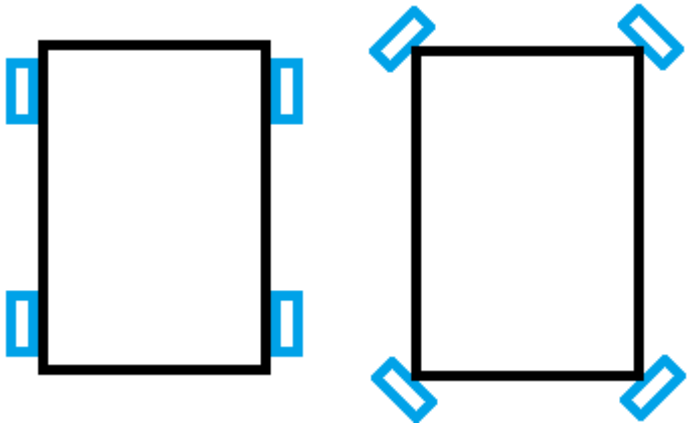
rullerende roterer rundt. Dette fører til at hjulets bevegelse kan være i flere retninger uten stor motstand.



Figur 4 Omihjul brukt på robotvognen

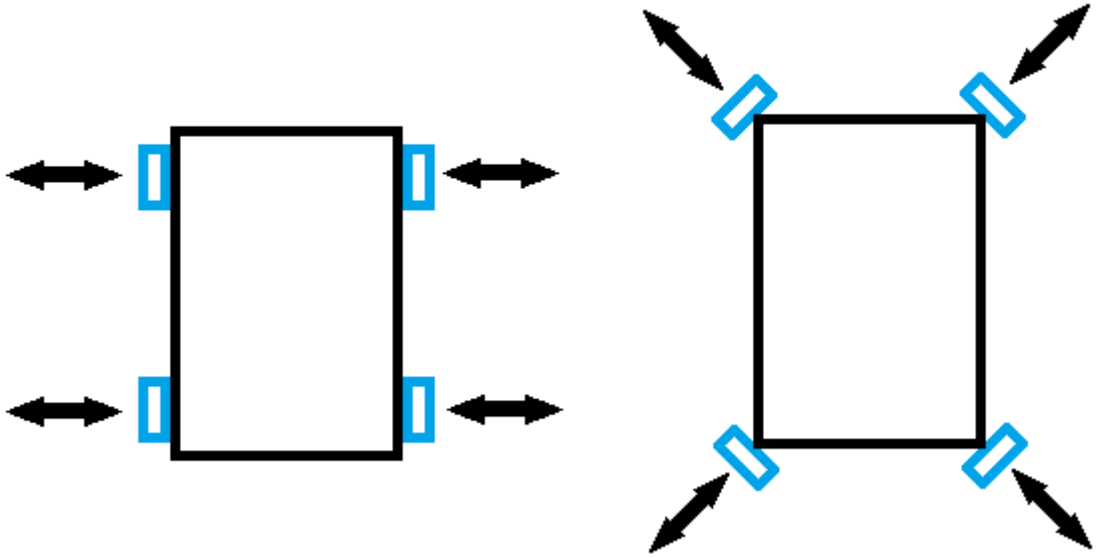
Dette kan også være en ulempe hvis aksen som hjulene roterer rundt er parallelle i det punktet hvor rullerene er i kontakt med bakken da det ikke vil være noen motstand for å rulle på rullerne.

Konstruksjonen gjør at det finnes to måter å feste hjulene på et chassis, enten langs sidene, slik som vanlige hjul, eller i hjørnene. Det vil også være mulig å bruke en kombinasjon av dette.



Figur 5 Plassering av omnihjul på vogn

Figur 5 viser hvor hjulene kan plasseres på vognen, fordelen med plasseringen vist til høyre, atlså hjørnene er at roboten ikke sklir til høyre eller venstre.



Figur 6 Omnihulenes ukontrolerte bevegelse

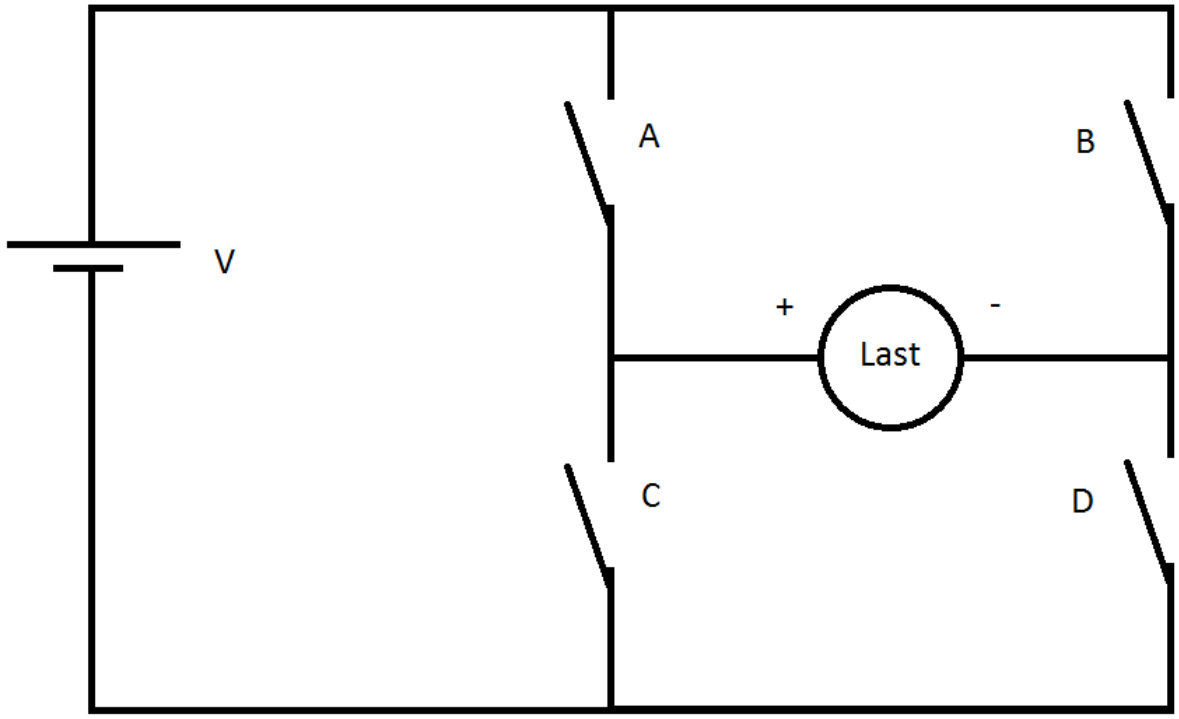
I figur 6 er den retningen omnihjulene ikke kan kontrollere sin bevegelse i tegnet inn med piler. I metoden til høyre ser vi at alle fire hjulene tillater bevegelse til høyre eller venstre. Mens i eksempelet til høyre ser vi at det alltid er to hjul som stopper de andre to hjulene.

En annen bakdel med omnihjul er at de ikke er like jevne som normale hjul noe som skaper mer vibrasjoner. De er også mer kompliserte med det ekstra laget av hjul på hjulet. Alt dette gjør at de ikke egner seg for høye hastigheter. Det er derimot ikke noe problem å bruke dem inne da hastigheter inne i bygninger hvor mennesker og robot oppholder seg i samme rom ikke burde være høyere enn gangfart (ca 5 km/t).

Fordelen ligger i at begge oppsettene i figur 5 kan rotere om sin egen akse, og gjøre dette uten skade på gulvet.

1.3.3 Teori om H-bro

En H-bro er en enkel innretning som benyttes til å endre polariteten fra en likestrømskilde. Kjernen er fire brytere koblet i en H med lasten mellom seg (4), Figur 7 er en enkel H-bro.



Figur 7 Enkel H-bru

Tabellen under beskriver hvordan kombinasjoner av bryterne lager forskjellige effekter for lasten:

A	Lukket	Åpen	Lukket	X	Lukket	Åpen
B	Åpen	Lukket	X	Lukket	Lukket	Åpen
C	Åpen	Lukket	Lukket	X	Åpen	Lukket
D	Lukket	Åpen	X	Lukket	Åpen	Lukket
Resultat Last	Spenningen for lasten blir V	Spenningen for lasten blir -V	Kortslutning av V	Kortslutning av V	Kortslutning av Lasten	Kortslutning av Lasten

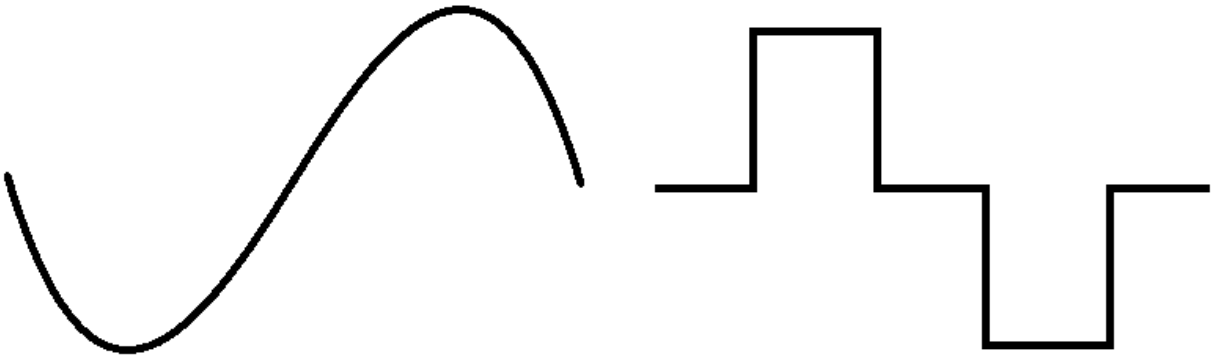
I tabellen betyr «Lukket» at bryteren leder strøm, «Åpen» at den ikke leder strøm og «X» at det kan være begge tilsandene. Det er vær å merke seg at det ikke har mye effekt å lukke kun en av bryterne i en H-bro, da det vil føre til at en av koblingspunktene til lasten vil ligge flytende. Det er vanlig at H-broer lages med halvlederbrytere istedenfor fysiske brytere, gjerne i en fysisk pakning som da også kan ha mer avanserte funksjoner som strømmåling og beskyttelsesutkobling.

Typisk bruksområde for H-Broer er i dag for styring av likestrømsmotorer slik at de kan rotere begge veier, og til bruk i DC til AC invertere.

1.3.4 Teori om DC til AC invertere

DC til AC inverter er en innretning som genererer vekselstrøm fra likestrøm, Typisk går det fra enten 12V eller 24V (bil og lastebil batterier) til 115V /60Hz eller 230V / 50Hz vekselstrøm. Det finnes i hovedsak to typer invertere: modifisert sinusbølge og ren sinusbølge. Forskjellen mellom disse ligger i hvordan spenningskurven til vekselstrømmen ser ut. Bølgeformen fra inverterene av typen ren sinusbølge har en tilnærmet perfekt sinusform, derav navnet, mens invertere av modifisert

sinusbølge typen har en bølgeform som ligner på en firkantpuls, men med et område mellom pulsene hvor signalet er 0 volt.



Figur 8 Ren sinus vs modifisert sinus

Bølgeformen til venstre i figuren over tilsvarer en ren sinus, mens bølgeformen til høyre tilsvarer en modifisert sinus.

En uønsket effekt hos modifisert sinus formen er at de direkte overgangene skaper store mengder støy i enheter med kapasitiv og/eller induktiv last. THD Støyen ligger på 45% av signalet for modifisert sinus (5). THD Støyen på den tredje overharmoniske ligger på ca 33% (5). En annen utfordring er at følsom elektronikk som benytter overgangen fra positiv til negativ halvperiode for synkronisering og tidskontroll vil få problemer når det ikke er ett gitt tidspunkt men en legere periode som ligger ved null volt signal.

Den eneste fordelen den modifiserte sinusbølge inverteren har er pris, den er enklere å bygge da den kun består av to hovedkomponenter, en boost regulator for å øke spenningen fra likestrømmen til den ønskede toppverdien på kurven og en H-bro for å snu polariteten til nevnte regulator. En Ren sinus inverter kan enten bygges som en eller flere modifiserte sinus invertere med kraftige filetere for å glatte kurven, eller som en motor som drar en generator slik at den genererer vekselstrømmen. Begge løsningene er dyrere og mer komplekse enn den enkle modifiserte sinus inverteren.

1.3.5 Teori om Xmega prosessoren

Atmel X-mega er en prosessorfamilie med både 8 og 16 bit mikrokontrollere med et stort utvalg av funksjoner, i denne oppgave er det bare benyttet tre av disse og de vil derfor bli nærmere forklart i denne teksten. De tre funksjonene er USB tilkobling, PWM utganger og intern watchdog. All informasjonen i dette delkapittelet er hentet fra databladene xmega256A3 intro.pdf og xmega256A3 alt.pdf som befinner ser på den vedlagte DVDen under Dokumenter og annet\Datablader, disse var de nyeste når oppgaven startet, men oppdaterte dokumenter kan lastes ned fra (6).

1.3.5.1 PWM

PWM systemet til Xmega prosessoren går på «peripheral clock» og denne må derfor settes på før det kan brukes.

PWM signalet genereres med en teller, denne teller fra null til en forutbestemt verdi mellom en og 65535 (2^{16} – 1 da registeret er 16 bit). Tellingen kan skaleres med en «prescaler» av enten 1, 2, 4, 8, 64, 256 eller 1024 klokkeslag på «peripheral clock» det tar å telle «en» oppover.

Genereringen av PWM signalet starter med at telleren en null, signalet er da høyt. Telleren teller oppover til den når et bestemt tall mellom null og det forutbestemte tallet telleren teller til. Når dette skjer blir signalet endret til lavt frem til telleren starter på null igjen.

«duty cycle» til signalet blir da: $\frac{Tall}{Top} = \%$

Hvor:
Top er det strøste tallet telleren teller til
Tall er tallet hvor signalet går fra høyt til lavt når telleren når det
% er «duty cycle»
samtlige enheter er enhetsløse

Bærefrekvensen for PWM signalet blir da: $\frac{Hz}{Pre*Top} = bHz$

Hvor:
Hz er frekvensen til «peripheral clock»
Pre er «prescaler» verdien
Top er det strøste tallet telleren teller til
bHz er bærefrekvensen til PWM signalet
Hz og bHz har enhten 1/s mens Pre og Top er enhetsløse.

Et eksempel på det over ville være:
Top satt til 2000
Pre satt til 8
med en «peripheral clock» på 16 MHz

Dette gir en bærefrekvens på $\frac{1600000}{8*2000}$ = 1000 altså 1kHz

Hvis Tall settes til 800 vil da «duty cycle» bli $\frac{800}{2000} = .4$ altså 40 , ved en bærebølge på 1Khz tilsvarer dette en tid på $\frac{.4}{1000} = .0004$ eller .4 ms hvor signalet er høyt

1.3.5.2 USB

xmega256A3BU mikrokontrolleren har en USB til seriel krets innebygd, denne må konfigureres og settes opp korrekt slik at mikrokontrolleren skal oppføre seg som en standard com enhet i Windows med de korrekte driverne. Alt dette har Atmel allerede laget et meget hendig eksempel på i sitt Atmel Software Framework (7), dette er kode som Atmel legger ut slik at det kan forenkle bruken av deres mikrokontrollere. Koden er også i endring da Atmel retter bugs hvis de blir rapportert inn. Denne eksempelkoden genererer også en driver for Windows når den kompilerers i Atmel studio.

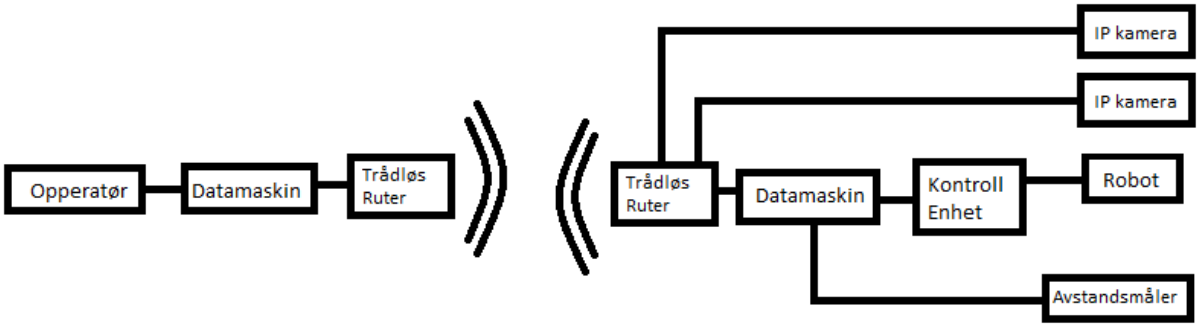
1.3.5.3 Intern Watchdog

Mikrokontrolleren har også en funksjon kalt watchdog, dette er en timer som, etter at den er startet, restarter hele mikrokontrolleren hvis den når null. Dette kan brukes til kontroll av enheten slik at den går til en forutbestemt sikker tilstand hvis noe skulle feile i kode eller hardware. Den drives av en innebygd 1 kHz laveffekts oscillator på mikrokontrolleren. Denne oscillatoren er ikke veldig nøyaktig, noe som kan føre til at watchdog trimeren ikke er like nøyaktig som mye annet på denne mikrokontrolleren.

1.4 Komponentbehov

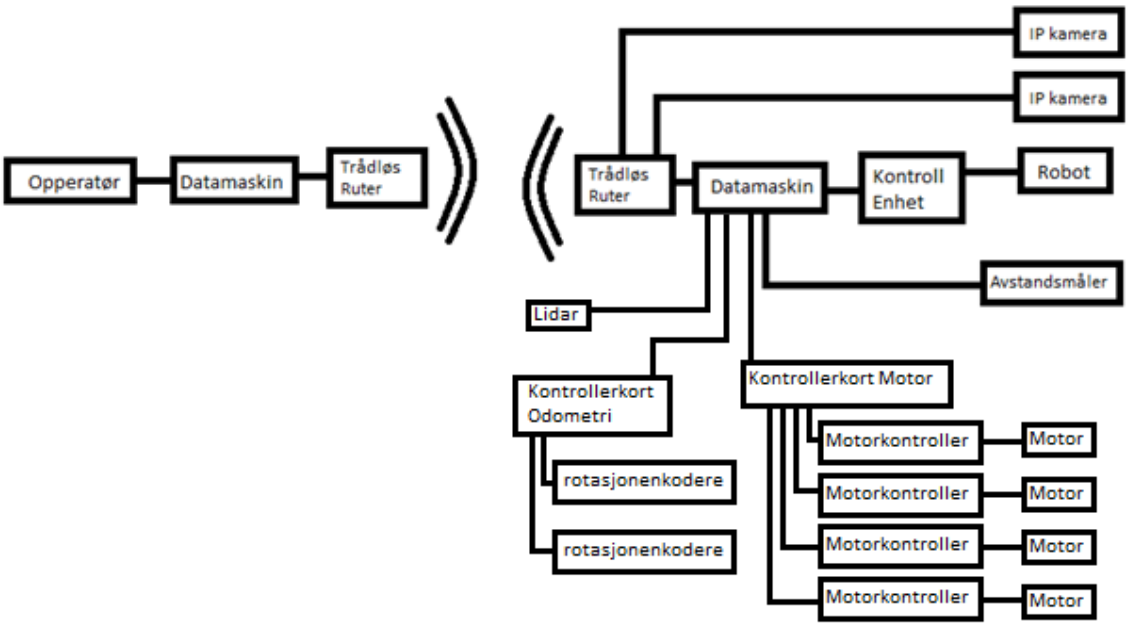
I denne delen av oppgaven defineres og beskrives hva som trengs av komponenter slik at vognen til roboten kan benyttes slik det er tiltenkt, som en bevegelig base for robotmanipulatoren.

Systemet på starten av oppgaven hadde en slik oppbygning:



Figur 9 Systemet ved oppgavens start

Avstandsmåleren er her montert på selve robotarmen for kollisjonsdeteksjon. I tillegg til dette ville det trenges et kontrollerkort som kunne styre de fire individuelle motorene som driver hvert sitt hjul. Det er fire motorer koblet til hvert sitt hjul for enkelthetens skyld da det bare er en kort aksling skrud fast på akseltappen til motoren som igjen er skrud fast i vognen. Dette fører til at vekten av hele systemet overføres via motorfestene og motoren til hjulene.

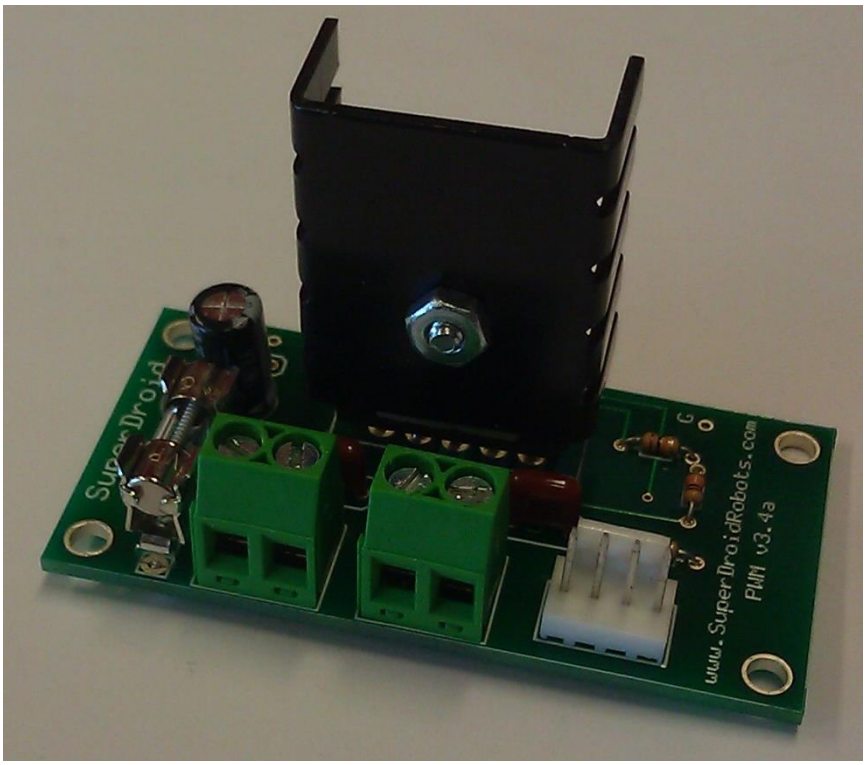


Figur 10 Systemet ved oppgavens slutt

1.4.1 Styring av motorkontrollere

Motorkontrollene består i hovedsak av en H-bro som styres av et PWM signal, et retningssignal og et bremsesignal. «duty cycle» til PWM signalet bestemmer hvor mye av tiden H-broen leder strøm til motorene, mens retningssignalet bestemmer polariteten. Bremsene kobler begge kontaktpunktene på motoren sammen slik at den bremser, det er da ikke mulig å ha drivkraft på motoren.

Strømmen til motorene kommer fra et 24V 9,6Ah Litium ion fosfor jern batteri (LiFoFe). Dette gir en vesentlig lavere spenning enn de 55V som er maks hva H-broen tåler. Motorene trekker 2,3 A hver ved rusing (når de står stille men har full spenning over seg). Videre er det montert en sikring på kontrollerkortet på 3,15 A for å sikre seg mot kortslutninger i ledningen til motoren.



Figur 11 En av motorkontrollerene

De tre overnevnte signalene har en «pull down» mostand på 1 kΩ, dette gjør at de holder seg logisk lave selv om det ikke er noe koblet til den kontakten.

H-broen som siter på kortet er av typen LMD18200, denne tåler opp til 3A og 55V (8), logisk høy for styrestrømmen må være mellom 2V og 12V, logisk lav ligger mellom -0,1V og 0,8V samtidig som den minste pulsbredden på både PWM, retting og bremse må være 1 µs. Maks strømtrekk er i tillegg 10 µA.

Minste pulsbredde på 1 µs gir noen føringer for hvilke bærebølger sammen med hvilken «duty cycle» som kan brukes på denne kontrolleren, hvis vi setter det inn i ligningen fra teorien om PWM får vi at:

$$\%/_{Hz} = 1\ \mu s$$

Hvor den laveste %, «duty cycle», delt på bærefrekvensen ikke kan være lavere enn 1 µs for at kontrolleren skal oppdage den. Ved en bærebølge på 1KHz blir dette

$$\% = 0{,}000001s * 1000s^{-1} = 0{,}001$$

Altså kan den minste «duty cycle» være på 0,001, eller en promille ved 1kHz.

En mikrokontroller som skal styre disse kontrollerene vil igjen måtte kunne enten kobles til en datamaskin eller direkte på et ip-nettverk for å kunne mota komadoer fra andre systemer. Typisk for mikrokontrollere er at de har en seriell port som kan mota og sende data til normale desktop systemer. Det er derimot en utfordring at seriellporter normalt ikke lenger er å finne på standard datamaskiner. Det vil da være en fordel om kontrolleren enten har en innebygd USB port eller en innebygd USB til Seriell port, et alternativ til dette ville være å skaffe en spesialisert datamaskin som har serieporter.

1.4.2 Odometri

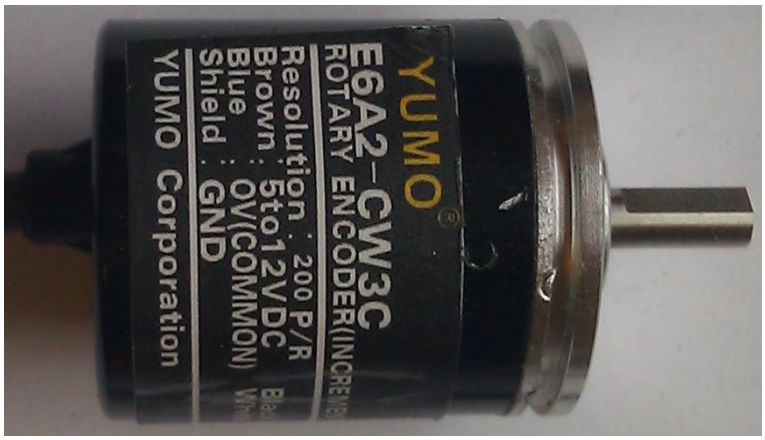
Det er ønskelig med odometri for å hjelpe SLAM algoritmen som Mikael Berg utviklet i sin master (9). Et viktig poeng er at modometrien ikke skal være til hinder for framkommeligheten til vognen som ble satt som et krav i prosjektoppgaven som ledet frem til denne masteroppgaven (1). Det var klart at det var nødvendig med to sensorer, en på hver side av vognen for å detektere svingning i tillegg til frem og tilbake bevegelse.

Løsningene som ble vurdert var: to enkodere hvis hjul ikke hadde pådrag. En løsning med to pc mus som ble holdt over overflaten roboten kjørte over og dermed detekterte bevegelse. Til slutt ble det vurdert å feste en liten metallbit på hjulene som drar roboten frem og detektere denne med et hall element.

	Fordel:	Ulempe:
Hall element	Enkel mekanisk og simpelt å hente ut data med en mikrokontroller	Ikke retningsdeteksjon
PC mus	Meget nøyaktig og bevegelsesdeteksjon i to akser	Meget vanskelig mekanisk løsning og komplisert å hente ut data
Enkodere	Retningsdeteksjon og enkelt å hente ut data med mikrokontroller	Noe komplisert mekanisk

Dette førte til at den mest tids og kostnadseffetive løsningen ble vurdert til å være to enkle rotasjonsenkodere.

Enkoderene er av typen Yumo E6A2 databladet er lagt ved på dvden og ligger i mappen datablader på dvden med navnet Yumo E6A2 datablad.



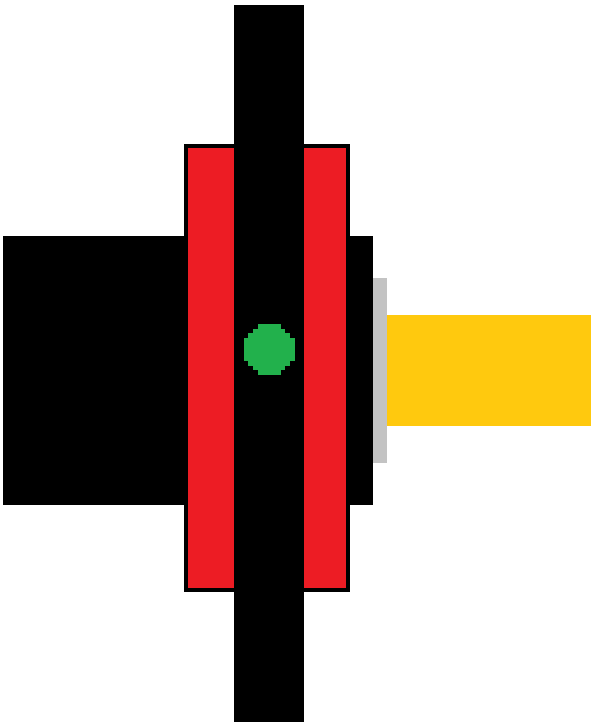
Figur 12 rotasjonenkoder

Enkoderene har fem koblingspunkter, hvor fire av disse har en fysisk tilkobling, Det er to for energitilkobling, 0V og Vcc hvor Vcc er mellom 5V og 12V, og to for gråkoding av data fra selve enkoderen. Denne gråkoden går rundt 200 ganger på en rotasjon, med 2 bit gråkoding blir det da 200 * 2^2 = 800 endringer ved en rotasjon. Med hjul på 10 cm diameter og en topphastighet på ca 1.5 meter i sekundet ville det tilsvare og detektere endringer i gråkoden

$$\frac{1{,}5}{0{,}1 * \pi} * 800 = 3819{,}72$$

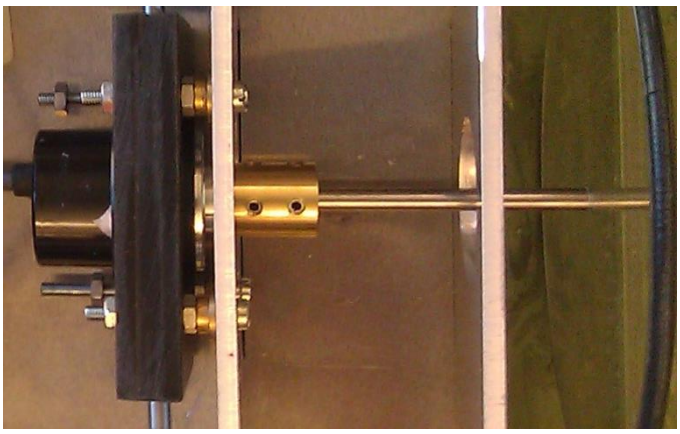
Ca 3800 ganger i sekundet.

Opphenget av enkoderene ble gjort meget enkelt, den er skrudd fast i en holder som kan vippe opp og ned, denne løsningen er sett i profil i figur 13



Figur 13 Oppheng av enkoder

Den røde delen av figur 13 er ringen som kan rotere noen grader rundt det grønne punktet i tegningen, dette gjør at hjulet som er koblet til den gule tappen på figur 13 kan bevege seg ca 4 cm fra høyeste til laveste posisjon.



Figur 14 Oppheng til roasjonenkodere på roboten

Det er vekten av hjulet som holder det på bakken da enkoderene er meget lette.

1.4.3 Valg av datamaskin til styring av robotarm og vogn

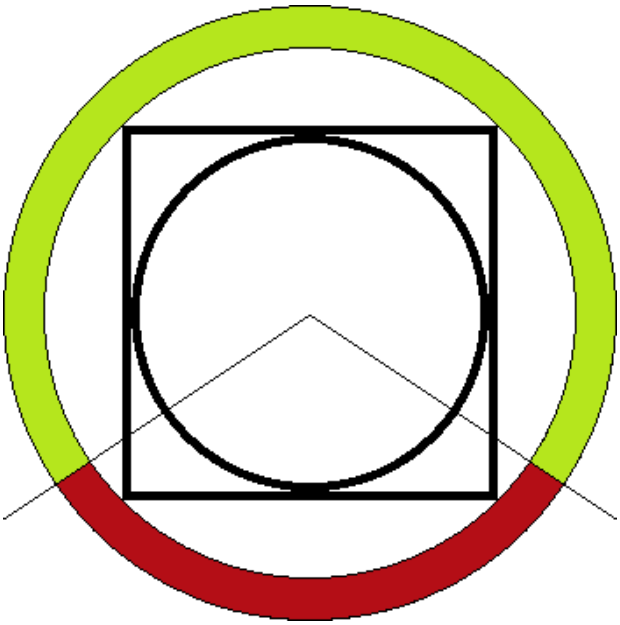
Siden programvaren som kontrollerer robotarmen allerede er skrevet for en x86 maskin som kjører Windows er det logisk å benytte den samme maskinen til alt annet som måtte trenge datakraft på vognen. Det ville dog være fordelaktig om denne maskinen tålte noe mer bevegelse enn en normal, datamaskin da den kom til å bevege seg en god del, og da det ikke er noen fjæring eller støtdempere på vogna vill det riste noe. Ellers vill den trenge USB porter, en til robotarmen, to til LIDAREN som Mikael berg bruker i sin oppgave(en for signal og en for strøm) og en til avstandsmåleren, en seriell port til pan-tilt enheten og en nettverksport så den kan koble seg på ruteren som også må være med.

1.4.4 Energi til datamaskin og robotboks

Kontrollboksen til robotmanipulatoren inneholder også strømforsynginen til denne. Denne strømforsyningen er laget for å fungere på 230 Volt vekselspenning. Da det ikke er ønskelig å plukke denne kontrollboksen fra hverandre siden den teknisk sett kun er på lån fra SINTEF er det nødvendig med en 230 V vekselstrømskilde på vognen. Siden det trengs en 230 V kilde kan denne også brukes til å drive datamaskinen nevnt over, dette gir en del større fleksibilitet i valget av denne datamaskinen. Denne kilden vill da være en inverter som omdanner likestrøm fra et batteri til 230 V vekselspenning.

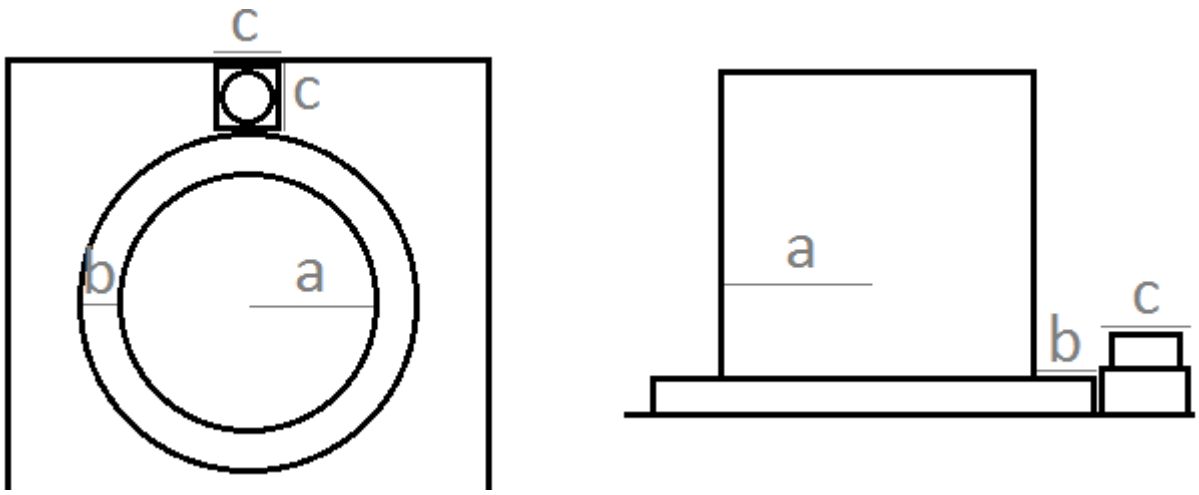
1.5 Plassering av lidar

Det er plassert en lidar på vognen i sammenheng med masteroppgaven til Mikael Berg (9), denne har et synsfelt på 240 grader, dette gir en dødsone på 120 grader bak denne lidaren.



Figur 15 Dødsoner på lidaren

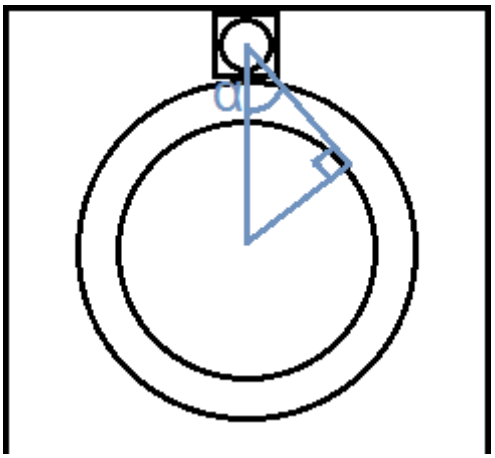
Det er hensiktsmessig å plassere lidaren fremst på vogna da det er minst sannsynlig at noe kommer i veien for den der, men rett bak denne kommer da foten til robotmanipulatoren, om denne ikke haver i dødsone kan dette gi merkelige utslag på algoritmene som benyttes for å bergene rommet. De fysiske målene er som følger:



Figur 16 Mål brukt i beregningen

	Størrelse	Plass på figur 16
Radius robot-manipulatorbase	18,3 cm	a
Leppe robot-manipulatorbase	3 cm	b
Ytre mål Lidar	5 cm	c
Lidar, fra kanten til målepunktet	2,5 cm	d

Selv om lidaren måler 5 x 5 cm måler den fra midten så avstanden i våre beregninger blir 2,5 cm



Figur 17 Vinkel α tegnet inn

Vinkelen vi er interessert i er pekt ut i bildet over som vinkel α. Denne vinkelen må være under 60grader for at foten skal have i dødsonen og dermed ikke bli sett av lidaren. Heldigvis er denne vinkelen nokså enkel å finne, da den inngår i den rettvinklede trekanten vist på figur 17.

Dette kan gjøres da radiusen til en sirkel alltid er ortogonal med kanten på sirkelen.

Dette gir oss følgende beregning:

$$Sin^{-1}\left(\frac{18{,}3}{18{,}3+3+2{,}5}\right) = 50{,}25$$

Altså er vinkelen α = 50,25 grader, noe som er mindre enn 60 grader som viser at robotmanipulatorbasen ligger i dødsonen til Lidaren.

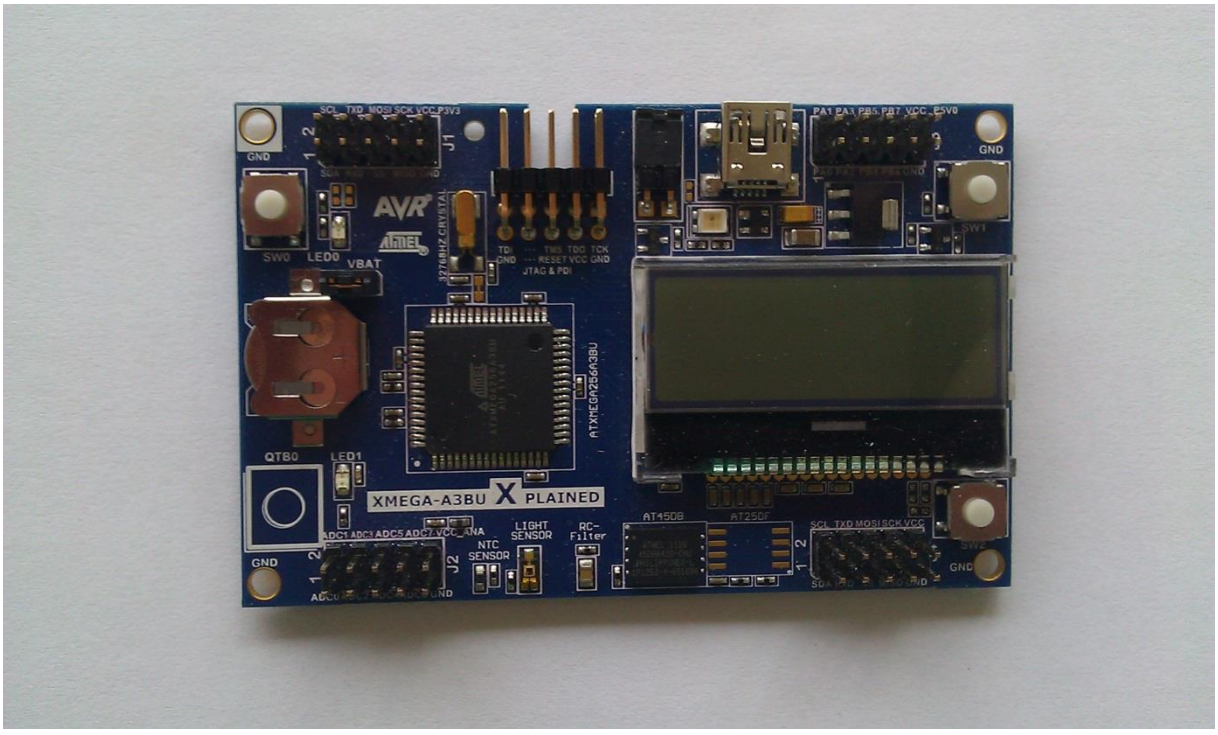
2 Valg av kommenter og deres bruk

2.1 Valg av motorkontrollerkort og kontrollerkort til enkoderene

2.1.1 hvorfor Xmega A3BU xplained kortet

Valget av løsning for styring av motorkontrollene (de som er beskrevet i 1.4.1) var noe av det første som ble gjort, Jeg hadde allerede testet et Atmel xmega A3BU xplained kort til personlig bruk. Her fant jeg at det hadde alle egenskapene som var påkrevd for å kunne fungere som et kontrollerkort. Det ble vurdert opp mot et annet Atmel xmega Xplained kort og en løsning med bruk av labview og dedikert I/O kort i datamaskinen. Xmega kortene ble foretrukket da det virket som en lettere oppgave å lage et enkelt program som tok i mot kommandoer over nettverket og sendte disse videre til kortene og leste av eventuelle verdier. Det var også viktig at det var lett å lage spesifikke programmer som kunne skrive og lese til kontrollerene slik at enkoderene kunne leses av og motorene styres. Dette var lettere med et Xmega xplained kort som kommer med innebygd USB til COM port overgang.

Det sto i hovedsak mellom to kort, enten «Xmega A3BU explained» eller «Xmega A1 explained» A3UB kortet var i mitt eie fra tidligere, personlige, prosjekter mens A1 kortet var tilgjengelig fra NTNU. Begge to inneholdt de ønskede fysiske egenskapene som kortet trengte (en del PWM og generell I/o og USB tilkobling for dataoverføring). Den avgjørende forskjellen var at A3BU kortet hadde et enkelt eksempel på dataoverføring over den virtuelle COM porten (7) som ble opprettet av kortet mens A1 kortet ikke hadde noe slikt eksempel. Dette førte til at valget falt på A3UB kortet.



Figur 18 Xmega A3BU Xplained

Kortet ble først forsøkt plassert på undersiden av vognen, men i denne plasseringen oppsto det problemer med koblingen mellom Windows og kontrollerkortet. Kortet ble så plassert på oversiden av vognen, dette førte til langt færre avkoblinger men er ikke fult så estetisk pent. Det er også verdt å

nevne at enkoderkortet som er av samme modell ikke har noe problem med å være plassert under vognen. Det har også blitt forsøkt å bruke dette enkoderkortet som motorkontrollerkort på plassen hvor enkoderkortet ikke har problemer, men dette ga de samme problemene på motorkontrollerkortet som beskrevet over.

2.2 Programmering av kontrollerkort

Det er brukt to separate «Xmega A3BU explained» kort, et til motorkontrollerkort og et som kontrollerkort for enkoderene, da dette gir mulighet for at forskjellige programmer kan bruke de.

I denne delen av teksten er det en del åtte bits registere som settes, dette vises med 0bxxxxxxxx hvor x er verdien i hver bit som setes, eksempelvis 0b00101010 tilsvarer 0x3A (hexadesimal) eller 42 i titallsystemet (32 + 8 +2 = 42)

2.2.1 Motorkontrollerkortet

Dette eksempelet ble hentet fra ASF [1] ved hjelp av Atmel studio, programmet er nokså simpelt, det starter med å initsalisere systemklokker og andre detaljer. For å få aktivert I/O delen av kortet må løkken som skrur alle «peripheral clocks» av i «sysclk_init(void)» funksjonen fjernes. Dette fører til at I/O enhetene på Xmega mikrokontrolleren fungerer.

Programmet til motorkontrollen har disse detaljene:

Etter at alle initsialiseringene er utført setes pin 0 til 3 på port C til utganger ved å skrive 0b00001111, altså sette de fire laveste bitene i det 8 bits registeret til 1, til PORTC.DIR registeret.

Så settes bit 2 i TCC0.CTRLA registeret til 1, dette slår på teller funksjonen til timer 0 på port C med en skalerer på 8. Dette fører til at telleren teller en oppover for hver 8. klokkepuls på eksternklokken som styrer den.

I TCC0.CTRLB Settes bit 0, 1, 4 og 5 høye. Bit 4 og 5 slår på PWM utgangene for kanal A og B som går til pin 0 og 1 på port C. bit 0 og 1 sette høye for å aktivere PWM på teller 0 på port C.

TCC0.PER er verdien telleren teller til før den starter på 0 igjen, dette tallet settes til 2000.

TCC0.CCA og TCC0.CCB er den verdien, når telleren når den, som gjør at utgangen som brukes til PWM signalet endres fra høy til lav for kanal A og B. Disse verdiene settes til 0 for å holde «duty cycle» til 0%, siden TCC0.PER er satt til 2000 vil «duty cycle» da være TCC0.CCA/2000 for kanal A, og TCC0.CCB/2000 for kanal B

Hoveddelen av programmet er en while(true)-løkke som hele tiden sjekker hvilke ASCII tegn som blir sendt over COM porten til kortet, hvis det enten er «h» eller «v» vil det neste tegnet leses av. Dersom dette er «+» eller «-» vil pinnen som er koblet til retningsvelgeren på motorkontrollen settes henholdsvis til høy eller lav. While løkken ser videre etter enten et punktum «.» eller et tall mellom «0» og «9». Tallet kan ganges med 10 for å få «duty cycle» verdien som settes. Detekteres det et punktum vil de fire neste sifrene leses av og benyttes som verdier for TCC0.CCA eller TCC0.CCB, avhengig av om det var «h» eller «v» verdi som ble detektert. Det er viktig å merke seg at det leses 4

[1] Atmel solution framework http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx

tegn etter punktum, derfor må det skrives fire tegn før kortet går videre til å lytte etter neste «h» eller «v» verdier.

Første gang denne løkken mottar «h» eller «v» vil den aktivere en watchdog som aktiveres etter ca 1 sekund uten nye kommandoer. Når denne watchdogen aktiveres vil hele programmet restarte. For å unngå dette må det komme en ny «h» eller «v» kommando innen det har gått 1 sekund, da resettes timeren til watchdoggen.

Mens while løkken gjør dette svarer den tilbake de tegnene kommandoen inneholder tilbake på COM porten, dette er hovedsakelig for at programmet som sender kommandoer skal få en bekreftelse på at de er utført.

2.2.2 Enkoderkortet

Enkoderkortet benytter det samme eksempelet som Motorkontroller kortet, men istedenfor å starte en timer og slå på PWM utganger benytter dette kortet eksterne interrupts for å detektere at gråkoden fra enkoderene har endret seg. Det er derfor andre bits som settes i initsialieringen av dette kortet.

I PORTC.DIR settes alle bittene til null, dette definere dem som innput.

Sei() funksjonskallet aktiverer globale interruptus

PORTC.INT0MASK og PORTC.INT1MASK definerer hvilke pinner som aktiverer hvilke interrups, bit 0 svarer til pin 0 osv. Så når PORTC.INT0MASK = 0b00000011 definerer den pin 0 og 1 på port C som eksterne interupts for interrupt 0, mens PORTC.INT1MASK = 0b00001100 da gjør det samme, men for pin 2 og 3 gjeledene for interrupt 1

PMIC_CTRL slår på forskjellige nivåer av interrupts, PMIC_CTRL = 0b00000001 slår da på lav nivå interrupts.

PORTC_INTCTRL bestemmer hvilket nivå intrrupt 0 og interrupt 1 skal ha, ved å sette denne til 0b00000101 gis disse interruptene lavt prioriteringsnivå.

Hovedoppgaven til dette programmet er å svare hvor mange endringer i gråkoden fra enkoderene som er detektert. Når programmet får meldingen «l» inn svarer det med «H:[htall] V:[vtall]» hvor [htall] og [vtall] er antall endringer, dette tallet er en 16 bit signed int, så det kan gå fra -32769 til 32768, hvor negative tall brukes for å representere bevegelse bakover.

Etter at verdiene er sendt settes [htall] og [vtall] til null slik at programmet hele tiden får avstand målt siden sist avlesning.

2.3 Begrunnelse for valgt programmering

Det originale eksempelet leser kun av tegnet som kommer inn og svarer det tilbake. Det ble lagt til en watchdog i koden til motorkontrollerkortet for å forsikre at vognen stopper hvis noe uønsket skjer og datamaskinen mister kontakt med kortet. Watshdog funksjonen ble ikke lagt til i enkoder kortet da det på langt nær er like viktig for sikkerheten at dette kortet fungerer.

Bærefrekvens ble funnet ved måling. Målet var ca 1KHz da dette lager en merkbar lyd for mennesker, og det var anbefalt i dokumentasjonen om motorkontrollene. Det eksperimentelle ved det ligger i at selv om prosessoren er klokket til 36Mhz, er ikke det klokkehastigheten til periferal klokken. Men det ble funnet at en prescaler på 8 og TCC0.PER på 2000 ga noe over 1kHz, noe som var en akseptabel frekvens.

2.3.1 Enkoderkortet

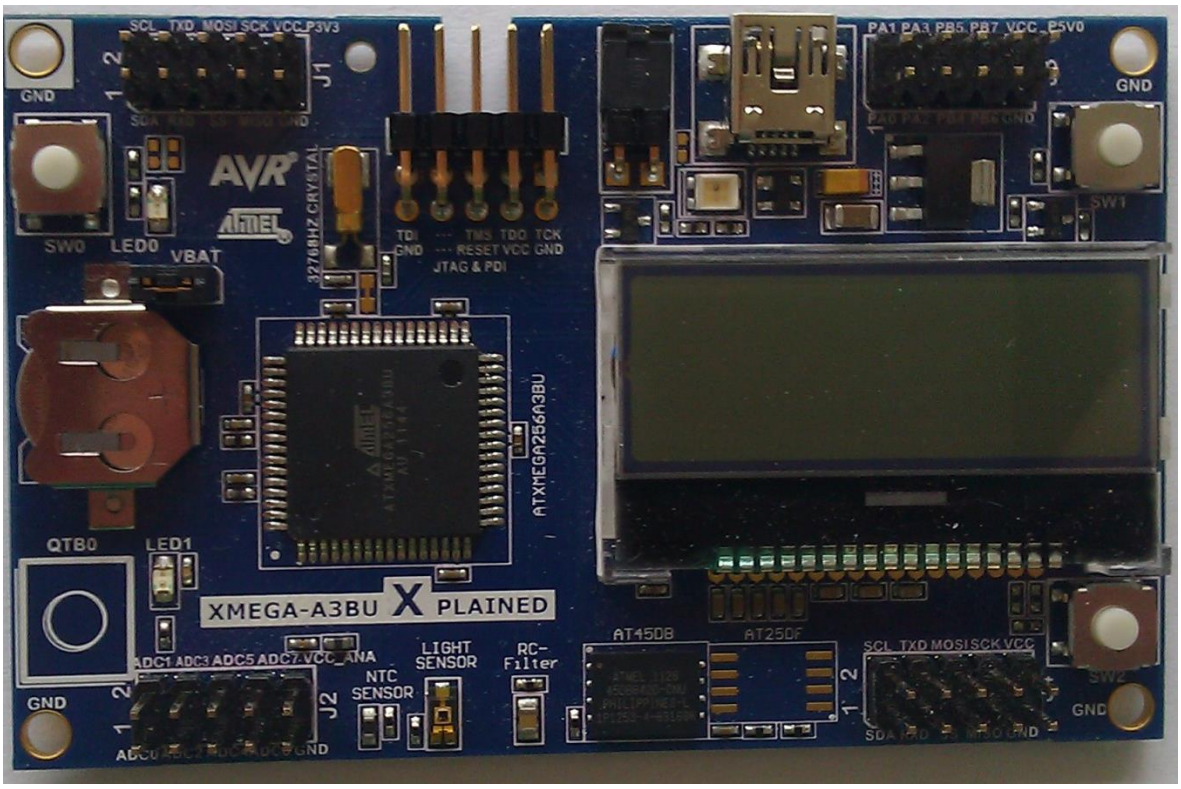
Kortet bruker interrupts for å detektere at gråkoden fra enkoderene endres. Dette fordi interrups er den sikreste metoden å detektere slikt på. Det hadde også vært mulig med polling (periodevis deteksjon av pinnen for sammenligning siden forrige deteksjon) av de fire pinnene som skal overvåkes, men dette ville krevd en pollefrekvens høy nok til å forsikre deteksjon. Interruptus derimot avbryter programmet som normalt kjører og detekter hvilken endring som har hendt før programmet går videre. Dette gir en mer elegant løsning enn polling.

Siden verdiene som blir svart er avstand siden sist avlesning er det viktig at programmet som leser av dette selv holder styr på tidsstempler hvis det er ønsket å kunne måle hastigheten eller lengde kjørt.

2.4 Kablingen til motorkontroll- og enkoderkortetne

Både motorkontrollkortet og enkoderkortet bruker port C for I/O, den er koblet til J1 headeren på Xplained kortet som befinner seg oppe til venste på bildet figur 19. Rent teknisk er pin 1 på headeren pin 0 på port C, pin 2 på headeren er pin 1 på port C osv. Pin 9 på headeren er jord og pin 10 er regulert 3.3 Volt. Av andre pinner i bruk er pin 10 på J3 oppe til høyre, dette er 5 volt.

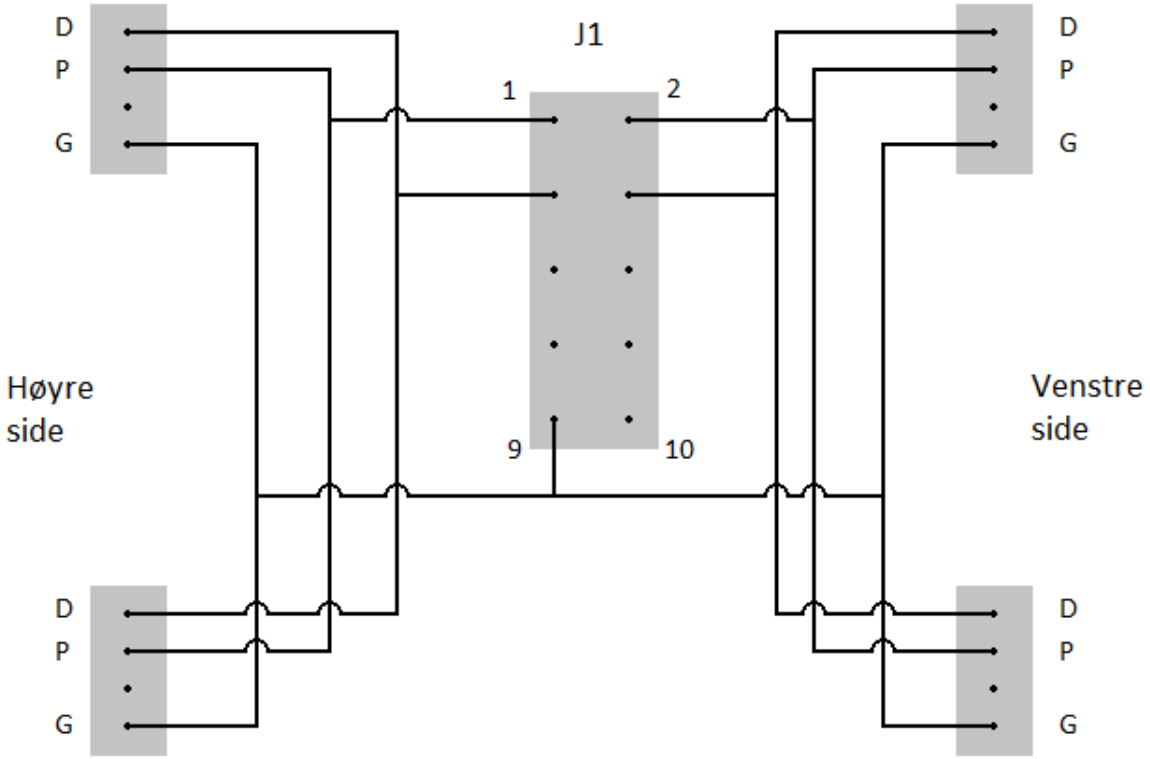
Plassering på kortet	Elektrisk kobling	Bruk Motorkontroll	Bruk enkoderkort
Pin 1 J1	Port C pin 0	PWM ut høyre side	Høy bit høyre enkoder
Pin 2 J1	Port C pin 1	PWM ut venstre side	Lav bit høyre enkoder
Pin 3 J1	Port C pin 2	Retningsturing ut H	Høy bit venstre enkoder
Pin 4 J1	Port C pin 3	Retningsturing ut V	Lav bit venstre enkoder
Pin 9 J1	Jord	Signaljord og 5 V jord	Signal jord
Pin 10 J1	3.3 Volt	Ikke i bruk	Spenning til avlesing
Pin 10 J3	5 Volt	Ikke i bruk	Spenning til enkoder



Figur 19 Nærbilde av Xmega-A3BU Xplained

2.4.1 Motorkontrollerkort

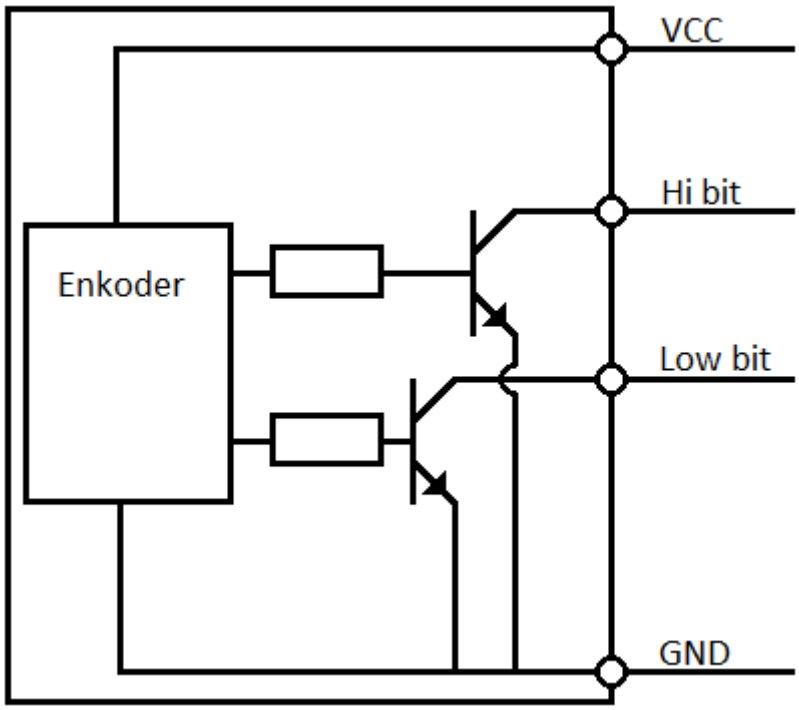
Det ble ikke sett på som nødvendig med beskyttelsesmotstander for utgangene som kobles på motorkontrollerkortet da hvert koblingspunkt trekker i størrelsesorden 1 mA med strøm, koblingsdiagrammet blir da meget enkelt:



Figur 20 Koblingsdiagram for motorkontrollerkortet

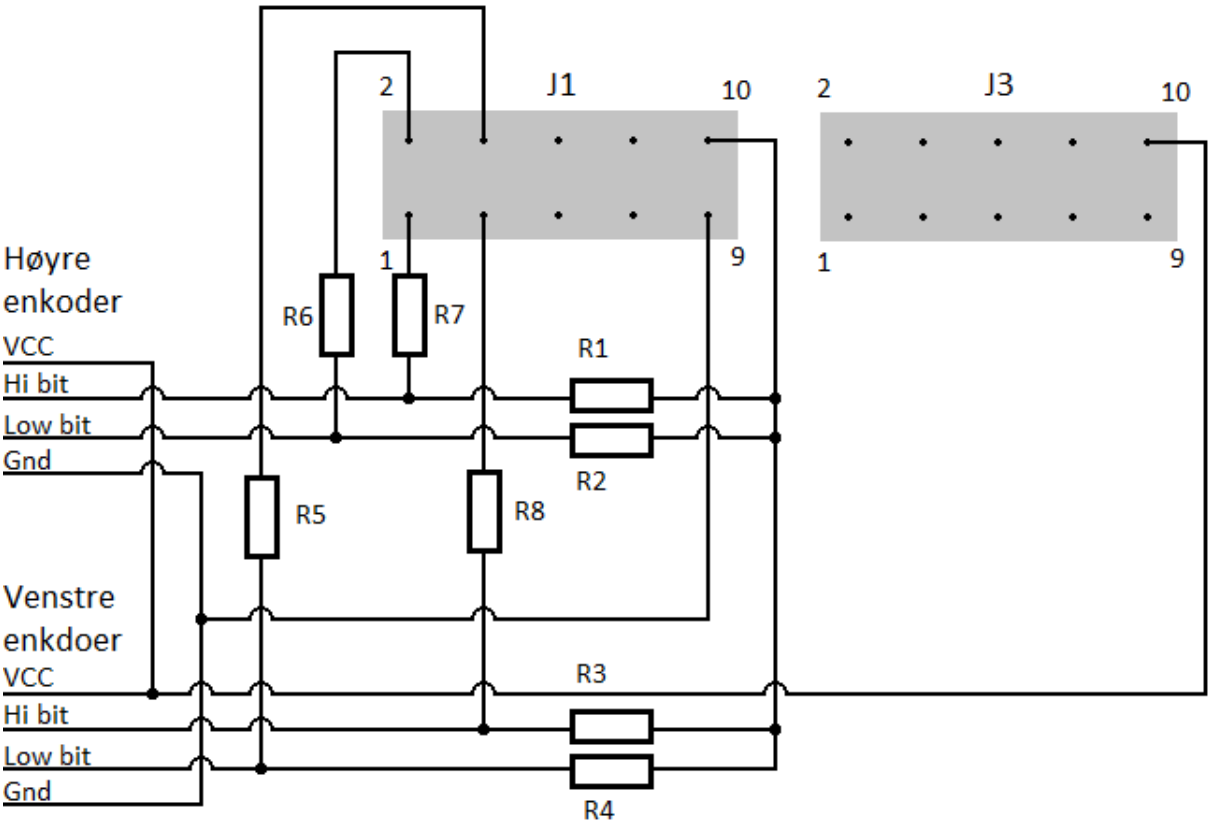
2.4.2 Enkoderkort

Enkoderene bruker noe som kalles «åpen kollektor» kolbing, dette betyr i praksis at det er en transistor som åpnes og lukkes som en bryter inne i selve enkoderen.



Figur 21 Kobling internt i enkdoeren

Dette fører til at det trengs en spenning og noen motstander for å detektere endringer i transistoren, det er også noen sikkerhetshensyn å ta da Atmel chippen ikke tåler mer enn 3.3 volt på inngangene, og enkoderen ikke tåler mer enn 30 mA gjennom transistoren.



Figur 22 Koblingsdiagram for enkoderkortet

R1 til 4 er 470Ω og R5 til 8 er 120Ω, dette fører til at det maks går 7 mA gjennom transistoren (3,3V/470Ω = 7mA). R5 til 8 er der som beskyttelsesmotstander for mikroprosessoren.

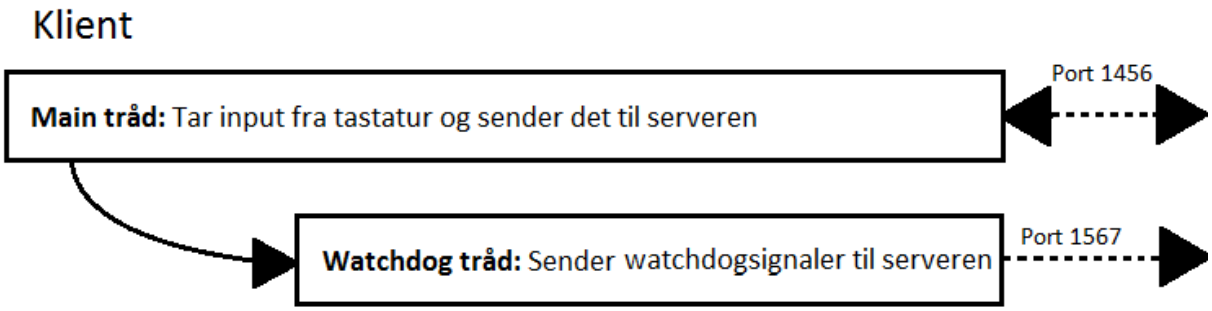
3 Programvaren som bruker kortene

Programvaren som styrer vognen består av en server som ligger på datamaskinen som står på vognen og en klient som ligger på maskinen brukeren benytter. Serveren mottar informasjonen fra klienten hvorpå den så sender meldinger videre til motorkontrollerkortet om hvilke pådrag som er ønskes. Det er også en hartbeat-tråd i klienten som hvert 100 millisekund sender en melding uavhengig av alt annet, hvis disse hartbeat meldingene ikke blir mottatt vil serveren sende stopp signal til motorkontrollerkortet. Dette gjøres for å sørge for at vognen stopper hvis operatøren mister kontakt med vognen, noe som vil føre til at vogen ikke lenger får hartbeat meldingene og kan utføre egnede tiltak.

3.1 Klienten

Det første klienten gjør er å hente inn IP adressen serveren ligger på, så startes en hartbeat tråd som separat fra alt annet sender et tall mellom 0 og 29 med 100 millisekunders mellomrom til serveren, tallet telles fra 0 opp til 29 før det starter på 0 igjen. Denne infromasjonen går på port 1567.

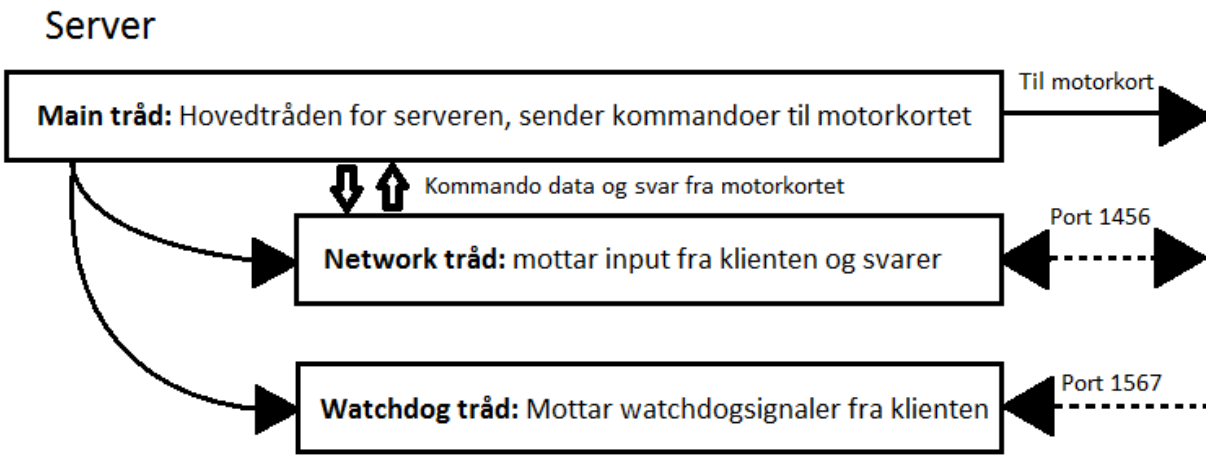
Hovedtråden fortsetter med å opprette en egen socket for input fra operatøren, porten til denne socketen er 1456. Hovedtråden begynner så å loope i en while(true) løkke hvor den registrer alle tastetrykkene som gjøres på tastaturet. Hvis det er A, S, D,W,X,R eller F sender den denne bokstaven til serveren. Den venter så på et svar fra serveren om strengen som motorkontrollerkortet svarer med før den skriver ut denne i konsollvinduet.



Figur 23 Klientens tråder og nettverkstrafikk

3.2 Serveren

Serveren starter med å hente inn hvilken COM port motorkontrolleren ligger på, enten via en config.txt fil eller ved en promt. Den lager så en port som kan brukes til å sende kommandoer til motorkontrollerkortet på. Så starter den en tråd for å motta kommandoer fra klienten, deretter en hartbeat mottaker tråd. Den går så inn i en løkke som leser av kommandoen fra klienten og sender korrekt input til motorkontrollerkortet.



Figur 24 Serverens tråder, nettverkstrafikk og interne datatrafikk

4 Valg av datamaskin

4.1 Utfordringer med strømmen

Den første datamaskinen som ble forsøkt brukt var en Dell Optiplex 9010, dette er en liten datamaskin på 31cm x 30cm x 9cm med en relativt kraftig Intel i5 3550 på 3.3 GHz og 8 GB med ram. Den ble forsøkt koblet til en modifisert sinus inverter, dette fungerte ikke. Det ble så testet med en eldre og noe større Dell Otiplex 990 på 41cm x 35cm x 10 cm, denne har en noe eldre Intel i5 2450P på 3.2 GHz og 8 GB ram. Denne hadde også problemer med å starte opp når den ble drevet fra en modifisert sinus inverter.

Det ble da kjøpt inn en ren sinus inverter for å prøve om dette ville ha positiv effekt. Begge datamaskinene ble da raskere til å starte opp, men 9010 maskinen hadde fortsatt problemer ved at den POSTet (power on self test) men lasting av Windows gikk ekstremt tregt; brukte ca 1 time på å komme til innloggingsskjermen. 990 maskinen klarte å komme inn i Windows uten store problemer, men enheter koblet til med usb mistet konstant tilkoblingen for så å komme tilbake, dette gjorde begge maskinene ubrukelige.

Det ble testet med en bærbar pc på inverterene, det viste seg at denne klarte fint å fungere og lade fra begge inverterene. Det ble deretter testet med en liten, selvbygd stasjonær datamaskin hvor strømforsyningen var av samme type som normalt brukes på bærbare pcer. Denne fungerte også godt på begge inverterene. Det ble da bestemt å bygge en datamaskin med samme kabinett og strømforsyning og teste om denne fungerte, det viste det seg at den gjorde og det er den som nå benyttes

4.2 Mulige årsaker til problemer

De to nevnte Dell maskinene hadde desidert størst problemer med den modifiserte sinus inverteren. Den selvbygde datamaskinen har samme type hardware som sitter i de to maskinene, det peker på at det er et problem i strømforsyningen til datamaskinene. PC strømforsyninger er i dag switshede, det betyr at det er en god del elektronikk som styrer selve strømforsyningen. Det er ikke utenkelig at strømforsyningene bruker frekvensen fra lysnettet til å reguelre seg selv på, når denne ikke har et klart definert nullpunkt kan dette skape problemer. En annen årsak kan være at firkantpulsen forvirrer elektronikken eller at den rett og slett ikke er bygd for slik. Sinus bølge inverteren burde uansett ha fungert, men da denne ikke gjorde dette kan dette peke på at det fortsatt er noe støy fra inverteren.

4.3 Valg av hardware til selvbygg

Det var nødvendig med en god del kraft i datamaskinen som satt på roboten. Dette da SLAM algoritmen som Mikael Berg benyttet seg av var meget tung på prosessor. Det var også nødvendig med et stort minne. Da de allerede eksisterende systemet for robotmanipulatoren benyttet Windows spesifikk kode og det viste seg at kun laptop-strømforsyninger fungerte tilfredsstillende på inverteren var det to klare valg, enten kjøpe inn en kraftig bærbar pc eller bygge en som brukte en laptop-strømforsyning. Problemet med å velge en laptop var at disse er mye dyrere enn en tilvarende stasjonær datamaskin. Da det var ønskelig å holde prisen nede ble det bestemt å forsøke å bygge en liten datamaskin med prosessorkraft tilvisende de allerede prøvde datamaskinene og tilsvarende mengde med ram.

Det første som ble valgt ut var kabinettet, da dette måtte ha en noe spessiel løsning for strømforsyning. Det som ble brukt i den selvbygde datamaskinen som fungerte var et hjemme kino pc kabinet, helt spesifikt er det et Antec ISK 110 VESA. Dette kabinettet tar hovedkort av typen ITX og benytter en 19 V, 90 W laptop- strømforsyning som så gjøres om til de normale ATX spenningene i en sekundær spennginsforsyning inne i kabinettet. Dette kabinettet har også et VESA feste for å kunne festet bli bak flatskjermer, denne ble brukt for å feste maskinen på en plate slik at det ble bedre plass på selve roboten selv om målen for selve kabinettet kun var 21cm x 22cm x 8cm.

Prosessoren ble valgt til en intel i5 3570 på 3.4 GHz, det ble også kjøpt inn 8 GB med minne til denne. Hovedkortet er av typen Zotac H61-ITX-A-E WIFI. Dette hovedkortet har et trådløst netverkskort slik at datamaskinen kan koble seg til andre trådløse nettverk enn det nettverket ruteren som er med roboten er koblet til. Det ble også kjøpt inn en SSD av typen Kingston SSDnow V300 60 GB. Det ble valgt en SSD da disse er mer resistente mot vibrasjoner enn det en vanlig harddisk er. Da det ikke var behov for mye lagringsplass, Windows, noen drivere og noen logger, var det mulig å benytte en liten SSD noe som ikke drev opp prisen nevneverdig.

Alt i alt ser spesifikasjonene slik ut:

CPU:	Intel i5 3570 3,4GHz
HK:	Zotac H61-ITX-A-E
RAM:	Corsair 8GB Value 1600MHz
SSD:	Kingston SSDnow V300 60GB
Chassis:	Antec ISK 110 VESA
PSU:	innebygd i chassis

5 Batteri og inverter

Da det ikke var ønskelig med en lang ledning som kan komme i veien og begrense robotens operasjonsområde var det klart at systemet trengte en egen energikilde. Det var allerede kjøpt inn et 24V LiFoFe batteri før jul som skulle drive motrorene, men da det var ønskelig å ha en lang opperasjontid mellom ladinger av batteri ble det klart at det kunne være lurt å ha alt annet gående på et separat batteri. Det enkleste ville da være å bruke et 12V blybatteri ikke ulikt det i biler. Det var dog en liten utfordring, kontrollenheten for robotarmen har en integrert strømforsyning, og i og med at det ikke er ønskelig å plukke denne fra hverandre var det nødvendig å generere 230 V på vogna.

Dette kan gjøres med en inverter, det finnes dog to typer som ble veid opp mot hverandre; modifisert- og ren sinus-inverter

5.1 Hvorfor velge modifisert sinusinverter

En modifisert sinus inverter har kun en fordel, og det er pris, den er merkbart billigere enn inverterene med ren sinus

Bakdelen er at disse lager mer støy i de elektriske kretsene, noe som kan føre til at følsom elektronikk som benytter seg av krysningen mellom positiv og negativ halvperiode for tidskontroll kan slutte å fungere. I tillegg bruker laster som ikke er rent ressistive mer effekt ved firkantbølgen som modifisert sinusinverter genererer enn ved rene sinusformer, hvor stor denne effekten er avhenger veldig av cosϕ til lasten.

5.2 Hvorfor prøve med ren sinusinverter

Inverterene med ren sinus har som navnet tilsier en ren sinus som utgang, dette løser problemene som firkantbølgen til modifisert sinusinverter har (elektrisk støy og potensielt høyere effekttrekk), Ulempen er at kvaliteten koster penger, ofte fem ganger mer enn modifisert sinus da det trengs store og/eller kompliserte kretser for å skape sinusformen, eventuelt en dynamo som drives av en likestrømsmotor.

5.3 Løsningen slik den fungerer nå

Modifisert sinusinverteren (stor, svart) driver alt som går på 230 volt bortsett fra kontrollboksen for robotarmen, den drive av ren sinusinverteren (liten, sølv)som gir en ren sinus.

6 Programmer til kontrollerkortene

Det er relativt enkelt å bruke klienten og serveren. Serveren må startes først slik at klienten har noe og koble seg til. Serveren trenger å vite hvilken COM port motorkontrollerkortet er koblet til, enten ved hjelp av config.txt filen eller promten som kommer når programmet starter. På samme måte må klienten ha IP adressen som serveren befinner seg på.

6.1 Bruke vedlagt program

Både serveren og klienten er skrevet i Visual studio 2012 med C# og .NET framework 4.5, det er derfor nødvendig å installere dette på maskinene som skal bruke server og klient programmet, installasjonsfiler for .NET finnes på vedlagt DVD.
For å starte serveren må datamaskinen som står på roboten konfigureres til å motta Remote desktop brukere, så må Remote desktop benyttes til å starte serveren.

En funksjon ved klienten er at den IKKE detekterer tastaturinput hvis konsollvinduet som kjører programmet ikke har fokus, dette er gjort da det generelt sett ikke er lurt å lage et program som konsekvent samler inn tastetrykk og sender dem videre over nettverket, Det ville da være en såkalt «key logger», noe som antivirus programmer er designet til å detektere og fjerne da det er en potensielt stor sikkerhetstrussel. Det er derfor viktig at fokus hele tiden ligger på programmet og ikke andre vinduer i Windows.

Klienten er relativt simpel, den starter med å lete etter en fil ved navn config.txt i samme mappe som klienten startes fra. Om denne filen finnes vil klienten anta at tekststrengen som ligger i denne filen er IP adressen som serveren befinner seg på, hvis denne strengen ikke er en IP adresse vil ikke programmet fungere.

Hvis klienten ikke finner config.txt vil den promte brukeren om en IP adresse.

Windows trenger drivere for at både motorkontrollerkortet og enkoderkortet skal fungere, disse befinner seg i den samme mappen som kildekoden til firmwaren til kortene ligger. Denne driveren genereres av ASF eksempelet som denne firmwaren er basert på, og fungerer derfor for begge kortene.

6.2 Hvordan lage egne x86 programmer

Det som trengs for å sende meldinger til både motorkontrollerkortet og enkoderkortet er at programmet som skal benytte disse kobler seg til COM porten de danner. Under Windows trengs det drivere til dette, mens Linux har innebygde drivere. Det er også vært å nevne at Windows 8 IKKE har drivere for kortene.

Com porten som brukes må settes opp på følgende vis:

Baud rate:	115200
Parity:	ingen
Data bits:	8
Stop bit:	en
Handshake:	ingen

For å gjøre dette i C# .NET framework kan følgende funksjon benyttetes:

```
SerialPort porten = new System.IO.Ports.SerialPort([port], 115200, Parity.None, 8,
StopBits.One);
porten.Handshake = Handshake.None;
```

Her lages en seriell port ved navn «porten», den peker til den COM porten ved som skrives i feltet [port], det er en string som f.eks. kan være COM4.

Forøvrig er klassen beskrevet godt her (10)
Vedlagt DVD inneholder også kildekoden til både server og klientprogrammet som en referanse eller startpunkt for videre utvikling, disse ligger under Programvare/Vognkontroll

6.3 Endring på firmwaren på motorkontrollerkortet og enkoderkortet

Den enkleste metoden for utvikling til Atmel proessorer er Atmel studio. I denne oppgaven har det blitt brukt versjon 5.0. Xmega-A3BU Xplained kortene har støtte for en bootloader som gjør det mulig å programmere kortene med Atmel FLIP, men den beste måten er ved bruk av en JTAG eller ISP programerer som støttes i Atmel Studio. Vedlagt DVD inneholder bootloader i Programvare\Firmware\Bootloader, Ellers ligger kildekoden til motorkontrollerkortet under Programvare\Firmware\Motor controll med timer og for enkoder kortet i Programvare\Firmware\ENCDER_counter. Hvis det ønskes å benytte FLIP anbefales det å laste ned nyeste versjon fra Atmel sine hjemmesider.

7 Utfordringer og kjente feil

7.1 Forsinkelse i svar fra serveren

Det er en forsinkelse i hvilke svar som kommer fra serveren tilbake til klienten på en melding, så hvis det sendes W R R, noe som ville få systemet til å skrive ut
h+0 v+0
h+1 v+1
h+2 v+2
kommer det i stedet

h+0 v+0
h+1 v+1
Dette har ingen praktisk betydning bortsett fra at utskriften fra klienten ikke er fullstendig up to date. Sendes en ny W kommando vil utskriften bil:

h+0 v+0
h+1 v+1
h+2 v+2

7.2 Motorkontrollerkortet mister kontakten

Det kan forekomme, dog sjeldent, at motorkontrollerkortet slår seg «vrangt» slik at ingen ting kan koble seg på det. Da må kortet restartes. Dette gjøres ved å trekke ut USB pluggen og sette den inn igjen, settes denne i en annen USB port vil COM porten endres i Windows.

7.3 Config.txt

Hvis det ikke er en COM port som finnes i config.txt til server programvaren vil hele programmet få en segmentation feil og krasje. Det er derfor viktig at enten config.txt er riktig eller at promten som kommer hvis config.txt ikke finnes i mappen til programmet brukes.

Det hender at server og/eller klient programmet har problemer med å lese av teksten som står i config.txt, da er det enkleste å slette denne config.txt og lage en ny txt fil med den samme strengen i.

7.4 Pan-tilt enheten

Pan-tilt enheten klarer ikke lenger å home til en fornuftig posisjon eller den bare går fremm og tilbake eller prøver å dytte seg forbi stopp-punktene sine. Dette skjer hver gang den har vært forsøkt brukt dette semesteret.

7.5 Manual move programmet

Manual move programmet klarer ikke lenger å plukke opp dataene fra IMUen feste til VR brillene, klientprogrammet krasjer hvis tråden som prøver å plukke opp dette signalet startes, derfor er denne tråden kommentert bort i programmet som ligger ved denne besvarelsen. Denne linjen ligger i filen ManualMoveDlg.cpp og ser slik ut:

```
CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE) startPanTiltClient, NULL, 0, NULL);
```

Manual move programmet på både serveren og klienten kaster to feilmeldinger når det starter, disse to må ignoreres (trykk på «ignore» knappen i feilmeldingen) før programmet starter.

8 Diskusjon og konklusjon

Systemet fungerer i sin nåværende konfigurasjon, det er ikke perfekt med noen små utfordringer med motorkontrollerkortet og problemene med IMUen og pan- tilt enheten. Men dette er til liten hinder for bruk av robotmanipulatoren sammen med vognen den nå står montert på.

Det nye programmet som styrer vognen er separat fra programmet som styrer robotmanipulatoren. Det samme gjelder hardwaren som styrer motorene på vognen. Dette gjør at det er lett for programmer skrevet av en 3. part å kontrollere vognen og det gjør også motorkontrollen mindre mottagelig for bugs og feil i «manual move» programmet som styrer robotmanipulatoren.

Alle målene oppgaven satte ansees derfor som oppnådd, robotmanipulatoren er nå en del av et fjernstyrt bevegelig robotsystem som kan fungere som en base for videre utvikling.

Forslag til videre arbeid

Da det allerede finnes kollisjonsunngåelse systemer på selve armen ville det være mulig å utvikle en løsning som får armen til en ønsket posisjon autonomt. Dette vil kunne brukes sammen med autonom vogn for større grad av selvstendighet fra mennesker.

Systemet for 3D visning fra kameraene begynner å bli gammelt, det benytter innganger som vil bli vanskeligere å finne på datamaskiner i fremtiden. Det kommer også generelle standarder for stereoskopiske 3D skjermer med directX 11.1 (11) som er planlagt introdusert i Windows 8. Det er også mulig med «head tracking» med en Microsoft Kinect (12).

Programvaren som styrer roboten kunne trenge en forbedring, det er blant annet to feilmeldinger som kommer hver gang programmet starter, det er trolig noe med nettverkssystemet og dette føles ikke som et polert produkt.

For øyeblikket kan bare én handling fra joystick'en gis om gangen, funksjonaliteten i dette systemet burde forbedres slik at flere akser kan styres samtidig.

Det kunne også være en idé å lage en simulator av roboten, for å kunne øve bruken uten fare for å ødelegge verken robot eller omgivelser.

Det er meget dårlig plass på selve vognen, Det hadde vært lurt og enten finne en måte å forminske mengden kabler eller lage en hylleløsning bakpå vognen for bedret organisering.

Vedlegg A: oppkobling av robotmanipulator

Brukerveiledning for å få maunal move programmet til å styre robotarmen, hentet fra (1)

A.1 Nødvendig utstyr

For og koble opp systemet trengs følgende utstyr:

1) To (2) datamaskiner med kabler (en for klient og en for server siden)
2) To (2) rutere (brukt TP-Link TL-WR841N)
3) En (1) robot (Intelitek scorebot er 4u) og kontrollboks med:
   a) To (2) påmonterte kameraer (TP-link TL-SC3430)
   b) Avstandsfinner
   c) Pan- tilt enhet (directed preseption pan tilt unit modell: ptu-46-17.5)
4) En (1) Nvis VR-briller med Inerticube2 og kontrollboks
5) En (1) Kontrollboks for pan tilt unit (pan tilt controler modell: PTU-46)
6) En (1) Joystick av typen Logitech force 3D PRO

I tillegg trengs det følgende kabler og strømforsyninger:

1) Fire (4) TP kabler, 2 meter eller mer, cat. 5 eller bedre
2) To (2) apparat strømkabler med jording
3) To (2) 12 V 1 A adaptere, en til hvert kamera
4) To (2) 9 V 0,6 A adaptere, en til hver ruter
5) En (1) RS232 kabel til pan tilt kontroller
6) En (1) 9 – 30 V, 2,5 A adapter til pan tilt kontroller
7) En (1) 24 V 0,75 A adapter til Logitech force 3D PRO
8) En (1) InterSense USB to RS232 adapter til InertiaCube 2 (modell 100-SC210-USB0) som er festet på VR-brillene
7) To (2) USB A til B kabler (se fig. 26 neste side)
8) To (2) analoge DVI kabler (DVI-A)

DVI-A kabler kan lages ved å bruke en D-sub kabel med D-sub > DVI overganger i hver ende, se fig. 25

Figur 25 D-sub kabel med D-sub > DVI overgang

Figur 26 USB A til B kabel

A.2 Konfigurering og oppkobling

Alle programmene og driveriene skal befinne seg på DVDen som ligger bakerst i dette heftet

A.2.1 Installering av programvare

Følgende programvare trengs installert på klienten:

1) Installer DirectX 9 fra mappen DXSDKINSTALL
2) Installer programvare for orienteringssensoren fra mappen Device Tool2
3) Installer VLC 2.0.3 fra mappen VLC

Følgende programmer trengs på serveren

1) Installer DirectX 9 fra mappen DXSDKINSTALL

2) Installer drivere og programvare for robotarmen, disse finnes under install i mappen Programvare_intelitek-robot

A.2.2 Oppkobling

Klienten er sentrert rundt den ene datamaskinen, til denne kobles følgende enheter til:

1) Logitech force 3D PRO
2) Nvis VR-briller med Inerticube2 og kontrollboks
3) En (1) ruter (brukt TP-Link TL-WR841N)

Til dette trengs følgende kabler og adaptere

1) En (1) 24 V 0,75 A adapter til Logitech force 3D PRO
2) To (2) analoge DVI kabler (DVI-A)
3) En (1) apparatstrømkabel med jording
4) En (1) InterSense USB to RS232 adapter til InertiaCube 2
5) En (1) TP kabler, 2 meter eller mer, cat. 5 eller bedre
6) En (1) 9 V 0,6 A adapter

24V adapteren kobles til Logitech force 3D PRO joysticken, sett så i USB kabelen fra joysticken i datamaskinen, windows finner driverne selv hvis maskinen er koblet til internett.

Koble de to analoge DVI kablene fra datamaskinen til kontrollboksen for VR-brillene, koble så til strømkabelen (alle disse kontaktpunktene er bak boksen).

Kolbe til InterSense USB to RS232 adapteren i USB porten og sett telefonkontakten fra VR brillene i den andre enden av kabelen (den lille boksen).

Koble klient datamaskinen til en av de 4 ruter portene på ruteren, ikke koble den til WAN porten. Sett i 9 V Adapteren.

Serveren er sentrert rundt den andre datamaskinen og roboten, derfor trengs følgende enheter:

1) En (1) ruter (brukt TP-Link TL-WR841N)
2) En (1) Kontrollboks for pan tilt unit (pan tilt Controller modell: PTU-46)

I tillegg må det sees til at følgende enheter sitter på roboten:

1) To (2) påmonterte kameraer (TP-link TL-SC3430)
2) Avstandsfinner
3) Pan- tilt enhet (directed preseption pan tilt unit modell: ptu-46-17.5)

Til dette trengs følgende kabler og adaptere:

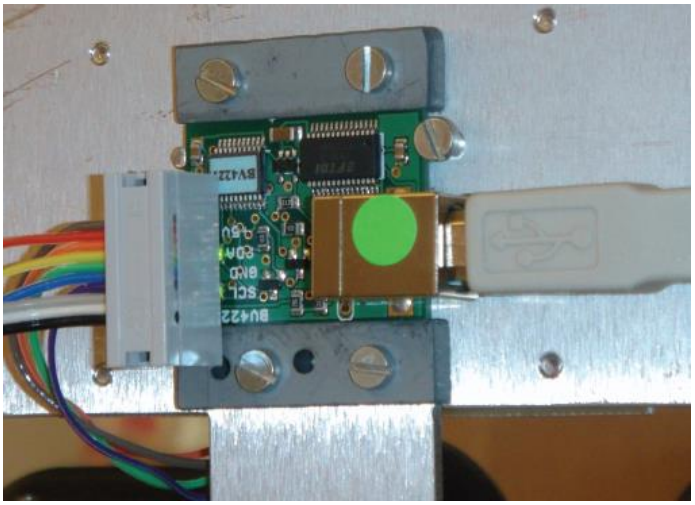
1) Tre (3) TP kabler, 2 meter eller mer, cat. 5 eller bedre
2) En (1) apparatstrømkabel med jording
3) En (1) RS232 kabel
4) To (2) 12 V 1 A adaptere

5) To (2) USB A til B kabler
6) En (1) 9 – 30 V, 2,5 A adapter
7) En (1) 9 V 0,6 A adapter

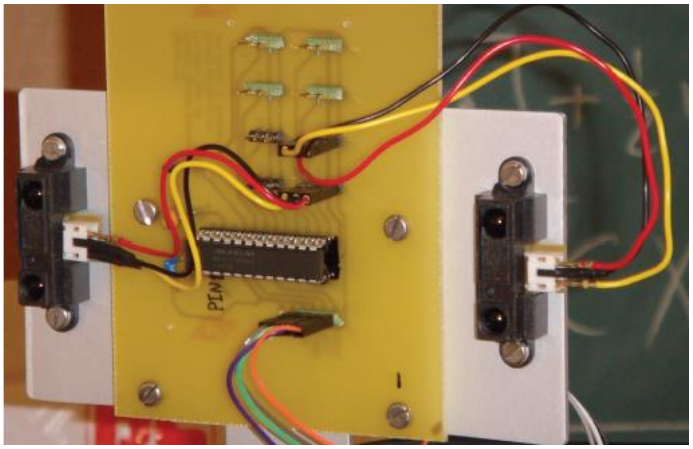
Koble robotarmen til kontrollboksen med kabelen som kommer fra robotarmen, kolbe så USB kabelen fra kontrollboksen til server datamaskinen, koble så til apparatstrømkabel til kontrollboksen.

Koble 9-30 V adapteren til pan tilt kontrolleren, koble så RS232 kabelen til den samme kontrolleren og så i datamaskinen. Den er programmert til å bruke COM1, men dette kan endres i scontroller.h i kildekoden til serveren.

Koble den andre USB kabelen fra datamaskinen til avstandsmålerne på roboten, driverne ligger under mappen Driver_USB_I2C_Termial hvis de trengs. Her er det ingen settup-fil. Produsenten oppgir at drivere og dokumentasjon også er tilgjengelig på www.byvac.co.uk, pic32.byvac.com eller www.pin1.org.



Figur 27 USB til I2C modul bilde fra (2)



Figur 28 Kabeloppsett for avstandsmåler bilde fra (2)

Koble så kablene til I2C modulen som på bildet over, og kolbe avstandsmålerne slik figur 28 viser.

Koble Datamaskinen til en av de 4 ruter portene på ruteren, ikke koble den til WAN porten. Sett i 9 V Adapteren.

Sett koble så de to nettverkskameraene til den samme ruteren med de de to gjenværende TP kablene, koble så til 12 V 1 A adapteren til de to kameraene.

A.2.3 Nettverk og VLC

Det første som burde konfigureres er nettverket, om det allerede er konfigurert hopp over denne delen. Først konfigurerer ruteren på klient siden:

1. Åpne en nettleser
2. Skriv inn 192.168.0.1 i adressefeltet (dette er standardadressen til ruteren)
3. Brukernavn er «aspunvik» og passord 1q2w3e (standard brukernavn og passord er «admin» og «admin»)
    a. For og bytte brukernavn og passord gå til «System Tools» og velg «Password»
4. Velg «Wireless Settings» under «Wierless» i menyen til høyre
    a. Endre «Wireless Network Name» til RV1
    b. Set «mode» til «11n only»
    c. Sørg for at «Enable Wireless Router Radio» og «Enable SSID Broadcast» er huket av
    d. Lagre
5. Velg «Wireless Security» under «Wierless»
    a. Huk av for «WPA/WPA2 personal»
    b. Velg «WPA2-PSK» under «Version»
    c. «PSK Password» settes til et ønskelig passord, «1q2w3e4r5t6y7u8i9o0p» fungerer godt
    d. Lagre og restart ruteren

Ruteren er nå konfigurert med et sikret trådløst nettverk, bytt så til ruteren som står på server siden:

1. Åpne en nettleser
2. Ruteren er satt default til 192.168.0.1 fra fabrikken (hvis den allerede er konfigurert er det 192.168.0.200)
6. Brukernavn er «aspunvik» og passord «1q2w3e» (standard brukernavn og passord er «admin» og «admin»)
    a. Dette kan endres under «System Tools» – «Password»
3. Velg «Wireless Settings» under «Wierless» i menyen til høyre
    a. Endre «Wireless Network Name» til RV2
    b. Sett «mode» til «11n only»
    c. Sørg for at «Enable Wireless Router Radio» og «Enable SSID Broadcast» øer huket av
    d. Huk av «Enable WDS Bridging»
        i. «SSID(to be bridged)» settes til «RV1»
        ii. Trykk på «Survey» under «BSSID(to be bridged)»
        iii. Finn RV1 på listen som kommer opp og trykk på «connect»
        iv. «Key type» settes til «WPA-PSK/WPA2-PSK»
        v. «password» settes til passordet på RV1 (hvis den er konfigurert som over er det «1q2w3e4r5t6y7u8i9o0p»)
        vi. Lagre og restart ruteren
4. Velg «Wireless Security» under «Wierless» admin
    a. Huk av for «WPA/WPA2 personal»

    b. Velg «WPA2-PSK» under «Version»
    c. «PSK Password» settes til et ønskelig passord, «1q2w3e4r5t6y7u8i9o0p» fungerer godt
    d. Lagre og restart ruteren
5. Under «Network» gå til «LAN» og sett «IP Adress» til «192.168.0.200»
6. Gå til «HDCP»
    a. Under «HDCP» settings velges «HDCP server» til «Disable»
    b. Lagre og restart ruteren

Ruteren til serveren er nå konfigurert korrekt, restart maskinen som skal bli server og start nettverkskameraene.

På ruteren som står på klient siden, RV1 på adresse 192.168.0.1 gjøres så følgende:

1. «HDCP client list» under «HDCP»
2. Sjekk at det er 4 enheter der, det to datamaskinene som skal være server og klient, samt to nettverkskameraer som identifiserer seg som «TL-SC3430»
3. Gå så til «Address Reservation»
    a. Trykk «Add New...»
        i. Skriv in MAC adressen til den maskinen som skal være server i feltet «MAC Address»
        ii. I Feltet «Reserved IP Address» skrives «192.168.0.103» med mindre en annen adresse er kompilert inn i koden
        iii. Lagre
    b. Gjør det samme for den andre datamaskinen og de to nettverkskameraene, men med andre adresser (forslag «192.168.0.100» og «192.168.0.102» til kameraene og «192.168.0..101» til klientmaskinen)
4. Restart ruteren, gjøres under «System Tools» > «Reebot»

Nå er nettverket satt opp og faste IP adresser er tildelt slik at alle enhetene er på samme plass hver gang, det er også kryptert så MAC filtrering er ikke nødvendig.

Konfigurering av nettverkskameraene:

1. Åpne en av adressene til nettverkskameraet (192.168.0.100 eller 192.168.0.102)
2. Brukernavn og passord er «admin» og «admin»
3. Trykk på «settings»
    a. Trykk på «basic»
    b. «camera»
    c. «General»
        i. På «RTSP» velg en port over 1000, «1024» og «1026» kan brukes
        ii. Trykk ok
    d. «MJPEG»
        i. Sett «frame rate» til «15»
        ii. Sett «Image Size» til «1280 x 1024»
        iii. «Quality» settes til «standard»
        iv. Sjekk at «Viewer authentication» er satt til «off»

v. Trykk ok

4. Gjør dette med det andre kameraet, husk å bruke annen port enn brukt på det første

Kameraene er nå konfigurert.

VLC konfigureres ved å gå til verktøy > innstillinger > input & codecs

Under nettverk her velger du «Low latency» for «Default caching policy» og «RTP over RTSP» for «Live555 stream transport».
Lagre.

Bildet vil nå være opp ned da kameraene er montert opp ned, dette ordnes slik:

1. Trykk på «Effekter og filtre» under «verktøy»
2. Velg fanen «videoeffekter»
3. Under denne velg «geometri»
4. Her hukes det av for «transform»
5. Og velges «Roter 180 grader»
6. Lukk

VLC vil nå rotere alle videoer 180 grader

A.3 Oppstart

For å starte hele systemet må enhetene slås på i en bestemt rekkefølge;

1. Start ruterene
2. Slå på datamaskinene og nettverkskameraene
3. Slå på kontrollenheten for pan tilt og kontrollenheten for roboten

For å sjekke at alle enhetene er kommet på nettverket kan det være lurt å logge på ruteren på klien siden, gå til “HDCP Client List” under HDCP og sjekke at alle de fire enhetene (to datamaskiner og to nettverkskameraer) er der.

Oppstart av Video delen; koble en iterasjon av VLC på hvert kamera, og oppstart av kontrollsystemet til roboten har ingen spesifikk rekkefølge.

A.3.1 Oppstart av video Feed

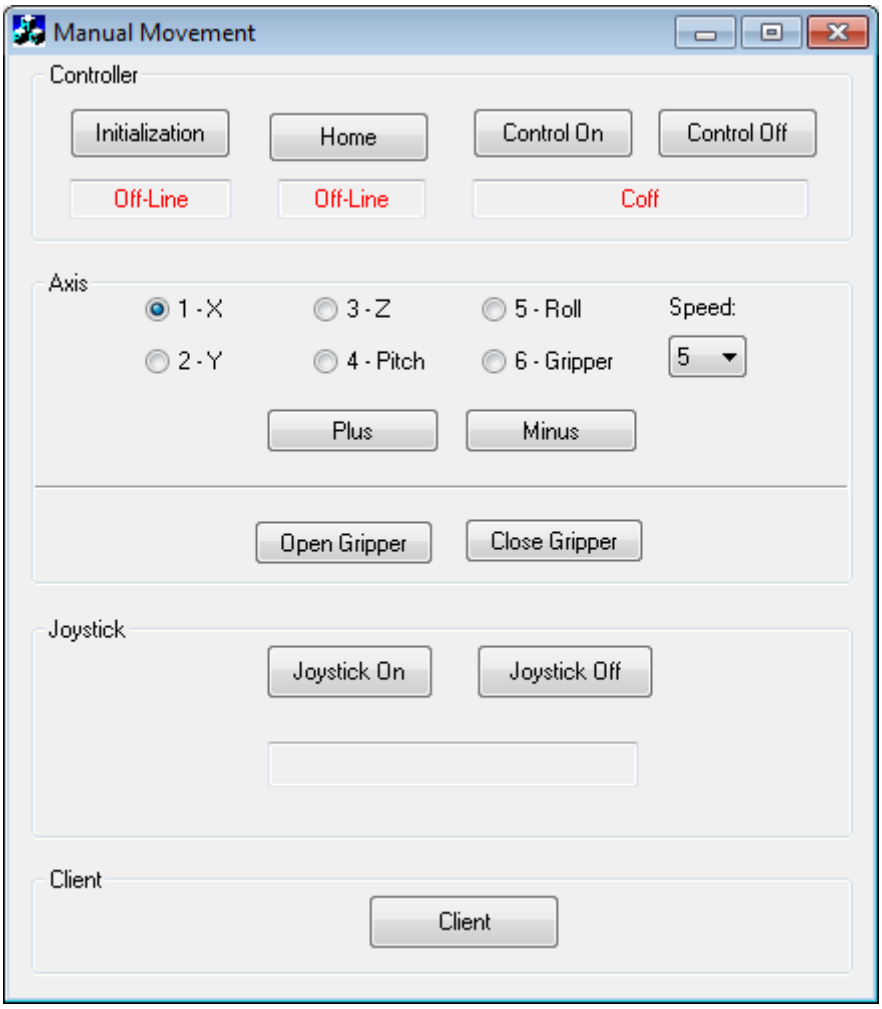
1. Start to iterasjoner av VLC
2. Under «Media» trykk på «Åpne nettverkskringkasting»
    a. Under «Please enter a network URL:» fyll in følgende: «rtsp://[ip til nettverkskamera]:[portnummer til samme kamera]/video.mjpg»
    b. Huk av for «vis flere valg»
    c. Under «mellomlagring» skriv «66 ms» istedenfor 666 ms
    d. Velg «spill av»
3. Videoen skal nå komme opp i vinduet til VLC, dette må gjøres for hver av VLC vinduene
4. Dra så hvert vindu over til sine respektive skjermer inne i VR-brillene
5. Dobbeltklikk på skjermbildet for fullskjerm

A.3.2 Oppstart av kontrollprogramvare

På serveren startes programvaren ved å klikke på filen "StartServer.bat", den befinner seg i mappen Programmer og kildekode
Det kommer to feilmeldinger, trykk «ignore» på begge to.

På klienten må først programmet IServer kjøre, det er programmet som henter informasjon fra inerta cuben på VR brillene.

Start klienten på klientmaskinen ved å trykke «StartClient.bat», den befinner seg i mappen Programmer og kildekode
Det kommer to feilmeldinger, trykk «ignore» på begge to.



Figur 29 Klientens gui

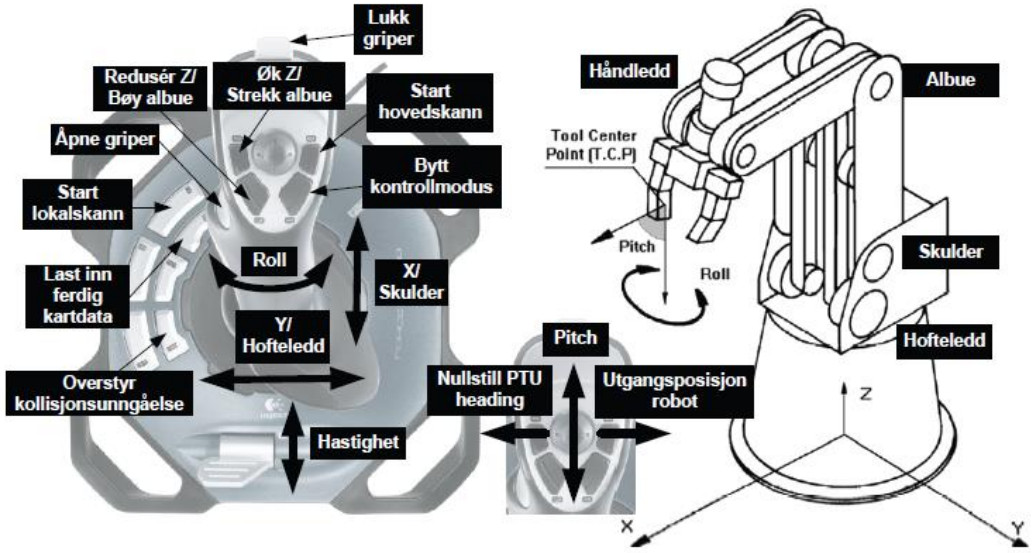
Trykk så på «Initialization»
Den øverste leden på kontrollboksen skal nå lyse grønt

Trykk deretter på «Home»
Roboten vil nå begynne å bevege seg, den «homer», altså finner startpunktet for bevegelse. Det er viktig å vente til den er ferdig med å bevege seg.

Tykk så på «Control On», så på «Joystick On»
Roboten skal nå fungere og la seg styre av joysticken

A.4 Bruk

Systemet skal nå være klar til bruk, kontrollene til joysticken er som følgende:



Figur 30 Joystick kontroller, bildehentet fra (2)

Vær oppmerksom på at joysticken kun kan sende en melding om gangen over nettverket, og at den må tilbake til nullpunktet før neste kommando kan sendes.

For å koble av, avslutt programmene på begge maskinene og slå av eksternt utstyr.

Vedlegg B: innhold på DVD

På DVD platen som er lagt ved denne oppgaven ligger det følgende mapper:
Programvare: programvare til å styre vogenen, og kildekode for server og klient applikasjonene
Dokumenter og annet: Datablad og annen dokumentasjon som har blitt benyttet under oppgaven
DVD_prosjekt: DVDplaten legt ved prosjektet som denne oppgaven baserer seg på.

Bibliografi

1. **Aspunvik, Petter.** *Robotisert vedlikehold Frobredelse til mobil enhet.* Tronheim : Institutt for teknisk kybernetikk, 2012.

2. **Bekken, Kristian Saxrud.** *Bevegelsesstyring av robbotarm og kamera med killisjonsunngåelse.* Tronheim : NTNU, 2010.

3. **Wagner, Jim.** Filtering PWM signals. [Internett] Oktober 2009. [Sitert: 9 April 2013.] http://www.proaxis.com/~wagnerj/PWMfil/PWM%20Filters.pdf.

4. **Brown, Jim.** Brief H-Bridge Theory of Operation. [Internett] April 1998. [Sitert: 15 April 2013.] http://www.dprg.org/tutorials/1998-04a/.

5. **Hahn, James H.** Modified Sine-Wave Inverter Enhanced. [Internett] 1 Aug 2006. [Sitert: 28 April 2013.] http://powerelectronics.com/discrete-power-semis/modified-sine-wave-inverter-enhanced.

6. Databalder xmega256A3BU. [Internett] Mai 2013. [Sitert: 14 Mai 2013.] http://www.atmel.com/devices/atxmega256a3bu.aspx?tab=documents.

7. USB Standard I/O Example for XMEGA-A3BU. [Internett] 23 April 2013. [Sitert: 14 Mai 2013.] http://asf.atmel.com/docs/latest/common.utils.stdio.stdio_usb.example.atxmega256a3bu_xmega_a3bu_xplained/html/index.html.

8. **Texas Instruments.** LMD 18200 Datablad. 2013.

9. **Berg, Mikael.** *Naviagtion with Simultaneous Localization and Mapping.* Trondheim : NTNU, 2013.

10. **Microsoft.** Serial Port Class. [Internett] [Sitert: 31 Mai 2013.] http://msdn.microsoft.com/en-us/library/system.io.ports.serialport.aspx?cs-save-lang=1&cs-lang=csharp#code-snippet-1.

11. —. [Internett] 2012. [Sitert: 09 Desember 2012.] http://msdn.microsoft.com/en-us/library/windows/desktop/hh404562(v=vs.85).aspx.

12. —. Kinect head tracking with a transparent display. [Internett] 05 Juni 2012. [Sitert: 09 Desember 2012.] http://blogs.technet.com/b/next/archive/2012/06/05/head-tracking-with-a-transparent-display.aspx#.UMS9-zMZytM.

13. **Yumo.** Datablad rostasjonenkoder E6A2.