

Systemskiss

Redaktör: Pål Kastman

Version 0.1

Status

Granskad		
Godkänd		

PROJEKTIDENTITET

HT1, 2014, Grupp 2
Linköpings Tekniska Högskola, ISY

Gruppdeltagare

Namn	Ansvar	Telefon	E-post
Pål Kastman	Projektledare	0703896295	palka285@student.liu.se
Hannes Snögren	Dokumentansvarig	0706265064	hansn314@student.liu.se
Alexander Yngve		0762749762	aleyn573@student.liu.se
Martin Söderén		0708163241	marso329@student.liu.se
Daniel Wassing		0767741110	danwa223@student.liu.se
Dennis Ljung		0708568148	denlj069@student.liu.se

Hemsida: <http://github.com/ultralaserdeluxe/gloria>

Kund: Tomas Svensson

Kontaktperson hos kund: Tomas Svensson

Kursansvarig: Tomas Svensson

Handledare: Peter Johansson

Innehåll

1	Inledning	1
2	Översikt av system	2
2.1	Kommunikation	2
2.2	Uppgraderbarhet	3
3	Huvudmodul	3
3.1	Hårdvara	4
3.2	Autonomt läge	4
3.2.1	Arbetsblock	5
3.3	Manuellt läge	6
3.3.1	Arbetsblock	6
4	Sensormodulen	6
4.1	Reflexsensormodul	6
4.1.1	Arbetsblock	6
4.2	Avståndssensor	7
4.2.1	Arbetsblock	7
4.3	Flödesschema	7
5	Styrenhet	7
5.1	Framdrivning	7
5.1.1	Arbetsblock	8
5.2	Robotarm	8
5.2.1	Arbetsblock	9
5.3	Processor	9

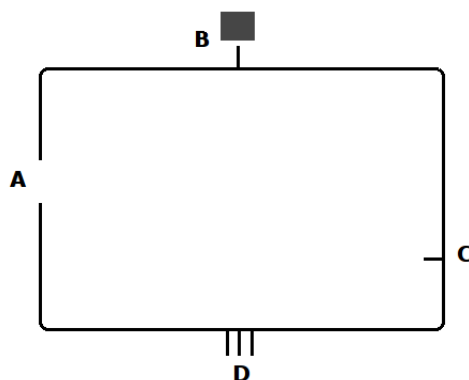
Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1		Första utkast		

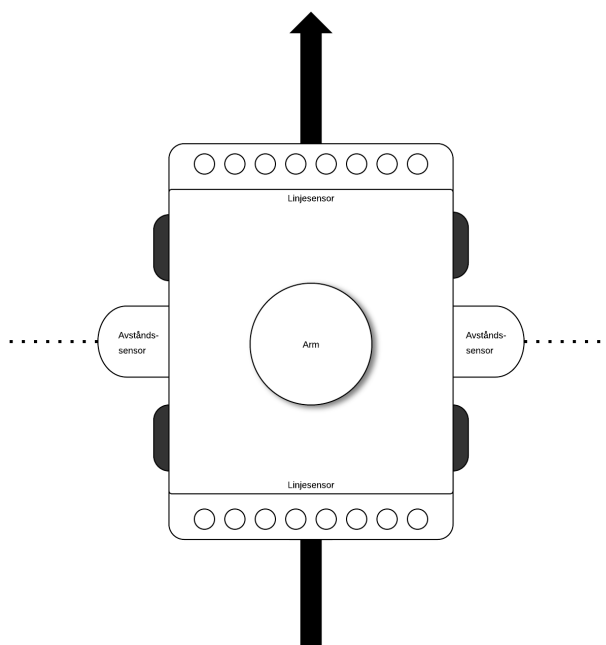
1 Inledning

Systemet kommer bestå av en robot som i sig består av tre undermoduler. Utöver detta tillkommer mjukvara på roboten och PC som används för manuell styrning och övervakning. Detta är en systemskiss som ska ge en grov översikt hur systemet ska implementeras. Detta ska sedan vara underlag för designspecifikationen.

Varje enskild modul kommer att ha en egen processor och dessa kommer sedan kommunicera med varandra. Systemet kommer kommunicera med en PC över blåtand. När systemet är klart för leverans ska det kunna följa en bana som visas nedan och plocka upp och sätta ned paket på de utsatta stationerna B och C. Den ska även kunna detektera avbrott (A) samt slutstation (D).



Banöversikt



Robot sedd ovanifrån

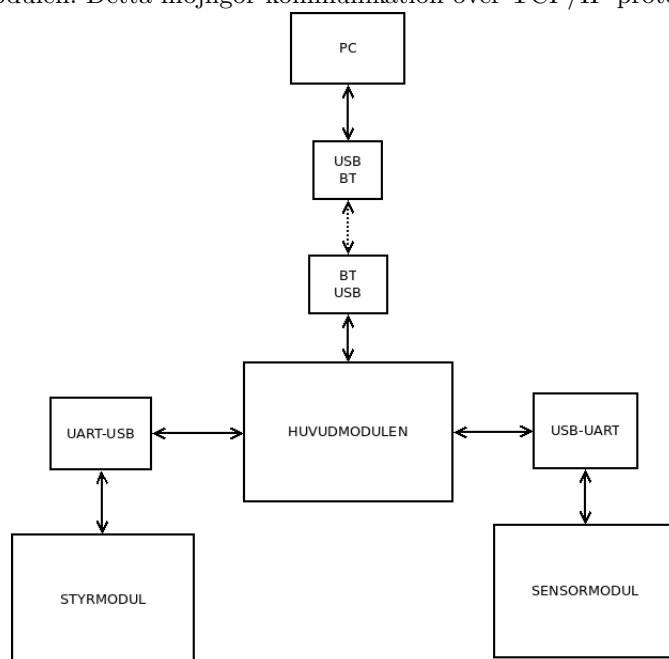
2 Översikt av system

På plattformen kommer tre moduler att installeras:

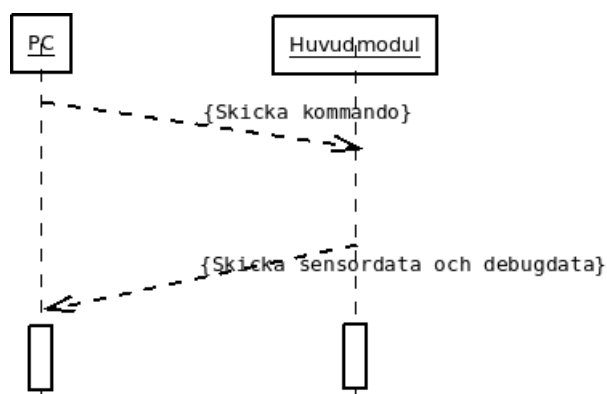
- Huvudmodul
- Styrmodul
- Sensormodul

2.1 Kommunikation

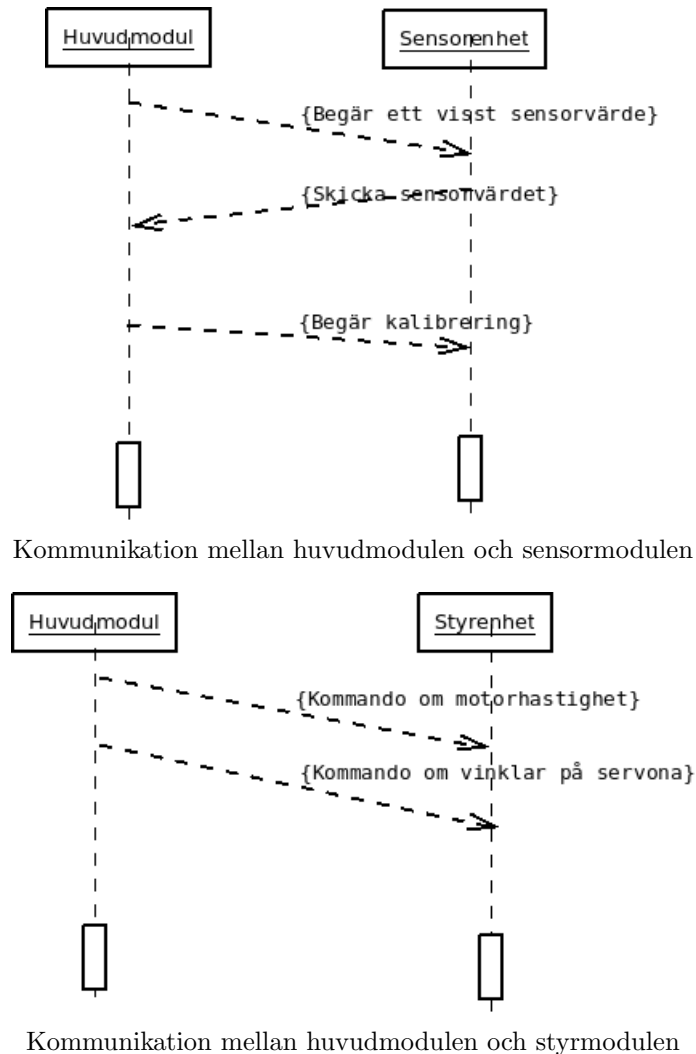
Kommunikationen mellan Huvudmodulen, styrmodulen och sensormodulen kommer förslagsvis att ske över UART. Huvudmodulen kommer isåfall ha två stycken USB-RS232 konverterare som kommer kopplas till var sin modul. Kommunikationen från huvudmodulen till PC kommer att ske över Bluetooth. Förslaget är att sätta upp ett PAN (PERSONAL AREA NETWORK) mellan PC:n och huvudmodulen. Detta möjliggör kommunikation över TCP/IP protokollet.



Flödesschema över kommunikationen i systemet



Kommunikation mellan PC och huvudmodulen



2.2 Uppgraderbarhet

Tydliga kommunikationsprotokoll ska existera eftersom det blir enklare att byta ut en modul. Kommunikationen mellan huvudmodulen och övriga moduler sker förslagsvis över UART. Eftersom detta sker med hjälp av USB-UART donglar isåfall så finns det möjlighet att lägga till fler moduler i framtiden eftersom Beagleboarden har fyra usbportar och tre kommer användas i systemet. Behövs fler i framtiden så kan en USB-hubb användas.

Kommunikationen mellan huvudmodulen och PC:n kommer förslagsvis att ske över bluetooth och isåfall kommer den användas för att sätta upp ett PAN. Över denna sätts en TCP/IP anslutning upp och data skickas över en Python-socket vilket leder till att bluetooth kan bytas ut mot WIFI eller en ethernetkabel.

3 Huvudmodul

Huvudmodulen kan ses som systemets "hjärna". Här sker alla beräkningar för att roboten ska kunna utföra sina uppgifter. Dessa uppgifter ska huvudmodulen hantera antingen via kommandon från en PC eller skötas helt autonomt. Detta är en väldigt kritisk modul då den kommer att utföra väldigt mycket uppgifter. Den har inte mycket hårdvara runt om sig men den kommer vara väldigt

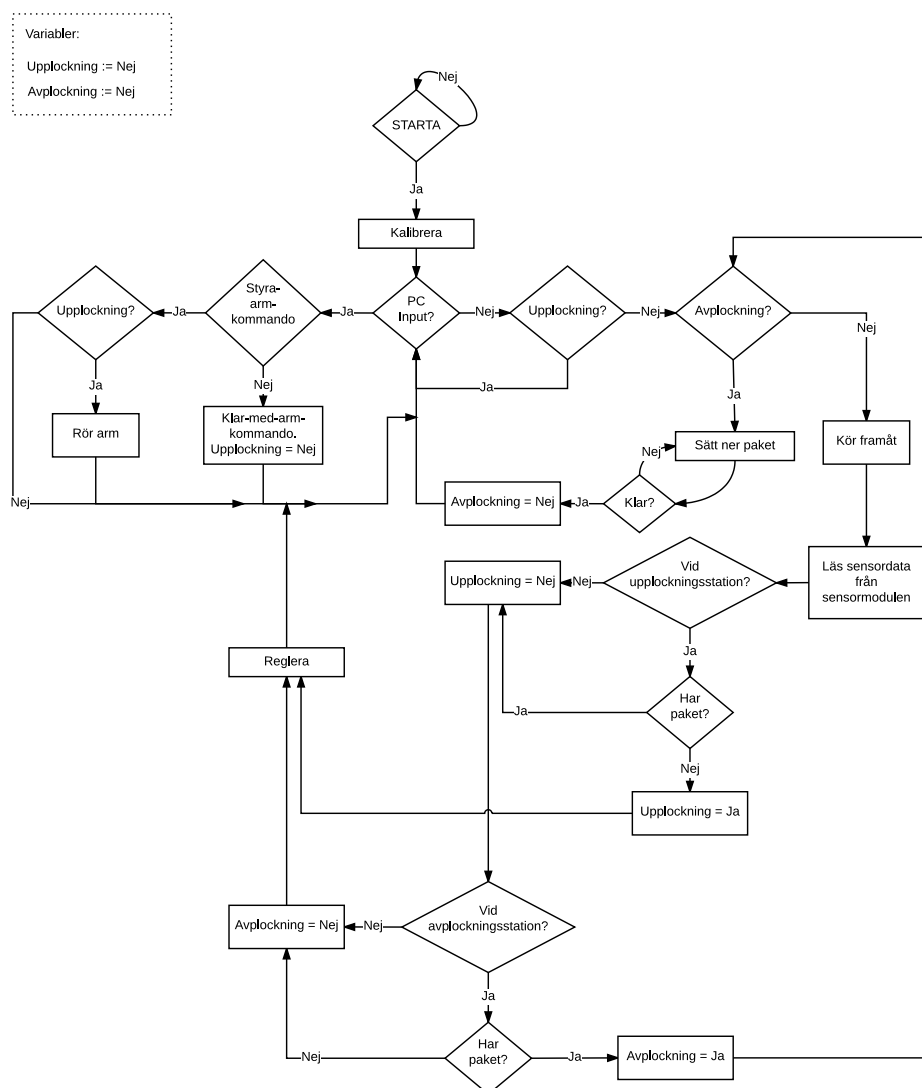
mjukvara för att utföra sina uppgifter. Här är det viktigt att all kod är väldigt bra strukturerad då den troligtvis kommer ändras väldigt mycket under tiden gång.

3.1 Hårdvara

Modulen ska förslagsvis bestå av en enkortsdator av modell Beagleboard. Den har en ARM Cortex-A8 processor som har en klockfrekvens på 1GHz. Denna behöver ett operativsystem för att kunna användas. Den enda hårdvaran som behövs för att kunna använda BB är ett minneskort för operativsystem och en bluetooth-dongel för kommunikationen med PC:n. Även två USB-UART konvertera behövs för kommunikationen med de övriga modulerna.

3.2 Autonomt läge

När roboten är satt i autonomt läge kommer all styrning skötas av en algoritm i huvudmodulen, med undantag för upphämtning av paket där användaren styr armen. Detta är en väldigt kritisk del då detta är denna algoritm som avgör robotens betende och som sedan kommer leda till att den kan fullfölja sina uppgifter. Troligtvis kommer en stor del av tiden läggas på att felsöka och optimera algoritmen. För att roboten ska kunna färdas framåt smidigt så kommer den behöva någon slags reglering. Förslagsvis ska en PD reglering implementeras som arbetar tidsdiskret. Detta gör att man måste sampla data från linjesensorerna med ett visst intervall. Detta intervall är ej bestämt än.



Flödesschema för huvudmodules styralgorithm

Ovanstående bild är en visualisering av hur det autonoma läget kommer att styras. Med hjälp av två stycken flaggor, Upplockning och Avplockning, kan systemet avgöra om den ska plocka upp eller lämna av ett paket eller bara fortsätta köra framåt längs banan. När Upplockning är satt till "Ja" väntar systemet på att användaren ska styra armen och sedan skicka ett färdigt-kommando".

3.2.1 Arbetsblock

- Vänta på startsignal
- Kalibrera sensorer
- Hämta sensorvärden från sensormodul

- Beräkna nästa lämpliga operation
- Skicka kommandon till styrmodulen hur roboten ska färdas
- Reglera för att följa linjen
- Om upplockning ska ske:
Vidarebefodra kommandon från användaren till styrmodulen för armen
- Skicka sensorvärden till användaren

3.3 Manuellt läge

När roboten är satt i manuellt läge väntar huvudmodulen på kommandon från användaren och utför sedan dessa.

3.3.1 Arbetsblock

- Hämta kommandon från användaren
- Vidarebefodra kommandon till rätt modul
- Skicka sensorvärden till användaren

4 Sensormodulen

Sensormodulen har som uppgift att läsa in data från robotens sensorer, omvandla dem om det är nödvändigt och vidarebefodra dem till huvudmodulen. För att roboten ska kunna hitta behöver den först veta vart den är och till detta så används två stycken linjesensorer. För att kunna detektera paket kommer roboten på var sin sida ha två avståndssensorer. Om det finns tid så ska det sitta en linjesensor på armen för att kunna detektera formen på paket vilket möjliggör utökning till autonom upplockning.

4.1 Reflexsensormodul

För att kunna beräkna vilken vinkel roboten har i förhållandet till linjen som den följer så har den en reflexmodul fram och en bak. Dessa består av 11 stycken reflexdetektorer som består av en lysdiod och en ljuskänslig transistor. Varje detektor har en enable ingång som slår på lysdioden och en OUT utgång som ger en analog signal mellan 0V och 5V. Dessa kommer läsas av var för sig och när en detektor inte läses av kommer enable vara låg för att inte påverka närliggande detektorer samt för att de inte ska överhettas. Avläsningen kommer att ska med två muxar av typen MC14067B vilket är en analog mux med 16 kanaler.

4.1.1 Arbetsblock

- kalibrera värden
- ställ in muxarna
- vänta på lysdiod för att stabilisera sig
- läs av analog värde
- omvandla analog värde till digital
- ta hänsyn till kalibreringen om det är en tejp eller ej under detektorn och trunkera datat efter detta
- skicka värden till huvudmodulen

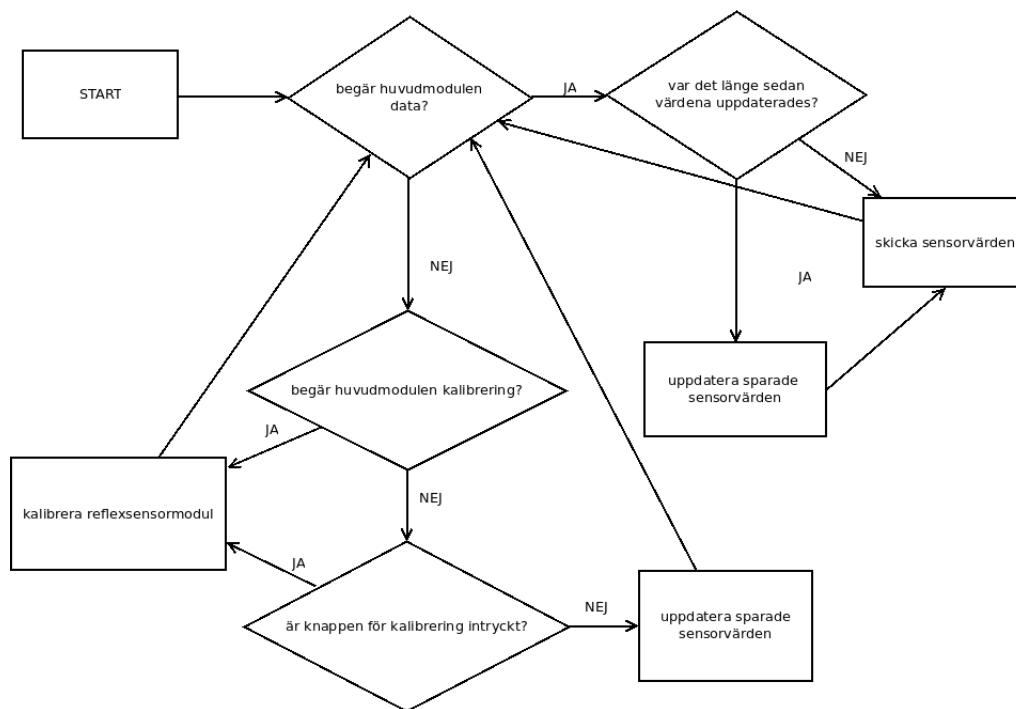
4.2 Avståndssensor

För att kunna detektera om det är ett paket på någon sida av roboten vid stationer så kommer det sitta en avståndssensor på vardera sida av roboten. Dessa kommer vara av typen GP2D120 som är en avståndssensor som använder IR för att generera en analog signal. Denna signal är ej linjär så antingen måste funktionen linjäriseras alternativt så används ett lookuptable som den interpolerar över. Då muxarna som används till reflexsensormodulen har 16 kanaler och de endast använder 11 så kopplas dessa in samma. Dessa kommer dock vara aktiverade hela tiden.

4.2.1 Arbetsblock

- ställ in muxarna
- läs av analog värde
- omvandla analog värde till digital
- kolla upp avstånden i ett lookuptable
- skicka värden till huvudmodulen

4.3 Flödesschema



Flödesschema för sensormodulen

5 Styrenhet

Styrenheten har till uppgift att driva de motorer som driver hjulen och de servon som styr armen.

5.1 Framdrivning

Motorenheten innehåller två hjulpar, de styrs separat för att göra det möjligt att svänga. Huvudmodulen skickar kommandon för hur snabbt vardera hjulpar skall rotera, styrenheten ser sedan till att motorerna kör i den efterfrågade hastigheten. Detta sker dock utan någon återkoppling från

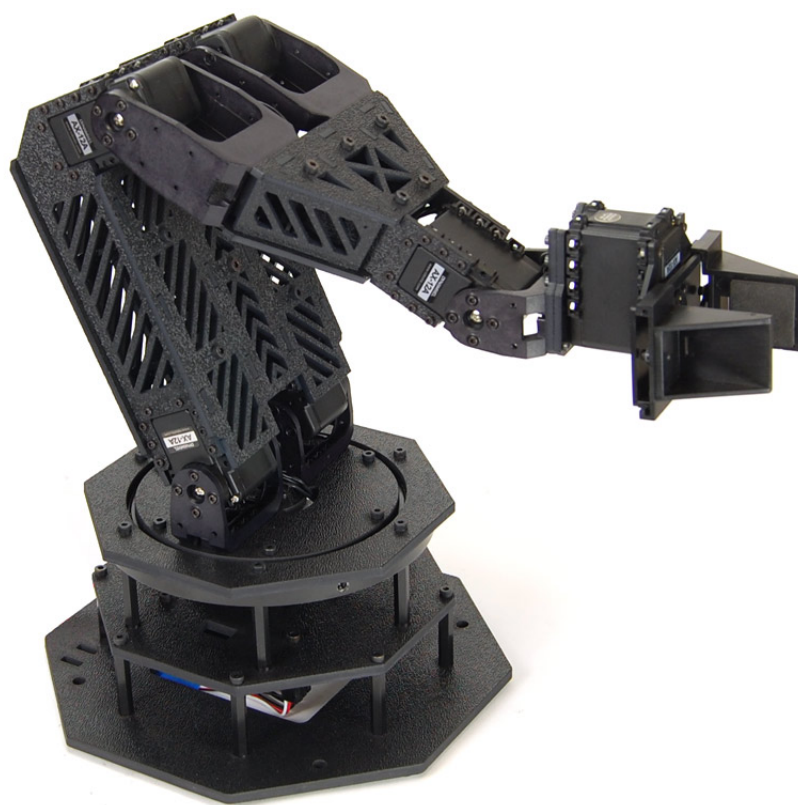
den verkliga hastigheten utan vi mäter upp vilken signal till motorerna som ger vilken hastighet och sedan används detta för att styra hastigheten under användning.

5.1.1 Arbetsblock

- Ta emot kommandon från huvudmodulen om vilken hastighet båda hjulparen ska hålla
- Omvandla hastigheten till lämpliga PWM signaler
- Starta motorn mjukt
- Lägg PWM signalerna på motorernas utgångar
- Stanna motorn mjukt

5.2 Robotarm

Robotarmen består av 7 servon av modell AX12-A. Dessa styrs genom att en målvinkel sätts (0-1023), med möjlighet att ändra hastighet, vridmoment och styra av/på. Från huvudmodulen får enheten målvinklar för varje enskild led. Styrmodulen är ansvarig för att se till att parallella servon körs synkroniserat, för att inte slita sönder varandra. Styrmodulen ansvarar också över att armen inte ska kollidera med resten av roboten. Den kommer att programmeras med vilka rymdmängder som den ej får passera och om den får kommandon som gör att den kommer passera dessa mängder måste den välja en annan väg än rakaste vägen.



5.2.1 Arbetsblock

- Ta emot kommandon från huvudmodulen om vilka vinklar alla servon ska ha
- Kolla om armen kommer röra sig igenom förbjuden rymd och beräkna en alternativ väg
- Skicka kommandon till servona och se till så att de rör sig mjukt

5.3 Processor

Eftersom motorenheten kommunicerar med huvudmodulen genom UART kommer kommandon att tolkas med avbrott. Förslagsvis ska en Atmega16 microcontroller användas.