

Designspecifikation

Redaktör: Hannes Snögren

Version 1.0

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	Hårdvaruansvarig	0762749762	aleyn573@student.liu.se
Martin Söderén	Mjukvaruansvarig	0708163241	marso329@student.liu.se
Daniel Wassing	Leveransansvarig	0767741110	danwa223@student.liu.se
Dennis Ljung	Testansvarig	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

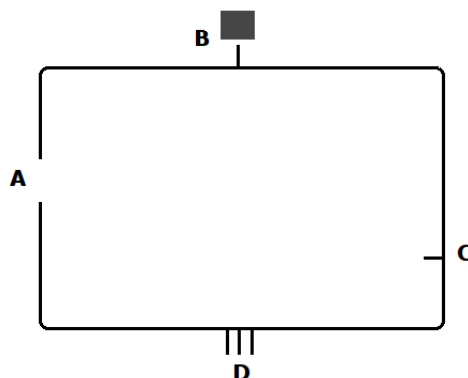
1	Inledning	1
2	Översikt av system	1
2.1	Kommunikationskanaler	1
2.2	Uppgraderbarhet	1
3	Protokoll för kommunikation	3
3.1	Kommunikation mellan PC-enhet och huvudenhet	3
3.1.1	Instruktioner	3
3.2	Kommunikation mellan huvudenhet och sensorenhet	4
3.2.1	Instruktioner	4
3.3	Kommunikation mellan huvudenhet och styrenhet	4
3.3.1	Instruktioner	5
4	PC-enhet	5
4.1	libgloria	5
4.2	GUI	6
4.3	Komponenter	6
5	Huvudmodul	7
5.1	Hårdvara	7
5.2	Mjukvara	7
5.2.1	Huvudtråden	8
5.2.2	Sensortråden	9
5.2.3	PC-tråden	9
5.3	Reglering	9
5.4	Styrning av arm	9
5.5	Komponenter	12
6	Sensorenhet	12
6.1	Reflexsensormodul	12
6.1.1	Kalibrering	12
6.2	Avståndssensor	13
6.3	Mjukvara	13
6.4	Komponenter	13
7	Styrenhet	13
7.1	Framdrivning	14
7.2	Robotarm	14
7.3	Mjukvara	14
7.4	Komponenter	15
8	Testning av system	15
8.1	Test av delsystem	15
8.2	Test av kommunikationskanaler	15
8.3	Integrationstest	15
	Referenser	16
	Bilaga A Kopplingsschema över sensorenhet	17
	Bilaga B Kopplingsschema över styrenhet	18

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2014-10-13	Första utkast	hansn314	
0.2	2014-10-28	Andra utkast	hansn314	
1.0	2014-11-03	Första version	hansn314	

1 Inledning

Den här designspecifikationen är menad att noggrant specificera hur det system som beskrivs i projektets kravspecifikation skall konstrueras. Detta system skall kunna följa en bana enligt figur 1 där det skall kunna plocka upp och sätta ned paket på de utsatta stationerna B och C. Det skall även kunna detektera och hantera avbrott i banan (A) samt slutstationer (D).



Figur 1 – Banöversikt

2 Översikt av system

Plattformen består av sammanlagt fyra enheter. En PC-enhet, en huvudenhet, en sensorenhet och en styrenhet.

PC-enheten är framförallt ett användargränssnitt som gör det enkelt för användaren att styra roboten och robotarmen men fungerar även som ett debugverktyg där användaren kan se robotens status och debuginformation. Huvudenheten står för huvuddelen av alla beräkningar roboten behöver göra så som regleringsalgoritm för linjeföljaren och koordinatkonvertering för armen. Sensorenheten innehåller den funktionalitet som krävs för att driva alla systemets sensorer och styrenheten innehåller den funktionalitet som krävs för att köra motorer och servon.

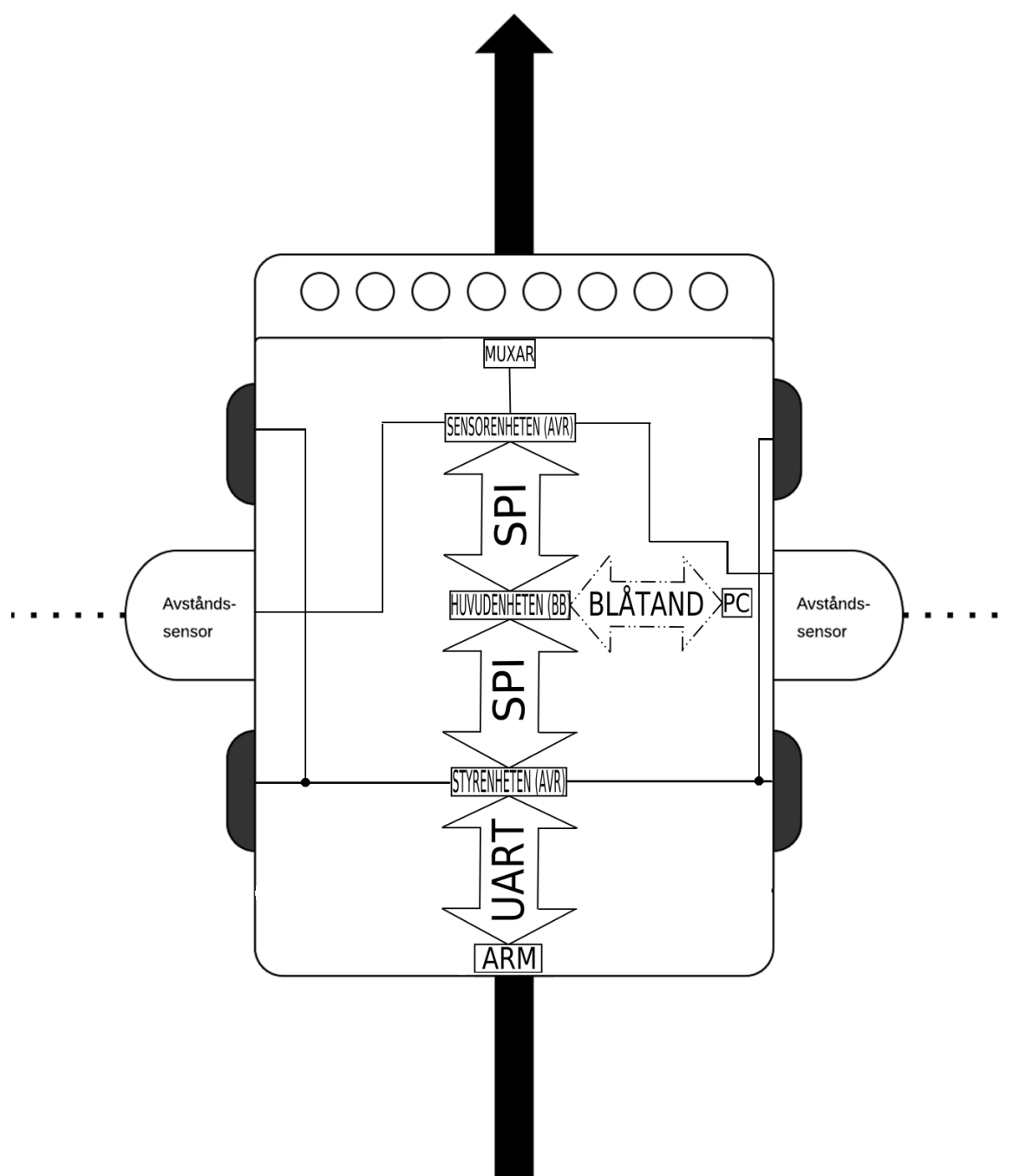
2.1 Kommunikationskanaler

I det fall att roboten arbetar i manuellt läge, skickar PC-enheten kommandon till huvudenheten över Blåtand. I annat fall arbetar roboten autonomt och PC-enheten skickar endast förfrågningar angående robotens status. Huvudenheten skickar kommandon till sensorenheten och styrenheten över en SPI-buss.

2.2 Uppgraderbarhet

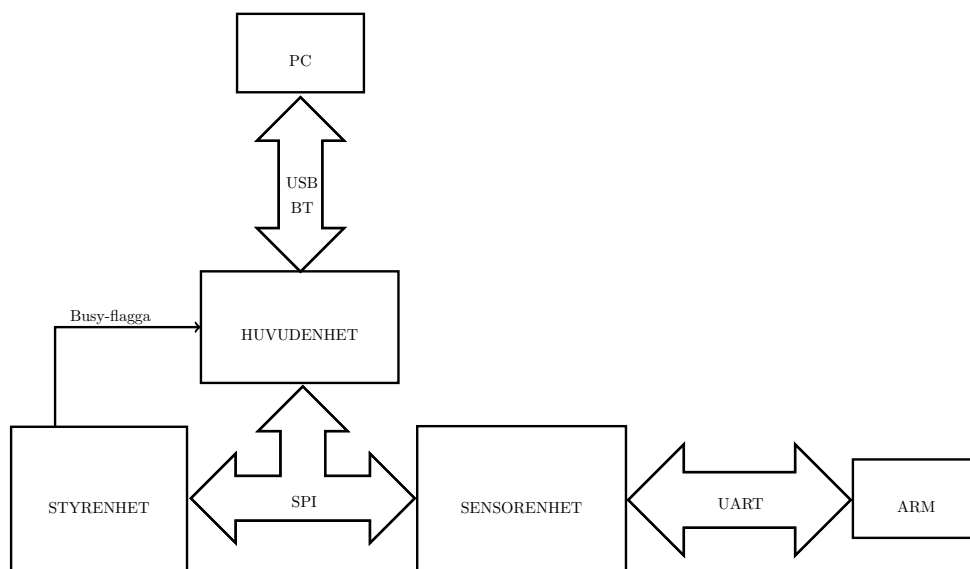
Kommunikation mellan huvudmodulen och PC-enheten kommer att ske över Blåtand och kommer att användas för att sätta upp ett PAN. Över denna sätts en TCP/IP anslutning upp och data skickas över en Python-socket vilket leder till att bluetooth kan bytas ut mot till exempel WIFI, en ethernetkabel eller annan överföringskanal som tillåter TCP/IP.

Kommunikationen mellan huvudmodulen och styrenheten samt sensormodulen sker över en SPI-buss med ett väldefinierat protokoll. Om det önskas kan sensormodulen och styrmodulen bytas ut mot andra moduler så länge de använder samma protokoll. Styrenheten kommunicerar med servona över UART och använder ett protokoll speciellt för Trossenrobotics Reactor vilket leder till att armen inte kan bytas ut mot någon annan.



Figur 2 – Översikt över roboten

3 Protokoll för kommunikation



Figur 3 – Flödesschema över kommunikationskanalerna i systemet

3.1 Kommunikation mellan PC-enhet och huvudenhet

Kommunikation mellan PC-enhet och huvudenhet sker via Blåtand. En instruktion ges som en sträng på formen $command_1 = arg_1, \dots, arg_N; command_2 = arg_1, \dots, arg_N$ med ett godtyckligt antal instruktioner där antalet argument per instruktion specificeras i avsnitt 3.1.1.

3.1.1 Instruktioner

Instruktion	Argument	Beskrivning
motor	L,R	L och R anger hastigheten på vänster, respektive höger hjulpar
arm	X,Y,Z,P,G	X,Y,Z är koordinaten i rummet dit armens gripklo skall röra sig. Armens fundament är origo, P anger handens vinkel i förhållande till XY-planet och G anger avståndet mellan gripklons klor
calibrate		Begär kalibrering av robotens sensorer
status		Begär statusrapport från robot
automotor	M	Ange om robotens motorer skall styras från PC-enheten (0) eller huvudenheten (1)
autoarm	M	Ange om robotarmen skall styras från PC-enheten (0) eller huvudenheten (1)
start		Initiera körning

Tabell 1 – Instruktioner från PC-enhet till huvudenhet

L och R:s begränsningar där 100 representerar maximal hastighet framåt och -100 representerar maximal hastighet bakåt

$$-100 \leq L \leq 100$$

$$-100 \leq R \leq 100$$

Begränsningar på X,Y,Z motsvarar hur långt roboten kan sträcka sig i varje led

$$-3800 \leq X \leq 3800$$

$$-3800 \leq Y \leq 3800$$

$$0 \leq Z \leq 4600$$

Begränsningarna på G motsvarar servots input för position

$$0 \leq G \leq 1024$$

Begränsningarna på P motsvarar vilken vinkel robotens gripklo kan ha mot XY planet.

$$-90 \leq P \leq 90$$

3.2 Kommunikation mellan huvudenhet och sensorenhet

Kommunikation mellan huvudenhet och sensorenhet sker via en SPI-buss. Instruktioner från huvudenheten ges i form av en byte. De fyra första bitarna anger vilken instruktion som skall utföras. De sista fyra anger vilken sensor instruktionen gäller. Tabell 2 och 3 under avsnitt 3.2.1 specificerar de instruktioner som finns respektive vilken adress som avser vilken sensor.

Sensorenheten svarar med endast data. En byte per fototransistor i linjesensorn. En byte för vardera avståndssensor. I fallet med instruktionen *läs data från alla sensorer* skickas sensorernas data seriellt i följande ordning: 1. Linjesensor, 2: Avståndssensor Höger, 3: Avståndssensor Vänster.

3.2.1 Instruktioner

Instruktion	Argument	Beskrivning
0000	A	Returnera sensordata för A
0001	A	Kalibrera sensor A

Tabell 2 – Instruktioner från huvudenhet till sensorenhet

Adress	Beskrivning
0000	Linjesensor Fram
0010	Avståndssensor Höger
0011	Avståndssensor Vänster
1111	Adressera samtliga sensorer

Tabell 3 – Adresser för instruktioner till sensorenhet

3.3 Kommunikation mellan huvudenhet och styrenhet

Kommunikation mellan huvudenhet och styrenhet sker via en SPI-buss. Det finns en busy-flagga kopplad till huvudenheten som går låg när armen är i rörelse.

En instruktion består av tre bytes. Den första byten innehåller instruktionen och vilket servo eller vilken motor som avses. De första fyra bitarna av denna byte definierar vilket kommando som skall utföras. De nästkommande fyra vilken enhet det skall utföras av. Tabell 4 och 5 i avsnitt 3.3.1 specificerar de instruktioner som finns och vilken adress som avser vilken motor eller servo. Efter instruktions-byten följer alltid två databytes. I de fall där endast en databyte är nödvändig används endast den första av databytesen. Oanvända databytes kasseras utan att tolkas av styrenheten.

3.3.1 Instruktioner

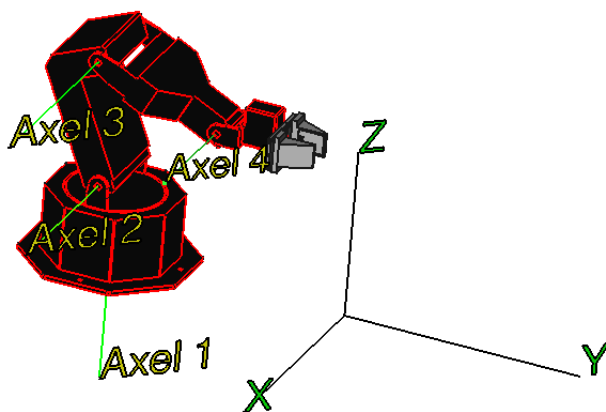
Instruktion	Argument	Beskrivning
0000		Stoppa samtliga servon och motorer
0001	A, D	Sätt register A till D
0010	A	Utför givna kommandon för A

Tabell 4 – Kommandon från huvudenhet till styrenhet

Adress	Beskrivning
0000	Höger hjulpar
0001	Vänster hjulpar
0010	Arm axel 1
0100	Arm axel 2
0110	Arm axel 3
1000	Arm axel 4
1011	Arm axel 5 (<i>gripklo</i>)
1100	Samtliga motorer
1101	Samtliga servon
1111	Samtliga motorer och servon

Tabell 5 – Adresser för adressering till styrenhet

Adresserna Arm axel 1 till 4, refererar till armens frihetsgrader enligt figur 4.



Figur 4 – Till vänster: Armens frihetsgrader Till höger: Armens koordinatsystem

4 PC-enhet

Programvaran på PC-enheten delas in i två komponenter, ett grafiskt gränssnitt som användaren interagerar med och en samling procedurer som det grafiska gränssnittet använder för att kommunicera med roboten.

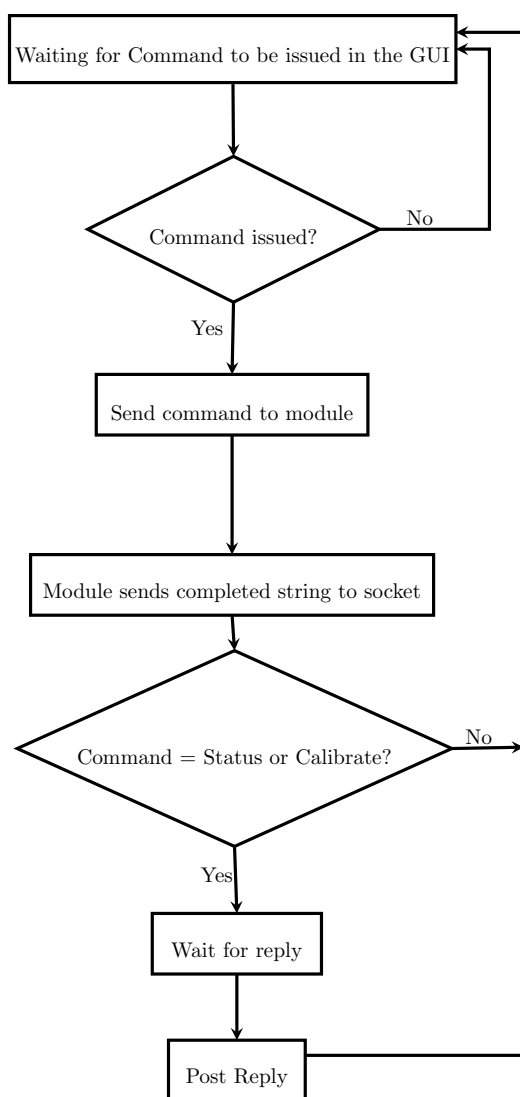
4.1 libgloria

libgloria är en Python-modul som kan importeras av andra program för att möjliggöra kommunikation med huvudenheten i Gloria-systemet. Vid initiering av modulen upprättas en Blåtandsförbindelse mellan PC-enheten och huvudenheten. Blåtandsförbindelsen sker över TCP via sockets.

Efter detta kan modulens funktioner användas för att skicka kommandon till roboten. Formatet på kommandon beskrivs i avsnitt 3.1.

4.2 GUI

Det grafiska gränssnittet ska skrivas i Python med hjälp av mjukvarubiblioteket Tkinter. Gränssnittet ska visa robotens nuvarande status, så som motorhastighet, sensorvärden och armens position. Det ska även innehålla reglage för att välja mellan autonomt och manuellt läge samt för att styra robotens arm och motorer. Gränssnittet ska innehålla så lite logik som möjligt och bara vara ett tunt lager ovanpå libgloria.



Figur 5 – Flödesschema över mjukvaran på PC-enheten

4.3 Komponenter

- En PC med Python 3 installerat
- En Blåtandsdongel

5 Huvudmodul

Huvudmodulen utför de flesta beräkningar nödvändiga för att roboten ska kunna utföra sina uppgifter. Dessa uppgifter ska huvudmodulen hantera antingen via instruktioner från PC-enheten eller helt autonomt. Detta är en kritisk modul då den kommer att utföra många uppgifter. Den behöver inte mycket hårdvara men den kommer innehålla majoriteten av robotens programvara.

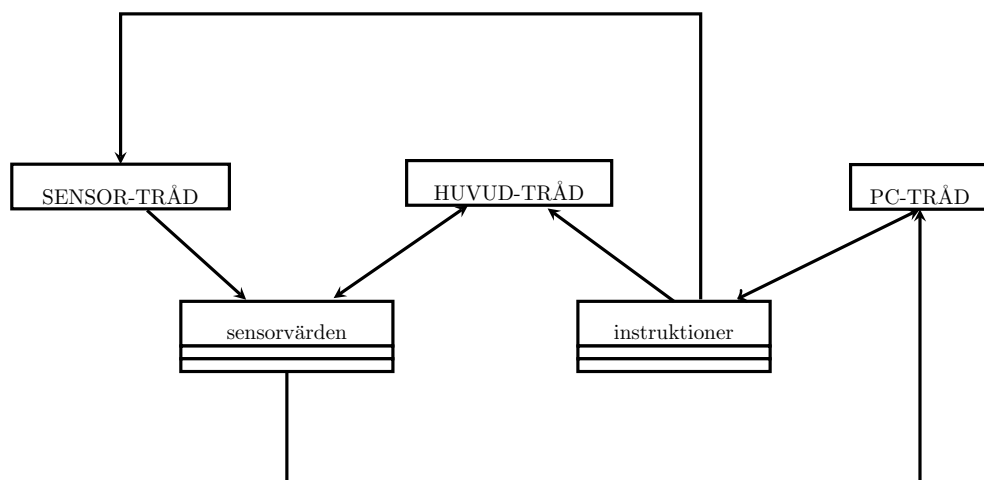
5.1 Hårdvara

Modulen ska bestå av en enkortsdator av modell Beagleboard-xM. 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 Beagleboard är ett minneskort för operativsystem och en Blåtandsdongel för kommunikationen med PC-enheten. Kommunikation med sensorenhet och styrenhet sker över SPI.

5.2 Mjukvara

Mjukvaran som behövs för att implementera all funktionalitet hos huvudmodulen kommer att vara skriven i programspråket Python. Koden för programmet kommer vara trådbaserad och möjligen objektorienterad.

Programmet ska delas in i 3 trådar. Dessa körs kontinuerligt och delar på två trådsäkra listor: sensorvärden och instruktioner. I instruktionslistan ligger alla argument till tillhörande instruktioner från PC-enheten och i sensorvärden ligger all data från sensorenheten. Listorna kan trådarna antingen läsa eller skriva till (se figur 6).



Figur 6 – Trådarna och listorna de delar på

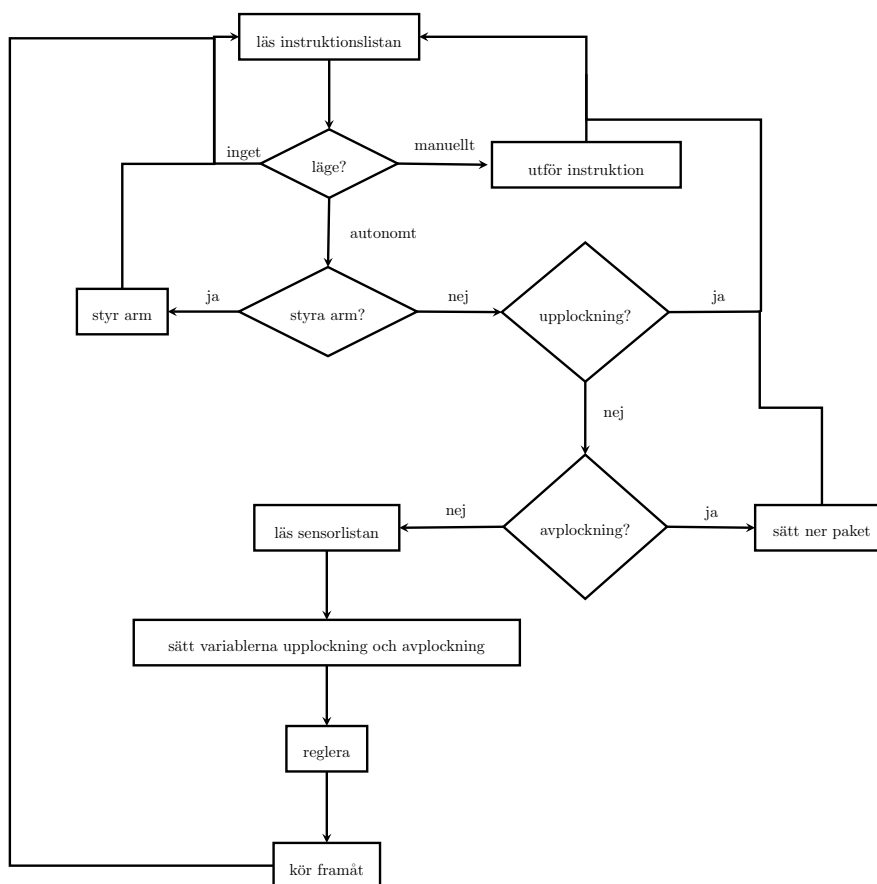
Det som behövs göras på Beagleboard innan den kan börja användas är:

- Installera Angstrom operativsystem
- Uppdatera operativsystem
- Installera Python
- Konfigurera SPI
- Patcha kärnan för att stödja SPI
- Installera Git

- Installera utvecklingsmiljö
- Installera Blåtandsmodulen
- Sätta upp ett PAN

5.2.1 Huvudtråden

I huvudtråden ligger huvudloopen för programmet. Den börjar med att läsa värdet från instruktionslistan där manuellt eller autonomt läge avgörs och anropar respektive funktion. Användaren ska alltså kunna växla mellan dessa lägen när denne vill (mellan varje iteration av huvudloopen). Figur 7 illustrerar programflödet.



Figur 7 – Flödesschema för huvudtråden

5.2.1.1 Autonomt läge

Här ska kod för autonom aktivitet hos systemet ligga. Sensorvärdeslistan behöver läsas av varje iteration för att avgöra om systemet befinner sig vid en av- eller upplockningsstation samt om systemet behöver regleras. Om systemet befinner sig vid en upplockningsstation behöver vi läsa av instruktionslistan för att få argument om att styra armen. Om systemet befinner sig vid en avplockningsstation ska systemet låsa sig tills paket är avsläppt.

Om systemet inte befinner sig vid en av- eller upplockningsstation skickas värden till styrenheten att köra roboten framåt. Dessa värden beräknas i slutet av programmet med regleringsalgoritmen utifrån linjesensordata från sensorvärdeslistan.

5.2.1.2 Manuellt läge

I manuellt läge läses argumenten från instruktionslistan av, tolkar dessa och skickar vidare lämpliga värden till styrenheten.

5.2.2 Sensortråden

Sensortrådets uppgift är att uppdatera sensorvärdeslistan med fräscha värden från sensorenheten.

5.2.3 PC-tråden

PC-tråden delar upp strängen med data från PC-enheten och lagrar instruktionerna och deras argument i instruktionslistan.

5.3 Reglering

För att undvika att roboten hackar sig fram och istället rör sig med smidiga rörelser så ska motorns styrsignaler regleras med en tidsdiskret PD-regulator. En PID regulator är onödig då vi inte har något stående fel i teorin. Om roboten är mitt på linjen och står rakt så kommer felet vara noll. I praktiken kommer det inte vara så då roboten troligtvis inte kommer stå rakt på linjen och motorerna kommer säkert åka i olika hastigheter. Dock så blir en PD-regulator tillräckligt bra. En annan anledning till att inte ha med I-delen är att det är svårt att implementera det bra i ett tidsdiskret system.

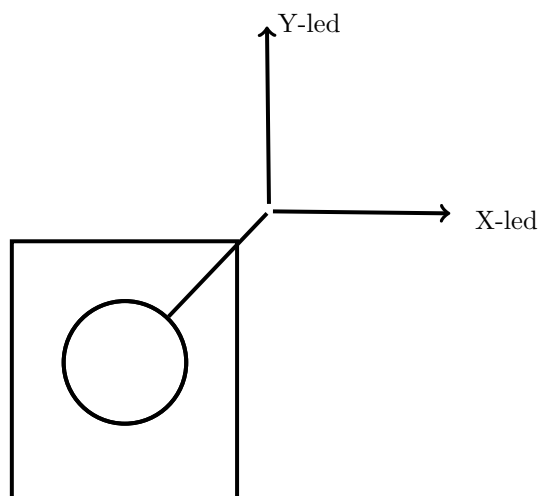
- $e[n]$ Är felet vid sampling n
- $e[n - 1]$ Är felet vid sampling $n-1$
- $u[n]$ är styrsignalen till motorerna
- K_P är en konstant för den proportionella delen av regleringen
- K_D är en konstant för den deriverande delen av regleringen
- ΔT är tiden mellan sampling n och $n-1$

$$u[n] = K_P * e[n] + K_D \frac{e[n] - e[n - 1]}{\Delta T}$$

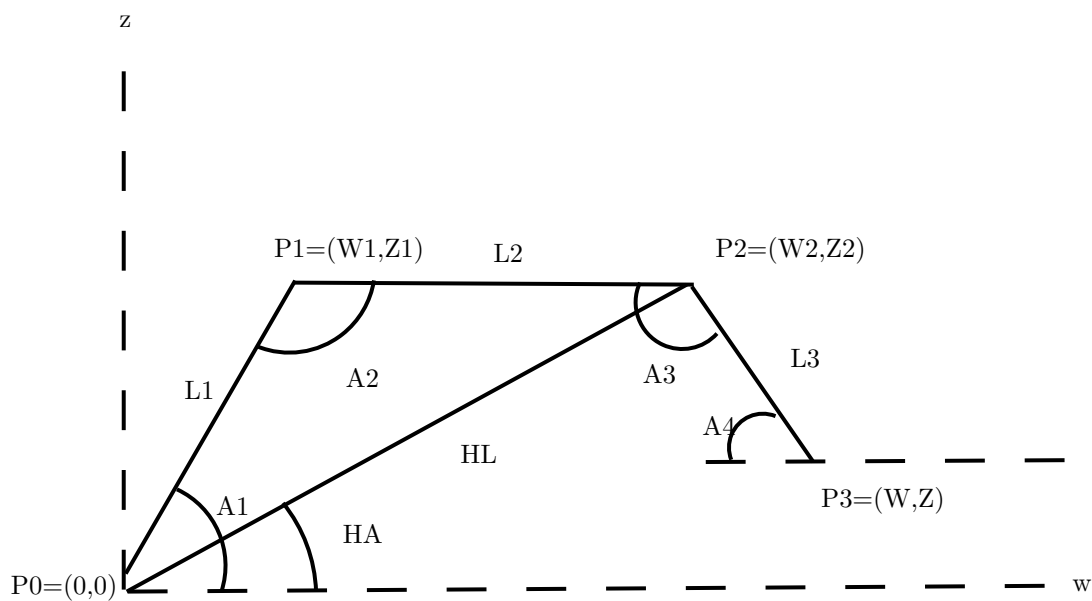
Roboten är begränsade av hårdvaran hur ofta den kan läsa av data från reflexsensorna. En rimlig uppdateringsfrekvens skulle kunna vara 25 Hz[1]. Så ΔT kommer vara 40ms.

5.4 Styrning av arm

Omvandlingen från x,y,z koordinater till vinklar för armens servon kommer att ske i huvudenheten då det är en krävande operation. Tanken är att när roboten stannar vid en station så ska användaren kunna styra robotens arm med en joystick eller från tangentbordet. När användaren beordrar armen att röra sig framåt så ska armen röra sig i positiv y-led i förhållande till roboten. När användaren beordrar roboten att styra åt höger så ska armens ändpunkt röra sig i positivt x-led i förhållande till roboten.



Figur 8 – Armens koordinatsystem i förhållande till roboten. Upp är robotens färdriktning



Figur 9 – Armens vinklar

För att detta ska vara möjligt så behöver vi använda lite trionometri. Omvandlingen sker på följande sätt: Antag att roboten start i:

$$X, Y, Z, GR = 0, 0, 0, \pi/2$$

Där GR(GripRadian) är vilken vinkel gripklon har till z-axeln.

Vi omvandlar:

$$X, Y, Z, GR \rightarrow A, W, Z, GR$$

Där A är armens vinkel mot positiva x-axeln och W är armens utsträckning.

$$A = \tan^{-1}\left(\frac{Y}{X}\right)$$

$$W = \sqrt{X^2 + Y^2}$$

Genom denna omvandlingen så kan vi göra om detta till ett 2-dimensionellt problem istället för ett 3-dimensionellt problem. Vi får in X,Y,Z,GR och ställer in robotens vinkel mot x-axelns positiva del och sedan ser vi till att roboten bara har rätt utsträckning och höjd.

$$P_4 = (W, Z)$$

$$P_0 = (0, 0)$$

$$W_2 = W - \cos(GR) * L_3$$

$$Z_2 = Z - \sin(GR) * L_3$$

$$H_L = \sqrt{W_2^2 + Z_2^2}$$

$$H_A = \tan^{-1}\left(\frac{Z_2}{W_2}\right)$$

Cosinussatsen ger:

$$A_1 = \cos^{-1}\left(\frac{L_1^2 + H_L^2 - L_2^2}{2 * L_1 * H_L}\right) + H_A$$

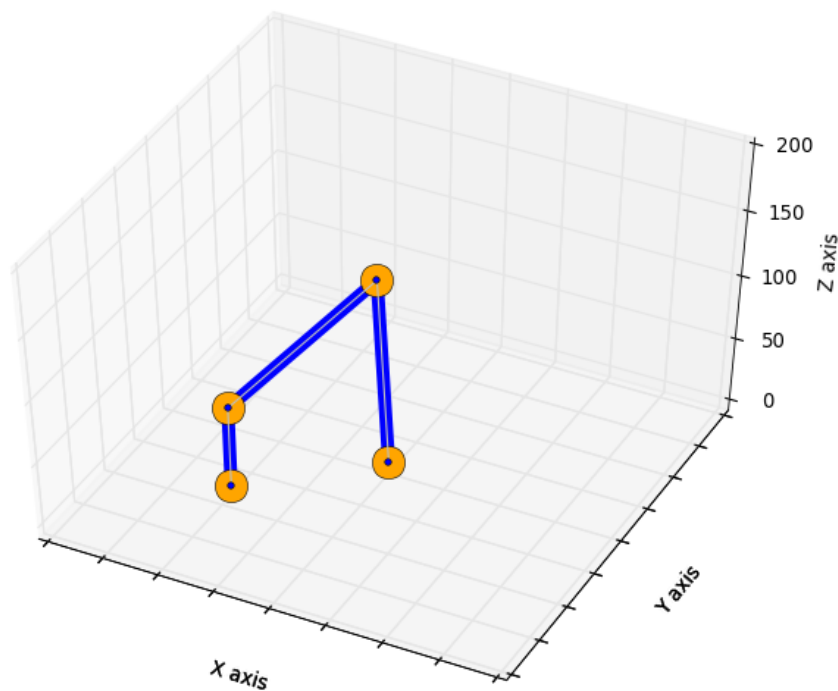
$$W_1 = \cos(A_1) * L_1$$

$$Z_1 = \sin(A_1) * L_1$$

$$A_2 = \tan^{-1}\left(\frac{Z_2 - Z_1}{W_2 - W_1}\right) - A_1$$

$$A_3 = GR - D_2 - D_3$$

Då har vi alla vinklar vi behöver. Sedan är det upp till styrenheten att omvandla dessa vinklar till positioner på servona på armen. Beräkningarna har simulerats i ett eget python-program (figur 10).



Figur 10 – Simulerad arm

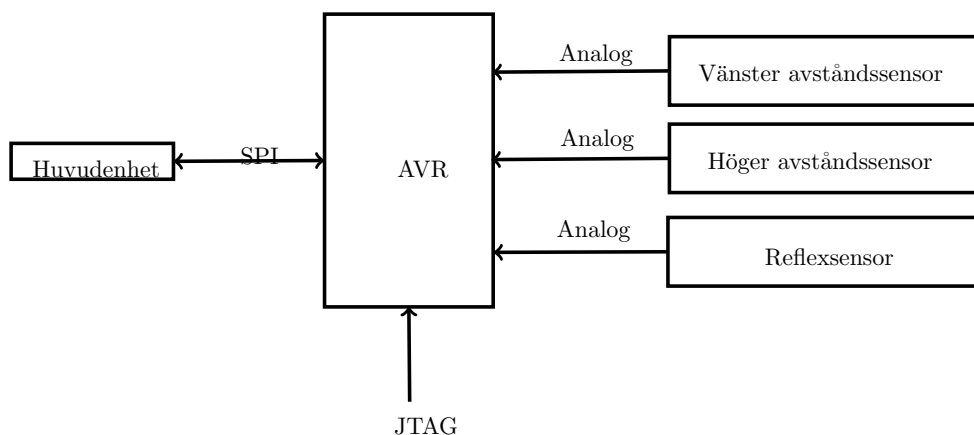
5.5 Komponenter

Följande komponenter är nödvändiga för konstruktion av huvudmodulen.

Komponent	Tillgänglighet
Beagleboard-xm	Tillgänglig
Blåtandsdongel Belkin f8t016	Tillgänglig
Minneskort	Skall beställas
Nivåomvandlare Ti TXB0108	Skall beställas. Förslagsvis från electrokit[2]
Wifidongel	Tillgänglig

6 Sensorenhet

Sensorenheten har till uppgift att läsa in data från robotens sensorer, tolka den och vidarebefodra den till huvudenheten. En reflexsensormodul används för att roboten skall kunna hålla sig på banan. För att kunna detektera paket kommer roboten ha en avståndssensor på vardera sida.



Figur 11 – Blockschema över sensorenheten

6.1 Reflexsensormodul

Reflexsensormodulen består av 11 reflexsensorer. En reflexsensor består av en lysdiod och en fototransistor. Fototransistorn har ett analogt utvärde mellan 0 och 5V beroende på hur mycket ljus som fångas upp. Genom att sätta enable för en lysdiod till 1 och sedan läsa av den tillhörande fototransistorn kan vi avgöra om underlaget är ljust eller mörkt. Fototransistorn läses av med en AD-omvandling på AVRen. Detta görs för varje reflexsensor och på så sätt kan vi detektera tejpens position under sensorn eftersom tejpens banan består av fär förutsättas ha en annan färg än golvet. Eftersom AVRen har ett begränsat antal pinnar med AD-omvandling, muxar vi fototransistorernas utgångar till en enda pinne på AVRen. Då vi inte vill ha mer än en lysdiod igång samtidigt kommer vi använda ytterligare en mux för att styra en enablesignal till den lysdiod vi vill använda, övriga kommer vara avslagna.

6.1.1 Kalibrering

Beroende på vilket underlag roboten arbetar på kommer värdena för golv och tejp att variera. Vi behöver därför kunna kalibrera sensorn för att sätta standardvärden. Detta kommer ske på ett sådant sätt att vi först låter roboten titta på bara golvet och sedan en bit tejp. Vi sparar de värden vi får under dessa testfall och använder dem som referens när vi ska detektera golv eller tejp.

6.2 Avståndssensor

För att kunna detektera paket kommer roboten ha en avståndssensor på höger respektive vänster sida. Dessa kommer vara av typen GP2D120 och använder IR för att generera en analog signal. Denna signal läses av med en AD-omvandling på AVR:n. Då sensorns utsignal är olinjär kommer vi under utvecklingsfasen ta fram en tabell med närmevärden för olika distanser. Utsignalen jämförs med denna tabell för att estimeras det uppmätta avståndet till ett föremål.

6.3 Mjukvara

Sensorenheten kör kontinuerligt en loop där den läser och lagrar data från sensorerna. Då huvudenheten skickar en instruktion till sensorenheten körs en avbrottsrutin där instruktionen tolkas och utförs. Antingen kalibreras sensorerna eller så returnerar sensorenheten data för den adresserade sensorn.

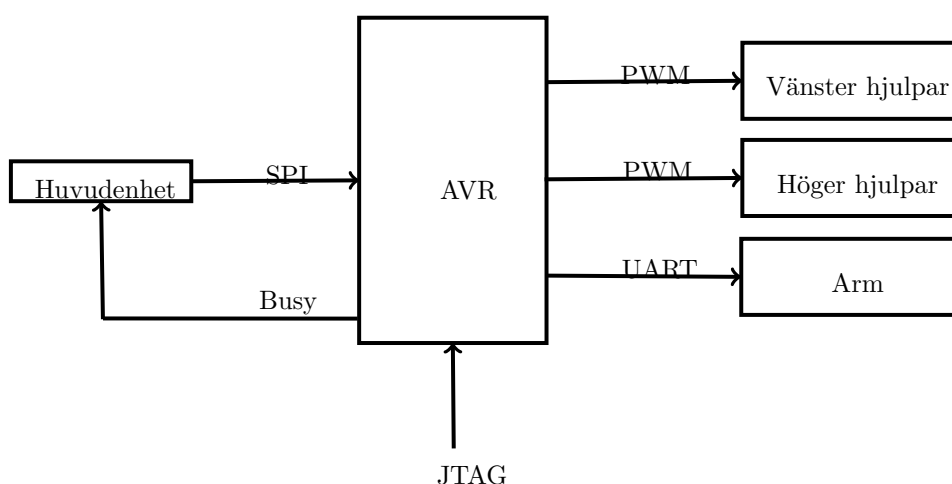
6.4 Komponenter

Följande komponenter är nödvändiga för konstruktion av sensorenheten.

Komponent	Tillgänglighet
En AVR av typ Atmega1284	Tillgänglig
En JTAG Ice 3	Tillgänglig
En Reflexsensormodul	Tillgänglig
Två muxar av typ MC14067B	Tillgängliga
Två avståndssensorer av typ GP2D120	Tillgängliga
En avstudsad tryckknapp	Tillgänglig

7 Styrenhet

Styrenheten har till uppgift att driva de motorer som driver hjulen och de servon som styr armen. Styrenheten består av en AVR av typ Atmega1284, två motorpar och en robotarm av typ Trossenrobotics Reactor[4]. AVR:n är ansluten till huvudenheten med en SPI-buss och en busy-pinne. Instruktioner från huvudenheten fås enligt protokoll definierat i sektion 3.3.



Figur 12 – Blockschemat över styrenheten

7.1 Framdrivning

Styrenheten innehåller två motorpar. De är anslutna med två PWM-signaler till AVRen, en till höger respektive vänster hjulpar. Från huvudenheten får styrenheten önskad hastighet på vardera hjulpar, vilka den applicerar och har så möjlighet att styra roboten framåt, bakåt, höger och vänster. Styrenheten är ansvarig för att i mjukvara implementera acceleration av motorerna så att roboten rör sig med jämna och mjuka rörelser.

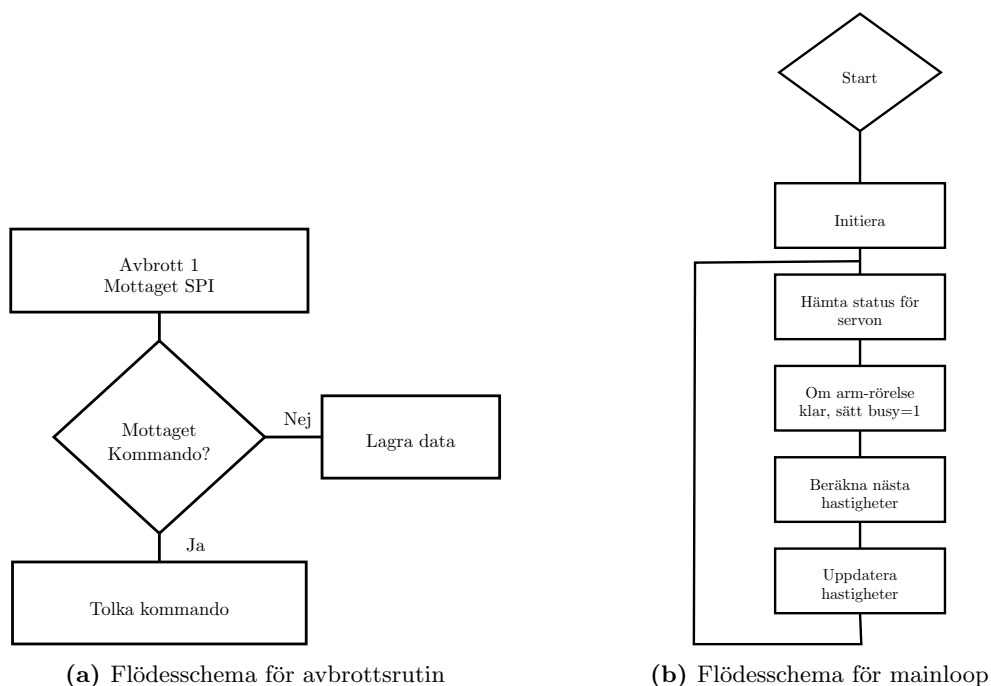
7.2 Robotarm

Robotarmen består av 7 servon av modell AX12-A[3]. De är anslutna till AVRen med en UART-buss. Dessa servon styrs genom att en målvinkel sätts (0-1023) med möjlighet att ändra hastighet, vridmoment och styra av/på. Från huvudenheten får styrenheten målvinklar för varje enskild led. Styrenheten är ansvarig för att se till att parallella servon körs synkroniserat för att inte slita sönder varandra. Styrenheten är också ansvarig för att servona accelererar och bromsar något i sina rörelser för att robotarmen skall röra sig mjukt och utan ryck. Så länge robotarmen är i rörelse sätts en busy-flagga. Den låter huvudenheten utföra köade instruktioner för robotarmen så snart som det är möjligt.

7.3 Mjukvara

Mjukvaran på styrenheten kommer i första hand att vara avbrottsstyrd. Figur 13a illustrerar programflödet för avbrott. Vid instruktioner för att uppdatera målposition eller hastighet kommer enheten lägga den i en kö. När enheten ombes utföra alla instruktioner kommer den gå igenom kön och uppdatera målpositioner och hastigheter på servon och motorer enligt tidigare inkomna instruktioner.

När AVRen inte är upptagen med avbrott körs en loop där alla servons hastighet och position läses av, jämförs med dess målposition och uppdateras med en ny hastighet. Figur 13b illustrerar programflödet. Detta ger robotarmen jämna och mjuka rörelser.



Figur 13 – Styrenhetens mjukvara

7.4 Komponenter

Följande komponenter är nödvändiga för konstruktion av styrenheten.

Komponent	Tillgänglighet
Två hjulpar	Tillgängliga
En AVR av typ Atmega1284	Tillgänglig
En Trossenrobotics Reactor	Tillgänglig
En JTAG Ice 3	Tillgänglig
En avstudsad tryckknapp	Tillgänglig

8 Testning av system

Tre typer av test kommer att utföras för att försäkra oss om att systemet fungerar som planerat.

8.1 Test av delsystem

Varje delsystem kommer att testas för sig, för att se till att de utför de funktioner vi önskar. Styrmodulen och sensormodulen kommer att testas med hjälp av en JTAG Ice 3. Huvudmodulen har liksom PC-enheten ett ordentligt operativsystem installerat, vilket gör att vi kan använda en skärm eller ssh och köra testprogram direkt på enheten.

8.2 Test av kommunikationskanaler

Varje delsystem kommunicerar med minst en annan enhet. Test utförs för kommunikationen mellan varje par av delsystem.

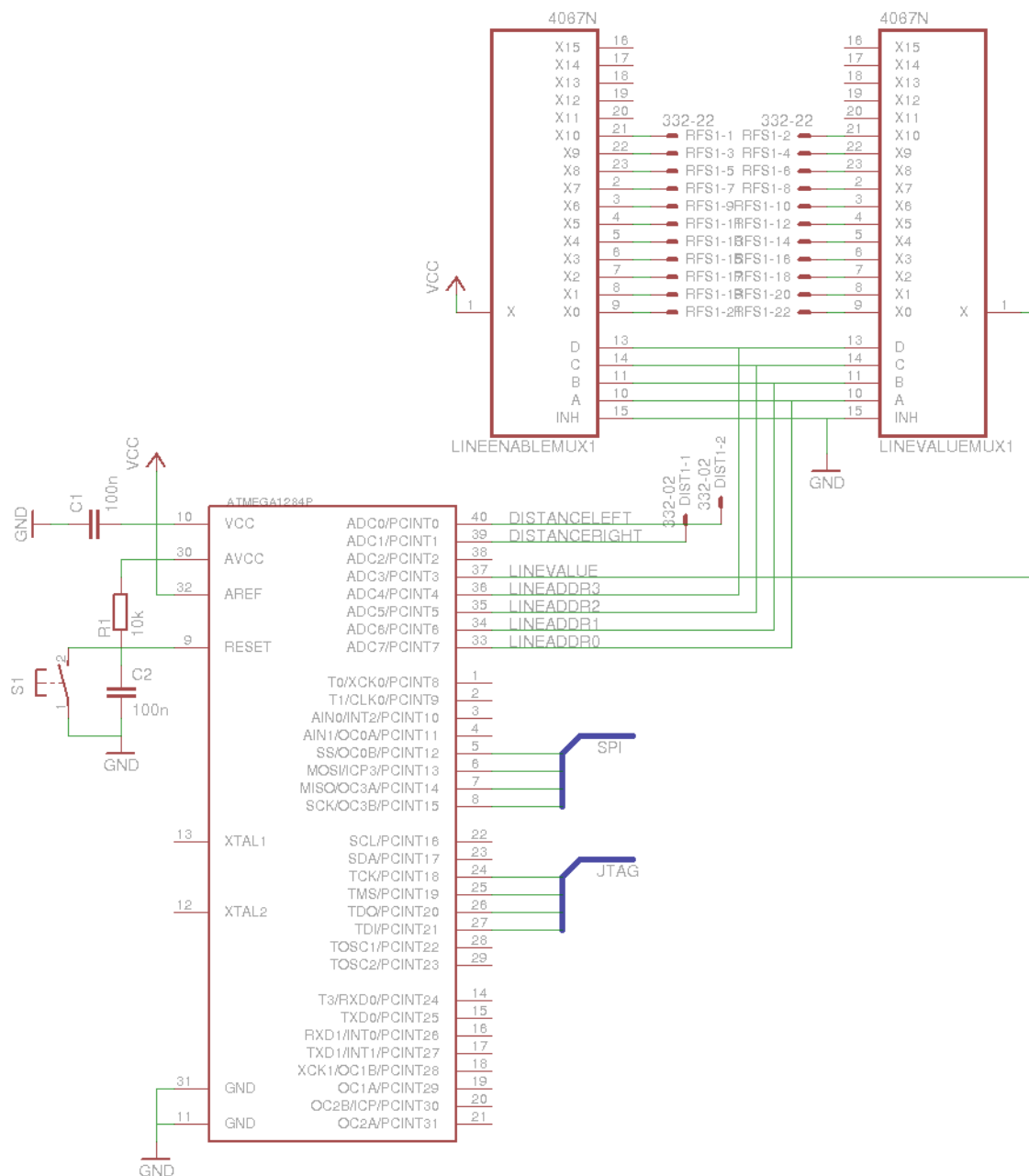
8.3 Integrationstest

Integrationstest kommer att utföras i två steg. Först utförs ett test av samtliga system exkluderat PC-enheten. Det vill säga styra motorer, servon och läsa sensordata från huvudmodulen. I sista steget utförs ett fullt integrationstest då roboten styrs från en extern PC.

Referenser

- [1] Johansson, Peter, handledare i kursen TSEA29, Linköping, 2014-10-08 (Möte).
- [2] <http://www.electrokit.com/nivaomvandlare-8-kanaler.50717>, information hämtad 2014-10-13
- [3] http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm, information hämtad 2014-10-24
- [4] <http://www.trossenrobotics.com/p/phantomx-ax-12-reactor-robot-arm.aspx>, information hämtad 2014-10-24

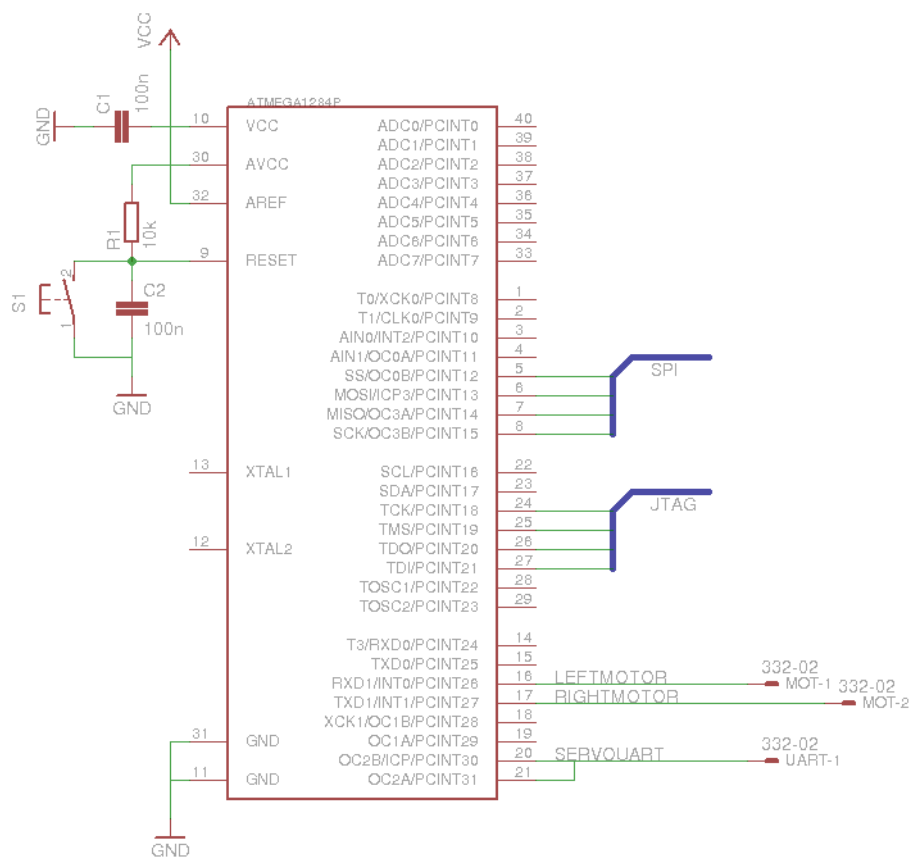
Bilaga A Kopplingsschema över sensorenhet



Figur 14 – Kopplingsschema över sensorenhet

Signalerna RFS går till reflexsensormodulen. Jämna nummer är utsignaler från fototransistorerna. Udda är enablesignaler till lysdioderna. DISTANCELEFT och DISTANCERIGHT är utsignalerna från avståndssensorerna. LINEADDR-signalerna kontrollerar vilken lysdiod i reflexsensormodulen som är aktiv. SPI-bussen ansluts till huvudenheten. JTAG-bussen kopplas till JTAG ICE-3.

Bilaga B Kopplingsschema över styrenhet



Figur 15 – Kopplingsschema över styrenhet

LEFTMOTOR och RIGHTMOTOR kopplas till motorernas styrportar. SERVOUART kopplas till DATA-porten på armens första servo. SPI-bussen ansluts till huvudenheten. JTAG-bussen kopplas till JTAG ICE-3.