SYSTEMSKISS

Version 0.1

Status

Granskad	Jennifer Santos	2023-09-21
Godkänd		

PROJEKTIDENTITET

G06, HT2023, charty
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Hugo Nilsson	kundansvarig (KUN)	073-429 33 26	hugni385@student.liu.se
Jennifer Santos	dokumentansvarig (DOK)	070-863 59 08	jensa682@student.liu.se
Edvard Wetind	designansvarig (DES)	072-717 51 15	edvwe024@student.liu.se
Elin Rydebrink	testansvarig (TST)	070-315 69 83	eliry213@student.liu.se
Elliot Norlander	kvalitetssamordnare (QS)	070-719 90 17	ellno907@student.liu.se
Jacob Sjölin	implementationsansvarig (IMP)	070-861 16 57	jacsj573@student.liu.se
Elliot Norlander	projektledare (PL)	070-719 90 17	ellno907@student.liu.se

Kursansvarig: Anders Nilsson, anders.p.nilsson@liu.se

Kund: Anders Nilsson, anders.p.nilsson@liu.se
Handledare: Petter Källström, petter.kallstrom@liu.se

Innehåll

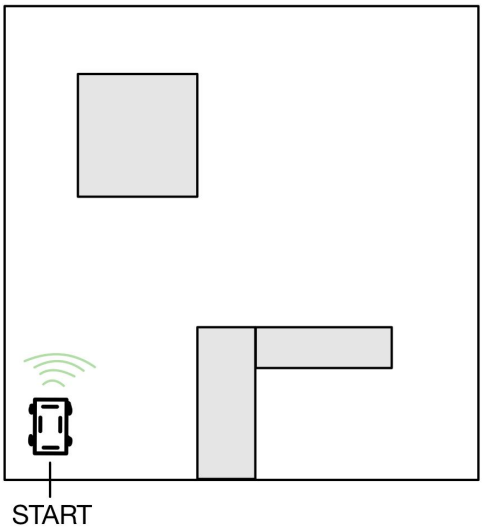
Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
v0.1	21 sept	Första utkastet	JSa, EN, HN, EW, JSj	JSa

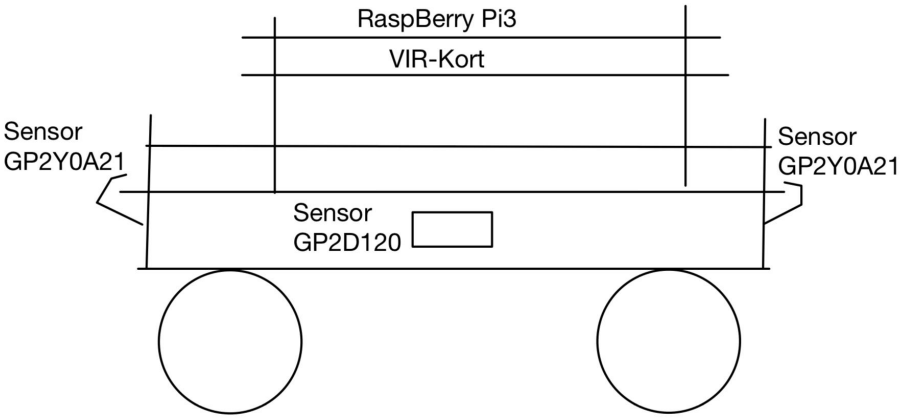
1. Inledning

I kursen TSEA29 - Konstruktion med mikrodatorer (KMM) ingår det att skapa en autonom robot som utför ett enklare uppdrag. I detta fall handlar denna uppgift om att navigera och kartlägga en miljö uppbyggd i kartong. Som en del av förberedelserna av detta projekt skapades denna systemskiss. Systemskissen innehåller en översiktlig beskrivning av hur roboten ska konstrueras, dess olika moduler samt ett översiktligt blockschema av produkten.

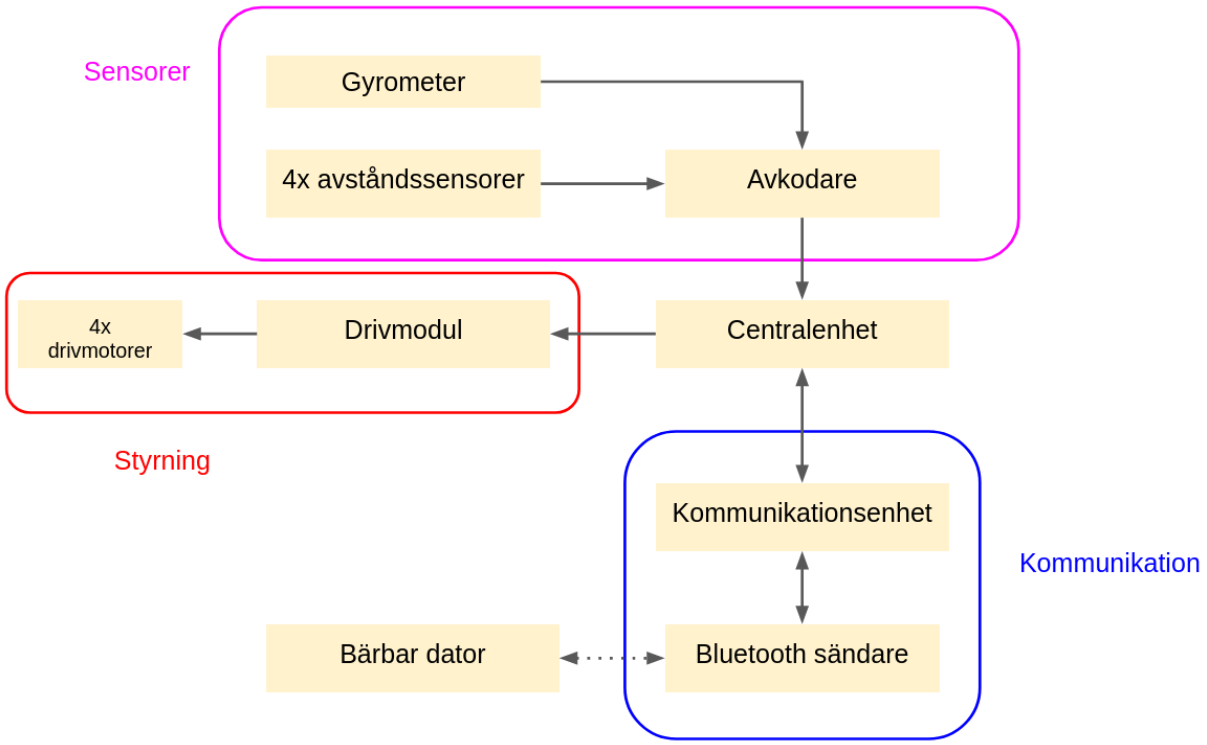
2. Systemöversikt



Figur 1, skiss över konstruktionen i miljö.



Figur 2, övergripande skiss av konstruktionen.



Figur 3, systemöversikt.

2.1 Hårdvara

Konstruktionen består av ett flertal olika hårdvarukomponenter. Bland annat kommer ett flertal olika processorer, sensorer och sändare användas. I detta avsnitt beskrivs dessa mer ingående.

2.1.1 Sumo-chassi

Roboten kommer byggas på ett så kallat sumo-chassi (SC). Med denna typ av chassi monteras batterier och elektronik ovanpå, medan sensorer kan fästas på olika delar av chassit. SC drivs av två motorer, en för varje sida av de fyra hjulen. Motorerna styrs av två signaler, Dir och PWM. Dir styr åt vilket håll motorn ska gå och PWM styr i vilken hastighet motorn ska snurra. [1]

2.1.2 Avståndssensorer

För att samla in data kring den miljö roboten befinner sig i behöver sensorer användas. Tre stycken sensorer av typen GP2D120 kommer att användas. Denna sensortyp har ett arbetsområde på 4-10 centimeter. Det vill säga att den kan upptäcka hinder och väggar på avstånd mellan 4 och 10 centimeter.

Genom att montera en sensor på sidorna samt en på bakänden av SC kan cirka 270° av robotens omgivning skannas av. Genom att sedan montera en sensor av typen GP2Y0A21 på fronten av SC kan 360° av robotens omgivning skannas av. GP2Y0A21 har ett arbetsområde på 10 - 80 centimeter, detta är fördelaktigt att använda i fronten av SC då roboten kan skanna av stora delar av banan i ett svep. [2][3]

2.1.3 Raspberry Pi

Raspberry Pi 3B+ är en 64-bitars ARM-baserad single board computer med en processorklocka på 1.2 GHz. Denna processor kommer hantera stora delar av robotens beräkningar samt bluetooth-kommunikationen med den externa bärbara PC:n. Flera moduler kommer att sitta på Raspberry Pi:n däribland centralenheten, bluetooth, avkodare av data från sensorerna samt stora delar av kommunikationen mellan de olika modulerna. Även om alla olika komponenter kommer att finnas på Raspberry Pi-kortet,bevaras fortfarande modulariteten. Det vill säga att om man till exempel vill testa bluetoothen kan man göra det utan att behöva bry sig om centralenheten.

2.1.4 Processorer

För att reglera styrning, riktning och hastighet på hjulen kommer två AVR-baserade processorer av typen ATmega1284 att användas. Det är också dessa processer som tar emot information som sensorerna samlar in. Den ena processorn kommer styra hjulen via en A/D omvandlare. Den andra processorn kommer att vara en kombination av centralenheten, och lyssna på sensorerna. Denna kommer att kommunicera med våran Raspberry Pi, lyssna på sensorerna och skicka information.

2.1.5 Bluetooth

Med hjälp av den inbyggda bluetoothmodulen i Raspberry Pi 3 kommer information kunna kommuniceras trådlöst mellan roboten och datorn. För detta krävs ingen hårdvarumässig utökning.

2.1.6 Virkort

För att koppla samman de olika komponenterna kommer virkort att användas. Virkort är en slags byggbräda för elektroniska komponenter. Komponenterna kopplas in på ovansidan, och kablar dras på undersidan.

2.1.7 Gyrometer

Enligt vår *banspecifikation* [5] kommer roboten att köra in i återvändsgränder. Detta kräver att roboten vänder på platsen. En gyrometer, som mäter hur många grader roboten har roterat, kommer att användas för att underlätta. Vi kommer använda oss av MLX90609. Denna är en enkel gyrometer som endast mäter vridningen i ett plan, dock kommer vi inte behöva någon mer avancerad sensor än denna.

2.2 Delsystem

Roboten består av tre större delsystem. Dessa delsystem hanterar styrning, sensorinformatik samt kommunikation.

Delsystemet för styrning kommer ta in ett givet körkommando från centralenheten. Utifrån detta kommando kommer den designerade processorn att behandla kommandot och skicka de nödvändiga signalerna till motorerna.

Delsystemet för sensorinformatik kommer använda avståndssensorer för att beräkna avståndet mellan roboten och olika objekt på kartan. En avkodare kommer sedan behandla informationen och skicka vidare den till centralenheten.

Delsystemet för kommunikation har i uppgift att kontinuerligt förse en extern enhet med information från centralenheten samt kunna mottaga kommandon som skickas från den externa enheten. För att utföra detta kommer en typ av bluetooth-protokoll att användas.

2.3 Centralenhet

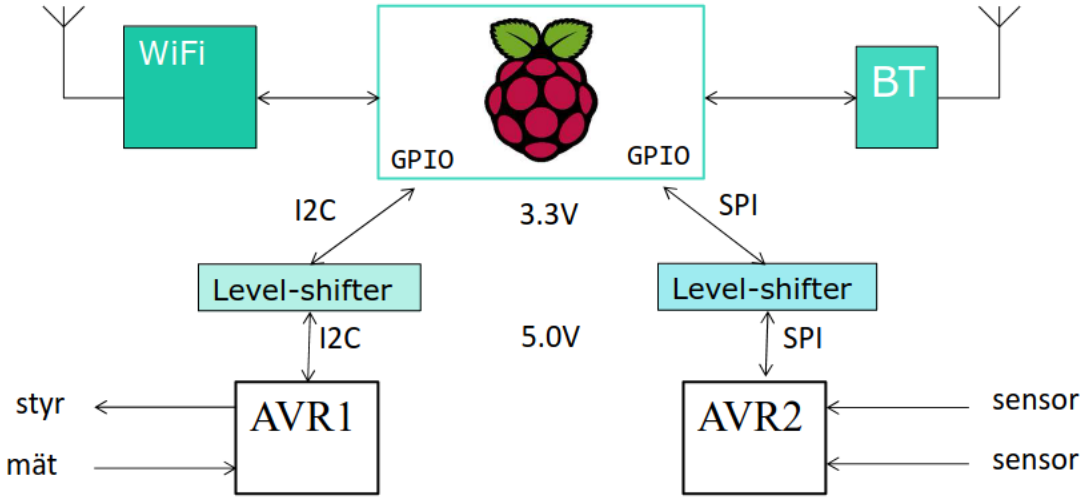
Centralenheten fungerar som en knutpunkt mellan de olika delsystemen. Det är dessutom bara centralenheten som har förmågan att skicka och ta emot information trådlöst. Centralenheten måste alltså kunna hantera en stor mängd information från flera olika källor, för att sedan skicka vidare denna på ett korrekt vis. Denna enhet kommer fysiskt vara belägen på vår Raspberry Pi 3.

2.4 Informationsflödet mellan system

Informationsflödet mellan de olika modulerna kommer att ske trådbundet då det ger en snabb och säker koppling. Vi kommer också använda oss av en trådlös bindning mellan vår centralenhet och externa dator där vårt visuella gränssnitt ligger. Kommunikationen mellan de olika modulerna kommer att vara lite olika.

Mellan sensormodulen och centralenheten skickas informationen som SPI signaler, då dessa innehåller mycket information som kommer att kunna uppdateras kontinuerligt. Därför använder vi SPI signaler för att kunna skicka så mycket information som möjligt som externa datorn sedan kommer kunna göra om till en kartbild.

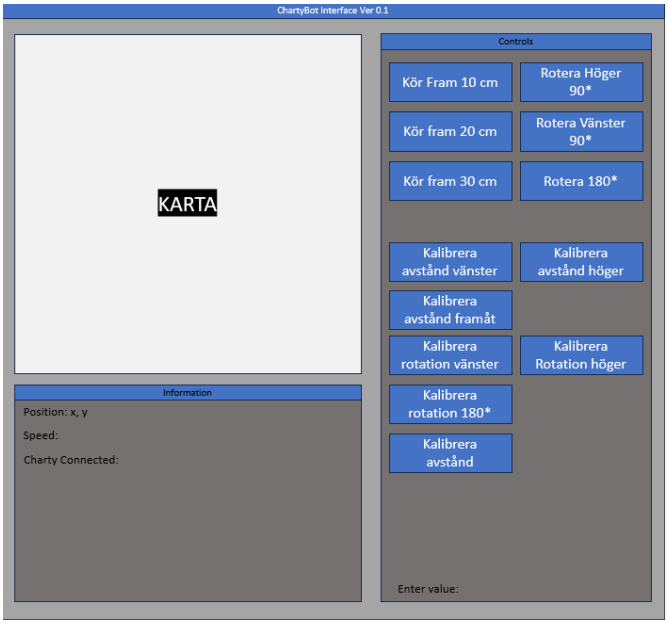
Mellan centralenheten och styrningen kommer en I2C signal att skickas. Detta då centralenheten är en master medans styrningsprocessorn kommer att vara dennes slave. Vi måste inte heller skicka lika mycket information, vilket gör detta enklare att implementera och vi behöver inte använda UART protokollet.



Figur 4, informationsflödet mellan systemen.

2.5 Gränssnitt

Kommunikation mellan roboten och användaren sker genom ett gränssnitt på användarens dator. Själva informationsutbytet mellan gränssnittet och roboten kommer ske över bluetooth. Gränssnittet skickar kommando utefter användarens önskningar, till Raspberry Pi-enheten som sedan sköter kommunikationen mellan styrenheten och sensorerna. Via gränssnittet är det möjligt att styra roboten manuellt genom att köra framåt ett önskat avstånd och svänga vänster eller höger. Utifrån informationen som avståndssensorerna upptar kommer även en mycket enkel karta ritas upp i fönstret. Klienten tillåter manuell kalibrering av sensorerna och styrenheterna för att försäkra att mätdatan blir korrekt. Dessutom går det att se aktuell mätdata som skickas tillbaka från roboten exempelvis: position, avstånd till väggar, hastighet.



Figur 5, ChartyBot interface.

2.6 Modularitet

Den färdiga produkten kommer att vara en modulär konstruktion. Detta innebär i praktiken att en del av roboten, exempelvis en sensor, kan tas bort och ersättas med en identisk enhet och fortsätta fungera utan ytterligare arbetsbörda. Modularitet är en viktig egenskap för konstruktionen då det innebär att en defekt komponent enkelt kan bytas ut. Detta försäkrar användarvänlighet vid problem, samt en längre livstid. Utbytbara enheter i den slutgiltiga produkten är: AVR1, AVR2, alla typer av sensorer och styrenheter.

2.6.1 Felsökning

Modularitet förenklar felsökningsarbetet då det blir lättare att avgöra vilken modul som är felaktig. Detta är viktigt då felsökning på hela konstruktionen kommer att vara för komplex. Det är därför avgörande för felsökningen att kunna isolera felen.

2.6.2 Testning

Något som är bra med modularitet gällande testning är att enskilda delars funktionalitet kan testas separat, vilket innebär att eventuella misstag blir mindre omständiga att åtgärda. Detta gör att flera delar kan utvecklas parallellt, då man kan testa enskilda moduler var för sig.

2.6.3 Uppgradering

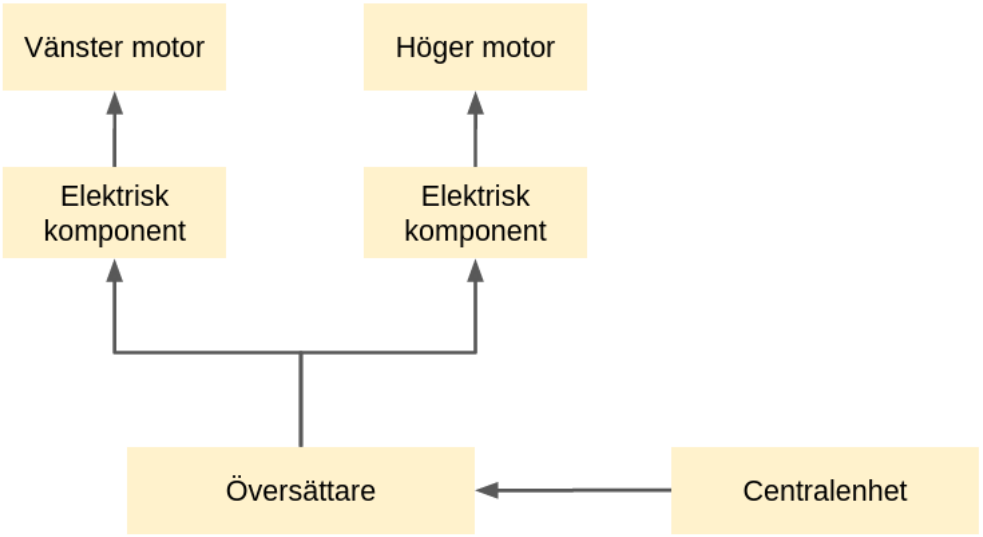
Konstruktionen är i princip helt modulär, vilket innebär att eventuell uppgradering kommer att ske på ett enklare sätt. Detta kräver att den nya enheten är kompatibel, har samma funktionalitet och kommunikationsprotokoll. Skulle detta inte vara fallet måste man bygga om konstruktionen vid uppdatering.

3. Delsystem 1 - Styrning

För att roboten ska kunna köra måste den på något sätt kunna styras. Detta kommer att ske i en kombination av olika moduler som bestämmer vart roboten ska köra.

Hastigheten på roboten bestäms av DIR och PWM signaler. DIR står för “*direction*” och PWM står för “*pulse width modulation*”. Från vår externa bärbara dator skickas signaler som bestämmer vart roboten ska köra antingen manuellt eller via en algoritm. Detta skickas till kommunikationsmodulen som skickar informationen till styrningsmodulen. Styrningsmodulen översätter, med hjälp av tidsreglering, vilka signaler som skickas till motorerna.

3.1 Blockschema



Figur 6, blockschema över styrningen.

3.2 Reglering

Från centralenheten skickas information till drivmotorerna. Denna information kan till exempel handla om rotationshastighet och rotationsriktning. Hur roboten ska köra regleras i den externa bärbara datorn, då det är tyngre beräkningar. Det kommer att finnas två sätt att sköta regleringen. Roboten ska kunna styras av användaren från datorn, detta innebär då att motorerna på roboten lyssnar på användarens instruktioner direkt. I det andra fallet ska roboten räkna ut själv vart den ska köra.

Det kommer att vara lite olika regler som används beroende på ifall roboten körs i en korridor eller letar runt på en större yta. För att roboten ska kunna köra stabilt i en korridor behöver roboten beräkna hur nära den är väggarna, samt ifall den tycker att den inte är i mitten av korridoren. Detta kan göras enligt formel 1. Där e[n] är skillnaden av avståndet till höger och avståndet till vänster. K_P och K_D är skalärer som man ska kunna ändra i realtid medan roboten kör. Detta då vi kan behöva ändra hur aggressivt roboten svänger när den har lämnat mitten av korridoren. När roboten inte är i en korridor kommer formel 1 att ge felaktiga värden. Detta innebär att utanför korridorerna kommer inte formeln att användas.

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

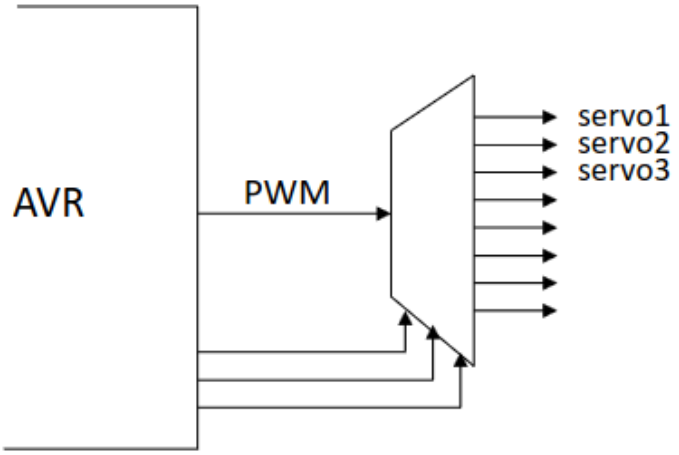
Formel 1, beräkning av avstånd till korridorens väggar.

3.3 Tidsreglering

Drivservot styrs av DIR och PWM signalerna. Dessa skickas från en översättare som översätter hur man vill att bilen ska styras. För att upprätthålla en konstant hastighet kommer signalerna att skickas var 20:e millisekund. Detta kommer att kontrolleras av en timer i AVR-processor.

DIR signalen bestämmer vilket håll man vill att drivservot ska åka åt. Detta kommer att vara en puls mellan 1-2 millisekunder där 1ms betyder framåt, 1,5ms betyder neutral och 2ms betyder bakåt.

PWM signalen bestämmer hastigheten som servon ska driva hjulet på. Tiden som signalen är hög bestämmer hur stor effekt drivservon ska arbeta på. Detta sker också i intervaller på 20ms. Till exempel om signalen är hög i 5ms sedan låg i 15ms kommer servot att köra på 25% av maximal effekt.

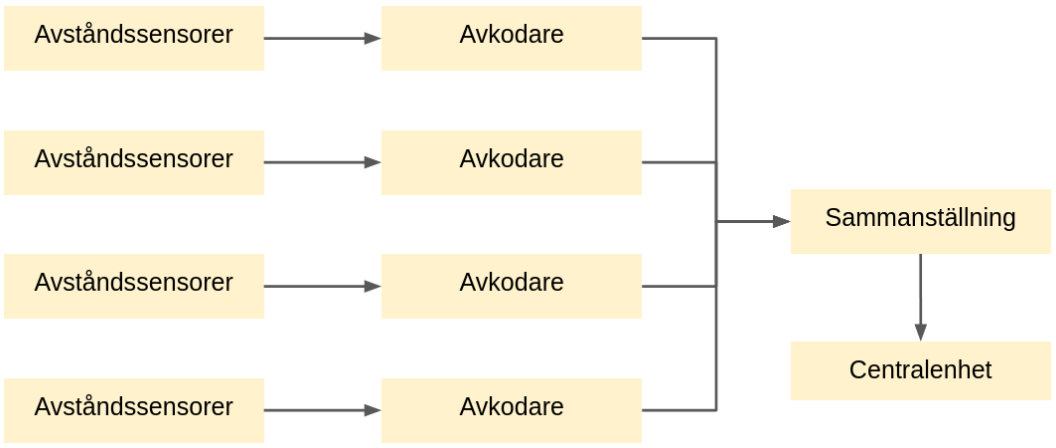


Figur 7, kommunikation mellan AVR och servo.

4. Delsystem 2 - Sensorer

För att kunna ta reda på omgivningens utseende används ett delsystem för sensorinformatik. Centralenheten ska löpande få en sammanställning av avkodad avståndsinformation om objekt i direkt närhet.

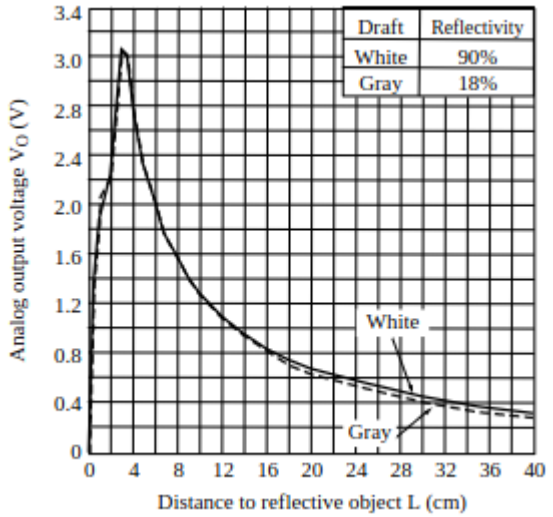
4.1 Blockschema



Figur 8, Blockschema över sensorer.

4.2 Avkodning

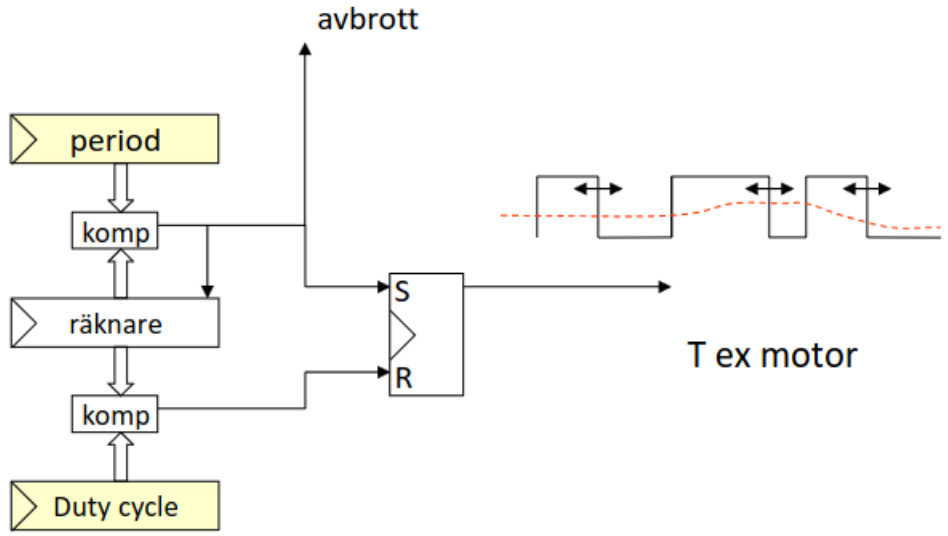
Avståndsensorerna är 4 stycken seriellt kopplade slaves som skickar SPI signaler till master som i detta fall är en Atmel CPU. Denna signal måste därefter skickas till Raspberry Pi 3 för avkodning. Avståndet som avläses från sensorerna representeras av ett spänningsvärde och därför används avkodningsfunktioner. Då två olika sorters avståndssensorer kommer att användas behövs två separata avkodningsfunktioner. I figur [9] visas spänningen som skickas från GP2D120 då den mäter. Vi kommer använda oss utav vita väggar enligt *banspecifikationen* [5].



Figur 9, spänningsvärden från GP2D120.

För att kunna ha nytta av signalen måste vi göra om den från en analog till en digital signal. Detta kan göras via en A/D omvandlare liksom den i figur [10].

En A/D omvandlare fungerar så att den kollar på hur hög spänningen är och omvandlar till olika långa intervaller av 1:or och 0:or. Det vill säga, desto högre spänningen är desto fler 1:or. Detta gör det möjligt att använda analoga signaler på vår externa dator för att kunna kartlägga området.

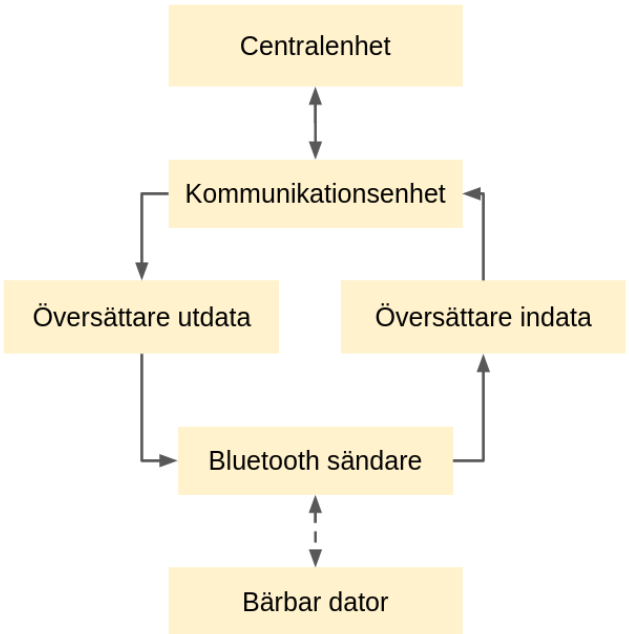


Figur 10, A/D omvandlare.

5. Delsystem 3 - Kommunikation

Roboten ska på ett användarvänligt sätt kunna kommunicera med användaren. Detta kommer ske på den externa bärbara datorn via ett grafiskt gränssnitt. För att köra roboten kommer det krävas trådlös kommunikation, via bluetooth.

5.1 Blockschema



Figur 11, blockschema över kommunikation.

5.2 Bluetooth

Raspberry Pi har en inbyggd Bluetooth 4.2 BLE läsare som gör det möjligt att ansluta bluetoothenheter. Ett externt paket “Bluez” kommer att installeras, vilket är ett Linux Bluetooth system som låter Raspberry Pi kommunicera med Bluetooth classic och Bluetooth low energy enheter. Bluetooth tillämpningen kommer att skrivas i C.

6. Referenser

Nedan visas de referenser vi utgått ifrån.

Elektroniska källor

[1] https://www.da.isy.liu.se/vanheden/pdf/terminator_prel.pdf
[2] GP2D120
[3] GP2Y0A21

[4] Kravspecifikation kommer då v1.0 är klar

[5] Designspecifikation kommer då v1.0 är klar