DESIGNSPECIFIKATION

Version 1.0

Status

Granskad	Elliot Norlander	2023-10-19
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: Theodor Lindberg, theodor.lindberg@liu.se

Innehåll

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
V0.1	5/10-2023	Första utkast	JSA, EN, HN, EW, JSJ, ER	JSA
V0.2	12/10-2023	Andra utkast	JSA, EN, EW	JSA
V1.0	19/10-2023	Finalisering	EN, EW, JSA	EN

1. Inledning

Målet med projektet är att skapa en autonom robot som kan navigera, kartlägga och rita upp ett från början okänt område. Detta dokument innehåller en ingående beskrivning av systemets konstruktion och funktionalitet.

För att möjliggöra en konstruktion av denna typ krävs en hel del hårdvara. Bland annat kommer olika mikrokontrollers, drivmotorer, virkort samt sensorer att användas. För en ingående beskrivning av funktionalitet och samspel mellan dessa, hänvisas läsaren till de specifika tillhörande avsnitten.

2. Styrning

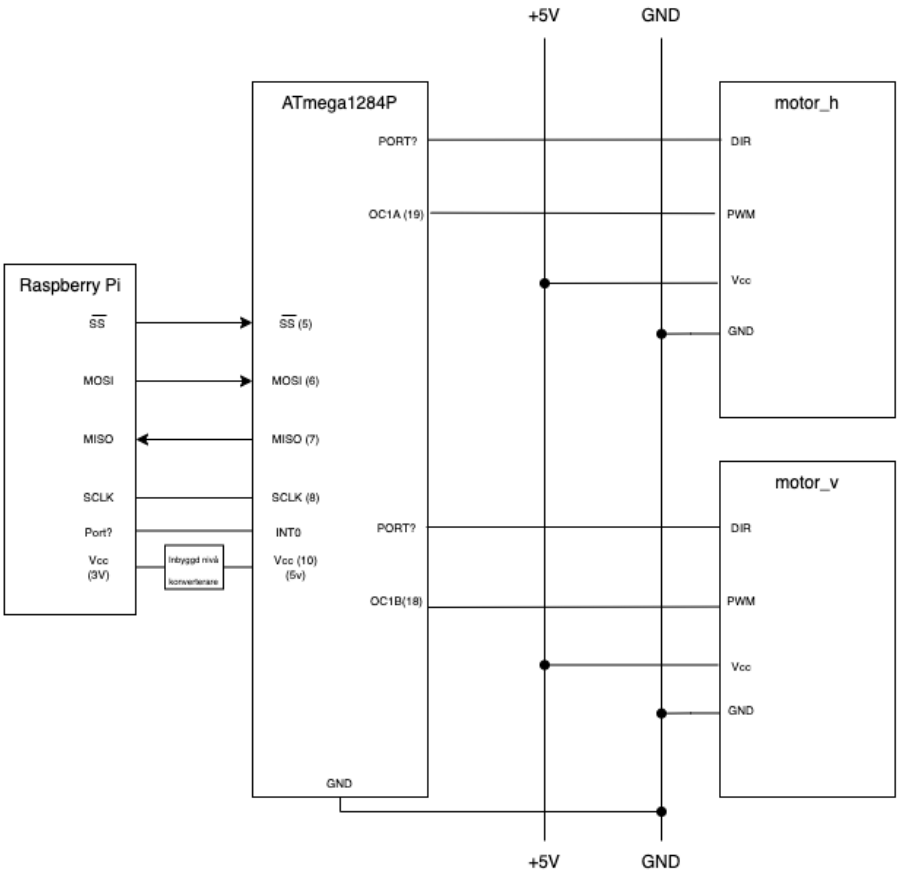
2.1 Hårdvara

Styrningsenheten består i huvudsak av fem komponenter, en mikrokontroller och fyra motorer. Motorerna styrs sidvis och parvis i form av två signaler, DIR och PWM. DIR styr motorns rotationsriktning, och kopplas direkt till mikrokontrollern. PWM styr motorns rotationshastighet och är en pulsbreddsmodulerad signal. Med hjälp av mikrokontrollern kan en pulssignal med eftertraktad pulsbredd genereras. För höger och vänster motor kommer denna signal skickas från porten “OC1A” respektive “OC1B”.

Förutom motorerna kommer mikrokontrollern även kopplas till en Raspberry Pi. Här används signalerna SS, MISO, MOSI och SCLK. Dessa signaler delas mellan mikrokontrollern för styrning och mikrokontrollern för sensorhanteringen. Mer information om kommunikationen mellan olika delsystem finns under rubrik 5. Kommunikation mellan delsystem.

2.2 Kopplingsschema

I figur 1 finns en detaljerad bild av styrsystemet.



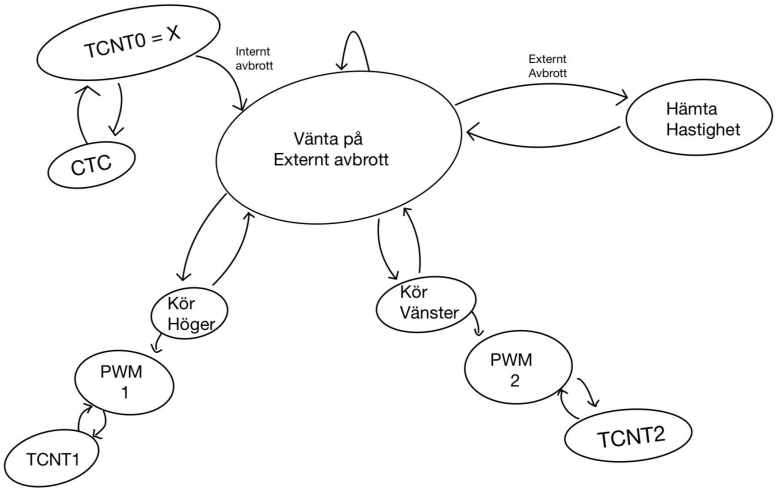
Figur 1, kopplingsschema för styrsystem.

2.3 Process

För att underlätta förståelsen kring arbetsprocessen för styrsystemet se figur 2. Systemet kommer använda en kombination av interna avbrott från en räknare och externa avbrott från Raspberry Pi. De interna avbrotten är nödvändiga då processorklockan för ATmega1284 är för hög (1 MHz) samt då processorns huvudlopp inte kan garantera konstanta tidsintervall. Alltså behöver en separat timer göra detta. För ATmega1284 används den inbyggda komponeneten “Waveform Generator”. Denna har i sig två lägen som kommer användas, Clear Timer on Compare match (CTC) och Phase Correct (PWM).

Styrsystemet kommer använda CTC för att skapa de interna avbrotten. Detta görs genom att låta den interna räknaren TCNT0 räkna till ett visst tal. När generatorn märker att det är en match mellan räknaren och det förutbestämda värdet börjar räknaren om, och ett internt avbrott skickas från generatorn i form av att CTC flaggan tänds. PWM används för att generera den pulsbreddsmodulerade signalen som styr hastigheten på motorerna. För att göra detta används två separata räknare, TCNT1 och TCNT2. Dessa får en hastighet bestämd av

Raspberry Pi och med hjälp av denna information kan två signaler, PWMX, $X\epsilon(1, 2)$, genereras med olika pulsbredd. PWMX, $X\epsilon(1, 2)$ skickas sedan till motorerna.



Figur 2, tillståndsdiagram för styrsystem.

3. Sensorer

3.1 Hårdvara

3.1.1 Avståndssensorer

Två sorters IR-avståndsmätare kommer att användas. Den ena sorten har beteckningen GP2D120 och kan mäta ut avståndet till objekt som befinner sig mellan 4 och 30 cm bort. Den andra sensorn vid namn GP2Y0A21 har arbetsområdet 10 till 80 cm. Systemet använder tre IR-avståndsmätare, två av modell GP2D120 och en av modell GP2Y0A21, som genererar 25 mätvärden per sekund. De mätvärden som är relevanta i detta fall fås som analog spänning och ligger i intervallet 0-3V. IR-mätaren kopplas ihop med en AVR, som har en A/D omvandlare. Spänningssignaler som genereras från varje avståndsmätare skickas till respektive A/D omvandlare som i sin tur översätter varje signal till ett värde på 10 bitar. Avkodning av det tal som erhållits genom A/D omvandlingen ska ingå i en sammanställning av resultaten från samtliga tre avståndsmätare, som i sin tur skickas till centralenheten.

3.1.2 Halleffektsensor

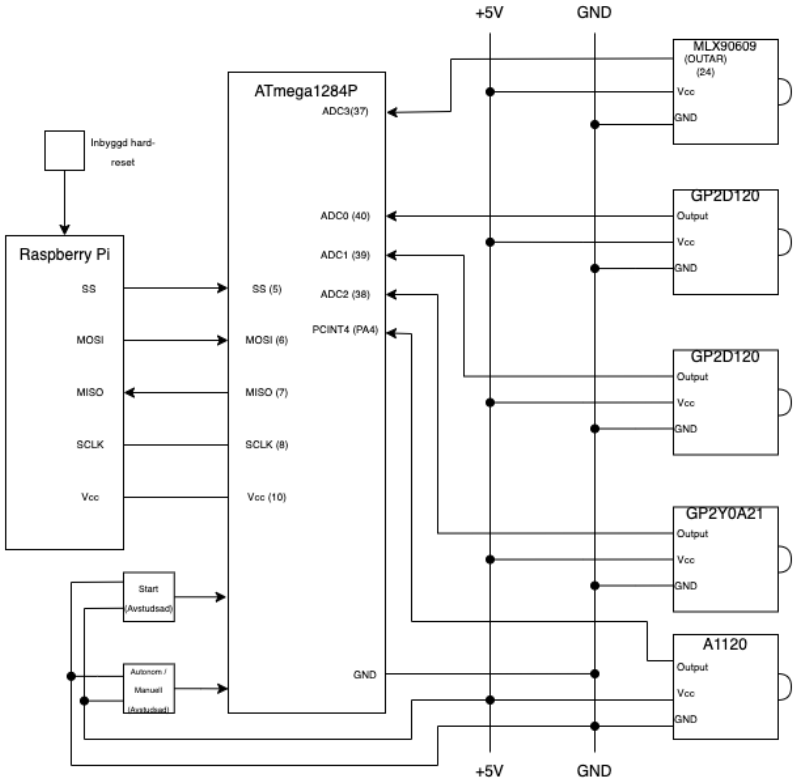
Utöver IR-sensorerna kommer en halleffektsensor användas. Syftet med den är att kunna mäta hur långt roboten kör. Detta görs genom att placera små magneter på insidan av robotens hjul. När hjulen sedan snurrar triggas sensorn när magneten i hjulet kommer tillräckligt nära sensorn.

3.1.3 Gyrosensor

Roboten kommer även att behöva göra exakta 90° svängar. För att detta ska vara möjligt måste systemet använda en gyrometer av typ MLX90609.

3.2 Kopplingsschema

I Figur 3 nedan finns ett detaljerat kopplingsschema för sensorsystemet.

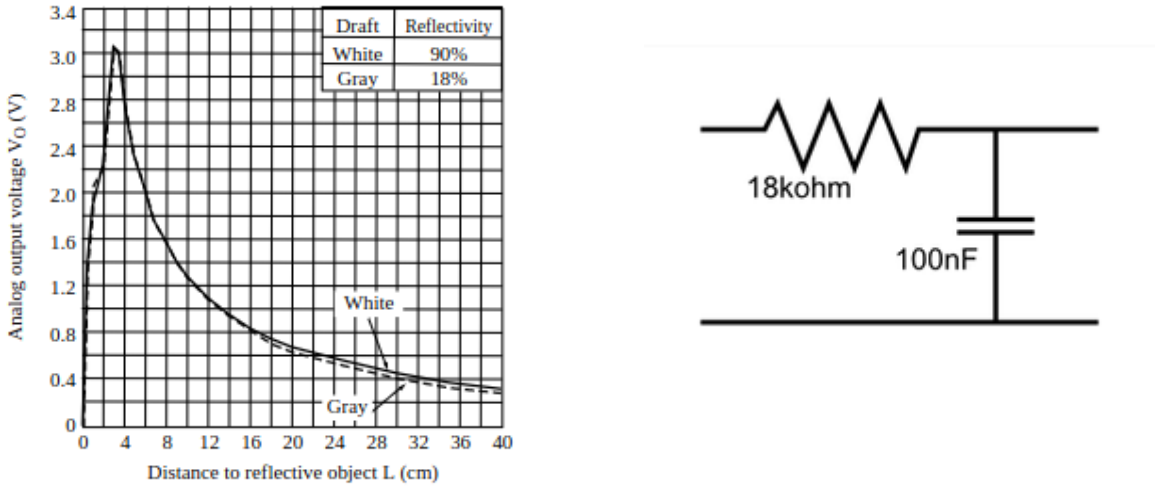


Figur 3, kopplingsschema för sensorsystemet.

3.3 Avkodning

3.3.1 Avkodning av avståndssensorer

Då A/D omvandling genomförts och ett 10 bitars tal har erhållits är det nästkommande steget att avgöra vilket avstånd som detta tal representerar. Beroende på hur hög spänningen är som skall omvandlas kommer det 10 bitars långa talet att innehålla olika många ettor där en större mängd ettor innebär en högre spänning. Då två olika sorters avståndsmätare, GP2D120 och GP2Y0A21, används kommer således tal som av samma storlek att representera olika avstånd. För att kunna översätta det 10 bitars långa talet behövs således en funktion för varje sorts avståndsmätare som översätter talet till ett avstånd. För att undvika brus lägger vi till ett lågpassfilter för varje sensor mellan sensorn och AVR-processorn, se figur 4.



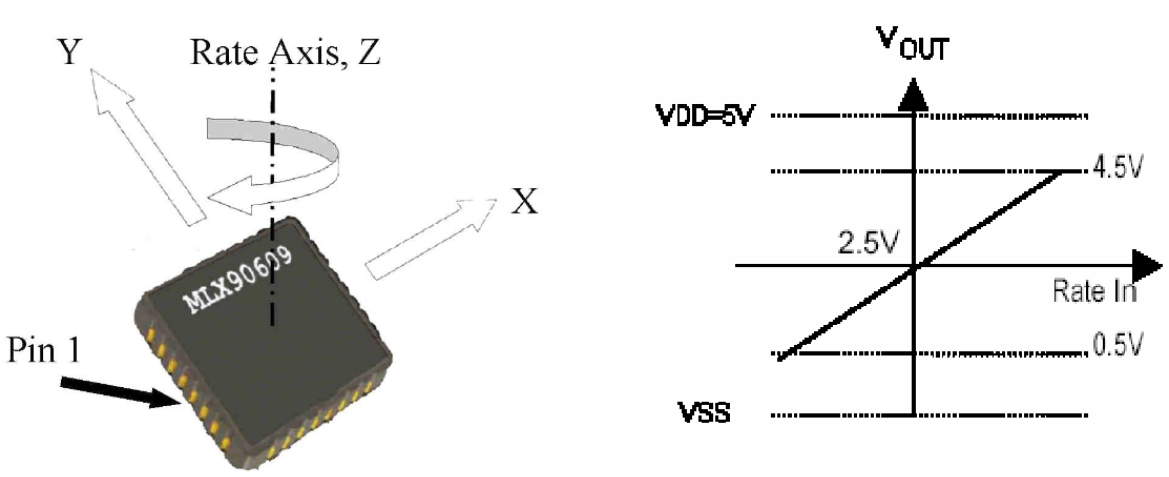
Figur 4, spänningsvärden från GP2D120 samt lämpligt LP-filter.

3.3.2 Avkodning av halleffektsensor

Halleffektsensorn A1120 ger en logiskt hög (1) respektive låg (0) signal. När sensorn detekterar en magnetisk sydpol av tillräckligt hög styrka ger sensorn utsignal 1, annars är den 0. Genom att kontinuerligt mäta antalet höga signaler, kan man veta hur snabbt och hur långt roboten kört.

3.3.3 Avkodning av gyrosensor

Gyrosensor skickar en analog signal mellan 0.5V och 4.5V till processorn där 0.5V = max rotation moturs, 2.5V = ingen rotation och 4.5V = max rotation medurs. Den analoga signalen ändras linjärt, dvs att vi enkelt kan räkna ut hur långt roboten har roterat med ekvationen $\theta = U_{in}\omega_{max}\Delta t$.



Figur 5, spänningsvärden från MLX90609

4. Trådlös Kommunikation

4.1 Bluetooth

På datorn kommer en simulerad seriell port mellan datorns bluetooth och Raspberry-Pi:n att skapas. Datorn kommer att ta emot information från roboten, som i sin tur används i gränssnittet för att simulera sensordata. Via gränssnittet kommer det vara möjligt att styra roboten manuellt, denna kommunikation kommer också att ske via bluetooth. Detta görs genom olika kommandon med tillhörande ID som kommer att användas för att skicka och ta emot data, se tabell 1 och 2. Den seriella kommunikationen som skickas från roboten till datorn sker via en avbrottsrutin, med tillhörande flagga. Eftersom informationen är seriell kommer den att ges i formen av en int: <id>.

Typ av data	**ID**
Kör fram	1
Kör bak	2
Rotera höger	3
Rotera vänster	4
Kalibrera sensor vänster	5
Kalibrera sensor höger	6
Kalibrera sensor fram	7

Tabell 1, information från dator till robot.

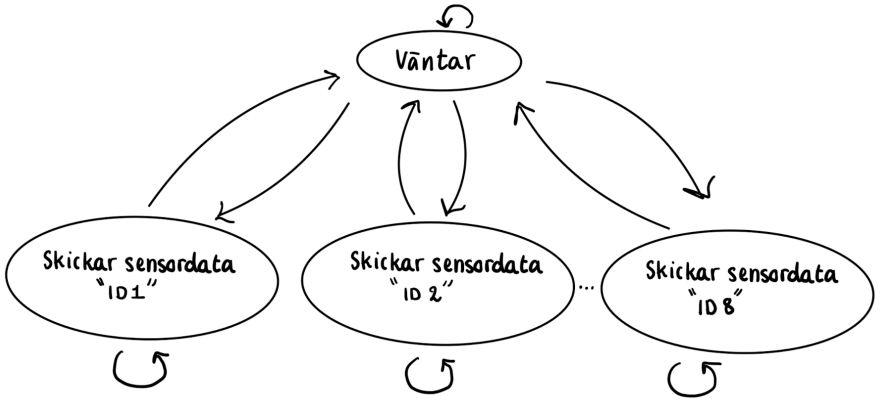
Typ av data	**ID**
Senaste kartritning från centralenhet	11

Tabell 2, information från robot till dator.

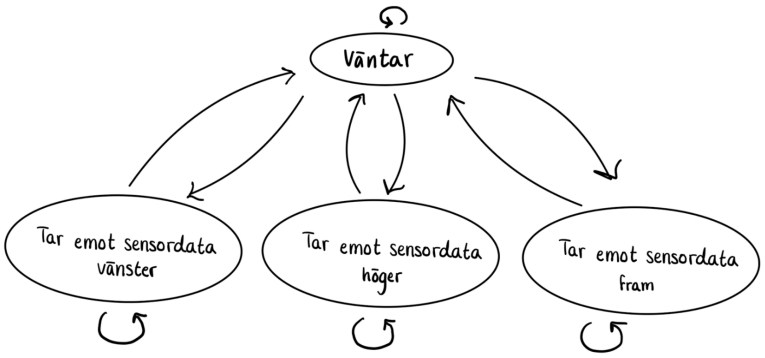
4.2 Installation

Först måste alla paket uppdateras för att se till att all Raspberry-PI:s programvara är uppdaterad. Sen installeras det externa paketet “Bluez”, som ger bättre Bluetooth support och underlättar implementationen. Därefter måste Bluetooth GUI verktyget installeras, vilket görs i Raspbian. Därefter kan man para Raspberry-Pi med önskad Bluetoothenhet.

4.3 Pseudokod



Figur 6, tillståndsdiagram över den seriella kommunikationen från datorn till roboten.



Figur 7, tillståndsdiagram över den seriella kommunikationen från roboten till datorn.

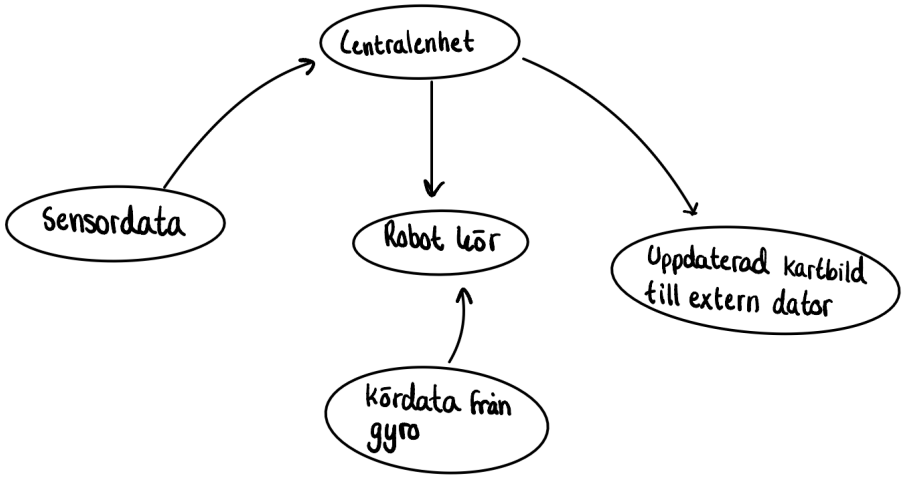
5. Kommunikation mellan delsystem

Kommunikation mellan delsystemen ska ske, bortsett från kommunikation mellan extern dator och robot, över SPI. För en beskrivning av SPI se avsnitt 5.2. Den färdiga konstruktionen består av fyra delsystem: centralenhet, sensorenheten, kommunikationsenheten och styrenheten, se figur 7. Att SPI valdes som kommunikationsprotokoll beror på dess förmåga att snabbt skicka stora mängder data, vilket kommer att vara nödvändigt på grund av sensorenhetens stora produktion av data, se tabell 3. Dessutom är det möjligt för huvudenheten att kommunicera med sensorerna, vilket är nödvändigt för eventuell kalibrering av instrumenten. Styrenheterna hade i teorin klarat sig

bra med det långsammare, enkelriktade I2C eftersom ingen information kommer att skickas tillbaka till RP. För simplicitetens skull valdes SPI även för kommunikation med styrenheten.

Namn	**Från enhet**	**Skickas till enhet**	**ID**	**Antal bitar data**
Sensordata 1 - fram	Sensorenhet	Centralenhet	1	8
Sensordata 2 - höger	Sensorenhet	Centralenhet	2	8
Sensordata 3 - vänster	Sensorenhet	Centralenhet	3	8
Sensordata 4 - rotation uppnådd	Sensorenhet	Centralenhet	4	8
Sensordata 5 - distans uppnådd	Sensorenhet	Centralenhet	5	8
Kördata	Centralenhet	Styrenhet	6	8
Uppdaterad kartbild	Centralenhet	Extern dator	7	8
Kalibrera sensorn	Centralenhet	Sensorenhet	8	8
Mät rotation	Centralenhet	Sensorenhet	9	8
Mät distans	Centralenhet	Sensorenhet	10	8
Starta huvudloop	Startknapp	Centralenhet	11	8
Töm minne och starta om	Resetknapp	Centralenhet	12	8

Tabell 3, kommunikation mellan modulerna.

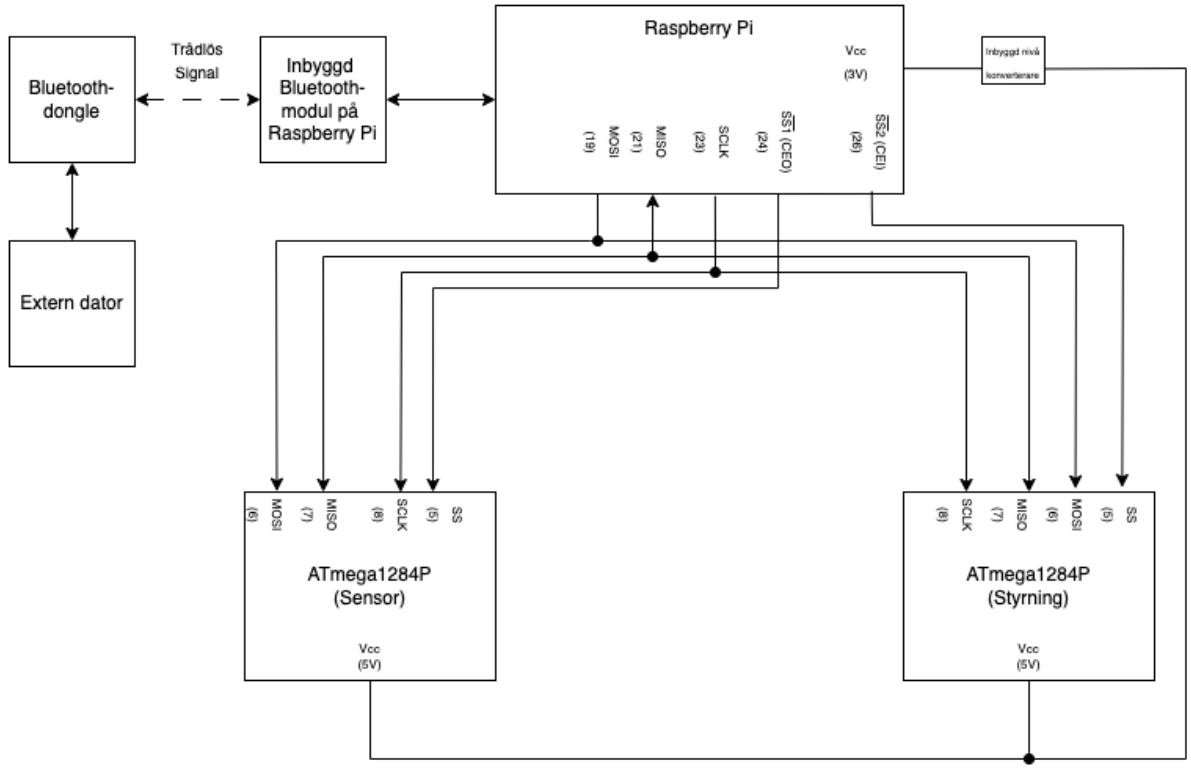


Figur 8, tillståndsdiagram över kommunikation mellan delsystem. (Endast principskiss, gyron kommer inte skicka data direkt till styrmodulen).

5.1 Kopplingsschema

Nedan, i Figur 7, syns ett kopplingsschema över hur de olika delmodulerna sitter kopplade med SPI. Numret "under" vardera signal på schemat avser den pinne på kortet som tar emot den specifika signalen. Vilken av modulerna som Raspberry Pi (RP) kommunicerar med bestäms av signalen SS1 eller SS2 (Slave Select). Notera att "SS" på de periferala modulerna är en inverterad signal, modulen väljs som aktiv slav då insignalen är låg.

RP-enheten har i schemat tre pinnar som inte är de ovan beskrivna SS1 eller SS2. Dessa är SCLK, MISO och MOSI. SCLK, Serial Clock, för att synka kommunikationen mellan master och slave. Master i konstruktionen kommer att vara RP. MISO (Master In, Slave Out) och MOSI (Master Out, Slave In) är de två signaler som gör tvåvägs-kommunikation mellan enheterna möjligt. Master skickar och tar emot en bit i taget i takt med klockan, slaven gör detsamma.



Figur 9, kopplingsschema över robotens olika moduler.

5.2 SPI

SPI betyder Serial Peripheral Interface och väljs eftersom det möjliggör extremt snabb

kommunikation mellan datorn och sensorerna. Kommunikationshastigheterna befinner sig inom intervallet 100 kHz till 50 MHz. En av protokollets svagheter är att det krävs fyra pinnar för sammankoppling, vilket kan begränsa antalet enheter med SPI per buss. SPI är även av typen full duplex, vilket innebär att information kan skickas åt båda hållen. Detta är mycket användbart vid kalibrering, då sensorerna måste kunna ta emot det nya faktiska mätvärdet. Robotens sensorer är typiska exempel på enheter som kommer att agera som slave. Det är konstruktionens Raspberry Pi som kommer att agera master för robotens alla enheter, däribland sensorerna.

6. Mjukvara

I detta avsnitt diskuteras mjukvara och till viss utsträckning hur denna mjukvara kommer att fungera.

6.1 Huvudloop

Båda enchipsprocessorerna kommer att drivas av avbrott. Det kommer finnas en huvudloop som inte har mycket funktionalitet utan väntar bara på avbrotten. Med hjälp av avbrott kommer vi kunna skicka synkroniserad information mellan de olika systemen. En fallgrop med avbrott är att ett nytt avbrott kan komma medan en annan avbrottsprocess körs, vilket kan leda till förlorad information. För att åtgärda detta, kommer inkommande avbrott att köas under tiden en annan avbrottsprocess körs. Då de processer som körs är väldigt enkla och snabba för hårdvaran, kommer dessa avbrott inte att prioriteras olika utan vi kommer använda en busy-wait kö.

För styrningsmodulen kommer det att finnas tre stycken avbrott.

- Ett internt avbrott för varje hjul, som säger till datorn att skickar körningsdata till motorerna. Detta måste ske kontinuerligt och styrs av en timer.
- Ett externt avbrott från centralenheten som inträffar då man vill ändra hastigheten på hjulen. Detta avbrott sker för att ge processorn ny information om hastigheten för hjulen.

För sensormodulen kommer det att implementeras fyra stycken avbrott.

- Ett internt avbrott för varje sensor. Avbrotten kommer att säga till när det finns ny information tillgänglig, eftersom sensorerna inte har tidsbestämd uppdatering. Dessa kommer ske då A/D omvandlingen är klar så kommer A/D omvandlaren att tända en flagga så kan man hämta informationen.
- Ett externt avbrott från centralenheten som säger till ifall man vill kalibrera om sensorerna. Denna kommer ske via SPI kommunikationen genom att ha tänt flaggan SPIE(SPI Interrupt Enable) i registret SPCR(SPI Control Register). Sedan generera avbrottet genom att sätta SPIF(SPI Interrupt Flag) till 1 i registret SPSR(SPI Status Register)

För centralenheten kommer det vara tre stycken avbrott.

- När slavarna har ny data tillgängligt kommer det att tändas en flagga hos mastern, då skapas ett avbrott i mastern för att hämta den nya datan.
- Ett externt avbrott från den externa datorn då man vill att roboten ska svänga eller ändra hastighet.

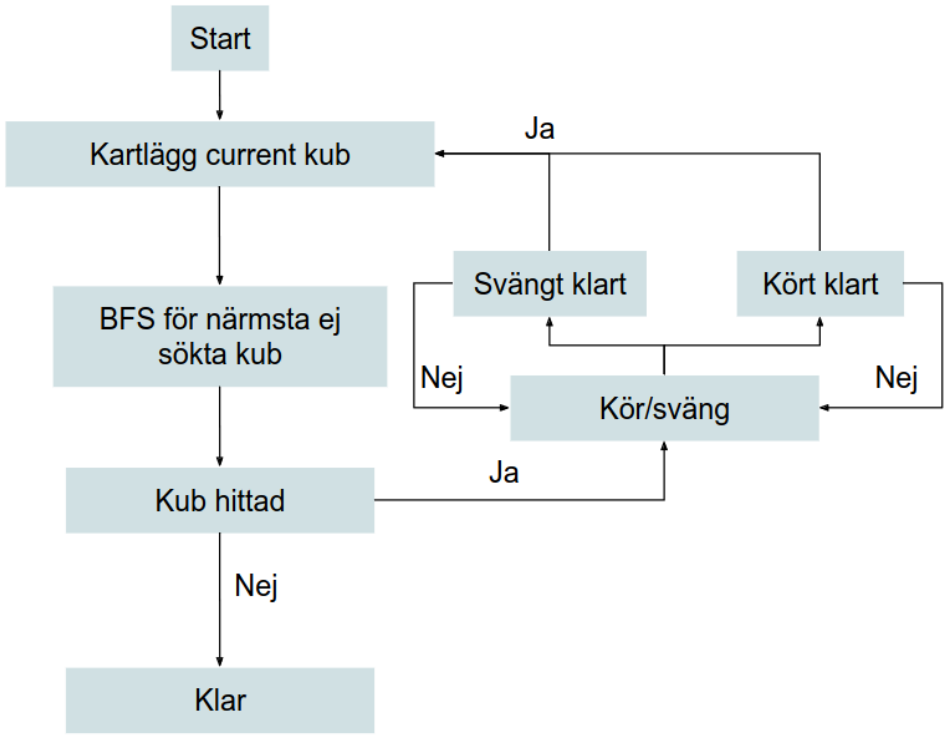
6.2 Datastrukturer

Informationen som skickas från sensorerna kommer endast att bestå av några bytes data åt gången, som sedan vidarebefordras direkt till centralenheten. Likaså kommer styrningen endast behöva ha koll på enstaka bytes av data som representerar riktning och hastighet.

Detta gör att roboten inte behöver implementera datastrukturer då mängden data är enkel nog att använda i råformat. Däremot måste datastrukturer användas på den externa datorn, eftersom information om kartan över hela modellen kommer att sparas. Detta blir dock inget problem, så programmet kommer att skrivas i Python, där önskade datastruktur t.ex. listor redan finns implementerat i språket.

6.3 Styrningsalgoritm

Styrningsalgoritmen kommer vara en enkel algoritm som letar efter nästa tomma ruta med BFS och sedan köra dit och spara den rutan som kartlaggd. Då roboten kommer starta på en ej sökt kub börjar algoritmen med att söka av kuben roboten står på. Sedan letar den upp nästa ej sökta kub för att köra dit. Detta kommer garantera att roboten kommer kunna köra på hela banan då endast väggarna kommer stoppa roboten från att kolla ifall den kan köra dit.



Figur 10, flödesschema på algoritm

7.Implementeringsstrategi

Utvecklingen kommer ske i enlighet med PARK-modellen. Det innebär att man iterativt arbetar med momenten Planera, Agera. Reflektera och Korrigera. Planera handlar om informationssamling och strukturering. Agerandet innefattar implementation och konstruktion. Reflektion handlar om anledningar till varför det (inte) fungerar och korrigering om åtgärder, uppdateringar och optimeringar.

8. Testning

8.1 SPI-testning

Avser:
Sensormodul och styrmodul.

Utförande:
Testa och se till att alla sensorer samt styrmodulen kan kommunicera med datorn via SPI. Se till att information kan skickas åt båda hållen. Se över kalibrering.

8.2 Sensortestning

Avser:
Sensormodul.

Utförande:
Testa sensorerna för att ta reda på om hårdvaran fungerar som den ska. IR-sensorerna testas för att se om rätt antal mätvärden fås samt att mätvärdena är ett spänningsvärde mellan 0-3V. Kollar också att kalibreringen av sensorerna fungerar.

8.3 Styrtestning

Avser:
Styrmodul.

Utförande:
Testa att styra och rotera roboten i olika riktningar för att se till att allt fungerar enligt beskrivningen.

8.4 Hårdvarutestning

Avser:
Sensormodul, styrmodul och chassi.

Utförande:
Kontrollera att alla sladdar är rätt kopplade och se till att inget glappar.

8.5 Kartläggningstestning

Avser:
Sensormodul och mjukvara.

Utförande:
Testa att data från sensorer skickas till datorn samt att det visas en karta på skärmen.

8.6 Mjukvarutestning

Avser:
Mjukvara.

Utförande:
Mjukvaran testas regelbundet under projektets gång.