Systemskiss

Patrik Sletmo

Version 1.0

Status

Granskad	Patrik Sletmo	2016-09-22
Godkänd	Patrik Sletmo	2016-09-28

Projektidentitet

Grupp 3, 16/HT, KarToffel

Linköpings tekniska högskola, institution

Namn	Ansvar	Telefon	E-post
Patrik Sletmo	Projektledare	070 783 57 61	patsl736@student.liu.se
Rebecca Lindblom		073 436 40 79	rebli156@student.liu.se
Matildha Sjöstedt		070 515 84 11	matsj696@student.liu.se
Sebastian Callh		073 820 46 64	sebca553@student.liu.se
Anton Dalgren		076 836 51 56	antda685@student.liu.se
Matilda Dahlström		070 636 33 52	matda715@student.liu.se

Hemsida: https://github.com/SebastianCallh/kartoffel-tsea29

Kund: Mattias Krysander, 013 - 28 2198 , matkr@isy.liu.se

Kursansvarig: Tomas Svensson, 3B 528, +46 (0)13 28 1368, tomas.svensson@liu.se

Handledare: Anders Nilsson, 3B 512, +46 (0)13 28 2635, anders.p.nilsson@liu.se

Innehållsförteckning

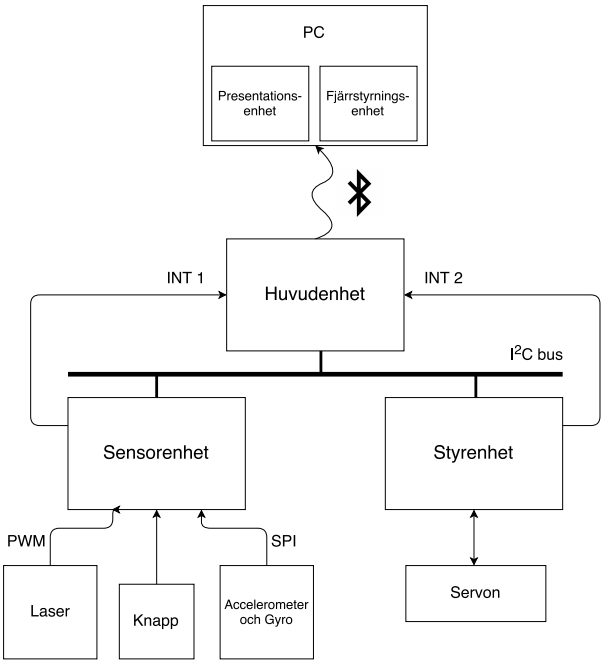
Dokumenthistorik

Version	Datum	Utförda ändringar	Utförd av	Granskad
1.0	2016-09-28	Slutgiltig version	Grupp 3	Patrik Sletmo
0.1	2016-09-22	Första utkastet	Grupp 3	Patrik Sletmo

1 Inledning

Den här systemskissen beskriver i mer teknisk detalj konstruktionen av roboten och tillhörande mjukvara. Dokumentet ska reflektera vår faktiska relisation av projektet och kan mycket väl komma att ändras allt eftersom projektet fortskrider och designbeslut behöver omvärderas.

2 Systemöversikt

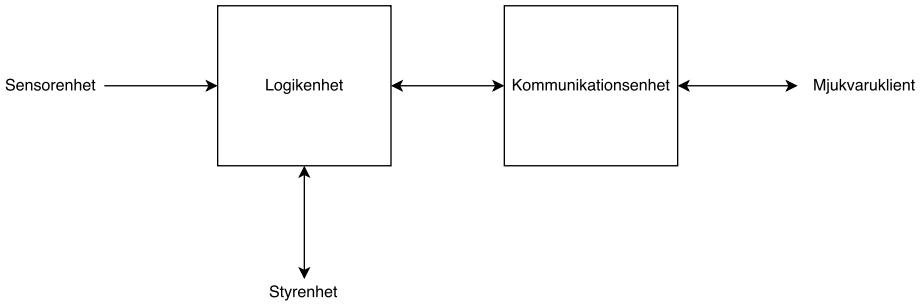


Figur 1: Översikt över systemet

Vårt system (se ovanstående figur 1) är uppdelad i fem stycken enheter (eller *moduler*), där två utav dem - presentations- och fjärrstyrningsenheten - ingår i vår mjukvaruklient och resten ingår i själva roboten. Kommunikationen mellan mjukvaruklienten och roboten sker via Bluetooth, medan den interna kommunikationen mellan robotens enheter sker via en I2C-buss. Här är huvudenheten ensam master och styr- och sensorenhet slaves. Andra protokol som vi använder i robotens interna kommunikation är PWM, SPI och UART. Sensorer som är kopplade till sensorenheten inkluderar en laser, knapp samt en accelerometer och två gyroskop. Till styrenheten kopplas servon som styr hjul och rotation av lasern.

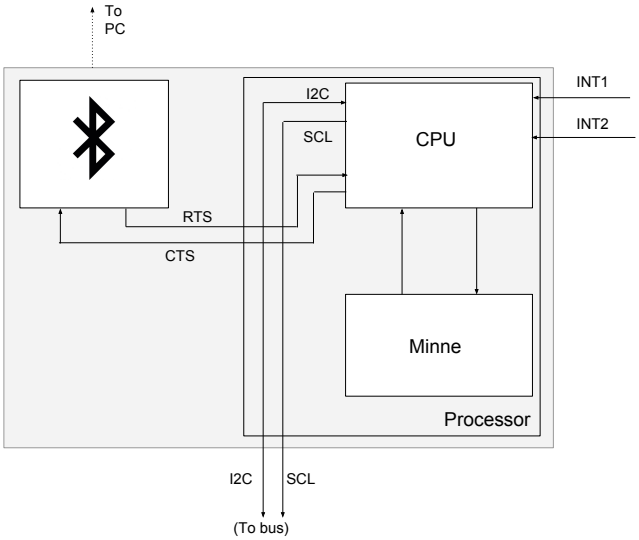
3 Huvudenhet

Huvudenheten består av två undermoduler, logikenheten och kommunikationsenheten (se figur 2). Logikenheten fattar robotens alla beslut genom att betrakta sensordata från sensorenheten och sedan beordra styrenheten att navigera på lämpligt sätt, och kommunikationsenheten hanterar all kommunikation med mjukvaruklienten.



Figur 2: Ett övergripande blockschema över huvudenheten

3.1 Realisering



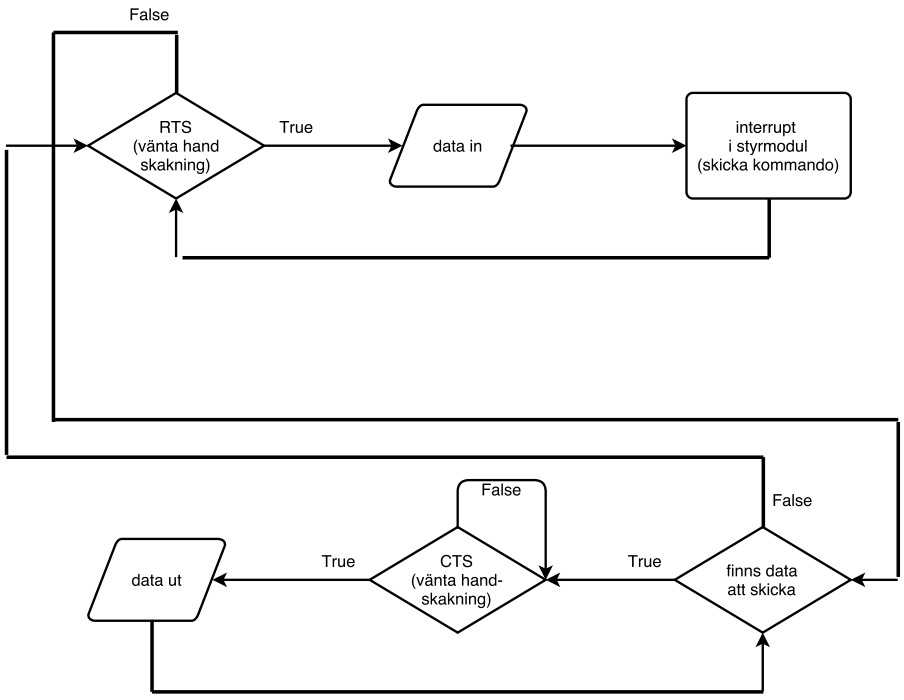
Figur 3: Översiktsblockschema över huvudenheten

Figur 3 visar hur kommunikationen sker inom huvudenheten på ett översiktligt plan. Processorn innehåller minne där vi kommer lagra data, variabler, kon-

stanter samt själva programmet som styr huvudenhetens logik (se *Logikenhet*). Processorn kommunicerar till omvärlden på två sätt, via en Bluetoothenhet och via en I2C-buss. Kommunikationen via Bluetooth bygger på en slags handskakningsprincip mellan processorn och den PC som Bluetoothenheten är kopplad till. Processorn skickar signalen *Clear To Send* till PC för att tala om att den är redo att ta emot data. PC:n skickar signalen *Request To Send* till processorn när den är redo att ta emot data. I2C-bussen används för kommunikation mellan huvud-, sensor- och styrenhet. Huvudenheten är ensam master på bussen och styr på så vis kommunikaitonen. Följdaktligen är den även ansvarig för att generera klockpulsen SCL. När antingen sensor- eller styrenheten är redo att skicka data eller låta huvudenheten läsa data, kommer de skicka ett avbrott (*interrupt*) till huvudenhetens processor. Den processor vi har tänkt att använda oss utav i huvuenhten är en ATmega1284 processor, med 128kB flashminne samt extra 20 kB för variabler och konstanter och en klocka på 0-20 MHz.

3.2 Kommunikationsenhet

All kommunikation med mjukvaruklienten sker över Bluetooth. PC:n ansluter sig till robotens Bluetooth-modul och blir därför master i kommunikationen. Informationen som ska skickas seriellt i paket om 7-8 bitar exklusive startbit och stopbitar. Innan överföring sker måste en så kallad handskakning ske mellan enheterna. En enhets RTS (*Request To Send*) signal talar om, om den är redo att skicka eller ta emot data, denna signal läses av på den andras CTS (*Clear To Send*) ingång. Data skickas till och från två separata in- och utgångar på Bluetooth-modulen.



Figur 4: Ett flödesdiagram på kommunikationen mellan PC:n och roboten sett ifrån roboten.

Flödesdiagrammet 4 visar hur kommunikation sker mellan PC:n och roboten. Roboten växlar mellan att kolla om det finns data att hämta eller om den får skicka data. Om det finns data att läsas in kommer den att läsas in till huvudmodulens processor. Efter inläsningen är klar återgår kollen efter data som kan skickas eller läsas in. Om det finns data att skickas inväntar roboten ett klartecken att skicka data och efter att den har skickat färdigt återgår logiken till att vänta på data att tas emot eller skickas igen.

3.3 Logikenhet

Logikenheten utgör hjärnan i roboten. Den hanterar både navigationsbeslut och själva kartläggandet. Utöver det styr den också kommunikationen på I2C-bussen.

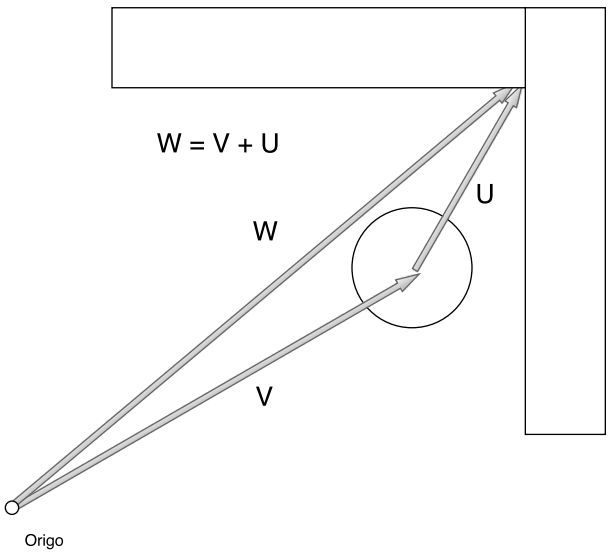
3.3.1 Navigation

Allt eftersom vi scannar vår omgivning så kommer roboten identifiera olika punkter som är relevanta att åka till för att fortsätta utforskandet. Den kommer fortsätta att navigera till relevanta punkter tills den är klar med utforskandet. En relevant punkt kan t.ex. vara en punkt kring ett hörn som skymmer vår

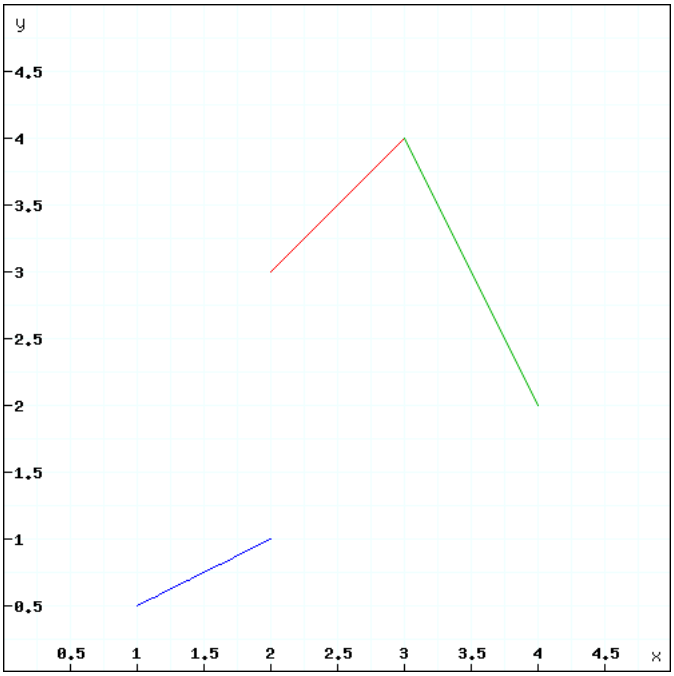
syn, eller en punkt som är för långt bort för att vi ska kunna veta säkert vad som finns där. För att hantera svängar kommer roboten antingen röra sig i ett diskret rutnät och rotera på plats i kvarts varv, eller kunna röra sig "kontinuerligt" och följa bezier-kurvor för svängar. För att veta vilken vinkel och position roboten har så kommer den använda sig att mätdata från ett gyroskop och en accelerometer, och genom att veta vilken vinkel den har kommer den kunna göra korrekta svängar.

3.3.2 Kartläggning

Allt eftersom roboten utforska den omkringliggande miljön kommer den att spara ned allt den upptäcker. Eftersom roboten rör sig under kartläggningen kommer nya hörn att behöva översättas till ett gemensamt koordinatsystem innan de sparas ned. Roboten löser det genom att vid start definera ett koordinatsystem med utgångspunkt där den startar och sedan, genom att veta sin egen position, översätta de nyupptäckta punkternas position till det ursprungliga koordinatsystemet med hjälp av vektoraddition. Se figur 5 för ett exempel.

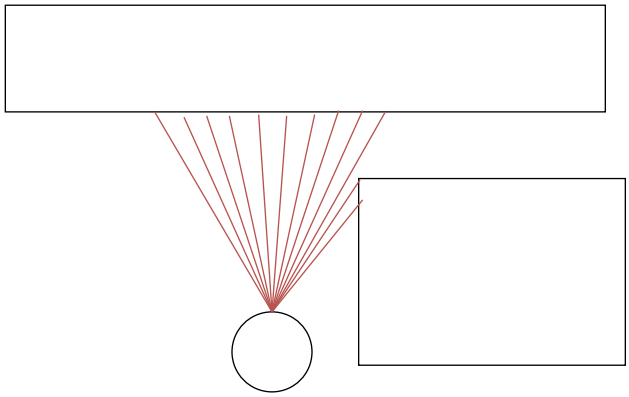


Figur 5: En illustration över hur koordinater för nyfunna hörn beräknas innan de sparas. V är robotens ortsvektor, U hörnets vektor i förhållande till roboten och W hörnets ortsvektor.

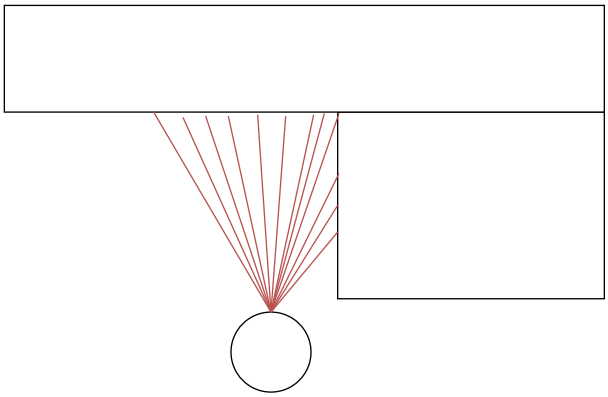


Figur 6: En linjär-interpolerad bild utav avståndsavläsningar från lasersensorn där x är vår sensors vinkel och y det uppmätt avståndet.

För att kartlägga omgivningen så konstruerar roboten upp en graf, med hörn som noder och väggar som bågar. För att detektera var hörn finns så läses lasersensorn av med jämna mellanrum. Vi kan visualisera det avlästa avståndet som en funktion av laserns vinkel som i figur 6, där diskontinuiteter och lokala maximum/minimum representerar hörn. En diskontinuitet betyder att vi hittat ett utstickande hörn med golvyta bakom sig och ett lokalt maximum/minimum innebär att vi hittat ett hörn på banan. I figuren så kommer punkten som ligger på (1, 0.5) sparas, tillsammans med en båge till punkten som ligger på (2, 1). Eftersom vi här upptäcker en diskontinuitet så kommer ingen båge sparas mellan (2, 1) och (2, 3), men bågar kommer ansluta (2, 3) till (3, 4) och (3, 4) till (4, 2). Se figur 7 och figur 8 för exempel på hur miljön kan se ut i de olika fallen. Eftersom upplösningen på avläsningar på korta avstånd och långa avstånd kan skilja sig mycket kommer värdemängden defineras på ett interval där en jämförelse mellan mätningarna fortfarande är meningfull. För att det inte ska bli ett problem att läsa av samma hörn flera gånger så kommer nya noder som introduceras i grafen att kollas mot redan existerande noder. Om avståndet mellan dem av mindre än en viss felmarginal så kommer det konstateras att de i själva verket är samma nod och den nya kommer ignoreras.

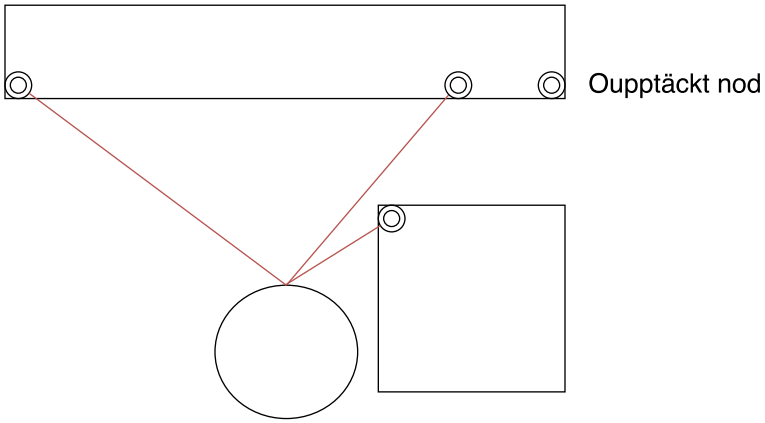


Figur 7: Ett fall där laseravståndsgrafen kommer innehålla en diskontinuitet.



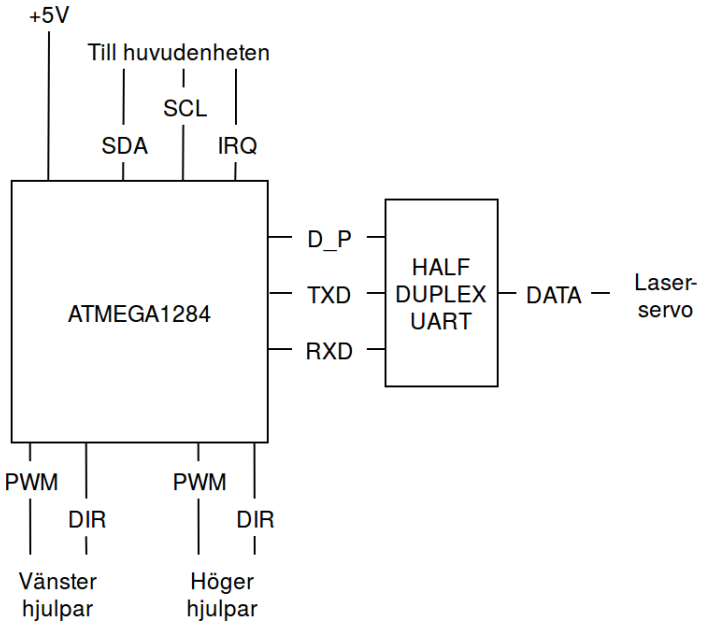
Figur 8: Ett fall där laseravståndsgrafen kommer innehålla ett lokalt maximum.

Roboten vet att den är klar när grafen innehåller fler än en komponent och alla komponenter innehåller en cykel, vilket motsvarar att alla väggar är sammanlänkade till ett rum/en köks-ö. Vid en diskontinuitet kommer en nod sparas för både hörnet och för nästa påträffade yta. Noden för nästa påträffade yta behövs för att vi ska kunna veta var någonstans på kartan vi fortfarande behöver utforska. Ett exempel syns i figur 9.



Figur 9: En illustration där en lövnod kommer skapas.

4 Styrenhet



Figur 10: Översiktsblockschema över styrenheten

Styrenheten (se figur 10) är över gränssnittet I2C kopplad till huvudenheten.

Styrenheten har en processor av typ ATmega1284.

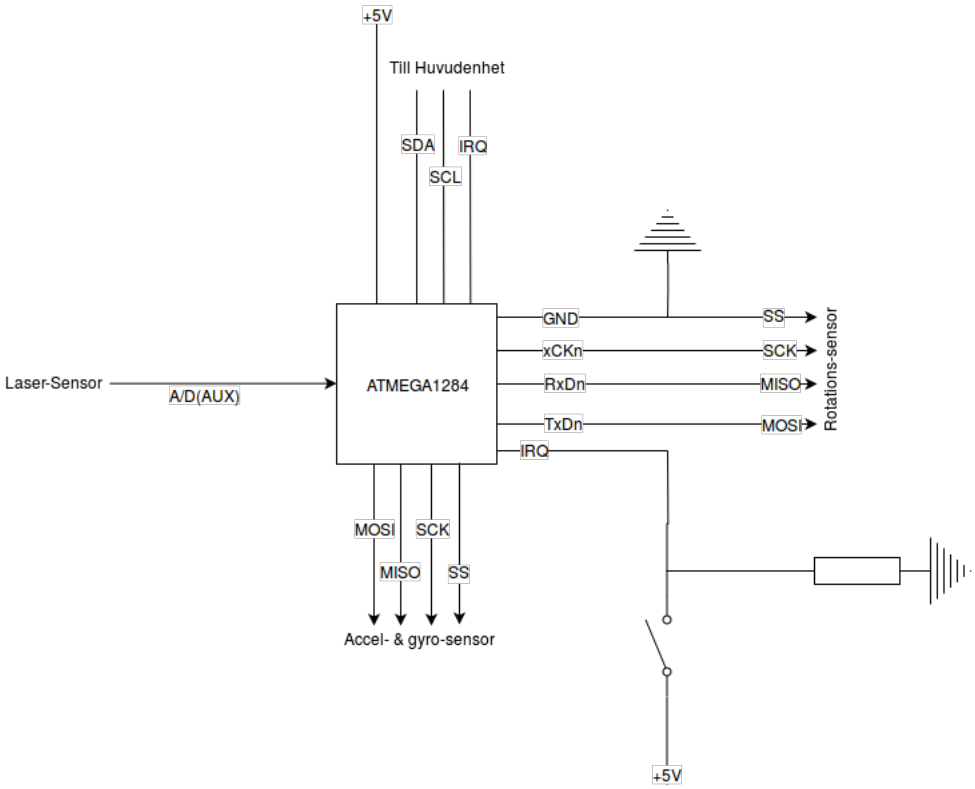
Styrenheten har fyra hjulservon indelade i två hjulpar. De två hjulparen kan styras oberoende av varandra. Dessa kontrollerar robotens position och kommunicerar med processorn med PWM och en riktningssignal.

Styrenheten har även ett rotationsservo som kontrollerar en avståndssensors rörelse. Detta servo kommunicerar med processorn via en half-duplex UART.

4.1 Rotation

Styrenhetens rotationsservo kommer att ha två sensorer fastsatta på sig, en avståndssensor samt ett gyro. Dessa sensorer hör till sensorenheten och kommer inte att kommunicera direkt med styrenheten. Sensordata processas av sensorenheten och skickas sedan till huvudenheten. Därefter tas ett beslut i huvudenheten baserat på sensorenhetens data hur rotationsservot ska styras.

5 Sensorenhet



Figur 11: Översikt över sensorernas koppling till sensorenheten

Sensorenheten (se figur 11) är kopplad till huvudenheten över gränssnittet I2C.

Sensorenheten ska ha en lasersensor kopplad till sig som kommunicerar med processorn över PWM.

Sensorenheten ska ha en accelerometer och gyrosensor kopplad till sig som ska hålla koll på hur mycket roboten har roterat och hur snabbt den åker. Dessa kommunicerar med varandra över SPI.

Sensorenheten ska ha en rotationssensor kopplad till sig som håller koll på hur mycket lasersensorn har roterat. Dessa kommunicerar med varandra över SPI.

Sensorenheten ska ha en knapp kopplad till sig på en av interrupt pinarna. Detta gör att när knappen trycks ned så växlar vi mellan autonomt och manuellt styrläge på roboten.

5.1 Avståndssensor

För att med så hög precision som möjligt kunna mäta avståndet till de kringläggande väggarna använder sig roboten av en LIDAR-Lite v2 laseravståndsmätare. Denna sensor är monterad tillsammans med ett servo på toppen av roboten för att kunna rotera fram och tilbaka och mäta avstånd i 360°. Den sensordata som rapporteras av lasern kombineras med data från ett gyro som också monterats på rotationsservot för att kunna ge en beskrivning av området runt roboten.

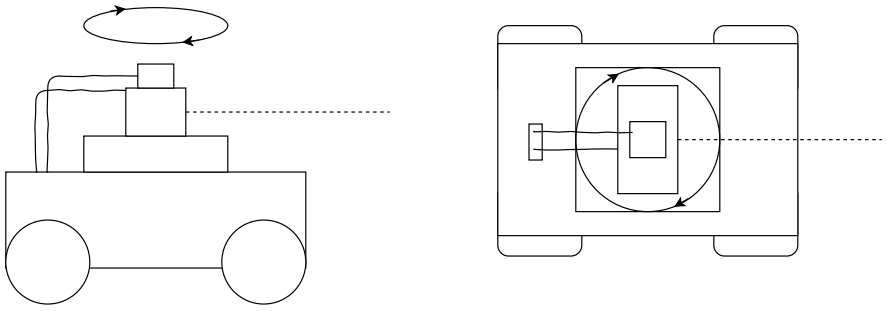
5.1.1 Begränsningar

Att montera två sensorer på en roterande del medför en del problem med hur t.ex. sladdar hanteras. Servot måste rotera i svepande rörelser fram och tillbaka för att inte dra sönder sladdarna och längden på sladdarna får inte vara för lång för att undvika att de hamnar ivägen.

5.1.2 Rotation

För att både kunna leverera korrekt data till huvudenheten samt undvika att förstöra hårdvaran på grund av utdragna sladdar behöver detektionen av sensorservots nuvarande rotation hålla sig innanför en viss felmarginal. Dels så rapporterar servot AX-12 sin nuvarande rotationsvinkel via dess dataport och dels kommer gyrosensorn ge en uppskattning om hur servot roterat. Blir felet för stort kan AX-12-servots rotation manuellt ändras till en given vinkel. I och med att data om rotationen kommer rapporteras från både sensorenheten (gyro) och styrenheten (servo) så sker beräkningen av vinkeln i huvudenheten.

5.1.3 Montering

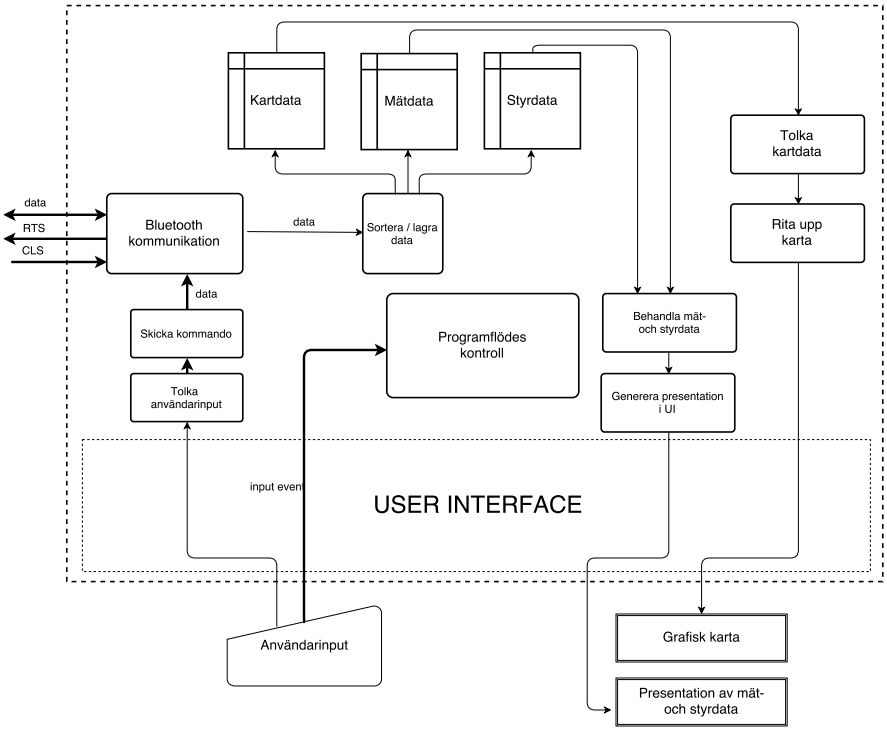


Figur 12: Skiss över montering av laser

Ett AX-12-servo monteras ovanpå roboten i mitten (se figur 12). Ovanpå servot monteras laseravståndssensorn som i sin tur har gyrot monterat ovanpå. Från

både servot och gyrot går fria sladdar som är kopplade till sensorenheten. De sladdar som behövs för att koppla in servot kommer inte ligga ivägen för rotationen och är därför undantagna från figuren. Förslagsvis går någon sorts ränna på båda sidorna av rotationstornet där sladdarna kan röra sig fritt (ej inkluderat i figur).

6 Mjukvaruklient



Figur 13: Översiktligt flödesschema över mjukvaruklienten

I mjukvaruklienten ingår presentations- och fjärrstyrningsenheten. Mjukvaruklienten ska presentera den karta som med robotens hjälp genererats över rummet, samt låta användaren styra roboten när den är i sitt manuella läge. Ovanstående figur (13) visar ett översiktligt flödesschema av hur mjukvaruklienten fungerar. Mjukvaruklienten kommunicerar med roboten via Bluetooth. När data tas emot så ska denna sorteras utifrån om det är kartdata, mätdata eller styrdata - vilket kommer kännas igen av någon form av ID i början av överföringen. Datan lagras sedan tills klienten är redo att behandla den för presentation, vilket ska ske löpande - dvs. data ska presenteras och kartan växa fram allt eftersom roboten kör. Kommandon för att styra roboten manuellt skickas via Bluetooth till robo-

ten. En lista på förflyttningskommandon som roboten stödjer kan återfinnas i kravspecifikationen. Som ett sekundärt krav har vi att man ska kunna byta mellan manuellt och autonomt läge genom mjukvaruklienten (detta är ej illustrerat i figuren ovan). Signalerna RTS och CLS förklaras i avsnittet *Huvudenhet*.

Referenser

[1] Atmel. 8-bit avr microcontroller with 128k bytes in-system programmable flash, September 2016.

[2] Dynamixel. Dynamixel ax-12, September 2016.

[3] FireFly. Firefly bluetooth modem - bluesmirf gold, September 2016.

[4] InvenSense. Mpu-6000 and mpu-6050 product specification, September 2016.

[5] Melexis. Mlx90609 angular rate sensor (standard version), September 2016.

[6] Linköpings Universitet Niklas Carlén, ISY. Fyrhjulig robot - terminator, tekniks specifikation, September 2016.

[7] Linköpings Universitet Niklas Carlén, ISY. Motorstyrning med pwm, September 2016.

[8] Sparkfun. Lidar-lite v2 overview, September 2016.

[9] Unknown. Så här fungerar tryckknappen, September 2016.