



SYSTEMSKISS

Redaktör: Herman Lundkvist

Version 0.2

Status

Granskad	Herman. John	2014-09-26
Godkänd		



MINOTAUROS

Grupp 6, HT 2014, Labyrintrobot
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Adrian Sidenvall	projektledare (PL)	072-732 10 18	adrsi117@student.liu.se
John Litborn	dokumentansvarig (DOK)	072-928 63 42	johli603@student.liu.se
Albert Karlsson	Sensorehetsansvarig (SA)	076-113 54 78	albka735@student.liu.se
Anthon Johansson	Mjukvaruansvarig (MVA)	072-312 86 69	antjo383@student.liu.se
Tommy Nguyen	Testansvarig (TA)	070-750 62 14	tomng154@student.liu.se
Johan Lindberg	Motorehetsansvarig (MA)	076-822 50 14	johli393@student.liu.se
Herman Lundkvist	Beslutsenhetsansvarig (BA)	070-277 47 82	herlu184@student.liu.se

Kund: Mattias Krysander

Kontaktperson hos kund: Mattias Krysander, 013-28 21 98, matkr@isy.liu.se

Kursansvarig: [Tomas Svensson](mailto:tomas.svensson@isy.liu.se), 3B:528, 013-28 13 68, tomas.svensson@isy.liu.se

Handledare: Olov Andersson, [Telefon: +46 \(0\)13 28 2658](tel:+46013282658), [Epost: olov@isy.liu.se](mailto:olov@isy.liu.se)



Innehåll

1 Inledning	1
2 Översikt	2
2.1 Relationer mellan delsystemen	3
2.2 Gränssnitt	4
2.2.1 Förslag 1	4
2.2.2 Förslag 2	4
3 Delsystem	4
3.1 Beslutsenhet	4
3.2 Sensorenhet	5
3.2.1 Sensorer framåt och bakåt	5
3.2.1.1 Förslag 1	5
3.2.1.2 Förslag 2	5
3.2.2 Sensorer på sidorna	5
3.2.2.1 Förslag 1	5
3.2.2.2 Förslag 2	5
3.2.3 Gyro	6
3.2.3.1 Förslag 1	6
3.2.3.2 Förslag 2	6
3.3 Motorenhet	6
3.4 Programvara	6



Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2014-09-25	Första versionen	Herman, John, Albert, Johan	Herman, John
0.2	2014-10-01	Rättade till formateringsfel, gjorde några förtydlingar, tog bort förslag om gyro och gjorde den obligatorisk, lade till bild över GUI.	Herman, Albert	



1 Inledning

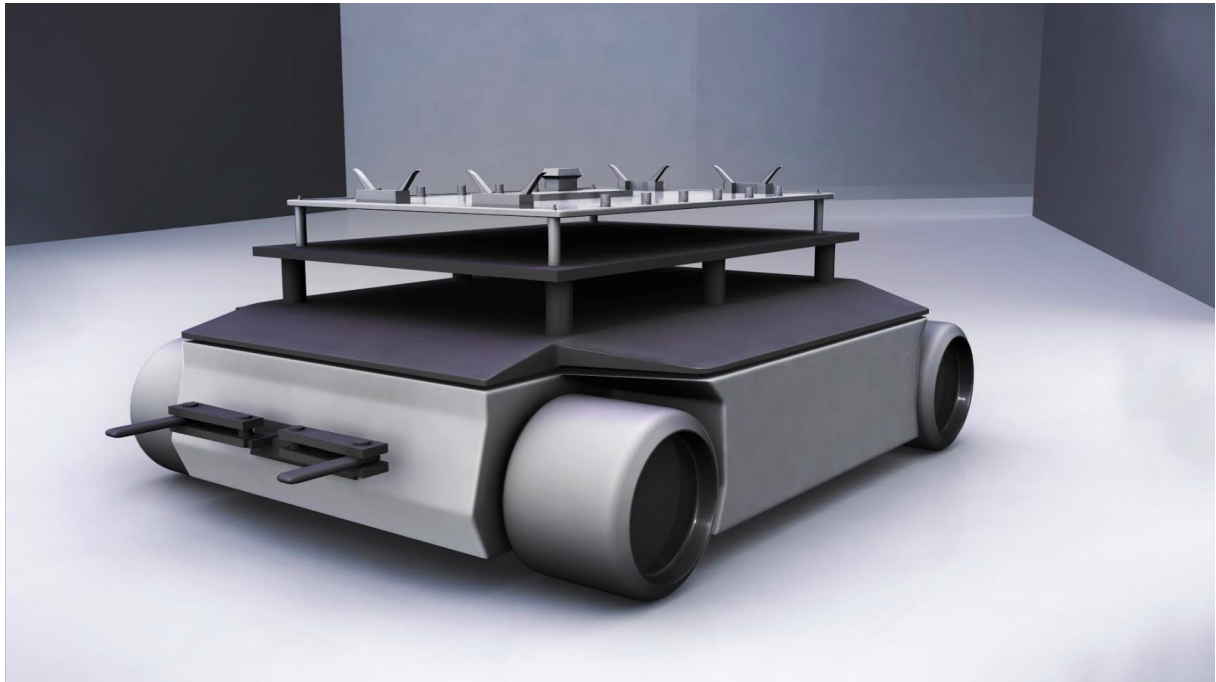
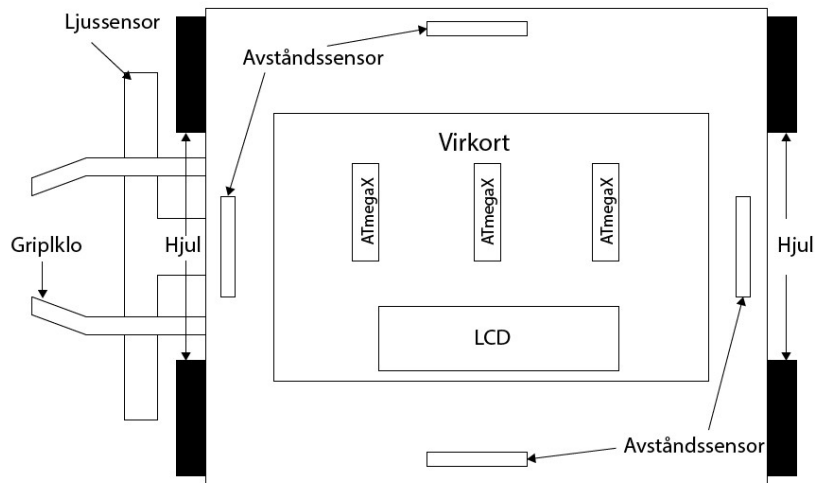


Illustration 1: Rendering över roboten i en labyrint.

Roboten som ska konstrueras ska klara av att utforska en labyrint, hitta och plocka upp ett föremål för att sedan ta den närmsta vägen ut. Illustration 1 visar hur den färdiga roboten kan tänkas se ut. Roboten ska byggas på ett chassi med fyra hjul. Avståndssensorer, gyro och ljussensor krävs för att kunna navigera labyrinten. I roboten ska det finnas tre olika moduler som alla innehåller en processor som är fastsatta på ett virkört. Det ska också finnas en skärm som visar sensorinformation. Figur 1 visar placeringen av de olika komponenterna. Se kravspecifikation [1] för mer ingående detaljer om robotens egenskaper.



Figur 1: Skiss över placering av robotens komponenter.

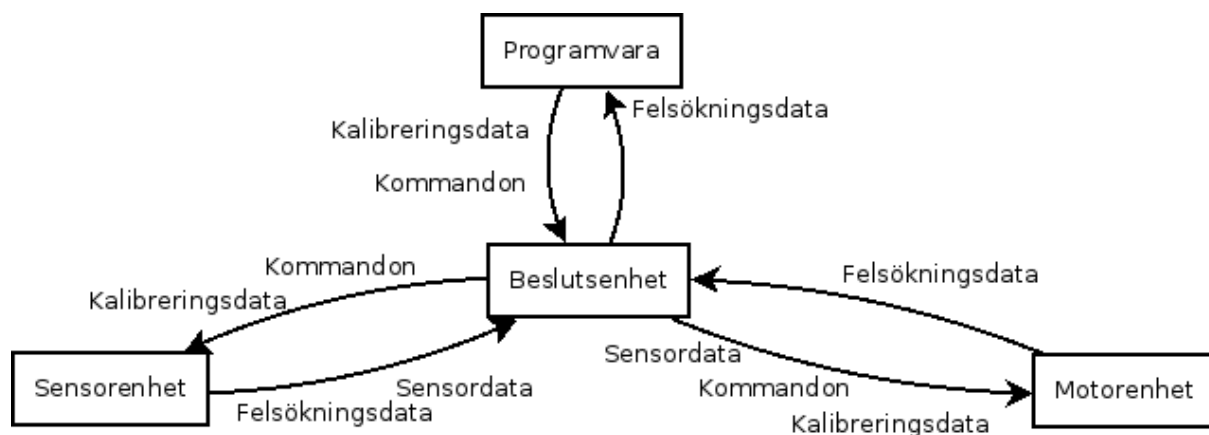
2 Översikt

Systemet består av en programvara och en robot. Roboten består i sin tur av tre enheter, där varje enhet implementerar funktionaliteten hos en av enheterna som beskrivs i kravspecifikationen. De tre enheterna är:

- en beslutsenhet, som motsvarar kommunikationsenheten;
- en sensorenhet, som motsvarar sensorenheten;
- och en motorenhet, som motsvarar styrenheten.

Dessa enheter innehåller alla en AVR-processor var.

2.1 Relationer mellan delsystemen



Figur 2: Delsystemen med informationsflöde.



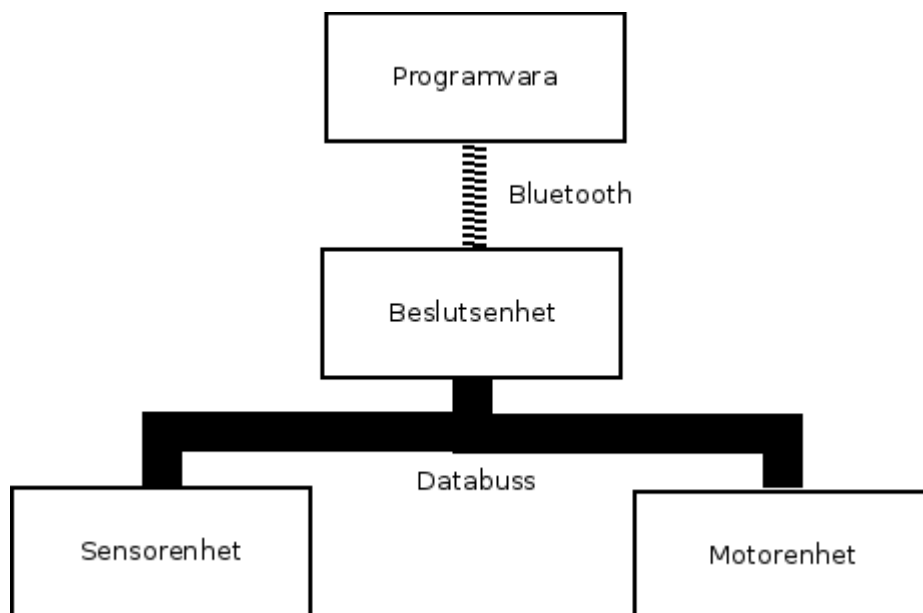
Beslutsenheten kan ses som navet i systemet, då den sköter kommunikationen mellan alla övriga delsystem, se Figur 2. Beslutsenheten tar emot data från sensorenheten, som den tolkat från olika sensorer, och bygger upp en modell av labyrinten, vilken den kan ta beslut ifrån. Alternativt kan besluten tas av användaren. I detta fall interagerar användaren med programvaran som sedan skickar kommandon till beslutsenheten.

Besluten realiseras sedan genom att kommandon först skickas från beslutsenheten till motorenheten, t.ex.: ”kör rakt fram”, ”rotera vänster”, och ”grip föremål”. Sedan utförs kommandona av motorenheten med hjälp av motorerna och servon till griparmen. För vissa kommandon behöver motorenheten data från sensorenheten, t.ex. för styralgoritmen. I dessa fall kopierar beslutsenheten data från sensorenheten och skickar data vidare till motorenheten. Utöver detta skickas felsökningsdata (t.ex. rådata och behandlat data från sensorerna, styrsignaler till motorer och servo, och olika variabler som ger information om tillståndet hos de olika enheterna) från sensorenheten och motorenheten via beslutsenheten till programvaran. Kalibreringsdata, t.ex. värden på parametrar i olika algoritmer, kan även skickas från programvaran till beslutsenheten och eventuellt vidare till sensor- eller motorenheten.

Slutligen, kan beslutsenheten läsa av en knapp och en brytare för att växla mellan autonomt läge och fjärrläge.

För att undvika kopiering av sensordata i beslutsenheten, övervägdes att sensor- och motorenheterna skulle kunna kommunicera med varandra direkt. Denna idé valdes dock bort, eftersom att det skulle innebära lägre modularitet, eftersom sensor- och motorenheterna skulle bli beroende av två gränssnitt.

2.2 Gränssnitt



Figur 3: Delsystemen med gränssnitt.

Robotens enheter ansluts med en gemensam databuss och programvaran och Beslutsenheten ansluts med Bluetooth, se Figur 3. För databussen ges två förslag.



2.2.1 Förslag 1

Enheterna ansluts med I²C, med beslutsenheten som master. Nackdelen med denna metod, jämfört med SPI är att överföringshastigheten är lägre.

2.2.2 Förslag 2

Enheterna ansluts med SPI, med beslutsenheten som master. Nackdelen med denna metod, jämfört med I²C, är att man behöver använda fler anslutningar, fem (SCK, MOSI, MISO, och två SS; en för motor- och en för sensorenheten) jämfört med två.

3 Delsystem

Nedan beskrivs delsystemen mer ingående.

3.1 Beslutsenhet

Som tidigare beskrivits, är beslutsenheten den centrala modulen i roboten som tar alla beslut och samordnar de andra delsystemen. Beslutsenheten hanterar även kommunikation med programvaran på datorn via bluetooth. Se figur 1.

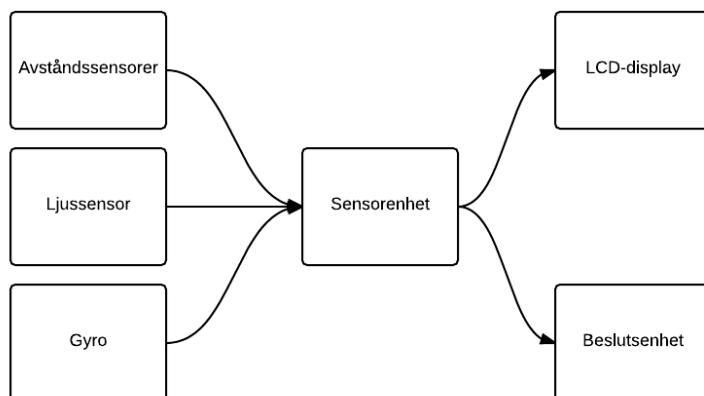
Beslutsenheten kommer att ha följande funktionalitet:

- Kommunikation med programvara: kommunikation med programvara sker via bluetooth med modulen firefly[2]. Till programvaran skickas felsökningsdata från sensor-, motor- och även beslutsenheten själv. Från programvaran skickas kalibreringsdata, samt kommandon när roboten är i manuellt läge. Kalibreringsdatan skickas under utveckling och testning av roboten istället för att den ska sparas som konstanter i de olika enheterna, vilket skulle försvåra testning och utveckling.
- Kommunikation med motorenhet: kommunikation med motorenhet sköts via databussen. Till motorenheten skickas kommandon (t.ex. kör, sväng, grip), sensordata (till t.ex. PD-reglering) samt kalibreringsdata (främst till PD-reglering). Från motorenheten till beslutsenheten skickas felsökningsdata samt bekräftelse på vissa kommandon (t.ex. grip).
- Kommunikation med sensorenhet: kommunikation med sensorenhet sköts via databussen. Till sensorenheten skickas en del kommandon (t.ex. Starta kalibrering), samt kalibreringsdata. Från sensorenheten skickas sensor- och felsökningsdata.
- Beslutstagande: I autonomt läge kommer beslutsenheten köra en state-machine som med hjälp av data från de andra enheterna beslutar vad som ska göras och skickar kommandon till de andra enheterna.
- Utforskning av labyrint: labyrinten kommer att utforskas m.h.a. en djupet-först-sökning.
- Beslutsdelegering: I manuellt läge kommer beslutsenheten delegera kommandon den får via bluetooth till de andra enheterna.



3.2 Sensorenhet

Roboten kommer att ha en sensor framåt och eventuellt en bakåt. Den kommer även ha sensorer på vardera sida för att känna av väggar och korsningar och ett gyro för att hålla koll på rotationer. En ljussensor kommer användas för att upptäcka tejpmarkeringar. Det sensorenheten kommer att göra är att tolka och behandla det sensorerna ger, skicka vidare dem till beslutsenheten, enligt Figur 4 och sensordata ska även visas på en LCD-display.



Figur 4: Översiktlig bild över sensorer och hur dess data skickas

3.2.1 Sensorer framåt och bakåt

3.2.1.1 Förslag 1

Roboten kommer ha två ultraljudssensorer, en framåt och en bakåt. Har man en sensor bakåt kan man hålla koll på hur långt man har åkt från en viss punkt med hjälp av sensorn bakåt om sensorn framåt inte kan få någon signal. Denna information kan roboten sedan använda när den ska åka tillbakavägen med föremålet, för att veta när den kan öka hastigheten och när den ska förbereda sig på att svänga. Detta avstånd skulle man dock kunna räkna ut m.h.a. en avståndssensor framåt, men med två stycken minskar felmarginalen.

Dessutom blir det möjligt för roboten att köra baklänges i Labyrinten (förutom när den behöver upptäcka tejpmarkering). Detta skulle kunna användas i återvändsgränder, eller när roboten har plockat upp föremålet, för att slippa rotera ett halvt med roboten.

3.2.1.2 Förslag 2

Roboten kommer att ha en ultraljudssensor framåt, eftersom vi antagligen kommer kunna navigera igenom labyrinten utan att ha en extra sensor.



3.2.2 Sensorer på sidorna

3.2.2.1 Förslag 1

På sidorna av roboten kommer det sitta två optiska sensorer som har en kort räckvidd eftersom roboten kommer aldrig att befinna sig långt ifrån väggarna på sidorna. Dock måste man linjärisera signalen som man får från sensorerna.

3.2.2.2 Förslag 2

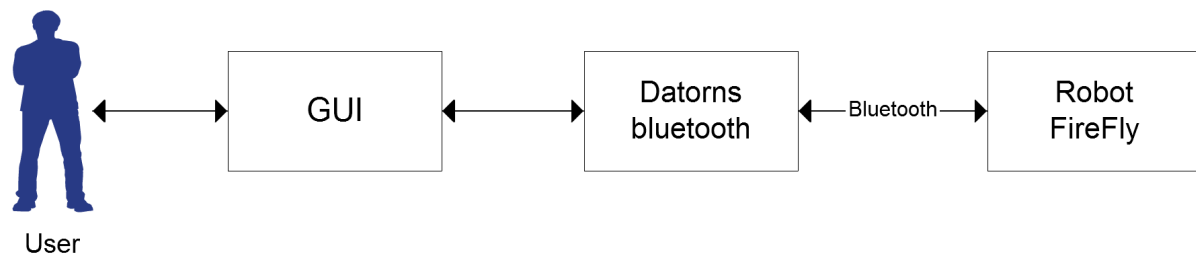
Roboten kommer ha två ultraljudssensorer på vardera sida, som både kan täcka långa och korta avstånd.

3.3 Motorenhet

Det motorenheten kommer göra är att följa kommandon som beslutsenheten kommer ge genom att reglera motorerna. Kommandona roboten ska kunna göra är t.ex. svänga åt höger eller vänster, åka rakt fram och greppa med griplon.

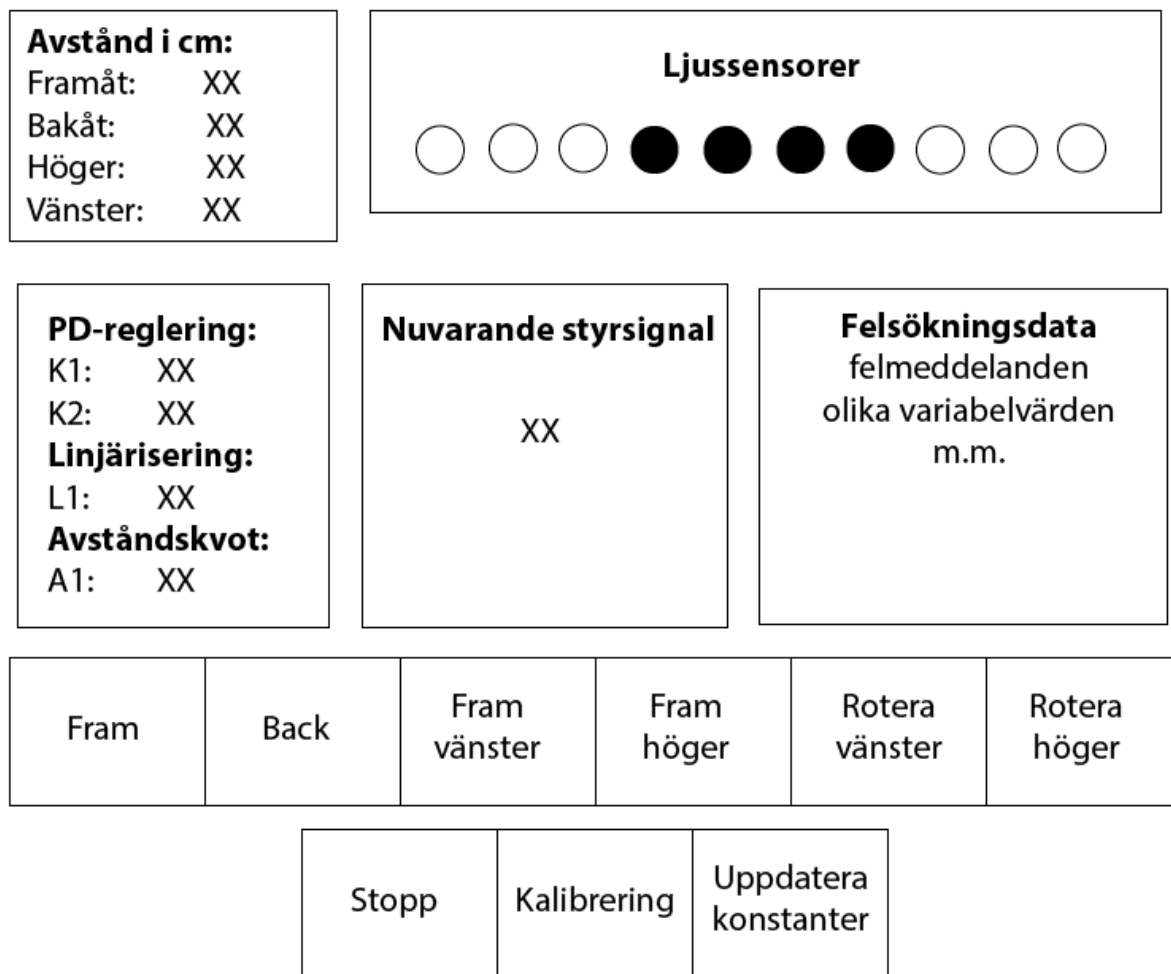
Det kommer även finnas en PD-reglering i motorenheten för att se till så att roboten åker rakt.

3.4 Programvara



Figur 5: Relationerna mellan de olika aktörerna och komponenterna.

Programvaran kommer innehålla ett GUI som visar sensorinformation, har knappar för att skicka olika kommandon och komponenter som möjliggör inställning av kalibreringskonstanter. Programvaran kommer köras på en dator med bluetooth som möjliggör all kommunikation med roboten. Se Figur 5 för relationer. Figur 6 Visar hur det grafiska användargränssnittet kan tänkas se ut.



Figur 6: Förslag på utseendet hos programmet.



Referenser

Projektdokument

[1] Redaktör: JohnLitborn, Kravspecifikation, version 1.0. projekt: Minotauros. Kurskod: TSEA29, Linköpings tekniska högskola.

Elektroniska källor

[2] ”Firefly, Bluetooth Modem – BlueSMiRF Gold”. Hämtat från
<<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/firefly.pdf>> 2014-09-24