

Teknisk dokumentation

Redaktör: Hannes Snögren

Version 0.1

Status

Granskad		
Godkänd		

PROJEKTIDENTITET

HT2, 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	Produkten	1
3	Systemet	1
4	Kommunikation	1
4.1	PC-enhet \longleftrightarrow Huvudenhet	1
4.2	Huvudenhet \longleftrightarrow Styrenhet	1
4.3	Instruktionslista	2
4.4	Huvudenhet \longleftrightarrow Sensorenhet	2
4.5	Instruktionslista	2
5	PC-enhet	3
5.1	Hårdvara	3
5.2	Mjukvara	3
6	Huvudenhet	3
6.1	Teori	3
6.2	Hårdvara	3
6.3	Mjukvara	3
7	Styrenhet	3
7.1	Hårdvara	3
7.2	Mjukvara	3
8	Sensorenhet	4
8.1	Hårdvara	4
8.1.1	Reflexsensormodul	5
8.1.2	Avståndssensorer	5
8.2	Mjukvara	6
9	Slutsatser	6

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1				

1 Inledning

Systemet Gloria är en lagerrobot som kan följa en bana, plocka upp paket vid särskilda stationer och sätta ned dem vid nästa tomma station.

Projektet utfördes som ett moment i kursen TSEA29 vid Linköpings Universitet under HT 2014[1]. Syftet med projektet var att ge gruppmedlemmarna övning i konstruktion och utveckling med mikrodatorer och erfarenhet i att jobba enligt en projektmodell, i det här fallet LIPS.

Det här dokumentet dokumenterar hur systemet är designat och fungerar. Syftet är dels att kunden skall kunna lösa uppkomna problem eller vidareutveckla systemet och dels att utomstående part i utbildningsyfte skall kunna förstå hur systemet fungerar.

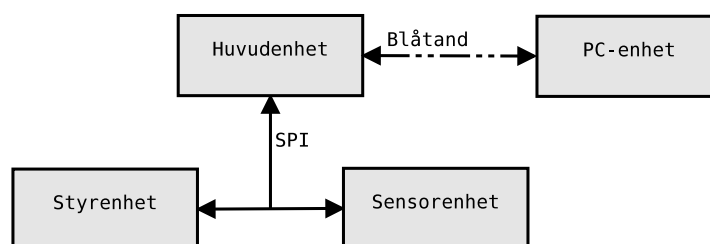
2 Produkten

Systemet Gloria är en fyrhjulig robot med en robotarm. **Den kan massor med saker och skitmycket funktionalitet**

3 Systemet

4 Kommunikation

Överväger att lägga figur 1 i Systemet och protokollet under varje enhet.



Figur 1 – Översikt systemets kommunikationskanaler

Figur 1 ger en översikt av vilka moduler som kommunicerar med varandra och på vilket sätt detta sker. Styrenheten och Sensorenheten använder samma SPI-buss fast med olika Slave Select-pinnar.

4.1 PC-enhet \longleftrightarrow Huvudenhet

PC-enheten och Huvudenheten kommunicerar över Blåttand. **Hur set det här protokollet ut i verkligheten?**

4.2 Huvudenhet \longleftrightarrow Styrenhet

Huvudenheten kommunicerar med Styrenheten över SPI. Storleken på en instruktion är antingen fyra eller sju bytes lång. För att förebygga synkroniseringsproblem skickas först två startbytes bestående av hexadecimalt *0xff* följt av längden på instruktionen (startbytesen och längdbyten räknas ej med). Därefter följer instruktionsbyten som består av två delar. De fyra högsta bitarna anger vilken instruktion som ska utföras enligt tabell 1, och de fyra lägsta vilken servo eller motorpar instruktionen rör enligt tabell 2. I fallet att instruktionen är *Sätt register A till D* krävs dessutom två databytes.

Då ett motorpar adresseras anger den minsta biten i den första databyten vilken riktning motorparet skall röra sig. 0 anger framåt, 1 bakåt **Stämmer?**. Den andra databyten anger vilken fart vi vill att motorerna skall röra sig i. Notera att motorerna sätts med kommandot.

De servon vi arbetar med har en upplösning på 10 bitar både för position och hastighet. Vi måste alltså ha två databytes när vi vill ändra någon egenskap hos dessa. De två minsta bitarna i den första databyten är de två högsta bitarna och därefter följer den andra databyten.

Tabell för att illustrera bitar hit och dit?

4.3 Instruktionslista

Instruktion	Argument	Beskrivning
0000		Stoppa samtliga servon och motorer Behöver implementeras
0001	A, D	Sätt register A till D
0010	A	Utför givna kommandon för A
0011	A, D	Sätt servohastighet för A till D
0100	A	Returnera Status för A Ska det användas?

Tabell 1 – Kommandon från huvudenhet till styrenhet **Stämmer tabellen?**

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 2 – Adresser för adressering till styrenhet

4.4 Huvudenhet \longleftrightarrow Sensorenhet

Huvudenheten kommunicerar med Sensorenheten över SPI. Protokollet mellan dessa moduler är väldigt primitivt då det endast finns en instruktion, returnera sensordata för begärd sensor. Instruktionen anges av de 4 högsta bitarna av instruktionsbyten enligt tabell 3. Huvudenhetens begäran är då bara på en enda databyte och Sensorenheten svarar med de 8 högsta bitarna för den efterfrågade sensorn. Vilken sensor som enheten returnerar data för anges av de 4 minsta bitarna i instruktionsbyten enligt tabell 4.

4.5 Instruktionslista

Instruktion	Argument	Beskrivning
0000	A	Returnera sensordata för A

Tabell 3 – Instruktion från huvudenhet till sensorenhet

Adress	Beskrivning
0000	Linjesensor 1
0001	Linjesensor 2
0010	Linjesensor 3
0011	Linjesensor 4
0100	Linjesensor 5
0101	Linjesensor 6
0110	Linjesensor 7
0111	Linjesensor 8
1000	Linjesensor 9
1001	Linjesensor 10
1010	Linjesensor 11
1011	Avståndssensor Höger
1100	Avståndssensor Vänster

Tabell 4 – Adresser för instruktioner till sensorenhet

5 PC-enhet

5.1 Hårdvara

5.2 Mjukvara

6 Huvudenhet

6.1 Teori

Hur fungerar regleringen Hur beräknar vi armens vinklar? Hur tolkar vi sensordata?

6.2 Hårdvara

Den hårdvara som ingår i huvudenheten består av en Beagleboard **typ/modell av Beagle-board?**. På denna sitter en Blåtands-enhet av typ **Typ/modell av BT?** monterad för kommunikation med PC-enheten. **Ska SPI-kabel nämnas? Bild på BB?**

6.3 Mjukvara

Mjukvaran är uppdelad i **2?** trådar, där varje tråd fyller en specifik funktion. Sensortråden hämtar kontinuerligt uppdaterad data från sensorenheten, PC-tråden hanterar alla förfrågningar från PC-enheten och maintråden beslutar vad som ska skickas till styrenheten. **subsubsections för varje tråd med mer ingående?**

Schema över asmycket trådar

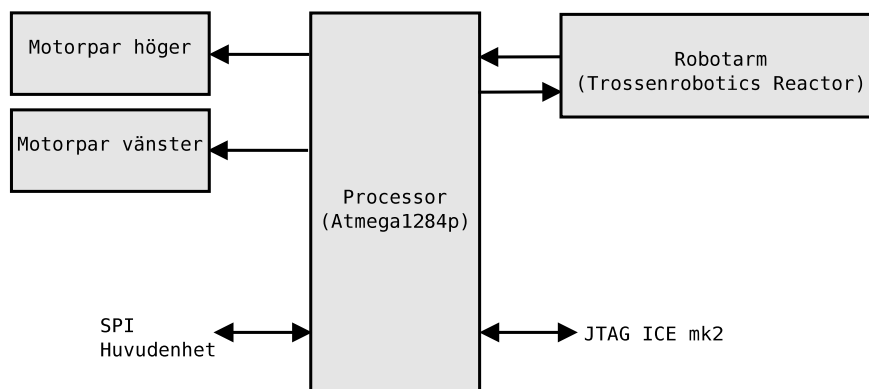
7 Styrenhet

Styrenheten har till uppgift att ta kommandon från huvudenheten och producera ut signaler som motorer och servon kan förstå.

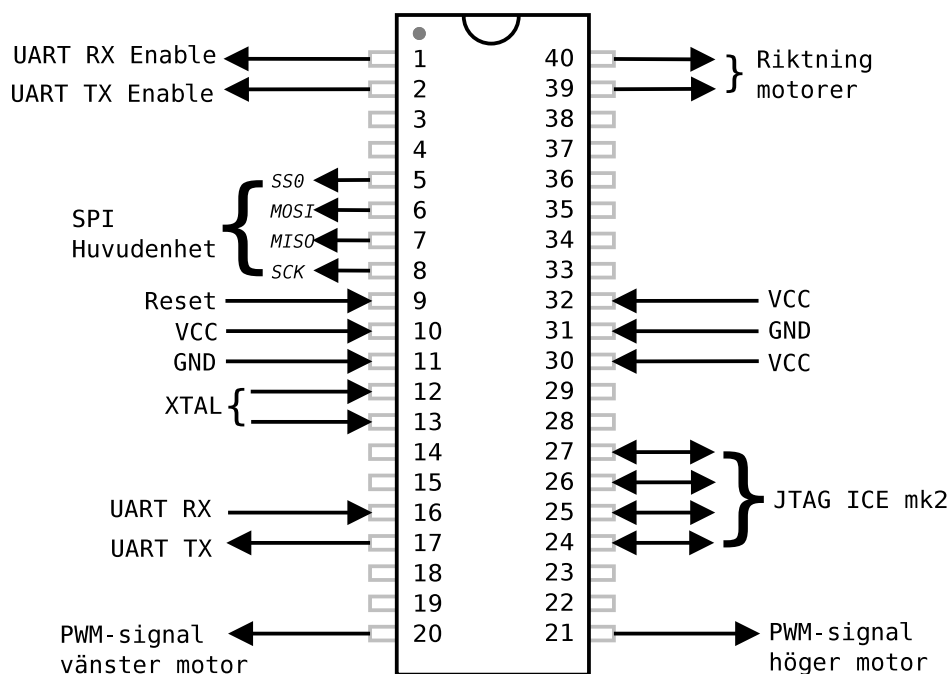
7.1 Hårdvara

7.2 Mjukvara

Flödesschema styrenheten



Figur 2 – Översikt av styrenheten



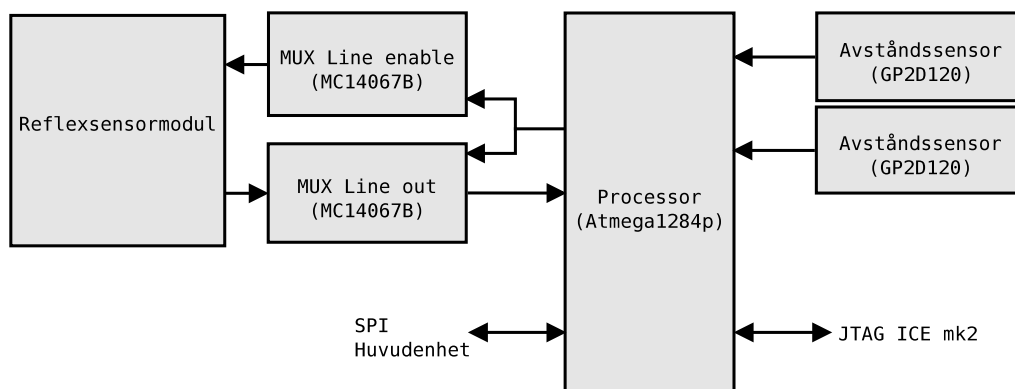
Figur 3 – Schema över hur enkretssdatorn i styrenheten är ansluten till övrig hårdvara.

8 Sensorenhet

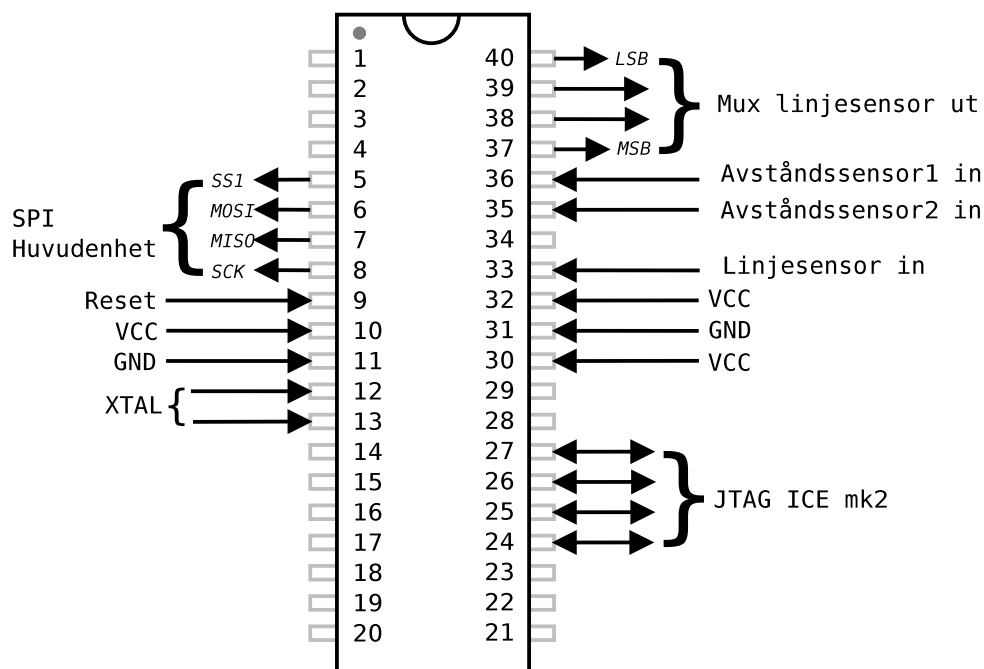
Sensorenheten har till uppgift att förse huvudenheten med sensordata. Sensordatan den returnerar är obehandlad rådata.

8.1 Hårdvara

Sensorenheten består av en Atmega1284p, en reflexsensormodul och två avståndssensorer. Reflexsensormodulen är kopplad via två 16-1 muxar till enkretsdatorn. Den ena muxen används för att styra en konstant hög enable-signal till rätt reflexsensor och den andra för att välja rätt ut-signal. Båda muxarna styrs av samma styrsignal från processorn. Enkretsdatorn är ansluten till huvudenheten med en **20lol-pinnarskabel?** över vilken de kommunicerar via SPI.



Figur 4 – Översikt av sensorenheten



Figur 5 – Schema över hur enkretssdatoren i sensorenheten är ansluten till övrig hårdvara.

8.1.1 Reflexsensormodul

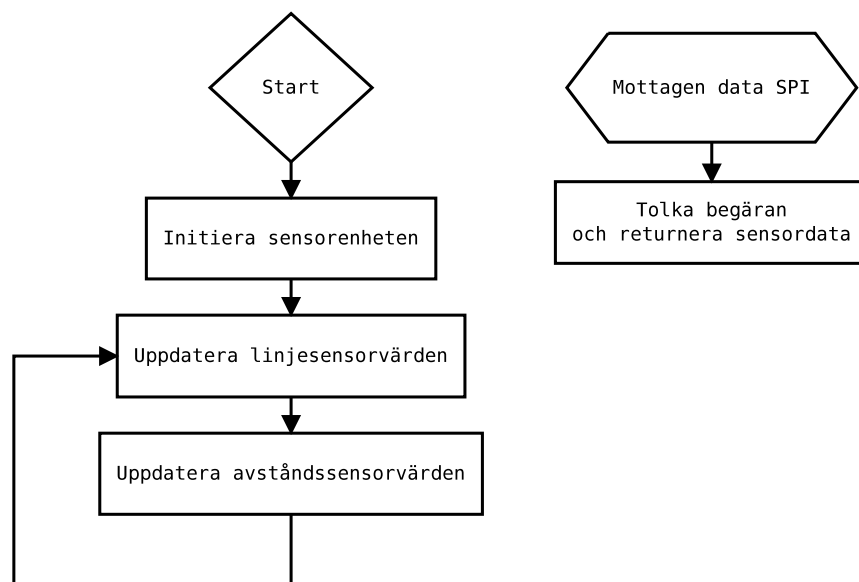
Reflexsensormodulens syfte är att detektera banan längs vilken roboten skall röra sig. Den består av 11 reflexsensorer som i sig består av en IR-diod och en fototransistor. Utsignalen ligger mellan 0V och 5V. De ger låg utspänning då underlaget reflekterar mycket ljus, och hög utspänning då lite ljus reflekteras.

8.1.2 Avståndssensorer

Avståndssensorerna är av typen GP2D120 vilken använder ljus för att detektera avståndet. Dess utsignal är en analog spänning mellan 0V och 3.2V.

8.2 Mjukvara

All mjukvara på sensorenheten är skriven i *C* och finns på enkretsdatorn. Den består av två delar, illustrerat i figur 6.



Figur 6 – Till vänster sensorenhetens **huvudslina**, till höger den avbrottsrutin som körs vid mottagen data över SPI.

För det första körs en mainloop där sensordata kontinuerligt uppdateras. Först itererar vi igenom Reflexsensorerna genom att först styra om enable-signalen till den aktuella sensorn och därefter utföra en AD-omvandling på den signal vi får tillbaks och sedan gå till nästa. Därefter utför vi i tur och ordning en AD-omvandling på insignalerna från avståndssensorerna. Alla värden sparas i en global datastruktur.

För det andra tar sensorenheten emot begäran om sensordata från huvudenheten över SPI. När en sådan förfrågan inkommer triggas ett avbrott i vilket sensorenheten svarar med det aktuella sensorvärdet. Då inget i avläsningen av sensorerna är tidskritiskt behöver vi inte oroa oss för när dessa avbrott kommer.

9 Slutsatser

Referenser

- [1] <http://www.isy.liu.se/edu/kurs/TSEA29/>, information hämtad 2014-09-**tidigt?**.
- [2] <http://www.commsys.isy.liu.se/sv/student/kurser/TATA62/lips>, information hämtad 2014-09-25.
- [3] LIPS, Studentlitteratur, Tomas Svensson, Christian Krysander **Gör rätt**
- [4] <http://www.trossenrobotics.com/p/phantomx-ax-12-reactor-robot-arm.aspx>, information hämtad **hittepå**.
- [5] http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm, information hämtad 2014-10-24.
- [6] <http://www.atmel.com/images/doc8059.pdf>, information hämtad **hittepådatum**.
- [7] http://www.sharpsma.com/webfm_send/1205, information hämtad **Hittepådatum**.
- [8] <http://www.ubuntu.com>, **Hur refererar vi ubuntu?**
- [9] <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/reflexsensormodul.pdf>, information hämtad **hittepådatum**.
- [10] <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/4067b.pdf>, information hämtad **hittepådatum**.
- [11] <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/sn74ls125.pdf>, information hämtad **hittepådatum**.
- [12] <http://www.ti.com/lit/ds/symlink/txb0104.pdf>, information hämtad **hittepådatum**.

Resurs/datablad för Beagleboard?
Datablad för kristall/oscillator
Datablad för tryckknapp