

Mapmaster2001

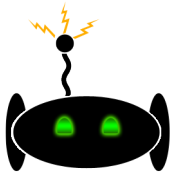
27 februari 2014

Designspecifikation

Kandidatprojekt Y - Grupp 8 - VT2014
Version 0.1

Status

Granskad	Av vem	Datum
Godkänd	Av vem	Datum



PROJEKTIDENTITET

Grupp 8, 2014/VT, MapMaster2001
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Jens Edhammer	Dokumentansvarig (DOK)	076-030 67 80	jened502@student.liu.se
Erik Ekelund	Designansvarig (DES)	073-682 43 06	eriek984@student.liu.se
David Habrman		976-017 71 15	davha227@student.liu.se
Tobias Grundström	Testansvarig (TES)	073-830 44 45	tobgr602@student.liu.se
Hans-Filip Elo		073-385 22 32	hanel742@student.liu.se
Niklas Ericson	Projektledare (PL)	073-052 27 05	niker917@student.liu.se

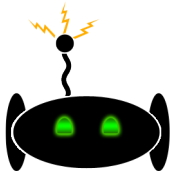
E-postlista för hela gruppen: mapmaster2001@cyd.liu.se

Kund: Mattias Krysander, Linköpings Universitet, 581 83 LINKÖPING,
013-28 21 98, matkr@isy.liu.se

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

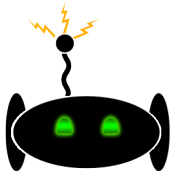
Kursansvarig: Tomas Svensson, 3B:528, 013 28 21 59, tomass@isy.liu.se

Handledare: Peter Johansson, 013-28 1345 peter.a.johansson@liu.se



Innehåll

1	Kommunikationsmodul	1
1.1	Kommunikationsfall	1
1.2	Komponenter	1
1.3	Buss	1
1.3.1	Styrord	1
1.3.2	Ack-paketets struktur	3
1.4	LCD-Display	4
1.4.1	Uppstart	4
1.4.2	Skrivning	6
1.5	Blåtand	6
1.6	Switchar	7
2	Styrmodul	8
2.1	Komponenter	8
2.2	Reglering	10
2.2.1	styralgoritmer	10
2.2.2	PD-reglering av styrning	10
2.2.3	kartabstraktion	10
2.3	kartläggningssalgoritmer	10
2.4	Brandhårdssökning	10
2.5	Kommunikation	10
3	Sensormodul	10
3.1	Komponenter	10
3.2	Kod	10
4	Saker som kan vara bra att ha!	10
4.1	Punktlista	10
4.2	Ekvationer	11
4.3	Bilder	11



1 Kommunikationsmodul

Modul för att hantera kommunikation mellan robotens olika delkomponenter samt med persondatorn via Blåtand. Kommunikationsmodulen som syns i figur 3, kommer att agera master på robotens interna buss. Vid kommunikation mellan övriga moduler, dvs. sensor- och styrmodulen, kommer denna gå via kommunikationsmodulen. Kommunikationsmodulen kommer alltså att leverera sensordata mellan roboten och mjukvaran som körs på persondatorn.

1.1 Kommunikationsfall

Ett par exempel på kommunikationsfall.

Kommunikationsmodulen skickar data mellan de olika enheterna. Ett par olika kommunikationsfall demonstreras nedan i flödesdiagrammet i respektive figurer.

Fall 1: Kommunikationsmodulen skickar manuella styrkommandon till styrmodulen, Fall 2: Sensormodulen signalerar att ny sensordata är redo, se Fall 3: Kartdata skickas från kommunikationsmodulen till PC, se Fall 4: PC signalerar nödstop, se figur 1.

1.2 Komponenter

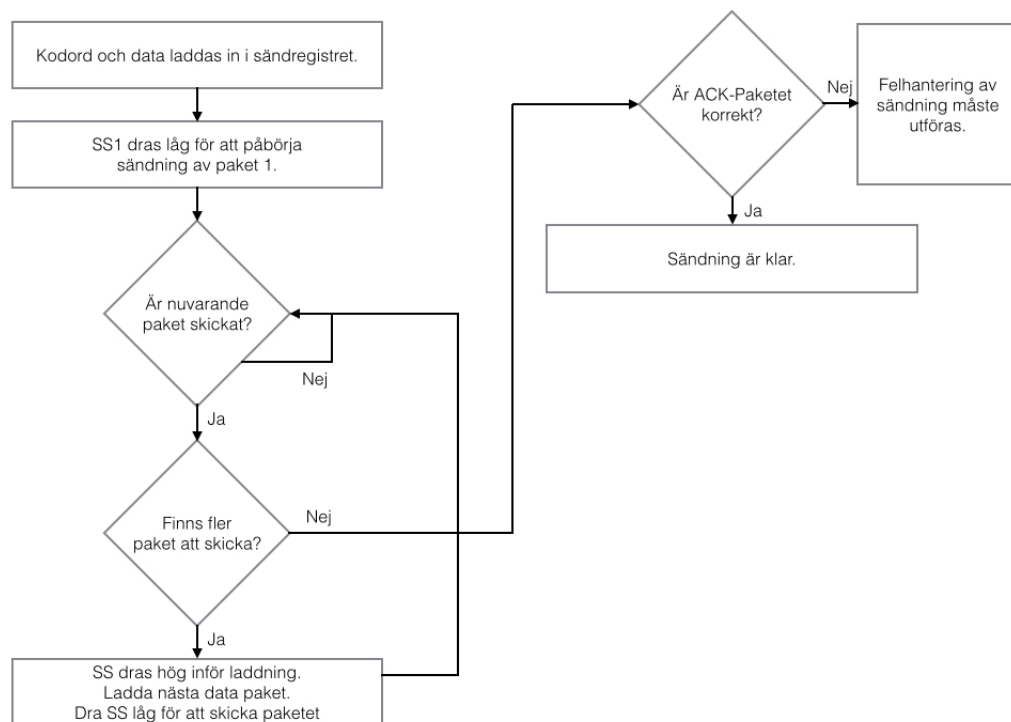
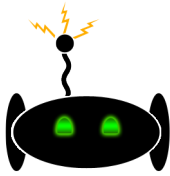
- AVR processor av typen ATMega1284
- Blåtandsdongel, Firefly, BlueSmirf Gold
- LCD-Display
- Två stycken knappar

1.3 Buss

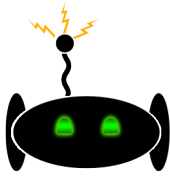
Mellan processorerna kommer SPI-kommunikation att användas. Kommunikationsmodulen agerar master medan styr- och sensormodul agerar slave.

1.3.1 Styrord

Kommunikation kommer starta med att ett styrord skickas. Ett styrord kommer bestå av ett 8-bitars paket (xxxxyyyy). De fyra mest signifikanta bitarna står för kommandot medan de 4 minst signifikanta bitarna står för antalet byte som kommer att följa som



Figur 1: Flödesdiagram för Fall 1.



argument till kommandot. I tabellerna nedan visas detta mer utförligt.

xxxx	Kommandonr.
0000	1
0001	2
...	...
1111	17

yyyy	Antal bytes data
0000	1
0001	2
...	...
1111	17

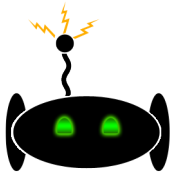
Detta betyder alltså att master kan skicka 16 unika kommandon till respektive slave och varje slave kan skicka 16 unika kommandon till master. Kommandon kan följas av ett upp till och med 17 bytes av data. I händelsen att ett meddelande inte behöver skicka data så skickas en byte nollor.

Kommandon kommer att kunna betyda olika på de olika processorerna. 0001 kan t ex på slave1 betyda åk rakt längs väggen, på slave2 betyda läs av sensor 3, och på master uppdatera kartabstraktion med följande värden. Om likadana kommandon finns på flera processorer så ska de ha samma kodord på alla processorer som delar kommandot. Ett kommando är reserverat på båda slaves för ett kommando från master som säger du kan skicka information nu. En slave kommer också att kunna skicka ett avbrott till Master för att be om kommunikation. En lista av de olika kodorden finns i appendix.

På både mottagar- och sändarsida kommer en buffer i form av en länkad lista finnas. Denna lista fylls med data som tas emot och läses från när data ska skickas.

1.3.2 Ack-paketets struktur

Ett sista paket som följer direkt efter datapaketet kommer att skickas. Detta paket är ack-paketet och kommer vara additionen av samtliga datapaketets lägsta bit. Detta ger mottagarsidan en möjlighet att kontrollera att datapaketet mottogs korrekt. Både sändaren och mottagaren skickar ett sådant ack-paket samtidigt, detta ger oss möjlighet att göra om sändningen direkt, utan att lämna ett eventuellt avbrott. Försök att skicka data utförs tre gånger, om det fortfarande inte lyckats skicka korrekt data, avbryts försöken och lämnas över till felhantering.



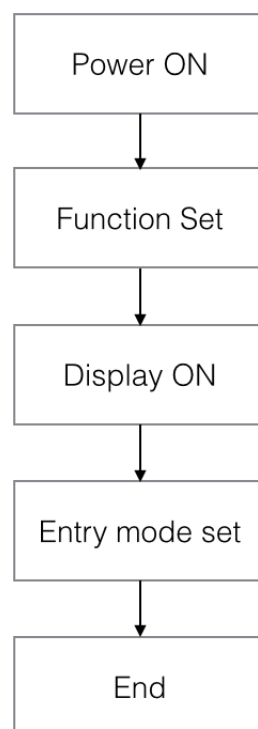
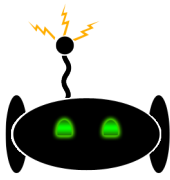
1.4 LCD-Display

Sensordata och annan information från roboten kommer att visas på en alfanumerisk-display, närmare bestämt LCD JM162A. Displayen kommer att kunna visa 2 rader \times 16 tecken. Displayen kommer anslutas till processorn enligt kretschemat i appendix ?. Vid varje uppstart kommer initiering av LCD-display göras och då följa nedanstående flödesschema. Varje steg startas med att processorn skickar ett kommando till ingångarna på LCD-displayen. Kommandon finns specificerade i databladet, se (fotnot <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/jm162a.pdf>).

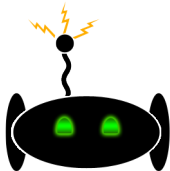
1.4.1 Uppstart

Vid uppstart av systemet kommer LCD-displayen att initieras enligt flödesschemat 2

- Power ON - sätter på strömförsörjningen
- Function set - Sätter överfördatalängden till 8 bitar och displayläget till 2-rader.
- Display ON - slår på displayen och slår på markören.
- Entry mode set - sätter markörens riktning vid skrivning
- End - Slut på initieringen



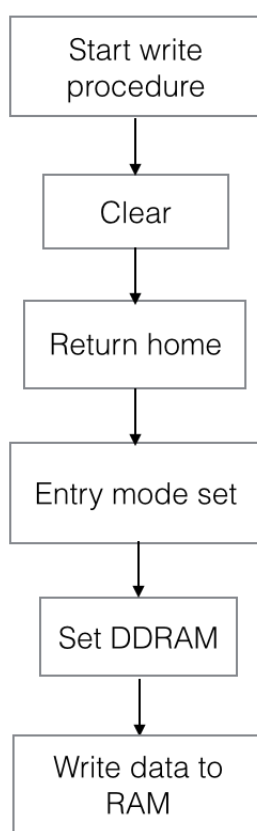
Figur 2: Uppstart av LCD-display



1.4.2 Skrivning

Vid skrivning till LCD-displayen kommer detta ske enligt flödesschemat 4

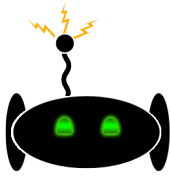
- Start write procedure - Startar skrivning till LCD-displayen
- Clear - Rensar hela displayen
- Set DDRAM - gör DDRAMet tillgängligt
- Write data to RAM - Skrivning av data in till RAM ifrån DDRAM



Figur 3: Skrivning till LCD-display

1.5 Blåtand

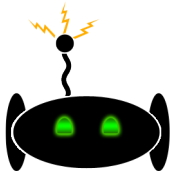
Blåtandskommunikationen kommer att utföras av ett Firefly-Bluesmirf gold (FBG) modem som parkopplas mot en persondator. FBG kommer att skicka information till persondator via protokollet RS232 enligt vad som är angivet på Vanheden. Se schema i



appendixet Scheman för anslutning av TxD och RxD, vilka sköter sändning och mottagning av data.

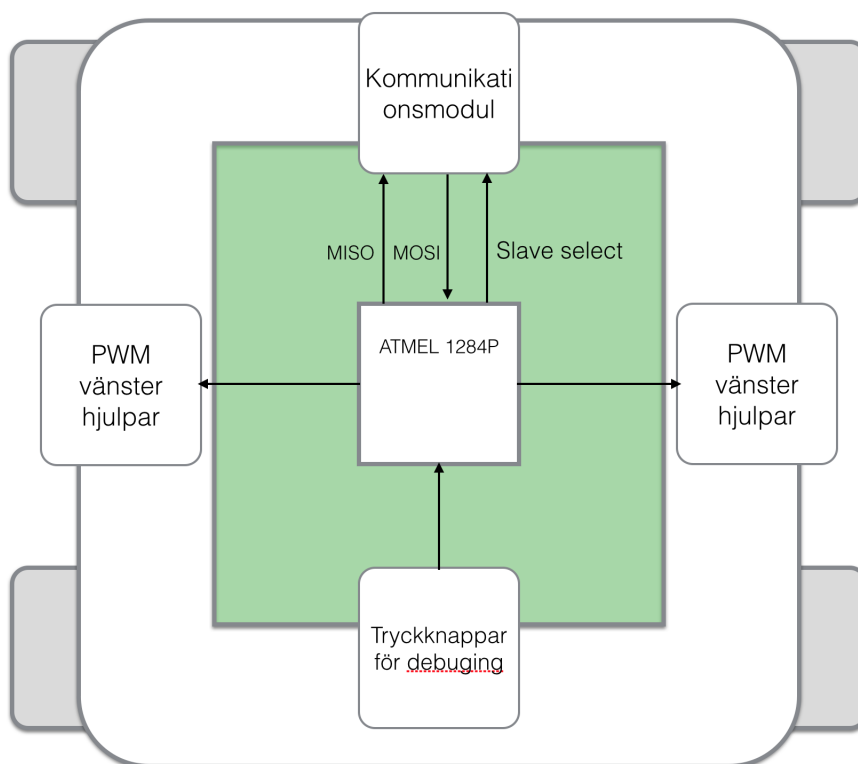
1.6 Switchar

En switch ska styra om roboten exekverar programkod för autonom styrning eller manuell styrning. Ett avbrott signalerar till processorn att den ska byta programkod, avbrottet ligger på pin16 (INT0) se appendix A. Av/På-knappen styr strömförsörjningen till samtliga moduler.



2 Styrmodul

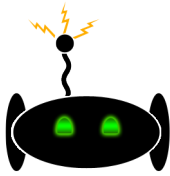
Styrmodulens (slave) huvudkomponent kommer att bestå av en AVR-processor, Atmel ATmega1284p som får instruktioner från kommunikationsmodulen (master) via bussen. Processorn kommunicerar även med drivkretsen på chassit. Chassit är av typen Terminator som drivs av en NiMH- eller NiCd-ackumulator på 7.2V. Det har även fyra stycken växlade DC-motorer som är kopplade till varsitt drivhjul.



Figur 4: Översikt styrmodul

2.1 Komponenter

- Chassi
- Systemchip - Atmel ATmega1284p
- Tryckknapp för debugging av styrfunktionalitet
- Batteri, 7.2 V
- Resetknapp



Motorerna kommer att styras parvis med två signaler per sida:

- DURL - Styr den vänstra motorns rotationsriktning
- DIRR - Styr den högra motorns rotationsriktning
- PWML - Pulsbreddmodulerad signal som styr den vänstra motorns hastighet
- PWMR - Pulsbreddmodulerad signal som styr den högra motorns hastighet.

Dessa kopplas direkt från processorn till chassits drivkrets enligt bild. Styrmodulen kommer att vara ansluten till bussen med hjälp av MISO- och MOSI-pinnarna.

- MISO - Master Input Slave Output, data skickas till master
- MOSI - Master Output Slave Input, data skickas till slave
- MASTERINT - Skickar avbrott till master
- SLAVESELECT - Väljer denna processor som aktiv slav
- SCK - Serial Clock, klocka från master

Utöver detta kommer en tryckknapp anslutas till RESET för att förhindra att felaktig programmering av processorn.

För fullständigt kopplingsschema - se Appendix.

2.2 Reglering

2.2.1 styralgoritmer

Det ska implementeras två olika avsökningsalgoritmer. De kommer användas beroende på vilket uppdrag roboten ska genomföra. Algoritmerna kommer att skrivas med hjälp av C++ och köras från processorns interna minne.

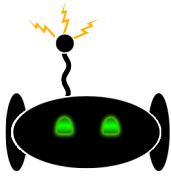
2.2.2 PD-reglering av styrning

Styrmodulen kommer att få sensordata skickad till sig från sensormodulen. Denna sensordata kommer att användas till att PD-reglera riktningen då man följer en vägg.

Den proportionella delen av regleringen kommer fås från avståndet till vägg vid sidan av roboten.

$$ep = \textit{Konstant} * \textit{avstndssensordata}$$

Den deriverande delen av regleringen kommer ges av linjär uppskattning av derivatan utefter de senaste mätvärdena. Derivatan ges alltså av:



$$ed = \textit{Konstant} * (\textit{data} / \textit{antalsampels})$$

Totala felet ges då av:

$$e = ed + ep$$

2.2.3 kartabstraktion

2.3 kartläggningssalgoritmen

2.4 Brandhårdssökning

2.5 Kommunikation

3 Sensormodul

3.1 Komponenter

3.2 Kod

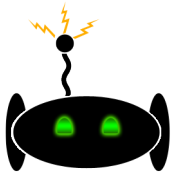
Ibland vill man skriva kod, då kan man skriva i annan font. Förslagsvis:

```
1
2 for (int i=0; i<iterations;i++)
3 {
4 do something hej hej
5 }
```

4 Saker som kan vara bra att ha!

4.1 Punktlista

- Detta är en punktlista!
- Punktlistor kan innehålla matte. $T_R \leq 10\%$

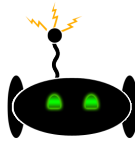


4.2 Ekvationer

som vi ser i ekvation 1

$$F(s) = K \frac{\tau_D + 1}{\beta\tau + 1} \frac{\tau_I + 1}{\tau_I + \gamma} \quad (1)$$

4.3 Bilder



Figur 5: Våran logotyp