Teknisk dokumentation

kamprobot

Johan Olin

Version 1.0

Status

|  |  |  |
| --- | --- | --- |
| Granskad |  |  |
| Godkänd |  |  |

PROJEKTIDENTITET

Grupp 15, HT1-2015, Mr.Robot  
Linköpings tekniska högskola, ISY

|  |  |  |  |
| --- | --- | --- | --- |
| **Namn** | **Ansvar** | **Telefon** | **E-post** |
| Johan Olin | Projektledare (PL) | 072-3055650 | johol009@student.liu.se |
| Mattias Ulmstedt |  | 070-1454123 | matul773[@student.liu.se](mailto:eabs222@student.liu.se) |
| Per Olin |  | 072-3055440 | perol834@student.liu.se |
| Hans Tchou |  | 070-0535246 | hantc350[@student.liu.se](mailto:dords444@student.liu.se) |
| Tor Utterborn | Dokumentansvarig (DA) | 076-8821415 | torut235[@student.liu.se](mailto:eries555@student.liu.se) |
| Joacim Stålberg |  | 073-9400950 | joast229[@student.liu.se](mailto:frefs666@student.liu.se) |

**E-postlista för hela gruppen**: johol009@student.liu.se

**Kund:** Tomas Svensson  
**Kundtelefon** +46 (0)13 28 1368, [tomass@isy.liu.se](mailto:tomass@isy.liu.se)

**Kursansvarig**: Tomas Svensson, 3B:528, +46 (0)13 28 1368, tomass@isy.liu.se  
**Handledare:** Olov Andersson

# Sammanfattning

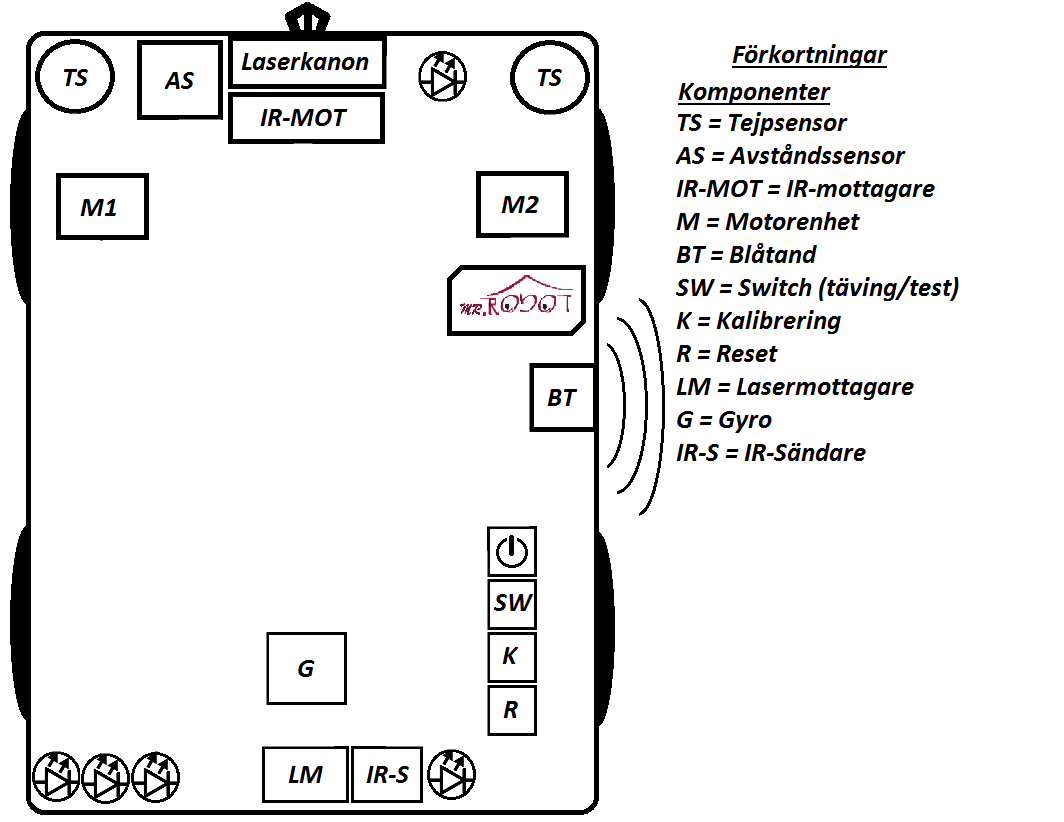
Mr.Robot är ett autonom kamprobot projekt som givits inom KMM-kursen TSEA29. Projektet har utförts av 6 datortekniska studenter som sammanlagt har arbetat 960 timmar på att bygga och programmera. Det som tas upp här är de tekniska detaljerna i roboten. Roboten är uppbyggd av tre moduler där de tre modulerna är:

* Målsökningsenheten
* Styrenheten
* Sensorenheten

Sensorenheten tar in mätvärden med dess ingående sensorer som sedan skickas vidare till målsökningsenheten för bearbetning av data. Målsökningsenheten tar beslut beroende på indata och skickar vidare instruktioner till styrenheten samt data till persondatorn. För att överföra data mellan enheterna så använder vi oss av överföringsmetoden UART(*universial asynchronous receiver/transmitter*). Styrenheten utför instruktionen enligt beslut från målsökningsenheten. Persondatorn visar önskad mätdata på skärmen i en GUI.

De ingående sensorerna och deras placering är enligt följande:   
  
Två tejpsensorer placerade i robotens främre hörn, två ultraljudssensorer placerade fram och bak, en laserdetektor där bak, IR-sensor fram och en gyromodul placerad i robotens mitt med hänseende på y-axeln (vertikal axeln sedd uppifrån i figur 0).

De ingående ställdonen och position är enligt följande: En IR-sändare bak, lasersändare fram, motorer som är inbyggda i chassit för att styra hjulen och till sist knappar samt LEDs vars position visas i figuren nedan och vars funktioner förklaras senare i dokumentet.



INNEHÅLLSFÖRTECKNING HÄR

Dokumenthistorik

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Datum | Utförda förändringar | Utförda av | Granskad |
| 0.1 | 2015-11-20 | Teknisk dokumentation – start | Samtliga gruppmedlemmar | Samtliga gruppmedlemmar |

# Inledning

Detta dokument innehåller en sammanfattning av en autonom kamprobot samt dess tekniska lösningar och strategi som del av examinationsmoment i projektkursen KMM(TSEA29 år 2015) på Linköpings universitet. Alla programkod som tillhör projektet bifogas längst bak i dokumentet.

## Parter

Kunden är Tomas Svensson och produkten levereras av grupp15 bestående av 6 gruppmedlemmar från årskurs 3 i D-programmet på Linköpings universitet.

## Syfte och mål

Syftet i och med projektet var att konstruera en robot från en liten början till en helhet som vi även programmerar in en AI(*artificial intelligence*), vilken tillåter att roboten på ett logiskt sätt kan navigera sig runt i en bana och iaktta handlingar. Roboten är speciellt anpassad för kamp mot andra kamprobotar. Operatören kan välja mellan två olika lägen på roboten; tävlingsläge där målet är att eliminera andra kamprobot eller testläge. Banan begränsas med svart eltejp.

Det stora målet är att lära oss att bygga en robot från scratch.

## Begränsingar

Vi har en maximal tidsbudget på 960 timmar sammanlagt till att utföra projektet.

## Definitioner

GUI – Graphical user interface  
PWM – Pulse-width modulator  
USB – Universal serial bus  
UART – Universial asynchronous receiver/transmitter  
AI – Atrificial intelligence  
PC – Personal computer

IR – Infraröd

LED – Light emitting diode

# Produkt

Produkten är en autonom kamprobot som tävlar mot andra kamprobotar. En autonom kamprobot är en robot som kan åka runt på en specifik bana.

## Grov beskrivning av produkten

Roboten är utrustad med olika sensormoduler som hjälper roboten att hitta och föra sig fram på banan. Roboten är även utrustad med lysdioder för att visuellt visa info om robotens tillstånd. Till roboten tillkommer mjukvara som visar robotens mätvärden. Dessutom har roboten ett reglage samt tre knappar. Reglaget ställer in vilket läge den ska befinna sig i: test- eller tävlingsläge. De tre knapparna är till för att aktivera, starta om samt kalibrering av sensorer.

## Produktkomponenter

* En kamprobot
* Medhörande mjukvara till kamprobot
* Användarmanual
* Teknisk dokumentation

Gränssnittet är implementerad i Java och samtliga enheter är programmerade i C++ kod i Atmel studio.

## Användning

Roboten används i en tävling mot andra autonoma kamprobotar. För vidare beskrivning av tävling och regler, se Appendix.A.

# Teori

Roboten är uppbyggt av tre olika enheter. Sensorenheten tar bara in rådata och skickar behandlad data till målsökningsenheten där den behandlar data vidare för att utföra alla beräkningar för att sedan skicka order vidare till styrenheten. När styrenheten får order från målsökningsenheten så skickas order till den modul som ska användas. Om den ska backa så får servomotorerna order från styrenheten.

T.ex. om roboten åker över en tejp så skulle sensorenheten tala om för målsökningen, i målsökningen så får vi absolut inte åka över tejp. Målsökningsenheten säger då till styrenheten att backa och sedan rotera.

Eftersom att all data skickas enkelriktad så…

Ange AI logik här etc.

Det är viktigt att roboten kan prioritera vissa funktioner så som t.ex. att gå osynlig direkt efter träff, så vi har utgått från det.

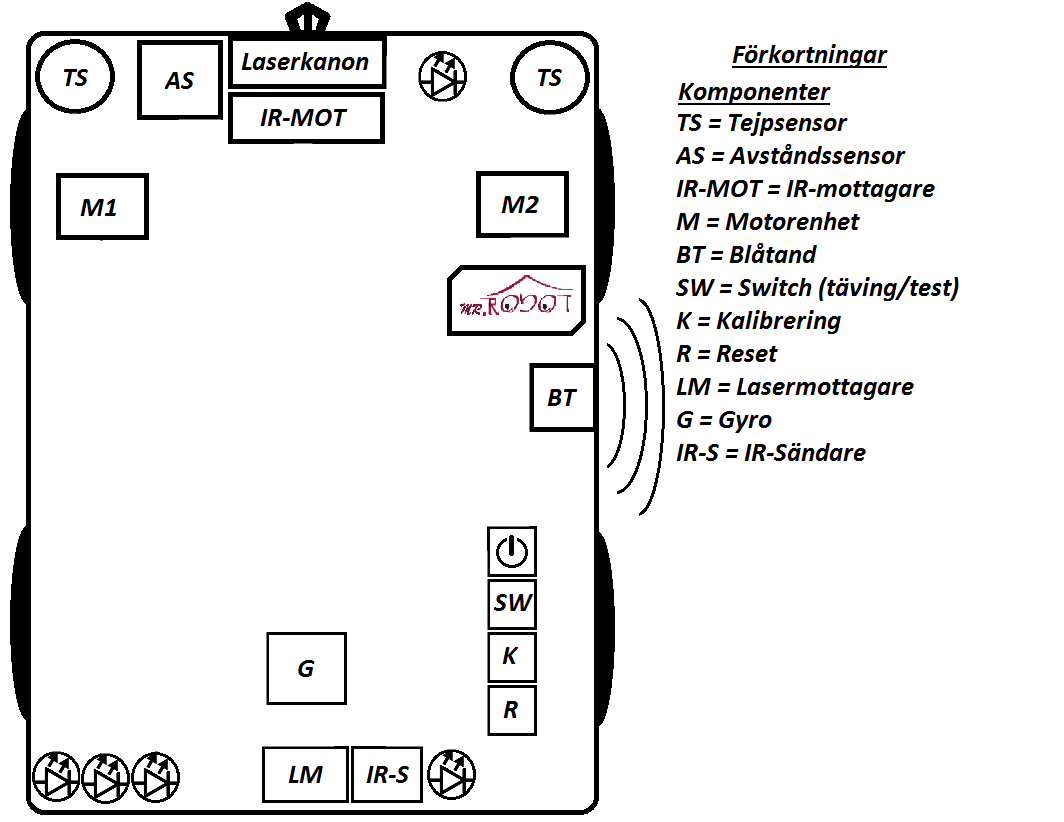
# Översikt av systemet

Systemet är uppbyggt av tre enheter: sensorenhet, målsökningsenhet och styrenhet. Vi har ett grafiskt användargränssnitt GUI, som visar sensordata och knappar som reglerar mellan användsingslägen. GUI har implementerats i Java. Nedan visar en översiktlig bild av hela systemet.

# C:\Users\hantc350\Desktop\TSEA29 bilder uppdaterad\simpelbildöversystemet.png

*Figur x. Denna bild visar en översikt av systemet, pilarna indikerar skickad data från modul.*

De ingående sensorerna och deras placering är enligt figur x.



*Figur x. Bilden visar systemet i sin helhet.*

# Gränssnittet mellan enheter

Mellan enheterna kopplar vi direkt vägar för kommunikation med UART. Vi Använder bluetooth, en standard för trådlös kommunikation för att kommunicera mellan robot och persondatorn, och protokollet som används där är även UART.

## UART

UART är ett protokoll för hur enheter kommunicerar. UART arbetar seriellt med startbit och stoppbit.

AVR-processorerna som används har portar för att skicka (TX) och ta emot (RX) data över UART. För att använda dessa så laddas vissa register med parametrar för överföringen, till exempel BAUD-rate, antal data- och stoppbitar mm. Här har vi använt oss av UART med olika parametrar.

### Parametrar

Data skickas asynkront mellan sensorenheten och målsökningsenheten enligt dessa specifikationer:

* BAUD-rate: 115200
* Databitar: 8
* Stoppbitar: 1
* Paritet: ingen
* Dubbelspeed: nej

Data skickas asynkront mellan målsökningsenheten och sstyrenheten enligt dessa specifikationer:

* BAUD-rate: 115200
* Databitar: 8
* Stoppbitar: 1
* Paritet: ingen
* Dubbelspeed: nej

### Vad ska skickas

**Sensorenhet → Målsökningsenhet:**

Meddelande 1 skickas med 200 Hz och meddelande 2-3 i 50 Hz eftersom att tejpsensorerna är mest kritiska. I maxhastighet (1 m/s) med mätningar i 200 Hz så mäts med intervallet 5 mm (och tejpen är 14-18 mm bred).

Meddelande 1:

Bit 0-1: Meddelande ID (00)

Bit 2-6: Främre avståndssensorn (ca 1 dm precision)

Bit 7: Tejpsensor 1 (vänster, 1 för tejp)

Bit 8: Tejpsensor 2 (höger, 1 för tejp)

Meddelande 2:

Bit 0-1: Meddelande ID (01)

Bit 2-6: Bakre avståndssensorn (ca 1 dm precision)

Meddelande 3:

Bit 0-1: Meddelande ID (10)

Bit 2: Aktiv IR-signatur (robot framför oss)

Bit 3-5: IR-signaturen

Bit 6: Laser (1 för träff)

Bit 7-8: 2 MSB Gyro (grader rotatation)

Meddelande 4:

Bit 0-1: Meddelande ID (11)

Bit 2-8: 6 LSB Gyro (grader rotatation)

**Målsökningsenhet → Styrenhet/PC:**

Bit 0-8: OP-kod för instruktion

## Bluetooth

Roboten skickar data till en persondator(PC) med hjälp av Bluetooth. Roboten använder sig av Firefly-modulen och persondatorn använder sig av en Bluetooth-pinne för att skapa en Bluetooth länk mellan dem. Bluetooth-modulen använder RS232-kommunikation för att kunna ansluta via Firefly till persondatorn. För ytterligare information om Bluetooth, se databladen i referenser.

### Parametrar

Data skickas via Bluetooth enligt dessa specifikationer:

* BAUD/BPS: 115200
* Databitar: 8
* Paritet: N (ingen paritets bit)
* Stoppbitar: 1

### Aktivering av virtuell länk

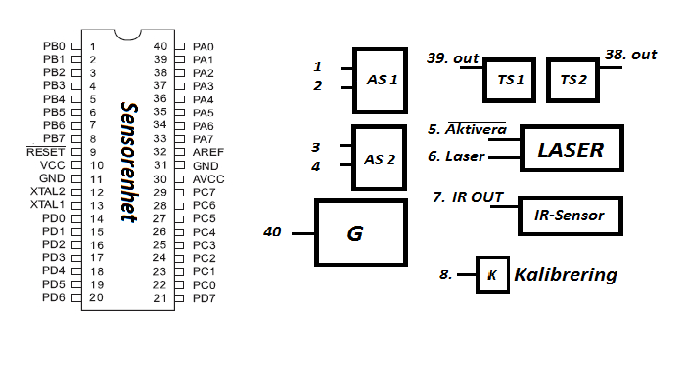
Vi följer hänvisningarna på Vanhedens hemsida om [Bluetooth](https://docs.isy.liu.se/twiki/bin/view/VanHeden/BlueTooth).

# Sensorenhet

Sensormodulen är den enhet som sköter A-D omvandlingar från sensorvärden och skickar vidare dessa till nästa enhet. I figur 2 kan man se de ingående sensorerna. Linjesensorn används för att hålla roboten inom banans gränser.

Vi placerar avståndssensorerna fram och bak för att ta reda på om eventuella hinder befinner sig framför eller möjliga fiender bakom. IR-sensorn ska känna av motståndarnas IR signaturer (placeras fram på roboten enligt Appendix A i kravspecifikationen). Lasersensorn känner av träff från motståndare. Gyron mäter vinkelfrekvensen som integreras för att få reda på hur många grader roboten har roterat. Sensorerna skickar data i mellan 100-1000 Hz.

I sensorenheten ingår ett antal sensorer och en processor. Som nämnt så sker kommunikationen till nästa enhet med UART. Processorns uppgift är i princip bara att omvandla analoga signaler till digitala signaler och skicka vidare dessa. Sensordata omvandlas till lämpliga värden.. Figur 3 nedan visar kopplingsschema för sensorenheten.

*Figur 3. Kopplingsschema för sensorenheten.*

## Komponenter

Följande komponenter relateras till sensorenheten:

* Processor (AVR ATmega 1284P)
* Tejpsensor x 2
* Avståndsensor
* IR-Sensor
* Gyro
* Lasersensor

För att se dessa komponenters monetringsplacering, se figur x.

### 

### Tejpsensorer

Tejpsensorerna är fastfästa vid de främre hörnen, en sensor i vardera hörn som känner av om en svart eltejp befinner sig under den. Dessa sensorer ska monteras så att de går strax en bit utanför roboten enligt figur 4 på så sätt att tejpen registreras innan det är för sent och roboten redan har kört över linjen. För mer information om tejpsensorn, se datablad i referenser.

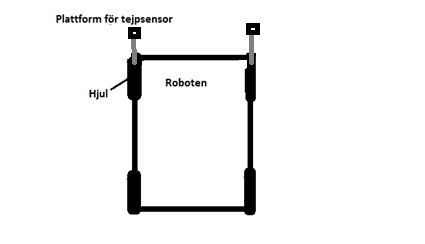
Tejpsensorerna har 3 pinnar:

* +5: kopplas till 5V matningsspänning
* GND: kopplas till jord
* OUT: modulens analoga utsignal

Utsignalen ger en analog spänning mellan 0-5V beroende på hur mycket ljus som reflekteras på underlaget. Om underlaget reflekterar lite ljus (som i fallet för svart eltejp) så blir utsignalen hög, och om underlaget reflekterar mycket ljus så blir utsignalen låg. För användningsområdet att detektera svart tejp bryr vi oss inte om exakt hur svart tejpen är, utan vi omvandlar utsignalen till en digital bit:

* 1 då tejp är under.
* 0 då ingen tejp under.

För att sensorn ska fungera bra så bra som möjligt så måste den vara monterad så att den sitter nära marken, ungefär bara några millimeter ifrån. För att kalibrera sensorerna till vad som räknas som tejp och inte så ska sensorerna placeras så att båda är rakt över en tejpbit, och sedan trycks en kalibreringsknapp. Vi sparar sedan detta analoga värde, och när sensorenheten sedan läser av sensorn för att avgöra om den befinner sig över en tejpbit eller inte så jämförs utsignalen med detta sparade värde med viss felmarginal. Det är bättre att kolla efter tejp än att kolla efter golv, eftersom att tejpen alltid kommer att vara samma färg. Figur x nedan visar en bild på var tejpsensorerna är monterade på roboten.



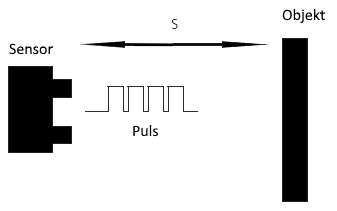
Figur x. Tejpsensorerna monteras enlig figur.

### Avståndsensor - SRF04 Ultrasonic Range Finder

Ultraljudssensorn används för att mäta avståndet mellan objekt framför roboten. För att beräkna avstånd mellan sensor och robot använder vi formeln 1. För mer information om avståndsensorn, se datablad i referenser.

*Formel 1. Formeln beräknar sträckan mellan sensor och robot. S är sträckan(m), 170 är ljudets hastighet dividerat med två och t är tiden(s).*

Sensor skickar ut en puls som färdas sträckan 2S eftersom pulsen studsar tillbaka när den träffar ett objekt. När pulsen kommer tillbaka, kommer tiden det tog för pulsen att färdas fram och tillbaka beräknas, se Figur 5.

**

*Figur 5. Bilden visar hur sensorn fungerar*

### IR-sensor – IRM-8601S

IR-sensorn sitter på robotens främre del där den har som uppgift att upptäcka andra kamprobotar. Sensorn detekterar IR-ljusvågor som andra robotar sänder ut från sina IR-sändare eller som också kallas ”IR-fyr” och på så sätt identifieras motståndaren framför. För mer information om IR-sensorn, se datablad i referenser.

IR-sensormodulen har dessa följande tre pinnar:

* VCC: kopplas till 5V matningsspänning
* GND: kopplas till jord
* OUT: modulens digitala utsignal

IR-sensor tar in digitala vågor som sänds från en IR-sändare. IR-sensorn är programmerad så att den kan avläsa en startpunkt och avgöra om det är en lång eller kort våg samt om det är en fiende framför. För att tolka vår egen signatur som ”vänlig” så måste IR-sensorn känna igen IR-signaturen som vår egen IR-sändare skickar. Detta görs genom att spara vår förvalda signatur i en variabel och fiender detekteras genom att jämföra vår signatur med den signatur som upptäcks.

En signatur innehåller tre bitar där varje bit bestäms genom detektion av en lång hög aktiv signal på 1.2ms eller kort aktiv signal på 0.6ms.

Eftersom att IR-sensorn inte kan ta emot kontinuerliga ljuspulståg för att avkodas direkt, så varierar insignaler mellan av och på med en viss period.

### GYRO – MLX90609 Angular Rate Sensor

Gyrot används för att beräkna om roboten roterat tillräckligt för att nå sin uttänkta rotation. Gyrot kan som maximalt mäta upp till 300 graders rotation per sekund, där 0.5V betyder 300 grader per sekund motsols och 4.5v betyder 300 grader per sekund medsols. 2.5V betyder att det inte är någon rotation. Med hjälp av de intervallen, 0.5 – 2.5 och 2.5 – 4.5 beräknas robotens nuvarande rotationshastighet. För mer information om gyrot, se datablad i referenser.

### Lasersensor

Lasersensormodulen kommer att vara monterad längst bak i mitten av roboten, och dess syfte är att detektera träff av en laser från en annan kamprobot. När den blir träffad så ska den iaktta en funktion som påvisar en träff. För mer information om lasersensorn, se datablad i referenser.

Lasersensormodulen har följande pinnar:

* VCC: kopplas till 5V
* 3x GND: minst 2st bör kopplas till jord
* LASER: utgång som ger logisk 1:a vid detektering av laser
* AKTIVERA: negativ flank aktiverar laserdetektorn

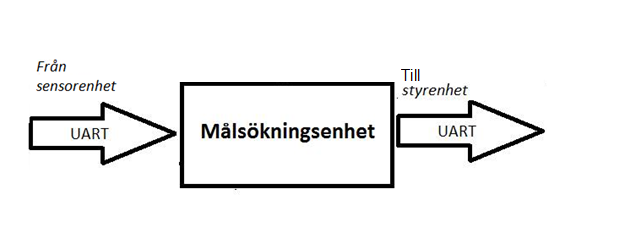
Efter spänningstillslag samt efter varje detektering av laser så ges signalen *AKTIVERA* en fallande flank för att återaktivera detekteringen. Förutom att detektera laser kan modulen även användas som en IR-fyr, men det kommer inte att användas i det här fallet.

## ADC – Analog to Digital Conversion

För att ta reda på det digitala värdet av en analog signal så jämför man två intervall. Det ena intervallet är 0V – 5V, det analoga intervallet och det andra är 0 – 255 eller 0- 1023 (beroende på om man använder 8 eller 10 bitar). 0V motsvarar 0 i det digitala intervallet och 5V motsvarar 255 eller 1023 i det digitala intervallet.

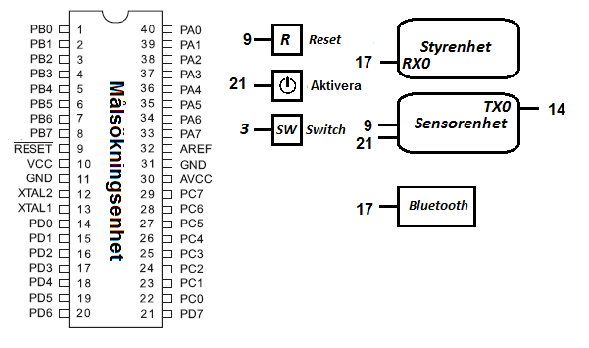
# Målsökningsenhet

Målsökningsenheten kan ses som ”hjärnan” i roboten där huvudprogrammet existerar. Algoritmen för AI samt funktioner för att skicka ut data till andra enheter är inprogrammerad här. I figur 6 beskrivs i stora drag dataflödet. Data skickas med UART till både laptop och styrenhet. Kommunikationen sker direkt från och till de andra enheterna och till persondatorn sker detta via Bluetooth.



Figur 6. Blockschema över målsökningsenheten.

Målsökningsenheten tar emot data från sensorenheten och utifrån den gör logiska val för vilka instruktioner som vill skickas till styrenheten. Beroende på vilket läge (test-/tävlingsläge) roboten befinner sig i ska den ha ett annorlunda beteende. Figur 7 nedan visar kopplingsschemat för målsökningsenheten.



*Figur 7. Kopplingsschema för målsökningsenheten.*

## Komponenter

Följande komponenter relateras till målsökningsenheten:

* Processor (AVR ATmega 1284P)
* Blåtandsmodul
* Aktiveringsknapp
* Resetknapp
* Switch (Tävling/Test)
* Kalibreringsknapp

### Aktiveringsknapp

När roboten får spänning så väntar den på att aktiveringsknappen blir nedtryckt. Fram tills att detta händer så sitter programmet i en oändlig loop som inte gör någonting, precis innan main-loopen. Denna aktiveringsknapp är kopplad till alla tre processorerna, vilket gör att alla kommer att starta sina program samtidigt.

while(ej aktiverad){

do nothing

}

Mainloop{

…

…

}

### Resetknapp

När resetknappen, som är kopplad till alla tre processorerna, trycks ner kommer programmen att hoppa in i en rutin för att nollställa alla variabler, register mm. Efter detta så hoppar de upp till loopen innan main loopen i väntan på återaktivering av aktiveringsknappen.

### 

### Switch(Tävling/Test)

Roboten har en switch som byter mellan dem olika lägena; tävlings- och test läge.

### Kalibreringsknapp

När kalibreringsknappen aktiveras så kalibreras tejpsensorerna.

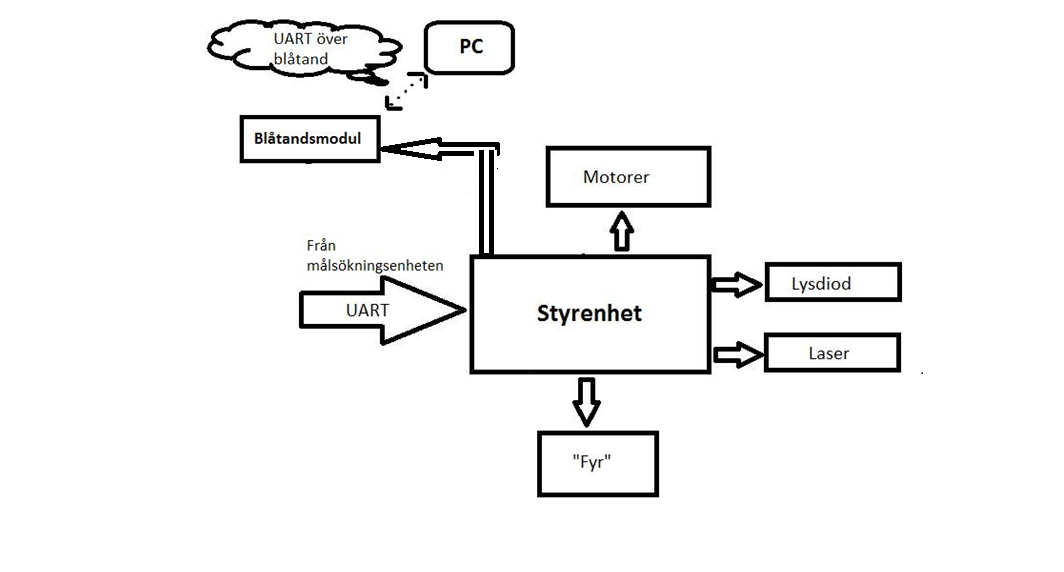
## AI

Skillnaden mellan tävlings- och testkoden är att roboten kan dö i tävlingsläget. Detta medför att i tävlingsläget så rör sig roboten med det i åtanke och försöker då att undvika att bli beskjuten, medan i testläget behöver roboten inte oroa sig om detta. Denna skillnad gör att roboten rör sig annorlunda i test- och tävlingsläget. Se pseudokod i Appendix A för tävlings- och testkod.

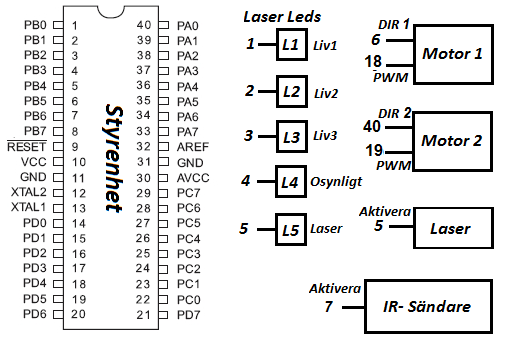
Inläsning av banan och flowchart över program?

# Styrenhet

Styrenheten är den enhet som ”väcker roboten till liv”. Den styr allt rörligt och synligt på roboten, t.ex. de olika motorerna på roboten och den får lasermodulen att skicka ut en laser samt dioderna att lysa. Den agerar efter order från målsökningsenheten och kan ses som en ”slav” till denna. Figur 8 visar blockschemat över styrenheten och figur 9 visar kopplingsschemat.



Figur 8. Blockschema över styrenheten.



*Figur 9. Kopplingschema för styrenheten.*

## Komponenter

Följande komponenter relateras till styrenheten:

* Processor (AVR ATmega 1284P)
* Motorer
* Laserkanon
* LEDS
* IR sändare

För att se dessa komponenters monetringsplacering, se figur x.

### Motorer

Robotens chassi har 4 DC-servomotorer på 7.2V med upp till 291RPM, där 2 hjul sitter på vardera motstående sida. Vid höger respektive vänster sväng roterar en av motorerna medurs och den andra moturs, på så sätt snabbas rotationen. Det går även att stanna- eller sakta ner ena motorn för att utföra mjukare svängar, men det är inte nödvändigt.

Motorerna ansluts till styrenheten med flatkablar och agerar utifrån order därifrån. För mer information om motorerna, se datablad i referenser.

Servomotorerna styrs parvis med två signaler per sida:

* DIR: styr motorernas rotationsriktning. Kopplas direkt till microcontroller
* PWM: styr motorernas hastighet. Kopplas direkt till microcontroller

Varje motor styrs av en DIR ingång samt av en PWM ingång, där DIR bestämmer dess rotationsriktning och PWM bestämmer dess rotationshastighet. För PWM ingångarna så ställs processorn in så att den skickar vågor med en bestämd duty cycle. Ju högre duty cycle desto högre rotationshastighet på motorn. För att få roboten att stå stilla, så sätts bara duty cyclen på 0.

Tabell 1 nedan visar output för de olika insignalerna DIR1 respektive DIR2.

|  |  |  |
| --- | --- | --- |
| DIR1 | DIR2 | U |
| 0 | 0 | Åk bakåt |
| 0 | 1 | Sväng vänster |
| 1 | 0 | Sväng höger |
| 1 | 1 | Åk framåt |

*Tabell 1. Robotens beteende U beroende på insignalerna DIR1 och DIR2.*

Funktioner som roboten kan utföra med servomotorerna:

* Åka rakt fram
* Svänga höger
* Svänga vänster
* Stå still
* Backa

### 

### Laserkanon

Längst fram i mitten av roboten monteras en lasermodul. Denna modul kommer att användas för att avfyra en laserstråle på måltavlor som är fienderobotarnas lasersensor i tävlingen. För mer information om laserkanonen, se datablad i referenser.

Lasermodulen har endast 2 pinnar:

* VCC: matas med 4.5V spänning när lasern ska lysa
* GND: kopplas till jord

Eftersom att processorn bara kan ge 5V så behöver vi lägga till ett motstånd mellan VCC och matningen från processorn.

### LED till lasern

Eftersom att roboten inte ska kunna spam-skjuta så monteras även en LED vid lasermodulen som tänds när lasern är aktiv och kan avfyra en laser. Samma signal som styr lasern styr alltså även denna LED.

### LED för att visa liv

Roboten har tre lysdioder som representerar antalet liv roboten har kvar. Vid fullt liv är alla tre lysdioder tända och vid träff släcks en av dioderna. Lysdioderna släcks från höger till vänster.

### LED Osynlighet

OsynlighetsLEDen lyser när roboten blir osynlig. Roboten blir osynlig då lasersensorn detekterar och

IR-sändaren inaktiveras i 5 sekunder.

### IR-sändare

IR-sändaren (IR-fyren) sitter på bakre delen av roboten där den har som uppgift att sända ut IR-ljus som en annan kamprobot ska kunna detektera för att avgöra om vi är en fiende. IR-sändaren fungerar i princip som en vanlig LED, förutom att det är infrarött ljus som emitteras istället. IR-sändaren sänder en unik signatur som är 38 KHz. Den unika signaturen väljs från de åtta olika signaturerna som finns på ”fyrarna”. För mer information om IR-sändaren, se datablad i referenser.

IR sändaren skickar upprepande 4 pulser inklusive startsignalen s.k. header där en header är en hög signal på 2.4ms samt hög signal på 1.2ms respektive 0.6ms för att avgöra signatur. En hög signalpuls på 1.2ms motsvarar 1 bit och 0.6ms motsvarar 0 bit.  
  
Den förbestämda IR signaturen som vår robot sänder är 110.

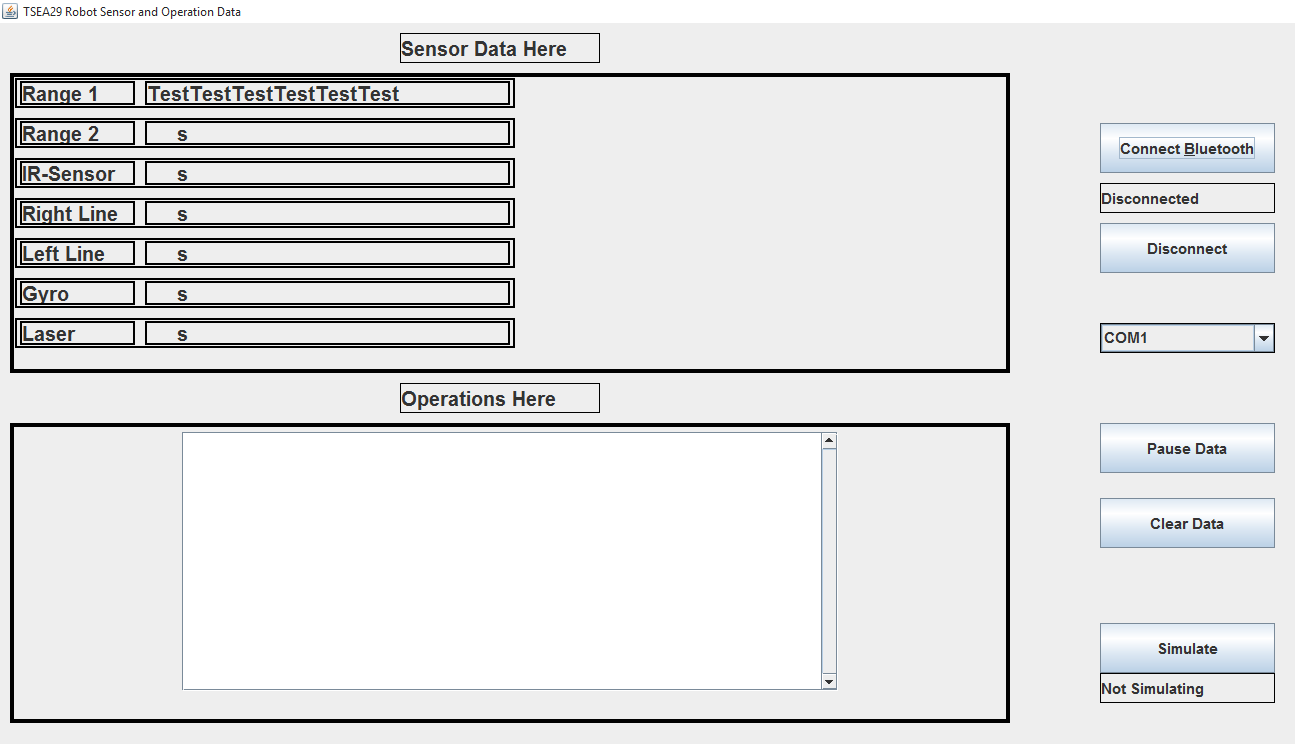
Vid träff inaktiveras IR sändaren i fem sekunder för att inte förbli synlig.

IR-sändaren har två pinnar:

* VCC: matas med viss spänning
* GND: kopplas till jord

# Programvara till laptop

Programvaran till persondatorn är skriven i Java. Programvaran tar emot data från målsökningsenheten via Bluetooth. De olika sensorernas data presenteras i en lista för att fullfölja kraven i kravspecifikationen. Den nedre delen av GUI:t visar historiken över de olika kommandon roboten har utfört. Tanken är att operationshistoriken framförallt hjälper oss att ”debugga” roboten och göra arbetet med den lättare. Figur 10 visar hur vi hade tänkt oss att det grafiska användargränssnittet skulle se ut.



*Figur 10. Skiss över vårt GUI*

# Implementationsstrategi

Konstruktionen sker inifrån och ut, d.v.s. att vi bygger varsin moduler och komponenter och sätter sedan ihop de för att tillslut få en färdig robot.

För att se till så att alla saker fungerade som vi hade tänkt så provades de innan vi satte ihop roboten. Då vi bygger roboten modul för modul är detta viktig, så man vet att alla moduler fungerar innan man gör de sista testerna med allt ihopkopplat.

Detta är något som måste ske för att vi ska kunna säga/bekräfta att en aktivitet eller milstolpe är nådd. Många av komponenterna och modulerna kunde testas enskilt utan att koppla ihop hela roboten. Detta gjorde att vi kunde bygga roboten modul för modul. De större testerna som att se till att roboten kunde åka runt i en bana eller att upptäcka en fiende och skjuta mot den kräver att flera komponenter/moduler var klara.

För att testa modulerna genomfördes tester på varje modul, vissa av modulerna behövdes flera tester samt kräva att andra moduler fungerade. Alla sensorer, laserkanon, IR-sändare, motorer, knappar, reglage samt gyro testades individuellt med egna program. Dessa tester utfördes för att se till att modulerna fungerar, men främst för att bekräfta att programmen fungerade. Dessa program fungerade liknande eller var exakt de program som vi använde vid hopkoppling och konstruktion av roboten.

Andra tester som t.ex. Bluetooth testades utan att hela systemet/konstruktionen var klar. Man kunde simulera data som skickades via UART till Bluetooth-modulen som sedan skickade data till en persondator med vår programvara på.

För att testa styrenheten, sensorenheten och målsökningsenheten gjordes enskilda tester, men man kunde inte vara helt säker på att allt skulle fungera innan man kopplat ihop hela roboten.

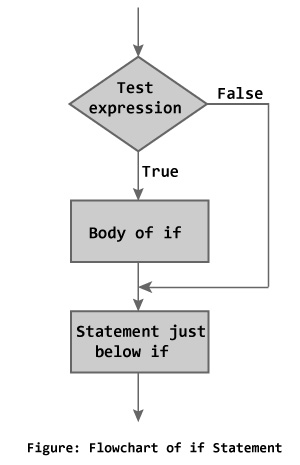
De sista testerna som skedde var att se till att roboten var redo för tävling och test. Det innebar att roboten skulle kunna åka runt i en bana och skjuta på andra robotar eller ”fyrar”. Om dessa tester fungerade så kunde vi säga att roboten var klar, redo för tävling och redovisning.

Vi skickar data från roboten till en persondator som visar alla sensorvärde samt vilka operationer som styrenheten gör, åk fram, skjut laser etc. Det finns även LEDs på roboten som lyser eller är släckta beroende på olika omständigheter, visa hur mycket liv roboten har kvar eller visa att vi skjuter etc.

Tillläg: Lägg till real bilder, hur de programmeras, kablar och verktyg. Problemlösningar och metod samt motivera varför? Vad har testats och hur? Signal behandling? Filter? Sampling? Flowchart för kod? Design? Uppkopplingen? Kvarvarande problem att lösa, eller kan uppkomma? Bifoga?

STRUKTURERA,  
sammanfattning ->  
Översikt av system samt ytterligare detalj på moduler, etc:

Regleralgoritm



Referenser

Bluetooth

[https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/firefly.pdf](https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/firefly.pdf%20%202015-10-22)  2015-10-22 kl 16:00

<https://docs.isy.liu.se/twiki/bin/view/VanHeden/RS232> 2015-10-22 kl 16:00

Tejpsensor

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/reflex_sensor.pdf> 2015-10-22 kl 16:00

Avståndssensor

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/srf04.pdf> 2015-10-22 kl 16:00

IR-sensor

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/irm-8601s.pdf> 2015-10-22 kl 16:00

Gyro

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/MLX90609_datasheet.pdf>

2015-10-22 kl 16:00

Lasersensor

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/laserdetektor_modul.pdf>

2015-10-22 kl 16:00

Motorer

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/terminator_prel.pdf>

2015-10-22 kl 16:00

Laserkanon

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/lasermodul.pdf> 2015-10-22 kl 16:00

IR-sändare

<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/sir204.pdf> 2015-10-22 kl 16:00