**Designspecifikation**

G07

Version 1.0

Status

| Granskad | FR, FN, JV, CN, ML | 2023-10-10 |
| --- | --- | --- |
| Godkänd | TL | 2023-10-10 |

**S.A.N.T.A M.A.R.I.A**

Grupp 07, 2023HT  
Linköpings tekniska högskola, ISY

| **Namn** | **Ansvar** | **Telefon** | **E-post** |
| --- | --- | --- | --- |
| Felix Ramnelöv | Projektledare (PL) och Dokumentansvarig (DOK) | 073-510 14 07 | [felra653@student.liu.se](mailto:felra653@student.liu.se) |
| Filip Nygren | Versionshantering (GIT) och Systemarkitekt (SA) | 076-059 66 76 | [filny841@student.liu.se](mailto:filny841@student.liu.se) |
| Hannes Lindström | UX-designer (UX) och Implementationsansvarig (IMP) | 070-847 82 39 | [hanli001@student.liu.se](mailto:hanli001@student.liu.se) |
| Jacob Volz | Leveransansvarig (LEV) och Ekonom (EKO) | 076-306 07 27 | [jacvo343@student.liu.se](mailto:jacvo343@student.liu.se) |
| Christoffer Näs | Integrationsansvarig (ITG) | 072-568 45 61 | [chrna581@student.liu.se](mailto:chrna581@student.liu.se) |
| Mikael Lundgren | Testansvarig (QA) | 070-754 76 70 | [miklu523@student.liu.se](mailto:miklu523@student.liu.se) |

**E-postlista för hela gruppen**: [TSEA29\_2023HT\_XX-Grupp7@groups.liu.se](mailto:TSEA29_2023HT_XX-Grupp7@groups.liu.se)

**Kund:** Anders Nilsson VALLA, B-Huset, Ingång 27, Rum 3B.512,  
 +4613282635, [anders.p.nilsson@liu.se](mailto:anders.p.nilsson@liu.se)

**Kontaktperson hos kund:** Anders Nilsson VALLA, B-Huset, Ingång 27, Rum 3B.512,  
 +4613282635, [anders.p.nilsson@liu.se](mailto:anders.p.nilsson@liu.se)

**Kursansvarig**: Anders Nilsson VALLA, B-Huset, Ingång 27, Rum 3B.512,  
 +4613282635, [anders.p.nilsson@liu.se](mailto:anders.p.nilsson@liu.se)

**Handledare:** Theodor Lindberg, VALLA, B-huset, Ingång 25,  
 Rum 3B:551A, [theodor.lindberg@liu.se](mailto:theodor.lindberg@liu.se)

**Innehåll**

[**1. Inledning 1**](#_heading=h.usng7ab5misq)

[**2. Sammanfattning 1**](#_heading=h.tyjcwt)

[2.2 Delsystem 1](#_heading=h.kj2gcjcgxbgz)

[2.3 Blockschema 1](#_heading=h.o2rgui95sqc4)

[2.4 Komponenter 2](#_heading=h.386knshiqe2o)

[2.4.1 Beräkningsenheter 2](#_heading=h.ykxbwwtriys4)

[2.4.2 IC-kretsar 2](#_heading=h.qzegecsxp2b5)

[2.4.3 Sensorer 2](#_heading=h.ic33d0jfzoeo)

[2.4.4 Ställdon 2](#_heading=h.dh8efkhtoqz3)

[**3. Beskrivning av delsystemen 3**](#_heading=h.56hupp1troah)

[3.2 Huvudmodul 3](#_heading=h.5x2zwysahru)

[3.2.1 Beskrivning 3](#_heading=h.k1mp3xzhhuff)

[3.2.2 Kopplingsschema 3](#_heading=h.72msigps8wl)

[3.2.3 Komponentbudget 4](#_heading=h.53kkasu8wylz)

[3.2.4 Prestandabedömning 4](#_heading=h.c2fg3nm52p6t)

[3.2.5 Programkodsbeskrivning 4](#_heading=h.ese4if5g46fk)

[3.3 Kommunikationsmodul 4](#_heading=h.foy2k7r9glsi)

[3.3.1 Beskrivning 4](#_heading=h.66zg2oli88ul)

[3.3.2 Kopplingsschema 5](#_heading=h.lf7f9kggv4yv)

[3.3.3 Komponentbudget 5](#_heading=h.u3eoxggqflb0)

[3.3.4 Programkodsbeskrivning 5](#_heading=h.xbissqgbcv3w)

[3.4 Styrmodul 7](#_heading=h.bn1f27wyfc2q)

[3.4.1 Beskrivning 7](#_heading=h.shznv514fbdj)

[3.4.2 Kopplingsschema 7](#_heading=h.orc5q6acb5h7)

[3.4.3 Komponentbudget 8](#_heading=h.xetmn3m9upi9)

[3.4.4 Prestandabedömning 8](#_heading=h.94dtppjoqxyu)

[3.4.5 Programkodsbeskrivning 8](#_heading=h.ul4ks0nvatjd)

[3.5 Sensormodul 11](#_heading=h.rezgp1accphp)

[3.5.1 Beskrivning 11](#_heading=h.rgvhlvujiyg)

[3.5.2 Kopplingsschema 11](#_heading=h.rygmvxpgf6p5)

[3.5.3 Komponentbudget 11](#_heading=h.7hn6gbia7gz8)

[3.5.4 Prestandabedömning 12](#_heading=h.rzg9r321sfy)

[3.5.5 Programkodsbeskrivning 12](#_heading=h.bq7kcdg7qmho)

[3.6 Extern dator 14](#_heading=h.u05ydmbzchne)

[3.6.1 Beskrivning 14](#_heading=h.qj19hx4wesdm)

[3.6.2 Kopplingsschema 14](#_heading=h.wl3p11i60toy)

[3.6.3 Komponentbudget 14](#_heading=h.vtjgktz37b88)

[3.6.4 Programkods beskrivning 14](#_heading=h.fjj6byk840yu)

[**4. Beskrivning av kommunikation mellan delsystemen 16**](#_heading=h.zdw03auojwsd)

[4.2 Huvudbuss 16](#_heading=h.cjbqj4nnh7bj)

[4.2.1 Roller 16](#_heading=h.9h5oguni46qv)

[4.3 Bluetooth 17](#_heading=h.gk4q2f92sh2w)

[4.3.1 Kommunikation med extern dator 17](#_heading=h.6b4028e6q5c9)

[**5. Implementeringsstrategi 18**](#_heading=h.ydq27d3cy9cd)

[5.2 Strategi 18](#_heading=h.et2i2yyv237y)

[5.3 Testning och feedback 18](#_heading=h.4qgm49zedp4m)

[**Referenser 19**](#_heading=h.44sinio)

[Publicerade källor 19](#_heading=h.s5ky0ouketel)

[Elektroniska källor 19](#_heading=h.hfdgrh43dilm)

[Opublicerade källor 19](#_heading=h.v0ob9vunavod)

[Personlig kommunikation 19](#_heading=h.r9nvsmxrpss8)

**Dokumenthistorik**

| **Version** | **Datum** | **Utförda förändringar** | **Utförda av** | **Granskad** |
| --- | --- | --- | --- | --- |
| 1.0 | 2023-10-10 | Första version | G07 | TL, FR |
| 0.2 | 2023-10-09 | Små förändringar | G07 | FR |
| 0.1 | 2023-10-05 | Första utkastet | G07 | FR |

# Inledning

I det här dokumentet presenteras systemet, dess komponenter samt designbeslut, kopplings- och flödesscheman. Dessa beslut beskrivs på en högre nivå för hela systemet men även på en lägre nivå för vardera delsystem. Dessutom finns ett avsnitt som diskuterar vidare kommunikation mellan delsystem samt ett avsnitt som behandlar implementeringsstrategier.

# Sammanfattning

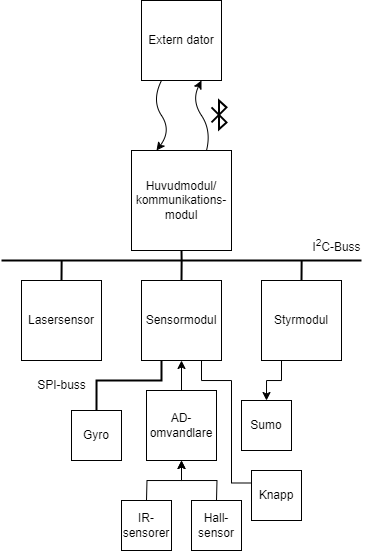
Robotens uppgift är att autonomt ta sig runt i en bana och kartlägga området. Med hjälp av olika sensorer samlar roboten in data om hur området är uppbyggt. Det skickas över till en extern dator som ritar upp grafiskt hur det ser ut.

## Delsystem

Systemet är uppbyggt på fem moduler: huvudmodul, kommunikationsmodul, sensormodul, styrmodul och en extern dator. Huvudmodulen sköter kommunikationen mellan modulerna på roboten och I2C-bussen. Den har även algoritmen för robotens autonoma körning. Sensormodulen samlar data om hur området ser ut runt omkring roboten. Datan skickas till algoritmen för autonom körning och extern dator för att rita upp kartan.

## Blockschema

Figuren nedan visar ett blockschema över hela systemet.



1. Blockschema över hela systemet

## Komponenter

Nedan listas de komponenter som kommer användas.

### Beräkningsenheter

| **Komponent** | **Antal** |
| --- | --- |
| ATmega16 | 2 |
| Raspberry Pi 3 | 1 |

### IC-kretsar

| **Komponent** | **Antal** |
| --- | --- |
| EXO3 | 1 |

### Sensorer

| **Komponent** | **Antal** |
| --- | --- |
| GP2D120 | 2 |
| LIDAR-Lite V2 | 1 |
| A1120 | 1 |
| MLX90609 | 1 |
| Avstudsad trycknappsmodul 2x1 | 1 |

### Ställdon

| **Komponent** | **Antal** |
| --- | --- |
| Sumo/Terminator | 1 |

# 

# 

# Beskrivning av delsystemen

I detta avsnitt beskrivs de olika ingående delsystemen.

## Huvudmodul

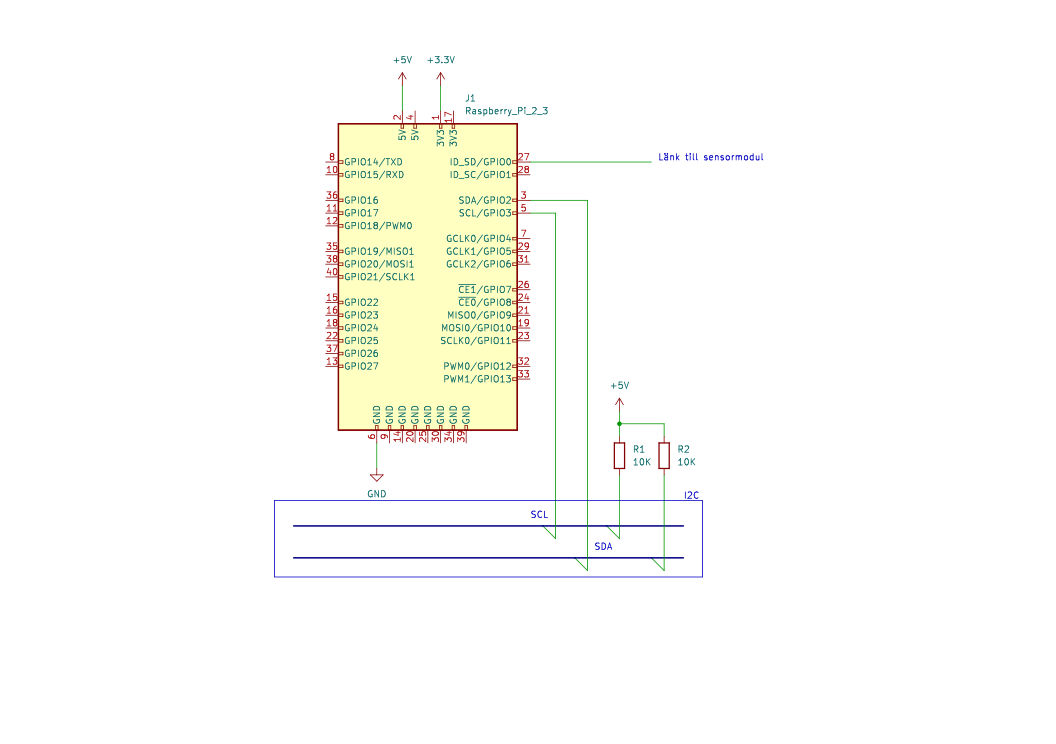
I följande del beskrivs huvudmodulen.

### Beskrivning

Huvudmodulen hämtar data från sensormodulen för att kunna köra kartläggningsalgoritmen. Den berättar för styrmodulen hur roboten ska åka och skickar data över kartlagt område till kommunikationsmodulen. Huvudmodulen är också master på I2C-bussen.

### Kopplingsschema

I figuren nedan visas kopplingsschemat för huvudmodulen. Det är en enkel koppling då det enbart är en Raspberry Pi som är kopplat till en I2C-buss. Dessutom finns det en avstudsad tryckknapp som är inkopplad för att välja mellan autonomt och manuellt styrläge.



1. Kopplingsschema för huvudmodulen.

### 

### Komponentbudget

Tabellen nedan visar komponentbudgeten för huvudmodulen.

| **Komponent** | **Antal** |
| --- | --- |
| Raspberry Pi 3 | 1 |

### Prestandabedömning

Minnet samt antalet I/O-portar räcker gott för modulens ändamål. Tabellen nedan beskriver portar med antal som finns och antal som krävs.

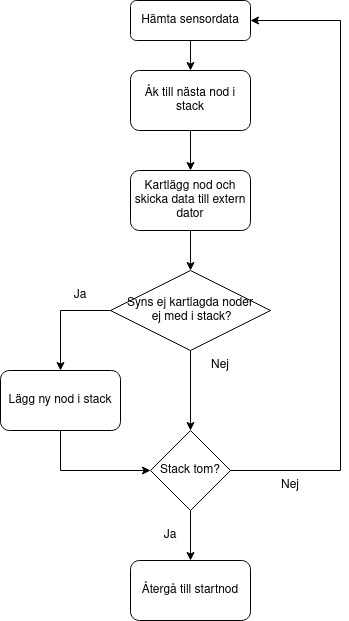
| **Port** | **Antal** | **Krav** |
| --- | --- | --- |
| SDA | 1 | 1 |
| SCL | 1 | 1 |

### 

### 

### Programkodsbeskrivning

Kartroboten ska söka området med en djupet först sökning. Banan är uppbyggd med moduler med måtten 40x40 som representeras av noder i ett träd.

1. Flödesschema för huvudmoduls programkod.

Huvudmodulen kommunicerar till styrmodulen vilket håll roboten ska åka. Vid instruktioner där det ska utföras en rotation, hämtas den initiala rotationen för att sedan jämföras med den förändrade rotationen tills det att en fullständig rotation har genomförts.

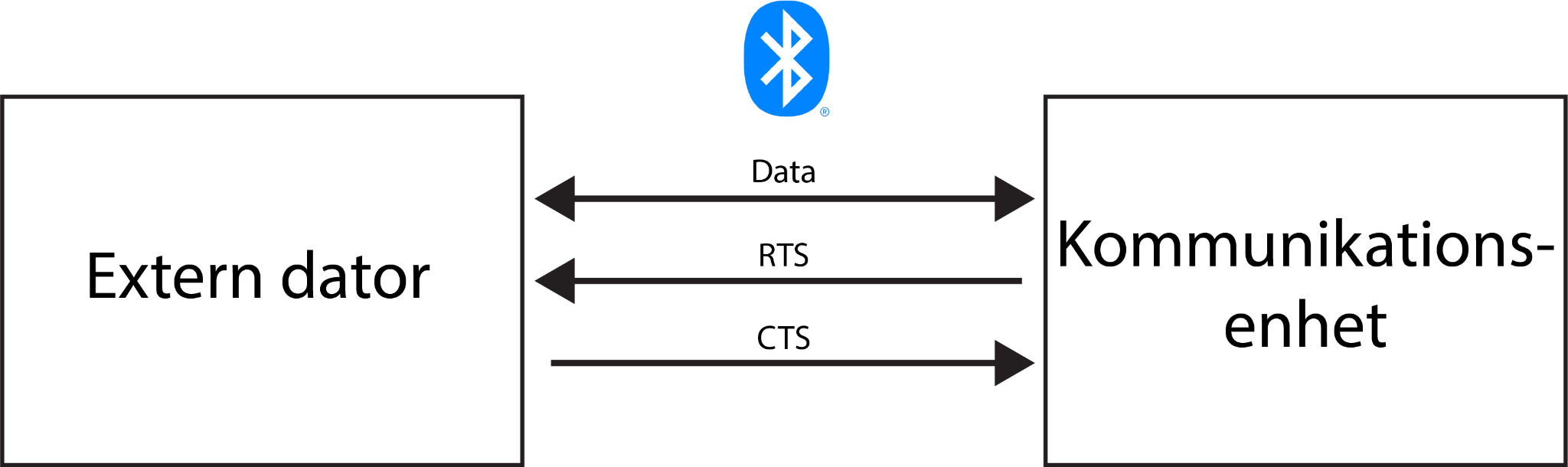
## Kommunikationsmodul

I följande del beskrivs kommunikationsmodulen.

### Beskrivning

Kommunikationsmodulen sköter all kommunikation mellan robot och extern dator. För att kunna rita upp kartan och styra roboten manuellt behöver kommunikationsmodulen kunna kommunicera med en extern dator, det sker över bluetooth. Signalen CTS används av mottagaren för att meddela avsändaren att enheten är redo att ta emot data. RTS signalen använder avsändaren för att be om tillstånd från mottagaren att kunna skicka data.

### Kopplingsschema



1. Bluetoothkommunikation mellan kommunikationsmodul och extern dator.

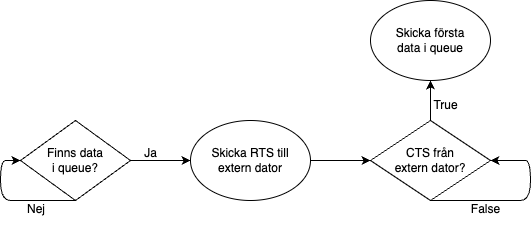
### Komponentbudget

| **Komponent** | **Antal** |
| --- | --- |
| Raspberry Pi 3 | 1 |

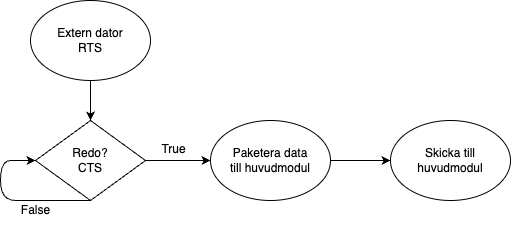
Värt att tillägga är att samma Raspberry Pi som i huvudmodulen används för kommunikationsmodulen.

### Programkodsbeskrivning

Data från huvudmodulen sparas i en lista med längd 2, där index 0 är vad för data som skickas, sparat som en string och index 1 är själva datan. Detta sparas sen i en queue, som sedan skickas till externa datorn så länge queuen inte är tom.



1. Flödesschema för data från kommunikationsmodulen till extern dator.



1. Flödesschema för data från extern dator till kommunikationsmodulen.

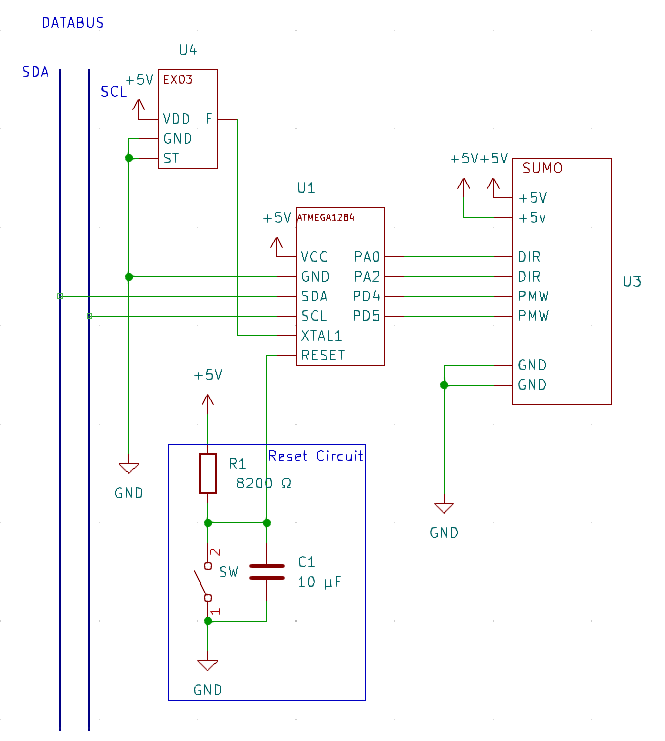
## Styrmodul

I följande del beskrivs styrmodulen.

### Beskrivning

Robotens motorik utövas genom styrmodulen. Styrmodulen består av en AVR (ATmega1284) som hanterar externa styrsignaler i form av bit-vektorer från I2C-bussen och konverterar dessa till olika typer av rörelser på roboten. Rörelsen i roboten åstadkoms via fyra signaler från AVR:en till ställdonet, två signaler för hjulbandens acceleration samt två signaler för hjulbandens rotationsriktning.

### Kopplingsschema



1. Kopplingsschema för styrmodulen.

Styrmodulens micro controller *ATmega1284*  tar in en *SDA* signal innehållande styrdirektiv samt en *SCL* signal innehållande systemets klocka ifrån I2C bussen.Datorn räknar ut rätt värden för rotationshastighet och rotationsriktning utifrån intagen data och skickar vidare värden för rotationshastighet samt rotationsriktning till ingångar på *SUMO.* En *EXO3* är inkopplad till *ATmega1284* för att få en pålitlig frekvens på klockan.

### Komponentbudget

| Komponent | Beskrivning | Antal |
| --- | --- | --- |
| ATmega1284 | Styrmodulens microcontroller | 1 |
| SUMO | Robotens chassi, motorer och hjulband | 1 |
| EXO3 | Extern klocka | 1 |

### Prestandabedömning

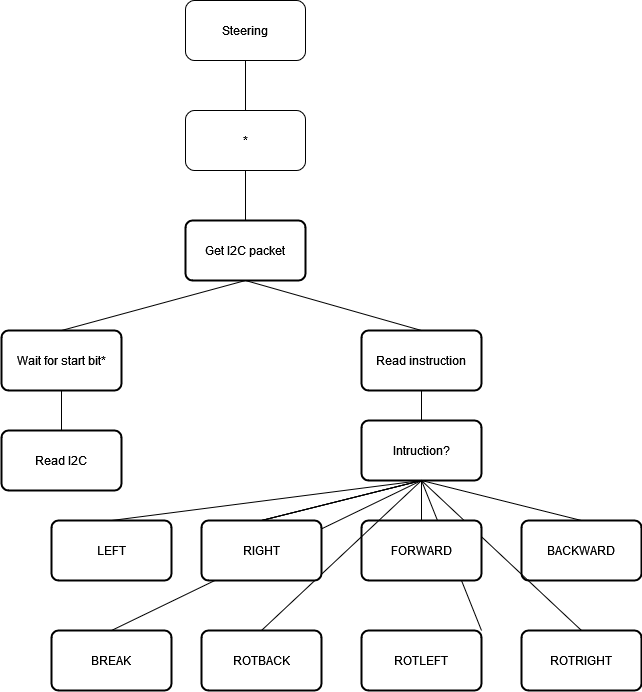
ATmega1284 bedöms innehålla ett tillräckligt stort minne och tillräckligt snabb processor för att kunna utföra dess uppgift. SUMO-komponenten har tillräcklig rörelsefunktion för att kunna utföra efterfrågade rörelsemoment.

| **Port** | **Antal** | **Krav** |
| --- | --- | --- |
| XTAL | 2 | 1 |
| PORTA | 8 | 2 |
| PORTB | 8 | 1 |
| PORTD | 8 | 2 |
| SDA | 1 | 1 |
| SCL | 1 | 1 |

### 

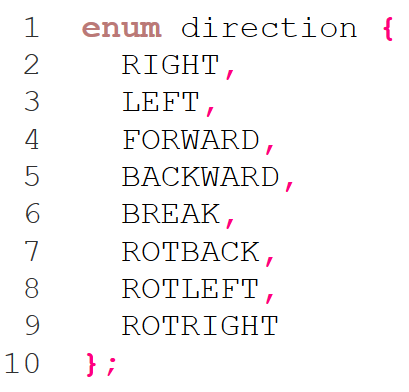
### Programkodsbeskrivning

Styrmodulens programkod har en huvudloop som körs kontinuerligt. Inledningsvis vid en iterering väntar styrmodulen på att en startbit inkommer från data-bussen. Därefter blir instruktionen inläst från data-bussen och omvandlad till ett *instruction-*element som sedan används i en switch-case sats för att exekvera instruktionen. Detta innebär att tillsätta de portar som är kopplade till acceleration och riktning till det passande värdet. Efter en instruktion är exekverad returnerar funktionen tillbaka till huvudloopen vilket markerar en avslutad iterering.



1. JSP diagram över programkoden för styrmodulen.

Inkommande styrkommandon från data-bussen matas bitvis in i en *character Array*. Styrkommandon representeras med datastrukturen enum med namnet *instruction* enligt figur 7.



1. Datastruktur för styrriktning.

## 

## Sensormodul

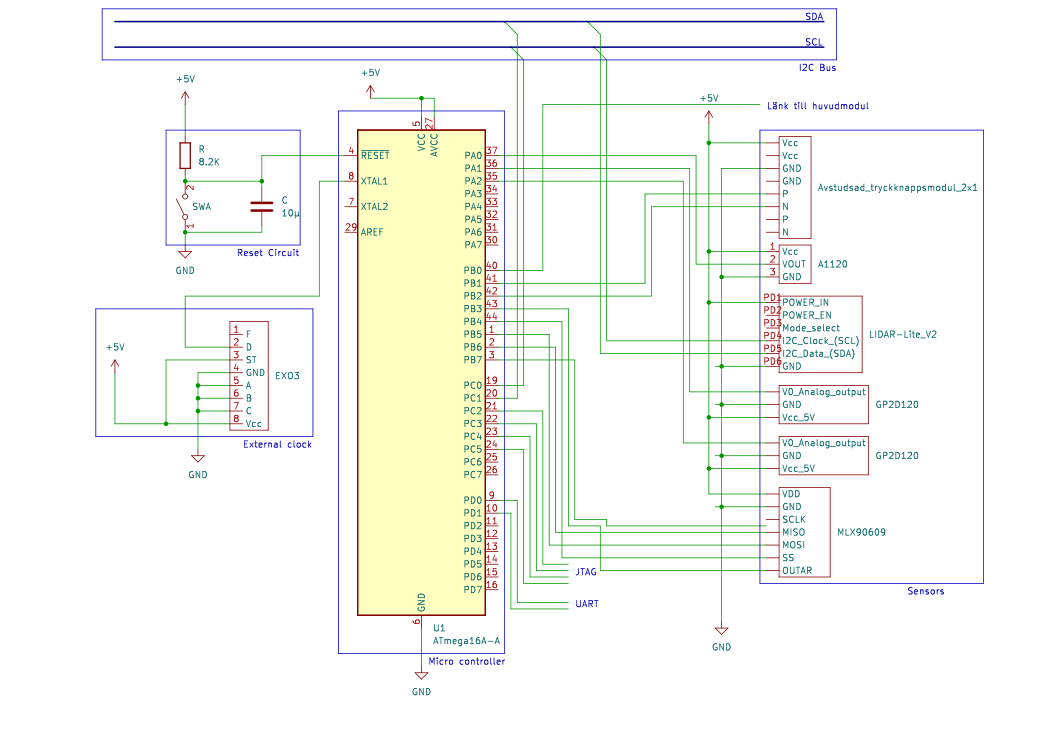
I följande del beskrivs sensormodulen.

### Beskrivning

Sensormodulen har en viktig uppgift, att genom sensorer ta in data och föra vidare till huvudmodulen. Eftersom sensormodulen är kopplad till flera sensorer kommer data få tas i olika steg och sparas i minnet. Sedan kommer huvudmodulen som master efterfråga data från sensormodulen genom I2C.

### Kopplingsschema

Figuren nedan är ett kopplingsschema över sensormodulen. Modulen är uppbyggd runt en ATmega16A som är kopplad till en I2C-bus för att kommunicera med huvudmodulen. I sensormodulen ingår dessutom en reset-krets för att göra reset, en extern klocka då den interna klockan ej fungerar väl. Den externa klockan är uppbyggd med en EXO3 (samma som i styrmodulen). Slutligen finns det fyra sensorer i modulen, en A1120 (halleffektssensor), en LIDAR-Lite V2 (lasersensor), MLX90609 (Gyro) och två GP2D120 (IR-sensor). Halleffektsensorn och IR-sensorerna ger analog data som AD-omvandlas i micro controller:n medan lasersensorn har möjlighet för I2C vilket gör att en omvandling inte behöver ske. Gyrot kopplas in direkt i micro controllern genom SPI. Dessutom är det specificerat vart JTAG samt UART kopplas in för programmering och debugging.



1. Kopplingschema över sensormodulen

### Komponentbudget

Tabellen nedan listar de komponenter som behövs i modulen.

| **Komponent** | **Beskrivning** | **Antal** |
| --- | --- | --- |
| ATmega16 | Sensormodulens micro controller | 1 |
| GP2D120 | IR-sensor för reglering | 2 |
| LIDAR-Lite V2 | Lasersensor för reglering | 1 |
| A1120 | Används som odometer för reglering. | 1 |
| EXO3 | Extern klocka | 1 |
| MLX90609 | Gyro (Angular rate sensor) | 1 |

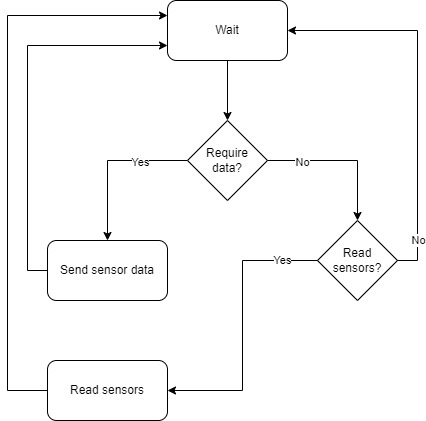
### Prestandabedömning

Minnet samt antalet I/O-portar räcker gott för modulens ändamål. Tabellen nedan beskriver portar med antal som finns och antal som krävs.

| **Port** | **Antal** | **Krav** |
| --- | --- | --- |
| SDA | 1 | 1 |
| SCL | 1 | 1 |
| A/D | 8 | 3 |
| RESET | 1 | 1 |
| JTAG | 1 | 1 |
| UART | 1 | 1 |
| PORTB | 8 | 1 |

### Programkodsbeskrivning

Figur 10 är ett flödesschema över sensormodulens mjukvara. Modulen kör en huvudloop där den väntar på svar från huvudmodulen som master gällande om data önskas. Om data önskas kommer det att skickas till huvudmodulen. Däremot om data inte önskas kommer modulen att ställas inför frågan om sensorer ska läsas. Om sensorer ska läsas av kommer en AD-omvandling att ske och all data från sensorer kommer att lagras i micro controllerns minne, redo för att skickas iväg. Lasersensorn som inte behöver AD-omvandlas kan skickas direkt till micro controllern via I2C. Exakt hur mjukvaran kommer se ut tas inte beslut om just nu, utan blir ett resultat av de designbeslut som tas på hårdvarunivå. Så enkla datastrukturer som möjligt kommer att användas.



1. Flödesschema för sensormodulen

## Extern dator

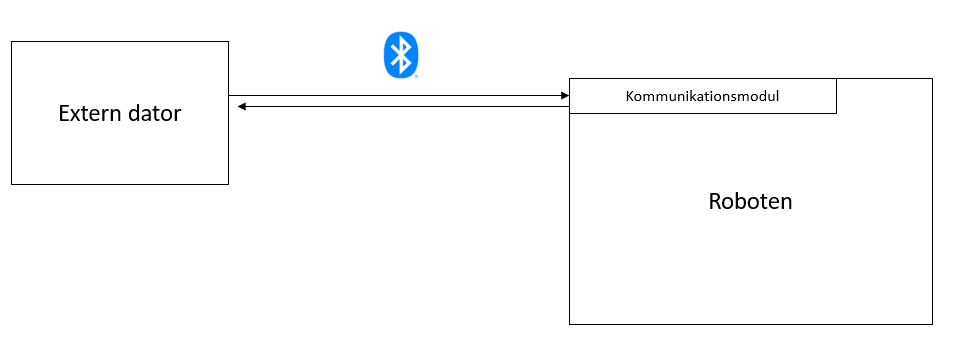
I följande del beskrivs den externa datorn.

### Beskrivning

Externa datorns uppgift är att grafiskt visa upp det kartlagda området som roboten befinner sig i för användaren. Den tar emot data trådlöst från kommunikationsmodulen medan roboten kör runt. Vid manuellt körläge skickar den externa datorn styrsignaler till roboten om hur den ska navigera.

### Kopplingsschema

Figuren nedan visar kopplingsschemat för den externa datorn.



1. Kopplingsschema för den externa datorn.

### Komponentbudget

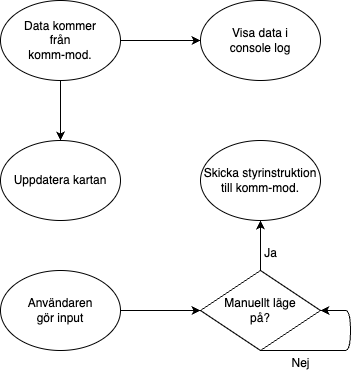
I tabellen nedan beskrivs komponentbudgeten för externa datorn.

| Komponent | Beskrivning | Antal |
| --- | --- | --- |
| Extern dator | Den externa datorn. | 1 |

### Programkods beskrivning

Programkoden för den externa datorn, det vill säga användargränssnittet, kommer att bygga upp ett grafiskt gränssnitt likt tidigare visat i systemskissen. Det är inte för tillfället specificerat vilket språk som kommer användas, det finns en rad olika alternativ med tanke på att operativsystemet på externa datorn är kompatibelt med mycket. Antagligen kommer ett språk av högre nivå att användas då det finns fler färdiga bibliotek och datorn har tillräckligt med prestanda. Bibliotek som kan komma att användas är för hantering av bluetooth och som framework för gränssnittet. Ett exempel på ett språk med de typer av bibliotek är python.

Externa datorn får data från kommunikationsmodulen som sedan används i användargränssnittet. Data kommer dels visas i konsolloggen för att kunna analyseras, men vi kommer även använda datan för att rita upp kartan grafiskt. I användargränssnittet finns även knappar för manuell styrning som låses upp när manuell styrning är aktiverad.



1. Flödesschema för användargränssnittet

# Beskrivning av kommunikation mellan delsystemen

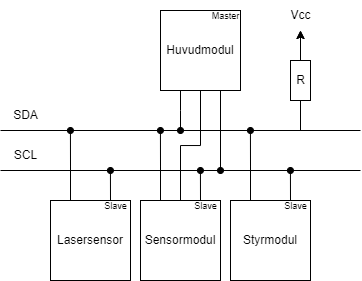
Kommunikation mellan moduler kommer att ske både fysiskt med en gemensam buss mellan komponenterna i roboten och trådlöst med bluetooth mellan kommunikationsmodul och den externa datorn. All kommunikation av samma typ kommer att göras på samma sätt för att göra saken enklare, det kommer leda till att huvudmodulen kan hantera all kommunikation på samma vis. Detta leder i sin tur till en snabbare utvecklingsgång och färre fel.

## Huvudbuss

Den fysiska kommunikationen kommer ske över en I2C-buss, där huvudmodulen skickar och tar emot data till och från sensormodulen samt styrmodulen. Dessutom är det huvudmodulens uppgift att skicka sensordata mellan lasersensorn och sensormodulen då den är uppkopplad direkt till bussen. Bussen i sig är två-trådad, en tråd för data (SDA) och en tråd för klocka (SCL). Till SDA måste dessutom ett pull-up-motstånd anslutas. Vi kommer använda oss av ett master-slave förhållande för kommunikation. Datan kommer att skickas som ett paket med all data på en gång då den efterfrågas.

### Roller

Figuren nedan visar master-slave förhållandet mellan modulerna. Pull-up-motståndet finns även utritat i figuren.



1. Schema över master-slave förhållandet mellan modulerna.

Det är huvudmodulen som är master och styr över kommunikationen på I2C-bussen. Det är inget problem då data från sensormodulen ska till huvudmodulen eller data från huvudmodulen ska till styrmodulen eftersom det är huvudmodulen som initierar den kommunikationen. Däremot behöver data från lasersensorn transporteras till sensormodulen. För att lösa detta är det möjligt att det behövs en extra länk mellan sensormodulen och huvudmodulen där sensormodulen kan tala om för huvudmodulen att den vill ha data från lasersensorn.

## Bluetooth

Komunikationen mellan extern dator och roboten sker över Bluetooth. Det är för att ha möjlighet att kunna rita upp kartan på extern dator och att styra roboten manuellt. Bluetooth finns förinstallerat på Rapberry Pi 3B+ vilket gör det möjligt för en extern dator med bluetooth att ansluta till kommunikationsmodulen och skicka data mellan varandra. Signalerna RTS (request to send) används för att meddela mottagaren att enheten är redo att skicka data, den går då aktivt hög. CTS (clear to send) svarar RTS signalen genom att också gå aktivt hög när enheten är redo att ta emot data från avsändaren. När avsändaren får in att CTS signalen från mottagaren är hög skickas data. När mottagaren inte kan ta emot mer data går CTS signalen låg och avsändaren inväntar med att skicka mer data fram tills att signalen blir hög igen. Dessa signaler är handskakningen mellan modulerna, vilket förebygger överbelastning och förlust av data.

### Kommunikation med extern dator

Till extern dator skickas data som kommunikationsmodulen tar emot från I2C-bussen. I modulen ompaketeras datan innan den skickas över till extern dator. Från extern dator tar kommunikationsmodulen emot data som ompaketeras och skickas ut till övriga moduler via I2C-bussen.

Data som skickas från kommunikationsmodul till extern dator:

* Kartdata vid nuvarande tidpunkt.
* Robotens position.
* Byte av styrläge.

Data som skickas från extern dator till kommunikationsmodul:

* Styrsignaler.
* Förfrågan om styrläge.
* Förfrågan om kartdata
* Förfrågan om robotens position

# Implementeringsstrategi

## Strategi

Projektets implementeringsstrategi är centrerat kring hårdvarunära design, det vill säga utifrån och in. Huvudprocessen är P.A.R.K, planera, agera, reflektera och korrigera. Att använda sig av hårdvarunära design innebär att utvecklingen sker i små, frekventa steg, med blicken framåt och grundlig reflektion både framåt och bakåt vid varje utfall. Det faktum att vi utvecklar utifrån och in gör att mjukvaran anpassas beroende på hårdvaran.

## Testning och feedback

“Black box”-testning är grundpelaren under projektet vad gäller testnings-ramverk och appliceras under modultestning samt systemtestning. Att testa med “Black Box"-testning innebär att endast indata och resultat är intressanta under test. En specifik input ska ge en specifik output. Feedback demonstreras fysiskt, till exempel att hjulband snurrar, eller virtuellt genom signaler i WaveForms eller respons i det grafiska gränssnittet på den externa datorn vid körning. Sampling sker i frekvensen 32Hz för att ge systemet relativt frekvent sensordata samt upphov till så många klockcykler som möjligt ägnat åt processering.

# Referenser

### Publicerade källor

### Elektroniska källor

### Opublicerade källor

### Personlig kommunikation