

# TSEA56 - Kandidatprojekt i elektronik

## Designspecifikation för resQ.PL

Version 0.1

Grupp 2

Agafonov, Nikolaj, `nikag669`

Berberovic, Adnan, `adnbe196`

Brorsson, Andreas, `andbr981`

Fridborn, Fredrik, `frefr166`

Oprea, Robert, `robop806`

Skytt, Måns, `mansk700`

11 mars 2015

Status

Granskad	Adnan Berberovic	11 mars 2015
Godkänd		

**PROJEKTIDENTITET**

2015/VT, Undsättningsrobot Gr. 2  
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Nikolaj Agafonov	Dokumentansvarig (DA)	072-276 99 46	nikag669@student.liu.se
Adnan Berberovic	Projektledare (PL)	070-491 96 07	adnbe196@student.liu.se
Andreas Brorsson	Testansvarig (TA)	073-524 44 60	andbr981@student.liu.se
Fredrik Fridborn	Designansvarig Sensormodul (DSE)	073-585 52 01	frefr166@student.liu.se
Robert Oprea	Designansvarig Styrmodul (DST)	070-022 10 18	robop806@student.liu.se
Måns Skytt	Designansvarig Kommunikationsmodul (DK)	070-354 28 84	mansk700@student.liu.se

E-postlista för hela gruppen: adnbe196@student.liu.se

Kund: Kent Palmkvist, 581 83 Linköping, Kundtelefon: 013-28 13 47, kentp@isy.liu.se

Kursansvarig: Tomas Svensson, 013-28 13 68, tomass@isy.liu.se  
Handledare: Olov Andersson, 013-28 26 58, Olov.Andersson@liu.se

## Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Grov beskrivning av Roboten . . . . .	1
1.2	Sensorplacering . . . . .	1
1.3	Motorbeskrivning . . . . .	2
<b>2</b>	<b>Detaljerad beskrivning av delsystem</b>	<b>2</b>
<b>3</b>	<b>Kommunikationsmodul</b>	<b>2</b>
3.1	Bluetoothmodem - Firefly . . . . .	2
3.2	Hårdvara . . . . .	3
3.3	Kopplingsschema . . . . .	3
3.4	Mjukvara . . . . .	4
<b>4</b>	<b>Sensormodul</b>	<b>5</b>
4.1	Hårdvara och komponentbudget . . . . .	6
4.1.1	Sensorplacering . . . . .	6
4.1.2	Optisk avståndsmätare . . . . .	6
4.1.3	Kort avståndsmätning . . . . .	7
4.1.4	Lång avståndsmätning . . . . .	7
4.1.5	Reflexsensor . . . . .	7
4.1.6	ATmega1284p . . . . .	7
4.1.7	Gyro . . . . .	8
4.1.8	Kopplingsschema . . . . .	9
4.2	Mjukvara . . . . .	9
4.2.1	Avståndssensor . . . . .	9
4.2.2	Reflexsensor . . . . .	10
4.2.3	Gyro . . . . .	10
4.2.4	Pseudokod . . . . .	10
<b>5</b>	<b>Styrmodul</b>	<b>12</b>
5.1	Hårdvara och komponentbudget . . . . .	12
5.2	Kopplingschema . . . . .	12
5.3	Programkod . . . . .	13
5.3.1	Mottagning av sensordata . . . . .	14
5.3.2	Uppdatering av karta . . . . .	14
5.3.3	Bestämmande av färdriktning . . . . .	14
5.3.4	Sändning av styrdata . . . . .	14
<b>6</b>	<b>PC</b>	<b>14</b>
6.1	Funktioner . . . . .	15
6.1.1	Manuell styrning . . . . .	15
6.1.2	Mätdata . . . . .	15
6.1.3	Karta . . . . .	15
<b>7</b>	<b>Implementeringsstrategi</b>	<b>15</b>

<b>8 Intern kommunikation</b>	<b>16</b>
8.1 Beskrivning av SPI protokollet . . . . .	16
8.2 SPCR - SPI Control Register . . . . .	17
8.2.1 SPIE- SPI Interrupt Enable . . . . .	17
8.2.2 SPE - SPI Enable . . . . .	17
8.2.3 DORD - Data Order . . . . .	17
8.2.4 MSTR - Master/Slave select . . . . .	17
8.2.5 CPOL - Clock Polarity . . . . .	17
8.2.6 CPHA - Clock Phase . . . . .	17
8.2.7 SPR1, SPR0 - SPI Clock Rate Select 1 and 0 . . . . .	17
8.3 SPSR - SPI Status Register . . . . .	18
8.3.1 SPIF - SPI Interrupt Flag . . . . .	18
8.3.2 WCOL - Write Collision Flag . . . . .	18
8.3.3 SPI2X - Double SPI Speed Bit . . . . .	18
8.4 Dataöverföring . . . . .	18
<b>Referenser</b>	<b>19</b>
<b>Appendix</b>	<b>20</b>
Appendix A . . . . .	20
Appendix B . . . . .	21
Appendix C . . . . .	24

## Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	11 mars 2015	Första utkastet	Grupp 2	Adnan Berberovic

# 1 Inledning

Detta dokument kommer att behandla hela projektets underfas, det vill säga implementationen av roboten i helhet.

Designen kommer framför allt att behandla projektet från ett hårdvaruperspektiv, men även lite mjukvarudesign kommer att ingå, dock inte mer än pseudokod.

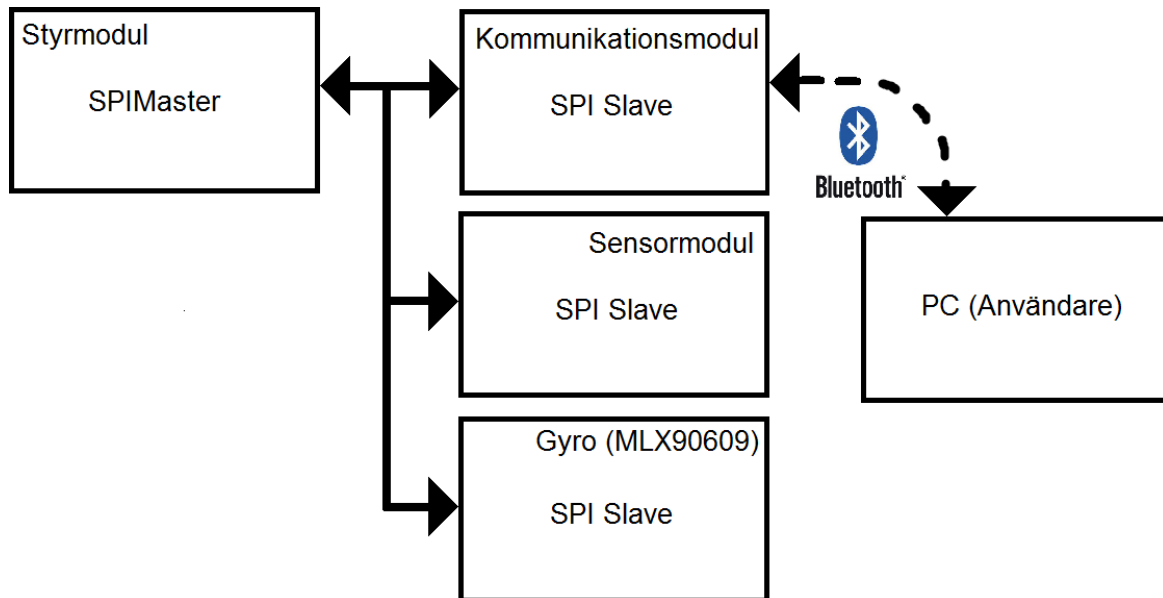
Inga register kommer att vara förkodade, men de kommer förklaras i sådan utsträckning att det under implementationens gång är lätt att inse vilket värde varje bit ska anta.

Nedan sammanfattas robotens design.

## 1.1 Grov beskrivning av Roboten

Roboten kommer i grov utsträckning bestå av 3 moduler enligt figur 1, en Styr-, Kommunikations och Sensormodul. En användare kommer ha möjligheten att kommunicera med roboten via ett program på en dator, där viss data kan visas, samt vissa funktioner utföras genom kommandon.

Samtliga delkomponenter kommer att beskrivas utförligt och ha en detaljerad design som kommer att följas när roboten konstrueras.



Figur 1: Översikt över hela systemet.

## 1.2 Sensorplacering

Roboten kommer att ha 7 avståndsmätare placerade strategiskt på roboten. Roboten kommer att ha en kort avståndsmätare på framsidan, en kort avståndsmätare på vardera främre respektive bakre sida, samt en lång avståndsmätare i mitten på vardera sida. Dessa kommer mäta avstånd som kommer vara till behov för PD-reglering av rörelserna, samt utritande av karta.

Det kommer finnas en reflexsensor på undersidan av roboten för att notera var någonstans målet befinner sig.

Ett gyro kommer att sitta på roboten för att mäta vinkelhastighet vid rotation i korsningar, för att roboten ska ha så stor noggrannhet vid rotation som möjligt.

### 1.3 Motorbeskrivning

Motorerna sitter redan på robotchassit. Det finns fyra motorer som styrs parvis, det vill säga båda på vänster respektive höger sida styrs av samma signal för att möjliggöra rotation. Notera att en motor styr ett hjul. Motorerna kommer styras av styrmodulen, och styrningen kommer att vara PD-reglerad för att med ett litet fel eliminera svängningar i raksträckor.

## 2 Detaljerad beskrivning av delsystem

I följande avsnitt behandlas alla ingående delmoduler var för sig. Kommunikationsmodulen sköter kommunikation med en PC via bluetooth och distribuerar data från denna vidare internt i roboten. Information från roboten (sensordata, kartdata och dylikt) skickas till PC:n där detta visas för användaren. Styrmodulen är master i den SPI-buss som används och hämtar sensordata från sensormodulen och korregerar sin riktning därefter.

## 3 Kommunikationsmodul

Kommunikationsmodulen sköter all kommunikation med omvärden. I detta fall en PC (eller annan Bluetoothenhet). Denna kommunikation sker via Bluetooth och det Bluetoothmodem som används är "FireFly", se avsnitt 3.1. Modulen består av en mikrokontroller som klockas av en EXO-3 kristaloscillator. Denna kopplas till bluetoothmodulen som agerar genomskinlig gateway för data mellan kontrollern och PC:n. Data transporteras till och från kommunikationsmodulen till styrenheten via SPI-buss. Den data som skickas är kartdata och eventuellt sensordata.

### 3.1 Bluetoothmodem - Firefly

Det bluetoothmodem som används är ett Blue SMiRF Gold (FireFly). För hårdvaruspecifikationer se Appendix A. För att kunna kommunicera med denna modul krävs att enheten stödjer SPP (Serial Port Profile).<sup>1</sup> Detta gör merparten av alla datorer och bland dessa den tilltänkta dator som ska användas för kommunikationen. FireFly är ett klass 1 bluetoothmodem vilket innebär lång räckvidd men mer strömkrävande, detta bör dock inte bli ett problem ty den tilltänkta körtiden är såpass kort och den ström som Bluetoothmodemet tar blir försumbar gentemot exempelvis drivande motorer.

När man ska kommunicera över bluetooth krävs det att baud-hastigheterna överensstämmer. För ATmega1284P mikrokontrollern bestäms detta med hjälp av oscillatorklockan i USART (universal synchronous/asynchronous receiver/transmitter). Modemet har en default baud-hastighet på 115200 bps (baud per second). Baud-hastigheten bestäms i ATmega1284p genom att oscillatorklockafrekvensen räknar ner en klocka som laddas med värdet i USART Baud

<sup>1</sup><https://developer.bluetooth.org/TechnologyOverview/Pages/SPP.aspx>

Rate Registret, se Appendix C. När baud-hastigheten i modemmet ska bestämmas måste först denna sättas i *command mode*<sup>2</sup> där, utöver baud-hastighet, bland annat enhetsnamn och PIN-kod kan ställas in. Det är även i *command mode* som talar om när man vill ansluta till en annan enhet eller söka efter andra moduler. Det andra läget som modemmet kan befinna sig i är *data mode*. När modemmet befinner sig i det läget agerar den som en transparent gateway mellan datorn och ATmega128p-kontrollern.

### 3.2 Hårdvara

I tabell 3 listas den hårdvara som krävs för att realisera kommunikationsmodulen. För att undvika otydliga logiska nivåer (d.v.s på gränsen mellan 1 och 0) används en pull-up resistor i samband med resetknappen. Kondensatorer används för att få en så jämn signalspänning som möjligt, förutsatt att kondensatorerna är snabbare på att korrigera snabba spänningsfall än batteriet.

Komponent	Specifikation	Antal
ATmega1284p	Microcontroller 128 kB	1 st
BlueSMiRF Gold	Bluetoothmodem	1 st
Resistor	Till reset	1 st
Kondensator	Till reset	1 st
Tryckknapp	Till reset	1 st
EXO-3	Kristalloscillator till AVR:en	1 st
JTAG ICE mkII	Programmera och emulera program på AVR	1 st

Tabell 3: Hårdvara som kommunikationsmodulen innefattar

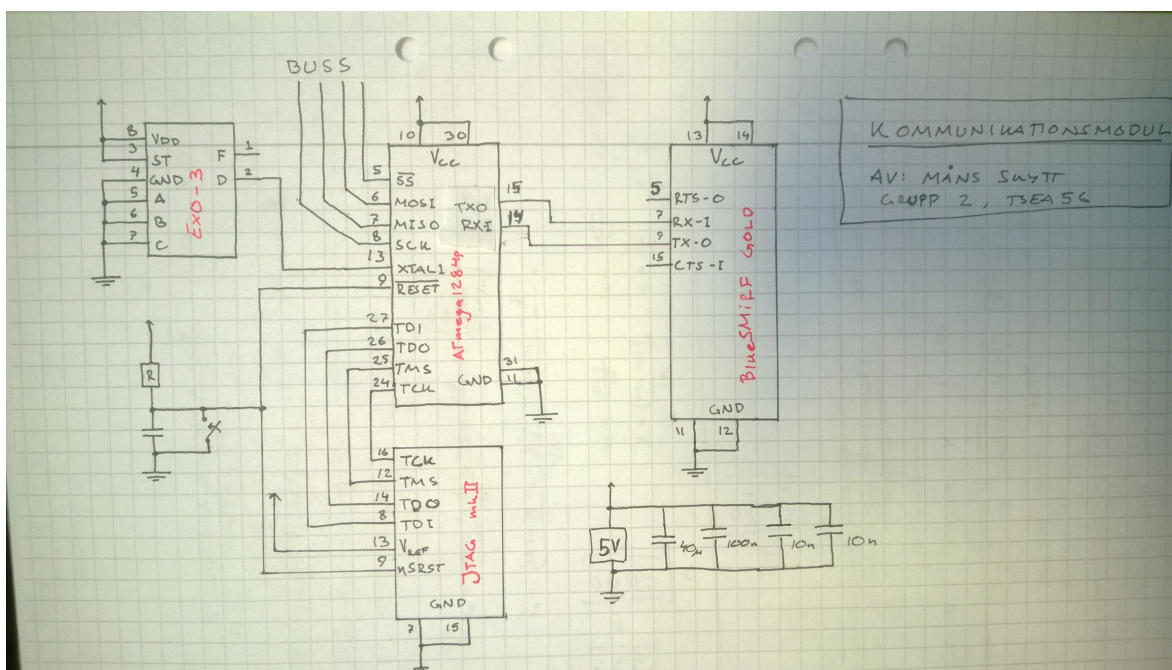
ATmega1284p mikrokontrollern ansluts till EXO-3 kristalloscillatorn som kan generera olika frekvenser. Det som kräver mest minne i modulen är programmet men med 128 kB flash-minne bör detta inte vara någon fara. I övrigt agerar kommunikationsmodulen som mellanhand mellan styrmodul och PC. Allt som behövs är därför register för lagring av data som ska skickas mellan PC (via bluetooth) och styrmodul (via ISP-buss).

### 3.3 Kopplingsschema

Hela kommunikationsmodulen och alla dess delkomponenter uppkopplade kan ses illustrerade i figur 2. Utmärkta pinnar som inte är anslutna är utmärkta för att dessa kan användas om ytterligare (mer komplexa) funktioner vill användas. Som ses i kopplingsschemat nedan finns tillräckligt med portar på mikrokontrollern för att ansluta nödvändiga komponenter.

<sup>2</sup>För att ändra modemets läge till *command mode* skickas strängen "\$\$\$" till modemmet. Detta måste göras inom 60 sekunder efter påslagning av modemmet



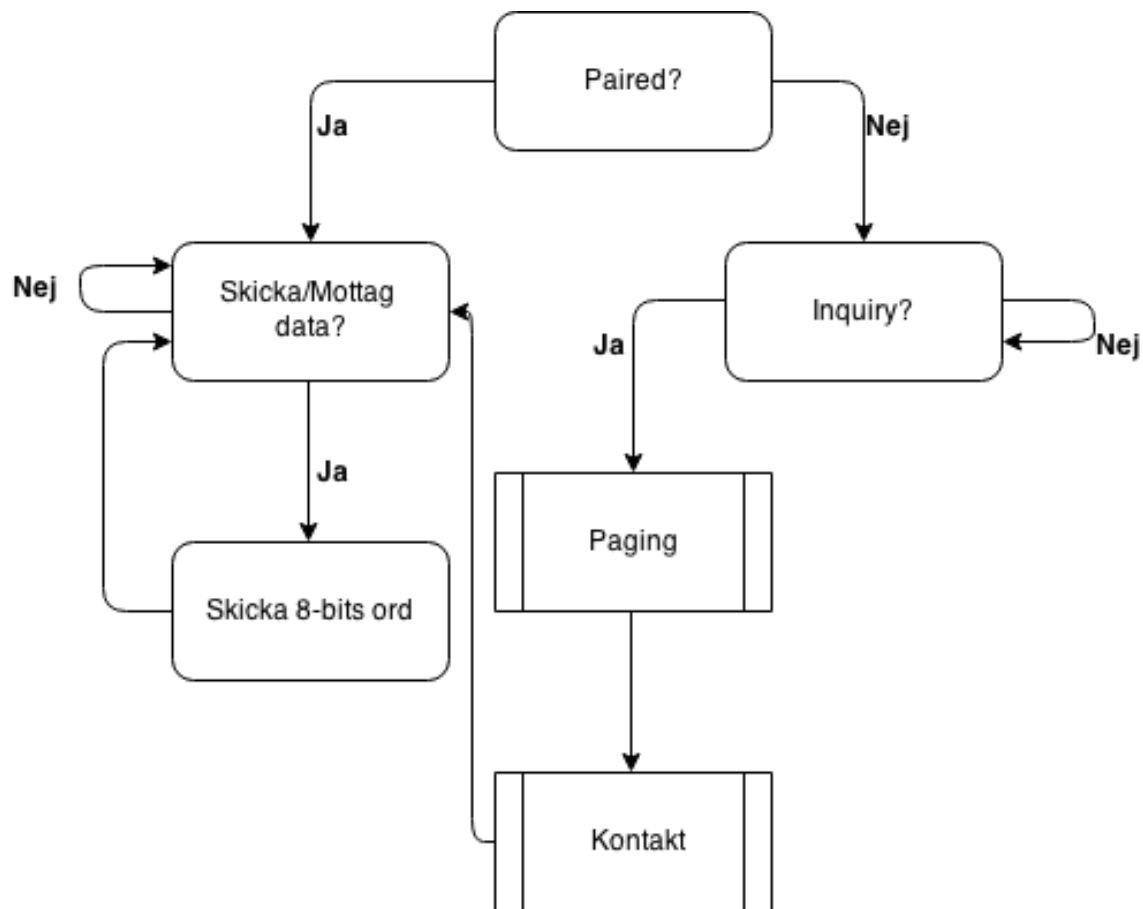


Figur 2: Kopplingsschema för kommunikationsmodulen

### 3.4 Mjukvara

För att upprätta kontakt mellan två bluetoothenheter måste slaven göra en så kallad inquiry scan där den är aktiverad och lyssnar efter inquiries. Om master-enheten får kommandot av användaren att gå in i inquiryläget och börjar sända och lyssna efter svar från en slavenhet. Inquiry görs för att bluetoothenheter ska hitta vilka andra tillgängliga enheter som finns i närheten. När bluetoothenheterna har hittat varandra så skapar de kontakt (för datautbyte) med hjälp av en pagingprocess. Denna kan göras med hjälp av olika paging-scheman men vid förstagångskontakt används alltid samma. När detta är gjort har kontakt upprättats och data kan börja skickas mellan enheterna.

Den process som genomförs vid kontaktskapandet och datautbytet kan ses i form av ett flödesschema i figur 3.



Figur 3: Flödesschema för stegen två enheter genomgår vid informationsutbyte via Bluetooth

## 4 Sensormodul

Sensormodulens uppgift är att ta in information från sensorerna som anges i avsnitt 4.1 och förmedla vidare till övriga moduler. Viss informationshantering kommer att ske i modulen och sedan skickas vidare till styrmodulen. Sensorinformationen är en förutsättning för regleringen av robotens rörelse samt kartläggningen av miljön.

## 4.1 Hårdvara och komponentbudget

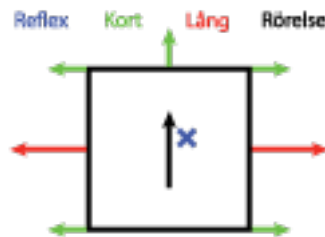
I detta avsnitt listas den hårdvara som ingår i modulen samt antalet komponenter.

Komponent	Specifikation	Antal
ATmega1284p	Microcontroller 128 kB	1 st
GP2Y0A41SK	Optisk avståndsmätare 4-30 cm	5 st
GP2Y0A21	Optisk avståndsmätare 10-80 cm	2 st
Resistor	$18k\Omega$ , till lågpasfilter	8 st
Kondensator	$100nF$ , till lågpasfilter	8 st
Kondensator	$1\mu F$ 3st och $24nF$ , till gyrot	4 st
Resistor	Till reset	1 st
Kondensator	Till reset och batteri	5 st
Tryckknapp	Till reset	1 st
JTAG ICE mkII	Programmera och emulera program på AVR	1 st
Reflexsensor	Ljusreflexmätare	1 st
MLX90609	Vinkelhastighetssensor (gyro) $300^\circ/s$	1 st

Tabell 4: Lista över hårdvara i sensormodulen

### 4.1.1 Sensorplacering

Sensorerna i avsnitt 4.1 placeras enligt figur 4. På robotens sidor placeras en kort avståndsmätare, grönmärkade i figuren, en i främre och en i bakre änden. Detta är för att åstadkomma god avståndsbedömning så att regleringen kan förbättras. På varsin sida placeras även en lång avståndsmätare, rödmärkade i figuren, som, när de korta inte kan se någon vägg, ska avgöra hur långt bort nästa vägg kan finnas. Detta kommer senare användas i kartläggningsprocessen. På robotens front sitter en kort avståndsmätare som detekterar när roboten måste ändra riktning. På undersidan sitter en reflexsensor, blåmarkerad i figuren, som detekterar när roboten har nått sitt mål.



Figur 4: Sensorplacering

### 4.1.2 Optisk avståndsmätare

Mäter avståndet till väggar med IR-sensorer. När IR-sensorerna mäter avståndet görs detta digitalt i kretsen för att sedan transformera det digitala värdet till en analog spänning. Detta

medför ett högfrekvent brus på ut-signalen, som kommer att filtreras bort med lågpassfiltret. Det sitter två olika sorters avståndsmätare på roboten, en sort för mätning av korta avstånd och en sort för mätning av längre avstånd[1][2].

#### 4.1.3 Kort avståndsmätning

Sensorerna för mätning av korta avstånd är effektiv mellan 4 till 30 cm. Roboten kommer majoriteten av tiden ha väggar inom de intervall, informationen från sensorerna behandlas av mjukvaran och skickas till styrmodulen som reglerar systemet så att roboten är centerad mellan väggarna [1].

#### 4.1.4 Lång avståndsmätning

Sensorerna för mätning av längre avstånd är effektiv mellan 10-80 cm. När roboten inte kan hitta en vägg med kort-avståndssensorerna kommer denna sensorn försöka hitta ett avståndet till väggen. Detta är för att hjälpa till i kartläggningen.[2]

#### 4.1.5 Reflexsensor

Reflexsensorn består av en lysdiod och en ljuskänslig transistor som mäter reflekterat ljus från underlaget. Om underlaget reflekterar lite ljus blir utsignalen hög och vice versa. Målet kommer markeras ut med en svart tejpbit. Den biten av underlaget reflekterar ljus dåligt jämfört med resten av labyrintens underlag. Således kommer utsignalen kalibreras så att hög signal motsvarar att roboten har nått sitt mål

Eftersom sensorn är kalibrerad för ett avstånd på några millimeter mellan transistorn och underlaget kommer sensorn placeras på robotens undersida.

Sensorn kopplas in med matningsspänningen +5 V och jord GND. Utsignalen är analog och kommer att variera mellan 0 och 5 V. [3]

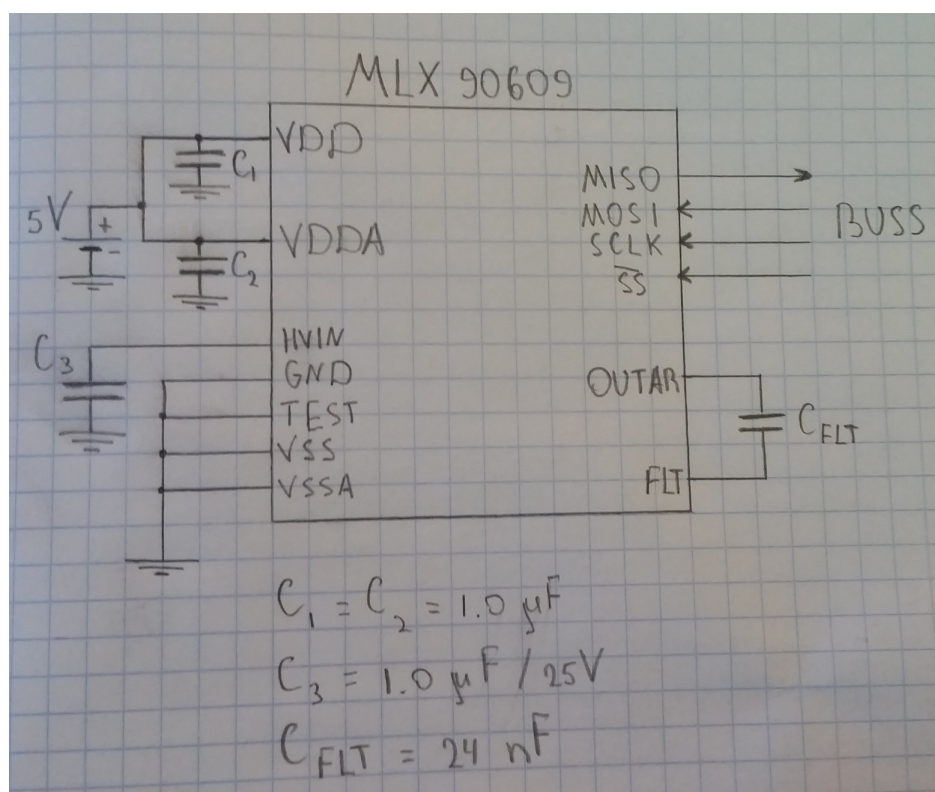
Eftersom att signalen är analog måste den A/D-omvandlas och lagras i minnet i modulens ATmega1284, se figur 6.

#### 4.1.6 ATmega1284p

Mikrokontrollern kommer att användas i huvudsak för att A/D-omvandla sensorsignalerna så att de kan föras vidare till övriga moduler via buss. Detta sker med de 8 inbyggda A/D-omvandlarna.

#### 4.1.7 Gyro

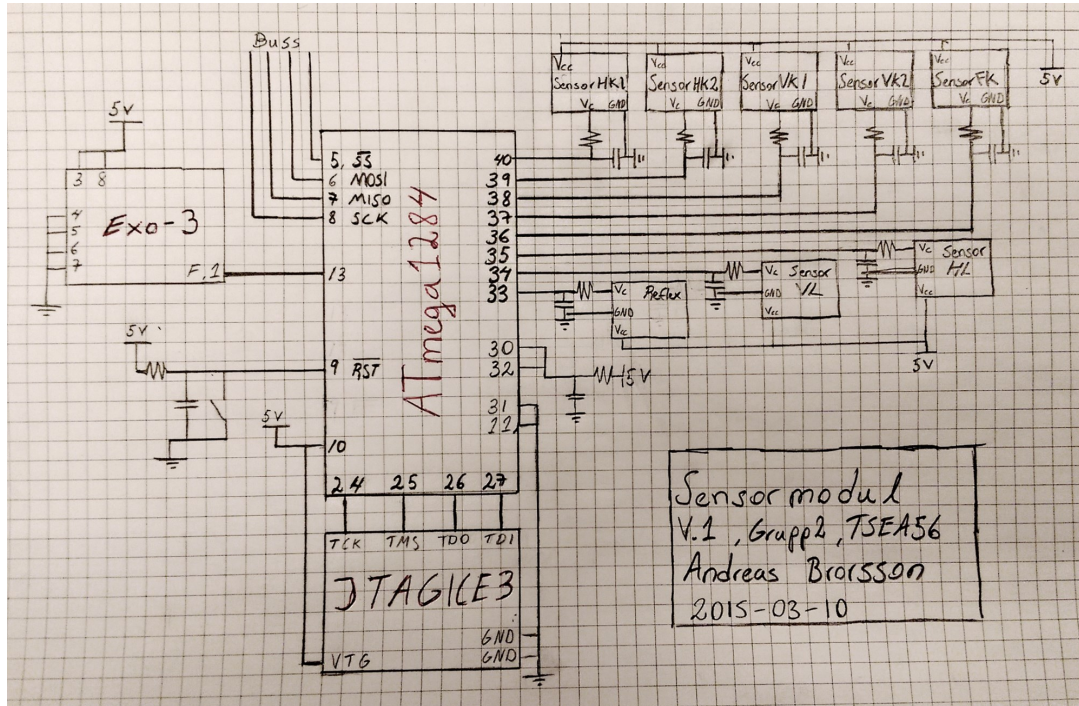
Gyrot är en enskild sensormodul (slave), som har för uppgift att mäta vinkelhastighet när roboten utför en sväng. Den är kopplad direkt via SPI buss till styrmodulen, därför att det är bara den digitala utgången i gyrot som kommer att användas, ej den analoga. Denna gyro-modulen innehåller en EEPROM och en intern A/D-omvandlare, som kan startas av styrmodulen. Antagligen, kommer gyrot att placeras i mitten av roboten.



Figur 5: Kopplingsschema på gyro.

#### 4.1.8 Kopplingsschema

För att uppkopplingen skall gå smidigt i labbet bifogas ett kopplingsschema för hårdvaran. Som synes räcker precis 8 A/D-portar på ATmega1284 för att alla sensorer ska få plats.



Figur 6: Kopplingsschema på sensormodulen.

#### 4.2 Mjukvara

Vid uppstart ska sensormodulen initieras. Detta innebär att in- och utgångar konfigureras att skicka och ta emot, att ADC konfigureras så att den kan ta emot signaler samt att ställa in avbrottsrutinen i processorn så att vi kan ta emot avbrott från ADC och SPI.

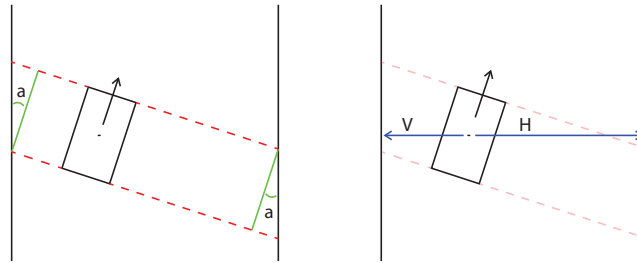
Sensormodulen kommer att genomföra enkel informationsbehandling av sensorinformationen för att styrmodulen ska få mer förfinad data. På så vis sparar man dels klockcykler i styrmodulen men minimerar även antalet transporter över bussen. Informationen från sensormodulen kommer att komprimeras till så få bitar som möjligt för att optimera informationsöverföringen.

##### 4.2.1 Avståndssensor

ADC:n konverterar varje analog signal från avståndssensorerna till digitala. Genom att testa vilka avstånd som ger vilken digital utsignal formas ett register som konverterar digital signal till ett avstånd i centimeter.

När avståndet ligger sparat i minnet kommer ATmega1284p genomföra enkla beräkningar som ger oss vår vinkel och offset från mitten av gången. Den informationen kommer komprimeras på så få bitar som möjligt och skickas till sensormodulen. Om man till exempel sparar vinkeln på de första bitarna sparar man offset på efterföljande.





Figur 7: Vad sensormodulen beräknar.

#### 4.2.2 Reflexsensor

Efter att reflexsensorns signal har genomgått AD-omvandling kommer mjukvaran att kontrollera om den uppmätta spänningen överensstämmer med den spänningen som genereras för en svart tejpbit. Detta värde bestäms genom testning och värdet matas in i mjukvaran. Resultatet från jämförelsen - om vi befinner oss över tejpbiten eller ej - skickas vidare till styrmodulen via MISO. Den kan läggas som en bit efter informationen från avståndssensorerna.

#### 4.2.3 Gyro

Gyromodulen ska initieras genom att in- och utgångar konfigureras på ett lämpligt sätt, SPI aktiveras, samt avbrottsrutiner från den interna A/D-omvandlaren tillåts. Data från styrmodulen (operations kod) överförs seriellt till MLX90609 med 8 eller 16 bitars ord, och svaret fås med 16 bitars ord. Data till och ifrån MLX90609 skickas med MSB först. Den data som skickas är i digitalt format, därför kan den direkt användas i styrmodulen.

#### 4.2.4 Pseudokod

Nedan följer pseudokod för main loop.

```
%Configure in-/output
%Configure ADC
%Configure interrupt
loop
  %Höj avbrottsnivån
  %Läs av sensorer
  %Sänk avbrottsnivån
  pause;
  jmp loop;
```

#### Algorithm 1: Pseudokod huvudloop sensormodul

Nedan följer pseudokod för kortdistanssensorerna.

```

if short_current_conversion_failed then
  | %convert again
else
  | %convert current to distance;
end
if no_distance_left then
  | % fire long sensor left;
else
  | %convert then send distance and angle
end
if no_distance_right then
  | %fire long sensor right;
else
  | %convert then send distance and angle
end

```

**Algorithm 2:** Pseudokod kort sensor

Nedan följer pseudokod för långdistanssensorerna.

```

if long_current_conversion_failed then
  | %convert again
else
  | %convert current to distance;
end
if no_long_distance_left then
  | %send unknown;
else
  | %convert then send distance and angle
end
if no_long_distance_right then
  | %send unknown;
else
  | %convert then send distance and angle
end

```

**Algorithm 3:** Pseudokod lång sensor

Nedan följer pseudokod för reflexsensorn.

```

if reflex_current_conversion_failed then
  | %convert again
else
  | %convert current to binary
end
if _no_black_tape then
  | %send 0
else
  | %send 1
end

```

**Algorithm 4:** Pseudokod reflexsensor



## 5 Styrmodul

Styrmodulen tar in data både från sensormodulen och från kommunikationsmodulen beroende på om roboten ligger i autonomt eller manuellt läge. När roboten är i autonomt läge får den strömmande data från sensormodulen som den sedan jämför med kartan som roboten själv bygger upp. Med informationen den fått och har görs ett beslut om vart den ska. När roboten vet vart den ska används reglerteknik för att hålla sig centrerad och att svänga utan att fastna mot väggar.

### 5.1 Hårdvara och komponentbudget

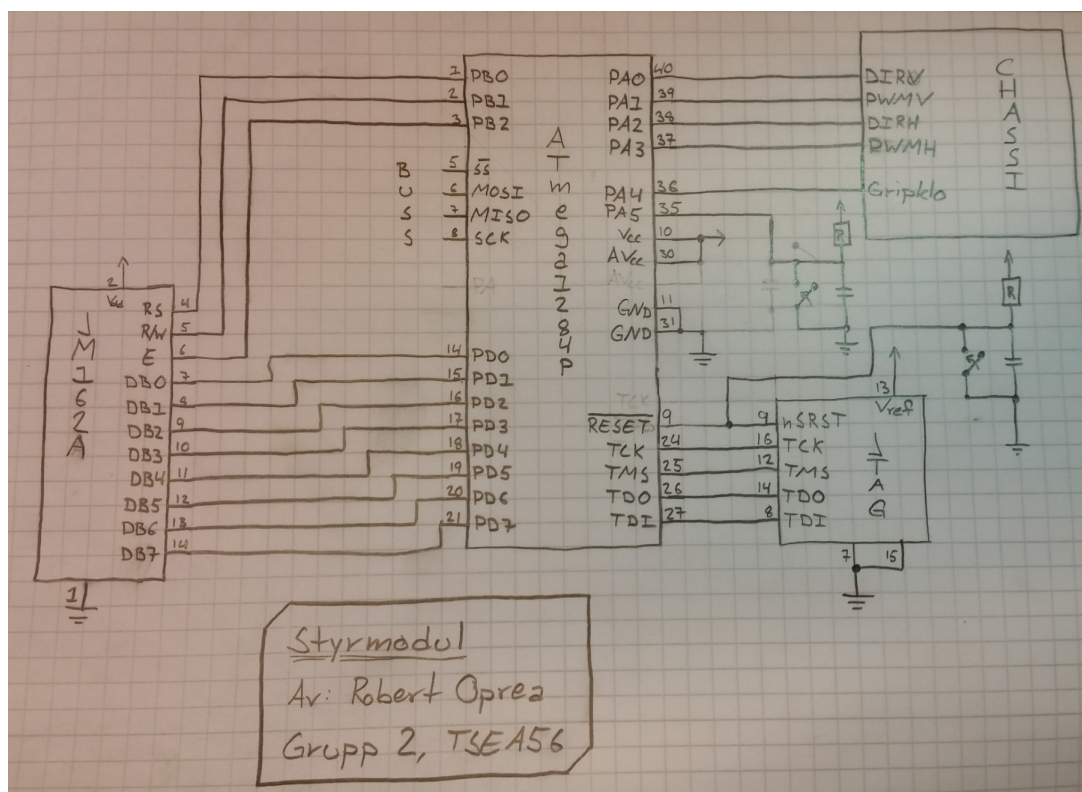
Nedan listas den hårdvara som ingår i styrmodulen.

Komponent	Specifikation	Antal
ATmega1284p	Microcontroller 128 kB	1 st
Gripklo	Gripklo för att bära mindre föremål	1 st
JM162A	LCD-display med 2 rader x 16 tecken	1 st
Tryckknapp	En tryckknapp för omställning mellan manuell och autonom styrning, och en till reset	2 st
Kondensator	En till tryckknappen och en till reset	2 st
Resistor	En till tryckknappen och en till reset	2 st

Tabell 5: Styrmodulens hårdvara

### 5.2 Kopplingschema

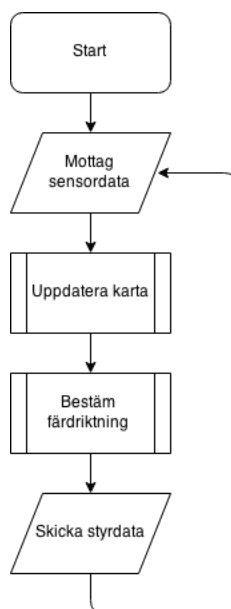
I figur 8 nedan ses kopplingsschemat för styrmodulen. Att notera är att portarna på mikrokontrollern räcker till.



Figur 8: Kopplingsschema för styrmodulen.

### 5.3 Programkod

Figur 9 illustrerar hur programkoden för styrmodulens huvudloop är tänkt att fungera.



Figur 9: Flödesschema för styrmodulens huvudloop.

### 5.3.1 Mottagning av sensordata

Sensordata om omgivningen skickas av sensormodulen via bussen, och ger värden på robotens vinkel samt offset. Dessa data används i nästföljande steg, dvs. vid uppdateringen av kartan samt när färdriktningen bestäms.

### 5.3.2 Uppdatering av karta

Det är känt att omgivningen kommer utgöras av en labyrint vars storlek är maximalt 6 x 6 meter och där väggar och korridorer är uppbyggda av rutor på 40 x 40 cm (för mer detaljer se tävlingsreglerna). Utifrån dessa antaganden kan kartan beskrivas genom ett 15 x 15 rutnät. Varje ruta kan sedan vara antingen startpunkt, okänd, vägg, väg, korsning eller mål, och utöver detta markeras även robotens position. Kartan lagras i en matris där varje element består av tre bitar som säger vad respektive ruta är för något.

Med hjälp av sensordata kan sedan roboten avgöra när den har förflyttat sig 40 cm framåt och således befinner sig i en ny ruta. Då uppdateras dels om rutan den befinner sig i är en korsning, eller om den är målet, dels de tre rutorna som angränsar till roboten (till vänster, framåt samt till höger) uppdateras från okänd till antingen väg eller vägg.

### 5.3.3 Bestämmande av färdriktning

Bestämmandet av färdriktningen kan sägas bestå av två delar, kortsiktigt respektive långsiktig färdriktning. Den kortsiktiga färdriktningen avser till att hålla roboten rak och centrerad med avseende på korridoren den befinner sig i. Detta åstadkoms med hjälp av en PD-reglering med avseende på robotens vinkel samt offset, och görs utifrån den mottagna sensordatan.

Den långsiktiga färdriktningen bestämmer till vilken ruta roboten ska färdas. I utforskningsfasen (dvs. innan målet lokaliserats) innebär detta t.ex. om roboten ska åka till rutan framför eller till en ruta till höger, och i så fall att den roterar sig rätt. Om roboten hamnat i en återvändsgränd ska den navigera sig tillbaka till senast besökta korsning, även om denna ligger flera rutor bort. I undsättningsfasen (dvs. när målet lokaliserats) ska roboten navigera tillbaka till startpunkten via den kortaste vägen.

### 5.3.4 Sändning av styrdata

När beslut har fattats om robotens agerande skickas sedan styr signaler till dens motorer så att den färdas i önskad riktning.

## 6 PC

För att användaren ska kunna styra och kommunicera med roboten kommer det att implementeras ett användargränssnitt där all relevant data, så som karta, styrdata, styrbeslut och mätdata från sensorn redovisas för användaren. Kommunikation mellan dator och robot (kommunikationsmodul) sker via bluetooth.

## 6.1 Funktioner

Programmet kommer att behövas för att:

- Styra roboten i manuellt läge.
- Kontinuerligt visa upp mätdata från roboten
- Redovisa kartan

### 6.1.1 Manuell styrning

Med hjälp av programmet kommer man kunna styra roboten manuellt när roboten har satts i läget för manuell styrning. Roboten ska kunna röra sig på samma sätt som den gör i autonomt läge, det vill säga, framåt, framåt vänster framåt höger, bakåt, samt rotera vänster och höger.

### 6.1.2 Mätdata

Användaren kommer att kunna be om mätdata från robotens sensormodul. Den data som kommer finnas tillgänglig är:

- Avstånd till en vägg, framåt eller i sidled
- Om roboten har hittat målet (tejpbiten) eller inte
- Vinkelhastighet, vid rotation i korsningar.

### 6.1.3 Karta

Användaren kommer att kontinuerligt se kartan uppdateras under robotens körning, fram tills dess att roboten har hittat en optimal väg mellan start och mål. Därefter kommer man endast se hur roboten rör sig i labyrinthen på den karta som finns.

## 7 Implementeringsstrategi

Först och främst kommer

- det verifieras att det räcker med pinnar (in- och utgångar) på varje processor i varje modul, för att hantera all data och signaler från alla komponenter som kommer att kopplas. Detta kan uppskattas genom att rita kopplingsscheman till olika delsystem.
- funktionaliteten av hårdvaran testas med till exempel en logikanalysator, för att avgöra om komponenter inte är trasiga.
- seriell överföring implementeras genom att skriva programkod för att skicka data seriellt mellan modulerna.
- busskommunikation mellan moduler testas och det verifieras att den funkar som den ska.
- alla moduler testas kontinuerligt under uppbyggandet och programmeringen, på så sätt att de kan även fungera var för sig.

## 8 Intern kommunikation

Kommunikationen emellan modulerna kommer att hanteras med en SPI buss. SPI protokollet tillåter en master och flera slaves.

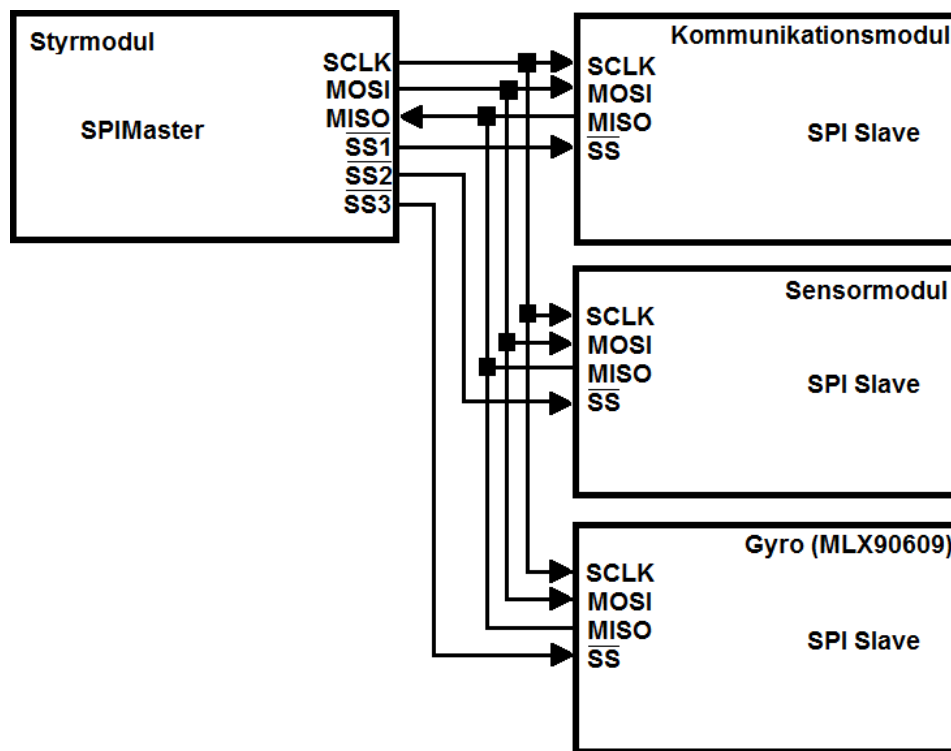
I denna konfiguration kommer Styrmodulen att vara master, medan de andra två modulerna, Sensor- och Kommunikationsmodulen, samt gyrot (MLX90609) är slaves.

### 8.1 Beskrivning av SPI protokollet

Kommunikationen mellan enheterna kommer att se ut enligt figur 10. Styrmodulen kommer att skicka en klockpuls via SCLK till alla slaves; alla SPI signaler kommer att synkroniseras enligt denna.

Varje slave kommer att få en  $SS_n$  (Slave Select Signal) enligt mastern, där mastern enligt denna väljer vilken enhet den ska kommunicera med. När  $SS_n$  bli låg så börjar kommunikation mellan mastern och slave  $n$ .

När en kommunikation har startats kommer mastern att skicka data till MOSI (Master out Slave in), samt sampla data från MISO (Master in Slave out) med vald slave. [4]



Figur 10: Blockschemat över kommunikation mellan enheter

Det finns fyra kommunikationsinställningar som bestäms enligt två parametrar: CPOL (eng. clock polarity) och CPHA (eng. clock phase).

De enheter som kommunicerar med varandra, det vill säga master och en slave, måste ha samma parametrar på CPOL, CPHA, samt samma klockfrekvens på SCLK för att kommunikation ska kunna ske mellan enheterna. Om olika slaves har olika konfigurationer på

CPOL och CPHA, samt SCLK frekvens, behöver styrmodulen konfigurera om dessa innan kommunikation med respektive slaveenhet.

## **8.2 SPCR - SPI Control Register**

ATMega1284P har ett register, SPCR, som avgör hur SPI kommunikationen konfigureras. En kort beskrivning av bitarna i registret följer nedan.

### **8.2.1 SPIE- SPI Interrupt Enable**

Biten exekverar SPI avbrottet om SPIF biten i SPSR registern är satt och om Global Interrupt Enable biten i SREG är satt.

### **8.2.2 SPE - SPI Enable**

Biten sätts till ett för att tillåta operationer via SPI.

### **8.2.3 DORD - Data Order**

Om biten sätts till ett, skickas LSB först, annars MSB.

### **8.2.4 MSTR - Master/Slave select**

Biten väljer mellan Master SPI mode och Slave SPI mode genom att sättas respektive till ett eller noll. Om SS är konfigurerad som ingång och det kommer in en nolla, kommer MSTR biten att sättas till noll av hårdvaran. För att aktivera Master SPI mode igen, måste biten sättas till ett.

### **8.2.5 CPOL - Clock Polarity**

Biten bestämmer hur klockan ska synkronisera signalerna.

Om CPOL sätts till en logisk nolla innebär detta att den första signalen sker när klockan går hög.

Om CPOL sätts till en logisk etta innebär detta att den första signalen sker när klockan går låg. [5]

### **8.2.6 CPHA - Clock Phase**

Biten bestämmer när data ska samplas.

Om CPHA sätts till en logisk nolla samplas signalen på leading edge, och ställer in på trailing edge.

Om CPHA sätts till en logisk etta så sker komplementet, dvs sampling på trailing edge och inställning på leading edge. [5]

### **8.2.7 SPR1, SPR0 - SPI Clock Rate Select 1 and 0**

De två bitar bestämmer SCK takten endast om ATMega1284P är konfigurerad som Master. SPR1 och SPR0 kan inte påverka Slave.

### 8.3 SPSR - SPI Status Register

ATMega1284P har även ett statusregister som innehåller flaggor för att hantera avbrott och konfigurera klockfrekvensen. De viktigaste bitar beskrivs nedan.

#### 8.3.1 SPIF - SPI Interrupt Flag

SPIF flaggan sätts när en seriel överföring är klar. Avbrottet genereras om SPIE i SPCR är satt till logisk ett och globala avbrott är tillåtna/aktiverade. Om SS är konfigurerad som ingång och det kommer in en låg signal, då det är SPI Master mode, kommer SPIF flaggan att sättas till ett. SPIF sätts till noll av hårdvaran då en avbrottsvektor exekveras.

Alternativt kan SPIF biten sättas till noll, genom att först läsa SPSR med SPIF satt, och sedan komma åt SPI Data Registern (SPDR).

#### 8.3.2 WCOL - Write Collision Flag

Biten sätts av hårdvaran om SPDR skrivs under data överföring. WCOL (samt SPIF) biten nollställs genom att först läsa SPSR med WCOL satt, och sedan komma åt SPDR.

#### 8.3.3 SPI2X - Double SPI Speed Bit

Om biten sätts till logisk ett, kommer SCK takten (frekvensen) att fördubblas när SPI är i Master mode.

### 8.4 Dataöverföring

Modulerna kommer att skicka eller ta emot data på begäran av styrmodulen. När en slave select går igång kommer kommunikation att påbörjas förutsatt att CPOL, CPHA och SCLK frekvensen stämmer överens mellan styrmodul och nuvarande slave.

Eftersom styrmodulen är den enda enhet som kan påbörja en kommunikation kommer det vid vissa tillfällen innebära att en slave-modul inte har begärd data tillgänglig. Då måste en definierad felkod återsändas, och styrmodulen får göra ett nytt försök vid nästa kommunikation.

Notera att sensormodulen aldrig kommer att ta emot data från styrmodulen.

## Referenser

- [1] -. (Hämtad 2015-03-07). Vanheden: kort avståndssensor, URL: <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/sharp-GP2Y0A41SK0F.pdf>.
- [2] —, (Hämtad 2015-03-07). Vanheden: lång avståndssensor, URL: <https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/gp2y0a21.pdf>.
- [3] —, (Hämtad 2015-03-07). Vanheden: reflexsensor, URL: [https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/reflex\\_sensor.pdf](https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/reflex_sensor.pdf).
- [4] F. Leens, "An introduction to i2c and spi protocols", *Instrumentation & Measurement Magazine, IEEE*, s. 9–10, 2009, issn: 1094-6969. DOI: 10.1109/MIM.2009.4762946.
- [5] Y. Tambi och M. Prasad. (2013). The spi of the avr. Y. Tambi, utg., URL: <http://maxembedded.com/2013/11/the-spi-of-the-avr/> (hämtad 2015-03-10).



## Appendix

### Appendix A

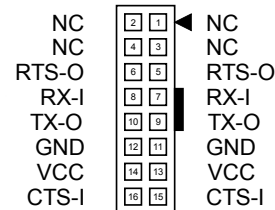
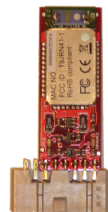
# FireFly

## Bluetooth Modem - BlueSMiRF Gold

### Specifications

- \* Class 1 Bluetooth Radio Modem
- \* Built in antenna
- \* Low power consumption: 25mA avg
- \* Operating Coltage: 3.3V-6V
- \* Serial communications: 2400-115200bps
- \* Frequency 2.4~2.524 GHz
- \* Hardy frequency hopping scheme
- \* Encrypted connection

### Modem & Connector



### Description

This modem works as a serial (RX/TX) pipe. Any serial stream from 2400 to 115200bps can be passed seamlessly from computer to target. The modem can be powered from 3.3v up to 6V and all signal pins are 3V-6V tolerant. No level shifting is required. However, connecting the modem to the serial port of a computer requires an RS232 to TTL converter circuit.

The modem has two LEDs, one red and one green. A flashing red LED means the modem is waiting for a bluetooth connection. A steady green LED means the modem has an active bluetooth connection.

The modem has duplicated signals on both sides of its connector. Only one of each signal should be used, however both of the ground (GND) pins and both power (VCC) pins should be connected for proper power supply.

Pin	Signal	Dir	Description
1, 2	NC	--	No Connection
3, 4	NC	--	No Connection
5, 6	RTS-O	Out	Request To Send (for hand shaking)
7, 8	RX-I	In	Receive, serial data input
9, 10	TX-O	Out	Transmit, serial data output
11, 12	GND	--	Ground, 0V
13, 14	VCC	--	Power supply, 3.3V-6V
15, 16	CTS-I	In	Clear To Send (for hand shaking)

## Appendix B



Datum 981112

**PRODUKTINFORMATION****HÄMTFAX**  
08-580 941 14**FAX ON DEMAND**  
+46 8 580 941 14**INTERNET**  
<http://www.elfa.se>

TEKNISK INFORMATION 020-75 80 20    ORDERTEL 020-75 80 00    ORDERFAX 020-75 80 10

TECHNICAL INFORMATION +46 8 580 941 15    ORDERPHONE +46 8 580 941 01    ORDERFAX +46 8 580 941 11

— Vi reserverar oss mot fel samt förbehåller oss rätten till ändringar utan föregående meddelande —

ELFA artikelnr.

Antal sidor: 03

74-540-02 EXO3/12,0000MHz krist osc

74-540-85 EXO3/16,0000MHz krist osc

74-540-28 EXO3/12,2880MHz krist osc

74-541-01 EXO3/18,4320MHz krist osc

74-540-44 EXO3/12,8000MHz krist osc

74-541-27 EXO3/19,6608MHz krist osc

74-540-51 EXO3/14,31818MHz kristOsc

74-541-43 EXO3/20,0000MHz krist osc

74-540-69 EXO3/14,7456MHz krist osc

## IQEXO-3

ISSUE 3; 12 JANUARY 1996

### Delivery Options

- Common frequencies are available from stock. Please see p122 for details

### Output Compatibility

- HCMOS

### Holder Style

- 8-pin DIL compatible plastic encapsulated

### Frequency Stability

- $\pm 100\text{ppm}$

### Operating Temperature Range

- $-10$  to  $70^\circ\text{C}$

### Storage Temperature Range

- $-55$  to  $125^\circ\text{C}$

### Mechanical Specification

- Mechanical Shock: 1000G,  $\frac{1}{2}$  sine wave impulse for 0.35ms in three directions
- Resistance to Soldering Heat: Leads dipped in soldering bath @  $260^\circ\text{C}$  for 10 secs
- Resistance to Solvent: Dipping in fluoric solvent for 90 secs
- Solderability: 90% of soldered surface masked with new solder after dipping in solder bath @  $230^\circ\text{C}$  for 3 secs
- Vibration: 10 to 55Hz 1.5mm displacement, 50 to 2000Hz 20G acceleration, 1 hour in each of three mutually perpendicular planes

### Handling & Operational Precautions

- Observe anti-static handling precautions
- A decoupling capacitor of  $<0.01\mu\text{F}$  should be connected between Pin 8 and Ground

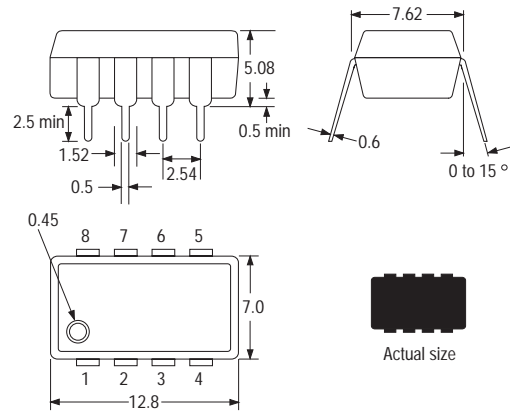
### Marking

- Model number
- Frequency

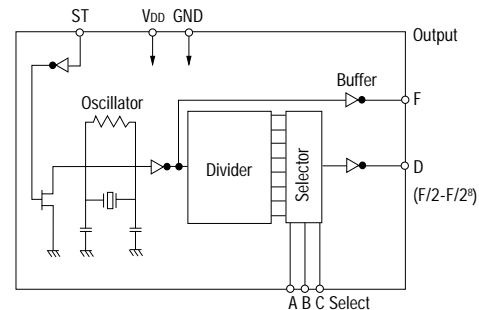
### Minimum Order Information Required

- Frequency + Model Number

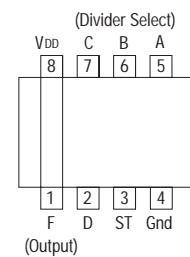
### Outline (scale 2:1)



### Circuit Diagram



### Terminal Connections



**Electrical Specification – maximum limiting values**

Supply Voltage	Input Voltage		Output Voltage		Rise Time ( $t_r$ )	Fall Time ( $t_f$ )	Current		Start-up Time		Duty Cycle	Aging
	High	Low	High	Low			Supply	Standby	Supply	Standby		
5.0V±1V	3.6V min	0.8V max	4.5V min	0.5V max	15ns max	15ns max	20mA	10.0μA	1.5ms	1.5ms	40/60%	±5ppm

**Terminal Connections**

- 1: Crystal frequency output ( $f_o$ )
- 2: Divided frequency output ( $f_o/2^n$ )
- 3: Standby terminal; Logic '1' to pin 3 enables oscillator output; logic '0' to pin 3 disables oscillator output; when disabled the oscillator output goes to the high impedance state
- 4: Ground
- 5: Programs division ratio
- 6: Programs division ratio
- 7: Programs division ratio
- 8: Power supply voltage

**Frequency Programming**

Input				Output	
Select			Pin 3 ST	Pin 1 Fundamental Oscillation	Pin 2 Divided Frequency
Pin 7 C	Pin 6 B	Pin 5 A			
L	L	L	H	$f_o$	$f_o+2$
L	L	H	H	$f_o$	$f_o+4$
L	H	L	H	$f_o$	$f_o+8$
L	H	H	H	$f_o$	$f_o+16$
H	L	L	H	$f_o$	$f_o+32$
H	L	H	H	$f_o$	$f_o+64$
H	H	L	H	$f_o$	$f_o+128$
H	H	H	H	$f_o$	$f_o+256$

**Crystal Frequency (MHz)**

Fundamental Frequency ( $f_o$ )	Divided Frequency ( $f_o/2^n$ )							
	$f_o$ (MHz)	$f_o+2$ (MHz)	$f_o+4$ (MHz)	$f_o+8$ (MHz)	$f_o+16$ (kHz)	$f_o+32$ (kHz)	$f_o+64$ (kHz)	$f_o+256$ (kHz)
	12.0	6.0	3.0	1.50	750.0	375.0	187.50	46.8750
	12.2880	6.1440	3.0720	1.5360	768.0	384.0	192.0	48.0
	12.80	6.40	3.20	1.60	800.0	400.0	200.0	50.0
	14.318180	7.159090	3.579545	1.789772	894.880	447.440	223.720	55.930
	14.50	7.250	3.6250	1.81250	906.250	453.1250	226.5620	56.640
	14.74560	7.37280	3.68640	1.84320	921.60	460.80	230.40	57.60
	14.91050	7.45520	3.72760	1.86380	931.9060	465.9530	232.9760	58.2440
	15.0	7.50	3.750	1.8750	937.50	468.750	234.3750	58.5930
	15.360	7.680	3.840	1.920	960.0	480.0	240.0	60.0
	15.97440	7.98720	3.99360	1.99680	998.40	499.20	249.60	62.40
	16.0	8.0	4.0	2.0	1000.0	500.0	250.0	62.50
	16.1280	8.0640	4.0320	2.0160	1008.0	504.0	252.0	63.0
	16.2570	8.125850	4.06425	2.032125	1016.0620	508.030	254.0150	63.5030
	16.3840	8.1920	4.0960	2.0480	1024.0	512.0	256.0	64.0
	17.734476	8.867238	4.433619	2.216809	1108.4	554.20	277.10	69.2750
	18.4320	9.2160	4.6080	2.3040	1152.0	576.0	288.0	72.0
	19.09090	9.545450	4.772725	2.386362	1193.1810	596.590	298.2950	74.5730
	19.20	9.60	4.80	2.40	1200.0	600.0	300.0	75.0
	19.66080	9.83040	4.91520	2.45760	1228.80	614.40	307.20	76.80
	20.0	10.0	5.0	2.5	1250.0	625.0	312.50	78.1250
	20.480	10.240	5.120	2.560	1280.0	640.0	320.0	80.0
	21.477270	10.738635	5.369317	2.684650	1342.3290	671.1640	335.5820	83.89550
	22.0	11.0	5.50	2.750	1375.0	687.50	343.750	85.93750
	22.11840	11.05920	5.52960	2.76480	1382.40	691.20	345.60	86.40
	24.0	12.0	6.0	3.0	1500.0	750.0	375.0	93.750
	24.5760	12.2880	6.1440	3.0720	1536.0	768.0	384.0	96.0

## ■ ATmega1284P

Figure 17-2 shows a block diagram of the clock generation logic.

The diagram illustrates the XCK pin control logic. It starts with the **UBRR** register, which feeds into a **Prescaling Down-Counter**. The output of the counter is **fosc**, which is divided by **UBRR+1** and then by **/2**, **/4**, and **/2** to produce the **U2X** clock signal. The **U2X** signal is then used to generate **txclk** and **rxclk** signals. The **U2X** signal is also used to generate the **UMSEL** signal. The **U2X** signal is also used to generate the **DDR\_XCK** signal. The **DDR\_XCK** signal is used to control the **XCK Pin** driver. The **XCK Pin** driver outputs **xcki** and **xccko** signals. The **xccko** signal is used to generate the **OSC** signal, which is used to clock the **Prescaling Down-Counter**. The **xcki** signal is used to clock the **Sync Register**. The **Sync Register** outputs the **UCPOL** signal, which is used to clock the **Edge Detector**. The **Edge Detector** outputs the **UMSEL** signal. The **UMSEL** signal is used to control the **U2X** clock signal.

<b>txclk</b>	Transmitter clock (Internal Signal).
<b>rxclk</b>	Receiver base clock (Internal Signal).
<b>xcki</b>	Input from XCK pin (internal Signal). Used for synchronous slave operation.
<b>xcko</b>	Clock output to XCK pin (Internal Signal). Used for synchronous master operation.
<b>f<sub>osc</sub></b>	XTAL pin frequency (System Clock).

**BAUD** Baud rate (in bits per second, bps)