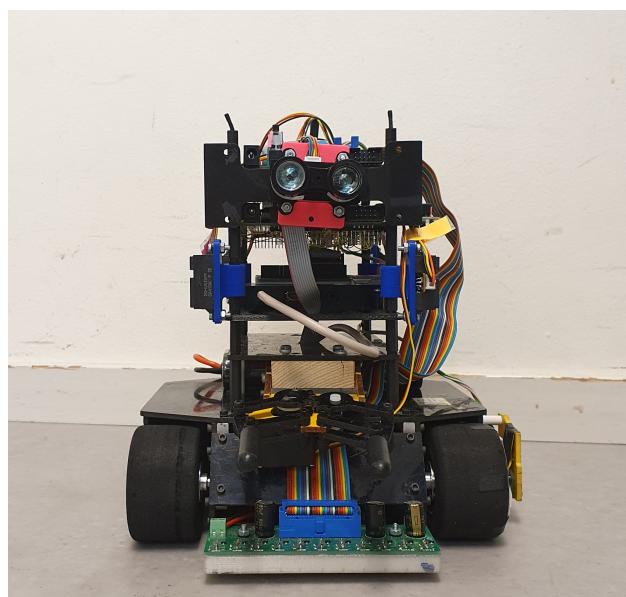


Kandidatuppsats i elektronik
Institutionen för systemteknik, Linköpings universitet, 2022

Konstruktion av en autonom undsättningsrobot

Design of an Autonomous Rescue Robot

**Alice Nicklasson Cedbro, Edvin Leuchovius,
Filip Johansson, Gunnar Arctaedius, Johanna Carlson,
Lovisa Byman, Maja Boström**



Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Hemsida: <http://www.isy.liu.se/edu/kurs/TSEA56/>

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Gunnar Arctaedi		gunar889@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Maja Boström		majbo451@student.liu.se

INNEHÅLL

1	Inledning	1
1.1	Syfte	2
2	Problemformulering	3
2.1	Kravbild	3
2.2	Utmaningar	3
3	Kunskapsbas	4
4	Genomförande	5
5	Teknisk beskrivning	6
5.1	Förstudier	6
5.2	Sensormodul	6
5.3	Styrmodul	8
5.4	Kommunikationsmodul	10
5.5	Datormodul	11
5.6	Integrering av moduler	12
6	Resultat	13
7	Slutsatser	14
7.1	Sensormodulen	14
7.2	Styrmodulen	15
7.3	Kommunikations- och datormodulen	15
	Referenser	16

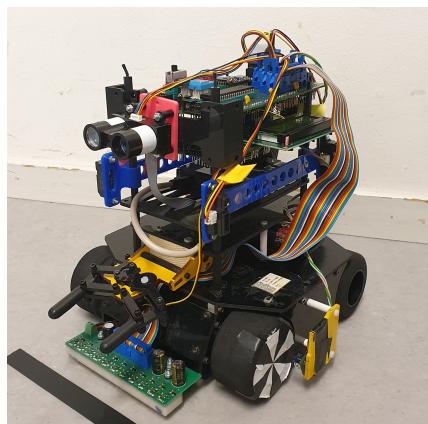
Appendix	18
A Projektdirektiv	19
B Kravspecifikation	21
C Ban- och tävlingsspecifikation	34
D Systemskiss	41
E Projektplan	60
F Designspecifikation	81
G Förstudie	113
G.1 Sensorer	113
G.2 Reglering	138
G.3 Kommunikation	156
H Användarmanual	187
I Efterstudie	207
J Teknisk Dokumentation	218

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2021-05-18	Första utkast	Alla	Alla

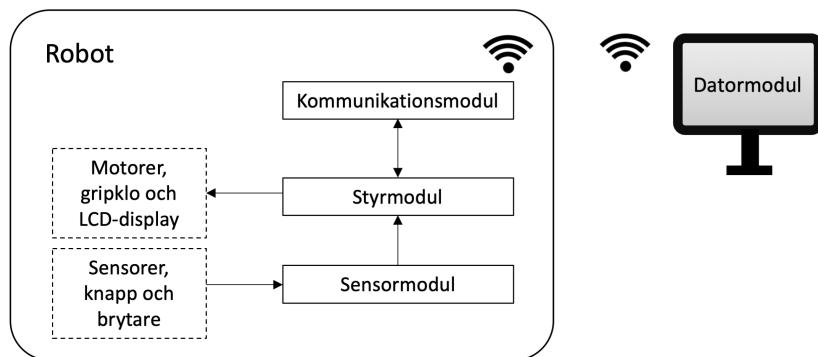
1 INLEDNING

Detta dokument beskriver de resultat som har uppnåtts i samband med ett kandidatprojekt i elektronik vid Linköpings universitet, och hur tiden som är avsatt för detta projekt har disponerats. Projektgruppens uppgift har varit att utveckla en autonom undsättningsrobot. Robotten är uppbyggd av ett chassi samt fyra stycken moduler. Tre av dessa moduler, sensormodulen, styrmodulen och kommunikationmodulen sitter på chassit och den fjärde, datormodulen, består av ett gränssnitt som kan användas på en fristående dator. Figur 1 visar robotens slutliga konstruktion.



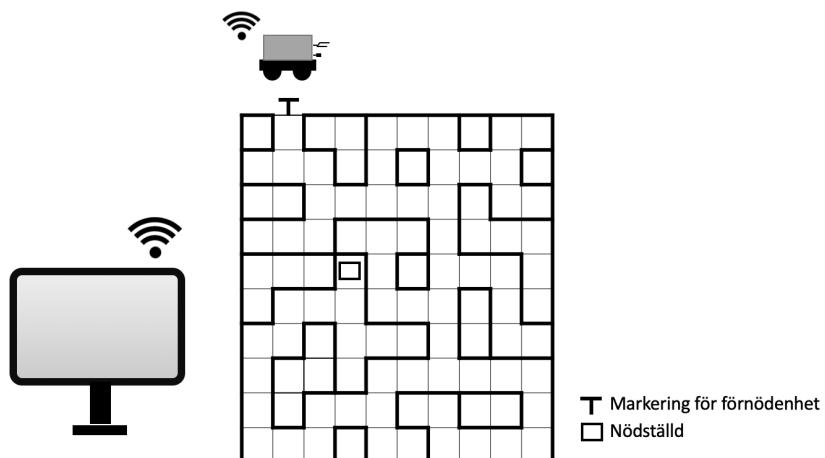
Figur 1: Slutlig konstruktion av undsättningsroboten.

Sensormodulen består av kommunikation mellan alla sensorer och en processor och skickar vidare bearbetad sensor-data till styrmodulen. Styrmodulen hanterar styrning och reglering av roboten samt planerar körning och kartlägger området. All information som styrmodulen har tillgång till skickas vidare till kommunikationsmodulen som hanterar kommunikationen med datormodulen. Slutligen presenteras all relevant data, såsom karta, reglerparametrar och data från sensorer på användargränssnittet i datormodulen. Figur 2 visar ett översiktligt flödesschema för kommunikationen i robotten.



Figur 2: Kommunikationsflöde i undsättningsroboten.

Robotens uppdrag var uppdelat i två faser. Den första fasen innefattade att söka igenom en byggnad efter nödställda och i samband med detta, hitta kortaste vägen till de nödställda från ingången. Figur 3 visar roboten i en skiss av miljön den är designad att utföra sitt uppdrag i. I den andra fasen gällde det att leverera en förnödenhet från ingången till de nödställda. Detta skulle göras på kortast möjliga tid.



Figur 3: Skiss av roboten i omgivningen uppdraget skulle utföras i.

Att uppdraget är applicerbart på verkligheten gör det till ett intressant uppdrag. Projektarbetet skulle kunna användas som grund för utveckling av en autonom undsättningsrobot för verkliga situationer i större skala. Vidare är uppdraget intressant då en autonom robot kan anpassas till många olika typer av situationer; därmed kan den utvecklade roboten modifieras för att användas inom fler områden än undsättning. Dessutom är automation generellt sätt ett högaktuellt ämne.

En stor utmaning med uppdraget var att kartlägga byggnaden roboten skulle kunna utföra sin uppgift i. Det krävdes mycket arbete för att implementera algoritmen bakom detta, där en ytterligare stor utmaning var att kartläggningen krävde mycket datamine för lagring. Ytterligare en utmaning var att andra fasen av uppdraget skulle genomföras på kortast möjliga tid, vilket krävde noggranna avvägningar mellan körhastighet, reglering, sensorsampling och algoritm-hastighet.

1.1 Syfte

Syftet med kappan är att ge en överblick av det genomförda projektet. Kappan ska ge läsaren förståelse för projektets helhet och har som syfte att ge läsaren en inblick i hur arbetet genomförts. Denna rapport kommer behandla det arbete projektgruppen gjort i samband med projektets olika faser. Det kommer redogöras mer om problemformuleringen och på vilka sätt den var utmanande och intressant. Innan utvecklingsfasen i projektet fördjupade sig projektgruppen på olika områden som rörde de tre modulerna; sensor-, styr- och kommunikationsmodulen. Dessa fördjupningar är de grundkunskaper som erhölls och de kommer tas upp kortfattat. Rapporten kommer även gå in på utvecklingen av roboten. Till sist beskrivs slutprodukten, resultatet i projektet och slutsatserna kring arbetet.

2 PROBLEMFORMULERING

Gruppen erhöll i början av projektet ett projektdirektiv för att kunna producera en robot som effektivt kan lokalisera, och sedan förse nödställda personer i en byggnad med förnödenheter [1].

2.1 Kravbild

Den framtagna undsättningsrobotens uppdrag var att med snabbhet lokalisera en nödställd och därefter kunna leverera en förnödenhet till den nödställda på snabbast möjliga tid. Beställarens önskan utöver detta var att roboten skulle vara moduluppbryggd i tre moduler: kommunikationsmodul, styrmodul och sensormodul. Detta för att i ett senare skede enkelt kunna byta ut en modul mot en annan. Utöver robotens tre moduler fanns även en önskan om att trådlöst kunna övervaka körningen och därvid tillkom en fjärde modul, datormodulen, där en användare från sin dator kan se intressanta data under körning. Ytterligare ett önskemål var att varje modul skulle innehålla minst en processor. Övriga funktionalitetskrav samt krav på projektets arbetsgång har beskrivits i ”TSEA56: Projektdirektiv för en undsättningsrobot”. [1]

Utifrån beställarens direktiv togs en ban- och tävlingsspecifikation [2] samt en kravspecifikation [3] fram. Förstnämnda utformades tillsammans med andra projektgrupper vilka hade samma uppdrag och specificerar hur robotens omgivning skulle byggas upp, hur den tävling beställaren utlyst mellan grupperna skulle gå till samt vilka regler som skulle gälla för denna.

Kravspecifikationen listar i sin tur generella krav på systemet samt mer specifika krav på respektive modul. Kraven togs fram av projektgruppen i dialog med beställare för att undvika otydligheter. De numrerades, gavs en kravversion som angav om det reviderats eller ej och formulerades så att de skulle vara enkla att verifiera. De gavs även en prioritet på skalan 1 – 3 där prioritet 1 innebar att kravet skulle vara uppfyllt vid den första delleveransen i projektet, vilken inom projektet benämns som BP5a, prioritet 2 innebar uppfyllt vid projektets slutliga leverans, benämnd BP5b, och prioritet 3 vilket var en kategori för sådant som skulle implementeras i mår av tid – det vill säga krav som ej var nödvändiga för godkänd leverans [3]. Kravspecifikationen listar även ekonomiska resurser, vilka för detta projekt endast består av kravet att arbetet ska ta 230 timmar per person att slutföra.

2.2 Utmaningar

Att formulera krav som var enkla att testa och verifiera utifrån det tildelede projektdirektivet visade sig vara svårare än förväntat. Det var lätt att formulera krav på sätt som vid ett tillfälle tycktes självklart, för att någon dag senare läsa om samma krav och få en helt annan bild av vad det innebar. Därväk kändes kommunikationen med beställaren viktig då denna gav en annan syn på hur kraven kunde uppfattas, så missförstånd kunde undvikas. För projektmedlemmarnas egen skull var det också tacksamt att i konstruktionsfasen ha tydligt uppradade och relativt enkla krav att testa.

Prioriteringen av krav var något som var hjälpsamt i början av projektet för att ha något att gå efter då tidplanering skulle göras, men som senare visade sig vara svårt att uppskatta rätt då vissa saker visade sig mer eller mindre viktiga att implementera före eller efter annat. Ett exempel på detta är LCD-displayen som gavs prioritet 3, men visade sig

vara till stor hjälp vid felsökning av sensorer och därfor implementerades redan innan BP5a. Så en utmaning och även en lärdom i detta fall blev att ha koll på och kanske vara beredda på att omförhandla kravens prioritet under projektets gång.

3 KUNSKAPSBAS

För att genomföra projektet har gruppen inledningsvis hämtat information från de förstudieuppgifter som gjordes i början av projektet, se Appendix G. Genom dessa gavs möjlighet att, för respektive modul i konstruktionen, formulera konkreta frågeställningar att göra djupare efterforskning kring. Resultaten från förstudierna gav sedan en god grund att stå på inför den kommande konstruktionsfasen, då mycket grundläggande kunskap hade inhämtats och många konstruktions- och implementationsbeslut därmed redan tagits i och med att frågeställningarna besvarats.

Gruppen utgick även från givet projektdirektiv, se Appendix A, där det beskrivs mer ingående hur roboten ska vara konstruerad för att möta beställarens önskemål, för att strukturera upp hur arbetet skulle fördelas samt utföras.

Majoriteten av all nödvändig komponentinformation som krävts för att implementera roboten har sökts fram från hemsidan Vanheden [4], distribuerad av Institutionen för systemteknik vid Linköpings universitet. På denna sida listas datablad till komponenter samt annan användningsbar information som är behjälplig i diverse datorteknikkursar som ges vid universitetet. Från denna sida hämtades alltså exempelvis datablad för processorerna, sensorerna och chassit samt instruktioner om hur dessa komponenter skulle kopplas in.

Tabell 1: Tabell över använd litteratur, datablad, dokumentation med mera

Källa	Format	Referens
Förstudie - Sensoruppgift	Förstudie	G.1
Förstudie inom reglering	Förstudie	G.2
Förstudie i Busskommunikation	Förstudie	G.3
Projektdirektiv	Direktiv	A
IR-sensor	Datablad	[5]
Tryckknapp	Datablad	[6]
Spänningsregulatorer	Datablad	[7]
Nivåskiftare	Datablad	[8]
Raspberry Pi	Datablad	[9]
Microprocessor ATmega1284P	Datablad	[10]
Reflexsensormodul	Datablad	[11]
Gyroskop	Datablad	[12]
Lidar Lite V3	Användarmanual och teknisk specifikation	[13]
Chassi	Teknisk specifikation	[14]
LCD-display	Specifikation	[15]
Extern klocka	Produktinformation	[16]
Kurshemsida för kursen TSEA56	Hemsida	[17]
Föreläsningsserie i kursen TSEA56	Föreläsning	[18]

I kursens undervisning har utöver själva projektarbetet även föreläsningar, laborationer och seminarium ingått [18]. Det förstnämnda gav en bra introduktion till bland annat lämplig hårdvara samt kommunikation mellan processorer vilken kunde användas till förstudierna, och laborationerna gav en första inblick i hur mjukvaruprogram som användes fungerade samt hur felsökning kunde utföras. Vidare har kursens hemsida [17] varit till stor hjälp för att exempelvis strukturera upp projektets arbetsgång med mallar och annan administrativ information.

Tabell 1 listar de ovan nämnda samt resterande kunskapskällor för projektet. Utöver dessa är det även värt att nämna diverse internetforum som en stor hjälp i arbetet med exempelvis felsökning och implementation i de fall då information från manualer och dylikt varit bristfällig eller helt saknats.

4 GENOMFÖRANDE

Projektet har genomförts enligt LIPS-modellen vilken är en modell för projektarbete som används speciellt inom undervisning. Modellen består av en före-fas, en under-fas, en efter-fas samt ett flertal milstolpar och beslutspunkter. Milstolparna används regelbundet internt inom projektgruppen och beslutspunkter sker i kontakt med beställare eller handledare. Dessutom finns ett antal dokumentationsmallar att tillgå samt mer ingående instruktioner om projektets olika faser och dokument.

I före-fasen deltog projektgruppen till att börja med på ett antal förberedande föreläsningar och laborationer. Därefter analyserades uppdragsbeskrivningen mer ingående och utefter denna skapades en kravspecifikation i samråd med beställaren, och vidare identifierades vad som i praktiken skulle utvecklas framöver. Sedan planerades arbetets tillvägagångssätt genom en projektplan, tidplan och systemskiss. I projektplanen och tidplanen bestämdes aktiviteter, samt när dessa skulle genomföras. De tre planeringsdokumenten var viktiga moment då gruppen tvingades tänka igenom det framtida arbetet noggrant och därmed fick en tydligare bild av vad som krävdes arbets- resp. tidsmässigt varje vecka.

Utifrån planeringen påbörjades under-fasen. Där producerades först en designspecifikation som grundades i systemskissen men som var mer utvecklad. I designspecifikationen bestämdes hårdvara, kopplingsscheman, funktion hos mjukvaran samt specifikationer kring kommunikationen mellan alla moduler. Arbetet fortsatte sedan med kodning och utvecklande av respektive modul. Projektgruppen delades då upp i mindre grupper som fokuserade på olika moduler, en grupp för sensormodulen, en för styrmodulen och en för kommunikation- samt datormodulen. Uppdelningen skedde naturligt utefter vilken förstudie respektive projektmedlem hade skrivit. Möten hölls veckovis för att uppdatera resterande projektmedlemmar om arbetet med vardera modul. Inför leverans jobbade projektgruppen mycket tillsammans med integrationen av modulerna samt systemtester.

Efter projektets genomförande levererades slutprodukten tillsammans med en användarhandledning och arbetet utvärderades. Det levererades också en del sluttillskrift som teknisk dokumentation, kandidatrapport och efterstudie. Slutligen presenterades arbetet genom en framläggning, opposition av en annan gruppars arbete och en tävling mot andra undsättningsrobotar genomfördes.

5 TEKNISK BESKRIVNING

Nedan presenteras projektets förstudier tillsammans med respektive resultat. Vidare ges en övergripande bild av modulernas tekniska resultat, där fokus ligger på utvalda intressanta lösningar; mer ingående detaljerande beskrivningar gällande exempelvis kopplingar på hårdvarunivå eller kod går att finna i projektets tekniska dokumentation. [J](#)

5.1 Förstudier

Förstudien inom reglering undersökte bland annat hur en referenssignal kan genereras för att kunna styra robotten i labyrinten, se Appendix [G.2](#). I studien undersöktes två möjliga sätt att generera en lämplig referenssignal, nämligen pure pursuit och point follow control. Båda dessa har liknande nackdelar, bland annat att styrvinkelns måste ändras diskret för en perfekt följdning av referensignalen, vilket inte är möjligt utan att först stanna robotten. I slutändan användes ingen av dessa då det visade sig vara tillräckligt att direkt använda mitten av vägen som referenssignal. Förstudien undersökte även hur regleringen ska fungera för att kunna följa en referenssignal. PD-reglering togs upp som alternativ och det var i slutändan PD-reglering och öppen styrning som användes i projektet för regleringen av robotten.

Kommunikationsförstudien var inriktad på att jämföra olika busstandarer för överföring av data, se Appendix [G.3](#). Detta var givande inför valet av bussar som skulle användas för överföring av data. Frågeställningarna var att jämföra olika kommunikationsprotokoll för att veta vilka som passade projektet bäst. Detta ledde till valet av SPI som busstandard mellan kommunikationsmodulen och styrmodulen.

I sensormodulens förstudie var syftet att undersöka och få en förståelse för hur sensorer kan användas för att ta fram data som reglerbeslut kan genereras från. Olika sensorer undersöktes genom att leta fakta från datablad och internetsidor, och dessa sammaställdes sedan och jämfördes. Tyvärr kunde inga fysiska undersökningar genomföras på grund av restriktioner gällande att vistas i universitets lokaler. Resultaten som presenterades var att IR-sensorer tycktes bra för mätningar på kortare avstånd, så att fyra stycken placerade pekande ut mot väggarna i linje med fram- resp. bakhjulen skulle täcka vårt behov av mätningar i sidled. Även att en lidar skulle fungera bra för längre mätningar i framåtriktningen samt att ett gyro skulle täcka behovet av att beräkna styrningsvinkel. Gällande beräkning av tillryggalagd sträcka listades lidarn som det troligen bästa alternativet, men detta kom sedan att ändras under konstruktionen då precisionen för denna inte alls gick att lita på för mätningar under en meter. En hjulsensor ersatte i slutändan detta behov med enkel implementering.

5.2 Sensormodul

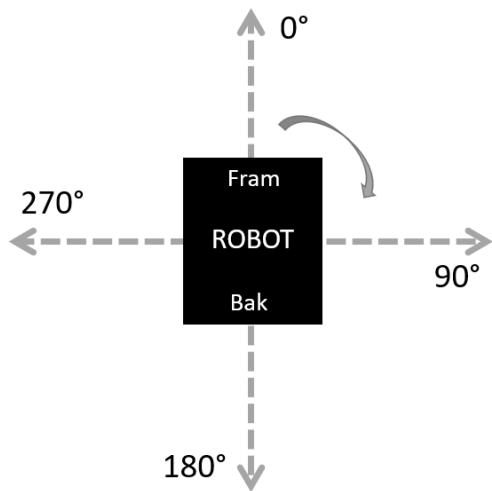
I sensormodulen skiljde sig svårighetsgraden gällande implementering en hel del mellan respektive sensor. Att lösa implementeringen av sensorerna var en intressant uppgift i sig, och som förväntat krävde vissa sensorer mer tid än andra. Detta berodde till stor del på hur mycket information som fanns att tillgå i respektive datablad, samt hur likt tillvägagångssättet var jämfört med de tidigare sensorerna.

Något som dels tog lång tid att implementera men som även var mycket lärorikt var gyroskopet, som är en sensor som används för att mäta vinkelförändringar. Problem som stöttes på med denna sensor ganska omgående var hur rådatan skulle bearbetas. Det första problemet var att rådatan i form av vinkelhastighet behövde konverteras mellan olika datatyper för att kunna bearbetas, och det andra hur den skulle integreras upp för att ge vinkeln i grader.

Konverteringen mellan datatyper gick att lösa relativt smidigt efter införanden av nya variabler så att de önskade flyttalsoperationerna gick att utföra, men att integrera och få ut vinkeln i grader var desto svårare. Lösningen blev att använda en av processorns inbyggda klockor för att mäta hur lång tid vår kod tar mellan varje gyromätning och sedan multiplicera den uppmätta vinkelhastigheten med denna. Dock visade det sig att denna vinkelhastighet ofta antog så små värden att räkneoperationerna resulterade i noll på grund av precisionsfel. För att lösa detta las mer area till under ”vinkelaccelerationsstapeln” innan den multiplicerades med tiden vilket gav ett bättre resultat.

Vidare uppstod utmaningen komma fram till hur denna vinkel skulle bearbetas för att fås i enheten grader. Detta framgick inte tydligt i databladet utan det var diskussioner med bland annat handledare som ledde fram till att vinkelsumman skulle multipliceras med gyrots så kallad sensitivity samt med processorns klockfrekvens dividerat med en skalningsfaktor 1024.

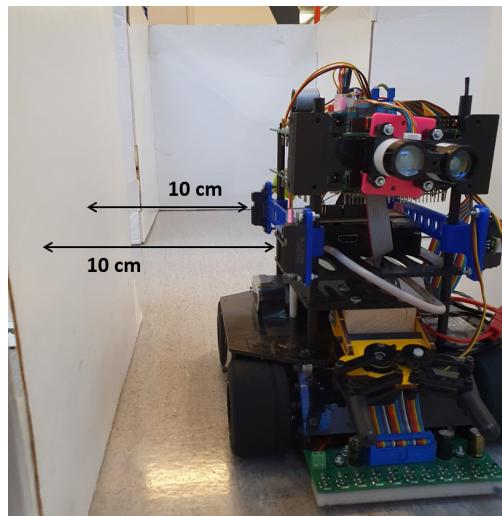
Gyrots konstruktion innebar att vinkelförändringen hela tiden mäts från tidigare position där tidigare position oavsett robotens placering alltid är 0° och den totala vinkelsumman behövde därmed bearbetas ytterligare. Robotens startposition bestämdes att alltid vara 0° och all vinkelförändring beräknades sedan relativt denna startposition, se Figur 4.



Figur 4: En illustration av roboten sedd ovanifrån samt hur vinkeln förändras.

En annan intressant lösning uppkom då vi märkte att gyrots precision inte var perfekt, utan att ett mindre men ändå betydande fel adderades till vinkeln efter svängar. Lösningen var att använda de fyra IR-sensorerna placerade i linje

med robotens hjul för att korrigera felet genom att se när höger respektive vänster sidas par IR-sensorer visar samma avstånd mot väggarna, se Figur 5. Detta indikerar att robotten står rakt i en korridor, och därmed kan gyrots eventuella fel på \pm några grader enkelt korrigeras då vi alltid vet att den totala vinkelsumman oavsett position i banan är multiplar av 90° .



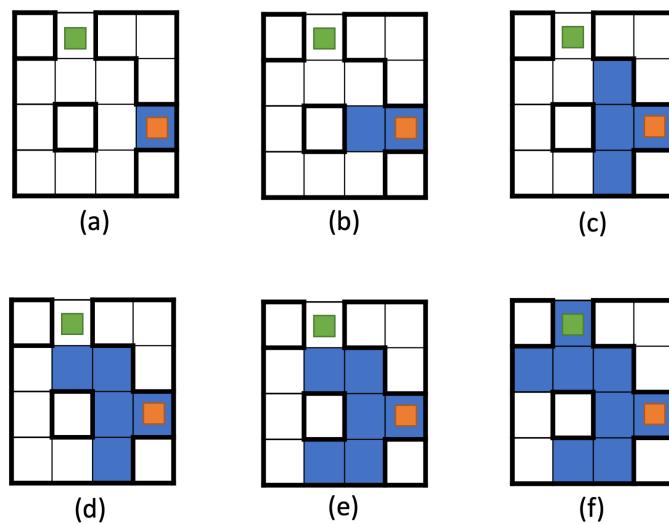
Figur 5: Robotten placerad rakt i en korridor efter sväng.

5.3 Styrmodul

En intressant lösning i projektet var att implementera hur den kortaste vägen mellan två punkter skulle tas fram utifrån en karta. Det finns flera kända algoritmer som löser problemet, men på grund av minnesrestriktionerna i mikroprocessorn som användes behövde en egen lösning skapas.

Lösningen till att ta fram kortaste vägen mellan en startmodul och en destinationsmodul blev att utgå från startmodulen och genom en loop lägga till alla tillgängliga moduler runt startmodulen på en kö. Sedan bearbetas alla dessa moduler genom att lägga till deras tillgängliga grannar som inte redan blivit bearbetade på kön, och så vidare. Till slut kommer den bearbetade modulen att vara destinationsmodulen, varvid påläggningen av moduler kan avslutas. Då finns alltså en lång kö med moduler som börjar på startmodulen och slutar på destinationsmodulen. Figur 6 visar hur denna påläggning av modulerna på kön kan se ut i ett exempel.

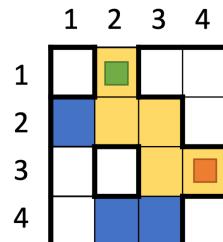
För att ta fram den kortaste vägen utifrån kön utnyttjas att det på kön finns sparat vilken modul som ledde till bearbetningen av nästa modul. Nu kan sista modulen på kön, som alltså är destinationsmodulen, undersökas och från den går det att lista ut vilken modul som ledde till att destinationsmodulen lades på. På detta sätt kan en loop lösa upp hela kön ända tills den når startmodulen, och då är alltså en väg från startmodulen till destinationsmodulen framtagen. Se Tabell 2 för ett exempel på hur kön som skapades i 6 skulle se ut. Se även Figur 7, där den kortaste vägen har hämtats utifrån kön. Eftersom alla moduler undersöks i en cirkel runt startmodulen kommer den kortaste vägen vara funnen då destinationsmodulen läggs till på kön första gången.



Figur 6: För att ta fram kortaste vägen mellan två moduler i labyrinten undersöks modulerna från startmodulen och bortåt, tills destinationsmodulen har hittats. De blåa rutorna är moduler som har lagts på kön, den orangea rutan markerar startmodulen och den gröna rutan markerar destinationsmodulen.

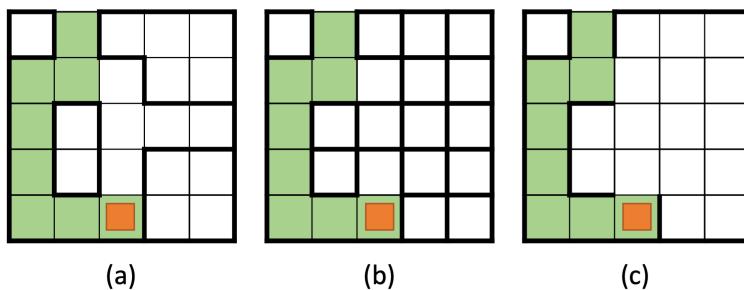
Tabell 2: Kön för exemplet i Figur 6

Position i kön	1	2	3	4	5	6	7	8
Koordinater för modulen	(3,4)	(3,3)	(2,3)	(4,3)	(2,2)	(4,2)	(1,2)	(2,1)
Tidigare moduls position i kön	-	1	2	2	3	4	5	5



Figur 7: Den kortaste vägen från en punkt till en annan, här representerad av den gula vägen. De blåa rutorna är moduler som har lagts på kön men som inte ingår i den kortaste vägen, den orangea rutan markerar startmodulen och den gröna rutan markerar destinationsmodulen. Kön representerad i Tabell 2 har använts för att ta fram modulerna som leder från startmodulen till destinationsmodulen.

Ett annat problem för styrmodulen som också hängde ihop med kartläggningen var hur roboten skulle veta när den kunde sluta kartlägga labyrinten. Enligt kraven skulle roboten sluta kartlägga när den hade hittat den kortaste vägen mellan den nödställda och ingången, men att veta när den gjort det var inte helt lätt.



Figur 8: Exempel på ej helt utforskad labyrint med de tillhörande två kartorna som används för att bestämma när den kortaste vägen mellan den nödställda och ingången har hittats. I alla tre bilderna representerar den gröna vägen modulerna roboten har besökt. Bild (a) visar den riktiga labyrinthen, bild (b) visar ”normal” kartan, och bild (c) visar ”no walls” kartan. I det här fallet är alltså inte den kortaste vägen mellan ingången och den markerade rutan än funnen.

Lösningen blev att använda två kartor som initieras på olika vis. Den ena kartan, ”normal”, antar att det är väggar överallt i labyrinthen, medan den andra, ”no walls” antar att hela labyrinthen är öppen. Båda kartorna uppdateras under kartläggningen och när en vägg påträffas läggs den till i ”no walls” kartan, och när en öppen modul påträffas tas motsvarande vägg bort från ”normal” kartan. Figur 8 visar ett exempel på en utforskad labyrint med robotens båda kartor utritade. Roboten använder bara ”normal” kartan för att navigera, eftersom den endast låter roboten köra där det är säkert att det inte finns någon vägg.

För att veta när den kortaste vägen mellan den nödställda och ingången är funnen jämförs den kortaste vägen mellan de två punkterna enligt ”normal” kartan och ”no walls” kartan. Den kortaste vägen enligt ”no walls” kartan representerar den kortaste möjliga vägen som skulle kunna finnas mellan två punkter. Om den kortaste vägen utifrån ”normal” kartan och ”no walls” kartan är lika långa måste alltså den kortaste vägen mellan de två punkterna vara funnen, och kartläggningen kan avslutas.

5.4 Kommunikationsmodul

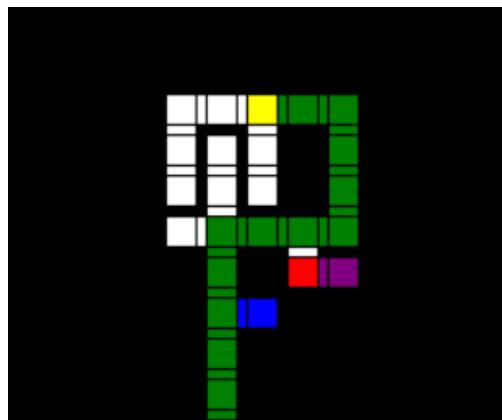
SPI-överföringen för styrmodulen är intressant då det är kommunikationsmodulen som styr överföringarna trots att det endast är styrmodulen som vet hur mycket information som ska överföras vid varje överföring. Till en början var tanken att varje överföring skulle ske med en fast storlek och detta implementerades i kommunikationsmodulen. Det blev dock problem då kartan som var 50 gånger större än all annan data sammanlagt skulle överföras. Problemet med att ha en fast storlek på överföringarna var att det inte blev någon flexibilitet, alla överföringar måste vara så stora att all data kan överföras. Detta ledde till mycket bortslösad tid från båda modulers håll, främst då inte hela kartan uppdateras varje överföring.

Lösningen till detta var att från kommunikationsmodulens håll göra två överföringar, en liten med fast storlek där storleken på nästa överföring skickas och en annan med denna storlek. Detta löste problemet med bortslösade överföringar och skapade mer flexibilitet för styrmodulen att själva välja storleken vid varje överföring.

5.5 Datormodul

En tekniskt intressant lösning som skapades för datormodulen var valet att ha två separata program, ett för det grafiska gränssnittet och ett för den trådlösa kommunikationen. Då kommunikationen till roboten var det första problemet som löstes så fanns redan ett fungerande program för detta vilket alltid lyssnade efter en överföringsbegäran från roboten. Ett andra program för det grafiska gränssnittet behövdes även för om kommunikationsprogrammet skulle användas för det grafiska gränssnittet också riskerar att en överföring från roboten blir förbisedd i då programmet är upptaget med att rita grafiska element.

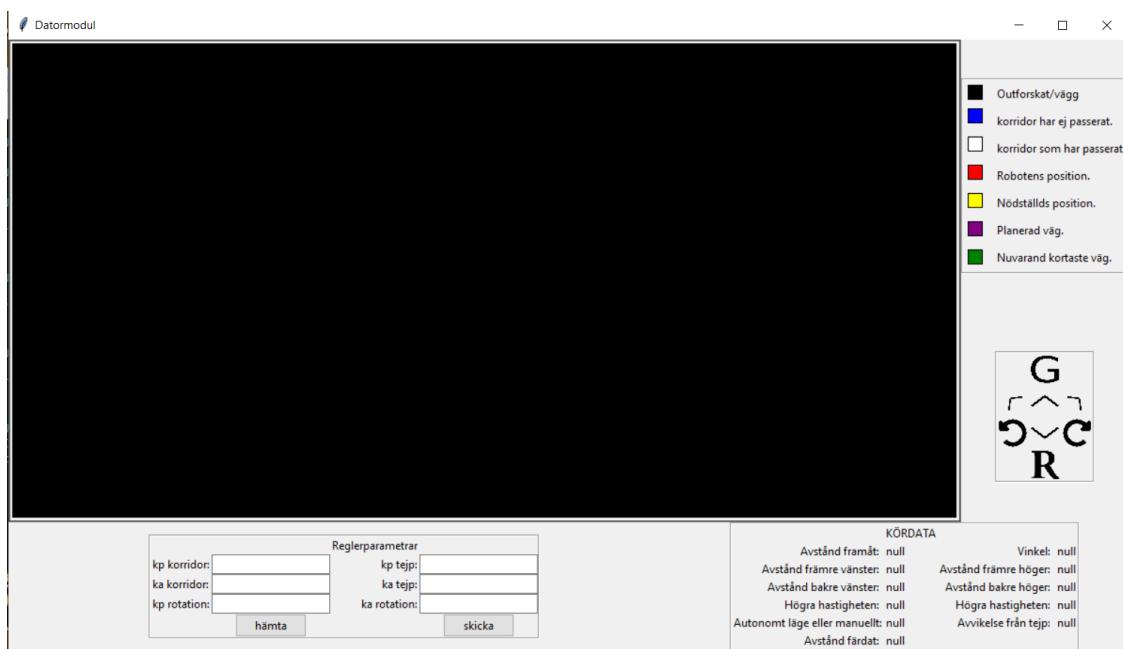
Ett av kraven på datormodulen var att all data som erhölls från roboten skulle sparas. Lösningen blev att kommunikationsprogrammet sparar all data som erhålls från roboten på filer, varefter programmet för gränssnittet läser filerna för att presentera informationen på skärmen för en användare, ett exempel på presentationen är kartan i Figur 9.



Figur 9: En utforskad karta visualiseras av användargränssnittet.

Kommandon från användaren via det grafiska programmet, se Figur 10, till kommunikationsprogrammet förmedlas på motsvarande sätt genom att dessa kommandon sparas på i fil, vilken kommunikationsprogrammet sedan läser och vidarebefordrar till roboten.

En annan intressant lösning var framställningen av kartan för användaren. Den uppdateras i realtid i samband med att roboten kör och skickar kontinuerligt ny kartdata, information om styrbeslut, planerad kortaste väg, position för robot samt position för nödställd. Kartan är uppbyggd av ett rutmönster av rektanglar och kvadrater som representerar väggar respektive moduler i den fysiska labrynten. Varje fyrkant placeras på skärmen genom att specificera dess koordinater i pixlar, vilket gör det svårare att skapa rutmönstret av olika stora fyrkanter som ligger intill varandra.



Figur 10: Den visuella delen av datormodulen.

5.6 Integrering av moduler

Integreringen av modulerna gjordes utifrån de systemgränser som bestämdes i designspecifikationen. Justeringar av systemgränserna var däremot nödvändiga i samband med att nya insikter framkom. De slutliga systemgränserna står specificerade i den tekniska dokumentationen.

Ett problem vid integreringen av moduler var att sensormodulen och styrmodulen utgick från att vinkeln gick åt olika håll. Detta löstes fort då problemet upptäcktes, men ledde först till svåridentifierade problem.

Ett annat integreringsproblem var mellan styrmodulen och kommunikationsmodulen, och bestod av antalet transaktioner mellan dessa då det sker ett avbrott i styrmodulen för varje byte som överförs. Avbrottet skedde för ofta och ledde då till att styrmodulen ej hade tillräckligt med tid för att styra roboten, vilken bland annat körde in i labryintens vägar. Därmed togs beslutet att begränsa antalet transaktioner till två transaktioner per sekund.

6 RESULTAT

Roboten har två olika körlägen, manuellt eller autonomt. En kort beskrivning av dessa samt hur roboten är menad att användas ges nedan. För en komplett användarhandledning, se Appendix H. Oavsett om roboten ska användas i manuellt eller autonomt läge så bör den vid körningens start ha ett fulladdat batteri anslutet via den svarta och röda kabeln nära huvudströmbrytaren, vilken är placerad baktill på robotens chassi. Strömbrytaren bör vara frånslagen i samband med detta, vilket innebär att strömbrytaren ska peka i robotens framåtriktning. När batteriet väl är monterat och stadigt placerat på chassis undre plan kan roboten sedan placeras på lämplig startposition inför körning.

I autonomt läge bör roboten placeras vid ingången till en, enligt ban- och tävlingsspecifikationen i Appendix C, tillåten labyrinth. Vidare bör roboten placeras på ett sådant sätt att reflexsensorn i den främre delen av chassit är mitt över toppen på den T-formade svarta markeringen med fronten riktad inåt mot labyrinten. Utöver det som anges i ban- och tävlingsspecifikationen är det även viktigt med tydlig kontrast mellan färgerna på golv och de svarta markeringarna för förnödenhetens position och för nödställda. Idealt bör golvet ha en vit, matt färg. Om roboten ska framföras i manuellt läge kan dess startposition vara godtycklig. Efter att roboten placerats i startposition kan den sedan börja användas genom att slå på strömbrytaren.

För att välja autonomt respektive manuellt läge finns en brytare placerad överst på roboten med markeringarna A respektive M. I autonomt läge startas sedan roboten med en grå startknapp, även den placerad överst, medan roboten i manuellt läge istället manövreras från en dator som är ansluten till robotens lokala accesspunkt. För en mer ingående beskrivning av hur roboten används, se Användarmanualen (Appendix H).

Gällande ställda krav uppfyller projektgruppens konstruerade undsättningsrobot vid rapportens inlämning samtliga krav av prioritet 1. Gruppen räknar dessutom med att uppfylla samtliga krav av prioritet 2 inför slutleveransen. I Figur 11 ses en bild på en körning inför slutleverans.



Figur 11: En pågående körning vid testning inför slutleverans.

7 SLUTSATSER

Projektet kan sammanfattas i tre olika faser, före, under och efter-fasen. I före-fasen gavs gruppen ett projektdirektiv vilket innebar att konstruera en autonom undsättningsrobot. Därtill skrevs en kravspecifikation vilken lade grunden till den funktionalitet roboten kom att innehålla. I under-fasen gjordes tre separata förstudier och den slutgiltiga designen av roboten bestämdes. Därefter påbörjades konstruktionen av roboten vilket inkluderade montering och hopkoppling av hårdvara samt implementering av mjukvara. Slutligen levererades roboten till beställaren. I efter-fasen skrevs en rad dokument och projektet avslutades med återlämning av nycklar.

Arbetsgången i konstruktionsfasen innebar att projektgruppen delades i mindre grupper och utvecklade varsin modul. Till följd av uppdelningen hade projektmedlemmarna i de mindre grupperna goda kunskaper om sin specifika modul men sämre kunskap om resterande moduler. Arbetssättet var tidseffektivt men försvårade vid testning och generell felsökning.

Den goda kommunikation inom projektgruppen förenklade dock genomförandet av projekt och speciellt integreringen mellan de olika modulerna. I slutändan blev alla projektmedlemmar nöjda med resultatet av projektet och alla krav med prioritet ett är vid rapportens inlämning uppfyllda. Dessutom är projektgruppen på god väg att uppfylla alla krav med prioritet två och tre. Mer detaljer om genomförandet och tekniska resultatet återfinns i efterstudien, se Appendix I.

Det framtida arbetet består av att genomföra framläggning och opposition samt att lämna in de dokument som ska levereras till beställaren. Vid en förlust i tävlingen som arrangeras mellan de grupper som har konstruerat varsin undsättningsrobot ska roboten monteras isär och lämnas tillbaka till handledaren. Vid en vinst ska roboten lämnas till beställaren utan att monteras isär. En efterstudie ska genomföras för att utvärdera arbetet som gruppen har gjort under projektet.

Det vi är mest nöjda med i projektet är att vi har haft en bra kommunikation sinsemellan sedan kursen startade. Vi gjorde ett tydligt gruppkontrakt och diskuterade hur vi skulle ta beslut och lägga upp en tydlig plan för projektet, och detta gav en bra grund att stå på för att genomföra projektet.

Det vi skulle göra annorlunda om vi skulle göra om samma uppdrag är att göra om tidsplanen med aktiviteter. Vi har inte gjort ett liknande projekt tidigare i samma storlek så var det en utmaning att veta vad vi behövde lägga tid på och vilka aktiviteter vi behövde. Vi skulle därför gjort aktiviteterna annorlunda för att få en tidsplan mer lik det vi faktiskt gjorde under projektets gång. Vidare hade vardera modul kunnat utvecklas om mer tid fanns.

7.1 Sensormodulen

Sensormodulen skulle kunna utvecklas med bättre och noggrannare kalibrering av sensorerna. Däremot, trots perfekt kalibrering av sensorer kvarstår en del precisionsfel från hårdvaran. Detta kan åtgärdas med fler och bättre sensorer med högre precision vilket skulle resultera i en bättre och mer exakt reglering. Bättre sensorer medför dock en större inköpskostnad, så de sensorer vi använde fungerade ändå väl för ändamålet.

En ändring som skulle göra uppgiften mer intressant sett från sensormodulens sida är att implementera en roterande lidar. En roterande lidar ger mycket information från omgivningen, vilket kan bidra med mycket mer data att reglera

utifrån för att få högre precision. Den roterande lidarn är mer anpassningsbar för olika omgivningar vilket gör den mer lämplig att använda i verkligheten.

7.2 Styrmodulen

För att förbättra styrmodulen finns det två algoritmer som skulle kunna förbättras, utöver regleringen i allmänhet. Regleringen just nu är inte optimal. Med mer tid skulle en bättre implementation kunna utföras, och reglerparametrarna skulle kunna förbättras en del.

Den ena algoritmen som kan förbättras är vad som sker om den nödställd är funnen men den kortaste vägen mellan den nödställd och utgången inte är det. Som det är nu så kommer roboten bara fortsätta kartlägga med sin vanliga algoritm tills den hittat kortaste vägen, men detta skulle kunna förbättras. Den skulle till exempel kunna utforska delarna nära utgången eller den nödställd först, eller till och med lista ut vilken potentiell väg som skulle kunna vara snabbare än den kortaste den funnit och se om denna potentiella väg går att köra eller inte.

Den andra algoritmen som kan förbättras är den som planerar körvägen mellan ingången och den nödställd. Nu prioriterar den svängar och raksträckor likadant, men en mer effektiv metod hade varit att prioritera raksträckor eftersom de tar kortare tid att köra än svängar.

I styrmodulen så lades mycket tid på att förbättra regleringen, vilket behövdes främst då labyrinthens korridorer tilläts variera ganska mycket i storlek och det var svårt att bygga en bra labyrinth där alla svängar var exakt 90 grader. Uppgiften hade varit mer intressant om mer tid hade varit tillgänglig för att förbättra kartläggningen av labyrinthen eller förbättra de olika algoritmerna som används för att till exempel hitta den kortaste vägen eller för att veta när kartläggningen är klar.

7.3 Kommunikations- och datormodulen

En sak som kommunikationsgruppen skulle vilja lägga mer tid på är att öka antalet kompletta överföringar per sekund mellan kommunikationsmodulen och styrmodulen. Om detta kunde implementeras skulle uppdateringshastigheterna i användargränssnittet att öka, så att informationen på skärmen uppdateras oftare. Det går också att förbättra programmet som körs på kommunikationsmodulen, så att det automatiskt letar efter datormodulen i adressintervallet för accesspunkten. Adressen behöver då aldrig ändras manuellt.

En ändring i projektdirektivet som skulle göra projektet mer intressant skulle kunna vara att kräva att information i datormodulen uppdateras med en viss frekvens, så att realtidsuppdateringen av exempelvis kartan blir jämnare. Detta skulle leda till en lösning som är mer intressant i hur de löser problemet med för många avbrott i styrmodulen.

REFERENSER

- [1] M. Krysander, “TSEA56: Projektdirektiv för en undsättningsrobot,” http://www.isy.liu.se/edu/kurs/TSEA56/projektdirektiv/Projektdirektiv_for_undsattningsrobot_21.pdf, [Online; hämtad april 30, 2021].
- [2] F. Johansson, O. Pihl, M. Ehn, D. Ekblad, M. Jämte, A. Thunberg, E. Sevonius, G. Arctaedius, and L. Byman, “Ban- och tävlingsspecifikation,” februari 2021, version 1.0.
- [3] A. Nicklasson Cedbro, E. Leuchovius, F. Johansson, G. Arctaedius, J. Carlson, and L. Byman, “Kravspecifikation,” februari 2021, version 1.0.
- [4] “Välkommen till vanheden!” <http://www.da.isy.liu.se/vanheden/>, [Online; hämtad april 30, 2021].
- [5] “GP2D120,” <http://www.da.isy.liu.se/vanheden/pdf/gp2d120.pdf>, [Online; hämtad april 9, 2021].
- [6] “Så här fungerar tryckknappen,” <http://www.da.isy.liu.se/vanheden/image/tryckknapp.png>, [Online; hämtad mars 1, 2021].
- [7] “Low drop fixed and adjustable positive voltage regulators,” <http://www.da.isy.liu.se/vanheden/pdf/ld1117v33.pdf>, [Online; hämtad mars 1, 2021].
- [8] “Nivåskiftare Raspberry ↔ AVR datasheet,” <http://www.da.isy.liu.se/vanheden/pdf/da-nivaskiftare.pdf>, [Online; hämtad februari 26, 2021].
- [9] “Raspberry Pi Compute Module 3+ datasheet,” https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf, 2019, [Online; hämtad februari 26, 2021].
- [10] “8-bit Microcontroller with 128K Bytes In-System Programmable Flash,” <http://www.da.isy.liu.se/vanheden/pdf/atmega1284p.pdf>, [Online; hämtad februari 25, 2021].
- [11] “Reflexsensormodulen,” <http://www.da.isy.liu.se/vanheden/pdf/reflexsensormodul.pdf>, [Online; hämtad februari 25, 2021].
- [12] “MEMS motion sensor: three-axis digital output gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/L3GD20.pdf>, [Online; hämtad februari 25, 2021].
- [13] “Lidar Lite v3 Operation Manual and Technical Specifications,” <http://www.da.isy.liu.se/vanheden/pdf/lidarlite3OMTS.pdf>, [Online; hämtad februari 25, 2021].
- [14] “Fyrhjulig robot – Terminator,” http://www.da.isy.liu.se/vanheden/pdf/terminator_prel.pdf, [Online; hämtad maj 14, 2021].
- [15] “Character Type Dot Matrix LCD Module JM162A,” <http://www.da.isy.liu.se/vanheden/pdf/jm162a.pdf>, [Online; hämtad maj 14, 2021].
- [16] “IQEXO-3,” <http://www.da.isy.liu.se/vanheden/pdf/exo3.pdf>, [Online; hämtad maj 14, 2021].

- [17] M. Kryssander, “TSEA56 - Kandidatprojekt i elektronik,” <http://www.isy.liu.se/edu/kurs/TSEA56/>, [Online; hämtad april 30, 2021].
- [18] ——, “Föreläsningar,” <http://www.isy.liu.se/edu/kurs/TSEA56/forelasning/>, [Online; hämtad april 30, 2021].

Appendix

A PROJEKTDIREKTIV

TSEA56: Projektdirektiv för en undsättningsrobot

Version: 1.0
Beställare: Mattias Krysander
Datum: 2021-01-15

Inledning

Vi vill producera en robot som effektivt kan förse nödställda i en byggnad med mat och andra förnödenheter. För att kunna göra detta måste roboten kunna utforska en okänd byggnad och bestämma kortaste vägen till de nödställda. Sedan ska den så snabbt som möjligt köra fram och tillbaks med förnödenheter till de nödställda. För att utvärdera hur man kan göra en sådan robot så önskar vi beställa ett antal prototyper av undsättningsrobotar. Dessa ska delta i en tävling där vi kan utvärdera olika konstruktionsalternativ. För att erhålla olika konstruktioner ombeds ni att göra en kravspecifikation i dialog med beställaren. Nedan ger vi er ett antal grundkrav som ska vara gemensamma i alla kravspecifikationer. Kraven är inte numrerade vilket de dock ska vara i era kravspecifikationer. Ni förväntas lägga till mer text som beskriver roboten i allmänna termer, figurer samt unika krav för just er robot. Kom ihåg att krav kan prioriteras.

Uppdraget

Uppdraget består av två körningar i en byggnad. I den första körningen ska roboten, från en definierad startposition, utforska byggnaden och bestämma den kortaste vägen till de nödställda. I den andra körningen ska roboten köra så fort som möjligt från samma startposition till de nödställda, lämna ”fornödenheter”, och sedan återvända till startpunkten och stanna där. Byggnaden består att ett korridorsystem och är uppbyggd av kartongväggar med moduler om 40 cm och rymmer inom ett område på maximalt 10*10 meter. Väggarnas position ska kunna avvika från rutnätet med ± 4 cm. Det kan finnas 4-vägskorsningar och återvändsgångar. Positionen för de nödställda är markerad på ett speciellt sätt. Det ska finnas flera vägar till de nödställda. Detaljer i bantekniken kan bestämmas mellan de deltagande grupperna och beställaren.

Systemarkitektur

Systemet ska bestå av en robot samt mjukvara till dator för styrning och övervakning. För att senare kunna testa alternativa sensorer, fjärrstyrningar och även styralgoritmer, ska roboten vara moduluppbyggd. Gränssnitten mellan modulerna ska vara noggrant specificerade i den tekniska dokumentationen. Man ska enkelt kunna byta ut en modul mot en annan. Varje modul ska innehålla minst en egen processor. Följande tre moduler ska ingå i konstruktionen:

- kommunikationsmodul (med trådlös länk, wifi eller blåtand)
- styrmodul (motorer, eventuell display, styrlogik)
- sensormodul (hit ska alla sensorer vara kopplade)

Fjärrstyrning och övervakning

Roboten ska kunna fjärrstyras via trådlös länk. Följande kommandon ska roboten reagera på: Fram, fram vänster, fram höger, back, stopp, rotera vänster och rotera höger. Under körning ska roboten fortlöpande skicka mätdata (avstånd till väggar, avlägd sträcka, vridning etc.), styrbeslut (sväng vänster, kör fram, plocka upp förnödenhet, etc) och styrdata (motorernas gaspådrag) till en bärbar PC. Dessa data ska skickas via en trådlös länk och presenteras på datorns skärm på ett användarvänligt sätt. Överfört data ska också kunna sparas på fil i ett format som tillåter plottering av valda signaler för en körning i t ex Matlab eller Python. Under körning ska den utforskade delen av byggnaden visas som en karta där robotens startposition, nuvarande position, nödställd och kortaste vägen mellan startpunkt och nödställd markeras.

Diverse övriga funktionskrav

Det ska finnas en brytare på roboten med vilken man väljer fjärrstyrningsläge eller autonomt läge och en knapp med vilken man startar den i tävlingen. Markeringen av de nödställda ska göras med en svart ruta på golvet (alternativa markeringar kan föreslås). Det ska finnas någon form av styralgoritm (till exempel PD-reglering), så att roboten kan köra i en korridor utan att ”slingra” sig fram (verifieras genom dokumenterade testkörningar på raksträckor). Parametrar till robotens styralgoritm ska kunna initieras via den trådlösa länken. Den första körningen är tidsbegränsad. Maximal tid sätts i samråd med beställaren. Det skulle kunna vara bra att ha en LCD- display på roboten som visar värden från valda sensorer. Roboten ska ha någon form av gripklo med vilken den kan greppa ett föremål innan start och sedan släppa det i målutan. Det vore trevligt om roboten autonomt kan greppa föremålet. Det räcker att utforska så mycket av byggnaden att den kortaste vägen mellan startpunkt och nödställd kan beräknas.

Tävlingsregler:

Vinnare är den robot som klarar att lämna förnödenheter på kortast tid och som kan rita upp en korrekt karta. Repeterbarhet ska dock kunna uppvisas. Reglerna bestäms i samråd med beställaren.

Övriga krav

Projektet ska bedrivas enligt LIPS-modellen och samtliga dokument ska utgå från LIPS-mallar. I förefasen ingår att projektgruppen ska ta fram en kravspecifikation, en systemskiss och en projektplan med tidplan. Samtliga dessa dokument ska godkännas av beställaren. Budget för förefasen finns på beställarens hemsida. Efter godkänd projektplan (BP2) ska projektet ta 230 arbets timmar/person att slutföra. Vid verifiering av baskrav (BP5a) ska roboten kunna köra runt i en bana. Detaljer för baskrav utarbetas i samråd med beställare. Vid slutleveransen (BP5b) ska det finnas en fungerande robot samt teknisk dokumentation med användaranvisning. Projektets delleveranser och slutleverans ska senast ske vid de datum som finns specificerade på beställarens hemsida. Även formen för slutleveransen beskrivs på denna hemsida.

B KRAVSPECIFIKATION

Kravspecifikation

4 februari 2021

Version 1.0



Status

Granskad	ANC, JC	2021-02-04
Godkänd	Kent	2021-02-04

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedi		gunar889@student.liu.se

INNEHÅLL

1	Inledning	1
1.1	Parter	2
1.2	Syfte och mål	2
1.3	Bakgrundsinformation	2
1.4	Definitioner	2
2	Översikt av systemet	2
2.1	Grov beskrivning av produkten	2
2.2	Ingående delsystem	3
2.3	Avgränsningar	3
2.4	Generella krav på systemet	4
3	Kommunikationsmodul	5
3.1	Krav	5
4	Sensormodul	5
4.1	Krav	5
5	Styrmodul	6
5.1	Krav	6
6	Datormodul	6
6.1	Krav	6
7	Ekonomi	7
8	Leveranskrav och delleveranser	7
9	Dokumentation	7
	Referenser	8

DOKUMENTHISTORIK

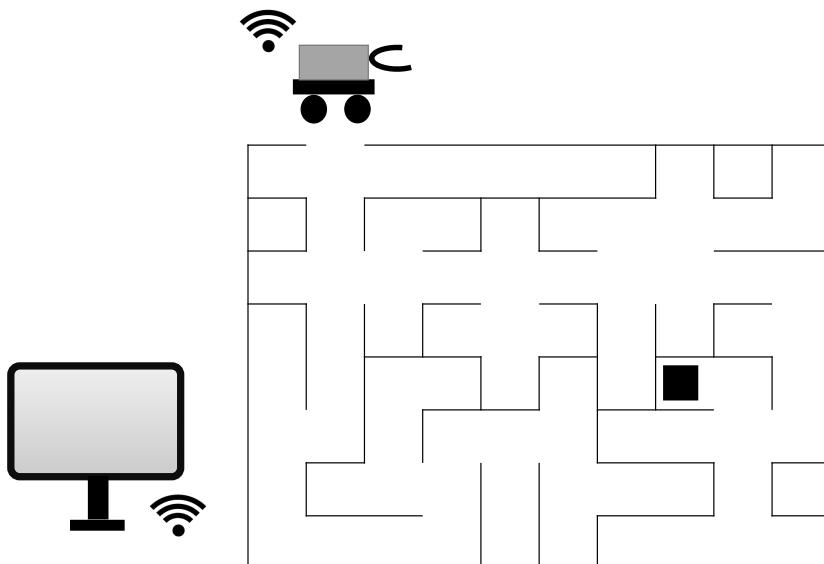
Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2021-01-28	Första utkast	Alla	Alla
0.2	2021-02-02	Andra utkast	Alla	Alla
1.0	2021-02-04	Tredje utkast	ANC, JC	ANC, JC

1 INLEDNING

Detta dokument beskriver de krav som förväntas gälla för en prototyp av en undsättningsrobot konstruerad i kursen TSEA56, kandidatprojekt i elektronik, under vårterminen 2021 vid Linköpings universitet. Kraven specificeras enligt nedan, där kravnummer anges i kolumn 1, kravets version (om det reviderats eller ej) i kolumn 2, beskrivning av kravet ges i kolumn 3, och kravets prioritet anges i kolumn 4. Prioritet 1 innebär att kravet skall vara uppfyllt vid projektets första delleverans av systemet, kallad BP5a. Prioritet 2 innebär att kravet skall vara uppfyllt vid projektets slutliga leverans, kallad BP5b. Prioritet 3 innebär att kravet inte är nödvändigt för slutleverans; dessa utförs i mån av tid.

Krav	Version	Beskrivning	Prioritet
------	---------	-------------	-----------

Figur 1 visar en förenklad bild av roboten i situationen för uppdraget. För specifikationer gällande byggnad/bana, förnödenheter samt nödställda och markering för dessa, se dokument "Ban- och tävlingsspecifikation"[\[1\]](#). Roboten skall utgående från en vald startposition undersöka en okänd byggnad, för att finna den snabbaste vägen till i förväg utplacerade "nödställda". Vidare skall roboten återvända till startpositionen, plocka upp förnödenheter med en fastmonterad gripklo samt leverera dessa till de nödställda på snabbast möjliga tid. För specifikationer gällande uppdraget, se "Ban- och tävlingsspecifikation"[\[1\]](#). Körningen skall kunna utföras både autonomt och manuellt, där styrningen i sistnämnda fallet sköts trådlöst via en dator. Relevant mät- och styrdatalia skall under körningen skickas till dator via trådlös länk.[\[2\]](#).



Figur 1: Systemet i dess omgivning. Roboten står markerad i startposition och den svarta rutan markerar nödställda.

1.1 Parter

Ingående parter för denna kravspecifikation är beställare (kund) Kent Palmkvist, ISY, handledare Peter Johansson, ISY, samt producenter tillika projektets utförare i form av projektgruppen listad i tabell under Projektdeltagare.

1.2 Syfte och mål

Projektets syfte är att ge erfarenhet av arbeten med större projekt inom elektronikkonstruktion samt en förståelse för och kunskaper om metoder för strukturerad konstruktion av stora, komplexa digitala system[3]. Målet med projektet är att konstruera en autonom undsättningsrobot i linje med de krav som listas i detta dokument samt leverera prototypen till beställare, inom ramar för budget.

1.3 Bakgrundsinformation

Beställaren har en önskan om att producera en robot som effektivt kan förse nödställda i en byggnad olika förnödenheter och anordnar tärför en tävling där olika konstruktioner ställs mot varandra. Detta för att kunna utvärdera olika konstruktionsalternativ, samt ge deltagarna i kursen TSEA56 erfarenheter inom projektarbete och elektronikkonstruktion.

1.4 Definitioner

Då projektets uppgift består av att konstruera en prototyp av en undsättningsrobot för en verlig situation används flera analogier till motsvarande verklighet. Byggnad anges återkommande som benämning på den bana där robotens uppdrag kommer att utvärderas. I denna byggnad kommer markeringar finnas vilka motsvarar nödställda i en verlig situation. Roboten ska alltså kunna återvända till markeringen för nödställda med något föremål som symboliseras någon förnödenhet.

2 ÖVERSIKT AV SYSTEMET

I detta avsnitt sker en översiktlig redogörelse över systemet.

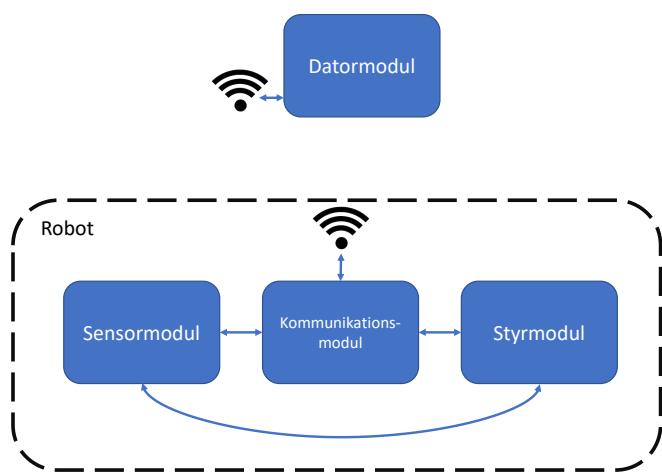
2.1 Grov beskrivning av produkten

Produkten består av en robot som ska kunna köra i ett plant område. Den ska både kunna styras manuellt med styrkommandon och kunna utföra sitt uppdrag autonomt. Mjukvara till en dator ingår för att kunna övervaka och styra roboten, vilket ska ske trådlöst.

2.2 Ingående delsystem

Figur 2 är en principskiss över systemets fyra delsystem. Den slutliga interaktionen mellan delsystemen kan komma att ändras, men utan att påverka slutproduktens funktionalitet.

Roboten kommer att bestå av tre moduler/delsystem: en *sensormodul* som samlar information från robotens omgivning, en *styrmodul* som reglerar robotens rörelser och en *kommunikationsmodul* som är ansvarig för den trådlösa kommunikationen mellan robot och dator. Utöver dessa definieras även ett fjärde delsystem som benämns *datormodul*, vilken utgör ett användargränssnitt till roboten för styrning, övervakning och lagring av data.



Figur 2: En översikt av systemet.

Se avsnitt 3 till 6 för en kort individuell beskrivning och kravsförteckning för respektive delsystem.

2.3 Avgränsningar

Roboten är begränsad till att bara kunna hantera körbanan enligt ban- och tävlingsspecifikationen, se dokument ”Ban- och tävlingsspecifikation”[1] samt ha en körförmåga som stämmer överens med tävlingsreglerna. Roboten ska även klara av uppgiften specificerad i tävlingsspecifikationerna tre av fyra gånger vid tävling och demonstration. Gripklon, vilken är fastmonterad på roboten, är begränsad till att enbart kunna greppa lätta föremål.

2.4 Generella krav på systemet

Krav	Version	Beskrivning	Prioritet
1	Original	Systemet ska bestå av en robot och ett datorsystem i en datormodul.	1
2	Original	Roboten ska vara moduluppbryggd.	1
3	Original	Modulerna ska vara utbytbara/oberoende.	1
4	Original	Varje modul ska innehålla minst en egen processor.	1
5	Original	I systemet ska det finnas en kommunikationsmodul.	1
6	Original	I systemet ska det finnas en styrmodul.	1
7	Original	I systemet ska det finnas en sensormodul.	1
8	Original	Under körning ska roboten fortlöpande ta styrbeslut.	1
9	Original	Datormodulen ska kunna spara sensordata.	1
10	Original	Datormodulen ska kunna styra gripklon i manuellt läge.	2
11	Original	Det ska finnas en brytare på roboten med vilken man väljer fjärrstyrningsläge eller autonomt läge.	1
12	Original	Det ska finnas en knapp på roboten med vilken man startar roboten i tävlingen.	2
13	Original	Kartläggningen ska göras på en begränsad tid, se "Ban- och tävlingsspecifikation" ^[1] .	2
14	Original	Roboten ska ha en LCD-display som visar värden från valda sensorer.	3
15	Original	Roboten ska ha en gripklo.	1
16	Original	Roboten ska ta den optimala vägen från startpunkt till den nödställda.	2
17	Original	Roboten ska endast kartlägga byggnaden tills den kortaste vägen mellan startpunkt och slutförslag har identifierats.	3
18	Original	Roboten ska klara av tävlingsuppgift specificerad i "Ban- och tävlingsspecifikation". ^[1]	1
19	Original	Roboten ska klara av tävlingsuppgift specificerad i "Ban- och tävlingsspecifikation" [1] 3 av 4 gånger.	2
20	Original	Det ska finnas en styrstrategi så att roboten kan köra rakt i en korridor och maximalt avvika 4 cm från mittlinjen.	2
21	Original	Roboten ska kunna fjärrstyras via trådlös länk från datormodulen.	1
22	Original	Roboten ska kunna hitta och greppa föremålet med gripklon autonomt.	3
23	Original	Roboten ska kunna utföra i manuellt läge: fram, fram vänster, fram höger, back, stopp, rotera vänster och rotera höger.	1

3 KOMMUNIKATIONSMODUL

Kommunikationsmodulen ska sköta kommunikationen mellan robotens övriga moduler och PC via trådlös länk.

3.1 Krav

Krav	Version	Beskrivning	Prioritet
24	Original	Roboten ska fortlöpande under körning skicka data till en bärbar PC.	1
25	Original	Parametrar till robotens styralgoritm ska kunna initieras via den trådlösa länken.	2

4 SENZORMODUL

Senzormodulen är den del av roboten som hanterar sensorer.

4.1 Krav

Krav	Version	Beskrivning	Prioritet
26	Original	Under körning ska roboten fortlöpande samla in mätdata från sensorerna.	1
27	Original	Senzormodulen ska kunna mäta avstånd på minst 24 cm.	1
28	Original	Senzormodulen ska kunna avläsa svarta golvmärkningar.	1
29	Original	Senzormodulen ska ha switch/brytare och startknapp anslutet till sig.	1

5 STYRMODUL

Styrmodulen styr alla robotens reglage och sköter därfor styrningen. Styrmodulen styr även en gripklo som sitter längst fram på roboten och som kan plocka upp ett förbestämt föremål.

5.1 Krav

Krav	Version	Beskrivning	Prioritet
30	Original	Under körning ska roboten fortlöpande samla styrdata.	1
31	Original	Styrmodulen ska kunna greppa samt släppa ett föremål med en gripklo.	1
32	Original	Styrmodulen ska kunna styra hjulen.	1

6 DATORMODUL

Datormodulen kommer att vara den delen av systemet som sparar data och är användargränssnittet som används för styrning av roboten. Daten kommer kontinuerligt att skickas till datormodulen via en trådlös länk.

6.1 Krav

Krav	Version	Beskrivning	Prioritet
33	Original	Datormodulen ska under körning rita upp den utforskade delen av byggnaden som en karta där robotens startposition, nuvarande position, nödställd och den kortaste vägen mellan startpunkt och nödställd markeras.	2
34	Original	Datormodulen ska kunna presenteras på den bärbara PC:ns skärm på ett användarvänligt sätt.	2
35	Original	Datormodulen ska kunna spara mätdata på fil.	2

7 EKONOMI

Ekonomiska resurser för projektet.

Krav	Version	Beskrivning	Prioritet
36	Original	Efter godkänd projektplan (BP2) ska projektet ta 230 arbetsstimmar per person att slutföra.	2

8 LEVERANSKRAV OCH DELLEVERANSER

Under projektets gång kommer ett flertal dokument tas fram och uppföljningar av krav kommer att ske enligt givna leveransdatum [4].

9 DOKUMENTATION

Tabell 8 listar de dokument som kommer att produceras.

Tabell 8: Dokument som skall produceras.

Dokument	Språk	Syfte	Format
Kravspecifikation	Svenska	Visa vilka krav som roboten i slutet ska testas mot.	PDF
Ban- och tävlingsspecifikation	Svenska	Specificera gemensamma regler för tävling samt specificera banans utformning.	PDF
Projektplan	Svenska	Ge en plan för när milstolpar ska nås.	PDF
Designspecifikation	Svenska	Robotens design att utgå från under uppbyggnad.	PDF
Reglerförstudie	Svenska	Studera möjliga reglerlösningar.	PDF
Sensorförstudie	Svenska	Studera möjliga sensorlösningar.	PDF
Kommunikation och Konstruktion förstudie	Svenska	Studera möjliga kommunikationslösningar.	PDF
Tidrapport	Svenska	Ge en översikt över varje gruppmedlems nedlagda tid.	xlsx
Teknisk dokumentation	Svenska	Förklara hur roboten är byggd.	PDF
Användarhandledning	Svenska	Förklara hur en användare ska använda systemet.	PDF
Efterstudie	Svenska	Ge en lätt översikt över hur projektet har gått.	PDF
Kappa	Svenska	Ge en komplett överblick av det genomförda arbetet.	PDF

REFERENSER

- [1] . Grupp 1, 2 och 3, “Ban- och tävlingsspecifikation,” 2021.
- [2] M. Krysander, “TSEA56: Projektdirektiv för en undsättningsrobot,” http://www.isy.liu.se/edu/kurs/TSEA56/projektdirektiv/Projektdirektiv_for_undsattningsrobot_21.pdf, [Online; hämtad 25-01-2021].
- [3] ——, “Kursinformation TSEA56,” <http://www.isy.liu.se/edu/kurs/TSEA56/kursinformation/>, [Online; hämtad 25-01-2021].
- [4] ——, “Leveranser i projektet 2021,” <http://www.isy.liu.se/edu/kurs/TSEA56/leveranser.html>, [Online; hämtad 25-01-2021].

C BAN- OCH TÄVLINGSSPECIFIKATION

Ban- och Tävlingsspecifikation

Filip Johansson, Oscar Pihl, Max Ehn, David Ekblad, Marcus Jämte,
Anton Thunberg, Eric Sevonius, Gunnar Arctaedius, Lovisa Byman

4 februari 2021

Version 1.0



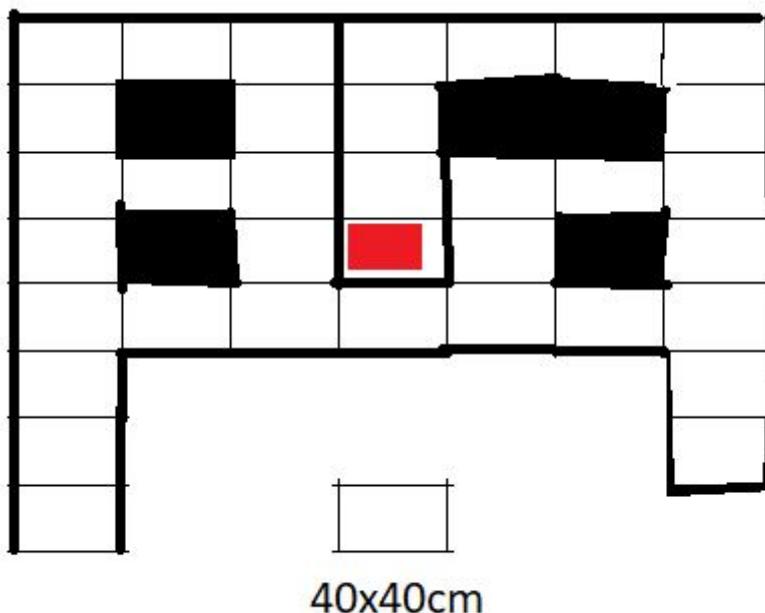
Dokumenthistorik

Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2021-01-29	Första utkastet	David Ekblad, Marcus Jämte, Anton Thunberg, Eric Sevönius, Gunnar Arctaedius, Lovisa Byman	Gunnar Arctaedius
0.2	2021-02-04	Har lagt till rubrik, dokumenthistorik, titel och författare. Krav har reviderats i enlighet med beställarens uppmaningar.	Max Ehn, Filip Johansson, Oscar Pihl	Filip Johansson
1.0	2021-02-05	Förtydligande gällande den rektangulära markeringens orientering i banan.	Filip Johansson	Filip Johansson

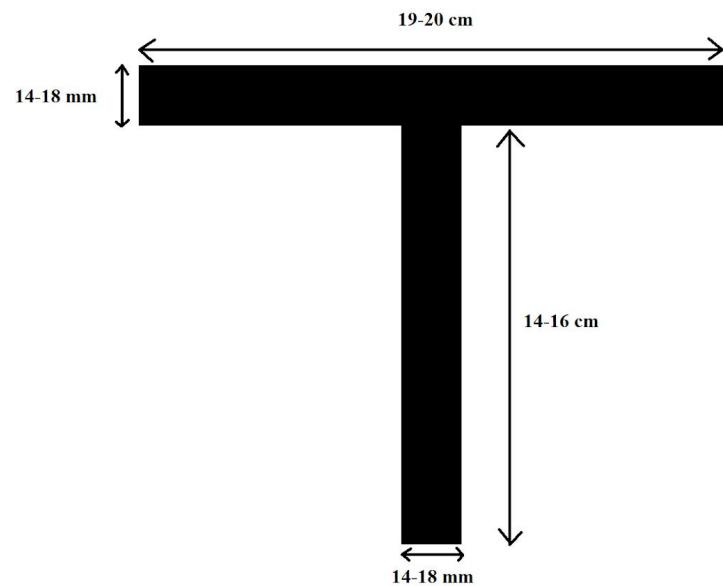
Banregler

Nedan definieras de banregler som kommer att gälla under tävlingen för samtliga involverade projektgrupper.

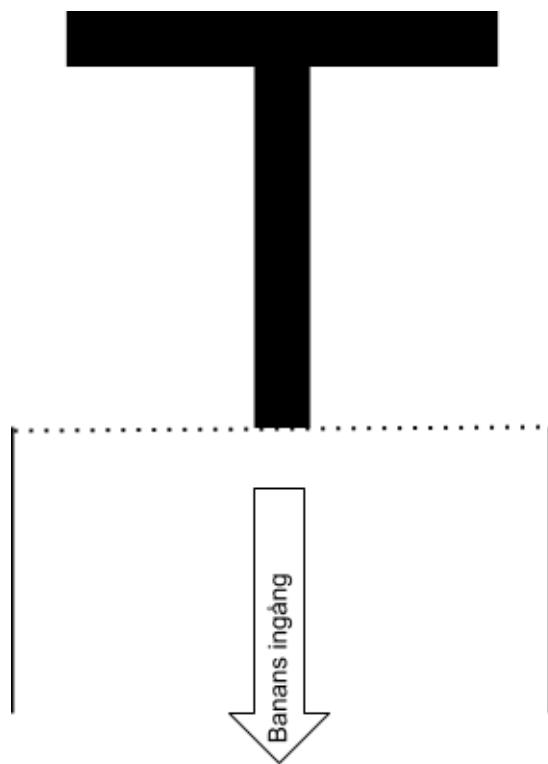
- Det är tillåtet med cykler i banan.
- Det är tillåtet med återvändsgränder.
- Det är tillåtet med fyrvägskorsningar.
- Banan skall bestå av moduler (rutor) om $40 \pm 4 \times 40 \pm 4$ cm och spänna upp ett rutnät av max 25x25 rutor. Se figur 1.
- Alla väggar (förutom dem som definierar ingången till banan) skall sluta i en annan vägg.
- Väggar ska ha vit färg.
- Golvet ska ha vit färg.
- Den nödställda skall markeras med en icke ifylld rektangel av svart tejp (elitejp) 30x20 cm, centrerad i en av banans moduler och med rektangelns kortsidor (20 cm) parallella med korridorens väggar.
- Förmödenhet skall vara markerad med ett "T" av elitejp (14-18mm bredd) med dimensioner (19-20)x(14-16) cm. Se figur 2.
- Förmödenhet ska vara placerad utanför ingången av banan. Orienteringen samt placeringen av "T-markeringen" är sådan att basen av "T-markeringen" är parallell med ingången. Se figur 3.
- Det ska finnas flera vägar till de nödställda.
- Banan ska delas upp i ett flertal korridorer och korsningar, där korridorens bredd är 1 modul.



Figur 1: Exempel på en bana



Figur 2: T-markeringen



Figur 3: T-markeringens orientering relativt banans ingång.

Tävlingsregler

Nedan definieras de krav och regler som påverkar hur de olika robotarna bedöms under tävlingsmomentet. Upplägget av tävlingen i sig beskrivs även.

Allmänna tävlingsregler

Varje grupp ska köra igenom båda faserna 3 gånger. Bästa tid för var fas, enligt tävlingsreglerna för varje sådan nedan, blir gruppens resultat tid i det tävlingsmomentet. Om ett lag inte klarar 2 utav 3 körningar så diskas gruppen. Att inte klara en körning definieras av:

- Inte klara tävlingsmoment ett på 10 minuter.
- Välta en vägg i tävlingsmoment två
- Släppa ner förnödenheten utanför rektangeln som den ska placeras i.
- Hela förnödenheten är utanför markeringen för den nödställda när den släpps ner.

Följande ger tidstillägg:

- Nudda en vägg, 10s.
- Någon del av förnödenheten är utanför markeringen för den nödställda när den släpps ner eller efter att hela körningen är klar, 15s.
- Hela förnödenheten är utanför markeringen för den nödställda efter att hela körningen är klar, 30s

Tävling 1

Första fasen kommer mätas som en separat tävling för ära. Här börjar tiden räknas när första delen av roboten passerar igenom ingången och slutar när den undsatte har blivit funnen och sista delen av roboten passerat ut genom ingången.

Tävling 2 (den stora tävlingen)

Den andra tävlingen, den som verkligen spelar roll, utspelear sig i fas 2. Tideräkningen börjar efter att förnödenheten har blivit upplockad och första delen av roboten åker in i banan. Sedan skall roboten släppa förnödenheten i rutan som markerar de nödställda. Sedan skall roboten återvända, tiden slutar när sista delen av roboten lämnar banan.

D SYSTEMSKISS

Systemskiss

Alice Nicklasson Cedbro, Maja Boström, Johanna Carlson, Lovisa Byman, Filip Johansson,
Edvin Leuchovius, Gunnar Arctaedius

30 april 2021

Version 1.0



Status

Granskad	MB	2021-02-16
Godkänd	Alla	2021-02-16

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedius		gunar889@student.liu.se

INNEHÅLL

1	Inledning	1
2	Översikt av systemet	1
3	Kommunikationsmodul	3
3.1	Trådlösa standarder	3
4	Sensormodul	4
4.1	Sensorer	4
4.2	Mikroprocessor	5
4.3	Mjukvara	5
4.4	Övrig hårdvara	6
5	Styrmodul	6
5.1	Beskrivning	6
5.2	Problem	6
6	Datormodul	11
6.1	Övervakning	11
6.2	Styra	11
6.3	Karta	12
	Referenser	13

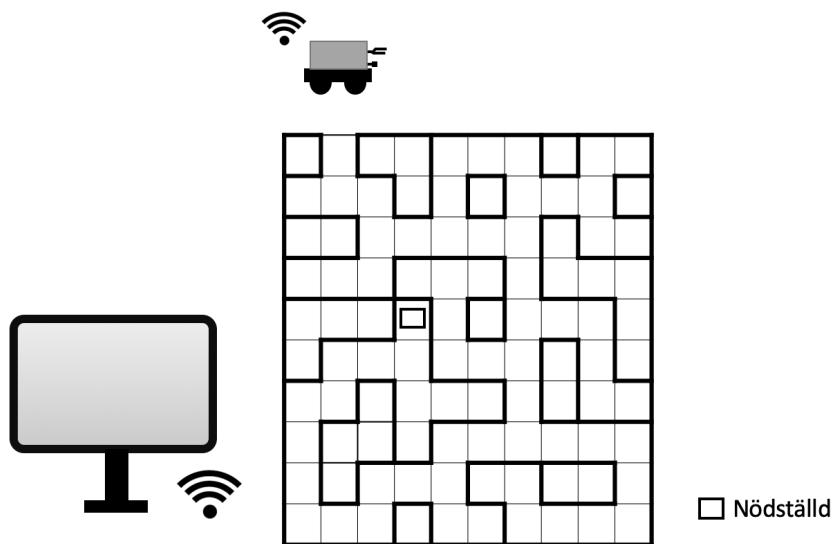
DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-02-11	Första utkast	Alla	Alla
1.0	2021-02-16	Andra utkast	LB	LB och MB

1 INLEDNING

Detta dokument utgör en skiss till en prototyp av en undsättningsrobot. Undsättningsroboten, häданefter kallad ”roboten”, ska autonomt kartlägga en byggnad, finna en nödställd, hämta en förnödenhet vid ingången till byggnaden och sedan ta snabbaste vägen genom byggnaden till de nödställda för att lämna förnödenheten.

Figur 1 visar hela systemet tillsammans med byggnaden. Systemet kommer att bestå av roboten som utför uppdraget, samt ett program på en bärbar dator för att kommunicera med roboten. Roboten kommer att ha flera motorer för att styra hjul och en gripklo, och den kommer att utföra uppdragene med hjälp av flera sensorer och processorer.



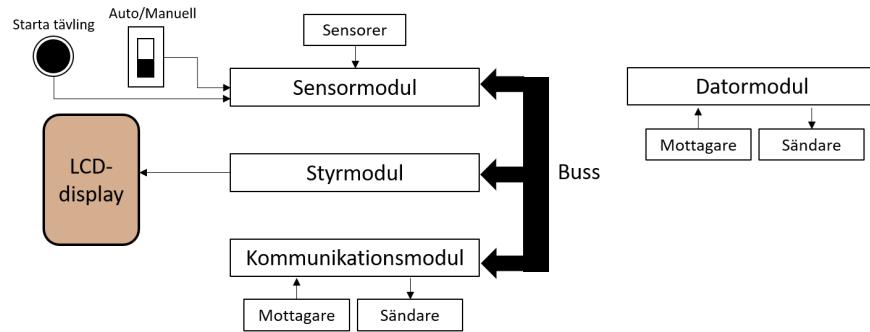
Figur 1: Roboten i dess omgivning.

2 ÖVERSIKT AV SYSTEMET

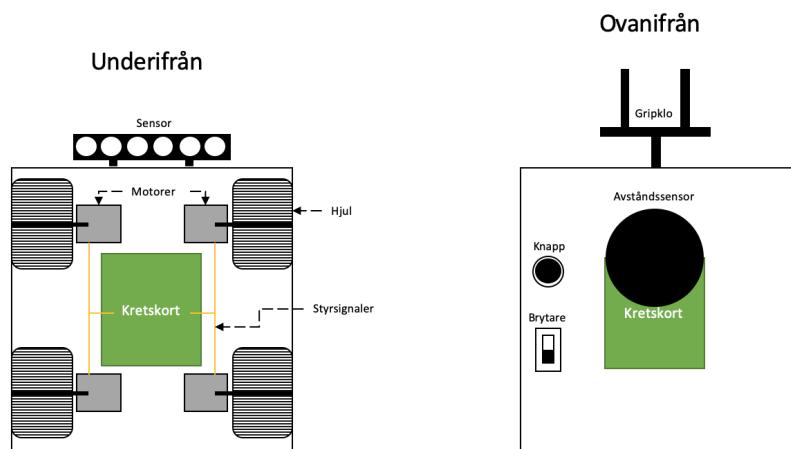
Enligt kravspecifikationen kommer systemet, både undsättningsrobot och övervakning/styrning, att bestå av en kommunikationsmodul, en sensormodul, en styrmodul samt en datormodul [1]. Figur 2 visar ett blockschema över hur modulerna och ett par viktiga gränssnitt kan kopplas samman.

Vid val av möjliga komponenter till roboten, så som sensorer, processorenhet, etc., används databladsservern Vanhe-den som tillhandahålls av ISY för detta projekt [2].

Figur 3 visar en schematisk bild av roboten och några av de ingående komponenterna. För konstruktion av roboten kommer det färdiga chassis Terminator – komplett med hjul, elmotorer, batteri, och virkort – att användas [3].



Figur 2: Blockschema som visualiseras hur de ingående modulerna kan kopplas samman.

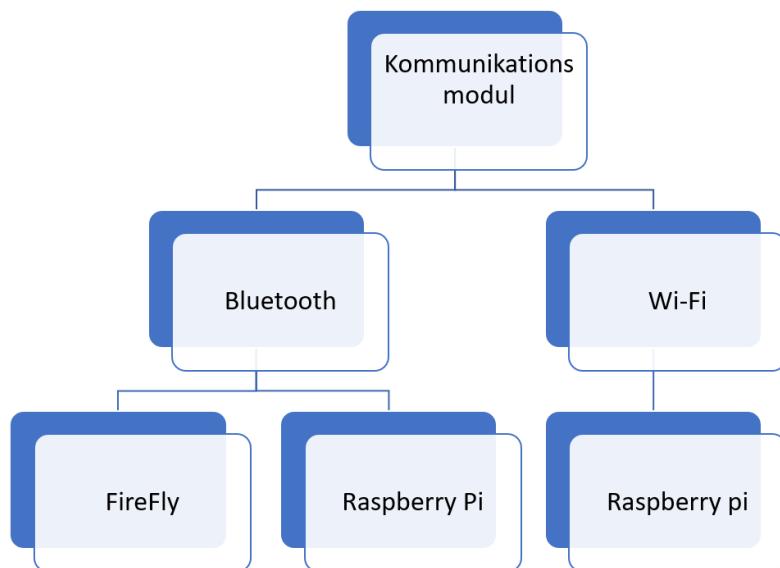


Figur 3: Schematisk bild av roboten.

3 KOMMUNIKATIONSMODUL

En av kommunikationsmodulens huvuduppgifter är att kommunicera med en PC trådlöst. Data ska alltså bådes kunna skickas till respektive tas emot av datormodulen. Därför finns det principiellt sätt två viktiga designbeslut: val av trådlös standard och val av processorenhet. Figur 4 visar ett antal olika sätt som dessa val kan kombineras på, vilka beskrivs översiktligt i detta avsnitt.

Data ska dessutom kunna transporteras till styrmodulen och sensormodulen på ett effektivt sätt.



Figur 4: Visualisering av möjliga designval.

3.1 Trådlösa standarder

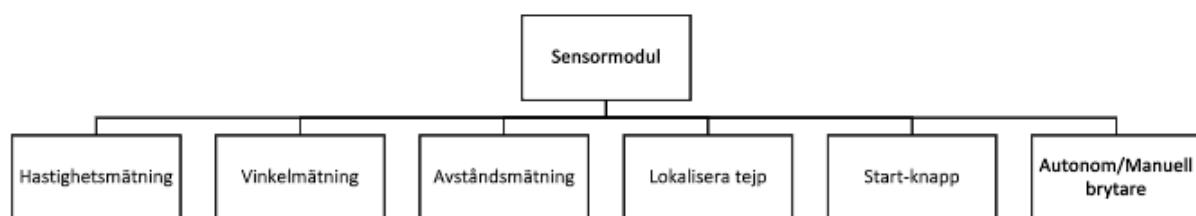
Ett alternativ är att kombinera en AVR-mikrokontroller med Bluetooth-modemet FireFly [4]. Modemet är fysiskt ett litet kort och har låg strömförsörjning, vilket ger fördelen att kortet enkelt kan monteras och anslutas. Det finns två relevanta AVR-modeller som är lättillgängliga för projektet: AVR16 och AVR1284. De har RISC-arkitektur och går att programmerabåde i C och assembler.

Ett annat alternativ är att använda enkortsdatorn Raspberry Pi, vilken har både Bluetooth och Wi-Fi inbyggt. Fördelar med Raspberry Pi är enkel programmering av mjukvara med många färdiga bibliotek som kan användas. Nackdelar som behöver ingå i övervägningen av detta designval är: att kortet kräver spänningssmatning och strömförbrukning på 5 V respektive 1.5 A, vilket kommer belasta Terminator-chassits batteri tungt och att kortet kommer således att kräva mer utrymme på chassit.

För att en AVR och en Raspberry Pi ska kunna kommunicera behövs en nivåskiftare [5] som höjer spänningen på bussen.

4 SENSORMODUL

Robotens sensormodul kommer innehålla olika sensorer för att läsa av omgivningen den befinner sig i, se figur 5, brytare för att byta mellan manuell och autonom körning samt en startknapp för att påbörja körningen.



Figur 5: Visuallisering av möjlig design av sensormodulen.

4.1 Sensorer

Sensorernas uppgift involverar att läsa av avstånd till väggar, söka efter tejpmärkningar fastsatta på marken, läsa av med vilken vinkel roboten styrs samt beräkna robotens hastighet. För en översikt av möjliga sensorer se tabell 1.

En 360-graders roterande laser (LIDAR)[6] kan användas för att mäta avstånd till väggar, och eventuellt beräkna tillryggalagd sträcka samt styrningsvinkel för roboten. Den roterande lasern specificerad i detta dokument kan mäta avstånd från 15 cm till 8 meter.

Reflexsensorer för att finna de nödställda respektive förnödenheter ses som ett möjligt alternativ då dessa sensorer kan registrera svarta tejpmärkningar på golvet vilket är vad nödställda och förnödenheter kommer markeras med. [7]

Robotens styrningsvinkel, dvs hur mycket roboten avviker från rak körning framåt, kan detekteras med hjälp av gyro. Alternativt kan en vinkelhastighetssensor[8] användas som mäter rotation i en led eller ett 3D-gyro[9][10] som mäter rotation i tre ledar. Även robotens hastighet kan mätas med gyro genom odometri. Vid odometri används hjulens rotation samt diameter för att uppskatta robotens nuvarande position samt tillryggalagd sträcka. Ett alternativ för att beräkna robotens hastighet är en fristående 3D-accelerometer.[11] [12] [13]

Sensortyp	Modell	Användningsområde
Roterande laser	RP-Lidar [6]	Mäter avstånd.
Reflexsensor	Reflexsensor [7]	Registrera svarta tejpmärkningar.
Gyro	MLX90609 [8]	Mäter vridning i en led.
3D-Gyro	Adafruit 10-DOF[9]	Mäter vridning i 3 ledar.
3D-Gyro	Adafruit Precision NXP 9-DOF[10]	Mäter vridning i 3 ledar.
Accelerometer	ST LIS3L02AS4 [11]	Mäter accelerationen.
Accelerometer	Adafruit 10-DOF[12]	Mäter accelerationen.
Accelerometer	Adafruit Precision NXP 9-DOF[13]	Mäter accelerationen.

Tabell 1: Lista över möjliga sensorer

4.1.1 *Placering*

För optimal avläsning bör den roterande lasern placeras centrerat på robotens ovansida (förbestämt fäste) där även accelerometern skulle kunna placeras. Reflexsensorn får bäst avläsning vid placering på undersidan av roboten nära dess framsida, möjligvis något framskjuten för att så tidigt som möjligt upptäcka tejp på marken.

Gyrot för hastighetsmätning är lämpligast att montera på hjulen, medan gyro för styrningsvinkel inte är lika beroende av placering utan även det skulle kunna placeras på robotens ovansida. Av praktiska skäl kan även accelerometrar placeras på ovansidan, men dessa skulle lika gärna kunna sitta på underredet.

4.1.2 *Tidssampling*

Troligtvis räcker en lägre samlingsfrekvens för roterande laser och reflexsensor då dessa har större avstånd att mäta. Gyrot samt accelerometer kommer troligen behöva högre frekvens då styrningen för de betydligt mindre avstånden hos hjulen kommer kräva högre förändringstakt för mjuk körning. Därmed skulle 200 gånger per sekund för reflexsensor och 100 gånger per sekund för roterande laser vara rimlig lägsta samplingfrekvens. För gyro samt accelerometer bör 1000 mätningar per sekund vara rimligt.

4.2 Mikroprocessor

Lämpliga mikroprocessorer till sensormodulen listas nedan 2 med mest väsentliga skillnad i kapacitet.

Modell	Flashminne [kByte]
ATmega1284P[14]	128
ATmega16[15]	16

Tabell 2: Lista över möjliga processorer

4.3 Mjukvara

Processorn i sensormodulen tar in sensordata och bearbetar denna till lämpligt format. Daten skickas sedan till styrmodul och kommunikationsmodul.

4.4 Övrig hårdvara

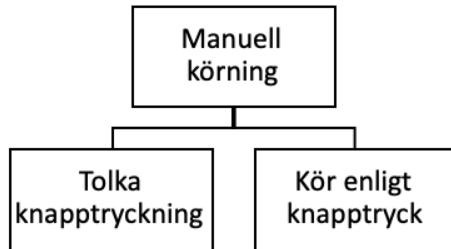
Sensormodulen skall innehålla en brytare för att möjliggöra växling mellan manuell och autonom körning, samt en knapp för att starta roboten vid tävlingsmomentets start. Möjliga alternativ för detta ändamål är en tryckknapp [16] eller en avstudsad tryckknappsmodul [17]. Brytaren ska placeras på sensormodulen där den är lättåtkomlig.

5 STYRMODUL

5.1 Beskrivning

Styrmodulen bestämmer vad roboten ska göra och styr motorerna för att utföra det. Den får information från sensormodulen och använder informationen för att göra val gällande vad den ska göra och hur motorerna ska styras. Styrmodulen skickar även all styrdata till kommunikationsmodulen, samt tar emot kommandon från gränssnittet via kommunikationsmodulen.

Styrmodulen kommer att bete sig olika i autonomt och manuellt läge. I manuellt läge kommer den att ta emot kommandon från användaren och sedan styra motorerna för att utföra kommandot, se fig 6. I det autonoma läget kommer styrmodulen att bestämma hur roboten ska köra och vad den ska göra. Det kommer bland annat innebära att kartlägga labyrinten, hämta förnödenheten, beräkna kortaste väg till de nödställda, med mera, se fig 7 och fig 8.



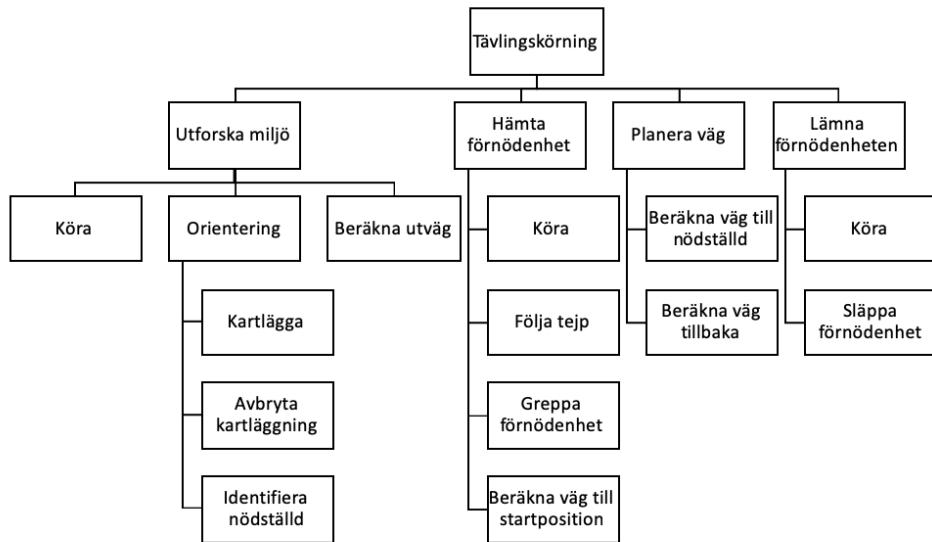
Figur 6: Handlingar att utföra under manuell köring.

Styrmodulen kommer bestå av en processor och fyra motorer som styrs parvis. Den kommer att vara sammankopplad med kommunikationsmodulen och sensormodulen. Den kommer även ha en LCD-display för att visa sensordata.

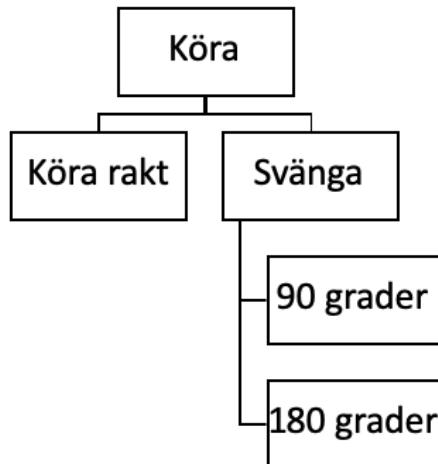
5.2 Problem

5.2.1 Reglering

Styrmodulen ska styra gaspådraget på motorerna för att få roboten att köra framåt och rotera. För att köra framåt ska robotens båda motorpar köra åt samma håll, och för att rotera ska motorparens köra åt olika håll. För att göra detta på ett kontrollerat sätt måste systemet regleras. Detta skulle kunna göras med en PID-regulator.



Figur 7: Handlingar att utföra under tävlingskörning.



Figur 8: Handlingar för att köra.

Systemet behöver även ha någon återkoppling för att kunna reglera gaspådraget, vilket innebär att data från sensorerna krävs för att mäta hastighet och vinkelhastighet. För att göra de mätningarna kan odometri, accelerometer och Lidar användas.

5.2.2 **Greppa fornödenhet**

Styrmodulen ska kunna hitta och plocka upp fornödenheten. För att göra detta ska den kunna hitta tejpen som markerar varft fornödenheten finns med hjälp av data från sensormodulen. Den ska sedan följa tejpen, ställa in sig rakt mot fornödenheten, och stanna vid en position som gör det möjligt för den att plocka upp objektet med gripklon. Gripklon ska alltså också stängas av med styrmodulen. Därefter ska roboten ta sig till starten av banan autonomt.

5.2.3 **Övergripande**

Roboten ska styras i två olika lägen, ett manuellt och ett autonomt. Det ska finnas en brytare som väljer vilket läge roboten är i, vilken tillhör sensormodulen. För styra roboten på olika sätt kan det antingen finnas två olika program som körs beroende på vilket läge roboten är i. Det kan också finnas ett övergripande program som alltid körs och som i sin tur kör olika funktioner beroende på om roboten ska köra manuellt eller autonomt.

5.2.4 **Kartläggning**

För att finna de nödställda kommer någon strategi för kartläggning att användas, till exempel SLAM. Efter att de nödställda har hittats beräknas kortaste vägen tillbaka till startpunkten som hittills har upptäckts. Därefter fortsätter kartläggningen. I denna fas av kartläggningen så kartläggs bara de delar av labyrinten som behöver kartläggas. Delar som inte behöver kartläggas är till exempel de områden som, under antagandet att området inte innehåller några väggar, ändå inte genererar en väg som är kortare än nuvarande väg.

5.2.5 **Beräkna kortaste vägen**

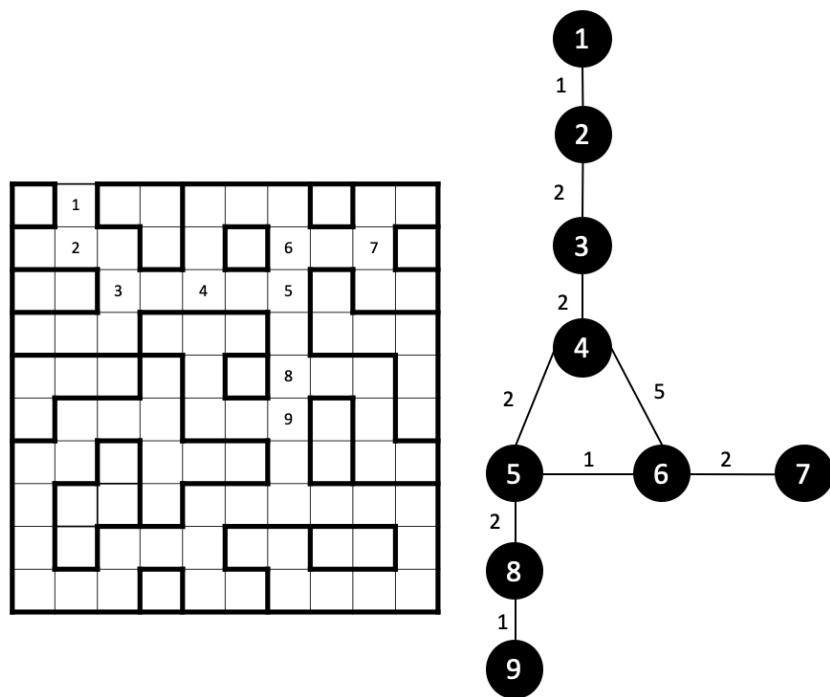
För att beräkna kortaste vägen till den nödställda kommer ett träd konstrueras, se exempel i fig 9, notera dock att trädet inte är fullständigt. Därefter beräknas kortaste vägen med någon algoritm för minsta uppspänrande träd, till exempel Dijkstras algoritm eller A*.

5.2.6 **Representation av kartan**

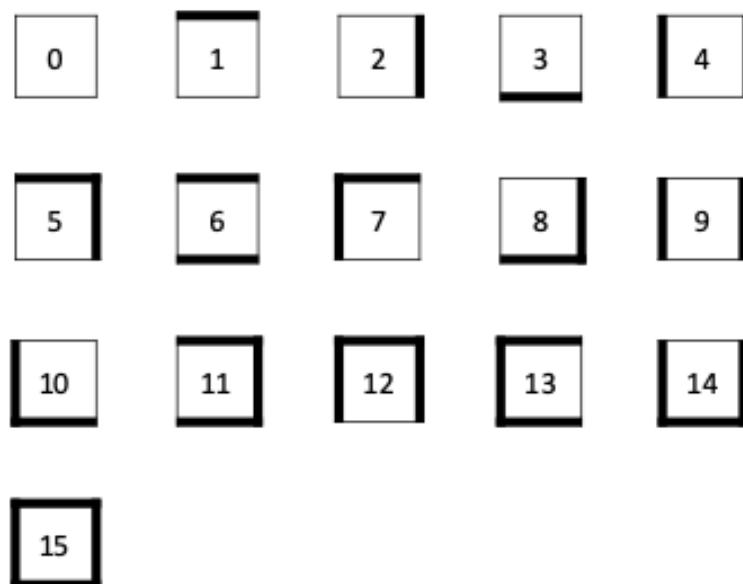
För att lagra kartan i styrmodulen och för att skicka kartan med hjälp av kommunikationsmodulen till datormodulen så kommer kartan att representeras som en matris. Matrisen kan se ut på lite olika sätt. Ett sätt är att låta varje element i matrisen representera möjliga väggar, där 1 betyder att det är en vägg där, och 0 betyder att det inte är en vägg där. En fördel med detta sätt att representera kartan är att det blir smidigt för datormodulen att rita upp karten.

Ett annat sätt att låta varje element i matrisen representera en modul i labyrinten. Modulerna ges ett värde beroende på var modulen har väggar, se fig 10. Fördelen med detta sätt är att det blir lättare att ta fram ett träd av noder för att sedan kunna beräkna kortaste vägen, se 5.2.5. Nackdelen är att informationen om vilka väggar som finns kommer att upprepas, eftersom två intilliggande noder båda kommer innehålla information om samma vägg.

Ett sätt att slippa upprepad information är att göra på samma sätt men istället bara spara information om övre och högra väggen i varje element. På så sätt upprepas ingen information men samtidigt blir det också svårare att ta fram trädet av noder som krävs för att kunna beräkna kortaste vägen.



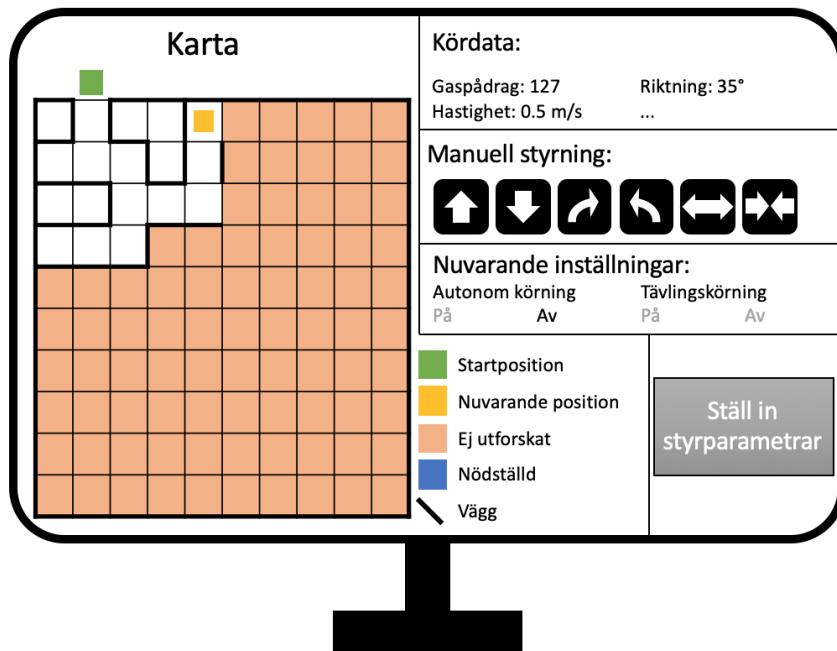
Figur 9: Träd för att kunna beräkna kortaste vägen till den nödställda



Figur 10: Värden på modulerna beroende på vart modulen har väggar.

6 DATORMODUL

Datormodulen är ett program som körs på en bärbar dator. Programmet kommer att bestå av ett grafiskt användargränssnitt som en slutanvändare kan använda för att övervaka och styra roboten.



Figur 11: Användargränssnitt.

6.1 Övervakning

All data som skickas till datormodulen sparas ner i separata filer som baseras på vilken typ av data det är. Styrbeslut samlas i en fil och de olika sensorernas data sparas alla i olika filer.

6.2 Styra

Den manuella styrningen kan komma att fungera på två olika sätt, diskret eller kontinuerlig. Att ha diskret styrning skulle innebära att när ett kommando ges utför roboten det och sedan väntar på nästa kommando, t.ex. om kommandot ”pil upp” gavs då kör roboten ett bestämt avstånd framåt och väntar sedan på nästa kommando. Att ha kontinuerlig köring skulle däremot fungera mer som en radiostyrd bil där roboten utför kommandon medan en knapp trycks, t.ex. om knappen som signalerar framåt trycks ner tio sekunder kör roboten framåt.

6.3 Karta

Datormodulen kan tänkas få kartdata i ett nödsystem från roboten och i så fall måste noderna tolkas för att få ut en karta.

Om datormodulen endast får sensordata och styrbeslut måste kartan byggas upp endast från denna data.

REFERENSER

- [1] A. Nicklasson Cedbro, E. Leuchovius, F. Johansson, G. Arctaedius, J. Carlson, and et al., “Kravspecifikation,” februari 2021, version 1.0.
- [2] “Vanheden,” <http://www.da.isy.liu.se/vanheden/page/robotplattformar/>, [Online; hämtad Februari 9, 2021].
- [3] N. Carlén, “Fyrhjulig robot – Terminator,” http://www.da.isy.liu.se/vanheden/pdf/terminator_prel.pdf, [Online; hämtad Februari 9, 2021].
- [4] “FireFly Bluetooth Modem BlueSMiRF Gold,” <http://www.da.isy.liu.se/vanheden/pdf/firefly.pdf>, [Online; hämtad Februari 8, 2021].
- [5] “Nivåskiftare Raspberry,” <http://www.da.isy.liu.se/vanheden/pdf/da-nivaskiftare.pdf>, [Online; hämtad Februari 8, 2021].
- [6] Slamtec, “RPLIDAR A2,” http://www.da.isy.liu.se/vanheden/pdf/rplidar_a2m8_v1.0.pdf, [Online; hämtad Februari 8, 2021].
- [7] “Reflexsensor,” http://www.da.isy.liu.se/vanheden/pdf/reflex_sensor.pdf, [Online; hämtad Februari 8, 2021].
- [8] “Angular Rate Sensor,” <http://www.da.isy.liu.se/vanheden/pdf/mlx90609.pdf>, [Online; hämtad Februari 8, 2021].
- [9] “MEMS motion sensor: three-axis digital output gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/L3GD20.pdf>, [Online; hämtad Februari 8, 2021].
- [10] “3-Axis Digital Angular Rate Gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/FXAS21002.pdf>, [Online; hämtad Februari 8, 2021].
- [11] “Mems Inertial Sensor,” <http://www.da.isy.liu.se/vanheden/pdf/st-lis3l02as4.pdf>, [Online; hämtad Februari 8, 2021].
- [12] “Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module,” <http://www.da.isy.liu.se/vanheden/pdf/LSM303DLHC.pdf>, [Online; hämtad Februari 8, 2021].
- [13] “6-axis sensor with integrated linear accelerometer and magnetometer,” <http://www.da.isy.liu.se/vanheden/pdf/FXOS8700CQ.pdf>, [Online; hämtad Februari 8, 2021].
- [14] “ATmega1284P,” <http://www.da.isy.liu.se/vanheden/pdf/atmega1284p.pdf>, [Online; hämtad Februari 8, 2021].
- [15] “ATmega16,” <http://www.da.isy.liu.se/vanheden/pdf/atmega16.pdf>, [Online; hämtad Februari 8, 2021].
- [16] “Så här fungerar tryckknappen,” <http://www.da.isy.liu.se/vanheden/image/tryckknapp.png>, [Online; hämtad Februari 9, 2021].

- [17] “Avstudsad tryckknappsmodul 2x1,” http://www.da.isy.liu.se/vanheden/pdf/avstudsad_tryckknappsmodul_2x1.pdf, [Online; hämtad Februari 9, 2021].

E PROJEKTPLAN

Projektplan

Alice Nicklasson Cedbro, Maja Boström, Johanna Carlson, Lovisa Byman, Filip Johansson,
Edvin Leuchovius, Gunnar Arctaedius

30 april 2021

Version 1.0



Status

Granskad	Alla	2021-02-16
Godkänd	Alla	2021-02-16

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedius		gunar889@student.liu.se

INNEHÅLL

1	Beställare	1
2	Översiktig beskrivning av projektet	1
2.1	Syfte och mål	1
2.2	Leveranser	1
2.3	Begränsningar	1
3	Organisationsplan	4
3.1	Villkor för samarbete	4
3.2	Arbetsinnehåll och ansvar	4
4	Dokumentplan	5
5	Utvecklingsmetodik	6
6	Utbildningsplan	6
6.1	Egen utbildning	6
7	Rapporteringsplan	6
8	Mötesplan	6
9	Resursplan	7
9.1	Personer	7
9.2	Material	7
9.3	Lokaler	7
9.4	Ekonomi	7
10	Milstolpar och beslutspunkter	7
10.1	Milstolpar	7
10.2	Beslutspunkter	7
11	Aktiviteter	9
11.1	Aktiviteter tillhörande sensormodulen	9
11.2	Aktiviteter tillhörande styrmodulen	9
11.3	Aktiviteter tillhörande kommunikationsmodulen	10
11.4	Aktiviteter tillhörande datormodulen	10
11.5	Övergripande aktiviteter	11
12	Prioriteringar	11
13	Projektavslut	11
	Referenser	12
A	Gruppkontrakt	13

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-02-11	Första utkast	Alla	Alla
1.0	2021-02-16	Andra utkast	Alla	Alla

1 BESTÄLLARE

Beställare för projektet är Kent Palmkvist. För kontaktinformation till beställare se Projektidentitet ovan.

2 ÖVERSIKTlig BESKRIVNING AV PROJEKTET

Projektet är ett kandidatarbete i elektronik där kursdeltagarna ges möjlighet att öva på praktisk elektronikkonstruktion. Det finns flera olika val av projektdirektiv, som var och en beskriver en datorstyrda apparat [1]. Projektgruppen har valt att konstruera en autonom undsättningsrobot.

2.1 Syfte och mål

Målet med projektet är att – tillsammans med projektgruppen och med stöd av handlare – designa och konstruera en autonom undsättningsrobot som uppfyller de krav vilka finns i kravspecifikationen som följer med projektet [2]. Roboten ska kunna köra i en bana och dessutom delta i en tävling mot andra projektgrupper. Figur 1 visar en schematisk skiss av systemet i den omgivning som tävlingen ska ske i.

Systemet är uppbyggt av fyra moduler; en sensormodul, en styrmodul, en kommunikationsmodul och en datormodul. Modulerna har ansvar över olika delar av systemet, och varje del kommer att vara individuellt utbytbar. Sensormodulen kommer att bestå av flera sensorer, styrmodulen av fyra parvis kontrollerade hjul och en gripklo, kommunikationsmodulen medel för att trådbundet och trådlöst kommunicera med de andra modulerna, och datormodulen ett användargränssnitt. Utöver detta kommer varje modul bestå av en minst en processor var. Sensormodulen, styrmodulen och kommunikationsmodulen bygger tillsammans upp roboten, och datormodulen sköts från en bärbar dator.

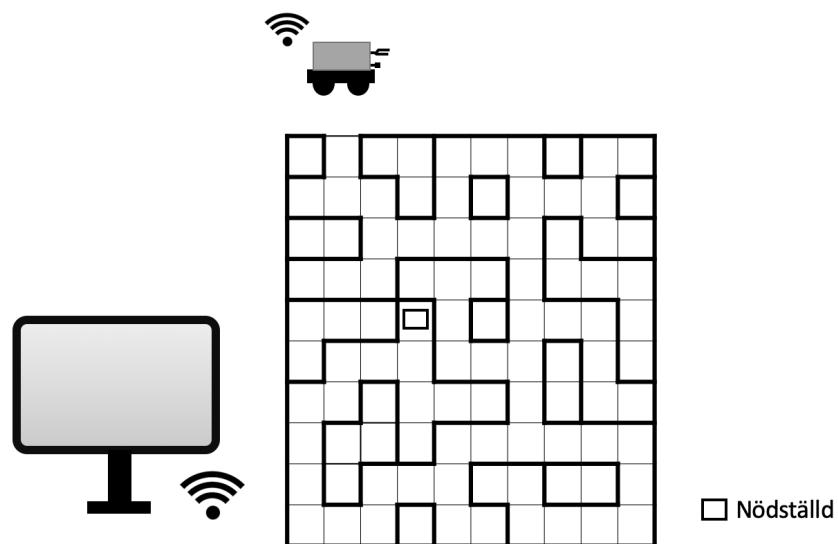
Detaljer kring tävling och bana specificeras i Ban- och tävlingsspecifikationen [3].

2.2 Leveranser

Leveranser kommer att ske enligt tabell 1. Dessa är hämtade från ”Leveranser” på kurshemsidan [4].

2.3 Begränsningar

Projektet kommer inte involvera att undersöka vidareutveckling av produkten, riskanalys eller utbildning av kund.



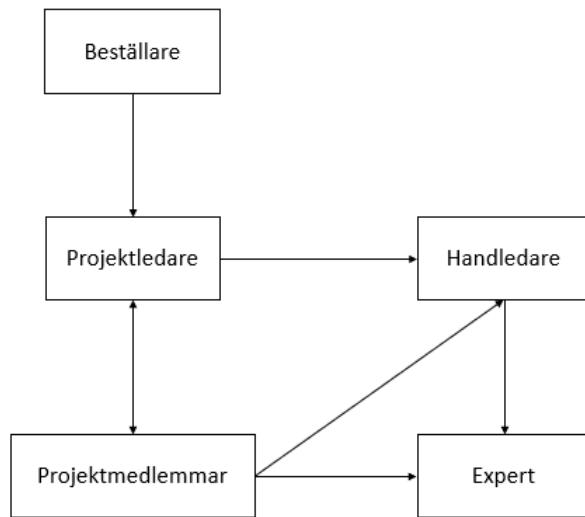
Figur 1: Roboten i dess omgivning.

Tabell 1: Leveranser under projektet.

Leverans	Datum	Tillhörande beslutspunkt
Systemskiss, projektplan, tidplan v0.1	21-02-11	BP2
Systemskiss, projektplan, tidplan v1.0	21-02-18	BP2
Tidrapport för systemskiss, projektplan och tidplan	21-02-22	BP2
Designspecifikation v0.1	21-03-03	BP3
Designspecifikation v1.0	21-03-12	BP3
Tidrapport	21-03-29	-
Tidrapport	21-04-06	-
Tidrapport	21-04-12	-
Presentation av nuvarande design	21-04-16	BP4
Tidrapport	21-04-19	-
Tidrapport	21-04-26	-
Tidrapport och statusrapport	21-05-03	-
Demonstration av baskrav	21-05-05	BP5a
Tidrapport	21-05-10	-
Tidrapport	21-05-17	-
Verifiering av slutkrav	21-05-19	BP5b
Kandidatrapport v1.0 exklusive appendix	21-05-18	-
Teknisk dokumentation, användarhandledning v1.0	21-05-19	-
Tidrapport	21-05-24	-
Framläggning och opposition	21-05-31	-
Tidrapport	21-05-31	-
Tävling och demonstration	21-06-01	-
Kandidatrapport inklusive appendix	21-06-07	-
Efterstudie	21-06-07	-
Tidrapport	21-06-07	-
Återlämning av utrustning	21-06-07	-

3 ORGANISATIONSPLAN

Kommunikationen mellan parter i projektet kommer att ske enligt figur 2. Organisationen består av beställare, projektledare, handledare, projektmedlemmar samt experter.



Figur 2: Projektets organisation

3.1 Villkor för samarbete

För att möjliggöra ett väl fungerade projektarbete har ett gruppkontrakt skrivits, där b.la. villkor för samarbete listas, se Appendix A.

3.2 Arbetsinnehåll och ansvar

I projektet kommer ansvarsområdena att uppkomma vartefter de respektive modulerna och delarna i projektet påbörjas. Ansvarsområdena och roller kommer att fördelas löpande allt eftersom behoven av en ansvarig för respektive del uppkommer. Detta för att alla ska ta ett mer gemensamt ansvar i projektets början, vilket förhoppningsvis leder till att projektmedlemmarna blir mer delaktiga i projektets alla delar. Roller som redan är satta ses i 5. Roller i appendix A.

4 DOKUMENTPLAN

Lista över alla dokument som ska produceras anges i tabell 2.

Tabell 2: Dokument som skall produceras under kursen.

Dokument	Ansvarig	Godkänner	Syfte	Distributeras till	Färdigt
Kravspecifikation	Projekt-medlemmar	Beställare	Visa vilka krav som roboten i slutet ska testas mot.	Beställare	21-02-03
Ban- och tävlingsspecifikation	Projekt-medlemmar	Beställare	Specificera gemensamma regler för tävling samt specificera banans utformning	Beställare	21-02-03
Projektplan	Projektledare	Beställare	Ge en plan för när milstolpar ska nås.	Beställare	21-02-18
Tidrapport	Projekt-medlemmar	Beställare	Ge en översikt över varje gruppmedlems nedlagda tid.	Beställare	21-02-18
Designspecifikation	Projekt-medlemmar	Handledaren	Robotens design att utgå från under uppgyggnad.	Beställare och handledare	21-03-12
Reglerförstudie	Projekt-medlemmar	Beställare	Studera möjliga reglerlösningar	Beställare	21-04-08
Sensorförstudie	Projekt-medlemmar	Beställare	Studera möjliga sensorlösningar.	Beställare	21-04-08
Kommunikation och konstruktion förstudie	Projekt-medlemmar	Beställare	Studera möjliga kommunikationslösningar.	Beställare	21-04-08
Kappa	Projekt-medlemmar	Beställare	Ge en komplett överblick av det genomförda arbetet.	Beställare	21-05-18
Teknisk dokumentation	Projekt-medlemmar	Beställare	Förklara hur roboten är byggd.	Beställare	21-05-19
Användarhandledning	Projekt-medlemmar	Beställare	Förklara hur en användare ska använda systemet.	Beställare	21-05-19
Efterstudie	Projekt-medlemmar	Beställare	Ge en lätt översikt över hur projektet har gått.	Beställare	21-06-07

5 UTVECKLINGSMETODIK

Utvecklingsmetodik för projektets aktiviteter bestäms internt mellan aktivitetens utförare, men jobb i grupper om två eller tre personer är att föredra. En aktivitet anses avslutad då ansvariga gruppmedlemmar har utformat, utfört och dokumenterat tester för denna.

6 UTBILDNINGSPLAN

6.1 Egen utbildning

Tre laborationer ger möjlighet till övergripande träning inför konstruktionen av roboten; den första laborationen handlar om mätteknik, den andra om programmering av AVR-mikrokontroller och den tredje om I2C-bussar.

Utöver detta så kommer tre förstudier att skrivas med målet att kunna ge kompletterande insikter framförallt inför konstruktionen av systemet. Förstudierna utgörs alltså av tre rapporter, listade nedan.

- En regleruppgift som undersöker vilka övergripande styrmoder som är nödvändiga från ett reglertekniskt perspektiv.
- En sensoruppgift som undersöker hur en roterande laser (LIDAR) kan på roboten.
- En kommunikationsuppgift som undersöker ”olika principer för kommunikation mellan processorer.” [5]

7 RAPPORTERINGSPLAN

Elva stycken tidrapporter med tillhörande statusuppdatering ska skrivas. Punkterna i dessa rapporter diskuteras av projektgruppen gemensamt under slutet av lämplig vecka och att skriva rapporterna samt uppdatera tidplan kommer någon i projektgruppen få i uppdrag att göra efteråt. Tidrapporterna kommer levereras till beställare och beskriva de framsteg som gjorts sedan tidigare statusrapport, aktuella problem och plan för kommande arbete. Leveransdatum för tidrapporter återfinns under 2.2, tabell 1.

8 MÖTESPLAN

I slutet av varje vecka ska ett avstämningsmöte utföras där punkterna till tidrapporten diskuteras. Detta ska vara ett kortare möte på runt 40 min.

9 RESURSPLAN

9.1 Personer

Projektdeltagarnas inställning är att i största möjliga mån arbeta med projektet under icke-schemalagd arbetsstid, dvs vardagar 8-17. Detta inkluderar inte studietid inför och under tentamensveckor (v.11-12), eller eventuella klämdagar. Hänsyn kommer även visas för privata resor sålänge detta meddelas i förväg och eget ansvar tas för omplanering av arbetsstid. Under vårterminens andra hälft är det även viktigt att kunna arbeta på plats då arbete med hårdvara i bokad labbsal är aktuell.

9.2 Material

Projektgruppens förhoppning är att de önskade sensorer, processorer, motorer m.m. som listas i systemskissen kan distribueras av kursansvariga, samt att chassi och övrig hårdvara finns tillgå i labbsalar eller genom utlämning.

9.3 Lokaler

Under projektets andra del kommer tid behöva spenderas i diverse labbsalar såsom Muxen och Visionen för att testa systemets helhet. Därav kommer en begränsning finnas i tillgänglighet i och med att bokning krävs för dessa salar. Således är planering kring tillfällena en viktig del under denna tid, liksom att ha uppsikt över hur många som vistas i lokalerna där ett maxantal är viktigt att respektera.

9.4 Ekonomi

Det totala antalet timmar på projektet är 1610 h detta innebär att varje person ska jobba 230 h.

10 MILSTOLPAR OCH BESLUTSPUNKTER

10.1 Milstolpar

I detta avsnitt definieras milstolpar för projektet, vilka utgör övergripande delmål i utvecklingsfasen. Dem går att se i tabell 3.

10.2 Beslutspunkter

Beslutspunkter är övergripande händelser i projektet som ser till att projektet flyter på enligt den valda projektmodellen. I princip så markerar en beslutspunkt slutet på en fas vilket ofta är en förutsättning för att nästkommande ska kunna påbörjas. Tabell 4 listar de beslutspunkter som fastställts av kursanvariga.

Tabell 3: Milstolpar.

Milstolpe	Beskrivning	Färdig
1	Kunna åka rakt reglerat.	v13
2	Kunna skicka data från kommunikationsmodul och ta emot data med datormodulen trådlöst.	v14
3	Sensormodulen klar.	v15
4	Köra roboten manuellt.	v16
5	Kan bestämma och åka kortaste vägen.	v16
6	Färdigmonterad och fungerande robot.	v17
7	Rita upp karta på datorn.	v18
8	Roboten är tävlingsklar.	v19

Tabell 4: Beslutspunkter.

Beslutspunkt	Innehåll	Datum
BP0	Projektgrupp och projektuppgift ska vara fastlagda.	2021-01-22
BP1	Godkänd kravspecifikation	2021-02-03
BP2	Godkänd systemskiss, projektplan och tidplan	2021-02-18
BP3	Godkänd designspecifikation	2021-03-12
BP4	Godkänd nuvarande design	2021-04-16
BP5a	Godkända baskrav	2021-05-05
BP5b	Godkända slutkrav	2021-05-19

11 AKTIVITETER

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
1	Original	Färdigställ designspecifikationen.	100 h	-

11.1 Aktiviteter tillhörande sensormodulen

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
2	Original	Identifiera sensordata med dator.	10	1
3	Original	Koppla ihop sensorer med processor.	20	2
4	Original	Filtera avståndsdata.	20	3
5	Original	Bearbeta avståndsdata.	20	4
6	Original	Få ut avstånd.	10	5
7	Original	Filtrera tejpdata.	20	3
8	Original	Bearbeta tejpdata.	20	7
9	Original	Få ut tejpdata.	10	8
10	Original	Filtera hastighetsdata.	20	3
11	Original	Bearbeta hastighetsdata.	20	10
12	Original	Få ut hastighet.	10	11
13	Original	Filtera vinkelhastighetsdata.	20	3
14	Original	Bearbeta vinkelhastighetsdata.	20	13
15	Original	Få ut vinkelhastighet.	10	14
16	Original	Reagera på knapptryck.	10	3
17	Original	Skicka data.	20	6, 9, 12, 15
18	Original	Montera sensorer, processor och knappar på robotten.	15	17

11.2 Aktiviteter tillhörande styrmodulen

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
19	Original	Styra motorer.	16	1
20	Original	Implementera regleralgoritm för att köra rakt.	40	19
21	Original	Implementera regleralgoritm för att rotera.	32	19
22	Original	Svänga 90°och 180°.	8	19, 21
23	Original	Styra gripklo.	10	1
24	Original	Reagera på simulerad sensordata.	8	-

forts. på nästa sida

<i>forts. från föregående sida</i>				
Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
25	Original	Reagera på data från sensormodul.	36	17
26	Original	Ta emot data från kommunikationsmodulen.	16	37
27	Original	Skicka styrdata.	16	1
28	Original	Kartlägga.	30	1
29	Original	Implementera algoritm för kortaste väg.	8	1
30	Original	Köra en planerad väg.	30	28, 29
31	Original	Avbryt kartläggning om kortaste vägen hittats.	16	30
32	Original	Registrera omgivning autonomt med styrbeslut.	60	19,25
33	Original	Ställa in sig autonomt efter tejp	20	25,20
34	Original	Visa data på LCD-display.	20	1
35	Original	Ändra regler-parametrar.	20	42
36	Original	Montera processor till styrmodulen.	8	29,32,33

11.3 Aktiviteter tillhörande kommunikationsmodulen

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
37	Original	Skicka data trådlöst.	50	1
38	Original	Ta emot data trådlöst.	50	1
39	Original	Ta emot sensordata.	25	17
40	Original	Ta emot styrdata.	25	27
41	Original	Skicka styrkommandon till styrmodul.	10	37
42	Original	Skicka reglerparametrar till styrmodul.	10	37
43	Original	Hantera tappad trådlös koppling.	30	1
44	Original	Montera hårdvara.	8	18, 36
45	Original	Test med sensormodul.	30	44
46	Original	Test med styrmodul.	30	44

11.4 Aktiviteter tillhörande datormodulen

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
47	Original	Ta emot data trådlöst.	30	37
48	Original	Spara data.	20	47
49	Original	Visa data.	10	48
50	Original	Tydligt användarvänligt gränssnitt.	10	49,52,53

forts. på nästa sida

<i>forts. från föregående sida</i>				
Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
51	Original	Skicka data trådlöst.	30	37
52	Original	Rita karta.	30	48
53	Original	Visa nuvarande inställningar.	10	47
54	Original	Manuell styrning (+ gripklo).	16	51, 23
55	Original	Ställa in reglerparametrar.	8	51
56	Original	Visa nuvarande kortaste väg.	10	47,29

11.5 Övergripande aktiviteter

Aktivitet	Version	Beskrivning	Tidsåtgång	Beroende av
57	Original	Integrera alla moduler.	30	45, 46
58	Original	Testa allt.	26	57
59	Original	Slutleverans.	28	58
60	Original	Buffert.	160	-
61	Original	Statusrapportering.	10	-
62	Original	Teknisk dokumentation.	30	1
63	Original	Användarhandledning.	20	-
64	Original	Möten.	60	-
65	Original	Efterstudie.	30	59
66	Original	Kappa.	-	1

12 PRIORITERINGAR

Viktigast vid förseningar är att vara tydlig mot de andra gruppmedlemmarna att en försening skett samt ta upp detta på ett möte. På mötet ska förseningens orsaker analyseras och diskussion tas upp om något kan omprioriteras så att mer tid kan överlätas till den försenade aktiviteten. Om förseningen hotar att gå ut över någon av leveranserna [4] ska ett möte med beställaren bokas så snabbt som möjligt.

13 PROJEKTAVSLUT

Då projektets sista leverans BP5b genomförts inleds projektets efter-fas. Denna innebär att efterstudie samt kandidatrapport färdigställs, och slutligen skall även all hårdvara och eventuellt nycklar återlämnas innan projektet kan anses avslutat.

REFERENSER

- [1] “Elektronik kandidatprojekt, 16 hp (tsea56),” <https://liu.se/studieinfo/kurs/tsea56/vt-2021>, [Online; Hämtad 2021-02-04].
- [2] J. C. L. B. F. J. E. L. G. A. Alice Nicklasson Cedbro, Maja Boström, “Kravspecifikation.”
- [3] M. E. D. E. M. J. A. T. E. S. G. A. L. B. Filip Johansson, Oscar Pihl, “Ban- och tävlingsspecifikation.”
- [4] “TSEA56 Kandidatprojekt i elektronik Leveranser i projektet 2021 web site,” <http://www.isy.liu.se/edu/kurs/TSEA56/leveranser.html>, [Online; accessed February 9, 2021].
- [5] M. R. Mattias Kryssanader, “Kandidatprojektet – skrivuppgift,” föreläsning.

A GRUPPKONTRAKT

Gruppkontrakt

Bakgrund

Det huvudsakliga syftet med att upprätta ett gruppkontrakt är att alla i gruppen ska vara överens om mål och arbetssätt för det fortsatta arbetet. Utarbetandet av ett gruppkontrakt leder till diskussion och reflektion kring frågor som är viktiga för gruppens arbete och trivsel. Vid utformandet av gruppkontrakten är det viktigt att alla i gruppen deltar aktivt och att hänsyn tas till alla gruppmedlemmars synpunkter och önskemål.

Basuppgifter

Gruppens beteckning

1

Gruppens medlemmar (namn och LiU-id)

Maja Boström, majbo451
Gunnar Arctaedius, gunar889
Lovisa Byman, lovby920
Edvin Leuchovius edvle009
Johanna Carlson, johca194
Alice Nicklasson Cedbro. alice327
Filip Johansson filjo844

I. Grundläggande rutiner

- Alla medlemmar ska ha med sig schema eller kalender till varje möte, eller på något annat sätt se till att man utan problem kan bestämma nästa mötestid.
- Vid "officiella" möten ska någon anteckna vad som gjorts, vad som beslutats och när nästa möte ska vara. Denna information läggs upp i teams kanalen.
- Försöka komma i tid till möten och om förhinder uppkommer så meddelar man på något sätt.
- Boka in mer exakta tider för veckan därpå varje fredag & göra en större grovplanering.
- Ha en öppen dialog om vi ses på campus eller har möten på distans.

2. Projektets målsättning

Beskriv kortfattat hur ni uppfattar projektuppgiften. Diskutera igenom och skriv ner vilken gemensam ambitionsnivå ni har på arbetet.

Godkänt på kursen.

Fungerande slutprodukt.

Lära sig något coolt.

Ha roligt.

3. Praktiska rutiner

Hur vill ni arbeta tillsammans? Hur ofta ska ni träffas? Hur långa ska mötena vara? Finns det några särskilda förutsättningar som styr var och när ni kan ha möten, t.ex. att någon pendlar eller spelar fotboll på torsdagar?

Bestämma och planera kommande veckas arbetstimmar och mötestider i slutet av veckan.

Vid spontana möten så måste alla inte delta om man inte kan.

Vi utgår att arbeta mellan 8-17, men om det behövs mer så lägger vi det utanför de timmarna.

4. Kommunikation inom gruppen

Hur ska ni kommunicera mellan mötena?

Vi har kontakt via messenger för snabb kontakt. Vid arbetsrelaterad info som inte är bråttom att lösa så kan man lägga in det i teams och skriva en påminnelse i messenger eller t.ex. ta upp på ett möte.

5. Roller

Ska varje person i gruppen ha en formell roll? Ska ni växla formella roller vid varje möte? Vilka roller behövs (t.ex. projektledare, sekreterare, dokumentansvarig)?

Alice är projektledare.

Filip är dokumentansvarig.

En person är Sekreterare under "officiella" möten som byts inför varje möte.

Vi har fria roller i gruppen tillsvidare.

6. Beslutsformer

Hur ska gruppen komma överens om vad som ska göras? Ska majoritetsbeslut gälla eller ska gruppen genom diskussion försöka få alla att vara överens om beslutet?

Vi har kommit överens om att det är majoriteten som vinner.
Om det inte är brådkande så kan man tänka på det tills vi är överens.

7. Ansvar

Hur ska arbetet fördelas i gruppen? Hur ska gruppen hantera om någon inte gör sin del av arbetet? Vad gör ni om någon inte dyker upp på möten? Hur mycket arbete ska ni göra mellan mötena?

Arbetsfördelningen tar vi lite vecka för vecka och delar upp det jämt.
Ta upp problem tidigt så att man kan lösa det snabbt.
Om man inte kan eller inte hinner med ett möte så säger man till innan.
Ta en avstämning i slutet av varje vecka.
Lite fika kan vara trevligt att bjuda på om man kommer sent :)

Underskrifter

Vi har tillsammans kommit överens om arbetsformerna i dessa dokument.

Alice N. Cedbro

Maja Boström

Ç---Δ---(Gunnar Artaedius)

Lovisa Byman

Jobanna Carlson

Edvin Leuchorius

Filip Johansson

F DESIGNSPECIFIKATION

Designspecifikation

Alice Nicklasson Cedbro, Maja Boström, Johanna Carlson, Lovisa Byman, Filip Johansson,
Edvin Leuchovius, Gunnar Arctaedius

30 april 2021

Version 1.0



Status

Granskad	FJ, ANC	2021-03-11
Godkänd	Peter Johansson	2021-03-11

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedius		gunar889@student.liu.se

FÖRKORTNINGAR

Förkortning	Betydelse
SoC	System on Chip
I ² C	Inter-Integrated Circuit. Det är ett sätt att kommunicera över en buss.
SPI	Serial Peripheral Interface. Det är ett sätt att kommunicera över en buss.
PD-reglering	En regleringsmetod som använder sig av proportionell och deriverande reglering.
JSP	Jackson Structured Programming. En metod för att strukturera upp program.
IR-sensor	Optisk avståndsmätare

TABELLER

Tabell 1	Data mellan modulerna	4
Tabell 2	Komponentbudget och placering av sensorerna	9
Tabell 3	Prestanda för ATPmega1284P	10
Tabell 4	Prestanda för Lidar Lite v3	10
Tabell 5	Prestanda för Gyroskop-L3GD20	10
Tabell 6	Prestanda för Avståndsmätare-GP2D120	10
Tabell 7	Betydelse av knappar.	11
Tabell 8	Komponentbudget för styrmodulen	12
Tabell 9	Alla avbrott som hanteras, varifrån de kommer samt om det är ett avbrott som hanteras i det manuella eller autonoma läget.	13
Tabell 10	Styrmoderna för roboten i det autonoma läget samt metod för reglering. I sista kolumnen syns även vilket fel som kommer användas som insignal till regulatorn.	13
Tabell 11	Komponentbudget kommunikationsmodul	18
Tabell 12	Datan som läses in vid respektive avbrott, vilken modul datan kommer ifrån och till vilken modul datan ska skickas vidare.	19

FIGURER

Figur 1	Roboten i dess omgivning.	1
Figur 2	En översikt av systemet.	2
Figur 3	Blockschema som visar modulernas dataförbindelser.	3
Figur 4	Datagram över paket som skickas mellan moduler.	4
Figur 5	Kopplingschema för sensormodulen.	7
Figur 6	Övergripande beskrivning av hur sensormodulens komponenter ska placeras på roboten.	8
Figur 7	JSP-diagram för sensormodulen.	8
Figur 8	Kopplingsschema för styrmodulen.	12
Figur 9	JSP-diagram över huvudloopen i det autonoma läget av körningen. Med fixering av sensor-datan menas att sensordatans nuvarande värde sparas för att samma värde ska användas i hela loopen. Utan detta steg finns risk för att ett avbrott uppdaterar sensordatans värde mitt i loopen.	14
Figur 10	De olika faserna i programmet. Alla faser innehåller stoppvilkor som beskriver när nästa fas ska påbörjas.	14
Figur 11	JSP-diagram över planeringsdelen i fasen då roboten letar efter den nödställde.	15
Figur 12	Kartan representeras som en matris där varje element i matrisen motsvarar en modul i labyrинten. Varje element får ett värde mellan 0 och 15 baserat på var motsvarande modul har väggar.	16
Figur 13	De pinnar på raspberryen som behövs för busskommunikation och med en nivåskiftnare som omvandlar 3.3 V till 5 V.	18
Figur 14	Principskiss av huvudprogrammet och avbrottsrutinerna.	19
Figur 15	En visualisering av hur användargränssnittet kommer att se ut.	21

INNEHÅLL

1	Inledning	1
2	Översikt av systemet	1
3	Kommunikation mellan delsystem	3
4	Sensormodul	7
4.1	Mjukvara	8
4.2	Hårdvara	9
5	Styrmodul	11
5.1	Hårdvara	12
5.2	Kopplingsschema	12
5.3	Mjukvara	13
5.4	Data	15
6	Kommunikationsmodul	17
6.1	Hårdvara	17
6.2	Kopplingsschema	18
6.3	Mjukvara	18
7	Datormodul	20
7.1	Mjukvara	20
8	Implementation	22
	Referenser	24

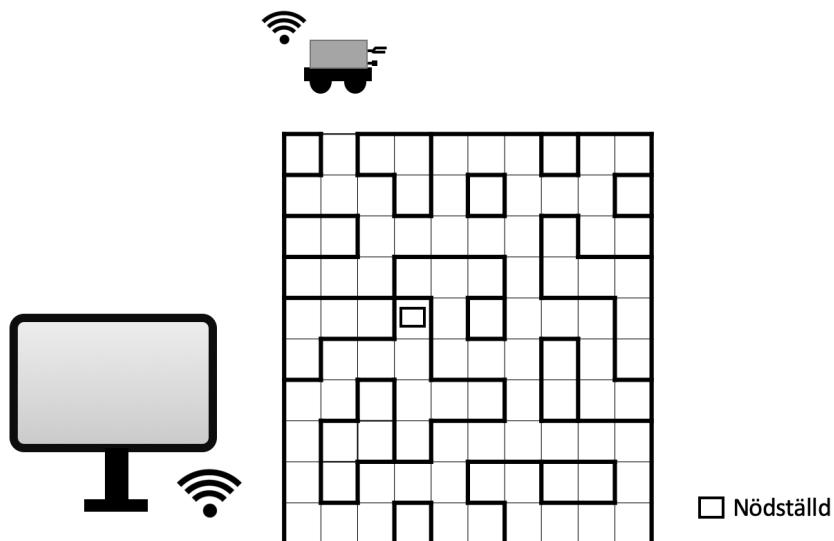
DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-03-01	Första utkast	Alla	Alla
0.2	2021-03-09	Andra utkast	Alla	Alla
1.0	2021-03-11	Tredje utkast	Alla	FJ, ANC

1 INLEDNING

I kursen TSEA56, Kandidatprojekt i elektronik, given vid Linköpings universitet våren 2021 ska en autonom undsättningsrobot konstrueras. Denna designspecifikation listar ingående beskrivningar av de delsystem undsättningsroboten ska bestå av, och för varje delsystem anges kopplingsschema samt vilken hårdvara respektive mjukvara delsystemen behöver. Vidare presenteras hur delsystemen ska kommunicera med varandra, samt en implementationsstrategi.

Figur 1 visar systemet i den omgivning roboten designas till. Labyrinten som är uppbyggd av moduler och maximalt är 25x25 moduler stor, innehåller en öppning tillika startposition samt en nödställd placerad i labyrinten (markerad med fyrkant). [1] [2] Målet med projektet är att undsättningsroboten autonomt ska kunna köra in i labyrinten, hitta kortaste vägen till de nödställda, åka tillbaka till startpositionen för att hämta en förnödenhet samt leverera denna till de nödställda på kortast möjliga tid under en tävling mellan olika konstruktioner. Vid vinst kommer roboten levereras till beställare Kent Palmkvist som därefter får ansvaret över roboten.

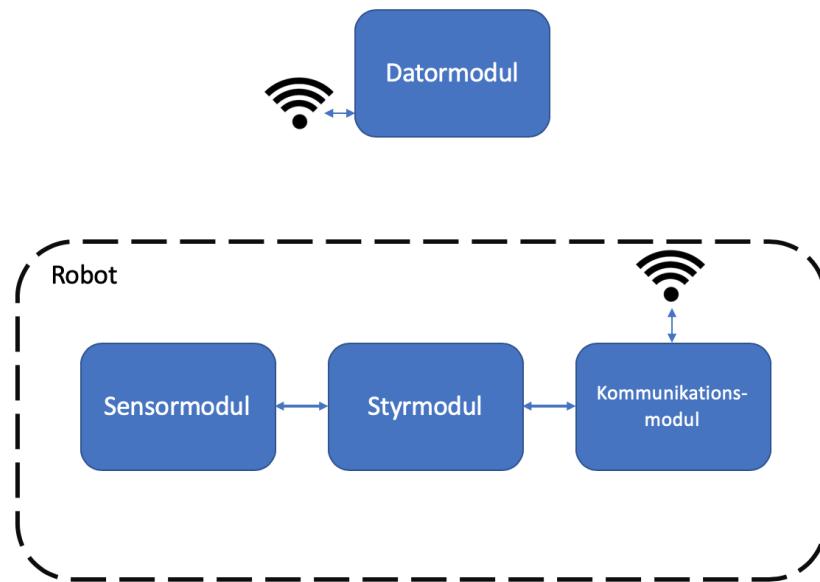


Figur 1: Roboten i dess omgivning.

2 ÖVERSIKT AV SYSTEMET

Detta avsnitt beskriver systemet och dess ingående moduler översiktligt utifrån de övergripande målen med produkten som utvecklas, det vill säga att konstruera en undsättningsrobot som kan utföra sitt uppdrag enligt projektets kravspecifikation [2].

De ingående modulerna är: en sensormodul, en styrmodul, en kommunikationsmodul, samt en datormodul. Figur 2 visar en principskiss av de ingående modulerna. En mera precis specifikation av modulernas interaktioner med varandra inleds i avsnitt 3.



Figur 2: En översikt av systemet.

Sensormodulen består av 9 st komponenter. Fyra stycken IR-sensorer, en Lidar Lite, en reflexsensormodul, ett gyroskop, en tryckknapp och en brytare, se tabell 2. Lidar-sensorn ska placeras på toppen av roboten riktad framåt. Två av IR-sensorerna ska placeras på höger sida av roboten och två av IR-sensorerna kommer att placeras på vänster sida. På respektive sida kommer den ena sensorn placeras långt fram och den andra kommer att vara längre bak. Reflexsensormodulen ska placeras på framsidan av roboten riktad nedåt mot marken. Gyroskopet, knappen och brytaren ska placeras på toppen av roboten, enligt figur 6.

Styrmodulen kommer att hantera robotens alla rörelser; med hjälp av styrkommandon från datormodulen i manuellt läge. I det autonoma läget är styrmodulen själv ansvarig för att beräkna styrbeslut utifrån sensordata från sensormodulen, rita karta och bestämma kortaste väg mellan olika moduler i labyrinten utifrån kartdata.

Kommunikationsmodulen kommer att hantera överföringen av data mellan styr- respektive sensormodulen på roboten och datormodulen.

Datormodulen har ett grafiskt användargränssnitt som används för att trådlöst styra roboten i manuellt läge och för övervakning i autonomt läge. I autonomt läge ska styrbesluten som skickas från styrmodulen på roboten tas emot och sparas. Det samma gäller sensordata i både manuellt och autonomt läge.

3 KOMMUNIKATION MELLAN DELSYSTEM

Kommunikationen mellan delsystemen, samt sensorer och knappar i sensormodulen kommer att sammankopplas via både I²C och SPI. Se figur 3.

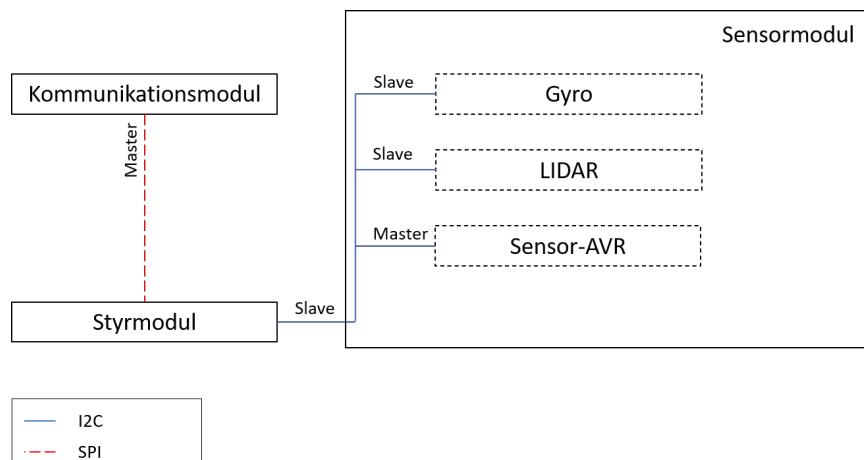
Mikrokontrollern som tillhör sensormodulen har alltså master-behörighet på I²C-bussen. Internt inom sensormodulen är två sensorer anslutna på denna buss som slavenheter; en Lidar för avståndsmätning och ett gyro för mätning av vinkelhastighet. Mikrokontrollern hämtar data från dessa sensorer över bussen för bearbetning.

Från ett externt perspektiv är styrmodulen ansluten som en slavenhet på I²C-bussen. Således är det sensormodulen som styr dataflödet mellan styrmodulen och sensormodulen, där sensormodulen uteslutande kommer att skicka data till styrmodulen.

Utöver I²C-bussen kommer kommunikationsmodulen och styrmodulen vara sammankopplade med en SPI-anslutning för direkt utbyte av styrkommandon och styrbeslut mellan dessa (beroende på om robotten körs i manuellt eller autonomt läge). Dessutom kommer styrmodulen att skicka vidare den sensordata, som ursprungligen levererades av sensormodulen, till kommunikationsmodulen.

Kommunikationsmodulen kommer att vara master på SPI-bussen, vilken kommer förse styrmodulen med styrkommandon i manuellt läge. I autonomt läge kommer den istället att hämta styrbeslut och sensordata.

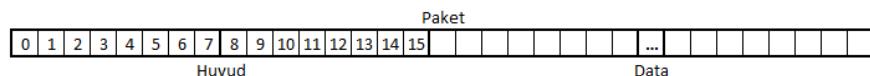
Det är kritiskt att de avbrottrutiner som genereras i styrmodulen av I²C- respektive SPI-bussen inte sker så pass frekvent att det påverkar styrmodulens förmåga att styra robotten obehindrat.



Figur 3: Blockschema som visar modulernas dataförbindelser.

Data som kommer att skickas mellan de olika modulerna finns listad i tabell ???. Även detaljer om formen på datan samt förklaring av värdenas betydelse återfinns i tabellen.

Varje datatyp kommer att skickas som paket, där varje paket består av ett pakethuvud om 16 bitar (se figur 4), följt av ett eller flera byte med data. Varje datatyp har ett specifikt 8 bitar långt ID-nummer placerad i pakethuvudets första byte (bit 0-7). När ett paket mottagits av någon modull kommer datans integritet att kunna kontrolleras genom en checksum-metod. Bit 8-15 i pakethuvudet är reserverat för detta ändamål.



Figur 4: Datagram över paket som skickas mellan moduler.

Tabell 1: Data mellan modulerna.

ID	Förklaring	Sändande modul	Mottagande modul	Datatyp	Tillåtna värden	Förklaring av värden
FD	Totalt antal byte data som ska skickas från en modul till en annan över SPI.	Styr och komm.	Styr och komm.	16-bit	0 till 65535	
FE	Stoppvillkor för dataöverföringar över SPI.	Styr och komm.	Styr och komm.	0-bit		Ingen data medföljer.
FF	Startvillkor för dataöverföring över SPI.	Styr och komm.	Styr och komm.	0-bit		Ingen data medföljer.
1	Avvikelse från mitten av tejp	Sensor	Styr och komm.	8-bit 2 kompl.	-5 till +5	+ betyder avvikelse till höger och - betyder avvikelse till vänster
2	Avstånd till höger vägg fram	Sensor	Styr och komm.		3 till 40	Avståndet i centimeter.
3	Avstånd till vänster vägg fram	Sensor	Styr och komm.		3 till 40	Avstånd i centimeter.
4	Avstånd till höger vägg bak	Sensor	Styr och komm.		3 till 40	Avståndet i centimeter.
5	Avstånd till vänster vägg bak	Sensor	Styr och komm.		3 till 40	Avstånd i centimeter.
6	Avstånd till vägg fram	Sensor	Styr och komm.	16-bit	0 till 4000	Avstånd i centimeter.
7	Vinkel (från gyro)	Sensor	Styr och komm.	16-bit	0 till 360	Mäts i grader där graderna ökar medurs.
<i>forts. på nästa sida</i>						

forts. från föregående sida						
ID	Förklaring	Sändande modul	Mottagande modul	Datatyp	Tillåtna värden	Förklaring av värden
8	Vinkelhastighet (från gyro)	Sensor	Styr och komm.			Mäts i grader. 0-360, där graderna ökar medurs.
9	Autonomt/manuellt läge på brytare	Sensor	Styr och komm.	8-bit	0,1 eller 2	Skickas ihop med data från startknapp.
9	Startknapp	Sensor	Styr och komm.	8-bit	0,1 eller 2	0 = manuellt läge, 1 = autonomt läge, ej startad, 2 = autonomt läge, startad
10	Reglerparametrar, K_p korridorskörning	Komm.	Styr	6 st. 16-bit		
ny	Reglerparametrar, K_a korridorskörning	Komm.	Styr	16-bit		
ny	Reglerparametrar, K_p rotera	Komm.	Styr	16-bit		
ny	Reglerparametrar, K_d rotera	Komm.	Styr	16-bit		
ny	Reglerparametrar, K_p , tejpföljning	Komm.	Styr	16-bit		
ny	Reglerparametrar, K_d/K_a , tejpföljning	Komm.	Styr	16-bit		
11	Styrkommandon	Komm.	Styr	8-bit	0 till 5	0 = kör fram, 1 = kör bak, 2 = rotera höger, 3 = rotera vänster, 4 = stäng gripklo, 5 = öppna gripklo
12	Gaspådrag höger hjulpar	Styr	Kommunikation	8-bit 2 kompl.	-100 till 100	Procent av maximal spänning. - för spänning bakåt, + för spänning framåt

forts. på nästa sida

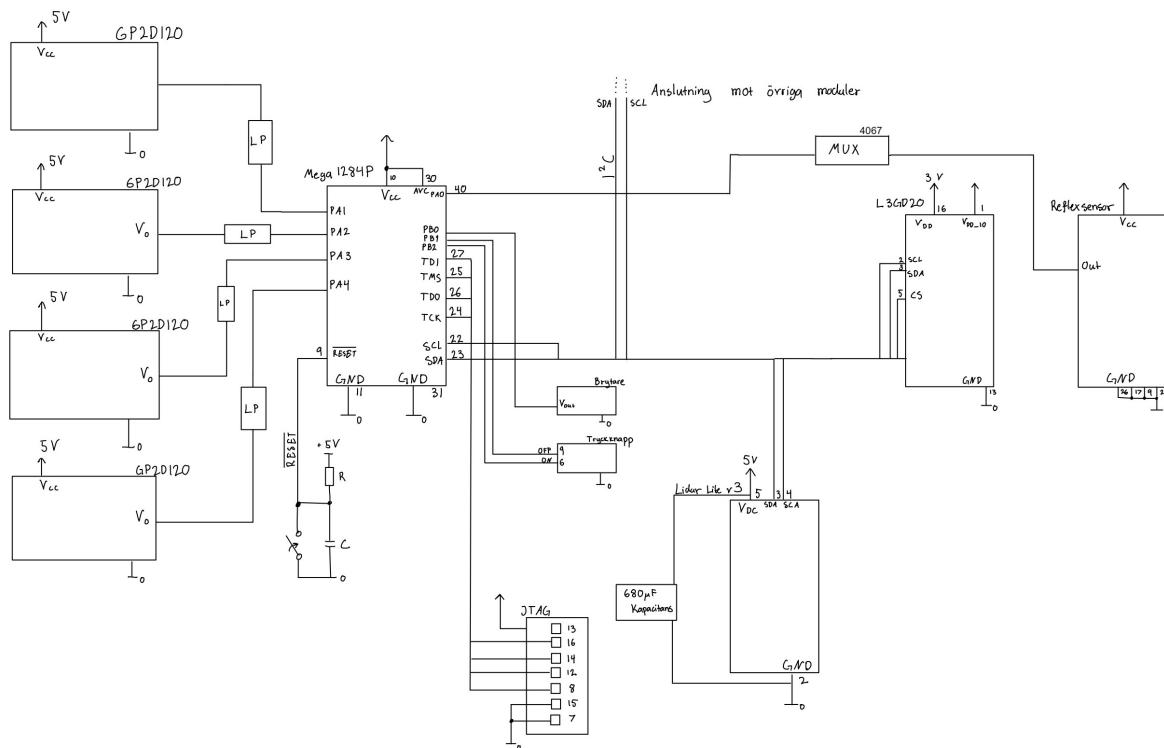
forts. från föregående sida						
ID	Förklaring	Sändande modul	Mottagande modul	Datatyp	Tillåtna värden	Förklaring av värden
13	Gaspådrag vänster hjulpar	Styr	Kommunikation	8-bit 2 kompl.	-100 till 100	Procent av maximal spänning. - för spänning bakåt, + för spänning framåt
14	Styrbeslut	Styr	Kommunikation	Vektor med 8-bit		Varje värde är ett beslut
15	Nuvarande kortaste väg	Styr	Kommunikation	Vektor med 8-bit		
16	Karta, infon här stämmer inte helt längre	Styr	Kommunikation	Array med 8-bit	0 till 15	Väggar för varje modul
17	Placering av nödställd, infon här stämmer inte helt längre	Styr	Kommunikation	16-bit	0 till 625	Värde utifrån vilken modul den befinner sig i
18	Placering av robot, infon här stämmer inte helt längre	Styr	Kommunikation	16-bit	0 till 625	Värde utifrån vilken modul den befinner sig i
19	Riktning på robot	Styr	Kommunikation	8-bit	0 till 3	
20	Placering av ingång, infon här stämmer inte helt längre	Styr	Kommunikation	8-bit	0 till 25	Den kartruta längs väggen som ingången finns i.
FD	Totalt antal byte data som ska skickas från en modul till en annan över SPI.	Styr och komm.	Styr och komm.	16-bit	0 till 65535	
FE	Stoppvillkor för dataöverföringar över SPI.	Styr och komm.	Styr och komm.	0-bit		Ingen data medföljer.
FF	Startvillkor för dataöverföring över SPI.	Styr och komm.	Styr och komm.	0-bit		Ingen data medföljer.

4 SENSORMODUL

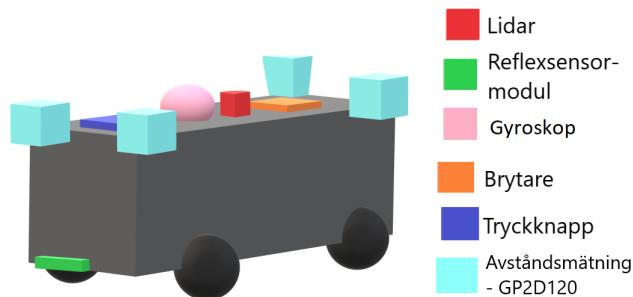
Sensormodulens ansvar är att samla in data om robotens omgivning, position samt dess styrningsvinkel för att sedan processera denna och skicka vidare till styrmodulen. Modulen har även ansvar för en tryckknapp som startar körningen samt en brytare som väljer manuell eller autonom styrning.

Processorn ATmega1284P kommer vara kopplad till sensorerna, knapparna samt styrmodulen enligt Figur 5. Vissa av sensorerna (lidar och gyroskop) samt styrmodulen kommer vara inkopplade till processorn via en I²C buss. All mottagning respektive sändning av data kommer hanteras i avbrottssutioner, se 4.1.

Det finns fyra olika sensor typer i modulen: en lidar för avståndsberäkning, fyra stycken IR-sensorer för avståndsberäkning, ett gyroskop för beräkning av styrningsvinkel samt en reflexsenor för avläsning av markeringar i marken, se Figur 6. Gyroskopet samt lidarn kommer placeras på robotens ovansida. IR-sensorerna kommer placeras två och två på robotens högra respektive vänstra sida. På vardera sida placeras ena sensorn mot robotens främre del och andra mot den bakre delen. Reflexsensorn placeras på robotens undersida mot framkanten, eventuellt lite framskjuten, för att så snabbt som möjligt kunna upptäcka tejmarkeringar i golvet. Tryckknapp samt brytare placeras i första hand på ovansidan för att enkelt kunna nås men är inte lika beroende av placering.



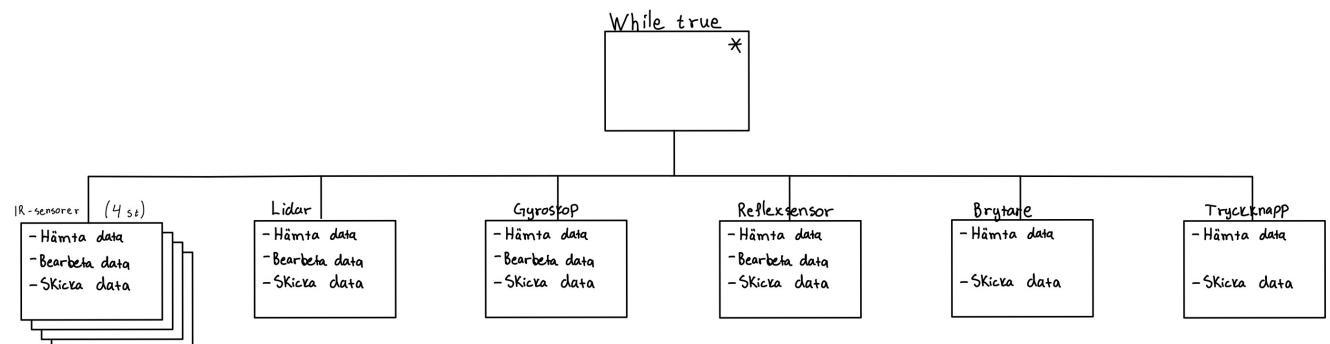
Figur 5: Kopplingsschema för sensormodulen.



Figur 6: Övergripande beskrivning av hur sensormodulens komponenter ska placeras på roboten.

4.1 Mjukvara

Sensormodulens mjukvara kommer bestå av en inledningsdel, en huvudloop och flera avbrottsrutiner. I inledningsdelen kommer alla avbrottsrutiner att initieras samt en intern klocka i processorna att konfigureras. Mer information kring avbrottsrutinerna och huvud-loopen följer nedan, samt kan avläsas i figur 7.



Figur 7: JSP-diagram för sensormodulen.

4.1.1 *Avbrott*

Det finns en avbrottsrutin och en tillhörande flagga för varje enskild sensor och knapp. Alla avbrott startas regelbundet med hjälp av en inbyggd klocka i processorn och kommer då hämta data till processorn. Processorn bearbetar datan samt skickar den vidare. När en avbrottsrutin är klar går programmet tillbaka till huvudloopen. Pågående avbrott kommer inte att avbrytas när ett nytt avbrott anropas, utan nya avbrott kommer att memoreras och läggas på kö via flaggor.

4.1.2 *Huvudloop*

Huvudloopen kommer endast bestå av en while-loop som alltid är sann. I loopen körs ett "wait for interrupt" för att spara på processorn.

4.2 Hårdvara

Det behövs fyra olika sensortyper (totalt 7 st sensorer) och två knappar i sensormodulen, se tabell 2.

Tabell 2: Komponentbudget och placering av sensorerna

Komponenter	Användningsområde	Antal [st]
Lidar Lite v3[3]	Avståndsmätning	1
Reflexsensormodul[4]	Tejpsökning	1
Gyroskop[5]	Mätning av vinkelhastighet	1
Tryckknapp [6]	Startar körföringen	1
Brytare	Väljer manuell eller autonom körföring	1
GP2D120[7]	Avståndsmätning	4

4.2.1 *Microprocessor*

Microprocessorn ATmega1284P [8] kommer användas i sensormodulen. Dess specifikationer, vilka listas i tabell 3, tillgodosser behoven för sensormodulen väl. ATmega1284P kan lagra 128 kB i flashminnet och utföra 20 miljoner instruktioner per sekund. Processorn kan hantera uppdateringsfrekvensen och lagringen som krävs från alla sensorer. Mest krävande sensorn är lidarn, avståndssensorn, som uppdateras maximalt 1000 gånger per sekund och sparar den uppmätta sträckan i 16 bitar vilket processorn kan hantera. Data kommer inte sparas tillfälligt utan all data kommer kunna hanteras direkt i processorn. Mikroprocessorns portar, 32 stycken, räcker möjligtvis inte för inkoppling av alla komponenter som sensormodulen består av, se figur 5. Beroende på antalet lysdioder som används i reflexsensormodulen kommer det eventuellt bli brist på portar med A/D omvandling. I sådana fall kommer en multiplexer kopplas mellan reflexsensormodulen och processorn. I figur 5 går även att avläsa till vilka sensorkomponenter pinnarna på processorn ska kopplas. Processorn kommer styra när data från sensorer hämtas.

Tabell 3: Prestanda för ATPmega1284P

Specifikation	Mått
Flashminne	128kB
I/O pinnar	32(40)
Klocka	20MHz

4.2.2 Lidar Lite v3

Sensorn Lidar Lite v3 har prestanda enligt tabell 4.

Tabell 4: Prestanda för Lidar Lite v3

Specifikation	Mått
Räckvidd	40m
Noggrannhet < 5m	$\pm 1\text{cm}$
Noggrannhet $\geq 5\text{m}$	$\pm 10\text{cm}$
Uppdateringshastighet	270Hz (Typiskt värde)
Kraft	5 Vdc (Typiskt värde)
Användargränssnitt	I ² C

4.2.3 L3GD20

Sensorn L3GD20 har prestanda enligt tabell 5.

Tabell 5: Prestanda för Gyroskop-L3GD20

Specifikation	Mått
Matningsspänning	2.4V till 3.6V
Användargränssnitt	I ² C/SPI
Utdata	16-bitar

4.2.4 GP2D120

IR-sensor [7] som kan detektera avstånd mellan 4 till 30 cm. Utsignalen för denna sensor är en analog signal.

Tabell 6: Prestanda för Avståndsmätare-GP2D120

Specifikation	Mått
Matningsspänning	4.5V till 5.5V
Avståndsmätning	4cm till 30cm

4.2.5 Knappar

Status för startknappen och brytaren för manuellt/autonoma läge kommer att skickas från sensormodulen tillsammans som ett 8-bitars tal, eftersom data som skickas över I²C-bussen alltid skickas 8 bitar i taget. LSB-biten kommer att indikera att roboten befinner sig i antingen manuellt eller autonomt läge, medan biten efter indikerar om startknappen har tryckts ner eller inte. Tabell 7 visar vad motsvarande värde på denna byte indikerar i decimalform.

Tabell 7: Betydelse av knappar.

Värde	Betydelse
0	Robot i manuellt läge.
1	Robot i autonomt läge, men väntar på startsignal.
2	Robot i autonomt läge med startsignal angiven.

5 STYRMODUL

Styrmodulen har som uppgift att styra roboten och ta beslut i det autonoma läget. Detta kommer att ske med en ATmega1284P som är kopplad till motorerna på roboten, gripklon och kommunikationmodulen samt sensormodulen.

I det manuella läget tar mikroprocessorn emot styrdata från användaren via datormoden och kommunikationsmodulen. De skickade kommandot tolkas i processorn som i sin tur bestämmer spänningen på motorerna för att utföra den begärda rörelsen. I det manuella läget kommer ingen reglering att utföras. De kommandon som hanteras är framåt, bakåt, rotera vänster och rotera höger, samt öppning och stängning av gripklon. Dessa implementeras genom väldigt enkla funktioner, som går ut på att antingen ge motorerna samma gaspådrag för att åka framåt eller bakåt, eller motstå gaspådrag i fallet att roboten ska svänga. Ingen data från sensorerna hanteras i detta läge, men kommer att skickas vidare till kommunikationsmodulen över SPI på begäran.

Det autonoma läget är mer avancerat och är det läge roboten kommer befina sig i under tävlingen. Här ska processorn ta in data från sensormodulen som den behandlar för att både ta beslut om framtida körning och reglera den autonoma körningen. Det är styrmodulens uppgift att bestämma bana och styra roboten genom labyrinten för att utföra uppdraget specificerat i ”Ban- och tävlingsspecifikationen” [1]. Processorn skickar även data till kommunikationsmodulen om: styrbeslutet som tas, gaspådrag som sätts på motorerna och gripklon, sensordata, kartan, uppskattad position i labyrinten samt planerad körväg. Modulen tar även emot data från kommunikationsmodulen om förändring av reglerparametrarna.

5.1 Hårdvara

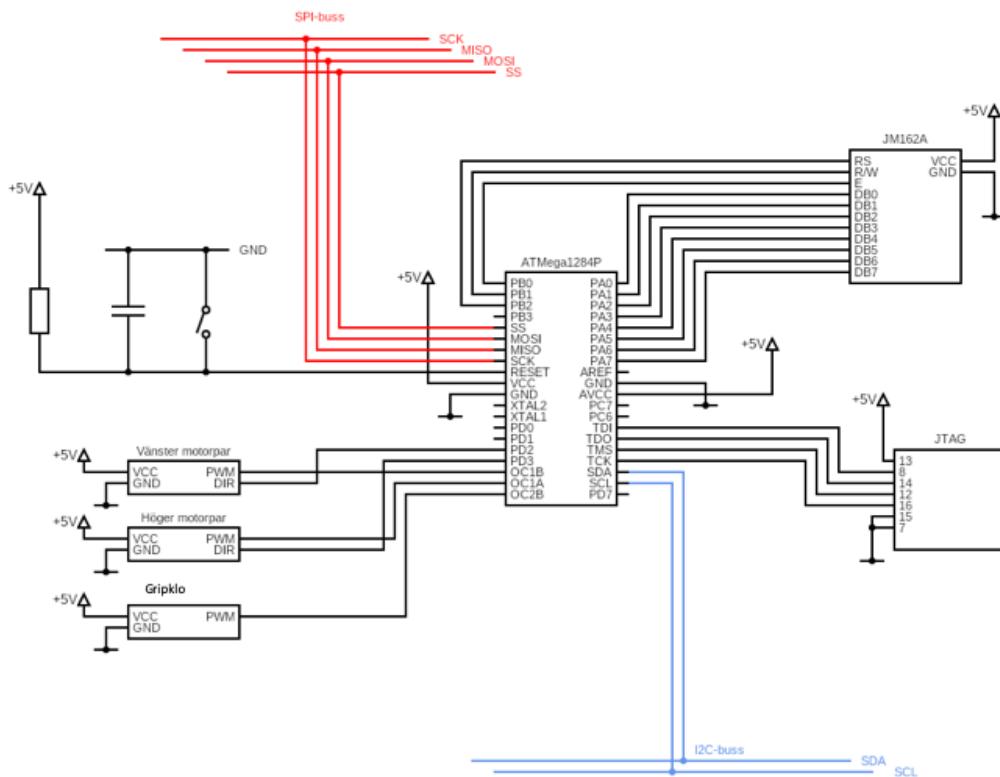
Hårdvaran för styrmodulen är listad i tabell 8.

Tabell 8: Komponentbudget för styrmodulen

Komponenter	Användningsområde	Antal [st]
Terminator	Chassi inklusive motorer, batteri och gripklo	1
ATmega1284P	Processor	1
JM162A	LCD-display	1

5.2 Kopplingsschema

Kopplingsschemat för styrmodulen kan ses i figur 8. Utifrån kopplingsschemat kan det konstateras att processorns portar räcker till.



Figur 8: Kopplingsschema för styrmodulen.

5.3 Mjukvara

Olika program körs beroende på om roboten är i autonomt eller manuellt läge. Båda programmen kommer till stor del styras av avbrott från de andra modulerna. I autonoma läget kommer ett huvudprogram att köras som blir avbrutet av avbrotten, men i det manuella läget väntar programmet endast på avbrott. De avbrott som hanteras beskrivs i tabell 9.

Tabell 9: Alla avbrott som hanteras, varifrån de kommer samt om det är ett avbrott som hanteras i det manuella eller autonoma läget.

Avbrott	Från	Manuellt/Autonomt
Styrkommandon	Kommunikationsmodul	Manuellt
Begäran om data	Kommunikationsmodul	Båda
Ändring av reglerparametrar	Kommunikationsmodul	Autonomt
Sensordata	Sensormodul	Autonomt

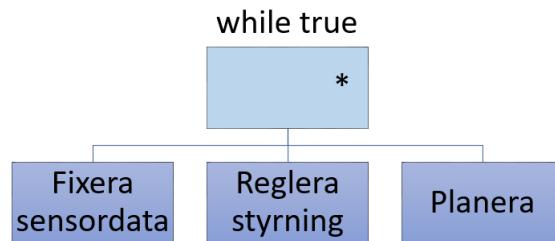
I det manuella läget körs ett program som väntar på att styrkommandon skickas från kommunikationsmodulen. När ett kommando tas emot sätts spänningen på hjulen olika beroende på vilket kommando som skickades. Skickas till exempel ett kommando för att rotera åt vänster sätts det högra hjulet till att köra fram och det vänstra hjulet till att köra bakåt.

I det autonoma läget kör roboten i olika styrmoder, som regleras olika. I tabell 10 syns de olika styrmoderna, och i fallet att en styrmod regleras visas även metod för reglering samt vilket mätvärde som används vid regleringen. Som syns i tabellen regleras alla styrmoder med en diskret form av PD-reglering. Koden kommer även vara väldigt lik i alla tre fall. Reglerparametrarna kan ändras via datorn, och justeras tills de anses uppfylla kraven tillräckligt.

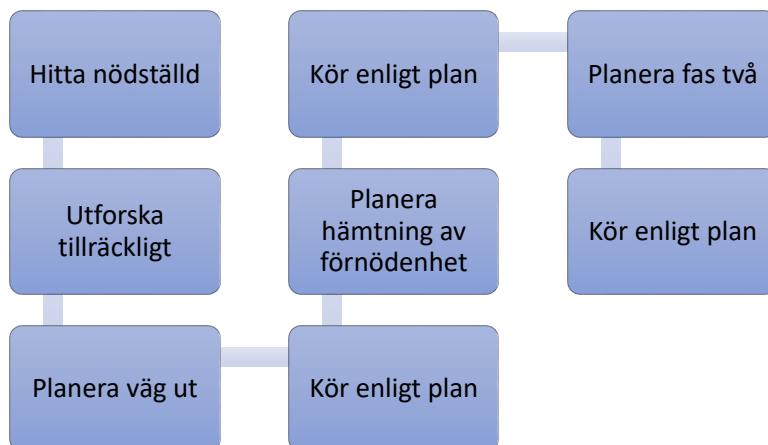
Tabell 10: Styrmoderna för roboten i det autonoma läget samt metod för reglering. I sista kolumnen syns även vilket fel som kommer användas som insignal till regulatorn.

Styrmoder	Metod för reglering	Fel
Köra rakt längst korridor	PD-reglering	Skillnad i avstånd till väggar
Rotera 90 grader runt mittaxeln	PD-reglering	Vinkel till ny riktning
Följa en tejpremsa	PD-reglering	Avstånd till mitten av tejpremsa
Greppa föremål med gripklo	Ingen	Inget

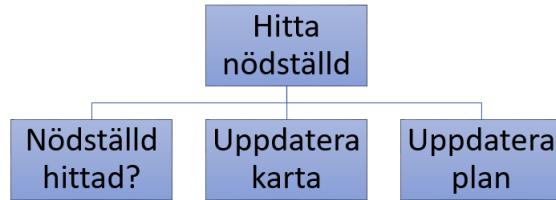
Figur 9 representerar huvudloopen i det autonoma läget. Först fixeras sensordatan för att förhindra att datan förändras av ett avbrott mitt i loopen, och sedan sköts reglering av körning och planering. Regleringen sköts olika i de olika styrmoderna som tidigare beskrivet. Programmet går igenom olika faser när roboten utför uppdraget, och de olika faserna förändrar vad som sker i planeringsdelen. Planeringsdels uppgift är att bestämma en plan kring framtida körning och bestämma nuvarande och kommande styrmoder. Figur 10 visar de olika faserna och i vilken ordning de körs. Fasen benämnd 'Planera fas två' skapar en plan för att köra kortaste vägen till de nödställda, lämna förnödenheten, och sedan köra kortaste vägen ut ur labyrinten.



Figur 9: JSP-diagram över huvudloopen i det autonoma läget av körningen. Med fixering av sensordatans nuvarande värde sparas för att samma värde ska användas i hela loopen. Utan detta steg finns risk för att ett avbrott uppdaterar sensordatans värde mitt i loopen.



Figur 10: De olika faserna i programmet. Alla faser innehåller stoppvilkor som beskriver när nästa fas ska påbörjas.



Figur 11: JSP-diagram över planeringsdelen i fasen då roboten letar efter den nödställda.

I figur 11 visas ett JSP-diagram över planeringen i en av faserna, den då roboten letar efter de nödställda i labyrinten. I första steget analyseras sensordatan för att undersöka om den nödställda har blivit hittad. Sedan analyseras sensordatan i syfte att uppdatera den inre representationen av kartan. Därefter uppdateras planen baserat på utfallet i de två tidigare stegen. I detta steg kan både nuvarande och kommande styrmoder komma att ändras, och fasen kan bytas. Alla olika faser har olika planeringsfunktioner som fungerar på olika vis. Något gemensamt är dock att alla kommer att använda sensordatan för att uppdatera planen, det vill säga ändra framtida styrmoder och fas.

5.4 Data

Styrmodulen sparar och behandlar olika data, både för att kunna utföra uppdraget och skicka datan till de andra modulerna. Processorn har ett dataminne på 16kB vilket, som kan ses i nedanstående rubriker, räcker.

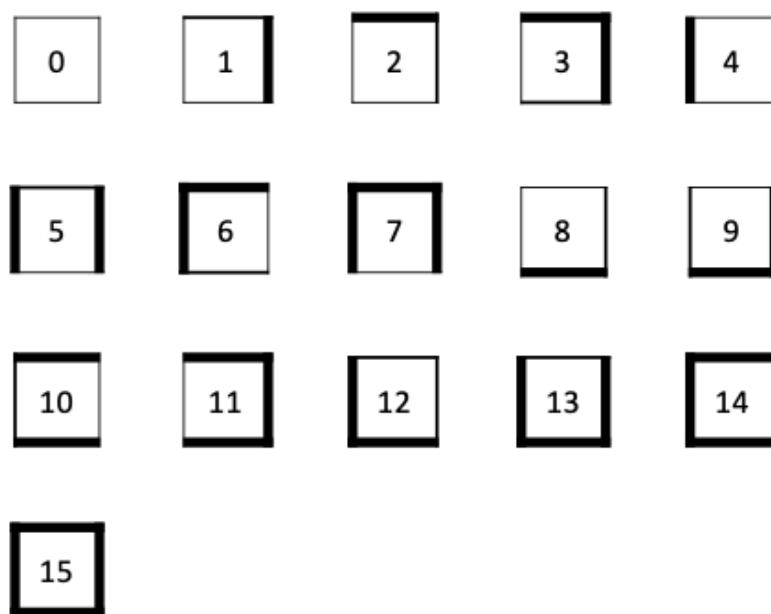
5.4.1 Karta

Kartan kommer att representeras som en matris där varje element i matrisen kommer representera en modul i rutnätet som labyrinten är uppbyggd av. Varje element i matrisen kommer att tilldelas ett värde beroende på vilken kombination av väggar modulen har enligt figur 12. Det innebär att representationen av kartan kommer kräva en matris med 25×25 element, där varje element består av ett 4 bitar stort tal. Det innebär att kartan kommer att kräva $25 \cdot 25 \cdot 4 = 2500$ bitar.

För att kunna placera ut roboten på rätt plats i labyrinten kommer den att representeras med ett tal beroende på vilken modul den befinner sig i, alltså ett tal mellan 0 och 625, där 0 innebär att roboten är utanför labyrinten och 1-625 representerar alla modulerna. Robotens riktning kommer att representeras som ett tal mellan 0-3 beroende på vilket väderstreck roboten rör sig i.

Den nödställdas placering kommer att representeras på samma sätt som robotens placering men eftersom den inte kan finnas utanför labyrinten så kommer den att representeras med ett värde mellan 1 och 625.

På liknande sätt som för roboten och den nödställdas placering kommer ingången till labyrinten att representeras. Den kommer att representeras med ett tal mellan 1 och 25 eftersom ingången bara kan vara längs ytterkanterna och eftersom robotens initiala riktning alltid kan väljas till uppåt.



Figur 12: Kartan representeras som en matris där varje element i matrisen motsvarar en modul i labyrinten. Varje element får ett värde mellan 0 och 15 baserat på var motsvarande modul har väggar.

5.4.2 Styrbeslut

Då roboten ska göra upp en plan för att sedan följa den måste robotens planerade körsättning sparades. Den planerade körsättningen, alltså robotens styrbeslut, kommer att sparas i en vektor. Varje möjligt styrbeslut, alltså styrmoderna listades i 10, representeras av en siffra och sparas i vektorn i ordningen beslutet ska utföras i. Antalet planerade styrbeslut kommer inte att överstiga 1000 stycken, och det finns inte mer än 10 olika styrmoder som då kan representeras med 4 bitar, vilket innebär att vektorn med styrbeslut aldrig kommer vara större än $1000 \cdot 4 = 4000$ bitar.

5.4.3 Gaspådrag

Gaspådraget kommer inom styrmodulen att bestå av två värden per hjulpar. Ett värde, antingen 0 eller 1, som anger rotationsriktningen för hjulparet och ett värde mellan 0 och den maximalt tillåtna spänningen som anger spänningen på hjulparet. Innan gaspådraget ska skickas till kommunikationsmodulen kommer gaspådraget att räknas om till ett tal mellan -100 och 100, där tecknet motsvarar riktningen på hjulet och värdet motsvarar procent av maximala spänningen.

6 KOMMUNIKATIONSMODUL

Kommunikationsmodulen utgör ett gränssnitt mellan datormodulen och övriga moduler på roboten för dataöverföring mellan dessa.

Modulen kan kommunicera med styrmodulen direkt via en SPI-anslutning, där kommunikationsmodulen är master. Den kommer därmed kunna skicka styrkommandon till styrmodulen, vilket är aktuellt när roboten används i manuellt läge. I autonomt läge kommer kommunikationsmodulen istället att hämta styrbeslut, kartdata och sensordata från styrmodulen, vilka skickas vidare till datormodulen. I autonomt läge kommer modulen även kunna ta emot reglerparametrar från datormodulen och skicka vidare till styrmodulen över SPI.

Kommunikationen med datormodulen sker med en Bluetooth-anslutning, vilken Raspberry Pi har inbyggt stöd för. Denna anslutning kommer att konfigureras på Raspberry Pi som en serieport.

6.1 Hårdvara

Hårdvarumässigt kommer modulen bestå av en Raspberry Pi 3 B+. Den strömförsörjs med 5.1V matningsspänning och en rekommenderad ström på 2.5A, via en microUSB-kabel.

SoC:en på Raspberry Pi har 4 stycken timers å 32-bitar (se avsnitt "12 System Timer" i [9]), vilka kan användas för att generera avbrott. En av dessa kommer att användas för kommunikationsmodulen och beskrivs i avsnitt 6.3.

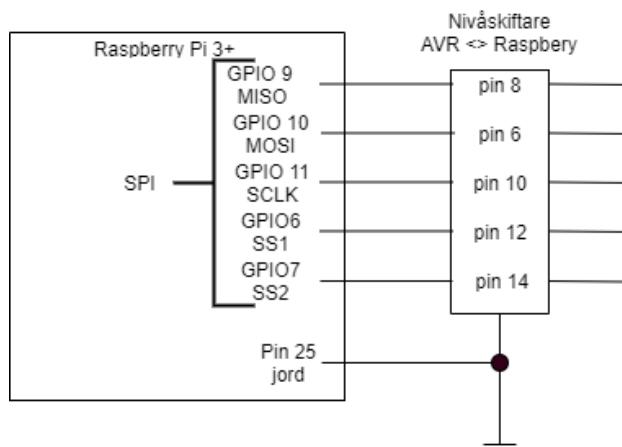
Raspberry Pi behöver anslutas till styrmodulen via SPI. GPIO-pinnarna på Raspberry Pi använder 3.3V spänning, medan AVR-kontrollerna använder 5V. För anslutning till Raspberry Pi behövs därför en nivåskiftare för de pinnar på Raspberryn's 40-pinnars kontakt som har SPI-funktion. En sådan nivåskiftare, anpassad för detta ändamål, finns att tillgå av ISY [10]. Då det inte finns någon reset-pin till Raspberry Pi 3 B+ behövs ingen resetkrets.

Tabell 11: Komponentbudget kommunikationsmodul

Komponenter	Användningsområde	Antal [st]
Raspberry Pi 3 B+ [11]	Bluetooth och processor	1
Nivåskiftningschip Raspberry Pi ↔ AVR[10]	Skiftning av spänning för busskommunikation	1

6.2 Kopplingsschema

Kopplingsschemat för modulen finns i figur 13.

**Figur 13:** De pinnar på raspberryn som behövs för busskommunikation och med en nivåskiftningschip som omvandlar 3.3 V till 5 V.

6.3 Mjukvara

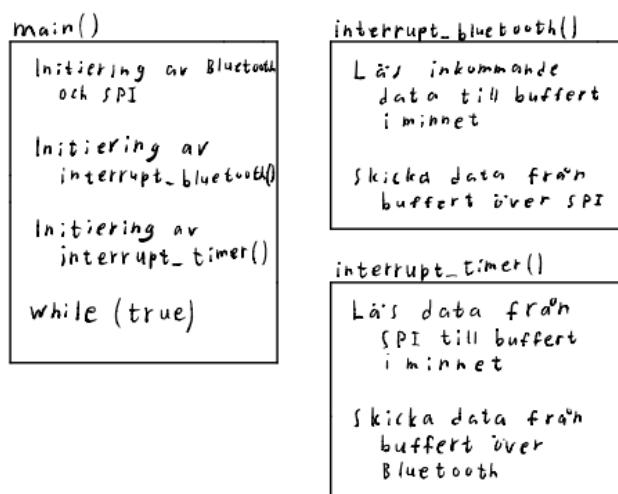
Mjukavaran i kommunikationsmodulen kommer att bestå av ett program. Programmet kommer inledningsvis att initiera kontakt med SPI och Bluetooth, samt upprätta en timer i det fall då roboten befinner sig i autonomt läge.

Eftersom kommunikationsmodulen tjänar som mellanhand till datormodulen och robotens övriga moduler, kommer huvudprogrammet inte utföra några instruktioner efter initieringen, utan kommer vänta på avbrott.

Det kommer att finnas två avbrotsrutiner: en som körs när data inkommer via Bluetooth och en som hämtar data via SPI av ett timer-genererat avbrott.

Uppräkningstiden för timern kan anpassas efter styrbeslutet, så att styrmodulen inte behöver bli avbruten oftare än nödvändigt av kommunikationsmodulen för överföring av styrbeslut. Styrbeslutet ska därefter skickas vidare till datormodulen.

Se figur 14 för ett principskiss över huvudprogrammet och de båda avbrotsrutinerna.



Figur 14: Principskiss av huvudprogrammet och avbrotsrutinerna.

6.3.1 Avbrotsrutiner

Det kommer som tidigare nämnts att finnas två stycken avbrotsrutiner. Samtliga avbrotsrutiner har som syfte att hämta in data från en modul och sedan skicka denna data vidare till en annan, se tabell 12.

Tabell 12: Daten som läses in vid respektive avbrott, vilken modul datan kommer ifrån och till vilken modul datan ska skickas vidare.

Avbrott	Från	Till	Vilken data
interrupt_bluetooth	Datormodul	Styrmodul	Styrkommandon och reglerparametrar.
interrupt_timer	Styrmodul	Datormodul	Styrbeslut och sensordata.

Det första avbrottet ska utföras då det kommer information från datormoden via Bluetooth. Denna informationen är styrkommandon eller reglerparametrar, vilka sedan kommer att skrivas till styrmodulen.

Det andra avbrottet kommer att utföras då en timer ger utslag. Detta avbrott ska skicka en begäran av styrdatal, sensor-data och reglerparametrar till styrmodulen och sedan mellanspara denna information. Informationen skickas sedan över Bluetooth till datormoden.

Ingen av avbrotten kommer att gå att avbryta utan de kommer istället läggas på en stack och utföras när avbrottet är färdigt.

7 DATORMODUL

Datormodulen ska bestå av ett program som kan installeras på en laptop med Bluetooth som ska kunna användas för övervakning och givandet av styrinstruktioner till roboten. Den kommer bestå av ett grafiskt gränssnitt med en karta av det utforska området samt framtida färdväg och till sist den nuvarande kortaste vägen. Utöver detta ska det finnas en ruta som skriver ut de nuvarande reglerparametrarna, gaspådrag för varje hjulpar och avstånden till väggar. Det ska också finnas ett antal iconer som en operatör kan klicka på för att skicka styrinstruktioner till roboten.

Datormodulen ska utöver det grafiska gränssnittet även spara ner alla styrbeslut som styrmodulen bestämmer, samt all bearbetad sensordata som styrmodulen använder som underlag för dess beslut.

Datormodulen behöver vara uppkopplad mot kommunikationsmodulen via Bluetooth.

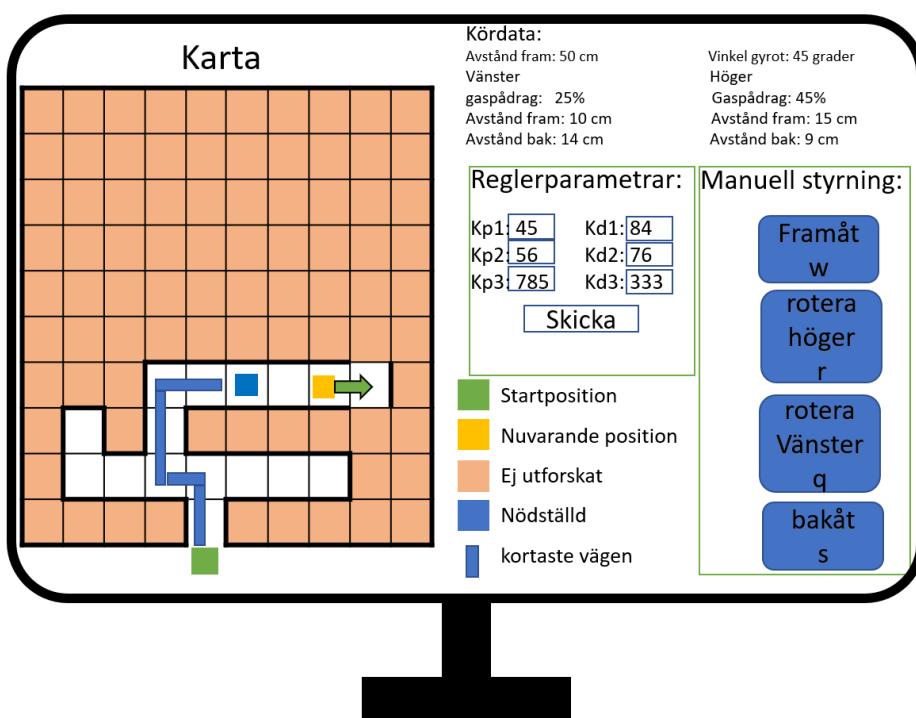
7.1 Mjukvara

Då labyrinten kommer att modelleras med olika siffror för antalet och orienteringen av väggarna (se figur 12 i avsnitt 5.4.1) kommer datormodulen ha ett rutnät, där svarta streck kommer att representera väggar och vit yta kommer representera kartlagd yta och en orange ruta representerar utforskat territorium se bild 15. Då det ej är känt var ingången till labyrinten är i relation till de ytter väggarna kommer kartan att startas vid mitten av den nedre kanten av kartfönstret och skiftas åt lämpligt håll under kartläggning för att hela kartan ska få plats i kartfönstret.

Nästa steg efter att den nuvarande utforska labyrinten är uppript är att lägga in robotens position som gul fyrkant. Sedan ska den nuvarande beräknade vägen vilket fås som en vektor av framtida styrbeslut vilket enkelt kan beräknas och läggas till som en pil.

Om den nödställda är hittad ska sedan den kortaste vägen mellan nödställd och ingången ritas ut, vilket kommer att representeras av en tjock blå linje.

Det kommer att finnas iconer som visar vilka tangenter ger kommandon till roboten och när denna tangent är nedtryckt kommer ikonen att ändra färg. Det kommer att finnas ett fönster med relevant sensordata och nuvarande reglerparametrar. Bredvid nuvarande reglerparametrar kommer en ruta där nya reglerparametrar kan fyllas i för att skickas att finnas.



Figur 15: En visualisering av hur användargränssnittet kommer att se ut.

7.1.1 Sparad data

Det kommer att finnas flera olika typer av data som ska sparas.

- Sensordata från olika sensorer.
- Nuvarande reglerparametrarna.
- Styrmodulens styrbeslut.

Dessa datatyper kommer sparas i olika filer med namn som säger var datan kommer ifrån. Filerna kommer namnges på formen datatyp_timme_dag_månad.txt och i filen kommer datan sparas på formen data kommatecken klockslag (som datormodulen sparade datan) och sedan upprepas detta. Se exempel nedan.

lidar1_14_26_02.txt

1	45, 03:45:34
2	86, 03:45:38
3	23, 03:46:01

8 IMPLEMENTATION

Implementationen av modulerna kommer ske parallellt, men det kommer läggas mest timmar i början på sensormodulen då denna behövs för all reglering i styrmodulen. Samtidigt kommer två personer börja jobba med styrmodulen för att implementera manuell styrning tillsammans med de två personer som jobbar med kommunikationsmodulen och datormodulen.

När sensormodulen är klar kommer personerna som jobbade med denna att fördelas över de tre kvarvarande modulerna med ett fokus på integration med kvarvarande moduler och sedan fokus på styrmodulen och att implementera den autonoma körningen. Under tiden kommer även kommunikationsmodulen och datormodulen att bli funktionellt färdiga, med ett undantag för att gränssnittet i datormodulen ej måste vara användarvänligt än.

Efter den autonoma körningen är implementerad kommer fokus ligga på att alla moduler är ordentligt integrerade och att en tävlingskörning kan utföras. Slutligen kommer fokuset ligga på finslip och optimering.

Sensormodulen kan testas på flera olika sätt. Avståndsmätningarna kan testas genom att se att informationen stämmer med ett bestämt avstånd. Gyrot kan testas genom att manuellt med gradskiva vrida ett antal grader för att se att vinkeln stämmer och reflexsenorn kan testas genom att föra en tejpbit in och ut ur dess sensorområde och undersöka datan som sensorn avger. Styrmodulen testas genom att undersöka hur roboten körs, och de reglerparametrar som används vid reglering kommer ändras via datormodulen tills dess att roboten uppför sig på önskat vis. Innan de olika modulerna är implementerade kan styrmodulen testas genom att undersöka hur robotens rörelse ändras när spänningen över hjulpare ändras, och även simulerad data kan användas som sensordata för att se hur roboten reagerar på sensordata.

Det kommer att finnas tre olika sätt att få feedback från systemet. Det första är att undersöka hur robotten rör sig under autonom körning. Det andra är att se informationen på skärmen till datormodulen och det tredje sättet är att plotta eller på annat sätt analysera den sparade datan.

Sampelfrekvenser har bestämts utifrån datablad och rimlighetskontroller. Given sampelfrekvens för de fyra IR sensorerna är 25 Hz (mätningar per sekund). Lidarsensorn kommer följa samplingsfrekvensen för IR sensornern och därmed också sampelas med 25 Hz. Reflexsensorn kommer sampelas med 200 Hz, gyroskopet med 380 Hz, knapp och brytare med 20 Hz vardera.

Utifrån sampelfrekvenserna och storleken på respektive sensordata som anges i tabell ?? sida ??, går det att göra en grov uppskattning av hur mycket datatrafik som kan komma att uppstå på I²C-bussen under drift. Bussen kommer att användas för att hämta data från två sensorer till sensormodulens processor. Denna data ska sedan, efter bearbetning, skickas till både styrmodulen. Gyroskopet kommer att använda en sampelfrekvens på högst 380 Hz och således den frekvens som behöver användas för sändning till övriga moduler.

Uppskattningen gjordes genom att multiplicera sampelfrekvensen för båda sensorerna med datatypens storlek för varje sensor och sedan summa dessa. Enligt uppskattningen kommer data behöver skickas med strax under 100 kbit/s, vilket är väl inom bussens maxhastighet på 400 kbit/s (i Fastmode).

REFERENSER

- [1] F. Johansson, O. Pihl, M. Ehn, D. Ekblad, M. Jämte, A. Thunberg, E. Sevonius, G. Arctaedius, and L. Byman, “Ban- och tävlingsspecifikation.”
- [2] A. Nicklasson Cedbro, E. Leuchovius, F. Johansson, G. Arctaedius, J. Carlson, and et al., “Kravspecifikation,” februari 2021, version 1.0.
- [3] “Lidar lite v3 operation manual and technical specifications,” <http://www.da.isy.liu.se/vanheden/pdf/lidarlite3OMTS.pdf>, [Online; Hämtad Februari 25, 2021].
- [4] “Reflexsensormodulen,” <http://www.da.isy.liu.se/vanheden/pdf/reflexsensormodul.pdf>, [Online; Hämtad februari 25, 2021].
- [5] “Mems motion sensor: three-axis digital output gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/L3GD20.pdf>, [Online; Hämtad februari 25, 2021].
- [6] “Så här fungerar tryckknappen,” <http://www.da.isy.liu.se/vanheden/image/tryckknapp.png>, [Online; hämtad 2021-03-01].
- [7] “<http://www.da.isy.liu.se/vanheden/pdf/gp2d120.pdf>,” <http://www.da.isy.liu.se/vanheden/pdf/gp2d120.pdf>, [Online; hämtad 2021-03-09].
- [8] “8-bit microcontroller with 128k bytes in-system programmable flash,” <http://www.da.isy.liu.se/vanheden/pdf/atmega1284p.pdf>, [Online; Hämtad februari 25, 2021].
- [9] “BCM2835 ARM peripherals,” <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/BCM2835-ARM-Peripherals.pdf>, 2012, [Online; hämtad 2021-02-28].
- [10] “Nivåskiftare Raspberry ↔ AVR datasheet,” <http://www.da.isy.liu.se/vanheden/pdf/da-nivaskiftare.pdf>, [Online; hämtad 2021-02-26].
- [11] “Raspberry Pi Compute Module 3+ datasheet,” https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf, 2019, [Online; hämtad 2021-02-26].

G FÖRSTUDIE

G.1 Sensorer

Förstudie - Sensoruppgift

Maja Boström, Alice Nicklasson Cedbro, Johanna Carlson

30 april 2021

Version 1.0



Status

Granskad	ANC, MB, JC	2021-04-06
Godkänd	Peter Johansson	2021-04-16

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

INNEHÅLL

1	Inledning	1
1.1	Syfte och mål	1
1.2	Definitioner	1
1.3	Avgränsningar	1
2	Problemformulering	2
2.1	Frågeställningar	2
3	Metod	3
4	Bakgrundsfakta	4
4.1	Lidar	4
4.2	IR-sensor	7
4.3	Gyroskop	9
4.4	Accelerometer	11
5	Resultat	14
5.1	Tillryggalagd sträcka	14
5.2	Avsökning av omgivning	14
5.3	Styrningsvinkel	15
6	Diskussion och slutsats	16
6.1	Källor	16
6.2	Metod	16
6.3	Slutsats	16
	Referenser	18

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-02-25	Första utkast	ANC,MB,JC	ANC,MB,JC
1.0	2021-04-06	Andra utkast	ANC,MB,JC	ANC,MB,JC

1 INLEDNING

I ett kandidatprojekt i elektronik för civilingenjörsprogrammet Teknisk fysik och elektroteknik vid Linköpings universitet ska en autonom undsättningsrobot konstrueras. Undsättningsroboten ska utforska en modellerad byggnad bestående av pappersväggar. För mer specifikationer kring banan, se Ban- och Tävlingsspecifikation [1]. I byggnaden ska roboten hitta nödställda, markerade med tejp på golvet samt kortast möjliga väg mellan de nödställda och byggnadens ingång. Därefter ska en förnödenhet plockas upp från startpositionen med hjälp av en gripklo samt transporteras till de nödställda på kortast möjliga tid. Roboten ska vara modulbyggd där en av modulerna ska hantera sensorer och sensordata. Förstudien kommer användas som underlag vid konstruktion och utveckling av sensormodulen.

I en autonom undsättningsrobot behövs sensorer för att uppskatta omgivningens position i förhållande till roboten, sin egen position, tillryggalagd sträcka samt färdriktning. Sensordata ligger till grund för de reglerbeslut som ska tas av roboten och bör därför vara så precis som möjligt. Vilka sensorer som lämpar sig för roboten till detta projekt bestäms av faktorer som ändamålsenlighet, tillförlitlighet, kostnad och tillgänglighet.

1.1 Syfte och mål

Syftet med förstudien är att få fördjupade kunskaper om sensorer och hur dessa kan användas för att ta fram lämplig data inför reglerbeslut. Kunskaperna kommer användas i utvecklandet av en autonom undsättningsrobot.

1.2 Definitioner

Allmänna förkortningar och termer som används i förstudien definieras i Tabell 1.

1.3 Avgränsningar

Förstudien har begränsats till att endast diskutera modeller av sensorer som finns tillgängliga i labbet vid konstruktionen av undsättningsroboten. Även vid sökning av bakgrundsfakta har vissa avgränsningar gjorts. Dels var vissa metoder som hittades mycket avancerade och därmed beskrivs dessa mer kortfattat och simpelt, dels har bakgrundsfakta respektive datablad endast sökts fram via UniSearch och VanHeden (se metod 3 för mer information).

Tabell 1: Tabell över termer och definitioner

Term	Definition
Lidar	Laserpositionsmätare
Tof	Time-of-flight. Tiden det tar för ljus att transporteras ett visst avstånd.
AMCW	Lidarteknik som står för: Amplitude modulation of a continuous wave
FMCW	Lidarteknik som står för: Frequency-modulated continuous-wave
Gyroskop (Gyro)	Vinkelhastighetsmätande sensor
Odometri	Metod för att få fram tillryggalagd sträcka genom mätningar på ett hjuls rotation.
Buffer	Plats att temporärt spara data på.
I^2C .	Seriell och synkron buss som kan användas för att integrera sensorer och kretsar[2].
UART	Liten krets i mikrokontroller för seriell kommunikation.
Triggning	Ett sätt att starta t.ex. en mätning.
Accelerometer	Ett verktyg för att mäta linjär acceleration och vinkelacceleration.
Kodomvandlare	Tar en insignal och omvandlar den till en önskad utsignal.
V_{cc}	Matningsspänning.
GND (Jord)	Jordning betyder att man ansluter olika ledare till en gemensam jordningspunkt (nollpunkt).

2 PROBLEMFORMULERING

För att en undsättningsrobot ska kunna genomföra sitt uppdrag är det nödvändigt att den har tillgång till olika sensordata. Det viktigaste är att ha kunskap om omgivningens utseende för att förhindra kollision mellan roboten och hinder. För att genomföra undsättningen på minimal tid är det även viktigt att veta tillryggalagd sträcka. Allt detta lade grunden till våra två inledande frågeställningar, se 2.1. Den sista frågeställningen grundas i att sensormodulen behöver förse styrmodulen med data så att roboten kan regleras. En sådan data kan vara en styrningsvinkel vilket är vinkelavvikelsen från mittlinjen, där mittlinjen är den linje som skär roboten mitt itu och som den ska följa framåt.

2.1 Frågeställningar

- Vilka metoder för beräkning av tillryggalagd stäck är lämpliga för undsättningsroboten?
- Hur kan ljussensorer användas för avsökning av omgivning och vilken sensor är mest lämplig för undsättningsroboten?
- Vilka sensorer kan användas för beräkning av styrningsvinkel?

3 METOD

Metoden som används för att svara på rapportens frågeställningar i kapitel 2 är att bakgrundsfakta om lidar, IR-sensorer, gyroskop och accelerometrar insamlats från artiklar och datablad. Alla artiklar har sökts fram från UniSearch via Linköpings universitets biblioteks hemsida och alla datablad har tagits från VanHeden. Vanheden är en hemsida som drivs av Linköpings universitet för projektkurserna inom datorteknik där datablad och annan användbar information finns. Informationen om sensorerna har sedan sammanställts under respektive sensorrubrik i kapitel 4. Dessa inleddes med en kort beskrivning av generell funktionalitet hos vardera sensorteknik, för att sedan lista och djupare beskriva de tillgängliga komponenterna i respektive sensorkategori.

Vidare gjordes jämförelser mellan de olika komponenterna i vardera sensorkategori i resultatdelen i kapitel 5 för att undersöka vilka alternativ som tycktes lämpligast för undsättningsroboten, och därefter har frågeställningarna besvarats. Resultaten har sammanfattats i diskussionsdelen i kapitel 6, där även slutsatser om förstudien dragits samt en diskussion om metodens respektive källornas tillförlitlighet förts.

4 BAKGRUNDSFAKTA

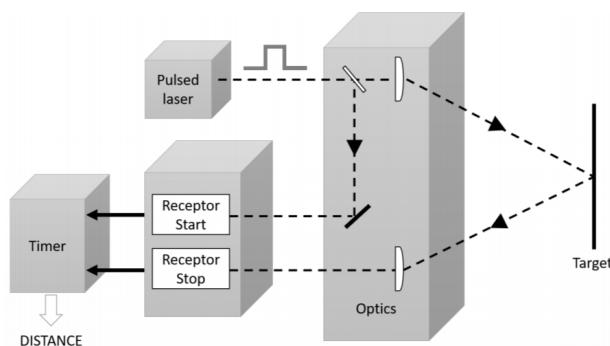
Här presenteras de sensorer som förstudien behandlar. En övergripande beskrivning av sensortypernas generella teknik presenteras först för respektive kategori och därefter följer sammanfattningar av de undersökta komponenternas tekniska specifikationer.

4.1 Lidar

“Light Detection And Ranging”, lidar, är en typ av sensor som grundar sig i mättekniken time-of-flight, vilket är tiden det tar för ljus att transporteras ett visst avstånd. Lidar sänder ut kortvariga optiska signalpulser, laserpulser, och detekterar de signalpulser som reflekteras tillbaka från omgivningen. Tiden mellan emitteringen och mottagandet mäts och sensorn kan därifrån beräkna avstånd och kartlägga omgivningen. Omgivningen återskapas som ett punktmoln i filformatet LAS. Lidar används i många autonoma fordon och bilar.

De optiska signalpulserna i lidarn kan sändas ut och detekteras på tre olika sätt. Det första tillvägagångssättet är en lasersignal som skickas ut i pulser, se Figur 1. Denna metod kallas därför puls-lidar. Tid mäts tills signalens echo mottagits och avståndet, R som signalen transporterats kan sedan beräknas enligt

$$R = \frac{c}{2} t_{of}.$$



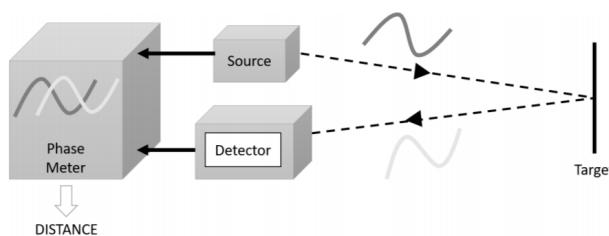
Figur 1: Illustration [3] av pulserande lidar.

Fördelarna med puls-lidar är att det är en simpel metod som fungerar för långa avstånd såväl inomhus som utomhus. Denna typ av lidar kan ha maximalt mätavstånd från några meter till kilometer. Nackdelar med puls-lidar är att de kan vara dyra om de ska verka på långa avstånd då det då krävs en mottagningdetektor med hög känslighet.

Det andra sättet bygger på "Amplitude modulation of a continuous wave"-tekniken, AMCW, se Figur 2. Modulerade ljusvågor (modulerade med frekvensen f_M) sänds då ut kontinuerligt och avståndet fås genom att jämföra fasen på de utsända vågorna med fasen på de mottagna, $\Delta\Phi$. Avståndet beräknas enligt

$$R = \frac{c}{2} \frac{\Delta\Phi}{2\pi f_M}.$$

AMCW har samma precision (centimeternivå) som pulslidar men kan inte mäta lika lång distans. AMCW-metoden fungerar endast inomhus.

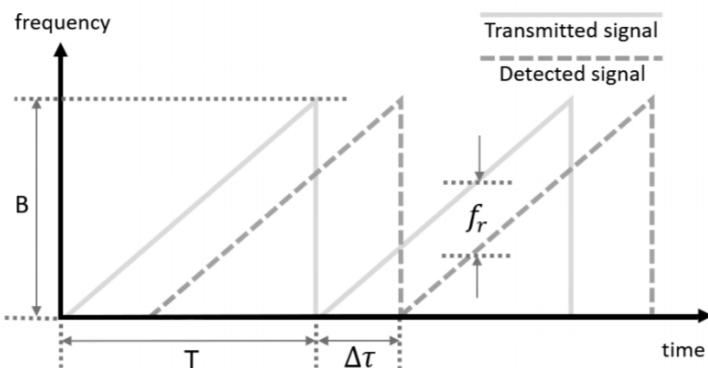


Figur 2: Illustration [3] av lidar med AMCW metoden.

Den tredje optiska signalmetoden för lidar grundas i "Frequency-modulated continuous-wave"-tekniken, FMCW, se Figur 3. Metoden har hög precision, milimeterprecision på både korta och långa avstånd. Vid användandet av FMCW-tekniken ändras en referenssignals period samtidigt som den sänds ut kontinuerligt. En stående våg med frekvensen f_r skapas mellan signalen som reflekteras från omgivningen och referenssignalen. Denna frekvens är densamma som den konstanta frekvensskillnaden mellan de två signalerna. Avståndet till materia i lidarns omgivning fås genom att frekvensen f_r är proportionell mot avståndet enligt

$$R = f_r \frac{cT}{2B}.$$

Proportionaliteten gäller endast om perioden, och därmed frekvensen på referensignalen, förändras linjärt, vilket illustreras tydligt i Figur 3. I denna figur är T tiden för en linjär ramp och B frekvensskillnaden i rampen, dvs. bandbredden.



Figur 3: Illustration [3] av FMCW-metoden.

Fördelar med FMCW-metoden är att en sensor av denna typ kan detektera hastigheten (inklusive riktning) samtidigt som avståndet och att precisionen på avståndet blir på millimeternivå. Metoden fungerar även bra både inom- och utomhus. En nackdel med FMCW är att tekniken är känslig för förhållandena inuti och utanför sensorn vilket ställer krav på hårdvaran. Denna teknik används inte i lika stor utsträckning i autonoma fordon som puls- och AMCW-lidar [3]

4.1.1 RPLIDAR A2

RPLIDAR A2 är en form av roterande laser, vilken kan mäta 360° i 2D med en räckvidd på 0.15 till 12 eller 18 m (går att optimera till 18 m med modifiering av algoritmer). [4] Lidarn har en samplingsfrekvens på 2 000–8 000 Hz dvs. kan ta upp till 8 000 samplingar per sekund, erfodrar strömförsörjning på 5 V samt har en seriell kommunikation genom UART. Samplingarna innehåller bl.a. information om avstånd till objekt, samt vinkelns med vilken mätningen utfördes. Med dessa kan lidarn genom en inbyggd kartläggningsteknik skapa en typ av översiktig karta över sin omgivning vilken gör den lämplig för navigering och lokalisering, exempelvis för mindre robotar som robotdammsugare. [5] En ytterligare funktion hos RPLIDAR A2 är att laserns rotationshastighet går att ställa in manuellt. Detta genom en anslutning i gränssnittet som kallas MOTOCTL, dit en PWM-signal i sådant fall kopplas med önskad frekvens.

4.1.2 LIDAR-Lite v2

LIDAR-Lite v2 är en laserlidar som bygger på pulstechniken beskriven under 4.1 med ett maximalt pulståg på 256 pulser. Med en vikt på endast 22 g kan denna lidar ses som ett lämpligt alternativ till små enklare robotar eller drönare.[6] Laserns våglängd ligger nominellt på 905 nm (osynligt, infrarött ljus) och har en divergens på approximativt $4 \text{ mRad} \times 2 \text{ mRad}$. Den erfodrar en matningsspänning på 4.75–5.5 DCV (maximalt 6 VDC), har en noggrannhet på $\pm 2.5 \text{ cm}$ samt upplösning på $\pm 1 \text{ cm}$.[7]

v2:an kan förrutom avståndsberäkning även mäta signalstryka och hastighet, med hjälp av sitt I^2C -gränssnitt. Gränssnittet möjliggör anpassning av interna kontrollergister för användaren, vilka finns i sensorns processor, som därmed t.ex. kan ändra inställningar eller få ut värden. Extern triggning är ett ytterligare ett sätt att aktivera sensorn, som för 3.3 V logic triggas på hög-låg flank. Mer om dessa register samt detaljer kring hur insamling och behandling av data går till återfinns i sensorns kontrollregisterokumentation.[8]

Lidarn har en samplingshastighet upp till 500 Hz, räckvidd på ca 40 m, och uppges fungera väl både inom- och utomhus enligt dess tekniska beskrivning. Linjäritet ej garanterad på mätningar under 1 m. [9]

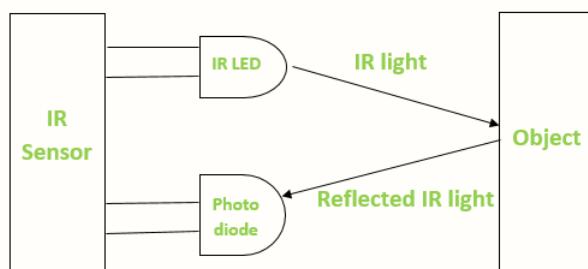
4.1.3 LIDAR-Lite v3

Lidar-Lite v3 innehåller alla funktioner som dess föregångare Lidar-Lite v2, beskriven ovan, har. [10] Även denna lidar har således en våglängd på 905 nm, divergens på 8 mRad, matningsspänning på 4.5–5.5 VDC och noggrannhet på $\pm 1\text{cm}$ respektive upplösning på $\pm 2.5\text{cm}$. Samplingshastigheten är också upp till 500 Hz samt räckviddden ca 40 m.[11]

Således verkar v3:an vara en nyare och troligen uppdaterad version av v2:an[12], då sistnämnda till skillnad från förstnämnda ej finns att tillgå i flertalet webshops.

4.2 IR-sensor

En IR-sensor avger ljus för att upptäcka objekt i dess omgivning. Den har en sändare som sänder iväg ljus och en detektor som upptäcker ljuset som har studsat på ett föremål och tillbaka till sensorn, se Figur 4.

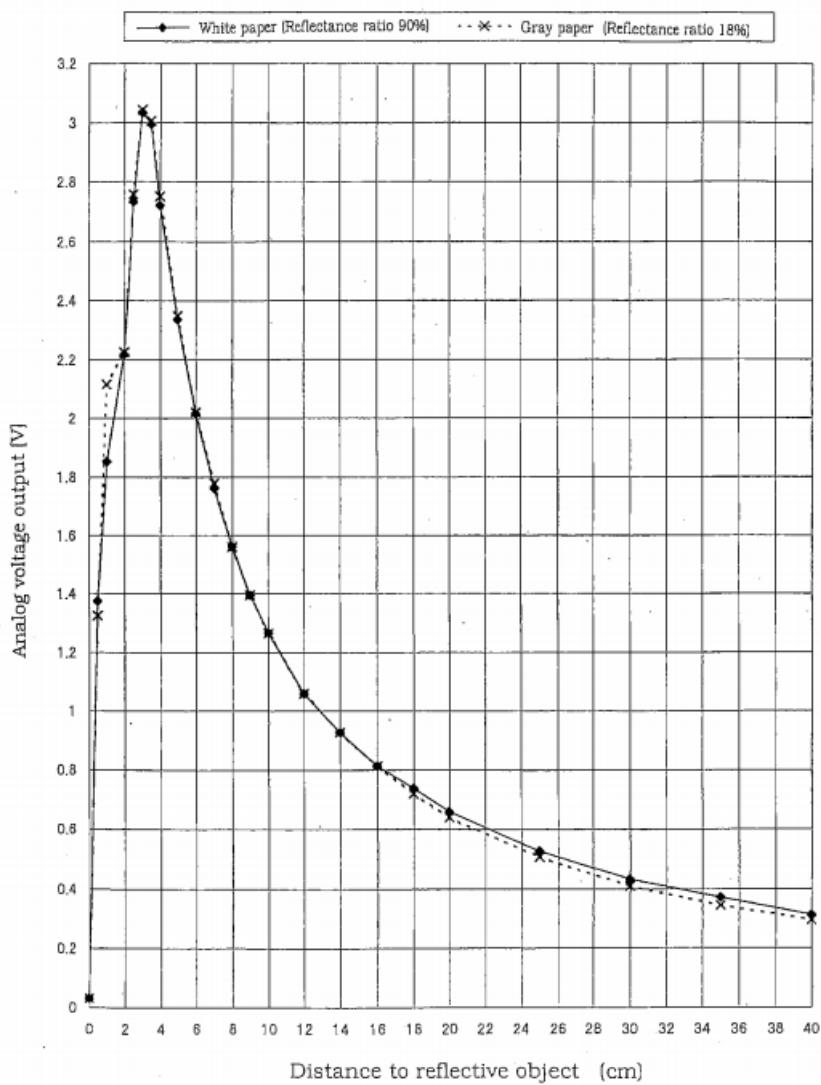


Figur 4: En demontering på hur ljuset studas på ett föremål och tillbaka till sensorn. [13]

Fotodioden, se Figur 4, har en resistans och en utspänning som ändras proportionellt med ljuset som reflekteras tillbaka från objektet. I artikeln skriven av Mustafa Özcan, Fuad Aliew och Haluk Görgün skriver de att infraröda sensorer är använda över hela världen för att uppskatta avstånd ty deras användbarhet och kostnadseffektivitet. En nackdel med denna typ av sensor är att den inte är linjär och kan ge låg noggrannhet och precision [14] vid för små eller för stora avstånd som sensorn inte klarar av.

4.2.1 GP2Y0A41SK0F

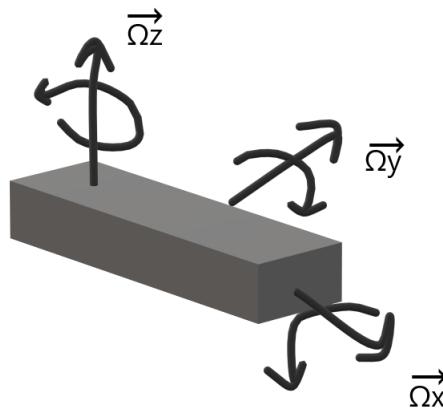
GP2Y0A41SK0F är en IR-sensor som mäter avstånd till olika föremål. Sensorn ger ut en analog signal och kan beräkna avstånd mellan 4 och 30 cm. För att stabilisera matningsspänningen bör en kondensator med värdet $10 \mu\text{F}$ placeras mellan V_{cc} och GND. Avståndet mellan ett objekt och sensorn avläses indirekt genom spänningen på sensorns utsignal, se Figur 5. Detta spänningsvärdet kan skilja sig något åt för likvärdiga avstånd beroende på hur reflektivt objektets material är vilket också kan avläsas i Figur 5. [15]



Figur 5: Visar hur sensorns utsignal varierar med avståndet och grått respektive vitt papper.

4.3 Gyroskop

Ett gyroskop är en sensor som kan bestämma ett fordons vinkelhastighet. Det är en vanlig sensor för att bestämma positionen och vinkelhastigheten av ett objekt. Axlarna hos ett gyroskop påverkas inte av att ett föremål vrider utan de kan ge stabilitet och fungera som referens i ett autonomt navigationssystem. [16] Ett gyro kan ofta detektera vinkelhastighetsförändringar i tre riktningar, se Figur 6.



Figur 6: Riktning av detekterbara vinkelhastigheter.

En metod för att bestämma vinkelhastigheten och positionen hos ett föremål med ett gyroskop är att använda en till två algoritmer, en för vinkelhastighet och en för positionen. [17] Metoden går ut på att först bestämma vilken vinkelhastighet som roboten har och därefter bestämma var roboten nu befinner sig. I artikeln ”*Indoor Localization for Skid-Steering Mobile Robot by Fusing Encoder, Gyroscope, and Magnetometer*” skriver författarna att deras metod kräver hög noggrannhet på vinkelhastighetsalgoritmen för att positionsalgoritmen ska ge ut ett bra resultat då de är beroende av varandra.

En annan metod för att bestämma vinkelhastigheten hos ett fordon är att använda sig av gyrobaserad odometri. [18] Metoden grundar sig i att man kan bestämma vinkelhastigheten från två olika data, en från ett gyroskop och en från hjulens odometri, se ekvation 1.

$$\Delta\omega = \omega_{gy} - \omega_{wo} \quad (1)$$

I ekvation 1 är ω_{gy} vinkelhastigheten från gyroskopet och ω_{wo} är vinkelhastigheten baserat på hjulens odometri. Denna metod hanterar inte om robotens hjul glider mot underlaget. Författarna skriver i sin artikel ”*Gyro-based odometry associated with steering characteristics for wheeled mobile robot in rough terrain*” att den grundläggande metoden kan utökas med flera avancerade beräkningar och relativa fel, se ekvation 2 och figur 7.

$$\hat{\omega} = (1 - w)\omega_{wo} + w\omega_{gy}, (0 \leq w \leq 1) \quad (2)$$

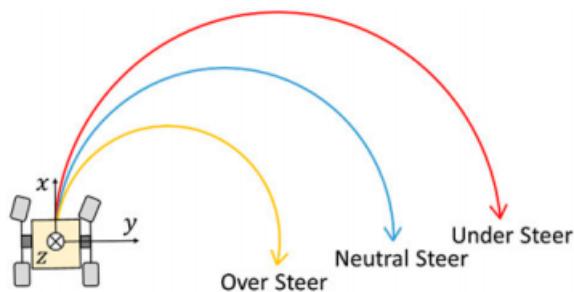


Figure 1. Stability factor.

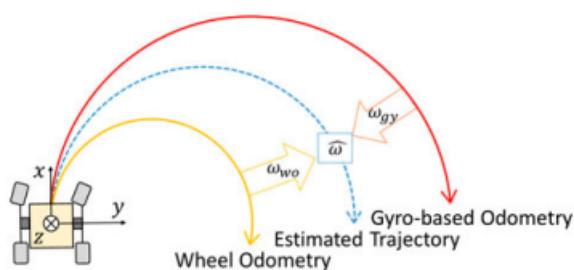


Figure 2. Approach of proposed method.

Figur 7: Figuren illustrerar beräkningen som görs enligt ekvation 2.[18]

4.3.1 Adafruit Precision NXP 9-DOF, FXAS21002C

Adafruits FXAS21002C är ett 3-axlat gyroskop med kapacitet att mäta vinkelförändringar från 250–2 000 °/s och erfordrar matningsspänning på 1.95–3.6 V. Denna spänning bör ej överstigas under längre perioder då detta kan leda till skada hos produkten.[19] För att kommunicera med gyrots interna register finns stöd för både seriell I²C- och SPI-kommunikation med master-/slave-förhållande där gyrot alltid är slav gentemot andra processorer.

FXAS21002C har 16-bitars utsignal i 2-komplements form som kommer i hastigheter mellan 12.4 och 800 Hz, och inbyggt lågpass- samt högpassfilter för att begränsa utsignalens bandbredd. För att minska belastningen på bussen för I^2C - respektive SPI-kommunikationen har gyrot även en s.k. FIFO-databuffer som kan lagra samplingar upp till 192 bytes, dvs. 32 bitar med 3-axeldata, så att belastningen på bussen minskar.

4.3.2 Adafruit 10-DOF, L3GD20

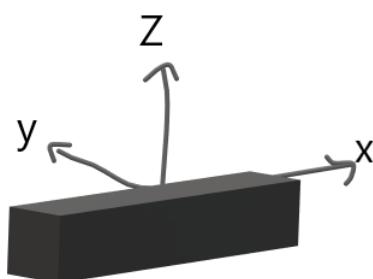
Det andra undersökta gyroskopet L3GD20 är snarlikt FXAS21002C. De har i princip samma tekniska egenskaper [20], med skillnaden att L3GD20:an har högre maximal matningsspänning på 4.8 V. L3GD20 kan således ses som ett likvärdigt alternativ till FXAS21002C om än enklare att få tag i då fler återförsäljare gick att finna.

4.3.3 MLX90609

MLX90609 är en vinkelhastighetssensor som även denna har 3 axlar, men något högre matningsspänning på 5 V (maximalt 5.25 V) samt en kapacitet att mäta vinkelförändringar upp till 300 °/s. Gyrot har ett SPI-gränssnitt att kommunicera med där sensorn alltid är slav.

4.4 Accelerometer

En accelerometer är en typ av sensor som kan användas till att mäta linjär acceleration genom att mäta kraftpåverkningar [21][22][23]. En accelerometer kan upptäcka ändringar i accelerationen i tre dimensioner, se Figur 8. Det är vanligt att använda accelerometrar för detektion av position och rotation av ett fordon. Andra tillämpningar är stegeräknare i smartklockor och användning för att beräkna hur snabbt man cyklar. En accelerometer är generellt billig att köpa. [24].



Figur 8: Riktning av detekterbara accelerationer.

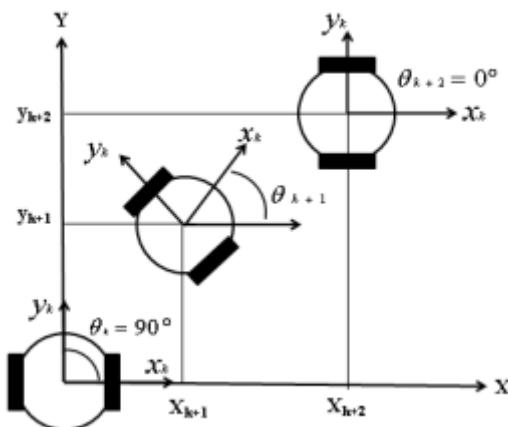
I artikeln "Velocity Estimation by Using Position and Acceleration Sensors" [25] skriver författarna om hur viktigt det är att kunna uppskatta hastigheten hos ett föremål. De tar upp en metod för att beräkna ett föremåls hastighet med hjälp av föremålets position, en kodomvandlare, och en accelerometer med 3 som utgångsekvation för att beräkna hastigheten.

$$\frac{x(k) - x(k-1)}{T} \quad (3)$$

I ekvation 3 representerar $x(k)$ föremålets position och T representerar samplingsperioden. Med omskrivningar och andra uträkningar utökar författarna ekvationen för hastighetsberäkning i tre dimensioner. I tre dimensioner är det många olika parametrar och många beräkningar som ska göras.

En accelerometer kan också användas för att beräkna den tillryggalagda sträckan som en robot har gjort. [26] En accelerometers utsignal kan dubbel-integreras med avseende på tiden för att erhålla den tillryggalagda sträckan hos ett fordon. Denna metod minskar i säkerhet med tiden på grund av integrationen, skriver författarna i sin artikel "Accelerometer for Mobile Robot Positioning". Ett sätt att öka säkerheten för att bestämma den tillryggalagda sträckan är att använda ytterligare sensorer som kan bekräfta avståndet som fordonet har rört sig framåt eller som på andra sätt ökar säkerheten på systemet.

En annan metod för att bestämma den tillryggalagda sträckan som en robot har gjort är att använda ett kalmanfilter. [27] För att bestämma metoden för att beräkna robotens orientation och position illustreras författarna den i artikeln, se Figur 9.



Figur 9: Robotens position och orientation illustreras i Figur [27].

Författarna skriver i sin artikel ”*Sensor Data Fusion using Unscented Kalman Filter for Accurate Localization of Mobile Robots*” att man kan använda accelerometrar, kodomvandlare och gyroskop för att erhålla information om hur robotten har rört sig. Även i denna metod kan felet att öka i takt på grund av den dubbla integrationen från accelerometern. Systemet kan också hantera om robotten skulle slira då den kan byta mellan två olika modeller, en som hanterar slir och en som inte gör det. Denna metod kan jämföras med en exakt laserpositions-sensor skriver också författarna i sin text.

4.4.1 **LSM303DLHC**

Accelerometern ”LSM303DLHC” är en sensor som kan mäta 3D-digital linjär acceleration och mäta 3D-linjärt magnetfält se Figur 8. Den kan appliceras till olika användningsområden som detektion av position med mera. [28] Accelerometern ger ut en digital utsignal som kan skickas genom I^2C .

5 RESULTAT

5.1 Tillryggalagd sträcka

Det finns flera olika metoder för att beräkna en tillryggalagd sträcka för en robot. Respektive metod som vi undersökt i denna förstudie använder sig av gyroskop, accelerometer eller lidar. Vid användande av gyroskop beräknas avståndet genom odometri, vid användande av accelerometer dubbelintegreras den framtagna accelerationen till en sträcka och lidarn har ett register som lagrar sträckan.

Att använda ett gyroskop eller en accelerometer är komplicerat då sensorerna måste användas i kombination med avancerade algoritmer som omvandlar sensordata till en sträcka. Dessa algoritmer kan ge större risk för felberäkningar då man använder sig av integration. Då lidarn redan är konstruerad att bara hämta den tillryggalagda sträckan från ett register är den metoden mycket simplare än att använda ett gyroskop eller en accelerometer.

Detta betyder att den lämpligaste metoden att beräkna den tillryggalagda sträckan för en robot är att använda en lidar, som svarar på vår frågeställning om vilken metod som är lämpligast för att beräkna den tillryggalagda sträckan för undsättningsroboten. Detta då denna metod är simlast men ändå har god noggrannhet.

5.2 Avsökning av omgivning

Till denna förstudie har två olika typer av ljus-sensorer undersökts vilka kan användas för avsökning av omgivning, lidar och IR-sensorer, där förstnämnda innehållar ett antal olika varianter som bygger på olika tekniker.

Utifrån projektets mål och syfte har jämförelserna mellan dessa komponenter gått ut på att väga kapacitet mot kostnad, mer konkret: hur stor noggrannhet och hur mycket funktioner kräver vårt projekt? Enligt specificerade regler för projektet kommer avståndsberäkningen endast ske på korta avstånd i sidled, samt förhållandevis långa avstånd i robotens framåtriktning. Detta leder oss till slutsatsen att enklare sensorer som mäter bra på korta avstånd ger tillräcklig noggrannhet i sidled, medan en sensor med längre räckvidd är att föredra framåt.

Därav presenteras en Lidar-Lite v2 eller v3 som det bästa alternativet för avståndsberäkning i robotens framåtriktning, då dessa har lång räckvidd på upp till 40 m men mäter sämre på korta avstånd. Vår uppfattning är att den långa räckvidden kan vara nödvändig för vissa konstellationer av avsökningsmiljön, och att precisionsmätningar på korta avstånd inte är lika viktiga i robotens framåtriktning. Då Lidar-Lite v2 och Lidar-Lite v3 verkar ha mycket små skillnader jämförelse med varandra får tillgången vid utlämning av komponenter vara den avgörande faktorn gällande vilken av dessa som kommer att användas till konstruktionen.

För avståndsberäkning i sidled gäller det omvänta i fråga om prioriterad noggrannhet; långa avstånd kommer inte behöva mäts utan det är robotens avstånd till respektive sidovägg som är aktuella. För detta ändamål tycks ett antal IR-sensorer uppfylla våra önskemål om noggrannhet väl då de verkar bra på korta avstånd om 4 – 30 cm men inte är ett alternativ för mätningar till längre avstånd. För få en så tydlig bild som möjligt av robotens position i förhållande till dess omgivning kommer fyra stycken IR-sensorer placeras på roboten, pekande ut mot väggarna i linje med fram- resp. bakhjulen.

Vidare är tanken att under projektets konstruktionsfas undersöka hur sensorerna svarar på avstånd utanför de intervall som de enligt databladet är garanterade en viss noggrannhet för. Detta för att kunna hantera datan som t.ex. en IR-sensor registrerar när roboten kör förbi en öppen gång och avståndet blir längre än 30 cm, eller när den framåtriktade lidarn hamnar nära en vägg och dess avsökningsområde blir mindre än en meter vilket enligt databladet kan orsaka vissa typer av mätstörningar.

5.3 Styrningsvinkel

För beräkning av styrningsvinkel verkar en enklare form av gyroskop vara lämpligt för projektet, detta då roboten inte är i behov av diverse extrafunktioner såsom temperaturavläsning. Således gäller även här resonemanget att mer komplexa komponenter troligen bara leder till att mer tid läggs på att förstå dem snarare än att snabbt kunna utföra mätningar och fokusera på att få dessa utförda med precision.

Detta tillsammans med en önskan om att kunna undvika nivåkonvertering till följd av olika matningsspänning leder oss till att de specifika komponentönskemålen därmed blir någon utav Adafruits 10-DOF eller MLX:an, vilka båda ligger på samma komplexitetsnivå relativt sett. Så även här får tillgången hos återförsäljaren (komponentutlämningen vid projektets konstruktionsfas början) ha en avgörande roll.

6 DISKUSSION OCH SLUTSATS

6.1 Källor

De källor som används är mestadels artiklar som är peer reviewed. Det betyder att publikationerna har godkänts av olika ämnes experter som har läst igenom de noggrant. Slutsatsen på de källorna är att de håller en hög kvalité och ger rapporten en högre standard än om rapporten skulle baserats på källor som inte är peer reviewed. Rapporten är också baserad på datablad om de olika komponenterna (tas upp i avsnittet 4) som är skriva av respektive tillverkare. De ska vara faktaenliga och det var även intycket vi fick men det kan inte bortses från att databladen kan vara partiska. Detta eftersom en tillverkare vill sälja sin produkt och därmed inte framhäva dess negativa egenskaper.

6.2 Metod

Metoden som används i denna förstudie har till viss del präglats av de begränsade möjligheterna att vistas i den miljö som projektets konstruktionsfas är planerad att vara i. Det har endast varit möjligt att söka efter relevant fakta på distans, och inte funnits tillfället att exempelvis köra tester eller undersöka hårdvara fysiskt.

Trots detta har utförandet fungerat väl, den mesta relevanta faktan för att besvara frågeställningarna har gått att finna via internet. Dock hade vissa typer av mätningar troligen underlättat och gjort resultaten mer pålitliga än nu när bara teori används som underlag. Det är även lättare att skaffa sig en helhetsbild av sensorernas komplexitet vid egna undersökningar, vilket möjligen kunde lett till snabbare insikt om att för många funktioner snarare är en nackdel än en fördel i vårt fall.

6.3 Slutsats

Metoden för denna undersökning har endast bestått av faktainsamling samt jämförelser utgående från teorin; inga fysiska tester eller undersökningar har kunnat utföras utan undertecknade har utgått från att informationen som bl.a. komponenternas tillverkare anger i sina datablad är tillförlitlig.

Ur denna har slutsatsen att för längre avståndsberäkningar och beräkning av tillryggalagd sträcka tycks en Lidar-Lite v2 eller Lidar-Lite v3 vara ett bra alternativ. Detta då tillvägagångssättet med lidarn är simplast med minst risk för missberäkningar av den tillryggalagda sträckan i jämförelse med andra tidigare nämnda alternativ. Lidarn ses även som det bästa alternativet för avståndsberäkningar i robotens framåtriktning då den har hög precision på upptill 40 m, med reservation för avvikelse på korta avstånd mindre än en meter, och inte har många (för detta projekt) överflödiga funktioner.

Därav dras även slutsatsen att IR-sensorer av typen GP2Y0A41SK0F är det lämpligaste sättet att beräkna robotens kortare avstånd i sidled till väggarna, då dessa har som högst noggrannhet för 4 – 30 cm vilket är inom ramarna för kraven på tävlingsbanan roboten ska verka i. De verkar väl använda i projekt likt detta och ses därmed som ett bra alternativ för undsättningsroboten.

Ett gyro utan många överflödiga funktioner och med matningsspänning som inte kräver nivåkonvertering är vad som slutligen önskades när beräkning av styrningsvinkel undersöktes, och detta stämde överens med beskrivningarna av Adafruits 10-DOF samt MLX:an.

I vissa fall i resultatdelen har flera komponenter angetts som alternativ till konstruktionen. Detta för att ge utrymme för viss flexibilitet ifall någon komponent saknas hos återförsäljaren (komponentutlämningen vid projektets konstruktionsfas början).

REFERENSER

- [1] . Grupp 1, 2 och 3, “Ban- och tävlingsspecifikation,” 2021.
- [2] A. Pini, “Varför det är så enkelt att ansluta integrerade kretsar med inter-integrated circuit-bussen (i2c) – och hur den ska användas,” <https://www.digikey.se/sv/articles/why-the-inter-integrated-circuit-bus-makes-connecting-ics-so-easy>, [Online; hämtad Februari 18, 2021].
- [3] M. B.-G. Santiago Royo, “An overview of lidar imaging systems for autonomous vehicles,” *Applied Sciences*, vol. 9, no. 19, p. 4093, 2019.
- [4] “Rplidar a2,” <https://www.slamtec.com/en/Lidar/A2Spec>, [Online; Hämtad Februari 25, 2021].
- [5] “Low cost 360 degree laser range scanner,” <https://www.slamtec.com/en/Lidar/A2Spec>, [Online; Hämtad Februari 25, 2021].
- [6] “Lidar lite v2 – avståndsgivare 0-40m,” <https://www.electrokit.com/produkt/lidar-lite-v2-avstandsgivare-0-40m/>, [Online; Hämtad februari 24, 2021].
- [7] “Lidar-lite specifications,” <http://www.da.isy.liu.se/vanheden/pdf/lidarlite2DS.pdf>, [Online; Hämtad februari 24, 2021].
- [8] “Detailed register definitions,” http://www.da.isy.liu.se/vanheden/pdf/pli-03_detailed_register_definitions.pdf, [Online; Hämtad februari 24, 2021].
- [9] “Technology and system hardware overview,” http://www.da.isy.liu.se/vanheden/pdf/lidarlite2_overview.pdf, [Online; Hämtad februari 24, 2021].
- [10] “Lidar-lite v3 optical distance measurement sensor,” <https://buy.garmin.com/en-US/US/p/557294>, [Online; Hämtad februari 24, 2021].
- [11] “Lidar lite v3 operation manual and technical specifications,” <http://www.da.isy.liu.se/vanheden/pdf/lidarlite3OMTS.pdf>, [Online; Hämtad februari 24, 2021].
- [12] “Lidar lite back with v3 after garmin aquisition,” <https://diydrones.com/profiles/blogs/lidar-lite-back-with-v3-after-garmin-aquisition>, [Online; Hämtad februari 24, 2021].
- [13] D. Priyanka, “Ir sensor working and applications,” <https://robu.in/ir-sensor-working/>, [Online; Hämtad april 2, 2021].

- [14] H. G. Mustafa Özcan, Fuad Aliew, “Accurate and precise distance estimation for noisy ir sensor readings contaminated by outliers,” <https://www.sciencedirect.com.e.bibl.liu.se/science/article/pii/S0263224120301706?via%3Dhub>, [Online; Hämtad april 2, 2021].
- [15] “Apto-analog devices division electronic components group sharp corporation,” <http://www.da.isy.liu.se/vanheden/pdf/sharp-GP2Y0A41SK0F.pdf>, [Online; Hämtad april 2, 2021].
- [16] C. McFadden, “What gyroscopes are, how they work, and their importance,” <https://interestingengineering.com/what-gyroscopes-are-how-they-work-and-their-importance>, [Online; Hämtad februari 24, 2021].
- [17] J. Q. Wenjun Lv, Yu Kang, “Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer,” <https://ieeexplore.ieee.org.e.bibl.liu.se/stamp/stamp.jsp?tp=&arnumber=8071153>, [Online; Hämtad februari 24, 2021].
- [18] G. I. Takuma Nagata, “Gyro-based odometry associated with steering characteristics for wheeled mobile robot in rough terrain,” <https://eds-a-ebscohost.com.e.bibl.liu.se/eds/pdfviewer/pdfviewer?vid=8&sid=926a6c4b-da53-4213-8173-4251d5fc4da6%40pdc-v-sessmgr04>, [Online; Hämtad februari 24, 2021].
- [19] “3-axis digital angular rate gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/FXAS21002.pdf>, [Online; Hämtad februari 24, 2021].
- [20] “Mems motion sensor: three-axis digital output gyroscope,” <http://www.da.isy.liu.se/vanheden/pdf/L3GD20.pdf>, [Online; Hämtad februari 24, 2021].
- [21] ST, “Accelerometers,” <https://www.st.com/en/mems-and-sensors/accelerometers.html>, [Online; hämtad Februari 18, 2021].
- [22] “Accelerometers and forces,” <http://cbakkennet.ipage.com/cga/packet/general/accelforces.html>, [Online; hämtad Februari 18, 2021].
- [23] “Linear accelerometer (inertial sensor + data manipulation),” <https://www.vieyrasoftware.net/sensors-sensor-modes>, [Online; hämtad Februari 18, 2021].
- [24] R. D. M. B. Mohamed Fazeen, Brandon Gozick and M. C. González, “Safe driving using mobile phones,” <https://ieeexplore.ieee.org.e.bibl.liu.se/stamp/stamp.jsp?tp=&arnumber=6171850>, [Online; hämtad Februari 18, 2021].

- [25] I. T. L. Wen-Hong Zhu, Senior Member, “Velocity estimation by using position and acceleration sensors,” <https://ieeexplore-ieee-org.e.bibl.liu.se/stamp/stamp.jsp?tp=&arnumber=4292361>, [Online; hämtad Februari 17, 2021].
- [26] H. H. S. Liu and I. Grantham K. H. Pang, Member, “Accelerometer for mobile robot positioning,” <https://ieeexplore-ieee-org.e.bibl.liu.se/stamp/stamp.jsp?tp=&arnumber=924763>, [Online; hämtad Februari 23, 2021].
- [27] W. H. H.-i. K. J.-h. K. C. L. K.-s. K. Muhammad Latif Anjum, Jaehong Park and D. il “Dan” Cho, “Sensor data fusion using unscented kalman filter for accurate localization of mobile robots,” <https://ieeexplore-ieee-org.e.bibl.liu.se/stamp/stamp.jsp?tp=&arnumber=5669779>, [Online; hämtad Februari 23, 2021].
- [28] “Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module,” <http://www.da.isy.liu.se/vanheden/pdf/LSM303DLHC.pdf>, [Online; hämtad Februari 18, 2021].

G.2 Reglering

Förstudie inom reglering

Gunnar Arctaedius & Lovisa Byman

19 april 2021

Version 1.1



Status

Granskad	Gunnar Arctaedius, Lovisa Byman	2021-02-25
Godkänd	Gunnar Arctaedius, Lovisa Byman	2021-02-25

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedi		gunar889@student.liu.se

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-02-25	Första utkast	GA, LB	GA, LB
1.0	2021-04-01	Första version	GA, LB	GA, LB
1.1	2021-04-16	Språklig rättning och mindre tekniska förtydliganden	GA, LB	GA, LB

INNEHÅLL

Figurer	V
1 Inledning	1
1.1 Syfte	1
1.2 Metod och disposition	1
2 Problemformulering	2
3 Litteraturstudie av reglering	2
3.1 Metoder för generering av referenssignal	2
3.2 Reglering	4
3.3 Följning av genererad referenssignal	5
4 Reglering av robot	6
4.1 Val av metod för generering av referenssignal	6
4.2 Val av reglermetod	6
4.3 Realisering av PD-regulator	7
4.4 Val av reglerparametrar	10
5 Slutsats och diskussion	10
Referenser	12

FIGURER

Figur 1	Bana genererad av pure pursuit.	3
Figur 2	Två steg av point follow control-metoden.	4
Figur 3	Skiss över roboten vid rotation.	8
Figur 4	JSP-diagram över koden som styr roboten.	9
Figur 5	Användning av hastigheten och vinkelns derivata vid körsättning i korridor.	11

1 INLEDNING

Denna förstudie är en del av kursen TSEA56 - Kandidatprojekt i Elektronik. I kursen ska en autonom robot utvecklas som ska kunna kartlägga en labyrint, hitta en nödställd i labyrinten, hämta en förnödenhet utanför labyrinten och därefter åka kortaste vägen till den nödställda, lämna förnödenheten och återvända ut ur labyrinten. Roboten kommer att bestå av tre moduler: en sensormodul, en kommunikationmodul och en styrmodul.

Styrmodulen kommer bland annat att styra roboten autonomt i ett antal olika styrmoder som behövs för att slutföra det givna uppdraget. Dessa styrmoder är följande:

- Köra längs mitten av en korridor utan att avvika åt sidorna.
- Rotera 90 grader runt mittaxeln.
- Följa en tejpremsa.
- Greppa ett föremål med en gripklo.

Med dessa styrmoder kommer roboten kunna köra genom labyrinten på ett önskvärt sätt och utföra alla de givna uppdragen. Roboten är begränsad till att röra sig inom en labyrint uppbygd av korridorer eller längs med en tejpremsa, och studien avgränsas därfor till att endast styra roboten i en sådan miljö. Roboten har fyra hjul som inte går att rotera, och styrsignalerna till roboten är två pulsbreddsmodulerade signaler, en som styr robotens vänstra hjulpar och en som styr robotens högra hjulpar. På roboten kommer det att finnas ett antal sensorer, nämligen ett gyro, fem avståndsmätare och en reflexsensormodul. Avståndsmätarna kommer att vara placerade så att en är riktad framåt, två åt höger och två åt vänster. Reflexsensormodulen kommer att vara placerad längst fram på roboten och riktad nedåt för att kunna registrera tejpmärkningar på marken. Gyrot kommer att mäta robotens vinkelhastighet.

Sensorerna som används och deras placering utgör ytterligare en begränsning; roboten vet inte var den ska svänga innan den är framme vid korsningen den ska svänga i. Detta innebär att roboten inte kommer att åka framåt samtidigt som den ska rotera i en korsning.

1.1 Syfte

Studien ska i huvudsak ge en kunskapsbas till regleringen i styrmodulen, som kommer att krävas för att få roboten att bete sig på önskvärt sätt. Syftet med förstudien är således att ta fram och diskutera olika metoder för att generera en referenssignal och för att följa den genererade referenssignalen, samt att ta beslut gällande vilka metoder för reglering som kommer att implementeras i styrmodulen.

1.2 Metod och disposition

För att uppnå målet med studien ställs frågeställningarna i avsnitt 2. För att besvara dessa frågor görs först en litteraturstudie där flera artiklar och arbeten inom relevanta områden sammanfattas och presenteras i avsnitt 3. Därefter betraktas studiens frågeställningar och med hjälp av informationen från källorna undersöks lösningar till dessa vilka presenteras i avsnitt 4. Dessa resultat sammanfattas i avsnitt 5 och slutsatser dras kring vilken lösning som kommer

väljas för kandidatprojektet. En diskussion om potentiella förbättringar av den valda reglermetoden kommer även att tas upp.

2 PROBLEMFORMULERING

För att roboten ska kunna köra enligt de styrmoder som är definierade i avsnitt 1 kommer följande frågeställningar att behandlas i rapporten:

- Vilken information och mätdata behövs för att kunna utföra styrningen i de olika styrmoderna?
- Hur kan en referenssignal genereras i de olika styrmoderna?
- Hur ska roboten regleras för att kunna följa de genererade referenssignaler?
- Hur ska reglerparametrar väljas?

3 LITTERATURSTUDIE AV REGLERING

I detta kapitel undersöks flera olika källor om generering av en referenssignal och om reglering. Dessa publikationer kommer att sammanfattas och presenteras för att skapa en kunskapsbas som används senare i projektet för att besvara studiens frågeställningar.

3.1 Metoder för generering av referenssignal

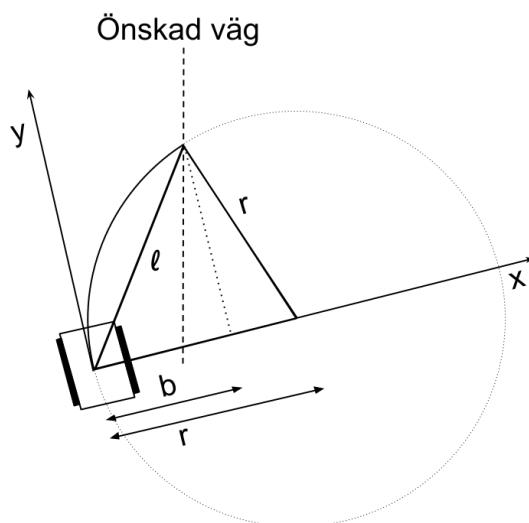
För att få ett fordon att följa en linje krävs en plan för hur den ska ta sig till linjen och därefter följa den på bästa sätt. Det kan göras genom att skapa en referenslinje att följa som utgår från fordonets position. För att generera sådana referenslinjer finns ett antal olika metoder, där nedanstående avsnitt behandlar två av dom.

3.1.1 *Pure pursuit*

Ett sätt att ta fram en referenslinje är att försöka efterlikna metoderna människor använder när de kör bil. Då hamnar bilen ofta en bit ifrån mitten av vägen, och föraren tar sig tillbaks till mitten genom att sikta in sig på en punkt längre fram. En metod som försöker hämta detta beteende kallas *pure pursuit* [1]. Metoden går ut på att fordonet försöker ta sig till en punkt en sträcka l fram på vägen, där l är en reglerparameter som kan ändras för att ändra hur fordonet uppför sig. En svängvinkel kan sedan beräknas för att få fordonet att följa en optimal cirkelbåge vilken används som referenslinje. Figur 1 visar cirkelbågen som skapas av metoden.

Cirkelbågens radie r kan beräknas genom ekvationen

$$r = \frac{l^2}{2b},$$



Figur 1: Banan genererad av pure pursuit. Cirkelbågen visar den planerade banan från robotens nuvarande position till den önskade positionen som är på den önskade vägen och på avståndet l från roboten. r visar radien på cirkelbågen, och b är avståndet till den önskade punkten i x -led i bilens koordinatsystem. Cirkeln som definierar cirkelbågen bestäms av att cirkelns mittpunkt är längs x -axeln i bilens koordinatsystem, samt att cirkeln går igenom både bilens mittpunkt och den önskade punkten.

där b är avståndet till den önskade punkten i robotens x -led. Upprepas den här processen flera gånger under körsättning så kommer fordonet ta sig till den önskade vägen på ett snabbt och mjukt sätt, under förutsättningen att fordonet lyckas följa referenslinjen.

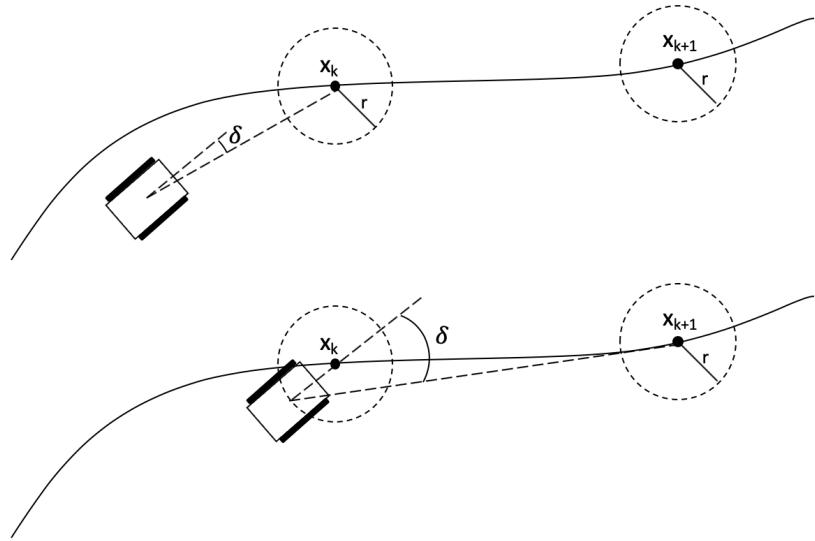
3.1.2 Point follow control

En annan metod för att generera en referenslinje baserat på en punkt en bit fram på linjen som ska följas är point follow control, vilken beskrivs i [2].

Point follow control-metodens första steg är att diskretisera linjen som ska följas till ett antal punkter. Styrningen går sedan ut på att beräkna vinkeln som fordonet ska köra för att komma tillräckligt nära nästa diskreta punkt, där tillräckligt nära definieras som en cirkel kring punkten med en fix radie, r , där r är en reglerparameter. När det kravet är uppfyllt och fordonet är tillräckligt nära punkten regleras fordonet istället mot nästa punkt på linjen för att komma tillräckligt nära den. I figur 2 visas två steg av metoden.

Ett problem med metoden är att styrvinkelns som beräknas ändras diskontinuerligt vid varje ny beräkning. Om fordonet rör sig framåt kan inte styrvinkelns ändras diskontinuerligt, så den framtagna referenslinjen kan inte följas exakt.

Det finns ett antal sätt att möjliggöra förbättringar av metoden och få en bättre följd. Ett sätt är att ta flera på varandra följande punkter i åtanke vid beräknandet av styrvinkelns, vilket kan ge en mer följsam reglering. Det är även möjligt



Figur 2: I figuren visas två steg av point follow control. x_k och x_{k+1} är två av de diskreta punkterna som den önskade linjen har delats upp i. I första steget är målet att ta sig till punkt x_k , vilket görs med styrvinkel δ . När bilen är på avståndet r från x_k beräknas en ny styrvinkel δ med målet att ta sig till punkt x_{k+1} .

att utföra en tätare diskretisering av referenslinjen, samt att ändra storleken på r för att skapa ännu mjukare styrning som fortfarande följer linjen tillräckligt noga.

3.2 Reglering

Då en referenssignal har genererats måste den också kunna följas för att vara till användning. För att lyckas med det används reglermetoder, där en sådan är PID. Denna metod tas upp i detalj i följande avsnitt.

3.2.1 PID-reglering

En PID-regulator består av tre olika delar, en proportionell del (P), en integrerande del (I) och en deriverande del (D), vilket beskrivs i [3]. I alla tre delarna används skillnaden mellan utsignalen och referensignalen, vanligen kallat felet, för att justera styrsignalen för systemet. P-delen av regulatorn resulterar i att styrsignalen ändras proportionellt mot felet. Problemet med att enbart använda P-delen av regulatorn är att det kommer att leda till ett statiskt fel på utsignalen om det finns konstanta störningar i systemet. För att motverka det kan den integrerande delen användas, då I-delen消除ar statiska fel. Den deriverande delen av regulatorn tar hänsyn till derivatan av felet för att justera styrsignalen. Det leder till en snabbare och bättre fölning av referensignalen. Vid brusiga mätsignaler kan dock derivatan påverkas mycket av bruset, vilket kan ge felaktig reglering av systemet.

En utökad PID-regulator används i [2] för att reglera ett fordon till att följa en referenslinje. Med en utökad PID-regulator kan en styrvinkel beräknas utifrån det vinkelräta avståndet mellan mitten av ett fordons styraxel och refe-

renslinen. Utökningen av PID-regulatorn består av ett filter för insignalen till D-delen och ett system som motverkar integratoruppvridning för I-delen.

I en annan publikation, [4], används P-reglering för att styra ett autonomt fordon med differentiell styrning. Även här tas PID-reglering upp som ett alternativ för att reglera styrningen med större noggrannhet. I båda dessa fall antas att avvikelsen från en referenssignal kan mätas, då den används vid regleringen.

3.2.2 Val av reglerparametrar

En metod för att välja reglerparametrar till en PID-regulator som inte kräver avancerade beräkningar eller modeller är Ziegler-Nichols metod, vilken tas upp i [5]. För att beskriva PID-regulatorn används följande form,

$$u(t) = K \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right),$$

där $u(t)$ anger styrsignalen till systemet och $e(t)$ är det aktuella felet. Parametrarna är K , T_I och T_D som finns i ekvationen, där K är förstärkningen.

Metoden går ut på att först nollställa $\frac{1}{T_I}$ respektive T_D , för att sedan öka förstärkningen K tills dess att systemet hamnar i ett stabilt oscillerande läge. Ett för lågt värde på K leder till att oscillationerna dör ut, och ett för högt värde gör att oscillationerna istället växer mot oändligheten. Det värde på K som ger ett stabilt oscillerande läge benämns K_u . Periodtiden mäts och benämns T_u . Utifrån dessa två uppmätta värden kan reglerparametrarna sättas enligt följande beräkningar:

$$\begin{aligned} K &= 0.6 \cdot K_u, \\ \frac{1}{T_I} &= \frac{2}{T_u} \text{ och} \\ T_D &= \frac{T_u}{8}. \end{aligned}$$

Författarna hävdar att dessa inställningar ger en bra reglering av de flesta system samtidigt som det håller systemet stabilt. De fortsätter med att ge förslag på finjusteringar för parametrarna utifrån experiment på det undersökta systemet. Om systemet beter sig stabilt kan förstärkningen K höjas tills systemet får önskad snabbhet utan att bete sig instabilt, och författarna menar att denna parameter kan höjas ända till K_u i vissa system. Om systemet däremot beter sig instabilt och oscillerar mer än önskvärt kan K eller T_D långsamt sänkas eller T_I ökas tills systemet når en stabil nivå.

3.3 Följning av genererad referenssignal

För att optimalt kunna styra ett system krävs som tidigare nämnts både någon form av referenssignalsgenerering, och en reglermetod för att följa referenssignalen. I detta avsnitt undersöks de två tidigare nämnda referenssignalsgenereringmetoderna i samband med PID-reglering.

3.3.1 Pure pursuit och PID

En obemannad golfbil utrustades i [6] för att kunna följa en önskad bana autonomt. Pure pursuit användes för att bestämma en referenslinje. I ett försök användes endast den genererade referenslinjen för att bestämma rattvinkeln,

och i ett annat försök användes PID-reglering för att följa den genererade referensvinkeln. Resultatet visade att bilen lyckades följa banan bra med enbart pure pursuit, men att felet minskade när även PID-reglering användes.

3.3.2 Point follow control och PID

I [2] har point follow control och utökad PID-reglering kombinerats och testats dels var för sig, och dels tillsammans. Då metoderna användes var för sig visade resultatet att fordonet väl följer en önskad linje då denna linje är rak. Om linjen som ska följas istället svänger och enbart point follow control används befann sig fordonet strax innanför kurvan. Samma test men med enbart den utökade PID-regulatorn visade att fordonets position var strax utanför kurvan. Genom att kombinera de båda metoderna uppnåddes en mycket bra följsamhet även då linjen som skulle följas svängde.

4 REGLERING AV ROBOT

Följande kapitel kommer att behandla genereringen av en referenssignal och regleringen av roboten i de olika styrmoderna utifrån informationen i avsnitt 3.

4.1 Val av metod för generering av referenssignal

I avsnitt 3.1 förklarades metoderna point follow control och pure pursuit som används för att generera en referenssignal. Ett problem med båda metoderna är att styrvinkeln ändras diskontinuerligt, vilket inte är möjligt med roboten som används i projektet om den samtidigt rör sig framåt. Det leder till att den genererade referensignalen inte kommer kunna följas exakt. För pure pursuit kan diskontinuiteten minskas genom att generera en ny referenssignal ofta, för att på så sätt få väldigt små ändringar för styrvinkeln. Även för point follow control kan diskontinuiteten minskas men då genom att diskretisera linjen som ska följas med tätare intervall. Ett problem som dock kan uppstå om punkterna ligger allt för tätt är att för lite hänsyn tas till linjens framtida bana, och planeringen sker till en punkt alltför nära roboten.

En liknande avvägning finns med pure pursuit, där en avvägning måste göras för att välja ett bra avstånd l till nästa önskade punkt. Väljs avståndet för stort kan det leda till dålig följning av den önskade vägen, men väljs avståndet för kort kan robotens rörelse bli oscillativ. Men med ett bra värde på l kan pure pursuit ge väldigt jämn följning av den önskade vägen, som på ett snabbt och smidigt sätt tar roboten dit den ska.

Ytterligare ett problem som kan uppstå med point follow control är att kravet på att robotens avstånd från punkten x_k ska vara mindre än r aldrig uppfylls. Det leder till att systemet blir instabilt och inte fungerar. Problemet kan lösas genom att öka radien r , men det leder också till att större differenser från linjen som ska följas tolereras.

4.2 Val av reglermetod

Målet med regleringen är att klara av att styra roboten i de olika styrmoderna listade i avsnitt 1. För att göra det behöver gripklon, robotens hastighet och robotens position regleras.

4.2.1 Reglering av gripklon

Med hänsyn till projektets omfattning kommer gripklon att regleras med öppen styrning, och ingen återkoppling kommer att användas. Istället kommer gripklon att slutas med en spänning som alltid är lika stor, och som anpassats

för att få önskad storlek på öppningen. Vad denna spänning ska vara kommer att mätas upp i förväg och vara baserat på storleken på föremålet som ska plockas upp.

4.2.2 Hastighetsreglering

Målet för roboten är att ta sig igenom labyrinten så snabbt som möjligt. Roboten antas ha så bra friktion mot underlaget att den inte slirar och att den stannar väldigt snabbt då styrsignalerna sätts till noll. På grund av det kommer hastigheten att regleras med hjälp av öppen styrning, med maximal motorspänning då roboten ska köra rakt. Då roboten ska rotera runt sin egen mittaxel kommer hastigheten att bestämmas av PD-reglering, som tar fram en styrsignal som bestämmer rotationshastigheten beroende på hur mycket mer roboten behöver rotera för att nå sitt mål. Se avsnitt 4.3 för detaljer kring PD-regleringen.

4.2.3 Positionsreglering

Som återgivet i avsnitt 3.2 är PID-reglering en metod som kan användas för att reglera roboten då felet relativt en referenssignal går att mäta. Oberoende av vilken av de två metoderna för generering av referenssignal vid rak körning i korridor som diskuterades i avsnitt 4.1 så kan felet från referenssignalen beräknas om avståndet från roboten till korridorens väggar samt robotens vinkel i förhållande till väggarna mäts.

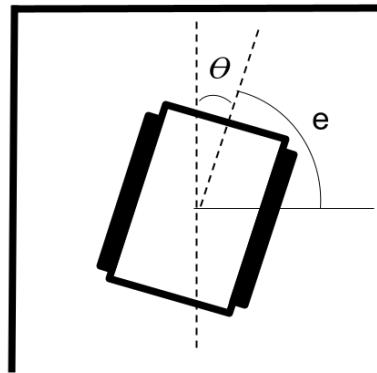
I avsnitt 1 beskrivs de sensorer som roboten är utrustad med. Både avståndet till väggarna och robotens vinkel i korridoren går att mäta med hjälp av de fyra avståndssensorerna på sidan av bilen. Avvikelsen från mitten av en tejpremsa kan mäts med hjälp av reflexsensorn. När roboten roterar mäts vinkelhastigheten med hjälp av gyrot och utifrån vinkelhastigheten kan vinkeln beräknas. Vinkeln kan sedan användas för att beräkna felet från den önskade vinkeln på 90 grader (se figur 3).

I de tre styrmoderna som är kopplade till körningen, alltså köra längs mitten av en korridor, följa en tejpremsa och rotera 90 grader runt mittaxeln finns möjligheten att mäta ett fel relativt en referensignal, vilket gör det möjligt att utföra regleringen med hjälp av PID-reglering. Eftersom PID-reglering ger bra följdning vid rätt val av reglerparametrar samtidigt som metoden är relativt enkel är metoden ett bra alternativ för projektet.

Som förklarat i avsnitt 3.2.1 är syftet med den integrerande delen av PID-regleringen att eliminera statiska fel vid störningar på systemet. Eftersom roboten rör sig i en begränsad miljö, där till exempel golvet inte lutar och roboten inte påverkas av vind eller andra yttre faktorer, så antas det statiska felet för styrmoderna listade i avsnitt 1 vara nära noll. Det innebär att den integrerande delen av regulatorn kan prioriteras bort. Däremot anses både P-delen och D-delen vara nödvändiga för tillräckligt snabb och följsam reglering och därför kommer PD-reglering att användas.

4.3 Realisering av PD-regulator

För att skapa en PD-regulator som kan styra roboten skrivs ett program i C som tar emot mätvärden från sensorer som kan mäta avstånd till väggarna på sidorna och framåt, samt hastighet framåt och vinkelhastigheten på roboten. I fallet då roboten följer en tejpremsa finns även möjlighet för att mäta avståndet till mitten av tejpremsan. Från dessa beräknas ett fel samt derivatan av felet vilka används för att ta fram en styrsignal till motorerna. Som nämnts i avsnitt 1 består roboten av fyra hjul, där det går att styra motorspänningen på varje sidas hjulpar. Styrsignalen används alltså för att sätta motorspänningen på båda hjulparen, så att roboten svänger in sig efter referensignalen. JSP-diagramet i figur 4 ger en skiss på hur ett sådant program skulle kunna se ut. Nedan visas en algoritm som använder PD-reglering



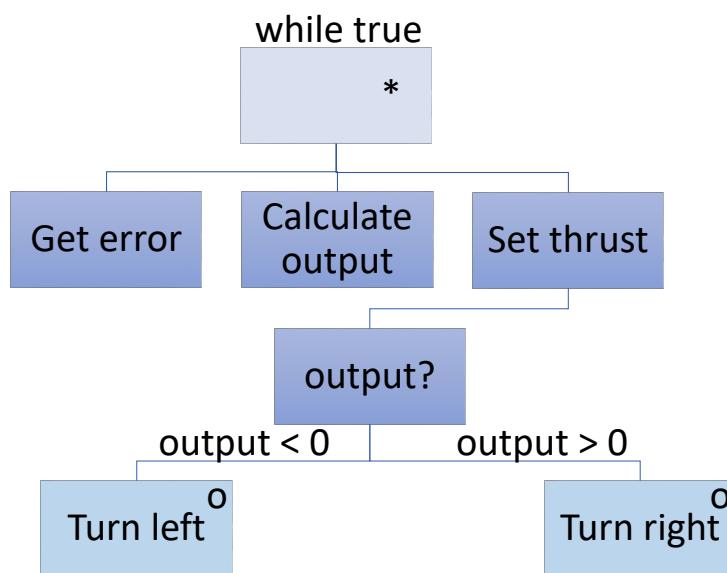
Figur 3: Skiss över roboten vid rotation. Vinkeln från referenslinjen, felet, betecknas e , och vinkeln som roboten har roterat betecknas θ .

med parametrarna K_p och K_d för att beräkna en styrsignal u utifrån referenssignalen r och mätvärdet y . De första två raderna beräknar felet x och derivatan av felet \dot{x} med hjälp utav det tidigare beräknade felet x_{n-1} samt samplingstiden T_s .

Algorithm 1: Beräkna styrsignal

$$\begin{aligned} x &= r - y; \\ \dot{x} &= \frac{x - x_{n-1}}{T_s}; \\ u &= K_p * x + K_d * \dot{x}; \end{aligned}$$

Styrsignalen u används sedan i nedanstående algoritm för att bestämma motorspänningen på de två sidorna på roboten. Olika algoritmer körs beroende på vilken styrmod systemet befinner sig i, och algoritmen nedan används vid rak körning i korridorer. I denna styrmod roterar båda hjulpare framåt, vilket anges genom att ändra värdet på rotationsvariablerna d_l och d_r . Om d_l sätts till 1 kommer det vänstra hjulparet att rotera framåt, och om variabeln sätts till 0 kommer det att rotera bakåt. d_r fungerar på motsvarande vis men för höger hjulpar. Som nämnts i avsnitt 4.2.2 så hålls hjulparen vid så hög spänning som möjligt, där motorernas maximala spänning benämns x_{\max} . I algoritmen sätts



Figur 4: JSP-diagram över koden som styr robotten. Rutorna representerar funktionsanrop, while-loopar, och if-satser. Vid if-satsen som jämför output syns längs axlarna vilket värde på output som leder till vilket funktionsanrop.

därför motorspänningen till max på ena hjulparet, och sänks från max i förhållande till storleken på styrsignalen på andra hjulparet. x_l och x_r representerar motorspänningen för vänster respektive höger hjulpar.

Algorithm 2: Rak körning i korridor

```

/* Sätt båda hjulparet till att rotera framåt */
dl = 1;
dr = 1;
if u > 0 then //sväng höger
  xl = xmax - u;
  xr = xmax;
else //sväng vänster
  xr = xmax - u;
  xl = xmax;
  */

```

Något som inte visas i algoritmen är fallet då styrsignalen har större belopp än den maximala spänningen x_{max} , men i detta fall kan det ena hjulparet helt enkelt stängas av. Snabbare svängar än så förväntas aldrig behöva utföras när robotten kör rakt i en korridor. I fallet då robotten ska svänga används följande algoritm, där motorerna sätts till att åka åt olika håll beroende på vilket håll robotten ska svänga åt genom att ändra värdet på rotationsparametrarna. Båda hjulparen hålls vid samma spänning för att undvika att robotten flyttar åt något håll. Styrsignalen kan då användas för

att bestämma hastigheten på båda hjulparen, så att roboten roterar snabbt när den har långt kvar, och saktar in när den närmar sig den önskade vinkeln.

Algorithm 3: Rotation

```

 $x_l = abs(u);$ 
 $x_r = abs(u);$ 
if  $u > 0$  then //sväng höger
   $d_l = 1;$ 
   $d_r = 0;$ 
else //sväng vänster
   $d_l = 0;$ 
   $d_r = 1;$ 
  
```

4.4 Val av reglerparametrar

I avsnitt 3.2.2 diskuterades Ziegler–Nichols metod för att finna lämpliga reglerparametrar till en PID-regulator. Men som diskuterades i avsnitt 4.2 kommer roboten som ska regleras i detta fall att endast använda en PD-regulator, vilket gör den tidigare betraktade metoden olämplig. Den kan dock fortfarande användas men med parametern $\frac{1}{T_I}$ satt till 0 då integrerande del saknas. Eftersom detta inte är användningsområdet för Ziegler–Nicholsmetoden lär justeringar behöva göras. Dessa kan göras för hand, vilket är möjligt eftersom reglerparametrarna för roboten kommer kunna ändras av användaren. Parametrarna får således justeras tills de bedöms reglera roboten så pass bra att den uppfyller de ställda kraven utan att bete sig instabilt.

5 SLUTSATS OCH DISKUSSION

För att få roboten att köra som önskat i de olika styrmoderna kommer roboten att regleras olika beroende på styrmod. När roboten kör rakt i en korridor kan pure pursuit eller point follow control användas för att skapa en referensignal och PD-reglering användas för att följa den framtagna referensignalen. Alternativt kan PD-reglering användas med mittlinjen av korridoren som referenssignal, utan att först generera någon referenssignal. Gripklon kommer att regleras med öppen styrning. Vid rotation kommer båda hjulparen ha samma motorspänning, men riktningen kommer att vara motsatt. Storleken på motorspänningen styrs med hjälp av PD-reglering. Vid tejpfoljning kommer differensen mellan spänningarna på hjulparen att bestämmas med hjälp av PD-reglering, men spänningen detta utgår ifrån kommer att vara konstant.

Då PD-reglering används behöver felet från referensignalen mätas, vilket innebär att avvikelse från mitten av en tejpremsa behöver mätas för att kunna följa tejp, vinkeln behöver mätas för att kunna rotera 90 grader och avståndet till väggarna i korridorerna måste mätas för att kunna köra rakt. Valet av reglerparametrar kommer att utgå från Ziegler–Nichols metod, men parametrarna kommer utöver det att behöva justeras tills regleringen fungerar på önskvärt sätt.

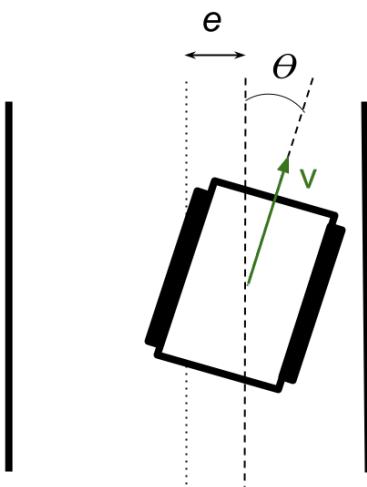
Hur användbara metoderna för att generera en referenssignal kommer att vara är inte säkert. Att använda den önskade linjen som referensignal med PD-reglering kan ge tillräckligt bra körlinje i rak korridor. Om detta inte är fallet går det att välja en av de två diskuterade metoderna för att generera en referenssignal i kombination med PD-regulatorn, något

som diskuterades i avsnitt 3.3. Vid behov kan även PD-regleringen komma att förbättras. Ett möjligt sätt att göra det är att använda en PID-regulator istället för en PD-regulator. Denna lösning är speciellt användbar om det framkommer att det statiska felet inte är noll. Ett annat sätt att förbättra regleringen är att använda mer information och mätdata. Vid körning i korridor kan till exempel derivatan av felet tas fram mer noggrant om vinkelns och hastigheten används (se figur 5). Istället för att beräkna derivatan av felet som differensen mellan två efterföljande mätningar av felet, som föreslaget i avsnitt 4.3, kan en mer exakt derivata beräknas. Derivatan av felet blir då

$$\dot{e}(t) = v \cdot \sin\theta,$$

där v är robotens hastighet framåt. För att beräkna derivatan på detta sätt kan hastigheten antingen approximeras utifrån motorspänningen på motorerna eller mäts med hjälp av en sensor.

Som tidigare nämnts i avsnitt 3.2.1 är ett problem med den deriverande delen av regulatorn att mätbrus kan påverka derivatan på ett icke önskvärt sätt. Skulle detta vara ett problem kan insignalen till den deriverande delen av regulatorn lågpassfiltreras för att minska det högfrekventa bruset på signalen.



Figur 5: Roboten i en korridor. Derivatan av felet kan tas fram med hjälp av hastigheten och vinkelns från mittlinjen.

REFERENSER

- [1] R. C. Conlter, "Implementation of the pure pursuit path tracking algorithm," The Robotics Institute, Carnegie Mellon University, Tech. Rep., 1992. [Online]. Available: http://www.enseignement.polytechnique.fr/profs/informatique/Eric.Goubault/MRIS/coulter_r_craig_1992_1.pdf
- [2] J. K. Kolb, G. Nitzsche, and S. Wagner, "A simple yet efficient path tracking controller for autonomous trucks," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 307–312, 2019, 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019. [Hämtat 17 februari 2021]. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319304227>
- [3] T. Glad and L. Ljung, *Reglerteknik, Grundläggande teori*, 4:16 ed. Lund: Studentlitteratur, 2006, pp. 17–20, ISBN 978-91-44-02275-8.
- [4] Mohd Saifizi Saidonr, H. Desa, and M. N. Rudzuan, "A differential steering control with proportional controller for an autonomous mobile robot," in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, 2011, pp. 90–94, [Hämtat 15 februari 2021]. [Online]. Available: <https://ieeexplore-ieee-org.e.bibl.liu.se/document/5759849>
- [5] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 220–222, 1942.
- [6] Y. Huang, Z. Tian, Q. Jiang, and J. Xu, "Path tracking based on improved pure pursuit model and PID," in *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT*, 2020, pp. 359–364.

G.3 Kommunikation

Förstudie i Busskommunikation

Edvin Leuchovius, Filip Johansson

15 april 2021

Version 1.1



Status

Granskad	FJ, EL	2021-04-15
Godkänd	Kent	2021-04-15

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedi		gunar889@student.liu.se

FÖRKORTNINGAR

Förkortning	Betydelse
UART	Ett gränssnitt som konverterar parallell data till seriell data och vice versa.
master	Den enhet på en buss som bestämmer vilken enhet som får skicka data.
slave	En enhet på en buss som ej kan bestämma när den får skicka information på bussen.

FIGURER

Figur 1	Ett exempel av hur data som skickas över UART kan se ut.	4
Figur 2	Ett exempel av koppling med en master och två slavar med individuell ”Slave select”. ”MOSI” betyder masters utgående dataledning. ”MISO” betyder masters ingående dataledning. SCLK är klockan. ”SDI” betyder slavens utgående dataledning. ”SDO” är slavens ingående dataledning.	5
Figur 3	Ett exempel av koppling med en master och två slavar i ”Daisychain” form. ”MISO” betyder masters ingående dataledning. SCLK är klockan. ”SDI” betyder slavens utgående dataledning. ”SDO” är slavens ingående dataledning.	6
Figur 4	Förhållandet mellan totala antalet byte och antalet databyte (nyttiga byte).	8

INNEHÅLL

Förkortningar	III
Figurer	IV
1 Inledning	1
1.1 Bakgrund	1
1.2 Problemformulering	1
1.3 Avgränsningar	1
2 Metod	1
3 Teori	1
3.1 Dataöverföring	2
3.2 Seriell kommunikation	3
3.3 Parallel kommunikation	6
3.4 Processorer	6
4 Diskussion	7
4.1 Jämförelse av overhead	7
4.2 Jämförelse av skalbarhet	8
4.3 Jämförelse säkerhet	9
4.4 Jämförelse av hastigheter	9
5 Slutsatser	10
Referenser	11
A Kravspecifikation	12

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2021-02-26	Första utkast	EL , FJ	EL, FJ
1.0	2021-04-08	Andra utkast	EL , FJ	EL, FJ
1.1	2021-04-15	Andra utkast	EL , FJ	EL, FJ

1 INLEDNING

1.1 Bakgrund

Denna rapport är en förstudie som ämnar att bistå med bakgrundsinformation om kommunikationen mellan processorer. Underlaget ska kunna användas för att underlätta designval inom projektet som utförs i kursen ”Kandidatprojekt i elektronik” vid Linköpings universitet. Målet med projektet är att konstruera en självkörande bil, en så kallad ”*undsättningsrobot*”, som kan utforska en labyrinth.

Roboten kommer att bestå av flera processorer, en för varje av tre stycken delsystem, så kallade ”*moduler*”, som måste kunna kommunicera med varandra. Dessa är: en kommunikationsmodul, en styrmodul och en sensormodul (se kravspecifikationen, Appendix A). Det är därför relevant att undersöka hur kommunikation mellan olika processorer kan implementeras.

1.2 Problemformulering

Förstudien ämnar att undersöka olika kommunikationsmetoder för processorer. Mer specifikt syftar detta i praktiken på dataöverföring mellan processorer, vilket är ett omfattande område inom datateknik. Det finns ett stort urval av protokoll för kommunikation, så det är av stor vikt att jämföra dessa rationellt.

- Vilka olika typer av kommunikationsprotokoll finns, relevanta för projektet?
- Hur fungerar de olika kommunikationsprotokollen, och hur kan de implementeras i hårdvara?
- Hur står sig de olika protokollen i jämförelse med varandra?

1.3 Avgränsningar

Förstudien kommer framförallt att fokusera på sådan kommunikation där processorerna är fysiskt ihopkopplade med ledningar, så kallade databussar.

2 METOD

Förstudien inleds med en litteraturstudie med teori om olika kommunikationsprotokoll, varefter en jämförelse utförs mellan protokollen för att hitta för- och nackdelar.

3 TEORI

Teoriavsnittet inleds med en allmän redogörelse för hur dataöverföring kan fungera inom ett system av sammankopplade enheter och tar upp lite vanliga begrepp, principer och signaler som är vanliga. Därefter ges bakgrundsinformation för ett par olika protokoll, framförallt seriell kommunikation, men även parallell kommunikation, tas upp översiktligt.

Avslutningsvis kommer en kort redogörelse för vilket stöd som finns för dem protokoll som tagits upp av två olika processorenheter: ATmega1284P och Raspberry Pi 3+.

3.1 Dataöverföring

Det är av central betydelse att kunna skicka data mellan olika enheter av ett system på ett effektivt sätt, och eftersom det är just olika enheter som ska kunna kommunicera med varandra. Det är därför av yttersta vikt att denna kommunikation är standardisering mellan enheterna. En enhet kan generellt sett vara av väldigt spridda slag, men syftar i denna rapport på just en processorenhet såsom en mikrokontroller eller en enkortsdator.[†]

Det kan vid studium av kommunikationen mellan enheter i ett system vara lämpligt att inledningsvis titta på dataöverföring mellan endast två enheter och där efter utöka till att titta på flera enheter som kommunicerar via samma buss (parvis per överföring).

Parvis dataöverföring

Ett övergripande första sätt att dela upp principer för överföringen av information mellan två enheter är efter: *öppen överföring, återkopplad överföring och överföring via buffert* (mellanlagring) [1].

Öppen överföring, även kallad synkron överföring, är när sändaren placerar data på bussen och samtidigt, under begränsad tid, signalerar till mottagaren att data finns tillgänglig på bussen via en styrsignal, betecknad \overline{DAV} ("data available strobe")[‡]. Det är sedan upp till mottagaren att läsa den data som finns på bussen, men sändaren får ingen bekräftelse huruvida överföringen lyckades. Detta medför dessvärre att den data som skickas inte når fram till mottagaren om mottagaren ligger nere, är upptagen eller reagerar för långsamt [1].

Återkopplad överföring kan ses som en utökning av den öppna överföringen, där mottagaren dessutom signalerar tillbaka till mottagaren att data mottagits; så kallad "acknowledgement" med beteckningen \overline{ACK} . Principen kallas även handskakning, eftersom både sändare och mottagare signalerar med varandra (\overline{DAV} respektive \overline{ACK}) [1].

Den återkopplade överföring är, på grund av styrsignalen \overline{ACK} , inte lika känslig för långsamma mottagare, eftersom sändaren kan vänta på \overline{ACK} . Samtidigt resulterar detta i ett problem om \overline{ACK} från mottagaren uteblir. Lösningen är att sändaren väntar en förbestämd tid på den uteblivna signalen och sedan avbryter och samtidigt startar en avbrottsrutin. Denna tid kallas "timeout" [1].

Databussar

Databussar används för att koppla ihop flera enheter av ett system och kan därför liknas vid en motorväg [1]. Eftersom många enheter är anslutna till samma buss är det, förutom data och styrsignaler, även nödvändigt för en sändare att ange vilken mottagare som är avsedd. En lösningen är att varje enhet har en fördefinierad adress som sändaren kommunicerar till samtliga enheter via bussen, varefter enheten med samma adress kan reagera.

[†] En mikrokontroller består av processor(er), minne och I/O-möjligheter, integrerade på ett enda chip, men är just bara ett chip. En mikrokontroller behöver således monteras och anslutas, till exempel på ett kretskort, i den produkt där den ska användas. En enkortsdator är istället en komplett liten dator ("plug-and-play").

[‡] Sträcket över betecknar att signalen är aktivt lågt.

Då i allmänhet endast två enheter åt gången kan använda en buss (en sändare och en mottagare), ges enheter anslutna till samma buss olika roller med avseende på kontrollen av bussen.

Enheter som kan initiera en transaktion med en annan enhet på bussen kallas ”*bus master*”, eller bara ”*master*”, medan enheter som endast kan acceptera en transaktion på begäran kallas ”*bus slave*”, eller ”*slave*” [2]. Detta ska inte förväxlas eller likställas med sändar- respektive mottagarenheter, eftersom en enhet med master-behörighet förutom att initiera sändning av data även kan begära data från en annan enhet, som således övergår till att agera mottagare efter begäran.

Flera enheter på samma buss kan ges master-behörighet, och då är det nödvändigt att använda arbitrering (”*arbitration*”), eftersom händelsen att flera master-enheter vill använda bussen samtidigt måste kunna hanteras. Det innebär att det finns någon mekanism som hanterar vilken av masterenheterna som, vid varje tidpunkt, ska få tillgång till bussen [1]. En vanlig benämning när det finns flera master på samma buss är ”*multi-master*”.

3.2 Seriell kommunikation

Vid seriell kommunikation mellan enheter överförs datan som en sekvens av bitar. En klassisk sådan seriell buss är ”*Inter Integrated Circuit*” (I²C), uppfunden av dåvarande Phillips Semiconductor år 1982 [3], tas upp i det här avsnittet.

Innan bussar så som I²C studeras närmare är det dock viktigt att observera att den data som ska skickas (från en processor) i princip alltid är att betrakta som parallell från början, i form av ett eller flera datorord i något register eller vid någon minnesadress. Det är därmed nödvändigt, inte minst för programmeraren av den enskilda processorn, att använda någon form av gränssnitt som hanterar denna övergång och dessutom hanterar flytten av data över själva bussen. Ett standardiserat sådant gränssnitt är ”*Universal Asynchronous Receiver/Transmitter*” (UART), vilket introduceras före I²C.

3.2.1 Universal Asynchronous Receiver/Transmitter – UART

UART är en förkortning som står för ”*Universal Asynchronous Receiver/Transmitter*”, en viktiga egenskaper av UART kan utrönas redan ur delar av namnet med ”Asynchronous” menas att det inte finns någon gemensam klocka för de två UART-modulerna som kommunicerar. De sköter istället detta att synkronisera tolkning av datan genom att i förväg bestämma hastigheten på dataöverföringen, detta och den bestämda formen på datan ger möjligheten att slippa använda en gemensam klocka.

UART är en krets som tar in parallell data för att skicka den över en databuss som seriell data till en annan UART som tar emot datan seriellt och vidarebefordrar datan som paralleldata [4]. För tvåvägs kommunikation krävs två ledningar då bussarna är enkelriktade för data.

Datan som skickas har alltid en start- och stopbit där slutbit signaleras av att hålla bussen konstant hög och start av hålla bussen låg ett tidsintervall se figur 1 . Med möjligheten att välja om en paritetsbit ska användas. Om paritetsbit används signalerar en detta till mottagaren att ett jämnt antal ettor skickats och en paritetsbit som är noll signalerar att ett udda antal ettor skickats. På detta sätt kan mottagaren veta om datan förändrats under överföringen. Mellan start och paritetsbiten kan 5-8 databitar skickas. I figur 1 kan ses ett exempel där datan 11000110 överförs med paritetsbit.

Hastigheten kan väljas, vanligtvis inom intervallet från 9,6 kbit per sekund till och med 1,5 Mbit per sekund[4].



Figur 1: Ett exempel av hur data som skickas över UART kan se ut.

3.2.2 Inter Integrated Circuit – I²C

I²C är en seriell buss för tvåvägs dataöverföring, som är utvecklad av Phillips Semiconductors. Bussen består av två ledningar:

- en dataledning, vilken i litteratur och datablad refereras till som “*Serial DAta line*” (SDA)
- och en ledning för synkronisering, kallad “*Serial Clock Line*” (SCL) [2].

Bussen har fullt stöd för multi-master [2], vilket innebär att det är möjligt att ha flera enheter anslutna på bussen som kan inleda en transaktion, antingen som sändar- eller mottagarenhet.

Varje transaktion börjar således genom att en enhet med master-behörighet signalerar på SDA-ledningen vilken enhet den vill utbyta information med, samt om master-enheten vill skicka eller ta emot data från denna. Det är efter denna initiering som sändar- respektive mottagarenhet anses som definierade vid en transaktion, så därefter kan sändningen av data till mottagaren ske.

All kommunikation på SDA-ledningen under en transaktion mellan två enheter, oavsett om det är initieringen eller själva dataöverföringen, sker på samma sätt; nämligen genom att 8 bitar läggs på SDA-ledningen av en enhet, vilken sedan kommer att lyssna på ledningen efter en ACK-bit från en enhet på bussen.

Varje ansluten enhet på bussen associeras med en unik adress, vilken är 7 bitar lång. [†] Initieringen av en transaktion sker därvid genom att en master-enhet lägger en adress på SDA-ledningen, följt av en R/W-bit (“read/write”) som anger om master-enheten ska vara sändare (write) eller mottagare (read) vid den kommande transaktionen. Notera att adress + R/W-bit tillsammans är 8 bitar lång.

Om det finns en enhet med motsvarande adress, som dessutom är tillgänglig, så kommer den att signalera en ACK över SDA-ledningen tillbaka till master, varefter själva dataöverföringen kan ske.

Dataöverföringen sker sedan genom att sändaren skickar 8 bitar i taget till mottagaren tills sändaren har data att skicka, men måste kontinuerligt mellan dessa 8 bitar lyssna efter en ACK-bit från mottagaren.

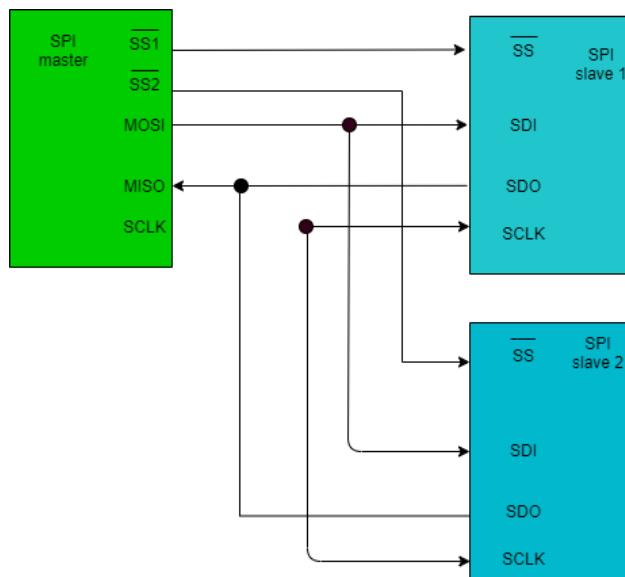
[†] Utökat 10-bitars läge stöds dock vid behov.

3.2.3 Serial Peripheral Interface – SPI

SPI är en seriell buss för tvåvägs dataöverföring som utvecklades av Motorola på 80-talet [5]. Det kräver två ledningar (“MOSI”/“MISO”) för dataöverföring då de är enkelriktade i varsin riktning, en ledning för klocka (“SCLK”) och ett antal ledningar beroende på hur “master”/“Slave” anslutningen är kopplad.

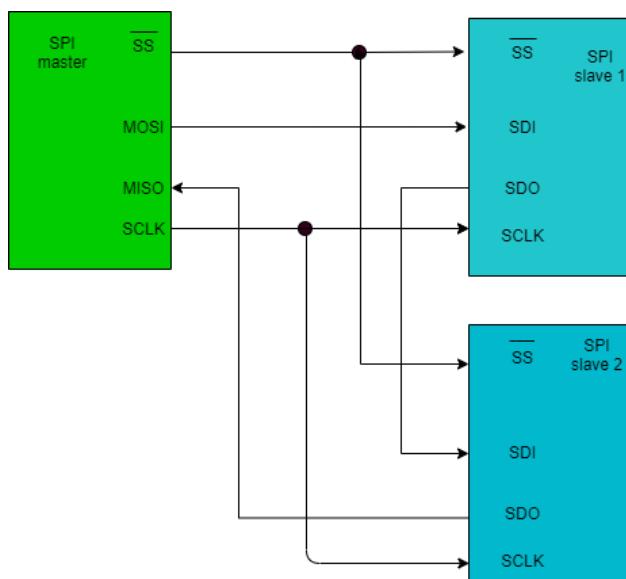
SPI är synkront, det kräver en gemensam klocka och finns ingen definierad överföringshastighet[6], utan det väljs utifrån den gemensamma klockan. Detta leder till att det finns stöd för överföringshastigheter i megabit per sekund området. Det finns ingen kontroll av att slaven mottagit informationen som skickats till den.

Det finns två huvudsakliga sätt att koppla en master (det är endast möjligt att ha en master i SPI) till flera slavenheter [7], där det ena sättet är att varje slav får en individuell ledning “Slave Select” (SS), se figur 2.



Figur 2: Ett exempel av koppling med en master och två slavar med individuell “Slave select”. “MOSI” betyder masters utgående dataledning. “MISO” betyder masters ingående dataledning. SCLK är klockan. “SDI” betyder slavens utgående dataledning. “SDO” är slavens ingående dataledning.

Den andra metoden är att koppla slavenheterna i en “Daisy Chain”[7], vilket innebär att dataledningen från master kopplas till slav nr 1, vars utgående dataledning kopplas till nästa slavens ingående dataledning (se figur 3). Detta fortsätter tills sista slaven vilken kopplar sin utgående dataledning till masters ingående dataledning.



Figur 3: Ett exempel av koppling med en master och två slavar i “Daisychain” form. “MISO” betyder masters ingående dataledning. SCLK är klockan. “SDI” betyder slavens utgående dataledning. “SDO” är slavens ingående dataledning.

3.3 Parallel kommunikation

Dataöverföring mellan enheter kan, som kontrast till seriell kommunikation, även implementeras genom att överföra flera bitar parallellt via separata dataledningar.

Antalet ledningar för seriella lösningar varierar, som tidigare nämnt, mellan dem olika protokollen. Protokoll för parallel kommunikation behöver däremot flera ledningar för att kunna överföra bitarna samtidigt.

3.4 Processorer

En processor är en den enhet som exekverar program i en dator. Detta genom att hämta lagrade instruktioner från minnet, antingen inbyggd eller externt för att sedan avkoda dessa och tillslut utföra dem.

3.4.1 AVR

AVR är en familj av mikroprocessor. Dessa är billiga enheter som kräver en låg strömtillförsel.

En av dessa är ATmega1284P, vilket är 8-bitars processor med stöd för flera av de ovan diskuterade busstandarderna [8]. Denna mikroprocessor har en maximal klockhastighet på 20 MHz och med ca en instruktion per klockcykel klarar den av 20 miljoner instruktioner per sekund.

Den har tre inbyggda SPI-moduler, en I²C och det finns även stöd för två USART(vilket är en modifierad version av UART där möjligheten finns att använda en gemensam klocka för att gör dataöverföringen synkron).

3.4.2 Raspberry Pi 3 B+

Raspberry Pi 3 B+ är en kraftfull 64-bitars dator[9]. Den har klockhastigheten 1,2 GHz. Den har två stycken SPI-moduler, två stycken I²C moduler och två stycken UART.

4 DISKUSSION

Hastigheten vid dataöverföring mellan enheter av ett system behöver minimeras, samtidigt som kostnader, fysiskt avtryck, samt framtida vidareutveckling och utbyggnad av systemet behöver vägas in vid valet av buss.

4.1 Jämförelse av overhead

Det finns flera olika sätt att se på prestandan i en överföring av data. Ett sätt är att jämföra hur många klockcykler som krävs för att överföra en byte data för respektive protokoll. I²C kräver till exempel 9 bitar vid initieringen inför varje dataöverföring och dessutom en bit för varje databyte som överförs. Om N anger antalet databytes som ska överföras, anger ekvation (1) det totala antalet bitar som funktion av antalet databytes mellan enheterna vid en överföring.

$$T_{I^2C}(N) = \underbrace{8 + 1}_{\text{Adress}+\overline{\text{ACK}}} + N \cdot \underbrace{(8 + 1)}_{\text{Data}+\overline{\text{ACK}}} \quad (1)$$

Detta motsvarar således ett overhead på $9 + 9N - 8N = N + 9$ bitar, eller $\frac{N+9}{8}$ bytes. Ett förhållande över I²C-bussens overhead (i bytes per databytes) ges då slutligen av ekvation (2).

$$q_{I^2C}(N) = \frac{\frac{N+9}{8}}{N} = \frac{1 + 9/N}{8} = \frac{1}{8} + \frac{9}{8N} \quad (2)$$

Värt att notera är att bussens effektivitet ökar med antalet byte som skickas, och att handskakningsprocessen, som är en del av det overhead som uppstår, syftar till att förbättra överföringens integritet.

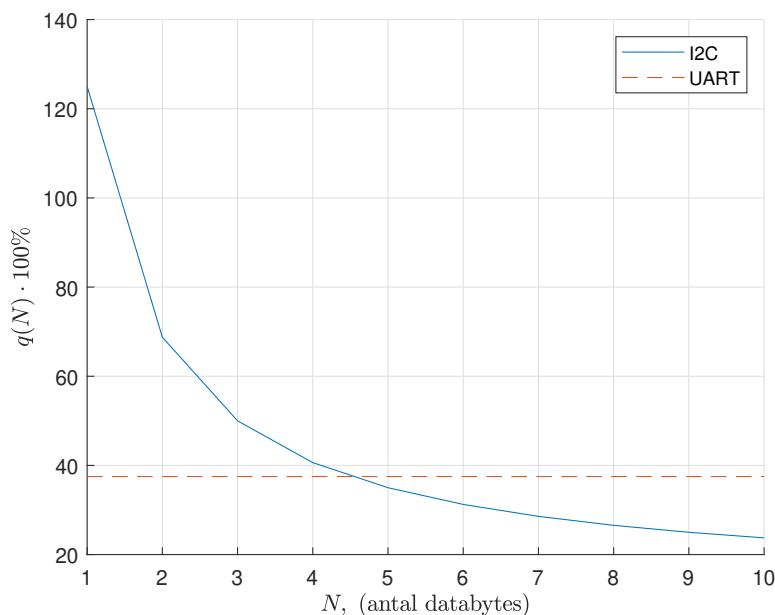
UART har också en del overhead då det vid varje dataöverföring måste komma en start- och slutbit, om paritetsbit används tillkommer ytterligare en bit som ej används för dataöverföring. För varje byte av data som överförs behövs alltså 3 st overhead bitar se ekvation (3).

$$q_{UART}(N) = \frac{3}{8} \quad (3)$$

Det är värt att notera att effektiviteten hos överföringen är konstant. Detta för varje byte som överförs måste en extra tre bitar överföras.

Figur 4 visar q_{I^2C} och q_{UART} som funktion av N .

SPI har inte heller någon motsvarighet till I²C-protokollets ACK/NACK, vilket ger att SPI i sin grundkonfiguration inte har något overhead.



Figur 4: Förhållandet mellan totala antalet byte och antalet databyte (nyttiga byte).

4.2 Jämförelse av skalbarhet

En konsekvens av att parallelportar i praktiken implementeras med ett större antal ledningar, jämfört med seriella protokoll, är att även ett större antal pinnar behövs. Huruvida detta utgör ett problem behöver avgöras från fall till fall. För till exempel en processor som ska anslutas till många enklare typer av enheter kan detta problem bli påtagligt, vilket är fallet för sensormodulen på undsättningsroboten.

SPI-protokollets motsvarighet till en adress är dess "slave select", vilket är en ledning direkt till den avsedda enheten. Detta är anledningen till att SPI saknar overhead, eftersom antalet fysiska ledningar istället ökar med fler enheter.

Ett system av enheter sammankopplade med SPI är inte heller lika skalbart som I²C, både för grund- och "Daisy Chain"-konfiguration. I grundkonfigurationen krävs många ledningar för att kunna ha de individuella "slave select", vilket innebär svårigheter att skala upp. "Daisy Chain" metoden däremot innebär att all data måste gå genom hela slavkedjan innan master kan få något svar vilket innebär en väsentlig förröjning om det finns många slaveneheter. Med en förröjning på svarsinformationen som ökar linjärt med antalet slavar som i värsta fall leda till att en processor står stilla i flera klockcykler vilket inte är bra.

När det gäller antalet ledningar som behövs för de olika protokollen är I²C det överlägsna eftersom denna endast kräver två ledningar. SPI däremot, kräver minst fyra ledningar och för varje slavenhet som läggs till måste en ledning läggas till. SPI kan teoretiskt hantera fler slavenheter än I²C kan men det kräver många "slave select"-ledningar vilket i praktiken inte är rimligt. Med fler ledningar ökar risken för växelverkan mellan ledningarna vilket kan leda till att data

på en ledning kan påverka andra ledningar. Det totala antalet ledningar kan uttryckas som ekvation 4 där L är totala antalet ledningar och M är antalet anslutna slavenheter.

$$L(M) = \underbrace{2}_{\text{Dataledningar}} + \underbrace{1}_{\text{klocka}} + \underbrace{M}_{\text{Slaveselect}} = 3 + M \quad (4)$$

I²C har ett max på slavenheter som är 128 stycken, detta då längden på adresserna är 7 bitar. För att SPI ska kunna ha fler slavenheter behövs alltså 128 "slave select"-ledningar och totalt 131 ledningar. I²C däremot har inga krav på ökade antal ledningar och kan därför hantera 131 slavenheter med endast 2 ledningar.

UART har ingen möjlighet för expansion utan är strikt mellan två enheter.

4.3 Jämförelse säkerhet

Det är inte endast överföringshastigheter och skalbarhet som är viktigt det är även viktigt att informationen når fram till sin destination. De olika protokollen har alla olika sätt att säkerställa att informationen når fram.

I²C har $\overline{\text{ACK/NACK}}$ som måste skickas mellan varje byte som överförs. Detta säkerställer att data har kommit fram det enda problemet är att det inte finns någon inbyggd kontroll av huruvida datan kom fram utan förvrängning. Detta går att implementera mjukvarumässigt men det kommer då att lägga till ytterligare overhead. Detta med handskakningen ser också till att en processor aldrig kommer att prata ut i intet, om data ska överföras måste en $\overline{\text{ACK/NACK}}$ att skickas tillbaks för att dataöverföringen ska fortsätta.

Eftersom SPI har inte någon motsvarighet till $\overline{\text{ACK/NACK}}$ kan inte masterenheten med säkerhet veta om slavenheten fungerar och kan läsa från bussen, eller om det överhuvudtaget finns några slavenheter kopplade, vilket kan leda till att den skickar ut information på en buss som ingen läser. Men även om det finns en slavenhet som är kopplad finns det inget som försäkrar att datan ej blivit förvrängd under sin färd.

UART har också ingen motsvarighet till handskakningen som I²C har så den kan prata ut i intet men den kan vara lite säker på att det åtminstone är något kopplat till andra änden då ledningen hålls hög. Paritetsbiten som kan användas är det viktigaste för att se till att data ej blir förvrängd. Den ser till att mottagaren vet med större säkerhet om antalet ettor är korrekt. Detta gör att för att förvränga data som skickas måste ett jämnt antal bitar förvrängas.

4.4 Jämförelse av hastigheter

I²C har i "fastmode" en överföringshastighet på 400 kbit per sekund. Detta till skillnad mot SPI som kan ha hastigheter i hastighetsordningen av tusentals bit per sekund. UART har dock den högsta hastigheten då den kan överföra information i 1,2 Mb/s.

5 SLUTSATSER

De olika protokollen är bäst i olika situationer. Det finns inget protokoll som alltid är bäst utan man måste ta hänsyn till vilken sorts data som ska överföras och om den är känslig för förvrängningar. Hänsyn måste också tas till hur många olika slavenheter som master vill kunna kommunicera med och hur många ledningar som är acceptabelt i situationen.

I²C är bäst då det är många enheter och stora mängder data som överförs vid varje transaktion. Men det räcker med att det är tre stycken slavenheter för att det ska vara värt att använda sig av I²C. Om antalet ledningar är något som är kritiskt för användningen är också I²C det bästa alternativet.

SPI ändå är bäst när det finns mer än en slavenhet som ska kommuniceras med färre än än tre stycken. Det kommer inte att kräva så många sladdar och det kommer inte finnas något overhead i standardkonfiguration. Om overhead är en kritisk del av systemet kan det vara en bra idé att använda SPI ändå.

UART är bäst när det är öppen dataöverföring mellan två stycken enheter det ger då en överlägsen överföringshastighet jämfört med SPI och I²C. Det är även bäst när det är viktigt att datan kommer fram oförändrad då det finns en kontroll av datan som sannolikt skulle upptäcka störningar.

För processorerna på roboten som ska konstrueras och som behöver interagera med många, eventuellt olika enheter, är det väl motiverat att välja I²C, eftersom denna buss gör det möjligt att ansluta ett närmast obegränsat antal enheter, även i efterhand. Dessutom erbjuder I²C flera mekanismer som ökar dataöverföringens robusthet och pålitlighet, genom en dedikerad klocksignal samt en handskakningsmekanism.

REFERENSER

- [1] A. Clements, *Computer Organization and Architecture — Themes and Variations*, international ed. Cengage Learning, 2014.
- [2] *The I2C-Bus Specification - Version 2.1 - Datasheet*, <http://www.da.isy.liu.se/vanheden/pdf/i2cspc2000.pdf>, Philips Semiconductors, 2000, [Online; hämtad 2021-02-19].
- [3] *I2C*, <https://en.wikipedia.org/wiki/I%C2%BA%C2%BC>, Wikipedia Foundation, [Online; hämtad 2021-02-18].
- [4] M. G. L. Eric Peña, “uart a hardware communication protocol Understanding Universal Asynchronous Receiver/Transmitter web site,” <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html#>, [Online; hämtad 2021-02-18].
- [5] K. Magdy, “SPI Tutorial With PIC Microcontrollers | Serial Peripheral Interface,” <https://deepbluembedded.com/spi-tutorial-with-pic-microcontrollers/>, [Online; hämtad 2021-02-22].
- [6] F. Leens, “An introduction to i2c and spi protocols,” *IEEE Instrumentation Measurement Magazine*, vol. 11, pp. 8–13, 2009.
- [7] P. Dhaker, “Introduction to SPI Interface web site,” <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>, [Online; hämtad 2021-02-20].
- [8] *ATmega1284p Revision 8059D Datasheet*, <http://www.da.isy.liu.se/vanheden/pdf/atmega1284p.pdf>, Atmel, 2009, [Online; hämtad 2021-02-24].
- [9] *Raspberry Pi Compute Module 3+ Datasheet*, https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf, The Raspberry Pi Foundation, 2019, [Online; hämtad 2021-02-26].

A KRAVSPECIFIKATION

Kravspecifikation

4 februari 2021

Version 1.0



Status

Granskad	ANC, JC	2021-02-04
Godkänd	Kent	2021-02-04

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedi		gunar889@student.liu.se

INNEHÅLL

1	Inledning	1
1.1	Parter	2
1.2	Syfte och mål	2
1.3	Bakgrundsinformation	2
1.4	Definitioner	2
2	Översikt av systemet	2
2.1	Grov beskrivning av produkten	2
2.2	Ingående delsystem	3
2.3	Avgränsningar	3
2.4	Generella krav på systemet	4
3	Kommunikationsmodul	5
3.1	Krav	5
4	Sensormodul	5
4.1	Krav	5
5	Styrmodul	6
5.1	Krav	6
6	Datormodul	6
6.1	Krav	6
7	Ekonomi	7
8	Leveranskrav och delleveranser	7
9	Dokumentation	7
	Referenser	8

DOKUMENTHISTORIK

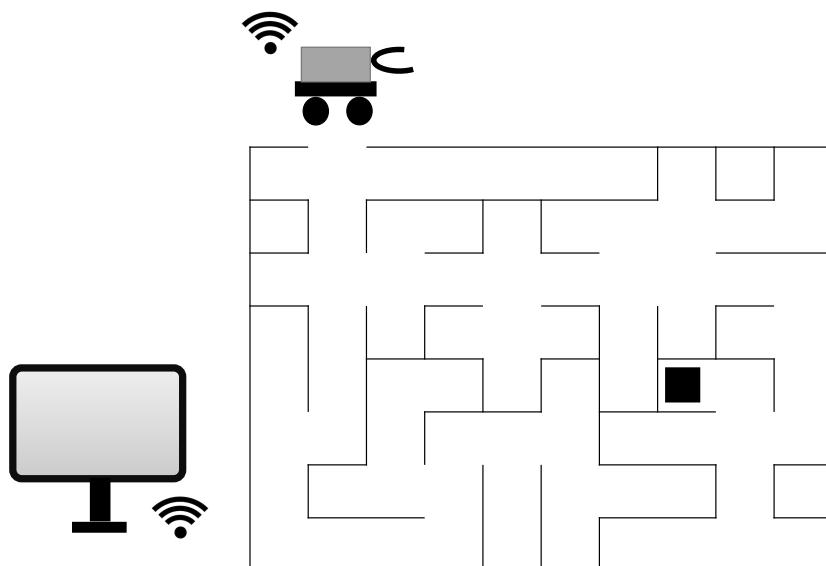
Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2021-01-28	Första utkast	Alla	Alla
0.2	2021-02-02	Andra utkast	Alla	Alla
1.0	2021-02-04	Tredje utkast	ANC, JC	ANC, JC

1 INLEDNING

Detta dokument beskriver de krav som förväntas gälla för en prototyp av en undsättningsrobot konstruerad i kursen TSEA56, kandidatprojekt i elektronik, under vårterminen 2021 vid Linköpings universitet. Kraven specificeras enligt nedan, där kravnummer anges i kolumn 1, kravets version (om det reviderats eller ej) i kolumn 2, beskrivning av kravet ges i kolumn 3, och kravets prioritet anges i kolumn 4. Prioritet 1 innebär att kravet skall vara uppfyllt vid projektets första delleverans av systemet, kallad BP5a. Prioritet 2 innebär att kravet skall vara uppfyllt vid projektets slutliga leverans, kallad BP5b. Prioritet 3 innebär att kravet inte är nödvändigt för slutleverans; dessa utförs i mån av tid.

Krav	Version	Beskrivning	Prioritet
------	---------	-------------	-----------

Figur 1 visar en förenklad bild av roboten i situationen för uppdraget. För specifikationer gällande byggnad/bana, förnödenheter samt nödställda och markering för dessa, se dokument "Ban- och tävlingsspecifikation"[1]. Roboten skall utgående från en vald startposition undersöka en okänd byggnad, för att finna den snabbaste vägen till i förväg utplacerade "nödställda". Vidare skall roboten återvända till startpositionen, plocka upp förnödenheter med en fastmonterad gripklo samt leverera dessa till de nödställda på snabbast möjliga tid. För specifikationer gällande uppdraget, se "Ban- och tävlingsspecifikation"[1]. Körningen skall kunna utföras både autonomt och manuellt, där styrningen i sistnämnda fallet sköts trådlöst via en dator. Relevant mät- och styrdatalia skall under körningen skickas till dator via trådlös länk. [2].



Figur 1: Systemet i dess omgivning. Roboten står markerad i startposition och den svarta rutan markerar nödställda.

1.1 Parter

Ingående parter för denna kravspecifikation är beställare (kund) Kent Palmkvist, ISY, handledare Peter Johansson, ISY, samt producenter tillika projektets utförare i form av projektgruppen listad i tabell under Projektdeltagare.

1.2 Syfte och mål

Projektets syfte är att ge erfarenhet av arbeten med större projekt inom elektronikkonstruktion samt en förståelse för och kunskaper om metoder för strukturerad konstruktion av stora, komplexa digitala system[3]. Målet med projektet är att konstruera en autonom undsättningsrobot i linje med de krav som listas i detta dokument samt leverera prototypen till beställare, inom ramar för budget.

1.3 Bakgrundsinformation

Beställaren har en önskan om att producera en robot som effektivt kan förse nödställda i en byggnad olika förnödenheter och anordnar därfor en tävling där olika konstruktioner ställs mot varandra. Detta för att kunna utvärdera olika konstruktionsalternativ, samt ge deltagarna i kursen TSEA56 erfarenheter inom projektarbete och elektronikkonstruktion.

1.4 Definitioner

Då projektets uppgift består av att konstruera en prototyp av en undsättningsrobot för en verlig situation används flera analogier till motsvarande verklighet. Byggnad anges återkommande som benämning på den bana där robotens uppdrag kommer att utvärderas. I denna byggnad kommer markeringar finnas vilka motsvarar nödställda i en verlig situation. Roboten ska alltså kunna återvända till markeringen för nödställda med något föremål som symboliseras någon förnödenhet.

2 ÖVERSIKT AV SYSTEMET

I detta avsnitt sker en översiktlig redogörelse över systemet.

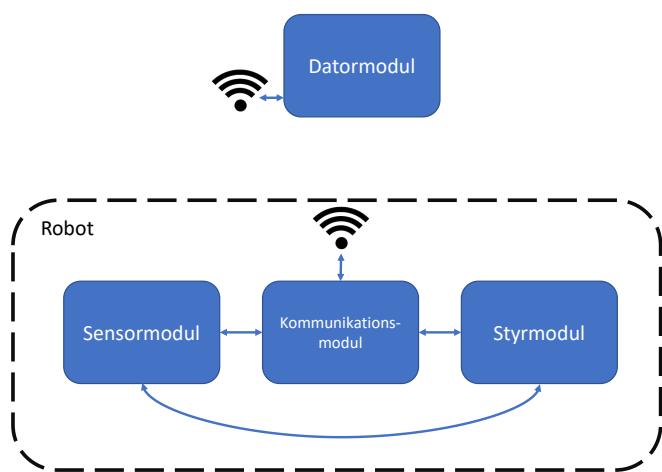
2.1 Grov beskrivning av produkten

Produkten består av en robot som ska kunna köra i ett plant område. Den ska både kunna styras manuellt med styrkommandon och kunna utföra sitt uppdrag autonomt. Mjukvara till en dator ingår för att kunna övervaka och styra roboten, vilket ska ske trådlöst.

2.2 Ingående delsystem

Figur 2 är en principskiss över systemets fyra delsystem. Den slutliga interaktionen mellan delsystemen kan komma att ändras, men utan att påverka slutproduktens funktionalitet.

Roboten kommer att bestå av tre moduler/delsystem: en *sensormodul* som samlar information från robotens omgivning, en *styrmodul* som reglerar robotens rörelser och en *kommunikationsmodul* som är ansvarig för den trådlösa kommunikationen mellan robot och dator. Utöver dessa definieras även ett fjärde delsystem som benämns *datormodul*, vilken utgör ett användargränssnitt till roboten för styrning, övervakning och lagring av data.



Figur 2: En översikt av systemet.

Se avsnitt 3 till 6 för en kort individuell beskrivning och kravsförteckning av respektive av dessa fyra delsystem.

2.3 Avgränsningar

Roboten är begränsad till att bara kunna hantera körbanan enligt ban- och tävlingsspecifikationen, se dokument ”Ban- och tävlingsspecifikation”[1] samt ha en körförmåga som stämmer överens med tävlingsreglerna. Roboten ska även klara av uppgiften specificerad i tävlingsspecifikationerna tre av fyra gånger vid tävling och demonstration. Gripklon, vilken är fastmonterad på roboten, är begränsad till att enbart kunna greppa lätta föremål.

2.4 Generella krav på systemet

Krav	Version	Beskrivning	Prioritet
1	Original	Systemet ska bestå av en robot och ett datorsystem i en datormodul.	1
2	Original	Roboten ska vara moduluppbryggd.	1
3	Original	Modulerna ska vara utbytbara/oberoende.	1
4	Original	Varje modul ska innehålla minst en egen processor.	1
5	Original	I systemet ska det finnas en kommunikationsmodul.	1
6	Original	I systemet ska det finnas en styrmodul.	1
7	Original	I systemet ska det finnas en sensormodul.	1
8	Original	Under körning ska roboten fortlöpande ta styrbeslut.	1
9	Original	Datormodulen ska kunna spara sensordata.	1
10	Original	Datormodulen ska kunna styra gripklon i manuellt läge.	2
11	Original	Det ska finnas en brytare på roboten med vilken man väljer fjärrstyrningsläge eller autonomt läge.	1
12	Original	Det ska finnas en knapp på roboten med vilken man startar roboten i tävlingen.	2
13	Original	Kartläggningen ska göras på en begränsad tid, se "Ban- och tävlingsspecifikation" ^[1] .	2
14	Original	Roboten ska ha en LCD-display som visar värden från valda sensorer.	3
15	Original	Roboten ska ha en gripklo.	1
16	Original	Roboten ska ta den optimala vägen från startpunkt till den nödställda.	2
17	Original	Roboten ska endast kartlägga byggnaden tills den kortaste vägen mellan startpunkt och slutförslag har identifierats.	3
18	Original	Roboten ska klara av tävlingsuppgift specificerad i "Ban- och tävlingsspecifikation". ^[1]	1
19	Original	Roboten ska klara av tävlingsuppgift specificerad i "Ban- och tävlingsspecifikation" [1] 3 av 4 gånger.	2
20	Original	Det ska finnas en styrstrategi så att roboten kan köra rakt i en korridor och maximalt avvika 4 cm från mittlinjen.	2
21	Original	Roboten ska kunna fjärrstyras via trådlös länk från datormodulen.	1
22	Original	Roboten ska kunna hitta och greppa föremålet med gripklon autonomt.	3
23	Original	Roboten ska kunna utföra i manuellt läge: fram, fram vänster, fram höger, back, stopp, rotera vänster och rotera höger.	1

3 KOMMUNIKATIONSMODUL

Kommunikationsmodulen ska sköta kommunikationen mellan robotens övriga moduler och PC via trådlös länk.

3.1 Krav

Krav	Version	Beskrivning	Prioritet
24	Original	Roboten ska fortlöpande under körning skicka data till en bärbar PC.	1
25	Original	Parametrar till robotens styralgoritm ska kunna initieras via den trådlösa länken.	2

4 SENZORMODUL

Senzormodulen är den del av roboten som hanterar sensorer.

4.1 Krav

Krav	Version	Beskrivning	Prioritet
26	Original	Under körning ska roboten fortlöpande samla in mätdata från sensorerna.	1
27	Original	Senzormodulen ska kunna mäta avstånd på minst 24 cm.	1
28	Original	Senzormodulen ska kunna avläsa svarta golvmärkningar.	1
29	Original	Senzormodulen ska ha switch/brytare och startknapp anslutet till sig.	1

5 STYRMODUL

Styrmodulen styr alla robotens reglage och sköter därfor styrningen. Styrmodulen styr även en gripklo som sitter längst fram på roboten och som kan plocka upp ett förbestämt föremål.

5.1 Krav

Krav	Version	Beskrivning	Prioritet
30	Original	Under körning ska roboten fortlöpande samla styrdata.	1
31	Original	Styrmodulen ska kunna greppa samt släppa ett föremål med en gripklo.	1
32	Original	Styrmodulen ska kunna styra hjulen.	1

6 DATORMODUL

Datormodulen kommer att vara den delen av systemet som sparar data och är användargränssnittet som används för styrning av roboten. Daten kommer kontinuerligt att skickas till datormodulen via en trådlös länk.

6.1 Krav

Krav	Version	Beskrivning	Prioritet
33	Original	Datormodulen ska under körning rita upp den utforskade delen av byggnaden som en karta där robotens startposition, nuvarande position, nödställd och den kortaste vägen mellan startpunkt och nödställd markeras.	2
34	Original	Datormodulen ska kunna presenteras på den bärbara PC:ns skärm på ett användarvänligt sätt.	2
35	Original	Datormodulen ska kunna spara mätdata på fil.	2

7 EKONOMI

Ekonомiska resurser för projektet.

Krav	Version	Beskrivning	Prioritet
36	Original	Efter godkänd projektplan (BP2) ska projektet ta 230 arbetsstimmar per person att slutföra.	2

8 LEVERANSKRAV OCH DELLEVERANSER

Under projektets gång kommer ett flertal dokument tas fram och uppföljningar av krav kommer att ske enligt givna leveransdatum [4].

9 DOKUMENTATION

Tabell 8 listar de dokument som kommer att produceras.

Tabell 8: Dokument som skall produceras.

Dokument	Språk	Syfte	Format
Kravspecifikation	Svenska	Visa vilka krav som roboten i slutet ska testas mot.	PDF
Ban- och tävlingsspecifikation	Svenska	Specifisera gemensamma regler för tävling samt specificera banans utformning.	PDF
Projektplan	Svenska	Ge en plan för när milstolpar ska nås.	PDF
Designspecifikation	Svenska	Robotens design att utgå från under uppbyggnad.	PDF
Reglerförstudie	Svenska	Studera möjliga reglerlösningar.	PDF
Sensorförstudie	Svenska	Studera möjliga sensorlösningar.	PDF
Kommunikation och Konstruktion förstudie	Svenska	Studera möjliga kommunikationslösningar.	PDF
Tidrapport	Svenska	Ge en översikt över varje gruppmedlems nedlagda tid.	xlsx
Teknisk dokumentation	Svenska	Förklara hur roboten är byggd.	PDF
Användarhandledning	Svenska	Förklara hur en användare ska använda systemet.	PDF
Efterstudie	Svenska	Ge en lätt översikt över hur projektet har gått.	PDF
Kappa	Svenska	Ge en komplett överblick av det genomförda arbetet.	PDF

REFERENSER

- [1] . Grupp 1, 2 och 3, “Ban- och tävlingsspecifikation,” 2021.
- [2] M. Krysander, “TSEA56: Projektdirektiv för en undsättningsrobot,” http://www.isy.liu.se/edu/kurs/TSEA56/projektdirektiv/Projektdirektiv_for_undsattningsrobot_21.pdf, [Online; hämtad 25-01-2021].
- [3] ——, “Kursinformation TSEA56,” <http://www.isy.liu.se/edu/kurs/TSEA56/kursinformation/>, [Online; hämtad 25-01-2021].
- [4] ——, “Leveranser i projektet 2021,” <http://www.isy.liu.se/edu/kurs/TSEA56/leveranser.html>, [Online; hämtad 25-01-2021].

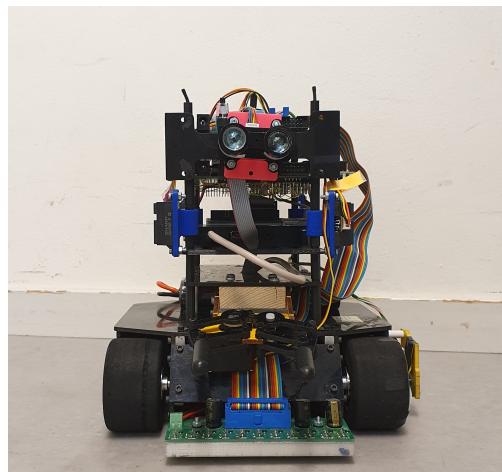
H ANVÄNDARMANUAL

Användarhandledning

Alice Nicklasson Cedbro, Maja Boström, Johanna Carlson, Lovisa Byman, Filip Johansson,
Edvin Leuchovius, Gunnar Arctaedius

7 juni 2021

Version 1.0



Status

Granskad	Alla	2021-05-19
Godkänd	Kent Palmkvist	2021-06-02

Projektidentitet

Grupp E-post: TSEA56_2021VT_O2_G1@groups.liu.se

Beställare: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Kund: Kent Palmkvist, Linköpings universitet
Tfn: +46 13 28 1347
E-post: kent.palmkvist@liu.se

Handledare: Peter Johansson
Tfn: +46 13 28 1345
E-post: peter.a.johansson@liu.se

Kursansvarig: Mattias Krysander
Tfn: +46 13 28 2198
E-post: mattias.krysander@liu.se

Projektdeltagare

Namn	Ansvar	E-post
Alice Nicklasson Cedbro	Projektledare	alice327@student.liu.se
Maja Boström		majbo451@student.liu.se
Johanna Carlson		johca194@student.liu.se
Lovisa Byman		lovby920@student.liu.se
Filip Johansson	Dokumentansvarig	filjo844@student.liu.se
Edvin Leuchovius		edvle009@student.liu.se
Gunnar Arctaedius		gunar889@student.liu.se

INNEHÅLL

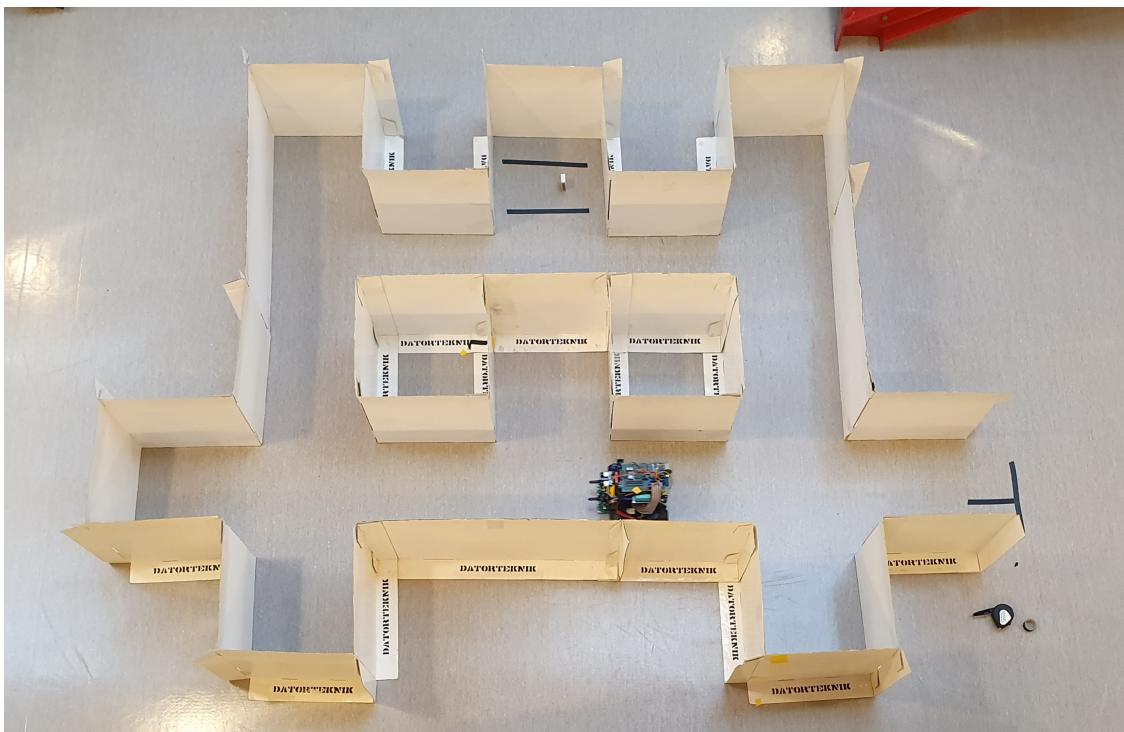
1	Inledning	1
2	Komponenter	1
3	Mjukvara	2
3.1	Kom igång med mjukvaran	2
3.2	Användargränssnitt	3
4	Användning	6
4.1	Före start	7
4.2	Startposition i manuellt läge	7
4.3	Startposition i autonomt läge	7
4.4	Starta roboten	7
A	Ban- och tävlingsspecifikation	9

DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2021-05-19	Första version	Alla	Alla

1 INLEDNING

Detta dokument beskriver handhavande för den undsättningsrobot som utvecklats i kursen Kandidatprojekt i Elektronik given vid Linköpings universitet våren 2021. Uppgiften har varit att utveckla en autonom undsättningsrobot som kan lokalisera nödställda genom att utforska en okänd labyrint. När detta är gjort ska robotten åka kortast möjliga väg tillbaka till startpositionen, hämta ett föremål med dess gripklo, och slutligen leverera detta föremål till de nödställda inuti labyrinten. Figur 1 visar ett exempel på undsättningssområdet i en labyrint. Detta dokument beskriver hur robotten är ämnad att användas, lämpliga miljöer för den att köra i samt hur mjukvara för övervakning respektive körning är tänkt att användas.



Figur 1: En pågående körning med robotten.

2 KOMPONENTER

Följande komponenter medföljer leveransen av projektet:

1. Undsättningsrobot,
2. 2st batterier,
3. Batteriladdare,

4. Användargränssnitt för trådlös manövrering och övervakning.

3 MJUKVARA

Med roboten medföljer även ett program för övervakning av autonom körning och styrning i manuellt läge.

3.1 Kom igång med mjukvaran

Följande instruktioner utgår från att användarens operativsystem är Microsoft Windows 10. Om mjukvaran installerats tidigare behöver ej steg 1–3 och 8–10 utföras.

1. Placera koden på datorn som ska användas.
2. Installera Python 3.8 eller högre.
3. Kontrollera att alla Pythonpaket som behövs är installerade, se Tabell 1 . Om paketet ”*PIL*” saknas installera istället paketet ”*pillow*”. Om det saknas något paket installera dessa † .
4. Kör ”*run_all.bat*” för att starta datormodulen, om detta fungerar hoppa till steg 7.
5. Öppna ett Powershell-fönster och navigera in i det in i mappen ”*websocke*”. Kör ”*server.py*” med Python. Om detta fungerar kommer det att stå i terminalen ”*Listening on port 6453x*”.
6. Öppna ett till Powershell-fönster och navigera in i startmappen för koden. Starta ”*GUI.py*” med Python. Om detta fungerar ska det grafiska gränssnittet startas. Om gränssnittet inte öppnas kan det bero på att ett Pythonpaket saknas.
7. Gå in i nätverkspanelen och anslut till det trådlösa nätverket ”Rover”. Ange lösenordet ”*rover1234*”. Om gränssnittet uppdateras är det nu klart och resten av stegen kan hoppas över.
8. Kolla upp den nuvarande IP-adressen till datorn, den bör vara på formen ”192.168.4.x”. Kom ihåg denna.
9. Öppna ett nytt Powershell-fönster. Kontrollera att ”*ssh*” är installerat i terminalen. Skriv ”*ssh pi@192.168.4.1*” i terminalen och tryck på ”*enter*”. Följ instruktionerna som ges för att godkänna anslutningen om det är första gången en inloggning sker från en ny enhet och ange slutligen lösenordet ”*rover1234*”. Terminalen är nu kopplad till roboten.
10. Navigera in i mappen ”*bin*”. Skriv ”*nano main.py*”. På rad 13 står det ”*PC_IP=192.168.4.x*”. Ändra IP-adressen på denna rad till din IP-adress. Kontrollera även att raden under stämmer överens med det som står i terminalen från steg 4. Tryck ”*ctrl-s*” för att spara och sedan ”*ctrl-x*” för att gå ur. Starta om roboten. Koppla upp dig mot dess trådlösa nätverk.

† Förslagvis med ”*pip*”.

Tabell 1: Pythonmoduler som används i scriptet "main.py".

Modul	Beskrivning
socket	Åtkomst till operativsystemets nätverksgränssnitt/BSD-socket. Används för att skicka och ta emot IP/TCP-paket över en Wi-Fi anslutning till en annan adress och portnummer.
spidev*	Åtkomst till SPI-enheter via spidev drivrutiner för Linux.
time	Används för att skapa fördräjning av programkod under exekvering med funktionen "sleep".
sys	Åtkomst till variabler som är relaterade till programexekveringen. Används för att hämta och sedan skriva ut felmeddelanden när undantag behöver hanteras.
math	Åtkomst till matematiska funktioner enligt C-standard.
Tkinter	Används för att skapa grafiska element.
csv	Används för att spara information från robot i ett lättplottat format.
datetime	Används under sparning av information från robot för att ge en tid då informationen mottagits.
os	Används för att frigöra filsökvägar från ett specifikt operativsystem.
PIL	Används för att visa bilder.

*Modulen ingår inte i Pythons standardbibliotek.

3.2 Användargränssnitt

Användargränssnittet gör det möjligt att dels styra undsättningsroboten i manuellt läge, och dels att övervaka undsättningsroboten i realtid under sitt uppdrag att automatiskt lokalisera och sedan undsätta en nödställd som befinner sig någonstans i byggnaden.

**Figur 2:** En översikt av användargränssnittet.

Nedan följer en beskrivning av siffrorna i Figur 2.

1. Under autonom körning kommer robotens utforskade karta att ritas ut i denna ruta.
2. Här kan robotens reglerparametrar visas. Om användaren trycker på hämta kommer de nuvarande reglerparametrarna att visas i rutorna. Om användaren trycker på skicka kommer de värden som står i rutan att skickas till roboten för att användas som nya reglerparametrar. Observera att innan de kan skickas MÅSTE alla fält ha siffror i sig.
3. Denna ruta kördata kommer under körningens gång att uppdateras med information som hämtas från roboten. Läget roboten är i betecknas med siffror. Om den är i läge 0 är roboten redo att motta styrkommandon från användaren. Om läget är 1 är roboten i autonomt läge och väntar på att den fysiska startknappen ska tryckas. I läge 2 kartlägger roboten.
4. Denna ruta är styrindikatorn, vars symboler lyser upp i rött om en styrknapp trycks ner.
5. Ruta med förklaring till kartans olika färger.

3.2.1 Visa och ändra reglerparametrar

Undsättningsroboten har totalt 6 stycken reglerparametrar. Dessa kontrollerar robotens rörelser i autonomt läge och deras respektive effekt anges i Tabell 2. Roboten levereras med en komplett uppsättning av optimala reglerparametrar för sitt uppdrag, så användare avråds från att ändra dessa, speciellt under en pågående räddningsinsats.

Tabell 2: Reglerparametrar och deras funktion

Parameter	Funktion	Tillåtna värden	Initieringsvärde
K_p korridor	Reglerar linjärt mot avståndet från mitten av en modul vid körning i korridor.	0 till 255	50
K_a korridor	Reglerar linjärt mot vinkelfelet vid körning i korridor.	0 till 255	45
K_p rotation	Reglerar linjärt mot vinkelfelet vid rotation.	0 till 255	16
K_a rotation	Reglerar utifrån förändringshastigheten av vinkelfelet.	0 till 65535	0
K_p tejp	Reglerar linjärt mot avståndet från mitten av en tejpremsa vid tejpföljning.	0 till 255	150
K_a tejp	Reglerar utifrån förändringshastigheten av avståndet från mitten av en tejpremsa vid tejpföljning.	0 till 65535	0

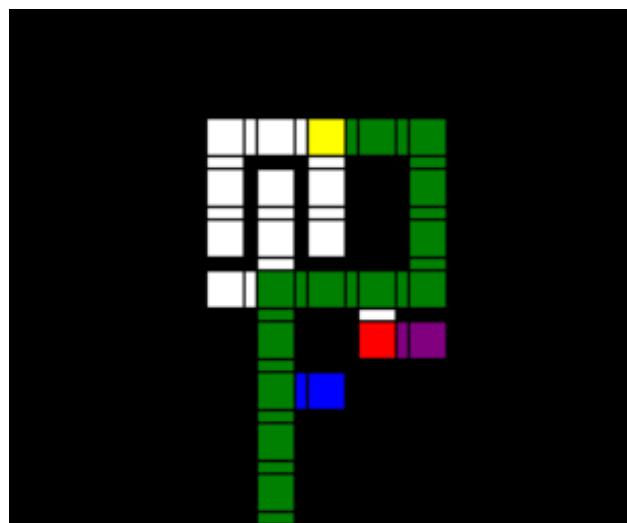
För att se robotens reglerparametrar kan användaren trycka på knappen "Hämta". Då läses de senaste reglerparametrarna in från fil och visas på skärmen i rutan för reglerparametrar. För att sedan ändra reglerparametrar kan användaren skriva in det önskade värdet i en av rutorna och trycka på knappen "Skicka" vilket då vidarebefordrar värdet till roboten.

Tabell 3: En förklaring av färgkodningen i kartan.

Färg	Förklaring
Svart	Ej utforskat eller upptäckt vägg.
Blå	Roboten har sett att det går att köra i denna ruta men har ej undersökt om de nödställda finns på positionen.
Vit	Besökt av roboten.
Röd	Markerar robotens nuvarande position.
Gul	De nödställdas position.
Lila	Robotens planerade färdväg.
Grön	Markerar den vägen roboten beräknar som den snabbaste mellan nödställda och ingång.

3.2.2 Beskrivning av kartan

Kartan representeras av ett rutnät med kvadrater och rektanglar enligt Figur 3, där olika färger markerar olika tillstånd, se Tabell 3 för förklaring till de olika färgerna. Kartan uppdateras två gånger i sekunden under körning med informationen roboten tar in.


Figur 3: En utforskad karta med hittad nödställda.

3.2.3 Manuell styrning

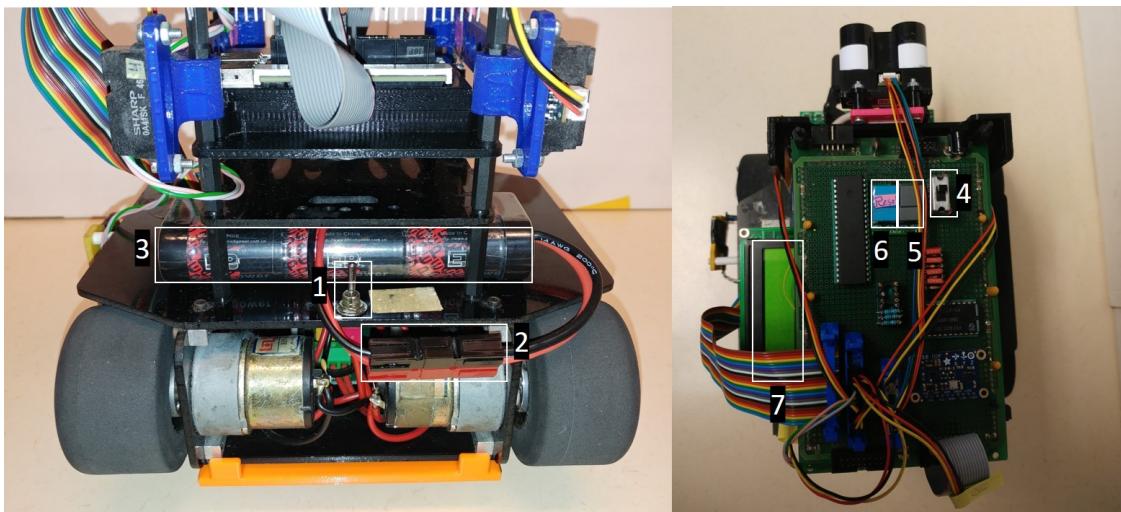
Det finns möjlighet för användaren att styra roboten manuellt från tangentbordet på datorn med användargränssnittet. De möjliga kommandon som då används kan ses i Tabell 4. När tangenter trycks kommer kommandon skickas till roboten, men om brytaren på roboten ej är i manuellt läge kommer roboten att ignorera dessa kommandon. Då roboten är i manuellt läge utförs överföringarna mellan robot och användargränssnittet oftare för att roboten ska bli mer responsiv.

Tabell 4: De olika kommandona och tillhörande tangent.

Förklaring	Tangent
Rakt fram	W
Rakt bakåt	S
Rotera höger	D
Rotera vänster	A
Höger sväng	E
Vänster sväng	Q
Öppna gripklon	G
Stäng gripklon	R

4 ANVÄNDNING

Detta avsnitt beskriver hur undsättningsroboten används och bör läsas noggrant innan start påbörjas för att undvika felaktig användning. Figur 4, tillsammans med nedanstående lista, visualiseras några viktiga komponenters placering på roboten.

**Figur 4:** Robotens komponenter.

1. Huvudströmbrytare – Spänningssätter robotens samtliga elektronikkomponenter när den är tillslagen.
2. Anslutningskontakt för batteriet.
3. Batteri
4. Brytare för att välja manuellt respektive autonomt läge. – En etikett på brytarens ena sida anger vilket som är manuellt respektive autonomt läge. M och A anger manuellt respektive autonomt läge.

5. Startknapp – Startar roboten givet att autonomt läge är valt med brytaren (nr. 4). Roboten bör vara korrekt placerad enligt vad som anges för start i autonomt läge.
6. Resetknapp – Återställer sensorer till startvärdet.
7. LCD-display– Skriver ut värden under körning.

4.1 Före start

Innan användning av roboten är det viktigt att ett fulladdat batteri är monterat på roboten. Detta kan åstadkommas genom att ladda ett batteri med den medföljande laddaren i ungefär en timme.

Kontrollera att strömbrytaren (nr. 1) är frånslagen (brytaren ska peka i samma riktning som robotens framåtriktning). I händelse av att det redan sitter ett batteri monterat och anslutet behöver dess anslutningskontakt (nr. 2) kopplas ifrån och sedan plockas bort. Anslut och montera sedan det laddade batteriet så att det ser ut som i Figur 4.

Placera sedan roboten på önskad startposition enligt antingen Avsnitt 4.2 eller 4.3.

4.2 Startposition i manuellt läge

Om roboten ska framföras i manuellt läge kan den placeras godtyckligt, idealt på ett plant underlag utan kanter där den inte kan välta, till exempel vid trappor, men det är framförallt upp till den som ska manövrera roboten att bedöma.

4.3 Startposition i autonomt läge

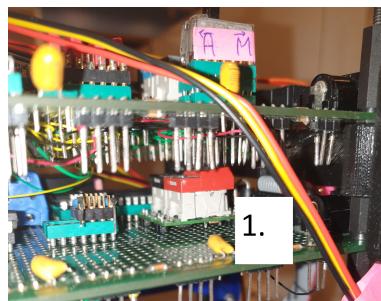
I autonomt läge bör roboten placeras vid ingången till en tillåten labyrinth enligt ban- och tävlingsspecifikationen, se Appendix A, på ett sådant sätt att den är placerad med reflexsensorn mitt över toppen på den T-formade svarta markeringen med fronten riktad inåt i labyrinthen. Utöver vad som anges i ban- och tävlingsspecifikationen så är det viktigt att det är tydlig kontrast mellan färgerna på golv och de svarta markeringarna för förnödenhetens position och för nödställd. Idealt bör golvet ha en vit, matt färg.

4.4 Starta roboten

För att köra roboten behöver den startas den enligt den ordning som anges nedan.

1. Sätt på strömmen på roboten genom att sätta på strömbrytaren (nummer 1) i Figur 4.
2. Välj manuellt eller autonomt läge med robotens brytare enligt nummer 4 i Figur 4. Om manuellt läge är valt är det nu möjligt att manövrera den från användargränssnittet av en separat enhet som är ansluten till roboten.
3. Tryck på reset-knapparna, nummer 6 i Figur 4 och på nr 1 i Figur 5.
5. Om användargränssnittet önskas vänta på koppling till robot och gå igenom de steg som tas upp under Avsnitt 3.1

- 6.** Tryck på robotens startknapp, enligt nummer 5 i Figur 4, för att starta den autonoma körningen. När roboten känner av tejp med tejpsensorn kommer den att börja utforska labyrinten.



Figur 5: Nummer 1. visar resetknappen på styrmodulen

A BAN- OCH TÄVLINGSSPECIFIKATION

Ban- och Tävlingsspecifikation

Filip Johansson, Oscar Pihl, Max Ehn, David Ekblad, Marcus Jämte,
Anton Thunberg, Eric Sevonius, Gunnar Arctaedius, Lovisa Byman

4 februari 2021

Version 1.0



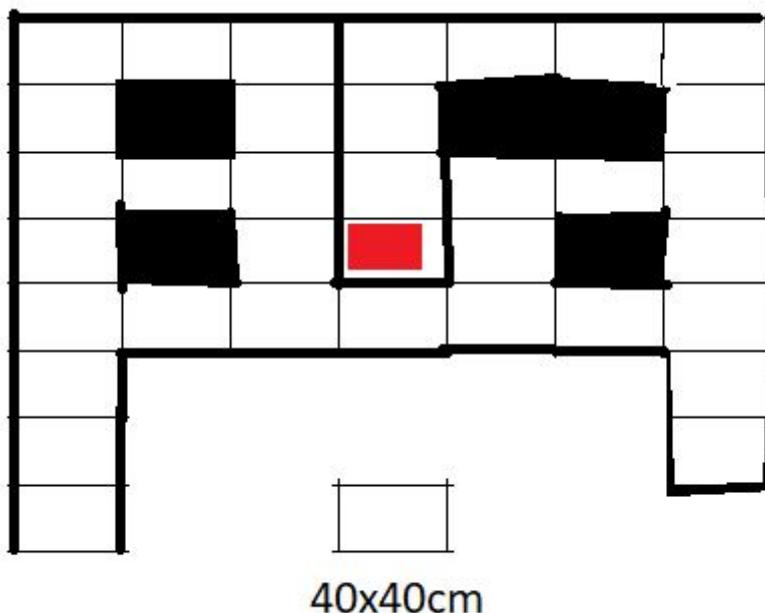
Dokumenthistorik

Version	Datum	Utförda ändringar	Utförda av	Granskad
0.1	2021-01-29	Första utkastet	David Ekblad, Marcus Jämte, Anton Thunberg, Eric Sevönius, Gunnar Arctaedius, Lovisa Byman	Gunnar Arctaedius
0.2	2021-02-04	Har lagt till rubrik, dokumenthistorik, titel och författare. Krav har reviderats i enlighet med beställarens uppmaningar.	Max Ehn, Filip Johansson, Oscar Pihl	Filip Johansson
1.0	2021-02-05	Förtydligande gällande den rektangulära markeringens orientering i banan.	Filip Johansson	Filip Johansson

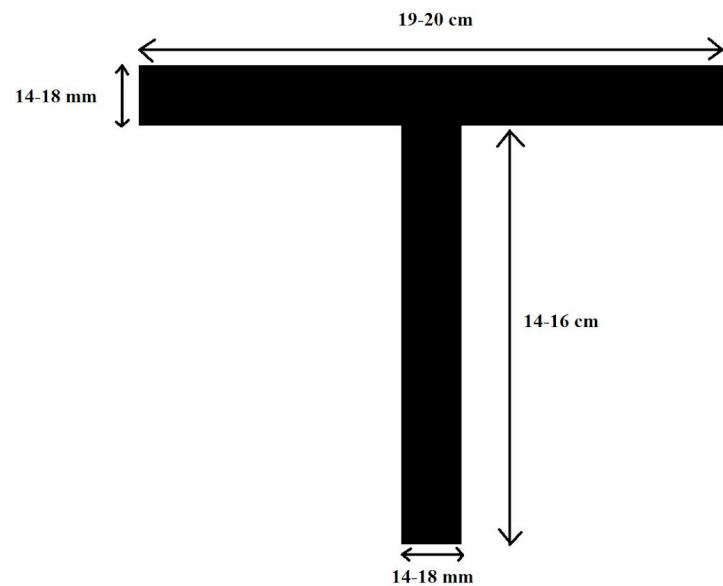
Banregler

Nedan definieras de banregler som kommer att gälla under tävlingen för samtliga involverade projektgrupper.

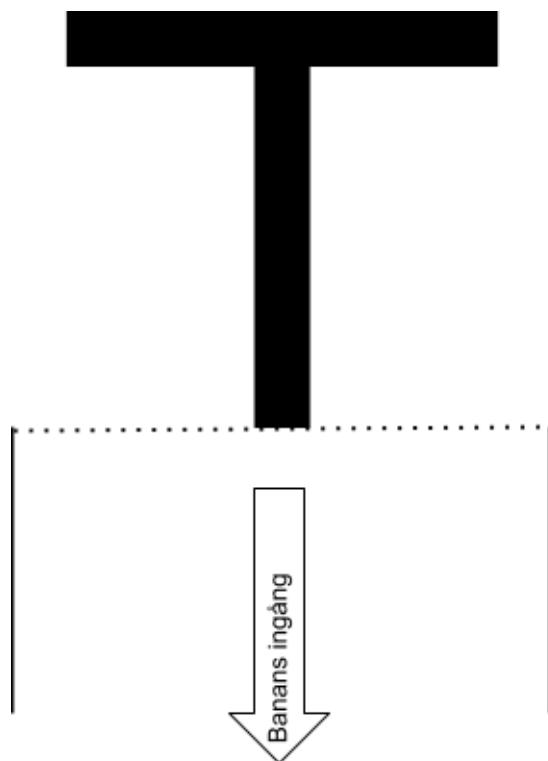
- Det är tillåtet med cykler i banan.
- Det är tillåtet med återvändsgränder.
- Det är tillåtet med fyrvägskorsningar.
- Banan skall bestå av moduler (rutor) om $40 \pm 4 \times 40 \pm 4$ cm och spänna upp ett rutnät av max 25x25 rutor. Se figur 1.
- Alla väggar (förutom dem som definierar ingången till banan) skall sluta i en annan vägg.
- Väggar ska ha vit färg.
- Golvet ska ha vit färg.
- Den nödställda skall markeras med en icke ifylld rektangel av svart tejp (elitejp) 30x20 cm, centrerad i en av banans moduler och med rektangelns kortsidor (20 cm) parallella med korridorens väggar.
- Förmödenhet skall vara markerad med ett "T" av elitejp (14-18mm bredd) med dimensioner (19-20)x(14-16) cm. Se figur 2.
- Förmödenhet ska vara placerad utanför ingången av banan. Orienteringen samt placeringen av "T-markeringen" är sådan att basen av "T-markeringen" är parallell med ingången. Se figur 3.
- Det ska finnas flera vägar till de nödställda.
- Banan ska delas upp i ett flertal korridorer och korsningar, där korridorens bredd är 1 modul.



Figur 1: Exempel på en bana



Figur 2: T-markeringen



Figur 3: T-markeringens orientering relativt banans ingång.

Tävlingsregler

Nedan definieras de krav och regler som påverkar hur de olika robotarna bedöms under tävlingsmomentet. Upplägget av tävlingen i sig beskrivs även.

Allmänna tävlingsregler

Varje grupp ska köra igenom båda faserna 3 gånger. Bästa tid för var fas, enligt tävlingsreglerna för varje sådan nedan, blir gruppens resultat tid i det tävlingsmomentet. Om ett lag inte klarar 2 utav 3 körningar så diskas gruppen. Att inte klara en körning definieras av:

- Inte klara tävlingsmoment ett på 10 minuter.
- Välta en vägg i tävlingsmoment två
- Släppa ner förnödenheten utanför rektangeln som den ska placeras i.
- Hela förnödenheten är utanför markeringen för den nödställda när den släpps ner.

Följande ger tidstillägg:

- Nudda en vägg, 10s.
- Någon del av förnödenheten är utanför markeringen för den nödställda när den släpps ner eller efter att hela körningen är klar, 15s.
- Hela förnödenheten är utanför markeringen för den nödställda efter att hela körningen är klar, 30s

Tävling 1

Första fasen kommer mätas som en separat tävling för ära. Här börjar tiden räknas när första delen av roboten passerar igenom ingången och slutar när den undsatte har blivit funnen och sista delen av roboten passerat ut genom ingången.

Tävling 2 (den stora tävlingen)

Den andra tävlingen, den som verkligen spelar roll, utspelear sig i fas 2. Tideräkningen börjar efter att förnödenheten har blivit upplockad och första delen av roboten åker in i banan. Sedan skall roboten släppa förnödenheten i rutan som markerar de nödställda. Sedan skall roboten återvända, tiden slutar när sista delen av roboten lämnar banan.

I EFTERSTUDIE

EFTERSTUDIE

Alice Nicklasson Cedbro, Maja Boström, Johanna Carlson, Lovisa Byman, Filip Johansson, Edvin Leuchovius, Gunnar Arctaedius

Version 1.0

Status

Granskad	Alice Nicklasson Cedbro, Johanna Carlson, Maja Boström	2021-06-07
Godkänd		2021-06-xx

PROJEKTIDENTITET

Grupp 1, 2021/VT
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Gunnar Arctaedius			gunar889@student.liu.se
Filip Johansson	Dokumentansvarig (DOK)		filjo844@student.liu.se
Edvin Leuchovius			edvle009@student.liu.se
Lovisa Byman			lovby920@student.liu.se
Johanna Carlson			johca194@student.liu.se
Maja Boström		070-7998640	majbo451@student.liu.se
Alice Nicklasson Cedbro	Projektledare (PL)		alice327@student.liu.se

E-postlista för hela gruppen: TSEA56_2021VT_O2_G1@groups.liu.se

Kund: Kent Palmkvist, LINKÖPING,
kundtelefon +46 13 28 1347, kent.palmkvist@liu.se

Kursansvarig: Mattias Krysander, +46 13 28 2198, mattias.krysander@liu.se
Handledare: Peter Johansson, +46 13 28 1345, peter.a.johansson@liu.se

Innehåll

1	Tidsåtgång	1
1.1	Arbetsfördelning	1
1.2	Tidsåtgång jämfört med planerad tid	1
2	Analys av arbete och problem.....	2
2.1	Reflektioner av de nya laborationerna	2
2.1.1	AVR-laboration.....	2
2.1.2	Mätteknik.....	2
2.1.3	AVR-laboration med I2C	2
2.2	Förstudier	2
2.2.1	Sensorförstudie.....	2
2.2.2	Kommunikationsförstudie	3
2.2.3	Reglerförstudie.....	3
2.3	Vad hände under de olika faserna (bra/dåligt/orsak)?	3
2.4	Hur vi arbetade tillsammans (ansvar, beslut, kommunikation etc.)?	4
2.5	Hur använde vi projektmodellen?.....	4
2.6	Hur fungerade relationen med beställaren?.....	4
2.7	Hur fungerade relationen med handledaren?	4
2.8	Distansläge.....	4
2.9	Tekniska framgångar/problem	5
2.9.1	Bluetooth.....	5
2.9.2	Minne	5
2.9.3	Gyro	6
3	Måluppfyllelse	6
3.1	Vad har uppnåtts?	6
3.2	Hur fungerade leveransen?.....	6
3.3	Hur har studiesituationen påverkat projektet?	6
4	Sammanfattning.....	7
4.1	De tre viktigaste erfarenheterna.....	7
4.2	Godta råd till de som ska utföra ett liknande projekt.....	7

1 TIDSÅTGÅNG

I projektets början skapades en tidplan för att ge gruppen en bra utgångspunkt inför projektet. Utformandet av en tidplan tvingade gruppen att tänka igenom vad som skulle göras och när. I tidplanen identifierades ett antal aktiviteter och den tänkta tidsåtgången för vardera aktiviteten. Projektet kan delas upp i tre faser: före, under och efter. Tidsåtgången för under-fasen, efter-fasen planerades till totalt 1 610 timmar vilket motsvarar 230 timmar per projektmedlem.

1.1 Arbetsfördelning

I tidplanen fördelade vi arbetet jämt mellan alla projektmedlemmar. Det var svårt att helt upprätthålla detta mot projektets slut då vi från början hade delat upp medlemmarna på de olika modulerna. Vi delade upp medlemmarna på de olika modulerna eftersom det arbetssättet kändes tidseffektivt. Att dela upp medlemmarna ledde till att vardera personen fick spetskunskap på just sin modul men saknade kunskap om de andra modulerna. Styrmodulen var den modul som generellt krävde mest tid. När de övriga modulerna var färdigställda var det dock svårt för personerna som jobbat med dessa att hjälpa till med styrmodulen. Det hade krävts mycket påläsning om styrmodulen och med avseende på vart vi då befann oss i projektet kändes det ineffektivt. Det beslutades istället att majoriteten av dokumentationen skulle skriva av de personer vars moduler var färdiga.

Projektet delades som tidigare sagt upp i tre faser. I före-fasen skrev vi dels dokument ihop i grupp och dels fördelades arbetet genom att var och en arbetade med varsin del i dokumenten. Om arbetet med ett dokument delades upp gick vi slutligen igenom arbetet tillsammans för att få det enhetligt. I under-fasen pågick arbetet parallellt för respektive modul och när vi skulle integrera modulerna med varandra så hade vi bra kommunikation. I efter-fasen så kollade vi på hur många timmar respektive person hade kvar för att arbetsfördelningen skulle vara jämn. De som hade flest timmar kvar fick ta en större del med skrivandet av dokumenten.

1.2 Tidsåtgång jämfört med planerad tid

I tabellen nedan kan man se hur vi använde timmarna i projektet gentemot hur vi hade planerat att jobba.

Fas	Planerad tid i timmar	Använt tid i timmar
Före	231	229
Under	1502	1372,5
Efter	108	270,5

2 ANALYS AV ARBETE OCH PROBLEM

För att analysera vårt arbete och de problem vi har stött på har vi diskuterat dels under projektets gång och dels efter färdigställandet av projektet.

2.1 Reflektioner av de nya laborationerna

2.1.1 AVR-laboration

En bra laboration då man fick lära känna programmet ‘Waveforms’ vilket används mycket under projektets gång. Laborationen gav en bra introduktion till programmet men tyvärr inte förståelse om hur det skulle kunna användas i projektet för t.ex. felsökning. Denna förståelse hade varit bra att få. I vårt fall fick vi denna förståelse från handledaren senare i projektet.

2.1.2 Mätteknik

Mättekniklaborationen var en bra labb där man lärde sig mycket om avbrott och hur ett program kan vara avbrottsstyrt. Laborationen hade bra svårighetsgrad och innehåll.

2.1.3 AVR-laboration med I2C

En bra och lärrik laboration om busskommunikation. Denna labb gjorde endast två av projektets medlemmar vilket vi såg som en nackdel. Busskommunikation är något som används av alla moduler och därmed till nästa år borde labb vara obligatorisk för hela gruppen. Om laborationen inte kan erbjudas hela projektgruppen ser vi gärna att det betonas att de som är masters på bussen borde göra labben vilket inte nödvändigtvis är kommunikationsmodulen. Vi valde att kommunikationsmodulsgruppen skulle göra labben medan master på I2C bussen sedna blev sensormodulen.

2.2 Förstudier

2.2.1 Sensorförstudie

Det upptäcktes tidigt att en viktig del för sensormodulen var avsökning av omgivning för att identifiera väggar i labyrinten. I sensorförstudien fokuserade vi därför på vilka sensorer som var bra för undersökning av omgivning och hinder. Ytterligare delar som vi ansåg viktiga var att kunna mäta styrningsvinkel samt tillryggalagd sträcka. Avsökning av omgivning och mätning av styrningsvinkel var speciellt viktiga för att kunna reglera roboten korrekt. De tre delarna som vi ville fokusera på gav upphov till tre frågeställningar.

- Vilka metoder för beräkning av tillryggalagd stäckå är lämpliga för undsättningsroboten?
- Hur kan ljussensorer användas för avsökning av omgivning och vilken sensor är mest lämplig för undsättningsroboten?
- Vilka sensorer kan användas för beräkning av styrningsvinkel?

Förstudien blev mycket intressant och gav användbara resultat. Inför framtidens liknande projekt kan vi rekommendera våra frågeställningar.

2.2.2 Kommunikationsförstudie

Kommunikationsförstudien gav en bra insikt i vilka olika typer av bussar som finns och vad deras för- och nackdelar är. Vi studerade vilka av ett urval busstandarder som var lämpliga för vår robot. Flera av busstyperna valdes bort på grund av att dessa inte var lämpliga att använda tillsammans med dem tillgängliga processorerna. Det återstod sedan att läsa på om alla bussar och skriva ner fakta och slutsatser. Förstudien upplevdes sammanfattningsvis som relevant enligt oss, men vi skulle vilja uppmana kommande kursdeltagare att eventuellt även undersöka några olika alternativ för trådlös kommunikation, då många principer samt terminologi är likvärdig, eller åtminstone nära relaterad med dem för fysiska bussar. Det hade varit användbart för oss med lite mer inledande bakgrund om trådlös kommunikation, till exempel vad gäller för- och nackdelar, för att snabbare komma igång med implementeringen i utförandefasen av projektet.

2.2.3 Reglerförstudie

Reglerförstudien syftade på att besvara följande frågeställningar:

- Vilken information och mätdata behövs för att kunna utföra styrningen i de olika styrmoderna?
- Hur kan en referensignal genereras i de olika styrmoderna?
- Hur ska roboten regleras för att kunna följa de genererade referensignalerna?
- Hur ska reglerparametrar väljas?

Förstudien gav en bra grund för regleringen och var till stor hjälp eftersom vi visste väldigt lite om hur roboten skulle regleras innan vi började med förstudien. Vissa delar av frågeställningen och förstudien visade sig dock inte vara så relevanta för projektet. I projektet använde vi oss enbart av mitten av en korridor som referenssignal men i förstudien tittade vi på två olika sätt att generera en referensignal på ett mer avancerat sätt vilket inte hade behövts. Överlag så skulle vi rekommendera uppgiften men det hade kanske varit bättre att fokusera mer på något annat än hur en referensignal bör genereras.

2.3 Vad hände under de olika faserna (bra/dåligt/orsak)?

Under före-fasen fick gruppen i uppdrag att, utifrån ett projektdirektiv, utforma en kravspecifikation med krav att uppfylla för att få ett godkänt projekt och därefter planera genomförandet. Allt detta var väldigt bra att göra för att få en tydlig överblick om vad som skulle göra i under-fasen och vilken teori vi behövde läsa in oss på.

I under-fasen gjordes förstudier och designspecifikation som presenterades på beslutspunkt 4. Därefter påbörjades konstruktionen av roboten. Hårdvara sammanställdes och mjukvara skrevs. Efter hand som något av kraven i kravspecifikationen bedömdes vara uppfyllt gjorde vi tester för att konfirma att kravet var uppfyllt. Om kravet uppfylldes gick vi vidare till nästa krav, annars gick vi tillbaka och konstruerade om roboten (mjukvara/hårdvara etc.) och testade igen. Detta var väldigt bra för oss för att vi kunde göra små tester allteftersom, vilket gjorde att vi kunde se till att vi inte gjorde för mycket som i slutändan kanske inte skulle funka ihop. I slutet, när vi hade bockat av tillräckligt många krav, genomfördes ett större test som motsvarade tävlingen som skulle genomföras under beslutspunkt 5. Detta test var väldigt bra, eftersom vi kunde bekräfta att vi kunde leverera en robot som uppfyllde alla krav enligt kravspecifikationen.

I efter-fasen skedde en leverans till beställaren och de slutliga dokumenten skrevs och lämnades in. Alla dokumenten var bra att skriva för att få en bra sammanfattning om vad gruppen hade gjort under projektet.

2.4 Hur vi arbetade tillsammans (ansvar, beslut, kommunikation etc.)?

I början av kursen utsåg vi Alice till projektledare som skulle ha det extra ansvaret för all mailkontakt med beställaren, boka in möten och lämna in tidrapporter. Under konstruktionsfasen delades projektgruppen upp i mindre grupper, en grupp för vardera modulen. Vilka personer som jobbade med vilka moduler föll sig ganska naturligt då projektmedlemmarna hade olika preferenser. Därmed hade vi tre olika grupperingar där varje grupp satte fördjupade sig i sin egen modul. Vi tog alla beslut tillsammans och diskuterade olika åsikter kring besluten. Då projektgruppen var oenig tillämpade vi majoritetsbeslut, alltså att alternativ med majoritet valdes. Vi skrev också ett gruppkontrakt tidigt i projektet för att alla skulle ha samma förutsättningar för en bra kommunikation och kunna undvika missförstånd med varandra.

2.5 Hur använde vi projektmodellen?

Vi använde LIPS-boken och LiU:s hemsida "Projektmodellen Lips". Där står det mycket om hur man ska förhålla sig till projektmodellen. Vi har använt oss av regler, instruktioner och dess mallar för att genomföra vårt projekt på ett bra och strukturerat sätt. I före-fasen tog vi reda på vad vårt uppdrag var för något och gjorde upp en bra bas för att ha en bra grund till när vi skulle genomföra under-fasen. I under-fasen konstruerade vi roboten och utförde kontinuerliga tester. Om ett test var bra påbörjades nästa steg. Om testet var dåligt började vi om på en ny design, ny konstruktion och testade igen. I efter-fasen levererades roboten och dokument skrevs och lämnades in.

2.6 Hur fungerade relationen med beställaren?

Relationen med beställaren har varit bra. Vi har mest haft mailkontakt för att boka in möten, vilket har fungerat bra. Vi har även fått nyttig feedback på våra dokument.

2.7 Hur fungerade relationen med handledaren?

Relationen med handledaren har varit väldigt bra och mycket användbar. Vi har fått bra svar på våra frågor och alltid känt att vi fått den hjälp som vi har behövt. Vidare har handledaren uppfattats som bra påläst och gett snabb respons, vilket uppskattades.

2.8 Distansläge

Distansläget har inte påverkat oss så mycket. I före-fasen körde vi flera möten på zoom och en del möten på plats vilket fungerade toppen. I under-fasen var alla på plats i muxen när de ville det och körde på zoom om man inte kände sig frisk eller inte ville vara på plats. Majoriteten av arbetstiden skedde på plats då projektgruppens medlemmar föredrog detta.

Det som har påverkat oss negativt är att kommunikationen blir lite svårare då det är på distans, eftersom diskussionerna inte blir lika flytande som när man ses i verkligheten. Det var väldigt givande i samband med utformningen av projektplanen att lista alla aktiviteter på en whiteboardtavla.

Något man kan ta med sig är att man kan ha beställarmöten på distans, då det fungerar minst lika bra som att ses i verkligheten.

2.9 Tekniska framgångar/problem

Här listas de tre största tekniska problemen som vi har haft.

2.9.1 Bluetooth

Felets typ - ange t.ex. hårdvarufel, mjukvarufel, systemintegrationsfel

Felets typ var ett systemintegrationsfel. Det var problematiskt att ordentligt förstå hur olika protokoll relaterade till Bluetooth fungerar, och inte minst vilket eller vilka som bör användas, då Bluetooth används för så många tillämpningar. Det var också svårt att alltid få en stabil koppling mellan två enheter, vilket eventuellt kan bero på inkorrekt användning.

Kort beskrivning av felets symptom

Det gick inget bra med Bluetooth då det var svårt att få det att fungera och det inte ville koppla till våra datorer.

Kort beskrivning av felets orsak t.ex. logiskt fel i programmering, timing-fel, glappkontakt, missförstånd av spec., felaktig design.

Det var svårt att förstå hur överföringar sker och det var inte en vanlig användning av Bluetooth. Detta gjorde att det inte fanns mycket dokumentation att tillgå om hur överföringar kan ske på ett bra sätt.

Kort beskrivning av hur ni fixade felet. T.ex. hur ni fixade mjukvarubuggen, fick ny hårdvara, virade om, gick runt problemet.

Vi bytte trådlöst kommunikationsprotokoll från Bluetooth till wifi.

Ungefärlig skattning av hur mycket tid som ni ägnade åt felsökningen (dvs. tid som kunde tjänats in om felet inte hade uppstått)

De första veckorna i under-fasen lades ner på Blåtand och sedan tog det ytterligare tid till att förstå och implementera Wi-Fi. Tiden som lades ner på detta kan nog uppskattas till ca 55 h per person alltså totalt runt 110h.

2.9.2 Minne

Felets typ

Mjukvarufel.

Felets symptom

För att kartlägga och beräkna kortaste väg, m.m. behövde vi fyra olika kartor. Från början var planen att dessa kartor skulle bestå av 8-bitars element och sparades var för sig men då räckte inte minnet på AVR:en till.

Felets orsak

Felet berodde på felaktig design.

Lösning

Vi löste problemet genom att göra om kartorna till binära kartor och sedan spara de i en och samma karta men på olika index.

Skattning av tid ägnad åt felsökning

Vi hade kunnat tjäna in ca 20h om felet inte hade uppstått.

2.9.3 Gyro

Felets typ

Mjukvarufel.

Felets symptom

Vi upptäckte att vi fick olika värden beroende på hur snabbt vi vred roboten. Vi försökte att få ut en vinkel utan att ta hänsyn till tiden vilket gav ett felaktigt resultat.

Felets orsak

Logiskt fel i programmering.

Lösning

Vi implementerade en klocka som gjorde att vi fick samma resultat oberoende på hur lång tid vridningen tog.

Skattning av tid ägnad åt felsökning

Den totala tiden som vi kunde tjänat in på tre personer är ca 30h.

3 MÅLUUPPFYLLElse

3.1 Vad har uppnåtts?

Projektet resulterade i en fungerande robot med bra tillförlitlighet. Alla krav enligt kravspecifikationen har uppnåtts, även de med prioritet tre. Vid tävlingskörning klarade roboten avsökning av labyrint på 1 minut och 22 sekunder samt leverans av fornödenhet på 24 sekunder. Hela tävlingskörningen gjordes utan några strafftillägg.

3.2 Hur fungerade leveransen?

Leveransen fungerade bra. Roboten kunde arbeta i den givna labyranten och klarade uppdraget. Roboten klarade vid leverans av kravet på tillförlitlighet men detta var ändå robotens svaga punkt.

3.3 Hur har studiesituationen påverkat projektet?

Vi har inte upplevt att vi har påverkats negativt av den nuvarande studiesituationen. Det som varit tråkigt är att tävlingar och liknande inte kan utföras på plats tillsammans med andra grupper och beställare. I övrigt känner vi att vi har fått bra vägledning genom projektet och fått den hjälp vi behöver av vår handledare.

4 SAMMANFATTNING

4.1 De tre viktigaste erfarenheterna

Den första viktigaste erfarenheten som gruppen har lärt sig är att det är otroligt viktigt och lönar sig att lägga mycket tid i början av projektet i före-fasen. Vi kollade upp mycket fakta och la mycket tid på att ordna aktiviteter som var lämpliga för just vårt projekt. Alla medlemmar känner att vi har tjänat mycket på det.

Den andra viktigaste erfarenheten som gruppen har lärt sig är vikten på en bra planering. Vi la mycket tid på hur vi skulle lägga upp våra veckor för att koordinera parallella kurser eller andra schemalagda aktiviteter som en gruppmedlem hade samt att se till att ingen modulgrupp behövde sitta och vänta på en annan grupp. Vi såg snabbt till att alla hade nya aktiviteter att börja med för att vara så effektiva som möjligt.

Den tredje viktigaste erfarenheten är att vi har lärt oss hur viktigt det är med kommunikationen mellan alla medlemmar. Vi gjorde en tydlig plan i början av projektet om hur vi ville att vår kommunikation skulle ske vilket lönade sig genom hela projektet. Vi tror inte att vi hade genomfört ett lika bra projekt om vi inte hade haft en bra kommunikation.

4.2 Goda råd till de som ska utföra ett liknande projekt

De goda råden som vi vill skicka med till andra som ska utföra liknande projekt är att:

- Dela upp allt arbete efter moduler men var också insatta i varandras arbete.
- Lägg mycket tid i början av projektet (planering, struktur, gruppkontrakt).
- “Communication is key”.

J TEKNISK DOKUMENTATION