

Konstruktion av en autonom undsättningsrobot

Kandidatprojekt i elektronik vid Linköpings universitet

Jesper Otterholm, Lage Ragnarsson, Erik Sköld,
Emma Söderström, Matilda Östlund Visén, Filip Östman

Version 1.1

Status

Granskad	JO, ESk, LR, FÖ	2015-05-21
Godkänd		

PROJEKTIDENTITET

2015/VT, Grupp 1

Tekniska högskolan vid Linköpings universitet, ISY

Gruppdeltagare

Namn	Ansvar	Telefon	LiU-id
Jesper Otterholm	Projektledare (PL)	073 800 03 17	jesot351
Lage Ragnarsson	Dokumentansvarig (DOK)	073 972 36 35	lagra033
Erik Sköld		073 905 43 43	erisk214
Emma Söderström		073 396 21 72	emmso236
Matilda Östlund Visén		073 817 15 90	matos000
Filip Östman		072 203 33 07	filos433

E-postlista för hela gruppen: jesot351@student.liu.se

Kund: Institutionen för systemteknik, Linköpings universitet

Kontaktperson hos kund: Kent Palmkvist, 3B:502, 013-28 13 47, kentp@isy.liu.se

Kursansvarig: Thomas Svensson, 3B:528, 013-28 13 68, thomass@isy.liu.se

Handledare: Olov Andersson, 3B:504, 013-28 26 58, olov@isy.liu.se

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2015-05-21	Första versionen	Grupp 1	JO, ESk, LR, FÖ
1.1	2015-06-03	Redaktionella ändringar	ESk, FÖ	

Sammanfattning

Syftet med denna rapport är att ge en överblick över hur projektet har genomförts samt de resultat som uppnåtts.

Målet med projektet var att konstruera en robot som kan utforska en labyrinth för att hitta en målmarkering och sedan återvända till starten när den kortaste vägen mellan start och mål är känd. Vid startområdet ska sedan ett föremål plockas upp för att levereras till målet. Utöver konstruktionen av roboten låg även stort fokus på planering och att arbeta i projektformen LIPS.

Roboten konstruerades med ett egenbyggt chassi och tre moduler: en styrmodul, en sensormodul och en kommunikationsmodul. Till detta hör en PC-modul för att visa karta och sensorvärden, samt styra roboten i manuellt läge. Roboten uppfyller alla de förstahandskrav som vi tog fram i inledningen av projektet angående funktionalitet och prestanda.

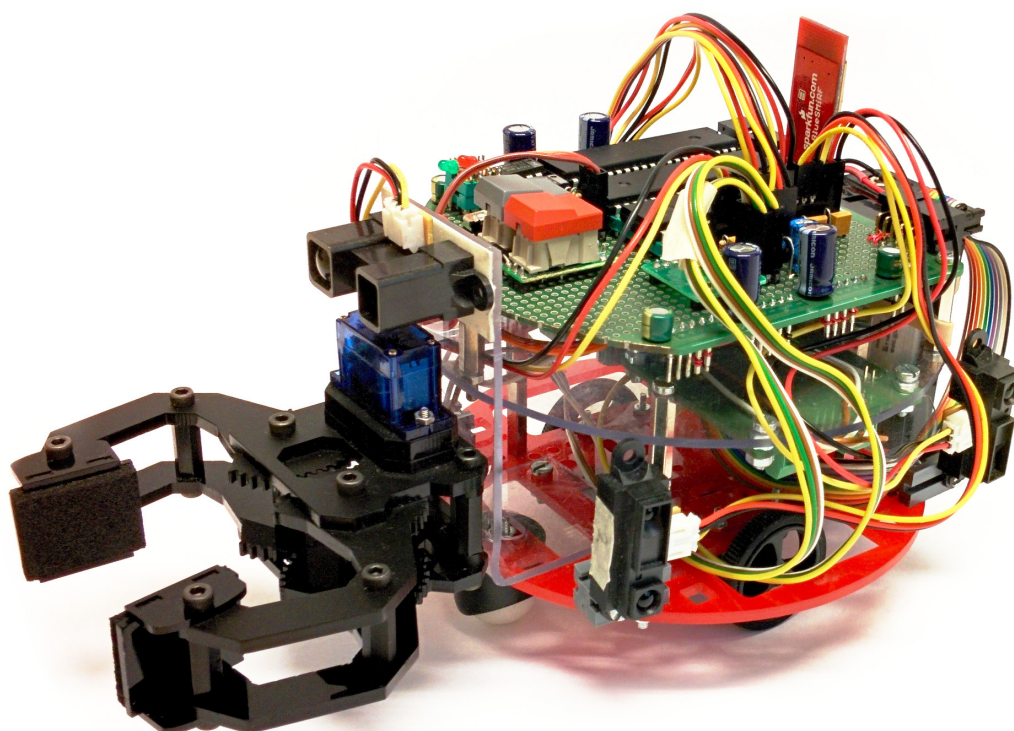
Innehåll

1	Inledning	1
2	Problemformulering	2
3	Kunskapsbas	3
4	Genomförande	3
5	Teknisk beskrivning	4
5.1	Styrmodul	5
5.2	Sensormodul	6
5.3	Kommunikationsmodul	7
5.4	PC-modul	7
5.5	Chassi	9
6	Resultat	9
7	Slutsatser	10
	Referenser	12
	Appendix	13
A	Projektdokument	13
B	Tävlingsdokument	13

1 Inledning

Denna rapport summerar kandidatarbetet *Konstruktion av en autonom undsättningsrobot* som har genomförts under vårterminen 2015 av sex studenter på civilingenjörsprogrammet *Teknisk fysik och elektroteknik* vid Linköpings universitet.

Uppgiften var att konstruera en robot som autonomt kan utforska och kartera en labyrinth, hitta en målmarkering i labyrinthen och leverera förnödenheter dit. Figur 1 visar den färdiga produkten.



Figur 1: Den konstruerade roboten Ronny¹.

Gruppen såg avsöknings- och karteringsfasen av uppdraget som den mer intressanta och lade därför fokus på att konstruera en snabb och smidig robot. Detta resulterade i valet att använda ett mindre, tvåhjuligt chassi med differentialdrift och stödkulor till skillnad från de lite större plattformar med fyra hjul som annars stod till förfogande. Det mindre chassit innebar många utmaningar med att få plats med de delar och komponenter som behövdes. Många timmar har lagts på att bygga chassidelar samt att montera komponenter på ett platseffektivt sätt.

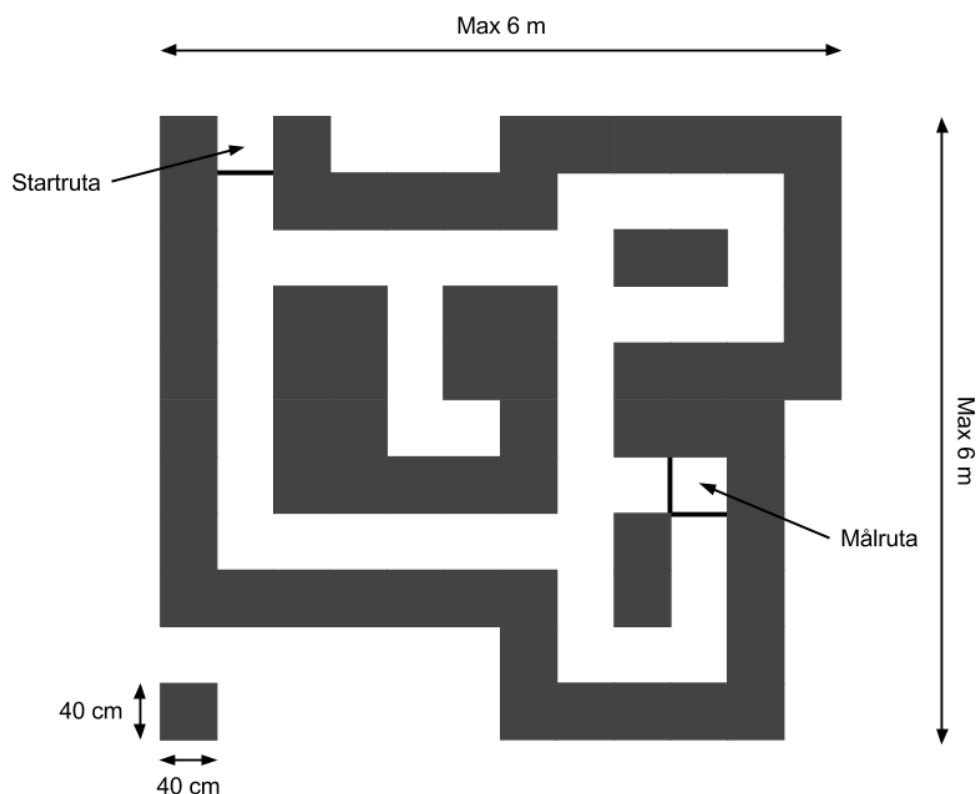
Roboten har konstruerats i moduler för att dela upp arbetet och separera olika logiska funktioner. Det finns fyra ingående moduler: styrmodul, sensormodul, kommunikationsmodul samt PC-modul. Vi beskriver här översiktligt både hårdvara och mjukvara, och hänsvisar till den tekniska dokumentationen [1] för detaljer.

¹Ronny är en akronym för RObot för autoNom Navigering i grottsYstem

I rapporten ges först en beskrivning av den problemformulering som legat till grund för arbetet, följt av en beskrivning av den kunskapsbas som utnyttjats. Därefter förklaras hur projektet har genomförts och vilka tekniska lösningar som tagits fram i konstruktionen av roboten och dess ingående moduler. Slutligen presenteras gruppens resultat och slutsatser.

2 Problemformulering

Grunden till problemformuleringen kommer från det projektdirektiv [2] gruppen erhöll av beställaren. I detta dokument efterfrågades en robot för autonom avsökning och kartering av ett grottsystem med målet att lokalisera nödställda. Då de nödställda har hittats ska den kortaste vägen mellan dessas position och startpositionen tas fram för att roboten sedan på snabbast möjliga vis ska kunna köra fram och tillbaka och transportera förnödenheter till de nödställda. Labyrinten består av ett rutnät av 40x40 cm stora rutor där var ruta antingen är farbar eller inte. Alla korridorer är en ruta bred och det förekommer 3- och 4-vägs korsningar samt loopar. Ett exempel på en möjlig labyrint illustreras i figur 2. Detaljer kring det uppdrag roboten



Figur 2: Exempel på labyrint med dimensioner.

förväntades genomföra hittas i banspecifikationen [3] samt i tävlingsreglerna [4].

Krav fanns även på en modulbaserad design för att möjliggöra testandet av alternativa lösningar på robotens olika funktionella komponenter. Dessutom efterfrågades

en kommunikationslänk till en PC för att kunna skicka telemetri under körning.

Ett viktigt moment i kursen var att ge övning i att arbeta i projektform utefter en given projektmodell och arbetet har därför bedrivits enligt LIPS-modellen [5].

Den kravbild som satts upp i samspråk med beställaren mynnade ut i en kravspecifikation [6] där projektets samtliga krav är väl kvantifierade och beskrivna.

3 Kunskapsbas

Genomförda kurser såsom *TSEA28: Datorteknik Y* och *TSEA51: Digitalteknik Y* har bidragit med den grundläggande kunskap inom digital elektronikkonstruktion och programmering av inbyggda system som varit nödvändig för genomförandet av arbetet. Under projektets gång har gruppen i stor utsträckning förlitat sig på de använda komponenternas datablad. Den typ av mikrokontroller vi har använt till de olika modulerna, *ATmega1284p*, har ett omfattande datablad [7]. Databladet har varit till stor hjälp under hela projektet.

Kunskap erhållen från kursen *TSRT12: Reglerteknik Y* bidrog till framtagningen av de regleralgoritmer styrmodulen förlitar sig på för att positionera roboten relativt väggarna i labyrinten.

Under projektets inledande fas genomfördes även en serie förstudier där områden som reglersystem och kartavsökning [8], trådlösa kommunikationsgränssnitt [9] samt olika sensoralternativ [10] undersöktes. Till dessa förstudier har i huvudsak databasen IEEE Xplore samt Linköpings universitetsbibliotek använts för informationssökning.

Kursen *TSTE05: Elektronik och mätteknik* gav goda förkunskaper om funktionen hos de vanligaste elektriska komponenterna, samt de grundläggande elektriska principerna. Mätteknikdelen av kursen gav oss goda förutsättningar för att använda ett oscilloskop vilket har varit användbart vid felsökning under hela konstruktionsfasen.

TDDC76: Programmering och datastrukturer är ytterligare en genomförd kurs som har bidragit till gruppens kunskapsbas i projektet. Programmeringsspråket C++ användes i kursen vilket har underlättat vid utvecklingen av både PC-modulen, som har skrivits i C++, och de övriga modulerna som har skrivits i C. Ytterligare erfarenheter inom områden såsom grafik och gränssnittsdesign har också varit nyttigt för PC-modulens grafiska utformning.

4 Genomförande

Kursen har genomförts i projektform med LIPS-modellen [5] som grund. Projektet har haft två stora faser: en planeringsfas och en konstruktionsfas. Under projektets två inledande månader har projektet planerats och en design för det fullständiga systemet tagits fram. Ett antal dokument har producerats under planeringsfasen: kravspecifikationen [6] fastställer den kravbild som tagits fram i samspråk med beställaren; systemskissen [11] syftade till att ge inledande designalternativ som

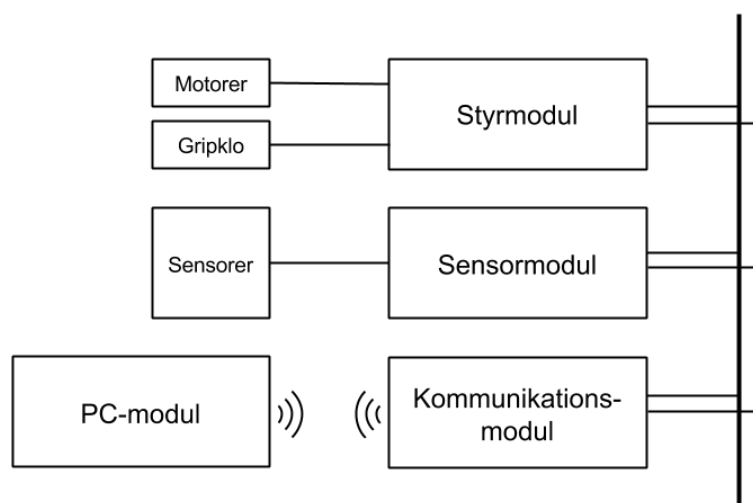
sedan låg till grund för det slutliga designvalet; designspecifikationen [12] ger en detaljerad beskrivning av systemets hårdvara; projektplanen [13], tillsammans med tidplanen, beskriver hur projektet ämnats genomföras. I och med arbetet med dessa dokument fastställdes uppdelningen av uppgifter mellan modulerna, bussgränssnitt, grundläggande mjukvarudesign samt riktlinjer för mekanisk konstruktion.

När dessa dokument var färdiga inleddes konstruktionsfasen. Arbetet delades in i de fyra delområdena styrmodul, sensormodul, kommunikationsmodul samt PC-modul, för att kunna utveckla delsystemen i roboten parallellt och föra arbetet framåt på ett effektivt sätt. När modulerna separat fungerade som förväntat fördes de samman för vidare utveckling av slutprodukten och integrationen av modulerna.

Under projektets gång har versionshanteringssystemet Git använts för att hålla god struktur på programmeringsfilernas versioner och dela dessa mellan projektmedlemmarna. Mjukvaran på robotens tre moduler utgörs av varsitt Atmel Studio-projekt men de kan öppnas tillsammans som en så kallad *solution*. PC-modulen är ett eget projekt i Visual Studio men vi har valt att samla alla dessa fyra projekt i samma Git-repo för att allt ska kunna hittas på samma ställe [14].

5 Teknisk beskrivning

Robotens moduler samverkar för att kunna dela och ta del av nödvändig information för sina respektive funktioner. Alla moduler har möjlighet att direkt dela information med övriga moduler via en I²C-buss. Modulernas relationer och samverkan med varandra och övriga komponenter illustreras i figur 3. Se den tekniska dokumentationen [1] för mer detaljer kring modulerna.



Figur 3: En översiktlig beskrivning av hur de olika modulerna samverkar med varandra och övriga komponenter.

5.1 Styrmodul

Styrmodulen är den modul som är ansvarig för roboten utför tävlingsuppdraget. För att åstadkomma detta krävs beslutsfattande gällande utforskning, kartering och navigering, samt motorstyrning, reglering och styrning av gripklo.

För att reglera, upprätta en karta och veta hur långt roboten har färdats får den sensordata från sensormodulen via en I²C-buss. Den tar in avståndsvärden till närliggande väggar för reglering såväl som kartering samt information om avlagd sträcka och rotation sedan det senaste meddelandet.

Robotens uppdrag har delats in i mindre deluppdrag och en tillståndsmaskin har konstruerats för att styra robotens beteende. Huvudloopens uppgift är därför enbart att byta till nästa tillstånd. Varje tillstånd är implementerat som en C-funktion som körs till dess att ett övergångsvillkor har uppfyllts. Exempel på tillstånd är *search_state* för att avsöka labyrinten och *return_state* för att återvända från målet till starten. Dessa tillstånd kan även anropa mindre tillstånd som mer betraktas som hjälpfunktioner. Dessa anropas som vanliga funktioner och går inte via huvudloopen. Dessa funktioner är små och relativt enkla såsom att rotera roboten 90 eller 180 grader. En detaljerad beskrivning av implementationen finns i den tekniska dokumentationen [1] och teorin beskrivs i förstudien om reglering och kartering [8].

Roboten använder två matriser för sin kartrepresentation och navigering. Den ena innehåller kartdata som roboten fyller i utifrån sina sensorer. Varje koordinat (ruta i labyrinten) kan ha värdet *UNMAPPED*, *WALL* eller *NOT_WALL*. I den andra matrisen lagras avståndsvärden som genereras av flood fill-algoritmen. Detaljer om algoritmen finns i den tekniska dokumentationen [1] och i förstudien om reglering och kartering [8].

Vi har valt att inte använda någon väggföljaralgoritm vid utforskningen då vi har antagit att det i varje korsning är lika sannolikt att målet finns i alla riktningar. Därför har vi valt att köra på ett sätt så att vi minimerar start, stopp och svängar så mycket som möjligt. Vi tror att detta är minst lika bra, om inte bättre än att systematiskt söka av labyrinten genom att följa en vägg. Dessutom kommer en ren väggföljare inte att fungera i de fall då det finns loopar i labyrinten, vilket innebär att det ändå krävs ytterligare logik för att klara avsökningen.

För att utforska labyrinten används en algoritm enligt följande:

1. Om möjligt: kör rakt fram.
2. Annars, om möjligt: sväng vänster och gå tillbaka till nummer 1.
3. Annars, om möjligt: sväng höger och gå tillbaka till nummer 1.
4. Annars, gör en flood fill till närmaste ruta som är utforskad, kör dit och gå tillbaka till nummer 1.

Under tiden som roboten kör runt och söker av labyrinten noterar den hur omgivningen ser ut med sina avståndssensorer och uppdaterar sin interna karta därefter.

När den uppdaterar sin karta skickar den även meddelande på bussen som kommunikationsmodulen vidarebefordrar till PC-modulen via Bluetooth.

När en målruta är funnen behöver roboten försäkra sig om att den kortaste vägen från start till mål är känd och kan därför behöva utforska vidare. För detta ändamål används den speciella kortastevägalgoritmen *flood_fill_home_optimistic*. Denna funktion genererar en rutt precis som de andra kortastevägenalgoritmerna med skillnaden att denna tillåter rutten att gå över utforskade rutor. På så vis fås en optimistisk gissning om vad som skulle kunna vara den kortaste vägen hem. Denna rutt kan då vara omöjlig att genomföra och därför kör roboten endast den första rutan i den genererade rutten. När den har kört en ruta uppdaterar den kartan utifrån sensorerna och gör en ny optimistisk skattning av kortaste vägen hem. Algoritmen undersöker även om den har upptäckt hinder som gör att den väg den börjat på inte längre är den optimala och backar i så fall tillbaka tills den återigen kan utforska den för tillfället bästa gissningen. Denna metod gör att vi kommer utforska de delar av labyrinten som kan tänkas ingå i den kortaste vägen och således måste inte hela labyrinten utforskas ifall målet hittas tidigt i avsökningsfasen.

Gripklon som finns monterad på robotens framkant har till uppgift att plocka upp och släppa förnödenheter. Då gruppen inte ansåg att gripklon var lika intressant som autonom navigering och kartering har den endast basfunktioner som öppna och stänga utan möjlighet att lyfta objekt från marken. Det finns inte heller något hjälpmedel i form av sensor eller liknande för att lokalisera förnödenheter vid upplockning vilket innebär att upplockningen är relativt osäker. Roboten kan behöva manuell hjälp med att positionera förnödenheterna i sin gripklo.

5.2 Sensormodul

Sensormodulen läser av och hanterar data från robotens samtliga sensorer. För att genomföra uppdraget behöver roboten kunna mäta avstånd till närliggande väggar, mäta avlagd sträcka samt upptäcka svarta tejpmarkeringar på underlaget. IR-sensorer ansågs i förstudien om sensoralternativ [10] som det mest lämpliga alternativet för avståndsmätningar i vår situation. Två sensorer på vardera sida om roboten används för reglering och kartering medan en sensor riktad framåt används för att avgöra om roboten nått slutet av en korridor.

För att avgöra hur långt roboten har färdats och för att säkerställa precision vid rotation används optiska pulsgivare monterade på motoraxlarna. I förstudien om sensoralternativ [10] undersöktes tröghetsnavigering med en IMU som alternativ men hanteringen av pulsgivare ansågs vara enklare att implementera och anpassa för den komplexitetsnivå vi önskar. Pulsgivarna genererar sinusformade ut signaler vilka via komparatorer med hysteres omvandlas till fyrkantspulser. Dessa används som externa avbrottssignaler. Detaljer kring hur dessa pulser omvandlas till millimeter avlagd sträcka och antal roterade grader finnes i den tekniska dokumentationen [1].

För att upptäcka markeringar på underlaget används en reflexsensor monterad på robotens undersida. Då tejp förekommer mycket sällan och tejpens är smal skulle man behöva sampla denna sensor ofta för att garantera att tejpens inte missas. För

att slippa detta tog vi fram en avbrottsbaserad lösning. Vi valde att använda mikrokontrollernas analoga komparator [7] för att jämföra sensorns analoga utsignal med ett referensvärde och vid lämplig övergång generera ett avbrott. För att möjliggöra kalibrering av komparatorns referensvärde används en kraftigt lågpasfilterad pulsbreddsmodulerad signal vars tillslagstid enkelt ställs in i mjukvara.

5.3 Kommunikationsmodul

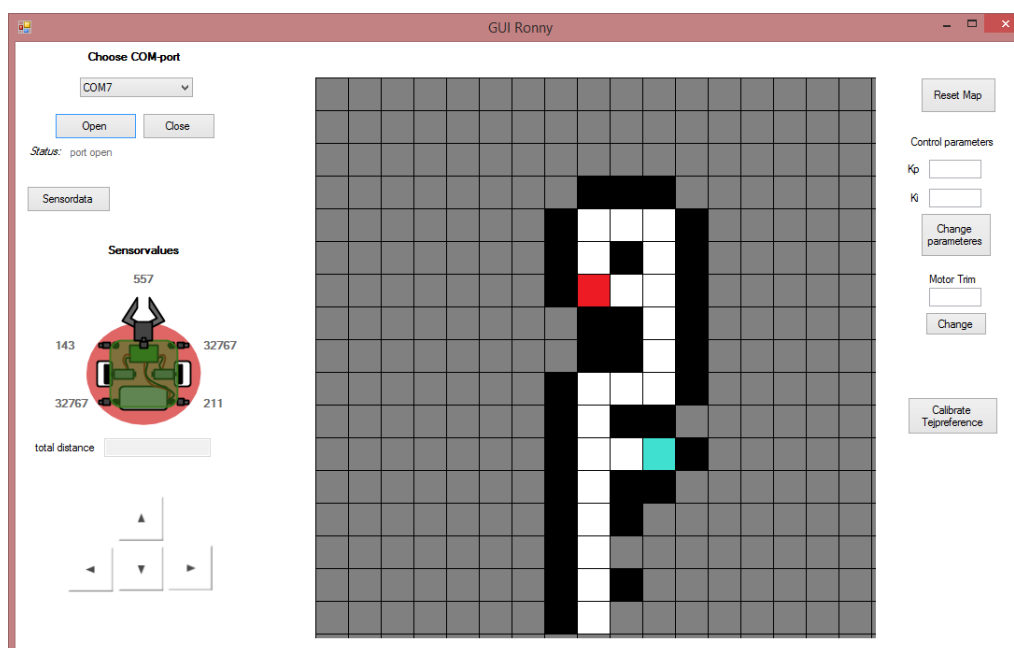
Kommunikationsmodulen fungerar som en länk mellan roboten och PC-modulen. Den interna transporten mellan de tre mikrokontrollerna sker via I²C, även kallat TWI av Atmel. Kommunikationsmodulen kommunicerar via en Bluetoothmodul trådlöst till PC-modulen. Det finns ett 20-tal olika typer av meddelanden som samtliga har blivit tilldelade en unik header bestående av en byte som skickas innan själva meddelandet. Vissa meddelanden får dock plats enbart i headern, som till exempel styrkommandon när roboten är i manuellt läge. Det största meddelandet är sensorvärden från sensormodulen där robotens fem IR-sensorer skickar tio bytes exklusive header, se tekniska dokumentationen för en mer ingående beskrivning av meddelandegränssnittet [1].

När ett meddelande mottagits använder modulen headern för att identifiera meddelandetyper och läsa av önskat antal bytes. Avläsningen sker från en buffer som fylls vid mottagningen av meddelandet. Bussen är implementerad med multimasterfunktion vilket gör att samtliga moduler kan ta kontroll över bussen om den är ledig. Vid sändning av ett meddelande fyller sändaren en buffer och startar bussen som sedan successivt läser av buffern och skickar byte för byte tills hela meddelandet är sänt. Både sändning och mottagning drivs helt med avbrott som I²C-hårdvaran triggar, vilket gör att modulerna kan hantera andra uppgifter parallellt med att en sändning pågår.

Logiken för kommunikationsmodulen är av enkel karaktär. Meddelanden som kommer från bussen skickas direkt till PC-modulen i den avbrottsrutin som I²C-hårdvaran triggar. När meddelanden från PC-modulen mottas sätts en flagga som kontrolleras i kommunikationsmodulens huvudloop och skickas vidare till korrekt modul med lämplig frekvens. Detta görs för att styrmodulen ska få styrkommandon med en förutsägbar frekvens men även för att bussen inte ska fyllas med styrkommandon.

5.4 PC-modul

Kommunikation med roboten sker via Bluetooth, vilket enligt förstudien om kommunikationsgränssnitt [9] var det bästa alternativet för trådlös kommunikation under rådande förhållanden. Inkommande data visas och utgående data styrs via ett grafiskt användargränssnitt utvecklat i Visual C++/CLI där en serieportklass finns fördefinierad. Upprättandet av serieportskommunikation prioriterades för att kunna felsöka robotens logik. Det visade sig att den fördefinierade serieportklassen var illa implementerad för just serieportskommunikation, vilket ledde till förseningar av PC-modulen. Det huvudsakliga problemet som uppstod med serieporten var varierande



Figur 4: Skärmdump av det grafiska användargränssnittet under körning.

längd på inkommande data och att serieporten inte meddelar vid varje mottagen byte. Detta löstes genom antagandet att den absolut första byten som kommer vid start av programmet är en header och vetskapen om hur många bytes som förväntas efter motsvarande header. Genom att läsa alla tillgängliga bytes i serieportens buffer, stega igenom dessa och vänta på eventuella bytes som fattas kan alla meddelanden tas emot och behandlas.

Visualiseringen av labyrinten sker genom att PC-modulen tar emot koordinater för körbar ruta (ritas ut med vit färg), vägg (svart), nødställd (röd) och nuvarande position (blå) och ersätter de grå rutorna med dessa, se figur 4. Dessa visas sedan i det grafiska användargränssnittet i ett 17×17 -rutnät. Kartdata sparas dock i en likadana kartrepresentation som används i styrmodulen, det vill säga en 33×33 -matris. Vid utritning i användargränssnittets mindre 17×17 -rutnät görs ett utsnitt ur det större 33×33 -rutnätet. Utritningen börjar därför alltid i en startkoordinat i 33×33 -rutnätet och därefter utifrån denna koordinat fylls rutnätet ut med 16 rutor i x- och y-led. Eftersom utformningen av labyrinten är okänd är startkoordinaten flexibel så att om inkommande koordinater befinner sig utanför det område som rutnätet visar kan startkoordinaten flyttas för att få med all kartinformation i utritningen. Även sensorvärden visas i användargränssnittet vid den utritade roboten.

Det finns även möjlighet att skicka kommandon till roboten i form av styrkommandon och diverse reglerparametrar. För ytterligare beskrivning av användargränssnittet, se den tekniska dokumentationen [1] och användarmanualen [15].

5.5 Chassi

Gruppen ville konstruera roboten med ett annat chassi än vad som traditionellt använts i denna kurs. Istället för de relativt stora fyrehjulsdrivna chassin som fanns att tillgå ville gruppen använda ett mindre, tvåhjulsdrivet chassi. Bakgrunden till detta var att vi ville bygga en liten och smidig robot och tyckte att de chassin som fanns att tillgå var onödigt stora. Gruppen valde att satsa på en tvåhjulsdriven robot med pulsgivare på motorerna för exakt avståndsberäkning och rotation.

På den röda bottenplattans undersida sitter två motorer och två stödkulor monterade. En stödkula sitter längst fram på roboten och en sitter längst bak. Roboten stöder dock bara på en stödkula i taget. På basplattan sitter även fästet för gripklon samt de fyra sidosensorerna monterade. På mittenplattan sitter två kort monterade: spänningsregulatorn och motordrivaren. I bakre änden av mittenplattan finns en utskärning för batteripacket så att det ska sitta stadigt. Högst upp på roboten sitter kretskortet med alla mikrokontrollerna, filter och knappar. Från detta kort dras sladdar till sensorer, gripklo, spänningsregulator och motordrivare. Robotens konstruktion kan ses i figur 1.

6 Resultat

Roboten har stöd för autonomt och manuellt läge. Via en skjutomkopplare på roboten väljs vilket läge roboten ska starta i. I autonomt läge genomsöker roboten labyrinten den placerats i till dess att de nödställda har hittats. Efter detta kontrolleras om kortaste vägen mellan start och mål är känd. Om så inte är fallet utforskas labyrinten till dess att kortaste vägen är känd varpå roboten återvänder till starten. Väl tillbaka vid starten plockar roboten upp förnödenheter med en gripklo varpå roboten transporterar och släpper av dessa vid de nödställda. Roboten återvänder därefter till starten och uppdraget är genomfört. Det finns möjlighet att följa robotens färd via ett PC-program som ritar ut robotens väg, ritar ut upptäckta väggar i labyrinten samt visar sensorvärden. I manuellt läge kan roboten styras från en PC via följande kommandon: *kör framåt*, *kör bakåt*, *rotera vänster*, *rotera höger*, *kör framåt vänster*, samt *kör framåt höger*. Då roboten befinner sig i autonomt läge kan inga styrkommandon eller reglerparametrar skickas från PC:n till roboten. För detaljerade användarinstruktioner till GUI och robot, se användarmanualen [15].

Den interna kommunikationen via I²C-bussen uppvisar ibland en viss förlust av data. Om en modul försöker skicka ett meddelande under tiden den redan sänder blockeras det andra meddelandet. Sändaren får då avgöra om det är viktigt att meddelandet går fram och får i så fall försöka igen. I manuellt läge kan efter en stund styrningen upphöra att fungera och en återställning av roboten är nödvändig. Det är i dagsläget oklart var detta problem ligger; om det är Bluetoothkommunikationen, busskommunikationen eller styrmodulen som upphör att fungera.

Vid kontrollerad körning rakt fram i korridor mäts avlagd sträcka med mycket god precision; felmarginalen ligger inom ett par procent. Avståndsmätningar till

närliggande väggar görs med en felmarginal inom 1 cm vilket är tillräckligt noggrant för robotens reglering som håller den i mitten av en korridor.

Bluetoothkommunikationen mellan robot och PC är robust vid autonom körning och ingen märkvärd mängd data går förlorad. Det grafiska användargränssnittet fungerar men det kan ibland förekomma att kartutritningen sker långsamt och inte i realtid jämfört med robotens nuvarande position.

Den färdiga produkten uppfyller samtliga grundkrav som tagits fram och specificerats i kravspecifikationen [6].

7 Slutsatser

Då flera i gruppen sedan tidigare hade särskilt intresse för de moment som ingick i projektet gick arbetet relativt snabbt framåt. Detta medförde även att flera delar hann finjusteras för ökad prestanda. Genom förstudierna gavs möjlighet att ta fram väl genomtänkta och genomarbetade val av komponenter och metoder som stora delarna av produkten består av.

Roboten som utvecklades var av, i sammanhanget, ny design vilket innebar en väsentligen mindre, smidigare och noggrannare robot än de som byggdes av andra grupper med samma uppgift. Att konstruera något helt nytt har varit en utmaning och tagit mycket tid. Många timmar har lagts på att modifiera tillgängligt material, bygga fästen och kompaktera designen för att allt skulle få plats på chassit vi valde att använda. Gripklon hade från början ytterligare ett servo för att kunna styra vinkeln gentemot marken men det monterades av för att spara plats. Ett fäste för klon och framåtsensorn byggdes av akrylplast tillsammans med mittenplattan och fästen för sensorerna, se figur 1. De batteripack som fanns att tillgå passade robotens form dåligt så ett nytt konstruerades för att passa sex stycken laddningsbara AA-batterier. De elektriska komponenterna behövde även monteras väldigt kompakt och vi har i efterhand insett att vi från början skulle ha designat ett eget kretskort. Vi hade då inte behövt separata kort för spänningsregulator och motordrivare och vi hade på så vis kunnat spara en hel del utrymme och vikt.

I efterhand inser vi att även om roboten blev betydligt mindre och smidigare av valet av nytt chassi, blev den samtidigt mer känslig för störningar från omgivningen vilket gjorde den svårare att ge en robust reglering. Olikheter i motorernas prestanda gjorde det dessutom svårt att få roboten att köra rakt fram. Ett försök att använda pulsgivarna för att uppnå rak körning utan stöd av väggar gjordes men övergavs snabbt på grund av stora svårigheter i att uppnå stabilitet och hantera numeriska fel. Att datan från pulsgivarna var tvungen att först sammanställas av sensormodulen och sedan skickas till styrmodulen på bussen gjorde det ännu svårare att uppnå önskad effekt. Med mer tid hade vi velat utforska detta ytterligare. Den slutliga lösningen blev ett trimvärde som korrigerar motorernas hastigheter. Detta värde kan ställas in från PC:n.

Bussens multimaster-funktion fungerar väsentligen som planerat men viss problematik har identifierats. Den slutliga implementationen ger till exempel inte utrymme för

snabb successiv sändning av meddelanden från en och samma modul. Detta beror på att endast en linjär buffer används vid sändning, vilket innebär att en sändare själv måste avgöra om det är värt att stanna upp och vänta tills ett tidigare meddelande har skickats klart innan sändning av den nya datan kan påbörjas. Ett försök att implementera en cirkulär buffer gjordes men övergavs efter många timmars felsökning. En mer robust busskommunikation som dessutom i större utsträckning sköter sig själv är dock något vi gärna hade jobbat vidare på om tid funnits.

Vid vidare utveckling av roboten skulle vi även kunna implementera autonom lokalisering och upplöckning av förnödenheter. Andra möjligheter inkluderar försök att åstadkomma en fungerande reglering vid högre hastigheter, eller att göra roboten mindre känslig för sensorfel och ojämnheter i banan. Om roboten hade konstruerats om på nytt hade vi övervägt att använda lite större hjul så att motorerna kunde monteras på ovansidan av bottenplattan. Detta hade skyddat motorerna och pulsgivarna bättre från damm och smuts från underlaget och därmed ökat komponenternas livslängd.

Referenser

- [1] J. Otterholm, L. Ragnarsson, E. Sköld, E. Söderström, M. Östlund Visén och F. Östman, "Teknisk dokumentation", Version 1.1, juni 2015.
- [2] T. Svensson, "Projektdirektiv för en undsättningsrobot", jan. 2015.
- [3] Grupp 1-6, "Banspecifikation", Version 1.0, febr. 2015.
- [4] —, "Tävlingsregler", Version 1.0, febr. 2015.
- [5] T. Svensson. (20 juni 2012). Projektmodellen LIPS, URL: <http://lips.isy.liu.se/> (hämtad 2015-05-18).
- [6] J. Otterholm, L. Ragnarsson, E. Sköld m. fl., "Kravspecifikation", Version 1.0, febr. 2015.
- [7] *Atmega1284p datasheet*, ATmega1284p, Rev. 8059D, Atmel, nov. 2009.
- [8] L. Ragnarsson och E. Sköld, "Förstudie, reglering och kartering för en autonom robot", Version 1.2, april 2015.
- [9] E. Söderström och M. Östlund Visén, "Förstudie kommunikation", Version 1.1, maj 2015.
- [10] J. Otterholm och F. Östman, "Förstudie, sensorer för en autonom robot", Version 1.1, maj 2015.
- [11] J. Otterholm, L. Ragnarsson, E. Sköld m. fl., "Systemskiss", Version 1.0, febr. 2015.
- [12] —, "Designspecifikation", Version 1.0, mars 2015.
- [13] —, "Projektplan", Version 1.0, febr. 2015.
- [14] —, *Ronny-the-robot*, <https://github.com/lragnarsson/Ronny-the-robot>, 2015.
- [15] —, "Användarmanual", Version 1.1, juni 2015.

Appendix

A Projektdokument

Denna rapport distribueras med en serie dokument framtagna under projektets gång. Dessa är:

Kravspecifikation En kravspecifikation framtagen av projektgruppen i samråd med beställaren. Krav på systemets funktionalitet och prestanda finns beskrivet här.

Systemskiss En skiss av produkten med olika designalternativ.

Projektplan En plan för hur projektet kommer att genomföras. Metoder, mål och arbetssätt finns här.

Tidplan En tidplan för arbetet. Baserad på aktiviteter specificerade i projektplanen.

Designspecifikation Ursprunglig hårdvarudesign för produkten. Bygger i övrigt vidare på arbetet i systemskissen.

Förstudie, reglering och kartering för en autonom robot En förstudie om hur en robot kan regleras att köra rakt i en korridor, hur den kan representera en karta samt finna kortaste vägen i labyrinten.

Förstudie, sensorer för en autonom robot En förstudie om vilka sensorer som finns, deras fördelar och nackdelar samt hur de lämpar sig för vår tillämpning.

Förstudie, kommunikation En förstudie om vilka kommunikationssätt som kan vara aktuella för vår tillämpning samt hur de fungerar.

Teknisk dokumentation Detaljerad beskrivning av den färdiga produkten.

Användarmanual Handledning för den som ska använda produkten.

Efterstudie Reflektion över arbetet i detta projekt.

B Tävlingsdokument

Denna rapport distribueras med två dokument framtagna tillsammans med de andra projektgrupperna med samma uppgift. Dessa är:

Banspecifikation Beskriver hur tävlingsbanan får utformas.

Tävlingsregler Listar de regler och måldirektiv som tävlingen utgörs av.