

Kandidatrapport

Redaktör: Erik Nybom

Version 0.1

Status

Granskad		
Godkänd		

PROJEKTIDENTITET

Grupp 1, VT14, Lagerrobot
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Karl Linderhed	Projektledare (PL)	073-679 59 59	karli315@student.liu.se
Patrik Nyberg	Dokumentansvarig (DOK)	073 -049 59 90	patny205@student.liu.se
Johan Lind		070-897 58 24	johli887@student.liu.se
Erik Nybom		070-022 47 85	eriny778@student.liu.se
Andreas Runfalk		070-564 23 79	andru411@student.liu.se
Philip Nilsson		073-528 48 86	phini326@student.liu.se
Lucas Nilsson		073-059 42 94	lucni395@student.liu.se

E-postlista för hela gruppen: tsea56-2014-grupp-1@googlegroups.com

Kontaktperson hos kund: Tomas Svensson, 013-28 13 68, tomass@isy.liu.se

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

Handledare: Anders Nilsson, 3B:512, 013-28 26 35, anders.p.nilsson@liu.se

Innehåll

1 Inledning	1
2 Problemformulering	1
3 Kunskapsbas	2
4 Genomförande	2
5 Teknisk Beskrivning	4
5.1 Chassi	5
5.2 Arm	7
5.3 Sensor	7
5.4 Kommunikation	8
6 Fördjupningsuppgifter	9
6.1 Sensoruppgiften	9
6.2 Servomotorer	9
6.3 Litium-jon-batterier	10
6.4 Linjeföljning	10
6.5 Inverterad kinematik	10
7 Resultat	10
8 Slutsatser	11
Referenser	12
A Kravspecifikation	13
B Banspecifikation	13
C Systemskiss	13
D Projektplan	13
E Tidplan	13
F Designspecifikation	13
G Teknisk dokumentation	13
H Sensoranalys	13
I Litium-jon-batterier och servomotorer	13
J Linjeföljning och reglering av robotarm	13

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2014-05-20	Första utkast.	EN	LN

1 Inledning

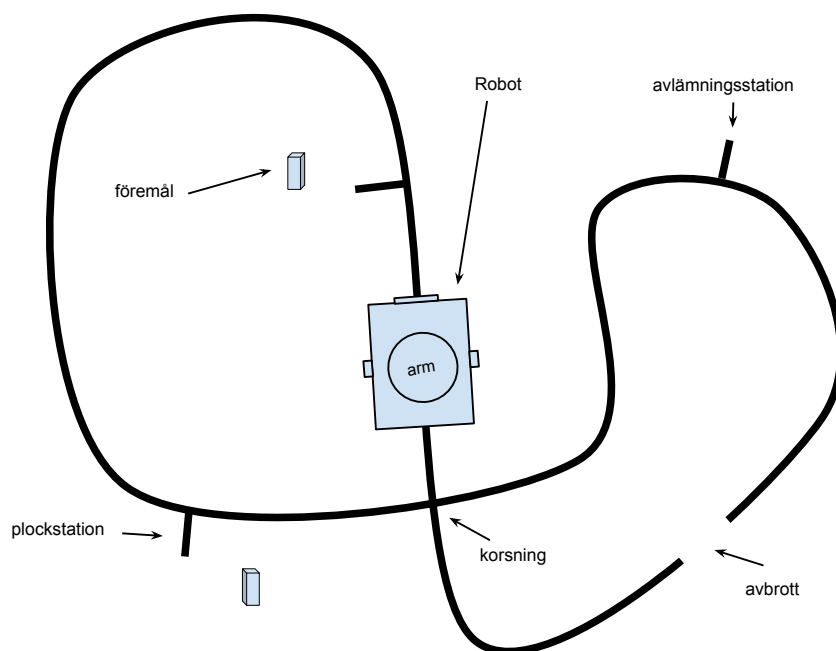
Sju studenter vid Linköpings Universitet valde i deras kandidatprojektkurs TSEA56 att konstruera en automatiserad lagerrobot utifrån beställarens krav på design och funktion. Projektet gjordes med avsikten att ge studenterna bildning och erfarenhet att med de kunskaper erhållna från tidigare kurser kunna genomföra ett omfattande uppdrag med samma metodik som förekommer i arbetslivet.

Produkten är en lagerrobot på fyra hjul med en robotarm monterad ovanpå. En linjesensor framtill används för att roboten ska kunna följa en linje bestående av svart tejp, detektera plock-och avlämningsstationer, hantera avbrott i tejpens samt hantera korsningar. Vid plockstationer söker roboten efter föremål med hjälp av en sidoskanner. Vid objektidentifiering beräknas en koordinat som armen använder sig av för att med hjälp av inverterad kinematik förflytta sig till föremålet autonomt. Roboten kan även styras manuellt i ett datorgränssnitt via blåtandskommunikation.

Projektgruppen fick robotikhårdvaran, hjulbasen med strömförsörjning och robotarmen, färdigbyggd från beställaren. Projektuppgiften bestod i att designa och utveckla kretskort med mikroprocessorer, elektronik och sensorer för att klara av ett givet uppdrag.

2 Problemformulering

Det problem som roboten i detta projekt ämnar att lösa uppstår i en tänkt lagersituation där ett antal föremål behöver flyttas runt mellan olika platser i lagret. För att roboten ska kunna navigera genom lagerutrymmet finns det en bana markerad med svart tejp på golvet. Längs med tejpens markeras de stationer där föremål ska plockas upp respektive lämnas av med hjälp av en bit tejp i en rät vinkel ut från den ursprungliga tejpelinjen.



Figur 1: Översiktlig bild av bana och robot

Roboten måste även kunna avgöra huruvida ett föremål ska plockas upp eller lämnas av vid varje station. Därför är stationerna utrustade med var sin RFID-tagga innehållandes ett för stationen unikt identifikationsnummer. RFID (Radio Frequency Identification) är en teknik som kan användas för att lagra information på små lagringsmedia (Lägg i fotnot). RFID-taggar används för att roboten ska veta ifall det finns föremål att plocka upp på stationen och vilket föremål som ska lämnas av vid vilken station. De grundläggande krav roboten måste uppfylla för att klara uppgiften inkluderar alltså bland annat att följa en linje, läsa av RFID-taggar samt att plocka upp föremål. Utöver dessa grundläggande krav är projektets tillämplighet på ett verkligt scenario i stor del avhängig på att kraven inte bara kan uppfyllas, utan även utföras autonomt. Av denna anledning så innefattar den utökade kravbilden, i kravspecifikationen markerad med lägre prioritet, i huvudsak krav vilka innebär en för roboten ökad autonomi.

En ökad grad av autonomi innebär i detta projekt huvudsakligen att roboten klarar av att plocka upp samt sätta ned föremål utan mänsklig inblandning. Detta medför att uppgiftens komplexitet snabbt ökar då kravbilden utökas förbi den grundläggande funktionalitet som i kravspecifikationen är beskriven med prioritet 1. Robotens förmåga att plocka upp ett föremål autonomt kräver inte bara förmågan att kontrollera armen, utan även ett system för att insamla information om föremålets exakta position samt att översätta och vidarebefodra denna information till armen. Vidare måste armen i sin tur kunna röra sig i rummet med en sån precision att upplockning av föremål kan göras utan att en människa kontinuerligt korregerar armens position.

3 Kunskapsbas

Majoriteten av komponenterna som använts i projektet finns dokumenterade med datablad och förklaringar på ISY:s (Institutionen för systemteknik, LiU) databladssida Vanheden (Ref) och det är i huvudsak därifrån som specifikationer och data har inhämtats. Där finner man bland annat datablad för microprocessorn ATmega 1284, som används för alla fyra delsystem i projektet.

Kunskapen som varit nödvändig för att genomföra projektet har till största del tillkommit projektgruppen under deras tidigare studier där gruppmedlemmarna bland annat har lärt sig programmering och enklare elektronikkonstruktion. Vidare har de primära källorna för ytterligare information varit de datablad som finns på Vanheden, samt projektgruppens handledare vid ISY.

I övrigt har de dokument som skapats i förstudien och uppstarten av projektet använts som stöd under det fortsatta arbetet. Särskilt har designspecifikationen legat till grund för allt det konstruktions- och programmeringsarbete som skett i projektet.

4 Genomförande

Projektet har genomförts enligt projektmodellen LIPS¹. Projektets ursprung var ett projektdirektiv från beställaren där en översiktlig beskrivning av roboten kravbilden gavs. Utifrån projekt-

¹Lätt Interaktiv Projektstyrning, en projektmodell vid Linköpings Universitet. I modellen genomförs ett projekt i tre faser: en förstudie, utförande, och en efterstudie. En projektgrupp har en tydlig projektledare, och jobbar mot en beställare som vid olika förutbestämda milstolpar tar beslut om projektets fortsatta genomförande.

direktivet inleddes projektet med att ta fram en tydlig kravspecifikation som skulle specificera exakt vad slutprodukten skulle bestå av.

Den projekttypen som skulle genomföras var ny för i år och har inte genomförts av tidigare studenter, projektgruppen fick således frihet att själva kunna experimentera med tankar och idéer för vad projektet skulle innefatta. Utifrån projektdirektivet tog gruppen fram ett antal krav på design och funktionalitet och rangordnade dem efter tre olika prioritetsnivåer, baserat på hur viktiga de bedömdes vara för att projektdirektivet skulle uppfyllas. Prioritet 1 innebar att kravet var tvunget att uppfyllas, prioritet 2 innebar att kravet borde uppfyllas i mån av tid, och prioritet 3 innebar att kravet skulle ses som ett förslag till framtida utbyggnad och förbättring. För mer detaljer, se kravspecifikationen i bilaga A.

När kravspecifikationen väl upprättats i samråd med beställaren påbörjades planeringen av projektets genomförande. En systemskiss skrevs som gav en övergripande bild av vad som krävdes utav roboten rent tekniskt. Den var upplagd som en sorts förstudie som beskrev olika möjliga designval och tekniska lösningar, för att samla in tankar och idéer på hur projektet skulle kunna genomföras, se bilaga C. Utifrån systemskissen upprättade man en projektplan och specificerade vilka aktiviteter som skulle genomföras i projektet. Alla aktiviteter organiserades i ett beroendeträd, som visar hur olika aktiviteter beror på att andra aktiviteter är genomförda, för att ligga som grund för planering av när de skulle genomföras. Projektplanen är bifogad i bilaga D. Till projektplanen upprättades också en tidplan för alla aktiviteter, som uppdaterades med jämna mellanrum under projektets gång. Denna specificerade noggrannt vem som skulle utföra vilken aktivitet när och hur lång tid den beräknades ta, se bilaga E.

Efter att projektplanen hade godkänts av beställaren och alla projektmedlemmar påbörjades själva utförandefasen av projektet med att skriva en designspecifikation. Designspecifikationen skulle ge en detaljerad bild av hur produkten skulle fungera och se ut, tillräckligt heltäckande och detaljerad för att kunna bygga hela produkten utifrån endast denna. Detta innebar att alla designbeslut både vad gäller hårdvara och mjukvara togs i arbetet med designspecifikationen på en sådan detaljnivå att konstruktion och programmering genast kunde påbörjas då detta dokument var färdigställt. Projektgruppen utgick från systemskissen och gjorde i princip en ytterligare fördjupning och förtydligande av den. Designspecifikationen återfinns i bilaga F.

Då designspecifikationen färdigställts påbörjades arbetet med att bygga roboten och skriva mjukvara. Detta arbete utfördes till stor del parallellt då arbetet bedrevs i grupper om två som utvecklade olika delsystem. Det första som gjordes var att etablera en grundläggande funktionalitet genom att direkt konstruera robotens kretskort, designa och programmera busskommunikation och funktioner för att mata ut information på robotens LCD-skärm. Parallellt med detta påbörjades inläsningen av sensordata och grundläggande kontroll av robotens hjul. Arbetet gjordes i den ordning som lagts fram i tidplanen, som tagit hänsyn till aktiviteternas inbördes beroenden, för att göra arbetet så effektivt som möjligt och minimera väntetid.

När man skriver ett mjukvaruprojekt av denna storlek (10 000 rader kod) är det viktigt att källkoden är organiserad och versionshanterad på ett bra sätt. I projektet har koden versionhanterats från första början med verktyget Git. Hemsidan GitHub(fotnot med beskrivning och url) har använts för att enkelt samordna arbetet mellan gruppmedlemmarna, och för att underlätta spårning av problem och visualisera versionshistoriken.

Mjukvaran har i huvudsak utvecklats med utvecklingsmiljön Atmel Studio utgiven av AVR-processorernas tillverkare Atmel, för den mjukvara som ligger på roboten, och utvecklingsmiljön QtCreator för persondatorns mjukvara.

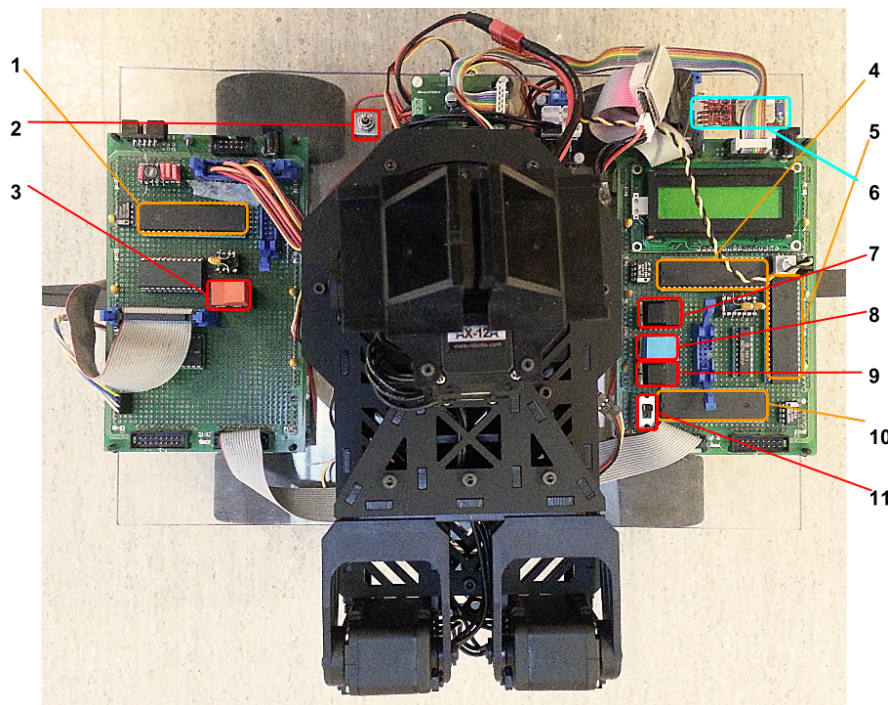
5 Teknisk Beskrivning

Roboten är uppdelad i fyra olika delsystem med varsitt ansvarsområde som vart och ett löser ett tydligt delproblem. Det första av delsystemen, kallat chassit, kontrollerar motorer och har en regleralgoritm för att se till att roboten följer linjen. Denna enhet har även det övergripande ansvaret för roboten. Det betyder till exempel att den säger till delsystem arm då det är dags att plocka upp ett föremål. Detta innebär att den också ansvarar för föremål ställs ned och plockas upp på rätt station.

Delsystem arm kontrollerar och styr de servon som sitter i robotarmen och ansvarar för funktionaliteten som krävs för att plocka upp samt sätta ner föremål vid plockstationerna.

Sensorenheten är den enhet som samlar in den information som resten av roboten använder för att ta de styrbeslut som krävs för att slutföra uppgiften. Denna enhet fungerar passivt och svarar bara på förfrågningar av andra enheter och tar inga egna initiativ för att samla information.

Kommunikationsenhetens uppgift är att kommunicera relevant information ut till omvärlden. Detta görs dels till datorn med hjälp av blåttand, och dels med hjälp av en display monterad på roboten. Även denna fungerar som en slav och ser endast till att skriva saker på displayen då någon säger åt den att göra detta. Den vidarebefordrar också data korrekt mellan robotens delsystem och en persondator via bussen samt blåttand.



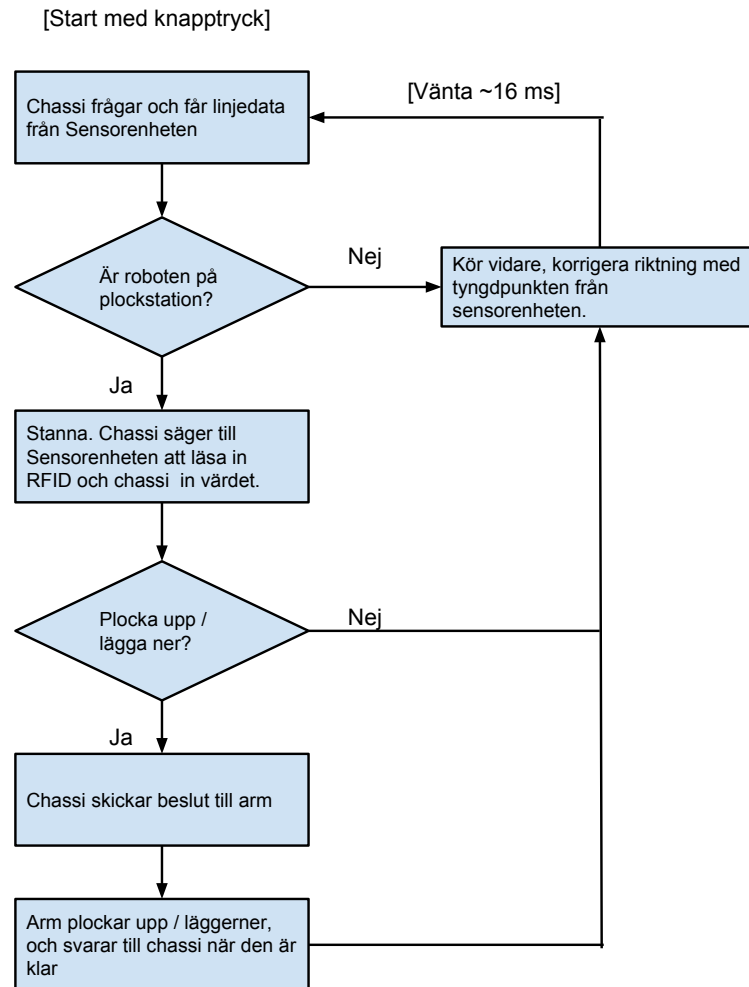
Figur 2: Förklaring av robotens delar

Nr	Komponent	Förklaring
1	Microcontroller ATmega 1284P	Sensorenhetens processor
2	Omkopplare	Strömbrytare till batteriet
3	Resetknapp	Nollställer sensorenhetens processor
4	Microcontroller ATmega 1284P	Kommunikationsenhetens processor
5	Microcontroller ATmega 1284P	Armenhetens processor
6	Bluetoothenhet BlueSMiRF Gold	För kommunikation mellan persondator och roboten
7	Resetknapp	Nollställer armenhetens och kommunikationsenhetens processorer
8	Startknapp för Auto-läge	För att starta autonom drift
9	Resetknapp	Nollställer chassienhetens processor
10	Microkontroller ATmega 1284P	Chassienhetens processor
11	Auto/manuell-omkopplare	För att växla mellan autonom och manuell styrning

5.1 Chassi

Chassienheten är den enhet som styr över de andra enheterna och bestämmer när de andra enheterna ska utföra sina uppgifter. Chassit är också den enhet som styr motorerna till hjulen. Motorerna är kopplade så att både fram och bakhjul på vardera sida får samma styrsignal, så styrningen blir liknande den på en bandvagn. Motorerna styrs genom en PWM-signal som skickas från chassienhetens processor, se bilaga G. Till chassienheten finns även en omkopplare där man kan välja mellan att roboten ska följa linjen autonomt eller om den bara ska gå att styra via persondatorn. Det finns även en tryckknapp som används för att påbörja linjeföljningsproceduren.

Chassits huvudprogram startas genom ett knapptryck eller via ett kommando från persondatorn. Då säger chassit till sensorenheten att börja söka efter en linje att följa. Ifall omkopplaren står på manuellt läge händer ingenting mer. Annars ber chassit sensorenheten med jämna mellanrum om en tyngdpunkt samt information om ifall roboten har en station till höger eller vänster. Detta görs fram tills dess att sensorenheten skickar att roboten är på station, figur 3 visar ett flödesschema över chassits program. Med hjälp av tyngdpunkten och PD-reglering beräknar chassit vilken hastighet hjulen på vardera sida ska ha för att det ska bli en mjuk lineföljning, se bilaga G.



Figur 3: Översiktlig bild av Chassits huvudprogram

När sensorenheten skickar att den har upptäckt en station stannar roboten. Sedan säger chassit till sensorenheten att läsa in RFID-taggen som förhoppningsvis ligger under roboten. Sensorenheten skickar tillbaka vilket värde RFID-taggen har. Ifall ingen tagg hittas kör roboten lite framåt och ber sensorenheten att göra en ny läsning. Hittas ingen tagg efter att roboten kört både framåt och bakåt en bit ger roboten upp och kör vidare till nästa station. Om det däremot hittas en tagg tar chassienheten beslut om vad som ska göras beroende på vilket ID taggen har. Beslutet beror på ifall det är en upplockningsstation eller avlämningsstation samt ifall roboten redan bär på något eller ifall stationen redan är avklarad.

Om roboten har stannat på en station där ett föremål ska plockas upp skickar chassit ett kommando till armen som talar om att ett objekt ska plockas upp och på vilken sida av roboten föremålet befinner sig. När armen sedan utfört sin uppgift enligt kommandot från chassit skickar armen tillbaka att den är klar och om den klarade av att plocka upp eller lägga ner objektet.

Roboten håller reda på huruvida alla stationer är avklarade genom att räkna hur många stationer den passerar innan den når samma station igen. När alla stationer är avklarade står roboten kvar på sista stationen och texten "Mission Complete" skrivs ut på LCD-displayen. Annars kör roboten vidare till nästa station. Vid varje beslutspunkt i chassits huvudprogram skickar chassit sitt beslut till kommunikationsenheten som i sin tur vidarebefodrar detta till den fjärranslutna

datorn. De olika beslutspunkterna är som följer: stanna vid plockstation, RFID inläst, skickar kommando till arm samt arm är klar.

5.2 Arm

Armen på roboten är av modell PhantomX Reactor från Trossen Robotics, med 7st AX-12A servon och 4st rörliga leder samt en gripklo. Servona skickar och tar emot data pararellt på samma kabel vilket gjorde att en tri-state buffer (fotnot: en krets som kan dela en kabel till en in- och en utport beroende på värdet på en enableport) varit nödvändig för programmet att kunna välja om den vill skicka till eller ta emot från servona. Servona kommunicerar med processorn genom USART. Varje servo har tar emot all data som skickas från processorn, men agerar endast på de kommandon som är riktade till dess eget unika ID.

Då kommandona som skickas är flera byte långa byggs dessa upp på förhand innan de skickas. Det är viktigt att det inte blir avbrott i kommunikationen för att processorn ska beräkna nästa byte. Om det inträffar skulle datan feltolkas och inte nå fram. Därför inaktiveras avbrott under själva överföringen. Vid varje sändning som riktas till ett enskilt servo fås ett svar tillbaka från med statuskod och eventuella parametrar. Detta svar används för eventuell felhantering. Eftersom gruppen valt att kunna styra robotarmen autonomt är det extra viktigt att kontrollera att varje instruktion blivit korrekt mottagen. Därför väntar programmet på svar från servona efter varje sändning gjorts. Dessa statuskoder kan sedan tolkas av programmet för att avgöra om kommandot nådde servot.

För ytterligare säkerhet gällande kommandon till servon, speciellt rörelser, har gruppen valt att endast sköta instruktioner genom funktionen reg-write som servona har. Med reg-write kan man ge ett servo en instruktion som endast utförs då ytterligare ett kommando, action-kommando, skickas till servot. Då all statusinformation finns tillgänglig kan programmet säkerställa att servona kommer agera som planerat innan rörelser utförs. Detta är väldigt användbart i de fall hela armen ska röras eftersom en led som inte rör sig kan ge en slutposition som i värsta fall skadar roboten. Vid både manuell och autonom styrning av armen beräknas ledvinklar med inverterad kinematik (Metoden för inverterad kinematik beskrivs i bilaga J). Armen kan styras manuellt i djupled, höjdled och rotation av basplatta. Gripklon kan öppnas och stängas manuellt. När armen rör sig manuellt räknas startpositionen ut och sedan adderas kontinuerligt önskad förändring så länge som armen förflyttas. Vid autonom rörelse får armenheten en vinkel och ett avstånd från basplattan till föremålet som ska plockas upp. Genom inverterad kinematik ställer armen in sig och greppar föremålet. Koordinaten som föremålet plockades upp på sparas ned och används vid avlämning för att garantera att föremålet inte ska falla omkull.

Huvudprogrammet för armenheten är passivt tills enheten anropas över bussen. Busskommandona sätter olika flaggor som huvudprogrammet kontinuerligt läser av.

5.3 Sensor

Sensorenheten har tre huvudsakliga uppgifter. Den är ansvarig för att ta fram data utifrån vilken roboten ska klara av att följa en linje samt hantera plockstationer. Vidare är dess uppgift att vid plockstationer tillhandahålla armenheten med koordinater till det objekt som ska plockas upp samt att läsa av RFID-taggar som ligger på banan i samband med plockstationerna. Information om vilken RFID-tag som ligger vid stationen vidarebefodras till chassienheten som utifrån detta avgör huruvida stationen i fråga är en plockstation eller avlämningsstation.

För att följa en linje krävs för det första att sensorenheten klarar av att detektera tejplinjen. Detta åstadkommes med hjälp av en reflexsensormodul bestående av elva separata reflexsensorer. En reflexsensor är i sin tur uppbyggd av en infraröd diod samt en fotokänslig transistor. Reflexsensormodulen är monterad längst fram på roboten ett par millimeter över golvet. När den infraröda dioden på varje reflexsensor tänds kommer detta ljus att studsas mot underlaget och ge upphov till en ström i transistorn. Ju mer ljus som reflekteras av underlaget, ju större kommer denna ström att vara. På detta sätt kan man då beräkna en tyngpunkt som motsvarar där linjen bör vara.

Sensorenhetens uppgift i robotens linjeföljning är alltså att samla in och omvandla den data som fås från reflexsensormodulen på ett sådant sätt att chassienheten kan utnyttja denna för att styra roboten längs med linjen. Detta görs genom att koppla dioder respektive fototransistorer till två stycken multiplexrar. Dessa multiplexrar styrs av sensorenhetens processor och används för att genom en A/D-omvandlare ta in data från var och en av reflexsensorerna som sitter på reflexsensormodulen. Denna data lagras sedan i processorn som ett åtta bitar stort heltalsvärde. Det innebär att varje reflexsensor kommer att registrera ett värde mellan 0 och 255, där 255 betyder mörkt underlag och 0 innebär ljust underlag.

Utifrån den data som lagrats i sensorenhetens processor beräknas sedan en tyngdpunkt på samma sätt som en linjetyngdpunkt i mekanik kan beräknas. Reflexsensorn ses som en linje där varje reflexsensors värde motsvarar tyngden för det segment av linjen den sitter på. Den tyngdpunkt som beräknas skalas till ett värde mellan 0 och 255, där 0 innebär att linjen ligger längst till vänster på sensorn och 255 innebär att linjen ligger längst till höger.

Sensorns andra uppgift, att lokalisera de objekt som ska plockas upp, görs med hjälp av två stycken sidoskannrar, en på höger respektive vänster sida av roboten. Varje sidoskanner består av ett servo av enklare modell på vilket en optisk avståndssensor sitter monterad. För att hitta objekt så stegas servot upp i små inkrement. Efter att servot har stegats upp läses avståndssensorn av. Ifall att avståndssensorn registrerar ett avstånd som är större än robotarmens räckvidd innebär detta att objektet inte finns i den aktuella vinkeln varpå servot stegas upp ytterligare.

När sidoskannern för första gången läser av ett avstånd som är mindre än robotarmens räckvidd innebär detta att ett objekt har detekteras. För att hitta objektets mittpunkt stegas nu servot upp ytterligare tills dess att ett avstånd som återigen är större än robotarmens räckvidd detekteras. Objektets mittpunkt ligger då i den vinkel och avstånd som är mitt emellan de vinklar där objektet först och sist upptäcktes. För att dessa värden ska kunna användas av armenheten måste de först omvandlas för att vara anpassade till armens koordinatsystem som har sin nollpunkt i mitten av roboten, och inte ute på sidan där sidoskannrarna sitter. Exakt hur detta görs finns ingående beskrivet i den tekniska dokumentationen, se bilaga G.

RFID-inläsningen görs genom att RFID-läsaren aktiveras och efter cirka 150 ms skickar den ett 12 bytes långt inläst ID via USART till processorn som lagrar detta värde i en buffert. Värdet i bufferten läses av och jämförs mot de ID:n som redan finns lagrade i processorns minne. Då en träff på ett av de lagrade ID:na sker med det inlästa skickas den siffra som står på RFID-taggen till chassit.

5.4 Kommunikation

Kommunikationsenhetens uppgift är att förmedla data mellan robot, dator och omgivningen. Detta sköts dels via den display som finns på roboten och dels via det blåtandsmodem som

sitter monterat på roboten. Displayen har främst använts till att felsöka roboten under utvecklingsarbetet. I den färdiga produkten har datorn istället tagit över en stor del av displayens informationsförmedlande uppgift. Displayen ger dock även i robotens slutgiltiga utförande information om plockstationer samt RFID-taggaras värde.

För att styra roboten finns ett program för en persondator där olika kommandon kan skickas till roboten. Från denna kan hjulen styras med antingen mus eller tangentbord och det går även att starta eller stoppa linjeföljning och sätta nya parametrar till regleringen. Vidare finns möjligheten att styra armen, antingen manuellt eller genom att aktivera förprogrammerade rörelser.

I persondatorgränssnittet presenteras även information som skickas från roboten. De styrbeslut som roboten tar under autonom körning presenteras i gränssnittets loggfönster och den från linjesensorn beräknade tyngdpunkten plottas i en realtidsgraf. Vidare presenteras de avstånd som fås in från sidoscannrarna i ytterliggare en graf. All data presenteras endast då den är relevant. Detta innebär att man exempelvis inte får någon graf över tyngdpunkten då roboten står vid en plockstation och inte heller någon data från sidoscannrarna då roboten rör sig mellan plockstationer.

6 Fördjupningsuppgifter

Som en del i kandidatprojektskursen utförde projektgruppen mer individuella fördjupningsuppgifter inom tre olika områden som är relevanta för projektet. Två gruppmedlemmar skrev en fördjupning om robotens sensorer (bilaga H), två skrev en uppgift om reglering av en linjeföljande robot och reglering av en robotarm (bilaga J), och tre skrev om litiumjonbatterier och om servomotorer (bilaga I).

6.1 Sensoruppgiften

I den skrivuppgift som gjordes om sensorer (se bilaga H) utreddes de olika sensorer som används till roboten. Huvudsyftet var att ta reda på om de skulle komma att vara tillräckliga för att roboten på ett tillfredsställande sätt skulle kunna klara av uppdraget. Vidare utreddes vilka alternativa sensorer som skulle kunnat användas för att få ökad prestanda med avseende på bland annat detektion av objekt.

Den primära slutsats som drogs i denna uppgift var att de val av sensorer som gjorts i designspecifikationen skulle komma att vara fullgoda för robotens förmåga att uppfylla de krav som satts i kravspecifikationen. De delar som man skulle tjäna mest på att byta ut är de servon och avståndssensorer som utgör sidoscannrarna. Mer högupplösta servon och avståndssensorer som inte sprider ljuset lika mycket skulle ge möjligheten till att mer precist detektera föremål. Detta i sin tur ger mer exakta koordinater till armen som då skulle kunna plocka upp föremål med högre precision.

6.2 Servomotorer

I den skrivuppgift som behandlade servomotorer (se bilaga I) undersöktes och redovisades hur ett servo fungerar, hur man kan få ett servo att röra sig utan kraftiga ryck och hur man styr servon av typen Dynamixel AX-12A som används i projektets robotarm. Slutsatserna som drogs av

undersökningen gav att ett servo har inbyggd återkoppling vilket gör att den alltid kommer att ställa in sig på önskad vinkel, oberoende av yttre störningar. Det framgick även att Dynamixel AX-12A har inbyggt stöd för att kunna röra sig utan kraftiga ryck och att om denna funktion inte funnits hade det varit mycket svårt att implementera detta på egen hand.

6.3 Litium-jon-batterier

I uppgiften om batterier (se bilaga I) presenterades hur litium-jon-batterier fungerar och vilka risker som finns vid laddning, transport och användning. Det presenteras vilka faktorer som påverkar ett batteris livstid negativt. Dessa är främst för stor urladdning, hög temperatur och hur laddningsströmmen regleras under laddning. För att minimera slitage ska batterierna övervakas under laddning så att spänningen inte blir för hög, samt bör laddningsströmmen begränsas till en låg ström under de första och sista 10 % av laddningscykeln.

Vidare redogörs kort för en del av den forskning som bedrivs om litiumjonbatterier och alternativ till tekniken.

6.4 Linjeföljning

I skrivuppgiften om reglering (se bilaga J) presenterades och testades en för industrin standardiserad regleringsmetod, PID-reglering. När man ska reglera insvängning för att automatiskt följa en linje så räcker det inte att styra endast med avseende på hur stor avvikelser är från mittpositionen för roboten. Detta på grund av att det väldigt lätt kan uppstå oscillerande rörelser och i vissa fall stationära fel, som oftast slutar med att roboten hamna helt ur kurs. Som verktyg för att lösa detta används PID-reglering för att styra roboten.

I dokumentet väljs en av de olika typerna av PID-reglering ut och parametrar till regleringen testas och redovisas. För att ta fram parametrar till regleraren hänvisas det till tester som gjorts på vår robot och som visat sig vara lämpliga.

6.5 Inverterad kinematik

I fördjupningsuppgiften inom inverterad kinematik (se bilaga J) beräknades en modell för en godtycklig treledad arm med roterande basplatta. Olika strategier för att genomföra inverterad kinematik diskuterades. Den valda lösningen var analytisk och visades matematiskt. En modell av armen i två dimensioner testades sedan i datorn för att se att modellen fungerade tillfredställande.

Modellen i fördjupningsuppgiften implementerades på roboten och visade sig tillräckligt exakt för att autonomt kunna plocka upp föremål från givna positioner.

7 Resultat

Projektgruppen har under projektets gång visat att en robot så som beskriven i den tekniska dokumentationen skulle vara ett fullvärdigt alternativ att användas i ett faktiskt lager. Detta har visats i de tester som utförts där roboten uppfyller alla krav med prioritet ett med god

repetierbarhet. De tester som genomförts har i huvudsak bestått av att låta roboten plocka upp och sätta ner objekt medan den har följt en upptejpad bana. För att säkerställa robotens förmåga att konsekvent uppfylla kraven har den testats på en rad olika banor designade för att tänja på de gränser som finns satta i banspecifikationen. Objektet har placerats i olika vinklar och på olika avstånd för att testa sidoskannrar och armenheten. Avbrott i tejen, kurvradie samt korsningar har hanterats i olika situationer ett stort antal gånger.

Vidare har robotens kapacitet för att helt autonomt plocka upp och sätta ner förmål enligt de RFID-taggar som ligger utlagda vid plockstationerna. All relevant sensordata vidarebefodras även till en persondator vilket möjliggör övervakning av roboten utan behov av direkt kontakt mellan människa och robot. Att roboten är autonom gör även att den är väldigt enkel att använda eftersom att den efter uppstart inte kräver någon vidare mänsklig inblandning.

8 Slutsatser

Gruppen har under projektet dragit stor nytta av det grundliga arbete som gjordes i designspecifikationen. Även om arbetet med designspecifikationen tog ansevärd mängd tid i anspråk har det sett över hela projektet lönt sig. Detta då hela gruppen redan på ett tidigt stadium i projektet haft en mycket god förståelse för hur roboten ska fungera. Då robotens delsystem senare under projektet skulle integreras med varandra var detta en stor fördel eftersom att alla medlemmar i gruppen haft en bra uppfattning av resterande medlemmars arbetsuppgifter. Mycket tack vare detta har utvecklingsarbetet till stor del kunnat utföras parallellt på de olika delsystemen.

Det gruppen är mest nöjda med vad det gäller den tekniska lösningen är den autonoma upplösningsningen. Roboten klarar av att konsekvent plocka upp föremål utan mänsklig inblandning samt att sätta ner dem utan att de välts eller tappas. Detta var från början inte något som sågs som en självklar funktion hos roboten och vidare har ingen av de andra två projektgrupperna med samma uppgift lyckats med detta.

Om mer tid gavs skulle vi främst vilja effektivisera funktionaliteten vi redan har. Något utmanande vore också att få roboten att interagera med liknande robotar i en lagermiljö. En trippmätare hade med fördel kunnat implementeras så att roboten alltid stannar exakt ovanför stationstejpen, oavsett vilken hastighet roboten håller. Beslut om ifall roboten tjänar på att vända om och köra tillbaka istället för att fortsätta framåt hade också varit en fördelaktig funktion för att ytterligare effektivisera tiden det tar för roboten att utföra sitt uppdrag.

Git och Github hade kunnat skötas snyggare och med tydligare indelning. Detta hade man kunnat löst genom att ha tagit med det tänkta upplägget i designspecifikationen och genom att se till att alla gruppens medlemmar varit mer insatta i hur Github fungerar. Om gruppen fick göra om samma uppdrag igen hade det antagligen blivit samma eller en mycket lik produkt med liknande implementationer som de som gjorts i det här projektet. Överlag tycks det svårt att utnyttja komponenterna på ett sådant sätt att roboten hade blivit avsevärt mycket effektivare i utförande av sitt uppdrag utöver implementation av de funktioner som hade gjorts om gruppen haft mer tid på sig. Viss funktionalitet hade också implementerats på ett mer generellt sätt, vilket hade kunnat effektivisera arbetet, men det hade med största sannolikhet lett till en liknande slutprodukt.

Referenser

- [1] Robotis, 2006, *User's Manual Dynamixel AX-12*
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/AX-12.pdf>, hämtad 2014-03-20
- [2] Vanheden 2003, *Servostyrning*
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/servostyrning.pdf>, hämtad 2014-03-19
- [3] Atmel Corporation, 2009, *ATmega 1284P Data sheet*
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/atmega1284p.pdf>, hämtad 2014-03-20
- [4] Anaheim Automation Inc., 2011, *Servo motor guide*
<http://www.anaheimautomation.com/manuals/forms/servo-motor-guide.php>, hämtad 2014-03-20
- [5] ISY. *Reflexsensormodulen* Hämtat från Vanheden:
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/reflexsensormodul.pdf>
- [6] Parallax Inc., 2008, *RFID Card Reader, Serial* Hämtat från Vanheden:
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/rfid-reader-v21.pdf>
- [7] SHARP. *GP2D120* Hämtat från Vanheden:
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/gp2d120.pdf>
- [8] ISY *Motorstyrning med PWM* Hämtat från Vanheden:
<https://docs.isy.liu.se/twiki/pub/VanHeden/DataSheets/pwm-motorstyrning.pdf>

- A Kravspecifikation**
- B Banspecifikation**
- C Systemskiss**
- D Projektplan**
- E Tidplan**
- F Designspecifikation**
- G Teknisk dokumentation**
- H Sensoranalys**
- I Litium-jon-batterier och servomotorer**
- J Linjeföljning och reglering av robotarm**