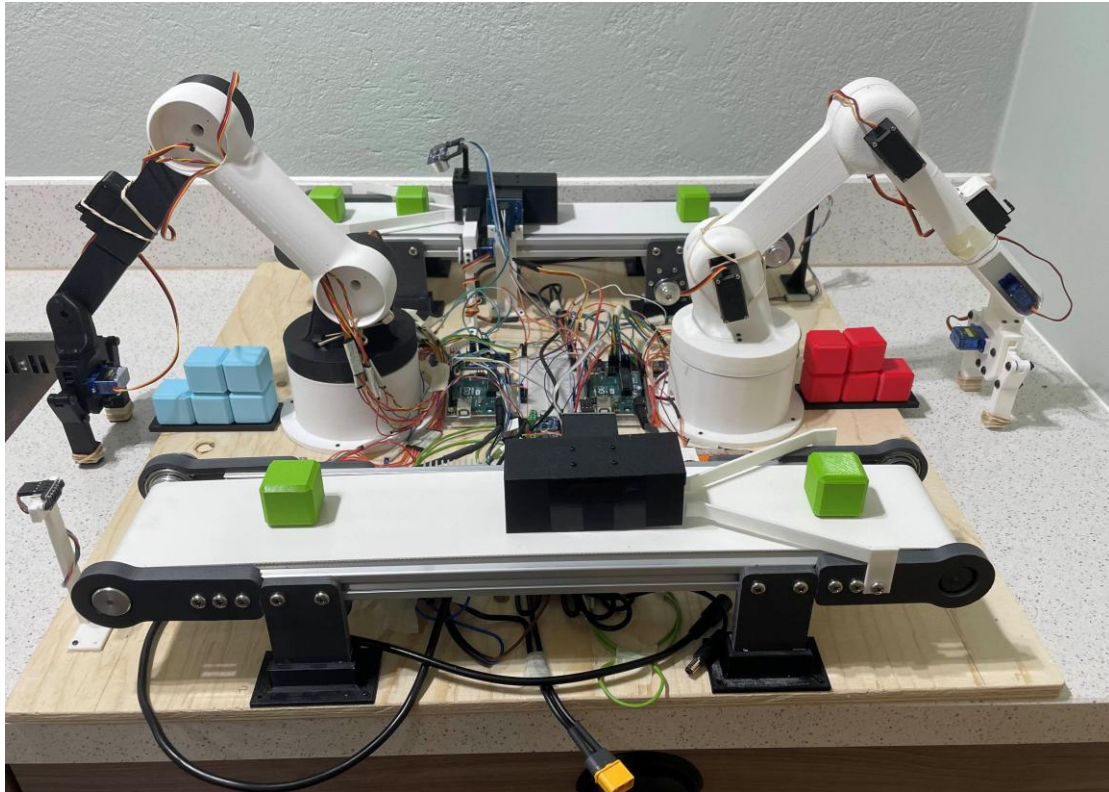




CHALMERS



Simulerad Produktionsmiljö med Robotarmar och Rullband

Utveckling av ett automatiserat system för färgsortering av lådor

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

Linus Ekedahl
Joshua Rutledge

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2025
www.chalmers.se

Examensarbete 2025

Simulerad Produktionsmiljö med Robotarmar och Rullband

Utveckling av ett automatiserat system för färgsortering av lådor

LINUS EKEDAHL
JOSHUA RUTTLEDGE



CHALMERS

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sweden 2025

Simulerad Produktionsmiljö med Robotarmar och Rullband
Utveckling av ett automatiserat system för färgsortering av lådor
LINUS EKEDAHL, JOSHUA RUTTLEDGE

© LINUS EKEDAHL, JOSHUA RUTTLEDGE, 2025.

Examensarbete 2025
Instutionen för Elektroteknik
Chalmers Tekniska Högskola
412 96 Göteborg
Sverige
Telefon +46 (0)31-772 1000

Framsida: Robotssystem med rullband

Göteborg, Sverige 2025

Sammanfattning

Automatisering är en stor del av framtida fabrikers möjlighet att effektivt kunna skapa och tillhandahålla produkter och tjänster. I takt med att industrin eftersträvar högre effektivitet och minskade produktionskostnader har automation blivit allt mer efterfrågat. Inom sorteringsmoment kan automation vara till stor hjälp för att minska beroendet av manuellt arbetet. Detta kan i sin tur leda till ökad precision inom tillverkningsprocessen samt arbetsmiljöförbättringar för de anställda.

Det här arbetet undersöker möjligheten att automatisera sortering av lådor med hjälp av två robotarmar placerade mellan två rullband. Syftet var att utveckla ett system av begränsad storlek som effektivt kunde identifiera färg på lådor, och därefter sortera dem baserat på förutbestämda kriterier. Systemet konstruerades med sensorer, servon, rullband samt egenkonstruerade robotarmar. Flera tester utfördes för att undersöka systemets precision och prestanda, däribland tester som pressade systemet till dess gränser.

Resultaten av testerna visade att systemet löste de förutbestämda kraven som ställdes. Vid höga hastigheter på rullbanden gjordes fler fel av robotarmarna, vilket indikerar att högre hastighet påverkar tillförlitligheten negativt. Utöver hastigheten på rullbanden påverkar sensorerna samt servomotorernas låga kvalitet systemets förmåga till upprepad precision.

Slutsatsen är att det är möjligt att skapa ett system med robotarmar och rullband för att sortera lådor baserat på färg. Men förbättringar samt komponenter av högre kvalitet krävs för ökad precision och stabilitet i systemet. Ett sådant system kan skalas upp och har potentiella tillämpningar inom logistik och förpackningsindustri. Systemet kan även användas i utbildningssyfte för att demonstrera principer inom områden som mekatronik och automation.

Abstract

Automation is a major part of the future of manufacturing, enabling factories to efficiently produce and deliver products and services. As the industry strives for higher efficiency and reduced production costs, automation has increased in demand. In sorting processes, automation can help reduce the reliance on manual labor, which in turn can lead to increased precision in the manufacturing process as well as improvements in the working environment for employees.

This project investigates the possibility of automating the sorting of boxes using two robotic arms placed between two conveyor belts. The goal was to develop a compact system capable of identifying the color of boxes and sorting them based on predetermined criteria. The system was constructed using sensors, servos, conveyor belts, and custom-built robotic arms. Several tests were conducted to evaluate the systems precision and performance, including tests that pushed the system to its limits.

The results of the tests showed that the system met the predetermined requirements. At higher conveyor belt speeds, more errors occurred with the robotic arms, indicating that higher speeds negatively affect reliability. In addition to conveyor speed, the systems ability to maintain consistent precision is also affected by the quality of the sensors and servos used.

In conclusion, it is possible to create a system with robotic arms and conveyor belts to sort boxes based on color. However, improvements and higher quality components are needed to increase the systems precision and stability. Such a system can be scaled up and has potential applications in logistics and packaging industries. The system can also be used for educational purposes to demonstrate principles in fields such as mechatronics and automation.

Keywords: Automation, Robotic arms, Conveyor belts, Color sorting, Mechatronics, Industrial automation, Sensors, Servo motors, Sorting system, Manufacturing efficiency

Förord

Detta examensarbete har utförts av två högskoleingenjörstudenter på Chalmers inom programmet mekatronik. Examensarbetet omfattar 15 högskolepoäng och genomfördes under vårterminen 2025 i samarbete med företaget Broccoli Engineering AB.

Vi vill tacka våra handledare Björn Bergholm och Ola Ljunggren för deras stöd och vägledning under arbetets gång. Slutligen vill vi även tacka vår examinator Bertil Thomas för hans stöd, engagemang och värdefulla synpunkter.

Linus Ekedahl och Joshua Ruttledge, Göteborg, 2025

Terminologi och Förkortningar

Nedan följer en lista av förkortningar som använts under arbetets gång.

| | |
|-----------------------|--------------------------|
| I²C | Inter-Integrated Circuit |
| ToF | Time of Flight |
| RGB | Red Green Blue |
| IR | Infraröd |
| PWM | Pulsbreddsmodulering |
| DC | Likström |
| AC | Växelström |
| CAD | Computer-Aided Design |

Innehåll

| | |
|---|------|
| Terminologi och Förkortningar | viii |
| 1. Inledning..... | 1 |
| 1.1 Bakgrund | 1 |
| 1.2 Syfte | 1 |
| 1.3 Avgränsningar | 1 |
| 1.4 Precisering av frågeställningen | 1 |
| 2. Teoretisk bakgrund | 2 |
| 2.1 Avståndssensor | 2 |
| 2.2 Färgsensor | 2 |
| 2.3 PCA9685 Servodrivare..... | 2 |
| 2.4 Motordrivare..... | 2 |
| 2.5 Buck-omvandlare | 3 |
| 2.6 I2C | 3 |
| 2.7 Inverterad Kinematik | 3 |
| 2.8 3D-Skrivare..... | 3 |
| 2.9 Ultraljudssensor..... | 4 |
| 3. Metod | 5 |
| 3.1 Metodansats | 5 |
| 3.2 Projektupplägg och arbetsgång | 5 |
| 3.3 Val av tekniker och komponenter..... | 5 |
| 3.4 Systemöversikt och designprinciper | 6 |
| 3.5 Utvärderingsmetod | 6 |
| 4. Konceptgenerering och val av komponenter | 7 |
| 4.1 Val av sensorer | 7 |
| 4.2 Val av servon | 7 |
| 4.3 Val av styrenhet | 8 |
| 4.4 Rullband..... | 8 |
| 4.5 Servodrivare..... | 8 |
| 4.6 Strömförsörjning..... | 9 |
| 4.7 Buck-omvandlare | 10 |
| 5. Konstruktion och design | 11 |
| 5.1 Systemlayout..... | 11 |
| 5.2 3D-modellering och komponentdesign..... | 12 |
| 5.2.1 Bottenkopp | 12 |
| 5.2.2 Toppkopp | 12 |
| 5.2.3 Arm 1..... | 14 |

| | | |
|-------|--|----|
| 5.2.4 | Arm 2..... | 15 |
| 5.2.5 | Arm 3..... | 16 |
| 5.2.6 | Gripklo..... | 17 |
| 5.3 | Elkonstruktion..... | 19 |
| 6. | Programmering och styrning..... | 20 |
| 6.1 | PWM och servon..... | 20 |
| 6.2 | Invers Kinematik..... | 20 |
| 6.3 | Programmering av gripklo..... | 21 |
| 6.4 | Sensorprogrammering..... | 22 |
| 6.5 | Rullbandstyrning..... | 22 |
| 6.6 | Huvudprogram..... | 23 |
| 7. | Iteration och förbättringar..... | 24 |
| 7.1 | Färgsensor förbättring..... | 24 |
| 7.2 | Fördelare..... | 24 |
| 7.3 | Grepp problem..... | 25 |
| 7.4 | Arm 1 gummiband..... | 25 |
| 8. | Testning och verifiering..... | 26 |
| 8.1 | Test av färgigenkänning..... | 26 |
| 8.2 | Test av avståndsmätning..... | 26 |
| 8.3 | Test av lådor på olika positioner..... | 26 |
| 8.4 | Test av robotarmarnas precision..... | 27 |
| 8.5 | Minimal cykeltid..... | 27 |
| 8.6 | Funktionstest..... | 27 |
| 9. | Resultat..... | 28 |
| 9.1 | Färgigenkänning..... | 28 |
| 9.2 | Avståndsmätning..... | 29 |
| 9.3 | Riktare..... | 29 |
| 9.4 | Robotprecision..... | 31 |
| 9.5 | Cykeltid..... | 31 |
| 9.6 | Totalt systemtest..... | 32 |
| 10. | Slutsats och Diskussion..... | 34 |
| 10.1 | Svar på frågeställningar..... | 34 |
| 10.2 | El och ström..... | 35 |
| 10.3 | Konstruktion..... | 35 |
| 10.4 | Servon..... | 36 |
| 10.5 | Sensorer och mjukvara..... | 37 |
| 10.6 | Etik och Hållbarhet..... | 38 |

| | |
|------------------|----|
| Referenser | 39 |
| Bilagor | 41 |
| Bilaga 1 | 41 |
| Bilaga 2 | 42 |
| Bilaga 3 | 43 |
| Bilaga 4 | 44 |
| Bilaga 5 | 45 |

1. Inledning

Denna rapport kommer att undersöka konstruktionen och utvecklingen av robotarmar som används för att flytta lådor mellan rullband. I detta kapitel kommer bakgrund, syfte, avgränsningar och frågeställning behandlas.

1.1 Bakgrund

Automatisering är en växande del av modern industri, där robotar kan användas för att öka effektivitet och minska behovet av manuellt arbete. Robotarmar är ett av de mest anpassningsbara och flexibla verktygen inom automation och kan programmeras för att utföra ett brett utbud av uppgifter. Ett vanligt problem inom tillverkningsindustri är att manuell sortering är tidskrävande, dyrt och kan leda till fel, speciellt när det handlar om stora mängder [1]. Genom att göra sorteringen automatisk kan man uppnå högre effektivitet och precision. Med detta vill vi tillsammans med Broccoli Engineering utforska möjligheten att simulera och utveckla en automatiserad lösning på detta sorteringsproblem. De vill också skapa en funktionell modell som går att ta med och visa upp på arbetsmässor. Detta arbete kommer att utveckla ett automatiserat system som kan sortera lådor som går på ett rullande band utifrån färg med hjälp av robotarmar.

1.2 Syfte

Syftet med arbetet är att konstruera en modell av en fabrik som inkluderar robotarmar, rullande band och lådor. Denna modell ska undersöka hur robotarmar kan användas för att effektivisera och automatisera ett system som flyttar och sorterar lådor baserat på deras färg. Modellen ska även vara portabel och möjlig att tas med på mässor för demonstration.

1.3 Avgränsningar

Följande avgränsningar har fastställts för projektet:

- Rullbanden kommer inte att tillverkas av oss utan köpas in som en färdig komponent.
- Storleken kommer att begränsas till ca 60x60 cm för hela modellen.
- Robotarmarna behöver inte plocka upp lådor utanför den förutbestämda upphämtningszonen.
- Robotarmarna ska initialt inte plocka upp lådor från rullbandet när det är i rörelse, men om tiden finns kan möjligheten utforskas.

1.4 Precisering av frågeställningen

Under arbetets gång ska dessa frågor besvaras:

- Hur kan robotarmar användas för automatisk sortering av lådor utifrån färg?
- Är det möjligt att konstruera en robotarm som på ett tillförlitligt sätt klarar av att plocka upp lådor från olika platser i ett område?
- Hur påverkar rullbandets hastighet systemets prestanda och noggrannhet?
- Vad kan man göra för att hantera överflöd av lådor på rullbandet?
- Hur påverkar olika faktorer som sensorers noggrannhet och den mekaniska konstruktionen systemets prestanda?
- Hur väl presterar det framtagna systemet, samt vilka begränsningar har den?

2. Teoretisk bakgrund

I detta kapitel beskrivs teorier och komponenter som har använts i projektet.

2.1 Avståndssensor

Avståndssensorn som används i projektet är av typen VL53L0X Time of Flight-Sensor (ToF). Sensorn fungerar genom att mäta tiden det tar för en infraröd (IR) ljusstråle att studsas på mätobjektet och sedan tillbaka till sensorn [2]. Time of Flight tekniken bygger på konceptet att ljus färdas med konstant hastighet, och därmed kan avståndet till föremålet beräknas med formeln i ekvation 1.

$$D = \frac{c \cdot t}{2} \quad (1)$$

Där:

- D är avståndet till föremålet
- c är ljusets hastighet
- t är den uppmätta tiden

Sensorn använder ett I²C-gränssnitt för att skicka data till mikrokontrollern [2]. Sensortypen är ett kostnadseffektivt och energisnålt val som kan användas i en mängd olika applikationer för att mäta avstånd.

2.2 Färgsensor

I projektet används en färgigenkänningssensor för att avläsa färgen på lådorna. Sensorn är av typen TCS3200 som använder ett TAOS TCS3200 RGB (Röd Grön Blå) sensorchip för att avläsa färg [3].

Sensorn fungerar genom att läsa av reflekterat ljus från ett objekt och sedan omvandla det till en frekvenssignal [3]. Sensorn består av ett rutnät av fotodioder, där vissa är känsliga för rött, vissa för grönt och vissa för blått ljus. Dessa fotodioder går att aktivera separat från varandra genom styrsignaler, vilket möjliggör mätning av ljusintensiteten för varje komponent separat. Ljuset omvandlas sedan till en fyrkantsvåg med en frekvens som är proportionell mot ljusintensiteten. Sensormodulen är även utrustad med fyra LED-lampor som är till för att belysa objektet där färg ska detekteras. Detta görs för att stabilisera ljusförhållandena samt öka mängden reflekterat ljus från objektet för att få en så bra avläsning som möjligt.

2.3 PCA9685 Servodrivare

PCA9685 är en pulsbreddsmodulerings-drivare (PWM-drivare) med 16 kanaler som gör det möjligt att styra flera PWM enheter samtidigt [4]. Den kommunicerar med mikrokontroller via I²C. Denna drivare kan användas för att styra flera servomotorer samtidigt och genererar PWM-signaler oberoende från en mikrokontroller, detta gör att den är icke-blockerande.

2.4 Motordrivare

Adafruit DRV8871 är en motorstyrningsenhet som är baserad på Texas Instruments DRV8871 H-brygga som möjliggör enkel motorstyrning av DC-motorer [5]. Modulen är till för att kunna styra både hastighet samt rotationsriktningen av DC-motorer. H-bryggan ändrar rotationsriktningen genom att växla polariteten på spänningen över motorn, medan en pulsbreddsmodulerad (PWM) styrsignal justerar rotationshastigheten genom att ändra spänningsnivån till motorn. Motordrivaren gör det även lätt att driva motorn

med en extern strömkälla, vilket är nödvändigt då strömmen som kommer från mikrokontrollern ofta är för låg för att driva motorn. Detta möjliggör att man endast behöver använda två styrsignaler från mikrokontrollern samt extern strömförsörjning för att styra motorn.

2.5 Buck-omvandlare

En buck-omvandlare eller nedkonverterare är en likström till likström konverterare som sänker spänning samtidigt som den höjer strömmen från ingången till utgången [6]. Effektiviteten av en buck-omvandlare kan vara väldigt hög, oftast över 90% vilket gör dem användbara när man har en krets med olika spänningsnivåer och bara en spänningsmatning.

2.6 I²C

Inter-Integrated Circuit (*I²C*) är ett kommunikationsprotokoll som kan användas för att överföra data och information mellan delar av ett system [7]. *I²C* kan användas för att överföra data mellan exempelvis en sensor och en mikrokontroller. En *I²C* kommunikationsbuss består av två ledningar, en Seriell datalinje (SDA) och en seriell klocklinje (SCL). *I²C* gör det möjligt att koppla flera enheter till samma kommunikationsbuss eftersom det är en seriell bussteknik, där varje enhet har en unik adress.

2.7 Inverterad Kinematik

Inom robotik är inverterad kinematik en matematisk modell som används för att beräkna vinklarna som en led behöver ha för att nå en bestämd ändlägesposition [8]. I en kedja av leder på en robotarm kan varje leds vinkel beräknas med hjälp av kinematiska ekvationer samt trigonometri. Med de beräknade vinklarna kan man uppnå en estimerad position på ändeffektorn som exempelvis kan vara en gripklo.

Ändläget som ska uppnås kan ges i formen (x,y,z) där variablerna x, y och z, representerar avståndet från ett bestämt nolläge i varje axel givet i millimeter. Dessa variabler är det som matas in i de kinematiska ekvationerna för att sedan ge vinklar som lederna ska ställas till.

2.8 3D-Skrivare

3D-utskrivning är en additiv tillverkningsprocess som skapar ett objekt genom att skriva ut ett material lager för lager. Processen börjar med användaren skapar en CAD-modell, som sedan bearbetas i en slicer (skivningsprogram). Slicern omvandlar CAD-modellen till G-kod, vilket är det format som 3D-skrivaren kan tolka och använda för att styra utskriften.

I projektet används 3D-skrivare av typen smältlagringstillverkning (Fused Deposition Modeling, FDM). FDM-skrivare är den vanligaste typen av 3D-skrivare och fungerar genom att smälta en plasttråd (Filament) genom ett varmt munstycke (Nozzle) [9]. Den smälta plasten byggs upp lager för lager i enlighet med CAD-modellen på skrivarens byggplatta.

Det går att använda ett antal olika Plasttyper i en FDM-skrivare, bland annat PLA (Polylaktid), ABS (Akrylnitril-butadien-styren), PETG (Polyetylentereftalatglykol), TPU (Termoplastisk polyuretan) [10]. Det finns även vissa kompositmaterial, så som Kolfiberförstärkt PLA, där kolfiber används för att öka hållfastheten. Alla olika

plasttyper har sina materialspecifika egenskaper som gör materialen mer eller mindre användbara i olika applikationer.

2.9 Ultraljudssensor

Ultraljudssensorer används i en rad olika tekniska sammanhang när man vill mäta avstånd utan fysisk kontakt med mätobjektet. Som namnet indikerar använder ultraljudssensorer ultraljudsvågor för att mäta avstånd. Vågen skickas ut för att sedan studsas på mätobjektet tillbaka till sensorn [11]. Tiden det tar för vågorna att återvända används för att beräkna avståndet enligt formeln i ekvation 2:

$$d = \frac{v * t}{2} \quad (2)$$

Där d är avståndet till objektet, v är ljudets hastighet i luft och t är den uppmätta tiden mellan utsändning och mottagning av signalen. Ultraljudssensorer är särskilt användbara när det kommer till att mäta avstånd i miljöer där optiska sensorer kan vara mindre effektiva, så som vid dåliga ljusförhållanden eller dammiga miljöer [11].

Ultraljudssensorer används inom bland annat parkeringsassistansen på bilar eller objektidentifiering i industriella produktionslinjer.

3. Metod

Detta kapitel beskriver metoden och tillvägagångssätten som har använts för att utveckla robotarmar med fem frihetsgrader som kan plocka lådor från rullband och sortera dem utifrån färg.

3.1 Metodansats

Projektet Simulerad produktionsmiljö med robotarmar och rullband har genomförts som ett ingenjörprojekt med praktisk tillämpning. Fokuset i detta projekt låg på att utforma, bygga och testa ett automatiserat sorteringsystem där robotarmar, rullband samt olika sensorer tillämpades för att sortera lådor baserat på färg. Arbetsmetodiken som har använts har varit iterativ, där de olika delsystemen för sig har testats och utvärderats. Detta stegvisa arbetssätt och testning av delsystem för sig skulle i slutändan leda till ett så välfungerande helhetssystem som möjligt utifrån kravspecifikationen.

3.2 Projektupplägg och arbetsgång

Projektet delades upp i 4 faser: uppstartsfas, koncept och design, programmering samt byggnation och testning. Därefter gjordes en tidsplan i form av ett GANTT- Schema där tidsåtgången för de olika faserna delades upp. Efter det togs en kravspecifikation fram där krav och önskemål från projektgruppen samt företaget skrevs ner. Detta för att det skulle vara enkelt att se vad projektet skulle uppnå, se bilaga 1 för GANTT-schemat och kravspecifikation. Under uppstartsfasen gjordes även en planeringsrapport samt inhämtning av nödvändig information inför projektets start.

3.3 Val av tekniker och komponenter

Typen av robotarm som valdes är en ledad 5-frihetsgrad robotarm. De tre första frihetsgraderna behövdes för att kunna röra robotarmarna i tre dimensioner. De två sista frihetsgraderna är pitch (nigning) och roll (rullning) och används för att manipulera robotarmens ändeffektor till önskad vinkel och position. Med dessa frihetsgrader kan alla positioner nås, men inte alla orienteringar av ändeffektorn. Den frihetsgraden som utelämnades är yaw (girning). Den valdes bort för att göra designen simplare och för att den inte var nödvändig för robotarmens syfte.

Servomotorer valdes för att driva robotarmarnas alla delar. För att välja lämpligt starka servon gjordes beräkningar på maximal last varje led kan ha. För robotarmarna är precision väldigt viktigt och därför var servon lämpliga då de kan ställas till en exakt vinkel och hålla den positionen under last. Denna form av kontroll, i kombination med servomotorers höga vridmoment är avgörande för robotik och därför valdes servon som drivande enhet.

Arduino Uno valdes som styrenhet för projektet. Detta var främst på grund av tidigare erfarenhet, dess användarvänlighet och att den uppfyllde alla krav som fanns på styrenheten. Raspberry Pi och ESP32 övervägdes också som alternativ. Dessa styrenheter erbjöd mer avancerade funktioner men kom även med en brantare inlärningskurva.

Robotarmarna behövde kunna interagera med systemet på ett meningsfullt sätt och utrustades därför med relevant indata. Utifrån systemets krav valdes två typer av sensorer: färgsensorer och avståndssensorer. Färgsensorns syfte var att förse robotarna med information om objektets färg för att möjliggöra sortering. Avståndssensorn gav information om var på rullbandet ett objekt befann sig för att möjliggöra en autonom upphämtning av lådor. Valet av sensorer motiverades med att ge robotarna den minimala

mängden indata som behövdes för att de skulle kunna arbeta korrekt och autonomt.

Robotarmarna bestämdes tidigt att de skulle skapas med hjälp av 3D-skrivare av typen FDM (Fused Deposition Modeling). Valet av tillverkningsmetod baserades på tidigare erfarenheter samt att hårdvaran fanns tillgänglig. Tillverkningsmetoden gjorde det även billigt att ta fram prototyper, och tillverkningstiden är kort. Under informationsinhämtningen noterades även att 3D-skrivare användes i hög utsträckning av både hobbyister och professionella för att producera robotarmar [12]. 3D-skrivare användes även för ytterligare delar som exempelvis tillverkning av monteringsstativ för både färgsensorer och avståndssensorer. De användes även för riktaren som slussar in lådorna till mitten av bandet så att lådorna alltid fick samma position i sidled, oberoende av initial placering av lådorna på bandet.

3.4 Systemöversikt och designprinciper

Det övergripande systemet består av två robotarmar som sorterar och förflyttar lådor som åker på två parallella rullband. Systemet består av två lika delsystem, där ett delsystem innefattar en robotarm, ett rullband, en färgsensor, en avståndssensor, en Arduino uno och drivarkretsar. Flödet var upplagt så att användaren sätter en låda på rullbandet, sedan åker lådan igenom riktaren som slussar in lådan i mitten av bandet om den inte stod där från början. Efter riktaren läses färgen på lådan av, vilket bestämmer vad lådans slutdestination är. Sedan stannar lådan vart som helst inom en upphämtningszon där avstånd läses av och lådan hämtas.

Beroende på färg sker följande hantering:

- Röd: Hämtas av robotarm 1, och lämnas på avlämningsplats 1.
- Grön: Flyttas över till Rullband 2 och fortsätter i systemet utan att lämnas av.
- Blå: Placeras på rullband 2 av robotarm 1, varpå robotarm 2 placerar lådan på avlämningsplats 2.

Denna process pågår så länge systemet är i gång och användaren sätter på lådor på rullbandet, vilket skapar en simulerad produktionsmiljö med hantering av lådor baserat på färg. Ett flödesschema på systemets process kan ses i bilaga 2.

3.5 Utvärderingsmetod

För att utvärdera projektet och besvara frågeställningarna bestämdes kriterier och testmetoder. Kriterierna som användes är följande: färgsorteringens noggrannhet, distanssensorns noggrannhet, cykeltid och robotarmarnas noggrannhet. För att mäta och utvärdera dessa kriterier valdes ett antal tester och mätmetoder. Mätmetoderna var följande: tidtagning av cykler, loggning av färgigenkänning, kontroll av distanssensorns mätningar och analys av robotens rörelser. Testerna som gjordes var att mata in lådor av samma färg, ställa lådor på olika avstånd framför distanssensorn, stresstesta systemet med många lådor av olika färger, mata in lådor på olika positioner av rullbandet, köra systemet på olika bandhastigheter och olika hastigheter av robotarmarna.

De tidigare nämnda kriterierna, mätmetoderna och testerna valdes för att skapa en helhetsbild av systemets prestanda. De är relevanta eftersom de är fysiskt kopplade till vad systemets syfte är. Fler kriterier och tester hade kunnat göras för att ytterligare utvärdera, men de som valdes ut var de mest kritiska och relevanta.

4. Konceptgenerering och val av komponenter

Nästa steg blev att ta fram olika koncept för hur projektets mål skulle nås. Detta innefattade systemkoncept, där olika idéer på placering samt hur hela systemet skulle utformas. Det togs även fram ett antal olika koncept för hur robotarmarna skulle konstrueras, detta gjordes först individuellt, för att sedan i grupp diskuteras och hitta vad som var bra, respektive mindre bra med varje koncept. På så sätt följde de bra attributerna med varje konceptiteration medan de sämre attributerna utelämnades, vilket ledde till en så optimal design som möjlig.

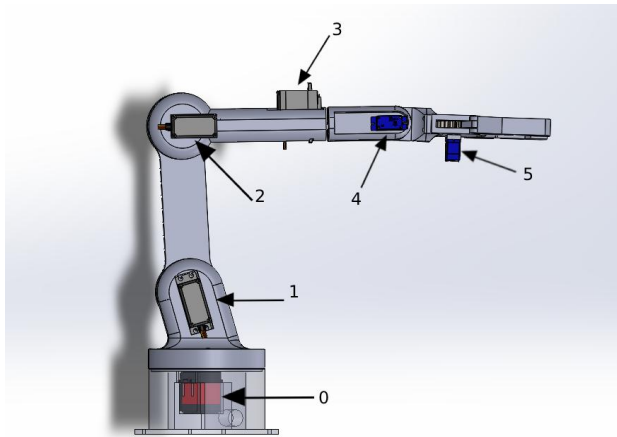
4.1 Val av sensorer

Redan i uppstartsfasen bestämdes vad för slags sensorteknik som behöves för projektet, en sensor för att läsa av färg på lådorna och en för att läsa av avstånd på lådorna. Då det finns många olika sensorer för detta börjades en informationssökning angående vilken sensor som var mest optimal för just detta projekt. I detta val gjordes forskning angående noggrannhet, pris samt tillgänglighet för ett antal olika sensorer. Valet landade på färgsensorn TCS3200 då det utifrån informationssökningen visade att sensortypen var vanlig och lätt att få tag på, den var billig samt att den skulle vara tillräckligt noggrann för projektet. Avståndssensorn som valdes var av typen VL53L0X Time of Flight-Sensor (ToF), detta då även denna var tillräckligt noggrann, billig samt att den var lätt att få tag på.

4.2 Val av servon

Förutom sensorer behövdes även servomotorer för att driva robotarmarna. Varje robotarm krävde sex servon, men då alla servon sitter på olika utsatta positioner krävdes olika prestanda beroende på positionen. För att göra rätt val av servon utfördes momentberäkningar för varje servoposition. Då exakt vikt och längd inte var känt gjordes beräkningarna med högre marginal för att vara på den säkra sidan. Valen av servon backades sedan upp av en informationssökning som visade att liknande projekt och konstruktioner använde sig av liknade servon.

Valet av servon för position 0 (se Figur 1), landade på två DIYMORE-TOOL 8120MG. Dessa servon kunde leverera en kraft på 20 Kg, och hade ett vinkelområde på 270 grader. Då båda robotarmarna skulle ha möjlighet att nå båda rullbanden, samt hela upphämtningszonen för det tillhörande rullbandet krävdes ett servo i botten som klarade av ett vinkelområde om minst 270 grader. För position 1 (se Figur 1), krävdes ett servo med 180 graders vinkelområde. Denna position var den mest utsatta gällande vridmoment, och därför valdes DIYMORE-TOOL DT996 som kunde leverera en kraft på 15 Kg. För position 2 och 3 krävdes något svagare servon men fortfarande relativt kraftfulla. Valet landade på Az-Deliverys MG996R som hade ett vinkelområde på 180 grader och kunde leverera en kraft på 11 Kg. Position 4 och 5 (se Figur 1), kontrollerade pitch (nigning) av klon samt öppning och stängning av klon. Under dessa rörelser utsattes inte servona för någon större belastning, och därför valdes Az-Deliverys SG90 servon. Dessa servon hade ett vinkelområde på 180 grader och kunde leverera en kraft på 1,8 Kg.



Figur 1: CAD-modell av robotarm där servo positionerna är numrerade

4.3 Val av styrenhet

För att uppnå målen med projektet krävdes någon typ av programmerbar styrenhet där flera krav ställdes. Eftersom robotarmarna skulle arbeta på separata uppgifter samtidigt, och att hela systemet krävde väldigt många input/output pinnar, bestämdes det att använda två styrenheter. Varje styrenhet skulle då kontrollera en robotarm, ett rullband, en färgsensor, en avståndssensor, en servodrivare samt en motordrivare. Både servodrivaren och avståndssensorn krävde I^2C protokoll för kommunikationen, vilket medförde att styrenheten var tvungen att stödja I^2C . Med dessa krav i åtanke kombinerat med tidigare erfarenhet valdes två Arduino Uno som styrenheter för projektet.

4.4 Rullband

En annan viktig komponent var rullbanden. Då projektet hade som krav att systemet inte skulle vara större än 60 x 60 cm sattes 60 cm som maximal längd för rullbanden. Ett annat krav var även att rullbanden skulle gå att reglera i bandhastighet. Efter en informationssökning landade valet på ett rullband från märket Toplionace. Detta rullband var 10x50 cm vilket var inom den förutbestämda max längden, och rullbandet kom även utrustat med en hastighetsregulator för att ändra hastighet på rullbandet. För att hastigheten på rullbandet skulle kunna styras av en arduino konstaterades snabbt att det inte fanns någon användning av hastighetsregulatorn som skickades med rullbanden. Eftersom rullbandet drevs av en 12V elmotor blev lösningen att använda sig av motordrivare för att kontrollera hastigheten på elmotorn. Kravet på motordrivarna var att de måste kunna leverera 12V samt minst 2A. Därför valdes Adafruit DRV8871 som är en motordrivare som är kapabel till att styra en motorspänning mellan 6-45V samt en ström på 3,6A.

4.5 Servodrivare

I varje delsystem krävdes minst sju PWM utgångar. Varje robotarm innefattar 6 servon, där alla servon kräver en dedikerad PWM signal, samt att motordrivaren också kräver en PWM utgång. Eftersom Arduino Uno endast har sex PWM utgångar krävdes en lösning för att möta kraven. Lösningen var att implementera en PCA9685 servodrivare för varje delsystem där alla servon styrs ifrån. På servodrivaren användes jordningspunkterna samt styrsignalerna till servona, och strömförsörjningen till servona kopplades direkt från buck-konverterarna. Servodrivaren hade möjligheten att generera egna PWM-signaler oberoende från Arduinon vilket gjorde den icke-blockerande. Detta medförde att Arduinon fortfarande kunde exekvera kod samtidigt som servodrivaren skickade ut PWM signaler till servona. Servodrivaren hjälpte även med andra saker, som att bidra med jämnare

rörelser och minska jitter i servona, samt att göra programmeringen mindre komplex.

4.6 Strömförsörjning

För att systemet skulle fungera krävdes en pålitlig strömförsörjning. För att veta kraven på denna krävdes beräkningar på strömförbrukningen hos alla komponenter i systemet. Strömförbrukningen för komponenterna hittades i respektive datablad för varje komponent. Varje servo krävde 6 V, och för alla 12 servon för båda robotarmarna krävdes en total ström på ca 23,4 A. Strömförbrukningen för servona räknades ut genom att ta den maximala strömmen som servona drar när rotorn är låst men fortfarande försöker driva den. Detta händer till exempel när servona håller den tyngsta belastningen den klarar i ett fast läge. Efter detta räknades strömförbrukningen ut för sensorerna. De sensorer som skulle drivas var två färgsensorer samt två avståndssensorer. Båda dessa typer krävde 5 V logik, samt att den totala strömförbrukningen för alla 4 sensorer låg på 0,84 A. De två Arduino unos som användes som kontrollenhet krävde ca 0,8 A tillsammans, och de två servodrivarna krävde 0,8 A tillsammans.

Databladet för rullbanden specificerade inte hur mycket ström rullbanden drar vid maximal hastighet. Det som specificerades var att motorerna är 12 V DC (Direct Current) motorer, och har en maximal motorhastighet på 116 RPM. Utifrån detta gjordes en uppskattning på att de två banden inte kommer dra mer än 2 A tillsammans vid maximal drift. Denna gjordes utifrån kunskap om liknande motorer samt att bandet inte kommer utsättas för någon större belastning vid drift.

När allt sammanställdes landade den totala strömförbrukningen på 27,84 A, vilket var den ström som hela systemet potentiellt skulle kunna konsumera. Nästa steg var att välja vilken spänning som strömkällan skulle leverera. Eftersom rullbanden arbetade med 12 V, och resterande komponenter krävde mindre spänning valdes 12 V som den spänning strömkällan skulle leverera. Eftersom hela systemet byggdes upp som två delsystem, krävde varje delsystem maximalt 13,92 A vid drift. Då varje delsystem var utrustat med en Buck-omvandlare som sänkte spänningen, men ökade strömmen krävdes inte fullt lika hög ström in från strömkällan. Kravet på strömkällan om man tar hänsyn till användningen av buck-omvandlare räknades ut enligt ekvation 3 till 5.

Effekten ut från Buck-omvandlaren vid 13,92 A och 6 V:

$$P_{Ut} = U_{Ut} \times I_{Ut} = 6 \text{ V} \times 13,92 \text{ A} = 83,52 \text{ W} \quad (3)$$

Effekt in vid 95% verkningsgrad hos Buck-omvandlaren

$$P_{in} = \frac{83,52 \text{ W}}{0,95} = 87,92 \text{ W} \quad (4)$$

Ström som krävs in till Buck-omvandlaren:

$$I_{in} = \frac{87,92 \text{ W}}{12 \text{ V}} = 7,33 \text{ A} \quad (5)$$

Detta betyder att varje Buck-omvandlare kräver en ström in på minst 7,33 A för att kunna leverera 13,92 A ut till varje delsystem. Om båda delsystemen krävde 7,33 A var krävdes

en total ström från strömkällan på minst 14,66 A, vilket avrundat uppåt blir 15 A.

När både ström och spänning var bestämt kunde en strömkälla letas upp och beställas. Valet landade på en strömkälla som kopplades till vägguttaget, för att sedan omvandla AC strömmen till 12 V DC, 15 A.

4.7 Buck-omvandlare

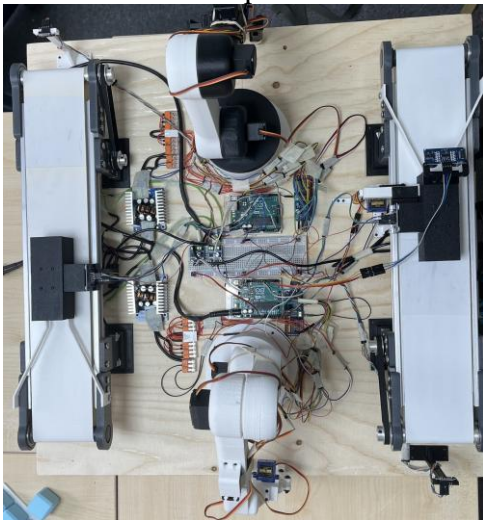
Då strömkällan levererade 12 V, men servona krävde en spänning på 6 V krävdes en buck-omvandlare för att sänka spänningen. Denna buck-omvandlare var tvungen att kunna sänka spänningen ner till 6V, och samtidigt kunna leverera tillräckligt med ström för att försörja robotarmens strömförbrukning. Då hela systemet är uppbyggt av två delsystem, bestämdes det för att köpa in två buck-omvandlare, en till varje delsystem. Den buck-omvandlare som valdes kunde justeras, och kunde leverera en spänning på 2-36 V DC, samt maximalt 20 A. Buck-omvandlaren hade även möjlighet att leverera 300 W vilket var fullt tillräckligt för systemet.

5. Konstruktion och design

Kapitlet konstruktion och design beskriver hur systemets mekaniska delar designades och skapades, hur hela systemet fungerar tillsammans och hur elektroniken i systemet är kopplat.

5.1 Systemlayout

Systemet utformades som två likadana delsystem som kunde arbeta och lösa uppgifter oberoende av varandra. Varje delsystem bestod av en robotarm, ett rullband, sensorer samt tillhörande elektronik som monterades på en basplatta. Robotarmarna monterades i horisontell linje med varandra, och i mitten av basplattan. Den vita roboten i den nedre delen i figur 2 kallas robot 1 och den svarta roboten i den övre delen kallas robot 2. De två rullbanden monterades parallellt på var sin sida av robotarmarna, vilket visas i figur 2. Detta skapade ett system där varje robotarm kunde hämta en låda från slutet av det första rullbandet, och lämna av den i början på nästa. I mitten av systemet skapades en yta där alla elektronikkomponenter monterades för att hålla systemet organiserat.



Figur 2: Bild av komplett system

Flödet började med att användaren placerade en låda på det första rullbandet. Lådan styrdes sedan in till mitten av rullbandet med hjälp av mekaniska riktare och dess färg lästes av en färgsensor. Efter färgen läses av stannar lådan inom den angivna stoppzonen. Stoppzonen är den del av bandet där lådan kan bli upphämtad på. Vart lådan stannar inom zonen hanteras av en randomiserings algoritm där positionen slumpas fram. I slutet av rullbandet är en avståndssensor monterad, som läser av avståndet lådan stannade på. Denna information skickas till roboten vilket triggar en upphämtning av lådan. Beroende på färg flyttas antingen lådan till en avlämningsplats, eller över till nästa rullband och in i delsystem två.

Basplattans storlek begränsades i kravspecifikationen till 60 x 60 cm, vilket påverkade både layouten av systemet och komponentval. Placeringen av komponenterna optimerades dels för att minimera kablage och oreda bland elektroniken, men även för att robotarmarna inte skulle kollidera.

Robotarna har var sitt koordinatsystem som används för att referera till positioner och riktningar. Koordinatsystemet för robot 1 har origo i mitten av roboten vid höjden av axelleden. Positiv x riktning är ut från plattan i den riktningen som roboten står mot i figur 2. Positiv y riktning är mot det högra bandet i figur 2. Positiv z riktning är uppåt från plattan. Robot 2 har ett identiskt koordinatsystem, men med positiv x och y riktning spegelvända från robot 1.

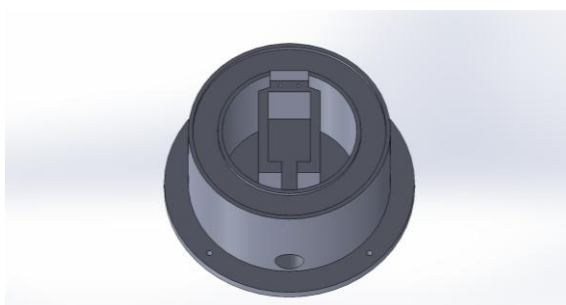
5.2 3D-modellering och komponentdesign

För att skapa systemets design gjordes initialt skisser och koncept. Skisserna gjordes för hand för att sedan ha något att utgå ifrån för CAD momentet. Metoden som valdes för att skapa robotdesignen var en iterativ process, där flera versioner skapades med nya förbättringar varje gång. Roboten är uppbyggd i olika segment som kopplas samman i leder. För att säkerställa funktionalitet, passform samt att kunna producera delarna skapades samtliga mekaniska komponenter i CAD-programmet SolidWorks.

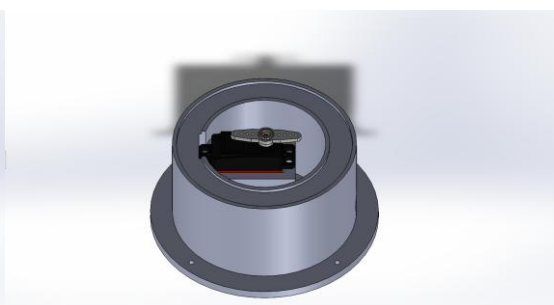
5.2.1 Bottenkopp

Bottenkoppen är den del av robotarmen som fäster i basplattan och visas i Figur 3. Bottenkoppen designades för att kunna fästas in i basplattan med fyra skruvar placerade runt om roboten. Inuti bottenkoppen konstruerades en servohållare, där servot monteras med två skruvar. Denna servohållare håller servot på plats och ger samtidigt gediget stöd i sidled för stabilitet. Då varje servo har tre kablar som sticker ut på kortsidan gjordes en utskärning i servohållaren för dessa. Kablarna fördes sedan ut genom hålet på baksidan av koppen för att sedan kopplas in till resterande elektronik.

Då en toppkopp skulle monteras på bottenkoppen som skulle snurras av servot i bottenkoppen krävdes en lösning för att minimera friktionen. Denna friktion uppstod emellan botten och toppkopp då robotarmen var utsträckt och skapar ett moment mot basen. Det bestämdes då att använda ett axialkullager, och det designades in ett spår på toppen av bottenkoppen så att lagret inte kunde glida i sidled. Servot monterades i koppen och servohornet skruvades fast på servot enligt figur 4.



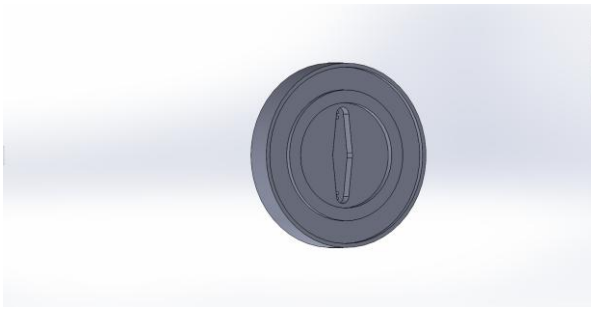
Figur 3: Bottenkopp



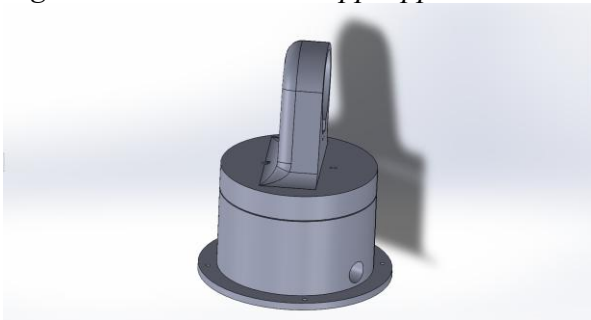
Figur 4: Bottenkopp med monterat servo

5.2.2 Toppkopp

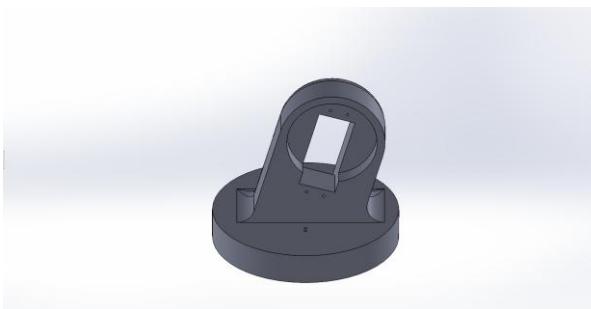
Toppkoppen var den del som monterades på bottenkoppen, och även där första armen fäster in. Toppkoppen designades med ett likadant spår för axialkullagret som bottenkoppen hade. Det gjordes även en utskärning så att servohornet kunde sänkas in i toppkoppen vilket visas i figur 5. Servohornet och därmed hela toppkoppen klämdes sedan ihop med fyra skruvar som monterades från ovansidan och på var sin sida om servofästningen, vilket visas i figur 6.



Figur 5: Undersidan av toppkoppen

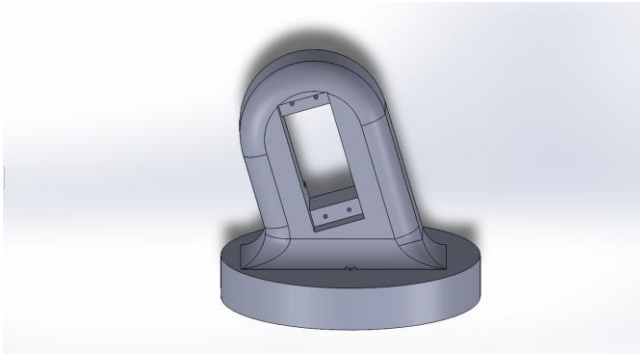


Figur 6: Hopmonterad botten och toppkopp



Figur 7: Toppkoppens översida samt cirkulärled

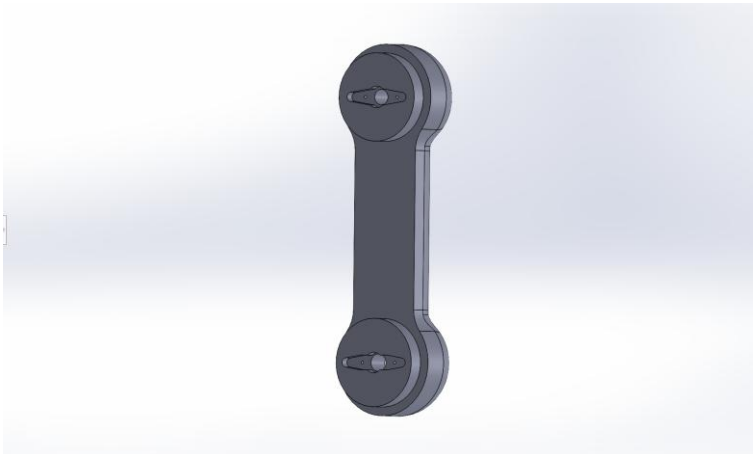
Därefter designades en plats så att servot kunde fästas in i toppkoppen, vilket visas i figur 8. Servot monterades in bakifrån med fyra skruvar för att sitta stabilt i konstruktionen. På andra sidan skapades en cirkulärled där den första armen skulle monteras i, vilket visas i figur 7. Eftersom alla delarna skulle 3D-skrivas var toleranserna en viktig del att tänka på vid designen av delar så som cirkulärleden. Det var viktigt att göra cirkulärledens hål i toppkoppen något större än armens matchande del för att få en så bra passform som möjligt. En för tajt led hade lett till att armen inte skulle kunna röra sig alls, och en för lös led hade resulterat i instabilitet och minskad precision i systemet. Valet av tolerans landade på ett mellanrum runtom på 0,5 mm, alltså gjordes toppkoppens hål 1 mm större i diameter än armens infästnings diameter.



Figur 8: Servoinfästning i toppkoppen

5.2.3 Arm 1

Nästa steg var att designa den första armen för roboten som skulle fästa in i toppkoppens cirkulärled. Cirkulärleden designades med en utstickande cylinder, vilket visas i Figur 9. Då denna arm skulle fästa i nästa arm med samma typ av cirkulärled speglades designen på båda sidor. På båda dessa skapades även utskärningar för servohornen som skruvades in i armen. Mellan lederna skapades en kabelkanal där servokablarna kunde ledas in för att dölja och hantera allt kablage som skapades, vilket visas i figur 10. Servohornet och därmed armen monterades in på servot i toppkoppen för att sedan skruvas in med en infästningsskruv som fäster hornet och armen till servot enligt figur 11.



Figur 9: Arm 1 sedd underifrån



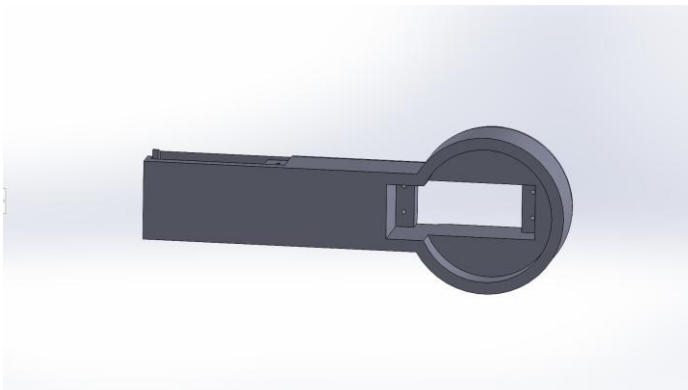
Figur 10: Arm 1 sedd ovanifrån med synlig kabelkanal



Figur 11: Arm 1 monterad på toppkopp

5.2.4 Arm 2

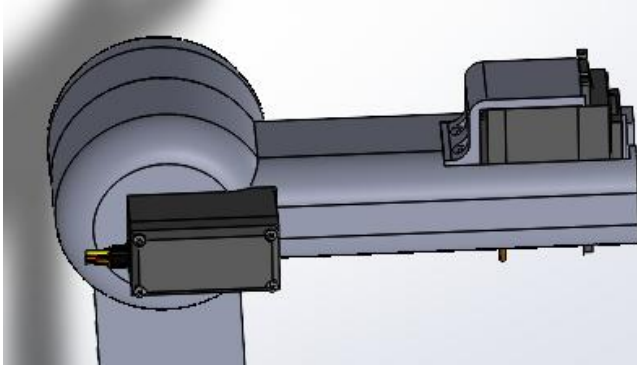
Som nämndes tidigare användes även här en cirkulärled för att koppla ihop arm 1 och arm 2. Designen innefattade även en utskärning där servot sänktes in och skruvades fast med fyra skruvar i varje hörn enligt figur 12. Därefter krävdes implementation i designen för ett servo som skulle ta hand om klons roll. Eftersom servots axel var tvungen att sticka ut i änden på Arm 2 bestämdes det att försänka in servot i armen på högkant. Problemet som då uppstod var att eftersom servot skulle fästas in på högkant, kunde inte servovingarnas infästningshål användas för att montera servot med skruvar som tidigare. Detta löstes genom att göra spår i botten av armen enligt figur 13. Det kvadratiske hålet längst in var till för att leda ut servokablarna. Det främre rektangulära hålet designades så att servoringen längst ner press-passades in i hålet. Därefter monterades en bricka för att hålla servot på plats så att servot inte kunde lyftas ut rakt upp, vilket visas i figur 14. Arm 2 monterades sedan på arm 1 enligt figur 15.



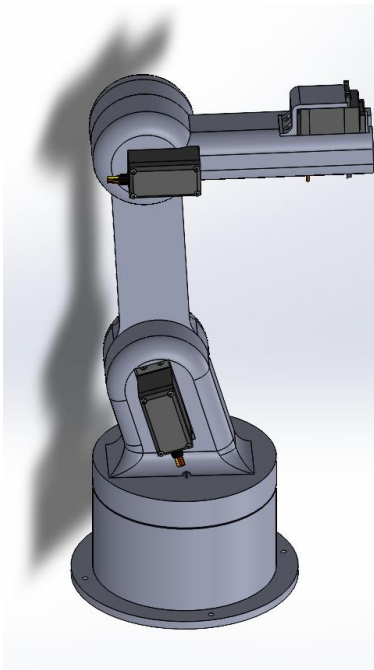
Figur 12: Cirkulärled och utskärning för servo i Arm 2



Figur 13: Utskärning i Arm 2 för servo som styr roll



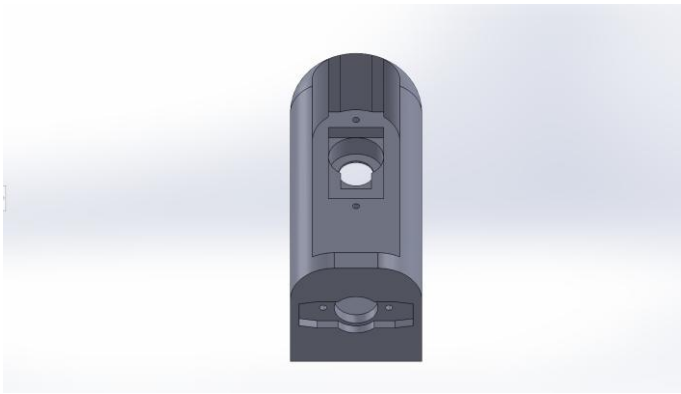
Figur 14: Arm 2 med monterad bricka för servo



Figur 15: Arm 2 monterad på Arm 1 samt resterande konstruktion

5.2.5 Arm 3

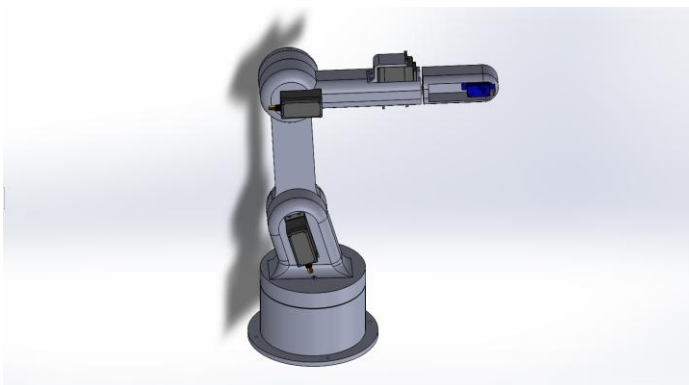
Arm 3 utformades för att kunna fästa in på arm 2 för roll-funktionen och på andra sidan kunna stödja infästningen av robotklon. För sammanfästningen av Arm 2 och 3 gjordes en utskärning för servohornet i botten av arm 3. Då arm 3 inte är tillräckligt bred gjordes en smalare variant av utskärningen, vilket visas i figur 16. Detta medförde att servohornet var tvunget att kapas i längd vid montering. Då griplons pitch skulle kontrolleras av ett mindre SG90 servo, gjordes en försänkning för detta servo högst upp enligt figur 17. Detta gjorde att SG90-servots axel kunde sticka ut på andra sidan och skapa en infästningspunkt för griplon. Hela arm 3 monterades sedan på resterande konstruktion enligt figur 18.



Figur 16: Servohornsutskärning på Arm 3



Figur 17: Servoförsänkning för SG90

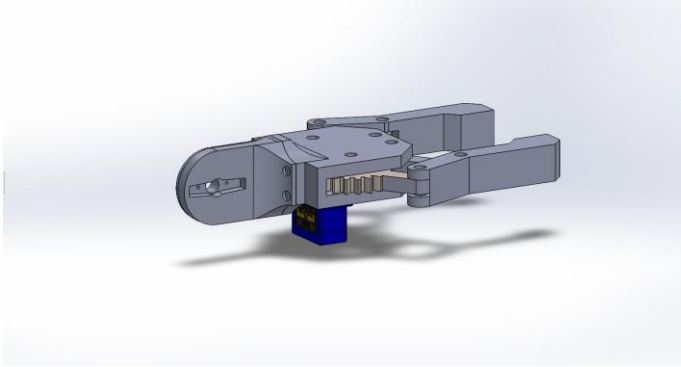


Figur 18: Arm 3 monterad på restarande konstruktion

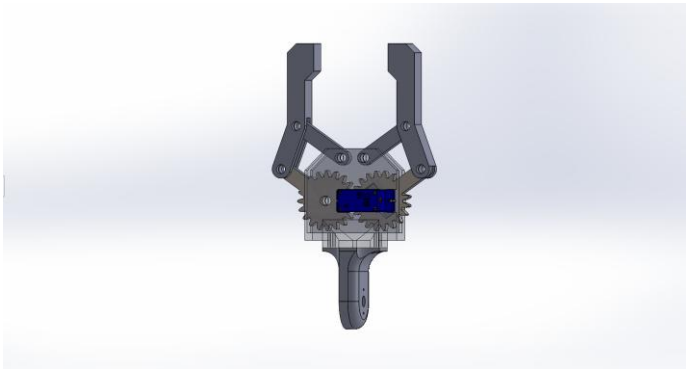
5.2.6 Gripklo

Gripklons första del designades för att kunna fästa in på arm 3s utstickande servoaxel. Detta gjordes på ett liknande sätt som de andra delarna, där en utskärning av servohornet gjordes, enligt figur 19. Själva gripklo-mekanismen består av två kugghjul, två griparmar samt två länkarmar. Det ena kugghjulet är i sin tur kopplat till ett SG90 servo som driver klons öppning och stängning, enligt figur 20. Då formen av lådan var kvadratisk, önskades en stängning där griparmarna alltid var parallella med lådans kanter. För att nå denna önskade rörelse krävdes det att länkarmarna, benämnda X i figur 21, hade samma längd som längden från mitten av kugghjulet ut till infästningen i botten av griparmen. Samt att längden mellan mitten av kugghjulet till länkarmens ena infästning, benämnd Y

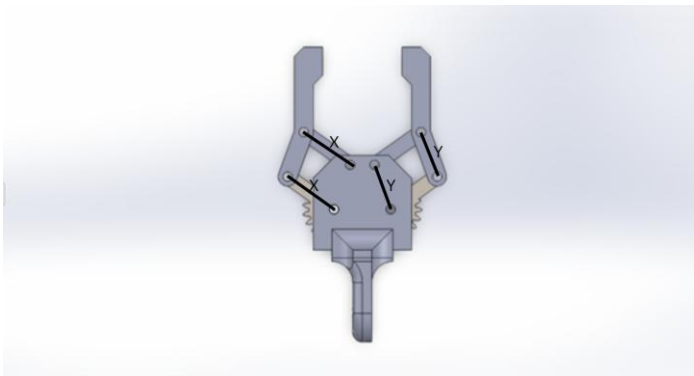
i figur 21, var lika lång som längden mellan de två infästningarna på gripklon. Hela klon monterades sedan ihop med M4-bultar med motsvarande muttrar på undersidan. Därefter monterades hela klon på resterande konstruktion enligt figur 22.



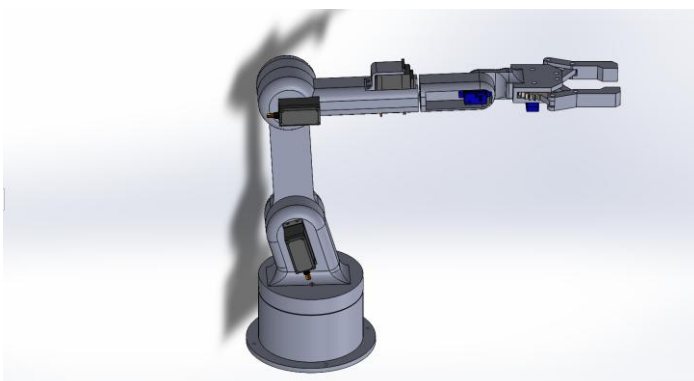
Figur 19: Gripklo med utskärning för servohorn



Figur 20: Intern mekanism av gripklo



Figur 21: Längder på gripklons delar



Figur 22: Gripklo monterad på resterande konstruktion

5.3 Elkonstruktion

Då hela systemet är uppbyggt utav två delsystem designades även elkonstruktionen med detta i åtanke. Den huvudsakliga strömmen är på 12V, 15A och kommer in till systemet via en strömadapter, som nämns mer i detalj i kapitel 4.6. Denna huvudström delas sedan upp till två Buck-konverterare för att föras ut till varje delsystem. Då huvudströmmen är 12V dras den även direkt till rullbanden, via motordrivare. Huvudströmmen dras även ut till båda Arduino Uno enheterna för att ge den ström. Varje delsystem är uppbyggt på samma sätt och börjar med en Buck-konverterare. Buck-konverterarna sänker spänningen till 6 V för att fördelas ut på robotarmarnas alla servon. Då färgsensorn och avståndssensorn i varje delsystem är ganska strömsnåla och dessutom kräver 5 V tas detta från arduinon. Arduinons 5 V delas upp på en kopplingsbräda som sitter monterad i mitten av systemet.

De servon som sitter på robotarmen styrs via en servodrivare. På servodrivaren utnyttjas endast jord och styrsignalen, strömmen dras direkt till servot från Buck-konverteraren, vilket beskrivs tydligare i kapitel 4.5. Delsystem 1 innefattar även ett servo till fördelaren samt en ultraljudsensor, vilka inte finns på delsystem 2. Servot till fördelaren styrs via delsystemets servodrivare, på samma sätt som servona till robotarmen. Ultraljudssensorn är av 5 V logik och drivs därför av Arduinons 5 V pinne via kopplingsbrädan. För att säkerställa en korrekt referenspunkt kopplades alla jord-kablar ihop. Ett elschema på hela kopplingen kan ses i bilaga 3

6. Programmering och styrning

För styrningen av systemets alla delar: robotarmar, gripklo, rullband och sensorer användes en Arduino Uno. Programmeringen gjordes med hjälp av Arduino IDE. Servomotorerna på robotarmarna styrs med hjälp av PWM-signaler och för att möjliggöra koordinatbaserad rörelse användes invers kinematik ekvationer för att lösa vinklarna som lederna behöver ha för en given position. Robotarmens olika parametrar som längder och avstånd från centrum har mätts ut för att användas i kinematikekvationerna. De längder som används är $L_1=150$ mm och $L_2=192,5$ mm, där L_1 är längden på arm 1 och L_2 är längden på arm 2.

6.1 PWM och servon

Eftersom robotarnas servomotorer styrs med hjälp av servodrivare används biblioteket `Adafruit_PWMServoDriver.h`. Med hjälp av detta bibliotek skapades ett servo objekt som kan användas för att ställa olika PWM nivåer på varje servo. Varje servo har en intern PWM räckvidd som hittades med hjälp av att öka och sänka signalen i steg tills den slutade röra sig åt varsitt håll. När räckvidden fastslagits kunde den omvandlas till ett vinkelintervall för att användas i kinematikberäkningar.

De servon som användes i robotarmarna har inte absoluta positionsgivare vilket gjorde att ett system för att hålla koll på nuvarande position behövde implementeras. När programmet startas körs en hemsekvens som kör alla servon till kända positioner. I slutet av denna hemsekvens lagras alla servons nuvarande pulser i en global array. I alla funktioner som förflyttar servon uppdateras denna array i slutet så att den nuvarande positionen alltid är rätt och läsbar av alla funktioner.

När en PWM signal skickas till ett servo roterar axeln direkt för att nå den skickade signalens motsvarande position. När servot förflyttar eller roterar delar som har betydande vikt och längd kan en sådan direkt rörelse bli ryckig och utgöra en säkerhetsrisk. Därför utvecklades en funktion `moveServoSmooth` för att röra servon mellan två vinklar på ett mjukt sätt. Den tar målpositionen och nuvarande positionen och rör sig med små steg och korta fördröjningar mellan för att skapa den mjuka rörelsen. En ytterligare funktion `moveAllServos` skapades som kan göra denna mjuka rörelse med alla servon simultant. Dessa två rörelsefunktioner kan ha olika hastighet genom att ändra fördröjningen vid varje loop och hur stora steg dem ska ta per iteration av loopen.

6.2 Invers Kinematik

Huvudfunktionen för koordinatbaserad förflyttning av robotarna är `moveXYZ`. Den tar fem parametrar: x, y, z, roll och pitch. Med hjälp av invers kinematik används de tre första parametrarna för att bestämma de tre första ledvinklarna.

Basrotationens vinkel θ_1 , definieras som vinkeln mellan x-axeln och målets position i xy-planet. Den beräknas enligt ekvation 6:

$$\theta_1 = -\arctan2(y, x) \frac{180}{\pi} \quad (6)$$

Funktionen `arctan2` används för att den kan korrekt hantera punkter i alla fyra kvadranter och ger därför en vinkel med rätt tecken beroende på värdena x och y. Vinkeln görs om från radianer till grader och inverteras sedan för att korrekt spegla den fysiska monteringen av servomotorn.

Avståndet från koordinatsystemets ursprung till målet i xy-planet (r) ges av ekvation 7. Det totala avståndet (D) från ursprunget till målet inklusive höjden som ges av z beräknas med ekvation 8.

$$r = \sqrt{x^2 + y^2} \quad (7)$$

$$D = \sqrt{r^2 + z^2} \quad (8)$$

Axelledens vinkel θ_2 , definieras som vinkeln mellan axelleden och xy-planet. θ_2 anger hur mycket armens första länk lyfts upp eller ner i förhållande till basplanet när den sträcker sig mot målet i höjden. För att beräkna den krävs två hjälpvinklar α och β . α definieras som vinkeln mellan linjen i xy-planet som går från basen till målet och linjen som går från basen till målet i 3D och beräknas med ekvation 9. β definieras som vinkeln mellan länkarmen L_1 och linjen till målpunkten D och beräknas med cosinuslagen i ekvation 10.

$$\alpha = \arctan 2(z, r) \frac{180}{\pi} \quad (9)$$

$$\beta = \arccos\left(\frac{L_1^2 + D^2 - L_2^2}{2L_1D}\right) \frac{180}{\pi} \quad (10)$$

α beskriver alltså höjdvinkeln från basen till målet och β är böjvinkeln som gör att L_1 pekar mot målpunkten och tar hänsyn till att armbågsleden L_2 också ska nå dit. θ_2 beräknas i ekvation 11 genom att lägga ihop α och β . Sedan justeras den enligt robotens koordinatsystem.

$$\theta_2 = 90 - (\alpha + \beta) \quad (11)$$

Armbågsledens vinkel θ_3 definieras som vinkeln mellan robotens två länksegment L_1 och L_2 . Denna vinkel talar om hur mycket armbågsleden är böjd eller sträckt när roboten sträcker sig mot målpunkten. För att beräkna vinkeln tillämpas cosinuslagen på den triangel som bildas av L_1 , L_2 och D . Den beräknas enligt ekvation 12.

$$\theta_3 = 180 - \arccos\left(\frac{L_1^2 + L_2^2 - D^2}{2L_1L_2}\right) \cdot \frac{180}{\pi} \quad (12)$$

Vinkeln som fås korrigeras med 180 grader för att överensstämna med den fysiska monteringen av servomotorn som styr denna led.

Dessa tre vinklar (θ_1 , θ_2 , θ_3) beskriver den konfiguration som krävs för att nå målpunkten (x , y , z). För att vinklarna ska kunna användas av servona behöver de först begränsas till det tillåtna intervallet och sedan översättas till den motsvarande pulsbredden, detta görs med funktionen `mapFloat`. För att avsluta funktionen skickas alla pulsbredder till funktionen `moveAllServos` för att uppnå alla beräknade vinklar simultant.

6.3 Programmering av gripklo

Ändeffektorn som monterats på robotarmens ände är en gripklo. För att roboten ska kunna utföra sitt arbete krävs det att ändeffektorns orientering och funktion manipuleras. Detta

görs genom att justera robotarmens pitch och roll för att nå olika vinklar och orienteringar. Dessutom styrs griplons öppning och stängning noggrant för att möjliggöra plockning av lådor.

Pitch och roll kan ansättas till roboten genom två sätt. Det ena är att direkt skicka vinkeln som önskas till funktionen `moveXYZ` som tidigare förklarades. Med detta sätt görs rörelserna simultant som resten av förflyttningarna i `moveXYZ`. När robotarmen kör in från sidan för att plocka upp en låda är detta beteende inte önskvärt och därför implementerades funktioner för enskild rörelse av pitch och roll, dessa benämns Pitch och Roll. Till dessa funktioner skickas önskad vinkel som sedan görs om till pulsbredd och skickas till `moveServoSmooth` för enskild mjuk rörelse.

Griplon har två funktioner för styrning, `openClaw` och `closeClaw`. Efter att servomotorn för griplon hade monterats testades olika pulsbredder för att hitta önskvärda lägen. När dessa hade hittats implementerades de i funktionerna. Eftersom griplon inte har lika mycket potential till katastrofala rörelser som resten av roboten skickas pulsbredden direkt till servot utan någon interpolering mellan för mjuk rörelse. I slutet av funktionen `openClaw` skickas pulsbredden noll för att effektivt stänga av det servot, detta görs för att undvika att servot blir för varmt.

6.4 Sensorprogrammering

För att robotarmen ska kunna utföra sina uppgifter behöver den både kunna detektera färg och läsa avstånd. För att uppnå detta har en färgsensor och avståndssensor implementerats.

Färgsensorn programmerades genom att första kalibrera den på objekt av olika färger och i olika ljusförhållanden. När värdena samlats in kunde gränser för färgerna röd, grön och blå skapas. När sensorn läser in RGB-värden jämförs dessa med gränserna som har satts för att avgöra vad för färg som finns framför sensorn. När en färg har upptäckts tilldelas ett specifikt värde till en variabel `colorVal` som används på andra ställen i koden för att veta vilken färg som senast upptäcktes. Om ingen gräns uppnås tilldelas värdet noll till `colorVal`, detta tolkas som att ingenting är framför sensorn.

Avståndssensorn kommunicerar med mikrokontrollern via I^2C och har programmerats med kodbiblioteken `Wire.h` och `VL53L0X.h`. Sensorn initialiseras först och om den inte lyckas, avbryts exekvering för att minska risker med systemet. När sensorns mätfunktion kallas så görs 5 separata mätningar i följd som sedan används för att beräkna ett medelvärde. Detta görs för att minska brus och förbättra noggrannheten på mätningarna. Även med små skillnader i ljus och omgivning kan sensorn leverera tillförlitliga mätningar med denna metod.

6.5 Rullbandstyrning

Rullbanden styrs med motordrivare som är kopplade till mikrokontrollern. Riktningen och hastigheten på banden styrs med PWM-signaler. Tre funktioner till rullbanden har skapats för dess styrning: `bandForward`, `bandBackward` och `bandStop`. För att systemet ska fungera med olika bandhastigheter tar `bandForward` och `bandBackward` önskad hastighet som en parameter. Dessa styrfunktioner fungerar genom att skicka olika signaler på de två pins som är kopplade till motordrivaren, pin 9 (IN1) och 10 (IN2).

Rullbanden initialiseras med att öka PWM-frekvensen till ungefär 31kHz för att ge tystare och jämnare motorstyrning. Funktionen `bandForward` aktiverar bandets rörelse framåt genom att skicka den medskickade hastighetsvariabeln till IN2 och skickar noll till IN1.

Funktionen `bandBackward` fungerar på samma sätt men med motsatta pins, hastighetsvariabeln skickas till IN1 och noll skickas till IN2. Funktionen `bandStop` skickar noll till både IN1 och IN2 vilket stoppar bandets rörelse.

En kvadratisk ekvation skapades för att kunna konvertera PWM-signalerna som skickades för att styra hastigheten till verklig hastighet mätt i mm/s. För att få denna funktion mättes tiden det tog för en låda att röra sig ett utmätt avstånd vid olika hastigheter av bandet. Den mätta tiden delades med sträckan och plottades sedan mot motsvarande PWM-signaler. Sambandet var inte linjärt och därför användes en kvadratisk kurva för att passa in mätvärdena, enligt ekvation 13.

$$Speed = -0.004435 * PWM^2 + 3.2687 * PWM - 394.93; \quad (13)$$

6.6 Huvudprogram

I huvudprogrammet utförs robotarnas huvudsakliga arbete i formen av en loop. Loopen börjar med att starta rullbanden vid den hastigheten som är bestämd i en global variabel. En variabel `randDelay` skapas som innehåller en slumpmässigt vald tid. Därefter körs en loop som kör färgsensorns huvudkod. Den fortsätter köra så länge `colorVal` är noll, alltså inget är framför färgsensorn. När färgsensorn har upptäckt en färg börjar en fördröjning som bestäms av `randDelay`, när fördröjningen är klar stannar bandet. Denna slumpmässiga fördröjning innan bandet stoppas görs för att roboten ska plocka lådor från hela upphämtningszonen och inte från samma position varje gång. När bandet har stannat kör distanssensorn sin kod. Resultatet från distanssensorn transformeras för att fungera med robotens egna koordinatsystem.

Nästa steg för huvudprogrammet är att roboten ska plocka upp lådan som kommit fram på bandet och hantera den utifrån färgen den har. Först öppnas klon för att säkerställa att roboten är redo att utföra en plocksekvens. Roboten körs sedan till koordinaten som angavs av distanssensorn. Eftersom bandet har konstant höjd och lådorna alltid befinner sig i mitten av bandet har plockningen samma y och z koordinat varje gång. När roboten har kommit fram pitchar den för att nå rätt vinkel och sedan stängs klon.

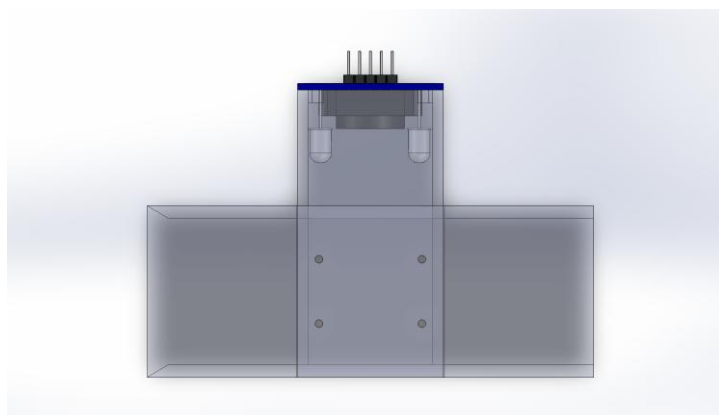
Vid detta stadiet sker olika saker beroende på värdet av `colorVal` från färgsensorn. För robot 1, om det är en blå eller grön låda förflyttas robotarmen till avlämningsplatsen på det andra bandet. Om det är en röd låda så förflyttas robotarmen till avlämningsplatsen för röda lådor framför roboten. För att uppnå flera avlämningar av röda lådor finns det en räknare som uppdaterar koordinaterna för avlämning efter varje röd låda. För robot 2, om det är en grön eller röd låda så förflyttas robotarmen till avlämningsplatsen på det andra bandet. Röda lådor ska inte vara på detta band men ifall färgigenkänningen misslyckats på föregående robot behöver detta fel hanteras genom att ge lådan ett till varv att korrekt identifieras. Om det är en blå låda så gör roboten på samma sätt som robot 1 för röda lådor men den blå avlämningsplatsen befinner sig framför robot 2. Efter hanteringen av lådan startar loopen om.

7. Iteration och förbättringar

Under arbetets gång har ett flertal förbättringar, funktioner och tillägg identifierats som har implementerats i samband med att problem uppstått eller förbättringsmöjligheter setts. Denna iterativa process har varit viktig i arbetet för att få en så välutvecklad och fungerande slutprodukt som möjligt.

7.1 Färgsensor förbättring

Ett problem som upptäcktes med färgsensorn är att den är väldigt känslig för ljusförhållandena i rummet som systemet befinner sig i. När solljus skiner rätt på sensorn slutar färggränserna fungera som vanligt eftersom de mätta RGB-värdena blir felaktiga. För att undvika detta problem utvecklades ett solskydd till färgsensorn. Solkyddet är konstruerat som en T-formad tunnel där lådor kan passera igenom den horisontella delen, och sensorn ansluts till tunneln via den vertikala delen som visas i figur 23. Med detta tillägg skapas en kontrollerad miljö för färgsensorn samtidigt som lådorna kan passera opåverkat.

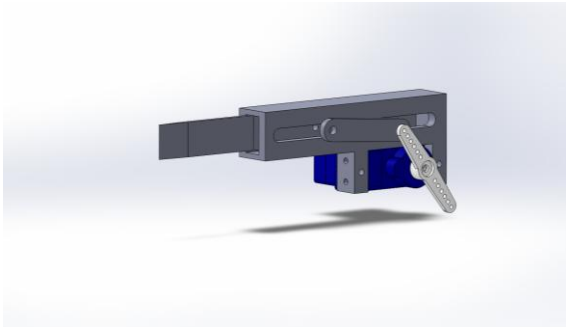


Figur 23: Solkydd till färgsensor

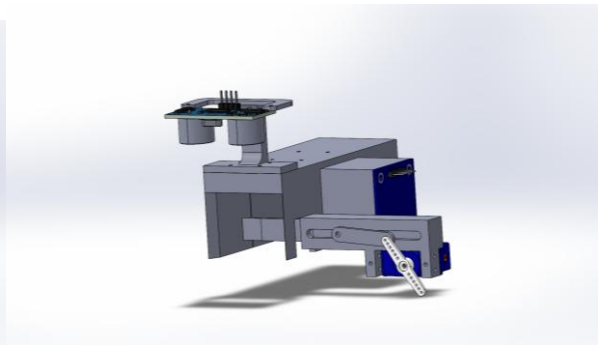
7.2 Fördelare

När flera lådor kommer i snabb följd kan systemet bli överbelastat. Färgsensorn kan möjligen hinna läsa av varje låda som kommer men roboten är inte tillräckligt snabb och lådorna kan vara i vägen för griplons funktion. För att undvika detta behövdes ett tillägg som gör att endast en låda i taget kan ta sig till upphämtningszonen. Lösningen som togs fram för detta var en fördelare som blockerar lådor. Komponenten är utformad som en kil som körs i en linjär rörelse i ett spår av ett servo, se figur 24. Uppgiften av fördelaren är att endast släppa förbi en låda, medan alla andra lådor temporärt blockeras.

För att fördelaren ska fungera som önskat behöver den kunna avgöra när den ska öppna respektive stänga för passering. Denna information ges utav en ultraljudssensor som monteras på toppen av solskyddet enligt figur 25. I denna figur visas även var fördelaren monterats relativt till färgsensorn. När en låda har blivit upptäckt av ultraljudssensorn öppnas fördelaren tillräckligt länge för att släppa förbi lådan varpå den genast stängs för att blockera eventuella lådor på kö. Eftersom lådor kan ligga precis intill varandra på rullbandet, har fördelaren en spetsig ände för att kunna lättare komma mellan två lådor när den ska stänga.



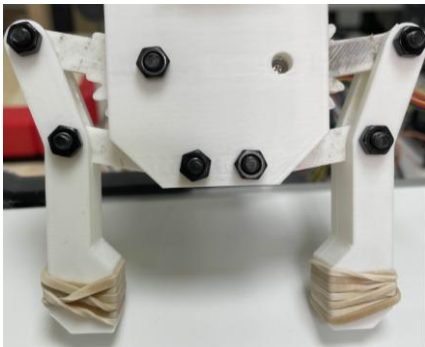
Figur 24: Fördelare



Figur 25: Ultraljudssensor

7.3 Grepp problem

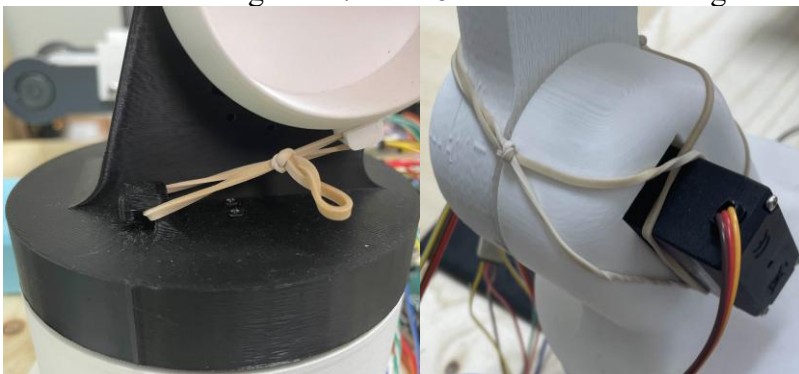
Det upptäcktes tidigt i utvecklingen av robotarmarna att gripklon hade problem med grepp. Plastytan var slät och kunde ibland tappa lådor. En simpel lösning för att ge gripklon extra grepp var att sätta gummiband runt änden som greppar. Gummibandens elastiska material gjorde att gripklon kunde hålla lådorna på plats och inte behöva klämma lika mycket. För att fästa gummibandet och få så jämn yta av gummiband som möjligt fästes de enligt figur 26.



Figur 26: Gummiband på gripklo

7.4 Arm 1 gummiband

Under utvecklingen upptäcktes det en svaghet i den första länken av robotarmen. Servomotorn för denna länk måste jobba mot armens vikt för att hålla den i ett stabilt läge, samt att den utsätts för ett stort vridmoment när en låda lyfts upp. För att öka stabiliteten vid statiska lägen samt hjälpa motorn vid lyft anpassades designen. En krok lades till på den nedre delen av arm 1 och på toppkoppen. Krokarnas funktion var att kunna fästa ett gummiband mellan som kunde assistera att hålla armen i ett stabilt läge och hjälpa den vid lyft. Dessa krokar lades endast till på robot 2, för att hjälpa robot 1 fästes ett gummiband runt hela leden. I figurer 27 och 28 kan dessa förbättringar ses.



Figur 27: Gummiband på krokar Figur 28: Gummiband runt led

8. Testning och verifiering

När systemet bedömdes vara komplett och fungerande genomfördes tester för att utvärdera och kontrollera om kraven uppfylldes. Det gjordes tester för att utvärdera enskilda komponenter och delar samt mer övergripande tester för att verifiera funktionen av det fullständiga systemet. För att få resultat som går att jämföra med varandra användes en metod för att hantera fel på ett rättvist sätt. Om systemet gjorde ett fel under testning noterades detta och sedan återinfördes lådan på slutet igen, detta för att alla lådor skulle fullständigt behandlas.

8.1 Test av färgigenkänning

Detta test fokuserade på att verifiera färgsorteringen. Det bestod av tre delar, en del med endast röda lådor, en med blåa och en med gröna. Testet genomfördes genom att placera lådor på början av rullbandet, låta dem passera genom färgsensorn och sedan observera var robotarmen placerade lådorna för att verifiera om den hanterat dem korrekt. Varje test genomfördes med fem lådor av respektive färg. För att minimera påverkande faktorer så kördes lådorna till samma plats på bandet varje test genom att använda en fast fördröjning i stället för den slumpmässiga.

Det röda testet genomfördes genom att placera lådorna på det första rullbandet och låta robot 1 hantera dem, och det blåa gjordes på samma sätt fast på det andra rullbandet och med robot 2. Det gröna testet genomfördes genom att placera lådorna på det första rullbandet, och låta dem genomföra en hel cykel av systemet, alltså hanteras av båda robotarna. Då de gröna lådornas uppgift var att förflytas runt på rullbanden utan att lämnas av på någon avlämningsplats gjordes detta.

I detta test kontrolleras även hur de röda och blå lådorna placeras på deras avlämningsplatser. För testet med de röda och blå lådorna bedömdes att testet hade avklarats om lådan lämnades på avlämningsplatsen. För testet med de gröna lådorna bedömdes att testet hade avklarats om lådan gjorde en hel cykel av systemet.

8.2 Test av avståndsmätning

För att kontrollera noggrannheten av avståndssensorerna gjordes detta test. Det genomfördes genom att placera en låda på fyra uppmätta avstånd och observera vilken mätning som gavs av sensorn. Positionerna som valdes var 20 cm, 15 cm, 10 cm och 5 cm. För varje position togs 5 mätvärden från sensorn. Genom att mäta på fyra olika avstånd kan det fås en bild av om sensorn presterar bättre på ett visst avstånd eller om den fungerar lika på alla avstånd. Testet upprepades tre gånger för varje sensor. Detta gjordes för att minimera brus vid enstaka försök och för att få en helhetsbild.

8.3 Test av lådor på olika positioner

Detta test syftade på att kontrollera om systemet klarar av när lådor placeras på olika positioner och med olika orientering. Det genomfördes genom att placera alla 15 lådor på bandet på fem olika sätt. Det första sättet var att placera lådorna längst bak på bandet och vinkelrät mot bandets kant, men att variera var lådorna placerades i y-led. Nästa sätt var längst bak på bandet och med olika vinklar samt variation i y-led. Dessa två sätt gjordes igen fast från längre fram på bandet där robotarmen släpper av lådor. Det sista sättet var att placera lådor med 45 grader vinkel i mitten av bandet och vid robotens avlämningsposition. Testet utvärderas genom att kontrollera om lådorna passerade riktaren som är placerad i början av bandet. Detta test gjordes för båda banden.

8.4 Test av robotarmarnas precision

En av de viktigaste aspekterna av systemet är robotarmens precision, därför genomfördes ett test som utvärderar hur noggrant roboten kan uppfylla koordinater. Metoden för att genomföra detta test var att köra båda robotarmarna till 5 bestämda koordinater. (190, 0, 160), (50, 50, 200), (40, -250, 120), (0, 250, 90), (135, 5, 20). För varje koordinat mättes den faktiska positionen som roboten nådde för att utvärdera precisionen. De fem koordinaterna som valdes var positioner som används i huvudprogrammet eller positioner som kunde förekomma i robotens bana.

8.5 Minimal cykeltid

För att avgöra vilken övre gräns systemet hade på hastighet och hur olika hastigheter påverkar systemet, utfördes ett test där bandhastighet och robothastighet höjdes tills cykeltiden inte förbättrades längre. För detta test användes 5 gröna lådor och tiden mättes från att första lådan placerades på bandet tills att 5 lådor hade gjort en hel cykel. En grundnivå med inställningar som fungerade väl bestämdes för att ha en cykeltid att utgå ifrån. Efter varje lyckad körning så ökades antingen bandhastigheten eller robotens rörelsehastighet. När systemet gick så snabbt att robotarna började göra tillräckligt många misstag att cykeltiden blev sämre bedömdes testet vara klart.

8.6 Funktionstest

En bedömning på systemets helhetskvalitet behövdes och därför gjordes ett funktionstest på systemet i sin helhet. Detta test gjordes först utan fördelaren och sedan med den i gång för att få resultat som inte påverkades ifall fördelaren orsakade problem. Testet gjordes med röda och blåa lådor eftersom gröna inte hade någon avlämningsplats och endast hade agerat som fördröjning för systemet. Det gjordes tester på tre olika hastigheter av bandet. Tidtagning av varje test gjordes för att kunna utvärdera cykeltid, tiden började när första lådan var på bandet och slutade när alla lådor var sorterade.

9. Resultat

I detta kapitel presenteras resultaten av de tester och mätningar som beskrivits tidigare. Syftet är att redogöra för systemets prestanda och hur väl det kan utföra dess uppgifter.

9.1 Färgigenkänning

Nedan presenteras resultaten från testet av färgigenkänning. Bilder på hur lådorna placeras kan ses i bilaga 4.

Tabell 1. Resultat från färgtest av röda lådor

| TEST NR | Läste av rätt färg? | Ställdes alla på rätt plats? |
|---------|---------------------|------------------------------|
| 1 | Ja | Ja |
| 2 | Ja | Ja |
| 3 | Ja | Nej |
| 4 | Ja | Ja |
| 5 | Ja | Ja |

Färgtestet för röda lådor visade att färgsensorn läste av rätt färg på alla lådor under testets gång. Robotarmen fick även rätt information om vilken färg som lästs av och placerade lådan på rätt avlämningsplats. Det som var avvikande under test var test nummer tre, där rätt färg avlästes, men roboten misslyckades att plocka upp lådan. Detta resulterade i att robotarmen förflyttades tomhänt till avlämningsplatsen för röda lådor och ändå försökte lämna av en låda.

Tabell 2. Resultat från färgtest av blåa lådor

| TEST NR | Läste av rätt färg? | Ställdes alla på rätt plats? |
|---------|---------------------|------------------------------|
| 1 | Ja | Ja |
| 2 | Ja | Ja |
| 3 | Ja | Ja |
| 4 | Ja | Ja |
| 5 | Ja | Ja |

Färgtestet för blåa lådor visade att färgsensorn läste av alla lådor korrekt, samt att robotarmen plockade upp och placerade alla lådor på rätt plats i samtliga försök.

Tabell 3. Resultat från färgtest av gröna lådor

| TEST NR | Läste av rätt färg? | Ställdes alla på rätt plats? |
|---------|---------------------|------------------------------|
| 1 | Ja | Ja |
| 2 | Ja | Ja |
| 3 | Ja | Ja |
| 4 | Ja | Ja |
| 5 | Ja | Nej |

Färgtestet för de gröna lådorna visade att lådornas färg lästes av korrekt i samtliga försök. Det som var avvikande under försök fem var att den första roboten plockade upp lådan och förflyttade den till nästa rullband, men när lådan släpptes på bandet studsade den och ramlade av.

9.2 Avståndsmätning

Nedan presenteras resultaten från testet av avståndsmätning. De fullständiga resultaten kan ses i bilaga 5.

Tabell 4. Resultat från avståndstest av robot 1

| Avstånd | Genomsnittlig sensormätning |
|---------|-----------------------------|
| 200 mm | 229,3 mm |
| 150 mm | 179,0 mm |
| 100 mm | 128,7 mm |
| 50 mm | 84,7 mm |

Detta test visar att avståndssensorn för robot 1 har en nästan konstant förskjutning på cirka 30mm. Vid kortare avstånd börjar förskjutningen bli större vilket kan ses i testet på avstånd 50 mm. Denna konstanta 30 mm förskjutning är bättre att ha än en rörlig förskjutning vilket gjorde den lätt att arbeta runt i koden och få exakta mätningar.

Tabell 5. Resultat från avståndstest av robot 2

| Avstånd | Genomsnittlig sensormätning |
|---------|-----------------------------|
| 200 mm | 197,9 mm |
| 150 mm | 149,9 mm |
| 100 mm | 105,5 mm |
| 50 mm | 58,5 mm |

Detta test visar att avståndssensorn för robot 2 fungerar bäst på avståndet 150 mm och blir drastiskt sämre vid korta avstånd. Eftersom förskjutningen var väldigt olika på de olika avstånden var denna sensor inte lika pålitlig och medförde problem i precisionen för robot 2.

9.3 Riktare

Följande test avsåg att testa hur väl de 3D-utskrivna riktarna riktade upp lådorna och forslade in dem i tunneln för färgsensorn. Under testet testades olika positioner på bandet samt olika vinklar på lådorna.

Tabell 6. Resultat från test för riktare på rullband 1

| Test Nr | Lådans vinkel i grader | Lådans Position på bandet | Antal passerade lådor |
|---------|------------------------|---------------------------|-----------------------|
| 1 | 0 | Längst bak | 15/15 |
| 2 | 0 | Längst bak | 15/15 |
| 3 | Olika Vinklar | Längst bak | 14/15 |
| 4 | Olika Vinklar | Längst bak | 14/15 |
| 5 | 0 | Robot avlämning | 14/15 |
| 6 | 0 | Robot avlämning | 15/15 |
| 7 | Olika Vinklar | Robot avlämning | 15/15 |
| 8 | Olika Vinklar | Robot avlämning | 15/15 |
| 9 | 45 | Robot avlämning | 12/15 |
| 10 | 45 | Robot avlämning | 11/15 |

Detta test utfördes för att testa funktionaliteten hos riktaren på band nummer 1. Lådans vinkel avser lådans infallsvinkel kontra en horisontell linje. Noll grader avser att den

platta sidan av lådan färdas rakt in mot tunneln. "Olika vinklar" avser att lådorna ställs utan någon speciell vinkel, vilket medför att de förs in mot riktarna olika varje gång. 45 grader avser när lådan har roterats 45 grader från noll, och således färdas med sitt hörn rakt in mot tunneln. Varje deltest utfördes med 15 lådor, där positionen i sidled på bandet ändrades kontinuerligt för testen med vinkel 0 samt "olika vinklar". För testen med vinkel 45 placerades lådan i mitten av rullbandet, detta då det identifierades att detta var den mest utmanande positionen. Lådans position på bandet avser positionen i längdriktningen. Positionen "Längst bak" avser så långt bak man kan komma på bandet och "robot avlämning" avser något längre fram, där roboten vanligtvis lämnar av lådor.

Från testresultatet syns det att både 0 grader och "olika vinklar" oftast hanteras helt korrekt av riktaren, vilket betyder att riktaren riktade upp lådan och forslade in den i tunneln utan problem. Av de 8 testen fås resultatet 15/15 fem gånger, och resultatet 14/15 tre gånger, så av totalt 120 lådor fastnade bara 3 lådor. Med fastna avses att lådan fastnade i mer än 5 sekunder och inte kom vidare. Något händer dock när man ser till test 9 och 10 där den utmanande positionen 45 grader testas. Då lådorna även placeras i mitten av bandet kan inte riktaren utnyttjas för att rikta upp lådorna. Detta medförde att några lådor kilade sig fast mellan riktarens slut och tunneln för färgsensorn. Totalt testades 30 lådor, och 7 fastnade.

Tabell 7. Resultat från test för riktare på rullband 2

| Test Nr | Lådans vinkel i grader | Lådans Position på bandet | Antal rätt |
|---------|------------------------|---------------------------|------------|
| 1 | 0 | Längst bak | 15/15 |
| 2 | 0 | Längst bak | 15/15 |
| 3 | Olika Vinklar | Längst bak | 14/15 |
| 4 | Olika Vinklar | Längst bak | 12/15 |
| 5 | 0 | Robot avlämning | 15/15 |
| 6 | 0 | Robot avlämning | 15/15 |
| 7 | Olika Vinklar | Robot avlämning | 14/15 |
| 8 | Olika Vinklar | Robot avlämning | 15/15 |
| 9 | 45 | Robot avlämning | 11/15 |
| 10 | 45 | Robot avlämning | 12/15 |

I testet för riktarna på rullband 2 ses ett likande resultat som på rullband 1. Riktaren hanterar 0 grader och "olika vinklar" bra med ett totalt resultat på 115/120 korrekt hanterade lådor, och 5/120 som fastnade. Positionen 45 grader visar sig fortfarande vara utmanande för systemet och ger ett totalt resultat på 23/30 lådor korrekt hanterade och 7/30 lådor som fastnade.

9.4 Robotprecision

Nedan presenteras resultaten från precisionstestet.

Tabell 8. Resultat från precisionstest av robot 1

| Koordinat | Mätt position |
|-----------------|-----------------|
| (190, 0, 160) | (192, 0, 156) |
| (50, 50, 200) | (51, 50, 204) |
| (40, -250, 120) | (45, -260, 118) |
| (0, 250, 90) | (0, 255, 95) |
| (135, 5, 20) | (144, 5, 18) |

För att beräkna det genomsnittliga felet per riktning läggs felet för varje position ihop och delas med antalet mätningar. Det som beräknas är det genomsnittliga absoluta felet. För robot 1 är det genomsnittliga felet följande: X-led: 3,4 mm, Y-led: 3,0 mm, Z-led: 3,4 mm.

Tabell 9. Resultat från precisionstest av robot 2

| Koordinat | Mätt position |
|-----------------|-----------------|
| (190, 0, 160) | (192, 0, 149) |
| (50, 50, 200) | (55, 50, 193) |
| (40, -250, 120) | (45, -250, 102) |
| (0, 250, 90) | (0, 250, 82) |
| (135, 5, 20) | (135, 10, 16) |

För robot 2 är det genomsnittliga felet följande: X-led: 2,4 mm, Y-led: 1,0 mm, Z-led: 9,6 mm. Dessa resultat måste tolkas försiktigt eftersom mätningen var väldigt svår att göra noggrant. Mätningarna gjorde med en linjal och det var svårt att mäta exakt från origo till robotens läge då dessa punkter ligger inuti robotarnas delar på grund av hur de är programmerade.

9.5 Cykeltid

Nedan presenteras resultaten från testet av minimal cykeltid

Tabell 10. Resultat från basnivåtest

| | |
|--------|--------------|
| Test 1 | Tid: 1:52,04 |
| Test 2 | Tid: 1:49,72 |

Inställningarna på basnivå testet var bandhastighet 200, robothastighet 10 och robotfördröjning 20 vilket är de inställningar som har använts mest genom utvecklingen. Bandhastigheten 200 är en PWM-signal som skickas till motordrivaren för rullbandet och motsvarar ungefär 81,4 mm/s. Robothastigheten 10 motsvarar att varje varv av rörelselooopen så ökar styrsignalen som skickas till servona med 10. Till exempel om bas servot ska från pulsbredd 400 till 500 kommer detta göras i 10 varv av rörelselooopen. Robotfördröjning 20 innebär att i slutet av varje varv av rörelselooopen så väntar roboten i 20 ms.

Tabell 11. Resultat från första ökningen

| | |
|--------|--------------|
| Test 1 | Tid: 1:18,36 |
| Test 2 | Tid: 1:19,63 |

Inställningarna för detta test var bandhastighet 200, robothastighet 20 och robotfördröjning 20. Denna ökning av robothastigheten gav en tydlig förbättring i cykeltid. Varje rörelse som robotarna gjorde under detta test var ungefär dubbelt så snabb som basnivån. Ett problem som uppstår vid dessa hastigheter är att servona börjar röra sig våldsamt och kan skada sig själva. Efter observation av robotarnas rörelse under detta test bedömdes det att denna robothastighet är den högsta som kommer testas.

Tabell 12. Resultat från andra ökningen

| | |
|--------|--------------|
| Test 1 | Tid: 1:45,62 |
| Test 2 | Tid: 1:12,26 |
| Test 3 | Tid 1:20,40 |

Inställningarna för detta test var bandhastighet 225, robothastighet 20 och robotfördröjning 20. Nu var bandhastigheten också ökad och systemet började få problem. Antalet misstag som observerades vid denna hastighet var mycket högre än tidigare vilket gjorde att i genomsnitt blev cykeltiden högre även fast banden körde mycket snabbare.

9.6 Totalt systemtest

Nedan presenteras resultaten för det fullständiga systemtestet.

Tabell 13. Resultat från fullständigt systemtest

| Typ av test | Tid | Röd avlämning | Blå avlämning | Plockmisstag | Tappade lådor | Fördelare stopp | Riktare stopp | Låda av band |
|-------------|------|---------------|---------------|--------------|---------------|-----------------|---------------|--------------|
| 1 | 3:51 | | | | 3V | | | |
| 1 | 3:35 | | | 2S | | | | |
| 2 | 4:44 | | | V | V | | | |
| 2 | 5:17 | 1 misstag | | SV | V | | S | |
| 3 | 3:49 | 1 misstag | | | 3V | | V | V |
| 33 | 3:47 | 2 misstag | 1 misstag | | V | | V | 3V |
| 4 | 2:30 | | | | | | | |
| 4 | 2:49 | | | | S | 1 misstag | V | |

Tabellförklaring:

V = Robot 1 eller band 1

S = Robot 2 eller band 2

Typer av test:

1 = En låda i taget, bandhastighet 200

2 = En låda i taget, bandhastighet 175

3 = En låda i taget, bandhastighet 225

4 = Flera lådor på samma gång, bandhastighet 200

Felen som noterades var följande: felplacering vid avlämningsplats, plockmisstag, tappad låda, fördelare stopp, riktare stopp, låda ramlade av band. Det tydligaste resultatet från detta test är att robot 1 greppade lådor lite för svagt eftersom den tappade många fler lådor än robot 2. Vid hastigheten 225 började lådor ramla av bandet vilket tyder på att stoppzonen inte var rätt kalibrerad för den hastigheten. Över lag gör roboten diverse misstag då och då vilket tyder på att antingen sensormätningarna inte är pålitliga eller att styrningen inte är tillräckligt robust.

10. Slutsats och Diskussion

Detta kapitel handlar om att besvara frågeställningarna som ställdes i kapitel 1.4, dra slutsatser om systemets prestanda samt att diskutera erfarenheter och idéer som uppkommit under projektets gång.

10.1 Svar på frågeställningar

- Hur kan robotarmar användas för automatisk sortering av lådor utifrån färg?

Med hjälp av färgsensorer och programmerad styrlogik kunde robotarmar användas för att automatiskt sortera lådor utifrån färg. Resultaten från färgtestet som gjordes visar även att färgigenkänningen fungerade varje gång utan problem. Det som inte fungerade varje gång var robotarnas styrning vilket tyder på att den kan utvecklas vidare för att få ett mer robust system.

- Är det möjligt att konstruera en robotarm som på ett tillförlitligt sätt klarar av att plocka upp lådor från olika platser i ett område?

Det är möjligt för robotarmen som har konstruerats att plocka upp lådor från flera positioner i upphämtningszonen. Detta visas i det totala systemtestet som gjordes där hela upphämtningszonen nyttjades för hämtning. En observation som gjordes var att robotarmarna hade fler problem med lådor längst ut i änderna av zonen. Det var där de flesta av plockmisstagen och de tappade lådorna skedde. Detta kan tyda på att avståndssensorn inte var lika bra kalibrerad i dessa områden samt att robotens styrning inte var tillräckligt noggrann.

- Hur påverkar rullbandets hastighet systemets prestanda och noggrannhet?

Testerna som gjordes visade att en ökning av rullbandets hastighet påverkade systemets prestanda negativt. Stoppzonen kalibrerades manuellt på normalhastigheten som var 200. När bandhastigheten ändrades skulle stoppzonen automatiskt beräknas utifrån den nya hastigheten, men denna uträkning var inte tillräckligt noggrann vilket ledde till att lådor hamnade utanför zonen vid andra hastigheter än 200. Förutom detta gjorde den ökade hastigheten att sensorer fick mindre tid att göra sina mätningar vilket ledde till mer fel från sensorer.

- Vad kan man göra för att hantera överflöd av lådor på rullbandet?

För att hantera överflöd av lådor på rullbandet infördes en mekanisk fördelare som temporärt blockerade nya lådor medan den föregående hanterades. Denna fördelare fungerade väl när sensorn som styrde den gav pålitliga mätningar, men det gjorde den inte hela tiden. På grund av monteringen av sensorn kunde den ibland ge felaktiga mätningar vilket gjorde att fördelaren slutade fungera rätt. Ett annat problem som uppstod när det var överflöd av lådor var att lådorna började fastna i sidled och ibland staplas ovanpå varandra när roboten lämnade nya lådor när det redan var fullt. För att lösa dessa problem hade det behövts en annan design av fördelaren och riktaren som möjliggjorde guidning av lådorna i flera riktningar än bara sidled.

- Hur påverkar olika faktorer som sensorers noggrannhet och den mekaniska konstruktionen systemets prestanda?

Sensorernas noggrannhet påverkade systemet väldigt mycket på ett antal olika sätt. Resultatet av avståndsmätningen visade att den sensorn hade en felmarginal och mycket brus. Detta ledde till att robotarna inte alltid körde till det mest optimala läget för

upphämtning av lådor. Innan solskyddet monterades för färgsensorn hade den väldigt stora problem med omgivande ljusnivåer. Resultaten var inte alls pålitliga och kunde ändras mycket under dagen beroende på hur mycket ljus som kom in i rummet som systemet stod i. Dessa sensorproblem beror förmodligen på att kvaliteten av de valda sensorerna inte är av högst grad samt att systemet förlitar sig på mätvärdena från endast en sensor.

Den mekaniska konstruktionen påverkade också systemet på olika sätt. I testet av riktaren upptäcktes det att den hade en stor svaghet när lådor placerades i mitten av bandet med en vinkel på 45 grader. Denna position kunde göra att lådor fastnade vilket tyder på att designen av riktaren inte är optimal. Resultaten från robotprecision testet visade att robotarna inte alltid nådde de exakta positionerna som de skulle nå. Detta kan bero på antingen att konstruktionen har problem eller att mjukvaran som styr robotarmarna inte var helt korrekt, eller en kombination av båda. För att avgöra tydligare vad felet beror på hade det behövts fler tester och mer robusta mätmetoder.

- Hur väl presterar det framtagna systemet, samt vilka begränsningar har den?

Generellt presterar det framtagna systemet väl då den kan lösa sin uppgift. De situationer där systemet börjar få problem är vid extrem belastning och gränsfall. Resultaten från cykeltid testet visade att när systemets pressas till sitt yttersta börjar servona få problem och systemet begår mycket fler fel. Funktionstestet visade att det finns vissa begränsningar som systemet har. På grund av valet av frihetsgrader och hur griplon är monterad så har robotarmarna inte möjlighet att plocka lådor på bästa sätt vid de yttersta kanterna av stoppzonen. Detta är en begränsning som ledde till tappade lådor och suboptimal placering av lådor.

10.2 El och ström

När beräkningarna för strömförbrukningen gjordes användes den maximala strömmen som servona drar när rotorn är låst men fortfarande försöker driva den. Detta ledde till att den uträknade strömförbrukningen blev väldigt hög, på hela 23,4 A för alla servon. Att alla servon skulle utsättas för denna typ av belastning, och även samtidigt, är väldigt orealistiskt. Detta har inte medfört några problem för systemet men har potentiellt lett till att strömförsörjningen blivit överdimensionerad. Detta har i sin tur medfört extra kostnader för komponenter, i form av kraftigare strömadapter, kraftigare kablar samt Buck-konverterare.

Strömförsörjningen som användes hade en märkspänning på 12 V men när den testades med en multimeter mättes den till 13,5 V. Detta lyfte frågor om den skulle vara säker att använda som strömkälla för arduinon. Arduino Uno fungerar bäst när den matas med mellan 7–12 V. Allt över 12 V kan leda till att spänningsregulatorn överhettas. Det valdes att fortsätta med den strömförsörjningen som mättes till 13,5 V och hålla god koll på värmegraden av arduinons spänningsregulator. Vid vissa längre körningar av systemet blev den varm att röra, men det blev aldrig ett större problem.

10.3 Konstruktion

När det kommer till själva konstruktionen av roboten har ett antal problem uppkommit på vägen. För det första så levererades servona med tillhörande servohorn som fästes på servoaxeln. Dessa servohorn var i plast, medan servoaxeln var av metall. Då roboten utsätts för ganska stora moment uppstod glapp och rörelser där servohornet fäster in i servoaxeln. Detta hade kunnat förminskas om servohornet hade varit av metall.

Ett annat problem med konstruktionen av robotarna var att det fanns glapp i många av delarna som skulle sitta ihop med varandra, till exempel lederna. Detta glapp kommer från att delarna skapades via 3D-printing. Delar som skrivs ut med denna metod kan krympa eller expandera på olika sätt. Detta var i åtanke under designen och delar som skulle sättas ihop designades därför med toleranser. Vissa av dessa toleranser blev inte perfekta och ledde därför till glapp eller förskjutningar av vissa delar.

Vissa delar på robotarna har hål för skruvar. Dessa delar var speciellt svåra att få bra kvalitet på utskrifterna. Många gånger blev hål ifyllda av 3D-printern och behövdes borrar ut eller i extrema fall skrivs ut igen. Den mest troliga orsaken till att hålen blev så dåliga kan vara för att slicern placerade supports inuti hålen som sen inte gick att få ut för dem smälte ihop med resten av delen. När detta upptäcktes började supportblockerare användas och det hjälpte att motverka problemet.

Tidigt i processen vid konceptgenereringen bestämdes det att robotarmen endast skulle ha 5 DOF i stället för 6. Detta gjorde att klons rörlighet skulle bli begränsad. Valet landade på att ha funktionen roll, i stället för Yaw. Detta medförde problem senare i processen då det var lite sent att ändra på designen. När lådorna stannar i slutet eller i början av stop-zonen har robotarmen ibland svårt att komma in i rätt vinkel för att plocka upp lådan på rätt sätt. Om roboten i stället hade försatts med Yaw funktion hade lådan kunnat plockas upp rakt framifrån, överallt i stopp-zonen. Rollfunktionen visade sig vara i princip oanvändbar för att underlätta upphämtningen i dessa lägen.

10.4 Servon

Det svåraste problemet som stöttes på i utvecklingen kom väldigt tidigt och följde med under hela processen. Problemet var att när servon stängs av kan de flyttas manuellt. När denna förflyttning sker vid avstängt läge uppdateras inte den interna positionsgivaren i servot på rätt sätt. När ström sedan skickas till servot kommer den att hastigt köras till det läget den var på vid senaste avstängning. Om servot då har flyttats vid avstängt läge kan denna hastiga förflyttning innebära stora problem. Ingen fullständig lösning hittades för detta problem, men ett sätt att kringgå det hittades. En säker position definierades i koden där roboten pekar rakt ut i en 90-gradig vinkel. Innan strömmen sätts på måste roboten fysiskt hållas upp i närheten av denna position och i början av koden körs alla servon till denna position. Denna metod minimerar servo snärten som hade skett om den riktiga positionen av ett servo var långt ifrån var koden antog att den var. Detta är dock inte en helt bra lösning för om en person som inte vet detta ska sätta på systemet kommer roboten snärta upp till den "säkra" positionen väldigt våldsamt vilket inte är optimalt.

Gripklon är utrustad med ett SG90 servo, vilket är ett väldigt billigt och låg kvalitativt servo. Detta i kombination med att servot är ganska svagt gjorde att det under drift blev väldigt varmt, samt då och då gav ifrån sig knäppande ljud. För att försöka lösa problemet kopplas strömmen på servot bort när det inte används, men detta hjälpte inte tillräckligt. Ett förbättringsalternativ hade kunnat vara att byta ut servot mot en något kraftigare modell samt av högre kvalitet.

10.5 Sensorer och mjukvara

Kvalitén på sensorerna var även den ett stort problem. Då sensorerna är systemets enda referenspunkt var de en viktig aspekt för att få ett någorlunda precist system. De sensorerna som köptes in var billiga och därmed av låg kvalitet vilket skadade systemets precision. Lösningen för att hantera störningar från avståndssensorn var att använda sig utav ett medelvärdesfilter som tog in ett antal mätningar och sedan tog ett medelvärde utav dessa. Detta hjälpte något men det hade kunnat bli bättre om man i stället implementerade till exempel ett kalmanfilter. Detta hade krävt lite mer CPU kraft samt mer modellering men hade troligtvis lett till ett mer exakt resultat. För att förbättra sensormätningarna ännu mer hade flera typer av exempelvis avståndssensorer kunnat implementeras för att sedan slås ihop med hjälp av sensorfusion. Då hade man kunnat implementera exempelvis en ultraljudssensor som också mäter avståndet på lådorna. Detta hade även kunnat hjälpt till i problemet med att ToF-sensorn hade dålig precision vid kortare avstånd. Då hade man kunnat vikta mätresultatet så att mätdata från ultraljudssensorn väger tyngre på korta avstånd.

När invers kinematik beräkningarna skapades var den ursprungliga planen att först lösa (x, y, z) positionen och sedan lösa orienteringen som krävdes. Eftersom robotens konstruktion endast hade roll och pitch var det inte möjligt att nå alla orienteringar. På grund av detta samt brist av tid bestämdes det att endast lösa (x, y, z) positionen med invers kinematik och att manuellt koda roll och pitch för ett antal olika fall. I efterhand var detta ett dåligt beslut och tog förmodligen mer tid än att skapa en rotationsmatris för roll och pitch. Den manuella kodningen av pitch och roll var inte anpassad för alla de olika lägena som lådan kunde stanna på. En annan svårighet med invers kinematik beräkningarna var att relatera theta vinklarna som gavs med den fysiska konstruktionen. De tre theta vinklarna som ska ge positionen för den sökta punkten behövdes manipuleras och förskjutas på olika sätt för att matcha hur servonas intervall och riktningar fungerade. Denna process innebar mycket testning och verifiering tills theta vinklarna fungerade med servona.

När fördelaren skulle implementeras i systemet gjordes detta först med `servo.h` biblioteket och den kopplades till pin 3 på arduinon. När systemet sattes på direkt efter detta hade rullbandet slutat fungera. Efter undersökningen upptäcktes det att `servo.h` biblioteket skriver över pins 9 och 10 vilket användes till rullbandet. För att lösa detta kopplades servot till fördelaren in i servodrivaren i stället och då började rullbandet fungera igen.

Under utvecklingen av mjukvaran för systemet skapades flera koddokument för olika ändamål. Ett huvuddokument per robot, ett dokument för manuell styrning per robot samt många andra för tester och övrigt. Anledningen varför det behövdes olika dokument för robotarna var att servona skiljde sig mellan dem och det ledde till att styrningen blev lite annorlunda mellan dem. Under testningen behövdes hastigheter och inställningar ändras flera gånger. För att göra detta smidigt utnyttjades `defines` som snabbt kunde ändras högst upp i dokumentet för att kontrollera hur inställningarna skulle vara ställda. En idé som uppkom av detta var att samla mer kod i samma dokument och att ha `defines` som bytte mellan vilken kod som skulle vara aktiv. Det hade samlat mer kod till samma ställe men också gjort läsningen av koden svårare.

10.6 Etik och Hållbarhet

Detta arbete påvisar hur automatisering kan bidra till ett mer hållbart och effektivt industrisamhälle. Genom att ersätta eller komplettera manuella sorteringsmoment i industrin kan man skapa en bättre arbetsmiljö där upprepade, monotona och farliga arbetsuppgifter kan automatiseras. Detta leder i sin tur till att skador och misstag kan undvikas vilket ökar säkerheten på arbetsplatsen.

Ur ett hållbarhetsperspektiv kan ett automatiserat system bidra till mindre produktionsfel, vilket i sin tur leder till att resurser utnyttjas på ett mer effektivt sätt. Genom att effektivisera processer och optimera energianvändning kan automatisering bidra till att minska utsläpp av växthusgaser. Då systemet som implementeras innehåller många olika komponenter bör en grundlig genomgång av alla ingående komponenters påverkan på miljön göras. Detta genom utförda livscykelanalyser som granskar hela systemets påverkan på miljön och dess ingående komponenters förmåga att återvinnas vid avveckling.

Inverkan på samhället i stort bör även diskuteras, då automatiserade system ofta påverkar arbetsmarknaden. Det kan leda till att arbetsuppgifter försvinner om de ersätts av exempelvis robotar, men nya roller skapas samtidigt inom programmering, underhåll och driftsättning. Det är därför viktigt att det i samhället finns underliggande ramverk som stödjer denna typ av omställning, och att möjligheterna till utbildning inom dessa områden uppmuntras.

Genom utvecklingen av ett automatiserat system för sortering av lådor baserat på färg påvisar arbetet att det är möjligt att skapa ett automatiserat sorteringsystem med begränsade resurser. Detta kan hjälpa mindre företag att se potentialen av ett automatiserat system och se dess fördelar ur ett hållbarhetsperspektiv. Systemet ska användas på arbetsmässor där de visas upp för unga personer. Genom att visa upp en modell där teknik används för att automatisera en verklig process kan väcka intresse för teknik och ingenjörsyrket. Det kan på så vis inspirera fler personer att utbilda sig inom ett tekniskt område, vilket i sin tur leder till ett mer hållbart tekniksamhälle.

Referenser

- [1] Munro, O. (2023). *Automation in Manufacturing: Uses, Examples, & Trends*. [online] Unleashedsoftware.com. Available at: <https://www.unleashedsoftware.com/blog/automation-in-manufacturing/> [Accessed 19 Apr. 2025].
- [2] www.st.com. (n.d.). *VL53L0X - Time-of-Flight ranging sensor - STMicroelectronics*. [online] Available at: <https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html> [Accessed 20 Apr. 2025].
- [3] Santos, R. and Santos, S. (2017). *Arduino Color Sensor TCS230 TCS3200 | Random Nerd Tutorials*. [online] Random Nerd Tutorials. Available at: <https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/> [Accessed 20 Apr. 2025].
- [4] Adafruit Industries (n.d.). *Adafruit 16-Channel 12-bit PWM/Servo Driver - I2C interface*. [online] www.adafruit.com. Available at: <https://www.adafruit.com/product/815> [Accessed 20 Apr. 2025].
- [5] Adafruit Industries (n.d.). *Adafruit DRV8871 DC Motor Driver Breakout Board - 3.6A Max*. [online] www.adafruit.com. Available at: <https://www.adafruit.com/product/3190> [Accessed 20 Apr. 2025].
- [6] Monolithic Power Systems (n.d.). *Buck Converters (Step-Down Converter)*. [online] www.monolithicpower.com. Available at: <https://www.monolithicpower.com/en/learning/mpscholar/power-electronics/dc-dc-converters/buck-converters> [Accessed 20 Apr. 2025].
- [7] Campbell, S. (2016). *Basics of the I2C Communication Protocol*. [online] Circuit Basics. Available at: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/> [Accessed 20 Apr. 2025].
- [8] University of Illinois Urbana-Champaign (n.d.). *InverseKinematics*. [online] motion.cs.illinois.edu. Available at: <https://motion.cs.illinois.edu/RoboticSystems/InverseKinematics.html> [Accessed 20 Apr. 2025].
- [9] Formlabs (n.d.). *3D Printing Guide: Types of 3D Printers, Materials, and Applications*. [online] Formlabs. Available at: <https://formlabs.com/eu/3d-printers/> [Accessed 20 Apr. 2025].

[10] Daley, S. (2022). *What is 3D Printing? How do 3D Printers Work? What can be 3D Printed? / Built In.* [online] builtin.com. Available at: <https://builtin.com/3d-printing> [Accessed 20 Apr. 2025].

[11] Jost, D. (2019). *What is an Ultrasonic Sensor?* [online] FierceElectronics. Available at: <https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor> [Accessed 20 Apr. 2025].

[12] 3DSourced. (2021). *The 10 Coolest DIY 3D Printable Robotic Arm Projects - 3DSourced.* [online] Available at: <https://www.3dsourced.com/guides/3d-printed-robotic-arms/> [Accessed 20 Apr. 2025].

Bilagor

Bilaga 1

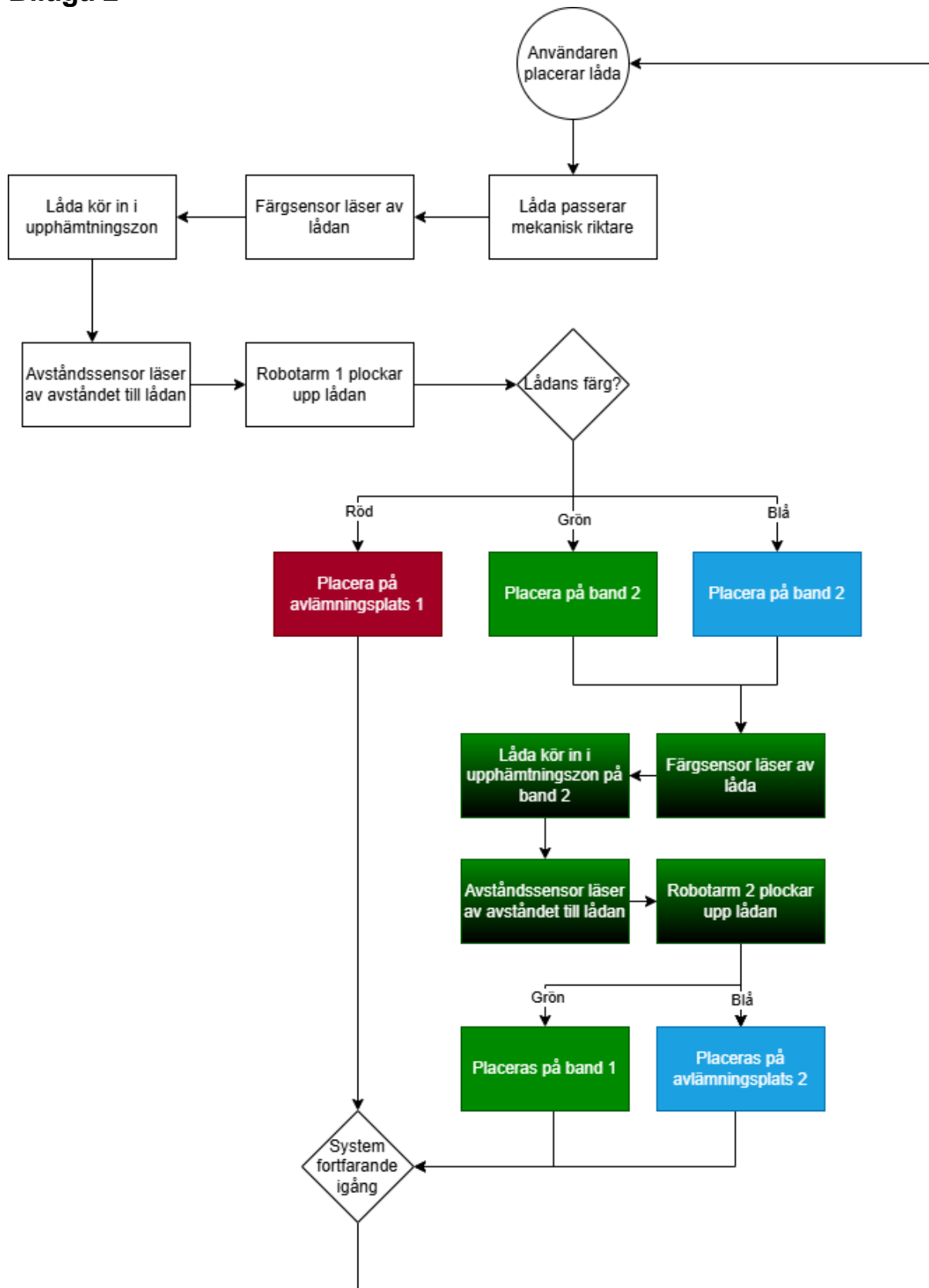
TIDSLINJE FÖR EXAMENSARBETE

| | | | |
|---------------|--|--------------|----------------------|
| PROJEKTNAMN | Simulerad Produktionsmiljö med Robotarmar och Rullband | FÖRETAGSNAMN | Broccoli Engineering |
| PROJEKTMEDLEM | Linus Ekedahl | DATUM | 25-02-03 |
| PROJEKTMEDLEM | Joshua Rutledge | | |

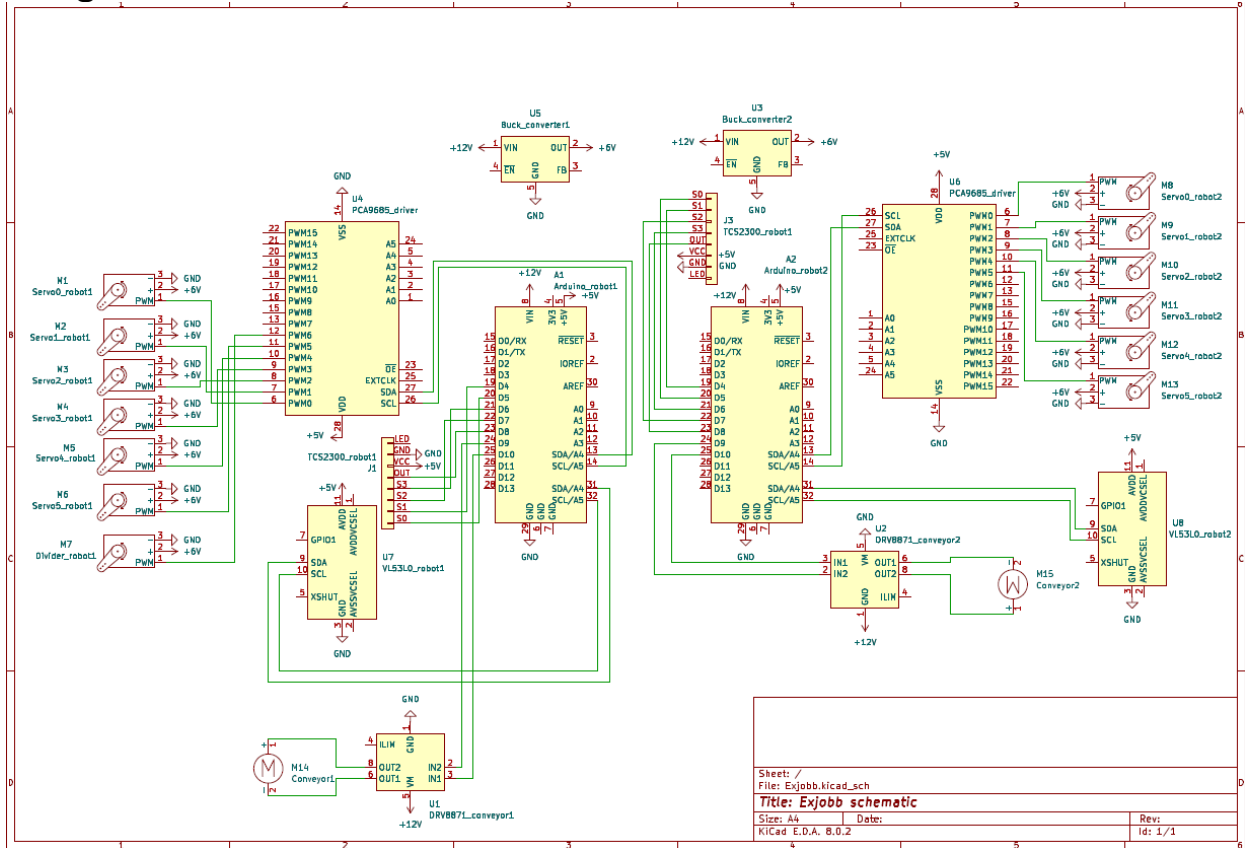
| FAS: | | DETALJER: | | PROJEKTVECKA: | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-----------------------------|---|--|---------------|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--|--|--|
| | | | | V.4 | V.5 | V.6 | V.7 | V.8 | V.9 | V.10 | V.11 | V.12 | V.13 | V.14 | V.15 | V.16 | V.17 | V.18 | V.19 | V.20 | V.21 | V.22 | V.23 | V.24 | V.25 | | | |
| 1 | Upptstartsfas | - Planeringsrapport - Inhämtning av information - Kravspecifikation | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Koncept och design | - Konceptgenerering - Konceptutvärdering - Val av komponenter och beställning - Design av hårdvara | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Programmering | - Styring av robotarm - Sensorer - Styring av rullband | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Byggnation och testning | - 3D-utskrift av delar - Montering - Elektronikkoppling - Testning | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Rapportskrivning | - Inledning - Teori - Metod och genomförande - Resultat och diskussion - Slutjusteringar | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Presentation och opponering | - Presentation - Opponering | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Kriterier | Målvärde | Kravtyp | Intressent | Verifiering |
|---------------------------|---|----------|---------------------------------------|----------------|
| 1. Funktion | | | | |
| 1.1 Rörelse av Robotarmar | 5 Frihetsgrader vardera | Krav | Projektgrupp | Observation |
| 1.2 Gripklo | Ska kunna öppnas med minst 4 cm | Krav | Projektgrupp | Analys/Mätning |
| 1.3 Upphämtnings zon | Robotarmarna ska kunna plocka upp lådor inom en upphämtnings zon som är minst 10 cm lång. | Önskemål | Projektgrupp | Testning |
| 1.4 Färgigenkänning | Systemet ska kunna identifiera och klassificera lådor baserat på färg | Krav | Projektgrupp/ Broccoli Engineering | Testning |
| 1.5 Sortering | Robotarmarna ska kunna plocka upp och placera lådor på rätt plats | Krav | Projektgrupp/ Broccoli Engineering | Testning |
| 1.6 Autonomi | Systemet ska kunna flytta lådor utan konstant mänskligt ingrepp | Önskemål | Projektgrupp/ Broccoli Engineering | Testning |
| 1.7 Kapacitet | Systemet ska kunna hantera minst 5 lådor samtidigt | Krav | Projektgrupp | Testning |
| 1.8 Lådor i rörelse | Robotarmarna ska kunna plocka upp lådor i rörelse. | Önskemål | Projektgrupp/ Broccoli Engineering | Testning |
| 1.9 Former på lådorna | Robotarmarna ska kunna plocka upp lådor av olika former. | Önskemål | Projektgrupp/ Broccoli Engineering | Testning |
| 2. Prestanda | | | | |
| 2.1 Lastkapacitet | Minst 50 gram | Önskemål | Projektgrupp | Testning |
| 3. Underhåll | | | | |
| 3.1 Byta ut delar | Max 20 min | Önskemål | Projektgrupp | Testning |
| 4. Storlek | | | | |
| 4.1 Dimensioner | 60x60cm | Krav | Broccoli Engineering | Mätning |
| 4.2 Vikt | <20kg | Krav | Projektgrupp | Mätning |
| 5. Komponenter | | | | |
| 5.1 3D-Utskrivet material | Alla egen designade delar i CAD skrivs ut med 3D-skrivare | Önskemål | Projektgrupp | Observation |
| 5.2 Inköp av komponenter | Komponenter ska i huvudsak köpas in från Europeiska företag | Krav | Broccoli Engineering | Analys |

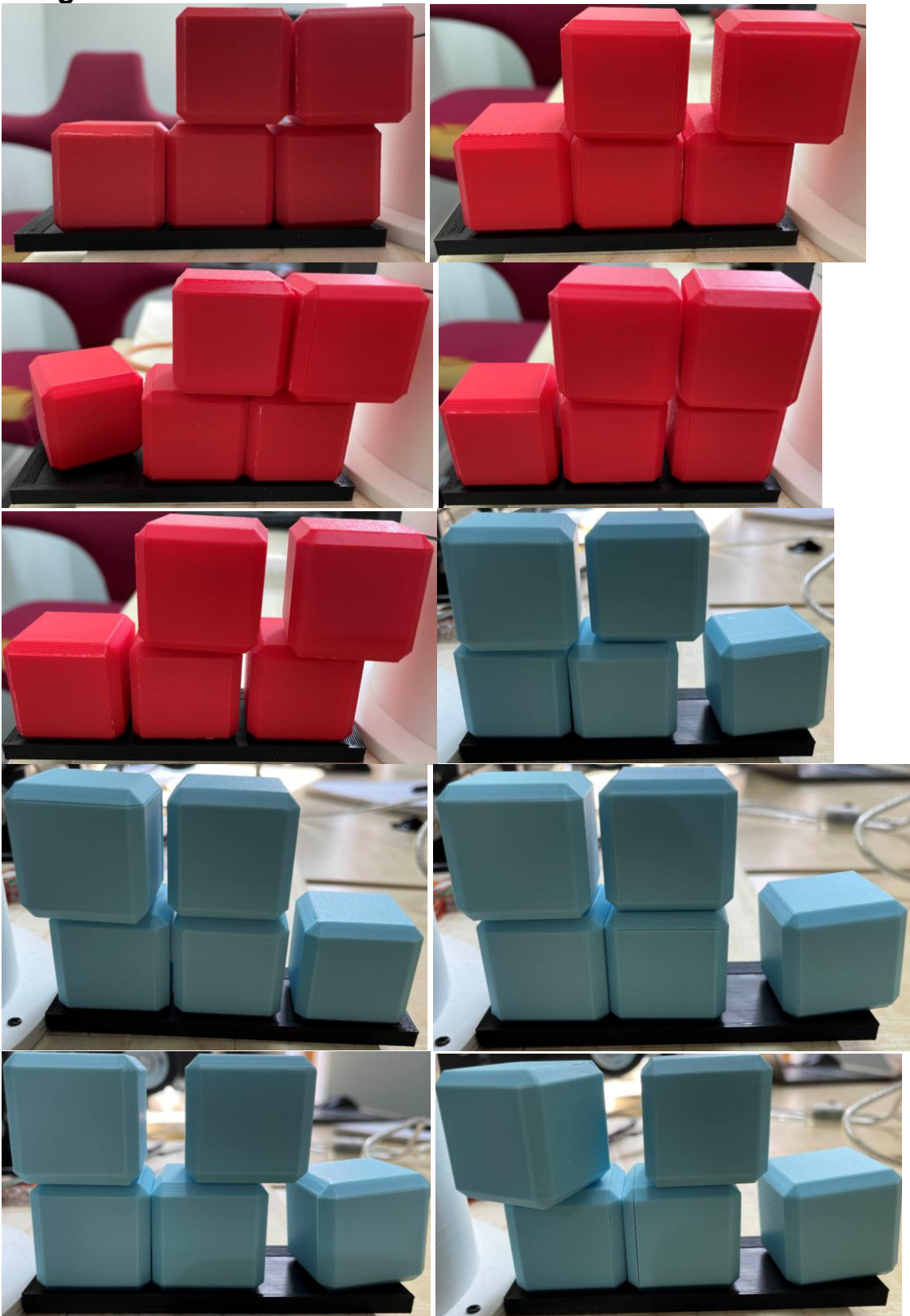
Bilaga 2



Bilaga 3



Bilaga 4



Bilaga 5

| Robot 2 Test | | | | | | | | | | |
|--------------|-------------|-----|-----|-----|-----|-----|-----|-----|-------------|--|
| | Mätvärde # | 1 | 2 | 3 | 4 | 5 | 6 | | | |
| | Avstånd 200 | 200 | 198 | 199 | 198 | 198 | 198 | | | |
| | Avstånd 150 | 151 | 149 | 150 | 150 | 150 | 150 | | | |
| | Avstånd 100 | 106 | 106 | 107 | 107 | 107 | 107 | | | |
| | Avstånd 50 | 60 | 59 | 59 | 61 | 59 | 58 | | | |
| Robot 1 Test | | | | | | | | | | |
| | Mätvärde # | 1 | 2 | 3 | 4 | 5 | 6 | | | |
| | Avstånd 200 | 228 | 227 | 230 | 229 | 228 | 231 | | | |
| | Avstånd 150 | 179 | 180 | 180 | 179 | 180 | 179 | | | |
| | Avstånd 100 | 127 | 129 | 128 | 129 | 129 | 129 | | | |
| | Avstånd 50 | 84 | 85 | 85 | 85 | 85 | 86 | | | |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | avg | |
| 198 | 198 | 199 | 198 | 198 | 197 | 197 | 195 | 198 | 197,9333333 | |
| 150 | 150 | 150 | 150 | 150 | 149 | 150 | 148 | 149 | 149,8666667 | |
| 104 | 106 | 105 | 106 | 105 | 104 | 105 | 105 | 105 | 105,5333333 | |
| 58 | 59 | 58 | 59 | 58 | 57 | 58 | 57 | 58 | 58,5333333 | |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | |
| 230 | 231 | 227 | 231 | 229 | 230 | 230 | 229 | 230 | 229,3333333 | |
| 180 | 179 | 179 | 178 | 179 | 179 | 178 | 177 | 179 | 179 | |
| 128 | 129 | 129 | 129 | 129 | 129 | 129 | 130 | 128 | 128,7333333 | |
| 86 | 85 | 85 | 84 | 84 | 84 | 84 | 84 | 84 | 84,6666667 | |