



Autonom robot för knapptryckning Autonomous robot for button pressing

SSYX05

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

Marcus Thersthol

Examinator Manne Stenberg
Institutionen för signaler och system
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige, 2015

Autonom robot för knapptryckning
Autonomous robot for button pressing
Marcus R.P. Thersthol

© Marcus R.P. Thersthol, 2015

Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Phone +46 (0)31- 771 1000

Cover:

The coverpicture shows the finished product. A dashboard is visible in the background.

Photographer: Marcus Thersthol

Department of Signals and Systems
Göteborg, Sweden 2015

Förord

Att i slutet av juni år tvåtusenfemton skriva de sista raderna på en examensarbetsrapport är inte helt självklart. För cirka tre år sedan påbörjades en utbildning vid Chalmers Tekniska Högskola, Mekatronikingenjörsprogrammet på 180 högskolepoäng. Dagen innan samlingen vid Götaplatsen, hade jag ännu inte bestämt mig för om jag skulle åka upp och påbörja utbildningen. Runt fyratiden på eftermiddagen svängde jag in på min pappa Peters jobb och där blev jag övertalad att åka upp och göra mitt bästa. Det blev till att packa snabbt och tack vare mamma Helena hann jag med sista tåget upp till Göteborg. Jag hade en liten stund dessförinnan slagit en pling till min farmors syster, som trots att en stressad Skåning ringt bara några timmar innan, ändå kunde hysa honom på köksgolvet, i sin trea på Olivedalsgatan, och detta i över tre månader. Tilläggas bör att segermaskinen IFK Göteborg, en dag i oktober 1904, startades vid samma plats, vad kunde gå fel? Jag vill därför börja med att tacka dig, Britt-Marie Thersthol! Utan din hjälp och ditt stöd hade jag inte haft en chans att ta mig hit.

Detta examensarbete omfattar 15 högskolepoäng och är början på en ny tidsepok. Nu är siktet inställt på en Master of Science inom Systems, Control and Mechatronics. Dit är det en liten bit kvar ännu.

Mer om detta examensarbete. Jag var nere i Skåne och hjälpte farsan att flytta ut ur mitt barndomshem, när jag fick ett samtal från Fredrik Andersson, där han gav mig order om att kontakta Björn Bergholm på Broccoli Engineering AB i Göteborg. Han satt inne på något intressant jobb som handlade om att utveckla en robot för att testa knapptryckningar på exempelvis bilar instrumentpaneler. Jag är ytterst tacksam för att jag fick kontakt med dig Björn Bergholm, och att du gav mig möjligheten att utföra examensarbetet hos er. Det har varit lärorikt, klurigt, spännande och väldigt kul att ta sig an en uppgift av detta slag. Tack till Tobias Olsson som varit min handledare på Broccoli. Jag passar även på att tacka alla er andra på Broccoli för att jag fått vara med här i 10 veckor.

En annan person som också ska ha ett stort tack är min handledare på Chalmers, Morgan Osbeck. Inte minst för att han fått mig att jobba dygnet runt de senaste veckorna så att jag kunnat redovisa examensarbetet, utan även ett tack för hjälpen med rapportskrivandet.

Manne Stenberg skall ha ett stort tack för att ha ställt upp som examinerare på detta arbete.

Många tack går även ut till familj och vänner för stöd och motivation! Extra stort tack till dig, lillasyster Josefin.

Marcus Thersthol
Göteborg, 2015

Sammanfattning

Denna rapport behandlar utvecklingen av en autonom robot för knapptryckning. Arbetet har utförts på Broccoli Engineering AB i Göteborg. En robot skulle utvecklas som kan trycka på knappar, dessutom skulle tryckkraftsintervall kunna ställas in så att inte knappar skulle gå sönder, samtidigt som tryckkraften skulle vara tillräcklig för att kunna generera ett knapptryck. Roboten skulle vara programmerad till att själv hitta till positionerna där knapparna finns. En senare uppgift var att möjliggöra CAN-kommunikation. Ett användargränssnitt skulle tas fram och där skulle olika parametrar gå att välja, exempelvis nedtryckningstid och tryckkraft.

Genom att använda aluminiumprofiler kunde en ställning byggas, samt en vagn som fungerade som en linjärenhet. Genom att använda optiska sensorer kunde gränser för förflyttning längs ställning och längs vagnen som har en hissfunktion. Ett linjärt ställdon är den del som utför knapptryckningen. Tryckkänsligt motstånd användes här för att säkerställa lämpliga tryckkrafter från ställdon mot testknapp.

För styrning av roboten användes en Raspberry Pi 2 Model B, samt stepper motor-HAT, PWM-HAT, ADC-HAT och en PiCAN, som är en typ av HAT som möjliggör CAN-kommunikation.

Tryckkänsliga motstånd fungerar bra till mätning av tryckkraften vid knapptryck. Det går att styra stegmotorer med Raspberry Pi 2 Model B, dock blir inte styrningen här så noggrann som önskats. När exempel med Python-kod testades så stegade motorerna mycket långsamt, dock såg det ibland ut som att de råkade backa. För detta arbete har C-kod används och en mjuk, snabb gång har använts. Till detta beteende har tidtagningsfunktion använts och sedan har avståndet vagnen rör sig mätts för att räkna ut ett värde på hur många sekunder var mm tar att köra. Utifrån detta värde kan nu vagnen köras till bestämda positioner.

Summary

This report describes the development of an autonomous robot for button pressing. While testing buttons, it's common to simulate the push by sending I/O-signals, rather than a real, physical push of the button. Therefore a robot able to push buttons was supposed to be developed. The force created by the push wasn't supposed to exceed the levels where the button would break down, at the same time, the force had to be enough to create a push. The robot was supposed to be programmed with the buttons positions to make sure it could find them while testing. A later task was to enable CAN-communication. A user-interface containing the opportunity for the user to set the different parameters, for example push-time and force applied on the button.

This work took place at Broccoli Engineering AB in Gothenburg.

A rail was built by aluminium profiles; a carriage was also built to move linearly along the rail. To realise movement along the x-, y- and z-axis, two stepper motors and one linear actuator were used. Photo Interrupters were used to limit the movement along the x- and y-axis. The linear actuator is the part that does the actual button pressing, which of course is made in the z-axis. A force sensitive resistor was used to secure button pressing in safe force intervals by the linear actuator upon the button.

A Raspberry Pi 2 Model B was used to control the robot. Stacked on the Raspberry Pi 2 Model B was also a stepper motor HAT, PWM HAT, ADC HAT and a PiCAN. The PiCAN was enabling CAN-communication. HAT stands for Hardware Attached on Top, which is exactly what it sounds like. ADC stands for Analogue-Digital-Converter and is for example used to convert the analogue signals generated by the force sensitive resistor to digital signals that the Raspberry Pi 2 Model B can handle.

The force sensitive resistors work well measuring the forces created when the button is pressed. It is possible to control stepper motors using a Raspberry Pi 2 Model B, unfortunately not as accurate as desired. For example, when Python code was tested, the stepper motor was stepping slowly, but once in a while, the rotation switched direction. This robot is programmed in C code, which let the stepper motor drive smooth, but fast. For this behaviour a timekeeping function was made and the distance the carriage travels one way was measured. From that a value for how many mm the carriage could travel per second was calculated. That value is later on used to position the carriage at desired positions.

Innehåll

1	Inledning	2
1.1	Bakgrund	2
1.2	Syfte	2
1.3	Avgränsningar	3
1.4	Precisering av frågeställningen	3
2	Problembeskrivning	4
2.1	Knapptryck	4
2.2	Förflyttning x- och y-led	4
2.3	Användargränssnitt	4
2.4	Inspelning och uppspelning av knapptryck	5
2.5	CAN-kommunikation	5
2.6	Teknisk Miljö	5
2.7	Konstruktions- och funktionsidé	5
3	Metod	6
3.1	Förstudie	6
3.2	WBS - Work Breakdown Structure	6
3.3	Arbetsgång	6
4	Val av hårdvarukomponenter	8
4.1	Aktuatorer	8
4.1.1	Stegmotor	8
4.1.2	Stegmotor förflyttning x-led	8
4.1.3	Stegmotor förflyttning y-led	9
4.1.4	Linjärt ställdon förflyttning z-led	11
4.2	Konstruktionen	12
4.2.1	MakerBeam	12
4.2.2	Actobotics (TM)	12
4.3	Val av styrenhet	13
4.3.1	Specifikationer för Raspberry Pi 2 Model B	13
4.4	Hardware Attached on Top och Breakout Boards	14
4.4.1	T-Cobbler Plus	14
4.4.2	PWM-HAT	14
4.4.3	Stepper Motor-HAT	15
4.4.4	ADC Pi Plus-HAT	16
4.4.5	PICAN CAN-Bus Board	16
4.5	Användargränssnitt	18
4.5.1	HDMI-Touch Screen	18
4.6	Sensorer och givare	19
4.6.1	Tryckkänsligt motstånd	19
4.6.2	Optisk sensor	20

5	MakerBeam-konstruktionen	21
5.1	Ställning med kuggstång	21
5.2	Linjärenhet	23
6	Koppling av komponenter	25
6.1	Uppkoppling av Raspberry Pi 2 Model B	25
6.2	Uppkoppling av AD-omvandlare	25
6.3	Uppkoppling av Stepper Motor-HAT	25
6.4	Uppkoppling av PWM-HAT	26
6.5	Koppling av tryckkänsliga motstånd	27
6.6	Koppling av optiska sensorer	27
6.7	Koppling av Firgellis linjära ställdon	27
6.8	Koppling av stegmotorer	28
6.9	Uppkoppling av PiCAN	28
7	Programmering	29
7.1	Val av programmeringsspråk	29
7.2	AD-omvandlaren, samt program för tryckkänsligt motstånd	29
7.3	PWM-HAT, rörelse av Firgellis linjära ställdon	29
7.4	Stepper Motor-HAT, rörelse av stegmotorer	29
7.5	PiCAN, mottagande av CAN-bus	30
8	Användargränssnitt	31
8.1	Val av utvecklingsmiljö	31
8.2	Grafiska användargränssnittet	31
8.3	Fysiska delarna	32
9	Test och verifiering av konstruktionen	33
10	Resultat	34
11	Slutsats	36
11.1	Resumé	36
11.2	Kritisk diskussion	36
11.2.1	Examensarbetstid	36
11.2.2	Förstudie, WBS och GANNT-schema	36
11.2.3	Genomförande	37
11.2.4	Resultat	37
11.3	Vidareutveckling	38
	Referenser	39

Beteckningar

ADC	Analogue to Digital-Converter
CAN	Controller Area Network
DSI	Display Serial Interface
GPIO	General-Purpose Input-Output
GTK	GIMP Toolkit
GUI	Graphical User Interface, grafiskt användargränssnitt
HAT	Hardware Attached on Top
HDMI	High Definition Multimedia Interface
HMI	Human-Machine Interface
I2C	Inter-Integrated Circuit
OBD-II	On-Board Diagnostics
OPIC	Optical Integrated Circuit ('OPIC' is a TradeMark of the SHARP Corporation)
PCB	Printed Circuit Board, mönsterkort. Kretskort utan komponenter monterade.
PWM	Pulse-Width Modulation, pulsbreddsmodulering
RAD	Rapid Application Development
RAM	Random Access Memory
SCL	Serial Clock Line
SDA	Serial Data Line
SoC	System-on-Chip
SPS	Samples Per Second
WBS	Work Breakdown Structure
XML	Extensible Markup Language

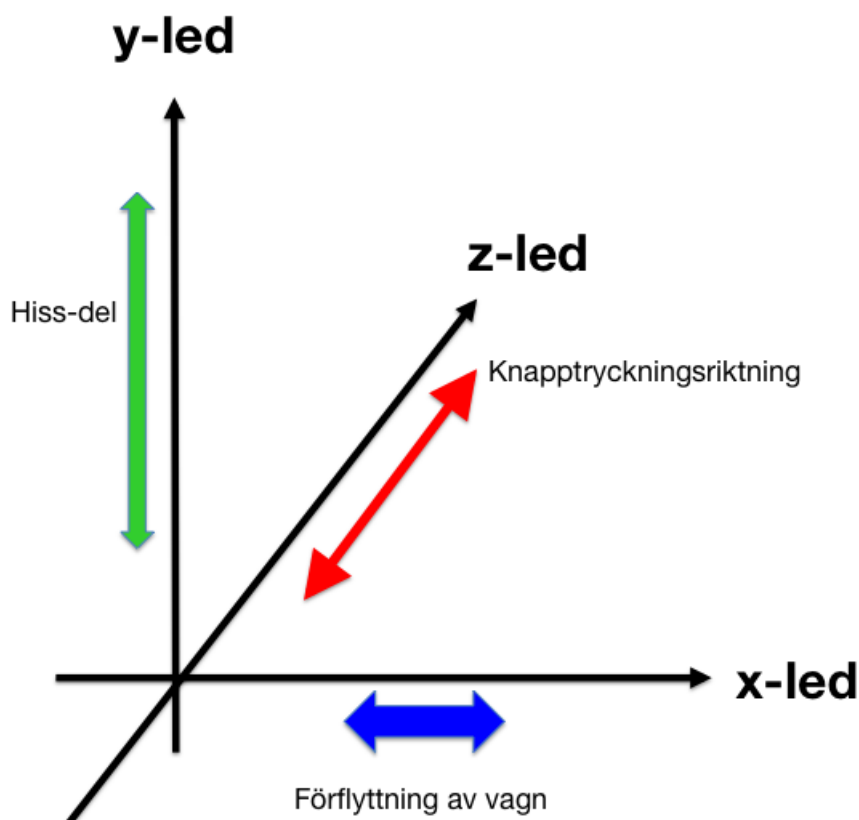
1 Inledning

1.1 Bakgrund

Inom fordonsbranschen är det svårt och tidskrävande att funktionstesta alla knappar i ett fordon, exempelvis knapparna på en instrumentpanel och se om dessa fungerar som de ska. Vanligen simuleras knapptryckningen av en I/O-signal till styrsystemet, som då ersätter det fysiska tryckandet på respektive knapp. Detta gör att det blir svårt vid tvister att bevisa att funktionen fungerar som det är tänkt, då det fysiskt inte har skett några egentliga knapptryckningar på varken en eller flera knappar samtidigt. Detta är speciellt viktigt då ett företag testat något nytt koncept och således behöver statistik för ett visst antal tryckningar för att säkerställa att en funktion fungerar på ett felfritt sätt.

1.2 Syfte

En anordning som kan trycka på fysiska knappar, samt på touch-paneler. Detta för att testa att knappar och skärmar fungerar som de ska. Knapptryck skall kunna ske med krafter i intervall som garanterar att knapptrycket registreras och att knappar och paneler inte går sönder.



Figur 1: Definition av koordinatsystemet för detta problem(Författarens bild)

Ett första delmål är att kunna trycka på en knapp med hjälp av någon form av aktuator. Ett annat delmål är att möjliggöra rörelser i x- och y-led. Vilka knappar som

skall testas skall gå att ställa genom ett användargränssnitt. Det finns även ett delmål att möjliggöra inspelning av knapptryck och sedan låta anordningen spela upp detta. Därefter skall CAN-kommunikation möjliggöras mellan anordning och testobjekt.

1.3 Avgränsningar

Det finns inga begränsningar på tiden det tar för roboten att flytta sig i x- och y-led.

Vid knapptryckning skall inte funktioner kopplade till knappen kontrolleras, exempelvis om en knapp har en liten lampa som skall lysa då knappen aktiverats, så behöver ingen återkoppling ske att just den lampan började lysa. Det viktiga är att knappen har tryckts och att en signal har genererats.

Vridfunktionen på knappar skall inte testas. Exempelvis ratten för att höja och sänka volymen i bilens ljudanläggning.

1.4 Precisering av frågeställningen

Hur kan knapptryckning genomföras?

Rörelse i x- och y-led kan ske på flera sätt, vilket passar bäst för denna anordning?

Det finns tre delmål.

Det första är att anordningen skall tilldelas olika ställbara parametrar, som skall kunna ställa in tryck-, -hastighet, -känslighet, -vinkel, nedtryckningstid. Dessa parametrar skall kunna ställas in via en dator där det även skall gå att välja vilka knappar som önskas bli nedtryckta, detta kan göras via ett användargränssnitt.

Hur kan tryckkraft kännas av och registreras?

Vilka sensorer finns för att klara av uppgiften?

Hur kan hastigheten ställas?

Vilket tryckkraftsintervall skall användas?

Det andra delmålet innebär implementering av en funktion på anordningen som möjliggör "inspelning" av knapptryckningar med hjälp av sensorer som läser in data som kan spelas upp av anordningen. Detta möjliggör så verkliga knapptryck som möjligt.

Hur kan inspelningen fungera?

Kan tryckkänsligt motstånd användas för att spela in hur hårt knappen tryckts vid inspelning?

Det tredje delmålet handlar om att möjliggöra CAN-kommunikation, där anordningen kan ta emot signalerna som genereras från knapptrycket.

Hur kan värdena från knapptrycken registreras?

2 Problembeskrivning

I detta kapitel presenteras de olika delproblem, delmål och krav som finns.

En autonom robot för knapptryckningstester skall utvecklas och byggas. Här skall roboten kunna utföra fysiska knapptryck, vanligen simuleras endast knapptryck genom att sända I/O-signal om att knappen har blivit tryckt, vid funktionstester. Positionerna på knapparna skall programmeras till roboten och denne skall sedan själv hitta till dem. Genom ett gränssnitt skall användare kunna ställa parametrar på exempelvis nedtryckningstid och tryckkraft.

2.1 Knapptryck

Det huvudsakliga problemet är att lösa knapptryckningar för test av bland annat instrumentpaneler i bilar. Knapptryckningarna är valda som förflyttning i z-led. Både knapptryckningar på fysiska knappar och tryck på touch-skärmar skall kunna ske.

Här är ett krav att trycket skall ske så pass snabbt att det efterliknar en människas hastighet och tid det tar för ett knapptryck. Krav existerar dessutom att knappar och skärmar ej får tryckas sönder. Dessutom finns det krav på att tryckkraft skall kunna ställas in av användaren och på så sätt även kunna registreras av anordningen. Förtydligande är att om värdet 3 N är önskat så ska knappen tryckas ned tills anordningen registrerar 3 N. Då även nedtryckningstid och hastigheten skall kunna ställas in, så räcker det inte att anordningen kan registrera tryckkrafter, utan programmering behöver även ske på sådant vis att knapptryck sker på ett önskat sätt.

För att en dator skall kunna behandla signaler ifrån, exempelvis tryckkänsliga motstånd så krävs AD-omvandling.

2.2 Förflyttning x- och y-led

För att möjliggöra testtryck på fler än en knapp, så behöver förflyttning på åtminstone en axel till finnas. I detta fall finns det knappar i olika nivåer, både på höjd och bredd, så förflyttning behöver ske i både x- och y-led. Vad gäller förflyttning i x- och y-led så finns det inga krav på hur fort det behöver gå.

Det finns önskemål om linjärförflyttning längs en bana där exempelvis kuggstång används. I x-led önskas en lösning där vagnen kan stanna då den når en viss gräns för att säkerställa att den inte rullar av banan på grund av att banan skulle ta slut.

2.3 Användargränssnitt

För att användaren på ett enkelt sätt skall kunna beställa vilka knappar som skall testas, behövs ett användargränssnitt. Det finns flera sätt att lösa detta problem, bland annat kan ett GUI skapas. Där kan exempelvis inmatningsalternativen visas på en display och ett tangentbord användas för att mata in önskade värden.

Touch-Screen, med ett tangentbord i displayen, är också en möjlig lösning. Angående användargränssnittet är det inga speciella krav, förutom möjligheten att ställa parametrarna.

2.4 Inspelning och uppspelning av knapptryck

För att få ett så verkligt värde på tryckkraftsparametern, kommer möjlighet att skapas för inspelning av tryckkraft. Genom att låta användaren testtrycka på en knapp antingen en gång eller flera gånger där sedan lämpliga värden väljes ut för att sparas undan och sedan kunna användas när knapptest körs. Lösningen på detta problem kan sedan implementeras i användargränssnittet där en möjlighet att själv spela in tryckkraft kan ligga med som ett extra val.

2.5 CAN-kommunikation

Om tid finns skall CAN-kommunikation implementeras. Det är inga krav att CAN-kommunikation skall hinnas med.

2.6 Teknisk Miljö

Roboten kommer att testa knapparna på paneler i Volvo-bilar. Den kommer efter initiering av valda testknappar att vara självgående, tills testet är klart. Då den testar knapparna i inomhusmiljö kommer den inte utsättas för extrema temperaturer, väderomskiftningar eller andra påfrestande miljöer. Den har därmed inget behov av en massa skydd mot fukt, damm eller dylikt.

2.7 Konstruktions- och funktionsidé

De konstruktions- och funktionsidéer som kommit fram skall förklaras i detta avsnitt. Funktionsmässigt så skall knappar kunna tryckas och detta samtidigt som någon form av tryckkraft registreras för att inte trycka för hårt eller löst. Själva knapptrycket kan konstruktionsmässigt ske med ett linjärt ställdon. Tryckkraft kan registreras med hjälp av en tryckkänslig resistor. Ifall en dator skall kunna ta emot den analoga signal som ett sådant motstånd avger, så måste en AD-omvandlare användas för att göra om den analoga signalen till digital.

En annan funktionalitet är att flera knappar skall kunna tryckas och testas. Då måste ställdonet flyttas runt i x- och y-led, utgående ifrån att ställdonet rör sig i z-led. Konstruktionsmässigt kan denna idé lösas genom att bygga upp en ställning med en bana för en linjärenhet att röra sig på. Linjärenheten måste då konstrueras för att passa denna bana. Dessutom bör någon form av kuggstång användas på banan för att kunna använda sig av funktionaliteten att röra sig längs banan. Funktionsmässigt kan någon form av motor användas och med hjälp av kuggdrev som förflyttar linjärenheten längs banan.

För att linjärenheten skall hålla sig kvar och inte åka av banan bör banan utrustas med sensorer som varnar om en viss punkt passeras. Funktionaliteten kan lösas med hjälp av programmering av sensorerna.

Förflyttning i y-led kan konstruktionsmässigt lösas med hjälp av att bygga linjärenheten hög och montera dit glidskenor för att förflyttningen skall ske stabilt. Funktionsmässigt kan förflyttningen lösas med en stegmotor och ledskruv. Som stopp kan även här någon sensor användas och funktion programmeras.

3 Metod

I detta kapitel kommer det tänkta tillvägagångssättet att redovisas. Till att börja med kommer en förstudie att ske för att samla in tillräckligt med information för att genomföra uppgiften. Problemet som skall lösas kommer därefter att delas upp i flera mindre delproblem och här skall en WBS användas för att bryta ner ursprungsproblemet till arbetsuppgifter. Därefter skall ett Gantt-schema skapas utifrån vad som framkommit i WBS.

Inköp av komponenter skall ske så tidigt som möjligt, för att inte kunna bli allt för påverkad av sena leveranser eller långa leveranstider. Det är dessutom tänkt att komponenter monteras allt efter de kommer och möjlig programmering görs i takt med att komponenter är kopplade. Så fort något framsteg eller annat viktigt för rapporten sker, skall detta antecknas och sedan skrivas in i rapporten.

3.1 Förstudie

De första veckorna av arbetstiden skall användas till förstudier. Dessa innebär bland annat att litterära studier, för att finna information om vilka komponenter som kan passa till olika lösningar, skall utföras. Förstudier skall även ske framför datorskärm och kommer då innebära sökande efter produkter och komponenter som tänkbart kan fungera till en lösning. För anteckningar då intressanta komponenter finnes. Bokmärken över dessa kan vara ett bra sätt att strukturerat samla dem.

3.2 WBS - Work Breakdown Structure

Samtidigt som förstudier genomförs skall parallellt en WBS utföras över problemet som skall lösas. De går ut på att bryta ner huvudmålet, i det här fallet en autonom robot för knapptryckning, i delprojekt. Delprojekten skall i sin tur brytas ner till uppgifter som i sin tur skall brytas ner till deluppgifter och slutligen bryts dessa ner till arbetsuppgifter. Ett förtydligande är att en WBS inte är någon form av planering, utan är till för att skapa en fullständig överblick av vilka arbetsuppgifter och insatser som kommer att krävas för att målet skall kunna uppnås [1].

3.3 Arbetsgång

När WBS är klar skall arbetsuppgifterna utifrån denna sammanställas i ett Gantt-schema för att strukturera upp arbetet. Detta schema skall följas för att underlätta och hålla struktur.

Ett nästa skede i arbetsgången är att börja beställa hem komponenter och produkter som kan tänkas behövas till att utföra arbetet. Det är viktigt att strukturera upp vilka delar som tillhör vilka dellösningar och på så sätt sortera delar i lådor efter detta.

Så fort komponenter börjar anlända skall arbete med att montera och koppla dessa ske. Har tillräckligt med delar blivit monterade för att möjliggöra tester och programmering, så skall programmering också påbörjas i takt med att dellösningar monterats ihop. När framsteg eller annat för rapporten intressanta händelser sker, skall dessa antecknas och dokumenteras till rapporten.

I slutskedet av arbetet skall rapporten sammanställas och slutföras. Här skall LaTeX användas.

4 Val av hårdvarukomponenter

I detta avsnitt beskrivs vilka hårdvarukomponenter som valts till roboten och lite fakta om dem.

4.1 Aktuatorer

4.1.1 Stegmotor

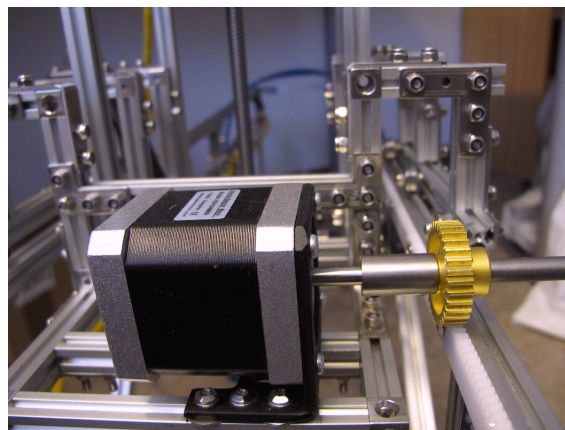
Det finns flera typer av stegmotorer, bland annat permanentmagnetiska och sådana med mjukjärnsrotor och variabel reluktans. Här kommer i huvudsak hybrida stegmotorer behandlas. En 2-fas hybrid stegmotor använder sig både av permanentmagnet och tandade rotor av mjukjärn. De kan ha två statorkor där tänderna är förskjutna med en halv tand. Genom att ändra riktningen på strömmen kan man ställa statorn till att vara nord- eller sydpol.

Stegmotorer styrs digitalt, steg för steg, vilket leder till att de kan få en väldigt precis-rad positionering utan att återkoppling behöver tillämpas. De har även högt vridmoment vid låga hastigheter, och när den är strömmatad och stillastående självlåser motorn sig tack vare det höga momentet. En nackdel som gäller 2-fas-motorer är att de kan leda till oscillationsproblem då de körs nära eller vid självsvängningsfrekvensen [2] [3].

4.1.2 Stegmotor förflyttning x-led

För att möjliggöra förflyttning i x-led används en stegmotor med 400 steg per varv. Den är av hybridtyp och bipolär, samt med standardmått NEMA 17. Axeln har diametern 5 mm.

Anledningen till att denna stegmotor valdes var bland annat att den inte är speciellt stor, den tar många steg per varv, alltså potential till att vara väldigt noggrann. Att välja en stegmotor med standardmått, så som NEMA 17 leder till ökat utbud av fästen som passar motorn och montering blir således enklare.



Figur 2: Stegmotor, förflyttning x-led (Författarens bild)

Specifikationer från tillverkarens produktblad, samt extra info från dess hemsida [4].

Tillverkare:

Wantai Motor

Modellnamn: 42BYGHM809

Generella specifikationer:

Motorns längd: 48 mm

Motorns vikt: 340 g

Elektriska specifikationer:

Stegvinkel: 0.9°

Antal faser: 2 st

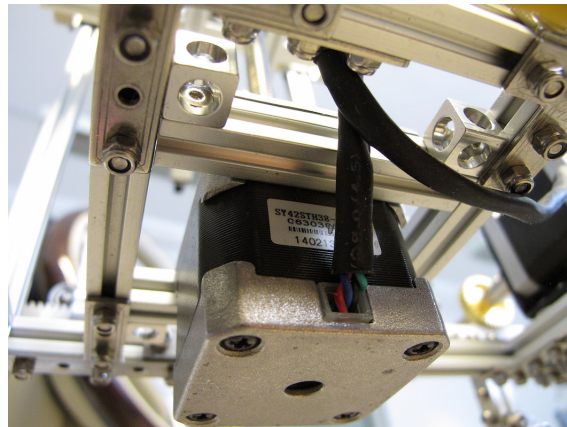
Märkspänning: 3.06 V

Märkström: 1.7 A per fas

4.1.3 Stegmotor förflyttning y-led

Vid förflyttning i y-led används en stegmotor med 200 steg per varv och som har en, redan på axeln, förmonterad ledskruv av rostfritt stål. Stighöjden per steg är 40 (mikrometer) och standardmåttet på motorn är NEMA 17. Stegmotortypen är en bipolär, hybrid stegmotor.

Denna stegmotor valdes för att den har en ledskruv redan förmonterad, storleken på motorn är bra och stighöjden per varv gör denna väldigt noggrann. Återigen används standarden NEMA 17, till vilken fästen från MakerBeam finns att köpa.



Figur 3: Stegmotor förflyttning y-led (Författarens bild)

Från tillverkarens specifikationsblad samt dess hemsida [5].

Tillverkare: Changzhou Songyang Machinery Electronics Co. Ltd
Modellnamn: SY42STH38-1684A

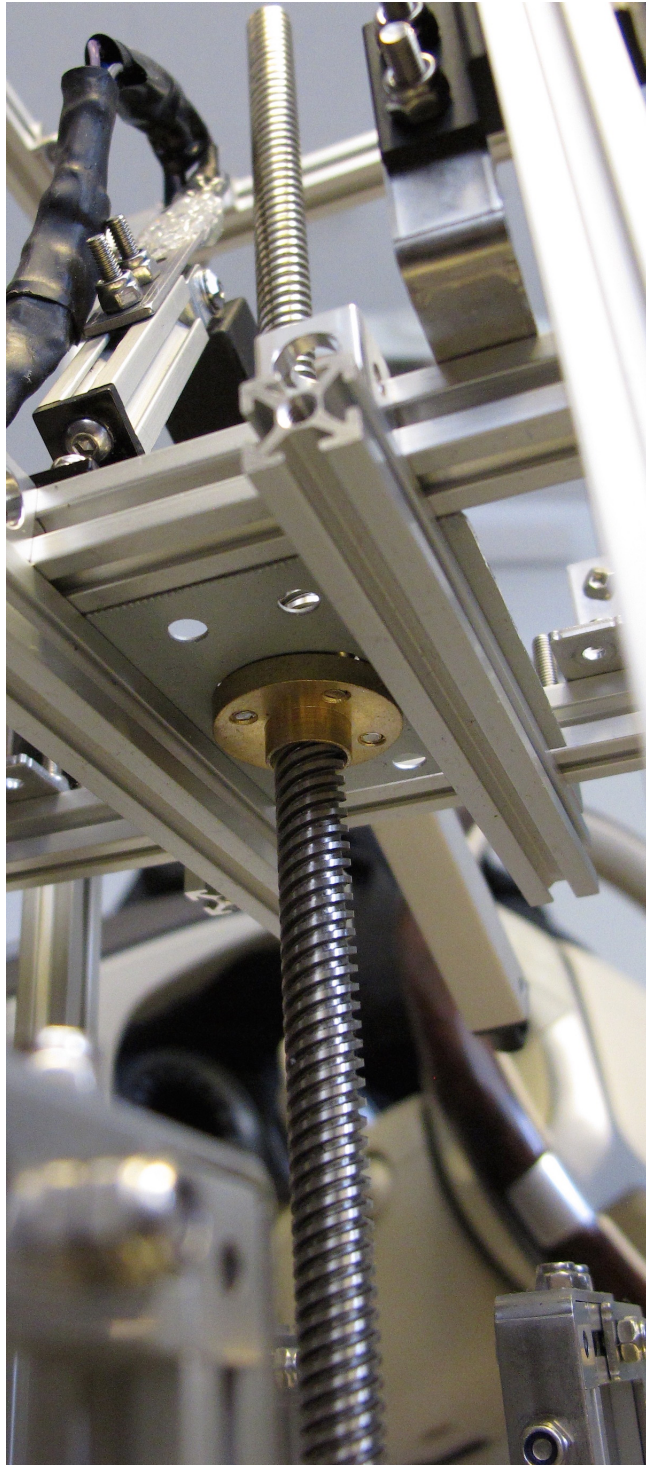
Generella specifikationer:

Maximal radiell kraft: 28 N (20mm från flänsen)

Maximal axiell kraft: 10 N

Vikt totalt: 385 g

Längd ledskruv: 280 mm



Figur 4: Ledskruv, monterad på stegmotor förflyttning y-led (Författarens bild)

Elektriska specifikationer:

Stegvinkel:	1.8°
Antal faser:	2st
Märkspänning:	2.8 V

4.1.4 Linjärt ställdon förflyttning z-led

Linjära ställdon kan användas i bland annat robotar, industriella automationslösningar och för positionering av antennreflektorer. De kan bestå av antingen AC- eller DC-motor som exempelvis kan driva en kulskruv för att skjuta ut ställdonets arm. Fircellis Miniaturställdon använder en DC-motor [6].

Den modell som valts för att kunna utföra knapptrycket är av I-modell, vilket innebär integrerad styrning. Valet gjordes för att detta ställdon bland annat har PWM-styrning, återkoppling samt RC-Servo-läge. Slaglängden är vald till 100 [mm] för att anordningen skall komma åt även knappar som sitter något längre bort. Nedan förklaras några olika funktioner som kan väljas genom vilken kabel som används. Den integrerade kretsen i ställdonet märker själv av vilken kabel som är kopplad och således vilket läge den ska använda. Det finns sex stycken kablar med olika färger. Vi kommer att använda fyra av dessa. Röd för +6 V, svart för jord, vit kabel för RC-Servo-styrning, samt den lila kabeln för återkopplingsfunktionen för att bestämma känna av positionering.

Här beskrivs de olika lägena som finns förprogrammerade på microcontrollern i ställdonet. Denna programvara går inte att programmera om av användaren. Det finns fyra möjliga gränssnittsmetoder att använda. 0-5 V, 4-20 mA, RC-Servo och PWM.

0-5 V- läget tillåter styrning med endast ett batteri och en potentiometer för att bestämma positionen för ställdonet.

4-20 mA-läget kan användas vid PLC-styrning och positionen av ställdonet bestäms med strömmar från 4-20 mA.

RC-Servo-läget använder en +5 V-pulsbreddssignal på kontakt 4 som en insignal för önskad position, sedan används 1.0 ms pulskommando för att kalla ställdonet till minimumläge och 2.0 ms pulskommando talar om ställdonet att gå till maximumläge.

PWM-läget styrs genom en enda digital utgång från någon annan microcontroller än den internt inbyggda i ställdonet. Önskad position genereras genom en arbetscykel av en +5 V, 1 kHz fyrkantsvåg kopplad till kontakt 2. Här bestäms utslaget genom att välja ett visst antal procent för arbetscykeln, där procenten motsvarar lika många procent av slaglängden.

Anledningen till att RC-Servo-läget valdes är för att det var lättast att välja position på det viset, då behöver bara ett värde väljas mellan 1,0 och 2,0 [ms] och motsvarande procent där, gäller för positionen i procent av slaglängden.



Figur 5: Fircellis Linjära Ställdon(Författarens bild)

Specifikationer från Fircelli Technologies Inc. datablad för produkten [7].

Tillverkare:	Firgelli Technologies Inc.
Modellnamn:	L12-100-50-06-I
Max effekt vid belastningsfall:	12 N vid 11 mm/s
Maximal effektivitet vid belastningsfall:	6 N vid 16 mm/s
Maximal hastighet:	23 mm/s (utan last)
Slaglängdsval:	100 mm (används för vald modell)
Vikt:	56 g
Positionsnoggrannhet:	0.3 mm
Livslängd:	1000 timmar vid arbetscykel på 20 %
Hållström:	450 mA vid 5V och 6V

4.2 Konstruktionen

Konstruktionen är uppbyggd huvudsakligen på MakerBeams aluminiumprofiler och sedan har några Actobotics-komponenter använts till att möjliggöra kraftöverföring från motor till rörelse längs konstruktionen.

4.2.1 MakerBeam

MakerBeam tillverkar aluminiumprofiler med så kallad T-skåra. Det startade som ett projekt år 2009 på en sida som heter kickstarter.com. En av de som köpte en sats med profiler då, blev så fäst att han gick i tankar om att starta försäljning av dem. Dock hann hans fru före och de började sälja dessa aluminiumprofiler från Nederländerna. De har funnits där sedan 2011.

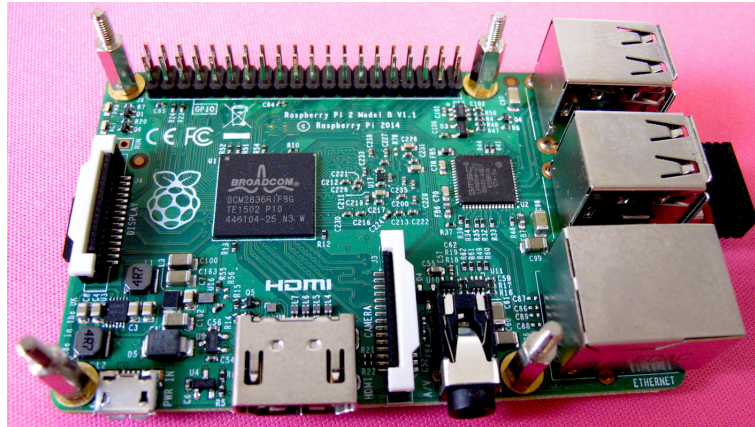
Anledningen till att jag valde att använda deras aluminiumprofiler var för att de tillhandahåller några av marknadens minsta dimensioner på dessa, de är endast 10x10 [mm] i tvärsnitt. Dessa profiler finns i längder från 40 mm till 1500 mm. Det finns flera typer av brickor i olika former, NEMA 17-fästen för stegmotorer, servomotorfästen, kullager, glidskenor och mycket annat för byggen av mindre projekt. Det som gjorde att jag fick upp ögonen för deras produkter var att jag fann en svensk återförsäljare av deras produkter, dock kunde denna återförsäljare inte sända iväg produkterna med något fraktbolag på grund av vikten. Detta var ett mindre problem då ett större utbud fanns i Nederländerna och i stort sätt alla byggdelar som behövs för att bygga anordningen kunde köpas från MakerBeam. En annan anledning till valet av MakerBeam är den låga vikten, endast 123 gram per meter. Deras specials kruvar har kvadratiska huvud gjorda för att precis passa i skårorna [8].

4.2.2 Actobotics (TM)

Actobotics(TM) är ett registrerat varumärke och de tillverkar diverse komponenter för robottillverkning. Deras delar är utvecklade för att kunna passa flera olika standarder genom att deras U-formade aluminiumplåtar har många hål. De delar jag valde från Actobotics var två kuggstänger i plast, med måttet 32P. Längden per kuggstång är ca 30,5 cm.

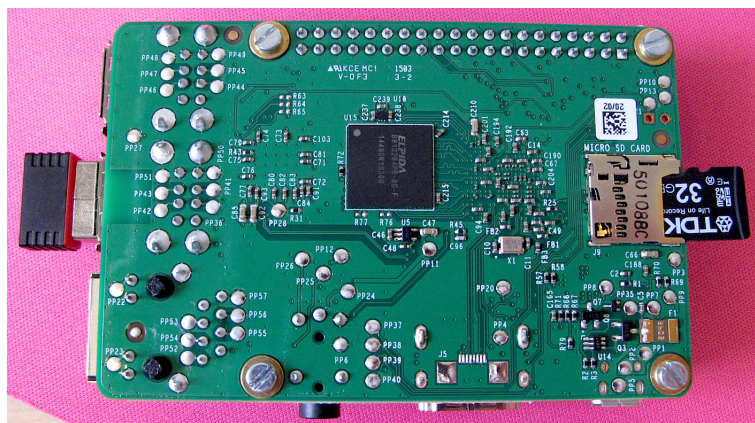
Från deras sortiment valdes även en D-formad 1/4"-axel, en koppling från 5 mm till 1/4" samt ett kuggdrev med stoppskruv och samma kuggavstånd som stängen. Kopplingen mellan 5 mm och 1/4" behövdes då stegmotorns axel endast var 5 mm [9].

4.3 Val av styrenhet



Figur 6: Raspberry Pi 2 Model B, ovansida(Författarens bild)

Raspberry Pi 2 Model B valdes som styrenhet då den för det första är billig. Den nya modellen är kraftfull för priset. Det finns en hel del extra komponenter att köpa till Raspberry Pi, bland annat färdiga kort för styrning av elmotorer, PWM-signaler, AD-omvandling och CAN-kommunikation. Detta är huvudanledningen till att en Raspberry Pi valdes. Då processorn är fyrkärning finns möjligheten att köra flera processer samtidigt, på olika kärnor.



Figur 7: Raspberry Pi 2 Model B, undersida(Författarens bild)

4.3.1 Specifikationer för Raspberry Pi 2 Model B

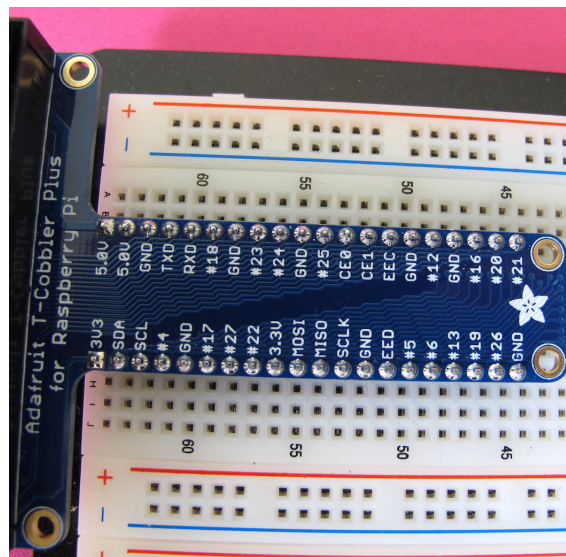
Följande specifikationer är tagna från Raspberry Pi 2, Model B's datablad [10].

Chip:	Broadcom BCM2836 SoC
Core architecture:	Quad-core ARM Cortex-A7
CPU:	900 MHz
Memory:	1 GB LPDDR2
Power:	Micro USB socket 5 V, 2 A.
Ethernet:	10/100 BaseT Ethernet socket.
Video Output:	HDMI (rev 1.3 and 1.4).
Audio Output:	3.5 mm jack, HDMI
USB:	4 * USB 2.0 Connector
GPIO Connector:	40-pin 2.54 mm (100mll) expansion header: 2*20 strip. Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines.

4.4 Hardware Attached on Top och Breakout Boards

4.4.1 T-Cobbler Plus

T-Cobbler Plus är en breakout board för Raspberry Pi 2 Model B och den förenklar utvecklingsarbetet genom att den lätt kan kopplas till en bread-board. Den levereras med en 40-pins bandkabel som kopplas direkt på Raspberry Pins 40 GPIO-stift [11].

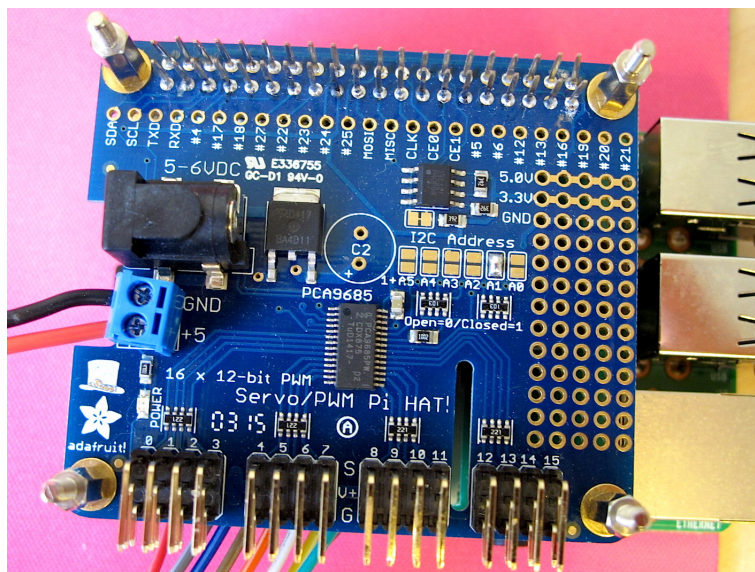


Figur 8: T-Cobbler Plus, monterad på bread-board (Författarens bild)

4.4.2 PWM-HAT

Adafruit Industries har tillverkat en 16-kanalers, 12-bitars PWM/Servo-HAT, för att kunna hjälpa Raspberry Pi med bland annat styrning av likströmsmotorer och fristående pulsbreddsmodulering. Detta för att Raspberry Pi inte är speciellt bra på att kontrollera komponenter som ställer höga krav på specifika och återupprepande pulser. Var PWM-HAT klarar av att kontrollera 16 servon och det går dessutom att stapla upp till 62

sådana HAT:ar på varandra. Styrningen går via 2 stycken I2C-pins från Raspberry Pi. PWM-HAT:en klarar av att köra pulsbreddsmodulering upp till 1,6 kHz med 12 bitars upplösning.



Figur 9: Adafruits PWM-HAT (Författarens bild)

Strömförsörjning sker dels via VCC från Raspberry Pi, där 3,3 V används för att driva PWM-chippet samt för att bestämma I2C-adressen. Det finns även möjlighet att koppla in annan strömkälla, antingen genom den blå kopplingslisten eller via likströmsjacket.

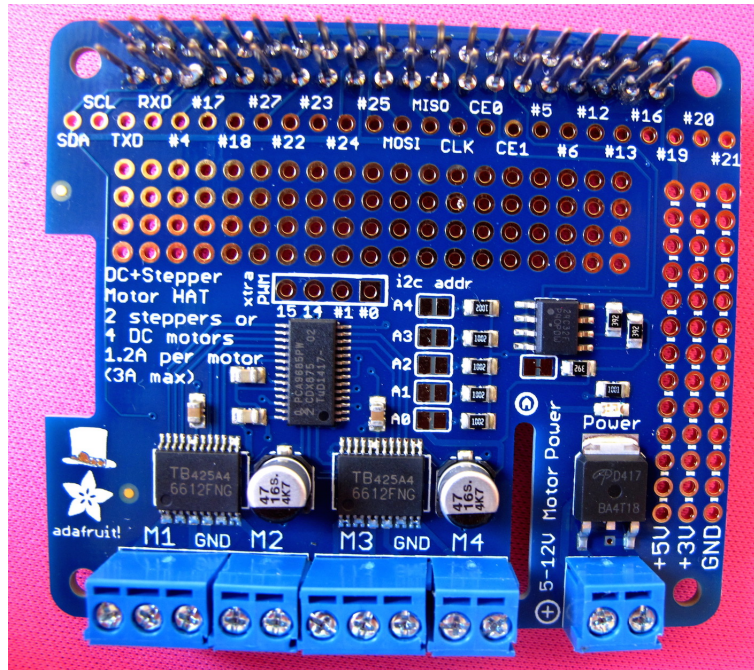
Anledningen till att denna PWM-HAT valdes var för att kunna driva det linjära ställdonet från Firgelli. Dessutom tillhandahåller detta kortet ett antal extra 5 V-, 3,3 V- och GND-källor, som kan utnyttjas till att driva optiska sensorer eller tryckkänsliga motstånd [12] [13].

4.4.3 Stepper Motor-HAT

Adafruit Industries har utvecklat denna stepper motor-HAT, som klarar av att driva två stycken stegmotorer, eller fyra stycken likströmsmotorer. På kortet är även en PCA9685 monterat och den klarar av att hantera hastighet och riktning på stegmotorerna och har en inbyggd oscillator för att styra pulsbreddsmoduleringen. Kortet är stackningsbart och använder I2C för att styra motorerna. Det går att montera 32 stycken Stepper Motor HAT:ar på varandra.

Var stegmotor drivs via en på kortet monterad TB6612FNG, som är en integrerad drivenhet för dubbla likströmsmotorer. Den klarar av max-spänning på 15 V, utströmmar på 1,2 A medelström eller 3,2 A toppström. De två TB6612FNG som är monterade på kortet agerar som 4 stycken H-bryggor. De har även ett skydd emot överhettning, som gör att den automatiskt slår ifrån om en viss temperatur överstigs.

Adafruit Industries Stepper Motor-HAT valdes då den för det första klarar av att driva två stycken stegmotorer, och för det andra att det är stackningsbart och kan köras via I2C [14] [13].



Figur 10: Adafruits Stepper Motor-HAT (Författarens bild)

4.4.4 ADC Pi Plus-HAT

Denna AD-omvandlare har valts för att kunna koppla in de tryckkänsliga motstånden och omvandla den analoga signalen till ett digitalt värde. Den kommer även att användas för återkoppling av position för Firgelli L12.

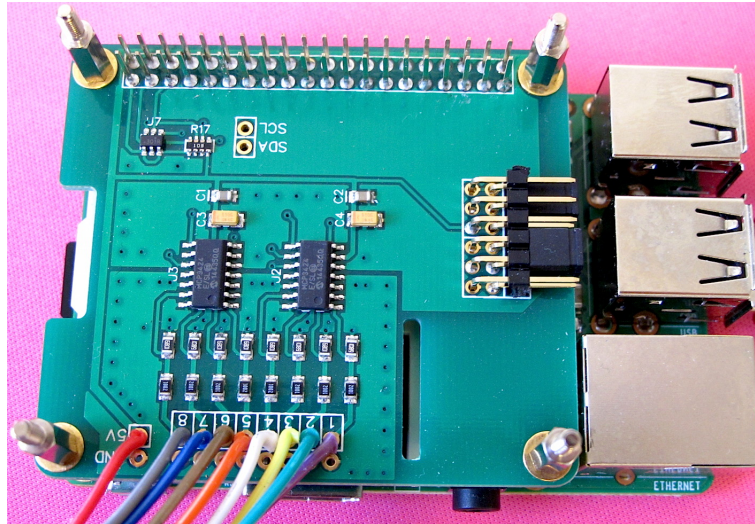
AB Electronics UK tillhandahåller en asymmetrisk analog till digital-omvandlare som fungerar med Raspberry Pi 2 Model B. Den kontrolleras med hjälp av I2C portarna och det går att koppla upp till fyra ADC Pi Plus på varandra. I2C-adresser går att välja med hjälp av jumpers på kortet. Referensspänningen på kortet är 2,048 V, med -V-benen är kopplade till jord. En spänningsdelare monterad på kortet flyttar inmatningsspänningsintervallet till mer lämpade 0 – 5,06 V. Två MCP3424 A/D-omvandlare är monterade på kortet, dessa ger tillsammans åtta analoga ingångar [15][16][17].

Inmatningsvärden och specifikationer från AB Electronics UK:s hemsida.

Vdd (5V-pinnen på I2C-bussen):	5.0 V
All AD in- och utmatning:	VSS -0,4 V till VDD +0,4 V
Ström vid inmatningspinnarna:	$\pm 2 \text{ mA}$
I2C SDA/SCL spänning:	5.0 V
I2C-portsström:	100 mA

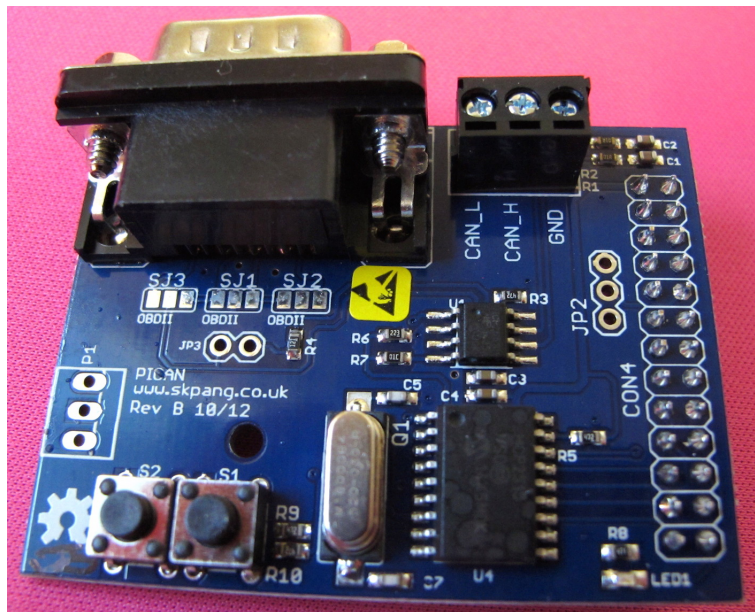
4.4.5 PICAN CAN-Bus Board

SK Pang electronics i Storbritannien tillhandahåller ett PiCAN-kort som möjliggör CAN-bus användning till Raspberry Pi. På kortet har det monterats en MCP2515 CAN-controller och en MCP2551 CAN-receiver, båda från Microchip. Koppling kan sedan ske med 3-vägars skruvterminal eller DB9 [18].



Figur 11: AD-omvandlaren från AB Electronics(Författarens bild)

Anledningen att denna komponent valdes var för att CAN-kommunikation ska möjliggöras. Den ska även fungera bra med Raspberry Pi.



Figur 12: PiCAN (Författarens bild)

Följande funktioner är tagna från SK Pang electronics hemsida.

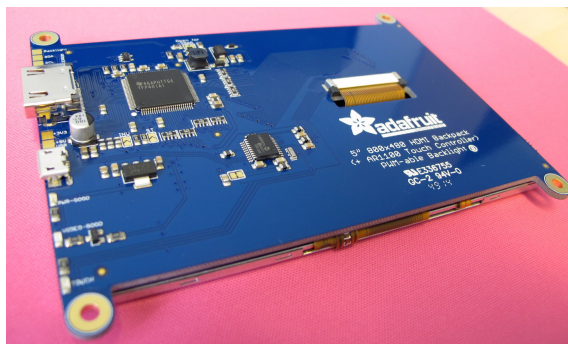
- CAN v2.0B vid 1 Mb/s
- Höghastighets SPI-gränssnitt (10 MHz)
- CAN-koppling via standard 9-vägars sub-D-anslutning eller med skruvterminal.
- Kan användas med OBDII-kabel.
- LED indikatorer.
- Två tryckbara användarknappar.

4.5 Användargränssnitt

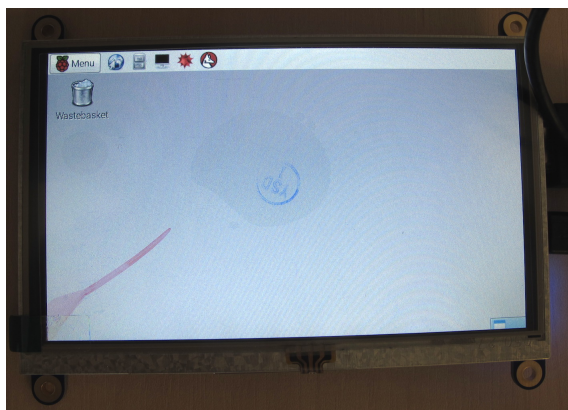
Komponenterna som är tänkta att användas till själva användargränssnittet är bland annat en touch-skärm från Adafruit som står att läsa om nedan. Denna är tänkt att sitta på en frontplåt med måtten 360 * 150 [mm] som egentligen är ämnad för montering på två universalchassin. Denna frontplåt är tänkt att fräsas upp för att skärmen skall passa och sedan göra hål för montering av flera olika knappar, med bland annat funktion för inspelning av tryckkraft.

4.5.1 HDMI-Touch Screen

Adafruit har utvecklat en 5"-touch-skärm som dessutom går att koppla till Raspberry Pi via HDMI och USB. Upplösningen är 800 * 480 pixlar, RGB. Varianten som kommer att användas till användargränssnittet har en AR1100 USB, resistiv touch-skärmsdrivenhet från Microchip. Denna har 10 bitars maximal touch-upplösning. När skärmen är inkopplad med USB, så dyker touch-delen upp som en musenhet [19].



Figur 13: Baksida av LCD-Touch-skärmen från Adafruit (Författarens bild)



Figur 14: Framsida, LCD-Touch-skärmen från Adafruit (Författarens bild)

Anledningen till att denna skärm valdes till användargränssnittet är att den är enkel att koppla in och går i princip att använda direkt. Det enda som krävs att göra innan användning första gången är att själv skala upplösningen för att passa Raspberry Pi.

4.6 Sensorer och givare

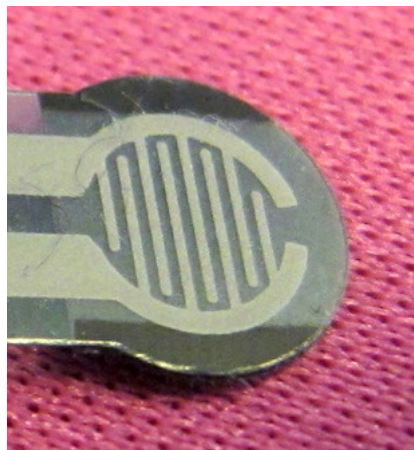
4.6.1 Tryckkänsligt motstånd

Interlink Electronics tillverkar ett antal tryckkänsliga motstånd, i olika storlek och form, för att användas inom olika HMI-lösningar. Motståndet har två pins som är kopplade till tryckkänsliga material som ligger inklämt mellan två lager av tjock polymerfilm. Då ytan på motståndet utsätts för en kraft, minskar resistansen, vilket lämnar möjligheter att bestämma ungefärliga tryckkrafter genom att omvandla uppmätt spänning till kraft.



Figur 15: Interlink Electronics FSR400, tryckkänsligt motstånd (Författarens bild)

Anledningen att just detta tryckkänsliga motstånd valts, är för att kraftkänslighetsintervallet passar bra för låga krafter. Det är litet och blir således lättare att placera i en mindre anordning. Lastceller och liknande är anpassade för mycket större krafter och passar inte till detta syfte. Hade återförsäljaren i Sverige haft samma serie, fast modellen som endast har den tryckkänsliga delen, utan de långa inplastade ledarna, så hade den typen varit att föredra.



Figur 16: Denna del utgör det tryckkänsliga partiet av motståndet. (Författarens bild)

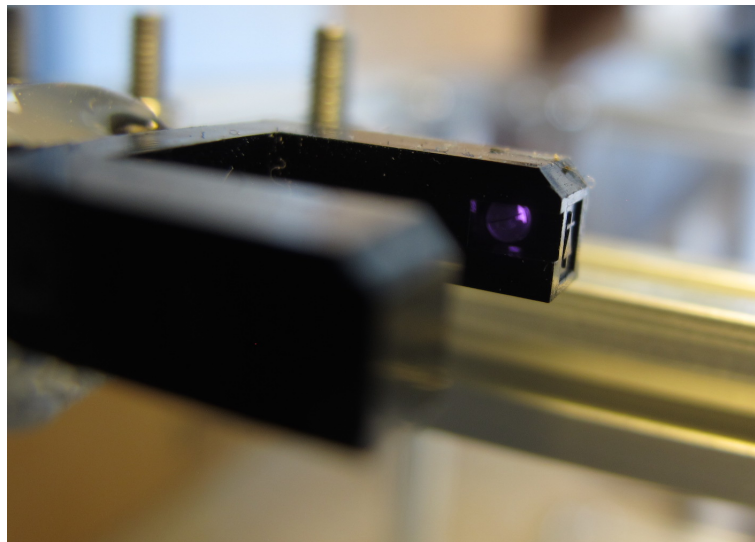
Denna typ av motstånd är väldigt tunna, trots det tåliga och tål upp till 10 miljoner mätningar. Följande specifikationer är tagna från företagets produktblad [20].

Produktnamn:	FSR (R) Model 400
Tillverkare:	Interlink Electronics
Aktiv yta:	Diameter på 5.08 mm

Nominell tjocklek:	0.30 mm
Aktiveringskraft (minsta):	0.2 N
Kraftkänslighetsintervall:	0.2 – 20 N
Kraftupplösning:	Konstant (analogt)
Resistans då oaktiverad:	Mer än 10 M Ω
Motståndets stigtid:	Mindre än 3 ms

4.6.2 Optisk sensor

Den optiska sensorn består av en ljuskälla som lyser på en detektor, bryts strålen slår den optiska sensorn om från 1 till 0. De kan användas för att upptäcka objekt. Anledningen till att denna är vald till denna anordning är att priset är lågt och den har potential att klara av det som önskas av den. Att upptäcka maximum och minimum positioner för stegmotorerna.



Figur 17: Optisk sensor, påslagen (Författarens bild)

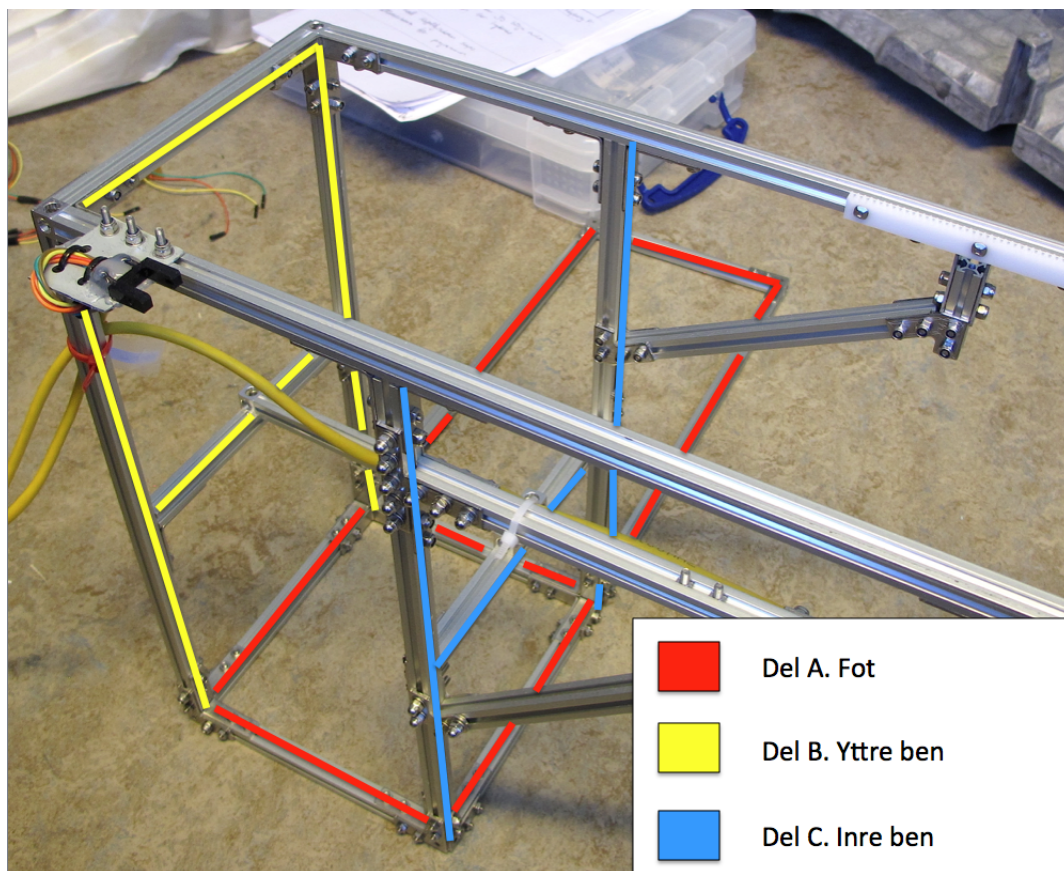
Just den som används är tillverkad av SHARP Corp. och har produktnamnet GP1A57HRJ00F. Den är transmissiv med OPIC (TM) utmatning. OPIC består av ett ljusigenkännande element samt signalbehandling. Denna optiska sensor passar extra bra tack vare ett brett avstånd, 10 [mm], mellan emitter och detektor [21].

5 MakerBeam-konstruktionen

Under de första veckorna fann jag det lämpligt att bygga ställningen som linjärenheten kan färdas på. Efter mycket letande så fastnade jag för MakerBeam, då de hade de delar jag tänktes behöva. Skissarbete drog i detta skede igång för att kunna bestämma hur mycket och vad som skulle beställas. I detta avsnitt kommer genomförandet av konstruktionen att beskrivas. Linjärenheten kommer även att refereras som vagnen i texten.

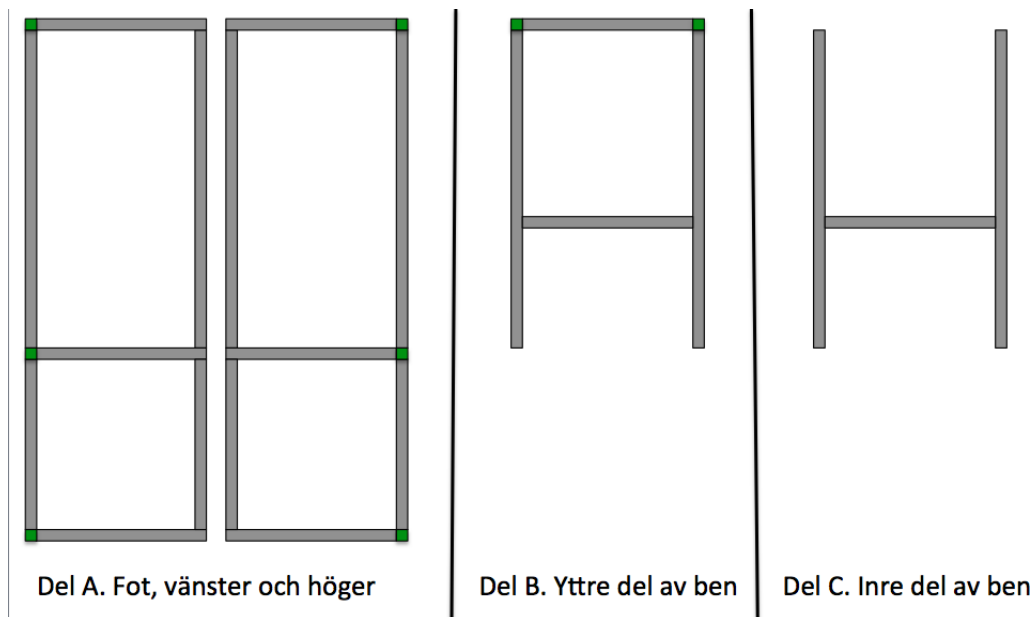
5.1 Ställning med kuggstång

Ställningen med kuggstång byggs upp av MakerBeam-aluminiumprofiler. För att vara stabil konstrueras två fötter, som är 30 cm långa och 15 cm breda. Höjden på ställningen är 32 cm. För att stabilisera ställningen under körning monteras två skruvtvingar som spänner fast ställningen mot bilens mittkonsol.

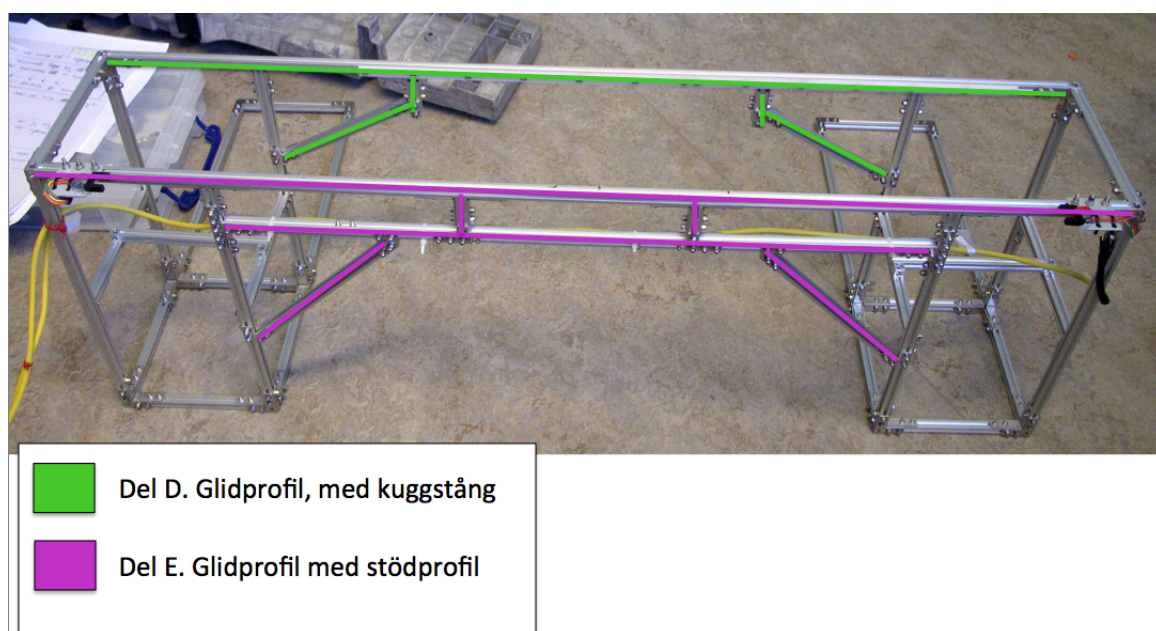


Figur 18: Delar tillhörande ställningen (Författarens bild)

De två profilerna som skall användas av linjärenheten för att rulla på är 90 cm långa. På den ena sidan monteras en 60 cm lång profil som skall göra ställningen styvare. Den ena 90 cm långa aluminiumprofilen utrustas med två kuggstänger. De monteras på ett sådant sätt att kullagren på vagnen inte riskerar att få kontakt med dem. För att få ställningen än mer stabil och styv vid profilerna som vagnen skall färdas på, monteras profiler i vinkel från sidan av benen upp till undersidan av profilerna vagnen rör sig längs. Detta är tydligare i bilderna på ställningen.



Figur 19: Ritningar på i tidigare bild utmarkerade delar (Författarens bild)



Figur 20: Delar tillhörande ställning (Författarens bild)

På ställningen monteras även två optiska sensorer som skall tala om när vagnen når max- och min-position. Sensorerna monteras på utklippta plåtdelar från en robot-bottenplatta. Tack vare hålen i denna plåt kan sensorerna lätta skruvas fast till ställningen och även vid behov flyttas för att ändra begränsningen på var vagnen får röra sig.



Del D. Glidprofil, med kuggstång



Del E. Glidprofil med stödprofil

Figur 21: Ritningar på tidigare utmarkerade delar (Författarens bild)

5.2 Linjärenhet

Det första skedet i byggandet av linjärenheten var att ta fram en bottenplatta som skulle klara av att få plats med alla delar, och kunna röra sig längs ställningen. Bottenplattan byggs som en ram av två stycken 20 cm-profiler på längden och två stycken 10 cm-profiler på bredden. På den ena sidan byggs en hiss-ram av fyra stycken MakerBeam 30 cm-profiler, som får fyra stycken 10 cm-profiler monterade som en kvadrat uppe för att hålla dem stabilt på plats.

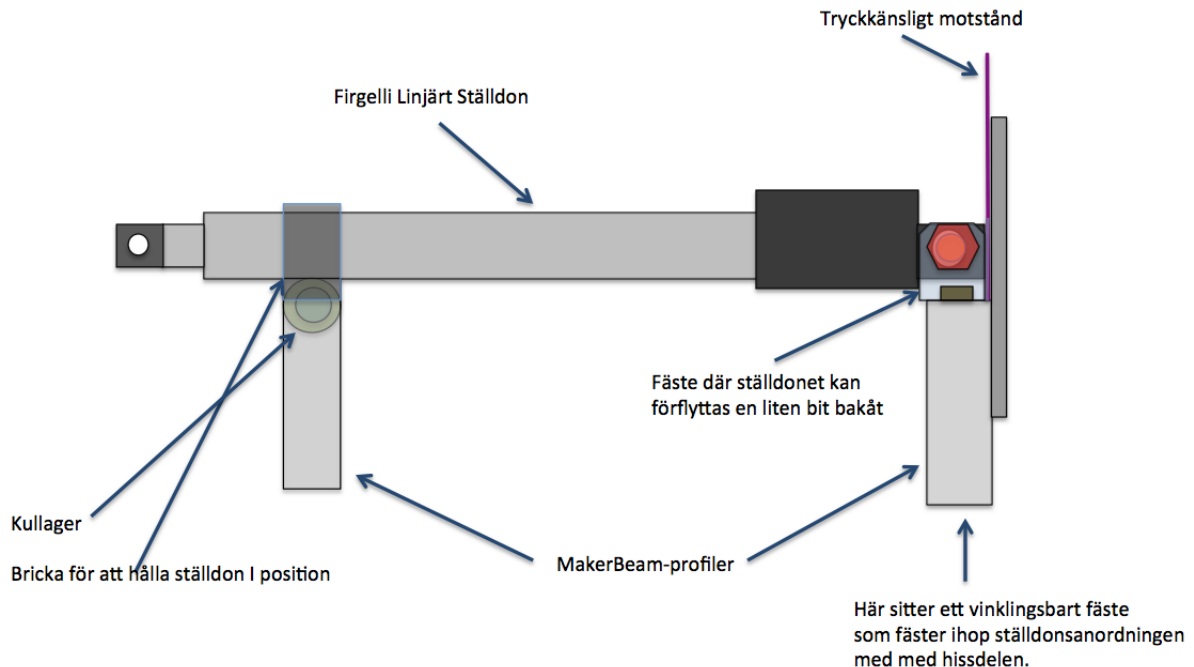
Därefter påbörjas montering av de olika delarna som tillhör linjärenheten.

Stegmotorn som förflyttar linjärenheten i x-led monteras på ett NEMA 17-fäste som är vinklat 90 grader. Detta gör att motorn enkelt kan fästas på två stycken MakerBeam-profiler, bland annat den ena 10 cm-profilen som tillhör bottenplattan. Den andra profilen monteras på ramen och skruvas fast på undersidan.

Stegmotorn, som används för att förflytta hiss-delen upp och ner, monteras på ett rostfritt NEMA 17-fäste från MakerBeam. Sedan monteras detta fäste mellan två stycken 6 cm-profiler, som i sin tur fästs på två 10 cm-profiler. Dessa två 10 cm-profiler monteras på bottenplattan med skruvar på under- och ovansidan. Stegmotorn för förflyttning i y-led är utrustad med en ledskruv, på denna monteras en hiss-del. För att kunna montera MakerBeam-profiler till denna fick en plåtbit klippas ut och borrar upp. På denna plåtbit monteras två stycken 10 cm-profiler som i sin tur monteras på två stycken 15 cm-profiler. På 15 cm-profilerna monteras totalt fyra stycken MakerBeam "vertical guidance brackets", för att möjliggöra en stabil förflyttning längs hiss-ramen som tidigare nämnts. Dessa glidskenor smörjs med kullagerfett för att få så låg friktion som möjligt.

På hiss-delen fästs även Firgellis linjära ställdon. En konstruktion görs där ställdonet monteras i rätt vinkel för att trycka på knapparna. Ställdonet sitter monterat för att

kunna glida en liten bit bakåt, för att trycka på den tryckkänsliga resistorn som monteras bakom. För att stötta ställdonet, vilar det på två kullager som monteras på en MakerBeam-profil på andra sidan av hiss-delen. På hissramen monteras även två OPIC-sensorer, en vid minimumläget och en vid maximumläget för y.



Figur 22: Modell av lösningen med Firgellis Linjära Ställdon (Författarens bild)

När väl detta är gjort återstår endast montering av skenor med kullager. På den sida som de optiska OPIC-sensorerna sitter på, för x-leds min- och max-läge, monteras tre stycken kullagerpar på en 20 cm-profil. Längst ut på vardera sida av profilen, monteras objekt som skall störa ljusstrålen hos de optiska sensorerna. På denna sida monteras även två stycken metallögon att dra kablarna igenom. Detta för att hålla kablarna en bit ut från vagnen så att de inte riskerar att trasslas in eller bli överkörda. Den sida där x-ledsdrivningen sitter, används en 15 cm-profil med tre stycken kullagerpar. Vagnen lyfts upp på ställningen och profilerna med kullager skjuts in till ett läge där de ligger bra, längs banan. Därefter spänns skruvarna åt och linjärenheten rullar fint längs banan. Stegmotorn för x-ledsförflyttning får en axelkoppling, från 5 mm till 1/4 tum, monterad. Sedan monteras axel och tillhörande kuggdrev flyttas till ett läge där det ligger bra mot kuggstången, sedan spänns det på plats.

6 Koppling av komponenter

I detta kapitel beskrivs hur komponenterna kopplats och om det krävts installationer eller om andra inställningar behövt ändras, förklaras tillvägagångssättet här.

6.1 Uppkoppling av Raspberry Pi 2 Model B

Raspberry Pi 2 Model B kopplas till en USB-laddare med en USB Micro B-kabel för att bli strömförsörjd. Skärmen kopplades in med HDMI-kabel, Ethernet användes för att kunna koppla in och uppdatera programvaran på operativsystemet, som fick brännas in på ett Micro SDHC-kort. USB-ingångar utnyttjades för mus och tangentbord.

Raspbian Wheezy är operativsystemet som används för denna Raspberry Pi. Programmet som används för programmeringen heter Geany, detta program installerades genom kommandot: `"sudo apt-get install geany"`.

6.2 Uppkoppling av AD-omvandlare

När Raspberry Pi var redo att användas skall andra komponenter kopplas till den. ADC Pi Plus, AD-omvandlaren från AB Electronics är redan vid leverans utrustad med extra långa stackningsbara stiftlistor, vilket tillåter ytterligare stackningsmöjligheter efter att den kopplats in på Raspberry Pi. Distanser var inköpta från AB Electronics och dessa används för att stabilisera stackningen av HAT-komponenter.

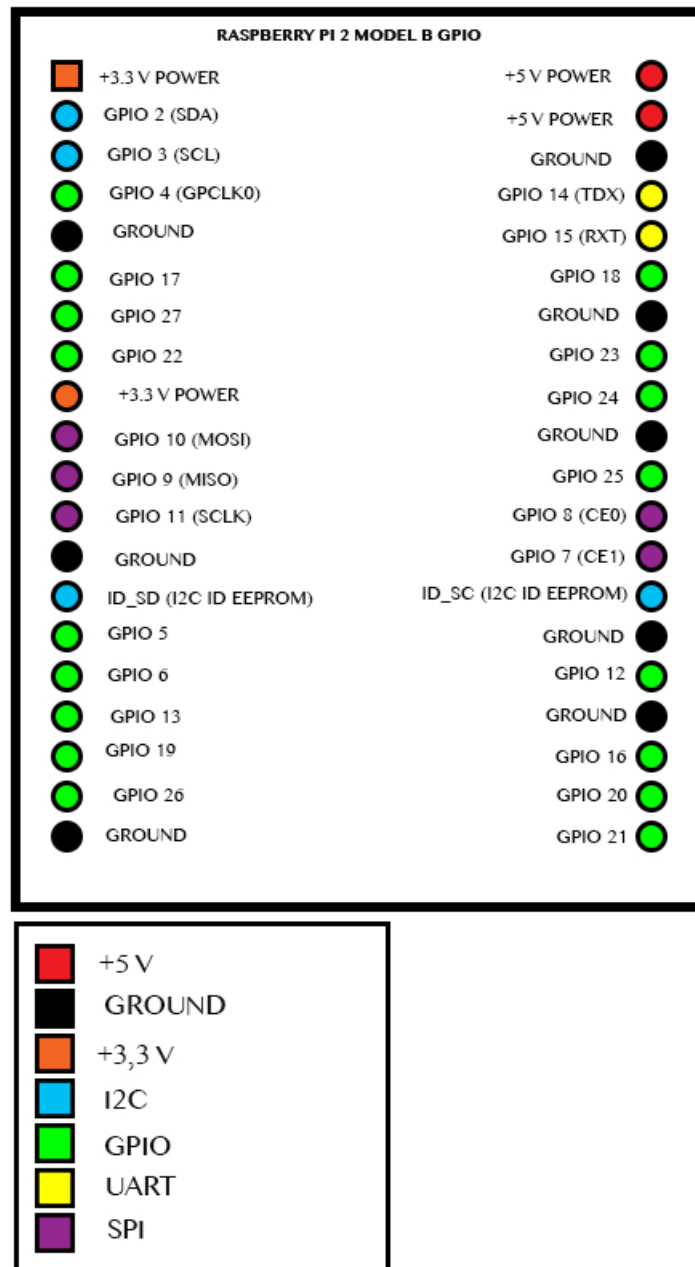
På ADC Pi Plus finns en I2C-adressväljare där två byglar används för att sätta adresserna analogt. I detta fall väljs den förvalda bygelkombinationen som genererar de två I2C-adresserna 0x68 och 0x69 för vår AD-omvandling. Kablar löds fast till de olika kanalerna på kortet för att underlätta koppling senare när fler kort har stackats ovanpå.

Kortet kopplas på Raspberry Pi och nästa set med distanser skruvas också dit för att hålla kortet på plats. Ström kopplas till Raspberry Pi och i terminalen körs kommandot: `"sudo i2cdetect -y 1"`. Detta kommando fungerar som en kontroll vid koppling att kortets adresser visas på den platsen de ska. Kortet använder endast två GPIO-pinnar från Raspberry Pi för att kommunicera, det är GPIO 0 som motsvarar I2C0_SDA och GPIO 1 som motsvarar I2C0_SCL.

6.3 Uppkoppling av Stepper Motor-HAT

Adafruits Stepper Motor-HAT levereras utan fastlödd stiftlist och anslutningsplintarna är inte heller ditlödda. Först lödes anslutningsplintarna på plats. Då stiftlisten som följer med inte är stackningsbar, har flera sådan beställts i förväg. En stackningsbar stiftlist lödes fast och kopplingskablar skruvas på plats i plintarna. Två kablar till strömförsörjning skruvas också in till plinten ämnad för det. Dessa två kablar kommer senare kopplas till PWM-HAT som har ett 2,1 mm likströmsjack som kommer att vara inkopplat.

På Stepper Motor-HAT går det att välja I2C adress genom att löda ihop två byglar på kortet. Det finns fem byglar och adressen väljs binärt. Jag valde att använda adressen 0x61, vilket innebar att jag lödde en bro mellan byglarna på A0. Det sitter en PCA9685 på denna Stepper Motor-HAT, den använder sig av + 3,3 V och de två I2C-pinnarna, alltså GPIO 0 och GPIO 1 från Raspberry Pi för att kommunicera med hjälp av I2C.



Figur 23: Karta över Raspberry Pi 2 Model B GPIO-stift (Författarens bild)

Stepper Motor-HAT monteras ovanpå stiftlisten från AD-omvandlaren. Sedan skruvas fyra nya distanser på för att hålla kortet på plats. Återigen ges Raspberry Pi strömförsörjning och I2C-adresserna kontrolleras igen.

6.4 Uppkoppling av PWM-HAT

Denna PWM-HAT från Adafruit, levereras utan fastlödda 3x4-stiftlistor, GPIO-stiftlist eller anslutningsplint. Då den levereras med stiftlistor av 3x4-typ som är ämnade för kabelkoppling rakt ovanifrån, fick vinklade 3x4-stiftlistor inköpas för att ytterligare stackning

skulle vara möjlig. Först lödes dessa stiftlistor fast, samt en stackningsbar stiftlist till GPIO-stiften. Anslutningsplinten lödes också fast.

Liksom på Stepper Motor-HAT, kan I2C-adress väljas genom att löda en bro mellan två byglar. Här väljs adress 0x42 genom att löda en bro på A1. Även på PWM-HAT sitter en PCA9685 som liksom Stepper Motor-HAT använder +3,3 V, GPIO 0 och GPIO 1 från Raspberry Pi.

PWM-HAT monteras ovanpå Stepper Motor-HAT och skruvas fast med muttrar för att det sista kortet som skall stackas, inte passar till distanserna. De två strömförsörjningskablarna som är kopplade till Stepper Motor-HAT, skruvas till anslutningsplinten på PWM-HAT. Raspberry Pi matas åter med ström för att möjliggöra I2C-testet och kontroll sker att adresserna stämmer.

6.5 Koppling av tryckkänsliga motstånd

De tryckkänsliga motstånden löds fast till två kopplingskablar för att inkoppling och test skall bli möjliga. Det ena tryckkänsliga motståndet monteras bakom Firgellis linjära ställdon där det limmas med värmelim för att hålla biten, efter den tryckkänsliga delen, styv. Annars kan den påverka motståndet något, utan att något utsätter det för kraft. Kablarna till detta motstånd löds till två kablar i en Ethernet-sladd, där de andra sex sladdarna kommer att användas till ställdonet. I motsvarande kablar på andra änden av Ethernet-sladden kopplas två kopplingskablar in. En av dessa kopplas till PWM-HAT och ett stift för +5 V. Den andra kopplas till Channel 1 på ADC Pi Plus. Det är med denna princip det andra tryckkänsliga motståndet, för inspelning av knapptryck, kopplas.

6.6 Koppling av optiska sensorer

För att kunna använda de optiska sensorerna krävs en mindre koppling, som i detta fall görs genom att löda fast ett 220 Ω -motstånd mellan jord och ben 2 (katod) på den optiska sensorn. Här passas även på att koppla in ben 5 till jord, samt att koppla ben 1 (anod) och ben 3 (Vcc) till 5V, på så sätt behöver endast tre kablar dras vidare från sensorn. Ben 4 är till för signal. När de fyra optiska sensorerna är monterade till anordningen kan kablarna kopplas in. På PWM-HAT lödes fyra stycken kopplingsladdar till jord och fyra stycken till +5 V. Till dessa kopplas sensorernas jord- och +5 V-kablar. De fyra signalkablarna, som har märkts med vilken sensor de tillhör, kopplas in till Raspberry Pi via GPIO-stiften. Här används GPIO 20 till x-min och GPIO 21 till x-max. Sensorerna för y-max och y-min kopplas till GPIO 26 respektive GPIO 16.

6.7 Koppling av Firgellis linjära ställdon

Detta ställdon har tillhörande sex stycken kablar. De som används för denna anordning är jord (svart), +5 V (röd), RC-styrning (vit) och återkoppling (lila). Trots att det endast är fyra stycken kablar som används så har samtliga sex blivit skarvade genom en Ethernet-sladd. Detta ifall det vid senare tillfälle skulle vara aktuellt med en annan typ av styrning. Ställdonet kommer att styras via PWM-HAT och till den kopplas jord, +5 V och RC-styrningskabeln. Återkopplingskabeln kopplas istället in till ADC Pi Plus Channel 5.

6.8 Koppling av stegmotorer

Det första som görs med stegmotorerna är att skarva om kablarna via en givarkabel. Givarkabeln har fyra kablar i sig, så en givarkabel används per motor. Noga löddes stegmotorernas kablar ihop med givarkabelns. Sedan användes krympslang för att isolera. I andra änden av givarkabeln lödes kopplingskablar i motsvarande färger på och krympförband används återigen för att isolera.

På Stepper Motor-HAT finns det två stycken TB6612-chip som vardera har två H-bryggor. Det första TB6612-chipet är kopplat till motorkoppling M1 och M2, här kopplas stegmotorn för förflyttning i x-led. Det andra TB6612-chipet är kopplat till motorkoppling M3 och M4. Till M3 och M4 kopplas stegmotorn med ledkula som används för y-ledsförflyttning. När det sedan är dags att driva motorerna så kopplas en 5 V-adapter in till 2.1 mm-jacket på PWM-HAT och förser sedan stegmotorerna med ström genom de två anslutningsplintarna på PWM-HAT och Stepper Motor HAT.

6.9 Uppkoppling av PiCAN

Vad gäller koppling av PiCAN så är den inkopplad genom 26-pins stiftlist, ej stackningsbar. Detta kort är till en början avsett för de äldre Raspberry Pi-modellerna. Kortet behöver en del inställningar och installationer för att börja fungera och en del är utförda. Inställningsmässigt är det grejer kvar innan CAN-kommunikation kommer att fungera.

7 Programmering

I följande avsnitt kommer programmeringen av den autonoma roboten för knapptryckning att beskrivas.

7.1 Val av programmeringsspråk

Då Raspberry Pi lämpar sig bra för att använda Python, så stod det för mig att välja mellan Python och C. C-programmering valdes av den orsaken att det är det språk som vi har programmerat mest med, i skolan och det är det språket jag känner mig mest bekant med.

7.2 AD-omvandlaren, samt program för tryckkänsligt motstånd

ADC Pi Plus programmerades till att utföra 240 SPS och detta med 12-bitars noggrannhet. Detta genom att sätta det programmerbara datamängdsvalet i microcontrollern Microchip MCP3224, som det finns två stycken på ADC Pi Plus, till 00. Detta gjordes för att antalet SPS i fallet där det linjära ställdonet ska känna av tryckkraft vid knapptryckning är viktigt att hålla högt. Ställdonet rör sig maximalt med 23 mm per sekund och om den högsta noggrannheten skulle användas, så hade maximalt antal SPS varit 3.75. Risken att trycka sönder knappen skulle då bli för stor.

7.3 PWM-HAT, rörelse av Firgellis linjära ställdon

PWM-HAT styr linjära ställdonet. Här programmeras en PWM-signal där värden på tillslagstiden ställs mellan 1,0 till 2,0 ms. Då kan donet köras till max slaglängd och även köras helt in. Värdena däremellan ändras linjärt med hur långt ställdonet ska gå.

Det tryckkänsliga motståndet är programmerat att hämta in värden på tryckkraft och när önskat värde är nått stannar ställdonet tills önskad nedtryckningstid är uppnådd, sedan åker ställdonet tillbaka in igen. Antalet knapptryck räknas och när detta antal är nått går programmet vidare till nästa uppgift.

Det linjära ställdonets återkopplingsfunktion tar emot värden för att kunna avgöra hur långt ut ställdonet är.

7.4 Stepper Motor-HAT, rörelse av stegmotorer

Vid programmering av stegmotorerna så ställdes en PWM-signal där motorn gick ganska snabbt, men i princip ljudlöst. Förskjutningen av de fyra PWM-signalerna som behövdes per motor programmerades och kontrollerades sedan i oscilloskop. De fyra PWM-signalerna är förskjutna med 90 grader mot varandra och beroende på hur, kan även positiv och negativ riktning ställas. När motorerna kördes fint, så testades att bara köra mindre sträckor upprepat och vagnen fortsatte att rulla en bit efter, i x-led. Det blev då dags att programmera in en broms till stegmotorn så att den både stannade fint och låg bromsad mot banan.

De optiska sensorerna som kopplats till Raspberry Pi lades till i programmet och en funktion skapades där motorn byter håll vid det tillfälle då en av de två sensorerna i

x-led fick störningar. En liten fördröjning lades till för att hinna rita streck vid ändlägena i x-led med en White board-penna. Sedan mättes avståndet däremellan. En tidsräknande funktion lades till och där skrev även programmet ut tiden. Tiden skilde endast 2 hundra radelssekunder när den kördes fram och tillbaka många gånger. Utifrån sträckan och tiden kunde en konstant tas fram för hur många sekunder det tog att köra en millimeter. Därefter kunde en funktion byggas för att välja hur långt stegmotorn skulle köra.

Programmeringen för y-ledsförflyttningen fungerar liknande. Även här optiska sensorer för att tala om ifall roboten varit för långt iväg. Stigningen på ledskruven är 0,4 mm per varv. Dock mäts även här tiden och max och min-läge.

Där efter skall alla positioner, för knapparna, läggas in. En hel del ska även göras för att få in detta program till att köra med det användargränssnittet som byggs i Glade. För att underlätta arbetet för roboten så skapades en egen tråd för att ha koll på själva styrningen av rörelser. I denna tråd ingick rörelse i positiv- och negativ-riktning både i x- och y-led. Tryckkänsliga motståndet och det linjära ställdonet ingick också i denna tråd.

7.5 PiCAN, mottagande av CAN-bus

SK-Pang Electronics PiCAN hann aldrig programmeras, den kopplades dock till Raspberryn och instruktioner från SK-Pang Electronics hemsida talade om steg för steg hur den skulle kunna gå att använda i Raspberry Pi. Det blev dock problem vid ett ställe och prioriteten hamnade på andra delar än CAN-kommunikation.

8 Användargränssnitt

Vid framtagning av ett användargränssnitt övervägdes att antingen basera det för att köra på en stor datorskärm och använda tangentbord för att mata in värden eller att basera det på en touch-skärm. Det senare alternativet som också har valts är en 5-tums touch-skärm med HDMI- och USB-ingångar. USB-ingången används som strömförsörjning till skärmen och hjälper även till med touch-funktionen.

Här är det tänkt att montera skärmen på en aluminiumplåt, som även kommer att rymma ett antal knappar, en knapp skall finnas för tryckinspelningsfunktionen. De andra kan finnas för framtida funktioner.

8.1 Val av utvecklingsmiljö

För att skapa ett grafiskt användargränssnitt i C-programmering kan GTK-biblioteket användas. I detta fall valdes två möjliga alternativ ut. Antingen kunde det grafiska gränssnittet utvecklas genom att skriva all kod, med hjälp av GTK-biblioteket, i Geany, eller så kunde någon typ av RAD-verktyg användas. Att sätta sig in i GTK-biblioteket och sedan skriva det grafiska gränssnittet i Geany och med C-kod skulle ta för lång tid. På grund av detta valdes utvecklingsmiljön Glade som passar in på det senare alternativet.

I Glade byggs användargränssnittets utseende upp genom att skapa fönster och i dessa dra in olika delar, så som tryckknappar, textrutor et cetera, som är önskvärda. Dessa olika delar tilldelas namn som sedan kan användas i funktioner i C-programmet sedan. Fördelen med att använda Glade är att utseendet på användargränssnittet hela tiden är synligt under utvecklingen och att koden sedan genereras utifrån valt utseende. Användargränssnittet sparas i XML-format och kan användas i C-programmeringen genom att använda GtkBuilder.

8.2 Grafiska användargränssnittet

I Glade har ett grafiskt användargränssnitt designats. Här kommer de olika delarna presenteras och tanken bakom dem förklaras.

Det första användaren möts av är en sida som välkomnar användaren, här står lite information om vilka som är bakom projektet.

Vid tryck på knappen "Meny" förflyttas nu användaren till en meny med fyra alternativ att välja på.

Vid tryck på knappen "About" förflyttas användaren tillbaka till välkomstsidan.

Trycker användaren på knappen "Spela in tryckkraft för att använda i testet" går programmet till en sida där användaren blir ombedd att trycka på en knapp ett visst antal gånger. Sedan förflyttas användaren till sidan med rubriken "Tryckkraft", där är nu ett genererat värde förvalt.

Vid tryck på "Köra med förvalda värden" sker följande. Programmet förflyttar användaren till sidan, med frågan "Vilka knappar önskas testas?", som visar upp en panel med knappar som dessutom är indelad i fyra rutor, "A", "B", "C" och "D".

Om användaren väljer knappen "Sätta parametervärden manuellt", sker ett hopp till nästa sida, med rubriken "Tryckkraft".

Här kan ett värde mellan x och y [N] sättas, alternativt kan en ruta kryssas i för att slumpa tryckkraften till värden mellan x och y inför varje testtryckning.

Det finns även tre knappval nedan. "Föregående" leder till föregående sida och om valet att ställa parametervärden manuellt har gjorts kommer användaren flyttas tillbaka till Menyn. Likaså om "Till Meny" valts. Har användaren däremot valt att själv spela in tryckkraften, kommer denne flyttas tillbaka till att göra så igen. Väljer användaren "Nästa" förflyttas denne till sidan med rubrik "Nedtryckningstid".

Här kan antingen nedtryckningstiden väljas manuellt eller om användaren så vill, kryssa i att den skall slumpas till en viss nedtryckningstid som kommer gälla för varje testtryckning. "Föregående" leder tillbaka till sidan "Tryckkraft". "Till Meny" kommer att fungera likadant var användaren än är, alltid tillbaka till menyn. "Nästa" leder i detta fall till sidan med rubrik "Träffnoggrannhet".

I detta fönster kan användaren välja ett eller flera alternativ om var på testknappen trycket skall ske. Om mer än ett val är gjort, slumpas alternativen mellan de olika knapparna som ska testas. De alternativ som finns är "I centrum av knappen", "Till höger" eller "Till vänster". "Föregående" leder till sidan "Nedtryckningstid". "Nästa" förflyttar användaren till sidan med rubriken "Vilka knappar önskas testas?"

Här finns instruktioner om att delar presenteras var för sig. Trycker användaren på "Föregående" kan det nu leda till två olika alternativ. Om användaren kom till denna sida från sidan "Träffnoggrannhet", kommer användaren skickas tillbaka till den sidan. Om användaren i menyn valde knappen "Köra med förvalda värden", kommer användaren nu förflyttas tillbaka till menyn. Knappen "Nästa" leder till sidan med rubriken "A. Vilka knappar önskas testas?"

Bilden på den sidan är uppdelad i flera rutor som är numrerade. Till höger kan användaren kryssa i vilka knappar som skall testas. När användaren är klar med sina val, trycks knappen "Nästa" och förflyttning sker till sidan med rubriken "B. Vilka knappar önskas testas?"

Här gäller samma princip som för den tidigare bilden med A. "Nästa" tar nu användaren till sidan med rubriken "C. Vilka knappar önskas testas?"

Här gäller samma princip igen och knapptryck på "Nästa" leder till en sida med rubriken "D. Vilka knappar önskas väljas?"

När valen nu är gjorda på denna sida, kommer knappen "Nästa" leda till en sida med rubriken "Hur många gånger ska var knapp testtryckas?"

Här ska användaren välja ett värde mellan x och y, om inget val görs så står 20 tryck/-knapp som förval. Vid tryck på "Nästa" skall användaren bekräfta att testprogrammet skall starta med de valda värdena, som då också står på skärmen.

8.3 Fysiska delarna

De fysiska delarna tillhörande användargränssnittet var tänkt att bestå av en touch-skärm monterad på en aluminium-panel. Dock återstår arbete med detta då ett hål behöver fräsas för skärmen. Cirkulära hål skulle fräsas för att kunna montera tryckknappar exempelvis för tryckkraftsinspelningsfunktionen. På baksidan av panelen fästes det MakerBeam-profiler som skulle göra det möjligt för panelen att stå på plana ytor, exempelvis ett bord. På knappen som skall spela in tryckkraft har ett tryckkänsligt motstånd monterats.

9 Test och verifiering av konstruktionen

Oscilloskop har använts för att kunna jämföra PWM-signalerna och deras förskjutning mot varandra. Det användes även för att se att det blev någon fyrkantsvåg överhuvudtaget.

Vid koppling och programmering har delar testats var för sig för att se om de fungerar. Tester har gjorts där bland annat vagnen har fått köra mellan max- och min-läge i x-led och där y-led samtidigt har körts. Under tiden de kördes har även det linjära ställdonet körts in och ut för att testa om alla delarna kan köras tillsammans. Detta är dock inte vad själva roboten ska göra egentligen, utan mer ett test att det gick att förflyttas i alla led samtidigt. Med hjälp av bromsfunktionen ligger vagnen stabilt på ställningen.

För stegmotorerna skapades även en funktion som med hjälp av den interna klockan räknade ut hur lång tid det tog att förflytta vagnen från ena sidan till andra. Avståndet däremellan mättes och en konstant räknades ut för hur många sekunder en millimeter tog att köra. Utifrån detta testades olika värden för hur långt vagnen skulle ta sig. Detsamma skedde i y-led. Detta kontrollerar ungefärligt hur stort felet är, genom att skriva ut tiderna på skärmen var gång den kört.

Ställdonet har testats att köra ut till maximal slaglängd och sedan längst in och detta i max-hastighet. Den har testats att sakta stegas ut genom att lägga till en for-loop som ökar tilltiden på PWM-signalen en liten bit i taget och med en liten fördröjning, på så sätt kunde ställdonet testas för långsam förflyttning.

Till en början testades funktionerna för de tryckkänsliga motstånden genom att endast trycka med fingret på dem och att i programmet skriva ut värdet, detta för att verifiera att de fungerade. Likadant med optiska sensorerna, de funktionstestades först med att skriva ut deras värden och sedan var och en av dem genom att störa dem.

10 Resultat

En anordning som kan förflytta sig i x-, y- och z-led har tagits fram. Positionering av anordningen sker med tidsräkning och inte genom räknade av pulser.

De två inledande frågorna från avsnitt 1.4 kan besvaras.

Hur kan knapptryckning genomföras?

Det kan genomföras med hjälp av ett linjärt ställdon.

Rörelse i x- och y-led kan ske på flera sätt, vilket passar bäst för denna anordning?

Det finns flera sätt att röra sig i x- och y-led. Då det endast är denna lösning som testas så kan inget svar ges på om detta sätt passar bäst. Vid projektets start undersöktes möjligheten att använda en robotarm, om vilket beslut fattades att inte använda.

Början på ett användargränssnitt har tagits fram. Test av touch-skärm har genomförts och det finns inga tecken som tyder på att användargränssnittet inte skulle gå att använda med hjälp av denna, samt fysiska knappar monterade på en aluminiumpanel.

Till denna del fanns följande frågor att besvara.

Hur kan tryckkraft kännas av och registreras?

Tryckkraft kan mätas med ett tryckkänsligt motstånd, som använts för denna anordning. Med hjälp av en AD-omvandlare kan värden skickas in till Raspberry Pi och där får programmet avgöra kraften med lämplig kod.

Vilka sensorer finns för att klara av uppgiften?

För att anordningen skall kunna veta position att utgå ifrån i x-led och y-led, så fungerar optiska sensorer utmärkt. De tryckkänsliga motstånden fungerar för att klara av uppgiften med hur hårt ställdonet skall trycka på knappen som testas.

Hur kan hastigheten ställas?

Hastigheten ställs via PWM-HAT och periodtiden på PWM-signalen. Den kan försiktigt stegas upp eller slås på fullt direkt. Det skiljer från 1.0 ms till 2.0 ms ifrån fullt indrag till fullt utslag. Värden däremellan bestämmer procentuellt positionen på ställdonets yttersta punkt i z-led. Genom att sakta öka värdet kan hastigheten sänkas. Full hastighet ges genom att direkt välja positionen ställdonet skall inta.

Vilket tryckkraftsintervall skall användas?

Det tryckkänsliga motståndet har ett mätområde från 0,2 – 20 [N], detta område är lämpligt att kunna välja värden i, som säkerhet kan intervallet 0,3 – 18 [N] vara mer lämpligt.

Det andra delmålet var att implementera en funktion som spelar in när användaren trycker på en knapp, registrera krafter och sedan spela upp denna kraft vid testet av knapparna. Här är det en del kvar att göra för att nå dit, användargränssnittet är det som begränsar där. Att registrera en kraft och sedan trycka med samma kraft på en annan knapp är programmeringsmässigt inte svårt.

Frågorna som tillhörde detta avsnitt var följande.

Hur kan inspelningen fungera?

Genom att använda ett tryckkänsligt motstånd av samma typ som för ställdonet kan en spänning registreras vid ett knapptryck där det är monterat, sedan ordnar en funktion i programmet att ställdonet vänder tillbaka när samma spänning uppnåtts vid knapptrycket.

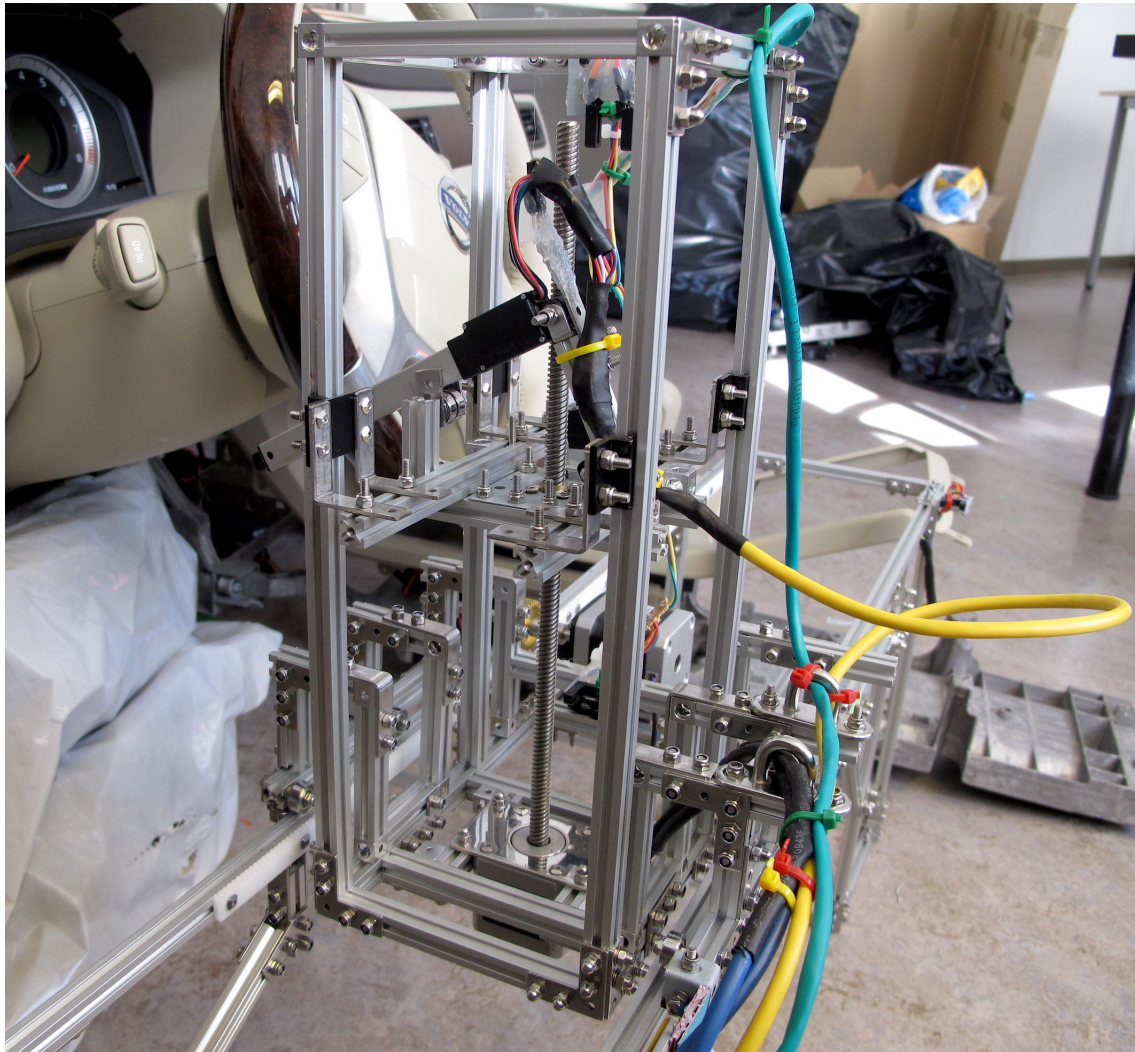
Kan tryckkänsligt motstånd användas för att spela in hur hårt knappen tryckts vid inspelning?

Ja.

Det tredje delmålet var att möjliggöra CAN-kommunikation mellan anordningen och de knappar som testas. Frågan som tillhörde denna del löd.

Hur kan värdena från knapptrycken registreras?

Genom att använda ett kort som heter PiCAN, som tillverkas av SK-Pang Electronics. Med hjälp av OBDII-kontakt och två kablar från kortet kommer Raspberry Pi sedan åt CAN-H och CAN-L.



Figur 24: Linjärenheten på ställningen (Författarens bild)

11 Slutsats

11.1 Resumé

Arbetet har flutit på bra, det var ett tag där leveranserna på vissa delar blev något utdragna. En miss i att inte beställa tillräckligt med specials kruvar till MakerBeam-konstruktionen, stannade upp det bygget i cirka en vecka. Detta var inte så farligt då kunde nya uträkningar göras för vad som behövde beställas. Att hinna allt detta och få artiklarna på en vecka är bra då MakerBeam skickas från Nederländerna. Att ta fram konstruktionen och finna saker som passar ihop var något joxigt, och vissa detaljer fick borrar och bockas för att lösa problemen. För att lösa problem med alla kopplingar, så har diverse kablar skarvats om och ihop med motorer och sensorer.

Det som blivit något lidande under projektet är programmeringsbiten. Sensorerna fungerar som de ska, likaså de tryckkänsliga motstånden. Ställdonet går att köra fram och tillbaka. Stegmotorerna går bra, fast snabbt. Stegningen har inte fungerat som tänkt, så positionering har skett genom att räkna ut tiden det tar för vagnen att köra hela sträckan och sedan avståndet. På så sätt blir det en hyfsad, ungefärlig position. Ett användargränssnitt har designats i utvecklingsmiljön GLADE, och fungerar ännu inte tillsammans med anordningen.

11.2 Kritisk diskussion

I detta avsnitt skall återkoppling ske mot inledningen. Här beskrivs styrkor i arbetet och även svagheter. Resultatet skall utvärderas, likväl som de andra väsentliga delarna av arbetet.

11.2.1 Examensarbetstid

Då jag fick tag på detta examensarbete hade de flesta andra examensarbeten varit igång i nästan en månad. Detta borde inte ha varit ett problem sett till tidsschemat för att få läsa Master of Science i höst. Dock blev det väldigt stressigt när datum om sista redovisningstid dök upp och jag vid det tillfället skulle ha veckor kvar av mitt arbete. Att inte hinna klart skulle innebära att Mastern får vänta ett år, med flytt från studentbostad och annat som följd. Denna stress har varit något påfrestande, för arbetet i sig har blivit något lidande av detta. Därför stannar jag kvar några veckor efter redovisning för att få iordning lite till.

11.2.2 Förstudie, WBS och GANNT-schema

Till att börja med kan en återblick göras till upplägget att starta med förstudie, göra en WBS och planera in denna i ett GANNT-schema. Detta kräver mycket tid i början. Det är många delar att bli insatt i och beslut ska tas så fort som möjligt, utan att slarva. Att utföra en WBS är bra då hela arbetet bryts ner till små hanterbara bitar. Det blir tydligt vad som skall göras och det blir lättare att planera vad som behöver köpas in.

Att placera allt detta i ett GANNT-schema, var i efterhand inte speciellt klokt. Hade ändringar kunnat göras så hade jag valt att behålla alla nedbrutna delarna från WBS

och placerat de på separata A4 sidor, dellösning för dellösning med tillhörande arbetsuppgifter. I själva GANNT-schemat hade jag haft delmålen inplanerade och så hade sedan tillbakablick fått göras till WBS för att ta reda på arbetsuppgifter under tiden dellösningar arbetas med.

11.2.3 Genomförande

Vid genomförandet fanns både positiva och negativa bitar. Det första som köptes in var en Raspberry Pi 2 Model B, som möjliggjorde att då inget annat fanns att göra, så gick det att bekanta sig med Raspberry Pi. GLADE installerades och i väntan på komponenter att bygga ihop så påbörjades inläring av hur GLADE fungerar. Här byggdes ett grafiskt användargränssnitt upp, dock endast designen. Det negativa var att när komponenter anlände och det började byggas på dem, så kom programmeringen i Glade av sig. Jag skulle ha haft bättre struktur på när jag skulle börja med något annat. Samtidigt var det bråttom att få ihop de mekaniska lösningarna tidigt för att bekräfta att de var möjliga lösningar.

Att det var viktigt med struktur och ordning på de olika komponenterna var jag tydlig med redan från början och det har fungerat över lag. Det blir en hel del kablar att hålla reda på. MakerBeam-delarna har legat för sig, elektronik-delarna för sig och kopplingskablar, knappar och annat har haft sin plats. Då blir det enklare att arbeta och onödig tid läggs inte på att leta runt i en enda röra.

En nackdel med genomförandet är att det skulle ha gått mycket snabbare. Om jag planerat på att få hela arbetet klart på 6 veckor, så hade jag haft råd med någon försening, men ändå hunnit tills redovisningsdatum. Samtidigt skall arbetet ta 10 veckor totalt.

Jag var dålig med att följa upp alla delar i GANNT-schemat, och detta på grund av ett allt för tydligt och nedbrutet schema. Det tog helt enkelt för lång tid att sitta och leta igenom schemat för var sak som skulle göras och när. Här har leveranser också haft en påverkan.

Vid rapportskrivningen har väldigt mycket tid gått åt för att försöka hitta programvara att ta fram kretsscheman i. DesignSpark har laddats hem, dock fanns inte i närheten av de delar jag behövde. Då fick internet användas till att hitta guider hur komponenter byggs upp i DesignSpark, det blev till sist för tidskrävande för att kunna sammanställa rapporten i tid. PowerPoint har fått agera ritningsprogram, likaså Photoshop, som också blev för svårt när olika delar skulle klippas ihop och sedan kopplas. Lagerfunktionerna där förstår jag mig inte på.

Programvara till Apple är ibland problematisk, jag kör även Windows på min MacBook, för att kunna köra många av de program som ibland behövs. Visio som skulle användas till flödesschema-uppbyggnad tog också sin tid att få igång, dock fick jag igång det.

11.2.4 Resultat

För tillfället så är den största svagheten programmeringen. Stegmotorn fick jag inte att gå som tänkt, steg för steg, utan får köras och tidtagning får styra när den ska stanna. Koordinater för testknapparna är ännu inte inlagt. Dessa behövs.

En styrka hos stegmotorer är att kunna förflytta sig stegvis. Utifrån detta räkna antalet pulser och således veta antalet steg. Då kan position bestämmas. En svaghet hos

stegmotorer är att om pulsen som skickas inte leder till ett steg, så räknas det ändå som ett steg. Det finns ingen återkoppling från stegmotorn som kan tala om position. Fördelarna med stegmotorn överväger nackdelarna, då stegmotorn är stark och klarar av att förflytta vagnen på ett säkert sätt.

Jag ser en stor nackdel i att jag ännu inte hunnit med användargränssnittet, då det hade varit snyggt att kunna beställa knapp att trycka på och att spela in egen tryckkraft och sedan testa knapparna med densamma.

En lösning att kunna trycka på touch-skärmar är på gång, toppen av en touch-penna har sågats av pennan och skall endast monteras på en utbytbar spets till ställdonet. Tack vare den bakom ställdonet placerade tryckkänsliga resistorn är det inga problem att byta tryckspets.

För att få ett bättre resultat så kunde någon typ av microcontroller kunnat användas för styrningen av alla delar och sedan hade Raspberry Pi använts till det grafiska. Stegmotorerna hade antagligen blivit enklare att styra om en microcontroller använts. Någon form av återkoppling kopplad till stegmotorerna hade varit fördelaktigt också.

11.3 Vidareutveckling

För att bestämma positionen hade en pulsgivare kunnat användas, för att räkna hur många pulser som hunnit gå fram till stegmotorn och således antalet steg som stegmotorn ska ta.

En så kallad SoftPot hade kunnat användas genom att fästa den på ytterligare en aluminiumprofil och sedan monterat en tryckande del ner från vagnen mot denna. En SoftPot är en potentiometer som vid tryckkraft någonstans på den kan avgöra positionen.

En förbättrad infästning för ställdonet och en servomotor för att kunna ställa nedtryckningsvinkel.

Referenser

- [1] J. Lilliesköld M. Eriksson. “*Handbok för mindre projekt*”. I: Första upplagan, 3. Stockholm, Sverige: Liber AB, 2009. Kap. Projektstart - Hjälp jag har blivit projektledare!
- [2] H. Krauss H. Schock S. Tillner Översatt av D.H. Hopewell H. Moczała J. Draeger. “*Small electric motors*”. I: English edition. (IEE POWER AND ENERGY SERIES 26). Trowbridge, UK: Redwood Books, (1998). Kap. 10. Stepper motors -principles and applications.
- [3] P. P. Acarnley. “*Stepping Motors -A Guide to Theory and Practice*”. I: Fjärde utgåvan. (Control engineering series: No. 63). Stevenage, UK: The Institution of Electrical Engineers, (2002). Kap. 1. Stepping motors.
- [4] Wantai Motor. Hybrid Stepper Motor Mod.
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Robotics/42BYGHM809.PDF> (Besökt 2015-06-21). (2011).
- [5] Changzhou Songyang Machinery och Electronics New Technic Institute. High Torque Hybrid Stepping Motor Specifications.
http://www.pololu.com/file/download/SY42STH38-1684A.pdf?file_d=0J715(Besökt 2015 – 06) (2013).
- [6] N. Sclater. “*Mechanisms and Mechanical Devices Sourcebook*”. I: Femte utgåvan. New York, Chicago, San Francisco, Lisabon, London, Madrid, Mexico City, Milano, New Dehli, San Juan, Seoul, Singapore, Sydney, Toronto: McGraw-Hill Education, (2011).
- [7] Firgelli Technologies Inc. Miniature Linear Motion Series - L12.
http://www.firgelli.com/Uploads/L12_datasheet.pdf (Besökt 2015-06-03). (2008).
- [8] MakerBeam. Specifications MakerBeam and OpenBeam.
http://www.makerbeam.eu/epages/63128753.sf/en_GB/?ObjectPath=/Shops/63128753/Categories (2015).
- [9] Actobotics. Actobotics(TM) is...
<https://www.servocity.com/html/actobotics.html> (Besökt 2015-06-10). (2015).
- [10] Adafruit Industries. Raspberry Pi 2, Model B.
<http://www.adafruit.com/pdfs/raspberrypi2modelb.pdf>. chrome-extension://mhjfbmdgcfjbbpaec (Besökt 2015-06-03). (2015).
- [11] Adafruit Industries. Assembled Pi T-Cobbler Plus- GPIO Breakout for RasPi A+/B+/Pi2.
<https://www.adafruit.com/products/2028> (Besökt 2015-06-03). (2015).
- [12] L. Fried. Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi.
<https://learn.adafruit.com/downloads/pdf/adafruit-16-channel-pwm-servo-hat-for-raspberry-pi.pdf> (Besökt: 2015-06-03). (2015).
- [13] NXP Semiconductors N.V. PCA9685.
http://www.nxp.com/documents/data_sheet/PCA9685.pdf(Besökt 2015 – 06 – 03). (2015).

- [14] L. Fried. Adafruit DC and Stepper Motor HAT for Raspberry Pi.
<https://learn.adafruit.com/downloads/pdf/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi.pdf> (Besökt: 2015-06-03). (2015).
- [15] Toshiba. TB6612FNG.
<http://toshiba.semicon-storage.com/info/docget.jsp?did=10660prodName=TB6612FNG>
 (Besökt 2015-06-03). (2014).
- [16] AB Electronics UK. ADC Pi Plus.
<https://www.abelectronics.co.uk/products/17/Raspberry-Pi-Raspberry-Pi-2-Model-B/56/ADC-Pi-Plus—Raspberry-Pi-Analogue-to-Digital-converter> (Besökt 2015-06-03). (2014).
- [17] MicroChip. MicroChip MCP3422/3/4.
<http://ww1.microchip.com/downloads/en/DeviceDoc/22088c.pdf> (Besökt 2015-06-21). (2009).
- [18] SK Pang electronics Ltd. PIKAN CAN-Bus Boards for Raspberry Pi.
<http://skpang.co.uk/catalog/pican-canbus-board-for-raspberry-pi-p-1196.html> (Besökt 2015-06-04). (2015).
- [19] L. Fried. Adafruit 5" and 7" 800x480 TFT HDMI Backpack.
<https://learn.adafruit.com/downloads/pdf/adafruit-5-800x480-tft-hdmi-monitor-touchscreen-backpack.pdf> (Besökt 2015-06-07). (2015).
- [20] INTERLINK ELECTRONICS. FSR (R) 400 Series Data Sheet.
http://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf (*Besökt* 2015 – 06 – 07). (2015).
- [21] SHARP DEVICES EUROPE. GP1A57HRJ00F.
<http://www.sharpsme.com/download/gp1a57hr-epdf> (Besökt 2015-06-07). (2005).