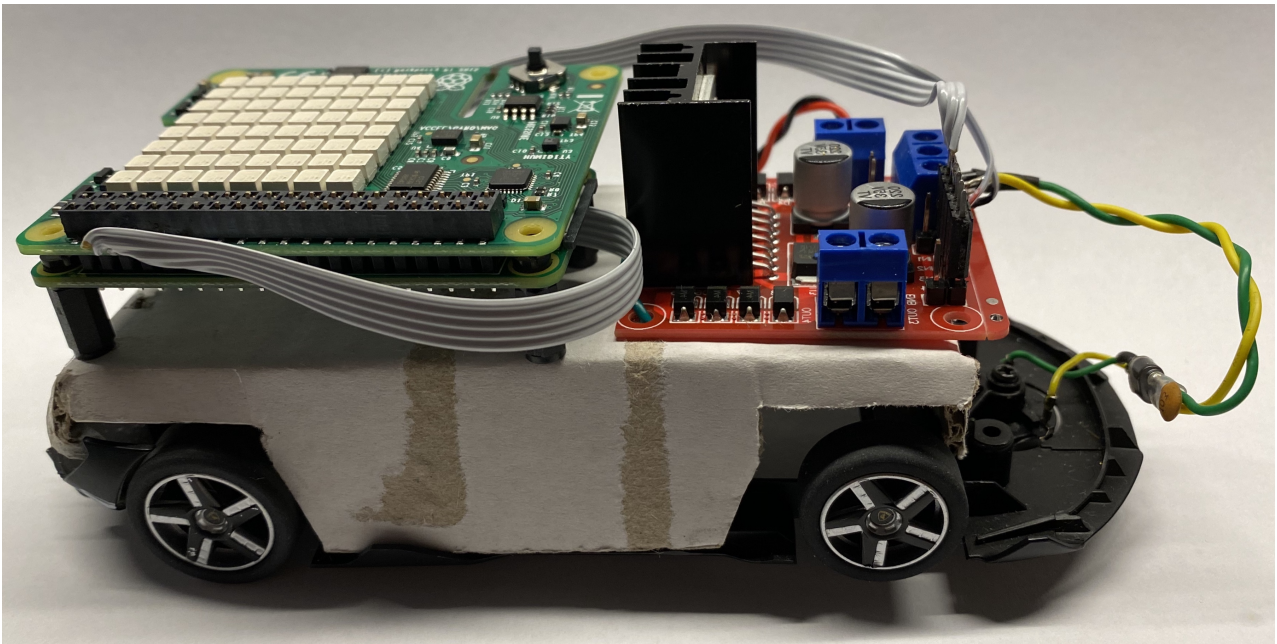




CHALMERS



AI på spårbilbana

MASKINEN MOT MÄNNISKAN

Examensarbete inom Data- och Informationsteknik

Max Petersson
Oscar Thureborn

EXAMENSARBETE

AI på spårbana
MASKINEN MOT MÄNNISKAN

Max Petersson
Oscar Thureborn

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2020

AI på spårbilbana
Maskinen mot människan
MAX PETERSSON
OSCAR THUREBORN

© MAX PETERSSON, OSCAR THUREBORN, 2020.

Examinator: Jonas Almström Duregård
Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola / Göteborgs Universitet
SE-412 96 Göteborg
Telefon: 031-772 1000

The Author grants Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Omslag: Omslaget visar en bild på bilen efter att konstruktionen vad färdig.

Institutionen för Data- och Informationsteknik
Gothenburg, Sweden 2020

Sammanfattning

Artificiell Intelligens blir allt vanligare för att övervaka prestanda hos system. Ofta skickas data från flera system till en molntjänst som sedan behandlar den. Desto fler system som kopplas upp, desto mer data skickas. För att undvika att överbelasta uppkopplingar hos system kan istället datan behandlas direkt på plats.

Ekkono har utvecklat ett koncept som de kallar "AI on the edge". Detta innebär att maskininlärning sker direkt hos det system som ska tränas istället för att behöva skicka data till en molntjänst eller liknande. Cybercom, som är ett IT-konsultbolag, vill lära sig mer om hur Ekkonos SDK (Software Development Kit) fungerar. Studenterna kom tillsammans med Cybercom och Ekkono fram till idén om skapa ett sk. "showcase" som kan användas för att öka intresset för deras koncept. Detta realiserades genom att modifiera en bil från en spårbilbana till att bli maskinstyrd och med hjälp av Ekkonos SDK ta bättre och/eller snabbare beslut än människan. Det huvudsakliga syftet med projektet var således att undersöka om den maskinstyrda bilen kunde köra runt en bana snabbare än en bil styrd av en människa.

Bilen utrustades med ett accelerometer- och motorstyrningskort samt en Raspberry Pi Zero W, en liten 1GHz enkortsdator som kör ett linuxbaserat operativsystem. På Raspberry Pi Zero W installerades sedan Ekkonos SDK som användes med accelerometern för att prediktera de krafter som skulle komma att påverka bilen och därefter bestämma hur mycket gaspådrag bilen skulle få i nästa steg.

Bilen lyckades köra runt banan med hjälp av Ekkonos AI men inte snabbare än vad en människa kunde göra. Detta berodde i stor del på tidsbegränsningar. Konstruktionen av bilen tog mer tid än väntat vilket resulterade i att den tid som kunde läggas ner på att implementera en väl fungerande AI-modell inte räckte till. Dessutom så finns det mycket att önska hos beslutsfattningsprocessen, den processen som tar det predikterade värdet och fattar ett beslut baserat på det. En större utmaning än väntat var också att inte designa programmet på ett sådant sätt att det blir väldigt skraddarsytt för en viss ban-design men inte fungerar på andra typer av banor. AI-modellen som användes var Random forest, vilket är en bra utgångspunkt då den är snabb att ställa upp utan att behöva vikta in-parametrarna. Hade det funnits mer tid till att implementera AI-modellen hade en Multilayer Perceptron modell valts att implementeras istället.

Nyckelord: Artificiell Intelligens, Spårbilbana, Random Forest

Abstract

Artificial Intelligence is rapidly growing in popularity, specially when it comes to monitoring performance in different systems. Many times the gathered data from the system will be transferred to a cloud service for training. The size and count of systems grows rapidly and more and more data will have to be sent. To prevent overloading of network connections the various models can be trained on the spot.

Ekkono has developed a concept they call "AI on the edge". This concept serves to save bandwidth and open up for even more systems and processes can be monitored and/or adjusted with AI. Cybercom, a IT consultant company and business partner with Ekkono, wanted to learn more about how Ekkono's SDK (Software Development Kit) works. The students, Ekkono and Cybercom came up with an idea to create a showcase that can be used to increase knowledge and interest for this specific concept. This was carried out by modifying one of the cars in a slot car track to become machine-controlled and with help of the Ekkono SDK take better and/or faster decisions than a human. Therefore the main purpose of the project was to examine whether the machine-controlled car could run a lap around the track faster than a car controlled by a human.

The modified car was equipped with an accelerometer, an engine-control card and a Raspberry Pi Zero W, a small 1 Ghz computer which runs a linux based operative system. The Ekkono SDK was installed on the Raspberry Pi Zero W and with help from the accelerometer the AI was set to predict the forces the car will experience in the future and therefore decide how the speed be adjusted in the next step.

The modified car was successful in completing laps with assistance of the Ekkono AI, however it was not faster than the human. This depended mainly because of the time limitations of the projects. The construction of the car took more time than expected. This resulted in less time to adjust the AI model and perfect the decision-making of the program. Furthermore the students did not want to tailor make a solution to fit just one type of track layout, so keeping the program general posed an even bigger challenge than expected.

The specific AI model that was used is called Random Forest, which serves as a good starting model considering it's easy to set up and no need for parameter-weighting. Had there been more time a stronger choice in model would probably have been a MLP (Multilayer Perceptron Model).

Keywords: AI, Slot car track, Random Forest

Förord

Detta examensarbete omfattar 15 hp och skrevs för institutionen för datateknik på Chalmers tekniska högskola. Arbetet utfördes av Max Petersson och Oscar Thureborn som studerande på Datateknik programmet.

Detta projektet knyter ihop tre intressenter. Cybercom, som är värd för projektet önskar att öka sin kunskap om diverse mjukvaror deras samarbetspartner Ekkono levererar. Ekkono vill ha fler projekt och produkter som ämnar att visa upp deras mjukvaror och modeller. Studenterna, som representerar Chalmers, vill lära sig mer om alla de beståndsdelar som behövs för att projektet ska fungera: AI, mjukvaruutveckling och elektroniken hos inbyggda system.

Supporten som studenterna har upplevt, från de personer som varit inblandade, har varit utomordentlig och vi vill specifikt nämna följande:

Vi vill tacka Henrik Lundqvist för möjligheten att genomföra examensarbetet på Cybercom och den hjälp som han har bidragit med.

Vi vill också tacka Henrik Linusson från Ekkono för hjälpen att ställa upp AI-modellen och möjligheten att arbeta med deras SDK.

Slutligen vill vi tacka Torbjörn Tjellén från Chalmers som inte bara har agerat projekthandledare utan även mentor och livräddare inom de elektro-relaterade utmaningar som uppstått.

Innehåll

1	Introduktion	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Mål	2
1.4	Avgränsningar	2
2	Teori	2
2.1	Artificiell intelligens	2
2.2	Acceleration, rörelsemängd och Newtons andra lag	3
2.3	Ledningsburen och emitterad störning	4
2.4	Spårstyrda bilbanor	4
2.5	Raspberry Pi Zero W	4
2.6	Raspberry Pi Sense HAT	5
2.7	L298 H-brygga	6
3	Metod	7
3.1	Val av hårdvara	7
3.2	Val av strömförsörjning	8
3.3	Metoder för test av utrustning	9
3.4	Val och implementering av AI-modell	10
4	Genomförande	11
4.1	Mätning av bilbanans elektriska delar	11
4.2	Installation av mjukvara och utvecklingsmiljö	12
4.3	Undersökning av tidsfördröjningar	13
4.4	Undersökning av viktpåverkan	14
4.5	Testkoppling av h-bryggan	15
4.6	Bilens konstruktion	16
4.7	Programvarans uppbyggnad	18
5	Resultat	19
5.1	Mätningar på systemets elektroniska delar	19
5.2	Massans påverkan av bilens prestanda	20
5.3	Tidsfördröjningar i systemet	21
5.4	Avläsning av accelerometervärden under körning	22
5.5	AI-modellens prediktionsförmåga	23
5.6	Maskinen mot människan	24
6	Diskussion	25
6.1	Maskinen mot människan	25
6.2	Utrustningen	26
6.3	Val av givare och axlar	27
6.4	Elektriska störningar och mätningar	27
6.5	Massans påverkan	27
6.6	Etik och hållbarhet	28
7	Slutsats	29
A	Datablad för L289 H-Brygga	30

Förkortningar

AI (Artificiell intelligens) - Imitering av naturlig intelligens.

API (Application Programming Interface) - Ett gränssnitt för utvecklare mot en viss produkt/tjänst.

CSV (Comma-Separated Values) - Ett vanligt filformat för data. Filer som enkelt kan öppnas i kalkylprogram (t. ex Excel, Numbers) där datan kan presenteras i tabeller eller diagram.

EMI (Electromagnetic Interference) - En benämning på den störningen som kan uppstå av elektromagnetism.

GPIO (General Purpose Input Output) - Generell benämning på de kontakter som finns på Raspberry Pi Zero W som sörjer för ihopkoppling av externa komponenter, både för in- och utdata.

LED (Light-emitting diode) - En elektronisk komponent av typen diod, som dessutom lyser.

MAE (Mean Absolute Error) - Ett måttetal som ofta används för jämförelse och kontroll av en prediktion. Skillnaden mellan det förutspådda värdet och det faktiska värdet.

PWM (Pulsbreddsmodulering / Pulse Width Modulation) - En metod som tillåter att styra effektmatning genom att slå på och av spänningen i pulser.

PH (Physics Handbook) - Physics Handbook är en bok innehållande samlingar av fysikaliska formler och tabeller. Boken förekommer i flertalet referenser och kommer därför att förkortas till akronymen PH.

RP-Zero (Raspberry Pi Zero W) - En enkorts dator från tillverkarna Raspberry Pi med namnet Zero W.

Sense HAT (Raspberry Pi Sense HAT) - Ett kretskort med flertalet ställdon och givare på. Tillverkarna heter Raspberry Pi och kortet Sense HAT.

SDK (Software Development Kit) - Ett "kit" eller paket av utvecklingsverktyg ämnat åt mjukvaruutvecklare.

SSH (Secure Shell) - Ett kryptografiskt nätverksprotokoll för operativsystem. Möjliggör koppling mellan en dator och en annan över t. ex wifi.

1 Introduktion

Artificiell Intelligens är ett av de snabbast växande och mest eftertraktade områdena inom forskning och har blivit allt vanligare att applicera inom industrin, exempelvis som övervakning av ett system. Ekkono är ett mjukvaruutvecklingsföretag som har utvecklat ett AI koncept och efterfrågar ett showcase som kan visa upp det.

Genom att applicera Ekkonos AI på en spårstyrd modellbil från en bilbana skall denna kunna göras självkörande. Bilen styrd av AI kan sedan användas för att tävla mot en bil styrd av en människa, vilket skulle vara ett interaktivt sätt att visa upp deras koncept.

För att genomföra detta skall den befintliga modellbilen utrustas med en enkortsdator innehållande Ekkonos AI samt övrig elektronisk utrustning som behövs för att göra bilen självkörande. Arbetet kring detta samt appliceringen av Ekkonos AI är dokumenterat i denna rapport.

1.1 Bakgrund

En människas förmåga att fatta beslut och lära sig av tidigare erfarenheter från omvärlden till den grad som oss är något som under en lång tid har varit unikt. Enda sedan andra världskrigets slut har människan försökt efterhärma detta, skapa en så kallad Artificiell Intelligens. Även om AI idag inte har nått långt nog för att helt återskapa eller simulera en människas intelligens så finns områden där den utför prediktioner som de flesta människor hade varit okapabla att utföra. AI blir allt vanligare för att optimera och övervaka prestanda hos system. Ofta kan stora mängder eller känslig mätdata behöva skickas till en server som använder sig av datan i en AI-modell.

Ekkono är ett mjukvaruutvecklings företag vars affärsidé bygger på ett SDK som ämnar att åstadkomma "AI on the edge". Det konceptet innebär att maskininlärning, som är en underkategori till AI, ska ske ute hos komponenterna. Tanken är att fler och fler maskiner och komponenter ska tränas med AI. För att detta ska gå så kan inte all data som alla dessa komponenter skapar och behöver laddas upp och ned från en molntjänst, för då hade uppkopplingarna blivit överbelastade. Vissa maskiner och komponenter kommer helt enkelt behöva läras upp på plats utan uppkoppling. Det kan även finnas säkerhetsaspekter där säkerhetskänslig data kan läcka genom att den skulle skickas till en molntjänst. Att utföra beräkningarna direkt hos enheten skulle då öka säkerhetsaspekten. En ytterligare anledning med att undvika skicka data från flera källor till en AI-modell är att varje källa lärs upp av dess egna omgivning. Detta kan ge ett mer individuellt optimerat system än ett generellt system. Ekkono efterfrågar ett showcase där detta SDK har använts.

Tanken är att göra detta med en spårbilbana. Detta är en typ av modellbilbana där två el-bilar tävlar mot varandra på strömförda spår. Strömmen ger kraft till en elmotor som driver bilen framåt längst spåret. De som tävlar kan reglera sin bils hastighet med hjälp av en handhållen dosa. Skulle man ge för mycket gaspådrag i en kurva så riskerar man att bilen trillar ur sitt spår och kraschar. Om en av bilarna styrdes med hjälp av Ekkonos mjukvara, medans den andra bilen fortsatt körs manuellt, skulle detta tillåta människan att tävla mot AI för att se om de kan besegra den.

Cybercom, som har ett samarbete med Ekkono, tog fram detta uppdraget tillsammans med studenterna. Cybercoms intresse ligger vid att lära sig hur Ekkonos SDK fungerar och vad som krävs för att applicera detta i ett framtida projekt.

1.2 Syfte

Syftet med projektet är att undersöka om en bil styrd av Ekkonos AI kan optimera körningen på ett sådant sätt att den kör bättre än en människa. En modell där detta uppfylls skulle ge Ekkono ett showcase som skulle dra intresse till deras produkt då potentiella kunder kan få en direkt förståelse för optimeringsmöjligheterna med AI. Det skulle också visa hur litet ett AI system kan vara eftersom all hårdvara skulle få plats på en modellbil. Detta inkluderar den utrustning som utför prediktionerna hos AI-modellen, som annars hade tagit upp plats i form av en server hos en molnbaserad tjänst.

Bilen ska kunna köras utan mänsklig interaktion, då måste en elektrisk krets konstrueras för att kunna styra bilens hastighet. Detta skall skötas med hjälp av en styrsignal, som beräknas utifrån prediktioner från Ekkonos AI.

Bilen skall konstrueras på ett sådant sätt att den utrustning som sitter på den inte skall utsättas för elektriska störningar som hindrar bilen från att köra eller kan utsätta omgivningen för störningar.

De komponenter som kommer behöva monteras på bilen för att kunna göra den självkörande kommer addera extra vikt. Eftersom bilen inte är konstruerad för att köra med extra vikt bör en undersökning av massans påverkan hos bilens prestanda genomföras för att se om bilen kan köra med den extra vikten. Detta skulle även behöva undersökas för att kunna jämföra bilens prestanda körd av en människa mot den styrd av AI-modellen.

AI-modellens förmåga att prediktera ett värde ökar med mängden datapunkter som den kan basera sitt beslut på. I detta fall kommer beslutet att baseras på tidigare mätningar hos bilens sensorer. Eftersom bilens hastighet måste kunna ändras inom loppet av hundradelars sekunder för att inte åka av banan måste frekvensen för hur ofta mätdata kan samlas in vara tillräckligt hög. Ett test skall genomföras hos de givare som kommer att användas för datainsamling för att ta reda på mätningens frekvens.

1.3 Mål

Det huvudsakliga målet för projektet är att modifiera en spårstyrd bil så att den köra av sig självt med hjälp av Ekkonos AI utan att veta hur bilbanan ser ut i förväg. AI modellen skall kunna köra bilen utan att åka av banan och skall göra det snabbare än en människa kan klara av det.

Utöver detta har projektet ytterligare fyra delmål för att kunna uppnå huvudmålet och undersöka syftet:

- Skapa en elektrisk krets som tillåter en dator att styra bilen.
- Förhindra att bilen utsätts eller utsätter omgivningen för störningar.
- Undersök hur vikten påverkar bilens varvtid.
- Undersök om den frekvens som data kan samlas in från givarna är tillräckligt hög för Ekkonos AI-modell.

1.4 Avgränsningar

Projektets fokus kommer att vara själva implementationen av Ekkonos AI-modell. I projektet kommer därför inte studenterna att utveckla någon egen AI-modell utan Ekkonos färdiga AI-modell kommer att användas.

Konstruktionen av den självkörande bilen kommer utgå från redan tillgängliga elektroniska komponenter. Ingen egen bilbana eller bil kommer att konstrueras från grunden. Istället kommer befintliga modeller som finns tillgängliga på marknaden att modifieras.

För att genomföra projektet kommer lämplig mätdata behöva samlas in, vilket i sin tur kräver instrument som kan mäta dessa. Som en begränsning för att spara tid i projektet kommer en Sense HAT, ett kretskort med ett urval av färdigt integrerade mätinstrument, användas istället för individuella mätinstrument som hade behövt kopplas och integreras. Strömförsörjningen till den enkorts dator och kretskortslogik som används i projektet kommer att tas från en powerbank. Den skulle kunna tas från bilbanans spår, men powerbanken tillåter bilen att köra av banan utan att förlora strömmen till enkorts datorn vilket kan vara skadligt för denna. Slutkonstruktionen av bilen, som modifieras, kommer att likna en prototyp av potentiell produkt. Både strömförsörjningen och valet av sensor bidrar till att alla komponenter inte kommer att få plats inuti den befintliga plast-karossen. Vissa av dessa avgränsningarna kommer i mer eller mindre utsträckning påverka bilens köregenskaper, huvudsakligen sett till vikt. Vidare så kommer inte själva bilbanan att modifieras inom projektets omfattning. Här skulle den kontroll som hör till spåret som bilen använder sig av förbi-kopplas så att det alltid ger maximal spänning, men för att spara tid så kräver prototypen att detta handtag helt enkelt trycks in.

2 Teori

Processen att konstruera en självkörande spårstyrd bil underlättas med vissa förkunskaper. Denna del av rapporten är konstruerad för att belysa det som inom projektet anses viktig kunskap. Detta innefattar grundläggande teoretisk kunskap samt informativ teori kring projektets utrustning.

Då projektet syftar till att applicera ett färdigt AI-API ges en kort förklaring av vad AI är samt algoritmen Random forest, vilket var den som användes i projektet. Utrustningen som monteras på bilen kommer att öka dess vikt, och därmed påverka dess prestanda. Därav förklaras det teoretiska sambandet mellan massan hos ett föremål och dess förmåga att accelerera eller ändra rörelsetillstånd. Vid konstruktionen av en elektrisk krets, som den i detta projekt, kan störningar uppstå inom det egna systemet eller störningar som påverkar andra runtomkringliggande system. Det är därför viktigt att förstå hur de uppstår och hur de motverkas. Slutligen presenteras informativ teori kring den utrustning som användes i projektet. Bilbanan, enkorts datorn, sensor kortet och motorstyrningen.

2.1 Artificiell intelligens

Artificiell intelligens, även kallat AI, är ett växande och populärt fält inom forskning. Arbetet kring området startade strax efter andra världskriget och begreppet myntades år 1956 [1]. Artificiell Intelligens hos datorprogram handlar om att efterlikna naturlig intelligens, som förmågan att lära sig av tidigare erfarenheter, uppfatta

språk, känna igen objekt i bilder. AI omfattar ett stort område och finns idag i stor utbredning runtomkring oss.

En typ av AI är den som försöker efterlikna oss människor. Sophia, en robot skapad av Hanson Robotics, använder sig av en sådan typ av AI. Den tillåter henne att känna igen både tal och ansiktsuttryck samt föra en konversation med en människa [2]. AI måste inte försöka efterlikna mänsklig intelligens på alla nivåer. Det kan handla om att efterlikna ett intelligent beteende. Exempelvis att köra en bil eller dra slutsatser och utföra prediktioner. I detta projekt kommer AI användas för att uppskatta förändringar i en accelerometer innan de händer. Dessa prediktioner kommer sedan användas för att styra bilens motor. Om bilens AI förväntar sig värden som indikerar att bilen riskerar att åka av kommer bilen att bromsa, eller gasa mer om den inte ser någon risk att åka av.

2.1.1 Random forest

Random forest är en trädbaserad maskininlärningsalgoritm som kan användas för att göra prediktioner, vilket kan användas för att styra bilens motor. Först skapas ett dataset med variabler över det man vill prediktera. Datasetet består av de mätbara värden man tror kan påverka det som ska predikteras. Om man exempelvis skulle vilja uppskatta accelerationen av bilen i en kurva kan man ta med mätvärden från accelerometern och gaspådraget som motorn får. Detta dataset fylls sedan på med uppmätt data för att skapa en tabell.

Utav detta dataset skapar algoritmen sedan ett nytt dataset genom att slumpmässigt plocka ut rader ur tabellen, här tillåts dubletter att plockas ut. Eftersom dubletter tillåts kommer detta göra att vissa värden uteblir, dessa kan senare användas för att testa prediktionsförmågan hos algoritmen då utgången redan är känd. Det nya datasetet används sedan för att skapa slumpmässiga beslutsträd, varifrån namnet Random Forest kommer. Desto fler träd som skapas desto bättre blir algoritmen på att prediktera.

När Random Forest algoritmen gör en prediktion baseras detta på ett majoritetsbeslut. Ingångsdaten för det värde som vill predikteras körs genom alla träd och därefter väljs den prediktion som stämmer överens med flest träd. Algoritmens precision kan nu testas på den data som uteblev under den slumpmässiga val processen eftersom resultatet redan är känt. Om algoritmen skulle genomföra dåliga uppskattningar så kan man skapa en ny skog av träd, där man till exempel varierar antalet grenar från varje nod, skapar fler träd eller ger modellen mer ingångsdata.

2.2 Acceleration, rörelsemängd och Newtons andra lag

Vikten hos en bil har betydelse för dess prestanda. I tillverkning av Formel 1 bilar är deras mål bland annat att skapa en bil som har så mycket kraft som möjligt per viktenhet, detta gör att bilen kan köra snabbare runt banan. Vikten har alltså en stor betydelse för prestandan hos en bil inom tävlingar, men det kan också påverka bilens förmåga att köra över huvud taget. Om en bil skulle lastas med för mycket vikt skulle tillslut motorn inte orka motverka de friktionskrafter som verkar på bilen. En vanlig bil kan till exempel inte dra ett fullastat lastbilssekipage. Men vad är det egentligen som gör att en bil som väger mer kör långsammare runt en bana? En förklaring till detta är hur massan av ett objekt påverkar dess acceleration.

Acceleration av ett föremål beskrivs genom dess ändring i hastighet över tid, se ekvation (1). Eftersom hastighet uttrycks som en vektor, med både storhet och riktning, så gör även acceleration det. Acceleration som verkar i samma vektor som ett föremåls färdvektor leder till ökad hastighet. Acceleration i motsatt riktning leder till minskad hastighet, även känt som deceleration eller retardation. Acceleration som verkar i en annan vektor än ett föremåls hastighetsvektor kommer att kunna ändra både riktningen och farten på föremålet, även kallat dess rörelsetillstånd.

$$a = \frac{dv}{dt} \quad (1)$$

Hur lätt ett föremål ändrar rörelsetillstånd beror på dess rörelsemängd [3]. Enligt ekvation (2), från PH, beror rörelsemängden på föremålets hastighet och massa. Om ett föremål har en stor massa krävs en stor kraft för att väsentligt ändra dess rörelsetillstånd, samma gäller ett föremål med stor hastighet.

$$p = mv \quad (2)$$

Newtons andra lag [3], se ekvation (3) enligt PH, säger att den kraft som verkar på ett föremål är proportionerlig med ändringen av dess rörelsemängd över tid.

$$F = \frac{dp}{dt} \quad (3)$$

Genom ekvation (1), (2) och (3) samt förutsättningen att massan hos ett objekt är konstant kan vi lösa ut ekvation (4). Kraften som verkar på ett föremål står i proportion till dess massa samt acceleration. Detta innebär även att accelerationen hos ett objekt är beroende av kraften som verkar på objektet samt dess massa.

$$F = \frac{dp}{dt} = \frac{d(mv)}{dt} = m \frac{dv}{dt} = ma$$

$$F = ma \quad (4)$$

2.3 Ledningsburen och emitterad störning

De störningar som behandlas i projektet är ledningsburna eller emitterade. Ett system kan bli stört på olika sätt. Om två ledare, tillhörande systemet, bildar en area kan dessa vara mottagliga för emitterade elektromagnetiska störningar. Ett exempel på ledningsburen störning i detta projektet är den spänning som kan uppstå i motorn och skickas tillbaka i systemet och skada eventuella komponenter. EMI-filer kan användas för att skydda system mot detta.

2.4 Spårstyrda bilbanor

Projektet syftar till att undersöka om Ekkonos AI kan optimera körningen hos en modellbil så att den kör bättre än en människa. För att förenkla antalet variabler i styrningen valdes detta att utföras på en spårstyrd bilbana då den enda variabeln som styr bilen är gaspådraget som ges till bilens motor. Bilen åker längst ett ledspår likt en spårvagn och drivs av en elektrisk motor som matas med likström över ett ledspåren. Bilarna styrs av en handkontroll som ökar eller minskar spänningen över motorn med hjälp av en potentiometer, som sitter i handkontrollen. En potentiometer är ett varierbart motstånd som i detta fallet är kopplat i serie med motorn. Resistansen i potentiometern omvandlas till värme, desto högre resistans desto mer energi omvandlas till värme. Detta resulterar i ett spänningsfall till motorn vilket kan användas till att öka/minska hastigheten av bilen genom att öka/minska motståndet. Potentiometern är kontrollerad av en "avtryckare" på handkontrollen, desto mer den trycks in desto snabbare kör bilen. Det finns en handkontroll för varje spår och de är direktkopplade via sladd till spårbilbanan.

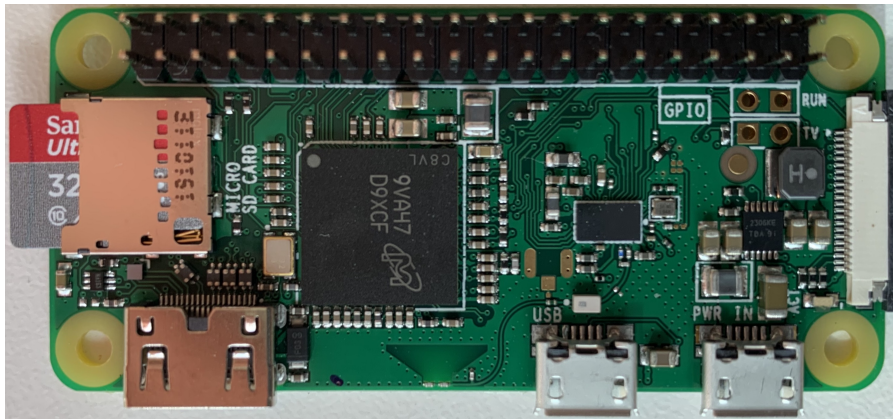
På undersidan av bilen sitter en magnet som attraheras av bilbanans ledspår. Detta gör att bilen får bättre grepp på banan och inte sladdar runt i kurvor. Bilarna är skalenliga och kan vara av olika storlek, vanliga skalor är 1:24 eller 1:32. Det finns flera tillverkare av spårstyrda bilar och bilbanor varav Scaletrix är en av de vanligaste. Bilbanan som användes i detta arbetet var ett standardsetet tillverkad av Scaletrix i skalan 1:32. Bilbanan har två spår där bilar kan åka på och kan konfigureras till fyra olika banor. Settet inkluderar två handkontroller samt två bilar. Figur 1 visar hur en sektion av bilbanan och bilarna ser ut.



Figur 1: Bilarna på bilbanan.

2.5 Raspberry Pi Zero W

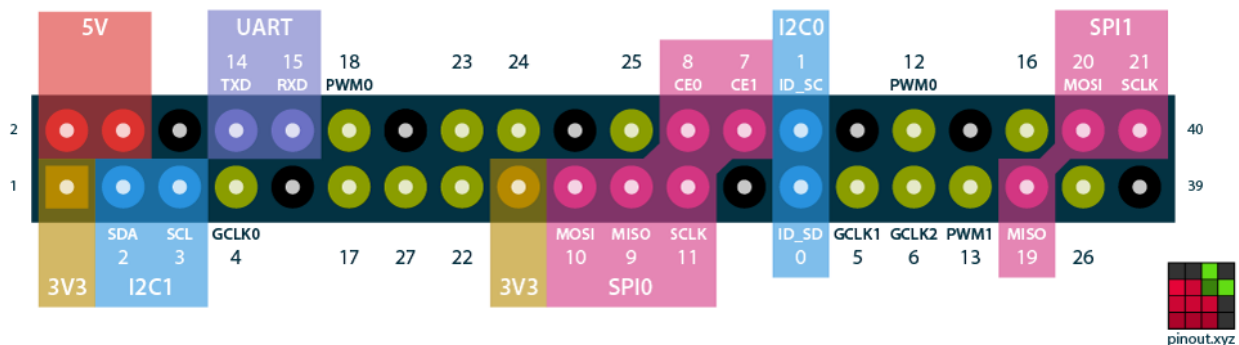
Raspberry Pi Zero W är en enkelkärning enkorts dator som lanserades 2017 som en utbyggnad av Rasperry Pi Zero [4]. Likt Rasperry Pi Zero har den en 1 GHz enkelkärning processor, 512MB RAM-minne, mini HDMI och USB portar, en 40-pinnars header och en mini USB-port tillämdad strömförsörjning. RP-Zero W modellen har utöver detta möjligheten att kommunicera via Wifi och Bluetooth. Detta kan användas för att köra teminalen på Rasperry via en annan enhet, exempelvis en bärbar dator. Operativsystemet som körs på RP-Zero W är Raspbian, ett linuxbaserat operativsystem som är utvecklat av Raspberry Pi. Operativsystemet är installerat direkt på ett micro SD-kort som sedan används som Raspberry Pi:ens disk. Figur 2 visar hur en Raspberry Pi Zero W ser ut.



Figur 2: Figuren visar en bild tagen ovanifrån på en Raspberry Pi Zero W.

RP-Zero W har som ovan nämndes en 40-pinnars header som kan användas för att kommunicera med omvärlden, se figur 3. De gröna pinnarna är GPIO-pinnar och de svarta är jord. Men vissa pinnar tillåter som figuren visar även kommunikation via flertalet olika protokoll, men kan också användas som GPIO-pinnar. För projektet behövde motorstyrningen logikspänning, två styrsignaler, en PWM-signal och jord för att sluta kretsen. Från pinne 4 plockades 5V logikspänning, 6 jord, 8 & 10 användes för styrsignaler och 12 för att generera en PWM-signal. Kommunikationen med projektets givare sker via I2C-protokoll på pinne 3 & 5.

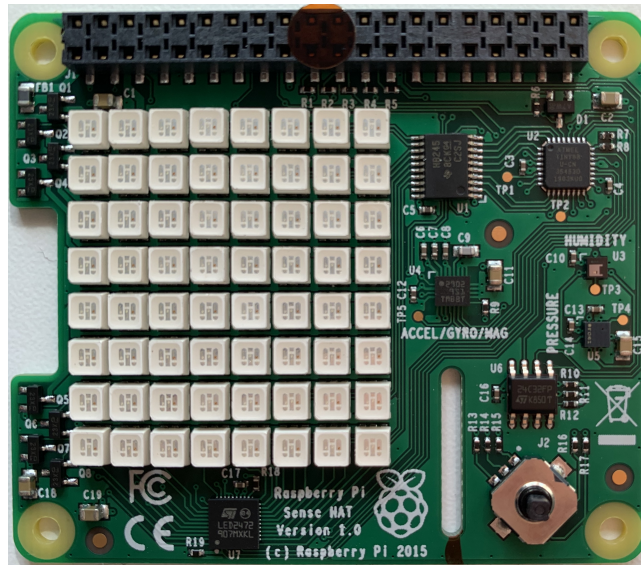
Raspberry Pi GPIO BCM numbering



Figur 3: En GPIO karta för en 40-pinnars Raspberry Pi modell. Källa: hämtad från Pinout[5].

2.6 Raspberry Pi Sense HAT

Raspberry Pi Sense HAT, se figur 4 är ett chip som innehåller flertalet olika sensorer. De givarna som berör projektet är gyrometer, accelerometer och kompass. Sense HAT fungerar tillsammans med RP-Zero, eftersom de båda är framtagna av samma tillverkare. Raspberry Pi har färdiga APIer som gör att de olika givarna är lättåtkomliga direkt efter installation.

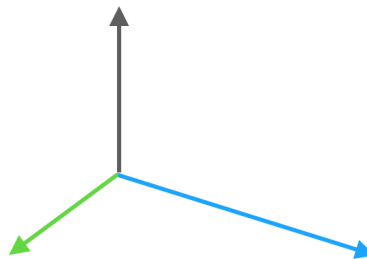


Figur 4: Figuren visar en bild tagen ovanifrån på en Raspberry Pi Sense HAT.

2.6.1 Accelerometers axlar

De olika axlarna som accelerometern mäter på benämns som Roll, Pitch och Yaw. Figur 5 visualiserar axlarna.

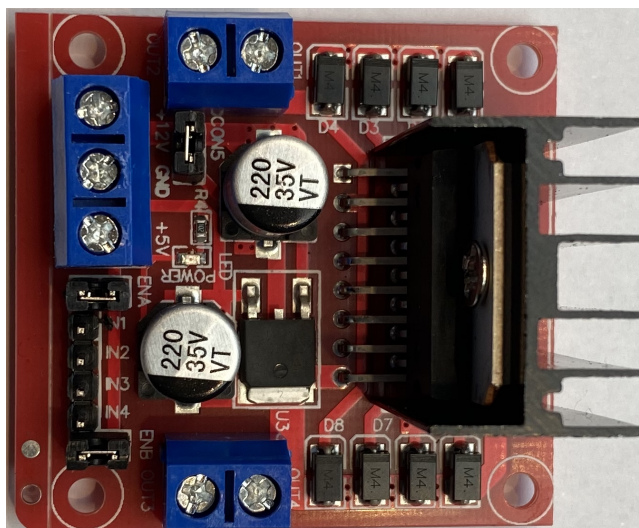
- Roll: Rotation runt blå axel. Uppstår om bilen skulle rulla runt.
- Pitch: Rotation runt grön axel. Uppstår om bilen skulle köra i upp- eller nedförsbacke.
- Yaw: Rotation runt grå axel. Uppstår om bilen svänger höger eller vänster.



Figur 5: De olika accelerometer-axlarna. Blå är bilens körriktning.

2.7 L298 H-brygga

L298 är ett kretskort som kan styra en motor med hjälp av en h-brygga, se figur 6. En h-brygga är en krets som tillåter att ändra polariteten på en matningsspänning över en elektronisk komponent. Detta är mycket användbart för elmotorer då detta möjliggör att byta rotationsriktning, bromsa eller frikoppla dem med styrsignaler. H-bryggan L298 tillåter en matningsspänning mellan 5-35V som kan styras med hjälp av en PWM-signal. Om matningsspänningen är under 12V så finns det en möjlighet att använda en 5V utgång från kortet, vilket exempelvis möjliggör att driva en mikrocontroller. Databladet för L298 finns tillgänglig i appendix A.



Figur 6: Figuren visar en bild tagen ovanifrån på en L298 H-brygga.

3 Metod

Projektets arbetsgång har kommit att beröra många områden. Val av utrustning, testning av utrustningen samt arbetet kring Ekkonos SDK som innefattade val, implementering och utvärdering av AI-algoritmen. Informationssökning för AI och krets-teori skedde utefter rekommendationer av projektets handledare. Informationen hämtades från böcker, databaser och sökmotorer.

I projektets inledande stadie hölls ett möte med Ekkono med syftet att visa möjligheterna och funktionaliteten av deras SDK - detta möte gav också en startpunkt för vilken typ av utrustning som skulle komma att behövas för att implementera AI-modellen. Bilbanan behövde undersökas för att ge en utgångspunkt för val vid resterande konstruktionen av bilen, som val av morstyrning. Störningar behövde också mätas upp så att de skulle kunna åtgärdas för att skapa ett säkert och stabilt system.

När all utrustning var införskaffad bestämdes vilka tester som skulle bli nödvändiga att utföra för att nå syftet med projektet. Bilen testades för att se om den skulle klara vikten av utrustningen och accelerometers prestanda undersöktes. Implementeringen av AI-modellen skedde utefter Ekkonos rekommendationer med Random forest som utgångspunkt.

3.1 Val av hårdvara

För att göra en spårstyrd bil självkörande med hjälp av AI krävs att den utrustas med hårdvara. En dator behövs för att köra AI-algoritmen, en uppsättning givare för att samla data till algoritmen och någon form av motorstyrning. En del av arbetsgången var att välja dessa komponenter. Valen baserades kring många aspekter. Storleken och vikten är två av de viktigaste aspekterna att ta hänsyn till, men att minska storleken av komponenter kan komma med andra utmaningar. Tidsåtgång, kunskapsområden, begränsningar och rekommendationer sätter restriktioner på urval av komponenter. Denna del ger en förklaring kring varför komponenter valdes samt hur de bidrar till resultatet av en självkörande bil.

3.1.1 Raspberry Pi Zero W

Ekkonos SDK har API skrivna för både C och Python, men möjligheten för handledning kring Ekkonos API var för oss begränsat till Python. Därav gjordes valet att den dator som skulle användas för projektet måste kunna köra Pythonskript. Storleken på bilen innebar också att storleken som datorn kunde ta upp blev begränsad, desto mindre desto bättre. Det finns flertalet passande enkortsdatorer som hade klarat detta jobbet. Av dessa har Ekkono utfört en hel del test kring enkortsdatorer från Raspberry Pi serien i tidigare projekt. Valet av dator gjordes därför från denna serie.

Den minsta enkortsdatorn från Raspberry Pi serien är Raspberry Pi Zero. Det bekräftades av Ekkono att RP-Zero har en processor som är kraftfull nog att utföra de nödvändiga beräkningarna för projektets ändamål och den får plats i bilen. RP-Zero har en Wifi-modell (RP-Zero W) vilket möjliggör kommunikation mellan RP-Zero W och en utvecklingsdator via SSH över Wifi-anslutning. Detta underlättar vid utveckling, debugging och testning. Därför valdes denna variant av RP-Zero.

3.1.2 Raspberry Pi Sense HAT

Som en utgångspunkt i projektet valdes Raspberry Pi Sense HAT som källan för givare/sensorer. Kortet är utrustat med både accelerometer, gyro och kompass vilket alla var intressanta mätvärden vid projektets utgångspunkt. Valet förstärktes av det faktum att Sense HAT är byggd för att användas på en RP-dator med 40-pinnars header, som RP-Zero W, och det finns ett färdigskrivna API att använda. De aspekter som talar emot Sense HAT som sensorchip är dess storlek och vikt. Sensorerna av intresse tar endast upp cirka 1 cm² av kortets 36 cm² i yta.

Det finns mindre kort, som endast innehåller en liknande sensor som den av intresse på Sense HAT. Men för att integrera ett sådant kort krävs kunskap som var utanför gruppens område. Att hitta rätt kort är bara början. Många av korten kräver lödning för att fungera och ett API måste skrivas för att kommunicera med kortet och samla in data på rätt sätt. För att spara tid till övriga delar av projektet valdes Sense HAT som utgångsläge. Den tekniska informationen som hittades kring Sense HAT var knapphändig. Ingen information kring samplingshastigheten hos sensorerna kunde hittas. Ett test kommer därför behöva genomföras för att ta reda på vilken frekvens data kan läsas från kortet. Frekvensen för dataavläsning kommer att ha betydelse för hur noggrant AI-modellen kan uppskatta ett värde. Skulle samplingshastigheten eller vikten av kortet visa sig vara ett problem kommer kortet att bytas ut till ett mindre kort med enbart accelerometer/gyrometer som har en högre samplingsfrekvens.

3.1.3 Motorstyrning

För att kunna göra bilen självkörande krävs något sätt att styra bilens elektriska motor. En h-brygga bestämdes vara ett passande redskap för att styra motorn med hjälp av styrsignaler genererade av RP-Zero. H-bryggor kan köpas separata eller integrerade i ett kretskort. Den integrerade varianten tillåter användaren att enkelt installera en h-brygga för motorstyrning. För att kunna välja en passande h-brygga för projektet behövdes matningsspänningen över motorn samt den maximala strömförbrukningen mätas.

3.2 Val av strömförsörjning

Flera komponenter i systemet behöver strömförsörjning och detta har varit ett viktigt område för konstruktionen av systemet. Den typen av bilbana projektet bygger på sköter sin strömförsörjning via spåren direkt till bilens motor. Detta grundkoncept ville behållas i den slutgiltiga konstruktionen, därmed undersöktes inga alternativa källor för strömförsörjning till bilens motor. När det kommer till att strömförsörja datorn och logikstyrning av övriga kretskort så fanns det flera saker att ta hänsyn till:

- En omodifierad RP-Zero tar sin ström från dess micro USB port.
- En dator av denna typen kan inte drivas med en strömkälla som har för stort rippel.
- En spänningsspike kan skada systemet.
- Ett spänningsfall kan också skada systemet om man har otur, dessutom så kommer flera omstarter av systemet bli väldigt påfrestande då program ska startas upp på nytt varje gång.
- Den strömförsörjning som monteras på bilen får inte väga för mycket.

Dessa punkterna resulterade i tre olika alternativ:

1. Plocka ut logikspänning (5 V) från bilbanans egna krets (spåret).
2. Använda vanliga batterier, alkaliska eller litium.
3. Använda en sk. powerbank (majoriteten har 5 V som standard).

Det finns för- och nackdelar med de olika sätten. Men det som valdes var nummer #3. Här följer resonemangen: **Vikt:** I viktspekten, som är en viktig aspekt, så är alternativ #1 bäst. För att driva RP-Zero på något annat än micro USB så krävs dock en skyddande krets då man inte, helt säkert, kan mata RP-Zero direkt via dess GPIO-ingångar. Denna kretsens vikt är förhållandesvis låg.

Säkerhet: Säkerhet gäller både skydd av kretsar och komponenter men syftar också på att RP-Zero inte ska stängas av i onödan. Eftersom powerbanken var på 2500 mAh och dessutom uppladdningsbar så blir alternativ #3 en vinnare. Vidare så uppstår inget rippel från powerbanken vilket innebär att RP-Zero är skyddad även mot spänningsspike och spänningsbortfall.

Tid och kunskap: Eftersom strömförsörjningen inte är eller har varit huvudsyftet med projektet så får lösningarna inte ta allt för mycket tid. Saknas kunskapen så tar det dessutom ännu mer tid att installera och/eller anpassa en lösning. I den här aspekten var alternativ #3 överlägset.

Eftersom alternativ #3 var ett så pass bra val i alla aspekter förutom just vikt så valde vi att öppna upp för en ny lösning här - nämligen att placera powerbanken innanför banan. Detta löser problemet med vikten men introducerar så klart ett beroende hos bilen, då den måste vara inkopplad, med sladd, till en strömförsörjning som inte sitter på själva bilen.

3.3 Metoder för test av utrustning

För att konstruera bilen behövdes undersökning och tester av utrustningen genomföras. Exempelvis behövdes mätningar på bilbanan och bilen utföras för att välja en passande motorstyrning. För att säkerställa om syftet kan genomföras måste bilen klara av att köra med den extra vikten från utrustningen. Det måste också säkerställas att sensorer kan leverera data med tillräckligt hög tidsupplösning samt vilken data som kommer att bli intressanta för AI-modellen att uppskatta. Metoden för testerna som genomfördes för att undersöka detta presenteras här.

3.3.1 Undersökning av bilbanan och bilen

För att kunna säkerställa vilken utrustning som behövs för att göra bilen självkörande är det viktigt att undersöka den befintliga konstruktionen hos bilen och bilbanan. Detta ger en insikt kring utgångsläget och vilka aspekter som är viktiga att ta hänsyn till under konstruktionen. Till exempel måste bilens matningsspänning över motorn och den maximala strömförbrukningen mätas upp för att kunna välja en passande motorstyrning. Det är också viktigt att identifiera eventuella ledningsburna eller emitterade störningar som kan påverka systemet, så att de kan åtgärdas.

Mätningarna utfördes med hjälp av ett oscilloskop. Eftersom felkoppling vid mätning kan skada utrustningen och vår erfarenhet kring mätning med ett oscilloskop är ytterst begränsad så genomfördes mätningen i samarbete med handledare.

Då en DC motor roterar kan störningar uppstå, dessa är speciellt märkbara då motorn startar från stillastående läge eller om motorn bromsas under fullt gaspådrag. För att mäta dessa ges fullt gaspådrag till bilen från stillastående läge medans mätningar utfördes över banan, för att se vilka störningar som skickades ut av motorn. Bilen hölls också stilla medans fullt gaspådrag gavs, för att sedan långsamt dras framåt så att motorn snurrade långsamt. Dessa två metoder för att mäta störningarna från motorn på bilen utfördes med olika mätpunkter. Dels mättes störningarna på bilen, på det egna spåret och de emitterade störningar som plockades upp av grannspåret.

För att mäta den maximala strömförbrukningen hos motorn skapas en kortslutning i handkontrollen med ett motstånd på 10Ω , detta motsvarar att koppla in ett motstånd i serie med motorn. Bilen hölls sedan stilla på banan utan att hjulen rörde på sig, detta maximerar strömförbrukningen hos bilens motor. Vid detta skede mättes spänningen över motståndet, vilket användas för att beräkna strömförbrukningen.

3.3.2 Viktens påverkan hos bilens prestanda

Vikten av bilen kommer att ha betydelse för dess prestanda. Med för mycket vikt kan det vara möjligt att bilen inte ens klarar av att köra eller att motorn överhettas och tar skada. En annan möjlighet skulle också vara att bilen blir så långsam att fullt gaspådrag kan ges utan att den kör av banan. I detta fallet finns det inte någon möjlighet att optimera körningen av bilen, och därmed ingen möjlighet att undersöka syftet för arbetet.

Utrustningen som kan komma att monteras på bilen vägs upp med en våg. Detta innefattar RP-Zero, Sense HAT, motorstyrning och alternativ för strömkälla. Bilen utrustas sedan med blyvikter som motsvarar vikten från de kombinationer av utrustning som kan komma att vara möjliga för projektet. Med detta menas elektroniken i kombination med olika tänkbara strömkällor.

För att mäta viktens påverkan hos bilens prestanda kommer bilen att manuellt köras runt banan under tidtagning. För att minska påverkan av felaktig tidtagning, då reaktionstiden hos en människa kommer att påverka tidtagningen, kommer bilen att köras 10 varv (ett heat) runt banan under varje mätning. För att ytterligare minska påverkan av den mänskliga faktorn kommer flera heat köras och mätas för varje vikt som skall undersökas. Utifrån detta kan en snitvarvtid beräknas för varje vikt, denna kan sedan jämföras med en icke modifierad bils snitvarvtid för att se viktens påverkan.

3.3.3 Tidsfördröjningar i systemet

För att kunna optimera körningen hos bilen behöver beslut kring att öka eller minska bilens hastighet tas inom väldigt små tidsmarginaler. Om man släpper gasen för sent i en kurva kommer bilen att köra av banan, om man gasar för sent förlorar man tid. Det är alltså viktigt att veta att systemet inte begränsas av dess förmåga att ändra en styrsignal till motorn. För att AI-modellen ska kunna fatta ett beslut inom den snäva tidsmarginalen så kommer hastigheten som data kan samlas in samt tiden mellan två mätningar att ha betydelse. Den tekniska information som hittades kring RP-Zero W samt Sense HAT innehöll ingen information kring dessa områden.

För att undersöka dessa fördröjningar så behöver de testas. Python-biblioteket "time" innehåller funktioner för att få den exakta tidpunkten vid ett givet tillfälle. Denna funktionen utnyttjades genom att spara en "start"-tid precis innan processen, som skulle testas, genomfördes och sedan en "slut"-tid efter processen var klar. Skillnaden mellan dessa två tidpunkterna är tiden (i sekunder) som en viss process tog. Testen kördes i flera iterationer för att få rättvisande värden. Samtliga följande del-processer behöver kontrolleras:

- Avläsning av givarvärden.
- Skrivning till ställdon (GPIO).
- Den interna processcykeln för Raspberry Pi Zero.

Slutligen utförs en tidtagning för hela programmet, detta genomförs då programmet är färdigställt och bilen kör på banan.

3.3.4 Insamling av Sense HAT data under körning

Syftet med dessa mätningarna var att få reda på vilka givare som skulle komma att behövas som in-värden för AI-modellen. För att modellera den datan som samlas in av givarna utfördes ett antal tester under vanlig körning. Med vanlig körning så menas att bilen styrs av den medföljande handkontrollen samt att bilen är så gott som omodifierad. Den enda modifikation som gjordes på bilen var att RP-Zero och Sense HAT fästes på taket.

Ett testprogram konstruerades som samlade in mätdata. Programmet tog som in-parameter den tid (sekunder) som data skulle samlas in under. Under mätningarna så skrevs tid, roll-, pitch-, och yaw-värdet till en csv-fil. Filen skickades och sparades på huvuddator för senare analys. Syftet med dessa tester var att få reda på hur accelerometern påverkades av de olika scenarion som bilen kunde tänkas ställas inför. Tester genomfördes för:

- Låg, konstant, hastighet (löstes av handkontrollens begränsningsmekanism).
- Mellan, konstant, hastighet (löstes av handkontrollens begränsningsmekanism).
- Mellan, anpassad, hastighet. Med anpassad menas att användaren försökte, i bästa mån, anpassa hastigheten efter raksträckor och kurvor.
- Höga, plötsliga, accelerationer.
- Kontrollerade avkörningar.
- Hög, anpassad, hastighet. Här försöktes en så bra varvstid som möjligt uppnås.

Utifrån dessa testkörningar undersöks datan för att hitta ett mönster i mätdata som kan användas för att detektera en kurva. Skulle inte detta vara möjligt så kan datan inte användas av AI-modellen för att prediktera när den är på väg att köra in i en kurva, och på så sätt sänka farten. Mest troligt kommer accelerationen kring roll-axeln att ge störst utslag - det är just rotation kring denna axel som får bilen att falla ur spåret och åka av banan. Men eftersom accelerationen kring de andra axlarna också kan vara intressanta för AI-modellen undersöktes även dessa.

3.4 Val och implementering av AI-modell

Implementeringen av Ekkonos AI skedde utefter deras rekommendationer. Deras SDK har stöd för många olika modeller men som en utgångspunkt valdes Random forest eftersom denna modell är enkel att implementera då ingångsvärdena inte behöver viktas. Detta ger en bra utgångspunkt för att bli bekant med deras API.

Modellen som skapades är av typen Random Forest, mer exakt en inkrementell Random Forest. Inkrementell innebär i sammanhanget att man kan skapa en helt tom modell som inte behöver för-tränas av en tidigare modell eller av annan data. Vidare så sätts ett antal attribut upp. Dessa attribut syftar till de attribut som ska användas för att träna modellen. Ett av dessa attribut väljes ut som mål-attribut, med andra ord det som användaren vill att modellen ska försöka prediktera. Horisonten för modellen är helt enkelt hur många steg framåt i tiden man vill att modellen ska prediktera för.

Attributen består av motorvärde(gas), Roll, Pitch och slutligen Yaw. Målattributet är Roll. Sammantaget innebär det att modellens uppgift är att, baserat på flera instanser av attributen skapa sig en uppfattning om hur roll-värdet kommer bete sig i framtiden.

3.4.1 Metod för utvärdering av AI modellen

Eftersom besluten om att gasa eller bromsa bilen är beroende på AI-algoritmens uppskattningar så är det viktigt att undersöka hur bra uppskattningar som modellen utför. Skulle modellen vara dålig på att uppskatta värden kan den komma att behöva ändras för att kunna genomföra bättre uppskattningar. Detta kan genomföras genom att öka antalet träd, ändra antalet förgreningar eller samla fler datapunkter mellan varje beslut.

För att utvärdera AI modellen används en model-evaluator som finns i Ekkonos SDK. Den kan kopplas på en modell och på så sätt successivt utvärdera modellen. Denna modellevalueringen kan användas till två saker. Dels för att utvärdera modellen och avgöra om den predikterar bra nog och dels för att ta ett beslut om när modellen är tränad nog för att programmet ska kunna börja ta egna beslut. Evalueringen presenterar ett absolut-värde för medel-felet hos modellen (MAE).

3.4.2 Människan mot maskinen

Det huvudsakliga syftet med rapporten är att undersöka om bilen som har utrustats med AI kan optimera sin körning på ett sådant sätt att den kör snabbare runt en bana än vad en människa kan köra. För att jämföra detta behöver en jämförelse mellan en bil körd av en människa och en bil körd av AI genomföras.

Eftersom att bilens prestanda påverkas av dess vikt och jämförelsen skall ligga mellan människan och AI-bilens prestanda så kommer bilen körd av människan att behöva modifieras efter som den är lättare. Den självkörande bilen och en vanlig bil kommer först att vägas. Sedan kommer blyvikter adderas till den vanliga bilen tills att dess massa är den samma som den självkörande bilen. Vikterna kommer monteras på bilens tak för att tyngdpunkten på de två bilarna ska bli mer likvärdiga. Detta gör att kraften som verkar på bilen i sidled kommer få liknande hävstång - vilket innebär att de har lika lätt att trilla av banan. Detta minimerar då viktens påverkan på resultatet.

Bilbanan har två spår som tillåter två bilar att tävla mot varandra. Men de olika bilbanor som kan konstrueras med de delar som fanns tillgängliga under projektet ger alltid ett spår en fördel. Detta beror på att ett av spåren får kortare körsträcka då den ligger innerst. För att detta inte skulle vara en faktor valdes att köra bilarna på samma spår - detta eliminerar då påverkan från spåren från resultatet.

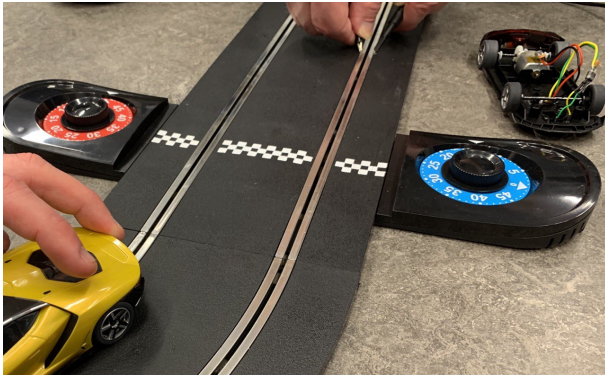
För att minska påverkan av den mänskliga faktorn kommer testerna utföras i heat, på samma sätt som testen för viktens påverkan. Tidtagningen för bilarna sker under ett heat, som även i detta test är 10 varv. Flertalet heats kommer att utföras för att ytterligare minska dessa påverkande faktorer. Utifrån tiderna av dessa test kan sedan den genomsnittliga varvtiden beräknas och resultatet för människan och maskinen kan jämföras.

4 Genomförande

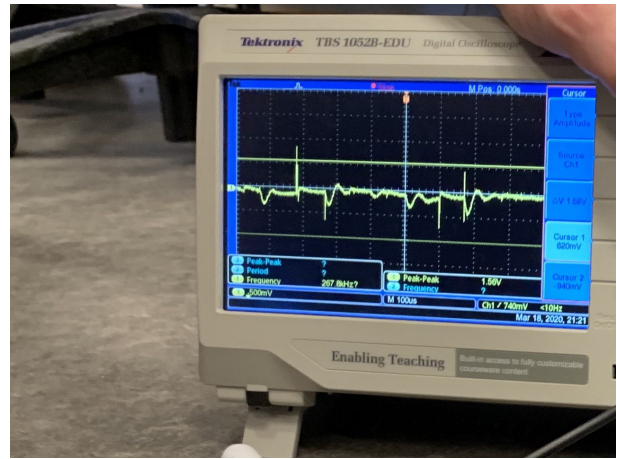
Projektet startade med att bilbanan undersöktes och i samband med detta lades en grov plan för strömförsörjning och vidare komponentinköp. På mjukvarusidan så påbörjades, tidigt, jobb med att sätta upp en utvecklingsmiljö och för-installationer för RP-Zero gjordes. Kunskapsinhämtning pågick löpandes för de olika Raspberry Pi API'erna och ett antal testprogram konstruerades. Under tiden som framsteg gjordes så testkördes olika delar både i sin helhet, på banan, men också isolerat. Kortare möten hölls med representant från Ekkono för att förstå SDK't och hur det senare skulle implementeras. Dessa bidrog även med värdefull insikt i vilka eventuella parametrar som skulle kunna komma att behövas. Slutligen knöts alla enskilda bitar ihop till en färdig prototyp med avsikt att ta smarta körbeslut baserat på artificiell intelligens.

4.1 Mätning av bilbanans elektriska delar

Under den inledande delen av projektet efter att bilbanan införskaffades genomfördes en undersökning och mätning av bilbanan och bilarna, detta genomfördes i tre syften. Dels för att få en förståelse för systemets konstruktion - en utgångspunkt för konstruktionen av systemet. Det är även viktigt att identifiera potentiella störningar som kan uppstå, så att de kan åtgärdas om de anses kunna störa utrustningen som ska användas. Dessa kan komma från bilens motor eller plockas upp i form av emitterade eller ledningsburna störningar från grannspåret. Figur 7 visar en undersökningen av störningarna från grannspåret. Det sista syftet med undersökningen var att undersöka spänningen och den maximala strömförbrukningen hos motorn för att kunna välja en passande motorstyrning.



(a) Undersökning av störningarna från grannspåret.



(b) De uppmätta störningarna.

Figur 7: Mätningar på potentiella störningar från grannspårets bil.

Testerna genomfördes med hjälp av ett oscilloskop. Figur 7 visar som tidigare nämnts ett exempel på det test som genomfördes för att identifiera eventuella störningar från bilen på grannspåret. Bilen matades med maximalt gaspådrag från handkontrollen och drogs långsamt längs spåret för att skapa maximala störningar. Vidare så mättes över det andra spåret för att detektera de eventuella störningar som uppstod.

Efter att alla mätningarna hade utförts så kunde en lämplig motorstyrning införskaffas. Dessa baserades som tidigare nämnts på den maximala strömförbrukningen av bilens motor och den spänning som ligger över den. Under väntan på leveransen av h-bryggan fokuserades arbetet på mjukvaran.

4.2 Installation av mjukvara och utvecklingsmiljö

Som operativsystem för RP-Zero valdes Raspian, ett linuxbaserat operativsystem som skapats av Raspberry Pi för deras datorer. Operativsystemet installerades på ett micro-SD kort som sedan agerar som hårddisken för RP-Zero. För att underlätta arbetet framöver skapades en SSH-anslutning mellan RP-Zero och en utvecklingsdator. Detta tillåter att, på ett säkert sätt, köra terminalkomandon på RP-Zero via utvecklingsdatorn. På så sätt behöver inte micro-SD kortet avlägsnas för att utveckla körbara program, det kan även användas för att initiera starten av dessa.

När SSH-uppkopplingen var initierad installerades Python och Sense HAT API:er. Därefter kördes en uppdatering för att säkerställa att de senaste versionerna av mjukvaran var installerad. Som ett sista steg kopierades Ekkonos SDK över från utvecklingsdatorn och installerades sedan på RP-Zero.

4.2.1 SSH-setup

För att skapa en SSH-uppkoppling så behövdes även en SSH-fil och en config-fil skapas på micro-SD kortet. SSH-filen möjliggör SSH-anslutning från en utvecklingsdator till RP-Zero. Config-filen innehöll namn och lösenord till det wifi-nätverket som användes för att kommunicera mellan de två enheterna. Detta berodde alltså på när och var utvecklingen skulle göras. Här användes en smartphone som wifi-hotspot för att kunna behålla samma wifi-nätverk hela tiden. Detta underlättade då arbetet skedde på olika arbetsplatser med olika wifi-nätverk. På så sätt behövde inte config-filen ändras vid starten av varje arbetspass.

4.2.2 Raspberry Pi Sense HAT

RP-Zero behövde installera ett par API'er för att få Sense HAT att bli brukbar. Efter att detta gjorts så konstruerades det ett testprogram som testade de olika givarna och ställdonen på Sense HAT. Eftersom Sense HAT och RP-Zero kommer från samma tillverkare så fungerade det direkt utan omständigheter. Sense HAT består av väldigt många fler givare och ställdon än vad projektet behöver men på grund av kompatibilitet så valdes den som utgångschip. Testerna som utfördes senare i projektet skulle komma att avgöra om detta kretskort behålls eller byts ut.

Figur 8 visar en skärmdump från huvuddatorns terminal. Först är det utskriften från ett testprogram som heter test.py som skriver ut data från accelerometern. Sedan syns kommandon som skrivs via huvuddatorn i RP-Zero, detta startar ett program som testar termometern på Sense HAT. Slutligen stängs RP-Zero av med kommandot shutdown".

```
{'yaw': 0.00023934267993671937, 'roll': 0.8376949540585608, 'pitch': 0.06513213030639285}
hello
^Z
[1]+  Stopped                  python test.py
pi@raspberrypi:~/Exjobb/Test $ nano test.py
pi@raspberrypi:~/Exjobb/Test $ python test.py
Temperature: 26.6126060486 C
26.6126060486
26.6126060486
pi@raspberrypi:~/Exjobb/Test $ nano test.py
pi@raspberrypi:~/Exjobb/Test $ sudo shutdown
```

Figur 8: Terminalen hos huvuddatorn när den är ansluten till RP-Zero och ett test-program köres.

4.3 Undersökning av tidsfördröjningar

En viktig aspekt i projektet för att generera så bra beslut som möjligt från AI-modellen är tid. Dels påverkar det hur aktuell datan från givare är, dels hur snabbt signalerna går ut till ställdonen (motorn) men också hur många beräkningar modellen kan genomföra. För att identifiera eventuella flaskhalsar och förbereda lösningen för trögheter så genomfördes ett antal test, ett exempel visas i figur 9. Eftersom Ekkono SDK är anpassat för RP-Zero så har det antagits att själva processorn på RP-Zero inte kommer vara en flaskhals och dess isolerade beräkningar och bearbetningar kommer vara försumbara, i tidsaspekten. Däremot när processorn ska kommunicera med utomstående komponenter så behöver signalerna gå igenom RP-Zero GPIO. Detta introducerar en fördröjning både hos själva RP-Zero men också hos en eventuell ny krets. Eventuella utskrifter på huvuddatorns konsol eller RP-Zero's konsol togs inte med i tidtagningarna då de inte representerar fortsättningen av projektet.

```
#Start stopwatch
start_time = time.time()

#Execute test
sense.get_accelerometer()

#Stop stopwatch
end_time = time.time()

#result is now the time difference in seconds.
result = end_time - start_time
```

Figur 9: Ett exempel på ett test med endast en läsning från givare. Där "result" har tiden (i sekunder) som det tog att genomföra läsningen.

4.3.1 Tidsfördröjning från givare till processor

Denna delen av testet berör huvudsakligen hur snabbt och ofta en givare kan läsas av. Alltså hur aktuell data som processorn (och modellen) får. En väldigt lång fördröjning här hade kunnat innebära att AI'n tar ett väldigt bra beslut, men ett par sekunder försent. Kortare fördröjning här kommer också öka mängden data som modellen kan få tillgång till under samma tidsperiod. Fördröjningar från givarna till processorn uppstår av flera skäl och vid flera tillfällen. Först hos själva givarna (t. ex. accelerometern) sedan hos Sense HAT och därefter kommunikationen med RP-Zero och slutligen till RP-Zero's processorn.

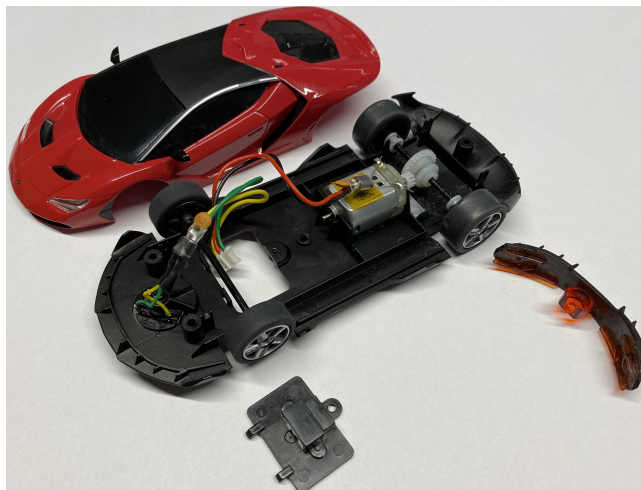
Testet konstruerades genom att låta en programrutin anropa Sense HAT's API i flera iterationer. I detta fallet så användes Sense HATens `get_accelerometer()`-anrop vilket läser av accelerometern. Under varje iteration och testet så lästes värdet och sedan skrevs in i en lista som sparade alla läsningar. Listans innehåll skrevs sedan ut för att kontrollera att värdena var rimliga men själva utskriften av listan skedde utanför tidtagningen. Tidtagningen gjordes med hjälp av Pythons egna tidsbibliotek. Läsningarna från givaren gjordes dels i ett statistiskt antal fall men också i en loop som exekverade flera olika antal gånger.

4.3.2 Tidsfördröjning från processor till ställdon

Detta partier behandlar hur snabbt ett ställdon (t. ex. en motor) kan få en signal för eventuell ökning eller minskning av hastighet. En väldigt lång fördröjning här hade kunnat innebära att beslutet som fattats av AI'n inte kommer fram till motorn i tid. Eventuell fördröjning hos själva motorn eller annan fysisk tröghet behandlas inte här. Testet konstruerades genom att låta en programrutin skriva till en av GPIO-kontakterna på RP-Zero i flera iterationer. Tidtagning gjordes med hjälp av Pythons egna tidsbibliotek. Varje iteration skickade ut en signal till en kontakt som i sin tur var kopplad till en LED. Både antalet skrivningar per test och antalet test ändrades under testets gång för att komma fram till ett så användbart resultat som möjligt.

4.4 Undersökning av viktpåverkan

Ett test genomfördes för att ta reda på de elektroniska komponenternas påverkan av bilens varvtid. I testet körde bilen i omgångar runt en bana med olika vikter. För att kunna genomföra testet vägdes först alla elektroniska komponenter var och en för sig och därefter vägdes bilen. Först vägdes bilen omodifierad, sedan utan karros och baklyckta då dessa delar gick att plocka av från bilen och till sist vägdes den även utan den borttagbara bottenplattan. Figur 10 visar hur bilen kunde demonteras i dess olika delar.



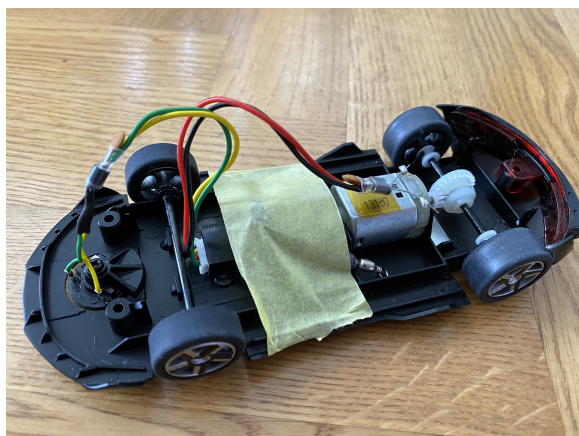
Figur 10: Isärplockad bil. Ovanför ses karrosen, under till vänster bottenplattan och till höger baklyktan.

I testet monterades bilbanan enligt sin enklaste form, en oval, för att ge föraren en så enkel bana som möjligt. Banan har två raksträckor och två högersvingar. Detta kommer kunna testa bilens förmåga att accelerera samt hålla sig kvar på banan under olika vikter, vilket direkt påverkar varvtiden för bilen. Under testet kördes bilen på det yttre varvet.

Bilen kördes under testet med fem olika vikter. Dels omodifierad men även med dess elektronik samt strömförsörjning. I elektroniken inräknas Raspberry Pi Zero, L298 H-brygga och Raspberry Pi Sense Hat. Testerna simulerar då följande:

- Omodifierad bil
- Bil utan kaross med elektronik
- Bil utan kaross med elektronik och powerbank
- Bil utan kaross med elektronik och 2x AA-batterier
- Bil utan kaross med elektronik och 9V-batteri

Bilen körde först omodifierad och sedan utan dess kaross med med vikter som motsvarar de komponenter som kommer att monteras på bilen. Vikterna som användes i detta test var blytyngder, av varierande vikter, som tejpades fast framför motorn på bilens underrede. Detta illustreras i figur 11.

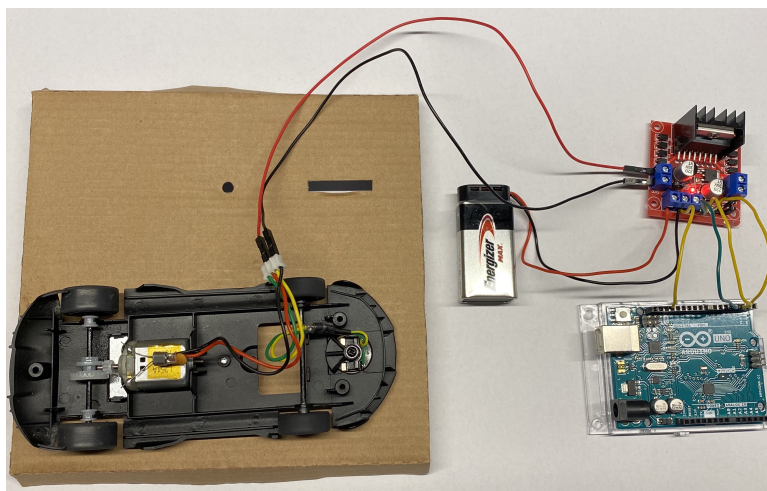


Figur 11: Bilden illustrerar hur vikterna monterades under testkörningen.

Testet utfördes av två personer. En person körde bilen medans den andra personen räknade antalet varv som bilen hade kört och tog tid. Bilen kördes 10 varv utan att den åkte av och tiden dokumenterades, detta upprepades 4 gånger för varje vikt på bilen som testades. Om bilen körde av under testets gång så började mätningen av den omgången om tills en lyckad omgång genomfördes.

4.5 Testkoppling av h-bryggan

För att kunna konstruera den slutgiltiga kretsen genomfördes en testkoppling med h-bryggan, se figur 12. Som strömkälla användes ett 9V-batteri och för signalgenerering en Arduino Uno. Detta testet hade kunnat utföras med Raspberry Pi Zero men Arduino Uno är en väldigt tålig enkortsdator som lämpar sig väl till att testa nya kopplingar då den har stor tålighet för att koppla fel.

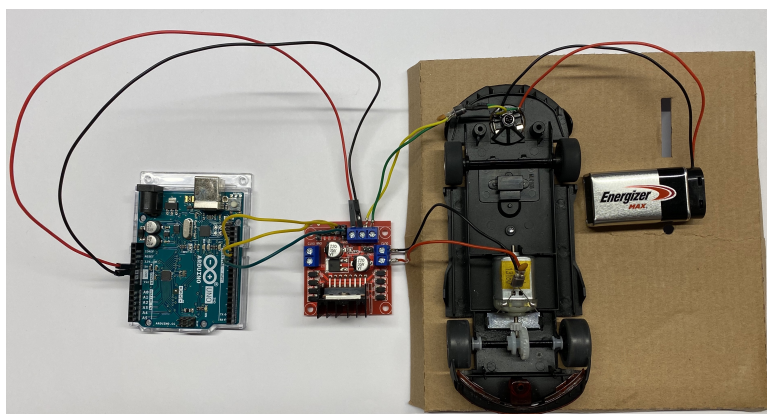


Figur 12: Kopplingen för genomförandet av testet med h-bryggan.

Den positiva polen från batteriet anslöts till h-bryggans positiva motordrivningsanslutning och den negativa polen till h-bryggans jord. Då spänningen från batteriet var under 12V lämnades bryggan till 5V regulatorn på plats för att driva kortets logik. Motorn anslöts sedan till ingångarna OUT1 och OUT2 på h-bryggan. För att kunna styra en motor ansluten till OUT1 och OUT2 används pinnarna ENA (Enable A), IN1 och IN2. På ENA sitter en brygga som kortsluter två pinnar. Denna kan lämnas på plats vilket blir likvärdigt med att ge fullt gaspådrag vid alla tillfällen. I detta test ska motorn styras med en PWM signal, så bryggan plockades bort från ENA pinnen.

För att generera styrsignalerna användes de digitala pinnarna 13-11 på Arduinon. Pinne 13 och 12 kopplades till IN1 respektive IN2 och pin 11 till den yttre pinnen på ENA (den närmst kanten av kortet). Slutligen drogs en kabel mellan Arduinons jord och h-bryggans jord. Ett kort program skrevs för Arduinon som sätter IN1 till hög och genererar en PWM signal som ger 50% gaspådrag under en sekund, sedan full gas under en sekund för att slutligen inte ge någon gas alls. Under testet fick Arduinon sin strömförsörjning från en USB-sladd.

Om spänningen som driver motorn överstiger 12 V måste bryggan till kortets 5V regulator kopplas bort. Detta gör att kortets inbyggda logik måste få ström från en annan källa. Då strömförsörjningen från bilbanan är 15 V så gjordes även en testkoppling då denna brygga plockades bort, se figur 13.

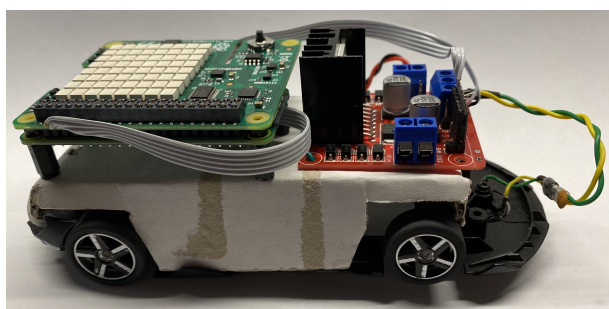


Figur 13: Koppling utan bryggan till kortets 5V regulator.

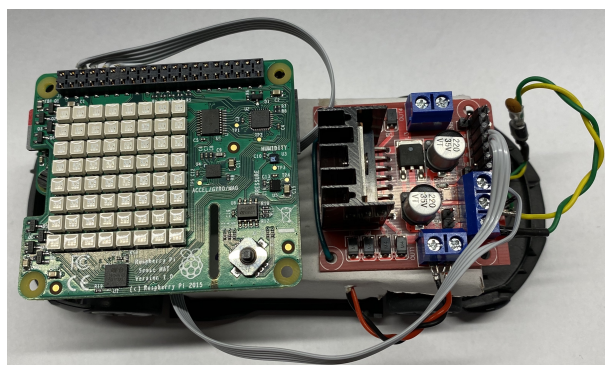
Denna koppling är väldigt lik den tidigare testkopplingen. I detta fallet har batteriet kopplats in via bilens kontakt mot bilbanan. Detta visar hur bilen kommer kopplas mot h-bryggan då ett batteri inte används utan strömen tas från bilbanan istället. Eftersom bryggan till kortets 5V regulator har kopplats bort så har en kabel kopplats mellan Arduinos 5V utgång till h-bryggans 5V ingång. Detta ger strömförsörjning till h-bryggans logik. För övrigt såg kopplingen likadan ut. Arduinos jord anslöts till h-bryggans jord och Arduinos digitala pinnar 13-11 kopplades till IN1, IN2 och ENA på h-bryggan.

4.6 Bilens konstruktion

Bilens konstruktion startade genom att plocka av så mycket delar som möjligt som inte påverkade bilens förmåga att köra. Detta innebar att bilens kaross, baklykta och bottenplatta avlägsnades. Figur 10 från sektion 4.4 visar hur bilen monterades isär. Eftersom de elektroniska komponenterna inte kan monteras direkt på bilens chassi konstruerades en upphöjd platå av kartong som dessa kunde monteras på. Figur 14 visar elektroniken monterad på bilen med hjälp av platån. H-bryggan är monterad på bilens främre del och Raspberry Pi Zero samt Sense HAT på den bakre delen.



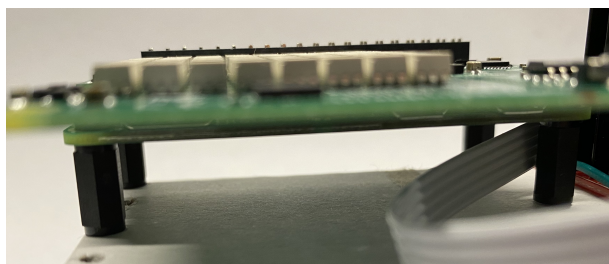
(a) Sidovy av bilen.



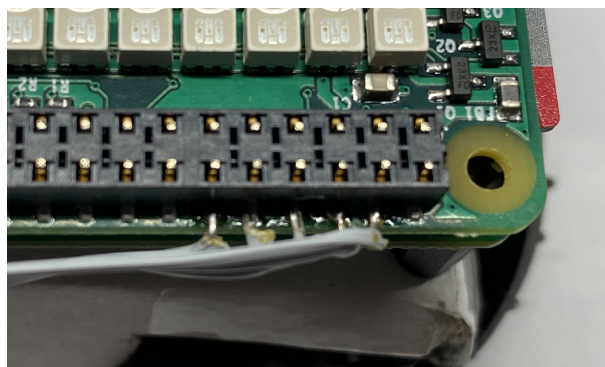
(b) Bilen ovanifrån.

Figur 14: Bilens uppbyggnad.

Sense HATen sitter monterad direkt på Raspberry Pi Zero, den hålls fast med hjälp av GPIO pinnarna. Raspberry Pi Zero har 4 distanser monterade på sin undersida, vilket används för att fästa den i platån, se figur 15. I figuren visas även hur en bandkabel har löts fast på Sense HATen. Dessa ingångar används inte av Sense HATen men har direkt kontakt med Raspberry Pi Zero vilket nyttjas för att nå de utsignaler som skickas genom de pinnarna. Bandkabeln är sedan kopplad till h-bryggan.



(a) Montering av Raspberry Pi Zero i bilen.

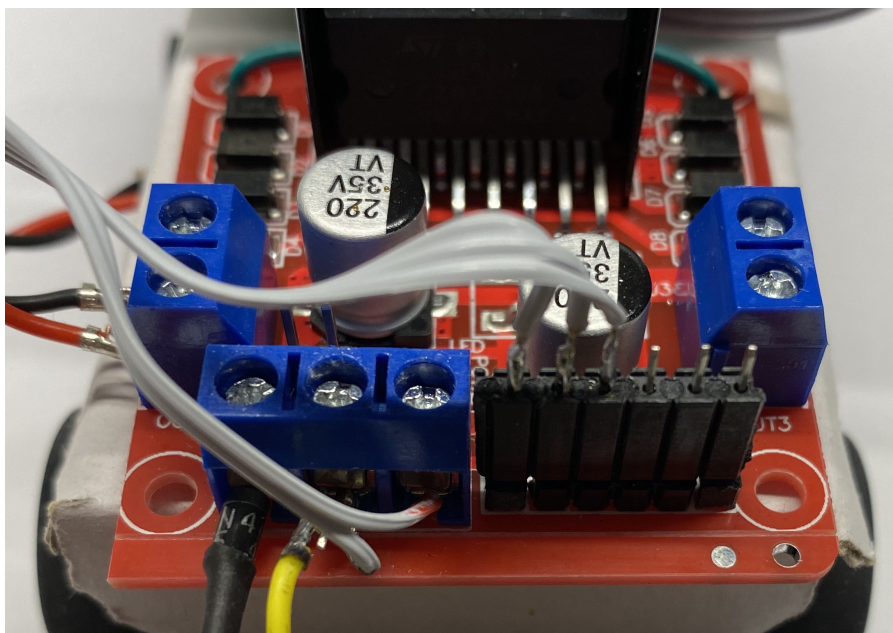


(b) Lödning på Sense HAT.

Figur 15: Installation av Raspberry Pi Zero och Sense HAT.

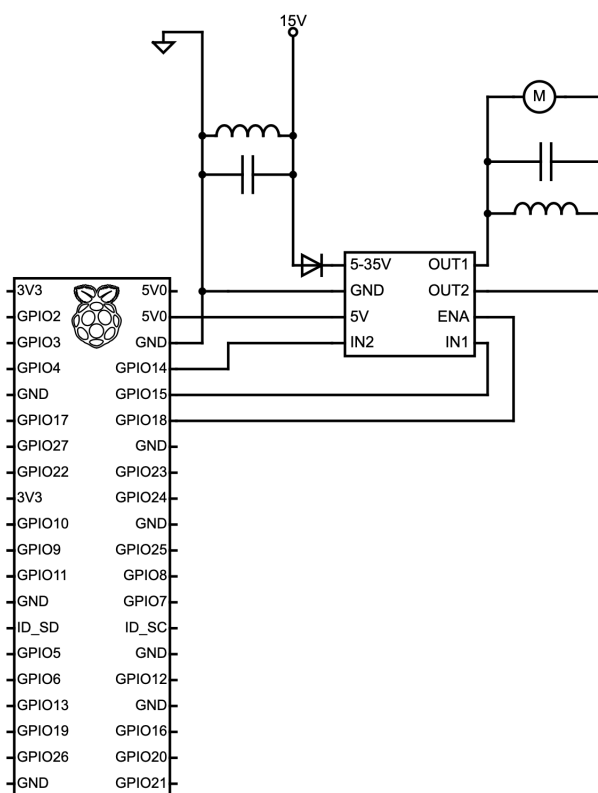
Inkopplingen av h-bryggan kan ses i figur 16. Kablarna från motorn är partvinnade och anslutna till IN1 och IN2 på h-bryggan. Kablarna för matningen från bilbanan är också partvinnade och är anslutna till h-bryggans matningsspänning och jord. I serie med matningsspänningen sitter en diod för att skydda de elektriska komponenterna om bilen skulle sättas åt fel håll på banan, vilket skulle resultera i att matningsspänningen gick in i jord på h-bryggans och Raspberry Pi Zero. Dioden hindrar därför att kretsen sluts om bilen kopplas in åt fel hål. Bandkabeln från Sense HAT har 5 kablar som är kopplade till h-bryggans: 5V ingång, jord, ENA, IN1 och IN2. De 5v som kommer från Raspberry Pi Zero försers L298 kortets logik med ström. Kablarna till ENA,

IN1 och IN2 är lödda till en stiftlist istället för att löda dem direkt till kortets pinnar. Detta gör att h-bryggan kan kopplas loss helt vid behov.



Figur 16: Koppling till L298 h-bryggan.

Ett kopplingsschema för bilen kan ses i figur 17. I verkligheten är kablarna mellan Raspberry Pi Zero och h-bryggan kopplade via kontakthuset på Raspberry Pi Sense HAT, enligt figur 15. Men eftersom dessa kan ses som en förlängning pinnarna på Raspberry Pi RP-Zero så har detta skippats att ritas upp i kopplingsschemat. På schemat kan det utläsas att en kondensator och ett EMI filter sitter kopplat parallellt med både inmatningen och över motorn. Dessa filter satts på bilen från början för att filtrera ut störningar från motorn och banan och lämnades på plats.



Figur 17: Bilens kopplingsschema. Till vänster Raspberry Pi Zero, i mitten L298 h-brygga.

4.7 Programvarans uppbyggnad

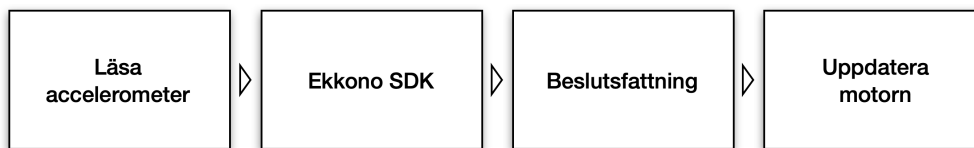
Programmet består av ett antal olika moduler. Vissa moduler, och dess API'er, skrevs specifikt för projektet medans andra fanns tillgängliga innan. Ekkono SDK är en stor del av programmets uppbyggnad och tillhandahålls av Ekkono. Diverse API'er och moduler specifika för Raspberry Pi-produkterna finns installerade på de respektive korten. Utöver de redan färdiga modulerna så skapades följande delar:

Accelerometer Programmodul för att komma åt Sense HAT's givare. Modulen ansvarar för avläsningar av Sense HAT, mer specifikt accelerometern.

Engine Programmodul för att kontrollera motorn. Modulen ansvarar för åtkomst till GPIO. Ett antal metoder för att kontrollera motorn.

Controller Huvudmodulen eller själva programmet. Sörjer för anropen till de olika API'erna och de andra modulerna. Controller-modulen ansvarar även för den delen av programmet som kallas beslutsfattningen.

I figur 18 visas de huvudsakliga delarna av programflödet. Dessa fyra steg repeteras så länge bilen är aktiv. Närmare förklaring av Ekkono SDK respektive Beslutsfattning följer i kommande sektioner.



Figur 18: De huvudsakliga byggklossarna i programmets struktur.

4.7.1 AI i mjukvaran

Den delen av programmet som utgör AI och maskininlärning går under namnet "Ekkono SDK". Det är nämligen det utvecklingspaketet (SDK) som Ekkono har tillhandahållit. Med dessa modeller och applikationer följer API'er. Den delen av mjukvaru-konstruktionen som berör just AI är alltså en serie instruktion som använder sig av detta SDK.

Först skapas en "Incremental" prediction model. Detta betyder att modellen kan skapas tom och successivt fyllas på med träningsdata. Modellen initieras tidigt med vilka attribut som användaren är intresserad av. I detta fallet är det EngineValue (PWM-signalen mellan 0.0 och 1.0), Roll, Pitch och slutligen Yaw. Vidare så förbereds modellen även på vilket attribut (värde) som man önskas få förutspått och i vilken tidshorisont det gäller. Av anledningar som tidigare diskuteras valdes här Roll som utvärde. Horisonten för prediktionen innebär hur många steg fram i tiden man vill ha prediktionen. Enheten steg används i detta fallet och värdet är 10. Eftersom medelhastigheten för bilbanan är hyfsat jämn så fanns här ingen anledning att räkna om antal läsning till sekunder eller antal läsningar till sträcka.

När maskinen (AI) har gjort sin prediktion på ett Roll-värde om 10 avläsningar framåt i tiden så skickas detta värde vidare till beslutsfattningsdelen av programmet.

4.7.2 Beslutsfattning

Beslutsfattningsmodulen av programmet är den delen som ansvarar för att avgöra vad motorvärdet (gasen) bör vara i nästa läge. Beslutsfattningens huvudmål är att med hjälp av en prediktion försöka optimera körningen på ett sådant vis så snabbaste varvtiderna kan uppnås. Det förutspådda Roll-värdet är det som modellen tror kommer ske om tio stycken avläsningar. Det är viktigt att ta hänsyn till de utmaningar och förutsättningar som just en bilbana har men samtidigt inte skraddarsy en lösning för just en bandesign då bandesignen mycket väl kan ändras. Motorn accepterar signaler mellan 0.00 och 1.00 men värden under 0.3 kan i vissa fall leda till att motorn inte orkar driva bilen framåt samtidigt som värden över 0.4 mycket väl kan resultera i en avkörning (beroende på kurva). Roll-värdet som predikteras av modellen behöver sättas i relation till det roll-värde som bilen har för stunden, då det är skillnaden som bidrar med den största utmaningen.

Beslutsfattningen är uppbyggd så att den reglerar hastigheten allt eftersom varje inläsning, träning och prediktion har genomförts. För att förtydliga så kan detta jämföras med ett annat tillvägagångssätt som hade kunnat gå ut på att försöka ta fram ett fel-värde. Det värdet skulle i så fall representera vilken hastighet bilen hade kunnat klara av jämfört med vilken hastighet bilen faktiskt har. Vidare så delades här beslutsfattningen in i två delar. Uppdelningen gjordes för att uppnå lite skillnad på beteendet huruvida bilen körde i max-hastighet eller ej. Mönstret som önskades uppnås var att om maskinen körde på maxhastighet så skulle den reagera på små förändringar samt sänka hastigheten med större steg. Om bilen körde långsamt och ändå upplevde att den skulle bromsa in så skulle mindre steg tas nedåt. Indelningen av beslutsfattningen är ett försök att efterhärma en typisk bilbana men samtidigt inte anpassas efter en viss bandesign.

4.7.3 Initiala träningsprocessen

Det som kallas den initiala träningsprocessen är en förkortad variant av den, hela, processen som syns i bild 18. Denna träningsprocess består endast av de två första byggklossarna. Detta innebär att under denna process så läses accelerometer av och sen tränas modellen med dessa värden samt motorhastigheten. Motorhastigheten styrs i en förutbestämd rutin i fem olika hastigheter där ingen hastighet är hög nog för bilen att köra av samtidigt som att ingen hastighet är låg nog för bilen att totalt stanna. Anledningen till denna process är att AI-modellen skapas från nytt vid varje körning och tränas ej av redan befintlig data eller redan skapad modell. Dess prediktionsförmåga är helt enkelt för opålitlig de första iterationerna. Denna träningsprocess pågår i 500 iterationer, cirka 10 sekunder. Efter träningsprocessen så kopplas även beslutsfattning och motoruppdatering in. Modellen fortsätter att tränas även efter träningsprocessen.

5 Resultat

Utifrån undersökningen utav bilen och mätning av störningar från motorn konstaterades att en befintlig filtrering var installerad för att delvis hantera dessa. Denna filtrering kompletterades genom att partvinna kablarna till motorn och de som kom från spåret. Utifrån resultatet av mätningarna valdes också L298 h-bryggan till bilens motorstyrning.

När motorstyrningen hade bestämts så genomfördes undersökningen av massans påverkan på bilen - då alla komponenter kunde vägas. Resultatet visade att vikten av utrustningen inte skulle komma att vara ett problem för bilens förmåga att köra runt banan. Däremot konstaterades att vikten påverkade bilens hastighet - vilket stämmer överens med teorin. Hastigheten som bilen körde med var dock tillräckligt hög för att en optimering av bilens hastighet skulle vara möjlig och bilen kunde fortfarande köra av banan om det körde för snabbt i en kurva.

När det konstaterades att vikten av utrustningen inte skulle komma att vara ett problem för projektets genomförande undersöktes om prestandan hos Sense HAT skulle komma att vara tillräcklig. Först undersöktes tidsfördröjningarna vid avläsning hos accelerometern i Sense HAT. Testerna visade att den kunde läsas av med en frekvens på 50Hz, vilket skulle komma att vara tillräckligt för projektets genomförande. Vid undersökningen av tidsfördröjningar testades även skrivhastigheten till GPIO pinnarna på RP-Zero W. Den tid som en skrivning kunde genomföras under var så pass liten att den kunde anses försumbar.

När avläsningshastigheten hos Sense HAT inte ansågs vara ett problem undersöktes den data som kunde läsas av från accelerometern för att bedöma om den skulle komma att vara användbar för AI-algoritmen. Resultatet av undersökningen visade att accelerationen runt roll-axeln skulle komma att ge störst utslag då bilen körde genom en kurva. Dock upptäcktes det att Sense HAT genomförde något typ av behandling av accelerometer datan, troligtvis för att ge stabilare mätningar. Detta gav en viss tröghet under väldigt plötsliga ändringar i acceleration runt en axel. Datan som avlästes bedömdes dock ge tillräckligt stora och snabba utslag för att användas till styrningen av bilen.

Vid kontrollerandet av modellens prediktionsförmåga så användes en model-evaluator. Prediktionsförmågan mäts i MAE och visade sig, som väntat, att gå ner allt eftersom modellen fick mer tid att träna. En felmarginal på 10-15 % observerades. Vidare så testades programmets (maskinen) resultat genom tidtagning vid körning. Tiderna jämfördes mot de referens tiderna (en människas tider) som hade uppmäts och det visade sig vara en jämn kamp. Maskinen uppmätte en genomsnittlig varvtid på 2,43 sekunder men människan var lite snabbare med en varvtid på 2,22 sekunder. Även om maskinen var genomsnittligt sämre så levererade den den snabbaste varvtiden på 2,14 sekunder, men den var inte konsekvent i sitt körande. Däremot observerades att maskinen mycket riktigt optimerade sin körning till viss grad.

5.1 Mätningar på systemets elektroniska delar

Vid mätningen av bilbanan genomfördes först en visuell inspektion av bilen. Där konstaterades att kondensatorer och ferritblock var installerade för att filtrera bort störningar från bilens motor. Resultatet av mätningarna på bilens motor visade dock att filtreringen inte hanterade all störning. Spänningspikar på 50V kunde uppstå när bilens motor belastades tungt eller gick från stillastående läge till plötsligt fullt gaspådrag. Denna spik varade under en väldigt kort period men kan vara skadlig för utrustning som RP-Zero beroende på hur de har kopplats. Eftersom RP-Zero och motorn inte delar strömförsörjning ansågs detta inte behöva åtgärdas för utrustningens säkerhet. De störningar som initialt fruktades komma från grannspåret ansågs efter mätningar vara försumbara i detta projekt. Det uppmättes störningar på +/-1,5V då bilen på banan bredvid utsätts för hög belastning eller plötsliga gaspådrag från stillastående läge. Skulle strömförsörjningen för logiken tas från bilbanans räls skulle detta behöva åtgärdas då ett spänningsfall på 1.5V har potential till att skapa situationer då kretskorten skulle få för låg matningsspänning. Detta kan då orsaka att utrustningen stängs av under en kort period, vilket kan orsaka skada.

För att kunna välja motorstyrning utfördes mätningar på bilens maximala strömförbrukning och spänningen över rälsen. Resultatet av dessa mätningar visade att den maximala strömförbrukningen för bilen var 800mA och spänningen var 15V. Med detta som utgångsläge valdes L298 H-bryggan som projektets motorstyrning. Den tål en maximal ström av 2A och kan hantera en matningspänning mellan 5-35V. H-bryggan behöver 5V logik-spänning för att fungera, denna tas från RP-Zero. L298 har stöd för att styra två DC motorer, vilket inte används i projektet men är vanligt förekommande bland motorstyrningskort då dessa utgångar kan användas för att styra en stegmotor istället. Då motorstyrningen sker med hjälp av en PWM-signal kan emitterade störningar uppstå från kablarna kopplade till motorn och de från bilbanans räls eftersom de bildar en ögla. Genom att partvinna dessa kablar får varannan ögla motriktad störspänning vilket gör att störningarna tar ut varandra.

5.2 Massans påverkan av bilens prestanda

Komponenternas massa befarades kunna vara ett problem, skulle den bli för stor så skulle inte motorn orka köra bilen. Resultatet av undersökningen visade dock att bilen klarade att köra med vikt som motsvarar den utrustning som skulle kunna moteras på bilen. Tabell 1 visar de individuella komponenternas vikter. Var för sig väger inte de elektroniska komponenterna särskilt mycket. Men vid en jämförelse med bilens vikt skulle bara monteringen av en powerbank på bilen fördubbla dess vikt, vilket enligt teorin kommer att halvera dess acceleration. Det noterades även att strömförsörjningen skulle komma att vara det största individuella bidraget, oavsett vilket typ av batteri som används.

Komponent	Massa (g)
Raspberry Pi Zero W	12
Raspberry Pi Sense Hat	18
L298 H-brygga	24
Powerbank	72
2x AA-batteri	55
9V batteri	48
Bil	77
Kaross + baklykta	23

Tabell 1: Uppmätta vikter hos projektets komponenter.

Bilen utrustades med vikter som skulle motsvara de olika konstruktionsalternativen som skulle kunna komma att användas. RP-Zero W, Sense HAT och h-bryggan motsvarade det konstruktionsalternativ som skulle bidra med minst öknin av massa. Bilen skulle då väga 108g och strömförsörjningen skulle antingen behöva tas från spåret i bilbanan eller via en sladd från en strömkälla vid sidan av bilen. Utöver detta utfördes tester med vikter som motsvarade att strömförsörjningen monterades på bilen.

Det var det direkt märkbart att bilens förmåga att köras ändrades så massan ökade, till och med den absolut minsta ökningen på 31g märktes. Bilen blev svårkontrollerad då den lätt började spinna med hjulen då den gick från låga hastigheter till fullt gaspådrag och vid kurvor sladdade bilen runt på banan, detta blev värre desto tyngre bilen blev. Att bilen sladdade i kurvorna berodde på att magneten som håller bilens bakdel i position över spåret inte längre orkade motverka de krafter som verkade på bilen i sidled. Detta gjorde det extra svårt att behålla hastigheten igenom kurvor då bilen måste återfå kontrollen från sladden innan gaspådrag kan ges igen. Enligt resultaten på varvtiden i tabell 2 märks en klar korrelation mellan ökad massa och ökad varvtid. Komponentkolumnen beskriver vilken utrustning på bilen som testet skulle simulera.

Komponenter	Massa (g)	Test 1 (s)	Test 2 (s)	Test 3 (s)	Test 4 (s)	Snittid 1 varv (s)
Omodifierad bil	77	18,55	18,43	19,45	18,80	1,88
Bil utan kaross med elektronik	108	19,99	20,38	21,47	21,44	2,08
Bil utan kaross med elektronik och powerbank	178	25,12	25,01	25,57	24,34	2,50
Bil utan kaross med elektronik och 2x AA-batterier	161	24,17	23,52	24,04	23,71	2,39
Bil utan kaross med elektronik och 9V-batteri	154	23,52	24,36	24,28	24,89	2,43

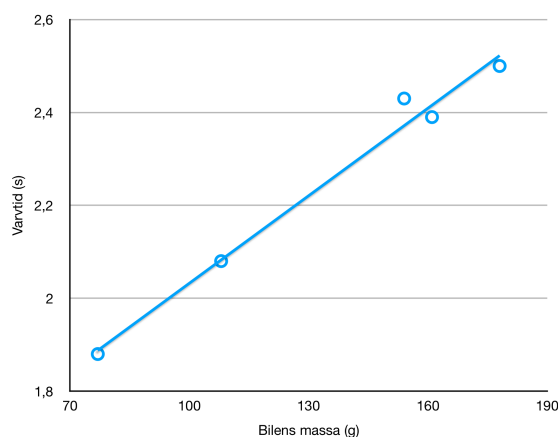
Tabell 2: Uppmätta tider samt uträknad snittid för ett varv med varierande massor.

Totalt utfördes 4 test för varje vikt där bilen körde 10 varv runt samma bana. Snitttiden beräknades sedan genom att addera tiderna och dividera med det totala antalet varv. Under tabellen ”Komponenter” nämns ordet ”elektronik” vilket syftar till RP-Zero, Sense HAT och L298 H-bryggan. Testet som utfördes med elektroniken men utan någon strömkälla simulerar att strömförsörjningen för de elektroniska komponenterna tas från bilbanans räls eller från sladd.

Figur 19 visar ett punktdiagram tillsammans med en trendlinje för varvtiden mot massan. Enligt resultatet kan man ana att en ökad massa innebär en linjär ökning i varvtid inom detta spann av vikter. Genom att titta på ekvation (4) från sektion 2.2 kan man utläsa att den kraft som verkar på ett objekt står i korrelation med dess acceleration och massa. Detta gör att vi kan beräkna accelerationen hos ett objekt med avseende på den kraft som verkar på det samt dess massa.

$$a = \frac{F}{m}$$

Eftersom motorn hos bilen förblir oförändrad kan vi anta att den resulterade kraften som verkar i bilens riktning förblir oförändrad oberoende på bilens vikt. Detta gör då att en ökning hos bilens massa leder till en minskad acceleration hos bilen, vilket i sin tur leder till en ökad varvtid.



Figur 19: Graf av bilens varvtid mot massa.

5.3 Tidsfördröjningar i systemet

De mätningarna som genomfördes för att ta reda på de olika tidsfördröjningarna i systemet visade att just fördröjningarna inte skulle påverka resultatet negativt. Tidsfördröjningarna på det slutgiltiga programmet stämde bra överens med de tidtagningar som gjorts på de diverse beståndsdelarna programmet bestod av. Uppdateringsfrekvenser i 40-50 Hz området ska enligt Ekkono och projektets egna mål inte bidra till några problem.

Testerna i tabell 3 visar resultatet från undersökningen av tidsfördröjningar i sektion 4.3. I tabellen syftar kolumnen ”Antal” till antalet process-cykler som testet mättes över.

Genomfört test	Antal	Snittid (s) / iteration
Process-cykel, utan läsning/skrivning från/till givare/ställdon	1000	5,25355e-06
Process-cykel, utan läsning/skrivning från/till givare/ställdon	1	4.60543e-05
Process-cykel, med läsning från givare	1000	0.02027
Process-cykel, med läsning från givare	1	0.11291
Process-cykel, med skrivning till ställdon	1000	0,00017
Process-cykel, med skrivning till ställdon	1	0.00160

Tabell 3: Uppmätta tider hos de olika testerna.

Slutgiltig mätning av tidsfördröjningar i systemet, där samtliga programmets steg inkluderas. Tidtagning inkluderar 1500 process-cykler där de första 500 består av träningscykler och de resterande 1000 består av vanliga kör-cykler. Det ända steget som skiljer de två olika cyklerna är beslutsfattningen. Eftersom beslutsfattningen är en intern beräkning i processorn som inte kräver någon läsning eller skrivning till eller från någon sensor så anses skillnaden försumbar, detta på grund av storleksskillnaden i snittiderna, se tabell 3.

Flertalet testkörning genomfördes och resultatet var väldigt konsekvent. Ett exempel på resultat visas i bild 20. Där mättes resultatet upp till 32,81781... vilket är representativt mot den slutliga medeltiden som räknades ut

till 32,802 sekunder (avrundat till tre decimaler). Dividerat på de 1500 cyklerna som genomfördes så resulterar detta i ungefär 45,7 iterationer per sekund, se beräkning 5.

$$\frac{1500}{32,802} = 45,7289189684 \quad (5)$$

I de tidigare mätningarna som gjordes över olika tidsfördröjningar i systemet, se bild 3, så kan man se att det är läsning från givare som är den överlägset långsammaste delen. Det finns två anledningar till att dessa tidtagningar är intressanta. Dels ska programmet vara snabbare än en människa så att om den tar ett "bättre" beslut än människan så ska inte dess beslutshastighet vara anledningen till att den förlorar eller kör sämre. Den andra anledningen är att AI-modellen inte ska prediktera ett sämre eller till och med ett skevt värde på grund av att den inte blir tränad tillräckligt snabbt. Den uppmätta frekvensen på cirka 45,7 Hz är i båda hänseenden väldigt tillfredsställande. För att säkerställa AI-modellens "krav" så har Ekkono kontaktats för att bekräfta att hastigheten är hög nog.

```
Done running
Total execution of 1500 (500+1000) iterations took: 32.81781339645386 seconds.
```

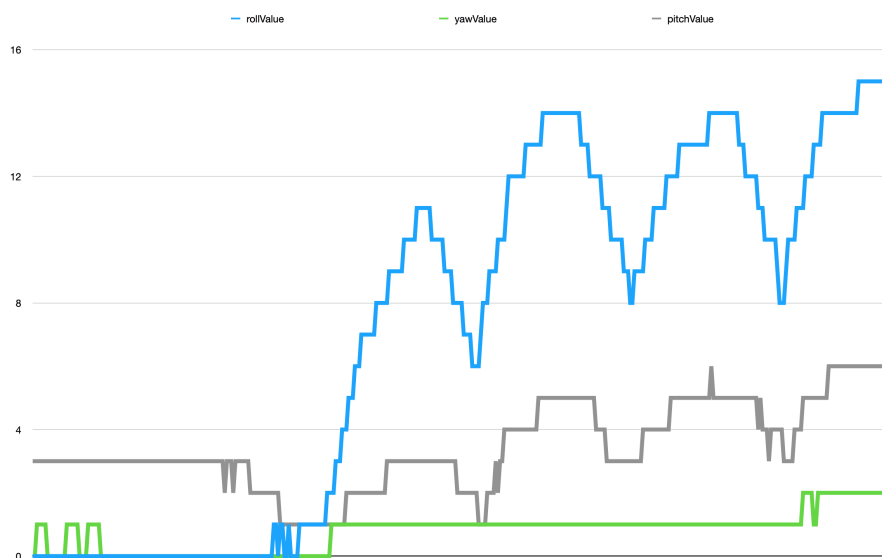
Figur 20: Utskrivet resultat för tidtagning av programmet.

5.4 Avläsning av accelerometervärden under körning

I detta avsnittet presenteras resultaten av de tester som genomfördes med hjälp av bilens accelerometer när bilen kördes under normala förhållanden, se sektion 3.3.4. RP-Zero och Sense HAT adderar lite vikt på bilen men det var ingen märkbar skillnad mot körning utan dessa kort. Vidare syftar "normala förhållanden" till att bilen kördes med den, till bilbanan, medföljda handkontrollen.

I figur 21, 22 och 23 representerar blå färg roll, grön färg yaw och grå färg pitch, se sektion 2.6.1. För samtliga diagram i avsnittet representeras tiden på x-axeln. Mätningarna startade från bilens stillastående läge.

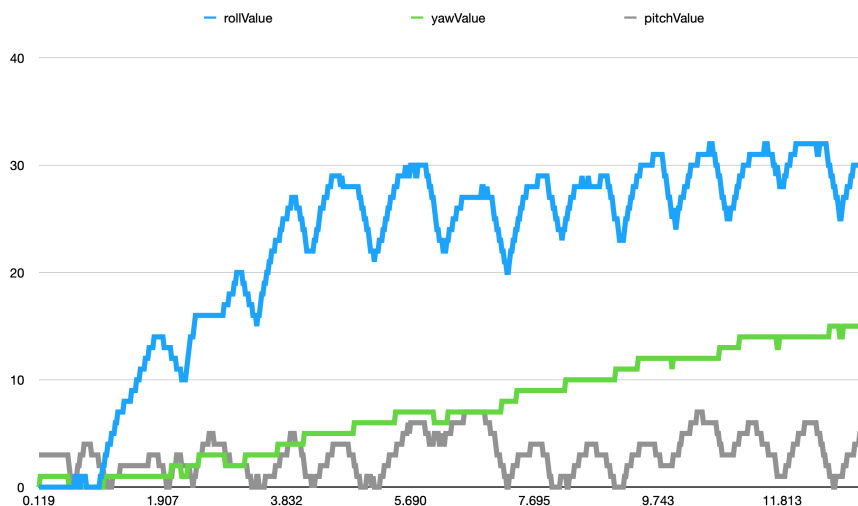
Värdena som avläses av accelerometern är dess grader runt en viss axel. I dessa diagrammen så är alla värden över 180 grader "neutraliserade" till att vara 360 - värdet. Detta innebär att värdet 355 presenteras som 5 (360-355). Detta är gjort för att ge tydligare diagram samt att hänsyn till riktning inte behövdes ta. I roll-led, som fokus låg, spelar det inte någon roll om bilen lutar/"rollar" 5 grader åt höger eller åt vänster. Dessa tester genomfördes på en oval bana med samma körriktning hela tiden. Vilket innebär att det fanns vissa riktningar och/eller rotationer som bilen omöjligt kunde uppnå.



Figur 21: Långsam, kontrollerad, hastighet.

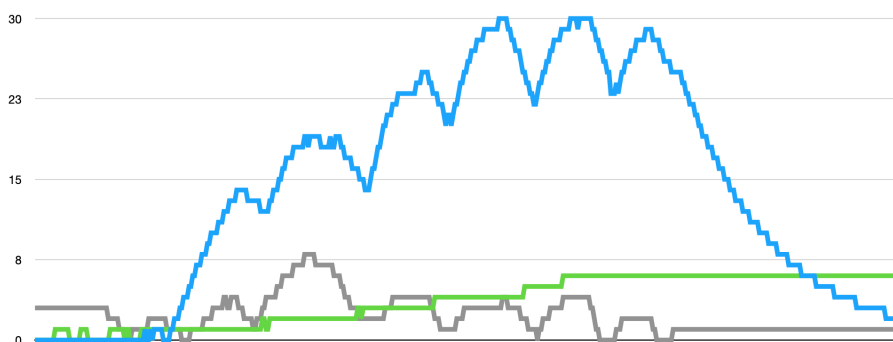
Redan i ett av de första testerna, se figur 21, då bilen kördes med långsam och konstant hastighet syntes utslag på roll, den blå kurvan. Detta var axeln som initialt förväntades visa mest och därmed vara mest användbar. Varje topp representerar en sväng på bilbanan, vilket leder till att ett varv består av två sådana toppar, och deras respektive dalar.

Liknande resultat observerades även på andra tester. T. ex. i ett test med hög och icke-konstant hastighet (högre på raksträckorna och långsammare i kurvorna), se figur 22, så påminner formationerna om varandra. Däremot avläses högre värden i detta testet. I topparna avläses 32 grader i roll-led.



Figur 22: Hög, anpassad, hastighet.

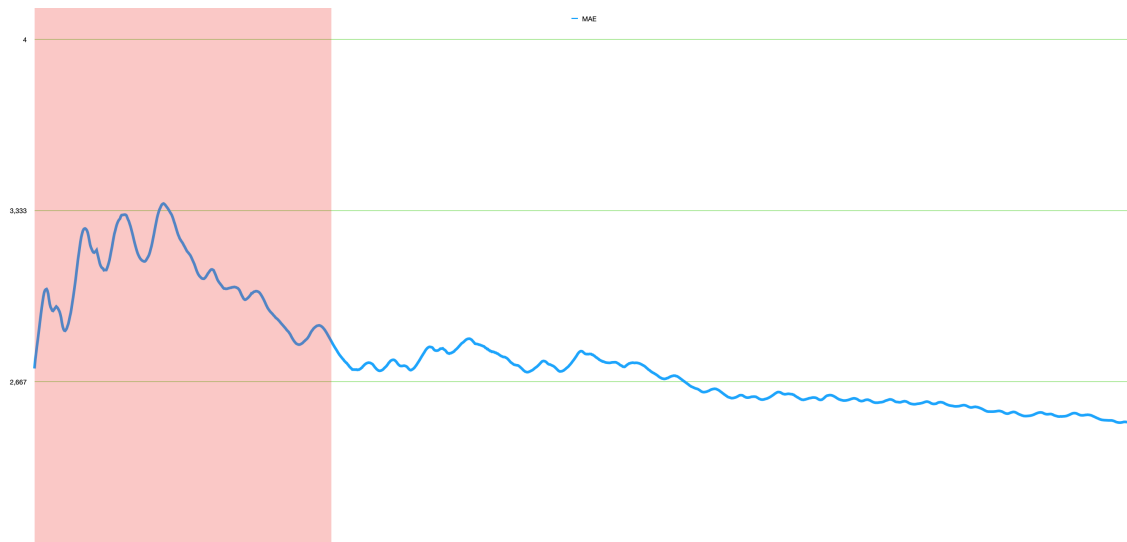
Vidare så uppmärksammades att ett roll-värde som snabbt steg tenderade att resultera i en avkörning. De första testen som genomfördes var en kort raksträcka in i en kurva. Om bilen framfördes på ett sådant sätt att den precis klarade sig från att åka av så hamnade roll-värdena på ungefär 20 grader. I de senare testen, t. ex figur 22, där flera varv kördes så klarar sig bilen från att åka av även på så höga roll-värden som 30. I figur 23 syns en avkörning efter tre varv.



Figur 23: Hög, anpassad, hastighet. Med avkörning.

5.5 AI-modellens prediktionsförmåga

Det, av projektet, använda SDK't tillhandahåller, via API't, en egen så kallad "model evaluator". Det är alltså en funktionalitet som bedömer hur pass bra modellen har förutspått värden. Om den ska förutspå nästa värde så kommer den alltså inte att veta hur pass bra modellen var (lågt MAE) förrän det värdet faktiskt har uppmätts. MAE mäts i absolut värde på just roll-prediktionen, vilket betyder att enheten är grader i roll-axeln. Figur 24 visar det uppmätta MAE-värdet över tid. Grafen består av ett rött parti som markerar det partiet där motorhastigheterna sätts till bestämda värden. Det partiet ämnar att ta hand om de mest dramatiska förändringarna i träningen. Efter detta partiet så styr programmet motorhastigheten själv och fortsätter träna modellen. Felet är som störst lite över 3,3 och som minst lite under 2,6. Däremot går trenden mot ett mindre och mindre fel allt eftersom antalet iterationer ökar och tiden går. Storleksordningen på de, under testen, uppmätta rollvärden är mellan 15 och 25, vilket innebär att fel-marginalen hos modellen är, efter den initiala träningen, mellan 10 och 15 procent. Allt eftersom modellen fortsätter köra så sjunker MAE men den når aldrig 0. Dessutom är värdet ett absolut-värde vilket betyder att man inte kan utläsa om det är underskattning av rollen eller överskattning av rollen som uppmätts.



Figur 24: Modellens MAE uppmätt över 1500 iterationer.

5.6 Maskinen mot människan

Det som skulle undersökas var om maskinen kunde vinna över människan och detta mäts i tid runt en bana (kortast vinner). För att minimera den mänskliga faktorn i tidtagningen så kördes flertalet varv i flertalet test för att få fram en snitttid för ett varv. Eftersom innerspåret på banan är kortare så använde sig också båda bilarna av samma (det yttre) spåret. När dessa tester genomfördes så mättes även den slutgiltiga vikten på den modifierade bilen till 120 gram och människans bil, den omodifierade, justerades till samma vikt och efterhärmande av tyngdpunkt, detta ger maskinen och människan så lika förutsättningar som möjligt.

De uppmätta tiderna för maskinen var konsekventa och den slutgiltiga snitttiden blev 2,22 sekunder för ett varv, se tabell 4. Den uppmätta snitttiden på bilen var 2,43 sekunder men med större osäkerhet. Första heatet för maskinen hade den kortaste snitttiden som till och med var snabbare än människans bästa snitttid. Det var dock svårt att få till liknande tider i de senare heaten och maskinen var oförutsägbar.

Förare	Heat #1	Snitt #1	Heat #2	Snitt #2	Heat #3	Snitt #3	Total snittid (s)
Människan	22,16	2,22	21,94	2,19	22,55	2,26	2,22
Maskinen	21,41	2,14	26,30	2,63	25,13	2,51	2,43

Tabell 4: Uppmätta heat-tider samt uträknad snitttid för ett varv.

Ett annat sätt att kontrollera och mäta resultatet i projektet upptäcktes i samband med justeringar och tidtagningar. Med en specifik bana och en specifik vikt (och tyngdpunkt) så gick det ganska enkelt att identifiera en maximal konstant-hastighet. Med konstant-hastighet menas den hastighet som bilen kan ha hela tiden utan att köra av banan. Den är alltså begränsad av vilken hastighet bilen kan ha i den skarpaste/svåraste kurvan av banan. När denna maximala konstant-hastigheten hade identifierats så kunde man observera att AI'n mycket riktigt kunde optimera körning till den grad att högre hastigheter kunde uppnås än konstant-hastigheten så länge den justerade hastigheterna inför/i kurvan. Som kan ses i bild 25 skrivs hastigheten som maskinen väljer ut och i det specifika fallet var den maximala konstant-hastigheten 0.39. I det specifika fallet som demonstreras i bilden så hade alltså bilen kört av i kurvan om den hade behållit 0.42 som hastighet. Den uppmätningen är inte tillräcklig för att konstatera att maskinen kör snabbare än människan men den är ett kvitto på att maskinen mycket riktigt kan optimera baserat på en prediktion samt tolka och tackla utmaningen på sitt sätt, utan vidare "förklaring" från skaparna.

```
Increasing speed to: 0.42
Increasing speed to: 0.42
Increasing speed to: 0.42
Increasing speed to: 0.42
Increasing speed to: 0.42
Lowering speed to: 0.41
Lowering speed to: 0.399999
Lowering speed to: 0.389999
Lowering speed to: 0.379999
Lowering speed to: 0.369999
Lowering speed to: 0.359999
Lowering speed to: 0.349999
Lowering speed to: 0.339999
Increasing speed to: 0.349999
Increasing speed to: 0.359999
Increasing speed to: 0.369999
Lowering speed to: 0.359999
Increasing speed to: 0.369999
Increasing speed to: 0.379999
Increasing speed to: 0.389999
Increasing speed to: 0.399999
Increasing speed to: 0.41
Increasing speed to: 0.42
Increasing speed to: 0.42
```

Figur 25: Modellens förmåga att anpassa hastigheten inför/i en kurva.

6 Diskussion

Projektet i sin helhet anser vi har uppnått sitt syfte. Vi har lyckats visa att det kan vara möjligt för Ekkonos AI att köra snabbare än en människa, under rätt förutsättningar. AI-modellens förmåga att uppskatta värdet på bilens roll-acceleration var tillräckligt bra för att styra bilen redan efter elva sekunder, detta var över vår förväntan. Modellen blev sedan bättre och bättre på att uppskatta värden desto längre den körde. Detta var väntat då modellen fick mer data att basera sina beslut på. Vad det gäller utrustningen så har den varit enkel att arbeta med och prestandan har varit tillräcklig. Det finns dock optimeringar att genomföra som skulle förbättra prestandan hos systemet. Vikten på systemet skulle kunna optimeras genom att byta ut Sense HAT till en accelerometer och det finns möjlighet att minska motorstyrningens vikt. Strömmkällan för logiken, alltså datorer och övriga kretskort, skulle kunna tas från banan - vilket då eliminerar dess beroende och påverkan av sladden från powerbanken. Den absolut största förbättringen som skulle kunna göras inom systemet är dock reglerloopen. Då tiden vid projektets slut inte tillät en implementering av en PID-regulator eller liknande styrsystem så tror vi att en stor optimering kan göras inom detta område.

6.1 Maskinen mot människan

Det huvudsakliga syftet med arbetet var att undersöka om Ekkonos AI kunde optimera körningen av bilen på ett sådant sätt att den körde bättre än en människa. Resultatet av jämförelsen mellan AI-bilen och den styrd av en människa visade att människan var bättre. Men den bästa heattiden sattes av bilen styrd av AI. Även om vi inte lyckades få AI-bilen att konsekvent köra snabbare än människan så visade vi att det var möjligt. Men hur rimligt är det resultatet egentligen? Urvalet av personer som körde bilbanan var begränsat till två personer, alltså visar resultatet deras snitttider. För att få ett bättre genomsnittstid för en människa skulle ett större antal förare behöva undersökas. Ett annat argument emot hurvida AI-modellen kunde köra snabbare än en människa är att den bästa bilbaneföraren inte körde bilen, men samtidigt kör AI-bilen förmodligen bättre än den sämsta. Det finns en stor gråzon i uttrycket "bättre än en människa" men vi anser att i detta fallet handlar det om en genomsnittlig människas förmåga. De flesta personer kör inte dagligen bilbanor och kommer förmodligen att köra av bilbanan om de försöker köra så snabbt de kan. Detta var något som vi märkte under undersökningen av viktpåverkan. Det var svårt att köra så snabbt som möjligt i bara 10 varv utan att trilla av, medans AI-bilen kunde klara det utan problem gång på gång. Det tog bilen ungefär elva sekunder att lära sig köra bilen med varvtider nära vår egen, vilket för oss var förvånansvärt bra. Detta är också ett argument för att den varvtiden som bör användas för människan ska vara ett genomsnittsvärde, eftersom den tidigare erfarenheten som människan och AI-bilen har av att köra bilen är likvärdig.

Den tid det tog för AI-modellen att tränas tills den var tillräckligt bra för att användas till att styra bilen var oss förvånansvärt låg. Som tidigare nämndes tog det bara elva sekunder för bilen att behandla tillräckligt

mycket data för att börja styra bilen själv utan att åka av. Då låg dess uppskattning inom 10% av det verkliga värdet. Detta blev sedan bättre desto längre bilen körde på banan, men optimeringar kring AI på bilen finns det utrymme för. Det finns många fler AI-algoritmer, och förmodligen en som hade fungerat bättre än Random forest. Men det var den algoritmen som vi blev rekommenderade att börja med eftersom den är enkel att komma igång med om man inte hade tidigare kunskaper inom AI. Hade vi haft mer tid till implementeringen av AI modellen hade vi gått över till att använda multilayer perceptron.

AI-bilen lyckades inte att köra i närheten av den potentiella maxhastigheten. Bilen styrdes av en PWM signal som kunde sättas mellan 0 och 1, där 1 är maximal gaspådrag. Bilen körde strax över den högsta möjliga hastigheten den kan hålla utan att köra av i en kurva, vilket var runt 0,39. Men den körde fortfarande så pass fort att den skulle åka av om den inte anpassade farten i kurvorna. Detta visade att AI-modellen kunde förutspå när kurvor skulle ske, detta märktes genom att bilen sänkte farten när den åkte in i kurvan och höjde den på vägen ut.

Mycket finns kvar att optimera för att få bilen att köra snabbare. Att starta projektet från det utgångsläge vi har nu hade varit önskvärt. Hade vi hunnit komma till stadiet där vi skulle implementera AI-modellen och reglerloopen tidigare hade vi kunnat uppnå ett mycket bättre resultat. Den sista delen av projektet lades fokus i första hand på AI-implementering, vilket lämnade väldigt lite tid till att konstruera en ordentlig reglerloop. Vid en fortsättning av detta projekt bör detta införas, förslagsvis en PID-regulator.

6.2 Utrustningen

Den utrustning som används under projektet har fungerat bra för dess syften. Den har varit simpel att använda och bra som utgångspunkter att lära sig ifrån. Ibland är det bäst att lära sig simma i den grunda delen av poolen innan man ger sig in på den djupa änden. Detta är något som vi har applicerat i detta projekt, vilket har varit till stor hjälp. Kring RP-Zero sågs inga förbättringsmöjligheter för detta projektet men Sense HAT och motorstyrningen lämnade möjligheter för förbättringar, men dessa krävde kunskap utanför vårt scope för att genomföra.

6.2.1 Raspberry Pi Zero W

Raspberry Pi Zero har varit en enkel dator att arbeta med - både under installationsprocessen, utvecklingen och testningen. Tack vare SSH-uppkopplingen kunde allt arbete ske på en utvecklingsdator vilket var en väldigt stor hjälp och tidssparande aspekt. Alternativet hade varit att koppla in sig via micro-USB eller använda startuprutiner som initierar testerna, men då måste dator startas om mellan varje test. Prestandamässigt har processorkraften hos RP-Zero inte varit något hinder. Enligt undersökningen av tidsfördröjningar i det slutgiltiga programmet, där AI-beräkningar och reglerloopen ingick, visade resultatet att regleringen av motorn skedde med en frekvens på 45 Hz. Datorn kan alltså ta ett beslut för hurvida den ska ändra bilens hastighet på 0,02 sekunder. Detta är långt under en vanlig mänskligas reaktionstid, som ligger runt 0,2-0,3 sekunder [6].

6.2.2 Raspberry Pi Sense HAT

Raspberry Pi Sense HAT har fungerat som en bra utgångspunkt för projektet vad det gäller givare. Den var enkel att komma igång med då den är byggd för att fungera direkt med RP-Zero. Läsastigheten hos givarna, som uppmättes till 50 Hz, var tillräckligt snabb. Däremot visade undersökningen av datan från givarna att den mest troligtvis behandlades till viss del av Sense HAT. Detta görs förmodligen för att få en jämnare kurva, vilket kan minska brus men med kostnaden av snabb reaktionstid vid plötsliga ändringar. Datan var dock tillräckligt snabb och gav tillräckligt stora utslag för att kunna avgöra när en bilen körde in i en kurva och därmed ta ett beslut att sänka hastigheten. Vid vidare utveckling av projektet hade dock Sense HAT ersatts med endast en accelerometer. Dels för att kunna göra avläsningar av obehandlad data och därmed få större noggrannhet på bilens acceleration, men också på grund av vikten och storleken. Den var klumpig och svår att montera ordentligt på bilen, ett mindre kort hade gjort denna process mycket enklare.

6.2.3 L298 H-brygga

L298 h-bryggan som användes till att styra bilens motor har fungerat bra. De instruktionerna som fanns att tillgå gjorde inkopplings och testningsprocessen av h-bryggan enkel. H-bryggan är dock överdimensionerad utifrån de uppmätta värden för matningsspänningen och den maximala strömförbrukningen av motorn. Den hanterar spänningar mellan 5-35 V och tål 2 A medans det på bilbanan uppmättes 15 V och 800 mA. Kortet är dimensionerat för att fungera under de maximala förhållandena vilket gör att värmeutvecklingen måste tas hand om. Detta sköts av en stor kylfläns som gör ett väsentligt bidrag till kretskortets vikt, vilket påverkar

bilens köregenskaper. Men denna modell var det bästa alternativet av de tillgängliga utbudet. Om möjligt rekommenderas det i ett framtida projekt att försöka hitta en mer optimerad h-brygga som har en mindre vikt. Det är också möjligt, om man har tiden och kunskapen, att konstruera en egen optimerad motorstyrning utifrån de förhållanden den ska arbeta under.

6.3 Val av givare och axlar

Vilka givare som behövs för projektet kan variera beroende på krav och noggrannhet. Efter att Sense HAT hade installerats så skrevs testrutiner för att läsa av de olika sensorerna och ganska snabbt kunde det konstateras att accelerometern, som givare, skulle räcka. En anledning till att ta hjälp av en kompass skulle vara att hålla reda på ett helt varv. På så sätt skulle bilen själv kunna veta om när den har kört ett helt varv (360 grader). Men eftersom ingen träning skulle vara varvspecifik och tävlingen mellan bilarna inte nödvändigtvis avgörs på varv så används ej kompassen. Det är en bra tanke att bibehålla för vidare fortsättning av projektet dock.

Efter att givare är valt så undersöktes de olika axlarna: roll, pitch och yaw, se figur 5 i sektion 2.6.1. Redan i början av projektet diskuterades just roll-axeln, alltså bilens rotation runt den längsgående axeln. Detta för att det uppskattades vara den största anledning till avkörning. Om bilen har för mycket kraft i sidled så kommer den att rulla av. Pitch behandlar upp- och nedaxeln, motsvarande utslag som skulle fås av upp- eller nedförsbackar. Eftersom bilen, mellan framhjulen, har en plastfena som håller kvar den i spåret (där strömförsörjningen finns) så uppskattades även yaw-axeln att vara sekundär. De första testen som genomfördes, se sektion 5.4, gjordes direkt med roll i fokus. Resultaten som avlästes var tillfredsställande och passade hypotesen bra.

En aspekt att ta hänsyn till i avläsning av värdena är bilens konstruktion, då val kring denna kan påverka. Eftersom prototypen är uppbyggd med papp-kartong så finns det en viss rörlighet i konstruktion, detta påverkar t. ex. roll-värdet på avläsningar. Eftersom papp-taket, där accelerometern är fäst, kan vrida på sig så kommer ett högre gradtal avläsas än vad faktiskt bilen upplever. En visst kalibrering kan behövas göra men i stora drag så kommer roll-förändringarna vara relativa varandra vilket innebär att det är skillnaden i roll som kommer vara mer avgörande än det faktiska värdet.

Modellen matas alltså med fyra stycken parametrar och producerar ett resultat. Modellen får tillgång till den nuvarande motor-signalen, pitch, yaw och roll. Ju fler parametrar desto bättre för en bättre uppskattning. Resultatet modellen ger är den uppskattade roll-en. Tanken här blir att om modellen får dessa parametrar så ska den uppskatta ett roll-värde för någon sekund fram i tiden. Baserat på denna uppskattningen så ska det tas ett beslut.

6.4 Elektriska störningar och mätningar

Vår kunskap inom området elektroteknik är mycket begränsad och vi har under projektets gång fått förlita oss på vår handledares vägledning inom detta område. Att störningar enkelt kan uppstå är kanske inte något man tänker på utan att vara insatt i området. De elektroniska produkter vi använder oss av idag måste åtgärda dessa störningar för att ens få säljas eller användas. Därför är det inte ett vanligt förekommande fenomen som vi upplever i vår vardag, men något som är viktigt att åtgärda vid konstruktionen av en elektrisk krets. Eftersom bilbanan är en kommersiell produkt antogs att en viss filtrering av störningar redan var installerad men att den inte skulle filtrera bort alla störningar. Resultatet av mätningarna visade att detta var fallet. Dessa störningar behövde dock inte åtgärdas då de inte var en fara för vårt system. Men en möjlig förbättring om detta projekt ska tas vidare är att RP-Zero tar sin strömförsörjning från spåret. Detta gör att bilen inte behöver kopplas med en sladd från en powerbank som hålls innanför spåret under körning, och det minskar vikten jämtom att montera powerbanken på banan. För att genomföra detta behöver de uppmätta störningarna i form av spänningsspikar på 50 V hanteras, då de kan leta sig in bakvägen i RP-Zero och skada den. En annan åtgärds punkt vid denna lösning är att hantera det rippel som uppstår i banan då strömförsörjningen är DC spänning konverterad från AC. För stort rippel kan skada RP-Zero genom att den får för hög spänning eller att spänningen blir så låg att den stängs av.

6.5 Massans påverkan

Att utrustningens massa skulle påverka bilen så att den körde långsammare var något som vi vid projektets utgångspunkt var medvetna om. Däremot hur mycket den skulle påverkas hade vi ingen uppskattning av. Dels för att vi inte visste, och fortfarande inte vet, alla de faktorer som påverkar hur snabbt bilen kan köra runt banan. Hela arbetet skulle kunna handla om just denna punkt, om man tillåter sig gå in på detaljnivå. Vi valde att utgå från massans påverkan på accelerationen, eftersom den är direkt märkbar då bilen körs. Enligt teorin kommer en fördubbling av massan att halvera accelerationen, men detta betyder inte att den hastighet som

bilen kan uppnå halveras, det tar bara längre tid att uppnå den. Detta är märkbart i figur 19 där en fördubbling av massan inte fördubblar varvtiden.

Men andra faktorer som friktion, bilens grepp och tyngdpunkt har också betydelse för detta och påverkas av hur stor och vart på bilen massan placeras. Eftersom utrustningen placerades på en plåt ovanför bilens motor gjorde detta att tyngdpunkten hamnade högt upp. Detta påverkade i sin tur hur snabbt bilen klarade att köra i en kurva innan den välte, en lägre placerade tyngdpunkt hade tillåtit bilen att köra snabbare. Det hittades dock inget enkelt sätt att montera utrustningen direkt på bilens underrede utan att permanent fästa den eller riskera att skada den. En möjlig lösning som sågs, men inte kunde implementeras i detta projektet, var att 3D-printa en fixtur. Denna skulle sedan kunna fästas i bilen varpå utrustningen fästs i fixturen.

6.6 Etik och hållbarhet

De kopplingarna till etik och hållbarhet som har stötts på under projektets gång har huvudsakligen handlat om självkörande bilar. Då bilen i det här projektet har körts i ett spår så har vi endast tagit hänsyn till ökning och minskning av hastighet. Däremot har AI använts för att förutspå kurvor och detta är något som väldigt tydligt går att koppla till de ”vanliga” självkörande bilarna som utvecklas idag. Om AI kan användas för att förutspå kurvor så kan två viktiga aspekter inom etik och hållbarhet tas hänsyn till, nämligen säkerhet och energiförbrukning.

Säkerheten handlar om eventuella hastighetsjusteringar som kan göras inför en kurva. I vårt fall var det för att inte köra av kurvan vilket direkt kan kopplas till ”verklig säkerhet”. Om ett AI system skulle upptäcka att den hastighet som föraren håller skulle vara så pass hög att den riskerar att köra av vägen kan bilen sänka hastigheten för att undvika en olycka. Vidare så kan även energiförbrukningen optimeras om kurvor kan förutspås. Även om vårt projekt endast behandlade kurvor så finns det självfallet fler ”hinder” och föremål än bara kurvor i den verkliga trafiken. Genom att samla in mer typer av data än den vi samlade in i projektet så kan säkerligen många av dessa predikteras.

Man kan också debattera för att kartor och navigationssystem är en tryggare informationskälla än just en AI-prediktion när det kommer till kurvor, korsningar och så vidare. Däremot skulle AI kunna användas i mer nischade och detaljerade användningsområden så som trafik-belastning eller motorförändringar. Om AI används för att prediktera körarens beteende i olika trafik-sammanhang (vilket inte ett navigationssystem kan göra) så kan både säkerhet och energiförbrukning förbättras. AI skulle kunna användas till att förutspå trafikmönster i största allmänhet och därför hjälpa föraren att avgöra vilka tider på dagen som är bränslesnål (minst tomgångskörning) eller säkrast (minst antal bilar i trafiken).

Med tanke på hur många fler delar av en självkörande bil som uppkopplas, jämfört med en traditionell bil, så är användningsområdet för en AI-prediktion stort. Ovan har vi nämnt ett par, vissa generella och vissa nischade, exempel. Vi tror nog att de flesta säkerhetsaspekterna som finns kommer att tas hand om lite mer absoluta och exakta system och således så har förmodligen AI-prediktionerna sin största roll just inom energiförbrukning.

7 Slutsats

Syftet med projektet var att undersöka om en bil styrd med hjälp av Ekkonos AI kunde optimera sin körning så att den körde snabbare än en människa. Detta anser vi vara möjligt då vi mätte varvtider där AI-bilen körde snabbare än en bil styrd av en människa, trots att reglerloopen var långt ifrån optimal. Vi lyckades dock inte uppfylla den del av vårt huvudsakliga mål som var att AI-bilen skulle köra snabbare än vad en människa kan klara av. Vi lyckades vid ett tillfälle göra detta men för att uppfylla vårt mål hade bilen behövt vara oslagbar, vilken den inte är. Vi anser att detta hade varit möjligt vid implementering av en bättre reglerloop. Bilen klarar dock att köra utan att veta hur banan ser ut i förväg och gör detta med hjälp av Ekkonos AI. Så dessa delar av det huvudsakliga målet är uppnådda.

De övriga målen som vi satte upp för projektet lyckades vi uppnå. Vi skapade en elektrisk krets som tillåter en Raspberry Pi Zero att styra bilen, och den utsätts inte för några störningar från bilbanan. Frekvensen som data kunde samlas från Raspberry Pi Sense HAT var tillräckligt hög för att bilen skulle kunna styras med Ekkonos AI. Men det finns utrymme för optimering kring utrustningen vid framtida projekt. Datan som samlades från Sense HAT behandlades vilket gjorde att plötsliga förändringar inte kunde detekteras då mätningar gav ett medelvärde. Genom att använda en annan accelerometer kan noggrannheten, och responsiviteten på mätdatan öka. Vikten på komponenterna var också av betydelse och kunde minskas. Det visade enligt testerna att vikten påverkar bilens varvtid negativt, vilket var väntat. Därför rekommenderas, om det är möjligt, att välja en accelerometer och motorstyrning som väger mindre vid ett framtida projekt. En annan rekommendation är att försöka placera komponenterna så lågt på bilen som möjligt, detta säker så tyngdpunkten och gör att bilen kan köra snabbare i kurvorna.

Sammanfattningsvis upplever vi att syftet med projektet uppnåddes, men resultatet hade stort utrymme för förbättring. Den största rekommendationen vi vill skicka vidare till ett framtida projekt är att införa en bättre reglerloop för att styra bilens motor, genom att exempelvis införa en PID-regulator. Vi rekommenderar också att välja ett mindre och mer responsivt accelerometerkort än Raspberry Sense HAT. Om detta lyckas införas och bilen kan köra snabbare än en människa utan att åka av banan är vår sista rekommendation att ta strömförsörjningen till Raspberry Pi Zero från bilbanans spår, så att den inte är beroende av en utomstående strömkälla.

Referenser

- [1] S. Russell and P. Norvig, *Artificial intelligence : a modern approach*. Harlow, United Kingdom: Pearson Education Limited, 2016.
- [2] H. Robotics. (Apr 23 2020) Sophia. [Online]. Available: <https://www.hansonrobotics.com/sophia/>
- [3] C. Nordling and J. Osterman, *Physics Handbook for Science and Engineering*. Lund, Sweden: Studentlitteratur, 2006.
- [4] R. Pi. (Apr 13 2020) Raspberry pi zero w. [Online]. Available: <https://www.raspberrypi.org/pi-zero-w/>
- [5] Pinout. (Apr 20 2020) Pinout! [Online]. Available: <https://pinout.xyz>
- [6] Humanbenchmark. (May 20 2020) Reaction time statistics. [Online]. Available: <https://humanbenchmark.com/tests/reactiontime/statistics>

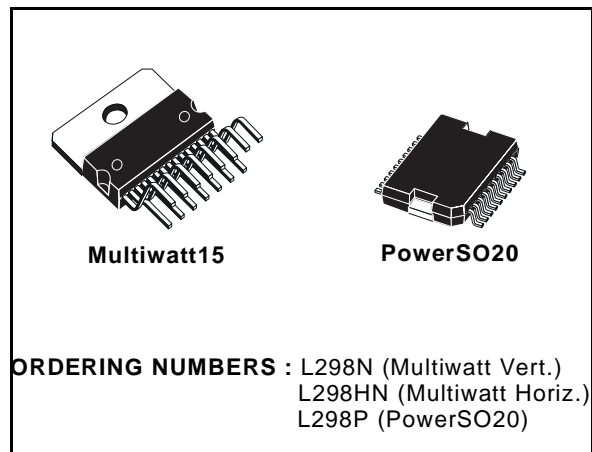
A Datablad för L289 H-Brygga

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

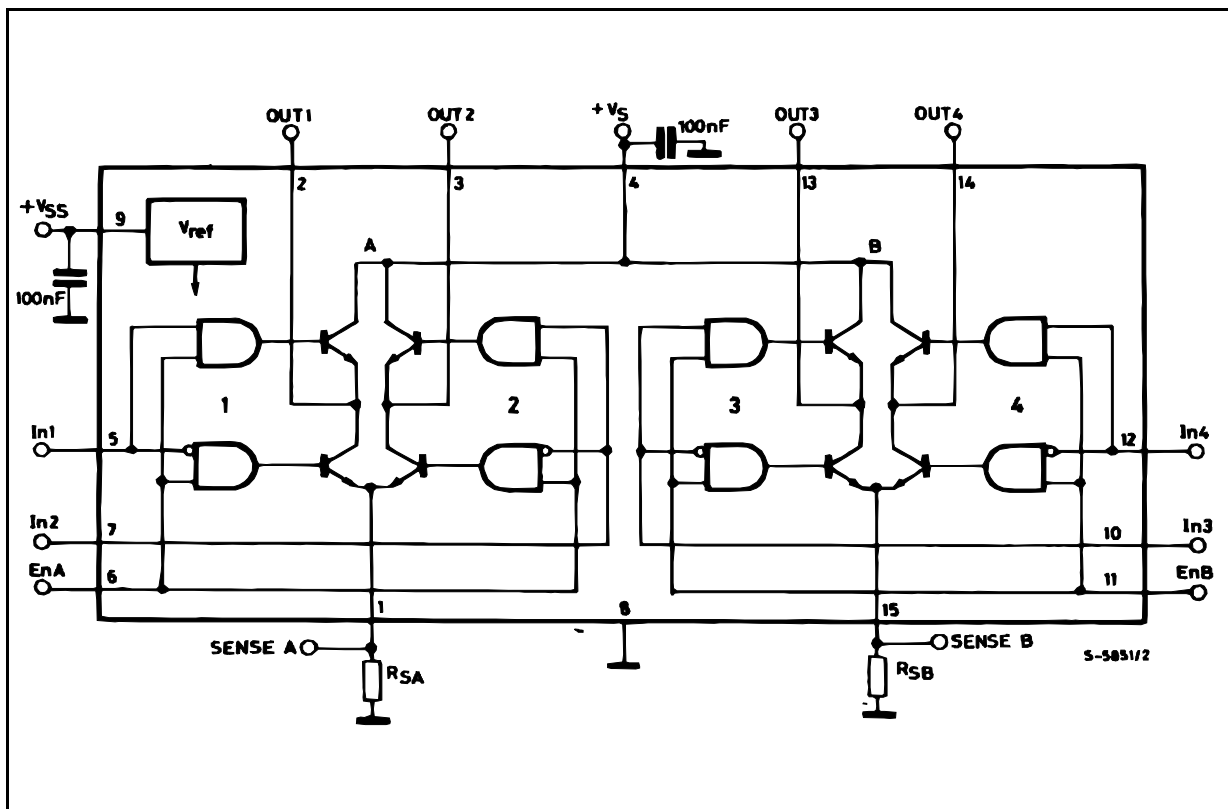
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

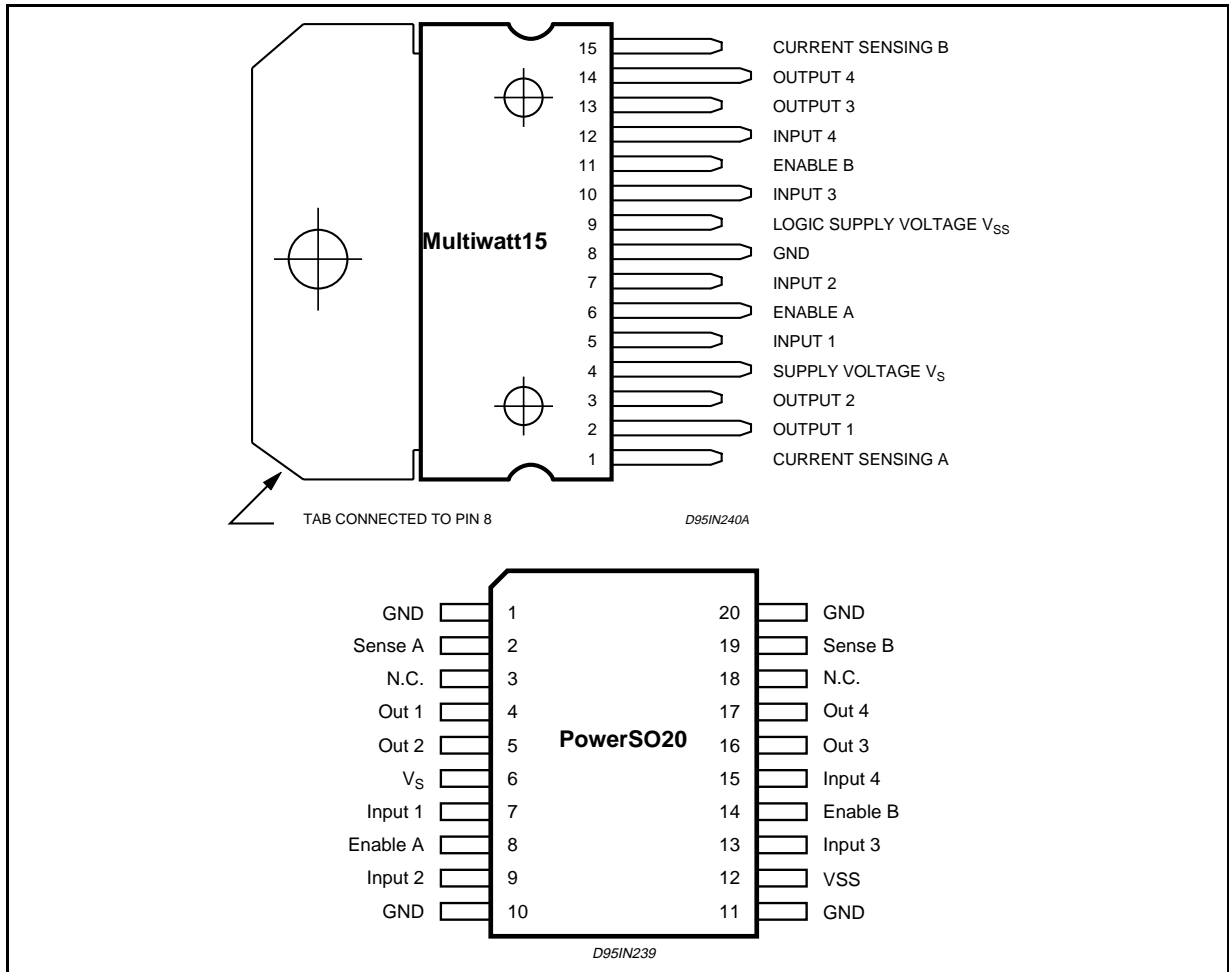
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			–10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} –0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			–10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} –0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

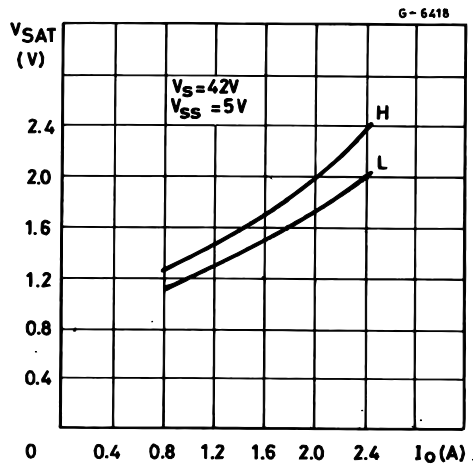
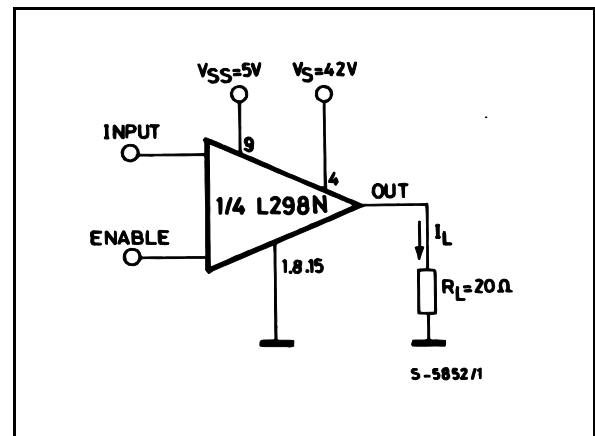


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

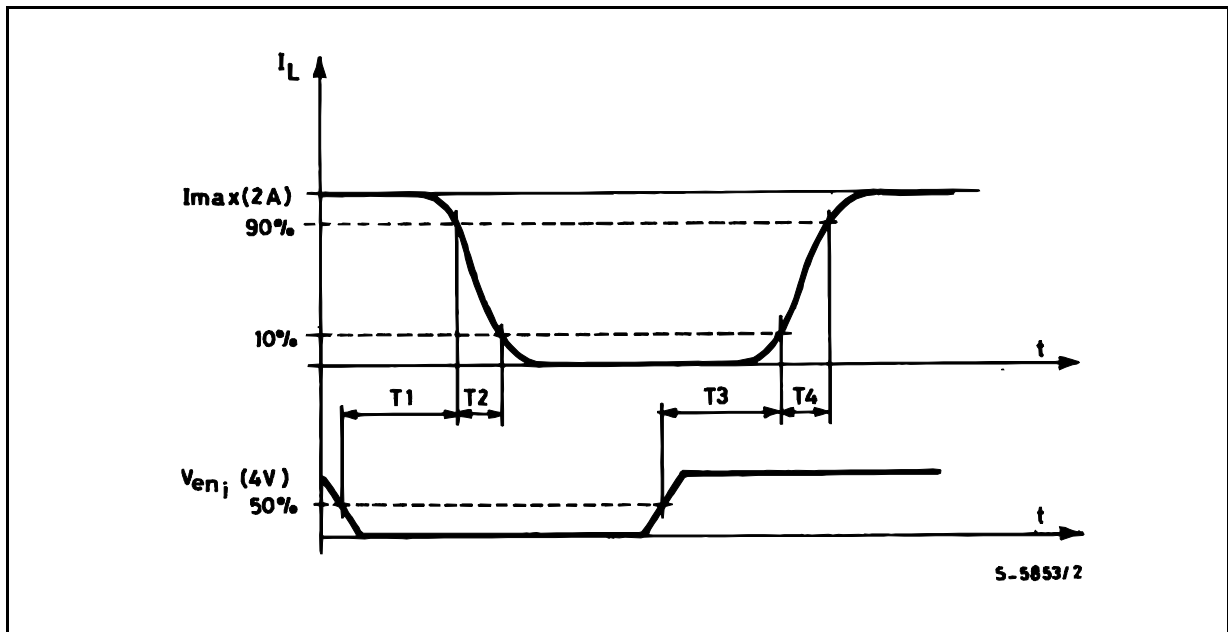
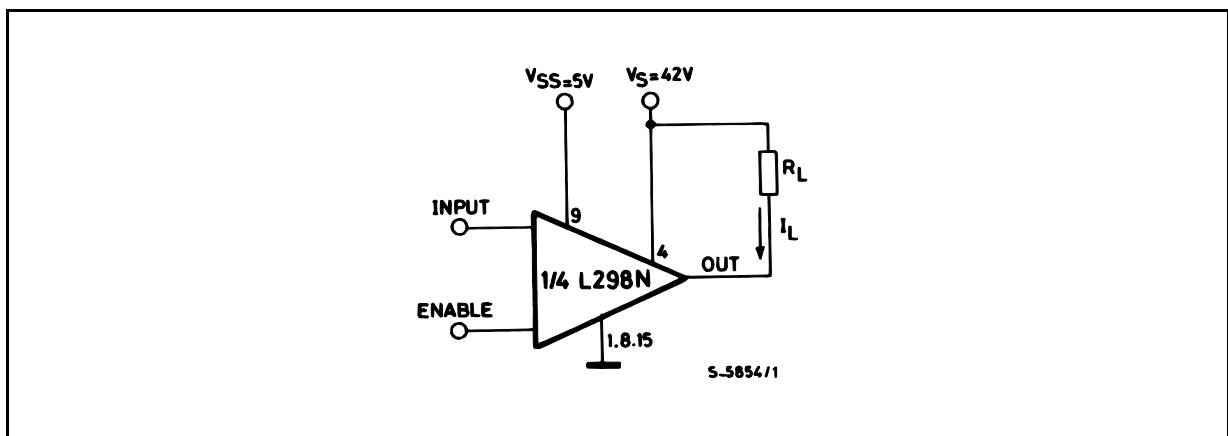


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

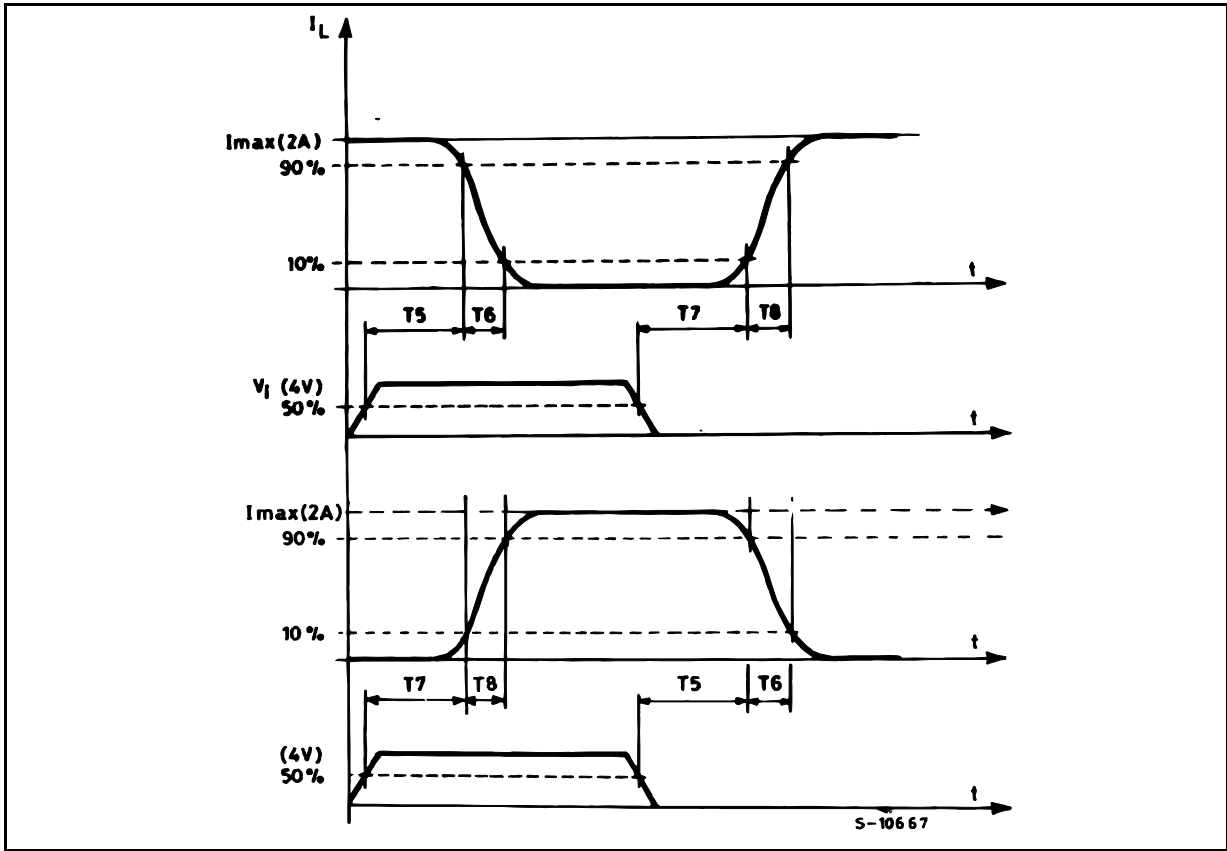


Figure 6 : Bidirectional DC Motor Control.

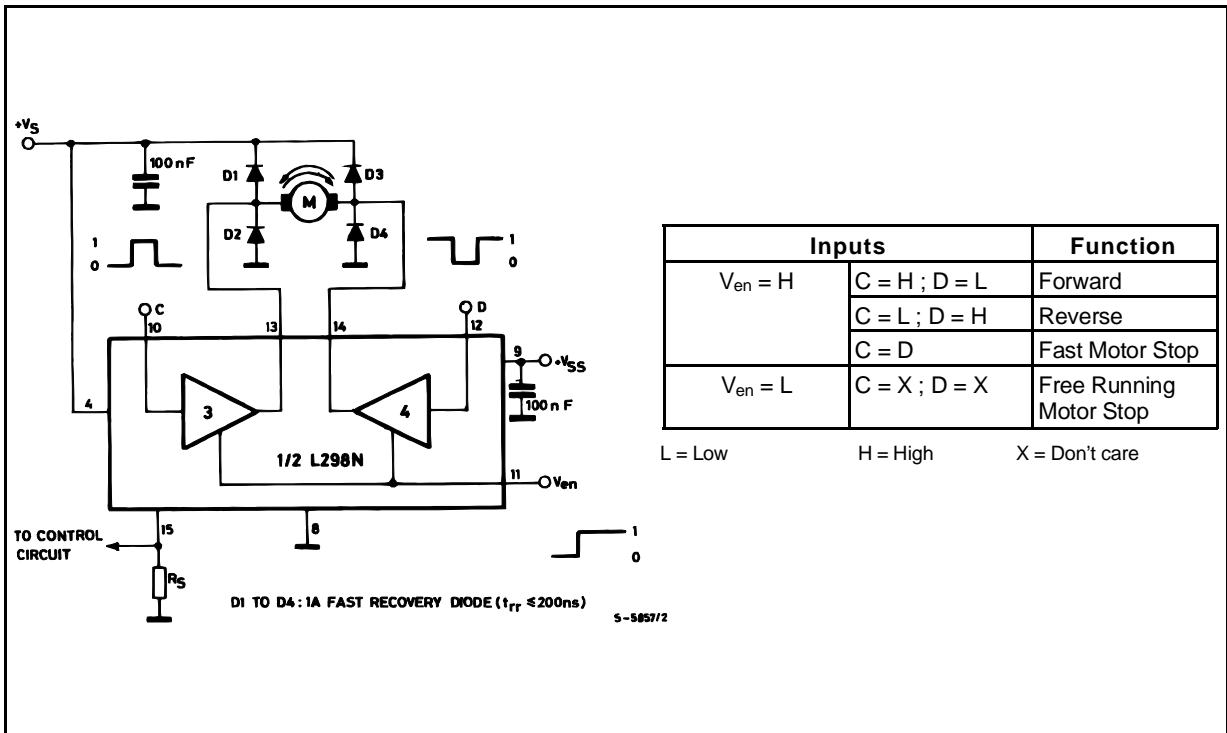
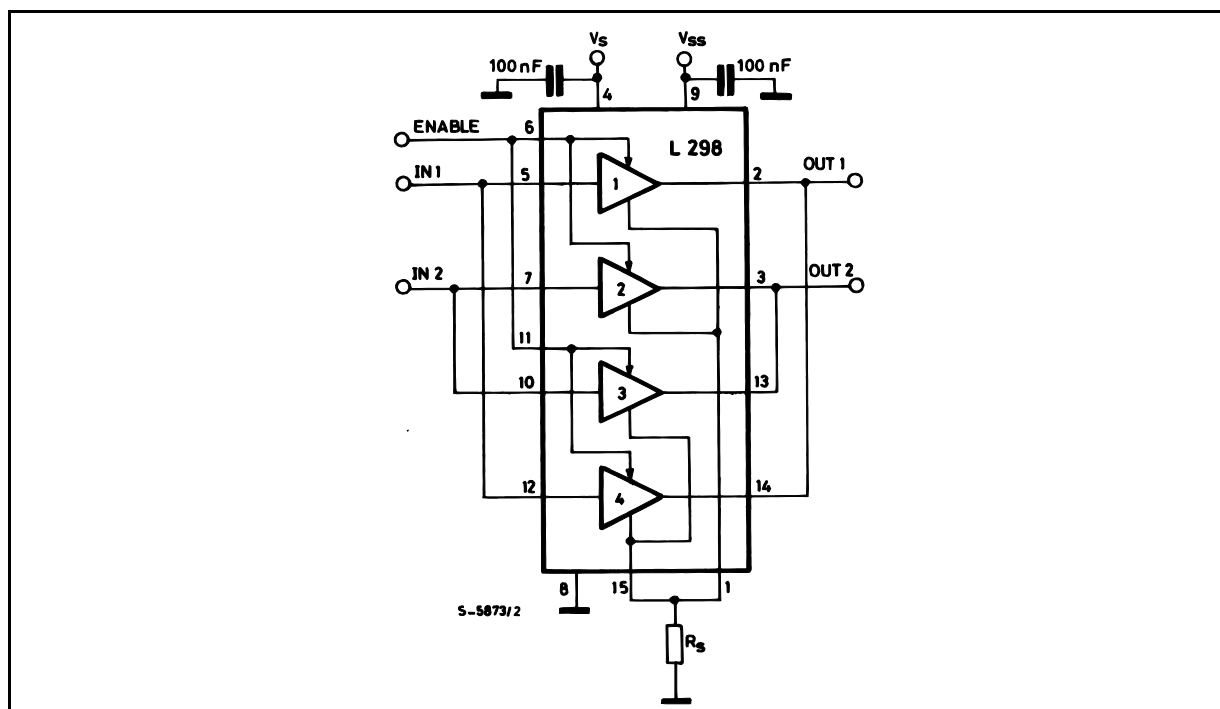


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In_1 ; In_2 ; EnA and In_3 ; In_4 ; EnB . The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($trr \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

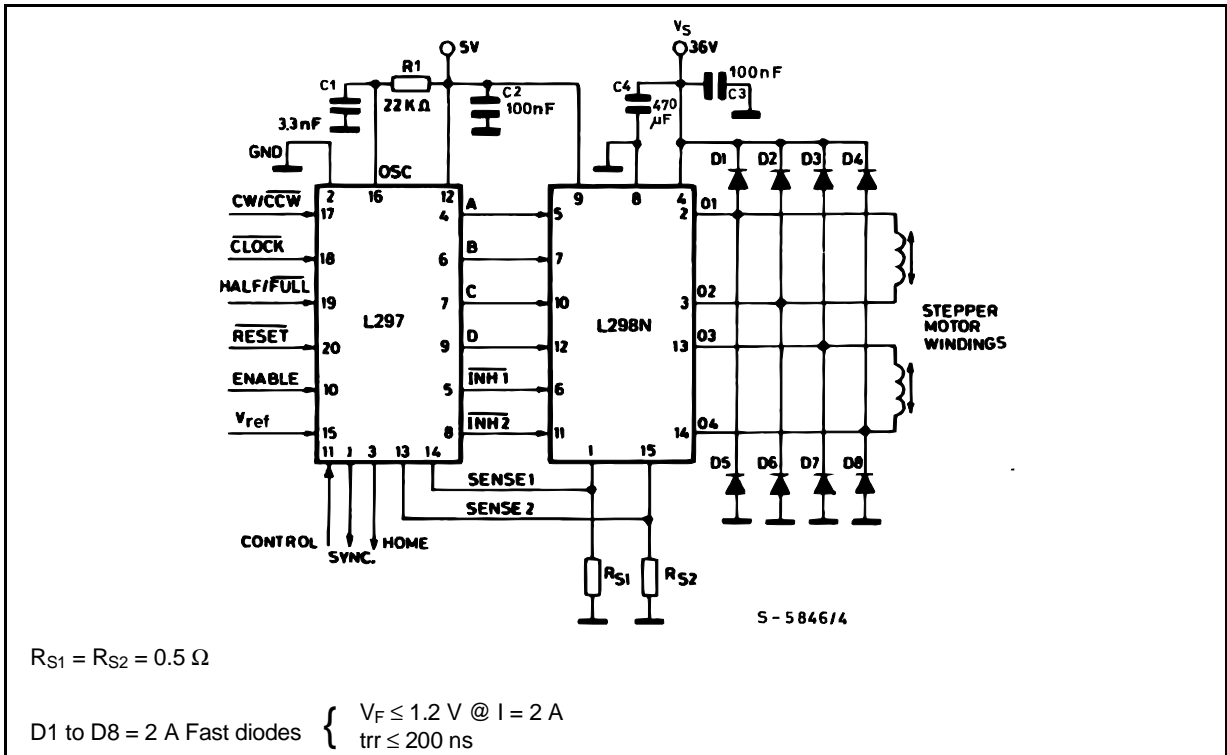


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

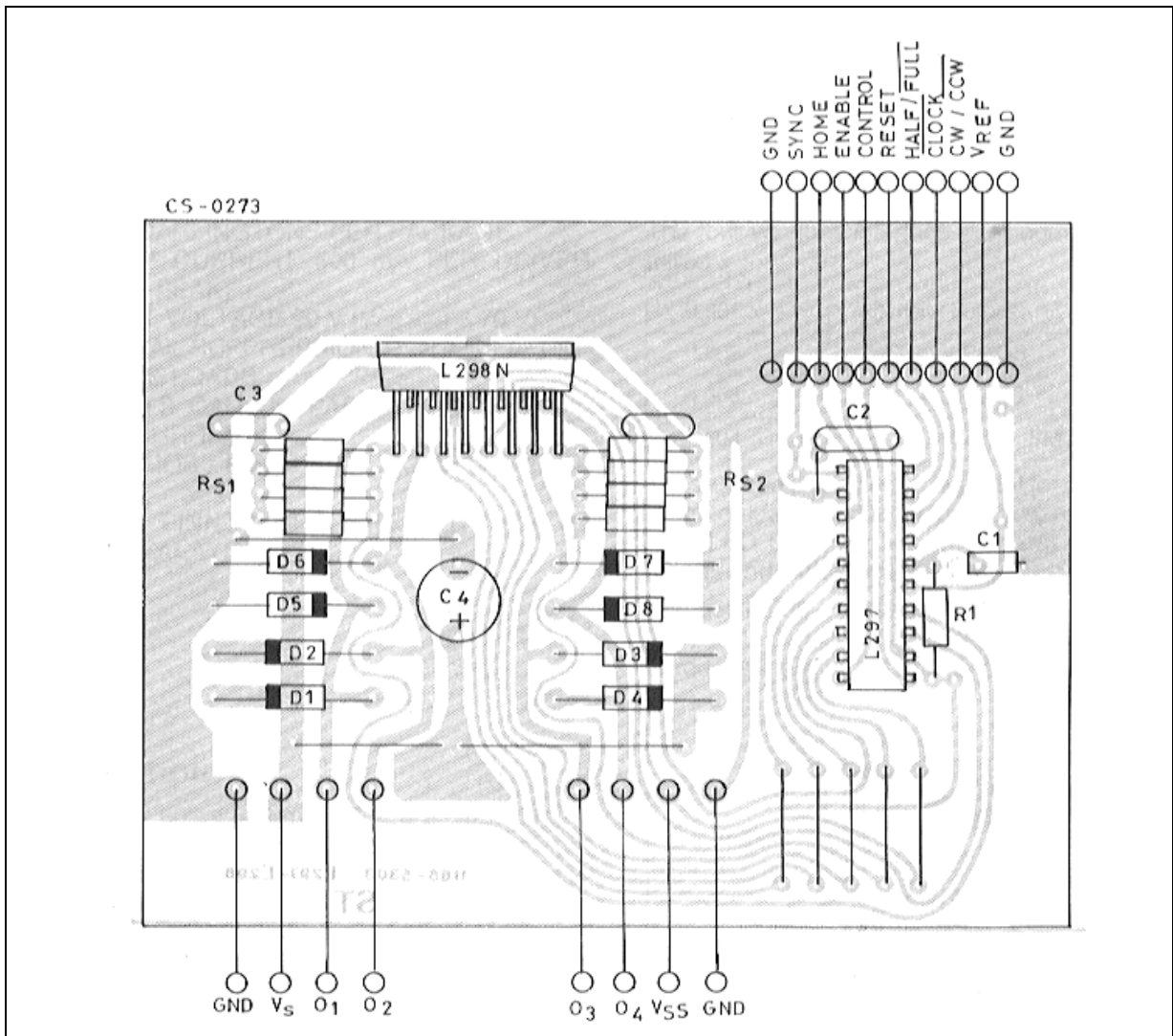
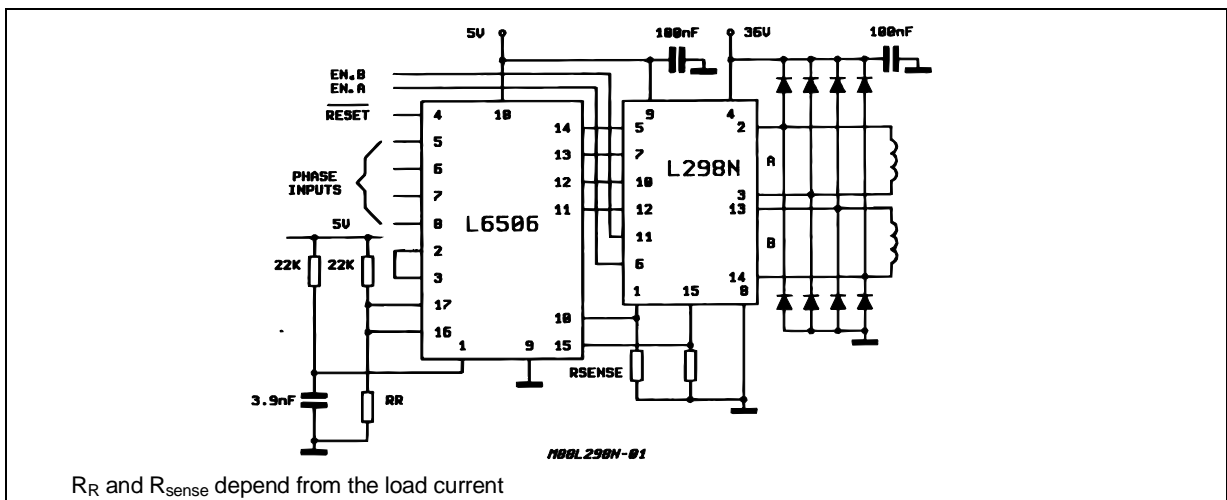
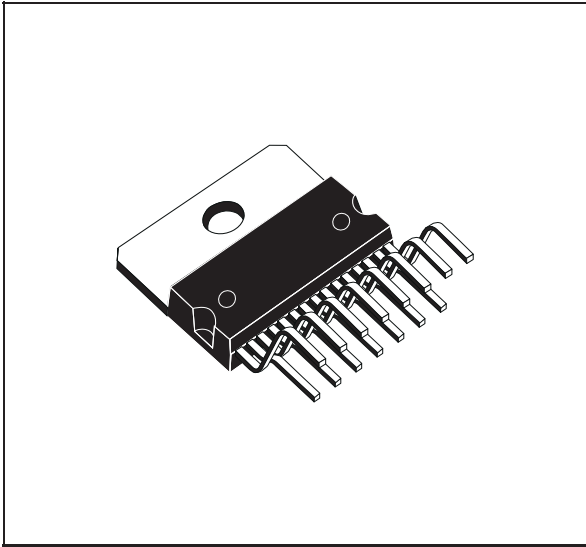


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

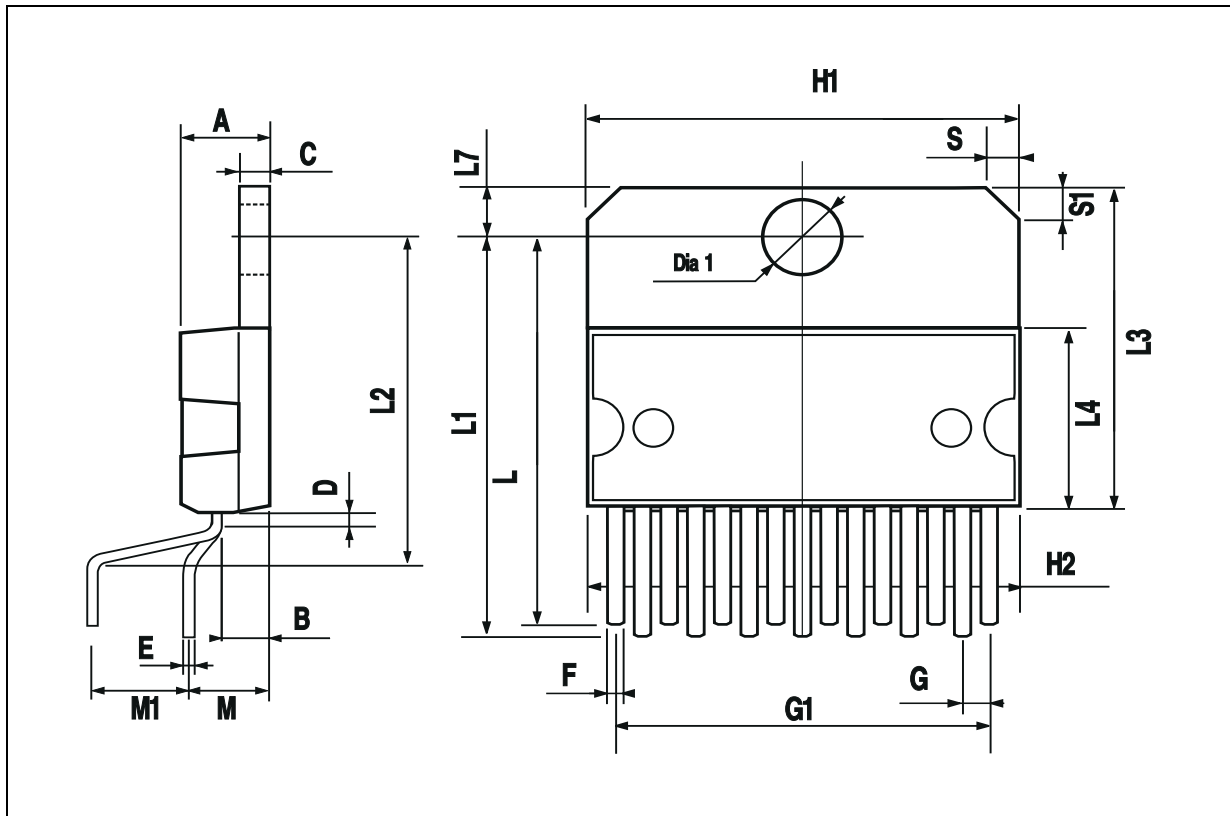


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

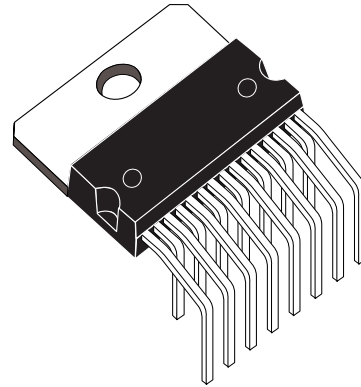


Multiwatt15 V

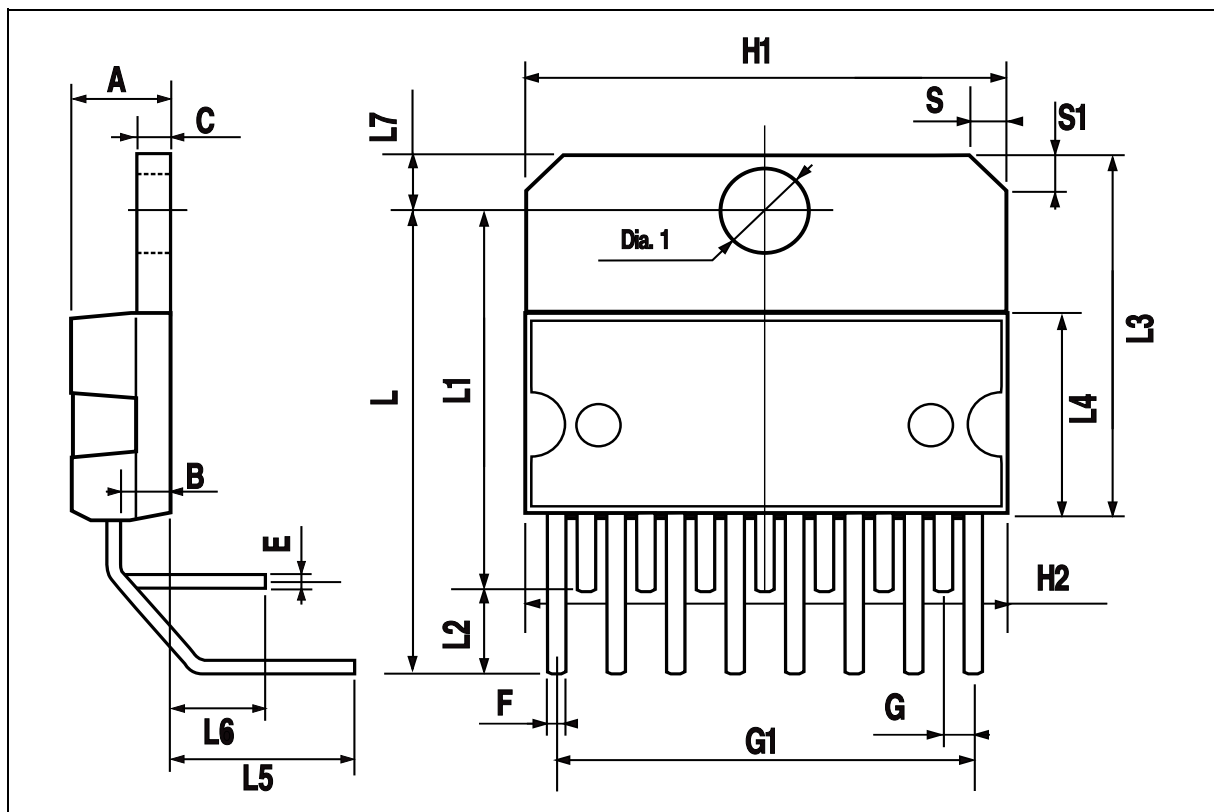


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



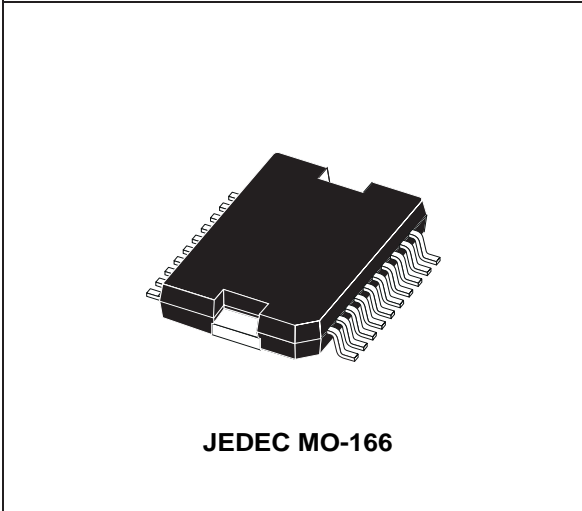
Multiwatt15 H



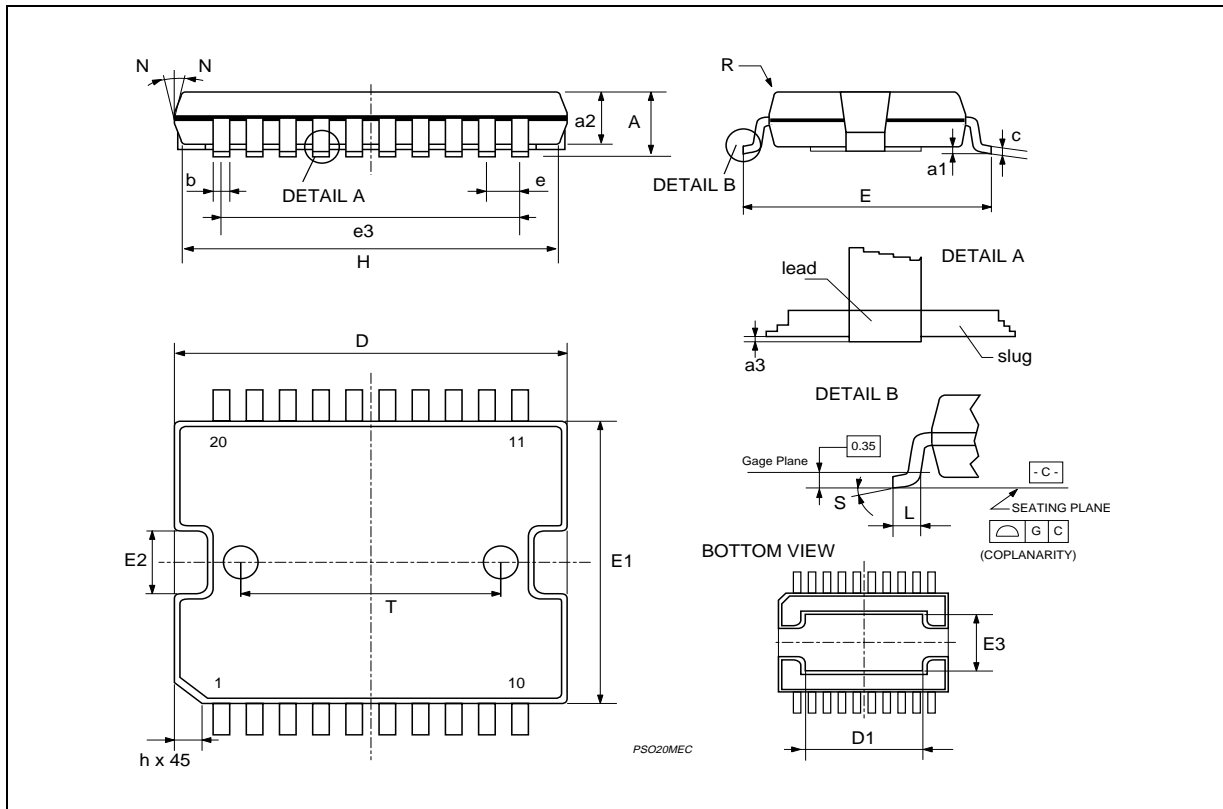
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>