
DIGITAL WORKSHOP

Remote Busdriver

Skrivet av: Ajla Cano, Filip Lindahl
CHALMERS 2017



Bakgrund

Idén med en självkörande bil har förut endast varit en vision men börjar bli mer och mer realistisk och efterfrågad. Det finns många fördelar med en självkörande bil; trafiken kan potentiellt bli mer säker, långa trafikköer skulle minska på grund av den optimerade körningen, mäniskor som inte är lämpade för körning behöver inte bero av andra mäniskor som kan köra och istället förlita sig på de autonoma bilarna.

Kollektivtrafik kan också förbättras av självkörande bilar genom att ersätta de "vanliga" bussarna med självkörande bussar. Precis som med allt annat, så kan det här området uppleva problem. Om programmet (Artificial Intelligence AI) upplever svårigheter eller ett problem som programmet inte känner igen och inte vet hur det ska agera, då kommer det behövas en plan B att falla tillbaka på.

Idén

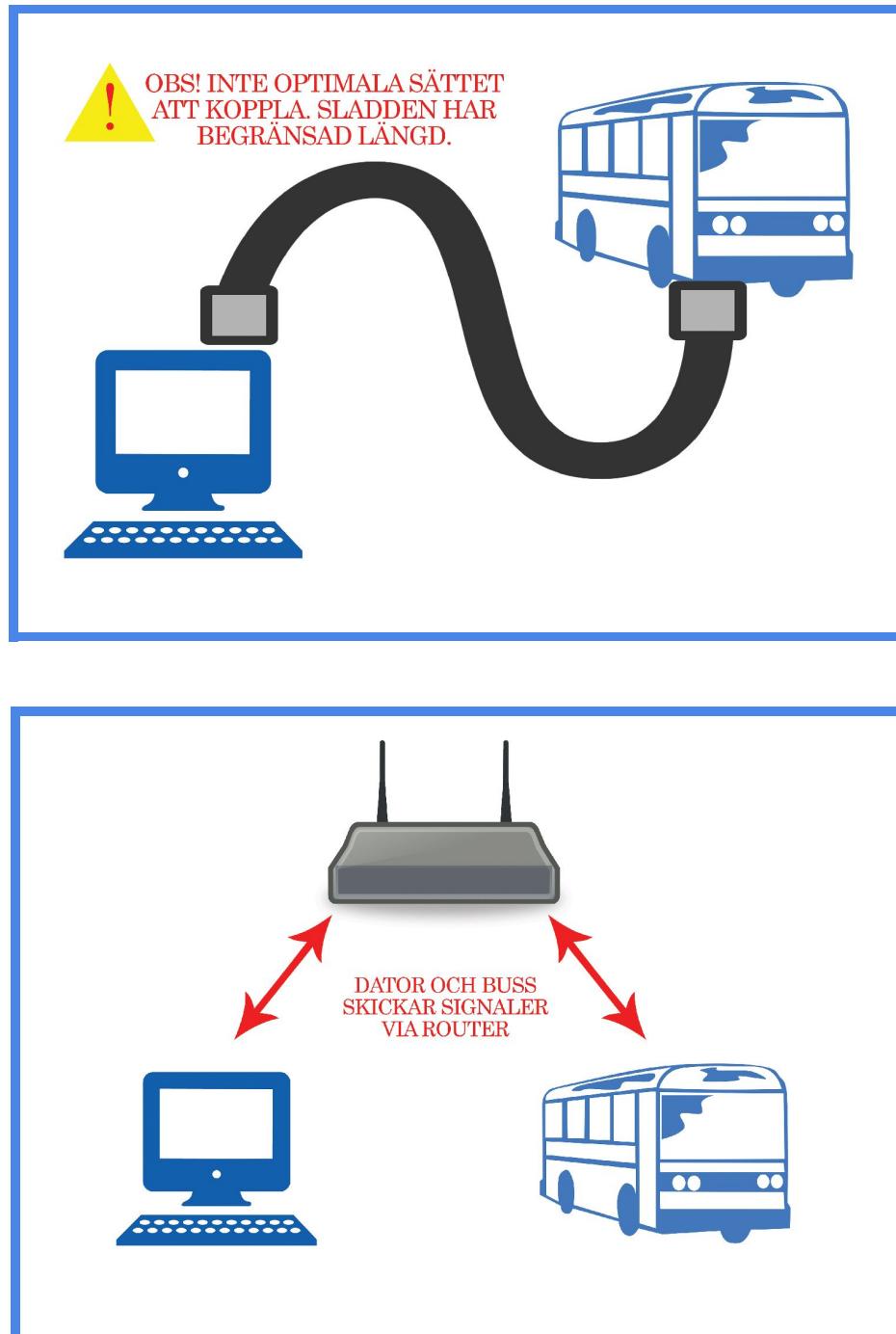
Autonoma bussar kan vara en lösning för bättre och säkrare kollektivtrafik i framtiden. Om något fel eller något problem uppstår så borde det finnas en lösning för det med. Men vad kan den lösningen vara för något? Kanske en busschaufför som sitter i någon byggnad långt ifrån och kan ta över styrningen genom fjärrstyrning och livesändning till chaufförens dator?

Det är exakt det som vårt projekt handlar om. Ni ska ta fram ett sätt att styra en autonom buss genom en dator och livesändning av vägen. Självklart är det här i en mindre skala än ute i den riktiga världen. Vi kommer använda oss av en vanlig dator (och lite programmering på den), en router (som kommer funka som länken mellan bilen och dator) samt en leksaksbil.

På nästa sida ser vi vad kopplingen mellan dessa är!

Hur går styrningen till?

Svaret är enkelt! Istället för att koppla bussen via en sladd till datorn så kan vi använda oss utav en router. Routern har ett nätverk som man kan koppla både bussen och datorn till. Det häftiga här är att routern i sig behöver inget internet! När datorn och bussen är kopplade till routerns nätverk så kan routern ses som en supersnabb brevbärare som skickar vidare signalerna av dem.



Komponenter

Innan vi slänger oss på programmet så ska vi ta en titt på vilka delar vi behöver för att bygga upp bussen.

Fjärrstyrd leksaksbil med två DC motorer;

Bilen har en främre och en bakre motor. Den främre motorn bestämmer om bilen ska svänga höger eller vänster, den bakre motorn bestämmer om bilen åker framåt eller bakåt.

Raspberry Pi 3:

Den används som ett verktyg för att skicka signalerna (höger, vänster, framåt, bakåt) till motorerna på bilen.

Raspberry Pi Kamera Modul:

Den används för att skicka bilder/video från bilen till datorn.

L293D Motor Driver IC:

Det här är en motorkontroll som översätter signalerna så bilen förstår dem.

Kopplingsbräda:

Ett verktyg som underlättar kopplingen.

Kopplingstrådar:

Dessa utgör själva kopplingarna.

Powerbank:

Den tillför ström till Raspberry Pi så den slipper vara kopplad till ett eluttag.

Kartong och limpistol:

För att montera allt.

Programvara

Vi måste också förbereda datorn genom att ladda ner några program som underlättar kopplingen för datorn och bussen.

PUTTY

Ett program för att kopplas in mot Raspberry Pi:en över nätverket (routern).

XMING

Ett hjälpprogram till Putty.

Mjukvara - Programmet

Vi antar att en redan förinstallerad Raspberry Pi finns, om så inte är fallet så kan man följa instruktionerna på deras egna hemsida om hur man ska gå till väga:

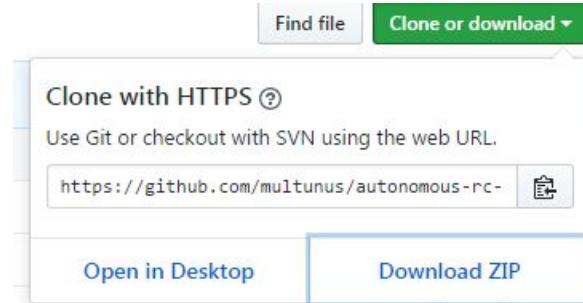
<https://www.raspberrypi.org/documentation/installation/noobs.md>

OBS! För detta behövs en dator samt en extra datorskärm, datormus och tangentbord.

Nu när vår Raspberry Pi har ett fungerande operativsystem så ska vi passa på och förbereda några saker till innan vi kopplar bort den från skärmen och kopplar den till bussen. Genom att följa länken nedan så kan vi ladda ned ett program som låter en fjärrstyra bilen.

<https://github.com/multunus/autonomous-rc-car>

Genom att trycka på den gröna knappen och trycka på Download ZIP så kan vi ladda ned programmet som behövs för att styra bilen.



Lättast är att använda Raspberry Pi:en för att ladda ned programmet och sedan lägga det i en mapp där man vet att det finns.

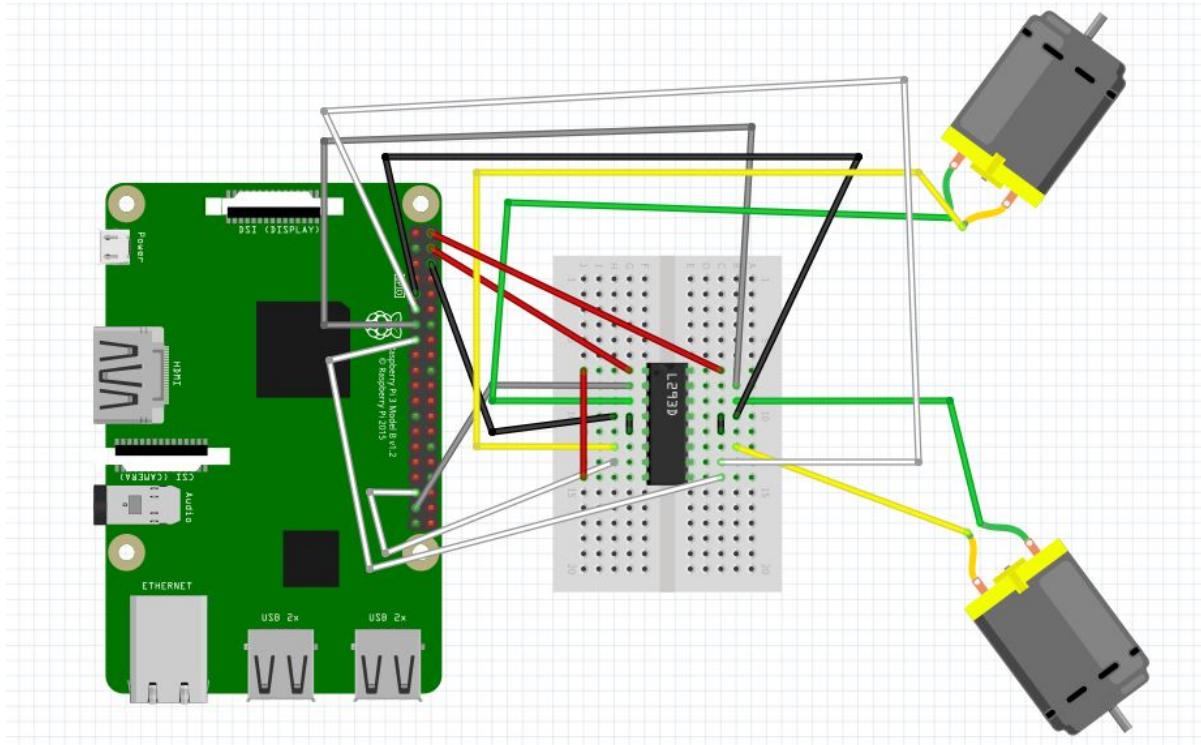
På våra bilar så hittar man mappen **autonomous-rc-car** direkt under mappen **Documents**. I den mappen så hittar man en fil **interactive_control_train.py** som vi sedan ändrat lite i och sedan döpt till **carcontrol.py**.

I vårt projekt kommer man använda sig av **carcontrol.py** för att styra bilen, men om man vill testa att göra projektet hemma så kan man använda sig av **interactive_control_train.py** istället.

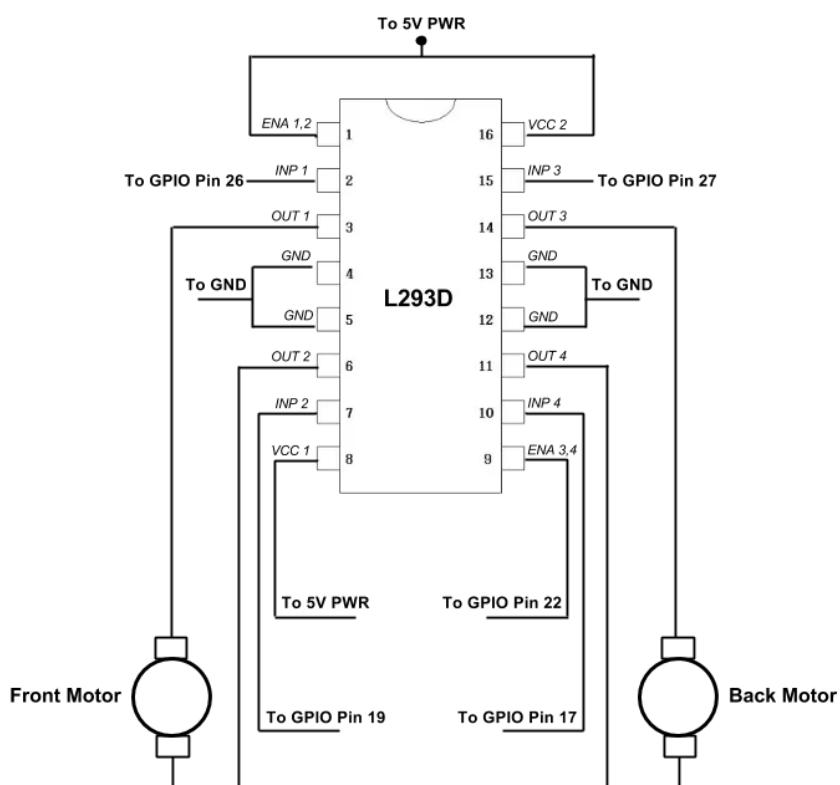
När programmet är nedladdat och man vet var det finns så ska man börja med kopplingen mellan komponenterna innan man konfigurerar anslutningen mellan dator och Raspberry Pi (Bussen).

Hårdvara - Kopplingarna

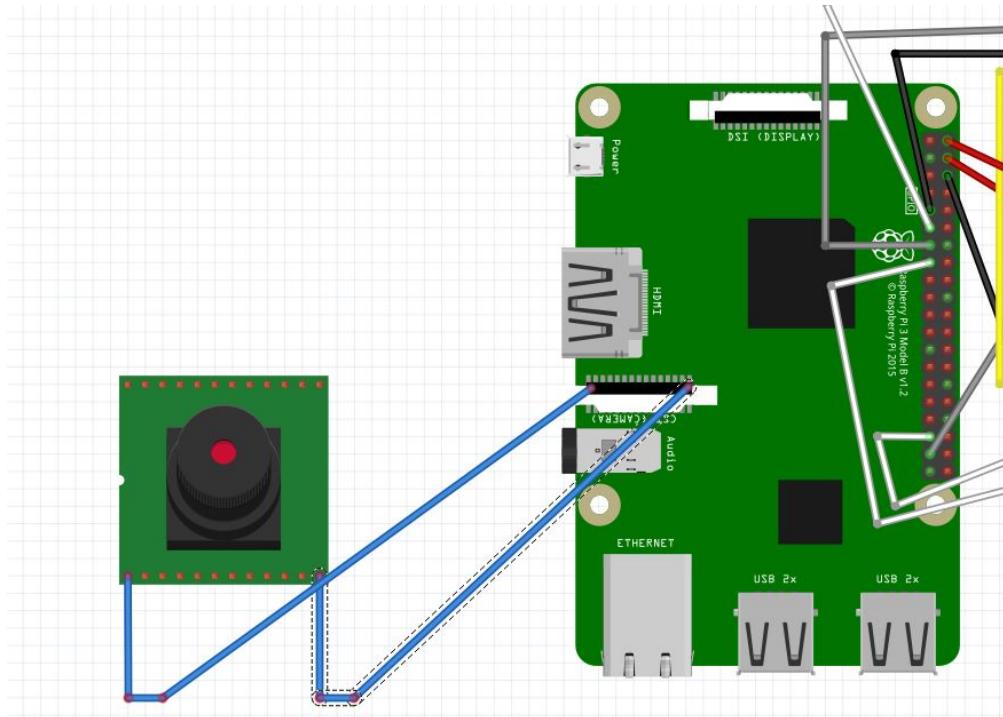
Det första steget vi måste göra är ta **motorkontrolle**n L293D och placera den mitt på **kopplingsbräden**. I bilderna nedan kan ni se hur resten av kopplingarna ser ut. Motorn i övre högra hörnet är den **främre motorn** och motorn i nedre högra hörnet är **bakre motorn**.



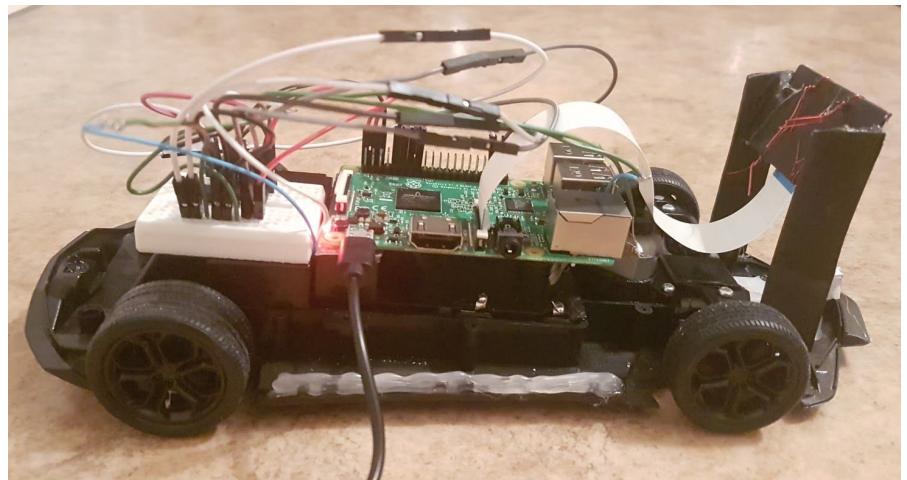
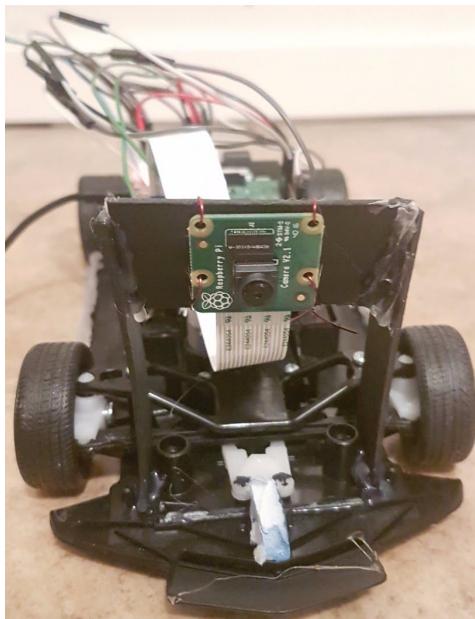
Nedan kan ni se en mer beskrivande bild på hur kopplingarna ser ut runtom motorkontrolle.



Nu när kopplingarna är klara för styrningen så kan vi koppla på **kameran** också. Nedan ser man vart den ska kopplas. **OBS!** Den **blåa remsan i slutet av kopplingsbandet** ska vändas **mot** portarna för USB och internet. Montera sedan kameran fram i bussen och resten av komponenterna i mitten av bussen.



Vid det här laget så borde bilen se ut så här:



Mjukvara - Anslutning och stream

Tidigare nämns programmet PUTTY och XMING, dessa ska installeras på datorn nu.

För att komma in på Raspberry Pi:en så måste man gå igenom följande steg:

1. Öppna programmet **PUTTY**.
2. Skriv **IP-adressen** till Raspberry pi:en i textfältet som heter **Host Name** på första sidan.
IP-adressen till Raspberry Pi:en kan man läsa av med hjälp av **Fing** (en android app) på en mobil som är uppkopplad på samma nätverk som Raspberry Pi:en.
<https://play.google.com/store/apps/details?id=com.overlook.android.fing&hl=sv>
3. Välj sedan **connection type: SSH**
4. Tryck på **SSH** och sedan **X11**
5. Bocka in **Enable X11 forwarding**
6. Klicka på **open**.
7. Skriv in ditt **användarnamn** på Raspberry Pi:en
8. Skriv in ditt **lösenord** på Raspberry Pi:en

Nu har vi kopplat upp oss mot Raspberry Pi:en och därmed kopplat datorn och bussen ihop!

Det är dags att köra kommandon och program på Raspberry Pi (Bussen) från vår andra dator. Detta funkar så länge de är uppkopplade mot samma nätverk.

Vi kan nu kontrollera bilen från datorn men vi kan inte se något från kamera-modulen som finns monterad på den!

För att kunna streama video från Raspberry Pi:en så ska man installera ett program som heter **UV4L**.

Vi använder oss av den här guiden:

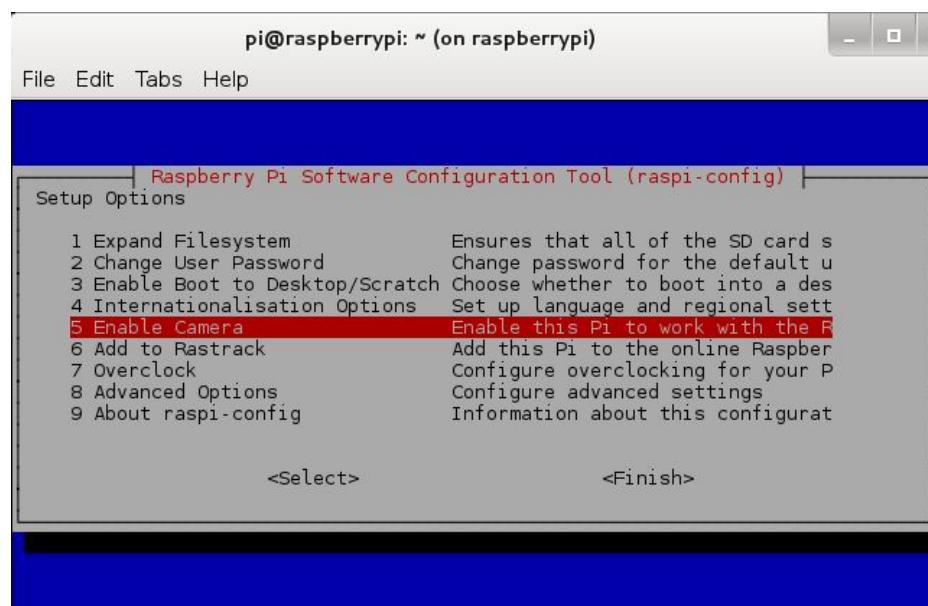
<https://www.linux-project.org/uv4l/installation/>

Kommandon och saker som ska göras sammanfattas nedan.

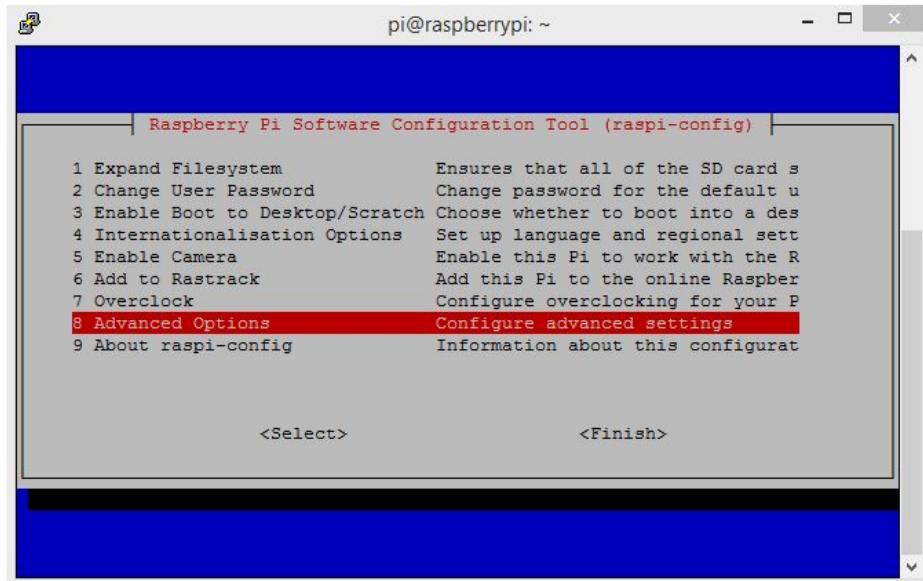
Vi använder terminalen vi fått genom att koppla oss till Raspberry Pi:en genom PuTTy och skriver:

sudo raspi-config

Ett fönster kommer dyka upp och där går man till **Enable Camera** och sätter igång kameran.



Sedan går vi i samma program till **Advanced Options** och sedan till **Memory Split** och ser till att det står 128(eller högre) i den rutan.



Väl där så ser vi till att kameran är aktiverad genom att

`curl http://www.linux-project.org/listing/uv4l_repo/lrkey.asc | sudo apt-key add -`

Sedan öppnar vi “/etc/apt/sources.list” med exempelvis GNU Nano genom att skriva:

`sudo nano /etc/apt/sources.list`

I botten av den filen lägger vi till:

`deb http://www.linux-project.org/listing/uv4l_repo/raspbian/ jessie main`

Efter det så kör vi dessa kommandon, ett och ett, i vår terminal.

`sudo apt-get update`

`sudo apt-get install uv4l uv4l-raspicam`

`sudo apt-get install uv4l-raspicam-extras`

`sudo service uv4l_raspicam restart`

`sudo apt-get install uv4l-server uv4l-uvc uv4l-xscreen uv4l-mjpegstream uv4l-dummy
uv4l-raspidisp`

`sudo apt-get install uv4l-webrtc`

```
sudo apt-get install uv4l-xmpp-bridge
```

```
sudo apt-get install uv4l-raspidisp-extras
```

När allt är installerat så kan vi säga åt Raspberry Pi:en (Bussen) att streama genom att köra kommandot:

```
sudo uv4l -nopreview --auto-video_nr --driver raspicam --encoding mjpeg --width 320 --height 256 --framerate 8 --server-option '--port=9090' --server-option '--max-queued-connections=1' --server-option '--max-streams=1' --server-option '--max-threads=4' --server-option '--editable-config-file=yes' --server-option 'config-password=111'
```

(**OBS:** Vid experiment hemma. Det är viktigt att se till att det inte finns några radbrytningar i det här stycket när man kör det ifrån terminalen, för annars kanske det inte fungerar. Kolla igenom allt och ta bort ”\” om det råkar finnas några.)

I vårt fall har vi sparat ner det hela i en fil som heter **stream-script.sh**, så vi kan starta streamen genom att gå in i mappen där programmet finns. Det går lika bra att skriva allt varje gång men det blir lite jobbigt.

När man kör detta så kan man komma åt streamen genom att använda en webbläsare på en dator/mobil (se till att dessa är kopplade till samma nätverk som Raspberry Pi:en (Bussen)).

I adressfältet skriver vi in ip-adressen till Raspberry Pi:en följt av :XXXX där X:en byts ut mot vad som står i kommandot ovan efter “**--port=**”.

Så om Raspberry Pi:en har adressen 192.168.0.5 och man använder kommandot ovan så skriver man i addressfältet: **192.168.0.5:9090**

Man kommer då till en sida med många rutor och val. Sedan klickar man på rutan där det står **MJPEG/Stills stream** för att se streamen.

UV4L Streaming Server



Så slutligen för att starta streamen och påbörja styrningen av bussen så tar vi vår terminal (och utgår ifrån att den är i hem-mappen) och skriver:

```
cd Documents
```

(Här går man in i Documents från hem-mappen.)

```
cd autonomous-rc-car
```

(Här går man in i autonomous-rc-car-mappen från Documents.)

```
bash stream-script.sh
```

(Här startas streamen.)

`python3 carcontrol.py`

(Här startas möjligheten att kontrollera bussen.)

Efter det så kan vi se vad bussen ser genom vår webbläsare och styra den med pil tangenterna!

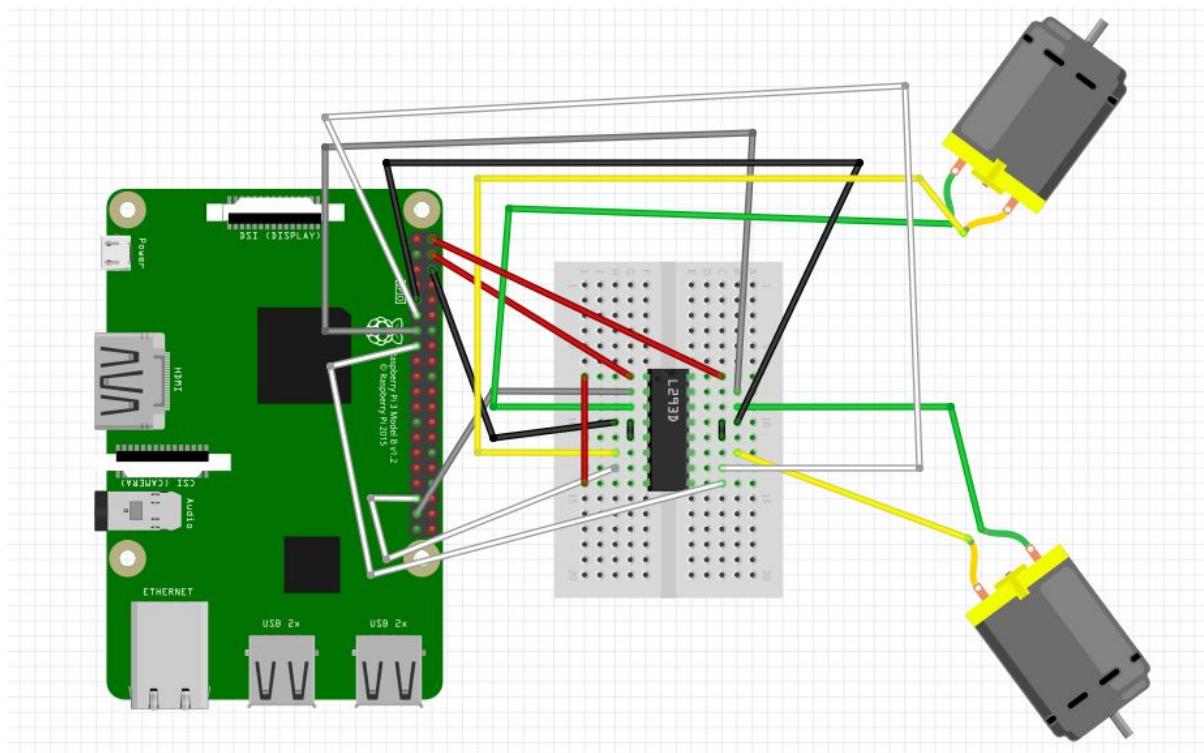
Uppgifter

1. Självkörande bilar har väldigt många fördelar: de är säkra i trafiken, de kör optimalt, alla kan åka i en självkörande bil oavsett om man har körkort eller inte, de kör mer miljövänligt än en vanlig förare osv.

Diskutera i grupp vilka nackdelar som kan finnas med en självkörande bil?

Vilka fel kan det uppstå?

2. Koppla upp Raspberry Pi:en till bussen enligt bilden nedan:



3. Efter ni har kopplat klart allt, räck upp en hand så ska vi hjälpa er starta livesändningen och styrningen av bussen. Ni ska sedan köra bussen efter den förbestämda banan. Se till att stanna vid busshållplatserna!