## Part 1: Soldering and Wiring

Get all the components needed for the project as described in the file **materials&tools_list.pdf** and follow the steps:

1. Solder wires to the DC motors

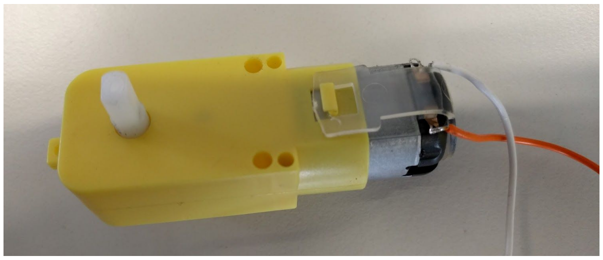


*Figure 1.1: Wired DC motor*

2. Solder wires to the 2 fans:

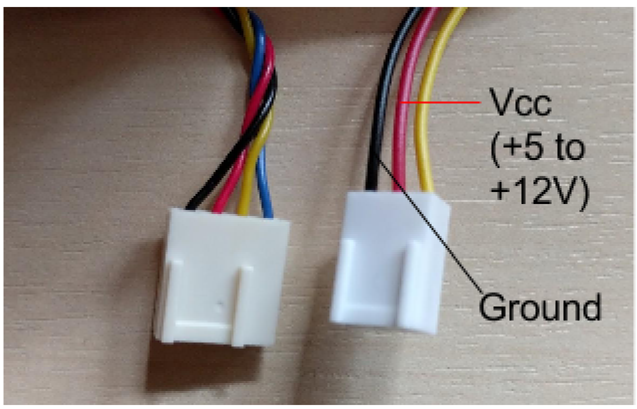



*Figure 1.2: Wiring of fan*

3. Solder resistors to the Ultrasonic sensor:

Connect the SRF02 ultrasonic sensor to the Raspberry according to the table below:

| Ultrasonic sensor pin | Raspberry Pi GPIO pin |
|---|---|
| 5V | 5V |
| GND | GND |
| SDA | SDA |
| SCL | SCL |

*Table 1: Wiring ultrasonic sensor and Raspberry Pi*

4. Solder red power wire to Raspberry Pi (**Optional**)

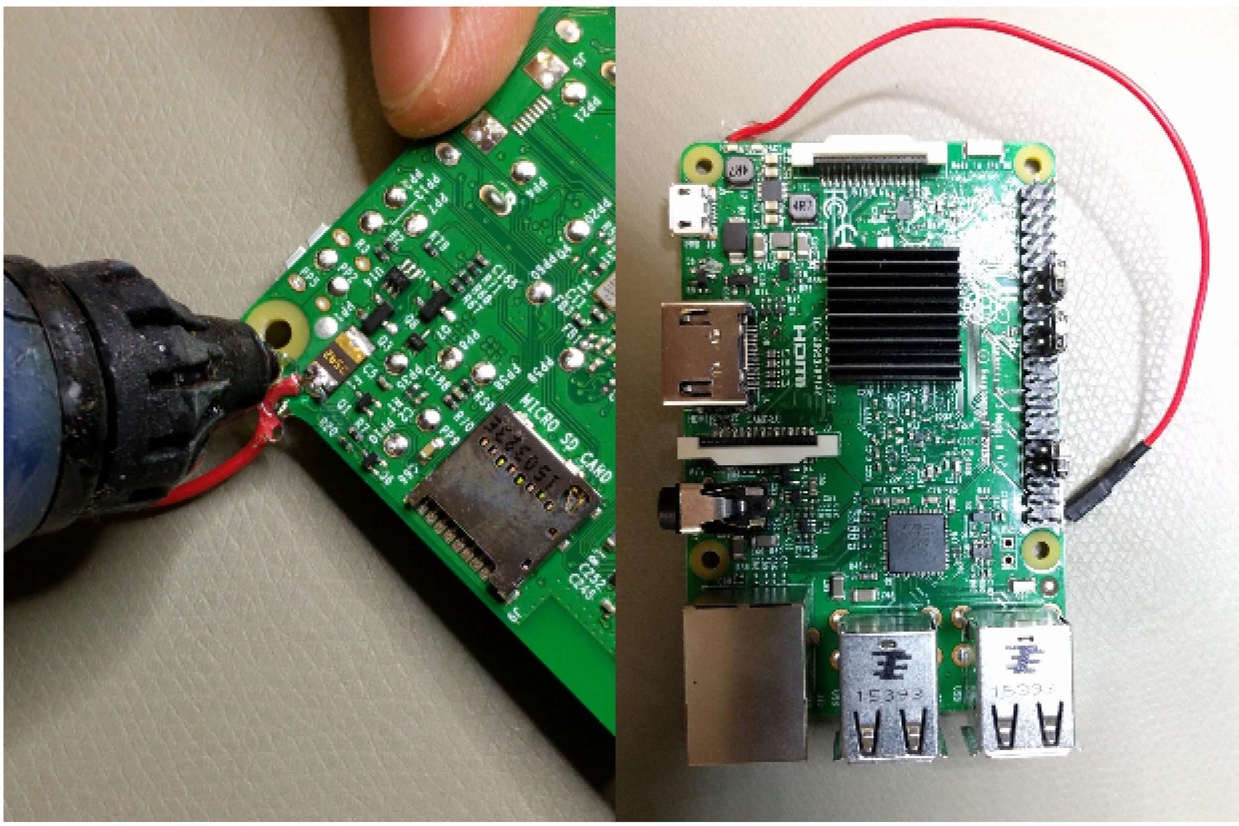
Solder red power wire with female pin header to +5V fuse of the Pi.

Use hot glue to fix the wire to the board if necessary.

This modification is needed to enable us to supply the Pi with +5V without the Micro USB port and still make use of all the protection circuitry.
If we would just use one of the +5V GPIO Pins, to supply the Pi with power, we would bypass all the build in protections.

If you don’t want to modify your Pi, you can just use one of the +5V GPIO pins, but be aware of the dangers.

Use hot glue to fix the wire to the board if necessary.



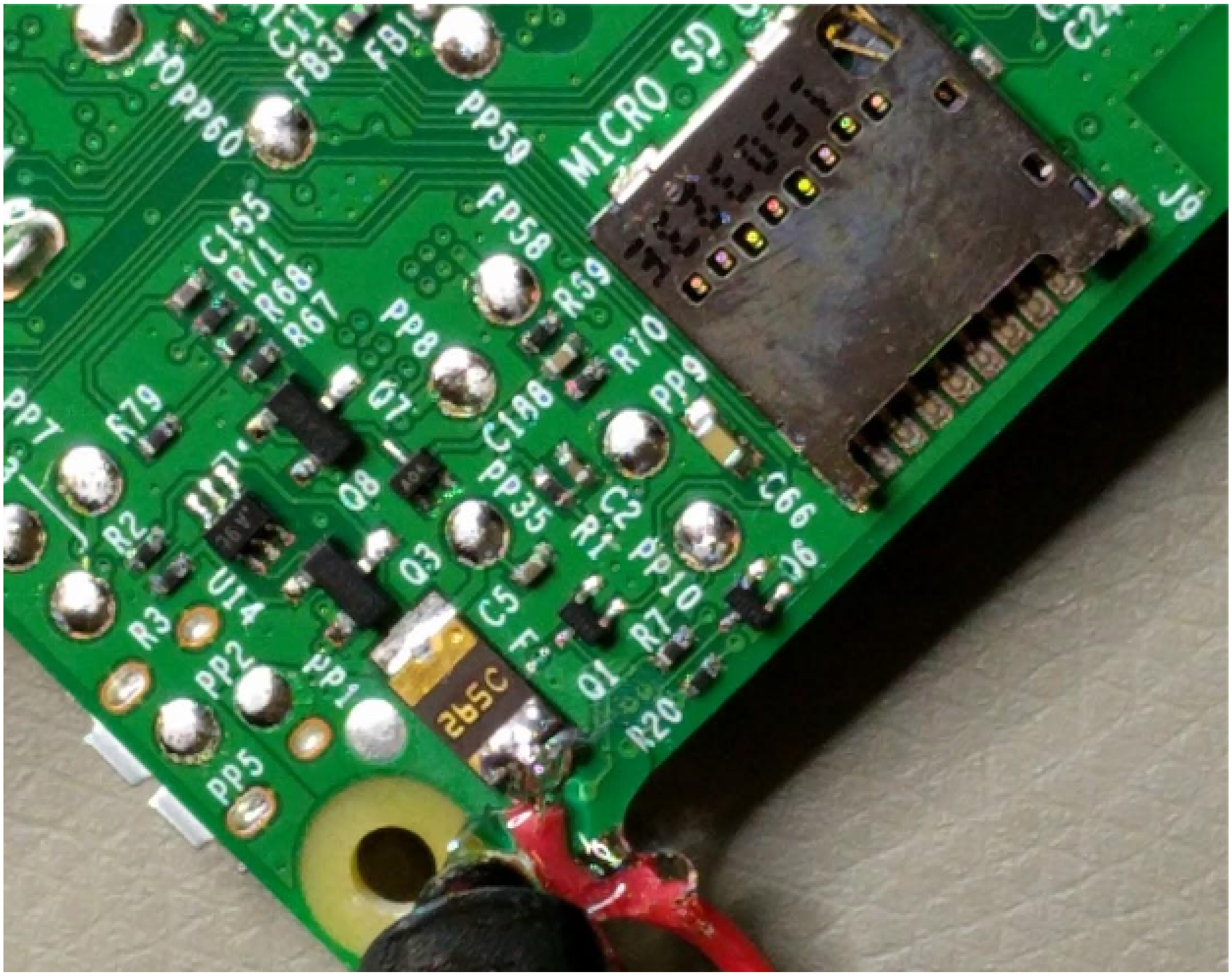


*Figure 1.3: Front and back view of power wire soldered to the fuse of the Raspberry Pi*

5. Solder wires to the step down converter

We solder red and black wires to the input of the converter, and 2 male pin headers to the end of the 2 wires.

We do the same for the output side except now we use a female pin header on the black ground wire so that we can directly plug it into ground pin on the Pi GPIO header.

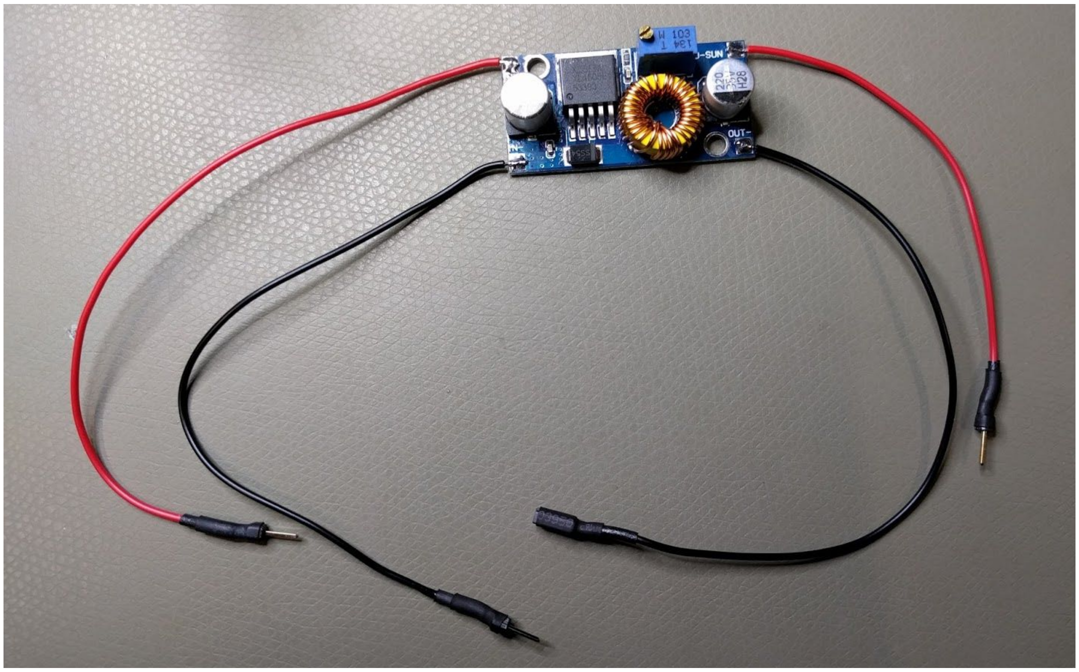


*Figure 1.4: Wired step down converter*

6. Solder wires to the loudspeaker - Optional

This part is to ensure that the loudspeaker can be plugged into and powered by the Raspberry Pi instead of its own battery. Thus this is an optional step for those who prefer to leave the loudspeaker as it is and not to open up the loudspeaker.

Since we have an old portable loudspeaker, we decided to make use it instead of buying new loudspeaker from the store (link to buy can be found the the Material list excel sheet).

It should be easy to trace the power line and ground plate on the loudspeaker's PCB based on its connection to the battery. Once we have found them, we need to solder to the loudspeaker is one red power wire and one black ground wire, both with female pin headers. However, do not do anything to the signal wire.

Final result should look like this:

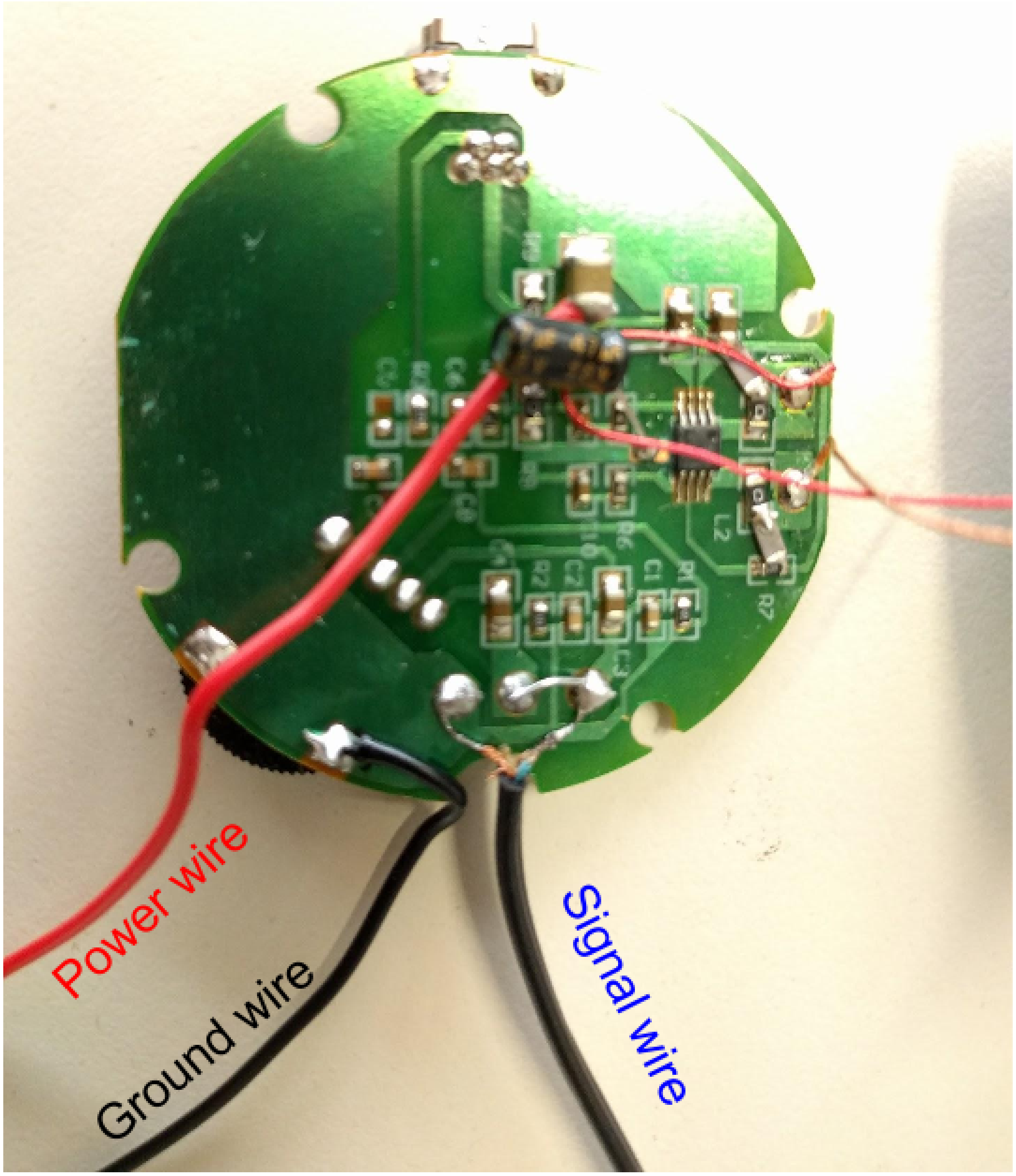


*Figure 1.5: Loudspeaker soldered with power and ground wires*

That’s all for soldering and wiring!

## Part 2: Building and Connecting wires

1. Get the laser cut parts needed for building the car

Open the design file **car_frame_design.pdf** in Adobe Illustrator and cut it in the laser cutter:

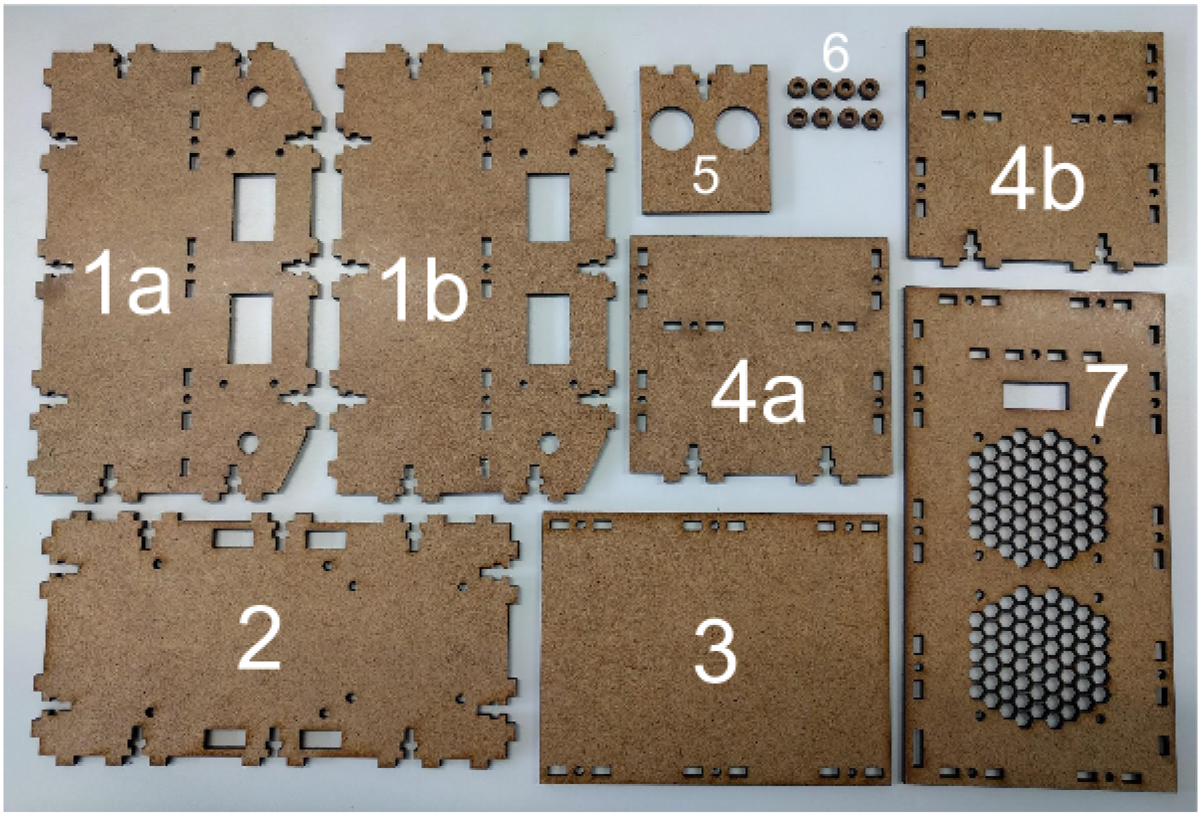



*Figure 2.1: All the laser cut parts required for building the car*

Table 2.1: Explanation of the parts

| **Part number in Figure 1** | **Part name** | **Quantity** |
|---|---|---|
| 1a, 1b | Long side plate | 2 |
| 2 | Middle plate | 1 |
| 3 | Bottom plate | 1 |
| 4a, 4b | Short side plate | 2 |
| 5 | Ultrasonic sensor holder | 1 |
| 6 | Spacer | 8 |
| 7 | Top plate | 1 |

2. Get all the wired components needed:

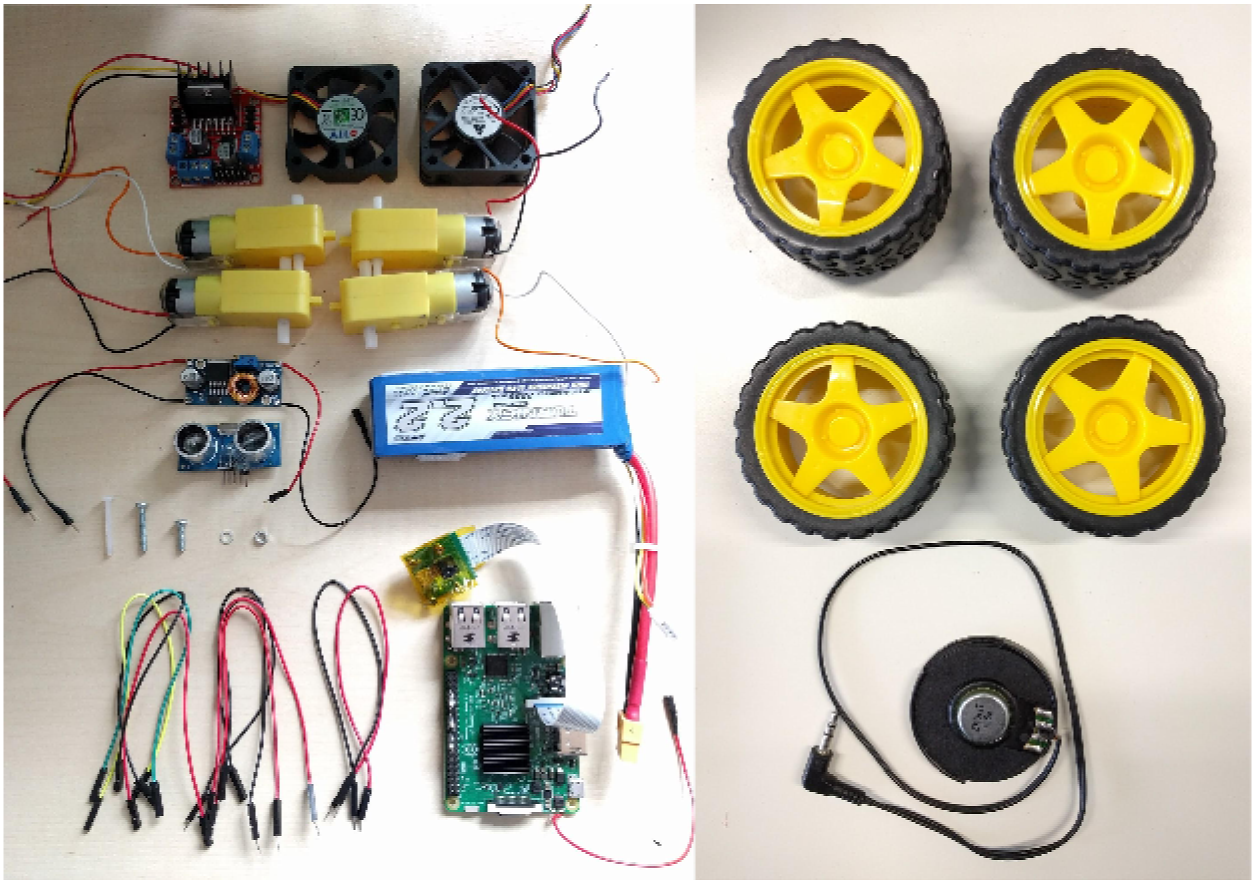


*Figure 2.2: All the wired components and parts needed for building the car*

3. Start building!

Follow these steps to build the car:

- Using 4 M3x25 nylon screws and 4 M3 nuts, fix 4 motors to the long side plates 1a and 1b and cut off the extra length of the screw as in the photo below:

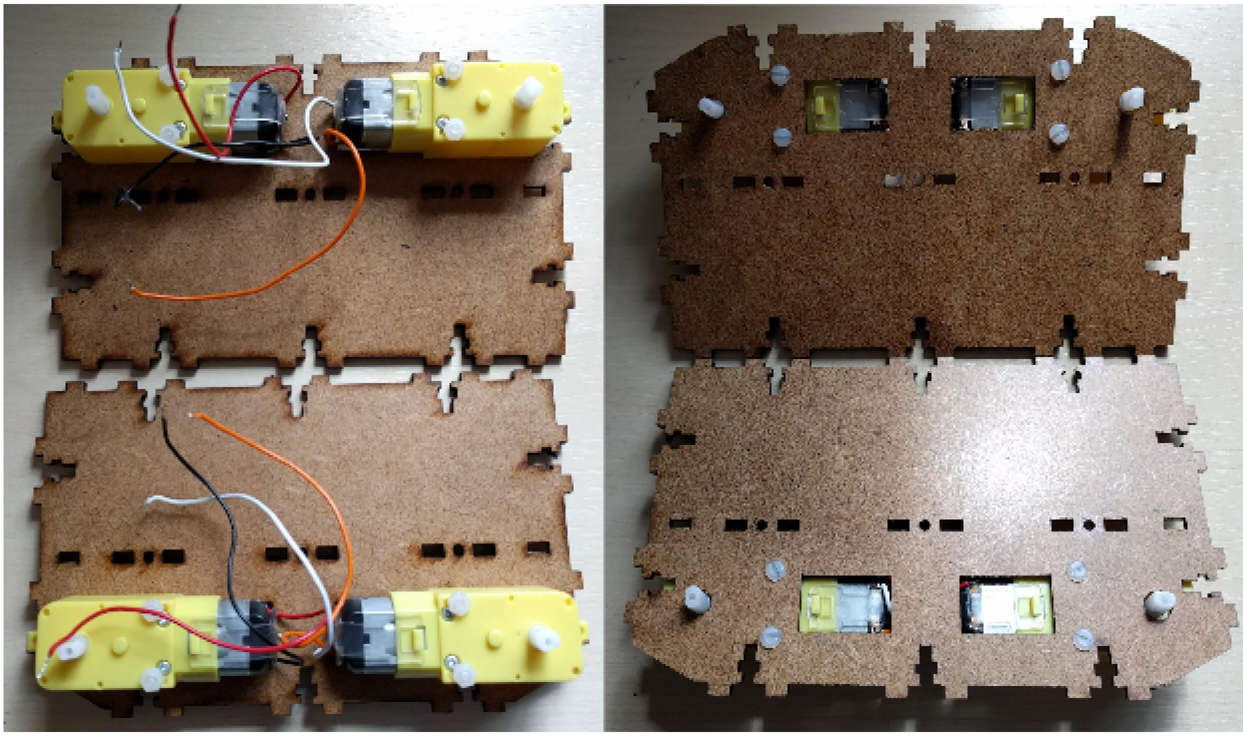


*Figure 2.3: Front and back view of 4 motors screwed down to the long side plates 1a and 1b*

- Using 4 M3x12 screws and 4 M3 nuts, fix the motor driver to the middle plate using 4 spacers in between the plate and the bottom of the motor driver. Make sure the H-bridge terminal of the driver is at the edge of the plate (to ensure easier wiring connections later), and the heatsink of the driver is around the middle of the plate. See the photo below:

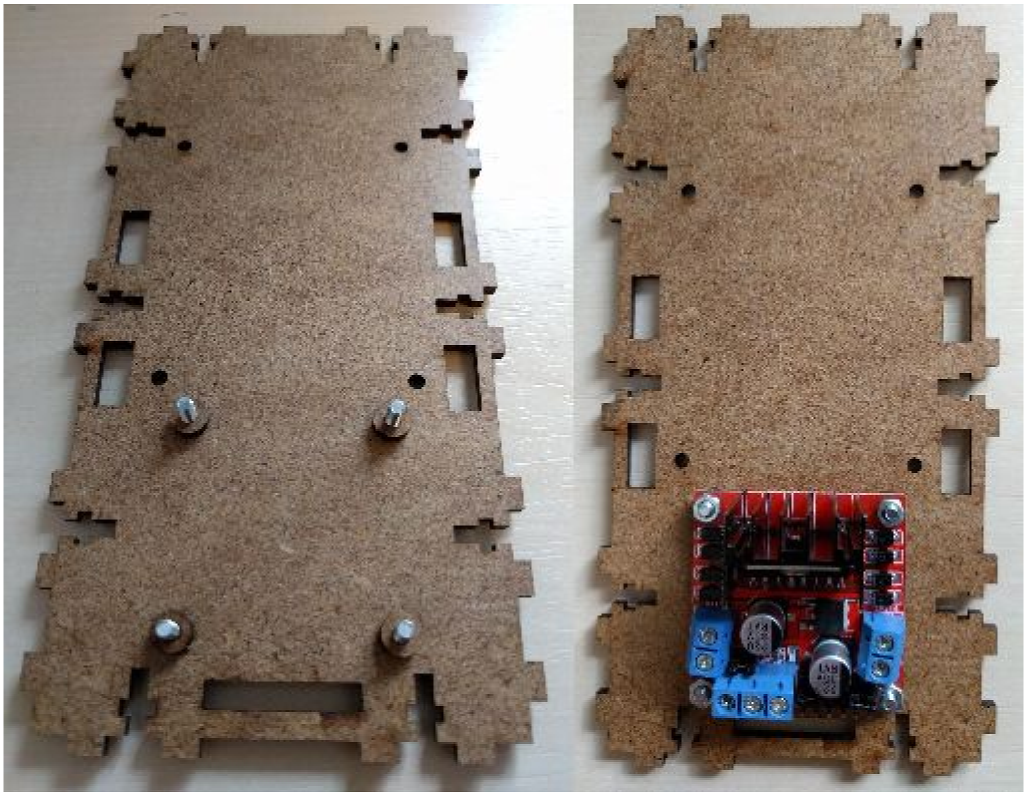


*Figure 2.4: Screwing the motor driver to middle plate (right) with spacers in between (left)*

- Thread the mounting holes on the Raspberry Pi such that they fit M3 screws:

*Figure 2.5: Threading the mounting holes on Raspberry Pi using M3 threading tool*

- Using 4 M3x12 screws and 4 M3 nuts, fix the Raspberry Pi to the same side of the middle plate, also use spacers in between as in the photo below:

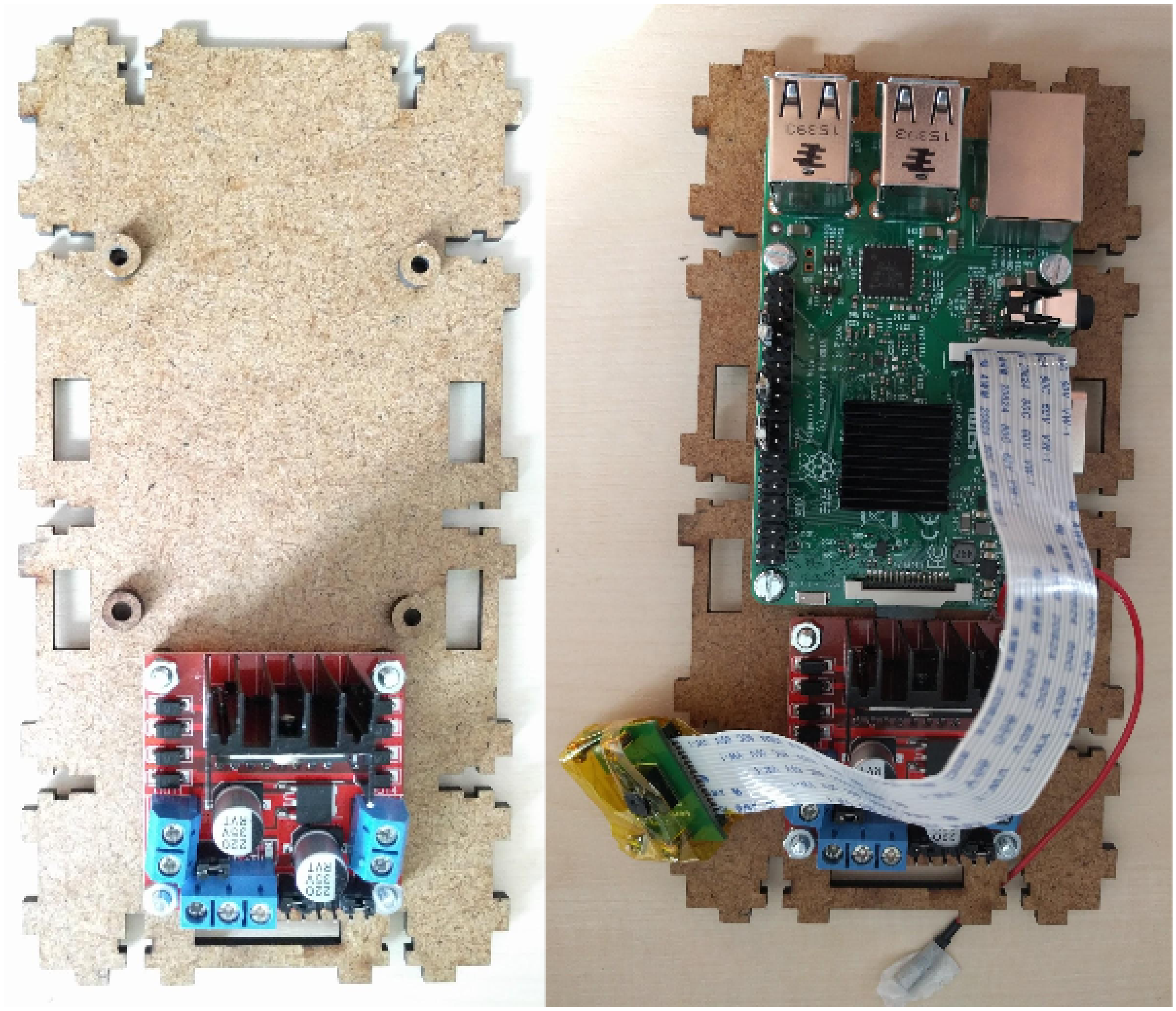


*Figure 2.6: Screwing the Raspberry Pi to middle plate (right) with spacers in between (left)*

- Using 8 M3x25 nylon screws and 8 M3 nuts, fix the 2 fans to the top plate. Make sure to use washers in between the plate and the bottom of the fans, and cut off the extra length of the screw:

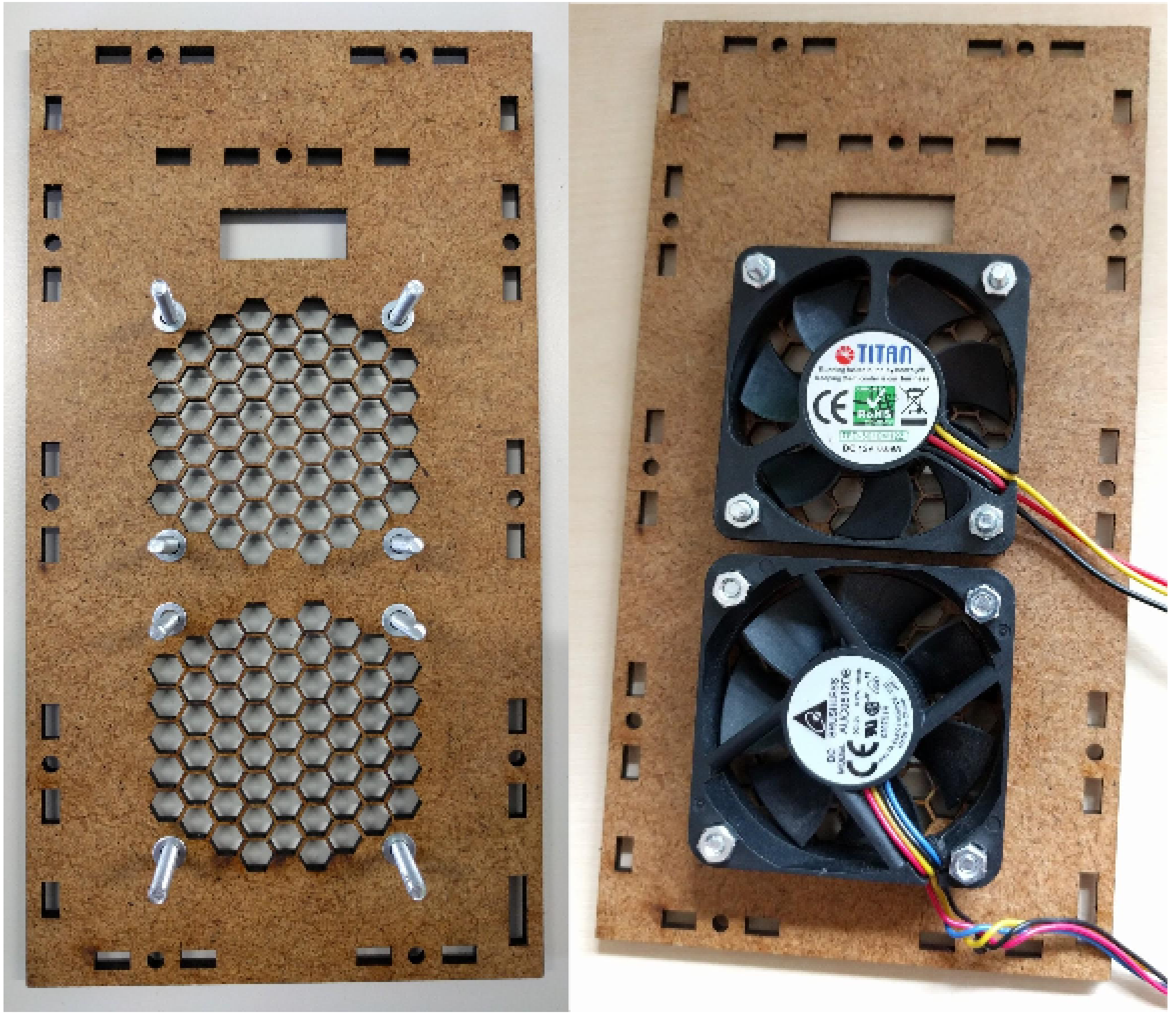



*Figure 2.7: Screwing the fans to the top plate (right) with washers in between (left)*

- Fix the ultrasonic sensor through the holes of the ultrasonic sensor holder:

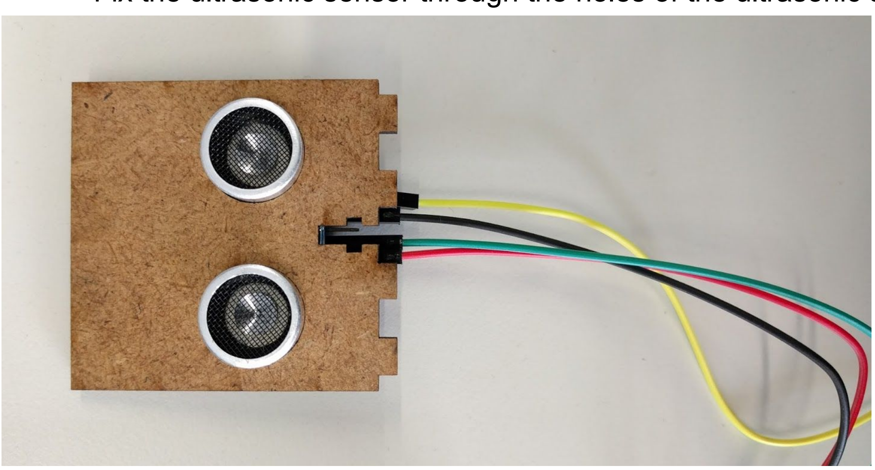


*Figure 2.8: Ultrasonic sensor in the ultrasonic sensor holder*

- Using a M3x12 screw and M3 nut, fix the ultrasonic sensor holder to the top plate. Make sure that the ultrasonic sensor holder and the fans are fixed to the opposite sides of the top plate, and the wires from the sensor go through the rectangular slot on the top plate:

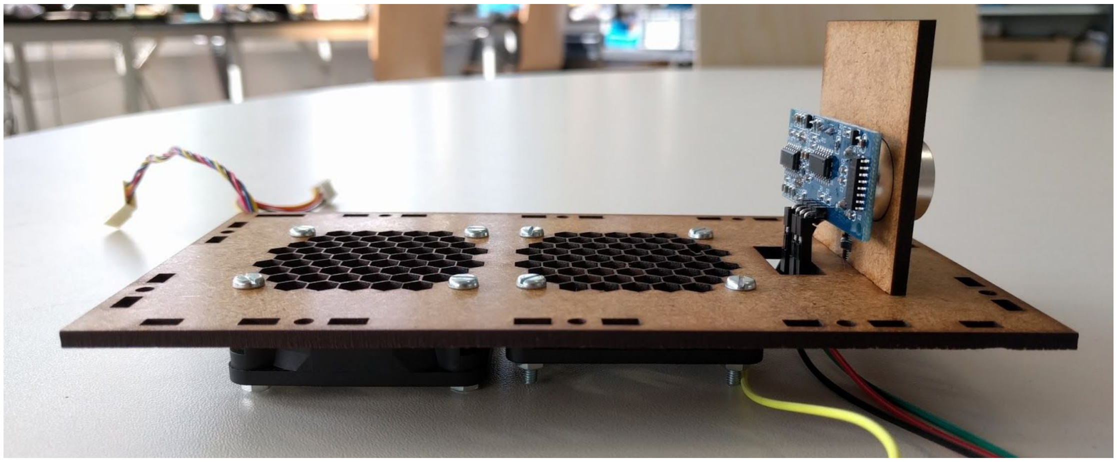


*Figure 2.9: Ultrasonic sensor holder screwed to the top plate.*

- Using 8 M3x12 screws and 8 M3 nuts, screw the middle plate (already fixed with motor driver and Raspberry Pi) with the 2 long side plates and any 1 one of the short side plate. This is to ensure structure for the next section, which is connecting the wires of the components together.

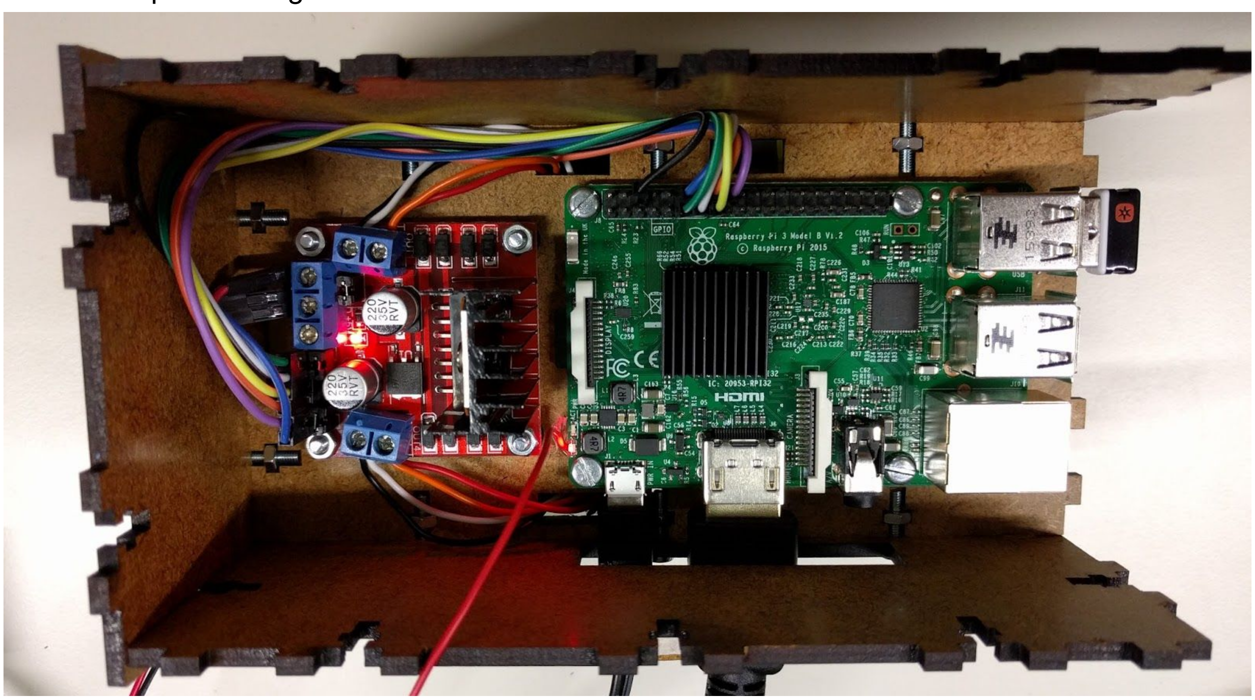


*Figure 2.10: Fix the long side plates and one short side plate to the middle plate*

- Although this is not the end of the building part, we proceed to connecting the wires first, which is easier to do at this stage than when the structure is completely built.

4. Connecting wires

- Connect the motors to the motor driver as in the figure below:

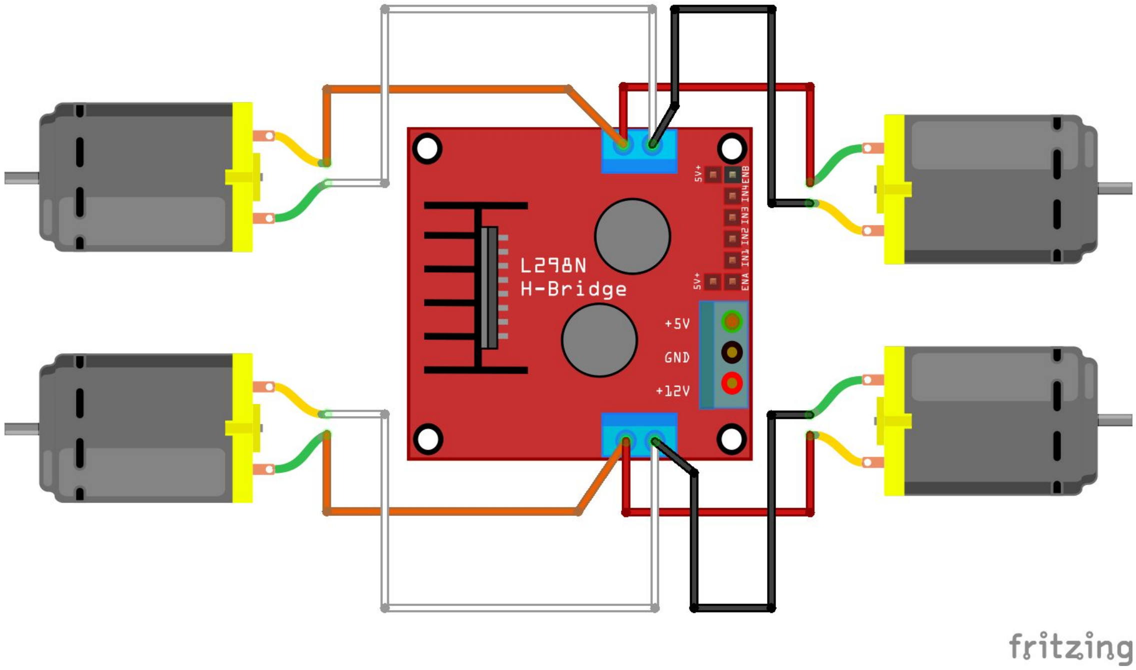



*Figure 2.11: Connecting wires of motors and motor driver*

- Connect the step down converter to the battery and to the Raspberry Pi. Make sure the +5V output from the converter go to the red power wire of the Pi (refer to Figure 1.5) and the ground pin of converter is connected to ground pin of Pi, as in the photo below (ignore the values written on the battery):

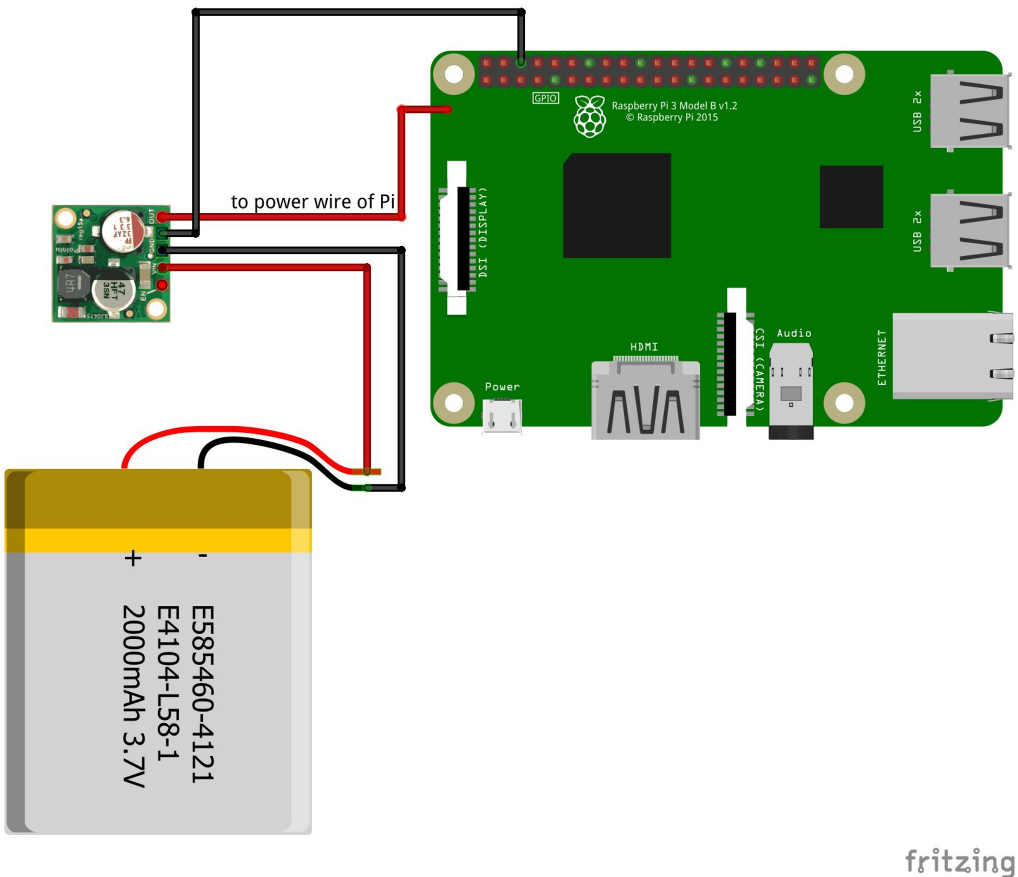



*Figure 2.12: Connecting wires between battery, step down converter and Raspberry Pi*

- Connect the power terminals of motor driver to the battery and to the Raspberry Pi. Make sure the positive terminal of battery is connected to +12V input of motor driver as in the photo below (motors are not shown for simplicity purpose):

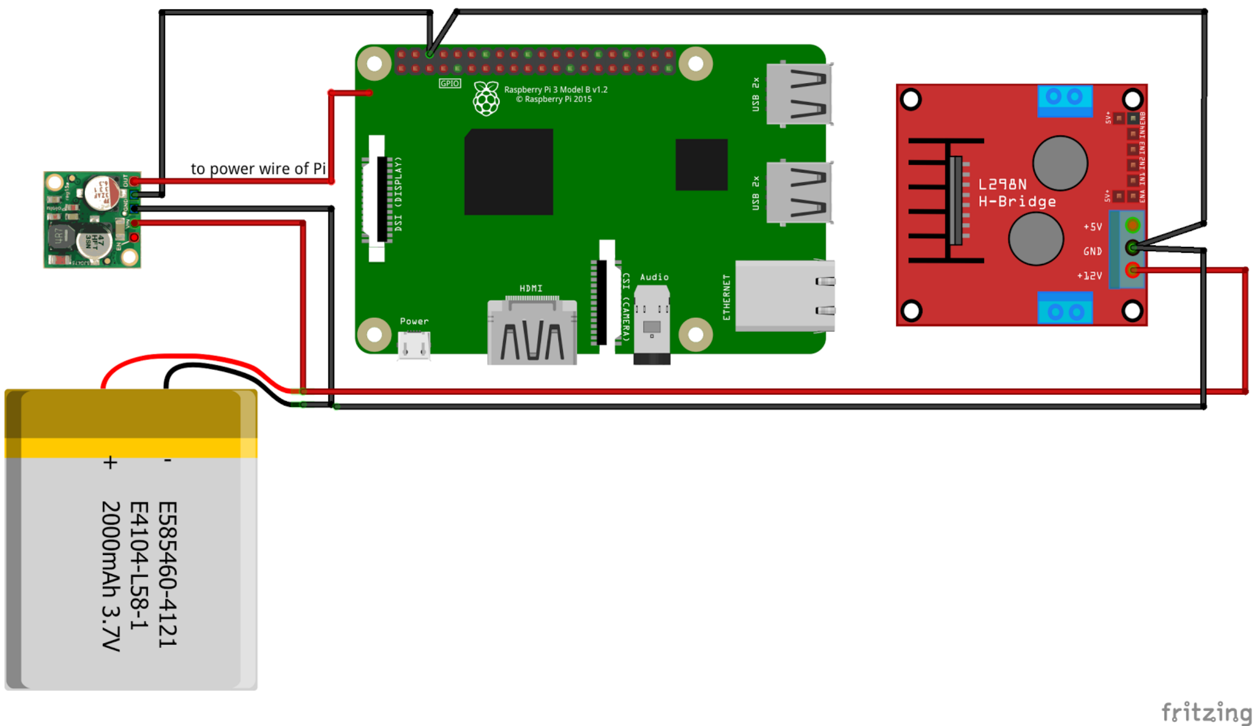



*Figure 2.13: Connecting wires between battery, motor driver and Raspberry Pi*

- Connect the input pins of motor driver to the Raspberry Pi as in the photo of table below (connection between the Pi and other components are not shown for simplicity purpose):

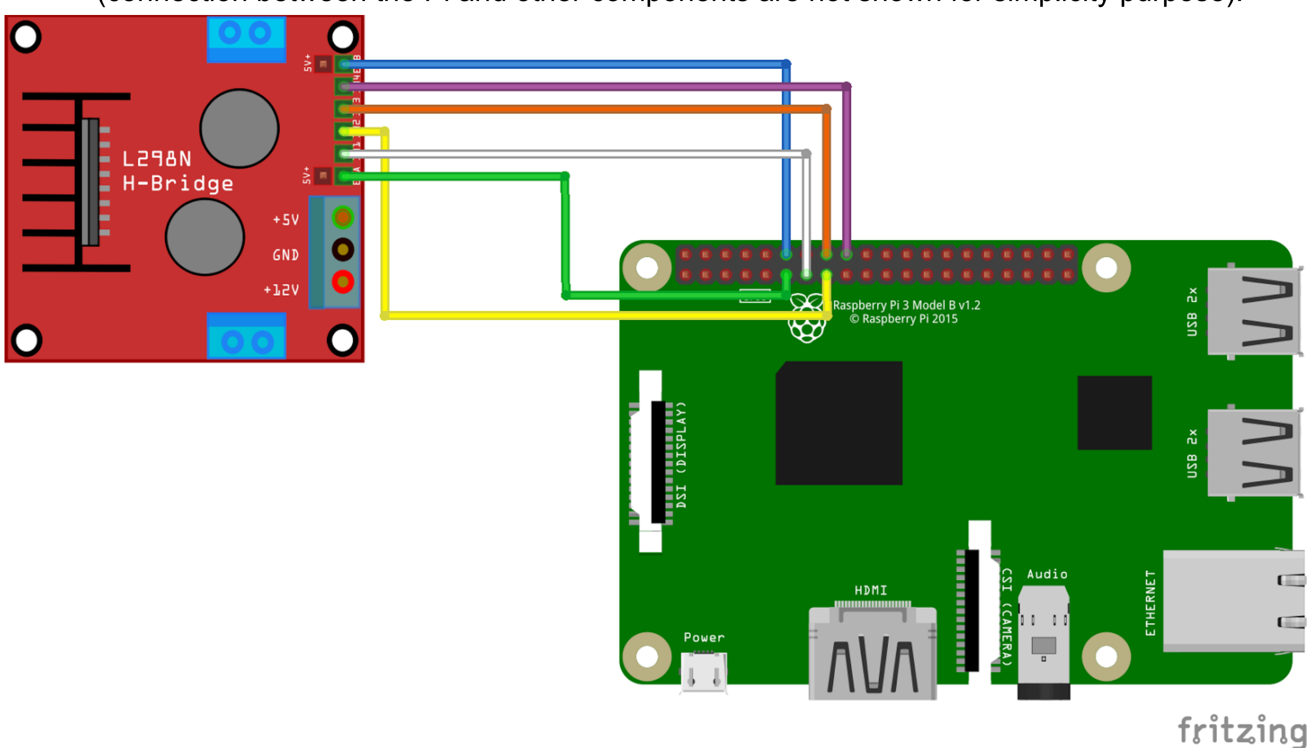



*Figure 2.14: Connection layout between motor driver and Raspberry Pi*

Table 2.2: *Connection instructions between motor driver and Raspberry Pi*

| Motor driver pin | Raspberry Pi GPIO pin | Color coding |
|---|---|---|
| ENA | GPIO 17 | Green |
| IN1 | GPIO 27 | White |
| IN2 | GPIO 22 | Yellow |
| IN3 | GPIO 23 | Orange |
| IN4 | GPIO 24 | Violet |
| ENB | GPIO 18 | Blue |

A diagram of the GPIO pins on Raspberry Pi can be found at https://www.element14.com/community/servlet/JiveServlet/previewBody/73950-102-10-339300/pi3_gpio.png

- Connect the ultrasonic sensor to the Raspberry Pi according to the schematic or table below (connection between the Pi and other components are not shown for simplicity purpose):

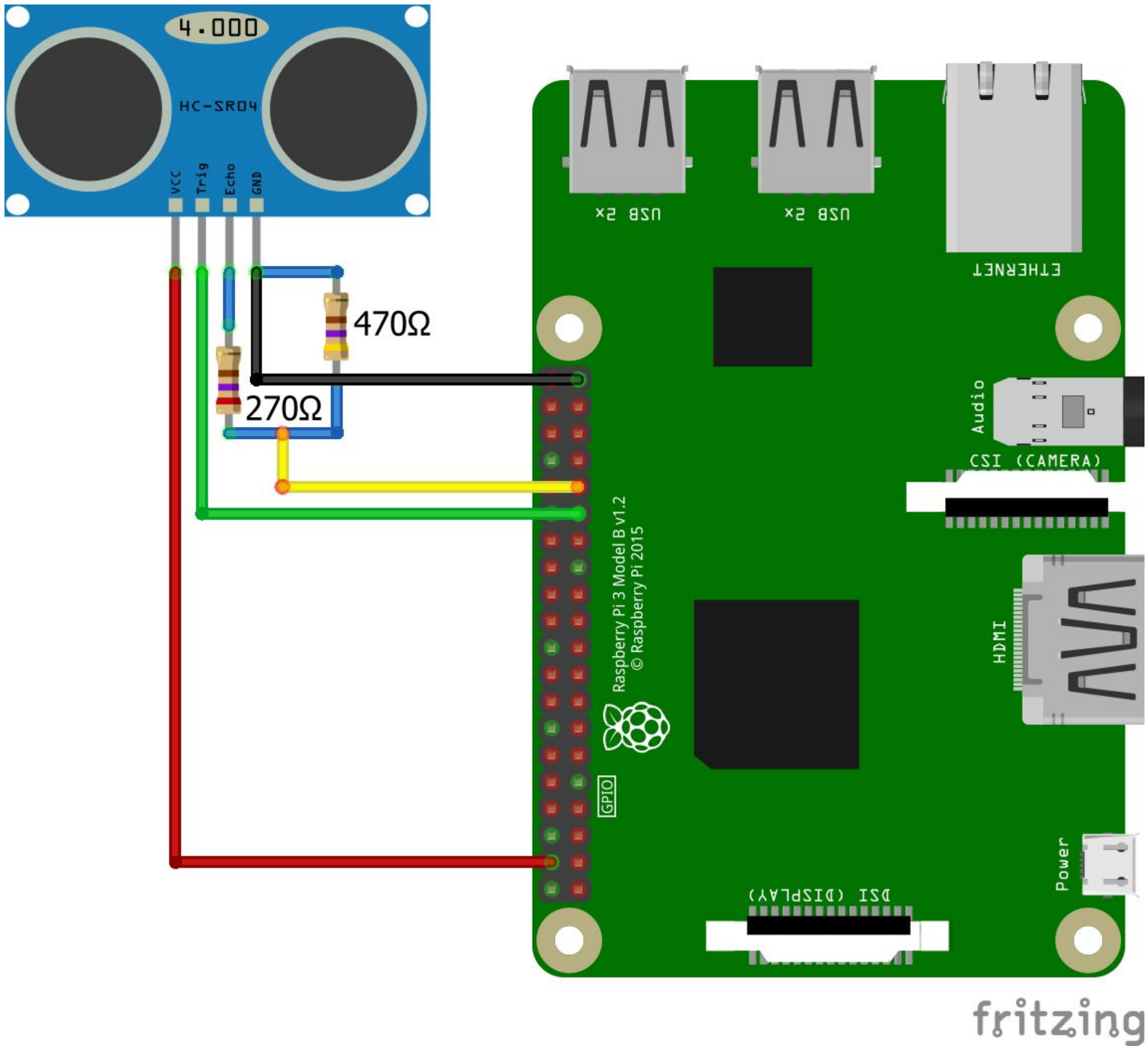



*Figure 2.15: Connection layout between ultrasonic sensor and Raspberry Pi*

Table 2.4: *Connection instructions between motor driver and Raspberry Pi*

| Ultrasonic sensor pin | Raspberry Pi GPIO pin | Color coding |
|---|---|---|
| Trigger | GPIO 05 | Green |
| 3.3V output | GPIO 06 | Yellow |
| Vcc | DC power 5V | Red |
| Ground | Ground | Black |

- Connect the loudspeaker to the Raspberry Pi according to the table below:

Table 2.4: *Connection instructions between loudspeaker and Raspberry Pi*

| Loudspeaker pin | Raspberry Pi GPIO pin | Color coding |
|---|---|---|
| Power | DC power 3.3V | Red |
| Ground | Ground | Black |
| Audio jacket | Audio input | None |

Attach the loudspeaker onto any of the side plates of the car where convenient.

- Connect the fans to the Raspberry Pi

Simply plug the red power wires and black ground wires of the fans to any DC power 5V pin and ground pin on the Pi.
Alternatively, in order to save the pins needed on the Pi, it is suggested to solder a Y-shape cable with 4 female pin headers which go to the 2 fans and 2 male pin headers which go the Pi as following:

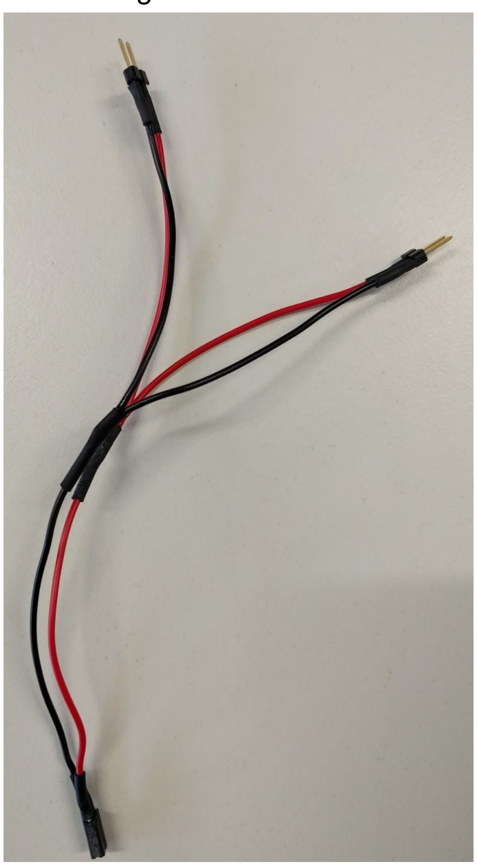


*Figure 2.16: Y-shape cable to connect 2 fans simultaneously to the Pi*

5. Building - continued

Now that we have finished connecting the wire, we can finish up the building of the car.

- Using 8 M3x12 screws and 8 M3 nuts, fix the remaining short side plate to the car. And then use 10 M3x12 screws and 10 M3 nuts to fix the top plate (already attached with fans and ultrasonic sensor) to the car. We will be able to have the following structure:

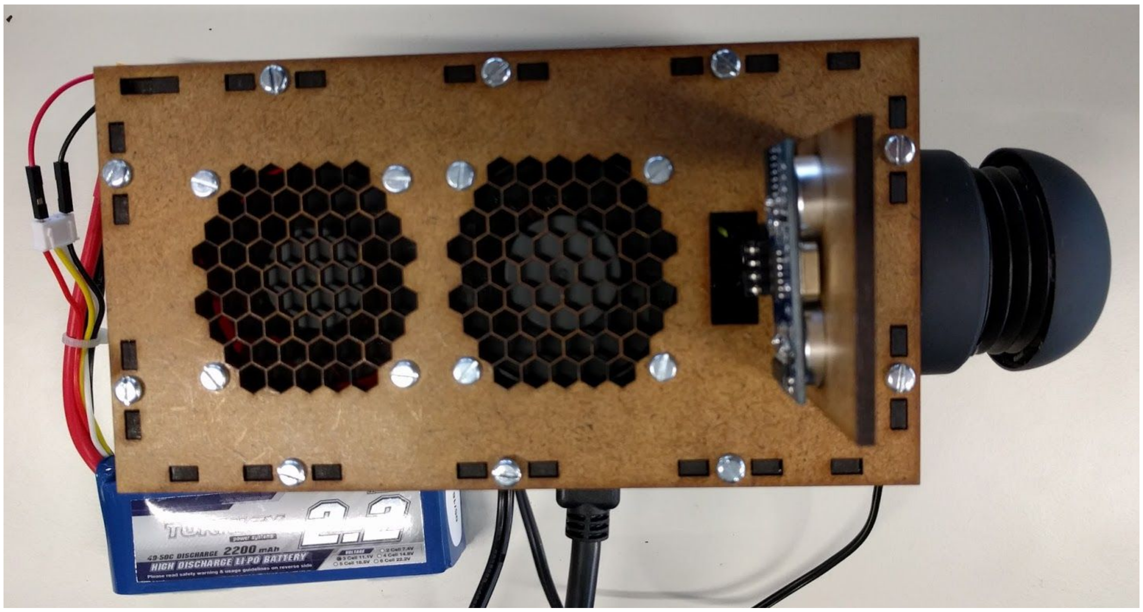



*Figure 2.17: Pi Cam Car with top plate and the remaining short side plate fixed*

- The last step now is to place the battery under the car in the space between the DC motors and close it with the bottom plate. Use 6 M3x12 screws and 6 M3 nuts to fix the bottom plate to the car.

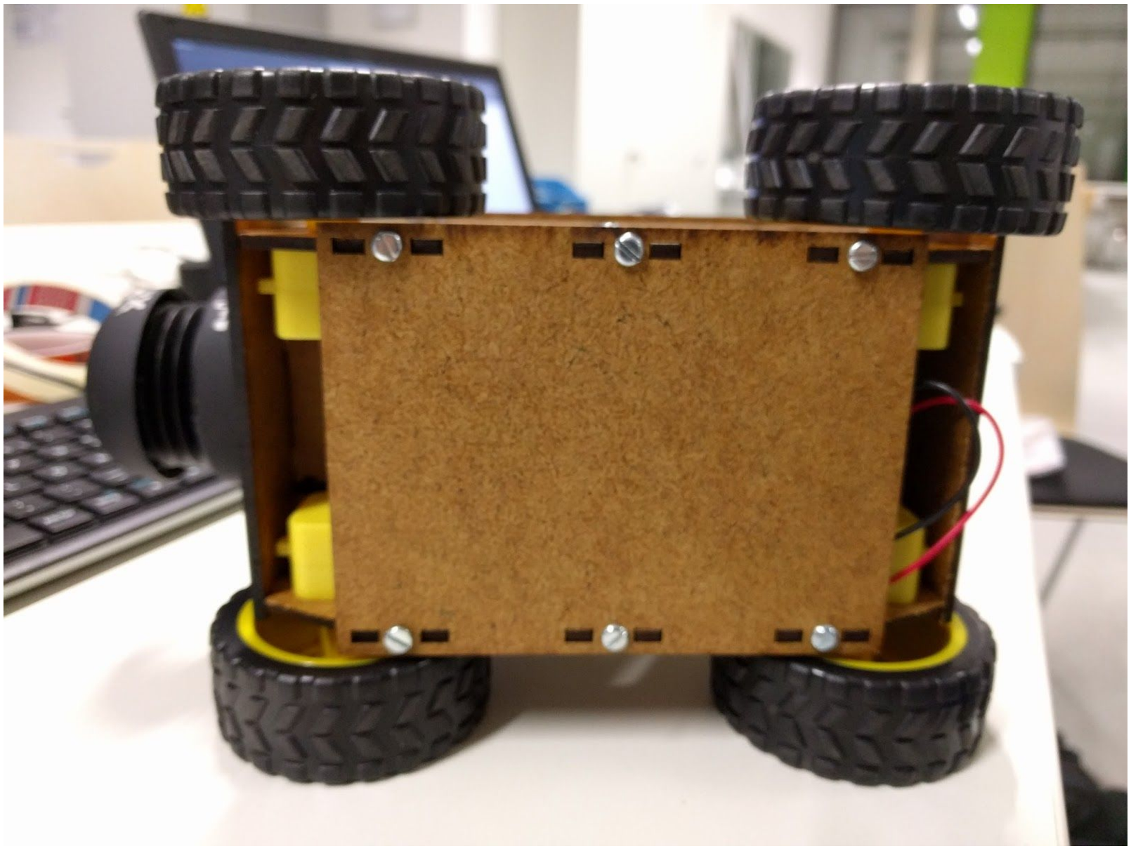


*Figure 2,18: Pi Cam Car with battery and bottom plate attached*

## Part 3: Setup Raspberry Pi

1. Install the Raspbian OS

- Download latest Raspian OS:
    - https://www.raspberrypi.org/downloads/raspbian/

- See Installation instructions for help:
    - https://www.raspberrypi.org/documentation/installation/installing-images/README.md

2. Update Raspbian

```
sudo apt-get update
sudo apt-get upgrade
```

3. Configure Raspian

sudo raspi-config

“1 Expand Filesystem” > “Ok”
“6 Enable Camera” > “Yes” > “Ok”
“9 Advanced Options” > “A8 Audio” > “1 Force 3.5mm (‘headphone’) jack”
“Finish” > “Yes”

4. Free some space

Free some space on your Pi by deleting the Wolfram engine (nearly 700MB):

```
sudo apt-get purge wolfram-engine
```

## Part 4: Install needed software on your Pi

1. Install remote desktop

Installing Remote desktop on the Pi will allow us to control and view the outputs from the Pi on our computer or laptop, as if we are having a monitor plugged into the Pi.
This function will be used in this project to view the output of the PiCam to our laptop to view what the car actually “sees”.

On the Raspberry input the following codes:

```
sudo apt-get install xrdp
sudo reboot
```

On user's computer or laptop:

Windows: remote desktop client is already installed
Mac OS X: remote desktop client is already installed
or find "Microsoft Remote Desktop" on iTunes
Linux: rdesktop: http://www.rdesktop.org/

2. Install OpenCV 3

Follow the instructions on the following link, make sure to use a virtual environment as suggested and use Python 3: http://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libgtk2.0-dev libatlas-base-dev gfortran python2.7-dev python3-dev
```

```
cd ~ && wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip && wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip && unzip opencv_contrib.zip && unzip opencv.zip && rm opencv_contrib.zip && rm opencv.zip
```

```
wget https://bootstrap.pypa.io/get-pip.py && sudo python get-pip.py && sudo pip install virtualenv virtualenvwrapper && sudo rm -rf ~/.cache/pip
```

```
echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile && echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile && echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile && source ~/.profile && mkvirtualenv cv -p python3 && source ~/.profile && workon cv
```

You should now see "(cv)" all the way left in your console. This it needed for the following commands.

```
pip install numpy && cd ~/opencv-3.1.0/ && mkdir build && cd build && cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
    -D BUILD_EXAMPLES=ON ..
```

The following command will compile OpenCV, the execution will take around 1h 12m on a Raspberry Pi 3.

```
make -j4
```

```
sudo make install && sudo ldconfig && cd /usr/local/lib/python3.4/site-packages/ && sudo mv cv2.cpython-34m.so cv2.so && cd ~/.virtualenvs/cv/lib/python3.4/site-packages/ && ln -s /usr/local/lib/python3.4/site-packages/cv2.so cv2.so
```

5. Install GPIO in cv virtual environment

```
pip install RPi.GPIO
```

6. Install picamera with array package

```
pip install "picamera[array]"
```

7. Install the audio player

```
sudo apt-get install mpg321 -y
```

8. Install the ultrasonic sensor package

```
sudo apt-get install python3-smbus
```

## Part 5: Test runs and Final run

Obtain the codes for the Pi from **code_package.zip** and unzip it to a folder. In order to test run, we need to connect the Raspberry Pi to a monitor.

1. Test run the car with the DC motors

On the Raspberry Pi, run the python file **motors/test_motorcontroller.py** in the unzipped folder.
The motor will make a distinct sound upon startup to notify that the DC motors work fine and there is no connection error.

2. Test run the car with ultrasonic sensor

Run the python file **sensors/test_srf02.py** in the unzipped folder.
The screen will print out the distance (in cm) from the sensor to any blocking surface at 90 degree angle in front of it.

3. Test run the Pi camera with face recognition feature

Proceed to run the python file **camera/test.py** in the unzipped folder.
We will now be able to view the streaming video from the camera, with any face recognised by the program being surrounded by a square with distinct outline, as in the photo below:

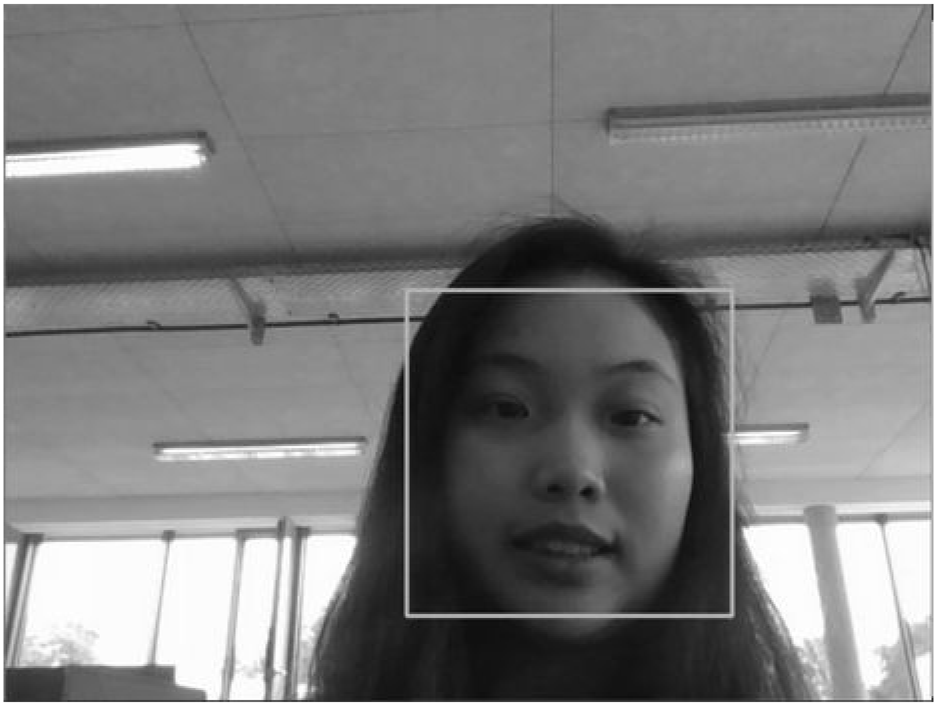


*Figure 5.1: Example of face recognition feature*

In order to stop video streaming, press “q”.

4. Test run the car with loudspeaker

Run the following code in the console to test for sound, replace 100% with a preferred volume level:

```
amixer sset PCM,0 100%
aplay sound/sound-files/clear.wav
```

When there is no setup problem, the included sound file **clear.wav** will be played.
You can always add your own audio files in the **sound-files** folder and play them by modifying the codes provided accordingly.

Now that all individual parts are working correctly, proceed to run the file **PiCamCar.py** and unplug your HDMI cable. Then fix the wheels onto the car so it can start running!

Open Remote Desktop to see in real time where the car is going.
In order to disable view mode, go to **PiCamCar.py** and set the variable `show` to `False` instead of `True` which is default.

The car is finally ready to run around!