SELF DRIVING CAR

By Felix Vandemaele

Installation Manual

Contents

1. Utilities:

Materials

- RC-car
- Raspberry Pi 3B+
- Micro-SD card min. 8GB
- Micro-SD card reader
- Power bank min. output of 2A
- Mini breadboard x2
- Micro USB cable
- Jumper wires
- Servo motor
- Blue painters' tape
- MCP 3008
- L293D DC controller
- Pi camera v2
- Some type of glue, I used a glue gun
- Screwdrivers
- PVC plate or cardboard could work as well
- Ethernet cable
- Potentiometer 10K
- Resistor 220Ω
- Clean Raspbian image
- Soldering iron

Software

- Visual Studio Code (or other editor with remote connect function)
- Putty
- WinSCP
- Win32DiskImager

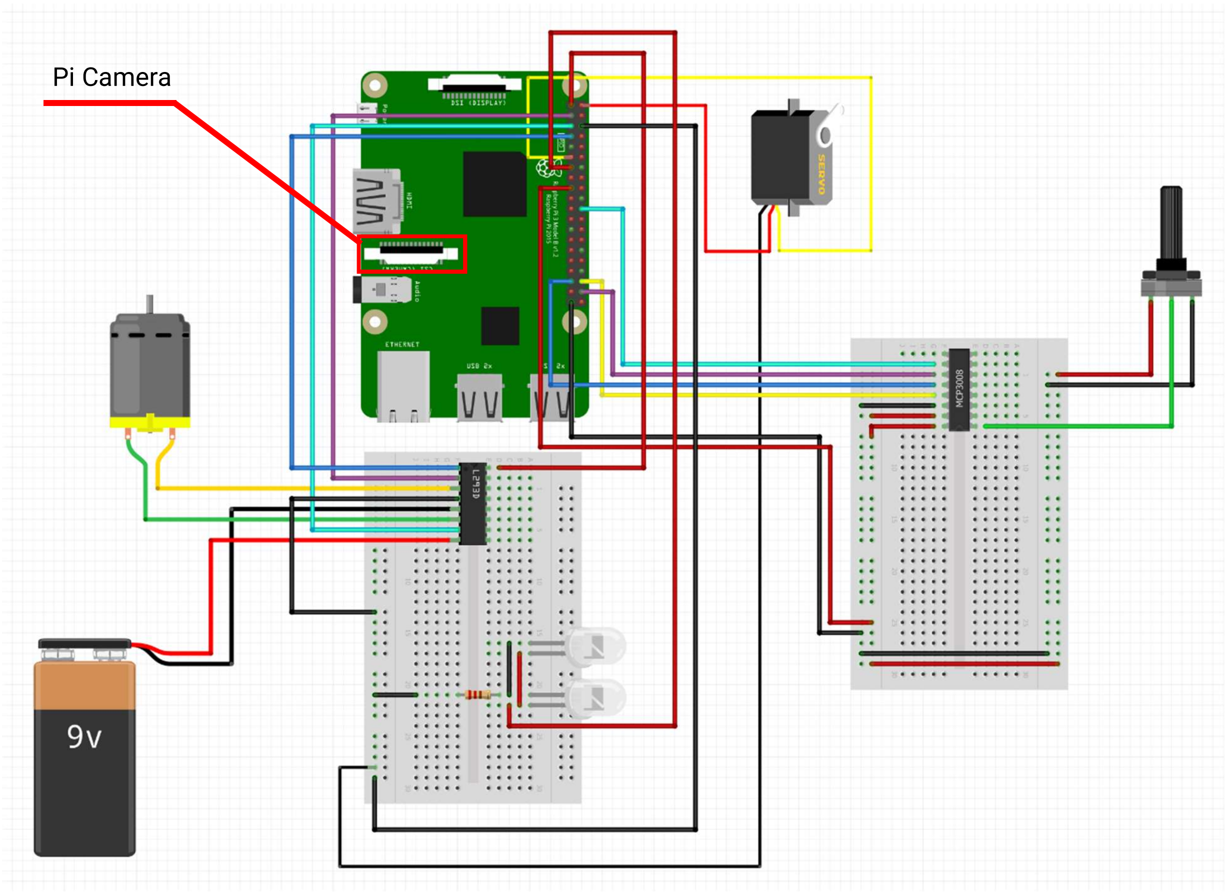
2. Setup Raspberry Pi

For this installation manual I assume that you have a clean install of Raspbian installed on your Raspberry Pi and that you have some basic knowledge of Linux commands and Raspberry Pi. If you don't know how to install your Raspberry Pi, first complete the following document.

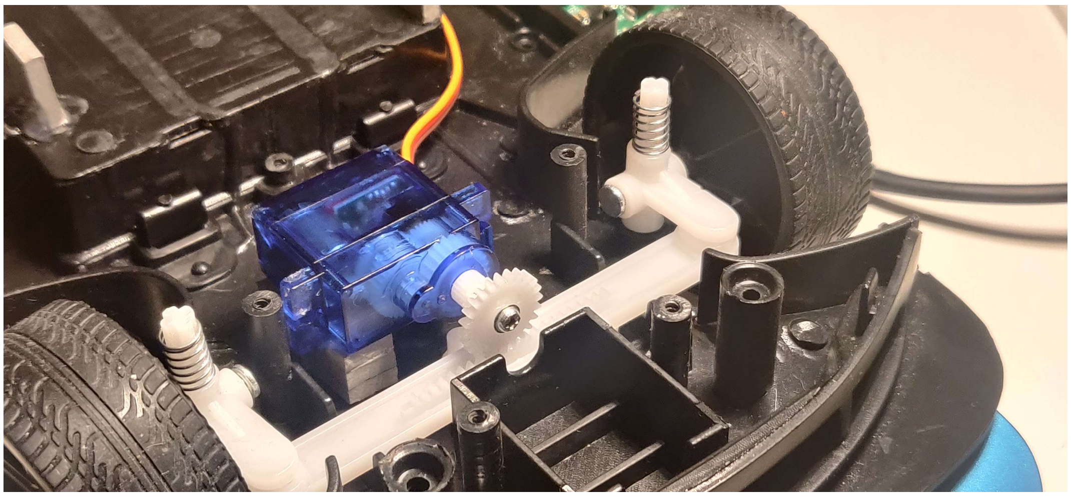
'Preparing your Raspberry Pi V1.docx'

3. Hardware

Connect everything to your Raspberry Pi using the following scheme.



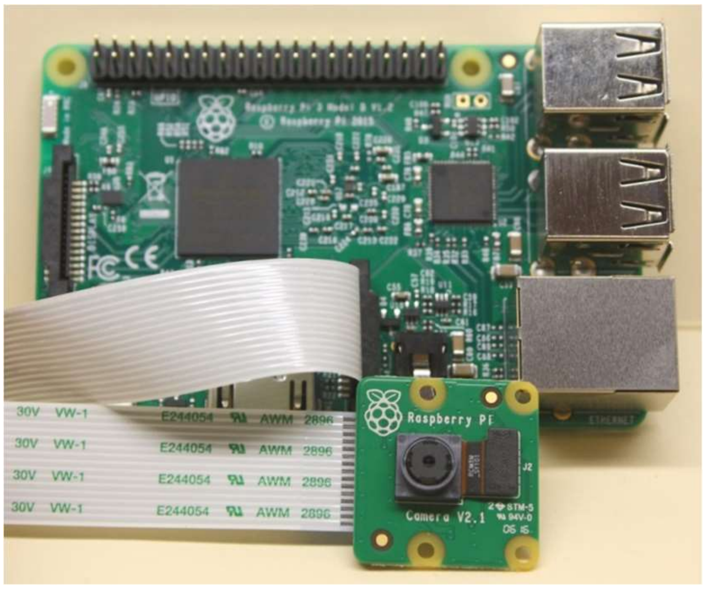
The car came with a DC motor for driving and another DC motor for steering. The problem of steering with a DC motor is that you can only steer left or right, but no angle in between. To steer multiple angles, you'll need to replace the DC motor with a servo motor. The steering mechanism is different for every RC-car, so you'll have to be creative to fit the servo. In the picture below you can see how I did it in my car.



Most RC cars use a standard DC motor to drive. Desolder the wires and connect them to the breadboard. The motor uses the regular 4.5V battery that came with the car. Most standard DC motors can handle up to 9V. This will give it more torque but will also generate some heat.

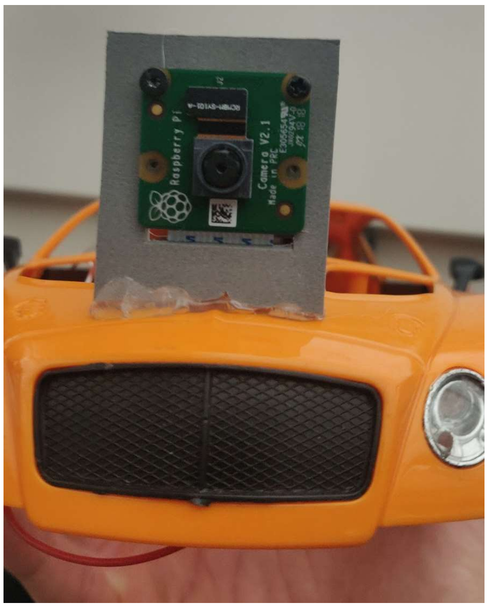
My RC-car already has headlights, so I connected them with the Raspberry Pi as well. I used a resistor, so they won't burn out.

Connect the flat cable of the Pi Camera to the bracket on the Raspberry Pi. Just like on the picture below.

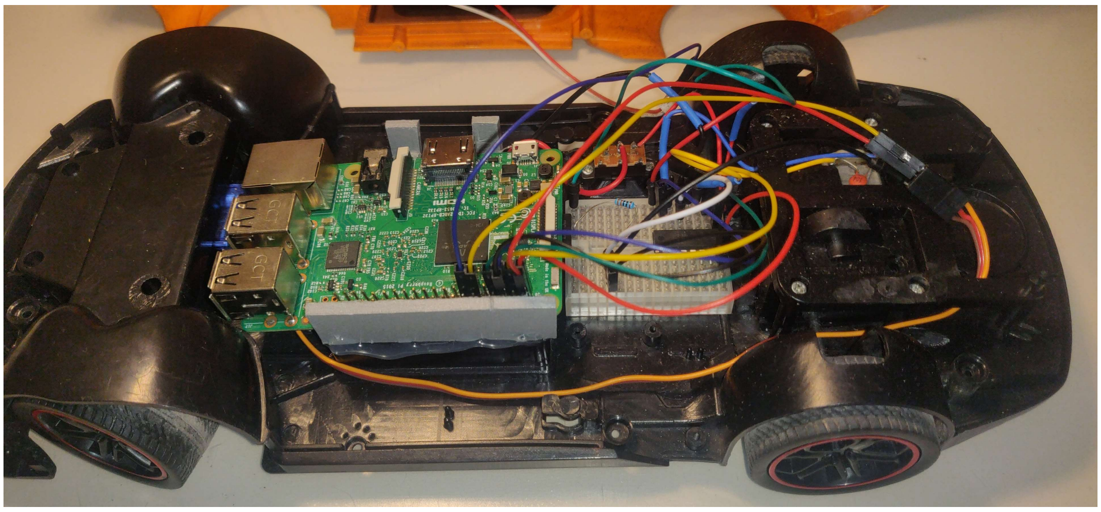


I mounted the Pi Camera on a piece of PVC board and hot glued it to the front of the car.

Make sure it's very sturdy, because you don't want the camera angle to change at any time. This could affect the performance of your car by a lot!!



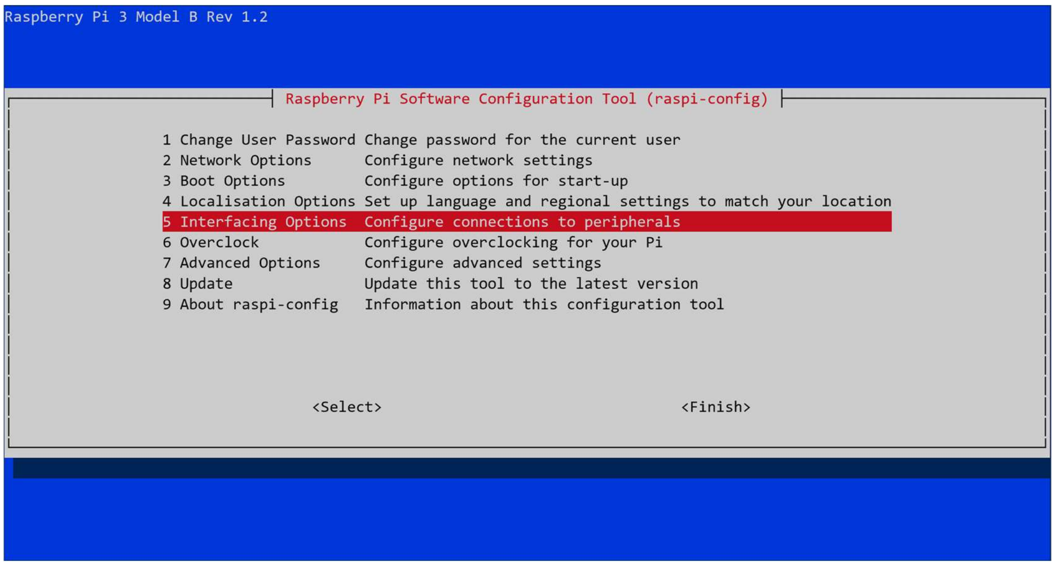
When everything is connected you can stuff it all inside the car. I used a hot glue gun and pieces of PVC board to make sure everything is secured (I think cardboard could do the job as well, but PVC board is a bit sturdier).



4. Enabling Hardware

To use the Pi Camera and the potentiometer, you must enable some settings.

```
sudo raspi-config
```



```
Raspberry Pi 3 Model B Rev 1.2

Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

                 <Select>                        <Finish>
```

Under ‘Interfacing Options’ enable ‘Camera’ and ‘SPI’.

Select ‘Finish’ and reboot the pi.

5. Installing Code

Clone the code from github into the user folder.

```
git clone https://github.com/vandemaelefelix/Project-4.git
```

6. Installing Python Packages

You can try to install the newest version of each package, but these versions worked for me so I advise you to use them.

Update Raspberry Pi

Begin with updating your Raspberry Pi so you have all the latest packages available.

```
sudo apt update
```

```
sudo apt upgrade
```

Installing OpenCV

!! The latest version of OpenCV doesn't work on Raspberry Pi so you'll have to install an older version !!

```
pip3 install opencv-python==3.4.6.27
```

```
pip3 install opencv-contrib-python==3.4.3.18
```

Installing TensorFlow

!! The latest version of Tensorflow doesn't work on Raspberry Pi so you'll again have to install an older version !!

```
pip3 install tensorflow==1.14.0
```

Install PiCamera

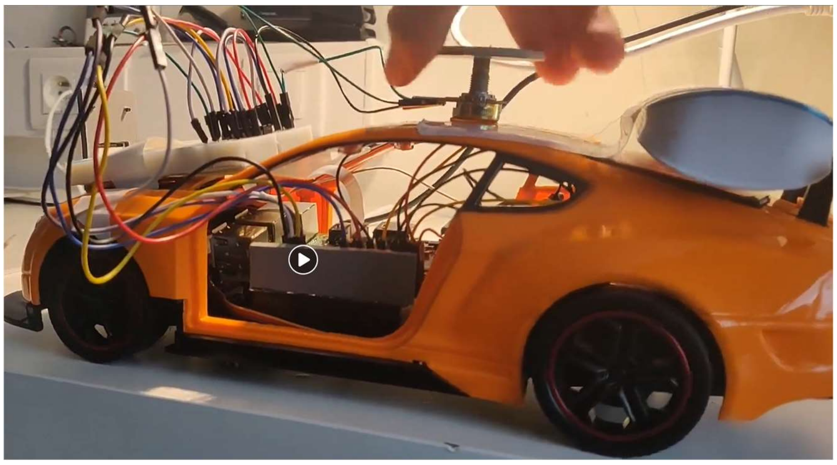
```
pip3 install picamera==1.13
```

Install Numpy

```
pip3 install numpy==1.16.2
```

7. Capturing Data

There are several ways to collect data. I mounted a potentiometer on my car to drive it manually. While driving, the camera takes pictures (size of pictures is 640 x 480) and saves them to a data folder.



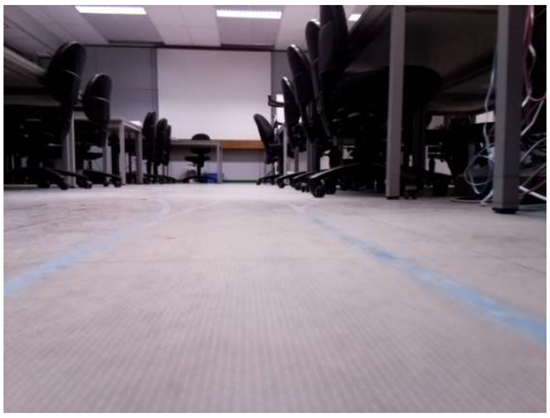
Depending on how robust you want the model to be, you can collect more and more data. I have about 3000 pictures. They're all taken on one track so it could work on other tracks as well, but not as good as on the original.

To start the 'capture_data.py' script, type:

```
cd code/
```

```
python3 capture_data.py -p
```

When you have enough images, you can transfer them to your computer to train the model.



8. Training Model

We'll train the model on our PC because Raspberry Pi isn't strong enough to handle such heavy tasks.

Before training you'll have to install several pip packages.

Installing Jupyter Lab

```
pip3 install jupyterlab
```

Installing Tensorflow

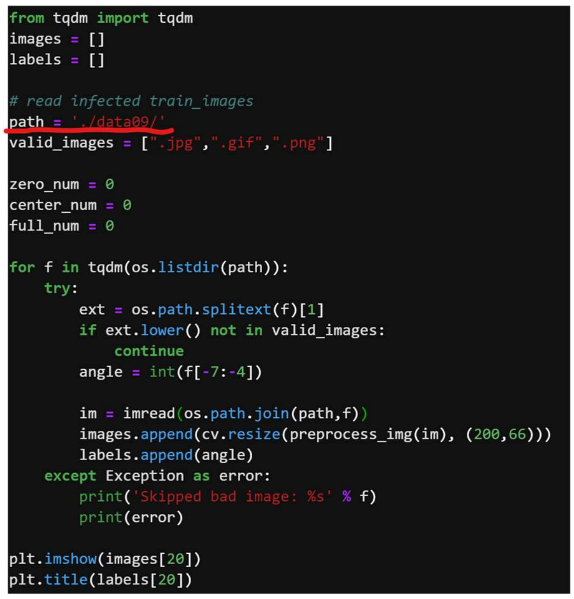
```
pip3 install tensorflow==2.1.0
```

Installing Keras

```
pip3 install Keras==2.3.1
```

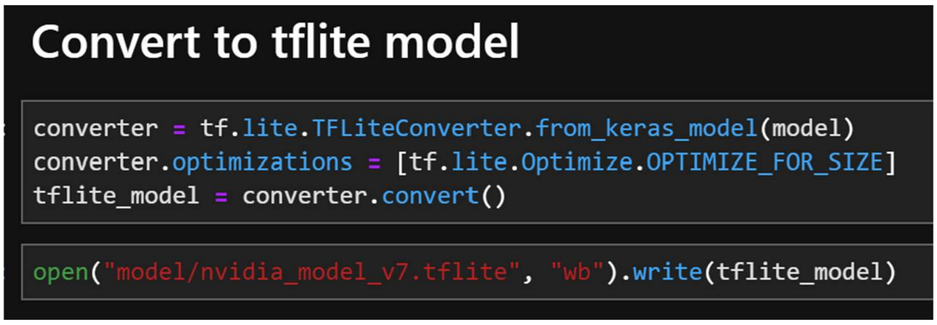
In the GitHub repository under 'neural_network' there is a Jupyter Notebook with the code to train your model.

Inside the Jupyter Notebook, there is a path that refers to a data folder. Edit this path so it leads to the data you just collected.



```python
from tqdm import tqdm
images = []
labels = []

# read infected train_images
path = './data09/'
valid_images = [".jpg",".gif",".png"]

zero_num = 0
center_num = 0
full_num = 0

for f in tqdm(os.listdir(path)):
    try:
        ext = os.path.splitext(f)[1]
        if ext.lower() not in valid_images:
            continue
        angle = int(f[-7:-4])

        im = imread(os.path.join(path,f))
        images.append(cv.resize(preprocess_img(im), (200,66)))
        labels.append(angle)
    except Exception as error:
        print('Skipped bad image: %s' % f)
        print(error)

plt.imshow(images[20])
plt.title(labels[20])
```

After training you can save the model as tflite file.



Convert to tflite model

```python
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
tflite_model = converter.convert()
```

```python
open("model/nvidia_model_v7.tflite", "wb").write(tflite_model)
```

When the model is converted, you can upload it to the Raspberry Pi using WinSCP.

9. Start Driving

You're all done now. The Raspberry Pi is completely setup. Follow the instructions in the user manual to start the car.