**Donkey RC Car Setup**

**A close up of a machine

Description automatically generated**

## Powering up the Donkey Car

## To turn on the donkey car you will need two batteries. one Ni-Mh battery that fits into the bottom slot of the donkey car and has some pins that hold it in. There is a small connector right by the slot that the Ni-Mh battery will plug into. In order to power the Pi, there is a Li-Po battery which will plug into the Pi thru a usb to micro usb cable. Once you have these two batteries plugged in the donkey car will be ready to roll or setup.

## Setup the Wi-Fi connection for Pi

To setup Wi-Fi access to the donkey car you will want to power up the Pi on the vehicle. Connect the pi to a monitor thru the HDMI port on the Pi and plug in a usb mouse and keyboard. You will want to select the Wi-Fi you wish to connect to on the raspberry pi.

**Connecting to the Pi**

If you followed the above instructions to add wifi access, your Pi should now be connected to your wifi network. Now you need to find its IP address so you can connect to it via SSH.

The easiest way (on Ubuntu) is to use the findcar donkey command. You can try ping raspberrypi.local. If you've modified the hostname, then you should try: ping <your hostname>.local. This will fail on a windows machine. Windows users will need the full IP address (unless using cygwin).

If you are having troubles locating your Pi on the network, you will want to plug in an HDMI monitor and USB keyboard into the Pi. Boot it. Login with:

* Username: **pi**
* Password: **1234**

Then try the command:

ifconfig wlan0

If this has a valid IPv4 address, 4 groups of numbers separated by dots, then you can try that with your SSH command. If you don't see anything like that, then your wifi config might have a mistake. You can try to fix with

sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

If you don't have a HDMI monitor and keyboard, you can plug-in the Pi with a CAT5 cable to a router with DHCP. If that router is on the same network as your PC, you can try:

**ping** **raspberrypi**.local

Hopefully, one of those methods worked and you are now ready to SSH into your Pi. On Mac and Linux, you can open Terminal. On Windows you can install [Putty](http://www.putty.org/), [one of the alternatives](https://www.htpcbeginner.com/best-ssh-clients-windows-putty-alternatives/2/), or on Windows 10 you may have ssh via the command prompt.

If you have a command prompt, you can try:

**ssh** **pi**@**raspberrypi**.**local**

or

**ssh** **pi**@<**your** pi ip address>

or via Putty.

* Username: **pi**
* Password: **1234**
* Hostname:<your pi IP address>

# **Drive your car**

If you are not already, please ssh into your vehicle.

### Start your car.

**Put your car in a safe place where the wheels are off the ground.** This is the step were the car can take off.

Open your car's folder and start your car.

cd ~/mycar

python manage.py drive

Once the car has been trained to use autopilot you can launch with the command:

python manage.py drive --model ~/mycar/models/mypilot.h5

This script will start the drive loop in your car which includes a part that is a web server for you to control your car. You can now control your car from a web browser at the URL: <your car's IP's address>:8887

**Building Donkey Car**

To build the donkey car you will need the following parts:

* Exceed Racing Desert Short Course Truck 1/16 Scale Ready to Run 2.4ghz (AA Blue)

<https://www.amazon.com/Exceed-Racing-Desert-Course-2-4ghz/dp/9269802086/ref=as_li_ss_tl?_encoding=UTF8&pd_rd_i=9269802086&pd_rd_r=78JFQN575NX4QN9YW9NA&pd_rd_w=WcZLu&pd_rd_wg=GWXCG&psc=1&refRID=78JFQN575NX4QN9YW9NA&linkCode=sl1&tag=donkeycar-20&linkId=1bc61faa0d090a48af6fbaeeaa069b0d>

* Element14 Raspberry Pi 3 B+ Motherboard <https://www.amazon.com/ELEMENT-Element14-Raspberry-Pi-Motherboard/dp/B07BDR5PDW?tag=donkeycar-20>
* HiLetgo 2pcs PCA9685 16 Channel 12-Bit PWM Servo Motor Driver IIC Module for Arduino Robot

<https://www.amazon.com/HiLetgo-PCA9685-Channel-12-Bit-Arduino/dp/B07BRS249H/ref=sr_1_4?keywords=servo+driver&qid=1578327015&sr=8-4>

* SainSmart Wide Angle Fish-Eye Camera Lenses for Raspberry Pi Arduino

<https://www.amazon.com/gp/product/B00N1YJKFS?tag=donkeycar-20>

* M3x10 screws <https://www.amazon.com/Screws-Mushroom-Phillips-Self-Tapping-Electronic/dp/B07NQCG6JP?tag=donkeycar-20>
* M2x6 screws <https://www.amazon.com/uxcell-Stainless-Phillips-Tapping-Screws/dp/B01KXTSW6Q?tag=donkeycar-20>
* 32GB Micro SD Card <https://www.amazon.com/dp/B06XWMQ81P/ref=twister_B07V2PRSXC?_encoding=UTF8&psc=1>

# Power Bank 10000mQSah,TONV Portable External Cell Phone Li-Polymer Battery’s 2 Input and 2 Output

<https://www.amazon.com/10000mah-TONV-Portable-Li-Polymer-Compatible/dp/B078TFHXVY/ref=sr_1_1?keywords=tonv&qid=1578335174&sr=8-1>

* Vibration-Damping Sandwich Mount https://www.mcmaster.com/5822k4

**3d Printing the body of the donkey car and adapters**

* Download the 3d print files for the Donkey Car located on the GitHub <https://github.com/michiganaerospace>
* You will need to print the Top and bottom Donkey body, the Back and front adapter, and the ping sensor mount
* Put the stl files into the slicer of your choice
* Print all the parts of the donkey car
* Remove / clean any residue from the 3d prints
* Assemble (<http://docs.donkeycar.com/guide/build_hardware/>)

**Installing software onto the donkey car**

* Plug raspberry pi into a HDMI monitor, power supply, usb keyboard, and usb mouse.
* Connect to a Wi-Fi source
* Update and upgrade

Open console and type the following code

|  |
| --- |
| sudo apt-get update  sudo apt-get upgrade |

* Changing Raspi-config

Open console and type the following code

|  |
| --- |
| sudo raspi-config |

* + Change default password to 1234
  + Enable interfacing options – I2C
  + Enable interfacing options – Camera
  + Advanced Options – Expand filesystem

Choose and hit enter then reboot

* Install Dependency’s

Open console and type the following code or copy and paste from <http://docs.donkeycar.com/guide/robot_sbc/setup_raspberry_pi/#step-5-connecting-to-the-pi>

|  |
| --- |
| sudo apt-get install build-essential python3 python3-dev python3-pip python3-virtualenv python3-numpy python3-picamera python3-pandas python3-rpi.gpio i2c-tools avahi-utils joystick libopenjp2-7-dev libtiff5-dev gfortran libatlas-base-dev libopenblas-dev libhdf5-serial-dev git ntp |

* Install Optional OpenCV Dependencies

Open console and type the following code or copy and paste from <http://docs.donkeycar.com/guide/robot_sbc/setup_raspberry_pi/#step-5-connecting-to-the-pi>

|  |
| --- |
| sudo apt-get install libilmbase-dev libopenexr-dev libgstreamer1.0-dev libjasper-dev libwebp-dev libatlas-base-dev libavcodec-dev libavformat-dev libswscale-dev libqtgui4 libqt4-test |

* Setup virtual environment

Open console and type the following code or copy and paste from <http://docs.donkeycar.com/guide/robot_sbc/setup_raspberry_pi/#step-5-connecting-to-the-pi>

|  |
| --- |
| python3 -m virtualenv -p python3 env --system-site-packages  echo "source env/bin/activate" >> ~/.bashrc  source ~/.bashrc |

* Install Donkeycar Python code

Open console and type the following code or copy and paste from <http://docs.donkeycar.com/guide/robot_sbc/setup_raspberry_pi/#step-5-connecting-to-the-pi>

Change to a directory you would like to be head of your project

|  |
| --- |
| mkdir projects  cd projects |

Get latest Donkeycar from Github

|  |
| --- |
| git clone https://github.com/autorope/donkeycar  cd donkeycar  git checkout master  pip install -e .[pi]  pip install tensorflow==1.13.1 |

* Install optional opencv

Open console and type the following code or copy and paste from <http://docs.donkeycar.com/guide/robot_sbc/setup_raspberry_pi/#step-5-connecting-to-the-pi>

|  |
| --- |
| sudo apt install python3-opencv |

If that doesn’t work try the next line

|  |
| --- |
| pip install opencv-python |

# Create your car application.

* Open your console and type the following to create a set of files to control your donkey

|  |
| --- |
| donkey createcar --path ~/mycar |

* To look at the myconfig.py in the newly created directory use the command

|  |
| --- |
| **nano** **myconfig**.py |

Each line has a comment mark. The commented text shows the default value. When you want to make an edit to over-write the default, uncomment the line by removing the # and any spaces before the first character of the option.

ie.

# STEERING\_LEFT\_PWM = 460

becomes

STEERING\_LEFT\_PWM = 500

* Configure I2C

Open your console and type the following

|  |
| --- |
| sudo apt-**get** install i2c-tools  sudo i2cdetect -y 1 |

This should show you a grid of addresses like if all plugged in correctly if not the same check wiring:

0 1 2 3 4 5 6 7 8 9 a b c d e f

**00: -- -- -- -- -- -- -- -- -- -- -- -- --**

**10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**70: 70 -- -- -- -- -- -- --**

Calibrating Car

* To edit your config file use this code

|  |
| --- |
| nano ~/mycar/myconfig.py |

## Steering Calibration

Make sure your car is off the ground to prevent a runaway situation.

* Turn on your car.
* Find the servo cable on your car and see what channel it's plugged into the PCA board. It should be 1 or 0.
* Run donkey calibrate --channel <your\_steering\_channel> --bus=1
* Enter 360 and you should see the wheels on your car move slightly. If not enter 400 or 300.
* Next enter values +/- 10 from your starting value to find the PWM setting that makes your car turn all the way left and all the way right. Remember these values.
* Enter these values in myconfig.py script as STEERING\_RIGHT\_PWM and STEERING\_LEFT\_PWM.

## Throttle Calibration

* Find the cable coming from your ESC and see what channel it goes into the PCA board. This is your throttle channel.
* run donkey calibrate --channel <your\_throttle\_channel> --bus=1
* Enter 370 when prompted for a PWM value.
* You should hear your ESC beep indicating that it's calibrated.
* Enter 400 and you should see your cars wheels start to go forward. If not, its likely that this is reverse, try entering 330 instead.
* Keep trying different values until you've found a reasonable max speed and remember this PWM value.

Reverse on RC cars is a little tricky because the ESC must receive a reverse pulse, zero pulse, reverse pulse to start to go backwards. To calibrate a reverse PWM setting...

* Use the same technique as above set the PWM setting to your zero throttle.
* Enter the reverse value, then the zero throttle value, then the reverse value again.
* Enter values +/- 10 of the reverse value to find a reasonable reverse speed. Remember this reverse PWM value.

Now open your myconfig.py script and enter the PWM values for your car into the throttle controller part:

* THROTTLE\_FORWARD\_PWM = PWM value for full throttle forward
* THROTTLE\_STOPPED\_PWM = PWM value for zero throttle
* THROTTLE\_REVERSE\_PWM = PWM value at full reverse throttle