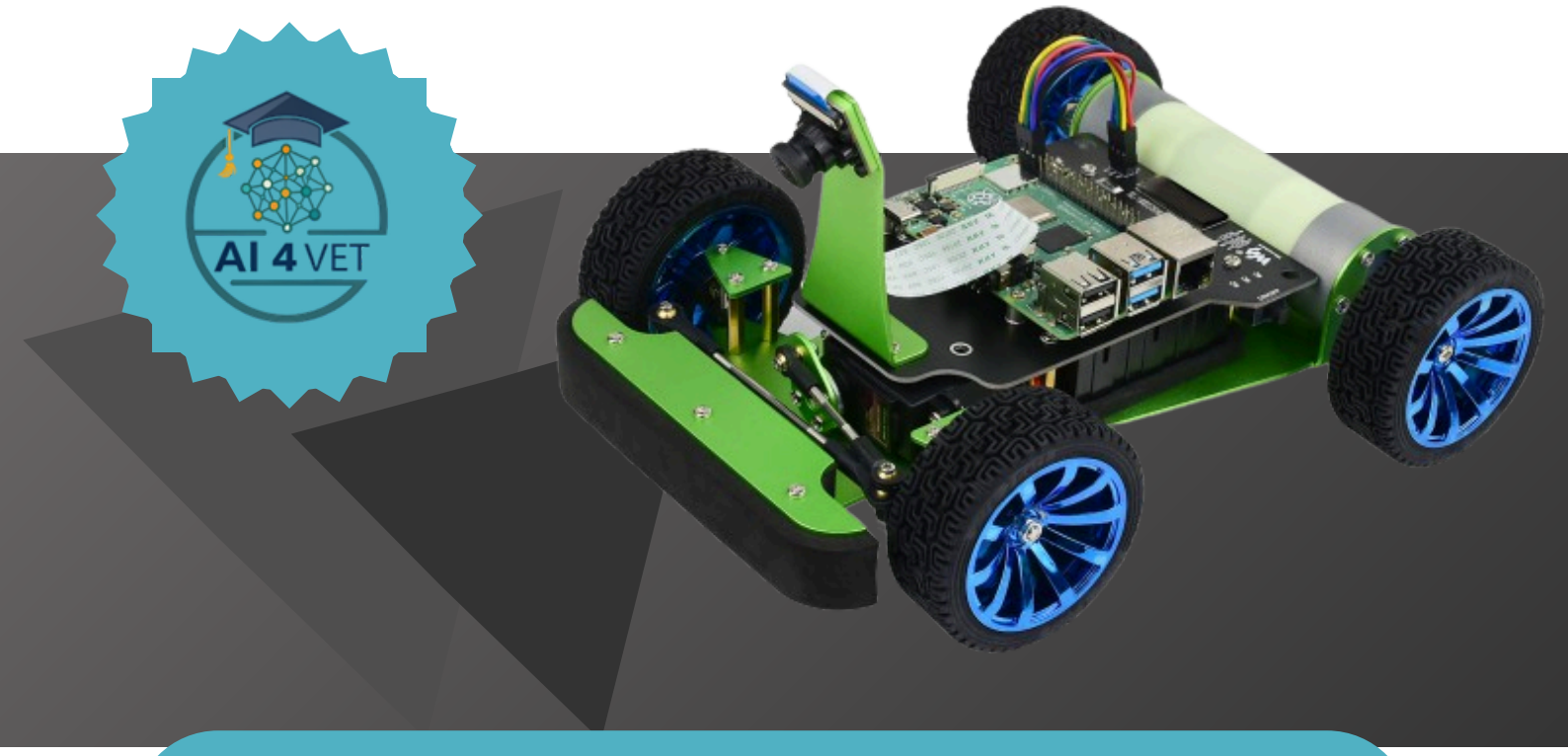
Co-funded by
the European Union

PIRACER

AI KIT

KA220-VET - Cooperation partnerships in vocational education and training
Project Title: ***AI tools for VET schools***
Document Date: February 2026



Tensorflow OpenCV Python Keras

Author: Bojan Ćirić,
co-authors:
Aleksandar Madić,
Boban Blagojević)

Introduction

This is a high-performance AI racing kit designed for Raspberry Pi. The expansion board integrates OLED, servo motor drive circuit, battery protection circuit, and so on, which does not need to solder the circuit, supports unloading charging, and does not need to repeatedly disassemble the battery. Fully upgraded chassis, front and rear axle differential lock, hydraulic shock absorber, four-wheel-drive independent suspension, carbon fiber chassis, and high-speed carbon brush motor. The software is fully compatible with the DonkeyCar open source project and supports functions such as deep learning, automatic driving, and automatic visual inspection.

Feature

- Support four 2600mAh 18650 batteries (not included), two parallel and two series output currents are larger, and the motor power is stronger.
- On-board HY2120 + AOD514 lithium battery protection circuit, with anti-overcharge, anti-over-discharge, anti-over-current, and short-circuit protection functions.
- On-board FP5139 automatic buck-boost voltage regulator circuit can provide stable 5V voltage to Raspberry Pi.
- On-board 0.91-inch 128 × 32 resolution OLED, real-time display of car IP address, memory, power, etc.
- On-board AINA219 chip is convenient for real-time monitoring battery voltage and charging current.

DonkeyCar for Pi-

Setup Raspberry Pi

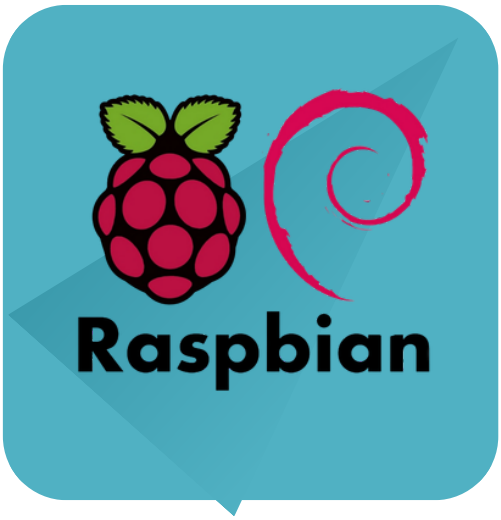
Step 1. Update System

- Flash the Bookworm version image.

Assuming your Raspberry Pi is already installed with the image and has started up. And the WiFi is connected, SSH is enabled, the I2C interface is started, and the file system is expanded.

- Update the system

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



Step 2. Setup Virtual Environment

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

Modifying the .bashrc in this way will automatically enable this virtual environment each time you log in. To return to the system Python, you can enter deactivate.

Step 3. Install Dependency Libraries

```
sudo apt install libcap-dev libhdf5-dev libhdf5-serial-dev
```

Step 4. Install Donkeycar Python Code

Create a project directory

```
mkdir projects
cd ~/projects
```

Get the latest donkeycar program from GitHub

```
git clone https://github.com/waveshareteam/donkeycar
cd donkeycar
git checkout master
pip install -e .[pi]
```

Step 5. Install OLED Display Service

Run the following command to install the OLED display service. OLED will display information such as the current IP address, battery voltage, charging current, etc.

```
cd ~
git clone https://github.com/waveshare/pi-display
cd pi-display
sudo ./install.sh
```

Step 6. Create Donkeycar from Template

Use the following command to create a donkeycar instance

```
cd ~/projects/donkeycar
donkey createcar --path ~/mycar
```

After the program runs, a series of files to control donkeycar will be automatically generated in the ~/mycar directory.

DonkeyCar for Pi-

WEB Control

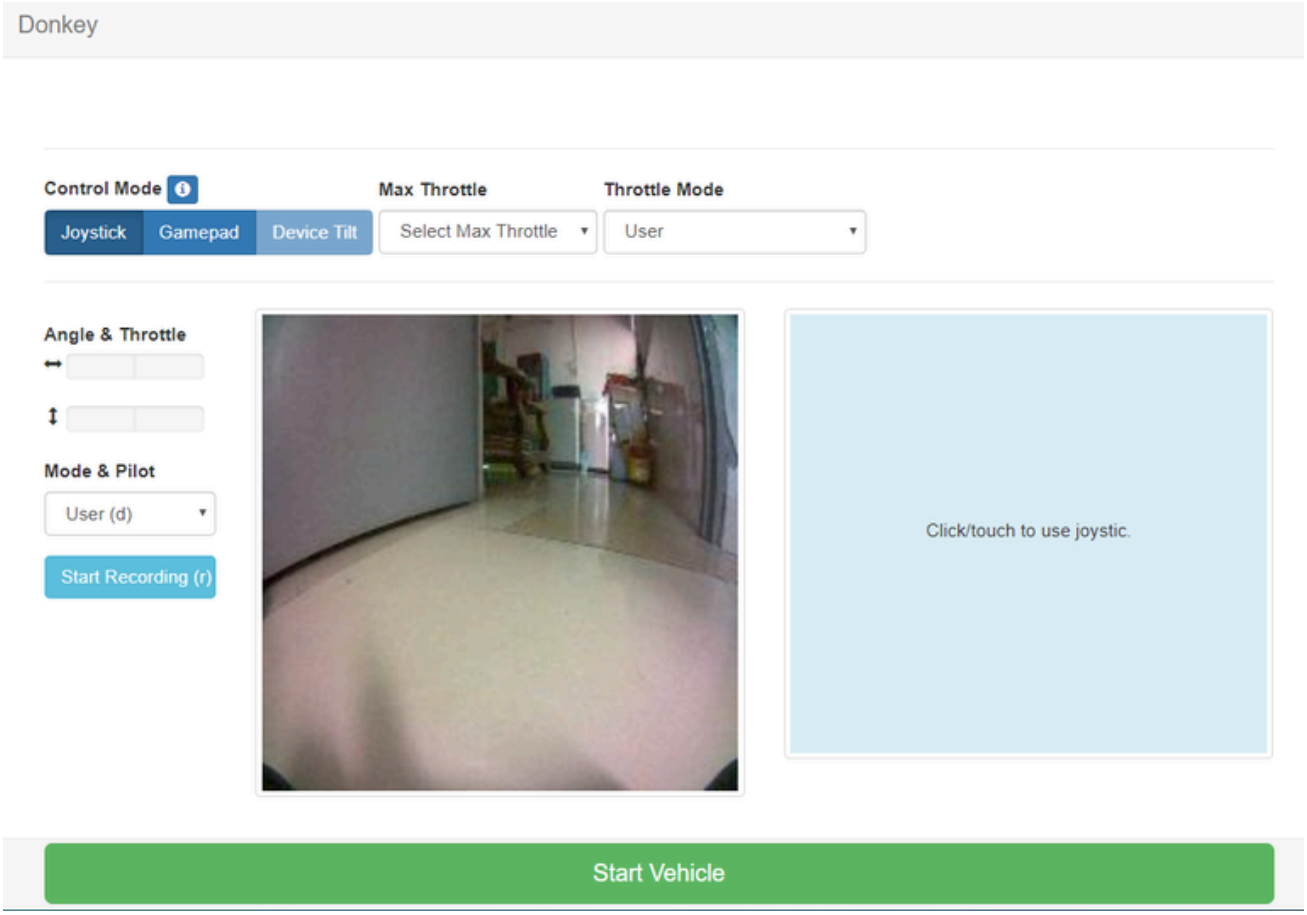
WEB Controlling

Open the terminal and run the follow commands:

```
pi@raspberrypi:~$ source ~/env/bin/activate
(env) pi@raspberrypi:~$ cd mycar/
(env) pi@raspberrypi:~/mycar$ python manage.py drive
```

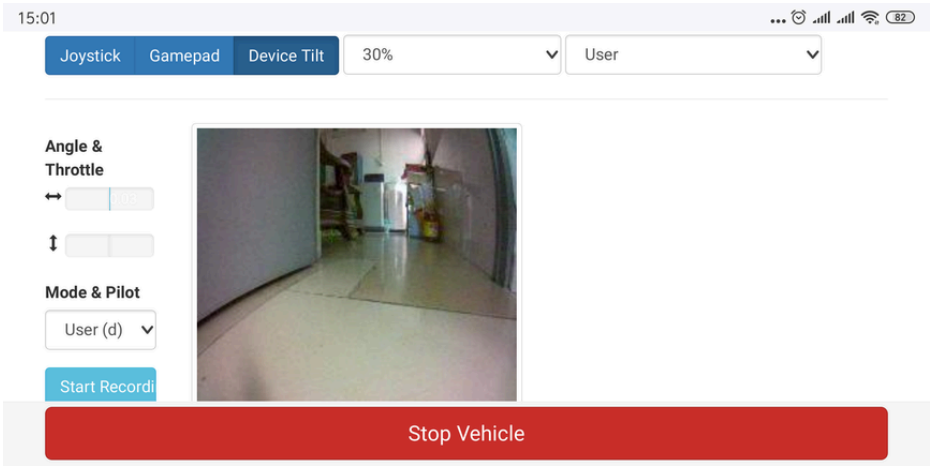
Note that you cannot use sudo in front of the command python manage.py drive, otherwise it cannot be run successfully.
Open the Chrome in host pc and go to **https://<raspberrypi_ip_address>:8887** which is the WEB control page of Donkeycar

Use the following command to create a donkeycar instance



Choose the Max speed for the Max Throttle option. Click the right joystick windows and drag it to move the PiRacer. The Angle & Throttle bars will display the steering angle and the motor speed. You can click the Start Recording button to capture images and save to path ~/mycar/data

You can also go into the WEB page by your telephone.

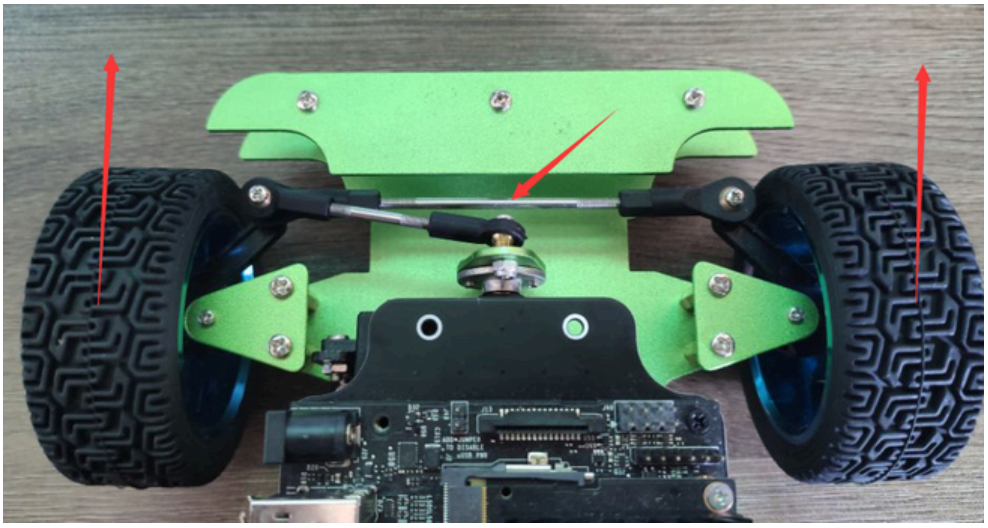


You can also control the car by keyboard
Space: Stop Moving
R: Recording
I: Speed up
K: Slow down
J: Turn left
L: Turn right

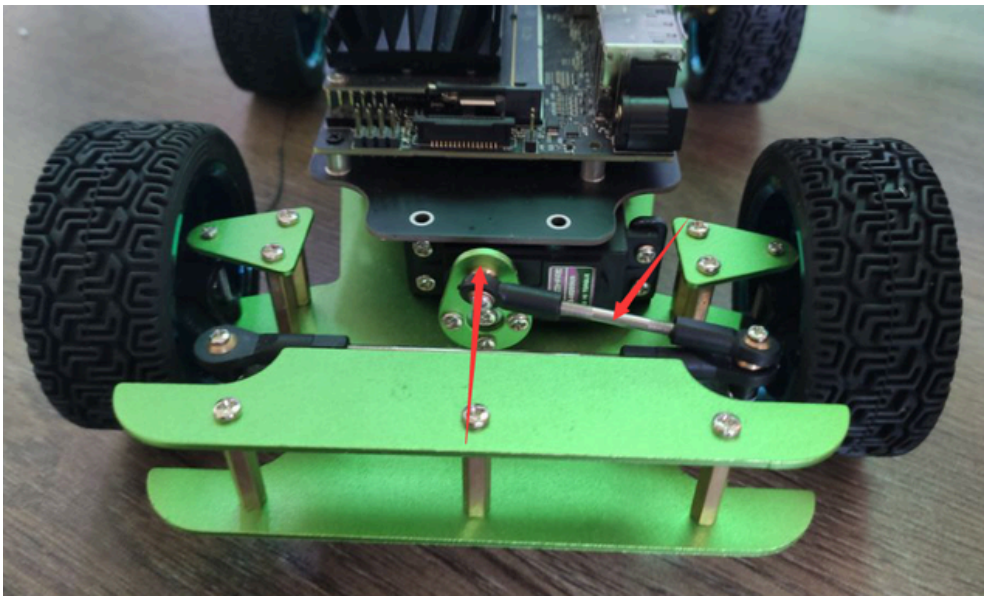
DonkeyCar for Pi-

Calibrate DonkeyCar

To make sure that the Donkeycar can turn around successfully, you need to calibrate the car both in hardware and software.
Keep the front wheels forward when assembling. You need to adjust the length of the pull-bars.



Adjust the steering servo



Open the terminal and use commands to calibrate

```
cd ~/mycar
donkey calibrate --channel 0 --bus=1
```

Type value 360 (or 300, 400) to check if the steering servo turns

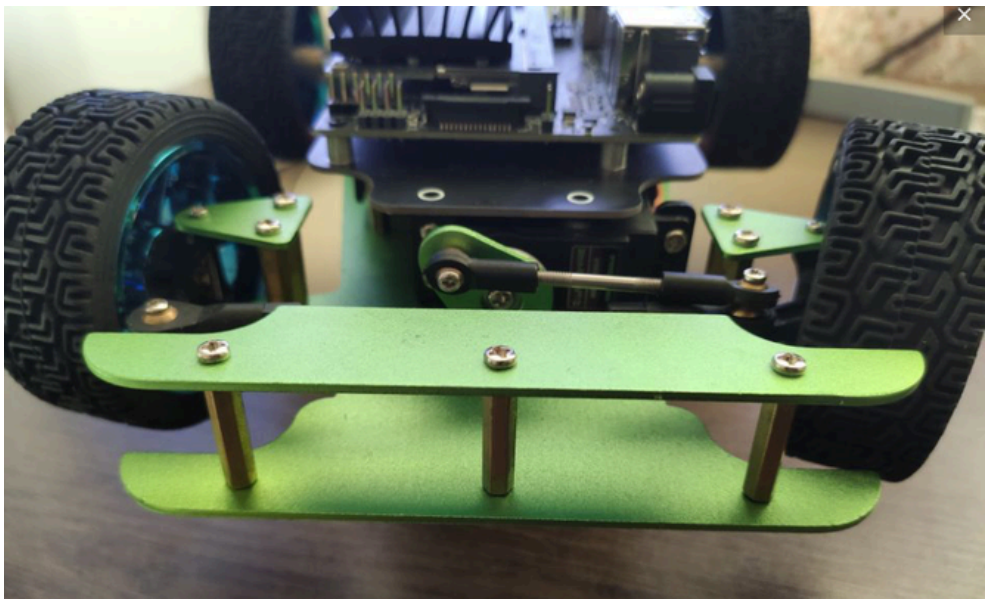
```
(env) pi@raspberrypi:~/mycar $ donkey calibrate --channel 0 --bus=1
using donkey v3.1.1 ...
sombrero enabled
init PCA9685 on channel 0 address 0x40 bus 1
Using PWM freq: 60

Enter a PWM setting to test ('q' for quit) (0-1500): 300
Enter a PWM setting to test ('q' for quit) (0-1500):
```

Adjust the value to make the servo turns to the center place (keep forward) and remember the value. For example, if the value 330 could work and let the servo keep in the center, you can test if 230 and 430 can let the servo turns to the far left and the far right.
Then Once you get the far left data and the far-right data, you can modify the config.py file and change the
STEERING_LEFT_PWM and **STEERING_RIGHT_PWM**.

```
#STEERING
STEERING_CHANNEL = 0            #channel on the 9685 pwm board 0-15
STEERING_LEFT_PWM = 460         #pwm value for full left steering
STEERING_RIGHT_PWM = 290        #pwm value for full right steering
```

Note that you cannot set the angle in which the steering servo turns too big or too small. You can adjust it as below (far right)

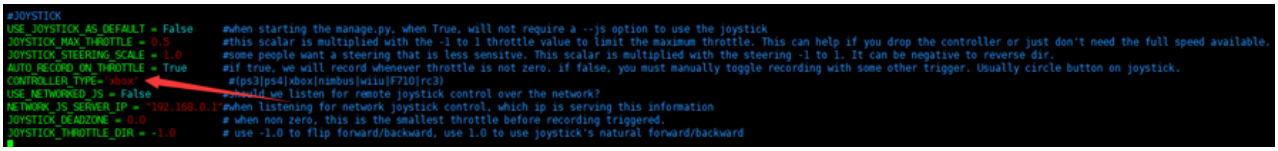


Finally, check if the throttle is set correctly as below

```
#THROTTLE
THROTTLE_CHANNEL = 0            #channel on the 9685 pwm board 0-15
THROTTLE_FORWARD_PWM = 4095     #pwm value for max forward throttle
THROTTLE_STOPPED_PWM = 0        #pwm value for no movement
THROTTLE_REVERSE_PWM = 4095     #pwm value for max reverse throttle
```

DonkeyCar for Pi- Teleoperation

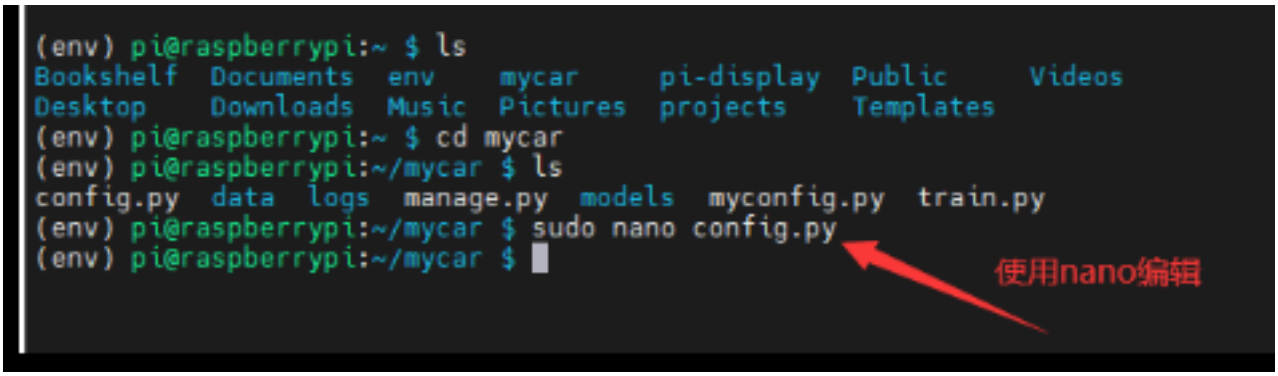
Open file **~/mycar/confgi.py** and find the blow part. Check if the **CONTROLLER_TYPE** is set as Xbox



Connect the USB adapter of Gamepad to Raspberry Pi
Open a terminal and run the following commands:

```
cd ~/mycar
python manage.py drive --js
```

If you want to enable the gamepad control by default, you can modify the config.py file, set **USE_JOYSTICK_AS_DEFAULT** to **True**. Edit with nano below:



```
#JOYSTICK
USE_JOYSTICK_AS_DEFAULT = True          #when starting the manage.py, when T
JOYSTICK_MAX_THROTTLE = 0.5             #this scalar is multiplied with the
JOYSTICK_STEERING_SCALE = 1.0           #some people want a steering that i
AUTO_RECORD_ON_THROTTLE = True          #if true, we will record whenever t
CONTROLLER_TYPE='xbox'                   #(ps3|ps4|xbox|nimbus|wiiu|F710)
USE_NETWORKED_JS = False                #should we listen for remote joysti
NETWORK_JS_SERVER_IP = "192.168.0.1"#when listening for network joystick
JOYSTICK_DEADZONE = 0.0                 # when non zero, this is the smalles
JOYSTICK_THROTTLE_DIR = -1.0            # use -1.0 to flip forward/backward

#For the categorical model, this limits the upper bound of the learned
#it's very IMPORTANT that this value is matched from the training PC co
#and ideally wouldn't change once set.
MODEL_CATEGORICAL_MAX_THROTTLE_RANGE = 0.5
```

Please press Ctrl+x to save, and then press Enter.

Wireless Gamepad

1. L-1: reduce the accelerator ratio.
1. L-2: Forward
2..R-1: Increase the accelerator ratio.
2. R-2: backward
3.Y: Emergency stop (driving forward).
4. B: Automatically record data, need to be connected to the Internet.

5. A: Switch mode
6. X: Erase the first ten data.
7. Front and rear joystick: control forward and backward movement.
8. Left and right joystick: control left and right movement.
9. START (triangle button): emergency stop after rushing out of the track, delete the last 5S record.

DonkeyCar for Pi-

Collect Data

Practice driving around the track a couple times.
When you're confident you can drive 10 laps without mistake, restart the python mange.py process to create a new tub session. Press Start Recording if using web controller. The joystick will auto record with any non-zero throttle.
If you crash or run off the track press Stop Car immediately to stop recording. If you are using a joystick tap the Triangle button to erase the last 5 seconds of records.
After you've collected 10-20 laps of good data (5-20k images) you can stop your car with Ctrl-c in the ssh session for your car.
The data you've collected is in the data folder in the most recent tub folder.

DonkeyCar for Pi-

Train Data

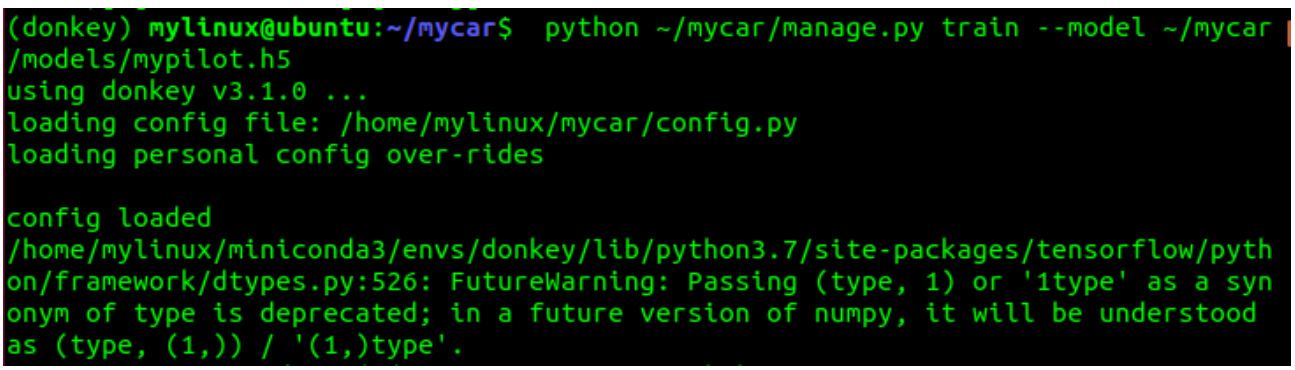
In a new terminal session on your host PC use rsync to copy your cars folder from the Raspberry P

```
rsync -rv --progress --partial
pi@<your_pi_ip_address>:~/mycar/data/
~/mycar/data/
```

Train model

In the same terminal you can now run the training script on the latest tub by passing the path to that tub as an argument. You can optionally pass path masks, such as **./data/* or ./data/tub_?_17-08-28** to gather multiple tubs. For example:

```
python ~/mycar/manage.py train --tub <tub folder
names comma separated> --model ./models/mypilot.h5
```



It will cost a long time to train a model, please be patient.
After training, you can get a model, you need to copy the module to your raspberry Pi and test it.

```
rsync -rv --show-progress --partial ~/mycar/models/
pi@<your_ip_address>:~/mycar/models/
```

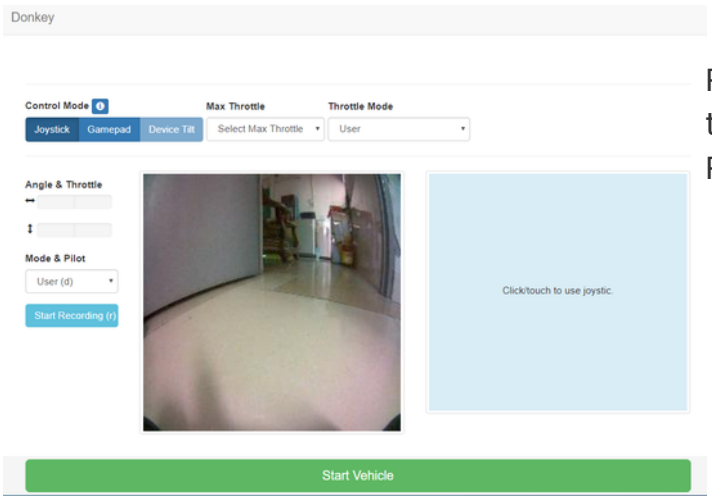
DonkeyCar for Pi-

Auto driving

Open a terminal and run the following commands:

```
pi@raspberry:~$ source ~/env/bin/activate
(env) pi@raspberry:~$ cd mycar/
(env) pi@raspberry:~/mycar$ python manage.py drive --
model ~/mycar/models/mypilot.h5
```

Open a browser and go to **http://<raspberrypi_ip_address>:8887**



Put your car in the map/floor, set the Mode & Pilot option to Local Pilot and test it.