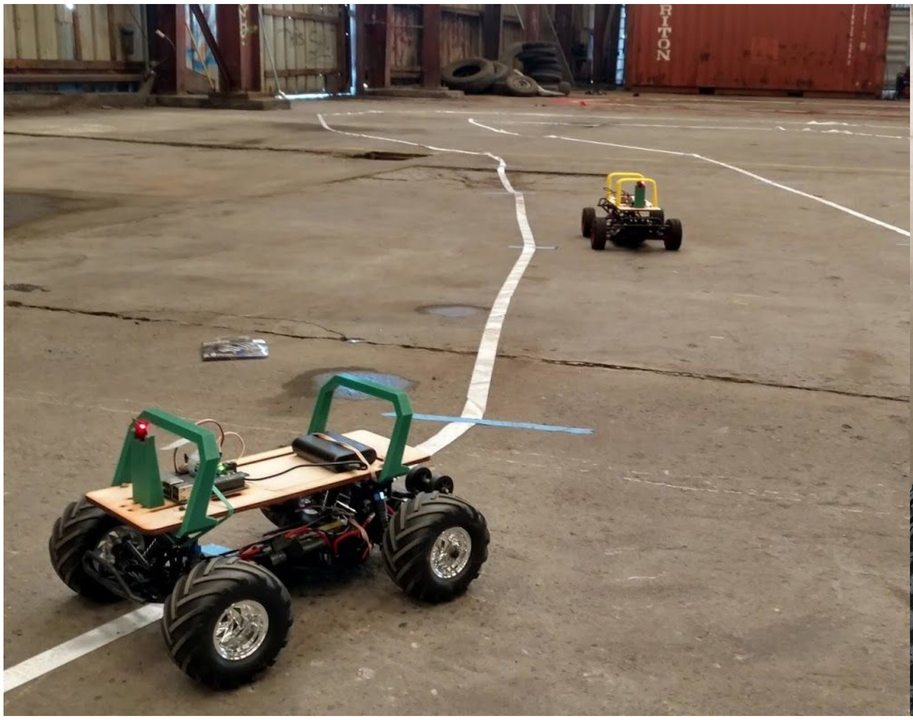
How to Build a Donkey Vehicle

This tutorial will show you how to modify an RC car with a Raspberry Pi and camera to make it self driving using the donkey python code. This car was designed to compete in the @DIYrobocars racing series in Oakland, CA.

Status: Vehicles can complete an autonomous lap around a lined course going ~1mph.



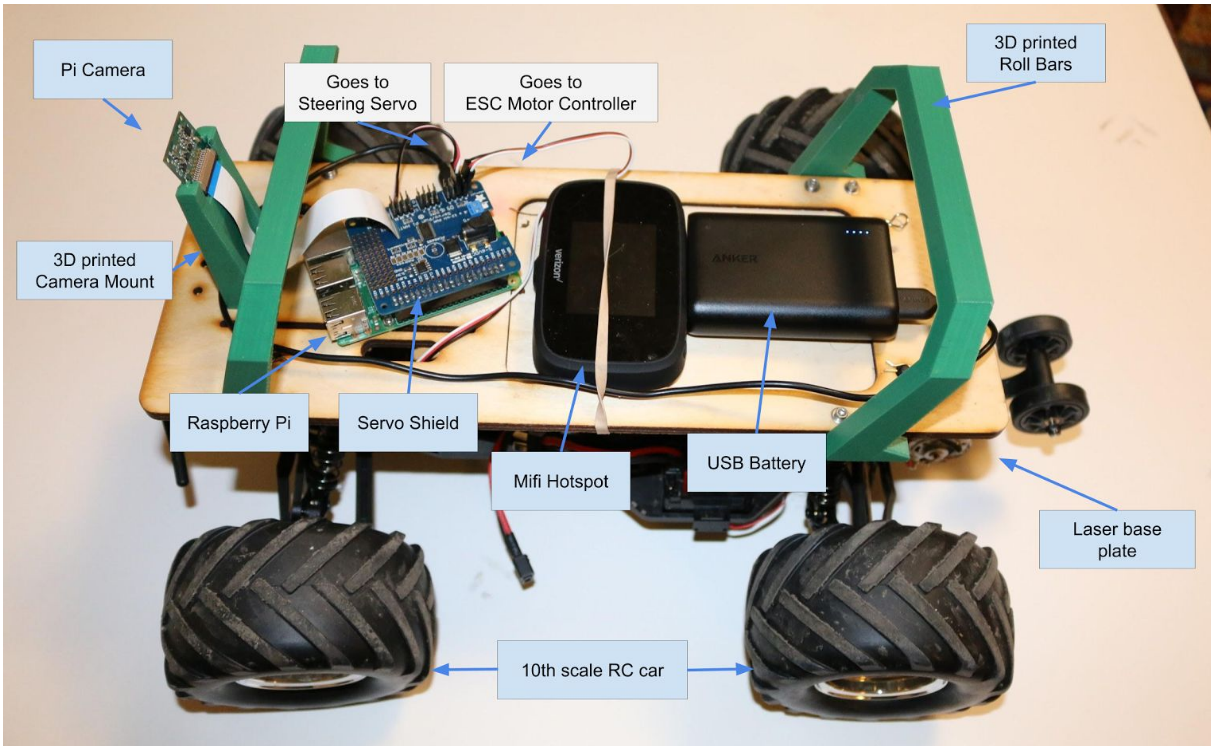
Skills Required

- Basic familiarity with bash command line + python scripting language.
- Basic soldering skills.
- Basic ssh skills
- Optional: 3d printing & laser cutting.

Build Steps

1. Get Parts (~$200 + 3 days)
2. Setup Raspberry Pi (30 min)
3. Install Pi Camera (30 min)
4. Solder and Install Motor Shield ($45 min)
6. Assemble Vehicle. (1hr - 2hr depending on quality)
7. Drive your vehicle. (30 min)

Get Parts



Required Parts

- Raspberry Pi 3 B ($35)
- Micro SD Card Sandisk U3 extreme card recommended ($8)
- Raspberry Pi Camera V2 ($15), V1 also works but has a narrower field of view.
- Adafruit Servo Shield ($18)
- USB Battery Pack with micro USB cable($15)
- Remote Control (RC) CAR ($80-300) (see recommendations)

Optional Parts

These parts are used to build a sturdy frame for your car. If you don't have access to a 3D printer & laser cutter, you can make these parts out of cardboard or foamboard.

- Servo extension cable (if needed)
- M2.5 standoffs to mount the Raspberry Pi
- Various M3 hardware components for assembling the hardware.
- Camera mount (3D printed)
- Base plate (laser cut)
- Roll bars (3D printed)

Tools

Tools you'll need to build a vehicle but will not needed to operate it once it's built.

- Monitor with HDMI connector
- USB Mouse
- USB Keyboard
- Soldering Iron + Solder
- MicroSD Card Reader

Setup Raspberry Pi

Donkey uses a small linux computer called a Raspberry Pi to control the vehicle. The following instructions will get this computer booted and the necessary software installed.

Turn on your Raspberry Pi

1. Follow these instructions to install the Raspbian Jessie operating system using the NOOBS method on your SD card.
2. Put the SD card into your Raspberry Pi.
3. Power up your Raspberry Pi using the USB cable.
4. Connect to the internet.

Install Basic Libraries

Since the Raspberry Pi is not as fast as larger computers, it can take a long time to install python packages (ie. numpy & PIL) using pip. Luckily Adafruit has precompiled these libraries into packages that can be installed via `apt-get`.

1. Open a terminal (Ctrl-Alt-t) and upgrade your system packages.

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

3. Install necessary libraries

```
sudo apt-get install xsel xclip libxml2-dev libxslt-dev libzmq-dev libspatialindex-dev virtualenv
```

4. Pandas & Jupyter Requirements

```
sudo apt-get install python3-lxml python3-h5py python3-numexpr python3-dateutil python3-tz python3-bs4 python3-xlrd python3-tables python3-sqlalchemy python3-xlsxwriter python3-httplib2 python3-zmq
```

5. Scientific Python

```
sudo apt-get install python3-numpy python3-matplotlib python3-scipy python3-pandas
```

Install Donkey

Run the following commands in a bash terminal on your raspberry pi.

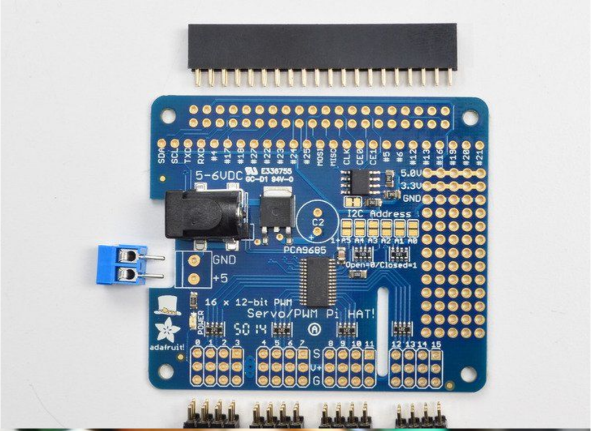
```
git clone http://github.com/wroscoe/donkey.git
cd donkey
virtualenv --system-site-packages -p python3 env
source env/bin/activate
pip install -e .[pi]
```

Install Pi Camera

Follow the regular instructions to install a Pi Camera. Essentially the steps are...

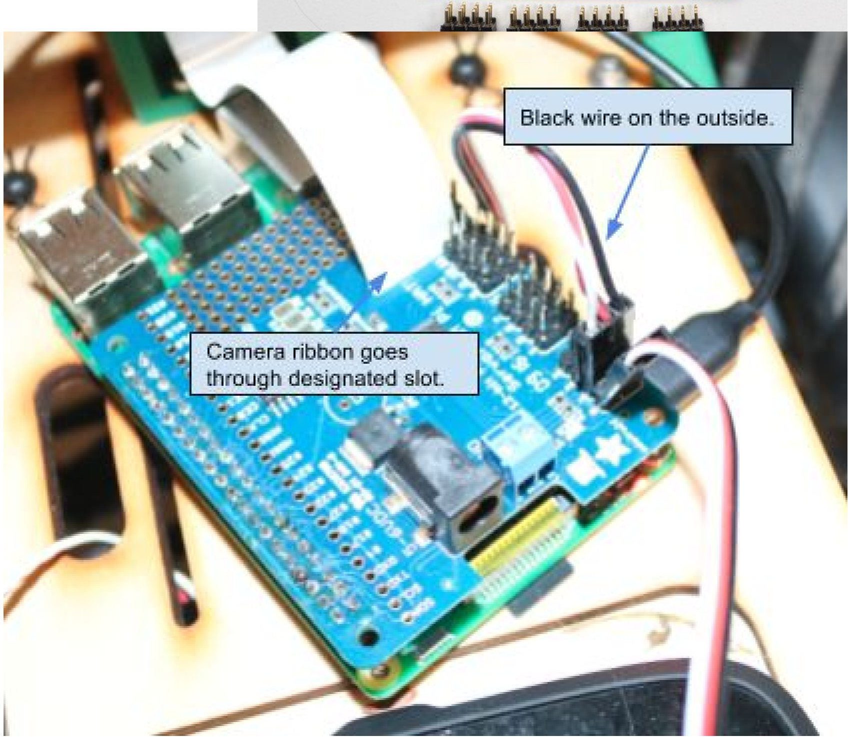
1. Connect your camera to the RPi.
2. Enable your camera in the Menu > Preferences > Raspberry Pi Config
3. Restart your Pi. `sudo reboot`

Solder and Install Motor Shield



Follow instructions to build a Adafruit 16-Channel PWM/Servo HAT for Raspberry Pi. Essentially the steps are…

1. Solder parts to circuit board.
2. Attach board to Raspberry Pi
3. Test if the board works.
4. Install needed libraries on the Raspberry Pi, this is tricky so follow the i2c instructions verbatim on the Adafruit website
5. Test that you can access the board from python.
6. Connect ESC to channel 0
7. Connect steering servo to channel 1



Black wire on the outside.

Camera ribbon goes through designated slot.

Assemble Vehicle

This section explains how to create the custom structural components for your vehicle and use them to securely attach your camera, pi and battery.

Base Plate

The base plate is the structural member that provides a flat surface to attach the Pi, battery and camera. The size of this piece will depend on the vehicle platform you choose. Here are CAD files for a few vehicle models. Modify as needed.

Camera Mount

The camera mount holds the Pi Camera at a standard height and angle. Use a 3D printer and this CAD file to make your own.

Roll Bars

Protective bars to prevent damage to your camera and electronics. Use a 3D printer and this CAD file to make your own.

Start a pilot server

A Donkey vehicle uses a separately hosted web server to route communications and perform autopilot calculations. To begin driving your vehicle from your phone you need to first start a donkey control server on your laptop or an remote server.

Use a personal computer (linux or OSX)

1. On a personal computer, follow the “Install Donkey” instructions you used to earlier. Except replace the last line with: `pip install -e .[server]`

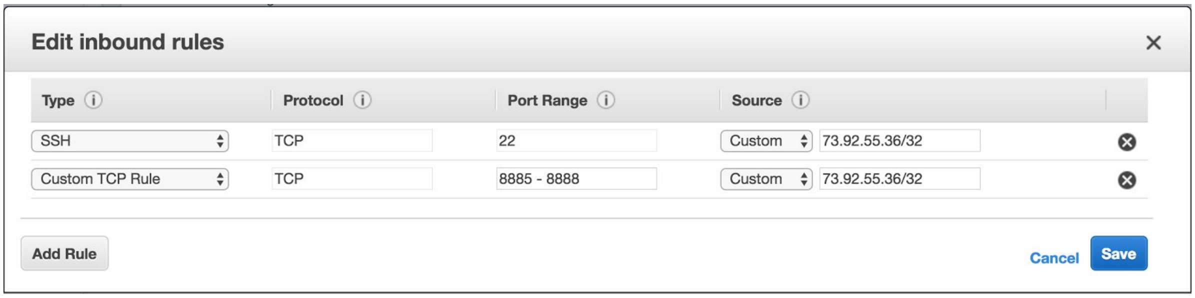
Use docker

1. Install docker if you don’t have it already.
2. Pull the docker image: `docker pull alanwells/donkey-server-jessie`
3. Start the docker container: `docker run  -i -t --net=host alanwells/donkey-server-jessie bash`
4. Run donkey server: python serve.py
5. Now you can go to localhost:8886 to drive your car.

Use an Amazon EC2 instance

Launch a remote server using Amazon EC2.

1. Login to Amazon EC2
2. Click Launch Instance
3. Make sure you're using the N. California region (recommend g2.2xlarge)
4. Click Community AMI’s
5. Search for “donkey” to find the instance (ami-d8e6b6b8)
6. Pull the most recent version of donkey. `git pull origin master`
7. When you launch your Security Groups should look something like this, but with the source IP changed to your IP. One note - we are not using a secure protocol or login, so this is the only thing protecting you from the internet.



Edit inbound rules

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 73.92.55.36/32
Custom TCP Rule	TCP	8885 - 8888	Custom 73.92.55.36/32

8. Once you launch the server either SSH to it or use the AWS console to get terminal access

```
cd donkey
git pull origin master
source env/bin/activate
```

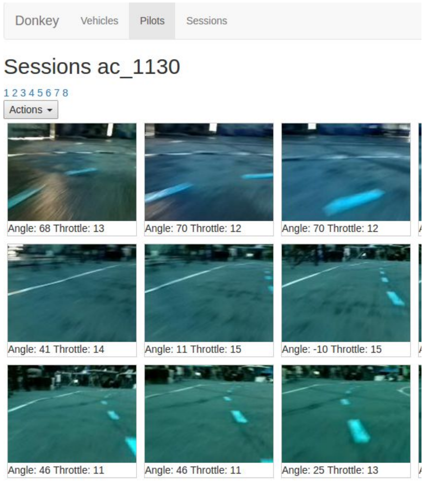
3. Start the server.

```
python scripts/serve.py
```

Drive the vehicle

1. Find your Raspberry Pi IP address. Assuming your Raspberry Pi is connected to the same local network as your computer, you can find the ip address of your Pi by running this command on your computer.
   ```
   sudo nmap -sP 192.168.1.0/24 | awk '/^Nmap/{ip=$NF}/B8:27:EB/{print ip}'
   ```
2. SSH into your Raspberry Pi using:
   ```
    ssh pi@<your raspberry pi ip address>
   ```
3. Activate the virtual environment.
   ```
    cd donkey
    source env/bin/activate
   ```
4. Connect your car to the server
   ```
    python scripts/drive --remote http://<your server ip>:8887
   ```
5. No go to <your server ip>:8887 in your web browser to start driving your car.

Random screenshots: (design/client side help needed :)



Sessions ac_1130

1 2 3 4 5 6 7 8

Angle: 68 Throttle: 13

Angle: 70 Throttle: 12

Angle: 70 Throttle: 12

Angle: 41 Throttle: 14

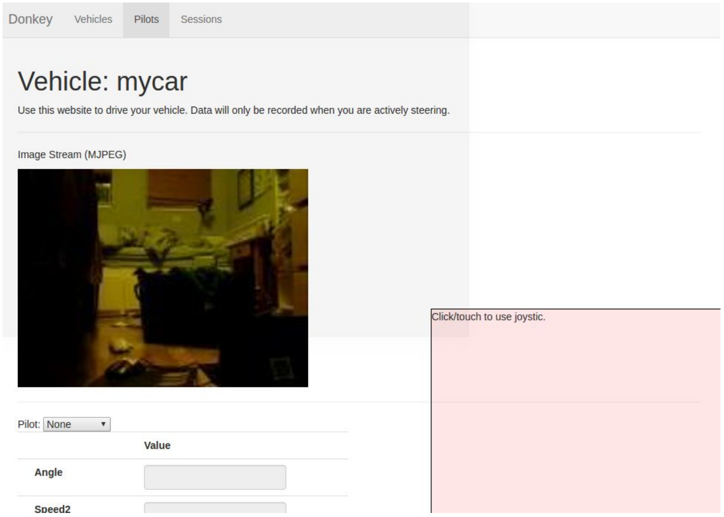
Angle: 11 Throttle: 15

Angle: -10 Throttle: 15

Angle: 46 Throttle: 11

Angle: 46 Throttle: 11

Angle: 25 Throttle: 13



Vehicle: mycar

Use this website to drive your vehicle. Data will only be recorded when you are actively steering.

Image Stream (MJPEG)

Click/touch to use joystic.

Pilot: None

	Value
Angle	
Speed2	

Train an autopilot

1. Python scripts/train.py --session <your session name> --pilot <name your autopilot>

Use an autopilot

1. Using a computer with a keyboard, refresh your vehicle page.
2. Click the Pilots dropdown to select the name of the pilot you created. Wait 5 seconds for the model to load.
3. Press S to give autopilot control of the steering.
4. Press D to give autopilot control of the steering and throttle.

TODO: make these functions available on phone.

FAQ

How do I calibrate my ESC?

If you have a brushless motor you'll need to calibrate your ESC before driving it. Brushed motors should be calibrated automatically. When not calibrated an ESC will periodically beep. When calibrated, it will beep a little song then stop beeping. See scripts/calibrate.py to see how this works.

What lag.

Vehicle Platform Recommendations

Angle Steering (RC Cars)

Almost any vehicle where the receiver is not integrated into the ESC motor controller will work. Here are the recommended cars that have been tested.

Scale	**Style**	**Model**	**Make**	**Motors**	**$**	**Tested**
1/10	Monster Truck	Conquest	Hellion	1 Brushed	?	Yes
1/16	Buggy	Exceed	NitroRCX	1 Brushed	70	No
1/10	Truck	Trooper	Basher	1 Brushless	160	Yes
1/16	Monster Truck	Magnet	NitroRCX	1 Brushed	80	Yes

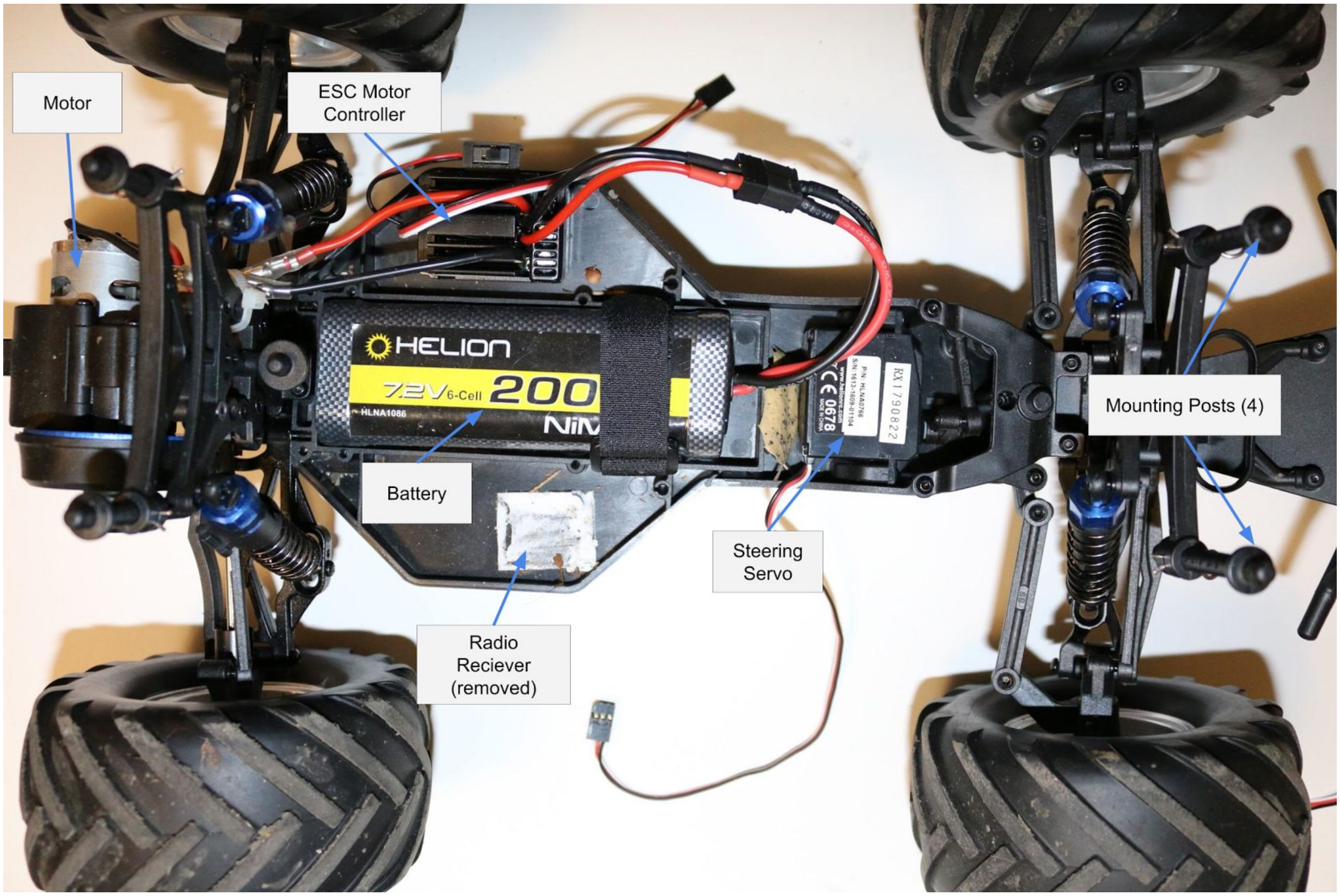
*while it is possible to have the Raspberry Pi provide power to the servo shield, you should never set up the shield this way. Steering servos pull too much power and create too much noise for the Raspberry Pi

Differential Steering

Scale	**Model**	**Make**	**Motors**	$	
1/18?	2WD Robot Kit	CanaKit	2 DC	13	

Tips for picking an RC Vehicle

- ESC and Servo should use a three wire connector.
- Brushed motors are easier to use because they do not need calibration.
- An ESC with a BEC or UBEC makes integration much simpler. A BEC provides 0 and 5 volts from the ESC through the 3 wire cable which can be used to power the servo shield. This is important because the servo shield does not get power from the Raspberry Pi by default*


Motor
ESC Motor Controller
Battery
HELION
7.2V 6-Cell 200
HLNA1086
NiM
RX1790822
Steering Servo
Mounting Posts (4)
Radio Reciever (removed)

Credits