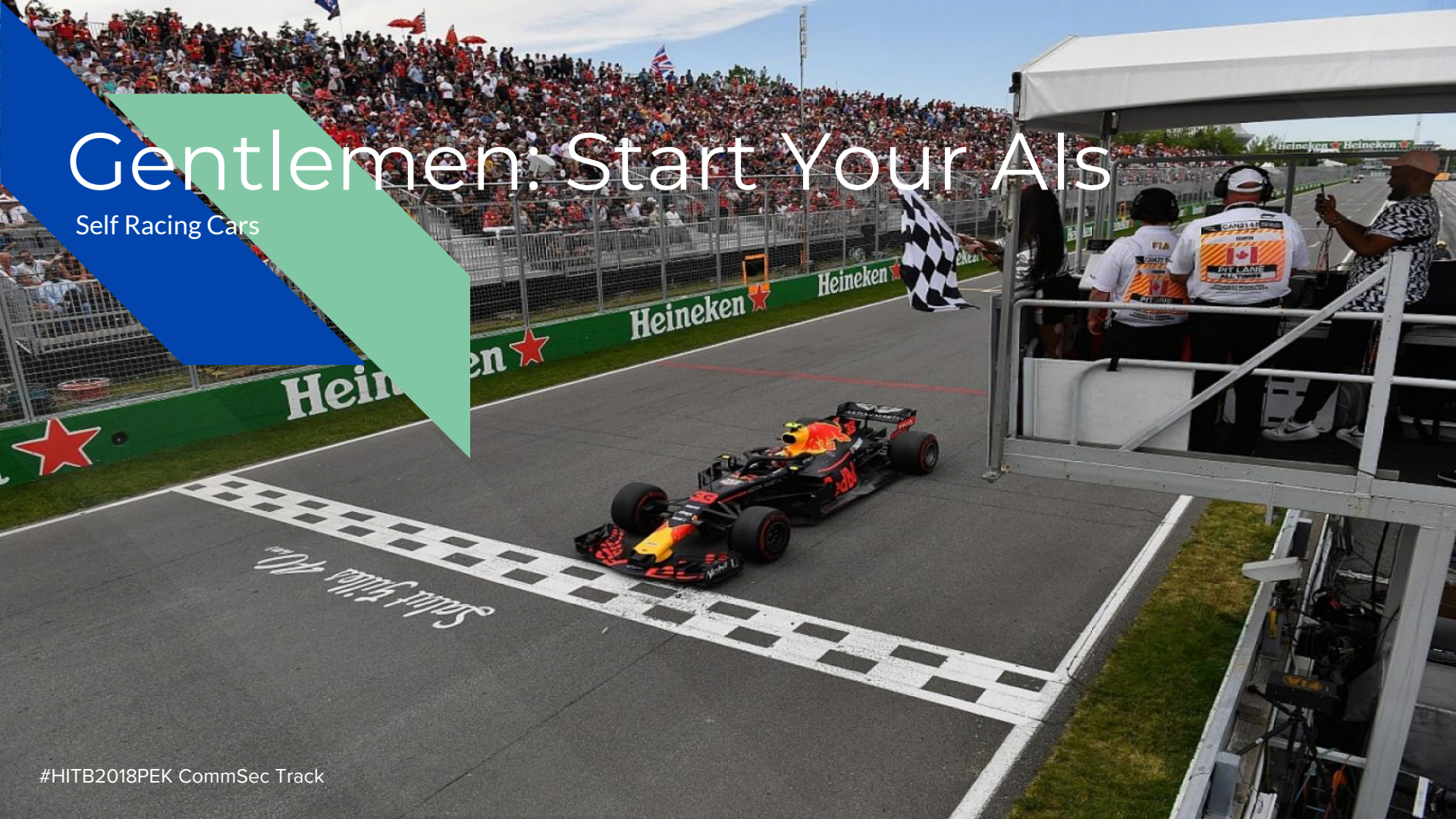


Gentlemen: Start Your AIs

Self Racing Cars

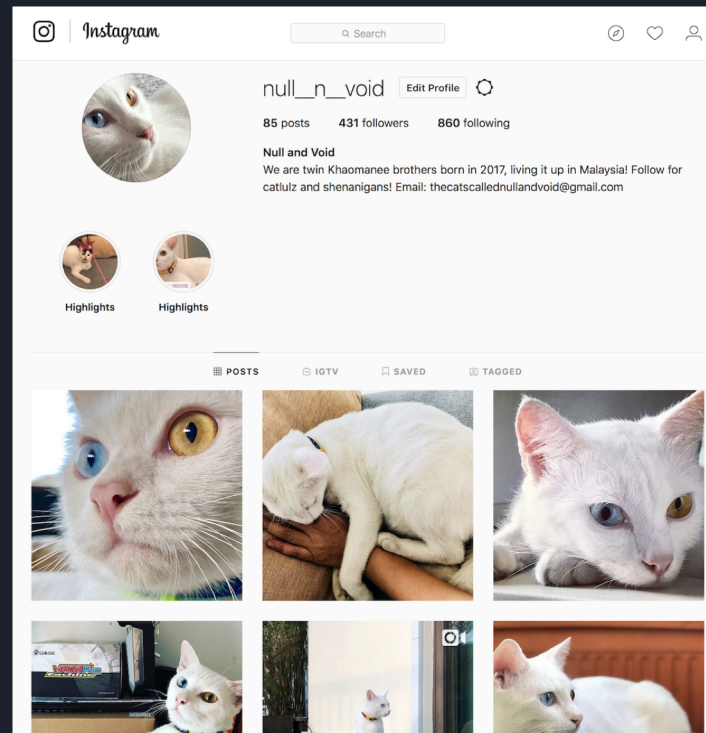


#whoami

- Founder / CEO @ Hack In The Box
- @L33tdawg on Twitter

#more

- Wrote code a long time ago
- Founder of Tumpang.la
- AI / ML Enthusiast
- I live in Malaysia with 2 'famous' cats



#State of the AI(rt)



FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

DeepRCar: An Autonomous Car Model

Bc. David Ungurean

Department of Applied Mathematics
Supervisor: Ing. Zdeněk Buk, Ph.D.

May 9, 2018

[https://dSPACE.cvut.cz/bitstream/handle/10467/76316/F8-
DP-2018-Ungurean-David-thesis.pdf](https://dSPACE.cvut.cz/bitstream/handle/10467/76316/F8-
DP-2018-Ungurean-David-thesis.pdf)

DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car

Michael G. Bechtel¹, Elise McElhiney¹, Minje Kim*, Heechul Yun¹

¹ University of Kansas, USA. {mbechtel, elisemc, heechul.yun}@ku.edu

* Indiana University, USA. minje@indiana.edu

Abstract—We present DeepPicar, a low-cost deep neural network based autonomous car platform. DeepPicar is a small scale replication of a real self-driving car called DAVE-2 by NVIDIA. DAVE-2 uses a deep convolutional neural network (CNN), which takes images from a front-facing camera as input and produces car steering angles as output. DeepPicar uses the same network architecture—9 layers, 27 million connections and 250K parameters—and can drive itself in real-time using a web camera and a Raspberry Pi 3 quad-core platform. Using DeepPicar, we analyze the Pi 3's computing capabilities to support end-to-end deep learning based real-time control of autonomous vehicles. We also systematically compare other contemporary embedded computing platforms using the DeepPicar's CNN-based real-time control workload.

We find that all tested platforms, including the Pi 3, are capable of supporting the CNN-based real-time control, from 20 Hz up to 100 Hz, depending on hardware platform. However, we find that shared resource contention remains an important issue that must be considered in applying CNN models on shared memory based embedded computing platforms; we observe up to 11.6X execution time increase in the CNN based control loop due to shared resource contention. To protect the CNN workload, we also evaluate state-of-the-art cache partitioning and memory bandwidth throttling techniques on the Pi 3. We find that cache partitioning is ineffective, while memory bandwidth throttling is an effective solution.

Keywords—Real-time, Autonomous car, Convolutional neural network, Case study

I. INTRODUCTION

Autonomous cars have been a topic of increasing interest in recent years as many companies are actively developing related hardware and software technologies toward fully autonomous driving capability with no human intervention. Deep neural networks (DNNs) have been successfully applied in various perception and control tasks in recent years. They are important workloads for autonomous vehicles as well. For example, Tesla Model S was known to use a specialized chip (MobileEye EyeQ), which used a vision-based real-time

task may be directly linked to the safety of the vehicle. This requires a high computing capacity as well as the means to guaranteeing the timings. On the other hand, the computing hardware platform must also satisfy cost, size, weight, and power constraints, which require a highly efficient computing platform. These two conflicting requirements complicate the platform selection process as observed in [25].

To understand what kind of computing hardware is needed for AI workloads, we need a testbed and realistic workloads. While using a real car-based testbed would be most ideal, it is not only highly expensive, but also poses serious safety concerns that hinder development and exploration. Therefore, there is a need for safer and less costly testbeds.

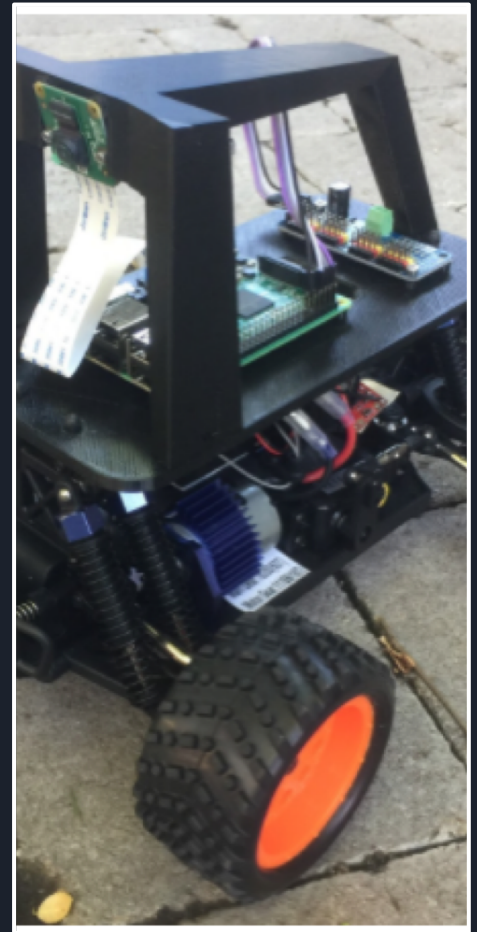
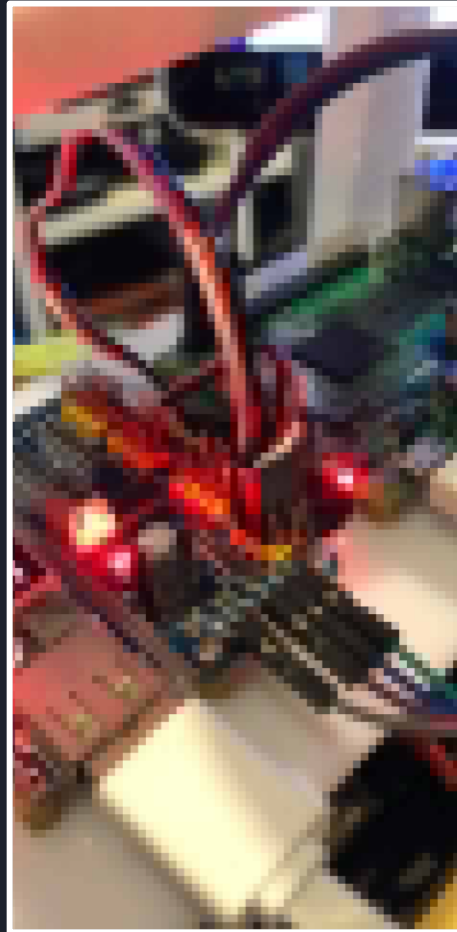
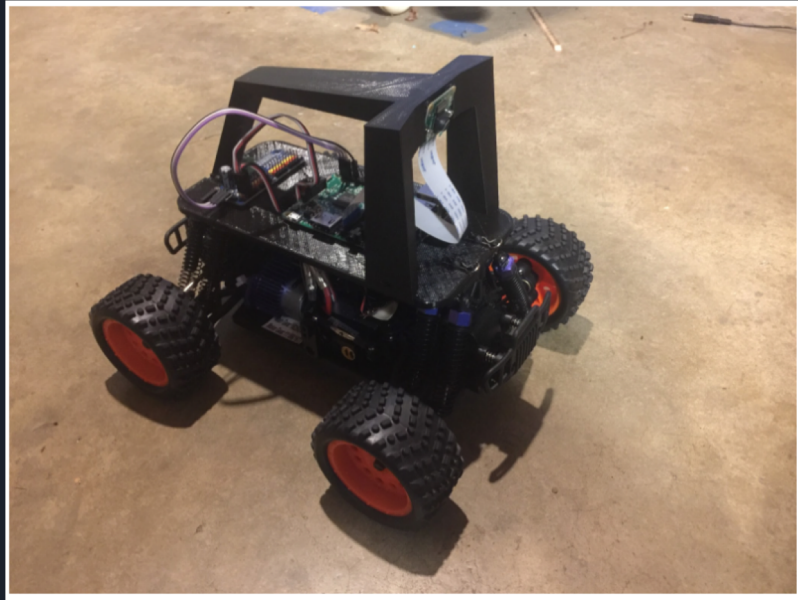
In this paper, we present DeepPicar, a low-cost autonomous car testbed for research. From a hardware perspective, DeepPicar is comprised of a Raspberry Pi 3 Model B quad-core computer, a web camera and a small RC car, all of which are affordable components (less than \$100 in total). The DeepPicar, however, employs a state-of-the-art AI technology, namely end-to-end deep learning based real-time control, which utilizes a deep convolutional neural network (CNN). The CNN receives an image frame from a single forward looking camera as input and generates a predicted steering angle value as output at each control period in real-time. The CNN has 9 layers, about 27 million connections and 250 thousand parameters (weights). DeepPicar's CNN architecture is identical to that of NVIDIA's real-sized self-driving car, called DAVE-2 [6], which drove on public roads without human driver's intervention while only using the CNN.

Using DeepPicar, we systematically analyze its real-time capabilities in the context of end-to-end deep-learning based real-time control, especially on real-time *inferencing* of the CNN. We also evaluate other, more powerful, embedded computing platforms to better understand achievable real-time performance of DeepPicar's CNN based control system

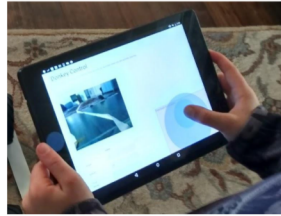
arXiv:1712.08644v4 [cs.OH] 30 Jul 2018

<https://arxiv.org/pdf/1712.08644.pdf>

#Donkey What?



#How Does It Work?



Steering (-1 to 1)
Throttle (-1 to 1)
Drive Mode (manual / auto)



Image (120 x 160)



Drive
Perfectly.

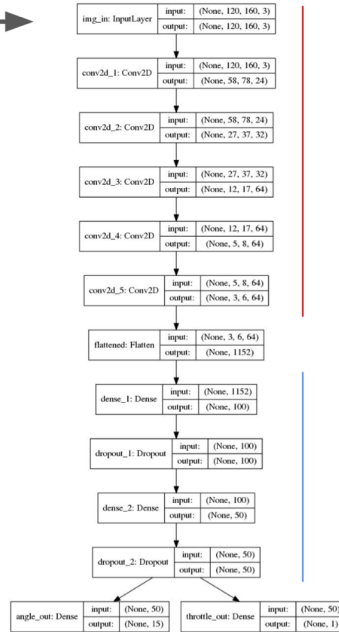
#How Does It Work?

Image Array



120 pixels high
160 pixels wide
3 RGB channels

Neural network.

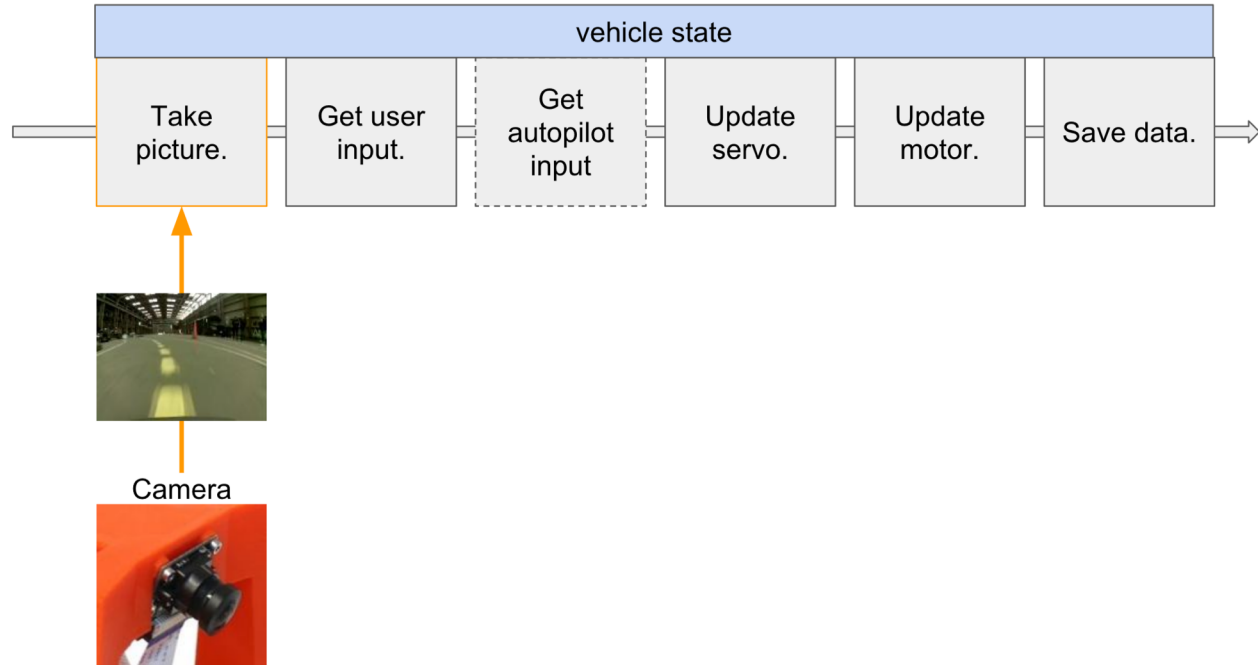


Convolution layers

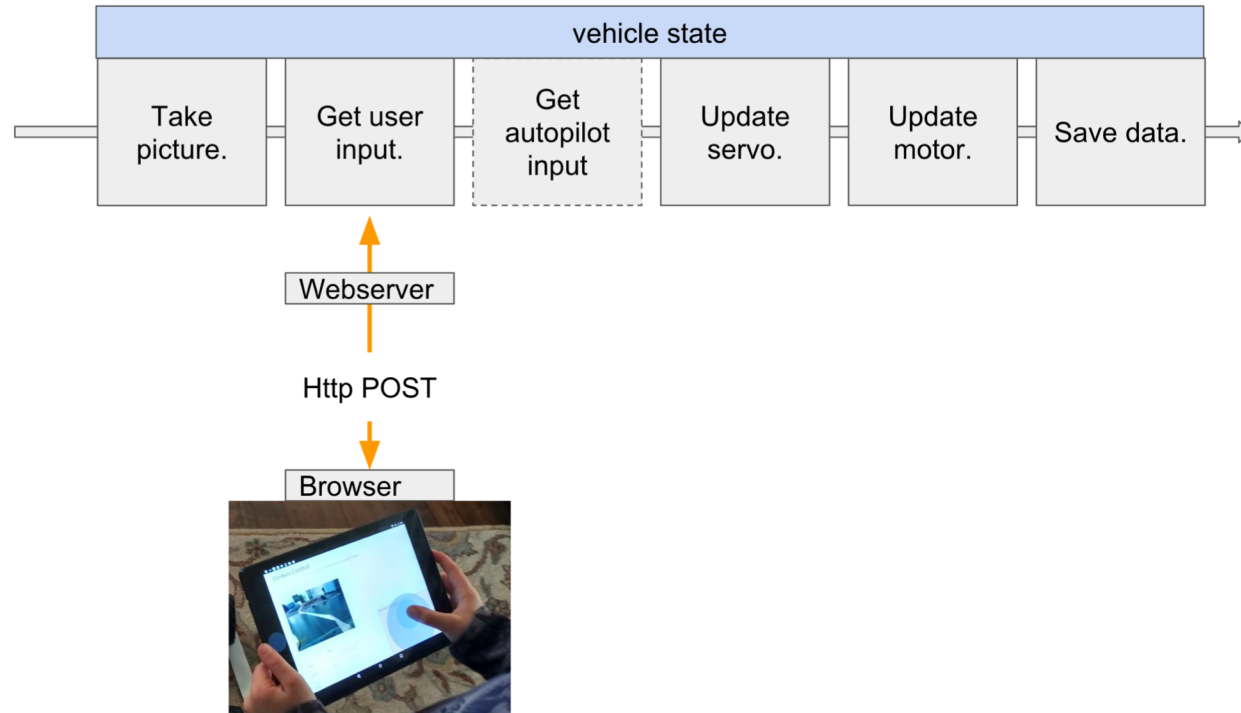
Fully connected layers

Steering + Throttle

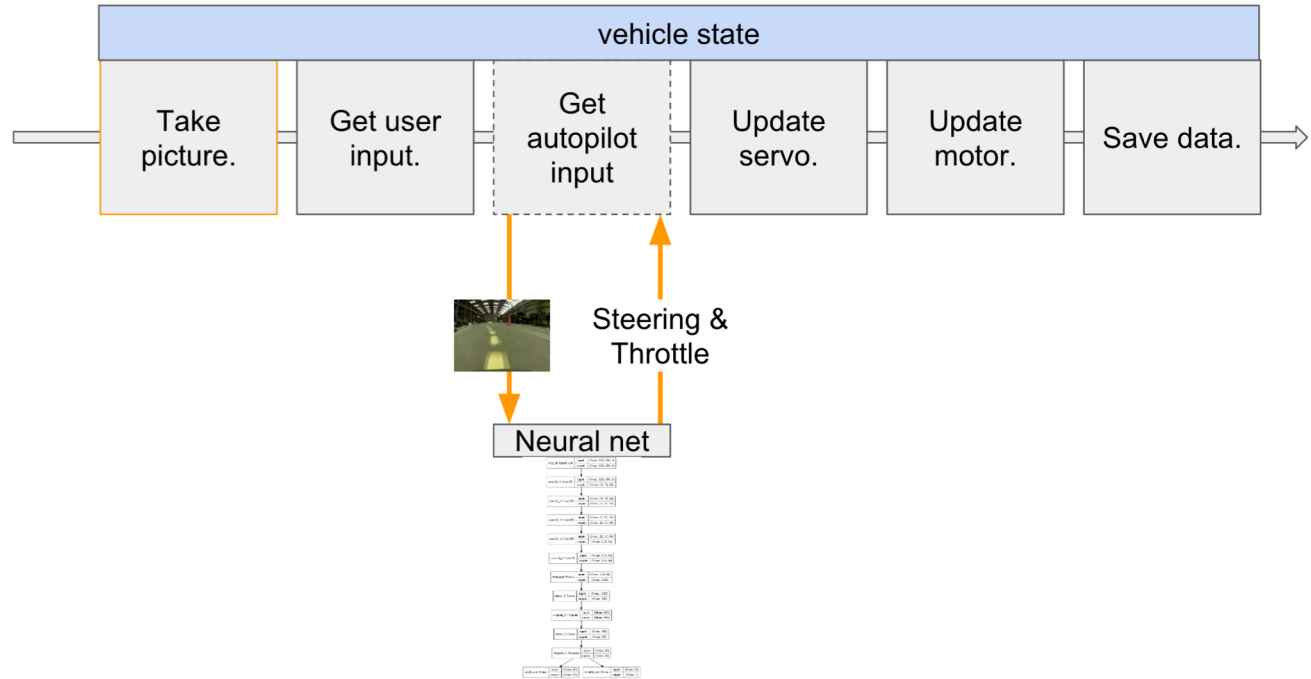
#How Does It Work?



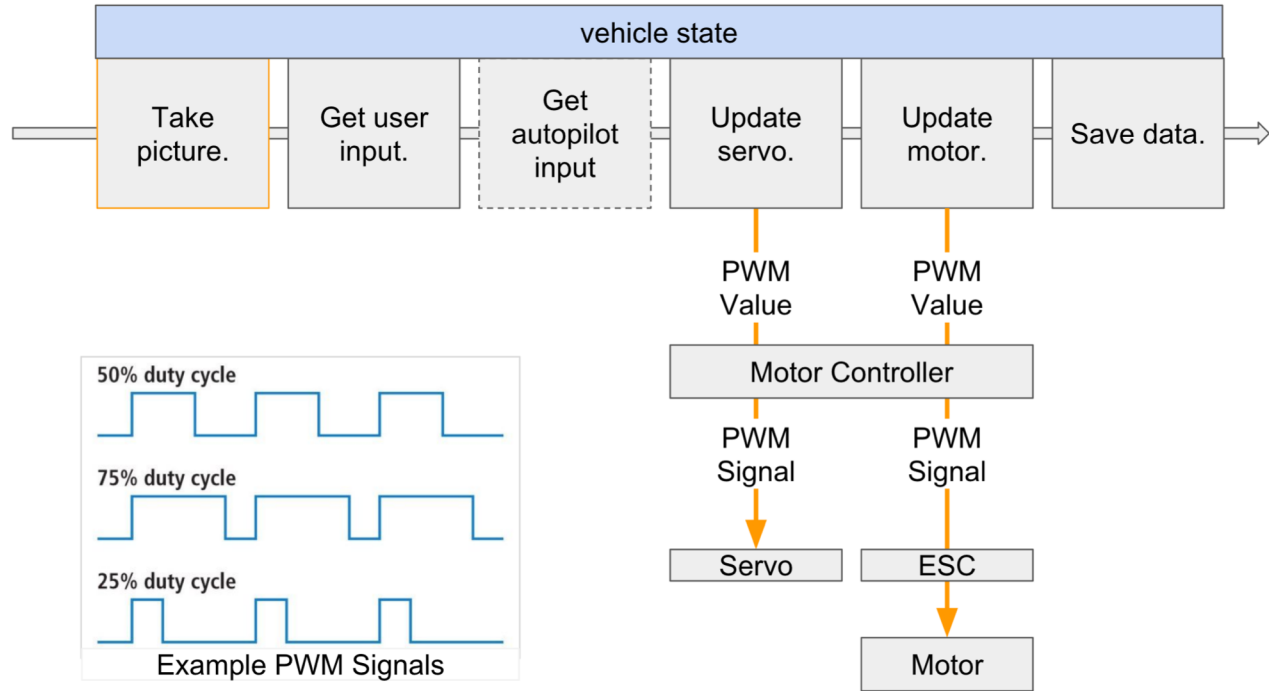
#How Does It Work?



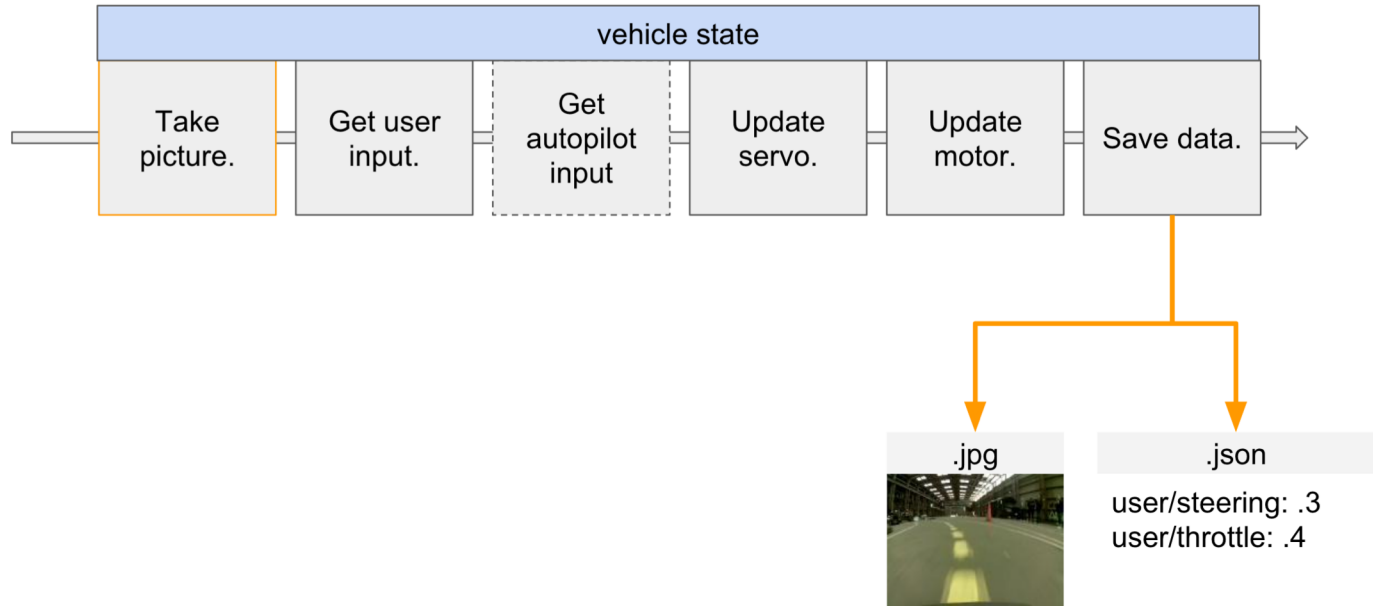
#How Does It Work?



#How Does It Work?



#How Does It Work?



#How Does It Work?

Keras / Tensorflow Autopilots

```
img_in = Input(shape=(120, 160, 3), name='img_in')
x = img_in
x = Convolution2D(24, (5,5), strides=(2,2), activation='relu')(x)
x = Convolution2D(32, (5,5), strides=(2,2), activation='relu')(x)
x = Convolution2D(64, (5,5), strides=(2,2), activation='relu')(x)
x = Convolution2D(64, (3,3), strides=(2,2), activation='relu')(x)
x = Convolution2D(64, (3,3), strides=(1,1), activation='relu')(x)

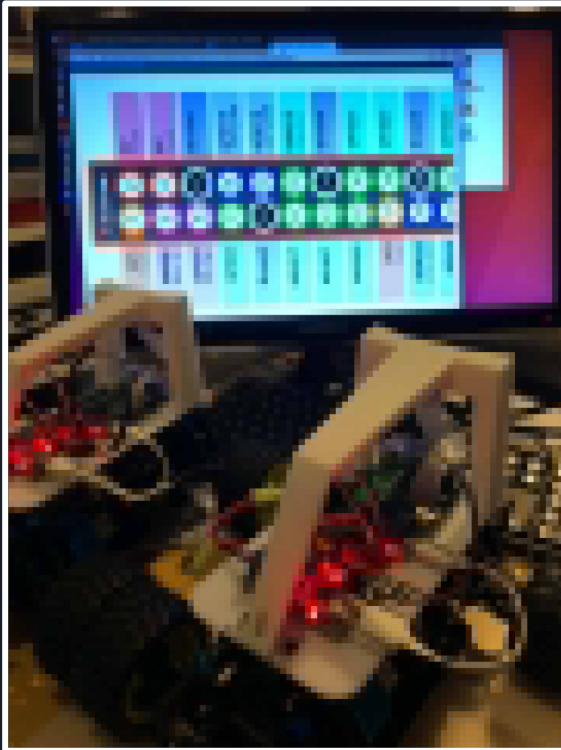
x = Flatten(name='flattened')(x)

x = Dense(100, activation='relu')(x)
x = Dropout(.1)(x)
x = Dense(50, activation='relu')(x)
x = Dropout(.1)(x)

#categorical output of the angle
angle_out = Dense(15, activation='softmax', name='angle_out')(x)

#continous output of throttle
throttle_out = Dense(1, activation='relu', name='throttle_out')(x)
```

#Hardware



Magnet Car (Red / Blue) or alternative

\$92

M2x6 screws (4)

\$6

M2.5x12 screws (8)

\$5

M2.5 nuts (8)

\$6

M2.5 washers (8)

\$7

USB Battery with microUSB cable

\$17

Raspberry Pi 3

\$38

MicroSD Card

\$20

Wide Angle Raspberry Pi Camera

\$25

Female to Female Jumper Wire

\$7

Servo Driver PCA 9685

\$12

3D Printed roll cage and top plate.

\$45

TOTAL:

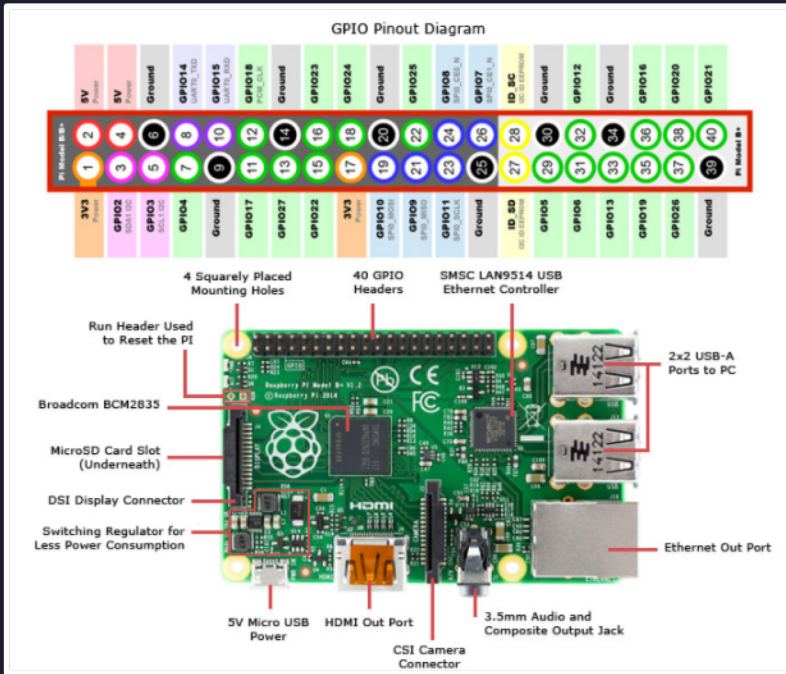
#Software

Raspberry Pi

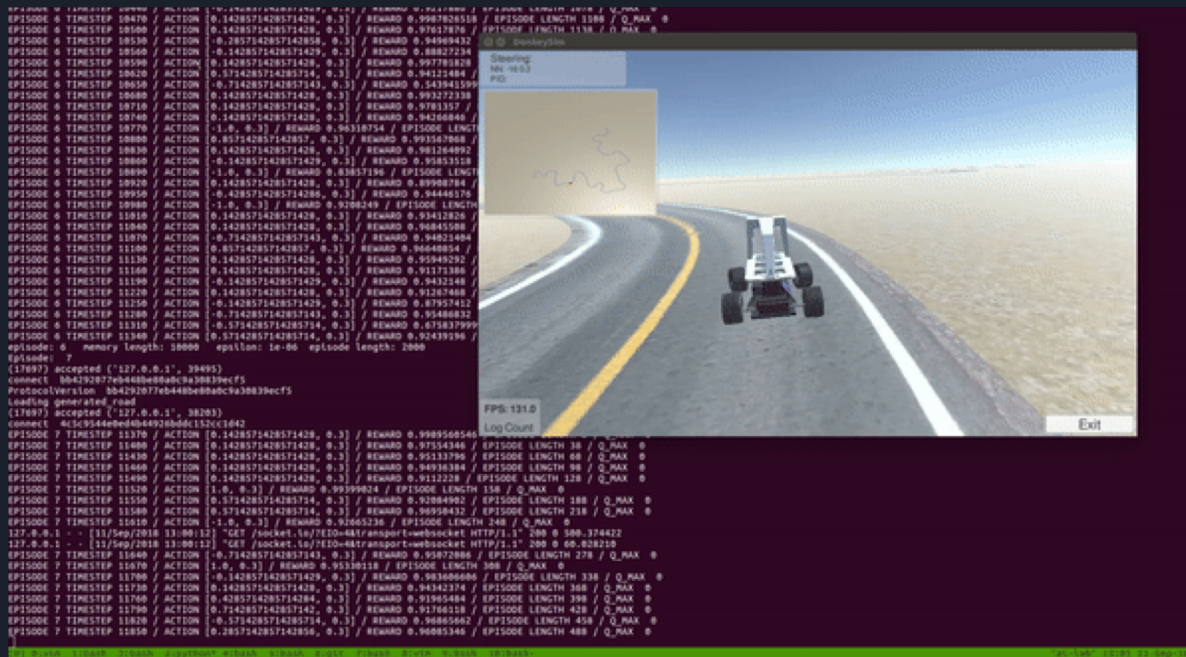
1. Download prebuilt zipped disk image (1.1GB)
2. Flash it
3. `git clone https://github.com/wroscoe/donkey`
4. `pip install -e .[pi]`

Linux / Host Machine

1. `sudo apt-get install virtualenv build-essential python3-dev gfortran libhdf5-dev`
2. `virtualenv env -p python3`
3. `source env/bin/activate`
4. `pip install tensorflow`
5. `git clone https://github.com/wroscoe/donkey`
6. `pip install -e .`



#Unity Simulator



The image shows a screenshot of the Unity Simulator interface. The main window displays a 3D environment with a white car on a road. The car is positioned on a road with yellow and white lane markings, driving towards a large white wall. The background is a clear blue sky and a flat, light-colored ground. In the bottom right corner of the 3D view, there is a small 'Exit' button. The top left corner of the 3D view shows 'FPS: 131.0' and 'Log Count'. The bottom of the screen is occupied by a terminal window displaying a log of actions and rewards. The log entries are as follows:

```
EPISODE 0 TIMESTEP 10470 / ACTION [0.1428571428571428, 0.3] / REWARD 0.9087826518 / EPISODE LENGTH 1088 / Q_MAX 0
EPISODE 0 TIMESTEP 10480 / ACTION [0.1428571428571428, 0.3] / REWARD 0.90378936 / EPISODE LENGTH 1118 / Q_MAX 0
EPISODE 0 TIMESTEP 10530 / ACTION [-0.2857142857142858, 0.3] / REWARD 0.94968432
EPISODE 0 TIMESTEP 10560 / ACTION [-0.1428571428571429, 0.3] / REWARD 0.88827234
EPISODE 0 TIMESTEP 10590 / ACTION [0.1428571428571428, 0.3] / REWARD 0.997781838
EPISODE 0 TIMESTEP 10620 / ACTION [0.5714285714285714, 0.3] / REWARD 0.94121484
EPISODE 0 TIMESTEP 10650 / ACTION [-0.7142857142857143, 0.3] / REWARD 0.543941599
EPISODE 0 TIMESTEP 10700 / ACTION [0.1428571428571428, 0.3] / REWARD 0.99232238
EPISODE 0 TIMESTEP 10710 / ACTION [0.1428571428571428, 0.3] / REWARD 0.97811357
EPISODE 0 TIMESTEP 10740 / ACTION [0.1428571428571428, 0.3] / REWARD 0.94268846
EPISODE 0 TIMESTEP 10770 / ACTION [-1.0, 0.3] / REWARD 0.9033751 / EPISODE LENGTH
EPISODE 0 TIMESTEP 10800 / ACTION [0.857142857142857, 0.3] / REWARD 0.99353968
EPISODE 0 TIMESTEP 10830 / ACTION [0.1428571428571428, 0.3] / REWARD 0.981264992
EPISODE 0 TIMESTEP 10860 / ACTION [-0.1428571428571429, 0.3] / REWARD 0.93833118
EPISODE 0 TIMESTEP 10890 / ACTION [-1.0, 0.3] / REWARD 0.83857196 / EPISODE LENGTH
EPISODE 0 TIMESTEP 10920 / ACTION [0.1428571428571428, 0.3] / REWARD 0.89988784
EPISODE 0 TIMESTEP 10950 / ACTION [-0.4285714285714286, 0.3] / REWARD 0.94448176
EPISODE 0 TIMESTEP 10980 / ACTION [-1.0, 0.3] / REWARD 0.9488249 / EPISODE LENGTH
EPISODE 0 TIMESTEP 11010 / ACTION [0.1428571428571428, 0.3] / REWARD 0.93432826
EPISODE 0 TIMESTEP 11040 / ACTION [0.1428571428571428, 0.3] / REWARD 0.96845588
EPISODE 0 TIMESTEP 11070 / ACTION [-0.7142857142857143, 0.3] / REWARD 0.94821464
EPISODE 0 TIMESTEP 11100 / ACTION [0.857142857142857, 0.3] / REWARD 0.94648854
EPISODE 0 TIMESTEP 11130 / ACTION [0.1428571428571428, 0.3] / REWARD 0.95949292
EPISODE 0 TIMESTEP 11160 / ACTION [0.1428571428571428, 0.3] / REWARD 0.91173386
EPISODE 0 TIMESTEP 11190 / ACTION [-0.1428571428571429, 0.3] / REWARD 0.9432346
EPISODE 0 TIMESTEP 11220 / ACTION [0.1428571428571428, 0.3] / REWARD 0.91267468
EPISODE 0 TIMESTEP 11250 / ACTION [-0.1428571428571429, 0.3] / REWARD 0.87957412
EPISODE 0 TIMESTEP 11280 / ACTION [-0.7142857142857143, 0.3] / REWARD 0.95486432
EPISODE 0 TIMESTEP 11310 / ACTION [-0.5714285714285714, 0.3] / REWARD 0.67837999
EPISODE 0 TIMESTEP 11340 / ACTION [0.5714285714285714, 0.3] / REWARD 0.97439196
episode: 0 memory length: 38864 epsilon: 1e-06 episode length: 2866
Episode: 7
[11697] accepted ('127.0.0.1', 38495)
connect 4b4292077eb448be8a0ca3a839ecf5
ProcessVersion bb4292077eb448be8a0ca3a839ecf5
Loading generated_map
[11697] accepted ('127.0.0.1', 38203)
connect 41c5c944ee04b4920b6d152c1042
EPISODE 7 TIMESTEP 11370 / ACTION [0.1428571428571428, 0.3] / REWARD 0.99856856 / EPISODE LENGTH 98 / Q_MAX 0
EPISODE 7 TIMESTEP 11400 / ACTION [0.1428571428571428, 0.3] / REWARD 0.9754346 / EPISODE LENGTH 98 / Q_MAX 0
EPISODE 7 TIMESTEP 11430 / ACTION [0.1428571428571428, 0.3] / REWARD 0.9513796 / EPISODE LENGTH 98 / Q_MAX 0
EPISODE 7 TIMESTEP 11460 / ACTION [0.1428571428571428, 0.3] / REWARD 0.94916384 / EPISODE LENGTH 98 / Q_MAX 0
EPISODE 7 TIMESTEP 11490 / ACTION [0.1428571428571428, 0.3] / REWARD 0.911228 / EPISODE LENGTH 128 / Q_MAX 0
EPISODE 7 TIMESTEP 11520 / ACTION [1.0, 0.3] / REWARD 0.99399624 / EPISODE LENGTH 158 / Q_MAX 0
EPISODE 7 TIMESTEP 11550 / ACTION [0.5714285714285714, 0.3] / REWARD 0.9284992 / EPISODE LENGTH 188 / Q_MAX 0
EPISODE 7 TIMESTEP 11580 / ACTION [0.5714285714285714, 0.3] / REWARD 0.96928432 / EPISODE LENGTH 218 / Q_MAX 0
EPISODE 7 TIMESTEP 11610 / ACTION [1.0, 0.3] / REWARD 0.953346 / EPISODE LENGTH 248 / Q_MAX 0
EPISODE 7 TIMESTEP 11670 / ACTION [1.0, 0.3] / REWARD 0.9538118 / EPISODE LENGTH 308 / Q_MAX 0
EPISODE 7 TIMESTEP 11700 / ACTION [-0.1428571428571429, 0.3] / REWARD 0.98368688 / EPISODE LENGTH 338 / Q_MAX 0
EPISODE 7 TIMESTEP 11730 / ACTION [0.1428571428571428, 0.3] / REWARD 0.94302374 / EPISODE LENGTH 368 / Q_MAX 0
EPISODE 7 TIMESTEP 11760 / ACTION [0.4285714285714284, 0.3] / REWARD 0.9196484 / EPISODE LENGTH 398 / Q_MAX 0
EPISODE 7 TIMESTEP 11790 / ACTION [0.7142857142857142, 0.3] / REWARD 0.91766518 / EPISODE LENGTH 428 / Q_MAX 0
EPISODE 7 TIMESTEP 11840 / ACTION [-0.5714285714285714, 0.3] / REWARD 0.9688896 / EPISODE LENGTH 458 / Q_MAX 0
EPISODE 7 TIMESTEP 11850 / ACTION [0.2857142857142858, 0.3] / REWARD 0.9688346 / EPISODE LENGTH 488 / Q_MAX 0
```

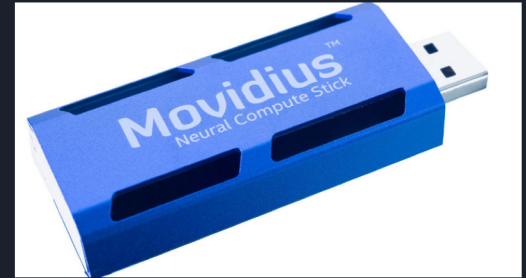
Download: <https://docs.donkeycar.com/guide/simulator/>

#HITB2019AMS - Gentlemen Start Your AIs

May 6th - 10th 2019

Hackerspaces / Individuals

- 1/16 scale Donkeycar - All hardware provided
- 2 batteries per team
- 3 days free practice (6 / 7 / 8th May)
- Qualifying - 9th May (2 sessions)
- Top 10 teams move to race day (10th May)



Finalists will be given Intel Movidius Neural Compute Stick - Go harder, go faster, be better!

Professional Teams

- Bring your own car - 1/10 scale
- No limit on hardware sensors
- Limited to 3 batteries per team
- 3 days free practice (6 / 7 / 8th May)
- Qualifying - 9th May (2 sessions)
- Race Day - 15 min race time



Get building, get racing, and see you in Amsterdam!

Questions?

`l33tdawg@hitb.org // @L33tdawg`