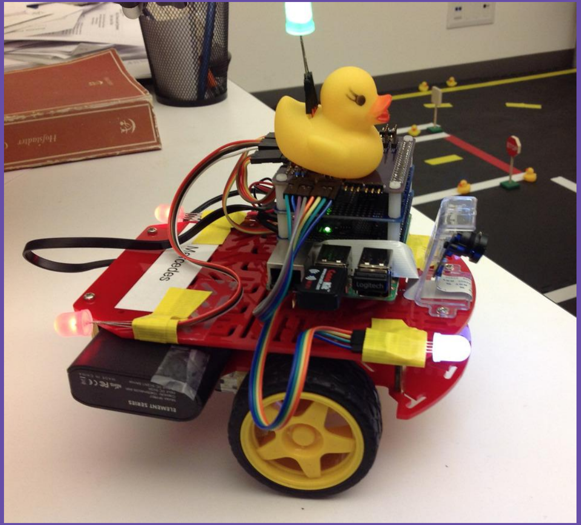
Duckietown

•••

Independent study led by Kirill Krinkin at JetBrains Research

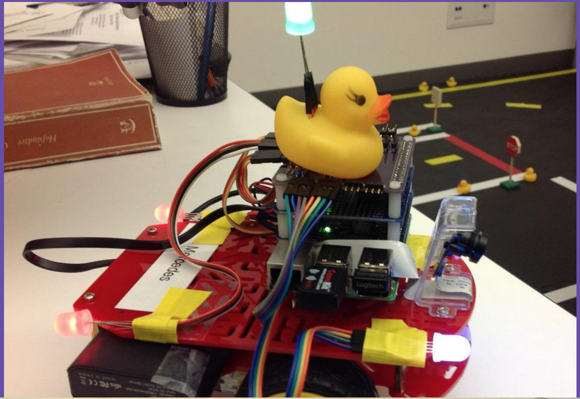
What is Duckietown?

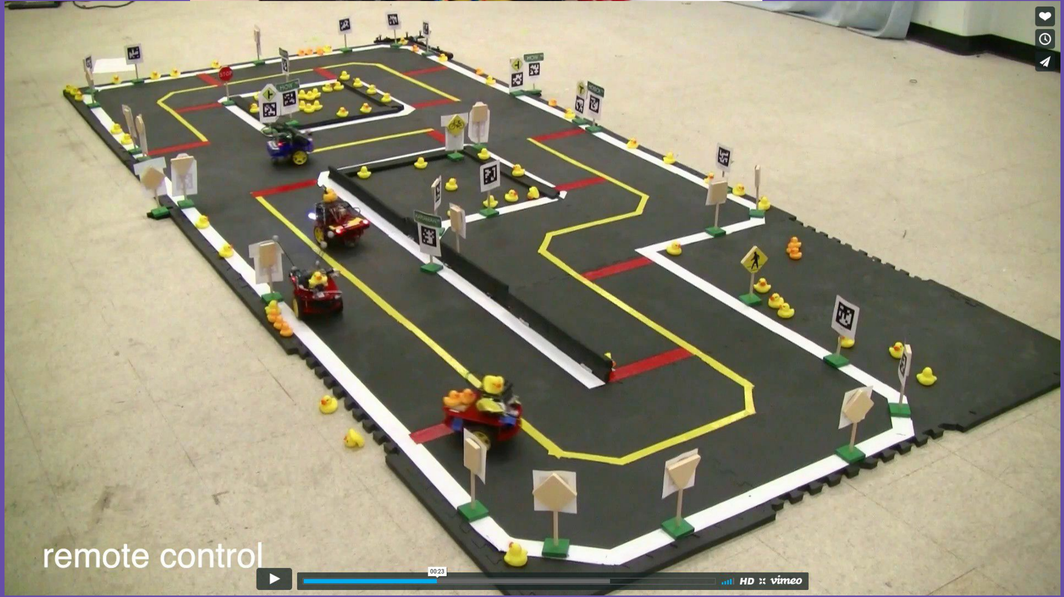
- Duckiebot



What is Duckietown?

- Duckiebot
- Duckietown (test platform)





Targets of Duckietown

- Globally: solution of the task’s robot’s autonomy in unknown or changing environment

Duckietown: an Open, Inexpensive and Flexible Platform for Autonomy Education and Research

Liam Paull, Jacopo Tani, Heejin Ahn, Javier Alonso-Mora, Luca Carlone, Michal Cap, Yu Fan Chen, Changhyun Choi, Jeff Dusek, Yajun Fang, Daniel Hoehener, Shih-Yuan Liu, Michael Novitzky, Igor Franzoni Okuyama, Jason Pazis, Guy Rosman, Valerio Varricchio, Hsueh-Cheng Wang, Dmitry Yershov, Hang Zhao, Michael Benjamin, Christopher Carr, Maria Zuber, Sertac Karaman, Emilio Frazzoli, Domitilla Del Vecchio, Daniela Rus, Jonathan How, John Leonard, Andrea Censi

***Abstract*—Duckietown is an *open*, *inexpensive* and *flexible* platform for autonomy education and research. The platform comprises small autonomous vehicles (“Duckiebots”) built from off-the-shelf components, and cities (“Duckietowns”) complete with roads, signage, traffic lights, obstacles, and citizens (duckies) in need of transportation. The advantage of the Duckietown platform is the combination of its low cost with its large range of functionalities. A Duckiebot senses the world with only one monocular camera and performs all processing onboard with a Raspberri Pi 2, yet it is able to: follow lanes while avoiding obstacles, pedestrians (duckies) and other Duckiebots; localize within a global map; navigate a city; and coordinate with other Duckiebots to avoid collisions. Duckietown is a useful tool for education, since institutions can save money and time by not having to develop all of the supporting infrastructure and capabilities that are tangential to the desired course material. All materials are available as open source, and the hope is that others in the community will adopt the platform for education and research. Four other institutions have used Duckietown to date.**

Fig. 1. In Duckietown, inhabitants (duckies) are transported via an autonomous mobility service (Duckiebots). Duckietown is designed to be inexpensive and modular, yet still enable many of the research and educational opportunities of a full-scale self-driving car platform.

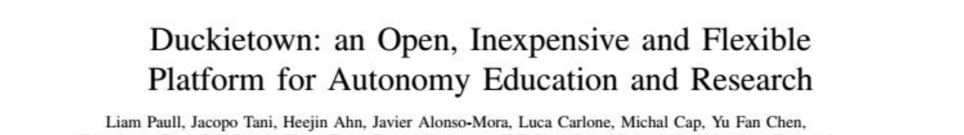
I. Introduction

Self-driving vehicles are poised to become one of the most pervasive and impactful applications of autonomy. However, difficult challenges still remain before their widespread deploy-

Duckiebots are capable of fairly complex single-robot and multi-robot behaviors. Duckiebots live in “Duckietowns”, colorful miniature environments that are assembled from modular tiles. The project’s whimsical and visually striking

Targets of Duckietown

- Globally: solution of the task's robot's autonomy in unknown or changing environment
- Targets of the last semester: creating test platform for experiments of autonomous mobile robots



Duckietown: an Open, Inexpensive and Flexible Platform for Autonomy Education and Research

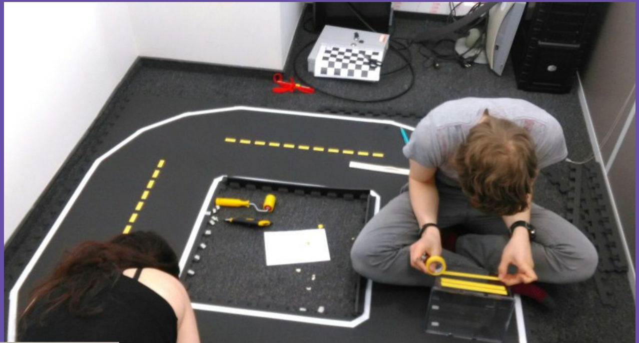
Liam Paull, Jacopo Tani, Heejin Ahn, Javier Alonso-Mora, Luca Carlone, Michal Cap, Yu Fan Chen,

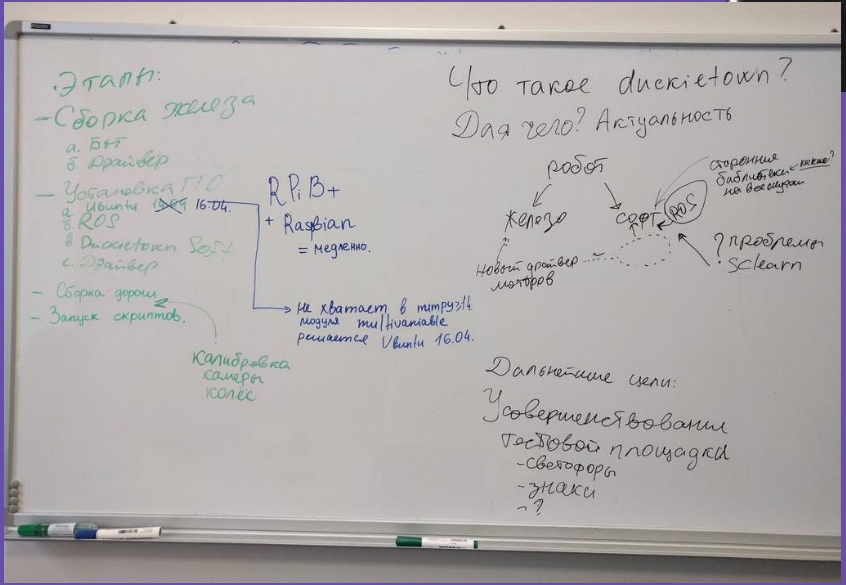
ted via an
gned to be
educational

obot and
etowns",
ed from
striking

Duckietown's team & location:

- Constantin Chayka
- Maksim Kuzmin
- Olesya Tishencko
- Anatoly Bendrikovsky





Этапы:

- Сборка железа
  - а. БНТ
  - б. Драйвер
- Установка ПО
  - а. Ubuntu ~~14.04~~ 16.04.
  - б. ROS
  - в. Duckietown Soft
  - г. Драйвер
- Сборка дороги
- Запуск скриптов.

Калибровка
камеры
колёс

RPi B+
+ Raspbian
= медленно.

Не хватает в питру≥14.
модуля multivariable
решается Ubuntu 16.04.

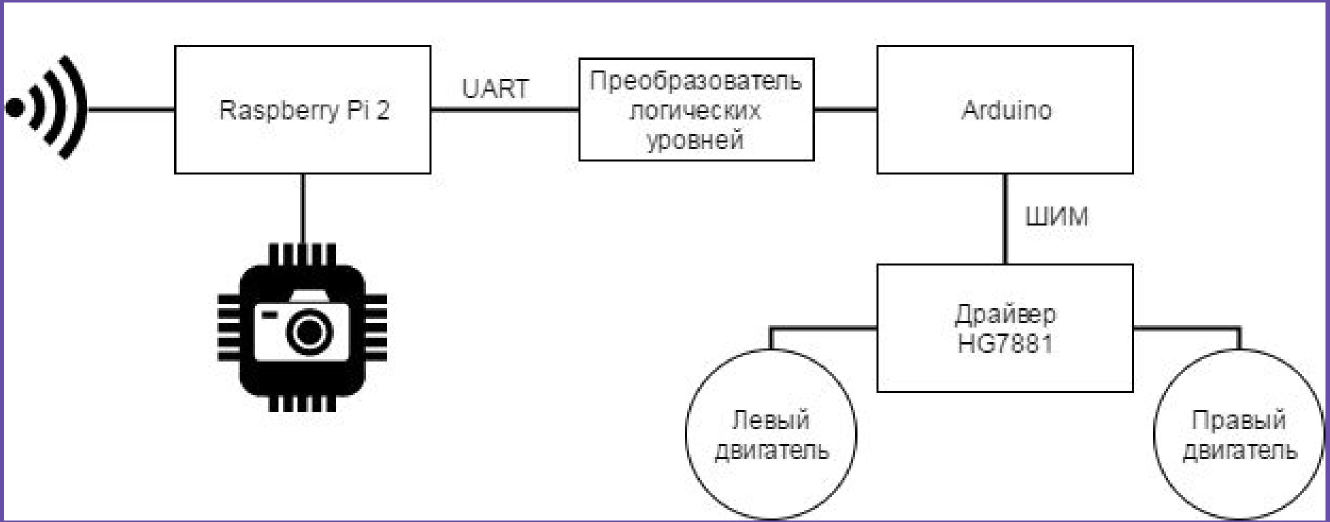
Что такое duckietown?
Для чего? Актуальность

робот
железо
софт
ROS
сторонние библиотеки на все случаи ← какие?
новый драйвер моторов
? проблемы
· sclearn

Дальнейшие цели:
Усовершенствовании
тестовой площадки
-светофоры
-знаки
-?

Saint-Petersburg,
Kantemirovskaya ul., 2
Tayms, off. 320

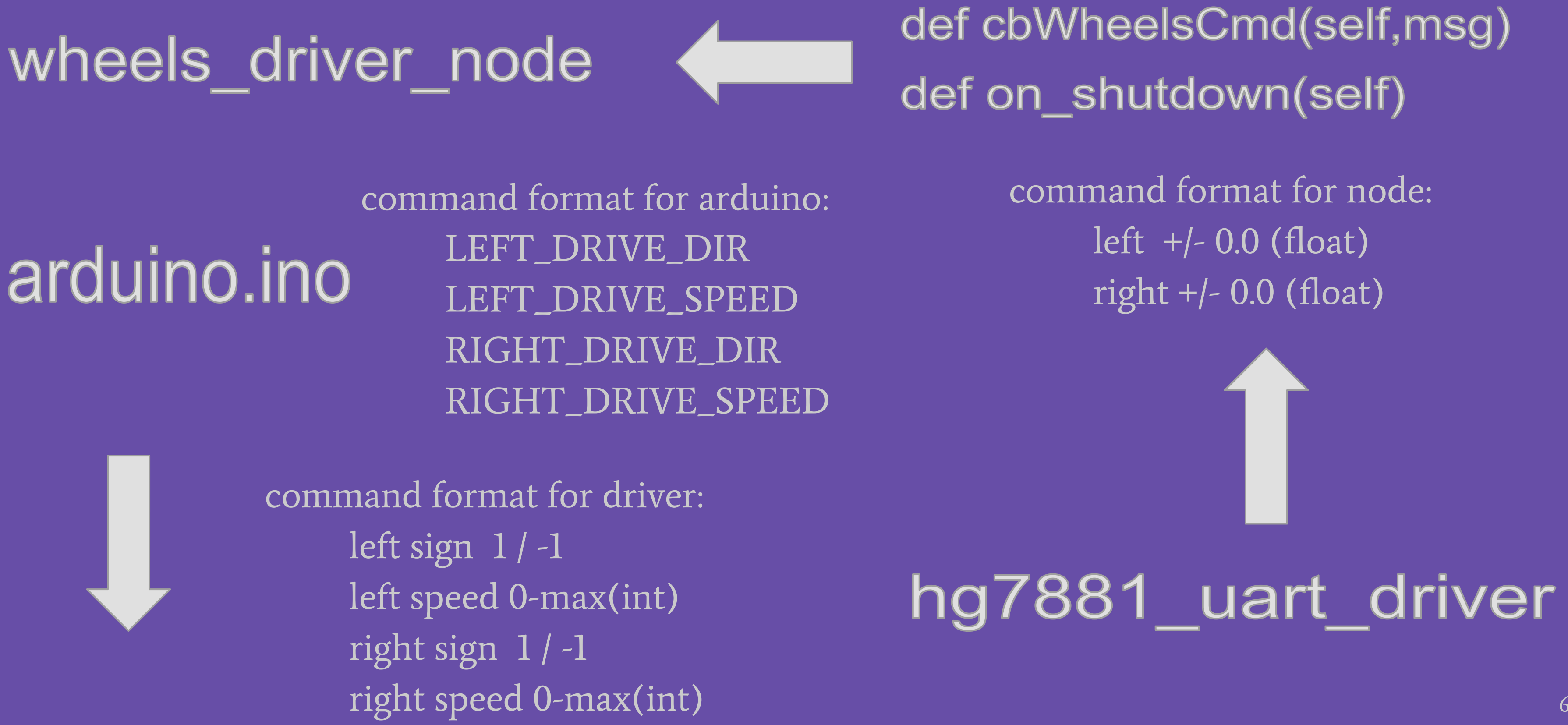
Duckiebot



Bot’s components:

- RispberryPi2
- RPi fisheye camera
- USB WiFi
- UART driver HG7881 + Ardiuno Uno
- Battery kit
- Chassis

Wheel’s driver



wheels_driver_node

def cbWheelsCmd(self,msg)

def on_shutdown(self)

arduino.ino

command format for arduino:

LEFT_DRIVE_DIR
LEFT_DRIVE_SPEED
RIGHT_DRIVE_DIR
RIGHT_DRIVE_SPEED

command format for node:

left +/- 0.0 (float)
right +/- 0.0 (float)

command format for driver:

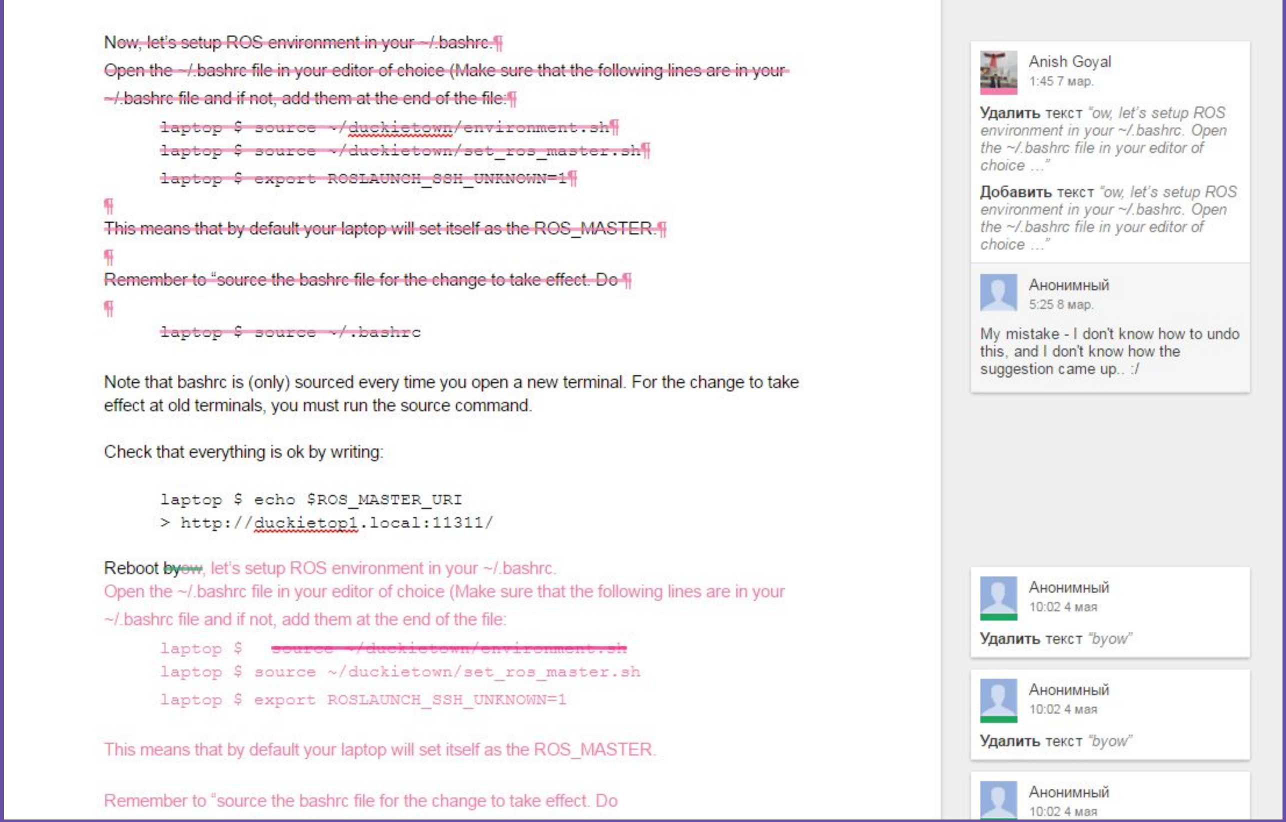
left sign 1 / -1
left speed 0-max(int)
right sign 1 / -1
right speed 0-max(int)

hg7881_uart_driver

Duckiebot

Software for robots:

- Ubuntu 16.04
- ROS Kinetic
- Duckietown software (by MIT)



N~~ow, let's setup ROS environment in your ~/.bashrc.~~¶

~~Open the ~/.bashrc file in your editor of choice (Make sure that the following lines are in your ~/.bashrc file and if not, add them at the end of the file:~~¶

~~`laptop $ source ~/duckietown/environment.sh`~~¶
~~`laptop $ source ~/duckietown/set_ros_master.sh`~~¶
~~`laptop $ export ROSLAUNCH_SSH_UNKNOWN=1`~~¶

¶

~~This means that by default your laptop will set itself as the ROS_MASTER.~~¶

¶

~~Remember to "source the bashrc file for the change to take effect. Do~~¶

¶

~~`laptop $ source ~/.bashrc`~~

Note that bashrc is (only) sourced every time you open a new terminal. For the change to take effect at old terminals, you must run the source command.

Check that everything is ok by writing:

```
laptop $ echo $ROS_MASTER_URI
> http://duckietop1.local:11311/
```

Reboot by~~ow~~, let's setup ROS environment in your ~/.bashrc.

Open the ~/.bashrc file in your editor of choice (Make sure that the following lines are in your ~/.bashrc file and if not, add them at the end of the file:

`laptop $` ~~`source ~/duckietown/environment.sh`~~
`laptop $ source ~/duckietown/set_ros_master.sh`
`laptop $ export ROSLAUNCH_SSH_UNKNOWN=1`

This means that by default your laptop will set itself as the ROS_MASTER.

Remember to "source the bashrc file for the change to take effect. Do

Anish Goyal
1:45 7 мар.

Удалить текст *"ow, let's setup ROS environment in your ~/.bashrc. Open the ~/.bashrc file in your editor of choice ..."*

Добавить текст *"ow, let's setup ROS environment in your ~/.bashrc. Open the ~/.bashrc file in your editor of choice ..."*

Анонимный
5:25 8 мар.

My mistake - I don't know how to undo this, and I don't know how the suggestion came up.. :/

Анонимный
10:02 4 мая

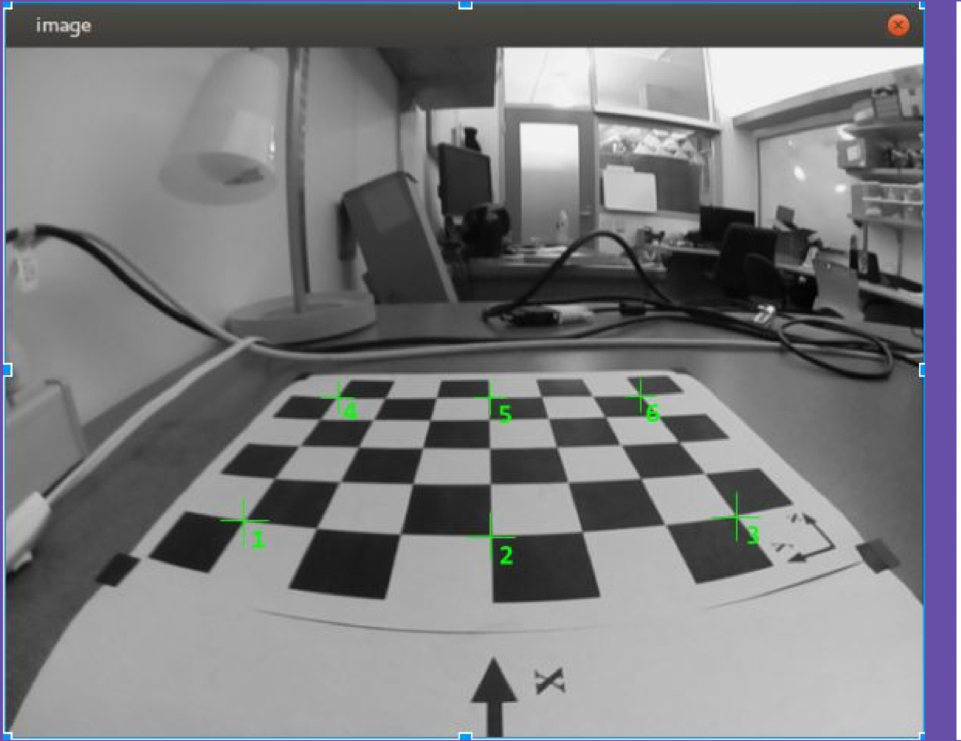
Удалить текст *"byow"*

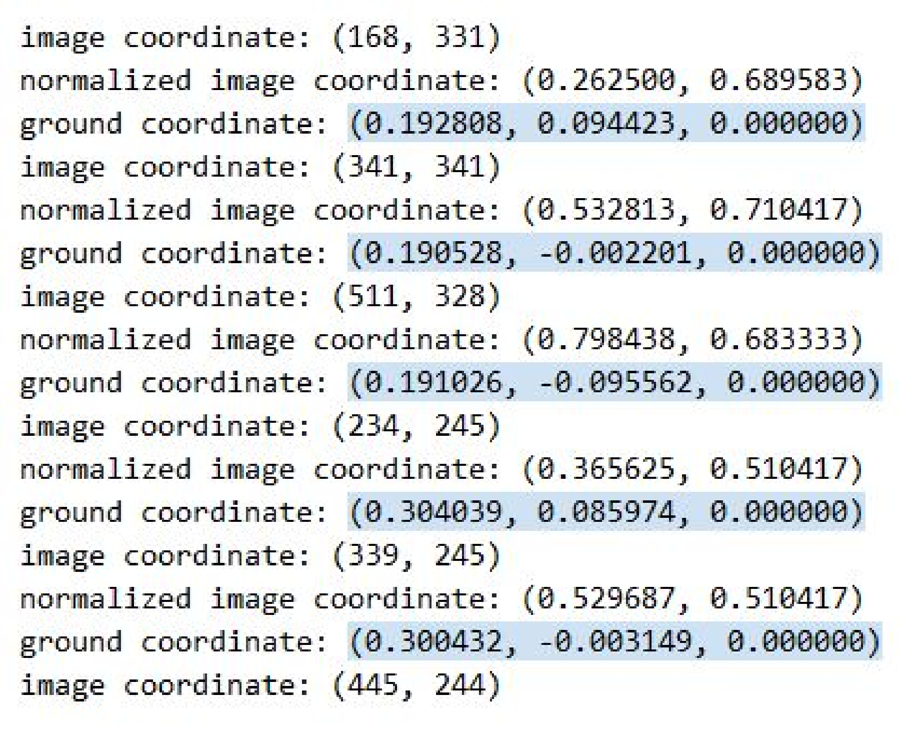
Анонимный
10:02 4 мая

Удалить текст *"byow"*

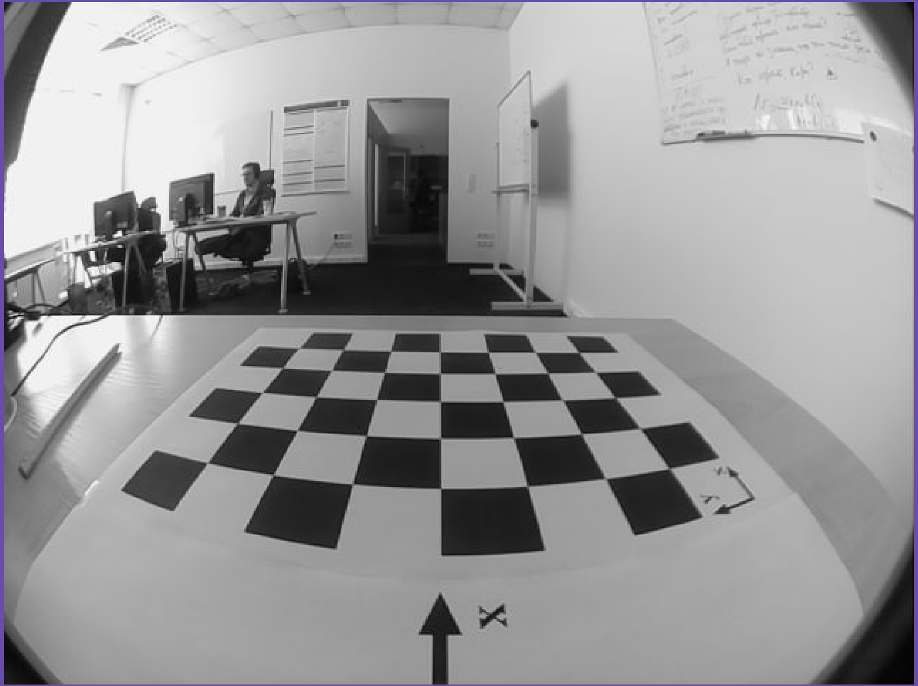
Анонимный
10:02 4 мая

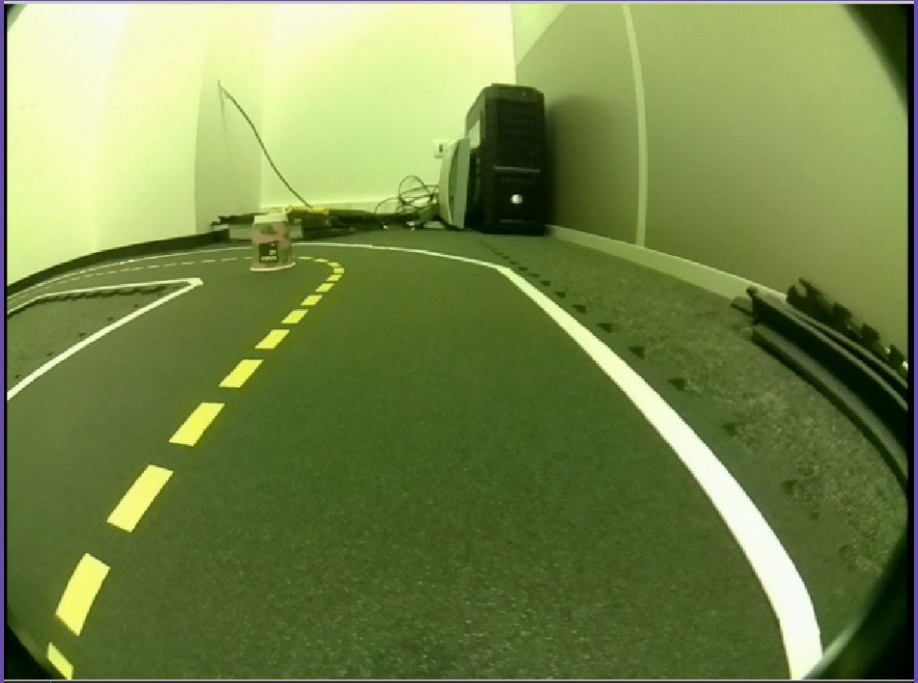
Camera calibration



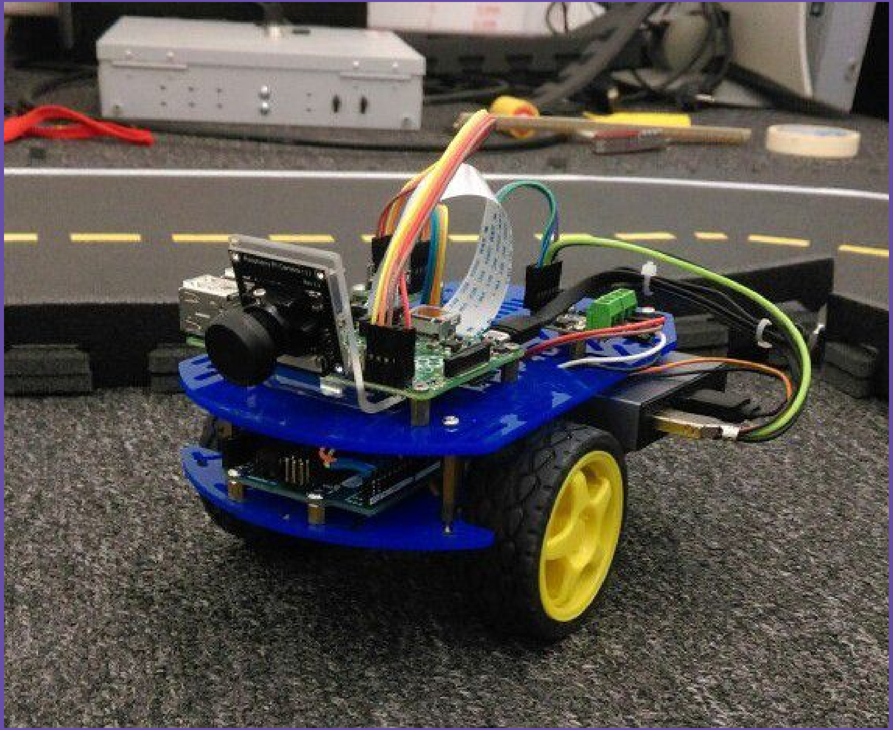

```
image coordinate: (168, 331)
normalized image coordinate: (0.262500, 0.689583)
ground coordinate: (0.192808, 0.094423, 0.000000)
image coordinate: (341, 341)
normalized image coordinate: (0.532813, 0.710417)
ground coordinate: (0.190528, -0.002201, 0.000000)
image coordinate: (511, 328)
normalized image coordinate: (0.798438, 0.683333)
ground coordinate: (0.191026, -0.095562, 0.000000)
image coordinate: (234, 245)
normalized image coordinate: (0.365625, 0.510417)
ground coordinate: (0.304039, 0.085974, 0.000000)
image coordinate: (339, 245)
normalized image coordinate: (0.529687, 0.510417)
ground coordinate: (0.300432, -0.003149, 0.000000)
image coordinate: (445, 244)
```

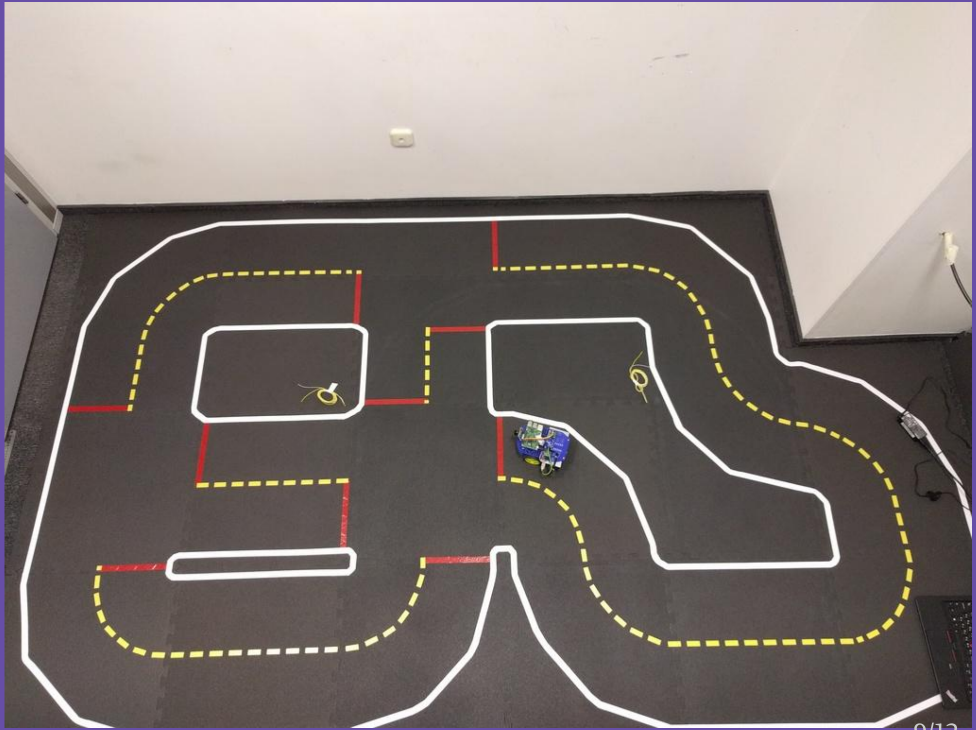
Camera calibration





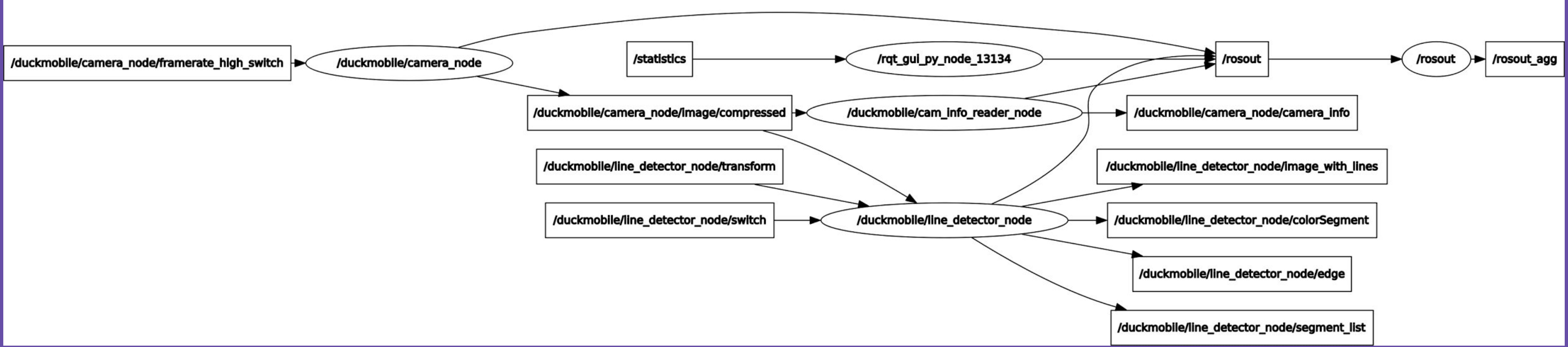
Duckiebot & duckietown





Line detection

A few words about ROS:


/duckmobile/camera_node/framerate_high_switch
/duckmobile/camera_node
/statistics
/rqt_gui_py_node_13134
/rosout
/rosout
/rosout_agg
/duckmobile/camera_node/image/compressed
/duckmobile/cam_info_reader_node
/duckmobile/camera_node/camera_info
/duckmobile/line_detector_node/transform
/duckmobile/line_detector_node/image_with_lines
/duckmobile/line_detector_node/switch
/duckmobile/line_detector_node
/duckmobile/line_detector_node/colorSegment
/duckmobile/line_detector_node/edge
/duckmobile/line_detector_node/segment_list

Line detection

Car’s vision and detection:

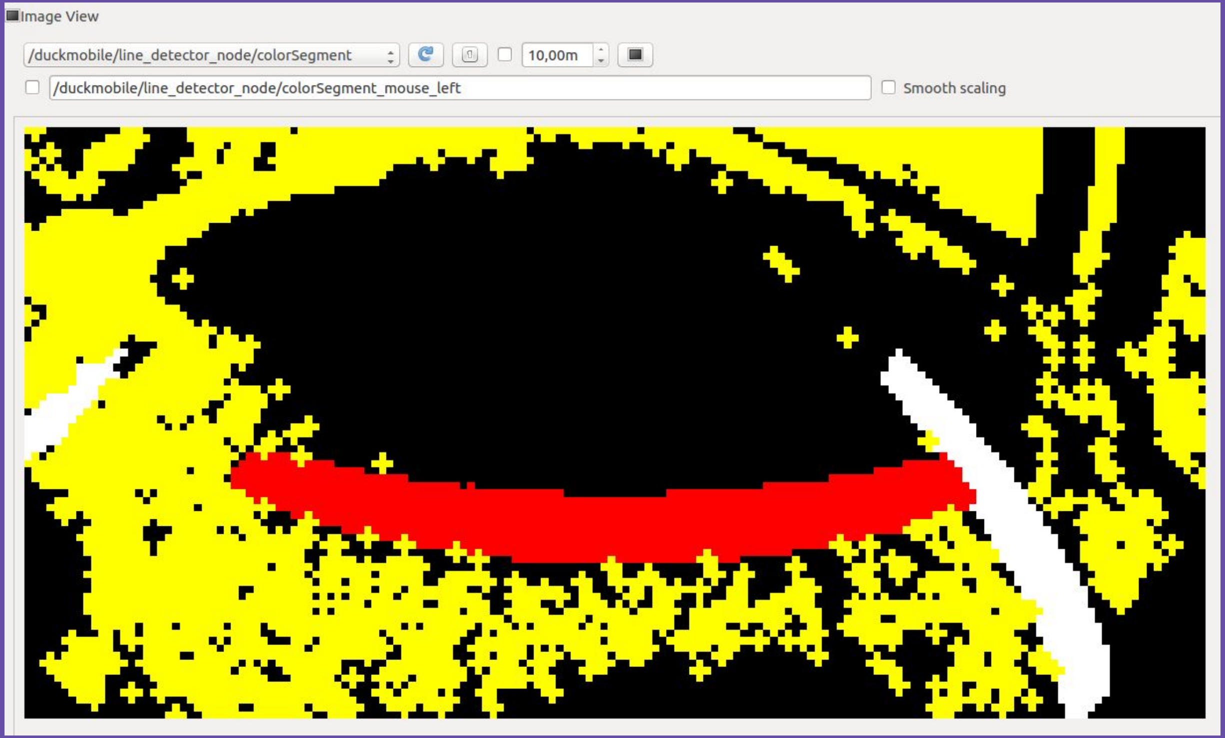


Image View

/duckmobile/line_detector_node/colorSegment

□ 10,00m

□ /duckmobile/line_detector_node/colorSegment_mouse_left

□ Smooth scaling

Line detection

Car’s vision and detection:

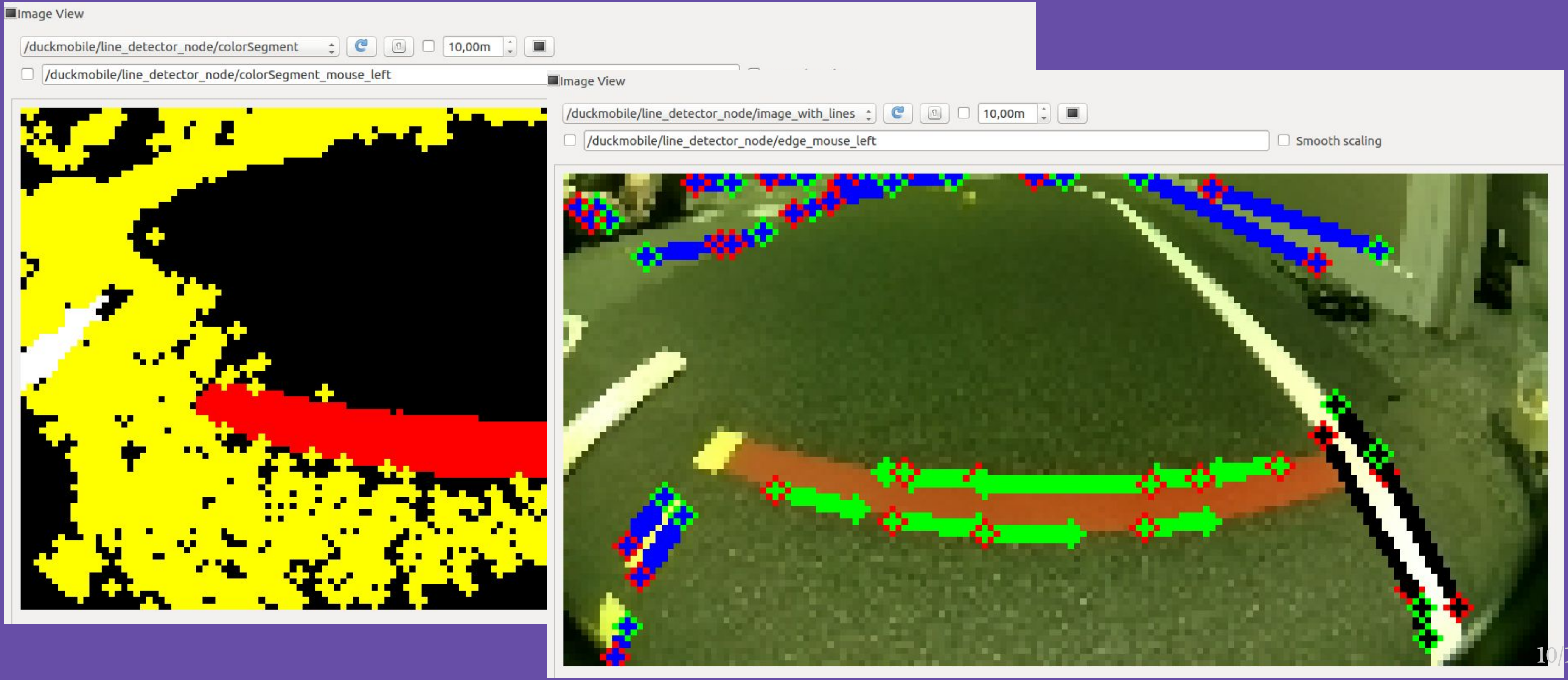


Image View

/duckmobile/line_detector_node/colorSegment

□ 10,00m

□ /duckmobile/line_detector_node/colorSegment_mouse_left

Image View

/duckmobile/line_detector_node/image_with_lines

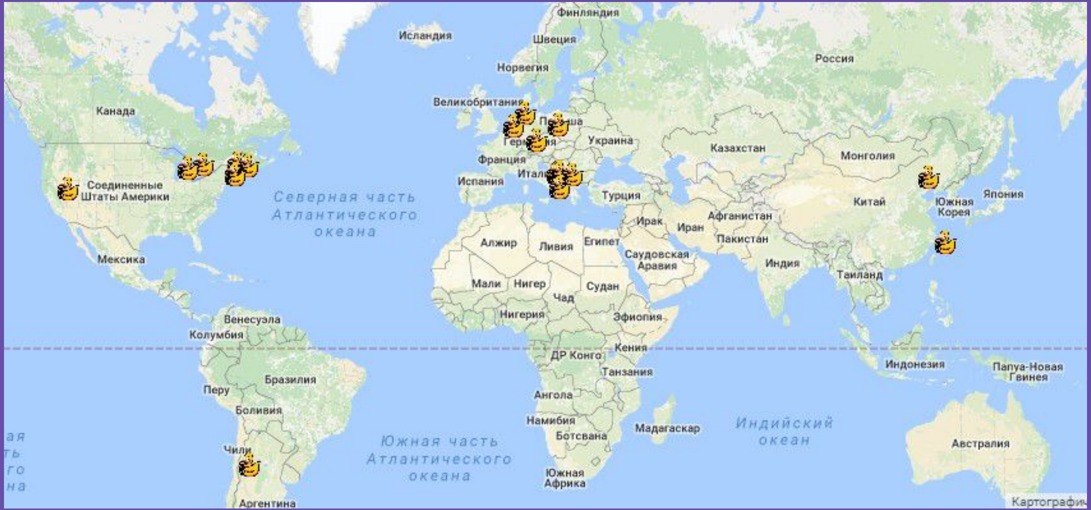
□ 10,00m

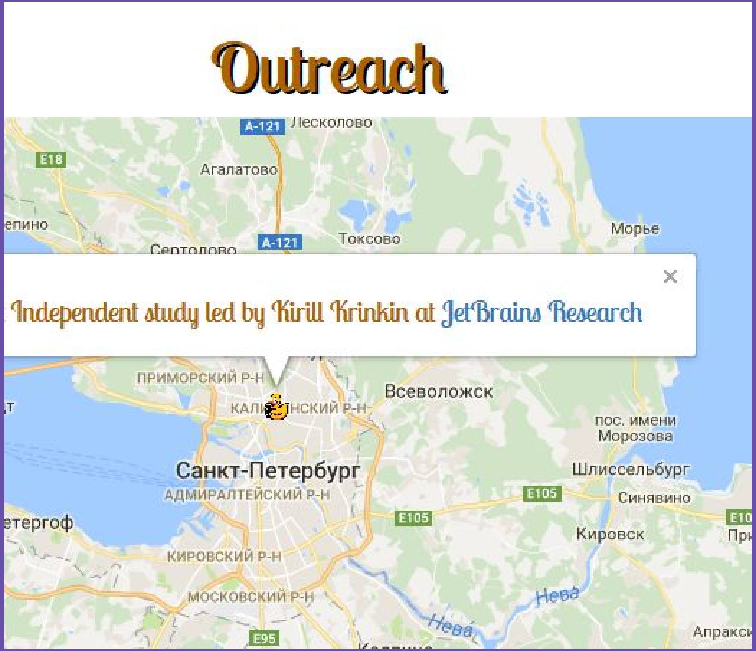
□ /duckmobile/line_detector_node/edge_mouse_left

□ Smooth scaling

Our current results:

https://www.youtube.com/watch?v=G5XkwwhuX5g&feature=youtu.be





Outreach

Independent study led by Kirill Krinkin at JetBrains Research

Welcome to Duckietown!