

# Инструкция по запуску робота Duckiebot

21.08.20

Код, установленный на компьютере можно собрать в контейнер (для этого в докерфайле установлены флаги для симулятора архитектуры ARM) и в нем скомпилировать, но запустить программу не получится, либо можно изменить код напрямую в контейнере. Для установки или сборки образа из исходного кода на рабочий компьютер выполнить:

1. Стянуть готовый образ с DockerHub:  
`docker push ematsuev/rpi-duckiebot-base`
2. Выполнить git clone  
`https://github.com/Laboratory-of-Embodied-Intelligence/Software.git && cd Software && docker build .`

raminaim/rpi-duckiebot-base:master-19 - это основной Docker-образ, в котором находится весь репозиторий duckietown/Software. Он сделан на основе duckietown/rpi-duckiebot-base:master-19, но пофикшен маленький баг, из-за которого бот не ездил в режиме автопилота.

Образ ematsuev/rpi-duckiebot-base синхронизируется с исходным кодом на github и собирается на DockerHub. Включает в себя изменения в образе raminaim/rpi-duckiebot-base:master-19, а также внедрен более стабильный метод распознавания линий, не требующий установки порогов для каждого из цветов.

**В первый раз после прошивки робота** скачать и запустить один из двух образов. С компа надо запустить следующее:

```
docker -H duckiebot.local run -it --net host --privileged --name base  
ematsuev/rpi-duckiebot-base:latest
```

либо (для старой версии, требующей настройки порогов)

```
docker -H duckiebot.local run -it --net host --privileged --name base  
raminaim/rpi-duckiebot-base:master-19
```

Это скачает на бота образ и запустит контейнер с именем base. Размер образа для скачивания - 2.8 Гб, поэтому ожидание может быть длительным. Если этот контейнер уже есть на боте, то можно просто запустить его и подключится к нему так, как написано ниже.

Далее вместо duckiebot или duckiebot.local подставить соответствующее имя бота. На 23.07 прошито 4 бота с именами duckiebot1 до duckiebot4.

Для запуска Lane\_following\_min необходимо:

1. Запустить бота, узнать его имя (у нашей единственной - duckiebot)  
(ssh duckie@ip)
2. Дать пару минут, пока на нем все прогрузится
3. С компа запустить контейнер base на боте ( docker -H duckiebot.local start base). Этот контейнер сделан нами, когда будут новые боты, инструкцию дополним
4. Подключится к контейнеру на боте ( docker -H duckiebot.local attach base ). После выполнения этой команды для доступа к терминалу на роботе нажать Ctrl+C.
5. Внутри контейнера необходимо запустить launch-файл, который поднимает все необходимые ноды:  
roslaunch duckietown\_demos lane\_following\_min.launch
6. С компа запустить контейнер с виртуальным джойстиком: dts duckiebot keyboard\_control duckiebot (ждать закидки образа на ноут минут 15, 4Гб объем, либо вручную загрузку через ssh docker pull duckietown/rpi-duckiebot-base:master19-no-arm)  
**Необходимо установленная duckietown-shell, подробнее [тут](#).**
7. Для просмотра изображения с камеры запускаем ещё один контейнер:  
dts start\_gui\_tools duckiebot. Можно увидеть изображение с камеры, выполнив отсюда rqt\_image\_view.

Можно слушать топики, **задавать пороги сегментации** и т.д., подключившись к контейнеру base в новом окне терминала с помощью команды:

```
docker -H duckiebot3.local exec -it base bash
```

По умолчанию публикуется только изображение с камеры, а после выполнения этой команды в новом терминале при запущенном rqt\_image\_view будут публиковаться также дополнительные картинки (маска с выделенными по цвету полосами, изображение с маркерами линий и т.д.) в топик

/image\_with\_lines (можно выбрать и увидеть результат в rqt\_image\_view):

```
rosparam set /duckiebot/line_detector_node/verbose true
```

Управление стрелочками, на клавишу A запускается “автопилот”, на клавишу S - “ручной режим”.

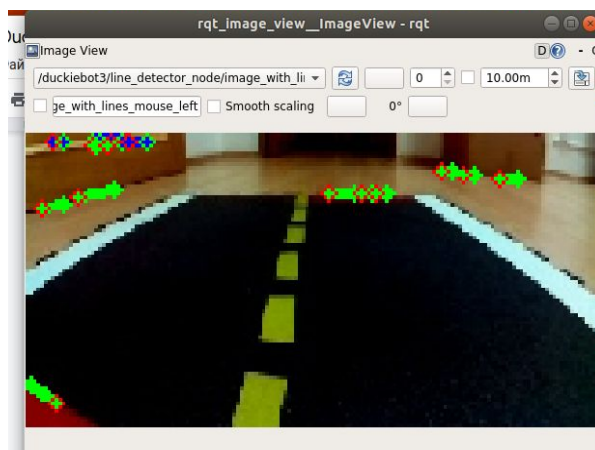
Для изменения конфига, в котором лежат цвета линий разметки для образа raminaim/rpi-duckiebot-base:master-19:

- зайти в контейнер base

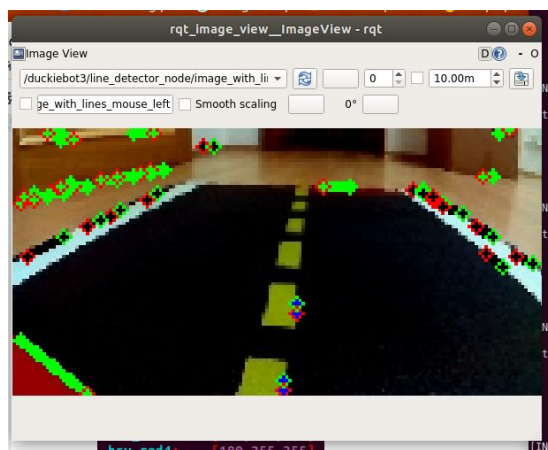
- открыть файл  
catkin\_ws/src/00-infrastructure/duckietown/config/baseline/line\_detector/line\_detector\_node/default.yaml
- на детекцию линий влияет структура **detector**:
  - hsv\_white1 - нижняя граница белого цвета в HSV
  - hsv\_white2 - верхняя граница белого цвета в HSV
  - Заполняется так: [ H/2, S/100\*255, V/100\*255 ]
  - Для желтого аналогично
  - Для красного второе и третье значение заполняется также во всех массивах, первое значение разбивается на два (190 -> 10).

Вот значения, при которых хорошо определяются линии на duckiebot2 (при полностью включенном свете):

```
hsv_white1: [60,10,170]
hsv_white2: [95,40,255]
hsv_yellow1: [25,160,150]
hsv_yellow2: [45,255,255]
hsv_red1: [0,140,100]
hsv_red2: [15,255,255]
hsv_red3: [165,140,100]
hsv_red4: [180,255,255]
```



до выставления порогов



после выставления порогов  
видно распознавание разметки посередине

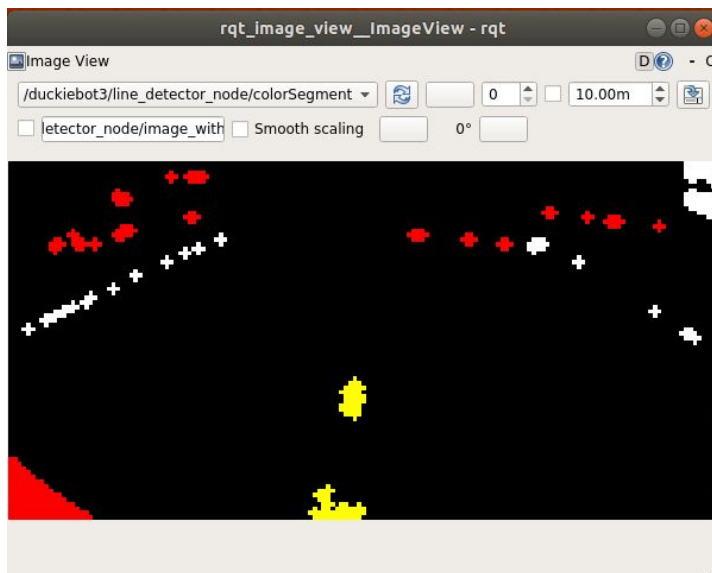
Калибровку делать по изображению из топика **/image\_transformer\_node/corrected\_image/compressed** и **colorSegment**. Для появления colorSegment в rqt\_image\_view нажать на кнопку обновления топиков рядом с их выбором.



Рис. Изображение из топика `/image_transformer_node/corrected_image/compressed`

Задачи:

- калибровка камеры (по шахматной доске). сильно влияет на точность движения по линии
- Калибровка цветов
- калибровка моторов (не главное)



## Установка операционной системы на роботе

не рекомендуется прошивать карту командой `dd`. Пробовали, разделы теряются. Подробнее тут: <https://www.raspberrypi.org/forums/viewtopic.php?t=46911>

Официальная инструкция по прошивке для ознакомления:

[https://docs.duckietown.org/DT18/opmanual\\_duckiebot/out/setup\\_duckiebot.html](https://docs.duckietown.org/DT18/opmanual_duckiebot/out/setup_duckiebot.html)

Для установки на Raspberry4 нужно выполнить команды, где вместо password нужно подставить пароль от wifi, к которому подключится робот.

1. `dts --set-version master19`
2. `dts init_sd_card --hostname duckiebot52 --wifi "CSI Robotics":password --country RU --experimental`