

Руководство пользователя виртуального тренажера для подготовки к соревнованиям «Аэробот-2024». Установка и настройка виртуального тренажера

1. Установить чистую [Ubuntu 22.04](#)
2. Сделать обновление ссылок и обновление пакетов ОС с помощью команд:

```
sudo apt update  
sudo apt upgrade
```

3. Установить [ROS2 Humble](#).
4. Внесем изменения в bash

```
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

5. Установить пакет setuptools.

```
pip3 install setuptools==58.2.0
```

6. Установить [Gazebo Garden](#).
7. Установить mavros:

```
sudo apt install ros-humble-mavros  
sudo apt install ros-humble-mavros-extras
```

8. Установить GeographicLib

```
sudo su  
  
wget  
https://raw.githubusercontent.com/mavlink/mavros/ros2/mavros/scripts/install_geographiclib_datasets.sh  
  
. install_geographiclib_datasets.sh  
  
exit
```

9. Установить colcon

```
sudo sh -c 'echo "deb [arch=amd64,arm64] http://repo.ros2.org/ubuntu/main
`lsb_release -cs` main" > /etc/apt/sources.list.d/ros2-latest.list'

curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc |
sudo apt-key add -

sudo apt update

sudo apt install python3-colcon-common-extensions
```

10. Установить репозиторий ros_gz

```
sudo sh -c 'echo "deb [arch=$(dpkg --print-architecture)]
http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" >
/etc/apt/sources.list.d/ros2-latest.list'

curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc |
sudo apt-key add -

sudo apt-get update

sudo apt install ros-humble-ros-gz-garden
```

11. Установить git

```
sudo apt install git
```

12. Установить виртуальный тренажер:

```
cd
git clone https://github.com/MaximVorotnikov/Firmware2.git
cd Firmware2
colcon build
```

13. Установить автопилот:

```
cd
git clone https://github.com/PX4/PX4-Autopilot.git --recursive
bash ./PX4-Autopilot/Tools/setup/ubuntu.sh --no-sim-tools
```

После этого обязательно перезапускаем компьютер, далее прописываем

```
cd PX4-Autopilot/  
make px4_sitl
```

Теперь нужно внести изменения в файловой системе пакета PX4-Autopilot
Для этого

```
cd ~/PX4-Autopilot/ROMFS/px4fmu_common/init.d-posix/airframes  
touch 4013_gz_x500_mono_cam_forward_down_drone  
sudo nano 4013_gz_x500_mono_cam_forward_down_drone
```

Вводим пароль и вставляем в этот файл следующее:

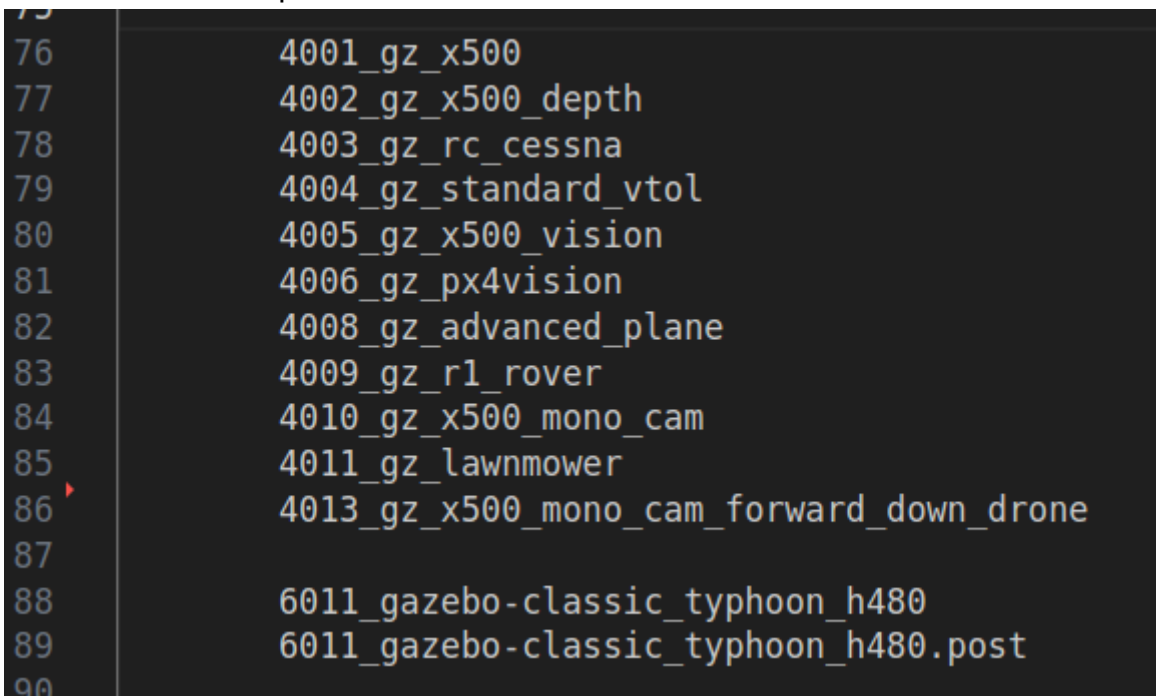
```
PX4_SIM_MODEL=${PX4_SIM_MODEL:=x500_mono_cam_forward_down_drone}  
  
. ${R}etc/init.d-posix/airframes/4001_gz_x500  
  
param set COM_OF_LOSS_T 20
```

Сохраняем и выходим (CTRL+S и CTRL+X)

Теперь нужно дополнить CMakeLists.txt

```
cd ~/PX4-Autopilot/ROMFS/px4fmu_common/init.d-posix/airframes  
sudo nano CMakeLists.txt
```

В самом CMake листе нужно дописать 4013_gz_x500_mono_cam_forward_down_drone
как показано на скриншоте:



```
75  
76     4001_gz_x500  
77     4002_gz_x500_depth  
78     4003_gz_rc_cessna  
79     4004_gz_standard_vtol  
80     4005_gz_x500_vision  
81     4006_gz_px4vision  
82     4008_gz_advanced_plane  
83     4009_gz_r1_rover  
84     4010_gz_x500_mono_cam  
85     4011_gz_lawnmower  
86     4013_gz_x500_mono_cam_forward_down_drone  
87  
88     6011_gazebo-classic_typhoon_h480  
89     6011_gazebo-classic_typhoon_h480.post  
90
```

После этого также выходим через CTRL+S и CTRL+X

Осталось сбилдить пакет

```
cd
cd ~/PX4-Autopilot
make px4_sitl
```

1. Внести source необходимых файлов в ~/.bashrc :

```
echo "export
GZ_SIM_RESOURCE_PATH=$HOME/Firmware2/src/aerobot_gz_sim/worlds:$HOME/Firmw
are2/src/aerobot_gz_sim/models:${GZ_SIM_RESOURCE_PATH}" >> ~/.bashrc
source ~/.bashrc
```

Перезапускаем терминал и можно запускать симулятор

Запуск симулятора

Запускать симулятор можно как с mavros, так и без него, в зависимости от того, кому он нужен, а кому нет:

1. Выполнить source необходимых файлов:

```
cd ~/Firmware2
. install/setup.bash
```

2. Запуск первого мира без mavros

```
ros2 launch aerobot_gz_sim aerobot_first_test.launch.py
```

3. Запуск второго мира без mavros с движущимся qr-кодами

```
ros2 launch aerobot_gz_sim aerobot_second_test.launch.py
```

4. Запуск второго мира без mavros со статичными qr-кодами

```
ros2 launch aerobot_gz_sim aerobot_second_test_without_car.launch.py
```

5. Запуск третьего мира без mavros

```
ros2 launch aerobot_gz_sim aerobot_third_test.launch.py
```

6. Запуск первого мира с mavros

```
ros2 launch aerobot_gz_sim aerobot_first_test_mavros.launch.py
```

7. Запуск второго мира с mavros с движущимся qr-кодами

```
ros2 launch aerobot_gz_sim aerobot_second_test_mavros.launch.py
```

8. Запуск второго мира с mavros со статичными qr-кодами

```
ros2 launch aerobot_gz_sim  
aerobot_second_test_mavros_without_car.launch.py
```

9. Запуск третьего мира с mavros

```
ros2 launch aerobot_gz_sim aerobot_third_test_mavros.launch.py
```

Интерфейс симулятора

Сцена первого задания выглядит таким образом:



Дополнительно

Корректировка параметров полётного контроллера может осуществляться через программу QGround Control, для этого ее необходимо установить по [ссылке](#). Программа при запущенном симуляторе должна автоматически подключиться к виртуальному полетному контроллеру.

Для работы с симулятором участникам доступен следующий список ROS2 топиков с данными:

`/uav1/scan` - данные с кругового лидара;
`/uav1/camera_down` - изображение с камеры, направленной вниз;
`/uav1/camera` - изображение с камеры, направленной вперед;
`/uav1/depth_camera` - камера глубины, направленная вперед;
`/uav1/mavros/imu/data` - данные с инерциальных датчиков (будет доступен при запуске с mavros)

По вопросам технической поддержки обращаться:

Воротников Максим Игоревич т.м. +7-918-098-01-91 (доступен WhatsApp, Telegram)

email: mvorotnikov@sfedu.ru