



РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
имени А.Н.КОСЫГИНА  
(Технологии. Дизайн. Искусство)

**Voltbro**

www.voltbro.com

## Регламент очного этапа Хакатона

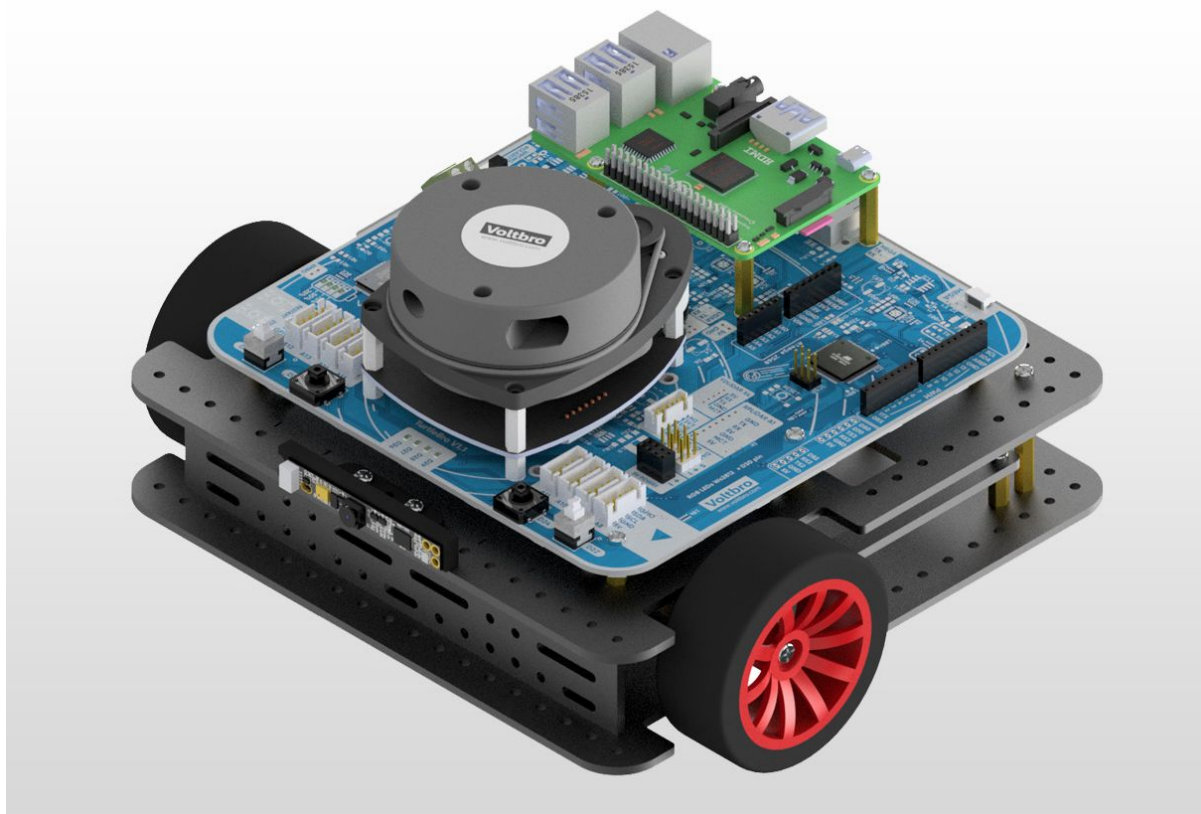
Российского Государственного Университета им А.Н.Косыгина

по Robot Operating System

от компании VoltBro

### Описание:

На хакатоне участники получают практические навыки программирования и администрирования сервисного робота [TurtleBro](#) под управлением [Robot Operating System \(ROS\)](#).



Мы предложим два этапа заданий на наших роботах:

1-й этап - **"Заведи себе робота"** - начальный уровень, простая веселая задача предназначенная для знакомства с ROS и испытания реальных роботов в деле. Мы ожидаем что время выполнения задания будет 2 часа. Данную задачу предложим в формате мастер-класса, т.е. участники будут делать ее вместе с организаторами, которые направят и помогут командам.

2-й этап - **"На старт! Внимание! Марш!"** - соревновательная задача, она посложнее, требует хорошего командного взаимодействия и необходимости вложиться в решение. На выполнение этой задачи участникам отводится 4 часа, но придется постараться.

### **Особенности проведения on-line формата:**

Все участники получают параметры удаленного доступа к роботам находящимся на полигоне организаторов. Каждый робот находится в своей тестовой зоне полигона, таким образом, чтобы одна команда не мешала другой. На полигоне также будут находиться техники, которые помогут участникам в случае проблем с роботом. Все участники покомандно заходят в Discord, основное общение происходит там. Участники сами могут выбирать альтернативные варианты общения внутри команд, но организаторы будут общаться с участниками только через Discord.

**Программа, календарь хакатона [доступна по ссылке](#).**

### **Описание Хакатона.**

#### **1-й этап - "Заведи себе робота"**

Данная задача будет решена в формате мастер-класса. Т.е. Организаторы покажут один из вариантов ее решения, комментируя все непонятные моменты в коде и алгоритмах. Далее участники самостоятельно запускают созданные программы и тестируют их выполнение на своих роботах. Основная цель - изучить возможности фреймворка ROS и написать свою первую программу для управления роботом.

**Задача:** Робот ставится "лицом" к стене полигона. После запуска программы он должен подъехать к стене ровно на 30 см. остановиться подождать несколько секунд и затем развернуться точно на 180 градусов.

Чем точнее и быстрее робот подъедет к стене, тем лучше.

Расстояние между стеной и роботом измеряется от переднего края рамы робота.

#### **2-й этап - "На старт! Внимание! Марш!"**

Данная задача является зачетной, и по ней будут выставляться баллы участникам.

Организаторы не будут помогать в решении этой задачи.

**Задача:** Задача: Подъехать на роботе к точке старта. Развернуться и остановиться у линии "Старт/Финиш". Затем проехать по полигону поворачивая в каждой вершине

прямоугольного полигона направо и остановиться после пересечения линии “Старт/Финиш”.

Для демонстрации выполнения программы представитель команды Участников запускает программу для показа экрана Судьям.

Линия “Старт/Финиш” нанесена на полу полигона следующим образом: Она является перпендикуляром к одной из стен полигона и отстоит от другой стены, расположенной спереди по ходу движения робота на 40 см. (см. рис. Схема полигона)



рис. Схема полигона

Перед началом выполнения задания робот устанавливается на стартовую позицию внутри полигона (см. Место установки робота). После чего судья подает команду “Начало попытки” и участник команды запускает программу управляющую роботом.

Робот должен подъехать к стене перед ним, на расстояние 20 см. от центра лидара, до стены и остановиться. После чего судья проводит измерение точности остановки. После чего судья дает команду “На старт” и участник должен продолжить работу программы, чтобы робот начал движение в сторону линии “Старт/Финиш”.

Далее робот должен подъехать к линии “Старт/Финиш” и остановиться перед ней таким образом, чтобы расстояние от центра лидара робота до левой стены полигона по направлению движения робота было ровно 20 см. А расстояние от линии “Старт/Финиш” до переднего края рамы робота было наименее возможным.

После измерения расстояния до линии “Старт/Финиш”, судья подает команду “Внимание!”, затем “Марш!”, по получении которой участник продолжает работу программы.

Программа должна быть написана таким образом, чтобы робот автономно останавливался перед линией “Старт/Финиш”, в каждом углу полигона и после пересечения линии “Старт/Финиш” после завершения полного “круга”.

Остановка в углах полигона должна осуществляться таким образом чтобы два перпендикуляра опущенные из центра лидара робота на соответствующие стены полигона были равны 20 см. каждый.

В каждом углу судья производит измерения точности остановки, после чего дает участникам команду “Продолжаем”, после которой участник, должен возобновить выполнения программы управляющей роботом.

После пересечения линии “Старт/Финиш” робот должен остановиться таким образом, чтобы расстояние от заднего края рамы робота до линии “Старт/Финиш” было минимальным. После полной остановки робота судья подает команду “Стоп”, измеряет расстояние от заднего края рамы робота до линии “Старт/Финиш” и попытка считается завершенной.

Время выполнения попытки фиксируется судьей между отданной судьей командой “На старт” и отданной судьей командой “Стоп”. Всего на выполнение задания дается 2 минуты (за вычетом приостановок контроля времени на измерения расстояний). Время на выполнение задание измеряется по личным часам судьи. Судья на свое усмотрение дает команду прекратить попытку, если видит, что время выполнения программы превышает 2 минуты. Результатом команды является набранное за прошедшее время количество баллов.

Если в процессе выполнения попытки, с момента старта из места установки робота, робот сбивает ограждение полигона, то попытка команды прекращается, баллы за пройденные идут в зачет, а баллы за непройденные точки обнуляются.

Если в процессе выполнения попытки робот “пропускает” остановку в каком либо углу, и останавливается только в “следующем”, то измерения в момент остановки ведутся от текущего положения остановившегося робота, до “пропущенного” угла, а не до ближайшего угла.

Если в процессе выполнения попытки возникают технические неполадки, например падает канал интернета или др., то команде может быть дана возможность перезаезда без потери попытки (на усмотрение комиссии судей).

Критерии оценки:

- Точность остановки в точках полигона - 10 баллов (max).

Команды выполнившие задание на точность остановки в промежутке измеренных расстояний между самой точной и самой неточной попыткой получают баллы обратно пропорционально измеренному расстоянию в попытке выполнения задания по формуле:

$bs = (1 - (|SLsc - SLsi|)/(|SLsw - SLsi|)) * 10$ , где:

bs - баллы команды за попытку выполнения задания на точность остановки

SLsw - сумма измеренных расстояний той командой, которое стало самым неточным выполнением задания из всех команд

SLsc - сумма всех измеренных расстояний от робота оцениваемой команды до соответствующих точек.

SLsi - идеальная сумма всех измеренных расстояний от робота до соответствующих точек.

*Примечание: Отклонение расстояний в углах полигона меряется по модулю отклонения от 20 сантиметровых перпендикуляров. Отклонение расстояний на линии "Старт/Финиш" меряется ДО линии "Старт" и х3 при ее пересечении и ОТ линии "Финиш" и х3 при "не полном пересечении".*

Количество попыток будет определяться по скорости прохождения финальных испытаний всеми командами и будет не менее двух. Но в случае, если останется достаточно времени организаторы могут добавить попыток прохождения всем командам. Команды вправе отказаться от прохождения любой попытки. В финальный зачет идут баллы лучшей попытки. Победителем становится команда набравшая наибольшее количество баллов в лучшей попытке среди всех команд.

### **Общие принципы оценки и судейства**

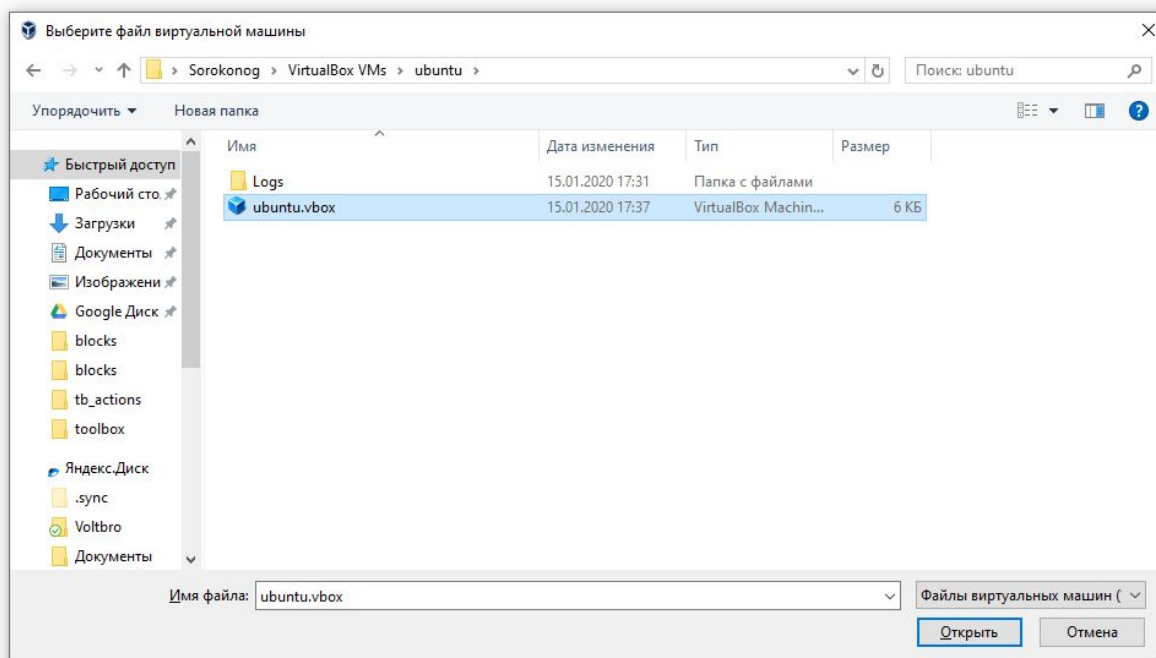
1. Мы за честную игру. Общие принципы честной игры предполагают, что все участники находятся в равных условиях и относятся друг к другу и судьям в духе спортивного соперничества.
2. Не допускается решение автономных задач на ручном управлении.
3. Читить плохо! Читингом считается выполнение задания таким образом, который по мнению большинства участников является нечестным.
4. Участники не должны мешать друг другу выполнять задания и работать в рамках хакатона. Нарушители будут подвергнуты общественному ostracismu и осуждению (у нас есть все логи).
5. Если какое-то задание команда пытается выполнить "нечестным" путем, судьи вправе обнулить результаты части задания или вовсе не засчитать попытку.
6. Во всех спорных ситуациях мнение судьи является решающим.
7. Решение судьи можно апеллировать к комиссии судей. Подача апелляции возможна только в письменном виде не позднее 15 минут с момента вынесения спорного решения.
8. Решение комиссии судей является окончательным.

## Рекомендации участникам при подготовке:

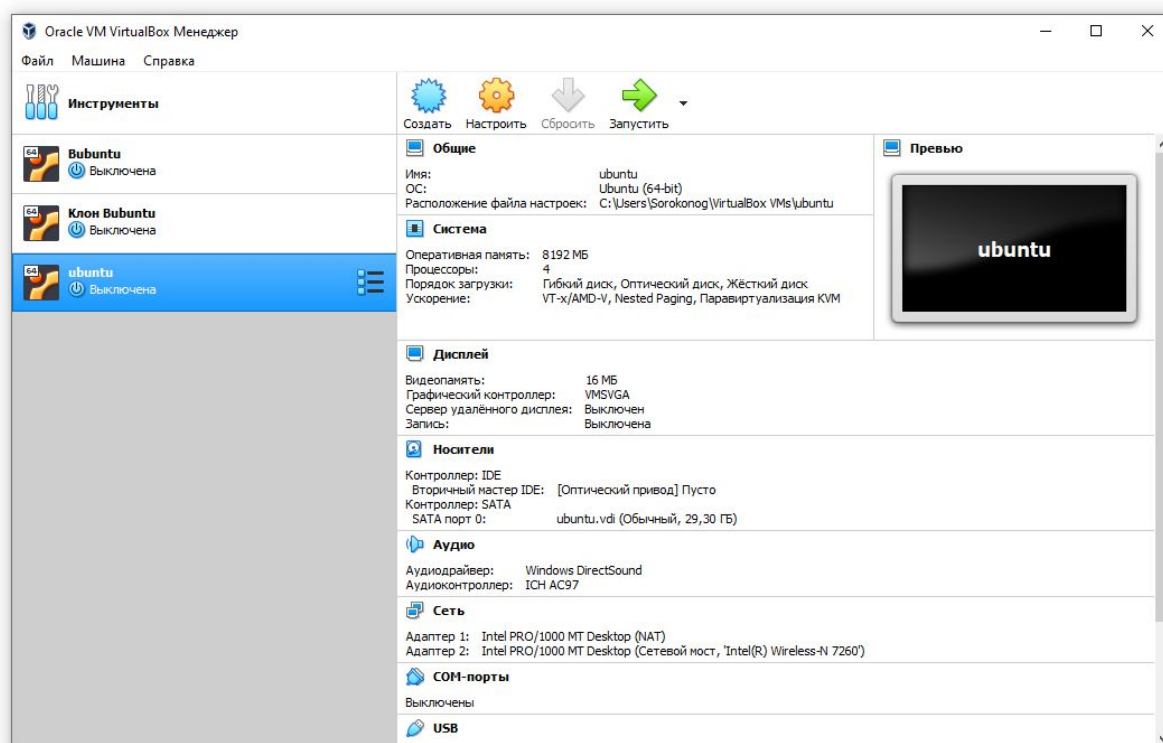
В этом разделе мы предлагаем рекомендации по подготовке к соревнованию, которые настоятельно просим выполнить.

1. Для более комфортной работы с роботом желательно самостоятельно установить на **собственные** ноутбуки\
2. Для того чтобы изучить основы ROS (Robot Operating System) и познакомиться с роботом TurtleBro, необходимо прочитать следующие материалы и пройти следующие курсы:
  - a. <http://learn.voltbro.ru/free/ros-intro/> Основы Linux + Python для ROS
  - b. <http://docs.voltbro.ru/starting-ros/> Книга "Введение в ROS"
  - c. <https://manual.turtlebro.ru> Инструкция к роботу TurtleBro
  - d. <https://stepik.org/course/3222/syllabus> Курс Введение в ROS
  - e. <https://stepik.org/course/73/syllabus> Краткое введение в Линукс (или любой другой курс по Линукс)
3. Мета-операционная система ROS работает под Linux, поэтому для того, чтобы запускать программы для ROS, необходимо установить на ваши компьютеры OS Linux и поставить на него ROS. Мы предполагаем, что вы работаете на компьютере с MS Windows версии старше MS Windows 8 с не менее 8Гб оперативной памяти и не менее 4-х ядерным процессором. Основной и наиболее удобный способ поставить Linux в этом случае - использовать виртуальную машину. Ниже пошаговая инструкция для установки виртуальной машины с настроенной OS Linux и установленной ROS:
  - a. Загрузить менеджер виртуальных машин Oracle VirtualBox.  
<https://www.virtualbox.org/wiki/Downloads>
  - b. Скачать образ виртуальной машины по ссылке:  
[https://yadi.sk/d/nn\\_WoywidiRnXDg?w=1](https://yadi.sk/d/nn_WoywidiRnXDg?w=1)
  - c. Добавить скачанную виртуальную машину в VirtualBox. Ctrl + A.
  - d. Настроить сетевые адаптеры в виртуальной машине в соответствии с реальными сетевыми адаптерами присутствующими на вашем компьютере.

е. Выберите файл с виртуальной машиной.



ф. Нажмите Открыть. Виртуальная машина появится в списке.



г. Запустите виртуальную машину кнопкой "Запустить"

h. Подождите пока она загрузится.

i. Доступ в виртуальную машину **login:** robot **password:** robot

4. В случае если основной компьютер у вас работает под ОС Linux, вам необходимо поставить ROS самостоятельно пользуясь официальной документацией с сайта РОС:

a. Ubuntu 20.04 LTS

<https://ubuntu.com/download/desktop/thank-you?version=20.04.1&architecture=amd64>

b. ROS Noetic

<http://wiki.ros.org/noetic/Installation>

c. Любую удобную среду разработки (редактор кода) для Python. Например VSCode.