

# **ИСПОЛЬЗОВАНИЕ АЛЬТЕРНАТИВНОГО ПО ПРИ РЕАЛИЗАЦИИ МОДУЛЯ РОБОТОТЕХНИКА**

**Родыгин С.Д., учитель труда (технологии)  
МБОУ лицей им. генерал-майора  
Хисматулина Василия Ивановича**

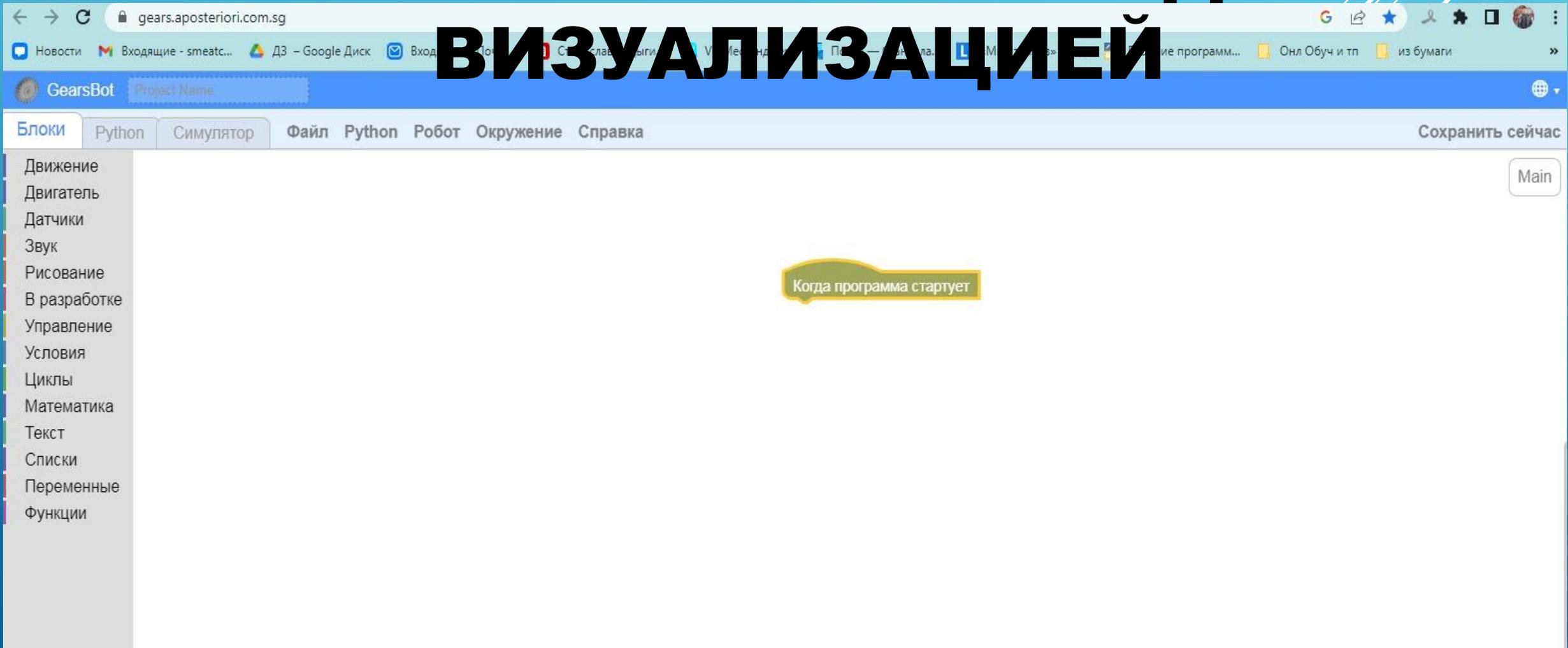
**2025**

**PECYPC:  
GEARS BOT**

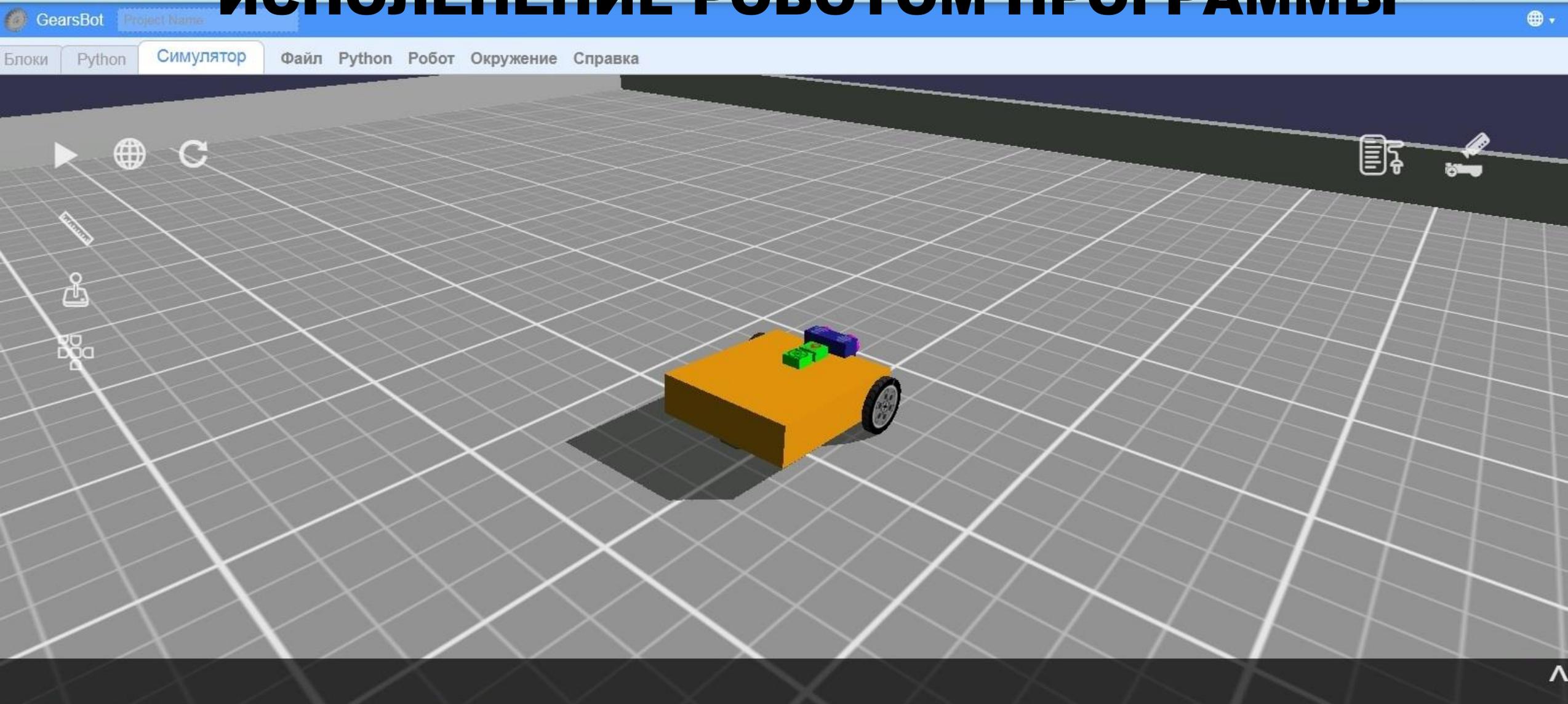
**[HTTPS://GEARS.APOSTERIORI.C  
OM.SG/](https://gears.aposteriori.com.sg/)**

# ПРЕДСТАВЛЯЕТ СОБОЙ СИМУЛЯТОР ПРОГРАММИРОВАНИЯ С УДОБНОЙ

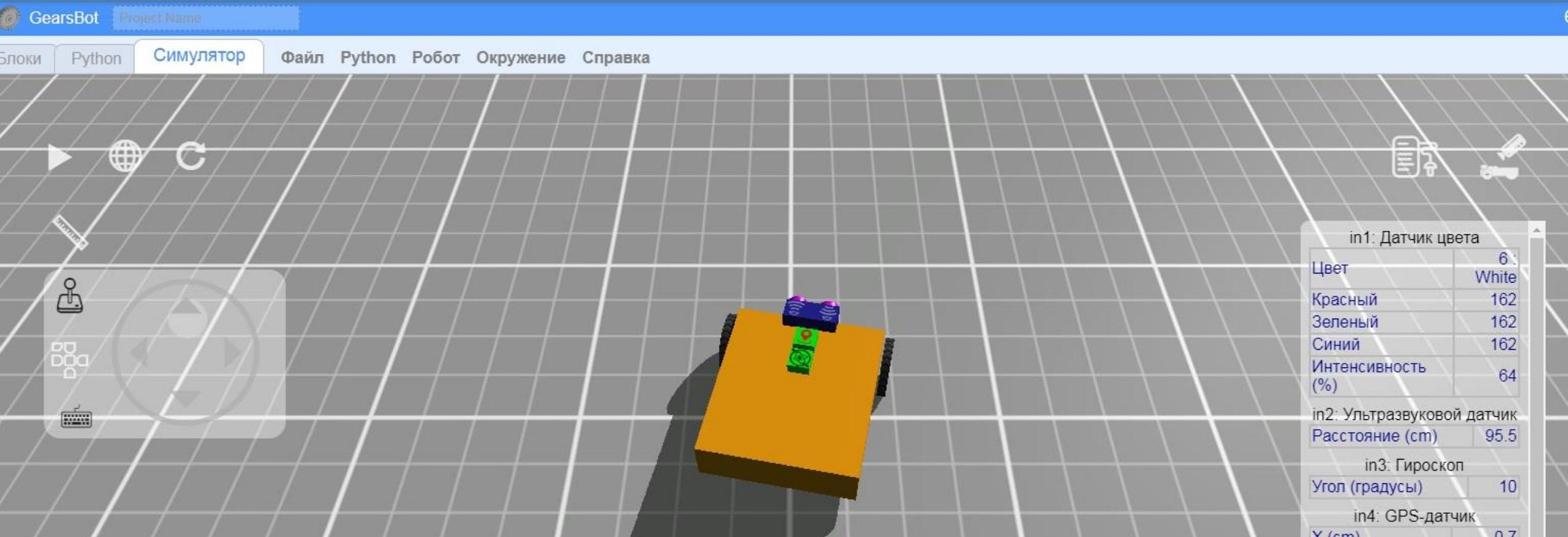
# ВИЗУАЛИЗАЦИЕЙ



# В ОТЛИЧИИ ОТ ДРУГИХ СХОЖИХ ПЛАТФОРМ ПОЗВОЛЯЕТ ПРОВЕРИТЬ И ПОСМОТРЕТЬ ИСПОЛНЕНИЕ РОБОТОМ ПРОГРАММЫ



# ВОЗМОЖНОСТЬ ПРИМЕНЕНИЯ НА ТЕЛЕФОНЕ, ЧТО МОЖЕТ ПОЗВОЛИТЬ РЕАЛИЗАЦИЮ МОДУЛЯ РОБОТОТЕХНИКИ ПРИ ПОЛНОМ ОТСУТСТВИИ ОБОРУДОВАНИЯ



# КОНЕЧНО С ОГОВОРКОЙ НА ОГРАНИЧЕНИЕ ИСПОЛЬЗОВАНИЯ СРЕДСТВ СВЯЗИ ПРИ ОБУЧЕНИИ И НЕ ОЧЕНЬ УДОБНЫМ ИНТЕРФЕЙСОМ ДЛЯ ТЕЛЕФОНА

Чтобы обновить Google Chrome, нужна Windows 10 или более поздней версии. У вас установлена Windows 7.

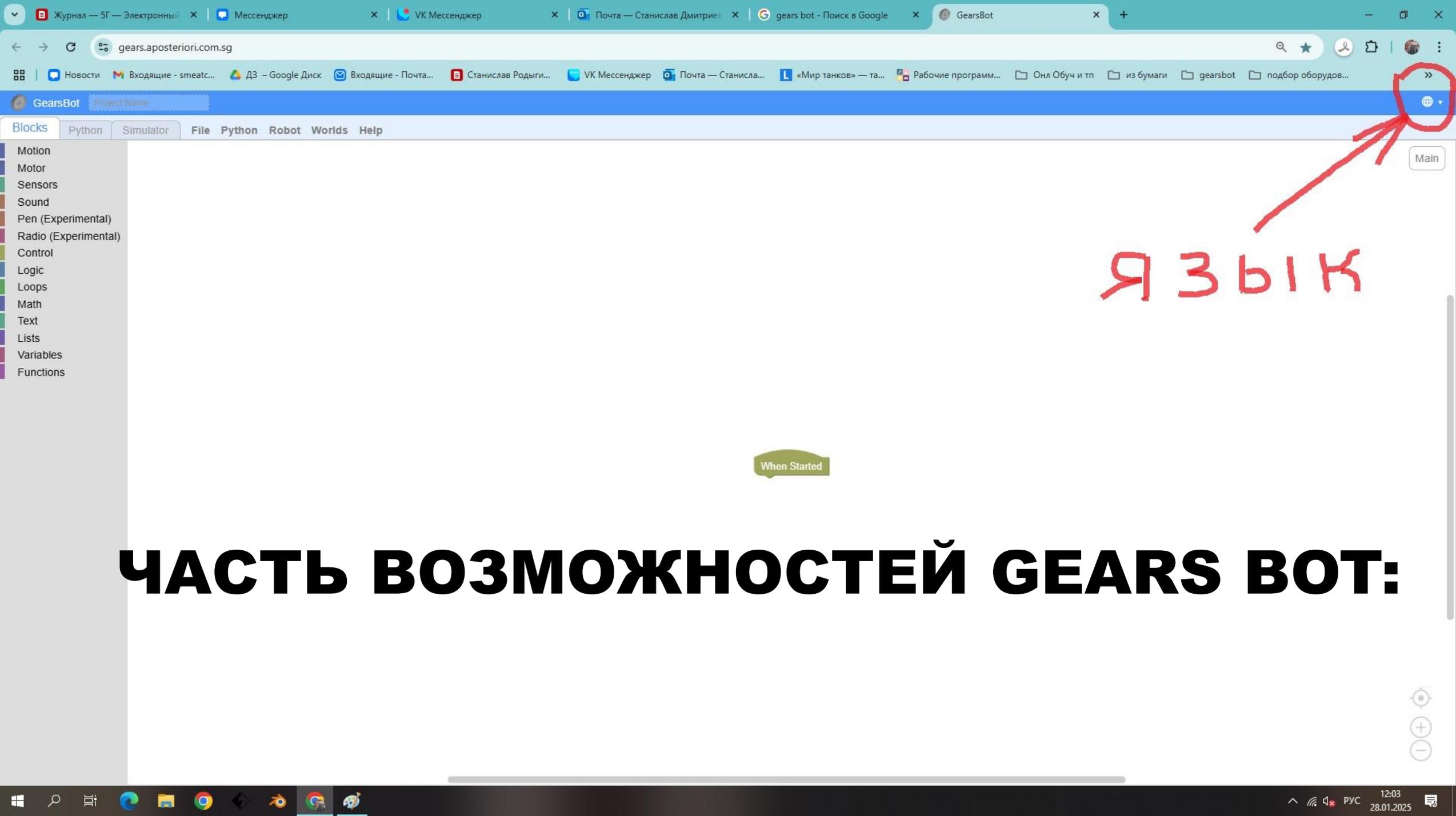
The screenshot displays the GearsBot web interface. On the left, a 3D simulation shows a yellow robot on a grid. A menu is open with options: "Выбрать робота", "Конфигуратор робота", "Загрузить из файла", "Сохранить в файл", "Показать текущую позицию", "Сохранить текущую позицию в настройки", and "Сбросить позицию в настройках". A table shows sensor data for "in1: Датчик цвета", "in2: Ультразвуковой датчик", "in3: Гироскоп", "in4: GPS-датчик", and "in5: Перо". On the right, a Python code editor shows the following code:

```
1 #!/usr/bin/env python3
2
3 # Import the necessary libraries
4 import time
5 import math
6 from ev3dev2.motor import *
7 from ev3dev2.sound import Sound
8 from ev3dev2.button import Button
9 from ev3dev2.sensor import *
10 from ev3dev2.sensor.lego import *
11 from ev3dev2.sensor.virtual import *
12
13 # Create the sensors and motors objects
14 motorA = LargeMotor(OUTPUT_A)
15 motorB = LargeMotor(OUTPUT_B)
16 left_motor = motorA
17 right_motor = motorB
18 tank_drive = MoveTank(OUTPUT_A, OUTPUT_B)
19 steering_drive = MoveSteering(OUTPUT_A, OUTPUT_B)
20
21 spkr = Sound()
22 btn = Button()
23 radio = Radio()
24 obtr = ObjectTracker()
25
26 color_sensor_in1 = ColorSensor(INPUT_1)
27 ultrasonic_sensor_in2 = UltrasonicSensor(INPUT_2)
28 evro_sensor_in3 = GyroSensor(INPUT_3)
```

**ЕСТЬ ОБУЧАЮЩИЕ ИНСТРУКЦИИ, НО  
ОНИ НЕ ВСЕГДА ПОЛНЫ И  
ПРАВИЛЬНО ПЕРЕВОДЯТСЯ:**

**[HTTPS://WWW.APOSTERIORI.COM.SG/G  
EARSBOT-TUTORIAL/](https://www.aposteriori.com.sg/gearsbot-tutorial/)**

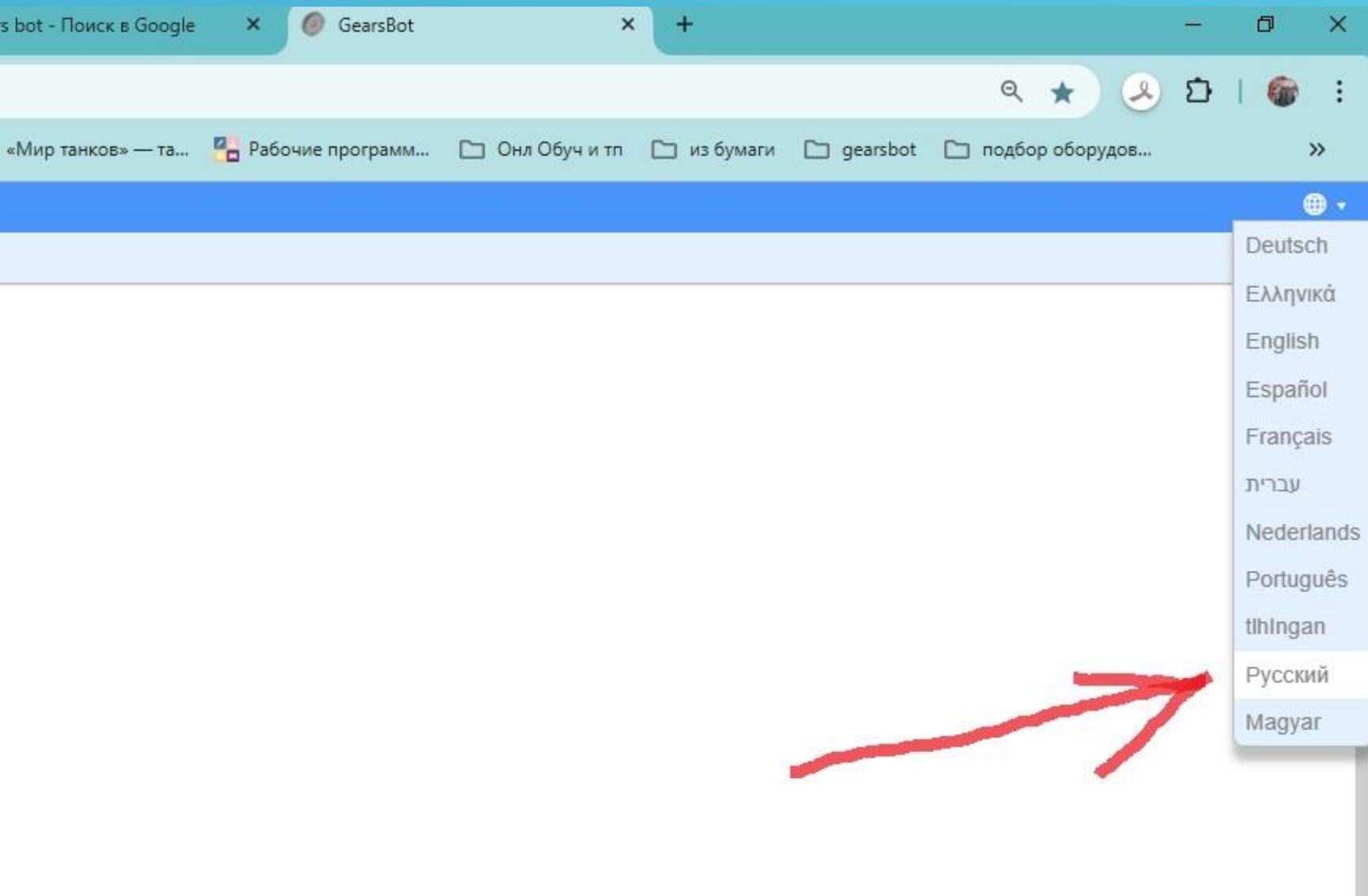
**[HTTPS://GITHUB.COM/QUIRKYCORT/GE  
ARS/WIKI](https://github.com/quirkycort/gears/wiki)**

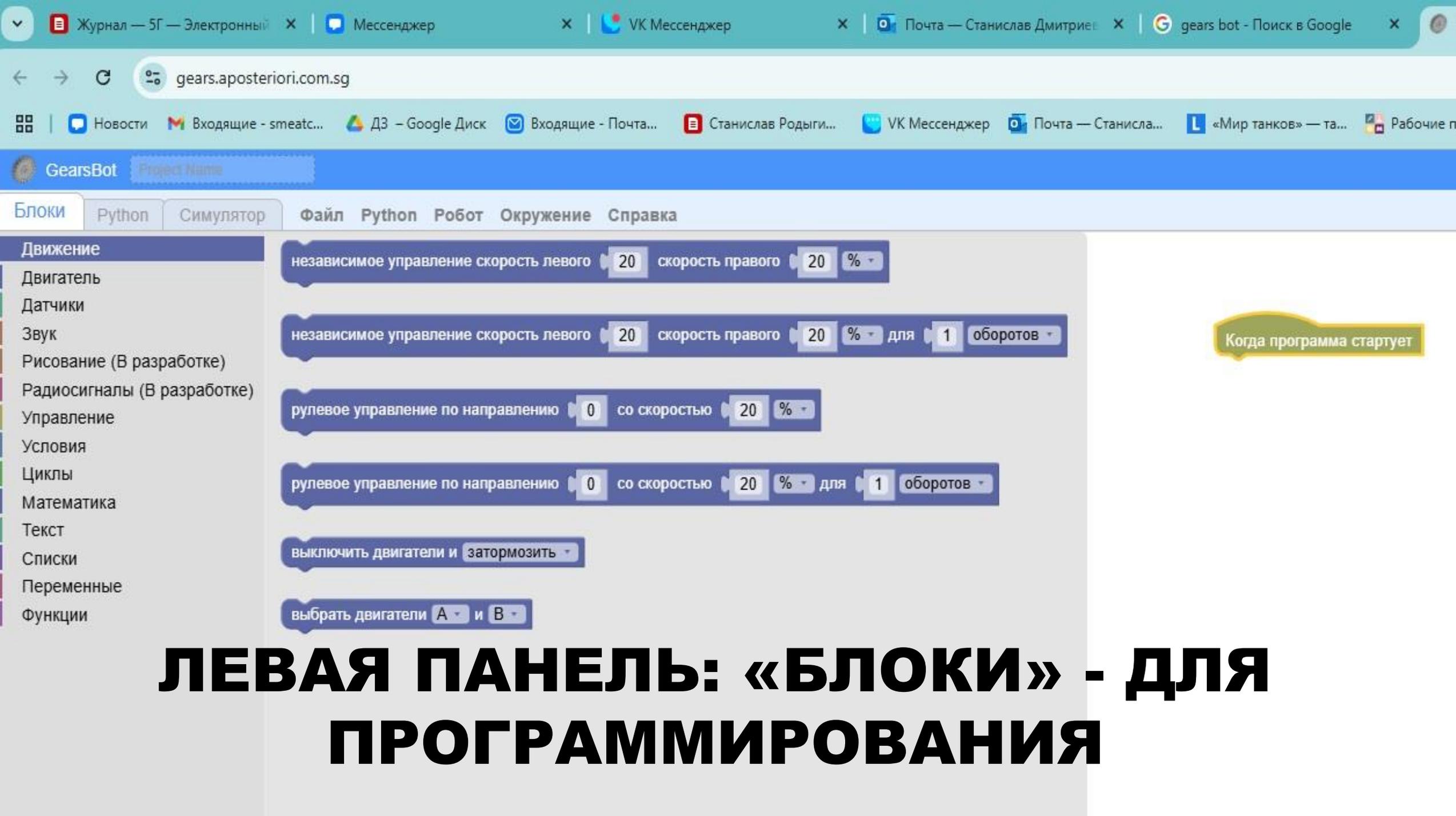


ЯЗЫК

# ЧАСТЬ ВОЗМОЖНОСТЕЙ GEARS BOT:

# СМЕНА ЯЗЫКА СПРАВА СВЕРХУ:





**ЛЕВАЯ ПАНЕЛЬ: «БЛОКИ» - ДЛЯ ПРОГРАММИРОВАНИЯ**

gears.aposteriori.com.sg

Новости | Входящие - smeats... | ДЗ - Google Диск | Входящие - Почта... | Станислав Родыги... | VK Мессенджер | Почта - Станисла... | «Мир тан

GearsBot Проект Name

Блоки Python Симулятор Файл Python Робот Окружение Справка

Движение  
Двигатель  
Датчики  
Звук  
Рисование (В разработке)  
Радиосигналы (В разработке)  
Управление  
Условия  
Циклы  
Математика  
Текст  
Списки  
Переменные  
Функции

color sensor sees черный

датчик цвета интенсивность отраженного света порт Auto

20 оборотов

черный в номер цвета

расстояние с ультразвукового датчика порт Auto в см

расстояние с лазерного дальномера порт Auto в см

гироскоп угол порт Auto

сбросить гироскоп порт Auto

gps X порт Auto

датчик касания порт Auto нажат(а)

датчик касания порт Auto нажат(а)

Центральная кнопка нажат(а)

ждать до Центральная кнопка нажат(а)

capture image with camera порт Auto

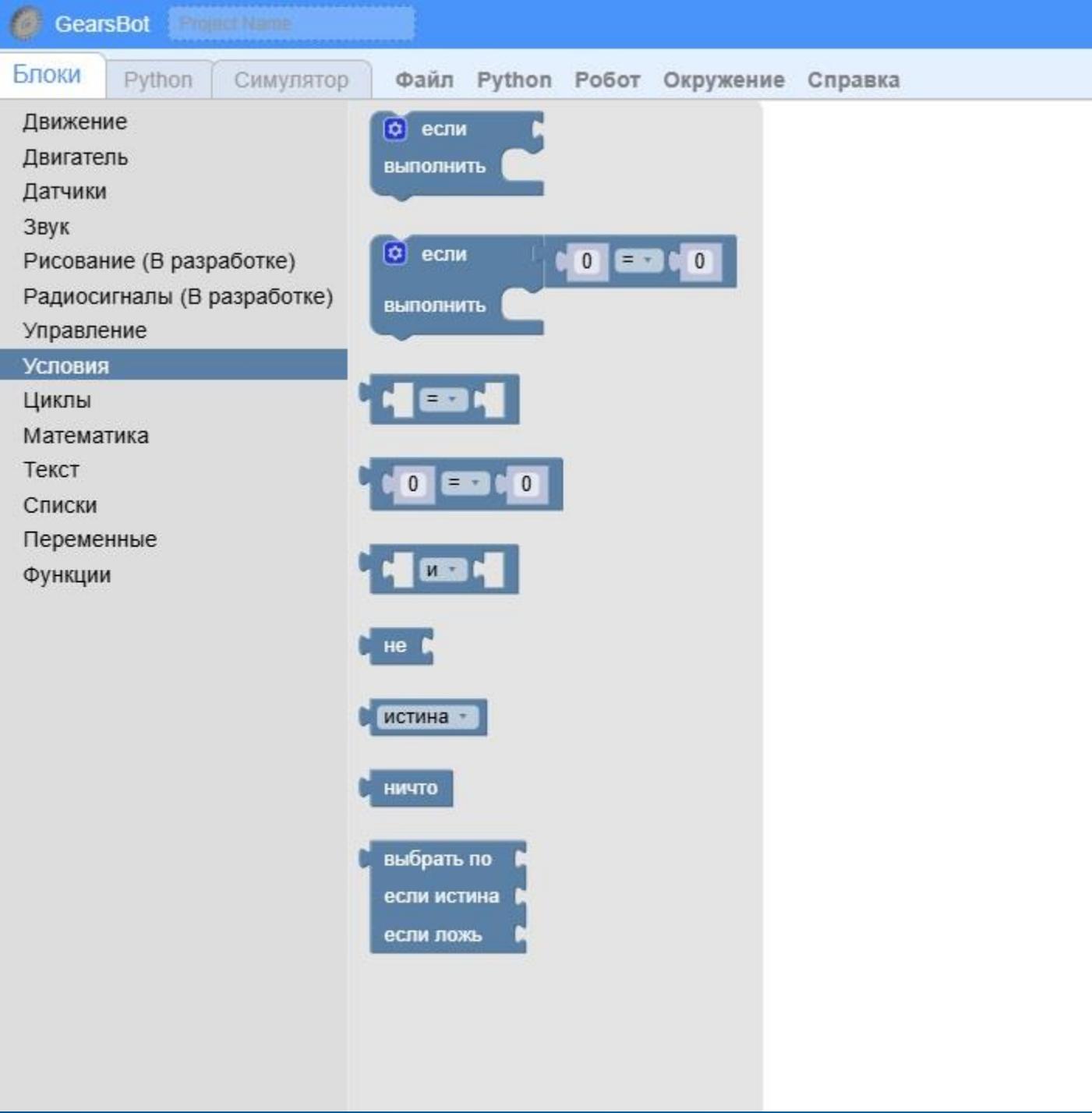
get RGB values from camera порт Auto

get HSV values from camera порт Auto

find blobs from camera порт Auto matching H: 100 140 S: 80 100 V: 20 100 and at least 10 pixels

элементы управления

**В КАЖДОЙ  
ВКЛАДКЕ  
НАБОР КОМАНД  
И  
ИНСТРУМЕНТОВ**



**Можно прописывать условия для дальнейших действий, в зависимости от нужных целей и включенных датчиков**

# ЕСТЬ ВКЛАДКА КОМАНДНОЙ СТРОКИ PYTHON:

```
1 #!/usr/bin/env python3
2
3 # Import the necessary libraries
4 import time
5 import math
6 from ev3dev2.motor import *
7 from ev3dev2.sound import Sound
8 from ev3dev2.button import Button
9 from ev3dev2.sensor import *
10 from ev3dev2.sensor.lego import *
11 from ev3dev2.sensor.virtual import *
12
13 # Create the sensors and motors objects
14 motorA = LargeMotor(OUTPUT_A)
15 motorB = LargeMotor(OUTPUT_B)
16 left_motor = motorA
17 right_motor = motorB
18 tank_drive = MoveTank(OUTPUT_A, OUTPUT_B)
19 steering_drive = MoveSteering(OUTPUT_A, OUTPUT_B)
20
21 spkr = Sound()
22 btn = Button()
23 radio = Radio()
24
25 color_sensor_in1 = ColorSensor(INPUT_1)
26 ultrasonic_sensor_in2 = UltrasonicSensor(INPUT_2)
27 gyro_sensor_in3 = GyroSensor(INPUT_3)
28 gps_sensor_in4 = GPSSensor(INPUT_4)
29 pen_in5 = Pen(INPUT_5)
30
31 motorC = LargeMotor(OUTPUT_C) # Magnet
32
33 # Here is where your code starts
34
35
```

ВТОРАЯ  
ВКЛАДКА  
ВРУЧНУЮ  
+ КОНТРОЛЬ

# МОЖНО ПРОВЕРИТЬ ПОДКЛЮЧЕННЫЕ УСТРОЙСТВА.

ОДНАКО ПРИ ИЗМЕНЕНИИ КОДА ВРУЧНУЮ В  
РУТНОМ - ДАЛЬНЕЙШАЯ РЕДАКЦИЯ КОМАНД  
БЛОКАМИ СТАНОВИТСЯ НЕВОЗМОЖНОЙ

```
9 from ev3dev2.sensor import *
10 from ev3dev2.sensor.lego import *
11 from ev3dev2.sensor.virtual import *
12
13 # Create the sensors and motors objects
14 motorA = LargeMotor(OUTPUT_A)
15 motorB = LargeMotor(OUTPUT_B)
16 left_motor = motorA
17 right_motor = motorB
18 tank_drive = MoveTank(OUTPUT_A, OUTPUT_B)
19 steering_drive = MoveSteering(OUTPUT_A, OUTPUT_B)
20
21 spkr = Sound()
22 btn = Button()
23 radio = Radio()
24
25 color_sensor_in1 = ColorSensor(INPUT_1)
26 ultrasonic_sensor_in2 = UltrasonicSensor(INPUT_2)
27 gyro_sensor_in3 = GyroSensor(INPUT_3)
28 gps_sensor_in4 = GPSSensor(INPUT_4)
29 pen_in5 = Pen(INPUT_5)
30
31 motorC = LargeMotor(OUTPUT_C) # Magnet
```

ПРОВЕРКА  
ПОРТОВ

# ИНТЕРФЕЙС СИМУЛЯТОРА:

Блоки Python Симулятор Файл Python Робот Окружение Справка



Запуск прог.

ручное управление

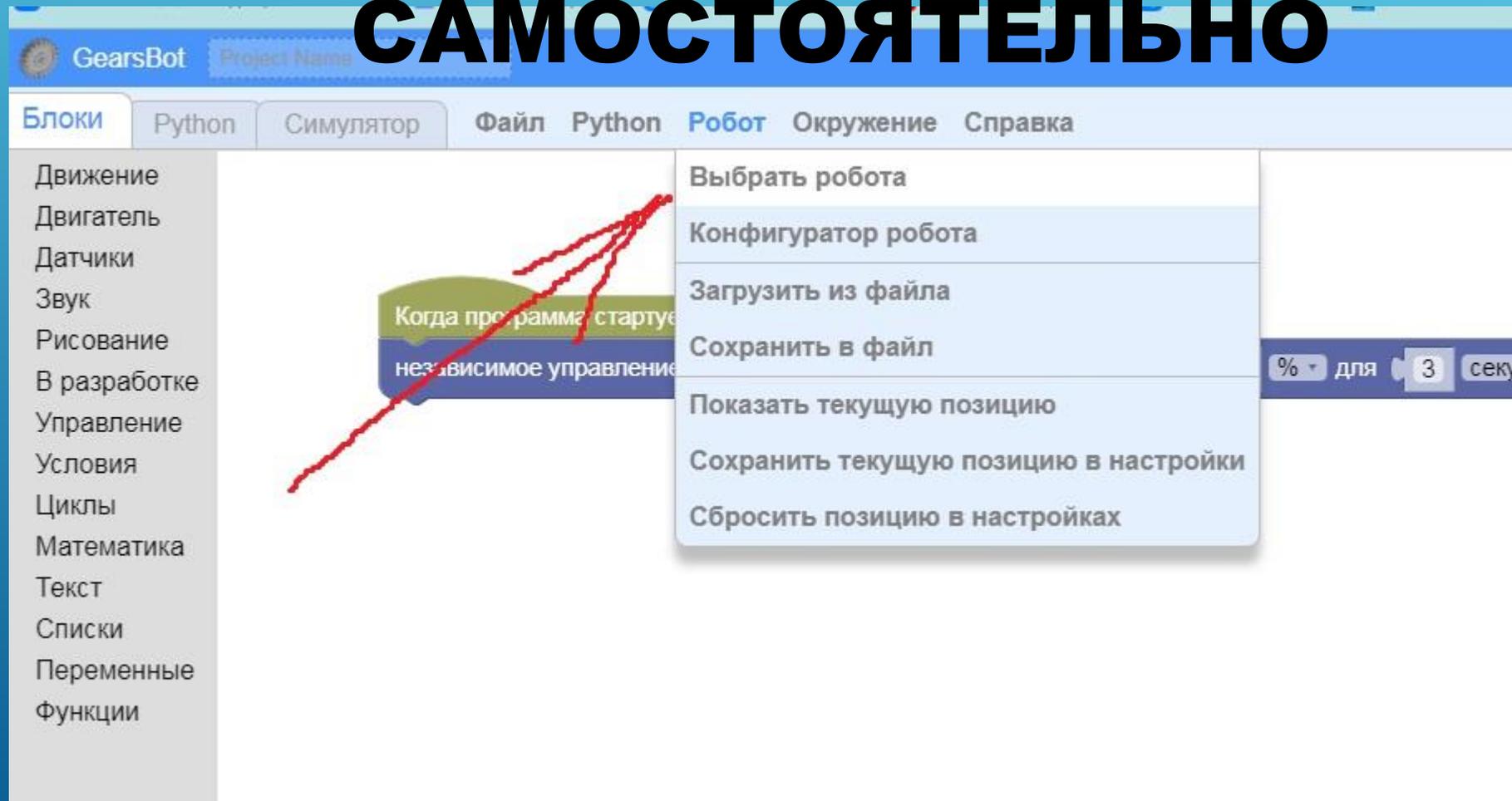
параметры устройств

in1: Датчик цвета	
Цвет	6-White
Красный	162
Зеленый	162
Синий	162
Интенсивность (%)	64
in2: Ультразвуковой датчик	
Расстояние (cm)	16.3
in3: Гироскоп	
Угол (градусы)	63
in4: GPS-датчик	
X (cm)	29.6
Y (cm)	66.5
Высота (cm)	6.3
in5: Перо	
outA: Левый двигатель	
Позиция (градусы)	1937
outB: Правый двигатель	
Позиция (градусы)	1600
outC: Электромагнит	
Сила магнита (%)	0

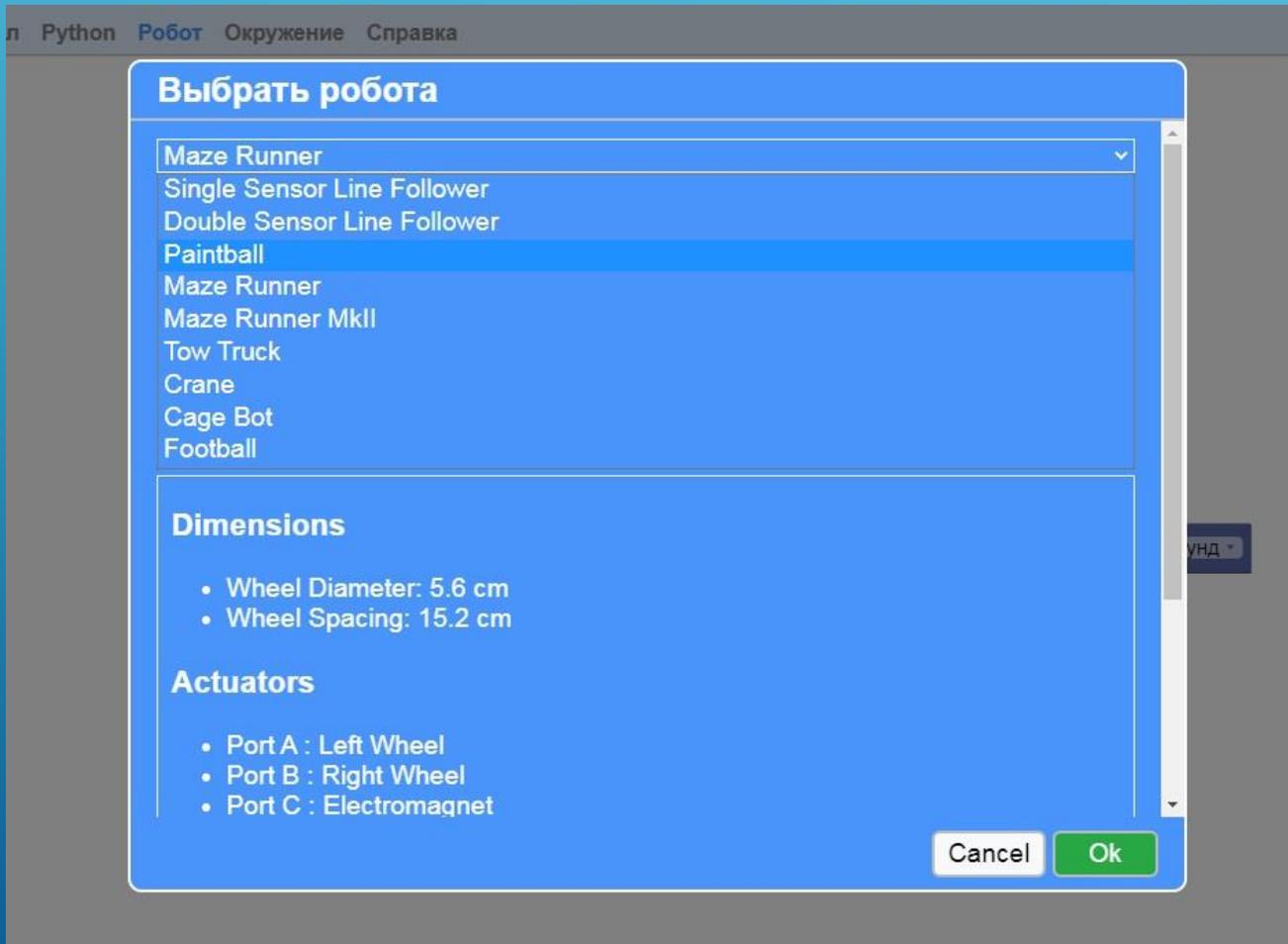
# НАБОР ПОДКЛЮЧЕННЫХ УСТРОЙСТВ, ДВИГАТЕЛЕЙ, ДАТЧИКОВ МОЖНО ВЫБРАТЬ В СЛЕДУЮЩЕЙ ВКЛАДКЕ

The screenshot displays the GearsBot software interface. At the top, there is a header with the GearsBot logo and a 'Project Name' field. Below the header is a navigation bar with tabs: 'Блоки', 'Python', 'Симулятор', 'Файл', 'Python', 'Робот', 'Окружение', and 'Справка'. The 'Робот' tab is selected and highlighted with a red bracket. On the left side, there is a vertical menu with categories: 'Движение', 'Двигатель', 'Датчики', 'Звук', 'Рисование', 'В разработке', 'Управление', 'Условия', 'Циклы', 'Математика', 'Текст', 'Списки', 'Переменные', and 'Функции'. The main workspace shows a block configuration for a robot. The block is titled 'Когда программа стартует' and contains the following configuration: 'независимое управление скорость левого' with a value of '40', 'скорость правого' with a value of '40', a unit selector set to '%', 'для' with a value of '3', and a unit selector set to 'секунд'. Red lines are drawn on the interface, pointing from the 'Робот' tab to the block configuration.

# ЕСТЬ НЕСКОЛЬКО ГОТОВЫХ ВАРИАНТОВ, ЕСТЬ ВОЗМОЖНОСТЬ СМОДУЛИРОВАТЬ САМОСТОЯТЕЛЬНО



# ЗНАНИЕ АНГЛИЙСКОГО НА БАЗОВОМ УРОВНЕ ТЕПЕРЬ ПРИВЕТСТВУЕТСЯ, ИБО ПЕРЕВОДА НЕТ

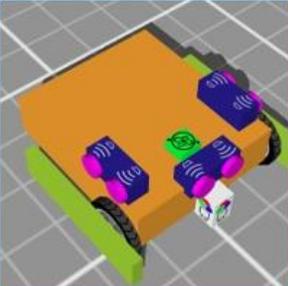


В нижней части  
отображаются  
параметры  
устройств и порты  
их подключения

# ЕСТЬ И КРАТКОЕ ОПИСАНИЕ ДЛЯ ВЫПОЛНЕНИЯ КАКИХ ЗАДАЧ ДАННЫЙ ТИП РОБОТА ЛУЧШЕ ПОДХОДИТ

Выбрать робота

Maze Runner



This robot is equipped with three ultrasonic sensor to facilitate maze navigation, and a color sensor to detect the end point.

An electromagnet at the bottom of the robot lets you pick up magnetic objects, and a gyro helps the robot move straight.

**Dimensions**

- Wheel Diameter: 5.6 cm
- Wheel Spacing: 15.2 cm

**Actuators**

- Port A : Left Wheel
- Port B : Right Wheel
- Port C : Electromagnet

Cancel Ok

Но опять же  
на  
АНГЛИЙСКОМ

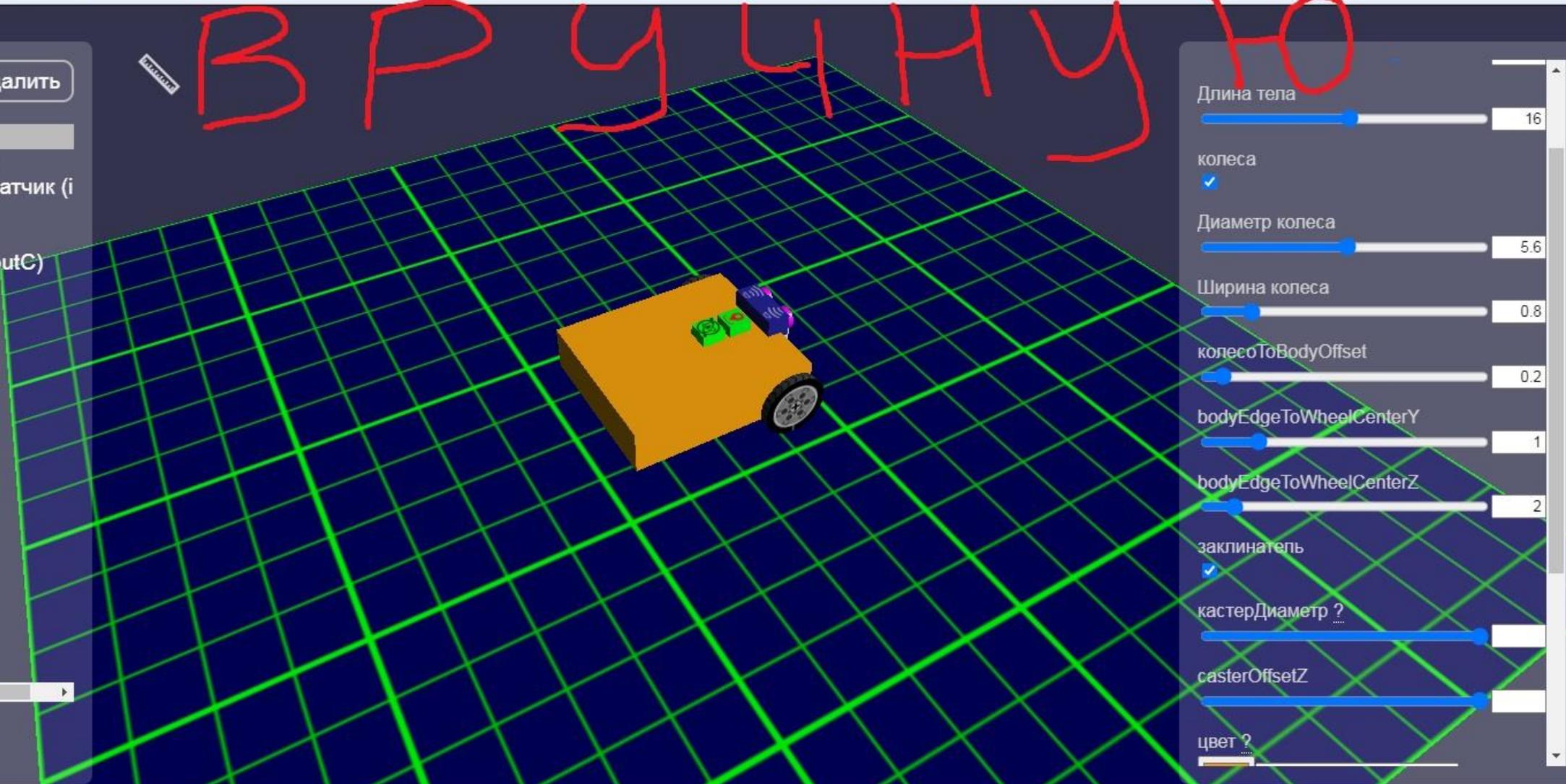
**ПРИ ВЫБОРЕ КОНФИГУРАТОРА РОБОТА  
ВРУЧНУЮ, НЕОБХОДИМ ОПРЕДЕЛЕННЫЙ  
ОПЫТ, ЧТОБЫ СМОДЕЛИРОВАТЬ  
ПОДКЛЮЧЕННЫЕ УСТРОЙСТВА И ИХ  
ПАРАМЕТРЫ В СООТВЕТСТВИИ С  
БУДУЩИМИ ЗАДАЧАМИ**

В Р У Ч И Н У Ю

Добавлять | Удалить

- Тело
- Датчик цвета (v1)
  - Ультразвуковой датчик (i
  - Гиродатчик (in3)
  - GPS-датчик (in4)
  - MagnetActuator (outC)
  - Ручка (in5)

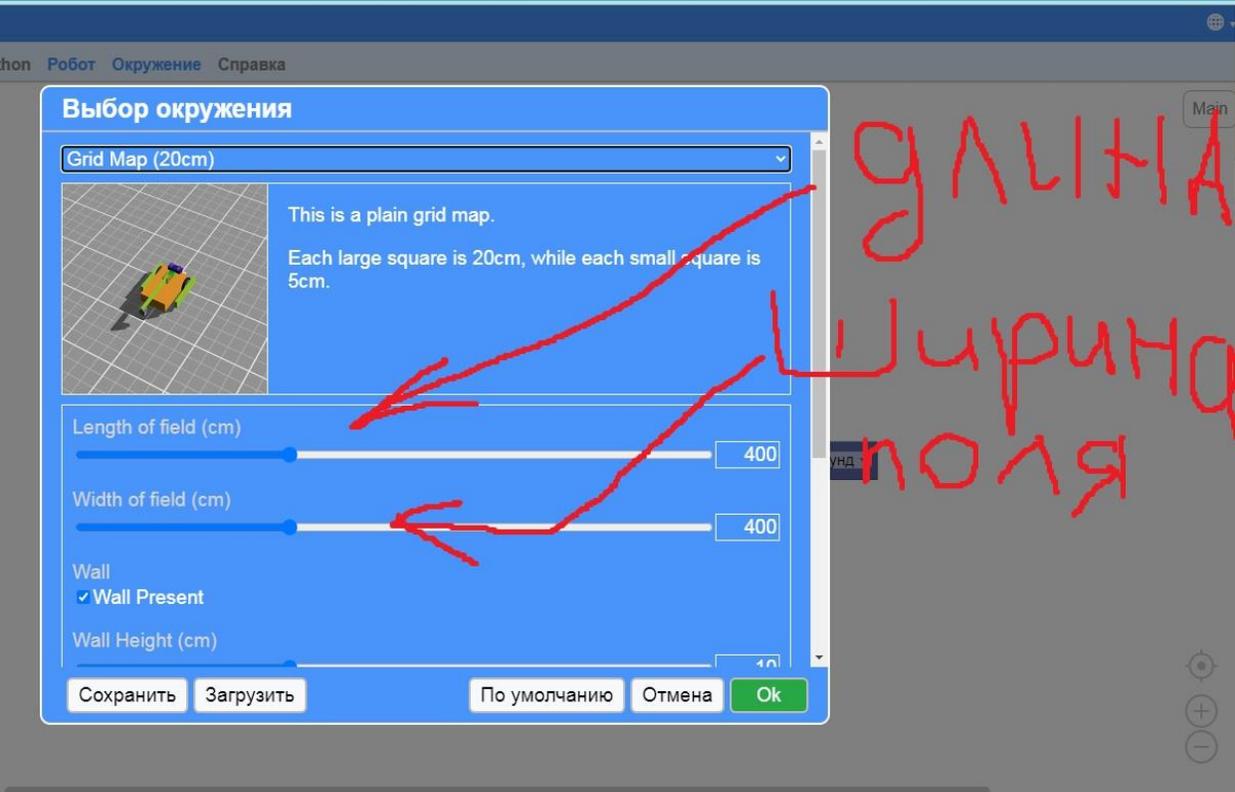
Отменить



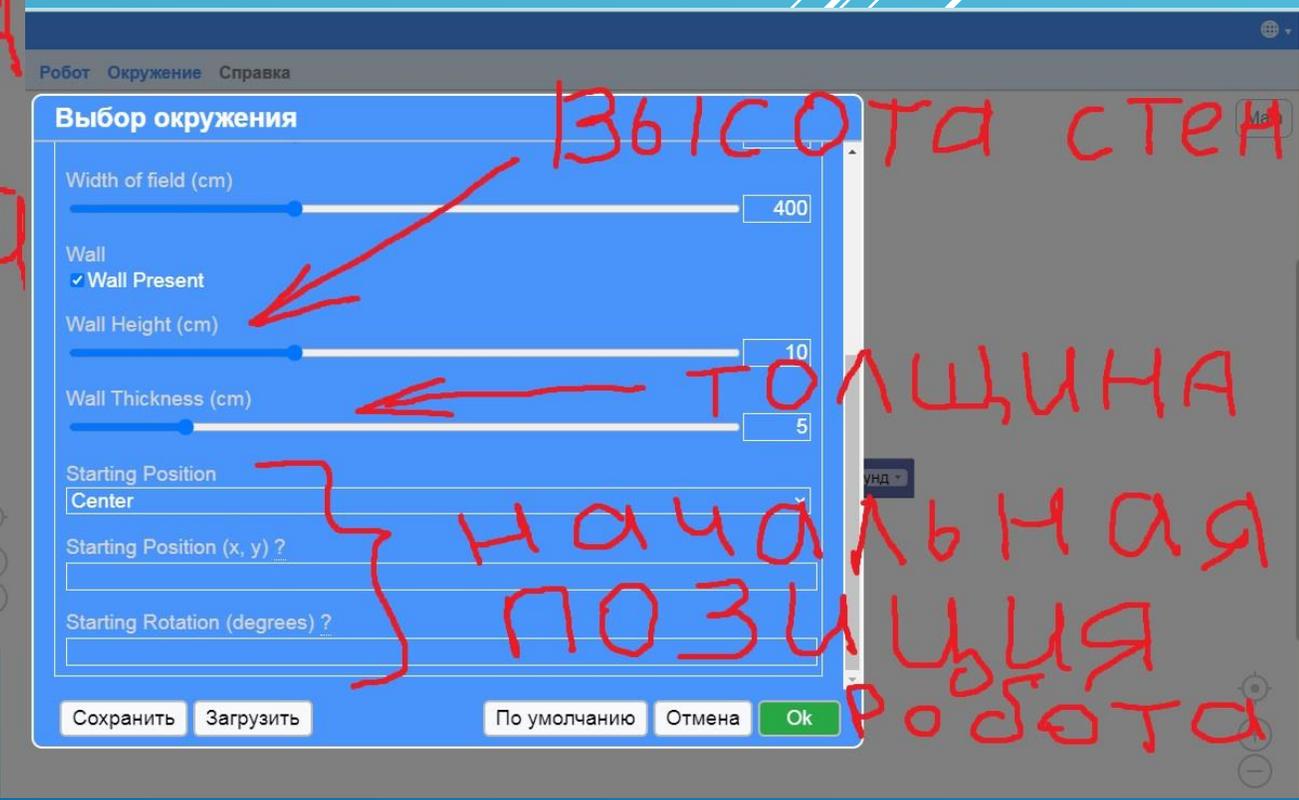
# ПО АНАЛОГИИ МОЖНО ВЫБРАТЬ ГОТОВЫЕ МОДЕЛИ ОКРУЖЕНИЯ ИЛИ СОЗДАТЬ СВОЙ:

The screenshot shows the GearsBot web interface. At the top, there is a browser window with several tabs open, including 'Новости', 'Входящие - smeatc...', 'ДЗ - Google Диск', 'Входящие - Почта...', 'Станислав Родыги...', 'VK Мессенджер', 'Почта - Станисла...', '«Мир танков» - т...', 'Рабочие программ...', and 'Онл O'. Below the browser tabs, the GearsBot logo and a 'Project Name' input field are visible. The main navigation bar includes 'Блоки', 'Python', 'Симулятор', 'Файл', 'Python', 'Робот', 'Окружение', and 'Справка'. A dropdown menu is open under 'Окружение', listing options: 'Выбор окружения', 'Создать окружение', 'Арена', 'Загрузить из файла', and 'Сохранить в файл'. To the right of the menu, there are three red arrows pointing downwards. On the left side, a vertical menu lists various categories: 'Движение', 'Двигатель', 'Датчики', 'Звук', 'Рисование', 'В разработке', 'Управление', 'Условия', 'Циклы', 'Математика', 'Текст', 'Списки', 'Переменные', and 'Функции'. At the bottom, a code block is shown with the following text: 'Когда программа стартует' followed by 'независимое управление скорость левого' with a value of '40', 'скорость правого' with a value of '40', a percentage symbol, 'для' with a value of '3', and 'секунд'.

# НАСТРАИВАЕМЫЕ ПАРАМЕТРЫ:



длина  
ширина  
поля

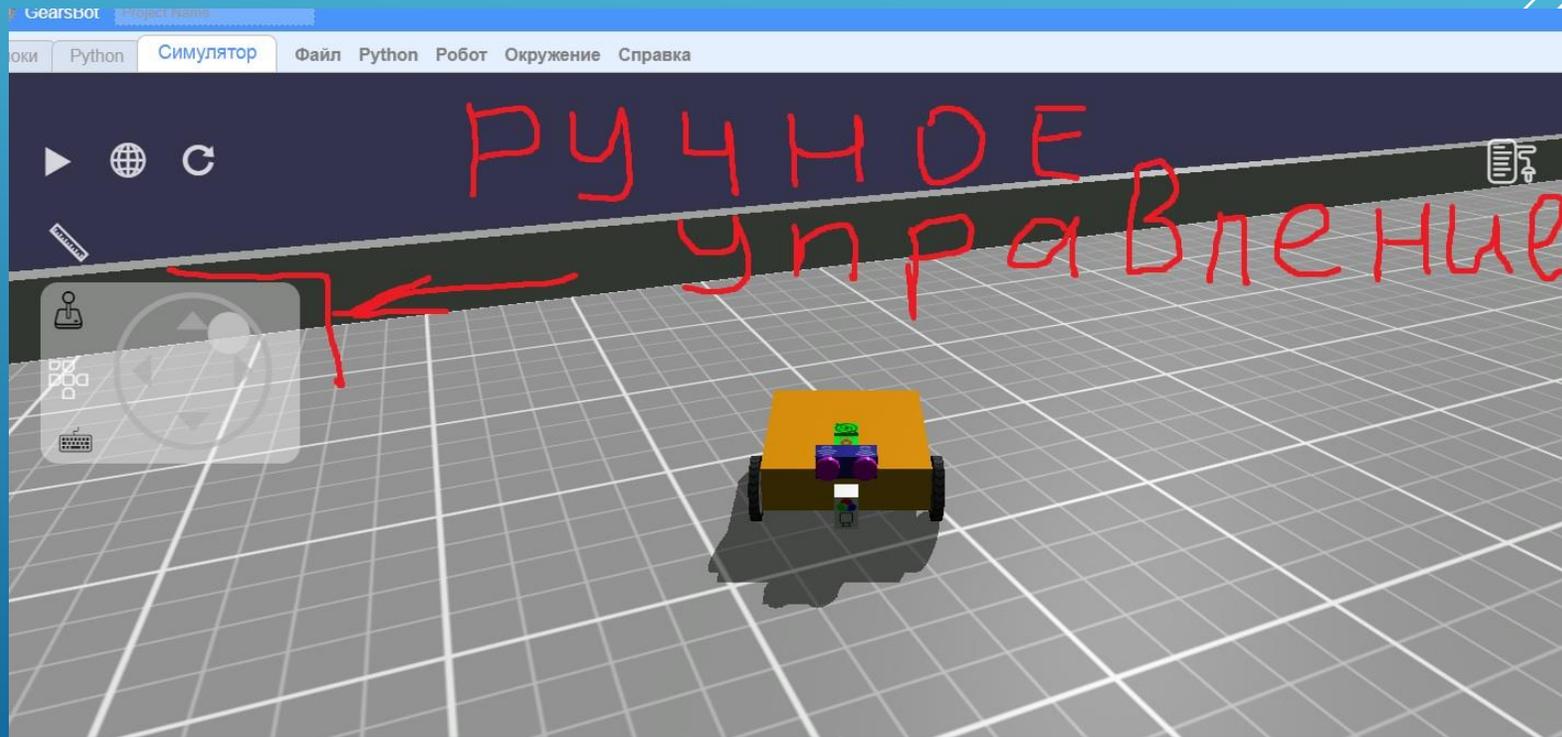


высота стен

толщина

начальная  
позиция  
робота

**РЕСУРС АВТОМАТИЧЕСКИ СОХРАНЯЕТ  
ВСЕ ИЗМЕНЕНИЯ. ОДНАКО ТОЛЬКО ДЛЯ  
ТОГО УСТРОЙСТВА(КОМПЬЮТЕРА) С  
КОТОРОГО ЗАХОДИЛИ**



# ПРИ РАБОТЕ В НЕСКОЛЬКИХ ВАРИАНТАХ РАБОТА И ОКРУЖЕНИЯ МОЖНО СОХРАНЯТЬ НУЖНЫЕ ПАРАМЕТРЫ:

Симулятор

Файл Python Робот Окружение Справка

Новая программа

Загрузить блок-программу с компьютера

Импортировать функции из блок-программы

**Сохранить блок-программу на компьютер**

Загрузить Python-код с компьютера

Сохранить Python-код на компьютер

Экспортировать zip-архив

независимое управление скорость левого 20 скорость правого 20 %

запустить двигатель порт A скорость 50 % до позиции 45 градусов и ждать выполнения

запустить двигатель порт B скорость 50 % до позиции 90 градусов и ждать выполнения

независимое управление скорость левого 30 скорость правого 30 % для 3 оборотов

рулевое управление по направлению 100 со скоростью 50 % для 2 оборотов

рулевое управление по направлению -100 со скоростью 20 % для 1 оборотов

остановить программу

Когда прог  
повторить  
выполнить

независимое управ

**СОХРАНЕНИЕ**

# ПРИ РАБОТЕ В НЕСКОЛЬКИХ ВАРИАНТАХ РАБОТА И ОКРУЖЕНИЯ МОЖНО СОХРАНЯТЬ НУЖНЫЕ ПАРАМЕТРЫ:

The image displays a software interface with several components:

- File Explorer (Left):** A window titled "Сохранить как" (Save As) showing the "ГМО на" folder. The search results are empty. The file name is "gridMap\_config.json" and the type is "JSON-файл (\*.json)".
- File Explorer (Center):** A window titled "Сохранить как" (Save As) showing the "gears bot" folder. The search results show a file named "gearsBot.xml" with a size of 6 KB. The file name is "gearsBot (1).xml" and the type is "XML Document (\*.xml)".
- Block-based Programming (Right):** A vertical stack of blue blocks. The top block is a percentage block. Below it are several "для" (for) blocks with values like "2 секунд", "1 оборотов", "4 оборотов", "3 секунд", and "3 оборотов". These are followed by "ждать выполнения" (wait for completion) blocks.
- Buttons (Bottom):** A green button labeled "остановить программу" (stop program) and a dialog box with the text "Starting Rotation (degrees) ?" and buttons "Сохранить" (save) and "Загрузить" (load).

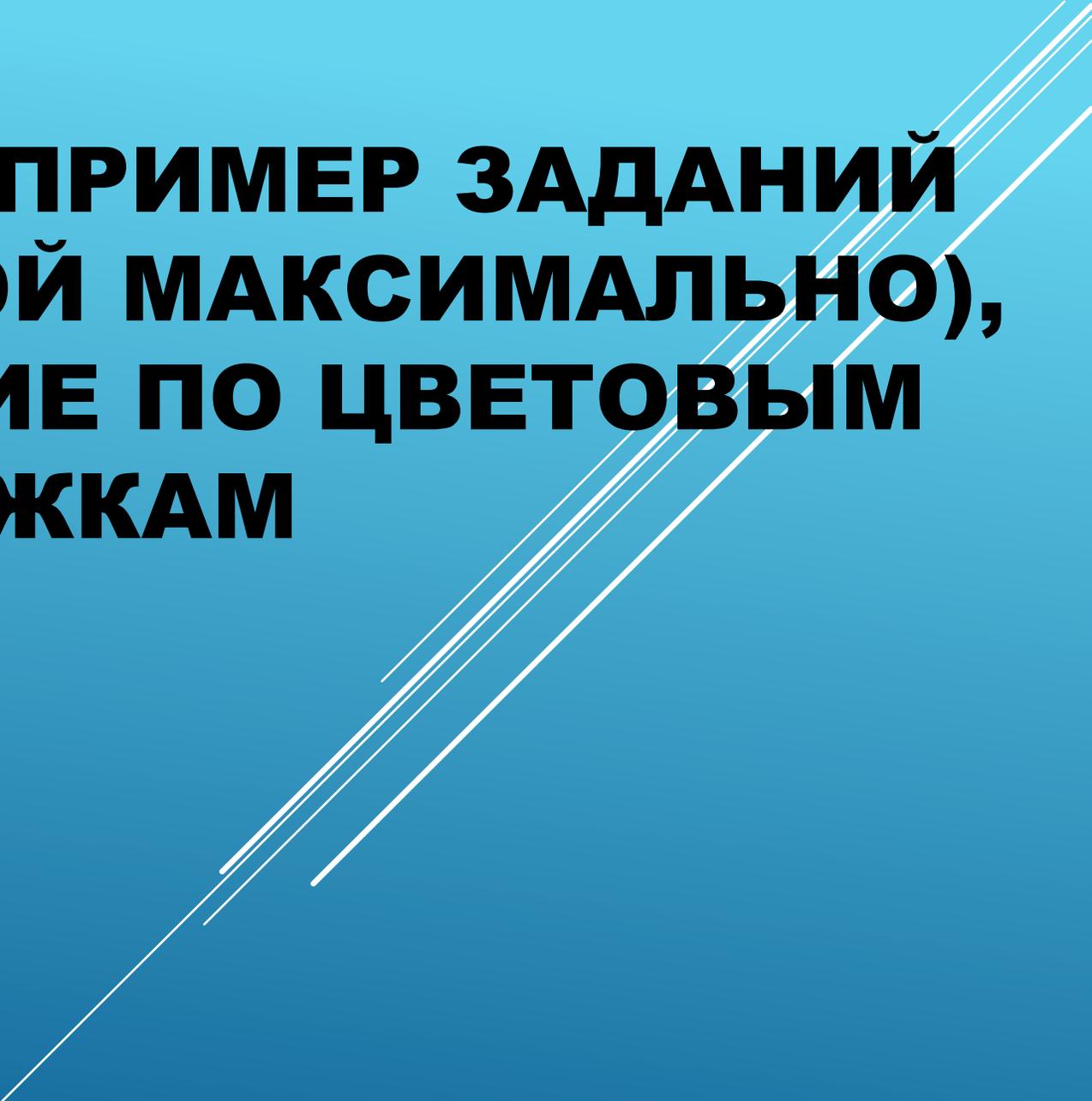
**ЧТО В ТОМ ЧИСЛЕ МОЖНО  
ИСПОЛЬЗОВАТЬ ДЛЯ ЗАДАНИЯ  
УЧЕНИКАМ НА УРОКАХ, ИЛИ В  
РАМКАХ ДОМАШНЕГО ЗАДАНИЯ,  
ЛИБО ДИСТАНЦИОННОГО ОБУЧЕНИЯ**

**То есть выполненную работу ученики  
могут предоставить в виде файла,  
который легко подгружается**

**В ПЕРСПЕКТИВЕ, УЧЕНИКОВ МОЖНО  
ДЕЛИТЬ НА БОЛЬШИЕ ГРУППЫ,  
ПРЕДВАРИТЕЛЬНО РАЗРАБОТАВ (В  
ИДЕАЛЕ ВМЕСТЕ С НИМИ) ТРЕБОВАНИЯ  
К СОЗДАНИЮ ОКРУЖЕНИЯ И ЗАДАНИЙ  
ДЛЯ РОБОТОВ.**

**НА ПЕРВОМ УРОКЕ СОЗДАЮТ ЗАДАНИЯ,  
НА ВТОРОМ ДАЮТ ИХ ДРУГОЙ ГРУППЕ  
ДЛЯ ВЫПОЛНЕНИЯ, САМИ БЕРУТ  
ЗАДАНИЕ ОТ НИХ.**

**САМЫЙ ПРОСТОЙ ПРИМЕР ЗАДАНИЙ  
ЛАБИРИНТ(ПРОСТОЙ МАКСИМАЛЬНО),  
ЛИБО СЛЕДОВАНИЕ ПО ЦВЕТОВЫМ  
ДОРОЖКАМ**

A decorative graphic consisting of several parallel white lines of varying lengths, slanted diagonally from the bottom-left towards the top-right, located in the lower right quadrant of the slide.

# **ВАРИАНТ ЗАДАНИЯ ДЛЯ УЧЕНИКОВ НА «ОЗНАКОМИТЕЛЬНОМ» ЭТАПЕ:**

**Разработать алгоритм движения робота,  
включающий следующие требования:**

- 1. Программа должна выполняться**
- 2. Движение по прямой вперед и назад, повороты**
- 3. Цикличность**
- 4. Применение разных единиц измерения:  
секунды, градусы, обороты колеса**
- 5. Минимум 10 команд для робота**

**ОЦЕНИВАНИЕ МОЖНО ПРОВОДИТЬ  
ПРОСТО ПО ТОМУ, СКОЛЬКО  
ПУНКТОВ ТРЕБОВАНИЙ**

**СОБЛЮДЕНО:**

**5 – ОТЛИЧНО,**

**4 – ХОРОШО...**

# ВАРИАНТ ВЫПОЛНЕНИЯ ЗАДАНИЯ БЕЗ ЦИКЛА

The screenshot shows the GearsBot software interface with a sequence of blocks for controlling two motors. The blocks are as follows:

- Когда программа стартует** (When program starts)
- независимое управление скоростью левого** (independent speed control left) **40** **скоростью правого** (speed right) **40** **%** для **2** **секунд** (seconds)
- независимое управление скоростью левого** (independent speed control left) **20** **скоростью правого** (speed right) **-20** **%** для **1** **оборотов** (revolutions)
- независимое управление скоростью левого** (independent speed control left) **70** **скоростью правого** (speed right) **70** **%** для **4** **оборотов** (revolutions)
- независимое управление скоростью левого** (independent speed control left) **-30** **скоростью правого** (speed right) **-30** **%** для **3** **секунд** (seconds)
- независимое управление скоростью левого** (independent speed control left) **0** **скоростью правого** (speed right) **50** **%** для **2** **оборотов** (revolutions)
- запустить двигатель порт A** (start motor port A) **скоростью** (speed) **50** **%** до **позиции** (position) **45** **градусов** (degrees) и **ждать выполнения** (wait for completion)
- запустить двигатель порт B** (start motor port B) **скоростью** (speed) **50** **%** до **позиции** (position) **90** **градусов** (degrees) и **ждать выполнения** (wait for completion)
- независимое управление скоростью левого** (independent speed control left) **30** **скоростью правого** (speed right) **30** **%** для **3** **оборотов** (revolutions)
- рулевое управление по направлению** (steering control by direction) **100** **со скоростью** (with speed) **50** **%** для **2** **оборотов** (revolutions)
- рулевое управление по направлению** (steering control by direction) **-100** **со скоростью** (with speed) **20** **%** для **1** **оборотов** (revolutions)

направление принимает значения от -100 (влево) до 0 (прямо) и до 100 (вправо)

# ВАРИАНТ С ЦИКЛОМ:

GearsBot Project Name

Блоки Python Симулятор Файл Python Робот Окружение Справка

Движение  
Двигатель  
Датчики  
Звук  
Рисование  
В разработке  
Управление  
Условия  
**Циклы**  
Математика  
Текст  
Списки  
Переменные  
Функции

повторить 10 раз  
выполнить

повторять, пока истина  
выполнить

цикл по i от 1 до 10 с шагом 1  
выполнить

для каждого элемента j в списке  
выполнить

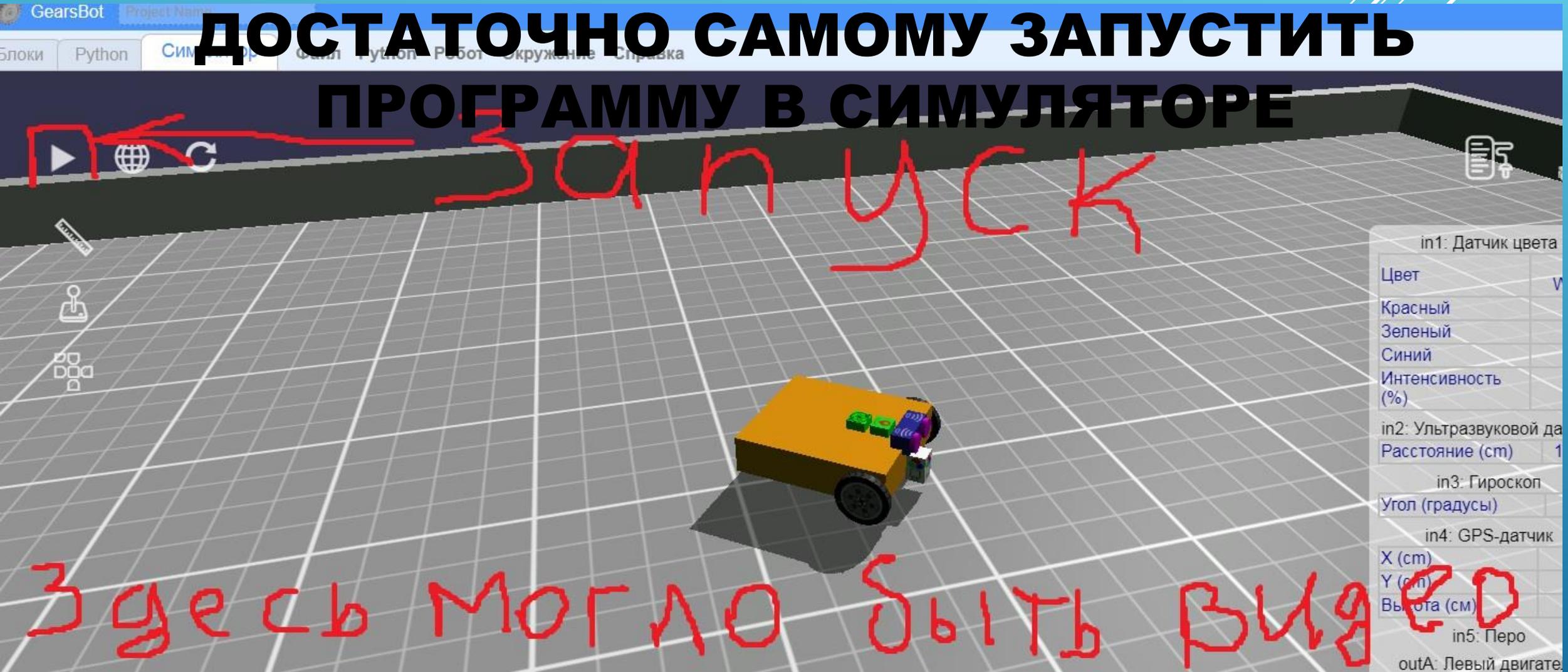
выйти из цикла

Когда программа стартует

повторить 2 раз  
выполнить

- независимое управление скорость левого 40 скорость правого 40 % для 2 секунд
- независимое управление скорость левого 20 скорость правого -20 % для 1 оборотов
- независимое управление скорость левого 70 скорость правого 70 % для 4 оборотов
- независимое управление скорость левого -30 скорость правого -30 % для 3 секунд
- независимое управление скорость левого 0 скорость правого 50 % для 2 оборотов
- запустить двигатель порт A скорость 50 % до позиции 45 градусов и ждать выполнения
- запустить двигатель порт B скорость 50 % до позиции 90 градусов и ждать выполнения
- независимое управление скорость левого 30 скорость правого 30 % для 3 оборотов
- рулевое управление по направлению 100 со скоростью 50 % для 2 оборотов
- рулевое управление по направлению -100 со скоростью 20 % для 1 оборотов

**ЧТО НЕМАЛОВАЖНО УЧЕНИК НА КАЖДОМ  
ЭТАПЕ МОЖЕТ САМ ПРОВЕРЯТЬ СЕБЯ, А В  
ДАЛЬНЕЙШЕМ И ОЦЕНИВАТЬ. ЕМУ  
ДОСТАТОЧНО САМОМУ ЗАПУСТИТЬ  
ПРОГРАММУ В СИМУЛЯТОРЕ**



# ПОСТЕПЕННО В ЗАДАНИЯ МОЖНО ДОБАВЛЯТЬ ПРИМЕНЕНИЕ ДАТЧИКОВ:

GearsBot gearsBot

Блоки Python Симулятор Файл Python Робот Окружение Справка

Движение  
Двигатель  
Датчики  
Звук  
Рисование  
В разработке  
Управление  
Условия  
Циклы  
Математика  
Текст  
Списки  
Переменные  
Функции

Когда программа стартует

независимое управление скорость левого 40 скорость правого 40 % для 3 секунд

рулевое управление по направлению 100 со скоростью 20 % для 2 оборотов

выключить двигатели и затормозить

независимое управление скорость левого 50 скорость правого 50 %

если

расстояние с ультразвукового датчика порт 2 в cm = 30

выполнить

выключить двигатели и brake

независимое управление скорость левого -30 скорость правого -30 % для 2 оборотов

рулевое управление по направлению 100 со скоростью 20 % для 90 градусов

независимое управление скорость левого 20 скорость правого 20 % для 2 секунд

остановить программу

**НА БАЗЕ ТАКИХ ЗАДАНИЙ МОЖНО  
СОСТАВЛЯТЬ И ЗАДАНИЯ ПО  
НАХОЖДЕНИЮ ОШИБОК В ГОТОВЫХ  
ПРОГРАММАХ.  
ЛИБО В ВИДЕ МИНИ ТЕСТОВ, ЛИБО С  
ПРЕДЛОЖЕНИЕМ ВАРИАНТА  
ИСПРАВЛЕНИЯ В GEARS BOT**

# ПРИМЕР: НАЙТИ КОМАНДУ, МЕШАЮЩУЮ ВЫПОЛНЕНИЮ ЦИКЛА

Блоки Python Симулятор

Движение  
Двигатель  
Датчики  
Звук  
Рисование  
В разработке  
Управление  
Условия  
Циклы  
Математика  
Текст  
Списки  
Переменные  
Функции

Когда программа стартует

повторить 2 раз

выполнить

- независимое управление скорость левого 40 скорость правого 40 % для 2 секунд
- независимое управление скорость левого 20 скорость правого -20 % для 1 оборотов
- независимое управление скорость левого 70 скорость правого 70 % для 4 оборотов
- независимое управление скорость левого -30 скорость правого -30 % для 3 секунд
- независимое управление скорость левого 20 скорость правого 20 %
- запустить двигатель порт A скорость 50 % до позиции 45 градусов и ждать выполнения
- запустить двигатель порт B скорость 50 % до позиции 90 градусов и ждать выполнения
- независимое управление скорость левого 30 скорость правого 30 % для 3 оборотов
- рулевое управление по направлению 100 со скоростью 50 % для 2 оборотов
- рулевое управление по направлению -100 со скоростью 20 % для 1 оборотов

остановить программу

# ДАННАЯ КОМАНДА ЗАПУСКАЕТ РОБОТА ЕХАТЬ ВПЕРЕД БЕЗ КАКИХ-ЛИБО УСЛОВИЙ(ВРЕМЯ, ОБОРОТЫ КОЛЕС, СРАБАТЫВАНИЕ ДАТЧИКОВ). ГРУБО ГОВОРЯ БЕСКОНЕЧНОЕ ДВИЖЕНИЕ ПО НАПРАВЛЕНИЮ

Когда программа стартует

повторить 2 раз

выполнить

независимое управление скорость левого 40 скорость правого 40 % для 2 секунд

независимое управление скорость левого 20 скорость правого -20 % для 1 оборотов

независимое управление скорость левого 70 скорость правого 70 % для 4 оборотов

независимое управление скорость левого -30 скорость правого -30 % для 3 секунд

независимое управление скорость левого 20 скорость правого 20 %

запустить двигатель порт A скорость 50 % до позиции 45 градусов и ждать выполнения

запустить двигатель порт B скорость 50 % до позиции 90 градусов и ждать выполнения

независимое управление скорость левого 30 скорость правого 30 % для 3 оборотов

рулевое управление по направлению 100 со скоростью 50 % для 2 оборотов

рулевое управление по направлению -100 со скоростью 20 % для 1 оборотов

остановить программу

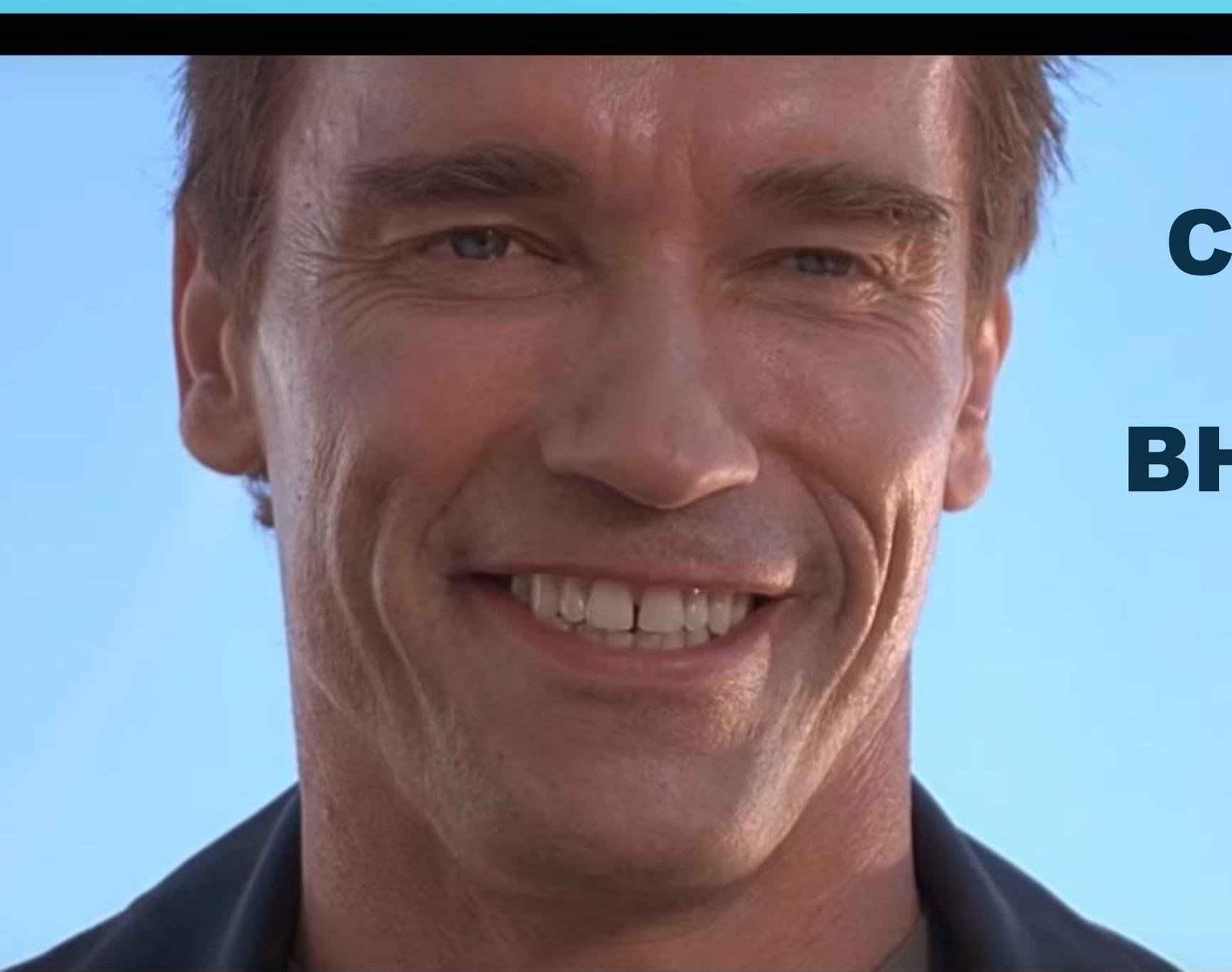
независимое управление скорость левого 0

ошибка

**РЕСУРС GEARS VOT ДАЕТ  
ВОЗМОЖНОСТЬ ДЛЯ ФОРМИРОВАНИЯ  
ПРАКТИЧЕСКИХ ЗАДАНИЙ ДЛЯ  
УЧЕНИКОВ РАЗНОЙ СЛОЖНОСТИ, В  
ТОМ ЧИСЛЕ ДЛЯ ГРУППОВЫХ И  
ТВОРЧЕСКИХ РАБОТ,  
ИНДИВИДУАЛЬНЫХ ПРОЕКТОВ  
(СОЗДАНИЕ УНИКАЛЬНЫХ ОКРУЖЕНИЙ  
И ЗАДАЧ С ВАРИАНТАМИ ИХ РЕШЕНИЯ  
САМИМИ УЧЕНИКАМИ)**

**ТАКИМ ОБРАЗОМ, РЕСУРС GEARS  
ВОТ ВПОЛНЕ ПОДХОДИТ ДЛЯ  
ИЗУЧЕНИЯ МОДУЛЕЙ, СВЯЗАННЫХ С  
РОБОТОТЕХНИКОЙ В РАМКАХ  
ПРЕДМЕТА ТРУД (ТЕХНОЛОГИЯ)**

**В ПЕРСПЕКТИВЕ ПЛАНИРУЮ  
РАЗРАБОТАТЬ МИНИ-КОМПЛЕКТ  
ЗАДАНИЙ РАЗНОГО УРОВНЯ  
СЛОЖНОСТИ С ПРИМЕНЕНИЕМ  
РЕСУРСА GEARS BOT И  
ПРОВЕДЕНИЕМ МАСТЕР-КЛАССА (ПО  
ЗАПРОСАМ ТРУДЯЩИХСЯ)**



**СПАСИБО  
ЗА  
ВНИМАНИЕ**