Оглавление

Пояснительная записка

Данная пояснительная записка к учебному курсу «Робототехника» составлена в соответствии с положениями **Инструктивно-методического письма (ИМП) на 2025-2026 учебный год**, подготовленного Национальной академией образования им. И. Алтынсарина. Программа базируется на реализации Государственного общеобязательного стандарта образования (ГОСО, Приказ № 348) и типовых учебных программ (Приказ № 399), ориентируясь на достижение конкретных целей обучения, таких как конструирование роботов (5.3.4.2), программирование поворотов и работы датчиков (5.3.3.2, 5.3.3.4), а также проектирование систем «умного дома» и «интернета вещей» (11.3.4.8, 11.5.2.4).

Согласно приоритетам ИМП на 2025-2026 год, курс направлен на развитие **проектного мышления, научно-исследовательской деятельности и цифровых компетенций** обучающихся. Ключевой особенностью нового учебного года является **интеграция искусственного интеллекта (ИИ)** в учебный процесс: использование ИИ-инструментов (ChatGPT, Perplexity, Tinkercad с элементами ИИ) для генерации идей, моделирования и анализа данных при создании робототехнических систем. Воспитательный компонент программы реализуется через единую программу **«Адал азамат»**, формируя у учащихся через инженерное творчество такие ценности, как трудолюбие, честность и созидательное новаторство.

Методология обучения опирается на **STEAM-подход** и использование виртуальных лабораторий (**Open Roberta Lab, Tinkercad**), что обеспечивает формирование функциональной грамотности в условиях цифровой трансформации. Содержательная часть программы синхронизирована с актуальными направлениями **международного олимпиадного движения**, включая регламенты **WRO (World Robot Olympiad)**, **FIRST LEGO League** и международного фестиваля **RoboLand**, охватывая такие дисциплины, как «Робо-сумо», «Кегельринг» и «Биатлон». Это позволяет подготовить конкурентоспособную личность, обладающую навыками критического анализа и способную применять теоретические знания математики и физики для решения прикладных инженерных задач.

Представленные материалы представляют собой учебное пособие по робототехнике и программированию для учащихся 5-х классов, ориентированное на практическое использование симуляторов. Основное внимание уделяется платформе Open Roberta Lab, где школьники учатся управлять виртуальными роботами, осваивая движение по заданным траекториям, геометрию и алгоритмы объезда препятствий. Текст подробно описывает работу с датчиками цвета, ультразвука и гироскопами, а также знакомит с базовыми логическими операциями и арифметическими вычислениями. Отдельные разделы посвящены интернету вещей (IoT) и основам электроники в среде Tinkercad, включая сборку схем на базе Arduino. Для закрепления знаний предлагаются логические задачи и математические упражнения, связанные с навигацией и физикой движения. В целом, пособие направлено на развитие навыков STEAM-образования через программирование автономных систем и моделирование «умных» устройств.

Тема. Робототехника. Примеры обучающих программ. Датчики

Цели обучения (ОШ):
5.3.4.2 приводить примеры разновидностей роботов и области их применения; 5.3.3.2 создавать программы для поворота робота на заданные градусы; 5.3.3.4 использовать датчик ультразвука для нахождения объекта;

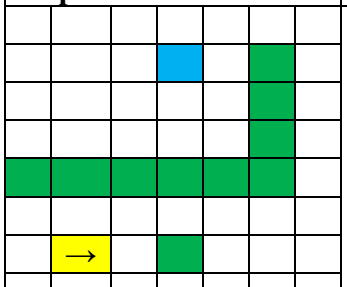
Задание 1. Используйте ресурсы Интернет для сбора, систематизации и визуализации информации (инфографика, визуальный органайзер, постер, презентация). Обсудите ответы в группе и ответьте на следующие вопросы:

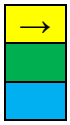
- «Что такое датчик? Где можно применить датчики?»;
- «Как и где в естественной среде обитания используется сенсоры и датчики?»;

Задание 2. Используйте текстовые команды из списка предложенных и опишите последовательности действий для достижения «роботом» (без датчиков) поставленной цели. Какое количество клеток пройдет ваш «робот»?

Список используемых текстовых команд:

- «Идти вперед»
- «Повторить»
- «Повернуть на 90 градусов вправо»
- «Повернуть на 90 градусов влево»

Игровое поле	**Действия (алгоритм)**
	
Количество пройденных клеток = ?	

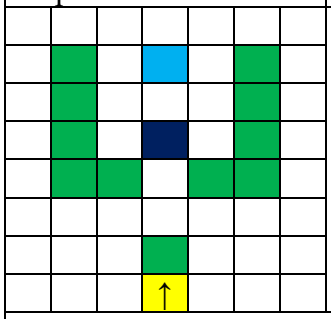


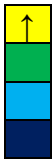
Робот (начальное направление)
Препятствие
Цель

Задание 4. Ознакомьтесь с правилами мини-игры:

1. Доставить артефакт до базы, указанной на карте;
2. Место положение артефакта не меняется;
3. В наборе действий может быть не более 10 команд;
4. В каждой команде не более 2 слов;
5. У робота отсутствует зрение.

Сформулируйте точные текстовые команды (алгоритм) для достижения «роботом» поставленной цели.

Игровое поле	Действия (алгоритм)
	
Количество пройденных клеток = ?	



Задание 3. «Система управления теплицей. Автоматический полив и освещение».
Используйте сервис https://logic.ly/demo для разработки логической схемы для представления системы управления теплицей.

Система управления теплицами имеет четыре входных параметра (H,D,T,W). По умолчанию состояние каждого параметра – это ИСТИНА (1).

Параметр	Описание параметра	Бинарное значение	Состояние
H	Влажность	0	Слишком низко
		1	Приемлемый
D	Время суток	0	Ночь
		1	День
T	Температура	0	Слишком высокая
		1	Приемлемая
R	Реле управления освещением	0	Разомкнуто
		1	Замкнуто

Система полива включается (X=1), если:

- либо если это дневное время и температура приемлемая;
- или влажность слишком низкая.

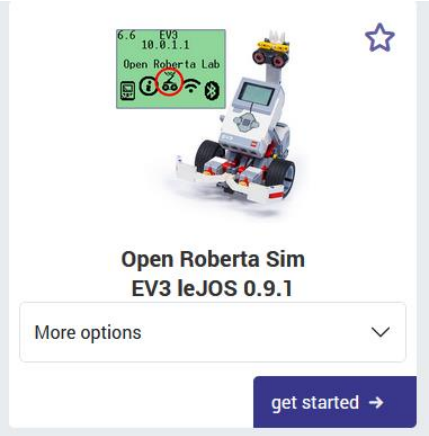
Свет включается (Y=1), если

- время суток ночь или реле управления замкнуто;
- и система полива включена;

Тема. Геометрия.

Задание 1. Треугольник.

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Используйте для изменения сцены элемент управления, указанный стрелкой на рисунке 1 «Элементы управления». Нажмите левой кнопкой мыши несколько раз чтобы загрузить сцену с координатами, а затем измените положение робота, как на рисунке 2 «Сцена».

Рисунок 1. Элементы управления

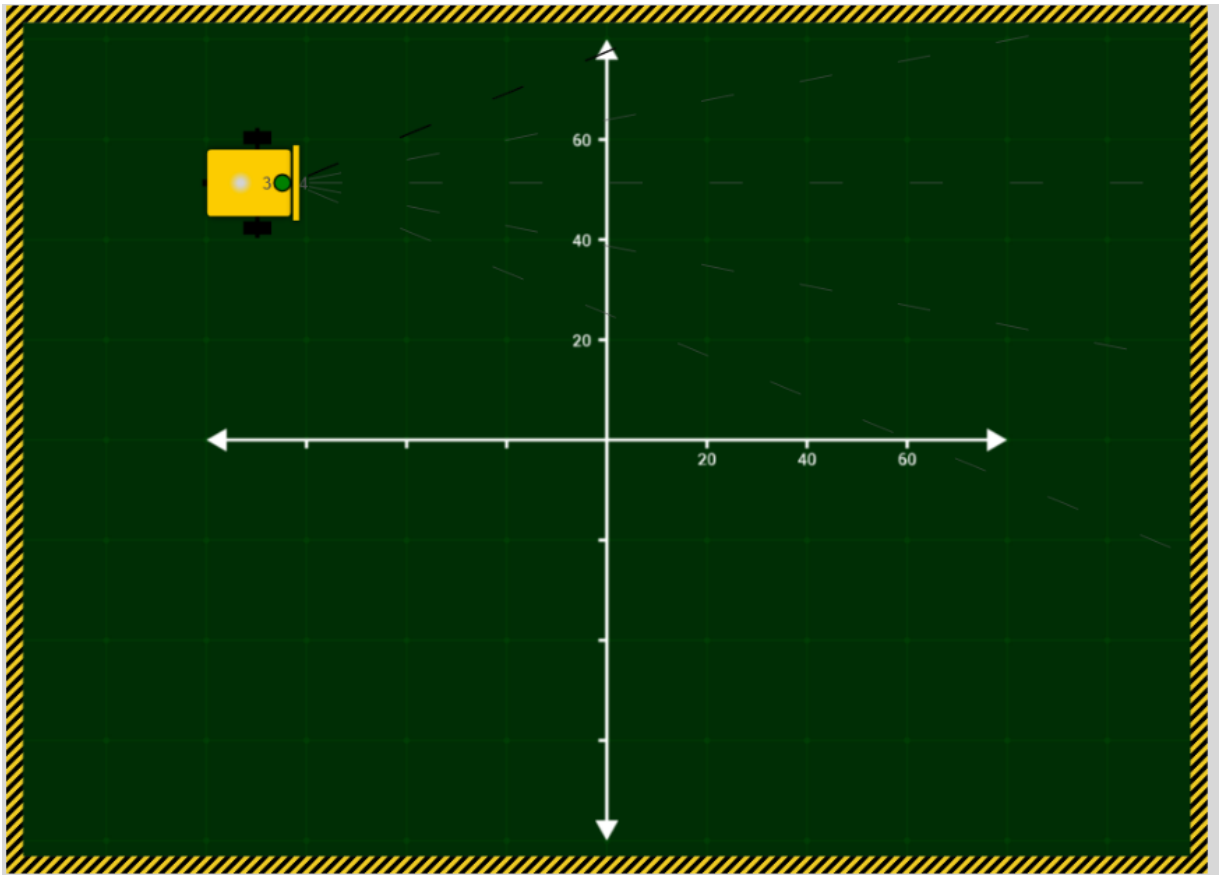


Рисунок 2. Сцена

Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Треугольник». Используйте следующие разделы «Действие», «Датчики», «Контроль», «Логика», «Математика» и «Переменные».

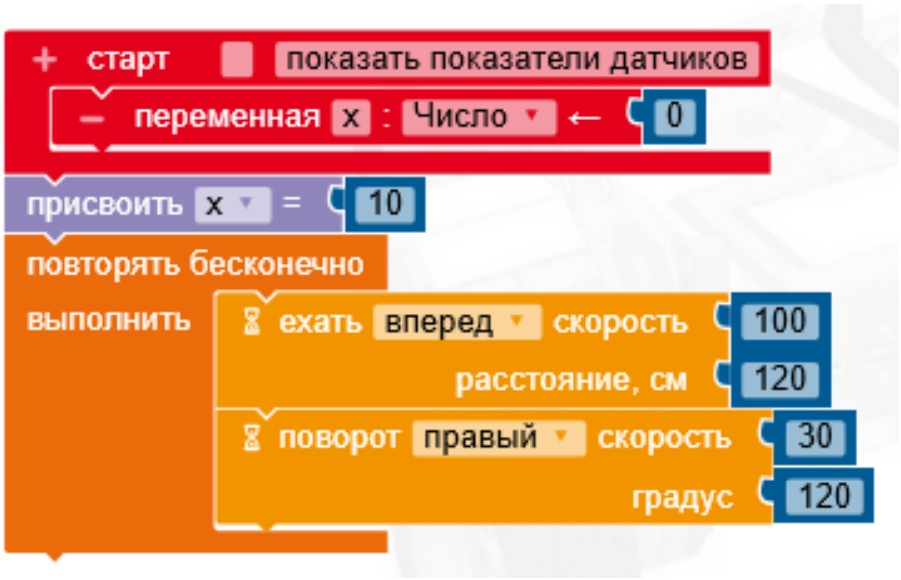


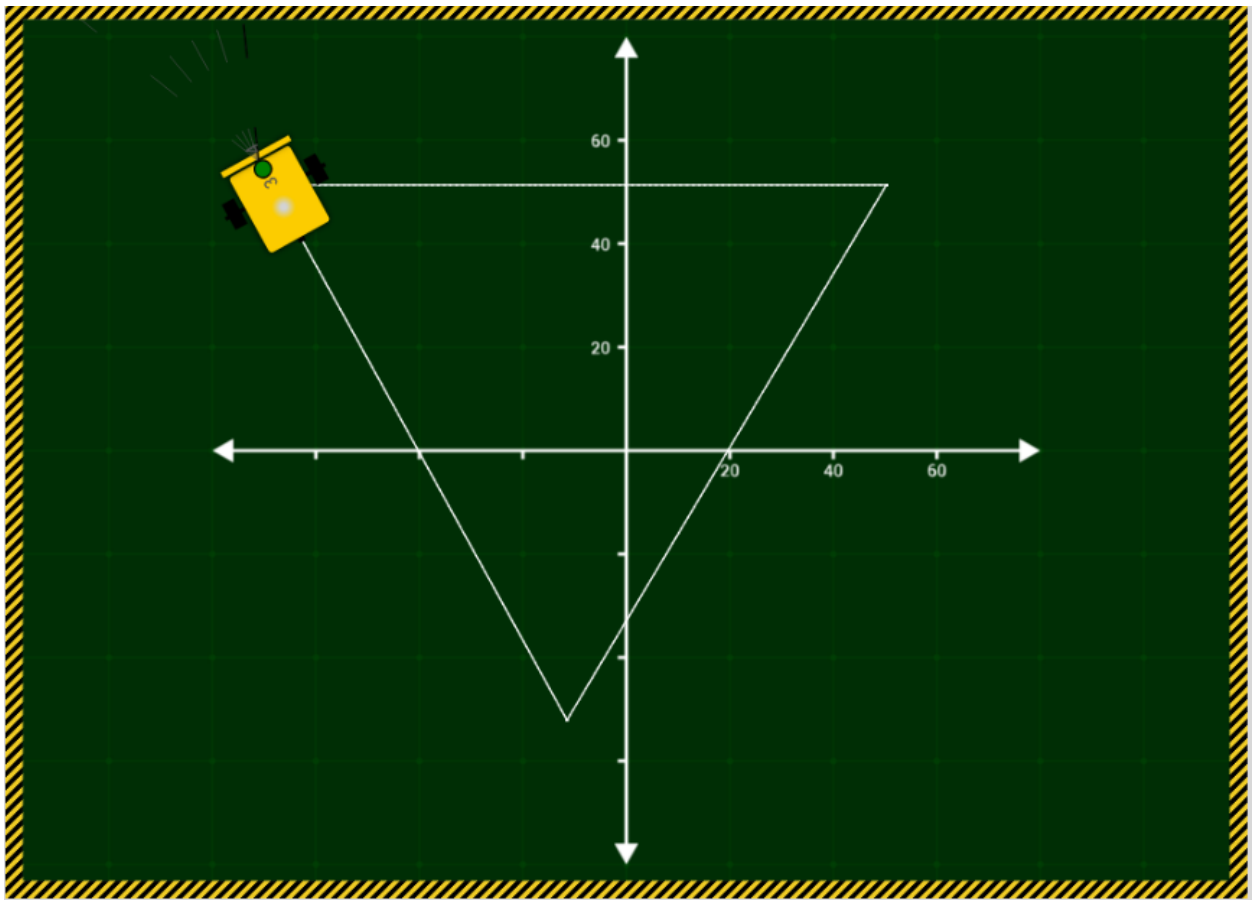
Рисунок 3. Пример программы «Треугольник»

Нажмите на элемент управления, выделенный прямоугольником на рисунке 4 «Enable/Disable robot draw trail». Нажмите левой кнопкой мыши один раз чтобы включить режим просмотра траектории (следа) движения робота.

Рисунок 4. Enable/Disable robot draw trail (След рисования роботом)

Для проверки нажмите на кнопке «Старт».

Ваш робот должен выполнять бесконечное движение вперед расстоянием 120 см со скоростью равной 100, затем повернуть на 120 градусов и продолжить выполнять движение расстоянием 120 см со скоростью равной 100, затем повернуть на 120 градусов, как показано на рисунке ниже.

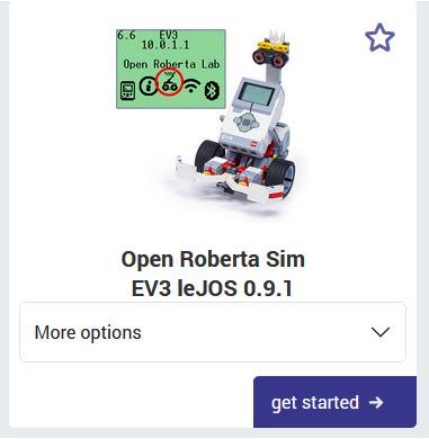


Дескрипторы:

- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Ехать вперед: скорость и расстояние»
- Используется действие «Ехать вперед: скорость и градус»
- Используется переменная из раздела «Переменные»
- Используется число из раздела «Математика»

Задание 2. Прямоугольник.

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Используйте для изменения сцены элемент управления, указанный стрелкой на рисунке 1 «Элементы управления». Нажмите левой кнопкой мыши несколько раз чтобы загрузить сцену с координатами, а затем измените положение робота, как на рисунке 2 «Сцена».

Рисунок 1. Элементы управления

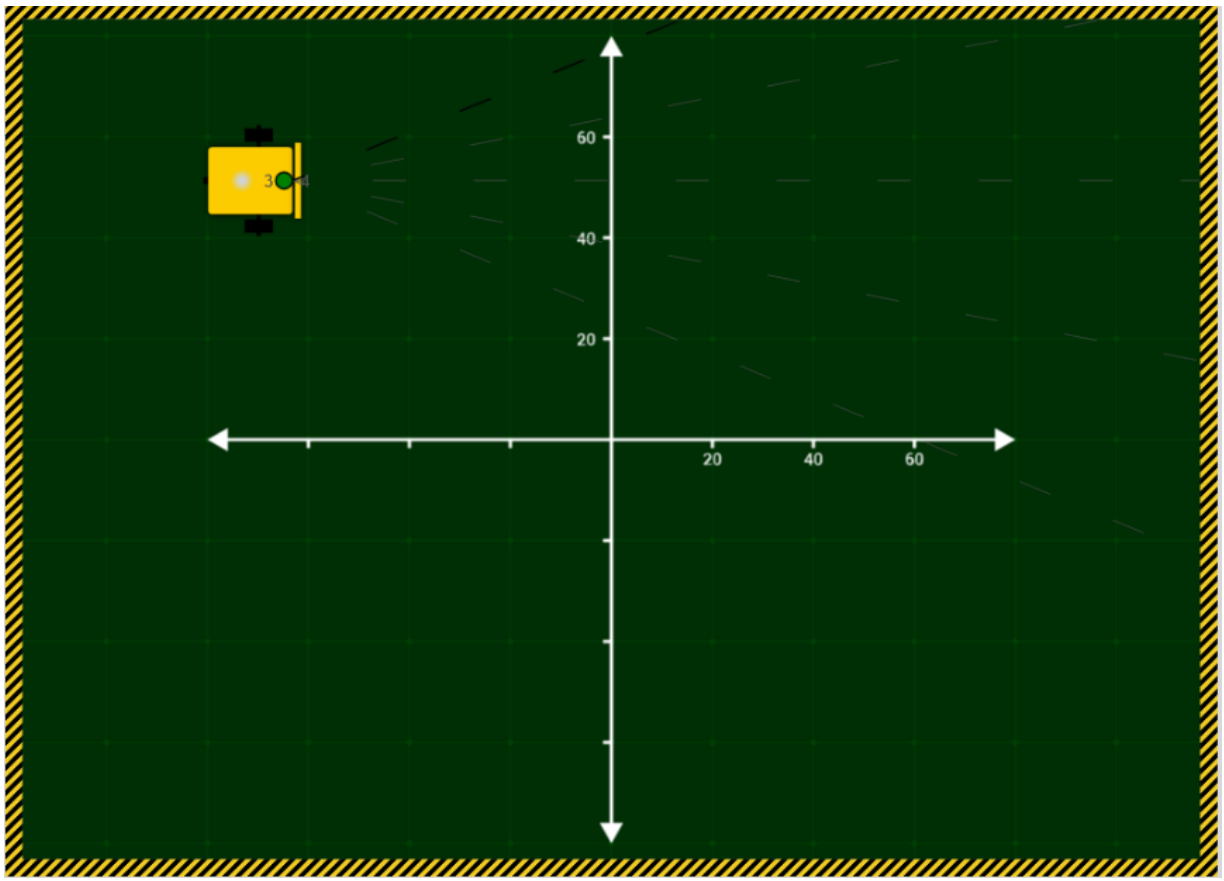


Рисунок 2. Сцена

Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Прямоугольник». Используйте следующие разделы «Действие», «Датчики», «Контроль», «Логика», «Математика» и «Переменные».

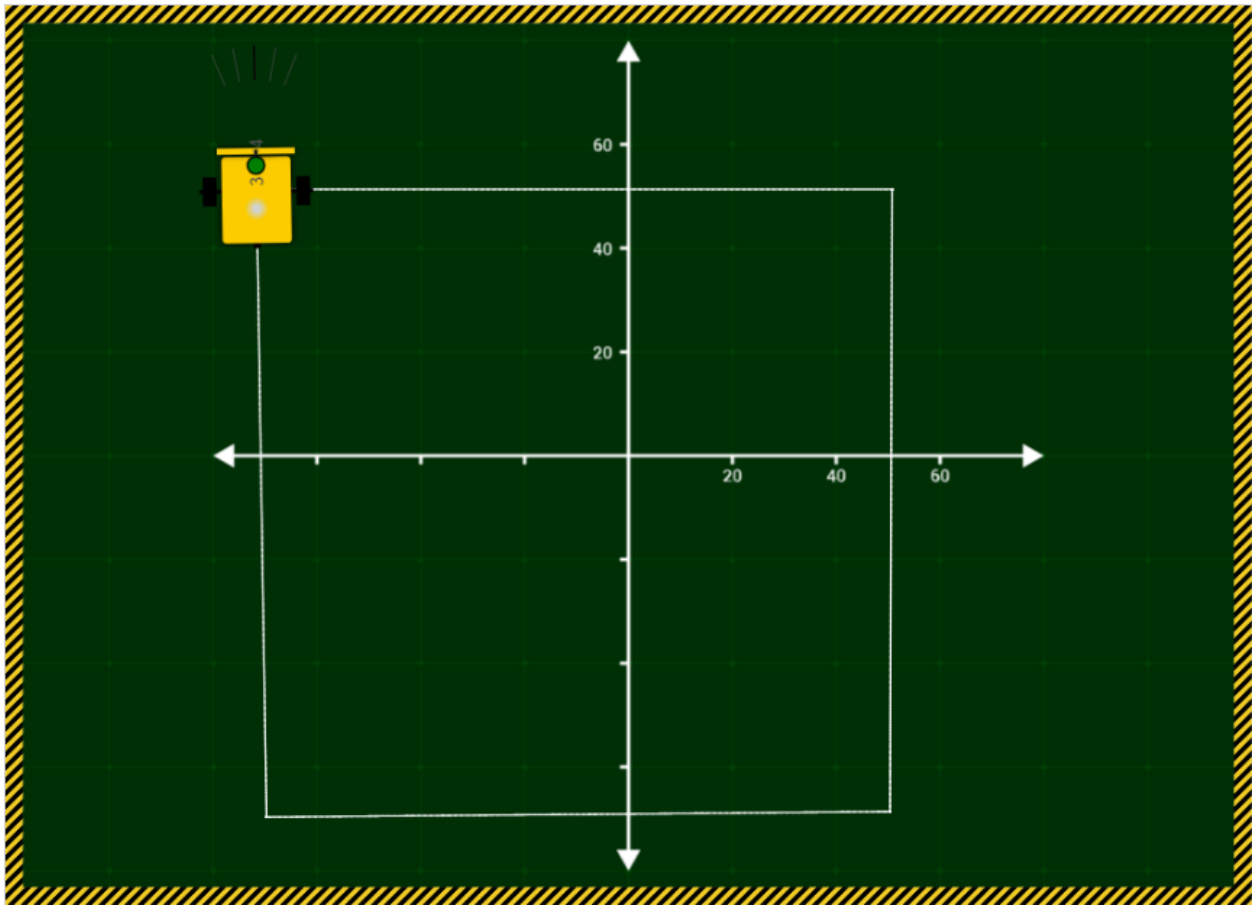
Рисунок 3. Пример программы «Прямоугольник»

Нажмите на элемент управления, выделенный прямоугольником на рисунке 4 «Enable/Disable robot draw trail». Нажмите левой кнопкой мыши один раз чтобы включить режим просмотра траектории (следа) движения робота.

Рисунок 4. Enable/Disable robot draw trail (След рисования роботом)

Для проверки нажмите на кнопке «Старт».

Ваш робот должен выполнять бесконечное движение вперед расстоянием 120 см со скоростью равной 100, затем повернуть на 89 градусов и продолжить выполнять движение расстоянием 120 см со скоростью равной 100, затем повернуть на 89 градусов, как показано на рисунке ниже.

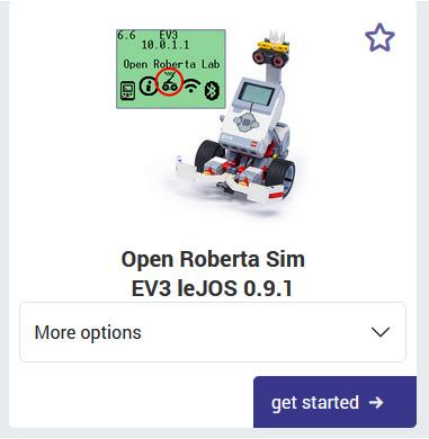


Дескрипторы:

- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Ехать вперед: скорость и расстояние»
- Используется действие «Ехать вперед: скорость и градус»
- Используется переменная из раздела «Переменные»
- Используется число из раздела «Математика»

Задание 3. «Квадратная» спираль

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Используйте для изменения сцены элемент управления, указанный стрелкой на рисунке 1 «Элементы управления». Нажмите левой кнопкой мыши несколько раз чтобы загрузить сцену с координатами, а затем измените положение робота, как на рисунке 2 «Сцена».

Рисунок 1. Элементы управления

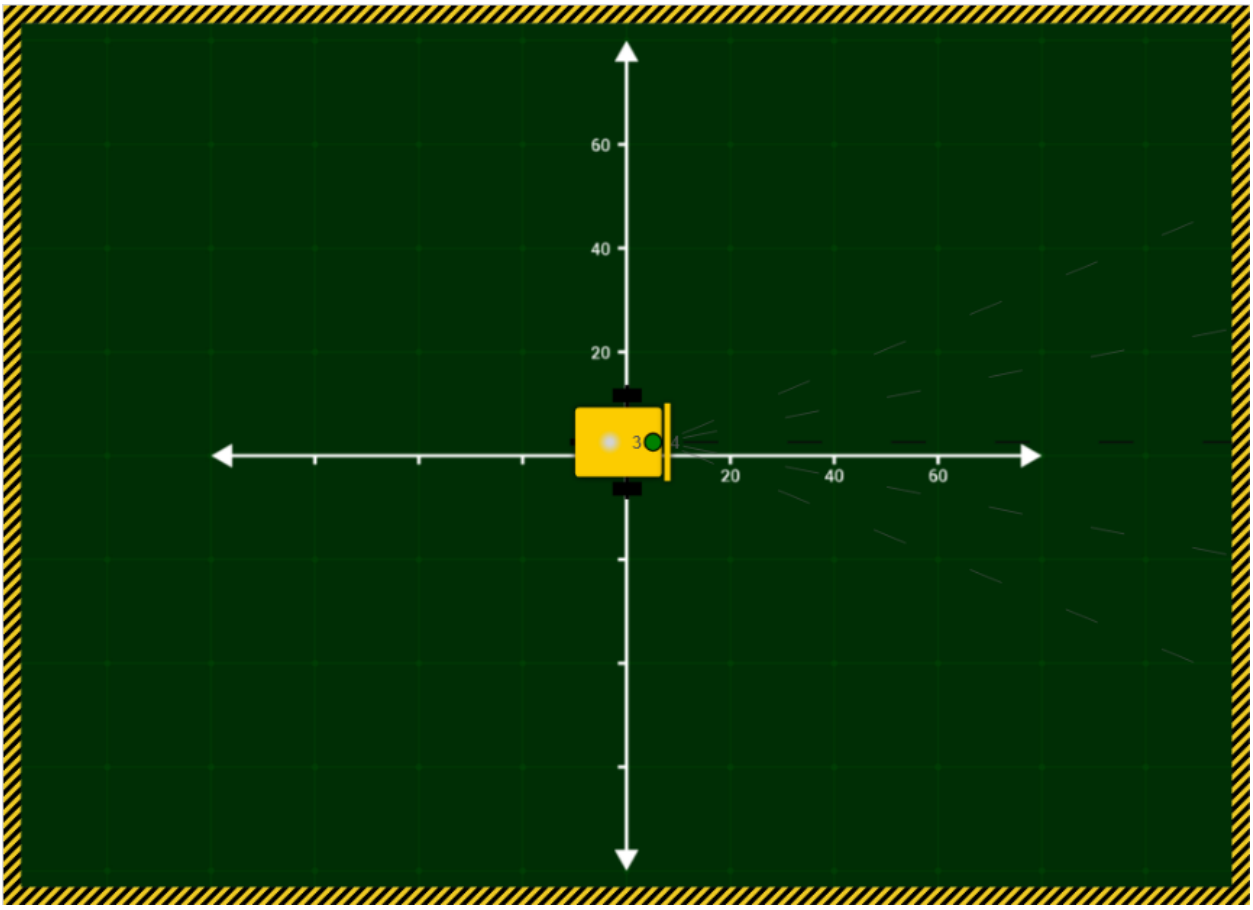


Рисунок 2. Сцена

Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Квадратная» спираль». Используйте следующие разделы «Действие», «Датчики», «Контроль», «Логика», «Математика» и «Переменные».

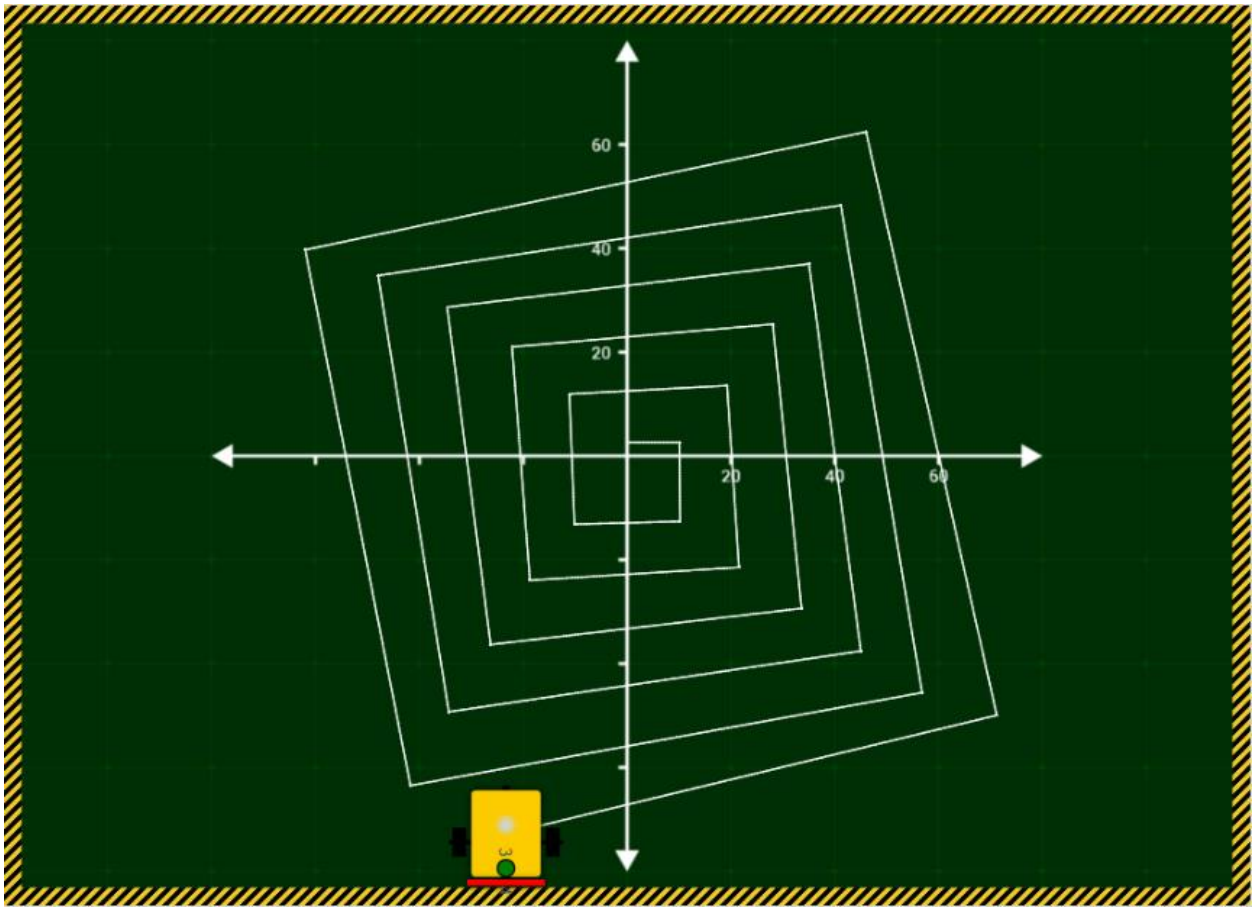
Рисунок 3. Пример программы «Квадратная» спираль»

Нажмите на элемент управления, выделенный прямоугольником на рисунке 4 «Enable/Disable robot draw trail». Нажмите левой кнопкой мыши один раз чтобы включить режим просмотра траектории (следа) движения робота.

Рисунок 4. Enable/Disable robot draw trail (След рисования роботом)

Для проверки нажмите на кнопке «Старт».

Ваш робот должен выполнять бесконечное движение вперед с начальным расстоянием 10 см со скоростью равной 50, затем повернуть на 89 градусов, а затем увеличить значения расстояния на 5 см, как показано на рисунке ниже.

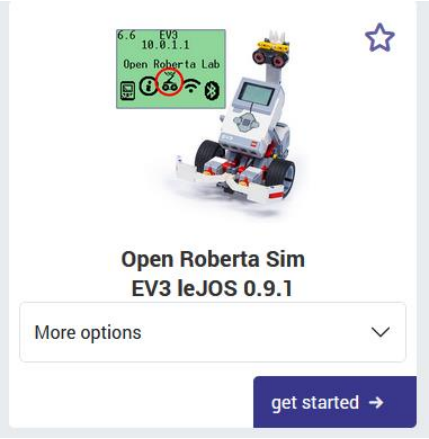


Дескрипторы:

- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Ехать вперед: скорость и расстояние»
- Используется действие «Ехать вперед: скорость и градус»
- Используется переменная из раздела «Переменные»
- Используется число из раздела «Математика»

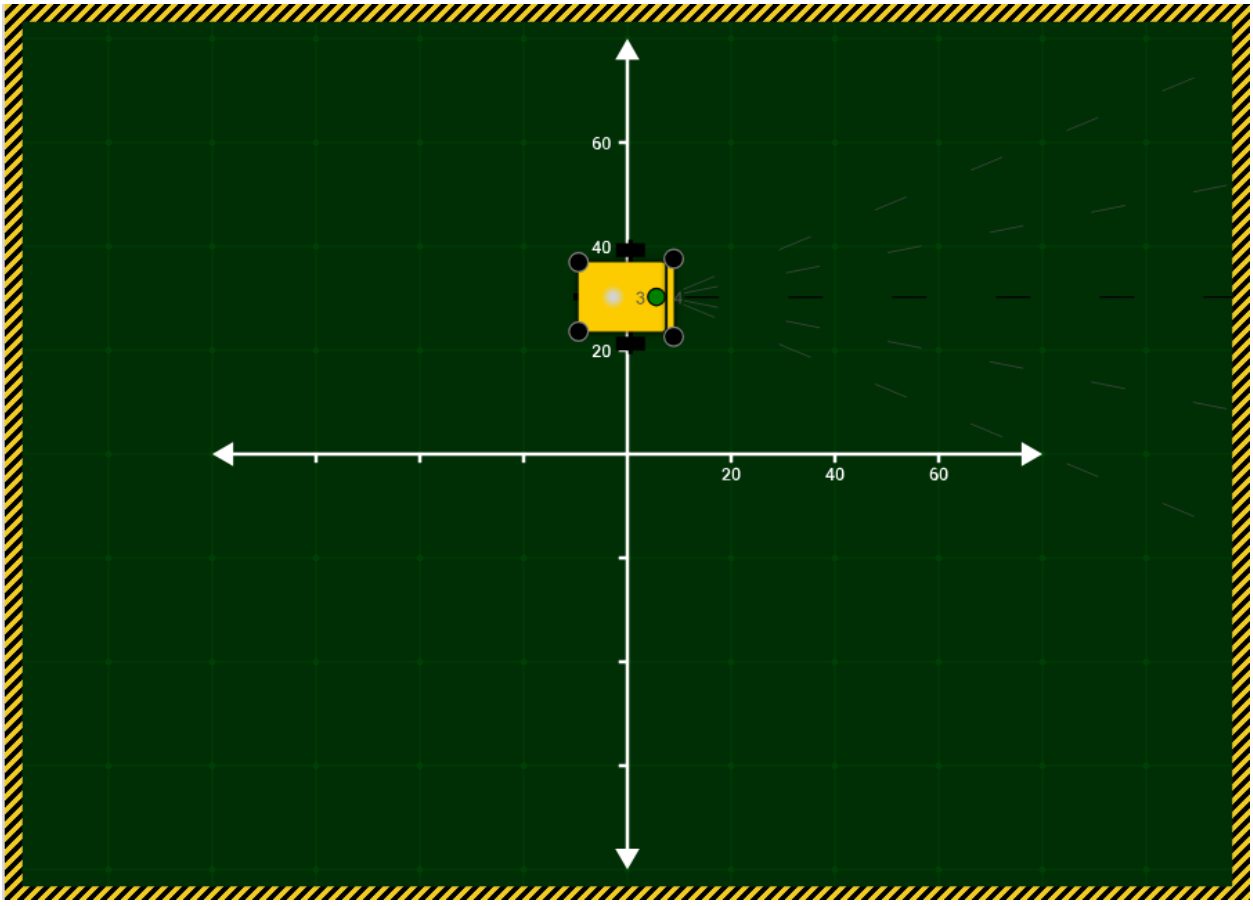
Задание 4. Круглая спираль

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Используйте для изменения сцены элемент управления, указанный стрелкой на рисунке 1 «Элементы управления». Нажмите левой кнопкой мыши несколько раз чтобы загрузить сцену с координатами, а затем измените положение робота, как на рисунке 2 «Сцена».

Рисунок 1. Элементы управления



Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Круглая спираль». Используйте следующие разделы «Действие», «Датчики», «Контроль», «Логика», «Математика» и «Переменные».

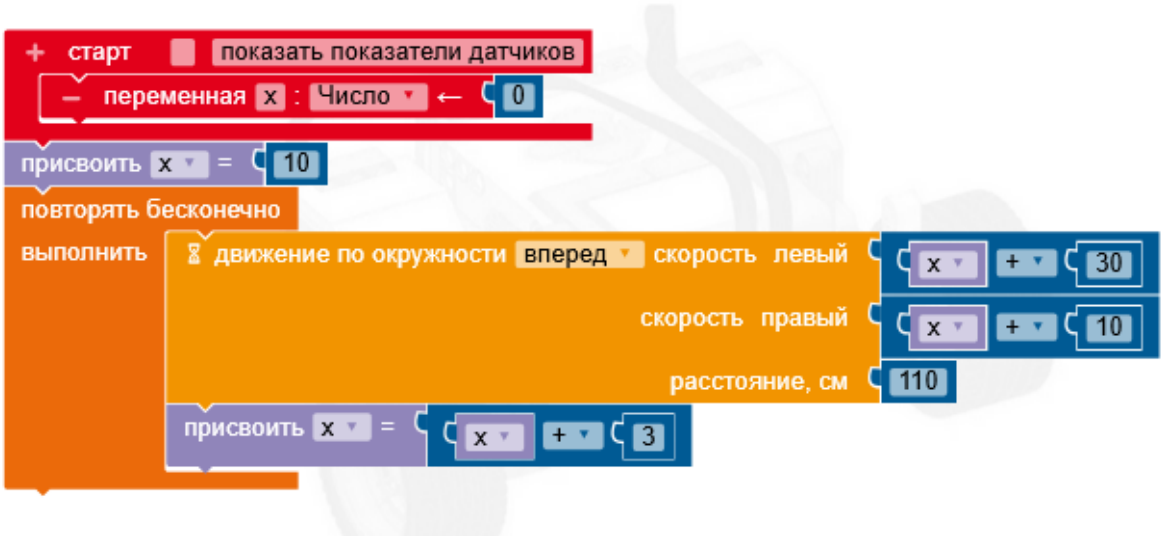


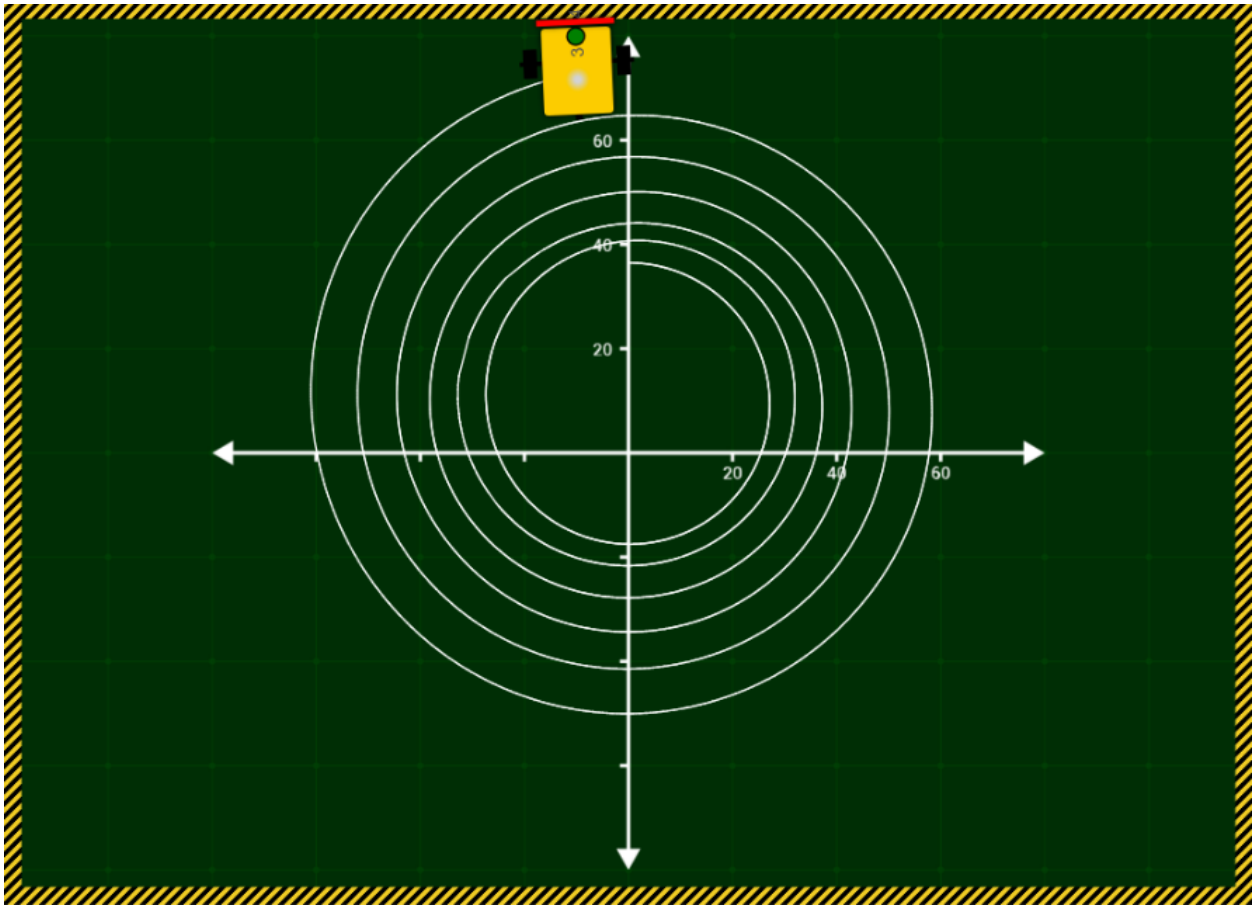
Рисунок 3. Пример программы «Круглая спираль»

Нажмите на элемент управления, выделенный прямоугольником на рисунке 4 «Enable/Disable robot draw trail». Нажмите левой кнопкой мыши один раз чтобы включить режим просмотра траектории (следа) движения робота.

Рисунок 4. Enable/Disable robot draw trail (След рисования роботом)

Для проверки нажмите на кнопке «Старт».

Ваш робот должен выполнять бесконечное движение вперед с начальным расстоянием 110 см и скоростью левого и правого моторов равных 30 и 10 соответственно, а затем увеличить значения скорости левого мотора на 3 и увеличить значения скорости правого мотора на 3, как показано на рисунке ниже.



Дескрипторы:

- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Ехать вперед: скорость и расстояние»
- Используется действие «Ехать вперед: скорость и градус»
- Используется переменная из раздела «Переменные»
- Используется число из раздела «Математика»

Тема. Определение цвета.

Задание1. Подготовка сцены. Откройте приложение Paint и используйте по порядку следующие инструменты: «Овал», «Изменение цветов» и «Заливка» в Панели инструментов меню «Главная». Используйте диалоговое окно «Изменение цветов», настройте цвет в соответствии с таблицей 1 «Настройка цвета» и нарисуйте с использованием инструмента «Овал» три овала: зеленый, желтый и красный, как показано на рисунке 1 «Пример поля» ниже. Разработайте картинку для движения робота (набор овалов трех цветов). Сохраните изображения на «Рабочий стол».

Таблица 1. Настройка цвета

Овалы	Числовые значение цвета		
	Красный	Зеленый	Синий
Зеленый овал	0	150	64
Желтый овал	255	237	0
Красный овал	255	0	0

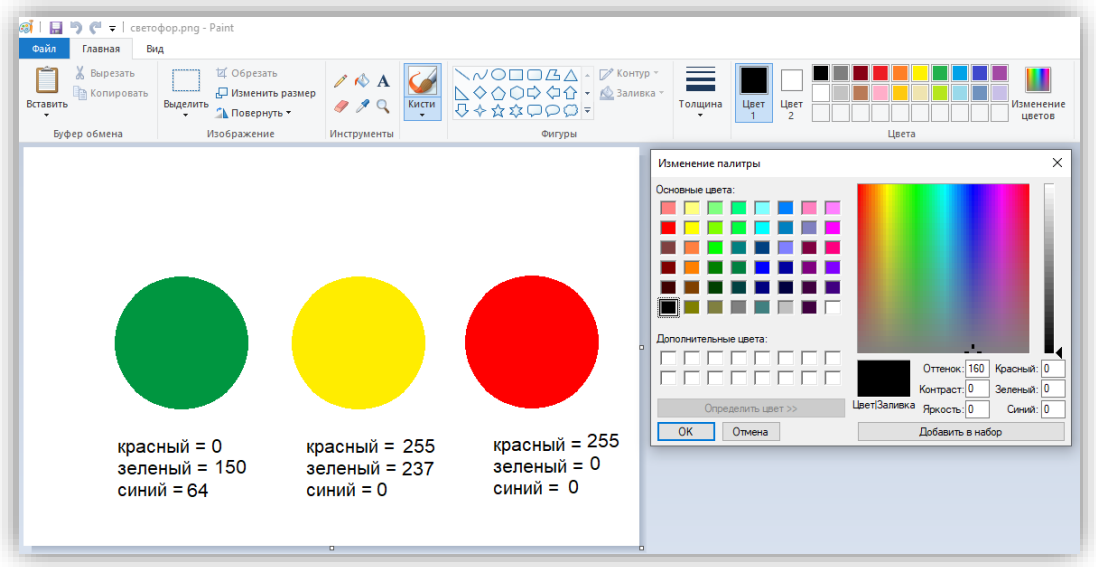
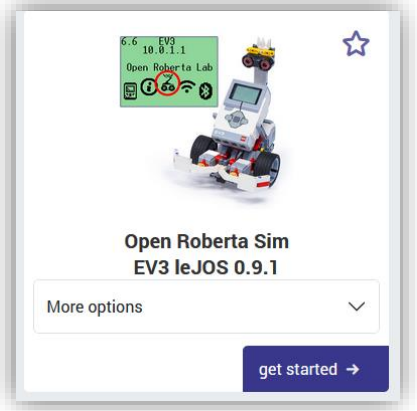
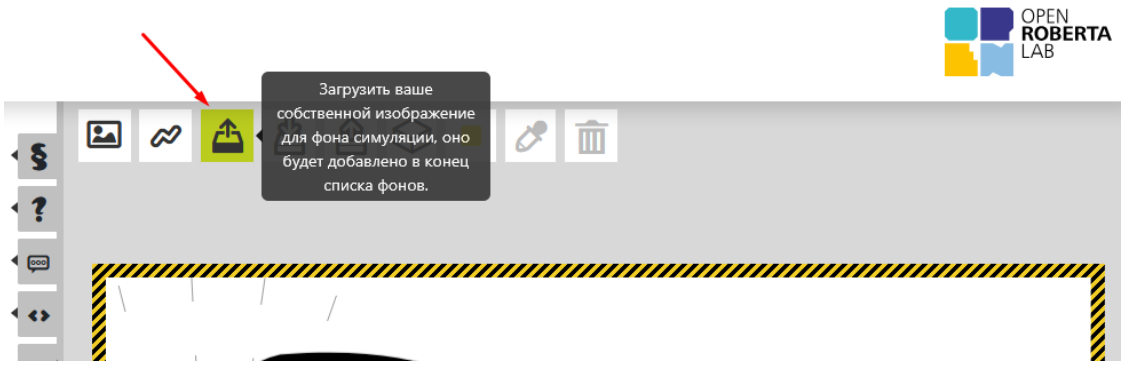


Рисунок 1. Пример поля

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Задание 2. Откройте сайт https://lab.open-roberta.org/ и выберите пункт (показан стрелкой)



Настройте сцену как на Рисунке 2 «Поле «Светофор» ниже. Используйте кнопки «стрелка влево» или «стрелка вправо» на клавиатуре и компьютерную мышь, чтобы изменить позицию робота. Разместите робота как в примере, представленном на изображении ниже.

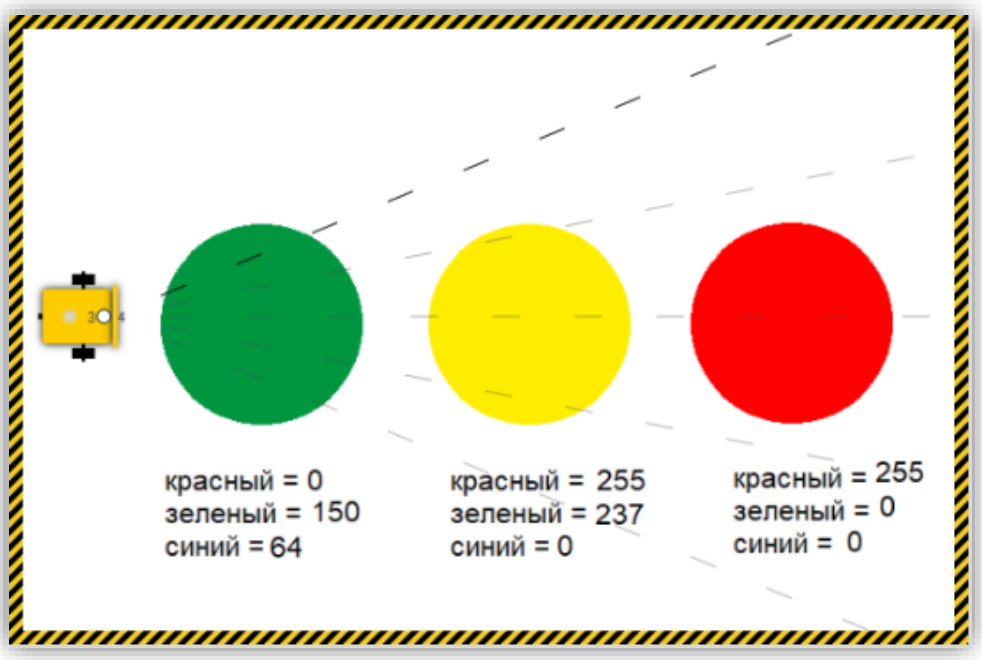


Рисунок 2. Поле «Светофор»

Задание 3. Составьте программу по примеру, представленному на рисунке 3 «Пример программы «Светофор». Обратите внимание, что в следующем задании вы можете изменять скорость движения робота в соответствии с цветом «Светофора».

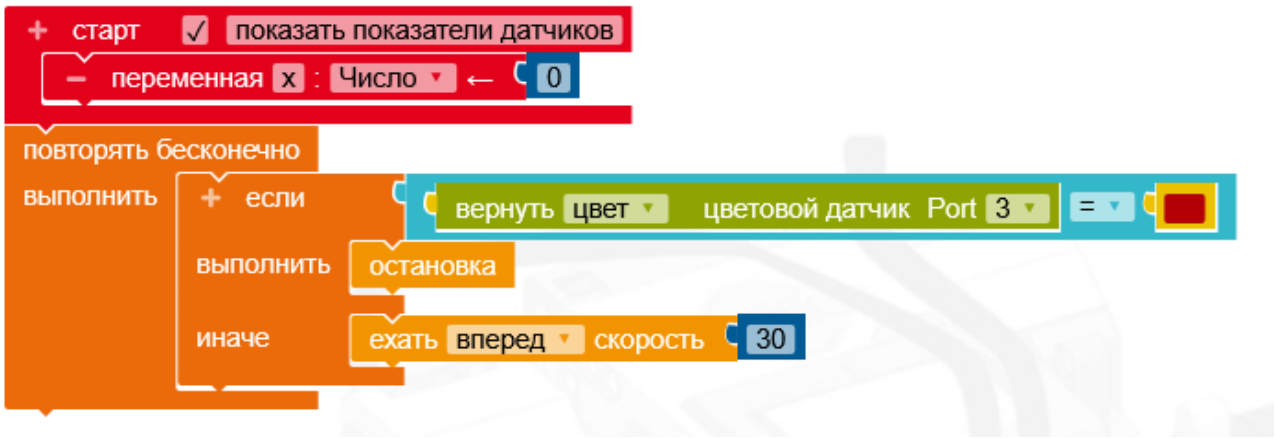


Рисунок 3. Пример программы «Светофор»

Для проверки нажмите на кнопке «Старт».

Ваш робот должен проехать со скоростью равной 30 по зеленому и желтому овалам и остановиться на блоке красного цвета как показано на рисунке ниже. Остановите программу.

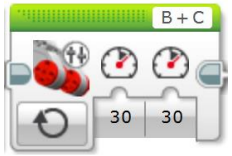
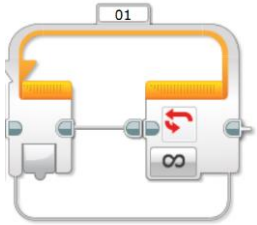
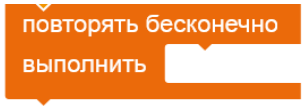
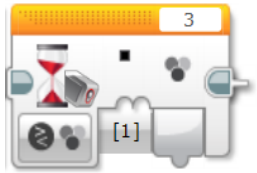
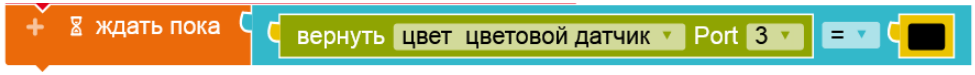
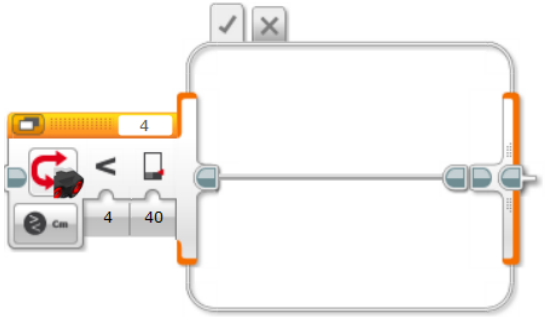
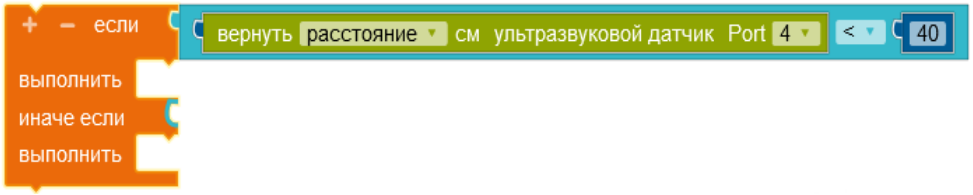
Задание 4. Составьте программу по примеру, представленному на рисунке 3 «Пример программы «Светофор 2».

Рисунок 4. Пример программы «Светофор 2»

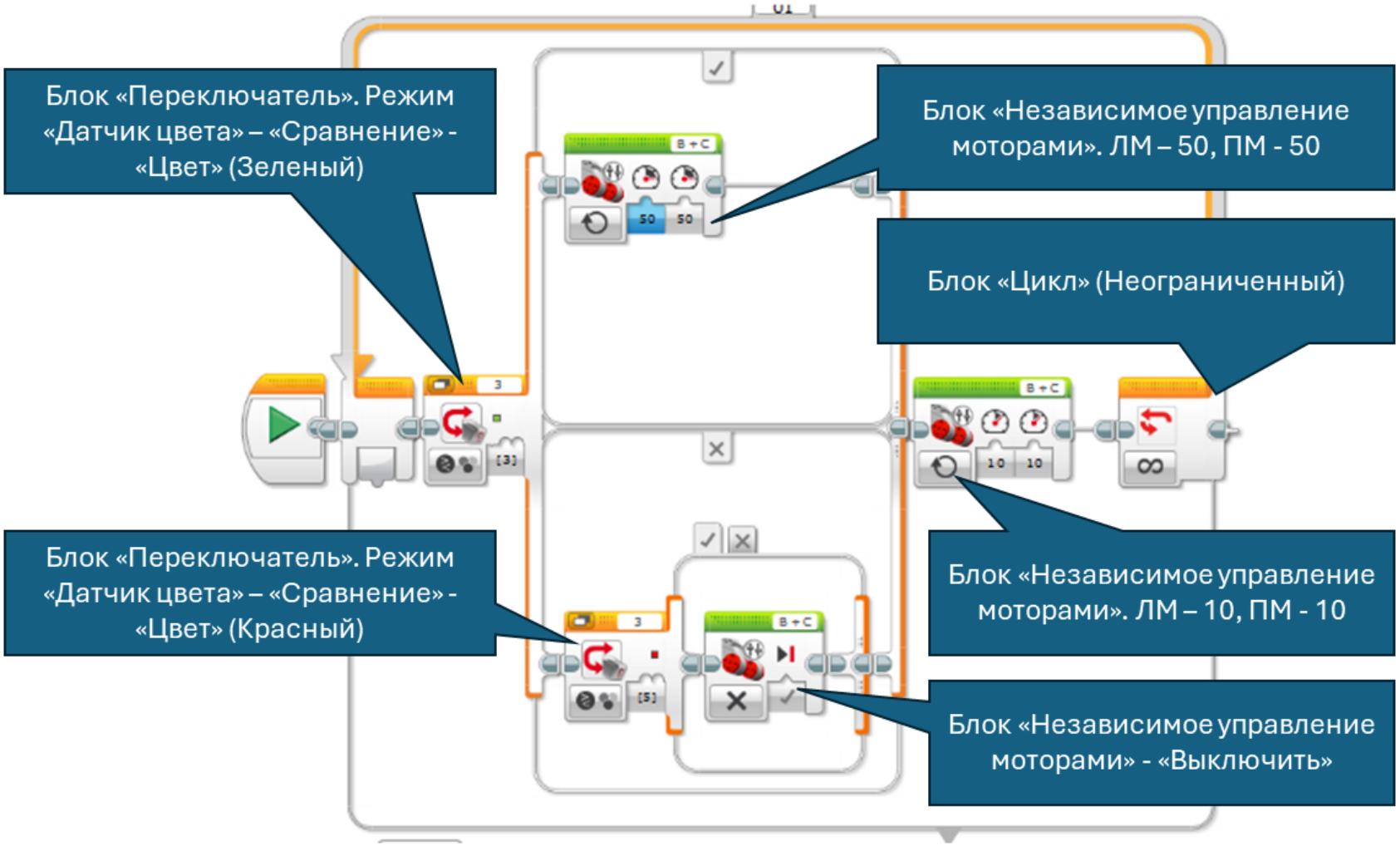
Для проверки нажмите на кнопке «Старт».

Ваш робот должен проехать со скоростью равной 70 по зеленому овалу, изменить скорость и замедлиться до скорости 10 проезжая желтый овал и остановиться на блоке красного цвета как показано на рисунке ниже. Остановите программу.

Сравнение инструментов Open Roberta Lab и Lego © Mindstorms EV3.

Блок EV3	Блок Open Roberta Lab	Назначение блока
		Управление моторами
		Цикл
		Ожидание
		Ветвление

Код программы «Светофор. Определение цвета» для платформы Lego © Mindstorms EV3



Дескрипторы:

- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок датчик цвета «Вернуть цвет»
- Используется Блок логика «Равно»
- Используется Блок действия «Ехать вперед»
- Используется Блок цвета: желтый, красный и зеленый цвет.

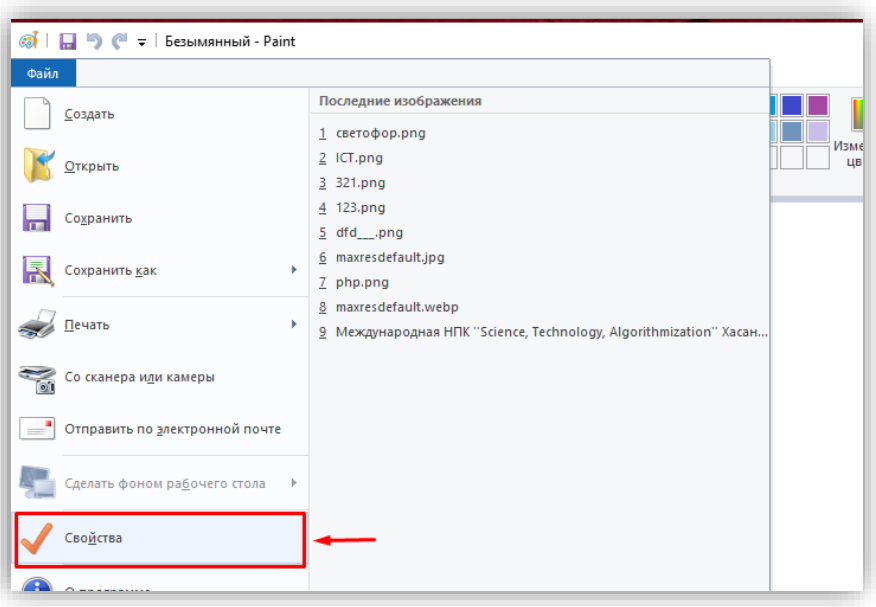
Тема. Сумо роботов.

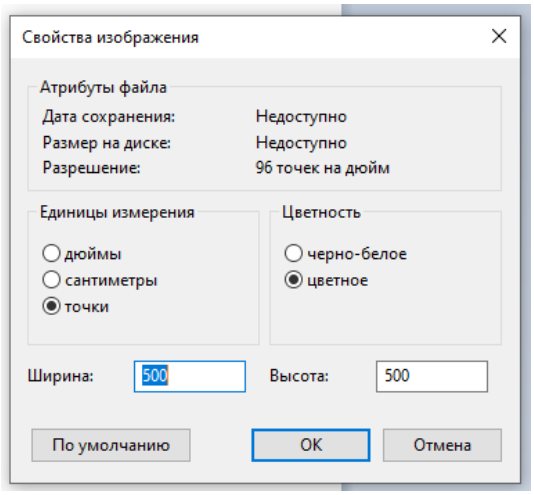
Задание1. Изменение размера изображения. Откройте приложение Paint. Выполните по порядку следующие действия.

1. Щелкните на пункте меню «Файл» для того чтобы открыть выпадающее меню.
2. Щелкните на пункте «Свойства.
3. Введите новые значения ширины изображения из таблицы 1
4. Введите новые значения высоты изображения из таблицы 1
5. Нажмите «Ок»

Таблица 1. Размеры изображения

Параметры изображения	Размер (в точках)	Размер (в см)
Ширина	500	13,23
Высота	500	13,23





Задание 2. Подготовка сцены. Используйте инструменты: «Овал», и «Заливка» (панель инструментов «Главная») для разработки картинки в соответствии с инструкцией ниже и по примеру представленному на рисунке 1.

1. Этап 1. Используйте «Овал» для рисования двух овалов
2. Этап 2. Используйте «Заливка» для заливки между внутренним и внешним овалами.
3. Этап 3. Сохраните получившееся изображения на «Рабочий стол» (Сохранить как → Рисунок в формате PNG) с именем файла «Сумо».

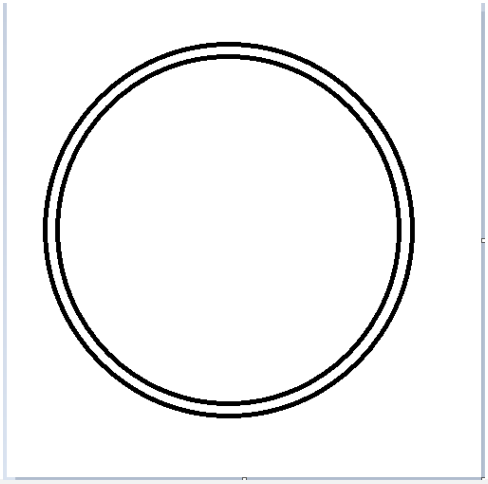


Рисунок 1. Пример поля. Этап 1.

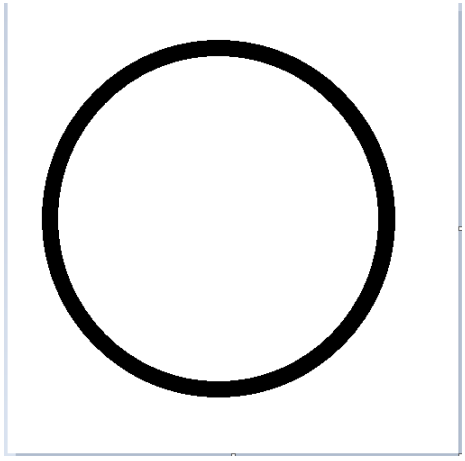
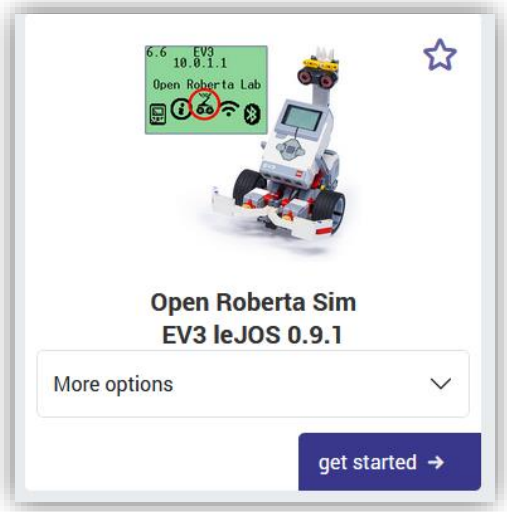


Рисунок 1. Пример поля. Этап 2.

Задание 3. Откройте сайт https://lab.open-roberta.org/ и выберите пункт как на рисунке ниже. Для настройки сцены выполните все следующие действия в строгой последовательности.



1) Загрузите собственное изображение с именем файла «Сумо» (Рисунок 2).

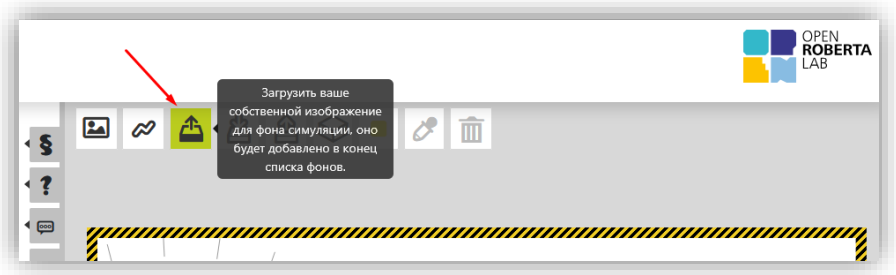


Рисунок 2. Загрузка собственного изображения.

2) Используйте кнопки «стрелка влево» или «стрелка вправо» на клавиатуре и компьютерную мышь, чтобы изменить позицию робота. Разместите робота как в примере, представленном на рисунке 3 ниже.

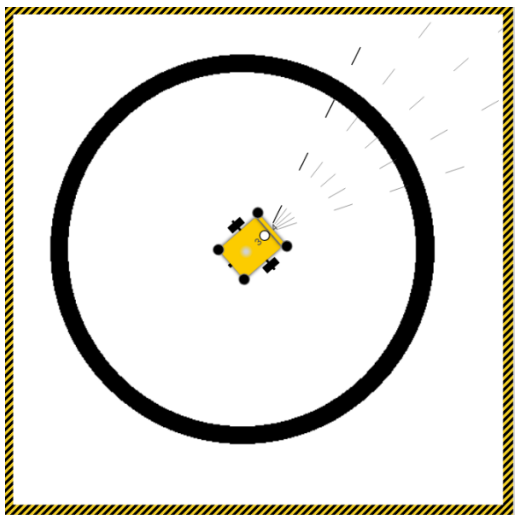


Рисунок 3. Размещение робота на поле «Сумо»

3) Добавьте один объект «Препятствие» круглой формы и разместите его как в примере, представленном на рисунке 4 ниже.

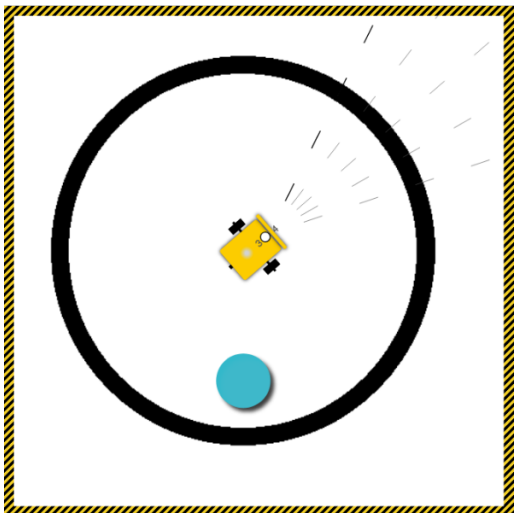


Рисунок 4. Размещение робота и препятствия на поле «Сумо»

Задание 4. Составьте программу по примеру, представленному на рисунках 5-6 «Пример программы «Сумо». Используйте следующие разделы «Действие», «Датчики», «Контроль», «Логика», «Математика». Обратите внимание, что в следующем задании вы можете изменить условие для того, чтобы робот не выходил за пределы круга.

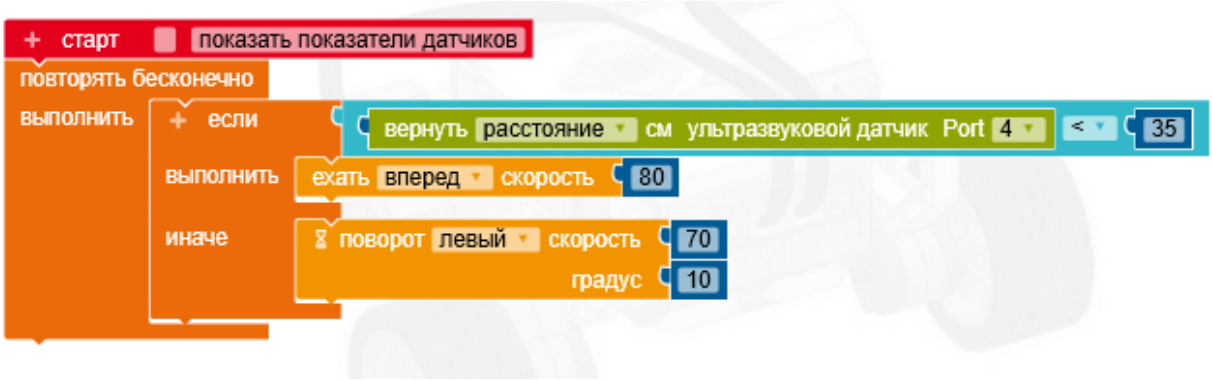


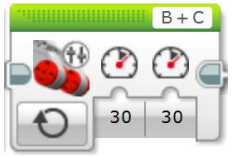
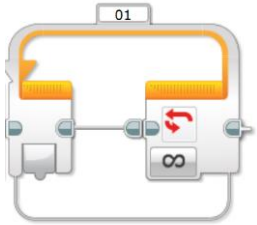
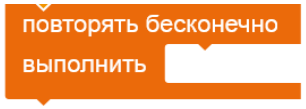
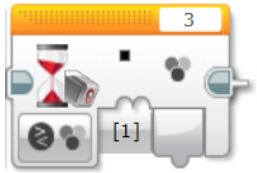
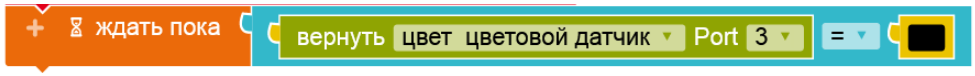
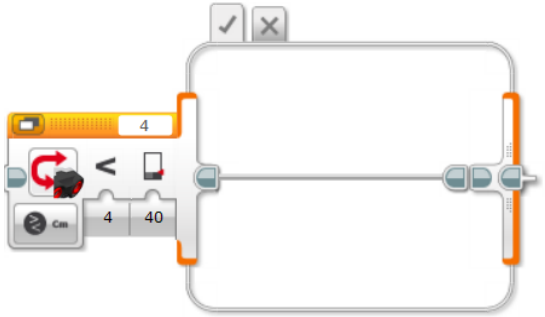
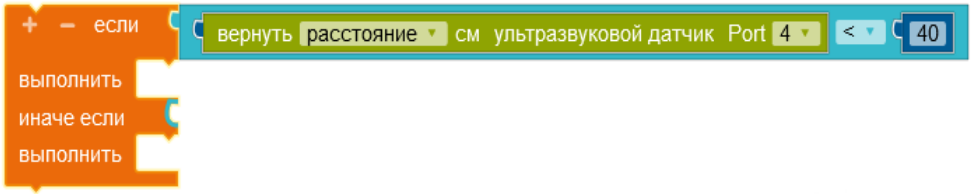
Рисунок 5. Пример программы «Сумо»

Рисунок 6. Пример программы «Сумо» с обнаружением линии

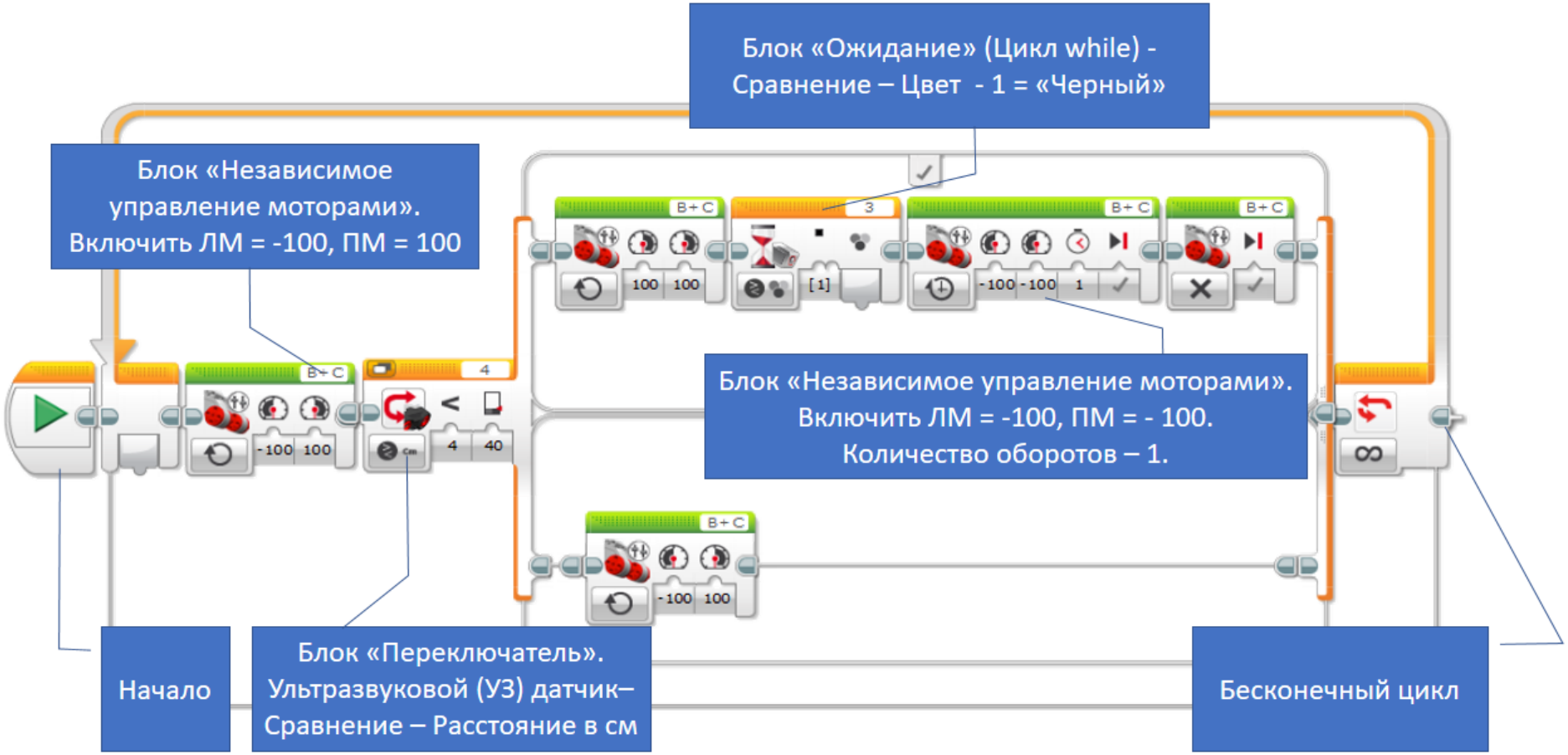
Для проверки нажмите на кнопке «Старт».

Ваш робот должен выполнять разворот на месте со скоростью равной 70 по и начать движение вперед при обнаружении препятствия как показано на рисунке ниже.

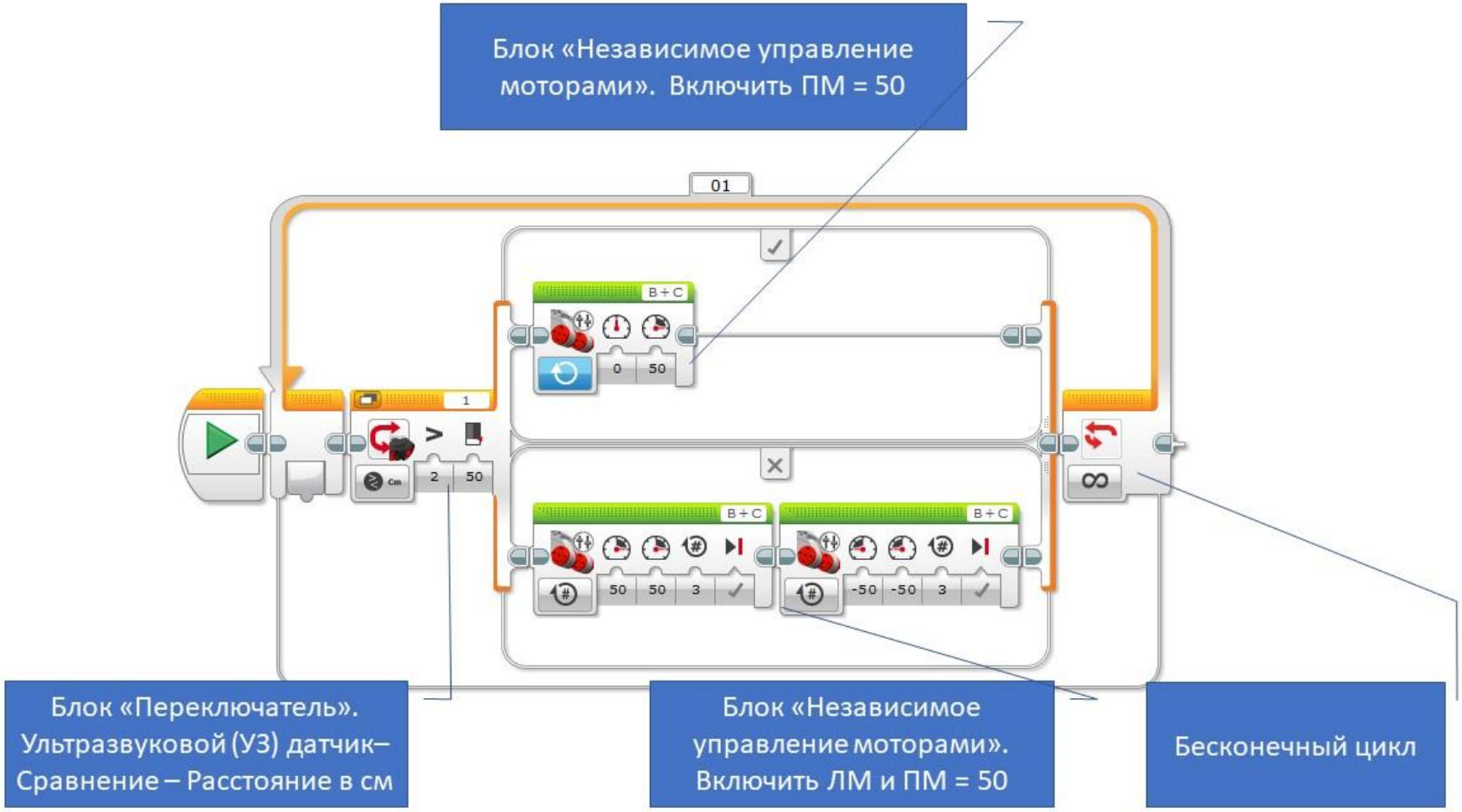
Сравнение инструментов Open Roberta Lab и Lego © Mindstorms EV3.

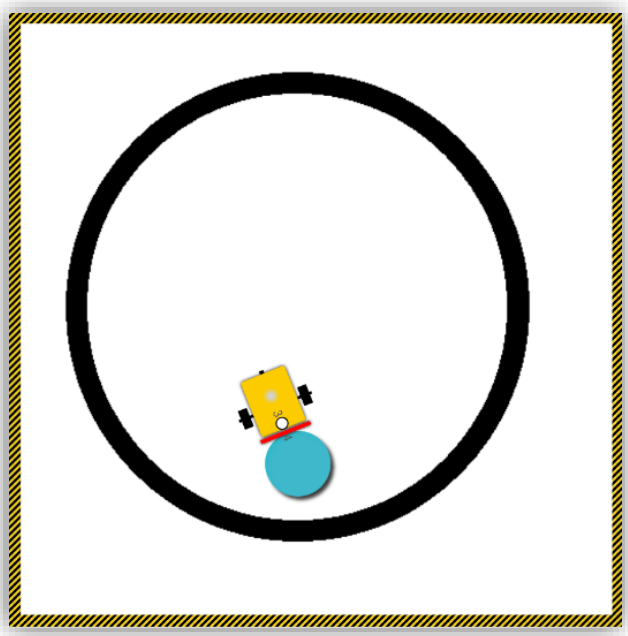
Блок EV3	Блок Open Roberta Lab	Назначение блока
		Управление моторами
		Цикл
		Ожидание
		Ветвление

Код программы «Сумо роботов» для платформы Lego © Mindstorms EV3



Код программы «Кегельринг» для платформы Lego © Mindstorms EV3





Дескрипторы:

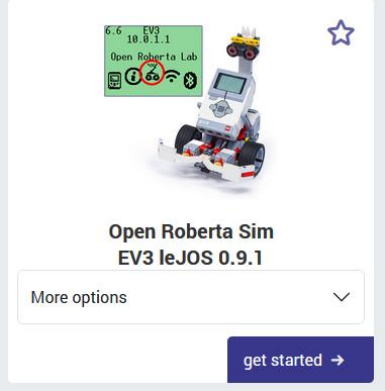
- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок датчик расстояния «Вернуть расстояние»
- Используется Блок логика «Равно»
- Используется Блок действия «Ехать вперед»
- Используется Блок математика: число

Остановите программу после выполнения задания.

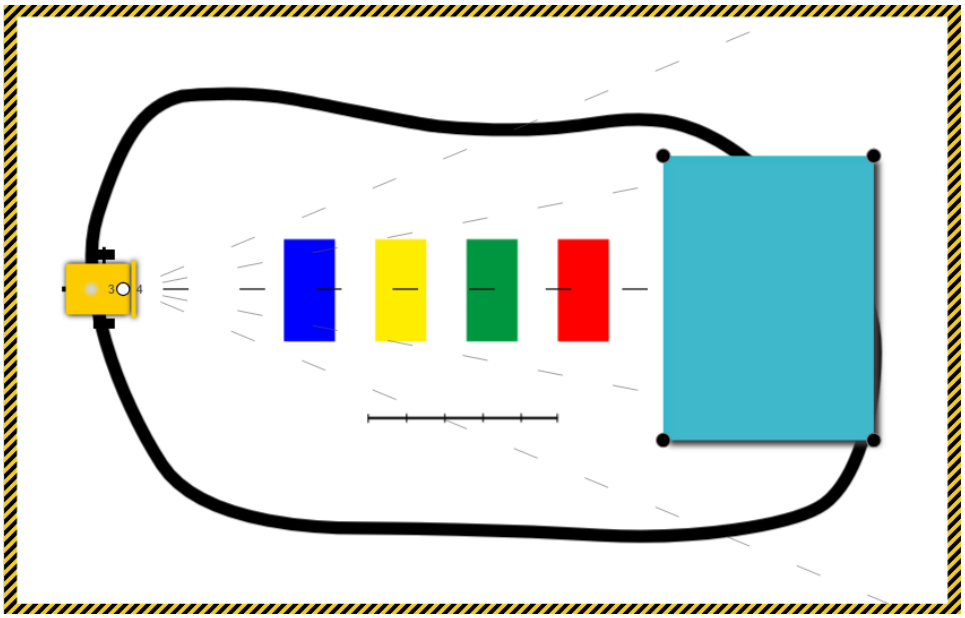
Тема. Движение в лабиринте.

Откройте сайт https://lab.open-roberta.org/

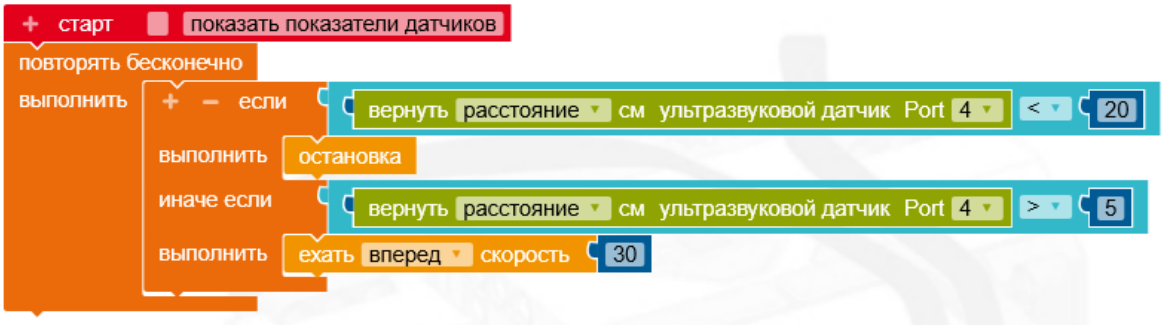
2. Выбрать пункт



1. Настройте сцену как на картинке ниже



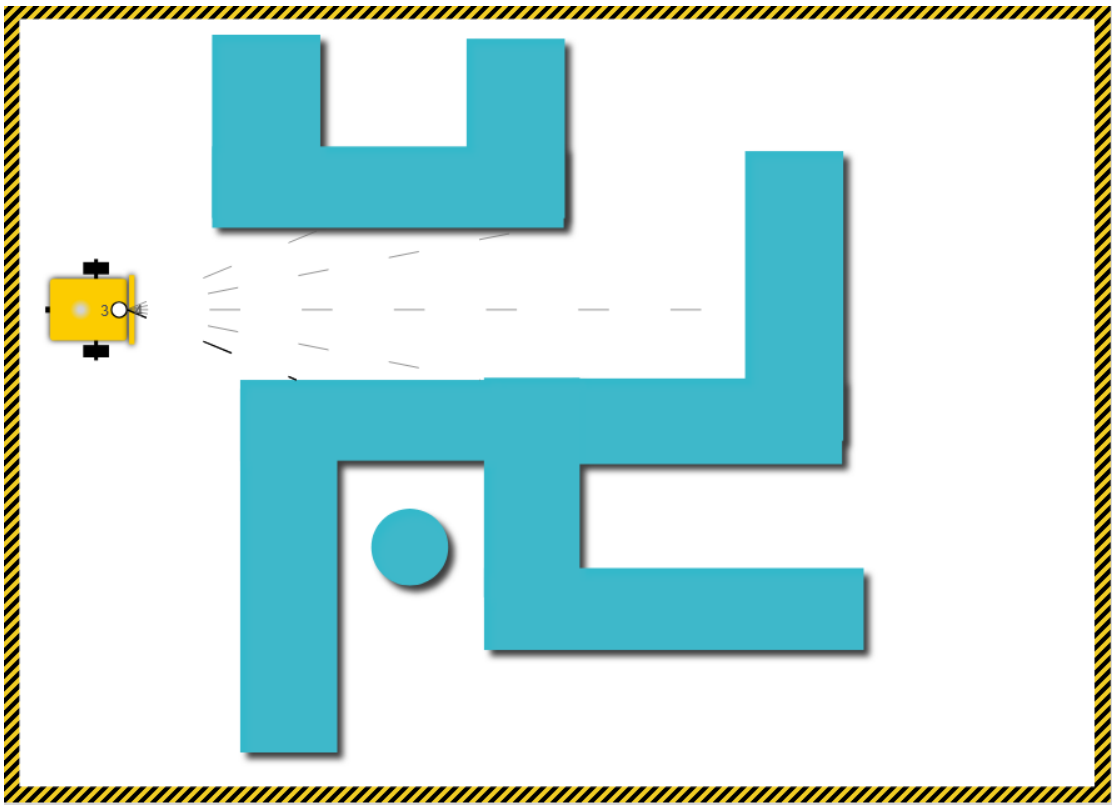
2. Составить программу



3. Нажать на кнопке «Старт» 

Задание 6. Виртуальная лаборатория Lego EV3/Spike. Инструкция Open Roberta Lab.

1) Открыть сайт https://lab.open-roberta.org/
2) Выбрать пункт в соответствии с инструкцией на обратной стороне листа
3) Настроить сцену (карту) с лабиринтом как на картинке ниже
   - Прямоугольники – это стены.
   - Робот не может проезжать сквозь стены
   - Круг – это цель
   - Роботу надо доехать до цели (до круга)

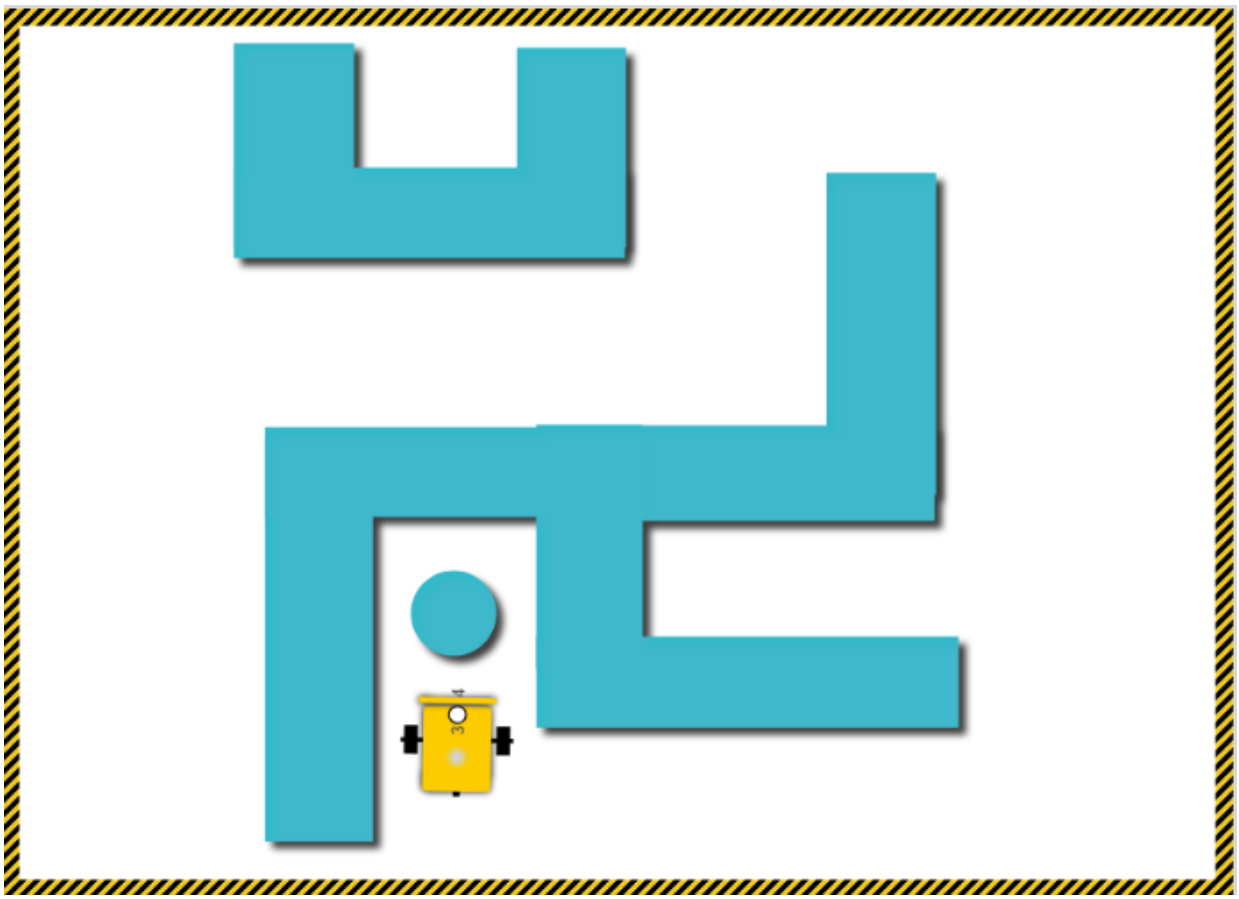


4) Составьте программу для того, чтобы робот достиг цели

Дескрипторы:

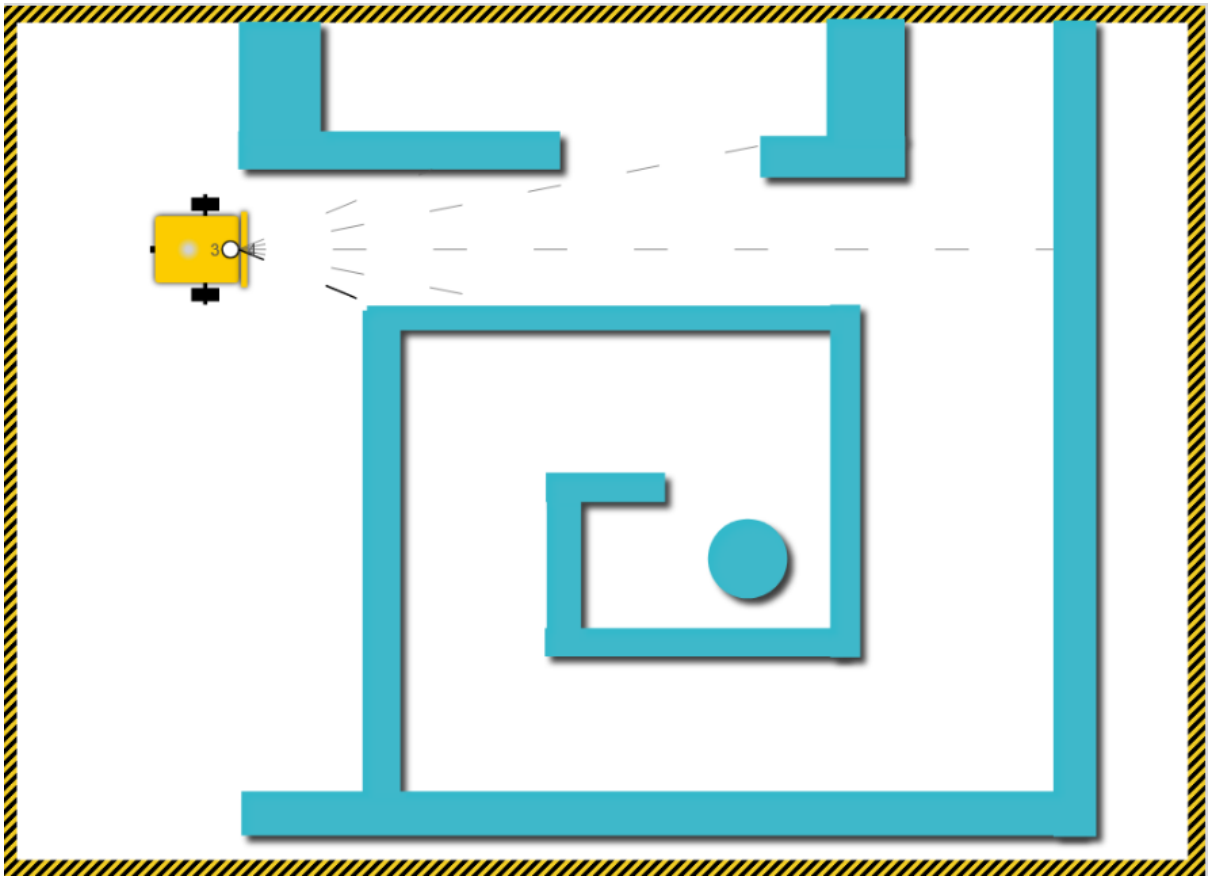
- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок ультразвуковой датчик «Вернуть расстояние»
- Используется Блок логика «Больше» или «Меньше»
- Используется Блок действия «Ехать вперед»

Ответ и результат:



Задание 7. Виртуальная лаборатория Lego EV3/Spike. Инструкция Open Roberta Lab.

1) Открыть сайт https://lab.open-roberta.org/
2) Выбрать пункт в соответствии с инструкцией на обратной стороне листа
3) Настроить сцену (карту) с лабиринтом как на картинке ниже
   - Прямоугольники – это стены.
   - Робот не может проезжать сквозь стены
   - Круг – это цель
   - Роботу надо доехать до цели (до круга)

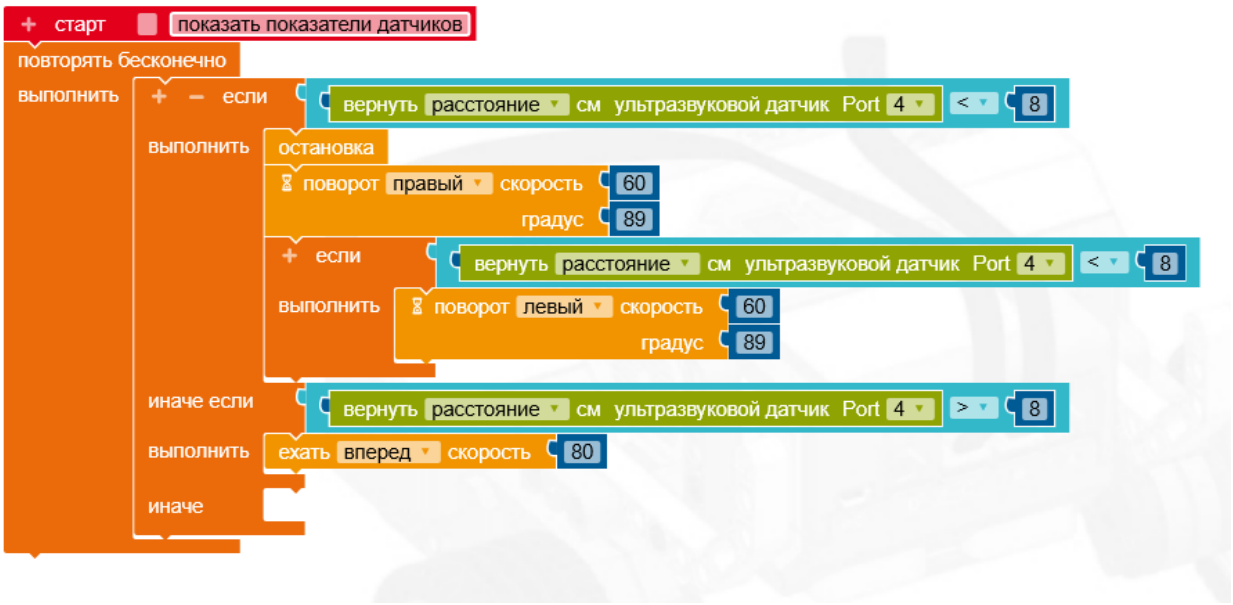


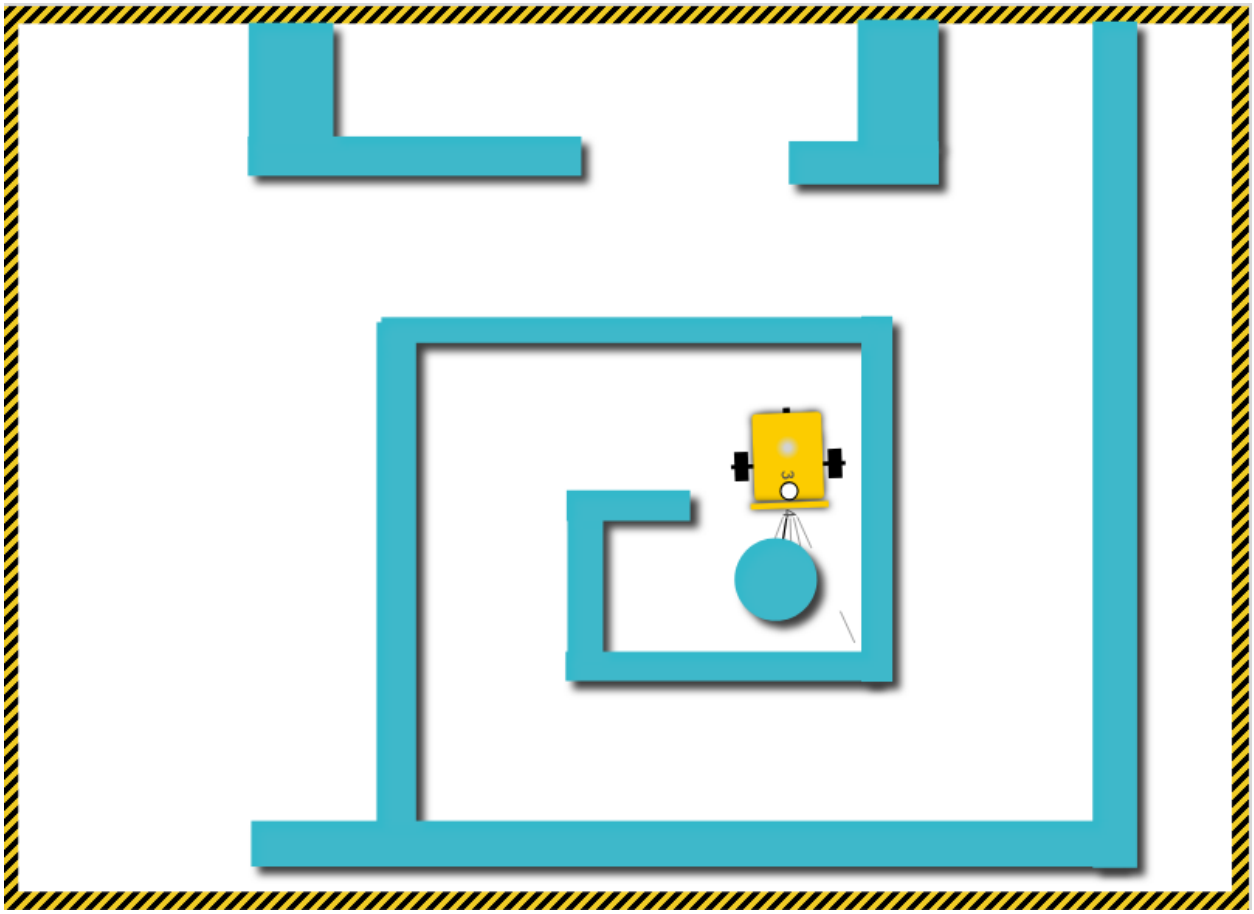
5) Составьте программу для того, чтобы робот достиг цели

Дескрипторы:

- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок ультразвуковой датчик «Вернуть расстояние»
- Используется Блок логика «Больше» или «Меньше»
- Используется Блок действия «Ехать вперед»

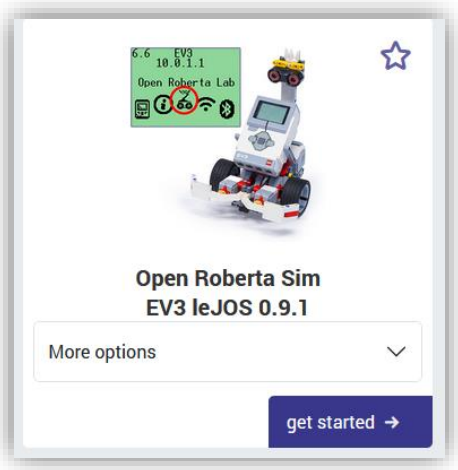
Ответ и результат:





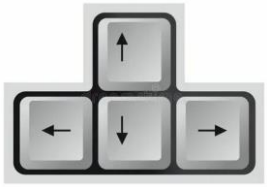
Тема. Движение по линии.

Откройте сайт https://lab.open-roberta.org/ и выберите пункт

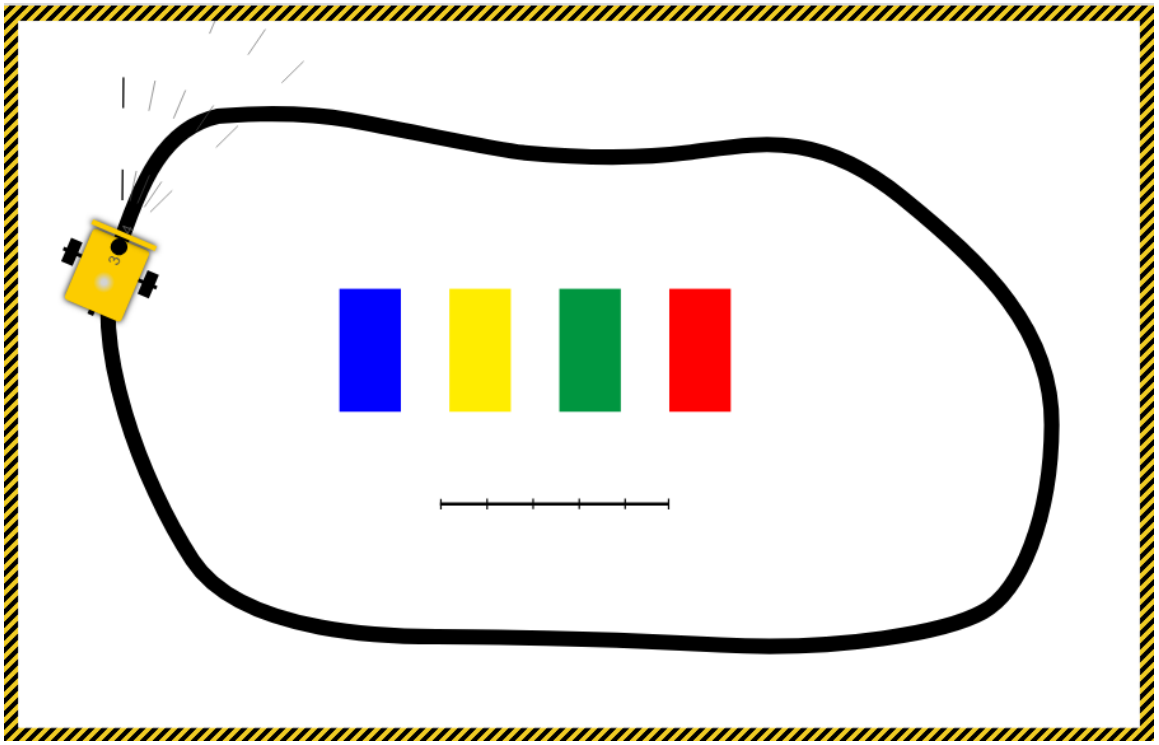


4. Настроить сцену как на картинке ниже

Найдите стрелки на клавиатуре.



Используйте кнопки «стрелка влево» или «стрелка вправо» на клавиатуре и компьютерную мышь, чтобы изменить позицию робота. Устанавливаем робота таким образом, чтобы он стоял на черной линии (как на рисунке ниже).



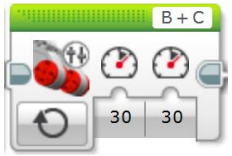
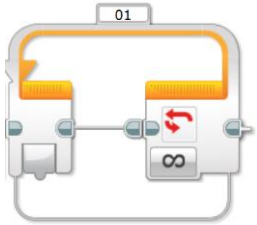
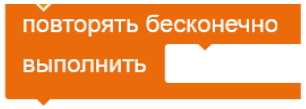
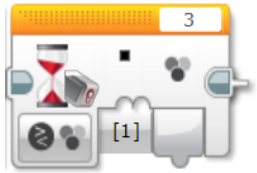
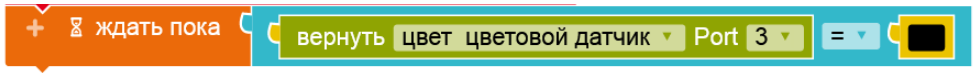
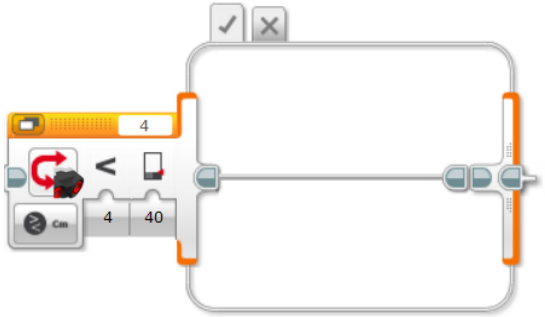
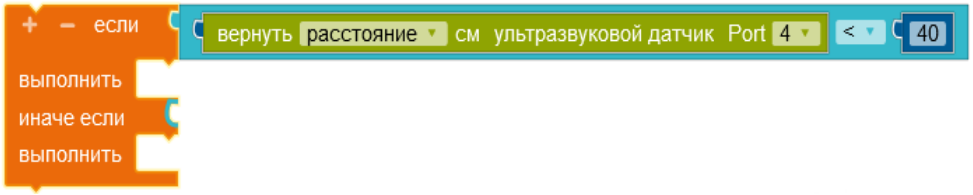
Задание 1. Составить программу

5. Нажать на кнопке «Старт» 

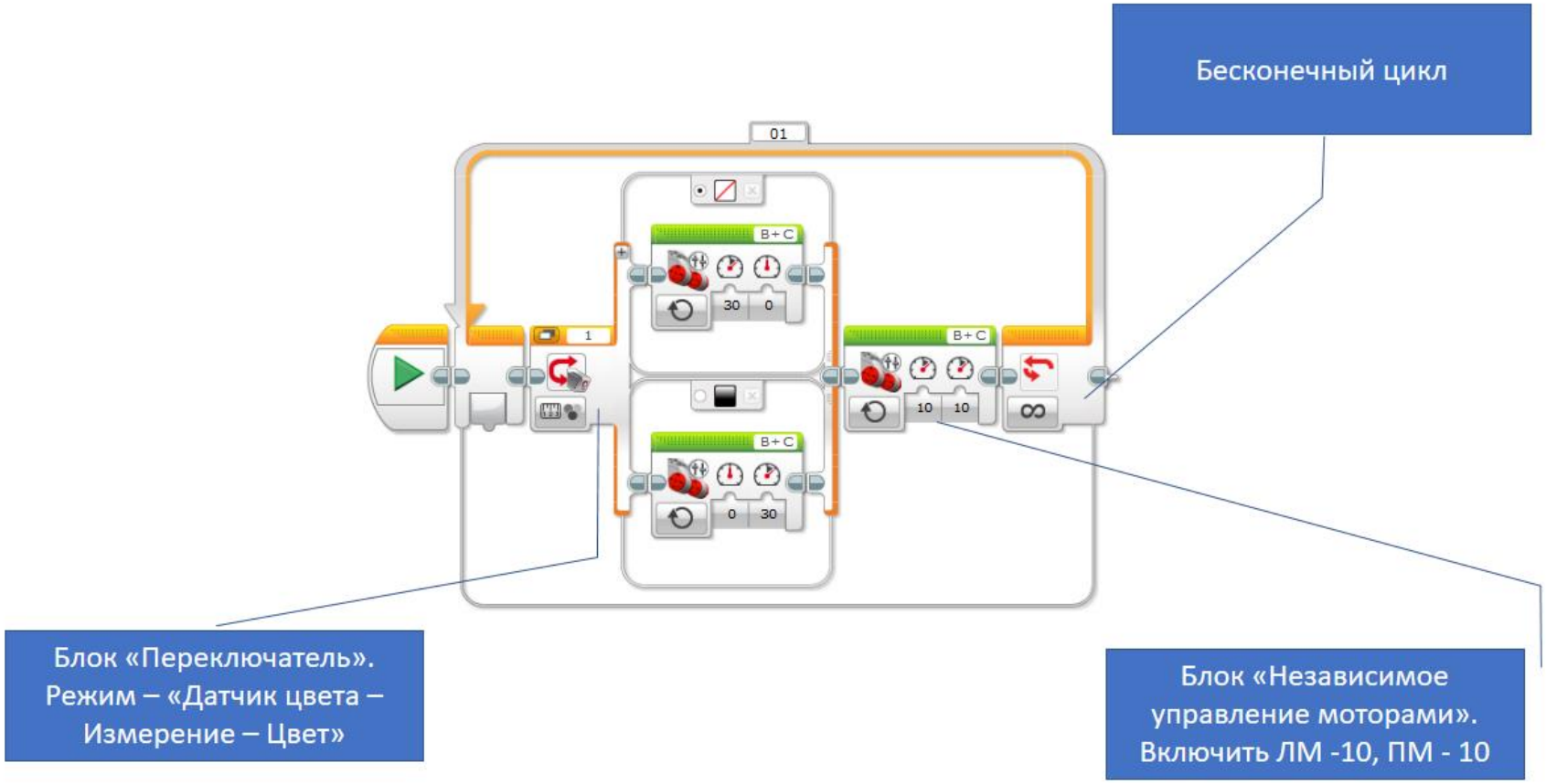
Дескрипторы:

- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок ультразвуковой датчик «Вернуть цвет»
- Используется Блок логика «Равно»
- Используется Блок действия «Ехать вперед»
- Используется Блок цвета: черный и белый цвет.

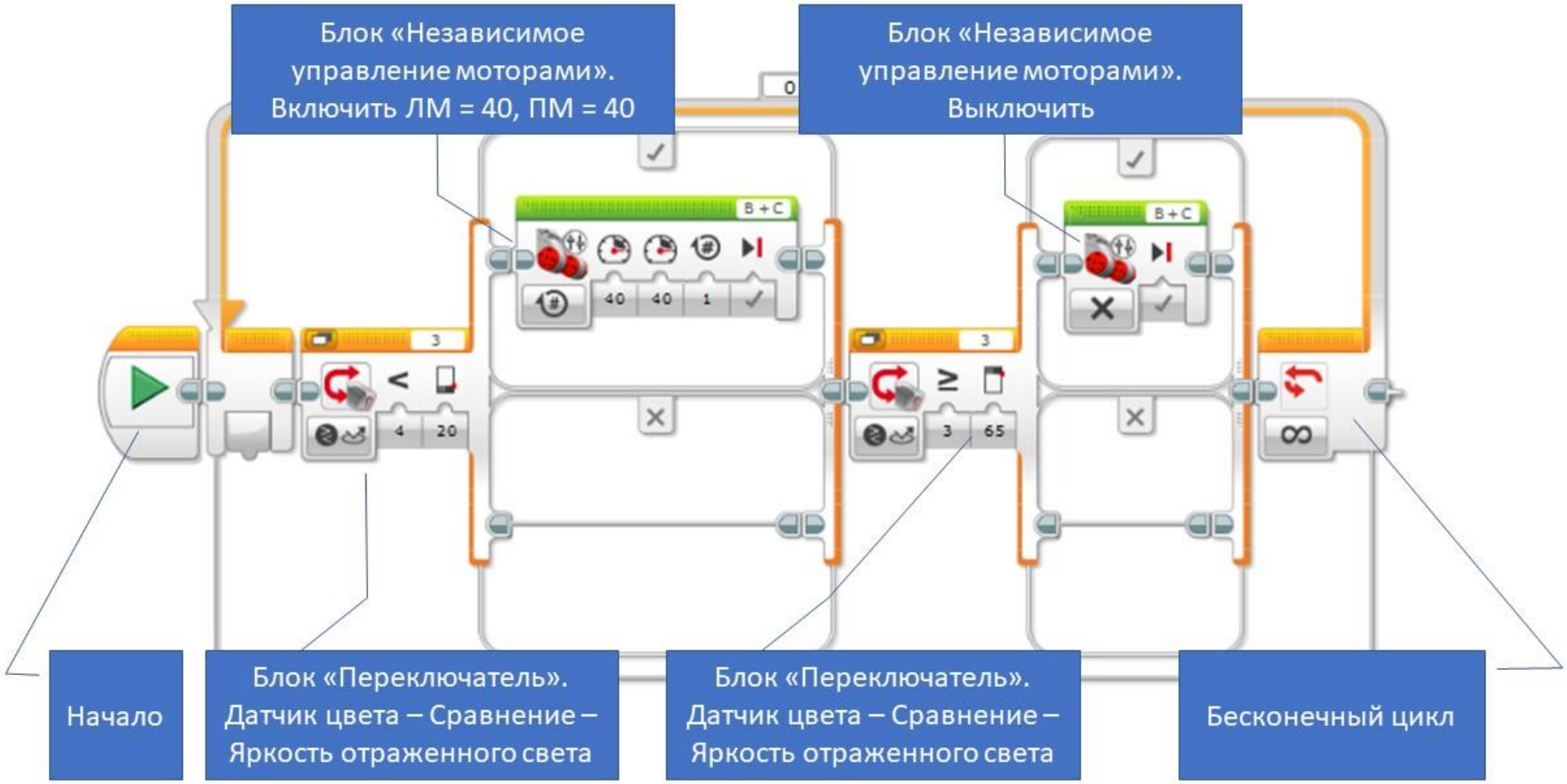
Сравнение инструментов Open Roberta Lab и Lego © Mindstorms EV3.

Блок EV3	Блок Open Roberta Lab	Назначение блока
		Управление моторами
		Цикл
		Ожидание
		Ветвление

Код программы «Движении по линии» для платформы Lego © Mindstorms EV3

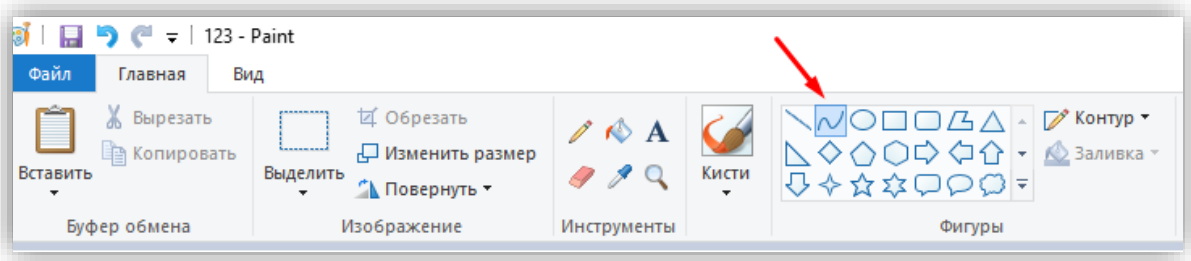


Код программы «Светофор. Определение цвета предмета» для платформы Lego © Mindstorms EV3

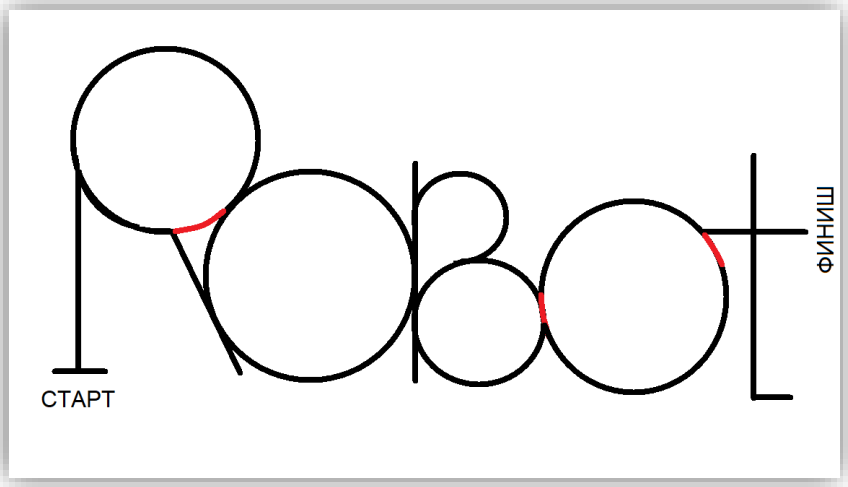


Задание 2. Движение по линии. Подготовка сцены.

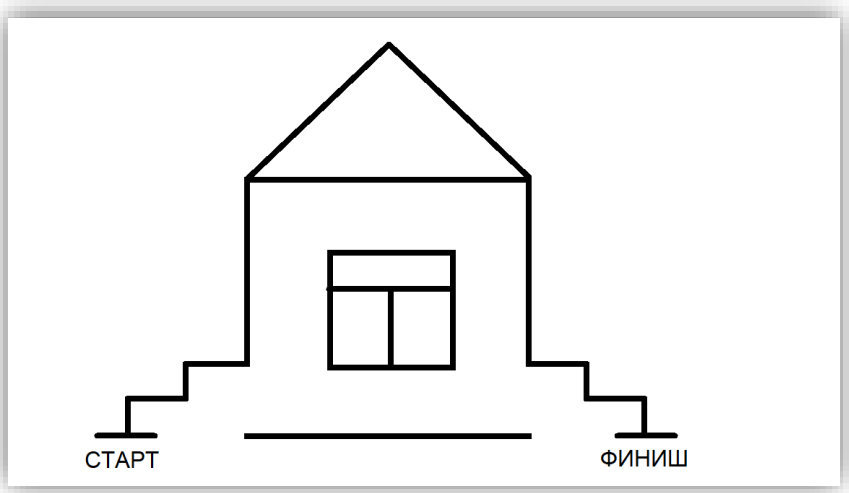
Откройте приложение Paint и используйте любые инструменты: «Линия», «Кривая», «Овал» или «Прямоугольник» в Панели инструментов меню «Главная»



Разработайте траекторию для движения робота (набор фигур и линий) как на рисунке ниже (Примеры 1–2) или создайте свою собственную траекторию используя перечисленные выше инструменты. Сохраните изображения на «Рабочий стол».

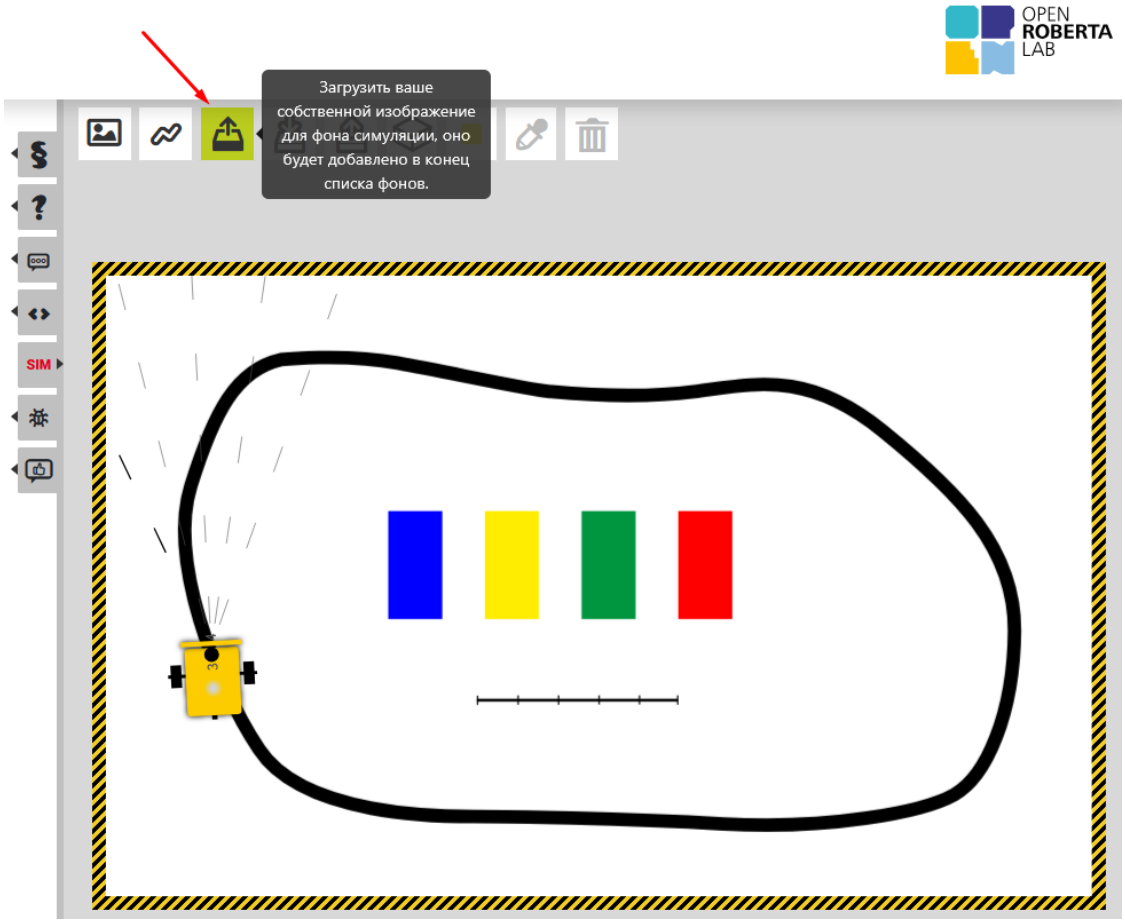


Пример 1. Пример траектории для движения робота

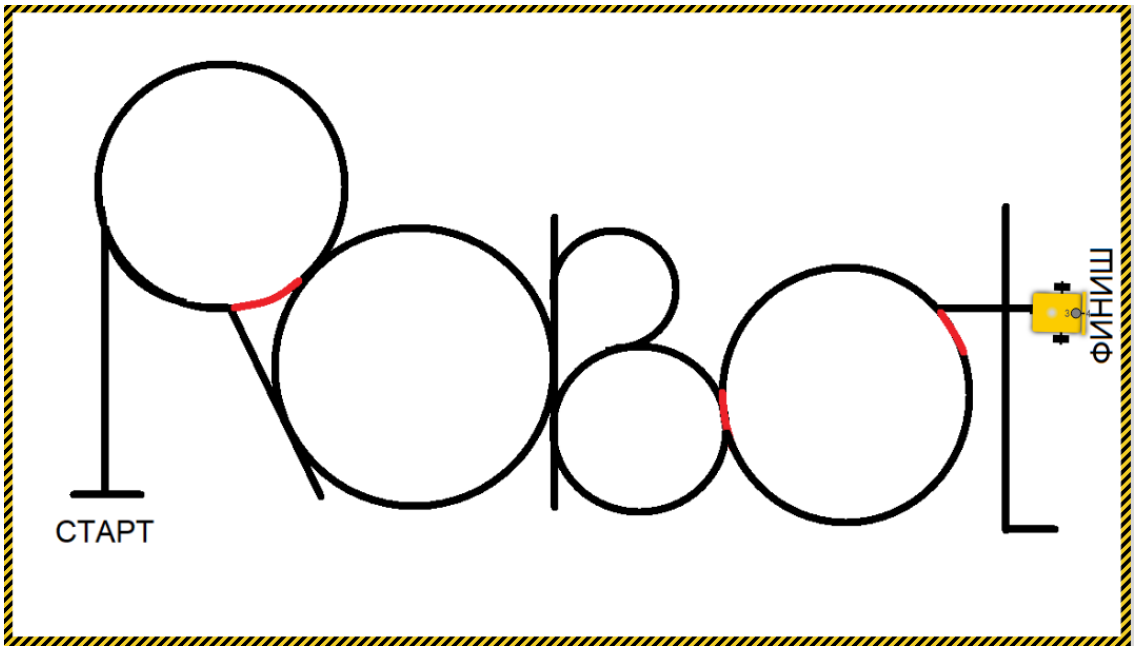


Пример 2. Пример траектории для движения робота

Откройте сайт https://lab.open-roberta.org/ и выберите пункт (показан стрелкой)



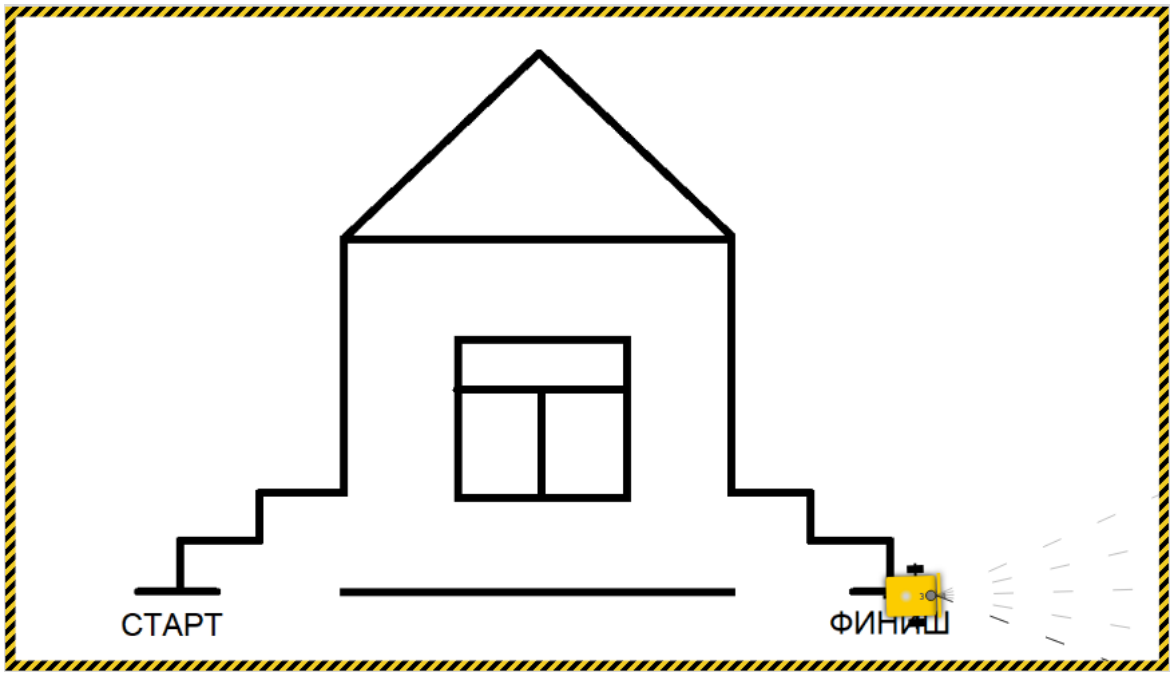
Загрузите изображение, которое вы создали ранее, например:



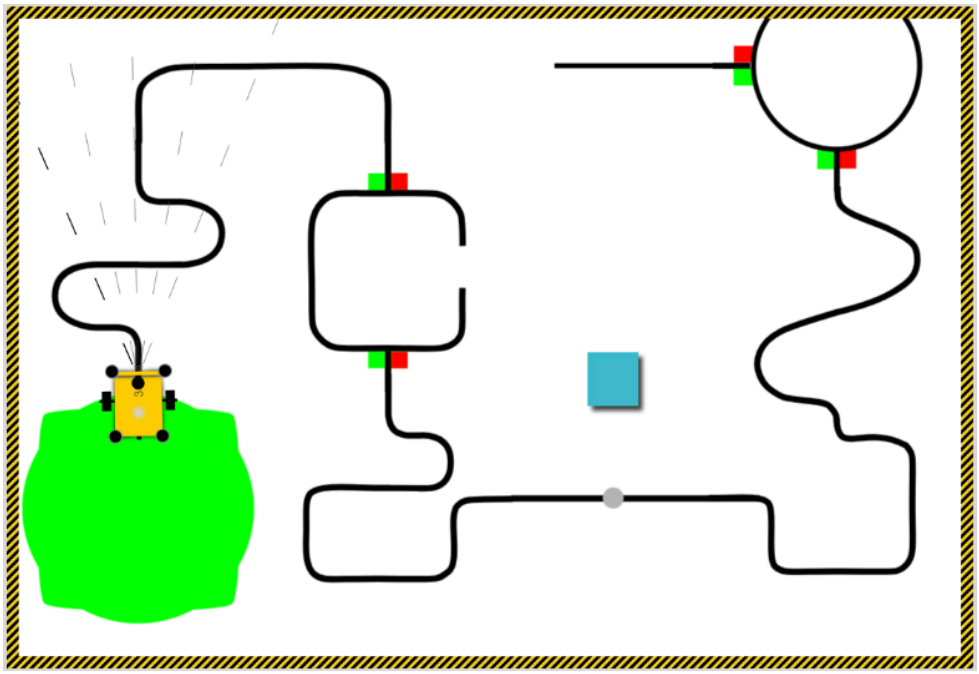
Составьте программу, как показано на рисунке ниже

Нажмите на кнопке «Старт»

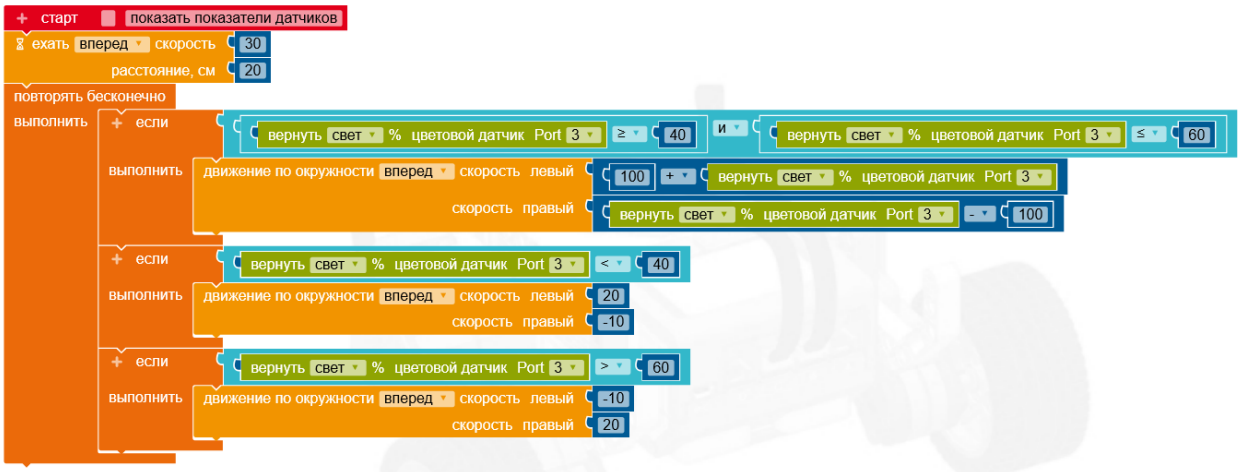
Пример 2



*Дополнительное задание. Загрузите сцену как показано на рисунке ниже



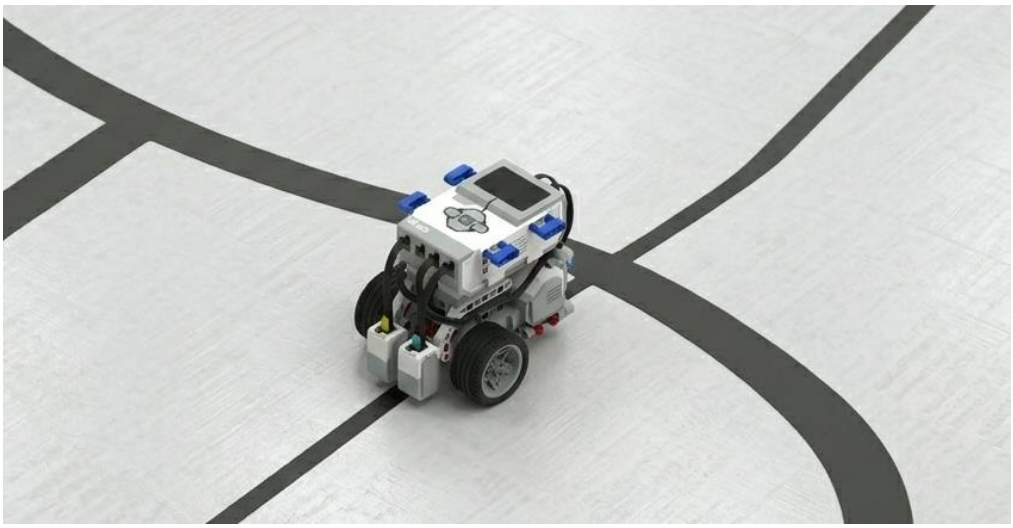
Составьте код программы как на рисунке ниже



Дескрипторы:

- Используется Блок цикл «Повторить бесконечно (выполнить)»
- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок ультразвуковой датчик «Вернуть цвет»
- Используется Блок логика «Равно»
- Используется Блок действия «Ехать вперед»
- Используется Блок цвета: черный и белый цвет.
- Изменяются свойства скорости и градуса в программе

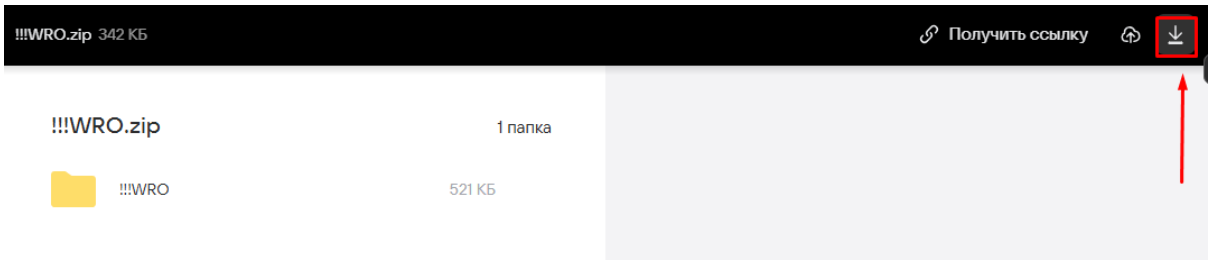
Задание 3. Движение по линии с двумя датчиками (Lego © Mindstorms EV3 ©).



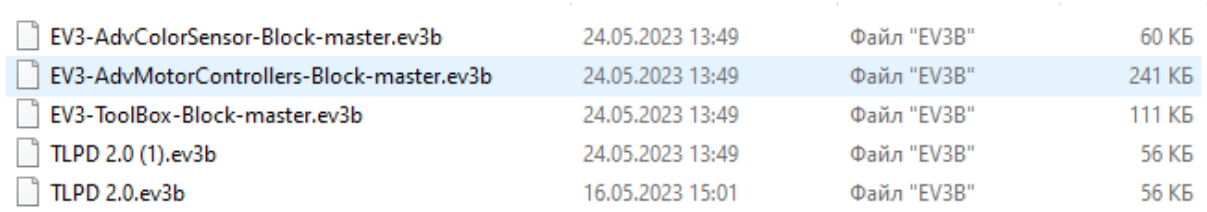
Пример робота Lego © Mindstorms EV3 © с двумя датчиками цвета

Создайте новый проект в среде LEGO MINDSTORMS Education EV3.

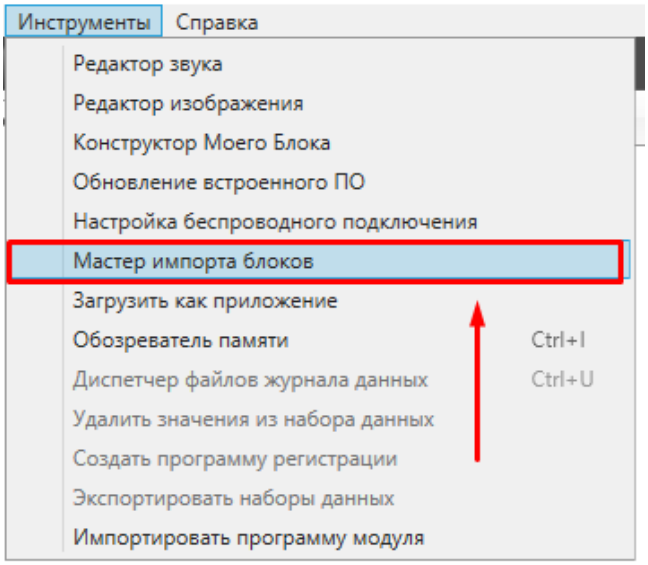
Скачайте по **ссылке** набор компонентов для движения по линии



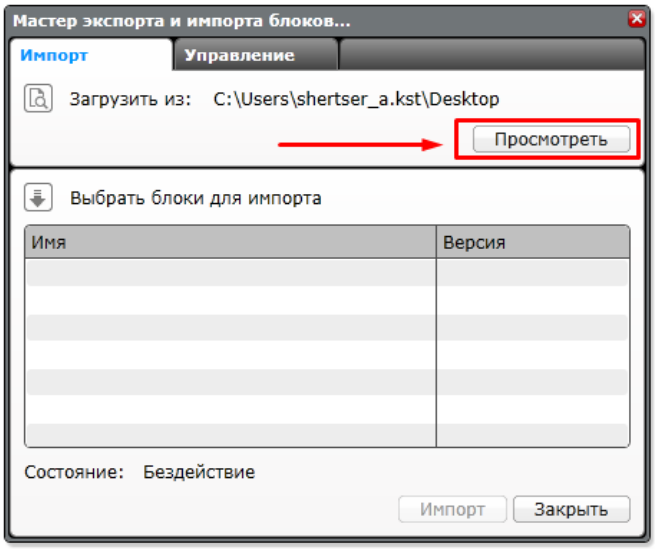
Распакуйте архив в одноименную директорию, например на «Рабочий стол»



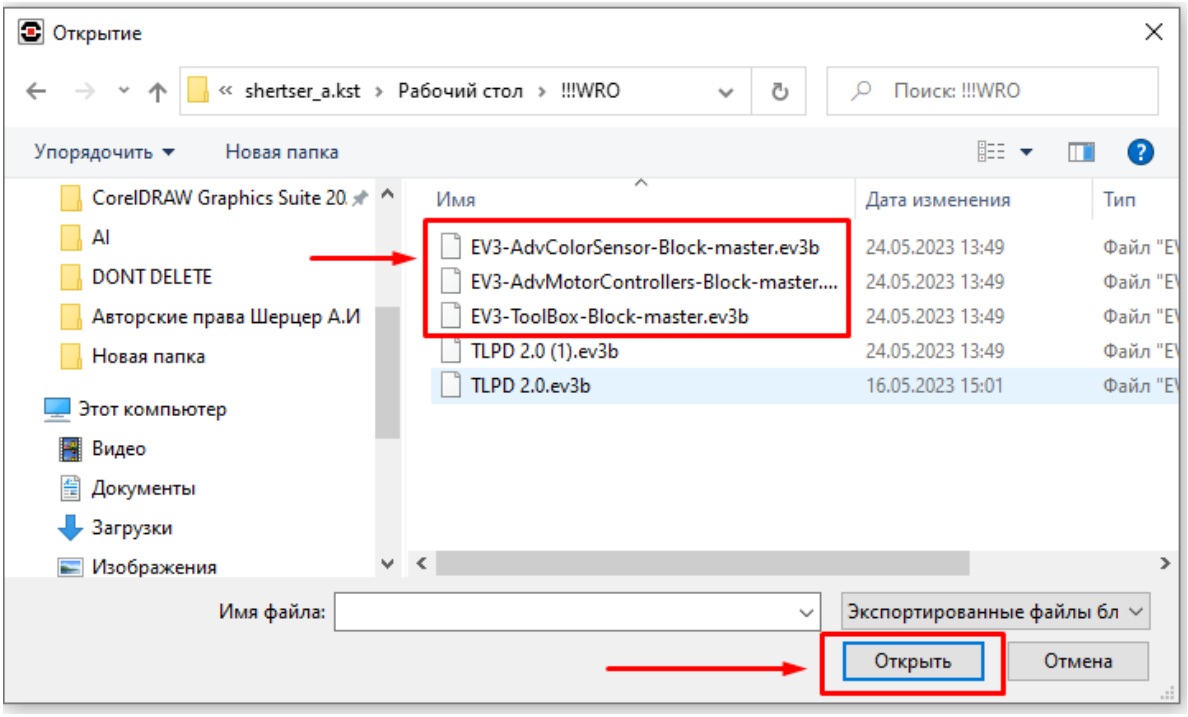
Откройте пункт меню «Инструменты» - «Мастер импорта блоков»



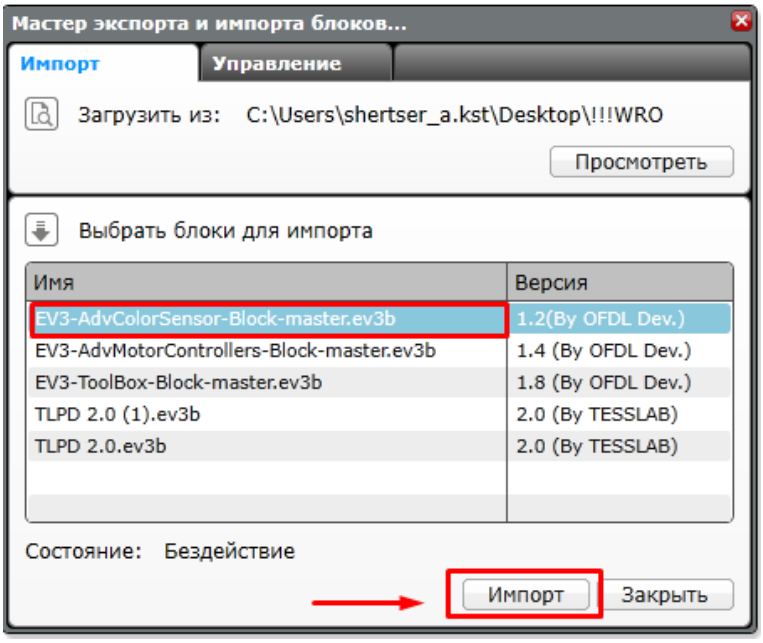
Нажмите на кнопке «Просмотреть»



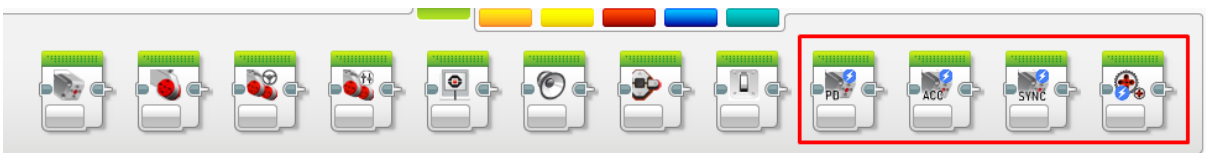
Выберите файлы и нажмите на кнопке «Открыть»



Выберите каждый блок для импорта по – отдельности и нажмите «Импорт» для каждого блока



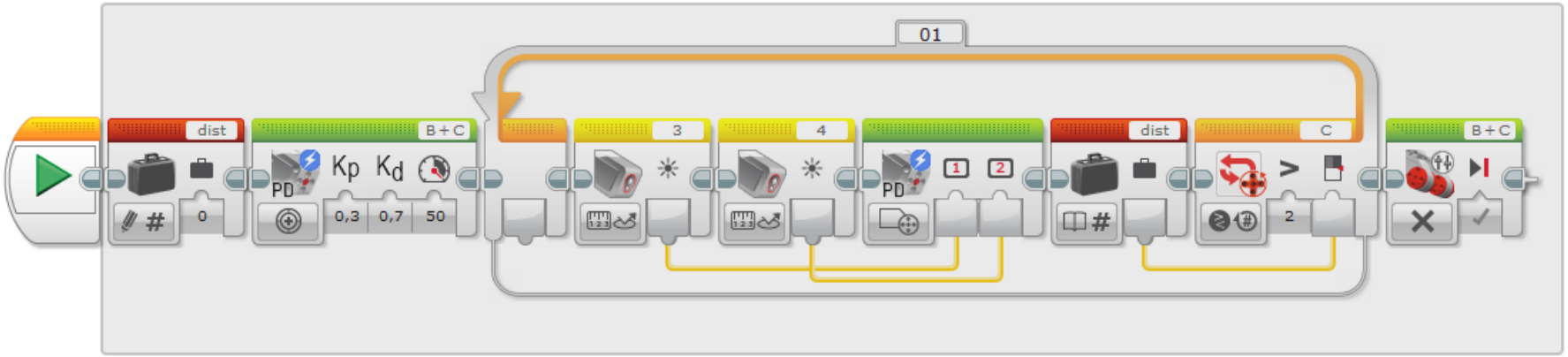
Перезапустите программное обеспечение LEGO MINDSTORMS Education EV3. В панели «Действия» должны появиться новые блоки для управления роботом.



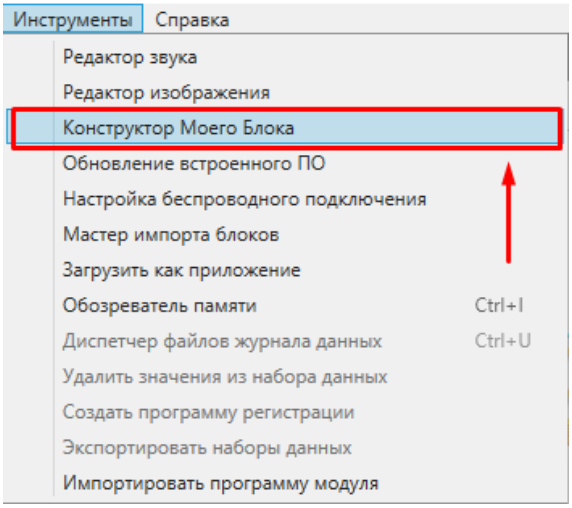
Создайте программу в соответствии с примером на изображении «Часть 1» ниже

Часть 1. Код программы «Движение по линии с 2 датчиками» для платформы Lego © Mindstorms EV3.

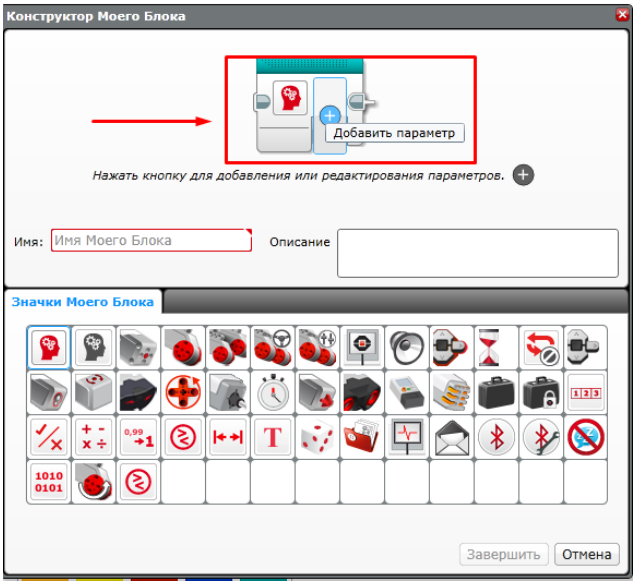
Выделите все блоки кода кроме блока старт

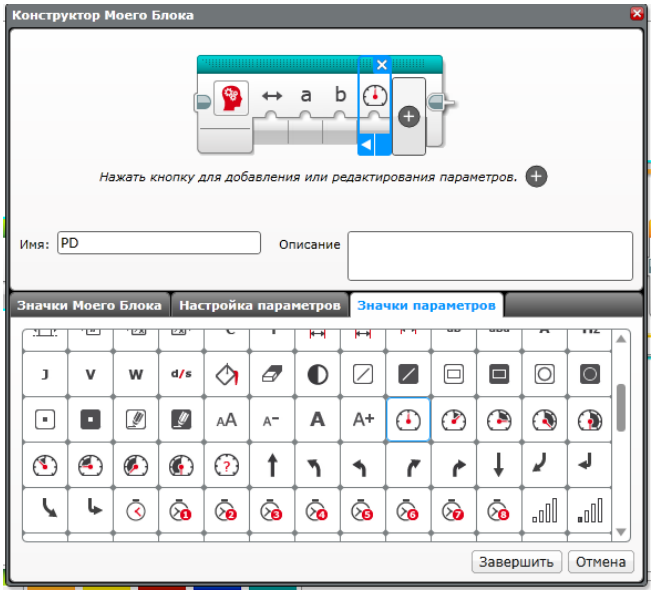


Откройте меню «Инструменты» - «Конструктор Моего Блока» в соответствии с инструкцией

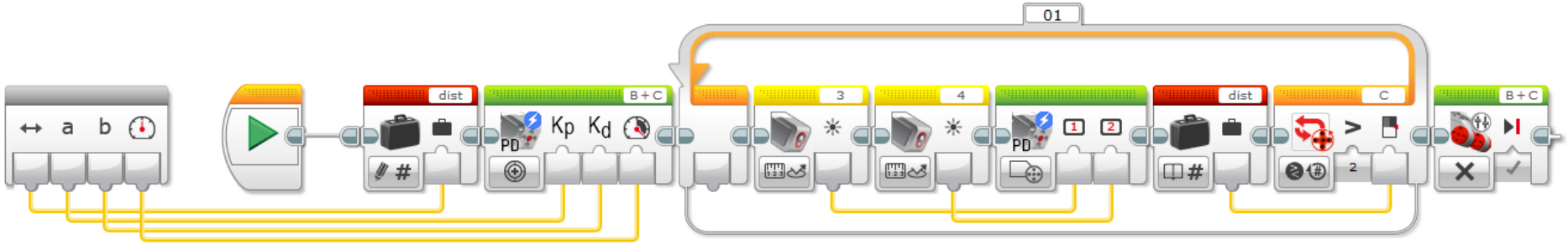


В открывшемся диалоговом окне нажмите четыре раза на кнопке показанной стрелкой на рисунке. При необходимости вы можете изменить значки параметров используя вкладку «Настройка параметров». Введите желаемое имя блока в текстовое поле с подсказкой «Имя Моего Блока» и нажмите на кнопке «Завершить». Пример оформления ниже.





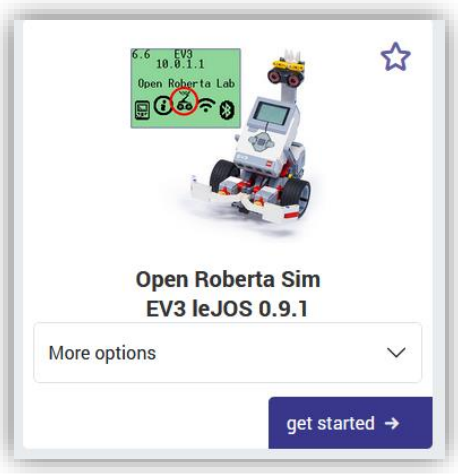
Часть 2. Код программы «Движение по линии с 2 датчиками» для платформы Lego © Mindstorms EV3. В программе добавится новый серый блок. Подключите с помощью соединительных линий ваш собственный блок серого цвета к основной программе в соответствии с примером на изображении ниже.



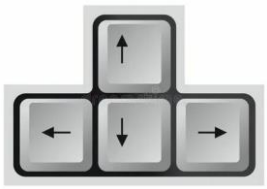
Часть 3. Код программы «Движение по линии с 2 датчиками» для платформы Lego © Mindstorms EV3. Обратите ваше внимание на то, что в вашем проекте две вкладки. Одна называется в соответствии с названием вашего блока в «Конструкторе Моего Блока», который вы определили на предыдущем этапе, а вторая имеет название Program. Со вкладкой Program нам и предстоит работать. Введите соответствующие параметрам вашего робота, поверхности стола, и освещения значения как показано на рисунке ниже. Запустите программу и наслаждайтесь результатом.

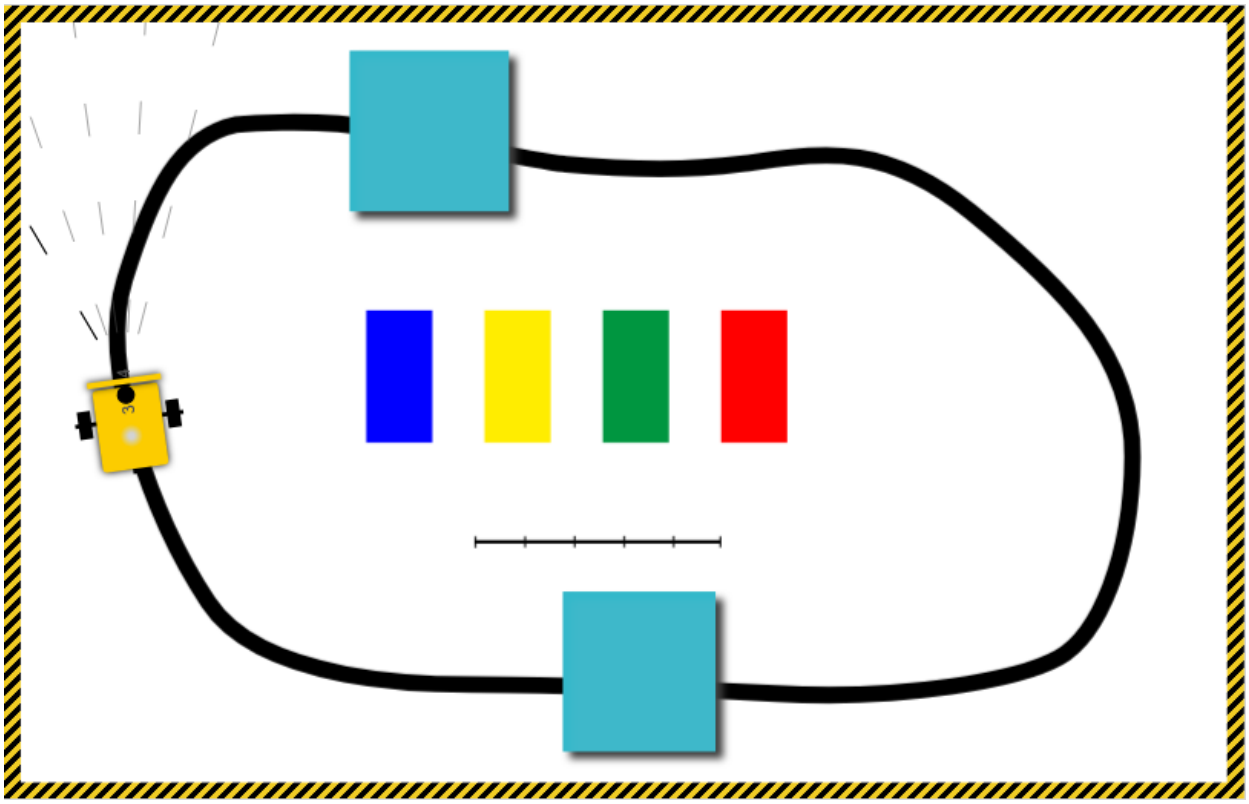
Тема. Биатлон (движение по траектории с препятствиями).

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Задание 1. Настроить сцену как на картинке ниже. Разместите робота и прямоугольные блоки препятствий вдоль траектории, как в примере, представленном на изображении ниже. Используйте кнопки «стрелка влево» или «стрелка вправо» на клавиатуре и компьютерную мышь, чтобы изменить позицию робота. Устанавливаем робота таким образом, чтобы он стоял на черной линии (как на рисунке ниже).





Задание 2. Откройте раздел функций (указан стрелкой на рисунке 1 «Раздел функций») и добавьте в программу блок «Выполнить что-то» (как показано на рисунке 2 «Блок выполнить что-то»). Измените текст внутри блока «Выполнить что-то» на «Movearoundtext» в соответствии с примером представленном на рисунке 3 «Измененный текст». Составьте программу по примеру, представленному на рисунке 4 «Код объезда препятствий». Обратите внимание, что в следующем задании вы можете изменять значения угла поворота и скорости движения робота в соответствии с размерами блоков препятствий.

Рисунок 1. Раздел функций	Рисунок 2. Блок «Выполнить что-то»

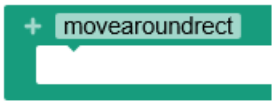
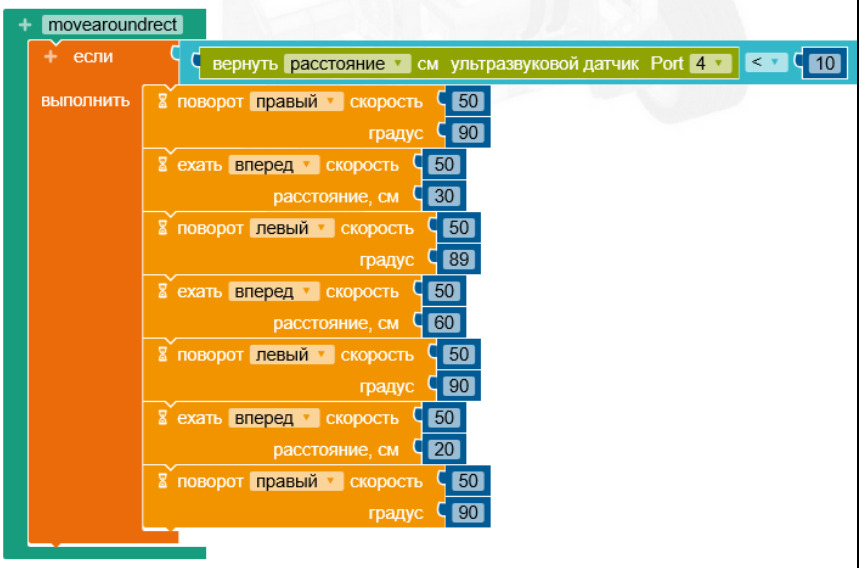

	

Рисунок 3. Измененный текст	Рисунок 4. Код объзда препятствия.

Задание 3. Составьте итоговый вид программы, показанный на рисунке 5 ниже. Используйте блок с названием «Movearoundrect» который вы создали в разделе функций.

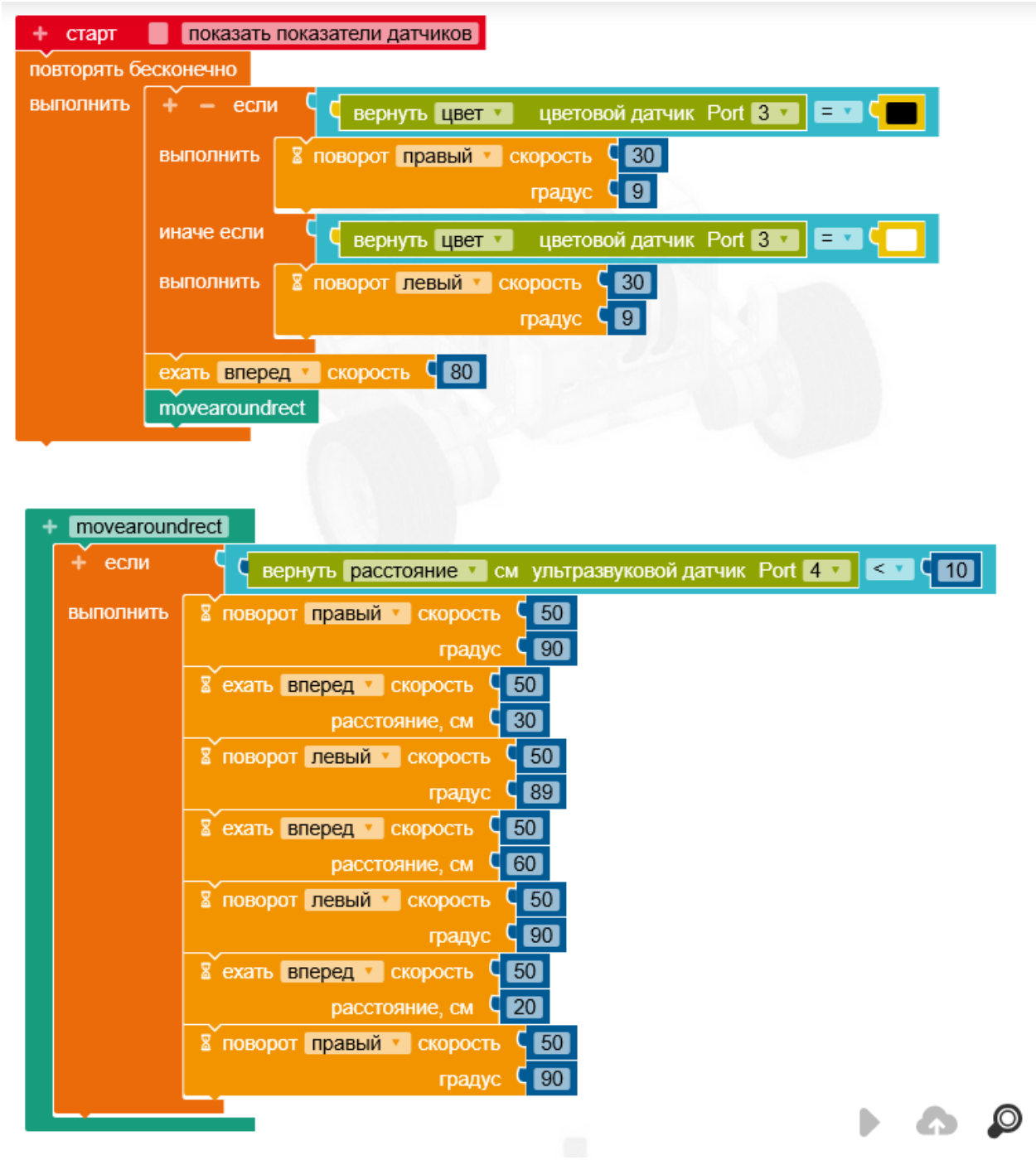


Рисунок 5. Итоговый вид программы

Нажмите на кнопке «Старт» 

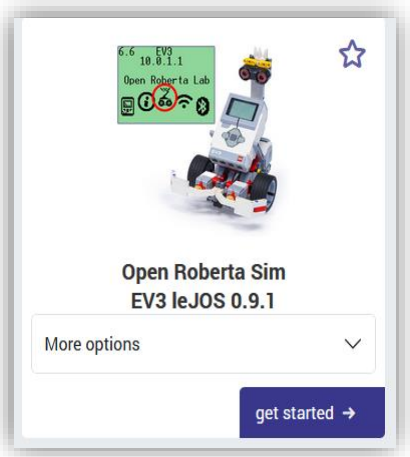
Дескрипторы:

- Используется Блок цикл «Повторить бесконечно (выполнить)»

- Используется Блок условие «Если (выполнить) Иначе (выполнить)»
- Используется Блок датчик цвета «Вернуть цвет»
- Используется Блок ультразвуковой датчик «Вернуть расстояние»
- Используется Блок логика «Равно»
- Используется Блок действия «Ехать вперед»
- Используется Блок цвета: черный и белый цвет.
- Использует блок препятствий в игровой сцене
- Использует Блок «Выполнить что-то» в разделе «Функции»
- Использует блок с названием функции при разработке программы

Тема. Вывод текстовой и аудио информации.

Задание 1. Откройте сайт https://lab.open-roberta.org/ и выберите пункт как на рисунке ниже. Для настройки сцены выполните все следующие действия в строгой последовательности.



1) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для изменения рисунка игрового поля («Среда») в соответствии с результатом показанным на рисунке 1 «Игровое поле»

2) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для отображения вида от лица робота, в четком соответствии с результатом показанным на рисунке 1 «Игровое поле»

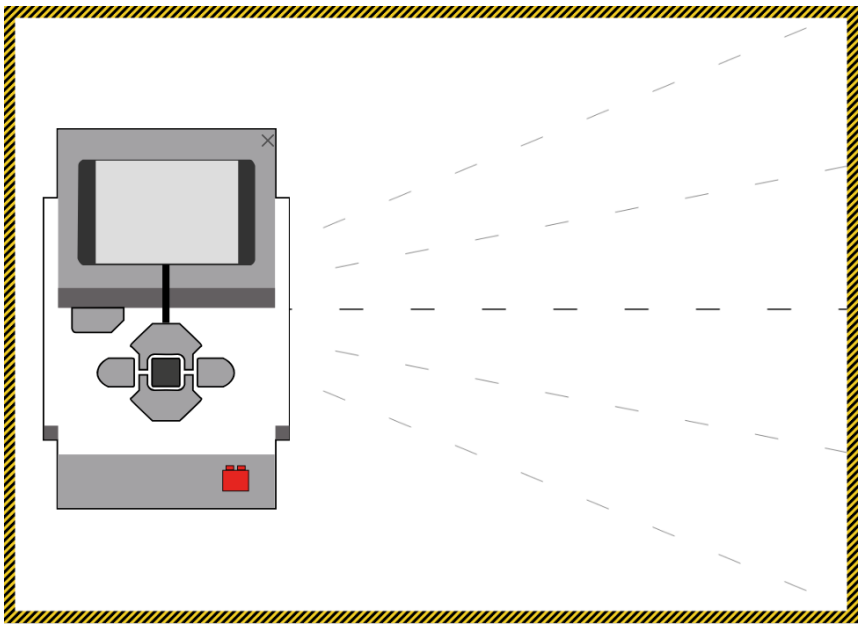


Рисунок 1. Игровое поле

Задание 2. Составьте программу по примеру, представленному на рисунке 2 «Пример программы «Вывод текста на экран». Используйте разделы «Действие» и «Контроль». Измените текст на «Добро пожаловать!»

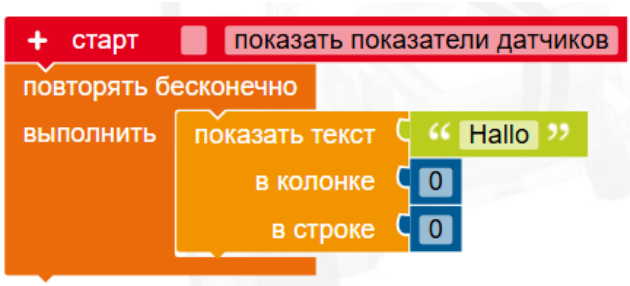


Рисунок 2. Пример программы «Вывод текста на экран»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести заданный текст в левом верхнем углу вида от лица робота как показано ниже в примере на рисунке 3 «Вывод текста на экран блока».

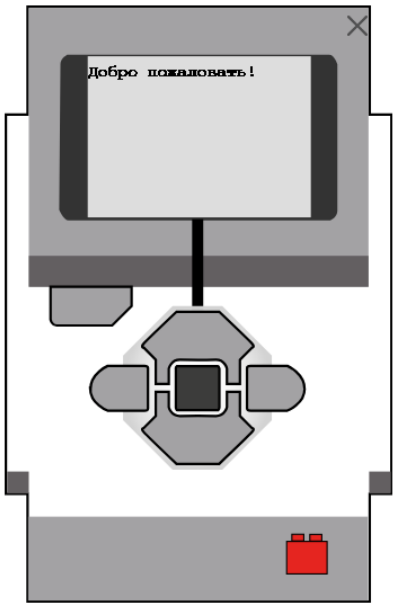


Рисунок 3. Пример «Вывод текста на экран блока»

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Показать текст» в колонке 0, строке 0
- Программа запускается и текст выводится на экран блока

Остановите программу после выполнения задания.

Задание 3. Дополните программу по примеру, представленному на рисунке 3 «Пример программы «Вывод текста на экран с озвучкой». Используйте разделы «Действие» и «Контроль». Добавьте один блок «Показать текст» и три блока «Say». Выполните следующие действия по порядку.

i) Измените текст «Hallo» в блоке «Показать текст» на текст «3 закона робототехники»;
ii) Измените текст «Hallo» в первом блоке «Say» на текст «Робот не может причинить вред человеку»;
iii) Измените текст «Hallo» во втором блоке (ниже) «Say» на текст «Робот должен повиноваться всем приказам»;
iv) Измените текст «Hallo» в третьем блоке (ниже) «Say» на текст «Робот должен заботиться о своей безопасности».

Три закона роботехники в научной фантастике — обязательные правила поведения для роботов, впервые сформулированные Айзеком Азимовым в рассказе «Хоровод» (1942). Законы робототехники Азимова построены по аналогии с категорическим императивом Канта и представляют собой наиболее известную попытку создания этических правил для роботов. По замыслу автора Три закона должны наложить этический оттенок на поведение машины.

Законы гласят:

1. Робот не может причинить вред человеку или своим бездействием допустить, чтобы человеку был причинён вред.
2. Робот должен повиноваться всем приказам, которые даёт человек, кроме тех случаев, когда эти приказы противоречат Первому Закону.
3. Робот должен заботиться о своей безопасности в той мере, в которой это не противоречит Первому или Второму Законам.

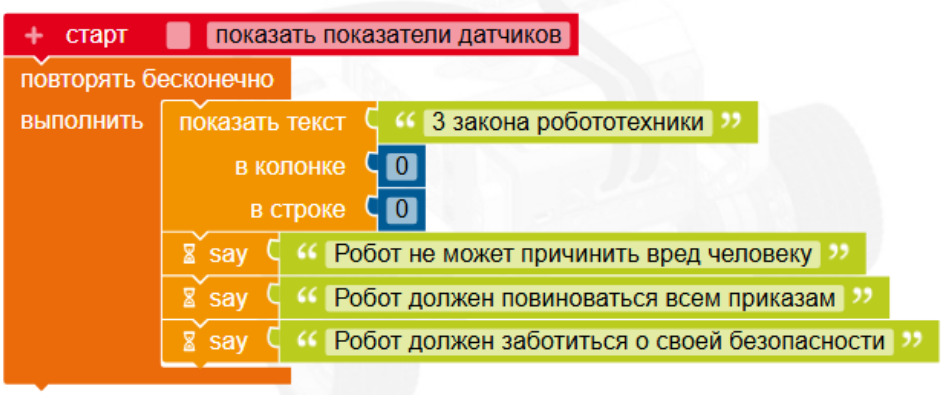


Рисунок 3. Пример «Вывод текста на экран блока»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести заданный текст в левом верхнем углу вида от лица робота как показано ниже в примере на рисунке 4 «Вывод текста на экран блока», и воспроизвести текст в форме искусственной речи.

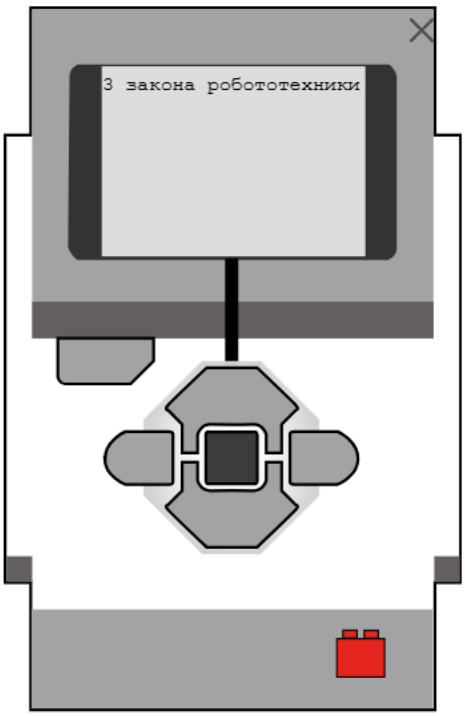


Рисунок 4. Пример «Вывод текста на экран блока»

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Показать текст» в колонке 0, строке 0
- Используется действие «Say» в колонке 0, строке 0
- Изменяется текст в действиях «Показать текст» и «Say»
- Программа запускается: текст выводится на экран блока
- Программа запускается: текст преобразуется в речь и воспроизводится

Остановите программу после выполнения задания.

Задание 4. Измените программу по примеру, представленному на рисунке 5 «Пример программы «Воспроизведение мелодии «В траве сидел кузнечик». Используйте разделы «Действие» и «Контроль». Добавьте один блок «Показать текст» и несколько блоков «Проиграть» изменяя длительность звучания ноты. Выполните следующие действия по порядку.

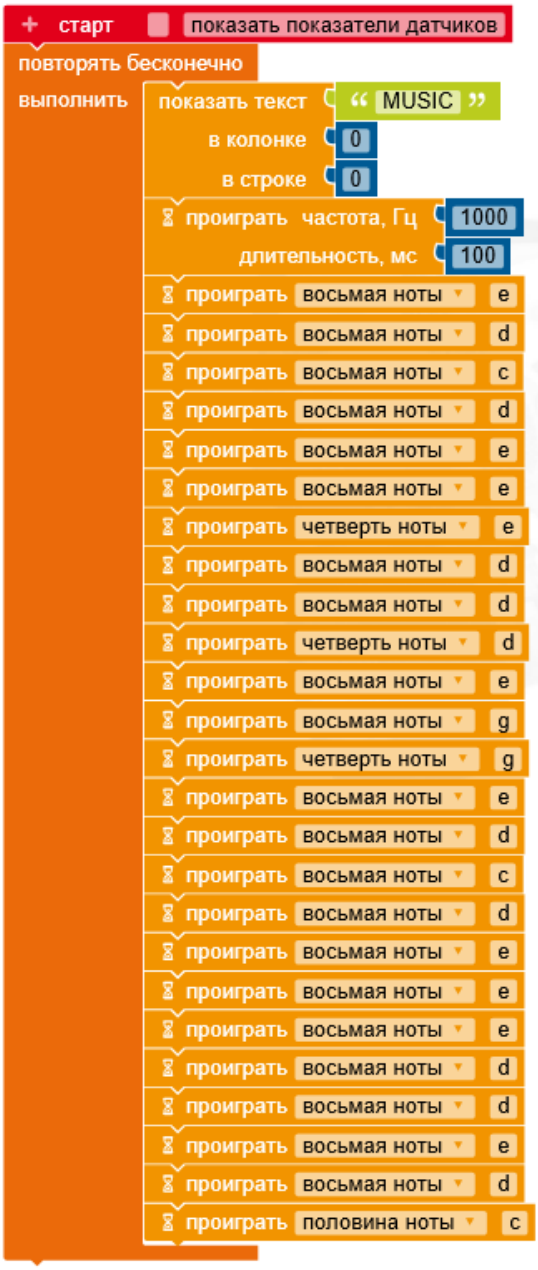


Рисунок 5 «Пример программы «Воспроизведение мелодии «В траве сидел кузнечик»

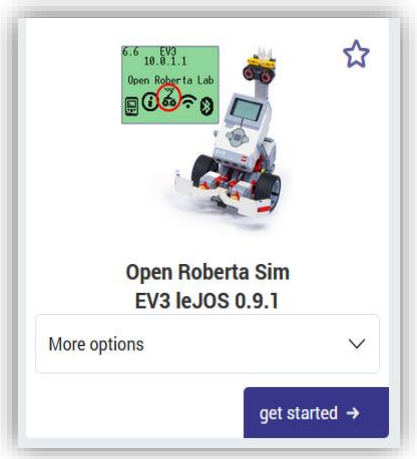
Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используется контроль «Повторить бесконечно (выполнить)»
- Используется действие «Показать текст» в колонке 0, строке 0
- Используется действие «Проиграть» ноту различной длительности
- Программа запускается: текст выводится на экран блока, мелодия воспроизводится

Остановите программу после выполнения задания.

Тема. Арифметические действия и работа с числами.

Задание 1. Откройте сайт https://lab.open-roberta.org/ и выберите пункт как на рисунке ниже. Для настройки сцены выполните все следующие действия в строгой последовательности.



3) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для изменения рисунка игрового поля («Среда») в соответствии с результатом показанным на рисунке 1 «Игровое поле»

4) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для отображения вида от лица робота, в четком соответствии с результатом показанным на рисунке 1 «Игровое поле»

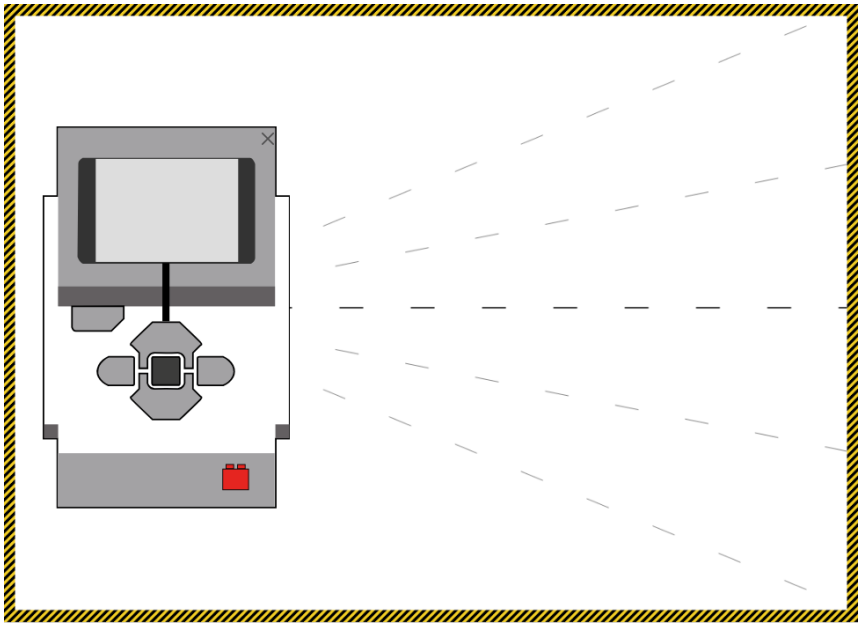


Рисунок 1. Игровое поле

Задание 2. Составьте программу по примеру, представленному на рисунке 2 «Пример программы «Вывод суммы двух чисел на экран». Используйте разделы «Действие» и «Контроль» и «Переменные». Для добавления новых переменных нажмите на кнопку «+» рядом с надписью «Старт». В дополнение измените значения чисел для переменной «x» и отдельно для переменной «y» на другие.

Рисунок 2. Пример программы «Вывод суммы двух чисел на экран»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести сумму двух чисел в левом верхнем углу вида от лица робота как показано ниже в примере на рисунке 3 «Вывод суммы чисел на экран блока».

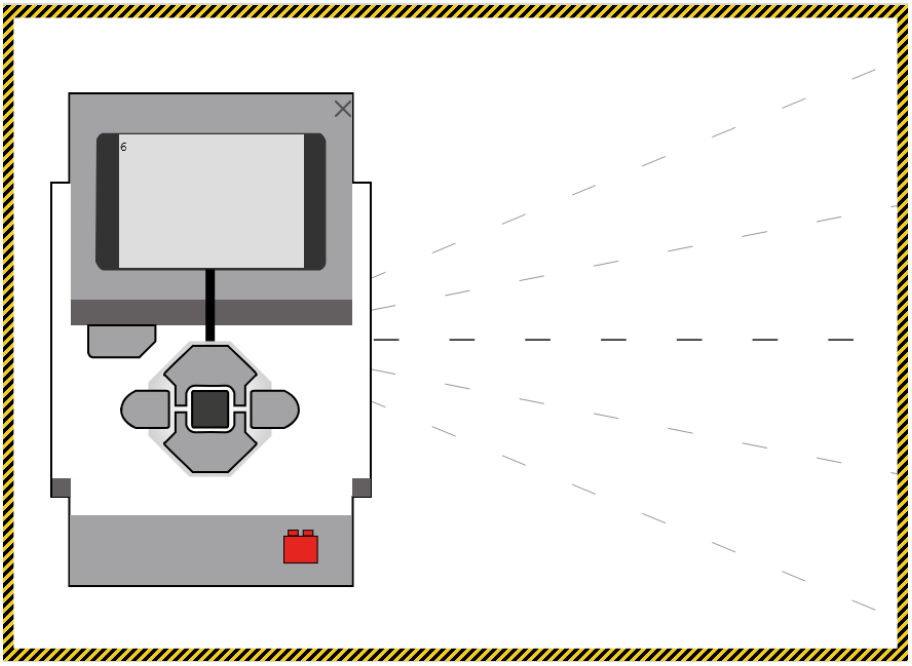


Рисунок 3. Пример «Вывод суммы чисел на экран блока»

Остановите программу после выполнения задания.

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используются переменные (x,y,z) и команда «Присвоить» из раздела «Переменные»
- Используется команда «Показать текст» в колонке 0, строке 0 из раздела «Действия»
- Используется команда «Ждать» из раздела «Контроль»
- Программа запускается и текст выводится на экран блока

Задание 3. Составьте программу по примеру, представленному на рисунке 4 «Пример программы «Вывод последовательности чисел на экран». Используйте разделы «Действие» и «Контроль» и «Переменные». Для добавления новых переменных нажмите на кнопку «+» рядом с надписью «Старт».

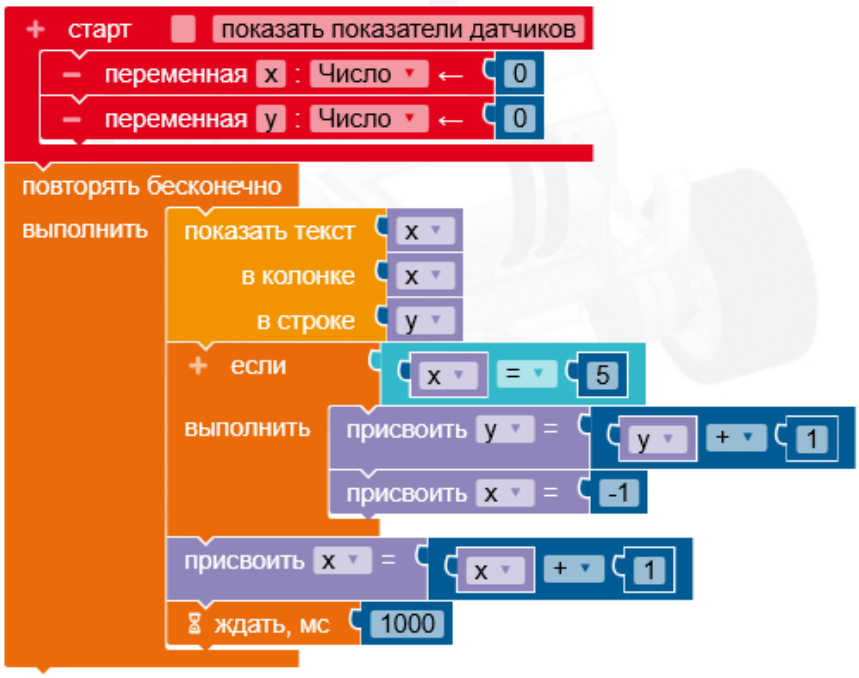


Рисунок 4. Пример программы «Вывод последовательности чисел на экран»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести числа от 0 до 5 вида от лица робота как показано ниже в примере на рисунке 5 «Вывод последовательности чисел на экран блока».

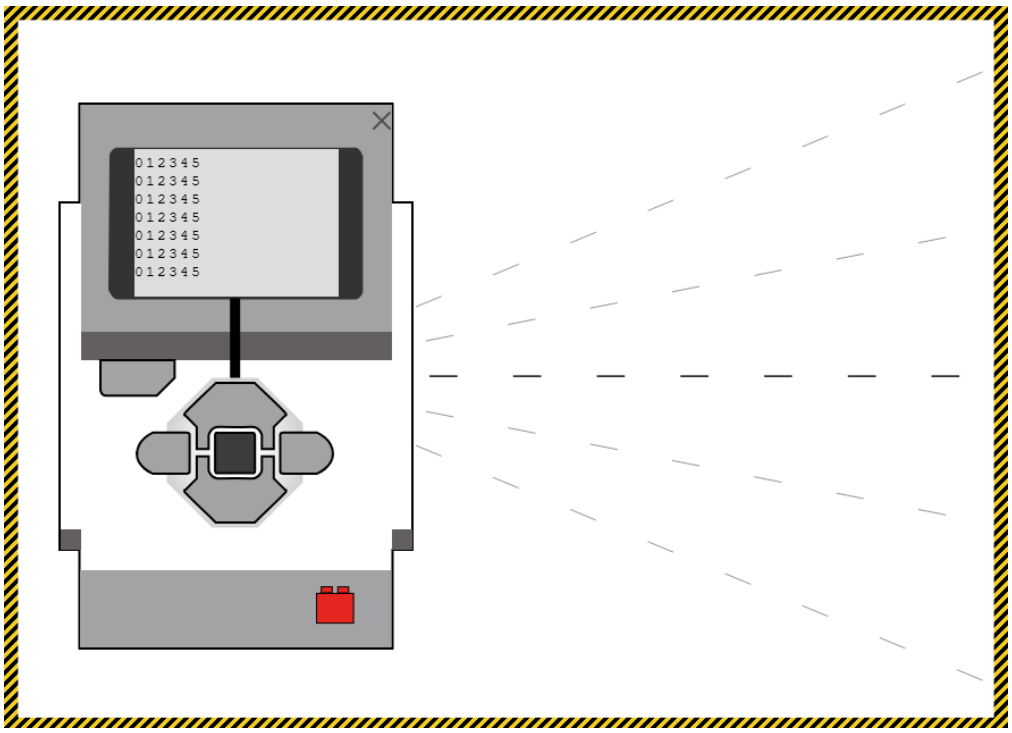


Рисунок 5. Пример «Вывод последовательности чисел на экран блока».

Измените программу как показано на рисунке 6 «Пример программы «Вывод последовательности чисел на экран». Используйте разделы «Действие» и «Контроль» и «Переменные». Объясните разницу между двумя программами представленными на рисунке 4 и на рисунке 6. Сделайте выводы.

Рисунок 6. Пример программы «Вывод последовательности чисел на экран»

Остановите программу после выполнения задания.

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используются переменные (x,y,z) и команда «Присвоить» из раздела «Переменные»
- Используется команда «Показать текст» в колонке 0, строке 0 из раздела «Действия»
- Используется команда «Ждать» из раздела «Контроль»
- Используется команда-оператор «Если – выполнить» из раздела «Контроль»
- Используется команда-оператор «Повторять бесконечно» из раздела «Контроль»
- Используется команда-оператор «Присваивание» из раздела «Логика» (блок «Равно»)
- Программа запускается и текст выводится на экран блока

Задание 4. Составьте программу по примеру, представленному на рисунке 7 «Пример программы «Вывод чисел в форме треугольника на экран». Используйте разделы «Действие» и «Контроль» и «Переменные». Для добавления новых переменных нажмите на кнопку «+» рядом с надписью «Старт».

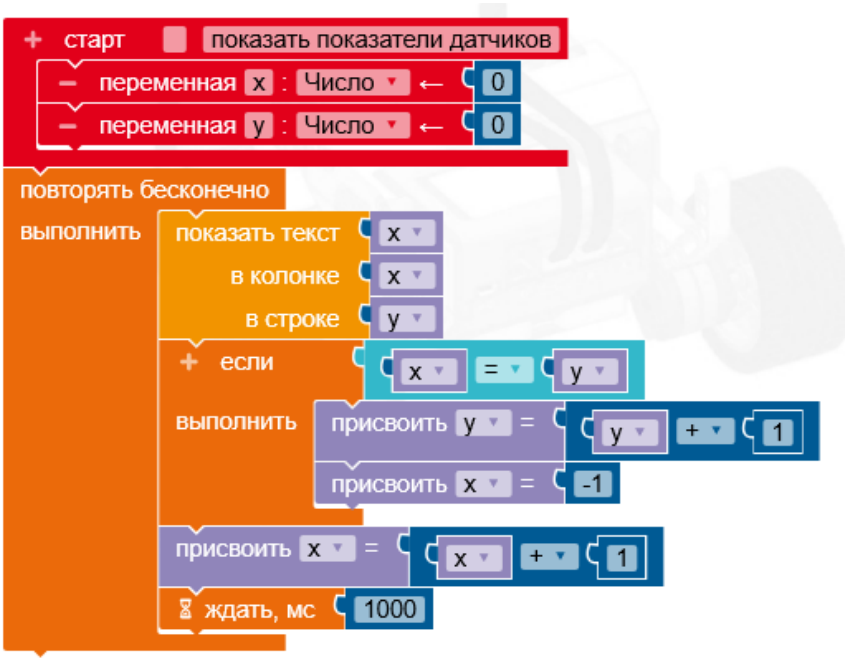


Рисунок 7. Пример программы «Вывод последовательности чисел в форме треугольника на экран»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести числа от 0 до 5 вида от лица робота как показано ниже в примере на рисунке 8 «Вывод последовательности чисел в форме треугольника на экран блока».

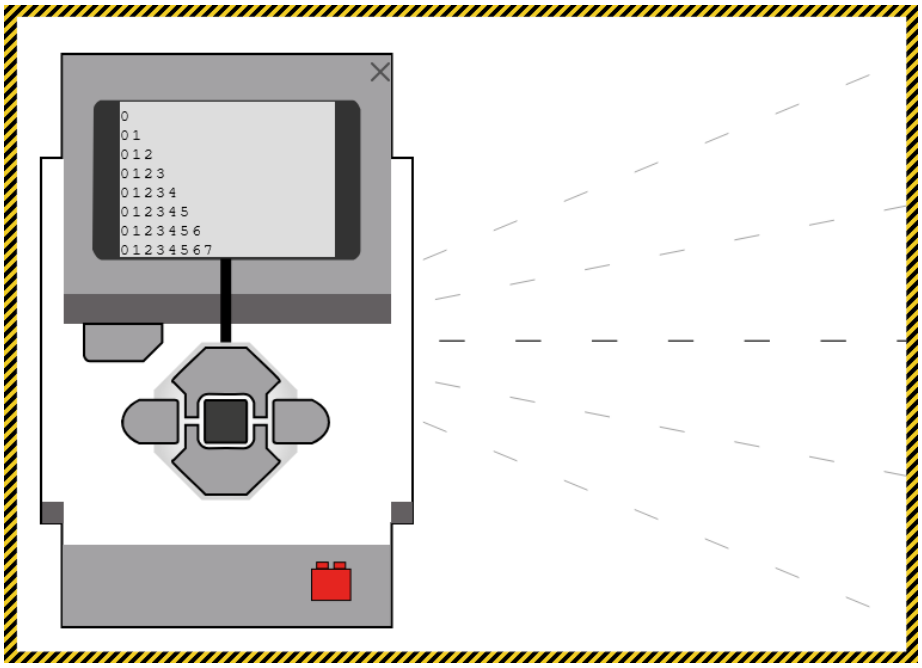


Рисунок 8. Пример «Вывод последовательности чисел в форме треугольника на экран блока».

Остановите программу после выполнения задания.

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используются переменные (x,y,z) и команда «Присвоить» из раздела «Переменные»
- Используется команда «Показать текст» в колонке 0, строке 0 из раздела «Действия»
- Используется команда «Ждать» из раздела «Контроль»
- Используется команда-оператор «Если – выполнить» из раздела «Контроль»
- Используется команда-оператор «Повторять бесконечно» из раздела «Контроль»
- Используется команда-оператор «Присваивание» из раздела «Логика» (блок «Равно»)
- Программа запускается и текст выводится на экран блока

Задание 5. Составьте программу по примеру, представленному на рисунке 9 «Пример программы «Вывод чисел в форме треугольника на экран». Используйте разделы «Действие» и «Контроль» и «Переменные». Для добавления новых переменных нажмите на кнопку «+» рядом с надписью «Старт». Объясните разницу между двумя программами представленными на рисунке 7 и на рисунке 9. Сделайте выводы.

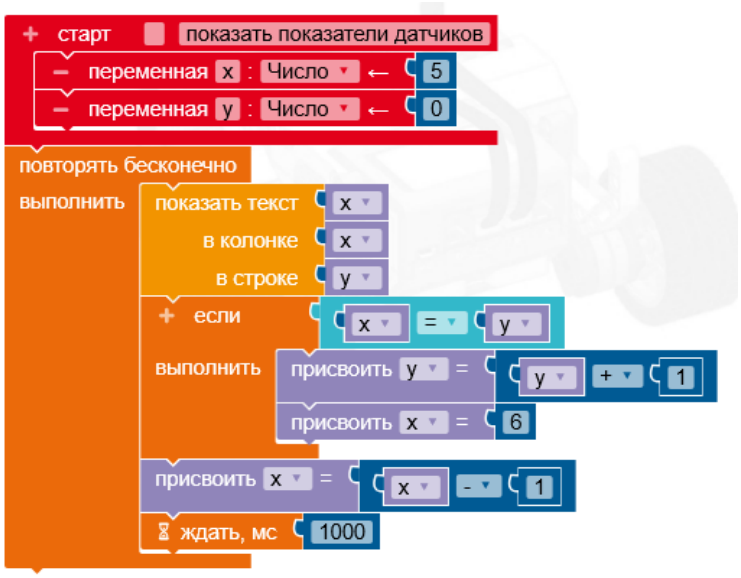


Рисунок 9. Пример программы «Вывод последовательности чисел в форме треугольника на экран»

Для проверки нажмите на кнопке «Старт».

Ваша программа должна вывести числа от 0 до 5 вида от лица робота как показано ниже в примере на рисунке 10 «Вывод последовательности чисел в форме треугольника на экран блока».

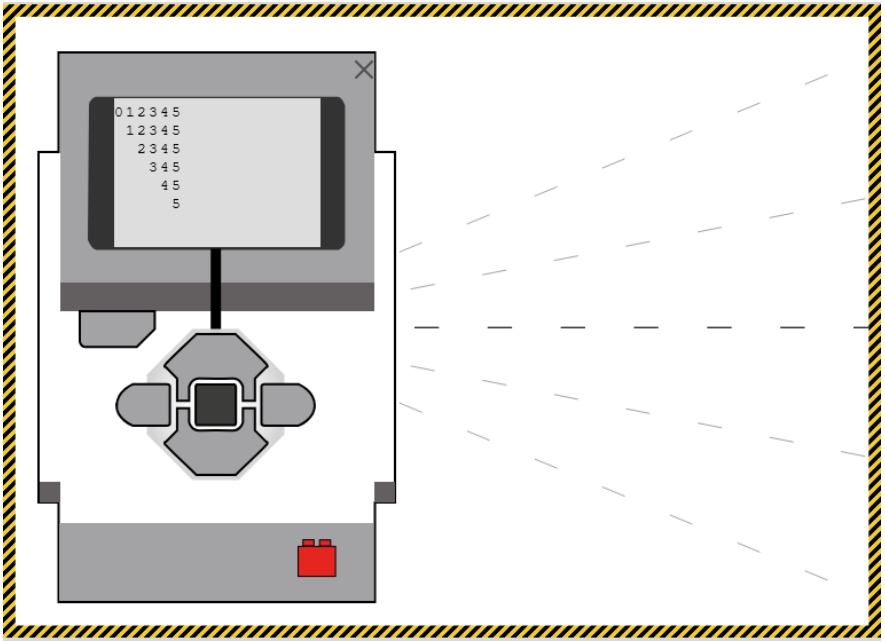


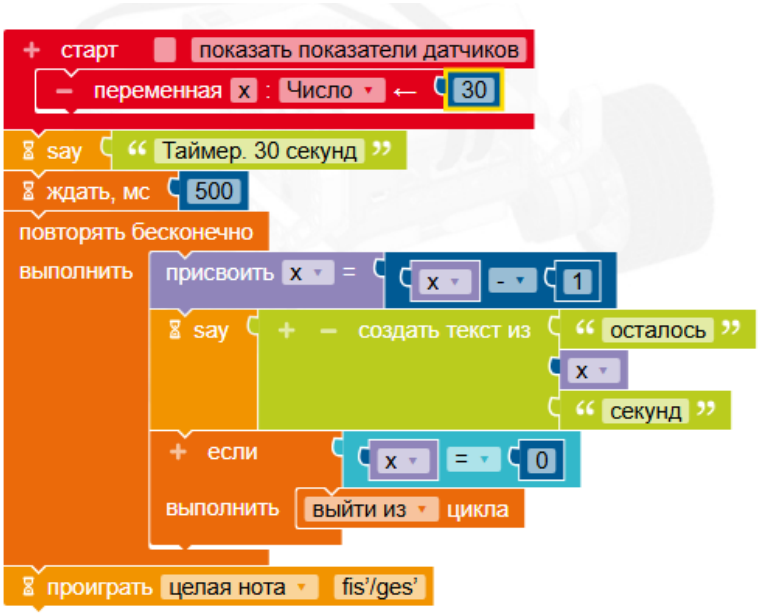
Рисунок 10. Пример «Вывод последовательности чисел в форме треугольника на экран блока».

Остановите программу после выполнения задания.

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используются переменные (x,y,z) и команда «Присвоить» из раздела «Переменные»
- Используется команда «Показать текст» в колонке 0, строке 0 из раздела «Действия»
- Используется команда «Ждать» из раздела «Контроль»
- Используется команда-оператор «Если – выполнить» из раздела «Контроль»
- Используется команда-оператор «Повторять бесконечно» из раздела «Контроль»
- Используется команда-оператор «Присваивание» из раздела «Логика» (блок «Равно»)
- Программа запускается и текст выводится на экран блока

Задание 6. Составьте программу по примеру, представленному на рисунке 11 «Пример программы «Счетчик чисел». Используйте разделы «Действие», «Текст» и «Контроль» и «Переменные». Для добавления новых переменных нажмите на кнопку «+» рядом с надписью «Старт». Сделайте выводы.

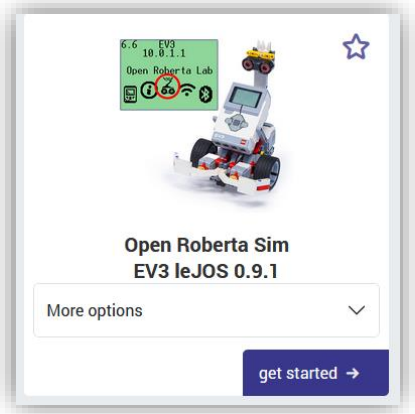


Дескрипторы:

- Используются переменные (x) и команда «Присвоить» из раздела «Переменные»
- Используется команда «Ждать» из раздела «Контроль»
- Используется команда-оператор «Если – выполнить» из раздела «Контроль»
- Используется команда-оператор «Повторять бесконечно» из раздела «Контроль»
- Используется команда-оператор «Присваивание» из раздела «Логика» (блок «Равно»)
- Программа запускается и текст воспроизводится

Тема. Элементарная тригонометрия.

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Задание 1. Замените фон нажав несколько раз кнопку (показана стрелкой), в соответствии с изображением «Координатная плоскость» на рисунке 1. Включите режим демонстрации траектории движения робота (показан стрелкой) и используйте кнопки (стрелка – влево, стрелка - вправо) на клавиатуре, а так же компьютерную мышь для изменения положения робота как показано на рисунке ниже.

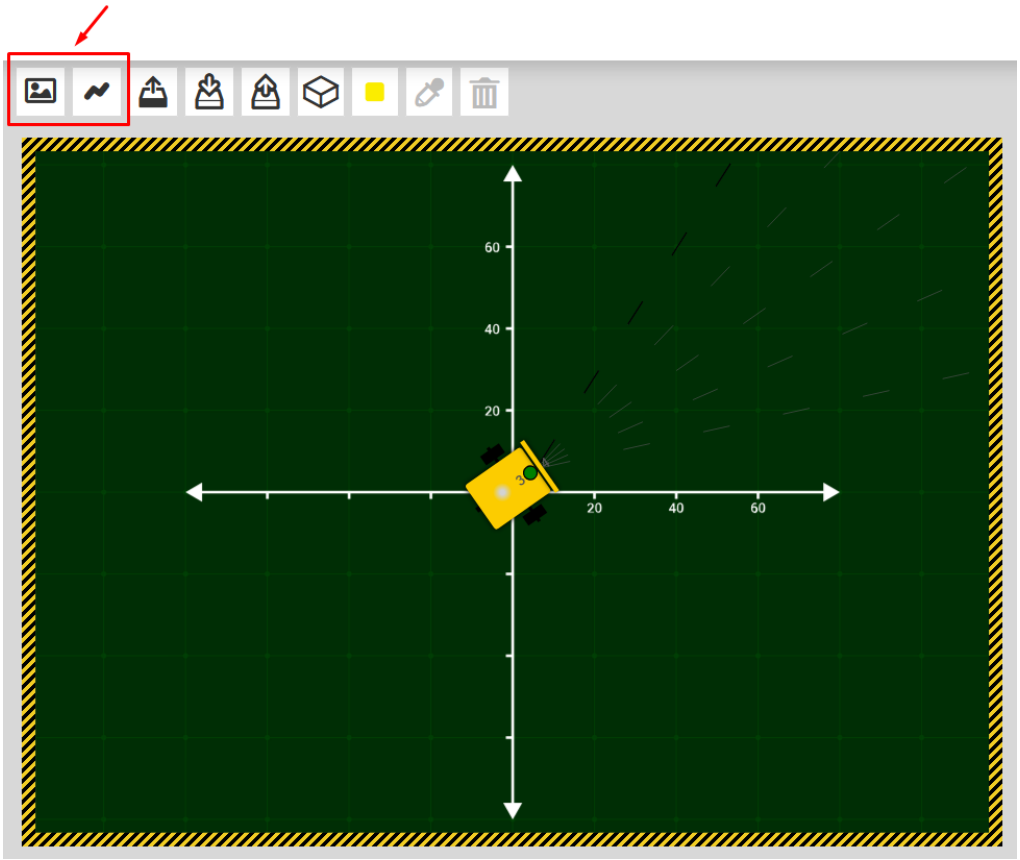


Рисунок 1. «Координатная плоскость»

Задание 2. Составьте программу по примеру, представленному на рисунке 2 «Пример программы «Вычисление значения синуса в градусах».

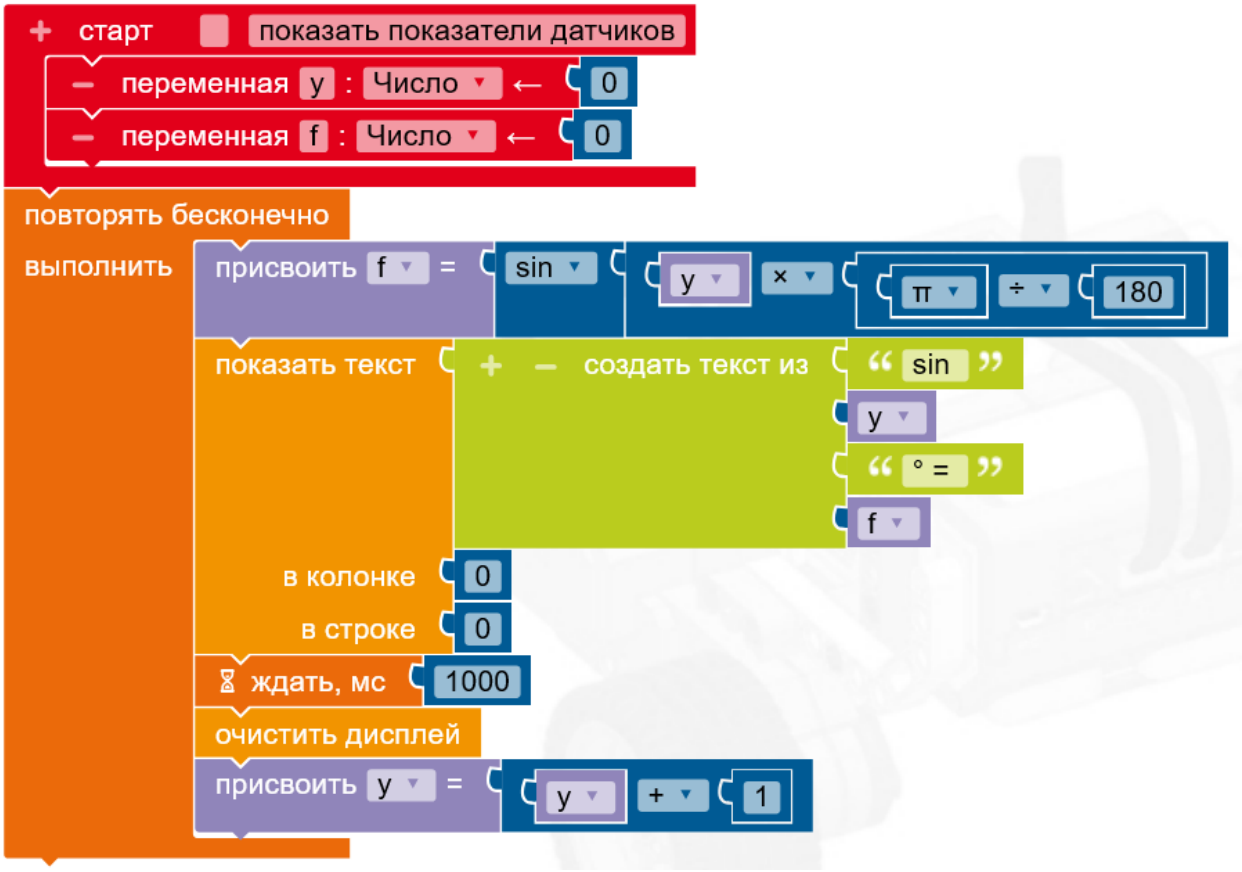


Рисунок 2. Пример программы «Синус»

Нажмите выделенную прямоугольником кнопку, представленную на изображении ниже для отображения вида от лица робота, в четком соответствии с результатом, показанным ниже на рисунке 3 «Пример вывода значений синуса (sin)».

Для проверки нажмите на кнопке «Старт».

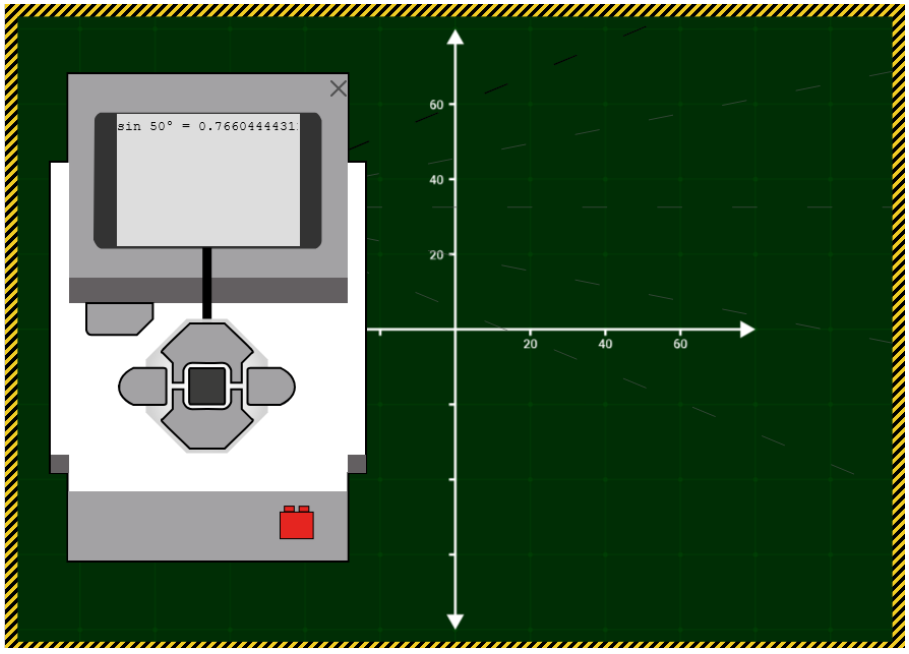
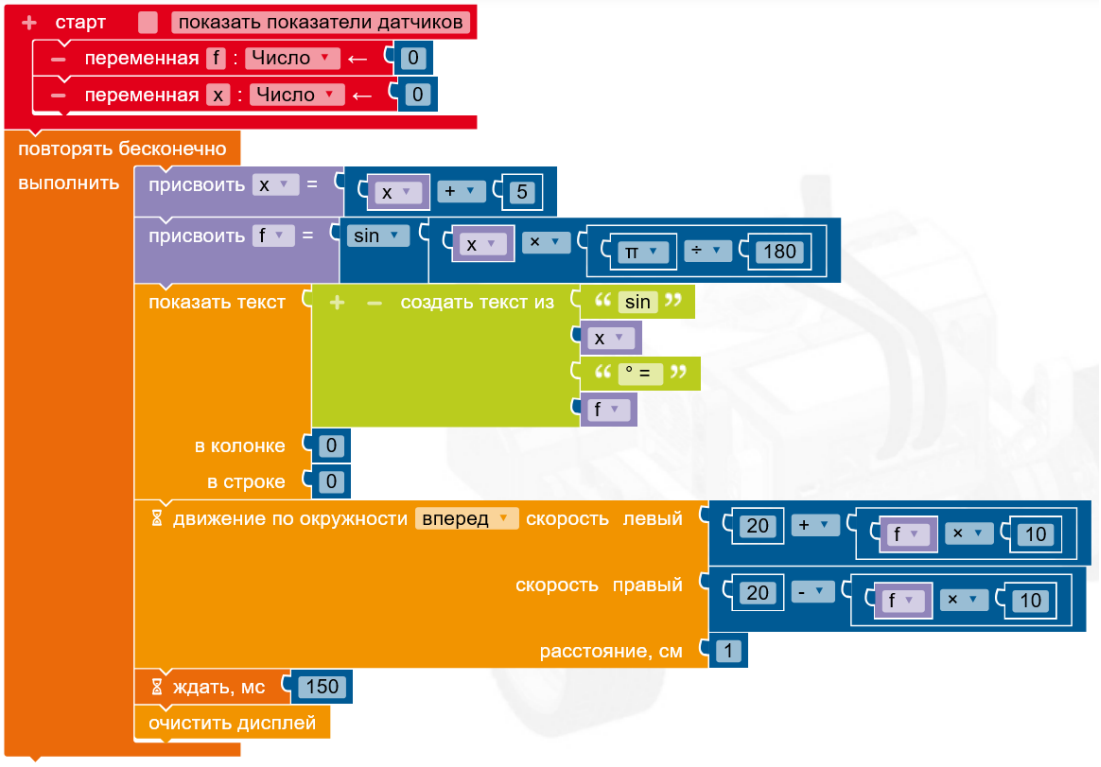


Рисунок 3. Пример вывода значений синуса (sin).

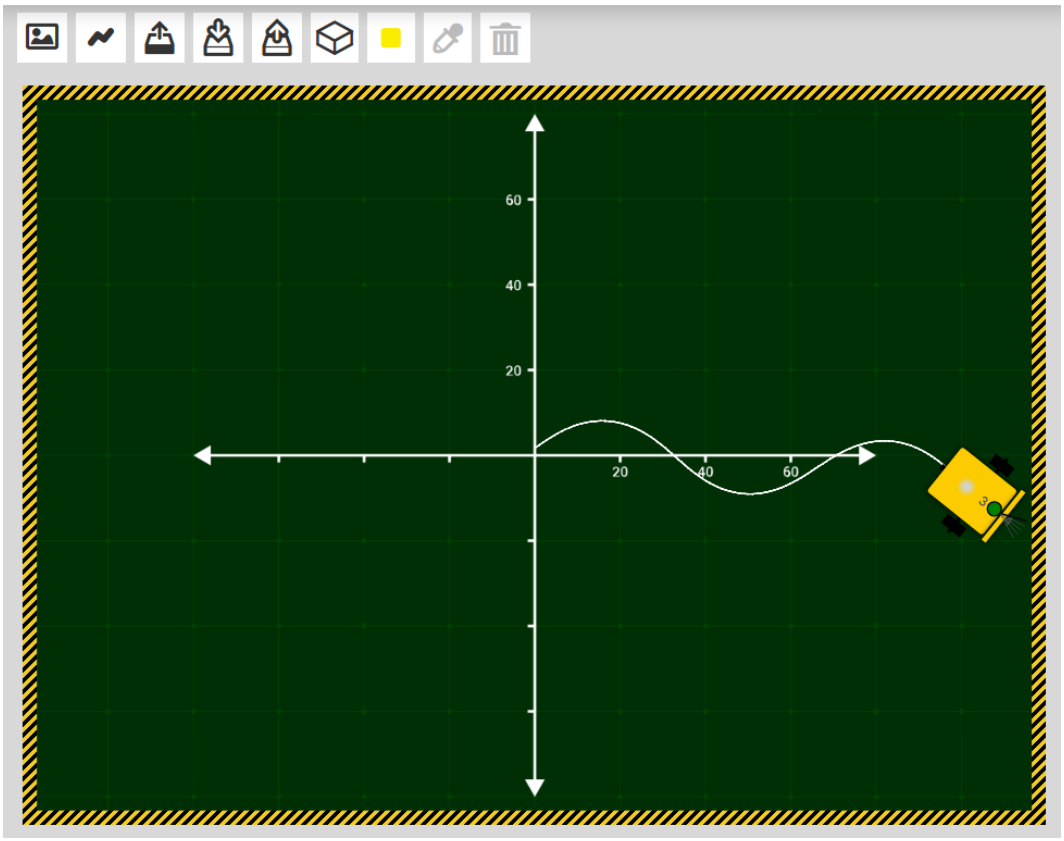
Остановите программу после выполнения задания.

Задание 3. Измените программу в полном соответствии с рисунком 4 «Движение робота по синусоеде».



Программа моделирует движение робота по траектории синусоиды. В бесконечном цикле значение переменной **x** постепенно увеличивается, затем вычисляется значение **f = sin(x)** (с переводом градусов в радианы). Эти значения используются для управления скоростью левого и правого моторов: за счёт разницы скоростей робот движется по волнообразной (синусоидальной) траектории. Дополнительно на экран выводится текущее значение функции. После небольшой задержки экран очищается, и процесс повторяется.

Для проверки нажмите на кнопке «Старт».



Остановите программу после выполнения задания.

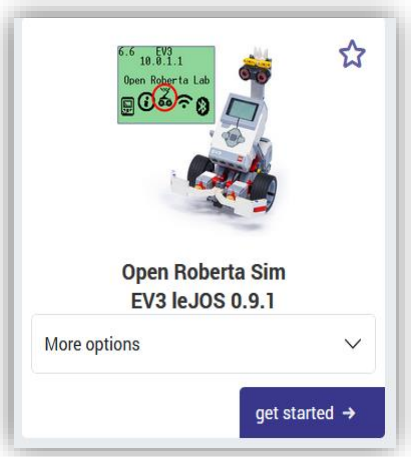
Дескрипторы:

• Создаются переменные **x** и **f** для хранения аргумента и значения функции.
• Устанавливаются начальные значения переменных (x = 0, f = 0).
• Используется бесконечный цикл «повторять бесконечно» для непрерывной работы программы.
• Используется действие «Присвоить переменной» для увеличения значения **x** на каждом

шаге.
• Используется математическая функция **sin** для вычисления значения **f**.
• Выполняется преобразование градусов в радианы (умножение на π / 180).
• Используется действие «Показать текст» для вывода значений **x** и **f** на экран.
• Используется блок «создать текст из» для форматированного вывода (sin(x) = f).
• Используется управление моторами: задаётся базовая скорость для движения.
• Используется выражение с переменной **f** для изменения скорости левого мотора.
• Используется выражение с переменной **f** для изменения скорости правого мотора.
• Создаётся разница скоростей моторов, обеспечивающая движение по кривой.
• Используется блок «движение по окружности» для плавного изменения траектории.
• Используется задержка (150 мс) для стабилизации движения и визуализации.
• Используется команда «очистить дисплей» для обновления данных на экране.
• Программа запускается: робот движется по траектории, повторяющей график синусоиды, отображая значения функции в реальном времени.

Тема. Базовые логические операции.

Задание 1. Откройте сайт https://lab.open-roberta.org/ и выберите пункт как на рисунке ниже. Для настройки сцены выполните все следующие действия в строгой последовательности.



5) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для изменения рисунка игрового поля («Среда») в соответствии с результатом показанным на рисунке 1 «Игровое поле»

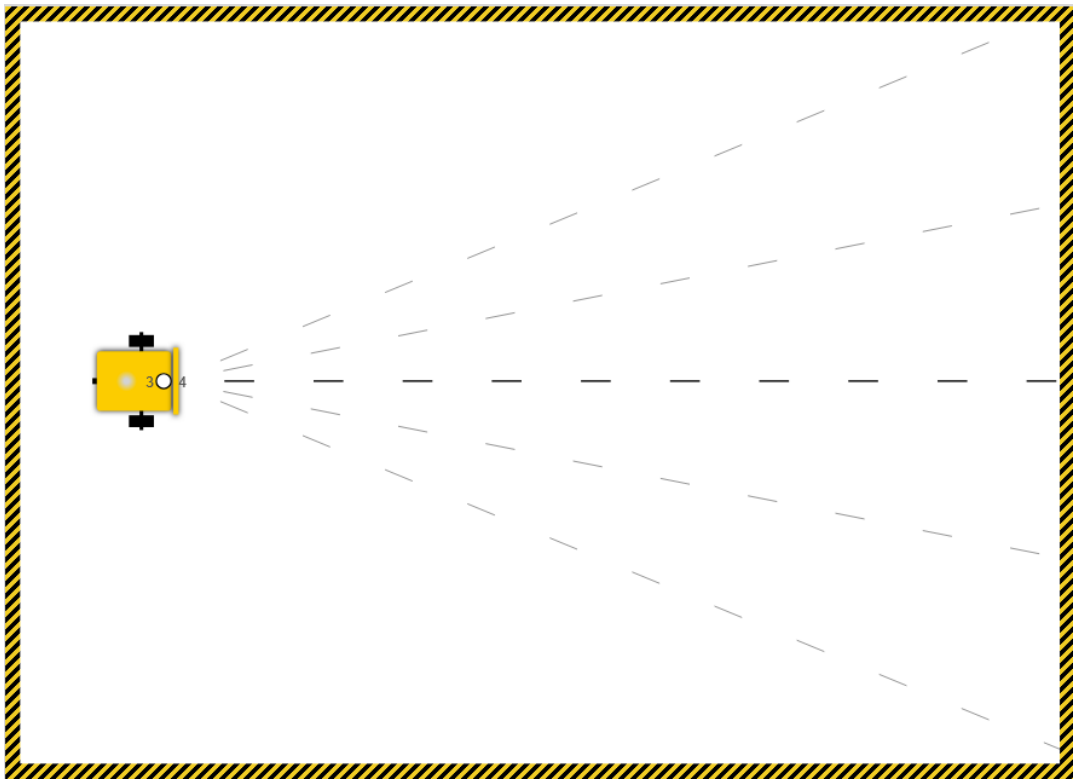


Рисунок 1. Игровое поле

Задание 2. Базовая логическая операция «НЕ» («Отрицание»).

a) Составьте программу по примеру, представленному на рисунке 2 «Пример программы «Базовая логическая операция «Отрицание». Используйте разделы «Действие», «Контроль», «Логика» и «Переменные».

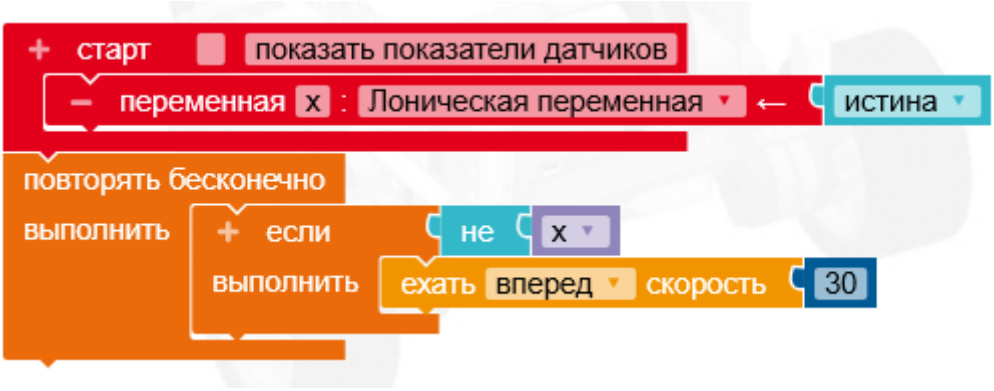


Рисунок 2. Пример программы «Базовая логическая операция «Отрицание»

Для проверки нажмите на кнопке «Старт». Проверьте перемещается ли ваш робот?

b) При каком начальном значении переменной «x» если перед ними стоит слово «не» будет перемещаться робот? Заполните письменно таблицу 1 следующими вариантами ответов: «Робот будет перемещаться», «Робот не будет перемещаться». В одной строке должен быть только один вариант ответа.

Таблица 1. Таблица истинности для базовой логической операции «Отрицание»

Переменная x	Результат
Ложь	Робот будет перемещаться
Истина	Робот не будет перемещаться

Остановите программу после выполнения задания.

Задание 3. Базовая логическая операция «И» («Конъюнкция»).

a) Составьте программу по примеру, представленному на рисунке 3 «Пример программы «Базовая логическая операция «Конъюнкция». Используйте разделы «Действие», «Контроль», «Логика» и «Переменные».

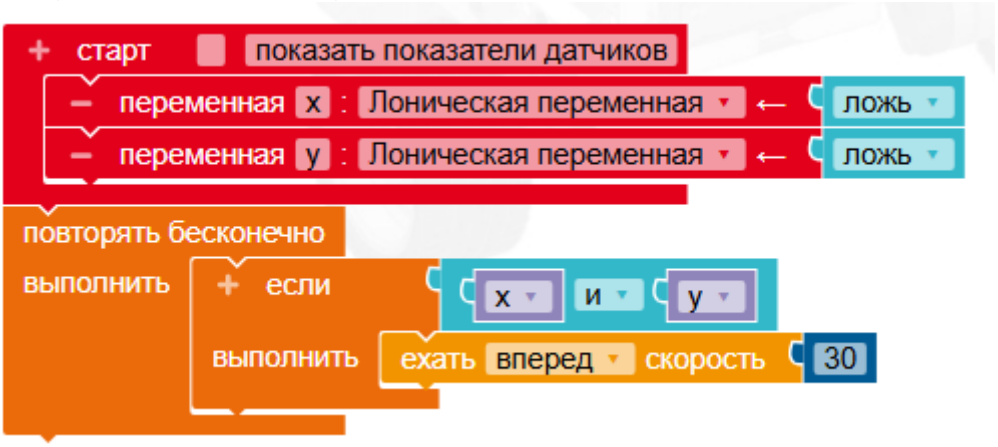


Рисунок 3. Пример программы «Базовая логическая операция «Конъюнкция»

Для проверки нажмите на кнопке «Старт».

b) При каких начальных значениях переменных «x» и «y» если между ними стоит слово «и» будет перемещаться робот? Заполните письменно таблицу 2 следующими вариантами ответов: «Робот будет перемещаться», «Робот не будет перемещаться». В одной строке должен быть только один вариант ответа.

Таблица 2. Таблица истинности для базовой логической операции «Конъюнкция»

Переменная x	Переменная y	Результат
Ложь	Ложь	Робот не будет перемещаться
Ложь	Истина	Робот не будет перемещаться
Истина	Ложь	Робот не будет перемещаться
Истина	Истина	Робот будет перемещаться

Остановите программу после выполнения задания.

Задание 3. Базовая логическая операция «ИЛИ» («Дизъюнкция»).

a) Составьте программу по примеру, представленному на рисунке 4 «Пример программы «Базовая логическая операция «Дизъюнкция». Используйте разделы «Действие», «Контроль», «Логика» и «Переменные».

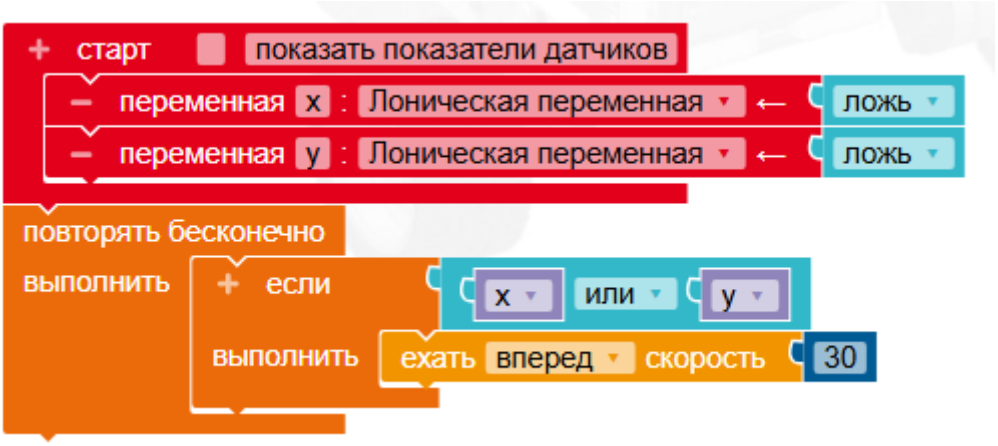


Рисунок 4. Пример программы «Базовая логическая операция «Дизъюнкция»

Для проверки нажмите на кнопке «Старт».

b) При каких начальных значениях переменных «x» и «y» если между ними стоит слово «или» будет перемещаться робот? Заполните письменно таблицу 3 следующими вариантами ответов: «Робот будет перемещаться», «Робот не будет перемещаться». В одной строке должен быть только один вариант ответа.

Таблица 3. Таблица истинности для базовой логической операции «Дизъюнкция»

Переменная x	Переменная y	Результат
Ложь	Ложь	Робот не будет перемещаться
Ложь	Истина	Робот будет перемещаться
Истина	Ложь	Робот будет перемещаться
Истина	Истина	Робот будет перемещаться

Остановите программу после выполнения задания.

Задание 4. Логическая операция «XOR» («Исключающее или»).

Запомните, что порядок выполнения логических операций, если нет скобок, следующий: 1) не (отрицание); 2) и (коньюнкция); 3) или (дизъюнкция).

a) Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Логическая операция «Исключающее или». Используйте разделы «Действие», «Контроль», «Логика» и «Переменные».

Рисунок 5. Пример программы «Логическая операция «Исключающее или»

Для проверки нажмите на кнопке «Старт».

b) При каких начальных значениях переменных «x» и «y» если выражение выглядит как «не x и y или x и не y» будет перемещаться робот? Заполните письменно таблицу 4 следующими вариантами ответов: «Робот будет перемещаться», «Робот не будет перемещаться». В одной строке должен быть только один вариант ответа.

Таблица 4. Таблица истинности для логической операции «Исключающее или»

Переменная x	Переменная y	Результат
Ложь	Ложь	Робот не будет перемещаться
Ложь	Истина	Робот будет перемещаться
Истина	Ложь	Робот будет перемещаться
Истина	Истина	Робот не будет перемещаться

Остановите программу после выполнения задания.

Сравните таблицы истинности для логической операций «И» (конъюнкция), «ИЛИ» (дизъюнкция), «XOR» (исключающее или). Сделайте выводы.

Задание 5. Составьте программу по примеру, представленному на рисунке 6 «Пример программы «Логические операции с датчиком». Используйте разделы «Действие», «Переменные», «Логика»

и «Контроль». Настройте ваше игровое поле («Среда») в соответствии с рисунком 1 «Игровое поле» темы «Базовые логические операции».

Рисунок 6. Пример программы «Логические операции с датчиком »

Для проверки нажмите на кнопке «Старт».

Ваша программа должна проверить значение логической переменной «x» и перемещать робота до тех пор пока значение логической переменной «x» не станет равно «Ложь». В свою очередь значение «Ложь» для логической переменной «x» наступит в тот момент когда расстояние до препятствия станет меньше 20.

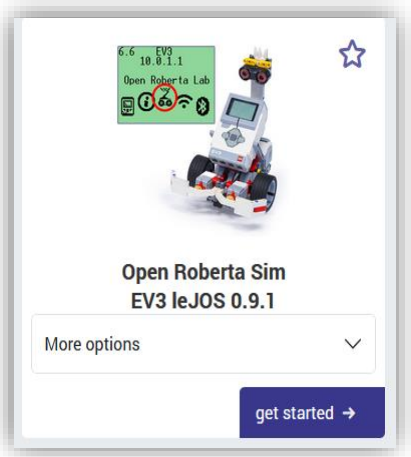
Остановите программу после выполнения задания.

Дескрипторы:

- Настраивается сцена, выводится вид от лица робота
- Используется команда «Повторить бесконечно» из раздела «Контроль»
- Используется команда «Если - иначе» из раздела «Контроль»
- Используются команды «НЕ», «И», «ИЛИ», «Равно» из раздела «Логика»
- Используются переменные и команда «Присвоить» из раздела «Переменные»
- Используется команда «Ехать вперед» и «Остановка» из раздела «Действие»
- Используется команда «Ультразвуковой датчик» из раздела «Датчики»
- Программа запускается: робот перемещается если результат «ИСТИНА»

Тема. Датчик «Гироскоп». Прямолинейное движение.

Задание 1. Откройте сайт https://lab.open-roberta.org/ и выберите пункт как на рисунке ниже. Для настройки сцены выполните все следующие действия в строгой последовательности.



6) Нажмите выделенную прямоугольником кнопку представленную на изображении ниже для изменения рисунка игрового поля («Среда») в соответствии с результатом показанным на рисунке 1 «Игровое поле»

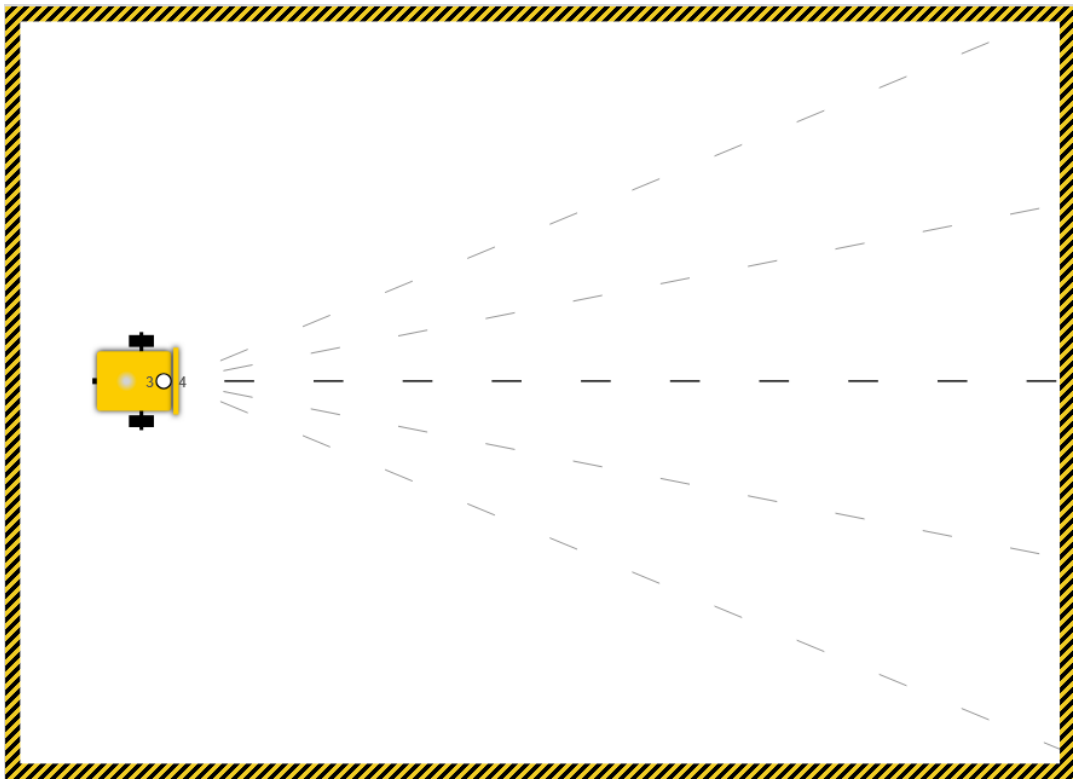


Рисунок 1. Игровое поле

Задание 2. Представьте, что вы готовите робота к важной миссии по изучению ровной поверхности. Ваша задача — написать программу, которая позволит роботу правильно подготовиться к движению и начать стабильное перемещение вперед. Чтобы робот не сбился с курса в будущем, перед стартом необходимо предусмотреть технические паузы для стабилизации систем и обязательно обнулить показания гироскопического датчика, который отвечает за точность направления. Робот должен использовать заранее заданную скорость, хранящуюся в специальной ячейке памяти, и продолжать движение бесконечно долго, пока его не остановят принудительно. Исправь ошибку с противоречивыми поворотами из предыдущей версии и добейся того, чтобы моторы работали синхронно. Рассмотри изображение ниже и составь программу по образцу представленному ниже.

Для проверки нажмите на кнопке «Старт».

Остановите программу после выполнения задания.

Объяснение. Работа программы начинается с самого верхнего блока «старт», который запускает выполнение всех команд по цепочке сверху вниз. Сначала мы создаем переменную под названием «speed» и записываем в неё число 30. Это можно сравнить с коробкой, в которую мы положили значение скорости, чтобы потом просто называть имя коробки, а не вписывать число каждый раз заново. Затем робот выполняет команду «ждать 1000 мс», что равно одной секунде. Эта пауза очень важна: она дает роботу время успокоиться после включения, чтобы его датчики не зафиксировали случайную тряску. Следующим шагом мы используем блок «сбросить гироскоп Port 2», который устанавливает текущее положение робота как точку отсчета — теперь робот считает, что он смотрит идеально прямо по направлению на 0 градусов. После этого мы снова добавляем секундную паузу, чтобы датчик успел зафиксировать свое новое состояние и робот был полностью готов к физическому действию. Наконец, мы переходим к оранжевому блоку «повторять бесконечно», внутри которого находится команда «ехать вперед скорость speed». В отличие от первого варианта, где робот получал команды повернуть в разные стороны

одновременно и стоял на месте, здесь он получает только одно четкое указание. Поскольку эта команда находится внутри бесконечного цикла, компьютер робота постоянно проверяет и подтверждает: «Да, я все еще должен ехать вперед со скоростью 30». В результате робот плавно и уверенно начинает свое движение по прямой линии.

Задание 2. Запрограммируйте мобильного робота в соответствии с программой, представленной на рисунке 2 «Пример программы «Движение прямо с гироскопом» для выполнения циклического движения по прямой линии. Робот должен проезжать ровно 50 сантиметров, используя показания встроенных датчиков оборотов колес для контроля дистанции. Чтобы робот не сбивался с курса и ехал максимально ровно, необходимо реализовать систему автоматической корректировки направления с помощью гироскопического датчика. В случае малейшего отклонения программа должна самостоятельно вычислять разницу между заданным курсом и текущим положением, плавно подстраивая скорость левого и правого моторов так, чтобы робот возвращался на идеальную прямую траекторию. Скорость движения при этом не должна быть слишком низкой, чтобы робот не застрял, и не слишком высокой, чтобы сохранялась точность измерений.

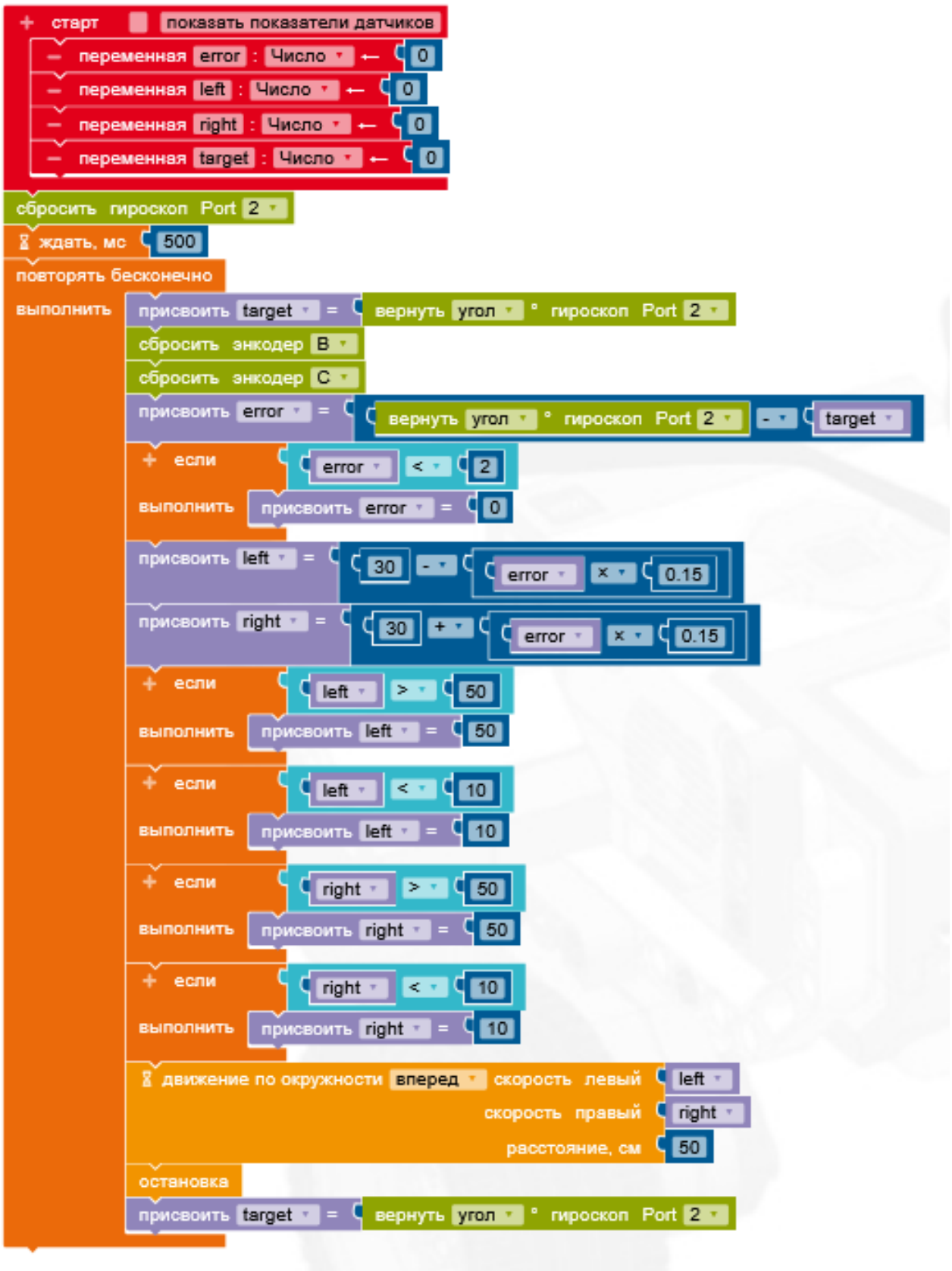


Рисунок 2. Пример программы «Движение прямо с гироскопом»

Для проверки нажмите на кнопке «Старт».

Остановите программу после выполнения задания.

Объяснение. Программа начинается с подготовки датчиков и создания переменных, которые будут хранить важные числа: величину ошибки, значения скорости для каждого мотора и целевой угол направления. Первым делом мы сбрасываем гироскоп в нулевое положение и даем роботу полсекунды на «покой», чтобы датчик откалибровался и не выдавал ошибок. Весь основной алгоритм помещен в бесконечный цикл, который раз за разом повторяет одни и те же действия. В начале каждого круга робот фиксирует свое текущее направление в переменную target и обнуляет счетчики в моторах, чтобы начать отсчет пути заново. Далее программа вычисляет «ошибку» — разницу между тем, куда робот должен ехать, и тем, куда он смотрит на самом деле. Чтобы робот не дергался из-за естественного дрожания датчика, добавлено условие: если отклонение меньше 2 градусов, оно считается нулевым. Основная магия происходит при расчете скоростей left и right: мы берем базовую скорость 30 и либо вычитаем из неё, либо прибавляем к ней ошибку, умноженную на коэффициент 0.15. Это заставляет один мотор вращаться чуть быстрее другого, возвращая робота на курс. Блоки проверки «если» следят за тем, чтобы скорость моторов всегда оставалась в пределах от 10 до 50 единиц — это гарантирует, что моторы не остановятся совсем и не разгонятся до опасных значений. Наконец, блок движения по окружности использует эти рассчитанные скорости, чтобы проехать ровно 50 сантиметров, после чего робот на мгновение замирает, обновляет целевое направление и начинает новый цикл движения по прямой.

Ниже представлена таблица 1 «Работа системы выравнивания», которая наглядно показывает, как «мозг» робота меняет скорость колес в зависимости от того, куда он отклонился. Это поможет понять, почему в программе используются именно такие формулы.

Таблица 1. Работа системы выравнивания.

Положение робота	Угол (гироскоп)	Ошибка (error)	Скорость левого (30 - error * 0.15)	Скорость правого (30 + error * 0.15)	Результат
Идеально прямо	0°	0	30	30	Робот едет ровно
Небольшой занос влево	-10°	-10	31.5	28.5	Правое колесо замедляется, левое ускоряется — робот возвращается вправо
Сильный занос влево	-40°	-40	36	24	Робот резко подруливает вправо
Небольшой занос вправо	10°	10	28.5	31.5	Левое колесо замедляется, правое ускоряется — робот возвращается влево
Сильный занос вправо	40°	40	24	36	Робот резко подруливает влево

Задание 3. Запрограммируйте мобильного робота в соответствии с программой, представленнойй на рисунке 3 «Пример программы «Движение по квадрату с гироскопом» для выполнения бесконечного цикла движений в замкнутом пространстве: робот должен проехать по прямой линии ровно 50 сантиметров, используя показания гироскопического датчика для коррекции курса, после чего совершить точный правый поворот на 90 градусов и повторить эти действия сначала. В процессе движения по прямой робот должен автоматически выравниваться, если его отклонит от курса, а при выполнении поворота плавно снижать скорость по мере приближения к цели, чтобы избежать проскакивания нужного угла из-за инерции.

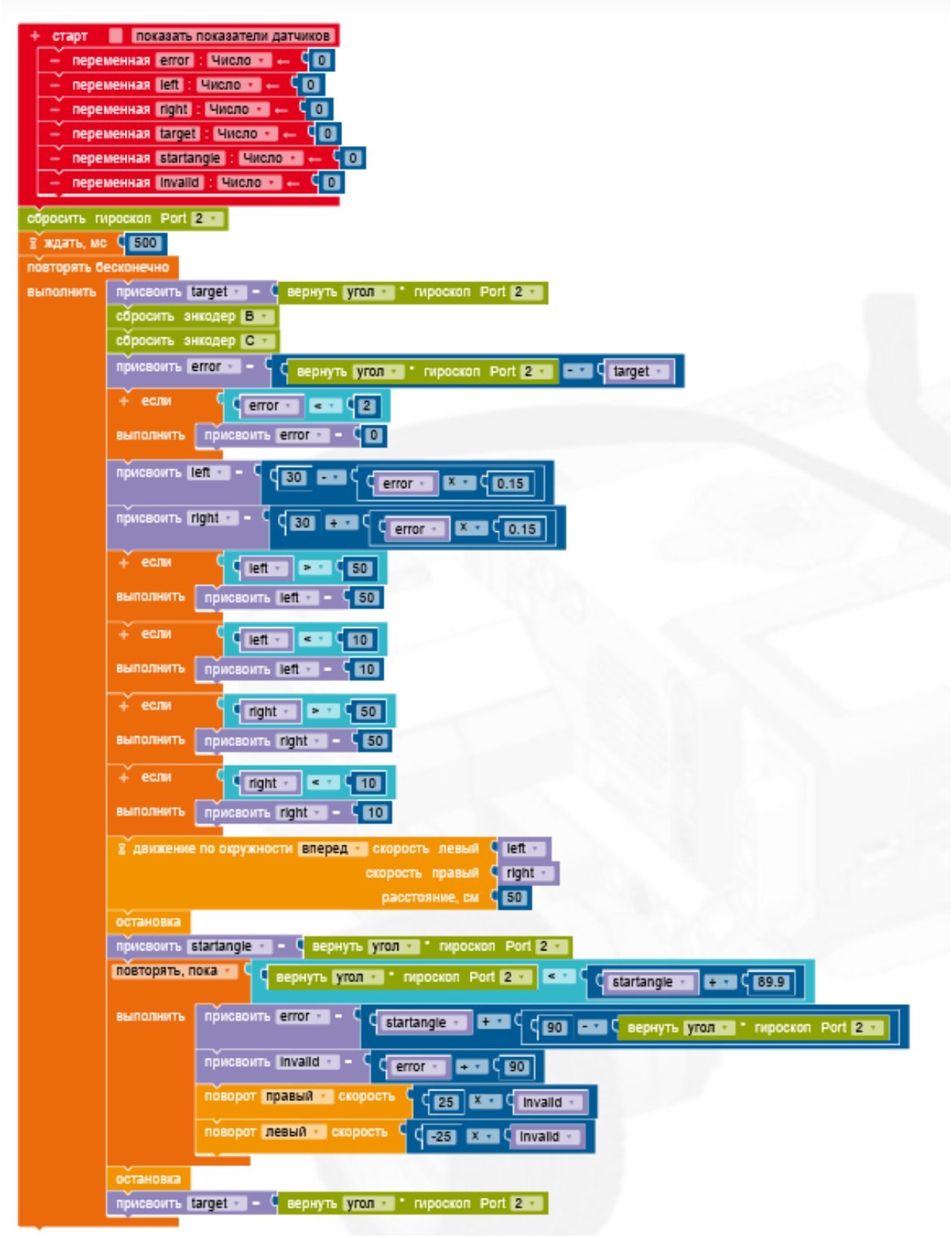


Рисунок 3. «Пример программы «Движение по квадрату с гироскопом»

Для проверки нажмите на кнопке «Старт».

Остановите программу после выполнения задания.

Объяснение. В самом начале программы мы подготавливаем «память» робота, создавая переменные для хранения расчетов скорости и углов, а также обнуляем гироскоп и даем ему полсекунды на стабилизацию, чтобы датчик не выдавал ложных значений. Основная часть кода заключена в бесконечный цикл, который начинается с того, что робот запоминает текущее направление в переменную target и сбрасывает счетчики вращения колес (энкодеры). Для движения по прямой используется блок коррекции: программа вычисляет разницу между желаемым направлением и текущим углом гироскопа, записывая её в error. Если это отклонение меньше 2 градусов, робот считает его незначительным и приравнивает к нулю, чтобы не дергаться по пустякам. Затем математическая формула рассчитывает скорости для левого и правого моторов: если робот уходит в сторону, один мотор замедляется, а другой ускоряется на величину, пропорциональную ошибке. Дополнительные блоки «если» следят, чтобы скорость не выходила за рамки безопасного диапазона от 10 до 50 единиц. После того как робот проедет 50 сантиметров, он переходит к фазе поворота: запоминает угол начала маневра в переменную startangle и запускает цикл, который работает, пока текущий угол не станет равен углу старта плюс 89.9 градуса. Внутри этого цикла вычисляется коэффициент invalid, который плавно уменьшается от 1 до 0 по мере того, как робот докручивается до нужной отметки. Это позволяет моторам вращаться быстро в начале поворота и почти замирать в конце для максимальной точности. Как только поворот завершен, робот полностью останавливается, обновляет значение целевого направления и снова отправляется в путь по прямой.

Дескрипторы:

- Создаются и инициализируются переменные target, error, left, right и startangle для управления логикой движения.
- Используется команда «Повторять бесконечно» из раздела «Контроль» для организации непрерывного цикла корректировки курса.
- Используется команда «Если» из раздела «Контроль» для проверки выхода значений скорости за установленные границы и фильтрации мелких ошибок датчика.
- Используются команды сравнения «Меньше», «Больше» из раздела «Логика» для анализа отклонения робота от целевого угла.
- Используются переменные и команда «Присвоить» из раздела «Переменные» для вычисления текущей ошибки и динамического изменения скорости моторов.
- Используются математические операции (вычитание, умножение, сложение) для расчета корректирующего воздействия на основе данных гироскопа.

- Используется команда «движение по окружности» из раздела «Действие» для раздельного управления скоростью левого и правого колеса.
- Используется команда «Гироскопический датчик» из раздела «Датчики» для получения текущего угла поворота робота в режиме реального времени.
- Используется команда «Энкодер» для обнуления пройденного пути перед началом нового цикла движения.
- Программа запускается: робот движется по прямой, автоматически исправляя отклонения, вычисляя разницу скоростей между моторами B и C на основе показаний гироскопа.

Тема. Случайные числа. Игра «Поле чудес».

Задание1. Подготовка сцены. Разработайте картинку для состоящую из четырёх секторов различного цвета. Откройте приложение Paint и используйте по порядку следующие инструменты: «Овал», «Линия», «Изменение цветов» и «Заливка» в Панели инструментов меню «Главная» в соответствии с инструкцией ниже.

1. Этап 1. Используйте «Овал» для рисования одного овала
2. Этап 2. Используйте «Линия» для рисования двух линий.
3. Этап 3. Используйте «Заливка» для заливки четырех секторов

Используйте диалоговое окно «Изменение цветов» и настройте цвета в соответствии с таблицей 1 «Настройка цвета», как показано на рисунке 1 «Пример поля «Поле чудес» ниже.

4. Этап 3. Сохраните получившееся изображения на «Рабочий стол» (Сохранить как → Рисунок в формате PNG или JPEG) с именем файла «Поле чудес».

Таблица 1. Настройка цвета

Секторы овала	Числовые значение цвета		
	Красный	Зеленый	Синий
Красный сектор	179	0	6
Желтый сектор	255	242	0
Синий сектор	0	87	166
Зелёный сектор	0	100	46

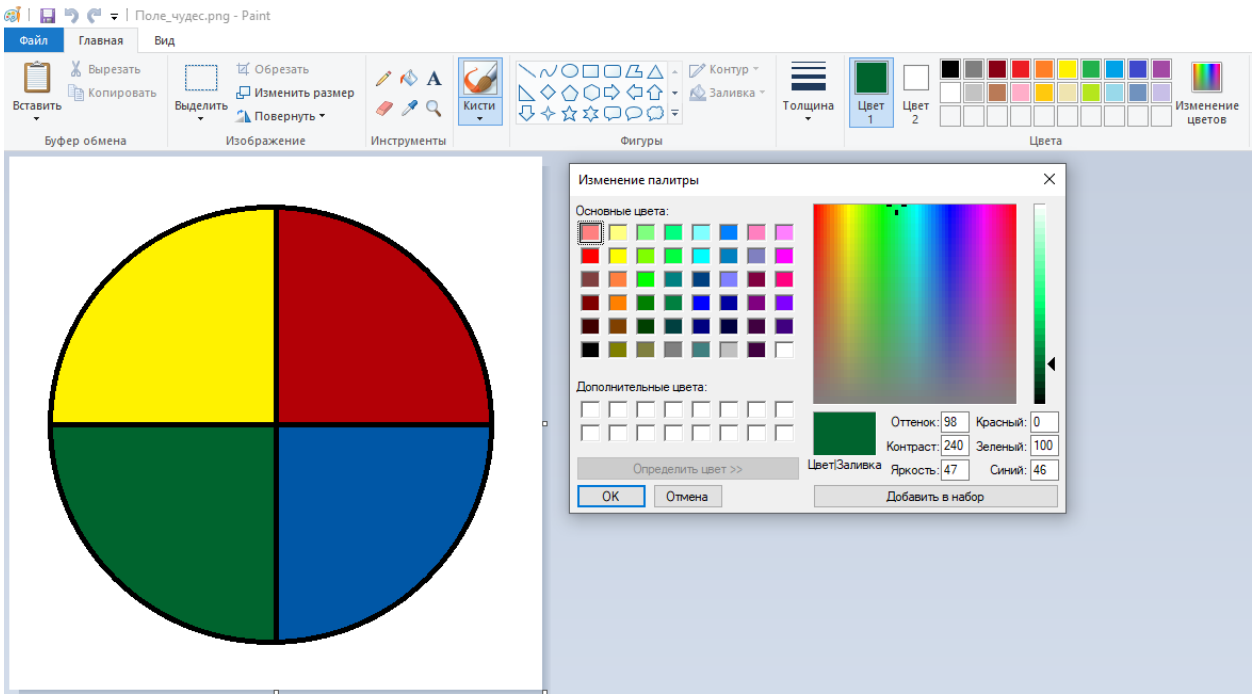
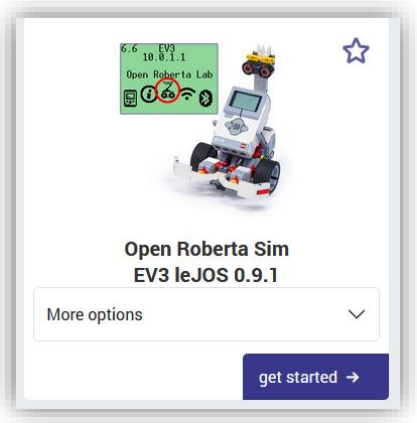
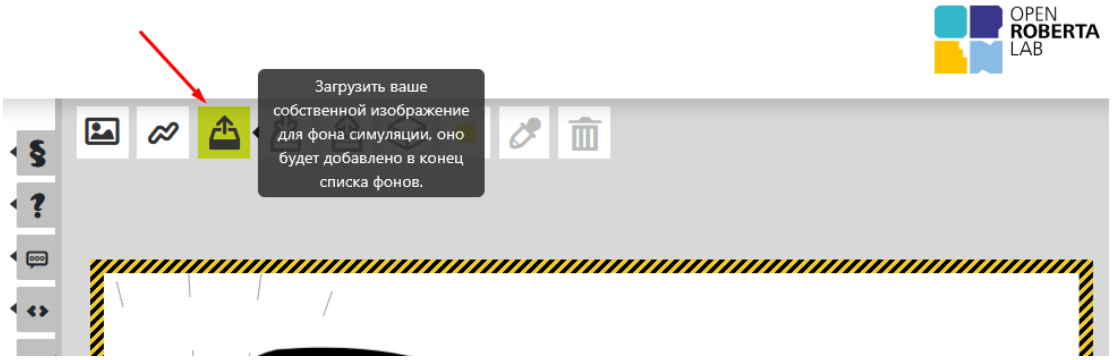


Рисунок 1. Пример поля «Поле чудес»

Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Задание 2. Откройте сайт https://lab.open-roberta.org/ и выберите пункт (показан стрелкой)



Настройте сцену как на Рисунке 2 «Поле чудес» ниже. Используйте кнопки «стрелка влево» или «стрелка вправо» на клавиатуре и компьютерную мышь, чтобы изменить позицию робота. Разместите робота как в примере, представленном на изображении ниже.

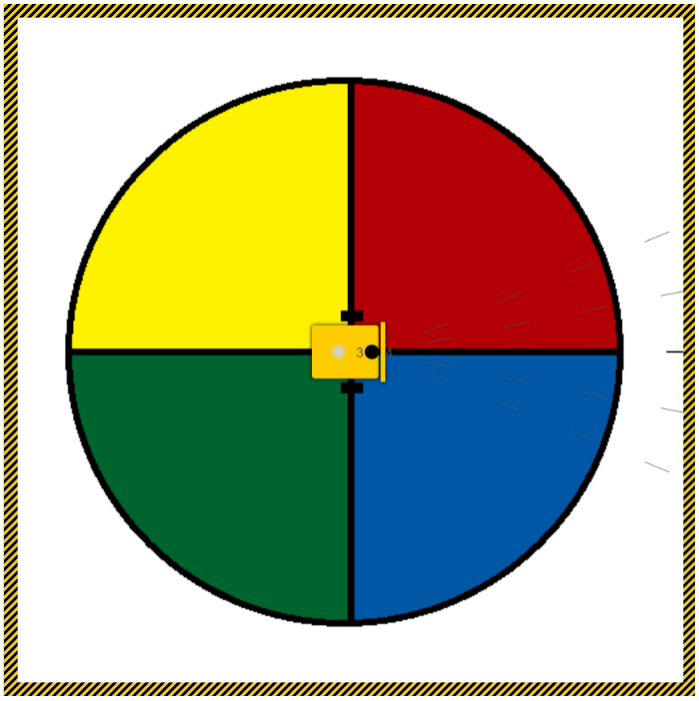


Рисунок 2. «Поле чудес»

Задание 3. Составьте программу по примеру, представленному на рисунке 3 «Пример программы «Поле чудес».

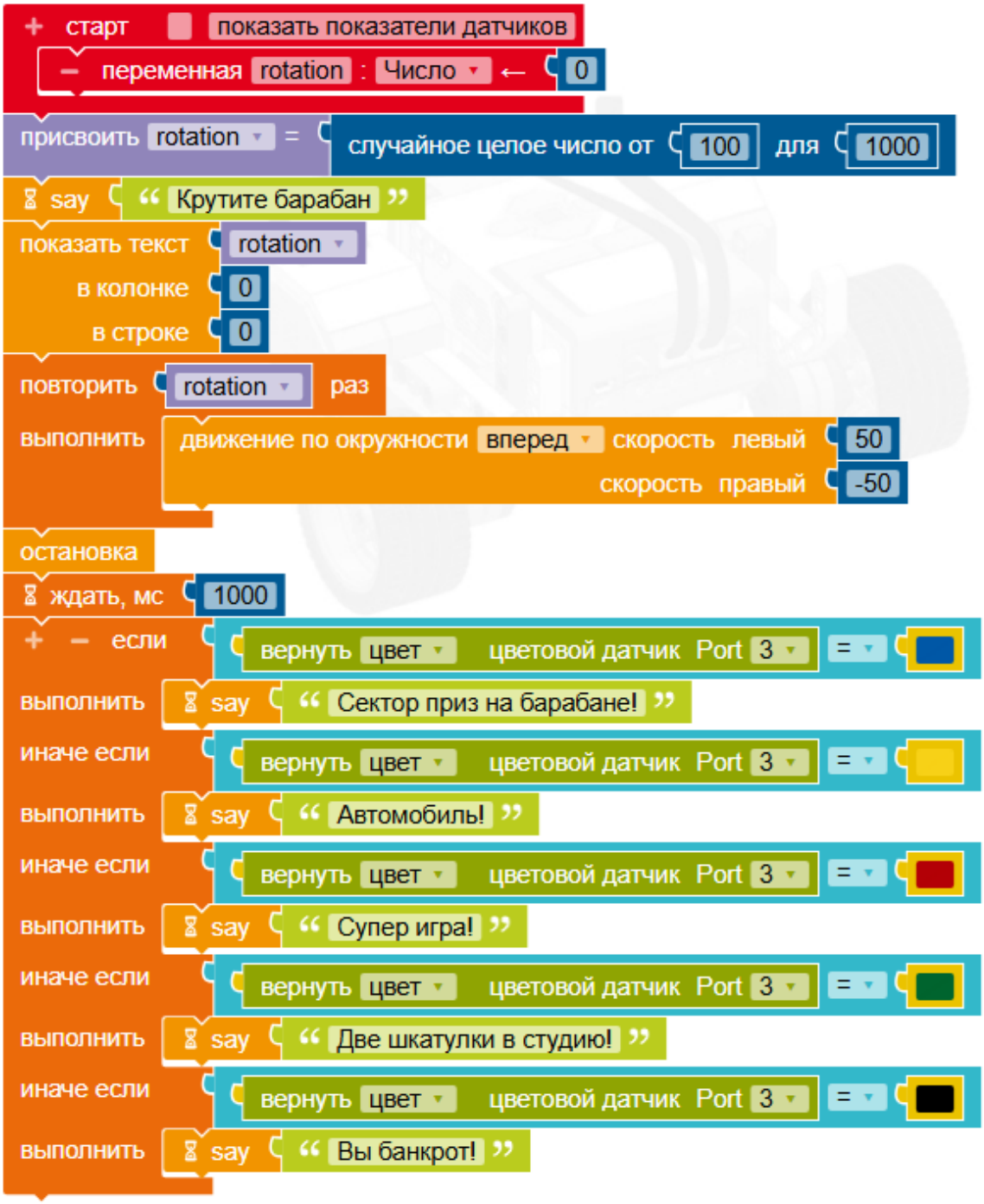


Рисунок 3. Пример программы «Поле чудес»

Нажмите выделенную прямоугольником кнопку, представленную на изображении ниже для отображения вида от лица робота, в четком соответствии с результатом, показанным ниже на рисунке 4 «Пример вывода генератора случайных чисел»ы.

Для проверки нажмите на кнопке «Старт».

Ваша программа имитирует игру **«Поле чудес»**. Сначала создаётся переменная rotation, в которую **генератор случайных чисел** записывает число от **100 до 1000**. Генератор случайных чисел — это команда, которая каждый раз выбирает **разное число случайным образом**, поэтому заранее нельзя узнать результат. Затем робот говорит «Крутите барабан», показывает это число на экране и **вращается на месте столько раз, сколько выпало**. После остановки цветовой датчик определяет цвет сектора барабана, на котором остановился робот, и выводит соответствующее сообщение — например, приз, автомобиль или «Вы банкрот!».

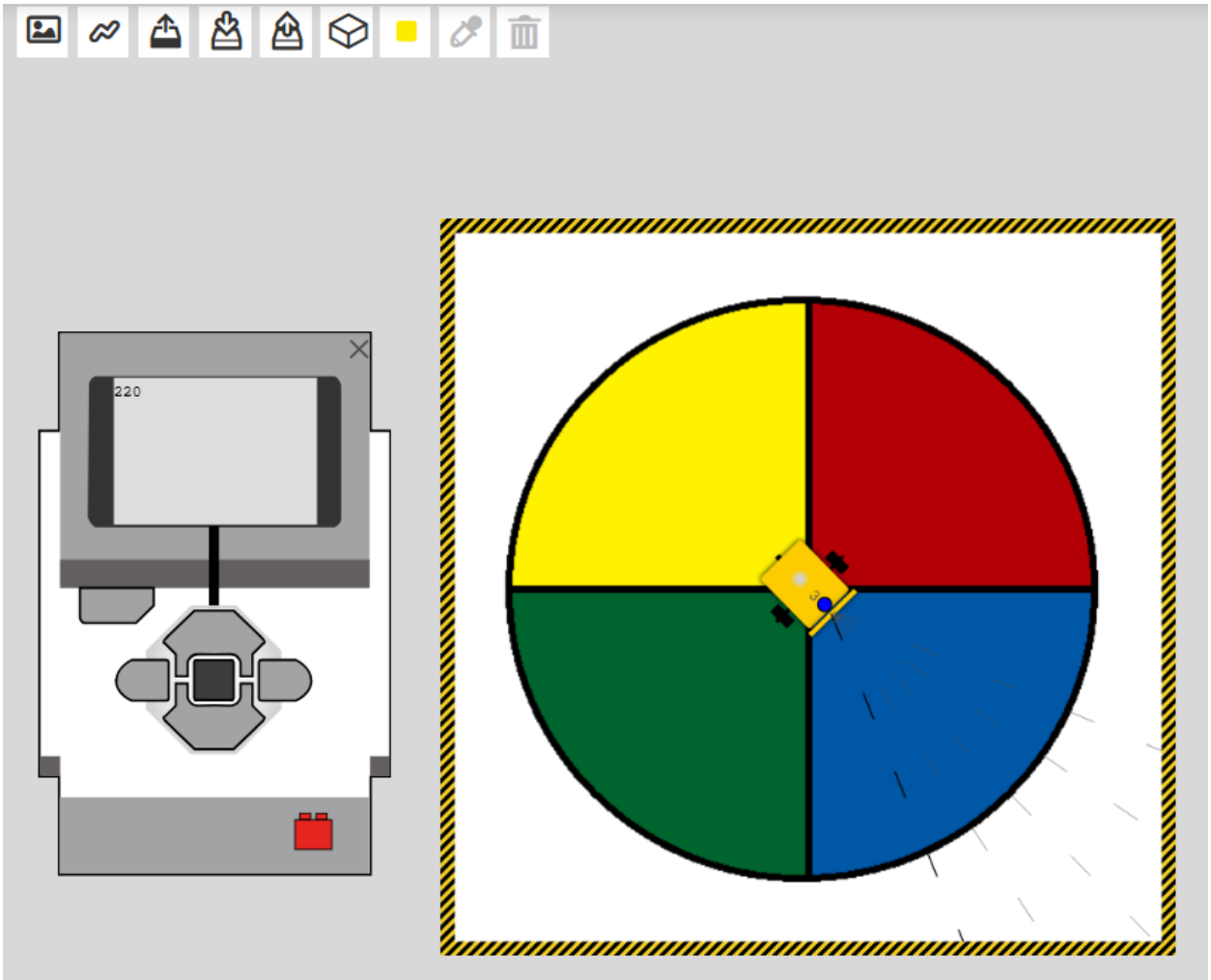


Рисунок 4. Пример вывода генератора случайных чисел.

Остановите программу после выполнения задания.

Дескрипторы:

• Создаётся переменная **rotation** и устанавливается начальное значение 0
• Используется действие **«Присвоить переменной»**
• Используется **генератор случайных чисел** (случайное число от 100 до 1000)
• Случайное число сохраняется в переменной **rotation**
• Используется действие **«Say»** для вывода сообщения «Крутите барабан»
• Используется действие **«Показать текст»** для отображения значения переменной **rotation**
• Используется контроль **«Повторить … раз»** с переменной **rotation**
• Используется действие **«Движение по окружности»** для вращения робота
• Используется действие **«Остановка»** после завершения движения
• Используется действие **«Ждать»** (1000 мс)
• Используется датчик **цвета (Port 3)**
• Используется контроль **«Если / иначе если»** для проверки цвета
• Для каждого цвета выводится сообщение с помощью действия **«Say»**
• Программа запускается: робот вращается случайное количество раз
• Программа запускается: определяется цвет сектора и озвучивается результат игры

Тема. Структура «Списки». Игра «Распознавание символов».

Задание1. Подготовка сцены. Откройте приложение MS Excel (Open Office Calc) и создайте новую книгу Excel. Измените ширину столбцов и строк в диапазоне A1:K11, в соответствии с примером показанным на рисунке 1 «Таблица Excel». Используйте диалоговое окно «Формат ячеек» (рисунок 2 «Диалоговое окно «Формат ячеек») для изменения цвета границы ячеек на серый.

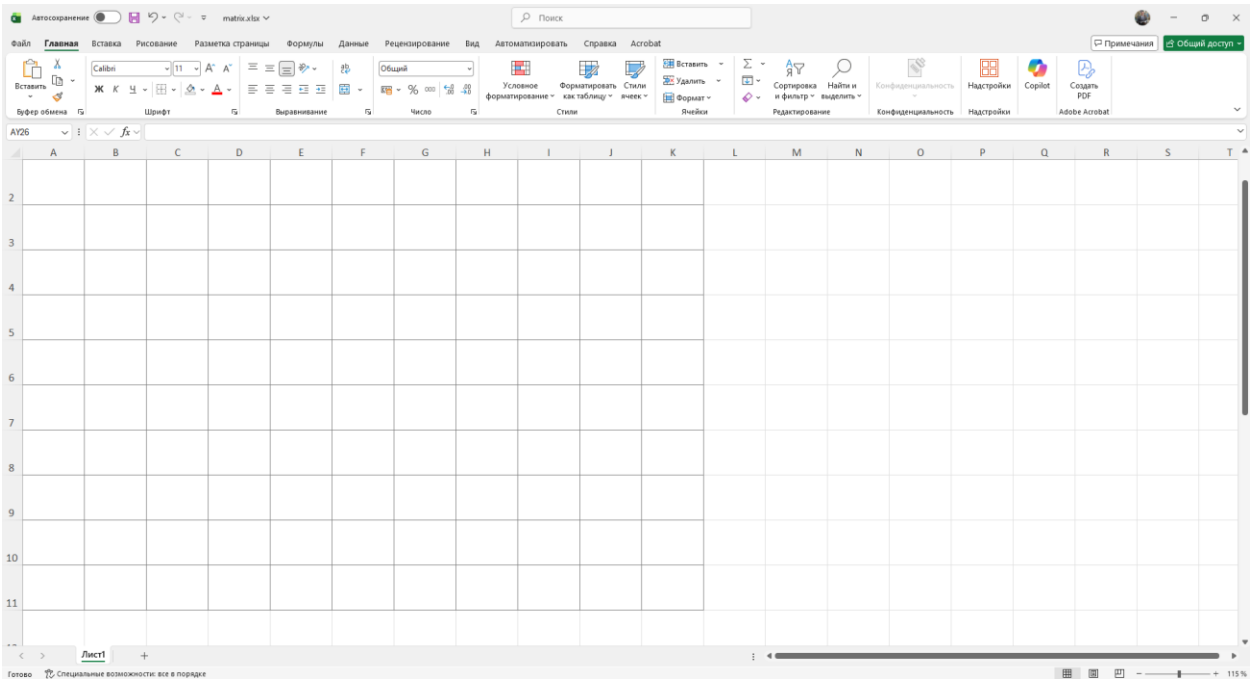


Рисунок 1. Таблица Excel.

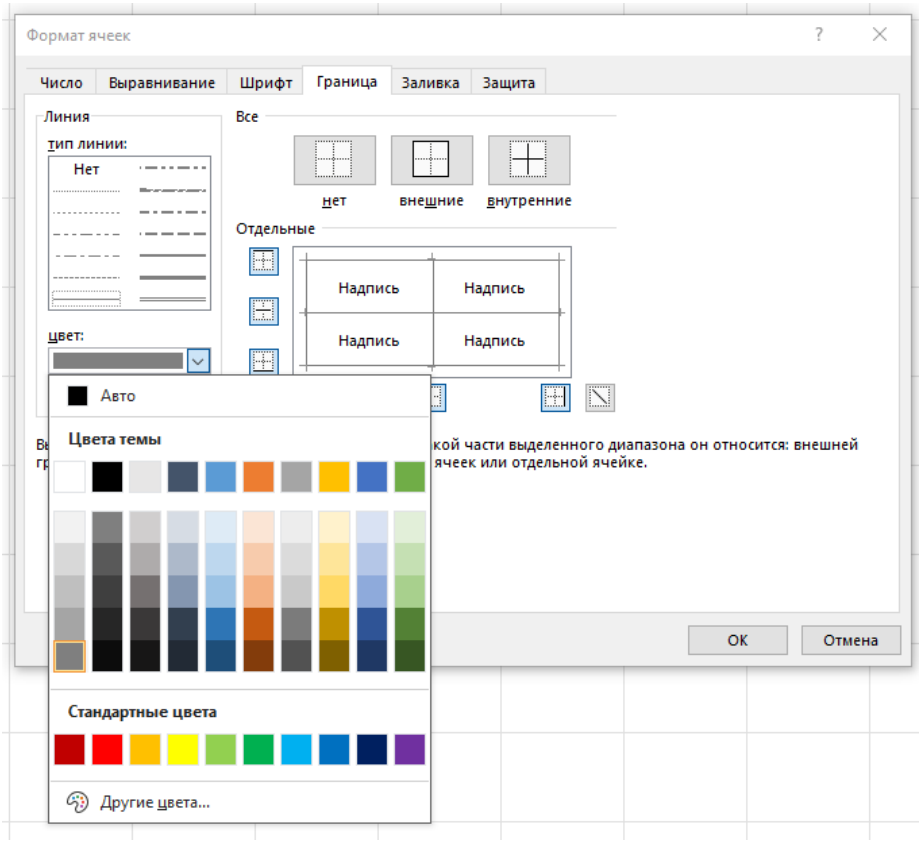


Рисунок 2. Диалоговое окно «Формат ячеек».

Используйте инструмент «Цвет заливки»  (чёрный) для изменения цвета нескольких ячеек как показано на рисунке 3 «Пиксельный символ» ниже.

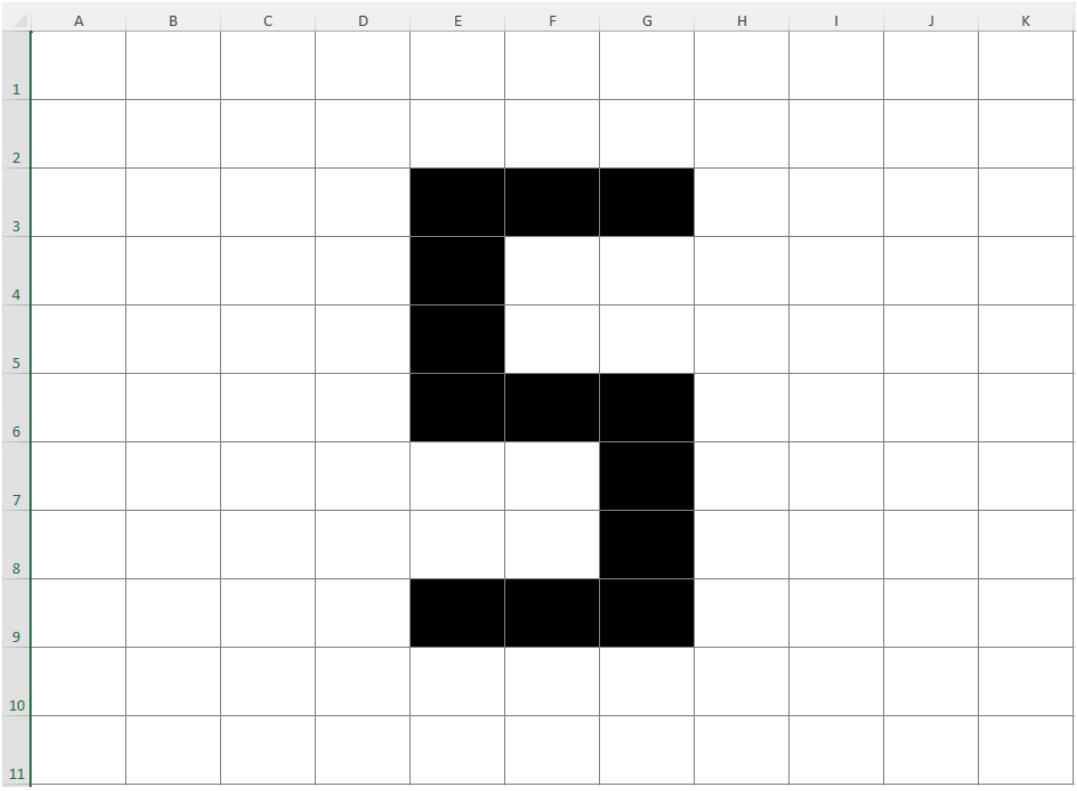
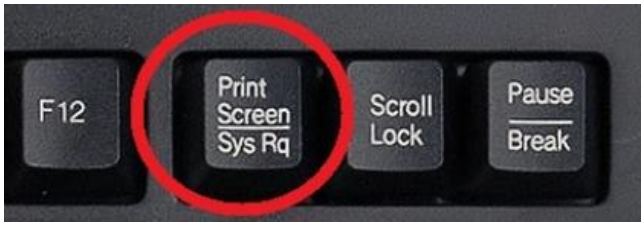
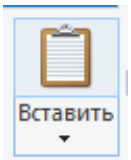


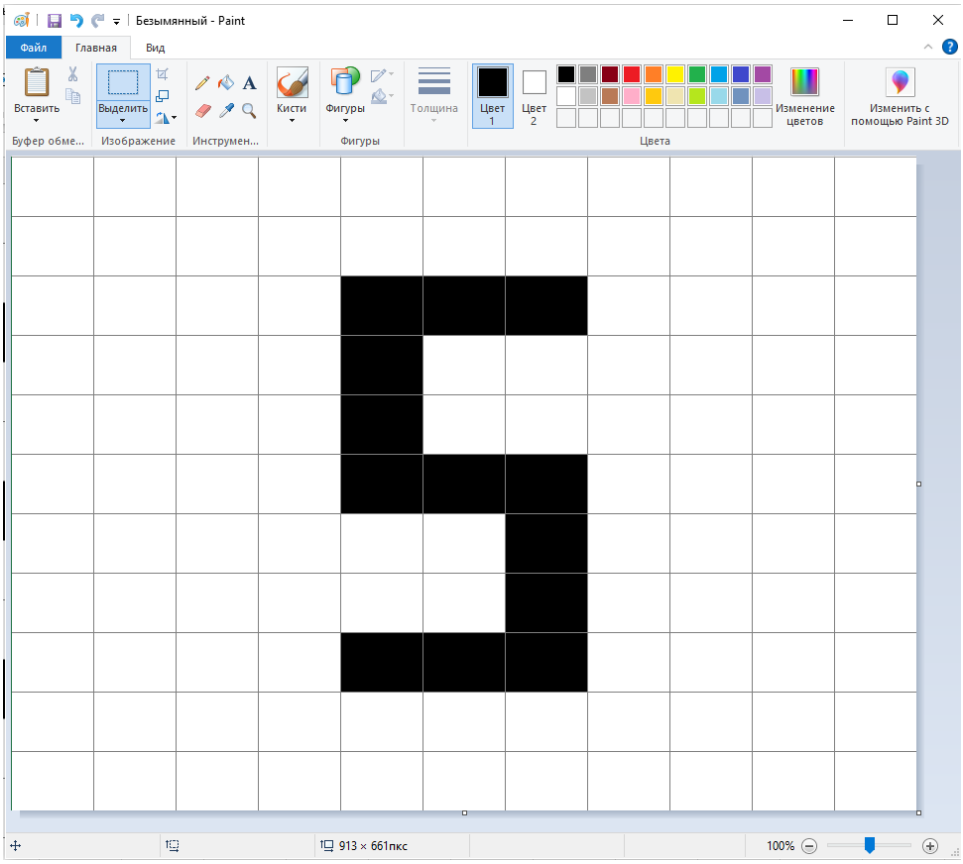
Рисунок 3. Пиксельный символ.

Сделайте снимок экрана с помощью кнопки PrintScreen на клавиатуре.

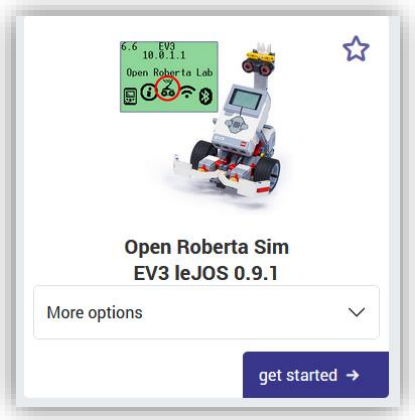


Откройте приложение Paint и используйте кнопку 

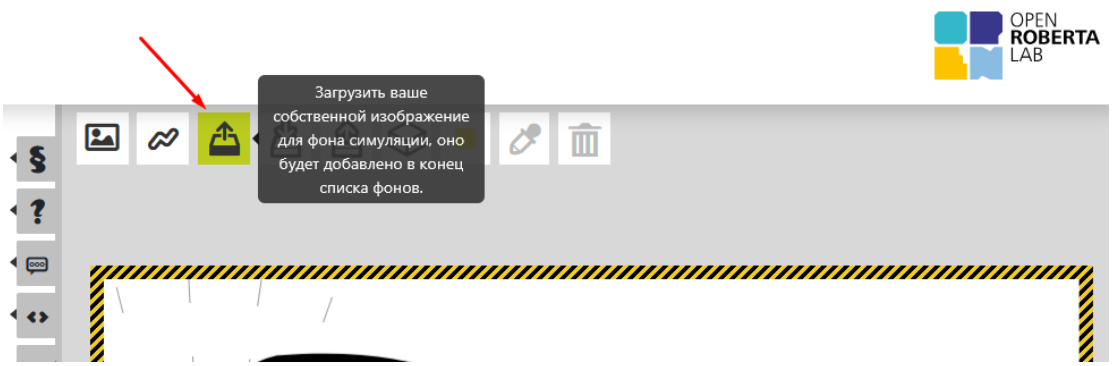
для того, чтобы вставить картинку из буфера обмена. Измените (при необходимости) размер изображения с помощью маркера в правой нижней части холста или используйте диалоговое окно «Свойства» в главном меню приложения.



Откройте сайт https://lab.open-roberta.org/ и выберите пункт



Задание 2. Откройте сайт https://lab.open-roberta.org/ и выберите пункт (показан стрелкой)



Настройте сцену как на Рисунке 4 «Пиксельный символ» ниже. Используйте компьютерную мышь, чтобы изменить позицию робота. Разместите робота как в примере, представленном на изображении ниже.

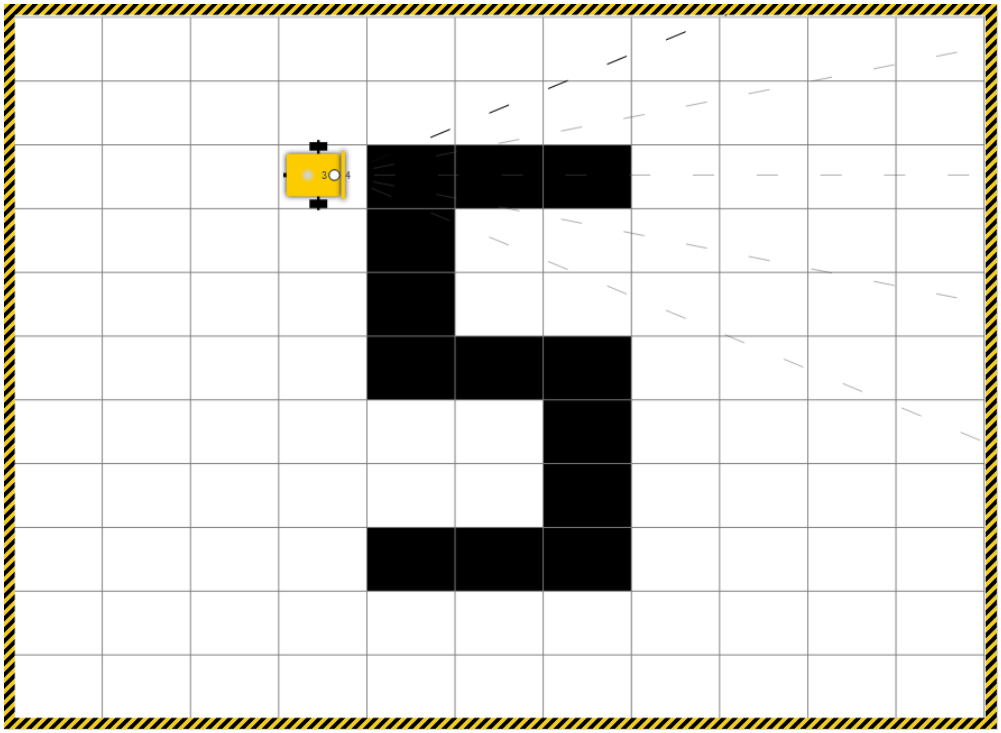


Рисунок 4. «Пиксельный символ»

Задание 3. Составьте программу по примеру, представленному на рисунке 5 «Пример программы «Распознавание символов».

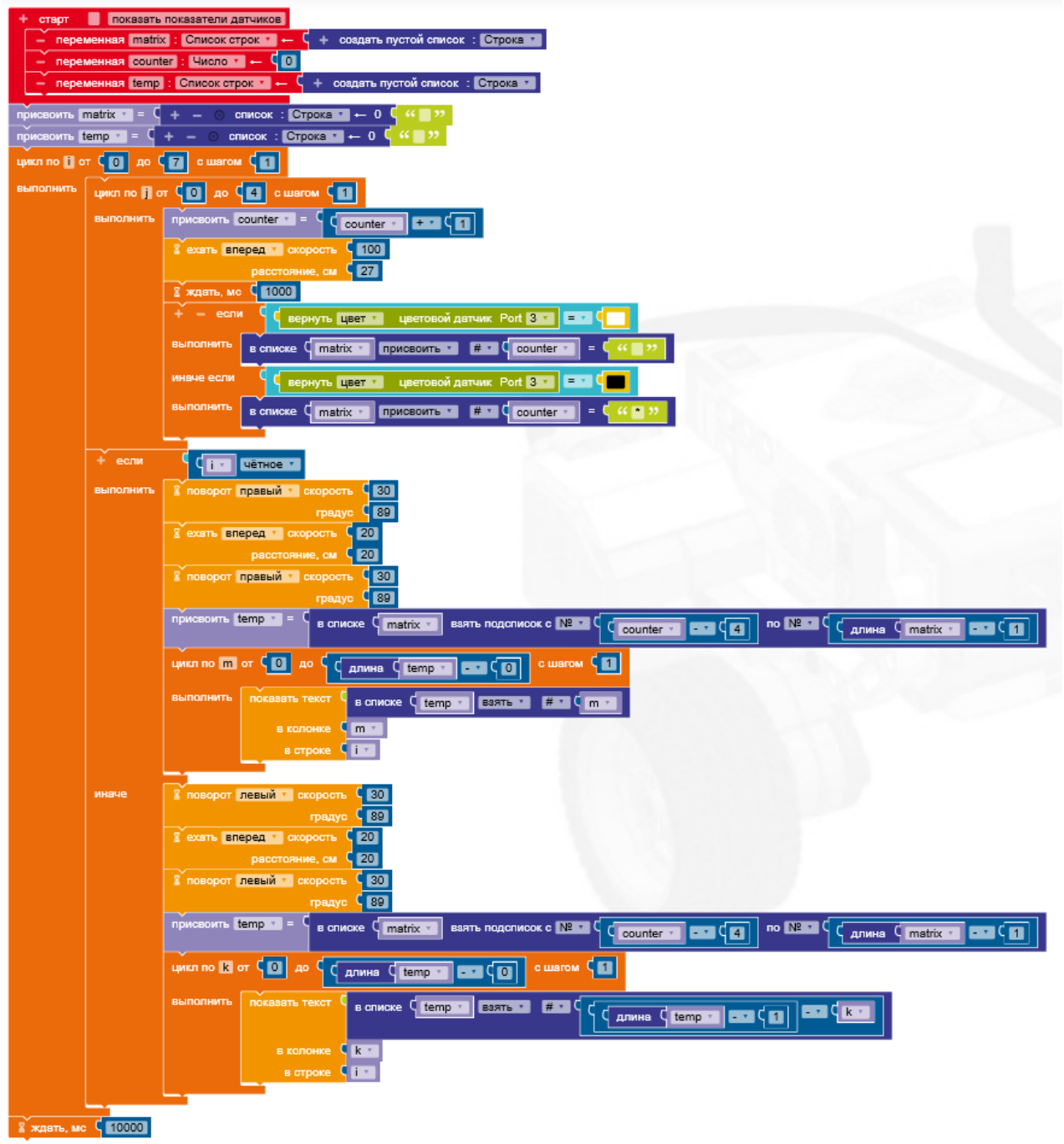


Рисунок 5. Пример программы «Распознавание символов»

Нажмите выделенную прямоугольником кнопку, представленную на изображении ниже для отображения вида от лица робота, в четком соответствии с результатом, показанным ниже на рисунке 6 «Пример вывода распознанного символа».

Для проверки нажмите на кнопке «Старт».

Эта программа имитирует работу **цифрового сканера** с помощью **вложенного цикла**, где внешний цикл отвечает за прохождение 8 рядов, а внутренний — за выполнение 5 последовательных шагов в каждом из них. На каждом шаге робот проезжает дистанцию и использует **условие распознавания**: цветовой датчик проверяет поверхность, и если фиксируется чёрный или белый цвет, программа выполняет **формирование списка**, записывая соответствующую метку в массив matrix под номером текущего шага counter. Чтобы охватить всё поле, робот перемещается «змейкой», меняя направление поворота в зависимости от чётности ряда, а в конце каждого пройденного этапа срабатывает алгоритм **вывода списков**: программа извлекает накопленные данные в промежуточную переменную temp и отображает их на экране контроллера в виде текстовой строки, превращая физические метки на полу в цифровую карту данных.

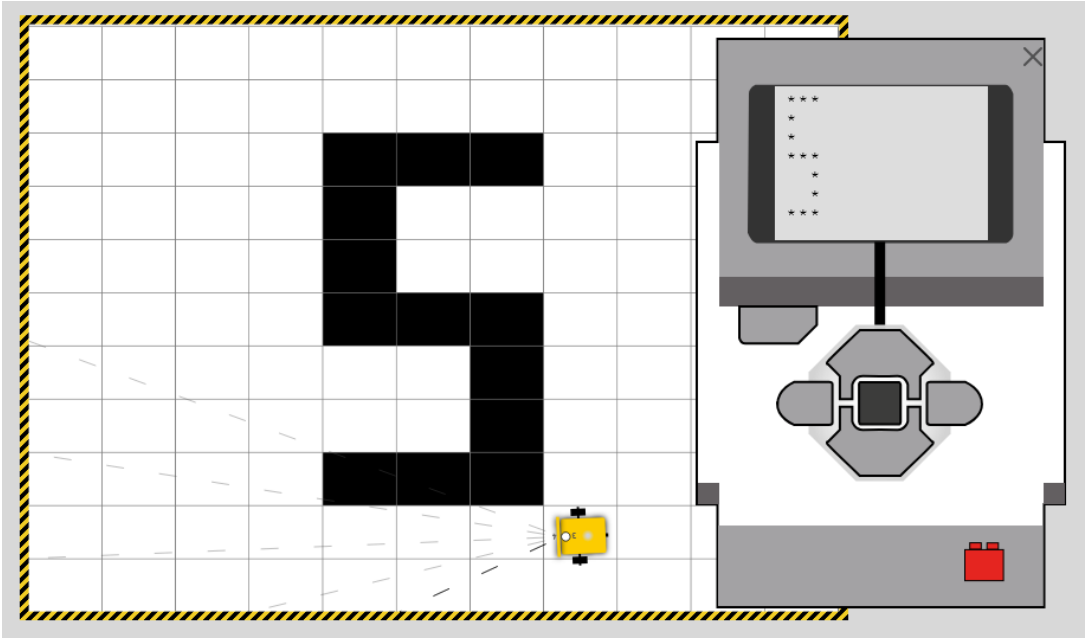


Рисунок 6. Пример вывода распознанного символа.

Остановите программу после выполнения задания.

Для лучшего понимания того, как робот переводит физические действия в цифровой код, ниже представлена таблица-схема работы программы. Она показывает взаимосвязь между движениями робота, индексами в памяти и итоговым выводом данных.

Этап (Цикл)	Действие робота (Физика)	Работа с данными (Списки)	Вывод на экран (Интерфейс)
Внешний цикл (i)	Определяет номер текущего ряда (от 0 до 7).	Подготовка временного списка temp для новой строки.	Переход на новую строку на дисплее контроллера.
Внутренний цикл (j)	Делает 5 шагов по 27 см в текущем ряду.	На каждом шаге counter увеличивается на 1.	Сбор данных о цвете в реальном времени.
Условие (Если/Иначе)	Проверка цвета датчиком (черный/белый).	Запись метки " " в общий список matrix по индексу counter.	—
Повороты (Четность)	Если ряд чётный — направо, если нечётный — налево.	Сохранение структуры «змейки» в памяти.	—
Формирование строки	Завершение прохода одного ряда.	Копирование части данных из matrix во временный список temp.	Отображение текстовой строки temp в колонке m и строке i.

Пояснение:

- **matrix** — это «большой склад», где хранятся все 40 считанных меток (8 рядов по 5 шагов).
- **counter** — это «порядковый номер», который не дает роботу записать два разных цвета в одну и ту же ячейку.
- **temp** — это «текущий черновик», в который робот копирует только последние 5 значений, чтобы красиво напечатать их на экране одной строкой.

Дескрипторы:

- Создаются переменные matrix, counter и temp для хранения данных.
- Устанавливаются начальные значения переменных (пустые списки и число 0).
- Используется внешний цикл «от 0 до 7» для управления проходом 8 рядов.
- Используется вложенный (внутренний) цикл «от 0 до 4» для совершения 5 шагов в каждом ряду.
- Используется действие «Присвоить переменной» для увеличения counter на каждом шаге.
- Используется действие «Ехать вперед» на расстояние 27 см в каждом цикле сканирования.
- Используется датчик цвета (Port 3) для распознавания поверхности.
- Используется логический контроль «Если / иначе если» для проверки считанного цвета (белый/черный).
- Используется действие «В списке... присвоить» для записи результата сканирования в matrix по индексу counter.
- Используется условие проверки чётности переменной i для выбора направления движения.
- Используются действия «Поворот» (правый/левый) на 89 градусов для реализации движения «змейкой».
- Используется операция «Взять подсписок» для копирования данных из matrix в переменную temp.
- Используется цикл для перебора элементов списка temp.

- Используется действие «Показать текст» для вывода считанных данных на экран контроллера в виде координат (колонка m/k, строка i).
- Программа запускается: робот последовательно сканирует поле и формирует цифровую карту в реальном времени.

Простые математические задачи по робототехнике.

1. **Маршрут робота**. Сценарий: Робот стартует в точке (0;0) на координатной сетке и выполняет команды: → → ↑ ← ↑ →. Вопрос: В какой точке окажется робот после выполнения всех команд?

Ответ: Робот окажется в точке (2;2).

2. **Ошибка в программе**. Сценарий: Робот должен пройти по квадрату со стороной 3 клетки и вернуться в исходную точку. Вопрос: Сколько команд поворота должно быть в правильной программе?

Ответ: В программе должно быть 4 команды поворота.

3. **Повтори команду.** Сценарий: Робот выполняет команду: «повторить 4 раза: шаг вперёд, поворот направо». Вопрос: Какую фигуру опишет робот?

Ответ: Робот опишет квадрат.

4. **Шаги робота.** Сценарий: Один шаг робота равен 25 см. Вопрос: Сколько шагов нужно, чтобы пройти 5 метров?

Ответ: Нужно сделать 20 шагов.

5. **Скорость движения.** Сценарий: Робот делает 2 шага в секунду. Вопрос: За сколько секунд он выполнит 30 шагов?

Ответ: За 15 секунд.

6. **Периметр маршрута.** Сценарий: Робот обходит прямоугольник размером 4×6 клеток. Вопрос: Сколько шагов сделает робот?

Ответ: Робот сделает 20 шагов.

7. **Угол поворота**. Сценарий: Робот делает полный оборот за 4 одинаковых поворота. Вопрос: Чему равен угол одного поворота?

Ответ: Один поворот равен 90°.

8. **Координатная сетка.** Сценарий: Робот находится в точке (2;3). Он делает 3 шага вправо и 2 шага вниз. Вопрос: В какой точке он окажется?

Ответ: Робот окажется в точке (5;1).

9. **Сбор данных.** Сценарий: Робот записывает данные каждые 10 секунд в течение 2 минут. Вопрос: Сколько измерений будет сделано?

Ответ: Будет сделано 12 измерений.

10. **Обороты колеса.** Сценарий: Диаметр колеса робота равен 28 см. Робот должен проехать 20 см. Вопрос: Сколько оборотов сделает колесо?

Ответ: Длина окружности ≈ 88 см. 20 : 88 ≈ 0,23 оборота.

11. **Длина пути.** Сценарий: Робот сделал 5 полных оборотов колеса диаметром 10 см. Вопрос: Какое расстояние проехал робот?

Ответ: Длина окружности ≈ 31,4 см. 5 × 31,4 ≈ 157 см.

12. **Периметр поля.** Сценарий: Робот обходит прямоугольное поле 5×3 м. Вопрос: Какое расстояние он пройдёт?

Ответ: Периметр = 16 м.

13. **Деление маршрута.** Сценарий: Маршрут длиной 120 см делится на равные участки по 15 см. Вопрос: Сколько участков получится?

Ответ: 120 : 15 = 8 участков.

14. **Подсчёт попыток.** Сценарий: Робот прошёл маршрут за 3, 4 и 5 попыток. Вопрос: Сколько всего попыток?

Ответ: 12 попыток.

15. **Частота ошибок.** Сценарий: Из 10 запусков робот ошибся 2 раза. Вопрос: Какова вероятность ошибки?

Ответ: 2 из 10 (20%).

16. **Проверка условия.** Сценарий: Робот выполняет команду, если температура ниже 30 °C. Вопрос: Выполнит ли он команду при 28 °C?

Ответ: Да, выполнит.

17. **Анализ данных движения**. Сценарий: Робот прошёл маршрут 6 раз. В 4 случаях время было меньше 20 секунд. Вопрос: В скольких случаях время было больше или равно 20 секунд?

Ответ: В 2 случаях.

18. **Нахождение ошибки.** Сценарий: Робот должен сделать 12 шагов, выполняя цикл по 5 шагов. Вопрос: Почему задание выполнится неверно?

Ответ: 12 не делится на 5 без остатка.

19. **Анализ поворотов.** Сценарий: Робот повернул направо 3 раза по 90°. Вопрос: В каком направлении он смотрит относительно начала?

Ответ: Влево.

20. **Логика координат.** Сценарий: Робот переместился с (2;2) в (2;7). Вопрос: На сколько единиц он сместился?

Ответ: На 5 единиц.

21. **Истинность высказываний.** Сценарий: A — «робот движется», B — «датчик видит препятствие». Робот движется, если НЕ B. Задание: В каком случае робот остановится?

Ответ: Когда датчик видит препятствие (B — истина).

22. **Логическая цепочка.** Сценарий: Если робот движется быстро, то он расходует больше энергии. Если расход энергии большой, то батарея разряжается быстрее. Задание: Какой вывод логически следует?

Ответ: Если робот движется быстро, то батарея разряжается быстрее.

23. **Выбор маршрута**. Сценарий: Есть три маршрута: A — 10 шагов, 4 поворота; B — 12 шагов, 2 поворота; C — 9 шагов, 6 поворотов. Робот выбирает маршрут с наименьшим числом шагов, а при равенстве — с меньшим числом поворотов. Задание: Какой маршрут выберет робот?

Ответ: Маршрут C (9 шагов).

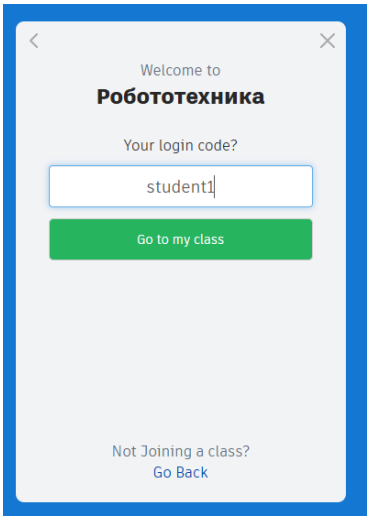
Тема. «Интернет вещей». Умный дом. Охранная сигнализация.

Цели обучения (ОШ): 11.3.4.8 описывать принципы работы "интернета вещей"; 11.5.2.4 организовывать передачу данных с датчиков умного дома; 11.5.2.5 разрабатывать программу для вывода данных, полученных с датчиков умного дома

Вам понадобится

1. **https://tinkercad.com**
2. **Arduino IDE**

Задание 1. Введите или **нажмите на ссылку** для присоединения к классу https://www.tinkercad.com/joinclass/XBHA6CDJI. После открытия сайта по ссылке **нажмите на кнопке «Join with login code»** («Присоединиться с помощью кода для входа»). Используйте для входа номер вашего ноутбука после слова student, в соответствии с примером показанным на рисунке ниже. Например номер вашего ноутбука 13, тогда ваш код для входа будет выглядеть как student13. После ввода кода нажмите на кнопке «**Go to my class**».



После входа в учетную запись (рисунок 1), нажмите на кнопке указанной стрелкой «**Create your first Circuits design**» («Создайте свой первый дизайн схем»).

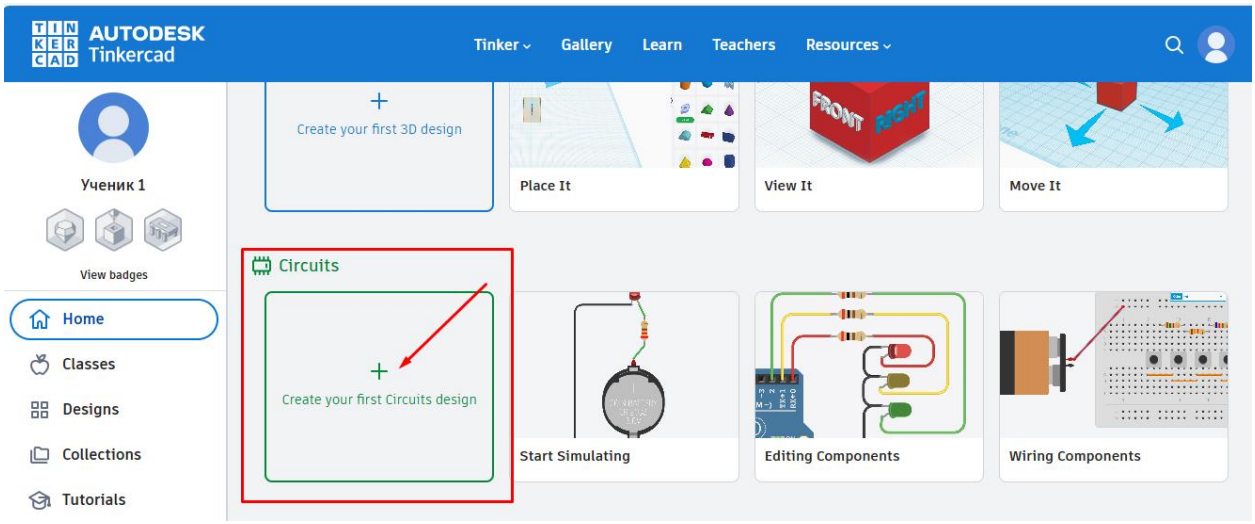
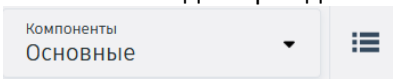


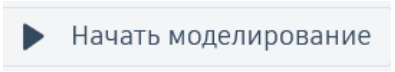
Рисунок 1. Главная страница учетной записи.

Все необходимые компоненты находятся справа (рисунок 2). Эти электронные компоненты нужны для создания различных устройств, для выполнения практической работы. Для размещения компонентов перемещайте компоненты с помощью мыши из панели справа в рабочую область.

Панель инструментов Виртуальной лаборатории электроники состоит из

-  Кнопка вращения деталью для эксперимента
-  Кнопка удаления выбранной детали для эксперимента
- 

Редактор кода на языке Arduino который выполняется в фреймворке
- 

Набор деталей и компонентов для эксперимента
- 

Кнопка старта симуляции

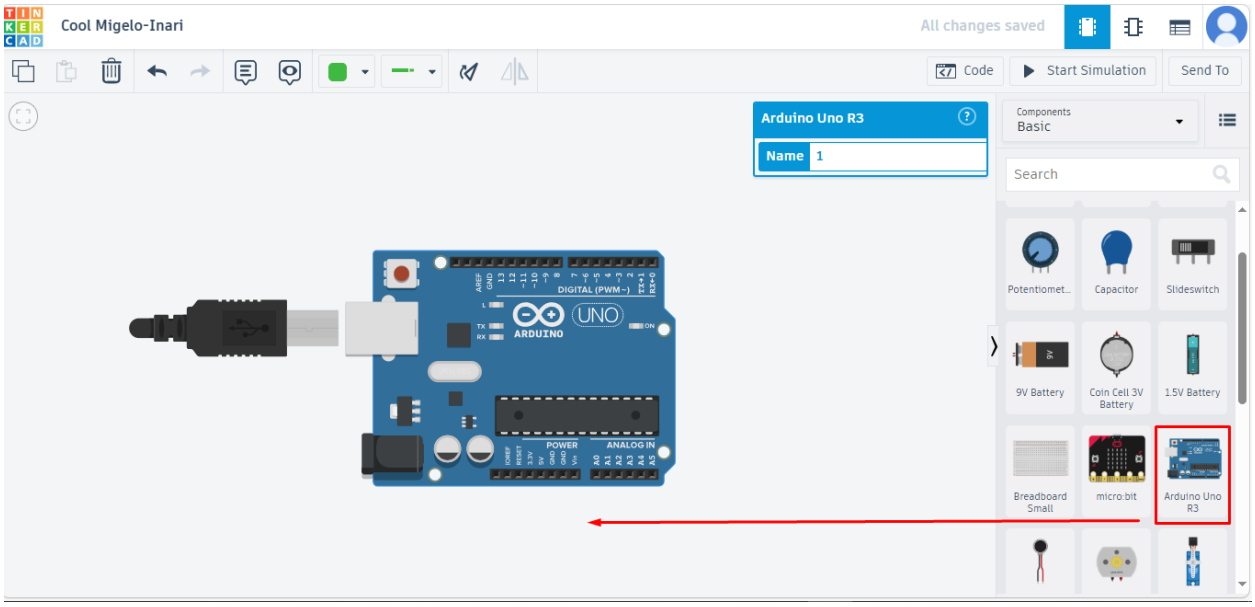
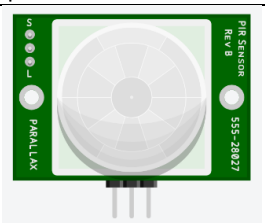
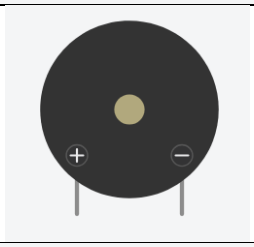
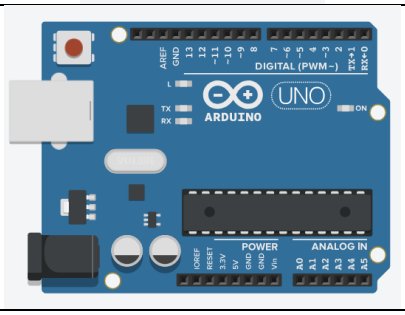
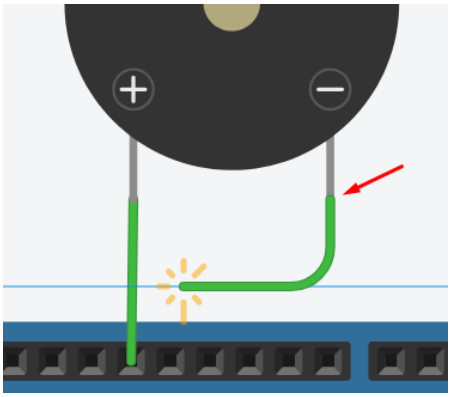


Рисунок 2. Палитра компонентов.

Название компонента	Внешний вид
Пироэлектрический ИК-датчик *ИК – сокращение от инфракрасный	
Пьезоэлемент	
Arduino UNO	

Задание 2. Соберите схему в соответствии с примером показанным на рисунке 2 «Электронная схема охранной сигнализации». Для добавления соединений между компонентами нажимайте на контакте устройства ввода информаци (Пироэлектрический ИК-датчик), или устройстве вывода информации (Пьезоэлемент) и тяните виртуальный «провод» как показано на рисунке ниже до контакта на плате Arduino UNO.



Подробно по ссылке https://youtu.be/XUF58SKqBtU?si=a8W32f22l2-UEyMr

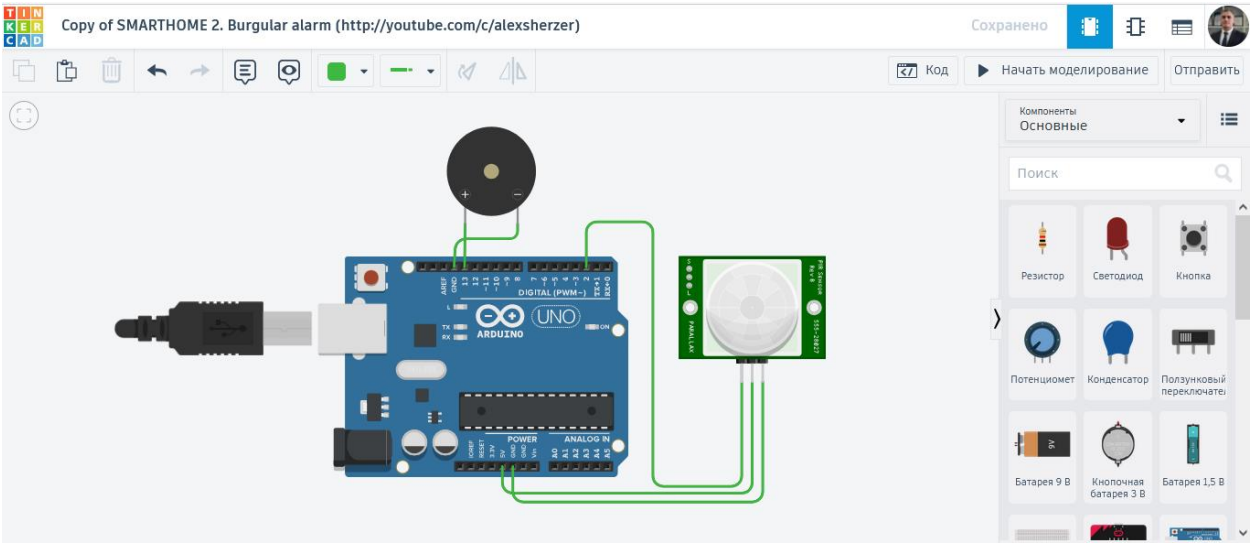
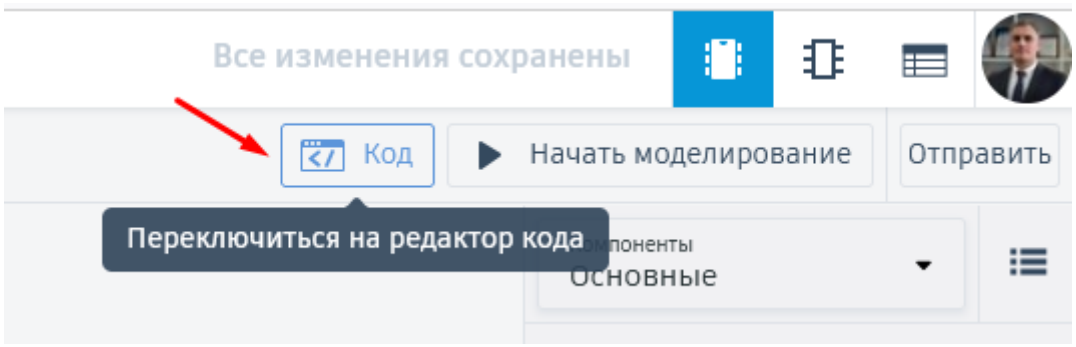


Рисунок 3. «Электронная схема охранной сигнализации»

Нажмите на кнопке «Код»



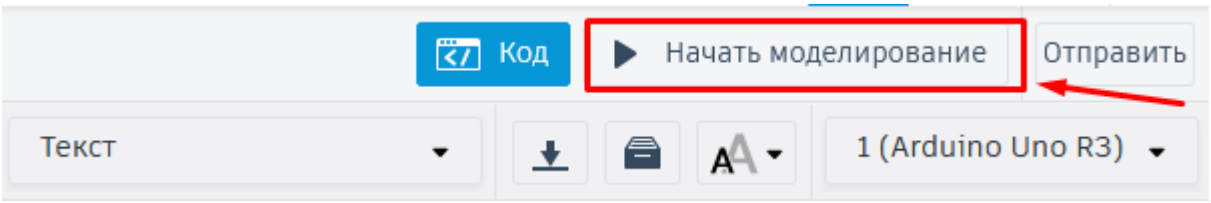
Задание 3. Допишите или измените код (текст программы который следует дописать выделен полужирным) в соответствии с примером ниже.

Код проекта «Электронная схема охранной сигнализации»

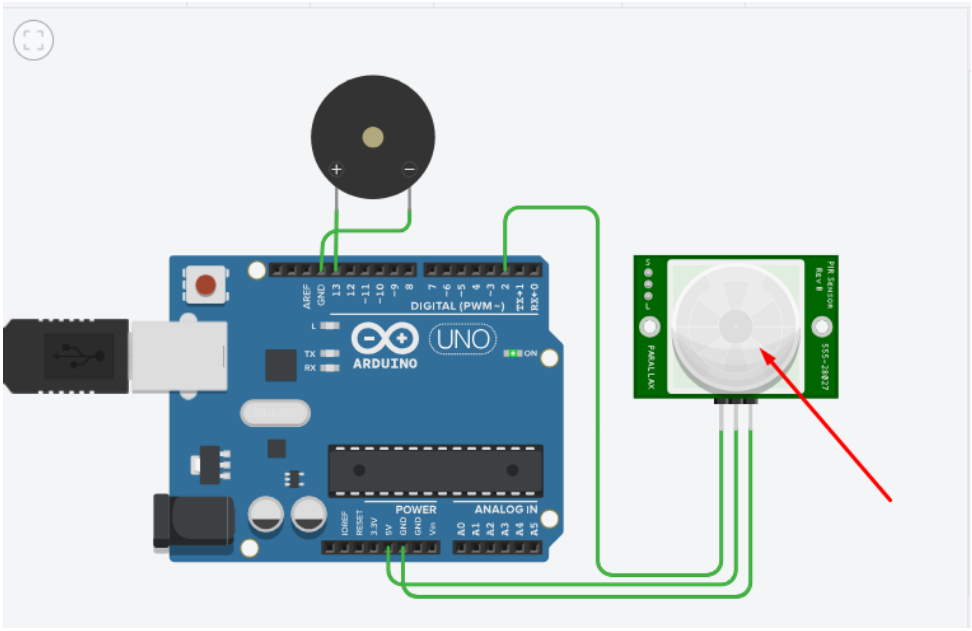
```
#define PIR 2
#define BZR 13
int val,state;

void setup()
{

  pinMode(PIR, INPUT);
  pinMode(BZR, OUTPUT);
  Serial.begin(9600);

}

void loop()
{

  state = digitalRead(PIR);
  Serial.println(state);
  if (state==HIGH) {
    tone (BZR,350,15);
    delay(250);
  }

}
```

Объяснение. Код для Arduino Uno настраивает пин 2 как вход для пироэлектрического ИК-датчика, а пин 13 как выход для пьезоэлемента, после чего в бесконечном цикле считывает состояние датчика с помощью digitalRead: при обнаружении движения (сигнал HIGH) с помощью функции tone() генерируется звуковой сигнал заданной частоты и длительности с небольшой задержкой delay, а текущее состояние дополнительно выводится в Serial Monitor для контроля работы системы.

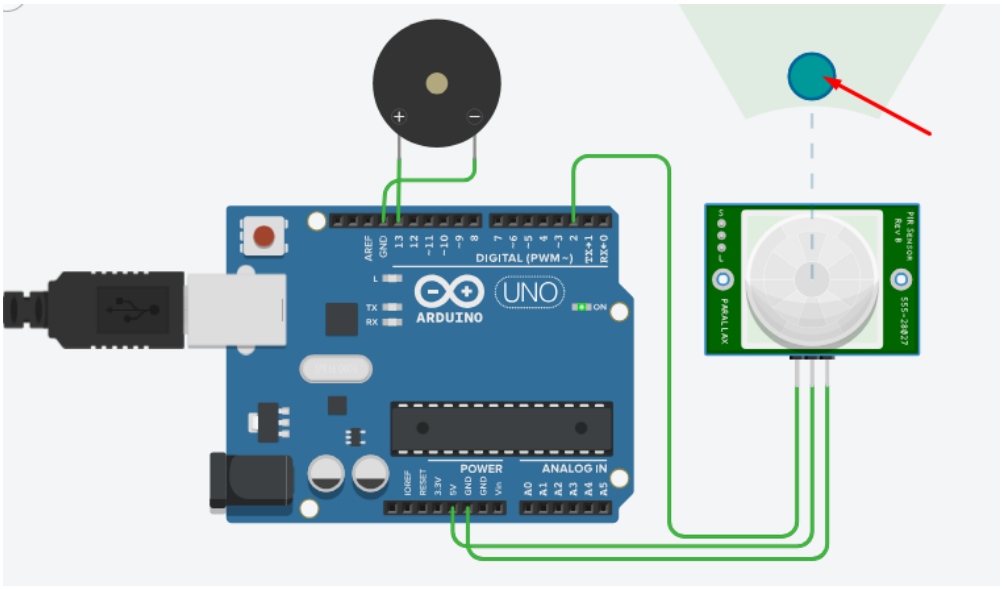
Задание 4. Нажмите на кнопке «Начать моделирование»



Нажмите на датчике (указан стрелкой)



Измените положение точки (указана стрелкой). Вы должны услышать звук пьезоэлемента, имитирующий охранную сигнализацию.

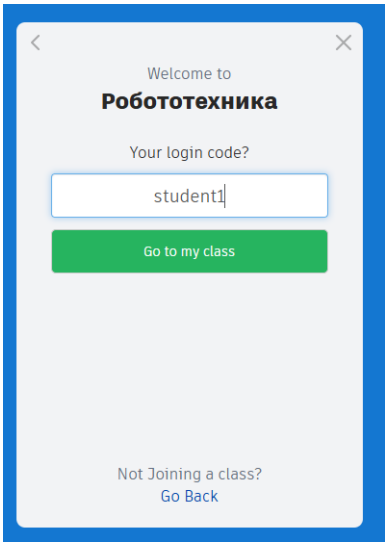


Дескрипторы

- Входит в Tinkercad и создаёт проект схемы
- Добавляет и размещает компоненты: Arduino Uno, PIR sensor, piezo buzzer
- Выполняет корректное подключение элементов к соответствующим пинам
- Открывает редактор и вводит/редактирует программный код
- Настраивает пины и реализует чтение данных с датчика (pinMode, digitalRead)
- Реализует условие включения сигнализации и генерацию звука (if, tone, delay)
- Запускает моделирование и тестирует работу датчика
- Анализирует результат работы и при необходимости исправляет ошибки

Тема. «Интернет вещей». Умный дом. Управление освещением.

Задание 1. Введите или **нажмите на ссылку** для присоединения к классу https://www.tinkercad.com/joinclass/XBHA6CDJI. После открытия сайта по ссылке **нажмите на кнопке «Join with login code»** («Присоединиться с помощью кода для входа»). Используйте для входа номер вашего ноутбука после слова student, в соответствии с примером показанным на рисунке ниже. Например номер вашего ноутбука 13, тогда ваш код для входа будет выглядеть как student13. После ввода кода нажмите на кнопке «**Go to my class**».



После входа в учетную запись (рисунок 1), нажмите на кнопке указанной стрелкой «**Create your first Circuits design**» («Создайте свой первый дизайн схем») или нажмите на кнопке «**Create**» → «**Cicruits**» («Создать» → «Цепи») как показано на рисунке ниже.

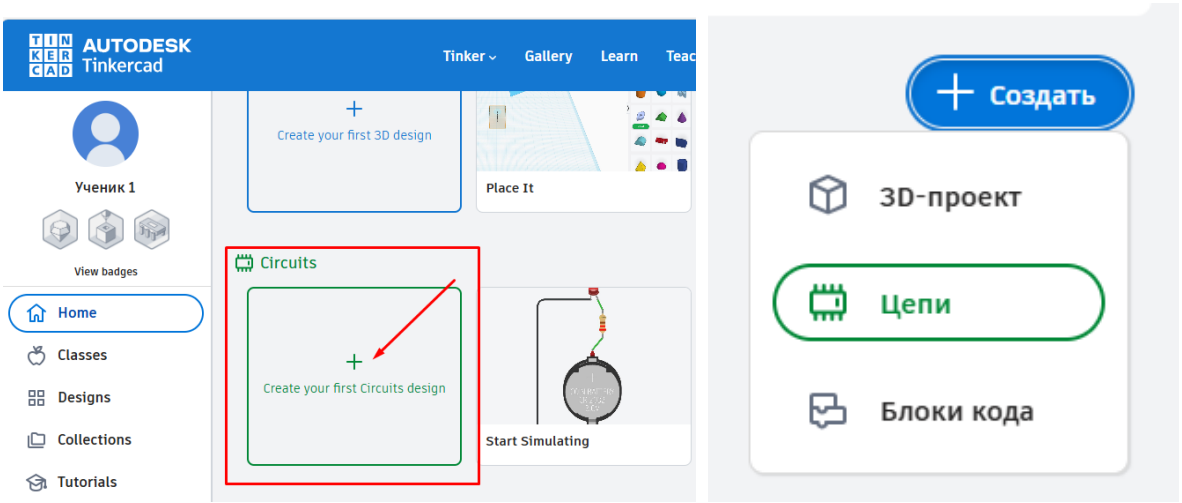


Рисунок 1. Главная страница учетной записи.

Все необходимые компоненты находятся справа (рисунок 2). Эти электронные компоненты нужны для создания различных устройств, для выполнения практической работы. Для размещения компонентов перемещайте компоненты с помощью мыши из панели справа в рабочую область.

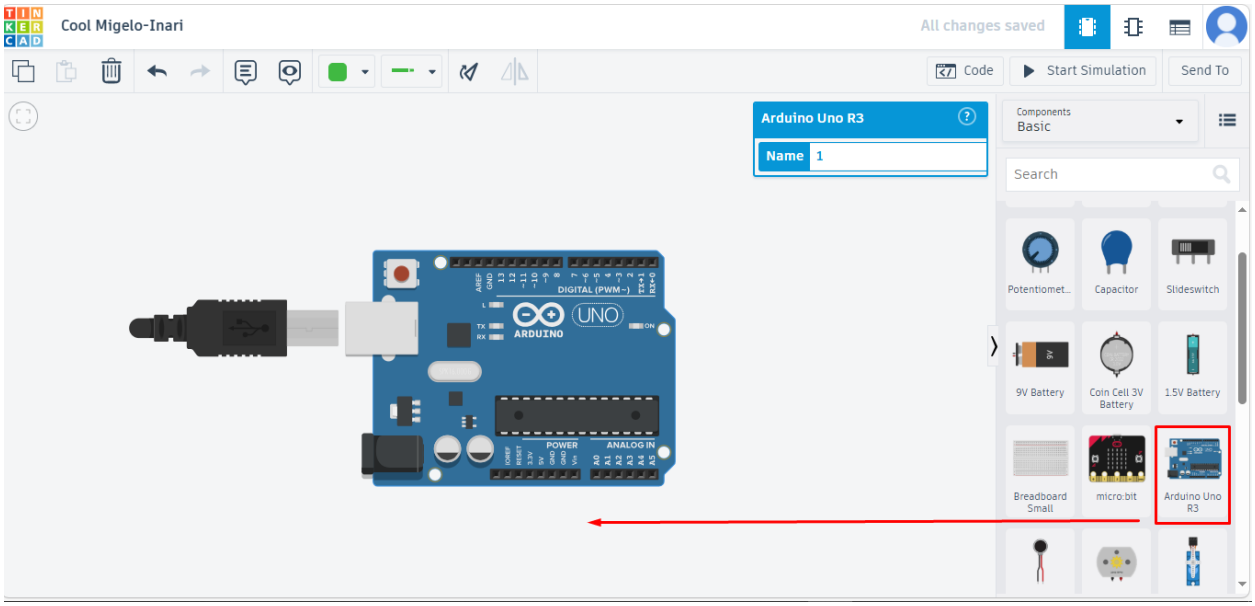
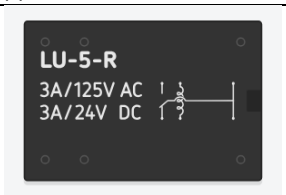
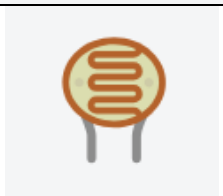
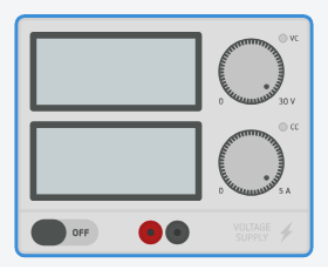
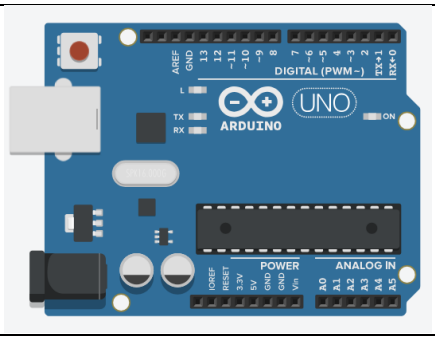
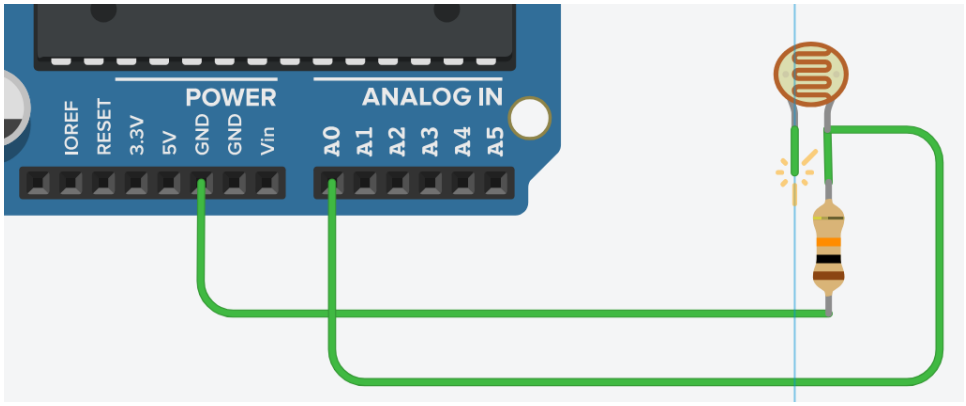


Рисунок 2. Палитра компонентов.

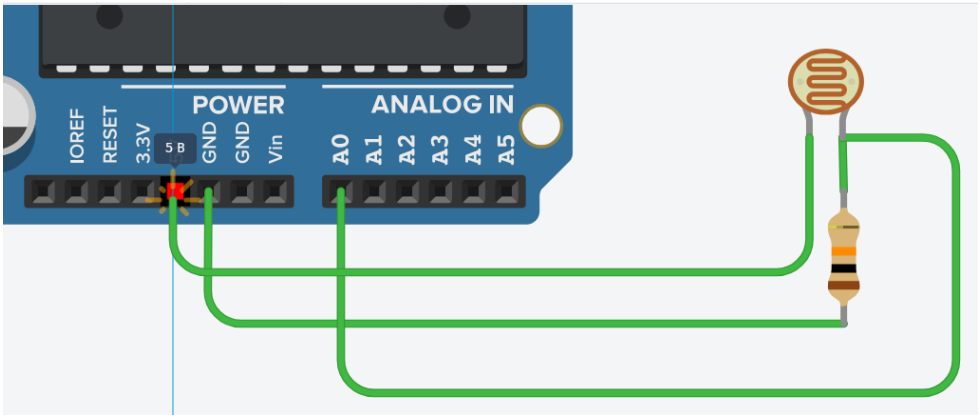
Название и назначение компонента	Внешний вид
Однополосное реле. Переключает электрическую цепь по сигналу (включает/выключает устройства).	
Резистор 10 кОм (10 kΩ). Ограничивает ток и защищает компоненты.	
Фоторезистор. Реагирует на свет, меняя своё сопротивление.	
Лампа накаливания. Светится при прохождении тока.	
Источник питания. Напряжение: 30 Вольт (30V). Сила тока: 5 Ампер (5A). Обеспечивает электрической энергией схему.	

Arduino UNO. Управляет работой всей схемы и выполняет программу.	

Задание 2. Соберите схему в соответствии с примером показанным на рисунке 3 «Электронная схема управления освещением». Для добавления соединений между компонентами нажимайте на контакте устройства ввода информаци (фоторезистор), или устройстве вывода информации (лампа накаливания) и тяните виртуальный «провод» как показано на рисунке ниже до контакта на плате Arduino UNO.



Шаг 1



Шаг 2

Подробно по ссылке https://youtu.be/RXSAYLt7hc8?si=AE_qsi1Us3Eakkqk

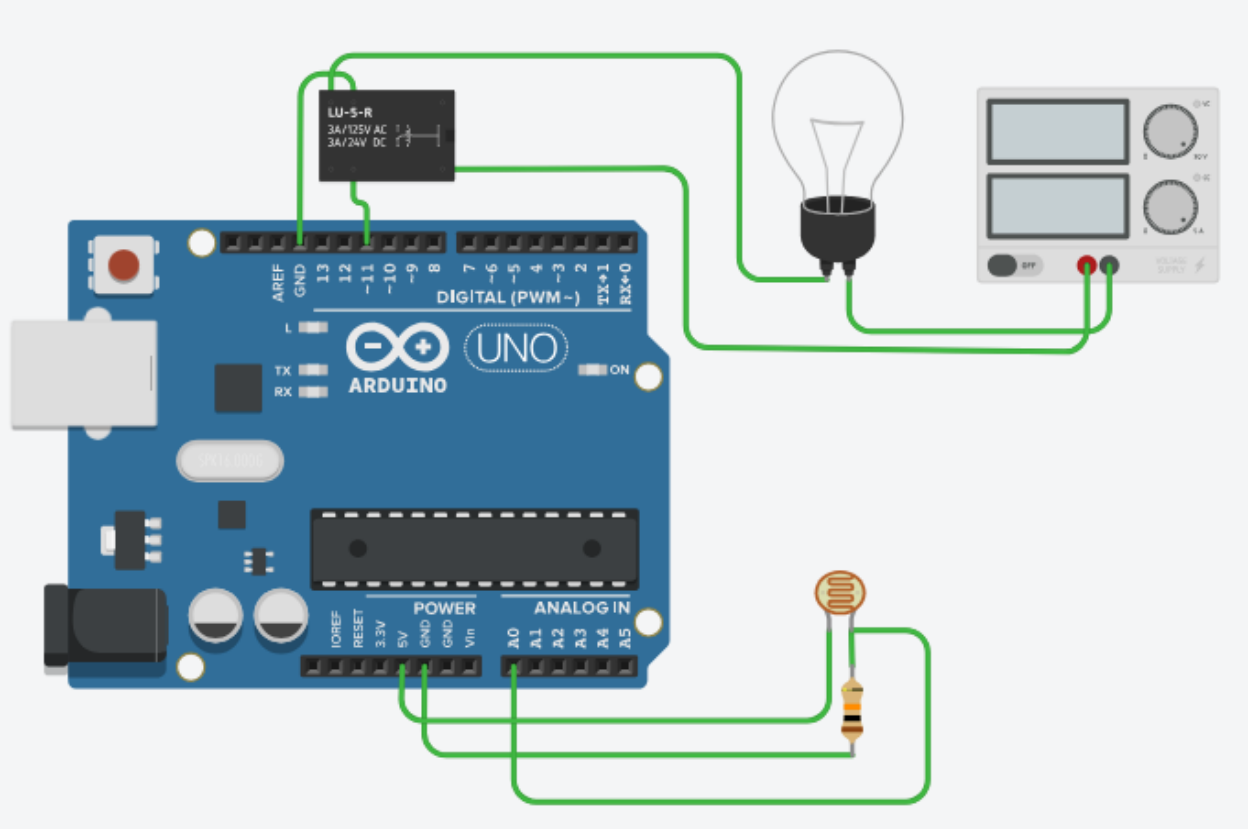
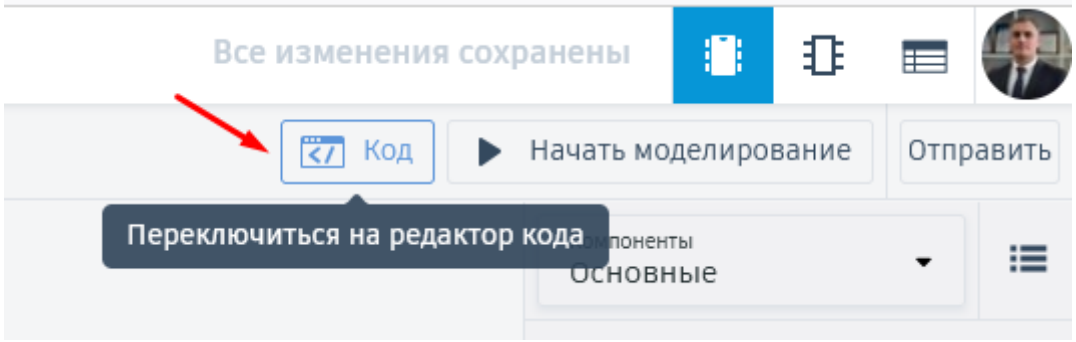


Рисунок 3. «Электронная схема управления освещением»

Нажмите на кнопке «Код»



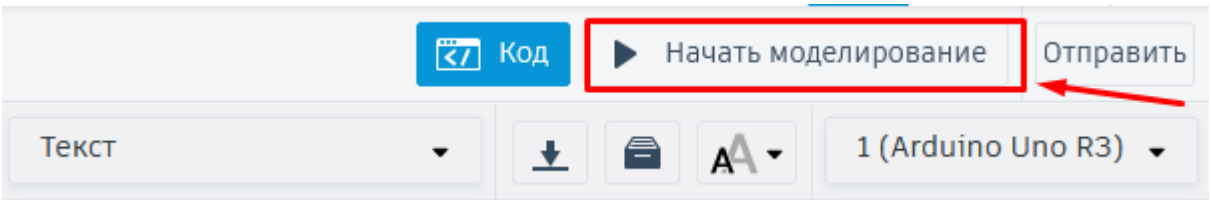
Задание 3. Допишите или измените код (текст программы, который следует дописать выделен полужирным) в соответствии с примером ниже.

Код проекта «Электронная схема управления освещением»

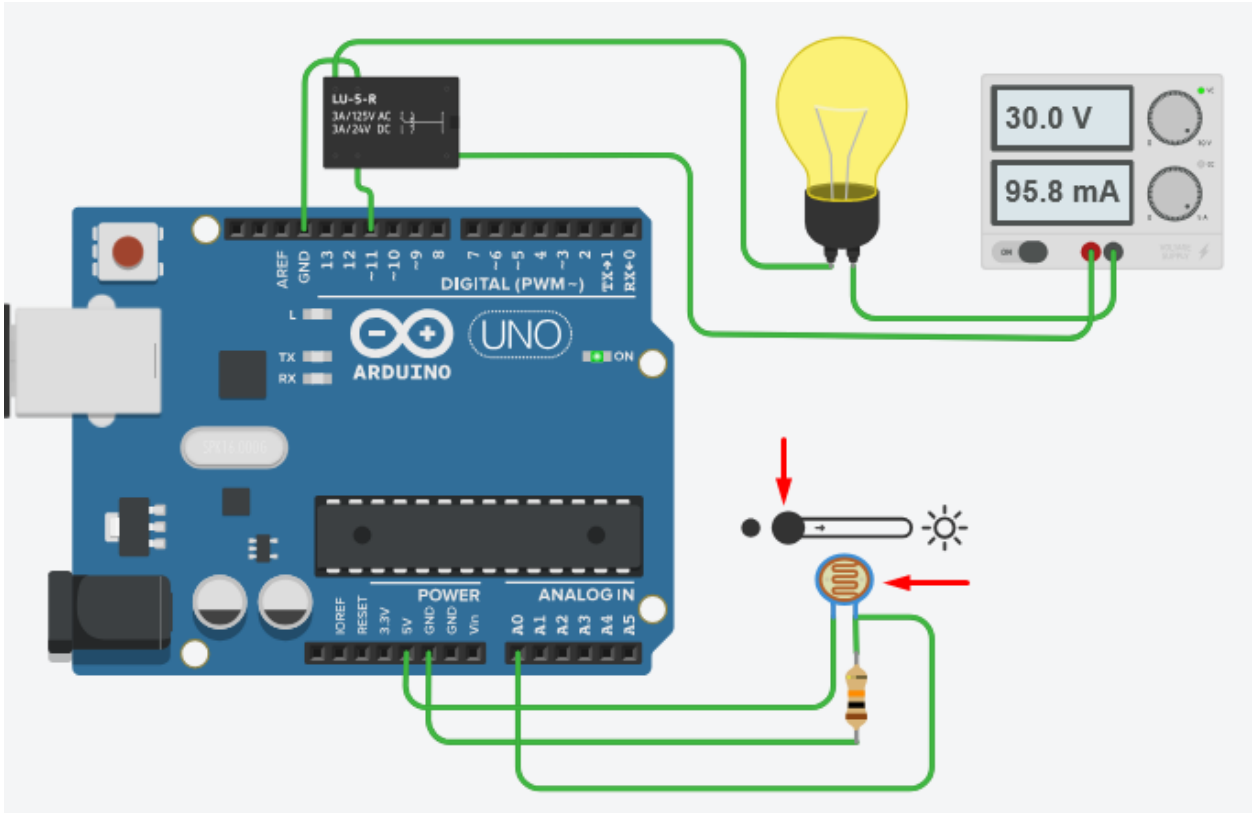
```
#define REL 11
#define PH 0
int val, state;

void setup()
{

  pinMode(PH,INPUT);
  pinMode(REL,OUTPUT);
  Serial.begin(9600);

}

void loop()
{
        state = analogRead(PH);
        val = map(state,50,970,67,0);
        analogWrite(REL,val);
        Serial.println(val);

}
```

Объяснение. На схеме показано, как плата Arduino управляет лампочкой с помощью реле в зависимости от освещённости: фоторезистор (датчик света) вместе с резистором образует делитель напряжения и подключён к аналоговому входу A0, поэтому Arduino может «понимать», светло или темно; реле подключено к цифровому пину 11 и работает как переключатель, который включает или выключает лампу; в программе сначала задаются номера пинов (REL — реле, PH — датчик), затем в setup() настраиваются режимы работы (датчик — вход, реле — выход) и включается связь с компьютером, а в loop() постоянно считывается значение света (analogRead), преобразуется в нужный диапазон с помощью map и подаётся на реле (analogWrite), при этом значение выводится в монитор порта, то есть чем темнее, тем сильнее сигнал на реле и лампа включается.

Задание 4. Нажмите на кнопке «Начать моделирование»



Нажмите на датчике «Фоторезистор» (указан стрелкой). Измените положение точки (указана стрелкой). Вы должны увидеть как изменяется яркость лампы накаливания в зависимости от освещения, где символ в форме солнца указывает на время суток – день.

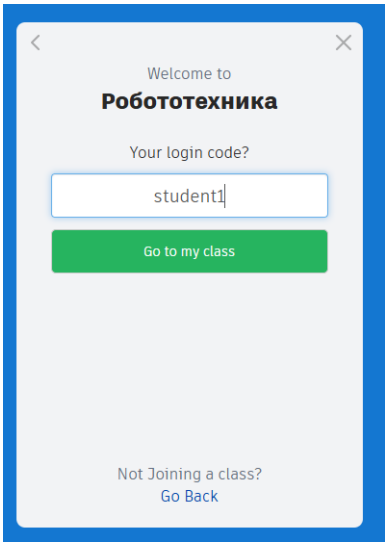


Дескрипторы

- Входит в Tinkercad и создаёт проект схемы
- Добавляет компоненты: Arduino Uno, реле, лампу, фоторезистор и резистор
- Выполняет подключение элементов к пинам (A0 для датчика света, 11 для реле)
- Собирает делитель напряжения с фоторезистором для получения данных об освещённости
- Открывает редактор и вводит/редактирует программный код
- Настраивает пины и реализует чтение аналогового сигнала с датчика (pinMode, analogRead)
- Преобразует полученные данные с помощью функции map
- Реализует управление реле через analogWrite в зависимости от уровня освещения
- Выводит данные в монитор порта (Serial.begin, Serial.println)
- Запускает моделирование и тестирует работу схемы (изменяет освещённость датчика)
- Анализирует результат работы и при необходимости исправляет ошибки в схеме или коде

Тема. «Интернет вещей». Умный дом. Проект «Электронный дверной замок»

Задание 1. Введите или **нажмите на ссылку** для присоединения к классу https://www.tinkercad.com/joinclass/XBHA6CDJI. После открытия сайта по ссылке **нажмите на кнопке «Join with login code»** («Присоединиться с помощью кода для входа»). Используйте для входа номер вашего ноутбука после слова student, в соответствии с примером показанным на рисунке ниже. Например номер вашего ноутбука 13, тогда ваш код для входа будет выглядеть как student13. После ввода кода нажмите на кнопке «**Go to my class**».



После входа в учетную запись (рисунок 1), нажмите на кнопке указанной стрелкой «**Create your first Circuits design**» («Создайте свой первый дизайн схем») или нажмите на кнопке «**Create**» → «**Cicruits**» («Создать» → «Цепи») как показано на рисунке ниже.

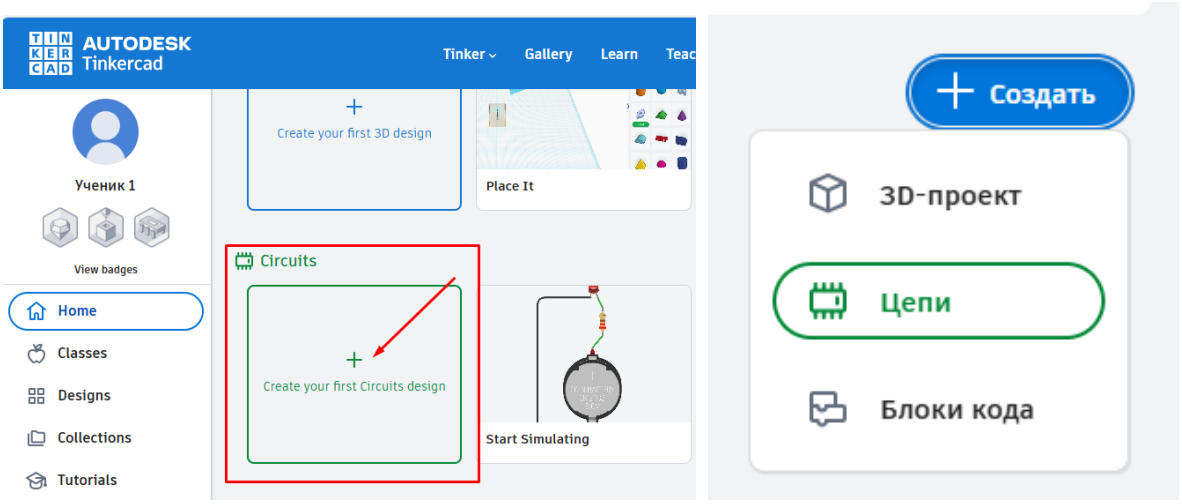


Рисунок 1. Главная страница учетной записи.

Все необходимые компоненты находятся справа (рисунок 2). Эти электронные компоненты нужны для создания различных устройств, для выполнения практической работы. Для размещения компонентов перемещайте компоненты с помощью мыши из панели справа в рабочую область.

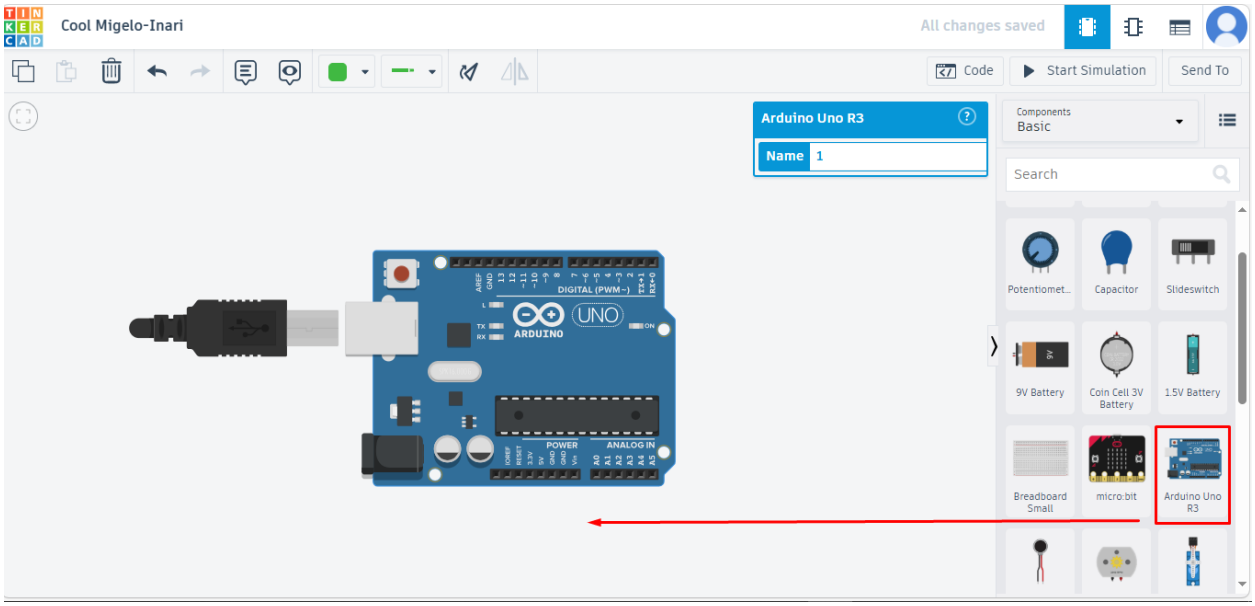
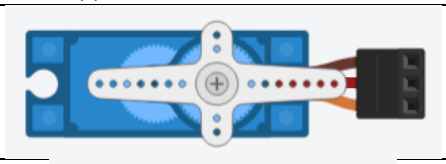
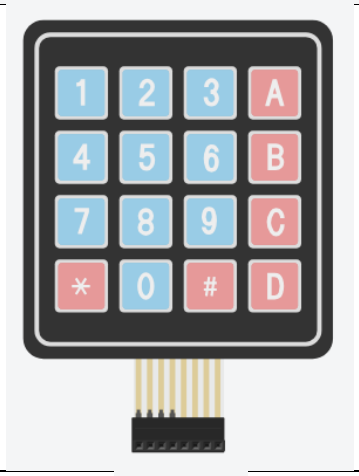
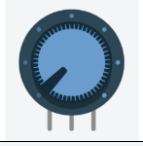
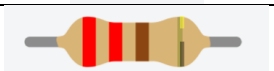
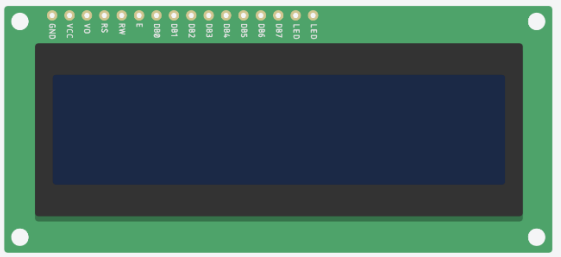
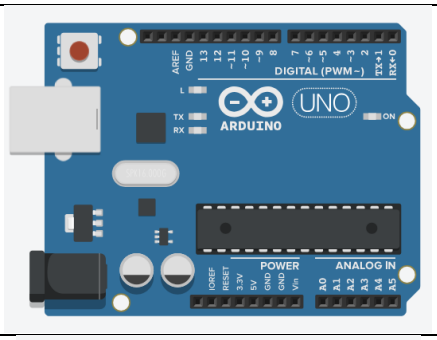
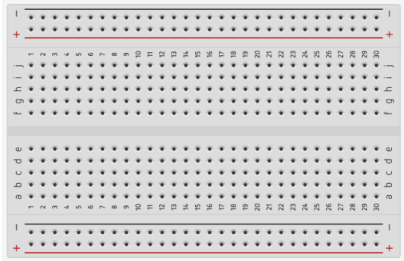
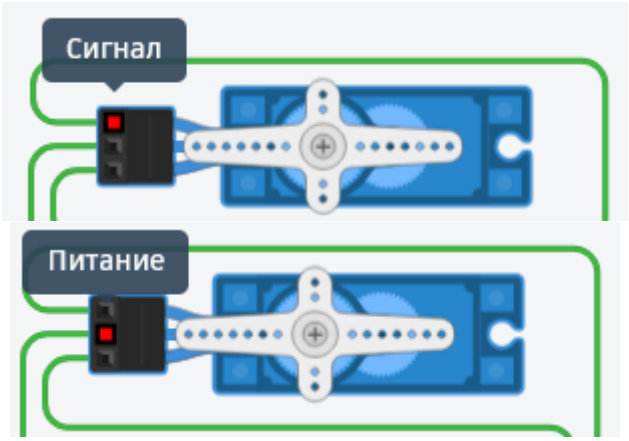


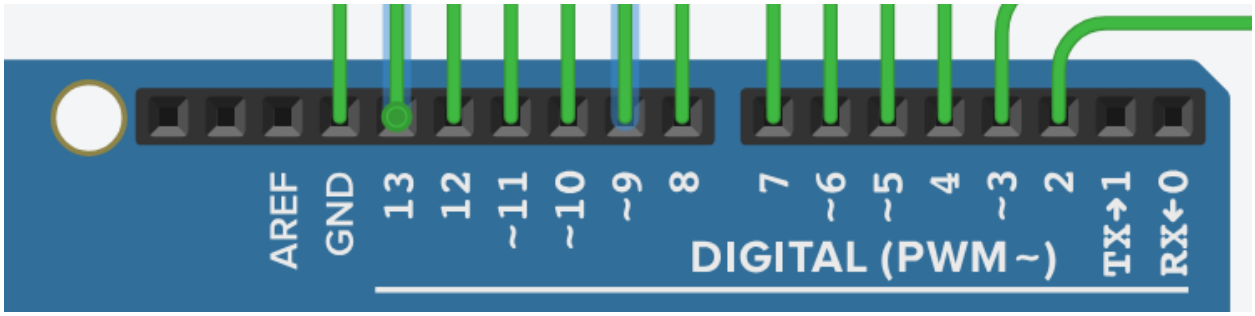
Рисунок 2. Палитра компонентов.

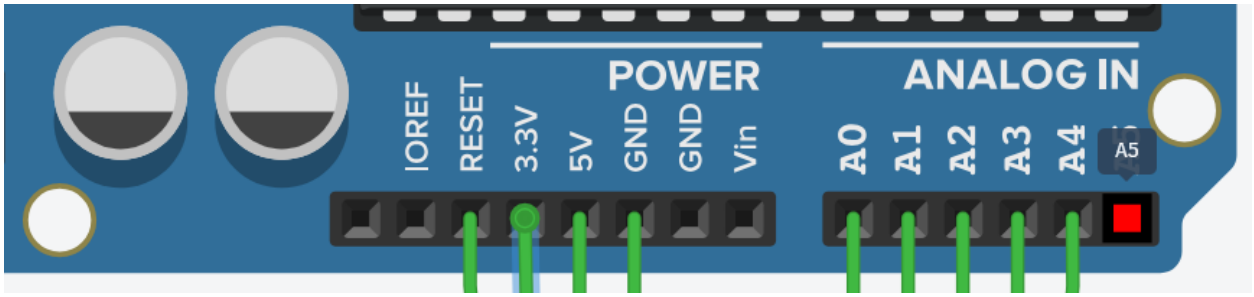
Название и назначение компонента	Внешний вид
Микросервопривод. Маленький мотор, который поворачивается на заданный угол и двигает детали.	
Клавиатура 4 * 4. Панель с 16 кнопками для ввода чисел и команд.	
Потенциометр (переменный резистор). Крутящаяся ручка для изменения значения (например, яркости или громкости).	
Резистор (220 Ом). Ограничивает ток и защищает элементы от повреждения	
ЖК (жидкокристаллический) экран размером 16 х 2. Дисплей для вывода текста (2 строки по 16 символов)	

Arduino UNO. Управляет работой всей схемы и выполняет программу.	
Макетная плата	

Задание 2. Соберите схему в соответствии с примером показанным на рисунке 3 «Электронный дверной замок». Для добавления соединений между компонентами нажимайте на контакте устройства ввода информаци (потенциометр и клавиатура), или устройстве вывода информации (микросервопривод и ЖК-экран) и тяните виртуальный «провод» как показано на рисунке ниже до контакта на плате Arduino UNO.



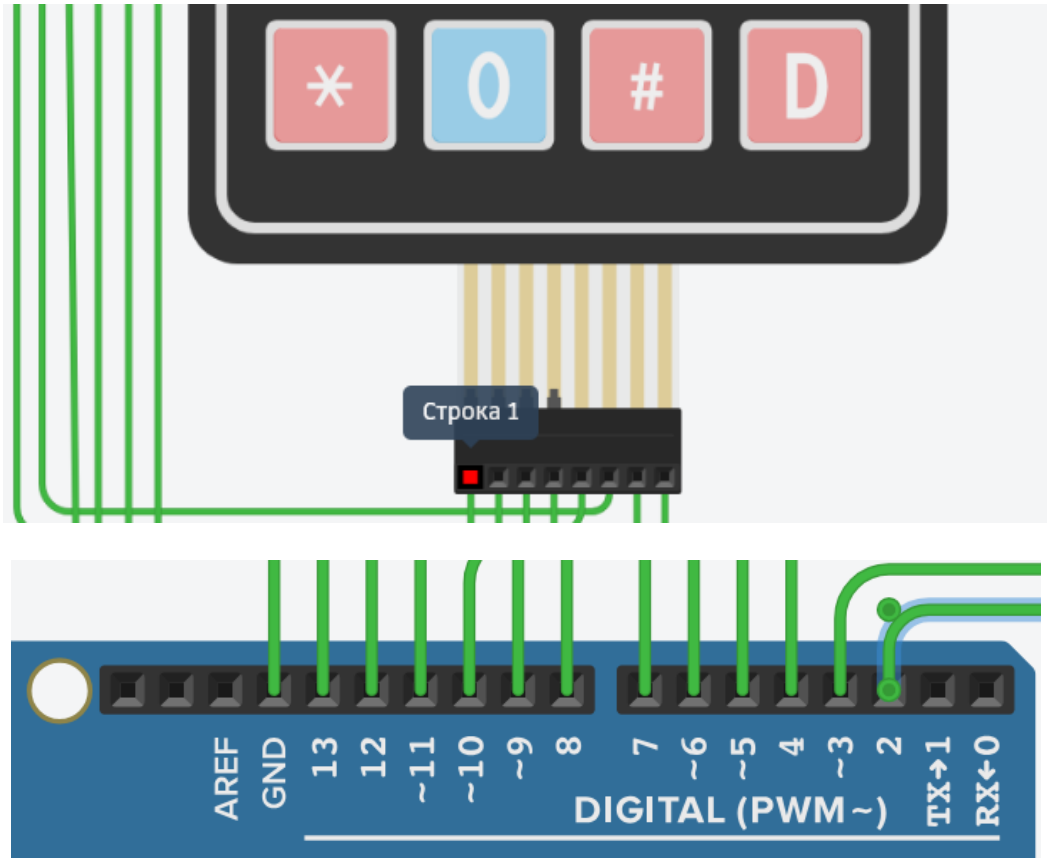




Шаг 1. Подключение микро сервопривода (см. Таблица 1. Подключение 1)

Таблица 1. Подключение 1.

Контакт микро сервопривода	Контакт платы Arduino
Сигнал (на рисунке выше)	Порт 13 (на рисунке выше)
Питание (на рисунке выше)	Порт 3.3V (на рисунке выше)
Земля	GND



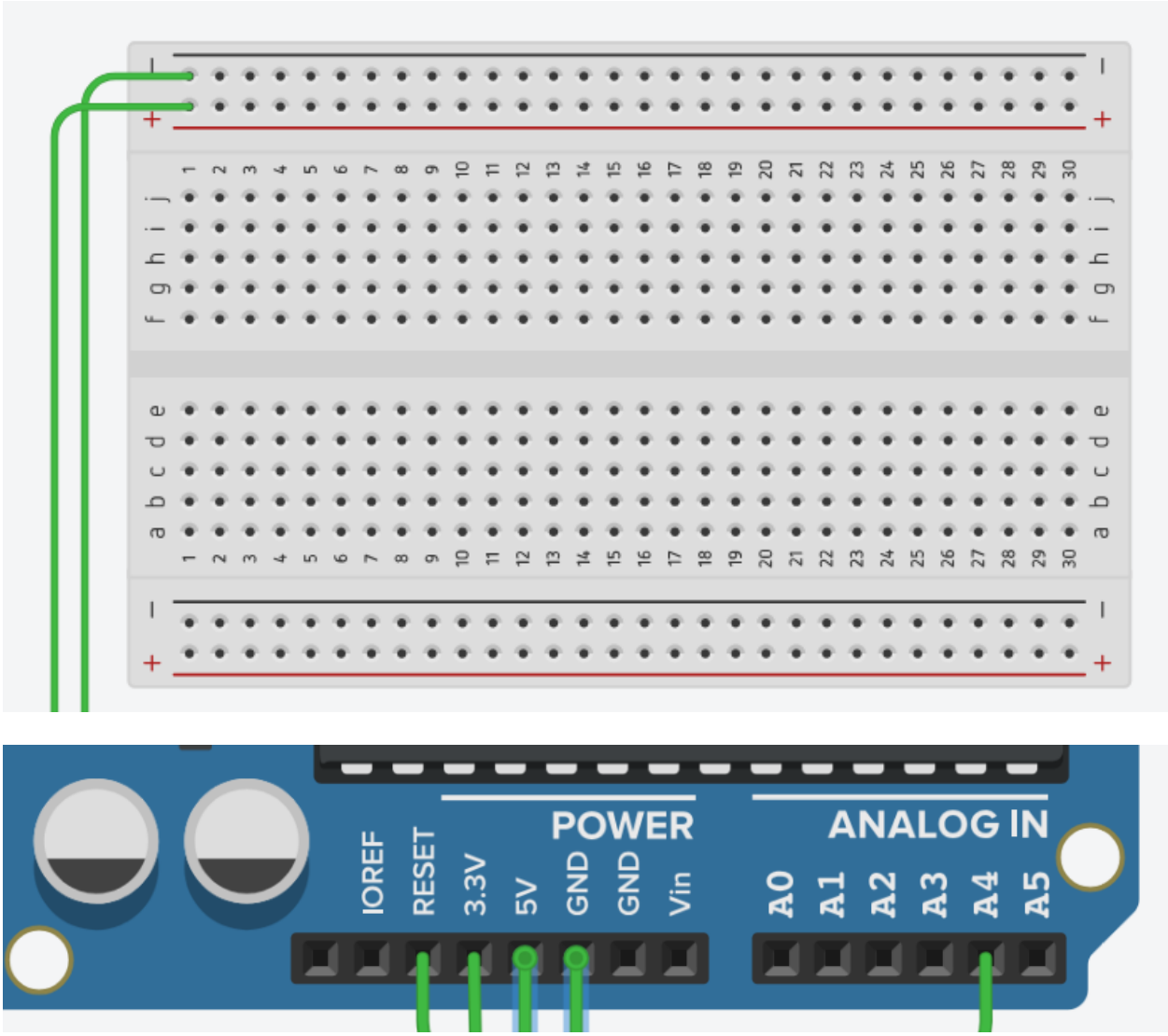
Шаг 2. Подключение клавиатуры 4 * 4 (см. Таблица 2. Подключение 2)

Примечание. Для понимания расположения контактов компонента «Клавиатура 4 * 4» подводите курсор мыши к соответствующим контактам в порядке слева – направо.

Таблица 2. Подключение 2

Контакт клавиатуры 4 * 4	Контакт платы Arduino
Строка 1 (на рисунке выше)	Порт 2 (на рисунке выше)

Строка 2	Порт 3
Строка 3	Порт 4
Строка 4	Порт 5
Столбец 1	Порт 6
Столбец 2	Порт 7
Столбец 3	Порт 8
Столбец 4	Порт 9

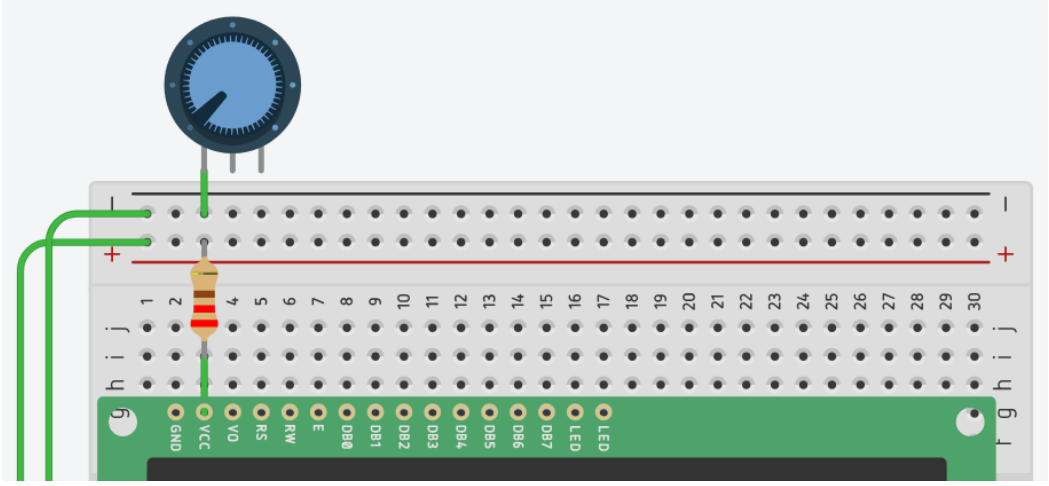


Шаг 3. Подключение макетной платы (см. Таблица 3. Подключение 3)

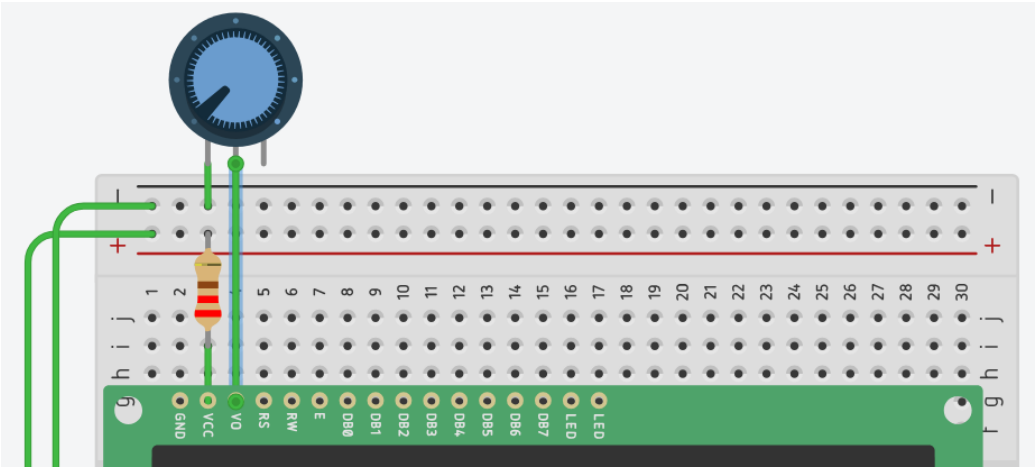
Таблица 3. Подключение 3

Контакт макетной платы	Контакт платы Arduino
Контакт + (на рисунке выше)	Порт 5V (на рисунке выше)
Контакт - (на рисунке выше)	Порт GND (на рисунке выше)

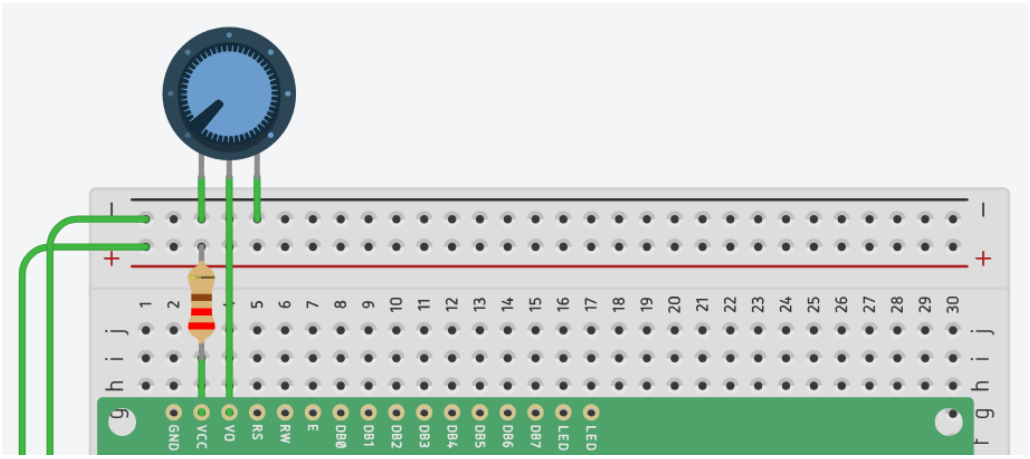
Подробно по ссылке https://youtu.be/DxK85YKvuUU?si=GGwHNQKBCm60YWSm



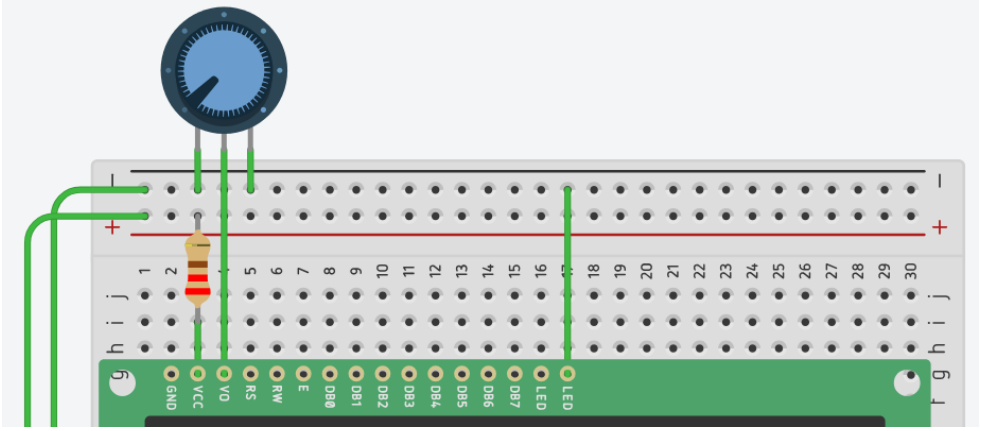
Шаг 4 a. Подключение резисторов и потенциометра



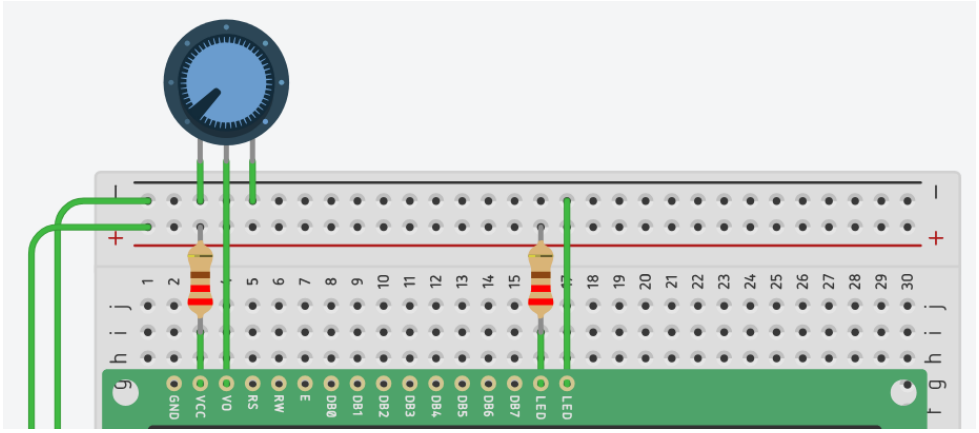
Шаг 4 b. Подключение резисторов и потенциометра



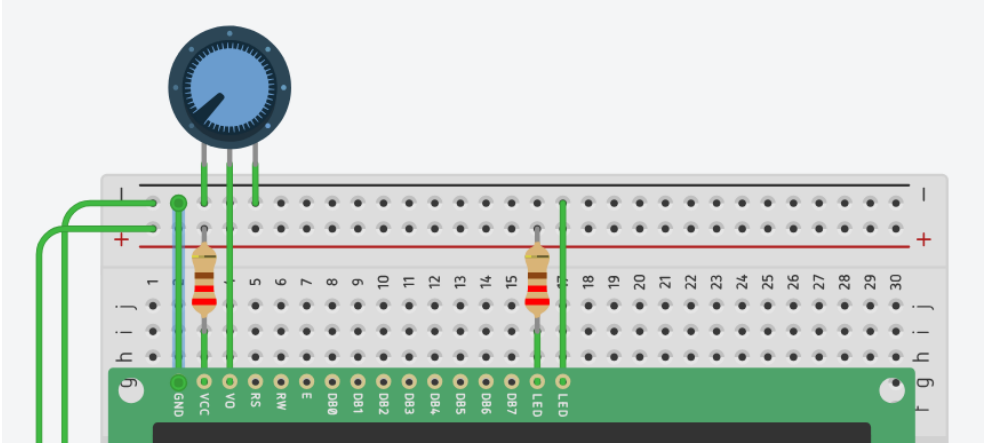
Шаг 4 c. Подключение резисторов и потенциометра



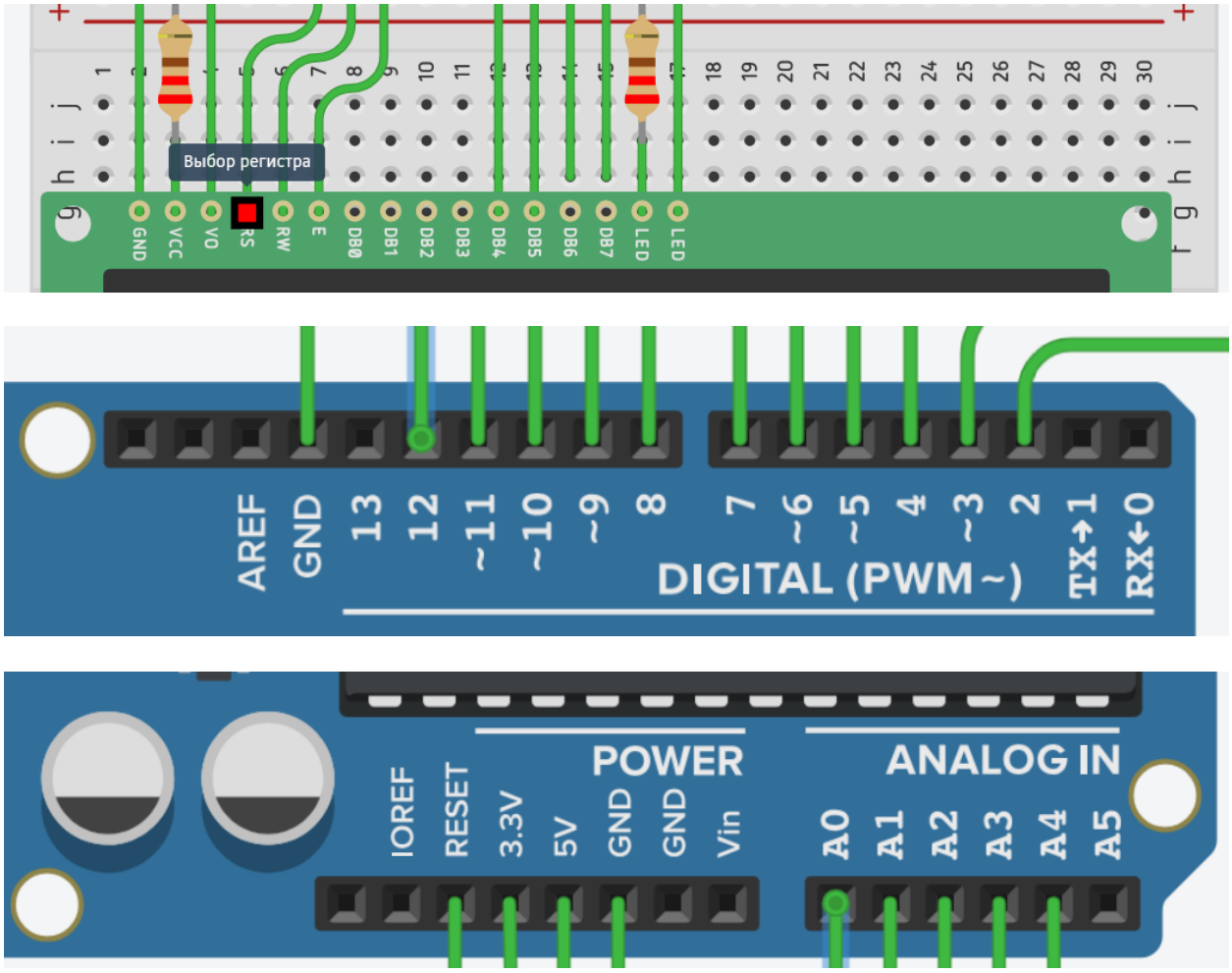
Шаг 4 d. Подключение резисторов и потенциометра



Шаг 4 e. Подключение резисторов и потенциометра



Шаг 4 f. Подключение резисторов и потенциометра

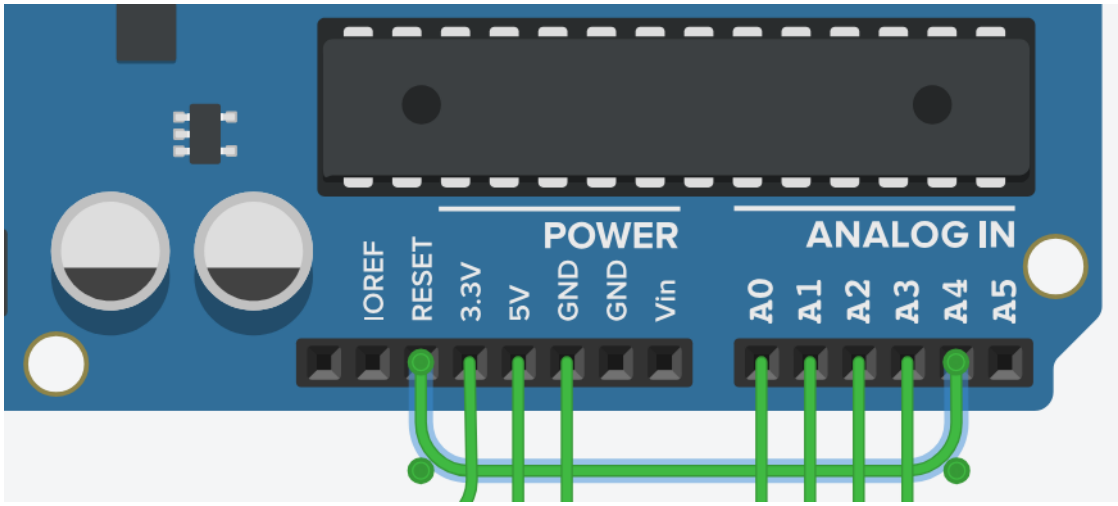


Шаг 5. Подключение ЖК экрана к Arduino (см. Таблица 4. Подключение 4)

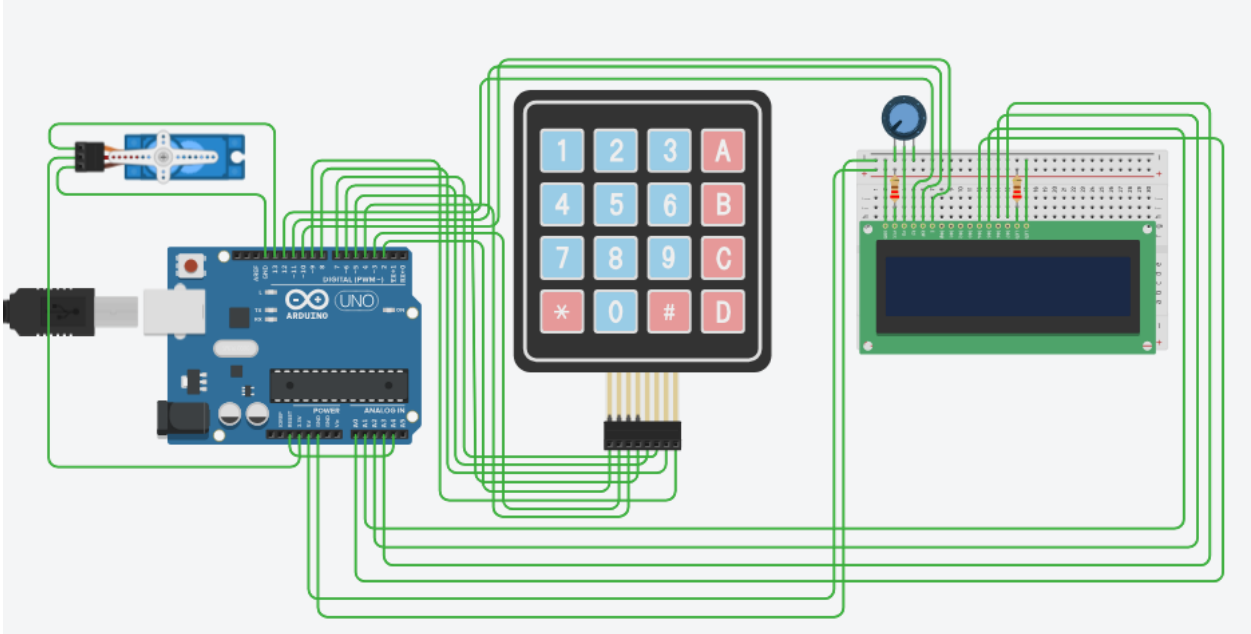
Примечание. Для понимания расположения контактов компонента «ЖК экран 16 * 2» подводите курсор мыши к соответствующим контактам в порядке слева – направо.

Таблица 4. Подключение 4

Контакт ЖК экрана	Контакт платы Arduino
RS (Выбор регистра) (на рисунке выше)	Порт 12 (на рисунке выше)
RW (Чтение/запись)	Порт 11
E (Включение)	Порт 10
DB4	A0
DB5	A1
DB6	A2
DB7	A3



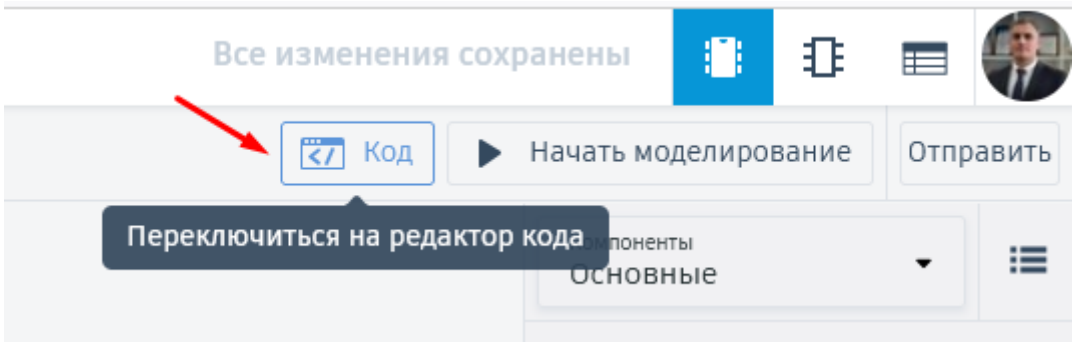
Шаг 6. Подключение аналогового порта A4 к порту Reset



Шаг 7. Полная электронная схема проекта «Электронный дверной замок»
Рисунок 3. «Электронный дверной замок»

Подробно по ссылке https://www.tinkercad.com/things/cMxZ6HEHc7A

Нажмите на кнопке «Код»



Задание 3. Допишите или измените код (текст программы, который следует дописать выделен полужирным) в соответствии с примером ниже.

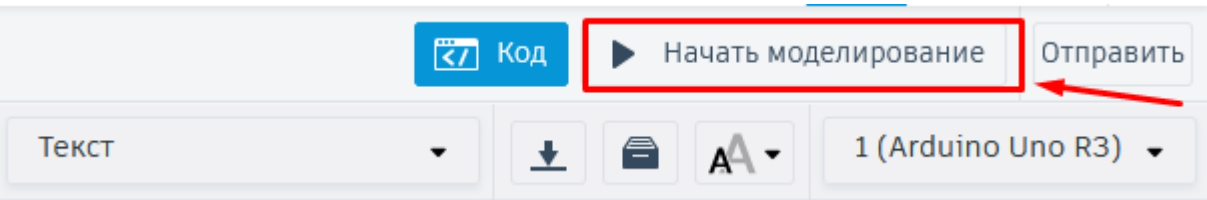
Код проекта «Электронный дверной замок»

```
#include<LiquidCrystal.h>
#include<Keypad.h>
#include<Servo.h>

LiquidCrystal lcd (12,11,10,A0,A1,A2,A3);
Servo myservo;

const byte ROWS=4;
const byte COLS=4;

char keys [ROWS][COLS] {

 {'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}

};

byte rowPins[ROWS]={2,3,4,5};
byte colPins[COLS]={6,7,8,9};

Keypad keypad = Keypad(makeKeymap(keys),rowPins,colPins,ROWS,COLS);

void setup() {

 myservo.attach(13);
 lcd.begin(16,2);
 lcd.print("INPUT PIN CODE");
 delay(1000);
 lcd.print(".");
 delay(800);
 lcd.print(".");
 delay(500);
 lcd.print(".");
 delay(250);
 lcd.clear();
 pinMode(13,OUTPUT);
 pinMode(A4,INPUT);
 Serial.begin(9600);

}


void loop () {

 char key;
 char answer[6]={'2','9','0','6','8','4'};
```

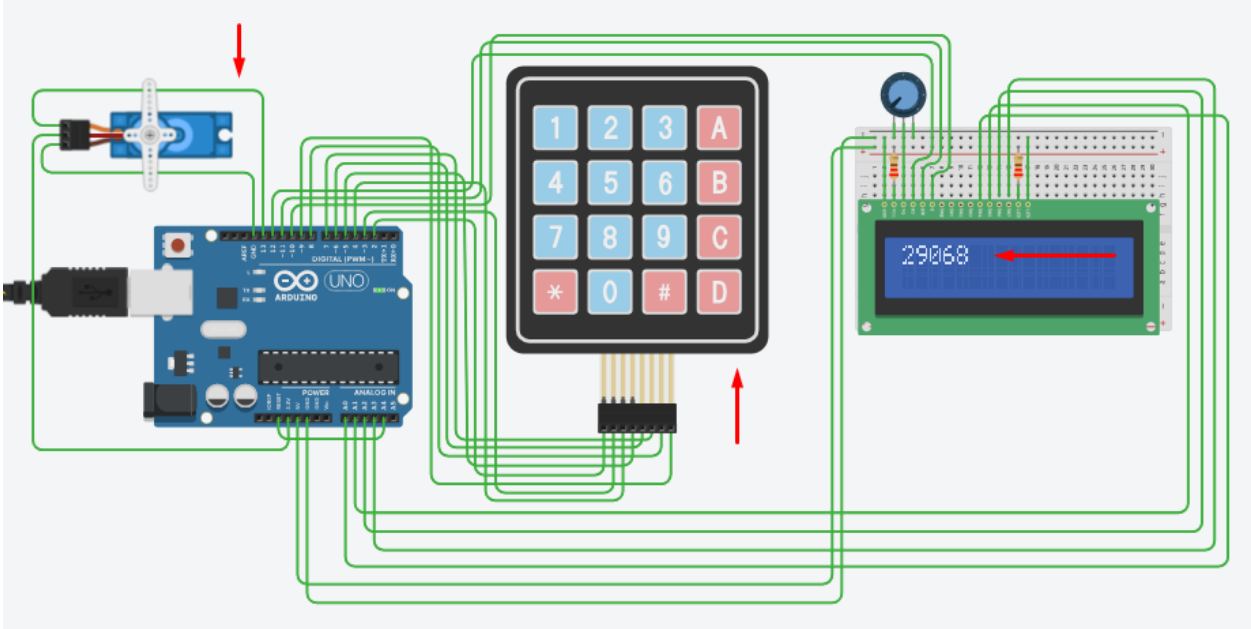
```
  char inputs[6];
  int i=0;

  while (i!=6) {
    key=keypad.getKey();
    if (key) {
      inputs[i] = key;
      lcd.setCursor(i,0);
      lcd.print(inputs[i]);
      i++;
    }
  }

  lcd.clear();

  for (int i=0;i<=5;i++) {
    if (inputs[i]==answer[i]) {
      lcd.setCursor(0,0);
      lcd.print("SUCCESS");
      myservo.write(270);
      delay(3000);
      analogWrite(A4,10);
      break;
    }
    else
    {
      lcd.setCursor(0,0);
      lcd.print("FAIL");\
      analogWrite(A4,10);
      break;
    }

  }

}
```

Объяснение. Объяснение. На схеме показано, как плата Arduino создаёт электронный «кодовый замок»: клавиатура 4×4 подключена к цифровым пинам и нужна для ввода цифр пароля; жидкокристаллический экран (LCD 16×2) подключён к пинам 12, 11, 10 и аналоговым A0–A3 и используется для вывода сообщений (например, «INPUT PIN CODE», «SUCCESS», «FAIL»); сервопривод подключён к пину 13 и выполняет роль замка (поворачивается — «открывает»); потенциометр регулирует яркость/контраст экрана; в программе сначала подключаются библиотеки для экрана, клавиатуры и сервопривода, затем в setup() настраиваются устройства (экран включается и выводит приглашение, сервопривод привязывается к пину, задаются режимы пинов), а в loop() программа ждёт, пока пользователь введёт 6 символов с клавиатуры, показывает их на экране, после чего сравнивает введённый код с правильным (290684): если совпало — выводится «SUCCESS» и сервопривод поворачивается (замок открывается), если нет — выводится «FAIL», и система остаётся закрытой.

Задание 4. Нажмите на кнопке «Начать моделирование»



Нажмите последовательно на компоненте «Клавиатура 4 * 4» следующие символы: 2, 9, 0, 6, 8, 4. Обратите внимание что вводимые символы отображаются на компоненте «ЖК-экран». После ввода последнего символа положение микросервопривода должно измениться на 90 градусов, а на ЖК-экране должна отобразиться надпись «SUCCESS».



Дескрипторы

- Входит в Tinkercad и создаёт проект схемы
- Добавляет компоненты: Arduino Uno, клавиатуру 4×4, LCD 16×2, сервопривод, потенциометр, резистор
- Выполняет подключение клавиатуры к цифровым пинам (2–9) для ввода кода
- Подключает LCD дисплей к пинам 12, 11, 10 и A0–A3 для вывода информации
- Подключает сервопривод к пину 13 для управления «замком»
- Подключает потенциометр для регулировки контрастности экрана
- Открывает редактор и вводит/редактирует программный код
- Подключает библиотеки для работы с LCD, клавиатурой и сервоприводом
- Настраивает устройства в функции setup() (lcd.begin, myservo.attach, pinMode)
- Реализует считывание нажатий клавиш с помощью keypad.getKey()
- Организует ввод PIN-кода из 6 символов и отображение на экране
- Реализует сравнение введённого кода с заданным паролем
- Программирует реакцию системы: вывод «SUCCESS» или «FAIL» на экран
- Реализует управление сервоприводом (поворот при правильном коде)
- Добавляет дополнительные действия (например, сигнал через аналоговый пин)
- Запускает моделирование и тестирует работу схемы (вводит разные коды)
- Анализирует результат работы и при необходимости исправляет ошибки в схеме или коде