

## Урок №12 - Сумо

**1** 1x 2x

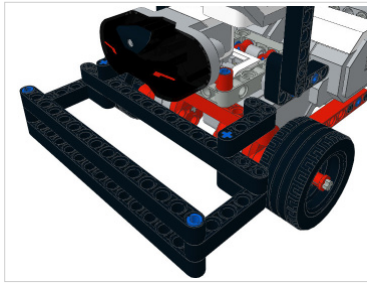
**2** 2x

**3** 4x

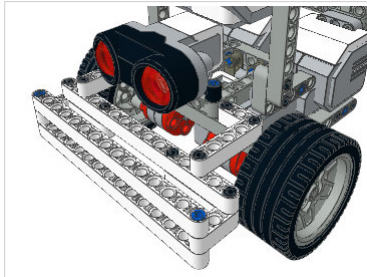
**4** 1x

Получившийся элемент закрепим на передней балке нашего робота.

Lego mindstorms EV3 Home



Lego mindstorms EV3 Education



Наш учебный робот готов. Приступим к созданию программы робота-сумоиста. Замечательно, если у вас есть возможность отлаживать программу, используя ещё одного робота! Если же нет, то ничего страшного: можно задействовать в качестве соперника, например, радиоуправляемую модель автомобиля или те же кегли от "Кегельринга".

### 12.3. Создание программы для соревнования "Сумо".

Первая мысль, которая приходит в голову: использовать программу для "Кегельринга", внеся в неё косметические изменения. Действительно, алгоритмы поведения робота в "Кегельринге" и в "Сумо" очень похожи. Они реализуют поиск объекта и выталкивание его за пределы поля. Можно загрузить в робота-сумоиста программу для "Кегельринга", но работать такой сумоист будет не очень эффективно. Тем не менее, знания, полученные на предыдущем уроке, пригодятся нам сейчас.

Настало время загрузить в среду программирования наш проект "lessons-2", создать в нём новую программу "lesson-12" и подключить робота к среде программирования.

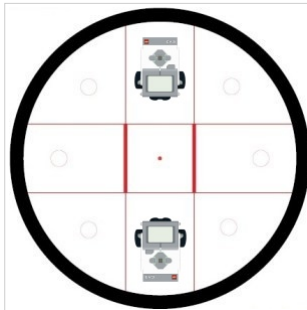
Поведенческую модель робота-сумоиста можно условно разделить на две части: **поиск соперника** и **атака соперника**. Сначала займемся реализацией первой части - **поиска соперника**.

Подробно пропишем последовательность действий нашего робота при обнаружении соперника на поле:

1. вращаться вокруг своей оси, пока впереди расположенный датчик не обнаружит соперника;
2. остановиться напротив соперника.

Эта последовательность действий полностью повторяет алгоритм поиска роботом кегли в "Кегельринге", но, так как, расстояние между роботами в "Сумо" может превышать расстояние от робота до кегли, то нам необходимо выбрать другое пороговое значение для используемого датчика.

Установим соперников на поле напротив друг друга, как показано на рисунке ниже.



Такое положение практически соответствует максимальному удалению роботов друг от друга во время состязания, поэтому текущее показание датчика, измеряющего расстояние до соперника можно взять за пороговое. **Важно:** так как пороговое значение будет достаточно большим - необходимо чтобы за пределами поля на расстоянии около 1 м. во время работы робота также отсутствовали посторонние предметы, способные помешать поиску.

На "Странице аппаратных средств", находящейся в правом нижнем углу среды программирования, выберем вкладку "Представление порта" (Рис. 1, 2 поз. 1) и снимем показание датчика, определяющего расстояние до соперника, установив соответствующий режим отображения показаний.

В нашем случае ультразвуковой датчик в режиме "Расстояние в сантиметрах" показывает значение - 56,1 (Рис. 1 поз. 2). За пороговое значение примем число - 57.

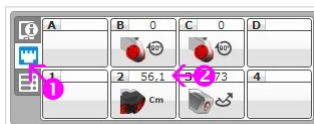


Рис. 1

Инфракрасный датчик в режиме "Приближение" показывает значение - 68 (Рис. 2 поз. 2). За пороговое значение примем число - 70.



Рис. 2

По аналогии с "Кегельрингом" мы можем запрограммировать нахождение роботом соперника, только, чтобы немного дистанцироваться от предыдущего урока, изменим направление вращения робота на противоположное:

#### Ультразвуковой датчик

1. Для того, чтобы заставить робота вращаться вокруг своей оси, воспользуемся программным блоком "Независимое управление моторами" "Зеленой палитры", Режим работы блока установим "Включить", значение мощности для порта "В" установим равным -30, значение мощности для порта "С" установим равным 30 (Рис. 3 поз.1).
2. Для поиска соперника используем программный блок "Ожидание" "Оранжевой палитры" в режиме "Ультразвуковой датчик - Сравнение - Расстояние в сантиметрах" с пороговым значением срабатывания датчика, равным 57 (Рис. 3 поз. 2).

- После того, как робот окажется напротив соперника, используя программный блок "Независимое управление моторами" "Зеленой палитры" выключим моторы (Рис. 3 поз. 3).



Рис. 3

#### Инфракрасный датчик

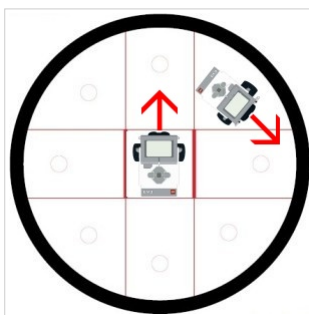
- Для того, чтобы заставить робота вращаться вокруг своей оси, воспользуемся программным блоком "Независимое управление моторами" "Зеленой палитры". Режим работы блока установим "Включить", значение мощности для порта "В" установим равным -30, значение мощности для порта "С" установим равным 30 (Рис. 4 поз.1).
- Для поиска соперника воспользуемся программным блоком "Ожидание" "Оранжевой палитры" в режиме "Инфракрасный датчик - Сравнение - Приближение", с пороговым значением срабатывания датчика, равным 70 (Рис. 4 поз. 2).
- После того, как робот окажется напротив соперника, используя программный блок "Независимое управление моторами" "Зеленой палитры" выключим моторы (Рис. 4 поз. 3).



Рис. 4

На этапе отладки этого алгоритма вам придется, подбирая значения "Мощность" моторов "В" и "С" а также пороговое значение датчика, добиться от вашего робота точного обнаружения и остановки строго напротив соперника. Только после этого можно будет переходить к программной реализации алгоритма атаки.

Если поиск соперника в "Сумо" очень похож на поиск кегли в "Кегельринге", то выталкивание соперника имеет важное отличие! Начиная атаку, первое, что необходимо сделать, это прямолинейно устремиться на максимальной мощности моторов в сторону обнаруженного соперника, проверяя датчиком цвета обнаружение границы ринга. Но ведь наш соперник тоже может двигаться! Поэтому вполне возможна ситуация, когда соперник выйдет в сторону из-под направления нашей атаки. В этом случае, наш робот, промахнувшись, будет двигаться в сторону границы ринга, теряя соперника и драгоценное время.



Следовательно, нам необходимо во время прямолинейного движения вперед анализировать оба датчика и прекращать атаку в случае, если робот **потеряет соперника** ИЛИ **робот достигнет границы ринга**. Поэтому нам необходимо отказаться от использования программного блока "Ожидание" "Оранжевой палитры" и самостоятельно в цикле получать и обрабатывать показания двух датчиков.

Приступим к поэтапной реализации алгоритма атаки соперника: для этого создадим в проекте временную программу "lesson-12-1" и начнем её наполнение программными блоками.

- Возьмем программный блок "Цикл" "Оранжевой палитры".
- Внутри блока "Цикл" поместим программный блок "Независимое управление моторами" "Зеленой палитры". Режим работы блока установим в значение "Включить" (Рис. 5 поз. 1), мощности моторов "В" и "С" установим в максимальное значение - 100 (Рис. 5 поз. 2).

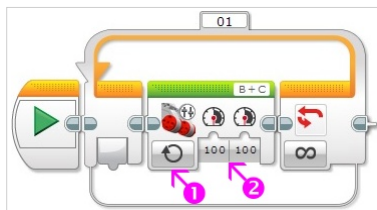


Рис. 5

- Следом за блоком "Независимое управление моторами" поместим программный блок "Датчик цвета" "Желтой палитры". Режим работы блока установим в значение "Сравнение - Яркость отраженного света" (Рис. 6)

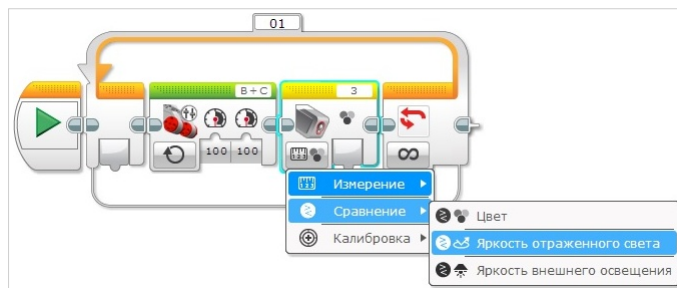


Рис. 6

В этом режиме программный блок "Датчик цвета" "Желтой палитры" визуально очень похож на программный блок "Ожидание" "Оранжевой палитры" в режиме "Датчик цвета - Сравнение - Яркость отраженного света". Но, в отличие от блока "Ожидание", этот программный блок не ждет выполнения условия, указанного параметрами "Тип сравнения" (Рис. 7 поз. 1) и "Пороговое значение" (Рис. 7 поз. 2), а сразу выдает логическое значение ("Истина" или "Ложь") в выходном параметре "Результат сравнения" (Рис. 7 поз. 3) и измеренное значение - в выходном параметре "Освещение" (Рис. 7 поз. 4).

Параметры "Тип сравнения" и "Пороговое значение" на Рис. 7 поз. 1, 2 установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 7 поз. 3) выдавал логическое значение "Истина" при пересечении датчиком цвета черной границы ринга.

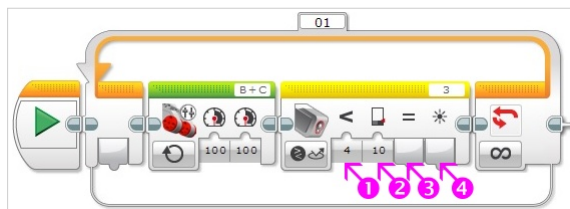


Рис. 7

4. В случае использования ультразвукового датчика за блоком "Датчик цвета" установим программный блок "Ультразвуковой датчик" "Желтой палитры". Режим работы блока установим в значение "Сравнение - Расстояние в сантиметрах" (Рис. 8 поз. 1). Параметр "Тип сравнения" (Рис. 8 поз. 2), параметр "Пороговое значение" (Рис. 8 поз. 3) установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 8 поз. 4) выдавал логическое значение "Истина" в случае потери из виду роботом соперника.

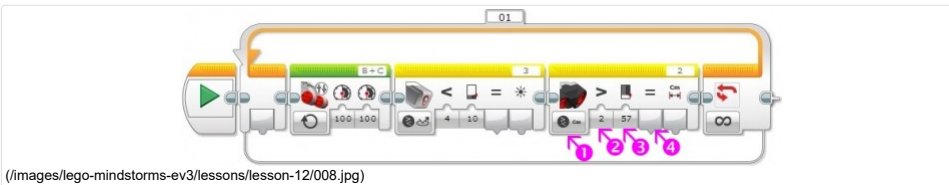


Рис. 8

В случае использования инфракрасного датчика за блоком "Датчик цвета" установим программный блок "Инфракрасный датчик" "Желтой палитры". Режим работы блока установим в значение "Сравнение - Приближение" (Рис. 9 поз. 1). Параметр "Тип сравнения" (Рис. 9 поз. 2), параметр "Пороговое значение" (Рис. 9 поз. 3) установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 9 поз. 4) выдавал логическое значение "Истина" в случае потери из виду роботом соперника.



Рис. 9

Давайте ещё раз проанализируем промежуточный код нашего алгоритма атаки: мы включили моторы на максимальную мощность и движемся вперед, постоянно в цикле опрашивая датчики. Если наш робот пересечет черную линию границы ринга, то значение выходного параметра "Результат сравнения" "Датчика цвета" примет значение "Истина". Если наш робот потеряет соперника, то значение выходного параметра "Результат сравнения" датчика, следящего за соперником, также примет значение "Истина". В любом из этих случаев нам следует прекратить атаку, завершив наш цикл. В этом нам поможет программный блок "Логические операции" "Красной палитры". Познакомимся с этим блоком подробнее: программный блок "Логические операции" предназначен для выполнения операций над логическими данными (Рис. 10).

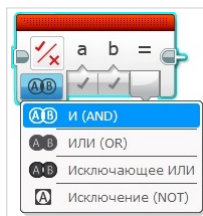


Рис. 10

Выбранный режим программного блока "Логические операции" "Красной палитры" определяет одну из четырех операций над логическими данными: "И (AND)", "ИЛИ (OR)", "Исключающее ИЛИ" и "Исключение (NOT)". Два входных параметра "a" и "b" (для операции "Исключение (NOT)" - один входной параметр "a") передают в программный блок входные значения, а результирующее значение выдается выходным параметром "Результат". Если вы ранее не сталкивались с логическими операциями, то можете ознакомиться с базовыми знаниями в прилагаемой справке под спойлером.

#### Логические операции

5. За программным блоком "Ультразвуковой датчик" или "Инфракрасный датчик" поместим программный блок "Логические операции" "Красной палитры".

- Выходной параметр "Результат сравнения" программного блока "Датчик цвета" (Рис. 11, 12 поз. 1) соединим с входным параметром "a" программного блока "Логические операции" (Рис. 11, 12 поз. 4).
- Выходной параметр "Результат сравнения" программного блока "Ультразвуковой (инфракрасный) датчик" (Рис. 11, 12 поз. 2) соединим с входным параметром "b" программного блока "Логические операции" (Рис. 11, 12 поз. 5).
- Режим работы программного блока "Логические операции" установим в "ИЛИ (OR)" (Рис. 11, 12 поз. 3). В этом случае результат выполнения логической операции будет принимать значение "Истина", только если будет выполнено одно из условий: датчик цвета пересек черную линию, робот потерял соперника.
- Установив режим программного блока "Цикл" в значение "Логическое значение" (Рис. 11, 12 поз. 7), выходной параметр "Результат" программного блока "Логические операции" (Рис. 11, 12 поз. 6) соединим с входным параметром "Пока не будет истина" программного блока "Цикл" (Рис. 11, 12 поз. 8). Данные настройки завершат выполнение цикла при "Истинном" результате выполнения логической операции.

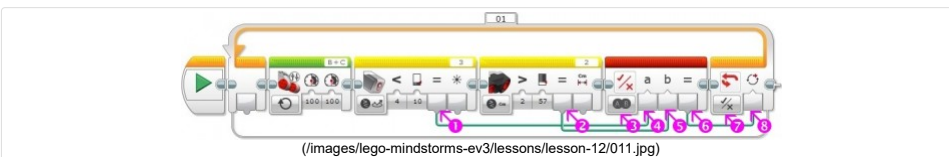


Рис. 11

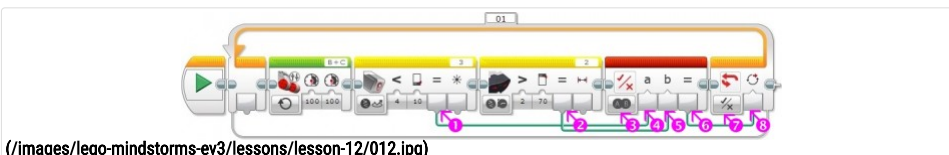


Рис. 12

Давайте протестируем получившийся алгоритм атаки! Для этого поместим нашего робота внутрь ринга, напротив установим неподвижного соперника и запустим программу атаки на выполнение. Наш робот должен уверенно вытолкать соперника за пределы ринга и остановиться над черной границей поля. Получилось? Значит наш сумоист верно контролирует границу ринга.

Проведем второй эксперимент: снова установим напротив робота неподвижного соперника и запустим программу атаки. Когда наш робот устремится к сопернику и приблизится достаточно близко, резко уберём соперника в сторону. Наш робот должен, потеряв соперника, остановиться.

Подведем итог: мы реализовали алгоритм поиска соперника и успешно его протестировали, также прошел проверку алгоритм атаки.

Законченная программа сумоиста должна в бесконечном цикле выполнять последовательно поиск соперника, а затем - атаку соперника. Можно было бы уже объединить обе части нашей программы, если бы не одно маленькое дополнение. Если наш робот остановился над границей ринга, то перед тем, как начать поиск, роботу следует, отъехав немного назад, вернуться внутрь ринга. Дополним нашу программу атаки следующим кодом: за пределами цикла атаки, воспользуемся программным блоком "Переключатель" "Оранжевой палитры". Режим работы блока "Переключатель" установим в "Датчик цвета - Сравнение - Яркость отраженного света". Параметры "Тип сравнения" и "Пороговое значение" установим аналогично ранее используемому в программном блоке "Датчик цвета" "Желтой палитры". Следовательно, если наш робот остановился над черной линией, то выполнение будет передано верхнему контейнеру программного блока "Переключатель". Именно в верхний контейнер поместим программный блок "Рулевое управление" "Зеленой палитры", с настройками параметров, заставляющими робота отъехать назад на один оборот моторов. В нижний контейнер программного блока "Переключатель" поместим программный блок, выключающий моторы (Рис. 13). Повторно протестировав алгоритм атаки, убедимся, что после того, как робот-сумоист вытолкнул соперника за пределы ринга, он вернулся немного назад.



Логин

Пароль

Секретный ключ

☐ Запомнить меня

Войти

Регистрация ➤ (/component/users/?view=registration&Itemid=101)  
Забыли логин? (/component/users/?view=remind&Itemid=101)  
Забыли пароль? (/component/users/?view=reset&Itemid=101)

Популярные статьи

- Урок №2 - Программирование движения робота (/lessons/lesson-2.html)
- Урок №12 - Сумо (/lessons-2/lesson-12.html)
- Урок №11 - Кергельринг (/lessons-2/lesson-11.html)
- Урок №1 - Знакомство с конструктором (/lessons/lesson-1.html)
- Урок №5 - Начинаем изучать датчик цвета (/lessons/lesson-5.html)

Последние комментарии

01 июля 2019

- Урок №7 - Изучаем ультразвуковой датчик (https://robot-help.ru/lessons/lesson-7.html#comment-586)  
...
- Урок №11 - Кергельринг (https://robot-help.ru/lessons-2/lesson-11.html#comment-581)  
Пётр007...
- Урок №11 - Кергельринг (https://robot-help.ru/lessons-2/lesson-11.html#comment-579)  
ВОНА...
- Урок №2 - Программирование движения робота (https://robot-help.ru/lessons/lesson-2.html#comment-574)  
...
- Урок №12 - Сумо (https://robot-help.ru/lessons-2/lesson-12.html#comment-564)  
Нурик...
- Урок №11 - Кергельринг (https://robot-help.ru/lessons-2/lesson-11.html#comment-563)  
Олег...
- Урок №11 - Кергельринг (https://robot-help.ru/lessons-2/lesson-11.html#comment-562)  
Тихон...
- Урок №12 - Сумо (https://robot-help.ru/lessons-2/lesson-12.html#comment-561)  
Олег...
- Урок №2 - Программирование движения робота (https://robot-help.ru/lessons/lesson-2.html#comment-560)  
Саммер...
- Урок №2 - Программирование движения робота (https://robot-help.ru/lessons/lesson-2.html#comment-559)  
sda...

RSS Лента



(/index.php?option=com\_content&view=category&id=20&format=feed&type=rss)



(/index2.php?option=com\_jcomments&task=rss\_full&no\_html=1)



(https://beget.ru/?id=20586)