



# Занятие по робототехнике: «Сумо»

Автор: Тимофеев Алексей Сергеевич,  
педагог дополнительного образования



# Введение

---

Цели занятия: познакомиться с одним из видов соревнований по робототехнике

# Задачи

---

- Привлечение детей к инновационному, научно-техническому творчеству в области робототехники;
- Формирование новых знаний, умений и компетенций в области механики и программирования;
- Развитие алгоритмического мышления.

# Оборудование

---

- 1. компьютер**
- 2. наборы Lego Mindstorms EV3**
- 3. ноутбуки среда программирования**
- 4. поля для заездов**
- 5. секундомер.**

# Сумо

---

Поведение нашего робота в «Сумо» будет похоже на поведение в «Кегельринге»: роботу также нужно найти объект внутри круга и вытолкнуть его. Только теперь вместо кегли мы будем выталкивать другого робота. В чем будет главное отличие?

Противник, в отличие от кеглей в «Кегельринге», тоже будет пытаться вас вытолкнуть.

# Сумо

---

Будет ли отличаться робот для «Сумо» от робота для «Кегельринга»?

Для поиска противника мы по-прежнему будем использовать датчик расстояния, а определять границу поля датчиком цвета.

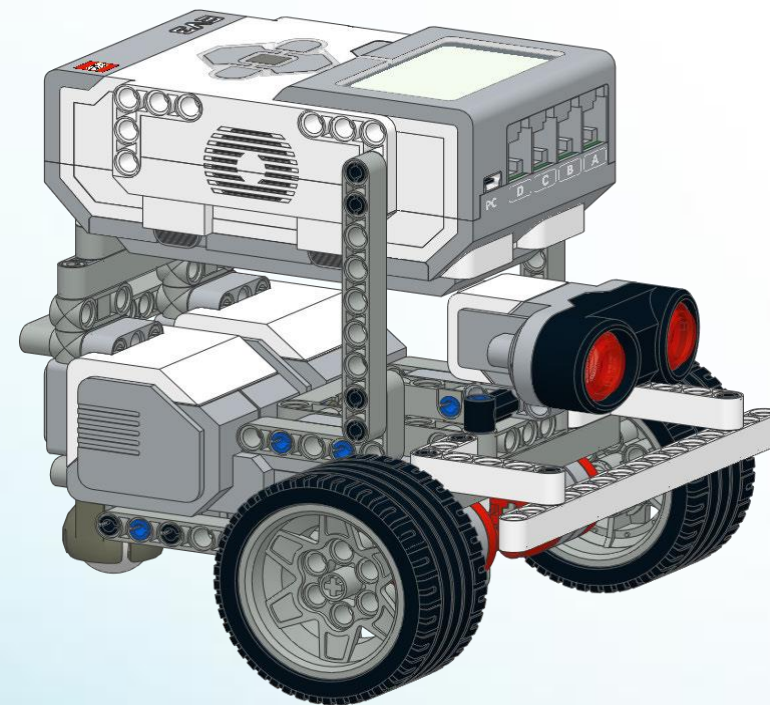


# Сумо

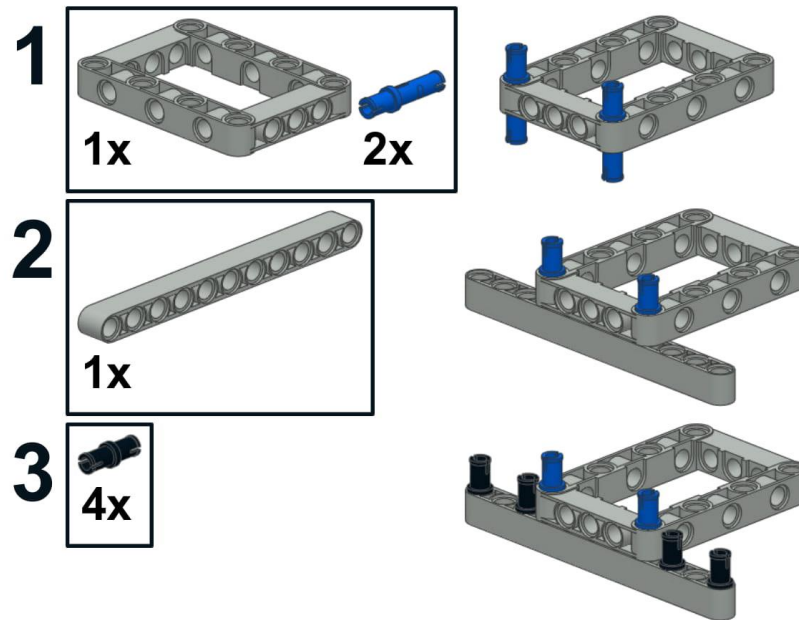
---

Давайте соберем такого робота, который имеет датчик цвета и расстояния.

Также не забываем, что он должен уметь вращаться вокруг своей оси для поиска врага.

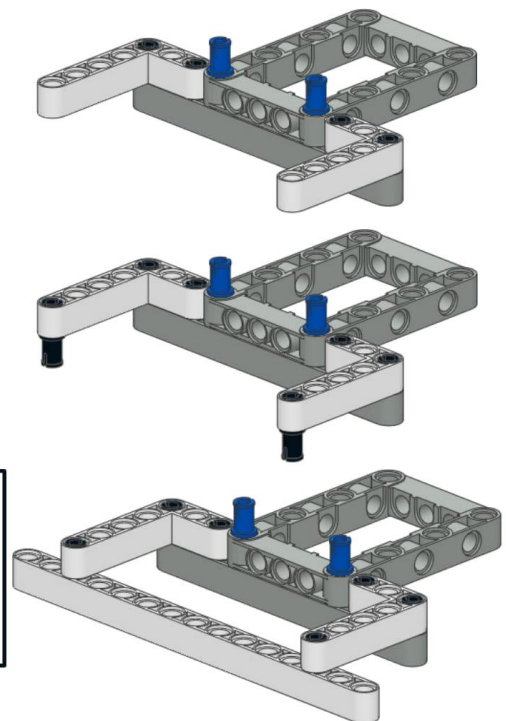
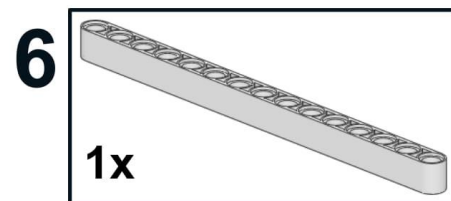
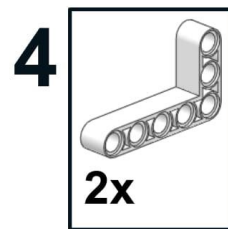


# Сумо



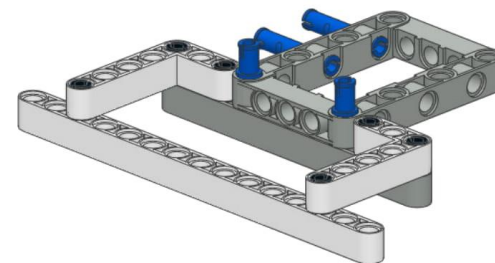


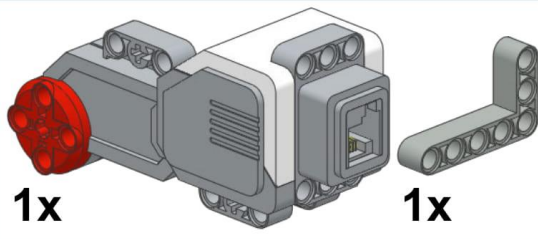
# Сумо

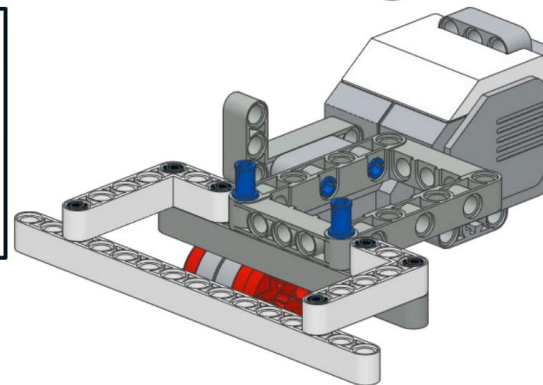


# Сумо

7   
2x

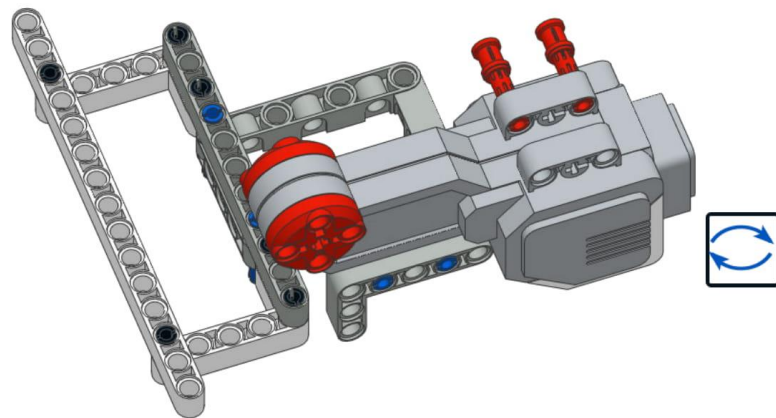


8   
1x 1x

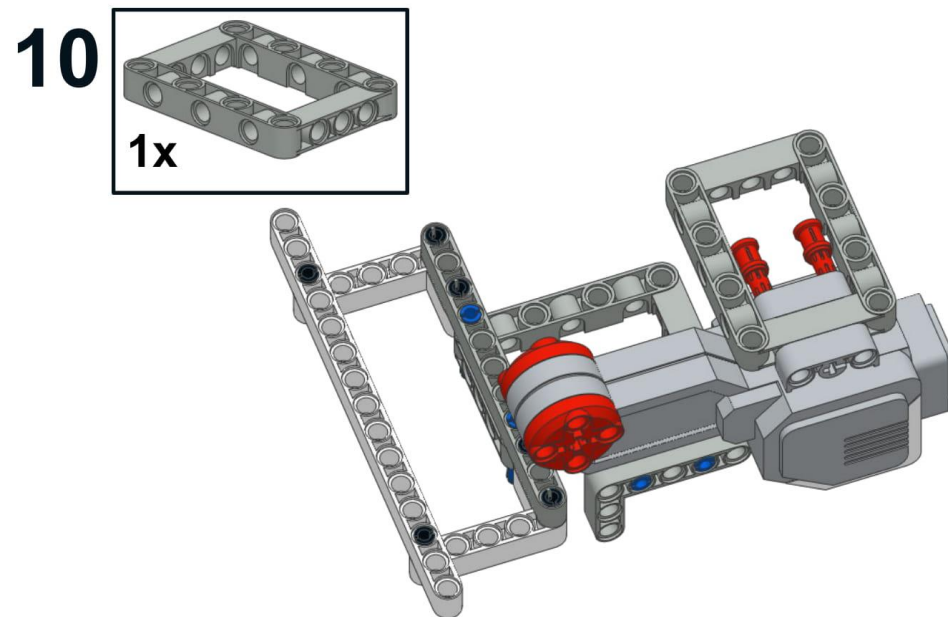


# Сумо

9  
2x

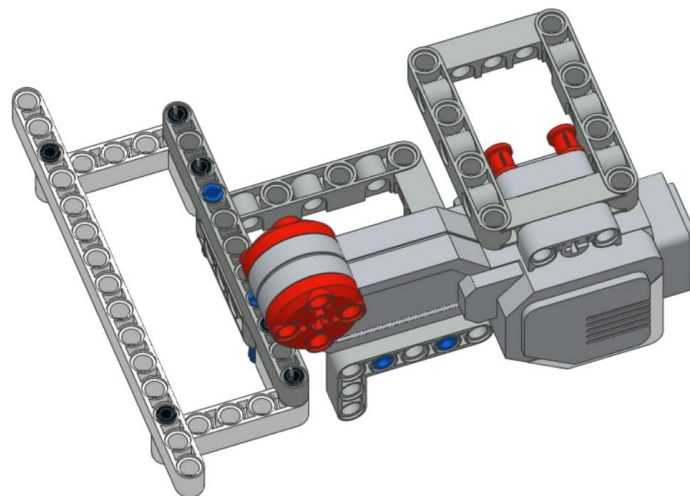


# Сумо

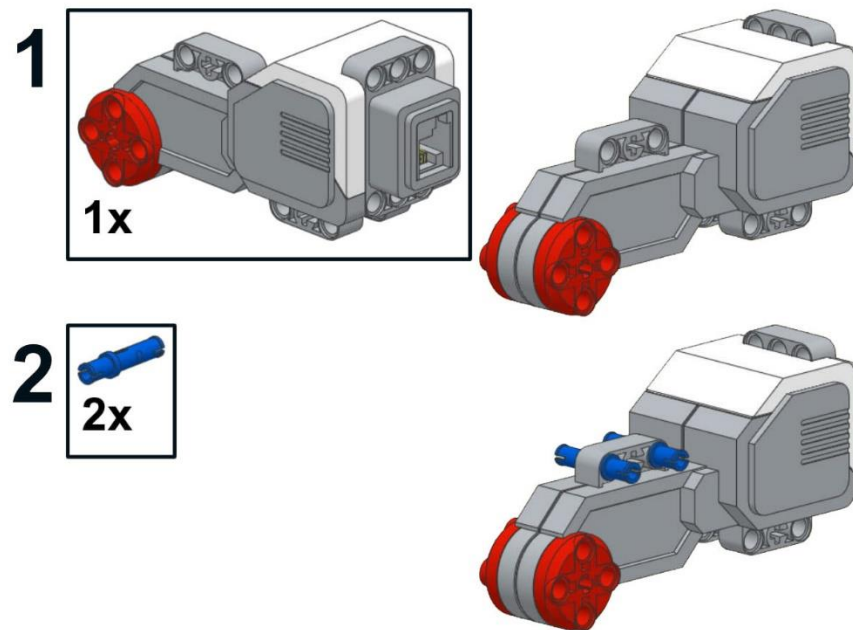


# Сумо

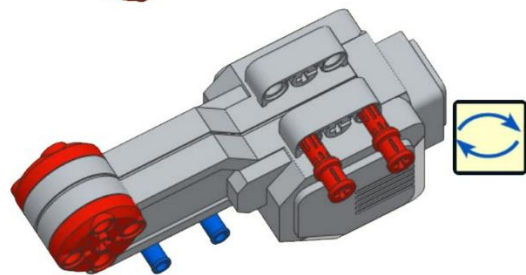
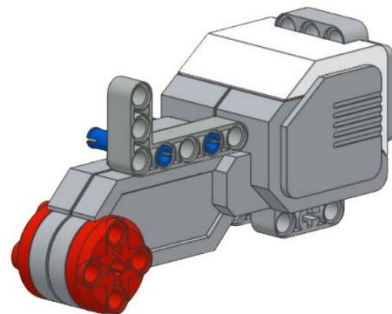
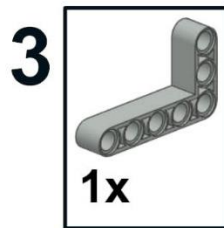
11



# Сумо

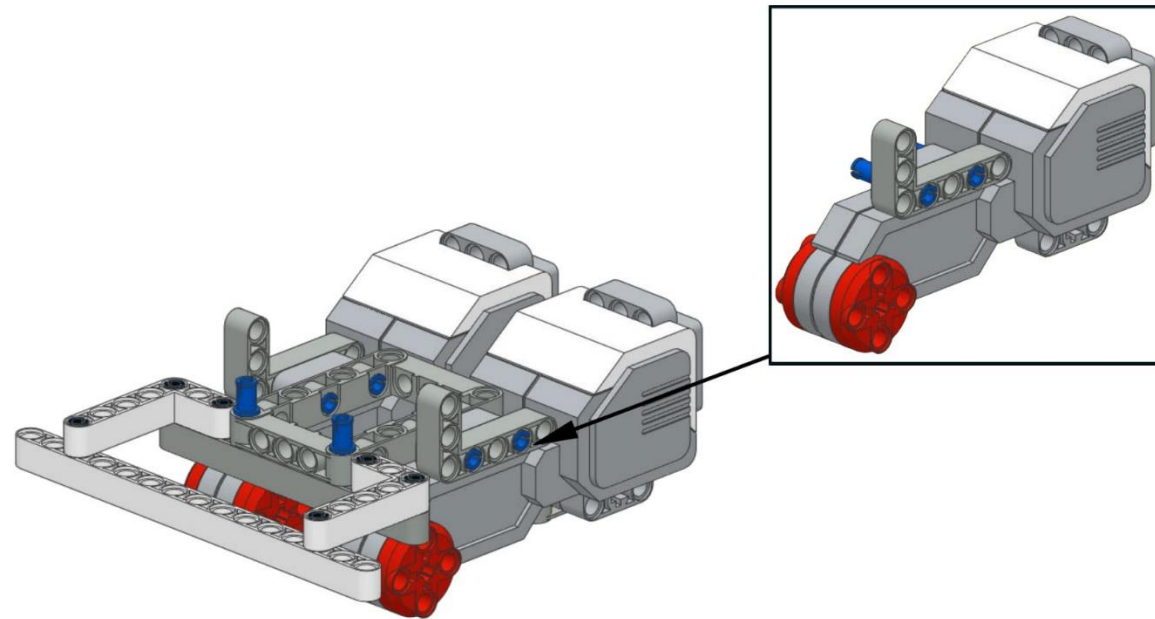


# Сумо



# Сумо

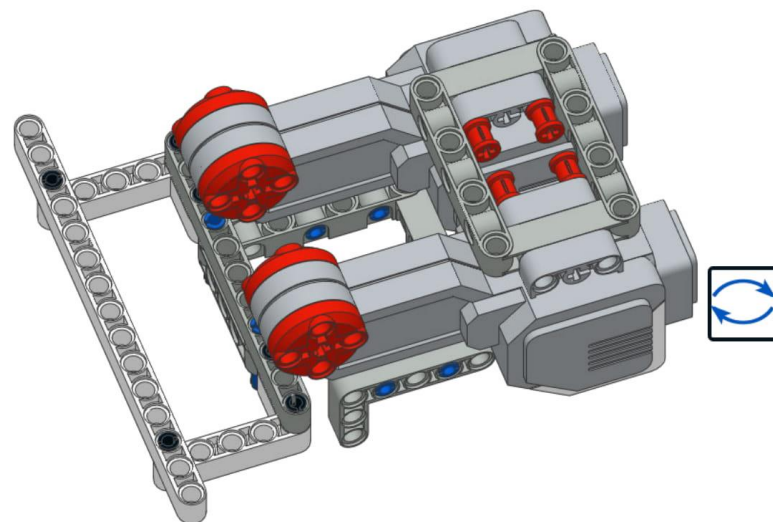
12



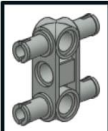


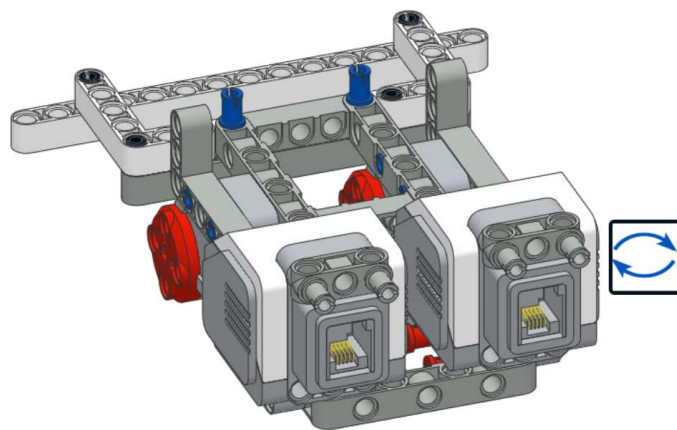
# Сумо

13



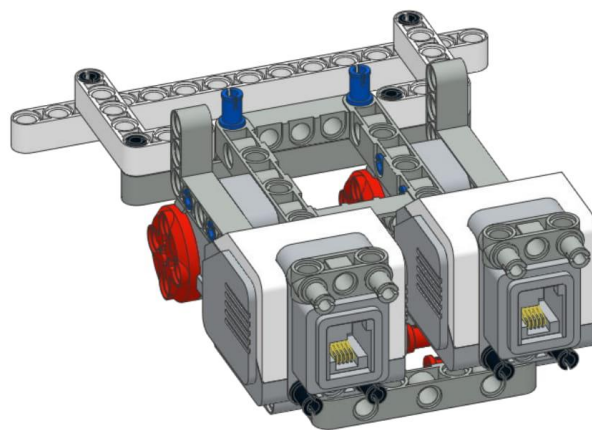
# Сумо

14   
2x



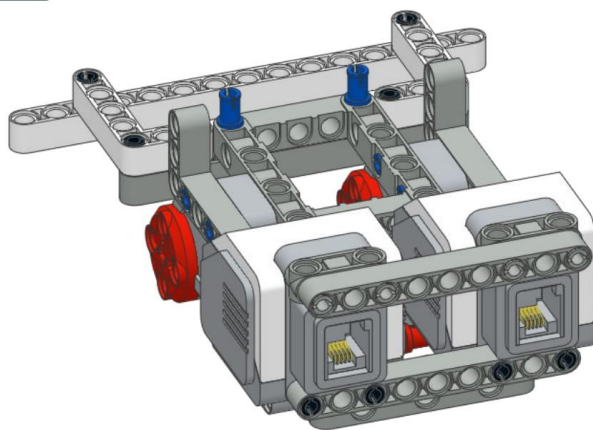
# Сумо

15   
4x



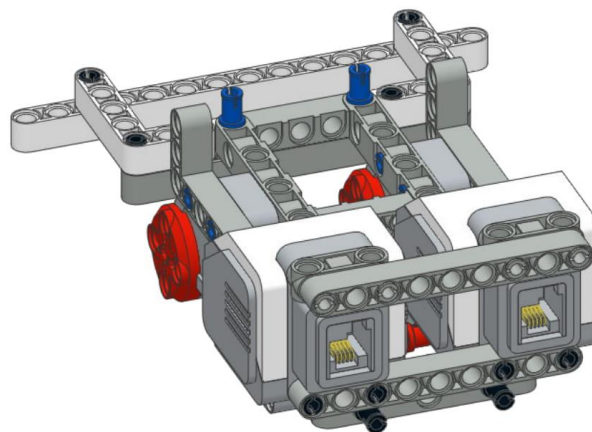
# Сумо

16  
2x



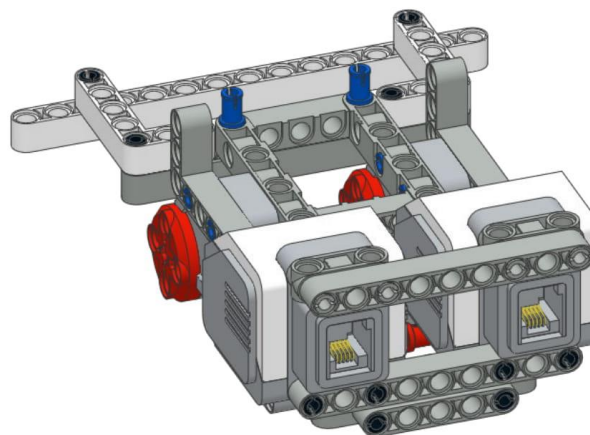
# Сумо

17   
2x



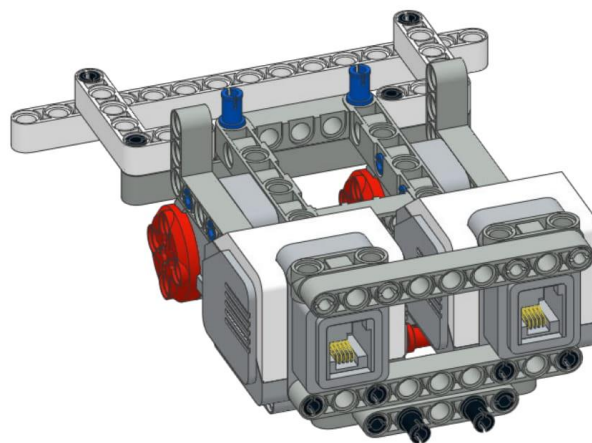
# Сумо

18  
1x



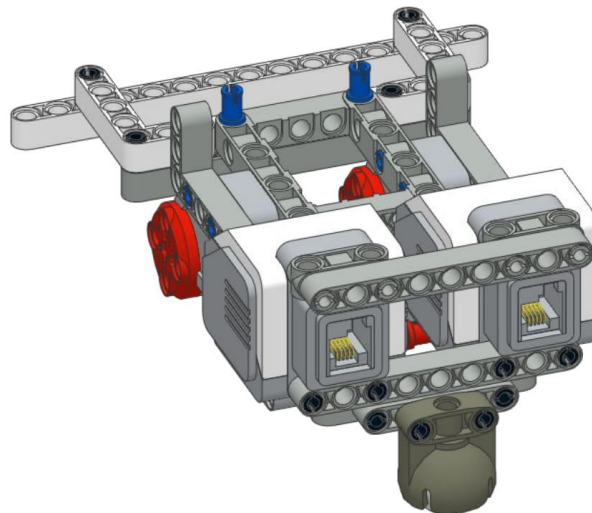
# Сумо

19   
2x



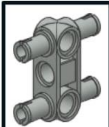
# Сумо

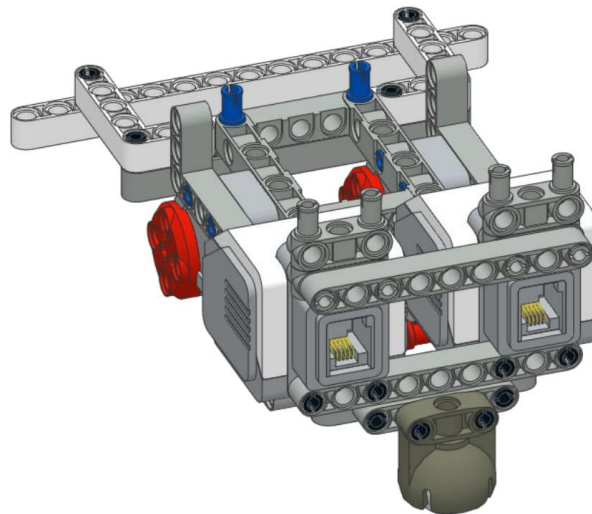
20  
1x





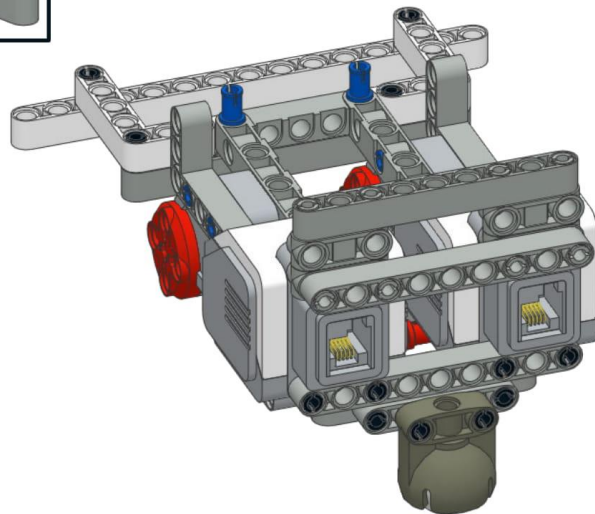
# Сумо

21   
2x



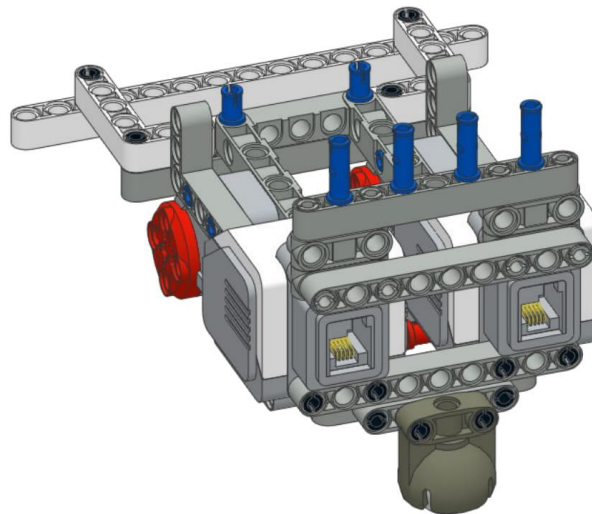
# Сумо

22  
1x



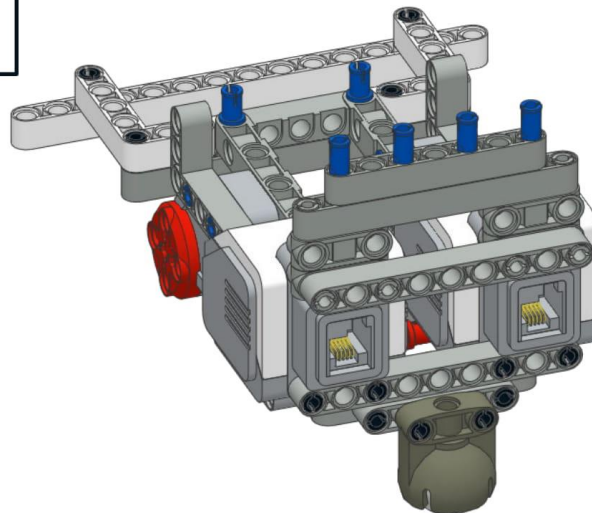
# Сумо

23   
4x



# Сумо

24  
1x

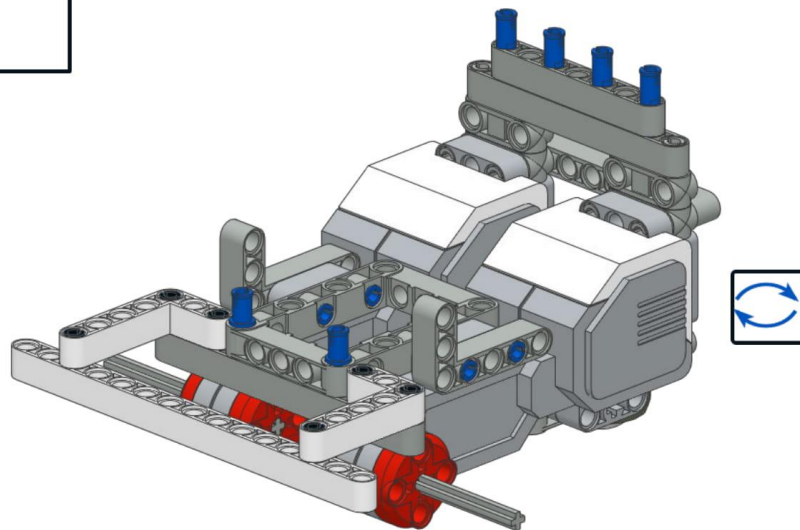


# Сумо

25  
2x

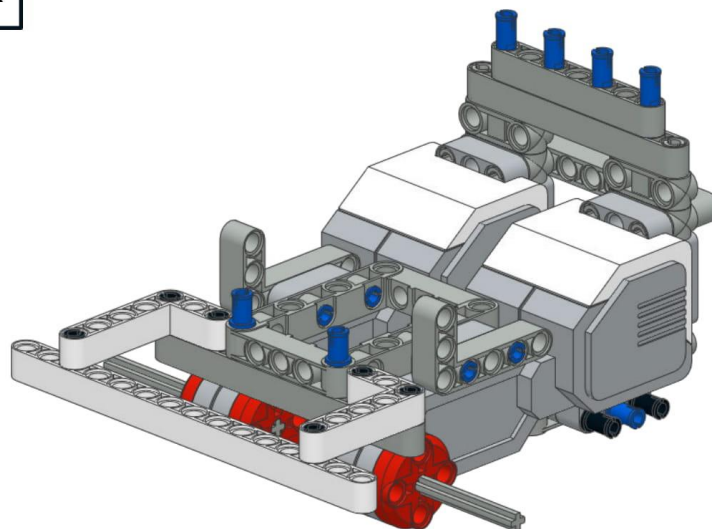


7



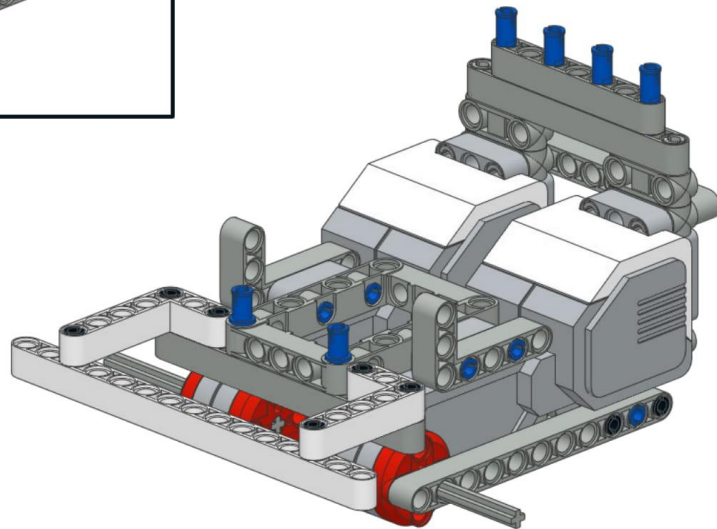
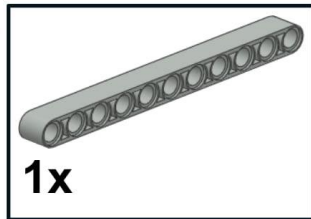
# Сумо

26   
2x 1x



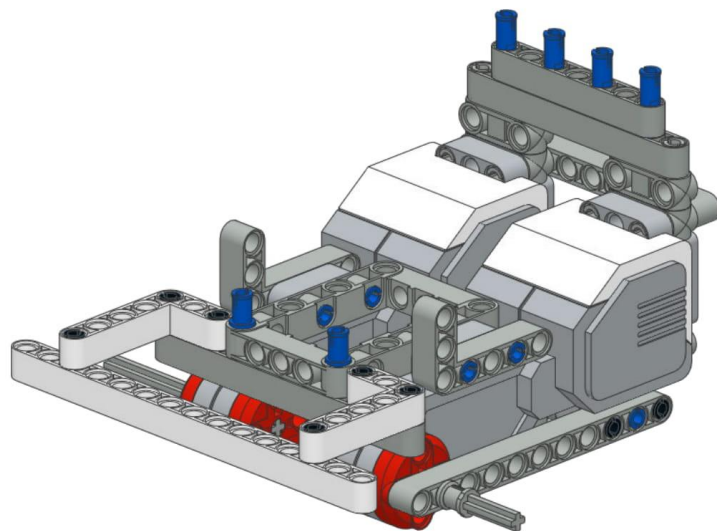
# Сумо

27



# Сумо

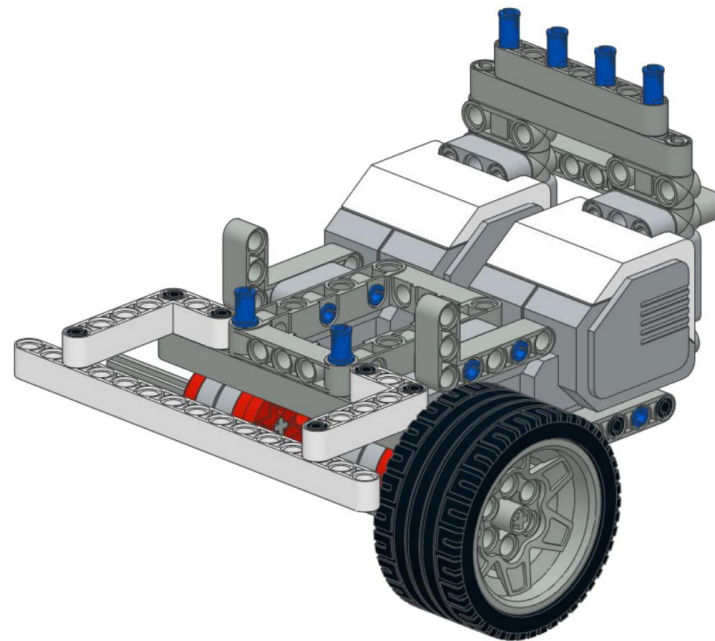
28   
1x





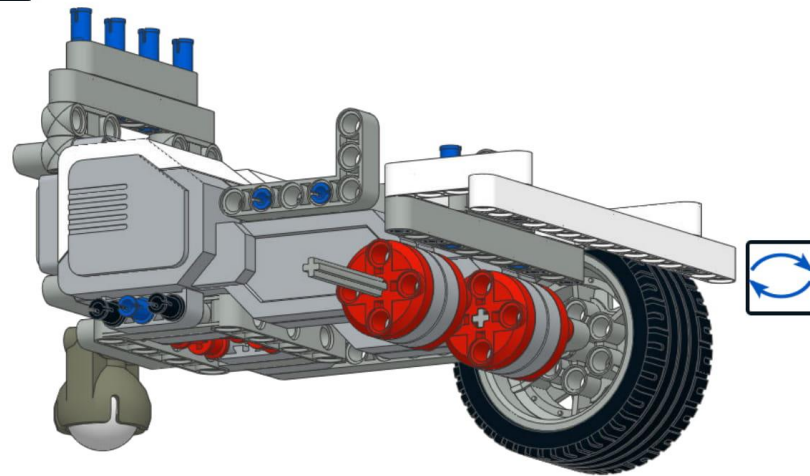
# Сумо

29



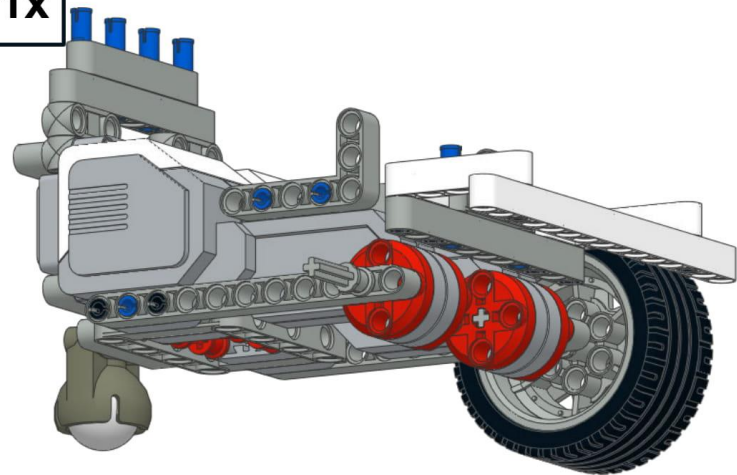
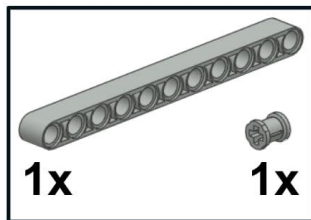
# Сумо

30   
2x 1x



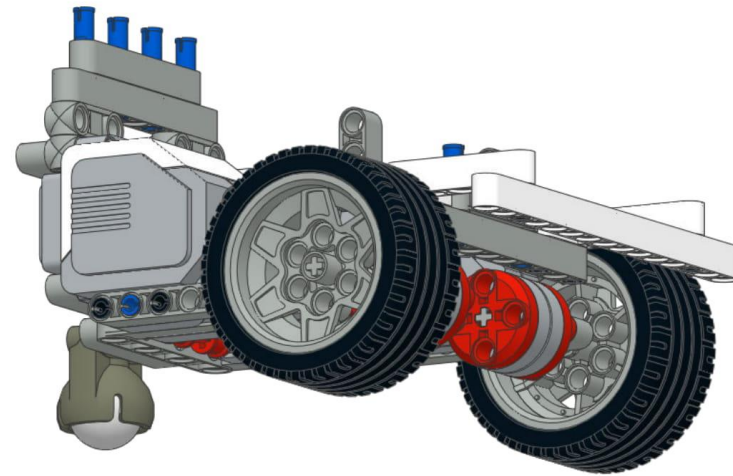
# Сумо

31

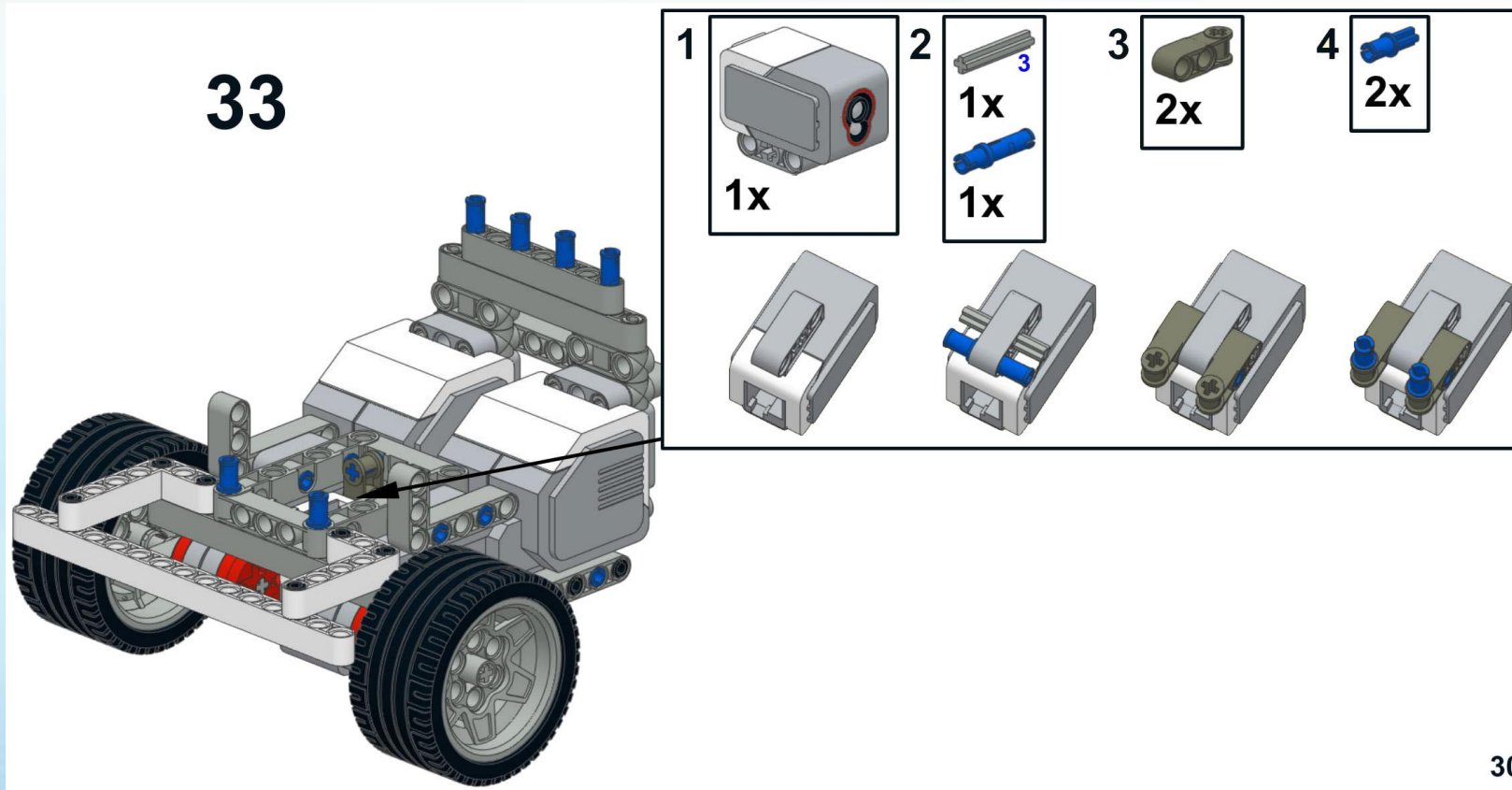


# Сумо

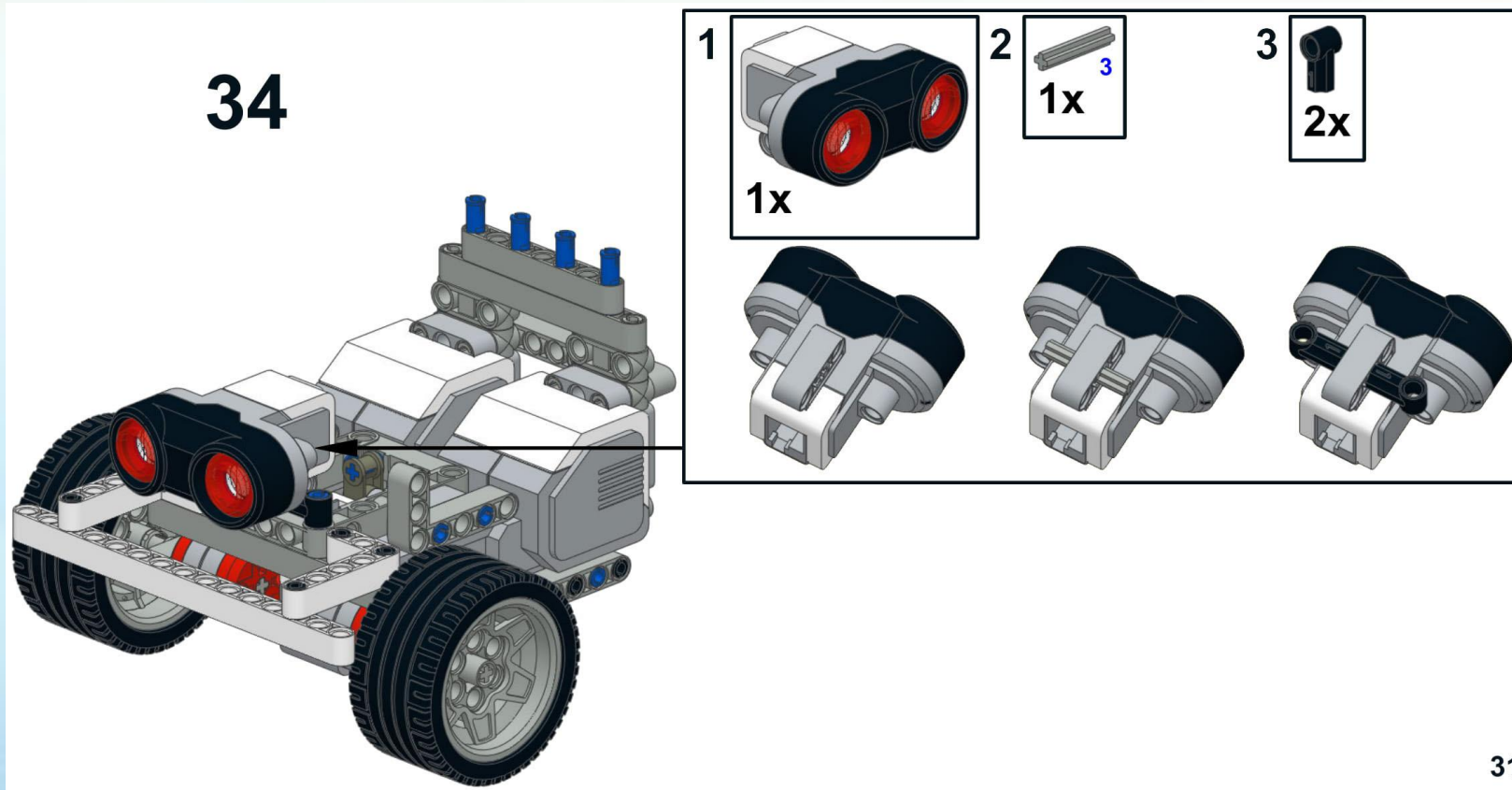
32



# Сумо



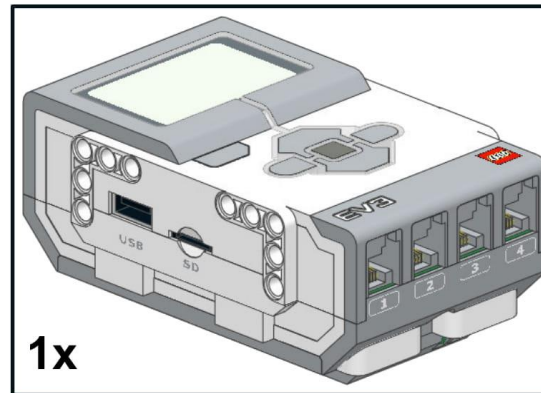
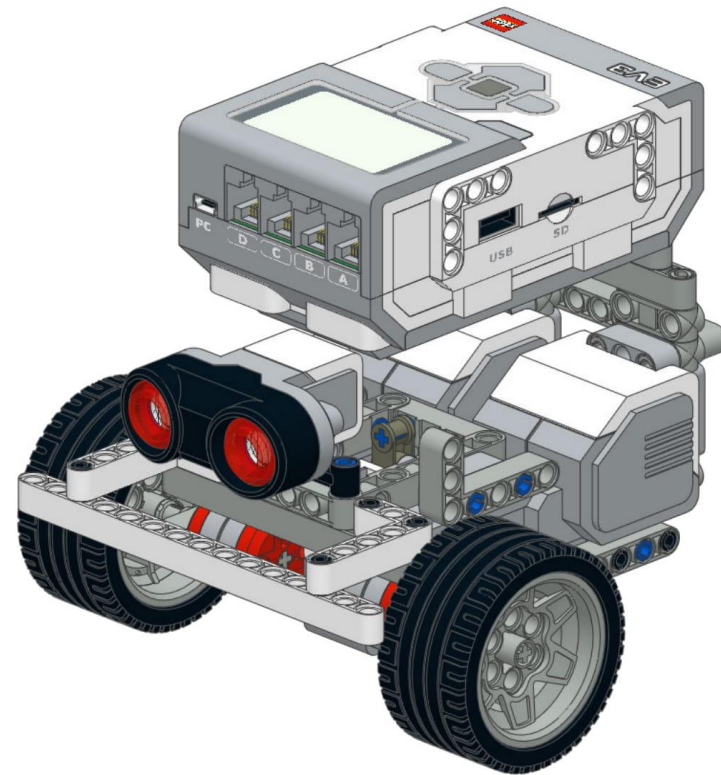
# Сумо





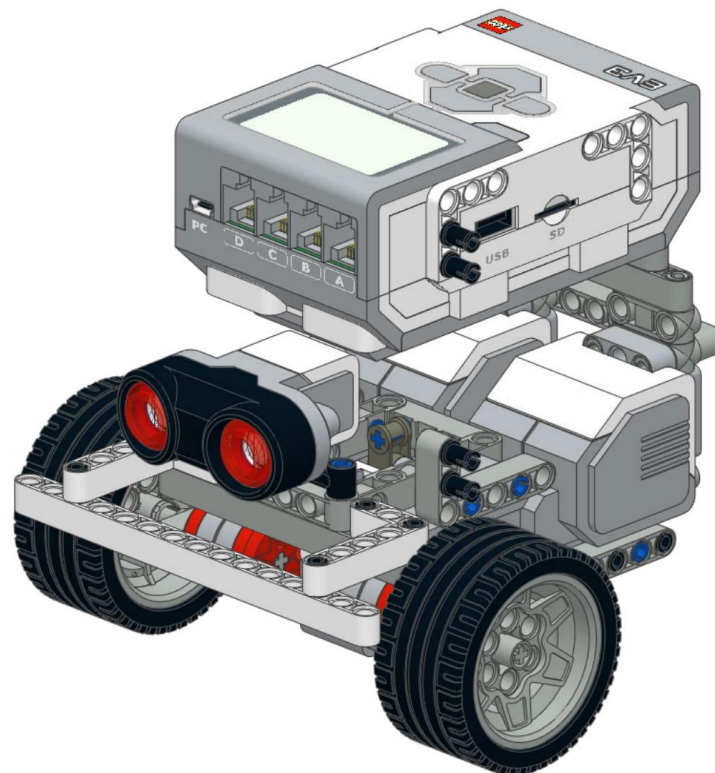
# Сумо

35



# Сумо

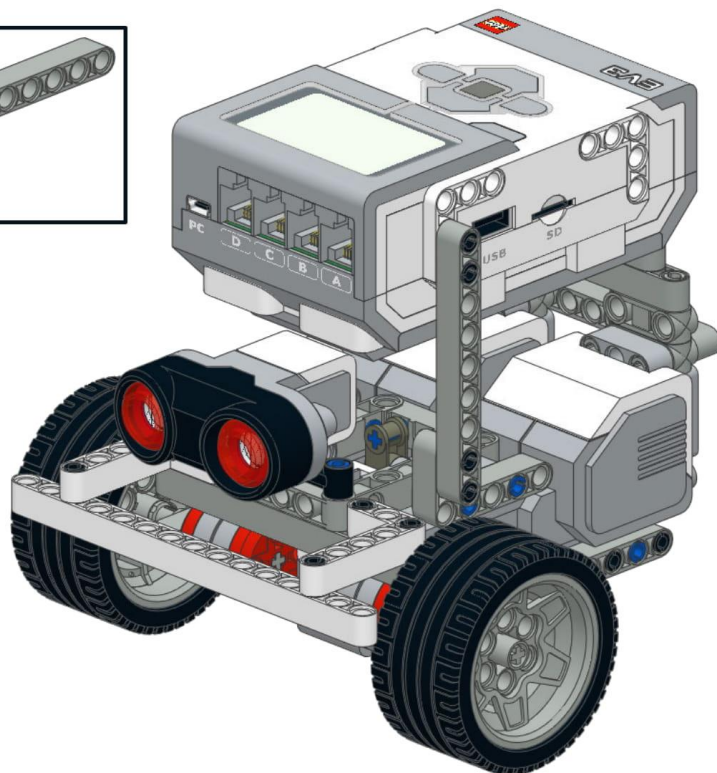
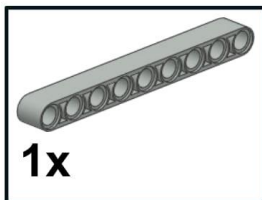
36  4x





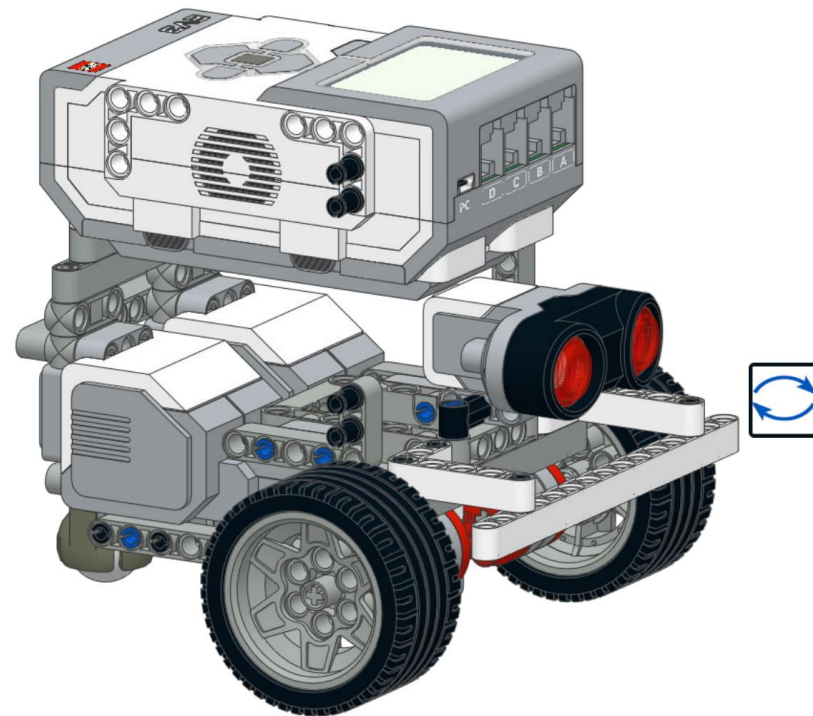
# Сумо

37



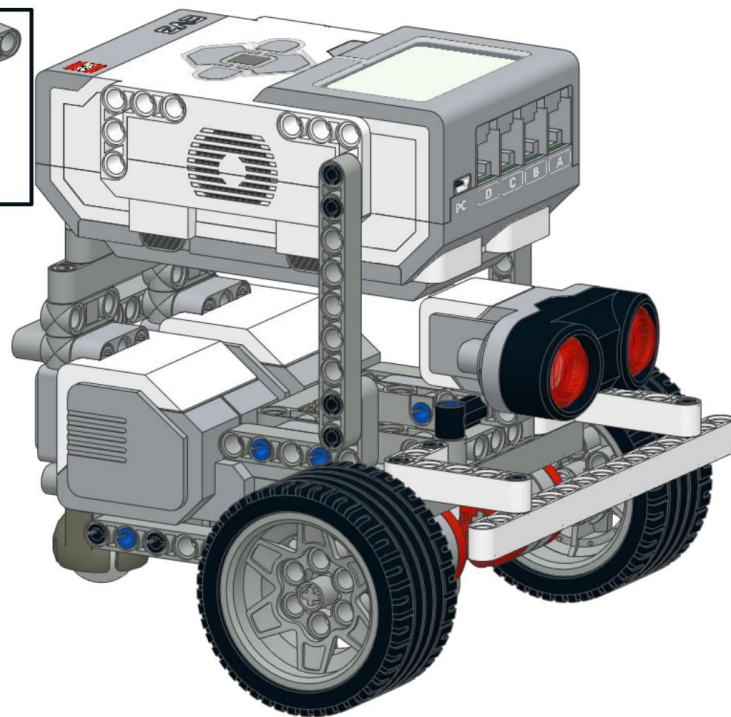
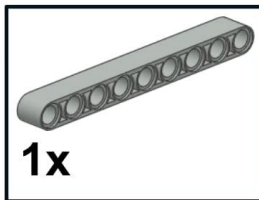
# Сумо

38   
4x



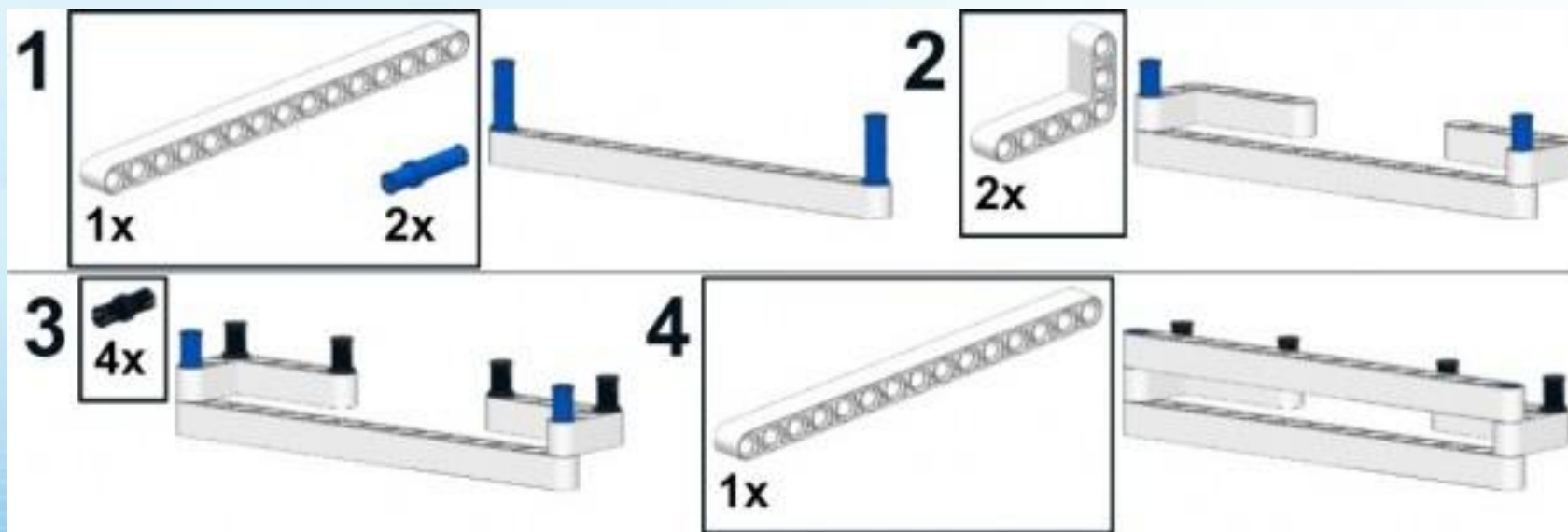
# Сумо

39



# Сумо

Основная часть нашего робота готова. Осталось только усилить переднюю часть, чтобы наш робот не повредил датчик расстояния при столкновении с противником.

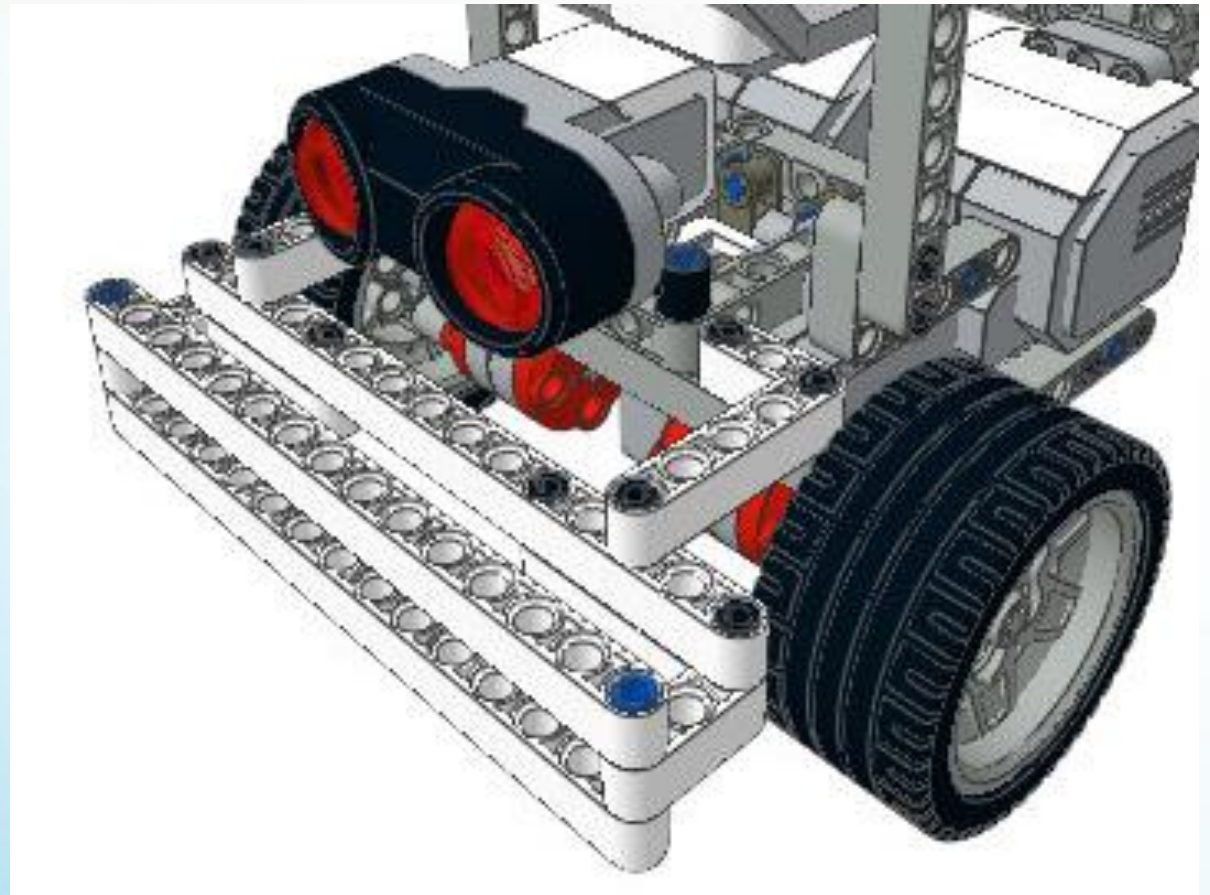


# Сумо

Подключаем моторы  
в порты В и С.

Датчик расстояния в порт 2.

Датчик цвета в порт 3.





---

# Программирование

---

СОЗДАДИМ ПРОГРАММУ

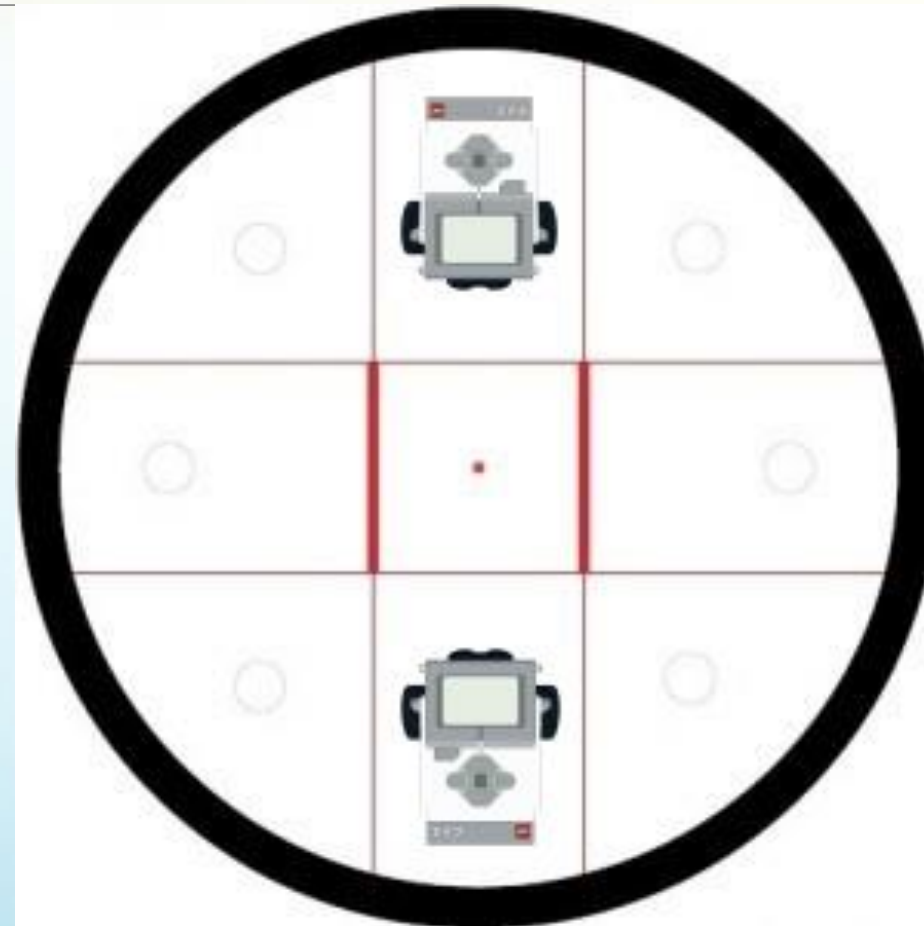




# Программирование

Первое, что нужно сделать – обнаружить соперника.

Для этого наш робот, подобно роботу из «Кегельринга», будет вращаться вокруг своей оси и останавливаться после обнаружения соперника



# Программирование

Так как теперь расстояние до объекта больше, чем в «Кегельринге», нам необходимо найти новое пороговое значение

Для этого установим роботов друг напротив друга как можно ближе к границам круга.

Такое положение соответствует максимальному удалению роботов друг от друга



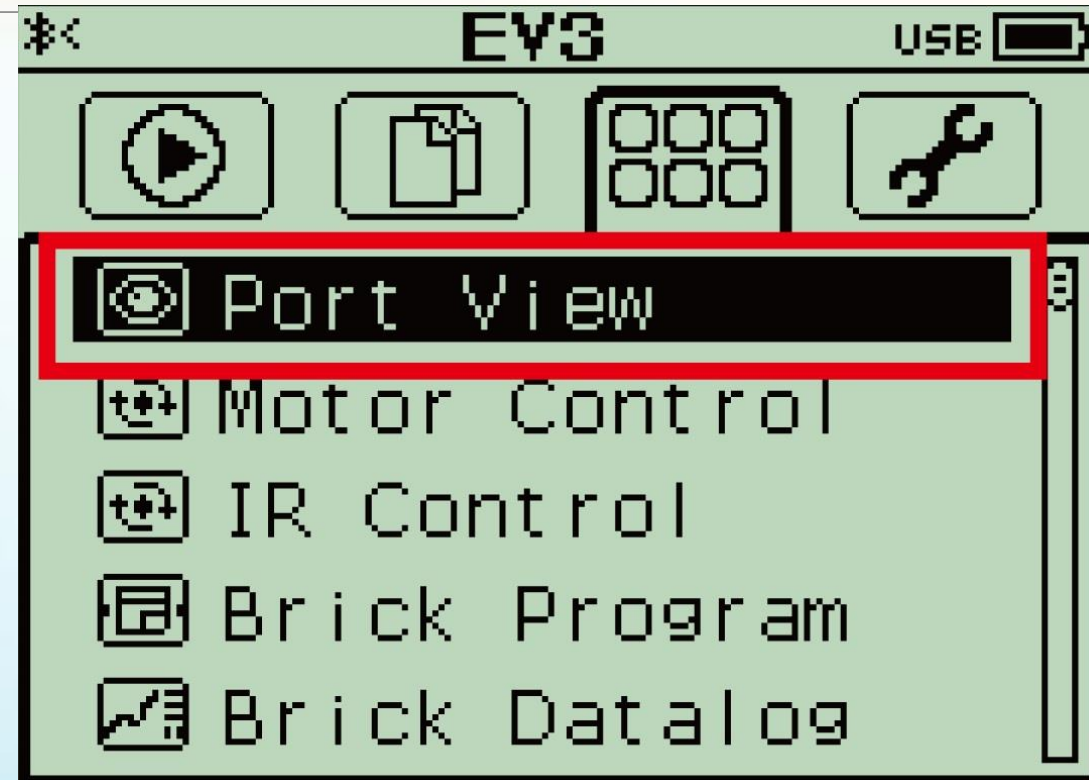


# Программирование

Снимем показания ультразвукового датчика с помощью блока EV3.

Датчик в режиме «Расстояние в сантиметрах» показывает значение 56.

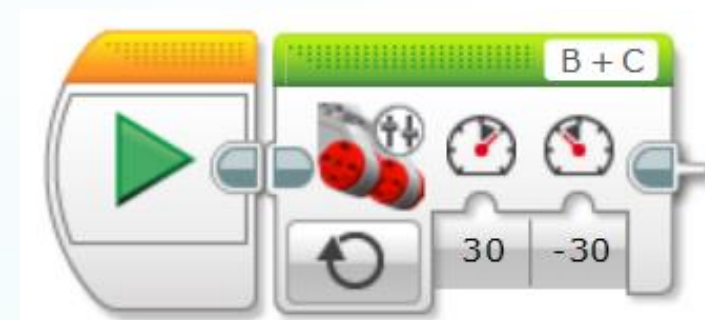
За пороговое примем 60.



# Программирование

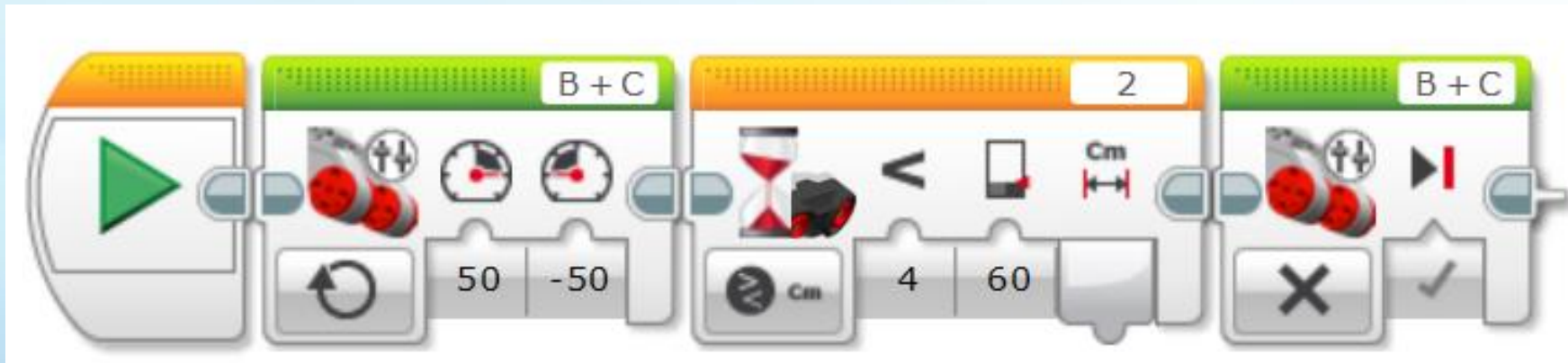
Для вращения вокруг своей оси воспользуемся блоком «Независимое управление моторами».

Для поиска соперника используем «Ожидание» с параметром «Ультразвуковой датчик», а пороговое значение укажем немного больше, чем расстояние до противника.



# Программирование

Теперь нужно экспериментально установить, при какой мощности моторов робот будет останавливаться точно напротив противника. Это значение и будем использовать в дальнейшем.



# Программирование

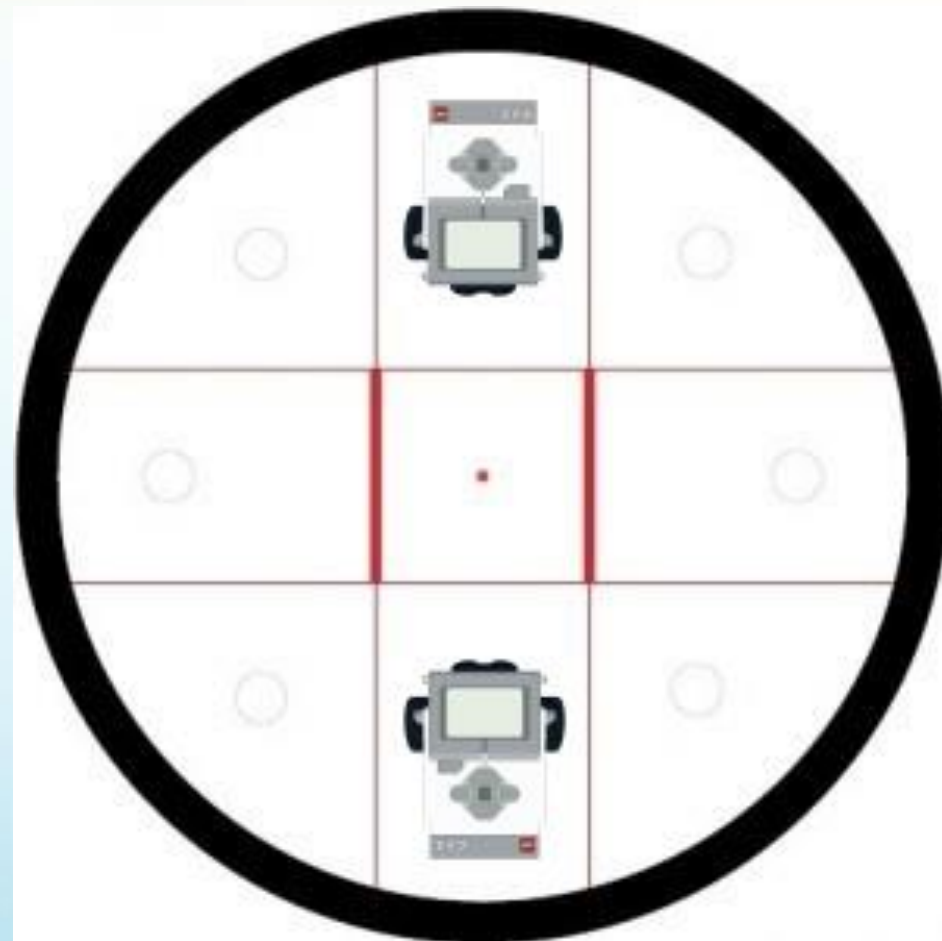
Поиск соперника в «Сумо» похож на поиск кегли в «Кегельринге».

Но вот поведение робота после нахождения соперника и выталкивание его за пределы поля имеет существенные различия.



# Программирование

После обнаружения соперника нужно устремиться в атаку на максимальной мощности моторов, при этом проверяя датчиком цвета, не вышел ли робот за границы ринга.



# Программирование

Но не все так просто! Не нужно забывать, что соперник тоже может двигаться. Поэтому возможна такая ситуация, когда соперник уйдет в сторону из-под направления нашей атаки.

В этом случае наш робот, промахнувшись, будет двигаться в сторону границы ринга, теряя соперника и драгоценное время.

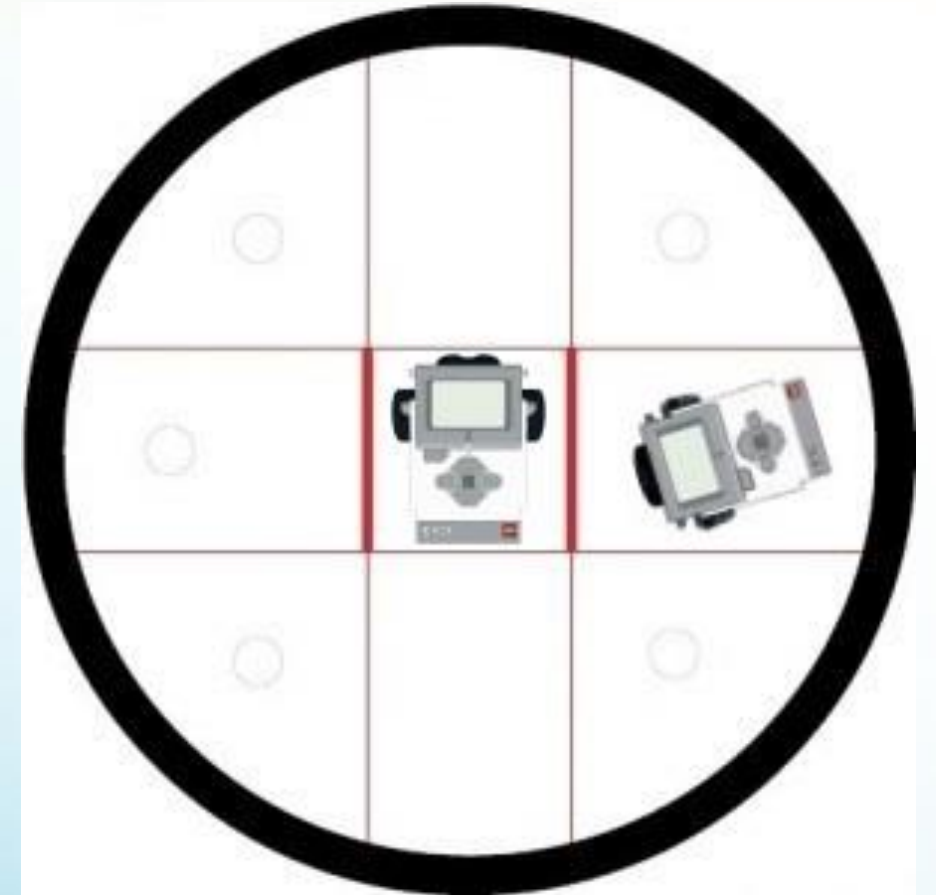




# Программирование

Следовательно, нам необходимо во время движения вперед анализировать оба датчика и прекращать атаку в случае, если робот потеряет соперника или достигнет границы ринга.

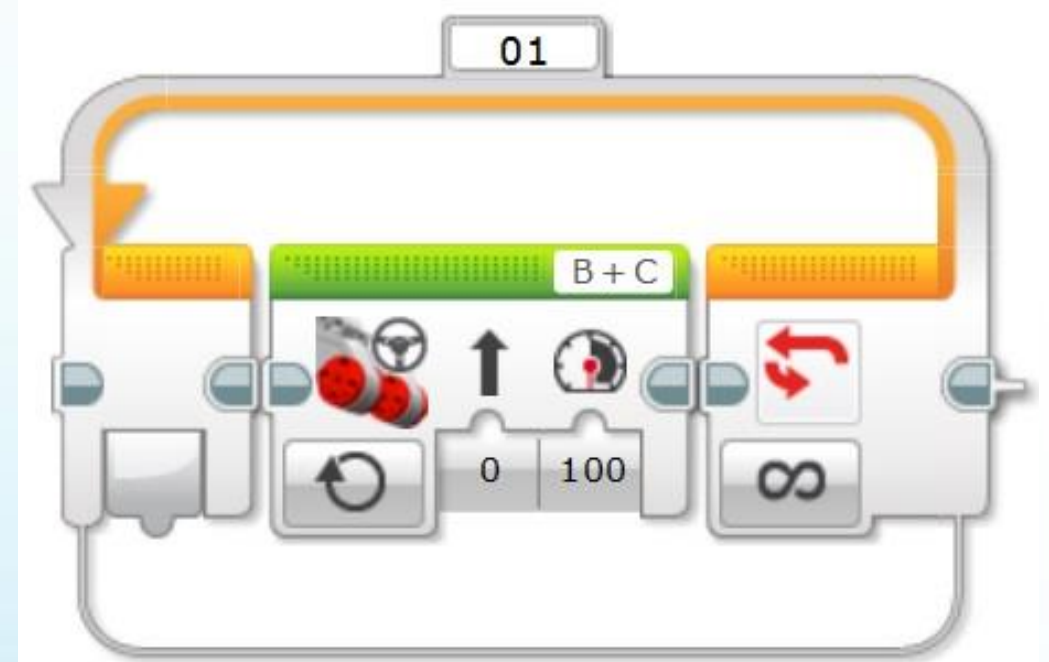
Поэтому нам необходимо отказаться от использования блока "Ожидание" и в цикле получать и обрабатывать показания двух датчиков.



# Программирование

Возьмем блок цикла и зададим роботу начало атаки.

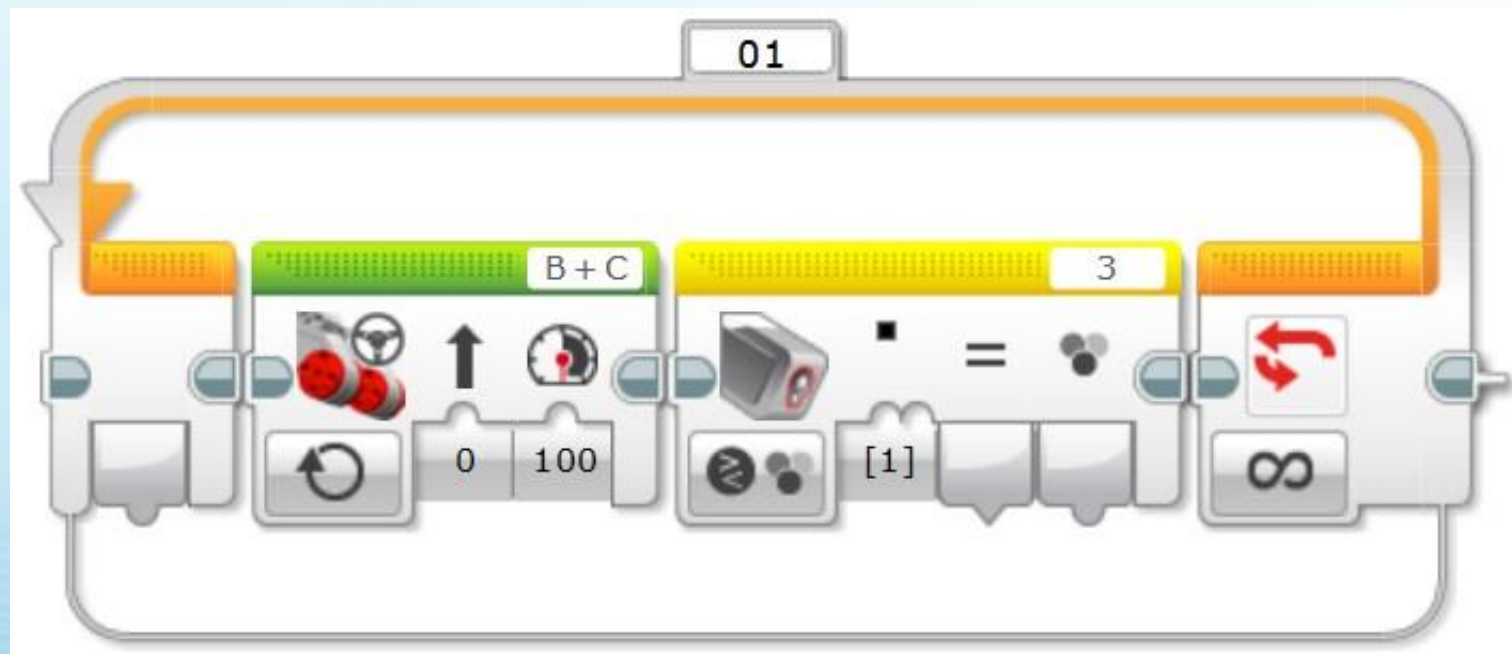
Для этого поместим внутрь цикла «Рулевое управление» и зададим максимальную мощность.





# Программирование

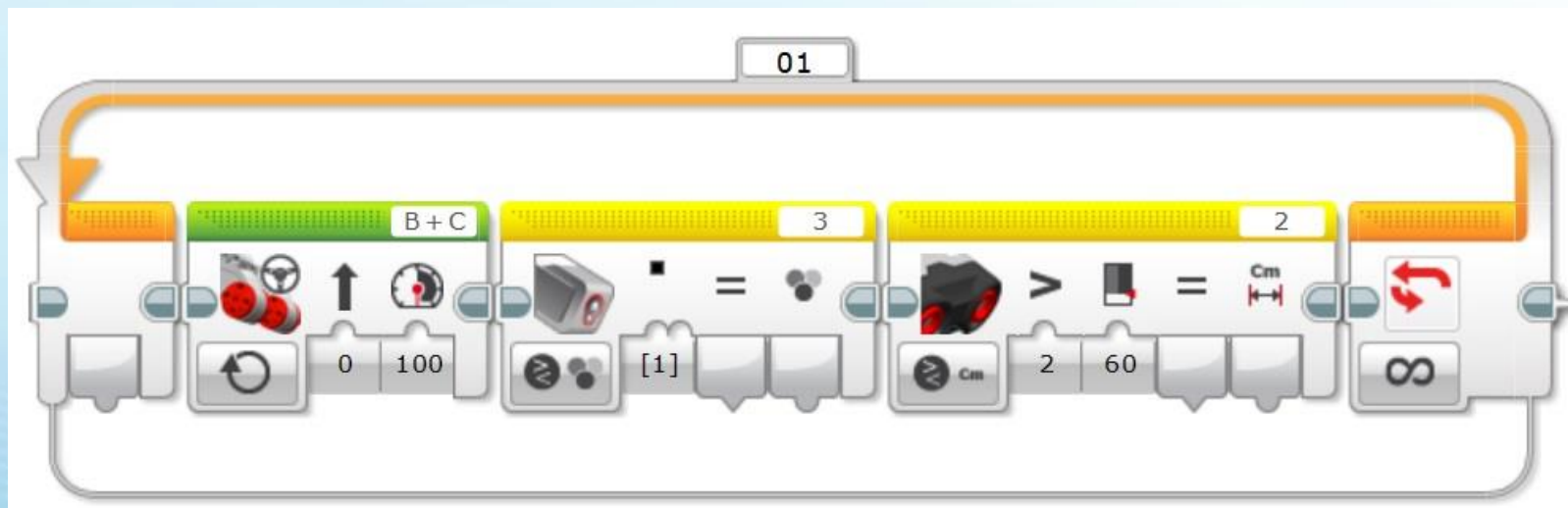
Следом за блоком «Рулевое управление моторами» поместим блок «Датчик цвета». Режим работы блока установим в значение «Сравнение – Цвет».



# Программирование

Установим блок «Ультразвуковой датчик». Режим работы блока установим в значение «Сравнение - Расстояние в сантиметрах».

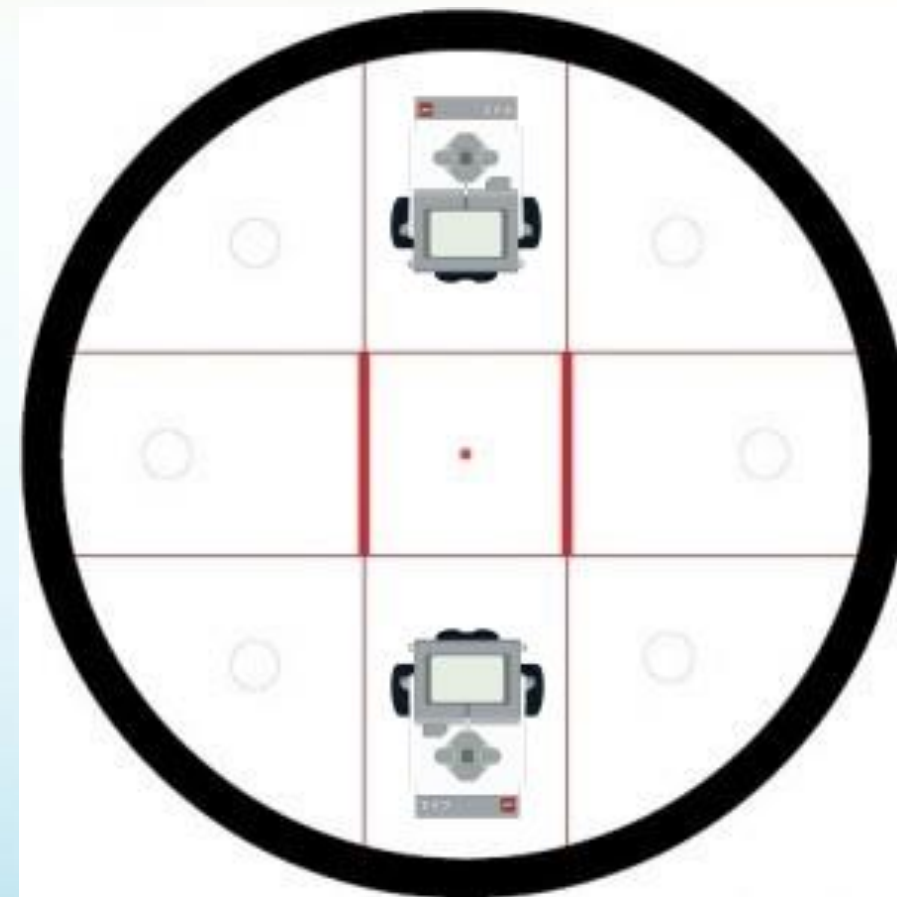
Пороговое значение зададим таким образом, чтобы при потере соперника из виду (расстояние увеличится и будет больше 60) блок выдавал логическое значение «Истина».



# Программирование

Давайте ещё раз проанализируем промежуточный код нашего алгоритма атаки: мы включили моторы на максимальную мощность и движемся вперед, постоянно в цикле опрашивая датчики.

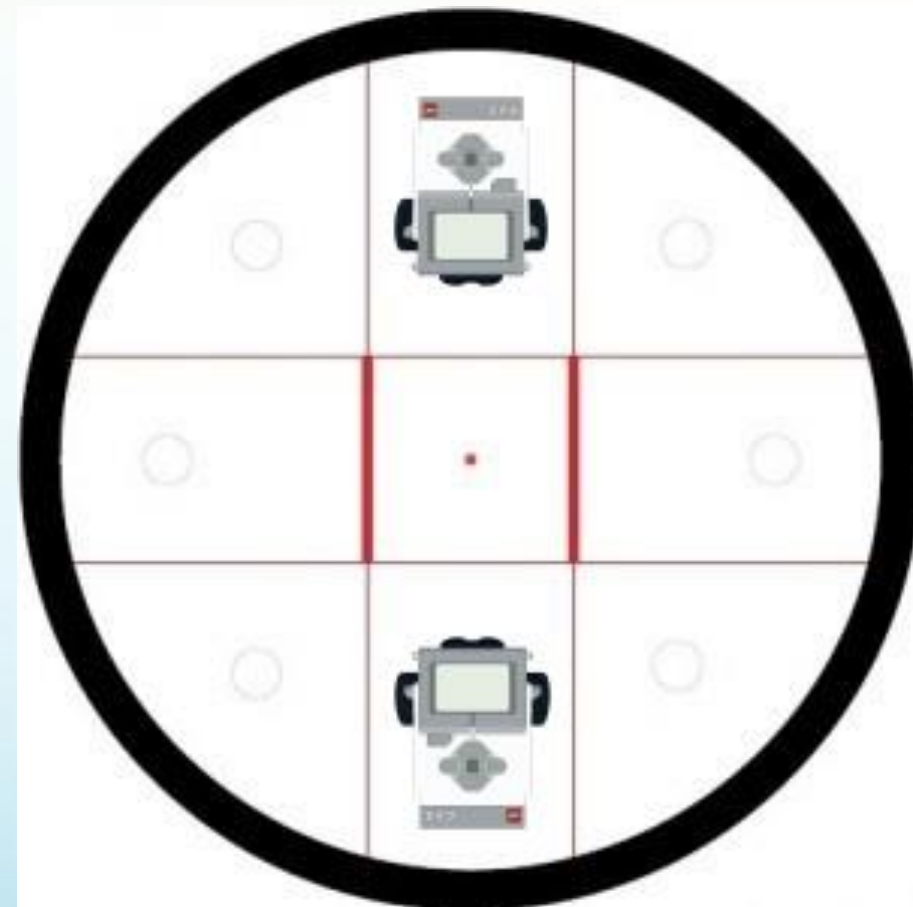
Если наш робот пересечет черную линию границы ринга, то значение выходного параметра «Результат сравнения» «Датчика цвета» примет значение «Истина»



# Программирование

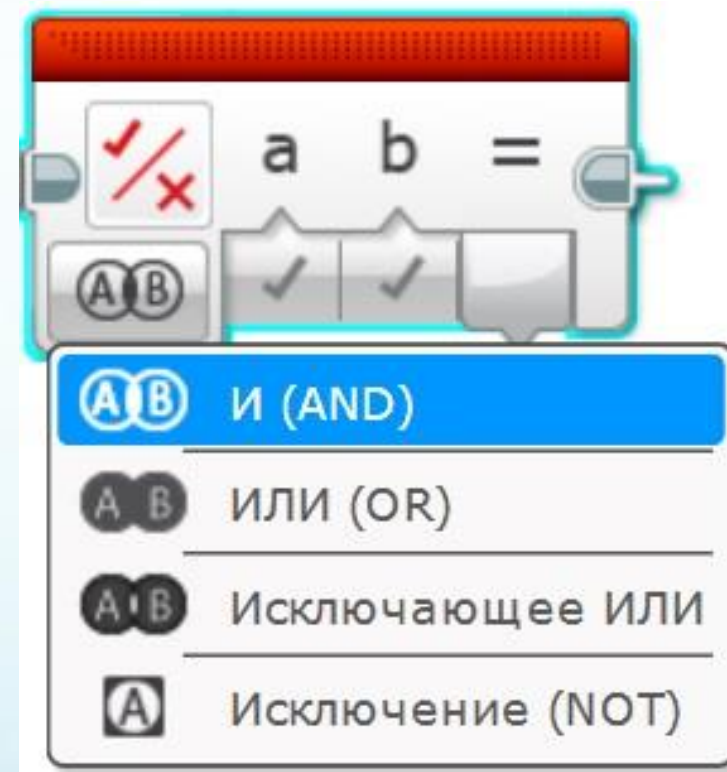
Если наш робот потеряет соперника, то значение выходного параметра «Результат сравнения» датчика, следящего за соперником, также примет значение «Истина».

В любом из этих случаев нам следует прекратить атаку, завершив наш цикл.



# Программирование

В этом нам поможет программный блок «Логические операции». Познакомимся с этим блоком подробнее: программный блок «Логические операции» предназначен для выполнения операций над логическими данными.



# Программирование

Выбранный режим программного блока «Логические операции» определяет одну из четырех операций над логическими данными:

«И (AND)»,

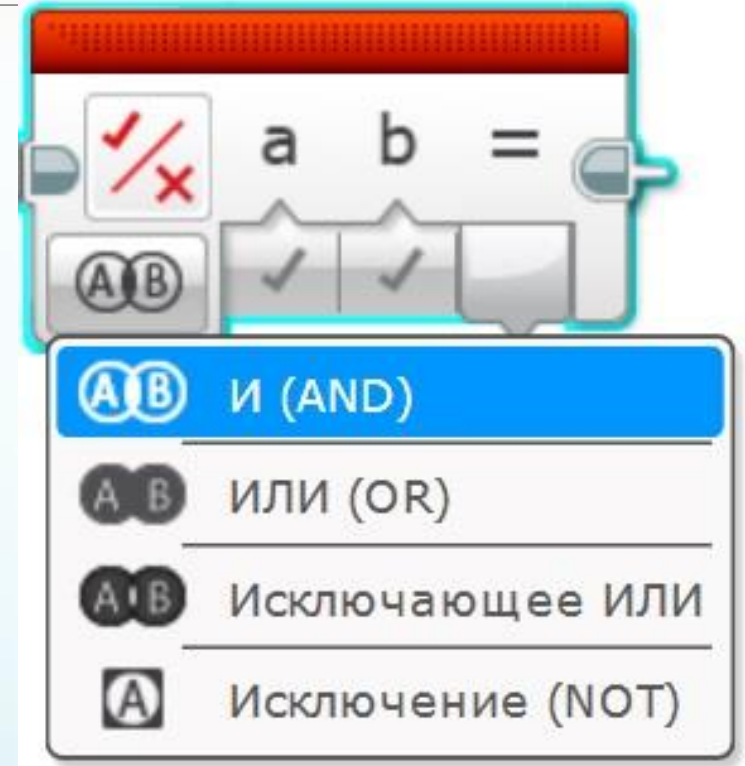
«ИЛИ (OR)»,

«Исключающее ИЛИ» и

«Исключение (NOT)».

Логическое значение может находиться в одном из двух состояний:

«Истина» или «Ложь».



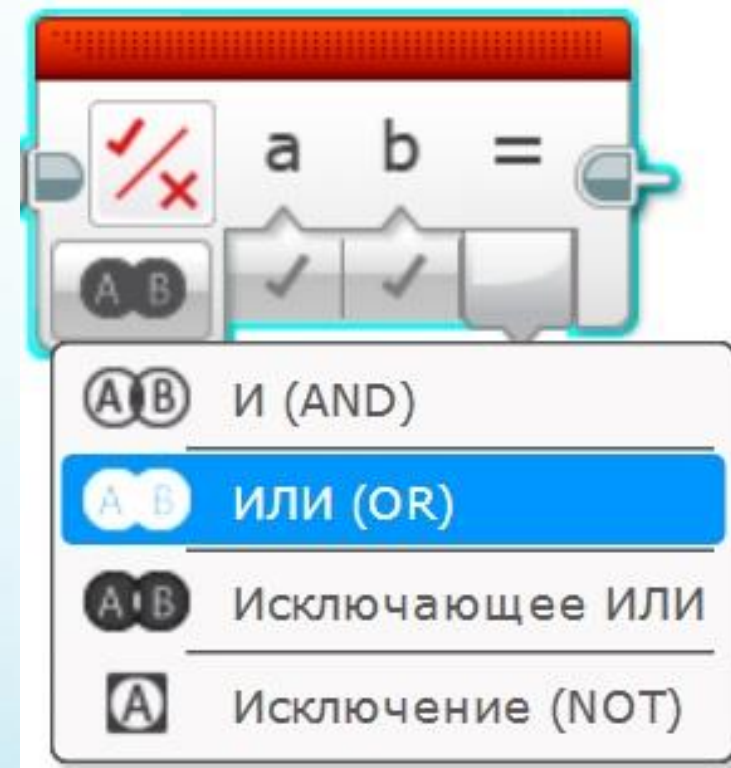


# Программирование

В нашем случае робот должен прекратить движение, если хотя бы одно из условий выполнено:

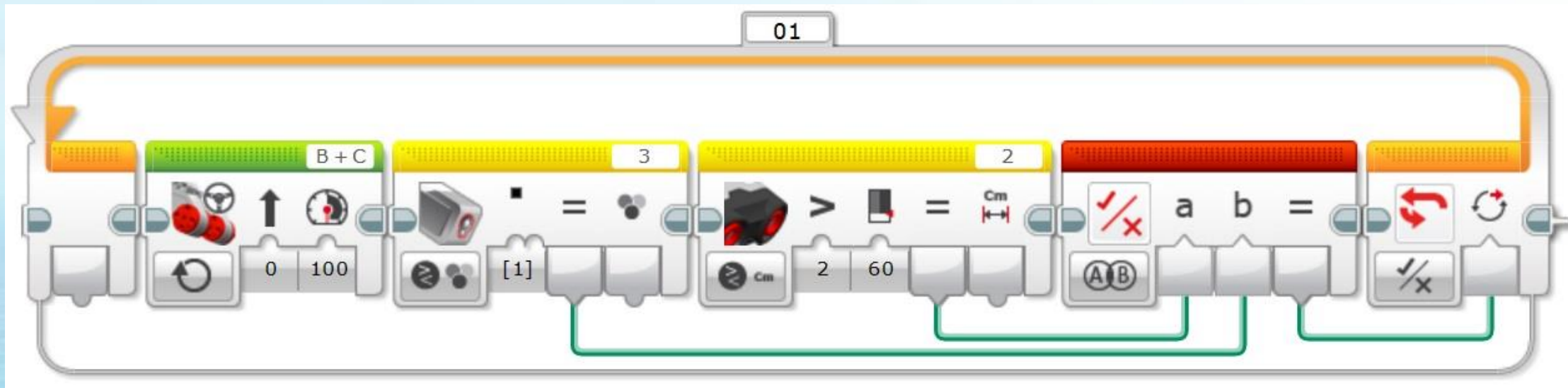
1. Робот пересек границу.
2. Робот потерял противника из виду.

Параметр «ИЛИ» выдаст «Истину» в том случае, если выполнено хотя бы одно условие.



# Программирование

Соединим каждое из условий с блоком «Логические операции». Ограничим цикл, задав параметр «Логическое значение». Любое из условий заканчивает цикл, поэтому результат блока «Логические операции» связываем с параметром цикла.





# Программирование

---

Протестируем получившийся алгоритм атаки!

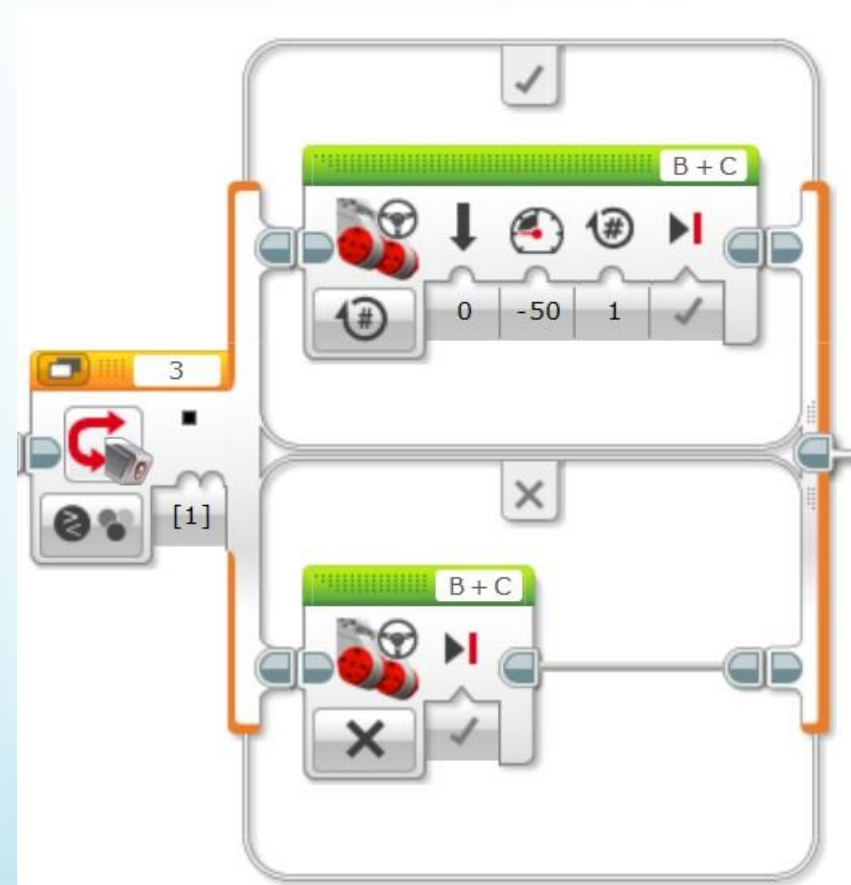
**Поместим робота в центр ринга, а неподвижного противника поставим напротив. Робот должен вытолкнуть его и остановиться на границе ринга.**

**Снова выставим роботов на исходные позиции и запустим программу атаки. Когда наш робот устремится к сопернику и приблизится достаточно близко, резко уберём соперника в сторону. Наш робот потеряет соперника из виду и остановится.**

# Программирование

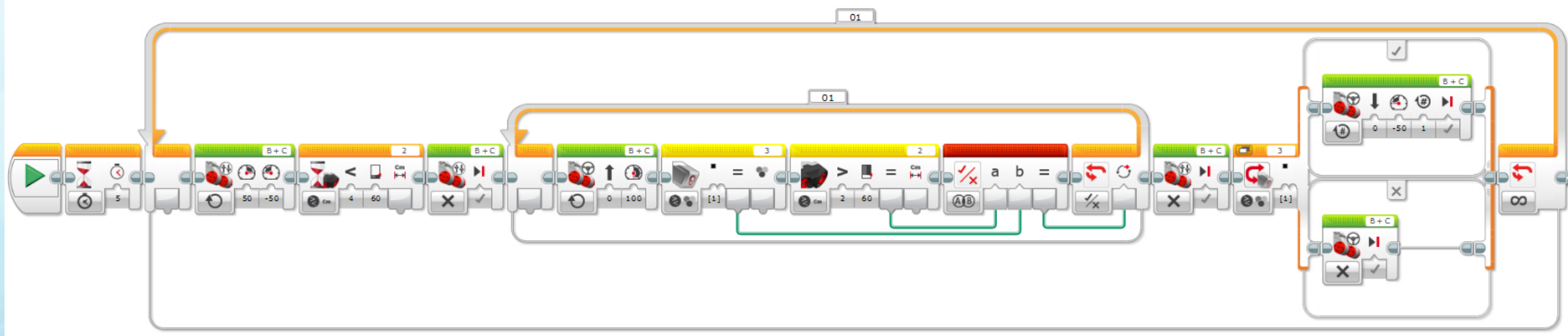
После атаки, если она оказалась неудачной (соперник вовремя увернулся, а робот доехал до границы ринга), нужно заехать внутрь ринга и заново начать поиск соперника на поле.

Для этого с помощью «Переключателя» проверим, находится ли робот на черной линии или нет.



# Программирование

Соединим программу воедино!



# Сумо

---

**Настало время выяснить, чей робот сильнее!**

**Разбейтесь на пары  
и посоревнуйтесь между собой.**

