

Р • О • Б • О • Ф • И • Ш • К • И

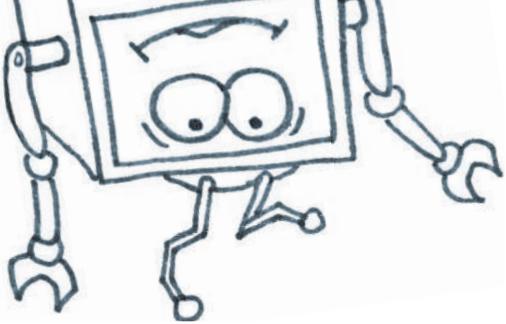


# КОНСТРУИРУЕМ РОБОТОВ

на **LEGO<sup>®</sup>MINDSTORMS<sup>®</sup>**  
Education EV3

Сборник  
проектов №1





# КОНСТРУИРУЕМ РОБОТОВ

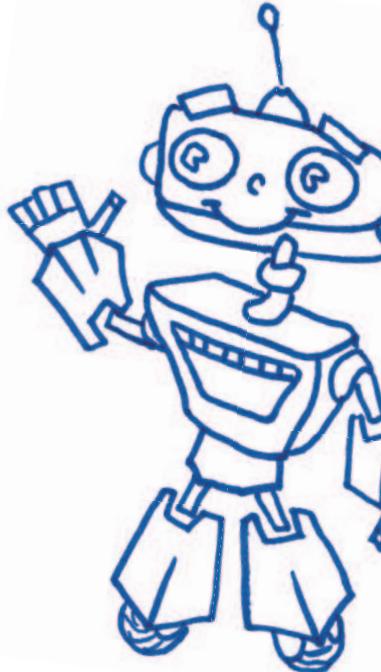
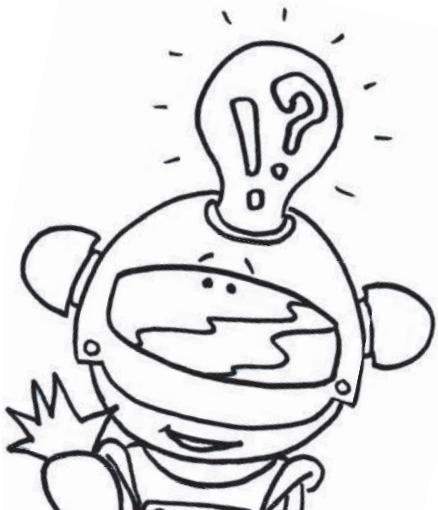
на **LEGO<sup>®</sup>** MINDSTORMS<sup>®</sup>  
Education EV3

Сборник  
проектов № 1

Электронное  
издание



Лаборатория знаний  
Москва  
2019



УДК 373.167

ББК 32.97

К65

*Серия основана в 2016 г.*

Ведущие редакторы серии

*Т. Г. Хохлова, Ю. А. Серова*

А в т о р ы:

Салахова А. А. (проекты «Сторожевая башня» в соавторстве, «Волшебная палочка» в соавторстве);

Красных А. В. (проекты «Сторожевая башня» в соавторстве, «Волшебная палочка» в соавторстве);

Тарапата В. В. (проекты «Тайный код Сэмюэла Морзе», «Волшебная палочка» в соавторстве);

Валуев А. А. (проект «Робочист спешит на помощь!»);

Сафули В. Г. (проект «Посторонним вход воспрещён!» в соавторстве);

Дорожкина Н. Г. (проект «Посторонним вход воспрещён!» в соавторстве);

Зайцева Н. Н. (проект «Человек – всему мера?» в соавторстве);

Цуканова Е. А. (проект «Человек – всему мера?» в соавторстве)

**Конструируем** роботов на LEGO® MINDSTORMS®  
K65 Education EV3. Сборник проектов №1 [Электронный  
ресурс] / сост. Ю. А. Серова. — Эл. изд. — Электрон. тек-  
стовые дан. (1 файл pdf : 251 с.). — М. : Лаборатория  
знаний, 2019. — (РОБОФИШКИ). — Систем. требования:  
Adobe Reader XI ; экран 10".

ISBN 978-5-00101-652-2

Стать гениальным изобретателем легко! Серия книг «РОБО-  
ФИШКИ» поможет вам создавать роботов, учиться и играть  
вместе с ними.

В сборник №1 вошли шесть проектов, с помощью которых  
вы сможете собрать роботов, выполняющих различные задачи:  
охраняющего территорию от посторонних, держащего в чистоте  
рабочие поверхности, умеющего рисовать и измерять длину  
любой линии, а также передавать сообщения.

Для технического творчества в школе и дома, а также для  
занятий в робототехнических кружках.

**УДК 373.167**

**ББК 32.97**

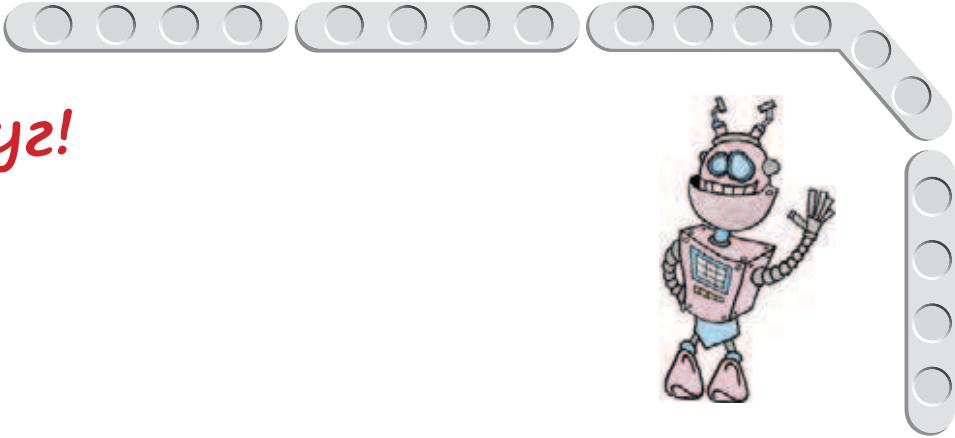
**Деривативное электронное издание на основе печатного  
аналога:** Конструируем роботов на LEGO® MINDSTORMS®  
Education EV3. Сборник проектов №1 / сост. Ю. А. Серова. —  
М. : Лаборатория знаний, 2019. — 248 с. : ил. —  
(РОБОФИШКИ). — ISBN 978-5-00101-190-3.

(6+)

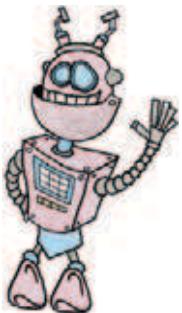
**В соответствии со ст. 1299 и 1301 ГК РФ при устраниении ограничений, установленных  
техническими средствами защиты авторских прав, правообладатель вправе требовать от  
нарушителя возмещения убытков или выплаты компенсации**

ISBN 978-5-00101-652-2

© Лаборатория знаний, 2018



# Дорогой друг!



Как видно, ты уже совсем не новичок в LEGO, раз добрался до набора LEGO® MINDSTORMS® Education EV3 и, конечно, быстро собрал всё, что там предлагалось!

Что же делать теперь? Набор дорогой, выбрасывать жалко, а у младшего братика (если он есть) пока другие игрушки. Не расстраивайся! Мы тебе поможем.

Из этого набора можно собрать ещё много интересных и полезных вещей. Например, ты можешь собрать:

- сторожевую башню с огневой поддержкой;
- робота, который может измерять длину любой линии: от прямой до самой извилистой, причём в любых единицах длины;
- робота, способного самостоятельно протирать пыль на любых горизонтальных поверхностях, а также доводить до блеска покрытие твоего рабочего стола;
- роботизированный охранный комплекс, защищающий твою комнату от посторонних;
- робота, способного создавать замысловатые узоры;
- кнопочный звуковой передатчик, с помощью которого ты сможешь общаться с другом на тайном языке или посыпать телеграммы одноклассникам.

## Задумайся над этим!

Фактически за какой-то час работы ты сумеешь пройти многовековой путь изобретателей прошлого!

Почему в настоящее время такое стало возможно? Можно ли изобрести что-нибудь новое, не зная, какие машины и механизмы существовали в прошлом? Как интересней работать — одному или вместе с другом?

### Внимание!

Ты можешь собрать свои достижения в робототехнике в электронное портфолио! Фотографирай или фиксируй на видео результаты своей работы, чтобы потом представить их для участия в творческих конкурсах. Результаты конкурсов и олимпиад засчитываются при поступлении в профессиональные учебные заведения.

### Оборудование:

- Базовый набор LEGO® MINDSTORMS® Education EV3.
- Компьютер (минимальные системные требования): Windows XP, Vista, Windows 7, Windows 8 (за исключением METRO), Windows 10 (32/64 бит), оперативная память не менее 1 Гб, процессор – 1,6 ГГц (или быстрее), разрешение экрана – 1024 × 600, свободное место на диске – 5 Гб.
- Программное обеспечение LEGO® MINDSTORMS® Education EV3 (LME-EV3).

### Обозначения

В тексте тебе встретятся обозначения, которые мы сейчас поясним на примерах.

1. Балка № 7 – это балка с семью отверстиями.
2. 3-модульный штифт – штифт, длина которого равна длине балки № 3.
3. Ось № 5 – ось, длина которой равна длине балки № 5.

# Сторожевая башня





# Секреты сторожевых башен



Свистать всех наверх! Ах, простите. Мы немного ошиблись с проектом. Но, поверьте, здесь тоже будет интересно и высоко. Как вы думаете, когда возникает необходимость сторожить, нужно присматривать только за внутренним вверенным пространством? Наверняка, играя в современные компьютерные игры с красочными открытыми мирами или смотря сериалы, вы сталкивались с огромными и зловещими дозорными башнями — последними оплотами на пути тёмных сил или орд северного противника.

Сторожевые башни действительно были созданы для того, чтобы нести дозор, поэтому с них можно смотреть вдаль на большие расстояния. Враг не пройдёт! Прежде чем собрать и запрограммировать собственный современнейший охранный комплекс, предлагаем совершить незабываемое путешествие в прошлое через страны и века, чтобы наверняка учесть всё необходимое. Ты же знаешь, что любой проект начинается с исследования имеющихся решений и области решений.



**Рис. 1.** Сторожевая постройка  
для охраны периметра

Сторожевые башни появились практически одновременно с первыми организованными армиями и городами. Первые из них были, если говорить честно, не совсем башнями — высокими постройками, размещенными на холме или в стене укреплений поселения. Чаще всего дозорные следили за приближением вражеских войск, однако в мирное время дозорные могли также наблюдать за правопорядком на тракте, ведущем к городу, и помогать стражникам на воротах. Иногда приходилось следить и за морем, причём сторожевые башни в этом случае совмещали функции маяка. В греческих полисах на сторожевых башнях начали использовать огонь, а затем и систему зеркал, чтобы передавать на дальние расстояния через цепь дозорных пунктов информацию. Это получалось намного быстрее, чем отправлять гонца, которого могли перехватить или убить по дороге.



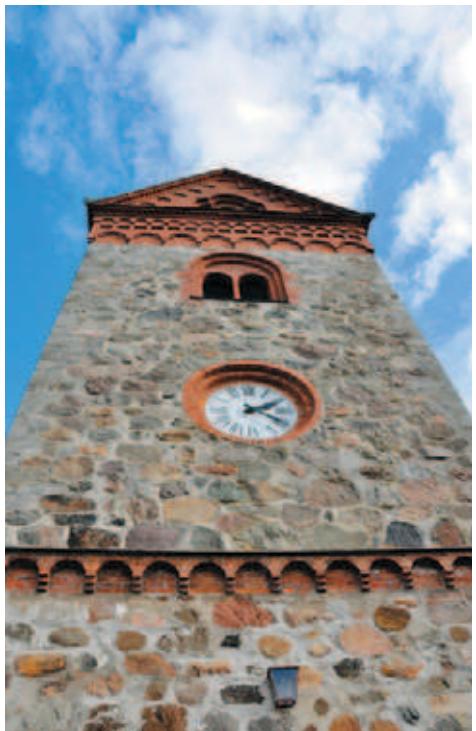
Итак, ты познакомился с прототипом телеграфа, однако зеркала иногда применялись весьма нетривиально. Известный античный математик Архимед при осаде его родного города Сиракуз в 213 году до нашей эры использовал сторожевую башню в оборонительных целях. Как впоследствии утверждал писатель Лукиан, основываясь на устных источниках, Архимед установил на башню шестиугольное зеркало, которое выполняло роль вогнутой линзы, собравшей в одной точке солнечный свет наподобие самого настоящего лазера, с помощью которого и были подожжены римские корабли. Как тебе утверждение о существовании до нашей эры сторожевых башен с лазером?

#### Кстати!

Знаменитый французский музей Лувр изначально был крепостью со сторожевой башней, однако сейчас башня практически разрушена: её остатки находятся на нижнем этаже музея в виде экспозиции.

В Европе в качестве сторожевых башен обычно выступали донжоны (от французского *donjon* — господская башня) — главные башни феодальных замков, находящиеся внутри крепостных стен (в отличие от более старых крепостных дозорных башен или башен, вынесенных за границы фортификаций). Мы уже упоминали римлян, но на территории «варварской Европы» (Франция, Германия и т. д.) большинство старых сторожевых башен построено именно ими, как и, кстати, многие дороги и акведуки. Самые интересные из европейских башен принадлежат испанским замкам и построены в XIII–XVI веках в период господства османов на полуострове. Это отдельные башни, расположенные вдоль стен и соединённые мостами. Они носят гордое название «Торре Альбаррана» (*Torre Albarрана*). Затем христианские ордены переняли манеру создания подобных сооружений.

Теперь перенесёмся на территорию современной Российской Федерации, а точнее в Итум-Калинский район Чечни. Словно бы в ответ Испании, в XI–XII веках здесь, на правом берегу прекрасной горной реки Чанты-Аргун недалеко от Ушканы были возведены две башни-близнецы, поражающие своим внешним видом проезжающих путников до сих пор. Ты наверняка видел подобные величественные постройки в при-



**Рис. 2.** Высокие башни монастырей и церквей иногда становились дозорными



**Рис. 3.** Ушканойские башни  
(фотография из архива Wikipedia)

сразу за тремя направлениями. У дозорных сооружений стала сильнее различаться специализация. Начали появляться первые башни противовоздушной обороны (ПВО) и отдельно системы охраны периметра от наземных противников. Особую роль импровизированные сторожевые башни сыграли в Битве за Москву (30 сентября 1941 года — 20 апреля 1942 года), особенно в её оборонительный период (с 30 сентября по 4 декабря). В это время москвичи использовали высокие постройки, чтобы разместить на них дополнительную огневую поддержку в виде зенитных установок. Например, сохранился кадр военной хроники с советскими зенитчиками на крыше гостиницы «Москва», отслеживающими в небе немецкие самолёты.

Сегодня средства ПВО, в том числе Московского округа ПВО, более совершенны технологически и в размере огневой силы. Часто современные системы охраны периметра и ПВО имеют интеллектуальные

системы дозора, позволяющие производить проверку в автоматическом режиме. Для этого используются различные технологии. Одной из них является применение сонаров.



**Рис. 4.** Репродукция снимка. Россия, Москва (из архива РИА Новости)

ключеческих фильмах, наивно полагая, что они возможны только в сказочных далёких пустынях. Дело в том, что у башен всего по три стены и нет крыш. Как же так? Ещё не догадался? Ушканойские сторожевые башни «встроены» в скалистую гору Селин-Лам, и, кажется, что их создал не человек, а сама природа. Однако такого рода дозорные башни можно часто встретить в этом районе, порой даже ещё более интересные — со всего одной рукотворной стеной.

С появлением воздушного оружия сторожевым башням значительно прибавилось работы. Теперь нужно было следить

#### Кстати!

Сонар — это сокращение словосочетания «Sound Navigation and Ranging» — звуковая навигация и определение дальности, то есть устройство гидролокации.



Новейшие башни и системы занимаются даже противокосмической обороной. Их современейшее оборудование позволяет производить мониторинг сразу по десяткам, а то и сотням параметров, с минимальной погрешностью позволяя обнаружить врага за сотни километров от охраняемого объекта.

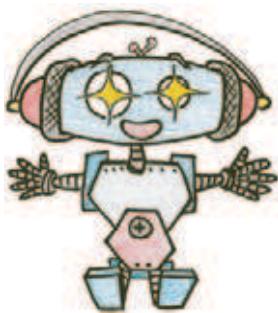


**Рис. 5.** Современное оборудование обнаружения

А сможешь ли ты построить современную сторожевую башню, оснащённую мощным оружием и всевидящим оком? Вопрос риторический: враг уже здесь! Видишь, как соседский кот смотрит на твою канарейку?!

Вперёд, дозорный!

# Этап 1. Устройство сторожевой башни



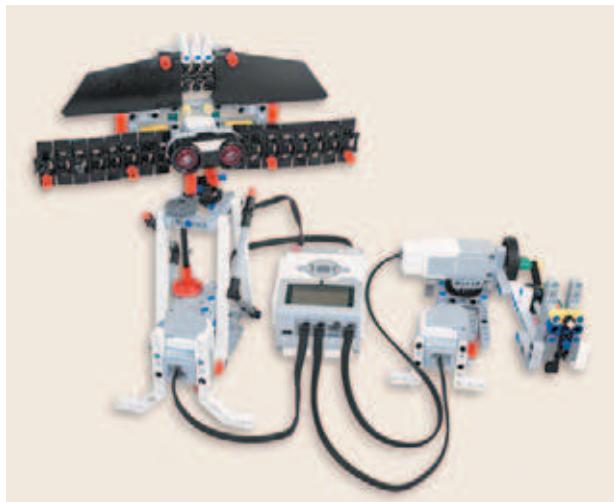
Рассмотри модель сторожевой башни и турель с пулемётом, собранную на основе набора LEGO® MINDSTORMS® Education EV3.

Попробуй выделить в них рабочие детали — модуль обнаружения, расположенный на верхушке башни; поворотный механизм, обеспечивающий вращение модуля обнаружения; датчик касания в нижней части

башни для калибровки угла поворота; неподвижную турель (основание для крепления пулемёта); пулемёт с магазином. Как ты думаешь, как будут подаваться патроны?

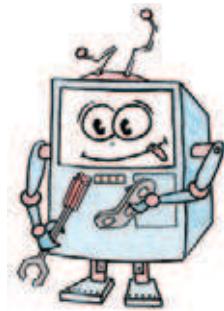
Обрати внимание, на вклейке в таблице в конце проекта даны все детали, которые потребуются тебе для сборки. Эта таблица поможет быстро найти то, что необходимо, и не ошибиться при конструировании.

Вот так выглядит сторожевая башня с огневой поддержкой, которые тебе предстоит собрать (рис. 6).



**Рис. 6.** Сторожевая башня с огневой поддержкой

# Этап 2. Сборка системы охраны периметра



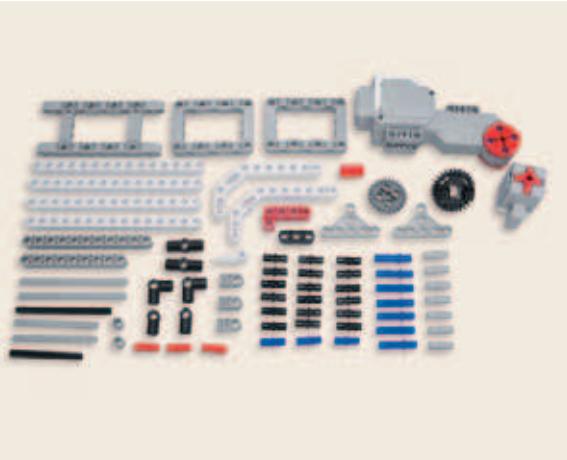
## ШАГ 1. СБОРКА БАШНИ С ПРИВОДОМ РАДАРА

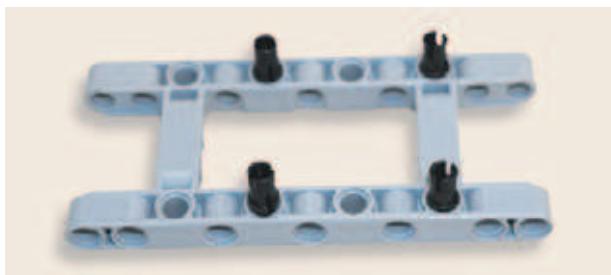
Детали для сборки:

- рамка  $5 \times 11$ , серая, 1x;
- рамка  $5 \times 7$ , серая, 2x;
- поворотный стол (серое вращающееся днище и чёрная вращающаяся верхушка) в сборе, 1x;
- балка № 3, чёрная, 1x;
- балка № 9, серая, 1x;
- балка № 11, серая, 1x;
- балка № 15, белая, 4x;
- балка двойная угловая  $3 \times 7$ , белая, 2x;
- балка прямоугольная  $2 \times 4$ , красная, 1x;
- ось № 2, красная, 3x;
- ось № 6, чёрная, 1x;
- ось № 7, серая, 2x;
- ось № 8, чёрная, 1x;
- ось № 9, серая, 2x;
- втулка — удлинитель оси, 2-модульная, красная, 1x;
- втулка, 1-модульная, серая, 2x;
- поперечный блок, 2-модульный, серый, 3x;
- треугольный фиксатор  $3 \times 5$ , серый, 2x;
- зубчатое колесо на 24 зуба, тёмно-серое, 1x;
- угловой блок 1,  $0^\circ$ , чёрный, 2x;

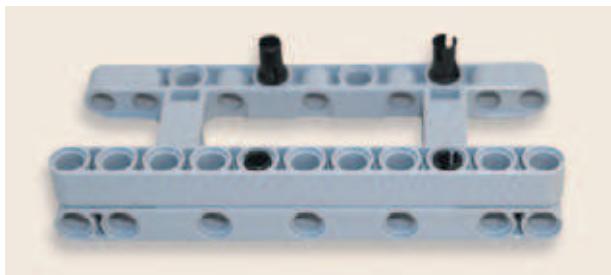


- угловой блок 2,  $180^\circ$ , чёрный, 2x;
- угловой блок 6,  $90^\circ$ , чёрный, 2x;
- стрелка, 3-модульная, белая, 1x;
- соединительный штифт, 2-модульный, чёрный, 18x;
- соединительный штифт, 3-модульный, синий, 7x;
- соединительный штифт, 2-модульный, серый, 7x;
- соединительный штифт, 2-модульный, синий, 3x;
- большой мотор, 1x;
- датчик касания, 1x.

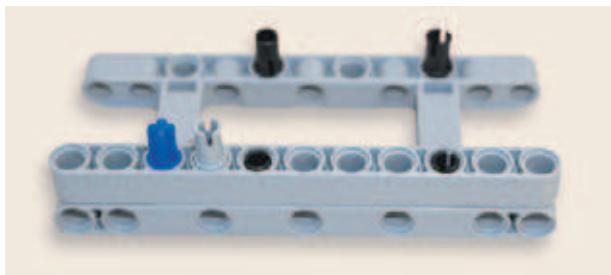




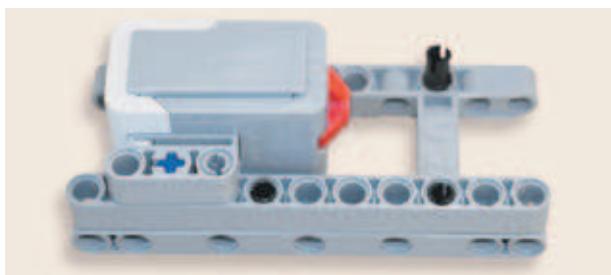
1. Положи серую рамку  $5 \times 11$  длинной стороной к себе. Вставь в её пятые и девятые модули по длиной стороне по одному чёрному штифту.



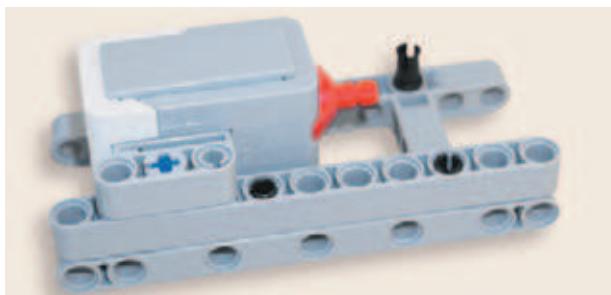
2. Надень сверху на свободные концы штифтов на длинной стороне серой рамки, расположенной ближе к тебе, серую балку № 11 пятым и девятым модулями.



3. Вставь в третий модуль серой балки № 11 синий 2-модульный штифт, а в четвёртый модуль этой же балки — серый 2-модульный штифт.



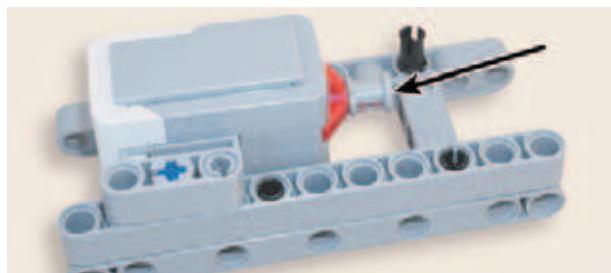
4. Надень на свободные концы штифтов датчик касания вторым и третьим модулями так, чтобы разъём находился слева, а кнопка — внутри серой рамки.



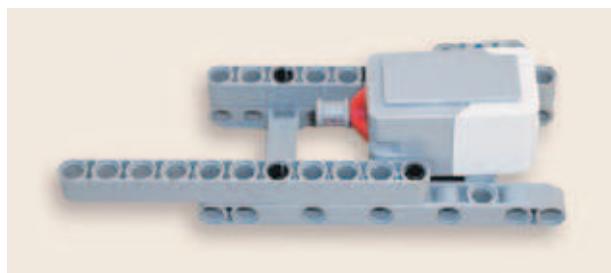
5. Вставь красную ось № 3 в гнездо датчика касания, находящееся на тактовой кнопке.



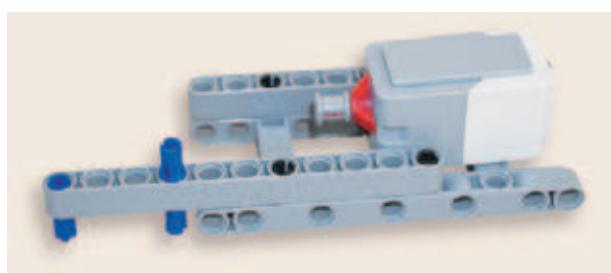
6. Надень на свободный конец оси серую 1-модульную втулку.



7. Теперь разверни рамку противоположной длинной стороной к себе и надень на свободные концы чёрных штифтов серую балку № 11 седьмым и одиннадцатым модулями, как показано на рисунке.



8. Вставь синий 2-модульный штифт снизу в первый модуль серой балки № 11, а в четвёртый модуль — синий 3-модульный штифт длинным концом вниз.



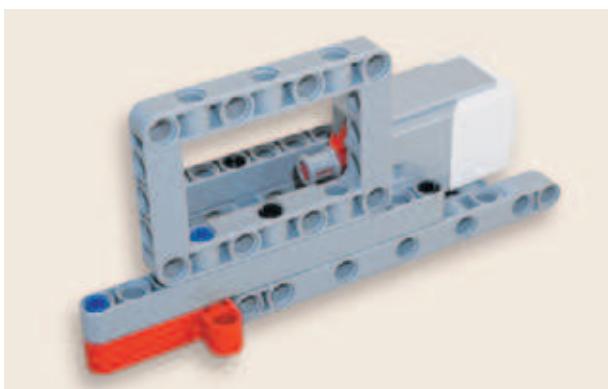
9. Надень снизу на свободные концы синих модулей красную прямоугольную балку  $2 \times 4$  так, чтобы короткая сторона красной балки находилась снаружи серой рамки и располагалась ближе к датчику касания.



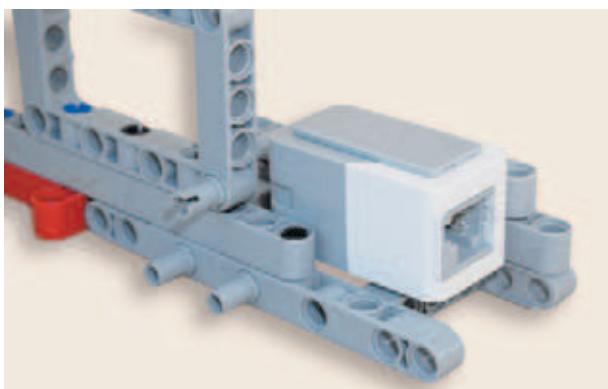
10. Вставь чёрный 2-модульный штифт в балку № 11 в шестой модуль.



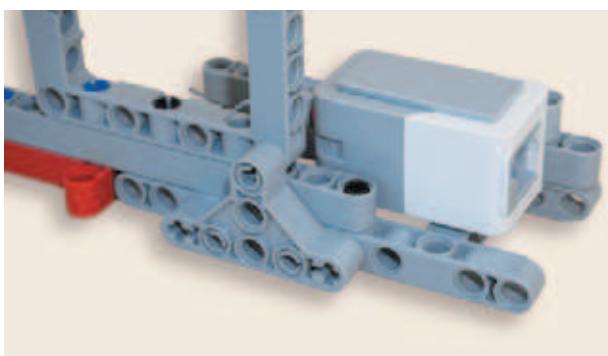
11. Надень серую рамку  $5 \times 7$  длинной стороной на свободные концы штифтов вторым и четвёртым модулями.



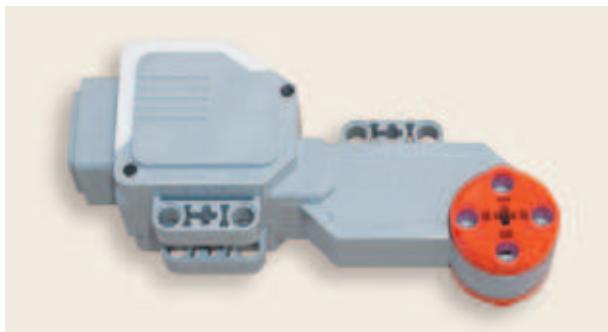
12. Теперь вставь серый штифт снизу в седьмой модуль серой рамки  $5 \times 7$ . Ещё вставь два серых штифта в длинную сторону серой рамки  $5 \times 11$  в четвёртый и шестой модули, как показано на рисунке.



13. Надень друг на друга на свободные концы серых штифтов два серых треугольных фиксатора.



14. Поверни большой мотор так, чтобы разъём располагался слева, а крепление возле привода — ближе к тебе.



15. Вставь чёрную ось № 6 в ступицу мотора насквозь.



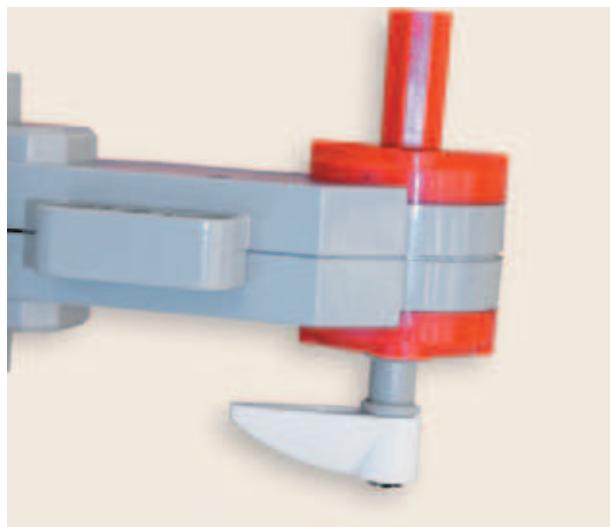
16. Надень на свободный конец оси, расположенный ближе к тебе, синюю 1-модульную втулку.

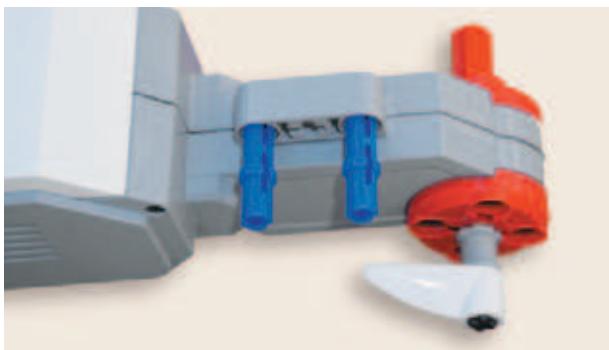


17. Надень на ось после втулки белую 3-модульную стрелку.



18. Надень на свободный конец чёрной оси с другой стороны от привода красную 2-модульную втулку (удлинитель оси).

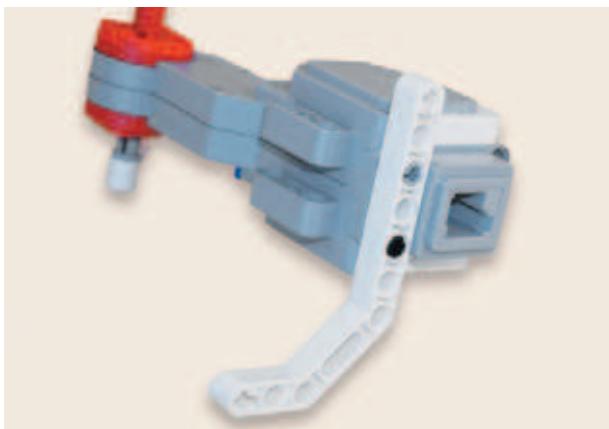




19. Вставь два синих 3-модульных штифта в крепление возле привода длинными концами в первый и третий модули. Таким образом, когда ты их задвинешь, штифты будут видны по обе стороны от крепления.



20. Теперь поверни большой мотор разъёмом к себе и вставь в ближнее к тебе крепление возле разъёма в верхний модуль серый, а в нижний чёрный 2-модульные штифты, как показано на рисунке.



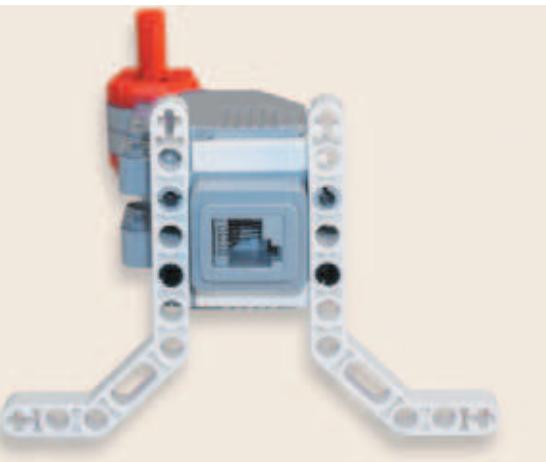
21. Надень на свободные концы серого и чёрного штифтов белую двойную угловую балку  $3 \times 7$  третьим и пятым модулями её длинной стороны так, чтобы короткая сторона была направлена к тебе.



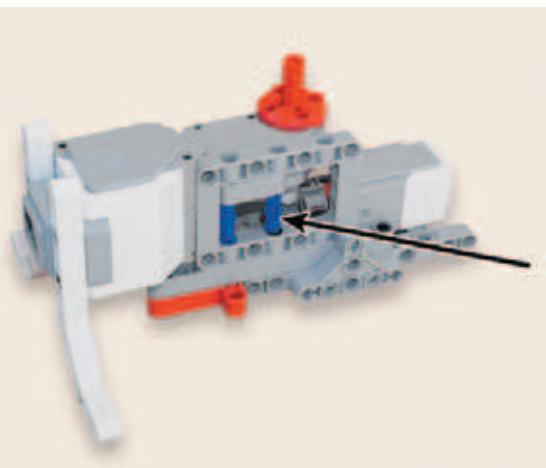
22. Теперь вставь один серый и один чёрный 2-модульные штифты в свободное крепление возле разъёма большого мотора в первый и третий модули.



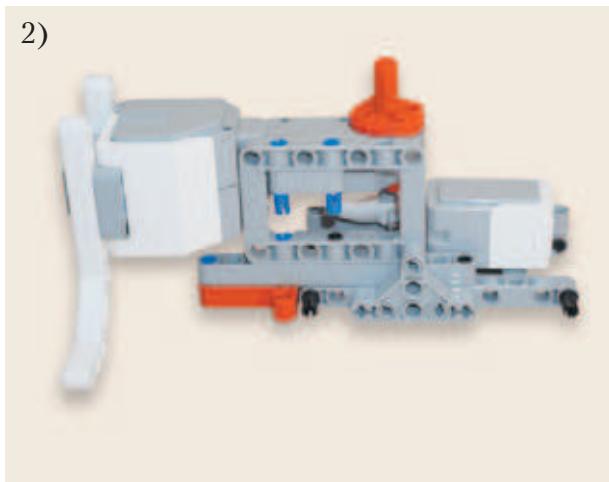
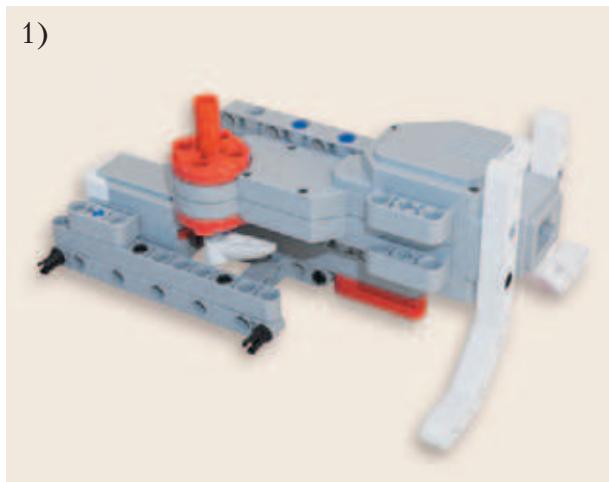
23. Разверни конструкцию, как показано на рисунке. Надень сверху на свободные концы серых штифтов белую двойную угловую балку  $3 \times 7$  третьим и пятым модулями её длинной стороны так, чтобы короткая сторона угловой балки находилась внизу справа.



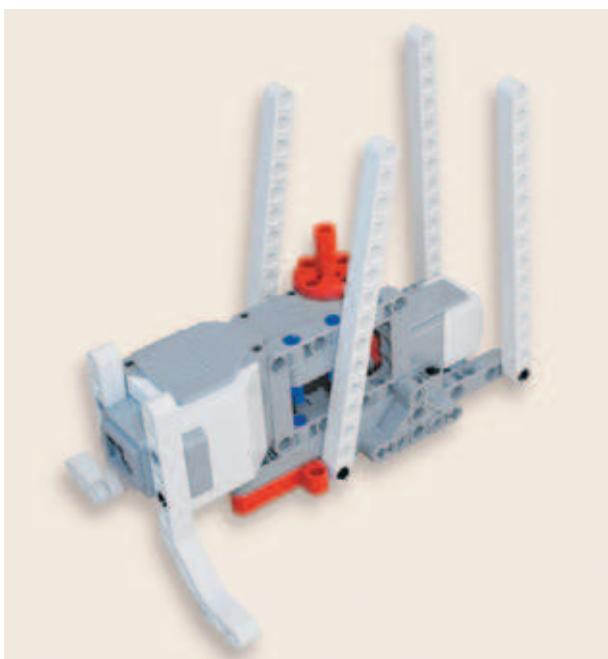
24. Разверни подготовленное основание для башни вертикальной серой рамкой  $5 \times 7$  к себе. Вложи большой мотор внутрь серой рамки таким образом, чтобы свободные концы синих штифтов в креплении возле привода вошли во второй и четвёртый модули серой рамки изнутри. При возникновении трудностей внимательно изучи рисунок.



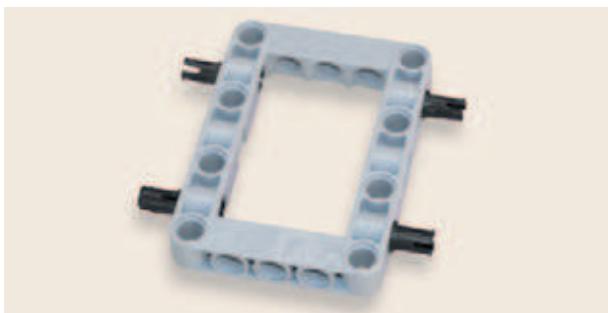
25. Возьми четыре чёрных штифта и вставь их по одному в крайние модули серой рамки  $5 \times 11$  с внешней стороны.



26. Надень на свободные концы чёрных штифтов четыре белые балки № 15 первыми модулями.



27. Вставь четыре чёрных штифта во вторые и шестые модули длинных сторон серой рамки 5×7 с обеих сторон.



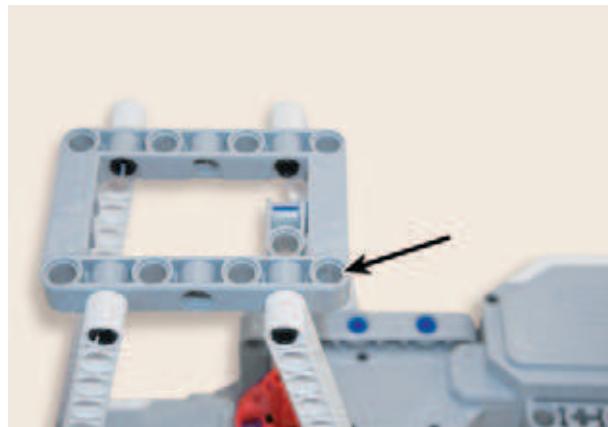
28. Надень на четыре свободных конца чёрных штифтов рамки последние модули белых балок № 15.



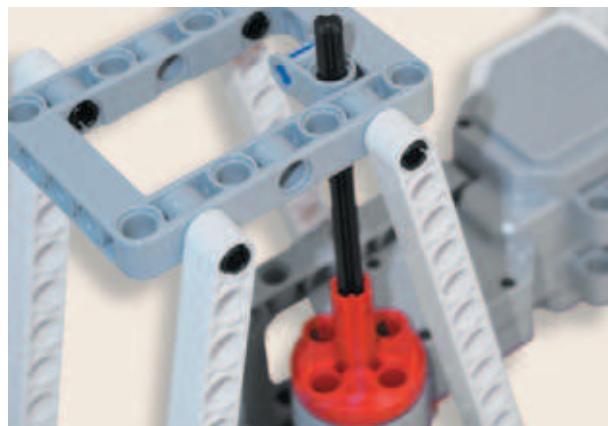
29. Вставь в крестовое отверстие 2-модульного поперечного блока один синий 2-модульный штифт.



30. Разверни собранную конструкцию рамки башни так, чтобы верхняя серая рамка находилась слева, а разъём большого мотора — снизу справа. Теперь вставь подготовленный серый блок свободным концом синего штифта во второй модуль верхней серой рамки  $5 \times 7$  изнутри правой короткой стороны. Далее поверни серый блок так, чтобы свободный модуль на сером поперечном блоке располагался напротив первого модуля короткой стороны серой рамки  $5 \times 7$ .



31. Продень чёрную ось № 8 в свободный модуль поперечного блока и вставь конец оси в красную втулку-удлинитель до упора.



32. Вставь в модули серой стороны собранного поворотного стола по одному синему 3-модульному штифту короткими концами.



33. Надень на свободные концы синих штифтов чёрную балку № 3 крайними модулями.



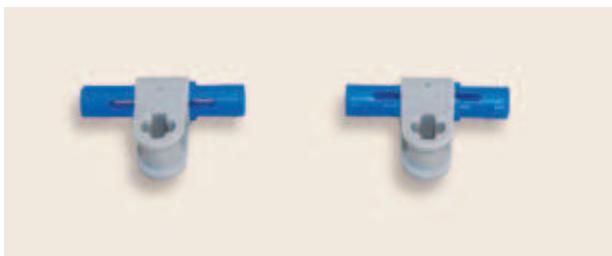
34. Вставь поворотный стол свободными концами синих штифтов в короткую сторону верхней рамки слева так, чтобы чёрная часть поворотного стола находилась сверху рамки.



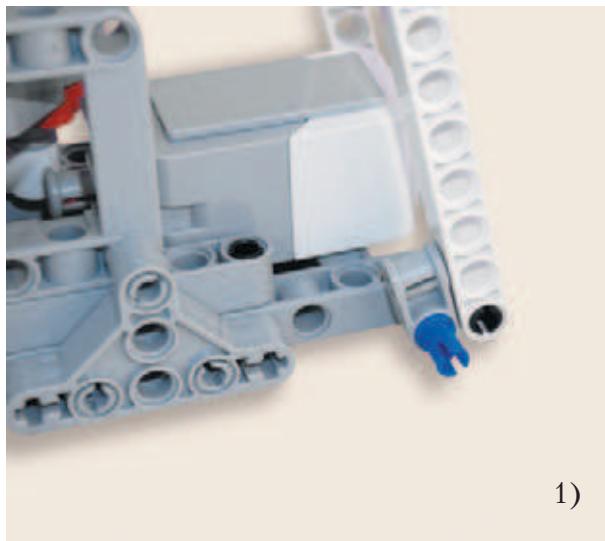
35. Надень тёмно-серое зубчатое колесо на 24 зуба на свободный конец чёрной оси поверх серой рамки.



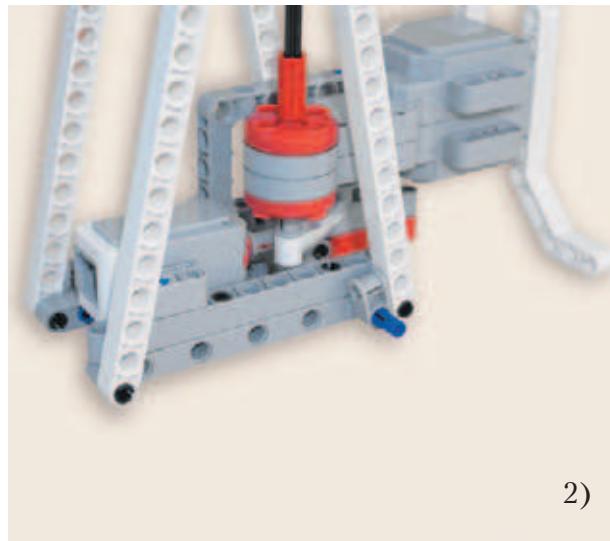
36. Вставь в два серых 2-модульных блока длинными концами по одному синему 3-модульному штифту.



37. Установи получившиеся элементы с обеих сторон в нижнюю серую рамку  $5 \times 11$  в предпоследние модули.



1)



2)

38. Теперь надень на свободные концы двух серых осей № 9 с одной стороны по одному угловому блоку 1,  $0^\circ$ .



39. Надень на противоположные концы осей по одному угловому блоку 2,  $90^\circ$ . Свободные модули блоков должны быть повернуты в свою сторону.



40. Вставь в свободные модули угловых блоков 2, 90° по одной серой оси № 7.

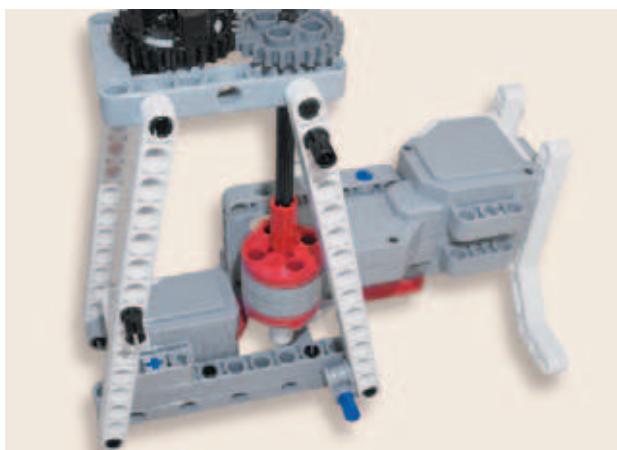


41. Надень на свободные концы серых осей № 7 по одному угловому блоку 6, 180°.

**Важно!** При установке угловых блоков обрати внимание на модули этих блоков. Отверстия модулей должны располагаться лицом к тебе.



42. Вставь в свободные модули угловых блоков 6, 180° по одной красной оси № 3.



43. Вставь один чёрный штифт в восьмой модуль белой балки № 15, расположенной слева. Ещё один чёрный штифт вставь во второй модуль белой балки № 15, расположенной ближе к большому мотору.



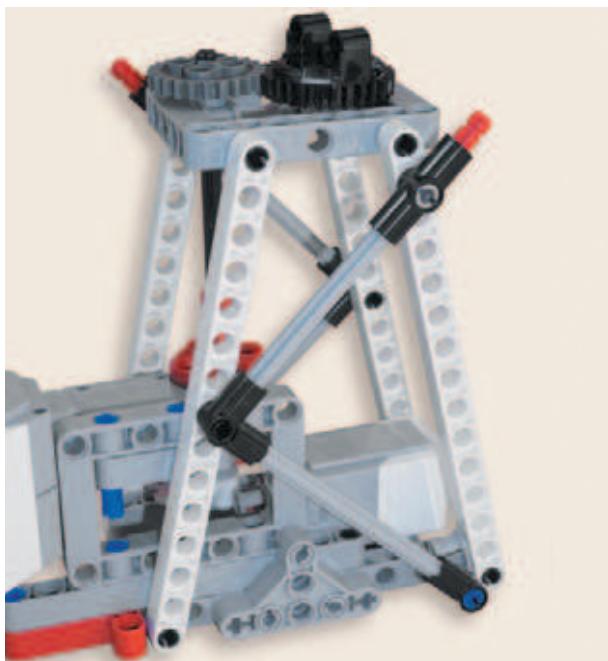
44. Надень собранную конструкцию осей с угловыми блоками свободными модулями угловых блоков на синий и два чёрных штифта, как показано на рисунке.



45. Переверни конструкцию башни так, чтобы разъём большого мотора находился слева, а датчик касания — справа. Вставь чёрный 2-модульный штифт в восьмой модуль белой балки № 15 слева, а серый 2-модульный штифт — во второй модуль белой балки № 15, расположенной справа.



46. Аналогично надень вторую конструкцию с осями с противоположной стороны башни.



Отлично! Ты хорошо постарался. Пусть шаг был долгим, но зато ты собрал красивую башню. Пора переходить к сборке турели.

## ШАГ 2. СБОРКА ПУШКИ



Детали для сборки:

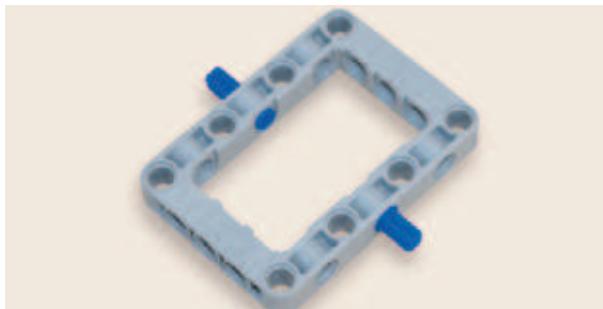
- балка № 3, красная, 1x;
- балка № 9, серая, 1x;
- балка угловая 3×7, серая, 2x;
- балка прямоугольная 3×5, белая, 2x;
- балка прямоугольная 2×4, красная, 1x;
- ось № 6, чёрная, 1x;



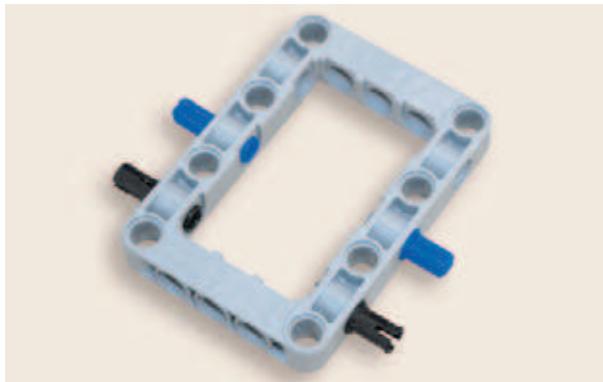
- большой мотор, 1x;
- рамка 5×7, серая, 1x;
- зубчатое колесо на 24 зуба, тёмно-серое, 1x;
- поворотный стол (серое вращающееся днище и чёрная вращающаяся верхушка) в сборе, 1x;
- втулка, 1-модульная, серая, 1x;
- соединительный штифт, 2-модульный, чёрный, 9x;
- соединительный штифт, 3-модульный, синий, 3x;
- соединительный штифт, 3-модульный, бежевый, 3x;
- соединительный штифт, 2-модульный, синий, 2x.



1. Вставь два синих 2-модульных штифта в четвёртые модули длинных сторон серой рамки  $5 \times 7$ .



2. Теперь вставь два чёрных штифта во вторые модули длинных сторон серой рамки.

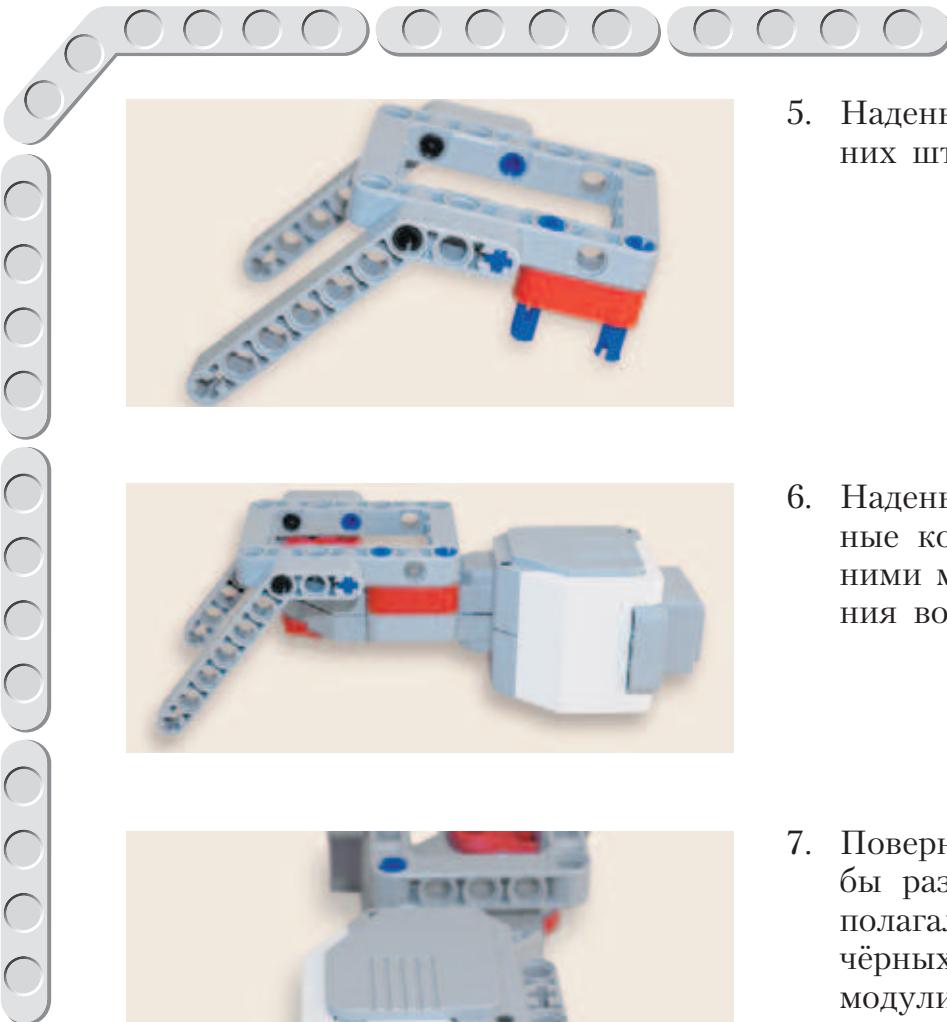


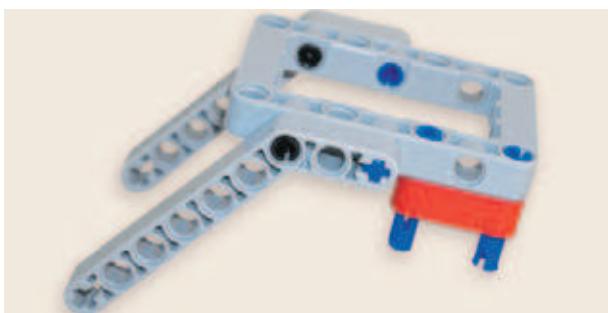
3. Надень на свободные концы синих и чёрных штифтов две серые угловые балки  $3 \times 7$  короткими сторонами так, чтобы длинные стороны серых балок были направлены вниз.



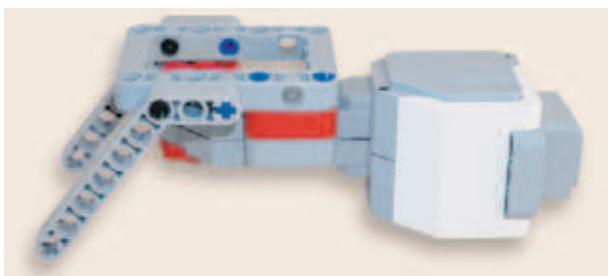
4. Вставь два синих 3-модульных штифта снизу в пятый и седьмой модули серой рамки короткими концами.



- 
5. Надень на свободные концы синих штифтов красную балку № 3.



6. Надень большой мотор на свободные концы синих штифтов крайними модулями верхнего крепления возле привода.



7. Поверни конструкцию так, чтобы разъём большого мотора располагался к тебе. Вставь четыре чёрных штифта по два в крайние модули возле разъёма большого мотора.



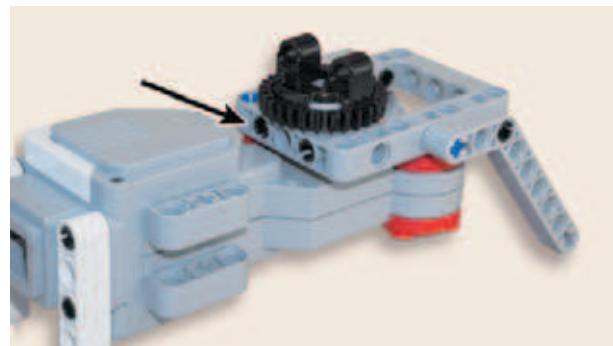
8. Надень две белые прямоугольные балки  $3 \times 5$  на свободные концы чёрных штифтов первыми и третьими модулями длинных сторон так, чтобы короткие стороны белых балок находились снизу и были направлены от мотора.



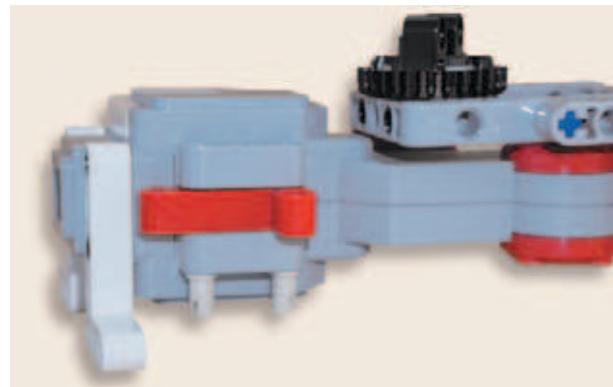
9. Вставь два чёрных штифта в се-  
роё основание поворотного стола.



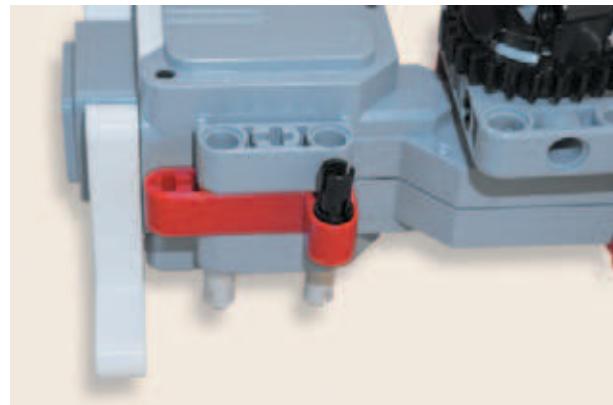
10. Переверни конструкцию так, что-  
бы привод большого мотора на-  
ходился справа, а разъём — сле-  
ва. Далее вставь поворотный стол  
свободными концами чёрных  
штифтов в левую, короткую сто-  
рону серой рамки изнутри.



11. Вложи красную прямоугольную  
балку в крепление возле разъёма  
так, чтобы короткая сторона крас-  
ной балки находилась справа. За-  
фиксируй её в креплении снизу с  
помощью двух бежевых штифтов,  
установив их во второй и четвёр-  
тый модули красной прямоуголь-  
ной балки.



12. Вставь сверху в свободный мо-  
дуль короткой стороны красной  
балки один чёрный штифт.



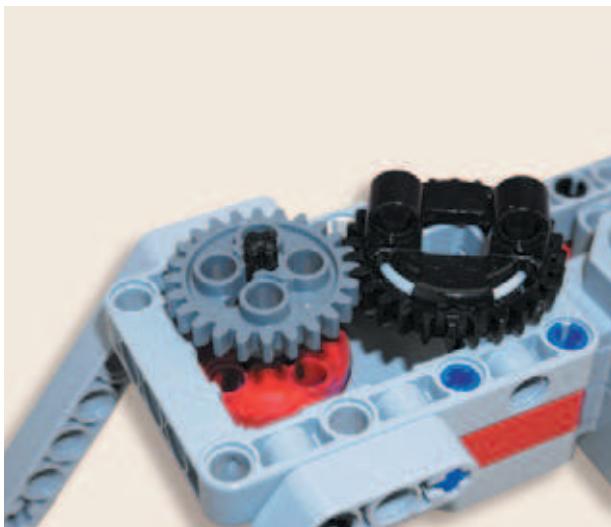
13. Надень серую балку № 9 четвёртым модулем на свободный конец чёрного штифта. Вставь синий и бежевый 3-модульные штифты длинными концами снизу в седьмой и девятый модули серой балки № 9. Они пройдут насквозь первый и третий модули рамки.



14. Вставь чёрную ось № 6 в ступицу большого мотора. Надень сверху на свободный конец оси серую втулку до упора.



15. Надень сверху на ось серое зубчатое колесо на 24 зуба.



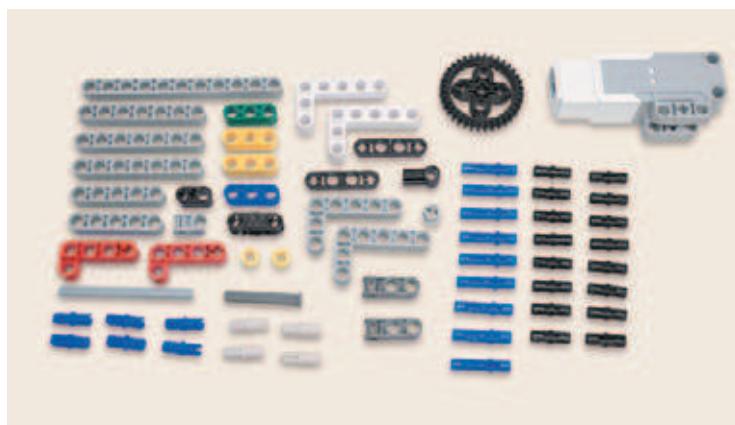
### ШАГ 3. СБОРКА ОРУДИЯ

Детали для сборки:

- балка № 11, серая, 1x;
- балка № 7, серая, 3x;
- балка № 5, серая, 2x;
- балка № 3, жёлтая, 2x;
- балка № 3, зелёная, 1x;
- балка № 3, синяя, 1x;
- балка прямоугольная 3×5, белая, 2x;
- балка прямоугольная 3×5, серая, 2x;
- балка прямоугольная 2×4, красная, 2x;
- двойной поперечный блок, 3-модульный, чёрный, 1x;
- угловой блок 1, 0°, чёрный, 1x;
- балка  $\frac{1}{2}$  № 4, чёрная, 2x;
- ось № 7, серая, 1x;
- ось № 4 с головкой, тёмно-серая, 1x;
- балка № 2 с поперечным отверстием, чёрная, 1x;



- поперечный блок, 2-модульный, серый, 1x;
- поперечный блок, 3-модульный, тёмно-серый, 2x;
- втулка,  $\frac{1}{2}$ -модульная, жёлтая, 2x;
- втулка, 1-модульная, серая, 1x;
- средний мотор, 1x;
- двойное коническое зубчатое колесо на 36 зубьев, чёрное, 1x;
- соединительный штифт, 2-модульный, чёрный, 16x;
- соединительный штифт, 2-модульный, синий, 6x;
- соединительный штифт, 3-модульный, синий, 9x;
- соединительный штифт с осью, 2-модульный, бежевый, 4x.



1. Поверни средний мотор к себе приводом так, чтобы крепления находились снизу. Вставь два чёрных штифта в вертикальные гнёзда у привода.



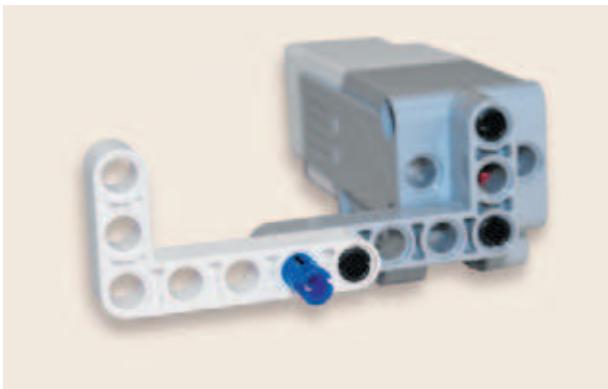
2. Надень на свободные концы чёрных штифтов серую прямоугольную балку  $3 \times 5$  короткой стороной так, чтобы длинная сторона серой балки была направлена влево от нижнего модуля.



3. Вставь синий 3-модульный штифт коротким концом в первый модуль длинной стороны прямоугольной балки. Затем вставь чёрный штифт во второй модуль прямоугольной балки.



4. Надень белую прямоугольную балку  $3 \times 5$  длинной стороной в пятый модуль на конец чёрного штифта, а четвёртым модулем — на конец синего штифта, как показано на рисунке.



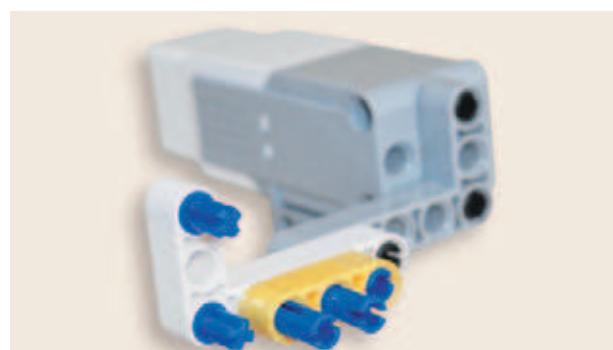
5. Вставь два синих 3-модульных штифта короткими концами во второй и третий модули длинной стороны белой балки.



6. Надень сверху на свободные концы синих штифтов во втором, третьем и четвёртом модулях жёлтую балку № 3 первым, вторым и третьим модулями.



7. Установи два синих 2-модульных штифта в первый и третий модули короткой стороны белой прямоугольной балки.



8. Надень на свободные концы синих штифтов чёрный двойной перечный 3-модульный блок.



9. Вставь в первый модуль зелёной балки № 3 бежевый 2-модульный штифт, а в третий модуль вставь до упора тёмно-серую ось № 4 с головкой.



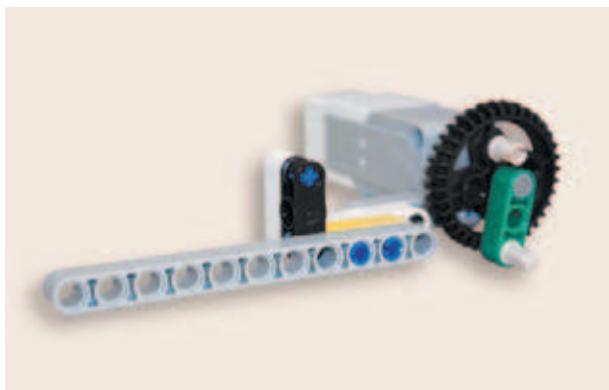
10. Вставь в центральное крестовое отверстие двойного конического колеса на 36 зубьев свободный конец тёмно-серой оси, а в левое крестовое отверстие — бежевый 2-модульный штифт.



11. Вставь свободный конец серой оси подготовленной детали в привод среднего мотора.



12. Теперь надень серую балку № 11 девятым и десятым модулями на свободные концы синих штифтов.



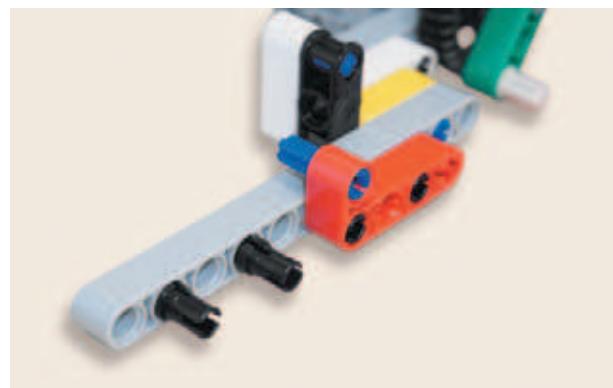
13. Установи четыре чёрных штифта во второй, четвёртый, шестой и восьмой модули серой балки № 11.



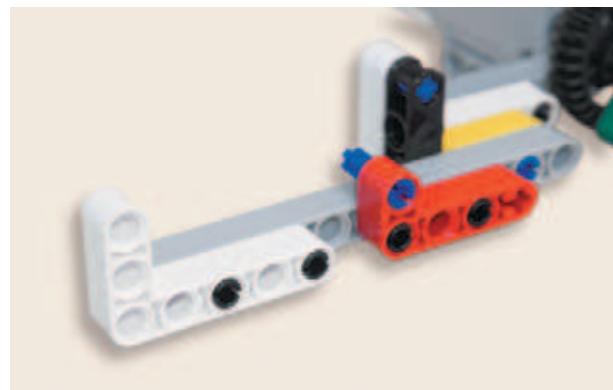
14. Надень на свободные концы чёрных штифтов в шестом и восьмом модулях серой балки красную прямоугольную балку  $2 \times 4$  длинной стороной первым и третьим модулями так, чтобы короткая сторона красной балки находилась слева вверху.



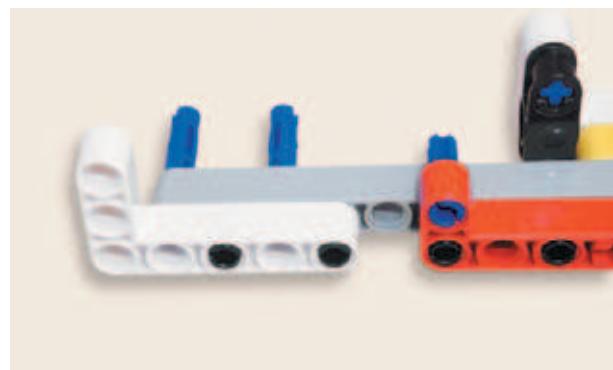
15. Вставь в первый модуль короткой стороны красной балки один синий 2-модульный штифт, как показано на рисунке.



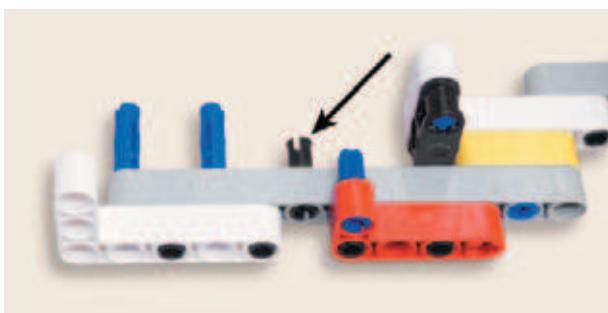
16. Надень на концы чёрных штифтов белую прямоугольную балку  $3 \times 5$  третьим и пятым модулями так, чтобы короткая сторона белой балки находилась слева вверху.



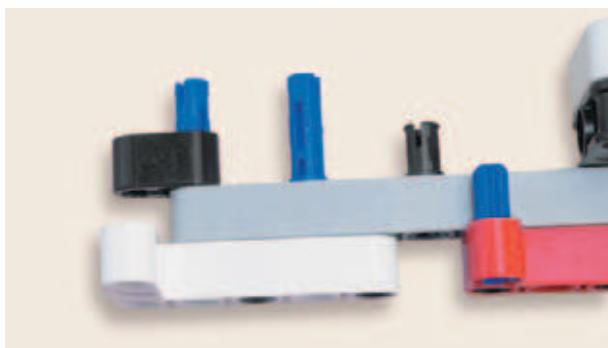
17. Вставь короткими концами два синих 3-модульных штифта в первый и третий модули серой балки № 11.



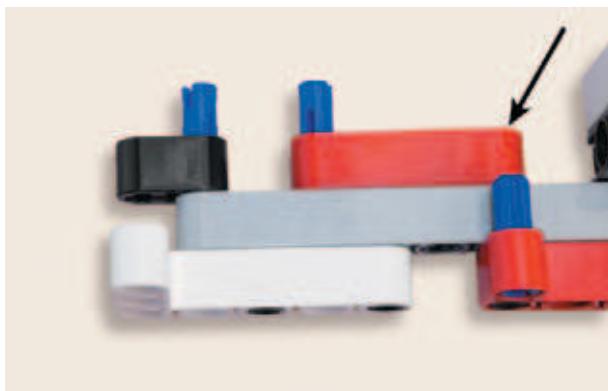
18. Вставь в пятый модуль серой балки № 11 один чёрный штифт.



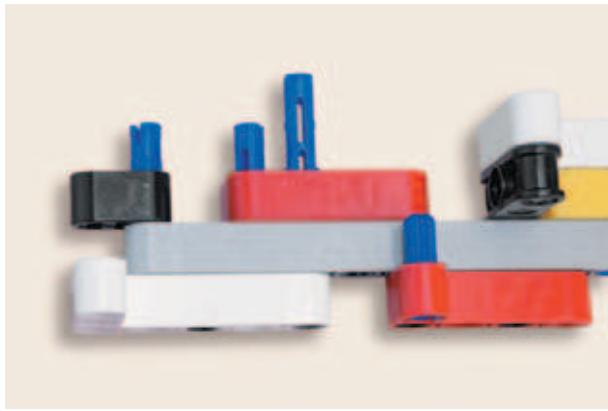
19. Надень до упора на свободный конец синего штифта в первом модуле серой балки чёрную балку № 2 с поперечным отверстием.



20. Установи красную прямоугольную балку  $2 \times 4$  на свободные концы синего и чёрного штифтов длинной стороной так, чтобы короткая сторона красной балки была на левой стороне и направлена вниз.



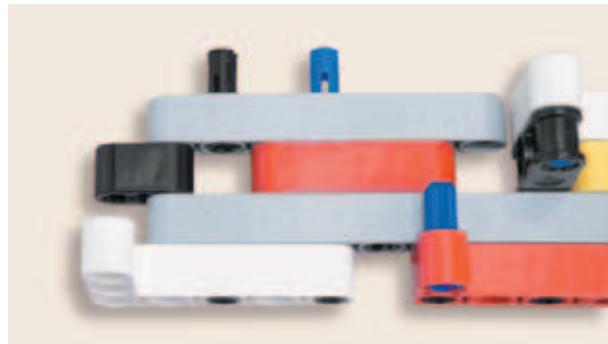
21. Вставь коротким концом во второй модуль длиной стороны красной балки один синий 3-модульный штифт.



22. Надень на свободные концы синих штифтов серую балку № 7 первым, третьим и четвёртым модулями.



23. Вставь во второй модуль серой балки № 7 один чёрный штифт.



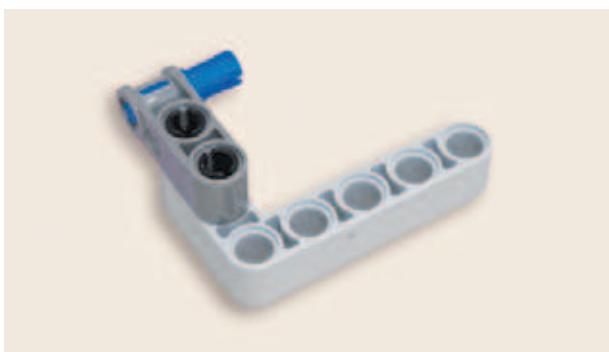
24. Расположи перед собой серую прямоугольную балку  $3 \times 5$  так, чтобы короткая сторона балки была с твоей стороны и направлена влево. Вставь в первый и второй модули короткой стороны два чёрных штифта.



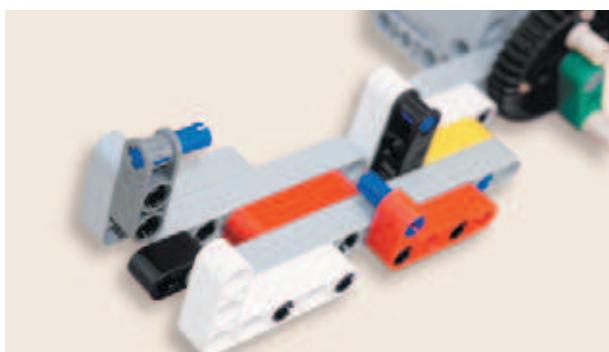
25. Надень на свободные концы чёрных штифтов вторым и третьим модулями тёмно-серый поперечный 3-модульный блок.



26. Установи в первый модуль тёмно-серого блока один синий 2-модульный штифт.



27. Установи на свободные концы чёрного и синего штифтов подготовленную серую прямоугольную балку  $3 \times 5$  третьим и пятым модулями длинной стороны.



28. Вставь чёрный штифт во второй модуль короткой стороны белой прямоугольной балки  $3 \times 5$ .



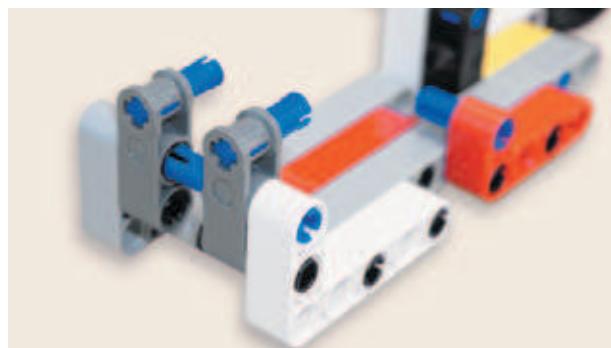
29. Вставь синий 3-модульный штифт в первый модуль короткой стороны белой прямоугольной балки.



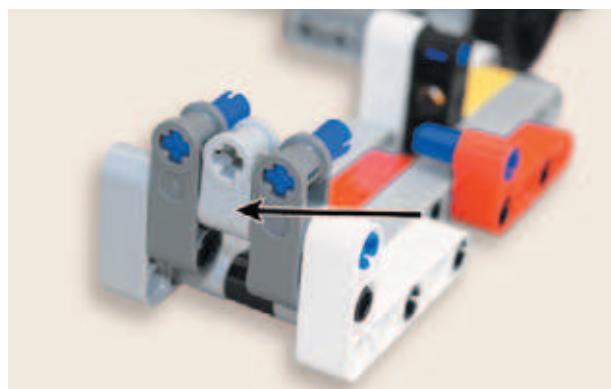
30. Надень на свободные концы синего и чёрного штифтов вторым и третьим модулями тёмно-серый поперечный 3-модульный блок.



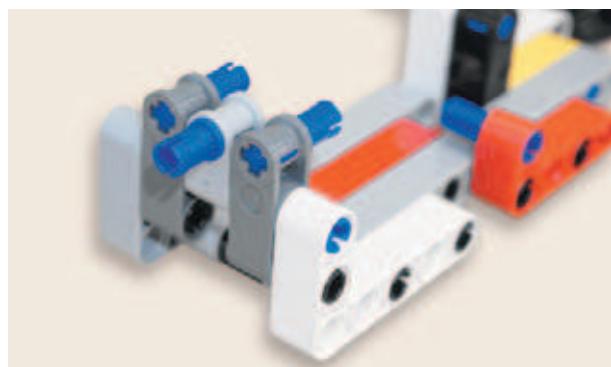
31. Вставь в первый модуль тёмно-серого блока один синий 2-модульный штифт.



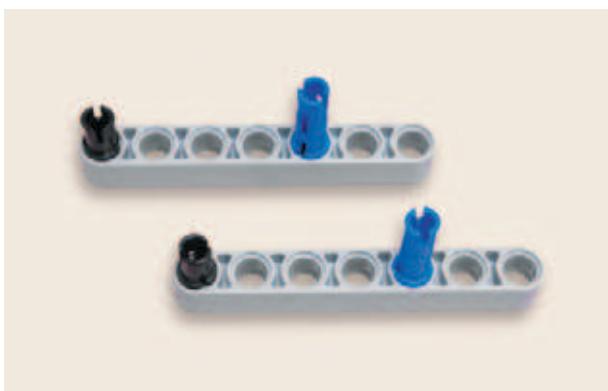
32. Надень на свободный конец синего штифта в тёмно-сером блоке серый поперечный 2-модульный блок вторым модулем. (Предварительно потребуется отсоединить белую прямоугольную балку.)



33. Вставь синий 2-модульный штифт в первый модуль серого 2-модульного блока.



34. Вставь по одному чёрному штифту в первые модули двух серых балок № 7, а в пятые модули — по одному синему 3-модульному штифту короткими концами.



35. Установи две серые балки № 5 на свободные концы чёрных и синих штифтов.



36. Вставь в четвёртые модули серых балок № 5 по одному чёрному штифту.



37. Надень на свободные концы синих штифтов жёлтую балку № 3 крайними модулями, а на концы чёрных штифтов — синюю балку № 3 крайними модулями, соединив таким образом серые заготовки.



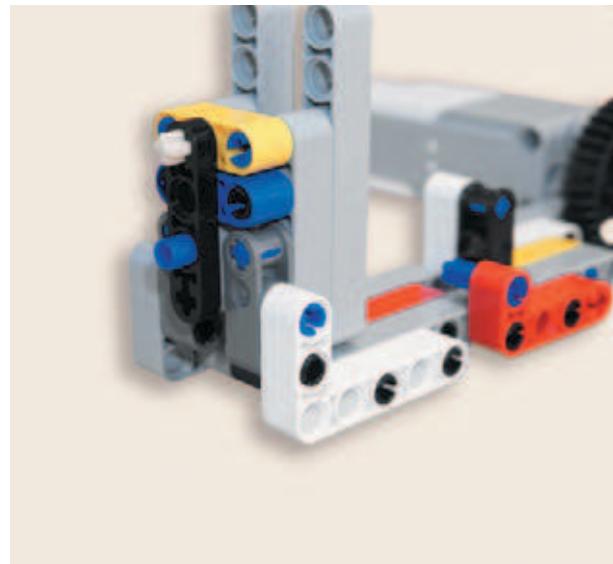
38. Вставь в средний модуль жёлтой балки один бежевый штифт.



39. Надень собранную конструкцию третьими модулями в серых балках № 5 на свободные концы синих штифтов в тёмно-серых блоках.



40. Установи на свободные концы бежевого и синего штифтов чёрную балку  $\frac{1}{2}$  № 4 первым и третьим модулями.

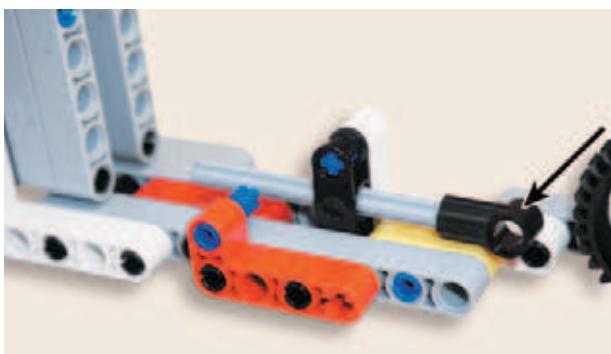


41. Теперь поверни конструкцию орудия так, чтобы дуло пулемёта находилось слева, а средний мотор — справа. Вставь серую ось № 7 в средний модуль чёрного двойного поперечного блока.

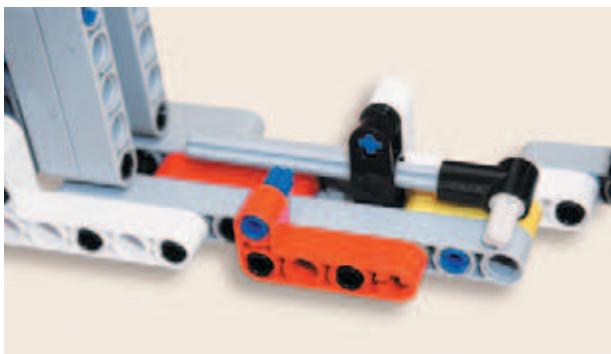


Осталось собрать спусковой механизм.

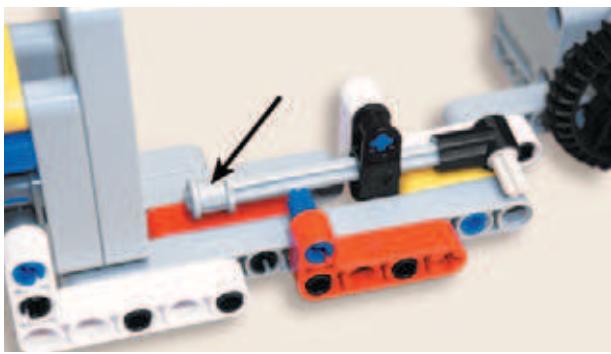
42. Надень на конец серой оси № 7 чёрный угловой блок 1,  $0^\circ$ .



43. Вставь в свободный модуль чёрного углового 1,  $0^\circ$  блока бежевый 2-модульный штифт.



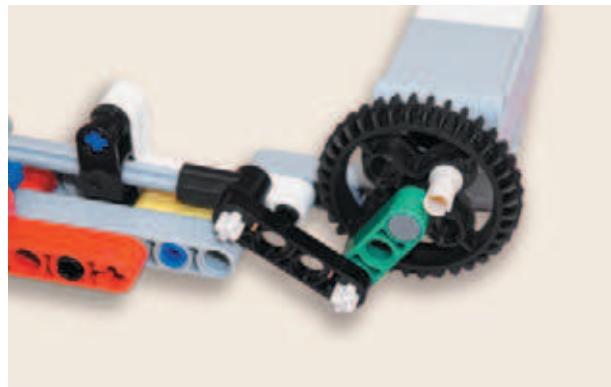
44. Надень на свободный конец серой оси слева серую 1-модульную втулку.



Тебе осталось соединить собранный спусковой механизм с приводом мотора.



45. Установи чёрную балку  $\frac{1}{2}$  № 4 крайними модулями на свободные концы бежевых штифтов в зелёной балке № 3 и чёрном угловом блоке 1,  $0^\circ$ .



46. Надень на свободные концы бежевых штифтов по одной жёлтой  $\frac{1}{2}$ -модульной втулке.



47. Надень резинку.



Ты молодец! Осталось совсем немного — собрать устройство обнаружения, а также украсить башню, чтобы она выглядела устрашающе.

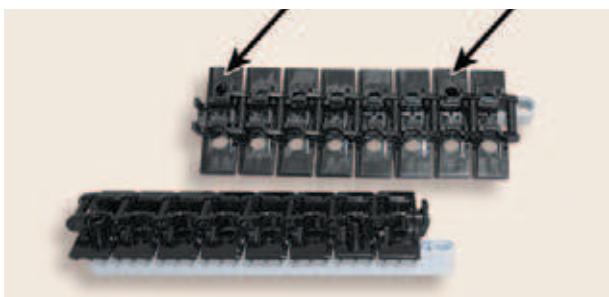
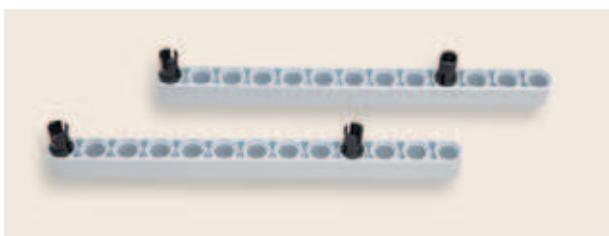
## ШАГ 4. СБОРКА ЛОКАТОРА

### Детали для сборки:

- балка № 13, серая, 3х;
- балка № 7, серая, 1х;
- балка № 5, серая, 2х;
- балка № 3, красная, 2х;
- балка № 3, зелёная, 2х;
- балка № 3, жёлтая, 2х;
- звено гусеницы, чёрное, 16х;
- ось № 7, серая, 1х;
- ультразвуковой датчик, 1х;
- балка прямоугольная 2×4, красная, 2х;
- зубчатое колесо на 4 зуба, чёрное, 3х;
- стрелка, 3-модульная, белая, 3х;
- втулка, 1-модульная, серая, 2х;
- втулка,  $\frac{1}{2}$ -модульная, жёлтая, 2х;
- угловой соединительный штифт, 3×3-модульный, серый, 2х;
- соединительный штифт, 2-модульный, чёрный, 15х;
- соединительный штифт, 2-модульный, синий, 6х;



- соединительный штифт, 2-модульный, бежевый, 2х;
- соединительный штифт, 3-модульный, синий, 7х;
- соединительный штифт со втулкой, 3-модульный, красный, 6х;
- правая криволинейная панель 5×11, чёрная, 1х;
- левая криволинейная панель 5×11, чёрная, 1х.



1. Вставь в две серые балки № 13 по одному чёрному 2-модульному штифту в первые модули и по одному — в десятые модули балок.
2. Собери две гусеницы по восемь звеньев. Теперь установи на свободные концы чёрных штифтов гусеницы первым и седьмым модулями в верхнем ряду.



3. Вставь по одному чёрному штифту в крепления возле разъёма ультразвукового датчика.



4. Надень подготовленные серые балки крайними модулями на свободные концы штифтов датчика, как показано на рисунке. На данный момент конструкции не хватает рёбер жёсткости, поэтому «крылья» локатора подвижны.

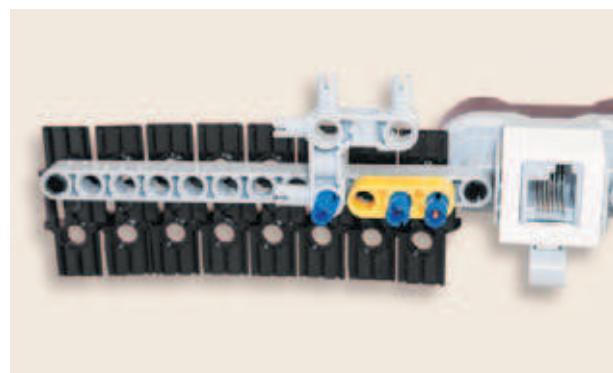
Конструкция будет достаточно тяжёлой и объёмной.



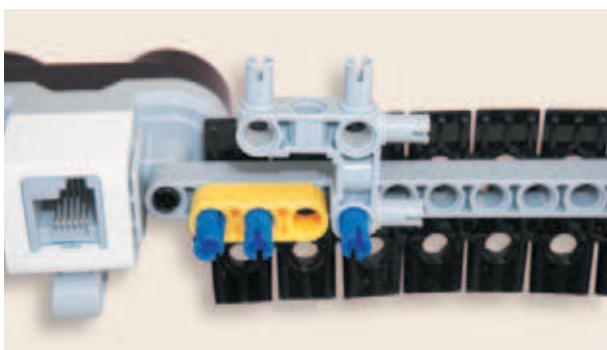
5. Вставь шесть синих 3-модульных штифта короткими концами в левую серую балку в девятый, одиннадцатый и двенадцатый модули, а в правую серую балку — во второй, третий и пятый модули.



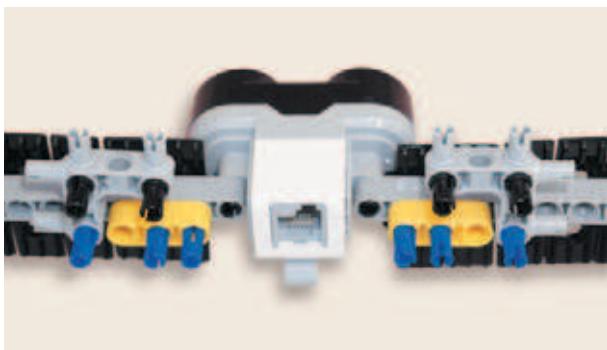
6. Надень на свободные концы синих штифтов левого «крыла» вторым и третьим модулями жёлтую балку № 3 до упора, а на конец штифта в девятом модуле — угловая соединительный 3×3-модульный штифт так, чтобы угловая часть находилась вверху и была направлена вправо.



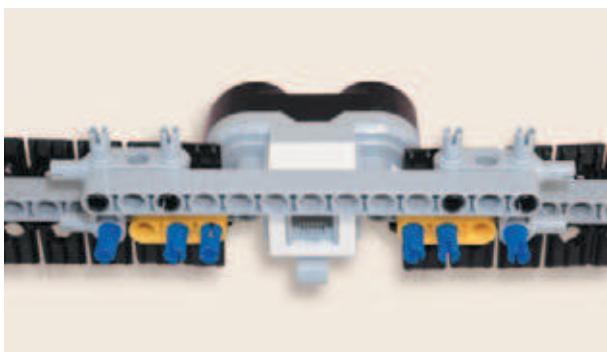
7. Повтори предыдущий пункт зеркально для правого «крыла».



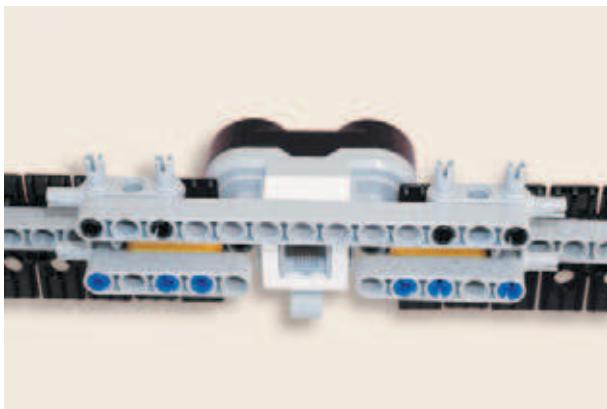
8. Вставь по одному чёрному 2-модульному штифту в горизонтальные модули угловых соединительных штифтов.



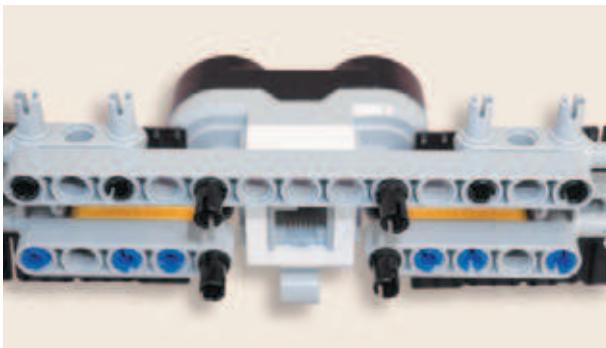
9. Добавь ребро жёсткости. Надень сверху на свободные концы чёрных штифтов серую балку № 13 первым, третьим, одиннадцатым и тринадцатым модулями.



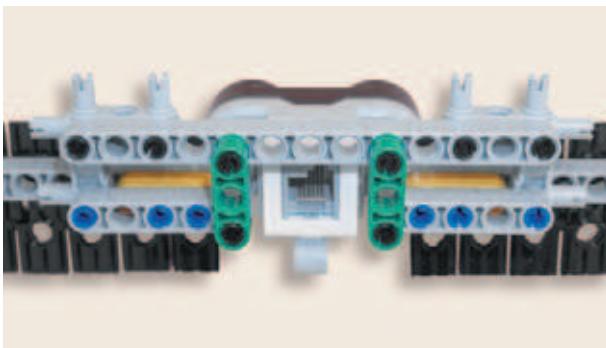
10. Установи одну серую балку № 5 на свободные концы синих штифтов в левой балке первым, третьим и четвёртым модулями, а вторую серую балку № 5 — на правую балку вторым, третьим и пятым модулями.



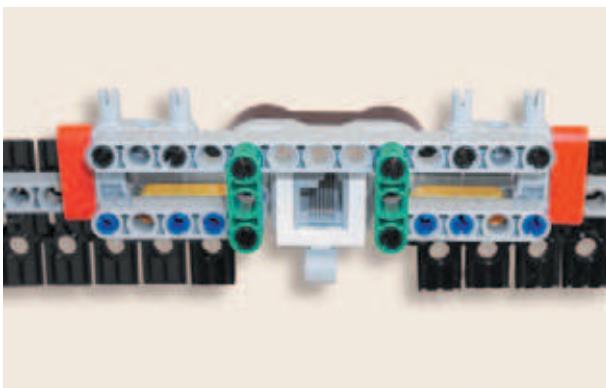
11. Вставь два чёрных штифта в серую балку № 13 в пятый и девятый модули. Далее установи ещё два чёрных штифта по одному в серые балки № 5 слева в пятый модуль, а справа — в первый модуль.



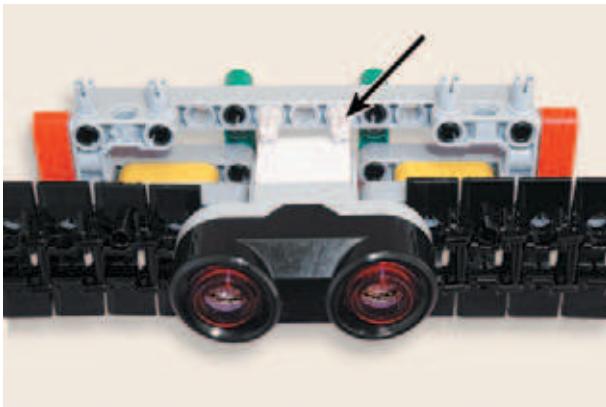
12. Надень на свободные концы чёрных штифтов в балках № 5 и в балке № 13 вертикально по одной зелёной балке № 3.



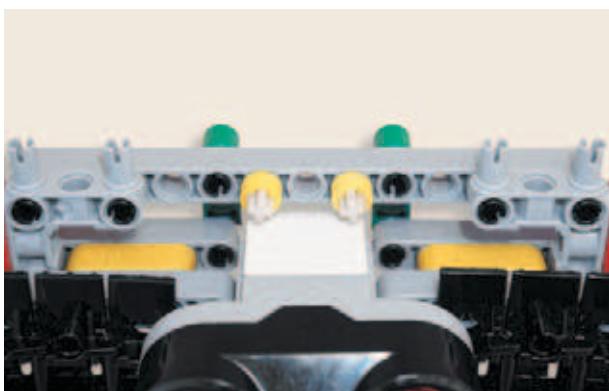
13. Надень на торцевые штифты серых угловых 3×3-модульных штифтов по одной красной балке № 3, как показано на рисунке.



14. Переверни собранную конструкцию датчиком к себе. Вставь два бежевых 2-модульных штифта в серую балку № 13 в пятый и седьмой модули.



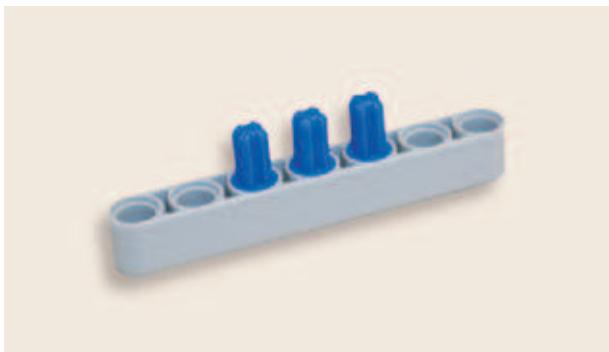
15. Надень на свободные концы этих штифтов по одной жёлтой  $\frac{1}{2}$ -модульной втулке.



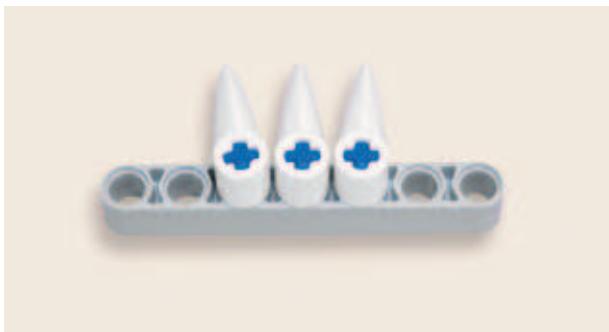
16. Надень на серую ось № 7 три чёрных зубчатых колеса на 4 зуба.



17. Теперь вставь в третий, четвёртый и пятый модули серой балки № 7 по одному синему 2-модульному штифту.



18. Надень на свободные концы синих штифтов по одной белой 3-модульной стрелке.



19. Расположи две криволинейные чёрные панели  $5 \times 11$  так, чтобы крепления длинных сторон располагались снизу, а крепления коротких сторон «смотрели» друг на друга. Продень насквозь серую ось № 7 с зубчатыми колёсами на 4 зуба во вторые модули крепления коротких сторон криволинейных панелей.



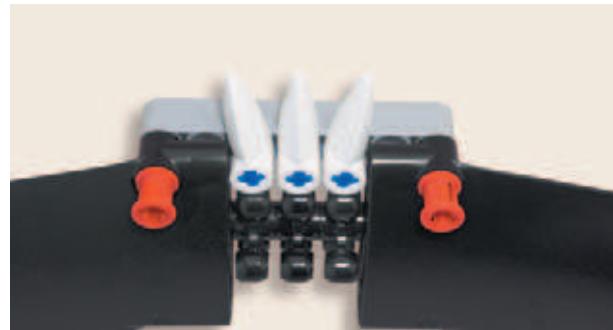
20. Надень на свободные концы серых осей по одной 1-модульной серой втулке.



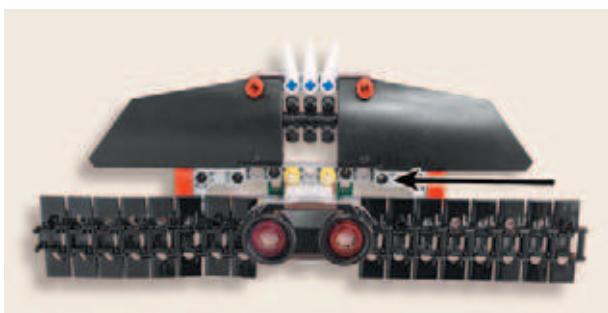
21. Переверни собранные панели и вставь два красных соединительных 3-модульных штифта в одиночные модули креплений вверху криволинейных панелей.



22. Надень на свободные концы красных штифтов серую балку № 7 вместе с белыми стрелками так, чтобы стрелки располагались сверху балки.



23. Положи собранную конструкцию радара так, чтобы разъём датчика находился сзади. Надень собранные вместе криволинейные панели на крепления серых угловых штифтов.



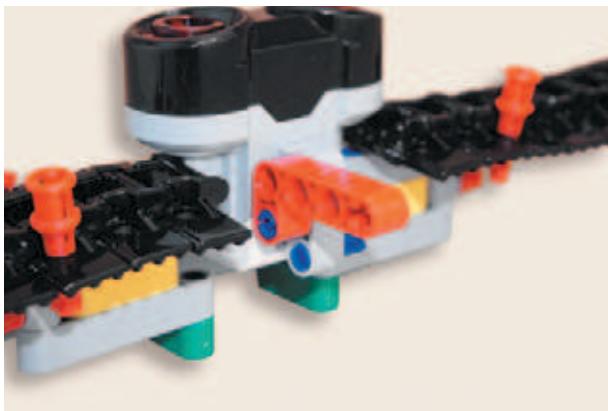
24. Вставь четыре красных 3-модульных штифта в нижний ряд модулей «гусениц» в левую панель — в первый и шестой, а в правую — в третий и восьмой модули.



25. Для того чтобы прикрепить радар к башне, вставь 2-модульный синий штифт во второй модуль крепления ультразвукового датчика слева, а справа — в третий модуль крепления датчика.



26. Надень слева на свободный конец синего штифта красную прямоугольную балку вторым модулем короткой стороной так, чтобы длинная часть балки была направлена к тебе.



27. Надень справа вторую красную прямоугольную балку первым модулем длинной стороны так, чтобы короткая сторона была направлена вверх.

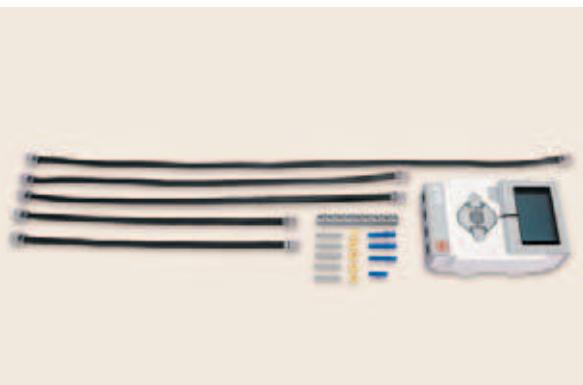


Ты молодец, защитник! Приступай к подключению коммуникаций!

## ШАГ 5. ОБЪЕДИНЕНИЕ КОМПОНЕНТОВ СИСТЕМЫ

### Детали для сборки:

- модуль EV3, 1x;
- кабель 25 см, 2x;
- кабель 35 см, 2x;
- кабель 50 см, 1x;
- балка № 7, серая, 1x;
- ось № 3, серая, 5x;
- втулка,  $\frac{1}{2}$ -модульная, жёлтая, 5x;
- соединительный штифт, 2-модульный, синий, 1x;
- соединительный штифт, 3-модульный, синий, 3x.



1. Для того чтобы прикрепить радар к башне, вставь синий 2-модульный штифт в чёрное крепление поворотного стола башни.



2. Теперь надень собранный радар крайним модулем красной балки на свободный конец синего 2-модульного штифта.



3. Вставь синий 3-модульный штифт длинной стороной через второй модуль короткой стороны правой красной балки в чёрное крепление поворотного стола, как показано на рисунке.



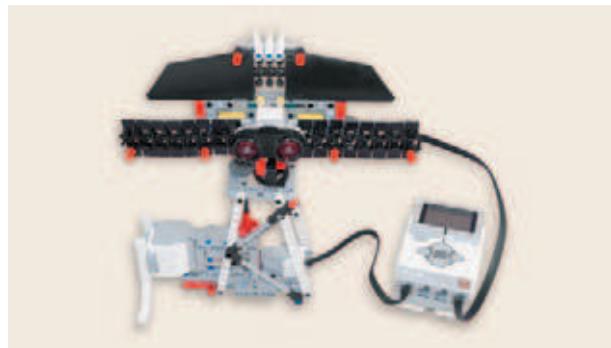
4. Помести пулемёт на турель. Для этого приложи пулемёт нижним креплением среднего мотора к креплениям поворотного стола. Вставь в первый и третий модули крепления мотора синие 3-модульные штифты длинным концом. Тебе осталось подключить провода.



5. Вставь один конец длинного кабеля 50 см в разъём ультразвукового датчика, а второй конец — в порт № 4 модуля EV3.



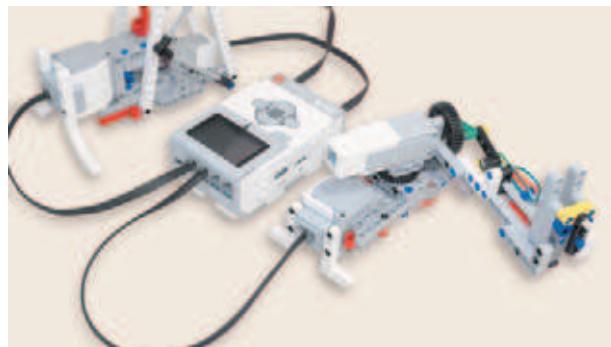
6. Соедини кабелем 25 см датчик касания и порт № 1 программируемого блока.



7. Соедини вторым кабелем 25 см большой мотор основания башни и порт D модуля EV3.



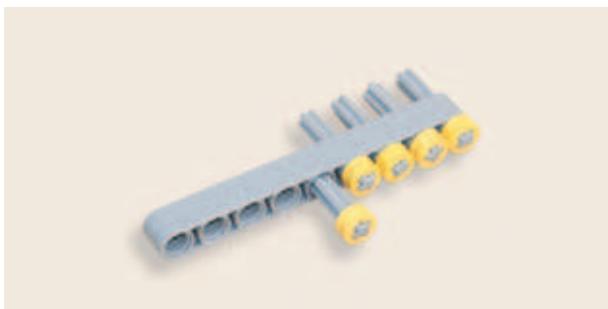
8. Соедини кабелем 35 см разъём большого мотора на основании пушки и порт С модуля.



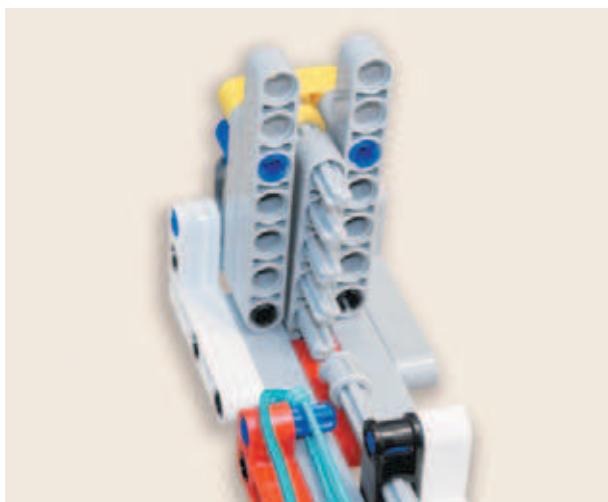
9. Вставь один конец кабеля 35 см в разъём среднего мотора на приводе пушки, а второй конец — в порт А модуля EV3.



Сторожевая башня готова! Можно было бы начинать программировать, но у нас осталось, юный защитник, одно незавершённое дело. Башня крепка, броня цела, радар есть... а где патроны к пулемёту? Давай их изготовим.



10. Надень на свободные концы пяти серых осей № 3 по  $\frac{1}{2}$ -модульной жёлтой втулке. Для создания обоймы вставь по одной подготовленной оси № 3 в модули серой балки № 7 с первого по пятый.



11. Для того чтобы зарядить пулемёт, возьми серую балку № 7 вместе с патронами и вставь её в приёмный механизм, как показано на рисунке.

Молодец! Теперь ты точно закончил, и враг больше не страшен!



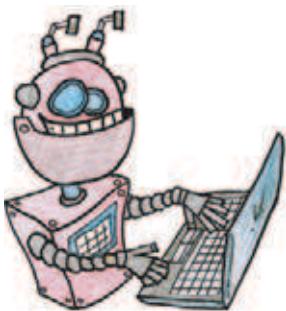
## Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LMEEV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел Загрузки -> LEGO® MINDSTORMS® Education EV3, где ты сможешь скачать установочный файл LME-EV3.

### Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

# Этап 4. Создание программы для сторожевой башни

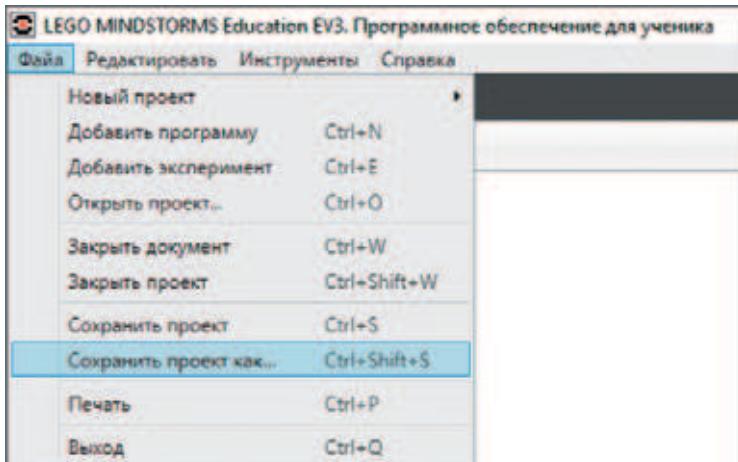


## ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

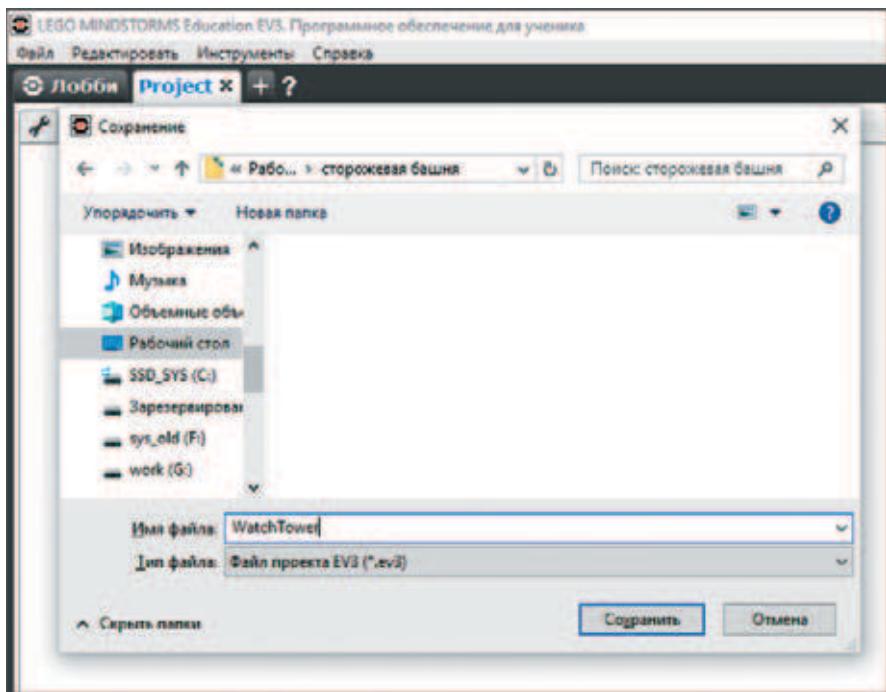
1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

## СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, то не нужно искать его в компьютере.



1. Для начала сохрани свой проект в памяти компьютера, чтобы не потерять его. Для этого нажми меню **Файл** в левом верхнем углу окна программы и выбери в контекстном меню пункт **Сохранить проект как...**
2. Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла: WatchTower** (в переводе с английского «Дозорная башня») в нижней части окна и нажми **Сохранить**.

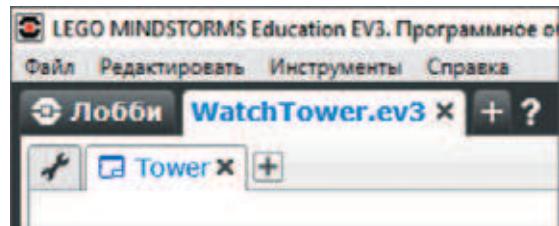


Именно так будет называться папка в памяти программируемого модуля EV3, где будут храниться все программы, которые ты напишешь в рамках этого проекта.

3. Осталось дать имя самой программе (в нашем проекте будет только одна). Для этого дважды кликни по имени **Program** в левом верхнем углу поля программы и задай новое имя: *Tower* (в переводе с английского «Башня»), далее нажми клавишу **Enter**.

#### Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они пишут, например, каждое слово с большой буквы (как это сделал ты) или используют нижнее подчёркивание.



## ЛОГИКА ПРОГРАММЫ

Сначала рассмотрим, как должны быть расположены башня и огневая установка. Радар и пулемёт должны быть направлены в одну сторону, чтобы выстрел выполнялся по цели, а не куда-то мимо.

В нашем роботе две большие составляющие, которые двигаются независимо друг от друга, — радар и пулемёт. Мы напишем два процесса и объединим их: отдельно для управления поворотами и отдельно — для обстрела врага.

После запуска радар выполняет сканирование местности: верхняя часть башни вращается от положения 0 градусов до максимально возможного угла. Когда достигается максимум, робот получает сигнал от

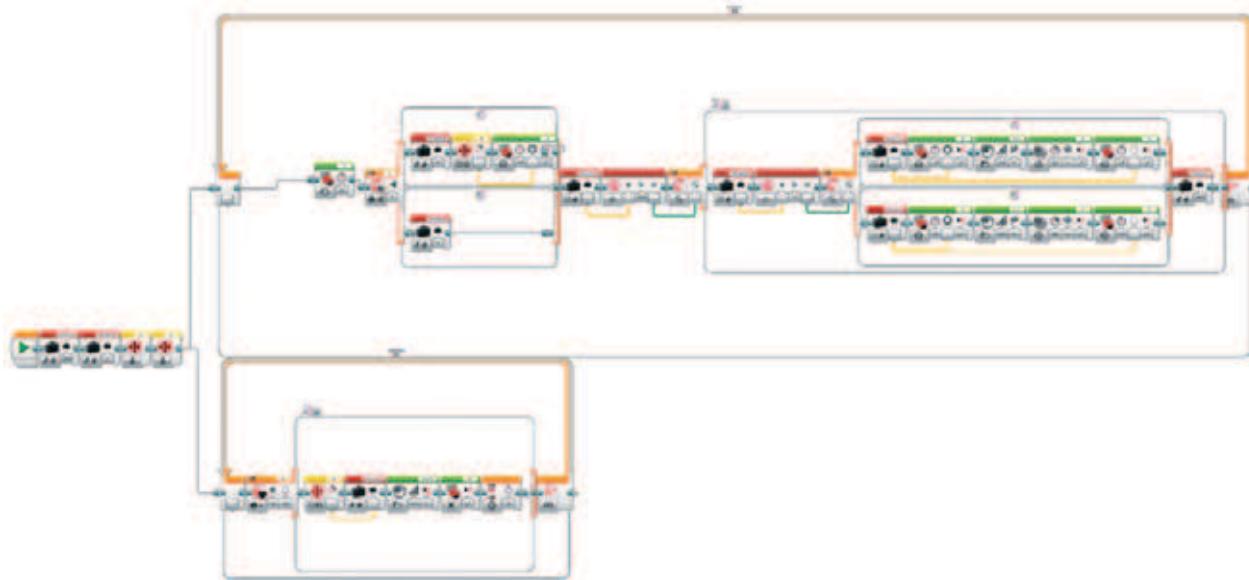
нажатой тактовой кнопки и меняет направление поворота радара. Радар возвращается в стартовое положение (0), и цикл повторяется. Пулемёт поворачивается до координат врага и производит выстрел, сопровождаемый звуком, затем пулемёт возвращается на исходную позицию.

В это время постоянно сканируется область перед ультразвуковым датчиком. Если

### Кстати!

В данной программе будет несколько интересных алгоритмических конструкций и приёмов программирования. Один из них — это *параллельное программирование*, то есть одновременное выполнение нескольких программ. Наша программа распараллеливается сразу после объявления глобальных переменных и сброса энкодеров.

обнаружен враг на расстоянии меньше или равном 30 см, в глобальную переменную заносится значение угла, на котором расположен враг. При обнаружении врага издаётся предупреждающий сигнал.

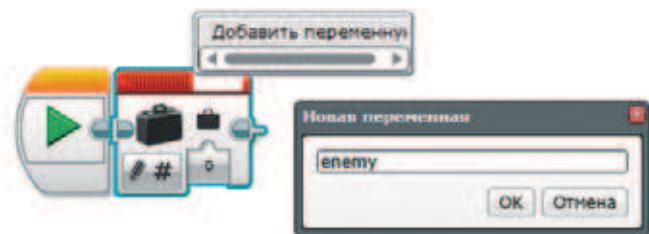


Мы решили, что такой способ с возвращением на исходные позиции и несинхронным движением башни и пулемёта выглядит для врага наиболее устрашающее. А ты с нами согласен?

## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ СТОРОЖЕВОЙ БАШНИ

Начинай составлять программу для сторожевой башни в открывшемся окне проекта.

- Для начала создай две глобальные переменные, то есть те, к которым доступ есть из любого места программы. Для этого вытащи красный блок **Переменная** сразу после блока запуска. Нажми на правый угол блока и **Добавить переменную**. Дай ей имя **enemuy**, что в переводе с английского означает «враг». Тут будет храниться информация о координате врага.

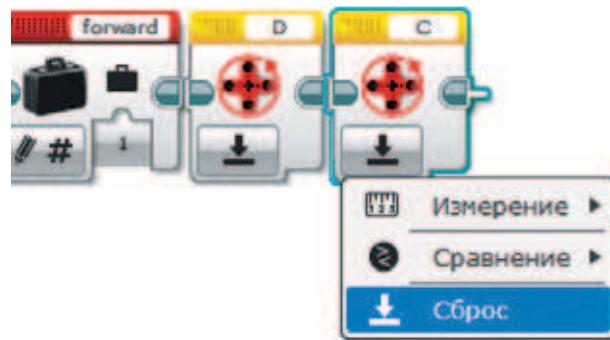


- Вытащи вторую **Переменную (красный блок)**. Её необходимо назвать **forward** — «вперёд». В неё будет записываться флаг о направлении движения сонара. Поставь по умолчанию значение 1 для **forward**. Для переменной **enemuy** установи значение **-500**.

Дело в том, что сонар будет вращаться от стартового положения на различное количество градусов как в положительную, так и в отрицательную сторону. Последнее происходит из-за погрешности моторов. На всякий случай мы задаём значение «с запасом».

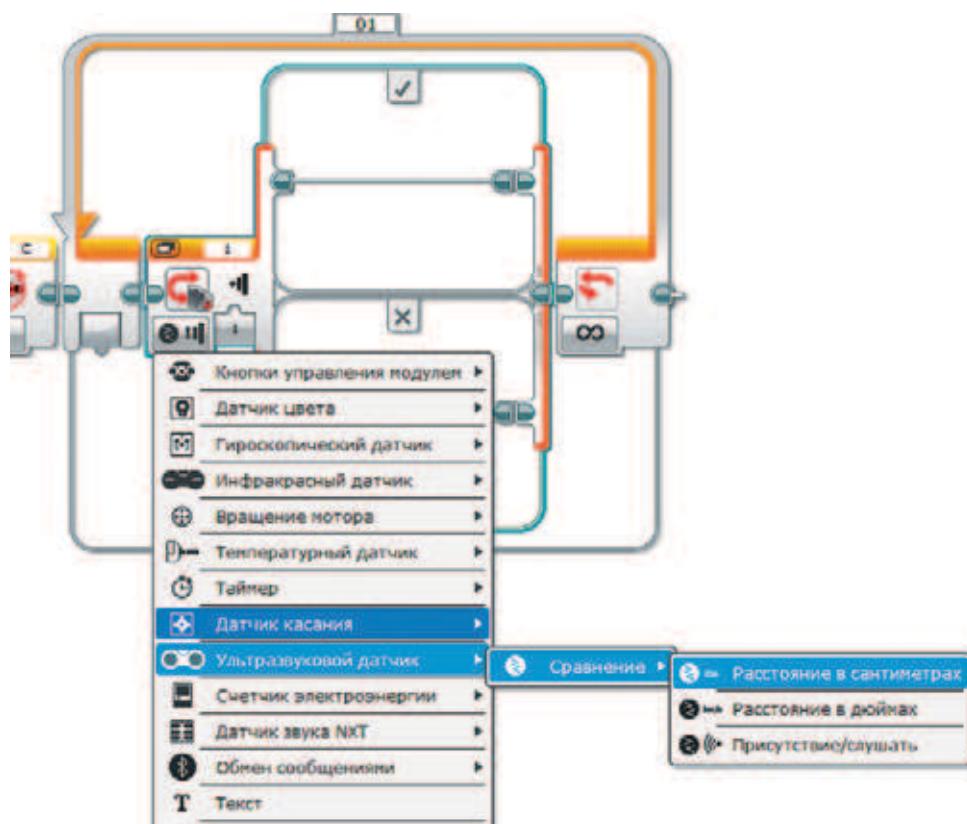
- Чтобы затем измерять координаты врага, надо сделать сброс энкодеров. Таким образом, твой робот будет считать выставленное при запуске программы положение моторов стартовым.

Вытащи два блока **Вращение мотора (жёлтый блок)** и выбери в свойствах **Сброс**. Убедись, что у тебя выбраны большие моторы правильно: **D** и **C** соответственно.



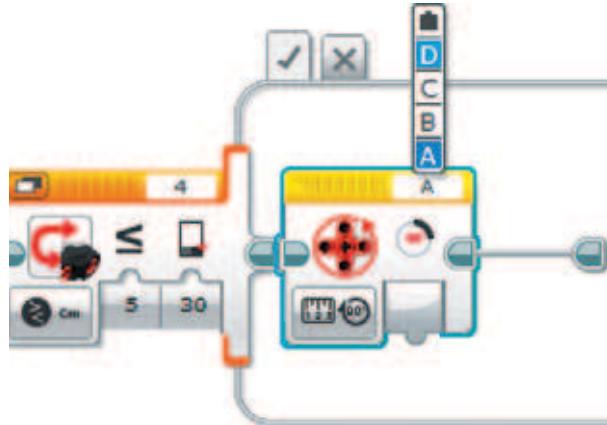
4. Далее нужно будет сделать два одинаковых по праву и выполняемых одновременно цикла — параллельных цикла. Однако предлагаю тебе начать с более простого и менее объёмного — работы сонара по обнаружению врага.

Вытащи блок **Цикл** (оранжевый блок). Сразу же внутрь цикла вложи **Переключатель** (оранжевый блок). Тебе нужно будет выбрать событие для переключения — условие. В нашем случае в качестве условия выступают ультразвуковой датчик и проверка расстояния меньше или равного 30 см перед башней. Выбери **Ультразвуковой датчик** — **Сравнение** — **Проверка расстояния в сантиметрах**.



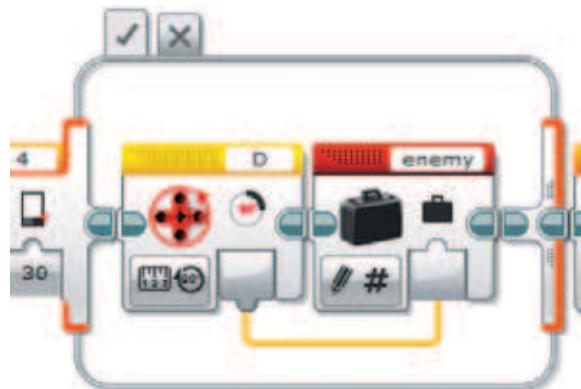
5. Убедись, что у тебя правильно выбран порт № 4 для датчика и введены параметры. Для удобства переключись на **Переключиться на вид с вкладками**.

6. Внутрь вкладки с положительным выполнением условия вытащи блок **Вращение моторов** (жёлтый блок). Проверь, что выбран мотор на порте **D** и выполняется **Измерение – Градусы**. Что здесь происходит? Нам же нужно сохранить координаты врага!

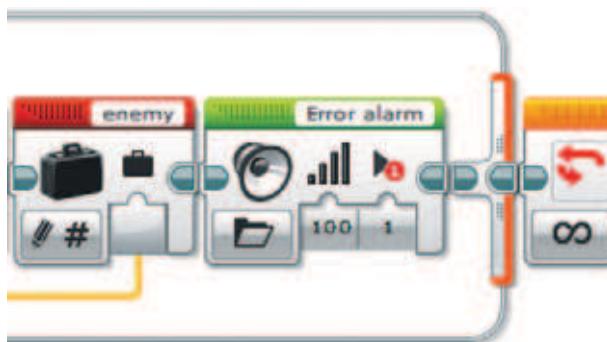


Теперь надо сохранить считанное значение.

7. Вытащи блок **Переменная** (красный блок) и выбери **Записать значение**. Убедись, что запись производится в переменную **enemy**. Соедини блок считывания с мотором и переменную.



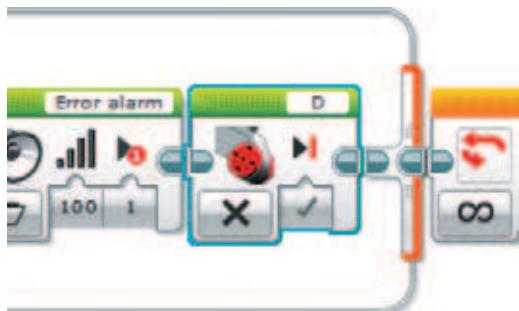
8. Добавь блок для сирены – **Звук** (зелёный блок). Убедись, что стоит режим **Воспроизвести файл**. Выбери звук **Error alarm** из стандартной библиотеки в папке **Информация**.



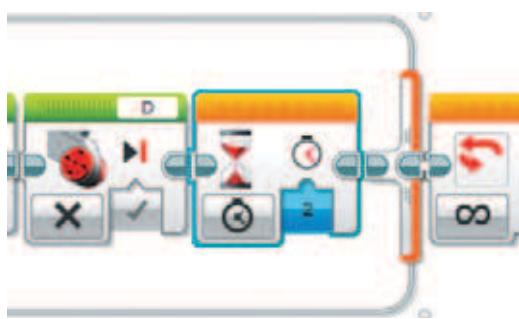
Оставь громкость максимальной. Звук должен воспроизводиться один раз.

Теперь нужно остановить движение сонара, сделав «его взгляд пристальным».

9. Вытащи блок **Большой мотор** (зелёный блок). Убедись, что сверху справа на блоке выбран мотор **D**. Выбери параметр **Выключить**. Перед продолжением движения башни нужно выждать трагическую паузу.

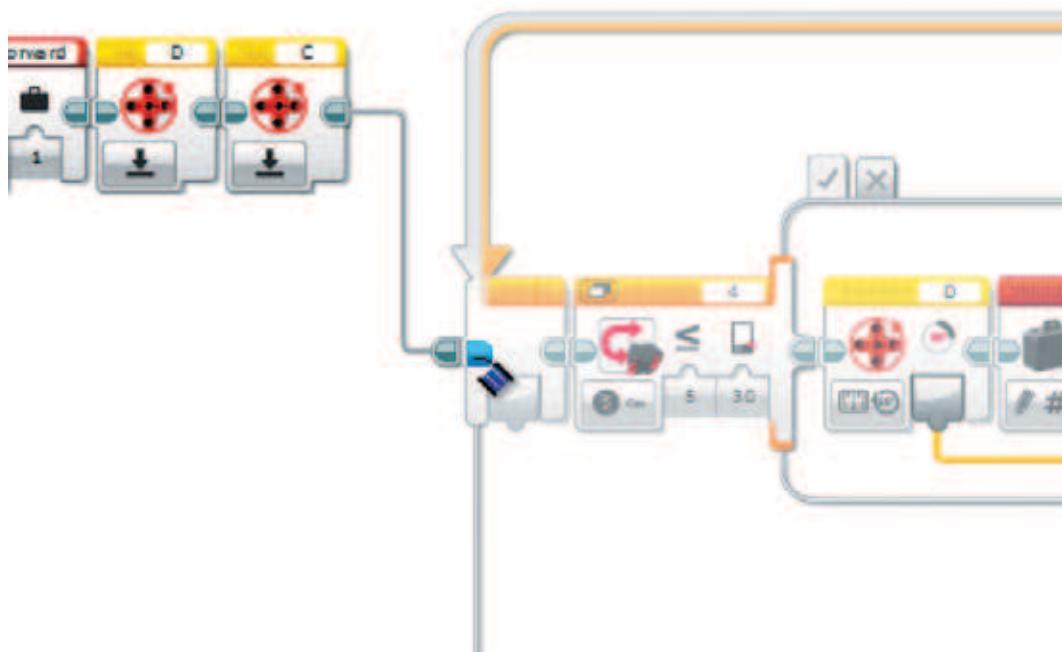


10. Вытащи блок **Ожидание** (оранжевый блок) и установи паузу на 2 секунды.

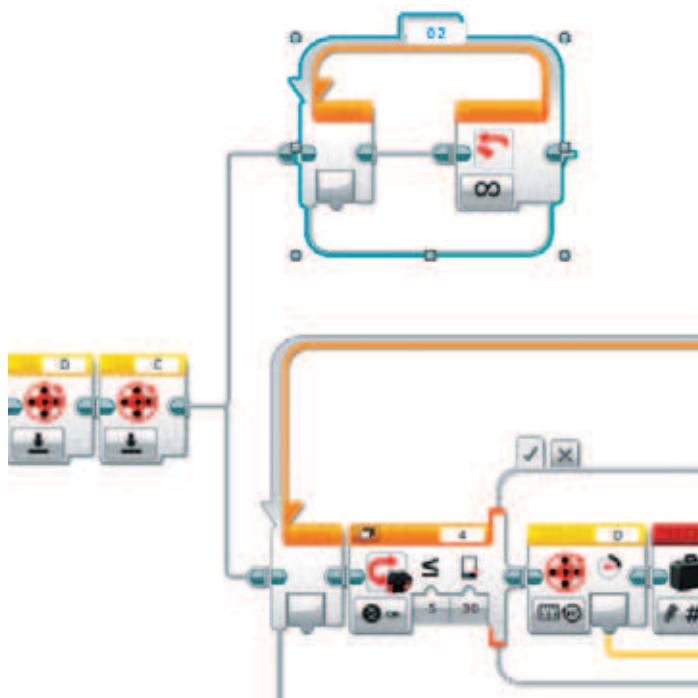


Отлично! Теперь можно заняться вторым циклом.

11. Внезапное задание: отсоедини готовый цикл. Он станет блёклым — неактивным. Не беспокойся! Соедини нить блоков остановки мотора и цикл.



12. Вытащи блок **Цикла** (жёлтый блок) и соедини его с блоком **Мотор**. Не забудь переименовать цикл, чтобы не возникло конфликта имён. Для этого измени название **01** в текстовом поле наверху блока цикла на значение **02**.



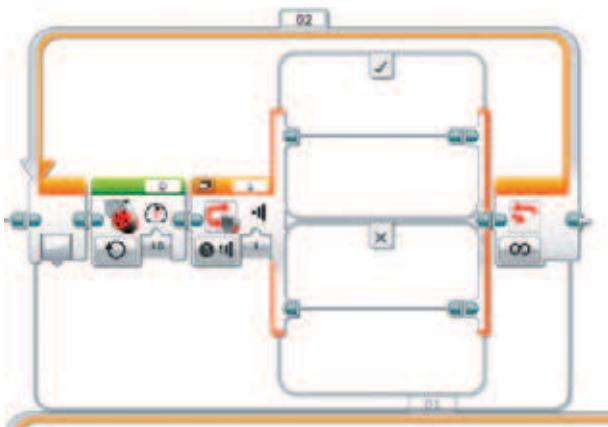
Начнём наш долгий процесс наполнения второго цикла.

13. Вытащи **Большой мотор** (зелёный блок) и установи режим **Включить** и мощность **10**, чтобы сонар вращался медленно.



Ты спросишь, как же вернуться в обратную сторону, ведь при текущем состоянии программы сонар будет постоянно пытаться повернуть в одну сторону? Существует несколько вариантов решения этой проблемы. Например, можно задать возврат после нажатия кнопки, а можно сначала узнать состояние кнопки и просто менять направление. Мы, для разнообразия, выбрали первый вариант. Второй делается проще, и ты можешь попробовать реализовать его самостоятельно.

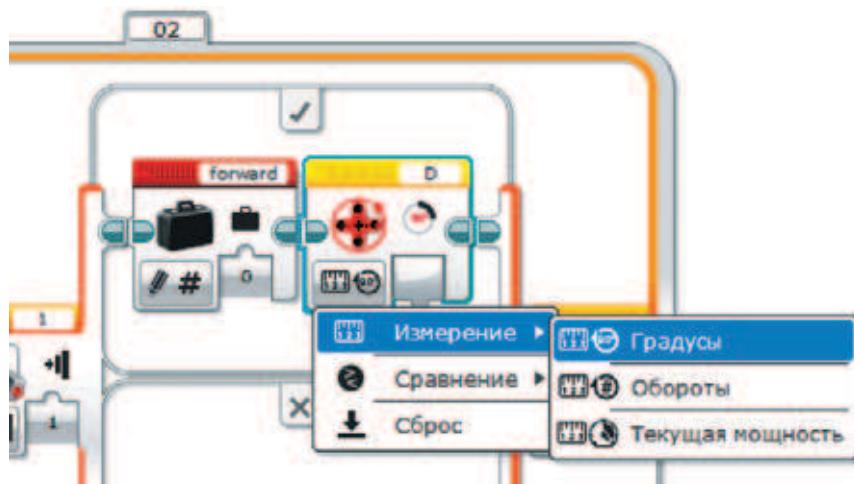
14. Тебе понадобится **Переключатель** (оранжевый блок). В нём будут указаны датчик касания, порт № 1 и режим ожидания. Не изменяй эти настройки.



15. Сначала наполни ветку с выполненным условием. Вытащи **Переменную** (красный блок) и выбери режим **Запись**, значение **0** и переменную **forward**. Таким образом, ты запишешь во флаг, что движение производится НЕ вперёд.

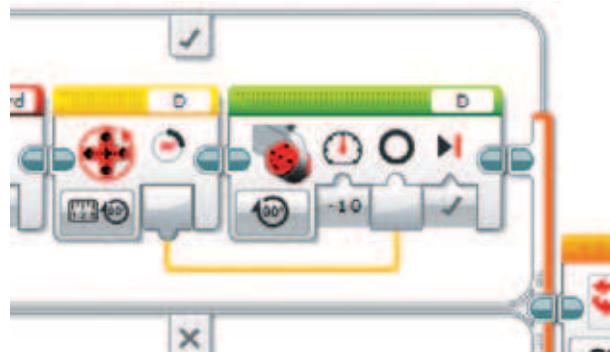
Надо вернуть сонар в исходное положение.

16. Сначала измерь, на сколько градусов вышел поворот, для точности. Вытащи блок **Вращение мотора** (жёлтый блок) и установи на нём следующие параметры: порт **D** и **Измерение → Градусы**.

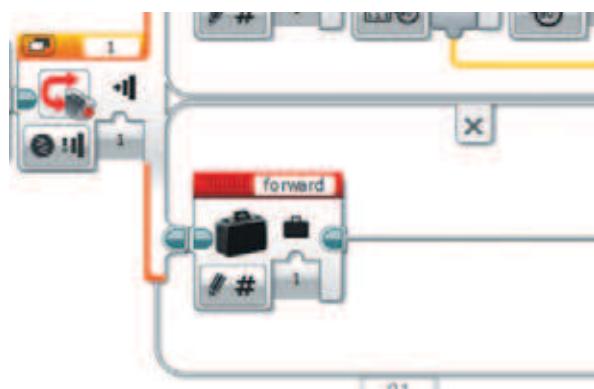


17. Включи мотор на нужное количество градусов в обратную сторону с помощью блока **Большой мотор** (зелёный блок) и режим **Включить на количество градусов**.

Убедись, что у тебя выставлен порт **D**. Соедини вход **Градусы** мотора и выход предыдущего блока. Так как вращать сонар надо в обратную сторону, установи мощность **-10**.

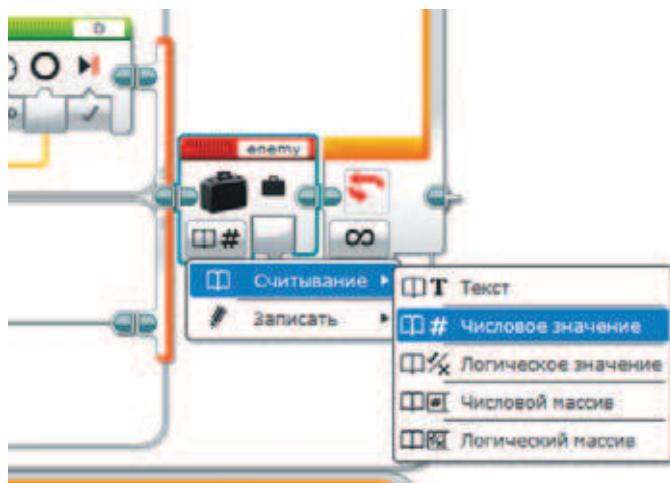


18. В нижнюю вкладку допиши обновление **Переменной** при прямом ходе. Проверь, что вверху справа написано **forward** и производится запись 1.

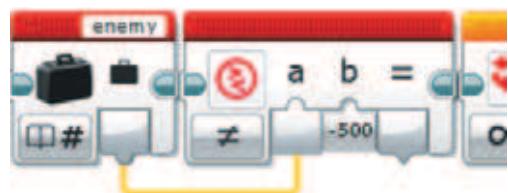


С ветвлением закончено! Теперь займёмся пулемётом.

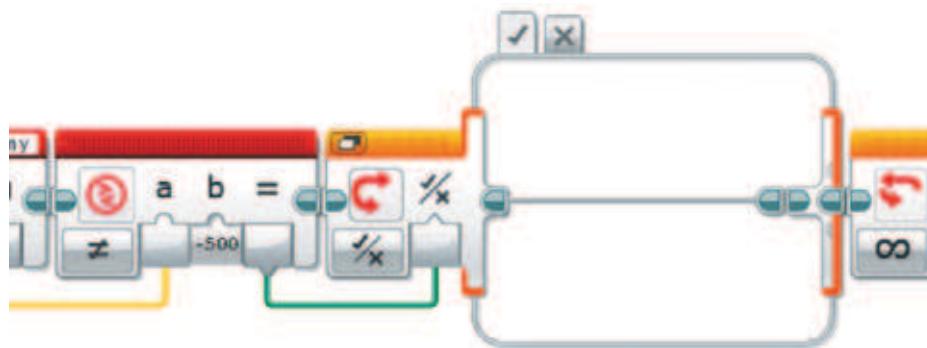
19. Считай координаты врага. Для этого вытащи блок **Переменная**, установи режим **Считывание** → **Числовое значение** и укажи переменную **enemtu**.



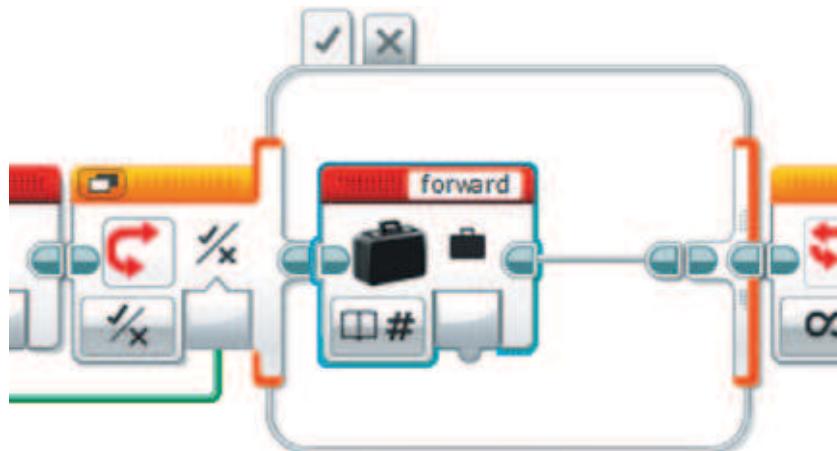
20. Чтобы узнать, был ли засечён враг, понадобится блок **Сравнение** (красный блок). Выбери режим **Не равно**. На первый вход протяни нить от предыдущего блока переменной, а на второй введи число **-500**, то есть значение для **enemy** по умолчанию.



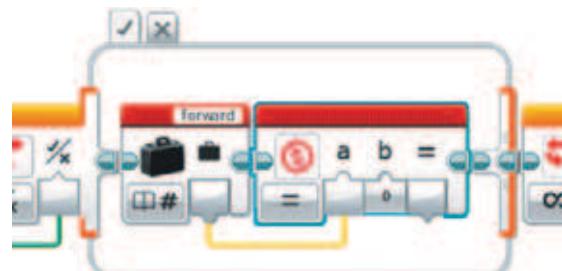
21. Если значение будет истинно, нужно выполнить выстрел. Но сначала вытащи ещё один блок **Переключатель** (оранжевый блок). Выбери режим **Логическое сравнение** и подай к нему на вход значение с выхода сравнения. Для удобства переключи на вид с вкладками.



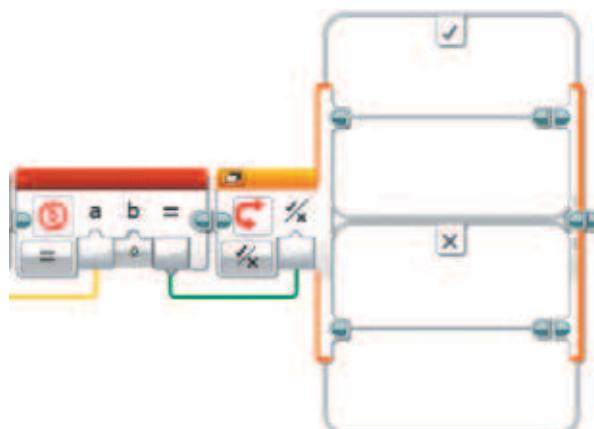
22. Ты же помнишь, что поворот на угол у нас считался вместе с направлением движения? Вытащи в переключатель блок **Переменная**. Выстави режим **Считывание → Числовое значение**. Удостоверься, что ты работаешь с **forward**.



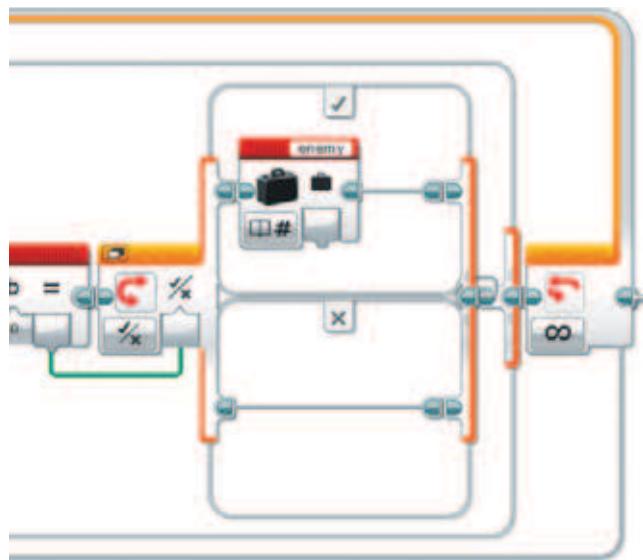
23. Опять понадобится блок **Сравнение** (красный блок), но на этот раз режим **Равно**, и сравнивать нужно будет значение, поданное от переменной в место **a**, и числовое значение **0** в ячейке **b**.

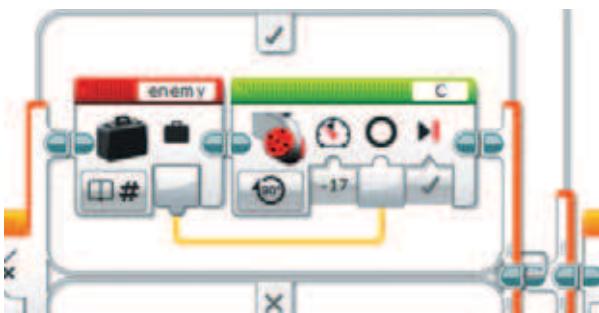


24. Снова используй ветвление. Вытащи блок **Переключатель** (оранжевый блок), однако сейчас понадобятся обе его вкладки. Выбери режим **Логическое значение** и подай значение с блока сравнения.

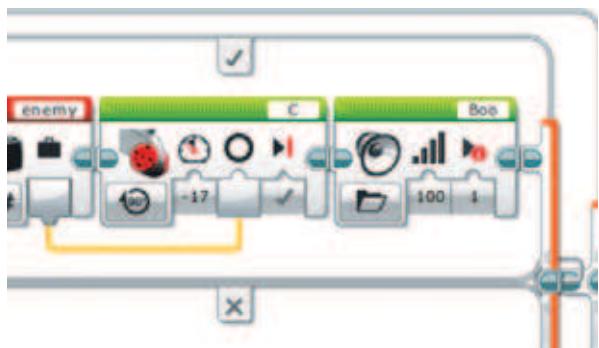


25. Если **forward = 0**, то движение происходит в обратном направлении. Следовательно, в верхней ветке переключателя (истина) мы будем составлять кусок, где будет использоваться отрицательная мощность мотора. Но сначала считай координату врага — вытащи блок **Переменная** (красный блок), выбери режим **Считать числовое значение** и убедись, что ты работаешь с переменной **enemy**.

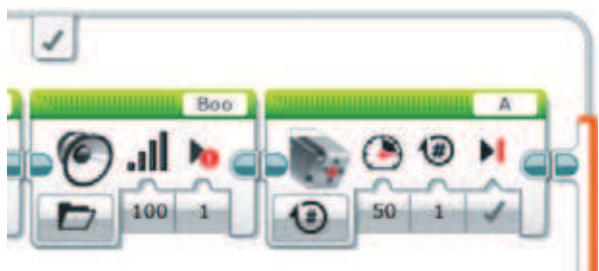




26. Используй **Большой мотор** (зелёный блок), чтобы повернуть мотор на турели, подключённый к порту **C**, на считаное количество градусов (режим **Включить на количество градусов**). Мощность, как договаривались, сделай отрицательной, например  $-17$ .



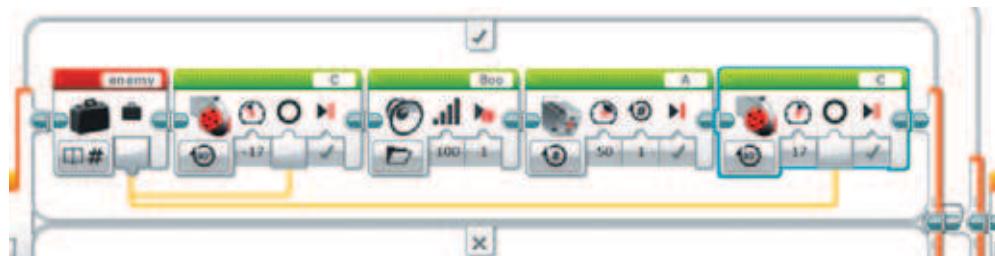
27. Пусть выстрел производится со звуком. Или, точнее, после звука. Вытащи блок **Звук** (зелёный блок) и установи ему режим **Воспроизвести файл**. Выбери из стандартной библиотеки папку **Выражения** и звук **Boo**. Пусть он воспроизводится однократно на полной громкости.



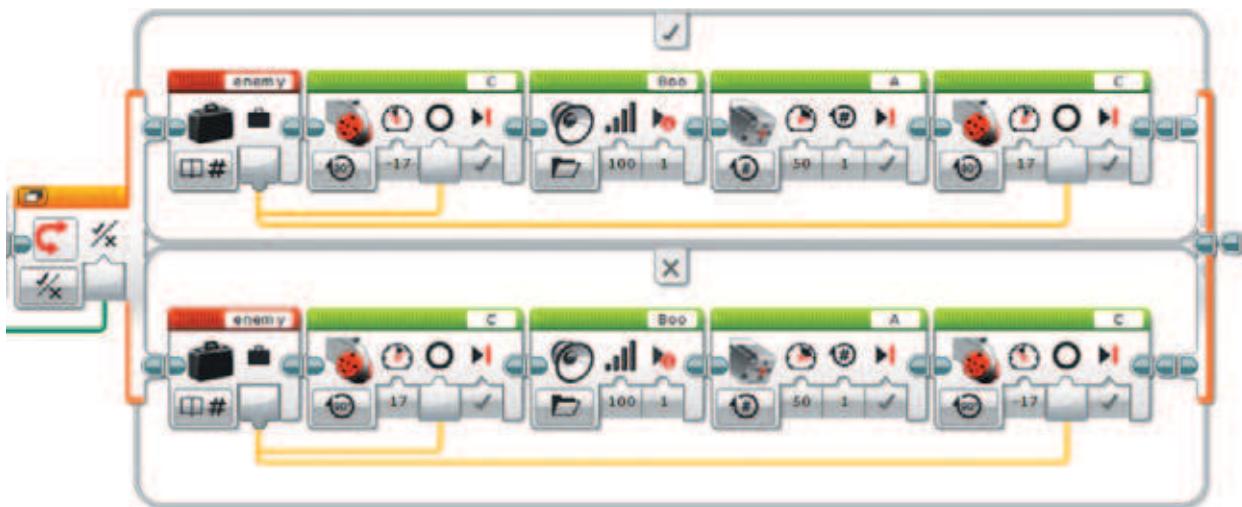
28. Произведи непосредственно выстрел пулемётом. Используй **Средний мотор** (зелёный блок) в режиме **Включить на количество оборотов**. Выстави мощность **50**, один поворот. Проверь, что наверху справа подписан порт **A**.

Теперь надо вернуть пулемёт на исходное положение.

29. Используй **Большой мотор** (зелёный блок) с режимом **Включить на количество градусов**. В качестве значения используй значение с выхода переменной, а мощность установи положительной ( $17$ ). Проверь, что ты управляешь мотором, подключённым к порту **C**. Теперь заполним блоки для ветки **forward = 1**.

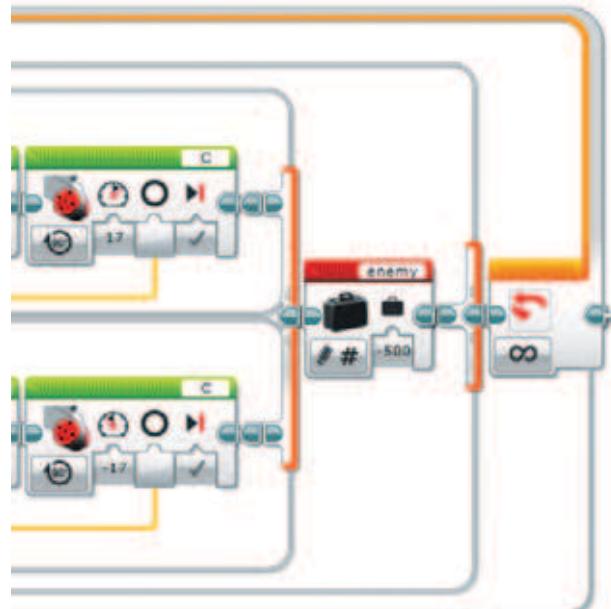


30. Скопирай содержимое верхней ветки в нижнюю вкладку, изменив во втором и последнем блоках знаки мощности.

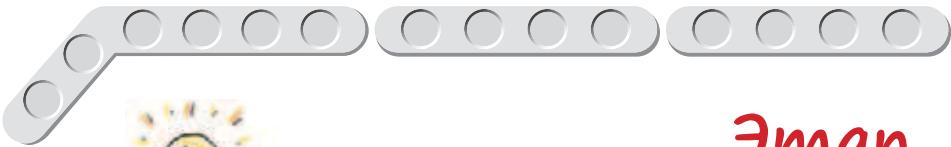


Остался последний блок, и программа готова.

31. Во внутренний цикл в конце после переключателей добавь **Переменную** (красный блок) в режиме **Записать → числовое значение**. Убедись, что ведёшь запись в переменную **enemy**. Запиши «сброс» к умолчанию — значение **-500**.



Поздравляем! Твоя программа готова! Можно запускать и приступить к охране рубежей!



## Этап 5. Загрузка программы и её тестирование

### ШАГ 1. ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

#### Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру, соединив порты PC на модуле и USB на компьютере. В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

### ШАГ 2. ТЕСТИРОВАНИЕ

1. Перед запуском программы обязательно убедись, что пулемёт и радар находятся в стартовом положении и имеют одну траекторию работы. При этом основание башни и турель будут находиться друг к другу под углом. Установи патроны.
2. Запусти программу и отойди из зоны сканирования. Радар должен сделать несколько поворотов туда-обратно, ни за что не цепляясь и возвращаясь в стартовое положение.
3. Помести руку или любой предмет в охраняемую зону. При обнаружении комплекс должен навести оружие, проиграть звук тревоги и выстрелить. Если выстрел был произведён мимо, скорректируй положение турели. При возникновении критической ситуации аварийно останови процесс выполнения программы нажатием на кнопку **Отмена** на программируемом блоке EV3 и устрани источник проблемы — он может заключаться в неправильном начальном положении механических частей радара или пулемёта или в программе.
4. Убедись, что вся программа выполняется корректно.





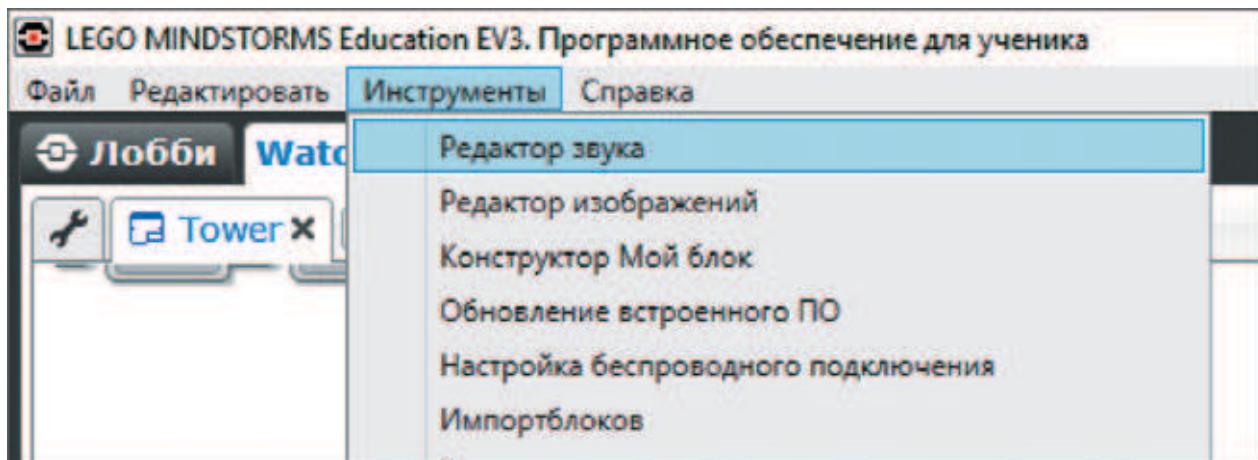
## А теперь...

У тебя есть самый настоящий охранный комплекс! Ура!

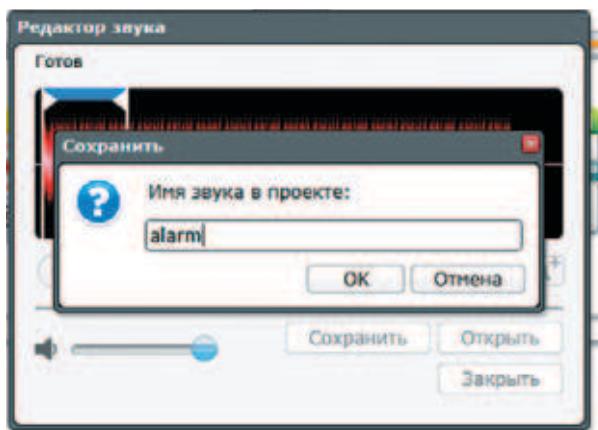
Но не нужно откладывать набор в сторону. Тебе предстоит реализовать много интересных идей, а мы тебе поможем — предложим несколько расширений для твоего проекта.

1. Дополни программу таким образом, чтобы после первичного обнаружения врага робот воспроизводил сигнал тревоги, но ждал несколько секунд и повторял проверку, а уже затем стрелял на поражение.
2. Ты можешь скачать дополнительные звуки, чтобы играть было ещё интереснее. Среда программирования LME поддерживает загрузку дополнительных звуков. Найди на странице электронных приложений к книге звуки сирены и выстрела и добавь их в проект.

Для этого тебе понадобится вкладка **Инструменты** → **Редактор звука** и **Открыть**.



Назови первый звук **alarm**, а второй — **gun**.



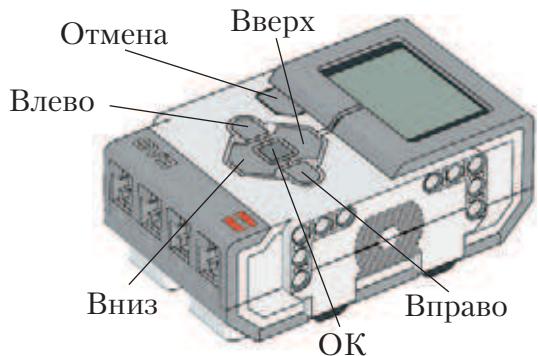
Затем их можно будет выбрать из звуков проекта, как обычно.

Особенно весело будет, если ты запишешь на диктофон фразу: «Стой! Кто идёт?».

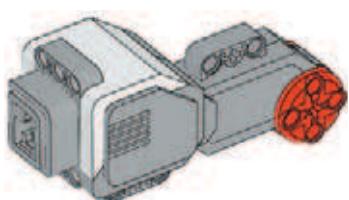
3. Сделай доклад о сторожевых башнях перед классом. Тебе помогут не только первая глава книги и Интернет, но и твои обширные знания, а также кинофильмы!
4. Устрой настоящую осаду башни, договорившись с друзьями. Тебе понадобится армия нападающих и защитников, но с этим же проблем не будет, верно?

## Перечень деталей

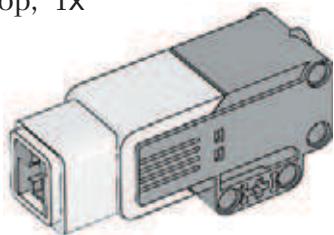
Модуль EV3, 1x



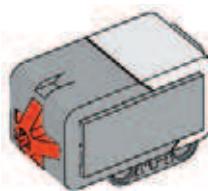
Большой мотор, 2x



Средний мотор, 1x



Датчик касания, 1x



Ультразвуковой датчик, 1x



### Зубчатые колёса

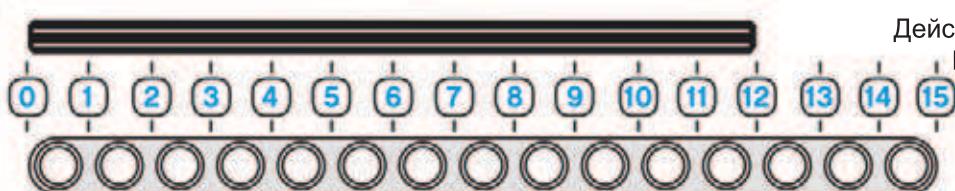
Двойное коническое зубчатое колесо на 36 зубьев, чёрное, 1x



Зубчатое колесо на 24 зуба, тёмно-серое, 2x



Зубчатое колесо на 4 зуба, чёрное, 3x



M 1:1

Действительный  
размер

### **Балки прямые**

Балка № 2, с поперечным отверстием, чёрная, 1x



Балка № 3, зелёная, 3x



Балка № 3, жёлтая, 4x



Балка № 3, чёрная, 1x



Балка № 3, красная, 3x



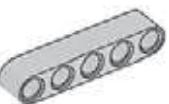
Балка № 3, синяя, 1x



Балка  $\frac{1}{2}$  № 4, чёрная, 2x



Балка № 5, серая, 4x

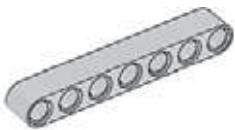


Вращающееся днище,  
28 зубьев, серое, 1x

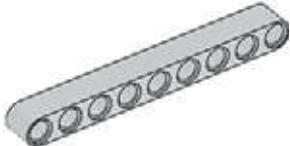


### **Балки прямые**

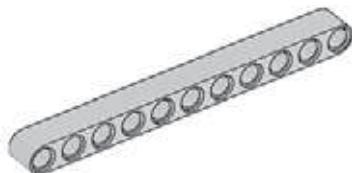
Балка № 7, серая, 5x



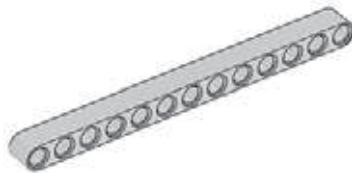
Балка № 9, серая, 2x



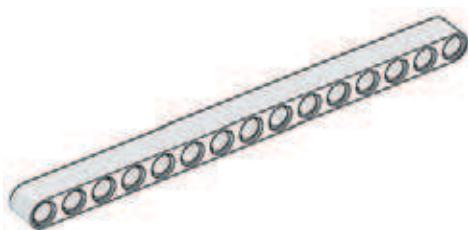
Балка № 11, серая, 2x



Балка № 13, серая, 3x



Балка № 15, белая, 4x



Вращающаяся верхушка,  
28 зубьев, чёрная, 1x

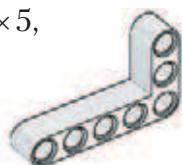


## Балки угловые

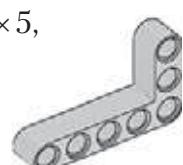
Прямоугольная балка  $2 \times 4$ , красная, 6x



Прямоугольная балка  $3 \times 5$ , белая, 4x



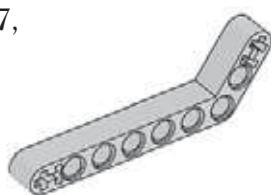
Прямоугольная балка  $3 \times 5$ , серая, 4x



Двойная угловая балка  $3 \times 7$ , белая, 2x



Угловая балка  $3 \times 7$ , серая, 2x



Треугольный фиксатор  $3 \times 5$ , серый, 2x



Стрелка, 3-модульная, белая, 4x



## Штифты

Соединительный штифт, 2-модульный, чёрный, 58x



Соединительный штифт, 2-модульный, синий, 18x



Соединительный штифт, 3-модульный, синий, 29x



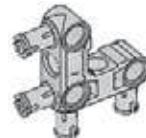
Соединительный штифт со втулкой, 3-модульный, красный, 6x



Соединительный штифт, 3-модульный, бежевый, 3x



Угловой соединительный штифт, 3×3-модульный, серый, 2x



Соединительный штифт, 2-модульный, серый, 7x



Соединительный штифт с осью, 2-модульный, бежевый, 6x

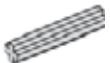


## Оси

Ось № 2, красная, 3x



Ось № 3, серая, 5x



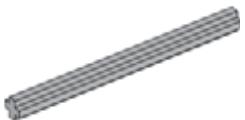
Ось № 4 с головкой,  
тёмно-серая, 1x



Ось № 6, чёрная, 2x



Ось № 7, серая, 4x



Ось № 8, чёрная, 1x



Ось № 9, серая, 2x



Звено гусеницы,  
чёрное, 16x



## Блоки

Поперечный блок,  
3-модульный,  
тёмно-серый, 2x



Двойной поперечный блок,  
3-модульный, чёрный, 1x



Угловой блок 1, 0°,  
чёрный, 3x



Угловой блок 2, 180°,  
чёрный, 2x



Поперечный блок,  
2-модульный, серый, 4x



Угловой блок 6, 90°,  
чёрный, 2x



## Втулки

Втулка — удлинитель оси,  
2-модульная, красная, 1x



Втулка,  $\frac{1}{2}$ -модульная,  
жёлтая, 9x



Втулка, 1-модульная,  
серая, 6x



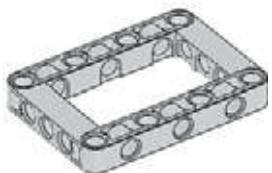
## **Кабели**

Кабель, 25 см, 2x  
Кабель, 35 см, 2x  
Кабель, 50 см, 1x

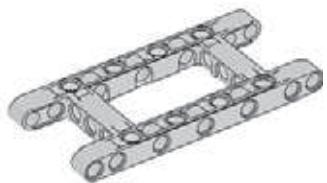


## **Рамки**

Рамка, 5×7, серая, 3x



Рамка, 5×11, серая, 1x



Левая криволинейная па-  
нель 5×11, чёрная, 1x



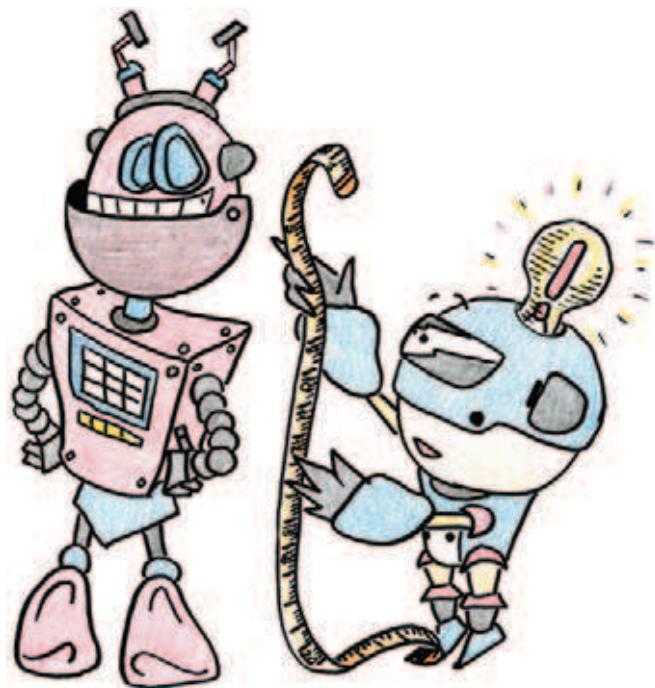
Правая криволинейная  
панель 5×11, чёрная,  
1x



**education**



Человек — Всему мера?



# История мер длины



Людям с незапамятных времён всегда нужно было что-то измерять. Как же ещё построить себе жилище по размерам и понять, поместится ли там вся семья? Или какого размера должна быть заслонка, чтобы закрыть печь? А как понять, далеко ты живёшь от своего друга или вы совсем близко?

Именно для этого люди и научились делать замеры длины, ширины, высоты, измерять расстояния. Но какими же инструментами люди могут измерять разного рода поверхности?



Рис. 1

Это линейка (рис. 1) — знакомый тебе с детства измерительный инструмент. С её помощью можно определить размеры любой плоской поверхности и получить их численное выражение в сантиметрах или миллиметрах.

А это строительная рулетка (рис. 2) — усовершенствованная компактная линейка, на которую нанесены разные единицы измерения длины: сантиметры, метры, дюймы. Рулеткой измеряют помещения, строительные конструкции и, с некоторой погрешностью, даже неровные поверхности.

Видов измерительных инструментов довольно много, однако люди ещё до их изобретения умели и считать, и измерять что-либо. Как же они это делали?

Да очень просто! Человек сам был всему мерой!

Первые представления о числе появились у людей в глубокой древности. Они воз-



Рис. 2



никли из необходимости подсчёта количества людей, животных, плодов, На ранних ступенях развития общества люди почти не умели считать. Они отличали друг от друга два и три предмета вместе. Например, о численности группы из двух предметов они говорили: «Столько же, сколько глаз у человека», а о 20 предметах — «Столько же, сколько пальцев у человека». Потом считали на пальцах. Когда пальцы на одной руке кончались, переходили на другую, а если на двух руках не хватало, переходили на ноги. Поэтому, если говорили, что у него «две руки и одна нога кур», это означало, что у него 15 кур, а если у кого-то было 20 коз, это называлось «весь человек», то есть две руки и две ноги.

Что же касается мер длины, то в Древней Руси люди пользовались тремя основными: «пядь», «локоть», «сажень». Выражение «семи пядей во лбу» используют в разговоре об очень умном человеке, имея в виду, что его ум настолько велик, что лоб разросся до семи пядей.

В Древней Руси о богатырях говорили: «У него косая сажень в плачах». А сколько это в сантиметрах? Можно ли измерить длину предмета древнерусскими мерами длины? Ответить на эти вопросы тебе поможет этот замечательный проект.

## СТАРИННЫЕ МЕРЫ ДЛИНЫ НА РУСИ

Начнём с самых маленьких.

1. **Вершок** (рис. 3). Наименование происходит от слова «верх» (пальца). Вершок упоминается в «Торговой книге», документе XVI века, как 1/16 аршина. Существовали доли вершка: «полвершки» и «четвертьвершки».

2. **Пядь** (рис. 4). Разновидностей пядей было несколько:

*малая пядь* — расстояние между раздвинутыми большим и указательным пальцами, соответствует 19 см; большая часть кирпичей XII века имеет ширину в пядь, то есть 19 см;

*большая (или великая) пядь* — расстояние между большим пальцем и мизинцем — около 22–23 см;

*пядь с кувырком* — пядь с добавлением двух суставов указательного пальца — около 27–31 см.



Рис. 3



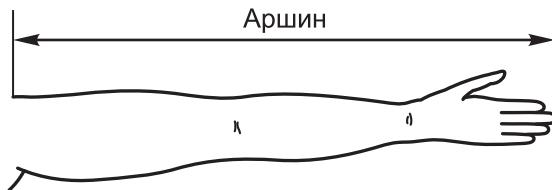
**Рис. 4**



**Рис. 5**

3. **Локоть** (рис. 5) — расстояние от локтевого сустава до концов вытянутых пальцев, соответствует двум большим пядям. Эту меру длины чаще всего использовали для измерения ткани.

4. **Аршин** (рис. 6). Эта мера заимствована с Востока. Считается, что её наименование происходит от наименования турецкой меры длины «аршин» (27,9 дюйма = 70,9 см) или от персидского слова «арши» — длина; на аршин обычно наносили деления в вершках.



**Рис. 6**

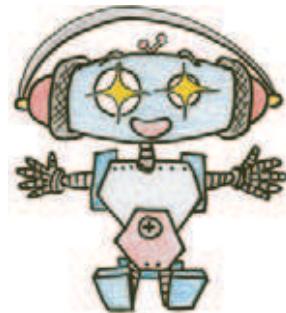
5. **Сажень** (см. рис. 5). Наименование «сажень» происходит от глагола «сягать» («досягать», «досягаемый»).

6. **Верста**. Для определения больших расстояний в Древней Руси существовала *верста*, или *поприще*. Верста была самой крупной единицей длины. Она складывалась из саженей. Учитывая, что сажень у каждого была своя, то в версте было от 500 до 750 саженей. Сейчас принято, что 1 верста = 1140 метров.

Тебе уже понятно, что результат измерения длины зависел от роста человека. Как ты думаешь, купцу при продаже ткани было выгодно иметь большой рост или маленький?

Все эти термины: пядь, локоть, сажень, верста, поприще встречаются уже в XI–XII веках. Меры длины, как и другие меры, не отличались устойчивостью и в период феодальной раздробленности на Руси колебались от одного княжества к другому, а также изменялись и во времени.

# Этап 1. Устройство робота-измерителя



Рассмотри модель робота-измерителя, собранную на основе набора LEGO® MINDSTORMS® Education EV3.

Попробуй выделить на ней рабочие детали — мотор и измерительное колесо.

Попробуй собрать эту модель.

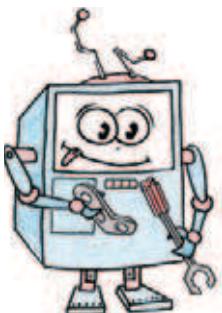
Обрати внимание, в конце проекта в таблице даны все детали, которые потребуются тебе для сборки. Эта таблица поможет быстро найти то, что необходимо, и не ошибиться при конструировании.

Вот так выглядит робот-измеритель, которого тебе предстоит собрать.

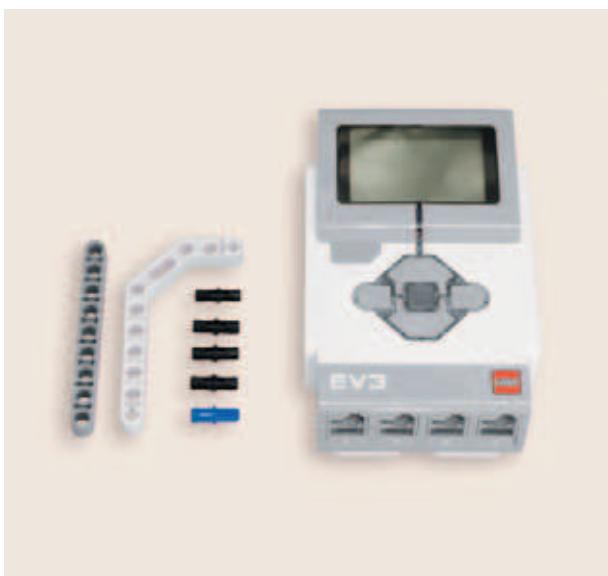




## Этап 2. Сборка робота-измерителя



### ШАГ 1. СБОРКА ОСНОВАНИЯ ДЛЯ КРЕПЛЕНИЯ МОТОРА



#### Детали для сборки:

- программируемый модуль EV3, 1x;
- балка № 9, серая, 1x;
- двойная угловая балка 3 × 7, белая, 1x;
- соединительный штифт, 2-модульный, синий, 1x;
- соединительный штифт, 2-модульный, чёрный, 4x.



1. Переверни программируемый модуль экраном вниз.
2. Со стороны портов, обозначенных цифрами, вставь по одному чёрному штифту в крайние боковые модули.



- Закрепи на этих штифтах серую балку № 9, как показано на рисунке.



- Со стороны портов, обозначенных буквами, вставь во второй модуль синий штифт, а в четвёртый и шестой – чёрные штифты.



- Закрепи на этих штифтах двойную угловую балку  $3 \times 7$ , как показано на рисунке.



Основание для крепления мотора готово!

## ШАГ 2. СБОРКА КРЕПЛЕНИЯ ИЗМЕРИТЕЛЬНОГО КОЛЕСА



### Детали для сборки:

- балка № 3, чёрная, 2x;
- угловая прямоугольная балка 2 × 4, красная, 1x;
- фиксатор, 3-модульный, чёрный, 1x;
- соединительный штифт, 3-модульный, синий, 2x;
- соединительный штифт, 2-модульный, синий, 4x;
- соединительный штифт, 2-модульный, чёрный, 4x.



1. Переверни программируемый модуль экраном вверх. Вставь на его правой боковой стенке в первый и третий модули в верхнем ряду синие 3-модульные штифты.



2. Надень на них до упора чёрную балку № 3.

3. Вставь в четвёртый модуль в этом же ряду синий 2-модульный штифт, а в шестой модуль — чёрный.



4. Закрепи на этих штифтах красную балку, как показано на рисунке.



5. Вставь в угловой модуль красной балки синий 2-модульный штифт.



6. Вставь до упора в двойную угловую балку  $3 \times 7$  во второй и третий модули чёрные штифты.



7. Закрепи на этих штифтах балку № 3, как показано на рисунке.



8. Вставь в её первый модуль чёрный штифт.



9. Установи на нём чёрный фиксатор (центральным отверстием).

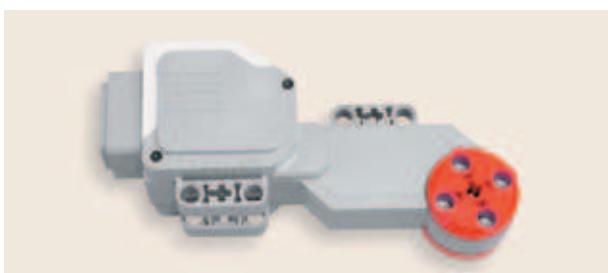


10. Вставь в крестовые отверстия фиксатора синие 2-модульные штифты.



Крепление для измерительного колеса готово!

### ШАГ 3. УСТАНОВКА МОТОРА



Деталь для сборки:

- большой мотор, 1x.



1. Расположи мотор так, как показано на рисунке.

2. Закрепи мотор на пяти синих штифтах. Обрати внимание — штифты должны войти в отверстия до упора!



### ШАГ 4. СБОРКА И УСТАНОВКА ИЗМЕРИТЕЛЬНОГО КОЛЕСА



Детали для сборки:

- ступица большая, 1x;
- шина большая, 1x;
- ось № 6, чёрная, 1x.

1. Натяни шину на ступицу. Проследи, чтобы она «села» ровно, без перекоса.



2. Вставь в центральное отверстие привода мотора ось № 6 до упора.



3. Надень на ось собранное колесо так, чтобы свободный конец оси «не выглядывал» из ступицы снаружи.



Измерительная часть робота готова! Остаётся подключить её к программируемому модулю.

## ШАГ 5. ПОДКЛЮЧЕНИЕ ИЗМЕРИТЕЛЬНОЙ ЧАСТИ К ПРОГРАММИРУЕМОМУ МОДУЛЮ



Деталь для сборки:

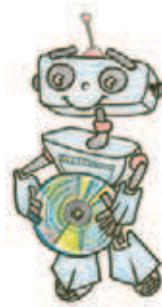
- кабель, 35 см, 1х.

1. Соедини выходной порт мотора с портом А программируемого модуля.



Поздравляем! Ты сумел собрать универсальный измерительный инструмент, с помощью которого сможешь определить размеры окружающих тебя предметов. И не только предметов, а даже живого существа, например длину твоего четвероногого друга, если он, конечно, будет не против. Но прежде готовое робототехническое устройство надо запрограммировать...





## Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

### Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://education.LEGO.com>.



## Этап 4. Создание программы для робота-измерителя

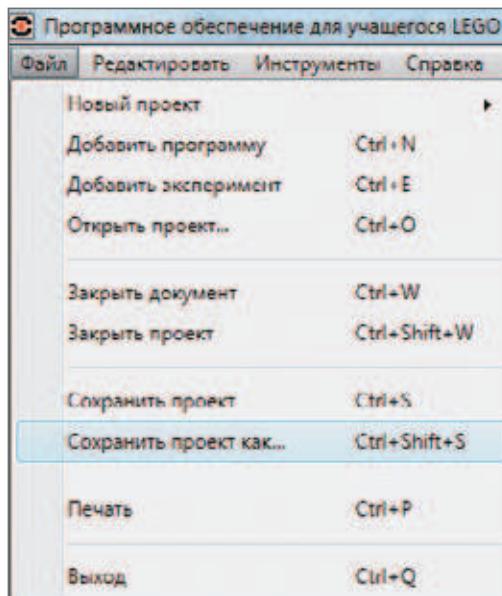
### ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

### СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

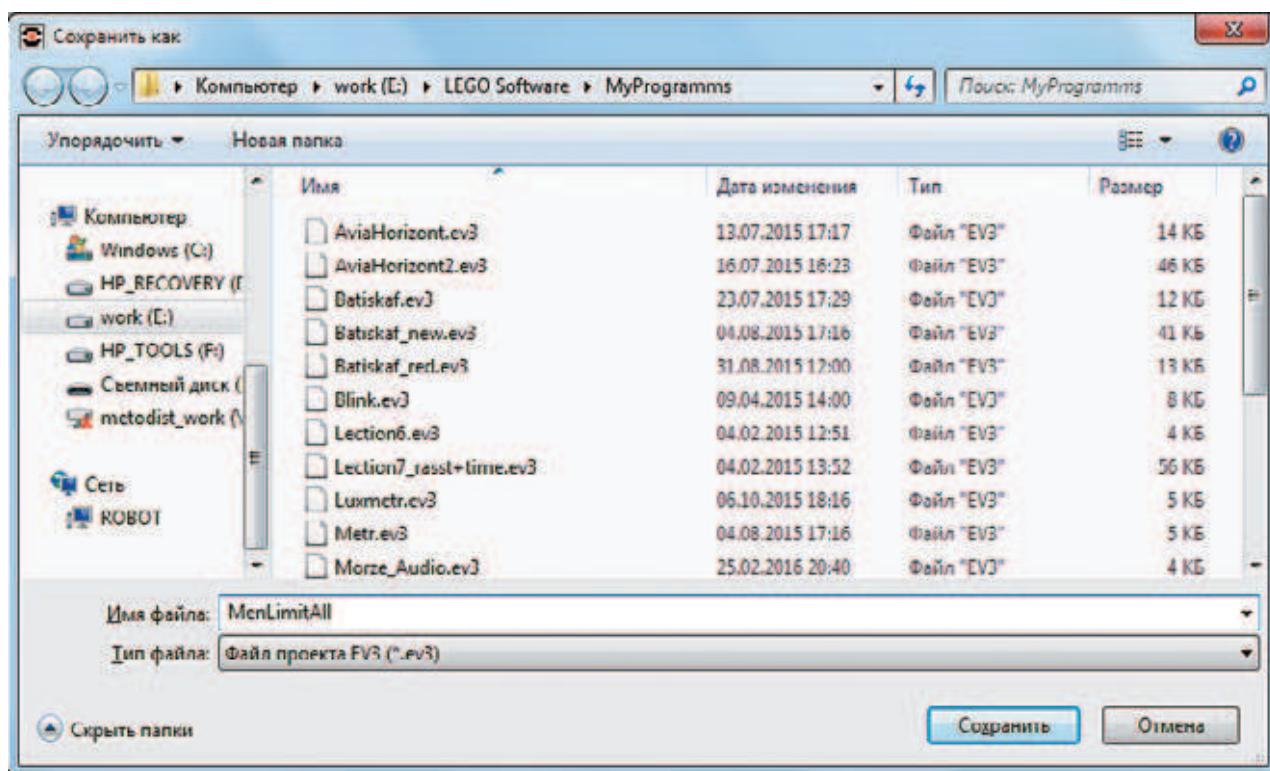
Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, не нужно искать его в компьютере, он всегда находится внутри робота.

Однако необходимо грамотно организовать хранение проектов. Папка с проектом должна иметь название, отражающее его суть, иначе тебе будет трудно её найти. Если в папке с проектом несколько программ, то назови их максимально понятно для себя.



1. Для начала сохрани твой проект в памяти компьютера, чтобы избежать его потери. Для этого выбери меню **Файл** в левом верхнем углу окна программы и выбери в контекстном меню пункт **Сохранить проект как...**

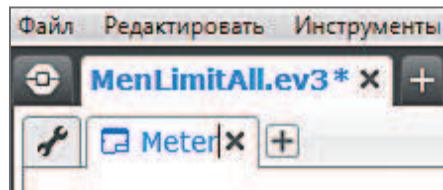
2. Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла**: *MenLimitAll* (в переводе с английского — «человек — всему мера») в нижней части окна и нажми **Сохранить**. Именно так будет называться папка в памяти программируемого модуля EV3, в которой будут храниться все программы, которые ты напишешь в рамках этого проекта.



3. Осталось дать имя самой программе (в нашем проекте она будет одна). Для этого дважды кликни по имени *Program* в левом верхнем углу поля программы, задай новое имя: *Meter* (в переводе с английского — «измеритель») и нажми клавишу **Enter**.

### Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они пишут, например, каждое слово с большой буквы (как это сделал ты) или нижнее подчеркивание.



## ЛОГИКА ПРОГРАММЫ

Работа робота-измерителя предельно проста: нужно считывать количество оборотов, которое сделало колесо, и умножать на длину его окружности. Моторы EV3 оснащены **энкодерами**<sup>1</sup>, которые в автоматическом режиме считывают данные о повороте *привода*<sup>2</sup>.

$$\text{Пройденное расстояние} = \text{Длина окружности колеса} \times \text{Количество оборотов колеса}$$

Количество оборотов колеса ты получишь из показаний счётчика мотора, а длину окружности колеса вычисли сам. Для этого измерь диаметр колеса и умножь его значение на 3,14. Должно получиться примерно 17,27 см.

### Кстати!

Какую бы окружность ты ни взял, её длина всегда будет в 3,14 раза больше её диаметра. Это число (или, как говорят математики, константу) называют числом Пи. Число Пи было известно ещё строителям древнего Вавилона и Египта. В Сиэтле (США) на ступенях перед зданием Музея искусств установлен памятник числу Пи. Есть у числа Пи и день рождения, даже два: 14 марта (3-й месяц и 14-й день) и 22 июля (22.07) – это число соответствует «приближённому Пи», записанному Архимедом в виде дроби 22/7.

Здесь получится интересная алгоритмическая конструкция **вложенных циклов** (то есть цикл в цикле).

В самой программе мы будем считывать показания мотора — сколько оборотов он сделал, затем умножать это число на 17,27 см. Результат будет выводиться на экран программируемого модуля. Так как значение на экране должно всегда обновляться, все эти действия помести в цикл. После каждого измерения показания прибора нужно сбрасывать, поэтому цикл будет завершаться по нажатию кнопки **OK** на программируемом модуле. Когда значение мотора сбрасывается, вся программа повторяется заново.

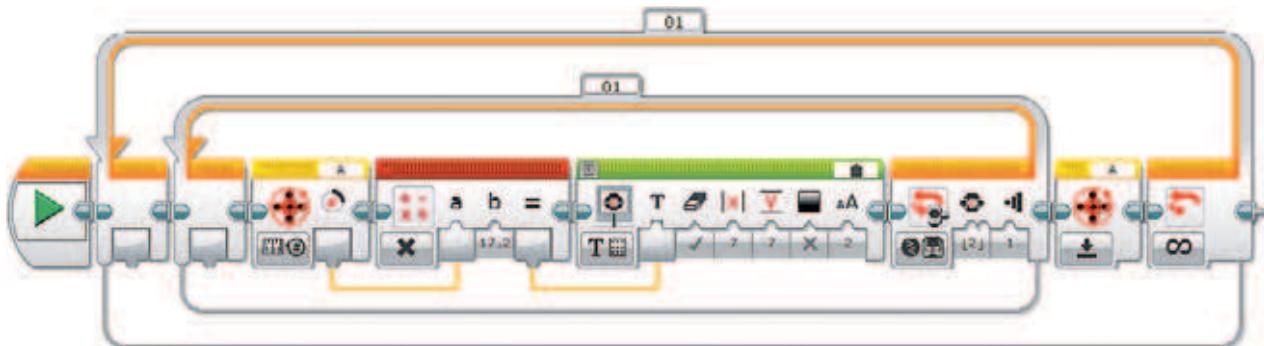
<sup>1</sup> Энкодер (в переводе с английского — «кодировщик») — устройство внутри мотора, позволяющее снимать различные показания: число совершенных оборотов или угол поворота привода.

<sup>2</sup> Привод — устройство, передающее вращение от мотора к роботу.

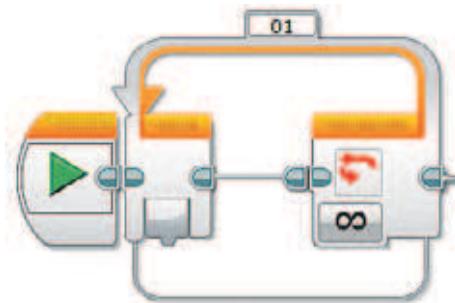
## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОТА-ИЗМЕРИТЕЛЯ

В открывшемся окне проекта начинай составлять программу для робота-измерителя.

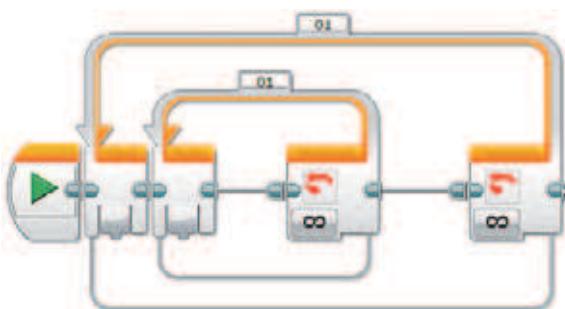
### Общий вид программы



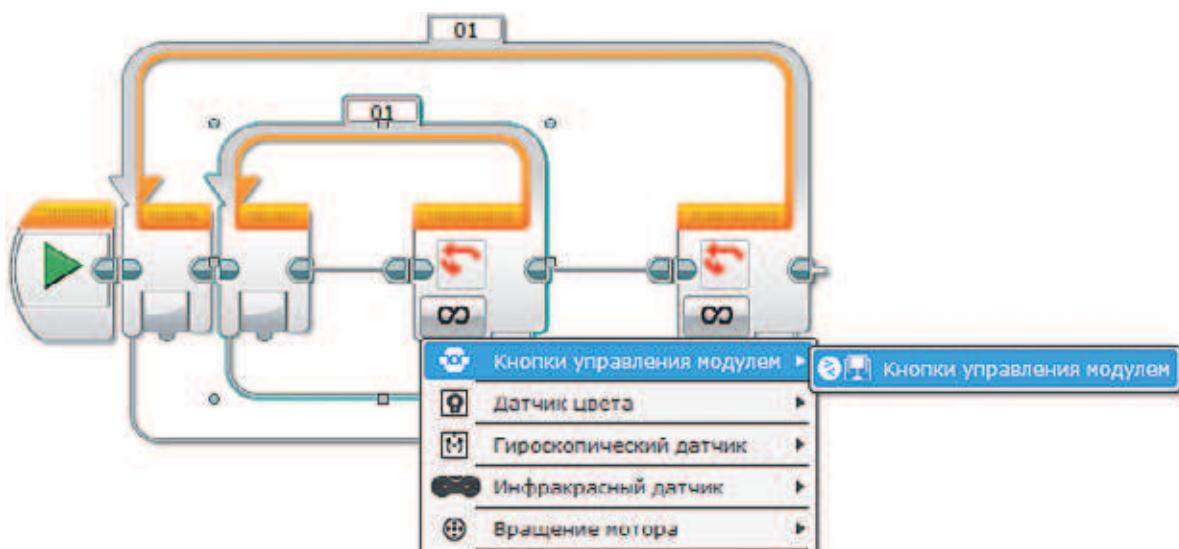
1. Для начала добавь **Цикл** (оранжевый блок), так как робот-измеритель должен уметь делать несколько разных замеров. Этот цикл будет **внешним** и бесконечным.



2. Помести внутри этого цикла ещё один **Цикл** (оранжевый блок), так как помимо неограниченного количества замеров, которые будет делать робот, ты ещё должен иметь возможность сбрасывать его показатели для новых измерений. Этот цикл называется **вложенным**.

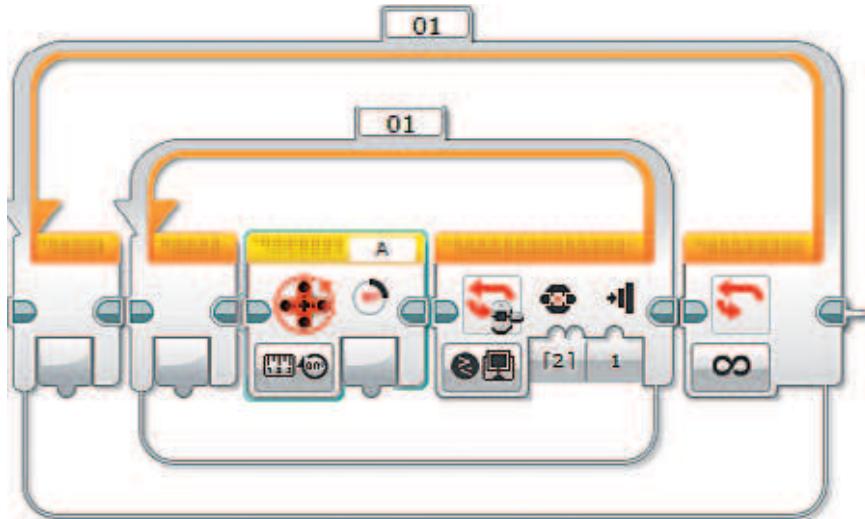


3. Вложенный цикл нужно настроить: выбери в нём опцию **Кнопки управления модулем** → **Кнопки управления модулем**.

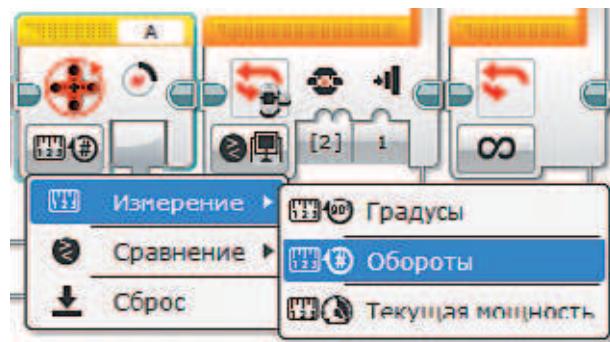


Настройки этого цикла оставь по умолчанию. Они говорят о том, что цикл завершится, как только будет нажата кнопка **OK** на программируемом модуле.

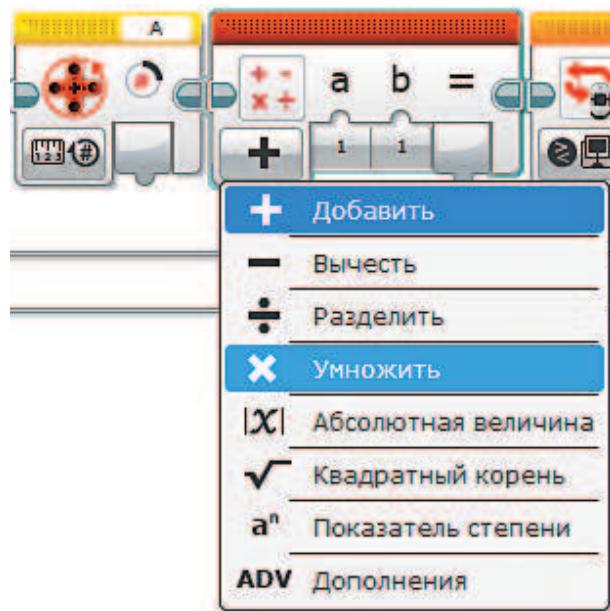
4. Теперь приступай к написанию основной программы, которая будет внутри вложенного цикла. С её помощью робот-измеритель сможет выполнять свою функцию. Добавь команду **Вращение мотора** (жёлтый блок) — с её помощью робот будет снимать показания мотора, к которому подсоединенено колесо. Буква в правом верхнем окошке данного блока должна совпадать с именем порта, к которому подключен мотор!



5. Выбери для команды **Вращение мотора** опцию **Измерение** → → **Обороты**.



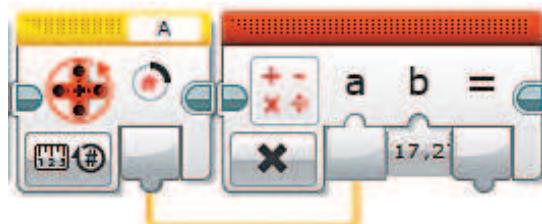
6. Теперь реализуй подсчёт по формуле, согласно которой тебе нужно умножить длину окружности колеса на количество совершённых оборотов. Добавь команду **Математика** (красный блок) и выбери опцию **Умножить**.

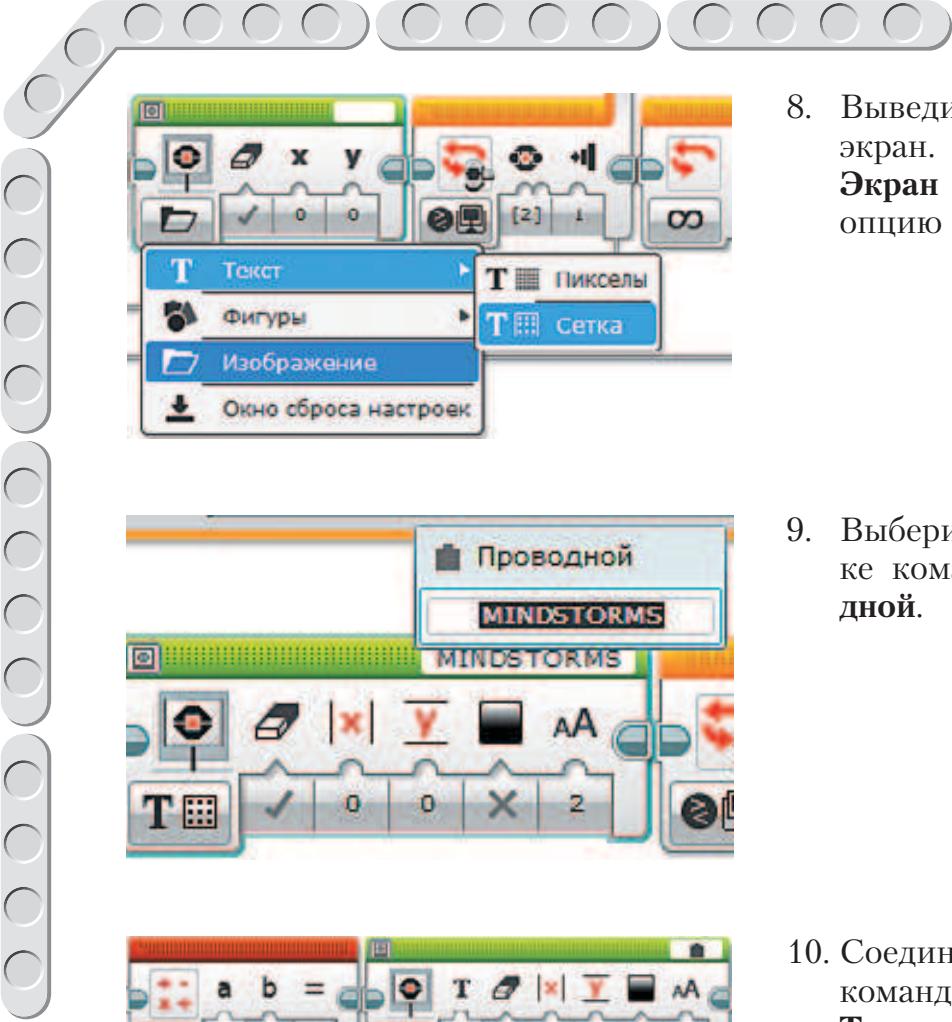


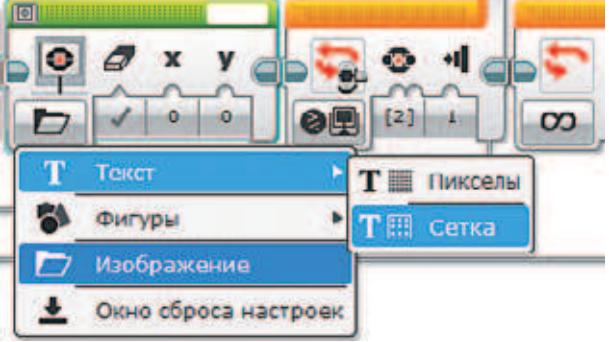
7. Соедини с помощью мыши выход команды **Вращение мотора** с входом **a** команды **Математика**. Запиши на выходе **b** с клавиатуры число **17,27** — это ранее вычисленная тобой длина окружности колеса.

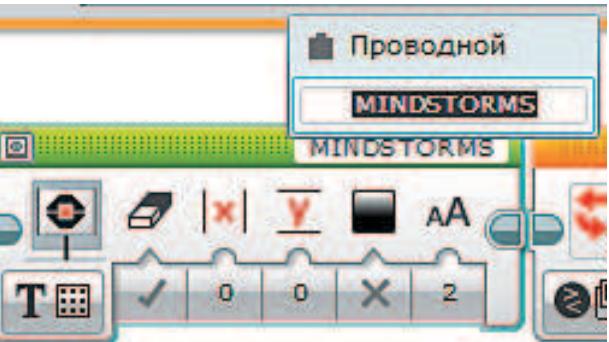
#### Внимание!

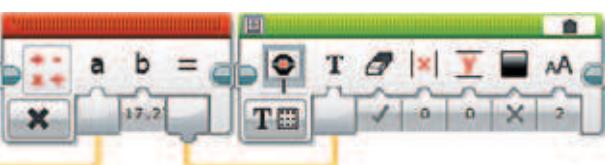
В зависимости от того, в какую сторону ты будешь вращать колесо робота-измерителя, перед числом 17,27 должен или не должен стоять знак «минус»! Ты это проверишь после того, как запустишь программу. При необходимости надо будет поставить знак «минус».



- 
8. Выведи полученное значение на экран. Для этого добавь команду **Экран** (зелёный блок) и выбери опцию **Текст → Сетка**.

- 
9. Выбери в правом верхнем окошке команды **Экран** пункт **Проводной**.

- 
10. Соедини с помощью мыши выход команды **Математика** с входом **Текст** команды **Экран**.

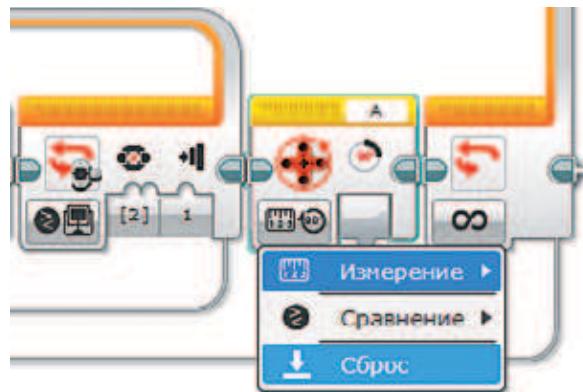
- 
- 
11. Задай в координатах **x** и **y** то место, где на экране будут располагаться значения измерений (удобнее разместить их в центре экрана). Для этого задай с клавиатуры для выходов **x** и **y** соответственно значения **7** и **7**.

12. Теперь, чтобы обеспечить возможность сброса значений на твоём измерителе при выходе из внутреннего цикла, необходимо сбросить показания мотора нажатием кнопки **OK**.

Для этого после окончания внутреннего цикла добавь команду **Вращение мотора** (жёлтый блок) и выбери опцию **Сброс**. Команда окажется точно между двумя циклами.

#### Внимание!

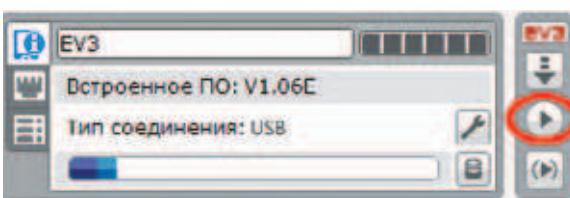
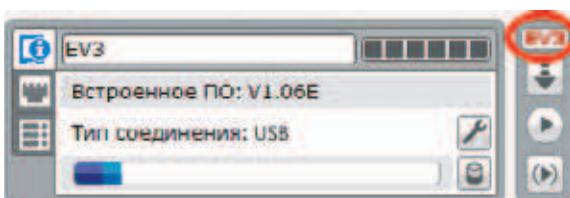
Буква в правом верхнем окошке этой команды должна совпадать с буквой порта на программируемом модуле, к которому подключён мотор.



# Этап 5. Загрузка программы и её тестирование



## ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ



Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

## ТЕСТИРОВАНИЕ

### Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

Если ты хочешь повторно запустить программу или продемонстрировать работу своего робота другу, то для его запуска компьютер тебе уже не нужен: включи программируемый блок и выбери папку **MenLimitAll** и в ней программу **Meter**.

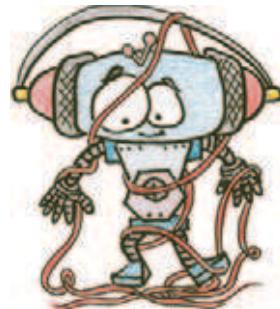
1. Измерь длину ладони с помощью линейки.
2. Проведи колесом измерителя по ладони.
3. Сравни результат на экране модуля с результатом измерения линейкой.

Если они одинаковы, то устройство функционирует правильно (при этом допустимы незначительные погрешности).

Если нет — скорее всего были допущены ошибки при сборке или написании программы. Начни с проверки программы.



## Этап 6. Человеческий фактор исключён!



Давай проверим, насколько точно человек может измерять своими мерами те или иные объекты. Для примера сравни результаты измерения длины твоего стола старинными мерами (вершком, малой пядью, великой пядью и пядью с кувырком) и роботом.

1. Измерь линейкой собственные меры: вершок, малую пядь, великую пядь и пядь с кувырком.
2. Запиши их значения в сантиметрах в скобках в таблице 1.
3. Посчитай, сколько твоих вершков, великих и малых пядей, пядей с кувырком составляет длина твоего стола. Запиши полученный результат в таблицу, в строке «Сколько мер уместилось».
4. Переведи результаты измерений в сантиметры с помощью формулы:

$$\boxed{\begin{array}{c} \text{Длина} \\ \text{стола} \\ (\text{в сантиметрах}) \end{array}} = \boxed{\begin{array}{c} \text{Твой} \\ \text{вершок} \\ (\text{в сантиметрах}) \end{array}} \times \boxed{\begin{array}{c} \text{Количество} \\ \text{вершков} \\ \text{в длине стола} \end{array}}$$

5. Переведи в сантиметры длину стола и по другим твоим мерам. Заполни до конца строку «Перевод в сантиметры» таблицы. А теперь настало время проверить точность измерений с помощью твоего робота-измерителя.
6. Проведи колесом робота вдоль стола. Запиши результаты в таблицу.

**Таблица 1**

	Вершок ( см)	Малая пядь ( см)	Великая пядь ( см)	Пядь с кувырком ( см)
Сколько мер уместилось				
Перевод в сантиметры				
Показания робота-измерителя				

Ну как? Что у тебя получилось? Отличаются ли показания робота от результатов твоих измерений? Как считаешь, можно ли пользоваться «человеческими» мерами в жизни, когда нет под рукой измерительных инструментов?

Теперь проделай то же самое с другими единицами измерения: локтем, аршином, маховой саженью, косой саженью. Ты можешь измерить, например, размер дверного проёма или даже рост своего друга. Может быть, именно у него окажется богатырская косая сажень в плечах? Или у тебя? Результаты замеров запиши в таблицу 2.

**Таблица 2**

	Локоть ( см)	Аршин ( см)	Маховая сажень ( см)	Косая сажень ( см)
Сколько мер уместилось				
Перевод в сантиметры				
Показания робота-измерителя				

### **Возможности твоего робота безграничны! Попробуй!**

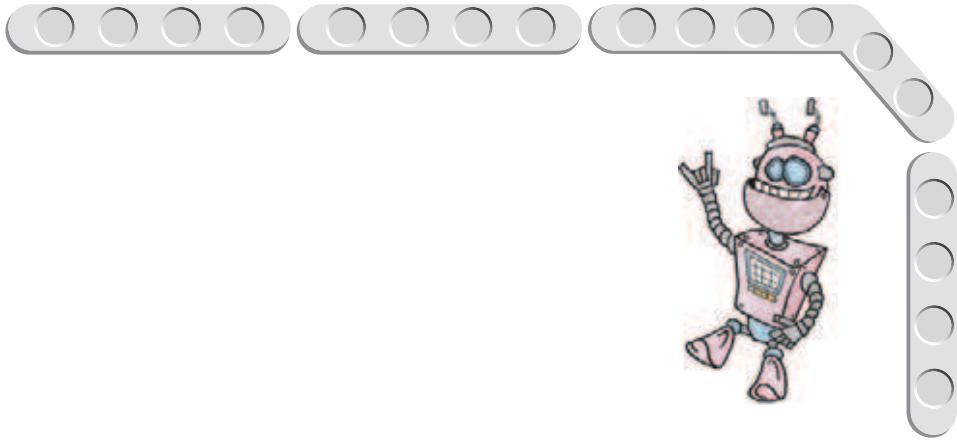
Ты можешь измерять любые поверхности в любых единицах измерения — для этого тебе надо только изменить программу. Сделай так, чтобы значения выводились на экран в дециметрах или дюймах!

### **Робот-измеритель может вычислять периметр и площадь правильных геометрических фигур!**

Измени программу так, чтобы робот подсчитывал их, зная не все стороны, но помни, что формулы подсчёта для разных фигур отличаются, и тебе надо будет учесть это в программе.

### **Сделай робота «человеком»!**

Твой робот может показывать результаты измерений не в сантиметрах, а в «человеческих» единицах измерения. Это довольно просто! Нужно лишь сделать перевод сантиметров, например, в пяди! Ты справишься!



# А теперь...



Перенеси длину своих мер в сантиметрах в таблицу. Если тебе понадобится что-то измерить, твой измеритель всегда с тобой!



ИМЯ, ФАМИЛИЯ

Дата			
Вершок			
Малая пядь			
Великая пядь			
Пядь с кувырком			
Локоть			
Аршин			
Маховая сажень			
Косая сажень			

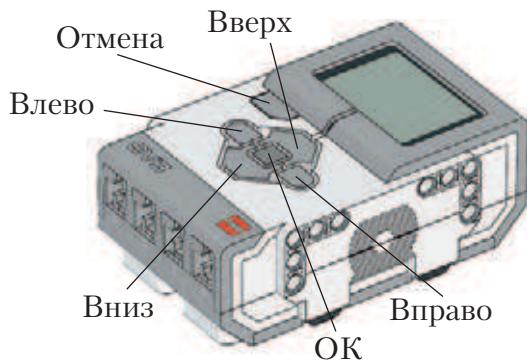


Дата в таблице указана неслучайно: пока ты растёшь, твои меры будут постоянно увеличиваться! Не забывай об этом и регулярно проверяй. Заодно узнаешь, как ты растёшь!

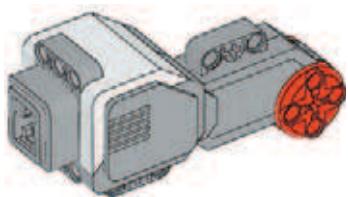
А ещё можешь проверить сам, действительно ли длина окружности в 3,14 раза больше её диаметра. Поэкспериментируй. Может, древние строители ошибались?

## Перечень деталей

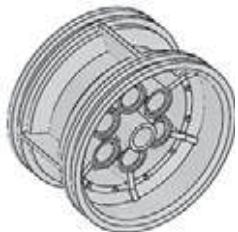
Модуль EV3, 1x



Большой мотор, 1x



Ступица, 43,2×26 мм,  
серая, 1x



Низкопрофильная шина, 56×28 мм,  
чёрная, 1x



## Кабель

Кабель, 35 см, 1x



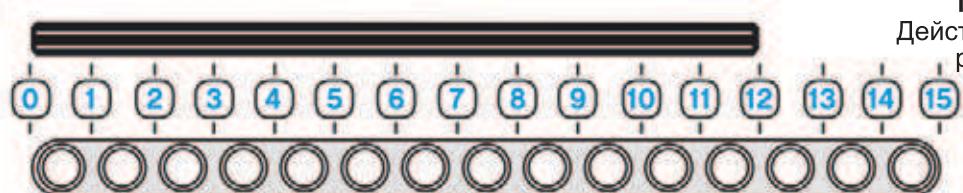
## Фиксаторы

Фиксатор, 3-модульный,  
чёрный, 1x



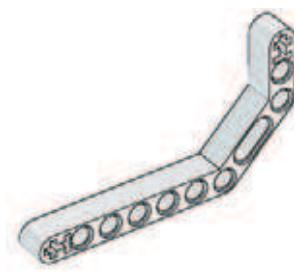
M 1:1

Действительный  
размер



### **Балки угловые**

Двойная угловая балка  $3 \times 7$ ,  
белая, 1x

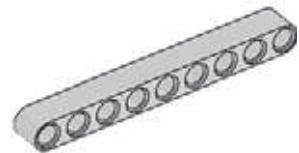


Прямоугольная балка  $2 \times 4$ ,  
красная, 1x



### **Балки прямые**

Балка № 9, серая, 1x



Балка № 3, чёрная, 2x



### **Штифты**

Соединительный штифт,  
2-модульный, чёрный, 8x



Соединительный штифт ,  
2-модульный, синий, 5x



Соединительный штифт,  
3-модульный, синий, 2x



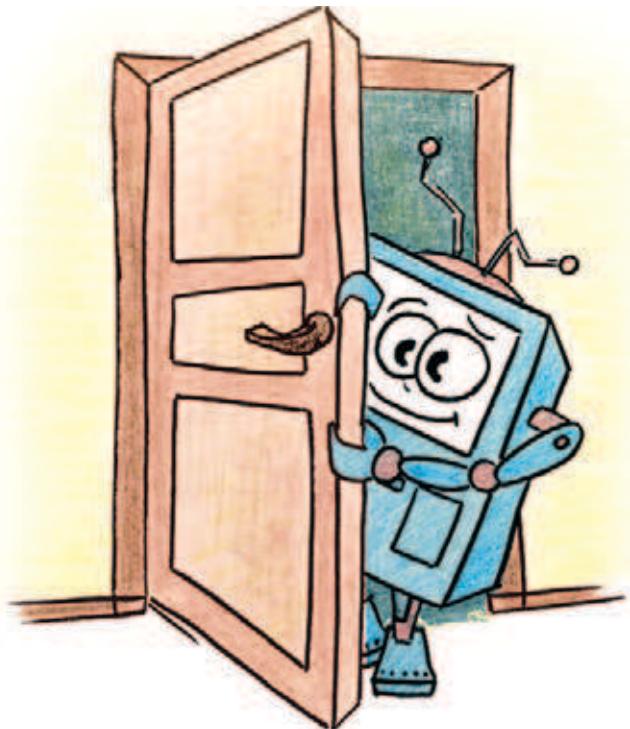
### **Ось**

Ось № 6, чёрная, 1x





Посторонним  
вход воспрещён!





# История развития систем контроля доступа



Во все времена люди старались защитить свой дом и своё имущество от незваных гостей. В XXI веке для контроля доступа в помещение применяются самые новые и сложные технологии, например распознавание голоса или сканирование сетчатки глаза.

А как было раньше? Как люди в давние времена защищали своё имущество?

Первые замки упоминаются даже в мифах и Ветхом Завете. Их делали из таких материалов, что даже нельзя представить, как они держались! Это были и дерево, и ткань, и даже камыш!

Первые ключи были найдены в гробнице Рамсеса II, египетского фараона. В гробницах стояли ловушки, которые очень чётко взаимодействовали. Это были самые настоящие системы контроля доступа!

В Древнем Китае делали железные и медные висячие замки, которые используются и сейчас. Очень часто замки украшали узорами или придавали им форму животных. Такой замок с поздравительными словами можно было получить в подарок. Уже в 870–900 годах в Англии появились цельнометаллические замки.

Потом пошла настоящая лавина изобретаемых замков. Появились кодовые замки. Одним из известнейших был кодовый замок «Эврика». Чтобы открыть его, не зная кода, нужно было перебрать 1 073 741 824 комбинаций, а это заняло бы более 2000 лет. Этим замком когда-то закрывали сейф Казначейства США.

Во времена царствования Николая II был выкован русский висячий замок. Хитрость замка была в том, что когда замок запирали, в замочную скважину вставляли «хитрую» пробку. Создавалось впечатление, что замочной скважины просто нет. А ключом служила округлая скоба сверху, на ней была специальная нарезка, которая открывала замок.

Сейфы тоже являются частью системы контроля доступа. Сейчас даже проводят соревнования по взлому сейфов. В США проводился такой конкурс, и победила 21-летняя девушка. Призом была сумма в 100 000 долларов. Представляете? Эти сейфы годами совершенствовали, а один человек за один раз разрушил все труды изобретателей.

В наше время население земного шара достигло почти 7,5 миллиардов, не счесть городов-миллионников. В состав инфраструктуры со-



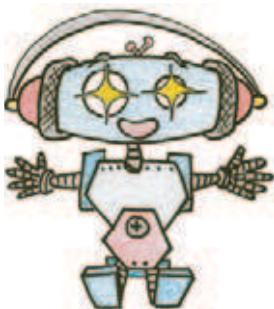
временного города входят офисные центры, банки, больницы и школы, промышленные предприятия, магазины, жилые и спортивные комплексы, парковки, музеи, метрополитен и многое другое. А легко ли попасть на их территорию? В магазины, например, вход свободный, вход в коттеджи может охранять злая собака или охранник. Въезд автомобилю на территорию завода может быть преграждён шлагбаумом, а проход человеку — турникетом.

Турникеты устанавливают везде, где требуется организовать контролируемый проход большого количества людей. Турникеты — часть **системы контроля и управления доступом (СКУД)**. Такая система решает, кого и в какое время пускать на заданную территорию через точки прохода — двери, ворота, контрольно-пропускные пункты. Мозг системы — контроллер, а ключи для входа — карточки или электронные ключи (брелоки), которые выдают людям, имеющим допуск на территорию. Ещё в состав системы входит считыватель карт. Он считывает информацию с брелоков и передаёт в контроллер для обработки. Над созданием современной системы контроля доступа работают инженеры-изобретатели. Попробуй и ты создать свою охранную систему.

Ты соберёшь модель системы контроля доступа СКД, с помощью которой можно определять количество людей, вошедших и вышедших из помещения за определённое время.



## Этап 1. Принцип работы системы контроля доступа

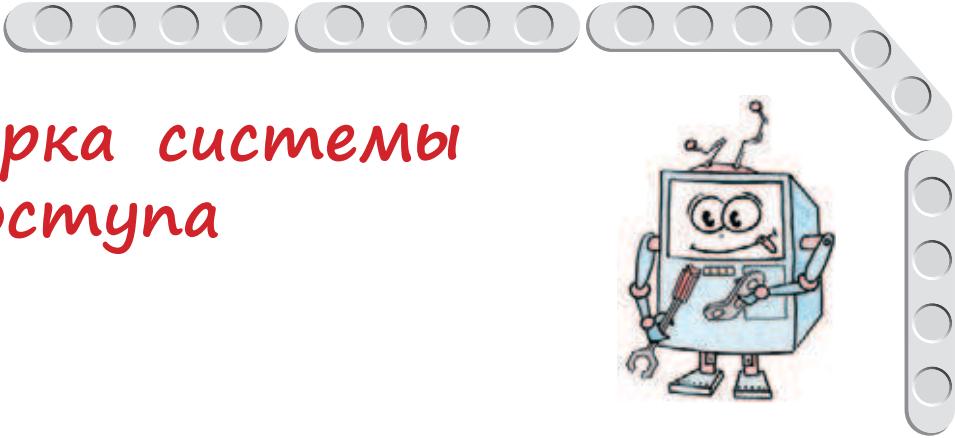


Как ты уже знаешь, мозгом системы является контроллер.

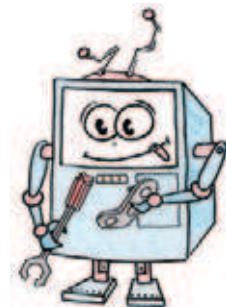
**Контроллер** — это блок управления, выполняющий логические операции. Это устройство сродни компьютеру, который использует считающее устройство для ввода информации. Но что общего у оборудования СКД и нашего набора LEGO® MINDSTORMS® Education EV3? Дело в том, что программируемый модуль EV3 также является своего рода компьютером: у него есть экран, память и процессор. В качестве устройства ввода информации могут быть использованы различные датчики. У насчитывающим устройством будет ультразвуковой датчик (УЗ-датчик).

Установи своего «сторожа» около двери, чтобы датчик «видел» всех, кто входит в комнату и выходит из неё. Датчик будет отмечать вошедших и вышедших из помещения людей, а программируемый модуль — вести подсчёт и выводить результат на экран.





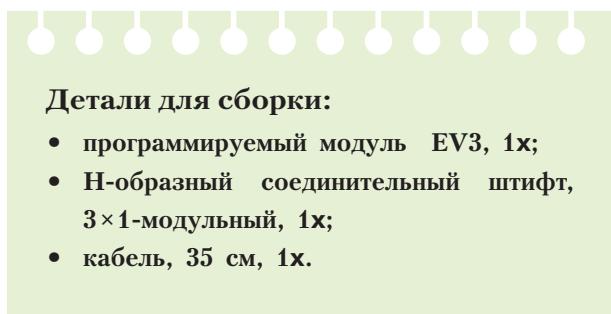
## Этап 2. Сборка системы контроля доступа



Теперь начинай собирать систему. Внимательно рассмотри рисунки и прочитай подписи к ним. Если что-то непонятно, обращайся за помощью к взрослым.

Обрати внимание, в конце книги даны все детали, которые потребуются тебе для сборки. Страницу можно отрезать.

### ШАГ 1. УСТАНОВКА Н-ОБРАЗНОГО ШТИФТА И ПОДКЛЮЧЕНИЕ КАБЕЛЯ К МОДУЛЮ



1. Установи с левой стороны программируемого модуля Н-образный соединительный штифт.
2. Подключи кабель к порту № 3. Теперь кабель подключён к модулю.



## ШАГ 2. УСТАНОВКА СИСТЕМЫ КРЕПЛЕНИЯ УЗ-ДАТЧИКА НА МОДУЛЕ



Детали для сборки:

- балка № 5, серая, 1x;
- соединительный штифт, 2-модульный, чёрный, 2x.



1. Закрепи на установленный в предыдущем шаге H-образный штифт балку № 5.

Штифт должен пройти через второе и четвёртое отверстия балки.

2. Установи два чёрных штифта в отверстия H-образного штифта. Система крепления УЗ-датчика к модулю готова.

## ШАГ 3. СБОРКА УЗ-ДАТЧИКА



Детали для сборки:

- соединительный штифт, 2-модульный, синий, 1x;
- соединительный штифт, 2-модульный, чёрный, 1x;
- УЗ-датчик, 1x.

- Установи в креплении датчика синий и чёрный штифты через один модуль.

УЗ-датчик готов.



#### ШАГ 4. ПОДГОТОВКА СИСТЕМЫ КРЕПЛЕНИЯ УЗ-ДАТЧИКА

Детали для сборки:

- УЗ-датчик, 1x;
- соединительный штифт, 2-модульный, чёрный, 2x;
- двойная угловая балка 3 × 7, 1x.



- Присоедини белую двойную угловую балку к датчику.

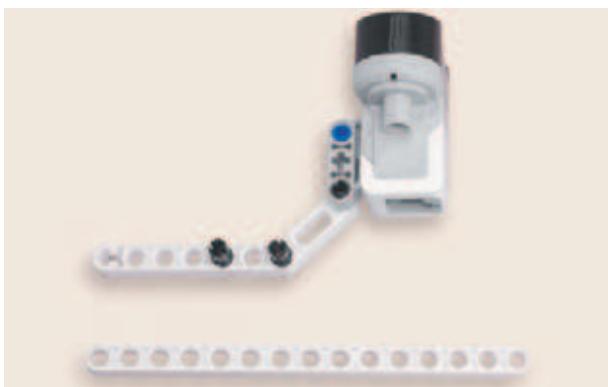


- Установи чёрные штифты в пятом и седьмом модулях (считая от себя) этой балки.



Подготовка системы крепления УЗ-датчика завершена.

## ШАГ 5. СБОРКА СИСТЕМЫ КРЕПЛЕНИЯ ДАТЧИКА



Детали для сборки:

- УЗ-датчик, 1x;
- балка № 15, белая, 1x.



1. Закрепи балку № 15: штифты должны пройти через тринадцатый и пятнадцатый модули балки № 15.

Система крепления датчика готова.

## ШАГ 6. ПРИСОЕДИНЕНИЕ ДАТЧИКА К МОДУЛЮ

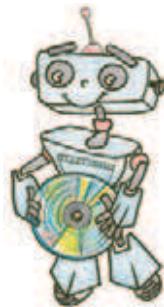


1. Прикрепи собранную на предыдущем шаге конструкцию к программируемому модулю: два чёрных штифта, установленные на программируемом модуле, должны войти в третий и пятый модули балки № 15 (считая слева направо).



2. Подключи кабель, идущий от порта № 3, к датчику.

Теперь датчик присоединён к модулю. СКД готова!

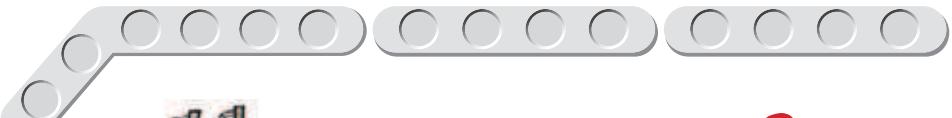


## Этап 3. Установка программного обеспечения на компьютере

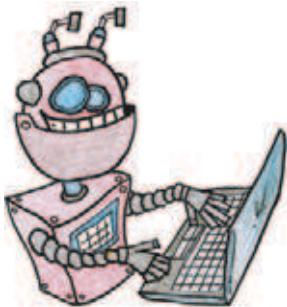
1. Если ты приобрёл базовый набор LEGO® Mindstorms® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

### Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.



## Этап 4. Создание программы для системы контроля доступа



### ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

### СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ СКД

В открывшемся окне проекта начни составлять программу для СКД.

### ЛОГИКА ПРОГРАММЫ

Создаются (объявляются) переменные **COUNT** и **VISITOR** и обнуляются. Первая будет считать количество проходов мимо датчика СКД, вторая — количество людей путём деления количества проходов на 2 и округлением результата деления в большую сторону.

Далее будет следовать пустой цикл, завершающийся, когда в поле зрения датчика окажется какой-либо объект (человек) на расстоянии, меньшем 50 см.

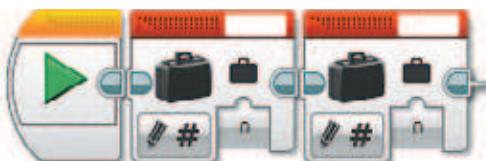
Как только это произойдёт, программа увеличит значение переменной-счётчика **COUNT** на 1, поделит это значение на 2, округлит и запишет в переменную **VISITOR**. После этого информация о срабатывании системы и о количестве прошедших людей будет выводиться на экран программируемого модуля EV3.

Чтобы избежать ложных срабатываний системы в случае остановки объекта в поле зрения датчика, в программу вводится блок ожидания.

Как только система придёт в состояние ожидания (охраны), программа повторится.

Таким образом, становится понятно, что программа циклична; её необходимо поместить в бесконечном цикле, **кроме двух первых команд обнуления переменных**.

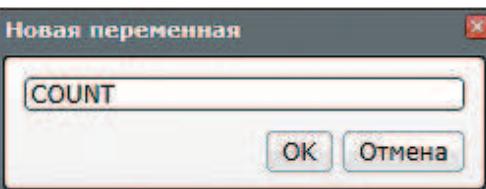
1. Прежде всего тебе понадобятся переменные **COUNT** и **VISITOR**. Добавь две команды **Переменная** (красный блок) в программу.



2. Теперь нужно создать каждую переменную. Кликни в пустую область в правом верхнем углу первого блока **Переменная** и выбери пункт **Добавить переменную**.



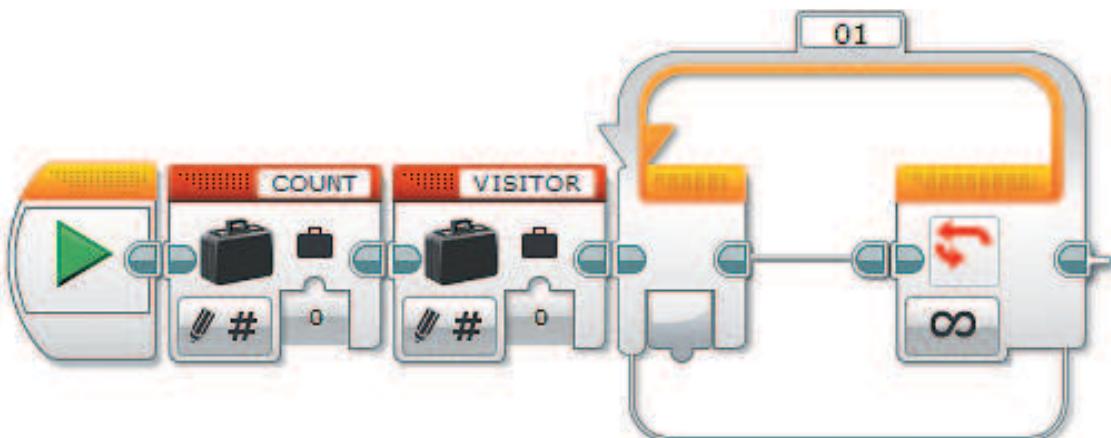
3. Введи в открывшемся окне название переменной **COUNT**, затем нажми **OK**.



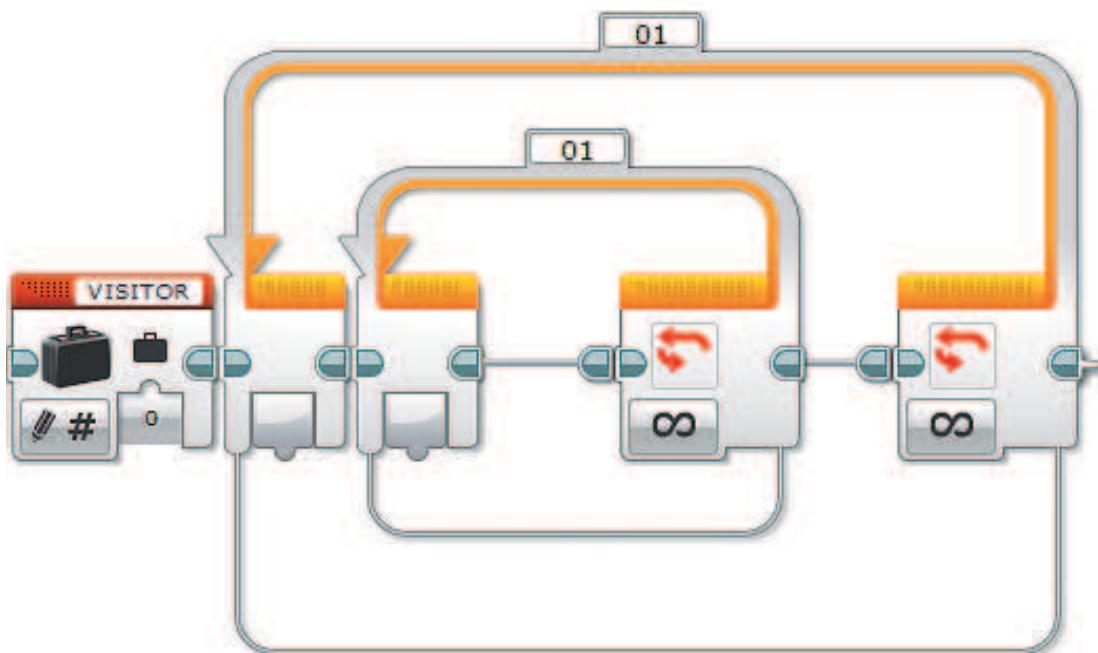
4. Аналогично создай переменную **VISITOR**. В результате получится так:



5. Далее тебе понадобится цикл (оранжевый блок). Так как работа системы не должна прерываться, по умолчанию выбери бесконечный цикл. Эту настройку дальше не менять.  
Остальная часть программы будет содержаться внутри этого цикла.



6. Внутрь бесконечного цикла добавь ещё один цикл (оранжевый блок).



7. Этот цикл надо настроить так, чтобы он завершался, если кто-либо приблизится к датчику расстояния (ультразвуковому датчику). Для этого выбери в опциях этого цикла **Ультразвуковой датчик** → → **Расстояние в сантиметрах**.

8. У цикла появятся два параметра: **Тип сравнения** и **Пороговое значение**. Именно они будут задавать условие выхода из него.

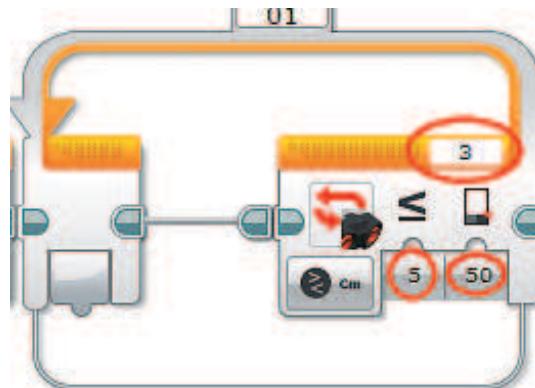


9. Выбери **Тип сравнения: 5** (меньше или равно). **Пороговое значение** оставь равным **50**.

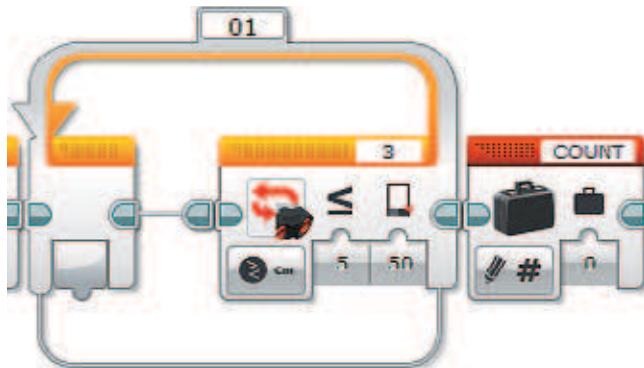
Внутри этого цикла не будет никаких команд, потому что, если никто не проходит через зону действия датчика, система не должна ничего делать.

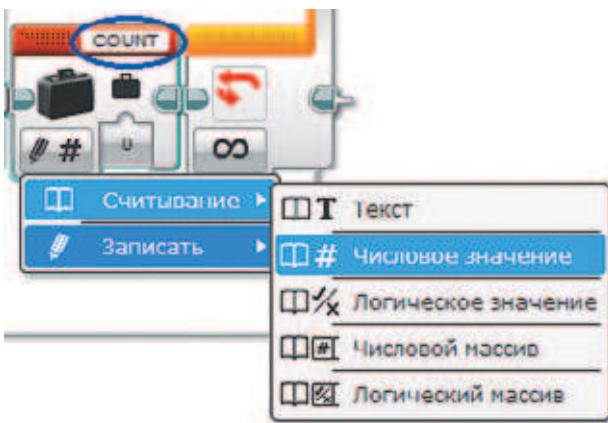
#### Внимание!

Не забудь проверить: датчик должен быть подключён к порту 3! При необходимости измени его.



10. Теперь напиши остальную часть программы, то есть если кто-то вошёл в комнату и внутренний цикл завершился. Сначала ты должен увеличить значение переменной счётчика **COUNT** на единицу. Добавь после внутреннего цикла команду **Переменная** (красный блок).

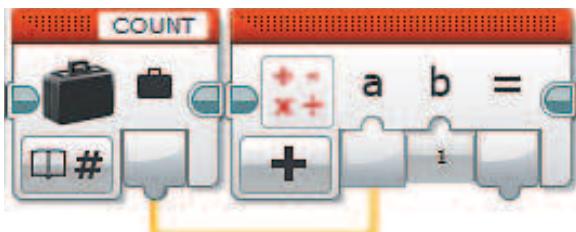




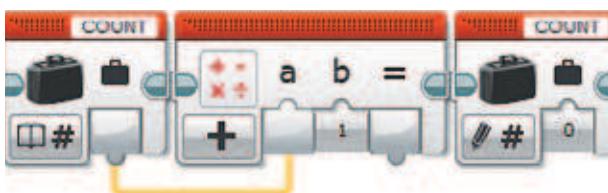
11. Проверь, что выбрана переменная **COUNT**. Измени опцию команды: Считывание → Числовое значение.



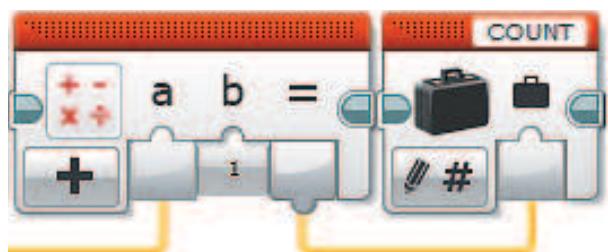
12. Добавь команду **Математика** (красный блок). По умолчанию она настроена на сложение двух чисел.



13. Соедини с помощью мыши выход переменной **COUNT** с входом **a** команды **Математика**. Вход **b** оставь без изменения.



14. Добавь ещё одну команду **Переменная** (красный блок). Проверь, что выбрана переменная **COUNT**.



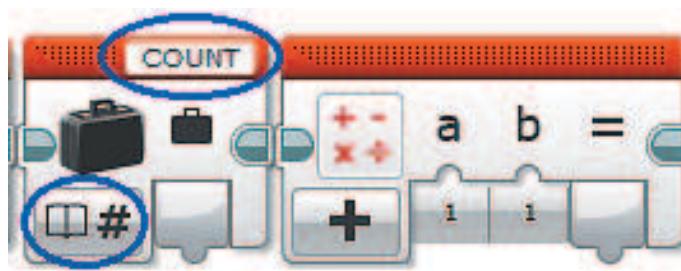
15. Соедини выход «=» команды **Математика** со входом команды **Переменная**.

Это значит, что предыдущее значение переменной **COUNT** стирается и записывается новое, увеличенное на 1 по сравнению с предыдущим.

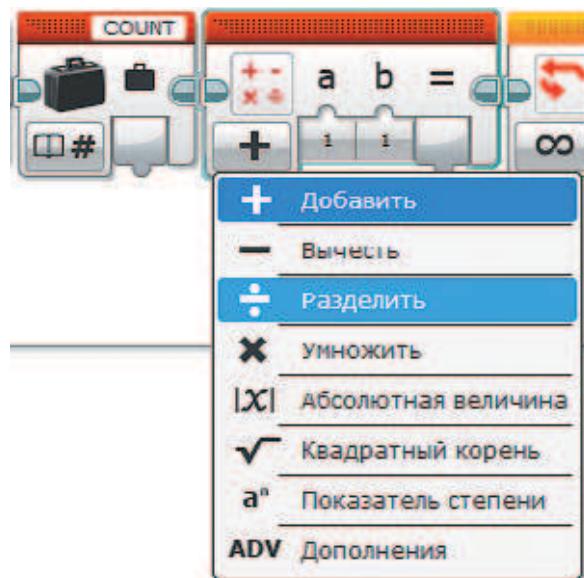
16. Так как мы хотим выводить на экран не просто количество проходов мимо датчика, а количество людей, то нам нужно преобразовать полученное значение. Будем считать, что два прохода мимо датчика — это вход и выход одного человека. Значит, чтобы посчитать общее количество людей, тебе надо разделить значение переменной **COUNT** на 2.

Для этого добавь ещё по одной команде **Переменная** и **Математика** (красный блок). Настрой команду **Переменная**, как в шаге 11: **Считывание** → **Числовое значение**.

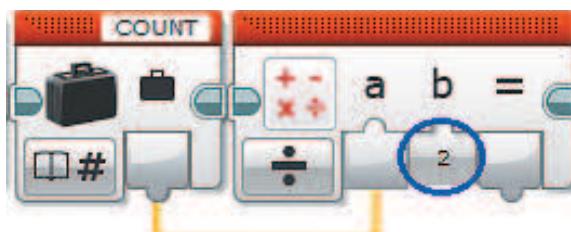
Не забудь проверить, что выбрана переменная **COUNT**!



17. Измени опцию команды **Математика** на **Разделить**.



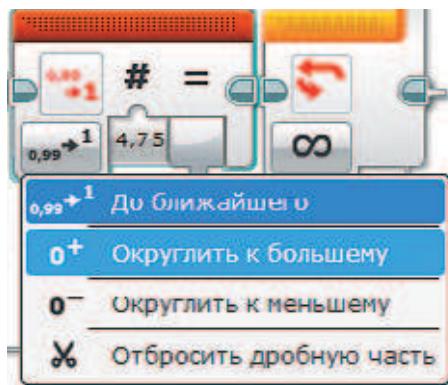
18. Осталось соединить выход переменной **COUNT** со входом **a** команды **Математика**, и на входе **b** вручную записать цифру **2**.



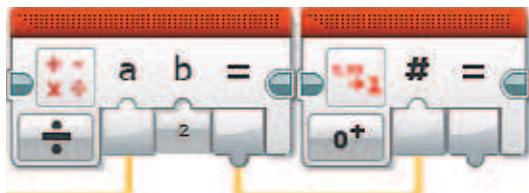


19. Теперь надо подумать. Результат вычислений может оказаться нецелым, а такого быть не может! (Вспомни мультфильм, где были полтора землекопа!) Значит, прежде чем работать с этим числом, тебе надо его округлить. Округлять надо в большую сторону, так как датчик может иметь погрешности в чувствительности измерений.

Добавь команду **Округление** (красный блок).



20. Измени опцию этой команды на **Округлить к большему**.

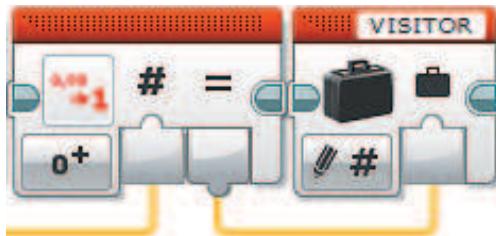


21. Соедини выход «==» команды **Математика** со входом команды **Округление**.



22. Запиши получившееся значение в переменную **VISITOR** — именно она отвечает за подсчёт людей. Добавь команду **Переменная** (красный блок). Выбери переменную **VISITOR**. Опцию менять не нужно, так как по умолчанию она стоит на запись числового значения.

23. Соедини выход «=>» команды **Округление** со входом переменной **VISITOR**.

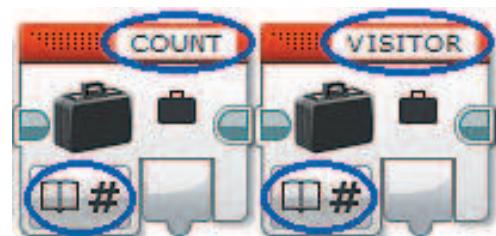


24. Осталось самое интересное – вывести значения на экран твоего EV3!

Добавь ещё две команды **Переменная** (**красный блок**).

В обеих переменных выбери опцию **Считывание** → **Числовое значение**.

Одну из них оставь **COUNT**, а вторую выбери **VISITOR**.



25. Теперь добавь команду **Текст** (**красный блок**). Она сформирует вывод на экран значений обеих переменных.

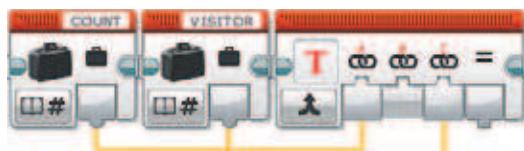


26. Соедини выход переменной **COUNT** со входом **A**.

Во вход **B** вручную поставь **три пробела**.

Выход переменной **VISITOR** соедини со входом **C**.

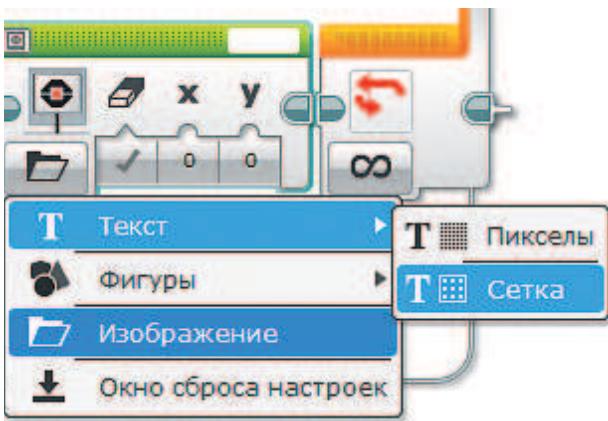
Ты сформировал строку, состоящую из значений переменных **COUNT** и **VISITOR**, разделённых тремя пробелами.



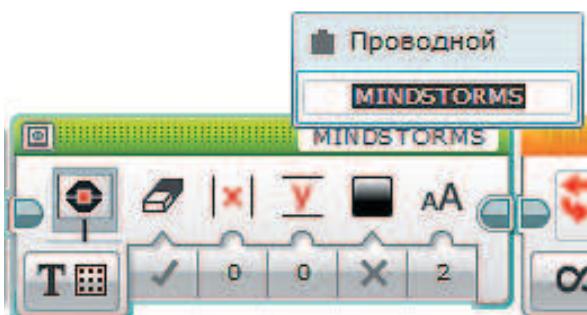
27. Осталось отправить эту строку на экран. Добавь команду **Экран** (**зелёный блок**).



28. Выбери опцию этой команды:  
**Текст** → **Сетка**.



29. Выбери в окне вывода текста (вверху справа) **Проводной**. Это значит, что текст для вывода на экран будет поступать из другого источника, а не прописан заранее.



30. Настрой каждый параметр команды **Экран** (слева направо).

**Очистить экран** — оставь галочку (**истина**), так как старые значения должны каждый раз удаляться.

**Координата X** (столбец) — поставь цифру **3**.

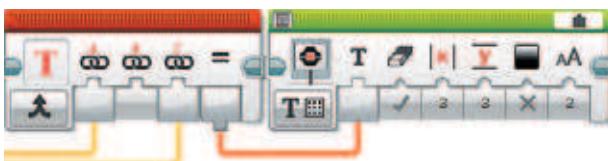
**Координата Y** (строка) — тоже поставь цифру **3**.

Теперь текст будет выводиться на середину экрана.

Остальные параметры оставь **без изменений**.



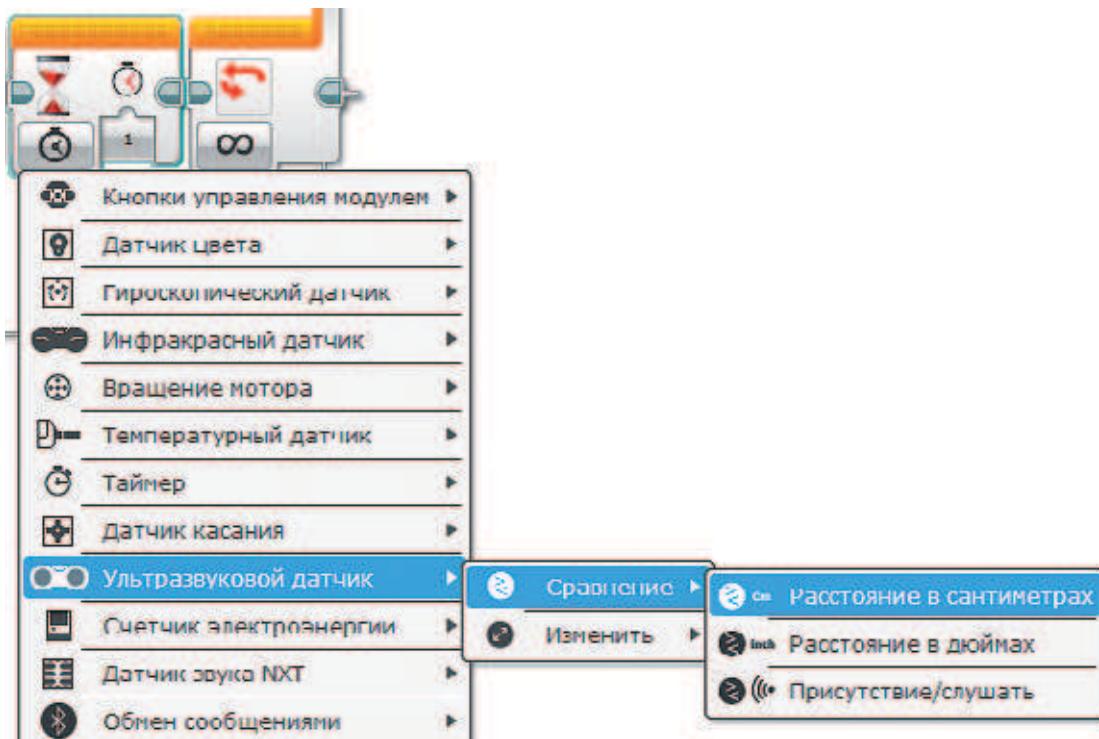
31. Соедини выход «==» команды **Текст** со входом **T** команды **Экран**.



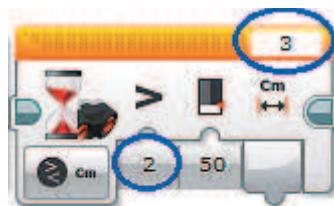
32. Финальный штрих твоей программы! Осталось задать ожидание для датчика до того, как он вернётся в обычный рабочий режим, то есть замеряемое расстояние до ближайшего объекта снова станет больше 50 см.  
Добавь команду **Ожидание** (оранжевый блок).



33. Выбери опцию **Ультразвуковой датчик** → **Сравнение** → **Расстояние в сантиметрах**.



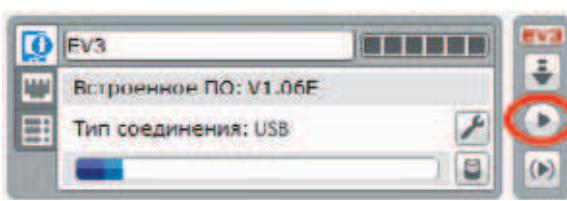
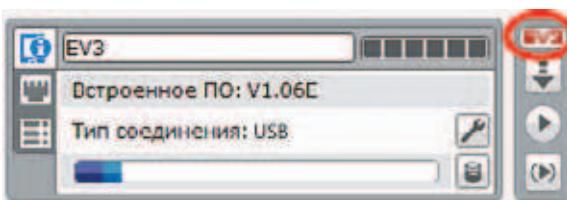
34. Настрой параметры команды (слева направо) **Тип сравнения** → **Больше (2)**. Пороговое значение оставь равным **50**.  
Проверь, что команда настроена на **порт № 3!**  
Программа для системы контроля доступа готова!



# Этап 5. Загрузка программы и её тестирование



## ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ



Загрузи свою программу.

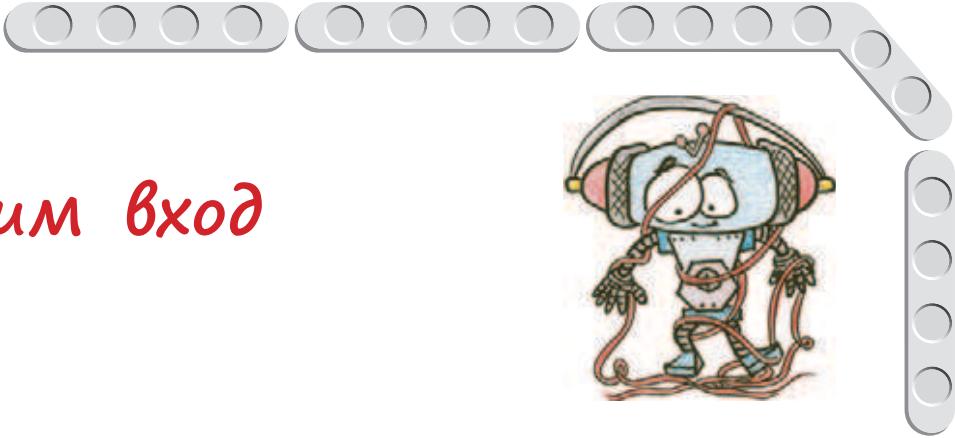
1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

## ТЕСТИРОВАНИЕ

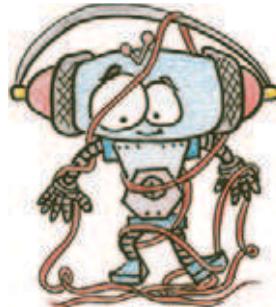
### Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

1. Проведи рукой на небольшом расстоянии от датчика твоей системы. Проверь: на экране программируемого модуля EV3 должны появиться цифры 1 и 1. Первая означает количество проходов мимо датчика, вторая — людей.
2. Проведи ещё раз; первое значение изменится, а второе нет. Допускаются небольшие погрешности.



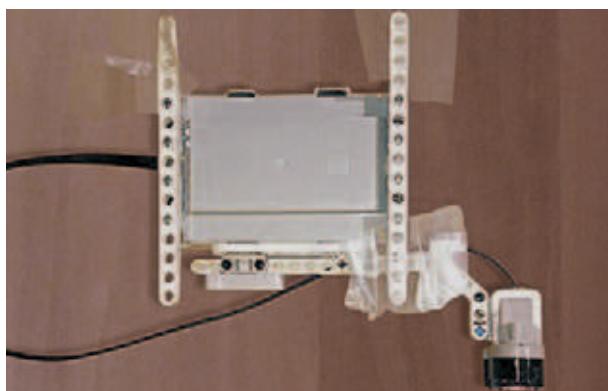
## Этап 6. «Посторонним вход воспрещён!»



1. Закрепи систему около двери внутри комнаты. Например, так: Мы использовали четыре Н-образных штифта, две белые балки № 15 и клейкую ленту.

Расположение СКД возле дверного проёма может быть разным. Постарайся подобрать оптимальное — больших погрешностей в измерениях возникать не должно. Поэтому, прежде чем оставить комнату под охраной, убедись, что система работает исправно: пройди несколько раз зайди в комнату и выйди из неё. Проверь подсчёт системы.

Если что-то не получается, попробуй располагать систему иначе, а также поправляй ультразвуковой датчик.



### Внимание!

Чтобы твоя система работала долгое время, поменяй настройки спящего режима: включи программируемый модуль EV3, выбери меню настроек (значок гаечного ключа) при помощи кнопок Вправо и Влево. В нём с помощью кнопок Вверх и Вниз найди параметр Sleep и поставь его в положение never. Также не забудь зарядить аккумуляторную батарею. Полного заряда батареи хватит примерно на один день.

Молодец! У тебя всё получилось! Ты собрал систему контроля доступа и запрограммировал её.

**А теперь...**



Посмотри, как многому ты научился!

Ты познакомился с технологией программирования действий и можешь вести подсчёт с выводом результатов на экран.

Научился использовать роботов для целей, имеющих практическое значение. В чрезвычайных ситуациях, зная точно, сколько людей находится в помещении, ты сможешь организовать их эвакуацию и т. д.

### **Попробуй поэкспериментировать**

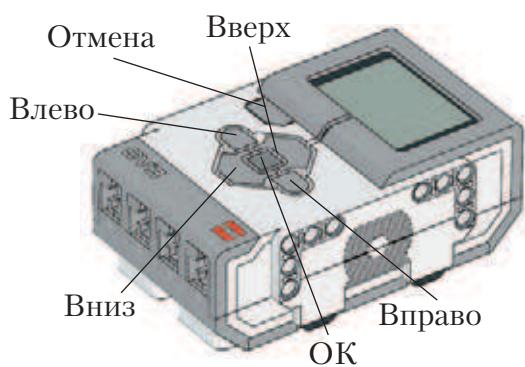
1. Придумай сценарий для следующего тестирования системы и проведи его. Сравни результат, выдаваемый системой, с результатом своего подсчёта.



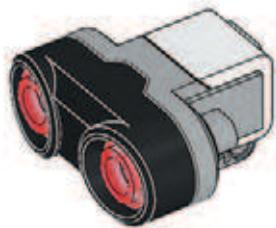
2. Подумай, где ещё можно применить принцип действия данного устройства.
3. Подумай, какие ещё датчики можно использовать в этой системе вместо УЗ-датчиков.

## Перечень деталей

Модуль EV3, 1x



Ультразвуковой датчик, 1x

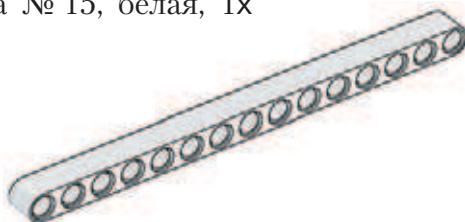


## Балки

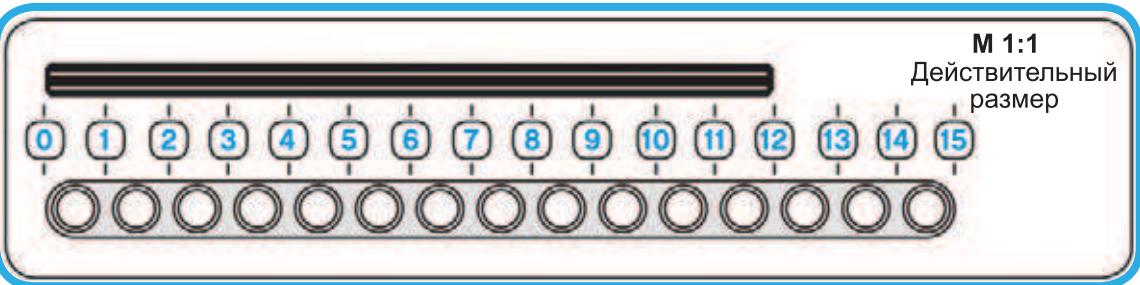
Балка № 5, серая, 1x



Балка № 15, белая, 1x



Двойная угловая балка 3×7, белая, 1x



## Штифты

Соединительный штифт,  
2-модульный, чёрный, 5x



Соединительный штифт,  
2-модульный, синий, 1x



Н-образный соединительный  
штифт, серый, 1x

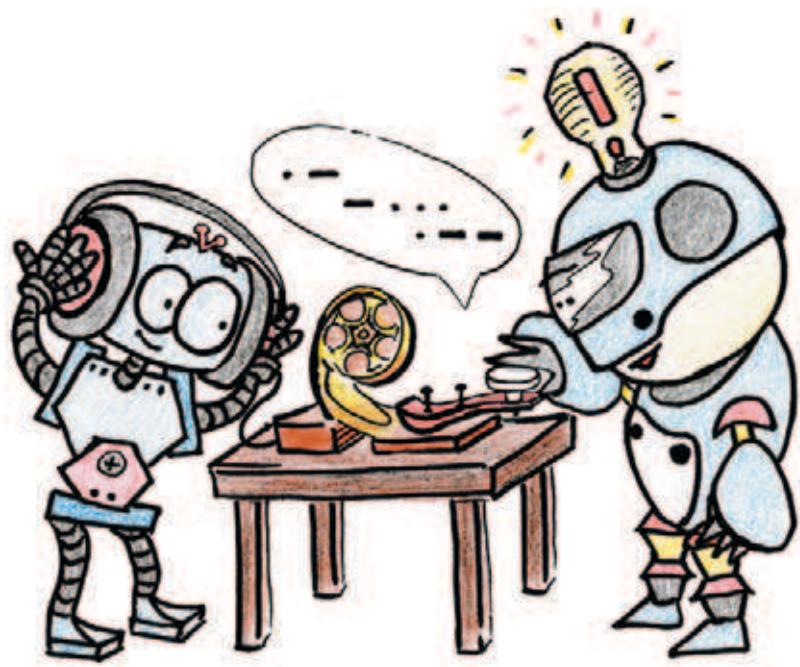


## Кабель

Кабель, 35 см, 1x



# Тайный код Сэмюэла Морзе



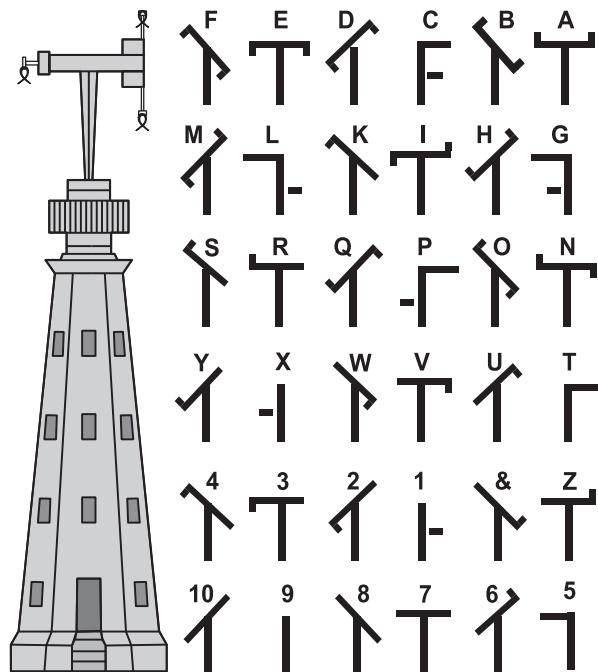
# История телеграфа



Ты, конечно, слышал, что XXI век, в котором мы живём, называется информационным. Это значит, что главным ресурсом в нём является информация.

Информация всегда играла очень важную роль в жизни человека. Говорят, что тот, кто владеет информацией, тот владеет и миром. Иное сообщение стоит дороже жизни. По преданию, 13 сентября 490 года до н. э. греческий воин-гонец, пробежавший из Марафона в Афины более 42 километров не останавливаясь в пути, упал замертво, но донёс весть о победе над персами.

Видишь, как важно, чтобы информация была доставлена вовремя! В Древней Греции информацию доставляли или быстроногие гонцы, или всадники. Других способов передать письмо не было. Но уже тогда существовал «телеграф» — передача сигналов на расстоянии с помощью условных знаков.



**Рис. 1.** Семафорный телеграф и его азбука

В Персии ещё в VI веке до н. э. сообщения от одного пункта к другому передавали звучными голосами рабы, а для усиления звука чуть позднее стали применять рупоры. Для передачи особо важных сообщений в Китае применялись гонги, в Африке и в других странах — барабаны, сигнальные костры и факелы, телескопы и подзорные трубы.

Но всё это были просто условные сигналы, но не слова.

И вот, в конце XVIII века появился оптический семафорный телеграф — передающее устройство из подвижных реек, которое устанавливалось на верху башни. Линия телеграфа состояла из цепочки таких башен, расположенных на расстоянии прямой видимости

друг от друга (рис. 1). Сигналы сообщения передавались последовательно от башни к башне и представляли собой слова, закодированные на языке семафора. Подобным языком пользовались и на флоте, только там были сигнальные флаги (рис. 2).

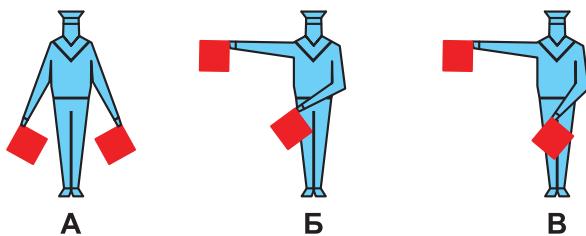
В России линия оптического телеграфа между Петербургом и Варшавой (1839) была самой длинной в Европе: на расстоянии 1200 километров было установлено 149 башен, их обслуживали почти 2000 человек; сигнал доходил за 15 минут, а депеша, состоявшая из 100 сигналов, за 35 минут. Оптический семафор в конце XIX века стали использовать и на железной дороге.

С развитием автомобильного движения появилась упрощённая разновидность оптического семафора — светофор.

Но чем сложнее становился окружающий человека мир, тем острее ощущалась потребность в быстром обмене информацией. Конечно, сейчас невозможно представить, что сообщение адресату может идти несколько недель или даже месяцев. Мы так привыкли к электронной почте и мобильной связи. Но тогда это было реальностью. А в 1837 году широкой публике был показан первый **электромагнитный телеграф**, который изменил представление человека о возможности передачи информации на большие расстояния. Не каждый современник знает о принципе действия телеграфа, но многие слышали словосочетание «азбука Морзе». Изобретателем азбуки и первого электромагнитного телеграфного аппарата стал Сэмюэл Морзе (1791–1872). Он был не знаменитым физиком или известным инженером, а просто — талантливым художником, человеком, который любил мечтать. Во время путешествия из Европы в Америку, Морзе рассказали о «чудесах» электромагнетизма. С этого момента им овладела мечта — создать машину для передачи информации с помощью электричества. После нескольких лет упорного труда, изучения неведомой науки — электротехники, окрылённый великой мечтой, Морзе знакомит мир со своими гениальнейшими изобретениями — телеграфным аппаратом и специальным кодом для передачи сигналов.

В аппарате Морзе для передачи сообщения использовался ключ, при помощи которого в линию посыпались короткие и длинные сигналы. Такой закодированный сигнал стал называться **кодом** или **азбукой Морзе**.

Кроме передачи сообщений делового характера аппарат Морзе в течение 90 лет передавал сигнал SOS — радиосигнал о помощи терпя-



**Рис. 2.** Передача информации на флоте с помощью флаговых сигналов

щим бедствие на море. Этот сигнал состоит из сочетания трёх точек, трёх тире и ещё трёх точек азбуки Морзе<sup>1</sup>.

Как единый сигнал бедствия SOS был принят морским сообществом 3 октября 1906 года. Впервые он прозвучал в эфире в том же, 1906 году, с парохода «Ирбис».

В 1927 году была установлена единая международная частота для передачи сигналов бедствия на море — 500 кГц, и было строго запрещено использовать её для иных трансляций. С момента ввода в действие сигнала SOS в радиоэфире сорок восемь раз в сутки, то есть каждые полчаса, наступали три минуты молчания. В это время радисты всех стран вслушивались в эфир, не раздастся ли от кого-нибудь призыв о помощи.

С 1 февраля 1999 года мир перешёл на новую сигнальную систему, позволяющую осуществлять связь в автоматическом режиме, с применением буквопечатающих устройств и факсимильных аппаратов.

История изобретения телеграфного аппарата и кода к нему — замечательный пример того, что мечта, если упорно трудиться, всегда может стать реальностью!

И ты можешь собрать настоящий звуковой передатчик, создать свой код, присвоив кодовым комбинациям другие значения, зашифровывать им свои сообщения и переписываться с другом. И никто, кроме вас двоих, не сможет прочитать такое письмо!

<sup>1</sup> Мнение, будто SOS — аббревиатура английского словосочетания «Save Our Souls» («Спасите наши души») или «Save Our Ship» («Спасите наш корабль») — красивая легенда. На самом деле расшифровки нет, просто такое сочетание «точек» и «тире» — наиболее простая и отчётливая комбинация.



# Этап 1. Изучаем способ кодирования, предложенный Морзе

Прежде чем ты соберёшь кнопочный звуковой передатчик, тебе необходимо овладеть телеграфной азбукой — способом кодирования текстовой информации при помощи длинных («тире») и коротких («точка») звуковых сигналов. Это очень простой принцип передачи сообщения. Посмотри в таблицу на странице 65. Видишь, каждой букве, цифре или символу соответствует определённая последовательность «точек» и «тире». Эту последовательность и принято называть **азбукой Морзе**. Надо заметить, что современная телеграфная азбука отличается от той, которую предложил Сэмюэл Морзе в 1837 году. Сейчас международный телеграфный код включает 26 букв латинского алфавита, 10 цифр, несколько знаков препинания и служебные символы. Для передачи кириллических букв используются коды сходных латинских букв, а для «Ч», «Ш», «Э», «Ю», «Я» — придуманы свои.

При передаче сигналов с помощью азбуки Морзе необходимо соблюдать несколько правил.

1. За единицу времени принимается длительность одной точки.
2. Длительность тире равна трём точкам.
3. Пауза между элементами одного знака — одна точка.
4. Пауза между знаками в слове — три точки.
5. Пауза между словами — семь точек.

Телеграфист при помощи телеграфного ключа передаёт принимающей стороне закодированное сообщение. Существует несколько разновидностей телеграфных ключей: механический (традиционный), полуавтоматический и автоматический. Традиционный телеграфный ключ (ключ Морзе) представляет собой подпружиненное коромысло, качающееся в вертикальной плоскости, установленное на достаточно устойчивом основании. Нажимая на конец этого коромысла, оператор замыкает единственный электрический контакт, при этом точка — это короткое нажатие, тире — длинное<sup>1</sup>. Телеграфист, принявший телеграмму, расшифровывает её и передаёт адресату.

Чтобы научиться передавать закодированные сигналы, тебе потребуется некоторое время. Но ты уже знаешь — чтобы приобрести полезный навык, необходимо упорно тренироваться. Попробуй написать сло-

<sup>1</sup> Информация взята с сайта <https://ru.wikipedia.org>

во «ПРИВЕТ» с помощью азбуки Морзе на листке бумаги. А теперь представь, что ты отбиваешь это слово ключом. Получилось? Если нет, то попробуй снова.

Посмотри на модель кнопочного звукового передатчика, который тебе предстоит собрать. С правой стороны от программируемого модуля EV3 (рис. 3) расположена кнопка, она будет служить тебе ключом для передачи звукового сигнала.

При нажатии кнопки (ключа) будет издаваться звуковой сигнал, названный морзянкой. Принимающая сигнал сторона слушает его и записывает закодированное сообщение в виде «точек» и «тире». Далее сигнал необходимо расшифровать.

Попробуй собрать настоящий передатчик и отправь другу секретное послание!



**Рис. 3.** Модель звукового передатчика

Азбука Морзе с добавлением букв кириллического алфавита выглядит так:

### Азбука Морзе, таблица кодов Морзе (морзянка)

БУКВЫ		Код Морзе
Рус.	Лат.	
А	A	• —
Б	B	— • •
В	W	• — —
Г	G	— — —
Д	D	— — •
Е	E	•
Ж	V	• • • —
З	Z	— — — •
И	I	• •
Й	J	• — — —
К	K	— • —
Л	L	• — • •
М	M	— —
Н	N	— — •
О	O	— — —
П	P	• — — — •
Р	R	• — •
С	S	• • •
Т	T	—
У	U	• • —
Ф	F	• • — •
Х	H	• • • •
Ц	C	— • — — •
Ч		— — — — •
Ш		— — — — —
Щ	Q	— — — • —
Ы	Y	— • — —
Ь, Ъ	X	— • • —
Э		• • — • •
Ю		• • — —
Я		— — • —

ЦИФРЫ	Код Морзе
0	— — — — —
1	• — — — —
2	• • — — —
3	• • • — —
4	• • • • —
5	• • • • •
6	— — • • •
7	— — — • •
8	— — — — •
9	— — — — —
ЗНАК, СИМВОЛ	Код Морзе
.	• • • • •
,	• — • — • —
:	— — — — • •
;	— • — — — •
!	— — • • — —
?	• • — — — •
=	— — • • —
«»	• — • • — — •
'	• — — — — —
( )	— • — — — • —
/ №	— — • • — —
-	— — • • • —
-	• • — — — • —
@	• — — — — —
←	• • • • • •



## Этап 2. Сборка кнопочного звукового передатчика

### ШАГ 1. СБОРКА ПОДСТАВКИ ПОД ПРОГРАММИРУЕМЫЙ МОДУЛЬ



#### Детали для сборки:

- программируемый модуль EV3, 1x;
- двойная угловая балка  $3 \times 7$ , 2x;
- угловой соединительный штифт, 3×3-модульный, 2x;
- соединительный штифт, 3-модульный, синий, 2x;
- ось № 10, чёрная, 1x;
- поперечный блок, 2-модульный, серый, 2x;
- поперечная балка  $2 \times 1$ , красная, 2x;
- соединительный штифт, 2-модульный, чёрный, 10x;
- рамка  $5 \times 7$ , серая, 1x;
- балка № 7, серая, 3x.



1. Переверни программируемый модуль задней стороной вверх так, чтобы порты, обозначенные цифрами, оказались перед тобой.



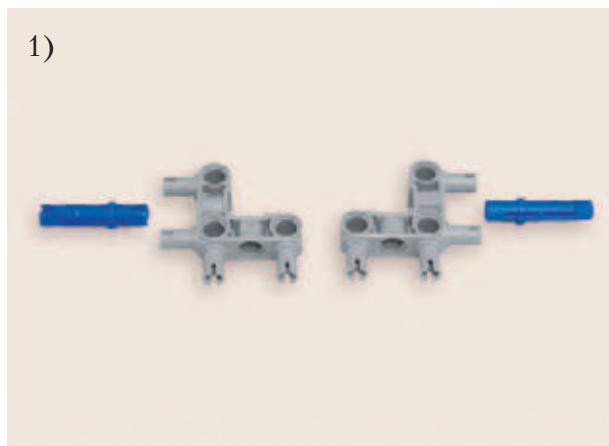
2. Вставь штифты в средние отверстия.



3. Установи балку № 7 на соединительные штифты так, чтобы все отверстия на балке совпали с отверстиями на программируемом модуле.

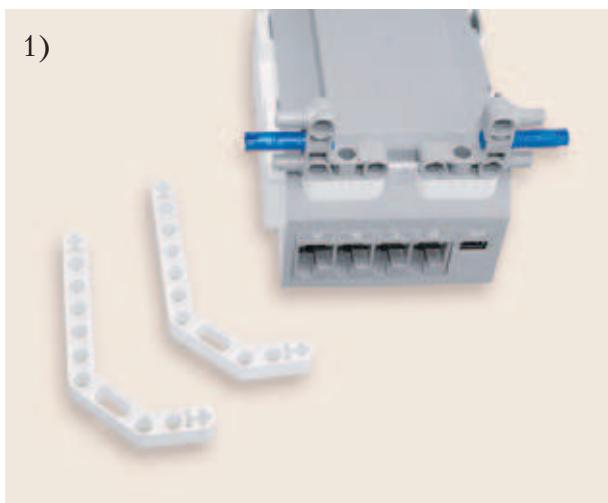


4. Вставь в боковые отверстия угловых соединительных штифтов синие 3-модульные штифты коротким концом. Затем поверни программируемый модуль портами, обозначенными буквами, к себе и вставь угловые штифты.



- Установи с обеих сторон программируемого модуля двойные угловые балки  $3 \times 7$  на синие и угловые штифты. При этом часть углового штифта должна войти в овальное отверстие, а синий штифт — в отверстие, расположенное ниже овального.

**Важно!** Балка наденется не до упора!



- Надень 2-модульные поперечные блоки на ось № 10.



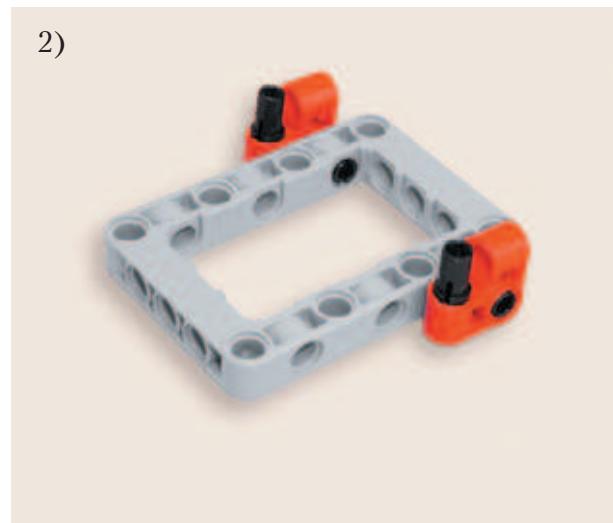
7. Закрепи получившуюся конструкцию в крестовых отверстиях угловых балок так, чтобы блоки «смотрели» на программируемый модуль.



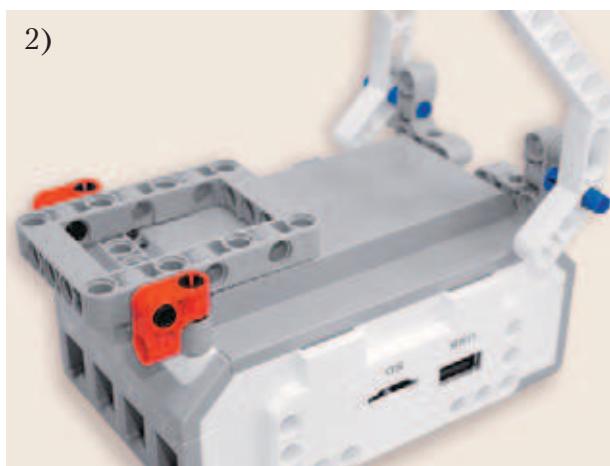
8. Вставь чёрные штифты в красную попеченную балку  $2 \times 1$ . Повтори всё это с другой балкой.  
У тебя получилось два держателя.



9. Подсоедини держатели к рамке  $5 \times 7$ .

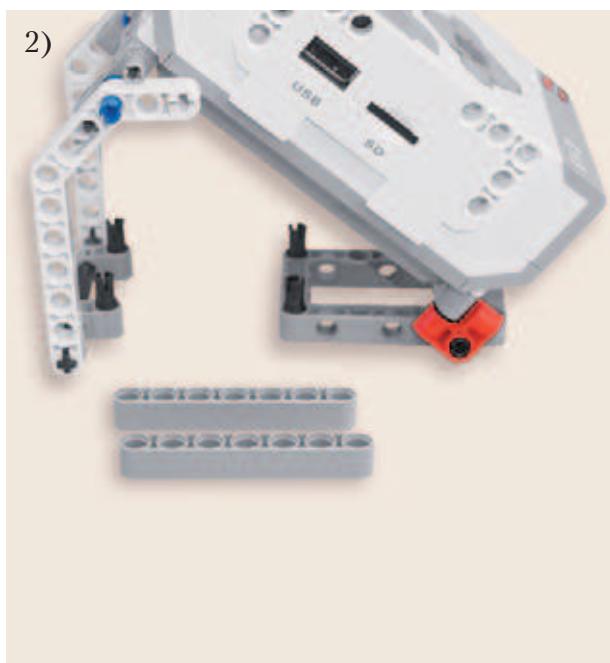


10. Установи получившуюся конструкцию на программируемом модуле. Для этого вставь свободные части штифтов в крайние отверстия на балке, которую ты добавил в пункте 3.

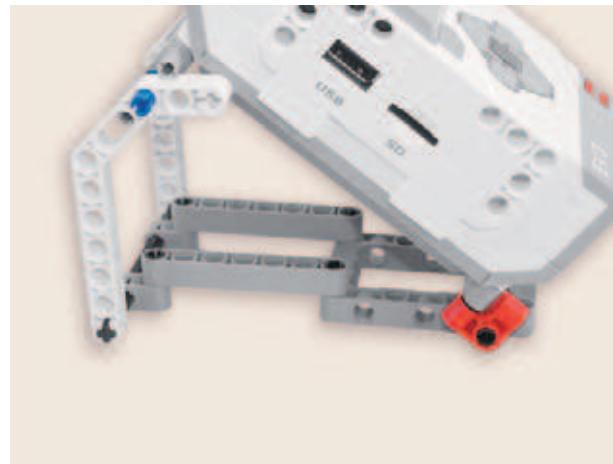


Теперь твоя конструкция может стоять на поверхности. Однако её необходимо укрепить.

11. Вставь четыре 2-модульных соединительных штифта в отверстия рамки и 2-модульных поперечных блоков.

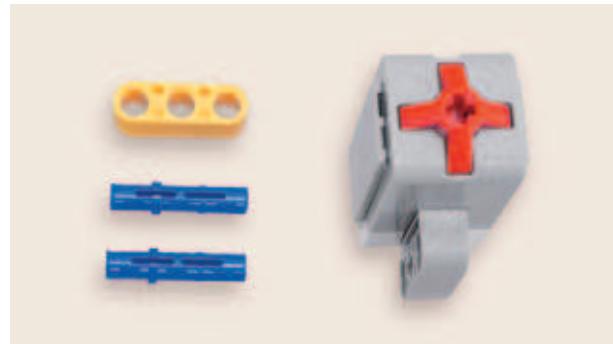
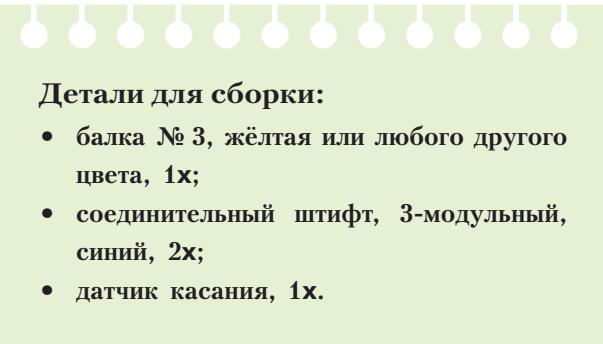


12. Надень балки № 7 крайними отверстиями на штифты.



Теперь подставка хорошо закреплена, программируемый модуль устойчиво стоит на ней.

## ШАГ 2. СБОРКА И КРЕПЛЕНИЕ КНОПКИ ТЕЛЕГРАФНОГО КЛЮЧА



1. Вставь синие штифты в средние отверстия жёлтой балки длинным концом.



2. Установи жёлтую балку на программируемом модуле сбоку. На синие штифты надень кнопку так, чтобы она «смотрела» на тебя. Кнопка телеграфного ключа готова.



### ШАГ 3. СОЕДИНЕНИЕ КНОПКИ С ПРОГРАММИРУЕМЫМ МОДУЛЕМ



#### Детали для сборки:

- подготовленный на предыдущих шагах программируемый модуль EV3, 1x;
- кабель, 25 см, 1x.

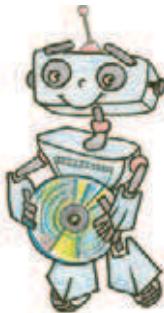
#### Внимание!

Все датчики подключаются к портам с цифрами, а все моторы — к портам с буквами.



1. Соедини кнопку с портом № 1 программируемого модуля с помощью кабеля.

Сборка кнопочного звукового передатчика завершена. Но без программы он не будет работать. Необходимо её написать.



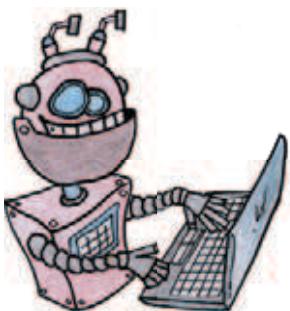
## Этап 3. Установка программного обеспечения на компьютере

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

### Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

## Этап 4. Создание программы для передатчика



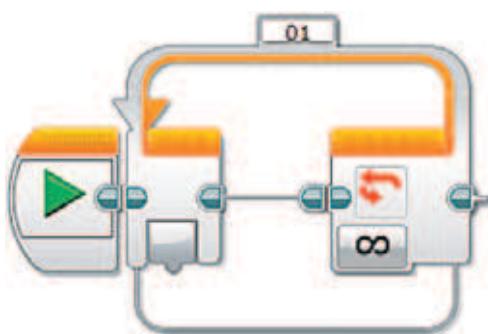
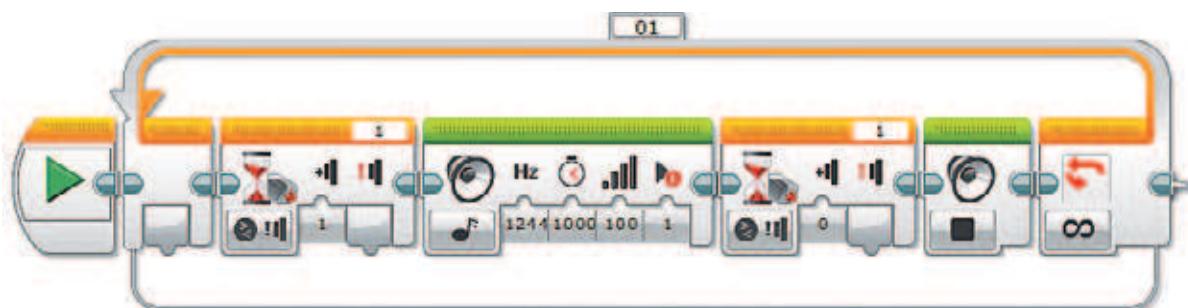
### ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

### ЛОГИКА ПРОГРАММЫ

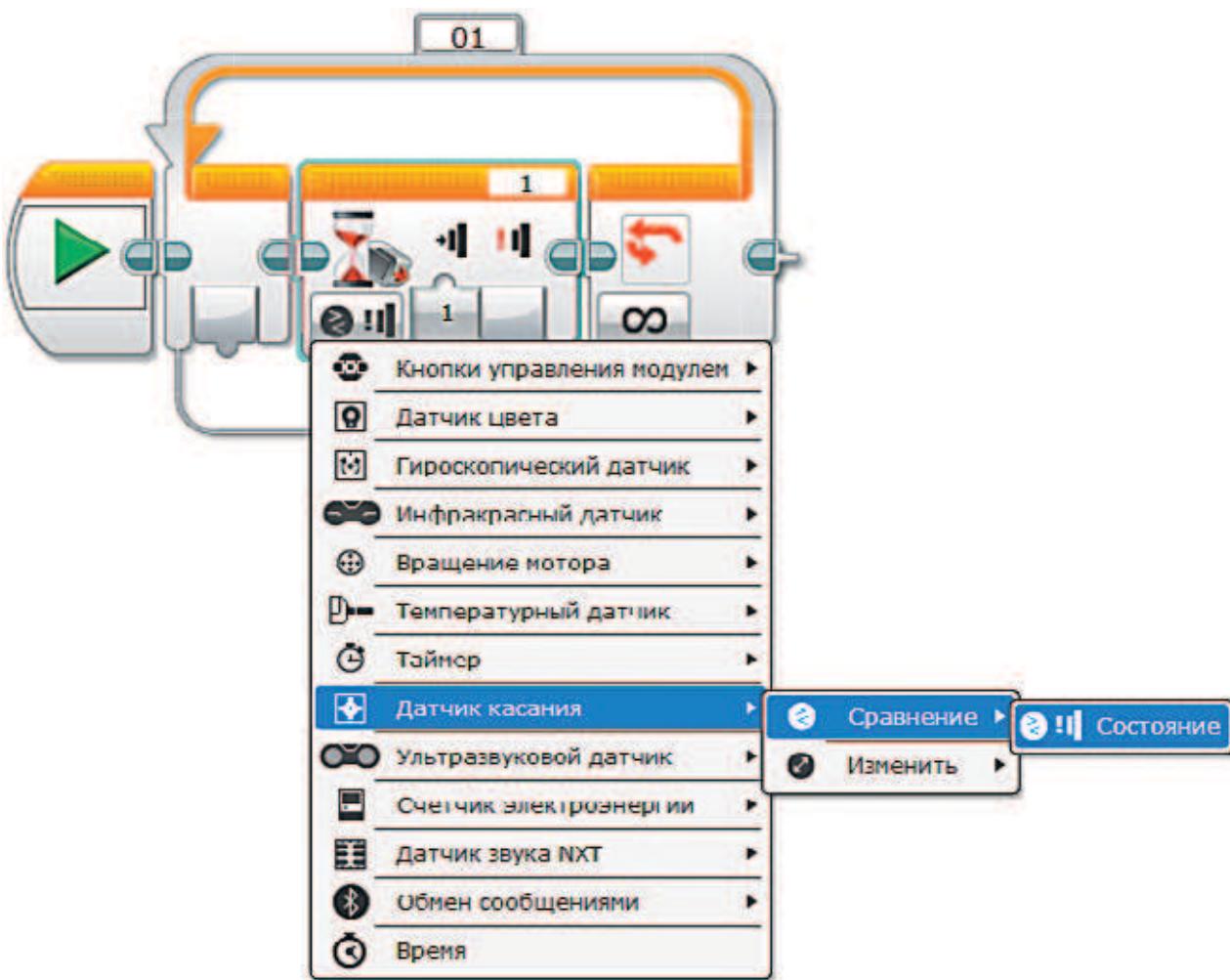
В начальный момент программа находится в режиме ожидания. При нажатии кнопки датчик начинает подавать звуковой сигнал (тон). При отпускании кнопки звуковой сигнал прекращается. Программа возвращается в начало и снова ожидает нажатия кнопки. И так по кругу. Отсюда видна цикличность нашей программы.

### ОБЩИЙ ВИД ПРОГРАММЫ

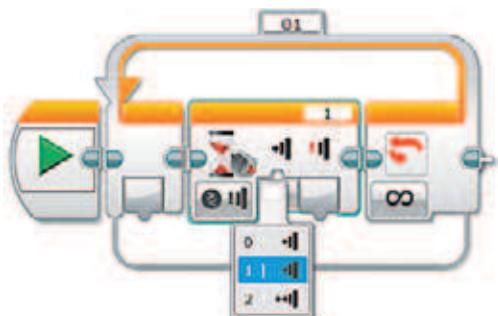
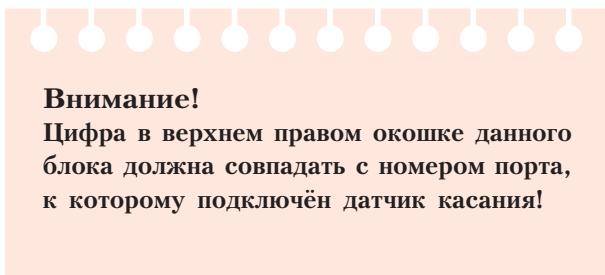


1. Для начала тебе нужно добавить цикл (оранжевый блок), так как передатчик будет посылать не один сигнал, а несколько. Все команды будут находиться внутри этого цикла. Цикл должен быть бесконечным, чтобы не было ограничения на количество подаваемых сигналов.

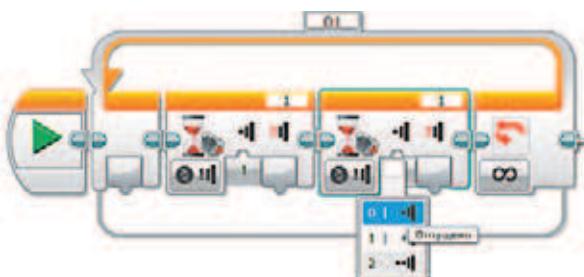
2. Помести внутрь цикла команду ожидания (оранжевый блок) и выбери опцию **Датчик касания** → **Сравнение** → **Состояние**.



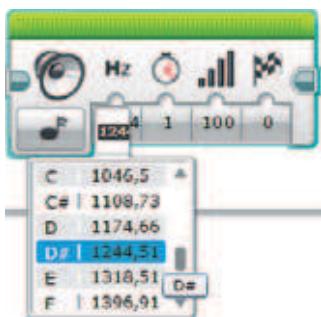
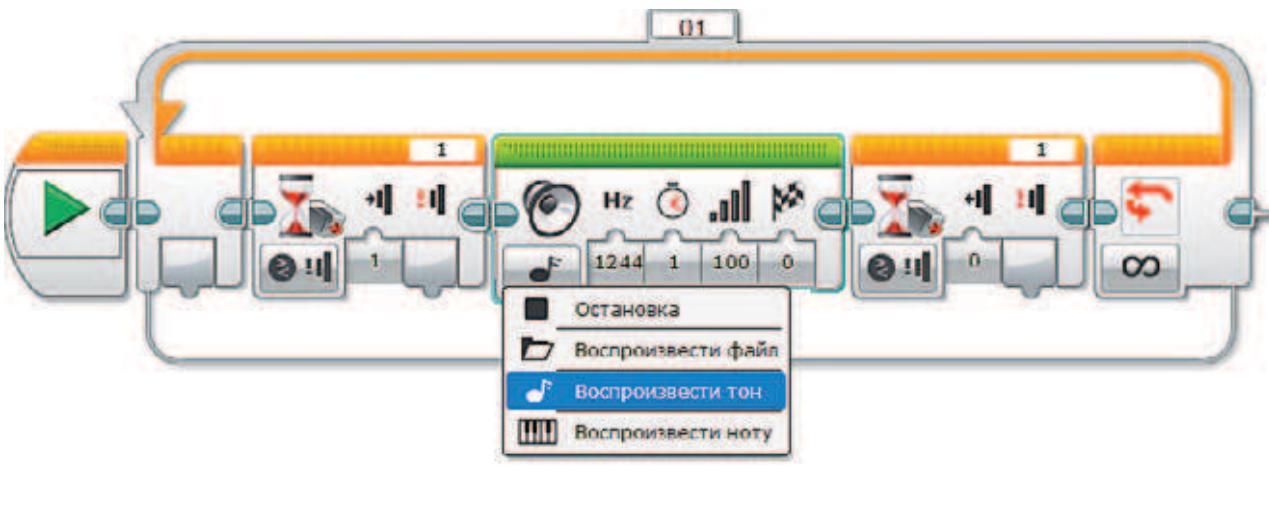
3. Выбери состояние **Нажатие** (1). В результате следующий шаг в алгоритме не будет выполнен, пока датчик касания не перейдёт в необходимое состояние **Нажатие** (1).



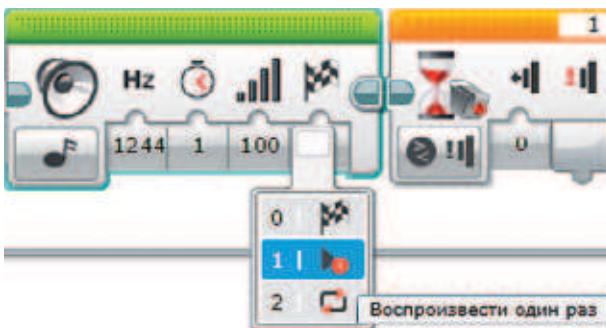
4. Добавь на следующем шаге такой же оранжевый блок, но выбери состояние **Отпущен** (0).



5. Добавь между командами **Ожидание** команду **Звук** (зелёный блок) и выбери опцию **Воспроизвести тон**.

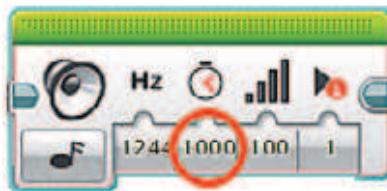


6. Выбери в этом же блоке частоту **D# 1244,51** (можно и другую, но данная частота хорошо подходит для восприятия на слух).



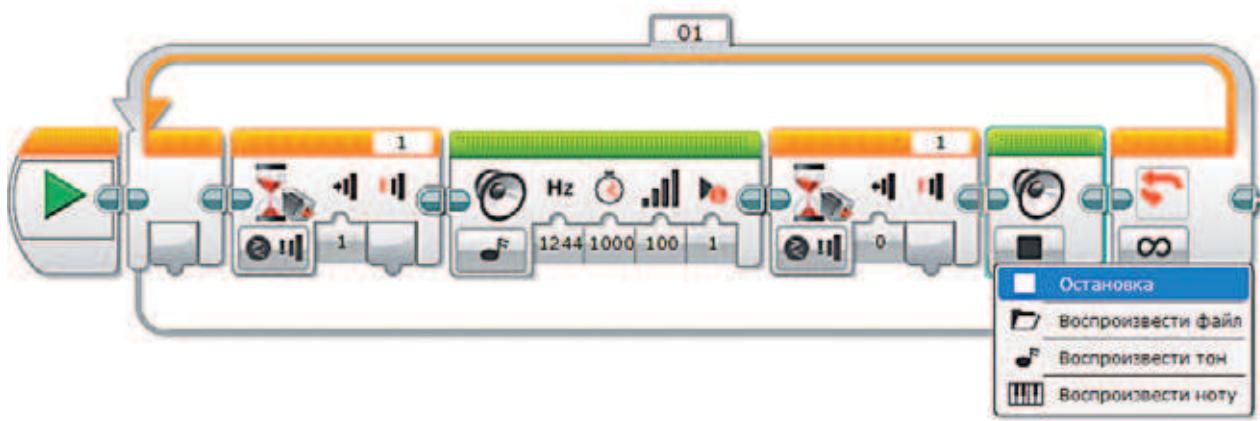
7. Выбери также тип воспроизведения **Воспроизвести один раз** (1), иначе сигнал будет подаваться в течение времени, выставленного в продолжительности, а не только при нажатой кнопке.

8. Чтобы можно было подавать сигнал любой длительности, выстави в продолжительности какое-нибудь большое число, например **1000**.



9. Осталось остановить звуковой сигнал в тот момент, когда кнопка будет отпущена. Если этого не сделать, то сигнал будет подаваться, в нашем случае, в течение 1000 секунд.

Для этого после команды **Ожидание** отжатия кнопки добавь ещё команду **Звук** и выбери опцию **Остановка**.

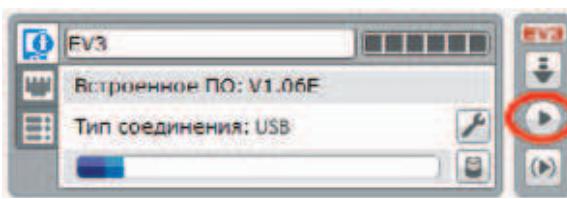
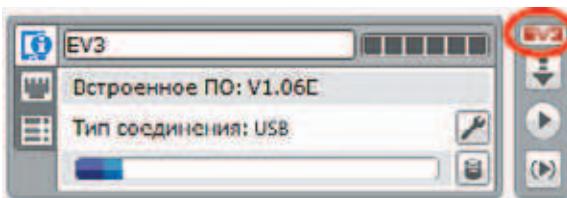


Программа готова.

# Этап 5. Загрузка программы и её тестирование



## ЗАГРУЗКА ПРОГРАММЫ



Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

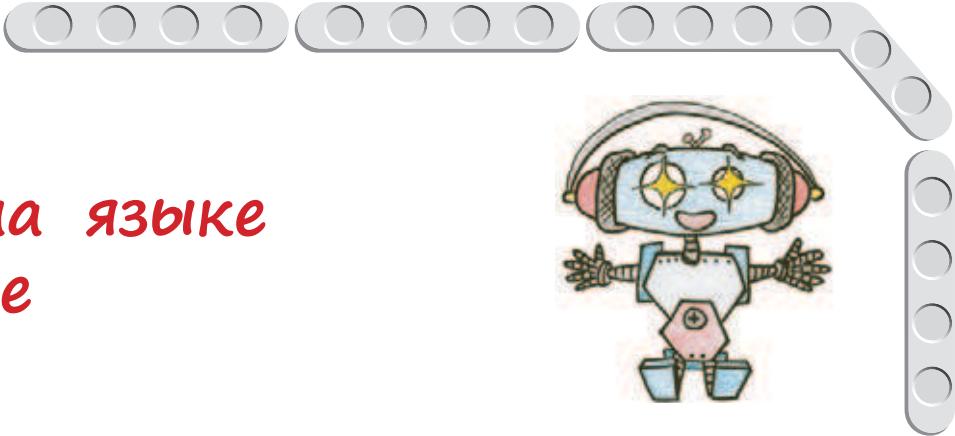
## ТЕСТИРОВАНИЕ ПРОГРАММЫ

### Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

Протестируй свой передатчик. При коротком нажатии кнопки должен подаваться короткий сигнал, при длительном — длинный.

Если передатчик функционирует неправильно, то, скорее всего, были допущены ошибки при написании программы. Перейди на этап 4 и проверь все шаги ещё раз.



## Этап 6. Поговорим на языке азбуки Морзе

Твой кнопочный звуковой передатчик готов. Теперь ты можешь отправлять любые сообщения на языке Морзе. На отправляющей стороне работают шифровальщик и телеграфист, а на принимающей — стенографист и дешифровальщик.

**Шифровальщик** — с помощью азбуки Морзе кодирует (шифрует) сообщение и пишет последовательность «точек» и «тире» на бумаге.

**Телеграфист** — с помощью передатчика отправляет закодированное сообщение, соблюдая правила Морзе (паузы между буквами и словами).

**Стенографист** — слушает поступающий сигнал и пишет последовательность «точек» и «тире» на бумаге.

**Дешифровальщик** — с помощью азбуки Морзе декодирует (расшифровывает) сообщение.

### НАЧИНАЙ ИГРАТЬ

Разделитесь с друзьями на две группы — отправляющих сообщение и принимающих его.

Представь, что первая группа находится на корабле, который попал в сильную бурю и терпит крушение. Необходимо отправить сигнал бедствия «SOS» и дождаться ответа «RCD» (от англ. Received — принято).

Вторая группа — спасатели. Они должны принять сигнал бедствия «SOS» и отправить в ответ сигнал «RCD».

Начинайте игру по следующему сценарию.

1. В минуту опасности, пользуясь международной азбукой Морзе и правилами кодирования Морзе, шифровальщик на корабле кодирует следующее сообщение:  
«SOS SOS SOS».
2. Телеграфист отправляет получившийся код с помощью передатчика.
3. Стенографист спасателей слушает сигналы и составляет код из «точек» и «тире».

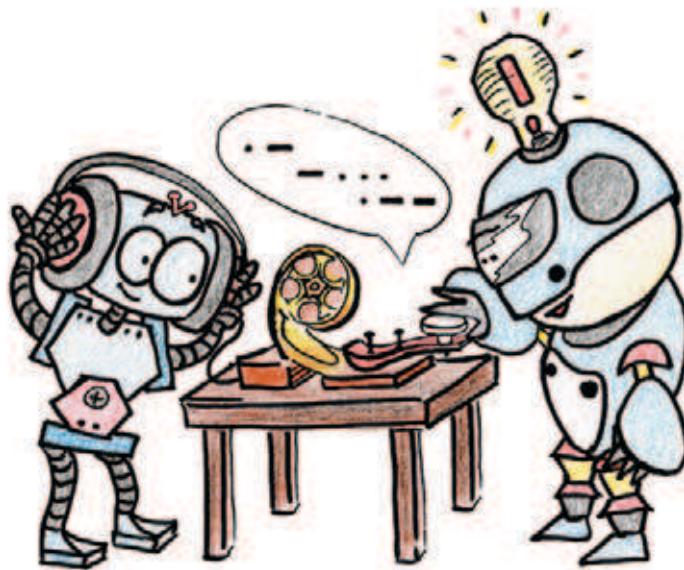
4. Дешифровальщик расшифровывает полученное сообщение.
5. Шифровальщик спасателей кодирует сообщение: «RCD», а телеграфист передаёт его.
6. Стенографист на корабле слушает сигналы и записывает сообщение — код из «точек» и «тире».
7. Дешифровальщик расшифровывает полученное сообщение.

Подумай, как ещё можно обмениваться информацией с помощью кода Морзе.

Поменяйтесь ролями отправляющей и принимающей стороны. Повторите все действия, только с кириллической азбукой Морзе. В данном случае сообщения могут быть: «СПАСИТЕ» и «ПРИНЯТО».

Можно придумать и другую игру! Например, одна из команд должна закодировать сообщение и отправить его остальным. Принимающие команды за установленное время декодируют сообщение. Команда, которой удалось первой верно произнести декодированное сообщение, получает право отправить своё сообщение остальным. И так далее.

Теперь ты знаешь, как передавать информацию с помощью азбуки Морзе, но это ещё не всё... Попробуй составить собственную азбуку кодов, которую будут знать только ты и твоя принимающая сторона (при этом можно кодировать не только буквы, но и целые слова и фразы). Теперь вы сможете общаться тайно — никто не будет знать ваших секретов!





## Этап 7. Отправляем телеграмму



Ты познакомился со способом кодирования информации и собрал передатчик, с помощью которого отправлял сообщения. Словом, ты окунулся в недавнее прошлое и стал заправским телеграфистом.

Но согласись, что принимать и расшифровывать морзянку непросто и даже утомительно. Нужно напряжённо вслушиваться, чтобы не ошибиться при расшифровке. Чуть отвлёкся – и прослушал.

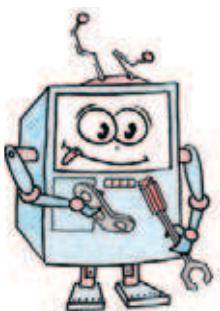
Но морзянку можно сделать видимой! И тогда стенографист будет не нужен. Сообщения будут отображаться в виде «точек» и «тире» на экране твоего программируемого блока EV3. Теперь и телеграфисту будет легче, и никто не услышит ваши сигналы. Мы заменим звук, который воспроизводился во время нажатия кнопки, на свет индикаторных лампочек на панели блока EV3. Кроме того, ты сможешь запрограммировать автоматический перенос строк на экране модуля EV3!

Это будет ненастоящая телеграмма, как на почте, потому что на нашем небольшом экранчике EV3 будут появляться не слова, а «точки» и «тире», отделённые друг от друга пробелом (этот процесс мы тоже автоматизируем).

Экран EV3, действительно, небольшой, поэтому можно передавать не слишком длинные сообщения.

Для этих целей тебе понадобится кнопочный текстовый шифратор. Попробуй собери его!

## Этап 8. Кнопочный текстовый шифратор



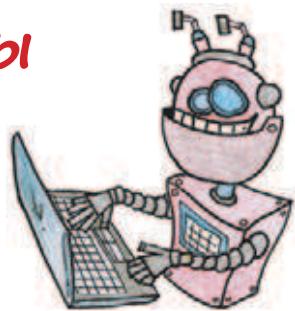
Этот шифратор будет отличаться от собранного ранее только программной составляющей. Никаких изменений в конструкцию вносить не будем (рис. 4).



**Рис. 4.** Модель кнопочного текстового шифратора



# Этап 9. Создание программы для шифратора



## ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

## ЛОГИКА ПРОГРАММЫ

Сначала необходимо очистить экран EV3 и объявить переменные  $x$  и  $y$ , которые будут определять место на экране, где печатается следующий символ. Это координаты:  $x$  будет задавать положение по горизонтали,  $y$  — по вертикали. Здесь необходимо вспомнить, что начало, точка  $(0; 0)$ , находится в левом верхнем углу экрана программируемого модуля EV3.

Далее надо составить похожий фрагмент программы, как в кнопочном звуковом передатчике, только вместо сигнала будем включать и выключать индикаторные лампочки блока EV3.

Также ты сможешь замерить время, в течение которого была нажата кнопка. Если это время будет коротким, значит, на экран выводится «точка», если длительным — «тире».

Если место на экране закончится, придётся сделать перенос строки, работая с координатами  $x$  и  $y$ . Также понадобятся они и при выводе очередного символа.

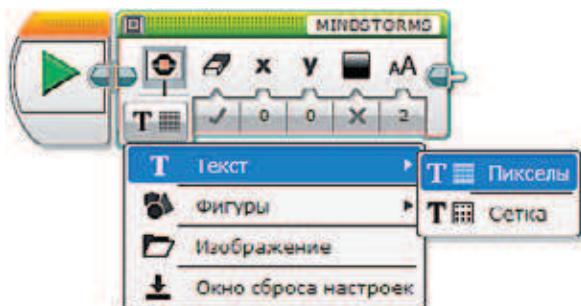
И последнее: необходимо автоматизировать постановку пробела между буквами при паузе. Для этого будем замерять ещё одно время — это время, за которое кнопка НЕ нажималась. Если оно длинное, ставим пробел (то есть, делаем отступ, снова работая с координатами  $x$  и  $y$ ).

Таким образом, тебе понадобятся два условия, при которых ты будешь проверять длительность времени: когда кнопка была нажата и когда она долго не нажималась.

Так как наша программа циклична, после вывода каждого символа необходимо возвращаться в начало.

## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ ШИФРАТОРА

В открывшемся окне проекта начинай составлять программу для шифратора.



1. Начни с очистки экрана. Если этого не сделать, то на экране будет отображаться стандартный логотип EV3 и название исполняемой программы. Для этого добавь команду **Экран** (зелёный блок) и выбери опцию **Текст → Пиксели**.
2. Заметь, в этом блоке по умолчанию стоит очистка экрана. Это говорит о том, что экран будет очищен перед выводом текста, который записан в правом верхнем окошке. Другими словами, при выполнении этой команды на экране появится надпись «MINDSTORMS». Чтобы на экран ничего не выводилось, поменяй значение в этом окошке на **Проводной**. Это будет означать, что текст формируется отдельно и отправляется на экран, но формироваться ничего, конечно же, не будет.
3. Теперь объяви две переменные с именами *x* и *y*. Для этого добавь две команды **Переменная** (красный блок). Затем кликни в правое верхнее окошко и выбери опцию **Добавить переменную**.

Тебе будет предложено ввести имя новой переменной. В появившемся окне введи **x**.

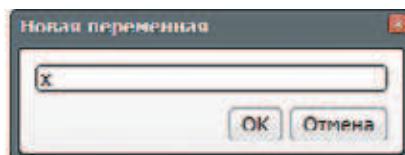
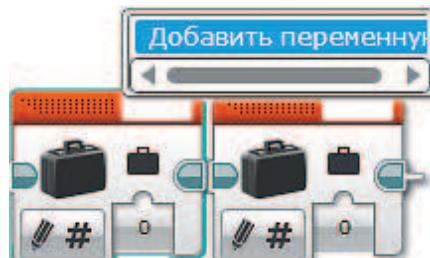
### Внимание!

Регистр и язык имеют значение! Вводи латинскую строчную (маленькую) букву **x**.

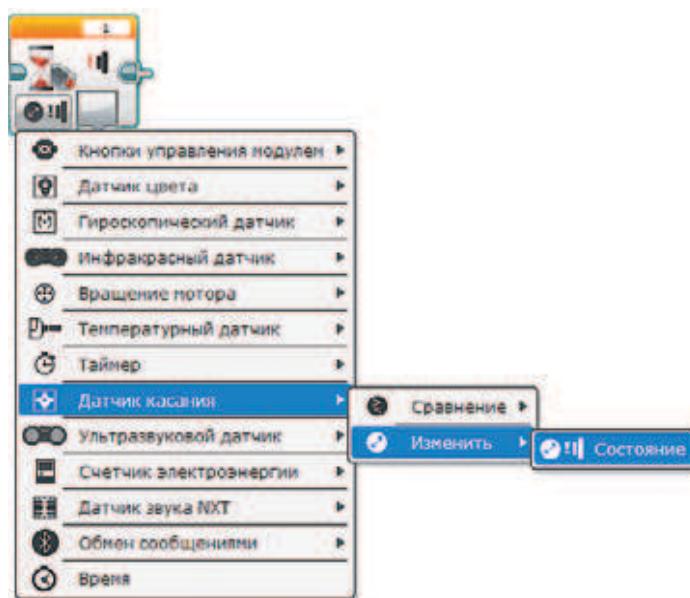
То же самое нужно сделать со второй командой, только ввести имя переменной **y**.

### Внимание!

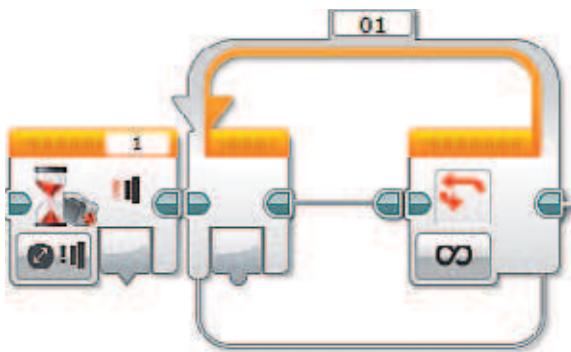
По умолчанию здесь выбрана опция Запись → Числовое значение и введено значение 0. Это означает, что в переменные записаны нули.



4. Последней командой перед большим циклом станет **Ожидание** нажатия кнопки, то есть основная часть программы (которая внутри цикла) не начнёт свою работу, пока не будет впервые нажата кнопка. В данном случае не будет важно, в какое состояние перешла кнопка. Важно лишь то, что её состояние изменилось, поэтому в опциях выбери **Датчик касания** → **Изменить** → **Состояние**.



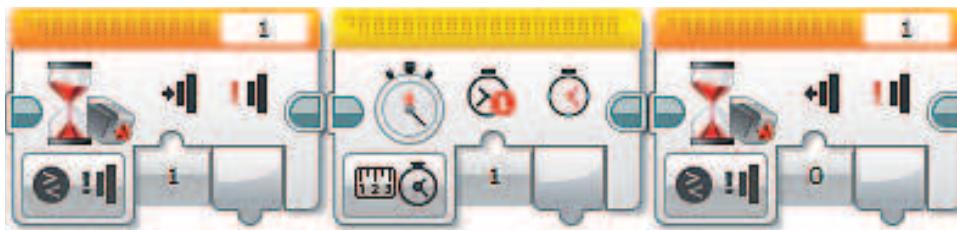
5. Затем добавь бесконечный цикл. Все остальные команды будут внутри него.



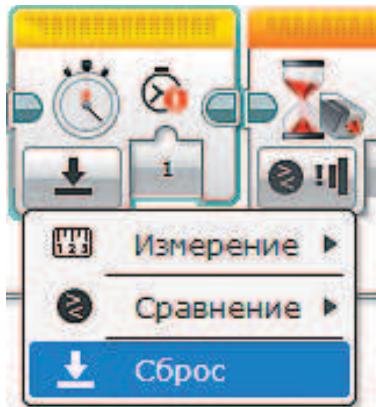
6. Первые две команды, как и в кнопочном звуковом передатчике, — **Ожидание** нажатия и отжатия кнопки.



7. Чтобы замерить время, в течение которого была нажата кнопка, добавь команду **Таймер** (жёлтый блок) после нажатия кнопки.



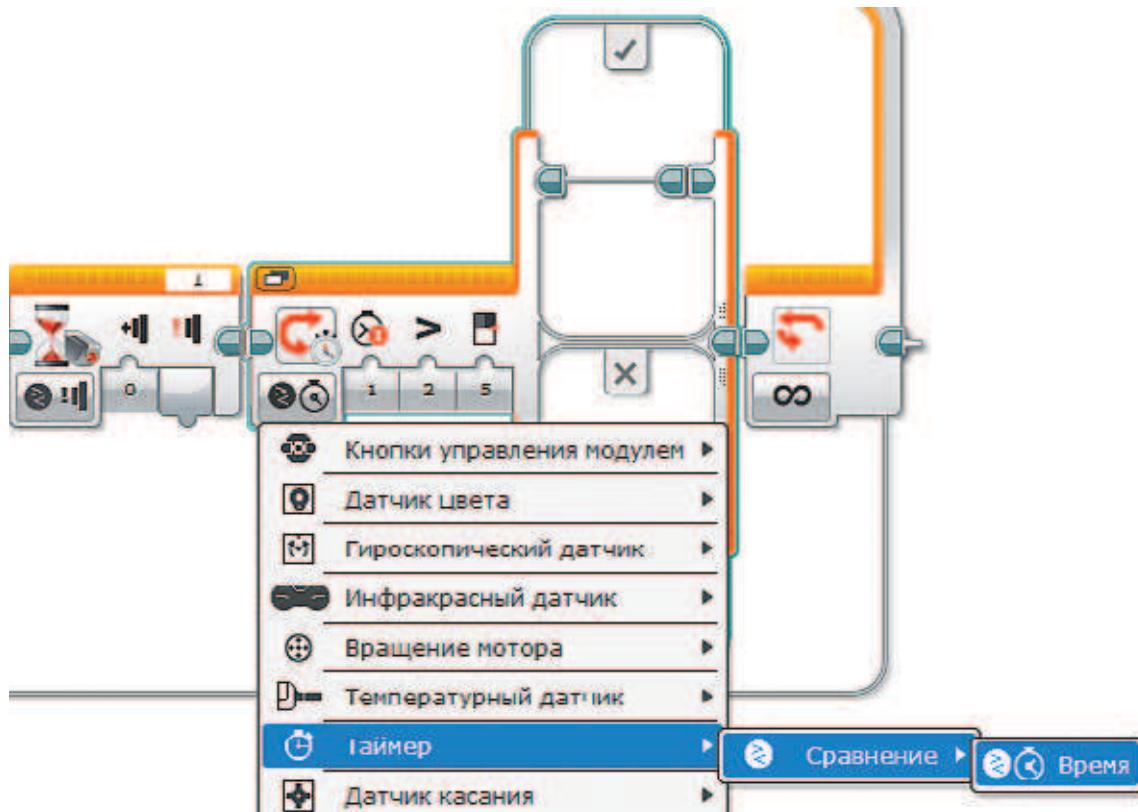
8. Выбери в команде **Таймер** опцию **Сброс**. В результате таймер обнулится и начнёт отсчёт времени с нуля.



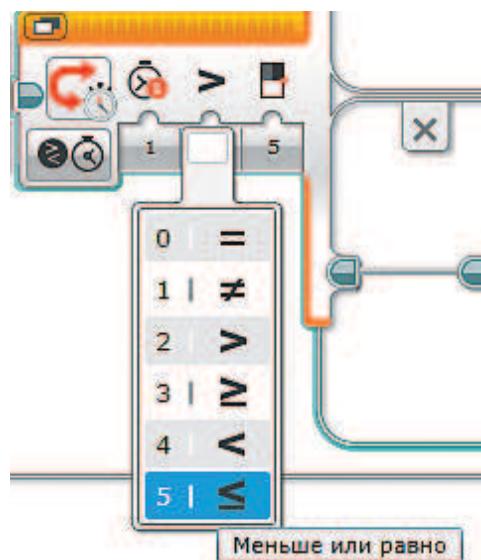
#### Внимание!

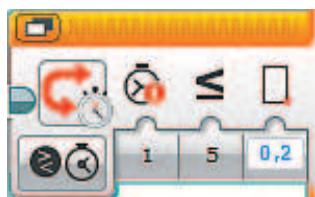
Цифра в окошке обозначает идентификатор таймера и говорит о том, что мы пользуемся таймером № 1. Всего может быть задействовано восемь таймеров.

9. Теперь следует проверить, сколько времени была нажата кнопка. Если она была нажата меньше или ровно 0,2 секунды, то программа выведет на экран точку, если больше — тире. Добавь команду **Переключатель** (или условие) и выбери опцию **Таймер → Сравнение → Время**.



10. Настрой условие. По умолчанию в данный момент считается показание таймера № 1. Далее следует выбрать другой тип сравнения — **Меньше или равно**.





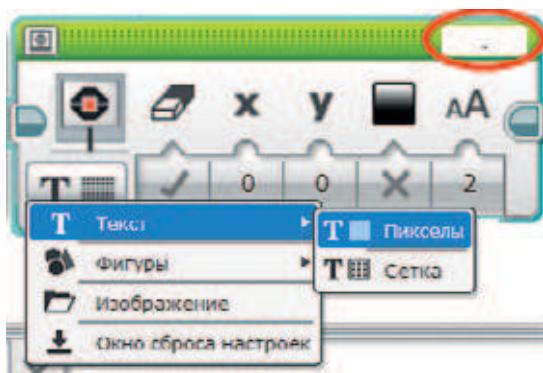
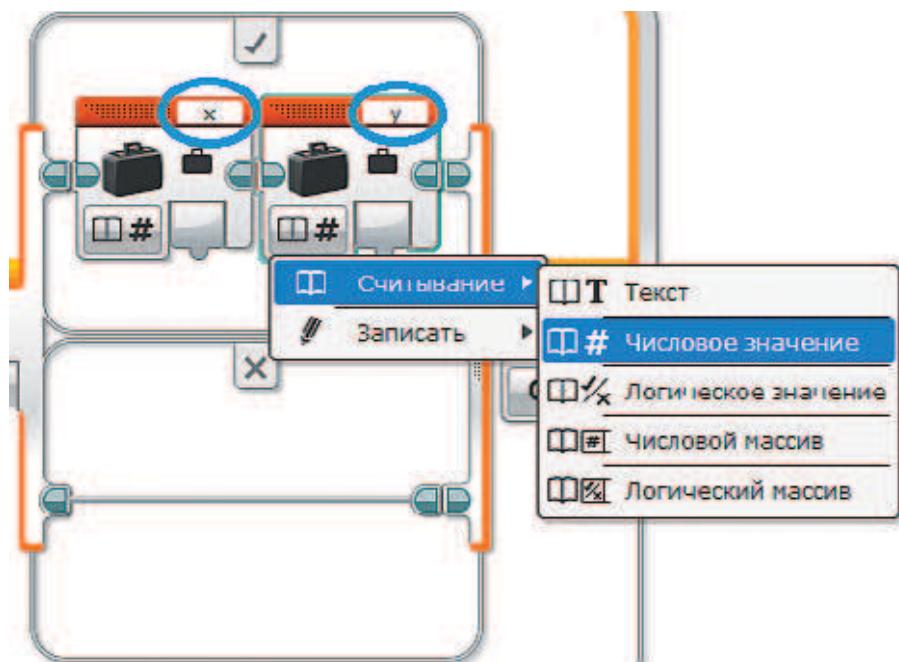
11. Задай пороговое значение **0,2**, так как проверяется условие: «Если кнопка была нажата меньше или ровно 0,2 секунды».

12. В верхней строчке пропиши события, которые наступают, если условие выполнено, а в нижней — если не выполнено.

Если кнопка действительно была нажата меньше или ровно 0,2 секунды, то выведи точку. Приступай.

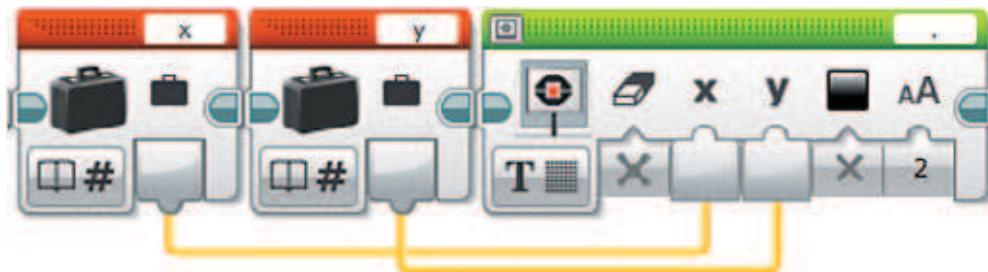
Для начала добавь две переменные и выбери опцию **Считывание** → → **Числовое значение**. С их помощью будет понятно, в какое место на экране вывести следующий символ.

**Обрати внимание** на имена считываемых переменных!



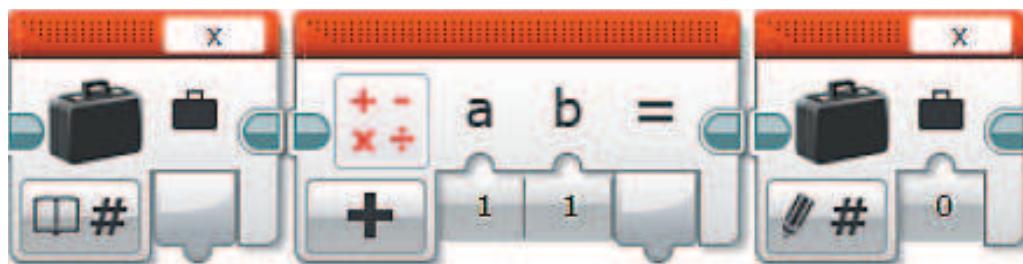
13. Теперь выведи точку с помощью команды **Экран**, в ней выбери опцию **Текст** → **Пиксели**, а вместо текста «MINDSTORMS» введи точку с клавиатуры.

14. Положение «точки» задаётся координатами **x** и **y**, которые считываем из одноимённых переменных. Для этого соедини выходные порты переменных с соответствующими входными портами команды **Экран**. Также необходимо присвоить параметру **Очистка экрана** значение **Ложно**, иначе при выводе очередного символа предыдущий будет удаляться.

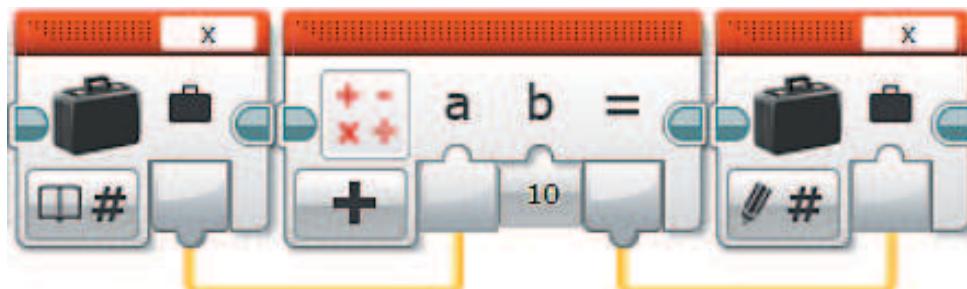


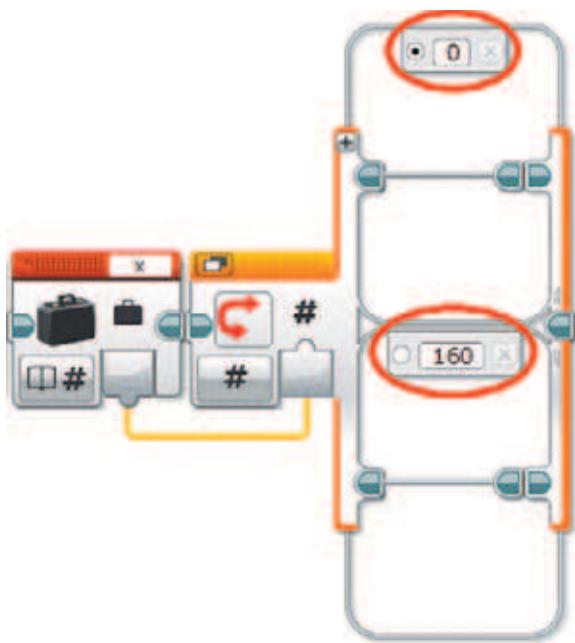
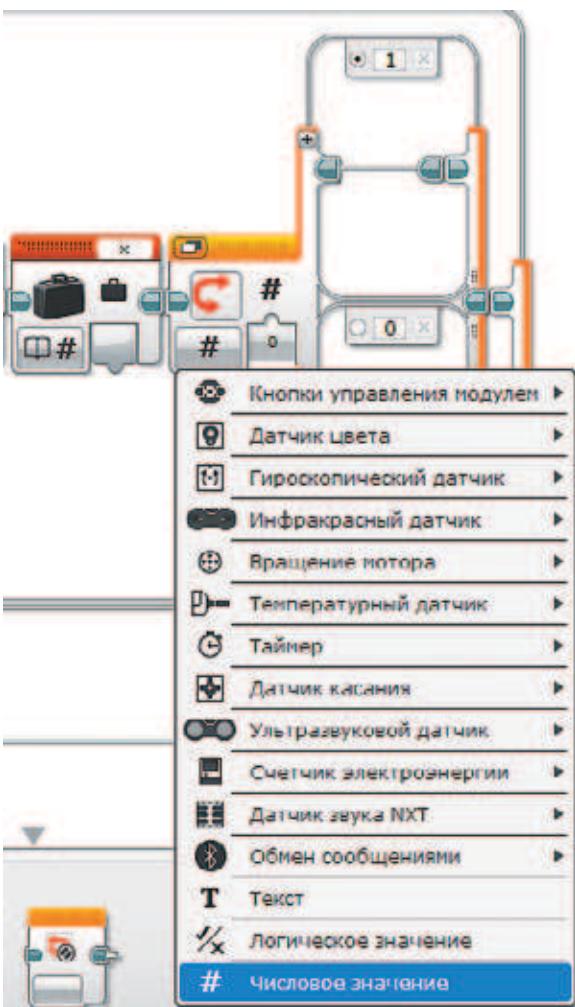
15. Далее задай отступ между знаками. Для этого увеличь значение переменной *x*. Ты сам можешь рассчитать размер отступа, однако оптимальным будет 10 пикселов.

Это можно сделать, считывая значения переменной *x* и добавляя к ней числа 10 с помощью команды **Математика** (красный блок). Добавь команду **Переменная**, выбери опцию **Считать** → **Числовое значение**, затем добавь команду **Математика**. По умолчанию выбрана опция **Добавить** и команда записи переменной.



16. Соедини выходной порт переменной с входным портом **a**. В порт **b** впиши число **10**. Выходной порт результата соедини с входным портом записи числового значения **x**.





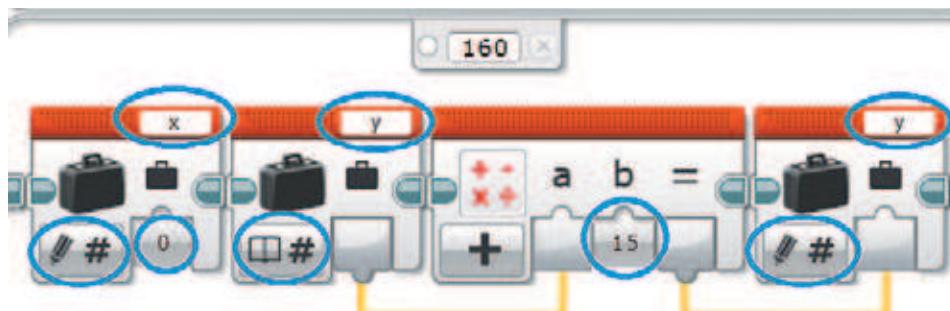
17. Теперь проверь, не достиг ли ты конца строки на экране. Для этого считай значение переменной *x* и проверь условие «Равно ли это значение 160?» (именно на 160-м пикселе и заканчивается строка экрана).

Добавь команду считывания переменной *x* и условие. В условии выбери опцию **Числовое значение**.

18. Соедини выходной порт переменной с входным портом условия. Выстави значение по умолчанию **0**, а альтернативным будет значение **160**.

Нас интересует только случай, когда мы достигли 160-го пикселя.

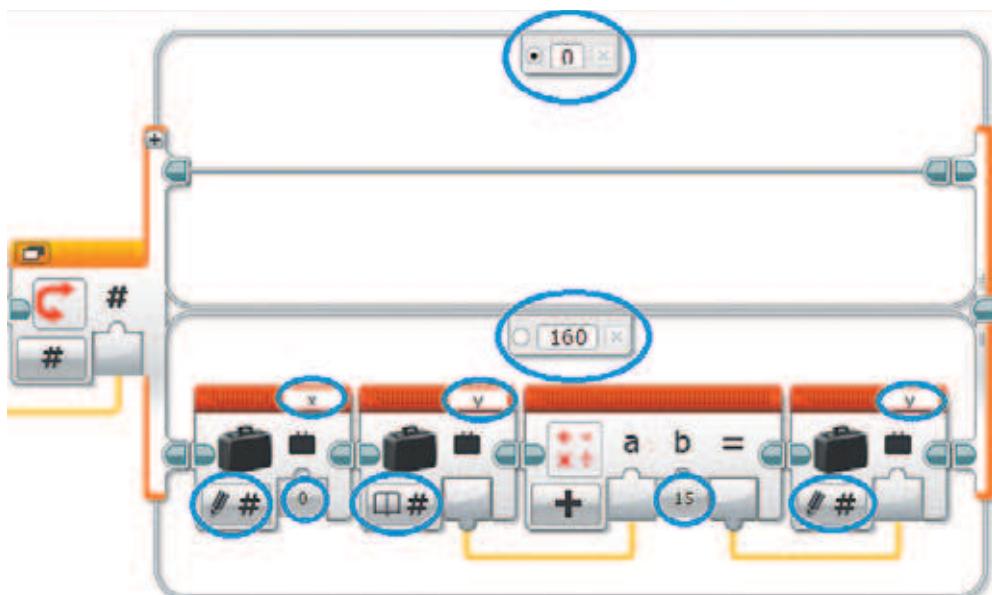
19. Обнули значение  $x$  (то есть вернись в начало строки) и увеличь значение  $y$  на 15 (то есть спустись ниже на строку).  
У тебя получится такая последовательность:



20. Теперь вернись к первому условию, где проверялось превышение 0,2 секунды, и пропиши всё то же самое в случае невыполнения условия. Разница будет лишь в том, что на экран будет выводиться не «точка», а «тире».  
Должно получиться следующее:

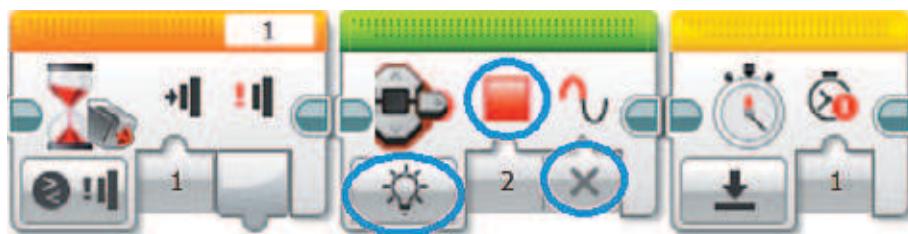


Продолжение фрагмента программы смотри на следующей странице. Для большей ясности последний блок повторён.

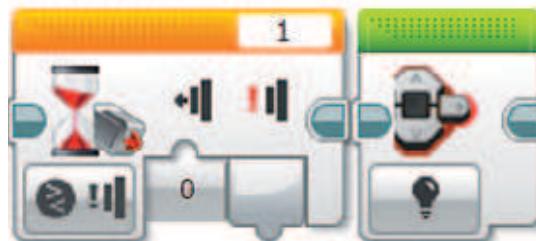


21. На следующем шаге добавь простую вещь — включение светового индикатора EV3 при нажатии кнопки. Это облегчит работу телеграфиста.

В начале программы после команды **Ожидание** нажатия кнопки добавь команду **Индикатор состояния модуля** (зелёный блок). Нужно, чтобы индикация горела постоянно и была красной. Для этого выбери опцию **Включить**, в настройках выставь красный цвет, а в пункте **Импульсный** (то есть мигающий) выстави **Ложно**.



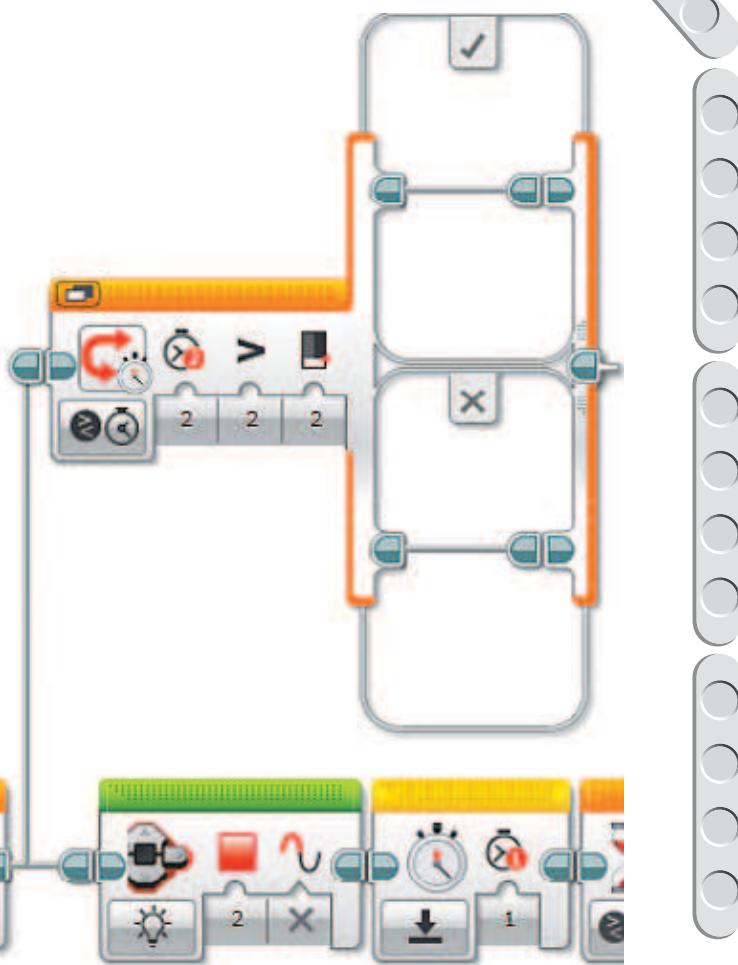
22. После того как кнопка отпущена, индикация должна выключаться. Для этого после отжатия кнопки добавь ещё одну такую команду и выбери опцию **Выключить**.



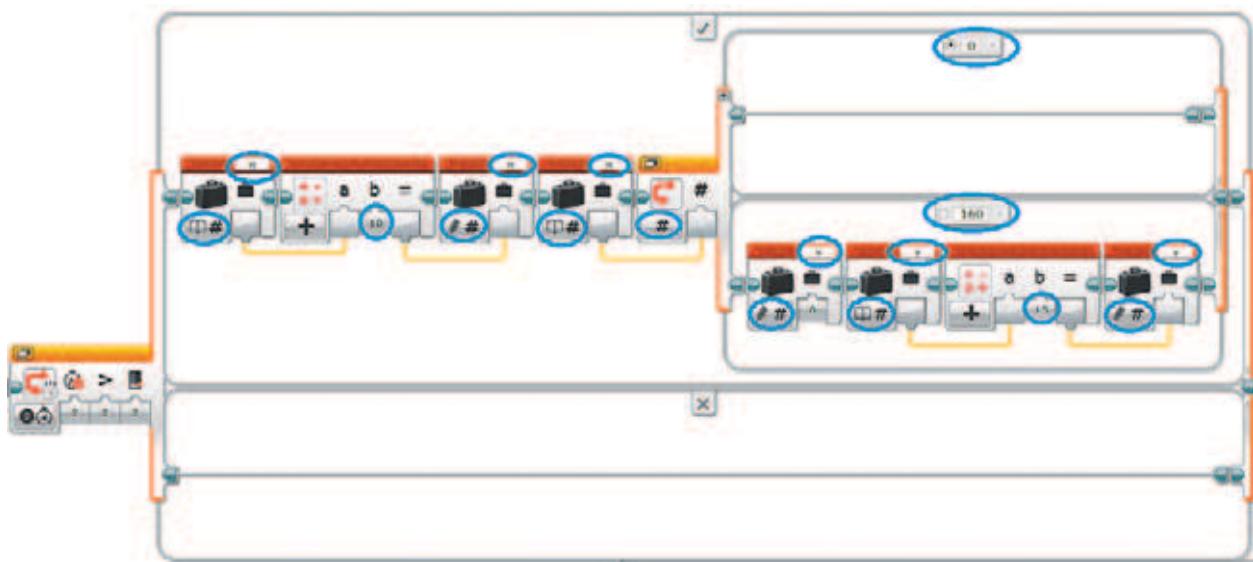
23. Осталось автоматизировать процесс постановки пробела. Добавь в **самом начале цикла** ещё один таймер, выбери опцию **Сброс** и присвой ему номер **идентификатора 2**. Он будет начинать работать в самом начале. Считывать его значение нужно каждый раз после нажатия кнопки. Если прошло много времени (допустим, 2 секунды), то программа будет ставить пробел.



24. После нажатия кнопки сделай второе ответвление и помести в него условие, проверяющее, сколько прошло времени. Если это время превышает 2 секунды, то программа будет ставить пробел, то есть увеличивать переменную  $x$  на **10**. Здесь же надо проверить, достиг ли ты края строки и нужно ли сделать перенос. Если прошло меньше или ровно 2 секунды, то ничего делать не нужно.



Подобную конструкцию ты уже делал на предыдущих шагах. В результате внутри условия должно получиться следующее:

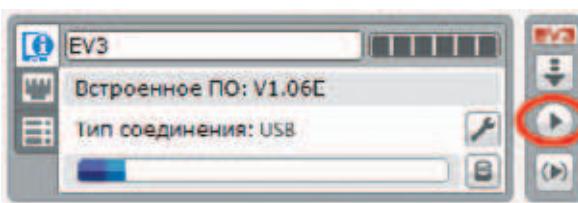
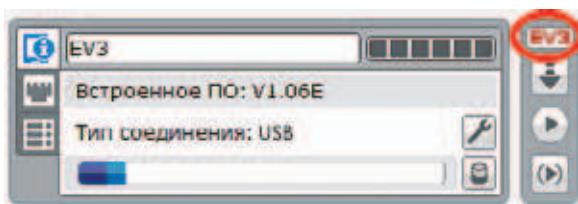


## Этап 10.

# Загрузка программы и её тестирование



## ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ



Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.

## ТЕСТИРОВАНИЕ

### Внимание!

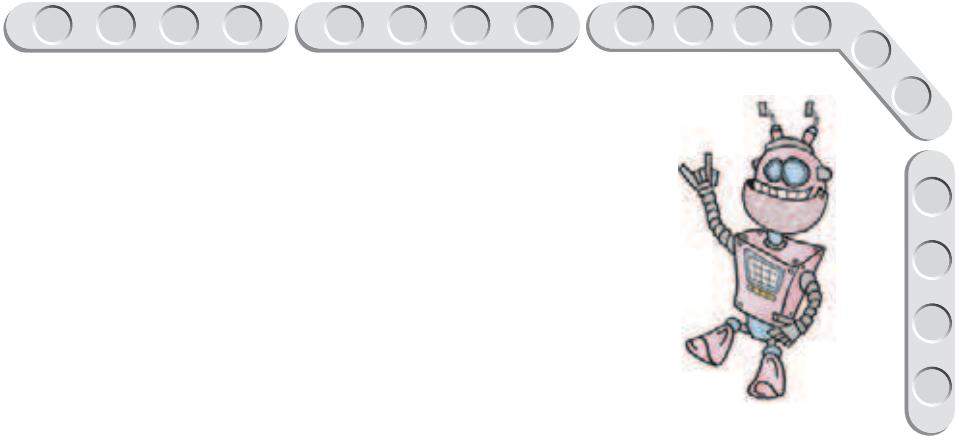
Наша программа – это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле.

Протестируй свой передатчик.

1. При коротком нажатии кнопки на экране должна появляться «точка», при длительном нажатии – «тире».
2. При каждом нажатии кнопки панель EV3 должна загораться красным.

3. Когда строка на экране заканчивается, должна начинаться новая.
4. Если кнопка не нажимается более 2 секунд, то при следующем нажатии должен появляться пробел.

Если передатчик функционирует неправильно, то, скорее всего, были допущены ошибки при написании программы. Проверь все шаги ещё раз.



## А теперь...



Позови своих друзей, чтобы вас было не меньше шести человек. Разделитесь на две группы по три человека.

В каждой группе определите следующие роли.

1. **Шифровальщик** — с помощью азбуки Морзе кодирует (шифрует) сообщение и составляет последовательность «точек» и «тире» на бумаге.
2. **Телеграфист** — с помощью текстового шифратора составляет телеграмму, соблюдая правила Морзе (паузы между буквами и словами).
3. **Дешифровальщик** — с помощью азбуки Морзе декодирует (расшифровывает) сообщение.

Составьте сценарий вашей переписки, например такой.

Представьте, что первая группа живёт на острове и очень любит читать. А вторая группа — библиотекари, которые живут на континенте. Островитянам недавно установили телеграф, и теперь они могут с его помощью заказывать книги. Но для этого они должны отправить заказ с названием книги.

Пока островитяне набирают своё сообщение, библиотекари пишут свой график работы в один из дней, например «Вторник 12–16».

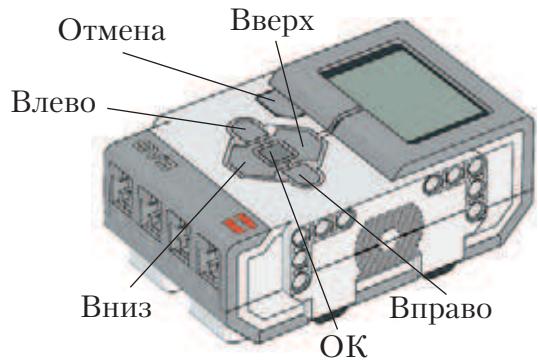
При получении островитянами расписания они должны отправить сообщение, в котором укажут время, когда они приплывут забирать заказ.

Когда библиотекари получают заказ, они должны ответить, имеется ли в наличии такая книга. Если да, то сколько экземпляров книги они готовы предоставить.

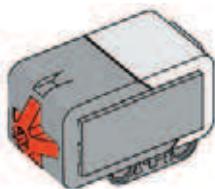
***Подумай и предложи свой сценарий для игры. Фантазируй — и у тебя всё получится!***

## Перечень деталей

Модуль EV3, 1x



Датчик касания, 1x



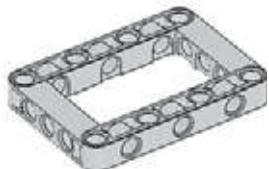
## Поперечный блок

Поперечный блок,  
2-модульный, серый, 2x



## Рамка

Рамка 5×7, серая, 1x



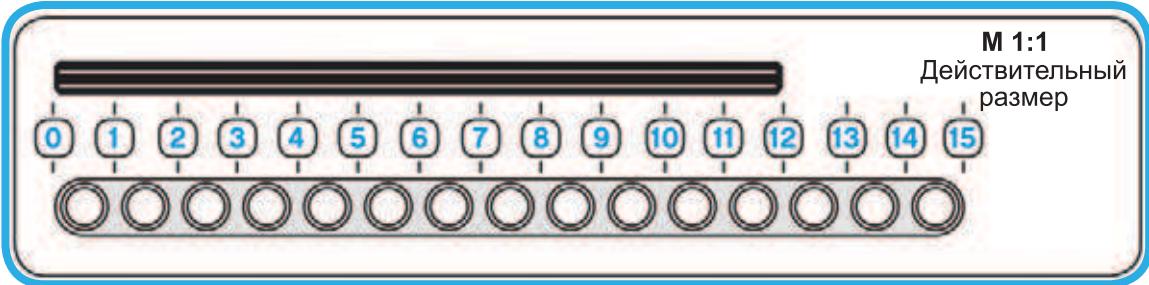
## Кабель

Кабель, 25 см, 1x



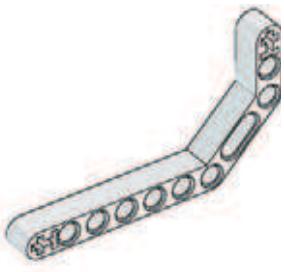
## Ось

Ось № 10, чёрная, 1x



## Балки

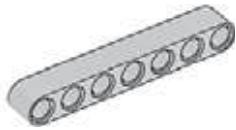
Двойная угловая балка  $3 \times 7$ ,  
белая, 2x



Поперечная балка  $2 \times 1$ ,  
красная, 2x



Балка № 7, серая, 3x



Балка № 3, жёлтая, 1x



## Штифты

Соединительный штифт,  
2-модульный, чёрный, 10x



Угловой соединительный  
штифт, 3×3-модульный,  
серый, 2x



Соединительный штифт,  
3-модульный, синий, 4x





Робочист спешит  
на помощь!



# История гигиены помещений



Ещё с давних времён чистоте и соблюдению гигиены придавалось огромное значение. Знаешь ли ты, что в Древней Греции Гигиена (олицетворение гигиены) была настоящей богиней, почитаемой наравне с Аресом, Афиной, Герой и другими хорошо тебе известными по мифам обитателями Олимпа? Имя богини произошло от греческого слова «сила». Таким образом, гигиена — это сила. Храмы Гигиене возводились по всей Греции с VII века до н. э., а статуям не было числа (рис. 1). Греки также использовали мыло, которое уже было в ходу в Вавилоне и Египте, а римляне даже применяли губки, пемзу и специальные скребки (стригил). Кроме того, в Греции и Италии и колониях к концу I века до н. э. во многих городах был проведён водопровод. Однако, несмотря на любовь к водным процедурам, в Древнем мире не уделяли должного внимания уборке помещений, вследствие чего личная гигиена оказывалась фактически бесполезной: всё сразу же пачкалось в накопившейся на стенах копоти и саже.



Рис. 1. Богиня Гигиена

Средневековье, пришедшее вместе со свержением и уничтожением «языческого» наследия, отбросило технический прогресс назад. Так, были разрушены термы (общественные бани), в которых римские граждане привыкли мыться, поскольку теперь они считались аморальными и порочными. Большее внимание стало уделяться вере и чистоте души, в то время как проблемы с состоянием тела и здоровья стали расцениваться как ниспосыляемые Господом испытания. Медицина также стала подпольной. Поскольку фактически не осталось мастеров, знавших тонкости в организации очистных сооружений и обслуживании городского хозяйства с гигиенической точки зрения, города пришли в упадок. Именно поэтому эпоху Средневековья и называют «Тёмными веками».



## Интересно знать!

В индийском городе Мохенджо-Даро, основанном на рубеже III–II тысячелетий до н. э., архитекторами была проложена сложная разветвлённая сеть канализации (рис. 2). В XVIII веке в Париже, столице Франции, являющейся сегодня законодателем мировой моды и красоты, не существовало системы канализации: у каждого дома была собственная постоянная и неочищаемая выгребная яма.

Улицы застали грязью, по ним невозможно было пройти (рис. 3). Порой прохожие теряли сознание от охвативших город зловоний. Кроме того, численность населения городов увеличивалась: под гнётом феодалов, бедствуя, крестьяне перебирались из деревень на новые места. Домов хватало не на всех, поэтому многие «новые горожане» селились прямо на улицах. Отходы скапливались из года в год. С улиц грязь заносилась в дома на обуви, одежду и коже. Повсеместно были распространены блохи, вши и другие паразиты. Кстати, ведь именно блохи, которые перемещались по странам в шерсти крыс, были разносчиками «чёрной смерти» — чумы. Кроме того, существовало суеверие, что через распаренную горячей водой кожу в организм проникают болезни. Особенно страдали города некогда прекрасной Италии, где господствовал способствующий разложению и гниению тёплый влажный климат.

Однако печальная участь постигла в основном католическую Европу. Например, в Османской империи было принято регулярно посещать бани. Мужские и женские бани находились в разных частях богатых домов, а более бедное население довольствовалось общественными за-



**Рис. 2.** Древняя система канализации

ведениями, которые спонсировались государством. Улицы османских городов также убирались. Религия не только не запрещала, но и поощряла соблюдение чистоты. В каждом богатом доме были специальные люди, обычно женщины, которые занимались наведением порядка, включая подметание и мытьё полов. В Азии также большое внимание уделялось гигиене и чистоте жилищ. Например, об этом свидетельствуют страшные традиционные сказки японцев о специфических они (злых духах, примерно как бесы и черти) и ёкаи (аналог славянской нечисти), наказывающих живущих в грязи и не убирающихся людей. Действительно, многие болезни в Японию и королевство Корё (современные Южная и Северная Корея) были завезены именно европейцами в период открытия границ.

На Руси люди, не содержащие свой дом в чистоте, также осуждались обществом. Для гигиены использовалось мыло, одежда стиралась особым образом с золой, а дома тщательно подметались вениками и убирались. Конечно, стены также были покрыты сажей и копотью от печей, особенно зимой, но несколько раз в год проводилась большая уборка, в которой участвовали все члены семьи. Изначально дни наведения порядка совпадали с языческими праздниками, приуроченными к определённым астрономическим явлениям, затем к церковному календарю.

Ты уже понял, что для соблюдения гигиены и сохранения здоровья крайне важно заниматься уборкой. Обычно выделяют три вида уборок: лёгкую (поверхности очищаются только от видимых пятен, а также вы-



**Рис. 3.** Картина Питера Брейгеля Старшего «Фламандские пословицы»



носится мусор), регулярную (ежедневную или еженедельную, при которой протираются видимые поверхности и раскладываются по своим местам вещи) и генеральную (при которой очищается любая минимальная поверхность, включая окна и зеркала, щели между дверями и т. д.). Существуют и другие классификации уборок, например по количеству используемой воды: сухая, влажная и мокрая уборки. При сухой уборке пыль и мусор механически удаляются с помощью сухой салфетки и веника или же с помощью механических подметальных машин, а также электровеников, электрощёток и пылесосов. Такой способ отлично подходит для ежедневной уборки, но не избавляет от паразитов и аллергенов, находящихся на коврах и в труднодоступных участках. В этом случае прибегают к влажной уборке полов — это чистка поверхностей и удаление грязи с использованием небольшого количества воды и чистящих средств. Обычно для влажной уборки используют отжатые после стирки тряпки, салфетки либо применяют швабры или моющие пылесосы.

Швабра (от нем. *Schwaber*) — самый распространённый и один из наиболее эффективных инструментов для уборки, который есть в каждом доме. Она представляет собой длинную деревянную рейку с горизонтальной перекладиной внизу, на которую наматывается половая тряпка. В последние годы хозяйки используют тряпки из микроволокна — специального впитывающего материала; такую тряпку можно использовать даже для сухой уборки. Ещё более эффективным является применение швабр с мопами — половыми тряпками, удобно фиксирующимися на перекладине благодаря пошиву (рис. 4). Обычно они производятся из хлопчатобумажного материала или микрофибры и используются вместе с ручкой и крепежом (флаундером). Мопы бывают двух видов: верёвочные (похожие на шерсть пастушьих собак породы пули) или плоские, прикрепляемые с помощью липучек. Швабры предназначены для уборки в труднодоступных местах, например под шкафом или кроватью. Стоит отметить, что влажная уборка полов и полок способствует увлажнению воздуха, профилактике аллергических, лёгочных, инфекционных и бронхиальных заболеваний и уничтожению клещей, блох и других паразитов.

Последний вид уборок — это мокрая уборка. Чем она отличается от влажной? Объёмом воды! При мокрой уборке используется большой объём воды и моющих средств. Обычно к этому способу уборки прибегают только во время комплексных работ, после ремонта или же при значительных загрязнениях, с которыми не удалось справиться при влажной уборке. Мокрая убор-



Рис. 4. Швабра  
с мопами

### Кстати!

Сегодня уборку часто называют на английский манер — клинингом. В крупных компаниях существуют целые клининговые отделы, занятые поддержанием чистоты и экологической безопасности рабочих мест сотрудников.

для строго отведённых областей (ломоечные машины в общественных зданиях, на платформах метро и производственных площадях.

Итак, оптимальной является влажная уборка. Как же сделать её удобнее и сэкономить время? Механизировать! Для очистки или полировки напольных покрытий применяется машинка под названием «полотёр».



**Рис. 5.** Одна из первых моделей полотёров

ка проводится в несколько этапов и занимает достаточно много времени, зато сопровождается дезинфекцией убираемого помещения. Однако её нельзя проводить, если в доме имеется ламинат, паркет или любое иное неводостойкое покрытие. Для ручной мокрой уборки применяют также швабры с мопами, добавляя очищающие средства и используя мопы (одна зона — один моп), или же по-

льше и время? Для очистки или полировки напольных покрытий применяется машинка под названием «полотёр». Она состоит из нескольких круглых врачающихся щёток из щетины и шайб из сукна или фетра, приводимых в движение электромотором. Такая машинка позволяет многократно сэкономить силы и время на уборке.

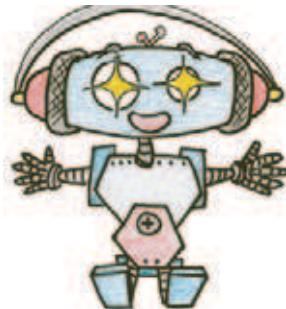
«Отцом» полотёра является австро-венгерский инженер Альфред Погранец, придумавший своё детище в 1904 году. К сожалению, его проект был несовершенен и инженер вскоре обанкротился. Но идея осталась жить. В 1912 году полотёр *Victor* фирмы AEG, доработанный Эрнстом Франке, поступил в серийное производство. Конструкция отличалась простотой и содержала лишь электродвигатель, съёмный фетровый диск и одну щётку (рис. 5). Однако заслуга Франке была в другом: он просчитал компенсацию момента силы, благодаря чему оператор полотёра мог легче перемещать машинку по всем направлениям. В СССР термин «полотёр» получил более общий смысл: им стали именовать любые машинки для очистки полов независимо от



того, являются ли они ручными, механическими или электрическими. Сегодня полотёры обычно имеют от одной до четырёх неподвижных или, наоборот, вращающихся щёток, электродвигатель (15 тыс. оборотов в минуту), щёткодержатели, ручки для удобного перемещения по поверхности, набор шайб и корпус. Принцип передачи вращения щёткам может быть совершенно любой, включая зубчатую или ременную передачу. Главное, чтобы они двигались со скоростью 600–900 оборотов в минуту. Более усовершенствованные модели комбинируются с пылесосом, имеют «плавающие» щётки, специальные контейнеры для жидкости и множество дополнительных насадок.

Отлично! Но как ещё можно упростить себе жизнь? Сделать так, чтобы полотёром не нужно было управлять вручную! И да, ты уже догадался, в этом тебе поможет робот! Вперёд, старший специалист клинингового отдела!

## Этап 1. Устройство робочиста



Рассмотри модель роботизированного полотёра (робочиста), собранную на основе набора LEGO® MINDSTORMS® Education EV3.

Попробуй выделить на ней рабочие детали — два мотора, в которых установлены уборочные валики; ультразвуковой датчик для обнаружения препятствий; датчик цвета для предотвращения падения с высоты; гироскопический датчик, обеспечивающий точность поворотов.

Попробуй собрать эту модель.

Обрати внимание: в конце проекта даны все детали, которые потребуются тебе для сборки.

Вот так выглядит робочист, которого тебе предстоит собрать (рис. 6).

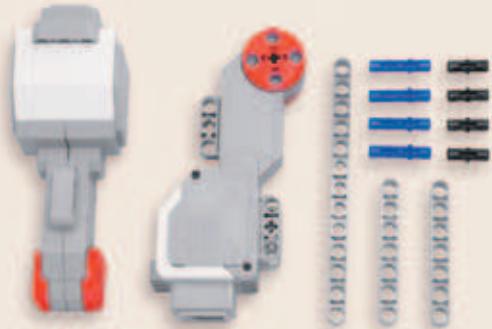


**Рис. 6.** Общий вид робочиста

## ШАГ 1. СБОРКА ПРИВОДОВ ДЛЯ РОБОЧИСТА

Детали для сборки:

- мотор большой, 2x;
- балка № 7, серая, 2x;
- балка № 13, серая, 1x;
- соединительный штифт, 2-модульный, чёрный, 4x;
- соединительный штифт, 3-модульный, синий, 4x.



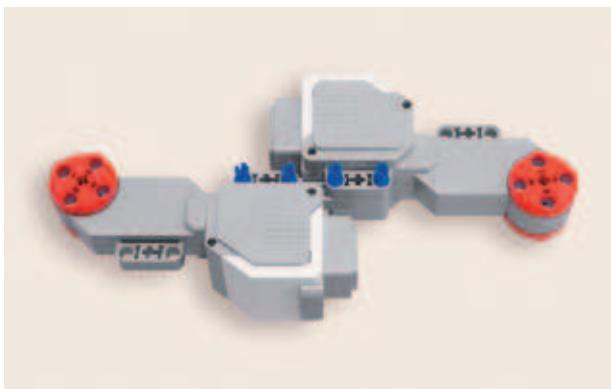
1. Возьми один большой мотор и положи его так, чтобы порт был слева, а привод — справа, и вложи в нижнее крепление между балками серую балку № 7.



2. Вставь в модули мотора два синих 3-модульных штифта длинными концами вперёд.



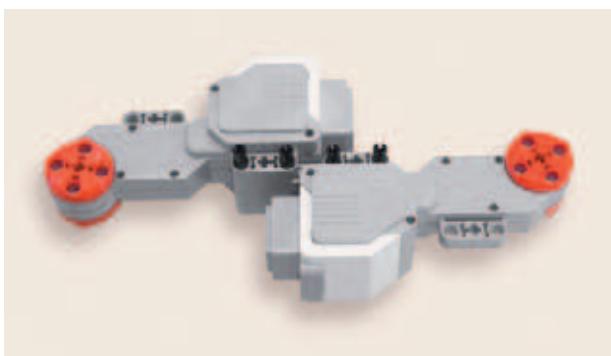
3. Возьми другой большой мотор и поверни его приводом влево, а разъемом вправо, но нижним креплением вверх. Расположи его так, чтобы балка № 7 была вложена между креплениями мотора, и закреши его синими 3-модульными штифтами, вставив их длинными концами вперёд.



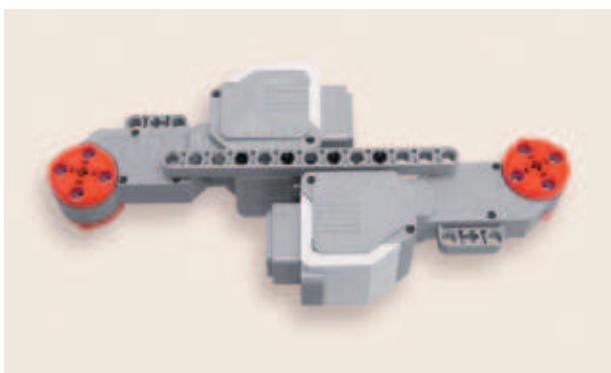
4. Теперь возьми другую балку № 7 и надень её сверху на свободные короткие стороны синих штифтов в первый, третий, пятый и седьмой модули.



5. Переверни собранную конструкцию балкой № 7 от себя и установи в нижние крепления двух больших моторов четыре чёрных 2-модульных штифта в отверстия на моторах.



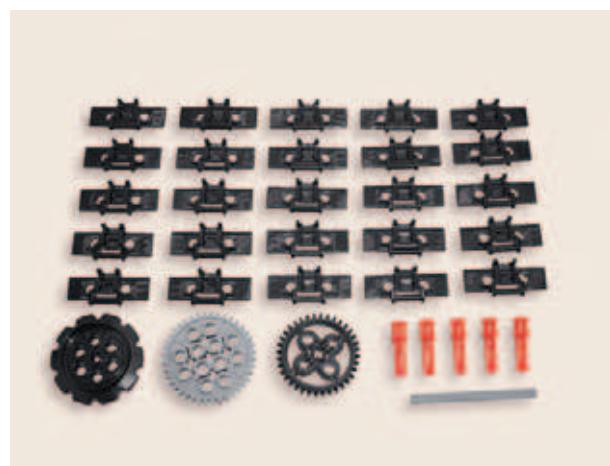
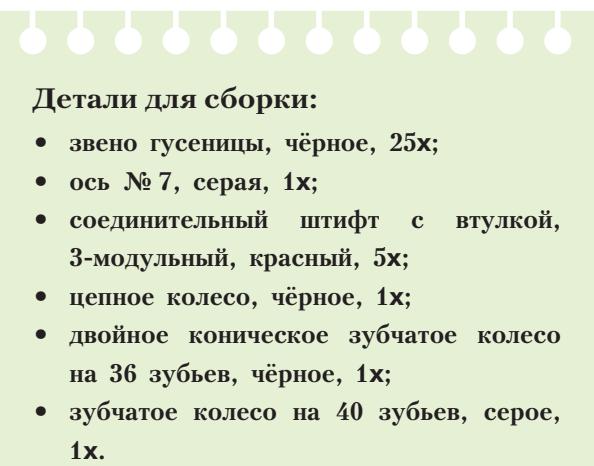
6. Надень сверху на чёрные штифты серую балку № 13 на четвёртый, шестой, восьмой и десятый модули.



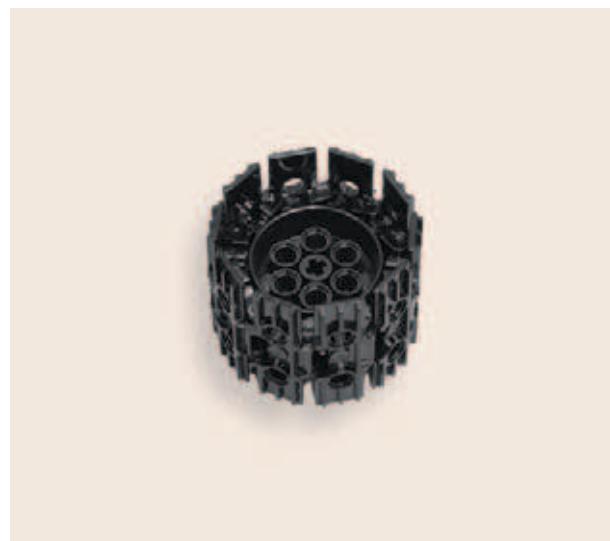
Подвижная часть робочиста готова!



## ШАГ 2. СБОРКА УБОРОЧНЫХ ВАЛИКОВ ДЛЯ РОБОЧИСТА



1. Возьми 10 звеньев гусеницы и соедини их. Потом возьми одно цепное колесо, оберни собранную гусеницу вокруг него и защелкни последнее звено так, чтобы оно оказалось внутри специальных упоров внутри кольца.



Однако уборочный валик по-прежнему маловат — увеличим его с помощью ещё одной гусеничной ленты.



2. Возьми оставшиеся 15 звеньев и собери вторую часть гусеницы. Оберни цепь поверх малой гусеницы.



3. Закрепи внешнее кольцо вместе с внутренним с помощью пяти красных штифтов, вставив их насеквоздь через совмещённые отверстия в гусеницах, как показано на рисунке.



4. Переверни получившуюся деталь так, чтобы все штифты оказались ближе к поверхности стола. В центр зубчатого колеса вставь в его крестообразное отверстие ось № 7 так, чтобы она выступала только с одной стороны.



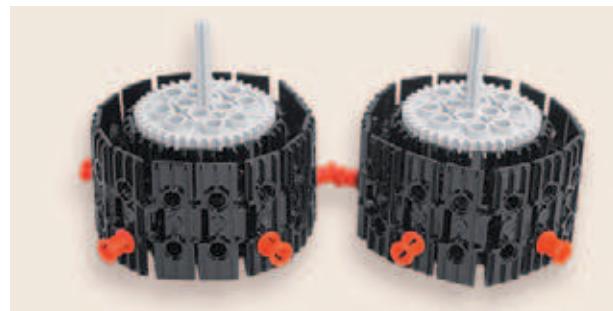
5. Надень на длинный конец оси № 7 одно двойное коническое зубчатое колесо на 36 зубьев.



6. Надень сверху серое зубчатое колесо на 40 зубьев.



7. Аналогично собери ещё один такой же уборочный валик.



8. Возьми собранную конструкцию с моторами и установи в приводы мотора по одному валику с обратной стороны от балки № 13 так, чтобы оси валиков не выступали сверху.

На этом сборка функциональной части завершена.



### ШАГ 3. СБОРКА КРЕПЛЕНИЯ И УСТАНОВКА МОДУЛЯ EV3 НА РОБОЧИСТА



#### Детали для сборки:

- соединительный штифт, 2-модульный, чёрный, 4x;
- соединительный штифт, 3-модульный, синий, 2x;
- программируемый модуль EV3, 1x;
- балка № 7, серая, 2x.



1. Возьми модуль EV3 и переверни его дисплеем вниз и портами, обозначенными буквами, к себе.



2. Возьми четыре чёрных соединительных штифта и вставь их в крайние модули EV3.

3. Надень сверху на штифты по одной балке № 7 первым и седьмым модулями.



4. Возьми два синих 3-модульных штифта и установи их в пятые модули балки № 7 длинными концами вперёд.



5. Надень сверху на свободные концы синих штифтов собранную конструкцию с моторами балкой № 13 в её первый и тринадцатый модули.



Отлично! Теперь у твоего робота появились «мозги», и он сможет самостоятельно убираться! Осталось совсем немного!

## ШАГ 4. УСТАНОВКА ДАТЧИКОВ НА РОБОЧИСТА



### Детали для сборки:

- гироскопический датчик, 1x;
- ультразвуковой датчик, 1x;
- датчик цвета, 1x;
- соединительный штифт, 2-модульный, чёрный, 6x;
- соединительный штифт, 3-модульный, синий, 2x;
- прямоугольная балка 3×5, белая, 2x;
- кабель, 25 см, 4x;
- кабель, 35 см, 1x.



1. Возьми собранную конструкцию вместе с EV3 и поставь её на опоры, чтобы порты, обозначенные цифрами, были слева.



2. Возьми два чёрных соединительных штифта и вставь их в горизонтальные модули слева на блоке EV3, расположенные ближе к цифровым портам: в первый и третий.



3. Надень на свободные концы чёрных штифтов гироскопический датчик разъёмом влево.



4. Теперь разверни собранную конструкцию.



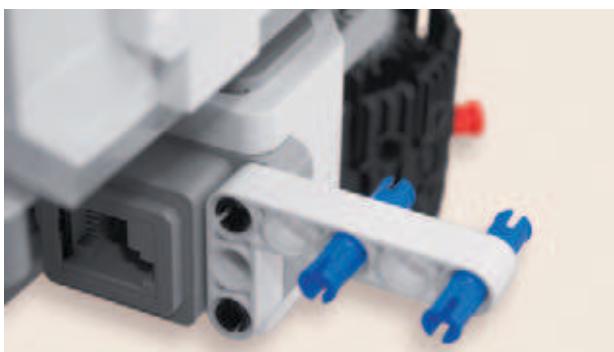
5. Возьми два чёрных штифта и установи их в первый и третий модули, расположенные на правом большом моторе.



6. Надень на свободные концы штифтов белую прямоугольную балку  $3 \times 5$  короткой стороной в первый и третий модули так, чтобы длинная сторона балки находилась сверху.



7. Вставь в первый и третий модули длинной стороны прямоугольной балки справа два синих соединительных штифта длинными концами вперёд.



8. Надень на синие штифты справа ультразвуковой датчик.



9. Установи слева от ультразвукового датчика белую прямоугольную балку 3×5 короткой стороной на свободные концы синих штифтов таким образом, чтобы длинный конец белой балки «смотрел» вниз.



10. Вставь в третий и пятый модули балки 3×5 два чёрных соединительных штифта.



11. Надень на свободные концы штифтов датчик цвета разъёмом вверх.



Теперь твой робот никогда не наткнётся на препятствие и не упадёт со стола!

## ШАГ 5. ПОДКЛЮЧЕНИЕ ПРОВОДОВ

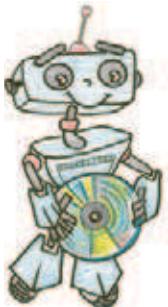
1. Возьми один кабель 25 см и соедини ультразвуковой датчик с цифровым портом № 3.
2. Возьми другой кабель 25 см и подключи гирокопический датчик к цифровому порту № 4.
3. Возьми третий кабель 25 см и соедини левый большой мотор с портом С.
4. Возьми четвёртый кабель 25 см и соедини большой мотор, «смотрящий» вправо, с портом В.
5. Возьми провод 35 см и соедини датчик цвета и порт № 2.



Молодец! Теперь ты сможешь порадовать родителей своим чистейшим столом и натёртыми до блеска полами!



## Этап 2. Установка программного обеспечения на компьютере



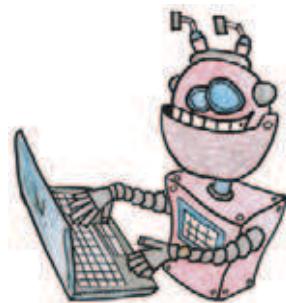
### Внимание!

При любых затруднениях, возникших при установке программного обеспечения, обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.

1. Если ты приобрёл базовый набор LEGO® MINDSTORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com>

и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3. После запуска установочного файла откроется окно мастера установки, в котором тебе нужно следовать пошаговым указаниям. Не забудь проверить установочный путь! Если на твоём жёстком диске не останется свободного пространства, придётся удалить ненужные программы или файлы.

# Этап 3. Создание программы для робочиста



## ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

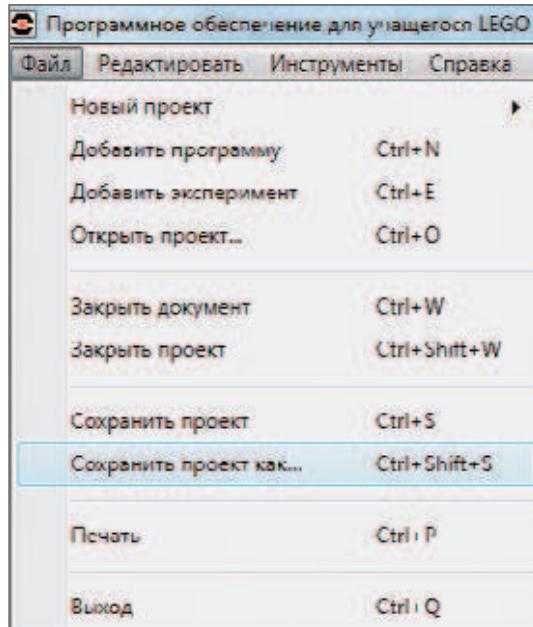
1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

## СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

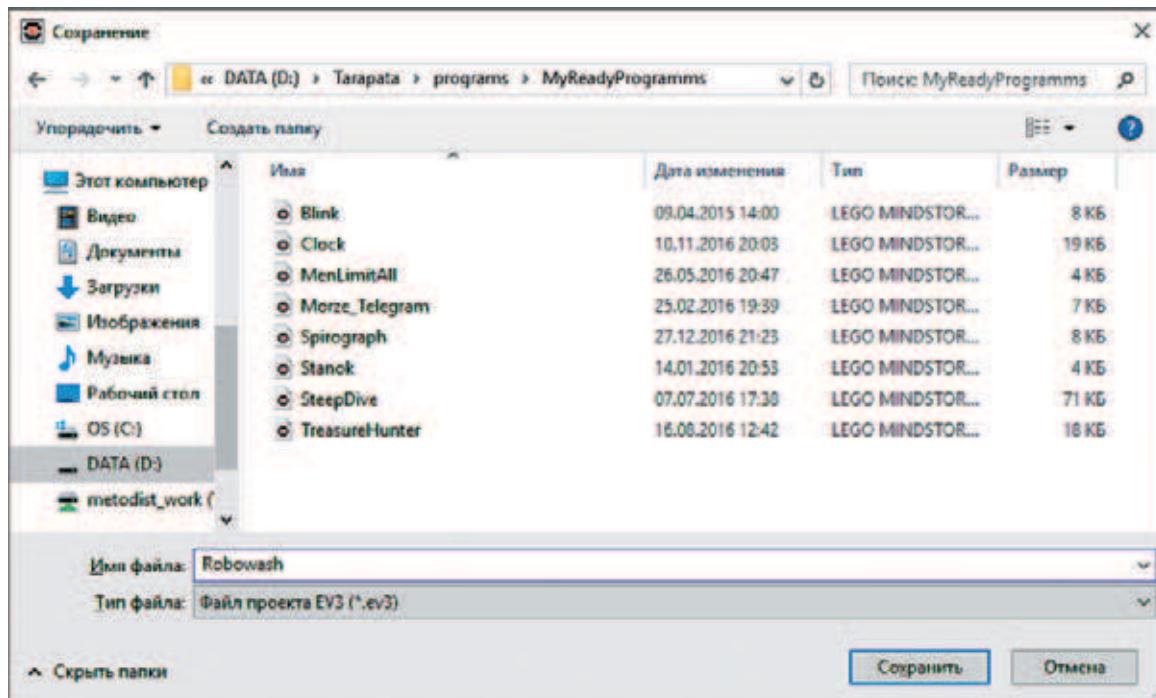
Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, не нужно искать его в компьютере, он всегда находится внутри робота.

Однако необходимо грамотно организовать хранение проектов. Папка с проектом должна иметь название, отражающее его суть, иначе тебе будет трудно её найти. Если в папке с проектом несколько программ, то назови их максимально понятно для себя.

1. Для начала сохрани свой проект в памяти компьютера, чтобы избежать его потери. Для этого выбери меню **Файл** в левом верхнем углу окна программы и в контекстном меню выбери пункт **Сохранить проект как...**



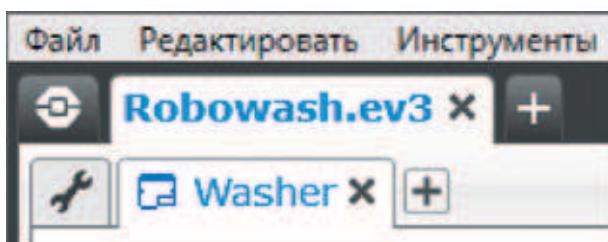
- Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла**: *Robowash* (в переводе с английского «Робомойка») в нижней части окна и нажми **Сохранить**.



### Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и без использования пробелов. Вместо них они, например, пишут каждое слово с большой буквы или используют нижнее подчеркивание.

- Именно так будет называться папка в памяти программируемого модуля EV3, в которой будут храниться все программы, написанные тобой в рамках этого или других проектов.



- Осталось дать имя самой программе (в нашем проекте она будет одна). Для этого дважды кликни по имени **Program** в левом верхнем углу поля программы, задай новое имя: *Washer* (в переводе с английского «Мойщик») и нажми клавишу **Enter**.



## ЛОГИКА ПРОГРАММЫ

Программа будет состоять из двух одновременно выполняющихся ветвей: одна будет отвечать за движение робота по прямой линии до препятствия и разворот, а другая — за предотвращение падения робота. Наш робот будет «ехать» вперёд, передвигаясь сначала одной стороной, затем другой, врача свои моющие валики. Каждый такой «шаг» будет осуществляться поворотами на 30 градусов от первоначального направления робота.

Как только в поле зрения ультразвукового датчика будет попадать какой-либо объект, робот будет разворачиваться на 180 градусов и направляться в противоположную сторону.

Если робот будет двигаться по столу, то он может упасть. Во избежание этого мы предусмотрим его аварийную остановку и откат обратно с изменением направления дальнейшего движения.

В основном мы будем работать с градусами — это единицы измерения углов. Если понятие градусной меры угла тебе незнакомо, то ты можешь обратиться за помощью к своему учителю, учебнику математики, родителям или Интернету.

### Кстати!

В данной программе будет несколько интересных алгоритмических конструкций и приёмов программирования. Один из них — это вложенные циклы, останавливающиеся при выполнении некоторых условий, и принудительные прерывания, а также запуск нескольких программ на параллельное исполнение.

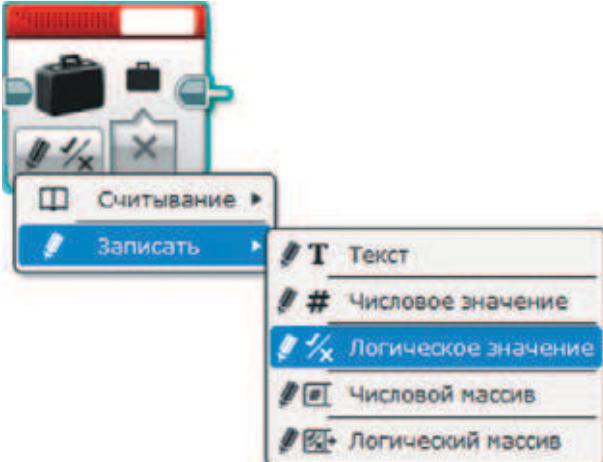
## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОЧИСТА

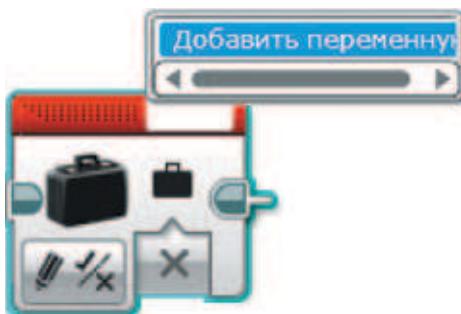
В открывшемся окне проекта начнай составлять программу для робочиста, перетаскивая блоки команд в рабочую область.

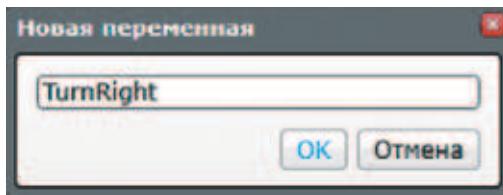
### Программа 1. Мойка

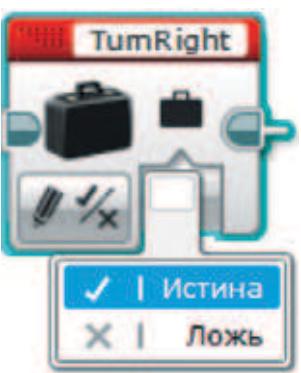
- Нам понадобится логическая переменная (она принимает только два значения: истина и ложь). После обнаружения препятствия роботу нужно разворачиваться каждый раз в разные стороны, чтобы он не ездил по одной и той же траектории. Задать начальное значение переменной нужно в самом начале, поэтому сразу после команды **Начало** добавь команду **Переменная (красный блок)**.



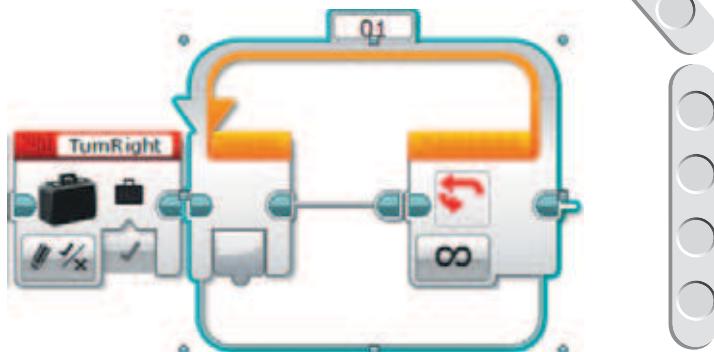
- 
2. Выбери опцию **Записать** → **Логическое значение**.

- 
3. Кликни по правому верхнему окошку и в открывшемся меню выбери **Добавить переменную**.

- 
4. Введи в открывшемся окне имя переменной **TurnRight** (в переводе с английского «Поворот направо») и нажми **OK** — она будет отвечать за то, в какую сторону будет разворачиваться робот (в правую или в левую), в зависимости от истинности или ложности её значения.

- 
5. Выбери в качестве начального значения переменной **Истина**.

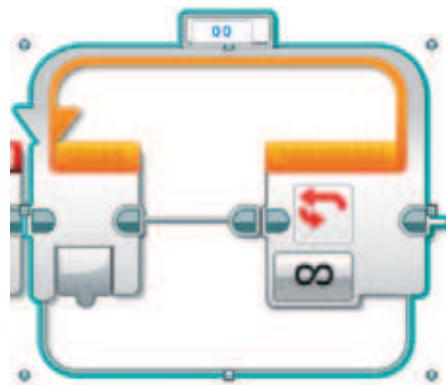
6. Далее добавь бесконечный **Цикл** (оранжевый блок), так как робот будет работать до тех пор, пока его не выключат (ну или пока не сядет батарейка). Вся остальная часть этой программы будет содержаться внутри этого цикла.



7. В нашей программе будет несколько циклов, поэтому необходимо их различать по номерам. В верхней части этой команды по умолчанию указан номер **01**. Такой цикл будет внешним, то есть находящимся на самом верхнем уровне, поэтому кликни дважды по номеру и замени его на номер **00**. Этот номер называется **идентификатором цикла**.

**Кстати!**

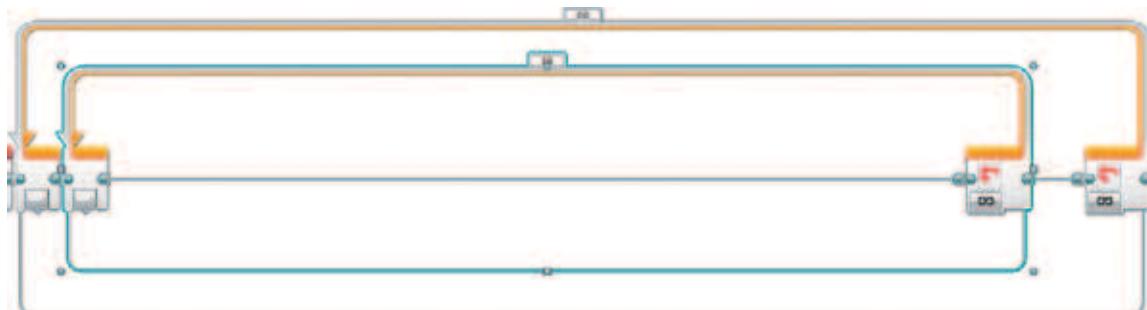
В компьютере, а также в программировании и информатике нумерация всегда начинается не с единицы, а с нуля. Это связано с электричеством — ноль означает отсутствие напряжения.



8. При выделении команды **Цикл** появляются кружочки и квадратики. Потяни за них и измени размер этого блока — сделай его побольше.



9. Добавь внутрь этого цикла ещё один бесконечный **Цикл** (оранжевый блок), расширий его так же, как и внешний, а идентификатор замени на число **10** (договоримся, что первая цифра будет указывать на уровень глубины цикла, а вторая — на номер цикла на этом уровне).

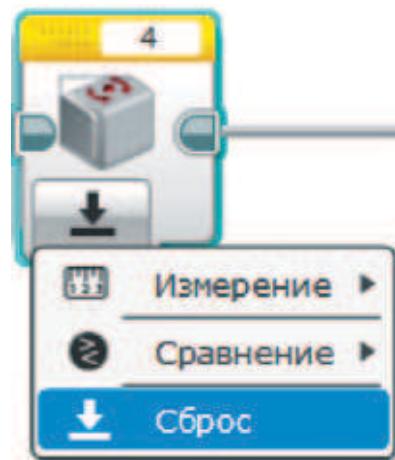


10. Добавь в этот цикл команду **Гирокопический датчик** (жёлтый блок).



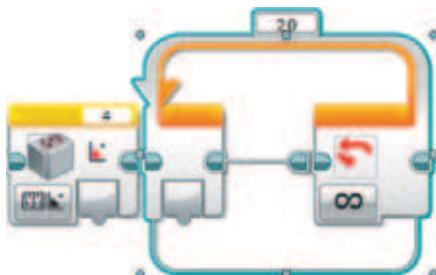
**Внимание!**

Проверь, чтобы в правом верхнем окошке этой команды значился порт, к которому подключён гирокопический датчик. В нашем случае это порт № 4.



11. Для точности замеряемого угла нужно сбросить показания гирокопического датчика. Выбери опцию **Сброс**.

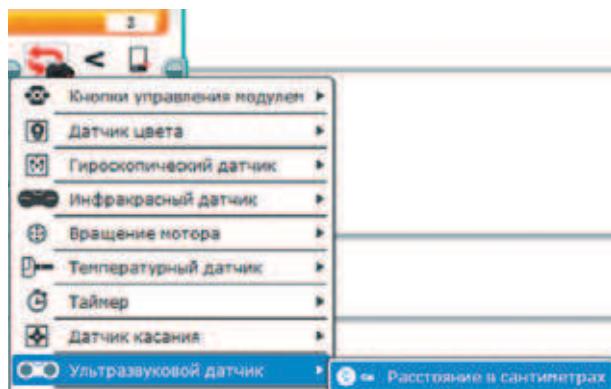
12. Далее тебе необходимо будет запускать поочерёдно валики робота, пока робот не «наткнётся» на препятствие. Для этого добавь ещё один **Цикл** (оранжевый блок) и задай его идентификатором число **20**.



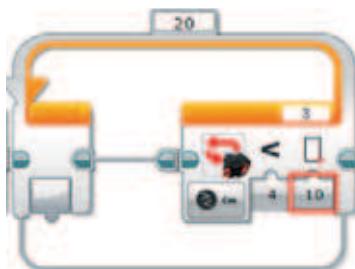
13. Этот цикл не будет бесконечным и прервётся, когда в поле зрения ультразвукового датчика попадёт какой-либо объект на расстоянии менее 10 см. Чтобы произошло прерывание, в опции цикла выбери **Ультразвуковой датчик → Расстояние в сантиметрах**.

#### Внимание!

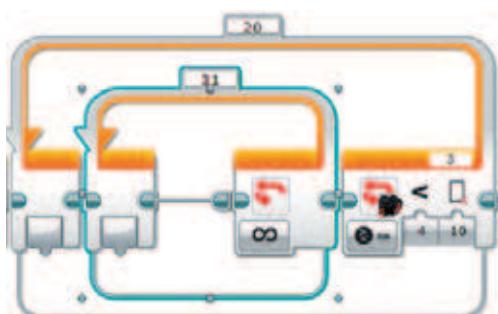
Проверь, чтобы в правом верхнем окошке этой команды значился порт, к которому подключён гироскопический датчик. В нашем случае это порт № 3.

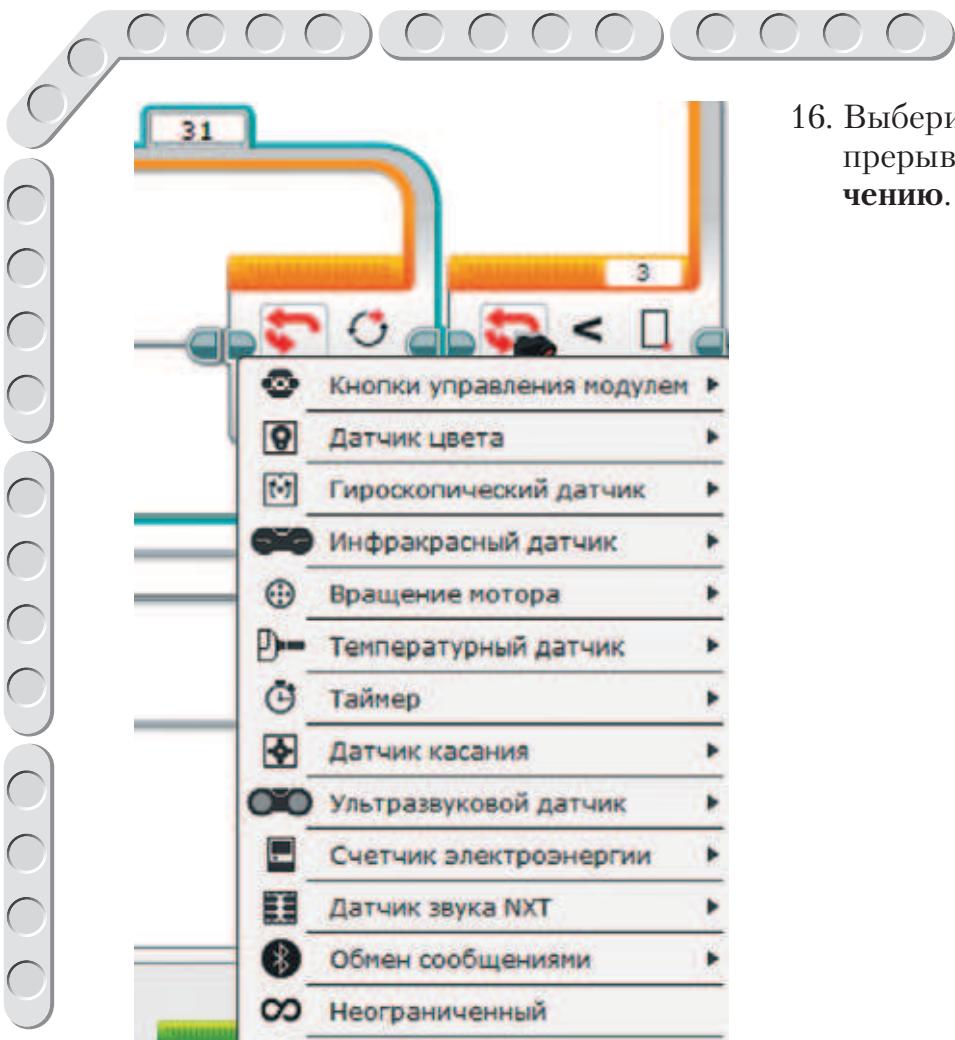


14. Введи в качестве порогового значения с клавиатуры число **10**.



15. Наверное, ты догадался, что внутри этого цикла будет ещё один цикл (вот такая сложная конструкция из четырёх циклов, но не беспокойся: дальше этого уровня мы не уйдём). Добавь команду **Цикл** (оранжевый блок) и задай его идентификатором число **31**.

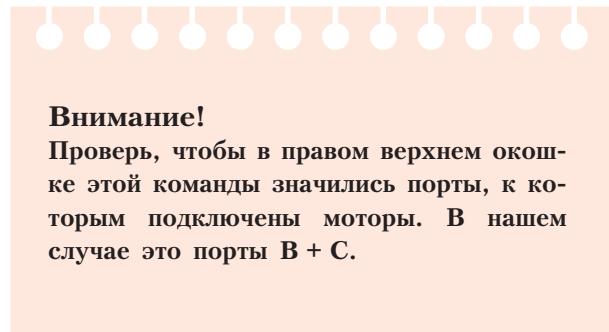




16. Выбери для этого цикла опцию прерывания по **Логическому значению**.



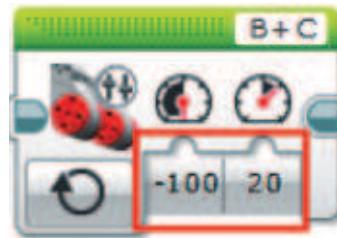
17. Внутри этого цикла мы должны запустить валики на движение, а продолжаться движение должно либо до того, как робот повернётся на 30 градусов, или до того, как «увидит» препятствие. Поэтому добавь в него команду **Независимое управление моторами** (зелёный блок).



18. Выбери опцию **Включить**.



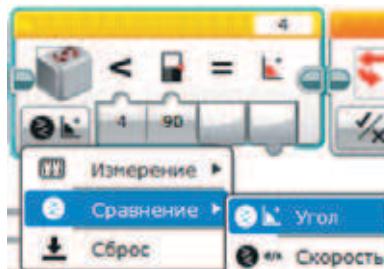
19. Включи один мотор на полную мощность, а другой на небольшую и в противоположную сторону, чтобы робот эффективнее поворачивался, поэтому задай: **Мощность на левый мотор: -100;** **Мощность на правый мотор: 20.**



20. Теперь добавь команду **Гирокос-  
тический датчик** (жёлтый блок).



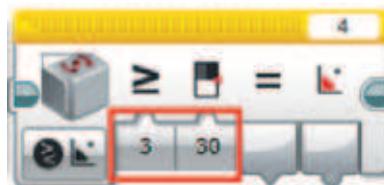
21. Выбери опцию **Сравнение** →  
→ **Угол.**



22. Робот должен повернуться на 30 градусов, поэтому задай: **Тип срав-  
нения: 3** (больше или равно); **Пороговое значение: 30.**

#### Внимание!

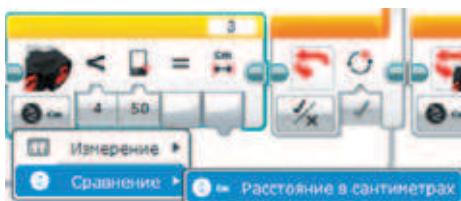
В разных наборах гирокостический датчик работает по-разному и значения при повороте направо и налево меняются как в положительную, так и в отрицательную сторону. Сравнение и пороговое значение нужно изменить на противоположные.



23. Далее добавь команду **Ультразвуковой датчик** (жёлтый блок).



24. Выбери опцию **Сравнение → Расстояние в сантиметрах**.



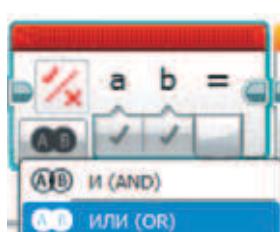
25. Задай в качестве **Порогового значения** число **10**.



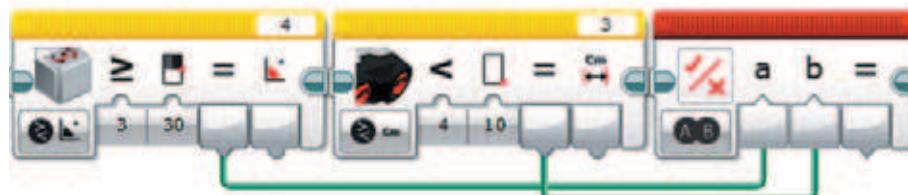
26. Теперь добавь команду **Логические операции** (красный блок).



27. Выбери опцию **ИЛИ**.

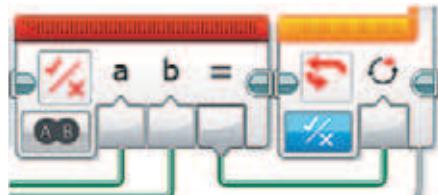


28. Теперь соедини выход **=** команды **Гирокопический датчик** со входом **a** **Логической операции ИЛИ**, а выход **=** команды **Ультразвуковой датчик** со входом **b**.

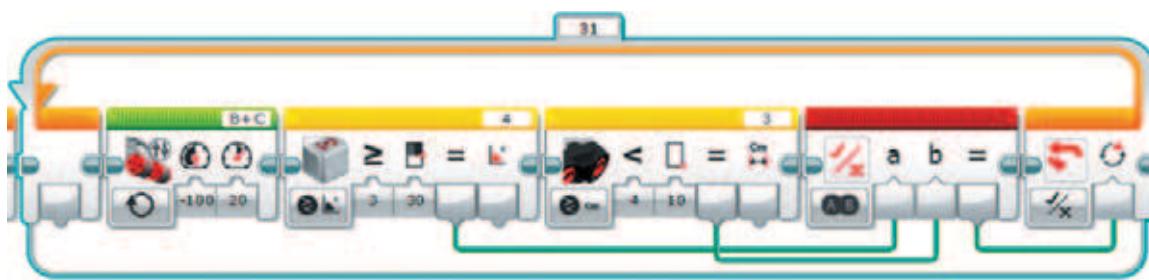


Созданная конструкция буквально говорит о следующем: «Робот уже повернулся на 30 градусов ИЛИ „увидел“ препятствие». Результатом этой операции является истина или ложь, причём истина получится при выполнении хотя бы одного из условий.

29. Если в результате операции получена истина, то цикл, в котором содержатся все эти команды (**31**), прервётся. Чтобы это произошло, соедини выход = **Логической операции** со входом **Цикла**.



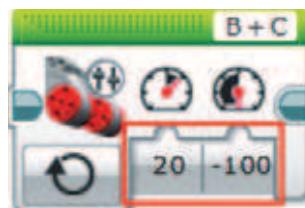
30. Теперь проверь себя — **Цикл 31** должен иметь следующий вид:



31. Далее необходимо создать похожий цикл, но для движения робота в другую сторону, то есть фактически цикл и его содержимое будут такими же, но с немного другими параметрами команд. Поэтому зажми клавишу **CTRL** на клавиатуре и потяни **Цикл 31** в сторону — создастся его копия. Помести её сразу после него и задай идентификатор **32** (второй цикл на уровне 3).



32. Теперь поработаем с его содержимым. Задай в команде **Независимое управление моторами**:  
**Мощность на левый мотор: 20;**  
**Мощность на правый мотор: -100.**



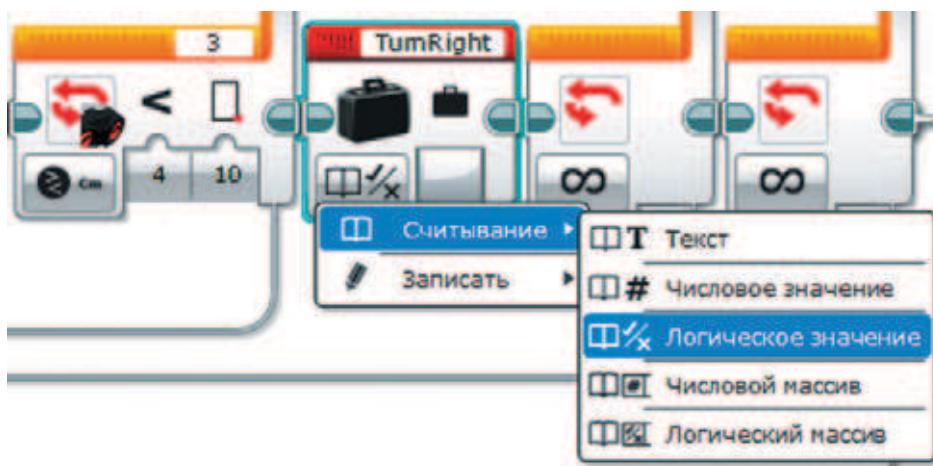
33. В команде **Гироскопический датчик**:

**Тип сравнения:** 5 (меньше или равно);

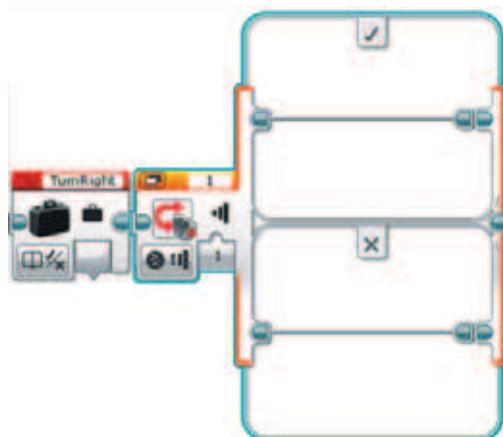
**Пороговое значение:** -30.

Остальные команды останутся без изменений.

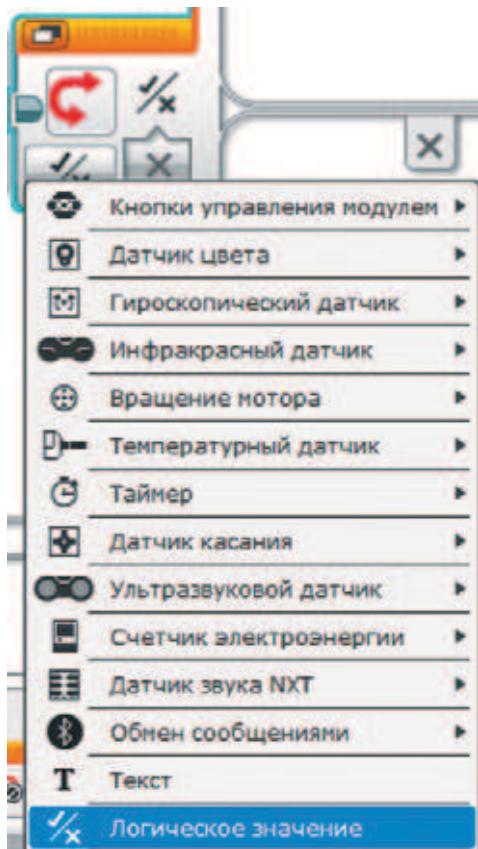
34. Отлично! Робот почти готов к работе, ты уже научил его основной функции. Однако нужно сделать его «умнее». После того как прервётся **Цикл 20** (он прерывается, если робот «видит» препятствие), нужно решить, в какую сторону ему разворачиваться. А за это отвечает содержимое переменной **TurnRight** — проверим её. Сразу за **Циклом 20** добавь команду **Переменная** (красный блок) и выбери опцию **Считывание → Логическое значение**.



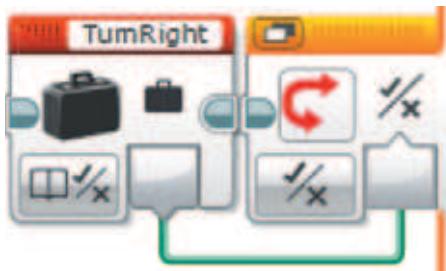
35. Далее добавь команду **Переключатель** (оранжевый блок).



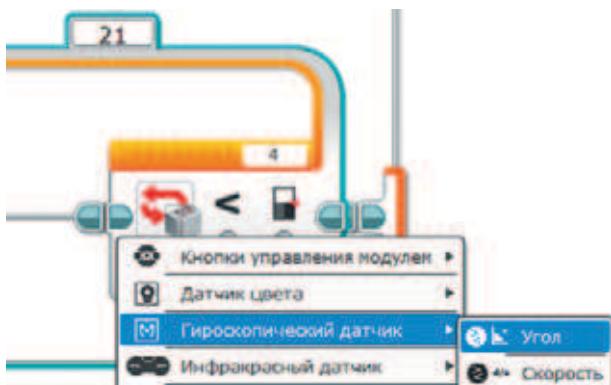
36. Выбери опцию **Логическое значение**.

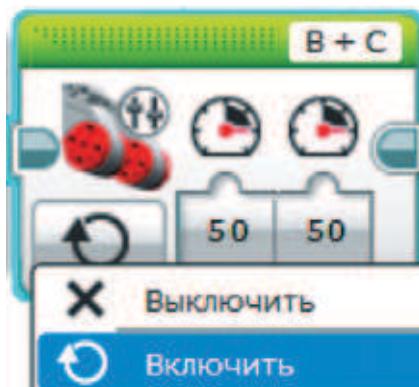


37. Соедини выход переменной **Turn-Right** со входом **Переключателя**.

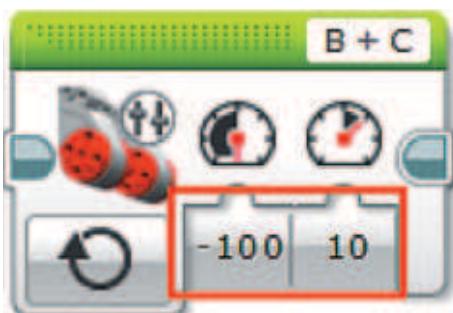


38. Если в этой переменной содержится истина, значит, робот должен повернуться направо на 180 градусов; если ложь, то налево. В верхней части **Переключателя** добавь **Цикл** (оранжевый блок), задай идентификатор **21** и выбери опцию **Гирокопический датчик** → **Угол**.

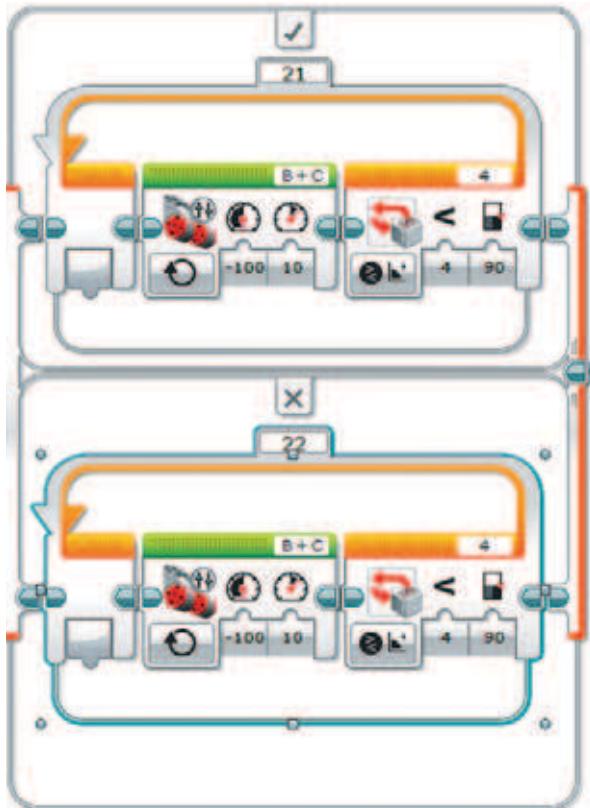




39. Внутри этого цикла нужна лишь одна команда **Независимое управление моторами** (зелёный блок) с опцией **Включить**.

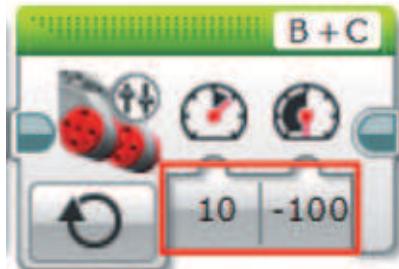


40. Настрой её параметры:  
**Мощность на левый мотор:** **-100**;  
**Мощность на правый мотор:** **10**.

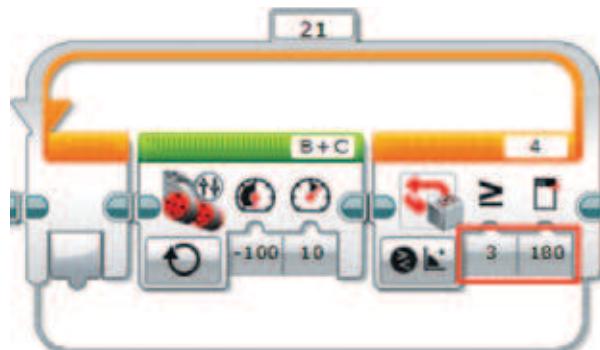


41. В нижней части **Переключателя** (в случае ложности переменной **TurnRight**) нужно создать точно такой же цикл, только настроить работу моторов противоположно. Скопируй цикл из верхней части: для этого удерживай клавишу **CTRL** и потяни этот цикл в сторону, а затем помести в нижнюю часть и смени идентификатор на **22**. В результате получится так, как показано на рисунке слева.

42. Смени параметры команды **Независимое управление моторами**:  
**Мощность на левый мотор: 10;**  
**Мощность на правый мотор: -100.**



43. Осталось поменять параметры прерывания циклов. Для **Цикла 21**:  
**Тип сравнения: 3** (больше или равно);  
**Пороговое значение: 180.**



44. Для **Цикла 22**:  
**Тип сравнения: 5** (меньше или равно);  
**Пороговое значение: -180.**



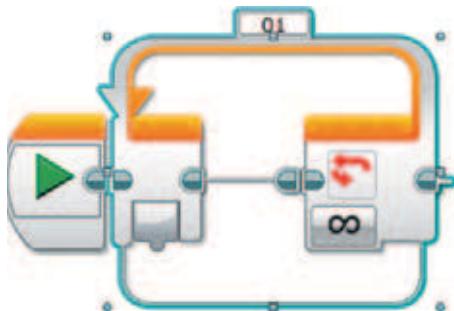
45. Осталась ещё одна команда в этой ветви. **Цикл 10** нужно прервать в аварийной ситуации, когда робот попадёт на край стола (или любой другой поверхности). При этом робот должен некоторое время не предпринимать никаких действий, кроме отката назад. Для этого после **Цикла 10** добавь команду **Ожидание** (оранжевый блок) и задай в качестве параметра задержки **5** секунд.



Молодец! Ты запрограммировал работу робочиста! Теперь остаётся предупредить аварию — падение робота с высоты!

## Программа 2. Аварийная ситуация

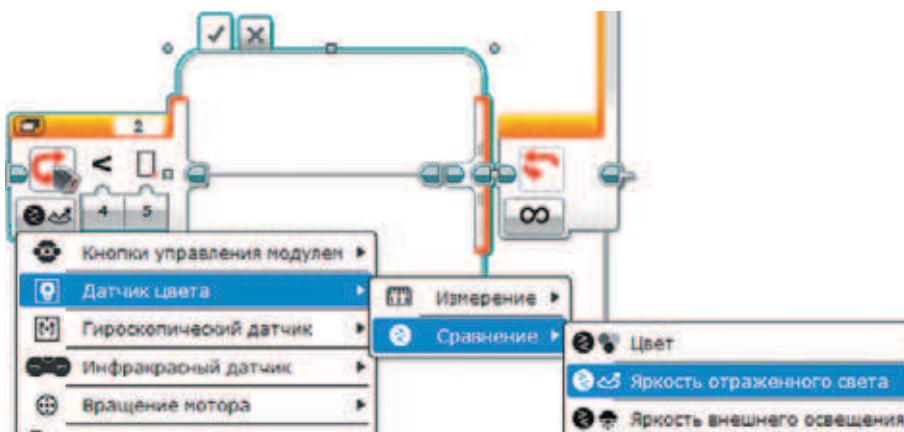
Как мы говорили, в твоей программе будут одновременно выполняться две её ветви. Поэтому сейчас ты добавишь ещё одно начало программы.



1. Добавь команду **Начало** (оранжевый блок) чуть ниже первой ветви. Вся остальная программа будет прописываться в этой ветви.

2. Как мы обсуждали в логике программы, нам понадобится предупредить падение робота. Для этого у нас есть датчик цвета, который может работать по яркости отражённого света. Как раз-таки если робот подъедет к краю, а датчик перестанет «видеть» поверхность, то это будет означать, что яркость отражённого света равна нулю. Создадим такую проверку и действия при аварийной ситуации. Так как проверка должна производиться всегда, добавь бесконечный **Цикл** (оранжевый блок), его идентификатор можно оставить по умолчанию **01**.

3. Единственное, что робот должен проверить, — это яркость отражённого света по датчику цвета, и если она упала, скажем, ниже пяти, то должен прерваться **Цикл 10**, а сам робот отъехать назад и повернуться в другую сторону. Поэтому внутрь цикла добавь команду **Переключатель** (оранжевый блок). Для удобства перейди к **Виду с вкладками**, нажав кнопку в левом верхнем углу. Выбери опцию **Датчик цвета** → **Сравнение** → **Яркость отражённого света**.



4. Настрой параметры **Переключателя**:  
**Тип сравнения:** 4 (меньше);  
**Пороговое значение:** 5.



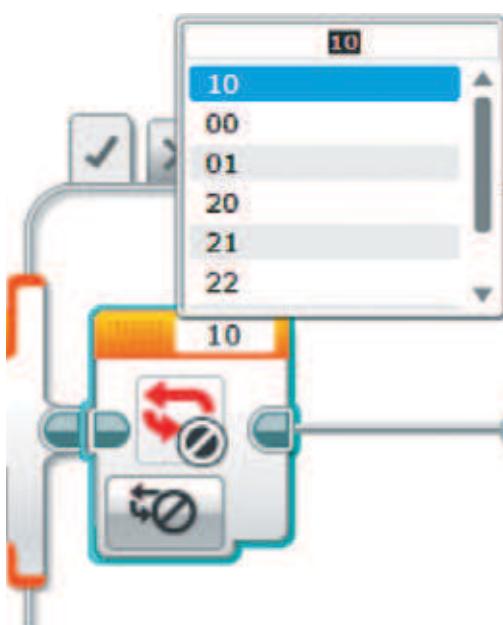
### Внимание!

Проверь, чтобы в правом верхнем окошке этой команды значился порт, к которому подключён датчик цвета. В нашем случае это порт № 2.

### Внимание!

Если планируется работа робота на тёмной поверхности, то необходимо замерить показания датчика цвета на ней и внести корректировки в параметр проверки. Для замеров ты можешь воспользоваться режимом Port View, доступным на вкладке настроек в программируемом модуле EV3.

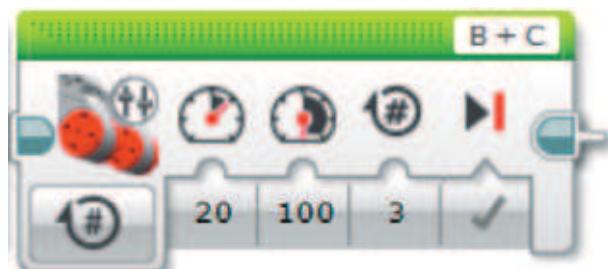
- Добавь внутрь **Переключателя** на вкладке, отмеченный галочкой (истина), команду **Прерывание цикла** (оранжевый блок) и выбери в правом верхнем окошке **Цикл 10**.



- Далее добавь команду **Независимое управление моторами** (зелёный блок), оставь опцию по умолчанию: **Включить на количество оборотов**.



- Осталось задать параметры работы моторов:  
**Мощность на левый мотор: 20;**  
**Мощность на правый мотор: 100;**  
**Обороты: 3.**



В нашем случае робот отъезжает назад и при этом рисунок поворачивается на 90 градусов. Это также зависит от типа поверхности работы робота и используемых моечных материалов на валиках. При необходимости измени параметр **Обороты**.

Поздравляем! Твоя программа готова к первому пуску!  
Совсем скоро твой рабочий стол и полы заблестят от чистоты, а родители будут нескованно рады и счастливы за своего юного инженера!

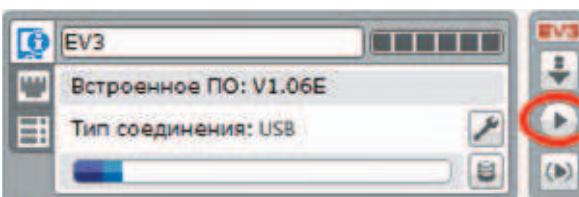
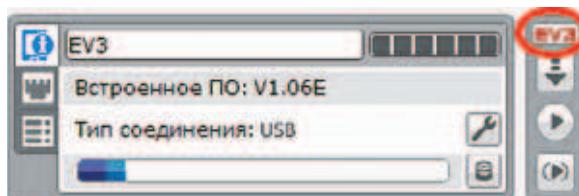
# Этап 4. Загрузка программы и её тестирование



## ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активным блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



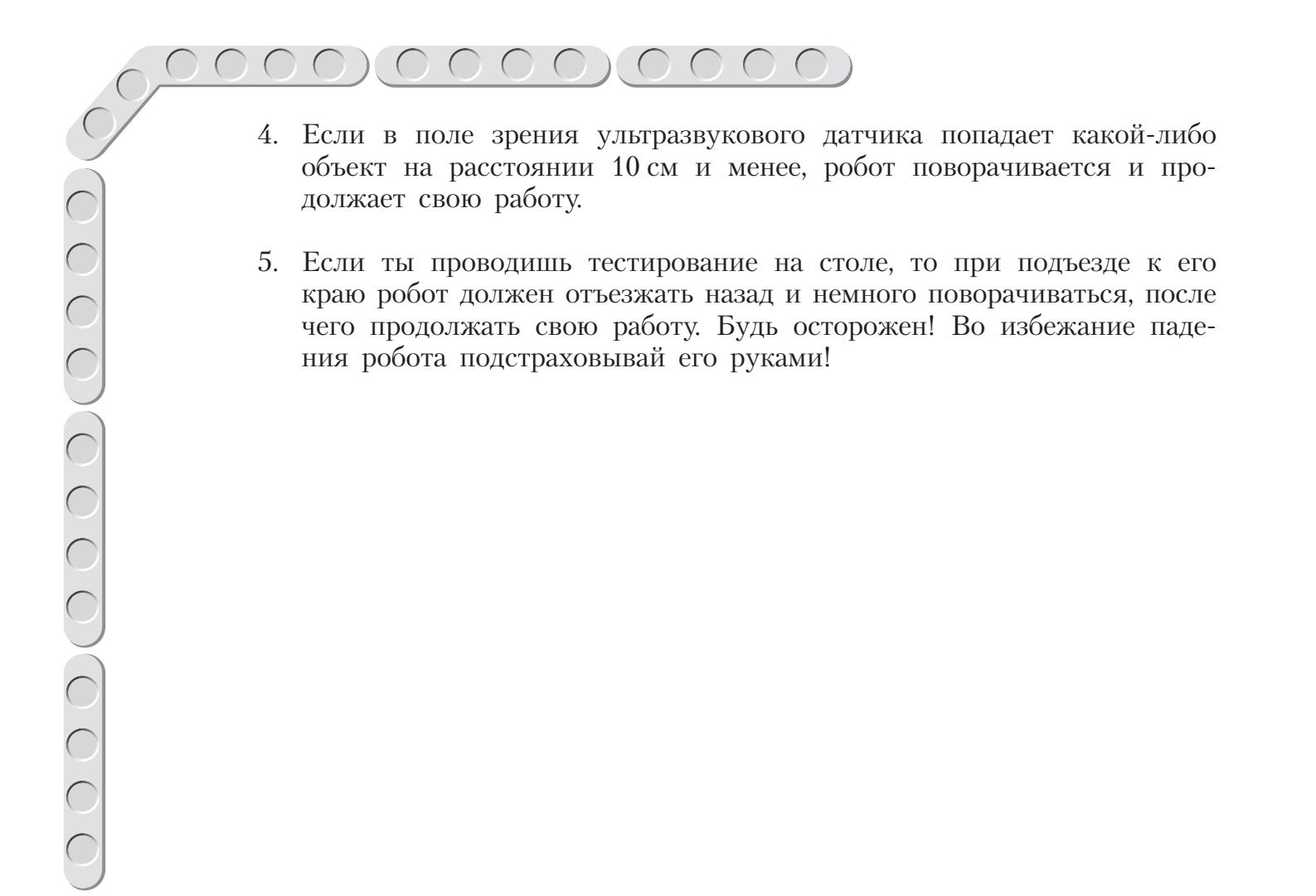
## ТЕСТИРОВАНИЕ

1. Перед запуском программы обязательно надень на моющие валики какие-нибудь тряпочки для протирки.
2. Убедись, что работу ничего не мешает для движения.
3. При движении робот совершает повороты своим корпусом на 30 градусов от начального положения то в одну, то в другую сторону.

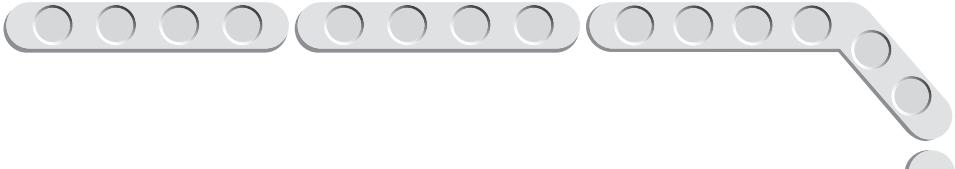
### Внимание!

Наша программа — это бесконечный цикл. Поэтому остановить её можно только вручную с помощью кнопки Отмена на программируемом модуле. Если ты хочешь повторно запустить программу или продемонстрировать работу своего робота другу, то для его запуска компьютер тебе уже не нужен.

Включи программируемый блок, затем выбери папку Robowash и в ней программу Washer.

- 
4. Если в поле зрения ультразвукового датчика попадает какой-либо объект на расстоянии 10 см и менее, робот поворачивается и продолжает свою работу.
  5. Если ты проводишь тестирование на столе, то при подъезде к его краю робот должен отъезжать назад и немного поворачиваться, после чего продолжать свою работу. Будь осторожен! Во избежание падения робота подстраховывай его руками!





## А теперь...

Надень на своего робота тряпочки для протирки, закрепив их, например, с помощью обычных канцелярских резинок.



Нанеси небольшое количество моющего средства или простой водопроводной воды и запусти робота по полу или своему столу (предварительно убрав с него все предметы).

Можешь отдохнуть и заниматься своими делами — робочист сделает всю грязную работу! Главное, не забывай вовремя менять накладки!

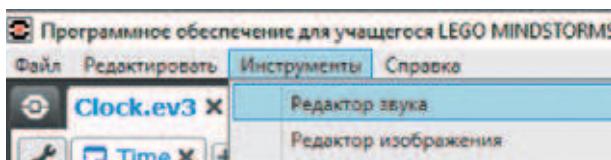
### Уборка с музыкой

Ты можешь добавить в программу своего робота воспроизведение музыки (или каких-то сигналов), воспользовавшись редактором звука, встроенным в программное обеспечение EV3.

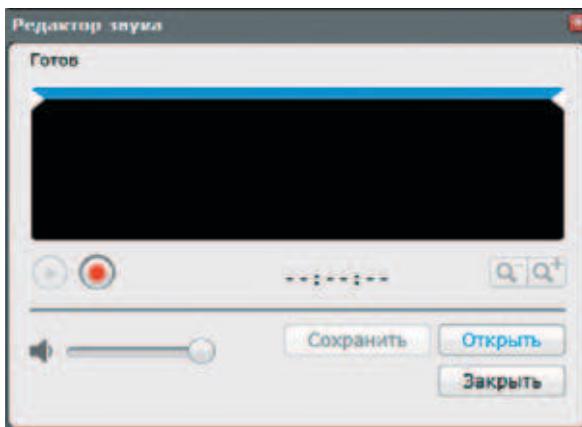
Давай посмотрим, как же научить твоего робота новому звучанию!

1. Снова открой свой проект **Robowash** в LME-EV3 (**Файл** → **Открыть проект** → **Открыть**).

2. Скачай либо запиши любую желаемую мелодию или звук и загрузи звуковой файл на компьютер.

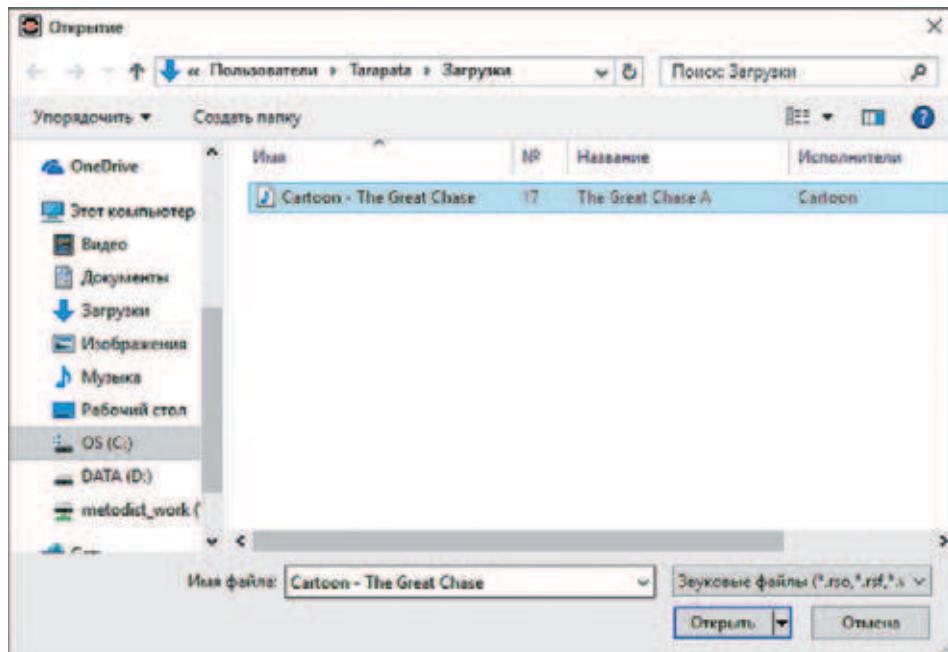


3. Открой сверху меню **Инструменты** → **Редактор звука**.

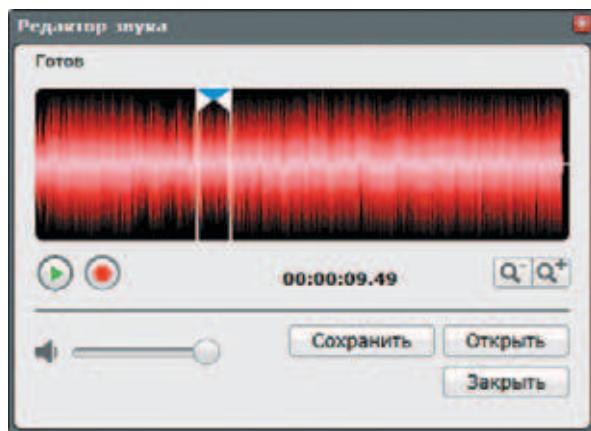


4. Нажми в открывшемся окне кнопку **Открыть**.

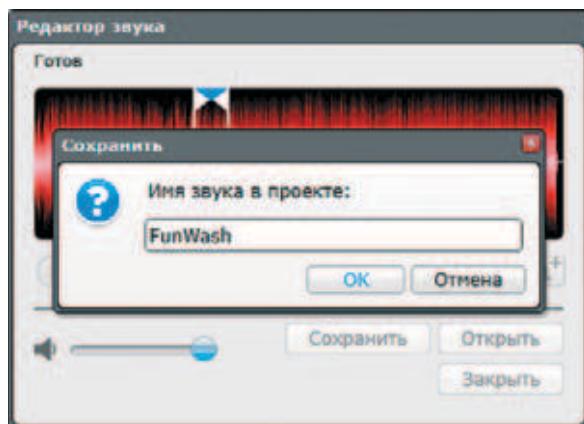
5. Найди в памяти компьютера сохранённый звуковой файл и кликни по нему дважды.



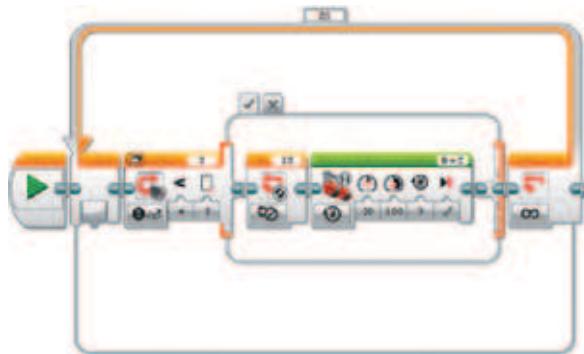
6. Звук откроется для предварительного редактирования. Так как ты ещё только начинающий инженер и твой конструктор тоже знакомит тебя с миром робототехники, длина звуковой дорожки ограничена 10 с. Поэтому тебе нужно будет выбрать понравившийся фрагмент и обрезать лишнее. Сделать это можно с помощью двух **бегунков сверху**: подвинь левый **бегунок вправо**, а правый **влево**.



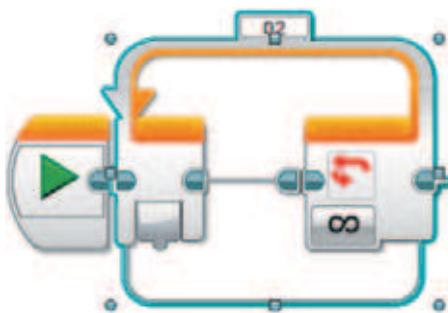
7. Далее можешь прослушать этот звук, сначала нажав кнопку с зелёным треугольником, а затем нажав кнопку **Сохранить**. В открывшемся окне задай имя, по которому данный звук будет идентифицироваться в твоём проекте. В нашем примере мы назовём его **FunWash** («Веселая мойка»). И нажми кнопку **OK**.  
Затем нажми кнопку **Закрыть**.



8. Теперь в программе тебе нужно создать ещё одну ветвь. Поэтому добавь ещё одну команду **Начало** (оранжевый блок).



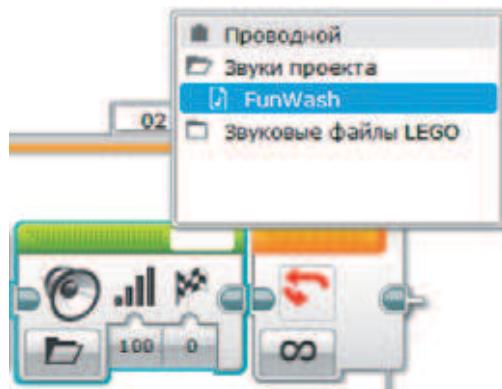
9. Далее добавь бесконечный **Цикл** (оранжевый блок).



10. Добавь внутри цикла команду **Звук** (зелёный блок), опцию **Воспроизвести файл** оставь по умолчанию.



11. Выбери в правом верхнем оконке папку **Звуки проекта** и в ней файл **FunWash**.

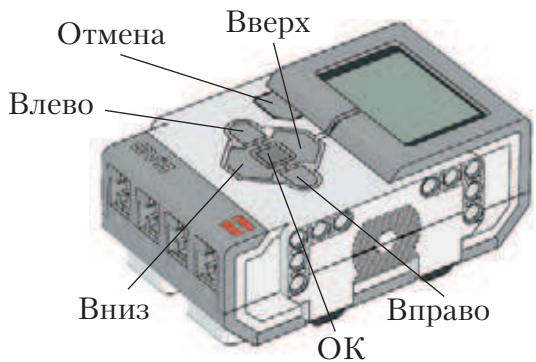


Готово! Теперь твой робот заиграет по-новому! Также ты можешь добавить ещё несколько различных звуковых файлов и наслаждаться многообразием звучаний!

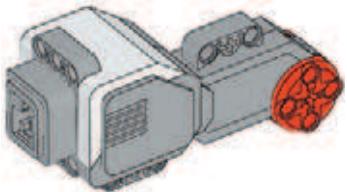
Интересно, а что получилось у тебя?

## Перечень деталей

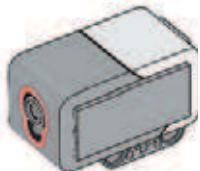
Модуль EV3, 1x



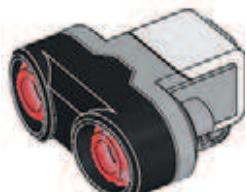
Большой мотор, 1x



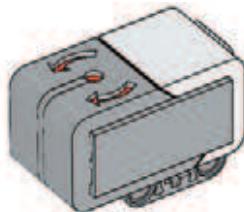
Датчик цвета, 1x



Ультразвуковой датчик, 1x



Гироскопический датчик, 1x



## Гусеничный узел

Цепное колесо, 40,7×15 мм, чёрное, 1x

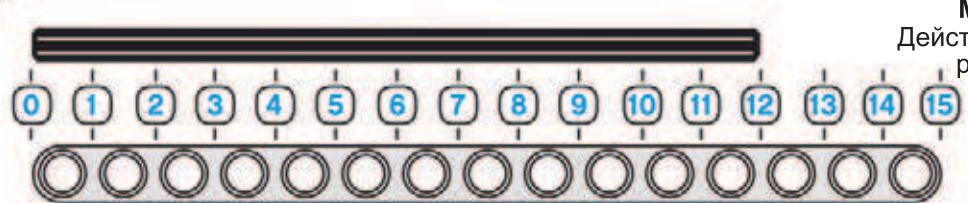


Звено гусеницы, чёрное, 25x



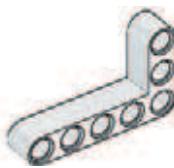
## Ось

Ось № 7, серая, 1x

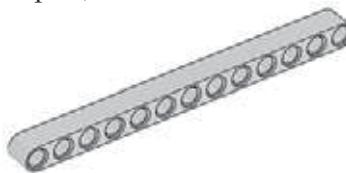


## **Балки**

Прямоугольная балка  $3 \times 5$ ,  
белая, 2x



Балка № 13, серая, 1x



Балка № 7, серая, 4x



## **Кабели**

Кабель, 25 см, 1x;  
Кабель, 35 см, 1x



## **Штифты**

Соединительный штифт,  
2-модульный, чёрный, 14x



Соединительный штифт  
со втулкой, 3-модульный,  
красный, 5x



Соединительный штифт,  
3-модульный, синий, 8x



## **Зубчатые колёса**

Зубчатое колесо,  
40 зубьев, серое, 1x

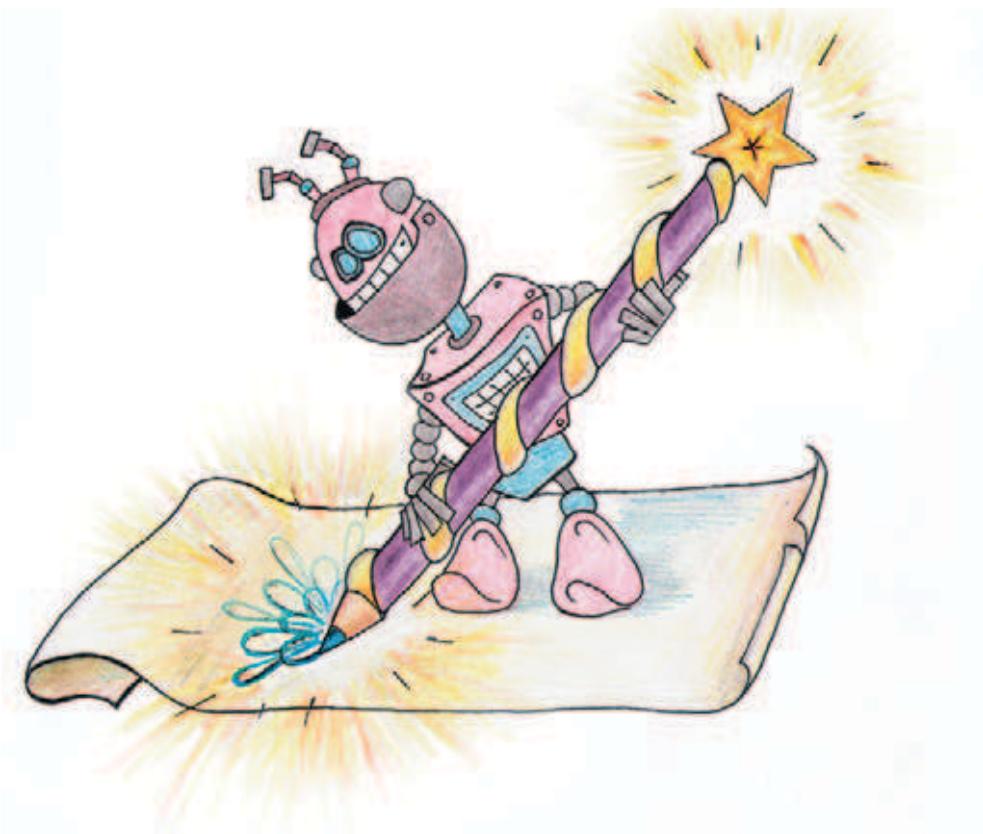


Двойное коническое зубчатое  
колесо, 36 зубьев,  
чёрное, 1x



**education**

# Волшебная палочка



# Графика в нашей жизни

Сегодня нас повсюду окружает компьютерная графика: рисунки на рекламных щитах и информационных листовках, иллюстрации в книгах, заставки в телепередачах, — во всех местах, куда достаёт взгляд. И всё многообразие изображений большинство людей привычно делит на два класса — выполненные рукой человека и компьютерные, хотя на самом деле компьютерная графика может очень сильно различаться в пределах своего класса.

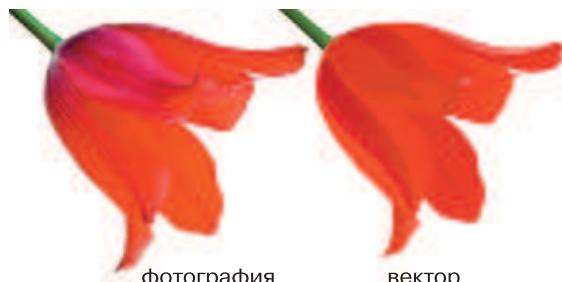
Существует три основных вида компьютерной графики: растровая, векторная и фрактальная. Основное их различие заключается в возможности увеличения без потери качества готового изображения при дальнейшей работе с ним.

Чаще всего мы имеем дело с **растровой графикой** — это фотографии или отсканированные из книги рисунки (рис. 1). Растровую картинку нельзя масштабировать без потери качества. Она построена из множества пикселей (точек) определённого цвета. При увеличении растровой картинки используются технологии, обеспечивающие так называемое *сглаживание* — автоматическую дорисовку отсутствующих точек, поэтому новое изображение получается размытым или, наоборот, обладает неестественными, рваными контурами.

**Векторная графика** (рис. 2) встречается на логотипах, эмблемах и прочих элементах, требующих хорошего качества при разных размерах. Векторная графика, как ясно из названия, использует векторы (отрезки,



**Рис. 1.** Растворная графика



**Рис. 2.** Сравнение растворной и векторной графики

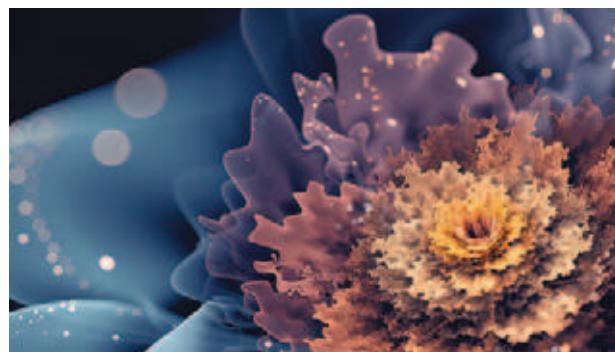


для которых указано направление). Концы отрезков задаются координатами точек. При увеличении вектора концы его отдаляются друг от друга, что превращает действие в чисто математическое вычисление.

**Фрактальная графика** — самоповторяющиеся изображения, выполненные с помощью вычислений (рис. 3). Для построения фрактала задаётся правило, согласно которому изображение строится из одних и тех же частей (геометрических фигур). У фрактальных изображений есть неоспоримое достоинство — они занимают малый объём памяти. В неё заносится описание всего лишь одного изображения, а новые структуры, получающиеся на каждом «новом витке», запоминать не требуется. К слову, витки эти задаются алгоритмами, которые обрабатывают данные с прошлыми параметрами, поэтому чаще всего построение идёт от большого к малому.

Естественно, что увеличивать изображение можно до бесконечности, при этом чёткость не будет теряться. Для построения фрактала можно использовать программы *Art Dabbler*, *Apophysis* или другие редакторы. Кстати, *Apophysis* находится в открытом доступе и для работы не требует установки на компьютер.

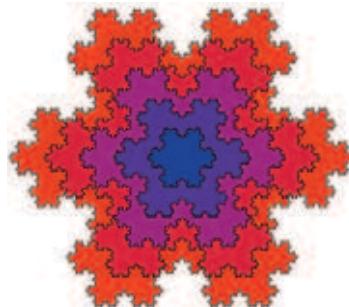
Фракталы встречаются не только на рисунках. Если приглядеться повнимательнее, их можно увидеть везде. Доказано, что в природе большинство узоров имеет фрактальную структуру. Возьмём, к примеру, цветок в горшке на подоконнике или дерево на улице (рис. 4). Они растут по определённым законам, которые можно выразить сложными формулами. Заметив это, математики принялись составлять формулы, описывающие природные фракталы. Так появились L-системы, деревья



**Рис. 3.** Фрактальный рисунок «Подводный» художника из Финляндии BoxTail



**Рис. 4.** Рост листа клёна можно представить как движение прожилок от основания к краям



**Рис. 5.** Снежинка Коха



**Рис. 6.** Самый популярный древнегреческий орнамент — меандр

правила — новые игры должны иметь высокую степень детализации. С каждым годом разрешение увеличивается, и совсем скоро виртуальную реальность трудно будет отличить от реалистичного изображения. Однако хранение текстур и форм требует много места: чем подробнее векторный рисунок, тем больше требуется памяти. Неудивительно, что от растровой графики в играх почти отказались, исключая 2D-квесты и поиск предметов, поскольку эти файлы занимают слишком значительный объём памяти. Современные видеокарты имеют достаточную вычислительную мощь, чтобы обрабатывать фрактальную графику в большом количестве. Холмы, горы, облака — всё это сделано в современных играх с помощью фракталов.

Термин «фрактал» был предложен французским математиком Бенуа Мандельбротом в 1975 году. Заметим, математика и раньше служила для составления рисунков. Стремясь украсить свое окружение, люди испокон веков использовали различные узоры. **Узор** — это рисунок, состоящий из геометрических фигур (например, линий, кругов или квадратов) разного цвета и размера. Обычно элементы в нём многократно повторяются (в фрактальном узоре минимальный размер элемента никак не ограничивается).

Повторяющиеся узоры называются **орнаментом**. Орнаментами украшаются здания, края тканей и многое другое. Идея повторяющихся по горизонтали или вертикали узоров настолько проста и одновременно удивительно красива, поэтому немудрено, что орнамент применяли с древних времен. Знание же геометрии позволяло совершенствовать узоры. Например, в Древней Греции в орнаментах использовалось чередование равных по размеру или форме элементов, чтобы создать ощущение движения узора (рис. 6). Орнамент не имел ни начала, ни конца — здания или кувшины покрывались непрерывным рисунком, означавшим круг бытия. Декоративные сюжетные элементы располагались вовсе не случайным образом, а в определённых местах, чтобы соблюсти принципы композиции.

Орнаменты можно было встретить не только в Европе. В Средние века в мусульманских странах это искусство получило повсеместное

Пифагора и другие математические сущности. А осенью жёлтые и красные пятна на всё тех же листьях подозрительно копируют снежинки Коха (рис. 5). Все упомянутые фракталы относятся к геометрическим фракталам, наиболее наглядно демонстрирующим самоподобие.

В 3D-моделировании фракталы используются для сокращения объёма памяти программного продукта. Время диктует свои



распространение и поражало сложностью и обилием форм. Причина крылась в религии: ислам накладывает запрет на изображение человека или иного существа с душой. Восточные художники, живущие среди пышной растительности у воды и привыкшие к ярким природным краскам, активно создавали прекрасные геометрические или растительные орнаменты. Их работы сильно отличались от произведений европейских декораторов, орнамент которых был однонаправленным (выполнялся либо по горизонтали, либо по вертикали). На востоке нашла распространение «ковровая» орнаментация, когда творцы полностью покрывали узором предмет, щедро вписывая в орнамент пословицы, афоризмы или строки из Корана.

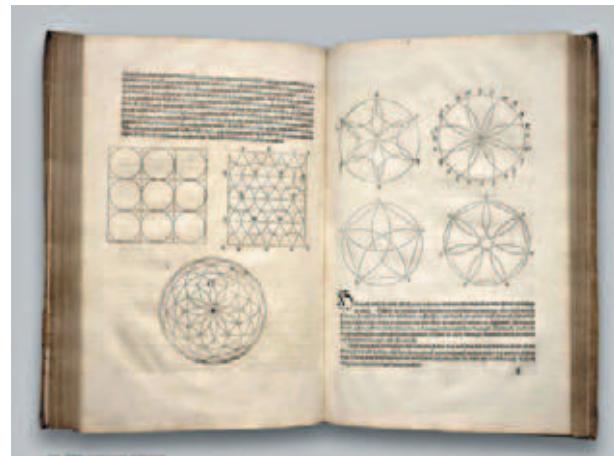
Геометрический исламский орнамент называется **гирих**, что в переводе с персидского означает «узел». В основе этого узора — геометрия круга, деление его на равные части; с использованием циркуля и линейки в дальнейшем строятся квадраты, прямоугольники, многоугольники. Чаще всего гирих встречается во внешнем оформлении мечетей или других важных зданий (рис. 7).

Домашнюю атмосферу на Востоке создавал орнамент под названием **ислими**, состоящий из стилизованного плюща и обиваемой им спирали. На изображении Ханского дворца ислими видно между окнами, которые выходят во внутренний дворик. Ислими покрывали также посуду, подносы и использовали при росписи книг.

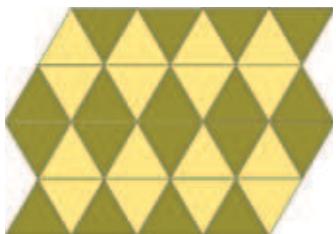
В конце XIII–XIV веков в Западной Европе, где в противовес лёгкости и яркости исламских узоров царил готический стиль, богатая знать стала украшать во дворцах полы рисунками из дощечек из различных пород дерева. Узоры также формировались в орнаменты, повторяющиеся, как и у мусульман, по горизонтали и вертикали. Однако европейцы это делали по-своему (рис. 8) используя **паркет (замощение, мозаика)** — раз-



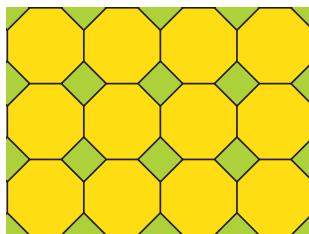
**Рис. 7.** Гирих на стенах Ханского дворца в Бахчисарае, Крым



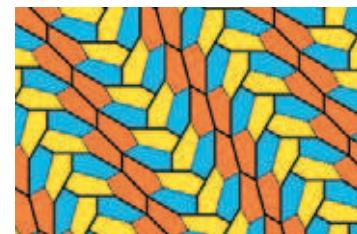
**Рис. 8.** Паркеты в книге «Underweysung der Messung», изданной в 1538 году. Автор книги — знаменитый немецкий художник Альберт Дюрер)



**Рис. 9.** Правильный паркет



**Рис. 10.** Полуправильный паркет



**Рис. 11.** Неправильный паркет

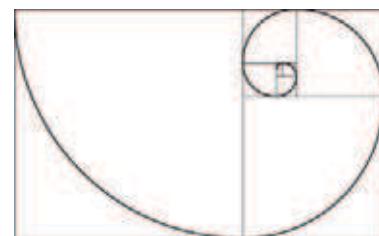
биение плоскости на фигуры без промежутков и перекрытий. Паркеты используют до сих пор, правда сегодня под этим словом чаще подразумевают деревянное покрытие, а не технику декоративного украшения.

Различают несколько видов паркетов. Если паркет полностью состоит из одинаковых правильных многоугольников (в правильной фигуре стороны и углы равны между собой) он называется *правильным* (рис. 9). Интересно, что правильными бывают только орнаменты, составленные из треугольников, квадратов или шестиугольников. В каждой вершине паркета должно сходиться целое число углов, сумма которых должна быть равна  $360^\circ$ . Если же выбраны правильные фигуры, сумма углов которых на вершине паркета даёт  $360^\circ$ , но при этом используются разные фигуры, паркет называется *полуправильным* (рис. 10). Этот вид используется чаще всего. *Неправильными* же называются паркеты, состоящие из неправильных фигур (в неправильной фигуре стороны и углы между собой не равны) — например, из стилизованных стрелок или звёзд (рис. 11). Иногда неправильные паркеты называют *картами*.

Ещё одним геометрическим способом заполнения плоскости, куда более близким к фракталам, чем орнаменты, является **спираль** — это кривая, которая огибает некоторую центральную точку или ось, постепенно приближаясь или удаляясь от неё (рис. 12). В зависимости от коэффициента кривая может быть похожа на круг или напоминать плавную закругляющуюся линию (рис. 13). Окружность, кстати, является частным случаем спирали, которая удаляется или приближается от точки (центра) всегда на одно и то же значение.

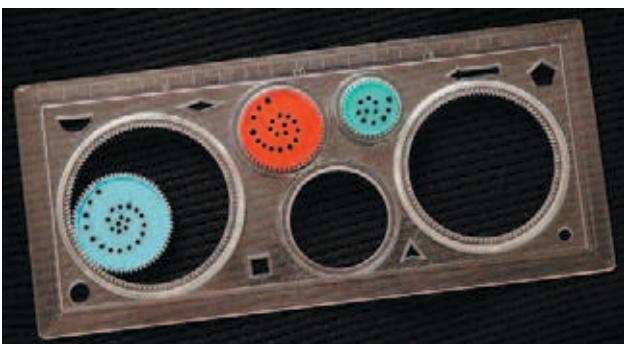


**Рис. 12.** Спираль

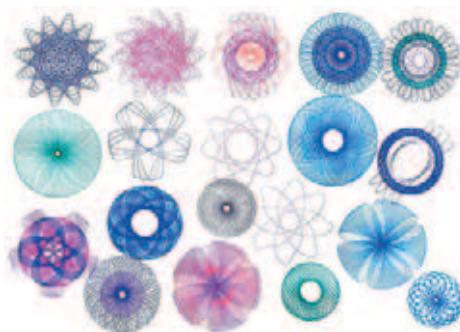


**Рис. 13.** Гиперболическая спираль





**Рис. 14.** Спирограф



**Рис. 15.** Узоры, полученные при помощи спирографа

В 1962 году британский инженер Денис Фишер, работая над чертежами взрывателей для авиабомб, задался вопросом: как быстро и качественно выводить равные элементы с плавно закругляющимися сторонами. Идея настолько сильно запала ему в душу, что некоторое время спустя он создал для этого специальный инструмент — **спирограф** (рис. 14). Наносить точные размеры и ускорить работу над серьёзным проектом с помощью изобретения не получилось, зато члены семьи Фишера были в восторге! Тогда инженер решил переформатировать своё изобретение в игрушку. Она состояла из пластмассовой пластины с вырезанными кругами разных диаметров и набором зубчатых колёс с отверстиями. Пластина прикладывалась к листу, в отверстие вставлялся грифель карандаша или стержень ручки, а затем колесо приводилось в движение. На бумаге появлялся спиральный след.

Покупатели оценили спирограф — это устройство было признано лучшей обучающей игрушкой в 1965 году и не сдавало позиций в этом рейтинге целых три года после первой продажи!

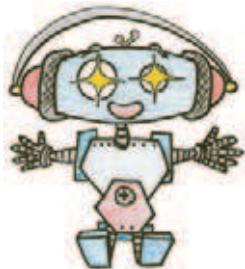
Спирограф может быть автоматическим. Специальная программа комбинирует различные коэффициенты, учитывающие размеры зубчатых колёс и число отверстий для пишущих элементов. Это даёт возможность создавать качественные и неповторимые узоры для оформления интерьеров. Особенно завораживающими получаются рисунки, содержащие одновременно орнаменты и спирали или даже фракталы (рис. 15).

Ты тоже можешь создать собственный автоматический спирограф, фантазийные возможности которого будут ограничены лишь написанной тобой программой! Так чего же ты ждёшь? Вперёд!

#### Кстати!

Автоматические спирографы используются также в медицине. В зависимости от силы дыхания человека у прибора изменяются параметры (коэффициенты спирографа), и это изменение наглядно представляется получающейся кривой.

## Этап 1. Устройство спирографа



Рассмотри модель спирографа, собранную на основе набора LEGO® MINDSTORMS® Education EV3.

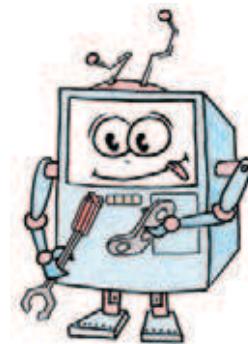
Попробуй выделить на ней рабочие детали — колёсная база, датчик касания и гироскопический датчик, крепление пера (маркер).

Попробуй собрать эту модель. Вот так выглядит робот-спирограф, который тебе предстоит собрать (рис. 16).



**Рис. 16.** Модель робота-спирографа

## Этап 2. Сборка робота-спирографа



### ШАГ 1. СБОРКА КОЛЁСНОЙ БАЗЫ

Детали для сборки:

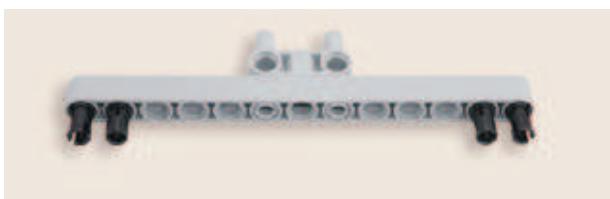
- большой мотор, 2x;
- низкопрофильная шина, чёрная, 2x;
- ступица, серая, 2x;
- втулка, 1-модульная, серая, 2x;
- стрелка, 3-модульная, белая, 2x;
- ось № 7, серая, 2x;
- двойная угловая балка, 3 × 7-модульная, белая, 2x;
- балка № 13, серая, 1x;
- соединительный штифт, 2-модульный, синий, 2x;
- Н-образный соединительный штифт, 3 × 3-модульный, серый, 1x;
- соединительный штифт, 2-модульный, чёрный, 6x;
- стальной шарик, 1x;
- шариковый подшипник, 1x.



1. Возьми балку № 13 и вставь чёрные штифты в первый, второй, двенадцатый и тринадцатый модули.



2. Вставь с обратной стороны балки в шестой и восьмой модули Н-образный штифт.



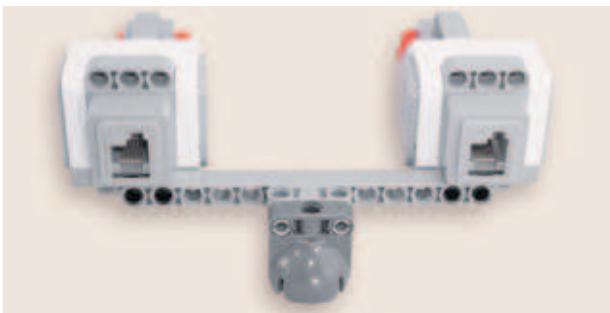
3. Возьми стальной шарик и вставь его в подшипник.



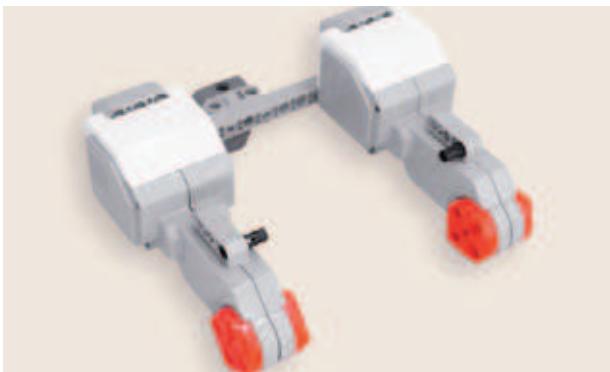
4. Установи собранное шаровое колесо на Н-образном штифте.



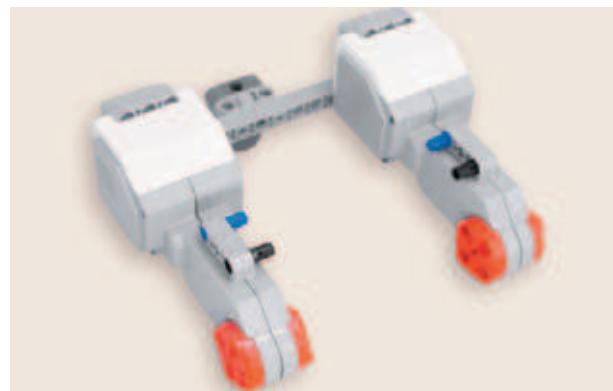
5. Возьми два больших мотора и соедини их между собой с помощью собранной детали, как показано на рисунке.



6. Разверни получившуюся конструкцию. С внутренней стороны моторов сверху вставь по одному чёрному штифту в ближние модули.



7. Аналогично установи через один модуль по одному синему штифту.



8. Далее надень на штифты по одной белой угловой балке  $3 \times 7$ , как показано на рисунке.



9. Возьми шины и надень их на ступицы.



10. Вставь в центральный модуль каждого колеса по одной оси № 7 так, чтобы они немного выступали с внешней стороны.



11. Надень на длинные концы осей по одной серой втулке.



12. Установи колеса на подвижной платформе. Для этого вставь оси в центральные части приводов моторов длинными концами вперёд, пока колёса не упрутся втулками.



13. Надень с внешней стороны на выступающие оси по одной белой стрелке.



Основа для спирографа готова! Осталось сделать всего пару шагов...

## ШАГ 2. УСТАНОВКА ПРОГРАММИРУЕМОГО МОДУЛЯ



### Детали для сборки:

- программируемый модуль, EV3, 1x;
- соединительный штифт, 2-модульный, чёрный, 6x.



1. Вставь с двух сторон программируемого модуля по три чёрных штифта в верхние модули.



2. Установи программируемый модуль на колёсную базу. Для этого закрепи его между двумя двойными угловыми балками, вставив выступающие чёрные штифты в их пятый, шестой и седьмой модули сверху.



На этом сборка основной части спирографа завершена, осталось добавить пусковой механизм. Так держать!

### ШАГ 3. ЗАКРЕПЛЕНИЕ ПУСКОВОЙ КНОПКИ, ГИРОСКОПА И ПОДКЛЮЧЕНИЕ МОТОРОВ И ДАТЧИКОВ



#### Детали для сборки:

- соединительный штифт, 2-модульный, синий, 2x;
- Н-образный соединительный штифт, 2x;
- гироскопический датчик, 1x;
- датчик касания, 1x;
- кабель, 25 см, 4x.



1. Возьми Н-образный штифт и установи его с левой стороны программируемого модуля, как показано на рисунке.



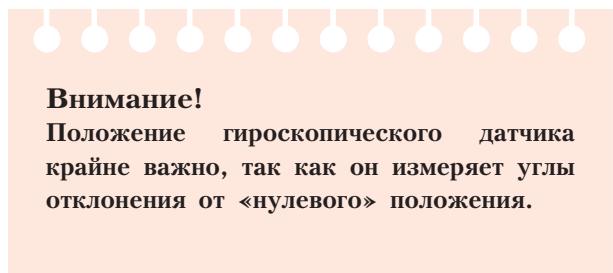
2. Вставь в его центральный модуль синий штифт.



- Закрепи на собранной конструкции датчик касания.



- Аналогичным образом закрепи гироскопический датчик с правой стороны программируемого модуля.



- Соедини с помощью четырёх кабелей:
  - левый большой мотор в порт А;
  - правый большой мотор в порт В;
  - гироскопический датчик в порт 1;
  - датчик касания в порт 2.





## Этап 3. Установка программного обеспечения на компьютере



1. Если ты приобрёл базовый набор LEGO® MIND-STORMS® Education EV3 (LME-EV3) с лицензией на программное обеспечение LME-EV3, то действуй так, как написано в информационном листке, вложенном в набор.
2. Если такой лицензии у тебя нет, то зайди на сайт <http://Education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.

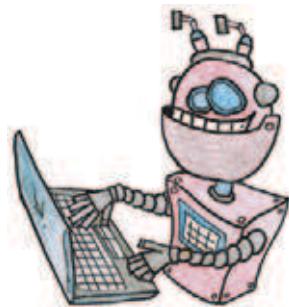


### Внимание!

При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу <http://Education.LEGO.com>.



# Этап 4. Создание программы для робота-спирографа



## ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

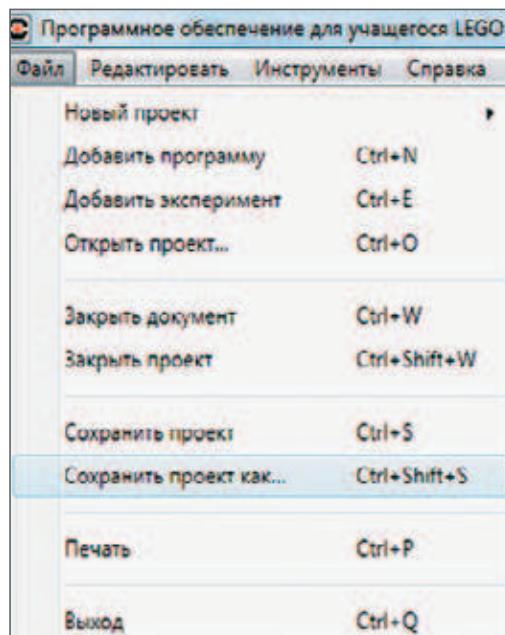
1. Запусти программное обеспечение LME-EV3.
2. Выбери в открывшемся окне слева пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

## СОЗДАНИЕ НОВОГО ПРОЕКТА В ПАМЯТИ EV3

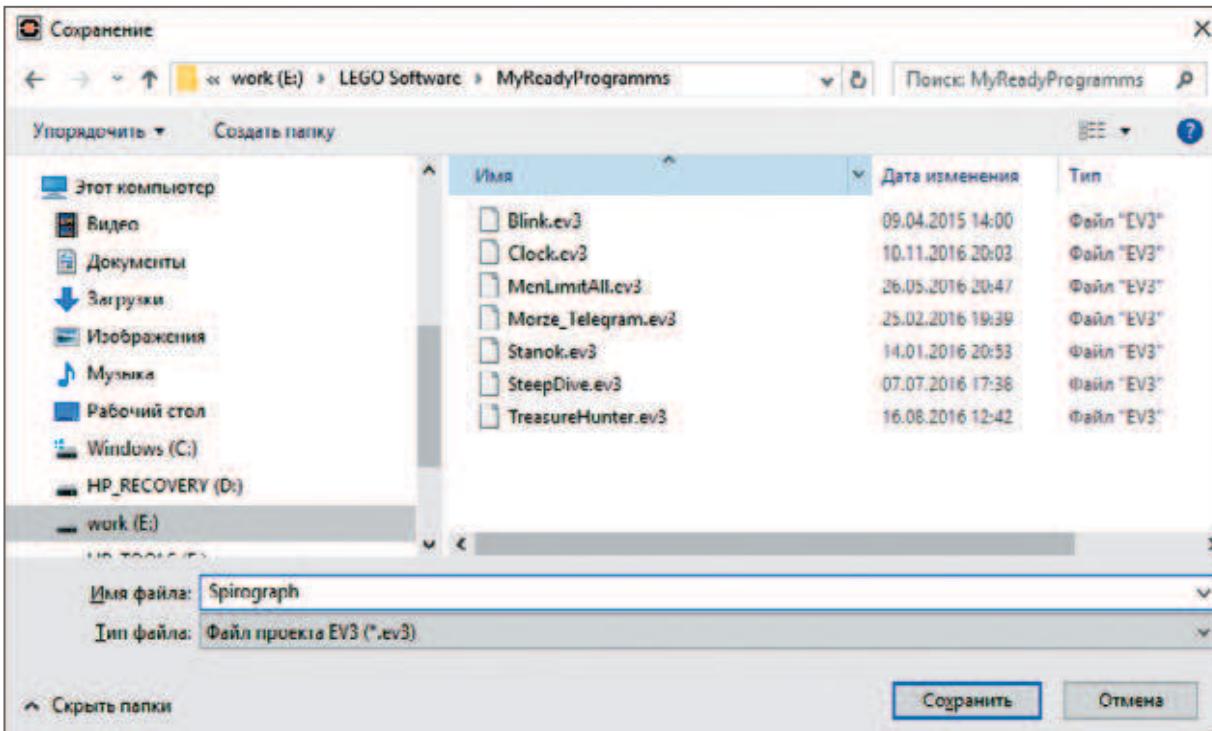
Программируемый модуль EV3 позволяет хранить в своей памяти десятки проектов. Это очень удобно: если захотел вернуться к какому-то проекту, не нужно искать его в компьютере, он всегда находится внутри робота.

Однако необходимо грамотно организовать хранение проектов. Папка с проектом должна иметь название, отражающее его суть, иначе тебе будет трудно её найти. Если в папке с проектом несколько программ, то назови их максимально понятно для себя.

1. Для начала сохрани свой проект в памяти компьютера, чтобы избежать его потери. Для этого выбери меню **Файл** в левом верхнем углу окна программы и в контекстном меню выбери пункт **Сохранить проект как...**



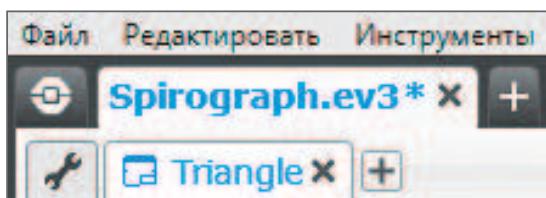
- Выбери в открывшемся окне удобное расположение для всех файлов твоей программы, затем задай **Имя файла**: *Spirograph* (в переводе с английского «Спирограф») в нижней части окна и нажми **Сохранить**.



#### Внимание!

Настоящие программисты всегда дают имена своим программам на английском языке и в названии не используют пробелы. Вместо них они пишут, например, каждое слово с большой буквы или добавляют нижнее подчеркивание.

Именно так будет называться в памяти программируемого модуля EV3 папка для хранения программ, которые ты напишешь в рамках этого проекта.



- Осталось дать имя самой программе (в нашем проекте их будет три). Для этого дважды кликни по имени *Program* в левом верхнем углу поля программы, и задай новое имя: *Triangle* (в переводе с английского «Треугольник») и нажми клавишу **Enter**.



## ЛОГИКА ПРОГРАММЫ

Исходя из названия в нашей первой программе за основу узора мы берём треугольник. Однако наш робот будет рисовать его, каждый раз немного смещаясь. Так что, строго говоря, именно треугольник мы не получим, но некоторую его часть вполне можно будет узнать.

После запуска программы робот будет ожидать нажатия датчика касания. Это нужно для того, чтобы робот не сдвинулся от своего начального положения при запуске. После этого роботу нужно выполнять всего лишь три действия: проезжать некоторое расстояние вперёд, разворачиваться (но не полностью!) и проезжать вперёд ещё раз точно такое же расстояние, как в начале. Таким образом, мы будем каждый раз получать «кривой» треугольник, а точнее две его стороны и немного закруглённую вершину. В результате повтора этих трёх действий будет получаться спирографический узор. В нашем случае получилось 14 повторов.

Приступим.

### Кстати!

В данной программе задействовано несколько интересных алгоритмических конструкций и приёмов программирования. Одни из них – это цикл с заданным числом повторений, а также использование датчика касания в качестве пускового механизма для основной программы.

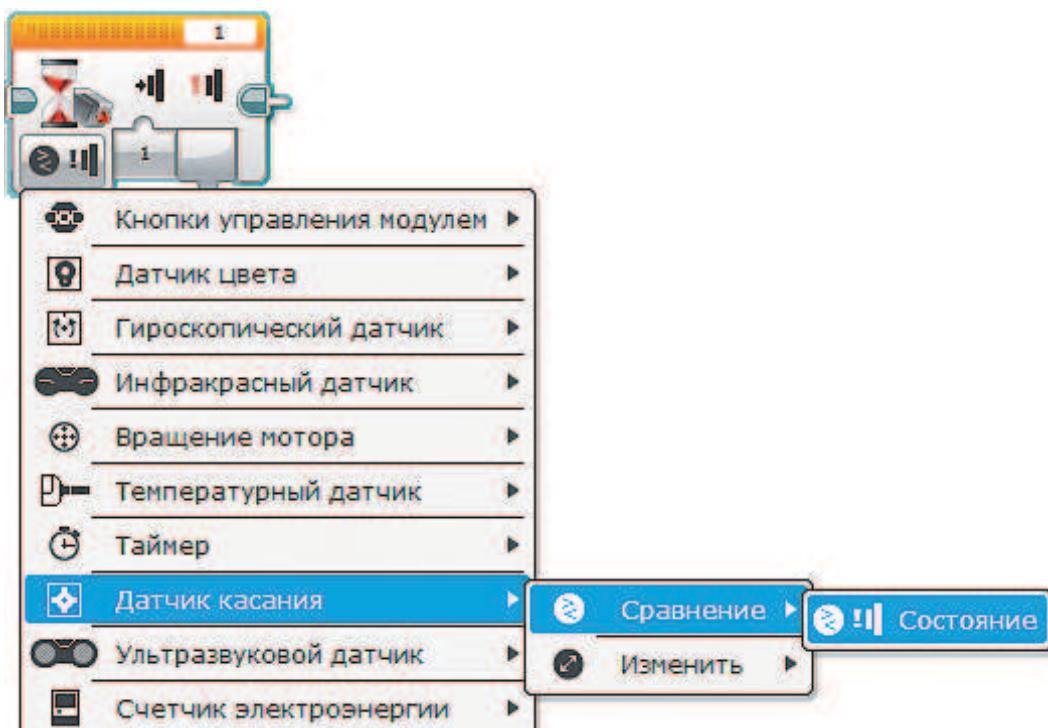
## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОТА-СПИРОГРАФА

В открывшемся окне проекта начинай составлять программу для робота-спирографа.

1. Для начала добавь в программу команду **Ожидание** (оранжевый блок).



2. Настрой опции этой команды: **Датчик касания** → **Сравнение** → **Состояние**. Параметр состояния оставь равным **1 (нажатие)**.

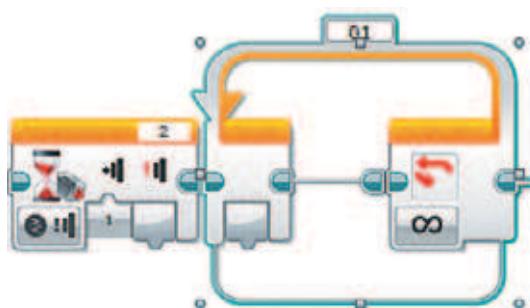


#### Внимание!

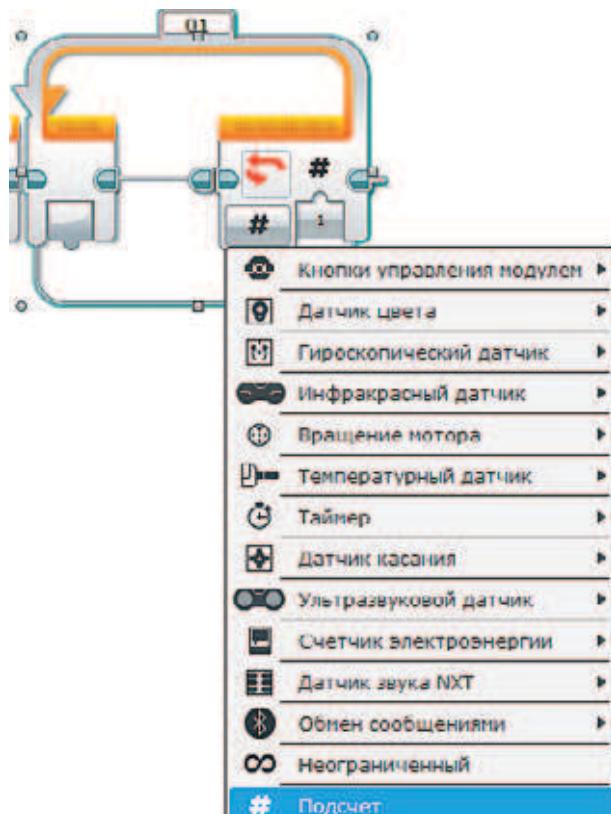
Проверь, чтобы в правом верхнем окошке значился правильный порт. Мы подключили датчик касания к порту 2.

Таким образом, программа будет ожидать нажатия датчика касания и только после этого продолжится.

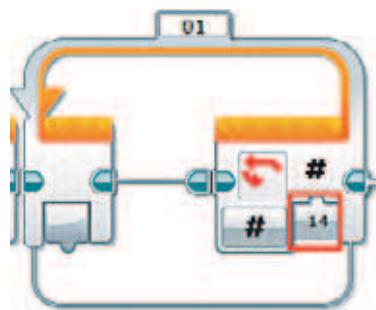
3. Далее добавь команду **Цикл** (оранжевый блок).



4. Выбери в нём опцию **Подсчёт**, так как мы знаем фиксированное число повторений.



5. Введи в параметре цикла с клавиатуры число **14** — как говорилось ранее, именно столько повторений понадобилось нам для того, чтобы узор оказался законченным. Это число у тебя может быть и другим (как и все остальные параметры), потому что на работу робота-спирографа сильно влияет, прежде всего, поверхность, на которой он рисует.
6. Далее внутри цикла добавь сразу три команды **Независимое управление моторами** (зелёный блок).

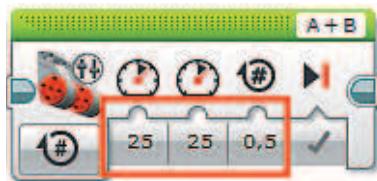


#### Внимание!

Проверь, чтобы в правом верхнем окошке каждой из этих команд значились порты А+В — именно к ним подключены моторы.

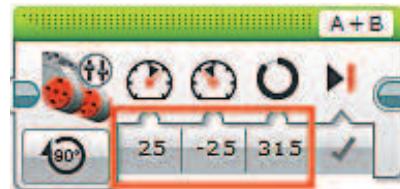


Осталось самое главное — настроить каждую команду.



7. Для первой команды:  
**Опция:** Включить на количество оборотов (по умолчанию);  
**Мощность на левый мотор:** 25;  
**Мощность на правый мотор:** 25;  
**Обороты:** 0,5.

8. Для второй команды:  
**Опция:** Включить на количество градусов;  
**Мощность на левый мотор:** 25;  
**Мощность на правый мотор:** -25;  
**Градусы:** 315.



9. Для третьей команды:  
**Опция:** Включить на количество оборотов (по умолчанию);  
**Мощность на левый мотор:** 25;  
**Мощность на правый мотор:** 25;  
**Обороты:** 0,5.

Молодец! Ты запрограммировал робота-спирографа! Теперь он готов изобразить «треугольный» узор!

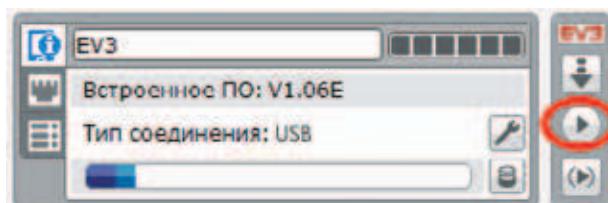
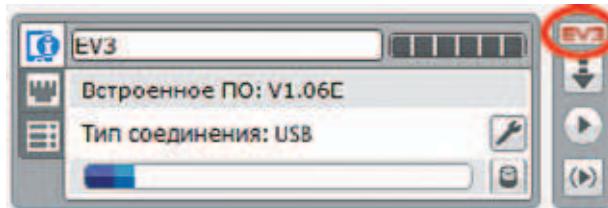


# Этап 5. Загрузка программы и её тестирование

## ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

Загрузи свою программу.

1. Подключи программируемый модуль с помощью USB-кабеля к компьютеру (соединив порты PC на модуле и USB на компьютере). В окне программного обеспечения станет активен блок EV3.
2. Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



## ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. После запуска программы робот не должен предпринимать каких-либо действий до того, как ты нажмёшь на датчик касания.
2. Поставь робота на ровную поверхность и нажми на датчик касания.
3. Робот должен проехать вперед по прямой короткое расстояние, затем развернуться (но не полностью!) и снова проехать короткое расстояние. Эти три действия должны повториться 14 раз.

### Внимание!

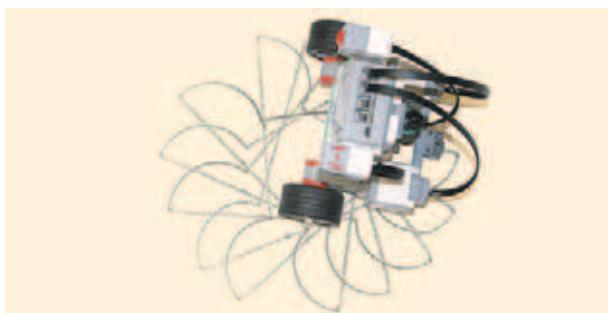
Помни, что наша программа – это конечный цикл. Остановить досрочно её можно только вручную с помощью кнопки Отмена на программируемом модуле. Если ты хочешь повторно запустить программу или продемонстрировать работу своего робота другу, то для его запуска компьютер тебе уже не нужен: просто включи программируемый блок, затем выбери папку Spirograph и в ней программу Triangle.

А теперь...

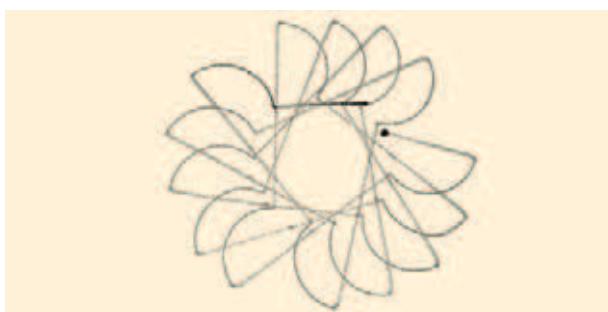


### «Треугольный» цветок

Прикрепи к своему роботу маркер (или любую другую пишущую принадлежность). Сделать это можно, например, с помощью канцелярских резинок.



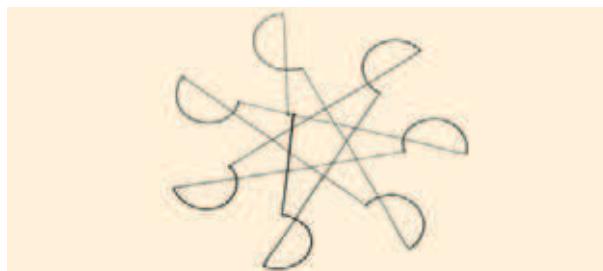
Теперь запусти робота на бумаге! Ну или на другой поверхности, которую не жалко 😊



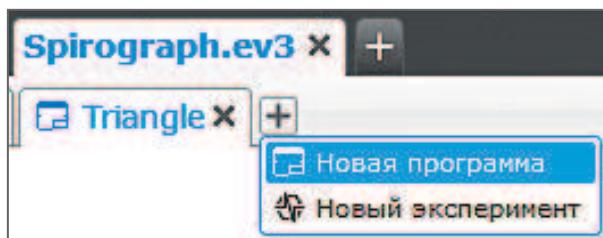
У нас он нарисовал «треугольный» цветок!

## Снежинка

Нам тоже понравилось программировать робота-спирографа на различные узоры! У нас получилась ещё вот такая снежинка!

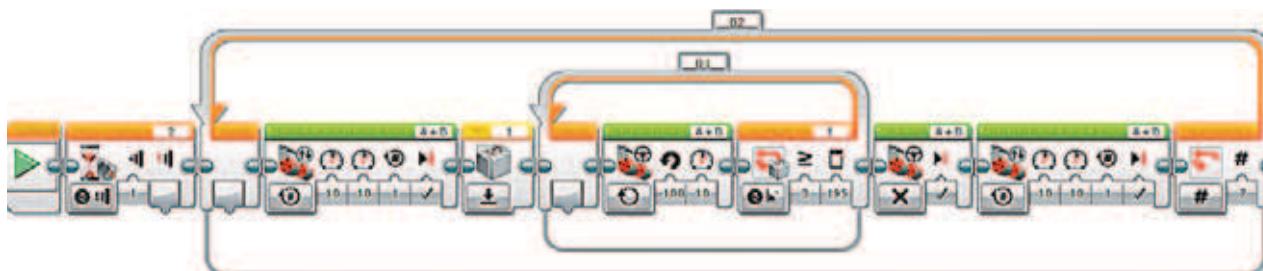


Попробуй и ты! Открой свой проект **Spirograph** и добавь новую программу. Для этого нажми рядом с первой программой **Triangle** на значок «+» и выбери пункт **Новая программа**.



Можешь назвать эту программу *Snowflake* (в переводе с английского «Снежинка»), как делал это и раньше: кликни дважды по имени *Program* и введи с клавиатуры новое название.

Вот общий вид программы:



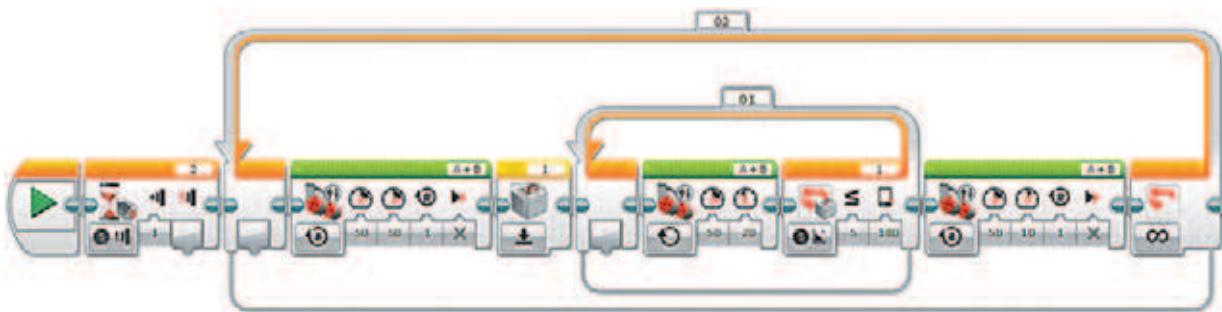
**Логика программы Snowflake.** После нажатия датчика касания программа входит в цикл, повторяющийся 7 раз. В этом цикле робот проезжает некоторое расстояние по прямой (один оборот колёс), после чего происходит сброс гироскопического датчика (для исключения ошибки) и начинается внутренний цикл, делящийся до тех пор, пока робот не повернётся на 195 градусов согласно показаниям гироскопа. В каждом наборе гироскопический датчик работает по-разному — иногда его положительные показания соответствуют повороту по часовой стрелке, а иногда — против часовой. Поэтому и настройка внутреннего цикла в такой программе может меняться: либо  $\geq 195$ , либо  $\leq -195$ . На это также влияет, в какую сторону робот поворачивает при работе. Внутри этого цикла, естественно, содержится команда поворота робота против часовой стрелки. Затем моторы останавливаются, и робот вновь проезжает некоторое расстояние вперёд по прямой.

## Замысловатый узор

Как ты видел во вводной части, с помощью спирографа можно изобразить самые замысловатые узоры! Вот и у нас получился вот такой «цветочек»!

Проверь, нарисует ли твой робот так же. Открой свой проект **Spirograph** и добавь новую программу. Можешь назвать её *Flower* (в переводе с английского «Цветок»).

Вот общий вид программы:



Логика программы *Flower* аналогична программе *Snowflake*, изменены лишь параметры внутри команд движения робота, а цикл включен в бесконечный режим, поэтому робота ты сможешь остановить только нажатием кнопки **Отмена** на программируемом модуле.

Попробуй разобраться и поэкспериментировать!



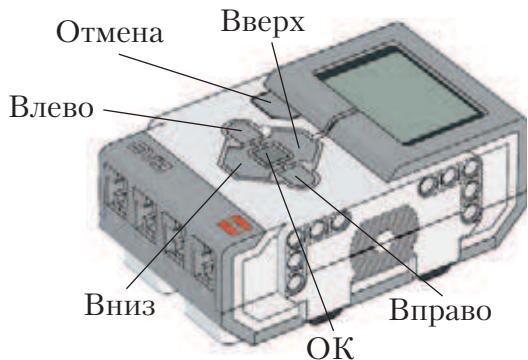
## Воображение!

Ну а теперь настало время включать воображение! Ты можешь составлять самые различные программы и по твоей указке робот-спирограф будет изображать новые красивые узоры! Быть может, он украсит твою комнату?

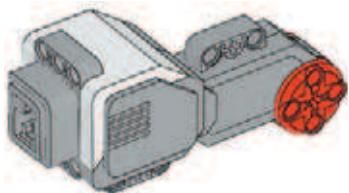
Используй маркеры разного цвета, можешь даже прикрепить к роботу сразу несколько фломастеров! Экспериментируй и становись настоящим программистом искусства!

## Перечень деталей

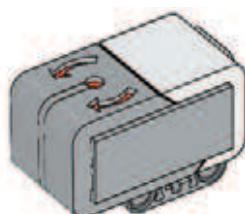
Модуль EV3, 1x



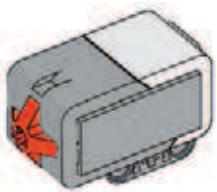
Большой мотор, 2x



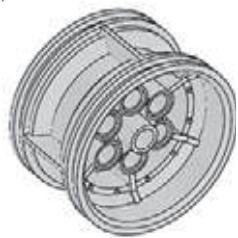
Гирокомпьютер, 1x



Датчик касания, 2x



Ступица, 43,2×26 мм, серая, 2x

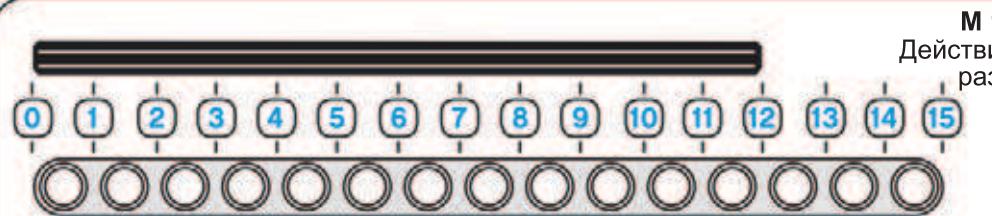


Низкопрофильная шина, 56×28 мм, чёрная, 2x



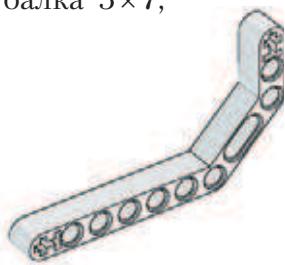
## Кабель

Кабель, 25 см, 4x

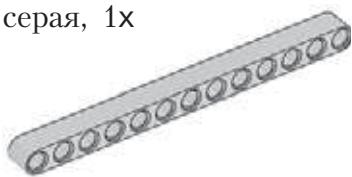


## Балки

Двойная угловая балка 3×7,  
белая, 2x



Балка № 13, серая, 1x



## Штифты

Соединительный штифт,  
2-модульный, чёрный, 12x



Соединительный штифт,  
2-модульный, синий, 4x



Н-образный соединительный  
штифт, серый, 3x



Шариковый подшипник,  
серый, 1x



Стальной шарик,  
серебристый металлик,  
1x



Стрелка, 3-модульная,  
белая, 2x

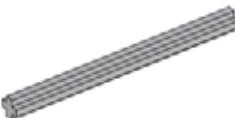


Втулка, 1-модульная,  
серая, 2x



## Ось

Ось № 7, серая, 2x



# Содержание

Дорогой друг! . . . . .	3
<b>СТОРОЖЕВАЯ БАШНЯ . . . . .</b>	<b>5</b>
Секреты сторожевых башен . . . . .	6
Этап 1. Устройство сторожевой башни. . . . .	10
Этап 2. Сборка системы охраны периметра . . . . .	11
Шаг 1. Сборка башни с приводом радара . . . . .	11
Шаг 2. Сборка пушки . . . . .	24
Шаг 3. Сборка орудия . . . . .	29
Шаг 4. Сборка локатора . . . . .	42
Шаг 5. Объединение компонентов системы . . . . .	49
Этап 3. Установка программного обеспечения на компьютере . . . . .	53
Этап 4. Создание программы для сторожевой башни . . . . .	54
Запуск программного обеспечения LME-EV3 . . . . .	54
Создание нового проекта в памяти EV3 . . . . .	54
Логика программы . . . . .	55
Составление программы для сторожевой башни . . . . .	57
Этап 5. Загрузка программы и её тестирование . . . . .	68
Шаг 1. Загрузка программы в программируемый модуль . . . . .	68
Шаг 2. Тестирование . . . . .	68
А теперь... . . . . .	69
Перечень деталей . . . . .	71
<b>ЧЕЛОВЕК – ВСЕМУ МЕРА? . . . . .</b>	<b>77</b>
История мер длины . . . . .	78
Старинные меры длины на Руси. . . . .	79
Этап 1. Устройство робота-измерителя . . . . .	81
Этап 2. Сборка робота-измерителя . . . . .	82
Шаг 1. Сборка основания для крепления мотора . . . . .	82
Шаг 2. Сборка крепления измерительного колеса . . . . .	84
Шаг 3. Установка мотора . . . . .	88
Шаг 4. Сборка и установка измерительного колеса . . . . .	88
Шаг 5. Подключение измерительной части к программируемому модулю. . . . .	90

<b>Этап 3. Установка программного обеспечения на компьютере . . . . .</b>	<b>91</b>
<b>Этап 4. Создание программы для робота-измерителя . . . . .</b>	<b>92</b>
Запуск программного обеспечения LME-EV3 . . . . .	92
Создание нового проекта в памяти EV3 . . . . .	92
Логика программы . . . . .	94
Составление программы для робота-измерителя . . . . .	95
<b>Этап 5. Загрузка программы и её тестирование . . . . .</b>	<b>100</b>
Загрузка программы в программируемый модуль. . . . .	100
Тестирование. . . . .	100
<b>Этап 6. Человеческий фактор исключён! . . . . .</b>	<b>101</b>
А теперь... . . . . .	103
Перечень деталей . . . . .	104
 <b>ПОСТОРОННИМ ВХОД ВОСПРЕЩЁН! . . . . .</b>	<b>107</b>
 <b>История развития систем контроля доступа . . . . .</b>	<b>108</b>
<b>Этап 1. Принцип работы системы контроля доступа. . . . .</b>	<b>110</b>
<b>Этап 2. Сборка системы контроля доступа . . . . .</b>	<b>111</b>
Шаг 1. Установка Н-образного штифта и подключение кабеля к модулю . . . . .	111
Шаг 2. Установка системы крепления УЗ-датчика на модуле. . . . .	112
Шаг 3. Сборка УЗ-датчика . . . . .	112
Шаг 4. Подготовка системы крепления УЗ-датчика . . . . .	113
Шаг 5. Сборка системы крепления датчика . . . . .	114
Шаг 6. Присоединение датчика к модулю. . . . .	114
<b>Этап 3. Установка программного обеспечения на компьютере . . . . .</b>	<b>115</b>
<b>Этап 4. Создание программы для системы контроля доступа . . . . .</b>	<b>116</b>
Запуск программного обеспечения LME-EV3 . . . . .	116
Составление программы для СКД. . . . .	116
Логика программы . . . . .	116
<b>Этап 5. Загрузка программы и её тестирование . . . . .</b>	<b>126</b>
Загрузка программы в программируемый модуль. . . . .	126
Тестирование. . . . .	126
<b>Этап 6. «Посторонним вход воспрещён!» . . . . .</b>	<b>127</b>
А теперь... . . . . .	128
Перечень деталей . . . . .	129
 <b>ТАЙНЫЙ КОД СЭМЮЭЛА МОРЗЕ . . . . .</b>	<b>131</b>
 <b>История телеграфа . . . . .</b>	<b>132</b>
<b>Этап 1. Изучаем способ кодирования, предложенный Морзе . . . . .</b>	<b>135</b>
<b>Этап 2. Сборка кнопочного звукового передатчика . . . . .</b>	<b>138</b>

Шаг 1. Сборка подставки под программируемый модуль . . . . .	138
Шаг 2. Сборка и крепление кнопки телеграфного ключа . . . . .	143
Шаг 3. Соединение кнопки с программируемым модулем . . . . .	144
<b>Этап 3. Установка программного обеспечения на компьютере . . . . .</b>	<b>145</b>
<b>Этап 4. Создание программы для передатчика . . . . .</b>	<b>146</b>
Запуск программного обеспечения LME-EV3 . . . . .	146
Логика программы . . . . .	146
Общий вид программы . . . . .	146
<b>Этап 5. Загрузка программы и её тестирование . . . . .</b>	<b>150</b>
Загрузка программы . . . . .	150
Тестирование программы . . . . .	150
<b>Этап 6. Поговорим на языке азбуки Морзе . . . . .</b>	<b>151</b>
Начинай играть . . . . .	151
<b>Этап 7. Отправляем телеграмму . . . . .</b>	<b>153</b>
<b>Этап 8. Кнопочный текстовый шифратор . . . . .</b>	<b>154</b>
<b>Этап 9. Создание программы для шифратора . . . . .</b>	<b>155</b>
Запуск программного обеспечения LME-EV3 . . . . .	155
Логика программы . . . . .	155
Составление программы для шифратора . . . . .	156
<b>Этап 10. Загрузка программы и её тестирование . . . . .</b>	<b>166</b>
Загрузка программы в программируемый модуль . . . . .	166
Тестирование . . . . .	166
<b>А теперь . . . . .</b>	<b>167</b>
<b>Перечень деталей . . . . .</b>	<b>168</b>
<b>РОБОЧИСТ СПЕШИТ НА ПОМОЩЬ! . . . . .</b>	<b>171</b>
<b>История гигиены помещений . . . . .</b>	<b>172</b>
<b>Этап 1. Устройство робочиста . . . . .</b>	<b>178</b>
Шаг 1. Сборка приводов для робочиста . . . . .	179
Шаг 2. Сборка уборочных валиков для робочиста . . . . .	181
Шаг 3. Сборка крепления и установка модуля EV3 на робочиста . . . . .	184
Шаг 4. Установка датчиков на робочиста . . . . .	186
Шаг 5. Подключение проводов . . . . .	189
<b>Этап 2. Установка программного обеспечения на компьютере . . . . .</b>	<b>190</b>
<b>Этап 3. Создание программы для робочиста . . . . .</b>	<b>191</b>
Запуск программного обеспечения . . . . .	191
Создание нового проекта в памяти EV3 . . . . .	191
Логика программы . . . . .	193
Составление программы для робочиста . . . . .	193

<b>Этап 4. Загрузка программы и её тестирование . . . . .</b>	<b>209</b>
Загрузка программы в программируемый модуль . . . . .	209
Тестирование . . . . .	209
<b>А теперь... . . . . .</b>	<b>211</b>
<b>Перечень деталей . . . . .</b>	<b>215</b>
<b>ВОЛШЕБНАЯ ПАЛОЧКА . . . . .</b>	<b>217</b>
<b>Графика в нашей жизни . . . . .</b>	<b>218</b>
<b>Этап 1. Устройство спирографа . . . . .</b>	<b>224</b>
<b>Этап 2. Сборка робота-спирографа . . . . .</b>	<b>225</b>
Шаг 1. Сборка колёсной базы . . . . .	225
Шаг 2. Установка программируемого модуля . . . . .	229
Шаг 3. Закрепление пусковой кнопки, гироскопа и подключение моторов и датчиков . . . . .	230
<b>Этап 3. Установка программного обеспечения на компьютере . . .</b>	<b>232</b>
<b>Этап 4. Создание программы для робота-спирографа . . . . .</b>	<b>233</b>
Запуск программного обеспечения . . . . .	233
Создание нового проекта в памяти EV3 . . . . .	233
Логика программы . . . . .	235
Составление программы для робота-спирографа . . . . .	235
<b>Этап 5. Загрузка программы и её тестирование . . . . .</b>	<b>239</b>
Загрузка программы в программируемый модуль . . . . .	239
Тестирование программы . . . . .	239
<b>А теперь... . . . . .</b>	<b>240</b>
<b>Перечень деталей . . . . .</b>	<b>243</b>

*Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"*

*Учебное электронное издание*

*Серия: «РОБОФИШКИ»*

**КОНСТРУИРУЕМ РОБОТОВ НА LEGO® MINDSTORMS® EDUCATION EV3.  
СБОРНИК ПРОЕКТОВ № 1**

*Составитель Ю. А. Серова*

*Для детей среднего школьного возраста*

*Ведущий редактор Ю. А. Серова*

*Художники В. А. Прокудин, Я. В. Соловцова, И. Е. Марев, Ю. Н. Елисеев*

*Фотосъемка: И. А. Федягин*

*Технический редактор Т. Ю. Федорова*

*Корректор И. Н. Панкова*

*Компьютерная верстка: Е. Г. Ивлева*

*Подписано к использованию 26.05.19.*

*Формат 210×260 мм*

*Издательство «Лаборатория знаний»  
125167, Москва, проезд Аэропорта, д. 3*

*Телефон: (499) 157-5272*

*e-mail: [info@pilotLZ.ru](mailto:info@pilotLZ.ru), <http://www.pilotLZ.ru>*