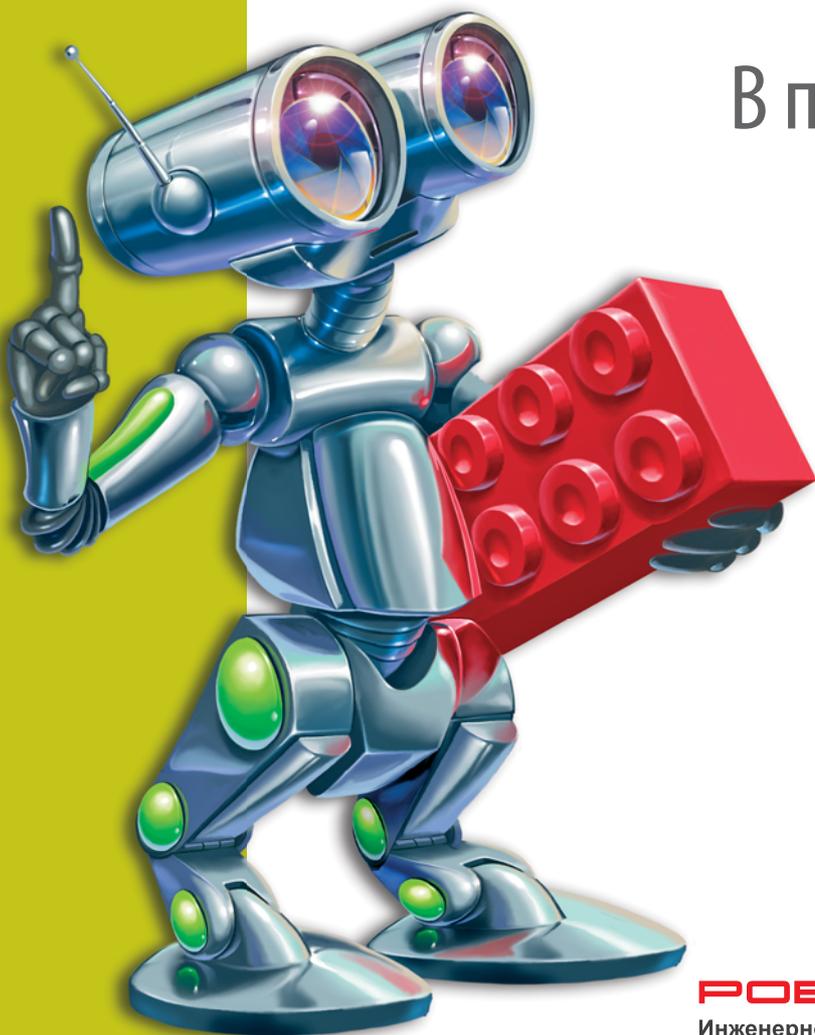


Р • О • Б • О • Ф • И • Ш • К • И

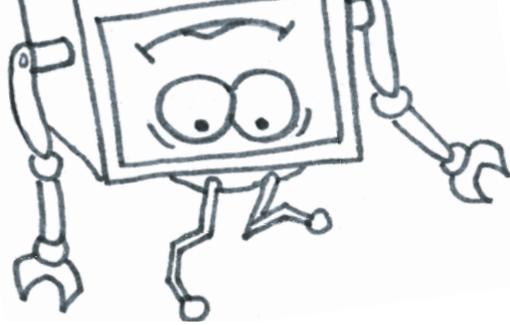
# КОНСТРУИРУЕМ РОБОТОВ

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

В поисках сокровищ



**РОБОТОТЕХНИКА**  
Инженерно-технические кадры инновационной России



Е. И. Рыжая, В. В. Удалов

# КОНСТРУИРУЕМ РОБОТОВ

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

В поисках сокровищ

Электронное  
издание



Лаборатория знаний  
Москва  
2017

УДК 373.167  
ББК 32.97  
Р93

*Серия основана в 2016 г.*

Ведущие редакторы серии *Т. Г. Хохлова, Ю. А. Серова*

Проект подготовлен под руководством В. Н. Халамова

**Рыжая Е. И.**

Р93 Конструируем роботов на LEGO® MINDSTORMS® Education EV3. В поисках сокровищ [Электронный ресурс] / Е. И. Рыжая, В. В. Удалов. — Эл. изд. — Электрон. текстовые дан. (1 файл pdf : 67 с.). — М. : Лаборатория знаний, 2017. — (РОБОФИШКИ). — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-00101-537-6

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

Вы соберёте из деталей конструктора LEGO® MINDSTORMS® Education EV3 робота, который умеет самостоятельно обнаруживать предметы, расположенные в комнате, подавать сигнал в случае находки и выводить на экран программируемого модуля их координаты.

Для технического творчества в школе и дома, а также на занятиях в робототехнических кружках.

**УДК 373.167  
ББК 32.97**

**Деривативное электронное издание на основе печатного аналога:** Конструируем роботов на LEGO® MINDSTORMS® Education EV3. В поисках сокровищ / Е. И. Рыжая, В. В. Удалов. — М. : Лаборатория знаний, 2017. — 64 с. : ил. — (РОБО-ФИШКИ). — ISBN 978-5-00101-053-1.

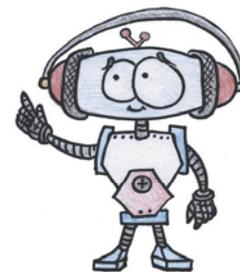
6+

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

ISBN 978-5-00101-537-6

© Лаборатория знаний, 2017

# Здравствуйте!



Издание, которое вы держите сейчас в руках, — это не просто описание и практическое руководство по выполнению конкретного увлекательного проекта по робототехнике. И то, что в результате вы самостоятельно сумеете собрать своими руками настоящее работающее устройство, — это, конечно, победа и успех!

Но главное — вы поймёте, что такие ценные качества характера, как терпение, аккуратность, настойчивость и творческая мысль, проявленные при работе над проектом, останутся с вами навсегда, помогут уверенно создавать своё будущее, стать реально успешным человеком независимо от того, с какой профессией вы свяжете жизнь.

Создавать будущее — сложная и ответственная задача. Каждый день становится открытием, если он приносит новые знания, которые затем могут быть превращены в проекты. Особенно это важно для тех, кто выбрал дорогу инженера и технического специалиста. Знания — это база, которая становится основой для свершений.

Однако технический прогресс зависит не только от знаний, но и от смелости создавать новое. Всё, что нас окружает сегодня, придумано инженерами. Их любопытство, желание узнавать неизведанное и конструировать то, чего никто до них не делал, и создаёт окружающий мир. Именно от таких людей зависит, каким будет наш завтрашний день. Только идеи, основанные на творческом подходе, прочных знаниях и постоянном стремлении к новаторству, заставляют мир двигаться вперёд.

И сегодня, выполнив этот проект и перейдя к следующим, вы сделаете очередной шаг по этой дороге.

Успехов вам!

*Команда Программы «Робототехника:  
инженерно-технические кадры инновационной России»  
Фонда Олега Дерипаска «Вольное Дело»*



## 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!

### Внимание!

Ты можешь собрать свои достижения в робототехнике в электронное портфолио! Фотографируй 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.

Что же делать теперь? Набор дорогóй, 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 хотел бы собрать робота для исследования пещер или подводных глубин, где прячутся самые интересные тайны и невиданные сокровища, затем нанести их местоположение на карту и отправиться за ними вместе с друзьями?

### Задумайся над этим!

Фактически за несколько часов 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?

# История эхолокации



Многие из вас бывали в дельфинарии на представлении дельфинов, где они прыгали через обруч, ловили мячики, ныряли и выпрыгивали из воды, и снова ныряли. Дельфины — удивительные животные. Они способны издавать звуки, недоступные человеческому уху. У дельфинов есть целая система звуковых сигналов — это «щелчки» или «свист», используемые дельфином для связи с сородичами или выражения эмоционального состояния.

В речи дельфинов учёные насчитали уже **186** разных «свистов»! У них примерно столько же уровней организации звуков, сколько и у человека, среди которых есть слоги, слова и фразы, а также сигналы для исследования обстановки, обнаружения препятствий и добычи.

Звуки распространяются не только в воздухе, но и в воде, и в твёрдых телах. Однако в различных средах звук распространяется по-разному — с разной скоростью. Быстрее всего звук «доходит» в стекле и в металлах, например в батареях водяного отопления (ты, наверное, с этим знаком). В воде звук распространяется медленнее, чем в металлах, но всё же в 4 раза быстрее, чем в воздухе. Получается, что рыбы услышат звук раньше, чем мы, стоя на берегу. Вот почему рыболовы не любят, когда кто-нибудь разговаривает около них! Причина в том, что звук — это колебания среды, и звуковые волны передаются быстрее в средах с большей плотностью.

Плотность воды больше, чем у воздуха, но меньше, чем у металла. Соответственно, и звук передаётся по-разному. При переходе из одной среды в другую скорость звука меняется.

Почему мы сейчас заговорили об этом? Да потому, что дельфины живут в воде! Под водой даже в солнечную погоду часто бывает плохая видимость из-за рыхлого дна (мутная вода) или из-за зарослей водорослей. Однако это совсем не мешает дельфинам охотиться, причём даже ночью, когда видимости совсем нет! Это загадка, которую учёные разгадали! Оказывается, водные животные полагаются не на зрение, а на другие органы чувств. Киты, морские свиньи (из семейства зубатых китов) и дельфины изучают окружающую среду и ориентируются в ней с помощью звуков, используя эхолокацию. Вот что означают их свисты, щелчки и щелчки!

**Эхолокация** у животных (рис. 1) — это процесс излучения и восприятия отражённых ультразвуковых (недоступных человеческому слуху) сигналов с целью обнаружения и получения информации о размерах добычи или препятствия. Животным, охотящимся под водой, острое зрение не требуется, а вот очень хороший слух просто необходим.

Летучие мыши живут не в воде, однако, как и дельфины, имеют свой мини-локатор. Крохотный летающий комочек, покрытый шерстью, живёт в настоящем

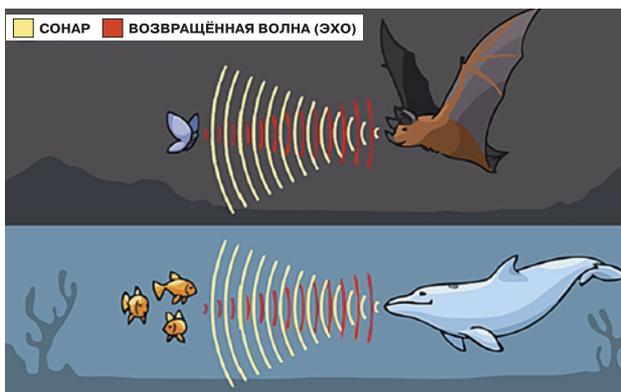


Рис. 1

мире звуков. Ещё в 1793 году выдающийся итальянский исследователь Ладзаро Спалланцани установил, что летучие мыши ориентируются и находят свою добычу с помощью слуха. Они издают не просто свист, не слышимый нашим ухом, а серию ультразвуковых щелчков.

Человек всегда внимательно присматривался к своим соседям по планете и старался применить в своей жизни подсмотренные у природы секреты. В XX веке даже возникла специальная наука — бионика (от греч. *Bios* — жизнь), изучающая возможности применения свойств и функций живых организмов в технике.

Одним из таких примеров и является эхолотатор.

**Локатор** (от англ. *Locate* — определять местоположение) — искусственное устройство или живой орган для обнаружения объекта и расстояния до него бесконтактным способом.

Вот различные примеры локаторов-устройств:

- лазерный дальномер (рис. 2) — это устройство для определения расстояния от наблюдателя до объекта с помощью световых волн;
- радиолокационная станция, или радар, — система для обнаружения

воздушных, морских и наземных объектов и для определения их дальности, скорости и геометрических параметров с помощью радиоволн;

- гидролокатор — устройство обнаружения подводных объектов с помощью ультразвуковых волн.

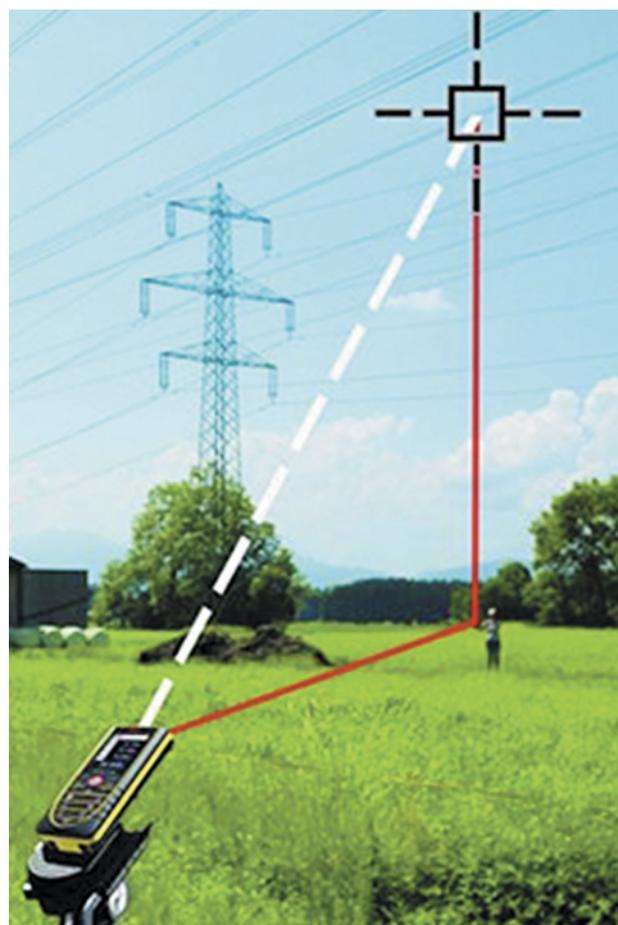


Рис. 2

Разновидностью гидролокатора является *эхолот* (рис. 3). Он состоит из ультразвукового передатчика (источника ультразвука), приёмника отражённых колебаний и компьютера для обработки полученных данных и составления карты рельефа морского дна.

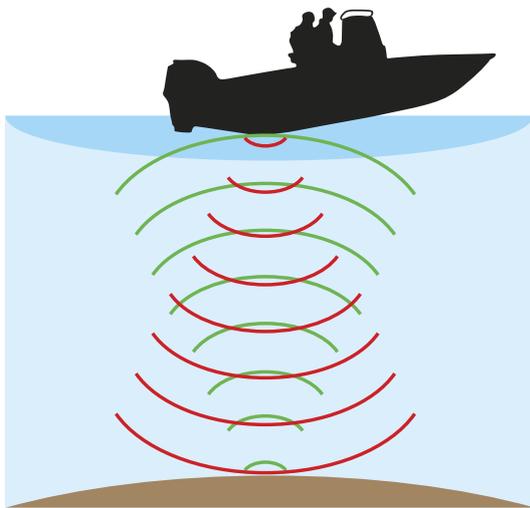


Рис. 3

*Навигационные* эхолоты применяют для определения глубины в целях обеспечения безопасности плавания судна, *промерочные* — для получения характеристики подводного рельефа. Эхолоты играют большую роль в морском и речном судождении. Если нельзя точно установить координаты корабля астрономическими методами по звёздам, поможет эхолот. В настоящее время почти для всех морей составлены подробные карты глубин, полученные при помощи эхолотов.

Для навигации и определения своего местоположения или местоположения других объектов с целью нанесения их на карту, моряки и подводники используют полярную систему координат (рис. 4).

В этой системе координат положение любого объекта на плоскости (например, на поверхности моря) однозначно задаётся двумя числами:

1) величиной **полярного угла**  $\phi$  в градусах (отсчёт идёт против часовой стрелки);

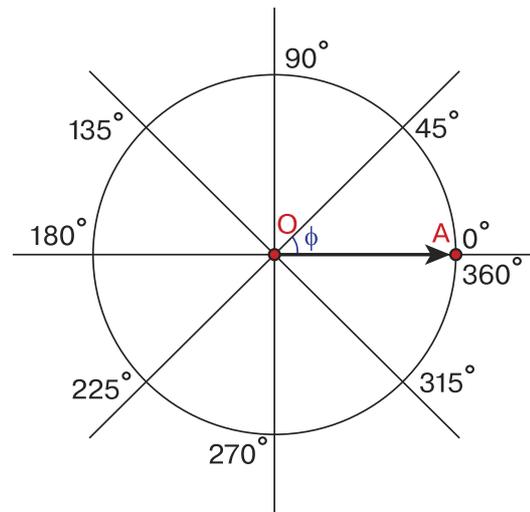


Рис. 4

2) величиной полярного радиуса — длиной отрезка **OA** в удобной единице измерения (например, в сантиметрах или километрах).

Обычно собственное местоположение совмещают с точкой  $(0;0)$ , которая соответствует точке **O** — началу отсчёта полярной системы координат, которую называют **полюсом**.

Именно по такому принципу работают радары (рис. 5), которые ты мог встречать даже в компьютерных играх.



Рис. 5

В рамках данного проекта мы с тобой соберём аналог модели поискового аппарата, оснащённого локатором, с помощью которого будем обнаруживать объекты и записывать их местоположение в полярных координатах.

### Обозначения:

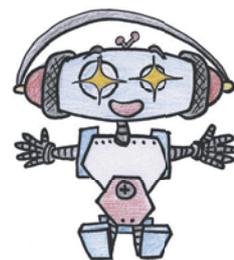
В тексте тебе встретятся обозначения, которые мы сейчас поясним на примерах.

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

### Оборудование:

- Базовый набор LEGO® MINDSTORMS® Education EV3.
- Компьютер (минимальные системные требования): Windows XP, Windows 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).
- Транспортёр.
- Измерительная рулетка или линейка.
- Бумага.
- Карандаш.

# Этап 1. Устройство робота-искателя



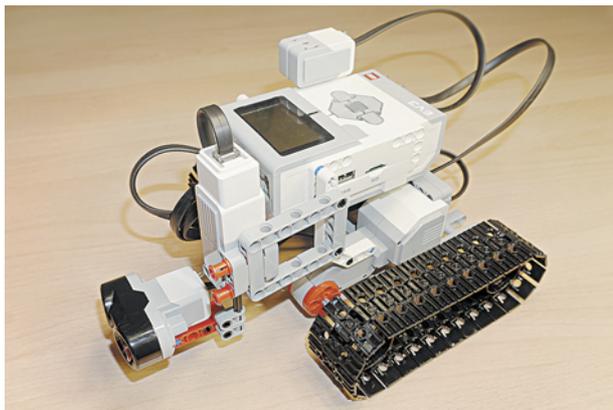
Рассмотри модель робота-искателя, собранную на основе набора LEGO® MINDSTORMS® Education EV3.

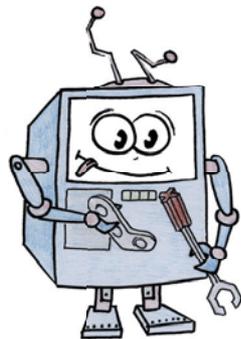
Найди в ней рабочие детали: гусеничное шасси на двух моторах; подвижный локатор со средним мотором и ультразвуковым датчиком; гироскопический датчик ориентации в пространстве.

Попробуй собрать эту модель.

Обрати внимание, в конце книги 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-искатель, которого тебе предстоит собрать.





## Этап 2. Сборка охотника за сокровищами

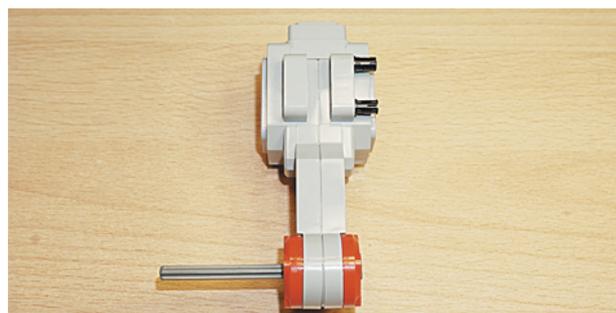
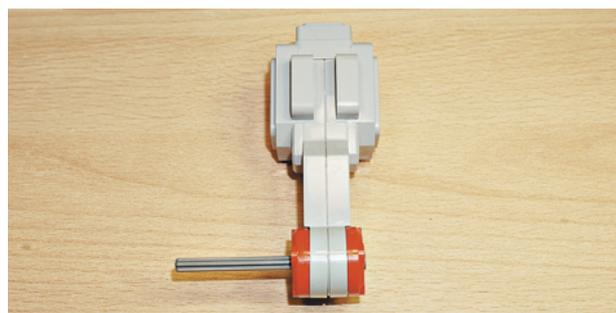
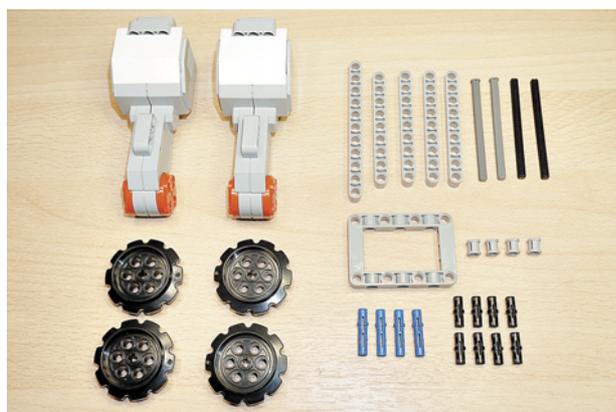
### ШАГ 1. СБОРКА ШАССИ

#### Детали для сборки:

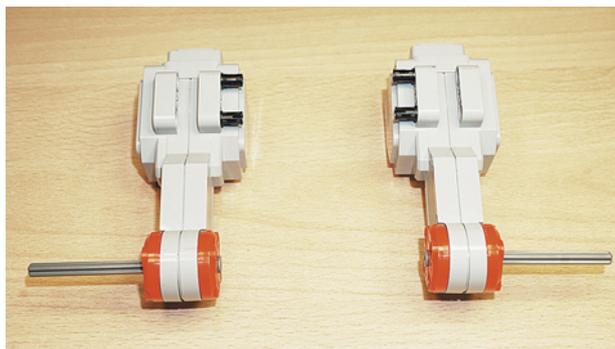
- большой мотор, 2х;
- балка № 11, серая, 1х;
- балка № 9, серая, 4х;
- ось № 8 с головкой, серая 2х;
- ось № 8, чёрная, 2х;
- рамка 5 × 7, серая, 1х;
- втулка, серая, 4х;
- цепное колесо, 4х;
- штифт 3-модульный, синий, 4х;
- штифт 2-модульный, чёрный, 8х.

1. Положи мотор, как показано на рисунке. С правой стороны продень в крестообразный модуль привода ось № 8 с головкой до упора.

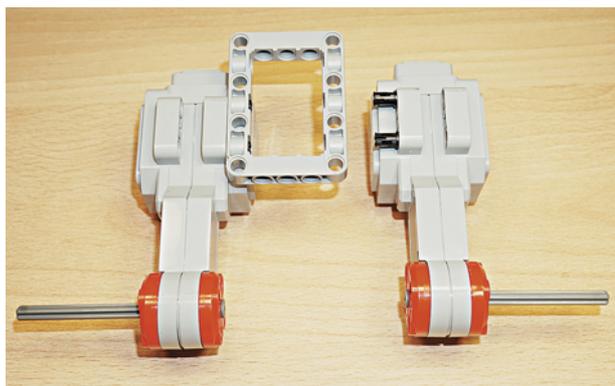
С той же стороны мотора вставь в первый и третий модули крепёжного элемента мотора по одному чёрному штифту.



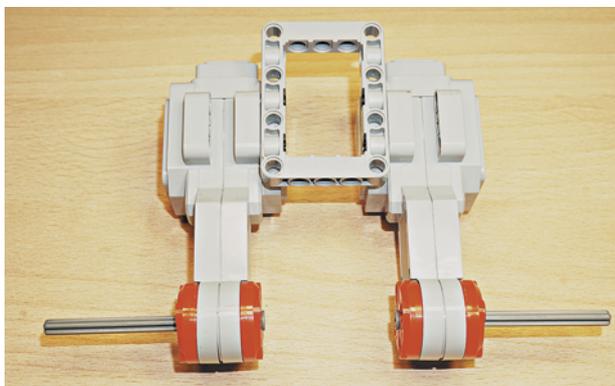
2. Возьми второй мотор и зеркально установи на нём те же элементы.



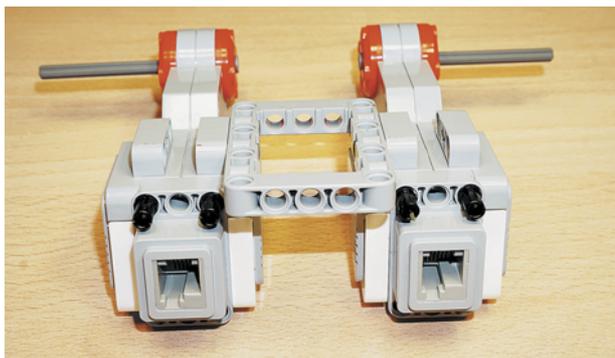
3. На чёрных штифтах левого мотора закрепи рамку 5 × 7 первым и вторым боковыми модулями.



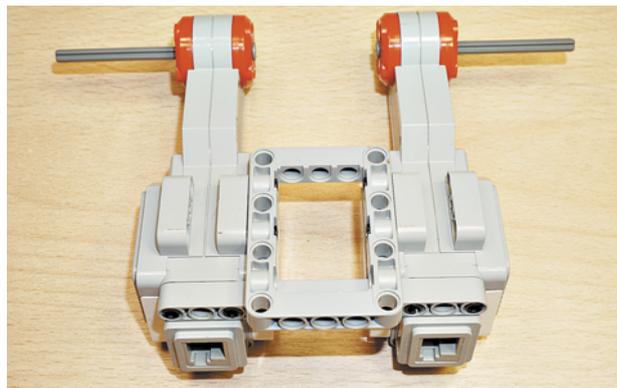
4. С другой стороны рамки симметрично закрепи правый мотор.



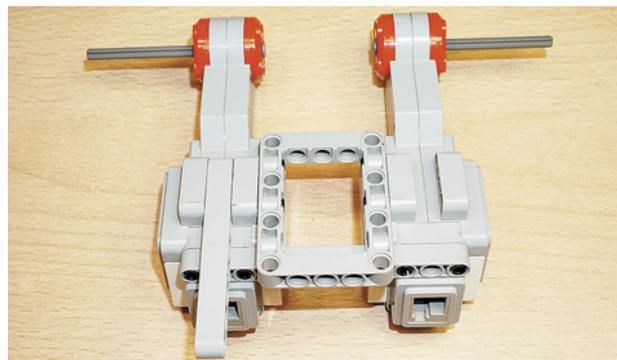
5. Разверни моторы осями от себя и вставь чёрные штифты в первый и третий верхние модули каждого из них.



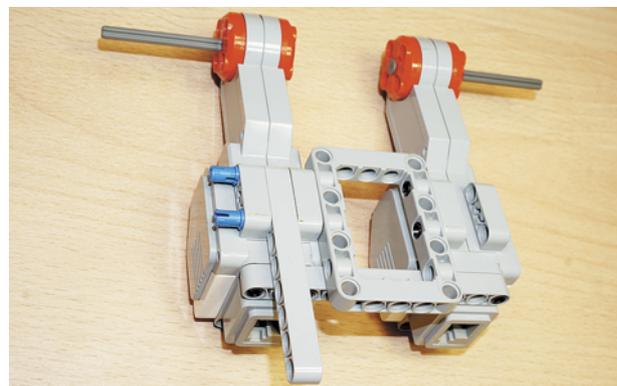
6. На этих штифтах закрепи балку № 11 (её середина окажется под рамкой 5 × 7).



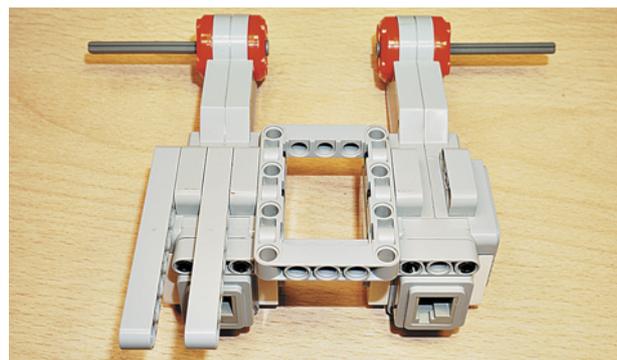
7. Между двумя крепёжными элементами левого мотора вложи балку № 9, как показано на рисунке.

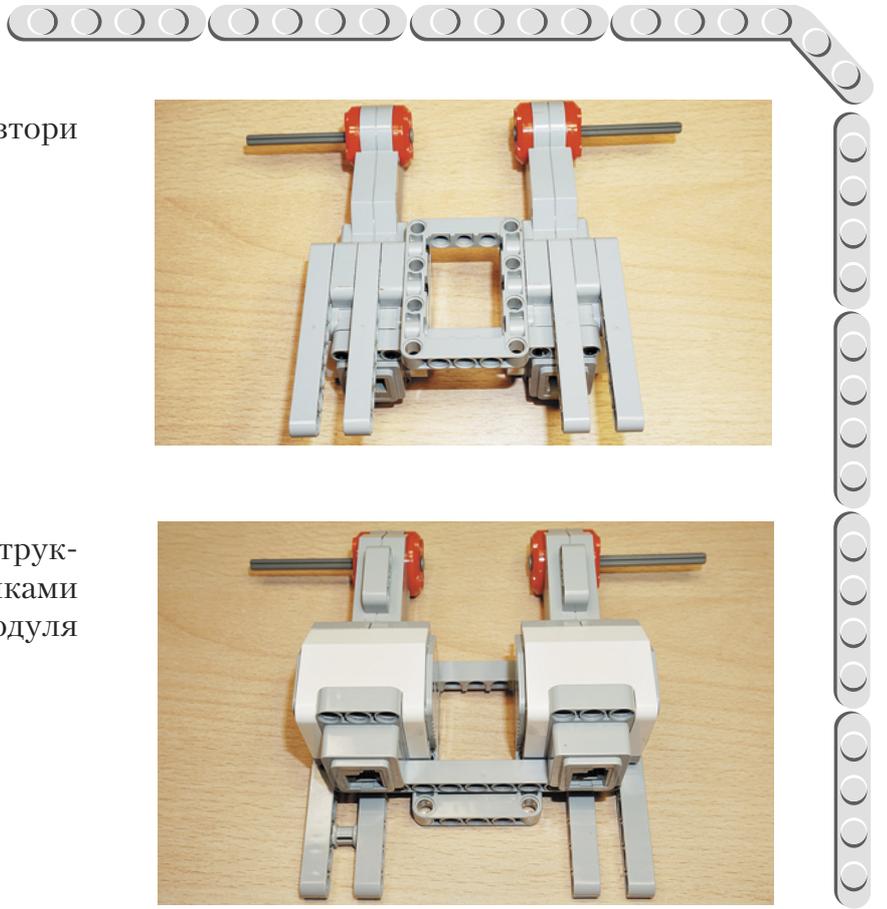


8. Слева в первый и третий модули левого крепёжного элемента мотора вставь синие штифты длинными концами вперёд. Они пройдут и сквозь модули балки.

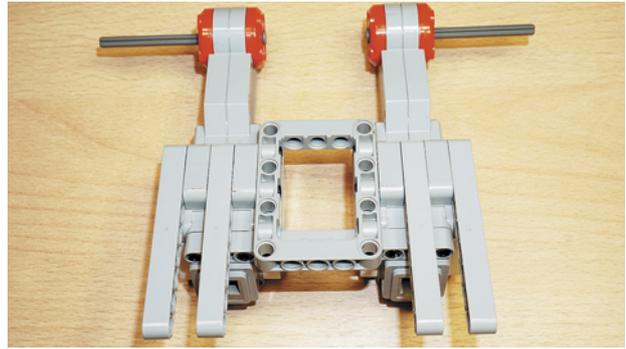


9. На свободные концы синих штифтов надень вторую балку № 9 крайними модулями.

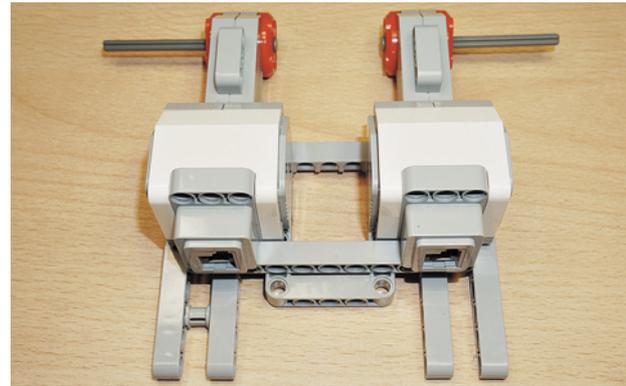




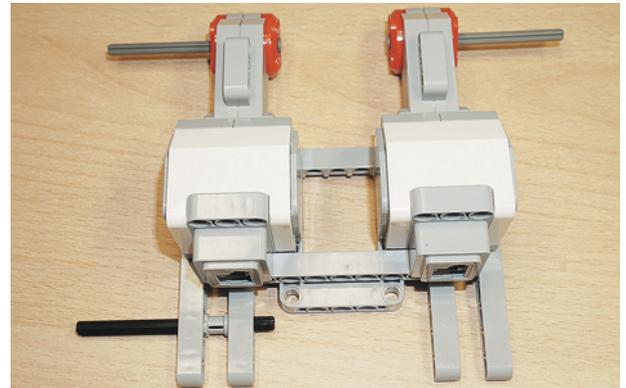
10. На правом моторе зеркально повтори пункты 7–9.



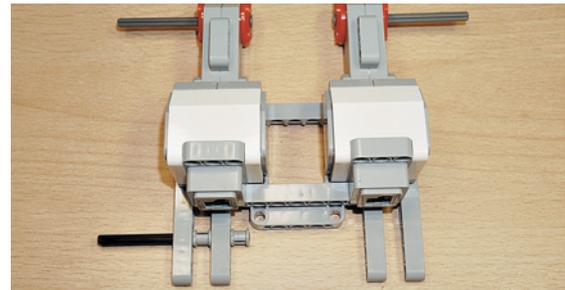
11. Переверни получившуюся конструкцию. Между двумя левыми балками № 9 напротив их третьего модуля вложи втулку.



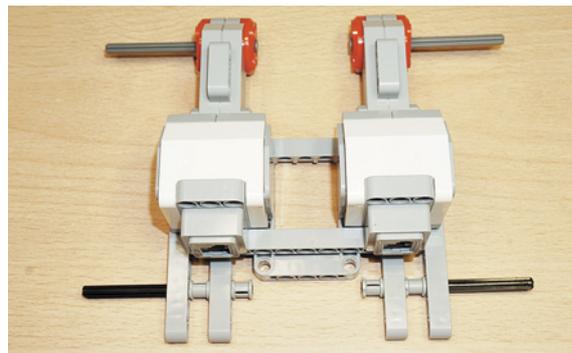
12. Продень с внешней стороны сквозь балки с втулкой чёрную ось № 8.



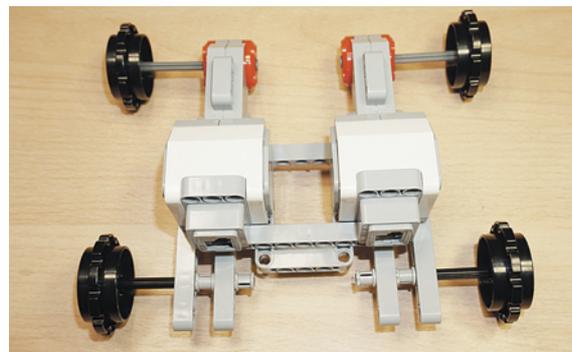
13. С внутренней стороны закрепи ось ещё одной втулкой так, чтобы ось не выглядывала из втулки.



14. Справа зеркально установи ещё одну чёрную ось № 8 с втулками.



15. На каждую из четырёх осей надень по одному цепному колесу так, чтобы концы осей не выступали снаружи.



Шасси готово!

## ШАГ 2. СБОРКА И КРЕПЛЕНИЕ ГУСЕНИЧНЫХ ЛЕНТ



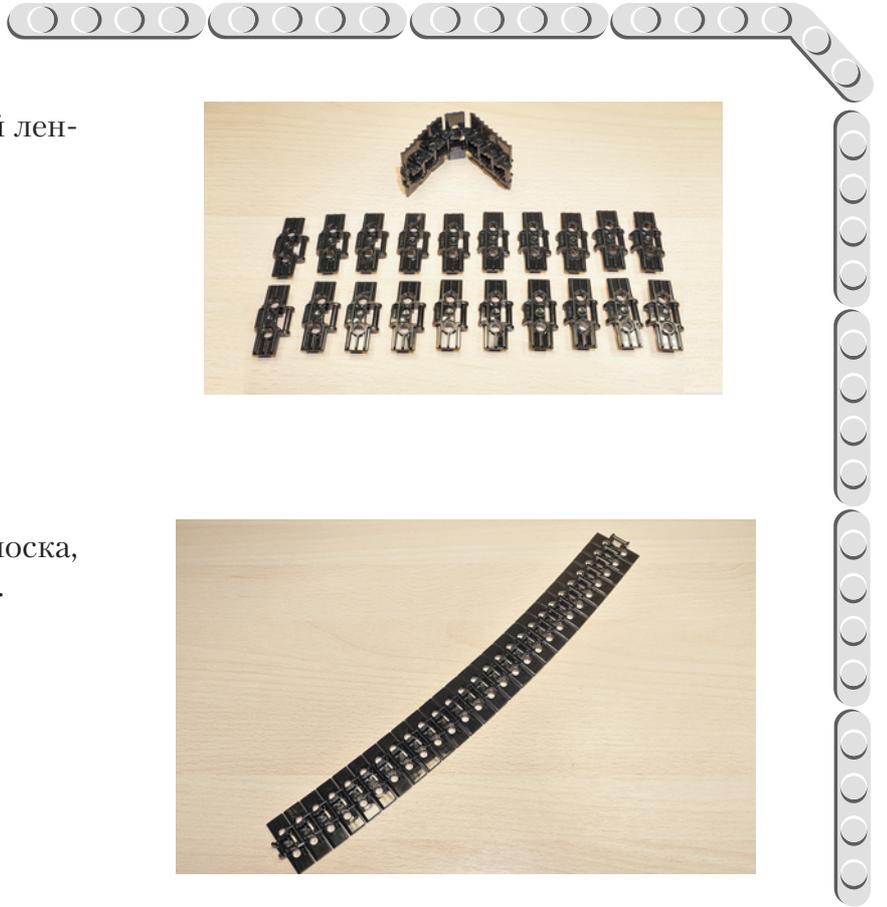
Детали для сборки:

- звено гусеничной ленты, 54х.

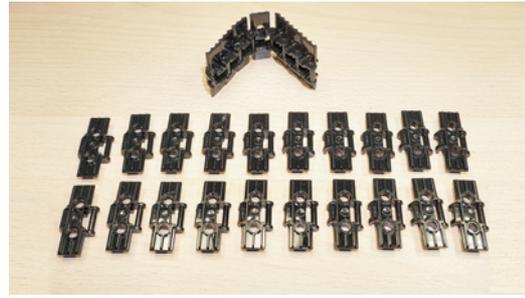


Тебе предстоит собрать и закрепить на шасси две гусеничные ленты. Собираются и крепятся они одинаково. Собери первую из них.





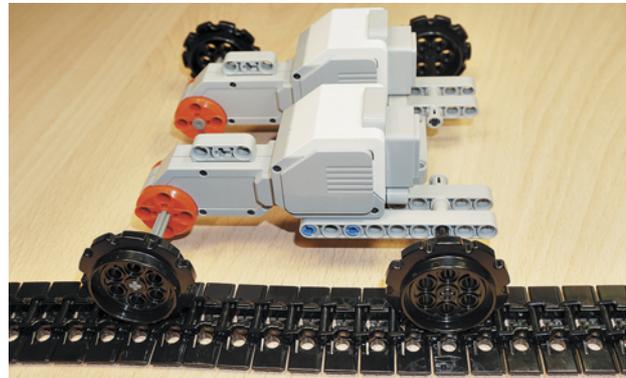
1. Отсчитай 27 звеньев гусеничной ленты и соедини их в цепочку.



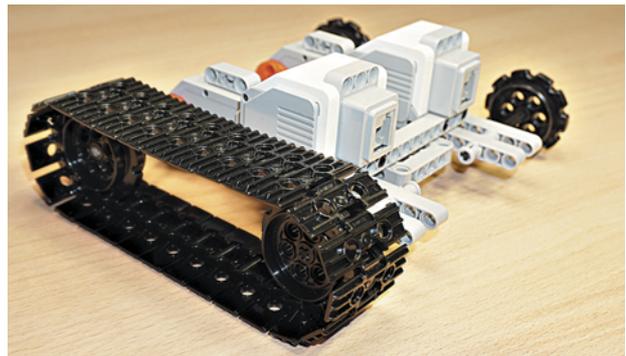
У тебя получится вот такая полоска, но не спеши скреплять её концы.



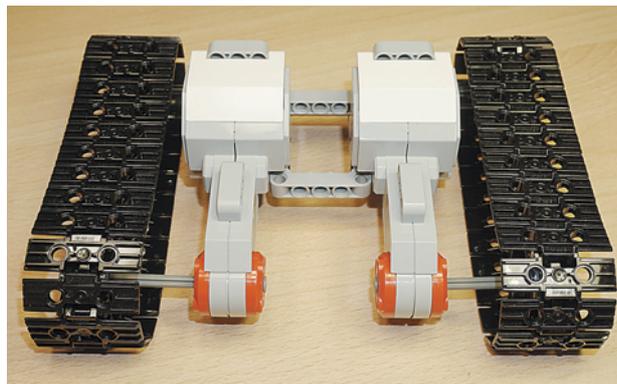
2. Поставь два левых цепных колеса на гусеничную ленту. Убедись, что выступающие части колёс легли в ямки на ленте.



3. Скрепи концы ленты, как показано на рисунке.



Точно так же собери и надень на цепные колёса вторую ленту.



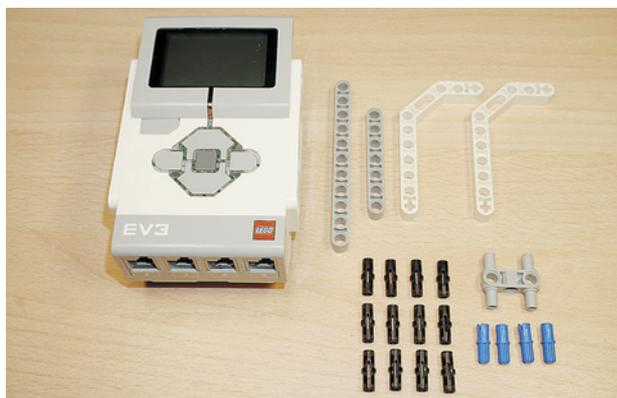
Теперь шасси «обуто»!

### ШАГ 3. СБОРКА КРЕПЛЕНИЯ БОРТОВОГО КОМПЬЮТЕРА

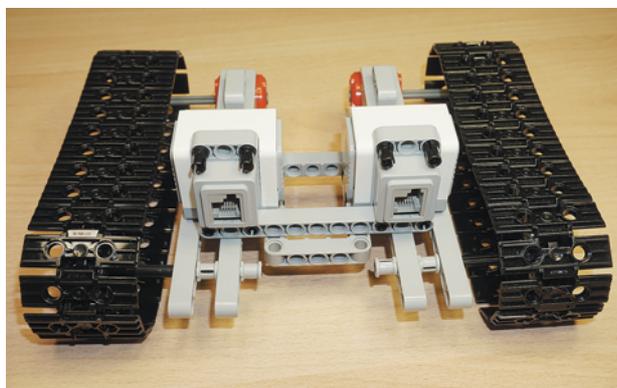


#### Детали для сборки:

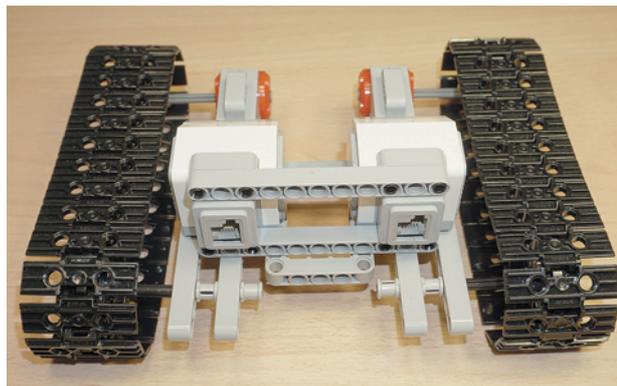
- программируемый модуль, 1х;
- двойная угловая балка 3 × 7, белая, 2х;
- балка № 11, серая, 1х;
- балка № 7, серая, 1х;
- Н-образный штифт, 3 × 1-модульный, серый, 1х;
- штифт 2-модульный, синий, 4х;
- штифт 2-модульный, чёрный, 12х.



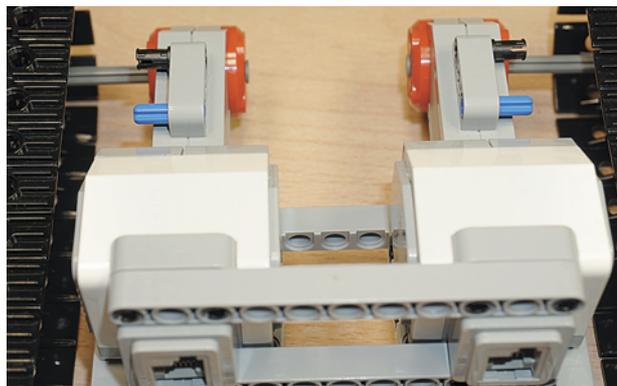
1. С задней стороны шасси на каждом моторе установи по два чёрных штифта в первом и третьем верхних модулях.



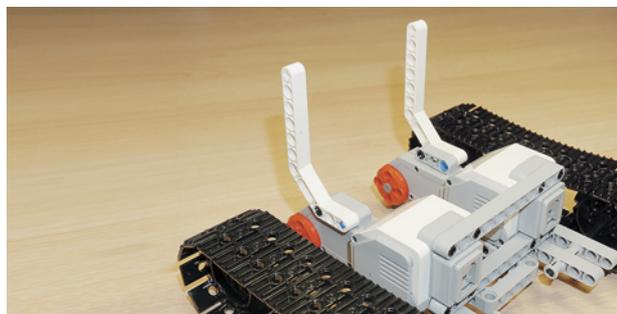
2. На этих штифтах закрепи балку №11.



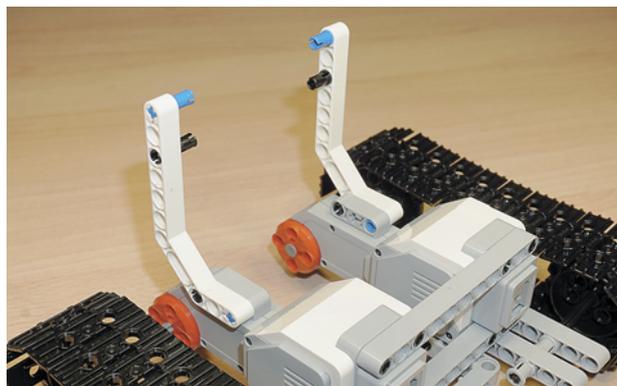
3. В передней части моторов с внешних сторон вставь в первые от тебя модули по одному синему штифту, а в третьей — по одному чёрному.



4. На установленных штифтах закрепи по одной двойной угловой балке  $3 \times 7$ , как показано на рисунке.



5. С внутренней стороны каждой из балок в первый крестовой модуль вставь по синему штифту, а в третий модуль — по чёрному штифту.



6. Возьми программируемый модуль и положи его экраном вниз. Со стороны портов, обозначенных цифрами, в средние модули вставь по одному чёрному штифту.



7. На этих штифтах закрепи балку № 7, чтобы все её модули совпали с отверстиями на EV3.



8. Установи H-образный штифт в третий и пятый модули балки № 7 и в его свободные модули спереди вставь чёрные штифты.



9. Переверни программируемый модуль экраном вверх. Установи его на шасси между двумя угловыми балками 3 × 7. При этом чёрные штифты должны войти в пятый и седьмой модули задней балки шасси. Штифты на угловых балках должны войти в боковые отверстия модуля EV3.



## ШАГ 4. СБОРКА И УСТАНОВКА ЛОКАТОРА

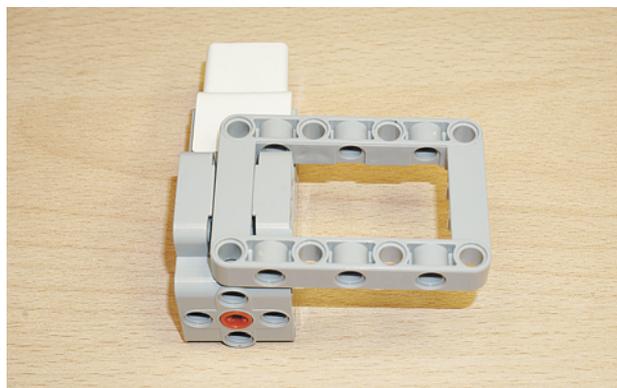


### Детали для сборки:

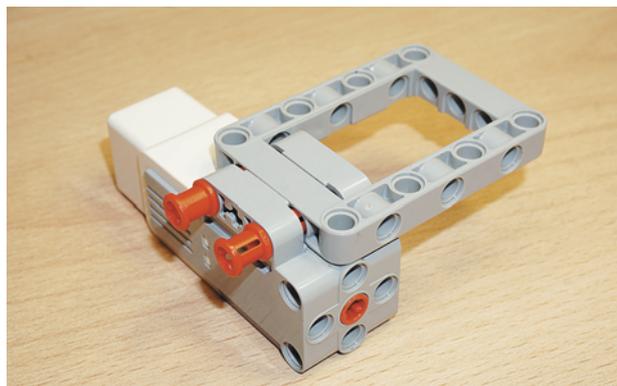
- средний мотор, 1х;
- ультразвуковой датчик, 1х;
- рамка 5 × 7, 1х;
- балка прямоугольная 2 × 4, красная, 1х;
- ось № 6, чёрная, 1х;
- поперечный блок 2-модульный, серый, 2х;
- штифт 2-модульный, синий, 1х;
- штифт 2-модульный, чёрный, 5х;
- штифт 3-модульный с фрикционной муфтой, красный, 2х.



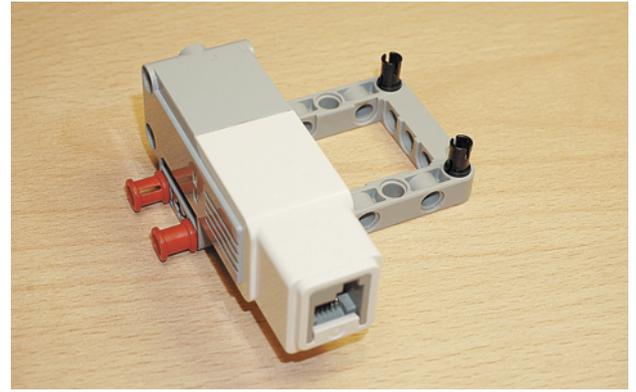
1. Возьми средний мотор и поверни его вверх дном. Вложи рамку 5 × 7 между двумя крепежами среднего мотора, как показано на рисунке.



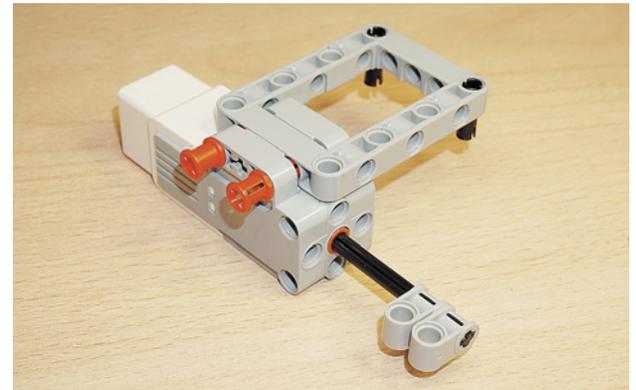
2. Зафиксируй рамку слева двумя красными штифтами с фрикционной муфтой.



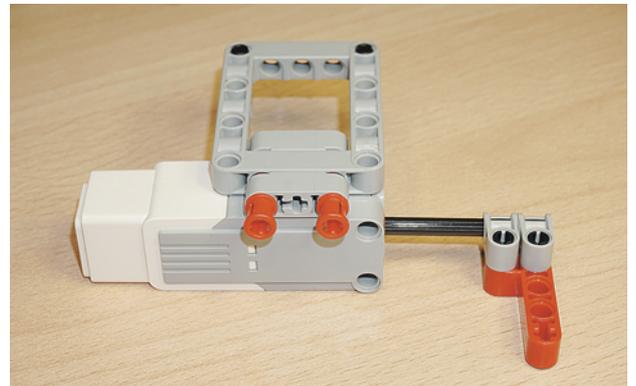
3. Переверни мотор. В крайние сверху отверстия рамки установи чёрные модули.



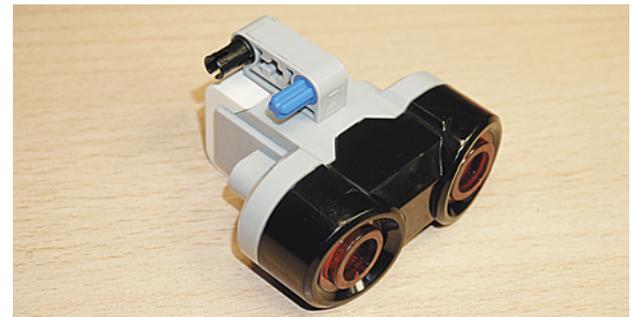
4. Ещё раз переверни мотор. Вставь ось № 6 в привод среднего мотора и надень на неё поперечные модули друг за другом так, чтобы ось не выступала за них.



5. В поперечные модули снизу вставь чёрные штифты и закрепи на них красную прямоугольную балку, как показано на рисунке.



6. В первый модуль ультразвукового датчика вставь синий штифт, а в третий — чёрный штифт.



7. Закрепи ультразвуковой датчик на красной балке снизу.



8. Установи получившуюся конструкцию с левой стороны программируемого модуля во второй и пятый модули угловой балки.



Локатор готов!

## ШАГ 5. СБОРКА И УСТАНОВКА ГИРОСКОПИЧЕСКОГО ДАТЧИКА



### Детали для сборки:

- гироскопический датчик, 1х;
- штифт 2-модульный, синий, 1х;
- штифт 2-модульный, чёрный, 2х;
- кабель, 25 см, 2х;
- кабель, 35 см, 2х;
- кабель, 50 см, 1х.

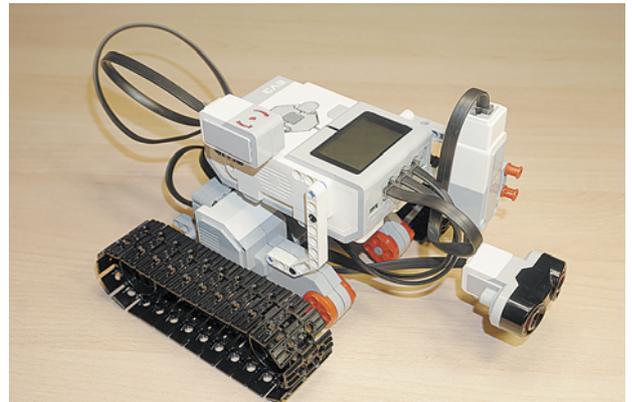
1. В модули с левой стороны гироскопического датчика установи друг за другом чёрный штифт, синий штифт и ещё чёрный.



2. Получившуюся конструкцию закрепи с правой стороны программируемого модуля в трёх свободных верхних отверстиях.

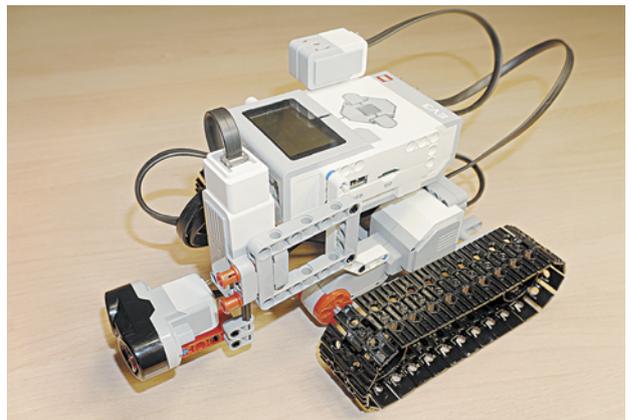


3. Подключи кабели:
  - а) правый большой мотор кабелем 35 см к порту С;
  - б) левый большой мотор кабелем 35 см к порту В;
  - в) средний мотор кабелем 25 см к порту D;
  - г) ультразвуковой датчик кабелем 50 см к порту 4;
  - д) гироскопический датчик кабелем 25 см к порту 3.



Проверь, чтобы кабели не касались подвижных частей робота-искателя.

Вот какой у тебя получился робот-искатель — вооружённый бортовым компьютером, двумя датчиками, тремя моторами и мощными гусеницами! К поискам он будет готов, когда ты вложишь в него инструкцию-программу.



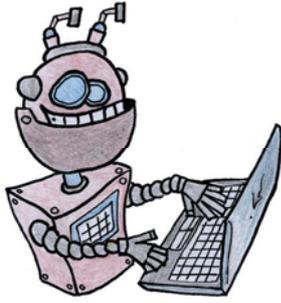
## Этап 3. Установка программного обеспечения на компьютере



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 на сайте <http://education.LEGO.com>



## Этап 4. Создание программы для робота-искателя

### ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

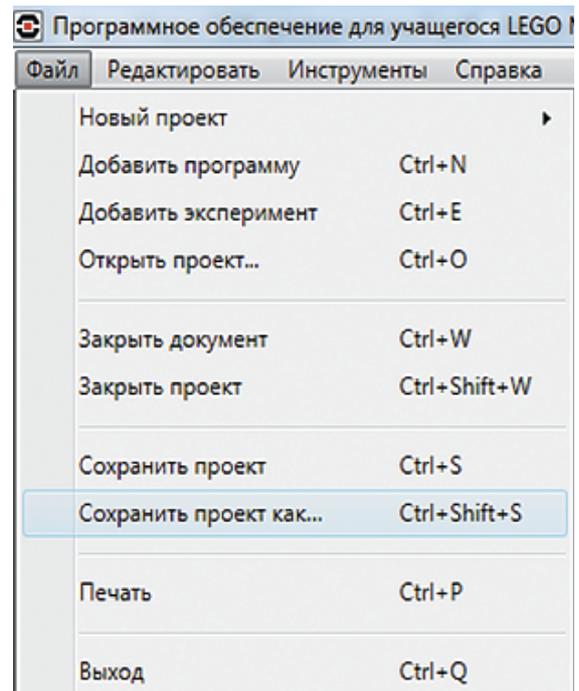
1. Запусти программное обеспечение LME-EV3.
2. В открывшемся окне слева выбери пункт меню **Файл**.
3. Выбери пункт **Новый проект** → **Программа** → **Открыть**.

### 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, где он записан на 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, чтобы не потерять проект, сохрани его в памяти компьютера. Для этого выбери меню **Файл** в левом верхнем углу окна программы и в контекстном меню выбери пункт **Сохранить проект как...**

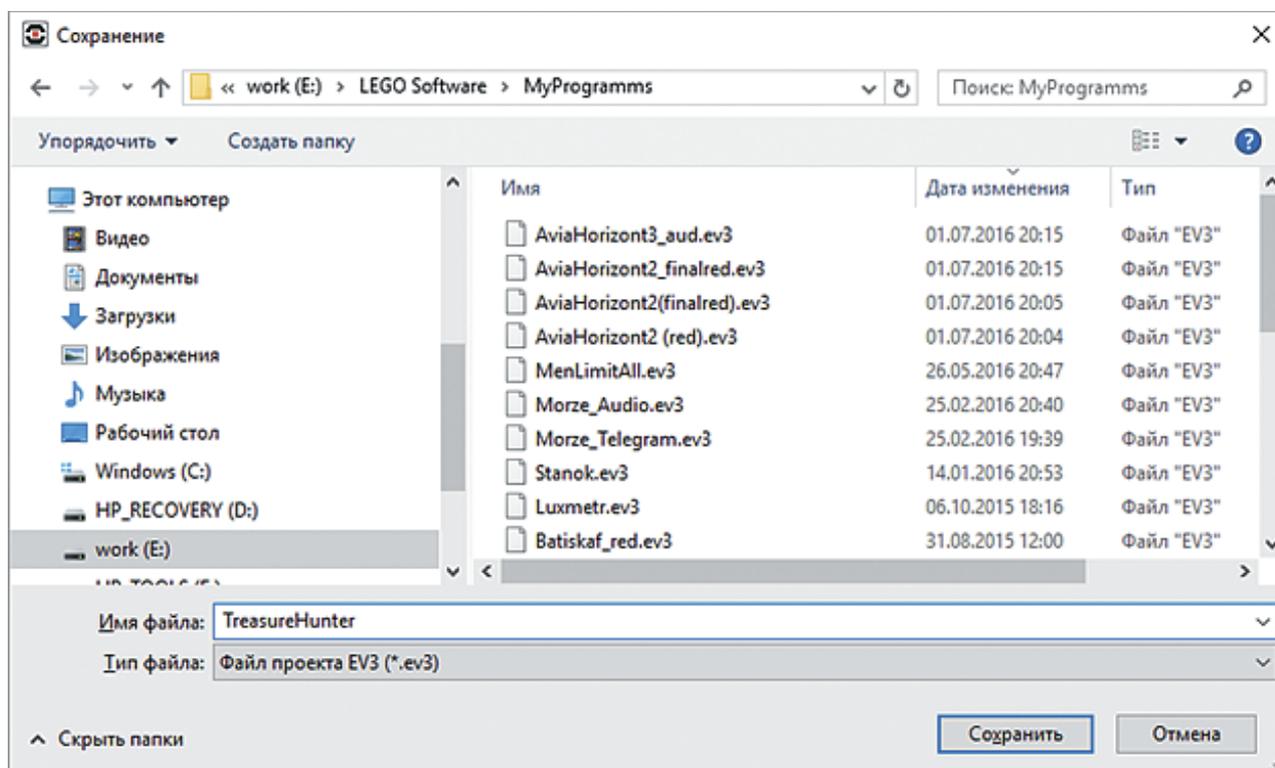


2. В открывшемся окне выбери удобное расположение для всех файлов твоей программы, затем задай **Имя файла**: *Treasure Hunter* (от англ. «Охотник за сокровищами») в нижней части окна и нажми **Сохранить**.

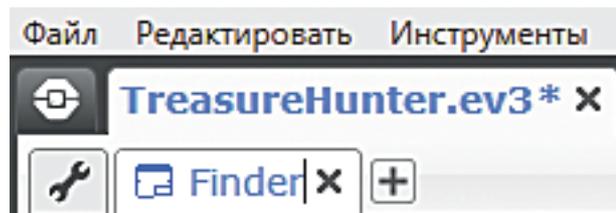
Именно так будет называться папка в памяти программируемого модуля EV3, в которой будут храниться все программы, которые ты напишешь в рамках этого проекта.

### 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 и ты) или добавляют нижнее подчёркивание.



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 и задай новое имя: *Finder* (от англ. «Искатель»), затем нажми клавишу **Enter**.



## ЛОГИКА ПРОГРАММЫ

Программа будет состоять из нескольких ветвей:

1. Исходные положения. Объявление необходимых переменных и настройка начальных параметров.
  2. Поиск сокровищ. Обеспечение поворотов локатора.
  3. Неизведанное. Управление движением робота на плоскости.
  4. Пластиры! Действия при обнаружении объекта (сокровищ).
  5. В трёх соснах. Исключение ошибки при полном развороте.
  6. Топография. Конечный цикл работы робота.
- Логику для каждой ветви мы опишем отдельно.

### Кстати!

В данной программе будет несколько интересных алгоритмических конструкций и приёмов программирования. Одним из таких приёмов будет **принудительное прерывание цикла**.

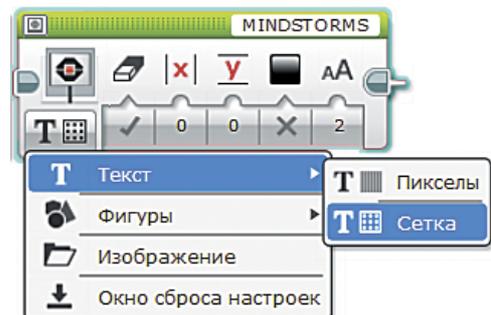
## СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ РОБОТА-ИСКАТЕЛЯ

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

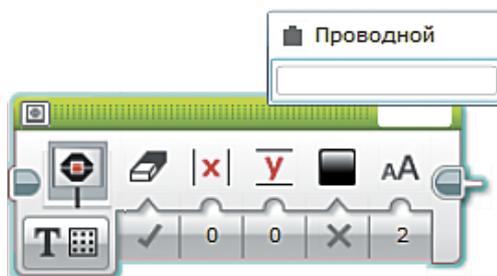
### Ветвь 1. Исходные положения. Объявление необходимых переменных и настройка начальных параметров

**Логика.** В этой части программы ты очистишь экран EV3, объявишь все нужные переменные и сбросишь показания гироскопического датчика.

1. Для начала нужно очистить экран от всех надписей. Для этого добавь команду **Экран** (зелёный блок).
2. Выбери опцию **Текст** → **Сетка**.



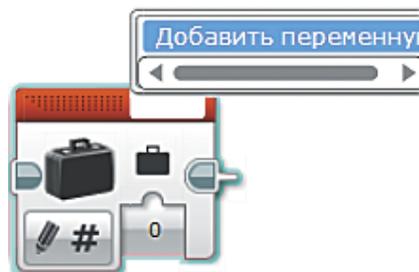
3. В правом верхнем углу, где написано слово MINDSTORMS, кликни мышью, сотри эту надпись и нажми клавишу **Enter**. Именно так ты очистишь экран, по сути выведя на него «пустой» текст.



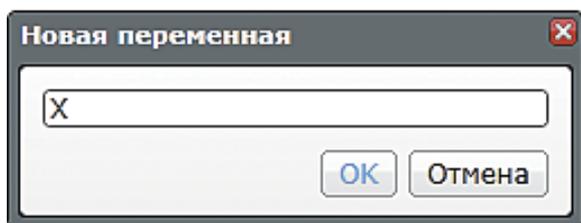
4. Теперь нужно объявить первую переменную. Для этого добавь команду **Переменная** (красный блок).



5. В правом верхнем окошке выбери пункт **Добавить переменную**.



6. В открывшемся диалоговом окне введи имя **X** («икс») и нажми **ОК**. Эта переменная будет отвечать за координату  $x$  (горизонталь) при выводе текста на экран EV3.



7. Точно так же создай ещё две переменные: **Y** (отвечает за координату  $y$  (вертикаль) при выводе информации на экран EV3) и **Count** (считает количество обнаруженных объектов).

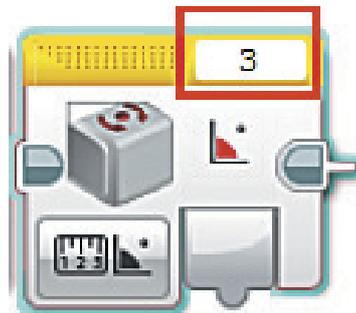


### Помни!

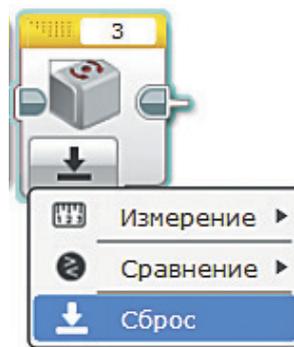
Настоящие программисты всегда дают имена переменным сообразно тому, какие данные и для чего они хранят.

Опции менять не нужно — переменные объявлены и *инициализированы* (заданы их начальные значения).

- И последнее, что осталось сделать, — сбросить показания гироскопического датчика. Добавь команду **Гироскопический датчик** (жёлтый блок). Проверь, чтобы в правом верхнем окошке был выбран порт, к которому подключён датчик (в нашем случае это порт № 3).



- Выбери опцию **Сброс**.



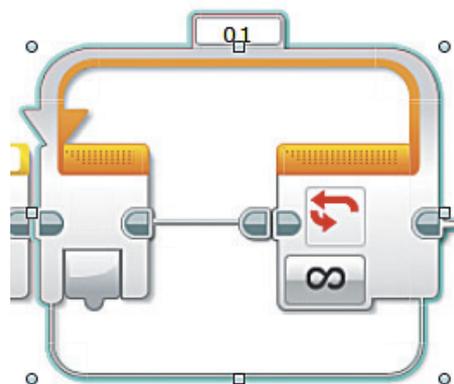
На этом первая ветвь программы завершена! Молодец!

## Ветвь 2. Поиск сокровищ. Обеспечение поворотов локатора

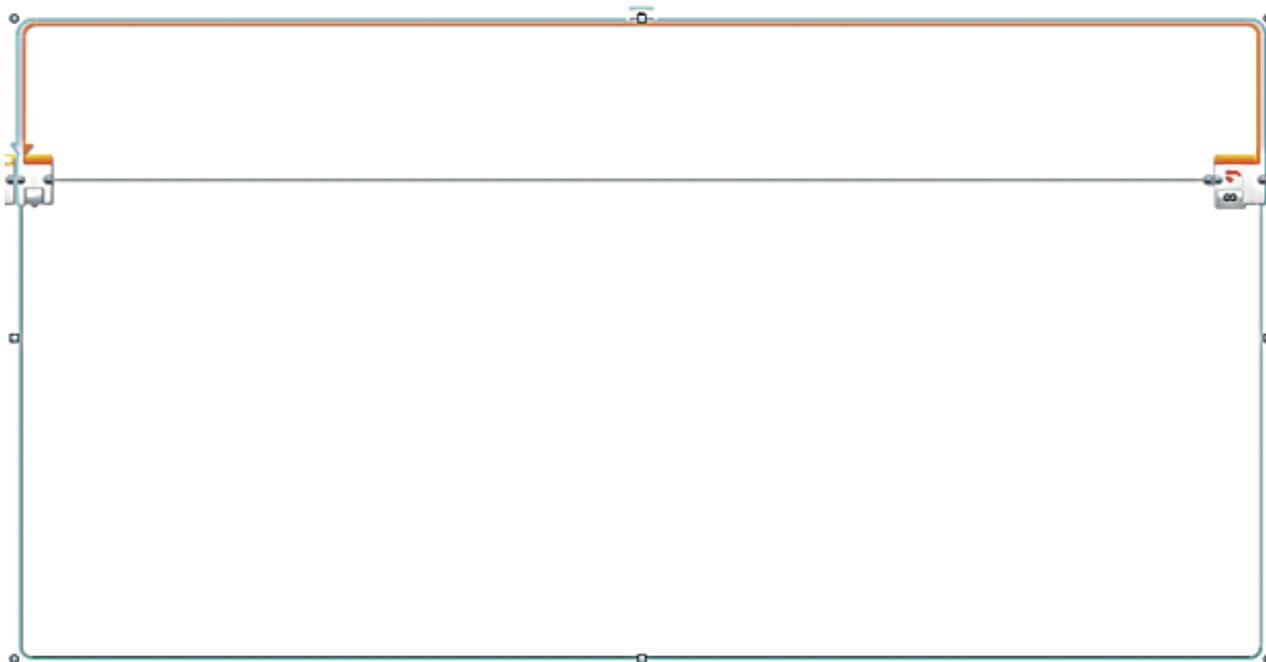
**Логика.** В этой части программы ты создашь основное рабочее поле и обеспечишь работу среднего мотора, который будет поворачивать локатор (ультразвуковой датчик) из стороны в сторону в поисках сокровищ. Вся остальная программа, за исключением последней ветви, будет заключена в одном большом бесконечном цикле.

Приступим.

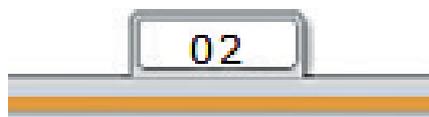
- Добавь команду **Цикл** (оранжевый блок).



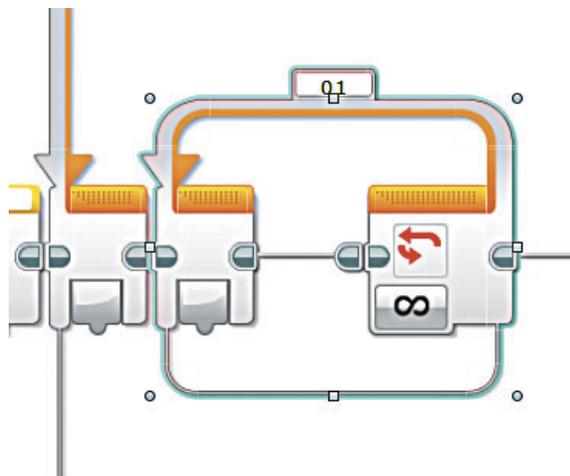
- И сразу увеличь её рабочее пространство, потянув мышью за окружающие его кружочки и квадратики.



- В верхней части этой команды замени цифры **01** на **02** – это уникальный идентификатор данного цикла, по которому к нему можно обращаться в программе (нам понадобится это позже).

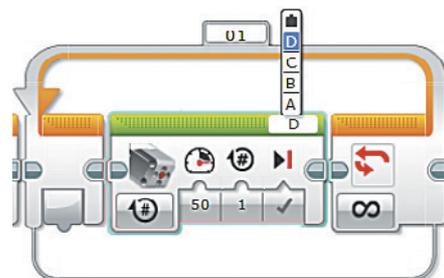


- Внутри этого цикла добавь ещё один **Цикл** (оранжевый блок).

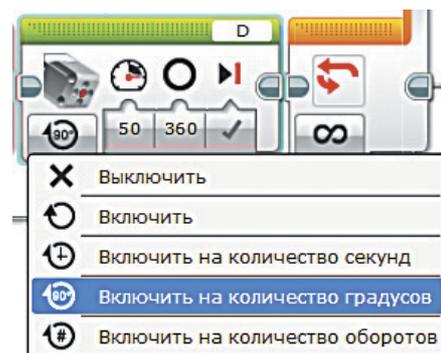


Идентификатором этого и других циклов будет оставаться 01.

5. Теперь надо задать повороты лока-  
тора. Добавь внутрь этого цикла ко-  
манду **Средний мотор** (зелёный блок).  
Проверь, чтобы в правом верхнем  
окошке значился порт, к которому  
подключён средний мотор (в нашем  
случае это порт D).

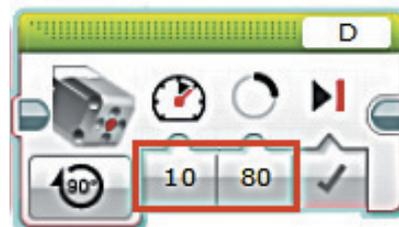


6. Выбери опцию **Включить на количе-  
ство градусов**.



7. Настрой параметры работы мотора:

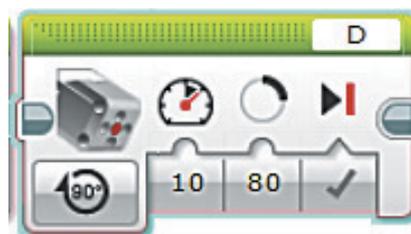
- **мощность: 10;**
- **градусы: 80.**



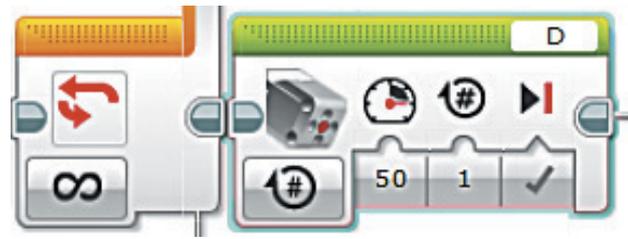
8. Итак, ты обеспечил поворот лока-  
тора только в одну сторону. Нужно задать  
его поворот и в другую сторону. Для  
этого тебе опять понадобится коман-  
да **Средний мотор** (зелёный блок),  
только значение градусов нужно за-  
дать вдвое большим и со знаком «минус»,  
чтобы привод двигался в противополож-  
ную сторону: **-160**.



9. Осталось вернуть лока-  
тор в началь-  
ное положение, после чего он начнёт  
повторять свои движения. Для этого  
добавь и настрой ещё одну коман-  
ду **Средний мотор** (зелёный блок),  
и в ней задай значение градусов **80**.

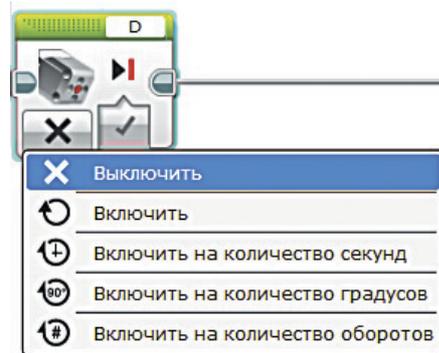


10. Далее в программе ты будешь принудительно прерывать этот цикл, как только сокровище (или любой другой объект) будет найдено. После того, как цикл прервётся, локатор должен остановиться. Для этого добавь ещё одну (четвёртую) команду **Средний мотор** (зелёный блок), но только уже после цикла.



11. И выбери опцию **Выключить**.

Первый рабочий элемент твоего робота-искателя теперь готов сканировать окружающее пространство!

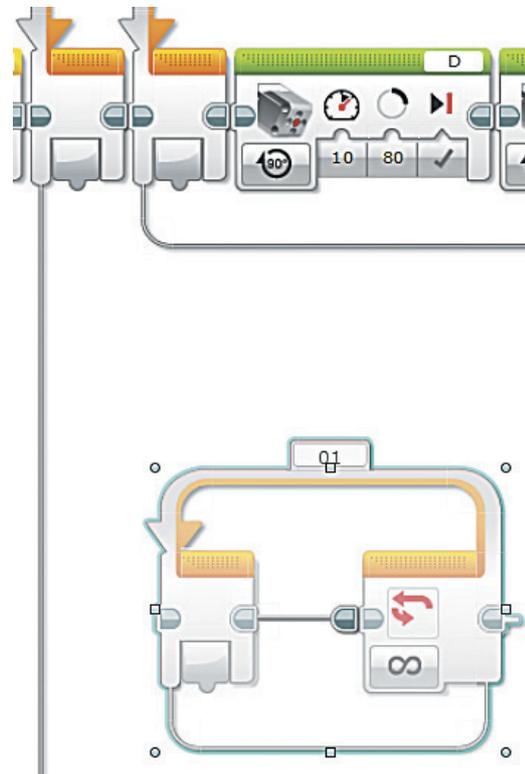


### Ветвь 3. Неизведанное. Управление движением робота на плоскости

**Логика.** В этой части ты задашь характер движения твоего робота по поверхности. Он будет ехать какое-то время вперёд и периодически немного изменять направление своего движения. Каждый раз, когда траектория будет изменяться, на экран будут выводиться две полярные координаты: пройденное расстояние и угол, под которым двигался робот до этого момента. Как и в предыдущей ветви, все команды будут заключены в цикле, после прерывания которого робот должен остановиться.

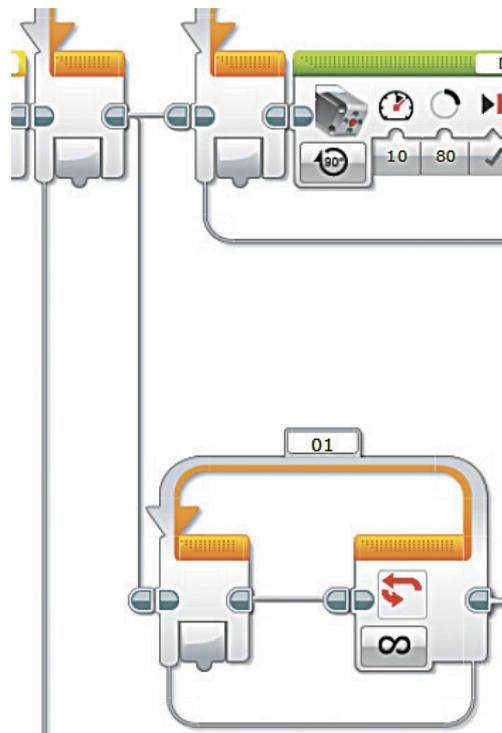
Приступим.

1. Под предыдущим циклом помести ещё одну команду **Цикл** (оранжевый блок).

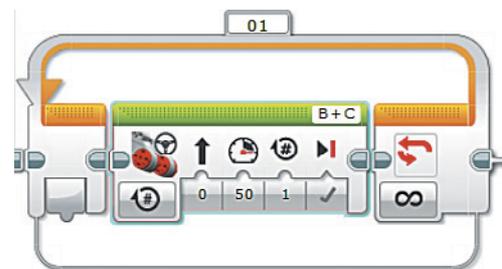


Как видишь, новая команда сейчас не подсвечена, поскольку она не привязана к общей связке и неактивна (то есть выполняться не будет).

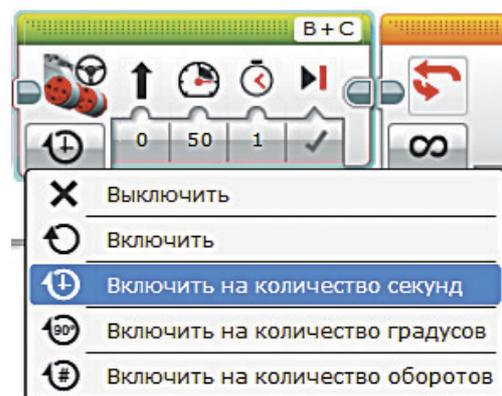
2. Чтобы её активировать, потяни с помощью мыши выходную связку от большого цикла и присоедини её к маленькому.



3. Внутри этого цикла добавь команду **Рулевое управление** (зелёный блок).

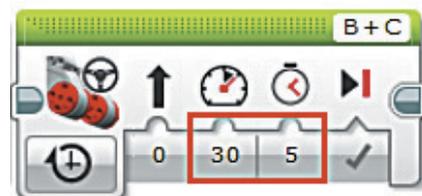


4. Выбери опцию **Включить на количество секунд**.



5. Теперь настрой параметры команды:

- **мощность:** 30;
- **секунды:** 5.



Таким образом, робот-искатель будет ехать вперёд 5 секунд.

Далее тебе надо рассчитывать пройденное роботом расстояние, чтобы в любой момент можно было вывести его на экран. Чтобы посчитать эту величину, нужно знать **Количество совершенных оборотов колеса** и **Длину окружности колеса**.

Количество оборотов ты можешь получить автоматически, так как все моторы оснащены энкодерами<sup>1</sup>, а вот длину окружности колеса нужно рассчитать.

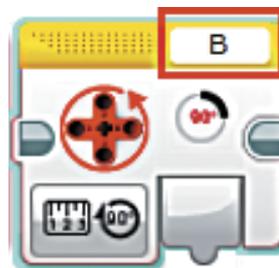
Ты можешь сказать: «Но ведь наш робот гусеничный! У него же нет колёс!». Однако движение гусениц обеспечивается цепными колёсами, которые заменяют колёса.

Измерь линейкой диаметр колеса — от верхней кромки гусениц до нижней. У нас эта величина оказалась равной 4,6 см. Теперь умножь это число на 3,14:

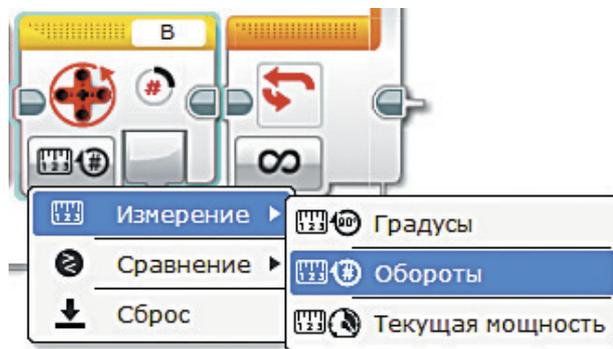
$4,6 \text{ см} \cdot 3,14 \approx 14,4 \text{ см}$  — именно столько будет проезжать робот за один оборот цепного колеса.

Продолжаем.

6. Добавь команду **Вращение мотора** (жёлтый блок). Для расчёта пройденного расстояния удобно ориентироваться на какой-то один двигатель, например, подключённый к порту В. Поэтому проверь, чтобы в правом верхнем окошке значился порт **В**.

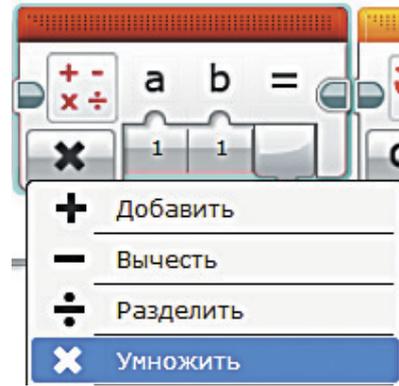


7. Выбери опцию **Измерение** → **Обороты**.

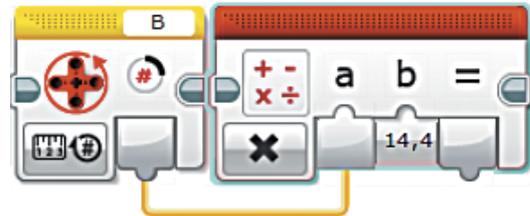


<sup>1</sup> Энкодер (от английского *Encoder* — кодировщик) — устройство внутри мотора, позволяющее снимать различные показания: число совершённых оборотов или угол поворота привода.

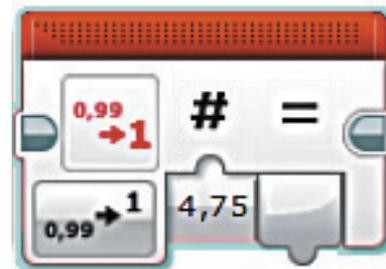
8. Далее добавь команду **Математика** (красный блок) и выбери опцию **Умножить**.



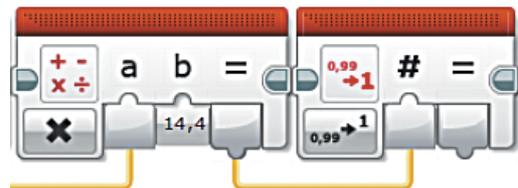
9. Соедини выход команды **Вращение мотора** с входом **a** команды **Математика**, во вход **b** введи с клавиатуры рассчитанную длину окружности колеса: **14,4**.



10. Полученное значение необходимо округлить до целого. Для этого добавь команду **Округление** (красный блок).



11. Соедини выход «**=**» команды **Математика** с входом «**#**» команды **Округление**.



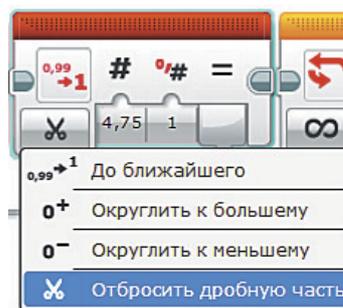
12. Теперь одна из координат — расстояние — готова к выводу. Подготовим вторую — полярный угол. Добавь команду **Гироскопический датчик** (жёлтый блок).



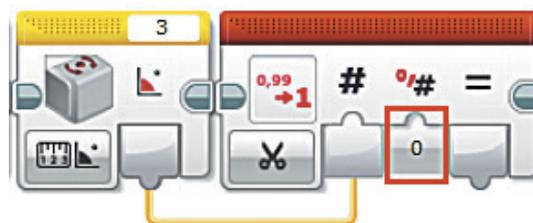
Вспомни полярную систему координат, о которой рассказывалось в начале проекта. Движение против часовой стрелки (для робота-искателя — поворот направо) было положительным направлением, а движение в обратную сторону (для робота-искателя — поворот налево) — отрицательным. Таким образом, значение полярного угла может быть как положительным, так и отрицательным.

У измеренного значения полярного угла отбрось дробную часть — слишком точные показания не нужны, так как показания с точностью до одного градуса для поиска сокровищ достаточно точны.

- Добавь команду **Округление** (красный блок) и выбери опцию **Отбросить дробную часть**.

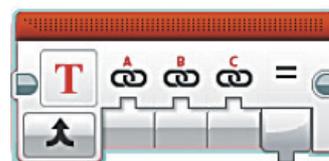


- Соедини выход команды **Гироскопический датчик** с входом **Ввод** команды **Округление**, во вход **Число десятичных знаков** введи с клавиатуры **0** — это означает, что вся дробная часть будет отсечена.

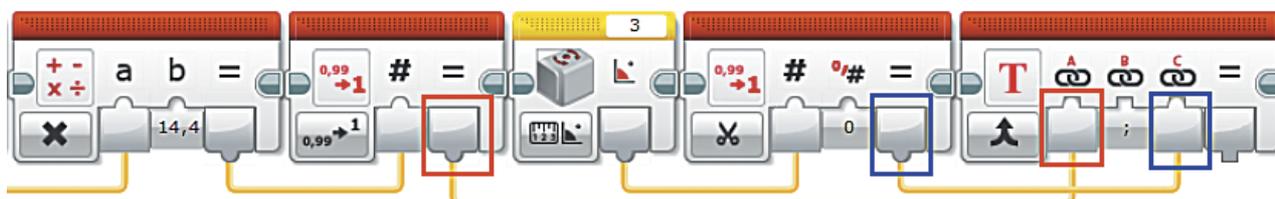


Теперь сформируем координаты для вывода на экран.

- Добавь команду **Текст** (красный блок).

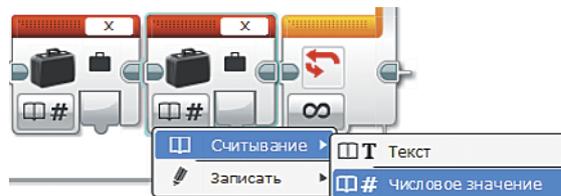


- Соедини выход «=» первой команды **Округление** от мотора со входом **A** команды **Текст**. Во вход **B** этой команды введи с клавиатуры знак «;», выход «=» второй команды **Округление** от гироскопического датчика соедини со входом **C**.

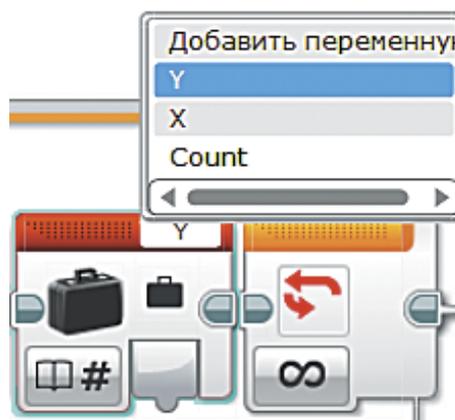


Сформированные координаты нужно вывести на экран, предварительно задав на экране место, куда их выводить. За это отвечают переменные **X** и **Y**.

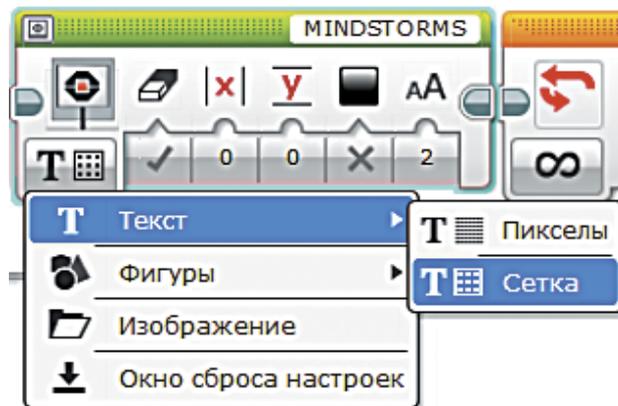
17. Добавь две команды **Переменная** (красный блок) и выбери в обеих опцию **Считывание** → **Числовое значение**.



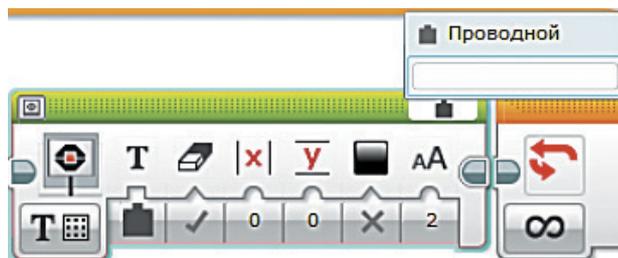
18. Во второй команде в правом верхнем окошке установи переменную **Y** для обработки.



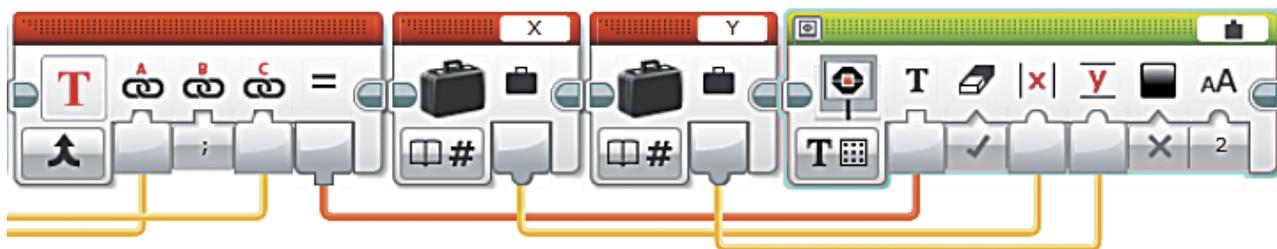
19. Теперь займёмся непосредственно выводом информации на экран. Добавь команду **Экран** (зелёный блок) и выбери опцию **Текст** → **Сетка**.



20. В правом верхнем окошке выбери пункт **Проводной**.



21. Соедини выход «= $\Rightarrow$ » команды **Текст** с входом **T** команды **Экран**, а выходы переменных **X** и **Y** с соответствующими входами **x** и **y**.



22. Настрой оставшиеся параметры команды **Экран**:

- **очистить экран:** ложь;
- **шрифт:** 1.

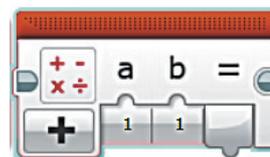


После того, как координаты выведены на экран EV3, нужно подготовить место для следующих. Для этого нужно сдвинуть одну из координат  $x$  или  $y$ . Начало отсчёта на экране находится в левом верхнем углу, значит, выводить координаты удобнее в столбик, двигаясь по вертикали (то есть увеличивая переменную **Y**).

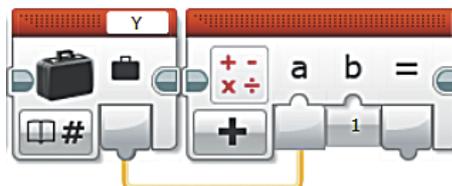
23. Добавь команду **Переменная** (*красный блок*), выбери опцию **Считывание**  $\rightarrow$  **Числовое значение** и для обработки установи переменную **Y**.



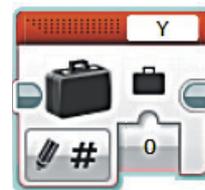
24. Далее добавь команду **Математика** (*красный блок*).



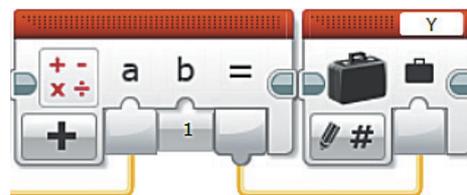
25. Соедини выход переменной **Y** с входом **a**, во входе **b** оставь **1**.



26. Полученное значение нужно перезаписать в переменную **Y**. Добавь команду **Переменная** (красный блок) и для обработки установи переменную **Y**.



27. Соедини выход «=» команды **Математика** с входом переменной **Y**.



Всё почти готово, но нужно ещё предусмотреть случай, когда место на экране EV3 кончилось и требуется начать записи в новом столбике. Этот фрагмент программы мы объединим в новую подпрограмму, поскольку он нам понадобится ещё раз.

Такие единообразные кусочки программы всегда удобно объединять в отдельные подпрограммы для экономии времени и рабочего пространства.

28. Перед тем как создать подпрограмму, нужно сначала добавить хотя бы одну команду из неё. У тебя это будет команда **Переменная** (красный блок).



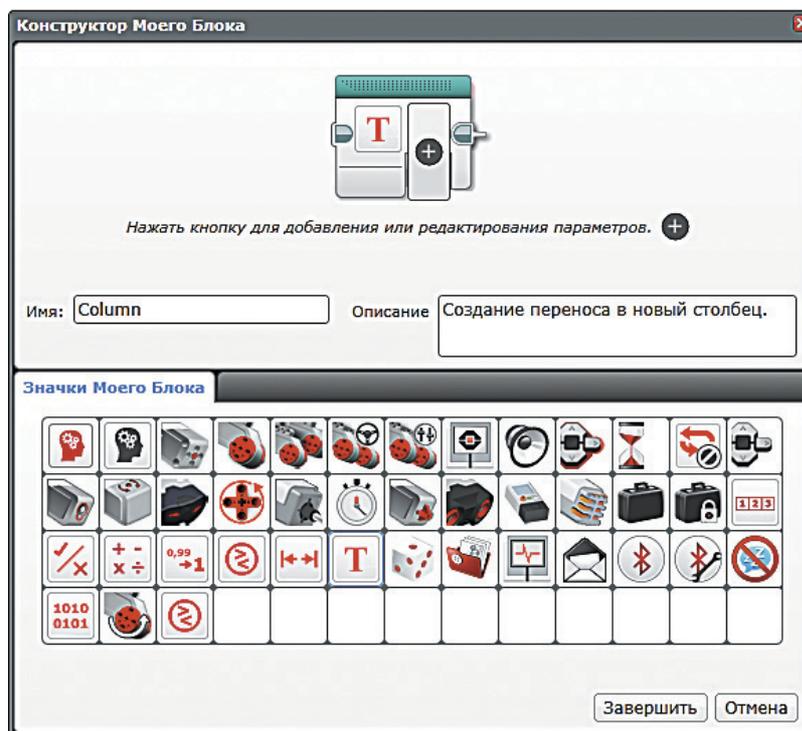
29. Когда ты добавляешь новую команду, она выделена голубым контуром (то есть активна). Если этого не произошло, кликни по данной команде мышью, и она активируется.

Теперь надо поместить эту команду в подпрограмму, то есть создать **свой блок**. Для этого в верхней части окна программы выбери меню **Инструменты** → **Конструктор Моего блока**.

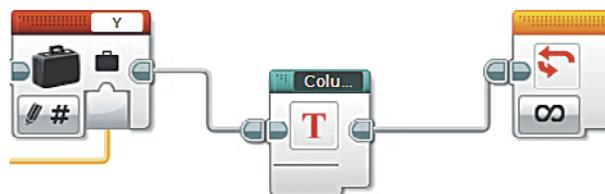
Инструменты	Справка
Редактор звука	
Редактор изображения	
<b>Конструктор Моего Блока</b>	
Обновление встроенного ПО	
Настройка беспроводного подключения	
Мастер импорта блоков	
Загрузить как приложение	
Обозреватель памяти	Ctrl+I
Диспетчер файлов журнала данных	Ctrl+U
Удалить значения из набора данных	
Создать программу регистрации	
Экспортировать наборы данных	
Импортировать программу модуля	

30. В открывшемся окне задай имя подпрограммы, напиши её краткое описание и выбери подходящую пиктограмму (маленькое изображение). Логично задать имя, например, **Column** (от англ. — «Столбец»). Помни, чтобы легко ориентироваться в программах, всегда следует называть их сообразно тому, какую функцию они выполняют. Затем нажми кнопку **Завершить**.

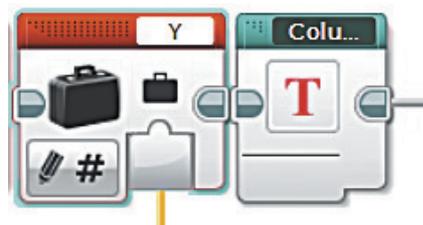
Краткое описание и пиктограмму можешь задать, как на рисунке или самостоятельно.



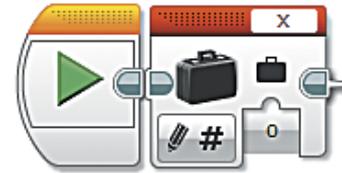
31. В основной программе моментально появится новая команда **Column**.



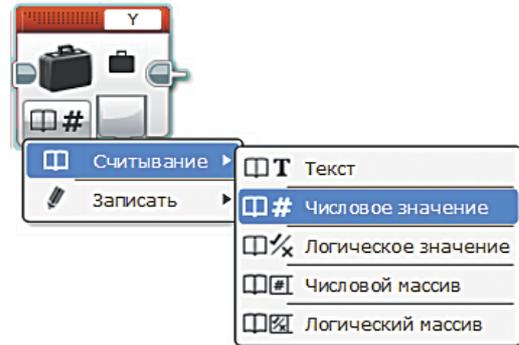
32. Для удобства работы можно убрать расстояние между этими командами, кликнув по выходной связке команды **Переменная**.



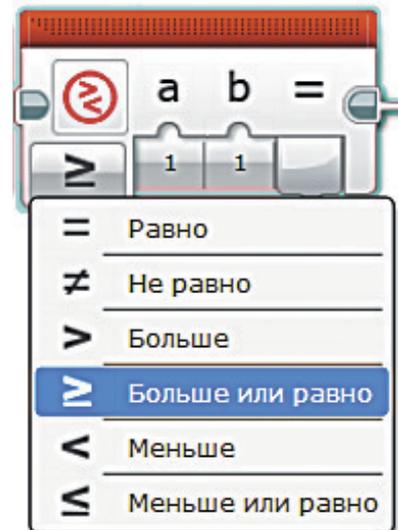
33. Теперь дважды кликни по созданной команде **Column**. На экране откроется новое рабочее поле для редактирования подпрограммы. Более того, сверху появится специальная вкладка с именем **Column**.



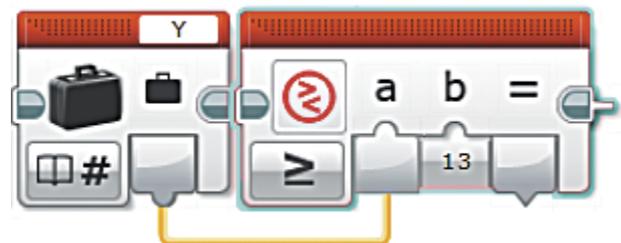
34. Теперь приступим непосредственно к самой подпрограмме. В уже добавленной команде **Переменная** выбери опцию **Считывание** → **Числовое значение** и установи для обработки переменную **Y**.



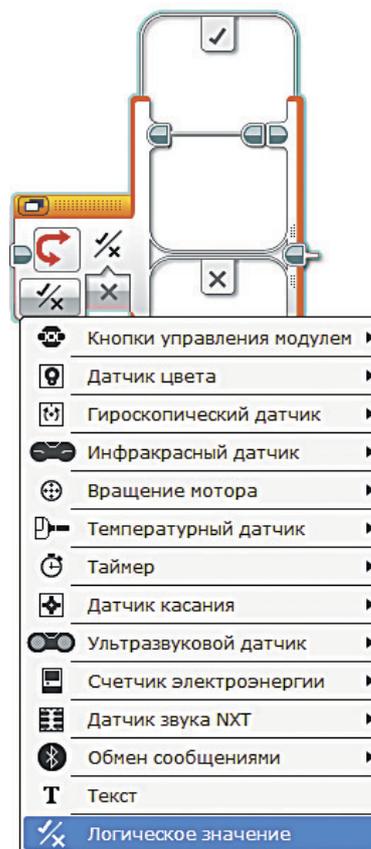
35. Добавь команду **Сравнение** (красный блок) и выбери опцию **Больше или равно**.



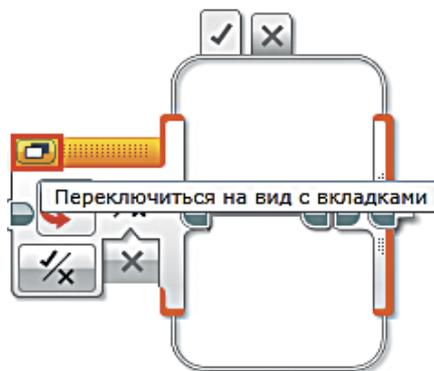
36. Соедини выход переменной **Y** с входом **a**, во вход **b** введи с клавиатуры число **13** — именно столько строк может уместиться на экране EV3.



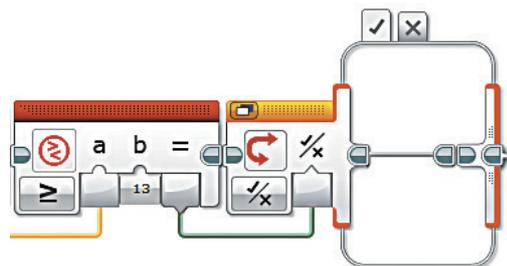
37. Таким образом ты проверяешь следующее условие: «Если количество строк стало равно 13, то...», и дальнейшие действия определяются истинностью или ложностью этого условия. Добавь команду **Переключатель** (оранжевый блок) и выбери опцию **Логическое значение**.



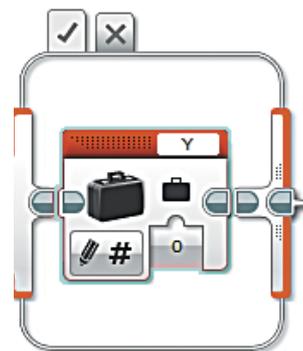
38. Перейди к **виду с вкладками**, нажав окошко в левом верхнем углу команды **Переключатель**.



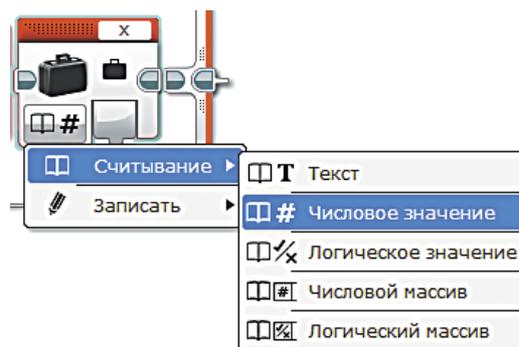
39. Соедини выход «=» команды **Сравнение** с входом **Переключателя**. Тебя будет интересовать только случай истинности проверенного условия, поэтому остальной фрагмент подпрограммы будет содержаться внутри **Переключателя** на вкладке с галочкой (истина).



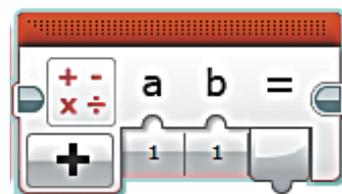
40. Для начала нужно вернуться к вер-ху экрана, то есть обнулить переменную **Y**. Добавь команду **Переменная** (красный блок) и выбери для обработки переменную **Y**.



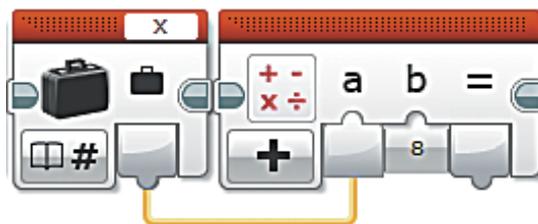
41. Теперь необходимо увеличить горизонтальную координату  $x$  для сдвига вправо. Добавь ещё одну команду **Переменная** (красный блок), выбери опцию **Считывание** → **Числовое значение** и для обработки установи переменную **X**.



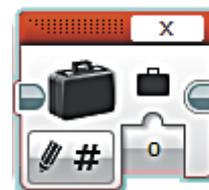
42. Далее добавь команду **Математика** (красный блок).



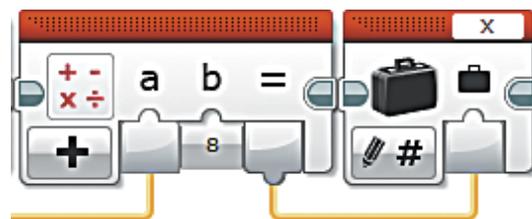
43. Соедини выход переменной **X** с входом **a** команды **Математика**, во вход **b** с клавиатуры введи число **8** (это ширина столбца).



44. Полученное значение нужно перезаписать в переменную **X**. Добавь команду **Переменная** (красный блок) и выбери для обработки переменную **X**.



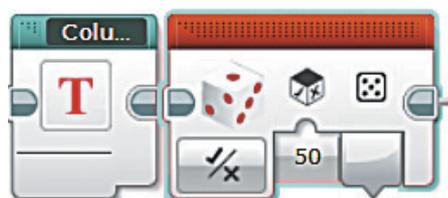
45. Соедини выход «= $\Rightarrow$ » команды **Математика** с входом переменной **X**.



На этом подпрограмма завершена. Вернёмся в основную программу *Finder*. Тебе осталось научить робота-искателя произвольно менять направление движения и останавливаться после выхода из цикла.

Пусть робот будет случайным образом поворачиваться немного влево или вправо.

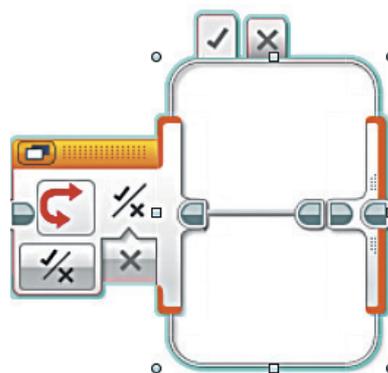
46. Чтобы задать случайное событие, добавь команду **Случайное значение** (красный блок) и выбери опцию **Логическое значение**.



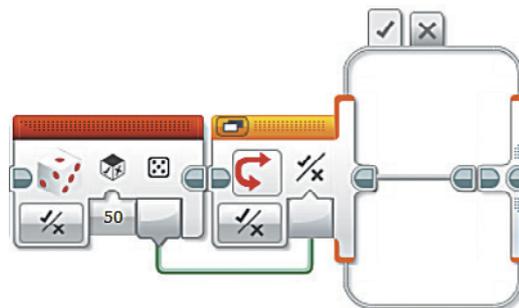
Выбранная опция позволяет с заданной долей вероятности выбирать между двумя событиями, а логическое значение как раз и может быть только «истиной» или «ложью».

Оставь параметр вероятности равным 50 (то есть робот будет поворачивать налево или направо с равной долей вероятности в 50%).

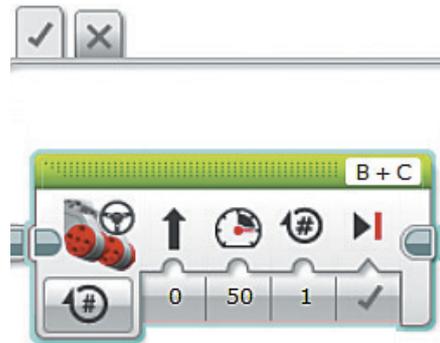
47. Далее добавь команду **Переключатель** (оранжевый блок), выбери опцию **Логическое значение** и перейди к виду с вкладками.



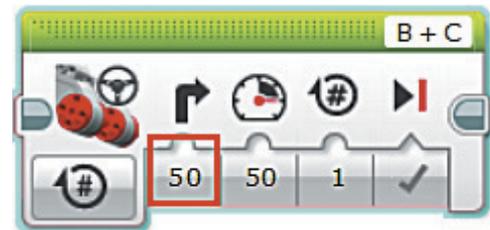
48. Соедини выход команды **Случайное значение** с входом **Переключателя**.



49. На вкладку **Истина** (галочка) добавь команду **Рулевое управление** (зелёный блок).

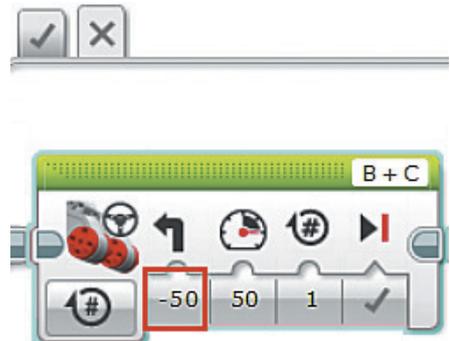


50. Настрой параметр **Рулевое управление** на значение **50**, остальные оставь без изменений.

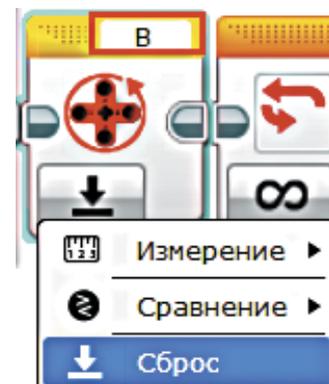


51. Ещё одну команду **Рулевое управление** (зелёный блок) нужно добавить и на вкладку **Ложь** (крестик), только параметр установить равным **-50**.

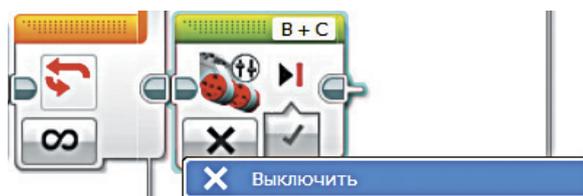
Чтобы начался новый отсчёт расстояния после изменения траектории движения, нужно сбросить показания мотора.



52. После **Переключателя** добавь команду **Вращение мотора** (жёлтый блок), выбери опцию **Сброс** и проверь, чтобы в правом верхнем окошке значился порт **В**.



53. Сразу после **Цикла** добавь команду **Независимое управление моторами** (зелёный блок) и выбери команду **Выключить**.



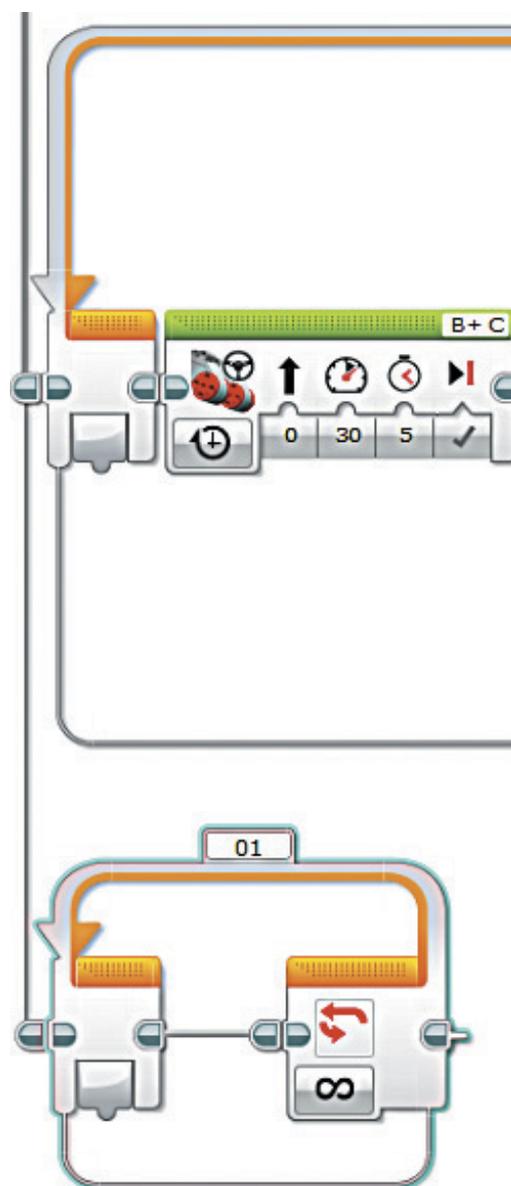
Поздравляем! Крупная ветвь программы завершена! Твой робот-искатель уже умеет ездить и поворачиваться, но это ещё не все его возможности! Вперёд!

#### Ветвь 4. Пластиры! Действия при обнаружении объекта (сокровищ)

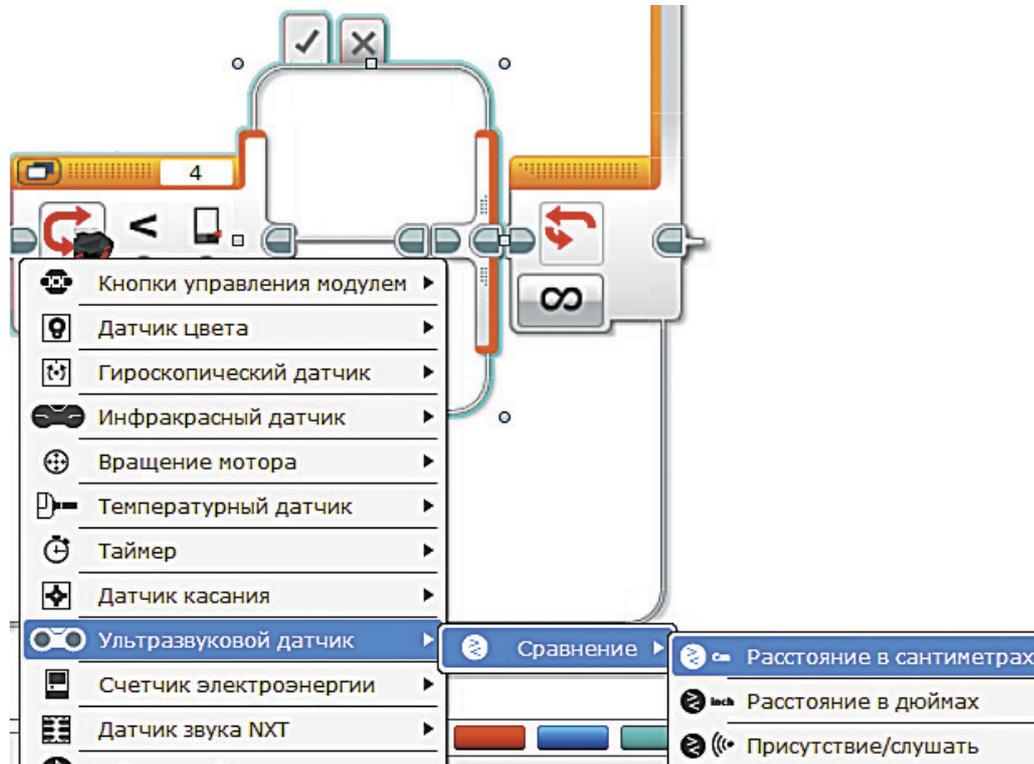
**Логика.** В этой части программы ты организуешь прерывание всех внутренних циклов при обнаружении сокровища (или любого другого объекта). Обнаружением будет считаться попадание в «поле зрения» ультразвукового датчика любого объекта на расстоянии 15 см или ближе.

Как только объект будет обнаружен, робот-искатель запишет его координаты со знаком «!», потом развернётся и продолжит поиски. Поиски он будет продолжать, пока не найдёт пять объектов.

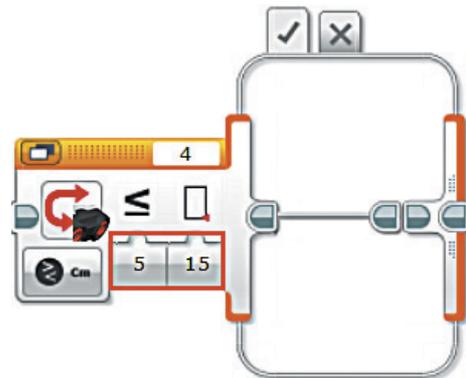
1. Вернись к началу большого цикла и добавь команду **Цикл** (оранжевый блок) ниже двух других. Не забудь соединить его общей связкой от большого цикла.



2. Внутри этого цикла помести команду **Переключатель** (оранжевый блок), выбери опцию **Ультразвуковой датчик** → **Сравнение** → **Расстояние в сантиметрах** и перейди к виду с вкладками.



3. Настрой следующие параметры:
- **тип сравнения:** 5 (меньше или равно);
  - **пороговое значение:** 15.

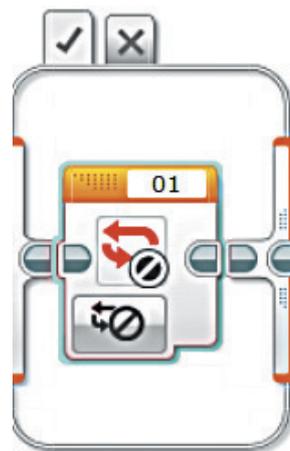


Проверь, чтобы в верхнем правом окошке значился порт № 4.

4. Внутри **Переключателя** помести команду **Прерывание цикла** (оранжевый блок).

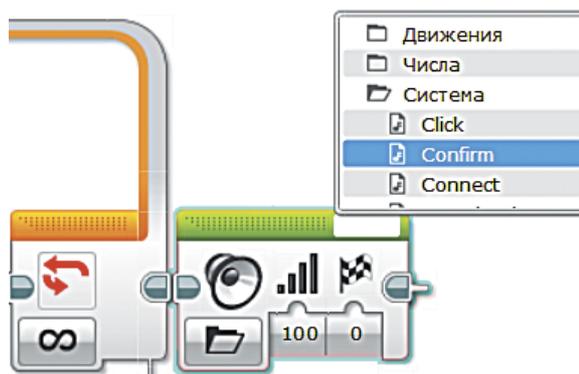
Заметь номер прерывания в правом верхнем окошке — эта команда прервёт все циклы с таким номером, как только робот-искатель обнаружит сокровище.

Ты можешь заметить, что цикл, внутри которого ты поместил эту команду, тоже имеет идентификатор **01**, значит, оставшаяся часть этой ветви будет находиться сразу после этого цикла.



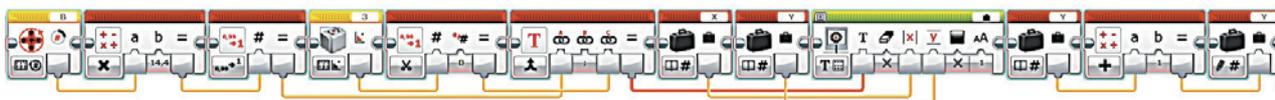
Для начала сделаем так, чтобы робот издавал звук при обнаружении объекта.

5. Добавь команду **Звук** (зелёный блок) и в правом верхнем окошке выбери папку **Звуковые файлы LEGO**, в ней папку **Система** и в ней файл **Confirm** (от англ. «Подтверждение»).

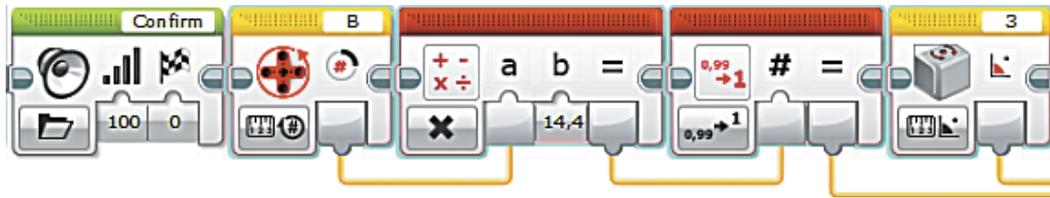


Теперь нужно вывести на экран координаты обнаруженного сокровища — этот фрагмент программы ты уже создал в предыдущей ветви, поэтому тебе нужно его просто скопировать и внести небольшие изменения.

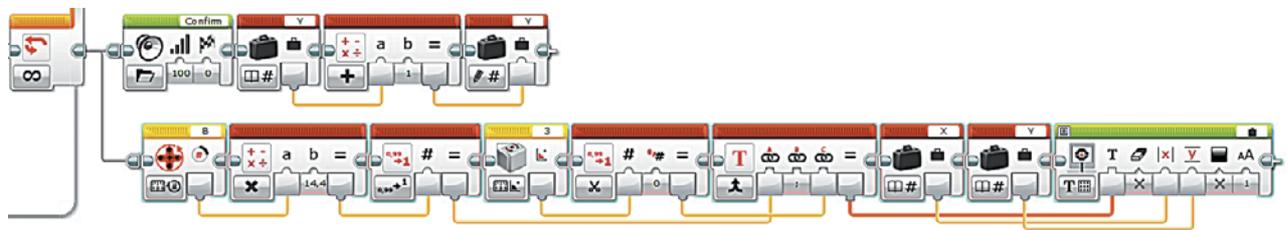
6. В предыдущей ветви найди вторую команду внутри цикла (это будет команда **Вращение мотора**). Удерживай клавишу **Shift** и кликай, начиная с неё, по каждой команде вплоть до подпрограммы **Column**.



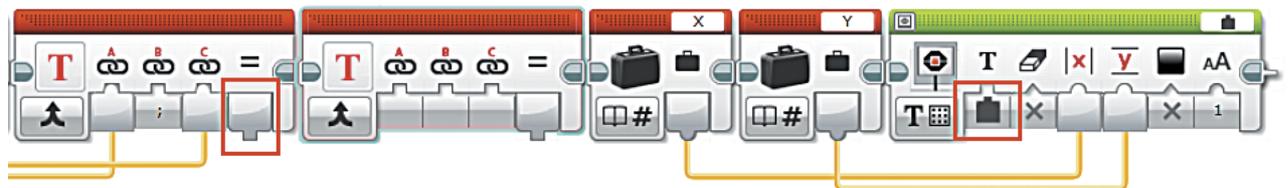
7. Нажми сочетание клавиш **Ctrl+C**, перейди к тому месту, где проигрывается звук и нажми сочетание клавиш **Ctrl+V**.  
В этом месте появится копия фрагмента программы. Присоедини его к команде **Звук**.



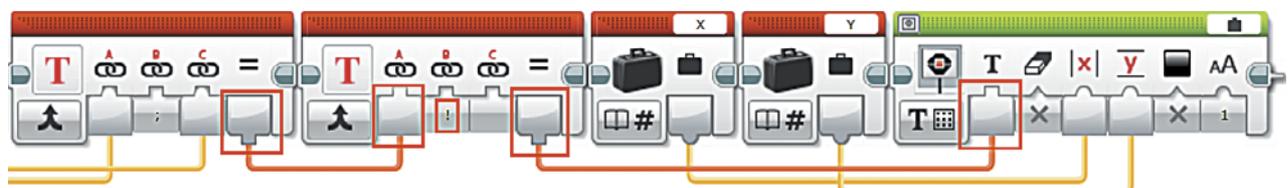
8. Теперь первые девять команд необходимо выделить, перетащить ниже и соединить с помощью выходной связи команды **Цикл**.



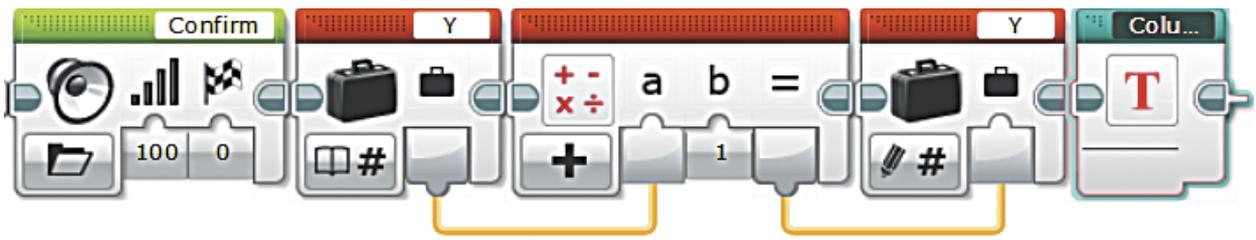
9. В нижней ветви добавь ещё одну команду **Текст** (красный блок) между командой **Текст** и **Переменная** и убери шлейф от команды **Текст** до команды **Экран**.



10. Соедини выход «**=**» первой команды **Текст** с входом **А** второй команды **Текст**; во вход **В** введи с клавиатуры знак «**!**»; вход **С** оставь пустым. Затем соедини выход «**=**» с входом **Текст** команды **Экран**. И вернись к верхней веточке.



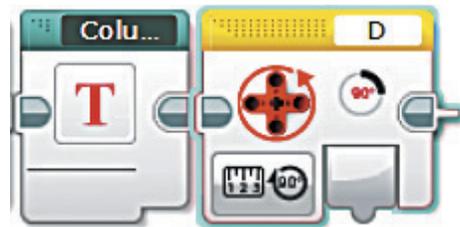
11. Добавь к ней подпрограмму **Column** (голубой блок).



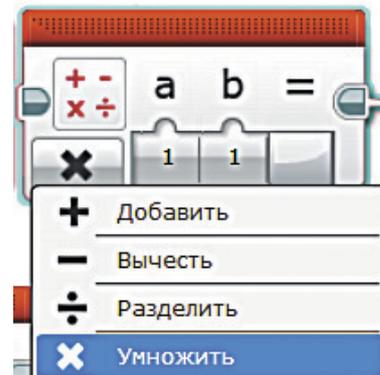
Вспомним, что в этой ветви программы все действия касаются ещё и того, что робот остановился. А значит, в неизвестной позиции остановился и локатор. Его необходимо вернуть в начальную позицию, иначе он выйдет из строя.

Чтобы вернуть локатор в исходное положение, нужно измерить, на сколько градусов он повернут в данный момент времени, и развернуть его на столько же градусов в обратную сторону. Для этого нужно *инвертировать* показание среднего мотора (умножить его на  $-1$ ).

12. Добавь команду **Вращение мотора** (жёлтый блок). Проверь, чтобы в левом верхнем окошке значился порт **D**.

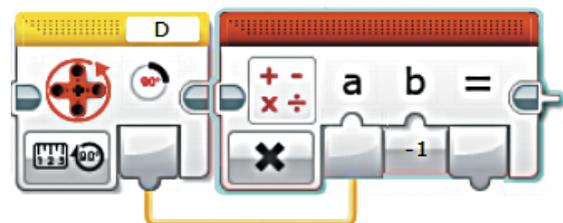


13. Далее добавь команду **Математика** (красный блок) и выбери опцию **Умножить**.

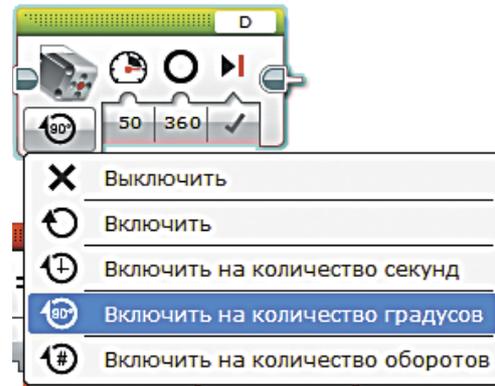


14. Соедини выход команды **Вращение мотора** с входом **a** команды **Математика**, во вход **b** с клавиатуры введи число  $-1$ .

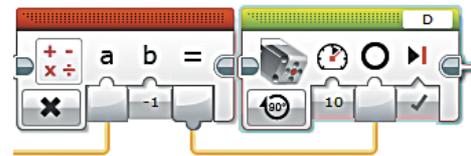
Теперь привод среднего мотора нужно повернуть на рассчитанную величину.



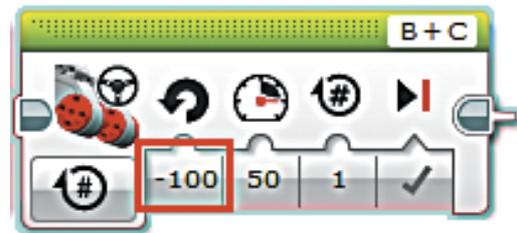
15. Добавь команду **Средний мотор** (зелёный блок) и выбери опцию **Включить на количество градусов**.



16. Соедини выход « $\Rightarrow$ » команды **Математика** с входом **Градусы**. Во вход **Мощность** введи с клавиатуры число 10.



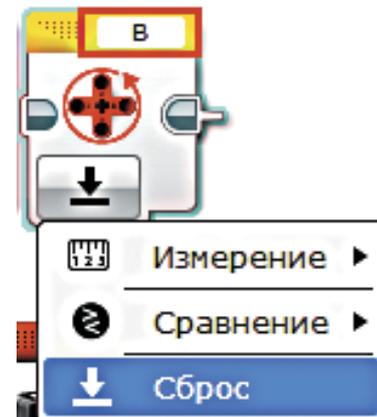
17. Теперь развернём нашего робота. Для этого добавь команду **Рулевое управление** (зелёный блок) и задай параметр **Рулевое управление** равным  $-100$ .



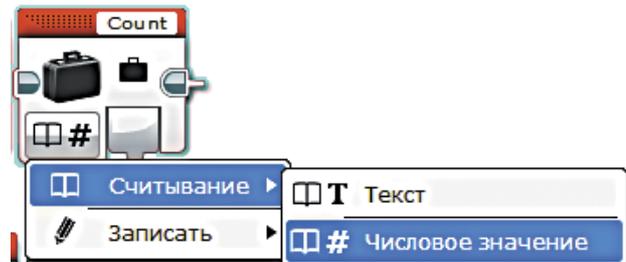
После того, как все расчёты завершены и робот развернулся, нужно, чтобы он начал отмерять расстояние с этой точки, а значит, пора обнулить показания энкодера мотора.

18. Добавь команду **Вращение мотора** (жёлтый блок), выбери опцию **Сброс** и проверь, чтобы в правом верхнем окошке значился порт **В**.

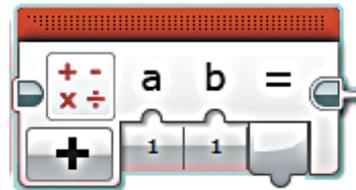
Осталось сделать так, чтобы робот-исследователь прекращал свои поиски после обнаружения пяти объектов (сокровищ). За это в твоей программе отвечает переменная **Count**.



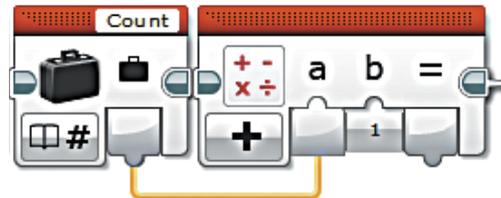
19. Добавь команду **Переменная** (красный блок), выбери опцию **Считывание** → **Числовое значение** и установи для обработки переменную **Count**.



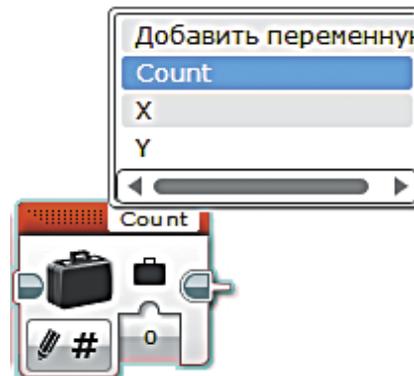
20. Далее добавь команду **Математика** (красный блок).



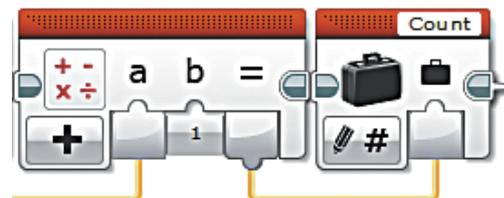
21. Соедини выход переменной **Count** с входом **a** команды **Математика**, во входе **b** оставь значение, равное 1.



22. Теперь добавь команду **Переменная** (красный блок) и выбери для обработки переменную **Count**.

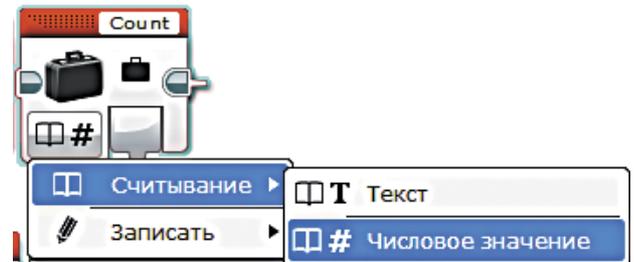


23. Соедини выход «= $\Rightarrow$ » команды **Математика** со входом переменной **Count**.



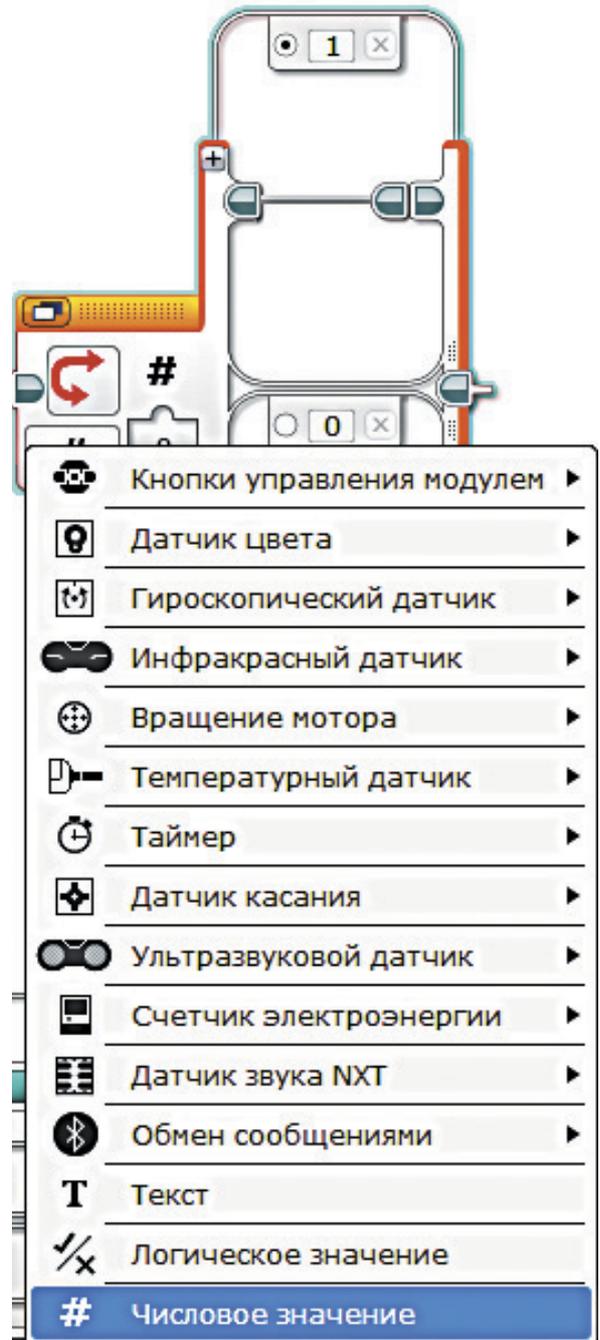
Таким образом, **Count** — это переменная-счётчик. После каждого её изменения нужно проверить — а не стала ли она равна пяти?

24. Добавь команду **Переменная** (*красный блок*), выбери опцию **Считывание** → **Числовое значение** и установи для обработки переменную **Count**.



25. Добавь команду **Переключатель** (*оранжевый блок*) и выбери опцию **Числовое значение**.

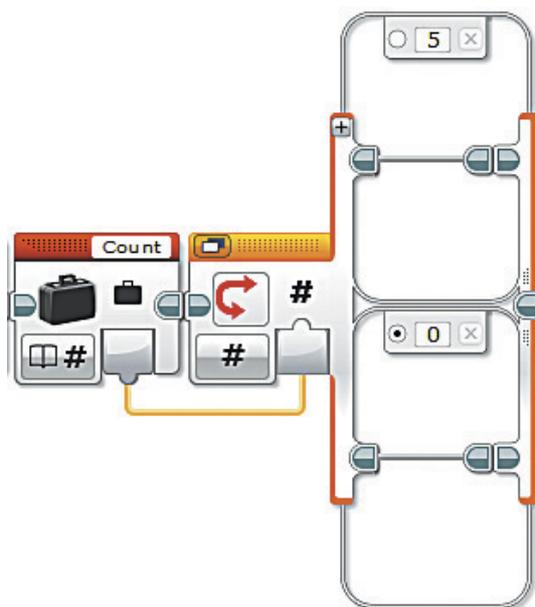
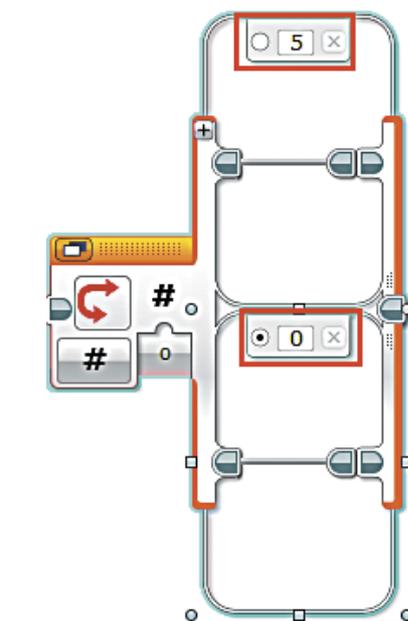
Этот переключатель будет проверять числовое значение переменной и выполнять действие в зависимости от того, чему она равна.



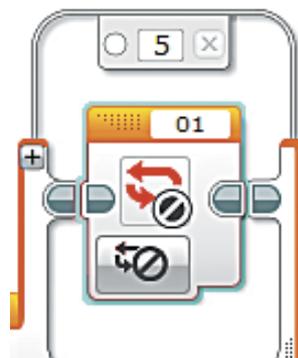
26. Для начала в верхней ветви действий измени число на **1** на **5** и переставь **точку значения по умолчанию** в нижнюю ветвь к числу **0**.

Это означает, что для значения «пять» ты пропишешь особые действия. Если переменная ещё не достигла этого значения, то будут выполняться действия из нижней ветви «по умолчанию» — её ты оставишь пустой.

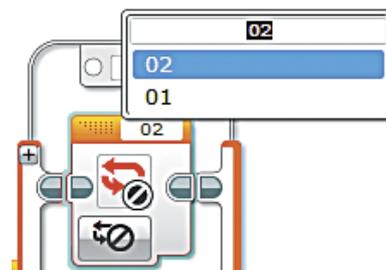
27. Теперь соедини выход переменной **Count** с входом **Переключателя**.



28. В верхней ветви нужно добавить всего лишь одну команду **Прерывание цикла** (оранжевый блок).



29. Однако на этот раз нужно прерывать уже весь большой цикл. Поэтому в правом верхнем окошке выбери идентификатор **02**.



На этом все сложности заканчиваются. Остаётся лишь довести твоего робота до совершенства!

### Ветвь 5. В трёх соснах. Исключение ошибки при полном развороте

**Логика.** Суть работы гироскопического датчика в том, что он измеряет угол отклонения от своего начального положения. Как известно, полный поворот тела вокруг оси — это угол, равный 360 градусам. При этом тело возвращается в первоначальное положение.

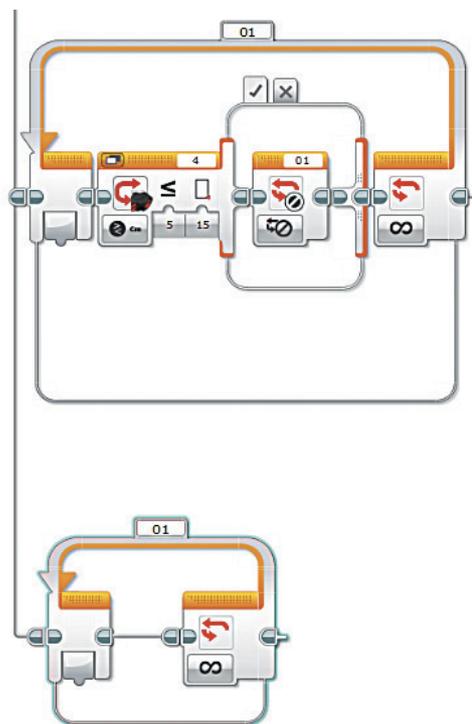
После того, как объект совершил поворот на 360 градусов, он вернулся в нулевое положение.

Именно это мы и должны предусмотреть для гироскопического датчика, так как он не обнуляет автоматически свои показания при полном повороте, а продолжает их прибавлять — так можно и в трёх соснах заблудиться.

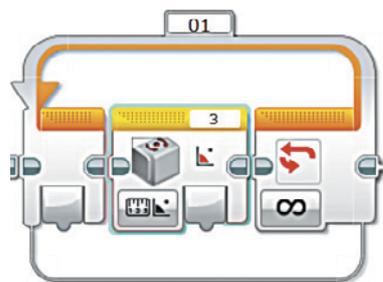
Приступим.

1. Снова вернись в начало большого цикла и добавь ещё одну команду **Цикл** (оранжевый блок) ниже трёх предыдущих. Не забудь соединить его выходной связкой большого цикла.

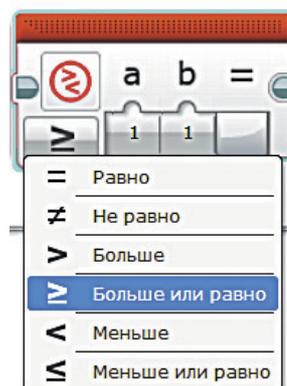
Остальная часть ветви будет внутри этого цикла.



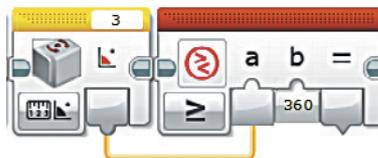
2. Добавь команду **Гироскопический датчик** (жёлтый блок).



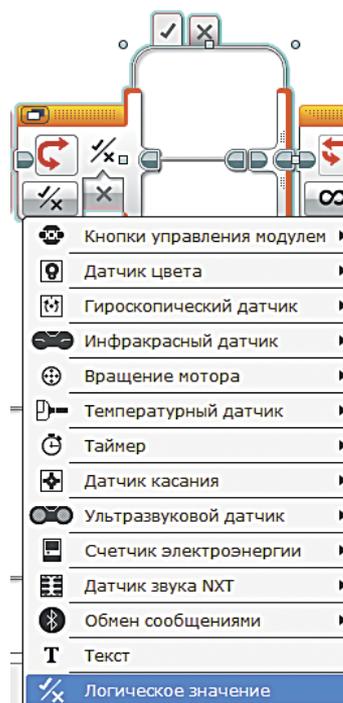
3. Добавь команду **Сравнение** (красный блок) и выбери опцию **Больше или равно**.



4. Соедини выход команды **Гироскопический датчик** с входом **a** команды **Сравнение**, во вход **b** введи с клавиатуры число **360**.

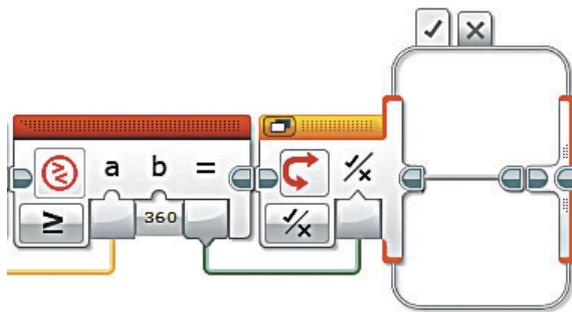


5. Далее добавь команду **Переключатель** (оранжевый блок), выбери опцию **Логическое значение** и перейди к виду с вкладками.



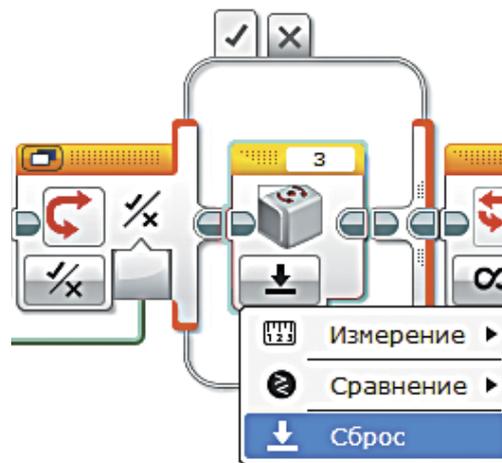
- Соедини выход « $\Rightarrow$ » команды **Сравнение** с входом **Переключателя**.

Таким образом ты проверяешь следующее условие: «Если робот повернулся на 360 градусов, то...». Остаётся прописать всего одно действие в случае истинности данного условия.



- Внутри переключателя добавь команду **Гироскопический датчик** (жёлтый блок) и выбери опцию **Сброс**.

Построй аналогичный фрагмент для случая: «Если робот повернулся на -360 градусов, то...», то есть для разворота в обратную сторону.

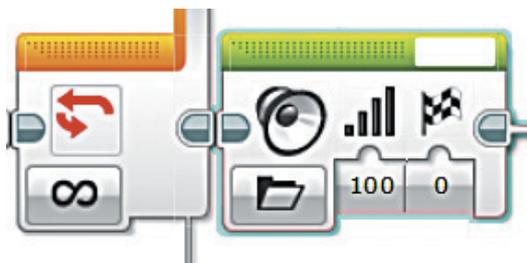


Так держать! Ты проделал большую работу! Остался последний шаг – сделаем его вместе!

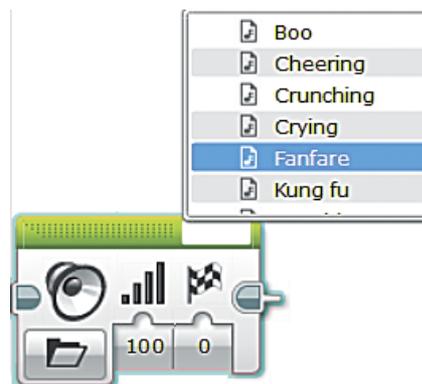
## Ветвь 6. Топография. Конечный цикл работы робота

**Логика.** После того, как большой цикл завершится, необходимо, чтобы робот сохранял своё активное состояние, иначе ты не сможешь прочитать все те координаты, которые он записал у себя на экране.

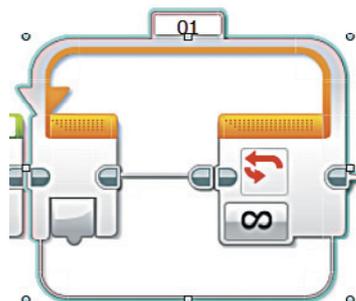
- Перейди в самый конец программы, где заканчивается большой цикл, и добавь для начала команду **Звук** (зелёный блок). Пусть робот воспроизведёт победный сигнал, означающий, что его работа завершена.



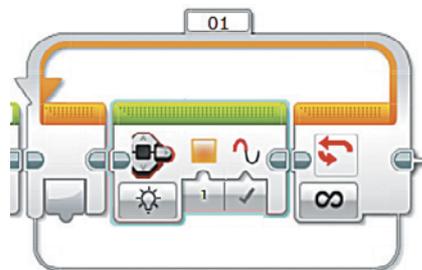
2. Выбери в правом верхнем окошке папку **Звуковые файлы LEGO**, в ней папку **Выражения** и в ней файл **Fanfare**.



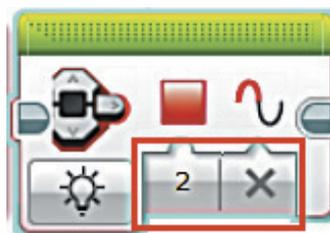
3. Далее добавь команду **Цикл** (оранжевый блок). Какой у этого цикла идентификатор — уже неважно. Сделаем так, чтобы робот по завершении поисков включал красную лампочку и держал её включённой бесконечно.



4. Внутри цикла добавь команду **Индикатор состояния модуля** (зелёный блок).



5. Настрой параметры этой команды:
- **цвет:** 2 (красный);
  - **импульсный:** ложь (иначе лампочка будет мигать).



***Вот и всё! Теперь твой робот готов отправиться на поиски сокровищ!***

**Внимание!** В твоём проекте содержится подпрограмма, поэтому при запуске робота-искателя ты должен загрузить в память EV3 основную программу **Finder** и запустить её.



## 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 локатор «смотрит» строго вперёд.
2. Робот должен начать движение вперёд, а его локатор должен поворачиваться влево и вправо на протяжении всего движения.
3. Если робот не встретил никакого препятствия в течение пяти секунд, то он должен немного повернуть влево или вправо. При этом на дисплее появятся две координаты: пройденный путь в сантиметрах и полярный угол.
4. Если робот встречает препятствие, он останавливается, издавая звук, и выводит на экран координаты с восклицательным знаком. После этого он разворачивается и продолжает поиски.
5. После того как робот обнаружит пять объектов, он должен остановиться, издать победный звук и включить красную индикацию на программируемом модуле.

#### **Внимание!**

Помни: 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 **TreasureHunter** и в ней программу **Finder**.

## Этап 6. Остров сокровищ



Позови своих друзей и сыграй с ними в «Остров сокровищ»!

### Сценарий игры

- 1) Положи в более-менее свободной от вещей комнате что-нибудь, что будет выполнять роль сокровища, например баночку с печенюшками.
- 2) Запусти в эту комнату робота-искателя для поиска.
- 3) В итоге ты получишь на дисплее EV3 примерно такой набор значений:  
60, 0; 18, 44!; 60, -42; 59, -74; 8, -29!; 60, -112; 27, -77!; 21, -164!; 60, -240;  
9, -200!.

Ты помнишь, что в каждой строке первая координата — это расстояние в сантиметрах, которое проехал робот, а вторая — под каким углом относительно начального направления он двигался.

Восклицательный знак рядом с координатами означает, что робот нашёл предполагаемое сокровище.

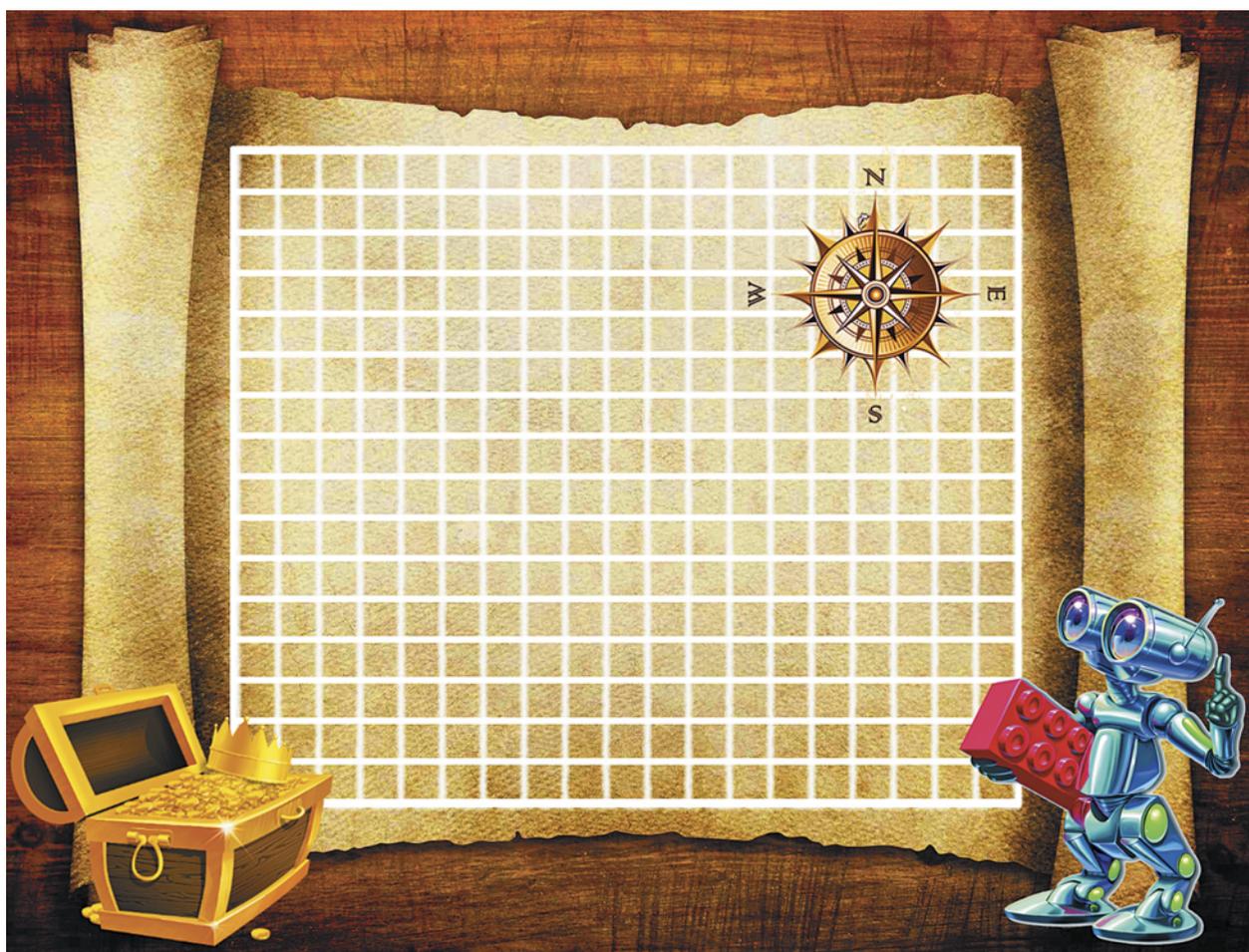
Теперь ты можешь начертить карту сокровищ, пользуясь линейкой для проведения ровных линий и транспортиром для изображения точных углов поворота!



Красные точки указывают на места, где могут находиться заветные сокровища!

*А теперь...*

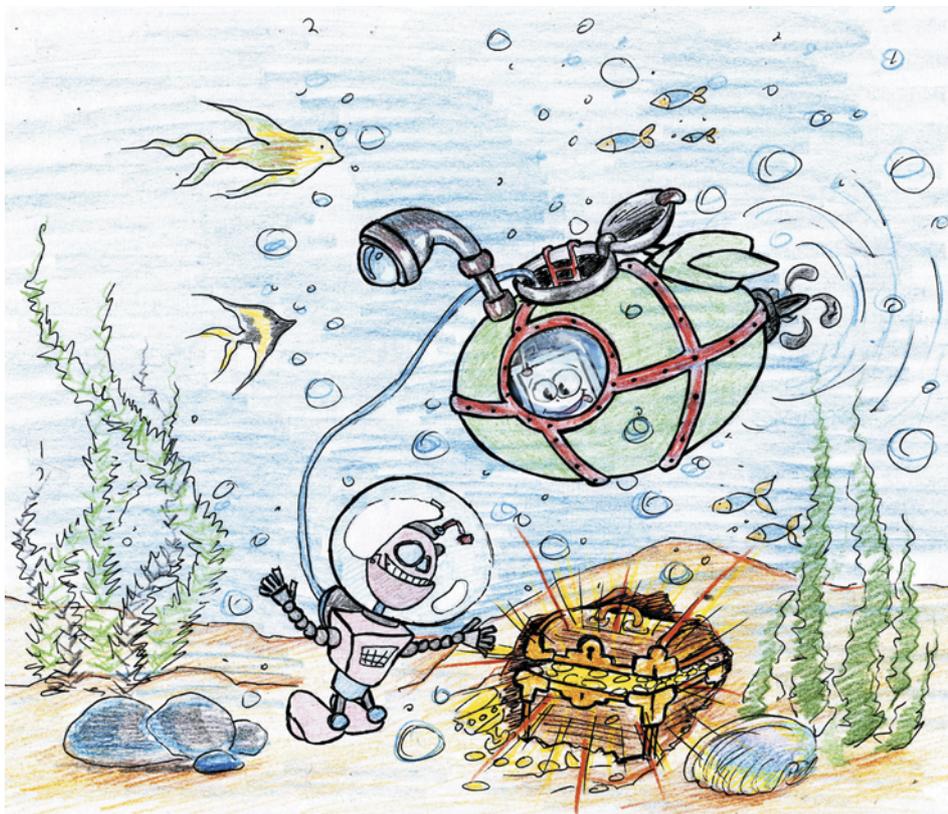
Начерти собственную карту сокровищ!



# До новых встреч!



Ты создал робота своими руками и научился чертить карты, где спрятаны сокровища, пользуясь полярной системой координат. Но впереди ещё так много интересного! В серии книг «РОБОФИШКИ» ты можешь познакомиться с другими замечательными проектами и стать настоящим изобретателем!



# Содержание

<b>Здравствуйте!</b> .....	<b>3</b>
<b>Дорогой друг!</b> .....	<b>4</b>
<b>История эхолокации</b> .....	<b>5</b>
<b>Этап 1. Устройство робота-искателя</b> .....	<b>9</b>
<b>Этап 2. Сборка охотника за сокровищами</b> .....	<b>10</b>
Шаг 1. Сборка шасси .....	<b>10</b>
Шаг 2. Сборка и крепление гусеничных лент .....	<b>14</b>
Шаг 3. Сборка крепления бортового компьютера .....	<b>16</b>
Шаг 4. Сборка и установка локатора .....	<b>19</b>
Шаг 5. Сборка и установка гироскопического датчика .....	<b>21</b>
<b>Этап 3. Установка программного обеспечения на компьютере</b> .....	<b>23</b>
<b>Этап 4. Создание программы для робота-искателя</b> .....	<b>24</b>
Запуск программного обеспечения LME-EV3 .....	<b>24</b>
Создание нового проекта в памяти EV3 .....	<b>24</b>
Логика программы .....	<b>26</b>
Составление программы для робота-искателя .....	<b>26</b>
Ветвь 1. Исходные положения. Объявление необходимых переменных и настройка начальных параметров .....	<b>26</b>
Ветвь 2. Поиск сокровищ. Обеспечение поворотов локатора .....	<b>28</b>
Ветвь 3. Неизведанное. Управление движением робота на плоскости .....	<b>31</b>
Ветвь 4. Пластины! Действия при обнаружении объекта (сокровищ) .....	<b>45</b>
Ветвь 5. В трёх соснах. Исключение ошибки при полном развороте .....	<b>54</b>
Ветвь 6. Топография. Конечный цикл работы робота .....	<b>56</b>
<b>Этап 5. Загрузка программы и её тестирование</b> .....	<b>58</b>
Шаг 1. Загрузка программы в программируемый модуль .....	<b>58</b>
Шаг 2. Тестирование .....	<b>58</b>
<b>Этап 6. Остров сокровищ</b> .....	<b>59</b>
<b>А теперь...</b> .....	<b>60</b>
<b>До новых встреч!</b> .....	<b>61</b>

### Оси

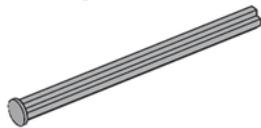
Ось № 6, чёрная, 1x



Ось № 8, чёрная, 2x



Ось № 8 с головкой, тёмно-серая, 2x



### Гусеничный узел

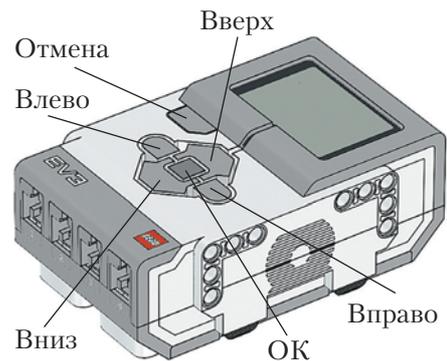
Цепное колесо  
40,7x15 мм, чёрное, 4x



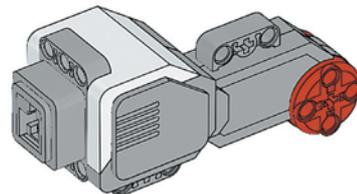
Звено гусеничной ленты,  
чёрное, 54x



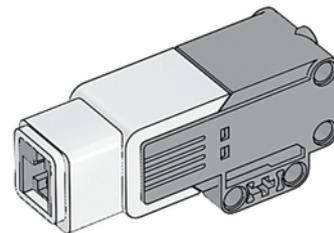
### Программируемый модуль EV3, 1x



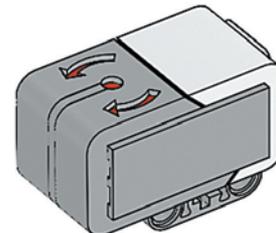
Большой мотор, 2x



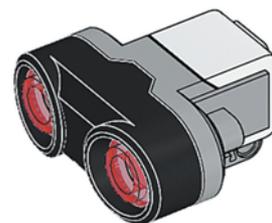
Средний мотор, 1x



Гироскопический датчик, 1x



Ультразвуковой датчик, 1x



### Втулки

Втулка, серая, 4x



### Кабели

Кабель соединительный,  
25 см, 2x;

Кабель соединительный,  
35 см, 2x;

Кабель соединительный,  
50 см, 1x

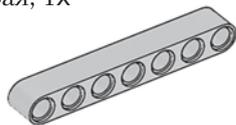


USB-кабель, 1x

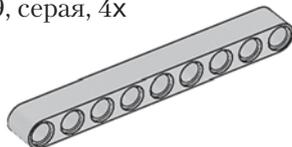


### Балки прямые

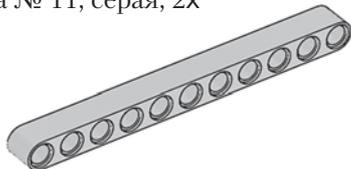
Балка № 7, серая, 1x



Балка № 9, серая, 4x

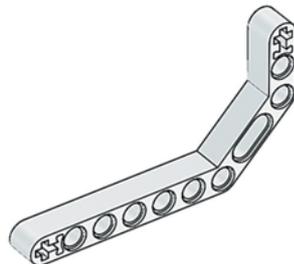


Балка № 11, серая, 2x



### Балки угловые

Двойная угловая балка  
3x7-модульная, белая, 2x

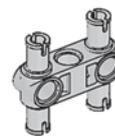


Прямоугольная балка  
2x4-модульная, красная, 1x



### Штифты

H-образный соединительный штифт, 3x3-модульный, серый, 1x



Соединительный штифт с втулкой, 3-модульный, красный, 2x



Соединительный штифт с фрикционной муфтой, 3-модульный, синий, 4x



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

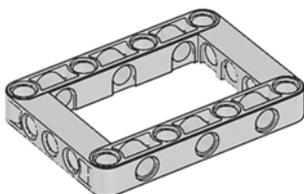


Соединительный штифт с фрикционной муфтой, 2-модульный, чёрный, 27x



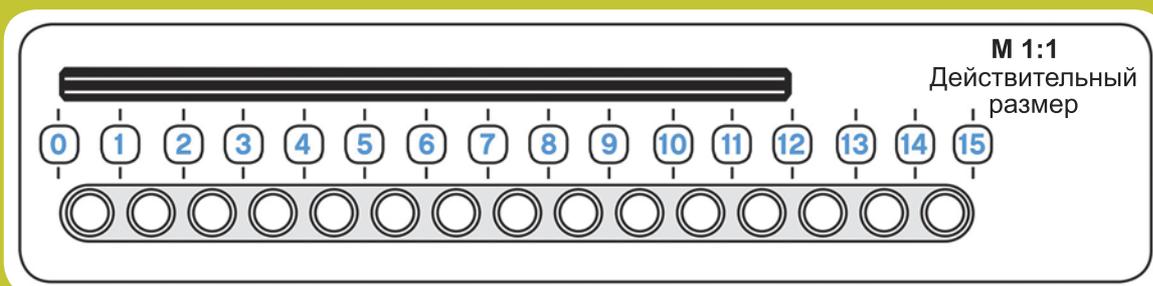
### Рамки

Рамка 5x7-модульная, серая, 2x



### Поперечные блоки

Поперечный блок,  
2-модульный, серый, 2x



*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»

**Рыжая** Елена Ивановна  
**Удалов** Виталий Владиславович

**КОНСТРУИРУЕМ РОБОТОВ НА LEGO® MINDSTORMS® EDUCATION EV3.  
В ПОИСКАХ СОКРОВИЩ**

*Для детей среднего и старшего школьного возраста*

Ведущий редактор *Т. Г. Хохлова*  
Руководители проекта от издательства *А. А. Елизаров, С. В. Гончаренко*  
Научный консультант *Н. Н. Самылкина*  
Ведущий методист *В. В. Тарапата*  
Художники *В. Е. Шкерин, Я. В. Соловцова, И. Е. Марев, Ю. Н. Елисеев*  
Компьютерная верстка: *Е. Г. Ивлева*

Подписано к использованию 10.02.17.

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>

# ЛОВИ НОВЫЕ «РОБОФИШКИ»

на **LEGO®**  
MINDSTORMS®  
Education EV3:

- ◆ «Крутое пике»
- ◆ «Человек – всему мера?»
- ◆ «Секрет ткацкого станка»
- ◆ «Тайный код Сэмюэла Морзе»
- ◆ «Посторонним вход воспрещён!»
- ◆ «В поисках сокровищ» и другие.

С серией **«РОБОФИШКИ»**  
самые удивительные  
и неожиданные идеи  
станут реальностью.

Создай своего робота,  
учись и играй вместе с ним!

Стань настоящим изобретателем!

Не пропусти интересные проекты  
на платформах **Arduino**  
и **ScratchDuino!**

info@pilotLZ.ru  
www.pilotLZ.ru

