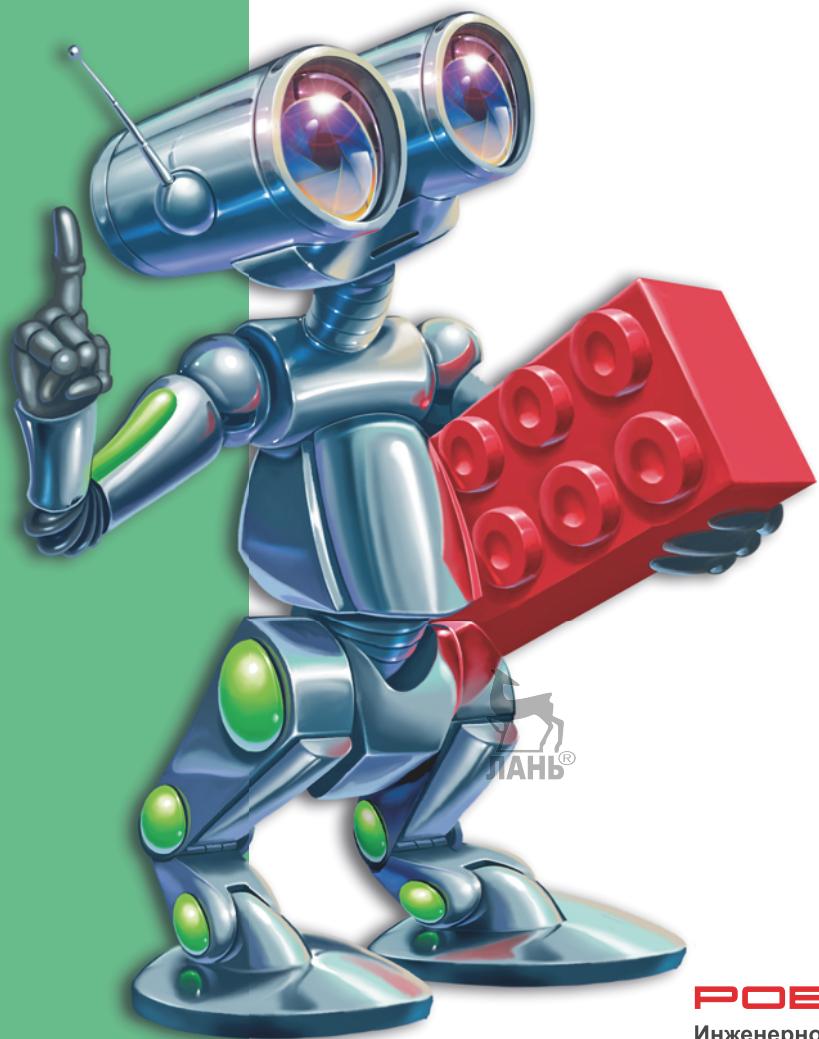


Р • О • Б • О • Ф • И • Ш • К • И

КОНСТРУИРУЕМ РОБОТОВ

на **LEGO[®] MINDSTORMS[®]**
Education EV3

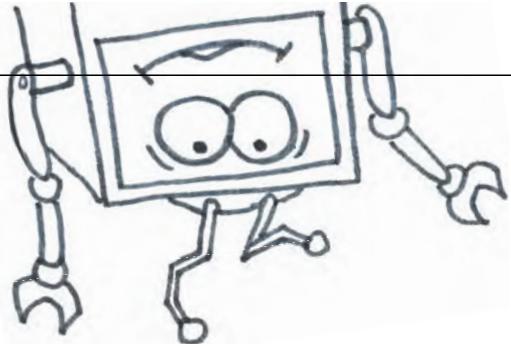


Тайный код
Сэмюэла Морзе

 **Лаборатория
ЗНАНИЙ**

РОБОТОТЕХНИКА

Инженерно-технические кадры инновационной России



В. В. Тарапата

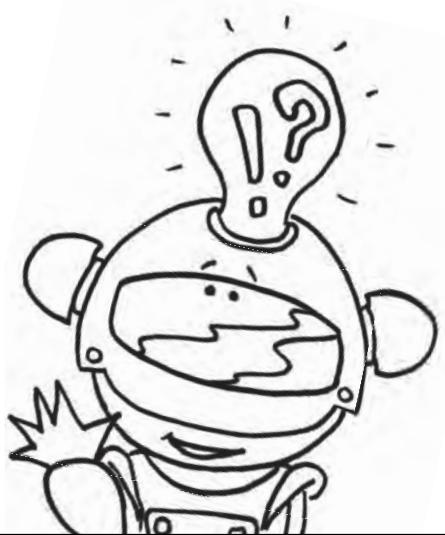
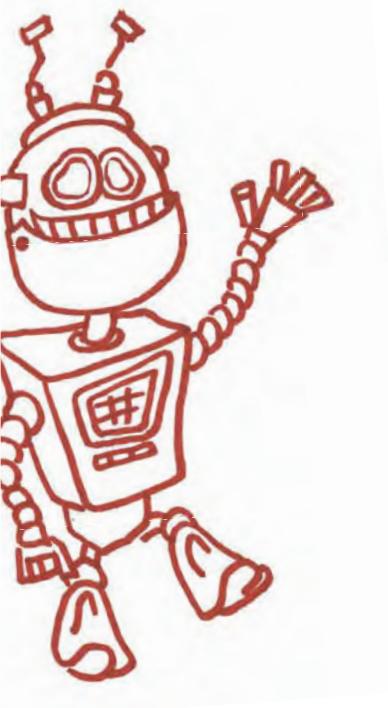
КОНСТРУИРУЕМ РОБОТОВ



на **LEGO[®]** MINDSTORMS[®]
Education EV3

Тайный код
Сэмюэла Морзе

3-е издание,
электронное



Лаборатория знаний
Москва
2021

УДК 373.167
ББК 32.97
Т19

Серия основана в 2016 г.

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



ЛАНЬ®

Т19 Конструируем роботов на LEGO® MINDSTORMS® Education EV3. Тайный код Сэмюэла Морзе / В. В. Тарапата. — 3-е изд., электрон. — М. : Лаборатория знаний, 2021. — 53 с. — (РОБОФИШКИ). — Систем. требования: Adobe Reader XI ; экран 10". — Загл. с титул. экрана. — Текст : электронный.

ISBN 978-5-93208-537-0

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

Вы познакомитесь с историей величайшего изобретения человечества — телеграфом; узнаете о знаменитом способе кодирования информации и сами освоите его. Всего за несколько часов вы соберёте из деталей конструктора LEGO® MINDSTORMS® Education EV3 самый настоящий передатчик, с помощью которого можно общаться с друзьями на тайном языке или посыпать телеграммы.

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

УДК 373.167
ББК 32.97

Деривативное издание на основе печатного аналога: Конструируем роботов на LEGO® MINDSTORMS® Education EV3. Тайный код Сэмюэла Морзе / В. В. Тарапата. — М. : Лаборатория знаний, 2016. — 48 с. : ил. — (РОБОФИШКИ). — ISBN 978-5-00101-021-0.

(12+)

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

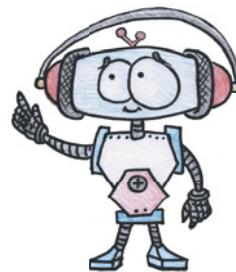


ЛАНЬ®

ISBN 978-5-93208-537-0

© Лаборатория знаний, 2016

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 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 связуете 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.

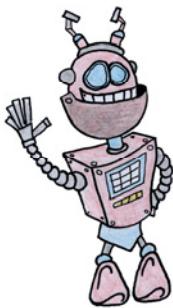
Однако технический прогресс зависит не только от знаний, но и от смелости создавать новое. Всё, что нас окружает сегодня, придумано инженерами. Их любознательность, желание узнавать неизведанное и конструировать то, чего никто до них не делал, и создаёт окружающий мир. Именно от таких людей зависит, 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 дорогой, выбрасывать жалко, а у младшего брата (если он есть) пока другие игрушки. Не расстраивайся! Мы тебе поможем.

Из этого набора можно собрать много интересных и полезных вещей, например кнопочный звуковой передатчик, с помощью которого ты сможешь общаться с другом на тайном языке или посыпать телеграммы одноклассникам.

Задумайся над этим!

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

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

История телеграфа



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

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

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

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

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

И вот, в конце XVIII века появился оптический семафорный телеграф — передающее устройство из подвижных реек, которое устанавливалось на верху башни. Линия телеграфа состояла из цепочки таких башен, расположенных на расстоянии прямой видимости друг от друга (рис. 1). Сигналы сообщения передавались последовательно от башни к башне и представляли собой слова, закодированные на языке семафора. Подобным

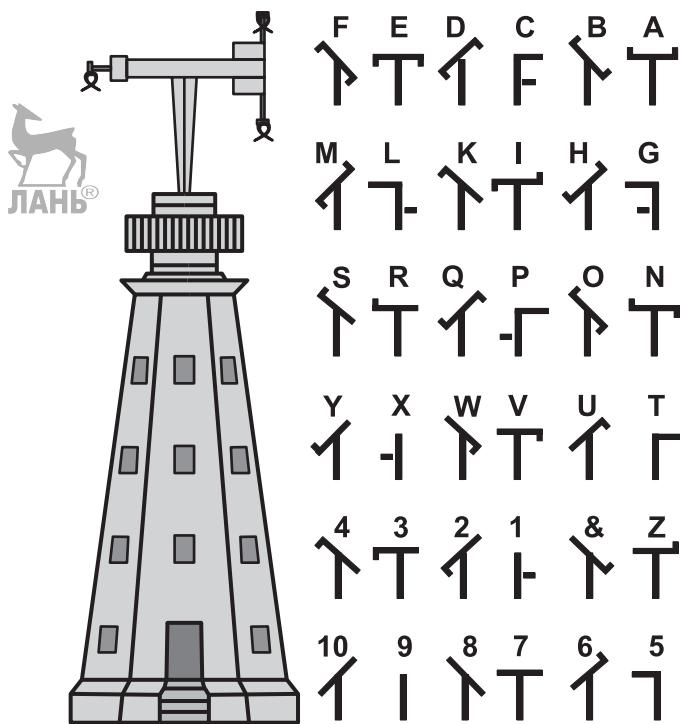


Рис. 1. Семафорный телеграф и его азбука

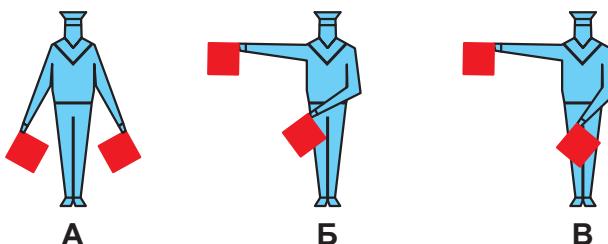


Рис. 2. Передача информации на флоте с помощью флаговых сигналов

языком пользовались и на флоте, только там были сигнальные флаги (рис. 2).

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

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

Но чем сложнее становился окружающий человека мир, тем острее ощущалась потребность в быстром обмене информацией. Конечно, сейчас невозможно представить, что сообщение адресату может идти несколько недель или даже месяцев. Мы так привыкли к электронной почте и мобильной связи. Но тогда это было реальностью. А в 1837 году широкой публике был показан первый **электромагнитный телеграф**, который изменил представление человека о возможности передачи информации на большие расстояния. Не каждый современник знает о принципе действия телеграфа, но

многие слышали словосочетание «азбука Морзе». Изобретателем азбуки и первого электромагнитного телеграфного аппарата стал Сэмюэл Морзе (1791–1872). Он был не знаменитым физиком или известным инженером, а просто — талантливым художником, человеком, который любил мечтать. Во время путешествия из Европы в Америку, Морзе рассказали о «чудесах» электромагнетизма. С этого момента им овладела мечта — создать машину для передачи информации с помощью электричества. После нескольких лет упорного труда, изучения неведомой науки — электротехники, окрылённый великой мечтой, Морзе знакомит мир со своими гениальнейшими изобретениями — телеграфным аппаратом и специальным кодом для передачи сигналов.

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

Кроме передачи сообщений делового характера аппарат Морзе в течение 90 лет передавал сигнал SOS — радиосигнал о помощи терпящим бедствие на море. Этот сигнал состоит из сочетания трёх точек, трёх тире и ещё трёх точек азбуки Морзе.*

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

В 1927 году была установлена единая международная частота для передачи сигналов бедствия на море — 500 кГц, и было строго запрещено использовать её для иных трансляций. С момента ввода

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

в действие сигнала SOS в радиоэфире сорок восемь раз в сутки, то есть каждые полчаса, наступали три минуты молчания. В это время радисты всех стран вслушивались в эфир, не раздастся ли от кого-нибудь призыв о помощи.

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

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

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



Оборудование:

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



Обозначения:

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

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



Этап 1. Изучаем способ кодирования, предложенный Морзе

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

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

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

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

Чтобы научиться передавать закодированные сигналы, тебе потребуется некоторое время. Но ты уже знаешь — чтобы приобрести полезный навык, необходимо упорно тренироваться. Попробуй написать слово «ПРИВЕТ» с помощью азбуки Морзе на листке бумаги. А теперь представь, что ты отбиваешь это слово ключом. Получилось? Если нет, то попробуй снова.



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

Азбука Морзе с добавлением букв кириллического алфавита выглядит так:

Азбука Морзе, таблица кодов Морзе (Морзянка)



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

ЦИФРЫ	Код Морзе
0	— — — — —
1	• — — — —
2	• • — — —
3	• • • — —
4	• • • • —
5	• • • • •
6	— — • • •
7	— — — — •
8	— — — — • •
9	— — — — — •

ЗНАКИ ПУНКТУАЦИИ И СЛУЖЕБНЫЕ СИМВОЛЫ

ЗНАК, СИМВОЛ	Код Морзе
.	• • • • •
,	• — • — — • —
:	— — — — • • •
;	— • — — — • —
!	— — — • — —
?	• • — — — • •
=	— — • • —
«»	• — • • — —
'	• — — — — —
()	— • — — — • —
/ №	— — • • —
—	— — • • • —
—	• • — — — • —
@	• — — — • —
←	• • • • • •

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

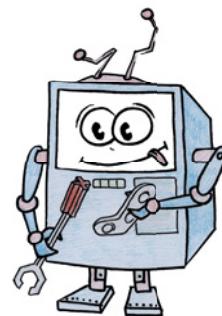
При нажатии кнопки (ключа) будет издаваться звуковой сигнал, названный морзянкой. Принимающая сигнал сторона слушает его и записывает закодированное сообщение в виде «точек» и «тире». Далее сигнал необходимо расшифровать.

Попробуй собрать настоящий передатчик и отправь другу секретное послание!

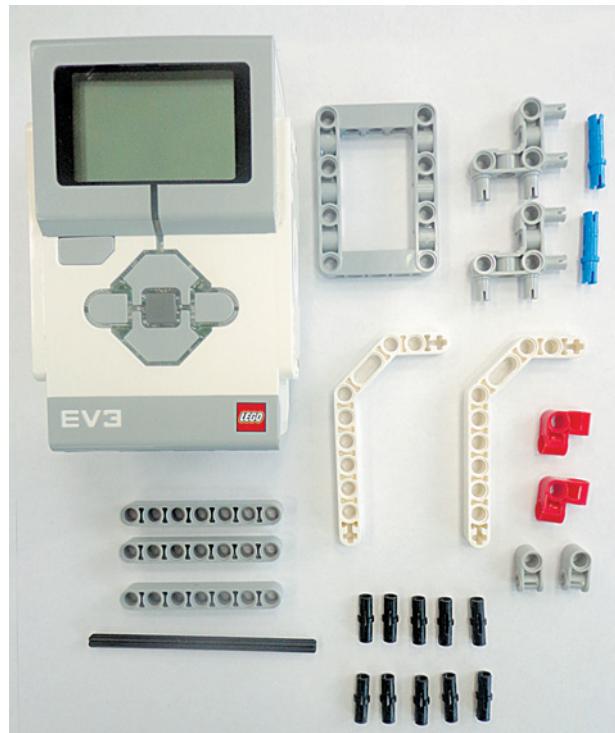
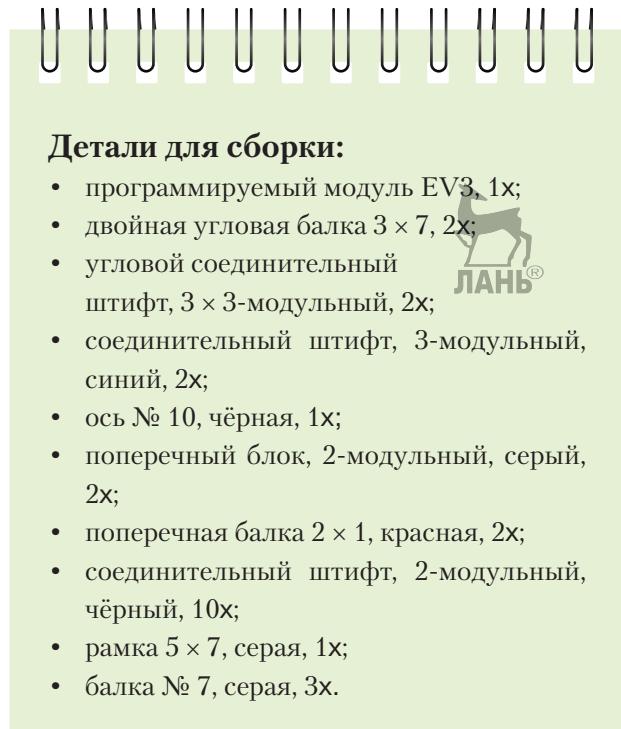


Рис. 3. Модель звукового передатчика

Этап 2. Сборка кнопочного звукового передатчика



ШАГ 1. СБОРКА ПОДСТАВКИ ПОД ПРОГРАММИРУЕМЫЙ МОДУЛЬ



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



2. Вставь штифты в средние отверстия.



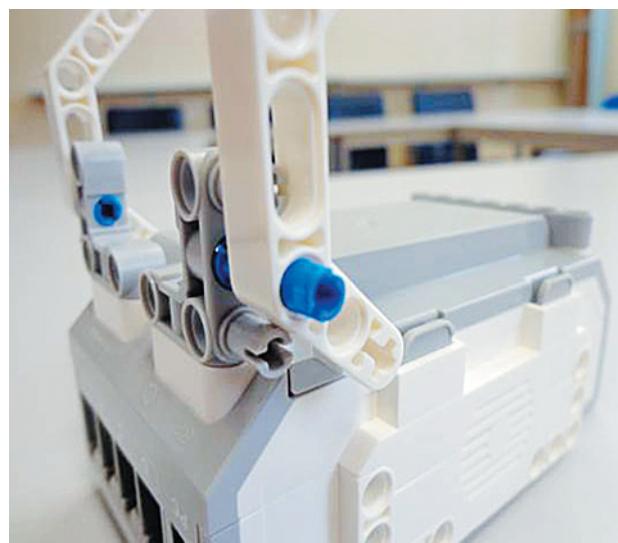
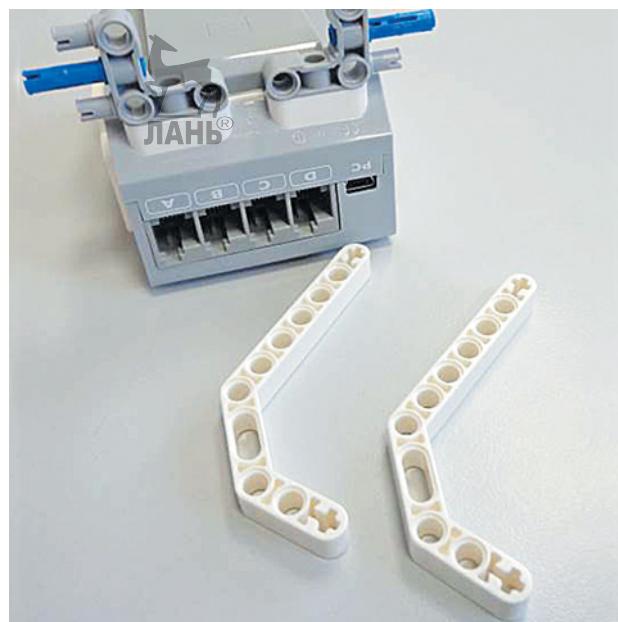
3. Установи балку № 7 на соединительные штифты так, чтобы все отверстия на балке совпали с отверстиями на программируемом модуле.



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



5. С обеих сторон программируемого модуля установи двойные угловые балки 3×7 на синие и угловые штифты. При этом часть углового штифта должна войти в овальное отверстие, а синий штифт — в отверстие, расположенное ниже овального.



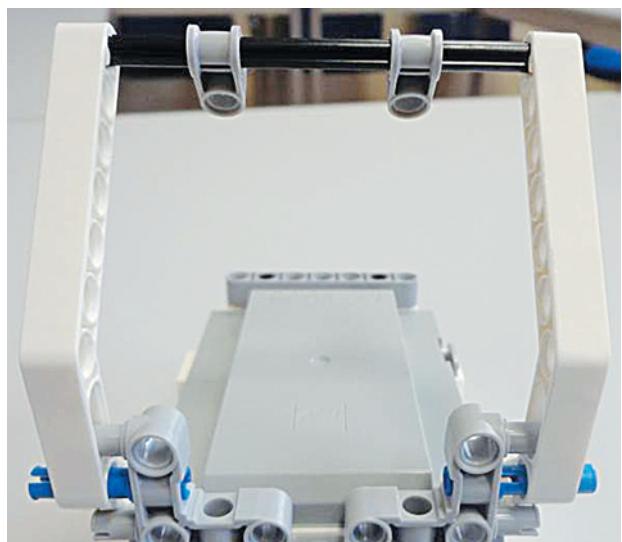
Важно! Балка наденется не до упора!



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



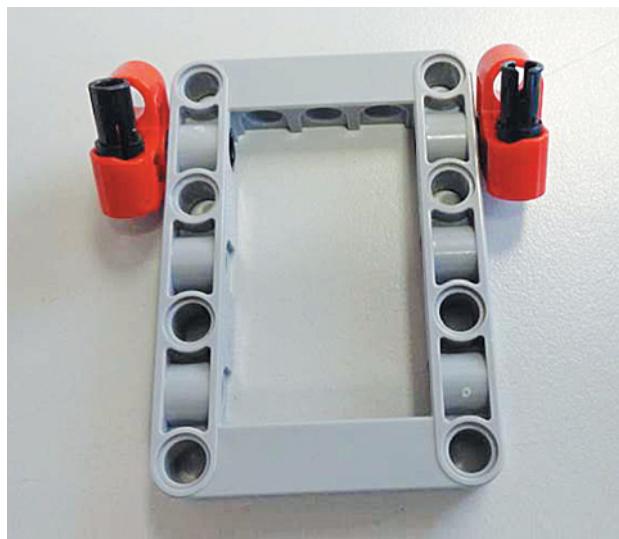
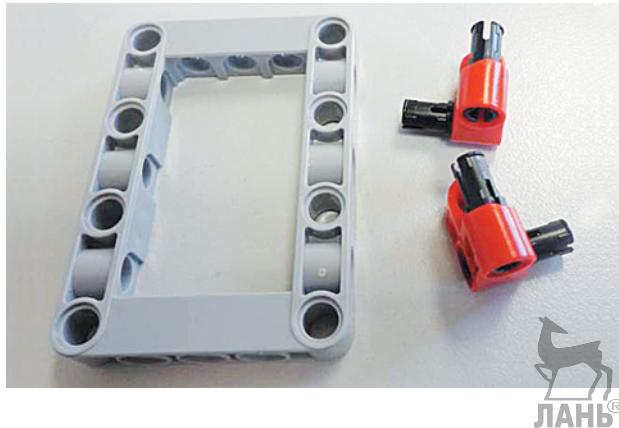
7. Закрепи получившуюся конструкцию в крестовых отверстиях угловых балок так, чтобы блоки «смотрели» на программируемый модуль.



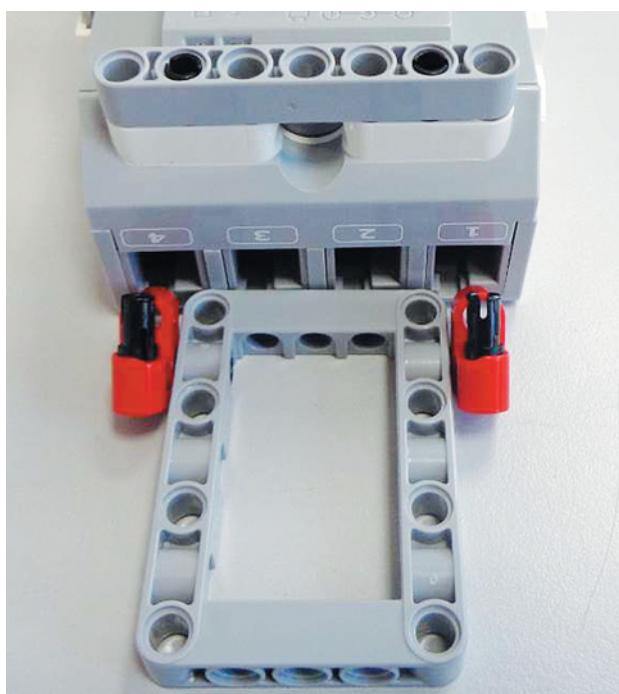
8. Чёрные штифты вставь в красную поперечную балку 2×1 и соедини их. Повтори всё это с другой балкой.
У тебя получилось два держателя.



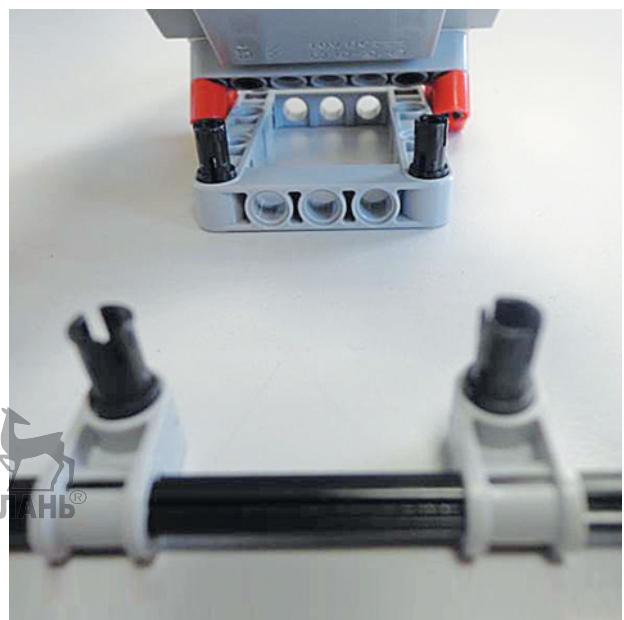
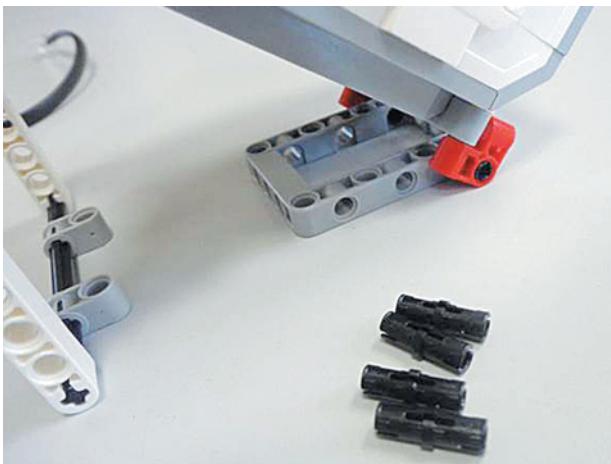
9. Подсоедини держатели к рамке 5 × 7.



10. Получившуюся конструкцию установи на программируемом модуле. Для этого свободные части штифтов вставь в крайние отверстия на балке, которую ты добавил в пункте 3. Теперь твоя конструкция может стоять на поверхности. Однако её необходимо укрепить.

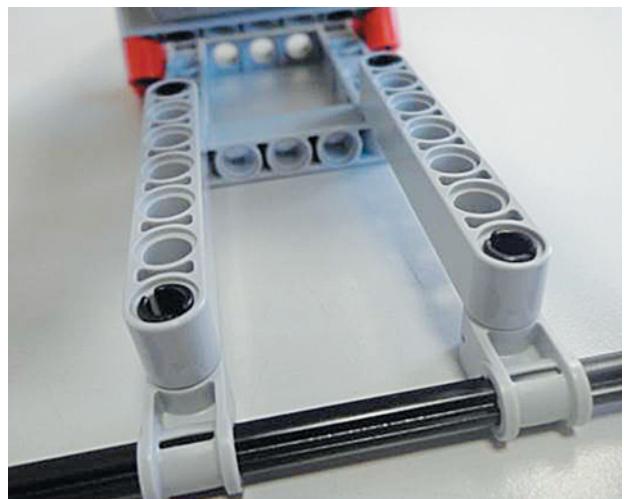
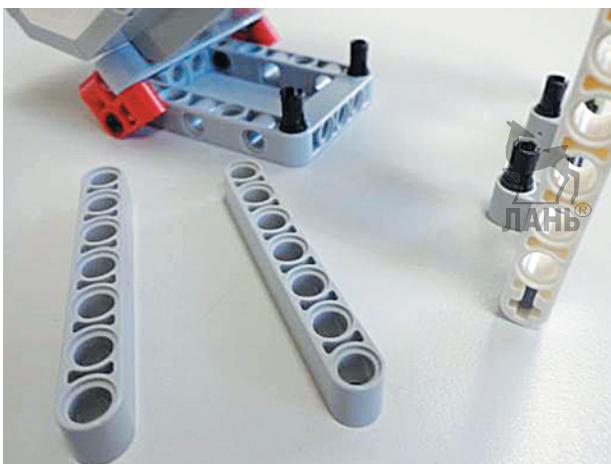


11. Четыре 2-модульных соединительных штифта вставь в отверстия рамки и 2-модульных поперечных блоков.



12. Балки № 7 крайними отверстиями надень на штифты.

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

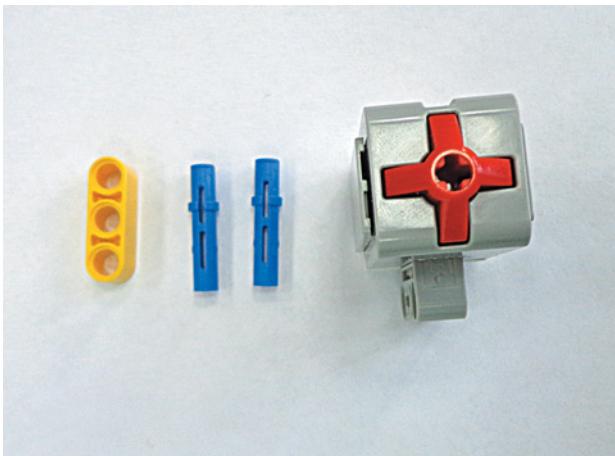


ШАГ 2. СБОРКА И КРЕПЛЕНИЕ КНОПКИ ТЕЛЕГРАФНОГО КЛЮЧА

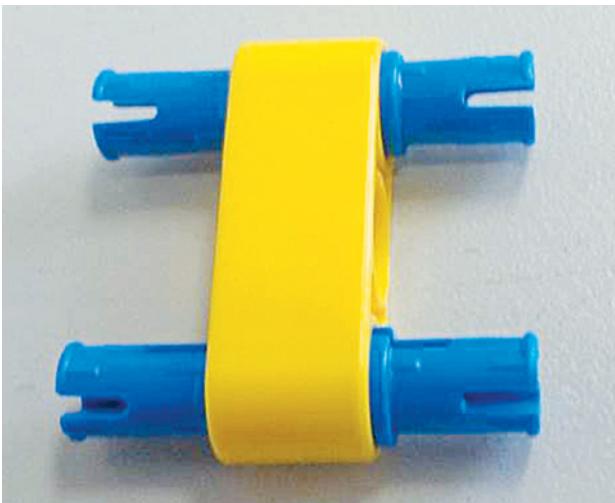
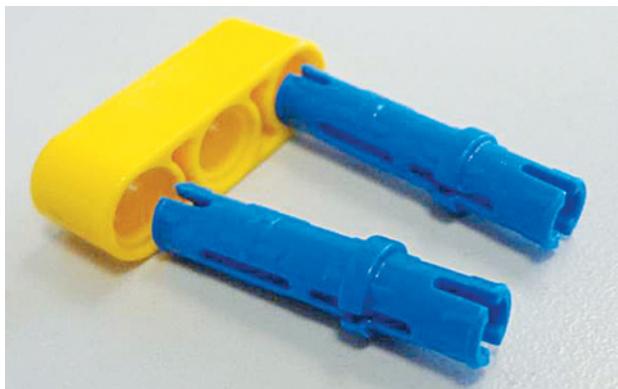


Детали для сборки:

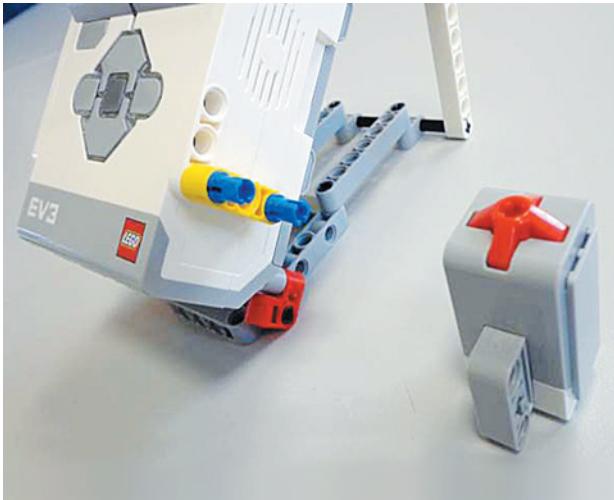
- балка № 3, жёлтая или любого другого цвета, 1x;
- соединительный штифт, 3-модульный, синий, 2x;
- датчик касания, 1x.



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



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



ШАГ 3. СОЕДИНЕНИЕ КНОПКИ С ПРОГРАММИРУЕМЫМ МОДУЛЕМ

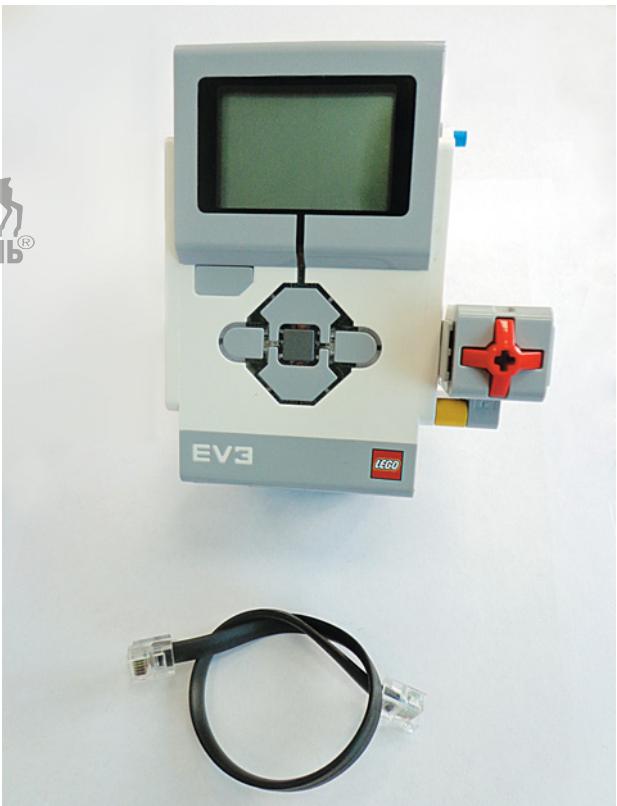


Детали для сборки:

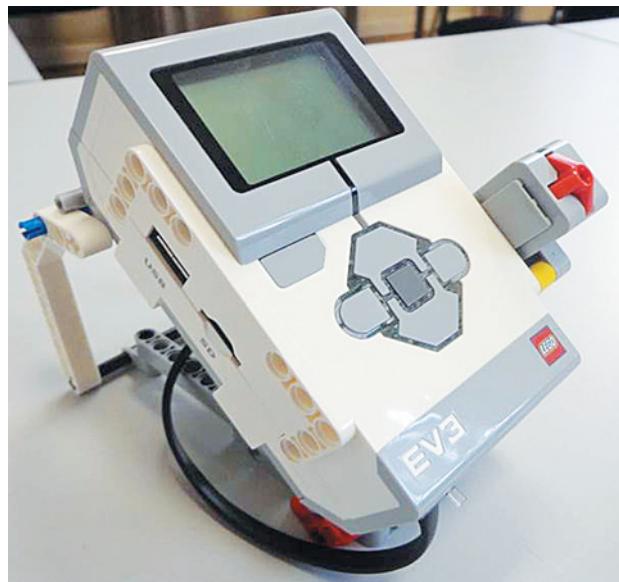
- подготовленный на предыдущих шагах программируемый модуль EV3, 1x;
- кабель, 25 см, 1x.

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

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



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





Этап 3. Установка программного обеспечения на компьютере

- 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.
- Если такой лицензии у тебя нет, зайди на сайт <http://Education.LEGO.com> и перейди в раздел «Техническая поддержка», где ты сможешь скачать установочный файл LME-EV3.



Внимание!

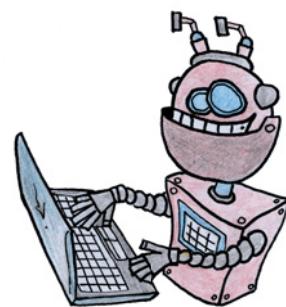
При любых затруднениях с установкой программного обеспечения обращайся в службу технической поддержки компании LEGO® Education по адресу
<http://Education.LEGO.com>

ШАГ 1. ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

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



Этап 4. Пишем программу для передатчика

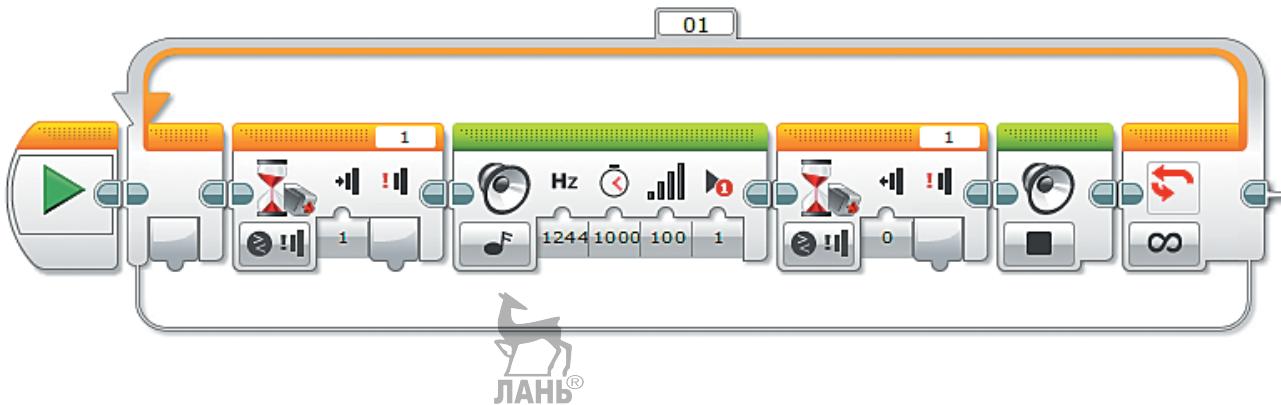


ЛОГИКА ПРОГРАММЫ

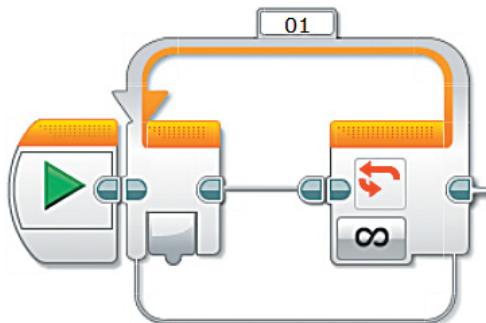
В начальный момент программа находится в режиме ожидания. При нажатии кнопки датчик начинает подавать звуковой сигнал (тон). При отпускании кнопки звуковой сигнал прекращается. Программа возвращается в начало и снова ожидает нажатия кнопки. И так по кругу. Отсюда видна цикличность нашей программы.

ЛАНЬ®

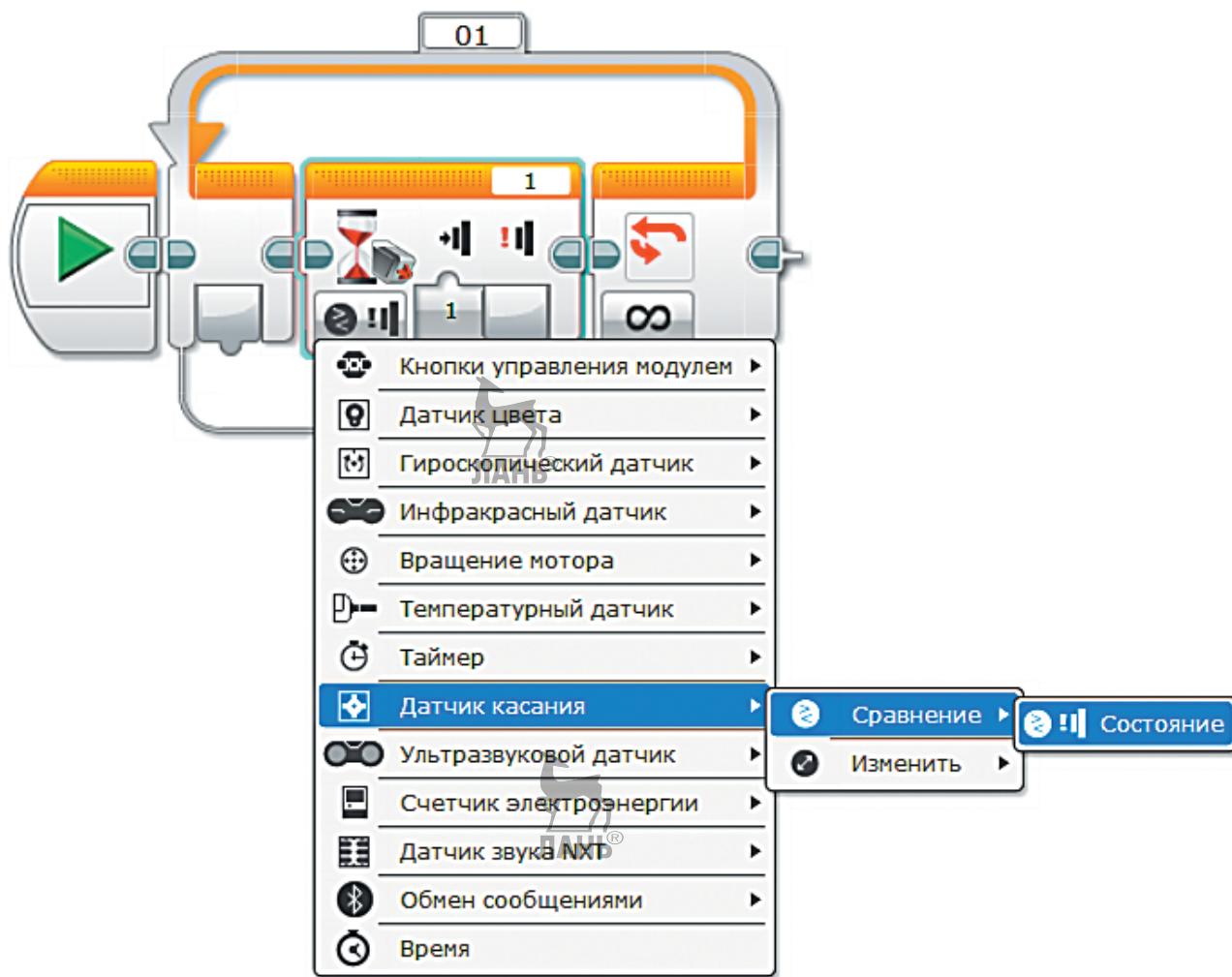
ОБЩИЙ ВИД ПРОГРАММЫ



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

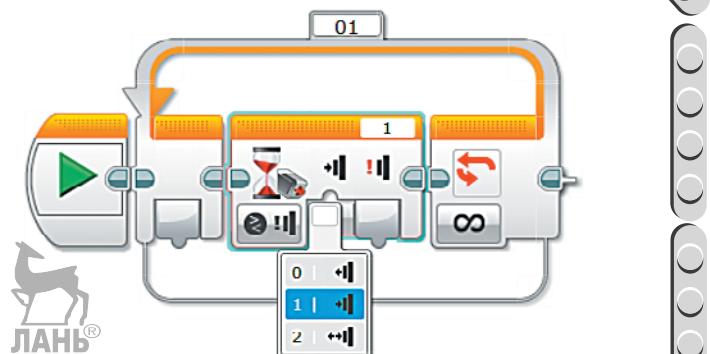


2. Внутрь цикла помести команду ожидания (оранжевый блок) и выбери опцию **Датчик касания** → **Сравнение** → **Состояние**.

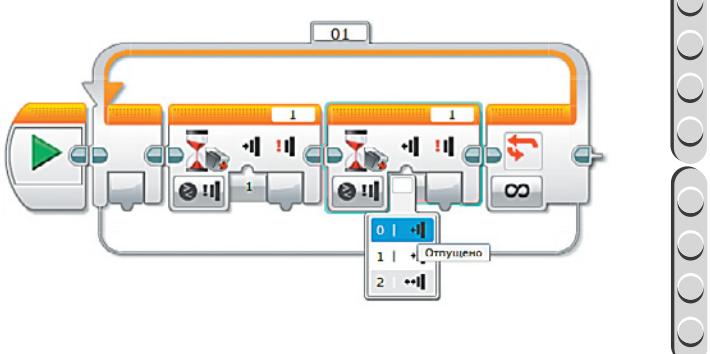


Внимание! Цифра в верхнем правом окошке данного блока должна совпадать с номером порта, к которому подключён датчик касания!

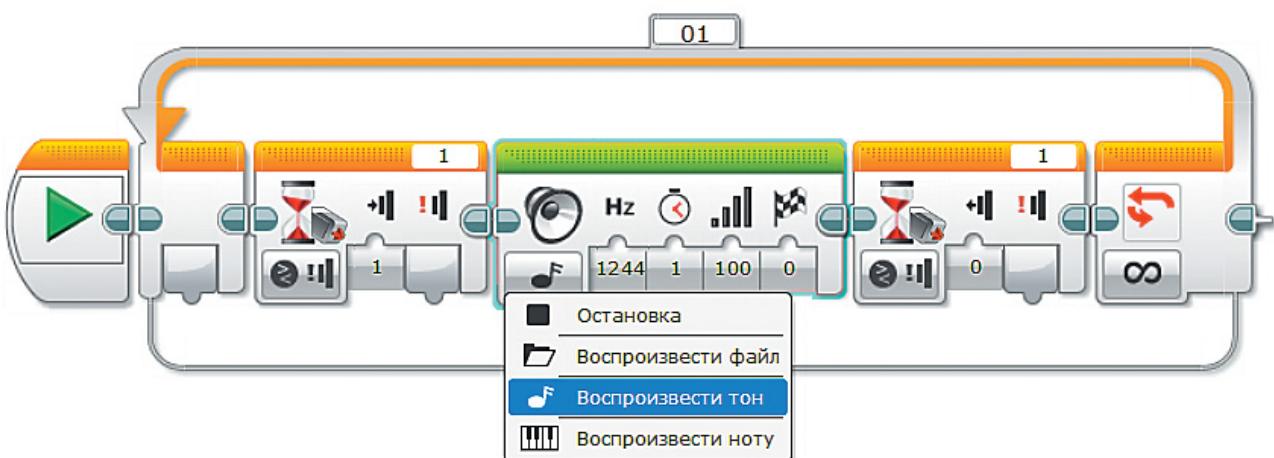
3. Выбери состояние **Нажатие** (1). В результате следующий шаг в алгоритме не будет выполнен, пока датчик касания не перейдёт в необходимое состояние **Нажатие** (1).



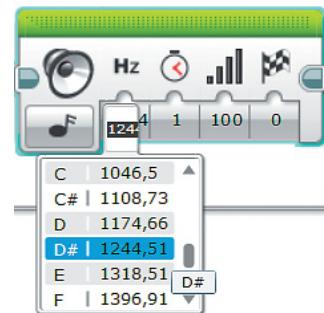
4. На следующем шаге добавь такой же оранжевый блок, но выбери состояние **Отпущенено** (0).



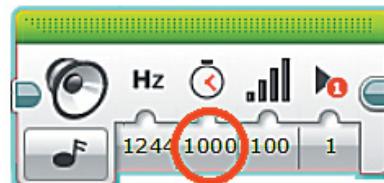
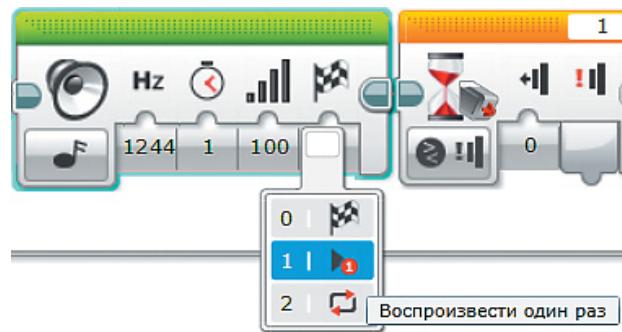
5. Между командами **Ожидание** добавь команду **Звук** (зелёный блок) и выбери опцию **Воспроизвести тон**.



- 1) В этом же блоке выбери частоту D# 1244,51 (можно и другую, но данная частота хорошо подходит для восприятия на слух).

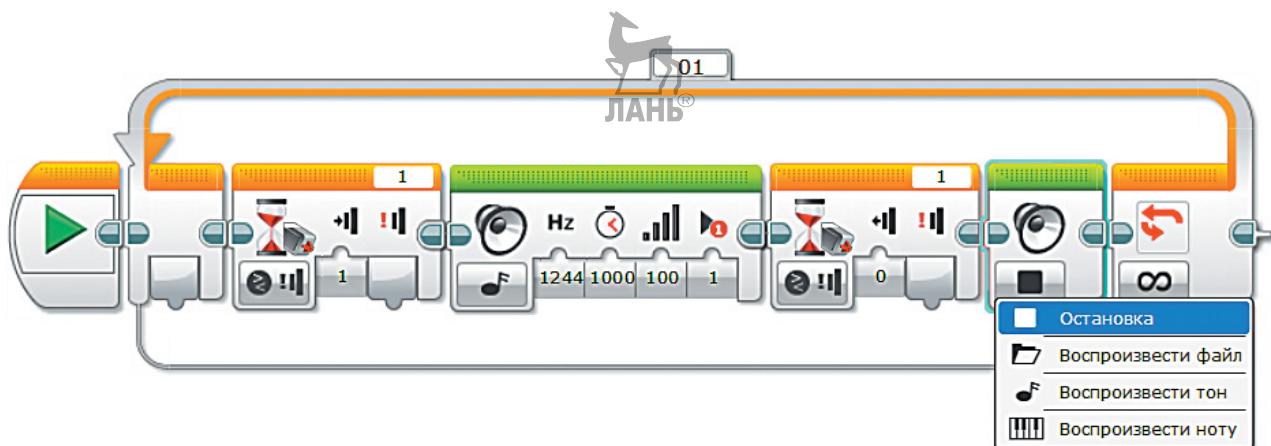


- 2) Выбери также тип воспроизведения **Воспроизвести один раз (1)**, иначе сигнал будет подаваться в течение времени, выставленного в продолжительности, а не только при нажатой кнопке.
- 3) Чтобы можно было подавать сигнал любой длительности, выстави в продолжительности какое-нибудь большое число, например **1000**.
6. Осталось остановить звуковой сигнал в тот момент, когда кнопка будет отпущена. Если этого не сделать, то сигнал будет подаваться, в нашем случае, в течение 1000 секунд.



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

Программа готова.



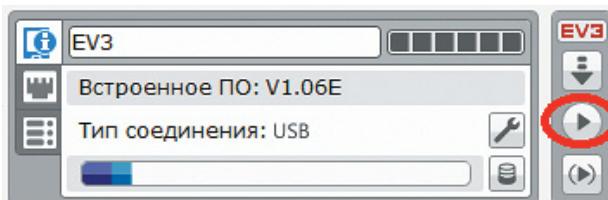
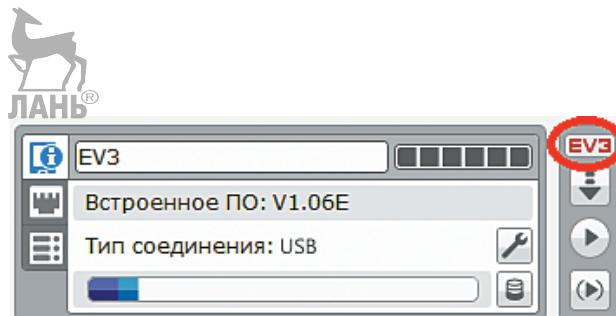
Этап 5. Загрузка программы и её тестирование



ШАГ 1. ЗАГРУЗКА ПРОГРАММЫ

Загрузи свою программу.

- 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.
- Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



ШАГ 2. ТЕСТИРОВАНИЕ ПРОГРАММЫ

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

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



Внимание!

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





Этап 6. Поговорим на языке азбуки Морзе

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

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

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

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

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

НАЧИНАЙ ИГРАТЬ

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

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

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

Начинайте игру по следующему сценарию:

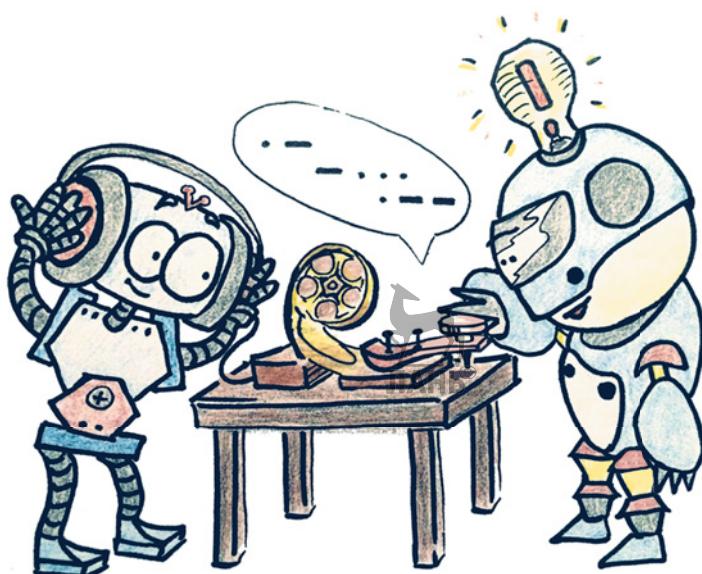
1. В минуту опасности, пользуясь международной азбукой Морзе и правилами кодирования Морзе, шифровальщик на корабле кодирует следующее сообщение: «SOS SOS SOS».
2. Телеграфист отправляет получившийся код с помощью передатчика.
3. Стенографист спасателей слушает сигналы и составляет код из «точек» и «тире».
4. Дешифровальщик расшифровывает полученное сообщение.
5. Шифровальщик спасателей кодирует сообщение: «RCD», а телеграфист передаёт его.
6. Стенографист на корабле слушает сигналы и записывает сообщение — код из «точек» и «тире».
7. Дешифровальщик расшифровывает полученное сообщение.

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

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

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

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





Этап 7. Отправляем телеграмму

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

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

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

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

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

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

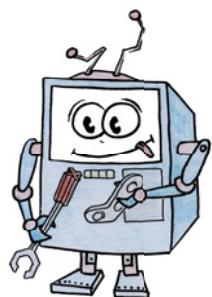


Оборудование:

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

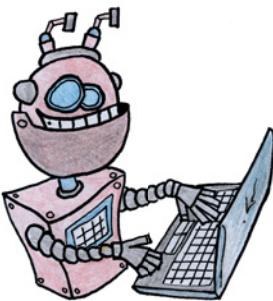


Этап 8. Кнопочный текстовый шифратор



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





Этап 9. Пишем программу для шифратора

ЛОГИКА ПРОГРАММЫ

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

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

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

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

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

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

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



ШАГ 1. ЗАПУСК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LME-EV3

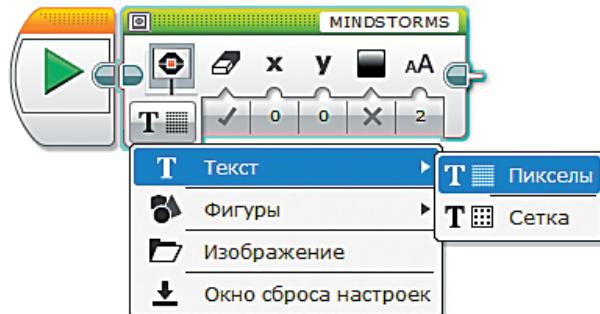
1. Запусти программное обеспечение LME-EV3.
2. В открывшемся окне слева выбери пункт меню **Файл**.
3. Выбери пункт **Новый проект → Программа → Открыть**.



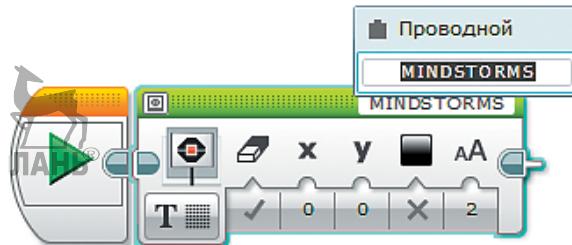
ШАГ 2. СОСТАВЛЕНИЕ ПРОГРАММЫ ДЛЯ ШИФРАТОРА

В открывшемся окне проекта начинай составлять программу для шифратора.

1. Начни с очистки экрана. Если этого не сделать, то на экране будет отображаться стандартный логотип EV3 и название исполняемой программы. Для этого добавь команду **Экран** (зелёный блок) и выбери опцию **Текст → Пиксели**.



2. Заметь, в этом блоке по умолчанию стоит очистка экрана. Это говорит о том, что экран будет очищен перед выводом текста, который записан в правом верхнем окошке. То есть при выполнении этой команды на экране появится надпись «MINDSTORMS». Чтобы на экран ничего не выводилось, поменяй значение в этом окошке на **Проводной**. Это будет означать, что текст формируется отдельно и отправляется на экран, но формироваться ничего, конечно же, не будет.



3. Теперь объяви две переменные с именами *x* и *y*. Для этого добавь две команды **Переменная** (красный блок). Затем кликни в правое верхнее окошко и выбери опцию **Добавить переменную**.



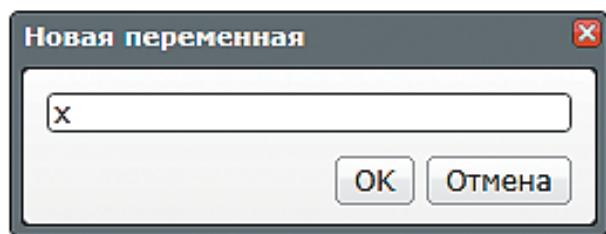
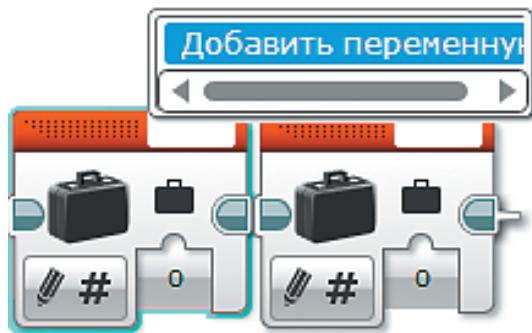
Тебе будет предложено ввести имя новой переменной. В появившемся окне введи *x*.

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

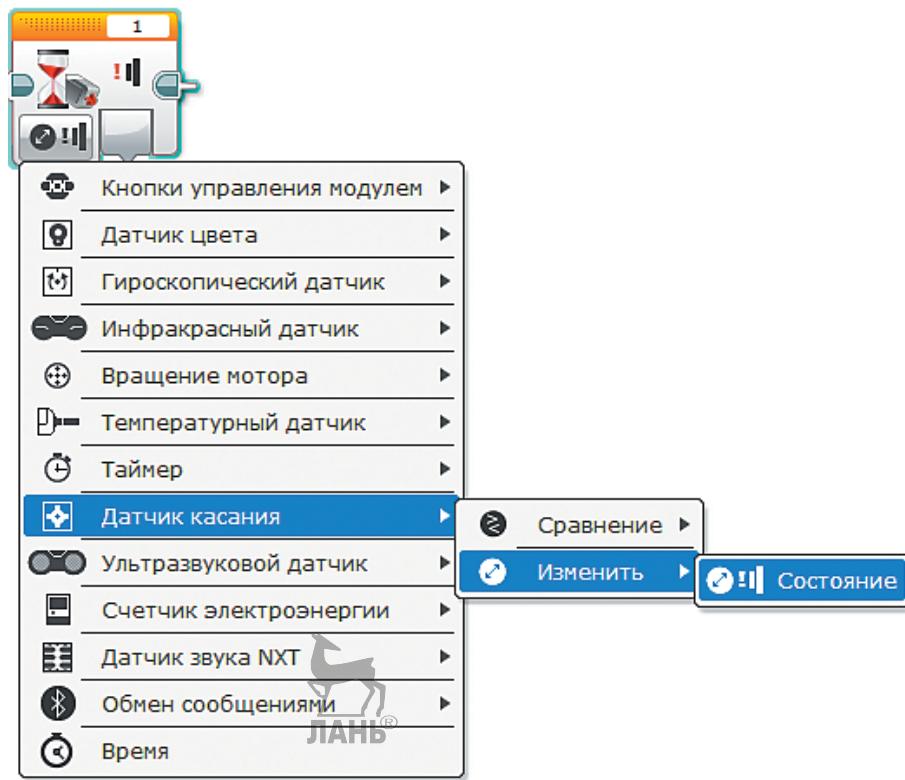
То же самое нужно сделать со второй командой, только ввести имя переменной «*y*».



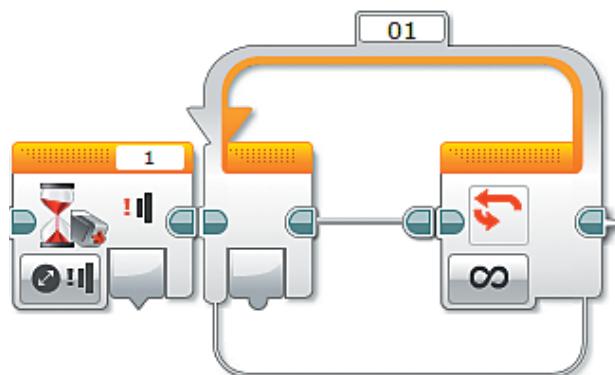
Внимание! По умолчанию здесь выбрана опция Запись → Числовое значение и введено значение 0. Это означает, что в переменные записаны нули.



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



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



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



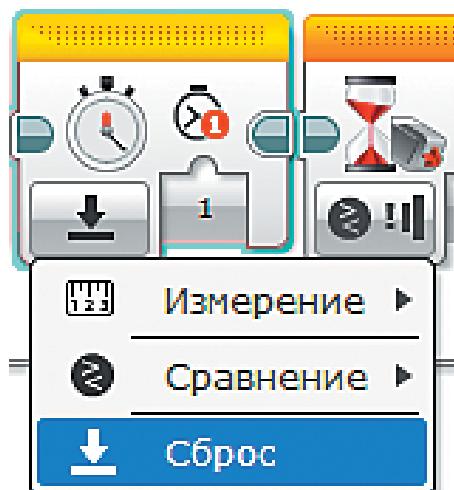
7. Чтобы замерить время, в течение которого была нажата кнопка, добавь команду **Таймер** (жёлтый блок) после нажатия кнопки.



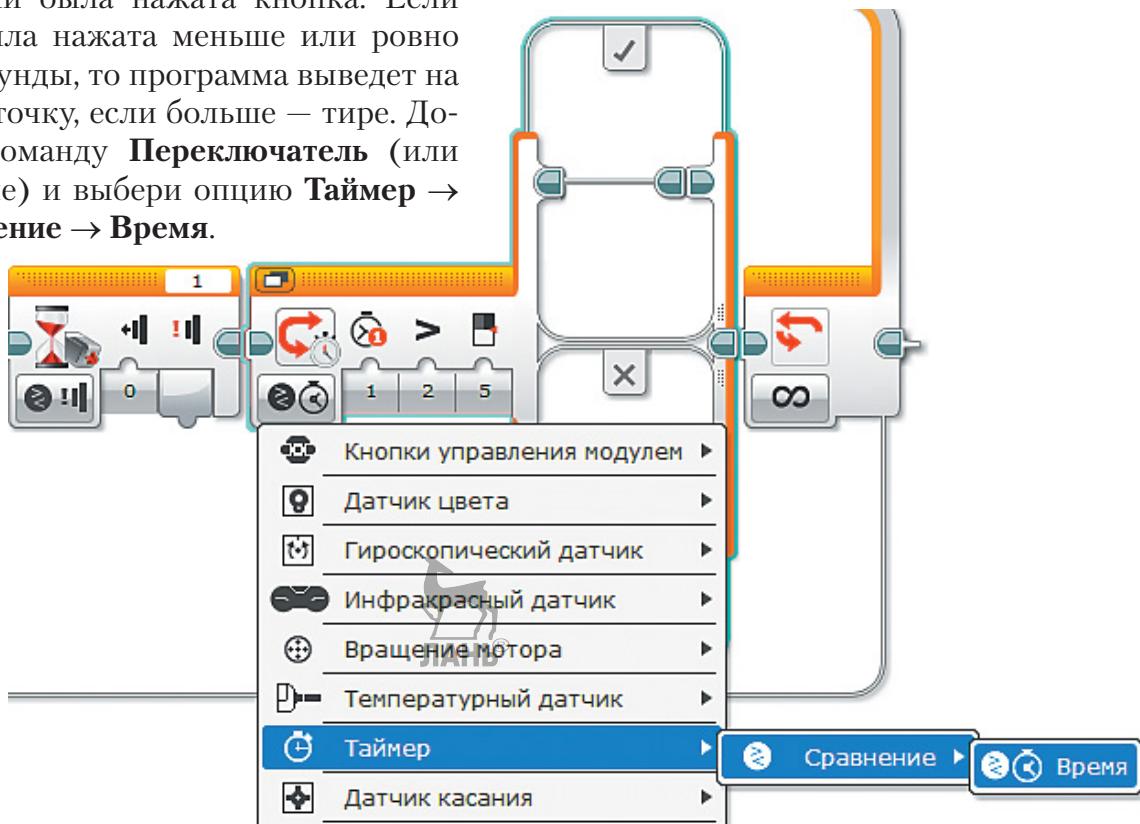
В команде **Таймер** выбери опцию **Сброс**. В результате таймер обнулит-
ся и начнёт отсчёт времени с нуля.



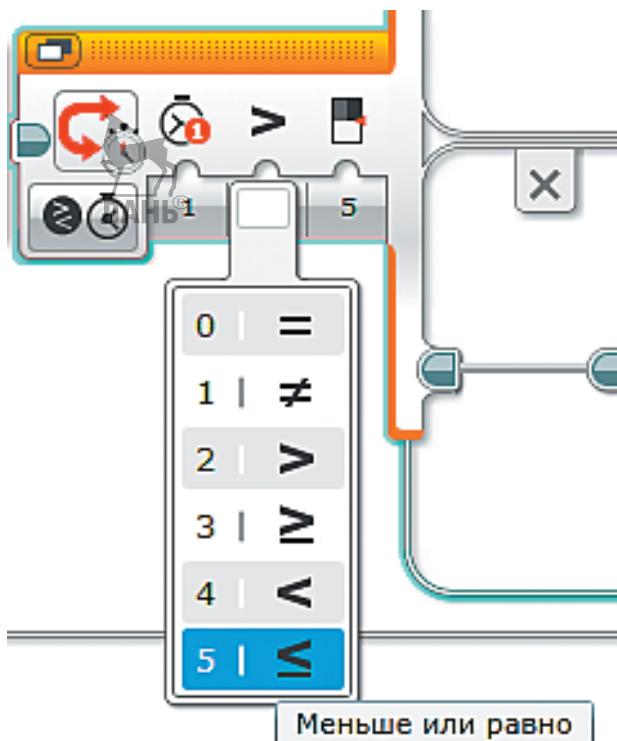
Внимание! Цифра в окошке обозначает идентификатор таймера и говорит о том, что мы пользуемся таймером № 1. Всего может быть задействовано восемь таймеров.



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



9. Настройка условия. По умолчанию в данный момент считается показание таймера № 1. Далее следует выбрать другой тип сравнения — **Меньше или равно**.



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

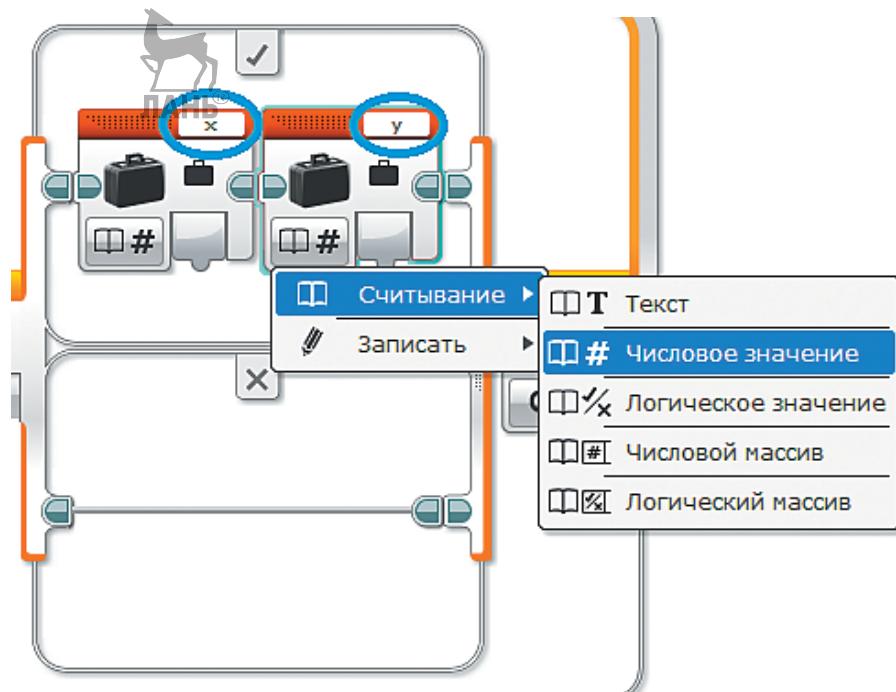


10. В верхней строчке пропиши события, которые наступают, если условие выполнено, а в нижней — если не выполнено.

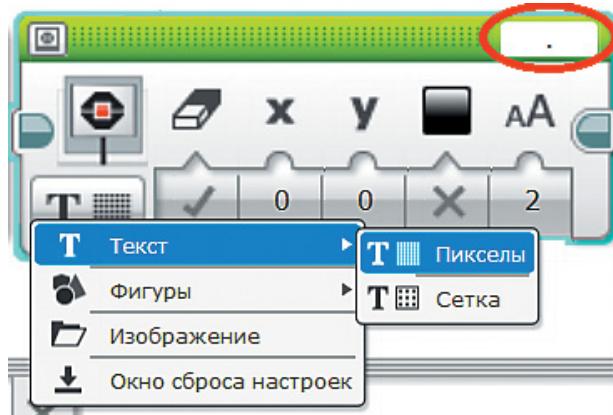
Если кнопка действительно была нажата меньше или ровно 0,2 секунды, то выведи точку. Приступай.

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

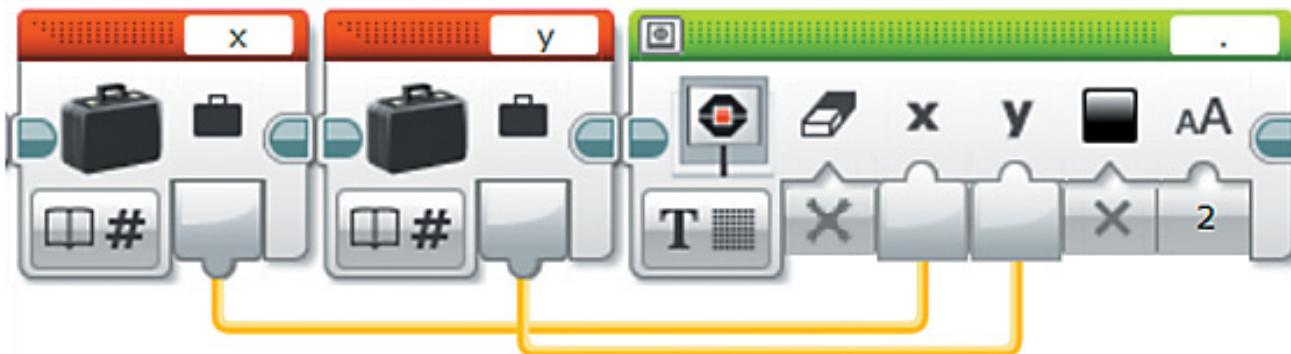
Обрати внимание на имена считывающих переменных!



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



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



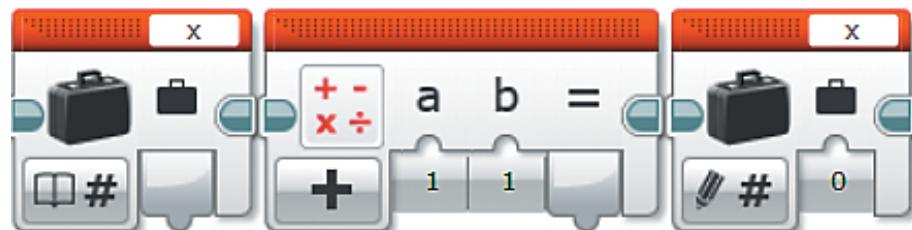
Далее надо задать отступ между знаками. Для этого увеличь значение переменной **x**. Ты сам можешь рассчитать размер отступа, однако наиболее оптимальным будет 10 пикселов.

Это можно сделать, считывая значения переменной **x** и добавляя к ней числа 10 с помощью команды **Математика** (красный блок).

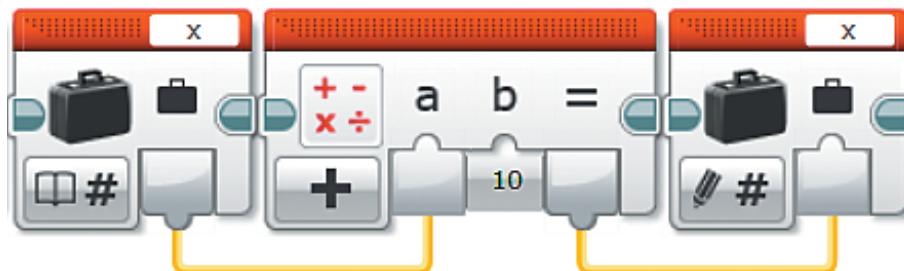




Добавь команду **Переменная**, выбери опцию **Считать** → **Числовое значение**, затем добавь команду **Математика**. По умолчанию выбрана опция **Добавить** и команда записи переменной.

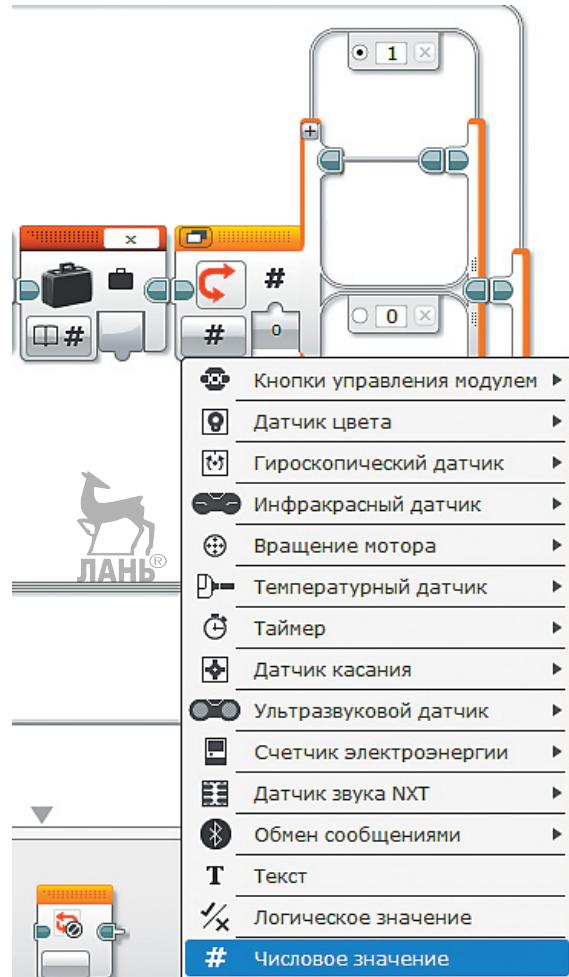


Соедини выходной порт переменной с входным портом **a**. В порт **b** впиши число **10**. Выходной порт результата соедини с входным портом записи числового значения **x**.

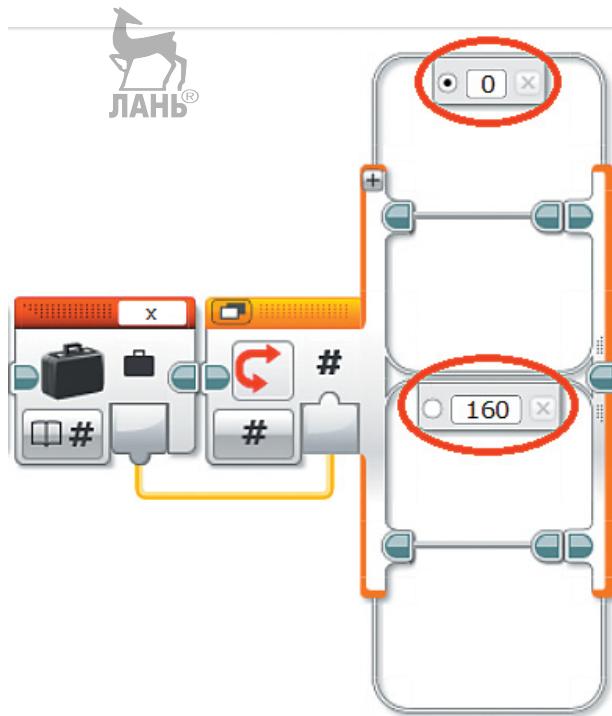


11. Теперь проверь, не достиг ли ты конца строки на экране. Для этого считай значение переменной *x* и проверь условие «Равно ли это значение 160?» (именно на 160-м пикселе и заканчивается строка экрана).

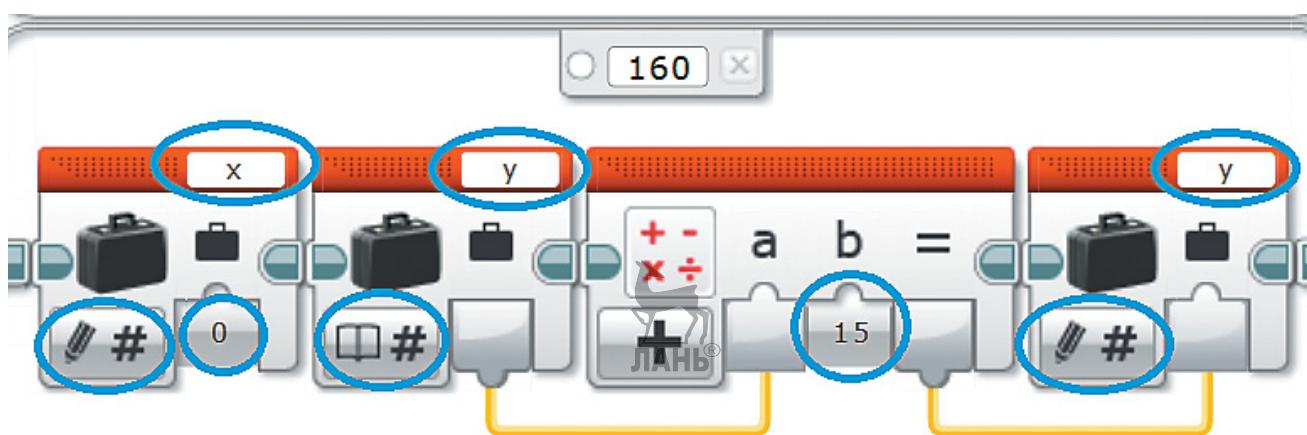
Добавь команду считывания переменной *x* и условие. В условии выбери опцию **Числовое значение**.



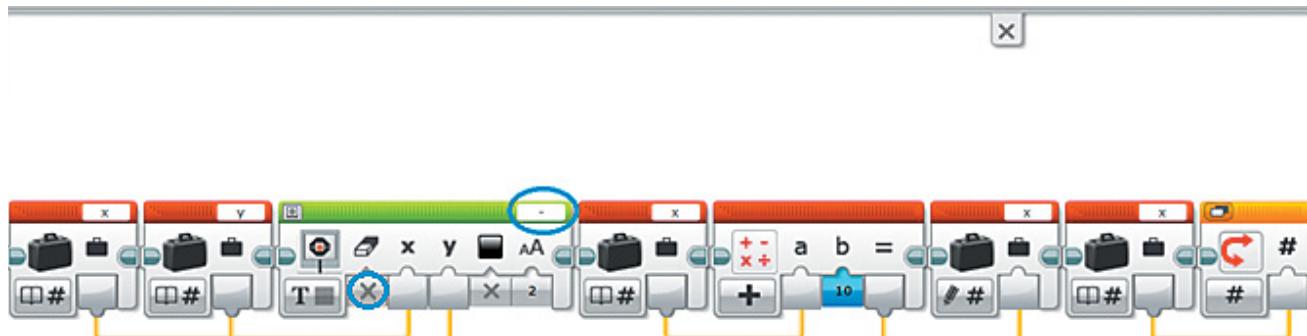
Соедини выходной порт переменной с входным портом условия. Выстави значение по умолчанию **0**, а альтернативным будет значение **160**.



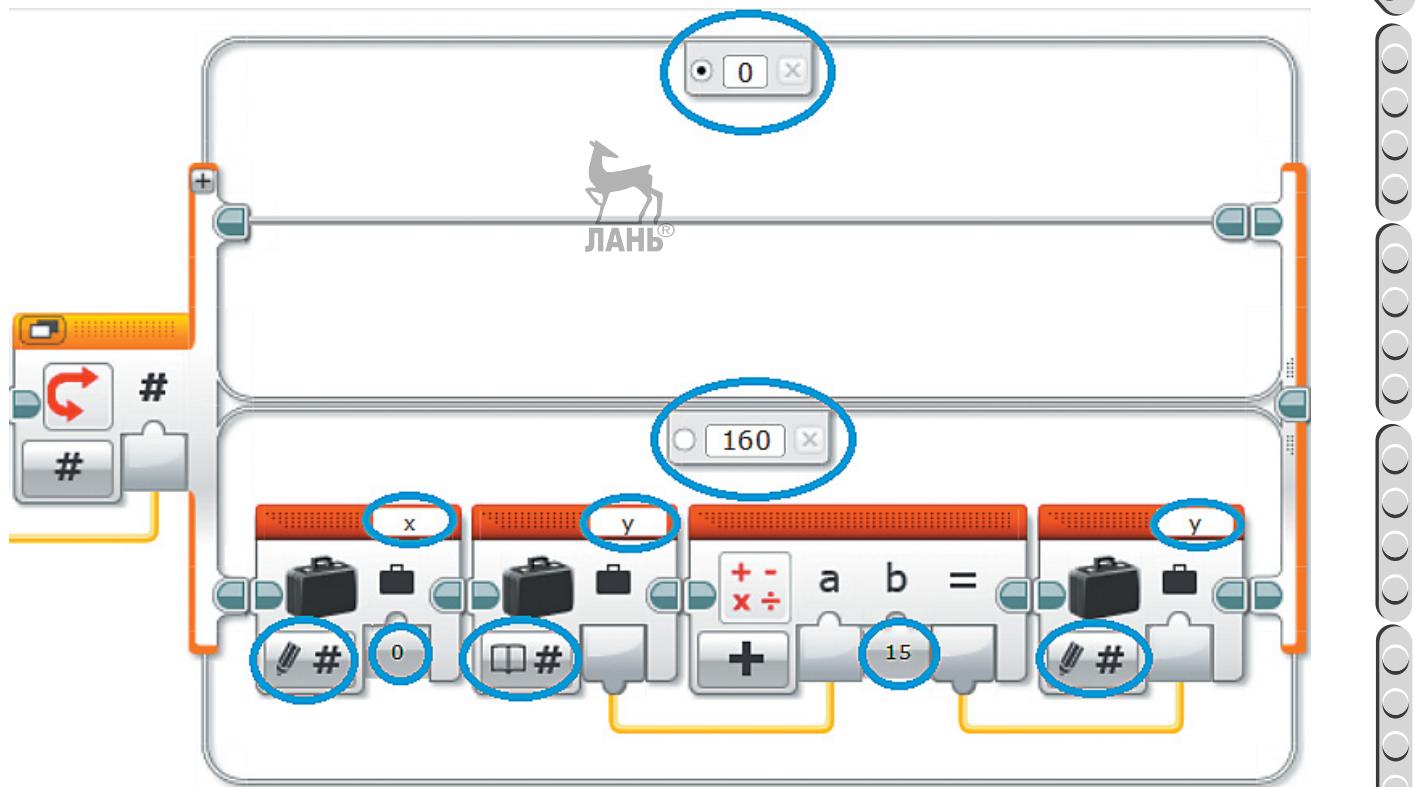
Нас интересует только случай, когда мы достигли 160-го пикселя. Тогда обнули значение x (то есть вернись в начало строки) и увеличь значение y на 15 (то есть спустись ниже на строку). У тебя получится такая последовательность:



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



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

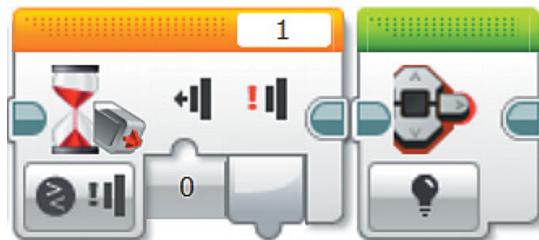


13. На следующем шаге добавь простую вещь — включение светового индикатора EV3 при нажатии кнопки. Это облегчит работу телеграфиста.

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



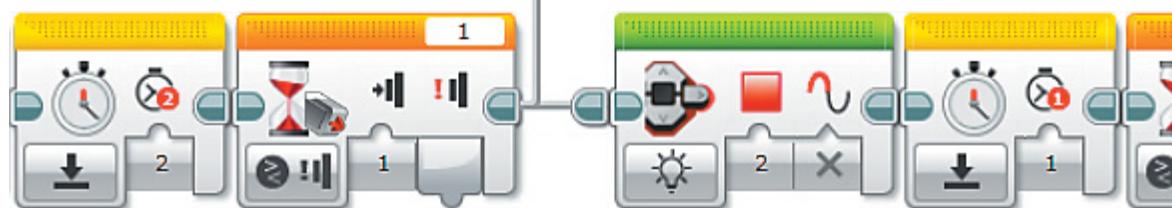
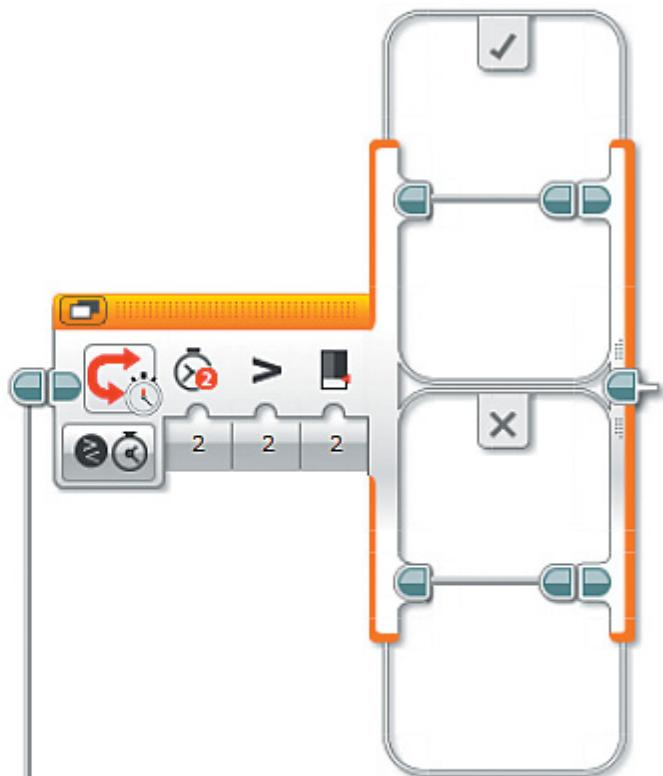
После того как кнопка отпущена, индикация должна выключаться. Для этого после отжатия кнопки добавь ещё одну такую команду и выбери опцию **Выключить**.



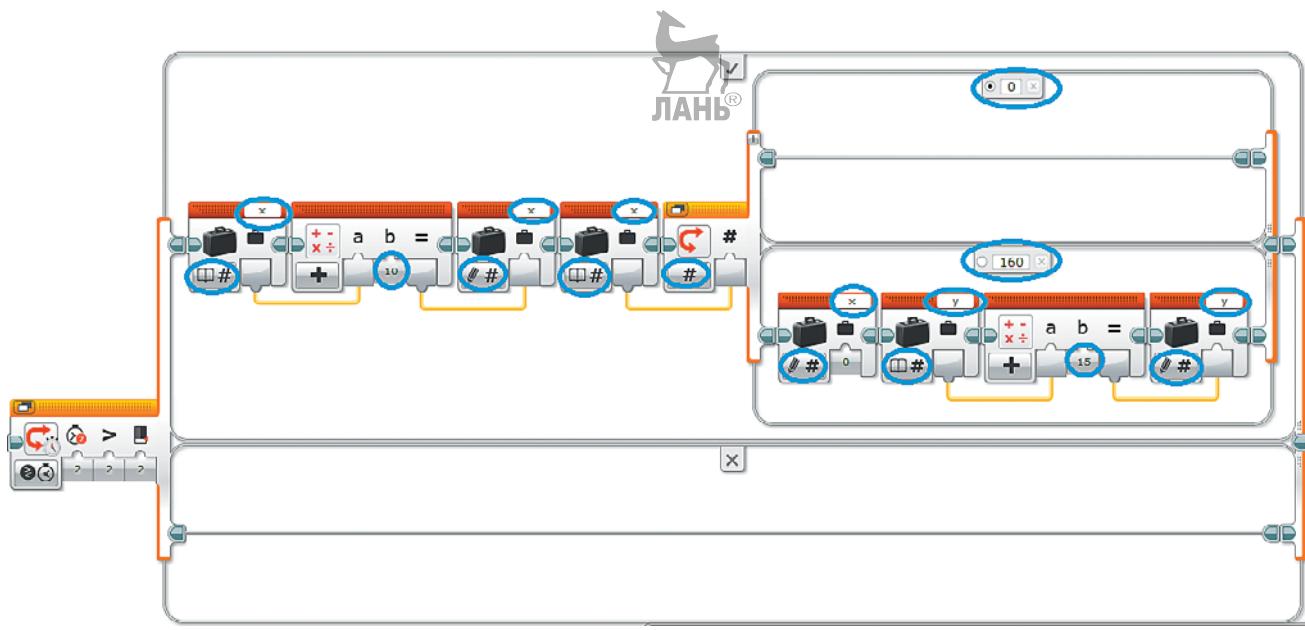
14. Осталось автоматизировать процесс постановки пробела. Добавь в **самом начале цикла** ещё один таймер, выбери опцию **Сброс** и присвой ему номер **идентификатора 2**. Он будет начинать работать в самом начале. Считывать его значение нужно каждый раз после нажатия кнопки. Если прошло много времени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



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



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



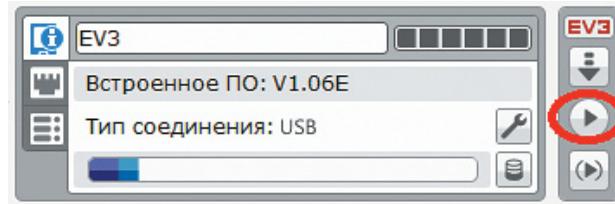
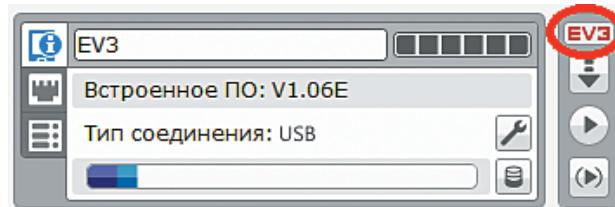


Этап 10. Загрузка программы и её тестирование

ШАГ 1. ЗАГРУЗКА ПРОГРАММЫ В ПРОГРАММИРУЕМЫЙ МОДУЛЬ

Загрузи свою программу.

- 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.
- Нажми кнопку **Загрузить и запустить программу**. Программа запишется в память программируемого модуля и сразу запустится.



ШАГ 2. ТЕСТИРОВАНИЕ



Протестируй свой передатчик.

- При коротком нажатии кнопки на экране должна появляться «точка», при длительном – «тире».
- При каждом нажатии кнопки панель EV3 должна загораться красным.
- Когда строка на экране заканчивается, должна начинаться новая.
- Если кнопка не нажимается более 2 секунд, то при следующем нажатии должен появляться пробел.

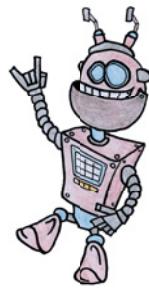
Если передатчик функционирует неправильно, то, скорее всего, были допущены ошибки при написании программы. Проверь все шаги ещё раз.



Внимание!

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

А теперь...



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

В каждой группе определите роли:

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



Составьте сценарий вашей переписки, например такой.

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

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

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

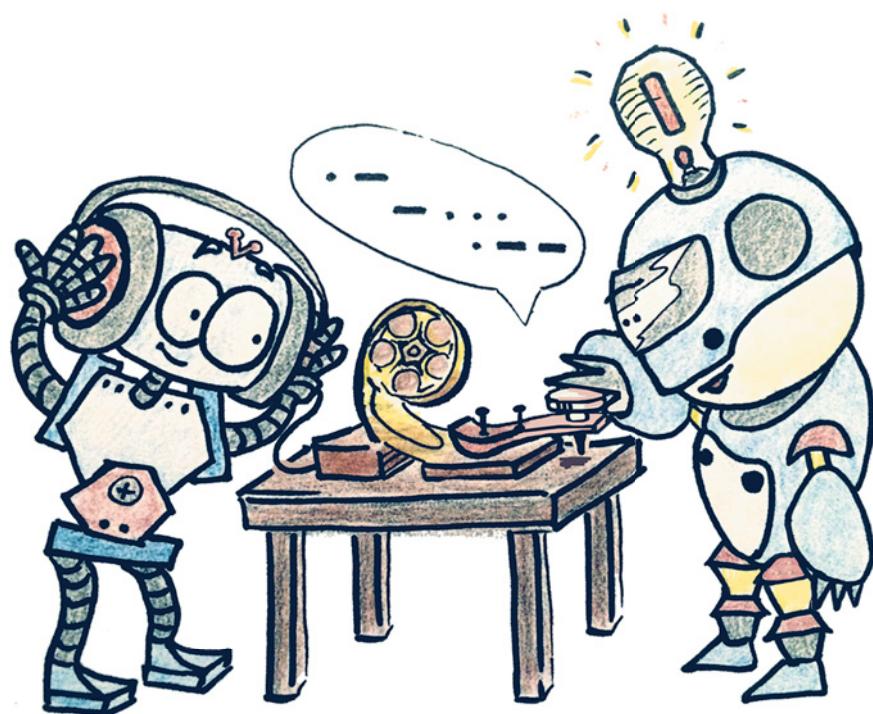
Когда библиотекари получают заказ, они должны ответить, имеется ли в наличии такая книга. Если да, то сколько экземпляров книги они готовы предоставить.

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



До новых встреч!

Ты узнал историю изобретения телеграфного аппарата и азбуки Морзе, научился собирать настоящий передатчик и отправлять закодированные сообщения. Но впереди ещё так много интересного! В серии книг «РОБОФИШКИ» ты можешь познакомиться с другими замечательными проектами и стать настоящим изобретателем!



Содержание

Здравствуйте!	3
Дорогой друг!	4
История телеграфа	5
Этап 1. Изучаем способ кодирования, предложенный Морзе	8
Этап 2. Сборка кнопочного звукового передатчика	11
Шаг 1. Сборка подставки под программируемый модуль	11
Шаг 2. Сборка и крепление кнопки телеграфного ключа	17
Шаг 3. Соединение кнопки с программируемым модулем.	18
Этап 3. Установка программного обеспечения на компьютере	20
Шаг 1. Запуск программного обеспечения LME-EV3.	20
Этап 4. Пишем программу для передатчика	21
Логика программы	21
Общий вид программы	21
Этап 5. Загрузка программы и её тестирование	25
Шаг 1. Загрузка программы	25
Шаг 2. Тестирование программы	25
ЛАНЬ®	
Этап 6. Поговорим на языке азбуки Морзе.	26
Начинай играть	26
Этап 7. Отправляем телеграмму	28
Этап 8. Кнопочный текстовый шифратор	29
Этап 9. Пишем программу для шифратора	30
Логика программы	30
Шаг 1. Запуск программного обеспечения LME-EV3.	31
Шаг 2. Составление программы для шифратора	31

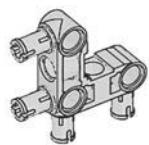
Этап 10. Загрузка программы и её тестирование	44
Шаг 1. Загрузка программы в программируемый модуль	44
Шаг 2. Тестирование	44
А теперь...	45
До новых встреч!	46





Штифты

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



Соединительный штифт, 3-модульный, синий, 4x



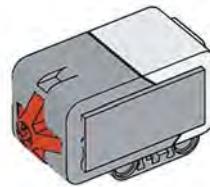
Соединительный штифт, 2-модульный, чёрный, 10x



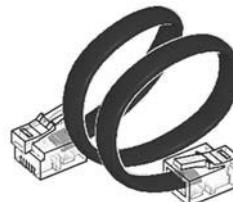
Программируемый модуль EV3, 1x



Датчик касания, 1x

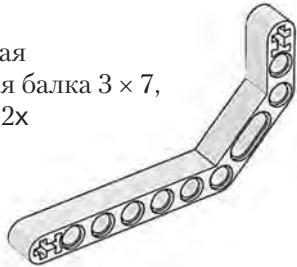


Кабель, 25 см, 1x



Балки

Двойная
угловая балка 3×7 ,
белая, 2x



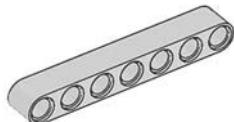
Поперечная балка 2×1 ,
красная, 2x



Балка № 3, жёлтая, 1x



Балка № 7, серая, 3x



Поперечные блоки

Поперечный блок,
2-модульный, серый, 2x



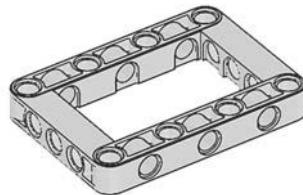
Оси

Ось, № 10, чёрная, 1x



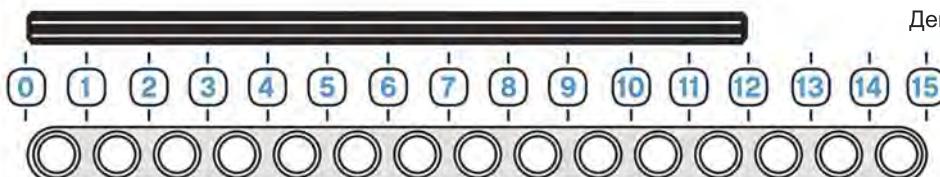
Рамки

Рамка 5×7 , серая, 1x



M 1:1

Действительный
размер





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"

Электронное издание для досуга

Серия: «РОБОФИШКИ»

Тарапата Виктор Викторович

**КОНСТРУИРУЕМ РОБОТОВ НА LEGO® MINDSTORMS® EDUCATION EV3.
ТАЙНЫЙ КОД СЭМЮЭЛА МОРЗЕ**

Для детей среднего и старшего школьного возраста

Ведущий редактор Т. Г. Хохлова

Руководители проекта от издательства А. А. Елизаров, С. В. Гончаренко

Научный консультант канд. пед. наук Н. Н. Самылкина

Ведущий методист В. В. Тарапата

Художники В. Е. Шкерин, Я. В. Соловцова, И. Е. Марев, Ю. Н. Елисеев

Компьютерная верстка: Е. А. Голубова

Подписано к использованию 05.04.21.

Формат 210×260 мм

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: 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!»
- 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

