

УДК 621.86/.87  
ББК 32.816  
В48

*Серия основана в 2016 г.*

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

**Винницкий Ю. А.**

В48 Конструируем роботов на ScratchDuino®. Первые шаги / Ю. А. Винницкий, К. Ю. Поляков. — 2-е изд. — М. : Лаборатория знаний, 2018. — 116 с. : ил. — (РОБОФИШКИ).

ISBN 978-5-00101-086-9

Тем, кто освоил LEGO и хочет двигаться дальше, открывая для себя безграничные возможности робототехники, мы предлагаем познакомиться с новой платформой, которая называется «ScratchDuino». Из этой книги, представляющей собой практическое руководство, вы узнаете об особенностях программирования в среде разработки Scratch, о том, как устроены роботы, научитесь собирать их и управлять ими. В итоге вы сможете придумывать собственные проекты, с помощью которых проверите, на что способны собранные вами роботы.

Для юных исследователей и программистов, а также всех тех, кто увлечен конструированием роботов.

**УДК 621.86/.87  
ББК 32.816**

12+

---

*Издание для досуга*

Серия: «РОБОФИШКИ»

**Винницкий Юрий** Анатольевич

**Поляков Константин** Юрьевич

**КОНСТРУИРУЕМ РОБОТОВ НА SCRATCHDUINO®.  
ПЕРВЫЕ ШАГИ**

*Для детей среднего и старшего школьного возраста*

Ведущий редактор *Ю. А. Серова*

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

Научный консультант *Н. Н. Самылкина*. Ведущий методист *В. В. Тарапата*

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

Компьютерная верстка: *В. И. Савельев*

Подписано в печать 20.06.17. Формат 70×100/16.

Усл. печ. л. 9,75. Заказ

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272, e-mail: info@pilotLZ.ru, <http://www.pilotLZ.ru>

## Здравствуйте!

Издание, которое вы держите сейчас в руках, — не просто книга. Это ключ. Он открывает двери, за которыми находится увлекательное и притягивающее пространство реальной робототехники. Оно называется «Мир ScratchDuino». Постепенно, шаг за шагом, этот мир станет знакомым и привычным. И то, что в результате вы самостоятельно сумеете собрать своими руками настоящее работающее устройство, — это, конечно, победа и успех!

Но главное — вы поймете, что такие ценные качества характера, как терпение, аккуратность, настойчивость и творческая мысль, проявленные при работе над проектом, останутся с вами навсегда, помогут уверенно создавать свое будущее, стать реально успешным человеком, независимо от того, с какой профессией 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. Все, что нас окружает сегодня, придумано инженерами. Их любознательность, желание узнавать неизведанное и конструировать то, чего никто до них не делал, и создает 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 вперед.

И тогда, уверенно освоив пространство ScratchDuino и перейдя последовательно к следующим мирам реальной робототехники, вы сделаете очередной шаг по этой дороге.

Успехов вам!

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

## От авторов

Вы открываете книгу, которая называется «Конструируем роботов на ScratchDuino». Это означает, что мы будем собирать самых разных по назначению роботов на этой платформе. Роботами называют любые устройства, которые выполняют какую-то работу автоматически, то есть без участия человека. Роботов вокруг нас становится с каждым годом все больше и больше. Они вместо человека убирают мусор в квартире, собирают точные электронные приборы на заводах, обслуживают посетителей в кафе и даже выполняют хирургические операции в больницах.

Но эта книга — особенная. Вы не просто прочитаете про роботов, но и научитесь управлять некоторыми из них.

Конечно, построить реальных роботов довольно непросто. Сначала нужно придумать конструкцию робота, затем изготовить все детали, собрать робота, и только потом можно будет приступить к испытаниям. Для нас пока такой путь слишком сложен, он займет очень много времени (возможно, несколько лет!).

Но есть хорошая новость: специально для того, чтобы научиться управлять роботами, выпускаются готовые наборы-конструкторы, в которых уже есть все необходимые части. Существует несколько типов таких наборов, например, LEGO MINDSTORMS, ТРИК и др.

Мы тоже будем использовать готовый набор, который называется «ScratchDuino». Он разработан российской компанией ЗАО «ТЫРНЕТ» и содержит готовую робоплатформу — специальную плату на колесах, в которую вставляется управляющий компьютер. К платформе остается только подключить нужные датчики (реагирующие на свет, касание и т.д.), и она готова к работе. Поэтому сборка робота выполняется очень быстро, и вот уже роботом можно управлять!

Из этой книги вы узнаете, как правильно настраивать робота, и научитесь писать программы для управления им. Первые работы вы будете делать вместе с нами — каждый шаг подробно объясняется в книге, а затем вас ждет работа над самостоятельными творческими проектами нашей серии «РОБОФИШКИ».

Успехов!

### **ВАЖНО!**

**Научившись собирать и программировать роботов, вы сможете провести их испытания в специально разработанных проектах. Проекты могут объединять знания и умения, полученные вами в ходе изучения нескольких предметов. В большинстве случаев это физика, математика, информатика и технология.**

**Результаты своих работ можно сохранить в личном портфолио (в цифровом виде) и использовать в самых разных конкурсах и даже при поступлении в институт.**

## **Благодарности**

Над проектом ScratchDuino работает много творческих людей, объединенных в сетевые сообщества. Но особенно хочется поблагодарить Анну Самарину, Дарью Буркут, Елену Вострикову и Леонида Захарова за идеи, советы, предложения, нашедшие отражение в книге.

# Роботы и робототехника

## Зачем нужны роботы?

Вы, конечно, слышали слово «робот» и знаете, что роботом называют техническое устройство, которое может заменить человека во время выполнения сложных, утомительных или опасных работ.

Само слово «робот» придумал чешский писатель Карел Чапек, он образовал его от чешского слова *robota*, что значит «подневольный труд».

Большинство роботов не похожи на человека. На современных заводах используется огромное количество промышленных роботов — станков с числовым программным управлением. Они работают по заложенной в них программе. Каждая программа соответствует определенной операции или детали. Для того чтобы перенастроить станок на изготовление другого типа деталей, достаточно просто заменить программу.

Роботы используются на конвейерных линиях, изготавливающих микросхемы для компьютеров: процессоры, память и др.

Все более популярными становятся технологии 3D-печати, позволяющие с помощью роботов специального типа (3D-принтеров) послойно изготавливать различные детали.

Нас окружают автоматизированные системы, которые тоже можно назвать роботами, например, система управления движением поездов метро, система управления отоплением дома.

Роботы, в отличие от людей, не устают, работают 24 ч в сутки, не болеют, могут работать в жаре и в холоде, в опасных для человека условиях. Во многих странах используют роботизированные установки пожаротушения. Уже продаются бытовые роботы — автоматические пылесосы, которые могут убрать пыль и самостоятельно вернуться на место для под-



**Карел Чапек**  
(1890—1938)



**Робот Asimo**

([www.robotonline.net](http://www.robotonline.net))

зарядки. В XXI в. роботам стали доверять даже выполнение хирургических операций. Для добычи нефти и газа в глубинах вод используются подводные робототехнические установки. В вооруженных силах многих стран «служат» боевые роботы, которые могут вести наблюдение и разведку, а также уничтожить противника самостоятельно, без риска для жизни солдат.

Некоторые роботы похожи на человека, их называют **андроидами**. Самый известный робот-андроид *Asimo* выпущен компанией *Honda*.

Существует особый класс роботов, которые работают самостоятельно. Они называются **автономными роботами**. Это, например, беспилотные автомобили и летательные аппараты, роботы для исследования космоса и Мирового океана.

Полностью автономный робот может:

- перемещаться и работать длительное время без вмешательства человека;
- собирать информацию об окружающей среде;
- приспосабливаться к изменению обстановки, изменяя алгоритм своей работы.

## Что такое робототехника?

**Робототехника** — это прикладная наука о разработке и использовании автоматизированных технических систем.

Робототехника опирается на достижения таких наук, как механика, автоматика, кибернетика, информатика. Некоторые алгоритмы управления роботами используют элементы искусственного интеллекта.

Слово «робототехника» (от англ. *robotics*) впервые использовал американский писатель-фантаст Айзек Азимов в 1941 г. Он же предложил три закона робототехники, которые определяют, как должен вести себя робот.

1. Робот не может причинить вред человеку или своим бездействием допустить, чтобы человеку был причинен вред.

2. Робот должен повиноваться всем приказам, которые дает человек, кроме тех случаев, когда эти приказы противоречат первому закону.

3. Робот должен заботиться о своей безопасности в той мере, в которой это не противоречит первому и второму законам.

## Из чего состоит робот?

Прежде всего робот — это механическое устройство. Поэтому его создание — это инженерно-конструкторская работа. Нужно, чтобы все детали были хорошо закреплены, и робот не потерял равновесия и не развалился во время выполнения задания. В то же время все подвижные детали должны исправно двигаться и крутиться.

Роботом нужно как-то управлять, поэтому необходима система управления, которая в современных роботах строится на микроконтроллерах.

**Микроконтроллер** — это миниатюрный компьютер (микросхема), все части которого размещены на одном кристалле кремния.

Микроконтроллер содержит процессор, разъемы для управления внешними устройствами (порты), оперативную и постоянную память. В постоянную память микроконтроллера записана программа, которую он при включении сразу начинает выполнять.

Для того чтобы робот мог двигаться, нужны *исполнительные устройства* — моторы, которые преобразуют электрическую энергию в механическую энергию вращения. Автономные роботы чаще всего передвигаются с помощью колес или гусениц, для неровных поверхностей иногда используют шагающие системы (для опоры им нужны небольшие участки поверхности, поэтому они проходят по труднодоступным местностям). Существуют также ползающие и плавающие роботы.

Передвижение роботов — это достаточно сложная самостоятельная задача. Например, современные роботы-спасатели должны уметь подниматься по лестнице здания.

Для того чтобы получать данные об окружающей обстановке, роботу нужны *датчики* (сенсоры, чувствительные элементы) — устройства, которые измеряют определенную физическую величину и выдают информацию о ней в виде электрических сигналов. Поскольку эти сигналы затем будет обрабатывать цифровой компьютер, их нужно преобразовать в двоичный код, в числа. Эту работу выполняет аналого-цифровой преобразователь (АЦП), который, как правило, встроен в микроконтроллер.

Существуют различные типы датчиков (**рис. 1.1**): температуры, давления, скорости, освещенности, касания («кнопки»), расстояния (они измеряют время, за которое ультразвук отражается от препятствия и возвращается обратно), инфракрасные датчики для поиска и обнаружения объектов и многие другие. Более сложные роботы имеют системы компьютерного зрения, использующие видеокамеры. Они могут «узнавать» простые предметы, определять их расположение в пространстве и достраивать невидимые части, используя информацию из своей базы данных.

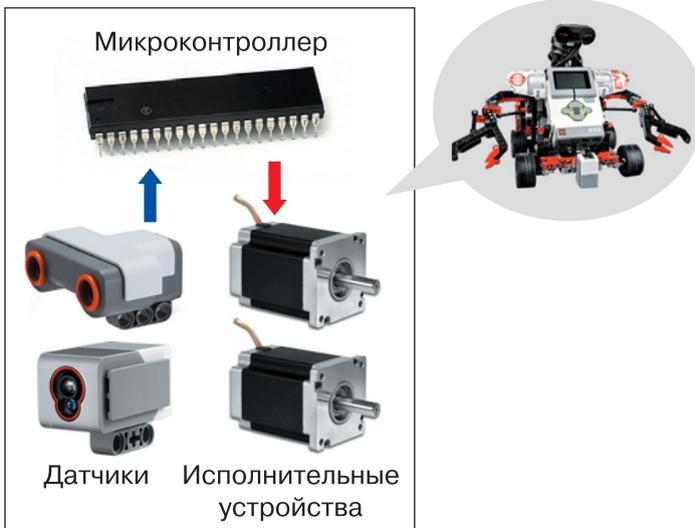


Датчики из наборов *ScratchDuino*  
(в корпусах с креплением на магнитах):



**Рис. 1.1**

Таким образом, встроенный в робота микроконтроллер управляет исполнительными устройствами и обрабатывает данные, поступающие с датчиков (**рис. 1.2**).



**Рис. 1.2**

## Какой язык понимают роботы?

Как вы уже знаете, движением робота управляет небольшой компьютер — микроконтроллер. Он выполняет команды, которые передает ему человек. У робота есть список команд, которые он понимает и умеет выполнять. Поскольку в компьютерах вся информация (в том числе команды!) хранится в виде чисел (кодов), дать задание роботу можно в виде цепочки чисел:

**15, 20, 77, 85, 32, 16, 14.**

Поняли ли вы, какое задание получил робот? Наверняка нет: откуда же вам знать, какую команду обозначает каждое число? Поэтому для того, чтобы проще было управлять роботами, люди придумали специальные языки, в которых команды записываются словами. Такие языки называются языками программирования, потому что мы программируем робота — сообщаем ему план действий.

Но робот не понимает «человеческие», естественные для нас языки программирования, он понимает только числовые команды! Поэтому нужна еще специальная программа — она называется **«транслятор»**, которая переводит программу, написанную человеком, в команды робота (числа). Таким образом, можно писать программы управления роботами на любом языке, для которого есть программа-транслятор.

Профессиональные программисты обычно пишут программы в виде текста в обычных текстовых редакторах. В конце книги мы увидим, как это делается, но сначала будем использовать графический язык Scratch. В Scratch программы собираются из готовых блоков с помощью мыши, поэтому программирование в Scratch доступно для всех.

# 2

## Программирование в среде Scratch

### Что такое Scratch?



Рис. 2.1

**Scratch** (в переводе с английского «царапина») — это учебный язык программирования, в котором можно составлять программы с помощью мышки из цветных блоков-кирпичиков, как из деталей конструктора. Конечно, Scratch — это не специальный язык для управления роботами. Его придумали для программирования несложных игр, в которых принимают участие нарисованные персонажи — *спрайты*. Поэтому прежде чем перейти к роботам, мы поиграем с главным героем Scratch — рыжим котом (рис. 2.1).

А затем те же самые приемы применим для роботов.

Посмотрите на экран сразу после запуска Scratch (рис. 2.2).

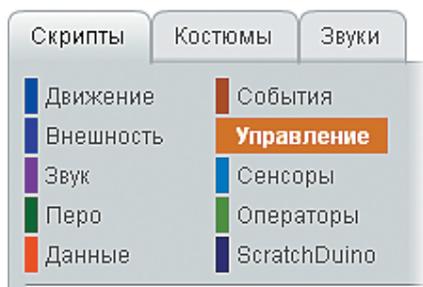


Рис. 2.2

Кот ходит по сцене — на рис. 2.2 она обозначена как область 1. В середине находится окно блоков (область 2), справа — область 3, где мы из блоков будем строить программу.

Программа для управления спрайтом состоит из **скриптов** — так в Scratch называются небольшие программки для обработки **событий**: нажатия на клавиши, щелчка мышкой и т.п.

Скрипты состояются из блоков, которые можно выбрать во вкладке «Скрипты» (**рис. 2.3**).



**Рис. 2.3**

Как видно на **рис. 2.3**, все блоки разбиты на группы: вместе объединяются блоки, решающие похожие задачи. При управлении роботами нас будут в первую очередь интересовать три группы блоков:

- «События» — это какие-либо изменения, например, нажатие на клавишу. С помощью событий мы будем управлять роботом, используя компьютер как пульт;
- «Управление» — это команды, которые позволяют многократно выполнять какие-то действия (они называются **циклами**) или выполнять действия при некотором условии (это **ветвления**);
- «Операторы» — это действия, которые можно выполнять с данными: складывать, вычитать, умножать, сравнивать и т.п.

## Как написать программу?

Давайте сделаем так, чтобы кот прошел вправо 10 шагов при нажатии на клавиатуре клавиши «стрелка направо».

Выделите группу «События» на вкладке «Скрипты» и перетащите блок **«когда клавиша ... нажата»** в область программы. В выпадающем списке выберите клавишу «стрелка направо» (**рис. 2.4**).



**Рис. 2.4**

Теперь снизу добавьте блок «идти 10 шагов» из группы «Движение», так чтобы выемка вверху этого блока состыковалась с выступающей частью предыдущего блока (рис. 2.5).

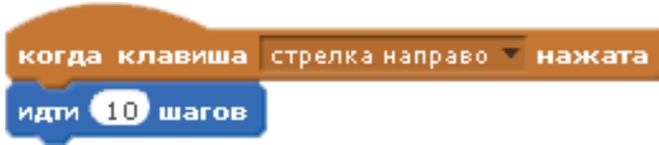


Рис. 2.5

Запустите скрипт, нажав на клавишу «стрелка направо».

Теперь измените количество шагов кота. Для этого введите в белое окошко внутри блока «идти 10 шагов» вместо числа «10» другое число, например «20». Затем проверьте работу программы. Такие изменяемые значения называются **параметрами команд**.

## Что такое циклы?

Иногда нужно заставить робота (или кота!) несколько раз сделать одну и ту же операцию. Для этого используйте в программе специальную команду, которая называется «**повторить**» (рис. 2.6).

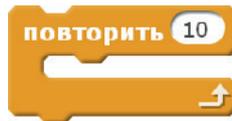


Рис. 2.6

В окошке команды введите число повторений, например 2, и пристыкуйте этот блок снизу к блоку «когда клавиша ... нажата», а команду «идти 10 шагов» поместите внутрь блока «повторить» (рис. 2.7).

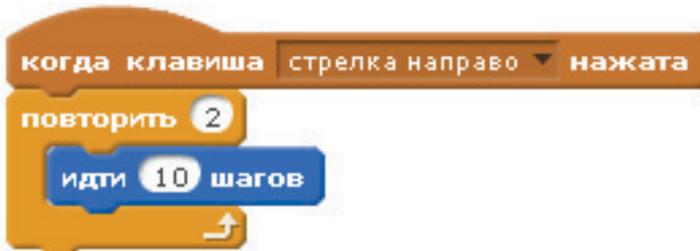


Рис. 2.7

Команды, которые находятся внутри цикла (их может быть несколько), называются **телом цикла**.

Проверьте работу этой программы. Смогли ли вы заранее определить, сколько шагов пройдет кот при нажатии на клавишу «стрелка направо»? Посмотрите, в каком месте поля можно обнаружить координаты кота (они обозначены как  $x$  и  $y$ ). Какая координата меняется при движении кота?

Теперь давайте попробуем поменять тип цикла. Из группы команд «Управление» добавьте в программу блок «**всегда**» (рис. 2.8).



Рис. 2.8

Замените им блок «**повторить 2**». Как вы думаете, остановится ли кот? Как можно догадаться об этом, не запуская программу? Запустите скрипт и проверьте, что получается.

Цикл, который мы только что применили, называется **бесконечным** — он никогда не остановится без вмешательства человека.

В работе иногда нужно срочно остановить робота, например, чтобы он не упал, дойдя до края стола. Сделаем так, чтобы наш кот останавливался при нажатии на клавишу «пробел» (она большая и ее не придется долго искать!). Для этого можно использовать третий вид цикла — цикл «**пока не ...**». Его также называют циклом с условием, потому что он останавливается, когда выполнилось какое-то условие (например, нажали на клавишу «пробел»).

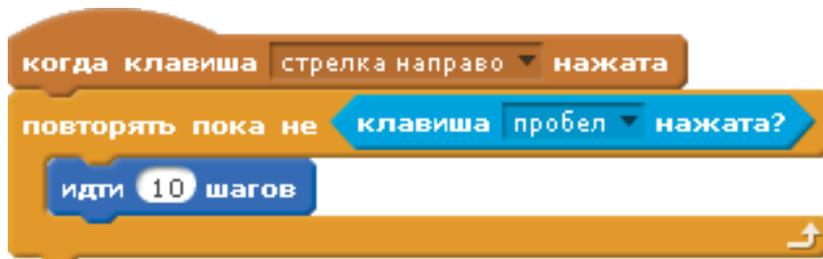
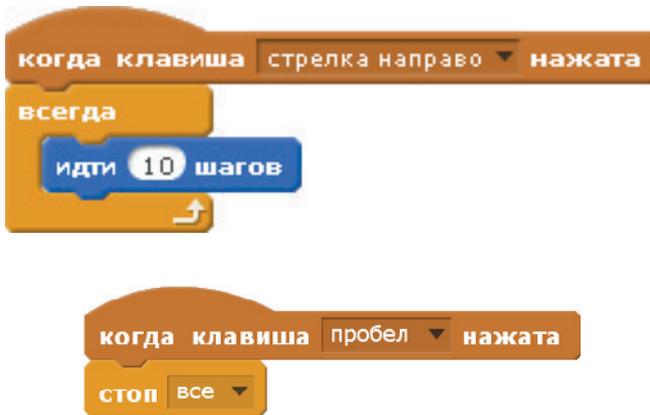


Рис. 2.9

На рис. 2.9 использован блок голубого цвета «**клавиша пробел нажата**» из группы «Сенсоры» (слово «сенсор» означает датчик). Проверьте, как работает этот скрипт.

Программы могут состоять из нескольких скриптов. Сделаем так, чтобы один скрипт задавал бесконечное движение кота (мы уже написа-

ли его раньше, см. **рис. 2.8**), а второй при нажатии на клавишу «пробел» останавливал все скрипты, то есть останавливал работу всей программы (**рис. 2.10**).



**Рис. 2.10**

Именно этот вариант мы будем использовать при управлении роботами. Он хорош тем, что по команде «**стоп все**» останавливаются сразу все скрипты в программе (а их может быть довольно много).

## Что такое переменные?

Очень часто в программе управления роботом нужно запоминать какие-то данные (числа или текст), чтобы потом их использовать. Для этого в памяти компьютера выделяются ячейки, содержание которых может меняться во время работы программы. Такие ячейки называются **переменными**. Для работы с переменными в Scratch нужно научиться:

- 1) создавать новую переменную в программе;
- 2) записывать значение в переменную (присваивать ей значение);
- 3) менять значение переменной;
- 4) использовать значение переменной для управления.

Чтобы создать переменную с именем **с**, выберите группу команд «Данные» на вкладке «Скрипты» и щелкните по кнопке «Создать переменную». Появляется окно, в которое нужно ввести имя переменной (**рис. 2.11**).

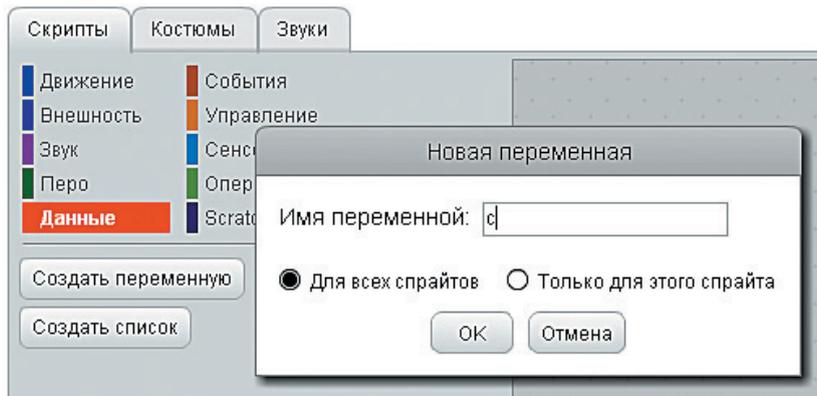


Рис. 2.11

Записать значение переменной можно с помощью команды **«задать значение»** из группы «Данные». Имя переменной выберите из выпадающего списка (в нем будут перечислены все доступные переменные), а нужное значение-параметр (например, 0) введите в белое поле (рис. 2.12).



Рис. 2.12

Для изменения значения переменной на некоторую величину тоже есть специальная команда. Так можно увеличить переменную **c** на 1:



Рис. 2.13

Подумайте, как можно уменьшить значение переменной на 5.

## Что такое ветвления?

Программируя роботов, нужно учитывать, что при выполнении задания они могут столкнуться с самыми разными условиями, которые заранее неизвестны. Поэтому программисты всегда стараются предусмотреть, что должен делать робот в каждом возможном случае.

В языках программирования есть специальные команды для выбора нужного варианта действий. Такие команды называются **условными операторами** или **ветвлениями**. Например, при выполнении скрипта, показанного на **рис. 2.14**, кот делает вперед 10 шагов, если значение переменной **с** равно 3, или мяукает, если значение этой переменной другое.

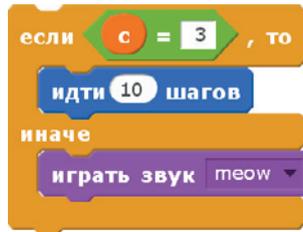


Рис. 2.14

Здесь используется новый блок из группы «Управление» (**рис. 2.15**).

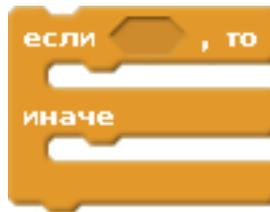


Рис. 2.15

Он позволяет выбрать один из двух вариантов действий.

Для того чтобы записать **условие** (сравнить значение переменной **с** и число 3), добавьте блок  из группы «Операторы». Вы можете самостоятельно изучить другие блоки из этой группы.

Обратите внимание, что имя переменной **с** находится в оранжевом кружке  — его нужно перетащить в окно блока  из группы «Данные»; там вы увидите такие же кружочки для всех созданных переменных.

Если условие **истинно** («верно», «выполняется»), то работает первая часть команды, расположенная между словами «**если**» и «**иначе**». Если условие **ложно** («неверно», «не выполняется»), то работает вторая часть, идущая после слова «**иначе**».

Иногда бывает так, что при невыполненном условии ничего делать не надо — все и так хорошо. Для этого в группе «Управление» есть сокращенная форма ветвления, которая состоит только из команды «**если**»; слова «**иначе**» там нет (**рис. 2.16**).

## Оглавление

Здравствуйте! . . . . .	3
От авторов . . . . .	4
Благодарности . . . . .	6
<b>Глава 1. Роботы и робототехника . . . . .</b>	<b>7</b>
Зачем нужны роботы? . . . . .	7
Что такое робототехника? . . . . .	8
Из чего состоит робот? . . . . .	9
Какой язык понимают роботы? . . . . .	11
<b>Глава 2. Программирование в среде Scratch . . . . .</b>	<b>12</b>
Что такое Scratch? . . . . .	12
Как написать программу? . . . . .	13
Что такое циклы? . . . . .	14
Что такое переменные? . . . . .	16
Что такое ветвления? . . . . .	17
Как комбинировать блоки? . . . . .	19
<b>Глава 3. Робоплатформа ScratchDuino . . . . .</b>	<b>21</b>
Знакомимся с робоплатформой. . . . .	22
Изучаем стандартный комплект робота. . . . .	25
Собираем и подключаем робота . . . . .	27
Запускаем среду управления роботом. . . . .	28
<b>Глава 4. Управление роботами . . . . .</b>	<b>31</b>
Что такое управление? . . . . .	31
Как выполняются команды? . . . . .	32
Как управлять робоплатформой? . . . . .	32
Что такое разомкнутые системы управления? . . . . .	34

Управляем движением робота с клавиатуры. . . . .	35
Управляем движением робота по прямой . . . . .	37
Что такое погрешность? . . . . .	39
Управляем движением робота с поворотами. . . . .	41
Проверь себя! . . . . .	42
<b>Глава 5. Обратная связь . . . . .</b>	<b>45</b>
Что такое обратная связь? . . . . .	45
Как связать датчик с компьютером? . . . . .	47
Что такое датчик касания? . . . . .	47
Что такое датчик освещенности? . . . . .	48
Что такое датчик линии? . . . . .	48
Что такое «инфракрасный глаз»? . . . . .	48
Как крепятся датчики? . . . . .	49
Как калибруются датчики? . . . . .	51
Используем датчик освещенности . . . . .	51
⚙ Калибровка датчиков освещенности . . . . .	52
⚙ Движение робота в зависимости от освещенности . . . . .	53
⚙ Поворот робота к свету . . . . .	55
⚙ Перетягивание светового каната . . . . .	58
Используем датчик касания . . . . .	59
⚙ Обнаружение препятствия . . . . .	59
⚙ Стучимся в дверь . . . . .	60
Используем датчик линии . . . . .	61
⚙ Обнаружение белого листа . . . . .	63
⚙ «Ох, доска качается, а я не упаду» . . . . .	64
⚙ Движение вдоль линии на двух датчиках . . . . .	65
⚙ Обнаружение и прохождение перекрестка . . . . .	66
⚙ Поворот на перекрестке . . . . .	67
Используем датчик «инфракрасный глаз» . . . . .	70
⚙ Пугливый робот. . . . .	72
Проверь себя! . . . . .	74
<b>Глава 6. Автономный робот. . . . .</b>	<b>76</b>
Что такое автономность? . . . . .	76

Как программировать Arduino? . . . . .	77
Знакомимся с Arduino IDE . . . . .	77
Управляем моторами . . . . .	80
⚙ Включение и выключение моторов. . . . .	81
⚙ Исследование текста программы . . . . .	86
⚙ Изменение направления вращения моторов . . . . .	88
⚙ Изменение программы движения. . . . .	90
⚙ Изучение поворотов . . . . .	90
Изменяем скорость робота. . . . .	92
⚙ Импульсное управление . . . . .	92
⚙ Движение с уменьшенной скоростью . . . . .	93
Используем датчики. . . . .	94
⚙ Калибровка датчиков . . . . .	95
⚙ Запуск программы кнопкой . . . . .	97
⚙ Кнопки для запуска и остановки программы. . . . .	99
Проверь себя! . . . . .	99
ПРИЛОЖЕНИЕ 1. Загрузка с флеш-накопителя . . . . .	101
ПРИЛОЖЕНИЕ 2. Восстановление прошивки ScratchDuino . . . . .	103
ПРИЛОЖЕНИЕ 3. Установка Arduino IDE. . . . .	106
ПРИЛОЖЕНИЕ 4. Установка и запуск ArduBlock . . . . .	108

# СТАТЬ

## ИЗОБРЕТАТЕЛЕМ ЛЕГКО!

Хочешь создать своего персонального робота, учиться и играть вместе с ним? Нет ничего проще! Книги серии **«РОБОФИШКИ»** познакомят тебя с удивительным миром робототехники.

**«Конструируем роботов на ScratchDuino®. Первые шаги»** – практическое руководство для тех, кто уже освоил начальные ступени робототехнического творчества и хочет двигаться дальше.

Прочитав эту книгу, ты:

- ◆ почувствуешь себя настоящим программистом, когда будешь создавать программы в средах разработки **Scratch** и **Arduino IDE**;
- ◆ со свойственным исследователю любопытством заглянешь внутрь робота и изучишь его возможности;
- ◆ с помощью датчиков и микроконтроллера научишь робота «чувствовать» и «думать» самостоятельно;
- ◆ ощутишь себя командиром, управляющим роботом через компьютер.

Не пропусти интересные проекты на платформах **LEGO® MINDSTORMS® Education EV3** и **Arduino!**

info@pilotLZ.ru  
www.pilotLZ.ru

