

**11-ое включение БелКУФ
(Белгородской киностудии
учебных фильмов)**

14 ноября 2020 года

**6-е занятие Клуба ИТР
(Клуб Информационных
Технологий и Робототехники)
Конструируем роботов на
ScratchDuino**



Конструируем роботов на ScratchDuino



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.

Конструируем роботов на ScratchDuino

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!

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)

Конструируем роботов на ScratchDuino



Робот Asimo
(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 уничтожать 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 летающие 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 будут 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 видеокamеры. Они могут «узнавать» простые предметы, определять их расположение в пространстве и обнаруживать невидимые части, используя информацию из своей базы данных.

Конструируем роботов на ScratchDuino



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).

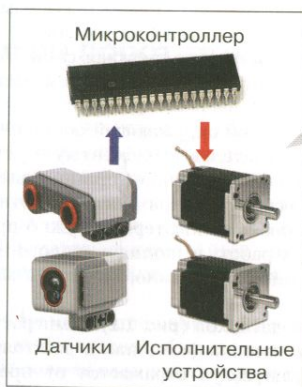


Рис. 1.2

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.

Конструируем роботов на ScratchDuino

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).

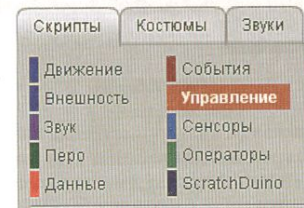


Рис. 2.3

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).

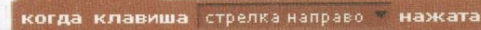


Рис. 2.4

Конструируем роботов на ScratchDuino

Теперь снизу добавьте блок «идти 10 шагов» из группы «Движение», так чтобы выемка сверху 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 пристыкните 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»).

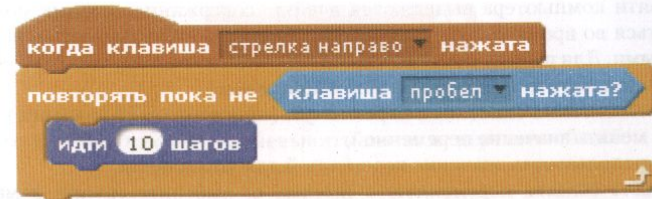


Рис. 2.9

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-

Конструируем роботов на ScratchDuino

16

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». Появится окно, в которое нужно ввести имя переменной (рис. 2.11).

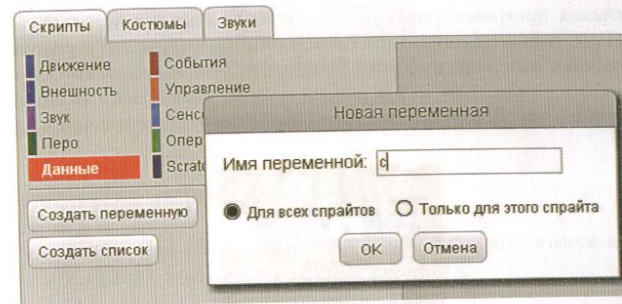


Рис. 2.11

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 то есть специальная команда. Так можно увеличить переменную **с** на 1:



Рис. 2.13

Подумайте, как можно уменьшить значение переменной на 5.

Что такое ветвления?

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

Конструируем роботов на ScratchDuino

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.

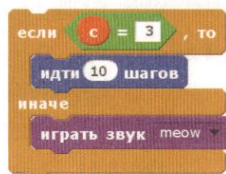


Рис. 2.14

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 **с = 3** из группы «Операторы». Вы можете самостоятельно изучить другие блоки из этой группы.

Обратите внимание, что имя переменной **с** находится в оранжевом кружке **с** — его нужно перетащить в окно блока **с = 3** 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**).



Рис. 2.16

Как комбинировать блоки?

В программе можно объединять управляющие блоки, стыкуя их друг за другом или вкладывая один в другой (**рис. 2.17**).



Рис. 2.17

Сначала переменной **с** присваивается значение 0. Затем начинается работа цикла, пристыкованного снизу. Цикл завершается, когда значение переменной **с** станет равным 4. Можно заметить, что в конце цикла (на каждом шаге!) значение **с** увеличивается на 1, то есть переменная **с** будет последовательно принимать значения 0, 1, 2, 3 и 4. Чтобы переменная **с** стала равной 4, нужно 4 раза увеличить ее на 1, поэтому цикл выполнится ровно 4 раза, а потом остановится.

Конструируем роботов на ScratchDuino

В начале тела цикла стоит ветвление: если значение `c` меньше 2 (первые два раза), то кот идет вперед, а иначе (последние два раза) идет назад и мяукает.

Для того чтобы кот не бегал слишком быстро, мы добавили в конце цикла команду «ждать» с временем выполнения 1 с (рис. 2.18).



Рис. 2.18

После каждого шага цикла (и каждого изменения переменной `c`) кот или робот будет одну секунду стоять на месте.

3

Робоплатформа ScratchDuino

Конечно, интересно читать про роботов и их использование, но намного интереснее самим научиться управлять таким аппаратом и исследовать его возможности. Именно этим в дальнейшем мы и займемся. Настала пора представить робота, с которым вы будете проводить исследования. Мы будем работать с роботом ScratchDuino, который был придуман и создан российскими специалистами. Его полное и правильное название — **ScratchDuino.Робоплатформа** (да, вот так, с точкой посередине!). Чтобы не путаться с точками в предложениях, мы будем чаще всего называть его просто **робот** или **робоплатформа**.

Название «ScratchDuino» образовано от двух названий: Scratch + Arduino.

Первая часть — Scratch — очень простая система программирования для начинающих, с которой мы познакомились в предыдущей главе. Scratch относится к **свободному программному обеспечению**: вы можете не только бесплатно использовать его, но даже изменять по своему усмотрению и распространять улучшенные версии.

Вторая часть — Arduino — электронный конструктор, который новички применяют для создания простых электронных устройств. Программируют Arduino также с помощью свободного программного обеспечения. Любой желающий может самостоятельно изготовить устройства на основе Arduino, дополнить их новыми блоками, совершенствовать программы, ничего не платя разработчикам и не нарушая при этом авторских прав.

Давайте теперь познакомимся с нашим роботом поближе.

Конструируем роботов на ScratchDuino

Знакомимся с робоплатформой

На рис. 3.1 показан набор «ScratchDuino.Робоплатформа».

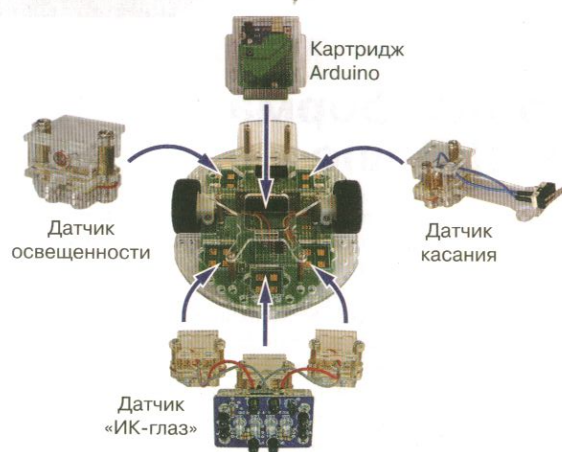


Рис. 3.1

В него входят следующие элементы.

- **Платформа с колесами** — это значит, что робот может ездить по столу или по полу. Каждое колесо вращается с помощью **мотор-редуктора** — так называют устройство, состоящее из мотора и редуктора (рис. 3.2).

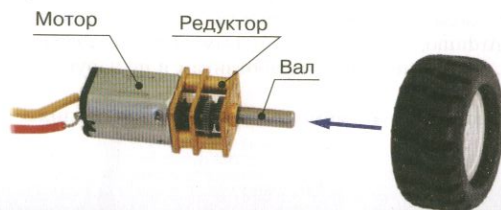


Рис. 3.2

Редуктор уменьшает скорость вращения вала мотора так, чтобы одновременно увеличить мощность — это позволяет такому маленькому мотору толкать нашего большого робота.

- Съемные **датчики** — это «органы чувств» нашего робота, с помощью которых он будет получать данные о внешнем мире, как мы получаем информацию с помощью зрения и слуха. Все датчики крепятся к платформе на магнитах, поэтому их легко устанавливать/снимать, менять местами. Попробуйте!
- Ну и самое главное, без чего робот вовсе не робот, — **микроконтроллер**, «мозг» робота. Как мы уже говорили, наш робот использует плату из конструктора Arduino, а если совсем точно, то плату Arduino UNO* с микроконтроллером ATmega328.

На плате Arduino собраны вместе микроконтроллер и контакты, к которым можно подключать любые устройства: моторы, лампочки, датчики. Есть также вход для питания и USB-разъем. Вся плата упакована в пластиковый корпус (рис. 3.3).

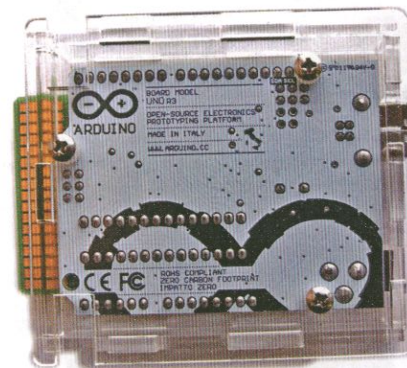


Рис. 3.3

Для того чтобы робот заработал, все его части нужно соединить между собой. Давайте разберемся, как же работает этот сложный механизм наш робот.

Во-первых, откуда-то должна поступать энергия, электрический ток (питание). Во-вторых, с помощью этой энергии должны крутиться колеса и работать датчики.

* В мире Arduino существует много типов плат, каждая из которых имеет свое название.

Конструируем роботов на ScratchDuino

Питание поступает к робоплатформе через плату Arduino, которая подключена к USB-разъему компьютера (в этом случае робот питается от компьютера) или к батарейке. Через электронные элементы и контакты платы Arduino ток подводится к моторам, датчикам, лампочкам (рис. 3.4). Программа управления (вы будете составлять ее сами!) определит, в какие именно моменты электрический ток поступает (или не поступает) к тем или иным частям робота.

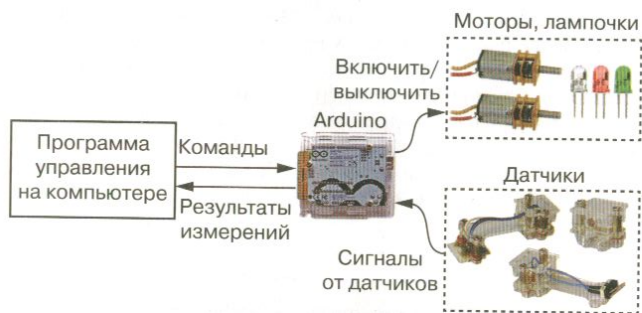


Рис. 3.4

На платформе есть пять посадочных мест — специально оборудованных площадок, куда на магнитах крепятся датчики (рис. 3.5). Каждое место имеет четыре контакта: через два контакта на датчик подается питание, а через два других от датчика к плате Arduino передаются результаты измерений.

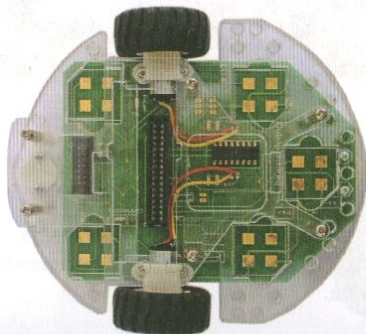


Рис. 3.5

Корпус робота специально сделан прозрачным для того, чтобы можно было увидеть все его составные части. Кроме этого, прозрачный корпус подчеркивает основную идею проекта ScratchDuino — открытость. Все чертежи и схемы робота ScratchDuino, все тексты программ для него находятся в свободном доступе. Как мы уже говорили, все желающие могут сами построить, изменить и улучшить робота ScratchDuino!

Итак, теперь вы знакомы с робоплатформой и готовы продолжить работу с ней.

Изучаем стандартный комплект робота

На рис. 3.6 показан стандартный набор деталей конструктора ScratchDuino.



Рис. 3.6. Комплект «ScratchDuino.Робоплатформа»

В комплект кроме самой робоплатформы с мотор-редукторами и колесами входят:

- два датчика касания;
- два датчика освещенности;
- два датчика линии;

Конструируем роботов на ScratchDuino

- самый большой датчик — «инфракрасный глаз», позволяющий роботу обнаруживать препятствия прямо перед ним, слева и справа.



Рис. 3.7. Правильное расположение трех магнитов на датчике

Каждый датчик вмонтирован в пластиковый корпус, который снабжен тремя магнитными креплениями (рис. 3.7).

Иногда магниты-крепления могут случайно прилипнуть к другому датчику или металлической детали робота. Поэтому следите, чтобы все магниты были на нужных позициях; если увидите, что это не так, верните магниты на место.

Помните, мы говорили о том, что в мире Arduino можно самостоятельно изменять и дополнять устройства? Вот для таких случаев в набор включили два пустых корпуса, в которые можно упаковать нестандартные датчики (например,

датчик звука или датчик расстояния).

Робоплатформа соединяется с компьютером с помощью USB-кабеля; по нему передаются питание и команды роботу, через него же поступают сигналы от датчиков.

Для управления роботом можно использовать и беспроводное соединение (Bluetooth). В этом случае он получает питание от батареек (для роботов есть два блока питания: на четыре батарейки размера АА и на одну батарейку типа «Крона»), а сигналы управления и данные датчиков передаются через специальное устройство связи — Bluetooth-адаптер, который вставляется в USB-порт компьютера.

Если вы перевернете робоплатформу вверх колесами, то на нижней стороне увидите переключатели (их еще называют **джамперами**). Положение этих переключателей задает способ подключения робота к компьютеру — по USB-кабелю или через Bluetooth (рис. 3.8).

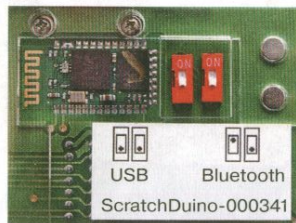


Рис. 3.8

В стандартный комплект входит также загрузочный флеш-накопитель, или флешка (рис. 3.9).

Загрузив компьютер с этой флешки, вы сразу получаете весь набор программно-обеспечения, необходимый для работы с роботом (рис. 3.10).



Рис. 3.9

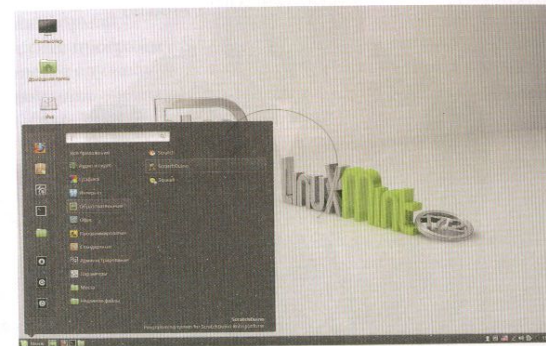


Рис. 3.10

О том, как использовать загрузочный флеш-накопитель, вы узнаете из приложения 1. Теперь вы знакомы с составом конструктора. Пришла пора собрать робота и подключить его к компьютеру.

Собираем и подключаем робота

Сначала вставьте прозрачный корпус с платой Arduino (он называется «картридж») в разъем на платформе (рис. 3.11).

СОВЕТ

Когда будете вставлять картридж в платформу, держите их в руках.

Конструируем роботов на ScratchDuino

При этом лучше держите обе части в руках: крепления моторов сделаны из пластика и могут сломаться, если сильно нажать на платформу (например, если вы поставите ее на стол и будете с силой давить на картридж сверху).

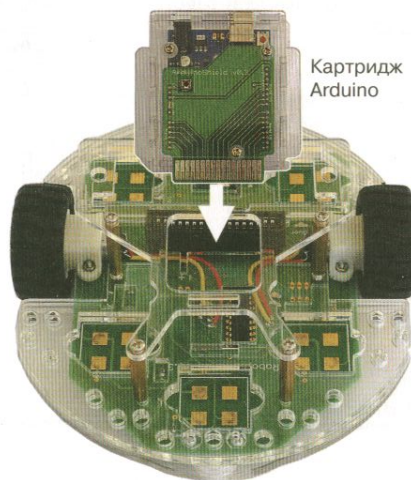


Рис. 3.11

Обратите внимание, что в платформе вырезы сделаны таким образом, чтобы картридж было невозможно вставить неправильно.

Мы будем использовать самый простой способ подключения робота — через USB-кабель. Нам не нужны батарейки, потому что робот получает питание по этому кабелю прямо от компьютера.

Убедитесь в том, что переключатели под робоплатформой находятся в положении «USB-подключение» (см. рис. 3.8). Затем с помощью USB-кабеля соедините картридж Arduino с компьютером.

Запускаем среду управления роботом

Наш робот может управляться двумя способами:

1) с помощью компьютера, который благодаря установленной на нем программе управления посылает роботу команды и принимает от него

данные датчиков; в этом случае робот должен быть все время соединен с компьютером USB-кабелем или через беспроводное соединение (Bluetooth);

2) с помощью микроконтроллера (напомним, что он находится на плате Arduino), в память которого загружена программа управления; тогда робот может действовать автономно, без компьютера.

Пока мы будем использовать первый способ, составляя программы в среде Scratch, точнее, в ее дополненном варианте, который называется «ScratchDuino» и содержит специальную группу команд для работы с робоплатформой.

Запустите программу ScratchDuino. Одновременно с окном программы откроется служебное окно — **Панель управления**. Если вы установили программное обеспечение правильно, робот будет обнаружен автоматически* (рис. 3.12).

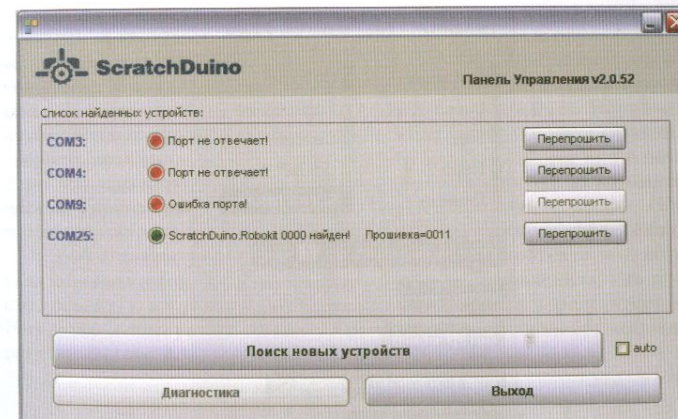


Рис. 3.12

Робот связывается с компьютером через **порт** — так называется соединение, через которое два устройства обмениваются данными. В нашем примере (см. рис. 3.12) этот порт обозначен как COM25. Если к этому же компьютеру подключить другого робота, для работы с ним программа создаст новый порт (с другим номером).

* С помощью «Панели управления» можно изменять программное обеспечение робота (прошивку). Если вам это интересно, загляните в приложение 2.

Конструируем роботов на ScratchDuino

СОВЕТ

Для того чтобы программа правильно определила роботоплатформу, перед подключением надо снять с платформы все датчики.

Вы уже знаете, что мы будем программировать робота с помощью среды ScratchDuino. Это версия популярной среды программирования для начинающих Scratch, которая была дополнена (ведь это свободное программное обеспечение!) группой команд ScratchDuino для управления нашей роботоплатформой (рис. 3.13).

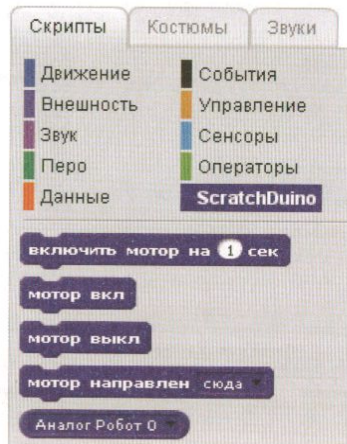


Рис. 3.13. Группа команд ScratchDuino

С тем, как работать в Scratch, вы познакомились в главе 2. Уже совсем скоро с помощью ScratchDuino вы научитесь создавать программы для управления роботом, а также загружать их в память микроконтроллера Arduino; так можно сделать режим робота полностью автономным (не зависящим от компьютера). Но до этого нас ждет много интересной исследовательской работы. Вперед!

4

Управление роботами

Что такое управление?

С древних времен человек хотел использовать предметы и силы природы в своих целях, то есть управлять ими. Управлять можно неодушевленными предметами (например, перекачивая камень на другое место), животными (дрессируя их), людьми (как начальник управляет подчиненными). Множество задач управления связано с техническими системами — автомобилями, кораблями, самолетами, станками. Например, нужно поддерживать заданный курс корабля, высоту полета самолета, скорость вращения двигателя, температуру в холодильнике или в печи. Если эти задачи решаются без участия человека, говорят об **автоматическом управлении**. Наука, которая пытается ответить на вопрос «Как нужно управлять?», называется **теорией автоматического управления**.

В любой задаче управления можно выделить два объекта: тот, которым управляют (он так и называется — **объект управления**), или просто «объект»), и тот, который управляет (он называется **регулятором**). Регулятор действует на объект управления, а тот выполняет команды регулятора (рис. 4.1).



Рис. 4.1

Регулятор стремится решить какую-то задачу, связанную с объектом, например, перевести робота в заданную точку. Его задача — изменить состояние объекта управления нужным образом.

11-ое включение БелКУФ
(Белгородской киностудии
учебных фильмов)
14 ноября 2020 года

6-е занятие Клуба ИТР
(Клуб Информационных Технологий и
Робототехники)
Конструируем роботов на ScratchDuino



**Конец 11 включения БелКУФ
ПРОДОЛЖЕНИЕ СЛЕДУЕТ**