

# THE ARDUINO INVENTOR'S GUIDE

LEARN ELECTRONICS BY MAKING  
10 AWESOME PROJECTS

BY BRIAN HUANG  
AND DEREK RUNBERG



**no starch  
press**

SAN FRANCISCO

# ARDUINO ДЛЯ ИЗОБРЕТАТЕЛЕЙ

ОБУЧЕНИЕ ЭЛЕКТРОНИКЕ  
НА 10 ЗАНИМАТЕЛЬНЫХ ПРОЕКТАХ

БРАЙАН ХУАНГ, ДЕРЕК РАНБЕРГ



Санкт-Петербург  
«БХВ-Петербург»  
2019

УДК 004  
ББК 32.973.26  
Х98

**Хуанг, Б.**

Х98     Arduino для изобретателей. Обучение электронике на 10 занимательных проектах: Пер. с англ. / Б. Хуанг, Д. Ранберг. — СПб.: БХВ-Петербург, 2019. — 288 с.: ил.

ISBN 978-5-9775-3972-2

В книге подробно рассмотрено 10 занимательных проектов с платой Arduino Uno (светофор, светодиодный экран, светочувствительный ночник, мини-теплица, мобильный робот, миниатюрное пианино и др.). Описаны принципы работы и взаимодействие различных электронных компонентов, чтение принципиальных и монтажных схем, создание и тестирование прототипов с помощью безопасной макетной платы. Показано, как собирать электрические схемы, разрабатывать программный код и создавать готовые конструкции. В каждом проекте приведены советы по его модификации и расширению возможностей. Приведены шаблоны корпусов и деталей, а также пошаговые фотографии их изготовления и сборки. На сайте издательства находятся исходные коды примеров из книги, шаблоны для конструкций проектов, а также коды для дальнейшего экспериментирования с проектами.

*Для радиолюбителей*

УДК 004  
ББК 32.973.26

**Группа подготовки издания:**

|                      |                             |
|----------------------|-----------------------------|
| Руководитель проекта | <i>Игорь Шишигин</i>        |
| Зав. редакцией       | <i>Екатерина Капальгина</i> |
| Компьютерная верстка | <i>Людмилы Гауль</i>        |
| Оформление обложки   | <i>Марины Дамбиевой</i>     |

© 2017 by SparkFun Electronics. Title of English-language original: The Arduino Inventor's Guide: Learn Electronics by Making 10 Awesome Projects, ISBN 978-1-59327-652-2, published by No Starch Press. Russian-language edition copyright © 2018 by BHV. All rights reserved.

© 2017 by SparkFun Electronics. Название английского оригинала: The Arduino Inventor's Guide: Learn Electronics by Making 10 Awesome Projects, ISBN 978-1-59327-652-2, опубликовано No Starch Press. Издание на русском языке © 2018 by BHV. Все права защищены.

Подписано в печать 31.07.18.  
Формат 84×108<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 30,24.  
Тираж 1500 экз. Заказ №  
«БХВ-Петербург», 191036, Санкт-Петербург, Гончарная ул., 20.

Отпечатано в соответствии с предоставленными материалами в ООО «ИПК Парето-Принт»,  
170546, Тверская область, Промышленная зона Боровлево-1, комплекс № 3А, www.pareto-print.ru

ISBN 978-1-59327-652-2 (англ.)  
ISBN 978-5-9775-3972-2 (рус.)

© 2017 by SparkFun Electronics  
© Перевод на русский язык, оформление. ООО «БХВ-Петербург», ООО «БХВ», 2019

---

# ПОСВЯЩАЕТСЯ

---

ЛИНДСИ ДАЙМОНД – ТЫ НАШ ЗАМЕЧАТЕЛЬНЫЙ РУКОВОДИТЕЛЬ И НАСТАВНИК!

ВСЕЙ КОМАНДЕ ИНТЕРНЕТ-МАГАЗИНА SPARKFUN!

ВСЕМ ЛЮБИТЕЛЯМ САМОДЕЛЬНЫХ КОНСТРУКЦИЙ И ИЗОБРЕТАТЕЛЯМ,  
КОТОРЫХ МЫ СМОГЛИ ВДОХНОВИТЬ НА ТВОРЧЕСТВО!

---

# О КОМПАНИИ SPARKFUN ELECTRONICS

---

Интернет-магазин розничной торговли SparkFun Electronics производит и продает устройства и компоненты, которые используются во многих любительских проектах, промышленных образцах и даже на Международной космической станции.

Мысль создать такую компанию пришла в 2003 году студенту Колорадского университета (г. Боулдер) Натану Зайдлю (Nathan Seidle), когда он сжег печатную плату. В те времена достать печатную плату было весьма трудной задачей. Для этого нужно было передать по факсу номер своей кредитной карточки в компанию, находящуюся в другой стране, после чего ожидать получения заказанной платы в течение шести-восьми недель. Натан решил, что он может сделать этот процесс более удобным, и задался целью создать компанию по поставке электронных деталей, устройств, компонентов и пр. Так и появилась компания SparkFun.com, которая в настоящее время предлагает свыше 3 тыс. различных наименований деталей всевозможных параметров и функционала для использования в цифровых электронных проектах. От базовых плат Arduino до модулей GPS — все это, вместе со всей необходимой документацией, доступно для покупки любому желающему в магазине SparkFun.

Отдел образования компании SparkFun разрабатывает курсы и учебные планы для студентов профильных специальностей, а также предлагает программы повышения профессионального уровня для преподавателей, занятых в сфере электроники и информационных технологий. Этот отдел является зачинателем многих инициатив в области вычислительной техники и любительских проектов, которые с успехом применяются в обучении.

Дополнительная информация о компании SparkFun и ее отделе образования доступна в Интернете на сайтах <https://www.sparkfun.com/> и <http://www.sparkfuneducation.com/>.

---

# ОБ АВТОРАХ

---

В одно время Брайан Хуанг (Brian Huang) и Дерек Ранберг (Derek Runberg) работали учителями. Брайан преподавал физику в средней школе и одновременно увлекался робототехникой, а Дерек работал учителем технического образования в восьмилетней школе — его любимым коньком было добиться от учеников максимального использования ими своих возможностей.

Они пришли к своему сегодняшнему статусу разными путями, разнятся и их подходы к изучению программирования и электроники, их преподавательская философия и их точки зрения на процесс познания учениками преподаваемых им дисциплин, так что нельзя сказать, что взгляды их всегда совпадают. Тем не менее они надеются, что совместно написанная ими книга сослужит вам хорошую службу и поможет встать на путь поиска приключений в мире изобретательства.

## Комментарий от Брайана

---

Хотя у меня формальное инженерное образование (я изучал электротехнику в институте), мое образование в основном концентрировалось на теории, эмулировании и моделировании, и меня никогда не учили, как пользоваться паяльником, выполнять токарные работы или создавать реальные проекты. По окончании института по будням я работал инженером, а по выходным волонтерствовал в Музее науки штата Миннесота (Science Museum of Minnesota). Именно работая в музее, я и обнаружил в себе любовь к преподаванию. Мне была предоставлена возможность вдохновлять детей на любопытство, задавать вопросы и размышлять о мире вокруг нас. Мой опыт работы в музее побудил меня изменить карьеру — получить степень магистра в области обучения и стать учителем физики в средней школе.



Мы с Дерекком дополняем опыт и квалификацию друг друга. Эта книга является квинтэссенцией нашего опыта преподавания и использования Arduino для

обучения. Как говорит Дерек, Arduino — просто используемое в наших проектах устройство более высокого уровня, чем раньше. Это было давно известно преподавателям и студентам Программы комплексной телекоммуникации Нью-йоркского университета. Точно так же помещение электронного устройства в корпус или просто ограничение его видимости каким-либо образом немедленно меняет способ взаимодействия с ним. Например, мячик для настольного тенниса, которым накрывают светодиод проекта, рассеивает его свет, что немедленно воздействует на способ нашего восприятия всего проекта. Своеобразное рассеивание света через оболочку мячика и одновременное ее подсвечивание воздействуют на наши эмоции совсем по-другому, чем простой горящий светодиод на макетной плате.

Мы основательно продумали, каким образом сделать изучение электроники и программирования доступными для всех. Мы надеемся, что проекты из этой книги помогут вам открыть в себе талант изобретателя.

## Комментарий от Дерекка

---

В отличие от Брайана, у меня нет формального образования по части электроники или программирования — все мои знания в этих областях получены путем самообразования. Я был учителем технического образования в восьмилетней школе, перед которым поставили задачу создать программу такого образования для XXI столетия. Составной частью моего видения этой программы была электроника, и на протяжении трех лет модуль Arduino, а затем язык Processing для его программирования заняли центральное место в моем классе. Я изучил Arduino с тем, чтобы предоставлять своим ученикам доступ к технологии, которой они могли бы управлять и с помощью которой могли бы самостоятельно создавать проекты. Мне пришлось также самому изучить электронику и программирование, чтобы иметь возможность обучать своих учеников этим предметам.

Многие проекты, представленные в этой книге, основаны непосредственно на моем опыте обучения Arduino. Мои ученики изучали программирование и электронику в силу необходимости — чтобы иметь возможность реализовать свои идеи, а не потому, что я, их учитель, заставлял их делать это. Я надеюсь, что мой вклад в эту книгу завоюет уважение моих учеников и предоставит электронику и программирование в таком формате, который пробудит также и ваше воображение.



## О техническом редакторе английского издания

---

Даниэль Хайенш (Daniel Hienzsch) является основателем компании Rheingold Heavy, которая предоставляет учебные материалы студентам и любителям-электронщикам. Раньше он в течение 20 лет работал в сфере информационных технологий, включая 10 лет в качестве ИТ-директора инвестиционного банка.

Дан — страстный приверженец образования, и он создал компанию Rheingold Heavy, имея целью обеспечить общество любителей-электронщиков материалами, которых так не хватало ему самому, когда начинал заниматься электроникой и информационными технологиями. Он также сертифицированный инструктор по плаванию с аквалангом.



---

# ОГЛАВЛЕНИЕ

---

|   |            |
|---|------------|
| <b>Введение</b> .....   | <b>XXI</b> |
| О чем эта книга?.....   | XXII       |
| Почему Arduino?.....  | XXII       |
| Чем эта книга отличается от других? .....                         | XXIII      |
| Необходимые компоненты и материалы .....                          | XXIII      |
| Необходимые инструменты.....                                      | XXV        |
| Компьютер.....  | XXV        |
| Состав книги .....  | XXVI       |
| Интернет-ресурсы .....  | XXVII      |
| Распространяем информацию: делимся результатами своей работы..... | XXVII      |
| <br>  |            |
| <b>Основы электроники.....</b>                                    | <b>1</b>   |
| Электричество: ток, проводимость и основная терминология .....    | 2          |
| Что такое электричество? .....                                    | 2          |
| Типы электрического тока .....                                    | 3          |
| Что такое цепь?.....  | 3          |
| Закон Ома .....   | 4          |
| Модель электрического тока: вода в трубе.....                     | 4          |
| Принципиальные схемы .....  | 4          |
| Создание прототипов схем.....                                     | 5          |
| Дискретные компоненты и адаптерные платы .....                    | 7          |
| Аналоговая и цифровая электроника .....                           | 8          |
| Что такое микроконтроллер? .....                                  | 8          |

## **Проект 1. Начало работы с Arduino..... 11**

|   |    |
|---|----|
| Необходимые компоненты .....                        | 12 |
| О плате Arduino .....                               | 12 |
| Доступная аппаратная платформа.....                 | 12 |
| Плата RedBoard компании SparkFun.....               | 13 |
| Установка Arduino IDE и драйверов.....              | 14 |
| Установка под Windows .....                         | 15 |
| Установка под OS X .....                            | 16 |
| Установка под Linux .....                           | 18 |
| Краткая экскурсия по среде разработки Arduino ..... | 18 |
| Изменение настроек по умолчанию .....               | 19 |
| Первое подключение Arduino к компьютеру.....        | 20 |
| Указание подключенной платы в IDE .....             | 21 |
| Выбор последовательного порта COM .....             | 22 |
| Программа «Здравствуй, мир!» для Arduino .....      | 23 |
| Поиск и устранение основных проблем с Arduino.....  | 24 |
| Анатомия скетча Arduino .....                       | 25 |
| Ключевые элементы скетча .....                      | 26 |
| Функция <i>setup()</i> .....                        | 27 |
| Функция <i>loop()</i> .....                         | 28 |
| Наш первый аппаратный компонент.....                | 29 |
| Идем дальше... .....                                | 30 |
| Экспериментируем с кодом.....                       | 30 |
| Модифицируем схему .....                            | 30 |
| Сохранение скетча .....                             | 31 |

## **Проект 2. Домашний светофор..... 33**

|   |    |
|---|----|
| Необходимые компоненты, инструменты и материалы ..... | 34 |
| Электронные компоненты .....                          | 34 |
| Прочие инструменты и материалы .....                  | 35 |
| Новый компонент: резистор .....                       | 36 |
| Создаем прототип светофора .....                      | 38 |

|   |    |
|---|----|
| Подключаем красный светодиод .....              | 38 |
| Подаем питание на макетную плату .....          | 39 |
| Добавляем желтый и зеленый светодиоды.....      | 41 |
| Программируем светофор.....                     | 41 |
| Проверьте параметры среды разработки.....       | 41 |
| Создаем переменные для номеров выводов.....     | 41 |
| Создаем функцию <i>setup()</i> .....            | 43 |
| Создаем функцию <i>loop()</i> .....             | 43 |
| Загружаем скетч в Arduino .....                 | 44 |
| Делаем светофор автономным .....                | 45 |
| Создаем корпус для светофора.....               | 46 |
| Делаем картонный корпус.....                    | 47 |
| Делаем линзы для светофора .....                | 50 |
| Делаем затенители .....                         | 51 |
| Вставляем светодиоды и подключаем Arduino ..... | 52 |
| Идем дальше. . . .....                          | 53 |
| Экспериментируем с кодом.....                   | 53 |
| Модифицируем схему.....                         | 54 |

### **Проект 3. Десятипиксельный анимационный дисплей ..... 57**

|   |    |
|---|----|
| Необходимые компоненты, инструменты и материалы ..... | 58 |
| Электронные компоненты .....                          | 58 |
| Прочие инструменты и материалы .....                  | 59 |
| Создаем прототип десятипиксельного дисплея.....       | 60 |
| Программируем десятипиксельный дисплей.....           | 62 |
| Пользовательские функции.....                         | 62 |
| Разрабатываем графику.....                            | 64 |
| Тестовый скетч .....                                  | 65 |
| Создаем функцию для отображения фигуры X.....         | 66 |
| Создаем функцию для отображения фигуры O .....        | 67 |
| Отображаем фигуры X и O.....                          | 68 |
| Создаем корпус для десятипиксельного дисплея .....    | 70 |
| Делаем картонный корпус.....                          | 70 |
| Подключаем электронику к дисплею.....                 | 72 |

|  |    |
|--|----|
| Создаем пиксельную анимацию .....              | 74 |
| Планируем последовательность анимации .....    | 74 |
| Создаем пользовательские функции .....         | 75 |
| Корректируем функцию <code>loop()</code> ..... | 76 |
| Идем дальше . . . .....                        | 77 |
| Экспериментируем с кодом .....                 | 77 |
| Модифицируем схему .....                       | 77 |

## **Проект 4. Измеритель скорости реакции..... 79**

|   |    |
|---|----|
| Необходимые компоненты, инструменты и материалы .....   | 80 |
| Электронные компоненты .....                            | 80 |
| Прочие инструменты и материалы .....                    | 81 |
| Новый компонент: кнопка .....                           | 82 |
| Принцип работы кнопок .....                             | 82 |
| Использование резисторов с кнопками .....               | 83 |
| Создаем прототип измерителя скорости реакции .....      | 83 |
| Программируем измеритель скорости реакции .....         | 85 |
| Создаем функцию <code>setup()</code> .....              | 85 |
| Создаем функцию <code>loop()</code> .....               | 86 |
| Тестируем скетч измерителя скорости реакции .....       | 89 |
| Следующий раунд .....                                   | 90 |
| Добавляем аркадный элемент .....                        | 90 |
| Полный код скетча для измерителя скорости реакции ..... | 93 |
| Создаем корпус для измерителя скорости реакции .....    | 94 |
| Вырезаем отверстия в корпусе .....                      | 95 |
| Собираем электронную часть .....                        | 95 |
| Декорируем корпус .....                                 | 97 |
| Идем дальше . . . .....                                 | 98 |
| Экспериментируем с кодом .....                          | 98 |
| Модифицируем схему .....                                | 99 |

|   |            |
|---|------------|
| <b>Проект 5. Разноцветный ночник.....</b>                       | <b>101</b> |
| Необходимые компоненты, инструменты и материалы .....           | 102        |
| Электронные компоненты .....                                    | 102        |
| Прочие инструменты и материалы .....                            | 103        |
| Два новых компонента .....                                      | 104        |
| Трехцветный (RGB) светодиод.....                                | 104        |
| Фоторезистор.....   | 105        |
| Создаем прототип ночника .....                                  | 107        |
| Собираем схему делителя напряжения .....                        | 108        |
| Подключаем трехцветный светодиод .....                          | 109        |
| Тестируем ночник с простым смешением цветов .....               | 110        |
| Программируем ночник .....                                      | 111        |
| Подготовка к проверке уровня освещенности .....                 | 112        |
| Управляем ночником в зависимости от уровня освещенности.....    | 112        |
| Предотвращение ложных срабатываний .....                        | 113        |
| Рекалибровка ночника .....                                      | 113        |
| Создаем другие цвета.....                                       | 114        |
| Создание аналоговых сигналов посредством ШИМ .....              | 114        |
| Смешение цветов посредством функции <i>analogWrite()</i> .....  | 115        |
| Определение значений цветов RGB с помощью цветоподборщика ..... | 116        |
| Ночник с задаваемым цветом.....                                 | 117        |
| Создаем абажур для ночника .....                                | 117        |
| Делаем картонный корпус.....                                    | 117        |
| Вставляем в абажур электронные компоненты .....                 | 121        |
| Да будет свет!.....   | 122        |
| Идем дальше... .....  | 122        |
| Экспериментируем с кодом.....                                   | 122        |
| Модифицируем корпус.....  | 123        |

## **Проект 6. Балансирная балка.....125**

|  |     |
|--|-----|
| Необходимые компоненты, инструменты и материалы .....      | 126 |
| Электронные компоненты .....                               | 126 |
| Прочие инструменты и материалы .....                       | 127 |
| Новые компоненты .....                                     | 128 |
| Потенциометр .....   | 128 |
| Серводвигатель.....  | 129 |
| Создаем прототип схемы управления балансирной балкой ..... | 131 |
| Программа для управления балансирной балкой .....          | 133 |
| Проверяем работоспособность машинки.....                   | 134 |
| Финальная версия скетча для игры в балансирную балку.....  | 135 |
| Собираем игру в балансирную балку.....                     | 137 |
| Вырезаем детали.....                                       | 137 |
| Собираем собственно балансирную балку.....                 | 138 |
| Собираем основание и прикручиваем сервомашинку .....       | 140 |
| Финальная сборка .....                                     | 142 |
| Идем дальше... ..  | 146 |
| Экспериментируем со схемой и кодом .....                   | 146 |
| Модифицируем проект .....                                  | 146 |

## **Проект 7. Миниатюрная настольная теплица ..... 149**

|   |     |
|---|-----|
| Необходимые компоненты, инструменты и материалы .....   | 151 |
| Электронные компоненты .....                            | 151 |
| Прочие инструменты и материалы .....                    | 153 |
| Новые компоненты .....                                  | 153 |
| Датчик температуры TMP36 .....                          | 153 |
| Электромотор .....                                      | 153 |
| NPN-транзистор .....                                    | 154 |
| Применяем системный подход .....                        | 154 |
| Собираем систему управления температурой .....          | 155 |
| Измерение температуры с помощью термодатчика TMP36..... | 156 |
| Подключаем датчик температуры .....                     | 156 |
| Программируем снятие показаний датчика температуры..... | 157 |

|   |     |
|---|-----|
| Собираем схему сервомашинки для управления окном.....               | 162 |
| Разрабатываем код для управления сервомашинкой .....                | 163 |
| Собираем схему для управления электродвигателем вентилятора.....    | 165 |
| Разрабатываем код для управления электродвигателем вентилятора..... | 168 |
| Изолируем влияние электродвигателя.....                             | 168 |
| Собираем корпус теплички .....                                      | 169 |
| Крепим сервомашинку для управления окном .....                      | 171 |
| Изготавливаем тягу .....  | 172 |
| Устанавливаем крышу.....  | 172 |
| Собираем контейнер для электродвигателя .....                       | 174 |
| Подключаем электронику .....  | 175 |
| Идем дальше... .....  | 176 |
| Экспериментируем с размерами теплицы.....                           | 176 |
| Модифицируем код.....   | 176 |

## **Проект 8. Робот-рисовальщик ..... 179**

|   |     |
|---|-----|
| Необходимые компоненты, инструменты и материалы .....         | 180 |
| Электронные компоненты .....                                  | 180 |
| Прочие инструменты и материалы .....                          | 181 |
| Два новых компонента .....                                    | 182 |
| Интегральная схема H-мостового драйвера электродвигателя..... | 182 |
| Электрический двигатель с редуктором.....                     | 184 |
| Создаем прототип схемы управления Рисоботом.....              | 185 |
| Разрабатываем код для управления Рисоботом.....               | 186 |
| Создаем пользовательскую функцию.....                         | 188 |
| Расчищаем код.....  | 188 |
| Подключаем второй электродвигатель.....                       | 189 |
| Проверяем работу обоих электродвигателей.....                 | 190 |
| Создаем платформу для Рисобота.....                           | 191 |
| Тестирование и отладка .....                                  | 194 |
| Танец робота — делаем повороты и рисуем узоры .....           | 195 |

|                          |     |
|--------------------------|-----|
| Идем дальше...           | 199 |
| Экспериментируем с кодом | 199 |
| Модифицируем код         | 200 |
| Бонус                    | 200 |

## **Проект 9. Хронометрист автогонок ..... 203**

|   |     |
|---|-----|
| Необходимые компоненты, инструменты и материалы | 204 |
| Электронные компоненты                          | 204 |
| Прочие инструменты и материалы                  | 206 |
| Новый компонент: жидкокристаллический дисплей   | 207 |
| Принцип работы хронометриста автогонок          | 208 |
| Собираем схему с ЖКД                            | 208 |
| Подключаем питание ЖКД                          | 209 |
| Настройка контраста ЖКД                         | 209 |
| Подключаем линии данных и управления            | 210 |
| Проверяем работу ЖКД                            | 211 |
| Добавляем остальные компоненты                  | 213 |
| Программа для хронометриста автогонок           | 215 |
| Быстрая проверка                                | 218 |
| Собираем гоночный комплекс                      | 218 |
| Собираем стартовую башню                        | 219 |
| Собираем и вставляем стартовые ворота           | 221 |
| Изготавливаем гоночную трассу                   | 222 |
| Монтируем фоторезистор                          | 223 |
| Тестирование и отладка                          | 224 |
| Идем дальше...                                  | 225 |
| Экспериментируем с проектом                     | 225 |
| Подключение ЖКД через модуль IIC/I2C            | 227 |
| Модифицируем предыдущие проекты                 | 229 |

## **Проект 10. Электронное мини-пианино ..... 231**

|   |     |
|---|-----|
| Необходимые компоненты, инструменты и материалы | 232 |
| Электронные компоненты                          | 232 |
| Прочие инструменты и материалы                  | 233 |

|   |     |
|---|-----|
| Новые компоненты .....                                    | 234 |
| Мембранный потенциометр.....                              | 234 |
| Пьезоэлектрический зуммер.....                            | 234 |
| Собираем схему .....                                      | 235 |
| Программируем электронное пианино .....                   | 237 |
| Тестируем работу зуммера .....                            | 237 |
| Создаем конкретные ноты .....                             | 239 |
| Создаем звуки посредством мембранного потенциометра ..... | 239 |
| Играем по нотам.....                                      | 241 |
| Собираем мини-пианино .....                               | 243 |
| Идем дальше... .....                                      | 245 |
| Экспериментируем с кодом.....                             | 245 |
| Модифицируем схему и код .....                            | 245 |
| Бонусный проект: цифровая труба .....                     | 246 |

## **ПРИЛОЖЕНИЕ. Дополнительные практические сведения**

### **по электронике..... 249**

|   |     |
|---|-----|
| Электрические измерения с помощью мультиметра ..... | 250 |
| Функциональные части мультиметра.....               | 250 |
| Определение неразрывности электроцепи .....         | 250 |
| Измерение сопротивления .....                       | 251 |
| Измерение напряжения .....                          | 252 |
| Измерение тока .....                                | 252 |
| Работа с паяльником .....                           | 253 |
| Разогревание паяльника .....                        | 254 |
| Советы по улучшению навыков пайки .....             | 254 |
| Очистка паяльника .....                             | 256 |
| Советы по работе с паяльником.....                  | 256 |
| Дополнительные инструменты для паяльных работ ..... | 256 |
| «Третья рука» .....                                 | 256 |
| Флюс-аппликатор .....                               | 257 |
| Косичка для удаления припоя .....                   | 257 |
| Вакуумный отсос.....                                | 258 |
| Полосатые резисторы .....                           | 258 |

# ВВЕДЕНИЕ

Приветствуем вас, уважаемые читатели. Эта книга научит вас работать с электронными компонентами, программировать электронные устройства и создавать на этой основе всевозможные интересные и полезные проекты. Любой человек может быть изобретателем, и наша книга пошагово проведет вас по последовательности проектов, в которых обычные детали сочетаются с мощностью модуля Arduino, помогая вам вдохновиться на собственные изобретения.

## О чем эта книга?

---

Эта книга основана на платформе Arduino ([www.arduino.cc](http://www.arduino.cc)), включающей микроконтроллерную плату, которую можно запрограммировать для управления источниками света, измерения температуры, реагирования на освещение, связи со спутниками глобальной системы позиционирования<sup>1</sup> и, вообще, для того чтобы с ее помощью создать много разных полезных и занимательных устройств. Язык программирования и среда разработки для платы, которые используются в книге, также являются компонентами платформы Arduino.

Эта платформа представляет собой мощный инструмент, с помощью которого любители электронного творчества могут оснащать свои проекты средствами управления. Поиск в Интернете по ключевым словам проекты Arduino возвращает миллионы результатов. Веб-сайты, такие как,

<sup>1</sup> GPS, от англ. Global Positioning System.

например, Instructables, hackster.io или YouTube, содержат тысячи проектов и идей их создания. Все это демонстрирует, насколько много есть мастеров на все руки, использующих Arduino.

Совместно с компанией SparkFun Electronics мы всячески стараемся побудить людей экспериментировать, играть и возиться с обычными бытовыми устройствами, добавляя в них новые электронные компоненты или модифицируя уже имеющиеся. Такая деятельность называется *хакерством*. Чтобы вы могли успешно ею заниматься, наша книга предоставит вам основные знания по электронике и программированию. Мы также надеемся, что она вдохновит вас на создание чего-то нового и уникального из обычных материалов, которые можно найти у каждого в доме.

## Почему Arduino?

---

Вы можете задаться вопросом, почему из существующих десятков разных микроконтроллеров и инструментальных платформ мы создаем еще один набор проектов для Arduino?

Ответ на этот вопрос заключается в том, что целью создания платформы Arduino было использование ее не любителями самодельного электронного творчества или инженерами промышленных предприятий, а студентами-проектировщиками из итальянского города Ивреа, — в качестве обучающей платформы, чтобы они могли создавать функционирующие проекты без необходимости изучать в течение нескольких лет теоретическую и практическую электротехнику, электронику и математику. Эта платформа была создана таким образом, чтобы свести к минимуму время от «ничего» к «круто!» — то есть от замысла к работающему проекту, — для людей без технического образования и опыта.

Платформа Arduino проявила себя настолько хорошо, что сообщество любителей-электронщиков также решило взять ее на вооружение. Этому способствовало несколько факторов: низкая цена, качественная документация, открытое аппаратное и программное обеспечение и т. п. Но мы считаем, что основной причиной такой популярности Arduino стала легкость обучения работе на этой платформе. Платформа Arduino — это открытый для любого желающего портал в мир созидания и изобретательства. Проекты из этой книги предназначены для энтузиастов, влекомых желанием познания и мотивируемых исходной сущностью платформы Arduino — быстрым и легким воплощением идеи в работающий проект людьми, не обладающими глубокими познаниями в электронике.

## Чем эта книга отличается от других?

Многие книги по программированию похожи на справочные пособия — они сразу же начинаются с написания кода или рассмотрения понятий из области электроники без предоставления каких бы то ни было начальных сведений. В результате большую часть времени они пылятся на полке, пока вам не понадобится информация об определенной команде или забытом понятии.

Эта книга иная. Ее целью является обучение новым знаниям посредством создания занимательных, интересных и практических проектов. Сложность и трудность проектов повышается от предыдущего проекта к последующему. Они дадут вам ответы на старые как мир вопросы: зачем я изучаю это? Почему это важно? Почему это должно меня беспокоить?

Мы предполагаем, что вы читаете эту книгу, потому что горите желанием познания нового или же ищете материалы, которыми хотите поделиться с другими. Будь вы интересующийся новичок-любитель, учитель, библиотекарь или родитель, эта книга станет для любого, кто хочет познавать, практическим руководством, а не справочным пособием, которое будет в основном стоять на полке.

Чтобы начать работать с Arduino, вам не нужен никакой опыт программирования или работы с электроникой. Мы заранее предполагаем, что вначале вы не знаете ничего в этих областях, и опасаетесь броситься в них сломя голову. Не беспокойтесь! Эта книга предоставляет достаточно вводного

материала, чтобы реализовывать простые проекты, и постепенно переходит к материалу для реализации более сложных и требовательных проектов.

Мы также знаем, что есть много людей, имеющих некий опыт работы с электроникой, которые хотят испытать что-то неизведанное, — возможно, новый подход к старому проекту. Например, найти не использованный ими ранее способ мигания светодиодом. Многие из наших проектов можно принять в качестве начальных для дальнейшего экспериментирования и модифицирования или как прототипы, на основе которых затем можно собрать проекты из лучших компонентов или с более привлекательным оформлением. В общем, эта книга направлена на активных экспериментаторов, которые без колебаний принимают за решение задачи, уделяя этому максимальное внимание.

Мы рекомендуем реализовывать проекты при чтении книги, последовательно обучаясь путем приобретения практического опыта. Наши проекты хорошо продуманы с тем, чтобы дать вам необходимые знания как по использованию инструментов, так и по программированию и сборке схем, а также по созданию вспомогательных конструкций из картона, проволоки и прочих обыденных материалов, которые без труда можно найти в любом доме. Удовольствие от обучения заключается в том, чтобы весь процесс был игрой, во что эта книга и старается его превратить.

## Необходимые компоненты и материалы

В проектах этой книги в основном используются электронные компоненты из нашего базового продукта — «Набора изобретателя SparkFun» (SparkFun Inventor's Kit, KIT-13969), но все эти компоненты можно свободно приобрести и по отдельности во многих интернет-

магазинах. Кроме того, в проектах задействованы также несколько компонентов, не входящих в этот набор, которые также можно приобрести в виде отдельного набора ([www.sparkfun.com/NoStarchArduino](http://www.sparkfun.com/NoStarchArduino)).

Списки необходимых компонентов приведены в табл. В.1 и В.2. В начале каждого проекта также приводится список необходимых для него компонентов и материалов.

Кроме того, для создания корпусов проектируемых устройств используются многие обычные

материалы: упаковочный картон, открыточный картон, соломинки для питья, бумажные тарелки и т. п. По мере реализации электронных проектов с использованием таких материалов, вы станете смотреть на все эти бытовые предметы и материалы по-новому.

**Таблица В.1.** Компоненты «Набора изобретателя SparkFun», используемые в проектах книги

| К-во | № компонента | Описание  |
|------|--------------|---|
| 1    | DEV-13975    | Плата SparkFun RedBoard (или другая плата, совместимая с Arduino) |
| 1    | CAB-11301    | Кабель Mini-B USB   |
| 1    | PRT-12002    | Беспаячная макетная плата   |
| 30   | PRT-11026    | Проволочные перемычки   |
| 20   | COM-12062    | Разные светодиоды   |
| 1    | COM-09264    | Светодиод RGB (общий катод)                                       |
| 20   | COM-11508    | Резисторы, 10 кОм   |
| 20   | COM-11507    | Резисторы, 330 Ом   |
| 2    | COM-10302    | Кнопочные переключатели   |
| 1    | COM-08588    | Диод  |
| 1    | COM-09806    | Потенциометр, 10 кОм  |
| 1    | COM-13689    | Транзистор NPN 2N222  |
| 1    | SEN-09088    | Фоторезистор  |
| 1    | SEN-10988    | Датчик температуры TMP36  |
| 1    | SEN-08680    | Потенциометр SoftPot, 50 мм                                       |
| 1    | COM-07950    | Пьезоэлектрический зуммер   |
| 1    | LCD-00709    | ЖК-дисплей, 16×2 символа  |
| 1    | ROB-09065    | Миниатюрный серводвигатель  |
| 1    | ROB-11696    | Электродвигатель  |

**Таблица В.2.** Дополнительные компоненты, используемые в проектах книги (не входящие в «Набор изобретателя SparkFun»)

| К-во | № компонента | Описание   |
|------|--------------|--|
| 1    | PRT-12043    | Макетная мини-плата  |
| 30   | PRT-13870    | Короткие (10 см) проволочные перемычки со штекерами на обоих концах  |
| 10   | PRT-09140    | Проволочные перемычки со штекером на одном конце и гнездом на другом |
| 20   | COM-12062    | Разные светодиоды (дополнительные)                                   |
| 1    | SEN-09088    | Фоторезистор (дополнительный)  |
| 1    | PRT-09835    | Держатель для 4-х элементов AA                                       |
| 4    | PRT-09100    | Элементы AA  |
| 1    | ROB-13845    | Драйвер электродвигателя — Н-мост TB6612FNG                          |
| 2    | ROB-13302    | Электродвигатели для любительских проектов с редуктором              |
| 2    | ROB-13259    | Резиновое колесико для использования с редукторным электродвигателем |
| 1    | COM-00102    | Миниатюрный ползунковый переключатель                                |

### Набор совместимых компонентов

Для выполнения проектов, описанных в книге, в оригинальном издании рекомендовано использовать набор компонентов SparkFun Inventor's Kit (<https://www.sparkfun.com/products/14265>). Если у вас его нет, предлагаем обратить внимание на наборы издательства «БХВ-Петербург» (<http://www.bhv.ru/books/kits>), специально подготовленные для книг по электронике, выпускаемых издательством. С их помощью вы можете выполнить эксперименты и проекты в том числе и из этой книги.

---

## Необходимые инструменты

---

Все, что из инструментов вам потребуется для реализации проектов книги — это ножницы, макетный нож и клеевой пистолет. Но это не означает, что можно использовать только эти инструменты. Если у вас есть доступ к лазерному резаку, используйте его. Само собой, если у вас чешутся руки распечатать проект на 3D-принтере, полный вперед! Проекты в книге предназначены для реализации с использованием упаковочного и открыточного картона, но вместо этих материалов можно использовать любой подходящий материал и способ для его обработки.

Вы не должны ничего для реализации проектов покупать, если вы не желаете делать это. Собственно говоря, в качестве материала для нескольких проектов из этой книги можно использовать и саму книгу. Будет просто замечательно, если вы так и сделаете. Как бывшим учителям, нам знакома ситуация, когда приходится работать с

очень ограниченным бюджетом, и мы всегда сосредотачиваемся на использовании наиболее экономных материалов — таких как картон, бумага, дерево и отходы пластмассы и металлов.

Большинство проектов этой книги предназначены для реализации в виде прототипов, которые можно легко разобрать и использовать их компоненты в других проектах. Но если вам сильно понравится какой-либо проект, и вы захотите сделать его постоянно действующим, его соединения можно выполнить пайкой (соответствующие инструкции приведены в *разд. «Работа с паяльником» приложения*). Инструменты и расходные материалы для пайки при создании прототипов электронных устройств сравнительно недороги — их можно приобрести как в магазине SparkFun ([www.sparkfun.com](http://www.sparkfun.com)), так и в любом ближайшем магазине хозяйственных товаров или инструментов.

---

## Компьютер

---

Наконец, для программирования Arduino вам потребуется компьютер и набор специальных программных инструментов. Работа с программным обеспечением для Arduino под силу любому компьютеру средних возможностей. Для компьютеров на ОС Windows подойдут операционные системы Windows XP, Vista, Windows 7, 8/8.1, 10 или более новая. Для компьютеров Mac самая последняя версия программного обеспечения Arduino требует операционную систему OS X 10.7 Lion или более новую. Если у вас компьютер с более или менее стандартной версией Linux, есть хорошие шансы, что для них также найдется подходящая версия программного обеспечения Arduino.

На момент подготовки этой книги поддержка устройств на iOS и Android обеспечивается только посредством бета-версии программного обеспечения Arduino, которое еще находится в процессе разработки и тестирования. Вы можете попробовать использовать это программное обеспечение, но оно может оказаться неработоспособным, а если и будет работать, то ненадежно.

Процесс установки и конфигурирования программного обеспечения для компьютеров Windows, Mac и Linux рассматривается пошагово в первом проекте.

## Состав книги

Книга содержит 10 практических проектов, а также краткое изложение основ электроники и приложение, в котором рассматриваются основы работы с паяльником, а также даются другие полезные советы. Проекты начинаются с простого мигающего светодиода и с каждым следующим проектом постепенно обрастают разными электронными компонентами, концепциями программирования и слоями сложности конструирования. Каждый проект содержит отдельные разделы по монтажу электрических соединений, программированию и сборке, что позволяет досконально разобраться с каждым из этих аспектов проекта. Проекты завершаются разделом *Идем дальше*, в котором излагаются идеи по экспериментированию с проектом и его модификации. Не забывайте, что эти проекты следует использовать в качестве отправной точки для своих разработок, а не как конечную цель.

- **Основы электроники.** Прежде чем приступить к работе с проектами, вам рекомендуется познакомиться с основами электричества и электроники и понятиями, которые используются в этой книге.
- **Проект 1. Начало работы с Arduino.** В проекте описывается установка программного обеспечения и даются основы приемов сборки и программирования схем на основе пошаговых инструкций при реализации проекта мигающего светодиода.
- **Проект 2. Домашний светофор.** Здесь исследуется работа с макетной платой и рассматривается управление несколькими компонентами одновременно на примере модели светофора из трех светодиодов.
- **Проект 3. Десятипиксельный анимационный дисплей.** Здесь производится расширение проекта светофора из *проекта 2* до девяти светодиодов в матрице размером 3×3, а также изучается создание пользовательских функций Arduino.
- **Проект 4. Измеритель скорости реакции.** В проекте описывается использование кнопки и светодиода при создании игры для измерения скорости реакции.
- **Проект 5. Разноцветный ночник.** Здесь делитель напряжения и датчик освещенности используются для определения уровня освещенности и активирования в зависимости от этого уровня многоцветного светодиода.
- **Проект 6. Балансирная балка.** В проекте используется серводвигатель под управлением внешнего устройства для создания настольной игры с балансирной балкой.
- **Проект 7. Миниатюрная настольная теплица.** В проекте создается модель теплицы, которая определяет температуру, включает вентилятор и открывает вентиляционное отверстие, если температура в ней станет слишком высокой. Отрабатывается здесь также управление электродвигателем с помощью транзистора.
- **Проект 8. Робот-рисовальщик.** Проект представляет собой введение в робототехнику с использованием Н-моста для регулирования частоты оборотов электродвигателя. Здесь осуществляется сборка простого робота, которого можно запрограммировать на рисование различных фигур.
- **Проект 9. Хронометрист автогонок.** В проекте рассматривается создание гоночной трассы для игрушечных машин, которая записывает их скорость. В конструкции используются сервопривод, светочувствительный датчик и жидкокристаллический дисплей. Проект настолько занимательный, что его можно подарить на Рождество!
- **Проект 10. Миниатюрное пианино.** Проект помогает создавать с помощью Arduino музыку, используя мембранный потенциометр в качестве клавиатуры. Здесь также исследуется пьезоэлектрический зуммер и рассматривается использование функции `tone()`. Хорошая возможность открыть в себе пианиста!
- **Приложение. Дополнительные практические сведения по электронике.** Здесь приводятся полезные советы по использованию мультиметра, паяльника и определению номинального значения резисторов по цветным полосам.

## Интернет-ресурсы

Все ресурсы, необходимые для проектов этой книги, доступны для загрузки, использования и модифицирования. Они содержат коды всех примеров, которые приводятся и обсуждаются в книге, шаблоны для конструкций проектов, а также коды для дальнейшего экспериментирования с проектами и их модифицирования.

Ресурсы организованы в виде ZIP-файла, который можно загрузить по адресу: [www.nostarch.com/arduinoinventor](http://www.nostarch.com/arduinoinventor).

Если вы застопоритесь на каком-либо аспекте того или иного проекта или у вас возникнут проблемы с его работой, можно всегда обратиться к этим файлам за справкой для решения проблемы.

### Электронный архив

Все необходимые для работы с проектами книги ресурсы вы также найдете в сопровождающем книгу электронном архиве, который можно загрузить с FTP-сервера издательства «БХВ-Петербург» по ссылке: <ftp://ftp.bhv.ru/9785977539722.zip> или со страницы книги на сайте [www.bhv.ru](http://www.bhv.ru).

## Распространяем информацию: делимся результатами своей работы

Компания SparkFun, будучи поставщиком аппаратных и электронных компонентов, прилагает также большие усилия, чтобы быть открытым источником информации, — это один из основных принципов, на котором она основана. Разрабатывая проекты, мы любим делиться своими идеями и файлами проектов с нашим сообществом, чтобы все желающие могли воспользоваться нашей базой знаний в своем следующем проекте. Мы бы хотели, чтобы вы также исповедовали такой подход и делились с другими информацией о своих проектах. Покажите их своим друзьям или опубликуйте в Интернете. В последнем случае это можно сделать во многих местах.

Например, поделиться своей работой можно в социальных сетях Twitter, Instagram или Facebook. Пометьте опубликованные проекты тегами

@sparkfunedu и @nostarch. Мы также предлагаем для общего доступа через Интернет галерею проектов InventorSpace, доступную по адресу: <https://invent.sparkfun.com>. Если у вас есть идея или проект, которыми вы хотите поделиться, вы можете опубликовать его в этой галерее. Мы надеемся, что эта книга вдохновит вас начать делать что-то поразительное.

Наконец, вы можете также отправить нам свои проекты, их фотографии или общие замечания и вопросы электронной почтой по адресу: [ArduinoInventorsGuide@sparkfun.com](mailto:ArduinoInventorsGuide@sparkfun.com). Время от времени мы выбираем некоторые замечательные проекты и фотографии для главной публикации в нашем блоге. Кто знает, возможно мы попросим у вас разрешения использовать один из ваших проектов в нашей следующей книге.

# ОСНОВЫ ЭЛЕКТРОНИКИ

Эта глава дает общие сведения об электричестве, электронике и схемотехнике для тех, у кого нет никакого опыта работы в этих сферах знания или этот опыт очень небольшой. Если вы почувствуете, что некоторые представленные в этом проекте темы вам более или менее знакомы, можете пропускать их и переходить к тем темам, о которых вы хотите знать более подробно, или вообще перейти сразу к проекту 1.

Тем не менее, даже если вы и не абсолютный новичок в электронике, мы рекомендуем вам прочитать эту главу, просто чтобы освежить свои знания. И пусть она и не представляет собой полное руководство по электронике (этой области посвящены целые тома монографий и учебные курсы), можете считать ее удобным справочным

пособием, предназначенным дать вам представление об основных ее понятиях и терминологии. Для тех же, кто хочет получить более глубокие знания по электричеству, электронике и схемотехнике, в конце этой главы приведен список рекомендуемой литературы.

## Электричество: ТОК, ПРОВОДИМОСТЬ И ОСНОВНАЯ ТЕРМИНОЛОГИЯ

---

Электричество — это очень странный зверь. Во многих отношениях оно предсказуемо, но временами может быть весьма коварным. Если просто посмотреть определение электричества в учебнике, то, скорее всего, оно не много скажет вам о том, что такое электричество, как оно работает и, самое главное, как его можно использовать. В данной главе предоставляются основы всего этого.

### Что такое электричество?

Чтобы ответить на этот вопрос, прежде всего нужно разобраться в строении атома. Атомы являются основными составляющими единицами всего, что находится вокруг нас, и нас самих. Состоят они из протонов, нейтронов и электронов. Протоны и нейтроны образуют ядро атома, а электроны вращаются вокруг ядра по более или менее круговым орбитам. Протоны и, соответственно, ядро атома обладают положительным зарядом, а электроны — отрицательным. Нейтроны заряда не несут. Количество электронов типичного атома равно количеству его протонов, поэтому атом имеет нейтральный заряд.

Под воздействием тех или иных сил несущие заряды электроны могут срываться со своих орбит и перемещаться внутри вещества. Такое перемещение зарядов и представляет собой вид энергии, называемый электричеством. Молния, которую наверняка когда-либо видел каждый из нас, является наглядной демонстрацией перемещения зарядов между облаками или облаками и землей.

Перемещение зарядов в веществе называется *током*. Величина тока измеряется в единицах, называемых *амперами* (А). Для удобства на практике могут использоваться и более мелкие единицы — например, *миллиамперы* (мА).

#### Примечание

Условно электрическим током называется перемещение положительных зарядов. И хотя в действительности ток создается перемещением электронов, которые несут отрицательный заряд, принято считать, что ток протекает от положительного источника к отрицательному.

За исключением нескольких явлений, таких как молния, сварочная дуга или разряд статического электричества, электрический ток обычно нельзя наблюдать непосредственно. Даже яркий свет молнии является лишь следствием изменения состояния молекул воздуха, вызываемого прохождением электричества через них.

Заряды перемещаются в веществе под воздействием внешней электрической силы и при наличии пути, по которому они могут перемещаться. Сила электрического воздействия создается *разностью потенциалов*, которая обычно называется *напряжением*. Напряжение и является силой, которая, в конечном счете, заставляет заряды

двигаться. Величина напряжения измеряется в *вольтах* (V). Для общей информации: типичные электрические батареи имеют напряжение в диапазоне от около 1,5 до 12 В. Батарея с напряжением в 12 В заставляет перемещаться большее количество зарядов, чем батарея с напряжением в 1,5 В.

## Типы электрического тока

Существуют два основных типа электрического тока: *постоянный* и *переменный*. Самый распространенный пример переменного тока — электричество в линиях электропередачи, которое заходит к нам в дом, в частности, через розетки. Переменный ток отлично подходит для выработки электричества (например, электростанциями), передачи его на дальние расстояния (например, от электростанции к нам в дом) и для приведения в действие мощных электрических устройств (например, электродвигателей и обогревателей). Но для работы большинства электронных приборов (телевизоры, компьютеры и т. п.) переменный ток не нужен — им требуется постоянный ток, который они получают с помощью специального устройства (обычно встроенного в вилку, вставляющуюся в розетку), преобразовывающего переменный ток в постоянный. Более глубокое рассмотрение переменного и постоянного тока выходит за рамки этой книги. В завершении скажем только, что для питания проектов в этой книге используется электричество постоянного тока.

## Что такое цепь?

Даже когда на заряды воздействует электрическая сила, понуждающая их перемещаться от более высокого потенциала к более низкому, между этими потенциалами необходим путь какого-либо рода. Путь, по которому заряды перемещаются от более высокого (положительного: «+») потенциала батареи к более низкому (отрицательному: «-») ее потенциалу, называется *электрической цепью*. Таким образом, электрическая цепь представляет собой замкнутый путь, проходящий от положительного вывода источника питания через потребителя тока (например: светодиод, резистор,

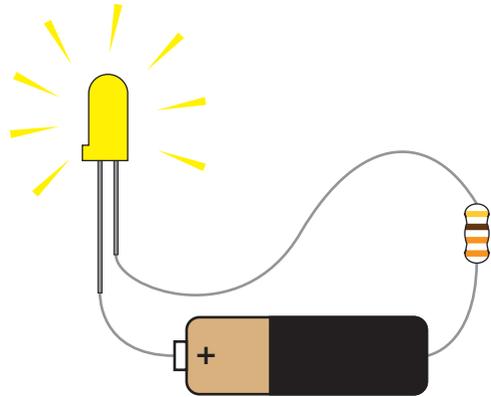


Рис. 1. Простая цепь постоянного тока

лампочку или электродвигатель) к отрицательному его выводу. На рис. 1 показана простая электрическая цепь, состоящая из батарейки, светодиода и резистора.

Но чтобы заряды (электроны) могли перемещаться по цепи, она должна быть из материала, проводящего электричество, или, иными словами, обладать *электрической проводимостью*. Проводимость того или иного материала зависит от вещества, из которого он состоит. Притом, что некоторые материалы обычно считаются абсолютными проводниками или непроводниками (изоляторами), проводимость большинства материалов варьируется в широком диапазоне. Иными словами, заряды перемещаются через некоторые материалы более свободно, чем через другие.

Это можно сравнить с ездой на машине по дорогам с разным покрытием. На гладкой асфальтированной дороге можно ехать намного быстрее, чем по проселочной или вообще по пересеченной местности. Таким образом, как дороги могут иметь разное покрытие, позволяющее развивать разную скорость перемещения по ним (разную автомобильную проводимость, так сказать), так же и разные материалы имеют различную электрическую проводимость, позволяющую более или менее свободное перемещение электрических зарядов через них. Величина, обратная проводимости, называется *сопротивлением* и выражает степень затруднения материалом перемещения по нему электрических зарядов.

## Закон Ома

Как вы уже, наверное, догадались, ток, напряжение и сопротивление взаимосвязаны. Эта взаимосвязь называется законом Ома и выражается следующей математической формулой:

$$U = I \times R$$

В этом уравнении буква  $U$  обозначает напряжение,  $I$  — ток, а  $R$  — сопротивление. (Не стоит пугаться, что мы сейчас начнем углубляться в математические дебри — это всего лишь одно из трех уравнений, которые вы увидите в этой книге.)

## Модель электрического тока: вода в трубе

Чтобы лучше понять прохождение тока по электрической цепи, представим себе трубу, по которой протекает вода. Открывая кран, мы позволяем воде из резервуара протекать по трубе и вытекать из другого ее конца (рис. 2).



Рис. 2. Модель электрического тока: вода в трубе

## Принципиальные схемы

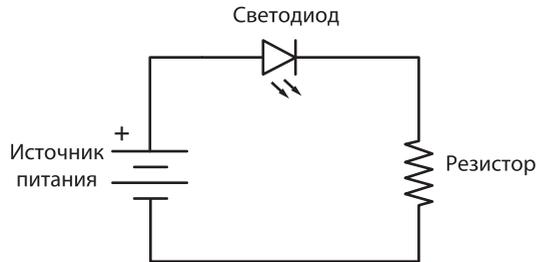
Представление электрической цепи рисунками ее компонентов выглядит красиво, но это не очень эффективно, особенно когда цепь состоит из большого количества элементов. Поэтому на практике цепь представляют упрощенными символами ее компонентов (рис. 3), а само такое представление называют *принципиальной схемой*.

Молекулы воды, перемещающиеся по трубе, представляют в нашей модели поток зарядов (ток). Закрывая или открывая кран, мы можем менять давление воды в трубе, — повышение давления воды ускоряет ее течение по трубе. Давление воды в трубе подобно напряжению в электрической цепи — повышение напряжения увеличивает ток. Труба представляет собой последнюю часть аналогии — пережав ее (уменьшив ее диаметр), мы создадим сопротивление протеканию воды. Такое повышение сопротивления замедлит протекание воды в трубе — возвращаясь к нашей модели, понизит величину тока в электрической цепи.

Эта модель хорошо описывает электрический ток, но мы не хотим собрать всю эту систему резервуаров, труб и кранов, чтобы вода только проливалась на землю (разве что собираемся поливать газон). Нам нужно, чтобы она делала какую-то полезную работу. В случае электрических цепей мы используем устройства, которые преобразуют электричество в полезную работу, — например, в свечение лампочки, вращение электродвигателя или звучание зуммера. Устройство, которое преобразовывает электрическую энергию в другой вид энергии, называется *нагрузкой*. То, что электрическую энергию можно преобразовывать в световую энергию с помощью лампочки, открыл Томас Эдисон. В этой книге мы будем заниматься такими и многими другими преобразованиями.

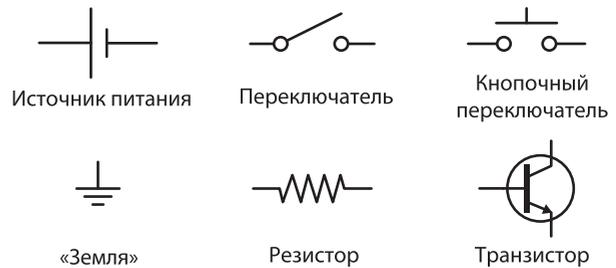
Принципиальная схема показывает, какие компоненты содержит электрическая цепь и каким образом они соединяются между собой<sup>1</sup>. Приведенная на рис. 3 принципиальная схема в действитель-

<sup>1</sup> Принципиальные схемы в этой книге отображаются согласно американскому стандарту IEEE (Institute of Electrical and Electronics Engineers, Институт инженеров электротехники и радиоэлектроники).



**Рис. 3.** Принципиальная схема электрической цепи, состоящей из батарейки, светодиода и резистора

ности представляет ту же самую электрическую цепь, что и на рис. 1. Прямые линии здесь — это провода, а компоненты обозначены соответствующими символами. На рис. 4 показано несколько распространенных схемных символов, которые используются в этой книге.



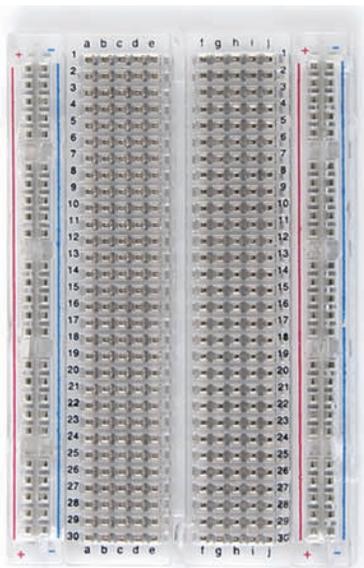
**Рис. 4.** Некоторые схемные символы стандарта IEEE

Формат схемных символов стандарта IEEE предназначен для быстрого представления компонентов схем с использованием очень простых линий и символов. Он признается и применяется для представления и распространения принципиальных схем во всем мире.

## Создание прототипов схем

По мере изучения проектов этой книги, вы будете создавать и проверять множество вариантов их разработки. При сборке схемы может понадобиться перегруппировать компоненты, убрать

некоторые из них или добавить новые. Этот процесс называется *разработкой прототипа*. Для разработки прототипов электронных устройств обычно используется *беспаечная макетная плата* (рис. 5).



**Рис. 5.** Прозрачная беспаячная макетная плата с горизонтальными контактными рядами и вертикальными шинами питания

Беспаечная макетная плата представляет собой пластмассовый прямоугольник с множеством отверстий. Эти отверстия расположены решеткой с расстоянием между ними в 0,1 дюйма (2,54 мм). Диаметр отверстий таков, чтобы выводы большинства электронных компонентов плотно входили в них.

### Ориентация макетной платы

Расположите макетную плату вертикально (книжная ориентация), чтобы буквы на ее концах были размещены правильным образом. При такой ориентации макетной платы горизонтальные группы из пяти гнезд называются *рядами*, а вертикальные секции по сторонам платы — *столбцами*.

Под отверстиями расположены небольшие зажимы, сделанные из мягкого металла (рис. 6). Эти зажимы электрически соединяют компоненты, вставленные в отверстия в том же самом ряду. Таким образом, отпадает необходимость соединять компоненты, скручивая вместе их выводы. Обратите внимание на то, что зажимы перекрывают только пять соседних отверстий в ряду. Макетная плата разделяется на две половины вертикальной выемкой, и зажимы рядов правой стороны не соединены с зажимами рядов левой.

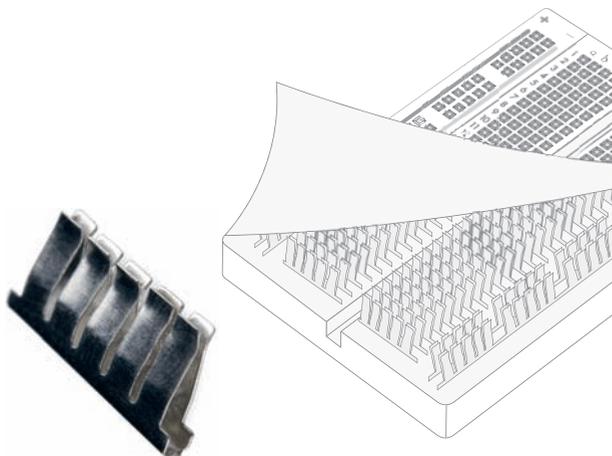


Рис. 6. Внутреннее устройство беспаячной макетной платы (справа) и крупный план металлического зажима (слева)

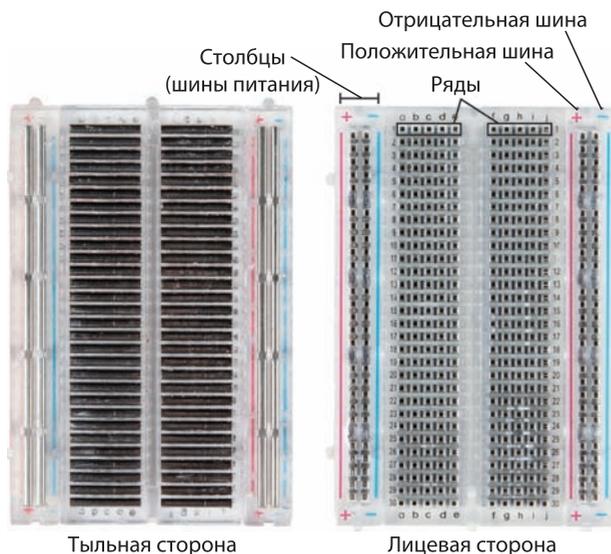


Рис. 7. На тыльной стороне беспаячной макетной платы видны горизонтальные контактные ряды и вертикальные шины питания

Форм и размеров беспаячных макетных плат доступно несколько, но большинство их будут иметь вертикальные столбцы гнезд на внешних сторонах. Эти столбцы гнезд называются *шинами питания*, и каждые пять гнезд в них соединены зажимом, так же, как и гнезда рядов (рис. 7). Столбцы шин питания часто обозначаются символами «+» и «-», указывающими полярность подключаемого питания, а также маркируются красным и синим цветом соответственно.

Проекты этой книги собираются на беспаячных макетных платах, чтобы в случае ошибки можно было бы быстро исправить ее, а также чтобы можно было усовершенствовать готовый проект, добавив к нему дополнительные необходимые компоненты (на рис. 8 показан вариант использования макетной платы для создания прототипа схемы, содержащей восемь светодиодов).

Для сборки крупных и сложных проектов рекомендуется иметь в наличии несколько беспаячных макетных плат, чтобы схему можно было собрать по частям, соединив их затем вместе. Это позволяет собрать и протестировать каждую часть проекта по отдельности, что намного легче, чем тестирование всего проекта.

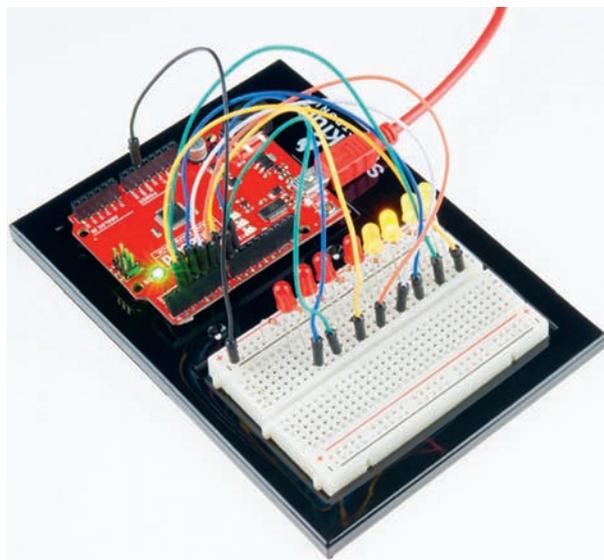


Рис. 8. Сборка схемы на макетной плате

## ДИСКРЕТНЫЕ КОМПОНЕНТЫ И АДАПТЕРНЫЕ ПЛАТЫ

Существуют буквально сотни, если не тысячи разнообразных электронных компонентов. Под *компонентами* в нашем случае мы подразумеваем дискретные электронные детали, т. е. самые элементарные составляющие электронных схем. В качестве примеров дискретных компонентов можно привести резистор, конденсатор и светодиод (рис. 9).

С другой стороны, *адаптерная плата* представляет собой электронную схему из нескольких компонентов, собранную на отдельной плате, — это облегчает установку ее на макетной плате проекта. Таким образом, адаптерные платы помогают значительно ускорить процесс создания прототипов.

На рис. 10 приведен пример сложного дискретного компонента — интегральной схемы датчика акселерометра (P/N ADXL345 компании Analog Devices) и его адаптерной платы, предлагаемой компанией SparkFun. Размеры самой интегральной схемы всего лишь 5×3 мм! Она оснащена миниатюрными металлическими выводами, выполняющими ту же функцию, что и длинные металлические выводы более традиционных дискретных компонентов, — таких как, например, резисторы. Но выводы этой интегральной схемы настолько малого размера, что подсоединить монтажные провода к ним практически невозможно. Поэтому она поставляется в составе адаптерной платы, где эти выводы соединены дорожками с пролуженными насквозь отверстиями на краю платы. Расстояние между этими отверстиями составляет точно 0,1 дюйма (2,54 мм), в результате чего они точно совмещаются с гнездами на безопасной макетной плате. Поскольку отверстия пролужены, в них можно впасть монтажные провода. Кстати, вместо проводов можно впасть в эти отверстия контактные штырьки (рис. 11), и тогда эту плату будет удобно вставлять в безопасную макетную плату. (Не озабочивайтесь, если вы никогда раньше не работали с паяльником. В *разд. «Работа с паяльником» приложения* вы найдете всю информацию, необходимую для выполнения монтажа пайкой.)

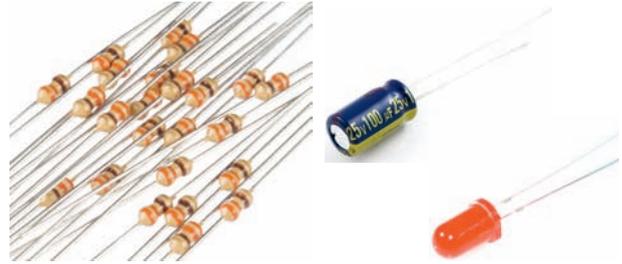


Рис. 9. Примеры дискретных компонентов: резисторы (слева), конденсатор (посередине) и светодиод (справа)

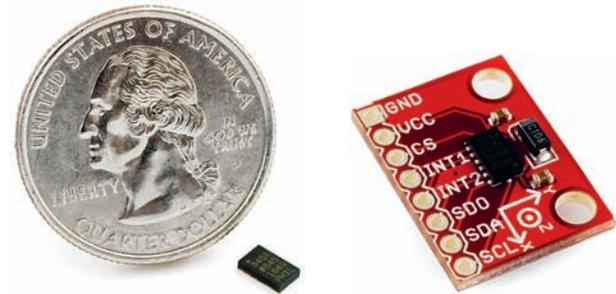


Рис. 10. Дискретный миниатюрный акселерометр (слева) и адаптерная плата, на которой он установлен (справа). Обратите внимание на пролуженные насквозь отверстия с левой стороны адаптерной платы

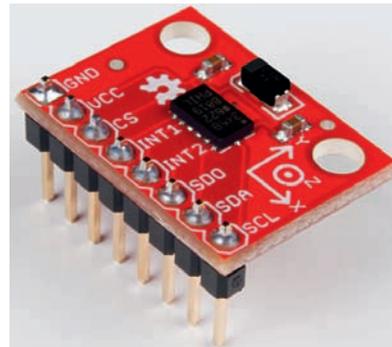


Рис. 11. Адаптерная плата ADXL345 с контактными штырьками

Обратите внимание, что каждое контактное отверстие адаптерной платы обозначено названием подключенного к нему вывода интегральной схемы акселерометра. Таким образом, адаптерная плата позволяет сразу же использовать ее в безопасной макетной плате, без необходимости тратить время на решение проблемы, каким образом подключить дискретный компонент.

## Аналоговая и цифровая электроника

Рассмотрев понятия тока, напряжения, сопротивления, электронных компонентов и схем, можно перейти к обсуждению двух типов электроники: аналоговой и цифровой. Эти два типа электроники взаимосвязаны, и для понимания работы электронных схем в целом необходимо разобраться с основными принципами работы каждого из этих двух типов.

*Аналоговая* электроника работает со значениями токов и напряжений, которые варьируются плавно в пределах определенного диапазона. Например, контроллер плавного изменения яркости освещения является аналоговым устройством. Аналоговое значение может быть нулевым, максимальным или каким угодно между этим двумя крайностями. С другой стороны, *цифровые* величины имеют только два значения: нулевое и максимальное или включенное и выключенное.

Устройства цифровой электроники, такие как микроконтроллер и микропроцессор, включают и выключают другие устройства в зависимости от запрограммированных в них условий. Устройства аналоговой электроники обычно плавно изменяют ток, напряжение или сопротивление, чтобы добиться того же результата.

Оба типа устройств имеют свои преимущества и недостатки, но в современной электронике нельзя обойтись только одним из них. Например, для работы цифрового термометра на микроконтроллере необходим целый ряд аналоговых компонентов.

## Что такое микроконтроллер?



**Рис. 12.** Микроконтроллерная плата RedBoard компании SparkFun (вверху слева) и плата Arduino Uno (внизу справа)

Микроконтроллер — это небольшой компьютер, который можно программировать, загружая в него набор инструкций, называемый *программой*. Микроконтроллеры применяются для автоматизации простых операций — например, управления температурой воздуха в помещениях или поливкой газона.

В проектах этой книги используется микроконтроллерная плата RedBoard компании SparkFun, которая на 100% совместима с платой Arduino Uno. Обе эти платы показаны на рис. 12.

В любой день вы в среднем можете использовать 15–20 микроконтроллеров, даже не подозревая об этом. Микроконтроллеры управляют вашей кофеваркой, будильником, микроволновкой и другими бытовыми устройствами (более-менее современными, конечно, а не доставшимися вам по наследству от бабушки). Ваш автомобиль может использовать от 5 до 10 микроконтроллеров для управления системой зажигания, тормозами,

### Примечание

Вы узнаете больше о плате Arduino, включая ее возможности и как ее программировать, в процессе реализации проектов. А сейчас вам будет достаточно знать, что микроконтроллер — это программируемый кремниевый «мозг», который значительно облегчает превращение идеи об автоматизировании какого-либо аспекта нашей жизненной деятельности в прототип электронного устройства, реализующего эту автоматизацию.

аудиосистемой и т. п. Современный мир практически стоит на микроконтроллерах. Эта книга поможет вам научиться использовать это обстоятельство для своей выгоды.

Мы надеемся, что представленное вам здесь краткое введение в основы электроники дало вам достаточно знаний для работы с дальнейшим материалом книги. Мы также надеемся, что вы найдете этот материал интересным, увлекательным и полезным. Итак, вперед, к первому проекту!

### ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ ПО ОСНОВАМ ЭЛЕКТРИЧЕСТВА И ЭЛЕКТРОНИКИ

Если вы желаете узнать больше об электричестве и электронике, мы рекомендуем вам ознакомиться со следующими книгами:

- Basic Electricity by the Bureau of Naval Personnel (Основы электричества. Учебник Бюро флотского персонала), Dover Publications, 1970.
- Arduino Workshop by John Voxall (Джон Баксол. Мастерская для Arduino), No Starch Press, 2013.
- Getting Started in Electronics by Forrest M. Mims III (Форрест М. Мимс III. Первые шаги в электронике), Master Publishing, 2003.
- Practical Electronics for Inventors, 4th edition by Paul Scherz and Simon Monk (Поль Шерц, Саймон Монк. Практическая электроника для изобретателей), McGraw-Hill Education, 2016.

Читателям в России могут быть полезны и такие издания на русском языке:

- Платт Ч. Электроника для начинающих, 2-е изд. — СПб.: БХВ-Петербург, 2017. — 416 с.
- Блум Дж. Изучаем Arduino: инструменты и методы технического мастерства. — СПб.: БХВ-Петербург, 2015. — 336 с.
- Ревич Ю. В. Занимательная электроника, 5-е изд. — СПб.: БХВ-Петербург, 2018.
- Монк С., Шерц П. Электроника. Теория и практика. — СПб.: БХВ-Петербург, 2018. — 1168 с.

1

# НАЧАЛО РАБОТЫ С ARDUINO

В этом проекте содержится вся необходимая информация, чтобы начать работать с Arduino. В нем мы познакомим вас с самой платой Arduino, расскажем, как установить среду программирования, а также поможем вам проверить работоспособность собранной схемы, загрузив и выполнив простую программу. Если вы все сделаете правильно, на плате должен будет мигать встроенный светодиод, — это станет подтверждением того, что вы уже готовы двигаться дальше, к новым увлекательным приключениям с Arduino.

## Необходимые компоненты

Для реализации этого проекта вам потребуются следующие компоненты (рис. 1.1):

- плата Arduino Uno (DEV-11021), или плата RedBoard компании SparkFun (DEV-13975), или любая другая совместимая с Arduino плата, 1 шт.;
- кабель USB (A – B) (CAB-00512) для платы Arduino Uno или кабель Mini-B USB (CAB-1101) для платы RedBoard, 1 шт.;
- светодиод (COM-09590 или COM-12062 для пакета, содержащего 20 шт.), 1 шт.



Рис. 1.1. Компоненты для первого проекта

## О плате Arduino

Плата Arduino представляет собой небольшое программируемое устройство, с помощью которого обычным электронным схемам можно придать определенный уровень интеллектуальности. Ее можно использовать для управления роботами, создания графики на светодиодах или даже как ручную игровую консоль. В этом разделе мы рассмотрим более подробно, что такое Arduino и как она может изменить ваше мировоззрение.

### Доступная аппаратная платформа

Плата Arduino подобна небольшому компьютеру. Ее можно программировать, используя очень

простые инструкции, а для ее питания достаточно лишь нескольких батареек типоразмера AA. Но Arduino отличается от обычного компьютера тем, что для обработки информации и выполнения действий в ней используется не процессор, а микроконтроллер — небольшая интегральная схема, выполняющая в проектах функцию своеобразного мозга. Она может получать входные данные от датчиков (например, оптических датчиков, датчиков температуры или просто переключателей) и выдавать выходные сигналы для управления светодиодами, электродвигателями, зуммерами и т. п. Плата Arduino (рис. 1.2) содержит все вспомогательные компоненты и схемы, необходимые для работы ее микроконтроллера.



Рис. 1.2. Arduino Uno — открытая программируемая платформа для любителей электроники

Программируется Arduino на языке, который фактически является версией языка C/C++. В качестве среды программирования используется Arduino IDE (интегрированная среда разработки). Разработчики среды оснастили ее многими готовыми функциями и библиотеками функций, чтобы упростить создание кода для интерфейса платы Arduino с другими аппаратными устройствами. Например, одной из таких функций является функция включения светодиода, которая сводит множество строк кода всего лишь в одну инструкцию.

## Плата RedBoard компании SparkFun

Существует много плат, официально использующих марку Arduino, но поскольку это открытая платформа (то есть исходный дизайн ее аппаратной и программной части доступен для просмотра и модифицирования любому желающему), существует также множество плат, производных от Arduino, ее клонов и совместимых с ней. Все эти платы разработаны по лицензии Creative Commons «Attribution-ShareAlike» (Атрибуция — С сохранением условий), и в часто задаваемых вопросах по Arduino (<https://www.arduino.cc/en/Main/FAQ>) указывается, что любой желающий «может свободно использовать и адаптировать [этот дизайн] для своих собственных нужд без необходимости спрашивать разрешения или платить за это». Производные платы работают с той же средой программирования, что и официальная Arduino, но часто их аппаратная часть настроена на лучшую производительность или модифицирована каким-либо иным образом.

Плата RedBoard компании SparkFun (рис. 1.3) является производной платой, совместимой с Arduino. Она основана на дизайне Arduino Uno, но имеет более стабильный интерфейс USB и оснащена мини-разъемом USB вместо разъема USB типа A. Во всем остальном она точно такая же, как и плата Arduino Uno, такой же формы и размеров.

Плата RedBoard содержит несколько ключевых компонентов, назначение которых необходимо

знать, чтобы быть в состоянии разбираться с материалом нескольких первых глав книги. Эти компоненты обозначены на рис. 1.3, а далее приводится их описание:

- **микроконтроллер ATmega328** — черная квадратная микросхема посередине платы. Это мозг Arduino;
- **Штыревые разъемы** — металлические выводы на плате, позволяющие считывать с платы выходные сигналы и отправлять ей входные. Организованы они в виде четырех групп разъемов, по две на каждой стороне платы Arduino. Каждый штыревой разъем пронумерован или обозначен его функцией. Нас будут интересовать разъемы **Digital** (0–13), **Analog In** (A0–A5) и **Power**;
- **Порт мини-USB** — этот порт используется для взаимодействия с Arduino и для его программирования. Для большинства проектов этой книги этот порт мини-USB также служит для подачи на плату питания. Если для платы требуется внешний источник питания, это будет обязательно указано;
- **Индикатор питания** — этот светодиод служит индикатором подачи питания на плату. Если по какой-либо причине на плату не поступает питание, светодиод не будет гореть;
- **Светодиоды TX/RX** — эти светодиоды мигают при передаче (TX) или приеме (RX) платой каких-либо данных;



**Рис. 1.3.** Плата RedBoard компании SparkFun, совместимая с платой Arduino Uno. Обратите внимание, что форма платы совпадает с формой платы Arduino Uno, показанной на рис. 1.2

- **Встроенный светодиод 13** — применяется для отладки. При первом включении Arduino, когда он не содержит никаких пользовательских программ, светодиод 13 должен мигать с частотой один раз в секунду. Этот светодиод подключен к выводу 13 Arduino, отсюда и его название;
- **Разъем внешнего питания** — цилиндрический разъем (гнездо) рядом с портом USB. Для питания Arduino требуется напряжение

5 В, хотя на плату можно спокойно подавать напряжение величиной от 7 до 15 В, не опасаясь ее повредить. Специальная микросхема на плате понижает это напряжение до 5 В, требуемых для правильной работы электронных компонентов платы.

Как и в случае со всеми платами, совместимыми с Arduino, для программирования платы RedBoard используется интегрированная среда разработки Arduino IDE.

## Установка Arduino IDE и драйверов

Прежде чем впервые подключать плату Arduino (или RedBoard) к порту USB компьютера, на компьютер необходимо установить среду разработки Arduino IDE.

### Примечание

Если вы все-таки подключили плату до установки среды и драйверов, не волнуйтесь, это не станет проблемой. Придется после установки среды просто перезагрузить компьютер, чтобы драйверы работали должным образом.

### Access the Online IDE



**ARDUINO WEB EDITOR**  
Start coding online with the Arduino Web Editor, save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards. The Arduino Web Editor is one of the Arduino Create platform's tools.

[Try It Now](#)  
[Getting Started](#)

### Download the Arduino IDE



**ARDUINO 1.8.1**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows Installer**  
Windows ZIP file for non admin install

**Windows app** [Get](#)

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

Рис. 1.4. Можно использовать онлайн-версию Arduino IDE или же загрузить последнюю версию для своей операционной системы

Среду разработки Arduino IDE можно загрузить по адресу: [www.arduino.cc/download](http://www.arduino.cc/download). На странице загрузки определитесь с версией среды, соответствующей операционной системе вашего компьютера, а затем щелкните необходимую ссылку (рис. 1.4). На следующей странице вас спросят, не желаете ли вы сделать взнос на развитие программного обеспечения Arduino (разработка и поддержка Arduino IDE выполняется за счет помощи и взносов сообщества ее пользователей).

Даже если у вас уже установлена Arduino IDE, рекомендуется загрузить и установить последнюю версию этой среды разработки. Среда Arduino IDE постоянно улучшается и обновляется, поэтому лучше пользоваться самым последним ее выпуском. Примеры в этой книге используют Arduino IDE версии 1.8.1 или более поздней.

На домашней странице веб-сайта Arduino также предоставляется возможность использования платформы Arduino Create, включая редактор кода с браузерным интерфейсом (см. рис. 1.4, *вверху*). Эта платформа позволяет программировать Arduino посредством веб-браузера, а также просматривать проекты и делиться ими с другими непосредственно в Интернете. На момент подготовки этой книги платформа поддерживается только операционной системой Windows и OS X.

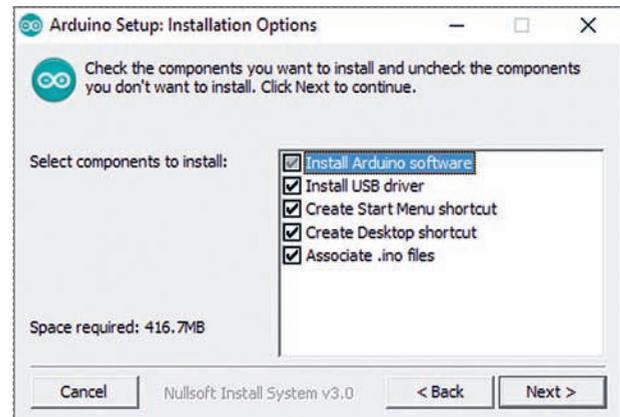
Независимо от вашего выбора: использовать Arduino Create или загрузить среду разработки — следуйте инструкциям для установки.

### Примечание

Если вы хотите использовать самую передовую версию программного обеспечения, на странице загрузок также предоставляются «ночные» сборки для следующего выпуска. Но для работы с этой книгой рекомендуется использовать только последний стабильный выпуск.

## Установка под Windows

Для компьютеров на ОС Windows предлагается загрузить версию Arduino IDE Windows Installer. Загрузите этот установочный файл в подходящую папку на своем компьютере и запустите его на выполнение. В окне **Installation Options** (рис. 1.5) установите флажок **Install Arduino software** (Установить программное обеспечение Arduino) (если он еще не установлен), а также флажки всех других опций, чтобы не устанавливать драйверы отдельно. Нажмите кнопку **Next** (Далее) и в следующем окне укажите папку, в которую следует установить программное обеспечение Arduino (рекомендуется выполнить установку в предлагаемую папку по умолчанию), и нажмите кнопку **Install** (Установить).



**Рис. 1.5.** Диалоговое окно **Installation Options** для установки программного обеспечения для Arduino. В обязательном порядке установите флажок для установки драйверов USB

Процесс установки может занять несколько минут, поэтому можете в это время сделать себе чашечку кофе. В зависимости от вашей версии Windows, в процессе установки может потребоваться дать разрешение на установку драйверов и, при желании, установить флажок, подтверждающий ваше доверие программному обеспечению от Arduino: **Always trust software from "Arduino srl"** (рис. 1.6).

Если вы не хотите подтверждать установку каждого драйвера по отдельности, установите этот флажок, разрешающий их установку по умолчанию. В любом случае нажмите кнопку **Install**, чтобы установить драйверы USB. Вот и все!

Установщик Arduino IDE обычно помещает ярлык среды программирования на рабочем столе. Щелкните на этом ярлыке двойным щелчком, чтобы запустить среду разработки Arduino IDE.



Рис. 1.6. Диалоговое окно разрешения установки драйверов

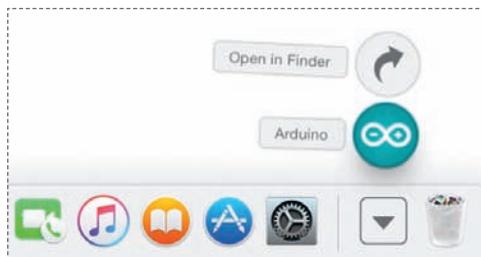


Рис. 1.7. Загруженный установочный пакет Arduino IDE будет сохранен в папку **Загрузки**. Щелкните опцию **Open in Finder**, чтобы переместить его в папку **Программы** (Applications)

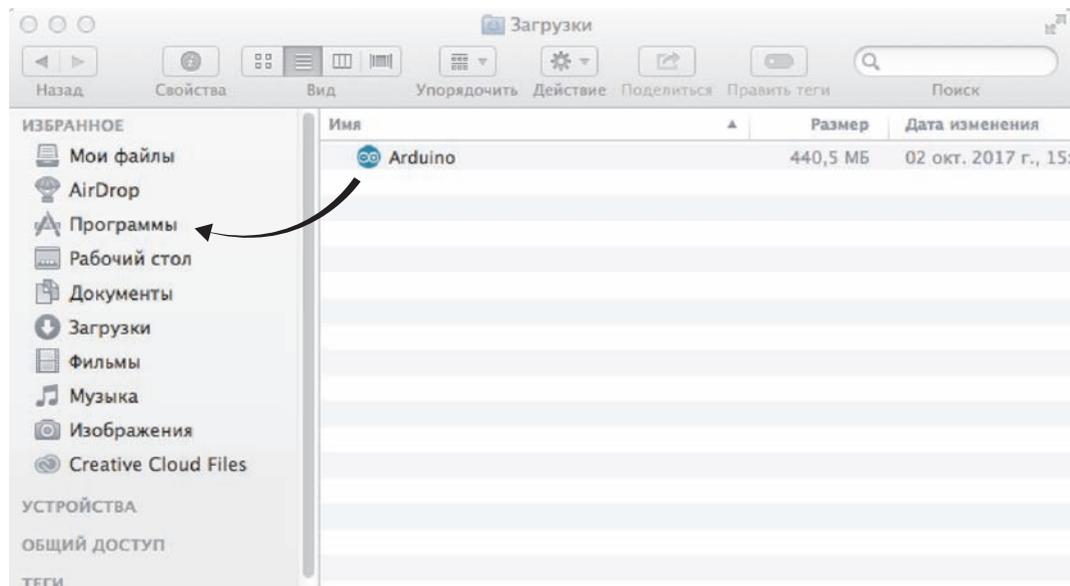


Рис. 1.8. Выберите установочный файл Arduino IDE и перетащите его в папку **Программы** (Applications)

## Установка под OS X

Для компьютеров Mac щелкните ссылку на версию Arduino IDE для OS X и следуйте инструкциям на открывшейся странице.

### Установка среды Arduino IDE

Загрузив установочный пакет, наведите курсор мыши на папку **Загрузки** и выберите опцию **Open in Finder** (Открыть в Finder) (рис. 1.7).

Затем выберите установочный файл Arduino IDE и перетащите его в папку **Программы** (Applications) (рис. 1.8). В большинстве случаев вам не придется устанавливать что-либо еще, и Arduino IDE можно будет запускать, как любую другую программу.

### Ручная установка драйвера FTDI на OS X для платы RedBoard

Драйверов, установленных со средой разработки Arduino IDE, будет *достаточно для работы стандартной платы Arduino Uno*. Но для платы RedBoard от SparkFun необходимо вручную установить дополнительный драйвер. Для взаимодействия с компьютером в плате RedBoard

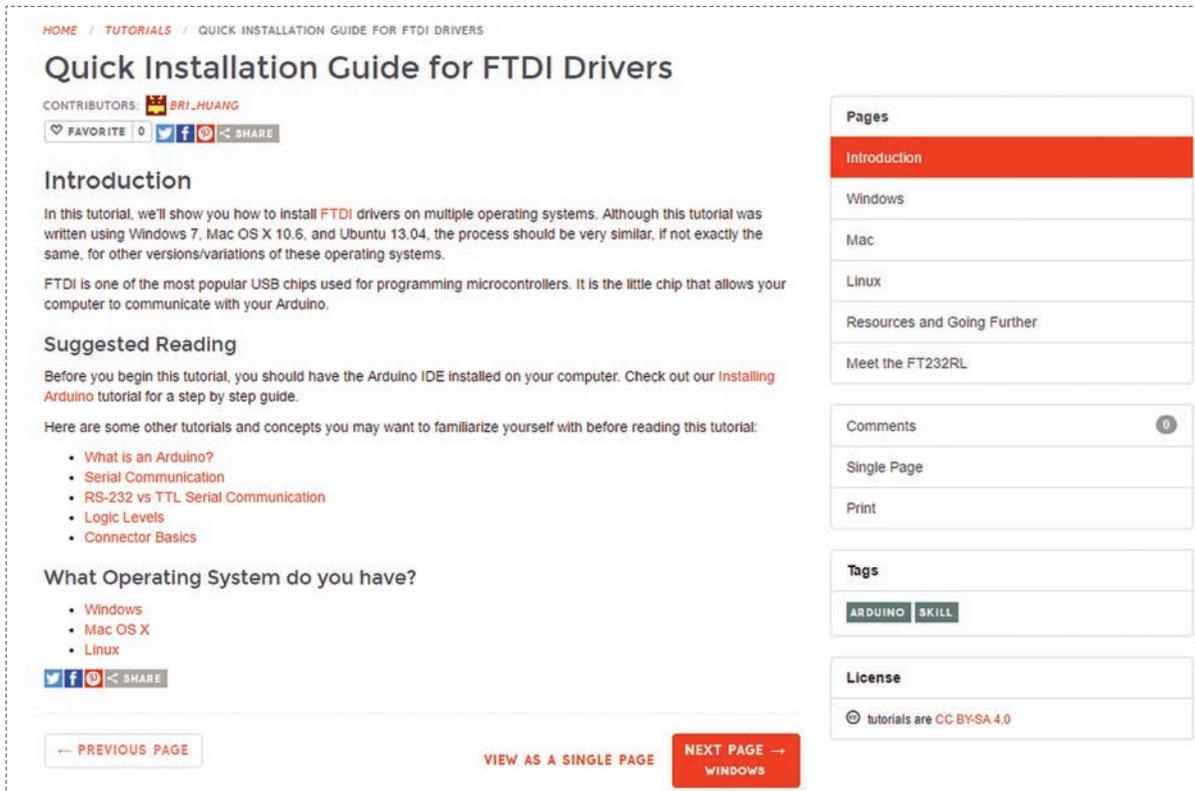


Рис. 1.9. Руководство по установке драйверов FTDI для платы RedBoard

от SparkFun используется микросхема USB компании Future Technology Devices International (FTDI). Вот для этой микросхемы и необходимо вручную установить драйвер. Для этого сначала откройте веб-страницу учебного пособия по установке драйверов FTDI (рис. 1.9) по адресу: [www.sparkfun.com/ftdi](http://www.sparkfun.com/ftdi).

Выберите ссылку для Mac OS X — будет выполнен переход в раздел с опциями выбора необходимого драйвера в зависимости от используемой версии OS X: один драйвер для версий операционной системы с Mac OS X 10.3 (Panther) по 10.8 (Mountain Lion) и другой — для версии OS X 10.9 (Mavericks) и более новых версий.



Рис. 1.10. Ручная установка драйвера FTDI на OS X

Загрузите необходимый драйвер, а затем щелкните на сохраненном файле двойным щелчком, чтобы запустить процесс установки. Откроется обычное окно установки программного обеспечения для Mac. Укажите жесткий диск, когда он будет обнаружен, и нажмите кнопку **ОК**. Запустится процесс установки, статус которого будет отображаться в индикаторе выполнения (рис. 1.10). Драйверы будут установлены, когда индикатор дойдет до конца.

Вот и все! Теперь можно запустить среду разработки, щелкнув двойным щелчком на ярлыке Arduino в папке **Applications**. Если среда была запущена до установки драйверов FTDI, нужно будет закрыть ее, а затем снова запустить. Это необходимо, чтобы правильно отображались последовательные порты.

### Примечание

В случае каких-либо проблем при установке драйверов, возможные решения можно найти по адресу: [www.sparkfun.com/macdriver](http://www.sparkfun.com/macdriver).

## Установка под Linux

Среда разработки Arduino доступна также и для пользователей Linux. Загрузите необходимую версию установочного файла среды для Linux: 32- или 64-разрядную. Затем распакуйте загруженный файл, используя программу xz-utils или любую другую программу архивации. Для самой последней версии Arduino IDE для Linux, возможно, придется также установить некоторые другие зависимые программы. Информацию о зависимостях

для конкретного дистрибутива Linux можно получить на веб-странице: <http://playground.arduino.cc/Learning/Linux>.

Для большинства дистрибутивов Linux (включая Ubuntu, Debian и Fedora) Arduino IDE можно установить из командной строки с помощью диспетчера пакетов apt-get. Откройте терминал и введите следующую команду:

```
sudo apt-get install arduino
```

### Примечание

В зависимости от диспетчера пакетов вашего дистрибутива Linux, установленная таким способом версия среды разработки Arduino может оказаться не самой последней доступной на веб-сайте Arduino.

По завершении процесса установки запустите установленную среду разработки Arduino. Среда разработки Arduino работает на Java и должна запускаться в X Window или другой подобной среде пользовательского интерфейса.

## Краткая экскурсия по среде разработки Arduino

Среда разработки Arduino используется для создания и отладки программ для Arduino. По терминологии Arduino программы называются *скетчами*. Среда разработки позволяет загрузить в Arduino скетч и управлять физическими устройствами.

Если вы этого еще не сделали, откройте сейчас установленную среду разработки Arduino. После начальной заставки откроется окно собственно IDE (рис. 1.11).

Панель меню (поз. ❶ на рисунке) содержит команды **Файл** (File), **Правка** (Edit), **Скетч** (Sketch), **Инструменты** (Tools) и **Помощь** (Help), с помощью которых можно открывать и сохранять файлы, загружать в Arduino код, редактировать параметры и т. п. Под строкой меню находится

панель с кнопками команд (поз. ❷, слева направо):  **Проверить** (Verify),  **Загрузка** (Upload),  **Новый** (New),  **Открыть** (Open) и  **Сохранить** (Save). Назначение этих меню и кнопок мы рассмотрим в процессе реализации проектов в книге.

Большая часть окна среды разработки состоит из белого поля (поз. ❸), в которое вводится код разрабатываемых скетчей. Под областью кода находится *строка предупреждений* (❹), а под ней — *окно сообщений* (❺). В этих элементах интерфейса отображаются состояние программы, предупреждения и ошибки. Так, если скетч содержит синтаксическую ошибку (например, неправильно набранную команду), в строке предупреждений

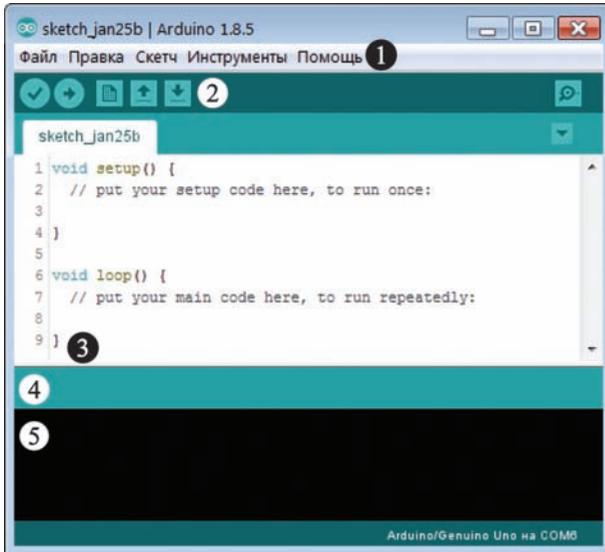


Рис. 1.11. Главное окно интегрированной среды разработки Arduino

будет выведено краткое сообщение об этом, а в окне сообщений — более подробная информация, включая действия среды разработки касательно ошибки. Поясним это на более конкретном примере. Если в окне кода набрать какие-либо слова, например свое имя, а затем нажать кнопку

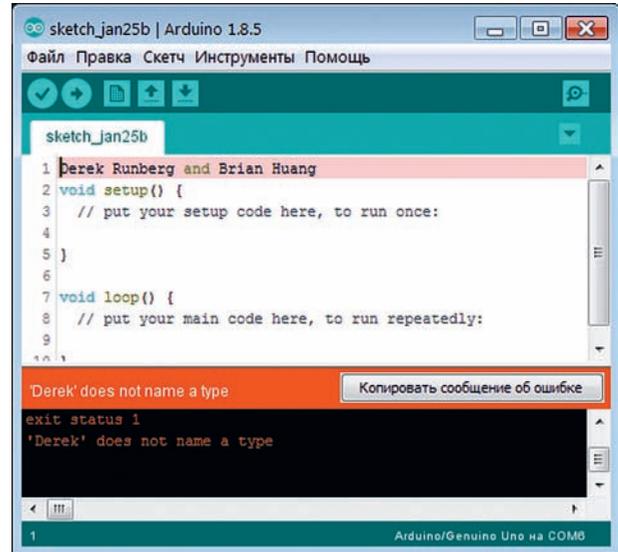


Рис. 1.12. Пример вывода предупреждения об ошибке в строке предупреждений и детальной информации о ней в окне сообщений

проверки  (**Проверить**), Arduino IDE задумается на мгновение, после чего выведет в строке предупреждений сигнал о наличии ошибки, а в окне сообщений — подробную информацию об этой ошибке (рис. 1.12).

## Изменение настроек по умолчанию

Arduino IDE является полностью открытой и настраиваемой средой программирования. Некоторые из ее второстепенных параметров доступны для настройки пользователем, чтобы облегчить ему написание и отладку кода для создания программ. Для просмотра и настройки общих параметров Arduino IDE выполните последовательность команд меню **Файл | Настройки** (File | Preferences) — откроется окно **Настройки** (Preferences) (рис. 1.13).

Один из рекомендуемых для настройки параметров — размер шрифта редактора кода в поле **Размер шрифта** (Editor font size), который можно задать таким, чтобы вам было удобно читать содержимое редактора. Также будет полезным

установить флажок для отображения нумерации строк **Показать номера строк** (Display line numbers) и снять флажок **Сохранять скетч при проверке или компиляции** (Save when verifying or uploading). Нумерация строк облегчает ориентировку в коде, а запрет на автоматическое сохранение файла скетча при каждой его проверке или загрузке в Arduino немного ускорит тестирование кода. Поскольку, как уже было отмечено ранее, платформа Arduino является полностью открытой, многие другие ее параметры также доступны для настройки посредством редактирования файла *preferences.txt* — см. опцию **Другие настройки можно редактировать непосредственно в файле** (More preferences can be edited directly in the file).

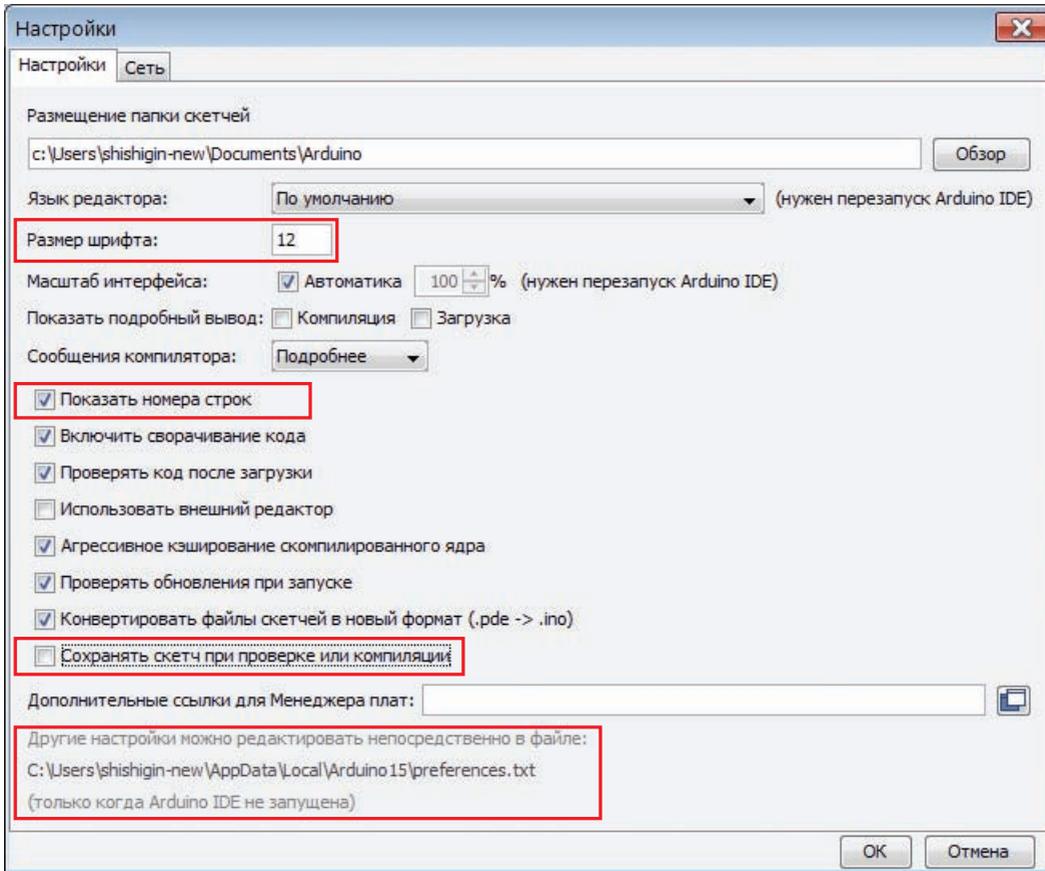


Рис. 1.13. Окно **Настройки** для настройки параметров Arduino IDE

## Первое подключение Arduino к компьютеру

Установив среду разработки и драйверы, можно подключить плату Arduino к порту USB компьютера, используя соответствующий кабель. При подключении должен загореться светодиод питания (обозначен на плате меткой **ON**), а если плата не содержит никаких пользовательских программ, также должен мигать и светодиод, обозначенный меткой **13** (так называемый *светодиод 13*) (рис. 1.14).

### Примечание

Если плата подключалась к компьютеру до установки среды разработки и драйверов, может понадобиться перезагрузить компьютер после установки этого программного обеспечения.

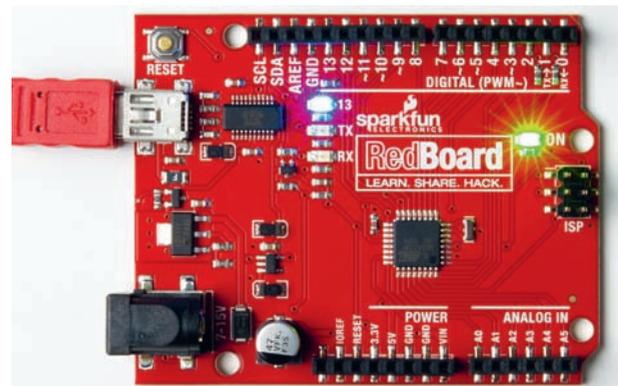


Рис. 1.14. При подключении к компьютеру платы, не содержащей пользовательских программ, кроме светодиода **ON**, мигает также светодиод **13**

В отличие от полноценного компьютера, Arduino может одновременно выполнять только одну программу (скетч). Плата Arduino получает питание от компьютера через кабель USB, которым она подключена к нему, и выполняет программу, загруженную производителем. Это стандартный скетч для проверки работоспособности платы, заключающийся в мигании светодиода. Когда плата Arduino подключена к компьютеру, в нее можно загружать другие скетчи, создаваемые в среде разработки.

## Указание подключенной платы в IDE

В зависимости от компьютера или операционной системы, компьютеру может потребоваться некоторое время, чтобы определить подключенное

к нему новое оборудование и сопоставить его с необходимыми драйверами. Когда компьютер определит подключенную плату, выберите команду меню **Инструменты** (Tools), а затем наведите курсор на опцию **Плата** (Board) — откроется список плат Arduino, поддерживаемых средой разработки (рис. 1.15).

Если вы используете стандартную плату Arduino Uno или плату SparkFun RedBoard, выберите опцию **Arduino/Genuino Uno**. В случае использования другой платы, выберите то название платы, которое прописано в ее документации. Как уже отмечалось ранее, в проектах этой книги предполагается использование платы Arduino Uno или производной от нее.

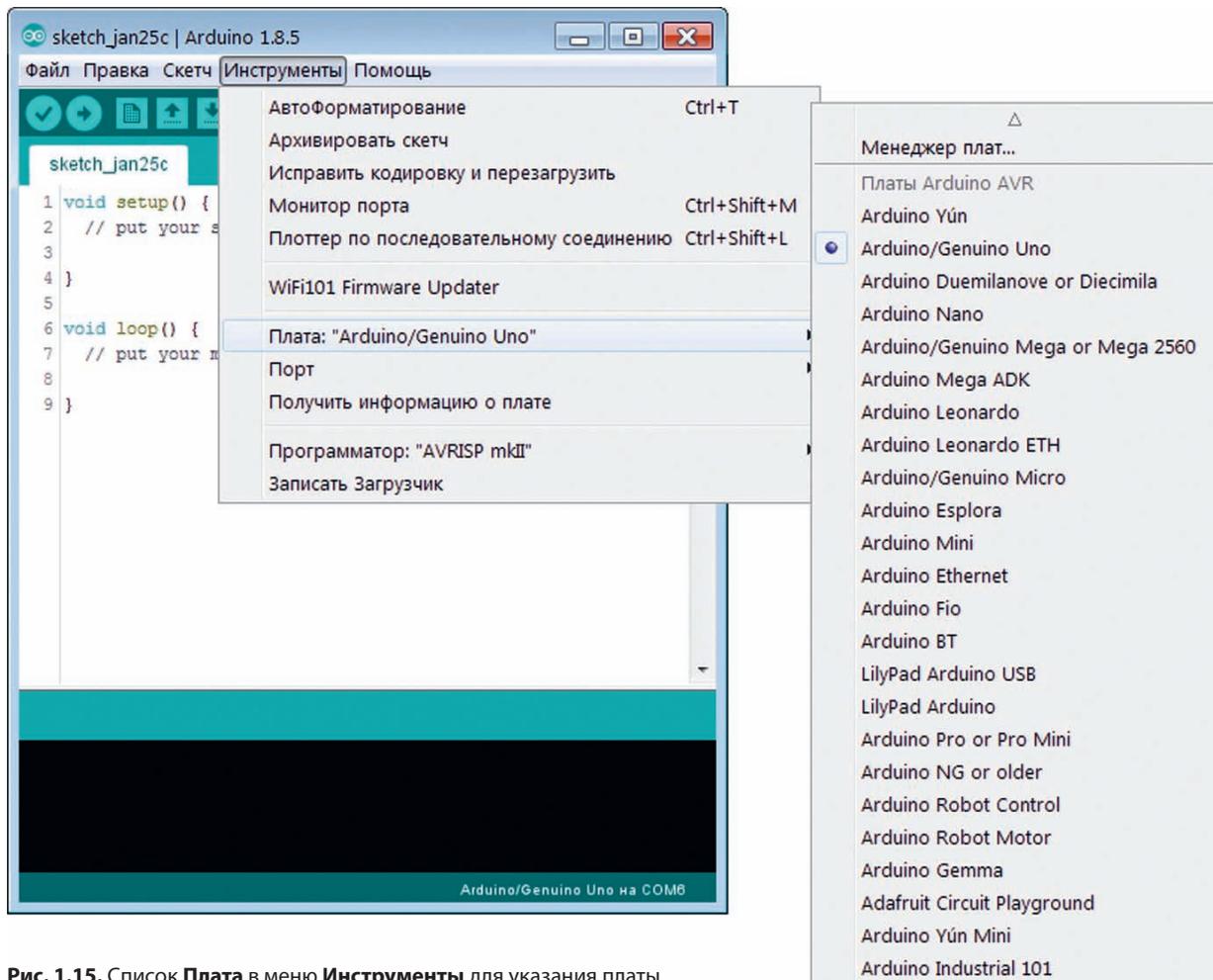


Рис. 1.15. Список **Плата** в меню **Инструменты** для указания платы

## Выбор последовательного порта COM

Подключаемым к компьютеру устройствам присваивается уникальный идентификатор последовательного порта COM. Нам необходимо указать среде разработки Arduino номер последовательного порта, выделенного системой для связи с платой Arduino. Для этого сначала выполните последовательность команд меню **Инструменты | Порт** (Tools | Port). Доступные в результате выполнения этой команды опции будут зависеть от типа операционной системы.

### Для Windows

Для компьютеров под Windows список будет содержать названия портов наподобие **COM3, COM4** и т. п. Выделенный для Arduino порт будет дополнительно обозначен (**Arduino/Genuino Uno**) (рис. 1.16). Выберите этот порт, щелкнув на нем мышью. В случае отсутствия опций выбора порта, обратитесь к разд. «Поиск и устранение основных проблем с Arduino» далее в этом проекте.

### Для Mac OS X и Linux

Для компьютеров под управлением Mac или Linux последовательный порт будет отображаться в формате `/dev/cu.usbserial-A<xxxx>`, где часть

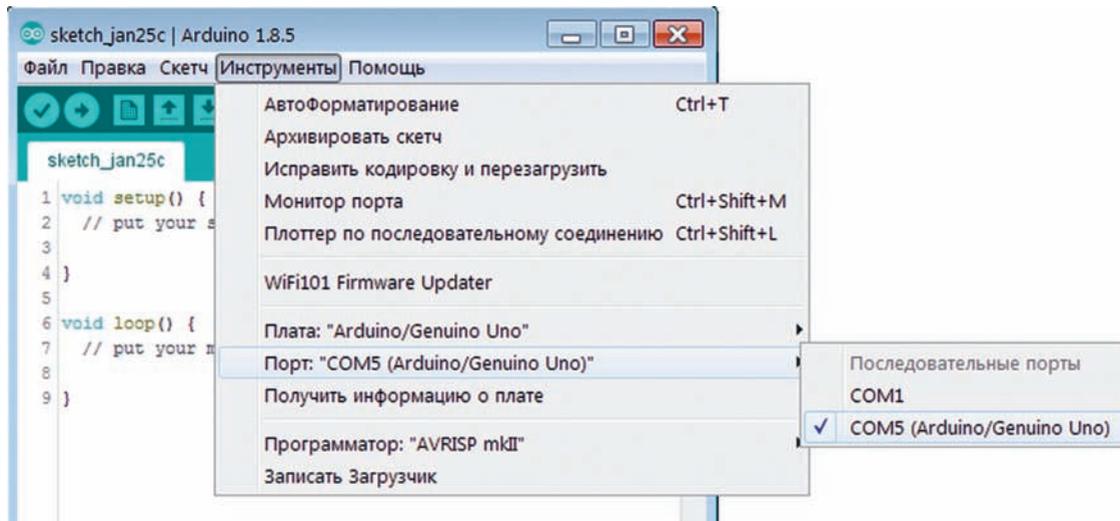


Рис. 1.16. Выбор последовательного порта COM для Arduino в компьютерах под управлением ОС Windows

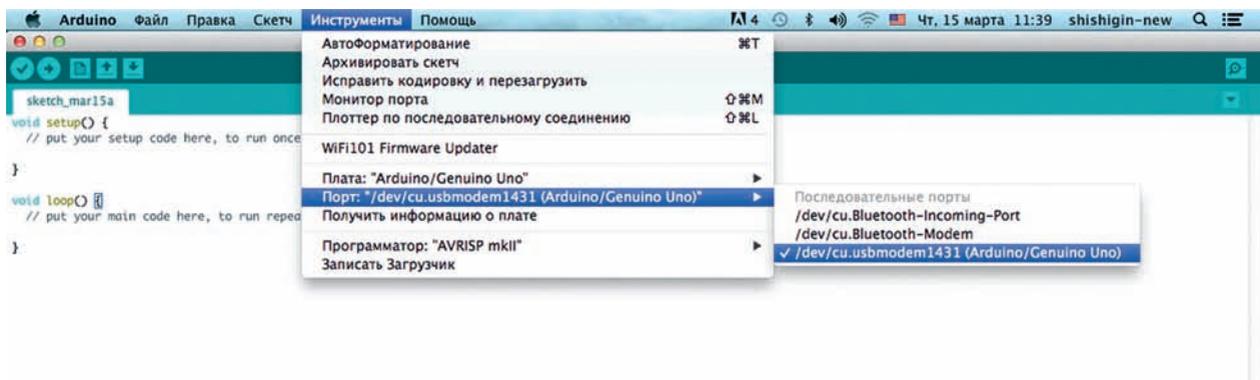


Рис. 1.17. Выбор последовательного порта COM для компьютеров под OS X и Linux

<xxxx> является строкой произвольных символов, уникальной для вашей платы Arduino. Выберите этот порт, щелкнув на нем мышью. Список может содержать несколько портов (рис. 1.17), но только

порт с этой уникальной строкой ID будет сопоставлен Arduino. В случае отсутствия опций выбора порта, обратитесь к *разд. «Поиск и устранение основных проблем с Arduino»* далее в этом проекте.

## Программа «Здравствуй, мир!» для Arduino

Программа «Здравствуй, мир!» является классической первой программой, создаваемой многими начинающими программистами. В большинстве языков программирования эта программа выводит на экран слова **Здравствуй, мир**. Но поскольку у Arduino нет экрана, то действием для него, сообщаящим, что программа выполняется, станет мигание встроенного светодиода 13.

Для нашего первого скетча мы воспользуемся примером, входящим в состав Arduino IDE. Подключите плату к компьютеру, в меню **Файл** (File) выберите последовательность команд **Примеры | 01. Basics | Blink** (Examples | 01. Basics | Blink) (рис. 1.18) — откроется новое окно редактора скетчей с загруженным в него скетчем Blink.

В этом окне выполните последовательность команд меню **Скетч | Загрузка** (Sketch | Upload) или щелкните на значке **Загрузка** (Upload) (см. *разд. «Краткая экскурсия по среде разработки Arduino»*). Среда разработки преобразует сравнительно доступный для человеческого понимания код скетча в набор нулей и единиц, который понимает Arduino (этот процесс называется *компиляцией*), а затем загрузит этот набор в плату.

Итак, щелкнув на значке **Загрузка**, наблюдайте за выводимыми сообщениями в строке предупреждений. Сначала здесь должно выводиться сообщение **Компиляция скетча...** (Compiling sketch...), а в правой части строки будет отображен индикатор выполнения компиляции. После

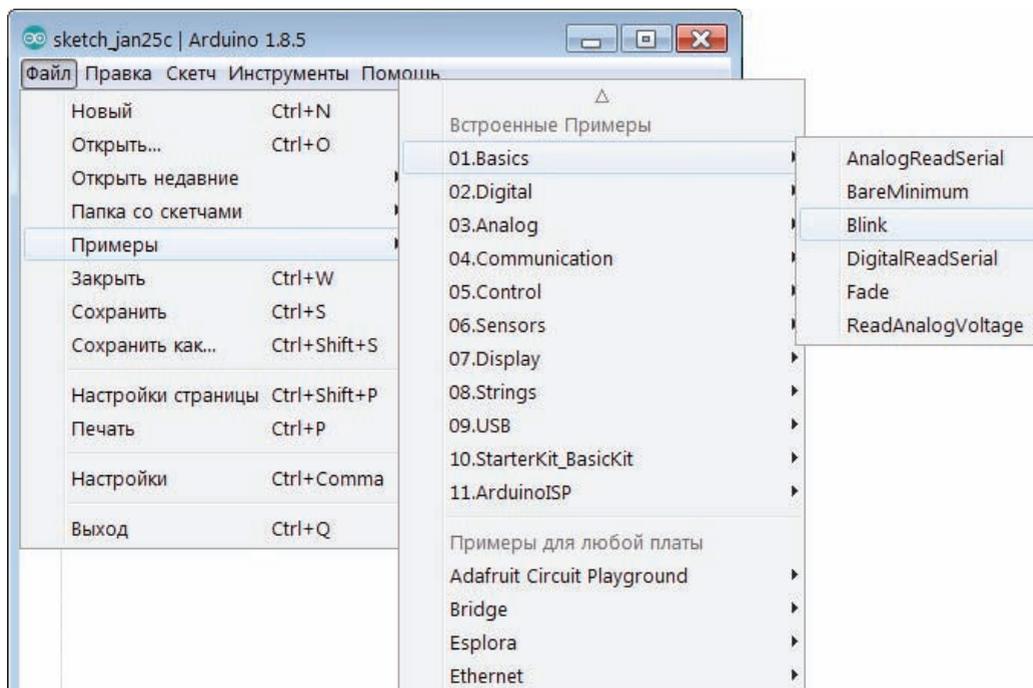


Рис. 1.18. Загрузка скетча Blink в редактор скетчей

завершения компиляции среда разработки начнет загружать скомпилированный скетч в плату Arduino, и в строке отобразится сообщение **Загрузка** (Uploading). При этом на плате Arduino



Рис. 1.19. Исполнение скетча: мигание светодиода 13

будут мигать светодиоды **RX** (прием) и **TX** (передача), подтверждая загрузку скетча в плату Arduino. Светодиод **RX** мигает, потому что Arduino принимает данные от компьютера, а светодиод **TX** — потому что Arduino отвечает компьютеру, подтверждая прием данных. По завершении загрузки скетча в плату в строке предупреждений выведется сообщение **Загрузка завершена** (Done uploading), светодиоды **TX** и **RX** мигать перестанут, и плата начнет исполнять загруженный скетч, мигая встроенным светодиодом 13 (рис. 1.19).

Если в процессе компиляции и загрузки что-либо происходит иначе, чем здесь описано, а в строке предупреждений выводятся какие-либо сообщения об ошибке, причиной может быть отсутствие взаимосвязи между компьютером и Arduino. Попробуйте решить проблему, руководствуясь советами, изложенными в разд. «Поиск и устранение основных проблем с Arduino», а затем попробуйте снова загрузить скетч в Arduino.

## Поиск и устранение основных проблем с Arduino

Как и любое другое программируемое электронное устройство, плата Arduino может иногда вести себя не так, как от нее ожидается. Здесь мы приводим несколько советов для таких случаев, чтобы помочь решить возможные проблемы.

1. Убедитесь, что кабель USB надежно вставлен как в разъем на плате Arduino, так и в разъем компьютера. Довольно легко вставить кабель в разъем на плате лишь частично. Попробуйте извлечь кабель из гнезда и затем снова вставить его.
2. Убедитесь, что в пункте **Плата** указана плата, которая фактически подключена к компьютеру. Для примеров в этой книге указывается плата Arduino/Genuino Uno.
3. Проверьте, что в пункте меню **Инструменты | Порт** выбран правильный порт. Выбранный порт обозначается галочкой или точкой. Если вы не знаете, какой порт выделен для
4. Проверьте, что вы случайно не добавили какие-либо лишние символы в скетч примера. В таком случае скетч не будет компилироваться.
5. На компьютере под Windows проверьте в Диспетчере устройств правильность установки драйверов для Arduino. В частности, посмотрите, не обозначен ли порт COM для платы Arduino восклицательным знаком. В таком случае нужно будет вручную переустановить драйверы.
6. Если сообщения об ошибках продолжают выводиться, переустановите драйверы платы. Дополнительные инструкции доступны на веб-странице [www.sparkfun.com/ftdi/](http://www.sparkfun.com/ftdi/).

Приведенные здесь шесть советов являются хорошей отправной точкой для новичков при решении наиболее распространенных проблем с Arduino. Если ни один из этих советов не поможет решить проблему, не паникуйте, сохраняйте спокойствие и помните, что не вы первый, кто

испытывал проблемы с Arduino. Если вы не имеете ни малейшего понятия, как решить вставшую перед вами проблему, можно попытаться найти ее решение на официальном форуме Arduino по адресу: <http://forum.arduino.cc/>.

## Анатомия скетча Arduino

Здесь мы пошагово рассмотрим скетч Blink, который загрузили в Arduino в разд. «Программа

"Здравствуй, мир!" для Arduino». Полный текст скетча приводится в листинге 1.1.

### Листинг 1.1. Скетч примера Blink

❶ /\*

Blink

Бесконечно включает светодиод на одну секунду, а затем выключает на одну секунду.

Большинство плат Arduino оснащены встроенным светодиодом, который доступен для управления пользователем. На платах Arduino Uno и Zero этот светодиод подключен к цифровому выводу 13, а на платах MKR1000 – к цифровому выводу 6. Системной константе LED\_BUILTIN присваивается правильное значение вывода подключения светодиода независимо от используемой платы.

Информацию о том, какой вывод конкретной платы Arduino подключен к встроенному светодиоду, можно узнать в технических спецификациях платы на сайте <https://www.arduino.cc/en/Main/Products>.

Этот код примера не защищен никакими авторскими правами.

модифицирован 8 мая 2014 г.  
Скоттом Фитцджеральдом (Scott Fitzgerald)

модифицирован 2 сентября 2016 г.  
Артуго Гвадалупи (Arturo Guadalupi)

модифицирован 8 сентября 2016 г.  
Колби Ньюманом (Colby Newman)

\*/

//Функция 'setup' выполняется один раз при подаче питания на плату  
//или при нажатии кнопки сброса.

```
❷ void setup() {
    //инициализируем цифровой вывод LED_BUILTIN для вывода данных
    pinMode(LED_BUILTIN, OUTPUT);
}

//функция 'loop' исполняется в бесконечном цикле
❸ void loop() {
    digitalWrite(LED_BUILTIN, HIGH);           //включаем светодиод(высокий уровень напряжения)
    delay(1000);                               //ждем одну секунду
    digitalWrite(LED_BUILTIN, LOW);          //выключаем светодиод(низкий уровень напряжения)
    delay(1000);                               //ждем одну секунду
}
```

---

При написании кода скетчей Arduino необходимо использовать специфические слова, знаки препинания и заглавные буквы. Эти элементы являются частью *синтаксиса* языка программирования. Чтобы среда разработки могла скомпилировать скетч должным образом, он должен быть написан словами и по правилам, которые эта среда понимает. Эти слова называются *ключевыми словами* — в зависимости от функции ключевых слов, в окне редактора среды разработки они отображаются шрифтами разного цвета: оранжевого, бирюзового или зеленого. Теперь давайте рассмотрим более подробно основные составные части нашего первого скетча.

## Ключевые элементы скетча

Скетч начинается с объявления нового *глобального пространства имен* (поз. ❶ скетча). Это область, в которой описывается выполняемое скетчем действие и которая часто также содержит другую информацию, такую как инициализация переменных и библиотечные операторы. Практически все скетчи содержат пространство имен. Пространство имен этого скетча содержит *комментарии*, объясняющие на обычном человеческом языке, что именно этот скетч делает. В редакторе скетчей среды разработки Arduino текст комментариев отображается шрифтом серого цвета. Комментарии длиной в одну строку

обозначаются двойной косой чертой (//) в начале комментария. Такие комментарии и ограничены одной строкой. Комментарии, занимающие более одной строки, заключаются в символы /\* в начале комментария и \*/ в конце. Обратите внимание, что не все комментарии размещены на отдельных строках, — некоторые из них находятся на одной строке с текстом кода, который они поясняют. Это никоим образом не влияет на исполнение скетча, поскольку IDE игнорирует комментарии. В отличие от кода, в комментариях можно использовать любые слова, пунктуацию и орфографию.

Основу любого скетча Arduino составляет определение двух основных функций: `setup()` (поз. ❷) и `loop()` (поз. ❸). *Функция* — это просто группа инструкций или строк кода, выполняющих определенную задачу. Каждая функция имеет тип данных, имя и несколько инструкций. Определение функции начинается с объявления *типа данных*, возвращаемых функцией. Обе функции: `setup()` и `loop()` — имеют тип данных `void`, что означает, что они не возвращают никаких значений.

За типом данных следует *имя функции*, содержащее в конце открывающую и закрывающую скобки. В эти скобки помещаются параметры, которые передаются функции. Параметры представляют собой значения, требуемые функции для выполнения ее работы. Функции `setup()` и `loop()` не требуют

никаких параметров, но в дальнейших проектах мы увидим функции, которые используют их. Наконец, далее следуют строки кода, составляющие собственно функцию, которые заключены в фигурные скобки {}.

Каждый скетч Arduino должен содержать функции `setup()` и `loop()`. При подаче питания на плату или при нажатии кнопки сброса сначала один раз исполняется функция `setup()`, а затем начинает бесконечно исполняться функция `loop()`. Использование этих функций сходно выпеканию пирожков: инструкции в функции `setup()` приготавливают все ингредиенты и принадлежности, после чего функция `loop()` выпекает партию за партией пирожки, пока вы не выключите печь (т. е. Arduino).

Теперь давайте рассмотрим, что делает каждая строка кода в функциях `setup()` и `loop()`.

## Функция `setup()`

Код функции `setup()` для скетча Blink приводится в листинге 1.2.

### Листинг 1.2. Код функции `setup()` для скетча Blink

```
void setup() {
    //инициализируем цифровой вывод
    //LED_BUILTIN для вывода данных
    pinMode(LED_BUILTIN, OUTPUT);
}
```

Эта функция содержит единственную строку кода, вызывающую функцию `pinMode()`. Контактные выводы 0–13 платы Arduino являются *контактами ввода/вывода общего назначения* — соответственно, их можно использовать как для ввода, так и для вывода данных. Режим работы любого из этих контактов задается функцией `pinMode()`. Для этого функции передаются два параметра: первый параметр указывает номер контакта — его значение может быть от 0 до 13, второй параметр определяет его конфигурацию.

Для обращения ко встроенному светодиоду в скетче Blink используется системная константа `LED_BUILTIN`, которая автоматически определяет необходимый контакт. Для большинства устройств Arduino константа `LED_BUILTIN` имеет значение 13. Обратите внимание, что в скетче эта константа отображается шрифтом сине-зеленого цвета заглавными буквами. Этот цвет обозначает, что `LED_BUILTIN` является ключевым словом с предопределенным значением, которое используется средой разработки.

Второй параметр (`OUTPUT`) — конфигурирует этот контакт для вывода данных. Обратите внимание, что ключевое слово `OUTPUT` также сине-зеленого цвета. Это означает, что это слово является еще одной константой, используемой в Arduino. Здесь есть еще несколько других вариантов, которые мы подробно рассмотрим в *проектах 4 и 9*, но на данном этапе будет достаточно знать, что этот код задает функционирование вывода в качестве `OUTPUT` для LED.

Простыми словами этот код можно выразить следующим образом: «контакт 13 будет выводить данные с Arduino».

Код вызова функции `pinMode()` завершается точкой с запятой, которая обозначает конец строки кода. Строки кода скетчей всегда должны заканчиваться точкой с запятой. Но если вы случайно забудете сделать это, не отчаивайтесь. Практически все программисты, одни чаще, другие реже, делают эту ошибку. Однако среда разработки Arduino всегда обнаружит ее и выдаст соответствующее сообщение, чтобы помочь вам определить ошибку и исправить ее.

### Примечание

Имена функций пишутся с использованием так называемого *верблюжьего регистра*<sup>1</sup>. В этом формате имя переменной начинается со строчной буквы, а все остальные слова, составляющие имя переменной, начинаются с прописной буквы.

<sup>1</sup> От англ. camel case.

## А ГДЕ ФУНКЦИЯ MAIN()? 1

Если вы немного знакомы с программированием и, в частности, с языками программирования C или C++, то можете задаться вопросом, где в скетчах Arduino функция main(). Загляните в папку программ Arduino — вы много узнаете о том, что в действительности происходит: после щелчка на значке **Verify/Compile** или **Upload** Arduino в фоновом режиме добавляет в код много других файлов, включая файл main.cpp. Далее приводится отрывок кода из файла main.cpp:

```
int main(void)
{
    init();
    initVariant();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    ❶ setup();
    for (;;)
    {
        ❷ loop();
        if (serialEventRun)
            serialEventRun();
    }
    return 0;
}
```

Видите, где происходит вызов функции setup() (поз. ❶)? Также обратите внимание, что функция loop() (поз. ❷) находится внутри бесконечного цикла, который Arduino реализует с помощью пустого цикла for(;;). Этим и объясняется ее бесконечное исполнение.

## Функция loop() 2

Теперь давайте рассмотрим функцию loop(), которая бесконечно повторяет исполнение всех содержащихся в ней инструкций от первой до последней. Код функции приводится в листинге 1.3.

### Листинг 1.3. Код функции loop() скетча Blink

```
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
        //включаем светодиод
        //(высокий уровень напряжения)

    delay(1000); //ждем одну секунду

    digitalWrite(LED_BUILTIN, LOW);
        //выключаем светодиод
        //(низкий уровень напряжения)

    delay(1000); //ждем одну секунду
}
```

Инструкция digitalWrite() позволяет нам включать и выключать выводы Arduino — такое действие называется *управлением состоянием вывода*. Эта функция также принимает два параметра: первый параметр указывает вывод (контакт), которым мы хотим управлять. В данном случае мы снова используем системную константу LED\_BUILTIN. А второй параметр определяет состояние этого вывода. Чтобы включить подключенный к выводу светодиод, скетч Blink посылает на вывод сигнал высокого уровня (HIGH), а чтобы выключить — сигнал низкого уровня (LOW).

Вторая задействованная в функции loop() инструкция (delay()) приостанавливает исполнение скетча на количество миллисекунд, указанное в передаваемом этой инструкции параметре. Плата Arduino Uno и совместимые с ней платы, такие как плата SparkFun RedBoard, выполняют 16 миллионов инструкций в секунду. Это настолько быстро, что человеческий глаз не в силах заметить изменение состояния светодиода, если соответствующие инструкции просто следуют друг за другом. Инструкция *приостановления исполнения скетча* позволяет нам управлять длительностью текущего

состояния светодиода. Например, инструкция `delay(1000)` приостанавливает выполнение скетча на 1000 мс, прежде чем исполнять следующую команду.

Следующие две строки кода похожи на первые две: первая указывает Arduino выключить светодиод, а вторая создает задержку в 1000 мс, прежде чем исполнять следующую инструкцию. После исполнения последней инструкции функция `loop()` повторно исполняет все содержащиеся в ней инструкции, и так до бесконечности.

### Примечание

Одним из лучших способов обучения с помощью кода примера будет его модификация. Попробуйте уменьшить значение времени задержки до 500 и щелкните на значке **Upload**. Как это повлияло на скорость мигания? А что, если передать функции `delay()` значение 5? Теперь интервал между включенным и выключенным состояниями светодиода будет всего лишь 5 мс. Можете ли вы видеть переход между этими состояниями? При какой самой меньшей задержке вы можете видеть мигание светодиода?

## Наш первый аппаратный компонент

Итак, светодиод, встроенный в плату Arduino, мы запустили. На следующем шаге мы добавим в схему первый внешний компонент, которым в данном случае также станет светодиод. Как уже упоминалось, контактные выводы Arduino используются для ввода и вывода сигналов с микроконтроллера. Продемонстрировать вывод сигналов с микроконтроллера можно с помощью простого светодиода. Возьмите светодиод и внимательно рассмотрите его. Он должен выглядеть примерно так, как показано на рис. 1.20.

Как можно видеть, выводы светодиода разной длины — один короче другого. Также при

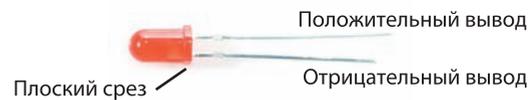


Рис. 1.20. Пример светодиода: длинный вывод — положительный, короткий — отрицательный

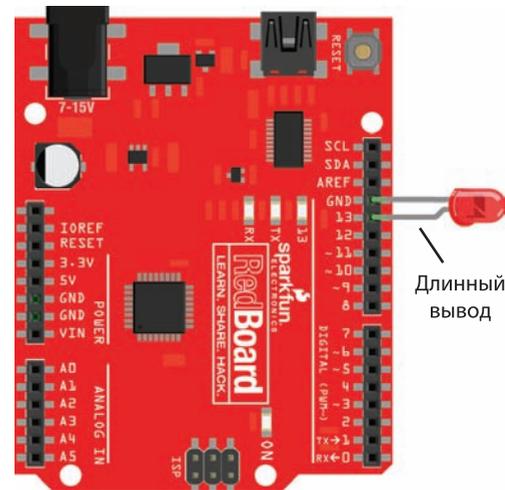


Рис. 1.21. Подключение внешнего светодиода к выводу 13 платы Arduino

внимательном осмотре можно заметить, что фанка в основании светодиода срезана со стороны короткого вывода. Так указывается полярность выводов светодиода: длинный вывод является положительным (анодом), а короткий — отрицательным (катодом).

Вспомните, что константа `LED_BUILTIN` имеет значение 13, то есть обозначает вывод 13 на плате Arduino. Соответственно, чтобы подключить внешний светодиод к Arduino, нужно всего лишь вставить его длинный вывод в гнездо 13 Arduino, а короткий — в гнездо вывода «земли», которое на плате обозначено меткой **GND**<sup>2</sup> и расположено рядом с гнездом вывода 13. Светодиод можно вставить и при включенном питании платы. При правильном подключении (рис. 1.21) он сразу же должен начать мигать. Если светодиод не мигает, скорее всего вы перепутали выводы. Ничего страшного, просто извлеките его и снова вставьте, поменяв выводы местами.

<sup>2</sup> Сокращение от англ. Ground — земля.

## Идем дальше...

Каждый проект этой книги завершается разделом с таким названием, в котором рассматриваются способы дальнейшего применения знаний, полученных в проекте. В этих разделах содержатся советы по использованию проекта в его текущем состоянии, экспериментированию с кодом, а также по модифицированию схемы проекта.

### Экспериментируем с кодом

Попробуйте сделать так, чтобы светодиод мигал какими-либо интересными последовательностями. Сначала скопируйте четыре строки кода функции `loop()` из листинга 1.1 и вставьте их после последней инструкции. В результате у вас будут две последовательности кода для мигания светодиодом, что даст больше возможностей для экспериментов. Так можно создавать последовательности мигания светодиода, изменяя время задержки как для включенного состояния светодиода, так и для выключенного. Например, последовательность, имитирующую биение сердца. Модифицированный соответствующим образом код исходного скетча `Blink` приводится в листинге 1.4.

#### Листинг 1.4. Код для последовательности мигания светодиода, имитирующей биение сердца

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(200);
  digitalWrite(LED_BUILTIN, LOW);
  delay(200);
  digitalWrite(LED_BUILTIN, HIGH);
  delay(200);
  digitalWrite(LED_BUILTIN, LOW);
  delay(800);
}
```

Усложним задачу — попробуйте создать скетч `Arduino` для мигания букв своего имени с помощью сигналов азбуки Морзе, то есть последовательностями коротких (точка) и длинных (тире) миганий. Чтобы помочь вам с этой задачей, на рис. 1.22 приводится код Морзе для букв английского алфавита. Для начала попробуйте создать код для мигания «S-O-S»: точка, точка, точка — тире, тире, тире — точка, точка, точка.

|         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|
| A ···—  | H ····· | O ———   | V ····· | 1 ———·  | 6 ····· |
| B ····· | I ···   | P ····· | W ———·  | 2 ———·  | 7 ····· |
| C ···—  | J ———·  | Q ···—  | X ····· | 3 ———·  | 8 ———   |
| D ···—  | K ···—  | R ····  | Y ———·  | 4 ———·  | 9 ····· |
| E ···   | L ····· | S ···   | Z ···—  | 5 ····· | 0 ———   |
| F ····· | M ———   | T ———   |         |         |         |
| G ———   | N ···   | U —··   |         |         |         |

Рис. 1.22. Код Морзе для букв английского алфавита и цифр

### Модифицируем схему

Мигающий светодиод дает нам целую кучу возможностей. Например, можно оснастить мигающими светодиодами какие-либо предметы у себя дома. В частности, хорошей идеей было бы украсить костюм для Хеллоуина каким-нибудь страшным существом с мигающими красными глазами. Для этого может потребоваться удлинить выводы светодиода, чтобы можно было спрятать плату `Arduino` где-либо в одежде (например, в кармане). Мы купили себе паука для Хеллоуина и встроили в него наводящие ужас мигающие красные глаза (рис. 1.23).



Рис. 1.23. Наводящий ужас паук с мигающими красными глазами

Управляемые мигающие светодиоды также хорошо подойдут и для масштабного моделирования. Использование светодиодов для фар автомобиля, освещения домов или уличных фонарей представляется прекрасным способом создать иллюзию реальности для любой масштабной модели или ландшафта (рис. 1.24).



**Рис. 1.24.** Масштабная модель автомобиля с фарами, управляемыми Arduino

## Сохранение скетча

Кроме экспериментирования с исходным скетчем Blink, возможно, вы захотите продолжить экспериментировать и с его модификациями. Для этого сохраните их, присвоив им описательные имена, которые напомнят вам, что именно делает конкретный скетч. Имя файла скетча не должно содержать никаких пробелов — если вставить пробел, Arduino заменит его на символ подчеркивания: «\_». По умолчанию среда разработки Arduino сохраняет скетч в папке Arduino, которая

находится в папке Мои документы. Скетчи можно сохранять в любой другой папке, но гораздо удобнее, когда они все хранятся в одном месте.

Если вы уверены, что полностью освоили мигающий светодиодом скетч Blink и чувствуете теперь, что готовы расширить свои знания, переходите к *проекту 2*. В нем мы покажем вам, как создать светофор под управлением Arduino.