

Дж. Бейктал

КОНСТРУИРУЕМ РОБОТОВ

на **Arduino**®

Первые шаги



Конструируем роботов на Arduino

Первые шаги



ARDUINO FOR BEGINNERS

ESSENTIAL SKILLS EVERY MAKER NEEDS

John Baichtal

que[®]

800 East 96th Street,
Indianapolis, Indiana 46240 USA

Дж. Бейктал

КОНСТРУИРУЕМ РОБОТОВ

на **Arduino**

Первые шаги

Перевод с английского
О. А. Трефиловой

3-е издание, электронное



Москва
Лаборатория знаний
2020

УДК 621.86/.87
ББК 32.816
Б41

Серия основана в 2016 г.

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

Бейктал Дж.

Б41 Конструируем роботов на Arduino®. Первые шаги / Дж. Бейктал ; пер. с англ. О. А. Трефиловой. — 3-е изд., электрон. — М. : Лаборатория знаний, 2020. — 323 с. — (РОБОФИШКИ). — Систем. требования: Adobe Reader XI ; экран 10". — Загл. с титул. экрана. — Текст : электронный.

ISBN 978-5-00101-900-8

Это практическое руководство для тех, кто делает первые шаги в робототехнике на платформе Arduino. С этой книгой вы разберетесь в основах электроники, научитесь программировать в среде Arduino IDE, работать с печатными платами Arduino, инструментами, соблюдать правила безопасности и многому другому. Вы также сможете выполнить разнообразные проекты и оценить невероятный потенциал Arduino, который вдохновит вас на творчество и изобретения, ограниченные только вашим воображением.

Для молодых изобретателей и программистов, а также всех тех, кто увлекается робототехникой.

**УДК 621.86/.87
ББК 32.816**

Деривативное издание на основе печатного аналога: Конструируем роботов на Arduino®. Первые шаги / Дж. Бейктал ; пер. с англ. О. А. Трефиловой. — 2-е изд. — М. : Лаборатория знаний, 2019. — 320 с. : ил. — (РОБОФИШКИ). — ISBN 978-5-00101-095-1.

16+

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

ISBN 978-5-00101-900-8

Authorized translation from the English language edition, entitled ARDUINO FOR BEGINNERS: ESSENTIAL SKILLS EVERY MAKER NEEDS; by JOHN BAICHTAL; published by Pearson Education, Inc, publishing as QUE Publishing. Copyright © 2014 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Авторизованный перевод англоязычного издания, под заглавием ARDUINO FOR BEGINNERS: ESSENTIAL SKILLS EVERY MAKER NEEDS; автор JOHN BAICHTAL; опубликованного Pearson Education, Inc, осуществляющим издательскую деятельность под торговой маркой QUE Publishing.

Copyright © 2014 by Pearson Education, Inc.

Все права защищены. Воспроизведение или распространение какой-либо части/частей данной книги в какой-либо форме, какими-либо способами, электронными или механическими, включая фотокопирование, запись и любые поисковые системы хранения информации, без разрешения Pearson Education, Inc запрещены.

© Перевод на русский язык, оформление, Лаборатория знаний, 2016

Оглавление

Об авторе	8
Посвящение	8
Благодарности.....	8
Здравствуйте!	9
Введение	10
О чем эта книга	11
Для кого эта книга	12
Как пользоваться книгой	12
Глава 1. Знакомство с Arduino	15
Arduino UNO: микроконтроллер для начинающих	15
Другие продукты Arduino.....	18
Электроника	19
Правила техники безопасности	29
В следующей главе	30
Глава 2. Макетирование	31
Сборка электрических схем с использованием макетных плат с гнездами, не требующих пайки	31
❁ Проект: мигающий светодиод на макетной плате	34
❁ Проект: лазерная сигнализация	37
❁ Проект: инфракрасный детектор	49
В следующей главе	52
Глава 3. Работа с паяльником	53
Паяльные принадлежности	55
Паяние	62
Распайка	65
Уборка	67
❁ Проект: кофейный столик со светодиодной лентой	68
В следующей главе	84

Глава 4. Настройка беспроводного соединения	85
Радиомодули XBee	86
Переходные платы для радиомодуля XBee	88
Компоненты радиомодуля XBee	88
Альтернативные беспроводные модули	90
⚙ Проект: беспроводное включение светодиода	91
⚙ Проект: беспроводной дверной звонок	95
В следующей главе	108
Глава 5. Программирование Arduino	109
Среда разработки Arduino	110
Скетч «Blink»	118
Учимся на примере кода	121
Функции и синтаксис	126
Отладка с помощью монитора последовательного интерфейса	131
Все о библиотеках	133
Ресурсы для изучения программирования	135
В следующей главе	136
Глава 6. Восприятие мира	137
Урок: датчики (сенсоры)	138
Знакомство с датчиками	140
⚙ Проект: «Лампа настроения»	146
⚙ Проект: керфбэндинг	154
В следующей главе	154
Глава 7. Управление жидкостью	155
Урок: управление потоком жидкости	156
⚙ Проект: емкость под давлением	159
⚙ Проект: робот для полива растений	162
В следующей главе	174
Глава 8. Ящик для инструментов	175
Набор инструментов для начинающего мастера	176
Работа с деревом	183
Работа с пластиком	192
Работа с металлом	198
Программное обеспечение	208
Электронная техника и инструменты	213
В следующей главе	222

Глава 9. Ультразвуковая эхолокация	223
Урок: ультразвуковая диагностика.....	224
❁ Проект: ультразвуковой ночник.....	226
❁ Проект: игрушка для кошки.....	228
Токарный станок 101.....	242
Техника безопасности при работе с токарным станком.....	244
В следующей главе.....	244
Глава 10. Генерация звука	245
Звуки электроники.....	246
❁ Проект: мелодичная кнопка.....	250
❁ Проект: звуковой генератор.....	253
В следующей главе.....	262
Глава 11. Отсчет времени	263
Сервер точного времени.....	264
Таймер Arduino.....	265
Модуль часов реального времени (RTC).....	266
❁ Проект: цифровые часы.....	266
❁ Проект: китайские колокольчики «Музыка ветра».....	270
Станки с числовым программным управлением (ЧПУ).....	280
В следующей главе.....	281
Глава 12. Безопасная работа с высоким напряжением	283
Урок: управление высоким напряжением.....	284
❁ Проект: контроллер для вентилятора.....	288
❁ Проект: лавовая лампа «Бадди».....	291
В следующей главе.....	298
Глава 13. Управление электродвигателями	299
Как управлять двигателями.....	300
Включаем двигатель с помощью T1P-120.....	303
❁ Проект: шаговый поворотник.....	305
❁ Проект: «Баблбот».....	308
Глоссарий	317

Об авторе

Джон Бейктал начал свою литературную деятельность с написания постов для легендарного блога «GreekDad» ежемесячного журнала «Wired» (Сан-Франциско, США), а также статей для журнала «MAKE Magazine», который поклонники субкультуры DIY (от англ. *Do It Yourself* — «сделай это сам») считают своей библией. Благодаря этому Джон стал писать книги об электронных игрушках, инструментах и другой любительской электронике. Он является соавтором книги «The Cult of LEGO» (издательство «No Starch Press») и автором книг «Hack This: 24 incredible Projects from DIY movement» и «Basic Robot Building with LEGO Mindstorm's NXT 2.0» (обе книги вышли в издательстве «Que»). Позже он написал цикл статей «Make LEGO and Arduino Projects» для журнала «MAKE Magazine» в сотрудничестве с Адамом Вольфом и Мэтью Бекером. Джон Бейктал живет в Миннеаполисе, штат Миннесота, США. У него есть жена и трое детей.

Посвящение

Посвящается Гарольду Бейкталу (1939–2013)

Благодарности

Я хочу поблагодарить мою любимую жену Элизу за ее терпение и поддержку; всех моих друзей-электронщиков за ответы на мои бесконечные вопросы; мою маму, Барбару, за работу над глоссарием; и моих детей, Айлин Арден, Розмари и Джека, за их любознательность и интерес.

Здравствуйте!

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

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

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

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

И тогда, уверенно освоив пространство Arduino, вы сделаете очередной шаг по этой дороге.

Успехов вам!

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

Введение

Выбирая в магазине электронной техники нужное вам устройство, вы никогда не задумывались над тем, что могли бы сделать его сами и получить от этого удовольствие?

Такая перспектива может напугать, ведь нужно изучить основы электроники, узнать, как работает каждая деталь и как управлять устройством с помощью микроконтроллера. Чтобы соединить все компоненты, придется научиться паять, а чтобы запрограммировать микроконтроллер, нужно освоить кодирование. Чтобы собрать электронную аппаратуру, нужно обладать определенными навыками работы с инструментами.

Звучит устрашающе?

Технология, которая позволяет не только осуществить эти мечты, но и сделать это с удовольствием, называется «Arduino». Это небольшая плата с микроконтроллером, простая в освоении и легкая в программировании. Она управляет электродвигателями и принимает сигналы от датчиков.

Цель данной книги — помочь научиться работать с инструментами, познакомиться с их многообразием, изучить основы программирования и конструирования. Все проекты разработаны с использованием микроконтроллера Arduino UNO.

Работа над книгой расширила мой кругозор. Надеюсь, и вы получите ценный опыт. С помощью Arduino вы сможете создать множество интересных вещей, и проекты, представленные в этой книге, — лишь только начало. Желаю удачи и приятного времяпрепровождения!

О чем эта книга

Эта книга написана с целью превратить простого обывателя в человека, увлеченного электроникой, моделированием и программированием.

- В **главе 1** «Знакомство с Arduino» вы найдете информацию об Arduino и электронике — именно то, что нужно для начала работы!
- В **главе 2** «Макетирование» в рамках проекта по электронике вы сделаете лазерную сигнализацию с помощью макетной платы, не требующей пайки.
- В **главе 3** «Работа с паяльником» вы научитесь соединять электронные детали с помощью паяльника. В рамках проекта вы сможете украсить кофейный столик светодиодной лентой.
- В **главе 4** «Настройка беспроводного соединения» вы узнаете о трех различных способах управления проектом с помощью беспроводных сигналов. На основе изученного материала вы сможете сделать беспроводной дверной звонок.
- В **главе 5** «Программирование Arduino» изложены основы управления Arduino с помощью программ, загруженных на плату. Я проведу подробный обзор программы Arduino, чтобы вы смогли изучить принцип работы с ней.
- В **главе 6** «Восприятие мира» вы познакомитесь с разнообразием датчиков и узнаете разницу между цифровыми и аналоговыми датчиками. Вы сделаете «Лампу настроения», которая меняет свой цвет в зависимости от уровня освещенности.
- В **главе 7** «Управление жидкостью» вы узнаете о трех способах перекачивания жидкости, один из которых ляжет в основу создания робота для полива растений.
- В **главе 8** «Ящик для инструментов» представлен широкий обзор инструментов, начиная с самых простых и привычных и заканчивая специальными инструментами для работ по дереву или металлу.
- В **главе 9** «Ультразвуковая эхолокация» вы узнаете, как с помощью неслышных для человеческого уха импульсов можно обнаруживать препятствия и измерять расстояния. Глава включает проект по созданию игрушки для кошки.
- В **главе 10** «Генерация звука» вы узнаете, как с помощью Arduino сочинить электронную музыку (также известную как «нойз»¹), и сделаете портативный звуковой генератор.
- В **главе 11** «Отсчет времени» вы познакомитесь с тремя способами, которые использует Arduino, чтобы следить за временем, а также сконструируете колокольчики «Музыка ветра», которые звонят в определенный час.

¹ Музыкальный жанр, в котором используются разнообразные звуки (чаще всего искусственного и техногенного происхождения). — *Прим. перев.*

- В **главе 12** «Безопасная работа с высоким напряжением» вы узнаете о трех способах безопасной работы с комнатной проводкой. Вы сделаете контроллер для лавовой лампы, с помощью которого лампа будет включаться и выключаться либо по расписанию, либо дистанционно.
- В **главе 13** «Управление электродвигателями» вы узнаете о способах управления электродвигателями и затем на основе полученных навыков сконструируете робота, пускающего мыльные пузыри.

Для кого эта книга

Эта книга написана для молодых людей, которые только начинают работать с Arduino, поэтому пока не обладают обширными познаниями. От них требуются лишь любопытство и желание трудиться.

Как пользоваться книгой

Книга написана доступным языком, и вам не понадобятся специальные инструкции для работы с ней. Тем не менее, поясню некоторые используемые вставки в тексте.

СОВЕТ

Это полезные рекомендации, которые помогут сэкономить время.

ДЛЯ СПРАВКИ

Это полезная, но не особо важная информация.

ТЕХНИКА БЕЗОПАСНОСТИ

Это предупреждение об опасности, грозящей при работе с оборудованием и инструментами.

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для каждого проекта в книге я привожу список деталей и приспособлений, которые понадобятся при конструировании.

- Arduino.
- Сервопривод.
- Сервонасадки (существует огромное количество насадок на ваш выбор).
- Модуль RTC «ChronoDot».
- Круглый стержень диаметром 6 мм и длиной около 20 см.
- Китайские колокольчики «Музыка ветра» (я использовал модель «Gregorian Chimes Soprano»).
- Фанера толщиной 5 мм для корпуса.
- Сосновые бруски для опорных блоков (25 мм).
- Крюк и гайка к нему.
- Саморезы для дерева № 8 × 12 мм.
- Саморезы для дерева № 6 × 50 мм.
- Саморезы для дерева № 4 × 12 мм.
- Болты 6 × 12 мм с фиксирующими шайбами и гайками, 24 шт.
- Болты 6 × 25 мм с фиксирующими шайбами и гайками, 12 шт.
- Дрель и разные сверла.
- Отрезная дисковая пила.
- Циркулярная пила.

КОД

Если для выполнения проекта понадобится код (скетч), то в книге будет представлен соответствующий пример. Однако вам не нужно переписывать код из книги. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

Перейдя по ссылке <https://github.com/n1/Arduino-For-Beginners>, вы сможете загрузить этот код и другие полезные файлы.

Пример кода:

```
int valve = 13;    // присваивает порту 13 значение «valve» (клапан)

int offhours = 0; // сколько часов до полива?
int offmins = 1;  // сколько минут до полива?
int spray = 10;   // длительность полива в секундах
```

```
void setup() {
  pinMode(valve, OUTPUT); // устанавливает порт valve в режим
                          // выхода (клапан открыт)
  Serial.begin(115200);
}

void loop() {
  int wait = (offmins * 60000) + (offhours * 3600000);
  // вычисляет миллисекунды

  digitalWrite(valve, HIGH);
  delay(spray * 1000); // длительность полива в миллисекундах
  Serial.println(offmins * 60000); // я использовал это при отладке
  digitalWrite(valve, LOW);
  delay(offmins * 60000); // прекращение полива на это количество
                          // миллисекунд
}
```

Знакомство с Arduino

Что необходимо знать для успешного создания проектов, описанных в этой книге? Оказывается, многое. Хорошая новость заключается в том, что эта глава подготовит вас к открытиям. Здесь изложены основы электроники, советы по соблюдению правил техники безопасности, а также представлен обзор Arduino UNO. Давайте начнем!

Arduino UNO: микроконтроллер для начинающих

Представьте компьютер, размещенный на печатной плате размером меньше игровой карты. Согласитесь, было бы замечательно, если можно было бы добавить несколько датчиков для изучения окружающей среды, кнопки для подачи команд и электродвигатели, чтобы привести объект в движение. На самом деле это возможно. Устройство, которое я только что описал, — это Arduino, платформа на базе микроконтроллера.

На **рис. 1.1** изображен пример того, что можно сделать с ее помощью. Это шестиногий робот *Hexy* от компании *ArcBotics*, который базируется на платформе Arduino. Робот поставляется с 20 сервоприводами и движется посредством инверсной кинематики — концепции робототехники, которая упрощает движение посредством использования заданных команд, как, например, «двигаться вперед». Несомненно, такая продвинутая модель, как шестиногий робот *Hexy*, совершенно отличается от устройства с мигающей лампочкой!

Несмотря на достоинства Arduino, его нельзя назвать первым микроконтроллером для любителей. У Arduino были предшественники, но ни одному из них не удалось достичь такого успеха. Ни одна из плат компаний-конкурентов не была так же проста в использовании. В действительности, плата Arduino была разработана специально для художников, студентов и обычных умельцев, которые хотели просто освоить механизм, особенно не углубляясь в детали.



Источник: ArcBotics

Рис. 1.1. Шестиногий робот *Hexy* компании *ArcBotics* демонстрирует возможности Arduino

Платформа Arduino совершенствуется: для нее разрабатывается множество проектов, сайтов и аппаратных средств. Такое разнообразие ресурсов сделало платформу более реальной, что привело к увеличению количества участников и проектов.

СОВЕТ МЫ ИСПОЛЬЗУЕМ UNO

Проекты, описанные в этой книге, основаны исключительно на использовании платформы Arduino UNO. Несмотря на то что существуют другие варианты и версии платформ в линейке Arduino, UNO является платой по умолчанию, поэтому в этой книге внимание сосредоточено именно на ней. Некоторые версии платформы Arduino больше по размеру и обладают большими возможностями, некоторые меньше и могут разбираться. UNO не только считается золотой серединой, но и оценивается большинством как базовая модель. В **главе 8** «Ящик для инструментов» вы найдете описание других моделей.

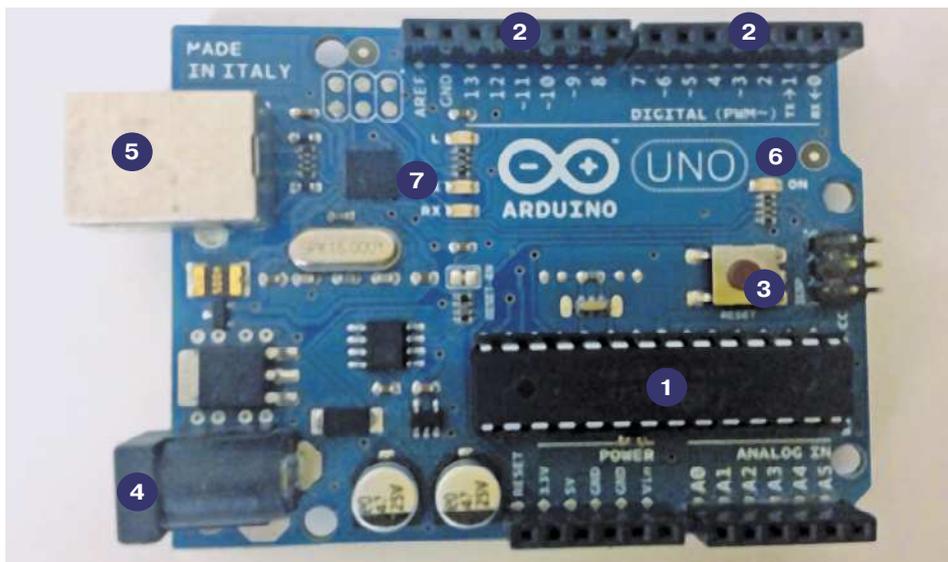


Рис. 1.2. Arduino, имея размер игровой карты, состоит из следующих блоков: 1 — микроконтроллер ATmega328; 2 — пины; 3 — кнопка сброса; 4 — разъем питания; 5 — USB-разъем; 6 — индикатор питания; 7 — индикаторы данных

Что именно вы получаете при покупке Arduino? Проведем небольшой обзор платы и ознакомимся с ее особенностями (см. **рис. 1.2**).

Arduino UNO состоит из печатной платы с микроконтроллером и других различных компонентов, прикрепленных к ней. На **рис. 1.2** изображены основные компоненты.

- **Микроконтроллер ATmega328** — «мозг» платы. ATmega328 имеет флеш-память объемом 23 кб, оперативное запоминающее устройство 2 кб и тактовую частоту 16 МГц. Такие характеристики не впечатляют, но для программ Arduino этого достаточно.
- **Пины.** К этим небольшим портам присоединяются провода. Например, можно подключить кнопку или электродвигатель. Функции пинов различаются; позже мы рассмотрим эти различия.
- **Кнопка сброса.** Если система не работает, нажмите на эту кнопку. Произойдет перезагрузка Arduino и автоматический перезапуск программы, загруженной на платформу.
- **Разъем питания.** Он соответствует сетевому адаптеру 9 В диаметром 2,1 мм с центрально-положительным пином. Вы также можете присоединить батарейку напряжением 9 В с соответствующим штекером. В **главе 8** мы рассмотрим различные способы подключения питания к вашим проектам Arduino.

- **USB-разъем.** Через этот разъем с помощью USB-кабеля, который часто используется для подключения принтера и других периферийных устройств, подается питание и осуществляется передача данных. Таким образом, вы можете создать проект без использования батареек.
- **Индикатор питания.** Этот индикатор загорается, когда плата получает питание.
- **Индикаторы данных.** Эти светодиоды мигают, когда данные загружаются на плату.

СОВЕТ

ЗАГРУЗКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ARDUINO

Из этой главы вы не узнаете, как запрограммировать Arduino, но вы можете научиться загружать программное обеспечение. Оно совместимо с операционными системами Windows, Macintosh и Linux и абсолютно бесплатно. Перейдите по ссылке <http://arduino.cc/en/Main/Software> и следуйте указаниям.

Более подробно о процессе загрузки рассказано в **главе 5** «Программирование Arduino».

Другие продукты Arduino

Как уже было упомянуто, существует целая экосистема версий платформы Arduino, а также дополнительные платы расширения (shields¹). Версии Arduino весьма разнообразны и включают как более мощные платформы для выполнения крупных проектов, так и небольшие для малых проектов, а также дополнительные платы расширения, которые позволяют, например, воспроизводить музыку и устанавливать интернет-соединение для определения координат посредством GPS.

Зачастую бывает, что проект, который вы обдумываете, уже кто-то реализовал и создал соответствующую плату расширения. Если вы хотите добавить в ваш проект определенные возможности, то сначала узнайте, нет ли соответствующей платы — не стоит загружать себя лишней работой. Еще лучше, когда платы можно скомпоновать стопкой и постепенно выстроить более сложные конструкции.

¹ Среди людей, работающих с платформой Arduino, также распространен вариант перевода «шилд», однако такое калькирование уместно скорее в разговорной речи, более правильный перевод — «плата расширения». — *Прим. перев.*

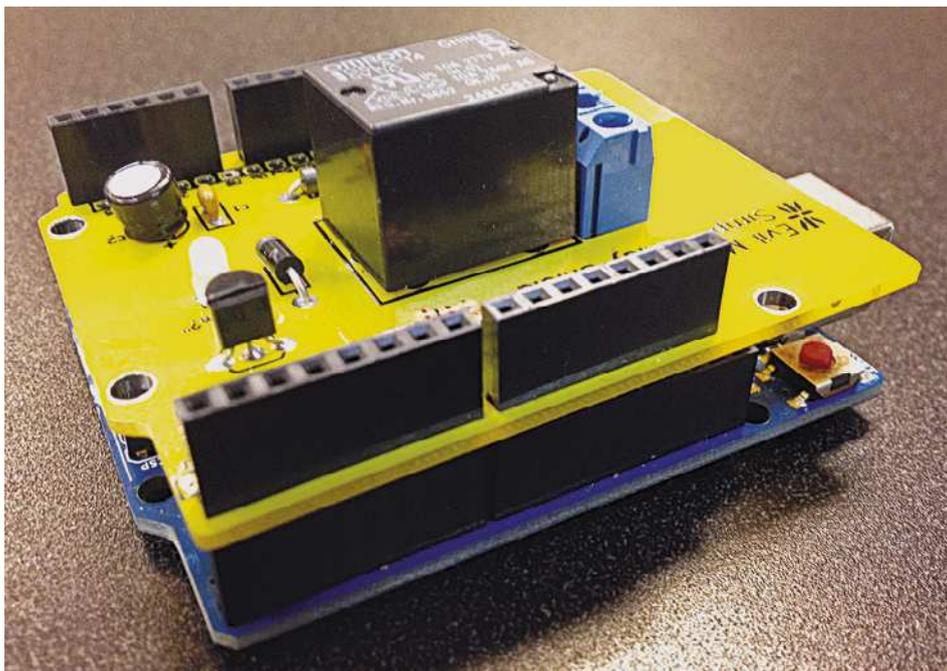


Рис. 1.3. Плата расширения с реле от *Evil Mad Science* вставляется в разъемы Arduino и позволяет управлять цепью высокого напряжения

Примером платы, размещенной поверх Arduino, служит плата расширения с реле (*relay shield*) (см. **рис. 1.3**). Плата, разработанная умельцами из *Evil Mad Science LLC*, использует реле, чтобы управлять цепью высокого напряжения. Например, ее можно использовать, чтобы безопасно включать лампу, подключенную к комнатной проводке.

Хотите узнать больше о платах расширения? В **главе 8** мы более подробно рассмотрим их и другие дополнительные платы.

Электроника

Arduino — это здорово, но чтобы создать что-то интересное, понадобятся дополнительные электронные компоненты. В проекте, представленном на **рис. 1.4**, для изготовления шоколадного молока используются электродвигатели LEGO и аквариумная помпа под управлением Arduino. В **главе 7** «Управление жидкостью» я покажу, как сделать простой насос. Далее представлен краткий обзор наиболее распространенных компонентов.

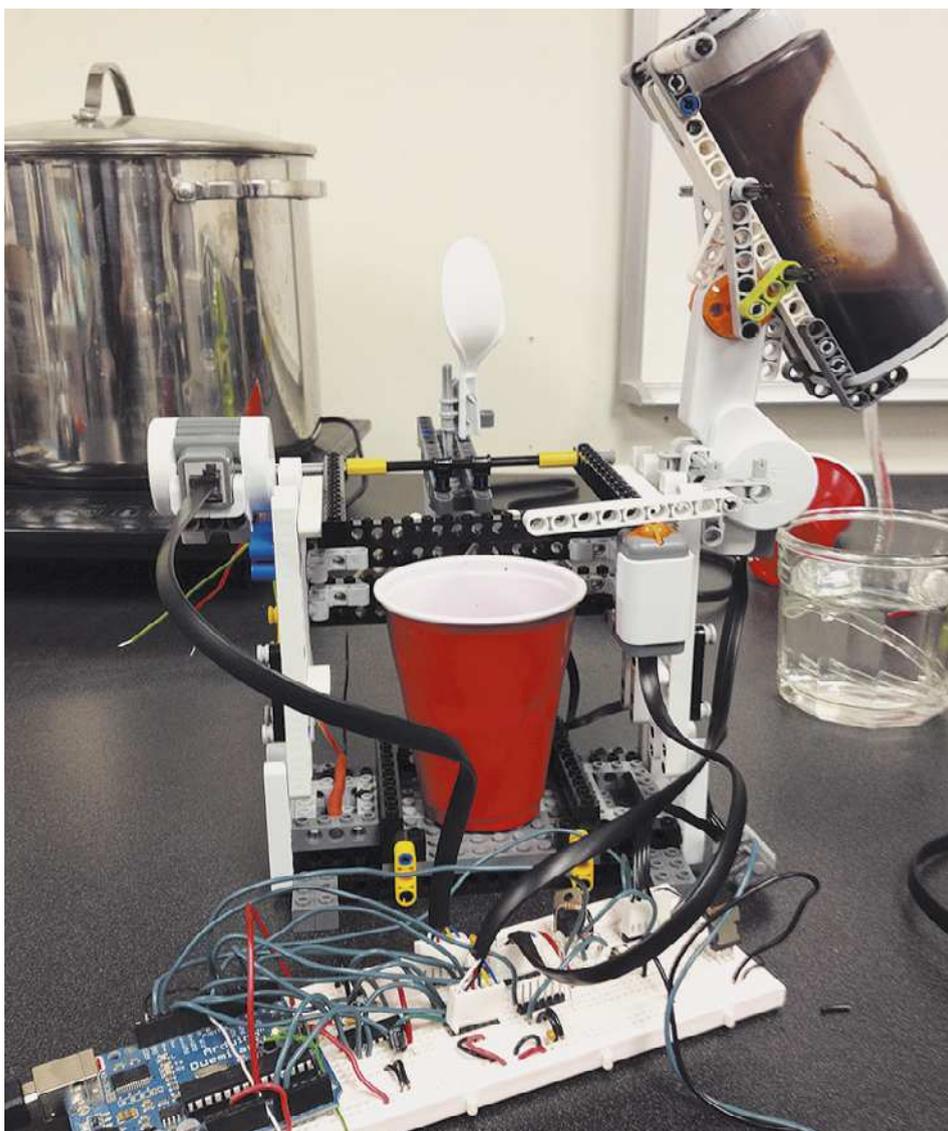


Рис 1.4. Для создания интересного проекта понадобится не только Arduino

ДЛЯ СПРАВКИ ЭТО ТОЛЬКО ОБЗОР

Существует большое разнообразие электронных компонентов. О некоторых я расскажу в книге позже, некоторые вам придется изучить самостоятельно. Цель, которую я преследую в следующих разделах, — познакомить вас с основными из них.

Светодиоды

Светодиоды (см. **рис. 1.5**) — это индикаторы в мире Arduino. Они бывают различных цветов и светят с разной степенью интенсивности. Некоторые обладают такими дополнительными особенностями, как мерцание или способность менять цвет в зависимости от входных данных программного обеспечения. Светодиоды, которые могут менять цвет, называются **RGB-светодиодами** (*red-green-blue*). Они понадобятся вам позже.



Рис. 1.5. Светодиоды — лампочки в мире электроники

Кнопки и переключатели

Человек задает команды Arduino посредством кнопок и переключателей (**рис. 1.6**). Благодаря их великому множеству, можно подобрать правильную конфигурацию проекта, над которым вы работаете. С помощью переключателей можно сделать всевозможные интересные вещи. Например, поочередно использовать две подпрограммы.



Рис. 1.6. С помощью кнопок и переключателей можно задавать Arduino команды

Потенциометры

С помощью этих элементов можно менять сопротивление в цепи, а значит и ток. Например, яркость светодиода можно сделать более интенсивной или менее интенсивной. С помощью потенциометров также можно управлять данными, если запрограммировать порядок выполнения действий в зависимости от положения регулятора. На **рис. 1.7** представлены различные технические варианты исполнения потенциометров.



Рис. 1.7. Потенциометры позволяют управлять параметрами электрической цепи путем вращения регулятора

Резисторы

Нельзя не согласиться с тем, что электричество — друг электронных компонентов. Однако слишком большое количество электрического тока может привести к их повреждению. Вот для чего нужны резисторы. Эти небольшие элементы ограничивают величину электрического тока. Единица измерения сопротивления — Ом. На **рис. 1.8** изображены резисторы, которые часто используются в конструкторских проектах. Их маркируют с помощью цветных полосок, таким образом можно определить величину сопротивления резистора. Подробнее о цветовой маркировке можно узнать в **главе 8**.

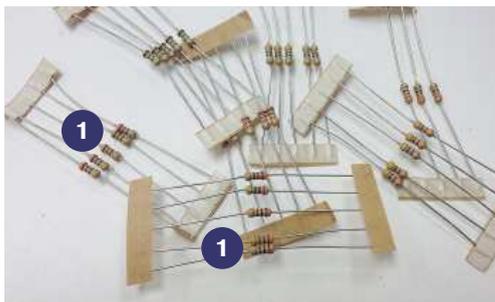


Рис. 1.8. Резисторы ограничивают величину электрического тока:

1 — цветные полосы обозначают величину сопротивления резистора в Омах

Конденсаторы

Конденсаторы — это накопители электрического заряда, поэтому их используют в качестве регулятора времени, потому что в паре с резисторами конденсатор разряжается с предсказуемой частотой. Из-за этой предсказуемости конденсаторы также используются для того, чтобы «улучшить» электронный сигнал. На **рис. 1.9** представлены различные варианты исполнения конденсаторов.

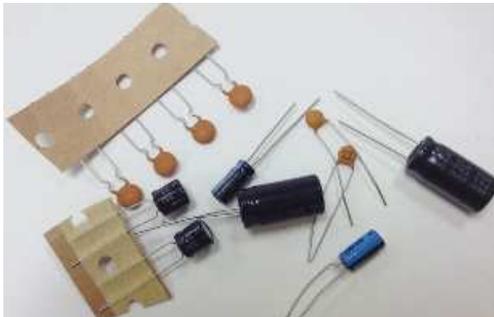


Рис. 1.9. Конденсаторы накапливают и отдают небольшое количество электрического заряда

Электродвигатели

На **рис. 1.10** представлены три основных вида электродвигателей, которые вам предстоит изучить.

- **Шаговые электродвигатели.** Вместо совершения хаотичного вращения шаговый электродвигатель вращается «шажками». Это позволяет управлять параметрами движения, поэтому данный тип двигателей применяется при автоматизированных фрезеровочных работах и при выполнении других задач, которые требуют постоянного контроля.



Рис. 1.10. Шаговые двигатели, сервоприводы и электродвигатели постоянного тока — основные виды электродвигателей в электронике:

1 — шаговые двигатели; 2 — сервоприводы; 3 — электродвигатели постоянного тока

- **Сервоприводы** — это электродвигатели со встроенными «энкодерами», которые отправляют информацию о местонахождении обратно на микроконтроллер. Сервоприводы зачастую используются в роботах, где управление электродвигателем играет второстепенную роль.
- **Электродвигатель постоянного тока** нельзя контролировать, так как нет обратной связи с микроконтроллером. Если ток есть, двигатель вращается. Если тока нет, то двигатель останавливается. Электродвигатели постоянного тока используются в проектах, где положение вала не имеет значения, например, в вертолете с дистанционным управлением.

Соленоиды

В отличие от электродвигателя, соленоид (см. **рис. 1.11**) с помощью электромагнита обеспечивает возвратно-поступательное движение сердечника. Соленоиды приводят в движение клапаны в двигателях: когда ток проходит через обмотку соленоида, клапан открывается. В отсутствие тока клапан закрыт.



Рис. 1.11. Соленоиды внешне похожи на электродвигатели, но обеспечивают возвратно-поступательное движение вала

Пьезодинамики

Основным звукоизлучателем в электронных комплектах служит пьезоэлемент, представленный на **рис. 1.12**. Вы подаете ток, и появляется звук. Довольно просто!



Рис. 1.12. Хотите создать звук с помощью Arduino? Подключите один из этих пьезоэлементов

Семисегментный индикатор

Хотите включить в свой проект дисплей, который будет отображать буквы и цифры? Классический выбор — семисегментный индикатор, состоящий из нескольких светодиодов (обычно семи), которые можно включать в разных комбинациях, составляя букву или цифру. Существует множество различных вариантов исполнения (**рис. 1.13**), однако принцип работы устройств одинаков.

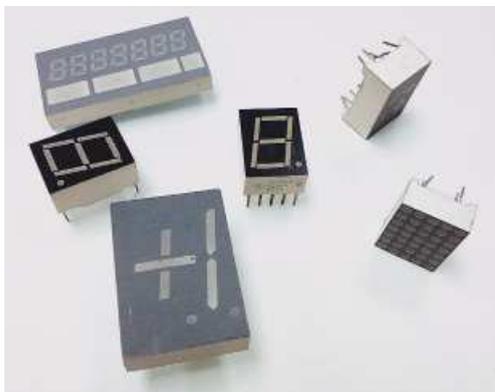


Рис. 1.13. Каждый индикатор состоит из нескольких сегментов-светодиодов, из которых можно составить буквы или цифры

Реле

Реле (см. **рис. 1.14**) похожи на электронные переключатели. Когда программа посылает на реле ток управления, другая схема активируется. Например, если вы хотите управлять лампой, подключенной к комнатной проводке без необходимости соприкоснуться напрямую с переменным током, можно использовать реле в паре с Arduino! На **рис. 1.3**, представленном ранее, изображена плата с расширением для Arduino, с помощью которой можно управлять реле.



Рис. 1.14. Реле — переключатели, управляемые с помощью Arduino, которые могут запускать схемы по команде

Интегральные схемы

Название «интегральная схема» (ИС) (см. **рис. 1.15**) говорит само за себя: это схема, заключенная в чип. Использование ИС упрощает ваши проекты — предполагается, что вы сможете найти правильную ИС. В качестве примера ИС можно привести микроконтроллеры, например, ATmega328, используемый в Arduino, таймеры, усилители.



Рис. 1.15. Интегральные схемы как совокупность электрических цепей умещаются в маленьком чипе

Датчики температуры

Датчик температуры (рис. 1.16) определяет температуру окружающей среды и передает показания Arduino. Этот датчик — прекрасное дополнение к проекту «Метеостанция», он также подходит для запуска вентилятора.



Рис. 1.16. Датчики температуры передают Arduino данные о состоянии окружающей среды

Датчик изгиба

Датчик изгиба (см. рис. 1.17) идеально подходит для мобильной электроники. В зависимости от изгиба меняется сопротивление датчика, так данные поступают на Arduino. Было бы здорово управлять рукой робота с помощью перчатки, оснащенной датчиком изгиба!



Рис. 1.17. Датчики изгиба реагируют на малейший изгиб. Роботизированную перчатку не желаете?

Датчик освещенности

Датчики освещенности (**рис. 1.18**) довольно часто используются в проектах по электронике. Действительно, в этой книге они будут задействованы несколько раз. По сути, датчик сообщает Arduino информацию о степени освещенности.



Рис. 1.18. Датчики передают Arduino информацию о степени освещенности

Ультразвуковой датчик

Ультразвуковые датчики (**рис. 1.19**) обнаруживают перемещения объектов в окружающем пространстве, посылая неслышные человеку радиопульсы в ожидании отраженного радиосигнала. Идея эхолокации заимствована у летучих мышей.



Рис. 1.19. Ультразвуковой датчик работает как радар

Правила техники безопасности

Эта книга предусматривает большой объем работы в мастерской, что подразумевает использование потенциально опасных инструментов. В этом разделе представлен обзор основных правил техники безопасности, которые действуют в большинстве ситуаций. В последующих главах будут рассмотрены более конкретные ситуации, а также соответствующие им правила техники безопасности. На **рис. 1.20** представлены два самых важных средства защиты — средства защиты органов зрения и слуха.

Вам необходимо соблюдать следующие правила.

- **Используйте средства защиты.** Защитные очки, средства защиты органов слуха, респираторы, защитная одежда часто необходимы, в зависимости от того, с каким инструментом вы работаете. Если вы имеете дело с раздражающими кожу веществами, то позаботьтесь о ее защите. Если у вас длинные волосы, то перед работой с электроинструментом соберите волосы в пучок, чтобы они не мешали. Всегда надевайте защитные очки, если существует риск того, что что-то может попасть в глаза, обычные очки не обеспечат надежную защиту.
- **Будьте бдительны.** Держитесь подальше от легковоспламеняемых веществ и предметов. Кроме того, перед работой нужно хорошо выспаться, у полуночников выше процент травматизма.



Рис. 1.20. Не стоит пренебрегать защитой органов зрения и слуха

- **Соблюдайте чистоту.** Не допускайте беспорядка на рабочем месте! Вероятность получить травму в захламленной мастерской гораздо выше, чем в убранном помещении.
- **Позаботьтесь об окружающих.** Не забывайте о тех, кто находится с вами в мастерской, на каком расстоянии они находятся от вас и от инструмента, с которым вы работаете. Например, если вы используете электропилу, а ваш друг рядом с грохотом роняет ключ, то это может привести к непредвиденным последствиям.
- **Изучите свой инструмент.** Нужно уважительно относиться к инструменту и не бояться его. Если вы держите в руках новый инструмент, то перед его использованием изучите инструкцию. Можете попросить более опытного пользователя проконсультировать вас. Если такого человека нет, то можно найти видео на *YouTube*, где демонстрируется принцип работы инструмента. Используйте инструмент только по прямому назначению. Например, многие люди получили травмы в результате того, что использовали отвертку в качестве гвоздодера.
- **Держите инструменты в исправном состоянии.** Если полотно пилы затупилось, то при распиливании придется приложить больше усилий, что увеличивает риск получения травмы.
- **Следите за своими пальцами.** В идеале, их десять, и все они вам нужны. Не забывайте об опасности при работе с мотопилой, сварочным аппаратом и даже обычным молотком, берегите пальцы.
- **Не забудьте об аптечке.** В дополнение к таким обычным средствам, как проспиртованный тампон, лейкопластырь и пинцет, не забудьте запастись марлевыми тампонами и бинтами, потому что травмы могут быть серьезными. Кроме того, солевой раствор для промывания глаз — хорошее средство для удаления раздражителей и даже опилок из глаз. В **главе 8** приводится полное описание аптечки первой помощи.
- **Не забудьте о средствах противопожарной безопасности: огнетушителях и датчиках дыма.** Не помешает и наличие раковины. В каждой мастерской должна быть раковина, чтобы смыть раздражители или промыть рану.

В следующей главе

В **главе 2** «Макетирование» вы узнаете, как создать электронные схемы, не требующие пайки, с использованием платы для прототипирования, которая называется макетной платой. Вы также узнаете, как сделать лазерную сигнализацию, чтобы защитить свой дом от непрошенных гостей.

Макетирование

В этой главе вы узнаете все о макетировании, быстром и легком способе создания прототипов проектов для Arduino. После того как материал будет изучен, вы возьметесь за свой первый проект: создание лазерной сигнализации для вашего дома! Также вы узнаете, как можно заменить лазер пассивным инфракрасным датчиком и как смоделировать и вырезать из фанеры корпус для лазерного модуля.

Сборка электрических схем с использованием макетных плат с гнездами, не требующих пайки

При создании проекта и новичок, и мастер первым делом изготавливают макет электронной схемы. Посмотрите на проект, представленный на **рис. 2.1**. На внесение изменений уйдет пара секунд. Идеально для создания прототипа!

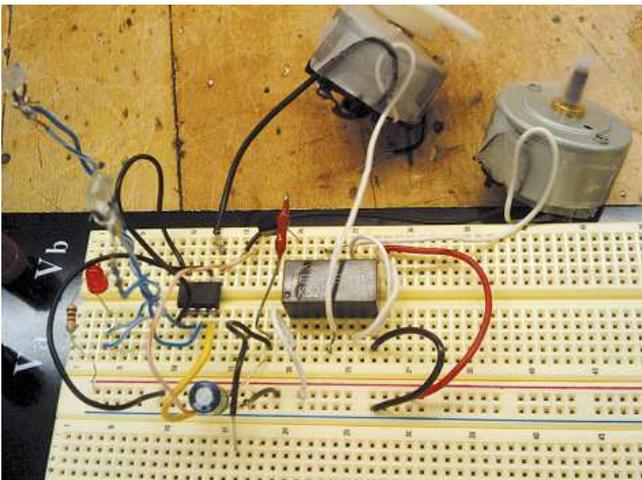


Рис. 2.1. Макетный проект легко создавать, он легко поддается изменению

Макетная плата, не требующая пайки, представляет собой плату, изготовленную из пластика, с контактными гнездами и скрытыми проводниками. По сути, эти проводники служат дополнительными кабелями в вашем проекте. Вы просто подключаете к плате Arduino электродвигатели, датчики, другие необходимые устройства и используете ее для управления соединением. На **рис. 2.2** представлена макетная плата.

Ниже описаны функции каждого соединения.

- А. **Заземляющая шина.** Служит для подключения Arduino. Заземляющие шины (GND) обычно обозначены синим или черным цветом.
- Б. **Шина питания.** Служит для подключения источника питания. Обратите внимание, что две шины не соединены друг с другом. Шины питания обозначены красным цветом.
- В. **Колодка с гнездами.** Колодка с рядами гнезд перпендикулярна базовым шинам и обозначена светло-голубым цветом. На вашей макетной плате такой маркировки не будет.
- Г. **Проводники.** Обозначены голубыми полосами.
- Д. **Координаты гнезд.** Это обозначение поможет при описании проекта. Например, фраза «подключите провод к гнезду H4» означает, что нужно найти четвертое по счету гнездо в ряду H и подсоединить к нему проводник.

Стандартные макетные платы состоят из пары шин, расположенных по краям: шины питания, обозначенной красным цветом, и шины заземления, обозначенной синим или черным цветом. Колодка с гнездами находится посередине, состоит из рядов с гнездами, которые соединены между собой скрытыми проводниками, обозначенными на **рис. 2.2** голубым цветом.

Проводники — это скрытые кабели, которые упрощают процесс подключения при выполнении проекта. Пучок подключенных проводов — ребус для новичка.



Рис. 2.2. Устройство стандартной макетной платы

Чтобы создать соединение между двумя компонентами (например, резистором и светодиодом), вставьте один конец провода в любое гнездо ряда со светодиодом, а второй конец провода — в гнездо ряда с резистором. Все достаточно просто!

И последнее, что вам нужно знать о макетных платах, не требующих пайки. У большинства плат на обратной стороне есть клейкий слой. С его помощью вы сможете прикрепить макетную плату к корпусу.

Питание и заземление

Не вдаваясь в подробности о том, как работает электричество, давайте рассмотрим два важных понятия, которые встречаются в проектах достаточно часто: «питание» и «заземление». В электрических схемах по проводу питания подается ток, а по проводу заземления он возвращается. На макетной плате шины питания и заземления разделены. Вы подключаете источник питания, будь то Arduino или вторичный источник, к шине питания, затем подключаете шину заземления к одному из пинов GND на Arduino.

Соединительные провода

Провода, которые в основном используются в проектах по макетированию, называются **соединительными** (*jumpers*). Такой провод можно с легкостью сделать самому, обрезав его до нужной длины. Лучше использовать одножильный провод; многожильный приходит в негодность быстрее.

Также можно купить специальные соединительные провода. Это тонкие провода разных цветов и длины с соединительными штырями на концах, как показано на **рис. 2.3**. Штыри соответствуют гнездам макетной платы, не требующей пайки, они достаточно износостойкие и долговечные.

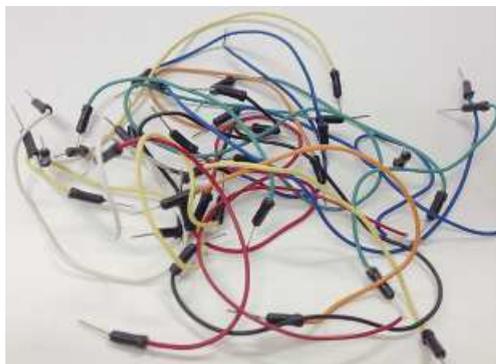


Рис. 2.3. Вы можете использовать практически любой провод для макетной платы, но предпочтительнее соединительные провода со штырьками

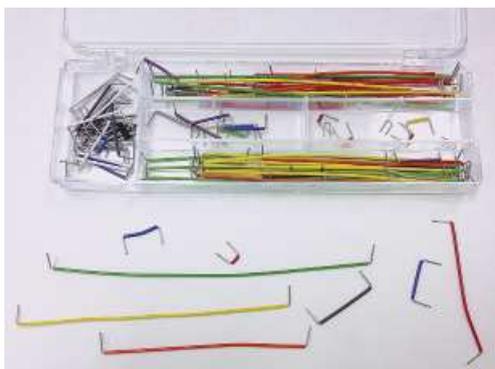


Рис. 2.4. Соединительные провода плотно прилегают к макетной плате, что обеспечивает порядок на рабочем месте

Другой тип соединительных проводов — нарезанные провода определенной длины, оголенные концы которых загнуты под углом таким образом, что их можно с легкостью вставить в гнезда макетной платы. Такие провода представлены на **рис. 2.4**.

Прежде чем взяться за основной проект, давайте научимся простейшему макетированию.

Проект: мигающий светодиод на макетной плате

Это простой проект, который можно выполнить за пару минут; он поможет вам разобраться в устройстве макетных плат. Цель проекта — подать питание на светодиод (**рис. 2.5**). Звучит просто? Так и есть.

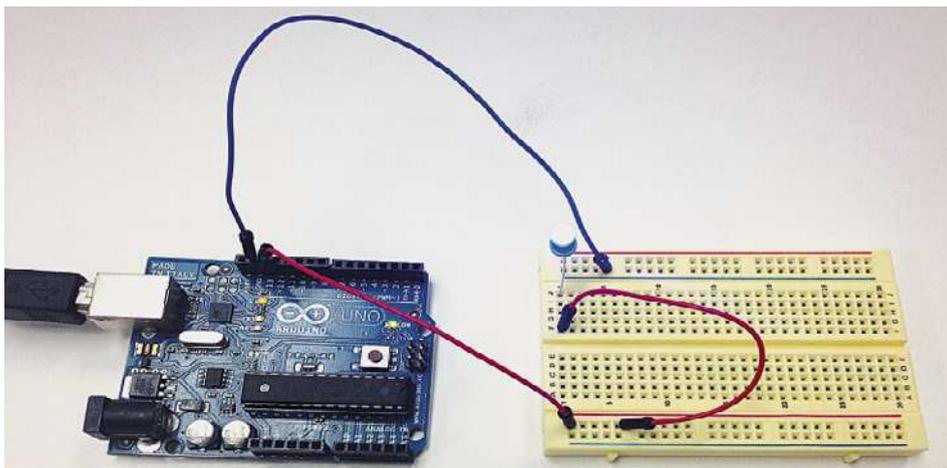


Рис. 2.5. Хотите сделать мигающий светодиод?

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

- Arduino UNO. Если вы еще не купили Arduino, то сейчас самое время! Лучше использовать версию Arduino UNO Revision 3, потому что (по крайней мере, на момент написания книги) это самая последняя и совершенная плата.
- Макетная плата. В этом проекте я использовал макетную плату половинного размера.
- USB-кабель. Для этого проекта можно использовать наиболее распространенный вид с плоским разъемом для подключения к компьютеру и квадратным для подключения к Arduino.
- Соединительные провода. Можно также использовать и обычные, подойдет любые, которые можно присоединить к макетной плате.
- Светодиоды. Поражают своей дешевизной!

Далее я использую программу для визуализации под названием «Fritzing». На **рис. 2.6** представлен пример работы с программой. Более подробно о программе *Fritzing* рассказано в **главе 8** «Ящик для инструментов».

1. Подключите светодиод к макетной плате. Длинный провод вставьте в гнездо J1, короткий — в шину заземления, расположенную рядом.

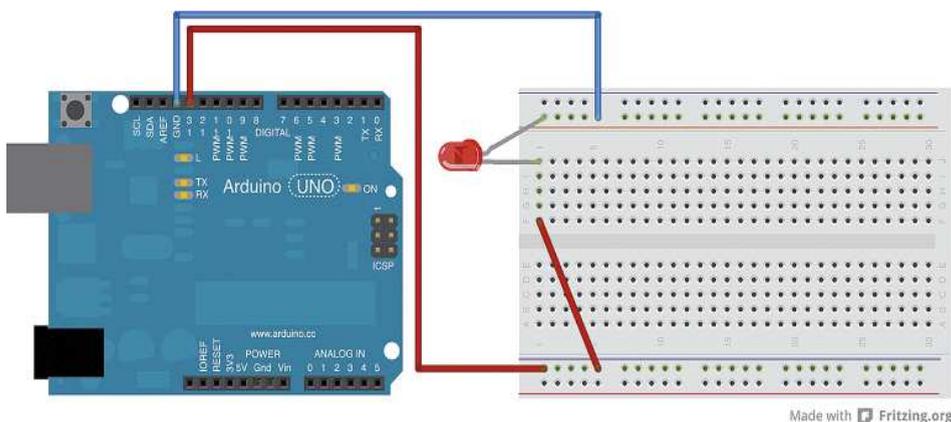


Рис. 2.6. Так мы подключаем макетную плату

2. Вставьте один конец провода в один из пинов GND Arduino (на **рис. 2.6** провод вставлен в пин GND, расположенный рядом с пином 13). Другой конец провода можно вставить в любое гнездо на шине заземления. Я выбрал пятое.
3. Подключите провод, который выходит из пина 13 на Arduino, к шине питания (обозначена красным цветом).
4. Подключите шину к колодке с гнездами, как было показано на **рис. 2.3**. Как вы видите, я выбрал гнездо F1.
5. Загрузите программное обеспечение Arduino с сайта <http://arduino.cc/en/Main/Software>, если вы не сделали это раньше. На странице изложена инструкция по установке. Для получения более подробной информации прочтите **главу 5** «Программирование Arduino». Загрузите пример «Blink»¹. В окне программного обеспечения Arduino выберите **Файл**, затем **Примеры** и далее **01.Basics**.
6. С помощью USB-кабеля подключите Arduino к компьютеру, который и станет источником питания проекта. Чтобы отправить программу на Arduino, выберите **Файл**, а затем **Загрузить**.

Вуаля! Светодиод на макетной плате должен замигать.

Если этого не произойдет, попробуйте выполнить следующие действия.

- Проверьте соединения на соответствие предшествующим инструкциям.
- Попробуйте подключить светодиод напрямую к Arduino. Вставьте длинный вывод в пин 13, а короткий в пин GND, расположенный рядом с пином 13. Если вы правильно загрузили программу, то светодиод должен замигать. Маленький светодиод на плате (обозначенный буквой «L») также мигает.
- И наконец, если что-то не заладилось, обратитесь к **главе 5**, в которой рассказано об исправлении ошибок при отладке проекта.

ДЛЯ СПРАВКИ

ПОДРОБНЕЕ О ПРОГРАММИРОВАНИИ ВЫ УЗНАЕТЕ ПОЗЖЕ

Если эти инструкции вас запутали, не отчаивайтесь! В **главе 5** вы узнаете, как программировать плату, и изучите процесс более подробно.

¹ В русифицированном интерфейсе программного обеспечения название примера не переведено. — *Прим. перев.*

⚙ Проект: лазерная сигнализация

Проект с мигающим светодиодом был только началом. Давайте сделаем что-то по-настоящему стоящее! Основной проект этой главы — изготовление лазерной сигнализации (**рис. 2.7**). Она состоит из небольшого лазера, который светит на датчик освещенности, подключенный к Arduino. Если датчик перестает обнаруживать свет, включается тревожная звуковая сигнализация. Это забавно, легко и, признаться, потрясающе!

Проект включает два модуля.

Первый модуль — это установка с лазерным излучателем. Нужно собрать аккумулятор, лазер и замок-выключатель, который запускает лазер.

Второй модуль состоит из Arduino, фоторезистора и зуммера.

Позднее вы узнаете об альтернативном способе включения сигнализации с помощью пассивного инфракрасного датчика. Это датчик движения, который используется в охранных системах.



Рис. 2.7. Лазерный луч запускает звуковую сигнализацию

ТЕХНИКА БЕЗОПАСНОСТИ

В этом проекте используется лазер малой мощности, как в лазерной указке. Этот лазер не станет причиной ожога и пожара. Тем не менее, даже такой лазер может повредить сетчатку глаза (см. **рис. 2.8**). Избегайте попадания луча лазера в глаза!



Рис. 2.8. Такие предупреждения не пишутся для красоты. Лазер может ослепить вас!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Вам понадобятся следующие компоненты.

- Arduino UNO.
- Источник питания для Arduino. Сетевой адаптер 9 В диаметром штекера 2,1 мм с центрально-положительным пином.
- Аккумулятор.
- Сугру (*sugru*). Это вещество, похожее на пластилин, а после затвердевания становится как резина. Более подробно о сугру написано в **главе 8**.
- Термоусаживаемые трубки.
- Лазерный излучатель. На **рис. 2.9** представлен лазерный излучатель, который я использовал в проекте.



Рис. 2.9. Лазерный излучатель 0,8 мВт с питанием от 3 В

- Фоторезистор 11 мм.
- Резистор 10К. Нужно приобрести в комплекте 100 шт.
- Макетная плата половинного размера.

- Два замка-выключателя.
- Зуммер.
- Провод.
- Соединительные провода.
- Изоляционные прокладки 9 мм.
- Крепежные винты № 4-40 × 25 мм.

Приступим к сборке! Начните с лазерного модуля (см. **рис. 2.10**), потому что это довольно просто. После этого переходите к сенсорному модулю.

Собираем лазерный модуль

Чтобы собрать лазерный модуль, выполните следующие действия.

1. Подключите красный провод аккумулятора к одному из выводов переключателя с ключом, как показано на **рис. 2.10**. Не имеет значения, к какому. О том, как присоединить провод к выводам замка-выключателя с помощью термоусаживаемой трубки, будет рассказано позже.
2. Подключите другой вывод переключателя с ключом к красному проводу лазера.
3. Соедините черный провод аккумулятора и черный провод лазера.

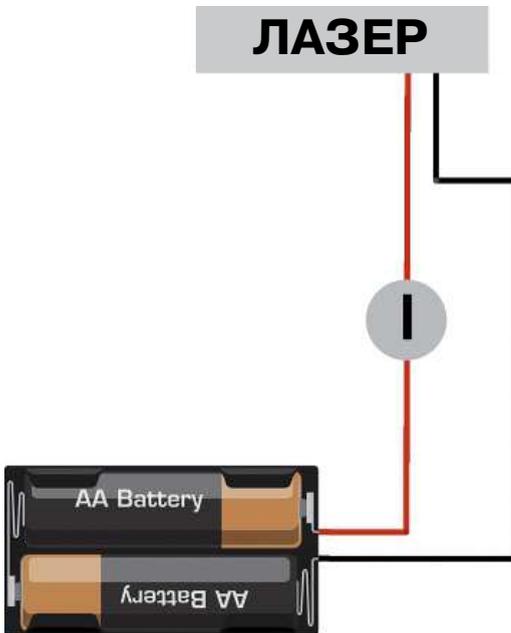


Рис. 2.10. Соберите лазерный модуль, как показано на рисунке

Собираем сенсорный модуль

Теперь перейдем к сборке сенсорного модуля (см. **рис. 2.11**).

Выполните следующие действия.

1. Подключите пин GND Arduino к шине заземления макетной платы.
2. Подключите фоторезистор и резистор к макетной плате, как показано на **рис. 2.11**. Обратите внимание, что один конец резистора подключается к шине заземления.
3. Подключите зуммер к Arduino. Вставьте черный провод в пин GND, а красный — в пин 13. (Обратите внимание, что на **рис. 2.11** я обозначил провода разными цветами, чтобы их можно было различить; у вас нет необходимости использовать разноцветные провода.)
4. Подключите замок-выключатель. На рисунке он изображен в виде серого круга с чертой. Один вывод подключите к пину 11, а другой — к шине заземления макетной платы.

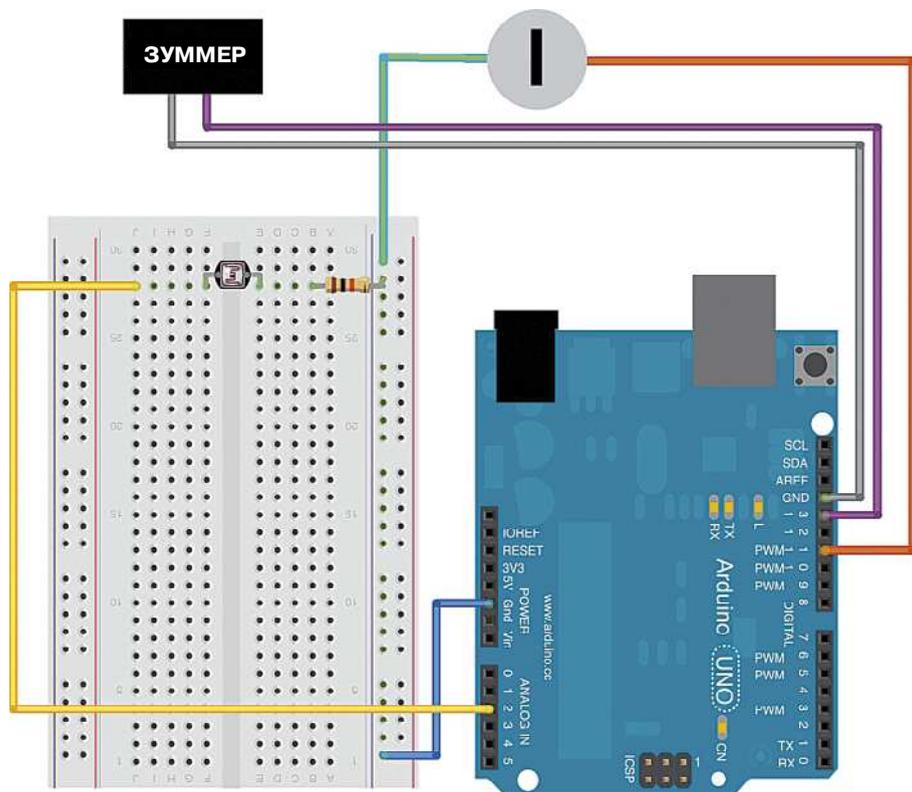


Рис. 2.11. Сборка сенсорного модуля может показаться более сложной задачей, но это только на первый взгляд!

ДЛЯ СПРАВКИ

КАК ПОЛЬЗОВАТЬСЯ ТЕРМОУСАЖИВАЕМОЙ ТРУБКОЙ

Термоусаживаемая трубка — великолепное изобретение. С ее помощью можно фиксировать провода на выводах. В сущности, термоусаживаемая трубка представляет собой эластичную трубку, которую надевают на провод и усаживают путем нагревания, что обеспечивает надежную фиксацию. С помощью термоусаживаемой трубки вы можете присоединить провода к выводам замка-выключателя. Для этого нужно выполнить следующие действия.

1. Снимите с одного конца провода примерно 1–1,5 см изоляционного материала и наденьте на провод термоусаживаемую трубку длиной в 2–2,5 см, как показано на **рис. 2.12**.



Рис. 2.12. Выводы замка-выключателя слишком гладкие, чтобы присоединить к ним провод, понадобится термоусаживаемая трубка

2. Обмотайте зачищенный конец провода вокруг одного из выводов как можно плотнее. Сделайте несколько витков проводом с изоляцией (см. **рис. 2.13**).



Рис. 2.13. Обмотайте провод вокруг одного из выводов

3. Наденьте трубку таким образом, чтобы она закрывала обмотанный провод вместе с выводом.

4. Приступите к усадке. Воспользуйтесь горелкой, паяльником или термофеном. Не обожгитесь и держитесь подальше от легковоспламеняющихся предметов. Трубка нагреется и надежно зафиксирует соединение. Результат работы должен выглядеть так, как на **рис. 2.14**. Теперь сделайте то же самое с другим выводом.



Рис. 2.14. Закрепите провод с помощью термоусаживаемой трубки

В **главе 3** «Работа с паяльником» вы узнаете, как сделать это соединение с помощью паяльника и припоя. Этот метод считается более надежным по сравнению с использованием одной только термоусаживаемой трубки!

Изготавливаем корпуса

Следующий этап проекта — изготовление корпусов. Я разработал дизайн коробок в программе *Adobe Illustrator* и вырезал панели с помощью лазерного резака. В сущности, корпус представляет собой деревянный ящик. Основная особенность дизайна — шипы размером 6 мм, которые вставляются в соответствующие пазы. В результате получается достаточно прочная конструкция.

Размер шипов панелей соответствует толщине материала (см. **рис. 2.15**). Благодаря этой особенности и точности лазерного резака получаются па-

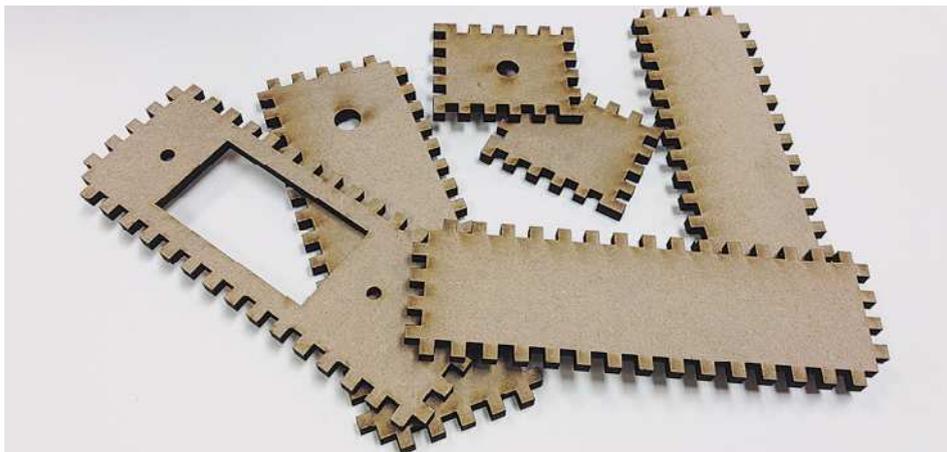


Рис. 2.15. Панели одного из корпусов готовы к сборке

нели, которые настолько хорошо соединяются друг с другом, что не требуют дополнительного склеивания: угловое соединение на прямой шип удерживает их вместе. Тем не менее, вам нужно будет склеить все панели, за исключением одной, которую вы посадите на крепежные винты.

Для того чтобы сделать корпус, необходимо выполнить следующие шаги.

1. Скачайте шаблоны с сайта <https://github.com/n1>.
2. С помощью лазерного резака вырежьте детали из МДФ (древесноволокнистой плиты средней плотности) толщиной 6 мм. Я использовал лазерный резак «Epilog» мощностью 35 Вт со следующими характеристиками: скорость — 10, мощность — 100, частота — 1000.

АЛЬТЕРНАТИВА ЛАЗЕРНОМУ РЕЗАКУ

Что вы говорите? У вас нет лазерного резака? В **главе 8** рассказано, как пользоваться этим замечательным устройством. В то же время, возможно, вам захочется сделать ящики из тех досок, что вы распилили сами, взять готовую картонную коробку или специальный корпус для проекта.

3. После того как детали будут вырезаны, приклейте первую сторону к основанию, как показано на **рис. 2.16**.



Рис. 2.16. Начните процесс сборки с приклеивания одной из сторон к основанию



Рис. 2.17. Добавьте капельку клея на каждый шип, чтобы скрепить детали

4. Затем склейте остальные детали, за исключением задней панели; она должна оставаться съемной, чтобы можно было добавить электронику. Предлагаю склеивать детали путем нанесения капли клея на каждый шип, как показано на **рис. 2.17** (наверное, я использовал слишком много клея). Протрите излишки клея и дайте корпусу высохнуть. Обратите внимание, что вам, вероятно, не придется использовать дополнительные методы фиксации; детали очень плотно прилегают друг к другу.

Итак, корпуса готовы; они должны выглядеть так, как показано на **рис. 2.18**. Возможно, вы захотите покрасить их.

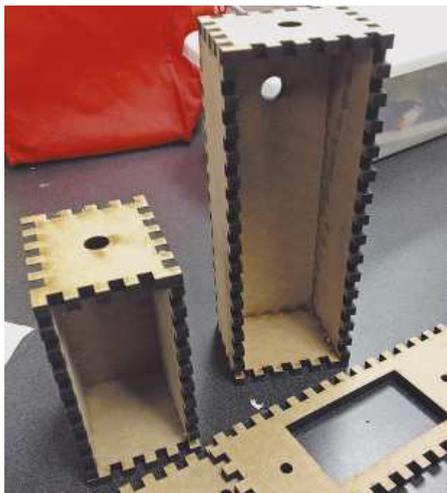


Рис. 2.18. Корпусы собраны. Теперь их можно покрасить и добавить электронику

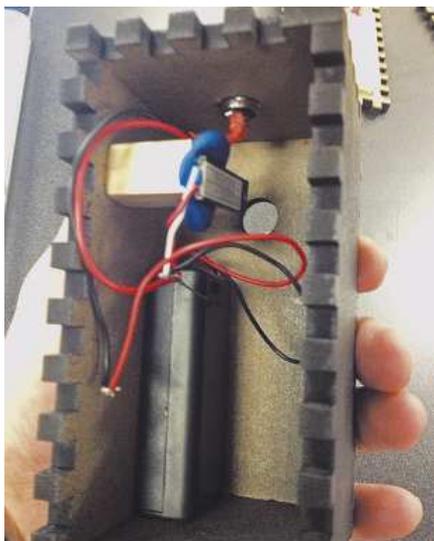


Рис. 2.19. Лазерный модуль. Комок синего цвета — сугру, с помощью которого лазер прикреплен к деревянному бруску

Настало время поместить электронику в корпус лазерного модуля (в тот, что меньше по размеру). Вы уже собрали внутреннее наполнение при выполнении шагов 1–4, поэтому будет легко!

1. Вы можете прикрепить аккумулятор к внутренней поверхности корпуса с помощью винтов или термоклеевого пистолета или оставить его незафиксированным.
2. Вставьте замок-выключатель в верхнее отверстие корпуса и затяните гайку. Если замок-выключатель будет вращаться, зафиксируйте место крепления с помощью термоклеевого пистолета.

Приклейте деревянный брусочек к внутренней поверхности корпуса, как показано на **рис. 2.19**. Я взял брусочек длиной примерно 2 см. С помощью кусочка сугру (подробнее об этом материале см. в **главе 8**) прикрепите лазер к бруску так, чтобы луч светил через отверстие на фронтальной панели корпуса. Убедитесь в том, что лазер будет светить именно так, как вы этого хотите. Оставьте сугру застывать на ночь: он зафиксирует лазерный излучатель.

3. Прикрепите заднюю панель к корпусу лазерного модуля с помощью крепежных винтов.

Теперь поместите электронику в корпус сенсорного модуля.

1. С помощью крепежных винтов № 4-40 и изоляционных прокладок № прикрепите Arduino к фронтальной панели корпуса (см. **рис. 2.20**). С помощью винтов закрепите зуммер как можно ближе к Arduino, чтобы их можно было соединить.

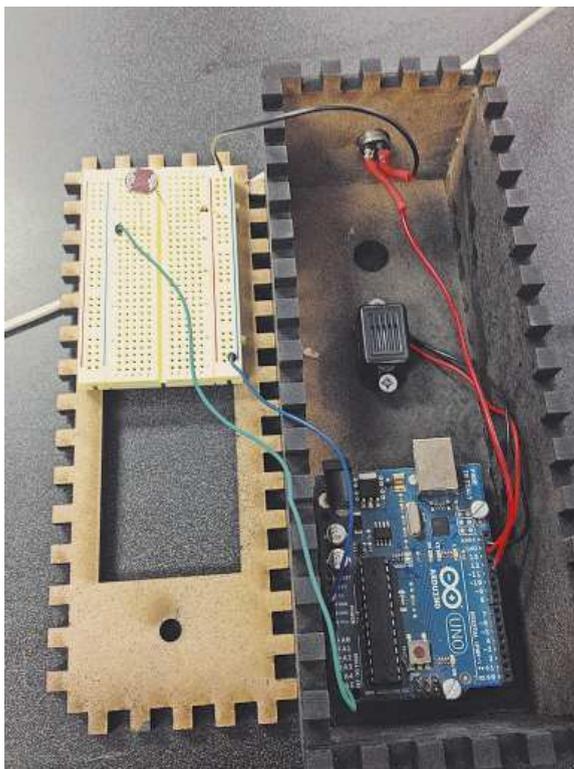


Рис. 2.20. Корпус сенсорного модуля без задней панели

2. Снимите защитную пленку с клейкого слоя на макетной плате и прикрепите плату к задней панели. Убедитесь, что вы оставили достаточно места для источника питания.
3. Вставьте замок-выключатель в верхнее отверстие корпуса, как в случае с лазерным модулем.
4. Если вы соединили все компоненты так, как это было описано ранее, установка должна быть готова. Когда модуль будет собран, убедитесь в том, что фоторезистор установлен так, что его видно через фронтальное отверстие.

Код для лазерного луча

Теперь вы можете запрограммировать Arduino. Обратите внимание, что у вас должна быть установлена последняя версия программного обеспечения Arduino, иначе могут возникнуть ошибки. Вы можете найти ее на сайте arduino.cc.

Загрузите код с сайта <https://github.com/n1/Arduino-For-Beginners>. Не знаете, как загрузить код на Arduino? Прочтите первую часть **главы 5**.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
// определяем, какие пины будут задействованы в программе
#define sensorPin A2
#define buzzerPin 13
#define keylockPin 11 int sensorValue = 0;
int threshold = 0;

// увеличение значения переменной, если звуковая сигнализация
// включается слишком быстро
void setup() {
    // в этой части устанавливаем каждый порт в режим входа или пина
    pinMode(keylockPin, INPUT_PULLUP);
    pinMode(buzzerPin, OUTPUT);
    pinMode(sensorPin, INPUT);
    Serial.begin(115200);
}
void loop() {
    // этот цикл активирует звуковую сигнализацию в зависимости
    // от положения замка-выключателя
    while (digitalRead(keylockPin) == HIGH) {
        senseIntruder();
    }
}
void senseIntruder() {
    // эта функция сравнивает значения датчика освещенности с пределом
    // возможных значений для отслеживания непрерывности излучения
    int sensorValue = analogRead(sensorPin);
    Serial.println(sensorValue); // для отладки

    if (sensorValue > threshold) {
        digitalWrite(buzzerPin, HIGH);
        delay(2000); // устанавливает длительность звуковой
                    // сигнализации,
                    // чем выше значение, тем продолжительнее сигнал
    }
    else {
        digitalWrite(buzzerPin, LOW);
    }
    delay(20);
}
```

Устанавливаем лазерный блок

Теперь, когда два модуля собраны, настало время установить их (на **рис. 2.21** изображен лазерный блок, собранный мной).

1. Выберите дверной проем или коридор, где вы хотите установить средство защиты. Установите два модуля так, чтобы лазерный луч попадал на датчик освещенности.
2. Найдите идеальное место для розетки сенсорного модуля, чтобы исключить случайное отсоединение, и подключите его.
3. Если это невозможно сделать, то можете подключить к Arduino батарейку 9 В, чтобы обеспечить питание. У компании *Adafruit* есть удобный аккумулятор со штекером, совпадающим с гнездом на Arduino.
4. После того как корпуса будут установлены, поверните ключ на лазерном модуле и запустите луч.
5. Поверните ключ на сенсорном модуле. Луч готов к обороне!



Рис. 2.21. Лазерный луч в работе. Да, изображение размыто. Вы когда-нибудь пробовали фотографировать невидимый луч света в темной комнате?

В зависимости от освещенности вашей комнаты необходимо внести следующие корректировки.

- Если звуковая сигнализация слишком быстро выключается, задайте другую пороговую величину в коде.
- Если звуковая сигнализация срывается, поменяйте резистор на макетной плате; вместо 10 кОм используйте 5 кОм. Это расширит рабочий диапазон датчика освещенности.

⚙ Проект: инфракрасный детектор

Разумеется, лазерная сигнализация — не самое надежное средство защиты. Существует еще один способ, пожалуй, более эффективный, но менее интересный — использование инфракрасного датчика (см. **рис. 2.22**) с целью обнаружения непрошенных гостей.

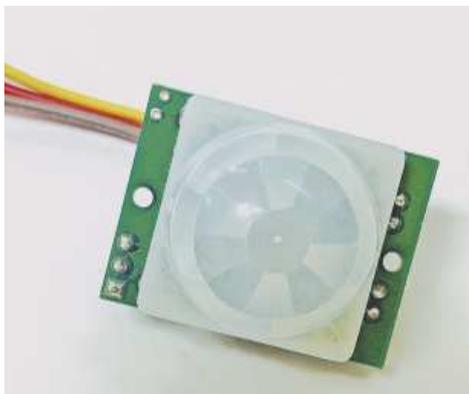


Рис. 2.22. Пассивный инфракрасный датчик — важный элемент в профессиональных системах безопасности. Почему бы нам не задействовать его?

Пассивный инфракрасный датчик обнаруживает еле уловимые колебания инфракрасного света в пространстве, фиксируя перемещения в контролируемой зоне. При обнаружении чего-либо пассивный инфракрасный датчик отправляет сигнал Arduino.

Подключаем пассивный инфракрасный датчик и зуммер

Вам понадобится пассивный инфракрасный датчик. Он состоит из пластикового колпачка, который защищает ИК-излучатель, и приемника. На плате датчика имеются три вывода: один для питания 5 В пина Arduino, другой посылает сигнал в пин 7, а третий соединяется с заземлением. Подключите пассивный инфракрасный датчик и зуммер, как показано

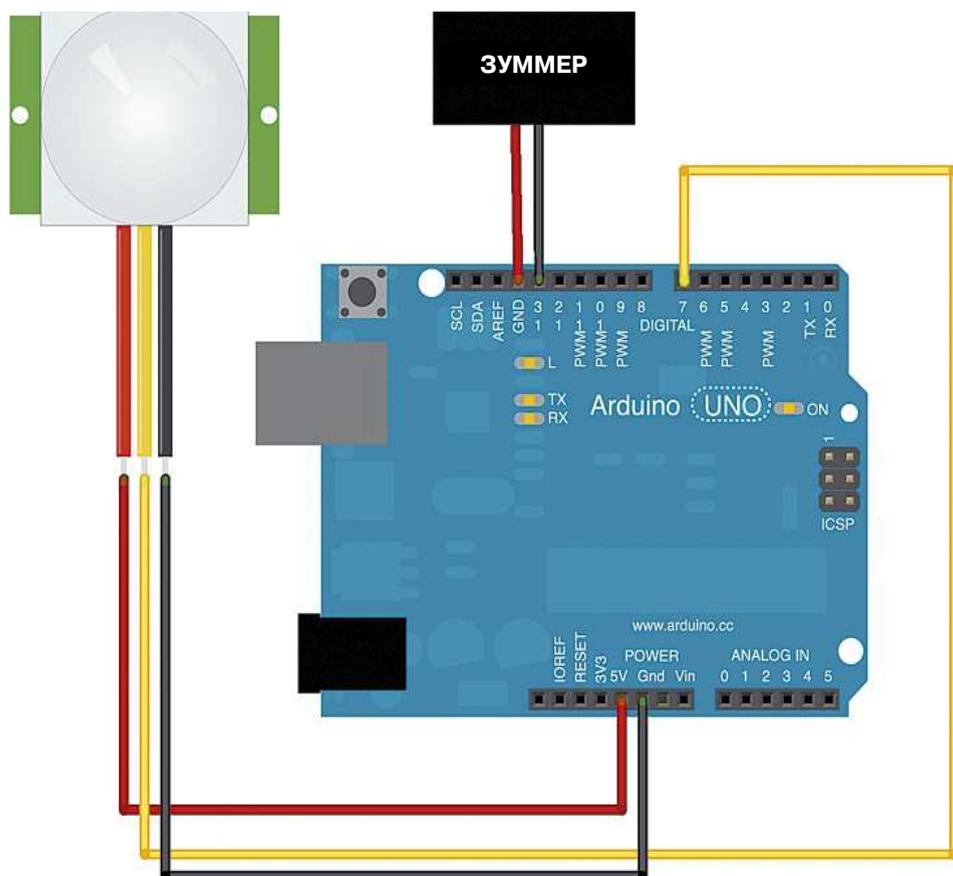


Рис. 2.23. Подключить пассивный инфракрасный датчик достаточно просто!

на **рис. 2.23**. Сборка закончена! Датчик имеет рабочую зону радиусом 6 м с обзором 120 градусов. Направьте датчик на входную дверь; если кто-нибудь войдет в нее или пересечет невидимую зону инфракрасного излучения, сработает сигнализация.

Код для инфракрасного детектора

Чтобы запрограммировать систему сигнализации, загрузите на Arduino следующий код:

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#define buzzerPin 13 // пин 13 управляет светодиодами

#define pirPin 4 // пин 4 принимает данные с датчика движения

int val = 0; // устанавливает значение по умолчанию для датчика
// движения

void setup()
{
  // устанавливаем зуммер и пассивный ИК-датчик в режим входа или выхода
  pinMode(buzzerPin, OUTPUT);
  pinMode(pirPin, INPUT);
  Serial.begin(115200);
}

void loop()
{
  // цикл находится в ожидании срабатывания ИК-датчика,
  // после чего выключается звуковая сигнализация
  val = digitalRead(pirPin);

  if (val == HIGH) {
    digitalWrite(buzzerPin, HIGH);
    Serial.println(val);

    delay(200); // длительность сигнала зуммера в миллисекундах
  }
  else {
    digitalWrite(buzzerPin, LOW);
  }
}
```

В следующей главе

Вы освоили макетирование. Пора учиться паять! В **главе 3** «Работа с паяльником» вы узнаете, как собрать схему с помощью паяльника и припоя. Также вы научитесь делать управляемую светодиодную ленту, которая создает превосходные световые эффекты.

Работа с паяльником

Припой — это легкоплавкий сплав свинца и олова, который используется для соединения электронных компонентов между собой и с печатной платой. Он проводит электричество так же, как провода, что позволяет функционировать электрической цепи (**рис. 3.1**).

Большинство электронных комплектов поставляются с печатной платой (*printed circuit board*, РСВ), изображенной на **рис. 3.2**, к которой вы припаиваете элементы. Как правило, плата состоит из листа фольгированного стеклотекстолита с многочисленными монтажными отверстиями и маркировкой в виде трафаретной печати с вспомогательным рисунком и текстом.



Источник: Waugh & Laugel

Рис. 3.1. Пайка может выглядеть довольно устрашающе, но паять на самом деле просто:

1 — спаянные соединения

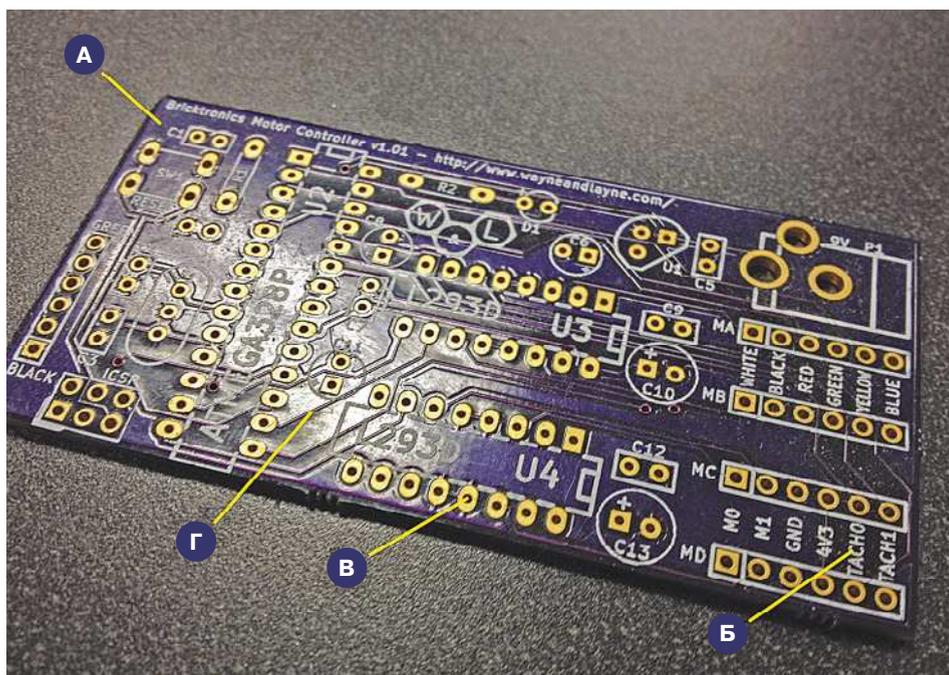


Рис. 3.2. Стандартная печатная плата

Ламинат имеет встроенные соединения, называемые *дорожками* (отмечены на **рис. 3.2**), которые соединяют все элементы в цепь. Но как подключить элементы к дорожкам? Если вы посмотрите на фото, то увидите крошечные металлические площадки вокруг каждого отверстия. Это *монтажные отверстия*. Если вы хотите прикрепить деталь к плате, пропустите выводы деталей (которые также называют ножками) через отверстия и припаяйте их в этом месте.

Воспользуйтесь следующими обозначениями, чтобы определить различные части печатной платы, изображенной на **рис. 3.2**:

- А — текстолит;
- Б — маркировка;
- В — монтажные отверстия;
- Г — дорожки.

Остальная часть этой главы посвящена обучению пайке.

ТЕХНИКА БЕЗОПАСНОСТИ

Неудивительно, что вы можете нанести вред своему здоровью во время пайки. Помните о следующем.

- Паяльник горячий, его жало может нагреваться до 600 °С! Он может стать причиной пожара, если вы не будете осторожны.
- Используйте защиту для глаз, когда отрезаете провод. Неплохо надеть защитные очки, даже если вы носите обычные очки. Иногда куски провода могут отлетать рикошетом!
- Пары химических веществ, выделяемые при пайке, ядовиты. Убедитесь, что у вас хорошая вентиляция или приобретите дымоуловитель (см. раздел «Вентиляторы и дымоуловители» далее в этой главе), чтобы воздух всегда был чистым.

После того как вы закончите работу над проектом (и даже в перерыве между работой), тщательно вымойте руки с мылом. Во время работы помните о том, что свинец токсичен, поэтому не трогайте руками лицо.

Паяльные принадлежности

Несомненно, для пайки нужен паяльник. Но многие не понимают, что помимо паяльника необходимо множество других принадлежностей. Ниже даются некоторые советы по приобретению оборудования.

Выбираем паяльник

Понятно, что вам нужен паяльник, но какой? Стоимость паяльников сильно варьируется. Базовая модель «паяльник-ручка» (**рис. 3.3**), как правило, состоит из рабочей части, обычно называемой жалом, и электрического шнура. Вы не можете ни регулировать температуру, ни делать что-либо еще.



Рис. 3.3. Паяльник-ручка — отличный выбор для начинающего монтажника



Рис. 3.4. Относительно дорогой «Weller WES51» обладает большими возможностями, чем паяльник-ручка

Более сложная модель, как «Weller WES51», изображенная на **рис. 3.4**, имеет больше возможностей. В комплект входит подставка для паяльника, чтобы раскаленное жало случайно ничего не прожгло. Подставка также имеет губку для чистки жала паяльника, что важно для поддержания его температуры.

Более впечатляющие, дорогие модели имеют более надежную подачу питания. Паяльник «Weller», изображенный на **рис. 3.4**, может нагреться точно до той температуры, которая вам необходима для пайки. Кроме того, паяльник имеет выключатель питания и даже светодиодный индикатор температуры жала. WES51 стоит на 100 долл. дороже, чем самый простой паяльник, но, поверьте мне, вы сразу почувствуете разницу. Самая большая ошибка в обучении пайке — пользоваться некачественным паяльником.

ЛУЖЕНИЕ ЖАЛА

Неважно, какой паяльник вы приобрели, главное, что у него есть жало — это та часть паяльника, которая плавит металл. Жало паяльника легко подвергается коррозии, которая ухудшает способность к нагреванию. Сохраняйте кончик паяльника настолько идеальным, насколько это возможно. После окончания работы погрузите жало в олово (**рис. 3.5**). Лужение означает покрытие кончика паяльника расплавленным оловом, защищающим жало от коррозии. Это необходимо делать постоянно во время пайки.



Рис. 3.5. Надо погружать кончик паяльника в олово до, во время и после пайки

Выбираем припой

Без припоя пайка невозможна. Давайте рассмотрим различные типы припоя (рис. 3.6).

- **Свинцовый или бессвинцовый.** Обычный припой представляет собой сплав олова и свинца, поэтому свинцовый припой наиболее распространен. В зависимости от особенностей пайки и личных предпочтений вы можете купить припой различных размеров (самый распространенный толщиной 0,8 мм) и сплавов (обычно 63/37 и 60/40). Большинство монтажников подтверждают, что свинец делает припой лучше. Однако экологические законы и опасения по поводу отравления свинцом вынуждают производителей и любителей использовать припой без свинца.

В бессвинцовых припоях свинец заменен на смесь сурьмы, цинка, серебра и других ингредиентов, которые варьируются от продукта к продукту. Многим любителям не нравится припой без свинца, поскольку он имеет более высокую точку плавления, чем свинец, поэтому не так легко плавится. Кроме этого, пайка получается более грязной и может быть подвергнута коррозии. Рекомендую использовать все-таки свинцовый припой, просто чаще мыть руки во время и после пайки.

- **С паяльным флюсом (с жидкой основой) или без флюса (с твердой основой).** Большинство припоев поставляется с паяльным флюсом. Он используется, чтобы химически очистить поверхность, усилить механическую связь между припоем и местом пайки и оптимизировать проводимость. Самый популярный флюс — канифоль, которую получают из очищенной сосновой смолы. Припой с канифольным флюсом сильно дымится, а дым может вызвать проблемы со здоровьем, такие как раздражение дыхательных путей, затрудненное дыхание и астму. Тем не менее, с помощью дымоуловителя (см. раздел «Вентиляторы и дымоуловители» далее в



Рис. 3.6. Припой отличается составом сплава и размерами. Убедитесь, что вы выбираете тот, который лучше всего подходит для вашего проекта

этой главе) и хорошей вентиляции можно избежать неприятностей. Припой на твердой основе не содержат флюса; вряд ли кто-то использует их, за исключением художников-витражистов, которым не нравятся пятна флюса на своих творениях.

- **Универсальный.** Я уже привел достаточно сведений о припоях. Поэтому позвольте мне предложить свинцовый припой 60/40 толщиной 0,8 мм на канифольной основе. Это универсальный припой, который я использую для всех своих проектов.

Другие необходимые компоненты

Итак, у вас есть паяльник и припой. Что дальше? Давайте рассмотрим некоторые дополнительные аксессуары, которые помогут сделать ваш опыт пайки легким и успешным.

Настольные тиски

Настольные тиски, которые называют «третьей рукой» (**рис. 3.7**), используются для удержания печатной платы во время пайки в устойчивом положении с помощью зажимов. Некоторые модели поставляются с увеличительным стеклом, которое может пригодиться при монтаже и пайке мелких и сложных элементов.



Рис. 3.7. Тиски удерживают печатную плату во время пайки



Рис. 3.8. «Panavise Jr.» — еще один отличный способ надежной фиксации вашей платы

Компания *Panavise* производит небольшие настольные тиски. Модель «Panavise Jr. Model 201» (**рис. 3.8**) чрезвычайно популярна среди любителей из-за надежного удерживания печатных плат во время пайки. По сути, это небольшие тиски, которые можно прикрепить к верстаку с помощью болтов; эти тиски прочно удерживают печатную плату в любом положении.

Плоскогубцы и кусачки

Вам обязательно понадобятся плоскогубцы и кусачки (**рис. 3.9**) для обрезки проводов и очистки их от изоляции. Некоторые модели сочетают оба инструмента, но я предпочитаю иметь отдельные.



Рис. 3.9. Плоскогубцы и кусачки — неотъемлемая часть любого набора инструментов

Острогубцы и пинцеты

Возможно, вам также потребуются острогубцы и гемостатический пинцет (**рис. 3.10**), чтобы брать мелкие предметы — в электронике много мелких деталей! Компания *Adafruit* предлагает ряд недорогих пинцетов с несскользящим покрытием, что помогает свести к минимуму вероятность случайного пробоя деталей статическим электричеством.



Рис. 3.10. Нужно взять и удержать мелкую деталь? Острогубцы и пинцеты — это то, что вам нужно!

Вентиляторы и дымоуловители

Если в вашей мастерской плохая вытяжка, в первую очередь нужно распахнуть настежь форточку! Постарайтесь также использовать вентилятор или дымоуловитель, чтобы обезопасить себя от паров канифоли. На **рис. 3.11** изображен обычный вентилятор, который можно купить практически везде. Профессиональные дымоуловители стоят дороже и содержат дополнительные компоненты, такие как фильтр из углеродного волокна.



Рис. 3.11. Простейший вентилятор

Антистатическая защита

При работе с электроникой надо обезопасить себя от электростатического разряда, так же известного как статика. Носите антистатический браслет (**рис. 3.12**) или работайте на резиновом коврикe, чтобы исключить наведение статического заряда.



Рис. 3.12. Антистатический браслет защитит вас от электростатического разряда

Подставка под паяльник и губка

Возможно, в дополнение к паяльнику-ручке вы захотите приобрести подставку под паяльник (**рис. 3.13**). Также вам обязательно понадобится губка, чтобы поддерживать в чистоте жало паяльника. Компания *Adafruit* предлагает хорошее решение — подставку с губкой. В комплект входит дозатор припоя.



Рис. 3.13. Горячий паяльник должен находиться на столе в безопасном месте

Паяние

Теперь, когда у вас есть все необходимое оборудование, можно переходить к пайке! Вот как это делается.

1. Подготовьте паяльное оборудование и рабочую зону (см. **рис. 3.14**). Убедитесь, что у вас достаточно рабочего пространства, что ваши кусачки, губка и другие инструменты лежат в полной боевой готовности. Подключите паяльник к розетке, и, если у модели есть тумблер, включите его. Если можно регулировать температуру, установите показатель на 370°C для оловянно-свинцового припоя или на 400°C для припоя без свинца.
2. Ваш паяльник нагревается. Если в нем имеется термореле, то вы узнаете, когда паяльник станет достаточно горячим. Если у вас базовая модель паяльника, необходимо проверить жало: коснитесь им влажной губки. Если он разогрелся, то шипящая струйка пара укажет на это.
3. После того как паяльник разогреется, растопите припой и окуните в него жало паяльника. Этот процесс называется **лужением**. Такая процедура облегчает теплопередачу и тем самым ускоряет пайку.
4. Когда вы будете готовы, перейдите к шагу 1 инструкции (предполагается, что вы работаете с базовым обучающим набором). В ин-

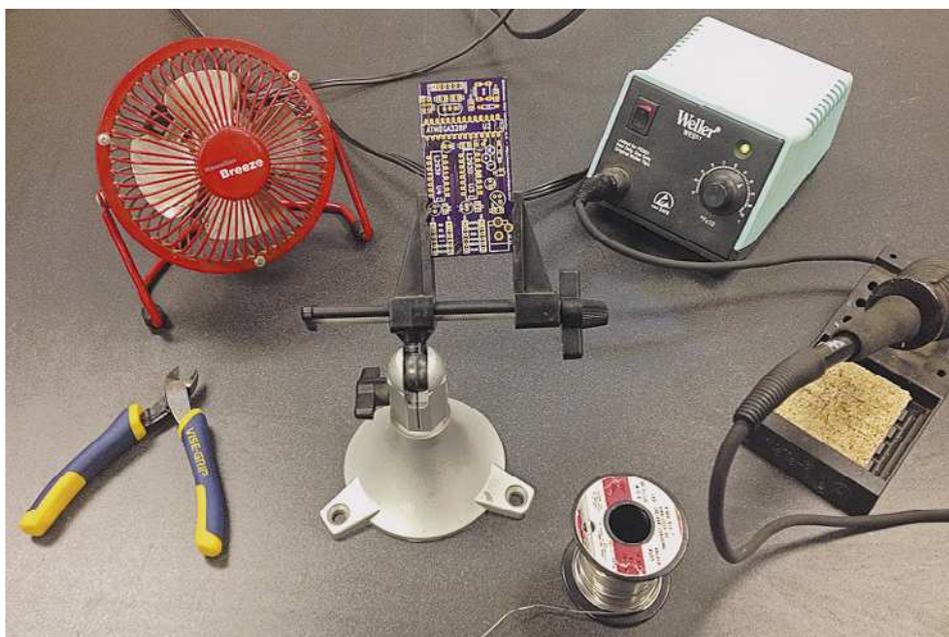


Рис. 3.14. Ну что, все готово? Тогда начнем!

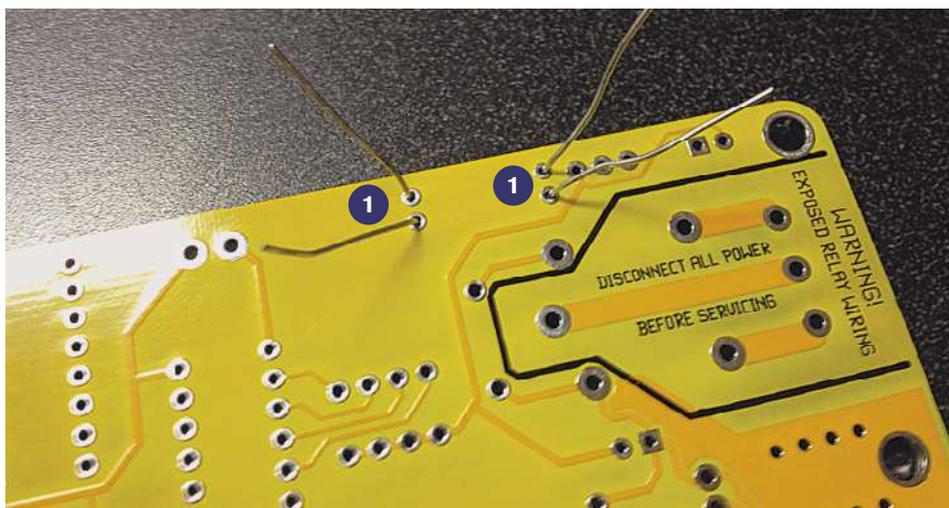


Рис. 3.15. Лучший способ предотвратить падение деталей с платы, когда вы переворачиваете ее, это отогнуть провода на обратной стороне:

1 — вполне достаточно отогнуть провод детали, чтобы предотвратить ее падение

струкции, входящей в комплект, обычно изложено, в какой последовательности проводить пайку. Пропустите провода деталей через отверстия в контактных площадках. Переверните плату и отогните назад провода (как показано на **рис. 3.15**), чтобы детали не упали с платы.

5. Прижмите провод к печатной плате жалом паяльника так, как показано на **рис. 3.16**. Подержите жало несколько секунд, чтобы провод и площадка разогрелись, и вы могли спаять их вместе.

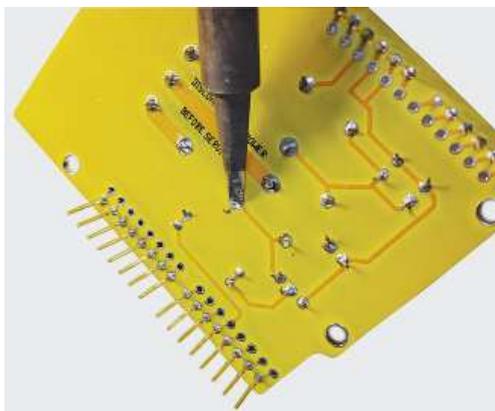


Рис. 3.16. Нагревание провода и площадки

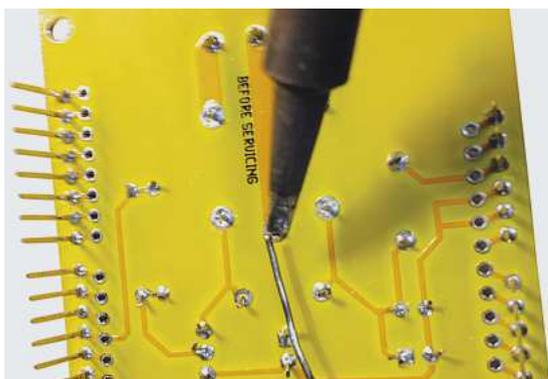


Рис. 3.17. Нагрейте провод и площадку и затем поднесите припой

6. Поднесите припой к жалу паяльника, приложенного к площадке печатной платы (**рис. 3.17**). Несколько секунд подержите так. Припой должен расплавиться и заполнить отверстие, образовав небольшую выпуклость. Уберите паяльник. На охлаждение уйдет меньше секунды, так что вам не придется ждать, чтобы перейти к следующему шагу.
7. Осмотрите место пайки. Если припой не заполнил всю площадку, то его было мало. И наоборот, если припой залил соседние площадки, схема будет функционировать не так, как предполагалось. В любом случае вы должны будете провести распайку (см. следующий раздел) и переделать свою работу.
8. Если место пайки выглядит хорошо, обрежьте лишние провода; они больше не понадобятся (**рис. 3.18**). Вы можете перейти к пайке следующей детали.

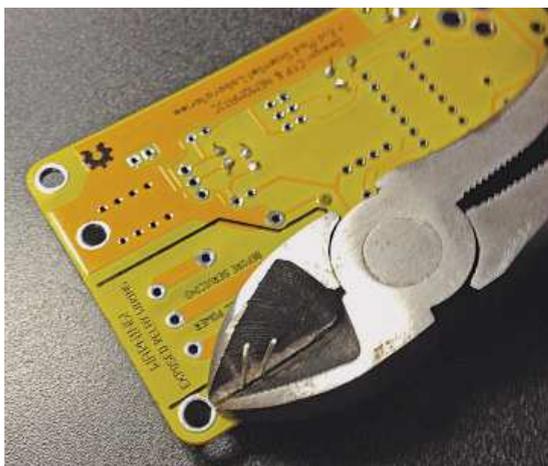


Рис. 3.18. Обрежьте лишние провода, если уверены, что деталь надежно закреплена

Распайка

Если пайка не удалась (см. **рис. 3.19**) или вы случайно припаяли неправильный участок или припаяли детали наоборот, необходимо удалить припой и переделать работу.

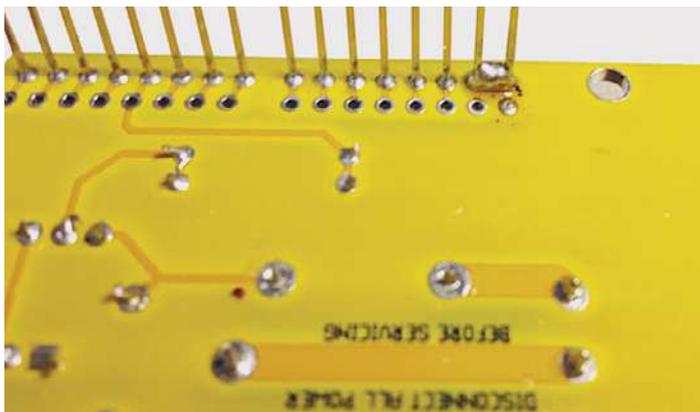


Рис. 3.19. Видите два слипшихся контакта? Это плохая пайка

При распайке используется ряд приспособлений (**рис. 3.20**), с помощью которых можно удалить расплавленный припой. К ним относятся демонтажная груша, медная оплетка и паяльный отсос. В итоге вы выберете наиболее удобный для вас способ (на заметку: мне больше всего нравится паяльный отсос).

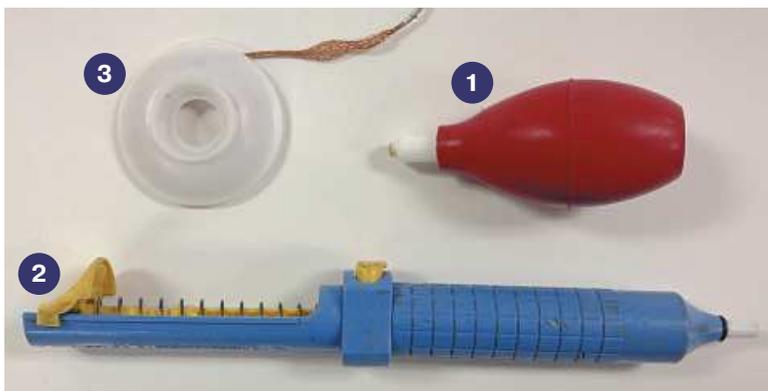


Рис. 3.20. Три инструмента, которые нужны для распайки:
1 — демонтажная груша; 2 — паяльный отсос; 3 — медная оплетка

Для распайки вам понадобятся:

- **демонтажная груша.** Это полый резиновый шарик с носиком. Для распайки поднесите носик сжатой груши к припою и расплавьте шарик паяльником. Чтобы убрать припой, разожмите грушу, припой всосется;
- **паяльный отсос.** Это пружинная версия груши, в которую входит поршень и кнопка. При распайке надавите на поршень до щелчка; щелчок означает, что инструмент готов к работе. Удерживая кончик отсоса рядом с шариком припоя, нажмите кнопку. Пружина расслабится, и поршень отскочит назад, создавая вакуум, который всосет припой с печатной платы (**рис. 3.21**);
- **медная оплетка.** Вместо того чтобы пытаться высосать припой, почему бы не впитать его, как лужу разлитого молока? Оплетка — это неплотно переплетенные жилы медного провода. Когда они касаются расплавленного припоя, он течет по оплетке от вашей платы.

После того как вы проведете распайку, внимательно осмотрите плату, чтобы убедиться, что на ней не осталось следов припоя, а затем перепаяйте деталь.



Рис. 3.21. Нагрейте неудачную пайку и удалите расплавленный металл!

СОВЕТ

Вот несколько советов, которые помогут вам освоить технику паяния.

- Лучше меньше, да лучше. Вам не нужно много припоя, чтобы провести качественную пайку. Слишком много припоя может замкнуть два контакта, а это плохо.
- Жало любит олово. Периодически окунайте кончик паяльника в олово, чтобы обеспечить лучшую теплопроводность припоя.
- Сгибайте провода. При пайке детали отгибайте провода, чтобы предотвратить выпадение детали с платы.
- Припаивайте один временный провод. Если вы беспокоитесь, что деталь сядет на плату неровно, припаяйте только один ее конец. Затем нагрейте припой еще раз и отрегулируйте посадку детали. После того как деталь встанет ровно, дайте припою остыть и припаяйте оставшиеся провода.
- Нагревайте площадку и провода, но не припой. Распространенная ошибка среди новичков — расплавить припой и размазать его по всему проводу. Сделайте по-другому: нагрейте площадку и провода, а затем поднесите припой и дайте ему расплавиться.
- Держите жало в чистоте. Периодически очищайте кончик паяльника, вытирая его о губку. Затем опускайте его в олово и продолжайте паять.
- Храните паяльник с жалом, защищенным оловом, это предохранит кончик паяльника от коррозии.

Уборка

Вы закончили! Поздравляем с освоением нового навыка. Теперь настало время убрать инструменты и рабочее место. Ниже изложен возможный порядок действий.

1. Опустите в олово жало паяльника. Вы уже знаете, как это делать. Покрытие жала оловом защищает его от коррозии, пока паяльник не используется.
2. Отключите от сети инструменты и уберите их на хранение.
3. Уберите свое рабочее место. На вашем рабочем столе и под стулом могут остаться крошечные капли токсичного свинца, а также обрезки проводов. Для уборки пола используйте щетку или пылесос. Протрите стол чистящим средством и обычной тряпкой.
4. И наконец, после того как вы убедитесь, что все вокруг блестит, тщательно вымойте руки.

⚙️ Проект: кофейный столик со светодиодной лентой

В этом проекте вы узнаете, как украсить обычный кофейный столик светящейся лентой, которая добавит изюминку к очередной кофе-сессии! Лента состоит из металлической полоски из фольги, усеянной светодиодами и микрочипами. С помощью программы Arduino вы можете контролировать яркость и цвет каждого светодиода (**рис. 3.22**). С помощью кнопки вы сможете выбрать нужный световой эффект за несколько секунд.



Рис. 3.22. Кофейный столик зажжет ваше общение — в буквальном смысле!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для этого проекта вам понадобится немного элементов.

- Arduino UNO.
- Блок питания для Arduino UNO, который также станет источником питания для светодиодной ленты.
- Цифровая светодиодная лента (можете приобрести ленту нужной длины).
- Кофейный столик.
- Соединительные провода.
- Кнопка (я использовал клавишную кнопку «U811SHZGE» от *Digi-Key*).
- Потенциометр.
- Кабельные стяжки (по желанию).
- Клеевой термопистолет и клей (по желанию).

Подготавливаем светодиодную ленту

Цифровые трехцветные светодиодные полоски (RGB) состоят из фольгированной ленты со встроенными крошечными микрочипами и светодиодами. На ленте длиной 1 м расположено 32 светодиода. Яркость и цвет каждого светодиода могут по отдельности контролироваться Arduino (рис. 3.23).

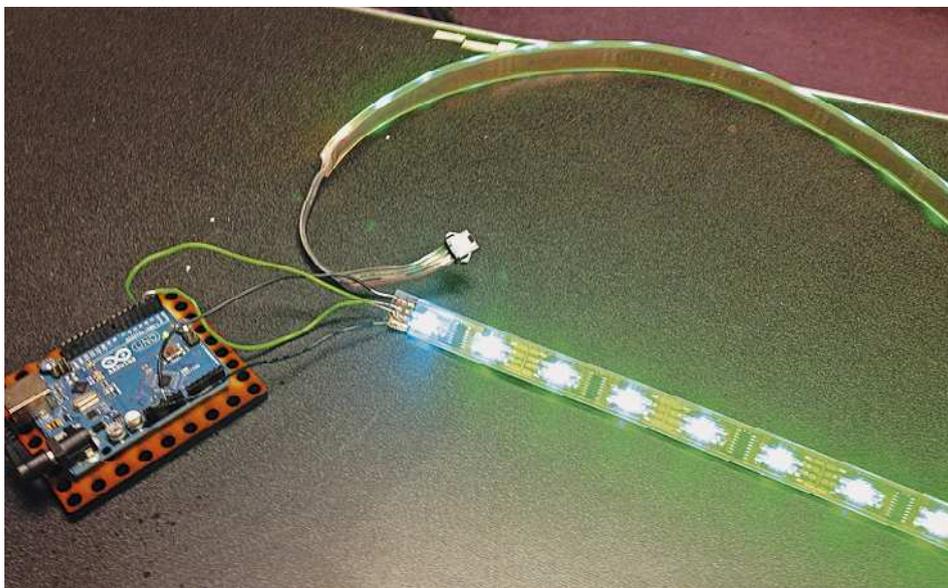


Рис. 3.23. Все, что вам нужно, чтобы управлять одной из этих светодиодных лент, — это Arduino и источник питания

Первое, на что вы обратите внимание, — что фольгированная лента покрыта прозрачной пленкой, которая защищает ленту от влаги. Но не торопитесь сразу же снимать ее. Вернуть защитную пленку на место будет очень трудно.

Вы покупаете ленту в виде рулона длиной 5 м, но хотите нарезать ее на меньшие кусочки. Можно ли при нарезке повредить вещь, которая стоит 30 долл. за метр? Можно. Но производители предусмотрели это и сконструировали ленту так, что ее можно резать в определенных местах. Безопасные линии разрезов между контактными площадками показаны на рис. 3.24.

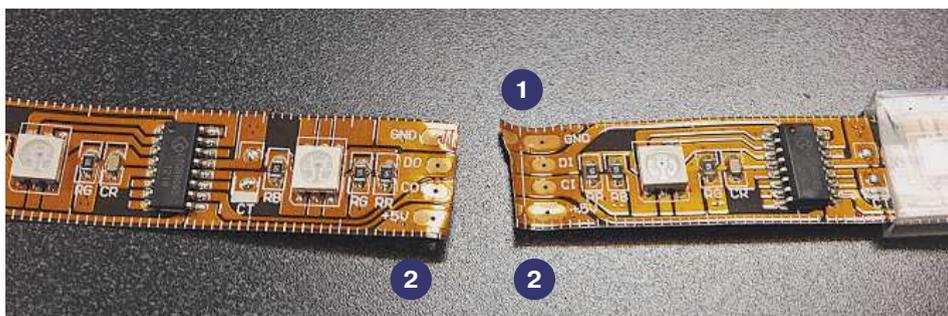


Рис. 3.24. Безопасное место разреза светодиодной ленты между контактными площадками:

1 — линия разреза; 2 — контактные площадки

Чтобы подготовить светодиодную ленту для проекта, выполните следующие действия.

1. Найдите ближайшее место безопасного разреза. Вы сразу увидите его, такие линии расположены через каждые 6 см. Просто нарежьте ленту на кусочки по безопасным линиям с помощью надежных ножниц. Это не хирургическая операция; простой разрез, показанный на **рис. 3.24**, сделан прекрасно.
2. Соедините нарезанные кусочки в одну ленту, чтобы было проще обернуть ее вокруг стола. Возьмите два кусочка и положите их рядом. Оттяните пластиковую защиту на 2–3 см, чтобы обнажить контактные площадки.
3. Капните припой на каждый контакт, а затем олово на концы соединительного провода (длина оголенного провода на перемычке составит примерно 6 мм) и припаяйте его на контакт, как показано на **рис. 3.25**. Некоторые любители дополнительно используют

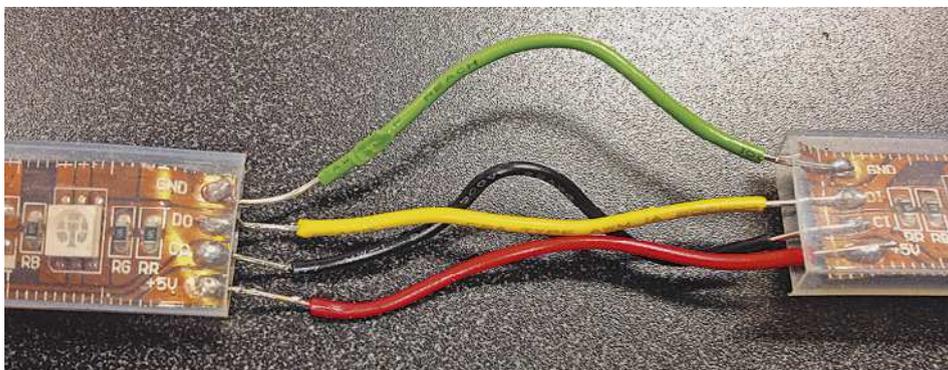


Рис. 3.25. Соедините каждый контакт с его приятелем-тезкой на другой полоске

термоклею для открытых контактов, чтобы обеспечить прочность соединения.

4. Соедините оставшиеся контакты точно так же.

В качестве альтернативы компания *Adafruit Industries* продает изолирующие колпачки и разъемы, которые делают использование световой ленты более удобным, потому что можно подключать и отключать отрезки ленты по своему желанию, что экономит время для пайки.

Прикрепляем светодиодную ленту к столу

Светодиодная лента должна быть равна периметру стола.

1. Измерьте ваш кофейный столик по длине и ширине, чтобы определить длину ленты. Имейте в виду, что лучше отрезать чуть меньше, чтобы конец ленты не выступал за край стола.
2. Разрежьте светодиодную ленту на правильные части. Размеры моего стола 100×50 см, поэтому я разрезал 3-метровую светодиодную ленту на два отрезка по 1 м и два отрезка по 45 см. (Возможно, вам и не понадобится уменьшать требуемую длину отрезков, это зависит от вашего кофейного столика).
3. Спаяйте отрезанные части. Вы уже знаете, как это сделать (см. раздел «Подготавливаем светодиодную ленту»).
4. Прикрепите светодиодную ленту к столу любым способом, в том числе с помощью горячего клея или кабельных стяжек. Метод, который вы выберете, зависит от вашего кофейного столика.
5. Подключите Arduino, кнопку и потенциометр, как показано на **рис. 3.26**.
 - Подключите светодиодную ленту к Arduino: пин GND на ленте идет к пину GND на Arduino. Контакт, помеченный как 5V, подключается к пину 5V на Arduino. Контакт DI подключается к пину 2, контакт CI подключается к пину 3.
 - Положительный вывод на кнопке подключается к пину RESET на Arduino, отрицательный — к GND.
 - Центральный пин на потенциометре подключается к пину A0 на Arduino, в то время как два других пина подключаются соответственно к пину GND и 5V, причем неважно, какой куда.

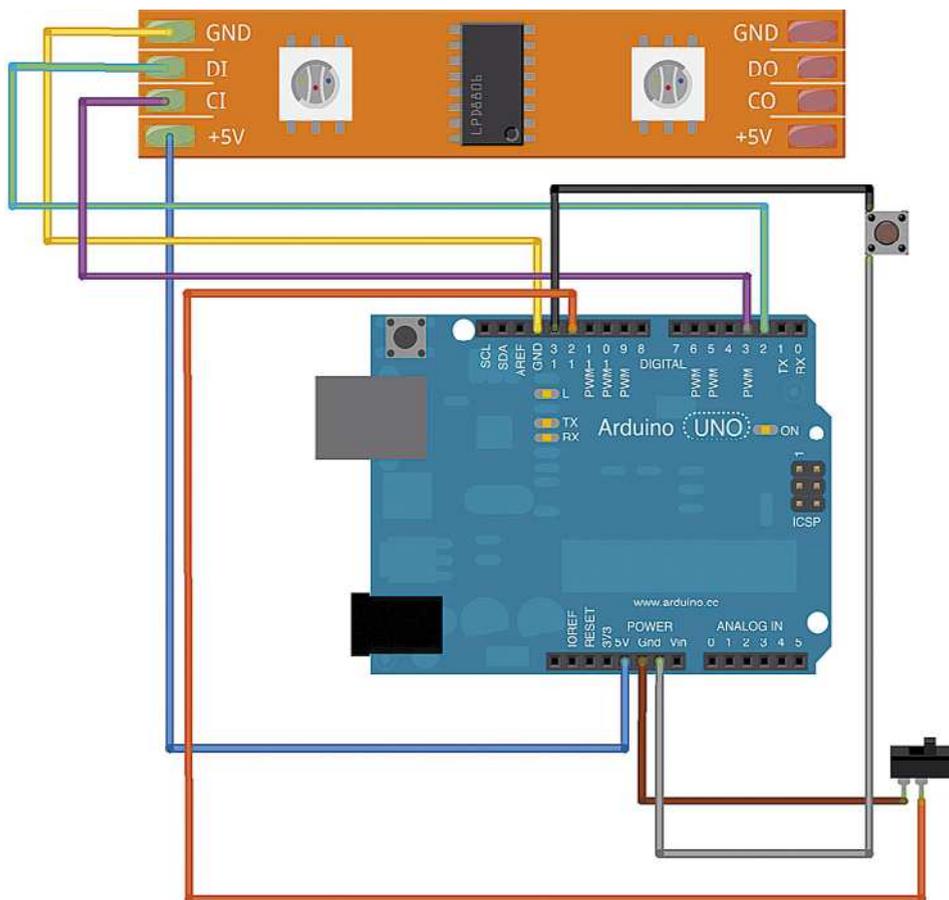


Рис. 3.26. Подключите светодиодную ленту так, как показано на рисунке

Изготавливаем корпус

Для этого проекта вы сможете сделать корпус из MicroRAX. Это серия довольно дешевых, но чрезвычайно прочных алюминиевых реек, которые используются для создания системных блоков и роботов. Вы можете приобрести набор, отдельные рейки фиксированной длины и более длинные рейки (до 2,5 м), которые можно нарезать на отрезки нужной длины. После того как вы сделаете основу из реек MicroRAX, добавьте панели из оргстекла, которые станут стенками корпуса (**рис. 3.27**).

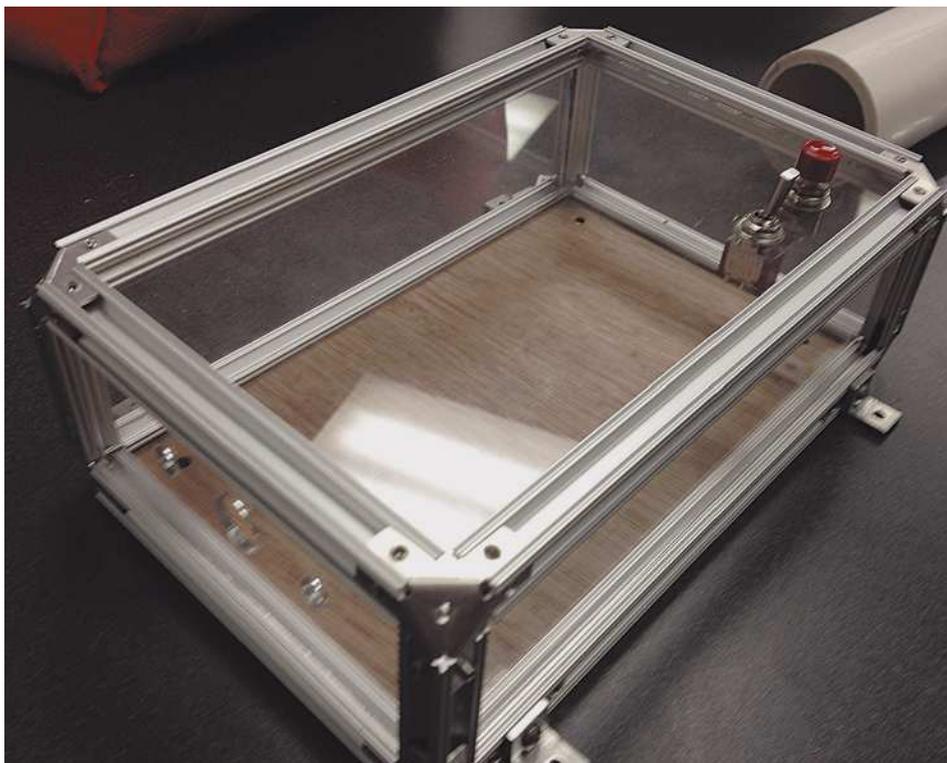


Рис. 3.27. Собранный корпус выглядит очень мило!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Используйте следующие комплектующие, чтобы построить свой корпус.

- Рейки MicroRAX:
 - на 40 мм, 4 шт.;
 - на 100 мм, 4 шт.;
 - на 160 мм, 4 шт.
- Угловые скобки (уголки), 8 шт.
- Прямоугольные соединительные скобки, 6 шт.
- Саморезы 3 мм.
- Панели из оргстекла или деревянные панели для боковых сторон корпуса.
- Изоляционные прокладки шириной 9 мм.
- Крепежные винты № 4-40 × 25 мм.



Рис. 3.28. Рейки MicroRAX выпускаются разной длины. Для своего корпуса я использовал рейки длиной 40, 100 и 160 мм

Сборка очень простая!

1. Соберите рейки с помощью прямоугольных соединительных скобок и саморезов диаметром 3 мм так, чтобы сформировать каркас. В дальнейшем вы добавите к каркасу листы оргстекла или древесины, чтобы сформировать днище, верх и боковые стороны корпуса.
2. Соберите рейки вместе. Я использовал рейки длиной 40, 100 и 160 мм (см. **рис. 3.28**), но вы можете использовать любую длину по своему усмотрению.
3. С помощью уголков (см. **рис. 3.29**) начните строить каркас коробки.
4. Вырежьте панели для боковых сторон. Свои панели я вырезал из оргстекла с помощью лазерного резака (**рис. 3.30**), но это вовсе не обязательно. Вы можете вырезать панели из дерева с помощью



Рис. 3.29. Уголки MicroRAX помогут легко соединить рейки различной длины

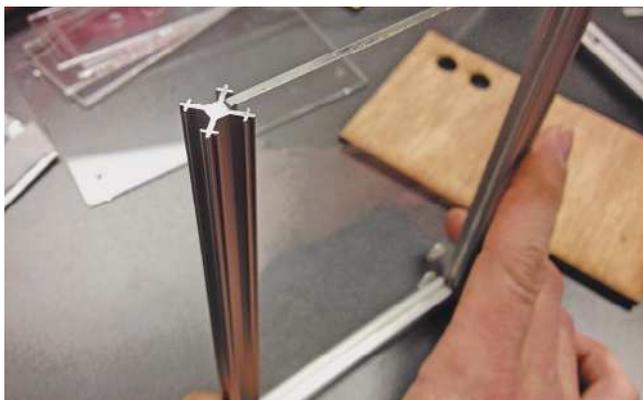


Рис. 3.30. Поместите панели в реечные пазы

обычной пилы. Сделайте панели того же размера, что и рейки, но добавьте 4–6 мм к длине и ширине панели, если вы используете прокладку толщиной 2,5 мм или тоньше, и 2–3 мм, если прокладка толщиной 5 мм. Если панели держатся неплотно и дребезжат, приклейте кусок резины в том месте, где панель отходит, чтобы зафиксировать ее.

5. При изготовлении каркаса не забудьте добавить угловые скобки (как показано на **рис. 3.31**), чтобы соединить корпус с кофейным столиком. Обратите внимание, что я сделал нижнюю панель корпуса из дерева, чтобы Arduino было проще подключить к корпусу. Завершите сборку других панелей и реек. Оставьте верхний проем открытым, чтобы потом можно было добавить электронику.



Рис. 3.31. Корпус начинает принимать свои очертания



Рис. 3.32. Крепежные винты надежно удерживают Arduino в правильном положении

6. Прикрепите Arduino к основанию корпуса. Если вы спланировали работу, можно заранее прорезать винтовые отверстия в панели лазером. Если вы, как и я, забыли это сделать, поступите следующим образом. Поместите Arduino на днище корпуса и карандашом сделайте отметки для отверстий на печатной плате Arduino. Просверлите отверстия. Затем пропустите крепежные винты № 4-40 через днище и добавьте прокладки шириной 9 мм. Поместите Arduino на винты и затяните гайки. Когда вы закончите, конструкция должна выглядеть так же, как на **рис. 3.32**.
7. После того как вы поместите содержимое в корпус, закройте крышку и слегка закрепите ее одним из винтов MicroRAX. Пока вы не закончите работу, не надо закручивать винты до конца. На **рис. 3.33** показано, как выглядит готовый корпус.

Управляем светодиодной лентой

С помощью потенциометра и кнопки можно выбрать один из восьми эффектов светодиодной ленты. Вот как это работает.

1. Потенциометр был отградуирован на восемь положений (определены значения от 1 до 8). (Я поясню, как это сделать, в **главе 5**.)

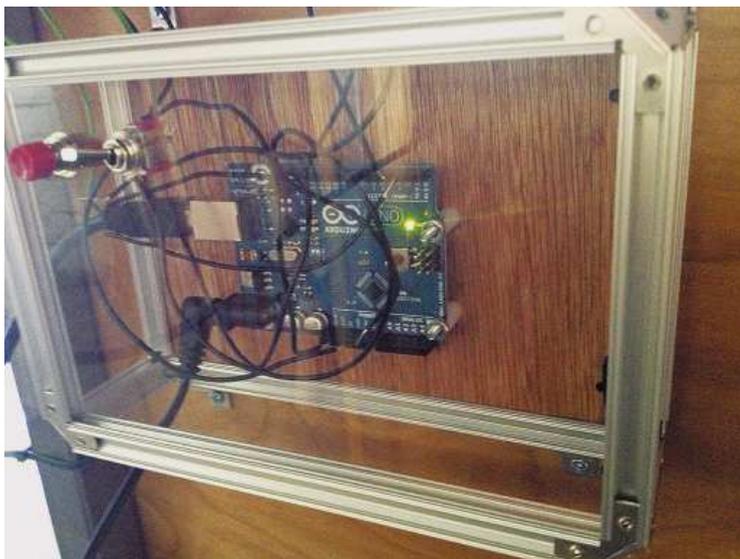


Рис. 3.33. Корпус, прикрепленный с обратной стороны кофейного столика

2. Когда скетч запускается, Arduino принимает определенное состояние и отображает нужный эффект, который выбран в настоящее время.
3. Однако, если вы хотите изменить эффект во время работы Arduino, вам нужно нажать на кнопку, которая сбрасывает программу, чтобы увидеть новый эффект светодиодной полосы. Это нужно сделать из-за того, что цикл скетча запущен, и программа не признает изменения потенциометра, пока вы не сделаете сброс.

Код для светодиодной ленты

Загрузите следующий код, чтобы «оживить» свой проект со светящимся столиком.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

О том, как загрузить код на Arduino, рассказано в **главе 2** «Макетирование» и **главе 5** «Программирование Arduino». Кроме того, вам понадобится библиотека LPD8806.h (подробнее о библиотеках см. в **главе 5**), которую можно загрузить, перейдя по ссылке <https://github.com/adafruit/LPD8806/blob/master/LPD8806.h>.

```
// этот скетч взят из примера кода LPD8806 от Adafruit
#include "LPD8806.h"
#include "SPI.h"

int pot1 = A1; int dataPin = 2; int clockPin = 3;
int toggleValue = 0; int toggle = 0;

// число 96 означает количество светодиодов на ленте,
// при необходимости измените число
LPD8806 strip = LPD8806(96, dataPin, clockPin);

void setup() {
    pinMode(pot1, INPUT);
    Serial.begin(9600);
    // включение светодиодной ленты
    strip.begin();

    // переключите режим, чтобы все светодиоды погасли
    strip.show();
}

void loop() {
    toggle = analogRead(pot1);
    int toggleStatus = map(toggle, 0, 1023, 0, 8);
    Serial.println(toggleStatus);

    switch(toggleStatus) {

        case 0:
            // перед запуском следующего эффекта удалите предыдущие данные
            for (int i=0; i < strip.numPixels(); i++) {
                strip.setPixelColor(i, 0);
            }
            break;

        case 1:
            // задайте следующие значения...
            colorChase(strip.Color(127,127,127), 20); // белый
            colorChase(strip.Color(127,0,0), 20); // красный
```

```
    colorChase(strip.Color(127,127,0), 20);    // желтый
    colorChase(strip.Color(0,127,0), 20);     // зеленый
    colorChase(strip.Color(0,127,127), 20);   // голубой
    colorChase(strip.Color(0,0,127), 20);     // синий
    colorChase(strip.Color(127,0,127), 20);   // пурпурный
break;

case 2:
    // сформируйте на ленте следующие цвета...
    colorWipe(strip.Color(127,0,0), 20);     // красный
    colorWipe(strip.Color(0, 127,0), 20);    // зеленый
    colorWipe(strip.Color(0,0,127), 20);     // синий
    colorWipe(strip.Color(0,0,0), 20);      // черный
break;

case 3:
    // эффект мерцания
    dither(strip.Color(0,127,127), 50); // голубой, медленно
    dither(strip.Color(0,0,0), 15);     // черный, быстро
    dither(strip.Color(127,0,127), 50); // пурпурный, медленно
    dither(strip.Color(0,0,0), 15);     // черный, быстро
    dither(strip.Color(127,127,0), 50); // желтый, медленно
    dither(strip.Color(0,0,0), 15);     // черный, быстро
break;

case 4:
    // загорание туда и обратно
    scanner(127,0,0, 30);                // красный, медленно
    scanner(0,0,127, 15);                // синий, быстро
break;

case 5:
    // эффект переливающейся волны
    wave(strip.Color(127,0,0), 4, 20);    // карамельная трость
    wave(strip.Color(0,0,100), 2, 40);    // ледяной
break;

case 6:
    // эффект радуги
    rainbowCycle(0); // позволяет совершить цикл достаточно быстро
break;

case 7:
    rainbowCycle(0);
break;
```

```

    case 8:
        // эффект мерцания
        dither(strip.Color(0,127,127), 50); // голубой, медленно
        dither(strip.Color(0,0,0), 15);    // черный, быстро
        dither(strip.Color(127,0,127), 50); // пурпурный, медленно
        dither(strip.Color(0,0,0), 15);    // черный, быстро
        dither(strip.Color(127,127,0), 50); // желтый, медленно
        dither(strip.Color(0,0,0), 15);    // черный, быстро
    break;
}
}

/* Вспомогательные функции */

// введите значение от 0 до 384, чтобы задать значение цвета;
// цвета переливаются от красного (r) к зеленому (g), затем синему (b)
// и снова к красному (r)

uint32_t Wheel(uint16_t WheelPos)
{
    byte r, g, b;
    switch(WheelPos / 128)
    {
        case 0:
            r = 127 - WheelPos % 128;    // красный угасает
            g = WheelPos % 128;          // зеленый загорается
            b = 0;                        // синий выключен
            break;
        case 1:
            g = 127 - WheelPos % 128;    // зеленый угасает
            b = WheelPos % 128;          // синий загорается
            r = 0;                        // красный выключен
            break;
        case 2:
            b = 127 - WheelPos % 128;    // синий угасает
            r = WheelPos % 128;          // красный загорается
            g = 0;                        // зеленый выключен
            break;
    }
    return(strip.Color(r,g,b));
}

// цикл задает цветовые оттенки, равномерно распределяющиеся по ленте
void rainbowCycle(uint8_t wait) {
    uint16_t i, j;

```

```
for (j=0; j < 384 * 5; j++) { // круг включает 5 циклов из всех
                               // 384 оттенков
for (i=0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i,Wheel(((i*384/strip.numPixels()+j)%384));
    }
    strip.show(); // считывание всех пикселей
delay(wait);
}
}

// выявление одной точки по всей ленте
void colorChase(uint32_t c, uint8_t wait) {
    int i;

    // начинаем с выключения всех пикселей
for(i=0; i<strip.numPixels(); i++) strip.setPixelColor(i, 0);

    // затем включаем пиксели поочередно
for(i=0; i<strip.numPixels(); i++) {
    strip.setPixelColor(i, c); // добавить следующий пиксель
    strip.show(); // обновить состояние светодиодов
    strip.setPixelColor(i, 0); // выключить пиксель,
                               // но не обновлять!

    delay(wait);
}
    strip.show(); // обновить, чтобы выключить последний пиксель
}

// постепенно заполняем ленту точками
void colorWipe(uint32_t c, uint8_t wait) {
    int i;

    for (i=0; i < strip.numPixels(); i++) { strip.setPixelColor(i, c);
    strip.show();
        delay(wait);
    }
}

// посредством «упорядоченного сглаживания цветовых переходов»
// происходит постепенное заполнение каждого пикселя,
// это похоже на произвольное сверкание,
// но на самом деле все происходит в определенном порядке
void dither(uint32_t c, uint8_t wait) {

    // определяем старший бит, чтобы обозначить индекс цвета пикселя
    int hiBit = 0;
```

```

int n = strip.numPixels() - 1;
for(int bit=1; bit < 0x8000; bit <<= 1) {
    if(n & bit) hiBit = bit;
}

int bit, reverse;
for(int i=0; i<(hiBit << 1); i++) {
    // меняем биты на i, чтобы создать упорядоченное сглаживание
    // цветовых переходов
    reverse = 0;
    for(bit=1; bit <= hiBit; bit <<= 1) {
        reverse <<= 1;
        if(i & bit) reverse |= 1;
    }
    strip.setPixelColor(reverse, c); strip.show();
    delay(wait);
}
delay(250); // задерживаем отображение на ¼ с
}

// «Сканер Ларсона» = перескакивающий световой эффект Cylon/KITT
void scanner(uint8_t r, uint8_t g, uint8_t b, uint8_t wait) {
    int i, j, pos, dir;

    pos = 0;
    dir = 1;

    for(i=0; i<((strip.numPixels()-1) * 8); i++) {
        // задайте 5 пикселей с центральной позицией pos.
        // Функция setPixelColor() отбросит ненужные пиксели на концах
        // ленты, не беспокойтесь. По краям мы сделаем цвета более
        // тусклыми для лучшего эффекта
        strip.setPixelColor(pos - 2, strip.Color(r/4, g/4, b/4));
        strip.setPixelColor(pos - 1, strip.Color(r/2, g/2, b/2));
        strip.setPixelColor(pos, strip.Color(r, g, b));
        strip.setPixelColor(pos + 1, strip.Color(r/2, g/2, b/2));
        strip.setPixelColor(pos + 2, strip.Color(r/4, g/4, b/4));

        strip.show();
        delay(wait);
        // можно было бы просто убрать крайний пиксель,
        // но гораздо проще удалить всё и создать новое.
        for(j=-2; j<= 2; j++)
            strip.setPixelColor(pos+j, strip.Color(0,0,0));
        // эффект отражения от краев
        pos += dir;
    }
}

```

```
        if(pos < 0) {
            pos = 1;
            dir = -dir;
        } else if(pos >= strip.numPixels()) {
            pos = strip.numPixels() - 2;
            dir = -dir;
        }
    }
}

// эффект синусоиды
#define PI 3.14159265
void wave(uint32_t c, int cycles, uint8_t wait) {
    float y;
    byte r, g, b, r2, g2, b2;

    // нужно разложить цвет на r-, g-, b-элементы
    g = (c >> 16) & 0x7f;
    r = (c >> 8) & 0x7f; b = c & 0x7f;

    for(int x=0; x<(strip.numPixels()*5); x++)
    {
        for(int i=0; i<strip.numPixels(); i++) {
            y = sin(PI * (float)cycles * (float)(x + i) / (float)strip.numPixels());
            if(y >= 0.0) {
                // максимумы синусоиды белого цвета
                y = 1.0 - y; // переводим Y в 0.0 (точка максимума)
                           // и 1.0 (середина)
                r2 = 127 - (byte)((float)(127 - r) * y);
                g2 = 127 - (byte)((float)(127 - g) * y);
                b2 = 127 - (byte)((float)(127 - b) * y);
            } else {
                // минимумы синусоиды черного цвета
                y += 1.0; // переводим Y в 0.0 (точка минимума) и 1.0
                        // (середина)
                r2 = (byte)((float)r * y);
                g2 = (byte)((float)g * y);
                b2 = (byte)((float)b * y);
            }
            strip.setPixelColor(i, r2, g2, b2);
        }
        strip.show();
        delay(wait);
    }
}
```

В следующей главе

Вы освоили макетирование и пайку. Теперь пришло время использовать эти знания и навыки в настоящем деле! В **главе 4** «Настройка беспроводного соединения» вы узнаете о том, как быстро создать беспроводную сеть. Затем вы используете эти знания при создании простого дверного звонка для вашего дома.

Настройка беспроводного соединения

В этой главе рассматриваются устройства беспроводных сетей, позволяющие работать совместно двум или более Arduino. Главным среди них является XBee — совместимый с Arduino радиомодуль, способный объединить микроконтроллеры в единую сеть. На **рис. 4.1** представлен один из моих проектов — LEGO-робот, который подключен к оснащённому модулем XBee Arduino. Вы можете узнать, как сделать этого робота, в моей книге «Сделай сам: проекты с LEGO и Arduino» («Make: Lego and Arduino Projects»). После изучения радиомодуля XBee вы перейдете к третьему проекту — созданию беспроводного дверного звонка!

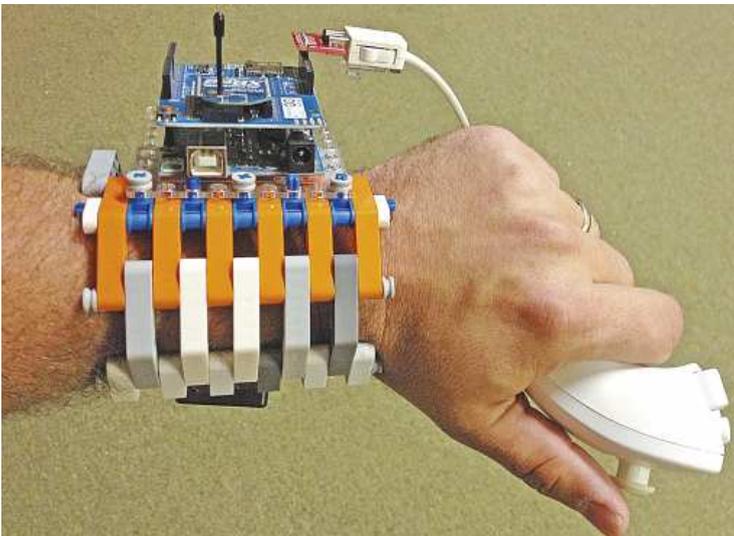


Рис. 4.1. Этот браслет, оборудованный радиомодулем XBee, позволяет управлять роботом дистанционно

Радиомодули XBee

Радиомодули XBee (рис. 4.2) основаны на технологии *ZigBee* — стандартном протоколе, который создает сеть из нескольких узлов с помощью последовательной беспроводной передачи данных. Это означает, что одновременно отправляется только один бит информации (0 или 1), что делает передачу медленной, но легко настраиваемой. Обычно протокол *ZigBee* используется в бытовой технике, поэтому изучение тонкостей работы радиомодуля поможет вам создать собственный беспроводной подъемный механизм для штор или выключателя!

Радиомодуль XBee по определению служит одним из способов связи для Arduino, с помощью которого можно организовать эффективную совместную работу двух и более платформ. В настоящее время выбор радиомодулей разных модификаций настолько велик, что вы можете в них потеряться. Давайте сосредоточимся только на конкретных четырех моделях XBee:

- XBee;
- XBee Pro;
- XBee серии 1;
- XBee серии 2.

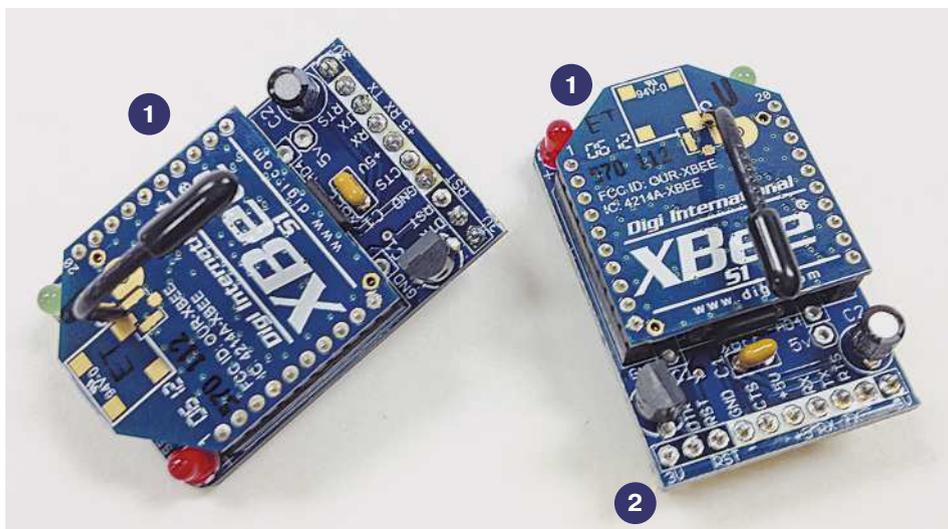


Рис. 4.2. Два радиомодуля XBee серии 1, прикрепленные к переходным платам Adafruit:

1 — контроллеры XBee; 2 — переходные платы

XBee или XBee Pro?

Сначала нужно определиться, хотите ли вы работать с базовым радиомодулем (XBee) или профессиональным устройством (XBee Pro). Различие между ними заключается в передающей мощности. Передаваемая мощность базовых XBee составляет 1 мВт (одна тысячная ватта), в то время как профессиональные радиомодули имеют мощность от 60 до 100 мВт, значительно расширяя дальность связи. На сколько именно? Это зависит от многих факторов, в том числе электромагнитных помех, типа антенны, физических препятствий и рельефа местности.

Компания *Digi International*, производитель радиомодулей XBee, предлагает различные модели в зависимости от требуемой зоны покрытия. Самая простая модель XBee мощностью 1 мВт рассчитана на распространение радиосигнала в радиусе 30 м в помещении и 100 м на открытой местности. Компания утверждает, что модель XBee Pro хорошо подходит для передачи сигнала на расстояние 100 м в закрытом помещении и на впечатляющие 1500 м на открытой местности. Конечно, для такой дальности связи необходимо открытое пространство без холмов и других физических преград. Любое препятствие снизит дальность эффективного действия радиосигнала.

Если вам не нужна дальность в 1,5 км, лучше отказаться от профессиональной модели, потому что она стоит дороже.

Серия 1 или серия 2?

Второе соображение относительно выбора радиомодуля XBee заключается в том, какой именно вид сети вы хотите настроить. Компания *Digi International* обозначила эту разницу в моделях «XBee серии 1» и «XBee серии 2».

- **Серия 1.** Предлагает простую настройку сети, которую не надо устанавливать специально. В принципе, каждый радиомодуль серии 1 общается с другими модулями этой серии в пределах определенного диапазона. Такая конфигурация известна как *сотовая сеть*. Это самый простой способ начать работу с беспроводной технологией.

Если вы хотите направить данные в один модуль, то должны использовать программное обеспечение для установки идентификатора на время передачи и приема сигнала. Звучит устрашающе, но это так же просто, как добавить одну цифру. Скажем, вы хотите направить данные в узел 5. Для этого вы добавляете цифру 5 в начале командной строки, а другие узлы будут игнорировать эту передачу.

- **Серия 2.** Более надежна, поскольку предлагает в дополнение к настройкам серии 1 создавать:
 - более сложные сети с узлами-координаторами, способными выдавать команды;
 - роутеры, которые отправляют и получают данные;
 - конечные устройства, которые только получают данные.

Вместе с тем, наличие этих замечательных возможностей вовсе не означает, что вы можете сразу использовать устройство. Прежде чем запустить радиомодули серии 2, необходимо их настроить. А радиомодули серии 1 можно включить в работу сразу, как только вы распакуете коробку! Кроме того, в серии 2 используется другой беспроводной протокол, из-за чего модули разных серий технически несовместимы (даже не пытайтесь пробовать).

Переходные платы для радиомодуля XBee

Радиомодули XBee удобны в обращении, но нужно уделить им немного внимания, прежде чем использовать в проекте с Arduino. Особенность заключается в том, что расстояние между контактами в этих радиомодулях составляет 2 мм, в то время как в Arduino — 2,5 мм. Решением служит небольшая печатная плата под названием «*переходная плата*». Другими словами, надо сделать микроплату, через которую можно подключить XBee к Arduino.

Самый простой вариант — печатная плата, с правой стороны которой расположены контакты. Однако более надежные переходные платы (см. **рис. 4.2**) имеют стабилизатор напряжения и индикаторы состояния, которые предохраняют радиомодуль от повреждения.

Компоненты радиомодуля XBee

Если вы посмотрите на радиомодуль XBee (см. **рис. 4.3**), то увидите пластину синего цвета размером с почтовую марку, с металлическими штифтами, выступающими снизу. Сверху расположена антенна. О том, как добавить ее к переходной плате, нужно рассказать более подробно, поэтому давайте познакомимся с различными компонентами устройства.

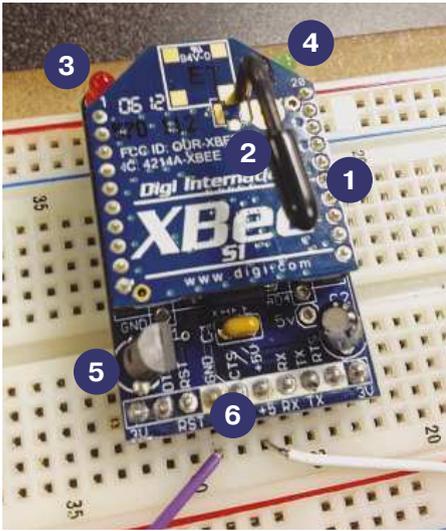


Рис. 4.3. Радиомодуль XBee с переходной платой прикреплен к макетной плате. Обратите внимание, что пины 5V и GND уже подключены к правильным клеммам

1. **Пины.** На рисунке вы можете увидеть, как расположены соединительные пины XBee. По ним подается питание, а также осуществляются получение и передача данных с Arduino. Пины подключаются к гнездам на переходной плате. Обратите внимание, что промежутки между пинами не совпадают с расстоянием на макетных платах.
2. **Антенна.** Вариант исполнения антенны зависит от типа XBee. Я думаю, что проволочная антенна подходит лучше всего, она жесткая и не гнется.
3. **Индикатор питания.** Загорается, когда плата подключена к источнику питания.
4. **Индикатор данных.** Мигает в момент прохождения данных через радиомодуль XBee.
5. **Стабилизатор питания.** Состоит из конденсатора и транзисторов, которые регулируют питающее напряжение радиомодуля. К сожалению, повышение напряжения может вывести устройство из строя. Однако стабилизатор защищает от скачков напряжения в момент переключения тока.
6. **Пины под макетную плату.** В отличие от пинов, подключающих XBee к переходной плате, эти пины расположены правильно. Они маркированы, поэтому вы можете сразу увидеть, для чего предназначен каждый контакт!

Альтернативные беспроводные модули

Наверное, вас не удивит, что радиомодуль XBee — не единственный в своем роде. Существует несколько отличных альтернатив, которые можно приобрести и использовать в проекте.

Freakduino Chibi

Freakduino Chibi — это радиомодуль, созданный токийским электронщиком Акиба (настоящее имя Крис Ванг). *Chibi* — это гибрид отдельной платы для контроллера и беспроводного модуля, Акиба объединил их. *Chibi* совместим с Arduino и использует тот же диапазон, что и радиомодуль XBee.

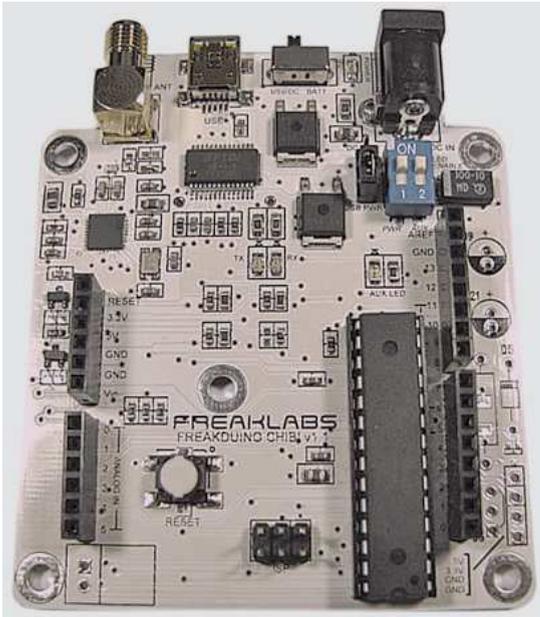


Рис. 4.4. По сути, *Freakduino Chibi* от *Freaklabs* — это Arduino со встроенным модулем беспроводной связи

JeeNode от JeeLabs

Основанный на той же концепции, что и *Chibi*, радиомодуль *JeeNode* включает микроконтроллер ATmega328P (тот самый микрочип памяти Arduino), а также встроенный беспроводной модуль. Радиомодули *JeeNode* небольшого размера и имеют меньше возможностей по сравнению с *Chibi*, но именно из-за конструктивных параметров и простоты использования у *JeeNode* много поклонников.

СОВЕТ**ВЫБЕРИТЕ РАДИОМОДУЛЬ XBEE СЕРИИ 1**

Существует такое количество радиомодулей, что я еще долго мог бы рассказывать о них, но, может быть, хватит? Я предлагаю остановиться на XBee, но не XBee Pro, а XBee серии 1. Это удивительно простой способ добавить беспроводное устройство в ваш проект, не тратя много денег. Кроме этого, работа с XBee серии 1 спасет вас от разочарований, которые могут постигнуть при работе со слишком сложным радиомодулем.

Проект: беспроводное включение светодиода

Ой, беспроводные радиомодули! Начинать работу с ними немного боязно. На самом деле бояться нечего, и я докажу это. Давайте создадим простую сеть (см. **рис. 4.5**), которая позволит общаться двум Arduino. В этом проекте вы соберете две одинаковые конструкции, каждая из которых состоит из Arduino и XBee вместе с кнопкой и светодиодом. При нажатии на кнопку на одной сборке светодиод загорится на другой, и наоборот! Вы сможете убедиться в том, что этот проект даст вам хороший старт на пути к созданию беспроводного дверного звонка — основного проекта этой главы.

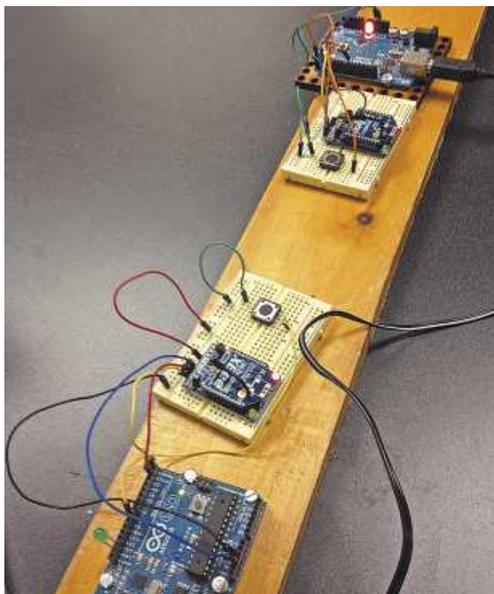


Рис. 4.5. Управляйте светодиодами с помощью Arduino с радиомодулем XBee

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Вы должны собрать два комплекта, поэтому деталей должно быть в два раза больше!

- Arduino, 2 шт.
- Радиомодуль XBee, 2 шт.
- Переходная плата, 2 шт.
- Кнопка, 2 шт.
- Макетная плата, 2 шт.
- Светодиод, 2 шт.
- Соединительные провода.

Следуйте этой пошаговой инструкции, чтобы собрать тестовую платформу радиомодуля XBee.

1. **Припаяйте переходные платы XBee**, если вы еще этого не сделали. В зависимости от вашего комплекта это может означать некоторые соединения. Однако в других наборах, возможно, нужно будет припаять еще и светодиоды, конденсаторы, и т. д.
2. **Подключите радиомодули XBee к переходным платам.** Обычно это означает размещение штифтов XBee в соответствующих гнездах на переходной плате. Просто следуйте инструкциям, которые входят в комплект набора.
3. **Подключите переходные платы с радиомодулями к макетным платам.** На **рис. 4.6** показано, как расположить плату.
4. **Подключите кнопки, светодиоды и соединительные провода** следующим образом (также см. **рис. 4.6**).
 - А. Пин GND на XBee — к пину GND на макетной плате. Подключите шину заземления на макетной плате к пину GND на Arduino.
 - Б. Пин +5V на XBee — к пину 5V на Arduino.
 - В. Пин TX на XBee — к пину RX на Arduino.
 - Г. Пин RX на XBee — к пину TX на Arduino
 - Д. Подключите вывод кнопки к пину 8 на Arduino, другой конец — к шине заземления.

В результате у вас должны получиться две одинаковые сборки. Если вы загрузите на них код Arduino, они должны работать одинаково. После того как вы создадите сеть, вы сможете расширить ее за счет третьего или четвертого комплекта, и все они будут работать так, как вы хотите. Если вы нажмете кнопку на одном комплекте, светодиоды загорятся на

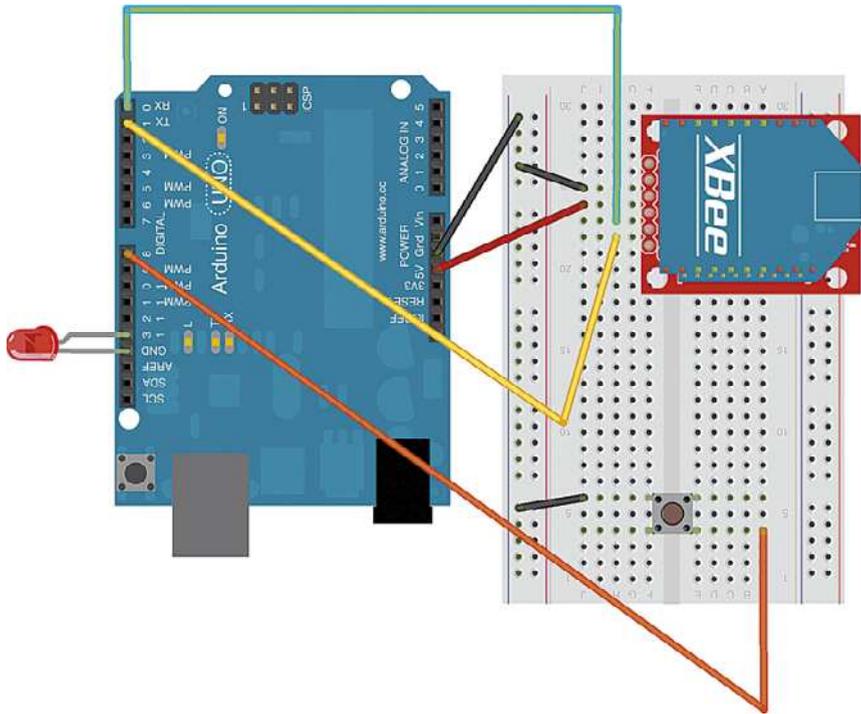


Рис. 4.6. На схеме показано, как собрать тестовые модули XBee

остальных! Конечно, это непрактично, но главное, что этот пример показывает, как легко можно создать XBee-сети.

Код для беспроводного светодиода

Загрузите следующий код на обе платы Arduino. Помните, что модули идентичны, вплоть до программного обеспечения. Если вы не помните, как загружать скетчи на Arduino, то обратитесь к **главе 5** «Программирование Arduino».

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#include <Wire.h>
const int buttonPin = 8;
const int ledPin = 13;
int buttonState = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void process_incoming_command(char cmd)
{
  int speed = 0;
  switch (cmd)
  {
    case '1':
    case 1:
      digitalWrite(ledPin, LOW);
      break;
    case '0':
    case 0:
      digitalWrite(ledPin, HIGH);
      break;
  }
}

void loop() {
  if (Serial.available() >= 2)
  {
    char start = Serial.read();
    if (start != '*')
    {
      return;
    }
    char cmd = Serial.read();
    process_incoming_command(cmd);
  }

  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    Serial.write('*');
    Serial.write(1);
  }
  else {
    Serial.write('*');
    Serial.write(0);
  }
  delay(50); // ограничивает периодичность обновления
}
```

⚙️ Проект: беспроводной дверной звонок

Теперь вы можете применить полученные знания о радиомодулях XBee на практике и приняться за более сложный проект: создание беспроводного звонка. На **рис. 4.7** изображена кнопка дверного звонка, а на **рис. 4.8** — блок зуммера, который спрятан подальше от глаз внутри корпуса.

Разумеется, вы можете сказать, что такие звонки уже существуют! Их можно купить в любом хозяйственном магазине. Однако этот звонок вы сделаете своими руками! А самое главное, вы станете ближе к Arduino, будете обращаться с ним, как уверенный пользователь. Вы сможете изменить эту платформу, чтобы она стала вашим уникальным, неповто-

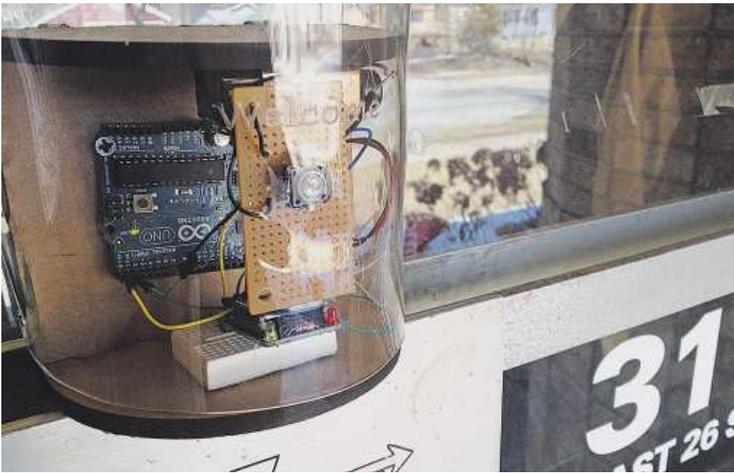


Рис. 4.7. Дверной звонок ожидает первых посетителей!



Рис. 4.8. Зуммер надежно прикреплен к фанере

римым Arduino! Что, если ваш Arduino сыграет музыкальную мелодию вместо банального трезвона, когда кто-то нажмет на кнопку?

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Так же, как в проекте «Беспроводное включение светодиода», описанном ранее в этой главе, вам понадобятся два Arduino, связанных беспроводной сетью. Однако в этом проекте один Arduino ожидает нажатия кнопки, в то время как второй включает зуммер по сигналу от первого.

- Arduino UNO, 2 шт.
- Беспроводной радиомодуль XBee, 2 шт.
- Переходная плата для радиомодуля XBee, 2 шт.
- Макетная мини-плата, 2 шт.
- Кнопка.
- Резистор 330 Ом.
- Зуммер.
- Клемма для батарейки 9 В.
- Разъем 9 В с вилкой «бочка».
- МДФ толщиной 6 мм для задней и боковых стенок корпуса.
- Оргстекло толщиной 5 мм для передней стенки корпуса.
- Болты № 4-40 длиной 25 мм.
- Клеевой термопистолет.

Кнопка

Кнопка, которая понадобится вам в проекте (см. **рис. 4.9**), — достаточно занимательная деталь, потому что у нее есть шесть выводов: по две пары положительных и отрицательных ножек, поэтому вы могли бы включать одновременно две цепи, когда кнопка нажата. Последние два вывода (белые ушки) нужны для питания светодиода. Убедитесь, что резистор присоединен к положительному контакту питания светодиода, чтобы он случайно не вышел из строя. В этом проекте я использую резистор 330 Ом.

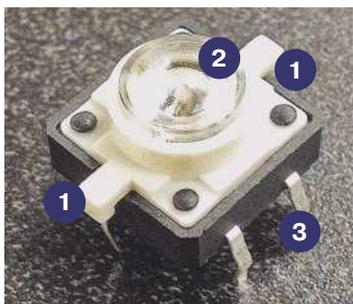


Рис. 4.9. Кнопка, которую вы используете в этом проекте, имеет шесть контактов:

1 — светодиодные контакты; 2 — светодиодная кнопка; 3 — переключатели (два расположены с противоположной стороны)

Инструкции для подключения дверного звонка

В проекте используются две платы Arduino, оснащенные радиомодулями XBee и переходными платами. У одного Arduino есть кнопка, у другого — зуммер, оповещающий вас о том, что кто-то пришел. Давайте начнем!

Модуль кнопки

Начнем с модуля кнопки (см. **рис. 4.10**), который включает следующие компоненты:

- А — батарейка 9 В;
- Б — радиомодуль XBee;
- В — макетная мини-плата;
- Г — Arduino UNO;
- Д — кнопка;
- Е — перфорированная панель.

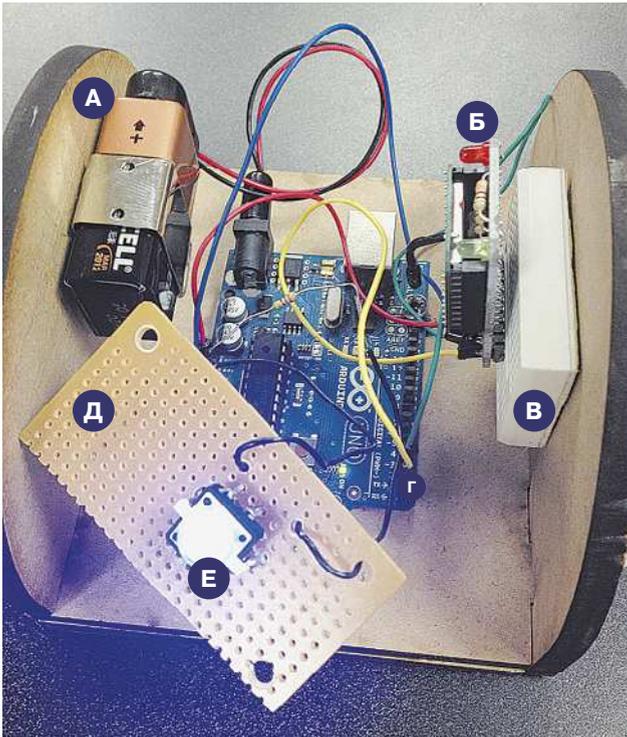


Рис. 4.10. Внешний вид модуля кнопки до установки передней панели

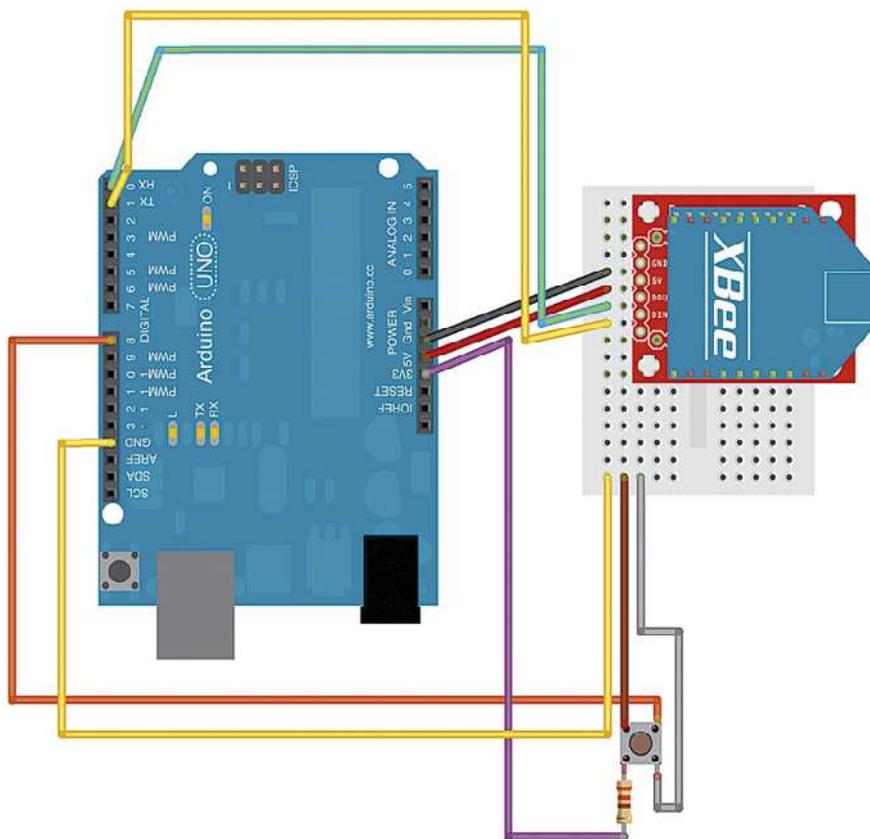


Рис. 4.11. Основные части модуля кнопки — кнопка, Arduino и радиомодуль

Теперь соберите эти части вместе, как показано на **рис. 4.11**. Вы можете следовать пошаговой инструкции.

1. Подключите радиомодуль Xbee с переходной платой к макетной мини-плате.
2. Подключите пин 5V на Xbee к пину 5V на Arduino, пин TX на Xbee к RX, пин RX на Xbee к TX и пин GND на Xbee подключите к любому свободному пину GND на Arduino.
3. Подключите один вывод кнопки к пину 8, а другой — к GND (я воспользовался макетной платой для соединения проводов GND с кнопкой).
4. Припаяйте один конец резистора и соединительный провод к выводу питания светодиода на кнопке, а другой его конец подключите к пину 3V3 Arduino. Второй отрицательный вывод светодиода идет к GND.

Модуль зуммера

Следующий этап — это сборка компонентов для модуля зуммера (см. **рис. 4.12**):

- А — Arduino UNO;
- Б — макетная мини-плата;
- В — радиомодуль XBee;
- Г — зуммер.

Соберите модуль, как на **рис. 4.13**.

1. Подключите радиомодуль XBee с переходной платой к макетной мини-плате.

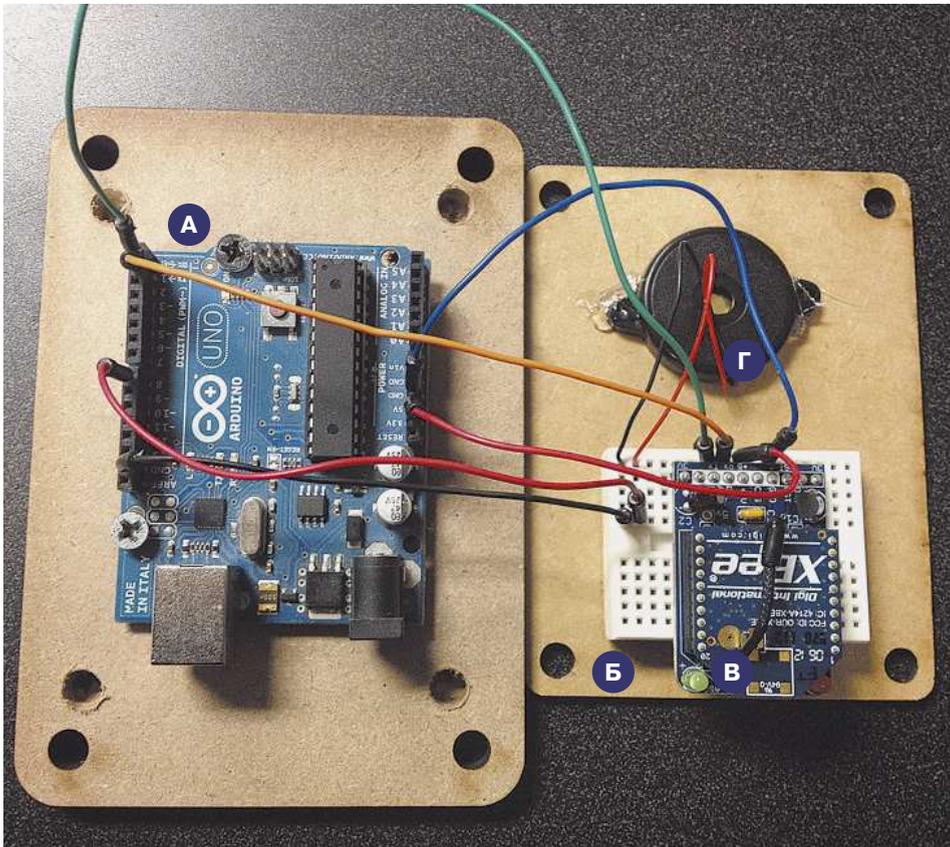


Рис. 4.12. Модуль зуммера в открытом виде. Внешние отверстия в корпусе предназначены для крепления к стене

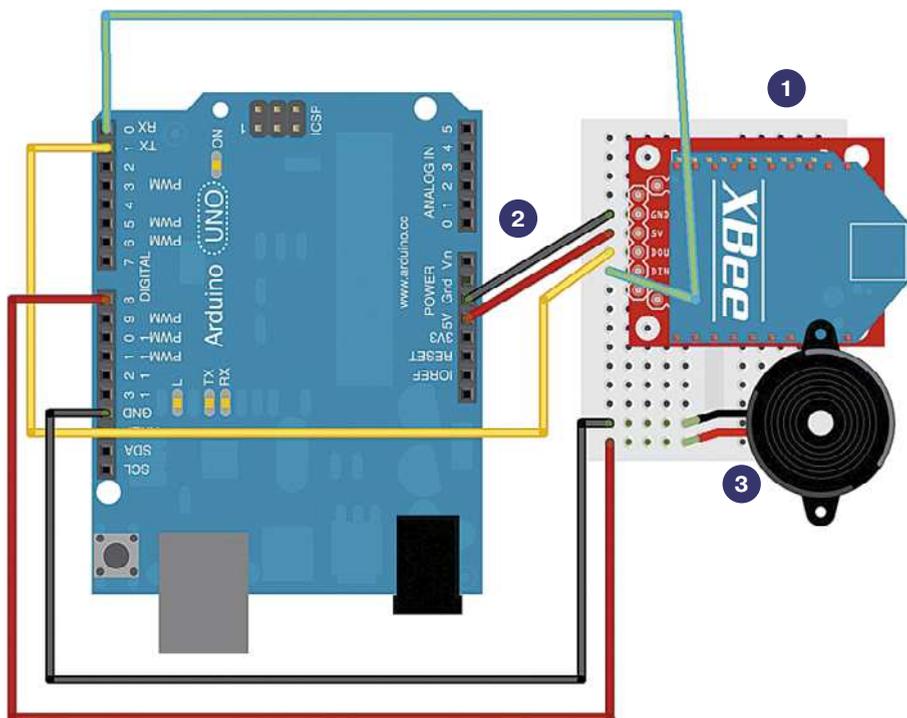


Рис. 4.13. Модуль зуммера состоит из Arduino, XBee и зуммера

2. Подключите пин 5V на XBee к пину 5V на Arduino, пин TX на XBee к RX, пин RX на XBee к TX и пин GND на XBee подключите к любому свободному пину GND на Arduino.
3. Подключите провода зуммера к макетной плате так, как показано на **рис. 4.13**. Если хотите, можете подключить провода напрямую к Arduino. Если вы выбрали этот способ, подключите красный провод к пину 8, а черный — к любому свободному пину GND.
4. Чтобы запитать модуль зуммера, используйте совместимый с Arduino блок питания или батарейку 9 В.

Изготавливаем корпуса для дверного звонка

В этом проекте вам нужно построить два корпуса: наружный будет находиться за дверью (см. **рис. 4.14**) и защищать элементы (я называю его «водонепроницаемый»), в то время как внутренний («домашний») корпус сделан «для красоты».

Корпус для модуля кнопки

Модуль кнопки находится с наружной стороны двери. Нажмите на кнопку, чтобы звонок зазвонил! Все, что вам нужно, чтобы сделать корпус, — это коробка с отверстием для кнопки, но я покажу вам, как сделать собственный корпус. Корпус, изготовленный мной, состоит из листа оргстекла, который я согнул, предварительно нагрев, а затем обернул им металлическую трубу, чтобы сформировать полусферу. Я присоединил ее к деревянному основанию (см. **рис. 4.14**), тем самым завершив изготовление корпуса. Вот пошаговая инструкция.

1. С помощью лазерного резака вырежьте верх, основание и заднюю часть корпуса из древесноволокнистой плиты средней плотности (МДФ) толщиной 6 мм. Если у вас нет лазерного резака, вы можете сделать коробку из обрезков древесины, использовать любой ненужный контейнер или купить готовый корпус.
2. Вырежьте лазерным резаком переднюю часть из оргстекла толщиной 5 мм. (Если вы хотите воспользоваться готовым дизайном коробки, который я использовал для создания деревянного основания и прозрачной передней части корпуса, обратитесь по ссылке <https://github.com/n1/Arduino-For-Beginners>.)
3. Склейте вместе деревянные части (верх, основание и боковины). Возможно, вам захочется их покрасить!



Рис. 4.14. Наружный корпус сделан из оргстекла на деревянном основании

4. Прикрепите электронику так, как показано на **рис. 4.10** и **4.12**. Прикрепите Arduino саморезами № 4-40 и приклейте с помощью клеевого термопистолета зуммер, комплект батарей и макетную мини-плату. Если вы не пользуетесь лазерным резаком, необходимо просверлить в оргстекле отверстия. Возможно, сначала вам нужно будет потренироваться на листе бумаги (сделайте разметку отверстий на бумаге и наложите ее на оргстекло).
5. Согните переднюю панель из оргстекла так, как описано в следующем разделе «Сгибание оргстекла».
6. Укрепите оргстекло так, чтобы можно было нажать кнопку через отверстие в пластике.
7. Установите устройство за дверью и с нетерпением ожидайте первого гостя!

Сгибание оргстекла

Для изготовления наружного корпуса модуля кнопки согните оргстекло путем нагревания (**рис. 4.15**), придав ему нужную форму. Эта задача легко осуществима, потому что не требует ничего необычного или редкого.



Рис. 4.15. Сгибание оргстекла — легкий процесс, который дает прекрасный результат!

Оргстекло (также известное как плексиглас) легко нагревать и формировать. При достижении правильной (не слишком высокой или слишком низкой) температуры оргстекло можно согнуть. При повышении температуры оргстекло размягчается. В этот момент его можно согнуть, как вы того хотите, и дать остыть в новой форме!

Для того чтобы начать работу, вам нужны три вещи.

1. **Оргстекло.** Я предлагаю стекло толщиной 3 мм, хотя вам, возможно, повезет с более толстым материалом.
2. **Форма.** Это поверхность, на которой будет охлаждаться и затвердевать горячее оргстекло. Вам необходима поверхность с той же кривизной, которая будет у готового изделия. Самая простая форма из всех — это край стола. Чтобы сформовать переднюю панель из оргстекла, которую вы видели на **рис. 4.7**, я использовал трубу соответствующего диаметра. Вы также можете найти трубу, профиль которой соответствует ожидаемой кривизне передней панели.
3. **Источник тепла.** Тепловые пушки (**рис. 4.16**) и пропановые факелы — самые распространенные инструменты для нагревания, хотя вы можете приобрести в магазине специальные полоски для нагревания оргстекла. Наконец, вы можете нагреть пластик в духовке. Этот метод не для слаонервных. Избегайте соприкосновения с ра-



Рис. 4.16. Тепловая пушка размягчает оргстекло

зогретым оргстеклом, иначе можете получить ожог второй или даже третьей степени.

Несмотря на то что для формования оргстекла может сгодиться любая термостойкая поверхность (если хотите, можете сделать собственную модель из куска дерева), в некоторых отношениях край стола — самое простое решение. Он твердый, поэтому пластик отлично сгибается под собственным весом, а форма поверхности стола позволяет сделать довольно точный изгиб в 90 градусов.

Чтобы согнуть пластик, используя край стола, выполните следующие действия.

1. Расположите оргстекло на краю стола именно там, где вы хотите его согнуть (**рис. 4.17**). Вам определенно захочется положить стекло под пресс, чтобы оно не двигалось. Когда стекло достигает нужной температуры, сила притяжения начинает тянуть мягкий акрил вниз, как показано на **рис. 4.17**.
2. Расположите акриловую заготовку, как задумали, и быстро обработайте ее, потому что после того, как стекло охладится, оно станет опять хрупким. Не пытайтесь согнуть его без повторного нагрева!



Рис. 4.17. Как только оргстекло нагревается, оно начинает гнуться

Корпус для модуля зуммера

Желательно не использовать пластик при изготовлении корпуса для зуммера. Кому нужен пластик в доме? Вместо этого, вы можете использовать простую компоновку из деревянных панелей, скрепленных с помощью болтов. Я вырезал лазерным резаком две дощечки, одна чуть больше другой, и просверлил ручной дрелью четыре дополнительных отверстия, как показано на **рис. 4.18**.

Затем я соединил две части с помощью латунных болтов № 10-24 длиной 6 см с медными шайбами и гайками. Этот корпус значительно легче сделать, чем предыдущий, и смотрится он великолепно!



Рис. 4.18. Я использовал лазер для вырезания боковин

Код для беспроводного дверного звонка

Загрузите код на плату Arduino. Если у вас возникнут трудности с загрузкой скетчей на Arduino, то обратитесь к **главе 5**. Перед этим вы можете загрузить код по ссылке <https://github.com/n1/Arduino-For-Beginners>.

Код для модуля кнопки

Скетч модуля кнопки состоит из цикла с ожиданием нажатия кнопки и выдачи радиосигнала после ее нажатия.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#include <Wire.h>

const int buttonPin = 8;
int buttonState = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  if (Serial.available() >= 2)
  {
    char start = Serial.read();
    if (start != '*')
    {
      return;
    }
    char cmd = Serial.read();

    buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH) {
      Serial.write('*');
      Serial.write(1);
    }
  }
}
```

```
else {
    Serial.write('*');
    Serial.write(0);
}
delay(50); // ограничивает периодичность обновления
}
```

Код для модуля зуммера

Код выглядит так же просто. Цикл отслеживает последовательный трафик и при обнаружении радиоимпульса от модуля кнопки выдает звуковой сигнал.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#include <Wire.h>

const int buzzerPin = 13;

void setup()
{
    Serial.begin(9600);
    pinMode(buzzerPin, OUTPUT);
}

void process_incoming_command(char cmd)
{
    int speed = 0;
    switch (cmd)
    {
        case 1:
            digitalWrite(buzzerPin, LOW);
            break;
        case 0:
            digitalWrite(buzzerPin, HIGH);
            break;
    }
}

void loop() {
    if (Serial.available() >= 2)
    {
        char start = Serial.read();
        if (start != '*')
```

```
    {
        return;
    }
    char cmd = Serial.read();
    process_incoming_command(cmd);
}

delay(50); // ограничивает периодичность обновления
}
```

В следующей главе

До сих пор мы обсуждали аппаратные дела, но теперь пришло время переключиться на другие вещи! В **главе 5** вы подробно рассмотрите код для Arduino и узнаете множество способов программирования, а также специфику форматирования, которая понадобится для создания собственной программы для Arduino.

Программирование Arduino

До сих пор вы сталкивались с программированием, только когда загружали код на платформу Arduino (рис. 5.1). В этой главе вы узнаете об основных областях программирования и о способах, которые помогут освоить программирование Arduino самостоятельно.



Рис. 5.1. Средства программирования Arduino позволяют управлять платформой

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

Среда разработки Arduino

Программное обеспечение для программирования Arduino называется интегрированной средой разработки (*integrated development environment*), или сокращенно IDE. Большинство программ создается в рамках этой среды. Выражаясь непрофессиональным языком, чтобы успешно отправить код на платформу, нужно запустить Arduino IDE. Давайте рассмотрим эту среду более подробно.

Окно программирования

В общих чертах IDE состоит из окна программирования и строки меню. На **рис. 5.2** представлена среда Arduino IDE. Рассмотрим интерфейс более подробно.

1. **Имя файла.** В данном случае *скетч* (*sketch*, так называются *программы*, написанные для Arduino), представленный на рисунке, называется «Blink». По умолчанию имя нового файла состоит из месяца и дня.
2. **Номер версии.** Не знаете, в какой версии вы работаете? Посмотрите на этот номер, чтобы узнать, какая версия загружена на ваш компьютер. Не забывайте обновлять программное обеспечение, потому что предыдущие версии могут содержать ошибки.
3. **Монитор порта.** Функция предназначена для работы с последовательным интерфейсом. Подробнее о мониторе последовательного интерфейса и его преимуществах вы узнаете позже.
4. **Выбор вкладки.** Если у вас открыто несколько скетчей, то с помощью этой кнопки можно переключаться между ними.
5. **Проверить.** Нажмите на эту кнопку, если хотите проверить синтаксис кода без сохранения или загрузки программы на Arduino.
6. **Загрузить.** Эта кнопка — ваш лучший друг! С ее помощью можно проверить код, скомпилировать его на язык программирования, понятный Arduino, и отправить на платформу.

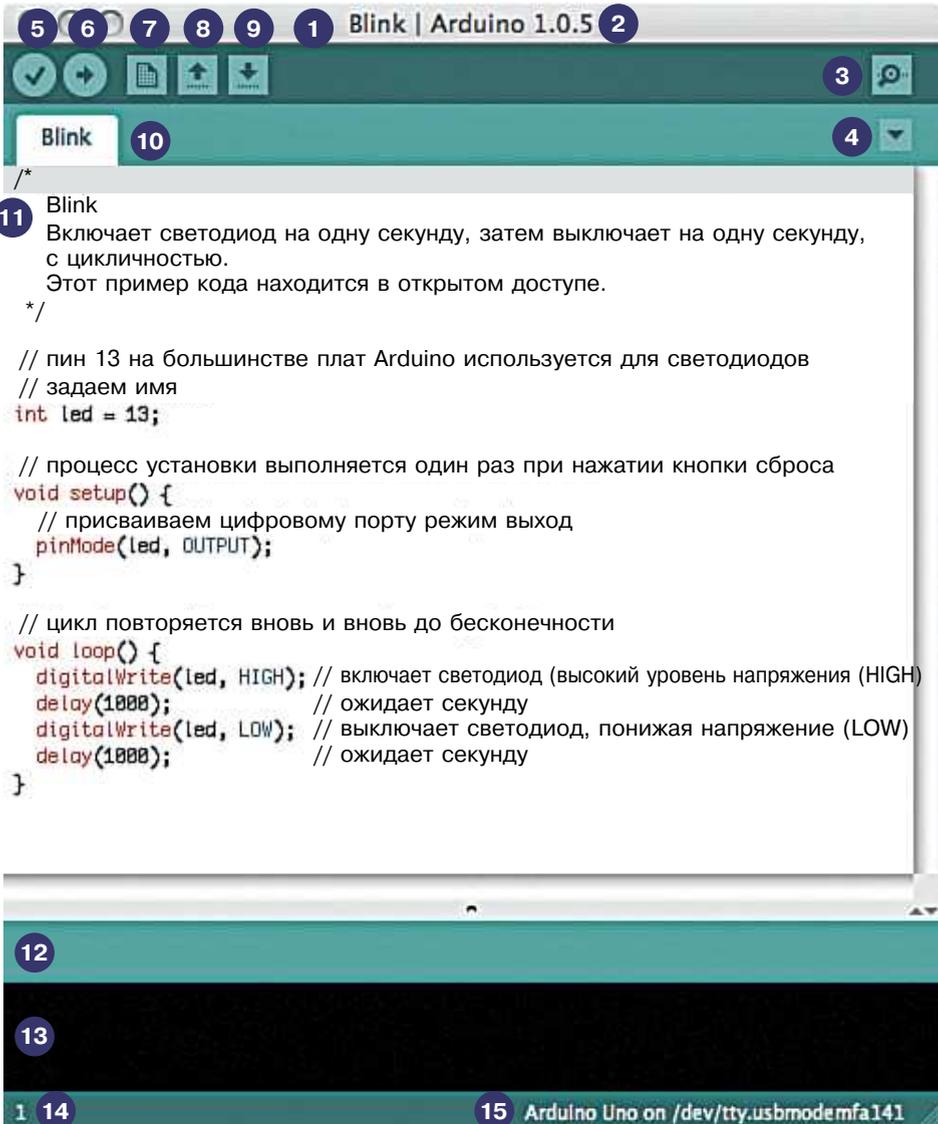


Рис. 5.2. Интерфейс среды Arduino располагает большим количеством опций

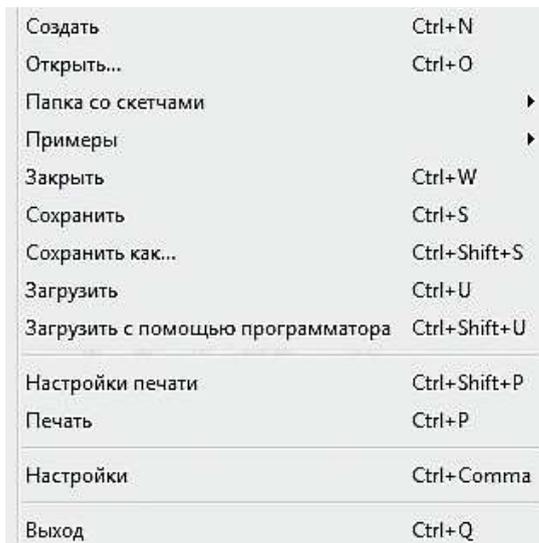
- 7. Создать.** С помощью этой функции можно создать новый скетч. Нажатие на кнопку никак не повлияет на уже открытые скетчи. По умолчанию имя нового файла — текущая дата.
- 8. Открыть.** С помощью этой кнопки можно открыть блокнот (*sketchbook*), который служит папкой по умолчанию для хране-

- ния скетчей. Отсюда вы можете открывать файлы, как делаете это обычно.
9. **Сохранить.** Обратите внимание, что скетчи Arduino нужно сохранять в папке. Если для нового файла нет папки, то ее создаст среда.
 10. **Вкладка.** У каждого окна есть собственное имя файла, которое дублируется в строке заголовка. Иногда скетчи могут состоять из нескольких вкладок. Загрузите пример скетча `toneMelody` (см. ниже, как найти примеры скетчей) и вы увидите принцип работы. Второй пример — файл с именем `pitches.h`. Это исходный файл, который должен исполнить основной скетч.
 11. **Текстовый редактор программного кода.** В этой части окна отображается скетч. Если скетч не загружен, то эта область остается пустой. Это основная рабочая область для написания скетча.
 12. **Строка состояния.** В строке синего цвета отображается статусное сообщение, обычно о состоянии загрузки. С ее помощью можно проследить процесс загрузки.
 13. **Окно состояния.** В окне черного цвета отображаются сообщения об ошибках во время загрузки. Программное обеспечение Arduino проверяет все отправленные коды и останавливает загрузку при наличии ошибки. При обнаружении проблемы в сообщении указывается причина ошибки, хотя зачастую ее сложно определить. Как видите, на **рис. 5.2** нет сообщения об ошибке.
 14. **Номер строки.** Каждая строка с кодом пронумерована. Такое обозначение помогает сориентироваться: числовое значение указывает, на какой строке находится курсор.
 15. **Модель платформы и номер порта.** В этой книге я работаю исключительно с UNO, но существуют и другие версии платформ Arduino. Выберите платформу, на которую вы выполняете загрузку, пока не появились сообщения об ошибке. Вторая часть информации — это номер порта. Не вдавайтесь в технические подробности; программное обеспечение определит его автоматически.

Меню

Теперь давайте рассмотрим опции меню. Некоторые из них вполне стандартны («Открыть файл»), в то время как другие дублируют некоторые команды из окна программирования. Тем не менее, они очень важны! Не имеет значения, какая операционная система (Mac OS X, Linux или Windows) установлена на вашем компьютере, вы не заметите особой разницы в меню. Просто изучите меню и выясните, для чего предназначена каждая команда.

На **рис. 5.3** представлено меню «Файл» для операционной системы Mac OS X. Если вы работаете в Linux или Windows, не беспокойтесь; меню выглядит примерно одинаково. Перейдем к обзору.



Создать	Ctrl+N
Открыть...	Ctrl+O
Папка со скетчами	▶
Примеры	▶
Заккрыть	Ctrl+W
Сохранить	Ctrl+S
Сохранить как...	Ctrl+Shift+S
Загрузить	Ctrl+U
Загрузить с помощью программатора	Ctrl+Shift+U
<hr/>	
Настройки печати	Ctrl+Shift+P
Печать	Ctrl+P
<hr/>	
Настройки	Ctrl+Comma
<hr/>	
Выход	Ctrl+Q

Рис. 5.3. Файловое меню помогает работать со скетчами

Меню «Файл»

Почти во всех приложениях меню «Файл» включает стандартные команды «Создать», «Открыть», «Заккрыть», «Сохранить» и «Сохранить как», принцип работы которых вы знаете. Ниже представлена информация об остальных командах меню «Файл».

- **«Папка со скетчами».** При выборе этого пункта появляется всплывающее окно с вашим блокнотом (sketchbook), и вы можете выбрать один из существующих скетчей. В сущности, это похоже на команду «Открыть» в окне программирования.
- **«Примеры».** Обеспечивает доступ к примерам скетчей и кодов, которые можно бесплатно использовать для создания собственных программ. Подробнее о примерах я расскажу позже в разделе «Учимся на примере кода».
- **«Загрузить».** С помощью этой команды можно отправить скетч, написанный в окне программирования, на Arduino. При этом предполагается, что плата должна быть подключена.
- **«Загрузить с помощью программатора».** Дополнительная команда. Ею лучше не пользоваться, так как вы можете удалить прошивку.

Меню «Правка»

При выборе меню «Правка» откроется список команд (рис. 5.4). Как и в меню «Файл», здесь существует стандартный набор команд, которые не нуждаются в объяснении. Это действия «Вырезать», «Копировать», «Вставить», «Выделить все», «Найти».

Отменить	Ctrl+Z
Вернуть	Ctrl+Y
Вырезать	Ctrl+X
Копировать	Ctrl+C
Копировать для форума	Ctrl+Shift+C
Копировать как HTML	Ctrl+Alt+C
Вставить	Ctrl+V
Выделить всё	Ctrl+A
Комментировать/Раскомментировать	Ctrl+Slash
Увеличить отступ	Ctrl+Close Bracket
Уменьшить отступ	Ctrl+Open Bracket
Найти...	Ctrl+F
Найти следующее	Ctrl+G
Найти предыдущее	Ctrl+Shift+G
Найти выбранное	Ctrl+E

Рис. 5.4. Помимо стандартных действий «Копировать» и «Вставить» в меню «Правка» есть дополнительные команды

В меню «Правка» есть ряд интересных опций.

- **«Копировать для форума».** Эта команда позволяет копировать скетчи Arduino с возможностью форматирования. При ее выборе текст форматируется и сохраняет свою наглядность при использовании на форуме.
- **«Копировать как HTML».** С помощью этой команды можно скопировать скетч, добавить соответствующие метки и разместить его на своей веб-странице.
- **Комментировать/Раскомментировать».** С помощью этой команды можно добавить однострочный или многострочный комментарий, и Arduino не будет принимать этот текст за часть кода. Более подробно о добавлении комментариев я расскажу в разделе «Скетч „Blink“».
- **«Увеличить отступ» и «Уменьшить отступ».** Эти команды позволяют изменить отступ. Если вы посмотрите на скетч Arduino, то увидите, что для обеспечения наглядности кода используются отступы.

Меню «Скетч»

В меню «Скетч» (рис. 5.5) представлено несколько команд, с помощью которых можно управлять скетчами. Помните, что программы, написанные для Arduino, называются скетчами.

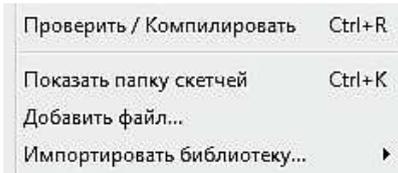


Рис. 5.5. Меню «Скетч» позволяет добавить в скетч библиотеки и файлы поддержки

- Функция **«Проверить/Компилировать»** осуществляет проверку кода на ошибки и компиляцию. Это хороший способ убедиться в том, что программа скорее всего заработает, потому что среда не будет компилировать плохой код. Я сказал «скорее всего», потому что даже если компиляция прошла успешно, это не означает, что все заработает так, как вы хотите.
- С помощью команды **«Показать папку скетча»** можно открыть папку с файлом текущего скетча. Как было отмечено, среда сохраняет каждый скетч в отдельной папке.
- Пункт меню **«Импортировать библиотеку»** позволяет добавить в скетч библиотеку. Библиотека функционирует как исходный файл. Это дополнительный код, который можно добавить в основной скетч для упрощения работы. Подробнее о библиотеках будет рассказано в разделе «Библиотеки».

Меню «Инструменты»

Инструменты — ваши помощники в программировании. На рис. 5.6 представлены команды этого меню.

- **«Автоформатирование»**. Эта команда упорядочивает каждую строку скетча, чтобы она выглядела определенным образом. Если вы откроете примеры скетчей, то обратите внимание, что они отформатированы для удобного чтения. Когда вы будете создавать программу, строки будут форматироваться автоматически, но это происходит не всегда!
- **«Архивировать скетч»**. С помощью этой команды можно создать ZIP-архив скетча.
- **«Исправить кодировку и перезагрузить»**. Эта команда исправляет проблемы, связанные с набором текста, например, иногда кавычки или знак ударения принимаются за код, а не за отдельный знак.
- **«Монитор порта»**. Этот инструмент, как и кнопка, о которой я говорил в разделе «Окно программирования», предназначен для

работы с последовательным интерфейсом. Это очень полезный инструмент, мы рассмотрим его подробнее в разделе «Отладка с помощью монитора последовательного интерфейса».

- **«Плата».** С помощью этого инструмента можно выбрать модель вашей платы. В этой книге используется только Arduino UNO, но существует множество разновидностей платформ Arduino. Чтобы успешно загрузить скетч на плату, нужно выбрать правильную модель. Если вы забыли модель и выбрали неправильную плату, то процесс загрузки прервется и появится сообщение об ошибке.
- **«Последовательный порт».** С помощью этого инструмента можно выбрать порт для загрузки скетча. Вам не нужно вникать в эти технические подробности, только если вы не являетесь продвинутым пользователем.
- **«Записать загрузчик».** С помощью этой команды можно отформатировать чип микроконтроллера Arduino. Это сложная тема, и я не касаюсь ее в книге.

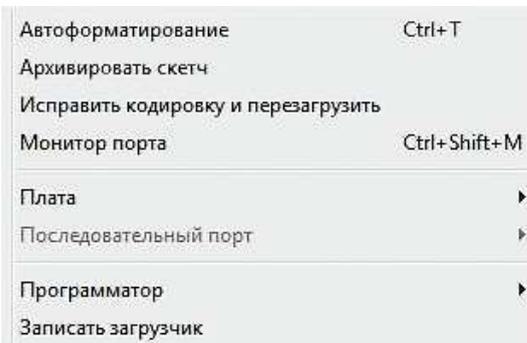


Рис. 5.6. В меню «Инструменты» есть множество опций, с помощью которых можно улучшить работу скетча

ДЛЯ СПРАВКИ

ИНСТРУМЕНТ «ПРОГРАММАТОР» НЕ РАССМОТРЕН В КНИГЕ

Инструмент «Программатор» — достаточно сложная тема, которая выходит за рамки этой книги. Тем не менее, вы можете найти больше информации по этой теме в книге «Sams Teach Yourself Arduino Programming in 24 Hours» (автор: Rich Blum), 2014 г.

Меню «Помощь»

В меню «Помощь» (рис. 5.7) представлен ряд документов, хранящихся на компьютере вместе с программным обеспечением Arduino. Это справочные материалы, которые я рекомендую посмотреть. В разделе «От-

ладка с помощью монитора последовательного интерфейса» вы узнаете о некоторых способах устранения неполадок в скетчах, но сначала изучите справочные материалы.

Начало работы	
Среда разработки	
Поиск неполадок	
Содержание	
Найти в справке	Ctrl+Shift+F
Часто задаваемые вопросы	
Перейти на Arduino.cc	
<hr/>	
О программе	

Рис. 5.7. В меню «Помощь» немного справочных материалов, но все они очень полезные

Больше информации о среде Arduino IDE вы найдете на странице <http://arduino.cc/en/Guide/Environment>.

Теперь, когда вы познакомились с системой меню, перейдем к скетчам.

ЗАГРУЗКА СКЕТЧЕЙ В ARDUINO

Возможно, вам не нужна лишняя информация. Вы только хотите узнать, как загрузить скетч (программу) в Arduino. Тогда эта инструкция для вас.

- 1.** Убедитесь, что у вас установлена последняя версия программного обеспечения Arduino.
- 2.** Подключите Arduino к компьютеру с помощью USB-кабеля. Иногда может всплывать окно с сообщением о том, что плата Arduino обнаружена. Не обращайтесь к нему внимание.
- 3.** В меню «Инструменты→Плата» проверьте, что вы выбрали нужную модель платформы Arduino. Больше информации об этом меню вы найдете в разделе «Среда разработки Arduino».
- 4.** Откройте скетч в программном обеспечении Arduino. Выберите команду «Загрузить», чтобы отправить скетч на плату. Светодиоды с маркировкой TX и RX будут мигать с высокой частотой при загрузке кода. Если не появится сообщение об ошибке, то код должен загрузиться автоматически.

Если в скетче есть ошибка, то он не загрузится на плату. Если вы хотите проверить соединение с помощью правильного кода, загрузите один из примеров скетчей (нажмите «Файл→Примеры»). Если это не сработает, то нужно проверить настройки и соединения.

Скетч «Blink»¹

У программистов есть забавная традиция: первая программа, которую нужно создать, должна вывести на экран фразу «Hello, world!» («Всем привет!»). В мире электроники аналогом служит создание мигающего светодиода, это и является задачей скетча «Blink».

Теперь подробнее о коде. Сначала убедитесь в том, что у вас установлена последняя версия программного обеспечения Arduino. Для этого запустите программу и посмотрите на строку заголовка (обозначение 2 на **рис. 5.2**), где отображен номер версии. Затем зайдите на сайт Arduino (arduino.cc) и выберите пункт «Download»² («Загрузить»), откроется страница, где будет указан номер текущей версии.

После того как вы убедитесь, что у вас установлена последняя версия, запустите скетч. Для этого выберите «Файл → Примеры → 01. Basics → Blink»³. В окне появится код, который вы можете отредактировать по своему желанию и затем сохранить его под новым именем. (Вы не можете сохранить его как «Blink», потому что этот файл доступен только для чтения.)

Итак, давайте подробнее рассмотрим код.

ДЛЯ СПРАВКИ

ЭТОТ КОД У ВАС УЖЕ ЕСТЬ

Я не стал приводить ссылку на сайт книги, с которого вы можете скачать пример кода, потому что этот код уже есть в среде Arduino IDE.

```
/*
  Blink
  Включает светодиод на одну секунду, затем выключает на одну секунду,
  с цикличностью.
```

Этот пример кода находится в открытом доступе.

```
*/
```

Как вы видите, первая часть скетча включает некую информацию, которая начинается со слеша со звездочкой и заканчивается теми же символами, стоящими в обратном порядке. Так обозначают комментарий, на который Arduino не нужно обращать внимание. Как правило, програм-

¹ В русскоязычном интерфейсе программы названия скетчей не переведены. Чтобы читатель, не владеющий иностранным языком, смог быстро сориентироваться, оставлен вариант без перевода. — *Прим. перев.*

² Сайт на английском языке. Для удобства поиска на странице сайта пункт «Download» не переведен. — *Прим. перев.*

³ В русскоязычном интерфейсе эти пункты меню не переведены. Чтобы читатель, не владеющий иностранным языком, смог быстро сориентироваться, оставлен вариант без перевода. — *Прим. перев.*

мисты вставляют такой текст, чтобы помнить, для чего предназначен код и зачем он был написан. Иногда такой текст используется в качестве наглядного пособия для других программистов.

Что если бы Arduino принял оставленную вами подсказку за часть кода? Это стало бы головной болью. Понять, что вы сделали не так, — дело непростое. Поэтому, если вы хотите вставить в свой код информационный блок, используйте эти символы.

В следующей части кода есть три интересные особенности. Во-первых, обратите внимание на двойной слеш в начале двух строк. Это еще один способ добавления комментария. В этом случае Arduino будет игнорировать всю строку после двойного слеша, благодаря чему вы сможете оставлять важные заметки для себя или других людей, например, для чего нужен пин 13.

```
// пин 13 на большинстве плат Arduino используется для светодиодов.  
// задаем имя  
int led = 13;
```

Вторая деталь, на которую следует обратить внимание, — это часть кода в третьей строке. Это объявление переменной. `int` означает «целое число» (*integer*). Не вдаваясь в подробности, скажу, что число должно находиться в диапазоне от $-32,768$ до $32,767$. Если это не так, используйте другой синтаксис, чтобы объявить переменную.

Слово `led` — имя переменной. Насколько вы помните, согласно программе мигающий светодиод подключен к пину 13. Но чтобы Arduino понял это, вы должны объявить, что мигание должно происходить на пине 13. Объявление происходит в два этапа. Первый этап — создание переменной с именем `led` и значением 13. Если вы хотите узнать больше об объявлении переменных, то посмотрите интересный урок на сайте Arduino: <http://arduino.cc/en/Reference/VariableDeclaration>.

Третья особенность — точка с запятой. Обычно в конце каждой строки с кодом стоит точка с запятой, так Arduino понимает, что нужно перейти к следующей строке. Если вы пропустите точку с запятой, то появится сообщение об ошибке и код не будет загружен.

Следующий блок начинается с команды `void setup`. На самом деле здесь две команды. «Void» — это ключевое слово для Arduino, которое указывает на то, что информация содержится только в этом блоке, и информация не возвращается во внешнюю программу. Для такой небольшой программы, как «Blink», этот нюанс не имеет значения.

```
// процесс установки выполняется один раз при нажатии кнопки сброса  
void setup() {  
    // присваиваем цифровому пину режим выход.  
    pinMode(led, OUTPUT);  
}
```

Функция `setup` запускается только один раз при включении питания и при нажатии на кнопку сброса. Идеально для инициализации переменных и других однократных задач.

Далее следует фигурная скобка `{`, которая сопровождается закрывающей код скобкой `}`. Это часть команды `void setup`. Любые функции внутри этих скобок запускаются в то же время, что и основная команда.

Во второй части объявления переменной функция `pinMode` устанавливает режим работы вывода Arduino: будет он отправлять данные (*output*) или принимать их (*input*). В нашем случае функция `pinMode` сообщает Arduino, что пин `led` (который до этого был заявлен как `Pin 13` (пин 13) работает на выход (*output*). Это означает, что Arduino запустит пин 13, как указано в программе, не пытаясь получить показания от какого-либо датчика, подключенного к пину.

И последний элемент этого блока — закрывающая фигурная скобка. Если вы вставляете фигурные скобки, не забывайте, что открывающая скобка всегда должна сопровождаться закрывающей скобкой. Если такой баланс будет нарушен, программа не заработает.

Заключительный блок кода — цикл (*loop*). В отличие от функции `setup`, которая запускается только один раз, цикл выполняется до тех пор, пока Arduino не выключится. Здесь обычно находится функциональная часть программы.

```
// цикл повторяется вновь и вновь до бесконечности
void loop() {
    digitalWrite(led, HIGH); // включает светодиод (высокий уровень
                             // напряжения - HIGH)
    delay(1000);             // ожидает секунду
    digitalWrite(led, LOW); // выключает светодиод, понижая
                             // напряжение (LOW)
    delay(1000);             // ожидает секунду
}
```

Цикл содержит команду, которая включает и выключает светодиод. В комментариях видно, что за это отвечает функция `digitalWrite`. Считывание значений с пина и передача их на пин осуществляется с ее помощью и функций `digitalRead`, `analogRead` и `analogWrite`. Как было отмечено, каждый пин может работать на вход и на выход. Кроме того, вывод может быть цифровым или аналоговым. О том, что именно это означает, вы узнаете в **главе 6** «Восприятие мира», а пока достаточно сказать, что в этом коде управление светодиодом осуществляется с помощью функции `digitalWrite`.

В мире Arduino ключевые слова `HIGH` и `LOW` означают подачу напряжения на пины. Понятно, что `HIGH` означает включение, а `LOW` выключение. Этот цикл включает светодиод на пине 13, затем ждет 1 с — `delay(1000);` команда просит Arduino подождать 1000 мс — затем выключает светодиод на 1 с, и цикл начинается снова.

Как вы видите, даже из самой простой программы можно узнать много интересного!

Учимся на примере кода

Вы можете найти множество примеров скетчей, если выберете в меню программного обеспечения Arduino «Файл→Примеры» (рис. 5.8). Есть примеры, которые служат хорошим учебным пособием для изучения сложных тем. Это подчеркивает важность примера кода, не столько как готового решения, сколько основы для изучения синтаксиса с целью изменения кода для своего проекта. Например, выберите «Файл → Примеры → Digital → Button». Этот скетч покажет, как включить светодиод нажатием на кнопку.

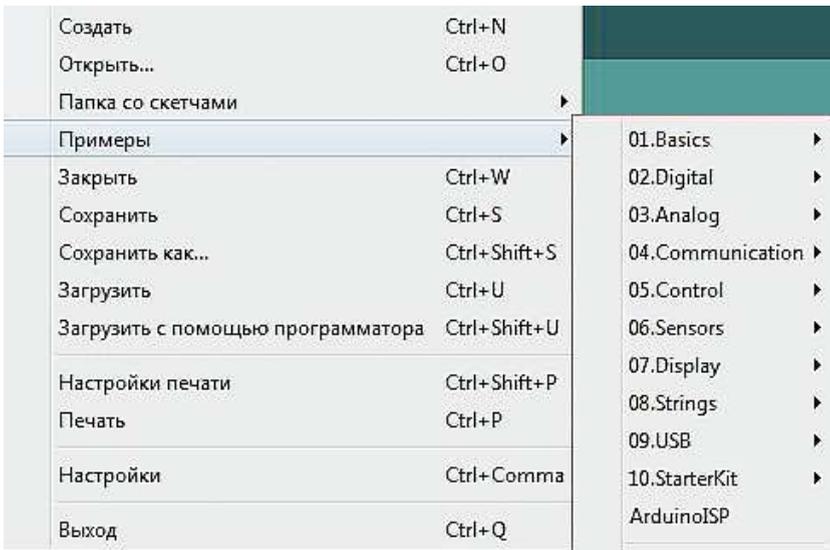


Рис. 5.8. Нужна помощь в написании скетчей? Посмотрите пример кода для Arduino

Итак, вы открыли скетч «Button». Давайте поработаем с ним.

Адаптируем код

Если вы хотите научиться писать код, первым делом откройте программу, которая действительно работает, и начните вносить в нее изменения, чтобы код отвечал задачам вашего проекта.

Например, давайте изменим номер пина. Зачастую вы можете менять входы, выходы и пины подключения. В примере «Кнопка» кнопка подключена к пину 2, а светодиод — к пину 13. Что произойдет, если вы перенаправите кнопку на другой пин? Давайте поменяем номер пина на 10. В 29 строке кода вы увидите следующее:

```
const int buttonPin = 2;
```

Замените значение 2 на 10 и загрузите скетч на плату. Вуаля! Вы закончили? Давайте проверим.

СОВЕТ

УБЕДИТЕСЬ, ЧТО ПЛАТА ПОДКЛЮЧЕНА

Разумеется, чтобы проверить скетч, нужно подключить плату, как указано в скетче.

В результате проверки вы поймете, что программа не работает. О чем это говорит? Оказывается, был важен пин 2. Это вывод данных прерывания, который может сорвать цикл. Возможно, вы обратили внимание, что в скетче цикл постоянно работает, ожидая команды «включить светодиод». Пин 10 не является выводом прерывания, поэтому он не может прервать цикл. Я расскажу об этом подробнее в разделе «Прерывания».

Итак, программа не работает. А что, если переместить светодиод на пин 12? Давайте сделаем это, но сначала поменяем пин с кнопкой на 2. Замените в следующей строке номер пина на 12:

```
const int ledPin = 13;
```

Ура! Все работает. Вы кое-чему научились.

Ищем пример кода

Да, в программном обеспечении Arduino есть несколько примеров скетчей, но их можно найти не только в программе. Обычно при выпуске нового продукта, совместимого с Arduino, производители предлагают скетч, который точно работает. Так, вы можете протестировать свою плату, пока изучаете синтаксис.

Наиболее верный способ найти пример кода для конкретного продукта — заглянуть на страницу продукта на сайте продавца или производителя. В качестве примера рассмотрим плату расширения с множеством светодиодов, представляющую собой светодиодную матрицу 9×14 , которую можно использовать в качестве информационного табло.

Что делать, если вы хотите взяться за собственный проект, а не покупать чей-то набор? Следующие рекомендации помогут вам найти нужный код для начала работы.

Arduino Playground

Если вы не можете найти пример, то следующий пункт назначения — это Arduino Playground (<http://playground.arduino.cc/>), технический дом платформы Arduino (рис. 5.10). Здесь вы найдете рекомендации, уроки и коды почти на каждый случай.

Main Site · Blog · Playground · Forum · Labs · Store Sign in or Register

ARDUINO PLAYGROUND

The playground is a publicly-editable wiki about Arduino.

The Arduino Playground

Welcome to the Arduino Playground, a wiki where all the users of [Arduino](#) can contribute and benefit from their collective research.

This is the place to post and share your own code, circuit diagrams, tutorials, DIY instructions, tips and tricks, and after all the hard work, to show off your projects! Anyone can edit and add to the pages here.

Arduino Playground is a **work in progress**. We can use all the help you can give, so please read the [Participate](#) section and get your fingers typing!

NOTE (20081002): Solved the problem for adding content in non-western Latin languages, now it is possible to start typing in any language on the [internet test here](#)

Manuals and Curriculum

- [Installing Arduino on Linux](#)
- [Board Setup and Configuration](#)
- [Development Tools](#)
- [Arduino on other Atmel Chips](#)
- [Interfacing With Hardware](#)
 - [Output](#)
 - [Input](#)
 - [User Interface](#)
 - [Storage](#)
 - [Communication](#)
 - [Power supplies](#)

Рис. 5.9. На сайте Arduino Playground программисты могут обмениваться идеями

Во время поиска материала на сайте вы должны помнить, что это не справочно-информационный ресурс, а именно playground. Поскольку он предназначен для тех, кто работает с Arduino, здесь есть многое, что, возможно, вам не понадобится: устранение ошибок, руководство по сборке схем и проекты, включающие неизвестные элементы, которых у вас нет. Если вы стремитесь узнать больше об Arduino, то не будете разочарованы. Но если вы ищете что-то конкретное, что точно будет работать, вы можете и не найти это.

Если новичок хочет разобраться в материалах, то лучше начать с раздела «Manuals and Curriculum»¹ («Руководство и учебные программы»). Он содержит множество материалов, доступных для скачивания, а также ссылки на сторонние ресурсы и учебные пособия для начинающих.

¹ Сайт на английском языке (русифицированная версия отсутствует), поэтому название раздела приводится на английском языке. В скобках дан перевод. — *Прим. перев.*

Библиотеки

Другой источник с примерами — библиотеки. Это файлы с кодом, которые выступают фоном в скетчах, не нарушая целостности самого скетча. Зачастую библиотеки содержат примеры скетчей, в которых показано, как пользоваться библиотекой. Хотите узнать больше о принципе работы сервоприводов? Загрузите библиотеку *Servo* (<http://playground.arduino.cc/ComponentLib/servo>) и установите ее, чтобы воспользоваться двумя скетчами, которые идут с ней.

Подробнее о библиотеках и их установке вы узнаете в разделе «Все о библиотеках».

Обмен примерами кода

Особенность Arduino — открытый программный код (см. ниже). Это означает, что Arduino разделяет принцип общедоступности. Например, если бы вы захотели самостоятельно сконструировать плату Arduino UNO и спаять все компоненты, вы безусловно могли бы сделать это. Обратной стороной такого обмена является то, что если вы разрабатываете проект с открытым исходным кодом, вы должны рассказать людям, о том, как вы написали его, чтобы они могли научиться у вас.

Это не означает, что вы должны делиться секретом, но при использовании ресурсов с открытым исходным кодом такой поступок — признак хорошего тона.

Вы можете поделиться кодом различными способами.

- **Arduino Playground.** Разместите код на своем сайте, а затем добавьте ссылку на Playground, который является публично редактируемым вики-ресурсом, как Википедия. Разместив код на своем сайте, вы можете быть уверены, что он не исчезнет просто так с Playground. (Не стоит забывать, что это ресурс, открытый для редактирования.) Он также посылает трафик на ваш сайт, что позволяет вам похвастаться своими проектами!
- **Code.Google.com** или **GitHub.com.** Эти сервисы предоставляют место для хранения и распространения кода, поэтому вам не понадобится размещать его на своем сайте. Помимо этого, здесь есть несколько заманчивых опций. Например, другие пользователи могут развить ваш проект, создав собственную версию вашего проекта.

Если вы делитесь кодом, не стесняйтесь добавлять подобный комментарий:

```
// Код написан Джоном Бейкталом / nerdage.net
// Если вы используете код, пожалуйста, укажите ссылку на первоисточник
```

Если вы заимствовали чужой код, то должны указать это в комментариях!

ЧТО ТАКОЕ ОТКРЫТЫЙ ИСХОДНЫЙ КОД?

Открытое аппаратное и программное обеспечение (**рис. 5.10**) — философия, которая поддерживает совместное использование исходного кода, схем, проектов и т. д. Давайте представим, что вы продаете набор по робототехнике. Люди могут приобрести набор в вашем интернет-магазине или же скачать файл и создать проект. Совершенно верно: вы обеспечиваете бесплатную загрузку всего, что нужно для создания робота, включая код Arduino, схемы печатных плат, и даже векторы для лазерного резака.



Рис. 5.10. Найдите логотип открытого аппаратного обеспечения. Он представлен в виде шестеренки с надписью OSHW (*open-source hardware*, открытое аппаратное обеспечение)

Возникает вопрос: насколько это целесообразно для бизнеса? Неужели кто-нибудь будет покупать набор, если его можно скачать бесплатно? Оказывается, на создание одной копии проекта уйдет больше средств и ресурсов, чем на покупку набора. Нужно сделать печатную плату, что подразумевает отправить заказ в специальную сервисную службу или сконструировать самостоятельно. Вам придется купить все электронные детали, а производители покупают оптом со скидкой и предлагают покупателю привлекательные цены. И наконец, лазерный резак — дорогое удовольствие, к тому же его надо найти. Производителям незачем бояться людей, конструирующих собственные платы. Такие люди вряд ли купят набор.

Однако у продавцов открытого аппаратного обеспечения есть причины опасаться конкуренции недобросовестных производителей, которые на основе устройств с открытым исходным кодом (как Arduino) производят свою продукцию в подпольных условиях и затем продают ее со скидкой, выдавая продукт за оригинал. Тем не менее вы, безусловно, можете создавать собственную версию проекта и затем продавать ее, но должны попробовать усовершенствовать или переделать проект, чтобы вас не уличили в плагиате!

Ниже приведены общепринятые правила в отношении использования открытого аппаратного обеспечения.

- **Укажите первоисточник.** Возможно, я не открою Америки, если скажу, что иногда компании хотят получить преимущество от использования открытого исходного кода, фактически не выдавая его.
- **Уважайте чужой труд.** Если при создании скетча вы используете чужой код, укажите это в комментариях. Постарайтесь упомянуть каждого, чей проект с открытым исходным кодом лег в основу вашего.
- **Если вы создаете проект с использованием открытого исходного кода, то откройте к нему доступ.** Брать и не отдавать взамен — признак дурного тона.

Если у вас все еще есть сомнения по поводу коммерческой целесообразности открытого аппаратного обеспечения, приведу в пример две компании: *Adafruit Industries* и *SparkFun Electronics*. Эти компании, чей оборот составляет миллионы долларов, специализируются исключительно на оборудовании с открытым исходным кодом.

Чтобы узнать больше об открытом аппаратном обеспечении, посетите сайт Ассоциации открытого аппаратного обеспечения (Open Source Hardware Association — OSHWA): www.oshwa.org/.

Функции и синтаксис

Давайте рассмотрим еще несколько функций, с которыми вам, вероятно, придется работать в процессе программирования Arduino. В этом разделе информация представлена в сжатом виде, но я привожу ссылки на страницы, где вы можете изучить материал более подробно.

Арифметические операторы

Arduino может выполнять математические операции, и не только сложные и вычитание. В среде Arduino есть библиотека `math.h`, отвечающая за выполнение функций, которые встроены в чип микроконтроллера

Arduino. Ее не нужно загружать дополнительно. Она может решать задачи не только из курса высшей математики, но и арифметики:

```
z = x + y;  
z = x - y;  
z = x * y;  
z = x / y;
```

Вы можете посетить сайт Arduino www.arduino.cc/en/Math/H и найти более подробную информацию о библиотеке `math.h`.

Массивы

Массивы — способ управления большим количеством информации. Вы всегда можете задать массиву функцию, потому что он обозначен в фигурных, а не в круглых скобках:

```
int myArray[10]={9,3,2,4,3,2,7,8,9,11};
```

Число в квадратных скобках — количество элементов в массиве, а числа в фигурных скобках и есть массив. Каждый элемент пронумерован, начиная с 0. Таким образом, первый элемент — это `myArray[0]`, а восьмой — `myArray[7]`.

Подробнее о массивах см. на сайте <http://arduino.cc/en/Tutorial/Array>.

Операторы сравнения

«Больше», «меньше», «равно», — это операторы сравнения. В Arduino они выглядят следующим образом:

```
== (оператор равенства)  
!= (оператор неравенства)  
< (оператор «меньше»)  
> (оператор «больше»)  
<= (оператор «меньше или равно»)  
>= (оператор «больше или равно»)
```

Оператор for

Функция `for` повторяет действие несколько раз. Например, если вы хотите, чтобы светодиод загорелся 10 раз, то функция `for` должна выглядеть следующим образом:

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  for (int item=0; item < 10; item++){
    digitalWrite(led, HIGH);
    delay(1000);
    Serial.print(item);
    digitalWrite(led, LOW);
    delay(1000);

    if (item==9){
      break;
    }
  }
}

void loop() {
}
```

Итак, что же происходит с кодом? Во-первых, обратите внимание, что оператор `for` находится внутри функции `void setup`, а не `void loop`. Это означает, что цикл `for` запустится только один раз. Если вы поместите оператор в `void loop`, то цикл будет повторяться от 0 до 9 бесконечно.

Во-вторых, цикл рассматривает целое число `item`, начальное значение которого равно 0. Когда цикл начинается заново, значение увеличивается на 1 (на это указывает `item++`). Когда значение достигает 9, цикл `for` прерывается и скетч переходит к выполнению функции `void loop`, которая в настоящий момент не задана.

Более подробно о цикле `for` вы узнаете в уроке Arduino на сайте <http://arduino.cc/en/Tutorial/ForLoop>.

Директива include

Директива `include` служит для включения в скетч другого файла, обычно библиотеки. Например, если вы хотите запустить шаговый электродвигатель с помощью библиотеки `stepper.h`, добавьте в начало скетча следующее:

```
#include <Stepper.h>
```

Более подробную информацию см. по адресу <http://arduino.cc/en/Reference/Include>.

Увеличение/уменьшение значения

Операторы увеличения или уменьшения соответственно увеличивают или уменьшают значение переменной на 1. Вы могли заметить это в примере с оператором `for`. Помните `item++`? Двойной плюс увеличивает значение целого числа «`item`» на 1. `item--` уменьшает значение на 1. Более подробную информацию об этих операторах вы найдете по адресу <http://arduino.cc/en/Reference/Increment>.

Прерывания

Как модифицировать в скетче часть `void loop`? В качестве примера рассмотрим измененный скетч «Blink». Я сделал так, что нужно щелкнуть выключателем, чтобы он заработал.

```
int led = 13;
void setup() { pinMode(led, OUTPUT);
}
void loop() {
    int switch1 = digitalRead(2);
    if (switch1 == HIGH) {
        digitalWrite(led, HIGH);
        delay(1000); digitalWrite(led, LOW);
        delay(1000);
    }
}
```

В данном случае основная особенность заключается в том, что выключатель 1 подключен к пину 2 на Arduino. У Arduino UNO есть два вывода прерывания: цифровые пины 2 и 3. Если вы захотите изменить или остановить цикл, воспользуйтесь этими пинами.

Оператор `if/else`

С помощью этой функции можно задавать условие. Если условие истинно, то выполняется одно действие, если ложно — другое. Оператор `if` присутствует в предыдущем скетче, где нажатие на кнопку определяет включение или выключение светодиода. А где же `else`? Часть `else` появляется, если условие `if` ложно. Например, если бы в предыдущем скетче было два светодиода, то можно было бы добавить `else`, чтобы зажечь второй светодиод при переключении выключателя:

```
switch1 == HIGH) {
    digitalWrite(led1, HIGH);
}
else
{
    digitalWrite(led2, HIGH);
}
```

Подробнее об операторе `if/else` см. на сайте <http://arduino.cc/en/Reference/else>.

Функция `map`

Функция `map` преобразовывает значения переменной из одного диапазона в другой. Например, возможные значения аналогового датчика от 0 до 1023 можно преобразовать в значения в диапазоне от 5 до 23. Синтаксис:

```
mySensor = map(mySensor, 0, 1023, 5, 23);
```

Надеюсь, вы поняли. Более подробная информация дана на сайте <http://arduino.cc/en/Reference/map>.

Генераторы случайных значений

Если вы хотите сгенерировать псевдослучайное число, воспользуйтесь простой функцией:

```
randomNumber = random(min, max);
```

Параметры `min` и `max` служат для обозначения соответственно нижнего и верхнего пределов возможных значений. Таким образом, функция `random(5, 10);` будет возвращать псевдослучайные числа в диапазоне между 5 и 10. А что означает псевдослучайное число? Оказывается, в компьютерах, в том числе в Arduino, все настолько логично и упорядоченно, что нельзя создать случайное число. Вместо этого используется такой прием, как подсчет количества миллисекунд с момента включения устройства. Хотя это ненастоящее случайное число, его можно считать таковым.

Операторы `switch/case`

Еще одно применение конструкции `if` — переключение между вариантами, зависящими от значения переменной. Предположим, что переменная `myVariable` может иметь значения *A*, *B* и *C*. Можно настроить программу так, что выполнение действий будет зависеть от результата. В примере Arduino выводит название фрукта:

```
switch (myVariable) {  
  case A:  
    Serial.print("Apple");  
    break;
```

```
case B:
    Serial.print("Banana");
    break;
case C:
    Serial.print("Cherry");
    break;
}
```

Более подробная информация представлена на сайте: <http://arduino.cc/en/Reference/SwitchCase>.

Циклы while

Функция `while` создает цикл, который будет исполняться бесконечно до тех пор, пока не будет выполнено условие. В следующем фрагменте кода `while` будет вычислять в цикле до тех пор, пока кнопка не будет нажата 100 раз:

```
buttonPress = 0;
while(buttonPress < 100){
    buttonpress++;
}
```

На сайте Arduino есть отличный урок по `while`: <http://arduino.cc/en/Tutorial/WhileLoop>.

Отладка с помощью монитора последовательного интерфейса

Самый легкий способ выполнить отладку скетча, который успешно загружается на Arduino, но не работает правильным образом, — рассмотреть его в мониторе последовательного интерфейса. Монитор последовательного интерфейса представляет собой окно, в котором отображаются данные, передаваемые последовательно на Arduino и поступающие с платы. Как это работает? При подключении платформы к компьютеру через USB-кабель среда Arduino получает информацию с платы согласно скетчу, и таким же образом можно отправлять данные на Arduino.

Давайте рассмотрим особенности монитора последовательного интерфейса (**рис. 5.11**).

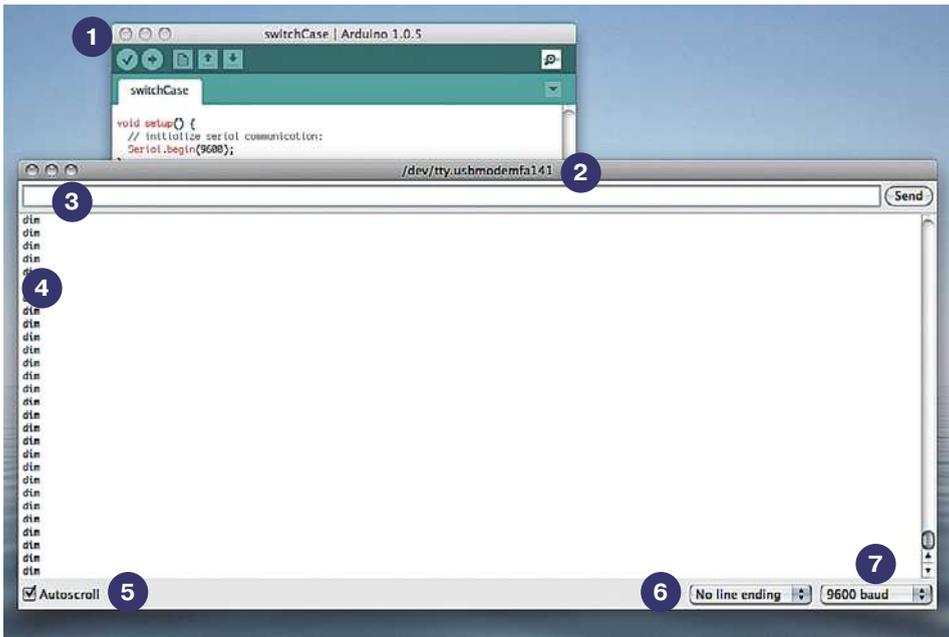


Рис. 5.11. Монитор последовательного интерфейса отображает данные, полученные от Arduino во время работы скетча «SwitchCase»

1. **Окно скетча.** Нужно открыть окно, чтобы запустить монитор последовательного интерфейса. Это необязательно должен быть скетч, с которым вы работаете.
2. **Номер порта.** Это необязательная информация для вас.
3. **Поле для ввода текста.** Хотите отправить текст на Arduino с помощью монитора последовательного интерфейса? Просто введите его в это поле и нажмите «Отправить». Обратите внимание, что в скетче должны быть правильные функции, чтобы с текстом можно было что-нибудь сделать.
4. **Область просмотра.** Здесь отображается текст.
5. **Автопрокрутка.** По умолчанию отмечена галочкой. По мере появления данных от Arduino автоматически прокручивает страницу к последней строке с информацией. Если галочка не стоит, то вам придется вручную перемещать элемент прокрутки в конец страницы, чтобы увидеть новые строки.
6. **Окончание строки.** Устанавливает опцию окончания строки текста. По умолчанию стоит значение «без окончания строки».
7. **Скорость передачи.** Это скорость соединения компьютера и Arduino. Она устанавливается с помощью выпадающего меню. Задать скорость можно с помощью кода, как описано далее.

Весьма любопытен тот факт, что с помощью этих заметок вы можете сообщить, где ошибка в вашем скетче. Один из способов сделать это — включить в код функции, осуществляющие передачу текста через последовательный интерфейс.

Рассмотрим скетч «AnalogReadSerial». Это один из примеров, включенных в программное обеспечение. Сначала обратим внимание на функцию, которая находится внутри `void setup()`:

```
void setup() {  
    // иницирует последовательный обмен данными на скорости 9600 бит/с  
    Serial.begin(9600);  
}
```

`Serial.begin()` иницирует последовательное соединение между компьютером и Arduino и задает скорость передачи данных в бит/с. 9600 — достаточно распространенный стандарт, вам не нужно устанавливать конкретную скорость; просто убедитесь в том, что скорость соответствует монитору последовательного интерфейса. Теперь посмотрим на цикл:

```
void loop() {  
    // считывание данных с аналогового пина 0  
    int sensorValue = analogRead(A0);  
    // передает данные через последовательное соединение  
    Serial.println(sensorValue);  
    delay(1); // приостановка считывания данных для обеспечения  
             // стабильности  
}
```

Обратите внимание на функцию `Serial.println()`. С ее помощью можно отправить информацию на монитор последовательного интерфейса; в данном случае выполняется считывание данных с аналогового входа 0. Что в этом примечательного? Если считывание не происходит, то вы сможете определить, что в программе произошла ошибка. Если у вас достаточно большой скетч, в котором периодически встречается функция `Serial.println()`, вы в буквальном смысле отслеживаете скетч по монитору и с точностью обнаруживаете ошибку. Попробуйте!

Все о библиотеках

Еще один ресурс Playground — библиотеки. Это набор функций, собранных в отдельный документ, так что вы можете ссылаться на них, не засоряя код.

Чтобы управлять подключенными к Arduino датчиками и другими устройствами, вам нужен код. Библиотека — лучший способ управления кодом. Если вы приобретаете новый компонент, не поленитесь найти соответствующую библиотеку — это сэкономит кучу времени!

Итак, давайте откроем библиотеку. Обратимся к библиотеке `Servo.h`, которую можно найти по ссылке <http://www.arduino.cc/en/Reference/Servo>. С ее помощью можно управлять сервоприводами. Ниже представлен пример того, как она работает. Существует основной скетч под названием «Sweep», с помощью которого можно управлять сервоприводами. Сразу замечу, что для работы понадобится библиотека:

```
#include <Servo.h>
```

Эта ссылка служит для обозначения того, что для запуска скетча нужна библиотека `Servo.h`. Если `Servo.h` находится вместе с остальными библиотеками, то загрузка соответствующих данных произойдет автоматически, и скетч заработает, как следует. Если библиотека отсутствует, то во время загрузки скетча на Arduino выведет сообщение об ошибке.

Теперь обратите внимание на один пример использования библиотеки. В скетче «Sweep» появился новый сервопривод:

```
Servo myservo; // создаем сервообъект для управления сервоприводом
```

Сервопривод (под названием `myservo`) создан так, как и описано в библиотеке:

```
class Servo
{
private:
    uint8_t _index;
    uint8_t _pin;
    uint16_t _duty;
    static uint8_t _count;
    static Servo* _servos[];
    static int8_t _current;
    static uint16_t _positionTicks;
    static void start();
    static void end();
    static void service();
public:
    Servo();
    uint8_t attach(int);
    void detach();
    void write(int);
```

```
uint8_t read();  
uint8_t attached();  
};
```

Скрытыми называются функции, осуществляемые самой библиотекой для выполнения фоновой работы, как, например, назначение байтов. *Открытые функции* вы применяете в самом скетче.

По мере усложнения кода вас может посетить мысль о создании собственной библиотеки. Хотя я не рассматриваю этот вопрос в своей книге, у Arduino есть замечательный урок на эту тему, который можно найти на сайте <http://arduino.cc/en/Hacking/LibraryTutorial>.

Ресурсы для изучения программирования

Можно было бы с легкостью включить в книгу 10 страниц с указанием ресурсов по Arduino, но я расскажу лишь о самых интересных.

Книги¹

- *Baichtal J., Wolf A., Beckler M.* Make: Lego & Arduino Projects. Make, 2012. В книге рассматривается управление роботами LEGO MIND-STORMS с использованием Arduino.
- *Banzi M.* Getting Started with Arduino. O'Reilly, 2011. Карманный справочник для начинающих. Автор — один из основателей Arduino, поэтому можете не сомневаться в достоверности информации.
- *Kelly J. F., Timmis H.* Arduino Adventures: Escape from Gemini Station. Apress, 2013. Книга для детей. Включает великолепные проекты для Arduino, которые преподносятся с забавным юношеским мироощущением.
- *Margolis M.* Arduino Cookbook. O'Reilly, 2011. Многие считают ее самой информативной книгой по Arduino. Вам хочется узнать, как с помощью светодиодов собрать семисегментный дисплей на макетной плате? Ответ вы найдете в «Cookbook».
- *Roberts D.* Making Things Moveю TAB Electronics, 2011. Это не совсем книга по Arduino, но в ней можно найти много информации по электронике и робототехнике.

¹ Книги не переведены на русский язык. Читатель сможет найти их только по названию на английском языке. — *Прим. перев.*

Интернет-ресурсы

- *Adafruit Industries* (adafruit.com) — один из основных сайтов по электронике. Здесь есть учебные материалы, программы, а также интернет-магазин, в котором можно найти все, что нужно для создания Arduino-проектов.
- *Arduino Playground* (playground.arduino.cc). Этот ресурс я уже упоминал в главе ранее. Это кладезь программ для Arduino, как законченных, так и пробных.
- *Instructables* (instructables.com). Здесь можно найти множество учебных материалов, в том числе по электронике и Arduino.
- *Make* (makezine.com). Старая добрая площадка для сторонников DIY-движения¹. *Make* сотрудничает с *Maker Faires* (makerfaire.com) и издает журнал.
- *SparkFun* (sparkfun.com). Такой же отличный магазин электроники, как и *Adafruit*, в котором можно найти учебные материалы, видео и другие ресурсы.

В следующей главе

В главе 6 «Восприятие мира» мы продолжим изучать датчики, которые фиксируют изменения в окружающей среде и передают информацию на Arduino. Вы узнаете, как сделать «Лампу настроения», которая на основе полученных данных реагирует на окружающую среду изменением цвета светодиода.

¹ Do It Yourself («Сделай это сам»). — Прим. перев.

Восприятие мира

Один из самых мощных инструментов, который вы можете подключить к Arduino, — это сенсорный датчик, небольшое электронное устройство, которое позволяет микроконтроллеру снимать показания окружающей среды, реагируя на них в соответствии с установленной программой. Например, можно запрограммировать Arduino на включение вентилятора, когда температура повышается до определенного уровня, или на выключение лампы, когда светит солнце.

В главе 1 «Знакомство с Arduino» вы кратко познакомились с датчиками ультразвука, температуры, гибкости и освещенности. Но существует и много других датчиков! В этой главе вы узнаете о том, как их контролировать. Затем вы используете полученные знания для того, чтобы построить сенсорно управляемый проект — «Лампу настроения», которая изменяет свой цвет в зависимости от освещенности (рис. 6.1).



Рис. 6.1. Узнайте в этой главе, как сделать яркое световое настроение

Урок: датчики (сенсоры)

Как работает датчик? Мы знаем о датчиках температуры и освещенности, но ни один из существующих датчиков не соответствует такому сложному понятию, как, например, «человеческий взгляд». Нет датчика «обоняния», который должен сработать, когда кто-то готовит поблизости хот-дог. Но на что способны датчики?

Датчик — это простая конструкция. Каждый датчик, как правило, измеряет один параметр. Например, пассивный инфракрасный датчик (описан в **главе 2** «Макетирование») регистрирует резкие изменения температуры в поле обзора и распознает их как движение. Датчик давления воздуха измеряет показания так же, как барометр, и отправляет изменения к Arduino. Фоторезистор ограничивает значение силы тока в зависимости от того, какую освещенность обнаруживает резистор.

На **рис. 6.2** изображен акселерометр, который представляет собой датчик, определяющий направление и скорость объекта движения.

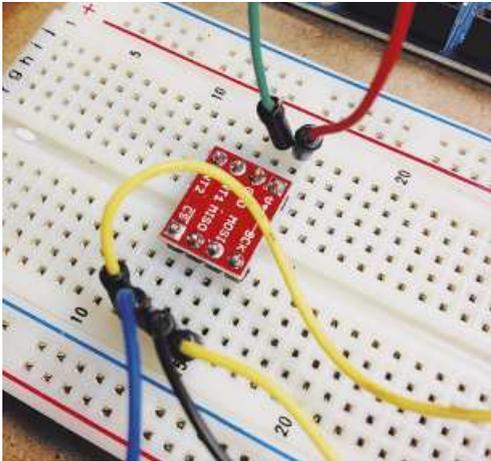


Рис. 6.2. Акселерометр может регистрировать, в каком направлении и как быстро вы движетесь

Существуют две категории датчиков: цифровые и аналоговые. Давайте рассмотрим различия между ними.

Цифровые и аналоговые данные

Существуют два способа передачи информации — *цифровой* и *аналоговый*. На платформе Arduino широко используются оба способа, поэтому вы должны их знать.

Цифровые данные

Цифровые данные состоят исключительно из «0» и «1» (см. **рис. 6.3**). Примером цифрового датчика служит акселерометр, который посылает в Arduino серию точек (скорость, направление и т. д.). Обычно цифровые датчики нуждаются в специальной микросхеме для интерпретации физических данных.

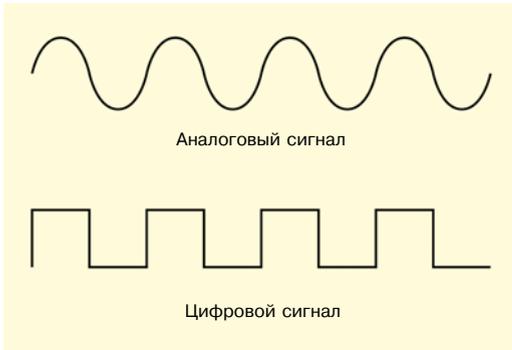


Рис. 6.3. Цифровая информация передается последовательностью чередующихся «1» и «0»; аналоговая информация представляет собой модулированный сигнал

Поскольку цифровые датчики передают только данные, они нуждаются в микроконтроллере, чтобы интерпретировать показания. Цифровой датчик нельзя использовать в качестве фоторезистора, хотя можно соорудить схему, управляющую микроконтроллером, чтобы сделать то же самое.

Аналоговые данные

Аналоговые данные передаются как непрерывный сигнал. Другими словами, аналоговый датчик не передает 1 и 0, как цифровые датчики. Вместо этого датчик модулирует непрерывный сигнал для передачи данных.

Вы можете использовать аналоговые датчики в схемах без микроконтроллера для интерпретации сигнала. Например, можно использовать аналоговый датчик света как фоторезистор.

Подключение цифровых и аналоговых датчиков

Неудивительно, что в Arduino предусмотрены контакты как для цифрового входа и выхода, так и для аналогового. Провод с данными от серводатчика для передачи данных к сервоустройству подключают к цифро-

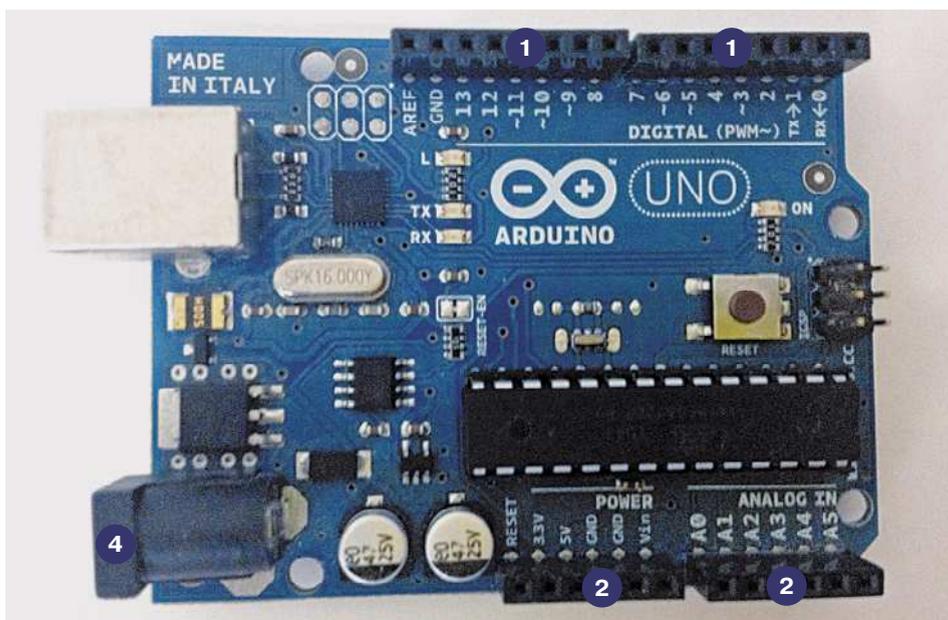


Рис. 6.4. Arduino подскажет, куда подключать ваши датчики:

1 — цифровые входы; 2 — аналоговые входы

тому входу, в то время как аналоговый датчик освещенности посылает информацию в аналоговый вход.

Как понять, где какой вход? Это легко определить, глядя на Arduino. Все цифровые входы расположены с одной стороны (верхняя часть **рис. 6.4**), а аналоговые входы — на противоположной (правая нижняя часть рисунка). Если вы забыли, просто посмотрите на подписи на самой плате.

Кроме того, при программировании скетча Arduino (так в мире Arduino называются программы) вы должны указать эти контакты. Я рассказываю, как это сделать, в **главе 5** «Программирование Arduino».

Знакомство с датчиками

Что именно представляет собой датчик? На определенном уровне это устройство, которое передает на Arduino информацию, основанную на измерении датчиком какого-либо внешнего фактора. Например, барометрический датчик (описан ниже) измеряет давление воздуха и возвращает информацию, основанную на этом измерении.

В следующих разделах будут описаны датчики, которые вы будете использовать в проектах с Arduino.

Акселерометр

На **рис. 6.5** изображен акселерометр ADXL362 в составе небольшой платы производителя *SparkFun*. Датчик передает данные в цифровом виде о том, в каком направлении и с какой скоростью он движется. Вы можете сделать забавные проекты с акселерометром, например, самобалансирующего робота, который использует акселерометр, чтобы оценить возможность опрокидывания, а затем переходит к балансировке.

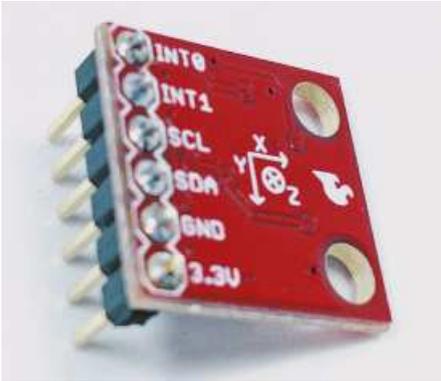


Рис. 6.5. ADXL362 *SparkFun* позволяет вам подключить акселерометр к плате

Барометр

Основной компонент метеостанций на ярмарке научных проектов учащихся — это барометрический датчик, в основном, цифровой барометр, подключенный к Arduino. Датчик, показанный на **рис. 6.6**, контролирует давление воздуха и передает показания на Arduino, используя I²C — способ передачи данных с одним проводом.

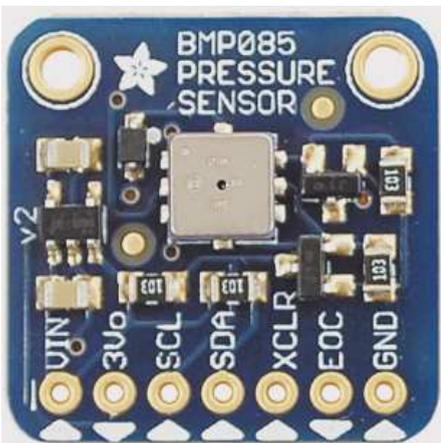


Рис. 6.6. Барометрический датчик BMP085 контролирует давление воздуха

Энкодер

Сервопривод, изображенный на **рис. 6.7**, оснащен датчиком вращения, который называется *энкодером*. Когда узел вращения двигателя поворачивается, энкодер отправляет данные обратно в Arduino для фиксации точного угла поворота. Один из вариантов использования этой особенности — создание ручки, которая управляет сервоприводом. Arduino считывает положение ручки энкодера и ждет ответного сигнала от сервопривода.



Рис. 6.7. Самодельный сервопривод передает данные через белый провод

Газовые датчики

Датчик определения содержания метана MQ4, изображенный на **рис. 6.8**, часто используется для сигнализации при утечке газа, а также хорош как газоанализатор.



Рис. 6.8. Беспокоитесь об утечке газа? Тогда эта штуковина определенно для вас

Датчик Холла

На **рис. 6.9** изображен датчик Холла, который обнаруживает присутствие магнитов. Это очень удобно для активации другого контура без необходимости находиться с ним в физическом контакте, например, если датчик отделен от цепи стеклом. Датчик отлично подходит для обнаружения объекта, который имеет встроенный магнит.



Рис. 6.9. В датчике Холла возникает разность потенциалов, когда рядом находится магнитное поле

Датчик инфракрасного излучения

В большинстве пультов дистанционного управления телевизором используют кодированные импульсы инфракрасного излучения (ИК). Например, сигнал «Выключить» имеет один код, а сигнал «Уменьшить звук» — другой. У телевизора есть ИК-датчик для получения сигнала.

Датчик ИК (см. **рис. 6.10**) можно использовать так же, как фоторезистор или датчик освещенности, за исключением того, что вы не увидите

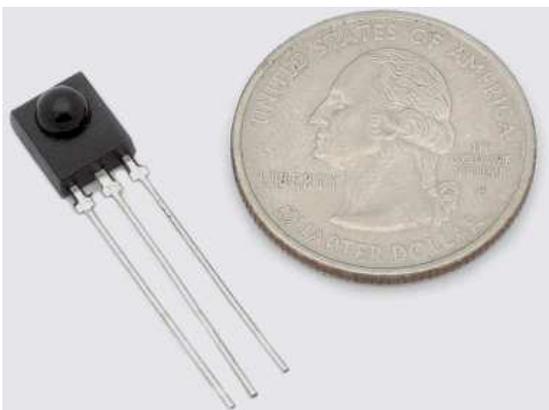


Рис. 6.10. Этот датчик определяет инфракрасное излучение

раздражающих вспышек света. Эта характеристика делает ИК-датчики полезными для проектов, в которых вам не хочется устанавливать яркие светодиоды или если не нравится мерцание света при неправильном запуске датчика.

Например, можно разместить ИК-датчик на роботе рядом с ИК-светодиодом, используемым в качестве датчика приближения. Когда луч светодиода отражается от объекта поблизости, датчик улавливает его и посылает сигнал на Arduino.

Обычно ИК-датчики выглядят как черные коробочки с выступающей лампочкой и тремя проводками.

Пьезоэлемент (датчик детонации)

Некоторые электронные детали могут работать как излучатели и как приемники. Например, светодиод работает после подачи небольшого напряжения, а его яркий свет в свою очередь генерирует крошечный импульс напряжения. Пьезозуммер, как часто называют датчик детонации, работает точно так же (см. **рис. 6.11**). Если вы посылаете напряжение через пьезоэлемент, он вибрирует и жужжит. Если вручную встряхивать пьезоэлемент (или периодически нажимать на него), генерируется небольшое количество напряжения. Это означает, что вы можете использовать пьезоэлемент в качестве датчика вибрации!

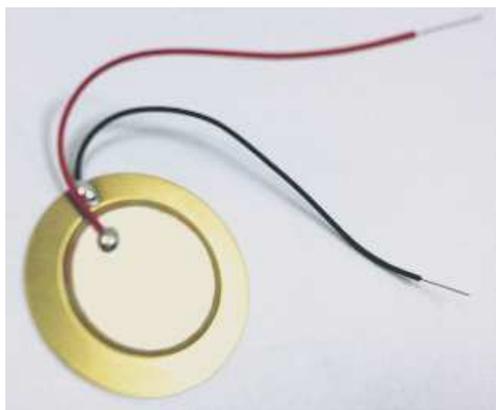


Рис. 6.11. Датчик вибрации очень похож на пьезоизлучатель, потому что это он и есть!

Датчики звука

Датчик звука (**рис. 6.12**) — это по сути микрофон. Звуковые колебания генерируют напряжение, которое улавливает датчик. Вы часто используете усилитель для динамика. Однако можно использовать датчик звука для запуска действий на Arduino без усилителя.



Рис. 6.12. Датчик звука, извлеченный из сломанной игрушки

Датчики наклона

Датчик наклона (**рис. 6.13**) представляет собой трубку с заваляцованным в ней шариком. Когда датчик находится в вертикальном положении, шарик соединяет два провода, но, если датчик наклонить, шарик откатывается от контактов и связь разрывается. По этой причине датчик иногда называют акселерометром для бедных.

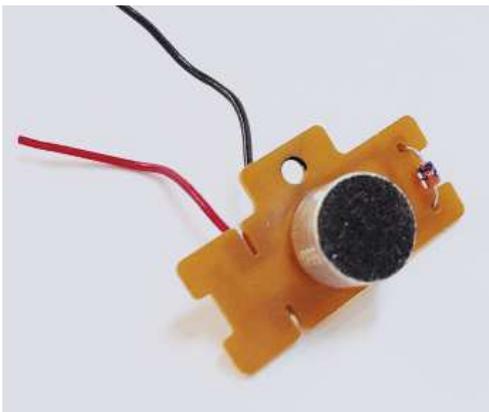


Рис. 6.13. Датчик наклона сообщает о своем положении

🔧 Проект: «Лампа настроения»

В этом проекте вы создаете устройство под названием «Лампа настроения» (рис. 6.14), на основе модуля «ShiftBrite», очень яркого RGB-светодиода, установленного на печатной плате, полезной для применения такого рода. Управление светодиодом осуществляется с помощью Arduino UNO (конечно!), подключенного к небольшой солнечной панели, используемой в качестве датчика освещенности, к датчику температуры и влажности воздуха, и к небольшому микрофону, выступающему в качестве датчика звука. Для постройки корпуса вы погрузитесь в мир *керфбэндинга* — это специальная техника прорезания щелей в деревянной панели, которые позволяют согнуть панель, не нарушая ее целостности.



Рис. 6.14. Датчик температуры и влажности воздуха (белый пластиковый модуль) и датчик звука (маленькая кнопка в середине изображения) помогают контролировать цвет лампы

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Вам понадобятся следующие комплектующие детали, чтобы построить «Лампу настроения».

- Arduino UNO.
- Модуль «ShiftBrite»: см. раздел «Для справки» после перечня для подробной информации.

- Датчик освещенности (вы можете использовать любой датчик освещенности или солнечную панель).
- Датчик температуры и влажности воздуха DHT22.
- Электретный микрофон, используется в качестве датчика звука.
- Абажур для «ShiftBrite» (я использовал стеклянный плафон от лампы для газонного освещения).
- Болты с гайками № 4-40 × 25 мм, 6 шт.
- Пластиковые втулки 9 мм, 4 шт.
- Небольшая переходная плата.
- Перемычки в ассортименте.
- Лист ДВП или МДФ размером 45 × 60 см.

ДЛЯ СПРАВКИ

МОДУЛЬ «SHIFTBRITE»

«ShiftBrite» — это светодиодный модуль, созданный Гарретом Мейсом и позволяющий поклонникам Arduino очень точно управлять RGB-светодиодом высокой яркости (**рис. 6.15**). RGB представляет красный, зеленый и синий цвета, которые в сочетании могут отображать любой цвет спектра. «ShiftBrite», следовательно, состоит из трех элементов, по одному для каждого цвета, и Arduino управляет значениями всех трех.



Рис. 6.15. Плата «ShiftBrite» обслуживает один RGB-светодиод

Можно только восхищаться тем, как Arduino управляет цветами и яркостью «ShiftBrite». Управление осуществляется с помощью последовательной связи, которая рассматривалась в **главе 4** «Настройка беспроводного соединения». Обычно Arduino управляет светодиодом путем изменения напряжения. На практике управлять большим количеством светодиодов непросто, поэтому Arduino имеет много выводов.

Решение заключается в использовании светодиодного драйвера, микрочипа, который управляет несколькими светодиодами одновременно. Драйвер принимает команды от Arduino с помощью последовательного соединения, где цифровыми компонентами, подобными «ShiftBrite», можно управлять с помощью нескольких проводов. Угадайте, что это? Это A6281, крошечный светодиодный драйвер, который управляет светодиодом «ShiftBrite». Этот драйвер не только позволит вам использовать данные, чтобы изменить цвет и яркость светодиода, но и подключить несколько «ShiftBrite» в ряд. Вся конструкция управляется теми же четырьмя штырьками данных, которые требуются для управления единичным модулем.

Вы можете узнать больше о «ShiftBrite» на странице производителя модуля по адресу: <http://macetech.com/blog/node/54>.

Инструкции

Давайте строить! Собрать «Лампу настроения» относительно просто, но есть пара сложных операций.

- 1. Вырежьте и соберите корпус.** Я использовал ДВП толщиной 5 мм и вырезал форму на лазерном резаке, используя технику под названием «керфбэндинг» («надрезать и согнуть») (см. «Альтернативный проект: керфбэндинг» в конце этой главы). Вы можете скачать файл, который я использовал на сайте <https://github.com/n1/Arduino-For-Beginners>. Кроме того, вы можете создать собственный корпус! Я вырезал большое отверстие для модуля питания и выемки вдоль верхних ребер для крепления верхней части корпуса (**рис. 6.16**).
- 2. Постройте платформу для светодиода.** Добавьте к корпусу светодиодную сборку. Я понял, как прикрепить плафон к деревянной коробке (см. шаг 5). Основной элемент крепления — небольшая панель, которая соответствует плафону по ширине и стягивается парой винтов (**рис. 6.17**).
- 3. Прикрепите «ShiftBrite».** Приклейте мини-плату к панели, используя двусторонний скотч, клейкую подложку панели или тер-



Рис. 6.16. Проект корпуса использует вид резки, который называется «керф-бэндинг» («надрезать и согнуть»), позволяющей древесине сгибаться по кругу для формирования стенок

моклей. Подключите «ShiftBrite», а затем подключите провода к соответствующим отверстиям на плате, как сказано в шаге 4.

- 4. Выполните проводные соединения «ShiftBrite».** На первый взгляд схема подключения «ShiftBrite» выглядит пугающей, но не тревожьтесь! На самом деле все довольно легко. Вам необходимо подключить всего шесть проводов.

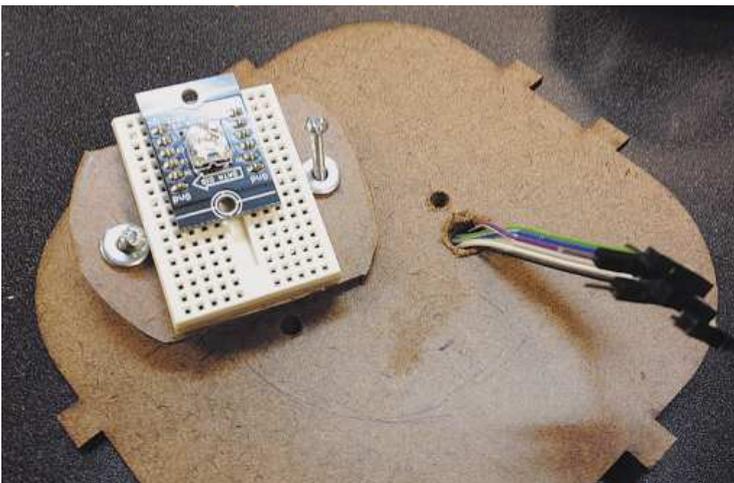


Рис. 6.17. «ShiftBrite» находится в плафоне на платформе с мини-платой для управления проводами



Рис. 6.18. Для моего проекта я использовал плафон от уличного светильника

- **V+** на «ShiftBrite» подключается к пину, помеченному как **5V** на Arduino.
 - **DI** на «ShiftBrite» подключается к пину 10.
 - **LI** подключается к пину 11.
 - **EI** подключается к пину 12.
 - **SI** подключается к пину 13.
 - **GND** подключается к пину GND.
5. **Прикрепите плафон.** Я использовал плафон от лампы, которые продаются во многих магазинах (**рис. 6.18**). Я разобрал лампу светильника и отложил в сторону все, кроме плафона. Чтобы прикрепить плафон, я просверлил два отверстия для винтов в верхней панели корпуса и большое отверстие для проводов. Я использовал небольшую дощечку, подходящую по размерам к плафону, для верхней панели корпуса, как показано ранее на **рис. 6.17**. Чтобы зафиксировать плафон на крышке, затянул болты.
 6. **Прикрепите Arduino к корпусу.** После того как вы прикрепили плафон и подключили «ShiftBrite», прикрепите и подключите Arduino. Я прикрепил Arduino к коробке с помощью болтов с резьбой № 4 через отверстия, просверленные в нижней панели, и укрепил плату пластиковыми втулками размером 10 мм. См. **рис. 6.19**.
 7. **Подключите датчики.** Датчик освещенности подключите к аналоговому пину 1 и GND (см. **рис. 6.20**). Датчик температуры подключите к пину 3V3 и аналоговому пину 3. Заземление совмещается с датчиком звука, который подключается в аналоговый пин 2. Это не слишком сложно!

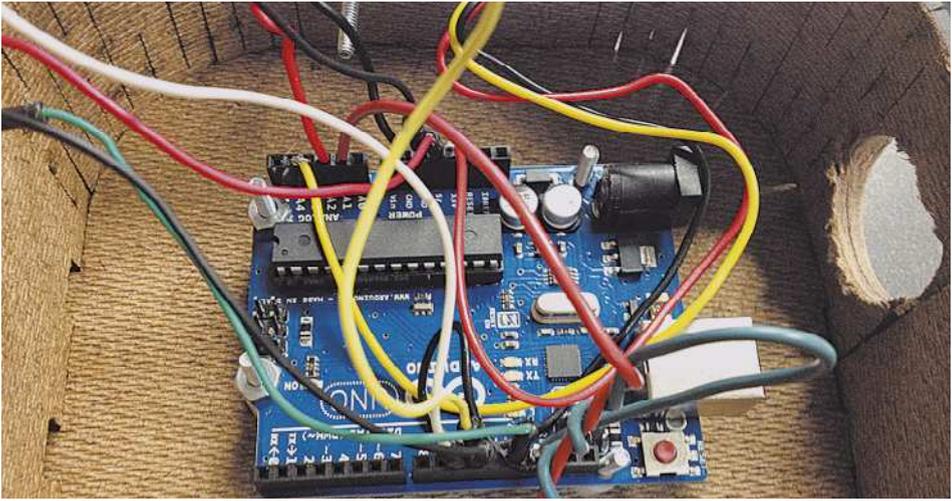


Рис. 6.19. Arduino, все подключено!

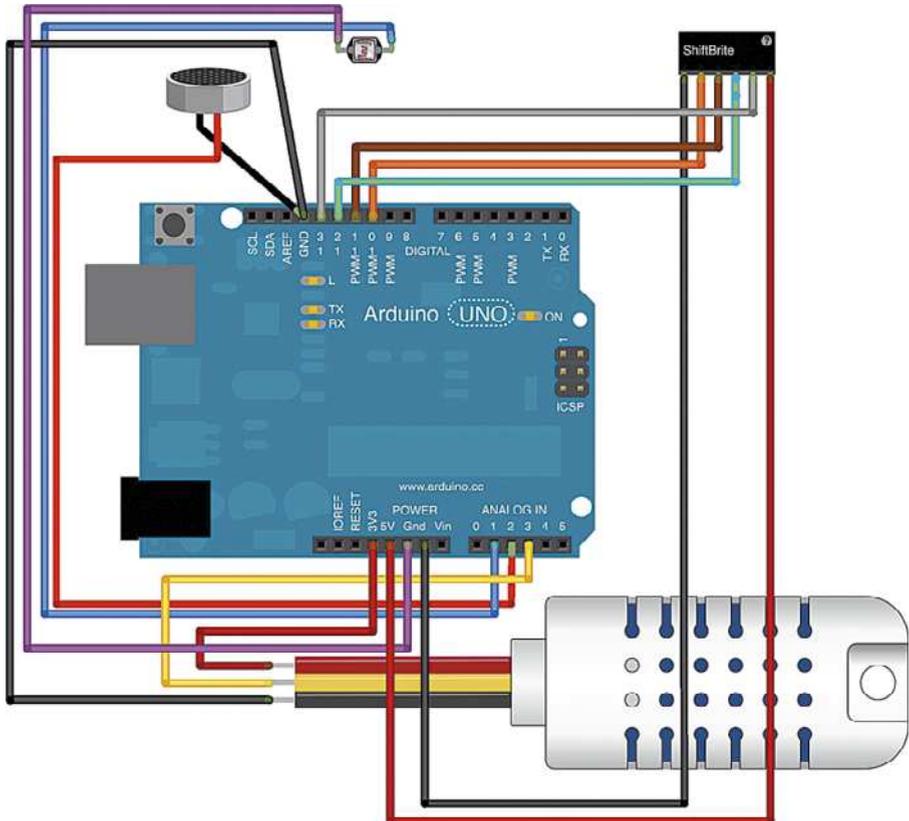


Рис. 6.20. Это куча проводов!

Код для «Лампы настроения»

Вот код для «Лампы настроения» Arduino. Естественно, вам захочется настроить код в зависимости от обстановки. Например, светлая комната может потребовать более высокого порога для отображения. Я покажу вам, как это сделать в эскизе в строке «настроить допуски здесь» («adjust tolerances here»). Все, что вам нужно сделать, — это изменить номер данной записи, чтобы изменить то, как «Лампа настроения» реагирует на состояние окружающей среды.

И напоследок: если вы не помните, как загрузить эскиз в Arduino, обратитесь к **главе 5**.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
// этот код основан на примере кода Гаррета Мэйса,  
// который взят с сайта macetech.com  
  
int datapin = 10; // DI  
int latchpin = 11; // LI  
int enablepin = 12; // EI  
int clockpin = 13; // CI  
int pspin = A1; // фотодатчик  
int sspin = A2; // датчик звука  
int thspin = A3; // датчик температуры и влажности  
  
int light = 0;  
int sound = 0;  
int temp = 0;  
  
unsigned long SB_CommandPacket;  
int SB_CommandMode;  
int SB_BlueCommand;  
int SB_RedCommand;  
int SB_GreenCommand;  
  
void setup() {  
    pinMode(datapin, OUTPUT);  
    pinMode(latchpin, OUTPUT);  
    pinMode(enablepin, OUTPUT);  
    pinMode(clockpin, OUTPUT);  
    pinMode(pspin, INPUT);
```

```
pinMode(sspin, INPUT);

digitalWrite(latchpin, LOW);
digitalWrite(enablepin, LOW);

    Serial.begin(115200); // устанавливает последовательное
                        // соединение
}

void SB_SendPacket() {
    SB_CommandPacket = SB_CommandMode & B11;
    SB_CommandPacket = (SB_CommandPacket << 10) | (SB_BlueCommand & 1023);
    SB_CommandPacket = (SB_CommandPacket << 10) | (SB_RedCommand & 1023);
    SB_CommandPacket = (SB_CommandPacket << 10) | (SB_GreenCommand & 1023);

    shiftOut(datapin, clockpin, MSBFIRST, SB_CommandPacket >> 24);
    shiftOut(datapin, clockpin, MSBFIRST, SB_CommandPacket >> 16);
    shiftOut(datapin, clockpin, MSBFIRST, SB_CommandPacket >> 8);
    shiftOut(datapin, clockpin, MSBFIRST, SB_CommandPacket);

    delay(1);
    digitalWrite(latchpin, HIGH);
    delay(1);
    digitalWrite(latchpin, LOW);
}

void loop() {

    light = analogRead(pspin) + 50; // настроить допуски здесь
    sound = analogRead(sspin) + 50; // настроить допуски здесь
    temp = analogRead(thspin) / 50; // настроить допуски здесь

    SB_CommandMode = B01; // укажите регистры управления током
    SB_RedCommand = 127; // полный ток
    SB_GreenCommand = 127; // полный ток
    SB_BlueCommand = 127; // полный ток
    SB_SendPacket();

    delay(2500);

    SB_CommandMode = B00; // укажите регистры управления
                        // широтно-импульсной модуляцией (ШИМ)
    SB_RedCommand = sound;
    SB_GreenCommand = light;
    SB_BlueCommand = temp;
    SB_SendPacket();

    Serial.println("Light: "); Serial.println(light);
    Serial.println("Sound: "); Serial.println(sound);
    Serial.println("Temp: "); Serial.println(temp);
}
```

⚙ Проект: керфбэндинг

Керфбэндинг — это умный способ сгибания древесины. Лучше всего использовать лазерный резак, но даже с обычной пилой можно получить прекрасный результат. Техника включает создание серии надрезов в материале близко друг к другу, что в конечном счете делает дерево тонким и гибким в определенных местах. Одно из применений этой технологии — сделать коробку с помощью пропилов для сгибания углов (**рис. 6.21**).

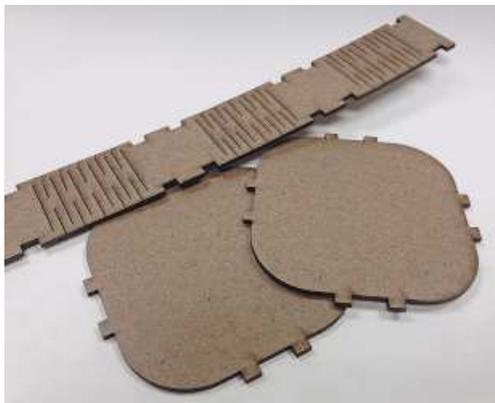


Рис. 6.21. Прорези, сделанные с помощью резака в этом ДВП (здесь закругленные), позволяют древесине гнуться

Если вы хотите создать коробку с помощью лазерной резки, у вас есть несколько вариантов.

- Вы можете загрузить дизайн коробки с сайта Thingiverse.com. Это ресурс для людей, использующих 3D-принтеры и лазерные резаки, и все проекты могут быть адаптированы. Например, дизайн коробки, который я использую в этой главе, — это адаптированный вариант макета № 12707 в «SNIJLAB’s Folding Wood Booklet design».
- Сервис MakerCase.com. Этот сайт позволяет ввести размеры вашей коробки, а также выбрать сведения о материале, такие как толщина и тип соединителя, а затем сервис генерирует для вас лазерные файлы. Просто здорово!

В следующей главе

В **главе 7** «Управление жидкостью» вы узнаете, как использовать Arduino для включения и остановки потока жидкости. Затем вы построите вместе с LEGO классного робота для полива растений.

Управление жидкостью

Вода и электроника — понятия несовместимые. Ну, в целом. Но в этой главе вы откроете три способа управления жидкостью с помощью электричества.

1. **Электромагнитный (соленоидный) клапан** — клапан, который открывается при подаче электрического напряжения.
2. **Емкость под давлением** — это контейнер с жидкостью, которая нагнетается с помощью воздушного насоса, заставляющего жидкость вытекать из выходной трубки (см. **рис. 7.1**).
3. **Перистальтический насос**, периодически сдавливающий трубку и заставляющий жидкость течь через нее, избегая соприкосновения с жидкостью.

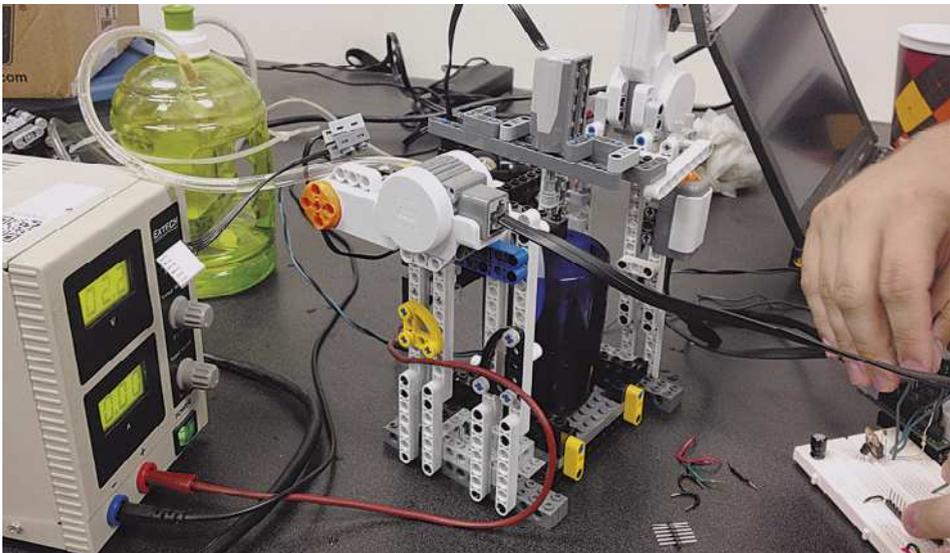


Рис. 7.1. Этот робот LEGO использует резервуар под давлением (зеленая емкость) для нагнетания шоколадного молока

После освоения теории этих методов (все они программируются Arduino) перейдем к практике. Вы построите робота, который будет поливать ваши растения по расписанию.

Урок: управление потоком жидкости

Давайте пробежимся по трем способам запуска и остановки потока жидкости.

Электромагнитный клапан

Начнем с электромагнитного клапана от компании *Adafruit*, который запускается с помощью команды от Arduino (см. **рис. 7.2**). Исходно клапан закрыт, но когда вы подаете напряжение 12 В, электромагнит срабатывает, клапан открывается, и вода начинает течь. Ура! Да здравствует прогресс! Как вы помните из **главы 1** «Знакомство с Arduino», в электромагните и двигателе используется один и тот же принцип передачи энергии. Но в двигателе ось вращается, а в электромагните перемещается поступательно. Получается, что электромагнит открывает и закрывает клапан.

Клапан включается в водопроводную систему, состоящую из пластиковых труб и соединителей диаметром 12,5 мм, которые вы можете купить в любом магазине сантехники. Я остановился именно на этом клапане для робота-поливалки из-за его гибкости. Это удобство таит в себе неоспоримую выгоду — вы сможете подключать вашего робота напрямую к основной системе водоснабжения, вместо того чтобы возиться с емкостью для воды.



Рис. 7.2. Электромагнитный клапан срабатывает, когда на него подается напряжение

Если вы выберете один из этих клапанов, помните, что для его точной работы необходимо давление жидкости примерно 0,2 атм и то, что клапан работает только в одном направлении.

Емкость под давлением

Недостатком электромагнитного клапана является его небезопасность с пищевой точки зрения (см. далее раздел «Безопасность пищевых продуктов»). Это означает, что вы не должны пользоваться клапаном при обработке продуктов и изготовления напитков. Подача жидкости под давлением воздуха с использованием резервуара решает эту проблему. Наиболее простой пример использования емкости под давлением — водяной пистолет (см. **рис. 7.3**).

Вот как работает резервуар и другие емкости под давлением.

1. Закрытый контейнер с жидкостью оснащен двумя трубками: одна расположена чуть выше уровня жидкости, а другая — вплоть до нижней части контейнера.
2. Когда воздух поступает через верхнюю трубку, он создает давление в контейнере, заставляя жидкость вытекать из второй трубки.
3. Поскольку эта трубка находится ниже поверхности жидкости, вода вытесняется.



Рис. 7.3. Ну что, «емкость под давлением» звучит знакомо? Да, это так! Это метод, с помощью которого многие водяные пистолеты выстреливают воду

Благодаря возможности резервуара работать с напитками он часто используется в дринкботах (*drinkbots*), также известных как барботы (*barbots*). Дринкботы (барботы) — это роботы, запрограммированные на приготовление коктейлей. Они наливают жидкость под давлением и смешивают ингредиенты в определенных количествах. Резервуаром является бутылка (например, рома), которая соединяется с воздушным насосом (компрессором).

Перистальтический насос

Другое устройство, безопасное с пищевой точки зрения, — это перистальтический насос. В нем используется двигатель для сдавливания трубки. Циклическое давление на трубку перемещает по ней жидкость. Это похоже на то, как работает наш пищеварительный тракт, заставляя пищу продвигаться по нему. Чаще всего вы можете видеть перистальтический насос в аппаратах для дойки коров — молоко проходит через безопасную (с пищевой точки зрения) трубку, оставаясь чистым для употребления.

Как и в случае с сосудом под давлением, изобретатели используют эти насосы, чтобы сделать дринкботы (роботы для разлива жидкостей). Перистальтические насосы стоят дороже, но они более управляемы с точки зрения запуска и остановки жидкости.

Насос, изображенный на **рис. 7.4**, можно приобрести у компании *Adafruit Industries*. Фактически это двигатель постоянного тока на 12 В со скоростью 5000 об./мин. Он имеет клеверообразную втулку, которая периодически сжимает трубку и заставляет жидкость продвигаться через нее.

Источник: Adafruit Industries



Рис. 7.4. Перистальтический насос ритмично сжимает резиновую трубку

⚙ Проект: емкость под давлением

Давайте построим емкость под давлением, управляемым Arduino! Вы используете аквариумный компрессор, чтобы создать давление в емкости для молока, как показано на **рис. 7.5**. Давление выталкивает содержимое емкости через выходную трубу. Вуаля, насос!

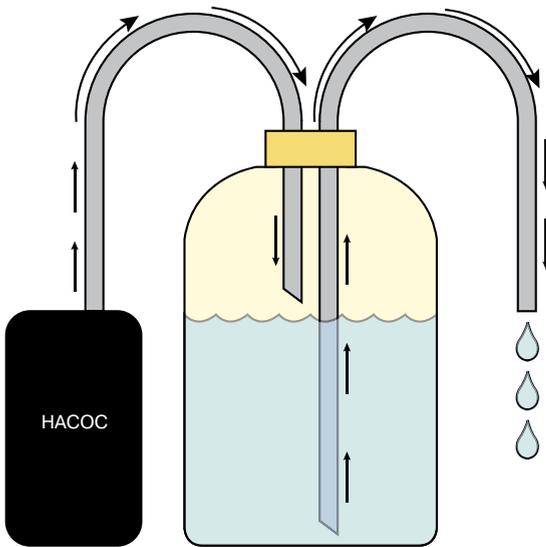


Рис. 7.5. Емкость под давлением вытесняет жидкость с помощью воздуха, нагнетаемого в контейнер

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

- Arduino.
- Емкость для молока; удобно использовать обычную трех- или пятилитровую канистру для воды.
- Насос для аквариума, работающий от батареек.
- Два отрезка трубки (я использовал трубку для напитков Tugon В-44-3 с внешним диаметром 6 мм).
- Транзистор Дарлингтона TIP-120 (см. **главу 13** «Управление электродвигателями», чтобы узнать о нем подробнее).
- Резистор 2,2 кОм.
- Диод 1N4001.
- Провод.
- Дрель и сверло 6 мм.

Инструкции

1. Просверлите два отверстия в крышке емкости для воды с помощью сверла диаметром 6 мм.
2. Пропустите трубки через отверстия. Конец одной трубки опустите до дна емкости. Конец другой трубки должен находиться над поверхностью воды. Посмотрите на **рис. 7.5** для того, чтобы понять, как правильно расположить трубки.
3. Подключите компрессор для аквариума так, как показано на **рис. 7.6**. В принципе, вы меняете тумблер компрессора на транзистор Дарлингтона — разновидность электронного переключателя, активируемого с помощью Arduino.
 - А. Соедините проводом пин 13 Arduino и левый пин транзистора, расположив между ними резистор. Это синий провод на **рис. 7.6**.
 - Б. Подключите источник энергии к положительной клемме насоса. Это зеленый провод на **рис. 7.6**.

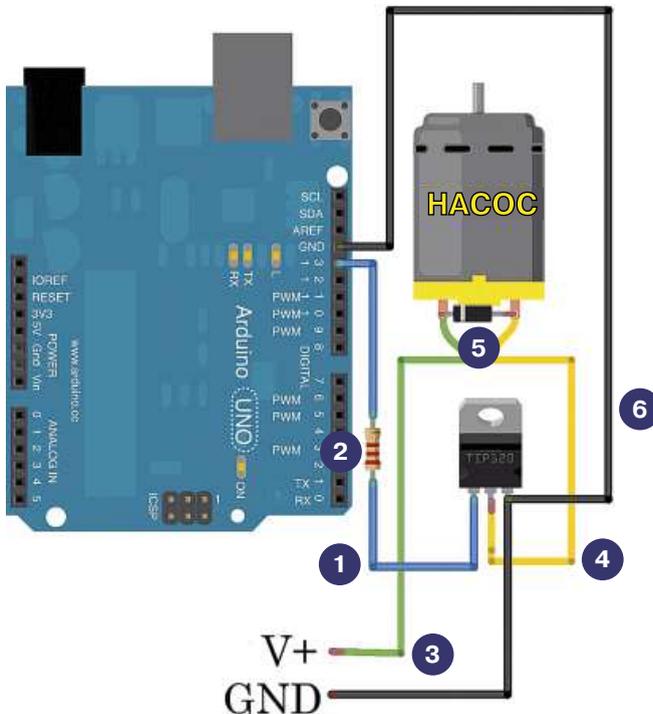


Рис. 7.6. Подключите ваш насос так, как показано здесь!

1 — проволочная перемычка; 2 — резистор; 3 — источник питания; 4 — средний пин транзистора соединен с отрицательной клеммой насоса; 5 — диод припаян между клеммами «+» и «-»; 6 — правый пин транзистора соединен с отрицательной клеммой на источнике питания и с одним из пинов GND на плате Arduino

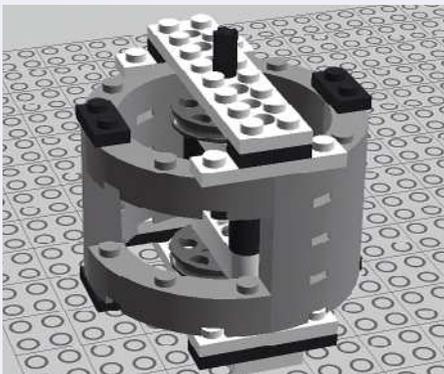
- В. Подключите средний пин транзистора к отрицательной клемме насоса. Это желтый провод на **рис. 7.6**.
- Г. Припаяйте диод между положительной и отрицательной клеммами насоса. Этот диод защищает от обратного тока от двигателя, который может воспламенить ваш Arduino! Диод имеет полярность, поэтому должен быть подключен правильно. На **рис. 7.6** показано, как это сделать.
- Д. Подключите правый пин транзистора к отрицательной клемме источника питания и к одному из пинов GND на плате Arduino. Это черный провод на **рис. 7.6**.

Код для емкости под давлением

Я собираюсь пропустить программирование, потому что все, что вам нужно, чтобы запустить насос, — это одиночный импульс от вашего Arduino. Я предлагаю использовать скетч «Blink» (для дополнительной информации см. главу 5 «Программирование Arduino») и изменение задержки от 1000 до 10 000. Благодаря им насос включится на 10 с и затем выключится на тот же промежуток времени с повторением цикла до тех пор, пока насос не выключится окончательно.

ПЕРИСТАЛЬТИЧЕСКИЙ НАСОС LEGO

Мигель Валензуэла построил из кирпичиков LEGO робота, выпекающего блины. Робот просто прелесть! Он может «печатать» буквы и геометрические фигуры из жидкого теста для блинов. Совершенно справедливо Мигель решил, что ему необходим робот для наливания сиропа в компанию к роботу для выпечки блинов. Поэтому он построил перистальтический насос из кирпичиков LEGO (см. **рис. 7.7**), который находится рядом с «Блинным роботом» (PancakeBot), готовый для наливки яблочного сиропа на ваши «напечатанные» блины.



Источник: Miguel Valenzuela

Рис. 7.7. Дозатор сиропа Мигеля Валензуэла — на самом деле перистальтический насос

ТЕХНИКА БЕЗОПАСНОСТИ

БЕЗОПАСНОСТЬ ПИЩЕВЫХ ПРОДУКТОВ

Работая с жидкостью, важно помнить о том, что она должна соприкасаться только с «безопасными для пищевых продуктов» поверхностями, если вы собираетесь потом пить. Но что это значит на самом деле?

В каждой стране свои нормативные документы о безопасности пищевых продуктов, но обычно безопасная (с пищевой точки зрения) поверхность должна быть свободной от загрязнений, представляющих опасность для человека, который ест или пьет с этой поверхности, и что при обычном использовании поверхность не должна разрушаться в результате жизнедеятельности загрязнителей.

Большинство производителей ставят на своей продукции знак «безопасно с пищевой точки зрения». Если на продукции этот знак отсутствует, считайте, что они НЕбезопасны.

Проект: робот для полива растений

Выполняя этот проект, вы сделаете робота, который будет поливать растения по заданному расписанию, освободив вас для другой полезной работы! В работе используется электромагнитный клапан (описанный ранее в этой главе), подключенный к системе подачи воды (см. **рис. 7.8**). Клапан управляется Arduino, который ежеминутно проверяет, не пора ли подавать воду. Наконец, вы построите изящный корпус LEGO для защиты Arduino от водяных брызг!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для того чтобы построить робота для полива растений, вам необходимы следующие комплектующие части.

- Arduino UNO.
- Электромагнитный клапан (я использовал клапан компании *Adafruit*, который имеет внутреннюю резьбу, соответствующую диаметрам ПВХ-фитингов 6 мм).
- Транзистор Дарлингтона TIP-120.
- Резистор 2,2 кОм.
- Батарейка 9 В.
- Разъем для батарейки 9 В.
- Диод 1N400.
- Провод.
- Прямоугольный отвод ПВХ без резьбы (гладкий).
- Прямоугольный отвод ПВХ с резьбой.



Рис. 7.8. Полейте это изнывающее от жажды растение с помощью удобного робота-поливальщика

- Тройной отвод ПВХ.
- Трубы ПВХ диаметром 12 мм: Я использую их следующей длины: 35 см, 53 см, а также три отрезка по 6 см.
- Клей для ПВХ.
- Муфта с обычной резьбой, 2 шт.
- Заглушка ПВХ.
- Переходник ПВХ для садового шланга.
- Деревянный черенок диаметром 18 мм, заточенный с одного конца.

ПВХ (ПОЛИВИНИЛХЛОРИД)

Немало производителей буквально молятся на ПВХ (см. **рис. 7.9**). Эти белые (обычно) пластиковые трубы используются в сантехнических целях в домашнем хозяйстве. ПВХ отлично подходит для циклического движения жидкости, но также находит множество применений в качестве строительного материала. ПВХ легко клеить, легко согнуть, если нагреть, и этот материал не протекает.

Источник: Miguel Valenzuela



Рис. 7.9. Поливинилхлорид (ПВХ) – это распространенный материал для обустройства дома и одновременно удобный материал для разнообразных поделок!

Инструкции

Начнем сборку вашего робота для полива растений из труб ПВХ.

1. Возьмите тройной отвод и присоедините к нему три отрезка трубы длиной 6 см так, как показано на **рис. 7.10**. Склейте части клеем. Отвод должен быть расположен так, чтобы один отрезок трубы был направлен вниз, один – вверх и один – назад.
2. Вставьте деревянный черенок в трубу, направленную вниз от тройного отвода. Черенок диаметром 18 мм плотно войдет внутрь трубки ПВХ диаметром 12 мм. Для надежности смажьте соединение

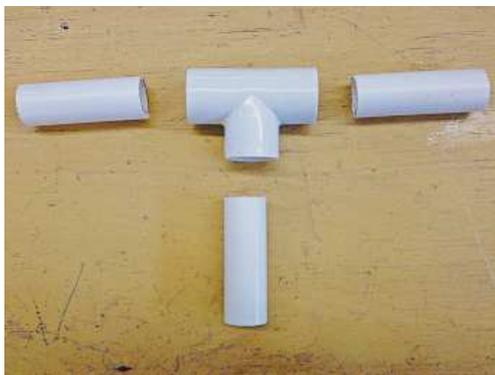


Рис. 7.10. Начинаем склеивать части ПВХ



Рис. 7.11. Приклеенный черенок подобен якорю

небольшим количеством клея. (Я так и не понял, почему черенок диаметром 18 мм меньше, чем трубка диаметром 12 мм. Математика, порой ты сводишь меня с ума!)

3. Вам понадобится заострить один конец черенка (я использовал для этого шлифовальный диск), чтобы древко с легкостью вошло в землю и удерживало всю конструкцию ровно. Вся сборка должна выглядеть так, как на **рис. 7.11**.
4. Присоедините переходник для садового шланга к трубке тройного отвода, направленной назад, закрепив его для надежности клеем. Вы можете видеть переходник на **рис. 7.11**.
5. Теперь прикрепляем клапан. Прикрутите две муфты к концам клапана так, как показано на **рис. 7.12**. Обратите внимание, что резьба на изделиях из ПВХ быстро изнашивается. После того как вы закрутили резьбу, обратное откручивание будет затруднительным, поэтому вы должны быть абсолютно уверены в ваших действиях.

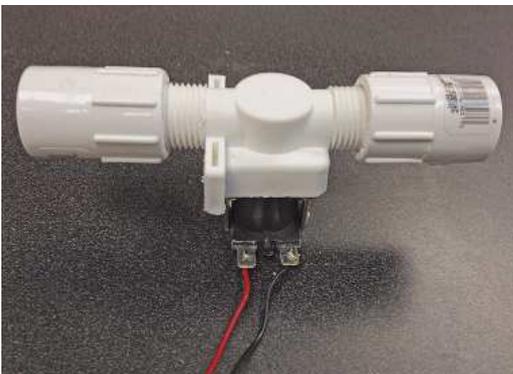


Рис. 7.12. Электромагнитный клапан в сборке с муфтами



Рис. 7.13. Прикрепление угловых отводов к каждому концу трубы длиной 35 см

6. Когда муфты надежно закреплены на клапане, приклейте комплект труб с тройником с нижнего конца клапана и комплект опрыскивателя с верхнего конца клапана. Но как понять, где верх, а где низ? Клапан работает в одном направлении, поэтому вы должны быть уверены, что входное отверстие находится снизу. Для этого найдите небольшую стрелку на корпусе клапана и убедитесь, что она направлена вверх.
7. Возьмите отрезок трубы длиной 35 см и приклейте один прямоугольный отвод к одному концу трубы и другой — к другому концу. На **рис. 7.13** показано, как это должно выглядеть.
8. Приклейте трубу длиной 53 см к гладкому (без резьбы) прямоугольному отводу.
9. А сейчас сделайте форсунку. Я взял стандартную заглушку ПВХ с резьбой и просверлил в ней несколько отверстий (см. **рис. 7.14**). Размер отверстий зависит от количества воды, которое должно вытекать из форсунки.

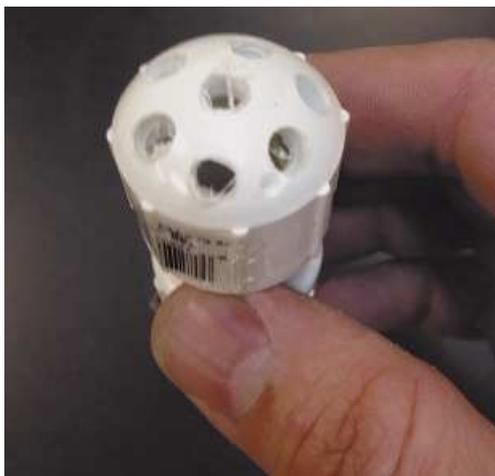


Рис. 7.14. Просверлите отверстия в заглушке, чтобы сделать разбрызгиватель



Рис. 7.15. Уже проглядывается макушка робота!

10. Прикрепите форсунку. Обратите внимание на **рис. 7.15**: я добавил небольшой модуль-удлиннитель, но это необязательно.

Электроника робота для полива растений

Теперь пришло время подключить электронику. Во многом это похоже на подключение емкости под давлением, описанной ранее в этой главе (см. **рис. 7.16**).

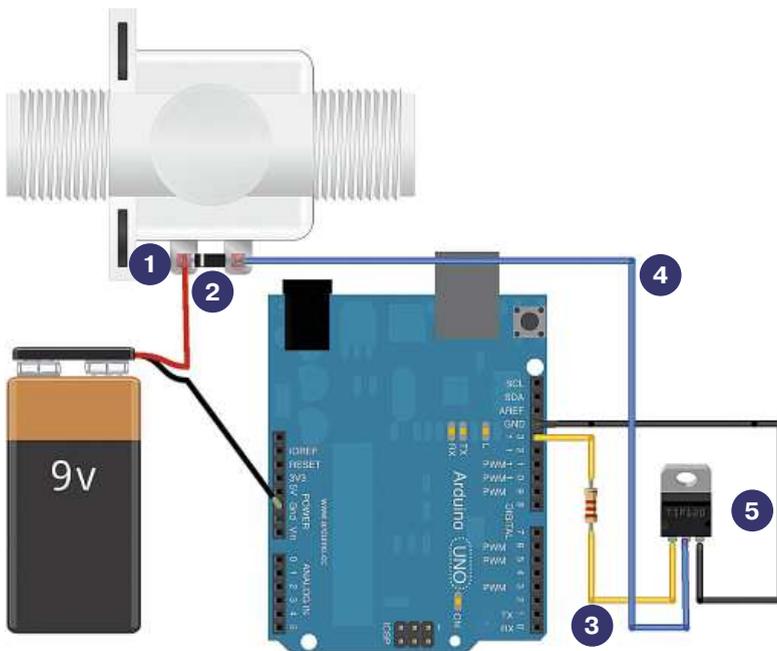


Рис. 7.16. Эта схема показывает, как подключить клапан:

1 — соедините клемму «+» батарейки с клеммой «+» электромагнитного клапана; 2 — соедините клеммы «+» и «-» клапана с помощью диода; 3 — соедините пин 13 на Arduino с левым пином транзистора; 4 — соедините клемму «-» клапана с центральным пином транзистора; 5 — соедините пин GND на плате Arduino с правым пином транзистора

Вот что вам нужно сделать.

1. Соедините проводом положительную клемму батареи и положительную клемму электромагнитного клапана. Это красный провод на **рис. 7.16**.
2. Соедините положительную и отрицательную клеммы клапана с помощью диода 1N4001. Полоска на диоде должна совпадать с положительной клеммой клапана.
3. Соедините проводом последовательно пин 13 на Arduino, резистор 2,2 кОм и левый пин транзистора. Это желтый провод на **рис. 7.16**.
4. Соедините отрицательную клемму электромагнитного клапана с центральным пином транзистора. Это синий провод на **рис. 7.16**.
5. Подсоедините правый пин транзистора к пину GND на Arduino. Это черный провод на **рис. 7.16**.

Корпус робота для полива растений

Сейчас вы можете построить из LEGO корпус, который скроет от глаз Arduino и другую электронику. На **рис. 7.17** показан дизайн корпуса.

Инструкции для сборки корпуса, показанного на **рис. 7.17**, можно найти в файле LEGO Digital Designer (LDD) по адресу: <https://github.com/n1/Arduino-For-Beginners>.

Вы можете бесплатно загрузить LDD с сайта ldd.lego.com, и он загрузится на любой платформе PC или Mac, ой, извините, Linux! Файл включает 3D CAD-рисунок модели, и вы можете дать команду LDD создать программную модель шаг за шагом прямо для вас. Это здорово! На самом деле там два файла — для крышки корпуса и для днища. Инструкции

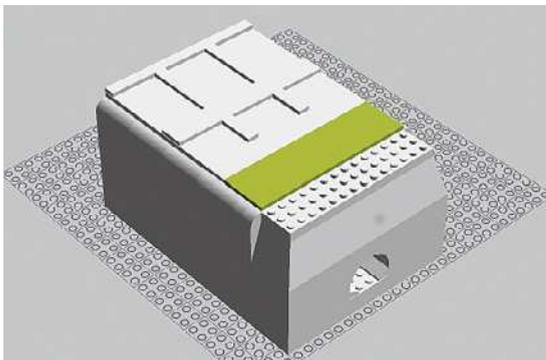


Рис. 7.17. Корпус из LEGO, смоделированный на компьютере

показывают вам, как собрать корпус из двух частей так, чтобы он мог защитить PVC.

Во время сборки корпуса вам захочется склеить кирпичики вместе. Фанаты LEGO ненавидят склеивать свои модели. А вы должны помнить о том, что внутренность корпуса должна всегда оставаться сухой, поэтому склеивать надо. Склейка к тому же предохранит корпус от случайного разрушения.

Размещение электроники

Вы должны прикрепить Arduino, батарейку и связанные с ними электронные компоненты к корпусу. Один из вариантов — использовать стяжки, чтобы прикрепить плату к одной из LEGO-деталей для техники (LEGO's Technic bricks, см. **рис. 7.18**). Они выглядят как обычные LEGO-детали, но имеют отверстия, в которые отлично проходят стяжки! Затем вы соединяете детали с внутренностью корпуса так, как вы соединяете любые детали LEGO.

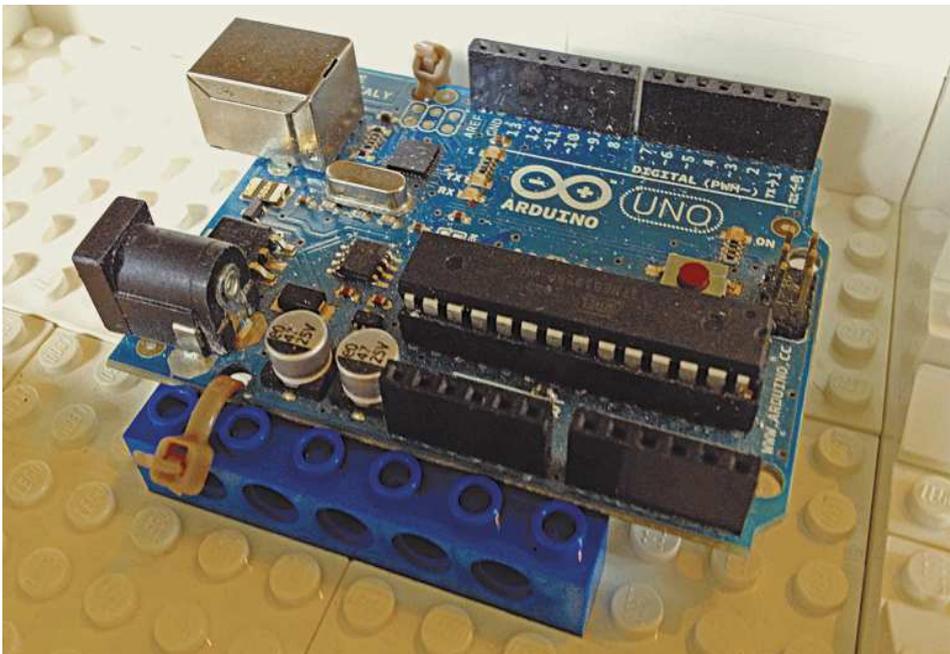


Рис. 7.18. Прикрепление Arduino к корпусу

Если у вас нет технических деталей LEGO, то можно использовать другой, клейкий, вариант, чтобы прикрепить плату. Может быть двусторонний скотч? В **главе 2** «Макетирование» я использовал суперкрутую замазку «сугру» (*sugru*), чтобы прикрепить к корпусу каверзный компонент. Сугру — не самый плохой вариант для данного проекта. Эту замазку можно также использовать для герметизации зазоров в местах входа труб в корпус для защиты от влаги, как показано на **рис. 7.19**.

После того как вы соедините две половинки корпуса LEGO, смажьте щель вокруг трубы небольшим количеством сугру. Замазка высохнет через 24 часа и не только защитит плату Arduino от попадания влаги, но также исключит вибрацию корпуса.



Рис. 7.19. Смажьте зазоры сугру, чтобы вода не попала в корпус

ДЛЯ СПРАВКИ

ПОЧЕМУ МЫ ДЕЛАЕМ ИМЕННО ТАК?

Самые сообразительные из вас, вероятно, думают: «Почему бы не просверлить отверстие в другом материале, близком по диаметру к ПВХ?». Вы правы, но мне очень хотелось задействовать в этом проекте именно LEGO.

Если вы следовали всем инструкциям, конечный результат должен выглядеть так, как на **рис. 7.20**. Поздравляю! Начинаем поливать!



Рис. 7.20. Вы закончили!

ПРОТОТИПИРОВАНИЕ С LEGO

Многие инженеры (серьезно!) используют кирпичики LEGO как часть прототипирования. Эти кирпичики экономят время, не требуют специального обучения для работы с ними и ждут не дожудтся, чтобы их использовали. Зачем тратить деньги на то, что отнимает много времени и стоит очень дорого?

Как вы убедились, выполняя этот проект, можно легко создать ящики из LEGO. Однако, используя робототехнический набор «LEGO MINDSTORMS» (LEGO MINDSTORMS Robotics), можно построить чрезвычайно сложных роботов. Хотите построить робота? Начинаем закреплять навыки дизайна с помощью «Виртуального конструктора LEGO» (LEGO Digital Designer) (**рис. 7.21**).

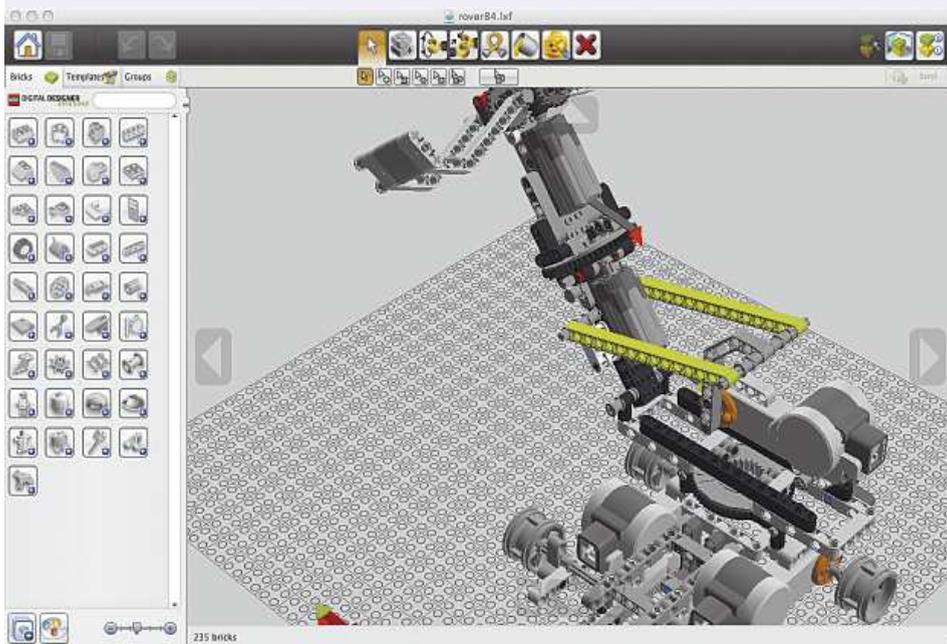


Рис. 7.21. Хотите построить виртуального робота? Виртуальный конструктор LEGO позволит вам строить онлайн!

Некоторые люди используют «Виртуальный конструктор LEGO» с твердым намерением заменить его реальным корпусом. Часто случилось, что коробка, которую я представлял в начале проекта, в конце воплощения замысла имела неправильный размер. С корпусом LEGO вы будете заранее знать размеры корпуса, прежде чем строить его настоящую версию!

Код робота для полива растений

Робот для полива растений имеет простой код. Так же, как в примере проекта, описанного ранее в этой главе, этот проект использует модифицированную версию скетча «Blink», которая включает робота через пин 13 на плате Arduino на определенное время, а затем отключает его на другой период. Это несложная программа, поскольку все, что вам нужно, — электрический импульс, который запустит электромагнитный клапан. Наиболее интересная часть (для меня) — это тайминг (расписание). Я разработал переменные значения, которые вы можете установить, чтобы контролировать частоту полива. Кроме того, вместо использования миллисекунд эти переменные включают часы и минуты, что более удобно.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

Чтобы узнать больше о том, как Arduino отслеживает минуты и секунды, см. главу 11 «Отсчет времени».

```
int valve = 13; // присваивает пину 13 значение «valve» (клапан)

int offhours = 0; // сколько часов до полива?
int offmins = 1; // сколько минут до полива?
int spray = 10; // длительность полива в секундах

void setup() {
    pinMode(valve, OUTPUT); // устанавливает пин valve в режим выхода
                           // (клапан открыт)
    Serial.begin(115200);
}

void loop() {
    int wait = (offmins * 60000) + (offhours * 3600000); // вычисляет
                                                         // милли-
                                                         // секунды

    digitalWrite(valve, HIGH);
    delay(spray * 1000); // длительность полива в миллисекундах
    Serial.println(offmins * 60000); // Я использовал это при отладке
    digitalWrite(valve, LOW);
    delay(offmins * 60000); // прекращение полива на это
                           // количество миллисекунд
}
```

В следующей главе

В главе 8 «Ящик для инструментов» описываются инструменты, которые я использовал в качестве прототипов при создании проектов для этой книги. Кроме этого, я привожу примеры оборудования, которое можно встретить в хорошо оборудованной мастерской.

Ящик для инструментов

Как вы уже могли заметить, вам потребуется много инструментов (рис. 8.1), чтобы реализовать различные проекты с Arduino. В этой главе вы познакомитесь с частью оборудования, которое, возможно, вам понадобится, на примере моего набора инструментов для начинающего мастера. После этого я подробно опишу мультиметры — бесценные измерительные приборы, которые обожают радиолюбители. Затем вы изучите инструменты для работы с деревом, пластиком и металлом. Наконец, я дам несколько советов по любительской электронике, например, как освоить мультиметр, собрать электронику из приборного хлама и расшифровать их маркировку.



Рис. 8.1. Просто удивительно, как много разных предметов может вместить обычный ящик для инструментов!

Набор инструментов для начинающего мастера

Вам необходим набор инструментов, но какой именно? Оборудование, которое находится в вашем основном ящике для инструментов (рис. 8.2), будет различаться — наборы инструментов так же уникальны, как и люди, которые ими пользуются. Только вы можете решить, что вам нужно. Тем не менее, предлагаю несколько идей по поводу того, что можно включить в набор.



Рис. 8.2. Тот самый ящик для инструментов, который я использовал во время работы над книгой, таская его в мастерскую и обратно каждый день

Основной мультиметр

Инструмент номер один для диагностики, который использует каждый радиолюбитель, — это мультиметр. Мультиметры — сложные измерительные приборы, которые применяются для измерения силы тока, напряжения, сопротивления, проверки наличия контакта и т. д. Далее в разделе «Электронные инструменты и техника» вы узнаете все о ручных приборах, а пока предлагаю включить в ваш набор недорогой мультиметр. Для быстрых измерений я использую модель DT-830B (см. рис. 8.3), которую можно купить в интернет-магазине.



Рис. 8.3. Мультиметр DT-830B — это замечательный недорогой измерительный прибор

Мультитул

Мультитул (мультиинструмент) — это ручной многофункциональный инструмент. В мультитулах обычно есть лезвия, ножницы, шило, плоскогубцы, кусачки и т. п. Я использую свой мультитул SOG Knives B61 (рис. 8.4) практически каждый день.



Рис. 8.4. Мультитул объединяет несколько инструментов в одном

Измерительная рулетка

Вы можете почувствовать себя плотником, измеряющим все, что попало, и рулетка (рис. 8.5) действительно незаменима. У меня их три: одна — в мастерской, другая — в ящике для инструментов, третья — дома.



Рис. 8.5. Нужно что-то измерить? Возьмите рулетку!

Паяльник

Мне нравится надежный паяльник, и мой «Weller WES51» (от компании *DigiKey*) выше всяких похвал, но он несколько великоват. А вот, микропаяльник (один из них показан на **рис. 8.6**) без блока питания, подставки или губки прекрасно устроится среди ваших инструментов в тех случаях, когда большой паяльник не нужен. Вам также понадобятся дополнительные принадлежности для паяния, которые подробно описаны в **главе 3** «Работа с паяльником».



Рис. 8.6. Иногда громоздкий паяльник не нужен

Цифровой штангенциркуль

Рулетки всем хороши, но иногда необходимо измерить что-нибудь с большой точностью. И тогда на помощь приходит штангенциркуль. Тот инструмент, которым я пользуюсь, имеет цифровой дисплей (см. **рис. 8.7**). Это модель Neiko 01407A.



Рис. 8.7. Цифровой штангенциркуль отображает измерения на жидкокристаллическом экране — необязательно угадывать размер!

Ножницы

Боюсь показаться старомодным, но ножницы — отличный инструмент для вашего набора. Да, вы все еще можете пользоваться для резки ножом или другим лезвием, но большинство из нас имеют большой опыт использования ножниц и могут точно резать именно этим инструментом. Хорошие ножницы стоят недорого и весят немного, так почему бы их не приобрести?

Ножи

Еще один хороший режущий инструмент — это любительские ножи (см. **рис. 8.8**). Если вам нужно отрезать что-то очень точно, возьмите одно из этих острых лезвий.



Рис. 8.8. Ножи отлично подходят для точной резки

Отвертки

Мне постоянно нужна отвертка, потому что всегда требуется что-то закрутить или открутить. Вам точно пригодится набор отверток. Особенно позаботьтесь о наличии маленьких отверток, потому что работа над проектом часто связана с мелкими крепежами.

Крепеж

Вам также понадобятся винты, болты, шайбы и т. д. Я часто роюсь в моей банке для крепежа (см. **рис. 8.9**) в поисках необходимой гайки или винтика, потому что это проще и быстрее, чем бежать в магазин. Хороший контейнер для хранения гаек и болтов можно сделать из пластиковой банки, которая с легкостью поместится в любом ящике для инструментов.



Рис. 8.9. Разнообразные крепежи в пластиковой банке

Плоскогубцы

Инструмент, который вам обязательно надо положить в набор, — это комбинация кусачек и плоскогубцев (**рис. 8.12**). Мне постоянно приходится отрезать и зачищать провода, поэтому я пользуюсь им каждый день.



Рис. 8.10. Режьте провода и очищайте их от изоляции с помощью этого инструмента

Суперклей

Суперклей склеивает почти все. Это одна из тех вещей, которая вам обычно не нужна, но когда она вдруг понадобится, вы будете рады новому тюбику в своем ящике для инструментов. Я сказал «новый», потому что после вскрытия вы вряд ли сможете использовать его еще раз.

Мини-фонарик

Другой практически бесполезный инструмент (но очень необходимый в чрезвычайной ситуации) — это небольшой фонарик-ручка.

Клеевой термопистолет

Если сомневаешься, надо склеивать! Клеевой термопистолет, подобный миниатюрной модели, представленной на **рис. 8.11**, — хороший инструмент для начинающего мастера, по крайней мере для временных исправлений. Термоклей годится на многое, начиная со склеивания коробки и заканчивая приклеиванием деталей в труднодоступных местах при их монтаже. Тем не менее известно, что все, что вы им приклеиваете, не будет долго держаться. Термоклей хорош только для временной фиксации.



Рис. 8.11. Клеевой термопистолет — секретное оружие изобретателя!

Лупа

Может быть, моя личная потребность, потому что я недавно стал пользоваться очками, но увеличительное стекло очень удобно. Приведу пример. Предположим, что вы спаяли схему на печатной плате, а она не работает. Если у вас есть возможность осмотреть следы пайки с помощью лупы, это значительно ускорит дальнейшую работу.

Письменные принадлежности

Вам необходимы разнообразные письменные принадлежности, чтобы делать отметки для резки или сверления на материале, для заметок, а прежде всего, для того, чтобы начертить ваш проект перед началом сборки.

Альбом для эскизов

Вам также понадобится альбом, в котором удобно делать эскизы своих идей. Большинство своих заметок я делал в обычных альбомах, и это вполне недорогое удовольствие.

Блоки питания и кабели

Я не знаю, сколько времени я потратил впустую в поисках нужных кабелей, которые показаны на **рис. 8.12**. Вам определенно понадобится совместимое с Arduino зарядное устройство, а также стандартный USB-кабель. Кроме того, позаботьтесь о кабеле для зарядки телефона. Запас кабелей в вашем ящике для инструментов избавит вас от массы хлопот, поверьте мне.



Рис. 8.12. Кабель USB и совместимый с Arduino блок питания — обязательны в ящике для инструментов

ИНДИВИДУАЛЬНАЯ УНИВЕРСАЛЬНАЯ АПТЕЧКА

Надеюсь, вы никогда серьезно не поранитесь во время работы, но вы должны быть уверены в том, что у вас под рукой всегда есть аптечка первой помощи для обработки обычных порезов и царапин. Просто удивительно, как часто в мастерской случаются мелкие травмы, будь то ожог от паяльника или царапины от циркулярной пилы. Ниже приведен перечень самых необходимых компонентов для оказания первой медицинской помощи.

- Классические бактерицидные пластыри разного размера.
- Бактерицидная мазь или раствор. Намажьте ее толстым слоем на вашу рану!

- Дезинфицирующие салфетки. Подходят для очистки кожи вокруг раны перед ее обработкой.
- Раствор для промывания глаз — стерильная жидкость в одноразовых пластиковых бутылочках. Если вам что-то попало в глаз, промойте его этой жидкостью.
- Дезинфицирующее средство для рук. Отлично подходит перед обработкой раны. Правда, не так хорошо, если поранена сама рука!
- Перекись водорода. Самый простой способ дезинфекции большой царапины!
- Марля. Входит в состав аптечки, как в виде бинтов, так и в виде отдельного куска и отлично подходит для перевязки больших ран.
- Медицинская лента. Используется для записи даты и времени ранения и наложения на повязку после перевязки.

Работа с деревом

Многие умельцы работают с деревом. Я не исключение, поскольку древесину легко найти, она недорогая и легкая в обработке. Если у вас есть доступ в магазин пиломатериалов (см. **рис. 8.13**), работать, конечно, будет гораздо легче. Но даже в домашних условиях можно сделать много полезных вещей из дерева.



Рис. 8.13. Имея доступ в магазин пиломатериалов, работается легче

Лазерный резак

Лазерный резак, или лазерный гравер (см. **рис. 8.14**), — это большая машина со сфокусированным лазерным лучом, который разрезает материал по схеме, заданной компьютерной программой. Если вы хотите построить коробку, все, что вы должны сделать, — это разработать модель с помощью векторного редактора, такого как *CorelDRAW* или *Adobe Illustrator*, вырезать ее с помощью лазера. Не существует более быстрого способа получить необходимое изделие. Недостатки? Ну, понятно, большинство людей не имеют доступа к лазерному лобзику. Тем не менее не расстраивайтесь. В разделе «Лазерные и ЧПУ-услуги» этой главы я посоветую места, в которые вы можете обратиться за помощью.

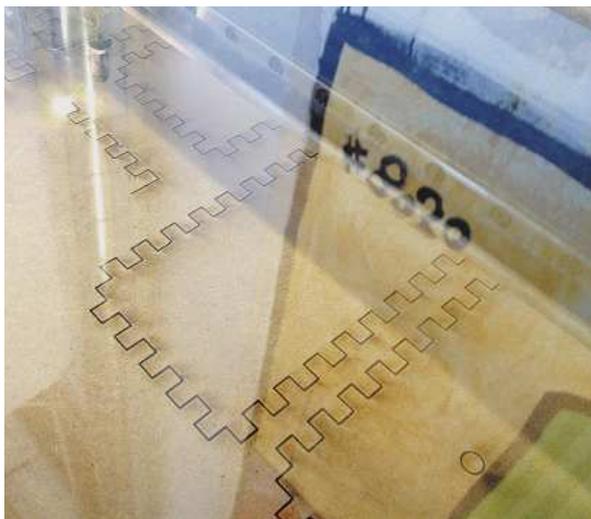


Рис. 8.14. Лазерный резак прожигает МДФ¹ толщиной 6 мм за секунды

Как пользоваться лазерным резаком

Пользоваться лазерным резаком очень просто. Я так говорю, несмотря на то что разные модели отличаются друг от друга. Каждая система имеет собственный интерфейс, поэтому создать универсальное руководство о том, как резать материал с помощью лазера, очень сложно. Однако вот вам подсказки по использованию лазерного лобзика марки «Эпилог» (Epilog), самой распространенной модели в США.

¹ МДФ (от англ. *Medium Density Fibreboard* — MDF) — древесноволокнистая плита средней плотности; плитный материал, изготавливаемый методом сухого прессования мелкодисперсной древесной стружки при высоком давлении и температуре. Отличается от древесно-стружечной плиты, ДСП. — *Прим. перев.*

- Подготовьте векторный файл в программе *Adobe Illustrator* или *CorelDRAW*. Векторы — это линии или дорожки в виде кривых. Все линии для резки должны быть очень тонкими, а формы для гравировки или запечатления должны быть растровыми изображениями, такими как фото или логотипы. Они будут выжжены на дереве, но не прорезаны насквозь.
- Выберите нужный материал и поместите его на лазерный столик. Я было обрадовался МДФ толщиной 6 мм, представляющей собой древесно-акриловый композит доски и акрила. Однако оказалось, что клей в фанере просачивается под воздействием лазера, препятствуя резке и создавая огромное количества нагара.
- На компьютере, подключенном к лазеру, запустите программу векторного дизайна (я использовал только *CorelDRAW*) и откройте файл с моделью, которую вы собираетесь вырезать. Перейдите в раздел «Настройки» и убедитесь, что выбрали три основных параметра, настроенные по вашему желанию: скорость, или как быстро будет двигаться лазерный луч; мощность, — это сила, с которой лазерный луч будет прожигать материал; частота, или как быстро пульсирует лазерный импульс.
- Если вы уже положили свой материал на лазерный столик и готовы к резке, нажмите кнопку «Печать», как если бы вы работали с обычным струйным принтером. Далее откроется диалоговое окно печати, в котором вы можете выбрать и другие варианты. Нажмите кнопку «Печать», и ваши векторы будут отправлены на лазер!

На самом деле пользоваться лазерным резаком очень легко. Проблемы, с которыми вы столкнетесь, будут связаны с неправильными параметрами установки и обугливанием или плавлением того, что вы пытаетесь разрезать. Все они разрешимы после определенного количества проб и ошибок. Часть опыта резки с лазером включает экспериментальный подбор правильных параметров установки лазера. Не расстраивайтесь, если ваш материал обуглился или лазер остановился на полдороге во время резки. Просто измените настройки и попробуйте еще раз.

Роторный инструмент

В противоположность лазерному резаку роторный инструмент (наиболее часто упоминаемая торговая марка — *Dremel*, лидер в этой категории) определенно не столь технологичен. Это небольшой двигатель с различными насадками, которые можно установить на его ступице. Роторный инструмент «Dremel 8220», изображенный на **рис. 8.15**, — это беспроводная модель, которая поставляется с зарядным устройством. Пусть эта модель и не такая мощная, как проводные, зато гораздо удобнее в работе! Она сопровождается пилами, дрелями, полировочными и шлифовальными насадками.



Рис. 8.15. Роторный инструмент — это отличная вещь для разрезания, вырезания и гравирования древесины

Воздушный компрессор с насадками

Многие любители пользуются пневматическими инструментами, такими как шуруповерт («пистолет» для ввинчивания шурупов), дрели, фен высокого давления и распылители для краски. Преимущество сжатого воздуха в том, что инструменты, работающие на его основе, мало весят, поскольку не имеют встроенного источника питания.

Кроме этого, потребуется только один компрессор, к которому можно подключать разные инструменты. Можно приобрести очень дешевый компрессор, особенно из ряда небрендовых маломощных моделей (**рис. 8.16**).



Рис. 8.16. За счет сжатого воздуха от компрессора работает много инструментов

Дрель

Я пользуюсь ручной дрелью и настольным сверлильным станком примерно три раза в неделю. Когда дрель укомплектована набором сверл, вы действительно можете сделать много полезной работы. Дома я пользуюсь аккумуляторной дрелью (марка «DeWalt» на 18 В), чтобы вкручивать шурупы и сверлить отверстия. Для точной работы я пользуюсь настольным сверлильным станком (рис. 8.17).

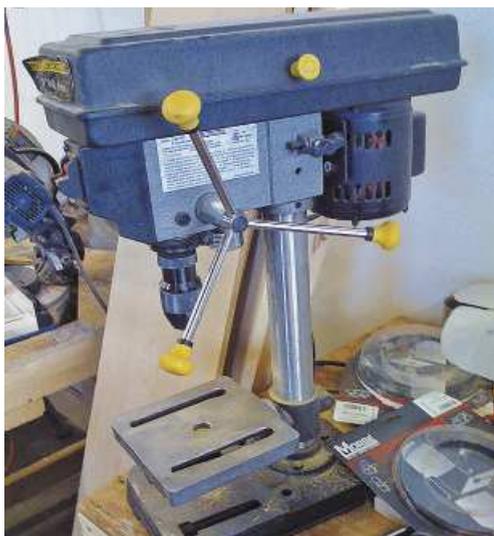


Рис. 8.17. Настольный сверлильный станок — это обязательная принадлежность для точного сверления отверстий

Фрезерный станок с числовым программным управлением (ЧПУ)

Фрезерный станок с ЧПУ — это управляемая компьютером фреза, которая следует векторному пути (так же, как и лазерный резак) и позволяет осуществлять резку и строгание с большой точностью (рис. 8.18). Одним из преимуществ станка с ЧПУ перед лазером является то, что некоторые модели могут работать в трехмерном измерении, вырезая, например, миску из древесного блока.



Рис. 8.18. Фрезерный станок с ЧПУ обрабатывает древесину с цифровой точностью

ДЛЯ СПРАВКИ

ЕЩЕ О СТАНКАХ С ЧПУ

В **главе 11** «Измерение времени» вы узнаете все о ЧПУ, а затем сделаете проект корпуса из настоящих комплектующих с помощью станка с ЧПУ. Заинтересовались этой технологией? Тогда читайте дальше.

Лазерные и ЧПУ-услуги

Если у вас нет лазерного лобзика или станка с ЧПУ, скорее всего вы почувствовали, что зашли в тупик, читая эту книгу. Как можно провести лазерную резку без лазера? Вот один из вариантов.

Нередко образовательные учреждения, такие как местные колледжи, и даже районные библиотеки открывают магазины самообслуживания с лазерными лобзиками и станкам с ЧПУ, доступными для посетителей. Поищите их!

Циркулярная пила

Циркулярная пила (см. **рис. 8.19**), приспособление, известное с незапамятных времен, незаменима для быстрой резки древесины большого размера. Это также самый опасный инструмент в любой деревообрабатывающей мастерской, поэтому необходимо знать, как им правильно пользоваться. Существует устройство «Стоп пила», которое опускает тормоз на лезвие пилы, если оно задевает кожу мастера. Обязательно спросите об этой функции при покупке пилы.



Рис. 8.19. Рабочая часть циркулярной пилы

Токарный станок

Я подробно рассмотрю токарный станок в **главе 9** «Ультразвуковая эхолокация», а сейчас дам только краткий обзор. Станок приводит в движение шпиндель (см. **рис. 8.20**), который вращает заготовку из дерева или металла, а вы обрабатываете ее с помощью токарных стамесок. С помощью этого станка вы можете сделать, например, декоративные ножки для стола. Токарный станок — это еще один опасный предмет в мастерской для деревообработки, потому что вращающийся шпиндель может мгновенно намотать на заготовку волосы или рукава одежды, что приведет к травме или даже к летальному исходу.



Рис. 8.20. Токарный станок вращает деревянную заготовку, помогая вам обрабатывать ее стамесками

Шлифовальный станок

Шлифование грубо обработанной древесины предполагает использование дискового или ленточного шлифовального станка либо, как на **рис. 8.21**, комбинированной модели. Как и большинство инструментов деревообрабатывающей мастерской, доступные варианты охватывают широкий диапазон цен и функций, предоставляя вам возможность выбрать единственный станок для ваших потребностей.



Рис. 8.21. Сочетание дискового и ленточного шлифовальных станков позволяет шлифовать древесину двумя различными способами

Работа с пластиком

Пластик (пластмасса) может быть очень гибким средством для работы над проектами. Его можно расплавить и переформовать, распилить и просверлить. Вы можете распечатать объекты в пластике (**рис. 8.22**) с помощью программы для 3D-прототипирования, установленной на вашем компьютере. Вы можете согнуть его тепловой пушкой, как было описано в **главе 4** «Настройка беспроводных соединений».

Пластик можно разрезать на совершенные по форме кусочки с помощью лазерного резака. Лазеру нравится акрил! Если использовать правильные настройки, лазер отлично режет пластик, делая ровные полированные края.

Однако вы уже миллион раз прочитали про лазерную резку в этой книге? Что еще можно сделать с пластиком? В следующих разделах рассматриваются способы, с помощью которых вы можете поиграть с этим материалом.

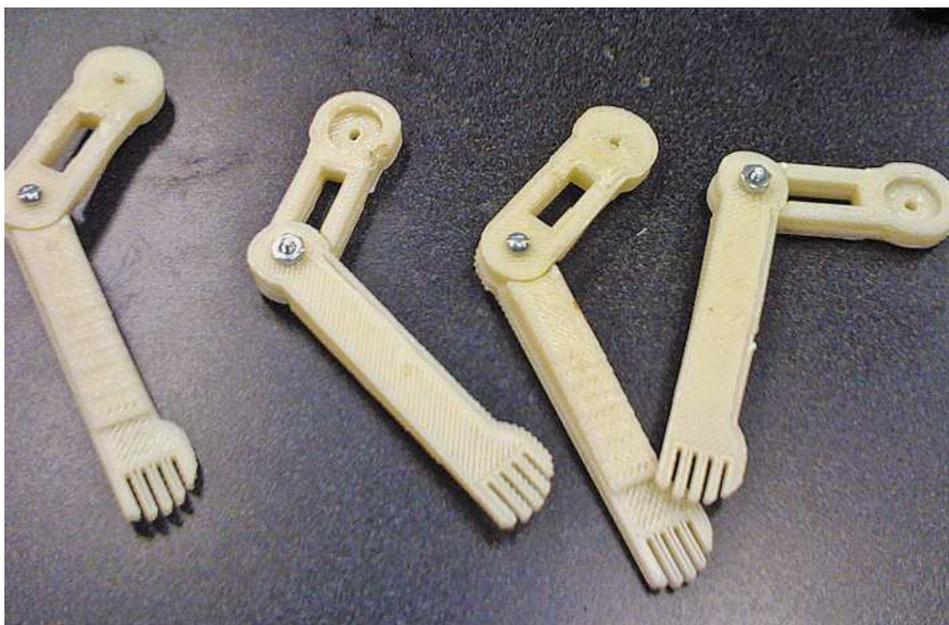


Рис. 8.22. Эти руки для роботов были напечатаны на 3D-принтере из пластика

3D-принтеры

Одно из самых уникальных и захватывающих событий в сфере работы с пластиком — это 3D-принтеры (**рис. 8.23**), которые экструзируют материал, т. е. продавливают расплавленный пластик через специальные отверстия (экструдеры) по определенной схеме. Это похоже на то, как станки с ЧПУ и лазерные резаки следуют заданной инструкции. Однако, если эти инструменты режут материал, то 3D-принтеры наращивают (добавляют) его. Принтер создает 3D-объекты путем экструзии пластмассы слой за слоем, пока объект не будет сформирован.

Раньше считалось, что 3D-принтеры были визитной карточкой успешных дизайнерских студий и магазинов промышленного дизайна. Однако за последние годы разработаны любительские 3D-принтеры, которые широко распространились по всему миру. Их стоимость гораздо ниже, чем стоимость профессиональных моделей.

Одна из самых успешных компаний, продающих 3D-принтеры, — это *MakerBot Industries*, которая создала 3D-принтер для капкейков с ЧПУ, который вы видите на **рис. 8.23**. *MakerBot* почти в одиночку превратила

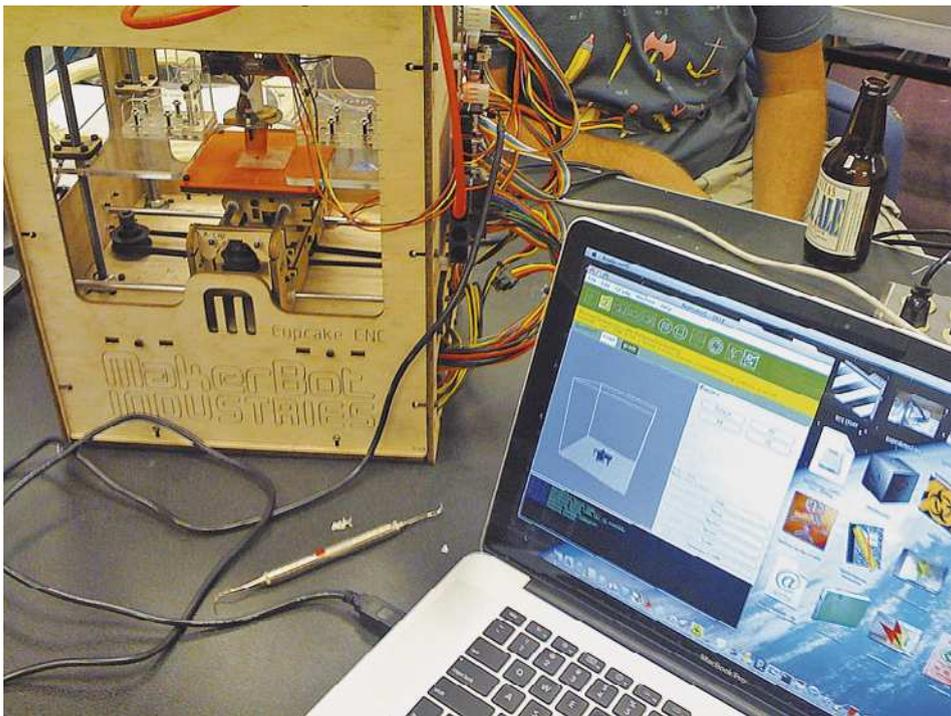


Рис. 8.23. 3D-принтер с ЧПУ для капкейков, изготовленный *MakerBot Industries*

3D-печать в явление, о котором многие слышали. Тысячи мастеров-любителей, учителей и промышленных дизайнеров имеют на своих столах настольные 3D-принтеры, каждое поколение принтеров становится немного лучше. Совсем скоро мы увидим, что эта технология станет еще более популярной.

LEGO

Зачем связываться с проблемами прототипирования деталей для проекта, если у вас уже есть куча деталей из комплекта LEGO (рис. 8.24)? У кирпичиков и стержней LEGO есть масса преимуществ.

- **Универсальность.** У кого из вас до сих пор хранится ведро деталей из LEGO? Наверняка у многих. Это означает, что в случае необходимости, вы всегда сможете построить коробку для проекта или опорную рамку из этих деталей.

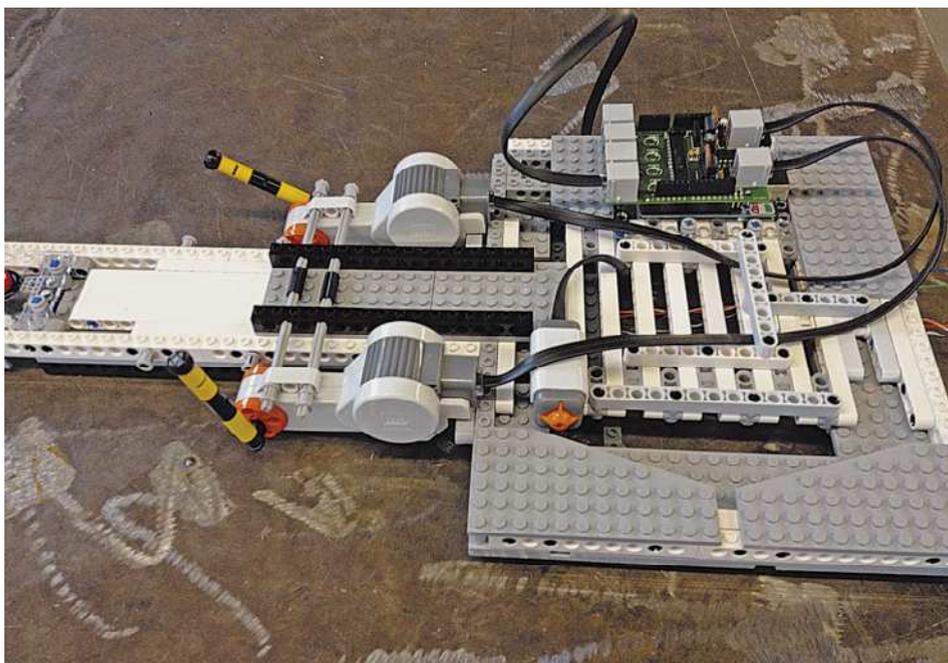


Рис. 8.24. LEGO-«гитара» с Arduino и платой расширения «Bricktronics»

- **Прочность.** Детали LEGO отлиты из АБС-пластика¹, одного из лучших в мире пластиков потребительского класса.
- **Робототехника.** Компания LEGO серьезно относится к своим комплектам по робототехнике. Многие инженерные и робототехнические программы начинают обучение с освоения LEGO-робототехники. Трудно не соблазниться изобилием двигателей, колес и датчиков!
- **Наличие дополнений.** Многие сторонние компании разработали продукты, которые расширяют функциональность LEGO-робототехники. Например, *Wayne & Layne* разработали шильд (панель управления) Arduino² с LEGO-совместимым тумблером, который позволяет управлять LEGO-роботом с помощью Arduino.

Сугру

Я уже упоминал сугру (рис. 8.25) несколько раз в этой книге. Напомню, что это пластиковый моделирующий клей, который ко всему прилипает и застывает в течение 24 ч. Вы можете использовать его, чтобы склеить два предмета, чтобы укрепить или починить сломанную вещь, добавить резиновую прослойку в рукоятку для инструмента. Есть даже производители, которые делают из сугру шаблоны для резиновых частей своих роботов.



Рис. 8.25. Сугру выпускается разных цветов

¹ АБС-пластик (акрилонитрилбутадиенстирол — АБС) — ударопрочный термопластик, завоевавший высокую популярность в промышленности и в аддитивном производстве. — *Прим. перев.*

² Шильд (от англ. *shield*) Arduino — это плата расширения контроллера Arduino. — *Прим. перев.*

Вакуумно-формовочное устройство (формователь)

Один умный способ сформовать пластик — сначала нагреть его, а потом с помощью вакуума придать желаемую форму. Пластик прилипает к стенкам, охлаждается и принимает форму другого объекта. Полученные пластиковые оболочки можно покрасить, они используются в качестве литейных форм. Вакуумный формователь, который вы видите на **рис. 8.26**, — это устройство, предназначенное как для нагревания пластика, так и для его формования.



Рис. 8.26. Вакуумный формователь сначала нагревает пластик, а затем использует вакуум, чтобы материал принял форму копируемого объекта

Экструдер

Экструдер (рис. 8.27) нагревает пластик и придает ему форму. Чтобы экструдер работал, вы уже должны иметь готовый шаблон; его изготовление может быть отдельной задачей. Вы нагреваете пластиковые гранулы, пока они не расплавятся, затем нагнетаете пластик в шаблон, где он быстро охлаждается. Вы слышали о машине «Mold-A-Rama»? Это пластик-формовочная машина, которая работает по принципу монетного автомата — так же, как экструдер.



Рис. 8.27. Экструдер расплавляет пластиковые гранулы и впрыскивает полученную массу в форму

Работа с металлом

Несмотря на то что обработка металла выглядит более пугающей, чем обработка дерева или пластика, работа с металлом (**рис. 8.28**) может быть чрезвычайно полезной, а также предполагает более долговечные результаты, чем другие материалы. В этом разделе вы узнаете об инструментах для металлообработки.



Рис. 8.28. Сварочные аппараты готовы к работе

Плазменные резаки

Лазерные резаки идеально подходят для прожигания дерева и пластика, но как же резать металл? Существует не так много лазеров потребительского класса, достаточно мощных, чтобы прожечь металл. Одно из решений представляет собой управляемый компьютером резак, который использует плазму или горячий газ, чтобы прожечь металл (**рис. 8.29**). Если вы хотите точно разрезать металл, этот инструмент в самый раз.



Рис. 8.29. Плазменный резак использует струю раскаленной добела плазмы, чтобы прорезать металл

ДЛЯ СПРАВКИ **ОПЦИИ ПЛАЗМЕННОЙ РЕЗКИ**

Следует отметить, что некоторые плазменные резаки управляются вручную, в то время как другие управляются двигателями. Выбирайте правильный инструмент для работы над каждым проектом.

Ленточная пила

Подобно ленточным пилам в столярной мастерской, в мастерской металлообработки вы можете увидеть ленточнопильный станок (рис. 8.30). Полотно пилы расположено горизонтально и поднимается вручную, а затем опускается вниз на разрезаемый кусок. Одновременно на режущую кромку наносится смазка, чтобы защитить полотно пилы от перегрева. Ленточная пила используется для резки металлических прутков и тонких кусков металла и не годится для толстых.



Рис. 8.30. Металлорежущий ленточнопильный станок — самый подходящий для резки тонких кусков металла

Шлифовальный станок

Шлифовальный станок отлично справляется с небольшими зазубринами на поверхности металла (рис. 8.31). Например, можно убрать следы коррозии или краски. Шлифовальный станок используют для отделки или даже резки металла.



Рис. 8.31. Отшлифуйте поверхность куска металла, чтобы довести его до совершенства

Сварочный аппарат

Классический инструмент слесаря по металлу, сварочный аппарат, отлично подходит для соединения двух заготовок из металла. Существуют три основных типа этих аппаратов.

1. **Ручной сварочный аппарат.** Также известен как ручной аппарат для дуговой сварки с защитной средой (Shielded Metal Arc Welding – SMAW); это основной современный метод сварки. Сварочный аппарат создает электрическую дугу между электродом и поверхностью для сварки. Расходуемый электрод горит и выделяет пары инертного газа, который защищает целостность сварного шва.
2. **Аппарат для МИГ-сварки.** Аббревиатура МИГ означает «металл–инертный газ». Проволока во время работы подается через сварочный пистолет в зону сварки. Ее плавление происходит от тепла дуги. Помимо того что проволока играет роль токопроводящего электрода, она является присадочным материалом. Сама зона сварки (сварочная ванна, зона дуги и электрод) защищена потоком газа (отсюда и название). Пистолет сварщика автоматически выдвигает катушку проволоки, чтобы сформировать шов. См. **рис. 8.32**.
3. **Аппарат для TIG-сварки.** TIG – это ручная сварка неплавящимися вольфрамовыми электродами в среде защитного газа аргона. Как и при МИГ-сварке, аппарат для TIG-сварки выдвигает металлическую проволоку, чтобы заполнить пробелы в сварном шве.



Рис. 8.32. Аппарат для МИГ-сварки

Алюминиевые конструкторы

Иногда вам не нужно резать, сверлить или сваривать металл, чтобы получилась деталь. Алюминиевые строительные наборы позволяют создавать конструкции, подобные LEGO, но гораздо более прочные, правда, и стоят они дороже. В следующих разделах мы рассмотрим некоторые популярные наборы.

Профили конструкционные 80/20

Алюминиевые профили, которые используют специалисты, имеют размеры 80/20, их также называют «Промышленный строительный набор 80/20». Профили 80/20 имеют множество размеров и конфигураций, в зависимости от того, в каком месте вы хотите установить другое оборудование, например маршрутизатор ЧПУ, показанный на **рис. 8.33**.

Важная особенность архитектуры профиля 80/20 — это Т-образный паз, который тянется по всей длине балки. Гайки и болты прикручиваются в любом месте вдоль паза, что позволяет создать конструкции из комбинаций профилей.

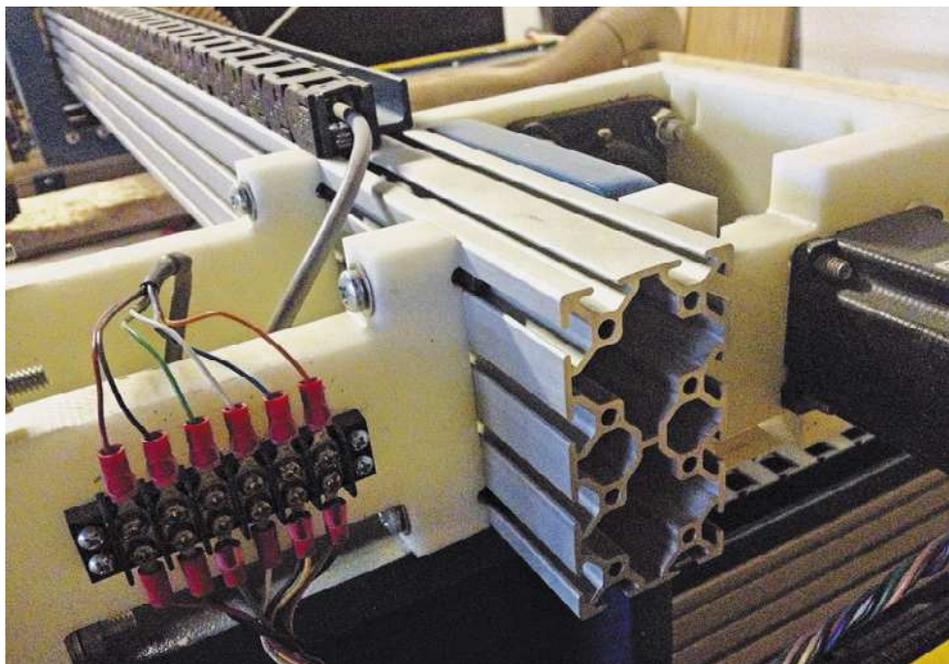


Рис. 8.33. Профиль 80/20 с закрепленным роутером ЧПУ, имеющим пластиковые детали шасси, которые привинчиваются в Т-образные пазы

MicroRAX

MicroRAX (**рис. 8.34**) — это меньший по размерам, но очень полезный Т-слотный¹ конструкционный алюминий. Дизайн был изобретен братьями-близнецами, работающими на складе в Сиэтле. Балки MicroRAX значительно тоньше, чем 80/20, имеют ширину 1 см (балки 80/20 имеют ширину 25 мм и 40 мм). Они также намного дешевле!

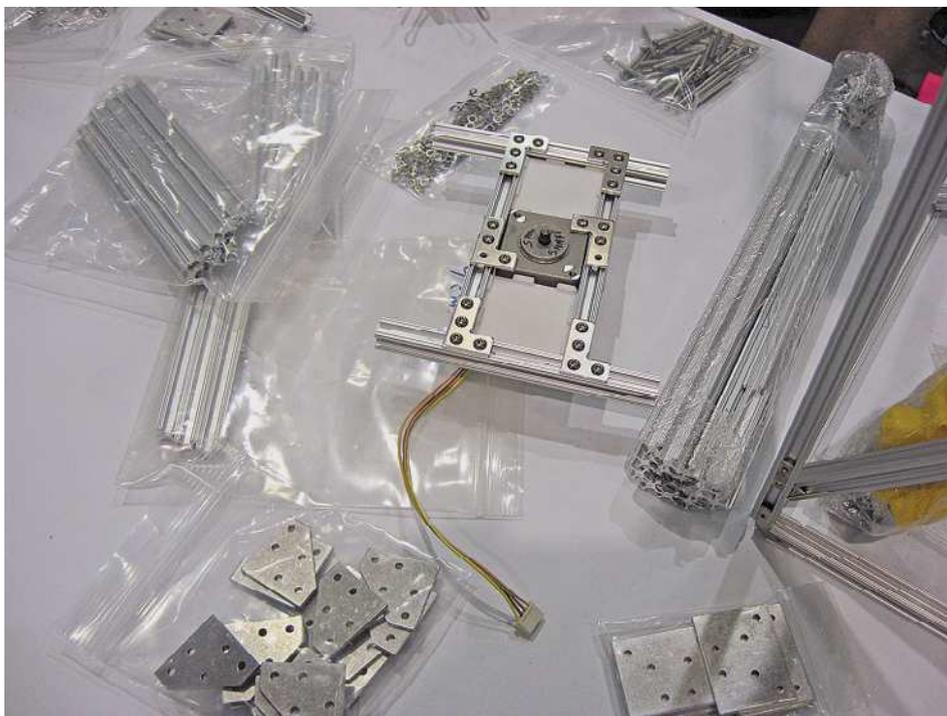


Рис. 8.34. Рамка из профиля MicroRAX сделана для поддержки шагового двигателя

¹ Т-слот — это Т-образный желобок, паз. — Прим. перев.

OpenBeam

Что, если вы изобрели Т-слотную систему балок, в основу которого заложена идея открытого аппаратного обеспечения? Это и есть идея *OpenBeam* («Открытая балка») (рис. 8.35), которая очень проста в использовании. Слоты *OpenBeam* на 100% совместимы с промышленными гайками и болтами, а ширина слота вмещает панели из балтийской березы толщиной 3 мм. Будучи компанией с открытым исходным кодом, *OpenBeam* щедро делится техническими деталями своего продукта, так что и вы можете внести свой вклад.

Источник: OpenBeam



Рис. 8.35. Разнообразие профилей и других аксессуаров от «OpenBeam»

Makeblock

Компания *Makeblock* выбрала иной подход, отличающийся от Т-слотной системы, создав сложный массив из реек, передач, колес и соединителей. *Makeblock* (рис. 8.36) был задуман в качестве полноценного комплекта для создания робота и имеет много полезных усовершенствований по сравнению с Т-слотами, например, резьбовые слоты для того, чтобы винты можно было вставить без гайки.

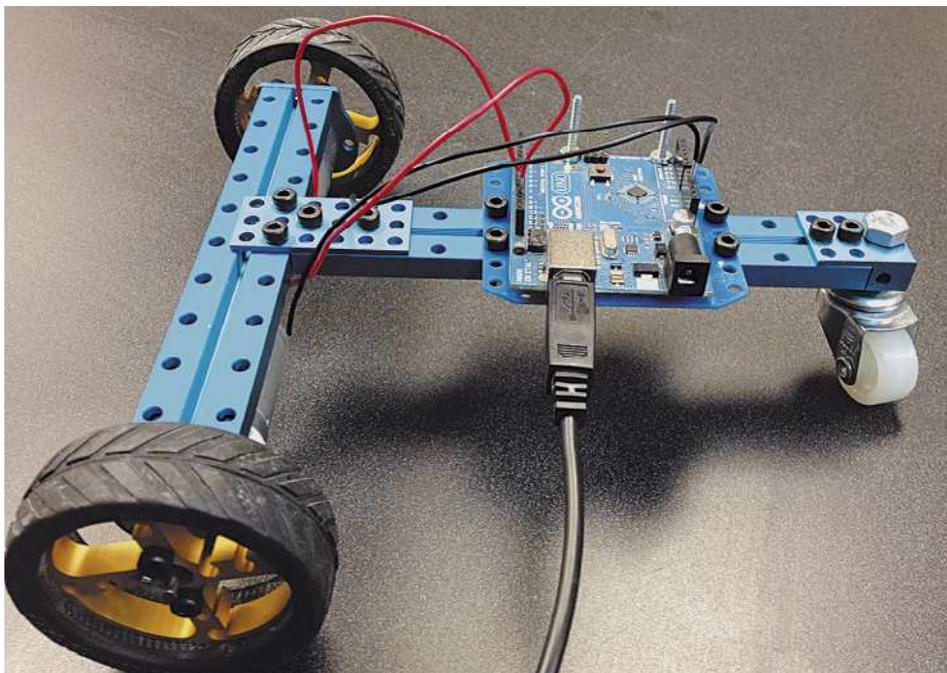


Рис. 8.36. Шасси робота *Makeblock* сформировано

VEX

VEX — это учебный алюминиевый монтажный строительный набор (показан на **рис. 8.37**), который состоит из перфорированных металлических реек с винтовыми отверстиями для крепления. Набор имеет собственную систему пользовательского микроконтроллера, в том числе дистанционное радиоуправление.

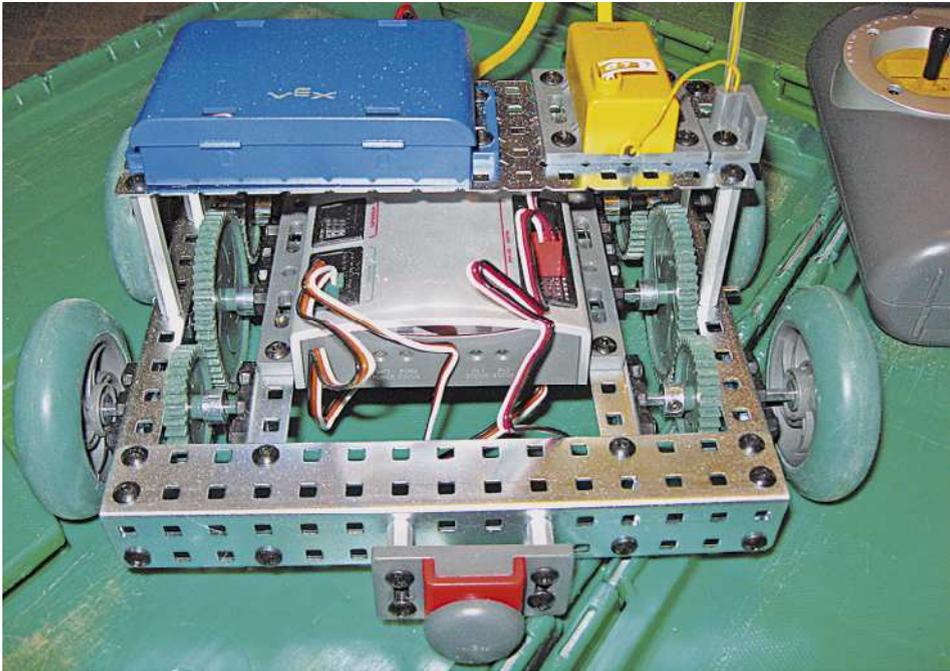


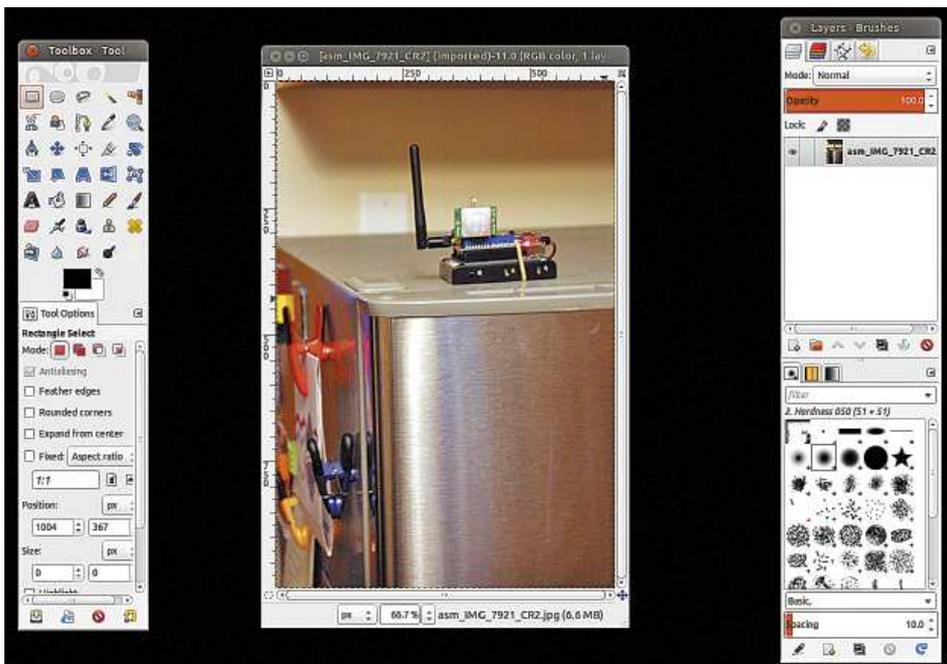
Рис. 8.37. Робот «VEX» с аккумулятором и беспроводным приемником, установленным наверху

Программное обеспечение

Не все инструменты материальны! Иногда программное обеспечение может быть большим подспорьем в разработке электронных схем и создания файлов для лазерной резки. Конечно, описанные далее ресурсы — лишь малая часть из всего софта, разработанного в настоящее время, но они одни из лучших.

GIMP

GIMP (GNU Image Manipulation Program) (**рис. 8.38**) — это мощный многоцелевой графический редактор, бесплатная версия классического графического редактора *Adobe Photoshop* с открытым исходным кодом. GIMP работает в самых популярных операционных системах: Windows, Mac, Linux. Меню и опции программы напоминают *Adobe Photoshop*. Вы можете узнать больше об этой программе на gimp.org.

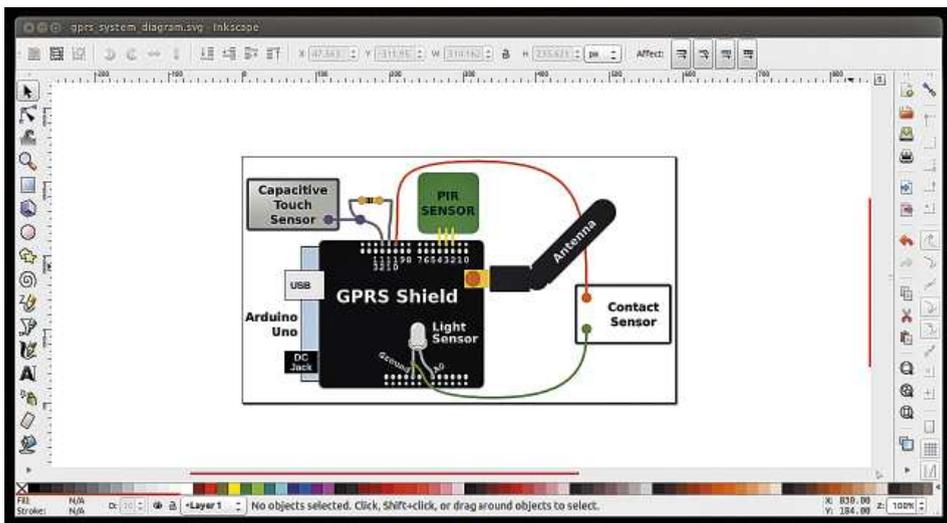


Источник: Adam Wolf

Рис. 8.38. GIMP — это бесплатная программа для работы с изображениями с открытым исходным кодом

Inkscape

Если GIMP — это открытая и бесплатная версия *Adobe Photoshop*, то *Inkscape* — это свободный и открытый эквивалент программы *Adobe Illustrator* (рис. 8.39). Программа позволяет проектировать и управлять векторной графикой, формировать алгоритм для станков с ЧПУ. *Inkscape* использует открытый стандарт SVG (Scalable Vector Graphics) от W3C в качестве формата по умолчанию, который можно открыть большинством программ для работы с векторной графикой, в том числе *Adobe Illustrator*.



Источник: Matthew Beckler

Рис. 8.39. *Inkscape* позволяет создавать и редактировать векторную графику

Fritzing

Вы уже знакомы с программой *Fritzing* (рис. 8.40) или по крайней мере с ее возможностями. Почти каждая схема в этой книге была создана с помощью этого софта. *Fritzing* — это программное обеспечение с открытым кодом, созданное для того, чтобы помочь дизайнерам и художникам перейти от физического проектирования к законченному продукту. В сущности, это инструмент начинающего компьютерного электромонтажника. Программа состоит из библиотеки готовых элементов, которые можно перетаскивать в создаваемый макет электросхемы. Вы даже можете сохранить ваш дизайнерский файл в *Gerber* — формате для создания печатных плат. Вместе с тем *Fritzing* пока выпущена в бета-версии, а это значит, что она находится в стадии разработки и не готова к официальному представлению. Тем не менее много людей в мире постоянно пользуются этой замечательной программой.

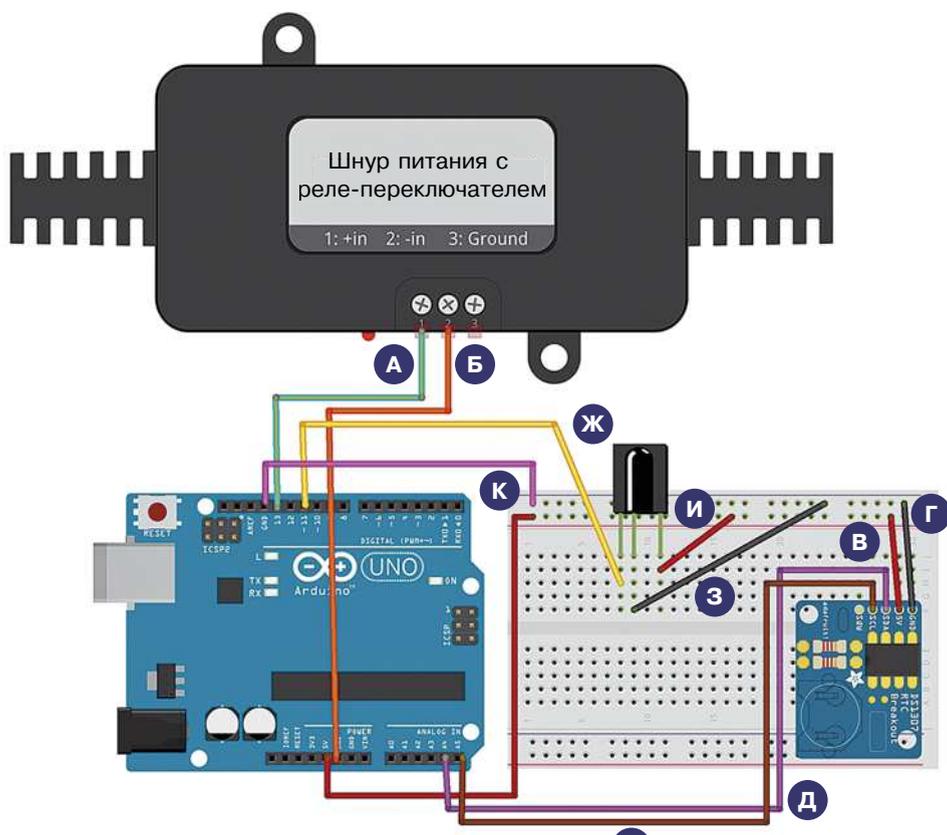
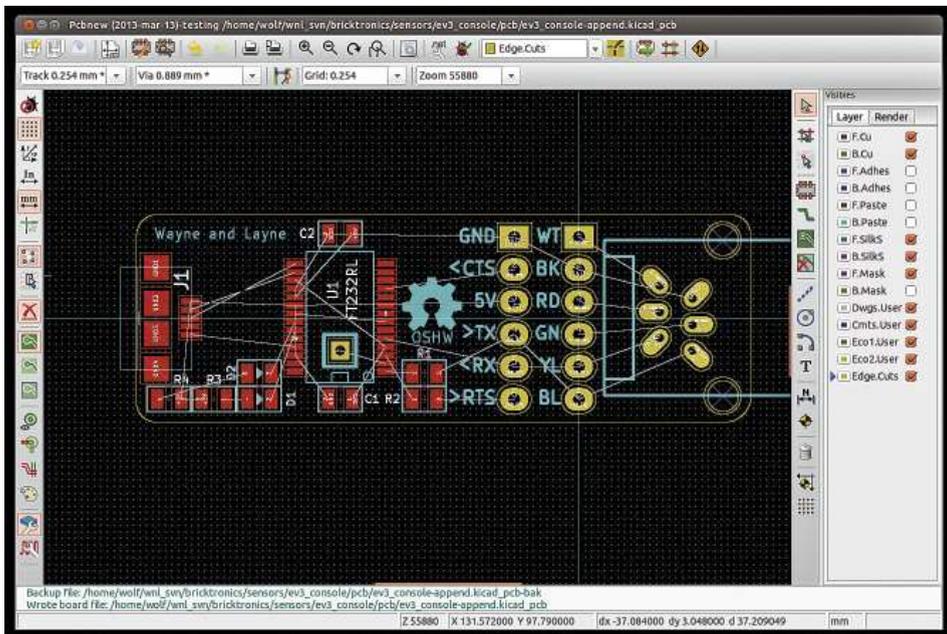


Рис. 8.40. *Fritzing* создает сложные электрические схемы, в которых несложно разобраться

Программное обеспечение KiCad PCB Layout

KiCad (см. рис. 8.41) — это наиболее сложное профессиональное программное обеспечение для разработки электронных схем. Так же, как *GIMP* и *Inkscape*, это бесплатная программа с открытым исходным кодом. Основное назначение *KiCad* — проектирование печатных плат для массового производства. Вот как это работает. Предположим, что вы создали цепь и хотите сделать из нее печатную плату (PCB). Вы запускаете *KiCad*, который позволит вам спроектировать печатную плату, провести трассировку всех соединений, сгенерировать файл в формате *Gerber* и вывести спецификацию необходимых компонентов для печати. Получится так же функционально, как у профессионального программного обеспечения, и без затрат.



Источник: Adam Wolf

Рис. 8.41. *KiCad* поможет вам сконфигурировать печатную плату

MakerCase

В этой книге представлено много корпусов, созданных лазерной резкой. Я создал их в *Adobe Illustrator* — программе, которой я привык пользоваться. И что же вам делать, если вы хотите создать коробку с помощью лазерной резки, а у вас нет доступа к программе векторной графики? Одно из решений — обратиться за помощью к MakerCase.com, сайту, который разработает векторы вашей коробки (**рис. 8.42**). Все, что вам нужно сделать, — это ввести размеры коробки и нажать на различные опции, а затем загрузить векторы с сайта. Вы готовы вырезать свой ящик!

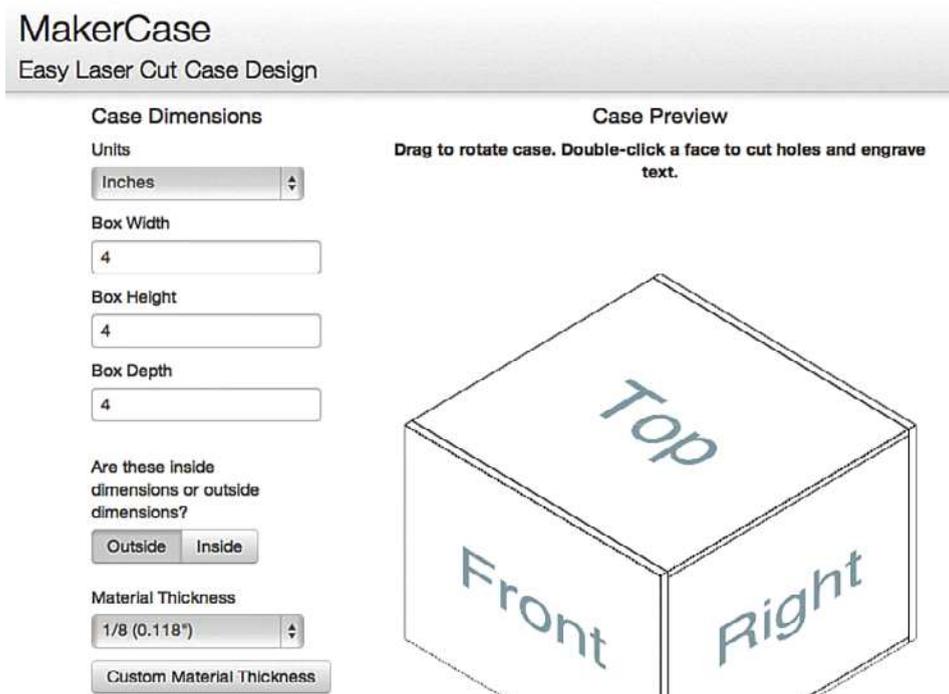


Рис. 8.42. *MakerCase* задает векторы для лазерной резки проектируемого корпуса

Электронная техника и инструменты

Это книга об электронике, поэтому будет справедливо включить электронные инструменты в главу об инструментах. Давайте начнем с самого полезного помощника в электротехнике — с мультиметра (рис. 8.43). Затем я опишу несколько других микроконтроллеров без Arduino, а также платы с подключенным Arduino. Я расскажу, как избавиться от нежелательного изобилия потребительских компонентов.

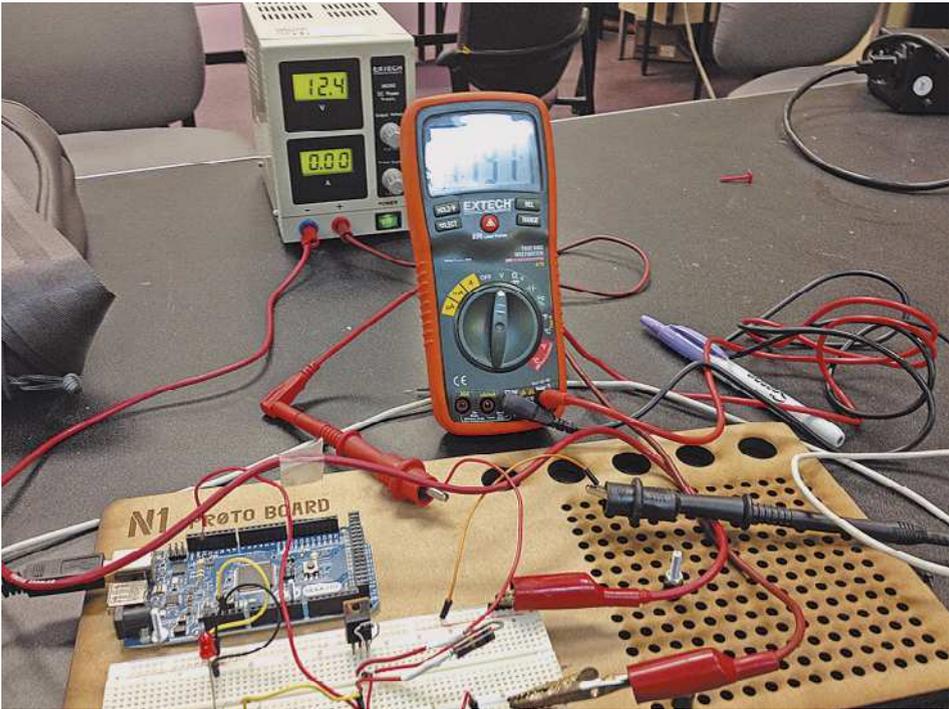


Рис. 8.43. Мультиметр — незаменимый инструмент в руках электротехника

Мультиметры (мультитестеры)

Я уже упоминал мультиметры бесчисленное множество раз в этой книге, но как именно пользоваться этим прибором? На **рис. 8.44** показан набор функций типичного недорогого мультиметра. Почему недорогого? Потому что для объяснения работы более сложных приборов потребовалась бы целая глава, и вы все равно вряд ли что-либо поняли бы из нее. Давайте сосредоточимся на одной из самых простых моделей.



Рис. 8.44. Даже самый простой мультиметр может быть на удивление сложным

- А. **Жидкокристаллический экран**, отражающий четыре параметра.
- Б. **Переключатель выбора функций**. Вы просто поворачиваете переключатель в сторону с указанием того параметра, который измеряете.
- В. **Напряжение постоянного тока (DCV)**. Поверните переключатель на любое значение, близкое по значению к тому, которое вы измеряете. Например, если вы тестируете батарейку 12 В, поставьте переключатель на 20. Приложите один щуп тестера к положи-

тельном контакту того, что вы измеряете, а другой — к отрицательному контакту.

- Г. **Напряжение переменного тока (ACV)**. Измеряется так же, как и напряжение постоянного тока. Я использую этот прибор для проверки комнатной электропроводки, поэтому установил переключатель на 200 В для измерения напряжения 110 В (в нашем случае 750 В и 220 В соответственно).
- Д. **Сила тока постоянного напряжения (в амперах) (DCA)**. Измерить силу тока с помощью такого тестера, как этот, непросто. Вы должны быть очень осторожны, чтобы не испортить прибор. Если вы посмотрите на шкалу тестера, то увидите диапазон от 200 миллиампер до 200 микроампер. Предохранитель в тестере защищает его в этом диапазоне, так что если сила тока элемента, который вы измеряете, превышает 250 мкА, он мгновенно перегорает.
- Е. **10 ампер (10 A)**. Вы можете пользоваться этой установкой, чтобы проводить измерения до 10 А. Однако, в отличие от измерения силы тока, о котором я рассказал, этот параметр незащищен, так что вы должны быть осторожны. Вы можете тестировать не более 10 с, а затем сделать перерыв на 15 мин между измерениями. При этом не забывайте, что DT-830B является дешевым прибором, а значит, не очень надежен.
- Ж. **Контактные гнезда**. Чтобы сделать большую часть своей работы, тестеру необходимы провода. Есть три гнезда; подключите черный провод к гнезду «СОМ», а красный провод к верхнему гнезду, если вы измеряете 10 А, или к среднему гнезду, если измеряете что-то еще.
- З. **Проверка транзисторов**. Это голубая панель, называемая hFE-розеткой, для проверки транзисторов. Чтобы проверить транзистор, установите ручку на «hFE» и подключите провод транзистора к голубой панели, на которой указано, в какое гнездо помещать каждую ножку транзистора.
- И. **Проверка сопротивления**. Поворотом переключателя выберите установку, соответствующую значению, которое вы тестируете.
- К. **Выключение прибора (OFF)**. Поверните ручку в это положение, чтобы выключить прибор.
- Л. **Проверка целостности цепи («прозвонка»)**. Приложите щупы прибора к двум концам измеряемой цепи. Если они связаны, зуммер даст сигнал.

Собираем электронные запчасти

У вас есть старая электронная игрушка? Тогда есть шанс, что ее запчасти еще можно пустить в дело. То же самое касается старых факсов, сканнеров, CD-плееров и другого электронного хлама, который пылится в кладовке без дела.

Недавно я разломал старика «Scooba» (см. **рис. 8.45**), автоматического робота-уборщика, который подметает и моет кухонный пол, пока вы отдыхаете. В дополнение к ожидаемым находкам двигателю и насосу я обнаружил такие полезные детали, как оптический датчик движения и инфракрасный маяк, направляющий робота в противоположную сторону после столкновения с препятствием.

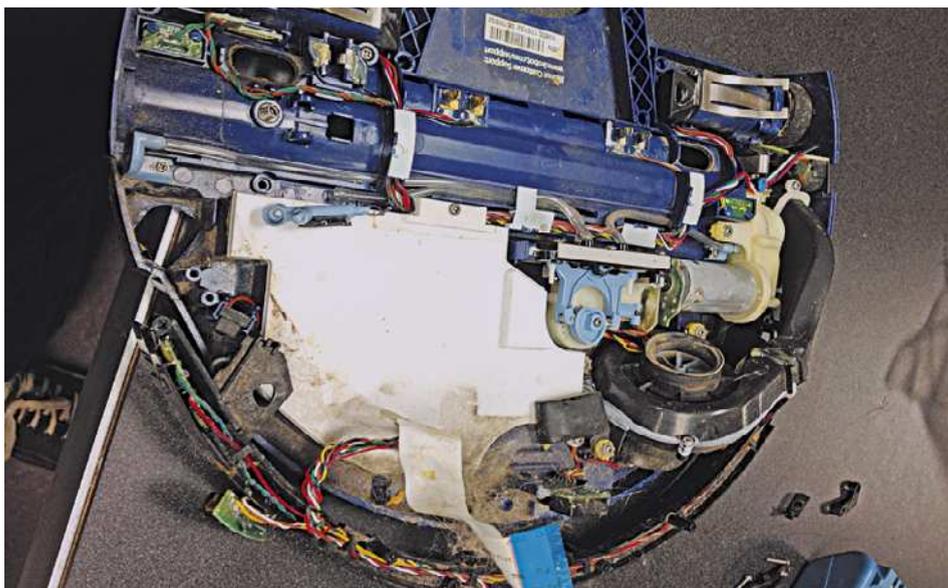


Рис. 8.45. Сломанный робот-уборщик «Scooba» был разобран на запчасти

Разбирать электронный хлам на запчасти — увлекательное дело. Вы можете не только набрать детали из ненужной техники, но также изменить работу действующего устройства, вскрыв его. Например, можно поменять один датчик на другой или использовать потенциометр вместо резистора. Эта разборка называется *контролируемой поломкой*, или *электросхемным бэндингом*¹. Этот термин используется для изменения схемы цепи, связанной со звуком. Например, в результате бэндинга ваш плюшевый мишка будет говорить глубоким, чувственным голосом.

¹ От англ. *circuit bending*, дословно «замыкание цепей». — Прим. перев.

Одним из препятствий для разбора устаревшей электроники на запчасти является то, что иногда (на самом деле, как правило) производитель использует скрытые «безопасные» винты, такие как HEX, *Torx* или треугольник *Stymie* ... ну, кто знает, какие еще? Может быть, они сделали их для того, чтобы дети не ломали свои игрушки, а компания не была завалена кучей писем от разгневанных родителей. Одним из решений вскрытия аппаратуры является обнаружение каждой спрятанной шляпки шурупа и применение, например, набора отверток, изображенного на **рис. 8.46**. Он предназначен для раскручивания различных винтов, саморезов, многие из которых вы, вероятно, не используете! А как это приятно, когда вы вдруг понимаете, что нашли все хитрые винтики, чтобы открыть сломанную игрушку!



Рис. 8.46. Этот набор включает более 100 различных размеров и конфигураций наконечников к отвертке для откручивания винтов безопасности

Из детских игрушек с дистанционным управлением можно вытащить двигатели и, конечно, колеса, а иногда аккумуляторные батареи и разномастные выключатели. Лучше всего, если можно починить приемник RC; если контроллер все еще работает, вы могли бы потенциально добавить эту функциональность другому роботу. У старых магнитофонов и кассетных плееров есть колонки, двигатели и зачастую классные переключатели, которые можно вытащить. Другая устаревшая техника представлена пьезоэлементами, клеммами аккумулятора и достаточно надежными корпусами. Я как-то сломал планшетный сканер и получил несколько хороших шаговых двигателей, а также отличные зубчатые колеса и приводные ремни. Это был хороший улов!

К сожалению, большинство современных электронных устройств разработано на платах с поверхностным монтажом компонентов. Это действительно крошечные электронные элементы, которые в основном устанавливаются на платах с помощью машин, так как человеческие пальцы слишком неуклюжие для этой операции. Это означает, что получение запчастей становится действительно затруднительным из-за мелких размеров деталей. Тем не менее для получения какой-либо ценной детали существуют способы для смягчения припоя на платах, после чего деталь можно просто скovyрнуть!

Обозначения и маркировка электронных компонентов

Если вы обнаружили где-либо загадочную электронную детальку, как узнать, что это такое? Конечно, светодиод смотрится иначе, чем датчик наклона, но иногда две радикально разные детали внешне похожи, особенно когда речь идет об интегральных схемах, которые все на одно лицо. Даже если вы сможете распознать элемент, нужно также суметь различить, к какому классу его отнести, потому что, например, два транзистора ведут себя по-разному.

В следующих разделах мы рассмотрим некоторые способы определения элементов.

Номера элементов

Самый простой способ определить элемент — найти место на корпусе, где напечатан номер детали производителя. Иногда эта маркировка включает последовательность букв и цифр, как чип драйвера двигателя L293D, изображенный на **рис. 8.47**. Вы должны уметь читать эту маркировку. В этом случае очевидно, что L293D — это номер детали, а остальное — внутренние ссылки от производителя.

Выберите элемент из коробки запчастей и запустите интернет-поиск по всем номерам, которые вы видите на корпусе. Скорее всего один из номеров даст вам результат, который вы ищете, например ссылку на сайт продавца этой детали или паспорт завода-изготовителя.



Рис. 8.47. Что это за деталь? Поищите в Интернете номера, напечатанные на корпусе

Техническое описание (спецификация)

Электронные детали создают для инженеров, а инженеры хотят иметь доступ ко всевозможной информации, чтобы принять решение о том, какие именно элементы необходимы для проекта. Когда инженер ищет информацию о детали, он загружает данные в формате PDF (рис. 8.48).



FAIRCHILD
SEMICONDUCTOR



October 2008

TIP120/TIP121/TIP122

NPN Epitaxial Darlington Transistor

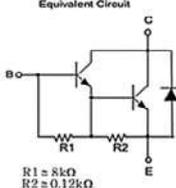
- Medium Power Linear Switching Applications
- Complementary to TIP125/126/127



TO-220

1.Base 2.Collector 3.Emitter

Equivalent Circuit



$R1 \approx 8k\Omega$
 $R2 \approx 0.12k\Omega$

Absolute Maximum Ratings* $T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Ratings	Units	
V_{CBO}	Collector-Base Voltage	TIP120	60	V
		: TIP121	80	V
		: TIP122	100	V
V_{CEO}	Collector-Emitter Voltage	TIP120	60	V
		: TIP121	80	V
		: TIP122	100	V
V_{EBO}	Emitter-Base Voltage	5	V	
I_C	Collector Current (DC)	5	A	
I_{CP}	Collector Current (Pulse)	8	A	
I_B	Base Current (DC)	120	mA	
P_C	Collector Dissipation ($T_A=25^\circ\text{C}$)	2	W	
	Collector Dissipation ($T_C=25^\circ\text{C}$)	65	W	
T_J	Junction Temperature	150	$^\circ\text{C}$	
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$	

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

© 2007 Fairchild Semiconductor Corporation
TIP120/TIP121/TIP122 Rev. 1.0.0

1

www.fairchildsemi.com

TIP120/TIP121/TIP122 — NPN Epitaxial Darlington Transistor

Рис. 8.48. Техническое описание транзистора TIP120 представляет инженерам подробную информацию о том, как использовать этот элемент

Поверьте, вы начнете собирать эти описания скорее всего из необходимости. Вы попытаетесь понять, для чего нужна каждая электронная деталь, а может быть, чтобы почувствовать себя настоящим специалистом. Если абстрагироваться, в мире существует огромный пласт информации, в которой может разобраться только инженер-электрик, не говоря уже о том, что она вряд ли может быть полезна простому обывателю.

Спецификации составляются не только для отдельных компонентов. Иногда их можно обнаружить для сборок и комплектов, например для предварительно спаянной платы, объединяющей кучу разных деталей. Например, набор «Три пятерки» («Three Fives Kit») от *Evil Mad Scientist Laboratories* поставляется с подробным техническим описанием, так что вы можете погрузиться в изучение каждой детали комплекта.

Цветная маркировка резистора

Резисторы — это каверзные штуки, потому что существует множество их конфигураций. Лучший способ определить вид резистора — взглянуть на цветную маркировку на корпусе. Возьмите резистор и посмотрите на него. Вы увидите четыре или пять цветных полос, как на резисторе 470 Ом (рис. 8.49).



Рис. 8.49. Что означают цветные полосы на резисторе?

Вот как это работает. На резисторе вы увидите три полосы, плюс четвертую полосу, часто немного смещенную вправо от других полосок. Эта полоска, как правило, имеет серебряный или золотой цвет.

Каждая цветная полоска соответствует определенной цифре.

Цвет	Цифра
Черный	0
Коричневый	1
Красный	2
Оранжевый	3
Желтый	4
Зеленый	5
Синий	6
Фиолетовый	7
Серый	8
Белый	9

Смотрим на группу из трех полос. Первые две полосы — это базовое значение. Итак, на **рис. 8.49** первая полоса (желтая) означает цифру 4, вторая полоса (фиолетовая) означает цифру 7. Итак, мы получили число «47». Третья полоса — это множитель. Численное значение этой полосы является на самом деле числом нулей, которые надо добавить к числу 47. Поскольку на **рис. 8.49** третья полоса коричневая, надо добавить один «0»; значение резистора будет 470 Ом. Если бы третья полоса была оранжевой, то резистор был бы на 47 000 Ом.

Четвертая полоска обозначает точность сопротивления резистора, так называемый допуск, указывающий процентное несоответствие номиналу. Это важно при сборке схем, в которых сопротивление должно быть откалибровано, а вам необходимо использовать компонент с низкой точностью. Точность номинала постоянного резистора обозначается в процентах и для четвертой полосы существуют специальные цветовые коды.

Цвет	Точность
Серебряный	$\pm 10\%$
Золотой	$\pm 5\%$
Красный	$\pm 2\%$
Коричневый	$\pm 1\%$

Большинство резисторов, с которыми вы столкнетесь, имеют золотую или серебряную четвертую полосу, которая отражает $\pm 5\%$ и $\pm 10\%$ точности соответственно. Для резистора 470 Ом, который мы изучаем, золотая полоса означает, что фактическое значение может варьировать в пределах 447–494 Ом.

Как же запомнить порядок цветовой маркировки? Начинающие инженеры и радиолюбители используют мнемонические приемы, придумывая связный порядок слов, в которых первая буква каждого слова соответствует букве, с которой начинается название цвета. Вы можете придумать его сами, а пока вот вам пример: **Черный Кот Крадется Огородами Животом Задевая Семена Фасоли, Спелые Бобы.**

Схематические обозначения

Электрические схемы — способ, которым инженеры изображают электрические соединения, кажутся очень сложными (см. **рис. 8.50**). На самом деле схемы состоят из ограниченного числа определенных символов, расположенных вместе.

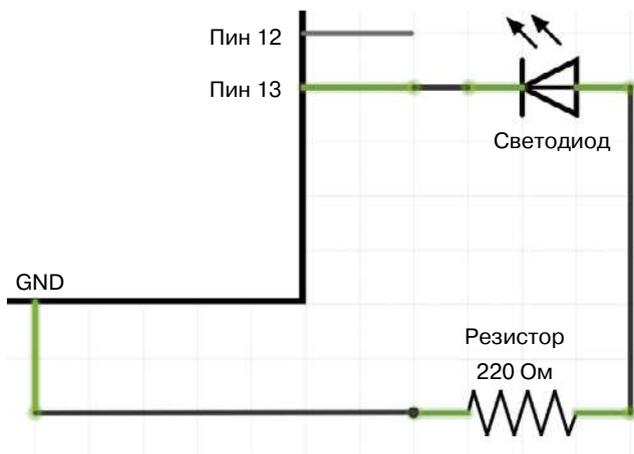


Рис. 8.50. Электрическая схема

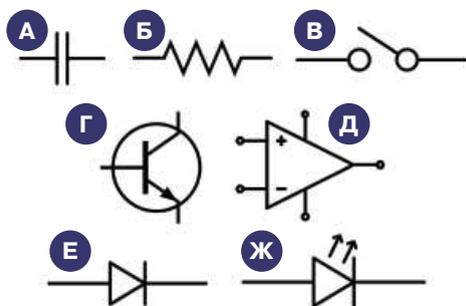


Рис. 8.51. Некоторые распространенные электронные компоненты:

А — конденсатор; Б — резистор; В — выключатель; Г — транзистор; Д — операционный усилитель; Е — диод; Ж — светодиод

На **рис. 8.51** показаны самые общеупотребляемые символы.

Их легко выучить, тем более что многие старые учебники и справочники используют только схематичные изображения, а не фотографии, чтобы описать схему.

В следующей главе

В **главе 9** «Ультразвуковая эхолокация» вы узнаете о датчиках ультразвука и возможности их использования. Вы сделаете забавную игрушку для кошки, которая обнаруживает присутствие животного и начинает с ним играть.

Ультразвуковая эхолокация

В этой главе мы изучим принцип работы ультразвукового датчика. Датчик посылает неслышные для человеческого уха звуковые импульсы, а затем прослушивает их отражение, вычисляя пройденное расстояние.

На основе ультразвуковых эхолокаторов делают дальномеры, но этот прибор также используется для обнаружения препятствий в зоне обзора. В качестве примера рассмотрим проект Стива Хофера «Tacit» (grathio.com/tacit; **рис. 9.1**). Это сонар для слабовидящих людей. Он оборудован парой ультразвуковых датчиков «Ping» с маленькими сервоприводами, которые сжимают запястье пользователя при обнаружении препятствия.



Источник: Steve Hofer

Рис. 9.1. «Tacit» имеет вид перчатки, которая сжимается при обнаружении препятствия

После того как вы освежите в своей памяти знания об эхолокаторе, выполните забавный проект — технологичную кошачью когтеточку. Это устройство знает, когда животное находится поблизости, и пытается играть с ним с помощью моторизованной игрушки, двигающейся в разных направлениях.

Урок: ультразвуковая диагностика

Сонар, который используется в проекте по созданию игрушки для кошки, — ультразвуковой дальномер «MaxBotix LV-EZ1». Это недорогой, но надежный датчик, у которого есть множество применений, например измерение расстояния и обнаружение объектов, в том числе прозрачных, как, например, оконное стекло или емкость с водой, которые создают некоторые затруднения для датчика освещенности.

Принцип работы состоит в том, что ультразвуковой модуль может и отправлять ультразвуковые импульсы с периодом примерно 20 импульсов в секунду, и одновременно принимать отраженные звуковые волны (**рис. 9.2**). Как уже упоминалось, по такому же принципу работает сонар летучей мыши.

Конечно, этот метод не является совершенным, потому что некоторые объекты определенной текстуры или формы могут исказить отра-

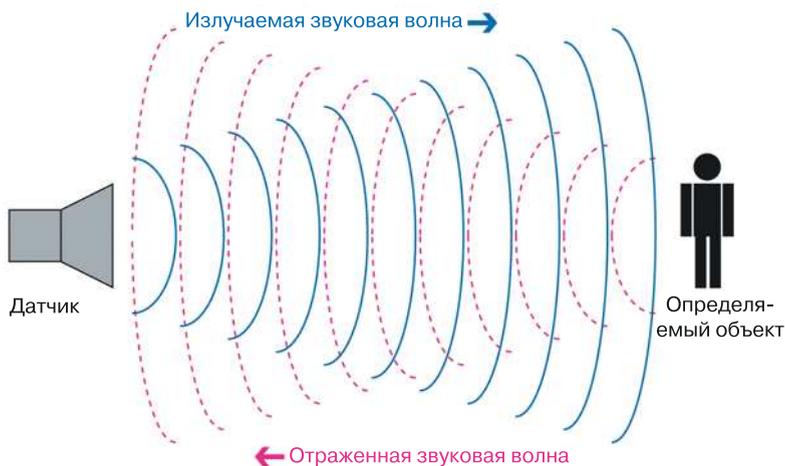


Рис. 9.2. Датчик ультразвука определяет объект с помощью отраженной звуковой волны

женный сигнал. Мягкий объект, такой как кошка или подушка, может поглотить звуковые волны, в то время как поверхность с выступающими острыми углами может изменить звуковую волну.

С какого расстояния датчик может уловить объект? Насколько точным будет полученный результат? Каковы минимальные размеры объекта, который можно обнаружить? Уф! Ответить на эти вопросы не так просто. Существует большое разнообразие моделей датчика ультразвука. Все они имеют разную чувствительность, углы зондирования, диапазон действия и разрешение. Например, ультразвуковой датчик «Parallax Ping» имеет диапазон действия 3 м, чувствительность 2 см и узкий угол обнаружения объекта. Датчик «MaxBotix EZ-0» имеет диапазон 6 м и очень высокую чувствительность, что делает его идеальным устройством для мониторинга, в отличие от «MaxBotix EZ-4» с очень узким углом излучения. Большинство начинающих пользователей выбирают «MaxBotix EZ-1», потому что его характеристики представляют оптимальное значение.

Другие датчики могут иметь более дальний диапазон или более широкий угол действия, обнаружить более мелкие объекты. Прежде чем купить датчик, проверьте его спецификацию. Наконец, если вы выбираете эхолот для точного обнаружения объектов на большом расстоянии, вы должны знать, что на работу датчика влияет изменение температуры.

Применение датчика ультразвука

Датчик ультразвука можно применять для разных целей. Вот некоторые примеры.

- Определение количества воды, оставшейся в резервуаре, путем направления звукового луча вниз на поверхность жидкости и вычисления оставшегося количества жидкости в зависимости от высоты уровня воды в резервуаре.
- Создание дисплея для магазина или киоска, который активируется, когда подходит покупатель.
- Создание системы сигнализации в заднем бампере автомобиля, которая издает предупреждающий сигнал водителю, если бампер приближается слишком близко к какому-либо объекту. Более того, такое устройство может совершенно точно определить, сколько осталось свободного места, что помогает сэкономить время при сложных параллельных парковках!
- Разработка модели железнодорожного макета, который точно направляет состав, соответственно переключая пути и открывая ворота.

Датчик ультразвука — отличное устройство, которое подходит для огромного числа проектов! Для любого творения, требующего точного определения небольшого расстояния, ультразвук — это то, что надо!

🔧 Проект: ультразвуковой ночник

Давайте выполним простой проект с использованием Arduino и ультразвукового датчика — сделаем осветительный прибор, который включается, когда вы проходите мимо датчика. Другими словами, ваше движение включает ночник (рис. 9.3).

Вам может показаться, что использовать Arduino в этом проекте — лишнее; для того чтобы светодиод сработал на датчик, микроконтроллер совершенно не нужен. Тем не менее эта книга об Arduino, так что без него никуда!

Принцип работы конструкции заключается в том, что сонар постоянно просматривает определенную зону (в переносном смысле слова, конечно) и включает светодиод, когда кто-то проходит мимо. Не сомневайтесь, что единственный светодиод может служить хорошим ночником; например, светодиод синего цвета испускает удивительное количество света! Если хотите, можете поменять светодиодный модуль на другой, например «ShiftBrite» или любой другой, чтобы по-настоящему осветить зону!

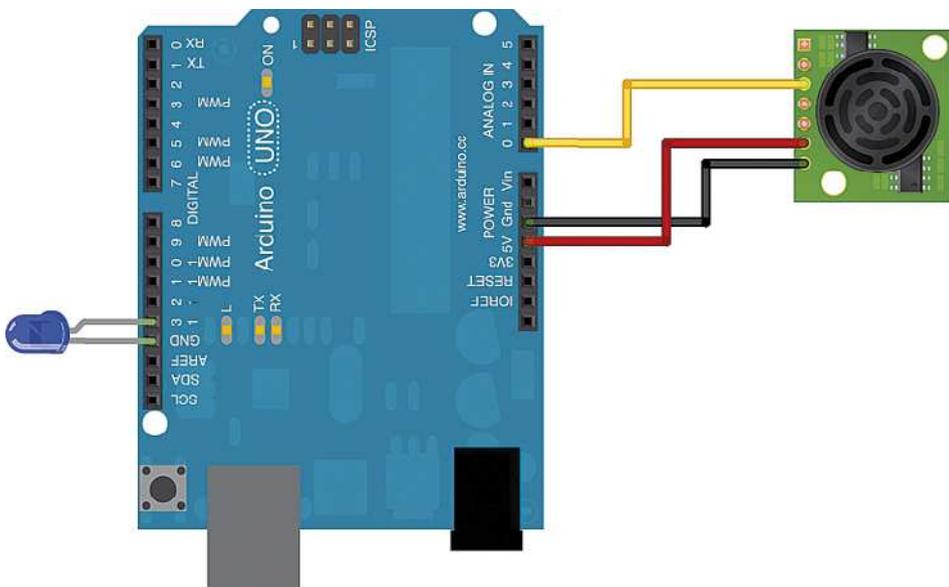


Рис. 9.3. Кажется, что это просто? Так и есть!

Код для ночника

Код предназначен для использования одного светодиода. Если вы решите использовать «ShiftBrite» или другой светодиодный модуль, вам нужно изменить код. См. код для основного проекта в **главе 6**, чтобы получить представление о том, как это сделать.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
int led = 13;
void setup()
{
  pinMode(led, OUTPUT);
}
void loop() {
  int distsensor, i;

  distsensor = 0;
  for (i=0; i<8; i++) {
    distsensor += analogRead(0);
    delay(50);
  }

  if (distsensor < 500) {
    digitalWrite(led, HIGH);
    delay(30000);      // ожидает 3 минуты, затем перепроверяет
  }
}
```

🔧 Проект: игрушка для кошки

Проект для этой главы предполагает создание забавной кошачьей игрушки, которая развлекает вашего домашнего питомца. Проект состоит из кошачьей когтеточки с моторизованной игрушкой, которая болтается туда-сюда, изображая что-то вроде летучей мыши, когда кошка подходит поточить когти (**рис. 9.4**).



Рис. 9.4. Проект разработан для развлечения вашей любимой кошки

Для данного проекта вы изготовите корпус (когтеточку), сделав деревянный цилиндр на токарном станке. Это машина, которая вращает кусок материала и позволяет придавать ему желаемую форму с помощью подручных инструментов. Если вы научитесь им пользоваться, то сможете сделать много полезных вещей.

К слову, добросердечные читатели могут обеспокоиться тем, что ультразвук навредит кошке. Не волнуйтесь, ультразвук находится за пределами кошачьего слухового диапазона, другими словами, кошки не слышат ультразвук!

В любом случае, давайте начнем!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Эта простая конструкция состоит из нескольких частей. Вам понадобятся следующие компоненты.

- Arduino UNO.
- Кабель USB для Arduino с сетевым адаптером или блок питания.

- Соединительные провода.
- Термоусаживаемые трубки.
- Ультразвуковой дальномер «MaxBotix LV-EZ1» (вы можете использовать другие марки и модели сонара).
- Сервопривод (я использовал «Hitec HS-322HD»; см. далее раздел «Сервоприводы»).
- Толстый провод (я использовал сварочный электрод, но нет ничего лучше проволочной вешалки-плечиков для пальто).
- Изоляционные прокладки (я использовал пластиковые втулки высотой 10 мм).
- Прямоугольные скобы.
- Крепежный винт № 4 с гайкой и шайбой.
- Саморезы для дерева № 4 × 20 мм.
- Саморезы для дерева № 6 × 20 мм.
- Саморезы для дерева № 6 × 50 мм.
- Столярный клей.
- Древесина для корпуса (я использовал сосновые доски).
- Любимая игрушка вашей кошки в виде помпона.
- Дрель и сверла для дрели (3 мм, 5 мм, 12 мм, 25 мм).
- Токарный станок.
- Циркулярная пила.
- Ленточная пила.
- Кольцевая пила для выпиливания отверстий.
- Клеевой пистолет.
- Острогубцы.

СЕРВОПРИВОД

В **главе 13** «Управление электродвигателями» подробно рассказывается о разных замечательных двигателях, в том числе сервоприводах, которые можно использовать в работе над проектами. Вместе с тем, давайте узнаем поподробнее о конкретном двигателе, который используется в этом проекте.

Сервопривод «Hitec HS-322HD» — это отличный универсальный двигатель. Если бы вы могли позволить себе только один двигатель, то выбрали бы именно этот. Вместе с тем, существует огромное разнообразие сервоприводов, и вы можете выбрать любой, если решите, что «Hitec HS-322HD» слишком медленный, или слишком большой, или недостаточно мощный. У каждого двигателя есть спецификация, которую можно изучить, прежде чем принять решение о покупке.

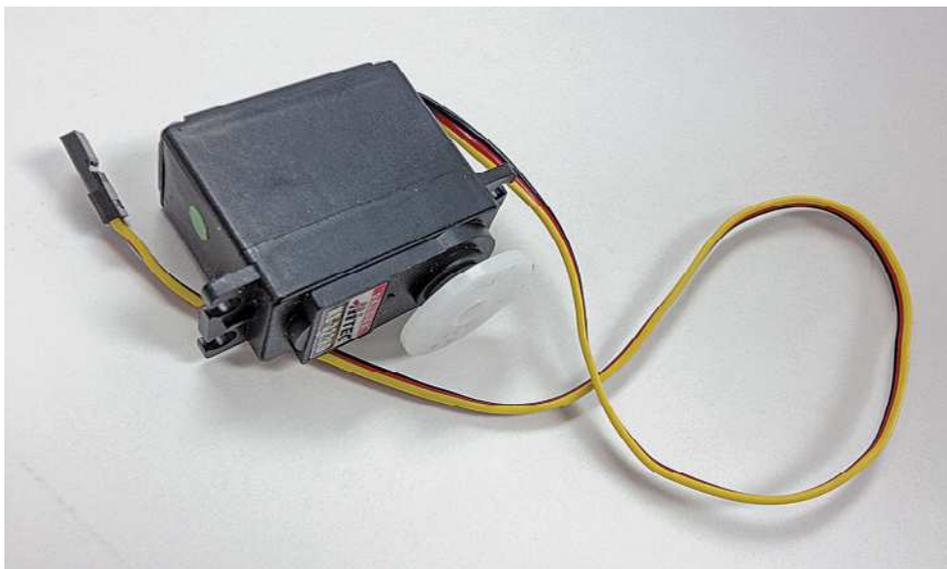


Рис. 9.5. «Hitec HS-322HD» — отличный универсальный сервопривод

Вот технические характеристики «Hitec HS-322HD». При напряжении в 4,8 В он поворачивается на 60° за 19 с, при 6 В — за 0,15 с. Крутящий момент (мощность двигателя) составляет 3 кг/см при 4,8 В и 3,7 при 6 В. О чем это говорит? По сути, это означает, что «Hitec HS-322HD» является «среднячком» во многих отношениях. Если вам нужен более быстрый двигатель, у компании «Hitec» есть в продаже подходящий экземпляр, который работает в два раза быстрее: вращается на 60° за 0,9 с при 4,8 В. При желании, можно купить и более мощные двигатели.

Инструкции

Сначала вы подключите электронику, а затем сделаете деревянный корпус для проекта, который напоминает кошачью когтеточку.

Подключите Arduino так, как показано на **рис. 9.6**.

1. Подключите желтый провод сервопривода к цифровому пину 9. Черный провод — к свободному пину GND. Пока не трогайте красный провод, он понадобится для другого.
2. Подключите провод данных ультразвукового датчика (фиолетовый на **рис. 9.6**) к аналоговому пину 0 на Arduino и провод заземления (серый на **рис. 9.6**) к пину GND. Оранжевый провод вы подключите в следующем шаге!
3. Единственная хитрость подключения Arduino заключается в том, что и двигателю, и датчику необходим пин 5V, а на микроконтрол-

лере только один такой вывод. Что же делать? Можно создать сращенный провод (см. **рис. 9.7**) путем спаивания трех проводов вместе и с последующей изоляцией места соединения с помощью термоусаживаемой трубки (изображено как красный провод, идущий от сервопривода, и оранжевый провод, идущий от датчика, на **рис. 9.6**). Одиночный провод идет к пину 5V на плате Arduino, в то время как два других соединяются с выводами датчика ультразвука и питания сервопривода.

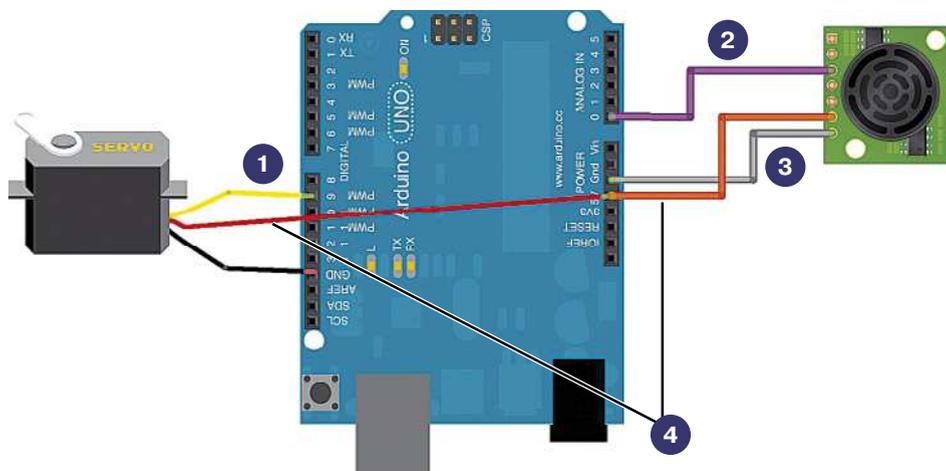


Рис. 9.6. Подключить игрушку для кошки легко:

- 1 — желтый провод сервопривода подключается к цифровому пину 9 на Arduino;
- 2 — провод данных ультразвукового датчика идет к аналоговому пину 0 на Arduino;
- 3 — провод заземления от ультразвукового датчика идет к пину GND на Arduino;
- 4 — сращенный провод подключает сервопривод и датчик к пину 5V на Arduino



Рис. 9.7. Сращенные провода

В следующем разделе вы узнаете, как сделать корпус в виде кошачьей когтеточки, а затем соедините все части проекта вместе.

Корпус

После того как вы собрали электрическую схему, самое время поместить ее в корпус. Но сначала надо его построить! Чтобы построить корпус круглой формы, показанный на **рис. 9.8**, воспользуйтесь токарным станком.

Забудьте о лазерном резаке! Чтобы сделать этот корпус из дерева, мы вспомним старую школьную технологию (см. **рис. 9.8**). Нарезьте несколько колец из древесины, склейте их, а затем отшлифуйте с внешней стороны с помощью электроинструмента под названием «токарный станок». Выполните следующие действия.

1. Начертите кольца на сосновых заготовках. В качестве шаблона я использовал рулон скотча (**рис. 9.9**). Необязательно брать древесину хорошего качества, в качестве материала я использовал обрезки древесины, оставшиеся от другого проекта. Для кошачьей игрушки я нарезал десять колец толщиной около 2,5 см, плюс пластины для дна и крышки.



Рис. 9.8. На токарном станке можно сделать цилиндр!



Рис. 9.9. Чертим кольца с помощью рулона скотча

2. Вырежьте круги с помощью ленточной пилы. Я предлагаю сделать прямые пропилы в древесине (**рис. 9.10**), поскольку лезвие ленточной пилы иногда имеет проблемы с пропилами по изогнутой линии.



Рис. 9.10. Вырезаем круги с помощью ленточной пилы



Рис. 9.11. Вырезаем внутренности

3. Вырежьте внутреннюю часть колец с помощью кольцевой пилы (**рис. 9.11**). Вероятно, мне нужно было сначала вырезать внутренности, прежде чем делать внешние пропилы, потому что круги стали вращаться вокруг дрели! После появления мозолей на пальцах, я догадался использовать зажим для крепления дисков.

СОВЕТ

НЕ СВЕРЛИТЕ ОТВЕРСТИЯ В КАЖДОМ КРУГЕ!

Не забудьте оставить пару деревянных дисков целыми (без вырезания внутренней части). Эти диски понадобятся для дна и верха корпуса!

4. Уложите кольца друг на друга, оставив два нетронутых диска для верха и низа корпуса. Расположите их настолько аккуратно, насколько сможете, и склейте вместе, как показано на **рис. 9.12**.



Рис. 9.12. Склейте кольца вместе, чтобы получился цилиндр

5. Придавите стопку колец грузом (рис. 9.13) и оставьте сборку сохнуть на ночь. Попробуйте сделать ваш штабель чуть аккуратнее, чем у меня, но он не должен быть идеальным. В конце концов, есть токарный станок, чтобы довести поверхность до совершенства.



Рис. 9.13. Стопке колец необязательно быть совершенной! Токарный станок обработает ее



Рис. 9.14. Размещаем склеенную стопку на токарном станке

6. Поместите склеенную стопку дисков на токарный станок (**рис. 9.14**). Вот где пригодились целые диски для верхней и нижней частей корпуса. Используйте токарные инструменты (см. ниже раздел «Токарный станок 101»), чтобы отшлифовать стороны.

Обработанная на токарном станке стопка дисков выглядит великолепно! Вы можете увидеть малую конусность в середине цилиндра, изображенного на **рис. 9.15**. Просто я слишком сильно нажал



Рис. 9.15. Токарный станок завершил работу



Рис. 9.16. Обрезаем и просверливаем цилиндр

на инструмент в середине процесса. Это можно было исправить, но я не беспокоился по этому поводу!

7. Обрежьте верхнюю и нижнюю части кольцевого цилиндра, чтобы поместить внутрь электронику, как показано на **рис. 9.16**, вырежьте отверстия для датчика ультразвука и источника питания.
8. Добавьте сервопривод. Нужно просверлить отверстие для толстой проволоки, на которой будет болтаться игрушка для кошки. Используйте сверло 5 мм, чтобы просверлить насквозь верхний диск. Затем просверлите вниз примерно на 6 мм более широким сверлом. Я использовал сверло 12 мм, чтобы приспособить ступицу сервопривода, которая несколько выступает (**рис. 9.17**). Кроме этого, надо просверлить отверстия для крепежа, который соединяет двигатель с корпусом, я использовал сверло 2,5 мм. Будет понятно, где сверлить эти отверстия, когда вы приложите двигатель.



Рис. 9.17. Сверлим отверстие для провода с углублением для втулки сервопривода

9. Присоедините толстую проволоку к втулке электродвигателя. Я использовал термоклей, но вы можете выбрать и более надежное средство. Затем проденьте провод через отверстие диаметром 5 мм, которое вы просверлили в крышке, и закрепите двигатель с помощью саморезов для дерева № 4 × 20 мм и пластиковых втулок диаметром 10 мм, как показано на **рис. 9.18**.

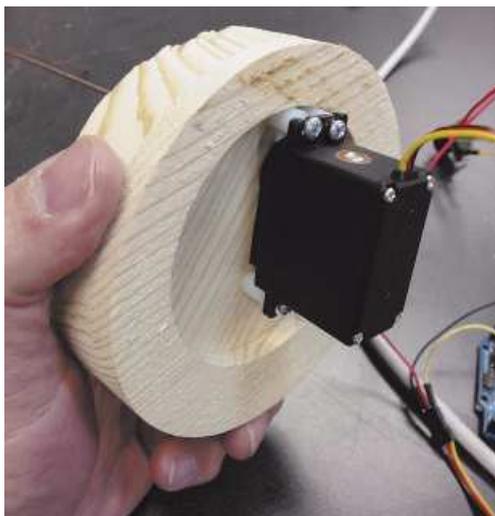


Рис. 9.18. Установка двигателя с помощью саморезов и втулок

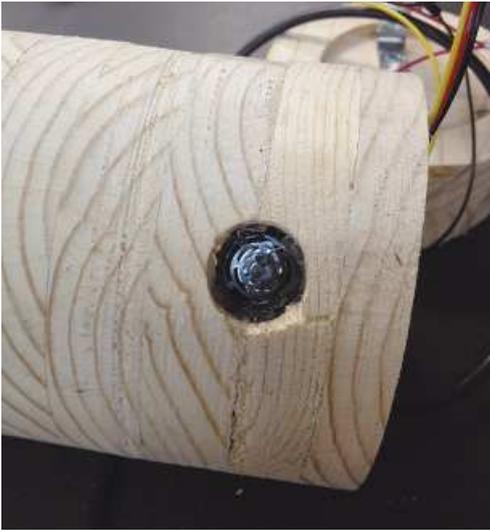


Рис. 9.19. Датчик ультразвука выглядывает из корпуса!

10. Вставьте датчик ультразвука внутрь через основание цилиндра и расположите его напротив отверстия, которое вы просверлили с помощью сверла 25 мм (**рис. 9.19**). Чтобы закрепить датчик, я использовал термоклей. На самом деле в корпусе датчика есть отверстия для винтов, но я решил, что они неудобны для крепежа.
11. Прикрепите Arduino так, как показано на **рис. 9.20**. Чтобы подключить Arduino, я использовал металлические кронштейны и крепежный винт № 4. Как расположить квадратный объект внутри цилиндра? Не раздумывая, закрепите его за один край!

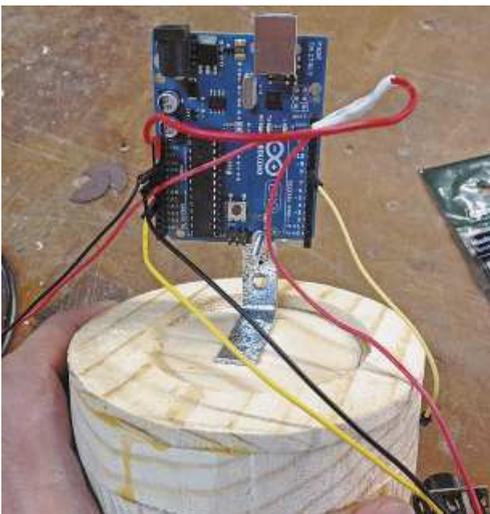


Рис. 9.20. Устанавливаем Arduino



Рис. 9.21. После того как двигатель установлен, закрутите верхнюю крышку

12. После того как вы установите двигатель и подключите все провода, закрутите верхнюю крышку (**рис. 9.21**) саморезами № 6 × 20 мм. Сделайте то же самое с дном.
13. Сделайте основание. С помощью циркулярной пилы я вырезал квадрат со стороной 25 см из МДФ толщиной 3 см, как показано на **рис. 9.22**. Он имеет солидный вес и придаст конструкции доста-



Рис. 9.22. Прикрепляем цилиндр к основанию

точную устойчивость. Прикрепите цилиндр к основанию с помощью саморезов для дерева № 6×5 мм. Одно уточнение: избегайте случайного попадания в саморезы, которые уже удерживают днище цилиндра. Самый простой способ — вкручивать саморезы в центре основания, а не по краям. Не волнуйтесь, вы не попадете в винты, которые крепят кронштейн в днище, потому что древесина слишком толстая.

ТЕХНИКА БЕЗОПАСНОСТИ

Кстати, вы видите, этот странный прорез на доске на **рис. 9.22**? Циркулярная пила захватила древесину и швырнула ее обратно в мою голову! К счастью, она промахнулась! Работа с деревом может быть опасной; убедитесь, что используете инструменты должным образом и с соблюдением всех мер предосторожности.

14. Оберните когтеточку куском ткани, чтобы кошка могла поточить когти, если ей наскучит играть с помпоном. Я использовал вельвет (может быть это и не самый лучший материал, но выглядит здорово!). Другие варианты — кусок старого ковра, приклеенного или прибитого металлическими скобками к дереву, или сизалевый шпагат, которым можно обмотать цилиндр. Я измерил окружность и высоту когтеточки гибкой рулеткой. Затем вырезал вельвет ножницами (**рис. 9.23**). После этого нанес клей на когтеточку, обернул ее тканью и подождал,



Рис. 9.23. Готовим ткань для когтеточки



Рис. 9.24. Кошке это нравится!

пока клей высохнет. В завершение, с помощью ножа я вырезал отверстия для датчика ультразвука и шнура питания. С помощью круглогубцев согните конец толстой проволоки в петлю и прикрепите к ней помпон. Вы закончили!

15. Теперь осталось найти кошку (см. **рис. 9.24**).

Токарный станок 101

Токарный станок (см. **рис. 9.25**) представляет собой двигатель, который вращает кусок дерева или металла вокруг своей оси так, что он может быть обработан режущим инструментом или отшлифован, отполирован, окрашен и т. п. Некоторые станки имеют крепления, позволяющие высверливать внутри заготовки цилиндр; к сожалению, в проекте я использовал станок без этой функции.

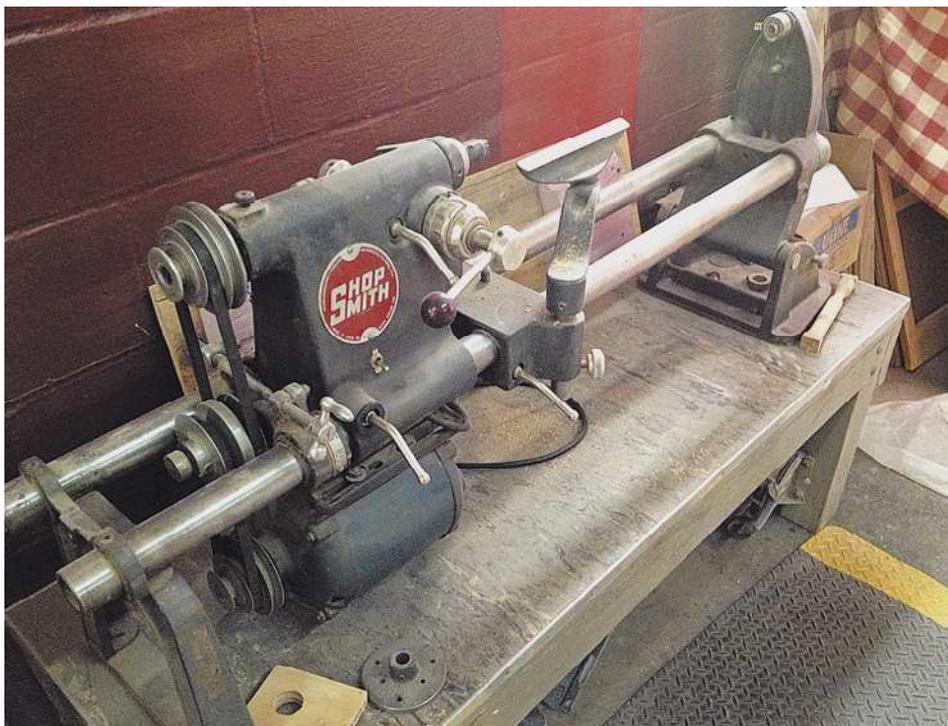


Рис. 9.25. Деревообрабатывающий токарный станок — отличный инструмент для любой мастерской

Итак, что вы хотите вырезать из вращающейся заготовки? В принципе, можно сделать красивые ножки столов и подсвечники. Хотя существует много типов токарных станков, эта дискуссия относится к классическим обсуждениям столярных инструментов.

Как работать с заготовкой на токарном станке?

1. Подготовьте заготовку. Можно заранее сформировать заготовку, приближенную к оригиналу, с помощью ручного инструмента. Этим вы сэкономите время работы на токарном станке.
2. Закрепите заготовку в токарном станке. На глазок отцентрируйте ее закрепление. Это можно проделать, перемещая заготовку на центре упорного винта или используя зубчатую оправку в месте крепления к шпинделю.
3. Вращая деталь на токарном станке, обрабатывайте ее с помощью деревообрабатывающей стамески (см. **рис. 9.26**), пока не добьетесь желаемой формы.



Рис. 9.26. Используйте деревообрабатывающие стамески с длинными рукоятками для врезки в крутящуюся древесину

Если вы хотите узнать больше о работе на токарных станках, я советую зайти на *YouTube* и ввести поисковый запрос: «Как работать с токарным станком по дереву».

Техника безопасности при работе с токарным станком

При работе со станком соблюдайте следующие очень важные правила безопасности.

- Важно, чтобы опытный оператор допустил вас к токарному станку; другими словами, чтобы кто-то провел с вами инструктаж по работе с машиной.
- Используйте средства защиты органов зрения и слуха при работе с устройством.
- Перед запуском двигателя убедитесь, что все регулируемые части станка в наличии и исправны.
- Токарный станок может зацепить волосы и рукава. Это опасно, так что спрячьте волосы под берет или косынку и избегайте таких деталей в одежде, как галстуки и пышные рукава.

В следующей главе

В **главе 10** вы узнаете, как с помощью Arduino генерировать электронные звуки. Вы создадите мелодичный портативный звуковой генератор.

Генерация звука

С помощью Arduino можно сделать множество полезных вещей, и одна из них — генератор звука! Все, что нужно, — это динамик, подключенный к паре портов на Arduino. Однако вы можете добавить кнопки, ручки и датчики, чтобы изменять свойства звука. В этой главе мы рассмотрим несколько способов генерации забавных звуков, а затем создадим занимательную вещицу, генерирующую звуки (**рис. 10.1**).

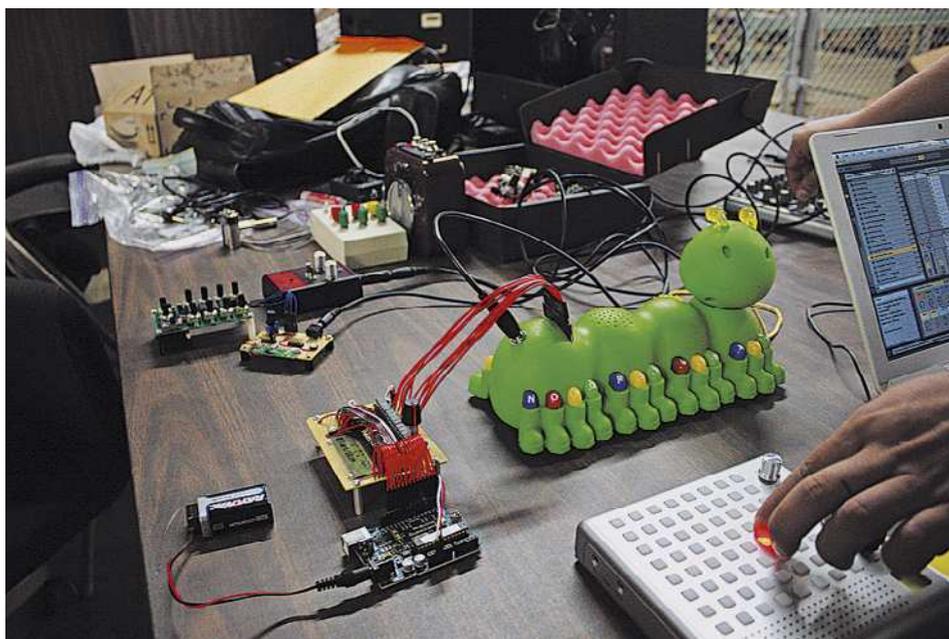


Рис. 10.1. Звуковой генератор

Звуки электроники

Люди создавали электронную музыку с момента зарождения электроники, и самые творческие и изобретательные на этом поприще — любители, которые увлекаются электроникой дома.

Например, рассмотрим феномен *электросхемного бэндинга* (*circuit bending*)¹. С помощью этого метода изменения и замыкания электрических цепей (рис. 10.2) были созданы многие модификации существующих звуковых генераторов, таких как игрушка «Поговори со мной» («Speak & Spell»)². Впечатляющие звуковые эффекты создаются методом подбора электронной начинки.



Источник: Pat Arneson

Рис. 10.2. Модифицированная игрушечная гусеница

¹ *Circuit bending* — метод произвольного замыкания (изменения) электрических цепей для создания звуковых эффектов, в частности электронных инструментов. Создателем считается музыкант Рид Газала, который в 60-е годы XX в., не имея средств купить электронные инструменты, попытался сделать их сам из примитивных звукоусиливающих приборов — гитарных усилителей и радиоприемников. — *Прим. перев.*

² «Speak & Spell» — это игрушечное электронное устройство производства компании *Texas Instruments* 1978 г., в работе которого был использован цифровой процессор обработки сигналов. Синтез человеческого голоса осуществлялся упреждающим линейным кодированием при создании математической модели речи. — *Прим. перев.*

Один из способов генерации различных звуков с помощью электроники заключается в замене постоянного резистора на потенциометр. Поворот ручки потенциометра меняет сопротивление и может изменить звучание игрушки. Другие методы заключаются в подборе разных конденсаторов, добавлении микрочипов-таймеров и даже в подключении к Arduino или другому микроконтроллеру для управления различными звуковыми эффектами игрушки.

Однако электросхемный бэндинг — не единственный метод создания электронной музыки. Помимо профессиональных звуковых генераторов, а также профессиональных музыкальных синтезаторов существует масса замечательных проектов. Многие из них основаны на изменении тона с использованием потенциометра и кнопок.

Некоторые любительские проекты оказались настолько успешными, что их производство было поставлено на коммерческую основу. Обычно эти звуковые генераторы продаются в виде наборов. Вы можете собрать их самостоятельно. Давайте рассмотрим несколько удачных примеров самодельной звуковоспроизводящей аппаратуры, которые стали самостоятельным коммерческим продуктом.

Синтезатор «Thingamagoor»

Компания *Bleep Labs* (Остин, штат Техас), взяв за основу метод подбора в схеме, создала синтезатор «Thingamagoor» (см. **рис. 10.3**). Он оборудован переключателями, ручками и кнопками, а также датчиком освещенности и светодиодной антенной. В синтезаторе можно выбрать и воспроизвести музыкальные, шумовые, трескучие и другие звуковые эффекты.



Рис. 10.3. «Thingamagoor» отлично выглядит и издает фантастические звуки

Кнопочный метроном

Разработчики электронных наборов компании *Wayne & Layne* создали аккуратный комплект, изображенный на **рис. 10.4**. Если простучать пальцем по пьезодинамике, то микроконтроллер обнаружит вибрацию и запишет шаблон, который вы настучали. Затем устройство воспроизведет запись, чтобы вы могли изменить ритм музыки в соответствии с вашим настроением.



Источник: Уэйне & Лэйне

Рис. 10.4. Тактильный метроном от компании *Wayne & Layne*

Синтезатор «LushOne»

Иэн Шарп создает сложные модульные синтезаторы, такие как «LushOne» (**рис. 10.5**). Вы можете управлять им с помощью компьютера, музыкальной клавиатуры, джойстика, переменного резистора или ультразвукового датчика. Модульные синтезаторы имеют несколько эффектов на отдельных печатных платах (отсюда и название). Можно подключать различные модули с помощью соединительных кабелей, создавая галерею настраиваемых эффектов.



Рис. 10.5. Синтезатор «LushOne», созданный Иэном Шарпом, хорошо смотрится внутри изящной музыкальной шкатулки

КОНСОЛЬ «ATARI PUNK»

Консоль «Atari Punk» появилась в 1980 г. в рамках проекта Форреста Мимса (Forrest Mims), который был включен в буклет «Radio Shack». Это простой звуковой генератор, собранный на двух микросхемах-таймерах NE555. Одна микросхема настроена на вывод в качестве генератора сигнала, который создает синусоидальный аналоговый сигнал, в то время как другая — это моностабильный мультивибратор (с функцией «вкл.» и «выкл.»). Оба таймера управляются потенциометрами. Вместе они создают забавное разнообразие электронных звучаний.

Консоль «Vibrati Punk», изображенная на **рис. 10.6**, была создана Иэном Шарпом из «Lushprojects.com». Это вариант консоли «Atari Punk» с низкочастотным генератором сигнала, который увеличивает шумовые эффекты и «размазанность» звука.



Рис. 10.6. Консоль «Vibrati Punk» — модификация консоли «Atari Punk» (APC)

⚙ Проект: мелодичная кнопка

Давайте ускоримся! Настала пора создавать звуки с помощью Arduino! Вы прослушаете песню через динамик, если нажмете на кнопку (**рис. 10.7**). Для электронного звучания я запрограммировал «Оду к радости»!¹

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Это быстрый проект, поэтому вам потребуется немного комплектующих.

- Динамик. Я предлагаю динамик 1 Вт диаметром 8 см на 8 Ом.
- Кнопочный переключатель.
- Резистор 220 Ом.
- Соединительные провода и макетная плата.

¹ «Ода к радости» — известное музыкальное произведение Л. Бетховена на слова Ф. Шиллера. Созданная композитором в 1823 г. в составе знаменитой 9-й симфонии, «Ода к радости» является официальным гимном Европейского союза. — *Прим. перев.*

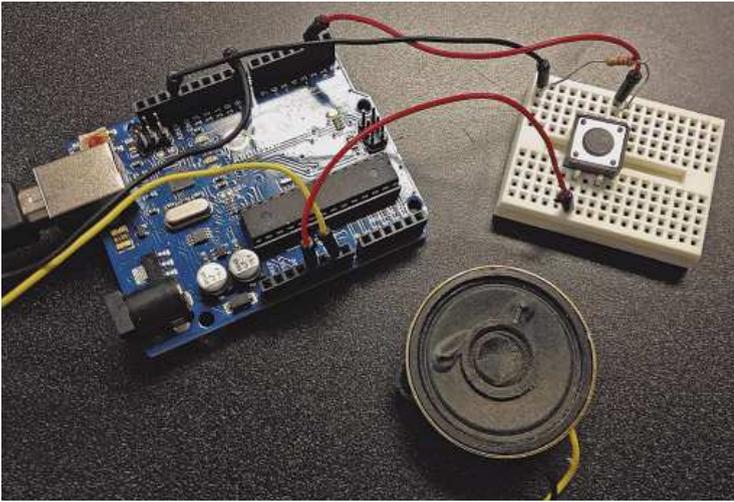


Рис. 10.7. В этом проекте вы услышите электронную «Оду к радости»

Инструкции

Этот проект очень легко собрать. Просто изучите схему подключений на **рис. 10.8**, чтобы понять, как соединить провода.

1. Подключите динамик к пину 8 (красный провод на **рис. 10.8**).
2. Подключите пин GND (черный провод) от динамика к Arduino.
3. Соедините пин 5V Arduino с одним из выводов кнопки. Это красный провод.

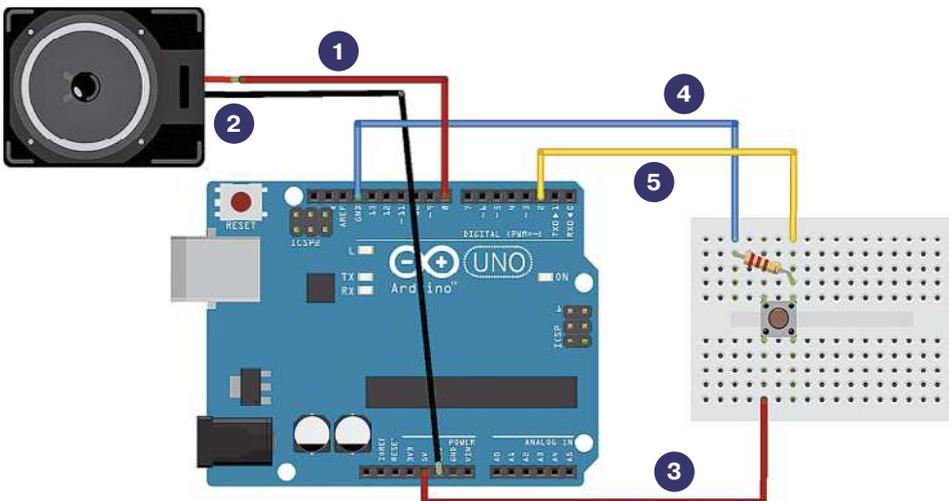


Рис. 10.8. Подключите Arduino так, как показано на рисунке

4. Подключите другой провод кнопки к пину GND через резистор (синий провод).
5. Подключите кнопку к пину 2 на Arduino (желтый провод).

Код для музыкальной кнопки

Может быть, выглядит он и не очень элегантно, но этот код научит вас воспроизводить звуки с помощью Arduino. Почему я говорю «не очень элегантный»? Вы видите строки `Tone()`? Я использую целый набор этих команд вместо функции `For loop`, которая описана в **главе 5** «Программирование Arduino». В этом случае я сделал скетч настолько наглядным, чтобы вы могли с ним повозиться.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
const int buttonPin = 2;
const int ledPin = 13;
int pbState = 0;

void setup() {
    pinMode(buttonPin, INPUT);
}

void loop(){
    pbState = digitalRead(buttonPin);
    if (pbState == HIGH) {

        tone(8, 247, 300);
        delay(500);
        tone(8, 247, 300);
        delay(500);
        tone(8, 262, 300);
        delay(500);
        tone(8, 294, 300);
        delay(500);
        tone(8, 294, 300);
        delay(500);
    }
}
```

```
tone(8, 262, 300);  
delay(500);  
tone(8, 247, 300);  
delay(500);  
tone(8, 220, 300);  
delay(500);  
tone(8, 196, 300);  
delay(500);  
tone(8, 196, 300);  
delay(500);  
tone(8, 220, 300);  
delay(500);  
tone(8, 247, 300);  
delay(500);  
tone(8, 247, 500);  
delay(650);  
tone(8, 220, 200);  
delay(250);  
tone(8, 220, 200);  
delay(250);  
}  
}
```

⚙ Проект: звуковой генератор

Испробовав «на зубок» мелодичную кнопку, давайте перейдем к настоящему проекту этой главы — звуковому генератору, который уместится на вашей ладони как игровой контроллер (см. **рис. 10.9**). Он издает всевозможные звуки. Вы можете сделать такое устройство сами. Давайте начнем!



Рис. 10.9. Проект звукового генератора с Arduino для создания сумасшедших звуков

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Несмотря на то что звуковой генератор небольшого размера (чуть больше, чем Arduino внутри него), для его сборки понадобится много комплектующих! Вот что вам нужно.

- Фанера толщиной 3 мм. Я использовал два куска размером 8 × 10 см для дна и крышки.
- Динамик. Я взял динамик диаметром 4 см на 8 Ом, 0,25 Вт. Он хорошо смотрится на передней панели генератора, но для меня оказался слишком тихим. Поэтому я предлагаю динамик диаметром 5 см на 8 Ом, 0,5 Вт.
- Переключатель.
- Потенциометры на 2 кОм, 2 шт.
- Датчик освещенности.
- Резисторы (я использовал 3,3 кОм и 470 Ом).
- Батарейка 9 В.
- Зажим для батарейки.
- Разъем питания для батарейки.
- Пластиковые втулки № 4-40 × 9 мм.
- Алюминиевые стойки (я использовал шестигранные резьбовые № 4-40 длиной 38 мм).
- Клеевой термопистолет.
- Дрель и сверла 4 мм, 9 мм, 20 мм.
- Болты № 4-40 × 25 мм.
- Болты № 4-40 × 6 мм.
- Гайки и шайбы № 4-40 в ассортименте.
- Красный, желтый и черный провода.
- Изоляционные трубки.

Инструкции

Итак, давайте примемся за сборку!

1. Вырежьте крышку и дно из фанеры, как показано на **рис. 10.10**. Я вырезал заготовки размером 8 × 10 см, обработав углы шлифовальным кругом.

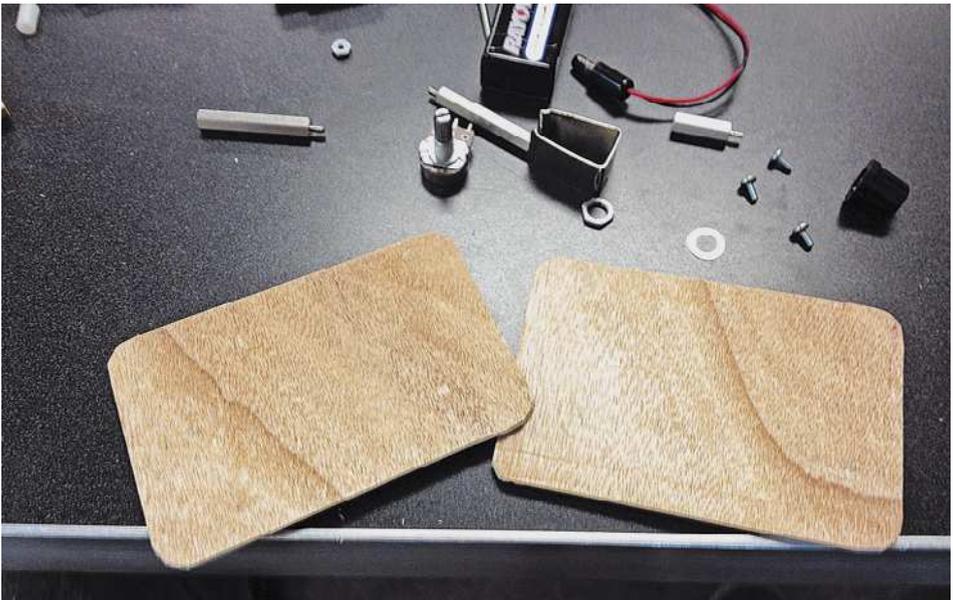


Рис. 10.10. Первый шаг — вырезать крышку и дно для звукового генератора

2. Начиная с нижней части, просверлите отверстия для втулок и зажима батареи, используя сверло 4 мм. Вкрутите болты 25 мм и пластиковые втулки для Arduino так, как показано на **рис. 10.11**.



Рис. 10.11. Вкрутите болты в просверленные отверстия

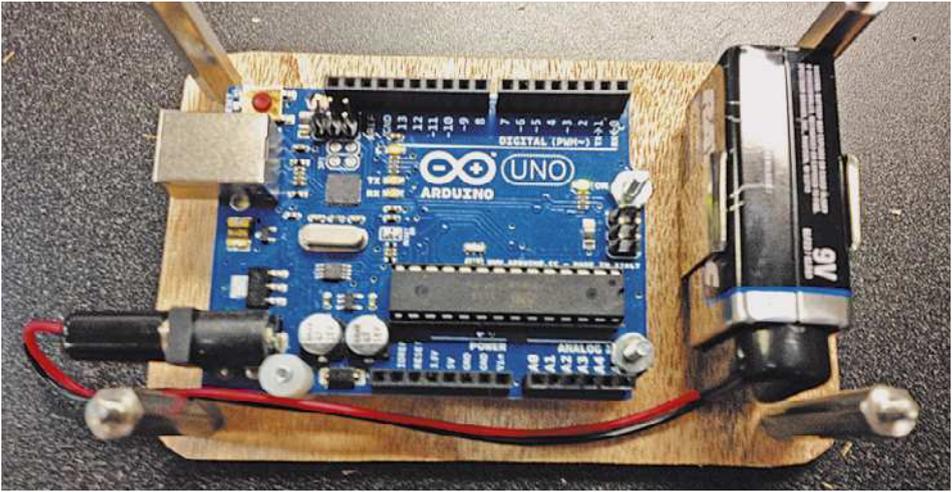


Рис. 10.12. Устанавливаем Arduino, батарейный блок и втулки

3. Прикрепите Arduino к нижней панели и зажим для батарейки с помощью болтов 6 мм. Можно также закрепить алюминиевые втулки, используя дополнительные болты 6 мм. Нижний модуль должен выглядеть так, как на **рис. 10.12**.
4. На верхней панели просверлите отверстия для динамика (сверло 20 мм), втулок (4 мм), потенциометра (9 мм), датчика освещенности (4 мм) и переключателя (6 мм). В результате панель должна выглядеть так, как показано на **рис. 10.13**.



Рис. 10.13. Верхняя панель перед монтажом

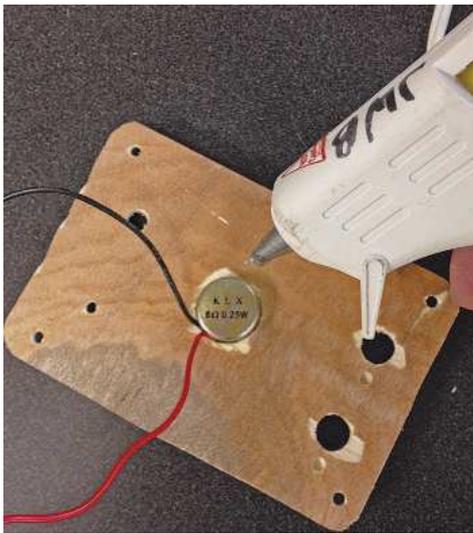


Рис. 10.14. Приклейте термопистолетом корпус динамика к верхней панели

5. Поместите заднюю часть динамика в отверстие диаметром 20 мм и зафиксируйте его с помощью термопистолета так, как показано на **рис. 10.14**. Убедитесь, что провода находятся под верхней панелью так, чтобы они могли дотянуться до Arduino. Обратите внимание, что если вы выберете другую модель динамика (большую по размерам), вам придется сверлить соответствующее отверстие.
6. Давайте займемся разводкой проводов! Припаяйте провода к потенциометрам так, как показано на **рис. 10.15** (вы уже делали это в **главе 3** «Работа с паяльником»). Не нарежьте слишком длинные провода. Чтобы сделать монтаж, достаточно длины 12–13 см. Сделайте то же самое с переключателем, а затем прикрепите эти детали (два потенциометра и переключатель) к верхней панели.

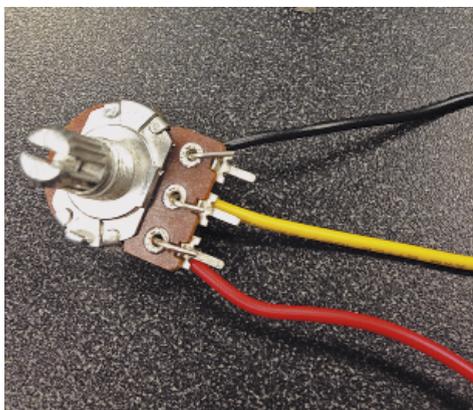


Рис. 10.15. Присоедините провода к потенциометрам так, как показано на рисунке

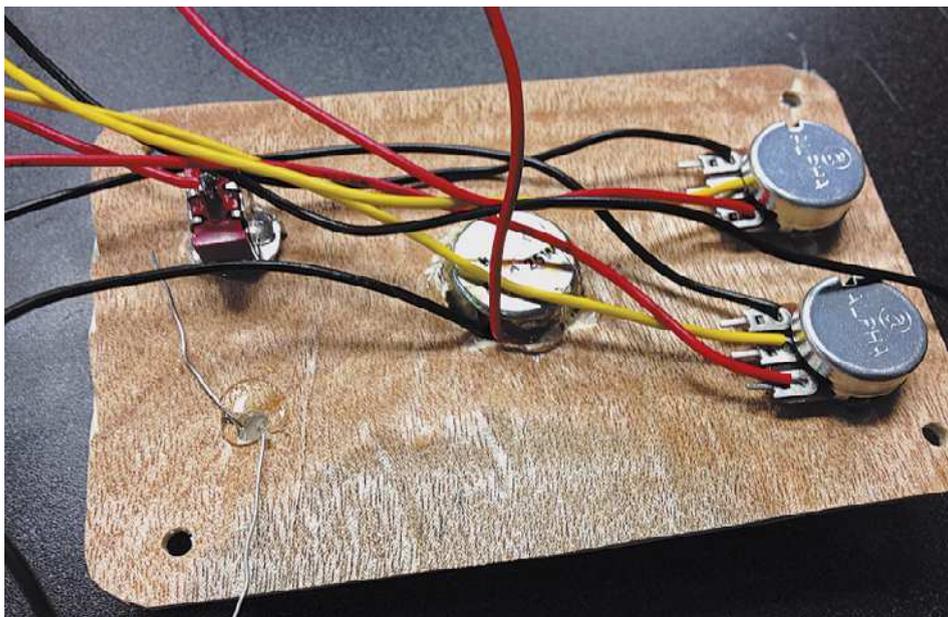


Рис. 10.16. Прикрепите фоторезистор к верхней панели с помощью термоклея.

7. Вставьте сверху датчик освещенности на место и зафиксируйте термоклеем (**рис. 10.16**). Постарайтесь не капать клеем на провода!
8. Соедините резистор 3,3 кОм с выводом «земля» (–) датчика освещенности, затем припаяйте желтый провод ввода данных (**рис. 10.17**). На свободном конце резистора закрепите черный провод, и он станет проводом заземления. Этот провод подключается к пину GND на Arduino.

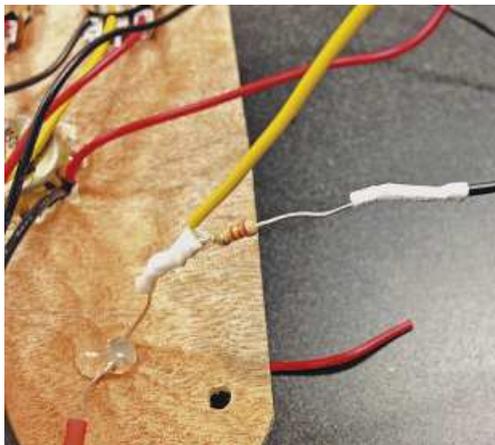


Рис. 10.17. Вывод «земля» (–) фоторезистора приобрел постоянное сопротивление

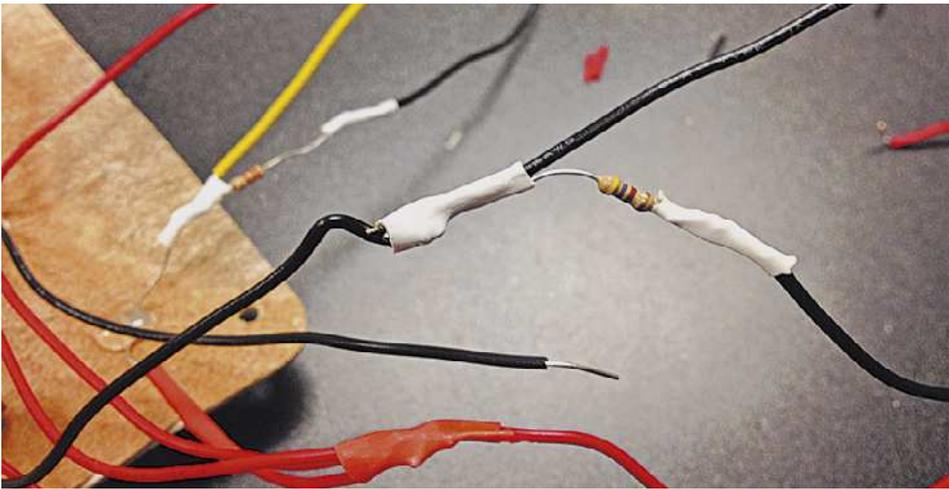


Рис. 10.18. Переключатель также стал с резистором

9. Повторите шаг 8 с переключателем. На один вывод переключателя закрепите резистор 470 Ом, а второй провод вместе с длинным отрезком припаяйте к концу резистора. Конец с напаянным резистором становится землей, в то время как провод без резистора — проводом данных. Эти провода изображены на **рис. 10.18**. Подключите провод заземления к свободному пину GND на Arduino.
10. Припаяйте черные (заземляющие) провода потенциометров к заземляющему (–) выводу динамика. Спаяйте провода с одним длинным по всей длине, а затем скрепите их термоусаживаемой трубкой. Вы объединяете три провода заземления в один (**рис. 10.19**). Подключите эту спайку к пину GND на Arduino.



Рис. 10.19. Объедините заземляющие провода в один



Рис. 10.20. Теперь объедините положительные провода

11. Спаяйте положительные провода потенциометров, датчика освещенности и переключателя вместе, объединив четыре провода в один, так, как вы это делали в шаге 8. Спрячьте сборку в термоусаживаемую трубку. Все должно выглядеть так, как на **рис. 10.20**. Эту сборку вы подключаете к пину 5V на Arduino.
12. Подключите положительный контакт динамика к пину 8, средний провод потенциометров — к пинам A1 и A2, переключатель — к пину 2 и датчик освещенности — к пину A0. Вы сделали это!

Схема должна выглядеть так, как на **рис. 10.21**, за исключением возможного отсутствия макетной платы. Все, что вам нужно сде-

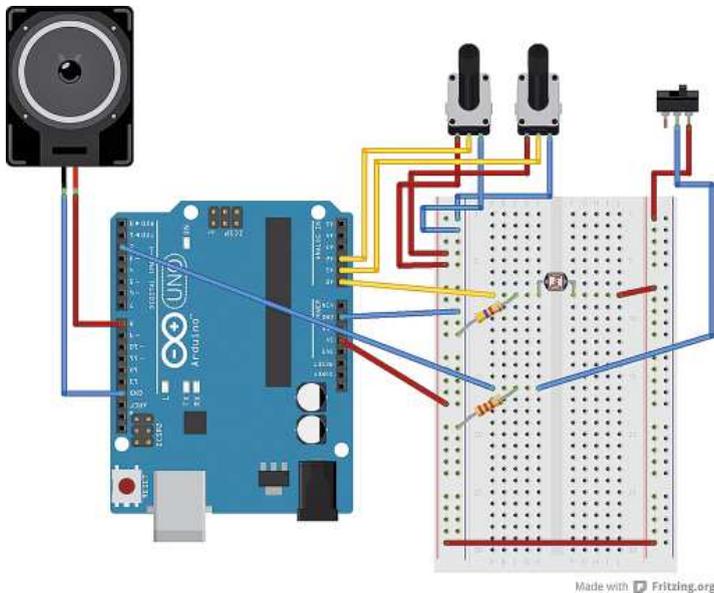


Рис. 10.21. Подключите составные части звукового генератора так, как показано на рисунке

лать, — это подключить батарейку 9 В через батарейную колодку, и вы в шоколаде!

Код звукового генератора

Код звукового генератора выглядит просто.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int sensorReading = analogRead(A0);
  int pot1 = analogRead(A1);
  int pot2 = analogRead(A2);
  int switch1 = digitalRead(2);

  int thisPitch = map(sensorReading, 600, 1000, 1000, 100);
  int potDelay = map(pot1, 0, 1023, 1, 100);
  int potDur = map(pot2, 0, 1023, 1, 50);

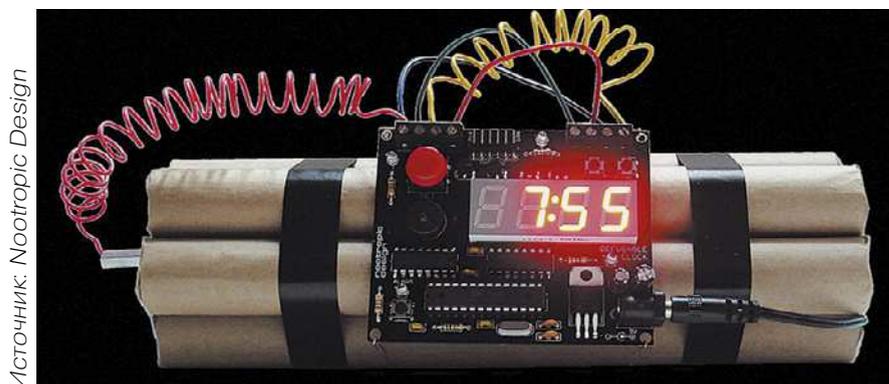
  if (switch1 == HIGH) {
    tone(8, thisPitch, potDur);
    delay(potDelay);
  }
}
```

В следующей главе

В **главе 11** «Отсчет времени» вы узнаете, как Arduino отслеживает время и как помочь ему сделать точный хронометраж. Затем вы сконструируете подвеску «Музыка ветра», которая звонит по часам, а не от дуновения ветра.

Отсчет времени

Как робот определяет время? Может, он смотрит на часы, как большинство из нас? Звучит невероятно, но это так. С помощью Arduino вполне возможно обращаться к «серверу точного времени» в Интернете. Более прозаичный и не очень точный способ — использовать встроенный таймер Arduino, чтобы знать, который час, или использовать выделенный модуль часов реального времени (*real-time clock* — RTC), чтобы отслеживать часы и минуты. В этой главе мы рассмотрим различные способы, с помощью которых Arduino следит за временем. Затем вы постройте управляемые электродвигателем китайские колокольчики «Музыка ветра»¹, которые срабатывают каждый час. На **рис. 11.1** показан пример интерактивных часов на базе Arduino.



Источник: Nootropic Design

Рис. 11.1. «Обезвреживаемые часы» производства компании *Nootropic Design* — интерактивные часы на базе Arduino, которые выглядят, как киношная бомба!

¹ Китайские колокольчики, или «Музыка ветра», — декоративное украшение, выполненное обычно из металлических трубок или бамбука, иногда с колокольчиками, бусинами и другими элементами. Издают мелодичные звуки при порывах легкого ветерка. — *Прим. перев.*

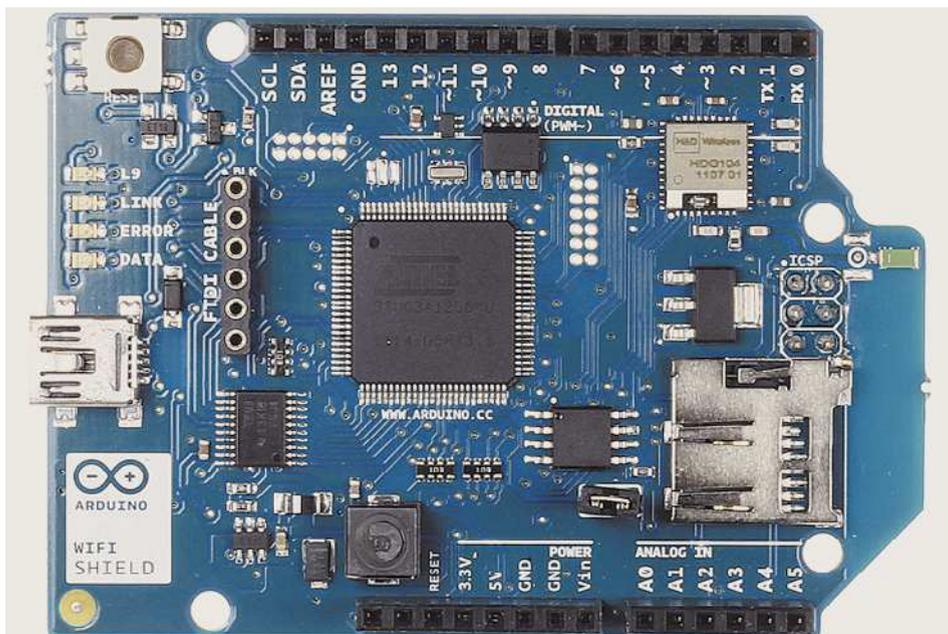
Сервер точного времени

Один из способов отслеживать время в проекте — иметь постоянный доступ к серверу точного времени в Интернете через Wi-Fi, как правило, с помощью платы расширения Arduino Wi-Fi (рис. 11.2). Эти сайты, называемые серверами NTP (network time protocol, протокол сетевого времени), обеспечивают точное время подключенных к Интернету устройств. Если у вас есть смартфон, вы, вероятно, заметили, что на нем не нужно устанавливать время, устройство определяет его автоматически. И все благодаря NTP-серверу!

ДЛЯ СПРАВКИ

ПОДРОБНЕЕ О ДОСТУПЕ К СЕРВЕРУ NTP

На сайте Arduino <http://arduino.cc/en/Tutorial/UdpNTPClient> вы найдете информацию о том, как получить доступ к NTP-серверу с помощью Arduino и платы расширения Arduino Wi-Fi.



Источник: Arduino.cc

Рис. 11.2. С помощью платы расширения Arduino Wi-Fi ваш робот может подключаться к беспроводной сети

Таймер Arduino

Главный чип Arduino, ATmega328P (см. **рис. 11.3**), содержит реле времени, которое относительно неплохо синхронизирует работу устройства. Так же как с помощью команды `delay(1000)` вы сообщаете Arduino подождать 1000 мс, так и таймер, встроенный в ATmega, сообщит, что время истекло.

Из-за скромной архитектуры платы таймер Arduino может следить за временем только 49 дней, поэтому необходим перезапуск, прежде чем он исчерпает лимит отсчета. Кроме того, точность измерения времени не является абсолютной. Блюстители точности огорчатся, узнав, что ATmega теряет около 2 с в сутки. В принципе, после того как таймер достигает 49-дневной отметки, измерение времени уже в высшей степени далеко от истины — ошибка составляет примерно 100 с. Поэтому большинство любителей электроники используют RTC, чтобы знать точное время.

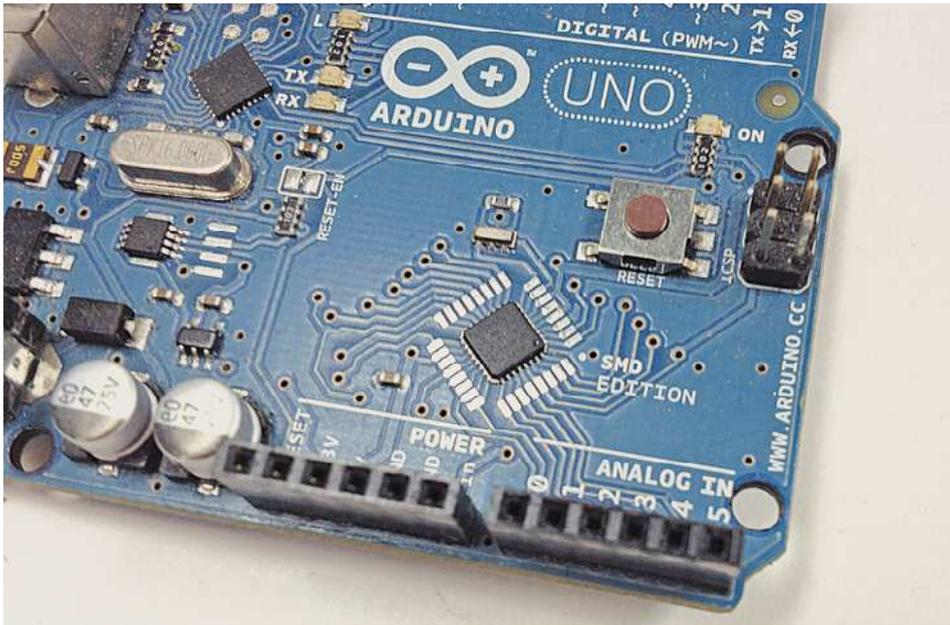


Рис. 11.3. Чип микроконтроллера Arduino имеет функцию синхронизации, которую можно использовать, чтобы корректировать время

Модуль часов реального времени (RTC)

Другой вариант отслеживания времени — подключение модуля часов реального времени (RTC), например «ChronoDot», который изображен на **рис. 11.4**. RTC состоит из печатной платы с чипом таймера высокой точности, а также миниатюрного аккумулятора (таблетки), который поддерживает установленное время, даже если плата отключена. При правильной настройке «ChronoDot» отклоняется менее чем на минуту за год благодаря наличию термостата, а надежный аккумулятор позволит отслеживать время в течение восьми лет.

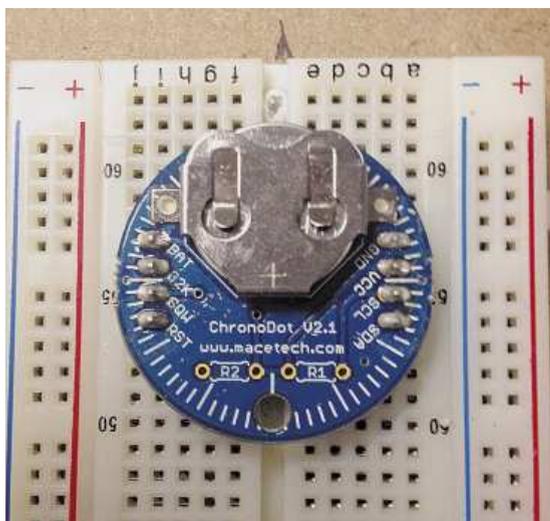


Рис. 11.4. Модуль RTC «ChronoDot», подключенный к макетной плате

Проект: цифровые часы

В этом проекте вы сделаете цифровые часы (**рис. 11.5**), обладающие прекрасной точностью отсчета благодаря модулю часов реального времени, смонтированному на небольшой плате, которая имеет чип таймера и резервный аккумулятор, обеспечивающий надежность. Выглядят часы не очень красиво, но вы можете поместить их в декоративную коробку.

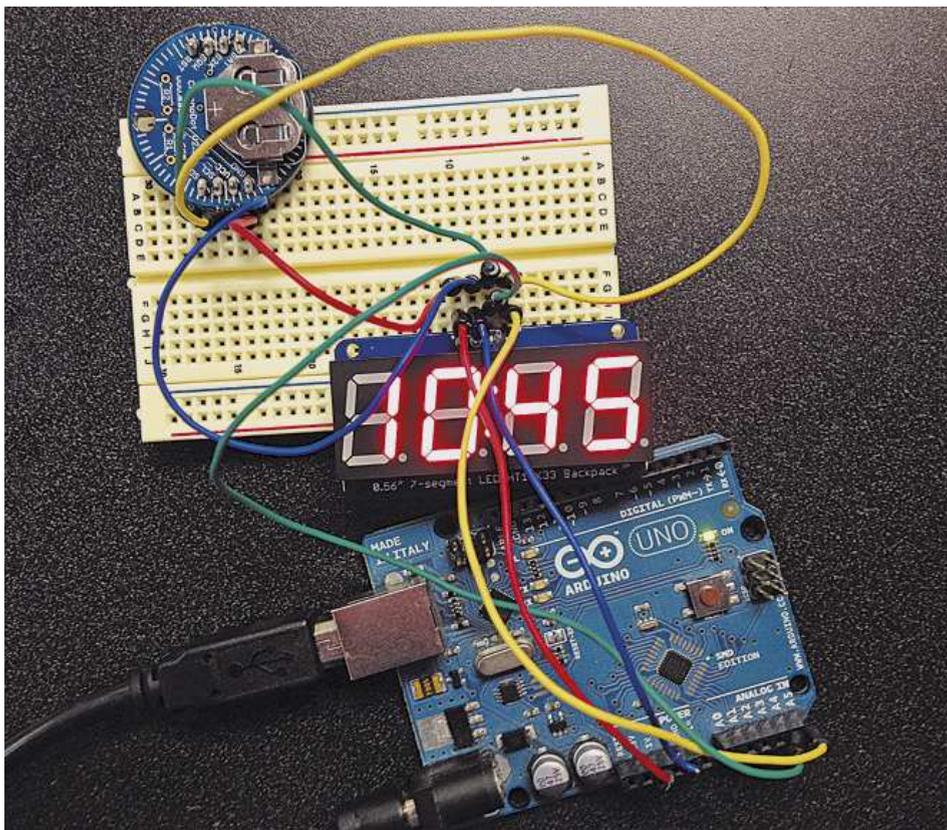


Рис. 11.5. Arduino, семисегментный индикатор и модуль RTC — это все, что нужно для создания часов!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для создания цифровых часов нужно совсем немного комплектующих.

- Arduino UNO с источником питания.
- Модуль RTC. Я использовал RTC «ChronoDot», но можно использовать и более дешевый набор коммутируемой платы RTC DS1307.
- Семисегментный дисплей компании *Adafruit*. Это бесценная панель состоит из семисегментного дисплея с печатной платой, специально разработанной, чтобы сделать ее более удобной для использования с макетной платой.
- Макетная плата половинного размера.
- Соединительные провода.

Инструкции

Давайте подключим цифровые часы, как показано на **рис. 11.6**. Обратите внимание, что изображенный на **рис. 11.6**, RTC — это модель DS1307, которую я упоминал в этой главе. По техническим характеристикам этот модуль похож на «ChronoDot», оба модуля используют одну и ту же библиотеку Arduino, поэтому не имеет значения, какой из двух модулей вы будете использовать в проекте. Давайте начнем!

1. Подключите семисегментный дисплей к макетной плате, убедитесь, что оставили достаточно места для соединительных проводов.
2. Прикрепите модуль RTC, также оставив место для проводов.
3. Подключите платы. Это непросто, потому что семисегментный дисплей и модуль RTC должны быть подключены к четырем одинаковым пинам на Arduino!
 - Подключите пин «+» на дисплее к пину 5V на RTC (на «ChronoDot» этот пин обозначен как «VCC») и к пину 5V на Arduino. Это красный провод на **рис. 11.6**.
 - Подключите пин «-» на дисплее к пину GND на RTC и затем к пину GND на Arduino. Это черный провод на **рис. 11.6**.
 - Подключите пин «С» (часы) на дисплее к пину SCL на RTC и затем к пину A5 (это не цифровой, а аналоговый пин!) на Arduino. Это зеленый провод на **рис. 11.6**.

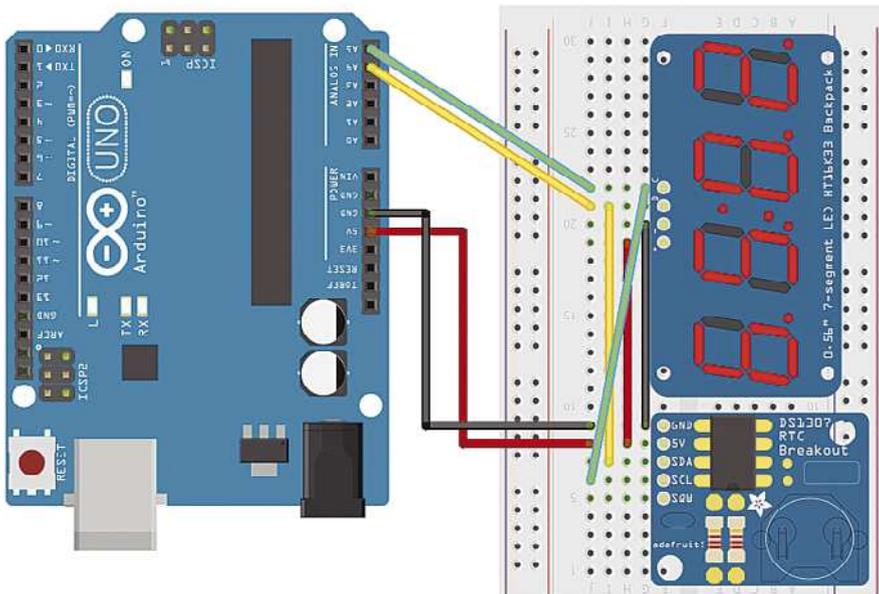


Рис. 11.6. Подключите часы, как показано на рисунке

- Подключите пин «D» (данные) на дисплее к пину SDA на RTC и затем к пину A4 на Arduino. Это желтый провод на **рис. 11.6**.

Вы справились с оборудованием! Теперь перейдем к программированию Arduino.

Код для цифровых часов

Теперь вы можете загрузить скетч на Arduino. Как и в предыдущих главах, вы можете загрузить скетч с сайта <https://github.com/n1/Arduino-For-Beginners>.

```
// этот код основан на примере текста Adafruit для модуля RTC

#include <Wire.h>
#include "Adafruit_LEDBackpack.h"
#include "Adafruit_GFX.h"
#include "RTClib.h"

RTC_DS1307 RTC;
Adafruit_7segment disp = Adafruit_7segment();

void setup()
{
  Wire.begin();
  RTC.begin();
  if (! RTC.isrunning())
  {
    RTC.adjust(DateTime( DATE , TIME ));
  }
  disp.begin(0x70);
}

void loop()
{
  disp.print(getDecimalTime());
  disp.drawColon(true);
  disp.writeDisplay();
  delay(500);
}

int getDecimalTime()
{
  DateTime now = RTC.now();
  int decimalTime = now.hour() * 100 + now.minute();
  return decimalTime;
}
```

🔧 Проект: китайские колокольчики «Музыка ветра»

В этой главе вы сделаете китайские колокольчики (см. **рис. 11.7**), которые звонят по времени благодаря модулю часов реального времени. Также вы изготовите корпус геометрической формы для размещения электроники с помощью инструмента под названием «фрезерный станок с ЧПУ».



Рис. 11.7. Вы научитесь делать китайские колокольчики!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

- Arduino.
- Сервопривод (я использовал модель HiTec HS-322HD).
- Сервонасадки (существует огромное количество насадок).
- Модуль RTC «ChronoDot».
- Мини-макетная плата.
- Батарейка 9 В и зажим для нее.
- Круглый стержень диаметром 6 мм и длиной 20 см.

- Китайские колокольчики «Музыка ветра» (я использовал модель «Gregorian Chimes Soprano»).
- Фанера толщиной 5 мм для корпуса.
- Сосновые бруски для опорных блоков 25 мм.
- Крюк и гайка к нему.
- Саморезы для дерева № 8 × 12 мм .
- Саморезы для дерева № 6 × 50 мм.
- Саморезы для дерева № 4 × 12 мм.
- Болты 6 × 2 мм с фиксирующими шайбами и гайками, 24 шт.
- Болты 6 × 25 мм с фиксирующими шайбами и гайками, 12 шт.
- Дрель и разные сверла.
- Отрезная дисковая пила.
- Циркулярная пила.
- Фреза для прорезки отверстий.
- Ленточно-шлифовальный станок.

Сервонасадки

Сервоприводы подключаются к колесам, осям и другим частям робота с помощью съемных насадок, которые называются сервонасадками (*servo horn*). Они представляют собой бруски или диски (см. **рис. 11.8**) с просверленными резьбовыми отверстиями и имеют зубчатое углубление под наконечник, который соответствует ступицам сервопривода. Сервонасадка крепится к оси сервопривода винтом.

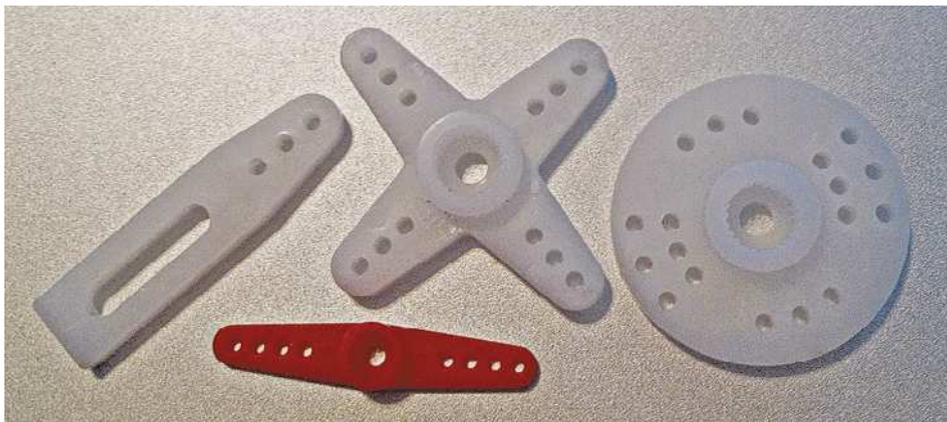


Рис. 11.8. Сервонасадки помогают прикрепить робота к сервоприводу для того, чтобы он двигался

Инструкции

Выполните следующие действия, чтобы сделать китайские колокольчики.

1. Вырежьте на станке треугольники для изготовления корпуса. Я воспользовался фрезерным мини-станком с ЧПУ (см. **рис. 11.9**). Если собираетесь делать это самостоятельно (без ЧПУ), надо вырезать равносторонние треугольники со стороной 8 см и просверлить отверстия 6 мм в трех углах. Файл, который я использовал для вырезания, доступен для скачивания по ссылке: <https://github.com/n1/Arduino-For-Beginners>.



Рис. 11.9. Фрезерный мини-станок с ЧПУ вырезает формы, которые понадобятся для этого проекта

2. Обработайте треугольники на ленточно-шлифовальном станке (см. **рис. 11.10**), чтобы сделать боковины гладкими.



Рис. 11.10. Уберите неровности на шлифовальном станке

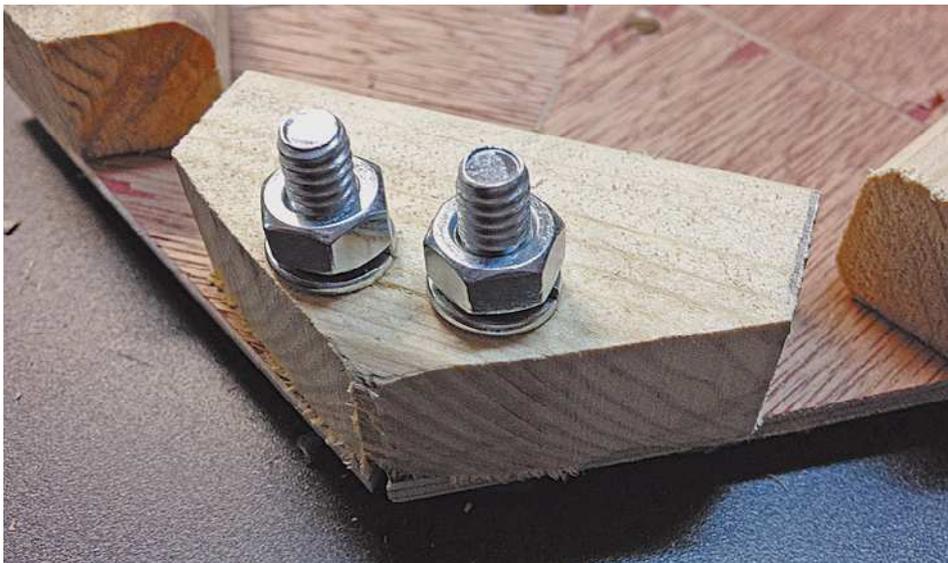


Рис. 11.11. Скрепленные бруски удерживают треугольные элементы, из которых сложены крышка и дно конструкции

3. Нарежьте опорные элементы из сосновых брусков толщиной 2,5 см. Постарайтесь, чтобы они выглядели так, как на **рис. 11.11**. Эти бруски усиливают верхнюю часть корпуса и служат опорой для боковых сторон. Подготовьте 12 блоков — они понадобятся и для нижней части.
4. Вырежьте закрепляющий диск из фанеры толщиной 5 мм с помощью циркулярной пилы. Все должно выглядеть так, как на **рис. 11.12**.



Рис. 11.12. Этот скрепляющий диск удерживает части крыши корпуса вместе



Рис. 11.13. Сделайте гексагональный корпус для китайских колокольчиков!

5. Просверлите шесть отверстий по 6 мм вокруг отверстия диаметром 15 мм. Убедитесь, что вы совместили эти отверстия с шестью отверстиями в центре верхней части корпуса.
6. Скрепите верхнюю часть корпуса соответствующими болтами. Сначала прикрепите треугольники к опорным элементам, а затем добавьте закрепляющий диск и крюк в центре. Все должно выглядеть так, как на **рис. 11.13**. Вполне возможно, что вам придется рассверлить центральное отверстие, чтобы треугольники встали на место.
7. Постройте нижнюю часть корпуса аналогично верхней, но с парой различий. Одно из них заключается в том, что вам не понадобится крюк. Вместо него в нижней части корпуса будет выступать дюбель. Второе различие — это то, что вместо центрального опорного диска вы воспользуетесь диском от китайских колокольчиков. Просверлите в нем семь отверстий так же, как вы сделали это на верхнем диске. (Обратите внимание, что на **рис. 11.14**, только три отверстия закреплены болтами. Ох уж эти болты!)
8. С помощью фрезы для прорезки отверстий вырежьте из фанеры круг диаметром 5 см и просверлите в центре отверстие диаметром 6 мм. Прикрутите сервонасадку к диску, используя саморезы для дерева № 4 × 12 мм, приклейте круглый стержень к отверстию в диске. Все должно выглядеть так, как на **рис. 11.15**. Как и диск, вся конструкция выглядит не очень привлекательно, но это только промежуточный шаг перед прикреплением сервонасадки к круглому стержню.



Рис. 11.14. Просверлите семь отверстий в верхней части китайских колокольчиков и прикрепите их к днищу корпуса



Рис. 11.15. Это и не должно выглядеть красиво

9. Вырежьте из фанеры боковые стороны корпуса. Они должны быть примерно 10 см высотой. Для того чтобы прикрепить панели к верхней и нижней частям корпуса, надо просверлить отверстия (**рис. 11.16**). Расположить эти отверстия совсем несложно, поскольку они должны прикрепить опорные бруски. Если хотите, в это же время можно прикрепить боковые панели к верхней и нижней частям корпуса.



Рис. 11.16. Боковая панель готова к монтажу

10. Прикрепите двигатель к верхней панели с помощью деревянных планок, нарезанных из фанеры и кубиков размером 5 см. С помощью саморезов для дерева № 8 × 12 мм прикрепите сервопривод к рейкам. Затем с помощью саморезов № 6 × 50 мм прикрепите рейки к опорным брускам. На самом деле ваша задача заключается в том, чтобы деревянные рейки были такой высоты, чтобы сервопривод не упирался в верхнюю панель и узел сервопривода совпал с круглым стержнем в нижней части корпуса. Все это показано на **рис. 11.17**.

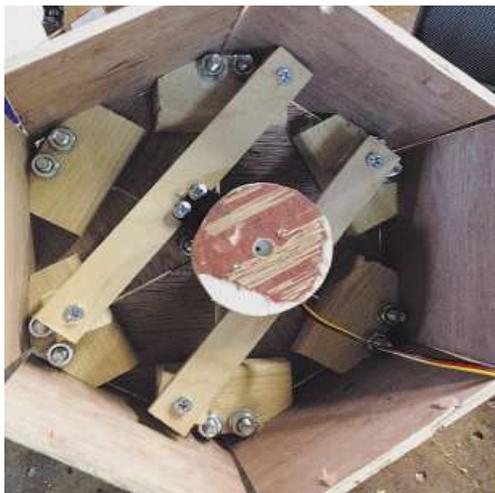


Рис. 11.17. Вид внутренней части корпуса снизу. Вы видите сервопривод, установленный в центре

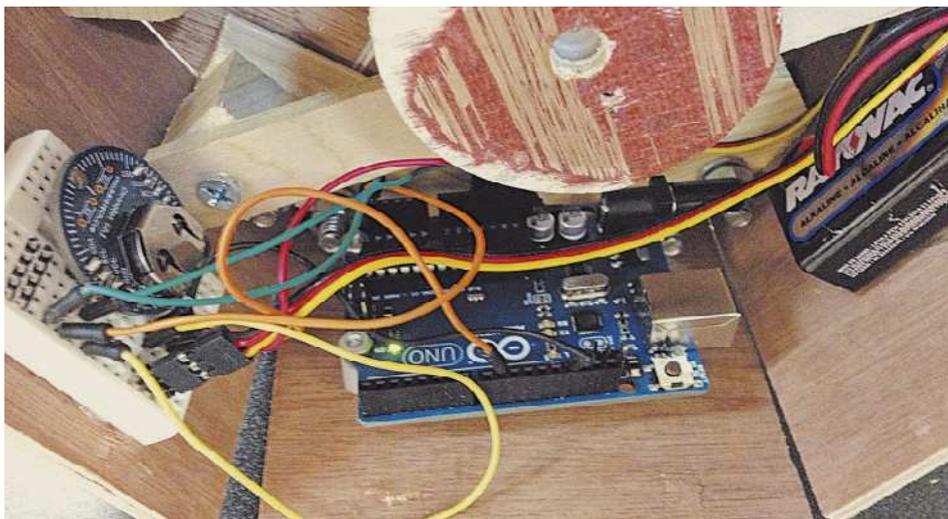


Рис. 11.18. Прикрепите электронику внутри корпуса

11. Прикрепите Arduino, макетную плату и комплект батареек внутри корпуса. Я предлагаю прикрутить Arduino к боковой панели с помощью крепежного винта, а макетную плату и батарейку приклеить термоклеем. Все это должно выглядеть примерно так, как на **рис. 11.18**.
12. Подключите компоненты так, как показано на **рис. 11.19**.

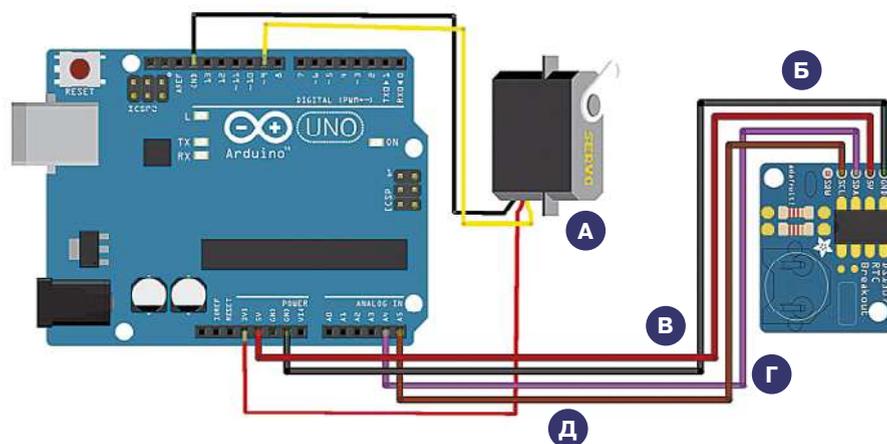


Рис. 11.19. Соедините компоненты, как показано на рисунке:

А — подключите сервопривод; желтый провод идет к пину 9 на Arduino, красный провод идет к пину 3.3V на Arduino, черный провод подключается к одному из пинов GND Arduino; Б — подключите пин GND модуля RTC к пину GND на Arduino; В — подключите пин 5V модуля RTC к пину 5V на Arduino; Г — подключите пин SDA (данные) модуля RTC к пину A4 на Arduino; Д — подключите пин SCL (часы) модуля RTC к пину A5 на Arduino



Рис. 11.20. Язык для колокольчиков скоро будет работать, как куранты

13. Просверлите отверстие в языке китайских колокольчиков (**рис. 11.20**). Язык будет вращаться с помощью сервопривода, и колокольчики будут звенеть. Однако пока не приклеивайте язык к круглому стержню!
14. Привинтите боковые панели к днищу шестиугольника. В завершение работы приклейте язык для колокольчиков на конце круглого стержня (**рис. 11.21**).



Рис. 11.21. Работа закончена, и китайские колокольчики готовы зазвенеть!

Код

Код достаточно простой и состоит только из функции определения времени с помощью модуля RTC и поворота сервопривода, когда минуты читаются как «ноль».

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#include <Wire.h>
#include "RTClib.h"
#include <Servo.h>

Servo myservo;
RTC_Millis RTC;
int pos = 0;

void setup()
{
  Serial.begin(57600);

  RTC.begin(DateTime( DATE , TIME ));
  myservo.attach(9); // подключаем сервопривод к пину 9
}

void loop() {

  DateTime now = RTC.now();

  int decimalTime = now.hour() * 100 + now.minute();
  Serial.print(decimalTime);
  delay (60000);

  Serial.println();

  if (now.minute() == 0) {
    for(pos = 0; pos < 180; pos += 1)
    {
      myservo.write(pos); delay(15);
    }
  }
}
```

Станки с числовым программным управлением (ЧПУ)

Для создания деревянных панелей в этом проекте я использовал станок с ЧПУ на примере фрезерного мини-станка с ЧПУ. Такой станок управляется компьютером, чтобы вращать фрезерный инструмент для шлифования, сверления и формования фигур из дерева, металла и других материалов. На **рис. 11.22** показан фрезерный станок с ЧПУ.

Вот как работает фрезерный станок с ЧПУ.

1. С помощью векторного редактора, например *Adobe Illustrator* или *CorelDRAW* разработайте дизайн изделия.
2. С помощью утилиты ЧПУ подготовьте эскиз изделия для обработки на станке. Один из примеров такой программы — «Vectric CUT2D» (vectric.com), которая пошагово проведет вас через процесс разработки. Программа поможет определить, как и в каком порядке будет обработана каждая деталь. Например, ваш эскиз включает окружность диаметром 2,5 см. В «CUT2D» вы можете задать команду фрезе двигаться по окружности.



Рис. 11.22. Фрезерный станок с ЧПУ — это ценный инструмент для дерева и металла

ДЛЯ СПРАВКИ

МОЖНО ВЫРЕЗАТЬ ДАЖЕ 3D-ФОРМУ

Имейте в виду, что станок с ЧПУ может работать с толстыми блоками материала, вырезая 3D-форму объектов.

3. Поместите заготовку на стол станка с ЧПУ и зафиксируйте, чтобы заготовка не смещалась во время работы фрезы. Убедитесь, что вы приготовили достаточно материала вокруг предполагаемой формы резки, чтобы не останавливать процесс обработки.
4. Загрузите файл с заготовкой на рабочий верстак станка. Возможно, вам захочется провести «сухую прогонку». Это означает, что фреза пройдет в нескольких сантиметрах над поверхностью материала, чтобы вы могли проконтролировать процесс. Вы можете посмотреть, не выходит ли край фрезы во время работы за границы заготовки, а значит, можете исправить неполадки до начала основного процесса.
5. И наконец, самое важное. Работа на станке с ЧПУ может быть опасной, потому что во время работы фрезы от заготовки летят во все стороны опилки и щепки. Наденьте защитные очки! Кроме этого, станок с ЧПУ достаточно шумный, поэтому также не забудьте надеть наушники.

В следующей главе

В **главе 12** «Безопасная работа с высоким напряжением» вы научитесь использовать электричество из розетки (без риска!), а также узнаете, как создать лавовую лампу, которой можно управлять с помощью Arduino.

Безопасная работа с высоким напряжением

Мы подсознательно боимся высокого напряжения, и правильно делаем! Трудно убить себя пальчиковой батарейкой, но, уверяю вас, совать палец в розетку не стоит. В этой главе вы ознакомитесь с некоторыми приемами безопасного использования высокого напряжения в своих проектах. Затем вы сконструируете лавовую лампу, которой можно управлять с помощью Arduino. Она будет включаться и выключаться по расписанию или по команде с пульта дистанционного управления (см. **рис. 12.1**). Именно этого не хватает лавовой лампе!



Рис. 12.1. Лавовая лампа «Бадди» управляет вашим любимым светильником с пузырями

Урок: управление высоким напряжением

Секрет безопасной работы под высоким напряжением очень прост — нужно просто подальше держаться от него! Я шучу, но на самом деле это правильный совет! Вместо этого давайте поручим всю опасную работу умной электронной детали, которая называется «реле» (см. **рис. 12.2**). Я уже упоминал о реле в **главе 1** «Знакомство с Arduino». По сути, реле коммутирует те процессы, которые запускает Arduino. Реле управляют напряжением, поэтому вам не нужно иметь с ним дело. Конечно, реле — это просто электронные компоненты, им необходима структура, например печатная плата. В следующих разделах мы рассмотрим три изделия, которые функционируют как реле. Вы можете использовать их в дальнейшей работе.



Рис. 12.2. Реле как билет-допуск к управлению высоким напряжением

Шнур питания с реле-переключателем

Шнур питания с реле-переключателем выглядит, как короткий удлинитель с встроенным блоком питания (см. **рис. 12.3**). Так и есть, за исключением того, что коробка блока питания имеет порты для подключения проводов, что позволяет подать напряжение по одному проводу от пина на Arduino. Шнур питания с реле-переключателем также имеет защитное заземление.



Рис. 12.3. Шнур питания с реле-переключателем в сущности представляет собой короткий провод со встроенной релейной платой

Что действительно отличает шнур питания с реле-переключателем от конкурентов, так это надежность. Вам нужно включить прибор в электрическую розетку? Тогда используйте шнур питания с реле-переключателем.

Простая плата расширения с реле EMSL

Простая плата расширения с реле EMSL (см. **рис. 12.4**) имеет совершенно иную конфигурацию, чем шнур питания с реле-переключателем. Данная плата была разработана умельцами из *Evil Mad Scientist Laboratories*, или сокращенно EMSL. Она работает как плата расширения для Arduino. Это означает, что контакты платы подходят к штифтам Arduino. Сама плата расширения тоже имеет штифты, позволяя пользователю не только управлять реле, но и наблюдать за работой датчиков или за миганием светодиодов при работе с Arduino.

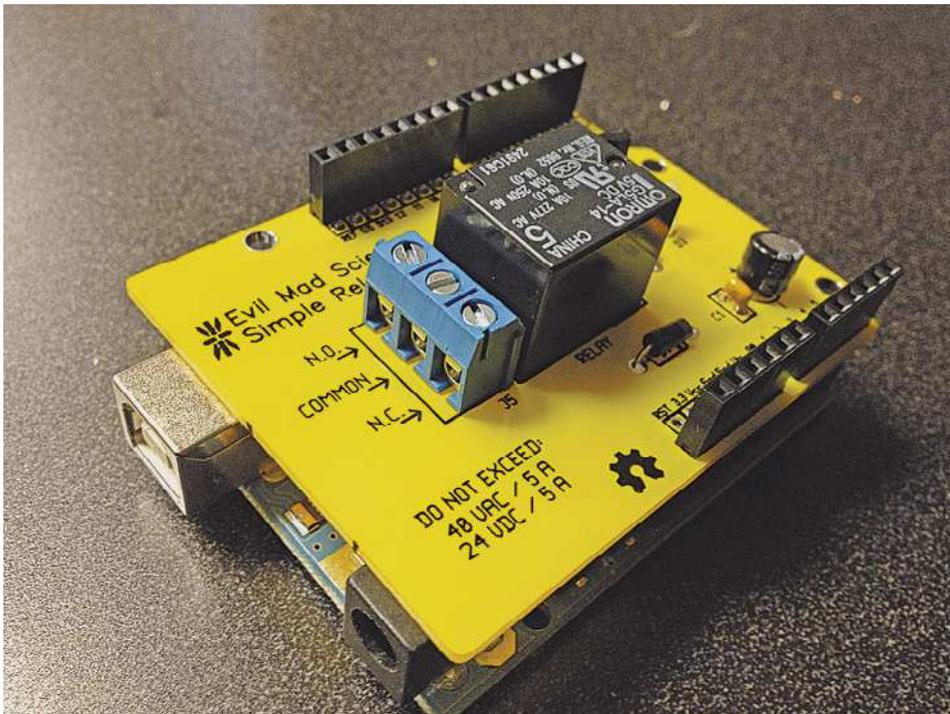


Рис. 12.4. С помощью платы расширения можно добавить реле к вашему Arduino. Вы можете видеть край Arduino, выступающий из-под платы расширения с реле

У этой платы есть несколько недостатков.

- Вы не можете использовать ее, чтобы переключать стандартный переменный ток 110 В¹, с которым справляется шнур питания с реле-переключателем. Ее максимальное напряжение — 40 В/5 А переменного тока или 24 В/5 А постоянного тока.
- Если вы подключаете провода высокого напряжения к плате вручную, то подвергаетесь потенциальной опасности.

В то же время эта плата стоит вдвое дешевле, чем шнур питания с реле-переключателем.

Плата управления реле «Beefcake»

Плата управления реле «Beefcake» от компании *SparkFun* стоит недорого, и проста в использовании. Эта плата имеет мощнейшее реле, позволяющее коммутировать напряжение до 220 В и 20 А (рис. 12.5). Однако ввиду отсутствия некоторых функций шнур питания с реле-переключателем и простая плата расширения с реле более предпочтительны.

- В отличие от простой платы расширения с реле «Beefcake», конструкция не позволяет подключать плату напрямую к Arduino.
- В отличие от шнура питания с реле-переключателем плата управления реле «Beefcake» неудобна в использовании и небезопасна в эксплуатации. Эта плата рекомендуется только для тех пользователей, которые имеют опыт работы с высоким напряжением.



Рис. 12.5. Плата управления реле «Beefcake» от компании *SparkFun* может коммутировать напряжение до 220 В/20 А

¹ В России стандарт 220 В. — Прим. перев.

ТЕХНИКА БЕЗОПАСНОСТИ

Как известно каждому, электричество опасно для жизни человека. Повреждение электрического оборудования может вызвать пожары, которые уничтожают имущество и наносят вред здоровью людей. Таким образом, очень важно, чтобы вы обращались с высоким напряжением крайне осторожно, или, еще лучше, не трогали его совсем! Знак, изображенный на **рис. 12.6**, предупреждает вас о потенциальной опасности электрического тока. Только не думайте, что этот знак всегда будет указывать зону возможного поражения электрическим током.



Рис. 12.6. Когда вы видите этот знак, знайте, что поблизости находится опасное высокое напряжение

Всегда соблюдайте следующие меры безопасности.

- Избегайте контакта с оголенными проводами и незащищенными разъемами, в том числе с платами якобы низкого напряжения. Все оборудование более 50 В должно иметь изоляцию или другую защиту для предотвращения случайного прикосновения.
- Обращайтесь с электрическими устройствами так, как если бы они были включены или находились под напряжением. Проверьте с помощью вольтметра, если вы не уверены в этом. Кроме того, следует знать, что некоторые детали, такие как конденсаторы, сохраняют заряд в течение длительного времени после того, как деталь отключена от источника питания.
- Отключите источник питания, прежде чем работать с любой частью оборудования.
- Избегайте токопроводящих инструментов, украшений и других предметов, которые могут проводить электричество через тело.
- Не используйте поврежденное или неправильно присоединенное электрооборудование, в том числе силовые кабели.
- Не используйте мокрое или влажное электрооборудование. Отключите оборудование и дайте ему высохнуть, прежде чем работать на нем. Даже водный конденсат может стать причиной смертельного удара током!

Не пытайтесь трогать, ремонтировать или даже приступать к проекту, связанному с высоким напряжением, если только вы **ДЕЙСТВИТЕЛЬНО** не уверены в том, что делаете!

🔧 Проект: контроллер для вентилятора

В рамках этого проекта вы сделаете контроллер для вентилятора, который включается, когда температура достигает определенного уровня (см. **рис. 12.7**). Контроллер состоит из Arduino UNO, датчика температуры и шнура питания с реле-переключателем. Все это затем подключается к обычному комнатному вентилятору.

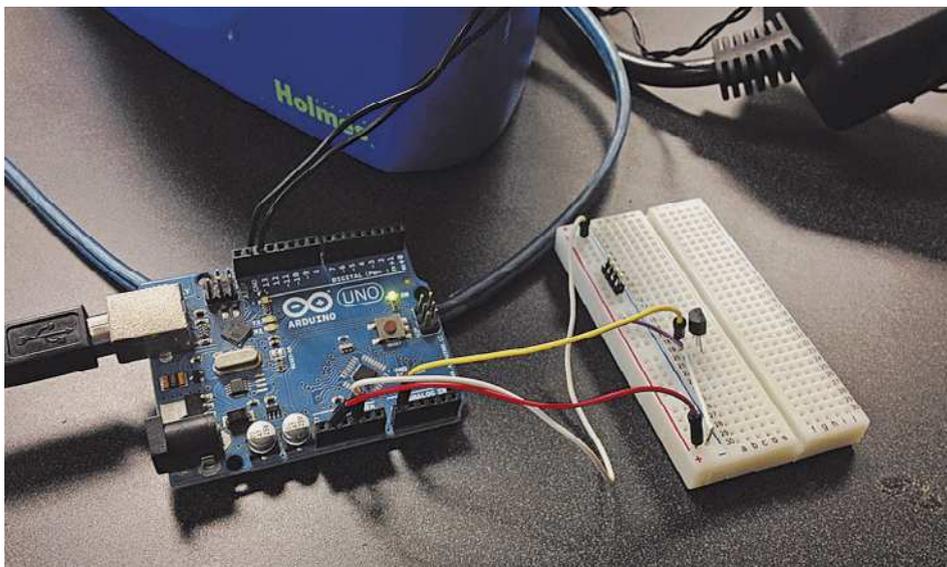


Рис. 12.7. Вентилятор включится, когда температура достигнет определенного значения

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для этого проекта вам нужно совсем немного комплектующих.

- Arduino UNO и блок питания.
- Шнур питания с реле-переключателем.
- Датчик температуры «LM355AZ».
- Макетная плата.
- Соединительные провода.
- Вентилятор, работающий от 110 В¹.

¹ В нашей сети 220 В. — Прим. перев.

Инструкции

Эта сборка проводится на скорую руку. Для нее необходимо только пять проводов и один электронный элемент (см. **рис. 12.8**).

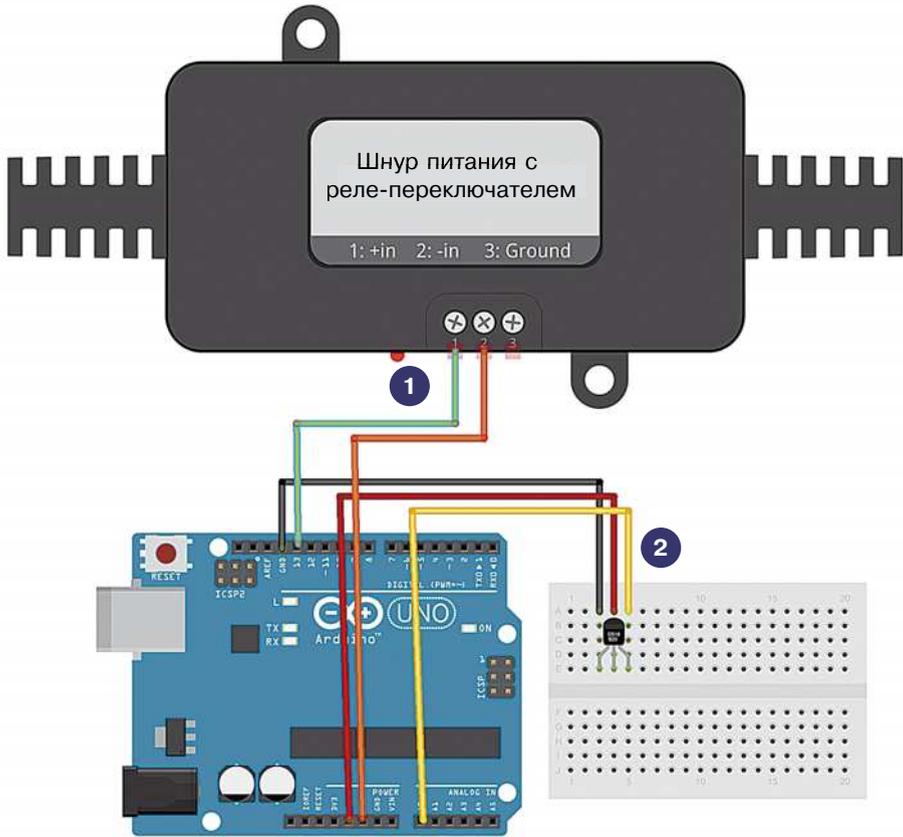


Рис. 12.8. Контроллер вентилятора собирается быстро и легко

1. Подключите первую и вторую клеммы шнура питания с реле-переключателем к пинам 13 и GND на Arduino. На **рис. 12.8** эти соединения показаны соответственно зеленым и оранжевым проводами.
2. Глядя на плоскую поверхность датчика температуры, подключите левый провод к GND (черный провод), центральный провод к 5V (красный провод) и правый провод (желтый) к A0.

В конце вставьте вилку шнура питания с реле-переключателем в электрическую розетку, а шнуровую розетку подключите к вентилятору.

Код для контроллера вентилятора

Загрузите следующий скетч на Arduino. Если вы забыли, как это сделать, обратитесь к **главе 5** «Программирование Arduino».

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
int sensorPin = A0; // подключите сюда датчик
int fanPin = 13;    // подключите сюда шнур питания
                    // с реле-переключателем
int sensorValue = 0; // переменная для хранения значения,
                    // поступающего от датчика

void setup() {
  pinMode(fanPin, OUTPUT); Serial.begin(9600); Serial.
  println("starting!");
}

void loop() {
  sensorValue = analogRead(sensorPin);

  Serial.print(sensorValue);

  // измените значение sensorValue в зависимости от температуры,
  // при которой вентилятор должен начать работу
  if (sensorValue >= 753)
  {
    digitalWrite(fanPin, HIGH);
    delay(10000); // продолжительность работы вентилятора
                  // в миллисекундах
  }
  else
  {
    digitalWrite(fanPin, LOW);
    delay(10000); // время в миллисекундах, оставшееся до очередной
                  // проверки датчика
  }
}
```

⚙ Проект: лавовая лампа «Бадди»

Все любят лавовые лампы — те самые уютно светящиеся лампы конусообразной формы, с пузырями внутри. На самом деле такая лампа очень просто устроена. В основании спрятана лампочка; при загорании она нагревает сосуд с воском и жидкостью. Когда воск нагревается, он всплывает, а охлаждаясь, снова опускается.

Недостаток лавовой лампы в том, что ей нужно время для разогрева. Если вы захотели полюбоваться лампой и щелкнули выключателем, вам понадобится еще час, чтобы лампа приобрела привлекательный вид. Было бы здорово составить расписание для лампы, чтобы она включалась за час до вашего возвращения домой! Кроме того, многие производители не рекомендуют держать лампу включенной более 10 ч, поэтому хорошо бы, чтобы лампа также автоматически выключалась! Наконец, мы очень ленивы и были бы счастливы, если бы лампу можно было включать и выключать с помощью обычного пульта дистанционного управления.

Все, что вам нужно, — это управляемая Arduino лавовая лампа «Бадди» (рис. 12.9) с запрограммированным расписанием работы и с дистанционным управлением. Вам повезло, это как раз проект этой главы!



Рис. 12.9. Лавовая лампа «Бадди» контролирует свою работу, так что можете отдыхать!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для создания лавовой лампы «Бадди» вам понадобятся следующие компоненты.

- Arduino UNO и блок питания.
- Шнур питания с реле-переключателем.
- Датчик инфракрасного излучения «TSOP38238».
- Пульт дистанционного управления *Sony* (подойдет любой).
- Модуль часов реального времени (RTC) (описан в **главе 11** «Отсчет времени»).
- Соединительные провода.
- Коробка (лучше покрытую алюминиевой фольгой).
- Удлинитель.
- Электрическая дрель.
- Сверло для дрели 6 мм.
- Сверло для дрели 12 мм.
- Клеевой термопистолет.

Перекодирование датчика инфракрасного излучения

Инфракрасные датчики (**рис. 12.10**) предназначены исключительно для определения инфракрасного света, и они очень избирательны. Датчик реагирует только на инфракрасное излучение (ИК), пульсирующее с частотой 38 кГц (это 38 000 импульсов в секунду), соответственно этому датчик переключает выходное напряжение. Если датчик обнаруживает пульсации 38 кГц, он выдает 0 В; в противном случае он выдает 5 В.

С числом 38 кГц возникает встречная проблема: как послать именно такой сигнал? Можно использовать электронные уловки с участием



Источник: Adafruit Industries

Рис. 12.10. Датчик инфракрасного излучения, подобный этому, улавливает сигналы с пульсацией 38 кГц, но игнорирует другое инфракрасное излучение

пульсирующего инфракрасного светодиода, но многие электронщики используют бытовые пульты дистанционного управления. Можно считать ИК-коды каждой кнопки, программист может дать команду Arduino выполнять различные действия для каждого кода. Другими словами, вы могли бы установить ИК-датчик на робота и контролировать его тем же пультом, с помощью которого управляете телевизором.

Инструкции

Вам предстоит справиться с более сложной задачей, чем при выполнении проекта, описанного в этой главе, где также был задействован шнур питания с реле-переключателем. Помимо ИК-датчика необходимо включить также модуль часов реального времени и, разумеется, Arduino. На **рис. 12.11** изображена сборка лавовой лампы «Бадди».

1. Подключите Arduino, шнур питания с реле-переключателем, модуль RTC и ИК-датчик так, как показано на **рис. 12.11**.
 - А. Подключите клемму 1 шнура питания с реле-переключателем к пину 13 на Arduino. Это зеленый провод на **рис. 12.11**.

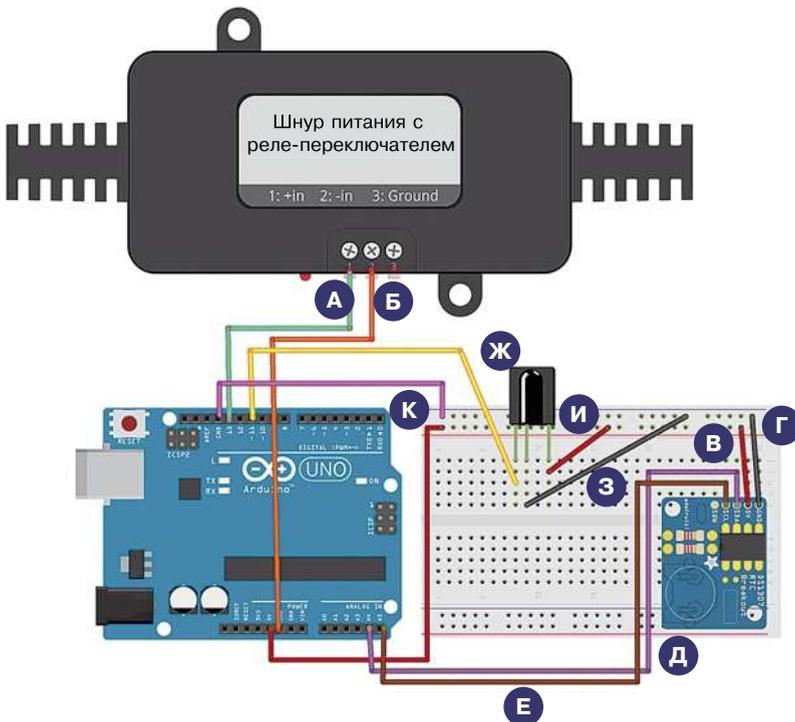


Рис. 12.11. Соберите лавовую лампу «Бадди» в соответствии с рисунком



Рис. 12.12. ИК-датчик с припаянными проводами

- Б. Подключите клемму 2 к пину GND на Arduino. Это оранжевый провод на **рис. 12.11**.
 - В. Подключите модуль RTC к макетной плате и соедините с пином 5V (красный провод) на шине питания макетной платы.
 - Г. Подключите GND (черный провод) к шине питания макетной платы.
 - Д. Подключите SDA (фиолетовый провод) к пину A4 на Arduino.
 - Е. Подключите SCL (коричневый провод) к пину A5 на Arduino.
 - Ж. Припаяйте соединительные провода (см. **рис. 12.12**) к ИК-датчику так, чтобы датчик можно было прикрепить внутри коробки.
3. Левый провод датчика, если смотреть на него со стороны выпуклости датчика инфракрасного излучения (желтый провод), подключите к пину 11 на Arduino.
 - И. Центральный провод ИК-датчика (черный провод) идет к шине заземления макетной платы.
 - К. Правый провод ИК-датчика (красный провод) подключите к шине питания макетной платы.
2. Просверлите отверстие диаметром 6 мм на задней стенке коробки. Просверлите отверстие 12 мм на передней стенке коробки.
 3. С помощью клеевого термопистолета приклейте ИК-датчик к передней стенке коробки, пропустив провода через отверстие так, как показано на **рис. 12.13**.
 4. С помощью клеевого термопистолета приклейте макетную плату и Arduino на место внутри коробки.
 5. Подключите провод питания Arduino и шнур питания с реле-переключателем к удлинителю. Я использовал удлинитель с тремя розетками, но можно взять другой, который вам нравится.



Рис. 12.13. ИК-датчик выглядит из коробки

6. Пропустите провод питания лавовой лампы через отверстие в задней стенке коробки так, чтобы его можно было подключить в разъем шнура питания с реле-переключателем. Сборка должна выглядеть примерно так, как на **рис. 12.14**.

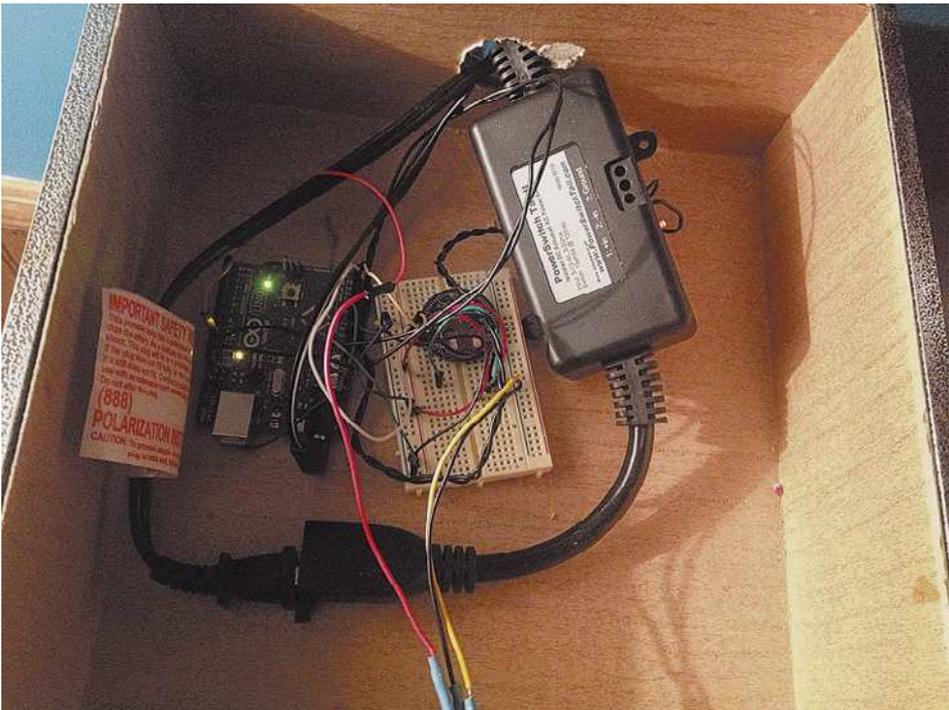


Рис. 12.14. Внутреннее наполнение лавовой лампы «Бадди»

Код для лавовой лампы «Бадди»

Это достаточно сложный код, он отвечает:

- за распознавание ИК-сигналов с пульта дистанционного управления;
- получение данных с модуля RTC.

Справиться с этими задачами вам помогут три библиотеки, которые нужно загрузить прежде, чем вы загрузите скетч.

- Библиотека `wire.h` уже установлена в Arduino, поэтому можете не беспокоиться о ней.
- Библиотеку RTC можно найти в хранилище GitHub в *Adafruit* по ссылке: <https://github.com/adafruit/RTClib>.
- И наконец, вы можете найти библиотеку `IRremote.h` у Кэна Ширрифа по ссылке: <https://github.com/shirriff/Arduino-IRremote/blob/master/IRremote.h>.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#include <Wire.h>
#include "RTClib.h"
#include <IRremote.h>

int RECV_PIN = 11;
int RELAY_PIN = 13;
int startTime = 1600; // в это время включается лампа -- 1600 означает
                      // 16:00 (4 часа вечера)
int stopTime = 2200; // в это время лампа выключается --
                     // заданное время 2200 или 22:00
                     // (10 часов вечера)

RTC_Millis RTC;
IRrecv irrecv(RECV_PIN);
decode_results results;

void dump(decode_results *results) {
    int count = results->rawlen;
    if (results->decode_type == UNKNOWN) {
        Serial.println("Could not decode message");
    }
}
```

```
for (int i = 0; i < count; i++) { if ((i % 2) == 1) {
    Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
}
else {
    Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);
}
    Serial.print(" ");
}
Serial.println("");
}

void setup()
{
    Serial.begin(57600);
    // в следующей строке на модуле часов реального времени
    // устанавливаются дата и время компиляции скетча
    RTC.begin(DateTime( DATE , TIME ));

    pinMode(RELAY_PIN, OUTPUT);
    pinMode(RECV_PIN, INPUT);
    Serial.begin(9600);
    irrecv.enableIRIn(); // включить приемник
}

int on = 0;
unsigned long last = millis();

void loop() {
    DateTime now = RTC.now();

    int decimalTime = now.hour() * 100 + now.minute(); Serial.
    print(decimalTime);
    delay (1000);

    Serial.println();

    if (decimalTime == startTime) {
        digitalWrite(RELAY_PIN, HIGH);
    }

    if (decimalTime == stopTime) {
        digitalWrite(RELAY_PIN, LOW);
    }

    if (irrecv.decode(&results) ) {
        // если с последнего приема ИК-сигнала прошла ¼ с,
        // реле переключится
        if (millis() - last > 250) {
            on = !on;
        }
    }
}
```

```
digitalWrite(RELAY_PIN, on ? HIGH : LOW);  
    dump(&results);  
}  
last = millis();  
irrecv.resume(); // получаем следующее значение  
}  
}
```

В следующей главе

Глава 13 «Управление двигателями» познакомит вас с различными методами управления двигателями, будь то шаговые электродвигатели, сервоприводы или старые добрые электродвигатели постоянного тока. Вы узнаете, как управлять ими с помощью Arduino. Затем вы создадите машину для пуска мыльных пузырей.

Управление электродвигателями

В этой главе вы расширите свои знания об электродвигателях, изучив платы, которые позволяют управлять всеми видами двигателей. Затем вы поработаете над проектом этой главы, «Баблботом», который дарит радость и пускает мыльные пузыри (**рис. 13.1**).



Рис. 13.1. Хотите украсить жизнь мыльными пузырями? Хорошо, не вопрос

Как управлять двигателями

Мы уже рассмотрели основные виды электродвигателей — шаговые, сервоприводы и электродвигатели постоянного тока. Теперь давайте поговорим о том, как управлять ими с помощью Arduino. Секрет заключается в том, что вы должны использовать чип управления двигателем, например «L293D». Он регулирует поток данных между двигателем и Arduino, позволяя управлять большим количеством двигателей, чем обычно.

Более того, чипы управления двигателем — это удобные «мозги» управления моторными платами, которые включают дополнительные функции, например подачу питания на отдельные двигатели. Давайте рассмотрим три интересных примера плат управления двигателем.

Плата расширения Adafruit Motor Shield

Плата расширения для управления двигателем от компании *Adafruit* (см. **рис. 13.2**) — это прекрасный инструмент для управления двигателями. Запуск двигателя с помощью Arduino ограничен количеством пинов и питающих контактов, потому что двигателям необходимо и то,

Источник: Adafruit Industries

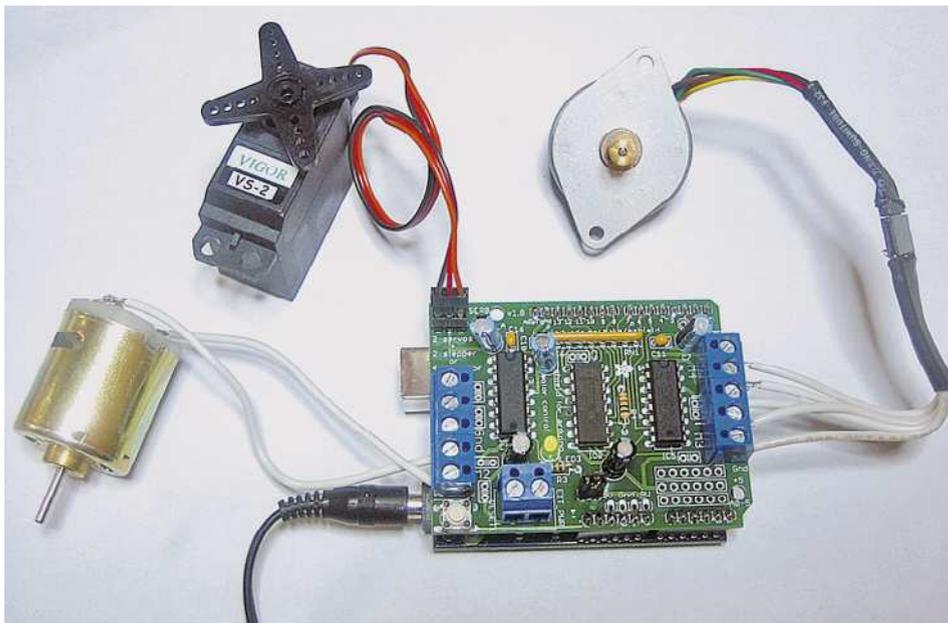


Рис. 13.2. Плата расширения для управления двигателем *Adafruit Motor Shield* (печатная плата в центре) может управлять электродвигателями постоянного тока, шаговыми двигателями и сервоприводами

и другое. Например, у сервопривода три провода, каждому из которых необходим собственный пин на Arduino. Более того, пин 5V на Arduino едва может справиться с одним проводом от сервопривода, не говоря уже о большем количестве проводов. Плата расширения подает питание и управляет данными таким образом, что нуждается в минимальном количестве ресурсов. Плата может запускать два сервопривода и два шаговых двигателя, или четыре электродвигателя постоянного тока вместо шаговых. Все это благодаря удобному форм-фактору платы расширения.

«EasyDriver» от Schmalz Haus

Шаговый контроллер «EasyDriver» (рис. 13.3) представляет собой более элегантное решение, чем полноценная плата расширения. Это одноплатный контроллер с входами для передачи данных и питания со встроенным стабилизатором напряжения, выравнивающим питание, поступающее на шаговый двигатель. Устройство меньше по размеру, чем плата расширения Arduino.

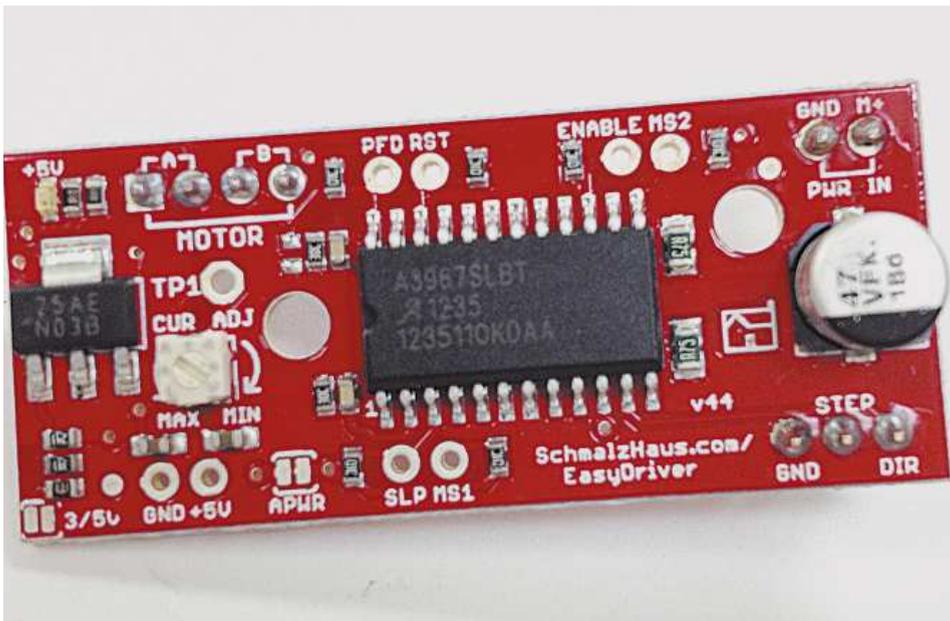


Рис. 13.3. «EasyDriver» с легкостью управляет шаговыми двигателями

«MegaShield» от Bricktronics

Благодаря трем чипам «L293D», возможности управлять шестью двигателями и принимать информацию от четырех датчиков плата расширения «MegaShield» от компании *Bricktronics* (рис. 13.4) позволяет управлять двигателями «LEGO MINDSTORMS», даже с размещением запатентованных кабелей LEGO.

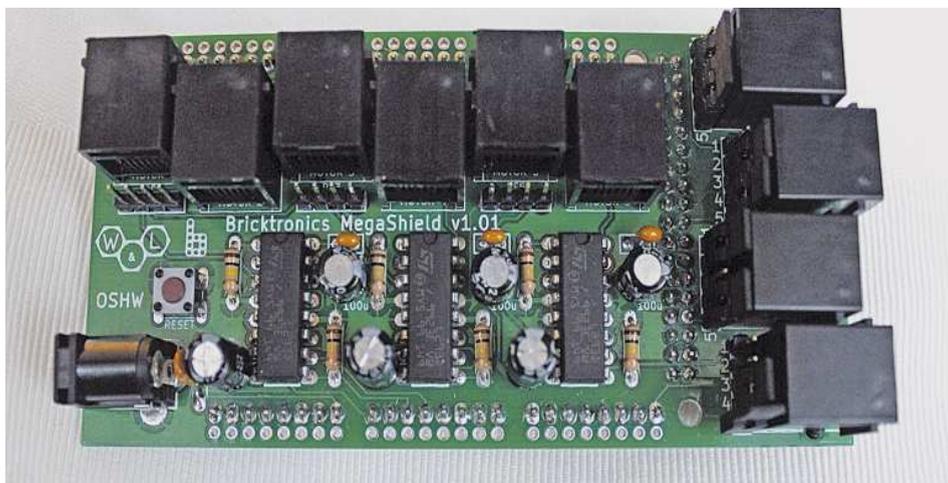


Рис. 13.4. Плата расширения «MegaShield» от компании *Bricktronics* управляет шестью двигателями и принимает информацию от четырех датчиков

Это довольно надежная плата, в основном соответствующая спецификации собственного микроконтроллера LEGO MINDSTORMS, совместимая с программированием платформы Arduino.

Запомните термин «MegaShield». Он связан с Arduino Mega, по-настоящему большой платформой Arduino с гораздо большей вычислительной мощностью, чем Arduino UNO. В основном эта плата расширения используется для Arduino Mega и не работает с Arduino UNO, который мы используем в этой книге. Не бойтесь, разработчики из компании *Wayne & Layne* также создали плату расширения и для Arduino UNO.

Включаем двигатель с помощью TIP-120

Одна из причин существования плат управления двигателями заключается в том, что Arduino трудно выполнять эту функцию собственными возможностями. Микроконтроллер имеет только два пина на 3,3V и 5V. Может быть, этого и достаточно для одного двигателя, но у робота обычно их несколько, и Arduino может не справиться. Одно из возможных решений — использовать плату управления двигателем, о которой я говорил в начале этой главы. А более простое и дешевое решение — использовать транзистор Дарлингтона, например TIP-120 (см. **рис. 13.5**)¹.

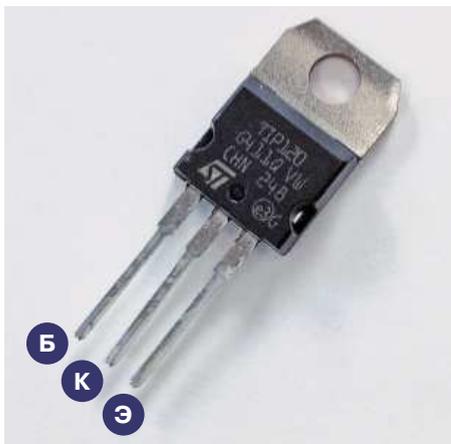


Рис. 13.5. Транзистор Дарлингтона управляет электрическим током, так что Arduino не придется об этом беспокоиться

Транзистор Дарлингтона — это полупроводниковый электронный коммутатор, который позволяет управлять током при подаче на вход небольшого потенциала.

Вот как он работает. Транзистор имеет три вывода, которые называются база (Б), коллектор (К) и эмиттер (Э). Вы можете увидеть их на **рис. 13.5**.

- **База** — этот вывод, через который поступает сигнал от Arduino.
- **Коллектор** — контакт, к которому подключается напряжение питания.
- **Эмиттер** — слой полупроводника, с помощью которого ток от коллектора подается на нагрузку, когда получает команду с базы.

¹ Составной транзистор (схема) Дарлингтона был изобретен в 1953 г. инженером *Bell Laboratories* Сидни Дарлингтоном. Схема является каскадным соединением двух (реже трех или более) биполярных транзисторов, включенных таким образом, что нагрузкой в эмиттерной цепи предыдущего каскада является переход база-эмиттер транзистора последующего каскада (т. е. эмиттер предыдущего транзистора соединяется с базой последующего), при этом транзисторы соединяются коллекторами. Коэффициент усиления по току пары Дарлингтона очень высок и приблизительно равен произведению коэффициентов усиления по току транзисторов, составляющих такую пару. — *Прим. перев.*

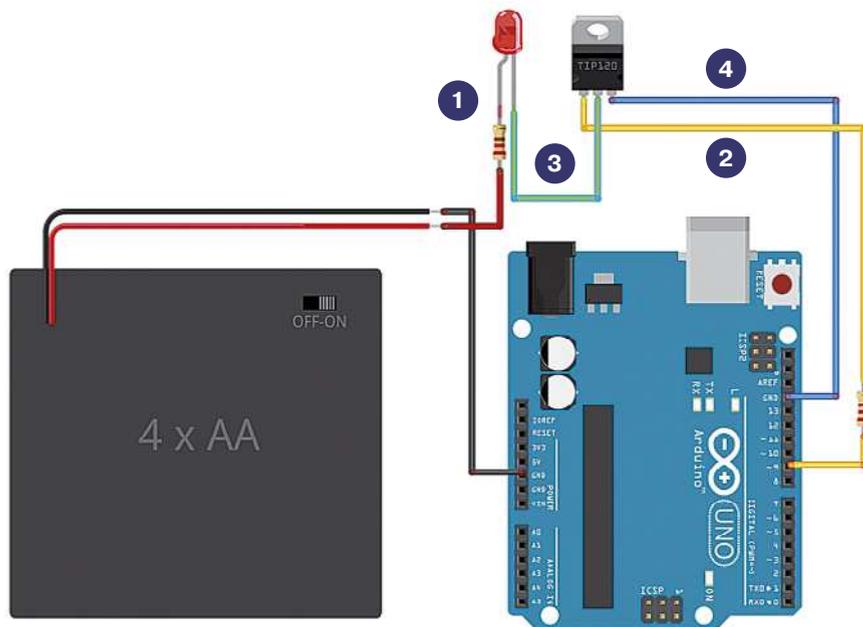


Рис. 13.6. Подключение составного транзистора TIP-120

На **рис. 13.6** показан простой пример того, как вы можете подключить светодиод, чтобы он включился после получения сигнала от Arduino.

1. Источник питания подключается к положительному проводу светодиода через резистор 220 Ом.
2. База подключается к цифровому пину Arduino; этот пин запустит электрическую схему.
3. Отрицательный пин светодиода подключается к коллектору транзистора.
4. Эмиттер, который должен подать напряжение при срабатывании, идет к земле. Что это дает? Подумайте об этом следующим образом: база транзистора является триггером, а коллектор и эмиттер — это электрическая цепь. Когда база получает сигнал, весь цикл, начиная с аккумулятора, проходит через резистор, светодиод, транзистор, а затем заземляется и сразу же становится замкнутой электрической схемой.

⚙ Проект: шаговый поворотник

В этом проекте вы создадите шаговый двигатель, который будет поворачиваться под управлением потенциометра (см. **рис. 13.7**). Как вы узнали из **главы 1** «Знакомство с Arduino», потенциометр — это аналоговое устройство, которое обеспечивает переменную величину сопротивления в зависимости от того, насколько повернута ручка прибора. Вы можете взять показание потенциометра и перевести его значение в градусы, которые могут быть использованы для задания угла поворота двигателя.

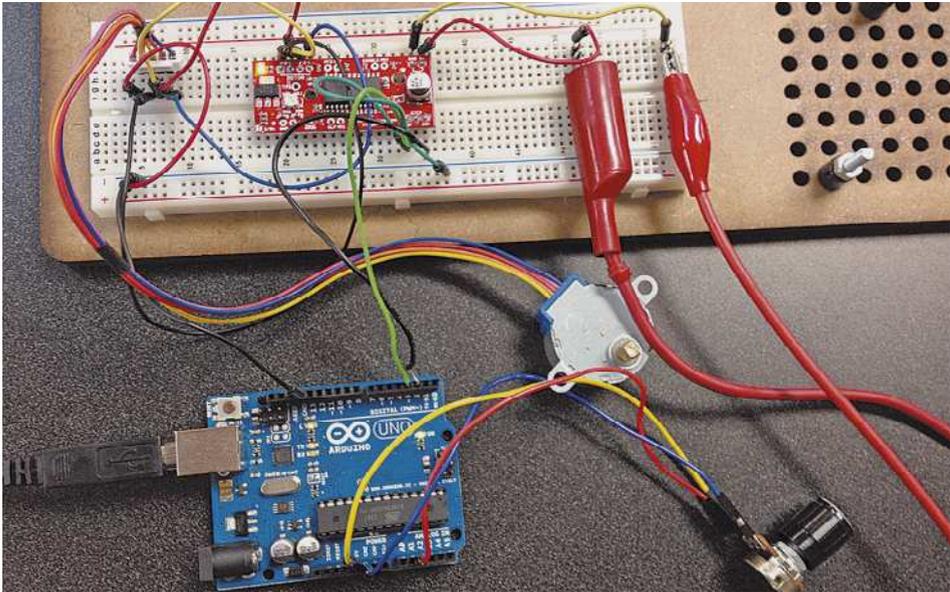


Рис. 13.7. Поверните шаговый двигатель в соответствии с показаниями потенциометра

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Для создания этого проекта вам понадобятся следующие элементы.

- Arduino UNO.
- Источник питания 12 В (можно использовать комплект из 8 батареек AA, если у вас нет другого источника питания).
- «EasyDriver» от *Shmalz Haus* (описан ранее в этой главе).
- Шаговый двигатель.
- Потенциометр (можно использовать любой).
- Соединительные провода.
- Макетная плата.

Инструкции

Этот проект включает удивительно большое количество соединений, но не теряйтесь, просто следуйте схеме на **рис. 13.8**.

1. Подключите потенциометр. Левую клемму (если клеммы обращены к вам) подключите к пину 5V на Arduino (серый провод), среднюю клемму — к пину A2 (зеленый провод) и правую клемму (коричневый провод) к пину GND.
2. Подключите «EasyDriver» к макетной плате. Полученное устройство подключите к Arduino с помощью двух соединительных проводов, отмеченных белым и фиолетовым цветом на **рис. 13.8**. Белый провод соединяет пин, обозначенный как «STEP», на печатной плате, с пином 3 на Arduino. Фиолетовый провод соединяет пин, обозначенный как «DIR» на печатной плате, с пином 2 на Arduino. Пин, обозначенный как GND на «EasyDriver», подключается к шине заземления.

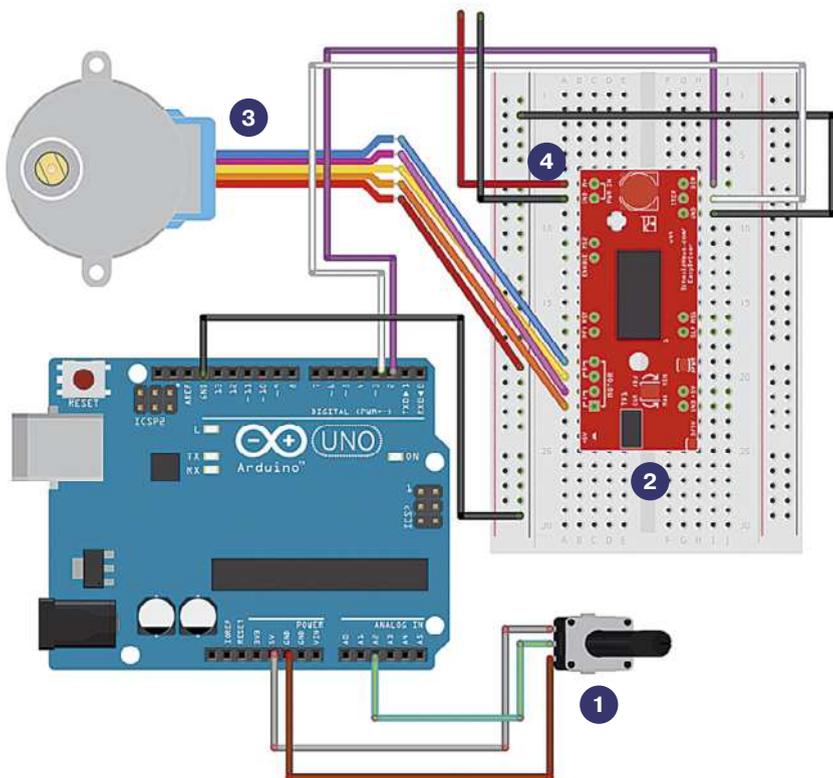


Рис. 13.8. Соберите проект в соответствии со схемой на рисунке

- Шаговый электродвигатель имеет пять проводов: красный, оранжевый, желтый, розовый и синий. Красный провод — это заземление, он должен быть подключен к шине заземления на макетной плате. Оставшиеся четыре провода подключаются в четыре пина, обозначенные как «MOTOR» на печатной плате «EasyDriver». Однако есть небольшой нюанс. Вы не можете просто подключить их в обозначенном порядке. Желтый и розовый провода надо поменять так, как отмечено на **рис. 13.8**.
- Наконец, подключите источник питания на 12 В. Он подключается на два пина, обозначенные как «PWR IN», я использовал красный и черный провода на **рис. 13.8**, чтобы показать, как это сделать. Также не забудьте подключить шину заземления макетной платы к пину GND на Arduino. Вы закончили!

Код для шагового поворотника

Используйте следующий код, чтобы запрограммировать шаговый поворотник.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
#define DIR_PIN 2
#define STEP_PIN 3
#define potPin A2

void setup() {
  pinMode(DIR_PIN, OUTPUT);
  pinMode(STEP_PIN, OUTPUT);
  pinMode(potPin, INPUT);
  Serial.begin(9600);
}

void loop(){
  int potReading = analogRead(potPin);
  Serial.println(potReading);
  rotateDeg(potReading, 1);
  delay(1000);
}
```

```
}  
  
void rotate(int steps, float speed) {  
  
    int dir = (steps > 0)? HIGH:LOW;  
    steps = abs(steps);  
  
    digitalWrite(DIR_PIN,dir);  
  
    float usDelay = (1/speed) * 70;  
  
    for(int i=0; i < steps; i++){  
        digitalWrite(STEP_PIN, HIGH);  
        delayMicroseconds(usDelay);  
  
        digitalWrite(STEP_PIN, LOW);  
        delayMicroseconds(usDelay);  
    }  
}  
  
void rotateDeg(float deg, float speed){  
  
    int dir = (deg > 0)? HIGH:LOW;  
    digitalWrite(DIR_PIN,dir);  
  
    int steps = abs(deg)*(1/0.225);  
    float usDelay = (1/speed) * 70;  
  
    for(int i=0; i < steps; i++){  
        digitalWrite(STEP_PIN, HIGH);  
        delayMicroseconds(usDelay);  
  
        digitalWrite(STEP_PIN, LOW);  
        delayMicroseconds(usDelay);  
    }  
}
```

Проект: «Баблбот»¹

В заключительном проекте этой книги вы превратите лето (или любое другое время года; зависит от того, когда вы читаете эти строки) в волшебную сказку с прекрасным «Баблботом», изображенным на **рис. 13.9!** Это простой робот в деревянном каркасе, который опускает ракетку для пузырей в лоток с мыльным раствором, а затем поднимает ее и выдувает пузыри. Он будет пускать пузыри до тех пор, пока не закончится мыльный раствор, или пока вам не надоест, и вы не выдерните вилку из розетки!

¹ «Баблбот» (от англ. *BubbleBot*) — робот, запрограммированный на выдувание мыльных пузырей. — *Прим. перев.*



Рис. 13.9. Отличный залп, «Баблбот»!

ПЕРЕЧЕНЬ КОМПОНЕНТОВ

Соберите следующие комплектующие, чтобы построить вашего «Баблбота».

- Arduino UNO.
- Плата расширения для управления двигателем.
- Сервонасадки (прекрасно подойдут те, которые входят в комплект сервопривода).
- Переключатель.
- Мини макетная плата.
- Комплект батареек 12 В.
- Компьютерный кулер.
- МДФ толщиной 6 мм для каркаса (должно хватить листа размером 45х60 см).
- Ракетка для пузырей (я использовал ракетку из набора «Miracle Bubbles»).
- Деревянный стержень диаметром 12 мм, длиной 24 см.
- Резьбовые стержни диаметром 6 мм, длиной 20 см, 2 шт. (вы можете купить любые резьбовые стержни в хозяйственном магазине).
- Саморезы для дерева № 4-40 х 9 мм, 2 шт.
- Гайки диаметром 6 мм со стопорными шайбами, 8 шт.
- Крепежные винты № 4-40 х 25 мм, 8 шт.
- Соединительные провода (те же самые, которые вы использовали для работы над проектами этой книги).
- Столярный клей.
- Термоклей и термопистолет.

Инструкции

После того как вы соберете все необходимое для проекта, можно приниматься за проект! Следуйте этим инструкциям, чтобы собрать собственного «Баблбота».

1. С помощью лазерного резака вырежьте части корпуса из МДФ (см. **рис. 13.10**). Вы можете загрузить дизайн деталей по ссылке: <https://github.com/n1/Arduino-For-Beginners>. Как вариант, постройте корпус привычным для вас способом и просверлите отверстия для крепежа.

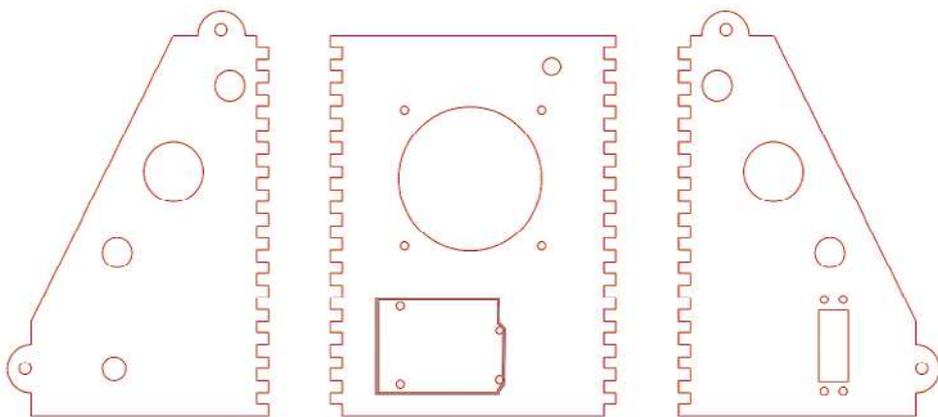


Рис. 13.10. Вырежьте эти детали с помощью лазерного резака, чтобы построить корпус для «Баблбота»

2. Соедините шипы боковых и задней частей корпуса и склейте их столярным клеем.

СОВЕТ

МОЙ ПРОТОТИП ВЫГЛЯДИТ НЕСКОЛЬКО ИНАЧЕ

Когда я собирал собственный «Баблбот», для скрепления частей корпуса использовал саморезы вместо углового соединения на прямой шип. Собрал робота, я несколько изменил тактику сборки каркаса, применив ранее полученные знания. Таким образом, существует несколько косметических различий между первым и вторым вариантами сборки, но в основном они одинаковы.



Рис. 13.11. Резьбовые стержни обеспечивают надежность каркаса

3. Вставьте резьбовые стержни с фиксирующими шайбами и гайками так, как показано на **рис. 13.11**. Затяните крепеж, зафиксировав каркас.
4. С помощью саморезов № 4 × 9 мм прикрепите сервонасадку с одного конца деревянного стержня так, как показано на **рис. 13.12**.



Рис. 13.12. Прикрепите сервонасадку с одного конца стержня



Рис. 13.13. Стержень выступает с противоположной стороны каркаса

5. Вставьте стержень в отверстие напротив сервопривода. Его обязательно крепить, поскольку противоположный конец соединен с сервоприводом. На **рис. 13.13** показано, как это должно выглядеть. Если стержень слишком длинный и его конец выступает из отверстия, вы можете спокойно обрезать его.
6. Просверлите направляющие отверстия и прикрепите ракетку для пузырей к стержню с помощью саморезов № 4 × 9 мм, как показано на **рис. 13.14**.



Рис. 13.14. Прикрутите ракетку для пузырей к стержню



Рис. 13.15. Тумблер включает и выключает машинку для пузырей

7. Вставьте тумблер в небольшое отверстие в верхней части задней панели. Прикрепите его с помощью шестигранных гаек, которые входят в комплект тумблера. Пропустите провода через отверстие вентилятора так, как показано на **рис. 13.15**.
8. Прикрепите кулер (см. **рис. 13.16**) к задней панели корпуса с помощью винтов № 4 × 25, шайб и гаек.



Рис. 13.16. Прикрепите кулер к задней панели корпуса

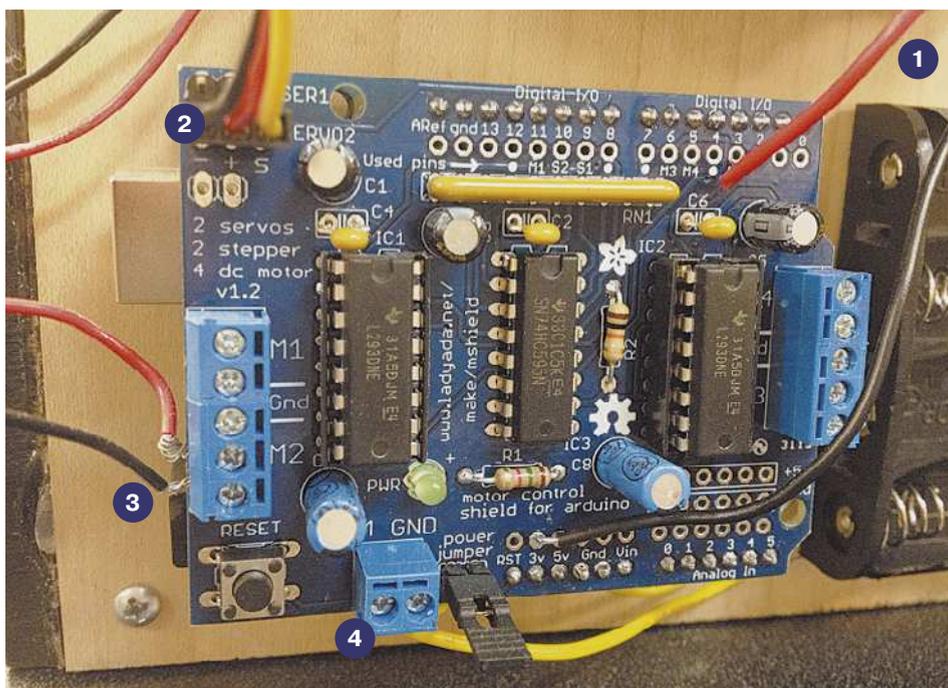


Рис. 13.17. Плата расширения для управления двигателем контролирует работу вашего «Баблбота»

Прикрепите Arduino к задней панели с помощью винтов № 4 × 25. Затем прикрепите плату расширения для управления двигателем к Arduino, соединив пины платы со штырями Arduino. Подключите плату расширения, следуя указаниям **рис. 13.17**.

1. Припаяйте провода к выключателю, если вы еще это не сделали, затем соедините черный провод с пином 3V на плате расширения и красный провод с пином 2. Возможно, вам нужно будет добавить капельку припоя для того, чтобы провода не отошли от платы.
2. Подключите провода сервопривода к пинам, отмеченным как «SERVO2». Красный провод надо подключить к «+», черный — к «-», а желтый — к «S». Обратите внимание, что сервоприводы объединяют свои провода в тройной вилке, так что все абсолютно надежно.
3. Подключите провода кулера к синей колодке с клеммами, обозначенной как «M2». Красный и черный провода подойдут к любой клемме «M2».
4. С помощью клеевого термопистолета приклейте комплект батареек 2 В к задней части корпуса. Подключите комплект ба-

тареек к синей колодке с клеммами, обозначенной как «+М» и «GND», чтобы можно было запустить двигатель.

9. С помощью клеевого термопистолета приклейте батарейный блок 9 В (не показан на **рис. 13.17**) к корпусу и подключите штекер к штепсельной вилке Arduino. Это обеспечит питанием Arduino.

Код для «Баблбота»

Загрузите следующий код вашего «Баблбота» в Arduino. Обратите внимание, что вам нужно загрузить библиотеку `AFMotor.h` от *Adafruit* из хранилища по ссылке: <https://github.com/adafruit/Adafruit-Motor-Shield-library>.

КОД

Вам не нужно вводить код вручную. Просто перейдите по ссылке <https://github.com/n1/Arduino-For-Beginners> и загрузите бесплатный код.

```
// этот код основан на примере кода платы расширения
// для управления двигателями Adafruit Motor Shield

#include <AFMotor.h>
#include <Servo.h>

AF_DCMotor motor(2);
Servo servol;
int toggle = 2;
int toggleStatus = 0;

void setup() {
  Serial.begin(9600);

  pinMode(toggle, INPUT);

  // включаем сервопривод
  servol.attach(9);

  // включаем электродвигатель № 2
  motor.setSpeed(200);
  motor.run(RELEASE);
}
```

```
int i;
void loop() {

    int toggleStatus = digitalRead(toggle);
    if (toggleStatus == 1) {
        Serial.println(toggleStatus);
        motor.run(FORWARD);
        for (i=255; i!=0; i--) {
            servol.write(i-90);
            delay(30);
        }

        delay(5000);
        Serial.println("waiting1");

        motor.run(FORWARD);
        for (i=0; i<2000; i++) {
            motor.setSpeed(i);
            delay(5);
        }
        delay(5000);
    }
}
```

Глоссарий

Аналоговые данные — данные, посылаемые в виде непрерывного синусоидального сигнала, меняющегося в зависимости от напряжения; отличается от цифровых данных, которые передаются в виде чередующихся последовательностей «1» и «0».

Барбот — робот, разработанный для того, чтобы делать и подавать коктейли.

Барометрический датчик — датчик, который определяет изменение атмосферного давления почти так же, как это делает барометр.

Библиотека — поддерживающий код, относящийся к скетчу Arduino, позволяющий вам сохранить основной скетч максимально простым.

Бит — единица измерения количества информации, обычно 0 или 1.

Вывод — провод или клемма на компоненте, к которой прикрепляется провод.

Вывод прерывания — вывод на Arduino, который может прервать запущенный цикл. Если вы захотите остановить цикл путем нажатия на кнопку, вам нужно подключить эту кнопку к выводу прерывания.

Датчик — электронное устройство, которое передает данные (или напряжение) об окружающей обстановке на микроконтроллер.

Датчик детонации — датчик, который определяет вибрацию и посылает напряжение в соответствующую часть схемы.

Датчик изгиба — датчик, встроенный в гибкий пластик. По сути, это переменный резистор, сопротивление которого меняется в зависимости от того, насколько согнут пластик.

Датчик наклона — датчик с завальцованным шариком внутри, который регистрирует наклон датчика.

Датчик освещенности — датчик, который определяет интенсивность освещения. Некоторые из них работают как переменные резисторы, где уровень освещенности меняет сопротивления, в то время как другие — это цифровые устройства, которые посылают на микроконтроллер числовые данные.

Датчик температуры и влажности — цифровой датчик, который фиксирует температуру и влажность и отображает числовое значение на микроконтроллер.

Диод — электронный элемент, который обычно позволяет электрическому току двигаться через себя только в одном направлении.

Заземление — обратный путь электрической цепи. На батарейках заземление отмечается как «-» (минус). В электронике термин «заземление» часто сокращенно обозначается как GND¹.

Замок-выключатель — выключатель, который управляется ключом, позволяя ограничить доступ нежелательных посетителей к проекту.

¹ GND — от англ. *ground* — земля. — Прим. перев.

- Изоляционные стойки** — металлические или пластиковые вставки, которые создают дополнительное пространство или поддержку между печатной платой и другими поверхностями.
- ИК-приемник** — датчик, определяющий инфракрасное излучение, которое пульсирует с частотой 38 МГц.
- Интегральные схемы (ИС)** — ряд миниатюрных схем, встроенных в один пластиковый корпус.
- Интегрированная среда разработки (Integrated Development Environment — IDE)** — программное обеспечение, с помощью которого программисты могут написать код.
- Инфракрасное (ИК-) излучение** — световой спектр частоты, находящийся за пределами восприятия человеческого зрения. ИК-излучение часто используется для передачи данных на небольшие расстояния, например, чтобы послать сигнал «выкл.» на телевизор.
- Керфбэндинг (техника «пропил-изгиб», от англ. *Kerf-bending*)** — технология лазерной резки, которая позволяет сгибать тонкие листы древесины с помощью специально сделанных прорезов в материале.
- Колodka с гнездами** — ряды с контактами на макетной плате, которые расположены перпендикулярно шине питания и шине заземления.
- Коммутационная плата** — небольшая печатная плата для управления единичным компонентом.
- Компиляция** — перевод информации с одного машинного языка на другой. Чаще всего используется для преобразования информации в машинный код.
- Конденсатор** — электронный компонент, который сохраняет небольшой заряд электричества в электростатическом поле.
- Лазерный диод** — электронный модуль, излучающий лазерный луч при подаче напряжения на клеммы.
- Лазерный резак**, также известный как лазерный гравер, — инструмент, прожигающий лазером тонкие материалы, такие как картон, МДФ и ДСП.
- Ленточнопильный станок** — электрическая пила, применяемая в металло- и деревообработке с использованием ленточного полотна в виде кольца.
- Макетная плата** — пластиковая плата с отверстиями и скрытыми проводниками, благодаря которым можно быстро собрать электросхему без припоя.
- Массив** — в терминологии программирования список значений, хранящихся для использования в будущем.
- МДФ (Medium Density Fiberboard — MDF)** — древесноволокнистая плита средней плотности, которая легко поддается обработке и часто используется в проектах.
- Микроконтроллер** — миниатюрный компьютер, который способен принимать данные от датчиков и активировать двигатели и лампы (светодиоды).
- Модуль режима реального времени (Real-Time Clock module — RTC)** — чип хронометража с резервной батареей, предназначенной для поддержания правильного времени в течение нескольких месяцев.
- Мультиметр** — основной инструмент электроинженера, сочетающий в себе вольтметр и омметр с дополнительными функциями.
- Мультигул (мультиинструмент)** — складной инструмент, часто в виде плоскогубцев, с дополнительными инструментами, такими как шило, лезвия, ножницы и т. д.
- Открытое аппаратное и программное обеспечение** — проекты по электронике, в которых коды и электронные разработки открыты для свободного доступа, использования и обсуждения. При этом каждый может модифицировать или создать заново предлагаемый продукт.
- Пассивный инфракрасный датчик** — инфракрасный датчик, который обнаруживает движение при незначительных изменениях температуры.

- ПВХ** — поливинилхлорид, также известный как ПВХ; это самые подходящие пластиковые трубки для выполнения проектов, чаще всего используются в качестве водопроводных труб.
- Перистальтический насос** — насос, который работает, периодически сжимая и разжимая трубку, через которую перекачивается жидкость. При этом жидкость не вступает в контакт с насосом и не загрязняется.
- Печатная плата (printed circuit board — PCB)** — композитная плата, покрытая проводящим электрический ток материалом, что позволяет гравировать схемы на плате и тем самым создавать электронные узлы.
- Пины** — контакты для питания и сбора данных на Arduino.
- Плазменный резак** — станок с ЧПУ, который режет металл в соответствии с заданной программой.
- Плата** — короткое название монтажной печатной платы.
- Плата расширения** — дополнительная печатная плата для Arduino. Она крепится сверху, подключает определенные выходы микроконтроллера и предоставляет Arduino дополнительные возможности.
- Последовательная передача (данных)** — метод передачи данных, при котором данные передаются по одному проводу, с последовательной передачей каждого бита информации.
- Последовательный мониторинг** — окно в интегрированной среде разработки (IDE) Arduino, где можно контролировать последовательный (серийный) трафик. Отличный инструмент для отладки программ.
- Потенциометр** — резистор переменного сопротивления, который регулируется поворотом ручки.
- Принципиальная схема** — изображение, представляющее электрическую цепь с символами, обозначающими разные элементы.
- Пьезодинамик** — элемент, издающий писк, когда на клеммы подается ток.
- Резистор** — электронный элемент, разработанный для ограничения электрического тока, чтобы защитить детали электрической схемы и регулировать напряжение в электрической цепи.
- Реле** — электромеханический выключатель, запускаемый микроконтроллером, который способен контролировать цепь высокого напряжения.
- Роторный инструмент** — небольшой электроинструмент с несколькими типами насадок, начиная с пилы и заканчивая шлифовальным диском. Наверное, вам знаком лидер среди производителей этих инструментов — компания *Dremel*.
- Светодиод (Light Emitting Diode — LED)** — лампочка в мире электроники.
- Светодиодный драйвер** — интегральная схема, которая способна управлять множеством светодиодов без использования пинов на Arduino.
- Семисегментный индикатор** — индикатор чисел или букв, который собран из семи небольших светодиодных сегментов.
- Сервопривод** — двигатель, оснащенный редуктором и энкодером, позволяющим с точностью управлять поворотом вала двигателя.
- Сетевая схема (ячейная сеть)** — сеть, состоящая из нескольких узлов, каждый из которых способен видеть любой другой узел.
- Скетч** — простонародное название программы для Arduino, которая управляет выводами микроконтроллера.
- Скорость передачи данных в бодах** — скорость, с которой передаются данные. Значение передачи приравнивается к количеству символов, отправленных в секунду. Так, 9600 бод равно 9600 символам, передаваемым в секунду.
- Соединительный провод (джампер, от англ. *jumper*)** — общий термин для обозначения соединительных проводов или проводников, используемых в проектах по электронике.
- Соленоид** — двигатель, который в отличие от вращательных движений совершает возвратно-поступательные движения. Часто используется в клапанах.

- Спецификация** — составленное производителем описание электронных деталей или функций, допусков и архитектуры сборки.
- Стабилизатор напряжения** — компонент, поддерживающий нужное напряжение в проекте по электронике.
- Станки с числовым программным управлением (ЧПУ)** — набор электрифицированного инструмента на одной станине, управляемой программой, которая задана на компьютере.
- Сугру (*sugru*)** — формовочная, быстрозастывающая замазка для многоцелевого использования.
- Термоусаживаемые трубки** — резиновые трубки, которые не проводят ток. Используются для изоляции соединения проводов. При нагревании трубки сужаются и изолируют оголенные провода.
- Токарный станок** — устройство для обработки дерева.
- Транзистор** — миниатюрный электронный переключатель, который управляется электрическим сигналом.
- Трехцветный светодиод (RGB LED)** — светоизлучающий модуль, состоящий из трех элементов красного, синего и зеленого цветов. Посредством одного или нескольких элементов можно создать большое разнообразие цветов.
- Ультразвуковой датчик** — датчик, который обнаруживает препятствия и измеряет расстояния, выдавая неслышимый сигнал и принимая отраженный.
- Циркулярная пила** — пила в виде рабочего стола с выступающей над поверхностью дисковой частью.
- Цифровые данные** — тип данных, который состоит исключительно из инструкций «да» и «нет», в противоположность аналоговым данным, которые состоят из разных уровней напряжения.
- Чип управления двигателем** — интегральная схема, оптимизированная для управления двигателями и расширяющая возможности Arduino.
- Шаговый электродвигатель** — двигатель, который разработан для пошагового вращения. Как правило, имеет четыре или более клеммы провода.
- Шина заземления** — полосковый проводник на макетной плате; как правило, отмечен черным или синим цветом и обозначается как заземление.
- Шина питания** — полосковый проводник на макетной плате, предназначенный для подачи напряжения на плату.
- Шнур с реле-переключателем** — удобный и безопасный сетевой удлинитель, включающий и выключающий питание при получении сигнала от внешнего микроконтроллера.
- Штангенциркуль** — инструмент для точного измерения маленьких размеров.
- Электродвигатель постоянного тока** — обычный двигатель, вращение ротора которого управляется подачей напряжения на клеммы.
- Электросхемный бендинг (*circuit bending*)** — техника модернизации электронных игрушек и устройств для изменения их поведения.
- Энкодер** — устройство, которое определяет угол поворота оси двигателя и возвращает это значение на микроконтроллер.
- 3D-принтер** — машина, способная вытягивать и наслаивать пластик, чтобы сформировать трехмерный объект.
- Bluetooth** — беспроводной протокол передачи данных малой мощности, используемый в компьютерных мышах, беспроводных наушниках и в других коммерческих целях.
- Fritzing** — бесплатное программное обеспечение для визуализации электронных деталей, созданное для проектирования электросхем в режиме онлайн. Посмотрите на изображения электронных схем в этой книге — все они сделаны с помощью этой программы.
- XBee** — модуль для популярного протокола *Zigbee*, который часто используется для домашней автоматизации.

Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"

Электронное издание для досуга

Серия: «РОБОФИШКИ»

Бейктал Джон

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

Для детей старшего школьного возраста

Ведущий редактор *Ю. А. Серова*

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

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

Ведущий методист *В. В. Тарапата*

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

Корректор *Л. И. Трифонова*. Компьютерная верстка: *О. Г. Ланко*

Подписано к использованию 18.05.20.

Формат 155×225 мм

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

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

СТАТЬ ИЗОБРЕТАТЕЛЕМ ЛЕГКО!

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

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

Прочитав эту книгу, ты научишься:

- ◆ делать лазерную сигнализацию, которая будет защищать твой дом от непрошенных гостей;
- ◆ использовать беспроводную связь для создания дверного звонка;
- ◆ собирать робота для полива растений, который сэкономит твое время;
- ◆ делать «Лампу настроения», меняющую свой цвет в зависимости от освещенности, и многому другому.

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

info@pilotLZ.ru
www.pilotLZ.ru

