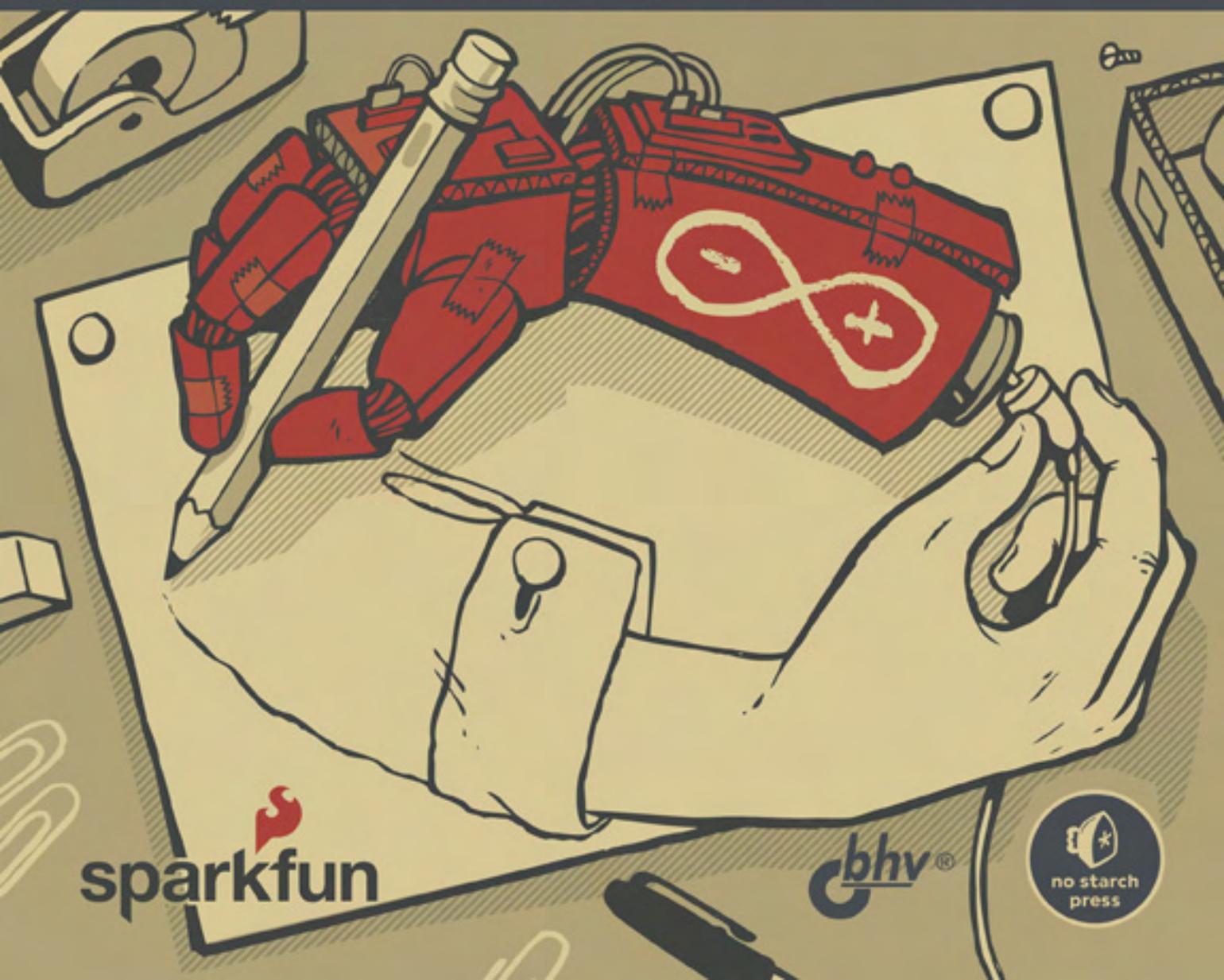


ARDUINO ДЛЯ ИЗОБРЕТАТЕЛЕЙ

ОБУЧЕНИЕ ЭЛЕКТРОНИКЕ
НА 10 ЗАНИМАТЕЛЬНЫХ ПРОЕКТАХ

БРАЙАН ХУАНГ, ДЕРЕК РАНБЕРГ



 sparkfun

 bhv®



THE ARDUINO INVENTOR'S GUIDE

LEARN ELECTRONICS BY MAKING
10 AWESOME PROJECTS

BY BRIAN HUANG
AND DEREK RUNBERG



no starch
press

SAN FRANCISCO

ARDUINO ДЛЯ ИЗОБРЕТАТЕЛЕЙ

ОБУЧЕНИЕ ЭЛЕКТРОНИКЕ
НА 10 ЗАНИМАТЕЛЬНЫХ ПРОЕКТАХ

БРАЙАН ХУАНГ, ДЕРЕК РАНБЕРГ



Санкт-Петербург
«БХВ-Петербург»
2018

УДК 004

ББК 32.973.26

X98

Хуанг, Б.

X98 Arduino для изобретателей. Обучение электронике на 10 занимательных проектах: Пер. с англ. / Б. Хуанг, Д. Ранберг. — СПб.: БХВ-Петербург, 2018. — 288 с.: ил.

ISBN 978-5-9775-3972-2

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

Для радиолюбителей

УДК 004
ББК 32.973.26

Группа подготовки издания:

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

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

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

«БХВ-Петербург», 191036, Санкт-Петербург, Гончарная ул., 20.

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

© 2017 by SparkFun Electronics
© Перевод на русский язык, оформление. ООО «БХВ-Петербург», ООО «БХВ», 2018

ОБ АВТОРАХ

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

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

Комментарий от Брайана

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



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

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

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

Комментарий от Дерек

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

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



О техническом редакторе английского издания

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

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



ОГЛАВЛЕНИЕ

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

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

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

Проект 2. Домашний светофор..... 33

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

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

Проект 3. Десятипиксельный анимационный дисплей 57

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

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

Проект 4. Измеритель скорости реакции..... 79

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

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

Проект 6. Балансирная балка..... 125

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

Проект 7. Миниатюрная настольная теплица 149

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

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

Проект 8. Робот-рисовальщик 179

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

Идем дальше	199
Экспериментируем с кодом	199
Модифицируем код	200
Бонус	200
Проект 9. Хронометрист автогонок	203
Необходимые компоненты, инструменты и материалы	204
Электронные компоненты	204
Прочие инструменты и материалы	206
Новый компонент: жидкокристаллический дисплей	207
Принцип работы хронометриста автогонок	208
Собираем схему с ЖКД	208
Подключаем питание ЖКД	209
Настройка контраста ЖКД	209
Подключаем линии данных и управления	210
Проверяем работу ЖКД	211
Добавляем остальные компоненты	213
Программа для хронометриста автогонок	215
Быстрая проверка	218
Собираем гоночный комплекс	218
Собираем стартовую башню	219
Собираем и вставляем стартовые ворота	221
Изготавливаем гоночную трассу	222
Монтируем фоторезистор	223
Тестирование и отладка	224
Идем дальше	225
Экспериментируем с проектом	225
Подключение ЖКД через модуль IIC/I2C	227
Модифицируем предыдущие проекты	229
Проект 10. Электронное мини-пианино	231
Необходимые компоненты, инструменты и материалы	232
Электронные компоненты	232
Прочие инструменты и материалы	233

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

ПРИЛОЖЕНИЕ. Дополнительные практические сведения

по электронике..... 249

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

ВВЕДЕНИЕ

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

О чем эта книга?

Эта книга основана на платформе Arduino (www.arduino.cc), включающей микроконтроллерную плату, которую можно запрограммировать для управления источниками света, измерения температуры, реагирования на освещение, связи со спутниками глобальной системы позиционирования¹ и, вообще, для того чтобы с ее помощью создать много разных полезных и занимательных устройств. Язык программирования и среда разработки для платы, которые используются в книге, также являются компонентами платформы Arduino.

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

¹ GPS, от англ. Global Positioning System.

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

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

Почему Arduino?

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

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

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

Чем эта книга отличается от других?

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

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

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

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

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

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

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

Необходимые компоненты и материалы

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

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

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

Кроме того, для создания корпусов проектируемых устройств используются многие обычные

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

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

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

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

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

Набор совместимых компонентов

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

Необходимые инструменты

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

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

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

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

Компьютер

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

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

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

Состав книги

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

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



ДОМАШНИЙ СВЕТОФОР

Первым вашим большим шагом в сторону управления миром посредством встроенных электронных устройств была настройка среды разработки Arduino и создание в ней скетча для мигания светодиодом. Что ж, это был большой шаг для Вас лично, но на длинной дороге к овладению Arduino проект с одним светодиодом — всего лишь маленький шагок. Следующим шагом в этом направлении будет расширение нашего первого проекта и создание скетча и аппаратной инфраструктуры для работы с тремя мигающими светодиодами.

Если вы готовы к этому, тогда приступим к разработке светофора для регулировки оживленного движения в коридоре вашего дома (рис. 2.1).

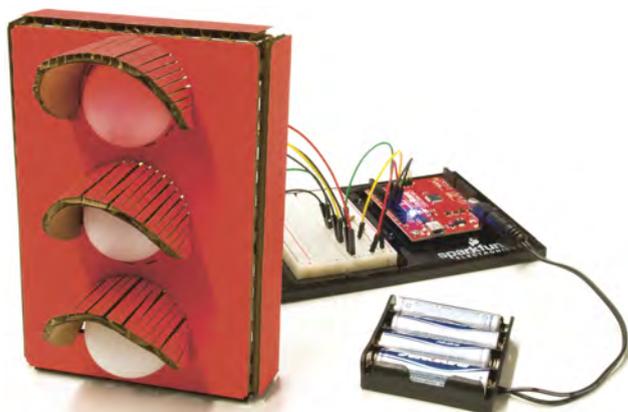


Рис. 2.1. Проект «Светофор» (Stoplight) в сборе



Рис. 2.2. Электронные компоненты для проекта «Светофор»

Необходимые компоненты, инструменты и материалы

Для этого проекта не требуется ничего сложного. Все электронные компоненты для него входят в стандартный состав набора изобретателя Inventor's Kit компании SparkFun, за исключением позиций, отмеченных звездочкой «*». Если вы используете свой набор компонентов или собираете их по отдельности, приведенный далее список необходимых электронных компонентов поможет вам определиться.

Электронные компоненты

Для реализации этого проекта вам потребуются следующие компоненты (рис. 2.2):

- плата RedBoard компании SparkFun (DEV-13975), или плата Arduino Uno (DEV-11021), или любая другая совместимая с Arduino плата, 1 шт.;
- кабель Mini-B USB (CAB-1101) или кабель USB, идущий в комплекте с вашей платой, 1 шт. (на рис. 2.2 не показан);
- безопасная макетная плата (PRT-12002), 1 шт.;

- светодиоды (COM-12062): красный (1 шт.), желтый (1 шт.), зеленый (1 шт.);
- резисторы 330 Ом (COM-08377 или COM-11507 для пакета, содержащего 20 шт.), 3 шт.;
- проволочные перемычки со штекерами на обоих концах (PRT-11026);
- проволочные перемычки со штекером на одном конце и гнездом на другом (PRT-09140)*;
- может пригодиться: держатель для четырех батареек типа AA (PRT-09835)* (на рис. 2.2 не показан).

Примечание

Компоненты, обозначенные звездочкой «*», не входят в состав стандартного комплекта изобретателя SparkFun Inventor's Kit, но предлагаются в отдельном дополнительном комплекте или могут быть приобретены вами по отдельности.

Прочие инструменты и материалы

Чтобы соорудить для электронной схемы корпус, чтобы он выглядел похожим на реальный светофор (см. рис. 2.1), вам потребуются дополнительные инструменты (рис. 2.3) и материалы (рис. 2.4).

Далее приводится краткий перечень этих инструментов и материалов:

- карандаш;
- макетный нож;
- металлическая линейка;
- плоскогубцы (бокорезы);
- инструмент для снятия изоляции;
- клей (клеевой пистолет или клей для моделирования);
- могут пригодиться: дрель и сверло диаметром 4,75 мм;
- может пригодиться: паяльник;
- может пригодиться: припой;
- может пригодиться: держатель «Третья рука»¹ (на рис. 2.3 не показана);
- гофрированный картон (приблизительно 30×30 см) или картонная коробка;
- мячик для настольного тенниса (2 шт.);
- шаблон корпуса (см. рис. 2.15 далее в этом проекте).

¹ См. https://ru.wikipedia.org/wiki/Третья_рука.

Примечание

Для работы с проектами этой книги будет весьма полезно иметь хороший чистый картон. Рекомендуется приобрести несколько листов картона в магазине товаров для рукоделия.



Рис. 2.3. Инструменты, рекомендуемые для проекта «Светофор»



Рис. 2.4. Материалы, рекомендуемые для проекта «Светофор»

Новый компонент: резистор

В предыдущем проекте (см. *проект 1*) мы подключили внешний резистор к плате Arduino напрямую, но в большинстве случаев светодиод лучше подключать через резистор, чтобы ограничить протекающий через него ток. Резисторы (рис. 2.5) используются практически во всех электронных устройствах и также потребуются для этого проекта.

Если вернуться к аналогии, представляющей электрический ток как воду, протекающую по трубе, то резистор можно представить в виде сужения в трубе, которое замедляет протекание воды (см. *разд. «Модель электрического тока: вода в трубе» на с. 4*). То есть резисторы ограничивают силу электрического тока.

Сопротивление измеряется в омах (и на схемах иногда представляется греческой буквой «омега»: Ω), разноцветные полосы на резисторах (см. рис. 2.5) обозначают номинальное значение сопротивления резистора. Описание системы разноцветных полос для обозначения номинального сопротивления резисторов и таблица значений полос приведены в *разд. «Полосатые резисторы» приложения*. Но для целей этой книги нам нужно определить только два разных значения резисторов: 330 Ом и 10 кОм. Резистор сопротивлением 330 Ом обозначается двумя оранжевыми

полосами и одной коричневой (см. рис. 2.5), а резистор сопротивлением 10 кОм — коричневой, черной и оранжевой полосами. Четвертая полоса на резисторе обозначает погрешность его действительного значения по отношению к номинальному. При этом серебряная четвертая полоска означает, что действительное значение сопротивления резистора будет в пределах 5%, а золотистая — в пределах 10% погрешности от указанного номинального. Для проектов этой книги не требуется большая точность, поэтому для резисторов далее будет указываться только их номинальное значение, что будет достаточно при любой погрешности.

Некоторые электронные компоненты, включая светодиоды, можно повредить, пропуская по ним слишком большой ток. Чтобы защитить такие компоненты от повреждения, протекающий через них ток ограничивают с помощью резисторов. Поэтому, в целях повышения уровня безопасности наших проектов, мы всегда будем подключать токоограничивающие резисторы последовательно светодиодам, чтобы ограничить протекающий по ним ток до безопасного уровня и избежать их перегорания или даже, в худшем случае, взрыва. (Да, слишком большой ток может действительно заставить компоненты буквально взрываться.)

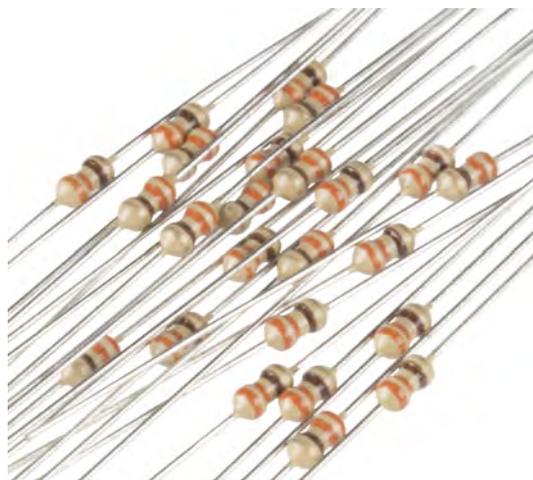


Рис. 2.5. Резисторы

ПОЧЕМУ В ПРОЕКТЕ «СВЕТОФОР» ИСПОЛЬЗУЮТСЯ РЕЗИСТОРЫ ЗНАЧЕНИЕМ 330 ОМ?

Средний светодиод красного цвета может выдержать нагрузку по току величиной около 20 мА. Обычно эти данные указываются в сопроводжающей компонент документации. Чтобы защитить светодиод от токов больших, чем допустимые, последовательно с ним подключается резистор. Но откуда мы знаем, что нужно использовать резистор сопротивлением 330 Ом?

Включенные выводы Arduino выдают напряжение величиной 5 В. Для включения светодиодов разного цвета требуются немного разные напряжения обычно в диапазоне от 2,0 до 3,5 В. Так, для включения красного светодиода требуется 2 В, что делает 3 В (при напряжении питания 5 В) лишними. Эти 3 В рассеиваются в виде тепла на резисторе или на любом другом компоненте, подключенном последовательно в цепи. Рекомендуемой практикой является ограничение тока светодиода до примерно половины максимально допустимого, что для красного светодиода с максимальным током 20 мА будет составлять 10 мА. Значение величины сопротивления

резистора для рассеивания 10 мА при напряжении 3 В можно рассчитать с помощью закона Ома (помните, что 10 мА = 0,01 А):

$$U = I \times R, \text{ откуда:}$$

$$R = \frac{U}{I} = \frac{3 \text{ В}}{0,01 \text{ А}} = 300 \text{ Ом}$$

Но значение 300 Ом не является стандартным значением резистора. Самым близким стандартным значением будет 330 Ом, и это вполне приемлемо. Использование резистора, подключенного последовательно в цепи светодиода, обеспечит долгое время бесперебойной работы светодиода. Поскольку этот резистор ограничивает ток, проходящий через светодиод, он и называется *токоограничивающим резистором*.

Если у вас есть под рукой резисторы разных значений, попробуйте использовать их со светодиодом и наблюдайте, что происходит. Резисторы с более высоким сопротивлением будут ограничивать ток в большей степени, а с меньшим — в меньшей. Что вы думаете будет, если вместо резистора сопротивлением 330 Ом использовать резистор 10 кОм?

Создаем прототип светофора

Теперь давайте приступим к созданию схемы светофора. Сначала рассмотрим его принципиальную схему, приведенную на рис. 2.6. Ее нужно смонтировать на беспаячной макетной плате, как показано на рис. 2.7.

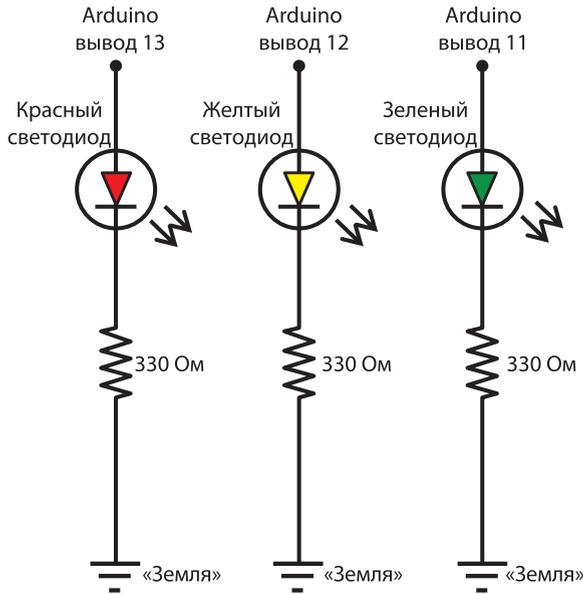


Рис. 2.6. Принципиальная схема проекта «Светофор»

Принципиальная схема (см. рис. 2.6) показывает электрическое подключение каждого компонента. Как можно видеть, выводы 13, 12 и 11 платы Arduino используются для управления отдельными светодиодами нашего проекта, при этом каждый светодиод подключен к отдельному резистору, который, в свою очередь, подключен к общему проводнику («земле»).

Примечание

Если вам необходимо освежить свои знания устройства и работы беспаячной макетной платы, вернитесь в *разд. «Создание прототипов схем»* на с. 5.

Подключаем красный светодиод

Теперь взглянем на монтажную схему проекта (см. рис. 2.7) и начнем преобразовывать принципиальную схему проекта в монтажную. В первом проекте мы заставляли мигать светодиод, встроенный в плату Arduino. Как мы помним, он подключен к выводу 13 на плате Arduino. Поскольку в этом проекте мы используем три светодиода, нам нужно самим выполнить их подключение к необходимым выводам платы. Следуя информации в принципиальной схеме (см. рис. 2.6) и в монтажной (см. рис. 2.7), подключите вывод 13 платы к положительному (длинному выводу) красного светодиода.

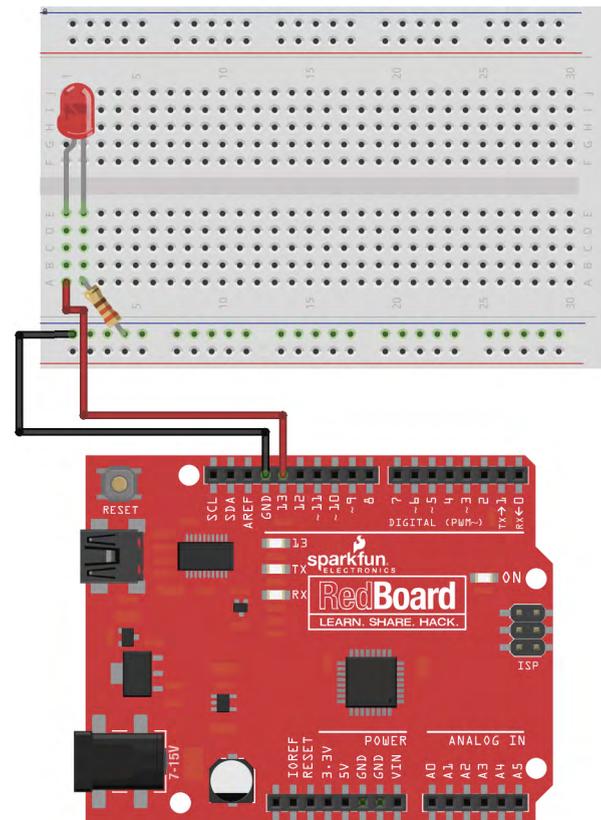


Рис. 2.7. Монтажная схема проекта «Светофор»: показано подключение красного светодиода, расположенного на монтажной плате, к выводу Arduino через токоограничивающий резистор

Чтобы выполнить это подключение, будет лучше сначала расположить плату Arduino рядом с беспаячной макетной платой именно так, как показано на рис. 2.7. (Такое размещение плат будет стандартным для всех проектов этой книги.)

Затем возьмите красный светодиод и резистор сопротивлением 330 Ом. Согните выводы резистора, как показано на рис. 2.8, чтобы его было легче вставить в гнезда беспаячной макетной платы. Рекомендуется при этом бокорезами обрезать выводы резистора наполовину, чтобы с ним было легче работать. В отличие от светодиодов, выводы резисторов не имеют полярности, поэтому не играет роли, какой вывод куда вставлять.

Горизонтальные и вертикальные ряды большинства макетных плат для удобства обозначаются соответственно цифрами и буквами (рис. 2.9). Ориентируясь на эту маркировку, вставьте светодиод в беспаячную макетную плату, как показано на рис. 2.7. Длинный (положительный, или анод) вывод вставляется в гнездо столбца **e** ряда **1** (то есть, в сокращенной записи, в гнездо **e1**) макетной платы, а короткий (отрицательный, или катод) вывод — в гнездо столбца **e** ряда **2** (**e2**).

Теперь очередь за резистором сопротивлением 330 Ом (с двумя оранжевыми и одной коричневой полосками). Вставляем один вывод резистора в любое гнездо в ряду **2** макетной платы, что подключает его к короткому (отрицательному) выводу светодиода (на рис. 2.7 этот вывод резистора вставлен в гнездо **a2** макетной платы). Помним при этом, что на всех стандартных беспаячных макетных платах гнезда столбцов **a** по **e** и **f** по **j** соединены между собой. Теперь вставьте другой вывод резистора в шину отрицательного питания макетной платы. Это будет столбец, обозначенный синей или черной линией и знаком минус (-).

Подаем питание на макетную плату

Возьмите две проволочные перемычки со штекерами на обоих концах. Рекомендуется, чтобы одна перемычка была черного цвета — для



Рис. 2.8. Сгибание выводов резистора

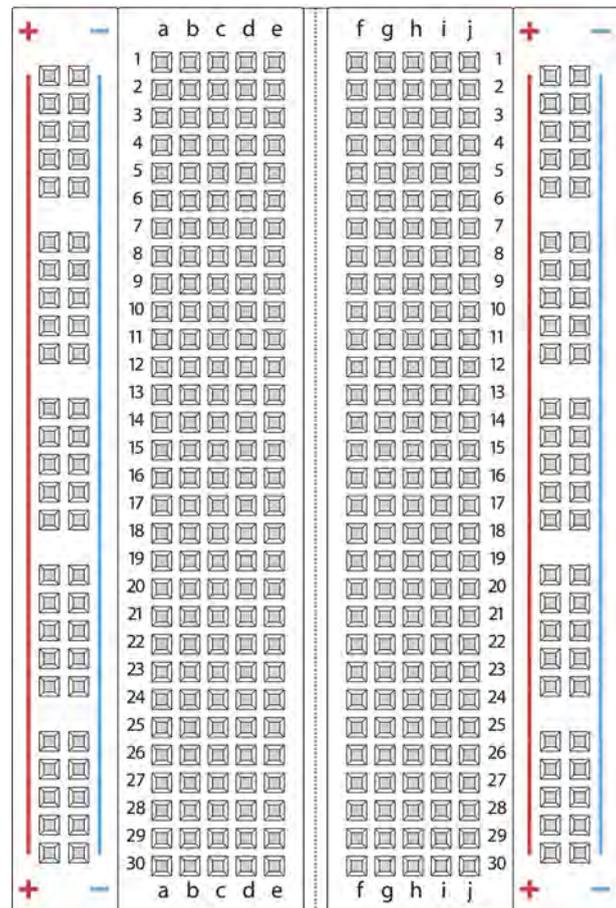


Рис. 2.9. Горизонтальные ряды гнезд беспаячной макетной платы обозначены цифрами, а вертикальные — буквами

отрицательного напряжения, а другая, для положительного напряжения, красного цвета. Этот подход будет применяться во всех проектах этой книги.

Вставьте один штекер черной перемычки в гнездо GND («земля») платы Arduino (на плате Arduino есть три гнезда, обозначенные GND, — можно использовать любое из них), а второй — в гнездо шины отрицательного питания на макетной плате. Питание на каждый светодиод будет подаваться с цифровых выводов платы. Поскольку вывод **13** будет подавать питание на красный светодиод, вставьте один конец перемычки в гнездо **13** на плате Arduino, а второй — в гнездо **a1** на макетной плате.

Подключите плату Arduino к порту USB компьютера — должен начать исполняться загруженный в нее ранее скетч Blink из первого проекта, в результате чего красный светодиод начнет мигать раз в секунду. В принципе, мигать должны оба светодиода: и тот, что на макетной плате, и тот, который встроен в плату Arduino, поскольку они оба подключены к одному и тому же выводу 13.

Если светодиод на макетной плате не мигает, тогда как встроенный светодиод мигает, проверьте надежность и правильность монтажной проводки, а также полярность подключения светодиода. Убедитесь при этом, что короткий вывод светодиода вставлен в гнездо во втором ряду макетной платы, и что резистор вставлен одним выводом в гнездо в этом же ряду, а вторым — в гнездо шины отрицательного питания.

Когда схема заработает должным образом, то есть светодиод на макетной плате будет мигать, отключите плату Arduino от компьютера, чтобы можно было смонтировать остальную часть схемы. Рекомендуем вам обзавестись хорошей привычкой всегда обесточивать плату при выполнении монтажа компонентов на ней.

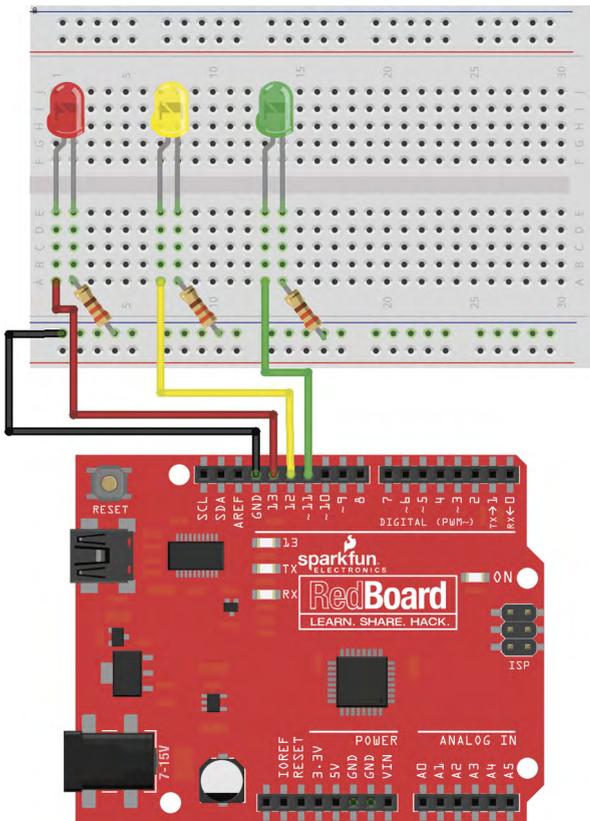


Рис. 2.10. Конечная монтажная схема прототипа проекта «Светофор»: светодиоды подключены к выводам 11, 12 и 13 платы Arduino

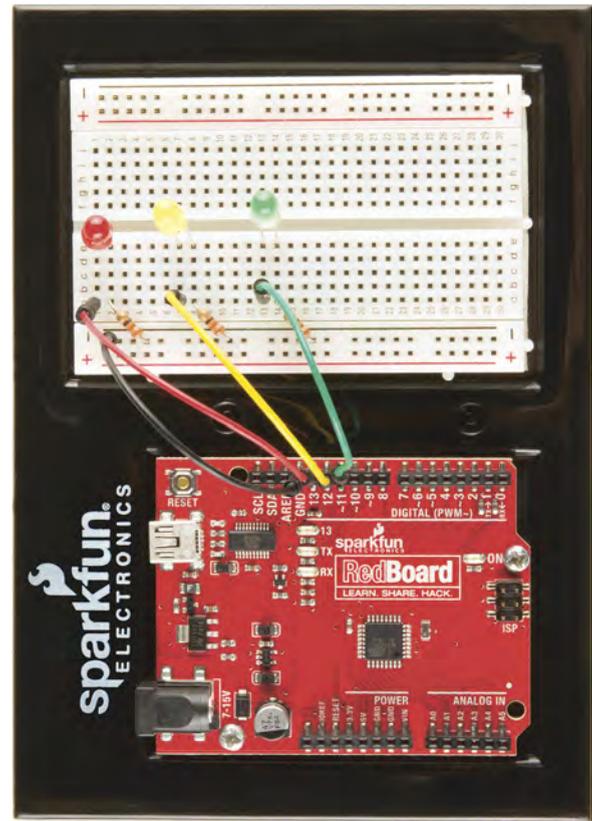


Рис. 2.11. Завершенный монтаж проекта «Светофор»: мы видим плату Arduino, светодиоды и резисторы

Добавляем желтый и зеленый светодиоды

Теперь подключите желтый светодиод к выводу 12 платы Arduino, а зеленый — к выводу 11. Следуйте тем же инструкциям, что и для подключения красного светодиода, но вставляйте каждый светодиод в отдельный ряд. К отрицательной шине питания макетной платы каждый светодиод, согласно принципиальной схеме, приведенной на рис. 2.6, подключайте через отдельный резистор. Конечный монтаж светодиодов показан на рис. 2.10. Обратите внимание, что возле каждого светодиода на макетной плате предусмотрено немного свободного пространства, чтобы компоненты не мешали выполнять монтаж.

Хотя здесь был изложен конкретный способ подключения компонентов на макетной плате, не

обязательно следовать ему. Можно использовать любую часть макетной платы, при условии, что соединяемые компоненты вставляются в гнезда одного и того же ряда. Готовая схема проекта должна выглядеть наподобие показанной на рис. 2.11.

Впрочем, кроме внешнего подобия настоящему светофору, нам нужно реализовать и функциональное ему подобие, для чего следует включать каждый светодиод на определенное время, а затем выключать его и включать следующий, и так далее. Плата Arduino, разумеется, способна использовать в программах инструкции разных типов, включая команды синхронизации для управления схемами, поэтому она справится с этой задачей без каких бы то ни было проблем.

Программируем светофор

Снова подключите плату Arduino к компьютеру, чтобы начать программировать скетч светофора. Откройте среду разработки Arduino, чтобы создать новый скетч.

Проверьте параметры среды разработки

Прежде чем начинать создавать новый скетч, желательно всегда сначала выполнить определенные организационно-подготовительные работы. Итак, первым делом проверьте, что установлены правильные тип платы (**Board type**) и порт (**Port**). Для этого выполните последовательность команд меню **Tools | Board**. Если вы используете плату SparkFun RedBoard или стандартную плату Arduino Uno, выберите опцию **Arduino/Genuino Uno**. Затем выполните последовательность команд меню **Tools | Port**. В компьютере под Windows плате Arduino должен быть присвоен порт COM с самым большим значением. Для компьютеров Mac или Linux последовательный порт будет отображаться в формате `/dev/cu.usbserial-A<xxxx>`, где

фрагмент `<xxxx>` является строкой произвольных символов, уникальной для вашего Arduino.

Создаем переменные для номеров выводов

Удостоверившись в правильности параметров среды разработки, можно приступить к созданию скетча. Как пояснялось в *разд. «Анатомия скетча Arduino» проекта 1*, базовый скетч Arduino состоит из двух частей: функции `setup()` и функции `loop()`. Большинство простых скетчей обычно будут иметь такую базовую структуру, но более сложные скетчи содержат много разных составляющих. Вот и в скетче Stoplight («Светофор») присутствует новый раздел, называющийся *глобальным пространством имен*. Этот раздел расположен перед функцией `setup()` и не относится ни к одной функции. В этом разделе, среди прочего, определяются переменные, которые служат контейнерами для значений, предоставляя эти значения в любом месте скетча, где они требуются. В скетчах Arduino можно использовать значения (данные) нескольких типов.

Типы данных

В языке Arduino используются несколько разных типов данных, некоторые из них вам придется применять в своих скетчах довольно часто. Далее приводится список основных типов данных, их употребляемые в коде названия и краткое описание.

- **Целое число (int)** — целые числа в диапазоне от -32 768 до 32 767.
- **С плавающей запятой (float)** — числа с десятичной дробью в диапазоне от -3,4028235E+38 до 3,4028235E+38.
- **Байт (byte)** — целые числа в диапазоне от 0 до 255.
- **Символ (char)** — символ, обозначаемый заключением в одинарные кавычки, например, 'a'.
- **Строка (string)** — последовательность символов, обозначаемая заключением в двойные кавычки, например, «привет».
- **Булево (Boolean)** — значение, которое может быть или истиной или ложью, что соответствует значениям 1 или 0 в скетче и ВЫСОКИЙ или НИЗКИЙ в переводе на уровни напряжения выводов платы Arduino.

При определении переменной в Arduino необходимо указывать ее тип данных. Этот процесс рассматривается далее, в *разд. «Меняющиеся значения»*.

Меняющиеся значения

Большинство значений, используемых в скетчах, оформлены в виде *переменных*. Переменную можно представить как многоцветный контейнер для единицы данных, которая может быть цифрой, буквой или даже целым предложением.

Но прежде чем использовать переменную, ее необходимо *объявить*. Объявление переменной состоит в присвоении ей имени, указании ее типа данных и присвоении ей исходного значения. Рекомендуем вам присваивать переменной

значение сразу при ее объявлении. Этот процесс выглядит следующим образом:

```
.....
1 int 2 val = 3 10;
.....
```

Объявление переменной состоит из трех частей: типа данных переменной **1**, ее имени **2** и ее значения **3**. Строка кода объявления переменной завершается точкой с запятой, которая указывает среде разработки конец команды языка Arduino. Использование точки с запятой в конце команды является обязательным. Упущение этой детали часто является причиной многих ошибок компилятора, или, как они *tot* называются, багов². Поэтому всегда будьте внимательны, чтобы не допускать эту ошибку.

В имени переменной можно использовать любую последовательность символов без пробелов, включая буквы и цифры. При этом, для первого символа имени переменной использовать цифры или специальные символы нельзя. Рекомендуется делать имена переменных как можно более описательными и в то же время как можно более короткими. При решении этой задачи следует придумывать описания и сокращения, проявлять максимальную изобретательность. В данном примере переменная называется *val* (сокращение от английского *value* — значение), переменная инициализируется (ей присваивается начальное значение) значением 10. Инициализация переменной при ее объявлении не является обязательной, но выработать у себя такую привычку полезно.

В нашем проекте создаются три переменных для хранения номеров выводов, к которым будут подключены светодиоды, управляемые посредством Arduino. Работать с переменной, которая описывает цвет светодиода, намного легче и удобнее, чем пытаться запомнить, какой светодиод подключен к какому выводу.

Итак, создайте новый скетч и вставьте код из листинга 2.1 в его глобальное пространство имен.

² От англ. *bug* — жучок. По легенде, от моли, застрявшей между контактами реле первого компьютера, что вызвало ошибку в его работе.

Листинг 2.1. Переменные для номеров выводов Arduino

```
byte redPin = 13;
byte ylwPin = 12;
byte grnPin = 11;
```

Примечание

Для удобства чтения при создании имен переменных используется верблюжий регистр, вследствие чего буква P в слове pin (вывод) пишется прописной. Верблюжий регистр позволяет разделять слова имени переменной без использования пробелов.

В этих трех объявленных нами переменных хранятся номера выводов, к которым подключаются светодиоды. Поскольку номера выводов в Arduino ограничены целыми числами в диапазоне от 0 до 13, мы используем тип данных `byte`. Использование этого типа данных возможно по той причине, что мы знаем, что номер вывода будет всегда меньше, чем 255. Обратите внимание, что имя каждой переменной описывает ее содержимое: `redPin` (красныйВывод) обозначает вывод Arduino для красного светодиода, `ylwPin` (желтыйВывод) — вывод для желтого светодиода и `grnPin` — вывод для зеленого светодиода. И, как можно видеть на рис. 2.10, номер вывода для красного светодиода — 13, для желтого — 12, а для зеленого — 11. Теперь в любое время, когда нам потребуется задействовать в нашем скетче номер вывода, мы можем использовать для этого описательное имя переменной.

Создаем функцию `setup()`

Продолжаем разработку скетча `Stoplight` («Светофор») добавлением в него функции `setup()`, текст которой приводится в листинге 2.2.

Подобно скетчу «Здравствуй, мир!» в *проекте 1* (см. функцию `setup()` в листинге 1.2), в скетче `Stoplight` («Светофор») функция `setup()` также выполняет конфигурирование цифровых выводов Arduino посредством функции `pinMode()`.

Листинг 2.2. Код функции `setup()` для проекта `Stoplight` («Светофор»)

```
void setup()
{
    //красный светодиод
    pinMode(redPin, OUTPUT);
    //желтый светодиод
    pinMode(ylwPin, OUTPUT);
    //зеленый светодиод
    pinMode(grnPin, OUTPUT);
}
```

Поскольку в этом проекте задействуются три разных цифровых вывода, функция `pinMode()` вызывается в скетче три раза. В каждом отдельном вызове функции передаются в качестве параметров переменная номера вывода ❶ (`redPin`, `ylwPin` и `grnPin`) и константа функции вывода платы `OUTPUT` ❷. Константа режима вывода данных `OUTPUT` используется потому, что наш скетч управляет светодиодами, которые являются устройствами *вывода* данных. С устройствами *ввода* данных мы познакомимся в *проекте 4*.

Создаем функцию `loop()`

Далее создадим функцию `loop()`. Настоящие светофоры в цикле последовательно зажигают красный, зеленый, желтый огни, поэтому в этом проекте мы будем делать то же самое. Скопируйте код из листинга 2.3 и вставьте его в свой скетч на место функции `loop()`.

Во избежание путаницы и аварий при проходе на светофор в коридоре, скетч `Stoplight` («Светофор») одновременно зажигает только один цвет светофора. Поэтому, когда включается очередной светодиод, все другие светодиоды должны быть выключенными. Например, если нужно зажечь красный свет, мы сначала вызываем функцию `digitalWrite(redPin, HIGH)`, а затем функции `digitalWrite(ylwPin, LOW)` и `digitalWrite(grnPin, LOW)`. Параметр `HIGH` в первом вызове функции

Листинг 2.3. Код функции loop() для проекта Stoplight («Светофор»)

```
void loop()
{
    //включаем красный свет
    digitalWrite(redPin, HIGH);
    digitalWrite(ylwPin, LOW);
    digitalWrite(grnPin, LOW);
    delay(2000);

    //включаем зеленый свет
    digitalWrite(redPin, LOW);
    digitalWrite(ylwPin, LOW);
    digitalWrite(grnPin, HIGH);
    delay(1500);

    //включаем желтый свет
    digitalWrite(redPin, LOW);
    digitalWrite(ylwPin, HIGH);
    digitalWrite(grnPin, LOW);
    delay(500);
}
```

переводит вывод redPin (вывод 13) платы в высокое состояние напряжения, включая подключенный к нему красный светодиод, а параметр LOW в двух последующих вызовах переводит выводы ylwPin и grnPin (выводы платы 12 и 11 соответственно) в низкое состояние напряжения, включая желтый и зеленый светодиоды. Поскольку Arduino работает на рабочей тактовой частоте 16 МГц (исполняя приблизительно одну инструкцию каждые 16 миллионных секунды), задержка между этими командами составляет порядка нескольких микросекунд. То есть эти команды исполняются настолько быстро, что для всех практических целей можно считать, что они исполняются одновременно. Наконец, обратите внимание на функцию delay(2000). Эта функция приостанавливает исполнение скетча и удерживает красный

светодиод включенным в течение 2000 мс или 2 секунд, прежде чем исполнится следующая инструкция.

Код для желтого и зеленого светодиодов работает по такому же принципу, устанавливая высокий уровень напряжения на соответствующем выводе платы и низкий на остальных и приостанавливая исполнение скетча на разные периоды времени. Попробуйте поэкспериментировать со значениями времени задержки в своем скетче Stoplight, соответствующими интенсивности движения в вашем коридоре. Помните, что значения, передаваемые функции delay(), устанавливают период времени в миллисекундах, в течение которого вы хотите удерживать светодиод во включенном состоянии.

Загружаем скетч в Arduino

Полный текст кода скетча для проекта Stoplight («Светофор») приведен в листинге 2.4. Вы до сих пор вводили в редактор код скетча частями (см. листинги 2.1–2.3). Теперь внимательно проверьте, что он соответствует коду в листинге 2.4, сохраните скетч, а затем загрузите его в Arduino, выполнив последовательность команд меню **Sketch | Upload** (Скетч | Загрузить) или нажав комбинацию клавиш <Ctrl>+<U>. В случае вывода средой разработки каких-либо сообщений об ошибках, снова проверьте свой код и убедитесь, что он в точности соответствует коду в листинге 2.4. Это означает, что он точь-в-точь повторяет написание каждого слова, использование заглавных букв и знаков препинания, не забудьте и о точке с запятой в конце каждой инструкции.

Когда все сделано правильно, светодиоды должны включаться и выключаться в цикле, подобном циклу работы настоящего светофора: сначала красный, за ним зеленый, а затем желтый на короткое время, после чего функция loop() начинает исполняться повторно, снова включая красный светодиод и т. д. Скетч должен работать таким образом бесконечно — до тех пор, пока на плату Arduino подается питание.

Листинг 2.4. Полный текст кода скетча для проекта Stoplight («Светофор»)

```

byte redPin = 13;
byte ylwPin = 12;
byte grnPin = 11;

void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(ylwPin, OUTPUT);
  pinMode(grnPin, OUTPUT);
}

void loop()
{
  //включаем красный свет
  digitalWrite(redPin, HIGH);
  digitalWrite(ylwPin, LOW);
  digitalWrite(grnPin, LOW);
  delay(2000);

  //включаем зеленый свет
  digitalWrite(redPin, LOW);
  digitalWrite(ylwPin, LOW);
  digitalWrite(grnPin, HIGH);
  delay(1500);

  //включаем желтый свет
  digitalWrite(redPin, LOW);
  digitalWrite(ylwPin, HIGH);
  digitalWrite(grnPin, LOW);
  delay(500);
}

```

Делаем светофор автономным

Когда плата Arduino подключена к компьютеру, питание на нее подается из порта USB компьютера. Но что, если мы хотим поместить наш проект в таком месте, куда не достает кабель подключения, или вообще взять его на какое-либо мероприятие, чтобы продемонстрировать свое творение друзьям? Для этого нам потребуется переносной источник питания, а именно портативный блок батареек. Помните, что плата Arduino оснащена цилиндрическим гнездовым разъемом питания, а также встроенным стабилизатором напряжения. Это делает возможным подключение платы к внешнему источнику питания напряжением в диапазоне от 6 до 18 В. На рынке предлагаются разнообразные держатели батареек, но авторам нравится показанный на рис. 2.12 держатель для четырех батареек типоразмера AA, который мы используем во многих наших проектах.



Рис. 2.12. Держатель на 4 батарейки AA с цилиндрическим разъемом

Извлеките кабель USB из разъемов компьютера и платы, вставьте батарейки в держатель батареек, а затем вставьте разъем держателя в разъем внешнего питания платы Arduino, как показано на рис. 2.13. Если батарейки заряжены, Arduino начнет исполнять загруженный в него скетч. При этом вы можете перемещать платы проекта куда угодно или же встроить их в модель светофора.



Рис. 2.13. Делаем проект «Светофор» автономным, используя держатель батареек в качестве источника питания

Теперь мы готовы перейти к реализации следующего уровня проекта, и далее, в разд. «Создаем корпус для светофора», мы рассмотрим, как преобразовать этот прототип в модель светофора, который можно будет установить в местах с интенсивным движением в вашем доме.

Создаем корпус для светофора

Когда Arduino более не привязан к компьютеру, любой электронный проект на его основе можно встроить в постоянный корпус. Три мигающих светодиода на макетной плате лишь с трудом можно представить в виде светофора. Для большей реалистичности надо вставить платы проекта в соответствующий корпус и оснастить светодиодами

линзами, которые сделают их лучше видимыми на расстоянии. Конечно, если вы хотите остановиться только на прототипе схемы, корпус и линзы делать не обязательно, но мы все же рекомендуем вам приобрести новый опыт изготовления корпусов для своих проектов.



Рис. 2.14. Корпус для модели светофора из картона и мячиков для настольного тенниса

Здесь мы покажем вам, как создать реалистично выглядящую модель светофора из гофрированного или открыточного картона, но можно использовать любой имеющийся под рукой материал. Проявляйте изобретательность! Наш пример, показанный на рис. 2.14, сделан из гофрированного картона, мячиков для настольного тенниса и некоторого умения работать руками.

Вы можете сделать корпус светофора сами, используя предоставленные здесь инструкции в качестве руководства. Впрочем, если вы хотите в точности воспроизвести этот проект, загрузите файл ZIP-архива, содержащий шаблоны и скетчи, по адресу: <https://www.nostarch.com/arduinoinventor/>. Все проекты в этой книге сопровождаются шаблонами, которые можно распечатать, перевести на картон и вырезать с помощью макетного ножа и металлической линейки.

Примечание

Если вам повезло, и у вас есть доступ к резальной машине наподобие Cricut, Silhouette Cameo или лазерному резаку, эти файлы можно легко приспособить под эти инструменты.

Распакуйте из сопровождающего книгу архива файлы проекта 2 (папка *P2_Stoplight*) и распечатайте файл шаблона *P2_StopLightTemplate.svg* в полный размер, чтобы его можно было использовать в качестве образца для вырезания корпуса. Затем вооружитесь предметами из разд. «Прочие инструменты и материалы» этого проекта и приступайте к работе над корпусом.

Делаем картонный корпус

Сначала вырежьте из распечатки шаблоны деталей корпуса, показанные на рис. 2.15. Корпус светофора вырезается из одного цельного куска картона, а затем сгибается по слегка прорезанным в картоне линиям.

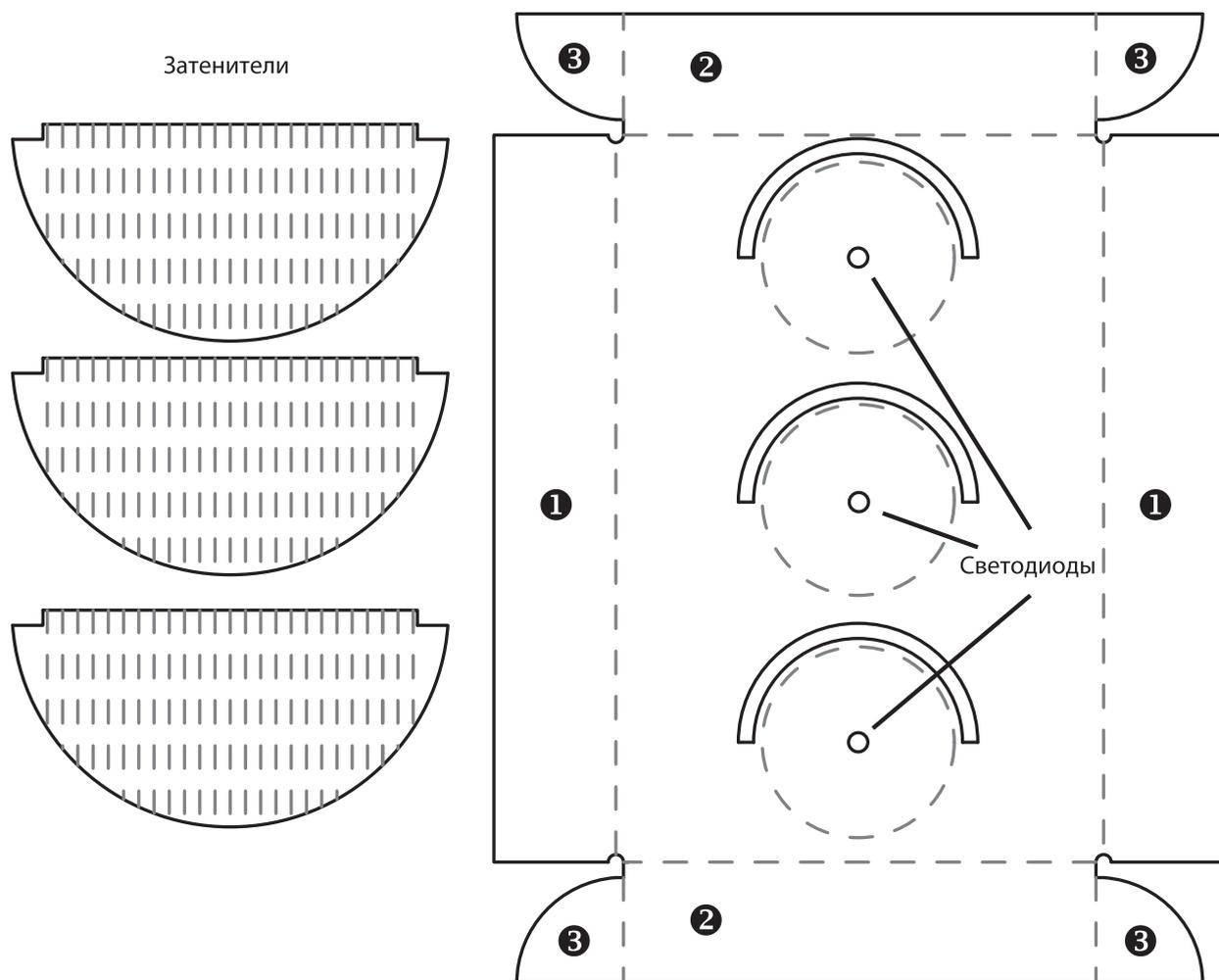


Рис. 2.15. Шаблон для корпуса модели светофора (в уменьшенном виде)



Рис. 2.16. Надрезание линий изгиба корпуса с помощью макетного ножа и металлической линейки



Рис. 2.17. Сверление отверстий под светодиоды



Рис. 2.18. Светодиоды, вставленные в заготовку корпуса модели светофора

Переведите сначала контуры шаблона на картон, а также наметьте пунктирные линии, используя для этого ручку или карандаш другого цвета. По этим линиям нужно будет сделать в картоне неглубокие прорезы, чтобы затем по ним можно было согнуть корпус. Но пока не прорезайте их.

Обозначив контуры шаблона корпуса на картоне, вырежьте корпус по контуру с помощью макетного ножа и металлической линейки (рис. 2.16). Если вы никогда раньше не работали с макетным ножом, обязательно сначала прочитайте *врезку «Безопасная работа с макетным ножом»* чуть далее в этом проекте. Слегка надрежьте вырезанный корпус по пунктирным линиям на внешней стороне корпуса. Сделайте пару неглубоких проходов ножом, чтобы можно было с легкостью согнуть корпус по этим линиям, но не прорезайте картон насквозь. Также, пока не вырезайте затенители.

Вырезав контур корпуса и надрезав его линии изгиба, сделайте в нем отверстия для светодиодов в местах, обозначенных на шаблоне маленькими сплошными точками внутри больших пунктирных кругов. Можно, конечно, просто проколоть отверстия остро заточенным карандашом. Но чтобы получить более аккуратные отверстия, рекомендуется высверлить их с помощью электродрели и сверла диаметром чуть меньше 5 мм (рис. 2.17).

Дело в том, что диаметр светодиодов практически равен 5 мм. Но нам нужны отверстия чуть меньшего диаметра, в которые диоды можно легко вставить, причем так, чтобы они не выпадали из них. С этой задачей отлично справится сверло диаметром около 4,75 мм.

Будьте осторожны при сверлении отверстий под светодиоды — в частности, смотрите, чтобы ваши пальцы с тыльной стороны картона не располагались под сверлом. Дырка в одном из них вам будет совсем ни к чему. Если у вас нет дрели или вы не уверены в своих способностях использовать ее, отверстия можно также просверлить без дрели — одним сверлом, вращая его пальцами.

Просверлив отверстия, извлеките светодиоды из макетной платы и вставьте их в просверленные

БЕЗОПАСНАЯ РАБОТА С МАКЕТНЫМ НОЖОМ

Вам придется часто использовать макетный нож (рис. 2.19) при работе над проектами этой книги, поэтому важно знать, как с ним безопасно работать. Как и любой другой инструмент, при неправильном использовании макетный нож может нанести телесные повреждения.



Рис. 2.19. Макетный нож

Далее приводятся несколько советов по безопасной работе с макетным ножом.

- При резке листовых материалов всегда тяните лезвие на себя. При толкании лезвия от себя или в любом другом направлении оно может соскользнуть или сломаться.
- Не спешите! Не пытайтесь прорезать материал через всю его толщину за один проход. Сделайте несколько проходов, прилагая среднее давление на лезвие. Это будет более щадящим для лезвия, а также позволит получить более аккуратные резы.
- Используйте прямую металлическую направляющую для лезвия — например, металлическую линейку. Использование деревянной или пластмассовой направляющей чревато возможностью врезания лезвия в направляющую с последующим отскакиванием от нее, возможно, в направлении вашей руки и в саму руку.
- Не держите пальцы на пути лезвия. При всей очевидности этого совета, люди умудряются порезаться именно таким образом.
- Не пытайтесь поймать нож, который начал скатываться со стола, — позвольте ему упасть, а затем просто поднимите с пола. При попытке поймать его прежде, чем он упадет, вы подвергаете себя риску уколоть лезвием руку. Это может быть очень больно.
- Наконец, используйте новые, острые и неповрежденные лезвия. Немедленно заменяйте сломавшиеся, а также затупившиеся лезвия. Резка бумаги и картона затупляет лезвия очень быстро. Всегда имейте под рукой запасные лезвия, и когда резать становится трудно, замените лезвие новым.

отверстия с тыльной стороны заготовки корпуса модели светофора, как показано на рис. 2.18. Помните, что в настоящем светофоре цвета идут в порядке: красный, желтый, зеленый — сверху

вниз. Запомните, как эти светодиоды были подключены к макетной плате, поскольку чуть позже их нужно будет снова к ней подключить.

Теперь согните заготовку корпуса модели светофора вдоль надрезанных линий, как показано на рис. 2.20: сначала согните внутрь вертикальные стороны ❶, а затем верхнюю и нижнюю стороны ❷ и язычки ❸ (нумерация частей заготовки корпуса светофора приведена на рис. 2.15).

Разместите язычки ❸ на внутренней стороне вертикальных сторон ❶, а затем приклейте их, как показано на рис. 2.21. Для этого можно использовать клеевой пистолет, скотч-ленту или



Рис. 2.20. Сгибание сторон корпуса светофора по надрезанным линиям



Рис. 2.21. Гибка и склейка картонного корпуса модели светофора

обычный канцелярский клей. Клеевой пистолет будет предпочтительней, поскольку это более удобно — клей затвердевает быстрее и держится довольно крепко.

Когда вы приклеите все язычки, у вас должна получиться неглубокая прямоугольная коробка, открытая с тыльной стороны.

Делаем линзы для светофора

Линзы для светодиодов светофора делаются из половинок мячиков для настольного тенниса, но можно взять и любой другой более или менее прозрачный материал.

Если используются мячики для настольного тенниса, аккуратно разрежьте их пополам. Для этого разместите мячик на коврик для резки или на толстом куске картона и крепко держите его пальцами с обеих сторон. Осторожно проколите мячик лезвием ножа сверху вниз (при этом будьте внимательны, чтобы лезвие не было направлено на удерживающую мячик руку) и начните его разрезать. Для этого понемногу вращайте мячик, разрезая его в процессе поворота, повторяя вращение, пока полностью не разрежете мячик. Непременно держите пальцы вдали от лезвия и всегда режьте на коврике для резки или на куске картона.



Рис. 2.22. Безопасный способ разрезки мячика для настольного тенниса



Рис. 2.23. Приклеивание половинок теннисных мячиков к корпусу светофора

Когда у вас образуются три половинки теннисного мячика (вообще-то, их будет четыре, но четвертую можно будет использовать для будущих проектов или сделать из нее шляпку для вашей любимой мягкой игрушки), приклейте их к корпусу светофора каплей клея, как показано на рис. 2.23.

Делаем затенители

Наконец, делаем и вставляем в светофор затенители. Чтобы получить хороший, правильный изгиб, нарежьте заготовки затенителей несколькими параллельными резами с интервалом между ними около 3 мм (рис. 2.24). На шаблоне затенителей (см. рис. 2.15) имеются соответствующие линии, по которым можно делать такие надрезы.

Согните надрезанные затенители в правильную дугу, как показано на рис. 2.25.

Вставьте согнутые затенители в корпус светофора — в отверстия над каждой линзой светодиода (рис. 2.26), а затем проклейте место соединения. Чтобы получить еще более реалистичную модель светофора, корпус можно окрасить пульверизатором в черный цвет. При этом обязательно удалите линзы или закройте их защитной липкой лентой, чтобы на них не попала краска.

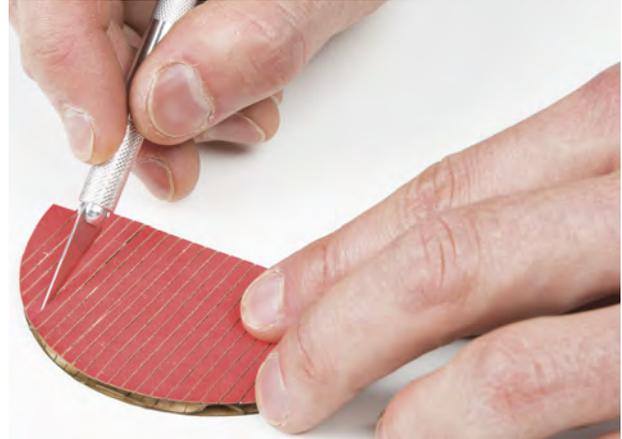


Рис. 2.24. Надрезание затенителей

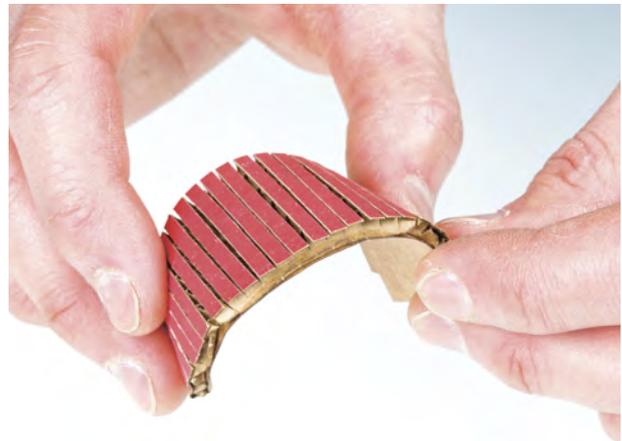


Рис. 2.25. Сгибание затенителя в дугу

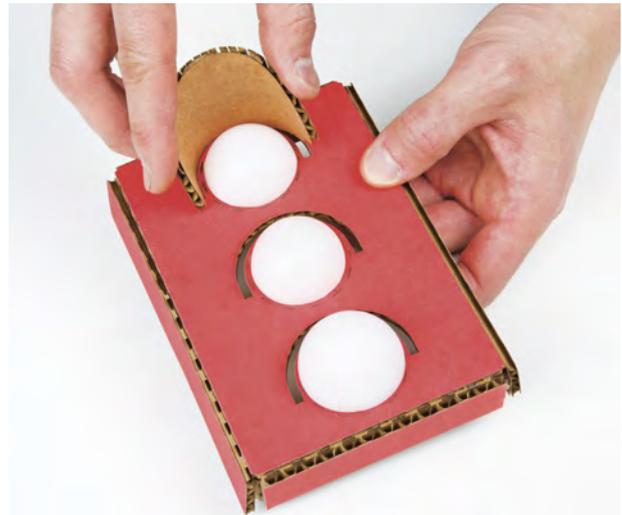


Рис. 2.26. Вставка затенителя в корпус

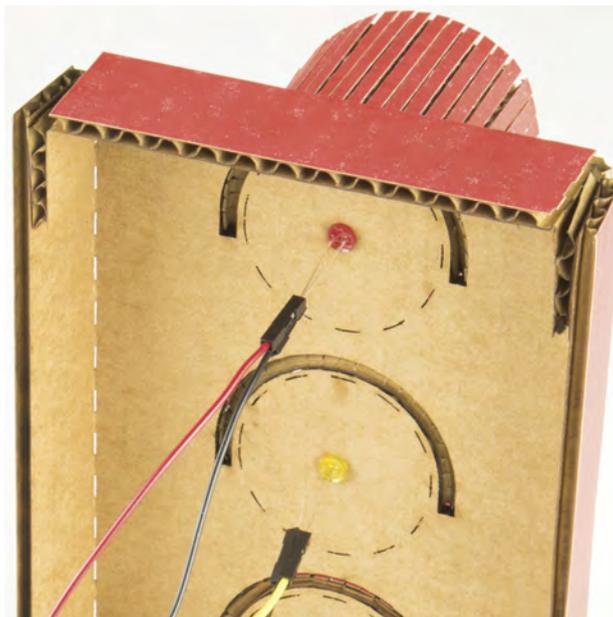


Рис. 2.27. Удлинение выводов светодиодов с помощью проволочных перемычек

Вставляем светодиоды и подключаем Arduino

Все, что нам осталось сделать, — это подключить ранее вставленные в корпус светофора светодиоды к Arduino. Чтобы подключать их было удобнее, необходимо удлинить выводы светодиодов. Для этого мы воспользуемся проволочными перемычками с гнездовым разъемом на одном конце и штыревым на другом (SparkFun PRT-09385 или самодельные). Всего нам нужно будет шесть таких перемычек. Выводы светодиодов мы удлиним, просто надевая гнездовые разъемы перемычек на выводы светодиодов. Чтобы не было путаницы с проводами, используйте перемычки одного цвета (например, черного) для отрицательных (коротких) выводов светодиодов и какого-либо отличного от него цвета — для положительных (длинных) выводов (рис. 2.27).

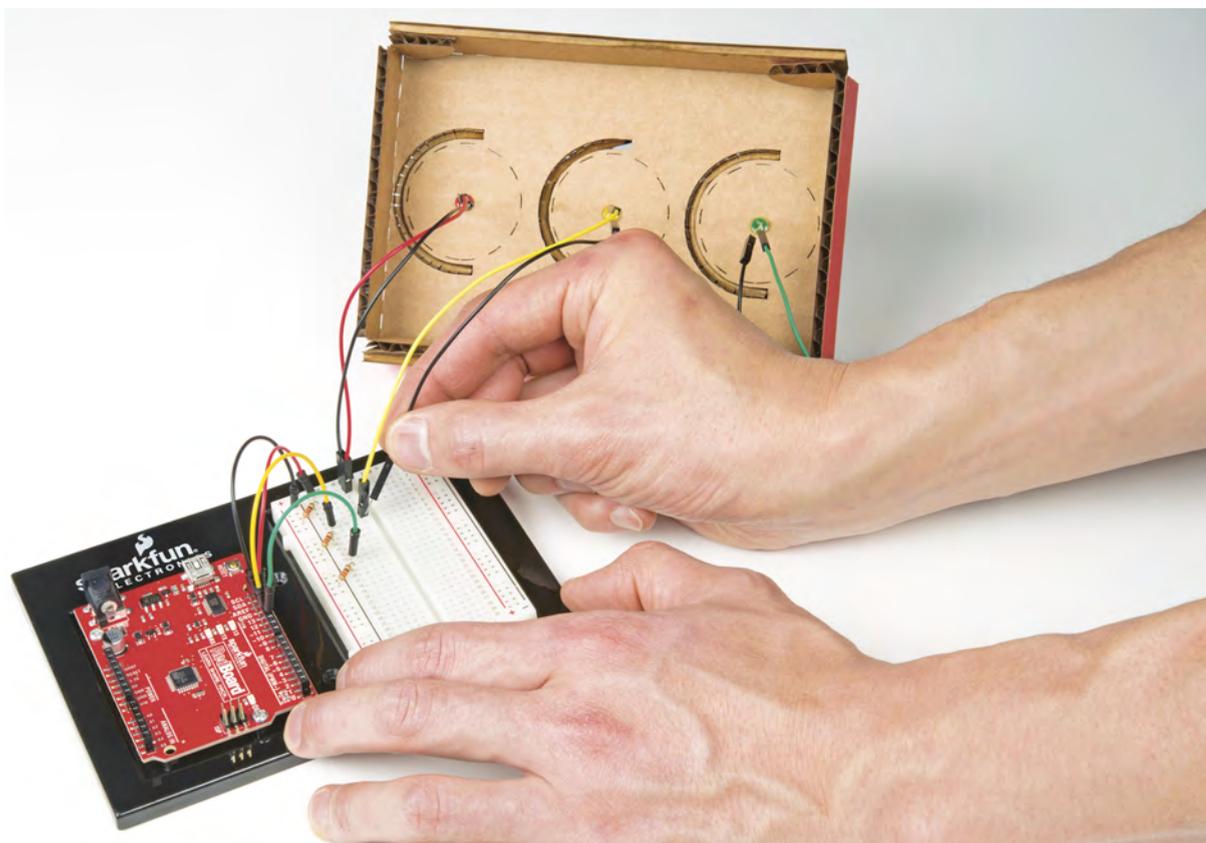


Рис. 2.28. Подключение удлинителей светодиодов к макетной плате

Затем подключаем другие концы проволочных перемычек к плате Arduino, вставляя штыревые разъемы от каждого светодиода в гнездо на плате, в которое этот светодиод был ранее вставлен (рис. 2.28). Опять же, будьте внимательны, чтобы не перепутать, какой светодиод подключать в какое гнездо. Если не помните, то сверьтесь с монтажной схемой на рис. 2.10.

Проверьте надежность соединений, подключив Arduino к компьютеру или к блоку батареек. Если какой-либо из светодиодов не включается, попробуйте пошевелить разъемы перемычек в местах подключения, а также еще раз убедитесь, что они вставлены в правильные гнезда на макетной плате. Модель светофора в ее законченном виде показана на рис. 2.29.

Плату Arduino вместе с макетной платой можно или оставить вне корпуса светофора, или же прикрепить их внутри с помощью клея или двусторонней липкой ленты. В любом случае включите питание своего светофора и установите его на каком-либо перекрестке с интенсивным движением, чтобы сделать его безопасным.

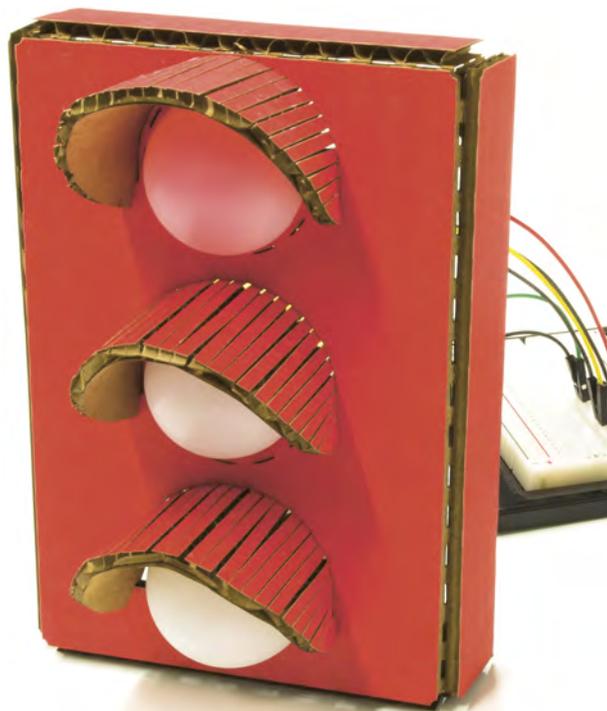


Рис. 2.29. Завершенный проект светофора

Идем дальше...

Возможностям, с которыми мы познакомились в процессе реализации проекта «Светофор» (Stoplight), — таким как синхронизация (тактирование) выходного сигнала для управления светодиодами, можно найти разнообразное применение как в доме в частности, так и в жизни в целом. Далее приводится несколько рекомендаций по другим применениям возможностей проекта «Светофор».

Экспериментируем с кодом

Основной возможностью, использованной нашим светофором, является отсчет времени с помощью таймера. Где еще можно применить таймер? Как насчет модифицирования кода, чтобы помочь с правильной выдержкой времени для варки яиц?

В таком случае красный светодиод остается включенным, пока яйцо еще сырое, желтый загорается, когда оно почти готово, а зеленый — когда полностью сварено.

Точные значения этих временных интервалов вам нужно будет установить самим, в зависимости от своих предпочтений касательно степени готовности вареных яиц. На длительность интервалов также будут оказывать влияние несколько переменных — таких как сила пламени, размер емкости для варки, количество воды в ней, а также размер яйца. Вам самому придется разобраться, как принять их все во внимание.

Для указания задержек в коде надо будет использовать большие числа, поскольку задаются они в миллисекундах. Вспомним, что 1 секунду



ПРИЛОЖЕНИЕ. Дополнительные практические сведения по электронике

В этом приложении рассматривается использование мультиметра и паяльника, а также предоставляется информация о том, как расшифровывать значение сопротивления резисторов по их цветным полосам.

Электрические измерения с помощью мультиметра

Мультиметр (тестер) является одним из незаменимых инструментов для диагностирования и поиска и устранения неисправностей в электронных схемах. Как можно понять по его названию, это прибор, способный измерять различные параметры электрических цепей, а именно ток, напряжение, сопротивление и неразрывность.

Теперь давайте разберемся, как с мультиметром работать. Для наглядной демонстрации мы взяли мультиметр VC830L компании SparkFun (TOL-12966), показанный на рис. П.1, но рассматриваемые приемы работы относятся к большинству мультиметров этой категории.

Функциональные части мультиметра

Мультиметр содержит три основные части: дисплей, ручку выбора функций и разъемы (рис. П.1).

Дисплей обычно может отображать четыре цифры и знак минус. Ручка выбора функции позволяет выбирать режим измерения определенного параметра электрической цепи: ток (мА), напряжение (В) или сопротивление (Ом). Каждый режим измерений разбит на несколько диапазонов с разными максимальными значениями.



Рис. П.1. Типичный мультиметр

В некоторых мультиметрах единицы измерения не отображаются. В таких случаях предполагается, что отображаемые значения измеряются в таких же единицах, как и выбранный максимальный предел данного режима измерений. Например, если указатель выбора режима установлен на максимальный предел 200 Ω (200 Ом), то отображаемые значения измеряются в омах. А если указатель установлен на 2, 20 или 200 k Ω , то отображаемые значения измеряются в килоомах.

Большинство мультиметров содержат в комплекте два щупа, которые вставляются в два из трех гнездовых разъемов на передней панели мультиметра. Эти разъемы обозначены: COM, mA Ω и 10A. Маркировка COM означает *общий*¹ (провод), и вставляемый в это гнездо щуп всегда подключается к «земле» или отрицательному полюсу питания схемы. В гнездо mA Ω вставляется щуп для измерения тока (mA — мА) до 200 мА, напряжения (V — В) и сопротивления (Ω — Ом). В гнездо 10A вставляется щуп для измерения токов, превышающих 200 мА.

Большинство щупов для мультиметров оснащены стандартным штекером типа «банан» на конце, который вставляется в гнездо мультиметра. Другой конец стандартного щупа оснащается штыревым пробником, но также имеются щупы с зажимом типа «крокодил», клипсой или иным типом наконечника. Для большинства измерений черный щуп вставляется в гнездо COM, а красный — в гнездо mA Ω .

Определение неразрывности электроцепи

Этот режим измерений обозначается на мультиметре символом, показанным на рис. П.2.

Определение неразрывности электроцепи является, пожалуй, самой важной функцией при

¹ От англ. Common — общий.

поиске и устранении неисправностей схем. Эта функция позволяет выполнить проверку электрической цепи на отсутствие разрывов в ней, а также убедиться в наличии или отсутствии электрических соединений.

Проверка выполняется подключением одного щупа к одному концу цепи, а другого щупа — к другому. При отсутствии разрывов встроенный в мультиметр зуммер будет издавать непрерывный тон. В ранних версиях тестеров вместо тона использовался звонок, и поэтому проверка на неразрывность цепи в профессиональных кругах называется *прозвонкой* электроцепи. С помощью этой функции можно проверить, какие гнезда макетной платы соединены между собой, а какие нет.

Диаметр щупов обычно слишком велик, чтобы вставить их непосредственно в гнезда макетной платы, но эта проблема решается просто: сначала провода вставляются в гнезда платы, а уже к проводам прижимаются пробники. Эту функцию также можно использовать для трассировки проводников схемы, приставив пробники к противоположным концам проводника. При проверке неразрывности цепи не имеет значения, какой щуп подключать к какому концу цепи, поскольку проверяется лишь сам факт соединения этих концов между собой электрически.

Измерение сопротивления

Этот режим измерений обозначается на мультиметре символом, показанным на рис. П.3.

Описанная только что проверка цепи на неразрывность является, по сути, видом измерения сопротивления. В частности, зуммер звучит и при низком сопротивлении измеряемого элемента цепи. Но чтобы узнать значение сопротивления, прибор необходимо переключить в режим измерения сопротивления. Для этого переключатель режимов нужно установить на одну из меток в области, обозначенной символом «омега» Ω , который представляет единицу сопротивления Ом. При этом нужно обязательно убедиться, что на измеряемый резистор или другой элемент не подается питание. Также, в целях повышения точности

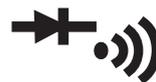


Рис. П.2. Символ режима определения неразрывности электроцепи



Рис. П.3. Символ режима измерения сопротивления

измерений, по крайней мере один вывод измеряемого элемента не должен быть ни к чему подключен. Как и многие другие электрические элементы, резистор имеет два вывода. Чтобы измерить его сопротивление, просто прикоснитесь концами щупов к его выводам. Так же, как и при определении неразрывности электроцепи, не имеет значения, какой щуп подключать к какому выводу резистора.

Область измерения сопротивления разбита на несколько диапазонов с разными предельными значениями измерения. Например, чтобы с точностью измерить сопротивление резистора малого номинала, переключатель режимов следует установить на указатель одного из низших пределов — например, на 200 Ω .

Если измеряемое сопротивление выше установленного максимального значения, на дисплее мультиметра будут отображаться символы [1.] без нулей. В таком случае просто установите переключатель на следующий более высокий предел и повторите попытку.

Попробуйте измерить сопротивление какого-либо резистора. В частности, попробуйте измерить сопротивление резистора номиналом 330 Ом (оранжевая, оранжевая и коричневая полоски). Какие показания у вас получаются на разных установленных пределах? Реальное сопротивление всех резисторов отличается от указанного номинального в пределах допустимого отклонения, или допуска. Такой допуск обычно составляет 5 процентов и указывается четвертой полоской. Каков процент отклонения измеренного вами значения сопротивления резистора от указанного номинального? Находится ли он в пределах допустимого?

Теперь измерьте сопротивление фоторезистора. При этом держите над фоторезистором руку или какой-либо непрозрачный предмет и замерьте сопротивление фоторезистора при разной высоте над ним затеняющего его объекта.

Измерение напряжения

Этот режим измерений обозначается на мультиметре символом, показанным на рис. П.4.



Рис. П.4. Символ режима измерения напряжения

Напряжение представляет собой разницу электрических потенциалов между двумя точками. Подобно режиму измерения сопротивления, режимы измерения напряжения также имеют несколько диапазонов с разными максимальными пределами.

Вы, наверное, обратили внимание, что режим измерения напряжения в действительности обозначается двумя символами: один — с прямыми линиями, другой — с одной волнистой линией. Символ с двумя прямыми линиями обозначает измерение напряжения постоянного тока, который является наиболее часто используемым типом тока в электронике (по крайней мере, в цифровой электронике). А волнистая линия обозначает переменный ток, примером которого может служить ток в электрической сети вашего дома. Таким образом, для измерения напряжения используется два режима: режим измерения постоянного напряжения и режим измерения переменного напряжения. Поэтому при измерении напряжения нужно в обязательном порядке правильно установить переключатель режимов. В подавляющем большинстве случаев это будет постоянное напряжение. Для большинства проектов этой книги самым лучшим диапазоном измерений будет диапазон с пределом 20 В, поскольку все напряжения в Arduino не превышают 5 В.

Для практики попробуйте измерить напряжение на плате Arduino. Для этого подключите плату Arduino к компьютеру и подсоедините черный

щуп к гнезду, обозначенному GND («земля»). Затем касайтесь красным щупом разных точек на плате (при этом соблюдая максимальную осторожность, чтобы случайно не закоротить щупом соседние дорожки, что чревато опасностью вывода платы из строя) или гнезд выводов. Какие показания отображаются для вывода, обозначенного 5 В (5V)? А для вывода 3.3 В (3,3V)?

Измерение тока

Этот режим измерений обозначается на мультиметре символом, показанным на рис. П.5.



Рис. П.5. Символ режима измерения тока

Ток представляет собой поток зарядов в замкнутой электрической цепи. Сила тока определяется как скорость перемещения зарядов по цепи и измеряется в амперах (А). Чтобы измерить скорость перемещения зарядов и, таким образом, измерить силу тока в цепи, необходимо сделать разрыв цепи в точке, в которой нужно измерить силу тока, и подсоединить щупы мультиметра к обоим концам разрыва. При этом переключатель режима надо установить на измерение тока, а также на диапазон с необходимым пределом. Если ожидается, что ток в точке измерения будет выше 200 мА, установите переключатель на значение 10 А и также вставьте красный щуп в гнездо, обозначенное 10 А. Если вы не уверены, какой может быть сила тока в измеряемой точке, это будет самый безопасный диапазон, с которого начинать измерение. Установка неправильного диапазона может повредить мультиметр.

Например, чтобы измерить ток, проходящий через простую цепь из светодиода и резистора, щупы можно вставить в разрыв цепи между светодиодом и резистором, как показано на рис. П.6. При этом путь тока должен проходить через мультиметр. Поскольку это последовательная цепь, ток можно измерять в разрыве в любой точке цепи: до светодиода, после светодиода или после резистора.

При измерении тока важно, чтобы измеряемый ток не превышал установленного предела измерений мультиметра. Сведения о максимально допустимой величине тока, который может измерять ваш мультиметр, можно найти в документации на него. Превышение этого предела может вызвать перегорание плавкого предохранителя мультиметра. Но не переживайте, если это случится, поскольку новый предохранитель стоит совсем недорого. Чтобы заменить перегоревший предохранитель, придется открыть заднюю крышку корпуса мультиметра, отвинтив крепежные шурупы. При подключении красного щупа в стандартный разъем mAΩ можно безопасно измерять ток величиной до 200 мА.

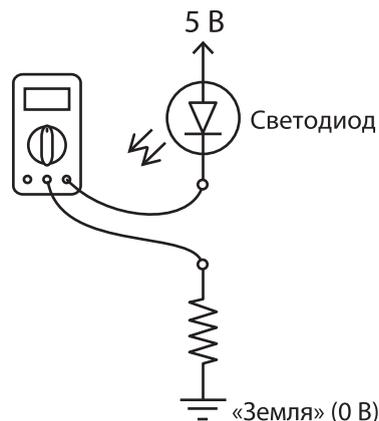


Рис. П.6. Подключение мультиметра в разрыв цепи для измерения тока

Работа с паяльником

Навыки пайки являются одними из самых основных навыков, которыми следует владеть при создании прототипов электронных проектов. Процесс пайки состоит из нанесения на спаиваемые компоненты расплавленного специального металла, называемого *припоем*, так, чтобы он обволакивал эти компоненты, и затем позволения ему остыть, надежно скрепив компоненты. Пример пайки показан на рис. П.7.

Обычный припой (рис. П.8) представляет собой проволоку из сплава олова и свинца и плавится при сравнительно низкой температуре. В частности, современные припои плавятся при температуре около 180 °С, что приблизительно равно температуре при выпечке печенья. Большинство припоев, используемых для пайки электронных деталей, имеют сердечник из *флюса*. Флюс — это

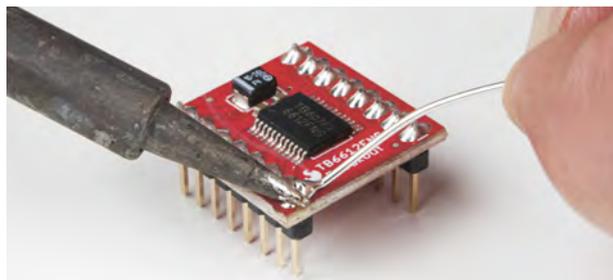


Рис. П.7. Пайка



Рис. П.8. Катушка припоя

специальное вещество, способствующее надежной адгезии (прилипанию) припоя к спаиваемым деталям. При плавлении припоя флюс помогает удалять грязь со спаиваемых поверхностей и улучшает их обволакивание припоем.

Припой плавится с помощью *паяльника*. Большинство паяльников имеют длину около 20–25 см и состоят из двух основных частей: деревянной или пластмассовой ручки и заключенного в металлический кожух нагревательного элемента с жалом (рис. П.9).

Существует много разных типов и стилей паяльников. Самые дешевые стоят около 10 долларов (60 рублей) и обычно имеют постоянный нерегулируемый нагрев. Но рекомендуется приобрести для работы паяльник с каким-либо типом

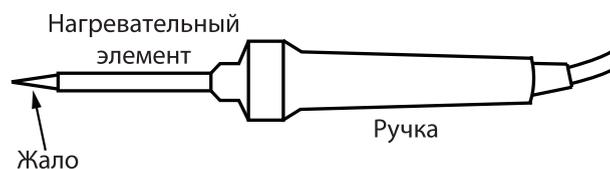


Рис. П.9. Типичный паяльник

регулировки нагрева жала. Оптимальная рабочая температура жала паяльника составляет около 345 °С. При слишком высокой температуре жало будет быстро окисляться и загрязняться, а при слишком низкой температуре припой не станет плавиться. Наличие регулировки температуры жала решает эти две проблемы и стоит нескольких лишних рублей.

Будьте осторожны при работе с паяльником — после включения жало нагревается очень быстро до температуры плавления металла. Это очень высокая температура! Всегда беритесь за паяльник со стороны ручки и никогда со стороны нагревательного элемента, даже когда вы думаете, что он выключен.

Также следует защитить рабочую поверхность стола, положив на нее кусок картона, коврик для резки или кусок фанеры или доски. Прежде чем приступить к работе с паяльником, всегда наденьте какую-либо защиту для глаз. Горячий припой и флюс имеют свойство иногда разбрызгиваться. Хотя эти брызги совсем небольшие, они достаточно велики, чтобы повредить ваши глаза. Так что лучше обезопасить их от этой неприятности.

Разогревание паяльника

Для работы с паяльником включите его и дайте время, чтобы жало нагрелось до рабочей температуры. В зависимости от типа паяльника это может занять от около 30 секунд до пары минут. Пока паяльник разогревается, убедитесь, что он

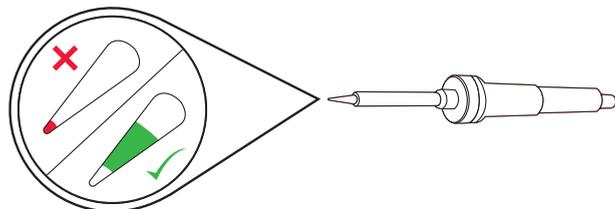


Рис. П.10. Температура стороны жала намного выше температуры кончика жала

Неправильно: Не используйте самый кончик жала.

Правильно: Используйте сторону жала, ближнюю к его кончику, — «сладкое пятнышко».

установлен на подставке таким образом, чтобы жало не касалось стола или другой поверхности, на которой вы работаете.

Разогретый надлежащим образом паяльник должен легко плавить припой. Проверьте это, коснувшись кусочка припоя стороной жала, ближней к его кончику. Это самая горячая область жала (рис. П.10), и именно ее нужно использовать для разогрева компонентов и плавки припоя. Если припой сразу же начал плавиться, жало достаточно горячее, и можно приступить к работе.

Советы по улучшению навыков пайки

Вопреки тому, что вы можете предполагать, при пайке мы не касаемся жалом паяльника непосредственно самого припоя, чтобы расплавить его. Вместо этого мы разогреваем жалом спаиваемые компоненты в течение около 2–3 секунд, а затем касаемся припоем нагретой точки, в результате чего он расплавится. Расплавленный припой всегда стремится к источнику тепла и оседает на самой горячей точке компонента. Если плавить припой, касаясь им непосредственно жала паяльника, он может собраться в каплю на жале и отказаться переходить на спаиваемые детали. Если такое случится, просто очистите жало паяльника (см. об этом далее) и повторите попытку. Удерживайте паяльник в своей «рабочей» руке со стороны ручки таким образом, как будто бы это карандаш. В другой руке удерживайте отрезок припоя. Будьте осторожны, чтобы не держаться за припой слишком близко к его расплавляемой области, чтобы не обжечь пальцы.

Прикоснитесь стороной жала паяльника к спаиваемым деталям. При этом жало должно касаться обеих спаиваемых деталей, чтобы они обе разогревались равномерно (см. рис. П.7). Отсчитайте три секунды: «раз одна тысяча, два одна тысяча, три одна тысяча»².

² Английский метод отсчета секунд: one one-thousand, two one-thousand, three one-thousand. Вроде бы должен работать и по-русски. Или пользуйтесь секундомером.

Затем, продолжая касаться жалом компонентов, прикоснитесь кончиком отрезка припоя к спаиваемому месту. Помните, что расплавленный припой стремится к источнику тепла.

Когда соединение содержит достаточно припоя, удалите остальной припой, но удерживайте жало на месте еще одну секунду. Это позволит припою равномерно распределиться и осесть в месте пайки. Удалите паяльник от места пайки и установите его обратно на подставку.

Поверхность застывшего припоя в хорошей пайке должна быть гладкой и немного блестящей.

Место пайки выводов деталей в отверстиях печатной платы часто выглядит как конус вулкана или шоколадная конфетка Hershey's kiss³. Чтобы научиться хорошо паять, необходима практика, поэтому если ваша пайка не выглядит гладкой и блестящей, попробуйте еще нагреть место соединения, чтобы снова дать припою растечься и осесть на деталях, или же добавьте еще чуток припоя.

На рис. П.11 показаны примеры правильного паяного соединения и некоторые распространенные огрехи пайки и возможные способы их устранения.

³ См. https://en.wikipedia.org/wiki/Hershey's_Kisses.

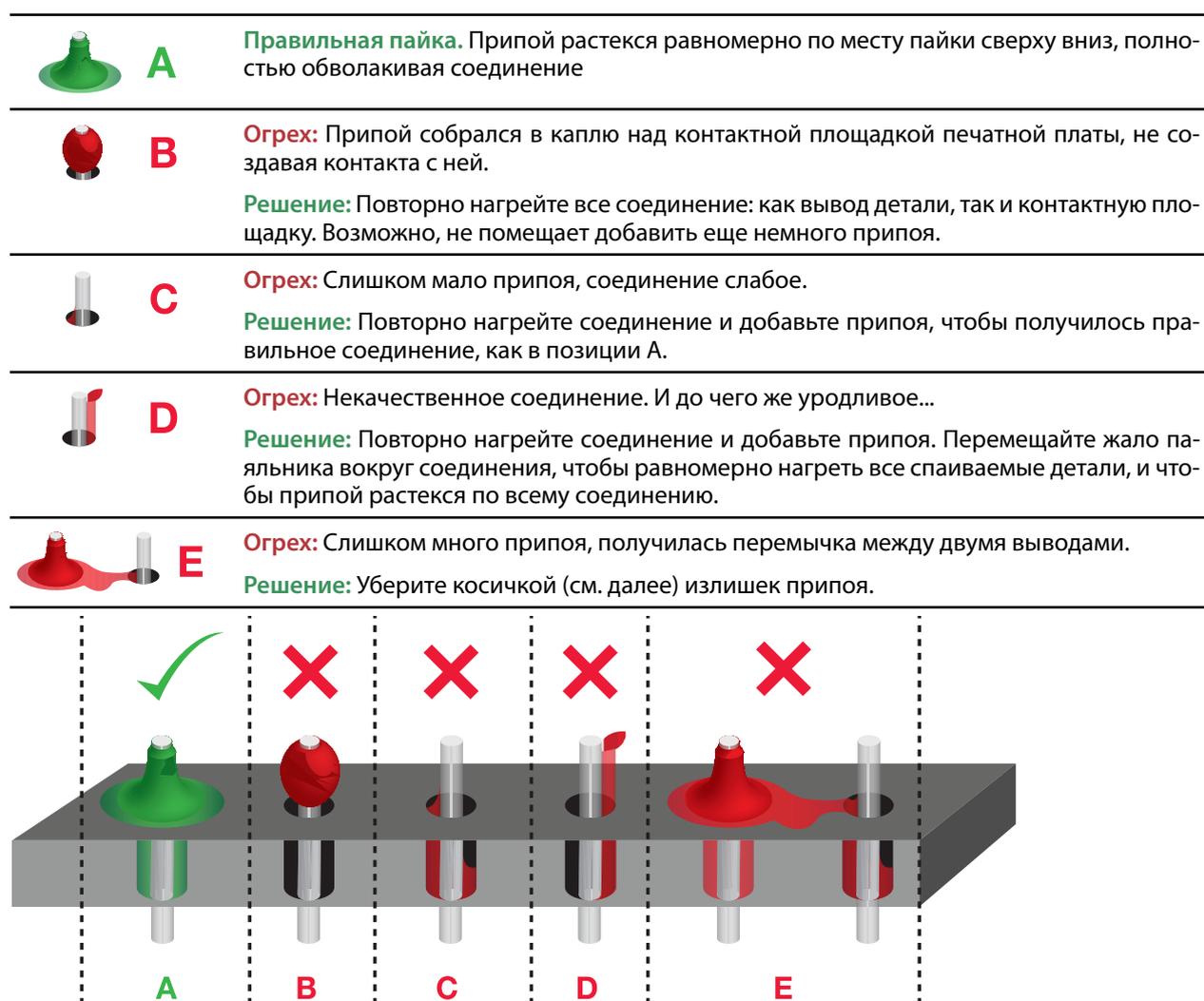


Рис. П.11. Правильное паяное соединение и наиболее распространенные огрехи пайки и способы их устранения

Очистка паяльника

Один из секретов получения хороших паяных соединений заключается в чистоте жала паяльника. Жало рекомендуется очищать перед каждым использованием паяльника. Это можно делать, нагрев паяльник и потерев жало о латунный скребок (наподобие скребка из нержавеющей стали для мытья посуды) или о мокрую губку, чтобы стереть излишек припоя или накопившиеся окисления.

Если жало настолько загрязнилось, что указанные методы очистки не работают, его можно очистить с помощью специальной смеси слабой кислоты и припоя Tip Tinner and Cleaner (TOL-13246). Для этого разогрейте паяльник, вставьте жало паяльника в состав для чистки и лужения и удерживайте в течение около 10–15 секунд, чтобы дать составу разъесть окисления и загрязнения. Затем извлеките паяльник и вытрите жало о губку. Повторите процедуру, если необходимо. После этого жало паяльника должно быть покрыто тонким слоем припоя.

Примечание

Некоторые типы припоя содержат свинец, который является токсичным металлом. Но независимо от типа используемого вами припоя, настоятельно рекомендуется мыть руки после выполнения паяльных работ.

Советы по работе с паяльником

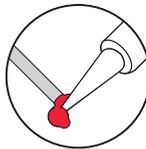
На рис. П.12 показано еще несколько советов по работе с паяльником.



Правильно: Жало паяльника касается одновременно как вывода компонента, так и кольца контактной площадки.



Правильно: Удерживая жало паяльника в контакте с выводом и кольцом контактной площадки, подавайте припой в место пайки..



Неправильно: Не расплавляйте припой на жале паяльника, нанося его затем жалом на соединение.



Правильно: Очищайте жало паяльника, когда на нем накопятся окисления.

Рис. П.12. Дополнительные советы и рекомендации по работе с паяльником

Умение работать с паяльником должно присутствовать в арсенале навыков любого любителя электроники. Если вы решите сделать свои прототипы постоянно действующими и прочными, монтаж пайкой обеспечит намного более надежные и долговременные соединения.

Дополнительные инструменты для паяльных работ

Далее кратко рассматриваются несколько дополнительных инструментов, использование которых поможет вам получать качественные паяные соединения на постоянной основе. Это инструменты для удерживания спаиваемых деталей, очистки соединений и удаления излишков припоя.

«Третья рука»

«Третья рука» — это, по сути, зажим для удерживания спаиваемых деталей, и он станет одним из

наиболее полезных помощников при выполнении паяльных работ. Существует много разновидностей «третьей руки», но большинство из них представляют собой просто пару зажимов типа «крокодил» на стойке с тяжелым основанием, с помощью которых можно закрепить спаиваемые детали, освобождая руки для держания паяльника и припоя. Многие варианты «третьей руки» оснащены увеличительным стеклом и небольшим держателем для паяльника. Пример такой «третьей руки» показан на рис. П.13.



Рис. П.13. «Третья рука» с увеличительным стеклом и держателем для паяльника

Флюс-аппликатор

Одним из секретов получения качественных паяных соединений является чистота спаиваемых деталей, что обеспечивается нанесением на место пайки флюса. Флюс — это чистящая жидкость небольшого уровня кислотности, которая часто изготавливается на основе сосновой канифоли. Флюс-аппликатор (рис. П.14) работает наподобие чернильного маркера — просто нажимаем кончиком маркера на спаиваемое соединение, пока на него не перейдет небольшая капля флюса. После этого приложите жало паяльника непосредственно в точку пайки и внесите припой. С использованием флюса припой расплавится намного быстрее и будет иметь лучшую адгезию со спаиваемыми деталями.



Рис. П.14. Аппликатор с водорастворимым флюсом

Флюс творит чудеса при пайке, но он слегка едкий, поэтому следите за тем, чтобы свести к минимуму контакт флюса с кожей, и мойте руки сразу же после работы с ним.

Косичка для удаления припоя

Иногда случается, что в место пайки наносится слишком много припоя, или припой попадает в место, где его не должно быть. С удалением излишков припоя вам помогут следующие два приспособления. Первым из них является косичка из тонких медных проводов, показанная на рис. П.15.

Косичкой пользуются следующим образом. Положите конец косички поверх места с лишним припоем и приложите к косичке жало паяльника. Когда косичка нагреется до достаточной температуры, она расплавит находящийся под ней припой и впитает его в себя. Убрав косичку, мы уберем вместе с ней и лишний припой.

Но нужно следить за тем, чтобы удерживать жало паяльника на косичке при ее снятии. Если убрать паяльник от косички слишком рано, она припаяется к соединению. В таком случае просто снова нагрейте косичку и удалите ее.



Рис. П.15. Косичка для удаления припоя

Вакуумный отсос

Вторым инструментом для удаления лишнего припоя является вакуумный отсос. Этот инструмент похож на большой шприц. При нажатии на поршень шприца он сжимает находящуюся под ним пружину, и поршень фиксируется в нижнем положении защелкой. При нажатии на кнопку освобождения защелки под воздействием пружины поршень шприца резко поднимается вверх, создавая вакуум в носике.

Разогрейте место с лишним припоем, чтобы он полностью расплавился и стал текучим. Продолжая разогревать припой, быстро приложите

носик отсоса к лужице припоя и нажмите кнопку освобождения поршня. Создаваемый поршнем вакуум должен всосать расплавленный припой в корпус отсоса.

Если убрать лишний припой с первого раза не получилось, повторите попытку. Использование этого инструмента требует определенной сноровки и быстроты действий. Иногда полезно добавить дополнительный припой к тому, который нужно убрать.

Полосатые резисторы

Резисторы бывают самых разных номиналов. Но как определить сопротивление резистора, если на нем нет никаких цифр или текста?

Значения сопротивлений резисторов обозначаются с помощью системы цветных полос. Принцип работы этой системы показан на рис. П.16.

Обычно резисторы имеют четыре или пять цветных полос. Последняя полоса обозначает допустимое отклонение в процентах действительного значения сопротивления резистора от номинального значения. Большинство резисторов имеют допуск величиной 5 процентов, который обозначается золотистой полосой. Это означает, что действительное значение резистора может отклоняться на 5 процентов от указанного номинального значения. Например, действительное сопротивление резистора с номинальным сопротивлением 10 кОм и допуском 5 процентов может быть где-то в диапазоне от 9,5 до 10,5 кОм. При этом считается, что такой резистор имеет сопротивление 10 кОм.

Маркировочные полосы читаются слева направо, с полосой допуска (обычно золотистой или серебристой), расположенной справа. В случае резисторов с четырьмя маркировочными полосами

первые две полосы обозначают базовое значение, а третья — степень множителя 10. Для резисторов с пятью маркировочными полосами базовое значение обозначается первыми тремя полосами, а четвертая полоса обозначает множитель.

Определим, например, сопротивление резистора, обозначенное коричневой, черной и оранжевой полосами. Согласно таблице, представленной на рис. П.16, коричневая полоса обозначает 1, а черная 0. Таким образом, базовое значение равно 10. Третья полоса, оранжевая, обозначает степень 3 множителя 10, то есть 10^3 или 1000. Умножая базовое значение 10 на множитель 1000, получаем 10 000 Ом или 10 кОм. Наконец, четвертая полоса, золотистая, обозначает допуск 5 процентов.

Чтобы справка по цветовому коду резисторов всегда была у вас под рукой, можно вырвать из книги страницу с таблицей и разместить ее в легкодоступном месте. Не волнуйтесь — мы не скажем библиотекаряше.

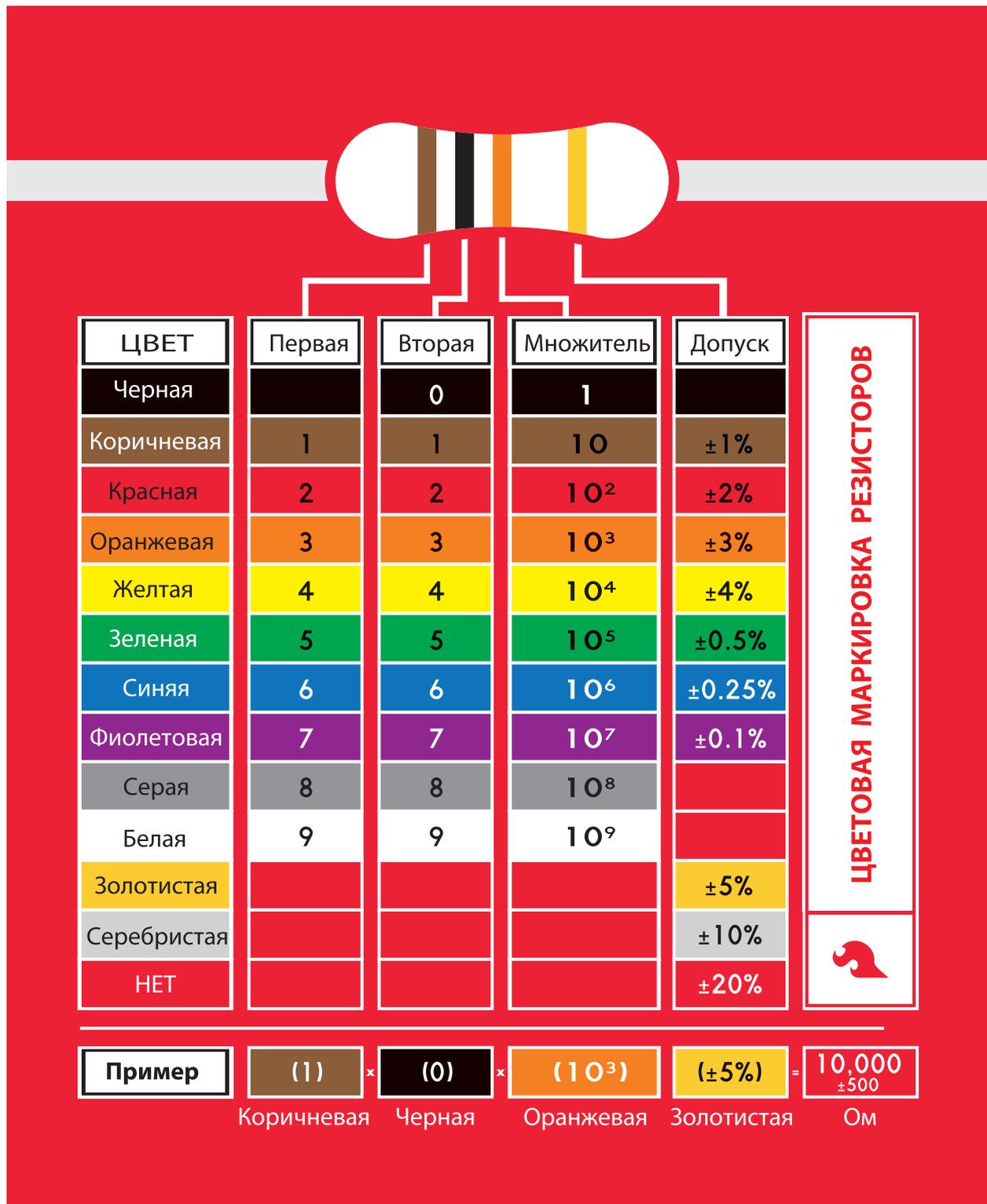


Рис. П.16. Шпаргалка цветовой маркировки резисторов

ARDUINO

ДЛЯ ИЗОБРЕТАТЕЛЕЙ

Набор
электронных
компонентов
+ КНИГА



СОСТАВ НАБОРА

- + Arduino UNO
- + Датчики
- + Двигатели, колеса
- + Светодиоды, ЖК-дисплей
- + Диоды, транзисторы
- + Макетные платы и провода
- + Книга Б. Хуанга, Д. Ранберга «Arduino для изобретателей. Обучение электронике на 10 занимательных проектах»



www.bhv.ru/books/200380