

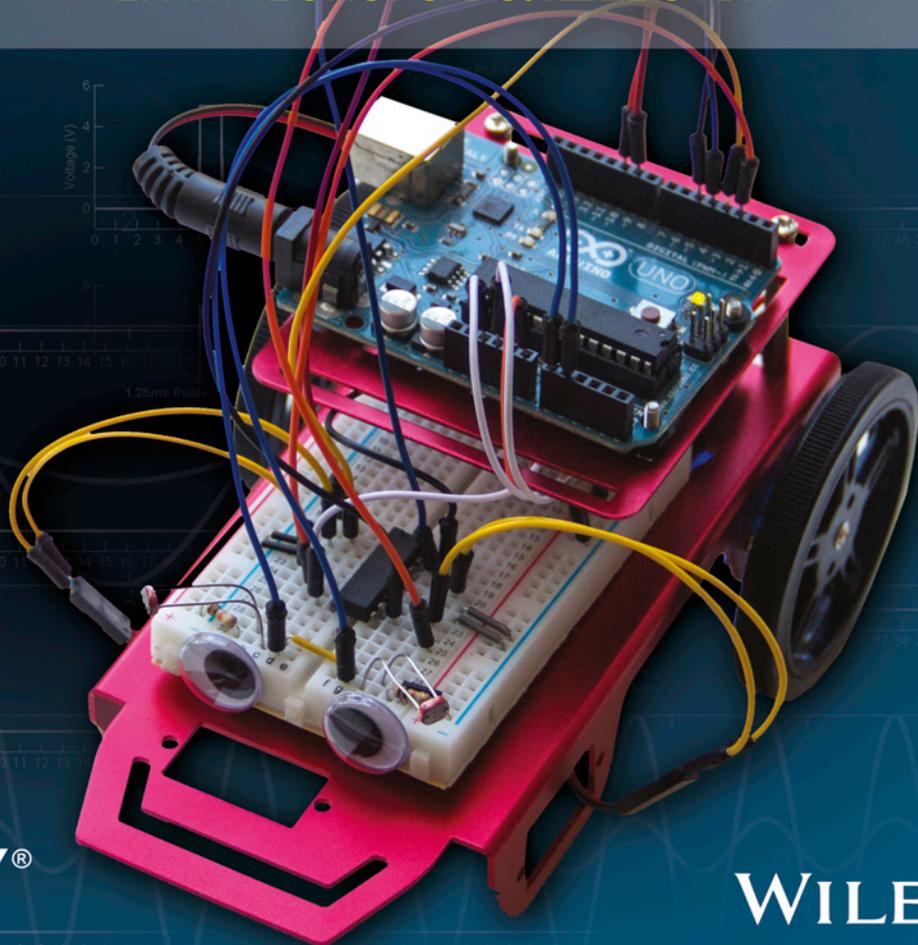
ВТОРОЕ ИЗДАНИЕ

Новые
проекты
и темы

Джереми Блум

ИЗУЧАЕМ
ARDUINO

ИНСТРУМЕНТЫ И МЕТОДЫ
ТЕХНИЧЕСКОГО ВОЛШЕБСТВА



bhv®

WILEY

EXPLORING ARDUINO®

Tools and Techniques for Engineering Wizardry

Second Edition

Jeremy Blum

WILEY

Джереми Блум

ИЗУЧАЕМ
ARDUINO

ИНСТРУМЕНТЫ И МЕТОДЫ
ТЕХНИЧЕСКОГО ВОЛШЕБСТВА

ВТОРОЕ ИЗДАНИЕ

Санкт-Петербург
«БХВ-Петербург»
2020

УДК 004
ББК 32.973.26
Б71

Блум Дж.

Б71 Изучаем Arduino: инструменты и методы технического волшебства. 2-е изд.: пер. с англ. — СПб.: БХВ-Петербург, 2020. — 529 с.: ил.

ISBN 978-5-9775-6736-7

Книга посвящена проектированию электронных устройств на основе микроконтроллерной платформы Arduino. Приведены основные сведения об аппаратном и программном обеспечении Arduino. Изложены принципы программирования в интегрированной среде Arduino IDE. Показано, как анализировать электрические схемы, читать технические описания, выбирать подходящие детали для собственных проектов. Приведены примеры использования и описание различных датчиков, электродвигателей, сервоприводов, индикаторов, проводных и беспроводных интерфейсов передачи данных. В каждой главе перечислены используемые комплектующие, даны монтажные схемы, подробно описаны листинги программ. Материал ориентирован на применение несложных и недорогих комплектующих для экспериментов в домашних условиях. Во втором издании обновлены проекты из предыдущего издания, добавлены проекты с шаговыми двигателями и беспроводной связью Bluetooth и Wi-Fi, а также более глубоко рассмотрены темы электротехники и разработки устройств.

Для радиолюбителей

УДК 004
ББК 32.973.26

Группа подготовки издания:

Руководитель проекта	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Сависте</i>
Компьютерная верстка	<i>Людмилы Гауль</i>
Оформление обложки	<i>Карины Соловьевой</i>

Authorized Russian translation of the English edition of **Exploring Arduino®: Tools and Techniques for Engineering Wizardry, Second Edition**, ISBN 978-1-119-40537-5 © 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana. All Rights Reserved.

This translation is published under license by BHV, © 2020.

Авторизованный перевод с английского на русский язык произведения **Exploring Arduino®: Tools and Techniques for Engineering Wizardry, второе издание**, ISBN 978-1-119-40537-5 © 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana. Все права защищены.

Этот перевод публикуется по лицензии издательством «БХВ», © 2020.

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

ISBN 978-1-119-40537-5 (англ.)
ISBN 978-5-9775-6736-7 (рус.)

© 2020 by John Wiley & Sons, Inc.
© Перевод на русский язык, оформление. ООО «БХВ-Петербург»,
ООО «БХВ», 2020

*Посвящается Лиа за ее помощь мне
и совет рассматривать каждую трудность
как благоприятную возможность.*

Об авторе

В настоящее время Джереми Блум занимает должность технического директора в компании Shaper (<https://www.shapertools.com>), где он изобретает новый способ использования ручных электроинструментов, применяя для этого машинное зрение. Ранее Джереми работал ведущим конструктором компании Google[x] и разрабатывал различные конфиденциальные продукты, включая Google Glass.

Джереми присвоена степень магистра по электронике и вычислительной технике в Корнеллском университете, в котором он ранее также получил степень бакалавра по тому же профилю. В Корнеллском университете он основал и возглавлял организацию по экологически щадящему проектированию Cornell University Sustainable Design; создал первое в своем роде офисное пространство для совместной работы студентов; а также проводил исследования в области робототехники и машинного самообучения.

Джереми занимался разработкой медицинских протезов, систем освещения с использованием волоконной оптики и светодиодов, бытовой автоматики, 3D-принтеров и сканеров, самособирающихся и самообучающихся роботов, переносных вычислительных платформ, а также устройств дополненной реальности. Его работы представлялись на международных конференциях и публиковались в рецензируемых научных журналах и в таких популярных изданиях средств массовой информации, как телеканал Discovery Channel, газета Wall Street Journal и журнал Popular Science. Журнал Forbes включил его в свой ежегодный список “30 младших, чем 30” в знак признания его деятельности, которая способствует техническому прогрессу в Америке. Он является соавтором нескольких патентов в области конструирования переносных вычислительных платформ и устройств дополненной реальности.

В свободное от разработки и реализации новых проектов время Джереми занимается обучением. Его статьи и видеоуроки помогают миллионам людей изучать проектирование в области электротехники и встроенного программного обеспечения. Его книга “*Exploring Arduino*” была переведена на многие языки и используется в качестве технического учебника во всем мире, включая его альма-матер, Корнеллский университет. Джереми увлекается применением технологии для улучшения жизни людей, и разработкой инструментов для этого. Узнать больше о Джереми можно на его веб-сайте: <https://www.jeremyblum.com>.

О техническом редакторе

Доктор Дерек Моллой (Derek Molloy) занимает должность доцента кафедры электронной техники инженерно-вычислительного факультета Городского университета Дублина в Ирландии. Он читает лекции для студентов бакалавриата и аспирантуры по объектно-ориентированному программированию для встроенных систем, цифровой и аналоговой электронике и сетевым встроенным системам. Он занимается исследованиями в основном в области компьютерного и машинного зрения, встроенных систем, трехмерной графики и визуализации, а также в области дистанционного обучения. Дерек ведет популярную серию видеоуроков на канале YouTube, благодаря которым миллионы людей смогли познакомиться с предметом встроенной операционной системы Linux и другими областями цифровой электроники. В 2013 г. он создал персональный веб-сайт/блог, в котором размещены его видеоуроки с вспомогательными материалами, исходный код программ и отзывы пользователей. Этот сайт ежедневно посещают тысячи людей. В мини-серии издательства Wiley он опубликовал свои книги *“Exploring BeagleBone”* (в 2015 г.) и *“Exploring Raspberry Pi”* (в 2016 г.). Ранее в этом году вышло второе издание его книги *“Exploring BeagleBone”*. Узнать больше о Дереке, его работе и других его публикациях можно на его веб-сайте: <http://derekmolloy.ie>.

Благодарности

На протяжении нескольких лет после выхода первого издания этой книги я получил огромное количество писем от читателей, в которых они сообщали обо всем, чему смогли научиться. Было высказано и немало конструктивной критики, в основном по поводу незначительных аспектов, корректировка которых может повысить качество книги. Я серьезно относился к этим замечаниям и старательно отслеживал их в течение последних лет. Я намереваюсь сделать второе издание этой книги еще более полезным, чем первое, в то же самое время сохраняя легкость изложения, за которую многие читатели выражали свою признательность. Поэтому БОЛЬШУЩЕЕ СПАСИБО всем тем, кто предоставил мне свои отклики о первом издании книги *«Изучаем Arduino»*.

Следующим делом я снова хочу выразить свою благодарность издательству Wiley. В течение нашей совместной работы сотрудники издательства

проявили себя великолепными партнерами, и я рад, что получил согласие выпустить второе издание книги. В частности, я хочу поблагодарить Джима Минатела (Jim Minatel), Адаоби Оби Тултона (Adaobi Obi Tulton), Доктора Дерека Моллоя (Dr. Derek Molloy), Мерилуизу Виак (Marylouise Wiack) и Атийаппан Лалит Кумара (Athiyappan Lalith Kumar).

Также говорю спасибо всем замечательным людям компании Adafruit, работавшим совместно со мной над обеспечением доступности наборов компонентов для проектов из этой книги. Компания Adafruit оказывает большую поддержку сообществам открытого аппаратного и программного обеспечения, и я не стал бы тем инженером, каким являюсь в настоящее время, без их отличных компонентов и руководств.

Когда я работал над первым изданием книги *«Изучаем Arduino»*, я одновременно обучался по программе магистратуры. Я получил степень магистра несколько лет тому назад, и теперь мне нужно концентрироваться на работе в компании Shaper. Я также должен высказать благодарность всем моим коллегам как в компании Shaper, так и в компании Google (мой предыдущий работодатель) за их постоянную поддержку и помощь в создании замечательных устройств.

Отдельную благодарность я хочу выразить моим преподавателям в Корнеллском университете, особенно профессору Франгоису Гимбretiере (Francois Guimbretiere), который вел занятия, где я познакомился с Arduino. После выхода первого издания этой книги он использовал ее в качестве учебника для своего курса. Мне доставляет большую радость осознавать, что я смог отблагодарить Корнеллский университет таким образом.

Наконец, я хочу поблагодарить моих родителей, моего брата, мою жену и моих друзей за их терпение и за постоянную поддержку и ободрение. Считаю, мне очень повезло, что в моей жизни есть такие замечательные люди.

Оглавление

Об авторе	7
О техническом редакторе	8
Благодарности	8
Об авторских правах.....	19
Введение	21
Почему Arduino?	22
Для кого эта книга.....	22
Чему вас научит эта книга	23
Стиль типографского оформления.....	24
Приобретение компонентов	24
Что вам потребуется.....	25
Исходный код и прочие электронные ресурсы	26
Список опечаток.....	27
Дополнительный материал и поддержка.....	27
Что такое Arduino?.....	28
О движении Open Source	29
Несколько советов читателю	29
Часть I. Технические основы платформы Arduino	31
Глава 1. Начало работы и основные сведения	
о платформе Arduino.....	33
Что вы узнаете из этой главы.....	33
Исследуем среду Arduino	34
Функциональные возможности Arduino	35
Платы Arduino	41
Создаем и исполняем первую программу	45
Загрузка и установка среды Arduino IDE.....	47
Запускаем среду IDE и подключаемся к плате Arduino	47
Структура нашей первой программы	49
Резюме	52
Глава 2. Цифровые входы и выходы	
и широтно-импульсная модуляция	53
Что вы узнаете из этой главы.....	53
Ввод цифровых сигналов	54
Подключение светодиода и работа с беспаячными	
макетными платами	54

Программирование цифровых выводов	60
Цикл for	61
Генерирование ШИМ-сигнала с помощью функции analogWrite()	63
Считывание входных цифровых сигналов	67
Использование понижающего резистора для считывания входных цифровых сигналов	68
Управляемый ночник на трехцветном светодиоде	76
Резюме	82

Глава 3. Считывание сигналов аналоговых датчиков 83

Что вы узнаете из этой главы.....	83
Аналоговые и цифровые сигналы.....	84
Сравнение аналоговых и цифровых сигналов	85
Преобразование аналоговых сигналов в цифровые	86
Опрос аналоговых датчиков с помощью Arduino.....	87
Считывание выходного сигнала потенциометра	88
Применение аналоговых датчиков.....	92
Переменные резисторы в качестве аналоговых датчиков	97
Резистивные делители напряжения.....	97
Управление выходным аналоговым сигналом посредством входного аналогового сигнала.....	100
Резюме	102

Часть II. Взаимодействие с окружающей средой..... 103

Глава 4. Использование транзисторов и управление электродвигателями постоянного тока..... 105

Что вы узнаете из этой главы.....	106
Электродвигатели постоянного тока.....	106
Работа с сильноточными индуктивными нагрузками.....	107
Управление скоростью вращения электродвигателя посредством ШИМ.....	113
Управление направлением вращения электродвигателя с помощью H-моста.....	115
Создаем шасси самоходного робота	125
Выбор компонентов для шасси робота	125
Собираем схему управления роботом.....	128
Разработка программы управления самоходным шасси	132
Сборка устройства.....	136
Резюме	137

Глава 5. Управление сервоприводами и шаговыми двигателями	139
Что вы узнаете из этой главы.....	140
Управление сервоприводами.....	140
Разница между сервоприводами кругового вращения и с поворотом в пределах сектора	140
Управление сервоприводом.....	141
Программа управления сервоприводом.....	144
Создание сканирующего дальномера	146
Шаговые двигатели и управление ими.....	150
Принцип работы биполярных шаговых двигателей.....	152
Приводим шаговый двигатель в действие	154
Создаем одноминутный хронограф.....	159
Сборка схемы хронографа.....	160
Программа хронографа	162
Резюме	167
Глава 6. Работаем со звуком	169
Что вы узнаете из этой главы.....	169
Принцип работы динамика	170
Свойства звука	170
Как динамик воспроизводит звук	172
Генерация звуков с помощью функции tone()	173
Подключение файла сопоставления нот частотам.....	173
Сборка схемы с динамиком.....	175
Создание звуковых последовательностей.....	177
Ограничения функции tone()	181
Миниатюрное пианино	181
Резюме	184
Глава 7. Последовательный интерфейс USB.....	185
Что вы узнаете из этой главы.....	185
Возможности последовательного обмена данными платформы Arduino	186
Использование встроенной или внешней микросхемы преобразователя USB/RS-232 компании FTDI или Silicon Laboratories	187
Платы Arduino с дополнительным микроконтроллером ATmega в качестве преобразователя USB/RS-232.....	190
Платы Arduino с основным микроконтроллером, поддерживающим интерфейс USB.....	191
Платы Arduino с возможностями USB-хоста	192

Прием данных от Arduino на компьютере.....	192
Использование команды <i>print()</i>	192
Специальные символы.....	194
Изменение представления типов данных.....	196
Передача данных с компьютера на плату Arduino.....	196
Настройка монитора порта для отправки команд.....	196
Получение данных, отправляемых с компьютера или другого устройства через последовательный интерфейс.....	197
Взаимодействие с компьютерной программой.....	206
Установка Processing.....	207
Управление приложением Processing посредством платы Arduino.....	208
Передача скетчем Processing данных на плату Arduino.....	211
Резюме.....	214
Глава 8. Эмуляция USB-устройств.....	215
Что вы узнаете из этой главы.....	215
Эмуляция клавиатуры.....	217
Ввод данных в компьютер.....	217
Управление компьютером с платы Arduino.....	221
Эмуляция мыши.....	223
Резюме.....	227
Глава 9. Сдвиговые регистры.....	229
Что вы узнаете из этой главы.....	229
Что такое сдвиговый регистр.....	230
Параллельная и последовательная передача данных.....	231
Использование микросхемы сдвигового регистра 74HC595.....	232
Загрузка в сдвиговый регистр данных с платы Arduino.....	235
Преобразования между двоичным и десятичным форматом.....	238
Световая анимация с помощью сдвигового регистра.....	239
Эффект «бегущего» светодиода.....	239
Гистограмма для реагирования на изменение входных условий.....	241
Резюме.....	244
Часть III. Интерфейсы для обмена данными.....	245
Глава 10. Шина I²C.....	247
Что вы узнаете из этой главы.....	247
История создания протокола I ² C.....	248
Схема подключения устройств I ² C.....	249

Взаимодействие и идентификация устройств.....	250
Аппаратные требования и повышающие резисторы	252
Как правильно выбрать номинал повышающего резистора	253
Взаимодействие с термодатчиком по шине I ² C	255
Сборка схемы устройства	255
Использование информации из справочного листка датчика ТС74.....	257
Программа для опроса датчика ТС74	259
Совместное использование сдвиговых регистров, последовательного интерфейса и шины I ² C.....	262
Собираем систему мониторинга температуры.....	262
Модифицируем скетч Arduino	263
Создаем скетч Processing.....	265
Резюме	269

Глава 11. Шина SPI и библиотеки

сторонних разработчиков..... 271

Что вы узнаете из этой главы.....	271
Общие сведения о шине SPI.....	272
Подключение устройств SPI.....	273
Конфигурация интерфейса SPI	274
Протокол передачи данных SPI	275
Сравнительные характеристики протоколов SPI, I ² C и UART	276
Взаимодействие с SPI-акселерометром	277
Что такое акселерометр?.....	278
Получаем информацию из справочного листка.....	280
Сборка схемы устройства	283
Разработка программы	285
Аудиовизуальный музыкальный инструмент на основе трехкоординатного акселерометра.....	292
Сборка схемы устройства	292
Модифицируем программу.....	294
Резюме	297

Глава 12. Взаимодействие с жидкокристаллическими

дисплеями..... 299

Что вы узнаете из этой главы.....	300
Подключение ЖКД к плате Arduino.....	300
Библиотека <i>LiquidCrystal</i> для работы с ЖКД.....	303
Вывод текста на дисплей	304
Создание специальных символов и анимация	306

Создаем термостат	311
Сборка схемы устройства	311
Отображение данных на ЖКД	314
Настройка порогового значения температуры с помощью кнопок.....	317
Код программы для звукового оповещения и управления вентилятором.....	319
Собираем все фрагменты кода в завершённую программу	320
Совершенству нет предела	325
Резюме	326

Часть IV. Прерывания, комбинирование функций и карты SD

Глава 13. Прерывания и другие специальные функции

Что вы узнаете из этой главы.....	329
Аппаратные прерывания	330
Опрос состояния и прерывания: преимущества и недостатки каждого подхода	331
Возможности аппаратных прерываний Arduino	333
Схема запуска прерывания кнопкой, оснащенной аппаратной защитой от дребезга.....	334
Прерывания по таймеру.....	343
Прерывания по таймеру.....	343
Установка библиотеки <i>TimerOne</i>	344
Одновременное исполнение двух задач	344
Музыкальный инструмент на прерываниях	345
Схема музыкального инструмента.....	346
Программа для музыкального инструмента	347
Резюме	349

Глава 14. Работа с картами памяти SD.....

Что вы узнаете из этой главы.....	351
Подготовка к регистрации данных.....	352
Файлы данных CSV-формата	353
Подготовка SD-карты для записи данных.....	353
Сопряжение платы Arduino с SD-картой.....	360
Шилды для SD-карт	360
Интерфейс SPI SD-карт	363
Запись на карточку SD	363

Считывание данных с SD-карты.....	368
Часы реального времени	372
Общие сведения о часах реального времени	372
Работа с часами реального времени	374
Регистратор проходов через дверь	384
Схема регистратора	384
Программа регистратора	386
Анализ данных.....	391
Резюме	392

Часть V. Беспроводная связь 395

Глава 15. Радиосвязь 397

Что вы узнаете из этой главы.....	397
Электромагнитный спектр	398
Радиоспектр	400
Принципы обмена данными по радиоканалу.....	402
Передача нажатий кнопки по радиоканалу	404
Подключение приемника к плате Arduino	404
Программа для работы с радиоприемником.....	405
Беспроводной дверной звонок	409
Схема приемной стороны звонка	410
Программа для приемной стороны звонка	411
Первые шаги на пути к «умному» дому — управление светильником	413
Электричество в доме и офисе	415
Принцип работы реле.....	416
Программа для управления реле.....	418
Подключение реле управления светильником к Arduino.....	420
Резюме	422

Глава 16. Беспроводная связь Bluetooth 423

Что вы узнаете из этой главы.....	423
Что такое Bluetooth?	424
Стандарты и версии Bluetooth	424
Профили Bluetooth и служба GATT BTLE.....	425
Установка связи между платой Arduino и смартфоном	427
Считывание датчика через канал BTLE	427
Передача данных со смартфона по каналу BTLE	442
Управление светильником посредством Bluetooth.....	453
Процедура сопряжения смартфона с устройствами BTLE	453
Программа для работы с датчиком присутствия	454

Сопряжение со смартфоном.....	460
Подключаем вместо светодиода светильник	462
Резюме	463
Глава 17. Wi-Fi и облачные хранилища	465
Что вы узнаете из этой главы.....	465
Работа Arduino в Сети	466
Сетевая терминология.....	466
Клиенты и серверы	470
Arduino с возможностями Wi-Fi	470
Управление платой Arduino через Интернет	471
Подготовка оборудования для управления вводом-выводом	471
Подготовка среды Arduino IDE для работы с платой Feather	473
Проверяем соответствие библиотеки Wi-Fi прошивке модуля Wi-Fi.....	475
Программа сервера для платы Arduino	476
Собираем весь код вместе для создания веб-сервера.....	488
Управление платой Arduino из локальной сети и извне ее.....	493
Взаимодействие с интерфейсом Web-API.....	498
Работа с интерфейсом Web-API для получения метеоданных	499
Вывод полученных данных на дисплей	512
Резюме	523
Приложение. Расшифровка справочных листков и изучение принципиальных схем	525
Работа со справочными листками	525
Составные части справочного листка	525
Цоколевка компонента	529
Изучение принципиальной схемы платы Arduino	532
Предметный указатель.....	535

Об авторских правах

Все изображения, значки и надписи на рис. 3.7 и рис. 10.3 принадлежат компании Analog Devices, Inc. (ADI) и защищены авторским правом © с 2019 г. Все права сохраняются. Эти изображения, значки и надписи воспроизведены с разрешения компании ADI. Воспроизведение, распространение или использование без письменного согласия компании ADI запрещено.

Содержащийся в данной книге материал, охраняемый авторским правом, принадлежащий компании Microchip Technology Incorporated, воспроизведен с разрешения. Все права сохраняются. Последующее воспроизведение без предварительного письменного разрешения компании Microchip Technology Inc. запрещено.

Названия Atmel, AVR, ICSP и In-Circuit Serial Programming являются торговыми марками или зарегистрированными торговыми марками компании Microchip Technology Inc.

Названия Arm и Cortex являются торговыми марками или зарегистрированными торговыми марками компании Arm Limited (или ее дочерних компаний) в Соединенных Штатах и/или в любой другой стране. Сопутствующая технология может быть защищенной частично или полностью патентами, авторскими правами, законами о промышленных образцах и законами о коммерческой тайне.

Введение

Первое издание этой книги, вышедшее в 2013 году, я открыл следующими словами: *«Мы живем в прекрасное время. Я люблю говорить, что мы уже живем в будущем».*

Я думаю, что с таким введением я загнал себя в угол, поскольку если 2013 год был «будущим», тогда я не совсем уверен, как можно называть настоящее! Далекое будущее? Очень далекое будущее? Тогда я хотел подчеркнуть этими словами, что в последнее время поступь прогресса была настолько стремительной, что возможности, предоставляемые даже поверхностными знаниями в области встроенных электронных устройств и программного обеспечения, расширяются буквально день ото дня.

Со времени выхода первого издания этой книги доступность электронных устройств и программного обеспечения продолжала повышаться с каждым прошедшим днем. В 2013 году я сомневался, стоит ли включать в книгу главу о подключении рассматриваемых в ней проектов к Интернету, поскольку в то время процесс для этого был довольно непростой. В 2013 году понятие «Интернет вещей» (IoT¹) было только начинающей входить в обиход модной фразой в среде фанатов в этой области. Сейчас же это одно из основных выражений глобального языка. Похоже, что в настоящее время любой товар содержит микроконтроллер. Все вещи стали «умными» и большинство из них также оснащены возможностью телефонной веб-связи. Я готов спорить, что в те времена, когда термин «Bluetooth» ассоциировался только с беспроводными наушниками для сотовых телефонов, вы вряд ли могли представить себе, что в недалеком будущем можно будет купить зубную щетку, оснащенную возможностью Bluetooth.

Принимая все это во внимание, я чувствовал, что настало время для выпуска нового издания книги «Изучаем Arduino». Здесь получили более обширное рассмотрение все темы, содержащиеся в первом издании. Обновлены все проекты добавлением в них новых заданий и подробностей, а также объяснены все проблемные вопросы, которые имелись у читателей первого издания. Но самое главное, в книгу добавлен большой объем нового содержания, включая множество подробностей по беспроводной связи, современному оборудованию для платформы Arduino, изменениям в экосистеме и программном обеспечении Arduino и многое другое.

¹ Англ. *Internet of Things*.

ПОЧЕМУ ARDUINO?

Доступные в настоящее время инструменты, о многих из которых вы узнаете в этой книге, предоставляют нам возможность и средства, чтобы заставить окружающий мир выполнять наши капризы. Еще совсем недавно нельзя было просто взять микроконтроллер и в течение нескольких минут создать устройство на его основе для управления окружающим нас миром. Микроконтроллер представляет собой программируемую интегральную схему (ИС), которая позволяет управлять работой сложных механических, электрических и программных систем с помощью сравнительно простых команд. Возможности в этом поле безграничны, и платформа микроконтроллера Arduino станет вашим любимым инструментом для исследования мира электроники, программирования, взаимодействия человека с компьютером, систем управления и многих других областей. В этой книге вы будете использовать Arduino для реализации самых разнообразных задач, от определения движения до создания беспроводных систем управления и связи через Интернет.

Независимо от вашего уровня опыта в любой технической области, будь то абсолютный новичок или бывалый профессионал, намеревающийся заняться разработкой встроенных систем, платформа Arduino будет замечательной начальной точкой. Или, возможно, вам требуется общая информация касательно разработки с использованием платформы Arduino? Эта книга отлично подойдет и для этой цели. В ней пошагово рассматриваются несколько отдельных проектов, но вы сможете легко возвратиться к изложенному ранее материалу, чтобы позаимствовать фрагменты программного кода, изучить лучшие способы решения задач, свериться с принципиальными схемами и др. Знания в области электротехники, системного проектирования и программирования, которые вы приобретете в процессе изучения этой книги, пригодятся вам во множестве других областей за пределами платформы Arduino и подготовят вас к работе с разнообразными техническими проектами, независимо от того, используют ли они Arduino или какую-либо другую микроконтроллерную платформу.

ДЛЯ КОГО ЭТА КНИГА

Эта книга будет полезной для энтузиастов Arduino с любым уровнем опыта. Каждая новая глава основана на материале предыдущих, содержит понятия и компоненты проектов из них для разработки более сложных идей. Но не волнуйтесь. При каждой встрече с новыми и сложными идеями предоставляются ссылки на первое упоминание соответствующих концепций и конструктивных модулей, чтобы помочь вам освежить свою память.

Предполагается, что читатели этой книги могут не иметь опыта работы в области программирования или электротехники. Учитывая пожелания читателей первого издания этой книги, я уделил особое внимание подробному объяснению тем, которые для некоторых могут показаться излишне сложными. Чтобы помочь читателям с разным уровнем опыта, книга содержит несколько факультативных разделов и врезок, которые более подробно объясняют определенное понятие. Хотя эти дополнительные материалы не обязательны для работы с Arduino, они дают более подробное рассмотрение ряда технических аспектов для более дотошных читателей.

ЧЕМУ ВАС НАУЧИТ ЭТА КНИГА

В этой книге нет готовых рецептов. Поэтому она не подойдет для тех, кто хочет просто следовать пошаговым инструкциям, подробно излагающим все аспекты разработки определенного проекта, но не объясняющим при этом, зачем все это делается. Данную книгу можно рассматривать как введение в электротехнику, вычислительную технику, проектирование изделий, а также в мышление на высоком уровне, используя платформу Arduino для получения практического опыта во всех этих областях.

Собирая устройства на базе Arduino из этой книги, вы научитесь не только выполнять монтаж компонентов, но также сможете разобраться в принципиальных схемах, узнаете, почему те или иные компоненты используются для определенных целей, а также научитесь читать справочные листки компонентов, чтобы выбрать наиболее подходящие детали при создании своих проектов. Для программной части проектов предоставляется полный код программы, но сначала рассматриваются и объясняются несколько фрагментов, образующих конечную программу. Такой подход поможет закрепить понимание конкретных функций программы и способствовать приобретению правильных навыков написания кода и пониманию алгоритмов, лежащих в основе программ.

Настоящая книга научит вас основам физики, алгоритмам, принципам цифрового проектирования, а также концептам программирования, специфичным для платформы Arduino. Я надеюсь, что в процессе реализации изложенных в этой книге проектов вы не только станете грамотным разработчиком устройств на платформе Arduino, но также приобретете навыки, необходимые для создания более сложных электронных систем, а также для инженерной деятельности в других областях и на других платформах.

СТИЛЬ ТИПОГРАФСКОГО ОФОРМЛЕНИЯ

Чтобы привлечь внимание к некоторой наиболее важной или полезной информации, в этой книге используется следующее типографское оформление и форматирование.

Внимание!

Обязательно обращайтесь внимание на оформленные таким образом отступления. Они предупреждают, когда неправильные действия могут повредить электронные компоненты.

Совет

Эти отступления содержат краткие советы, как выполнить текущую задачу легче и эффективнее.

Примечание

Эти отступления содержат дополнительную, потенциально важную, информацию, включая ссылки на видео и другие онлайн-материалы, которые могут помочь вам в разработке определенного проекта.

ПРИМЕР ЗАГОЛОВКА

Такие расширенные отступления предоставляют более подробную информацию по текущей теме или по связанным с ней аспектам.

ПРИБРЕТЕНИЕ КОМПОНЕНТОВ

При подготовке описания проектов в этой книге я приложил большие усилия, чтобы использовать в них компоненты, которые можно легко приобрести во многих магазинах электронных товаров как в Соединенных Штатах, так и в других странах. Я также заключил партнерство с компанией Adafruit (www.adafruit.com), которая является популярным розничным поставщиком электронных компонентов для любителей. Таким образом, все компоненты, необходимые для реализации представленных здесь проектов, можно приобрести у этого поставщика. Для вашего удобства на веб-странице <https://www.exploringarduino.com/kits> размещен список компонентов для проектов каждой главы.

В начале каждой главы также перечислены все компоненты, необходимые для реализации изложенных в ней проектов. Все эти компоненты можно приобрести у многих разных поставщиков. Издательство Wiley также предоставляет разнообразные электронные ресурсы для этой книги на своем

веб-сайте (www.wiley.com/go/exploringarduino2e) и, в частности, ссылки на многочисленные источники, где можно приобрести компоненты для проектов каждой главы.

ЧТО ВАМ ПОТРЕБУЕТСЯ

Кроме самих компонентов для сборки проектов Arduino, вам потребуются несколько инструментов и материалов. Самым важным из них будет компьютер с операционной системой, совместимой с интегрированной средой разработки Arduino IDE (Windows XP или более поздняя версия, Mac OS X 10.7 Lion, или более новая версия, или дистрибутив Linux). Когда необходимо, я буду предоставлять соответствующие инструкции для каждой из этих операционных систем.

В настоящее время доступна онлайн-среда разработки, но в этой книге мы будем в основном концентрироваться на среде IDE для персональных компьютеров. Но все инструкции для среды IDE для настольных компьютеров обычно также применимы и для онлайн-среды IDE. Круг читателей первого издания этой книги был очень обширным, охватывал множество стран с широким диапазоном скорости Интернета и надежности доступа к нему. Чтобы и в дальнейшем обеспечить всем желающим легкий и надежный доступ к платформе Arduino, мои инструкции будут в основном применимы к автономной среде Arduino IDE, поскольку не все имеют постоянный и надежный доступ к Интернету.

Для сборки и отладки аппаратных составляющих проектов в этой книге потребуются дополнительные инструменты. Они не только необходимы для реализации проектов из этой книги, но также пригодятся вам и для других проектов, которые вы, возможно, захотите реализовать, используя ваши знания и опыт электротехники, приобретенные с помощью этой книги. Я могу порекомендовать следующие основные инструменты и расходные материалы:

- паяльник и припой;

Примечание

Некоторые шилды¹ и микроконтроллерные платы, упоминаемые в последних главах данной книги, могут продаваться в виде набора, требующего сборки с помощью пайки, в основном впаивания штыревых контактов в отверстия в печатной плате.

- мультиметр (этот прибор будет полезным для отработки методов отладки, рассматриваемых в книге, но не является абсолютно необходимым);
- набор небольших отверток;
- пинцет;

¹ Это просто калька с английского *shield*, означает модуль или плату расширения. — Прим. пер.

- проволочные кусачки и инструмент для снятия изоляции с проводов;
- клеевой пистолет с нагревом;
- увеличительное стекло (размеры электронных компонентов продолжают уменьшаться, и иногда необходимо прочитать маркировку, нанесенную чрезвычайно мелким шрифтом на интегральных схемах или других компонентах, чтобы можно было посмотреть их характеристики в справочном листке или в Интернете).

ИСХОДНЫЙ КОД И ПРОЧИЕ ЭЛЕКТРОННЫЕ РЕСУРСЫ

Поддерживаемый автором сайт <http://www.exploringarduino.com> специально предназначен для сопровождения этой книги. С него можно загрузить исходный код примеров и проектов для каждой главы, а также видеоуроки, ссылки и другие полезные материалы. Обратите внимание на то, что по этому адресу доступны электронные ресурсы как для первого издания книги, так и для второго. Поэтому убедитесь, что вы переходите по ссылке для правильного издания книги, т. е. второго — **2nd Edition Content**. Электронные ресурсы для первого издания книги находятся по адресу <https://www.exploringarduino.com/content1/>, для второго — по адресу <https://www.exploringarduino.com/content2/>. В меню веб-сайта даются отдельные ясно различимые ссылки на электронные ресурсы разных изданий книги — **1st Edition Content** и **2nd Edition Content** — поэтому у вас не должно возникнуть никаких трудностей с выбором правильного ресурса.

Издательство Wiley также предоставляет электронные ресурсы для этой книги на веб-странице <https://www.wiley.com/en-us/Exploring+Arduino%3A+Tools+and+Techniques+for+Engineering+Wizardry%2C+2nd+Edition-p-9781119405375> в разделе **Downloads**. Ссылку на веб-страницу этой книги на веб-сайте издательства (<https://www.wiley.com>) можно также получить, выполнив на сайте поиск по ее номеру ISBN — 9781119405375.

Файлы архивов ZIP с исходным кодом для каждой главы можно скачать с любого из этих источников. Скачанные файлы нужно будет разархивировать с помощью соответствующей программы, которой оснащены все упомянутые ранее операционные системы.

Наконец, исходный код для проектов из книги также содержится на популярном веб-сайте для обмена открытым кодом GitHub.com (<https://github.com/sciguy14/Exploring-Arduino-2nd-Edition>). Если вам под силу разобраться с управлением версиями на GitHub, то файлы с исходным кодом можно также загрузить из этого хранилища. Ссылки на архив с кодом программ для

каждой главы, расположенный в хранилище GitHub, также предоставляются на веб-сайте <https://www.exploringarduino.com> на страницах соответствующих глав.

Примечание

Поскольку многие книги имеют похожие названия, если у вас возникнет необходимость выполнить поиск данной книги в Интернете, то для этого лучше всего указать не ее название, а код ISBN — 9781119405375.

Примечание

Некоторые адреса URL (особенно те, которые находятся вне моего контроля) могут быть слишком длинными или измениться в будущем. Чтобы было легче вводить длинные адреса URL, которые могут встречаться в книге, они заменяются короткими адресами URL с использованием суффикса сокращения для моего личного веб-сайта: **blum.fyi**. Например, сокращенный адрес URL **blum.fyi/jarvis** перенаправляет на страницу на моем веб-сайте для проекта JARVIS.

СПИСОК ОПЕЧАТОК

Мы прилагаем все усилия, чтобы обеспечить отсутствие ошибок в тексте и коде программ. Но никто не идеален, и ошибки случаются. Если вы обнаружите ошибку в этой книге, например, опечатку или неправильный код, то мы будем благодарны, если вы сообщите нам об этом. Предоставляя нам сведения об ошибках, обнаруженных вами в книге, вы поможете сэкономить другим ее читателям многие часы бесплодных попыток понять, почему определенный проект не работает должным образом. В то же самое время это может помочь нам предоставить вам еще более качественную информацию.

Просмотреть список уже обнаруженных ошибок в книге можно по ссылке **Errata** на веб-странице <https://www.wiley.com/go/exploringarduino2e>. На этой странице можно просмотреть все ошибки в этой книге, о которых нам сообщили и которые были размещены здесь редакторами издательства Wiley. Я также просматриваю все доклады об ошибках и размещаю их на веб-сайте <https://www.exploringarduino.com> на страницах соответствующих глав.

ДОПОЛНИТЕЛЬНЫЙ МАТЕРИАЛ И ПОДДЕРЖКА

В процессе изучения платформы Arduino у вас неизбежно будут возникать разного рода вопросы или, возможно, вам придется столкнуться с какими-либо проблемами. В этом отношении вы сможете оценить один из

самых лучших аспектов платформы Arduino — существование обширного онлайн-сообщества, к которому можно обратиться за советом или помощью в проблемных ситуациях. Эта чрезвычайно активная группа пользователей Arduino с готовностью поможет вам при различных затруднениях. Я веду список обновленных ресурсов поддержки по вопросам по Arduino, электротехнике и встроенному программному обеспечению на странице **Resources** веб-сайта **Exploring Arduino** (<https://www.exploringarduino.com/resources/>).

В прошлом я старался сам отвечать пользователям на их вопросы по Arduino, но, к сожалению, это больше невозможно из-за громадного количества вопросов, которые я получаю на моем веб-сайте, в Twitter, Facebook, на YouTube и других каналах. Я настоятельно рекомендую обращаться за помощью через форумы, ссылки на которые даются на странице **Resources** моего веб-сайта. Я могу почти гарантировать, что они ответят вам быстрее, чем я.

ЧТО ТАКОЕ ARDUINO?

Arduino — это аппаратная и программная платформа, с помощью которой можно создать прототип любого задуманного вами устройства. Это может быть система управления автоматическим поливом, или веб-сервер, или даже автопилот для мультикоптера.

Более конкретно, Arduino — это платформа для разработки устройств на базе микроконтроллера с простым и понятным языком программирования в интегрированной среде разработки Arduino IDE. А оснастив плату Arduino датчиками, приводами, световыми индикаторами, динамиками и более продвинутыми модулями расширения (называемыми *шилдами*), ее можно превратить в программируемый «мозг» для практически любой системы управления.

Трудно даже перечислить все, на что способна платформа Arduino, потому что ее возможности ограничены только вашим воображением. Таким образом, эта книга послужит руководством, знакомящим вас с функциональностью Arduino посредством реализации большого количества устройств, что позволит вам получить навыки, необходимые для разработки своих собственных проектов.

Более подробная информация о плате Arduino и ее разновидностях приводится в *главе 1*. Если вам интересно знать внутреннее устройство Arduino, то вам повезло — это полностью открытая платформа как в аппаратном, так и в программном аспекте, и все схемы и документация находятся в свободном доступе на сайте Arduino. Некоторые технические характеристики Arduino приведены в *Приложении*.

О ДВИЖЕНИИ OPEN SOURCE

Если вы новичок в области открытых источников информации (Open Source), то я рекомендую познакомиться с основными принципами этого движения. Здесь мы не будем вдаваться в подробности, а лишь немного коснемся идеологии данного движения, делающей работу с Arduino такой привлекательной. Получить более полное представление о нем можно на веб-сайте Ассоциации открытого аппаратного обеспечения (Open Source Hardware Association): blum.fyi/OSHW-Definition.

Поскольку Arduino является платформой с открытым аппаратным обеспечением, все проектные файлы, схемы и исходный код для нее доступны для любого желающего. Это означает, что вы можете не только с большей легкостью модифицировать Arduino для решения какой-либо узкой задачи, но также использовать эту платформу и сопутствующее программное обеспечение в своих разработках, производить и продавать клоны платы Arduino и т.п. На рынке предлагаются сотни плат, основанных на Arduino, при этом часто многие эти платы оснащены специализированными функциями.

Лицензия открытой платформы Arduino также разрешает коммерческое использование их разработок (при условии отсутствия торговой марки Arduino на таких разработках). Таким образом, если вы создали на основе Arduino оригинальное устройство и хотите превратить его в коммерческий продукт, то вы можете сделать это.

При всем этом непременно соблюдайте лицензионные требования для исходного кода и аппаратного обеспечения, приведенного в этой книге. Некоторые лицензии содержат требование указывать первого автора при публикации разработок на основе предыдущих работ. Другие же требуют всегда предоставлять доступ к внесенным вами улучшениям в исходный продукт по эквивалентной лицензии. Такое предоставление общего доступа помогает развитию сообщества Arduino и созданию онлайн-документации и поддержки, к которым вам часто придется обращаться в процессе экспериментирования с платформой Arduino. На все примеры программ в этой книге распространяются условия лицензии MIT¹, что позволяет использовать их без ограничений в любых целях.

НЕСКОЛЬКО СОВЕТОВ ЧИТАТЕЛЮ

Некоторые из вас, возможно, уже знакомы с моими популярными видеоуроками по изучению Arduino и основ электроники на канале YouTube

¹ Лицензия открытого программного обеспечения, разработанная Массачусетским технологическим институтом.

(<https://www.youtube.com/sciguy14>). Я постоянно ссылаюсь на них в этой книге, чтобы более подробно осветить изложенные темы. Если вам интересно узнать о том, какие замечательные вещи можно создавать, творчески сочетая электронику, микроконтроллеры и информатику, рекомендую просмотреть мое портфолио с самыми интересными проектами (<https://www.jeremyblum.com/portfolio/>). Подобно платформе Arduino, большинство моих разработок соответствует открытой лицензии, что позволяет вам с легкостью использовать мою работу для ваших собственных нужд.

Мне будет интересно узнать, как вы примените знания и навыки, полученные в результате прочтения этой книги. Я призываю вас поделиться своими открытиями со мной и остальным миром (используйте тег **#ExploringArduino** в социальных сетях). Желаю удачи в ваших экспериментах с Arduino!



Технические основы платформы Arduino

Глава 1. Начало работы и основные сведения
о платформе Arduino

Глава 2. Цифровые входы и выходы
и широтно-импульсная модуляция

Глава 3. Считывание сигналов аналоговых датчиков

Начало работы и основные сведения о платформе Arduino

1

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип A на B для Arduino Uno, тип A на Micro-B для METRO 328).

Исходный код и прочие электронные ресурсы

- Исходный код программ, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch1>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Занимаетесь ли вы электроникой только по выходным, чтобы узнать для себя что-то новое, являетесь ли начинающим инженером-электронщиком или разработчиком программного обеспечения, который хочет лучше понять аппаратную составляющую, на которой исполняется ваш код, — запуск в работу своего первого проекта Arduino непременно будет для вас стимулирующим событием. Прочитав вступление, вы уже получили определенное представление о платформе Arduino и ее возможностях, а теперь мы приступим к подробному изучению этих возможностей. В этой главе мы рассмотрим доступное нам аппаратное обеспечение, ознакомимся со средой разработки и языком программирования, а также напишем и заставим работать нашу первую программу. Когда вы поймете, какие возможности может предоставить платформа Arduino, создание этой простейшей программы для Arduino, которая мигает светодиодом, не будет представлять для вас никакой трудности.

Примечание

Изучение материала этой главы можно сопровождать просмотром видеоурока, представляющего платформу Arduino. Этот видеофайл находится на странице для главы 1 второго издания этой книги по адресу <https://www.exploringarduino.com/content2/ch1>.

ИССЛЕДУЕМ СРЕДУ ARDUINO

Для экспериментов с платформой Arduino необходимо наличие трех основных составляющих:

- платы Arduino (оригинальные или клоны);
- периферийное оборудование (включая как шилды, так и самостоятельно собранные схемы, которые мы будем рассматривать в этой книге);
- интегрированная среда разработки Arduino IDE.

Все эти системные компоненты работают совместно, позволяя реализовать с помощью платформы Arduino практически любую задачу.

Существует большой выбор плат разработки Arduino. Для большинства проектов в этой книге используются оригинальные платы Arduino разработанные одноименной компанией. Но в некоторых проектах последних глав применяются платы-клоны Arduino сторонних разработчиков, позволяющие расширить стандартные возможности Arduino дополнительными функциональностями, как, например, связь Bluetooth и Wi-Fi. Многие платы-клоны Arduino полностью совместимы с программами, библиотеками и аппаратным обеспечением платформы Arduino. Некоторые из этих плат являются точными копиями официальных плат Arduino, тогда как в другие добавлены дополнительные особенности или возможности. Все платы, предназначенные для реализации проектов в этой книге, можно программировать в одной и той же среде IDE. Когда необходимо, мы будем указывать специфичные особенности использования конкретных плат для разных проектов. Большинство проектов в этой книге ориентированы на плату Arduino Uno, поскольку, по сути, она стала платой по умолчанию для начальных этапов изучения Arduino. При этом в любом проекте, в котором используется плата Arduino Uno, ее можно заменить платой Adafruit METRO 328, которая функционально идентична ей. Эту плату можно будет увидеть вместо платы Arduino на некоторых фотографиях и видео, сопутствующих этой книге. В большинстве начальных учебных пособий по Arduino, которые можно найти в Интернете, рекомендуется плата Arduino Uno или ее разновидности. Но если вы предпочтете плату Adafruit METRO 328, для нее, возможно, придется установить драйверы, чтобы Windows определяла ее как Arduino Uno. Эти драйверы можно загрузить по адресу blum.fyi/adafruit-windows-drivers.

Внимание!

Остерегайтесь подделок. Платы Arduino и их клоны следует покупать только у поставщиков с хорошей репутацией (таких как поставщики, упоминаемые в этой книге). На рынке присутствует много компаний, которые изготавливают клоны популярных плат Arduino, используя некачественные компоненты.

Мы начнем наше знакомство с платформой Arduino, исследуя базовые возможности, поддерживаемые всеми платами Arduino. Затем мы рассмотрим различия между всеми современными платами, чтобы вы могли принять обоснованное решение при выборе платы для своего последующего проекта.

БОЛЬШОЙ РАСКОЛ И ПРИМИРЕНИЕ В РЯДАХ ARDUINO

Прежде чем приступить к изучению доступных опций в платформе Arduino, нам нужно разобраться с одним щекотливым вопросом, о котором все знают, но предпочитают обходить стороной. В частности, в период времени между выходом из печати первого и второго изданий этой книги команды разработчиков аппаратного и программного обеспечения Arduino рассорились между собой. Я не буду вдаваться в подробности этой ссоры или вставать на чью-либо сторону, а только изложу сухие факты. В основном компания Arduino разделилась на две команды, представляемые двумя веб-сайтами: Arduino.cc и Arduino.orb. Эти две группы начали выпускать слегка отличающиеся платы, разветвили кодовую базу и оспаривали подлинность оборудования, производимого противной стороной. К счастью, в настоящее время обе стороны этого противостояния нашли общий язык, и мы снова имеем единую платформу Arduino. В настоящей книге я обычно рассматриваю оборудование, предлагаемое группой Arduino.cc, хотя к тому времени, когда вы сможете приобрести эту книгу, обе группы Arduino снова должны объединиться. Если вас интересуют нюансы этой драмы, их можно узнать на веб-сайте организации Hackaday.com (<https://hackaday.com>), которая выпустила ряд докладов на эту тему. А дополнительные подробности о решении этого конфликта можно узнать на веб-странице blum.fyi/arduino-vs-arduino.

Функциональные возможности Arduino

Все платы Arduino поддерживают несколько ключевых функциональных возможностей. Уделите немного времени и исследуйте плату Arduino Uno, показанную на рис. 1.1; это будет нашей базовой конфигурацией.

Общую функциональность платы Arduino можно разбить на следующие функциональные группы:

- **Микроконтроллер** — основной компонент любой платы Arduino. Это «мозги» платы;
- **Программирование** — интерфейсы программирования позволяют загружать программы в плату Arduino;
- **Ввод/Вывод** — схемы ввода-вывода позволяют плате Arduino взаимодействовать с датчиками, приводами и прочими периферийными устройствами;

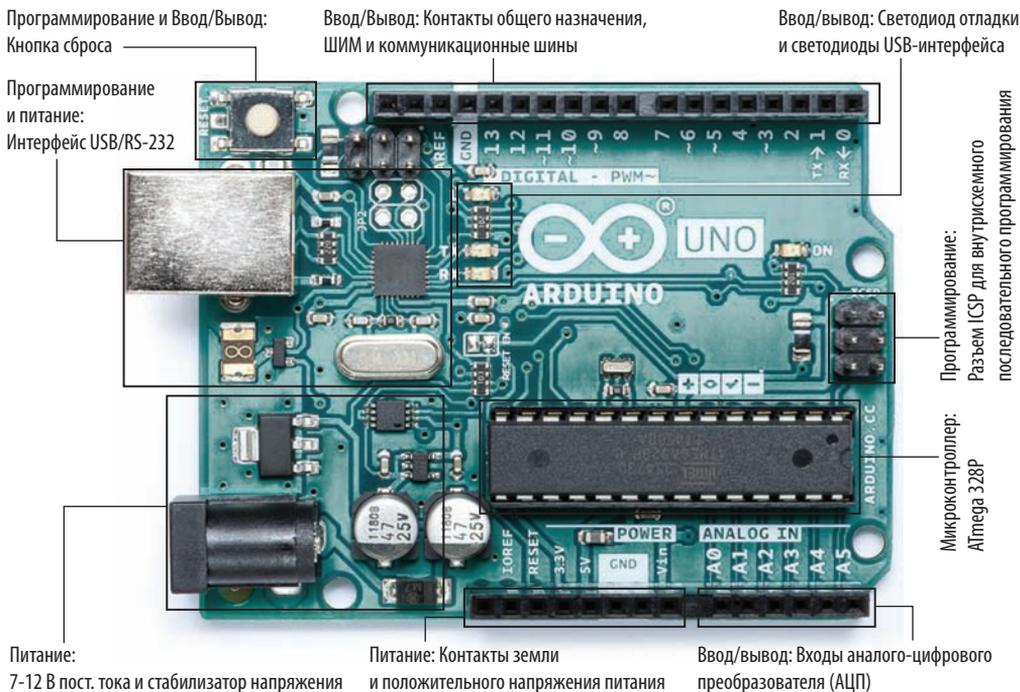


Рис. 1.1. Функциональные группы платы Arduino. Источник: Arduino, arduino.cc; примечания автора

- **Питание** — существуют разные способы подачи питания на плату Arduino. Большинство плат могут автоматически переключаться между несколькими источниками питания (например, от порта USB и от батареи).

Микроконтроллер

Самый важный компонент любой платы Arduino — микроконтроллер. Все первоначальные платы Arduino, включая Arduino Uno, использовали 8-разрядный микроконтроллер ATmega компании Atmel® на основе архитектуры AVR®. Например, в плате Arduino Uno на рис. 1.1 установлен микроконтроллер ATmega 328P. Для большинства проектов, которые вы скорее всего захотите реализовать, простого 8-разрядного микроконтроллера подобного этому будет более чем достаточно. Его мы и будем применять для большинства упоминаний в этой книге.

Примечание

Компания Atmel была недавно поглощена компанией Microchip, производителем микросхем, известным своими микроконтроллерами серий PIC. Производство микросхем ATmega продолжается под этой новой торговой маркой. Поэтому производитель микроконтроллеров ATmega может равнозначно называться Microchip или Atmel. Но, независимо от названия производителя, если микросхемы разных производителей имеют одинаковый номер артикула детали, они функционально идентичны.

Цифровые входы и выходы и широтно- импульсная модуляция

2

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаяная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Кнопка.
- Резистор номиналом 220 Ом (3 шт.).
- Резистор номиналом 10 кОм (1 шт.).
- Светодиод красный диаметром 5 мм.
- Трехцветный светодиод с общим анодом диаметром 5 мм.

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch2>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Мигание светодиодом может наполнить вас чувством гордости, но только на первых порах. Ведь если это все, что можно делать с помощью Arduino, то такое достижение не очень впечатляет. Но мы знаем, что плата Arduino способна на многое другое, и здесь мы продолжим узнавать об этих возможностях. Главное, что делает платформу Arduino таким универсальным инструментом, — возможность как ввода, так и вывода сигналов. При их совместном использовании круг возможных применений платформы почти безграничный. Например, с помощью геркона можно определить момент открытия вашей входной двери и проигрывать какой-либо звуковой сигнал в ответ на это событие: либо сигнал тревоги, либо сигнал приветствия. Или же можно создать электронный замок или устройство для цветомузыки.

В этой главе мы начнем приобретать знания и умения, необходимые для реализации подобных проектов. Мы исследуем возможности ввода цифровых сигналов, применение понижающих и повышающих резисторов и управление выводом цифровых сигналов. Большинство плат Arduino не имеют средств вывода аналоговых сигналов, но для многих задач эти сигналы можно эмулировать с помощью цифровой широтно-импульсной модуляции (ШИМ). В этой главе мы рассмотрим, как генерировать ШИМ-сигналы. Также мы выясним, как устранять дребезг контактов переключателей, что очень важно при считывании сигналов, вводимых пользователями. В конце главы у вас будет достаточно знаний, чтобы собрать и управлять программируемым ночником на трехцветном (красном, зеленом и синем) светодиоде.

Примечание

Излагаемый в этой главе материал по вводу и выводу цифровых сигналов, устранению дребезга контактов, широтно-импульсной модуляции и основным электротехническим понятиям можно сопровождать просмотром моих видеоуроков, в которых я рассматриваю эти темы. Эти видеоуроки находятся на веб-странице <https://www.exploringarduino.com/content2/ch2>.

ВВОД ЦИФРОВЫХ СИГНАЛОВ

В *главе 1* мы научились применять плату Arduino для мигания светодиодом. В этой главе мы продолжим исследовать возможности Arduino по выводу цифровых сигналов, включая следующие темы:

- установка режима вывода для контактов микроконтроллера;
- подключение к плате внешних компонентов;
- знакомство с основными понятиями программирования, включая константы, переменные и циклы;
- вывод цифровых, аналоговых и ШИМ-сигналов.

Подключение светодиода и работа с безопасными макетными платами

Для работы программы мигания светодиодом из *главы 1* кроме платы Arduino нам не нужно было никаких дополнительных компонентов, поскольку управляемый светодиод был встроен в саму плату. Но использование встроенного светодиода существенно ограничивает наши возможности, и поэтому мы рассмотрим, как подключить внешний светодиод к контакту 9 платы Arduino, применив для этого безопасную макетную плату. Подключение внешнего светодиода будет начальным этапом в процессе приобретения знаний, как собирать более сложные внешние схемы, с которы-

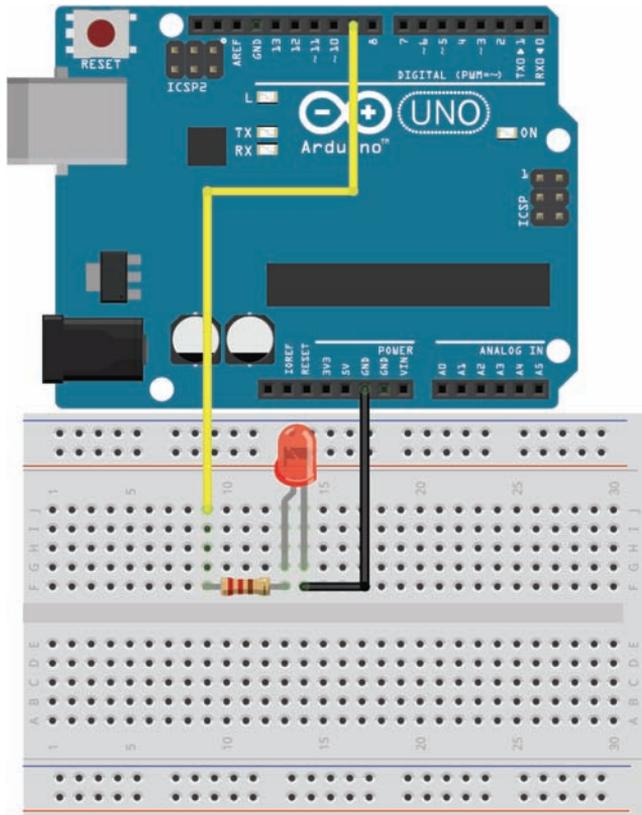


Рис. 2.2. Подключение светодиода к плате Arduino Uno (рисунок создан в программе Fritzing)

Светодиод подключается к контакту 9 разъема платы Arduino. Светодиоды всегда нужно включать с последовательным токоограничивающим резистором. Резисторы большего номинала снижают протекающий ток, в результате чего яркость свечения светодиода уменьшается. В данном примере мы выбрали резистор номиналом 220 Ом. На рис. 2.2 показано, как подключить к плате Arduino Uno светодиод и токоограничивающий резистор.

Закон Ома и уравнение мощности

Самое важное уравнение, которое должен знать каждый инженер-электрик, — закон Ома, описывающий взаимоотношения в схеме между напряжением (в вольтах), током (в амперах) и сопротивлением (в Омах). Схема — это замкнутая электрическая цепь, содержащая источник электрической энергии (например, батарею с напряжением 9 В) и нагрузку (устройство, потребляющее электроэнергию, например, светодиод). Прежде чем приступить к подробному рассмотрению этого закона, объясним, хотя бы поверхностно, значение каждого из составляющих его терминов.

Считывание сигналов аналоговых датчиков

3

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (3 шт.).
- Резистор номиналом 10 кОм (1 шт.).
- Подстроечный потенциометр номиналом 10 кОм.
- Фоторезистор.
- Аналоговый датчик (любой из следующих):
 - ◆ аналоговый датчик температуры TMP36;
 - ◆ ИК-дальномер GP2Y0A21YK0F производства компании Sharp в комплекте с JST-кабелем;
 - ◆ трехосевой акселерометр ADXL335, ADXL377 или ADXL326.
- Светодиод белый диаметром 5 мм.
- Трехцветный светодиод с общим анодом диаметром 5 мм.

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch3>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Нас окружает аналоговый мир. Хотя, возможно, вам приходится часто слышать, что наш мир становится цифровым, природа большинства обозримых явлений в нашей окружающей среде всегда будет аналоговой. Число возможных состояний, которые может принимать окружающий мир,

бесконечно, будь то солнечный свет, температура воды в океанах или концентрация загрязнителей в воздухе. В этой главе мы остановимся на разработке методов преобразования этих бесконечных возможных состояний в приемлемое количество цифровых значений, которые можно анализировать с помощью микроконтроллерной системы, наподобие нашей платформы Arduino.

В ней мы рассмотрим различия между аналоговыми и цифровыми сигналами и узнаем, как преобразовывать сигналы одного типа в другой, применяя для этого несколько аналоговых датчиков, подключаемых к Arduino. В частности, используя навыки, приобретенные в предыдущей главе, мы подключим к плате Arduino светочувствительный датчик для автоматического управления созданным ранее ночником. Мы также научимся работать с последовательным интерфейсом, чтобы посылать аналоговые данные с платы Arduino на компьютер, что откроет нам огромные возможности для разработки более сложных систем для передачи на компьютер данных о состоянии внешней среды.

Примечание

На веб-странице для этой главы (<https://www.exploringarduino.com/content2/ch3>) можно просмотреть видеурок на тему считывания аналоговых сигналов, а также очень подробный видеофильм, в котором рассматриваются различия между аналоговыми и цифровыми сигналами.

АНАЛОГОВЫЕ И ЦИФРОВЫЕ СИГНАЛЫ

Чтобы наши устройства могли взаимодействовать с внешним миром, они должны обладать интерфейсом для обмена аналоговыми данными. Вспомним, как в проектах из предыдущей главы мы использовали кнопочный переключатель для управления светодиодом. Такой переключатель является цифровым устройством ввода, поскольку он может иметь только два выходных состояния: включенное или выключенное, высокий выходной уровень или низкий, 1 или 0 и т.д. Ваш компьютер и подключенная к нему плата Arduino работают с цифровыми данными, которые представляют собой последовательности двоичных (или цифровых) значений, называемых битами. Каждый бит может принимать только одно из двух возможных значений.

Но явления в окружающем нас мире редко можно характеризовать только двумя способами. Взгляните в окно. Что вы там видите? Если это дневное время, то, наверное, вы видите солнечный свет, колышущиеся на ветру деревья и, возможно, проезжающие мимо автомобили или проходящих людей. Все эти воспринимаемые нами явления нельзя легко выразить двоичными данными. Например, солнечный свет не просто есть или нет, а его яркость варьирует

в течение дня. Сила ветра также не характеризуется двумя состояниями, ветер дует порывами с разной скоростью и в разных направлениях.

Сравнение аналоговых и цифровых сигналов

На рис. 3.1 показаны осциллограммы цифрового (слева) и аналогового (справа) сигналов.

Прямоугольный сигнал на левой осциллограмме принимает только два значения: 0 и 5 вольт. Подобно выходному сигналу кнопки в предыдущей главе, эти значения соответствуют низкому и высокому логическим уровням. Справа на рисунке изображена осциллограмма синусоидального сигнала. Хотя значения этого сигнала ограничены теми же пределами (0 и 5 В), между этими двумя граничными уровнями он может принимать бесконечное число значений.

Аналоговые сигналы, подобные изображенному на рис. 3.1, нельзя выразить дискретно, поскольку в пределах определенного диапазона они могут принимать теоретически бесконечное множество значений. Возьмем, например, в качестве аналогового сигнала, который мы хотим измерить, яркость солнечного света. Очевидно, что его яркость, или создаваемая им освещенность (измеряемая в люксах — световом потоке на единицу площади), ограничена определенным разумным диапазоном значений. В частности, можно обоснованно ожидать, что наши измерения будут находиться в диапазоне от 0 люкс (полная темнота) до 130 000 люкс под прямым солнечным светом. Если бы мы обладали измерительным устройством с бесконечной точностью, то могли бы измерить бесконечное число значений освещенности между этими двумя крайними величинами. Например, освещенность внутри помещения

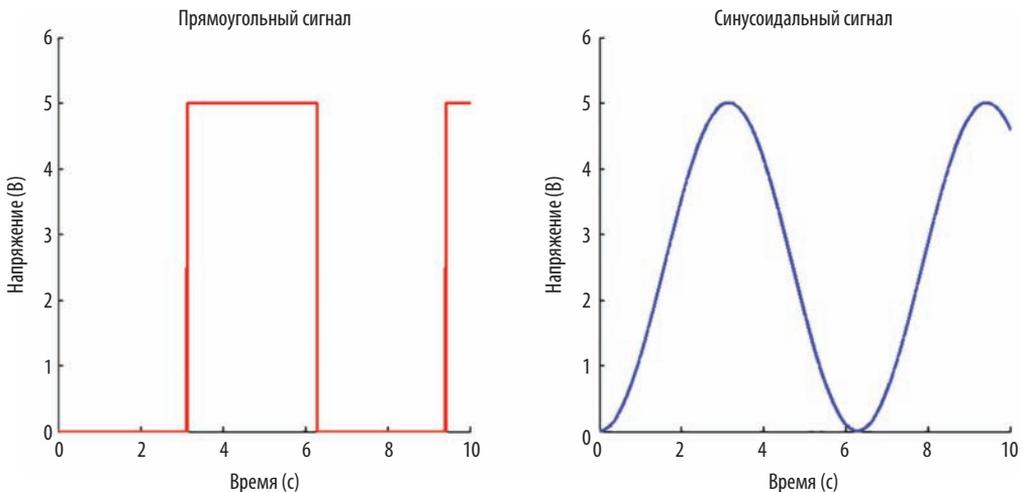


Рис. 3.1. Осциллограммы аналогового и цифрового сигналов (рисунок создан в программе MATLAB)



Взаимодействие с окружающей средой

Глава 4. Использование транзисторов и управление электродвигателями постоянного тока

Глава 5. Управление сервоприводами и шаговыми двигателями

Глава 6. Работаем со звуком

Глава 7. Последовательный интерфейс USB

Глава 8. Эмуляция USB-устройств

Глава 9. Сдвиговые регистры

Использование транзисторов и управление электродвигателями постоянного тока

4

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Макетная плата половинного размера.
- Набор проволочных перемычек.
- Резистор номиналом 1 кОм (1 шт.).
- Резистор номиналом 10 кОм (2 шт.).
- Подстроечный потенциометр номиналом 10 кОм (1 шт.).
- Фоторезистор (2 шт.).
- Батарея напряжением 9 В (1 шт.).
- Разъем для батареи напряжением 9 В (1 шт.).
- Микросхема L7805CV стабилизатора напряжения 5 В (1 шт.).
- Электролитический конденсатор, 10 мкФ, 50 В (2 шт.).
- Керамический конденсатор 0,1 мкФ (1 шт.).
- Диод 1N4001 (1 шт.).
- Биполярный NPN-транзистор PN2222 (1 шт.).
- Набор для сборки самоходного шасси робота с колесами и электродвигателями (1 шт.).
- Электродвигатель постоянного тока на 9 В (1 шт.).
- Микросхема сдвоенного H-моста драйвера электродвигателя TI L293D.

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch4>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Теперь мы знаем, как можно получить информацию о некоторых аспектах окружающего нас мира. Но как нам можно управлять этим миром? Проекты мигания светодиодом и автоматического управления ночником положили хорошее начало в этом направлении, но существующие возможности намного обширнее. С помощью транзисторов к плате Arduino можно подключать разные типы электродвигателей и приводов, что позволит ей выполнять различные физические действия. В частности, подключив к плате Arduino электродвигатели, мы сможем конструировать мобильные роботы, создавать роботизированные манипуляторы, расширить область действия разных датчиков и осуществить еще многое другое.

В этой главе мы рассмотрим, как управлять такими индуктивными нагрузками, как электродвигатели постоянного тока, как включать и выключать силовоточные устройства с помощью транзисторов, а также как использовать в своих проектах интегральные схемы. В конце этой главы мы соберем самоходное шасси, которым можно управлять дистанционно с помощью обычного фонарика.

Примечание

Дополнительная информация по электродвигателям и транзисторам предоставляется в видеоуроке на веб-странице для этой главы (<https://www.exploringarduino.com/content2/ch4>).

Внимание!

В проектах этой главы используется батарея напряжением 9 В для питания электродвигателей, требующих больше тока, чем может обеспечить плата Arduino. Это не настолько высокое напряжение, чтобы быть опасным для вас, но при неправильном подключении может возникнуть опасность для электронных компонентов ваших устройств. В процессе выполнения упражнений в этой главе внимательно следуйте инструкциям и схемам к ним. Не допускайте коротких замыканий (подключения положительного питания напрямую на землю). При соединении вместе общих шин нескольких источников питания ни в коем случае не соединяйте при этом вместе шины положительного питания этих источников. Например, в проектах шина земли источника питания напряжением 9 В подключается к шине земли источника питания для Arduino напряжением 5 В. Но положительные шины этих двух источников питания нельзя подключать к одной и той же шине положительного питания на макетной плате. Это выведет из строя стабилизатор напряжения 5 В на плате Arduino, что в свою очередь повредит микроконтроллер платы.

ЭЛЕКТРОДВИГАТЕЛИ ПОСТОЯННОГО ТОКА

Электродвигатели постоянного тока, которые используются в многочисленных бытовых приборах, вращаются с постоянной скоростью при подаче на них питания постоянного тока. Эти электродвигатели широко применя-

Сборка устройства

Этот проект можно легко реализовать с помощью компонентов схемы и набора для сборки самоходного шасси, предоставляемых компанией Adafruit. В этом разделе мы рассмотрим, как собрать такое шасси и установить на него компоненты схемы управления. Но если вы желаете сконструировать свою собственную платформу, то это будет даже лучше. Ее можно отпечатать на 3D-принтере или сделать из подручных средств. Но независимо от подхода для создания платформы самоходного шасси, все равно настоятельно рекомендуется использовать электродвигатели с редукторами и колесами, предоставляемые компанией Adafruit.

Сборку начинаем с крепления электродвигателей с редукторами к платформе. Для этого в комплекте с покупным набором шасси предусмотрены винты и гайки, для шасси вашей собственной конструкции подойдут и другие средства, например, клей. Затем надеваем колеса на валы редукторов. Установите на платформу шасси макетную плату со схемой управления и подключите электродвигатели к ней. Если потом обнаружится, что один или оба электродвигателя вращаются не в том направлении, просто поменяйте полярность их подключения к питанию. На рис. 4.13 показано, как может выглядеть собранное самоходное шасси (тарелочные «глаза» устанавливаются по желанию).

В процессе сборки самоходного шасси следует иметь в виду несколько аспектов. Если используется платформа компании Adafruit (или любая другая

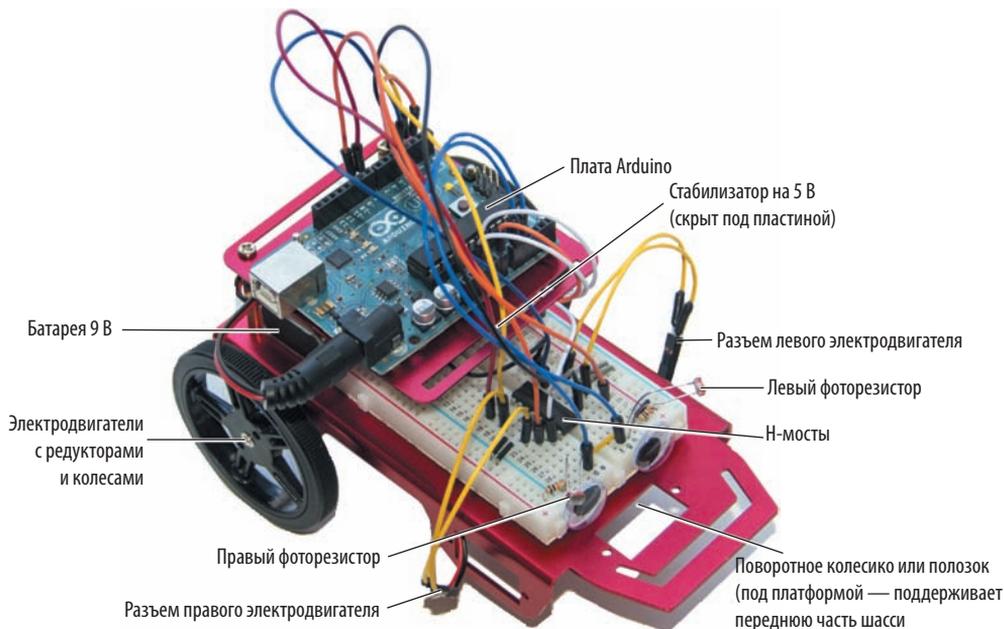


Рис. 4.13. Полностью собранное самоходное шасси

металлическая платформа), обязательно изолируйте нижнюю сторону платы Arduino от платформы. Из рис. 4.13 видно, что я решил эту задачу, установив плату Arduino на винтовые стойки с пластмассовыми прокладками, чтобы она не прикасалась к металлической пластине. Даже если ваша платформа анодированная, плату Arduino необходимо изолировать от нее с помощью прокладок или простой изоляционной ленты. Острые концы выводов на нижней стороне платы Arduino могут повредить покрытие, в результате чего на плате возникнет короткое замыкание. Также следует быть осторожным с конденсаторами, стабилизатором напряжения и другими компонентами с длинными выводами, установленными на макетную плату. Если требуется согнуть выводы таких компонентов, чтобы установить их в гнезда макетной платы, следите за тем, чтобы случайно не закоротить их.

Завершив сборку шасси, подключите к схеме батарею 9 В и с помощью фонарика запустите устройство в движение. В идеале шасси должно следовать за направлением луча фонарика. Если скорость шасси окажется слишком низкой, или схема будет слишком чувствительной к свету или, наоборот, недостаточно чувствительной, или вообще шасси поведет себя странным образом, подключите схему к компьютеру через порт USB, запустите программу монитора порта и проанализируйте выводимые в ее окне значения освещенности и скорости электродвигателей. Может потребоваться дополнительно откорректировать пороговые значения этих параметров, как описывалось ранее, или же отрегулировать положение фоторезисторов. Если уровень освещенности комнаты слишком высокий, то может потребоваться выключить свет и/или закрыть шторами окна, чтобы схеме было легче распознавать свет фонарика. Наконец, если какой-то из двигателей вращается не в том направлении, просто поменяйте полярность его питания.

РЕЗЮМЕ

В этой главе мы узнали следующее.

- Электродвигатели постоянного тока используют электромагнитную индукцию для преобразования изменений тока, протекающего в обмотке, во вращательное механическое движение.
- Электродвигатели представляют собой индуктивную нагрузку, и для безопасного сопряжения их с платой Arduino необходимо предусмотреть соответствующие схемы защиты и питания.
- Скоростью и направлением вращения электродвигателя постоянного тока можно управлять посредством сигнала ШИМ и H-моста.
- Подключив к плате Arduino аналоговые датчики и приводы, например, электродвигатели, можно создавать интерактивные проекты, в том числе управляемые роботы.

Управление сервоприводами и шаговыми двигателями

5

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Кнопки (2 шт.).
- Резистор номиналом 1 кОм (4 шт.).
- Подстроечный потенциометр номиналом 10 кОм (1 шт.).
- Светодиод синего цвета диаметром 5 мм (4 шт.).
- Батарея 9 В (1 шт.).
- Разъем для батареи напряжением 9 В (1 шт.).
- Микросхема L7805CV стабилизатора напряжения 5 В (1 шт.).
- Электролитический конденсатор, 10 мкФ, 50 В (2 шт.).
- Микросхема сдвоенного Н-моста драйвера электродвигателя TI L293D.
- Сетевой адаптер, 12 В (рабочий ток >500 мА).
- ИК-дальномер GP2Y0A21YK0FIR компании Sharp в комплекте с JST-кабелем.
- Стандартный серводвигатель с рабочим напряжением 5 В.
- Биполярный шаговый двигатель NEMA-17.
- Термоклей или клейкая лента.
- Круглый «циферблат» (можно использовать компакт-диск или вырезать из плотной бумаги).
- Зажим для бумаг.
- Палочка от мороженого.

Исходный код и прочие электронные ресурсы

- Исходный код, видеуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch5>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

В главе 4 мы научились управлять электродвигателями постоянного тока. Такие электродвигатели хорошо подходят в качестве тяговых. Но для точного позиционирования они не годятся, поскольку у них отсутствует встроенный механизм обратной связи и регулируется скорость вращения вала, а не его позиция. Чтобы точно знать абсолютное положение ротора электродвигателя, требуется какое-либо внешнее кодирующее устройство или система позиционирования. И наоборот, сервоприводам можно дать команду повернуть вал на определенный угол, в котором он будет находиться до поступления следующей команды. Такая возможность ценна тем, что она позволяет переместить определенную систему в известное положение. Сервоприводы используются для решения таких задач, как, например, перемещение защелки замка, вращение якоря на определенное число оборотов или установка точного размера диафрагмы объектива. Шаговые двигатели также позволяют точно управлять углом поворота своего ротора. Это делает их идеальными для применения в таких устройствах, как 3D-принтеры, прецизионные инструменты и измерительные приборы. В этой главе мы рассмотрим принципы работы как сервоприводов, так и шаговых двигателей. Управлять этими устройствами будем с помощью платы Arduino.

УПРАВЛЕНИЕ СЕРВОПРИВОДАМИ

Сервоприводы широко применяются как в любительской, так и профессиональной робототехнике. Их можно встретить в разнообразных устройствах — от радиоуправляемых моделей самолетов до дверных замков, управляемых через Интернет. Они доступны в широком диапазоне размеров и возможностей. Существует два основных типа сервоприводов: кругового вращения и с поворотом ротора в пределах установленного сектора.

Разница между сервоприводами кругового вращения и с поворотом в пределах сектора

Стандартные сервоприводы имеют ограниченный угол вращения вала (обычно от 0 до 180 градусов), поскольку их вал соединен с потенциометром,

который выдает информацию о текущем положении вала. Управление сервоприводом осуществляется подачей на него импульса определенной длительности. Длительность управляющего импульса определяет угол поворота вала сервопривода. Но если из сборки сервопривода удалить потенциометр, то вал сможет вращаться вкруговую, и в таком случае длительность импульса задает скорость вращения вала сервопривода.

В проектах, описанных в этой книге, мы будем использовать стандартные сервоприводы, вал которых проворачивается только на заданный угол в пределах определенного сектора. Если у вас есть желание поэкспериментировать с сервоприводом кругового вращения, можно переделать стандартный сервопривод, вскрыв его корпус и удалив из него потенциометр. Или же можно купить сервопривод, который уже модифицирован таким образом.

Управление сервоприводом

В отличие от электродвигателей постоянного тока, у сервоприводов три вывода: для подачи положительного питания (обычно красного цвета), земли (обычно черного или коричневого цвета) и сигнала управления (обычно белого или оранжевого цвета). На рис. 5.1 показан пример сервоприводов и их выводов. Но у двигателей некоторых производителей другая маркировка проводов, поэтому всегда сверяйтесь со справочным листком, чтобы правильно подключить сервопривод.

Разные производители сервоприводов могут применять слегка отличающиеся цветовые схемы выводов, но наиболее распространена маркировка, описанная выше. (Но если вы не уверены насчет своего сервопривода,

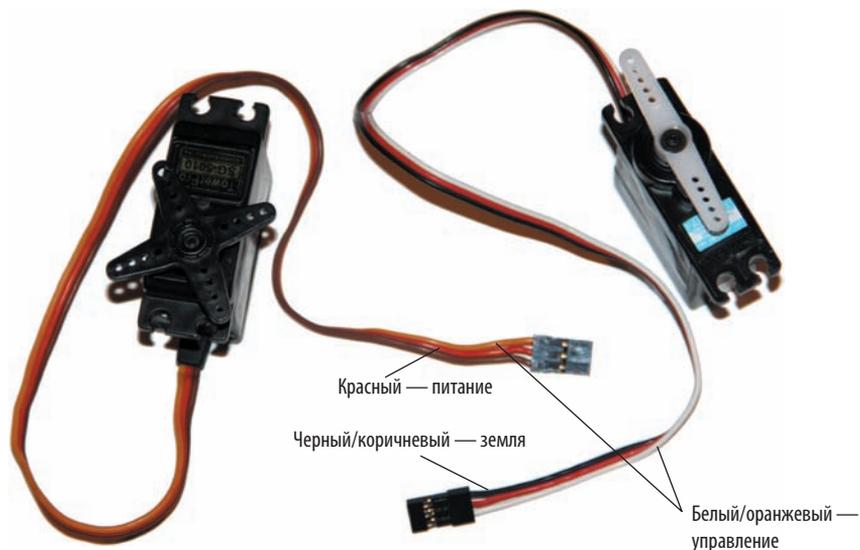


Рис. 5.1. Примеры серводвигателей и их выводов с цветовой маркировкой

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Кнопки (5 шт.).
- Резистор номиналом 220 Ом (1 шт.).
- Резистор номиналом 10 кОм (5 шт.).
- Потенциометр номиналом 10 кОм (1 шт.).
- Динамик с сопротивлением катушки 8 Ом (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch6>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Как известно, люди обладают пятью органами чувств. Из этих пяти мы не будем пытаться создать интерфейс с электронными устройствами для чувства вкуса — вряд ли у кого-то возникнет мысль облизывать плату Arduino. Мы не будем ничего изобретать и для чувства запаха. Единственное, что можно порекомендовать при появлении какого-либо запаха от ваших электронных устройств, — это немедленно прекратить текущий эксперимент и проверить, не горит ли какой-либо компонент. Остаются чувство осязания, зрение и слух. Мы уже создавали интерфейс для чувства осязания с помощью потенциометров и кнопок и для зрения с помощью светодиодов. А какой интерфейс можно создать для нашего чувства слуха? В этой главе

мы рассмотрим, как с помощью Arduino воспроизводить звуки, чтобы сделать более легким решение задачи обратной связи для наших проектов.

Существует несколько способов создавать звуки с помощью платформы Arduino. Наиболее простой — использовать функцию `tone()`, которой в этой главе уделяется самое большое внимание. Но существуют также различные шилды, которые, благодаря их дополнительным вычислительным возможностям, расширяют музыкальные возможности базовой платформы Arduino. (Шилдами на жаргоне Arduino называются платы расширения, которые вставляются сверху в разъемы платы Arduino, чтобы придать ей ту или иную дополнительную функциональность.) Если у вас плата Arduino Due, то генерировать звуки можно будет с помощью встроенного цифроаналогового преобразователя (ЦАП).

ПРИНЦИП РАБОТЫ ДИНАМИКА

Прежде чем приступать к созданию звуковых эффектов с помощью Arduino, нам нужно разобраться, что собой представляют звуки и как люди воспринимают их. В этом разделе мы рассмотрим, как возникают звуковые волны, каковы их свойства и как, манипулируя этими свойствами, можно создавать музыку, речь и т.п.

Свойства звука

Звук передается через воздух в виде волны повышенного давления. Вибрирующий объект, например, динамик, барабан или колокольчик, передает вибрации на соприкасающийся с ним воздух. В свою очередь, эти частицы воздуха передают полученную от объекта энергию другим окружающим их частицам воздуха, в результате чего те также начинают вибрировать. Эти вибрации создают волну повышенного давления, которая в конечном итоге доходит до нашей ушной барабанной перепонки и воздействует на нее. Посредством такой цепной реакции вибрирующих частиц и происходит распространение звука от его источника до приемника. Но зачем нам нужно знать все это, чтобы научиться создавать звуки с помощью платы Arduino?

Посредством Arduino мы можем управлять двумя свойствами этой волны вибрирующих частиц: ее частотой и амплитудой. Частота волны вибрирующих частиц представляет скорость колебаний частиц воздуха туда и обратно, а амплитуда — размах этих колебаний. В физическом аспекте звуки большей амплитуды громче, чем меньшей. Тон высокочастотных звуков выше (например, сопрано), тогда как низкочастотных — ниже (например, бас). На рис. 6.1 изображены синусоидальные звуковые волны с разными амплитудами и частотами.

Последовательный интерфейс USB

7

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (3 шт.).
- Подстроечный потенциометр номиналом 10 кОм (1 шт.).
- Светодиод красный диаметром 5 мм (1 шт.).
- Трехцветный светодиод с общим анодом диаметром 5 мм (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch7>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Вероятно, самая важная функциональность платформы Arduino — возможность программирования платы через последовательный порт USB. Платы Arduino можно запрограммировать, не применяя никакого дополнительного специального оборудования, например, программатора AVRISP mkII. Обычно для программирования микроконтроллеров требуется специальное внешнее устройство (как только что упомянутый программатор), через которое программируемый микроконтроллер подключается к компьютеру. В случае с платформой Arduino такой программатор, по сути, встроен в саму плату. Кроме того, этот программатор также предоставляет прямой доступ к встроенному в микроконтроллер универсальному синхронному/асинхронному приемопередатчику (УСАПП). Через этот интерфейс можно

обмениваться данными между компьютером и платой Arduino или между платой Arduino и другими устройствами, оснащенными таким интерфейсом, включая другие платы Arduino.

В этой и следующей главах мы рассмотрим практически все, что нужно знать о подключении платы Arduino к компьютеру через порт USB и обмене данными между ними по этому каналу.

ВОЗМОЖНОСТИ ПОСЛЕДОВАТЕЛЬНОГО ОБМЕНА ДАННЫМИ ПЛАТФОРМЫ ARDUINO

Разные платы Arduino реализуют последовательный обмен данными разными способами. Это касается как аппаратной реализации адаптеров USB/RS-232, так и программной поддержки различных возможностей. В этом разделе мы рассмотрим разные аппаратные интерфейсы последовательного обмена данными, предлагаемые на различных платах Arduino.

Примечание

Дополнительная информация по последовательной связи предлагается в видеоуроке на веб-странице для этой главы по адресу <https://www.exploringarduino.com/content2/ch7>.

Для начала нам нужно понимать разницу между последовательным портом и портом USB. Тем, кто помоложе, возможно, даже никогда и не приходилось иметь дело с последовательным портом (или портом RS-232), поскольку этот интерфейс был довольно давно заменен на USB. На рис. 7.1 показано, как выглядит стандартный разъем последовательного порта.

Самые первые платы Arduino оснащались разъемом последовательного порта, который позволял подключать плату к компьютеру с помощью 9-жильного кабеля. В настоящее время таким разъемом оснащаются разве что специализированные компьютеры, но на рынке предлагаются адаптеры, позволяющие подключать устройства RS-232 к порту USB компьютера.



Рис. 7.1. Разъем последовательного порта
(Источник: Википедия. Общественный доступ)

Микроконтроллер ATmega 328P, на котором основана плата Arduino Uno, оснащен встроенным аппаратным портом RS-232. На плату Arduino выведены его линии передачи (TX) и приема (RX), которые подключены к цифровым контактам 0 и 1 платы.

Как уже упоминалось в разделе «*Настройка загрузчика и встроенной программы Arduino*» главы 1, плата Arduino оснащена загрузчиком, позволяющим загружать в нее программы через последовательный интерфейс. Эти контакты мультиплексированы, и кроме последовательного интерфейса они применяются для обслуживания других функциональностей. Посредством специальной адаптерной схемы эти контакты также подключены к линиям передачи и приема разъема USB платы. Но хотя оба интерфейса (USB и RS-232) являются последовательными, они несовместимы напрямую друг с другом. Поэтому для обеспечения возможности взаимодействия платы Arduino с другими устройствами через интерфейс USB используется один из двух вспомогательных способов. Первый способ состоит в преобразовании с помощью дополнительной интегральной схемы протокола RS-232 платы Arduino в протокол USB компьютера, и наоборот. Эта микросхема может встраиваться в плату Arduino или поставляться отдельно с адаптерной платой или кабелем. Этот подход применяется в плате Arduino Uno, в которой встроенная микросхема выполняет преобразование между протоколами USB и RS-232. Второй способ состоит в использовании микроконтроллера со встроенным контроллером USB. Этот подход применяется, например, в плате Arduino Leonardo, оснащенной микроконтроллером 32U4.

Использование встроенной или внешней микросхемы преобразователя USB/RS-232 компании FTDI или Silicon Laboratories

Как упоминалось в предыдущем разделе, многие платы Arduino (а также их клоны) оснащены дополнительной микросхемой для преобразования между интерфейсами USB и RS-232. Одна из наиболее популярных таких микросхем — CP210x производства компании FTDI или Silicon Laboratories. При подключении этой микросхемы к компьютеру через разъем USB она отображается в разделе **Порты** диспетчера устройств как виртуальный последовательный порт COM, к которому можно обращаться, как к обычному последовательному порту RS-232. На рис. 7.2 показана плата Arduino Nano, в которой есть встроенная микросхема преобразователя USB/RS-232 компании FTDI (обведена белой линией).

В плате METRO 328 от компании Adafruit (которая подойдет вместо платы Arduino Uno для реализации проектов в этой книге) также имеется встроенная микросхема преобразователя CP2104.

Эмуляция USB-устройств

8

Список деталей и оборудования для проектов этой главы

- Плата Arduino Leonardo или Seeeduno Lite или Pololu A-Star 32U4 Prime LV.
- USB-кабель (тип A на Micro-B).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Кнопки (3 шт.).
- Резистор номиналом 220 Ом (1 шт.).
- Резистор номиналом 10 кОм (3 шт.).
- Фоторезистор (1 шт.).
- Светодиод красный диаметром 5 мм (1 шт.).
- Аналоговый датчик температуры TMP36 (1 шт.).
- Двухкоординатный джойстик (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch8>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

В главе 7 мы экспериментировали с последовательной передачей данных USB/RS-232 между компьютером и платой Arduino. Для этого мы подключали компьютер к плате Arduino через последовательный интерфейс, что позволяло любой программе, способной взаимодействовать с последовательным портом, «общаться» с платой Arduino. Такой метод вполне пригоден для обычного обмена данными, но он весьма далек от использования всех возможностей собственно USB-интерфейса.

Интерфейс USB является фактическим международным стандартом для подключения компьютерных периферийных устройств. Возможности этого интерфейса постоянно расширяются в результате добавления новых функций. Например, стандарт USB SuperSpeed и разъем USB-C позволяют передавать данные, видео высокого разрешения, обеспечивают напряжение, достаточное для зарядки аккумуляторной батареи подключенных устройств, и многое другое. Компьютер может определять разнообразные типы подключенных к нему USB-устройств. В этой главе мы рассмотрим, как платы Arduino, оснащенные встроенной поддержкой USB, могут действовать как устройства человеко-машинного интерфейса HID.

Примечание

Для упражнений в этой главе требуется плата Arduino со встроенной в основной микроконтроллер поддержкой USB-интерфейса, например, Arduino Leonardo. Платы Arduino, в которых поддержка USB-интерфейса обеспечивается дополнительной микросхемой, например, Arduino Uno или Adafruit METRO 328, для этих экспериментов не годятся. Как отмечалось в списке компонентов, клоны платы Arduino Leonardo предлагаются несколькими сторонними производителями. В частности, кроме платы Arduino Leonardo для упражнений в этой главе подойдет плата Seeeduno Lite (поставщик Adafruit) или A-Star 32U4 Prime LV (поставщик Pololu). Поставщики иногда могут испытывать трудности с получением оригинальных плат Arduino и могут предлагать платы-клоны. Но если клоны предлагает поставщик с хорошей репутацией, например, Adafruit или Pololu, то качеству таких плат-клонов можно доверять. Если вы используете плату Seeeduno Lite, установите ее переключатель напряжения на 5 В. Для программирования плат-клонов в меню выбора платы среды Arduino IDE нужно указать плату Arduino Leonardo.

Подобно другим платам Arduino, в которых поддержка USB встроена в главный микроконтроллер, плата Arduino Leonardo может эмулировать такие устройства, как клавиатура или мышь. В этой главе мы рассмотрим, как использовать плату Arduino Leonardo для эмуляции этих устройств.

Совет

При реализации примеров этой главы следует соблюдать осторожность, поскольку иногда могут возникнуть проблемы с перепрограммированием платы. Например, если загруженный в плату скетч эмулирует мышь и безостановочно перемещает курсор мыши по экрану, это может создать трудности с нажатием кнопки **Загрузить** в среде Arduino IDE. В таких случаях, чтобы перепрограммировать плату, нажмите и отпустите на ней кнопку сброса, одновременно нажав кнопку загрузки скетча в среде Arduino IDE. Таким образом плата будет оставаться в режиме работы загрузчика, что позволит начать ее программирование.

Драйверы для платы Arduino (или эквивалентной платы-клона) должны установиться автоматически при первом подключении платы к компьютеру. Но на некоторых компьютерах с ОС Windows с этим могут возникнуть проблемы. В таком случае следуйте инструкциям по ручной установке драйверов, предоставляемой на веб-сайте Arduino по адресу blum.fyi/installing

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаячная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (8 шт.).
- Светодиод красный диаметром 5 мм (8 шт.).
- Светодиод зеленый диаметром 5 мм (4 шт.).
- Светодиод желтый диаметром 5 мм (3 шт.).
- ИК-дальномер GP2Y0A21YK0FIR компании Sharp в комплекте с JST-кабелем (1 шт.).
- Микросхема сдвигового регистра SN74HC595N (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch9>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Несомненно, что в процессе работы над проектами из этой книги вы уже мысленно составили себе список своих собственных будущих проектов. Вполне возможно, что некоторые из них вам пришлось удалить из списка, поскольку плата Arduino просто не обладает достаточным количеством контактов для их реализации. Одно из самых распространенных применений платформы Arduino состоит в украшении светодиодными гирляндами всех более-менее подходящих для этого предметов. По гирлянде в каждый угол комнаты! Гирлянду на компьютер! По гирлянде на кота и собаку! Ну, на животных, наверное, не нужно.

Однако с реализацией этих проектов может получиться незадача. Как сделать гирлянду из, скажем, 50 мигающих светодиодов (или управлять большим количеством каких-либо других устройств), если плата, например, Arduino Uno, оснащена только 14 цифровыми контактами? Но эта кажущаяся трудность в действительности не представляет никакой проблемы, поскольку она с легкостью решается посредством использования сдвиговых регистров. Сдвиговые регистры позволяют расширить возможности ввода-вывода платы Arduino без необходимости приобретения более дорогой платы с микроконтроллером с большим количеством контактов ввода-вывода. В этой главе мы рассмотрим принципы работы сдвиговых регистров, а также реализуем как программы, так и схемы, необходимые для сопряжения платы Arduino со сдвиговыми регистрами с целью расширения ее возможностей цифрового вывода. После выполнения упражнений этой главы вы сможете самостоятельно применять сдвиговые регистры, а также будете обладать знаниями, необходимыми для принятия грамотных решений при разработке проектов с большим количеством цифровых выводов.

ВЫБОР ПЛАТЫ ARDUINO ДЛЯ РЕШЕНИЯ ЗАДАЧ

Как и в большинстве предыдущих глав, в этой главе в качестве платформы для разработки используется плата Arduino Uno (или эквивалентная плата Adafruit METRO 328). Для реализации упражнений в этой главе подойдет любая другая плата Arduino, но будет полезно рассмотреть факторы, вследствие которых предпочтительней выбрать для определенного проекта одну плату Arduino, а не другую. Например, вы, наверное, уже задаетесь вопросом, почему бы просто не взять плату Arduino с большим количеством контактов ввода-вывода, например, Mega 2560 или Due. Конечно же, такой подход был бы вполне логичным для реализации проектов, требующих большего количества контактов ввода-вывода. Но как инженер при разработке проекта вы всегда должны думать о других аспектах. Например, вычислительных мощностей платы Uno может быть вполне достаточно для проекта, но ее контактов ввода-вывода не хватает. В таком случае значительно дешевле добавить несколько сдвиговых регистров, чем задействовать плату более высокого уровня. Такая конструкция будет более компактной. Но за это придется расплатиться усложнением программы и, возможно, более длительной ее отладкой.

ЧТО ТАКОЕ СДВИГОВЫЙ РЕГИСТР

Сдвиговый регистр представляет собой устройство, которое принимает поток последовательных битов и одновременно выводит значения этих битов на параллельные выходные контакты. Сдвиговые регистры часто



Интерфейсы для обмена данными

Глава 10. Шина I²C

Глава 11. Шина SPI и библиотеки сторонних разработчиков

Глава 12. Взаимодействие с жидкокристаллическими дисплеями

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаяечная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (8 шт.).
- Резистор номиналом 4,7 кОм (2 шт.).
- Светодиод красный диаметром 5 мм (1 шт.).
- Светодиод зеленый диаметром 5 мм (4 шт.).
- Светодиод желтый диаметром 5 мм (3 шт.).
- Микросхема сдвигового регистра SN74НС595N (1 шт.).
- Датчик температуры ТС74А0-5.0VAT (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch10>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Мы уже умеем выполнять сопряжение с платой Arduino аналоговых и цифровых входных и выходных сигналов, но как насчет подключения более сложных устройств? Возможности платы Arduino (и, собственно говоря, любого микроконтроллера) можно расширить, подключая к нему разные внешние компоненты. В частности, упростить связь между микроконтроллером и разнообразными модулями можно с помощью многих микросхем, реализующих стандартные протоколы цифровой связи. В этой главе мы рассмотрим использование шины I²C.

Шина I²C позволяет реализовать надежную, высокоскоростную, двухстороннюю связь между двумя устройствами при минимальном количестве контактов ввода-вывода, чтобы упростить этот процесс. В зависимости от используемых компонентов и особенностей системы, скорость обмена данными по этой шине составляет от 100 килобит до нескольких мегабит в секунду. К шине I²C подключается одно ведущее устройство (обычно микроконтроллер или микропроцессор), осуществляющее управление шиной, а также одно или больше ведомых устройств, которые получают данные от ведущего устройства. В этой главе мы рассмотрим протокол I²C, а затем реализуем его для взаимодействия с цифровым датчиком температуры, возвращающим показания в значениях температуры, а не в произвольных аналоговых значениях. В этом мы будем полагаться на наши знания, полученные в предыдущих главах, добавив сведения из этой главы, чтобы расширить возможности наших предыдущих проектов.

Примечание

На веб-странице электронных ресурсов книги для этой главы (<https://www.explorin-garduino.com/content2/ch10>) можно посмотреть учебный видеofilm о шине I²C.

ИСТОРИЯ СОЗДАНИЯ ПРОТОКОЛА I²C

Работу коммуникационного протокола намного легче понять, если знать, как он эволюционировал с течением времени. Протокол I²C был разработан компанией Philips Semiconductors в начале 80-х годов прошлого столетия для обеспечения сравнительно низкоскоростной связи между разными микросхемами. К 90-м годам прошлого столетия протокол был стандартизирован, и другие компании начали использовать его, выпуская совместимые с ним микросхемы. Этот протокол часто называют двухпроводным протоколом, поскольку для обмена данными в нем предусмотрено всего лишь два провода: для данных и для сигнала тактирования. Хотя не все устройства законно используют для связи двухпроводный протокол (поскольку их производители не уплатили за это), обычно их все равно называют I²C-устройствами. Можно провести аналогию с копировальными аппаратами, которые называются ксероксами, хотя они не выпускаются компаний Xerox, которая была первой, создавшей копировальный аппарат электрографического типа с порошковым красящим элементом под этим торговым названием. Поэтому, если для устройства указывается, что оно использует двухпроводный протокол связи, то можно быть в большой степени уверенным, что этот протокол работает, как описано в этой главе.

Вам, возможно, также встретятся устройства, применяющие другой двухпроводный протокол, называющийся SMBus (System Management Bus —

Шина SPI и библиотеки сторонних разработчиков

11

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328.
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328).
- Беспаяечная макетная плата половинного или полного размера.
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (4 шт.).
- Трехцветный светодиод с общим анодом диаметром 5 мм (1 шт.).
- Пьезоэлектрический зуммер (1 шт.).
- Микросхема LIS3DH на адаптерной плате компании Adafruit (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch11>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Мы уже познакомились с двумя основными методами цифровой передачи данных, поддерживаемыми платформой Arduino: шиной последовательного интерфейса УАПП (RS-232) и шиной I²C. В этой главе мы рассмотрим третий метод цифровой связи, поддерживаемый Arduino: шиной SPI¹.

В отличие от шины I²C, шина SPI имеет несколько линий для обмена данными, а также использует отдельную линию для выбора каждого ведомого устройства. Хотя это увеличивает число проводов, в результате также устраняется необходимость в уникальном адресе для каждого ведомого устройства. Обычно с шиной SPI легче работать, чем с шиной I²C; она также

¹ Англ. *Serial Peripheral Interface* — последовательный синхронный периферийный интерфейс. — Прим. пер.

обеспечивает более высокую скорость передачи данных. В этой главе мы узнаем, как использовать встроенные аппаратные возможности SPI платы Arduino для взаимодействия с цифровым акселерометром. В частности, мы рассмотрим, как находить и устанавливать библиотеки сторонних разработчиков для облегчения сопряжения со сложным оборудованием. Для иллюстрации полезности этого подхода мы с помощью акселерометра будем управлять яркостью светодиода и выдавать звуковые сигналы, создав своего рода инструмент, реагирующий на движение звуковыми и визуальными эффектами.

Примечание

В первом издании этой книги для демонстрации принципов работы шины SPI использовалась микросхема цифрового потенциометра MCP4231, поддерживающая возможности SPI. В последнее время эти микросхемы трудно найти и, кроме того, проекты на их основе не особенно стимулируют умственные способности по сравнению с применением акселерометра. Но если вы хотите узнать, как реализовать сопряжение цифрового потенциометра с платой Arduino, это можно сделать, просмотрев учебный фильм на эту тему по ссылке blum.fyi/spi-digipot-tutorial.

ОБЩИЕ СВЕДЕНИЯ О ШИНЕ SPI

Шина SPI была разработана компанией Motorola для реализации полнодуплексной последовательной связи между ведущим устройством и одним или несколькими ведомыми устройствами. Поскольку для протокола SPI нет формального стандарта, часто можно встретить SPI-устройства, работающие немного по-разному. Например, количество передаваемых битов может быть другим, или линия выбора ведомого устройства может не использоваться. В этой главе мы сделаем акцент на сопряжении платы Arduino с устройствами, реализующими наиболее распространенные SPI-интерфейсы. Эти устройства поддерживаются средой разработки Arduino IDE и библиотеками сторонних разработчиков, которые мы будем использовать.

Внимание!

Следует иметь в виду, что, как уже упоминалось в тексте, в различных устройствах протокол SPI может быть реализован по-разному. Поэтому чрезвычайно важно внимательно ознакомиться с характеристиками устройства, которое планируется использовать, по его справочному листку.

Шина SPI может работать в четырех основных режимах, выбор конкретного из которых зависит от требований конкретного устройства. Устройства SPI работают в синхронном режиме, т. е. передача данных синхронизируется сигналом тактирования, подаваемым на линию SCLK. Данные могут вводиться в ведомое устройство или по положительному, или по отрица-

Взаимодействие с жидкокристалли- ческими дисплеями

12

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328 (1 шт.).
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328) (1 шт.).
- Беспаячная макетная плата половинного или полного размера (1 шт.).
- Набор проволочных перемычек.
- Кнопки (2 шт.).
- Резистор номиналом 220 Ом (1 шт.).
- Резистор номиналом 1 кОм (1 шт.).
- Резистор номиналом 4,7 кОм (2 шт.).
- Резистор номиналом 10 кОм (2 шт.).
- Подстроечный потенциометр номиналом 10 кОм (может входить в состав комплекта жидкокристаллического дисплея) (1 шт.).
- Батарея 9 В (1 шт.).
- Разъем для батареи напряжением 9 В (1 шт.).
- Микросхема L7805CV стабилизатора напряжения 5 В (1 шт.).
- Электролитический конденсатор, 10 мкФ, 50 В (2 шт.).
- Диод 1N4001 (1 шт.).
- Биполярный NPN-транзистор PN2222 (1 шт.).
- Динамик с сопротивлением катушки 8 Ом (1 шт.).
- Датчик температуры ТС74А0-5.0VAT (1 шт.).
- Миниатюрный вентилятор с бесщеточным электродвигателем с напряжением питания 5 В (1 шт.).
- Жидкокристаллический дисплей размером 2 строки по 16 символов со штыревыми разъемами (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch12>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Одно из важных преимуществ разработки автономных систем — их возможность работать независимо от основного компьютера. До настоящего времени, если мы хотели выводить из платы Arduino любой тип информации, более сложный, чем мигающий светодиод, нам требовалось подключение к внешнему компьютеру. Оснастив нашу плату Arduino жидкокристаллическим дисплеем (ЖКД), мы сможем легко отображать сложную информацию (значения показаний датчиков, данные синхронизации, параметры настроек, индикаторы хода исполнения и т.п.) непосредственно на самом устройстве Arduino, без необходимости связи с программой монитора порта, исполняющейся на основном компьютере.

В этой главе мы рассмотрим, как подключить ЖКД к плате Arduino и как использовать библиотеку LiquidCrystal для отображения текстовых и произвольных пользовательских символов на ЖКД. После того как вы освоите основные понятия в области жидкокристаллических дисплеев, мы применим некоторые компоненты из предыдущих глав, чтобы создать простой термостат, способный получать данные о температуре в помещении, сообщать эти данные нам и управлять вентилятором, чтобы регулировать температуру. Информация о значениях температуры будет отображаться на подключенном к плате Arduino жидкокристаллическом дисплее, а издаваемый динамиком звуковой сигнал известит нас о повышенной температуре, реагируя на которую Arduino включит вентилятор, чтобы попытаться снизить ее.

Примечание

На веб-странице для этой главы (<https://www.exploringarduino.com/content2/ch12>) можно просмотреть обучающий видеоклип о том, как подключить жидкокристаллический дисплей к плате Arduino.

ПОДКЛЮЧЕНИЕ ЖКД К ПЛАТЕ ARDUINO

Для проектов в этой главе нам потребуется ЖКД с параллельным интерфейсом. Такие дисплеи широкодоступны любых типов и размеров. Наиболее распространенный ЖКД поддерживает отображение двух строк текста по 16 символов каждая (16×2). Для подключения дисплея и взаимодействия с ним он снабжен разъемом из 16 штыревых контактов (14, если дисплей не оснащен фоновой подсветкой). В этой главе мы будем использовать один из таких дисплеев 16×2 с 16-контактным штыревым разъемом.

Если при покупке штыревой разъем вашего дисплея не был впаян в плату, то вам нужно будет припаять его самостоятельно, чтобы дисплей можно было установить на макетную плату. Дисплей с впаянным штыревым разъемом должен выглядеть, как показано на рис. 12.1.

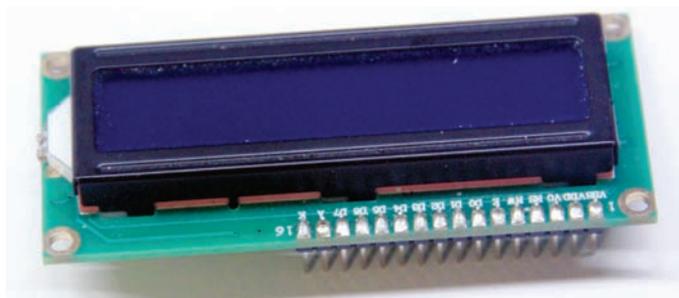


Рис. 12.1. Жидкокристаллический дисплей 16×2 с впаивным штыревым разъемом

Установив ЖКД на макетную плату, нам нужно подключить его к питанию и плате Arduino. Все ЖКД с параллельным интерфейсом имеют одинаковую цоколевку и поддерживают два режима интерфейса: 4-контактный или 8-контактный. Если все необходимые задачи можно выполнить, обеспечив связь между дисплеем и платой Arduino только по четырем линиям данных, то подключаем только соответствующие четыре контакта. Кроме контактов линий связи нужно также подключить контакты для разрешения дисплея, установки режима команд или отображения, а также выбора режима записи или чтения. В табл. 12.1 приведено краткое описание всех контактов дисплея.

Таблица 12.1. Назначение контактов дисплея 16×2 с параллельным интерфейсом

Номер контакта	Название контакта	Назначение контакта
1	VSS	Земля (общий)
2	VDD	Питание +5 В
3	V0	Настройка контраста (на потенциометр)
4	RS	Выбор регистра (режим отображения или команд)
5	RW	Запись/чтение
6	EN	Разрешение
7	D0	Линия данных 0 (не используется в 4-линейном режиме)
8	D1	Линия данных 1 (не используется в 4-линейном режиме)
9	D2	Линия данных 2 (не используется в 4-линейном режиме)
10	D3	Линия данных 3 (не используется в 4-линейном режиме)
11	D4	Линия данных 4
12	D5	Линия данных 5
13	D6	Линия данных 6
14	D7	Линия данных 7
15	A	Анод светодиода фоновой подсветки
16	K	Катод светодиода фоновой подсветки

IV Прерывания, комбинирование функций и карты SD

Глава 13. Прерывания и другие специальные функции

Глава 14. Работа с картами памяти SD

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328 (1 шт.).
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328) (1 шт.).
- Беспаяная макетная плата половинного или полного размера (1 шт.).
- Набор проволочных перемычек.
- Кнопка (1 шт.).
- Резистор номиналом 100 Ом (1 шт.).
- Резистор номиналом 220 Ом (3 шт.).
- Резистор номиналом 10 кОм (1 шт.).
- Трехцветный (RGB) светодиод с общим анодом диаметром 5 мм (1 шт.).
- Электролитный конденсатор, 10 мкФ, 50 В (1 шт.).
- Пьезоэлектрический зуммер.
- Микросхема 74АНСТ14 — шесть инвертирующих буферов на триггере Шмитта (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch13>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Все разрабатываемые нами до сих пор программы для Arduino были синхронного типа.

Это обстоятельство вызывает несколько проблем, одна из которых состоит в том, что пока выполняется функция `delay()`, не могут выполняться никакие другие операции. При этом плата Arduino способна решать несколько задач одновременно, поэтому при простое процессора теряются ценные процессорные циклы.

В этой главе мы рассмотрим, как использовать аппаратные и программные прерывания, чтобы придать скетчам Arduino возможность асинхронного исполнения. Прерывания обеспечивают асинхронное исполнение кода, иницилируя реакцию программы на определенные события, например, истечение определенного периода времени, изменение состояния входного сигнала и т.п. Как можно судить по их названию, прерывания позволяют прервать исполнение любой исполняющейся в настоящее время задачи, выполнить какую-либо другую задачу, а затем возвратиться к продолжению исполнения прерванной задачи. Мы рассмотрим, как вызывать прерывания по запланированным событиям или по изменению состояния сигнала на входных контактах. На основе полученных знаний мы создадим систему неблокирующих аппаратных прерываний, а также разработаем программу генератора звуковых сигналов по прерываниям таймера.

АППАРАТНЫЕ ПРЕРЫВАНИЯ

Аппаратные прерывания активируются изменением состояния контакта ввода-вывода. Прерывания этого типа могут быть особенно полезными в тех случаях, когда нужно изменить значение переменной состояния, не прибегая к постоянному опросу состояния выхода кнопки. В некоторых предыдущих главах при каждой итерации главного цикла `loop()` мы выполняли программную защиту от дребезга и проверяли состояние кнопки. Такой подход работает удовлетворительно в тех случаях, когда исполнение кода в цикле `loop()` не занимает много времени.

Но предположим, что нам нужно исполнить в цикле `loop()` довольно длительную процедуру. Например, нам нужно медленно постепенно повышать яркость светодиода или скорость вращения электродвигателя, используя для этого цикл `for()` и несколько операторов задержки `delay()`. Если управлять цветом или скоростью повышения яркости посредством кнопки, то мы пропустим нажатия кнопки, происходящие в течение исполнения функции `delay()`. Обычно человеческая реакция достаточно инерционна, что позволяет выполнить большое количество функций в цикле `loop()` программы Arduino, включая опрос кнопки при каждой его итерации, не пропуская нажатий кнопки. Но когда цикл `loop()` содержит медленно исполняющиеся компоненты, существует вероятность пропустить события внешнего ввода.

Для решения подобных проблем и предназначены прерывания. Определенные контакты платы Arduino могут вызывать внешние аппаратные прерывания. Аппаратура микроконтроллера знает состояние этих контактов и может асинхронно предоставлять их значение коду прикладной программы. Таким образом, при обнаружении внешнего аппаратного прерывания исполнение главной программы можно приостановить, чтобы выполнить

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328 (1 шт.).
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328) (1 шт.).
- Набор проволочных перемычек.
- ИК-дальномер GP2Y0A21YK0FIR компании Sharp в комплекте с JST-кабелем (1 шт.).
- Шилд регистрации данных со штыревыми разъемами для Arduino производства Adafruit (1 шт.).
- Таблеточная батарейка CR1220, 3 В, 12 мм (1 шт.).
- Карта памяти SD/MicroSD (рекомендуется карта формата SDHC емкостью 8 ГБ) (1 шт.).
- Сетевой адаптер, 5 В, 1 А, выходной USB-разъем (1 шт.).
- Компьютер, оснащенный считывателем SD-карт (или считыватель SD-карт с USB-интерфейсом) (1 шт.).
- Малярный скотч и/или полоски 3M Command для крепления картин.

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch14>.
- Исходный код для проектов этой главы можно также загрузить на вкладке Downloads веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Многочисленные устройства на основе Arduino используются для регистрации самых разнообразных данных: состояние погоды, атмосферные условия, проход посетителей в здание, электрическая нагрузка на предприятиях и в учреждениях и др. Благодаря своим небольшим размерам,

минимальному энергопотреблению и легкости сопряжения с громадным количеством датчиков устройства на основе Arduino представляют собой очевидный выбор для создания регистраторов данных, записывающих и хранящих информацию в течение определенного периода времени. Регистраторы данных часто применяются в разных условиях для сбора сведений о пользователях и окружающей среде, сохраняя их в энергонезависимой памяти какого-либо типа, например, на SD-карте.

В этой главе мы рассмотрим все, что может понадобиться о взаимодействии платформы Arduino с SD-картами. В частности, выясним, как записывать данные на SD-карту и считывать их с нее. Кроме того, мы рассмотрим, как использовать часы реального времени для добавления точных меток времени к собираемым данным. Также мы узнаем, как выводить на экран считываемые данные.

Примечание

Видеолип о регистрации данных с помощью Arduino, а также видеоурок на более сложную тему регистрации данных о местоположении с помощью GPS-приемника можно посмотреть на веб-странице для этой главы <https://www.exploringarduino.com/content2/ch14>.

ПОДГОТОВКА К РЕГИСТРАЦИИ ДАННЫХ

Системы регистрации данных не представляют собой ничего сложного. Они обычно содержат устройство сбора данных какого-либо типа (например, аналоговые датчики) для получения исходной информации. А для хранения больших объемов данных, полученных в течение длительного периода времени, предусмотрено какое-либо устройство накопления информации.

В этой главе особое внимание уделяется нескольким распространенным способам использования SD-карты памяти совместно с платой Arduino для хранения полезных данных. Для регистрации данных существует много разных применений. Далее приводится небольшой список подходящих проектов.

- Метеостанция для отслеживания условий освещенности, температуры и влажности в течение определенного периода времени.
- GPS-трекер и регистратор для слежения за местоположением.
- Система мониторинга температуры компонентов персонального компьютера.
- Система сбора данных об использовании освещения дома или в офисе.

Далее в этой главе мы создадим систему регистрации данных с ИК-датчиком расстояния для сбора данных о посещаемости комнаты.



Беспроводная СВЯЗЬ

Глава 15. Радиосвязь

Глава 16. Беспроводная связь Bluetooth

Глава 17. Wi-Fi и облачные хранилища

Список деталей и оборудования для проектов этой главы

- Плата Arduino Uno или Adafruit METRO 328 (1 шт.).
- USB-кабель (тип А на В для Arduino Uno, тип А на Micro-B для METRO 328) (1 шт.).
- Беспаячная макетная плата половинного или полного размера (1 шт.).
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (1 шт.).
- Пьезоэлектрический зуммер (1 шт.).
- Сетевой адаптер, 5 В, 1 А, выходной USB-разъем (1 шт.).
- Радиоприемный модуль с рабочей частотой 315 МГц (или подобный радиомодуль с рабочей частотой, разрешенной в вашей стране) (1 шт.).
- Однокнопочный пульт дистанционного управления с радиопередатчиком на частоте 315 МГц (или подобный пульт управления с рабочей частотой, разрешенной в вашей стране) (1 шт.).
- Модуль управляемого силового реле (реле IoT Power Relay компании Digital Loggers, Inc.) (1 шт.).
- Лампочка с сетевым питанием (1 шт.).
- Небольшая отвертка (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch15>.
- Исходный код для проектов этой главы можно также загрузить на вкладке Downloads веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Для многих микроконтроллерных проектов часто требуется наличие возможности радиосвязи. Оснастить устройство беспроводной связью можно многими способами, но проще всего использовать радиомодуль.

На рынке предлагается большой выбор радиомодулей разных форм и размеров, но почти все они работают одинаково: когда на передатчике изменяется состояние одного выхода, на выходе приемника тоже появляется соответствующий сигнал. По сути, это можно сравнить с однопроводной шиной, по которой за раз передается и принимается один бит информации. В этой главе мы познакомимся с основными принципами беспроводной связи, ограничившись простыми способами радиосвязи. В последующих двух главах мы расширим рассмотрение этой темы в область радиосвязи Bluetooth и Wi-Fi.

Примечание

Первое издание этой книги (выпущенное в 2013 г. издательством Wiley) содержало главу по радиосвязи с использованием радиомодулей XBee, которые работают, как беспроводной последовательный канал. Но радиомодули XBee трудно настроить для правильной работы, они дорого стоят и могут взаимодействовать только друг с другом. В этом издании я решил заменить эту главу двумя новыми главами по Bluetooth и Wi-Fi, поскольку большинство пользователей находят их более полезными и доступными, чем радиомодули XBee. Также я добавил настоящую главу по простым радиоканалам, поскольку она закладывает хорошую основу для понимания электромагнитного спектра и его значение для радиосвязи. Если же вы хотите научиться работать с радиомодулями XBee, я настоятельно рекомендую сначала ознакомиться с материалом этой главы, а затем просмотреть видеоруководство по работе с ними, которое можно найти по адресу blum.fyi/xbee-tutorial.

ЭЛЕКТРОМАГНИТНЫЙ СПЕКТР

Прежде чем приступать к рассмотрению работы радиосвязи, полезно получить базовые знания научных принципов, на которых она основана. Электромагнитный спектр, наверное, является одним из самых важных аспектов нашей жизни, о котором мы никогда не задумываемся. Данный спектр определяет всю излучаемую во вселенной энергию от гамма-лучей ультравысокой частоты до низкочастотных радиоволн. Посередине спектра находится видимый свет, наиболее знакомый нам тип электромагнитного излучения. Свет от разных источников — солнца, осветительных ламп, светодиодов наших устройств на Arduino — воздействует на наши глаза в виде электромагнитного излучения. Подобно тому, как глаз воспринимает электромагнитную энергию, которую затем наш мозг интерпретирует, как свет и извлекает из нее разную информацию, так и оснащенная «глазами» в виде радиоприемника плата Arduino может принимать электромагнитные излучения в виде радиоволн, а затем извлекать из них информацию. На рис. 15.1 приведено наглядное представление (созданное агентством НАСА) всех форм электромагнитных излучений.

Все электромагнитные излучения передаются в виде волн определенной частоты, которые распространяются в вакууме со скоростью света

Список деталей и оборудования для проектов этой главы

- Плата Feather 32u4 Bluefruit LE (с впаянными разъемами) (1 шт.).
- USB-кабель (тип А на Micro-B) (1 шт.).
- Беспаячная макетная плата половинного или полного размера (1 шт.).
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (1 шт.).
- Подстроечный потенциометр номиналом 10 кОм (или другой аналоговый датчик по вашему выбору) (1 шт.).
- Светодиод красный диаметром 5 мм (1 шт.).
- Сетевой адаптер, 5 В, 1 А, выходной USB-разъем (1 шт.).
- Модуль управляемого силового реле (реле IoT Power Relay от компании Digital Loggers, Inc.) (1 шт.).
- Лампочка с сетевым питанием (1 шт.).
- Небольшая отвертка (1 шт.).
- Смартфон с возможностями BTLE (iPhone или Android) (1 шт.).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch16>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Скорее всего, вам очень быстро надоело передавать через простой радиоканал своей плате Arduino по одному биту информации за раз. Как мы знаем из опыта работы с ноутбуком, смартфоном или устройствами IoT, при наличии радиосвязи другого типа возможности могут быть намного большими. В этой главе мы рассмотрим радиосвязь Bluetooth, которая является наиболее распространенным стандартом беспроводной связи в мире. Разные варианты Bluetooth используются в гарнитурах, клавиатурах, мышах, компьютерах, смартфонах, радиомаяках и т.п.

ЧТО ТАКОЕ BLUETOOTH?

В последнее время термин Bluetooth начали широко применять (часто неправильно) для описания любой беспроводной связи малого радиуса действия между двумя узлами. В действительности разная реализация Bluetooth для различных типов устройств предусматривалась с самого начала. Это обстоятельство часто создает путаницу, когда говорят, что какое-либо устройство поддерживает Bluetooth. Поэтому, прежде чем рассматривать разработку программного обеспечения для Bluetooth, полезно ознакомиться с разными версиями и профилями Bluetooth, а также с соответствующей терминологией.

Стандарты и версии Bluetooth

Исходный стандарт Bluetooth (IEEE 802.15.1) был создан институтом IEEE (Institute of Electrical and Electronics Engineers — Институт инженеров по электротехнике и радиоэлектронике США); в настоящее время эта технология находится в ведении специальной группы SIG Bluetooth.

Примечание

Стандарт Bluetooth — только один из стандартов, разработанных и поддерживаемых институтом IEEE. В ведении этого института находятся также другие технологии, с которыми вам, возможно, приходилось встречаться: Wi-Fi (IEEE 802.11); PoE (Power over Ethernet — питание через Ethernet, IEEE 802.3); стандарт POSIX, являющийся основой для обеспечения совместимости между разными версиями операционной системы UNIX (IEEE 1003); FireWire (IEEE 1394) и многие другие.

Спецификация Bluetooth прошла через несколько этапов развития, когда шаг за шагом добавлялись новые возможности и улучшения. Первая версия, выпущенная в 1998 г., была в основном направлена на обеспечение беспроводной альтернативы проводным каналам последовательной связи (таким же каналам последовательной связи, как и те, которые мы использовали в предыдущих главах для передачи сообщений с платы Arduino на компьютер.) В вышедшей в 2004 г. версии Bluetooth 2.0 была добавлена возможность более высокой скорости передачи данных, что проложило дорогу Bluetooth в качестве цифрового канала для передачи аудио (возможно, наиболее знакомое вам применение Bluetooth). Если вам когда-либо приходилось испытывать трудности, пытаясь выполнить сопряжение гарнитуры Bluetooth со своим смартфоном, то, возможно, вам будет легче, если узнаете, что трудности испытывали не только вы. Исходная спецификация не содержала явной поддержки потокового аудио, поэтому интегрирование этой возможности в спецификации Bluetooth, а также в радиоустройства и процессоры, сохраняя при этом обратную совместимость, было для группы Bluetooth SIG

Wi-Fi и облачные хранилища

17

Список деталей и оборудования для проектов этой главы

- Плата Adafruit Feather M0 Wi-Fi w/ATWINC1500 (с впаянной PCB-антенной) (1 шт.).
- USB-кабель (тип А на Micro-B) (1 шт.).
- Беспаячная макетная плата половинного или полного размера (1 шт.).
- Набор проволочных перемычек.
- Резистор номиналом 220 Ом (4 шт.).
- Резистор номиналом 4,7 кОм (2 шт.).
- Трехцветный (RGB) светодиод с общим анодом диаметром 5 мм (1 шт.).
- Пьезоэлектрический зуммер (1 шт.).
- Сетевой адаптер, 5 В, 1 А, выходной USB-разъем (необязательно) (1 шт.).
- Четырехразрядный 7-сегментный дисплей с интерфейсом I²C (30×14 мм, любого цвета) (1 шт.).
- Реквизиты доступа к сети Wi-Fi (и, факультативно, права администратора на доступ к маршрутизатору).

Исходный код и прочие электронные ресурсы

- Исходный код, видеоуроки и прочие электронные ресурсы для этой главы можно загрузить с веб-страницы <https://www.exploringarduino.com/content2/ch17>.
- Исходный код для проектов этой главы можно также загрузить на вкладке **Downloads** веб-страницы издательства Wiley для этой книги: <https://www.wiley.com/go/exploringarduino2e>.

ЧТО ВЫ УЗНАЕТЕ ИЗ ЭТОЙ ГЛАВЫ

Вот мы и достигли конечной темы этой книги. Подключение платы Arduino к Интернету можно сравнить, пожалуй, только с запуском ее в космос, чтобы сделать весь мир своей игровой площадкой.

В общем и целом возможность связи по Интернету — чрезвычайно сложная тема. О том, как лучше всего выполнить сопряжение платформы Arduino с Интернетом вещей (IoT¹), можно с легкостью написать целые тома.

¹ Англ. *Internet of Things*. — Прим. пер.

Но поскольку невозможно рассмотреть все существующие способы сопряжения платы Arduino с Всемирной сетью, в этой главе мы ограничимся взаимодействием платы Arduino (или любого другого устройства IoT) через интерфейс локальной сети. Также мы узнаем, как с помощью платы Arduino, оснащенной возможностью радиосвязи Wi-Fi, раздавать веб-страницы и работать с облачными данными. В частности, мы рассмотрим составные уровни сетевой топологии, процесс раздачи веб-страниц, а также выясним, как взаимодействовать с интерфейсом прикладного программирования (API) сторонних разработчиков.

РАБОТА ARDUINO В СЕТИ

Изложить все подробности работы Интернета в одной главе затруднительно, поэтому для ознакомительных целей взаимодействие Arduino с Интернетом можно, по сути, рассматривать, как показано на рис. 17.1.

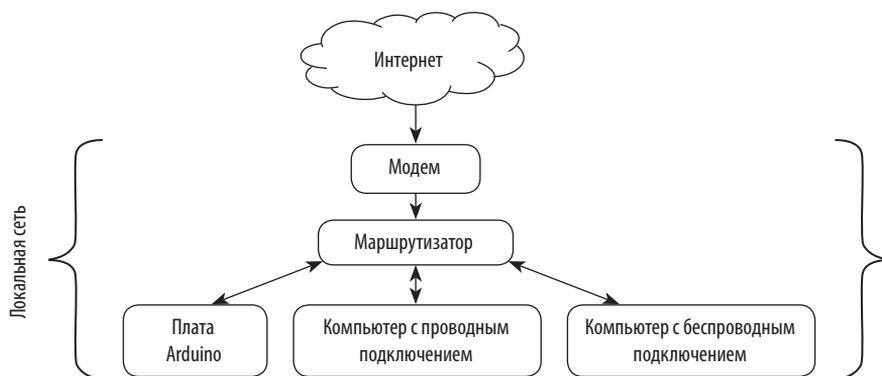


Рис. 17.1. Упрощенное представление Сети и локальной сети

Сначала мы рассмотрим работу только в пределах нашей локальной сети. При работе в локальной сети взаимодействие с платой Arduino через браузерный интерфейс возможно только в том случае, когда как Arduino, так и браузер подключены к одному и тому же маршрутизатору (по проводному или беспроводному каналу). А затем мы исследуем, как можно с помощью маршрутизатора получить доступ к плате Arduino из любой точки мира (по крайней мере, оттуда, где есть Интернет).

Сетевая терминология

Прежде чем приступать к рассмотрению работы с Arduino в Сети, ознакомимся с соответствующей терминологией. Далее приведен краткий список слов, понятий и сокращений, которые нужно усвоить, чтобы работать с материалом этой главы.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

А

Адрес

IP

 глобальный 468

 локальный 468

 шлюза 495

MAC 468

Акселерометр

 микросхема LIS3DH 277

 трехосевой ADXL335 93

Амплитуда волны 170

Амплитудная манипуляция 403

Аналого-цифровой преобразователь,

АЦП 86

Антенна несимметричная

четвертьволновая 404

Архитектура микроконтроллера

 8-разрядная 37

 32-разрядная 37

 ARM 38

 AVR 37

Б

Библиотека

 Adafruit BTLE SPI 431

 Adafruit GFX 516

 Adafruit LED Backpack 516

 Adafruit LIS3DH Library 286

 Adafruit nRF5 430

 Adafruit Unified Sensor Library 286

 Adafruit Universal Sensor 289

 JSON 502

 LiquidCrystal 300, 303, 305, 306

 Mouse.h 226

 RTClib 374, 381

 SD 360, 366

 Servo.h 144, 146

 Stepper.h 157, 158

 TimerOne 343, 344

 WiFi101 by Arduino 473

 Wire 256, 257, 259

Бит 84

Булевы переменные 75

В

Веб-сервер 471

Версия Bluetooth

 2.0 424, 425

 3.0 425

 4.0 425

Волатильная переменная 342

Д

Дальномер инфракрасный Sharp 92

Данные

 регистратор 352

 тип данных 61

 unsigned long 163

Датчик температуры

 AD7414 251

 TC74A0-5.0VAT 250

 TMP36 93

Делитель напряжения резистивный 97

Диапазон радиочастот ISM 401

Диод заграждающий 109

Директива

 RoHS 99

 препроцессора 375

Дисплей семисегментный 513, 514
 Длина волны электромагнитного излучения 400
 Дребезг контактов 72, 335

З

Заголовок ответа 481
 Загрузчик 40, 41
 Закон Ома 57, 58
 Запись в файл на SD-карте 364
 Запрос GET 485

И

Излучение электромагнитное
 длина волны 400
 частота 400
 Индикатор хода выполнения 308
 Институт IEEE 424, 469
 Интернет-подключение 425
 Интерфейс
 Web-API 498
 Facebook 499
 Git Hub 498
 Google Maps 498
 NASA 499
 Phillips Hue 499
 назначение 501

К

Карты
 MicroSD 353
 SD 353
 Клиент 470
 Ключевое слово
 (byte) 308, 311
 LED_BUILTIN 51
 Код ответа
 404 481
 HTTP 481
 Кодер поворотный 332
 Команда
 analogWrite() 64
 AT 433
 df 358

digitalRead() 71
 digitalWrite() 52
 GET 469
 ipconfig 496
 mkdosfs 359
 ping 479
 pinMode() 50, 51
 POST 469
 sudo 359
 unmount 358
 Wire.beginTransaction() 261
 Wire.endTransmission() 261
 Wire.requestFrom() 261
 Wire.write() 261

Комментарии

многострочные 50
 однострочные 50

Компания

Atmel 36, 37
 Digital Loggers, Inc. 416
 JeeLabs 374
 Maxim Integrated 372
 Microchip 36, 37
 Nordic Semiconductor 426
 NXP Semiconductors 372
 PJRC 344

Конденсатор развязывающий 127

Короткое замыкание 116

Корпус, тип

DIP 529
 SMD 529

Коэффициент заполнения сигнала 65

Л

Линия

MISO 273
 MOSI 273
 SCLK 273
 SS 275

Логический оператор ИЛИ 76

М

Макетная плата беспаячная 55
 Маршрутизатор
 администрирование 495

Микроконтроллер
 ATmega
 16U2 190
 32U4 439
 328P 525
 Cortex M0 471
 архитектура
 8-разрядная 37
 32-разрядная 37
 AVR 37, 38

Микросхема
 Н-моста L293D 117
 nRF5 430
 WiFi WINC1500 473, 475
 акселерометра LIS3DH 277
 гироскопа L3GD20H 297
 датчика температуры
 AD7414 251
 TC74A0-5.0VAT 250
 преобразователя
 USB/RS-232 CP2104 187
 сдвигового регистра
 74HC595 232
 стабилизатора напряжения
 L7805CV 127
 триггера Шмитта
 74АНСТ14 340
 часов реального времени
 DS1307 372
 PCF8523 372

Миллиампер 58

Многозадачность 332

Модификатор F() 375, 432

Модуляция
 цифровых сигналов 402
 широтно-импульсная 54, 64

Мощность
 номинальная 59
 уравнение 59

Н

Напряжение
 опорное 86
 электросети 414

О

Облако 467

Объект
 DateTime 382
 Serial 198

Объявление
 переменной 61

Оператор
 != 75
 <= 62
 #ifdef 381
 if() 71
 if...else 71
 include 146
 Keyboard.begin() 220
 return 76
 инверсии 76
 присваивания 75

П

Панель администрирования
 маршрутизатора 495

Переменная
 булева 75
 волатильная 340
 объявление 61

Период сигнала 67

Плата
 Adafruit METRO 328 534
 Arduino
 Due 37
 Feather 45, 427, 471
 Mega ADR 192
 Mega 2560 41
 MKR1000 51
 Seeeduno Lite 216
 Photon 45
 Uno 41
 макетная беспаячная 55

Порт
 RS-232 186
 перенаправление 497
 последовательный компьютера 48
 переадресация 493

- Преобразование NAT 468
 - Препроцессор
 - директива 375
 - определение FORCE_UPDATE 381
 - Прерывание Arduino
 - запуск 334
 - процедура обработки 333
 - Приложение
 - Adafruit Bluefruit LE Connect 440, 442, 452, 453
 - Провод
 - заземления 415
 - нулевой 415
 - фазный 415
 - Программа
 - alternate.ino 236
 - arduino_read_pot.ino 208
 - bargraph.ino 243
 - blink.ino 62
 - BTLE_led.ino 447, 457
 - BTLE_sensor.ino 435
 - car.ino 132
 - chronograph.ino 164
 - debounce.ino 74
 - entrance_logger.ino 387
 - fun_with_sound.ino 347
 - hbridge.ino 123
 - hw_multitask.ino 341
 - lamp_remote.ino 416
 - LCD_progress_bar.ino 308
 - LCD_text.ino 305
 - LCD_thermostat.ino 321
 - led.ino 60
 - lightrider.ino 240
 - list_control.ino 204
 - lock_computer.ino 222
 - motor.ino 113
 - motor_pot.ino 115
 - mouse.ino 224
 - music.ino 179
 - nightlight.ino 101
 - piano.ino 183
 - pot.ino 89, 193
 - read_temp.ino 259
 - rf_test.ino 407
 - rgb_nightlight.ino 78
 - sd_read_write.ino 369
 - sd_read_write_rtc.ino 375
 - stepper.ino 157
 - sweep.ino 149
 - timer1.ino 344
 - web_weather_display.ino 518
 - web_weather.ino 509
 - write_to_sd.ino 365
 - загрузчика Arduino 38
 - последовательной связи PuTTY 197
 - Программатор AVRISP mkII 39
 - Проект OpenWeatherMap 499
 - Протокол
 - Android Open Accessory 192
 - DHCP 470
 - FTP 426
 - HTTP 469
 - HTTPS 426
 - I2C 248
 - SMBus 248
 - SMTP 426
 - безопасности
 - WPA 477
 - WPA2 477
 - Профиль
 - Bluetooth
 - A2DP 426
 - GATT 426
 - HID 454
 - Прошивка для WINC1500 475
- Р**
- Радиомодули XBee 398
 - Радиоспектр 400
 - Реализация Bluetooth
 - Classic 425
 - Low Energy 425
 - BTLE 425
 - Регистр сдвиговый 230
 - Регистратор данных 352
 - Режим
 - транзистора насыщения 109
 - шины SPI 273

Резистор повышающий встроенный 160, 161

Реле 413

 IoT Power Relay 416
 принцип работы 416

С

Светодиод

 падение напряжения 58
 рабочий ток 58
 трехцветный 77

Сервер DNS 470

Сервопривод 140
 управление 141

Сигнал

 коэффициент заполнения 65
 период 67
 скважность 65
 тактовый
 полярность 273
 частота 343

 цифровой
 модуляция 402

Система

 DNS 468
 глобального позиционирования GPS 278

Служба

 HID Keyboard 454, 455
 DNS 426

Сопряжение Bluetooth 454

Состояние «плавающее» 68

Спектр электромагнитный 398

Специальный символ

 новой строки 194
 табуляции 194

Спецификатор const 61

Стабилизатор напряжения 127

Статор 107

Т

Таймер Timer1 343

Типы данных 61

Ток утечки 70

Триггер Шмитта 338

У

Управление сервоприводом 141

Уравнение мощности 59

Утилита JSON Pretty Printer 504

Ф

Файл

 CSV 353
 log.csv 365
 pitches.h 174
 заголовочный pitch.h 173
 типа CSV 217

Формат

 FAT16 354
 FAT32 354
 JSON 501

Фоторезистор 98

Функция

 analogRead() 88
 analogReference() 94
 attachInterrupt() 340
 boolean debounce (boolean last) 75
 cmd.toLowerCase() 443
 constraint() 100, 101
 createChar() 307
 debounce() 76, 317
 delay() 52, 329
 digitalPinToInterrupt() 333, 340
 digitalRead() 71, 184
 digitalWrite() 73
 forward() 123
 indexOf() 444
 isConnected() 432
 Keyboard.press() 223
 Keyboard.print() 221
 Keyboard.println() 221
 Keyboard.releaseAll() 223
 loop() 60
 map() 100, 101
 millis() 163

Mouse.isPressed() 227
 Mouse.press() 227
 Mouse.release() 227
 myStepper.step() 159
 noTone() 173, 183
 parseInt() 200
 Processing
 background() 210
 draw() 210
 fill() 268
 float() 211
 mousePressed() 214
 serialEvent() 211
 readDistance() 150
 readJoystick() 226
 reverse() 123
 RTC.adjust() 374
 RTC.now() 382
 Serial.available() 198
 Serial.begin() 90, 193
 Serial.parseInt() 203
 Serial.println() 193
 Serial.println() 89, 90, 193, 194
 Serial.read() 198
 setCursor() 304
 setTimeout() 205
 setup() 60
 shiftOut() 235, 239
 swap() 340
 tone() 170, 173, 181
 updateDateTime() 382
 void loop() 51
 void setup() 50
 waitForOK() 434, 435
 блокирующая 159
 пользовательская 73

Ц

Цепь
 резистивно-емкостная 336
 электрическая замкнутая 57
 Цикл for 62

Ч

Частота
 волны 170
 электромагнитного излучения 400
 тактового сигнала 343

Ш

Шилд
 для SD-карт 360
 Adafruit data logging shield 361
 Seed Studio SD Card Shield V4 361
 SparkFun Micro SD shield 361
 характеристики 360, 361

Шина

I2C 248
 SPI 271
 режимы 273
 Широтно-импульсная модуляция, ШИМ
 54, 64, 181

Э

Электродвигатель щеточный 107
 Электромагнитный спектр 398

Я

Язык
 HTML 469
 программирования Processing 206

Начните экспериментировать уже сегодня!

Arduino. Базовый набор 2.0

Если вы хотите не только изучить основы использования микроконтроллерной платформы Arduino для разработки электронных проектов, но и получить первичные практические навыки, то этот комплект для вас. В него входит популярная книга Джереми Блума «Изучаем Arduino: инструменты и методы технического волшебства», плата Arduino UNO с кабелем для подключения к ПК, макетная плата, электронные компоненты и брошюра с упражнениями.



«Изучаем Arduino». Книга + учебный набор для экспериментов

Набор разработан по материалам книги Джереми Блума «Изучаем Arduino: инструменты и методы технического волшебства». Вы научитесь работать с цифровыми контактами ввода-вывода, использовать широтно-импульсную модуляцию, считывать данные с аналоговых датчиков, работать со звуком, управлять двигателями постоянного тока и серводвигателями, применять сдвиговые регистры. Подключать светодиоды, кнопки, потенциометры, датчики, динамики и др.



Набор «Умный дом» для экспериментов с контроллером Arduino

Набор «Умный дом» содержит всё необходимое для разработки полезных проектов по автоматизации вашего жилища с использованием платы Arduino: Arduino UNO, макетную плату, электронные компоненты и краткое руководство.

В комплект входит популярная книга Джереми Блума «Изучаем Arduino: инструменты и методы технического волшебства», которая поможет вам изучить основы использования популярной микроконтроллерной платформы Arduino.



дерзай!



www.bhv.ru

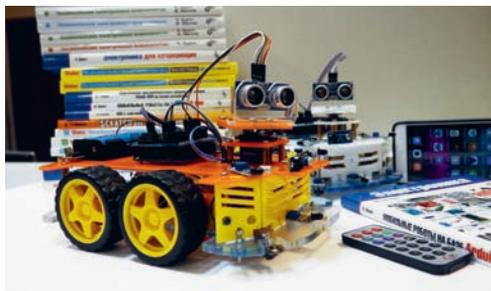
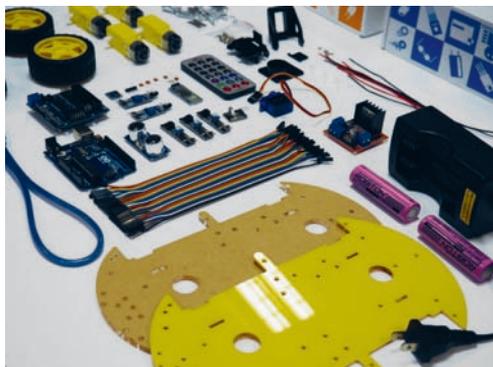
Мобильные роботы на базе Arduino + КНИГА

Отдел оптовых поставок:

E-mail: opt@bhv.ru

www.bhv.ru/books/robot

ISBN 978-5-9775-3774-2



Если вы хотите научиться конструировать мобильного робота, но у вас пока нет знаний, опыта, деталей и электронных компонентов, то этот набор разработан специально для вас. В его состав входят: плата Arduino, необходимые электронные компоненты, двигатели, 4-колесное шасси, а также популярная книга М. Момота «Мобильные роботы на базе Arduino».

Вы сможете:

- сконструировать базовую модель мобильного робота;
- дистанционно управлять роботом с помощью IR-пульта или смартфона Android;

- изучить базовые алгоритмы управления роботом;
- модернизировать и обучить базового робота движению по линии, выходу из лабиринта, игре в кегельринг.



Мобильные роботы на базе ESP32 в среде Arduino IDE

Отдел оптовых поставок:

E-mail: opt@bhv.ru



«Эта книга создана как руководство для начинающих Конструкторов, людей, которым нравится конструировать, энтузиастов Arduino, желающих освоить новые высокоскоростные контроллеры ESP32. Описано конструирование мобильных роботов на платформе ESP32, позволяющей реализовывать как простейшие, так и достаточно интеллектуальные устройства. А применение шаговых моторов позволяет создавать очень интересные высокоточные устройства, например робота с памятью траектории. Все проекты выполнены на единой двухколесной базе из простых в изготовлении деталей. В каждый новый проект вносятся небольшие схемные изменения и оригинальное программное наполнение, что совершенно преобразует поведение роботов. Программирование ведется в традиционной среде Arduino IDE.

Я уверен, что проекты не оставят вас равнодушными, особенно создание балансирующего робота. При внимательном и творческом подходе вы получите дополнительные знания в области физики и программирования и сможете конструировать своих оригинальных роботов».

Михаил Момот, автор книги

Вы узнаете как:

- ♦ Создавать роботов на шаговых двигателях
- ♦ Управлять роботом со смартфона через Bluetooth и WiFi
- ♦ Взаимодействовать с роботом удаленно через Интернет
- ♦ Передавать изображение от робота на смартфон
- ♦ Подключить к роботу смартфон в качестве IP-камеры для видеонаблюдения
- ♦ Сконструировать балансирующего робота

Момот Михаил Викторович, сотрудник Томского политехнического университета, основатель и директор фирмы «Юрга-Технологии-Инновации», занимающейся разработкой роботов. Увлекается робототехникой, поклонник и пропагандист проекта Arduino с 2014 года. Основатель неформального клуба робототехников «Лига роботов ЮТИ ТПУ», объединяющего школьников студентов, преподавателей и энтузиастов. Автор книги «Мобильные роботы на базе Arduino».

Отдел оптовых поставок:

E-mail: opt@bhv.ru

Расширяя знания, расширяй возможности!



Книга позволяет приобрести знания в области электроники, необходимые для преобразования инновационных идей в функционирующие устройства.

В практическом руководстве излагаются основы электричества и электроники, которые сопровождаются подробными, доходчивыми инструкциями, принципиальными схемами и иллюстрациями. Вы научитесь выбирать необходимые компоненты, безопасно собирать схемы, находить и устранять причины неполадок, а также собирать модульные прототипы. В четвертом издании книги добавлены полностью новые главы по датчикам, микроконтроллерам, модульной электронике и самым последним версиям программных инструментов.

В книге рассмотрены

- ♦ Резисторы, конденсаторы, индукторы и трансформаторы
- ♦ Диоды, транзисторы и интегральные схемы
- ♦ Оптоэлектронные компоненты, солнечные панели и фототранзисторы
- ♦ Датчики, модули GPS и сенсорные экраны
- ♦ Операционные усилители, фильтры, генераторы колебаний, таймеры
- ♦ Стабилизаторы напряжения, источники питания
- ♦ Цифровая электроника, жидкокристаллические и светодиодные дисплеи и логические элементы
- ♦ Микроконтроллеры и платформы для разработки прототипов устройств, включая Arduino
- ♦ Электродвигатели постоянного тока, сервоприводы и шаговые двигатели
- ♦ Микрофоны, аудиоусилители и громкоговорители
- ♦ Модульная электроника и разработка прототипов

Саймон Монк (Simon Monk) имеет степень бакалавра по кибернетике и вычислительной технике, а также докторскую степень по разработке программного обеспечения. Несколько лет занимался преподавательской и научной деятельностью. Затем стал одним из основателем компании Momote Ltd., где разрабатывал программное обеспечение для мобильных устройств. В настоящее время является профессиональным автором книг по электронике («Мейкерство. Arduino и Raspberry Pi. Управление движением, светом и звуком» и др.).

Пауль Шерц (Paul Scherz), инженер-механик, физик, изобретатель в области электронных устройств. Многолетнее увлечение электроникой позволяет ему успешно изучать взаимодействия элементарных частиц и физику плазмы на факультете ядерной техники и технической физики Вис-консинского университета.