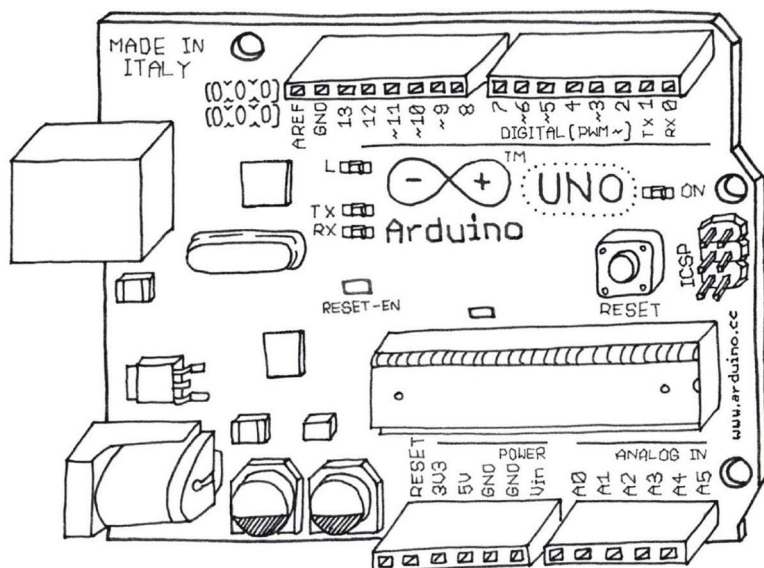


Make:

Первые шаги с ARDUINO

4-е издание



Платформа прототипирования
с ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

Массимо Банци
Майкл Шайло



УДК 004.89
ББК 32.973.26-018
Б23

Банци, М.

Б23 Первые шаги с Arduino. — 4 изд.: Пер. с англ. /
М. Банци, М. Шайло. — СПб.: БХВ-Петербург, 2023. —
288 с.: ил.

ISBN 978-5-9775-1708-9

Книга представляет собой введение в мир Arduino — самой популярной платформы для прототипирования электроники с открытым исходным кодом. В ней есть как простые проекты, так и вся необходимая информация для создания собственных конструкций, от списков компонентов до финишной отладки.

Рассмотрены основные принципы интерактивного дизайна и физических вычислений, современные платы Arduino и программные среды, основы электроники, прототипирование на макетной плате, создание принципиальных схем, облачные платформы. Подробно описан ряд практических проектов, в том числе автоматическая система полива растений и проекты с выходом в Интернет через Wi-Fi. Материал изложен с учетом последней версии Arduino IDE и новых плат на базе ARM.

*Для читателей, интересующихся электроникой
и конструированием*

УДК 004.89
ББК 32.973.26-018

© 2022 BHV

Authorized Russian translation of the English edition of *Getting Started with Arduino, 4th Edition* ISBN 9781680456936 © 2022 Massimo Banzi and Michael Shiloh.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Авторизованный перевод с английского языка на русский издания *Getting Started with Arduino, 4th Edition* ISBN 9781680456936 © 2022 Massimo Banzi and Michael Shiloh. Перевод опубликован и продается с разрешения компании-правообладателя O'Reilly Media, Inc.

ISBN 978-1-680-45693-6 (англ.)
ISBN 978-5-9775-1708-9 (рус.)

© Massimo Banzi, Michael Shiloh, 2022
© Перевод на русский язык, оформление.
ООО "БХВ-Петербург", ООО "БХВ", 2023

Оглавление

Об авторах	9
Предисловие к 4-му изданию	11
Предисловие	12
Благодарности	15
Обозначения, принятые в книге	16
Использование примеров программного кода	17
1/Введение	19
Для кого эта книга	20
Что такое интерактивный дизайн?	21
Что такое физические вычисления?	22
2/Путь Arduino	23
Создание прототипов	23
Arduino — для любителей мастерить	24
Мы любим свалки электроники!	25
Вам пригодятся игрушки	26
Совместное творчество	26
3/Платформа Arduino	27
Аппаратная часть Arduino	27
Интегрированная среда разработки (IDE)	30
Установка Arduino IDE на ваш компьютер	31
Установка IDE: MacOS	31
Настройка драйверов: MacOS	31
Выбор порта: MacOS	32
Установка IDE: Windows	33
Настройка драйверов: Windows	34
Выбор порта: Windows	34
Установка IDE: Linux	35
Настройка драйверов: Linux	36
Предоставление разрешений на последовательные порты:	
Linux	36
Выбор порта: Linux	36

4/Начало работы с Arduino	39
Анатомия интерактивного устройства	39
Датчики и приводы	40
Мигающий светодиод	40
Передайте мне пармезан	45
Arduino не для слабаков	46
Настоящие мастера пишут комментарии	46
Код шаг за шагом	47
Чем мы займемся дальше?	51
Что такое электричество?	52
Использование кнопки для управления светодиодом	55
Как это работает?	59
Одна схема, тысяча вариантов поведения	60
5/Расширенные функции ввода и вывода.....	67
Работа с другими датчиками включения/выключения	67
Самодельные датчики-переключатели	69
Управление светом с помощью ШИМ	70
Использование фоторезистора вместо кнопки	79
Аналоговый вход	80
Испробуйте другие аналоговые датчики	84
Связь через последовательный порт	85
Управление мощной нагрузкой	87
Сложные датчики	90
Алфавит Arduino	90
6/Processing и лампа Arduino	93
Планирование	94
Разработка кода	96
Сборка схемы	103
Окончательная сборка лампы	106
7/Облачный сервис Arduino Cloud	109
Облачная среда разработки Arduino	109
Хранилище проектов Project Hub	111
Облако Интернета вещей IoT Cloud	112
Возможности Arduino IoT Cloud	114
Стоимость использования Arduino Cloud	115

8/Автоматическая система полива сада.....	117
Планирование.....	119
Тестирование часов реального времени.....	122
Тестирование реле.....	128
Принципиальные электрические схемы.....	131
Проверка датчика температуры и влажности.....	142
Программирование.....	146
Установка времени включения и выключения.....	146
Проверка, не пора ли включить или выключить клапан.....	153
Проверка наличия дождя.....	158
Пишем окончательный код.....	160
Сборка схемы.....	170
Макетная плата Proto Shield.....	174
Размещение компонентов на Proto Shield.....	175
Пайка вашего устройства на Proto Shield.....	181
Тестирование собранного устройства.....	195
Размещение вашего устройства в корпусе.....	197
Тестирование готовой системы автоматического полива сада.....	200
Что можно попробовать сделать самостоятельно.....	201
Список покупок для системы полива.....	201
9/Семейство Arduino ARM.....	205
В чем разница между AVR и ARM?.....	205
Зачем нужны 32-разрядные микроконтроллеры?.....	206
В чем разница между микроконтроллером и микропроцессором?.....	207
Что лучше: AVR или ARM?.....	208
Знакомство с платами на базе Arduino ARM.....	208
Особые функциональные возможности.....	210
Рабочее напряжение.....	210
Максимальный ток через вывод.....	211
Цифроаналоговый преобразователь.....	212
USB-хост.....	212
Размер и форма плат на базе ARM.....	213
10/Интернет-подключение ARM и проект Fistbump.....	215
Интернет-проект Fistbump.....	215
Знакомство с MQTT: протокол передачи телеметрии.....	216
Аппаратная часть проекта Fistbump.....	218
Подключение к брокеру MQTT на Shiftr.io.....	222
Подключение проекта Fistbump к Интернету: код Arduino.....	222
Подключение проекта Fistbump к Интернету: веб-страница.....	227

11/Устранение неполадок	235
Понимание	236
Упрощение и разделение	236
Исключение и определенность	236
Тестирование платы Arduino	237
Тестирование схемы на макетной плате	239
Выявление проблем	241
Проблемы с установкой драйверов в Windows	242
Проблемы с IDE в Windows	242
Поиск COM-порта Arduino в Windows	243
Другие методы отладки	245
Как получить помощь онлайн	246
 ПРИЛОЖЕНИЯ	
А/Макетные платы	251
Б/Расшифровка обозначений резисторов и конденсаторов	255
В/Краткий справочник по программированию для Arduino	259
Структура программы	259
Специальные символы	259
Константы	260
Переменные	261
Область видимости переменной	263
Управляющие структуры	264
Арифметика и формулы	267
Операторы сравнения	267
Логические операторы	268
Составные операторы	268
Инкремент (++) и декремент (--)	269
Функции ввода и вывода	269
Функции времени	271
Математические функции	272
Функции случайных чисел	274
Последовательный обмен данными	274
Семейство плат Arduino	276
Клоны, совместимые платы и подделки	278
 Г/Условные обозначения на электрических схемах	281
Предметный указатель	285

1/ Введение

Arduino — это вычислительная платформа с открытым исходным кодом для создания *интерактивных объектов*, которые могут работать автономно или подключаться к Интернету. Изначально мы разрабатывали Arduino для творческих людей — художников, дизайнеров, — желающих использовать микрокомпьютеры в своих арт-проектах, не владея инженерными навыками в области электроники. Позже в эту платформу буквально влюбились миллионы остальных пользователей, которые хотят воплощать свои идеи в реальность при помощи цифровых технологий.

Аппаратное и программное обеспечение Arduino полностью открыты. Философия открытого исходного кода способствует росту сообщества, которое щедро делится своими знаниями. Такая концепция отлично подходит для начинающих, потому что они всегда могут получить помощь по любому вопросу в многочисленных онлайн-сообществах или в ближайшем магазине. Учебные примеры, как правило, содержат инструкции по созданию собственного проекта или служат отправной точкой для дальнейшего развития темы.

Программное обеспечение Arduino, известное как *интегрированная среда разработки* (Integrated Development Environment, IDE), является бесплатным. Вы можете загрузить его с сайта www.arduino.cc. Среда Arduino IDE основана на языке Processing (<http://www.processing.org>), который специально был разработан так, чтобы художники могли создавать объекты компьютерного искусства, не становясь профессиональными программистами. Arduino IDE может работать на Windows, Mac и Linux.

Плата Arduino Uno недорогая и вполне терпима к типичным ошибкам новичков. Если вы все-таки повредите микроконтрол-

лер — основной компонент Arduino Uno, — его можно приобрести отдельно для замены. В продаже также имеются базовые комплекты электронных компонентов для самостоятельного конструирования.

Проект Arduino был разработан студентами и преподавателями и стал очень популярным учебным пособием. Философия открытого исходного кода, благодаря которой возникло сообщество Arduino, распространяется на методы обучения, учебные планы и другую информацию.

Поскольку аппаратное и программное обеспечение Arduino распространяется по открытой лицензии, вы можете скачать конструкторскую документацию для любой платы Arduino, чтобы использовать ее в качестве компонента более сложной конструкции, создать свою собственную разработку или просто для того, чтобы разобраться в устройстве и работе Arduino. Вы можете сделать то же самое с программным обеспечением.

Платформа Arduino проста в использовании, и эта книга написана для того, чтобы помочь новичкам, не имеющим опыта работы с Arduino.

Для кого эта книга

Эта книга написана для новичков, людей без специального технического образования, которые хотят научиться творить с помощью электроники и программирования, поэтому она объясняет многие вещи предельно простым языком, способным вызвать острое недовольство у некоторых инженеров. Кстати, один из таких инженеров назвал вступительные главы первого издания "мишурой". Именно в этом суть! Посмотрим правде в глаза: быть профессионалом в чем-то и уметь объяснять — это две разные вещи. Если простые и понятные вещи, открывающие перед миллионами людей путь к творчеству, — это мишура, значит, нужно углубиться в ее изучение.

Эта книга не предназначена для обучения электронике или программированию, но, читая данную книгу, вы, безусловно, многому научитесь.

ОТ АВТОРА

После того как платформа Arduino стала популярной, я увидел, что экспериментаторы и любители всех мастей начали использовать ее для создания потрясающе красивых и оригинальных объектов. Я понял, что все они по-своему художники и дизайнеры, поэтому книга адресована и вам тоже.

— Массимо



В основу платформы Arduino легла дипломная работа, которую Эрнандо Барраган сделал на платформе Wiring во время учебы у Кейси Риаса и Массимо Банци на курсах человеко-машинного интерфейса Interaction Design Institute в Иврее.

Что такое интерактивный дизайн?

Платформа Arduino была создана для обучения *интерактивному дизайну*, особой дисциплине, которая во главу угла ставит *макетирование*. Существует множество определений интерактивного дизайна, но мы предпочитаем следующее:

Интерактивный дизайн — это творческий процесс, основанный на интерактивном опыте.

В современном мире интерактивный дизайн занимается формированием опыта взаимодействия между людьми и объектами. Это хороший способ исследовать создание красивых и, возможно, даже противоречивых взаимодействий между нами и технологиями. Интерактивный дизайн поощряет творчество как итеративный процесс, основанный на макетах или прототипах с постоянно растущей сложностью. Этот подход также встречается в некоторых методах классического дизайна и может быть расширен за счет создания прототипов с использованием передовых технологий; сюда относится и макетирование электронных устройств.

Конкретная область интерактивного дизайна, связанная с Arduino, часто называется *физическими вычислениями*¹ (или дизайном физического взаимодействия).

¹ Физическими вычислениями в области информационных технологий называют вычисления, непосредственно связанные с реальными физическими объектами (например, вычисления для расчета механических движений манипуляторов робота). — *Прим. пер.*

Что такое физические вычисления?

Физические вычисления основаны на использовании электроники для создания прототипов инновационных устройств. Сюда относится разработка интерактивных объектов, которые могут взаимодействовать с людьми с помощью датчиков и исполнительных механизмов, реагирующих на поведение человека и реализованных в виде программного обеспечения, работающего внутри микроконтроллера (миниатюрного компьютера на одном чипе).

В прошлом применение электроники при создании физического объекта означало необходимость постоянно иметь дело с инженерами и многократно создавать и переделывать электронные схемы. Эти проблемы мешали творческим людям напрямую играть с окружающей средой. Большинство инструментов разработки также предназначались для квалифицированных инженеров и требовали обширных знаний.

В последние годы микроконтроллеры стали намного дешевле и проще в использовании. В то же время персональные компьютеры стали быстрее и мощнее, что позволяет создавать более совершенные (и простые) средства разработки.

Революционный прорыв Arduino заключается в том, что эта платформа сделала инструменты разработки доступными новичку, позволяя людям начать создавать собственные проекты спустя всего день или два после прочтения книги. С Arduino новичок может очень быстро познакомиться с основами электроники и принципом работы датчиков и начать создавать прототипы с минимальными затратами.

2/Путь Arduino

Путь Arduino — это философия, основанная на действиях, а не на разговорах. Это постоянный поиск более быстрых и удобных способов создания лучших прототипов. Мы изучили множество технологий прототипирования и разработали новые способы прикладного мышления.

Классическая инженерия представляет собой строго оговоренный процесс перехода от пункта *A* к пункту *B*. Путь Arduino дает вам восхитительную возможность заблудиться по дороге и вместо этого найти *C*.

Путь Arduino открыт для любителей мастерить — вы можете смело экспериментировать с платформой и получать неожиданные результаты. Занимаясь поиском лучших способов создания прототипов, мы выбрали несколько инструментов программирования, которые позволяют без труда экспериментировать с программами и оборудованием.

В этой главе мы расскажем о идеях, событиях и людях, без которых был бы невозможен путь Arduino.

Создание прототипов

В основе платформы Arduino лежит конструирование *прототипов* — упрощенных макетов устройств, которые взаимодействуют с объектами, людьми и компьютерными сетями. Мы стремимся найти наиболее простой, быстрый и дешевый способ создания работающих прототипов.

Многие новички, которые впервые сталкиваются с электроникой, думают, что им необходимо научиться создавать все с нуля. Это

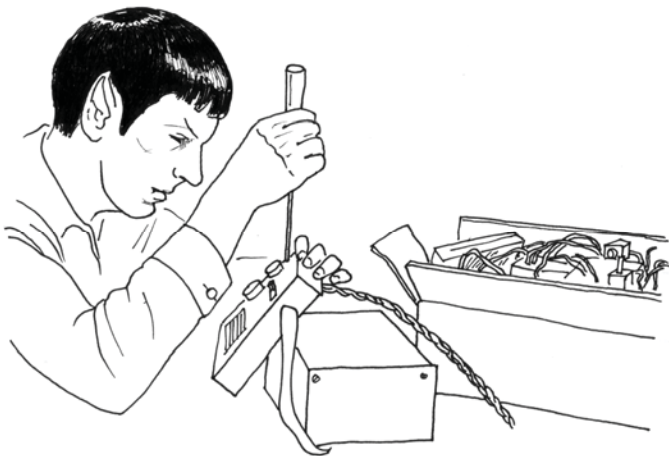
пустая трата времени и сил. На самом деле вам нужно как можно скорее убедиться, что ваше устройство работает, чтобы у вас появилось желание сделать следующий шаг или — так иногда бывает! — чтобы кто-то захотел дать вам много денег на продолжение разработок.

Вот почему мы разработали платформу для упрощения макетирования: зачем тратить время и энергию на создание конструкции с нуля — процесс, который требует времени и глубоких технических знаний, если можно взять готовые компоненты, переделать и скомбинировать их, чтобы использовать себе во благо огромную работу, проделанную большими корпорациями и квалифицированными инженерами.

Arduino — для любителей мастерить

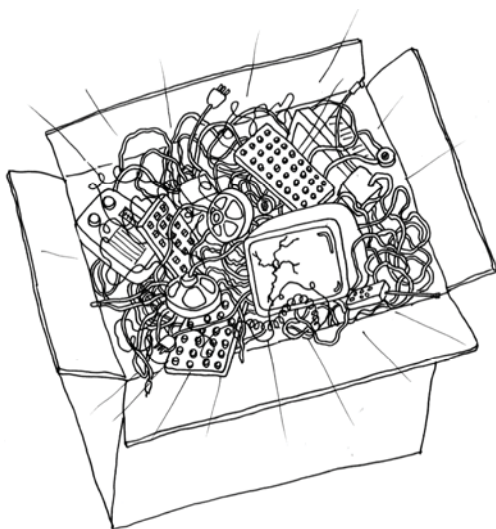
Мы считаем, что очень важно творчески экспериментировать с технологиями, переделывая разные устройства и программы, иногда без четко определенной цели.

Различные готовые устройства идеально подходят для любителей технического творчества. Один из лучших способов быстро добиться отличных результатов — приобрести дешевые игрушки или старое электронное оборудование, а затем переделать их.



Мы любим свалки электроники!

В наши дни люди выбрасывают много техники: старые принтеры, компьютеры, разные офисные машины, производственное оборудование и даже военные приборы. Свалки и распродажи старого электронного оборудования всегда пользовались большой популярностью у небогатых и начинающих любителей технического творчества. Особенно хорошо это было заметно в местечке Ивреа, где мы разработали платформу Arduino. В городе раньше располагалась штаб-квартира компании Olivetti, одного из крупнейших производителей офисной техники, выпускавшая компьютеры с 1960-х годов. В середине 1990-х старые компьютеры были выброшены на окрестные свалки. Там оказались горы компьютерных блоков, электронных компонентов и всевозможных странных устройств. Мы провели очень много времени на распродажах, покупая всевозможные устройства и детали буквально за гроши и приспособлявая их к нашим прототипам. Когда вы можете за бесценок приобрести тысячу динамиков, в конце концов вы обязательно придумаете, куда их применить. Прежде чем начать что-нибудь мастерить, хорошенько покопайтесь в своих запасах или посетите распродажи старых вещей.



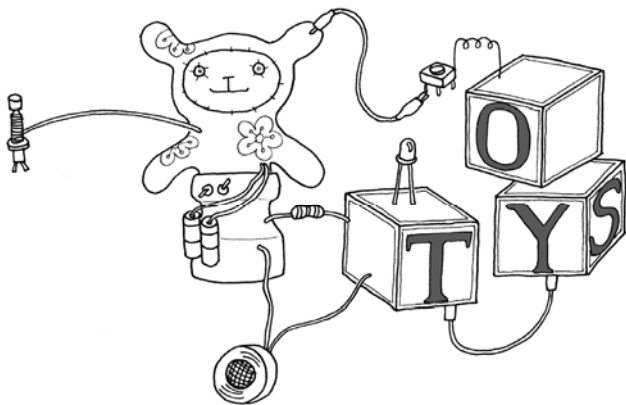
Вам пригодятся игрушки

Игрушки — великолепный источник дешевых компонентов. Их можно разбирать, переделывать и использовать повторно. Благодаря изобилию очень дешевых и разнообразных высокотехнологичных игрушек из Китая вы можете реализовать свои самые смелые идеи.

ОТ АВТОРА

Я стараюсь, чтобы мои студенты поняли, что технологии не страшны в изучении и не сложны в использовании. Один из моих любимых учебников — брошюра "Простые датчики и актуаторы" (<http://lowtech.propositions.org.uk>) Усмана Хака и Адама Сомлаи-Фишера. Я считаю, что у них получилось прекрасное руководство, и постоянно им пользуюсь.

– Массимо



Совместное творчество

Совместное творчество увлеченных людей — один из ключевых принципов в мире Arduino. На форуме www.forum.arduino.cc люди из разных уголков мира помогают друг другу больше узнать о платформе. Мы также создали веб-сайт под названием Project Hub (<https://create.arduino.cc/projecthub>), где одни пользователи описывают свои проекты и делают их доступными для других. Удивительно, как много интересного выкладывают в Интернет члены сообщества Arduino.

3/Платформа Arduino

Платформа Arduino состоит из двух основных компонентов: *платы Arduino*, которая является частью оборудования в создаваемом вами устройстве или объекте, и *интегрированной среды разработки Arduino*, или IDE — это программное обеспечение вы запускаете на своем компьютере. С помощью IDE вы создаете небольшую компьютерную программу (*скетч*), которую загружаете на плату Arduino. Скетч сообщает плате, что она должна делать.

Совсем недавно при создании нового устройства инженерам приходилось разрабатывать электронные схемы с использованием сотен различных компонентов со специфичными названиями, такими как резистор, конденсатор, катушка индуктивности, транзистор и т. д. У каждой схемы было свое конкретное назначение, и для внесения изменений в схему требовалось демонтировать провода, паять соединения и совершать много других сложных действий.

С появлением цифровых технологий и микропроцессоров многие функции, которые когда-то выполняли соединительные провода, заменило программное обеспечение. Компьютерную программу изменить намного легче, чем электронную схему. С помощью нескольких нажатий клавиш компьютера вы можете радикально поменять логику устройства и протестировать две или три версии за время, необходимое для пайки пары резисторов.

Аппаратная часть Arduino

Аппаратная часть платформы Arduino — это небольшая плата, на которой смонтирован целый компьютер в виде небольшой микросхемы (микроконтроллера).

ОТ АВТОРА

Этот компьютер как минимум в тысячу раз менее мощный, чем MacBook, на котором я написал эту книгу, но он намного дешевле и очень полезен для создания интересных устройств.

– Массимо

Посмотрите в центр платы Arduino Uno: вы увидите прямоугольный черный кусок пластика с 28 металлическими выводами (или, возможно, крошечную квадратную пластмассовую пластину, если у вас версия SMD). Это и есть микроконтроллер ATmega328, сердце вашей платы Arduino.



На самом деле существует множество плат Arduino, но наиболее распространена на сегодняшний день плата Arduino Uno, описанная в этой книге. В *главе 9* мы дадим краткий обзор всего семейства Arduino и расскажем, что отличает новое семейство ARM от плат AVR.

Создатели Arduino разместили на этой плате все компоненты, необходимые для правильной работы микроконтроллера и обмена данными с вашим компьютером. Версия платы, на которую мы в основном будем ориентироваться в этой книге, Arduino Uno, — самая простая в использовании и наиболее подходящая для обучения. Почти все, о чем мы будем говорить, применимо ко всем платам Arduino, как к самым новым, так и к более ранним.

На рис. 3.1 вы видите, что по краям платы расположены полоски с отверстиями, а возле них на плате нанесены надписи. Эти полоски представляют собой *разъемы*, которые служат для подключения к плате датчиков и исполнительных механизмов.



Датчик измеряет какой-то параметр в физическом мире и преобразует его в сигнал, понятный компьютеру, в то время как исполнительный механизм преобразует сигнал от компьютера в действие в физическом мире. Далее в этом разделе книги вы узнаете намного больше о сенсорах и исполнительных механизмах.

Поначалу все эти разъемы могут немного сбивать с толку. Сейчас мы объясним назначение входных и выходных контактов, которые вы научитесь использовать, прочитав эту книгу. Не беспокойтесь, если и после объяснений вы останетесь в замешательстве.

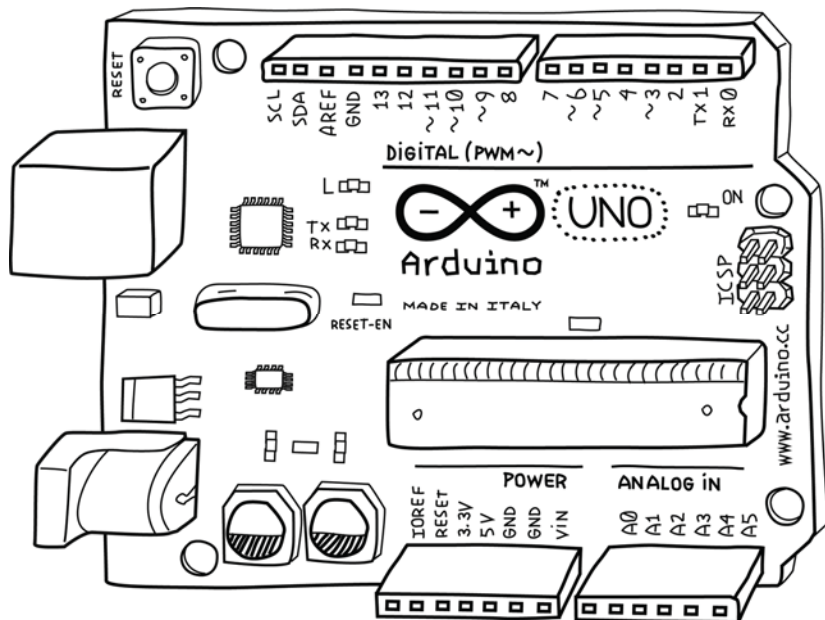


Рис. 3.1. Плата Arduino Uno

ве — в этой книге есть много новых понятий, и вам может потребоваться некоторое время, чтобы привыкнуть к ним. Мы еще не раз повторим объяснения в разном виде, и они станут для вас более понятными, когда вы начнете собирать схемы и получать результаты.

- ◆ 14 контактов цифрового ввода/вывода (вход/выход), пронумерованные от 0 до 13. Эти контакты могут быть как входами, так и выходами. Входы предназначены для считывания информации с датчиков, а выходы служат для управления исполнительными механизмами. Вы будете указывать рабочее направление сигнала (внутри или наружу) в скетче, созданном в среде IDE. Цифровые входы могут считывать только одно из двух значений, а цифровые выходы могут выводить только одно из двух значений: HIGH (высокий логический уровень) и LOW (низкий логический уровень).
- ◆ 6 контактов аналоговых входов (контакты 0–5). Аналоговые входные контакты применяются для считывания значений на-

пряжения с аналоговых датчиков. В отличие от цифровых входов, которые могут различать только два разных уровня (HIGH и LOW), аналоговые входы могут измерять 1024 различных уровня напряжения.

- ◆ 6 контактов аналогового выхода (контакты 3, 5, 6, 9, 10 и 11). На самом деле это шесть цифровых контактов, которые способны выполнять третью функцию: они могут работать как аналоговый выход. Как и в случае с контактами цифрового ввода/вывода, вы указываете в скетче, какую функцию должен выполнять контакт.

Плата может питаться от USB-порта вашего компьютера, большинства зарядных устройств USB или от сетевого источника питания (рекомендуется источник с выходным напряжением 9 В, цилиндрический разъем под центральный плюсовой контакт диаметром 2,1 мм). Всякий раз, когда подключено питание от сетевого источника, Arduino будет задействовать его, а когда сетевой источник отключен, Arduino будет получать питание от USB-разъема. Плата допускает одновременное подключение питания от сетевого источника и через USB-разъем.

Интегрированная среда разработки (IDE)

Arduino IDE — это специальная программа, которая работает на вашем компьютере и позволяет вам писать скетчи для платы Arduino на простом языке программирования, созданном на базе языка Processing (<http://www.processing.org>). Когда вы нажимаете кнопку загрузки скетча на плату, начинается волшебное действие: IDE переводит написанный вами код на язык C (который вообще-то довольно сложен для новичка) и передает его встроенному компилятору avr-gcc — это важный компонент программного обеспечения с открытым исходным кодом, который окончательно переводит ваш скетч на язык, понятный микроконтроллеру. Этот последний шаг очень важен, потому что именно здесь Arduino упрощает вашу жизнь, выполняя за вас наиболее сложную работу по программированию микроконтроллеров.

Последовательность разработки программы на Arduino в основном выглядит так:

1. Подключите плату к USB-порту компьютера.
2. Напишите скетч, который "оживит" плату.
3. Загрузите этот скетч на плату через соединение USB и подождите пару секунд, пока плата перезагрузится.
4. Наблюдайте, как плата выполняет команды скетча, который вы написали.

Установка Arduino IDE на ваш компьютер

Для программирования платы Arduino необходимо сначала установить IDE, загрузив соответствующий файл с веб-сайта Arduino (<https://www.arduino.cc/en/software>). Выберите подходящую версию для вашей операционной системы. Для Windows выберите вариант **Win 7 and newer** (Windows 7 и новее). На следующей странице веб-сайта вы можете поддержать разработчиков Arduino IDE денежным пожертвованием или сразу нажать кнопку с надписью **JUST DOWNLOAD** (Просто скачать). Сохраните файл и выполните инструкции в одном из следующих разделов в соответствии с вашей операционной системой.

Установка IDE: MacOS

Когда загрузка файла завершена, в зависимости от настроек вашего браузера, он может быть открыт автоматически, или вам придется открыть его вручную, обычно двойным щелчком по иконке файла.

Перетащите приложение Arduino в папку Applications (Приложения).

Настройка драйверов: MacOS

Arduino Uno использует драйвер, предоставляемый операционной системой MacOS, поэтому устанавливать его не нужно.

Завершив установку IDE, подключите Arduino Uno к Mac при помощи USB-кабеля.

Зеленый светодиод с надписью PWR на плате должен загореться, а желтый светодиод с надписью L должен начать мигать.



На экране своего компьютера вы можете увидеть всплывающее окно с сообщением об обнаружении нового сетевого интерфейса.

Если это произойдет, нажмите кнопку **Network Preferences** (Настройки сети), а когда это окно откроется, просто нажмите кнопку **Apply** (Применить). Устройство Arduino Uno будет отображаться со статусом **Not Configured** (Не настроено), но это не мешает правильной работе. Закройте окно системных настроек.

Теперь, когда вы установили программное обеспечение, вам нужно выбрать правильный порт для связи с Arduino Uno.

Выбор порта: MacOS

Запустите Arduino IDE либо через папку Applications (Приложения), либо с помощью встроенного средства поиска Spotlight.

В меню **Инструменты** Arduino IDE выберите пункт **Порт**, а затем укажите порт, начинающийся с `/dev/cu.usbmodem` или `/dev/tty.usbmodem`. Возле имени порта также может стоять надпись *Arduino/Genuino Uno*. Оба этих порта относятся к вашей плате Arduino, и не имеет значения, какой из них вы выберете.

На рис. 3.2 показан пример списка портов.

Все почти готово! Вам осталось лишь проверить, что в настройках Arduino IDE правильно указан тип используемой вами платы.

В меню **Инструменты** в Arduino IDE выберите пункт **Плата**, а затем выберите **Arduino Uno**. Если у вас другая плата, необходимо указать этот тип платы (название платы напечатано рядом с символом Arduino).

Поздравляем! Ваше программное обеспечение Arduino установлено, настроено и готово к использованию. Теперь вы можете перейти к *главе 4*.