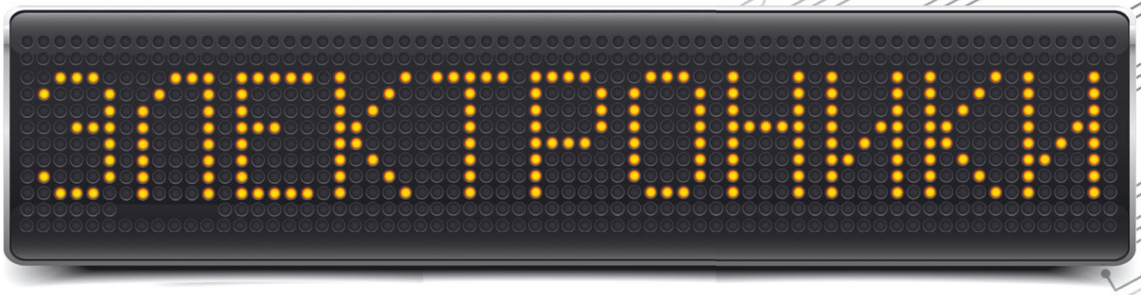
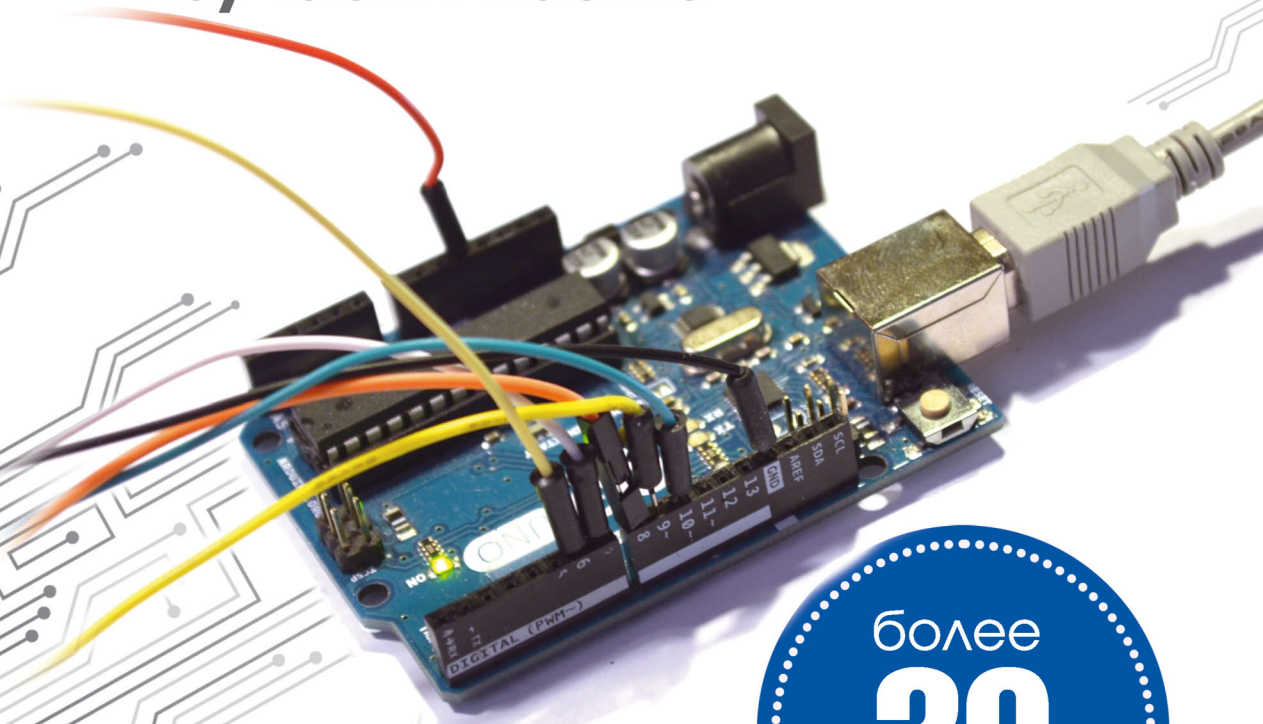
Ю. РЕВИЧ

АЗБУКА



Изучаем Arduino



более 30 СХЕМ

УДК 087.5: 621.38
ББК 32.85
Р32

Все права защищены.
Ни одна часть данного издания не может быть воспроизведена или использована в какой-либо форме, включая электронную, фотокопирование, магнитную запись или какие-либо иные способы хранения и воспроизведения информации, без предварительного письменного разрешения правообладателя.

Ревич, Юрий.
Р32 Азбука электроники. Изучаем Arduino / Ю. Ревич. – Москва: Издательство АСТ: Кладезь, 2017. – 224 с. – (Электроника для всех).

ISBN 978-5-17-102271-6

В книге читатель знакомится с основой современной электроники — микроконтроллерами на примере популярной любительской платформы Arduino. Начиная с простых схем, управляющих свечением светодиода, читатель постепенно переходит к более сложным устройствам: бытовой автоматике, часам и измерительным приборам, элементам конструкции механических устройств. Последовательно изложены принципы программирования в среде Arduino с подробными пояснениями.

Книга представляет собой продолжение книги «Азбука электроники. Электронные устройства своими руками», где можно найти необходимые элементарные физические сведения, а также свойства основных электронных компонентов. Книга предназначена детям среднего и старшего школьного возраста и их родителям, кружкам по электронике, а также всем тем, кто интересуется современной электроникой.

УДК 087.5: 621.38
ББК 32.85

ISBN 978-5-17-102271-6

© Ревич Ю.В., 2017
© ООО «Издательство АСТ», 2017

ОГЛАВЛЕНИЕ

РЕВОЛЮЦИЯ
ПОД НАЗВАНИЕМ ARDUINO 5

Что такое Ардуино? 8

ЧТО ТАКОЕ ARDUINO? 11

Основные платы Arduino 11
Среда программирования Arduino IDE 20
Язык программирования Arduino 29

ГЛАВА 1. ПРОСТЫЕ КОНСТРУКЦИИ НА ARDUINO 41

Эксперимент 1.
Мигаем светодиодом 41
Эксперимент 2. Управляем кнопкой 47
Эксперимент 3. Звуки Arduino 54
Эксперимент 4.
Пешеходный светофор 62

ГЛАВА 2. АНАЛОГОВЫЕ ВВОД
И ВЫВОД 65

Эксперимент 5.
Аналоговый вход 66
Эксперимент 6.
Аналоговый выход: плавное управление нагрузкой 69
Эксперимент 7. Аналоговый пульт управления 80
Эксперимент 8. Термостат 82

ГЛАВА 3. ДИСПЛЕИ 90

Эксперимент 9. Подключение цифрового дисплея к Arduino 92
Эксперимент 10. Простой электронный термометр 104

Эксперимент 11.
Arduino и графический дисплей 114
Эксперимент 12.
Подключение текстового дисплея 133

ГЛАВА 4. ПЕРЕДАЧА ДАННЫХ В ARDUINO 139

Интерфейс TWI (I^2C) 140
Эксперимент 13. Электронные часы 144
Эксперимент 14. Подключение цифровых датчиков 158
Эксперимент 15. Беспроводная передача данных 179
Глава 5. Познавательные проекты 189
Эксперимент 16. Пироэлектрический датчик
и охранная сигнализация 189
Эксперимент 17. Управляем с любого ИК-пульта 193
Эксперимент 18. Эксперименты с сервоприводом 197
Эксперимент 19. Ультразвуковой дальномер 205

ПРИЛОЖЕНИЕ 1. ПРОГРАММИРОВАНИЕ ARDUINO MINI И PRO MINI 209

Программирование Mini
с помощью платы Arduino Uno 209
Программирование Mini
через адаптер USB-UART 211
Программирование Arduino Mini через адаптер USB-UART
с автоматическим сбросом 212

ПРИЛОЖЕНИЕ 2. СЛОВАРЬ ЧАСТО ВСТРЕЧАЮЩИХСЯ АНГЛОЯЗЫЧНЫХ
АББРЕВИАТУР И ТЕРМИНОВ 216

ЧТО ТАКОЕ ARDUINO?

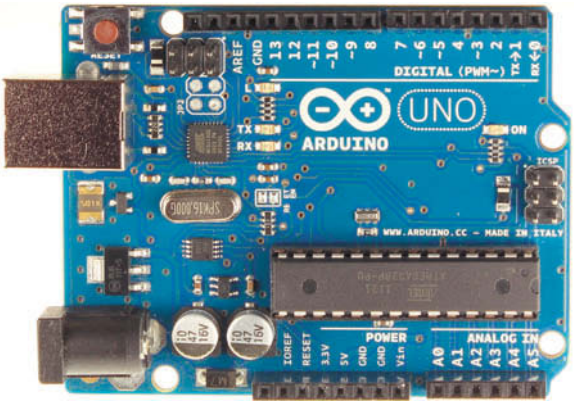
Как вы уже узнали из предисловия, Arduino — семейство плат на основе микроконтроллеров серии AVR фирмы Atmel. Все платы из этого семейства объединены единым принципом устройства и способами программирования, потому мы в дальнейшем будем часто упоминать *платформу* Arduino, имея в виду все семейство в целом.

В этой вводной главе мы прежде всего познакомимся с некоторыми самыми распространенными платами Arduino, на которые мы будем опираться в дальнейшем. Затем мы кратко познакомимся со средой программирования Arduino и некоторыми особенностями языка программирования, лежащего в ее основе.

Из сетевых ресурсов, посвященных Arduino, можно порекомендовать русскоязычный сайт ***arduino.ru***. В существенной своей части он представляет собой перевод англоязычного официального сайта ***arduino.cc***, хотя часто не успевает за обновлениями, потому содержит много устаревшей информации. Более интересен его украинский коллега на русском языке ***arduino.ua***, на котором, кроме официальной информации, можно найти много различных попутных сведений. Новейшие версии плат Arduino и документацию к ним лучше искать на англоязычном ***arduino.org*** — сайте европейской компании, разрабатывающей и выпускающей эти платы.

Основные платы Arduino

В предисловии также упоминалось, что ранее платы Arduino использовали контроллер ATmega168 (а еще ранее — самый первый представитель этой линейки ATmega8). И до сих пор на многих ресурсах в Сети можно встретить ссылки на принципиальные схемы и обозначения выводов плат, где показан именно ATmega168. Это не должно вас смущать, потому что эти контроллеры устроены одинаково и различаются лишь объемом памяти для различных целей, в том числе — для загрузки программы. Так что схемы

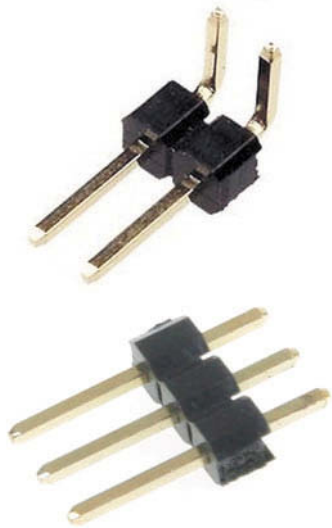


с обозначениями выводов на основе любой из этих разновидностей AVR-контроллеров пригодны для использования.

Сами платы, конечно, совершенствуются со временем и потому могут несколько отличаться. Хотя все новые их разновидности можно без раздумий ставить вместо старых — изменения почти не затрагивают главных особенностей.

Различных плат Arduino существует более полутора десятка разновидностей. Примеры из этой книги ориентированы на три варианта: самую известную плату Arduino Uno, а также две миниатюрных платы, удобных для применения в автономных малогабаритных конструкциях — Arduino Nano и еще меньшую по размерам Arduino Mini (не путать с Arduino Micro!). С точки зрения программирования эти платы устроены одинаково, и программу, отлаженную на наиболее удобной для макетирования плате Arduino Uno, можно без проблем закачивать в Nano и Mini.

Arduino Uno (см. рисунок на предыдущей странице) — основная (базовая) модель Arduino, на нее равняются все остальные. Ее размер (54×69 мм, без учета разъемов, выступающих за пределы платы) и конфигурация контактов стали стандартом, на который равняется множество электронных изделий, включая даже далекие от собственно Arduino. На размеры Arduino Uno обычно ориентируются также производители плат расширения, так называемых «шилдов» (shields). «Шилды», содержащие дополнительные компоненты или гнезда для их подключения, могут устанавливаться на плату Arduino Uno «в этажерку» друг над другом.



Другие платы и внешние датчики подключаются к Arduino Uno с помощью широко распространенных контактных колодок с шагом 2,5 мм (см. рисунок слева). Такие разъемы со штырями носят название PLS, ответные части-гнезда (такие, как установлены на плате Arduino Uno) называются PBS. Платы расширения имеют такие же разъемы, как на плате Arduino Uno, но с противоположной гнездам стороны разъемов у них торчат штыри, что и позволяет устанавливать платы «в этажерку».

Рассмотрим кратко основные компоненты платы Arduino Uno. Самая большая микросхема с 28 выводами (по 14 с каждой стороны) — это сам контроллер ATmega328.

Здесь он установлен на панельку, что позволяет извлекать его и, например, использовать отдельно в других схемах. Таким путем Arduino Uno можно использовать в качестве программатора для этой микросхемы.

Подробности для любознательных: альтернативные способы программирования Arduino

Контроллер ATmega328 в составе плат Arduino программируется без излишних сложностей через последовательный порт, размещенный на плате (см. далее). Однако, практически на всех платах Arduino имеется отдельный программирующий 6-контактный разъем (двухрядный со штырями, на фото Arduino Uno он расположен справа вверху от контроллера). Этот разъем позволяет не использовать среду Arduino, а программировать контроллер в любой удобной системе программирования, включая профессиональные пакеты. На любой плате Arduino обязательно имеются, как минимум, контактные площадки для установки такого разъема, позволяющего программировать контроллер, не извлекая его из схемы. Такая система носит название ISP, что означает In System Programming («программирование в системе»). Через нее же закачивается программа Bootloader, которая всегда присутствует в Arduino — именно она позволяет программировать контроллер через USB без лишних проблем.

Возможно и обратное: платы Arduino можно применять в качестве программатора — для загрузки программ, созданных в среде программирования Arduino IDE, в другие типы контроллеров семейства AVR. Тогда их можно использовать отдельно в схемах, где дополнительные устройства, размещенные на плате, не требуются, а возможности довольно-таки навороченного контроллера ATmega328 излишни. Пример того, как пользоваться этой возможностью, можно посмотреть, например, на сайте ***Amperka.ru*** (см. *http://wiki.amperka.ru/программирование:работа-с-attiny-через-arduino*).

На другом торце платы Arduino Uno размещен, во-первых, основной коммуникационный USB-разъем для подключения к компьютеру. По традиции он самого большого размера из всех многочисленных разновидностей USB (такие разъемы устанавливают сейчас только на крупногабаритной технике, например, на принтерах или сканерах). Потому к нему придется приобретать отдельный кабель, называемый USB-кабелем типа AB. Сторона A – это обычный прямоугольный USB, всем хорошо знакомый по флэш-накопителям, он вставляется в компьютер, а сторона B — квадрат со скошенными гранями, он

и вставляется в разъем на плате. Через USB-разъем производится программирование платы в среде Arduino и одновременно он же служит для посылки сообщений через последовательный порт контроллера при его работе.

Во-вторых, на той же стороне платы размещено гнездо для подключения внешнего источника питания. При подключении к компьютеру через USB-разъем плата будет получать питание 5 вольт через него, таким образом при программировании отдельного внешнего источника обычно не требуется. Если в гнездо питания вставить штекер от сетевого адаптера, то тогда уже питания от USB не требуется, и плата может работать отдельно от компьютера. Напряжение, которое выдает адаптер, может быть нестабилизированным и должно находиться в пределах 7–12 вольт. Теоретически преобразователь питания, встроенный в Arduino, позволяет подавать внешнее питание до 20 вольт, но он не установлен на радиатор и может перегреваться при значительной разнице напряжений между входом и выходом. Потому в описаниях Arduino обычно ограничивают питание значением 12 вольт, хотя при небольшой нагрузке платы можно подавать питание и немного выше этого значения.

Кстати, если внешний источник стабилизированный, то нижнее значение в 7 вольт также соблюдать необязательно — при необходимости Arduino Uno отлично работает от 5-вольтовых адаптеров или от батарейных блоков с напряжением даже ниже 5 вольт. При покупке адаптера надо учитывать, что диаметр штекера разъема питания составляет 5,5 мм, и более тонкие разъемы (например, от мобильных зарядников) сюда не подойдут. При необходимости подключения нестандартных источников или батарейных блоков можно приобрести универсальный разъем 5,5 мм, в котором установлена колодка для непосредственного подключения проводов от источника «под винт» (см. рисунок справа).

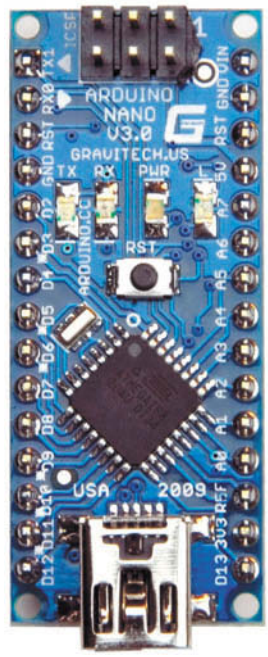
На контакты боковых разъемов, о которых мы говорили ранее, в основном выведены напрямую выводы самого контроллера, иногда их еще называют портами (хотя это и не совсем точно). Часть из этих выводов обозначается просто цифрами от 0 до 13 — это цифровые порты ввода-вывода, которые могут переключаться из состояния работы «на выход» в состояние «на вход» для приема сигналов логических уровней нуля и единицы. На схемах их часто обозначают цифрой

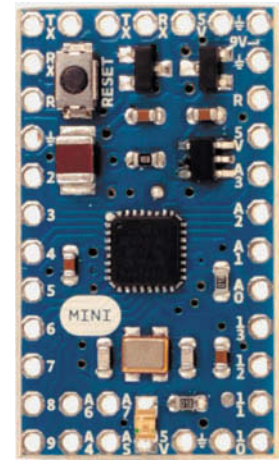
с добавлением буквы D (от слова digital, например, D3 или D12 — в тексте программы это все равно будет просто 3 или 12). Многие из них также задействованы в других функциях контроллера — например, выводы номер 0 и 1 представляют собой выводы RX и TX последовательного (serial) порта UART. Если в программе предусмотрен обмен сигналами между Arduino и компьютером или другими компонентами схемы через такой порт, то в качестве обычных входов-выходов выводы 0 и 1 задействовать нельзя. Последовательный порт используется почти в каждом проекте (хотя бы на стадии загрузки программы), и в качестве цифровых портов 0 и 1 практически не применяют. Потому на платах Mini и Nano (о которых далее) эти два вывода обозначены только, как выводы последовательного порта (RX и TX), а нумерация цифровых выводов начинается с двойки.

К выводу номер 13 традиционно подключен светодиод с токоограничивающим резистором (он размещен на плате недалеко от этого вывода и обозначен буквой L). Его можно использовать для проверки работы различных программ, причем работе вывода при другом его использовании этот светодиод не мешает. Есть и другие важные функции отдельных выводов, о которых мы поговорим при рассмотрении различным схем.

Шесть выводов с буквой A в обозначении могут работать в качестве аналоговых входов — к ним подключен аналогово-цифровой преобразователь (АЦП). Наличие таких входов — одно из самых главных преимуществ Arduino. Именно это качество превращает процесс проектирования, например, цифрового термометра в рутинную операцию, состоящую из написания нескольких строк в программе. Заметим, что аналоговые порты, если они не использованы по назначению, могут также служить в виде обычных цифровых портов ввода-вывода с номерами от 14 до 19.

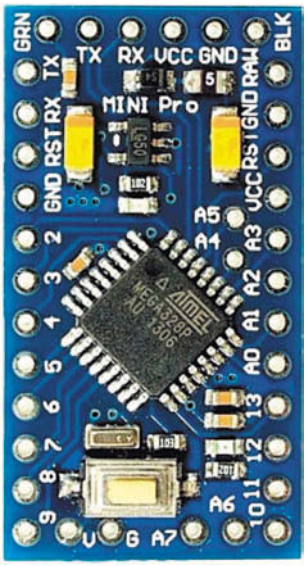
Кроме портов ввода-вывода, на боковые разъемы с другой стороны платы выведены многочисленные выводы, связанные с питанием, назначение которых понятно из надписей рядом. Следует отметить, что напряжение 3,3 вольта получается из отдельного преобразователя, установленного на плате, и часто бывает необходимо для питания внешних датчиков. Напряжение Vin повторяет то, что подается на разъем внешнего питания (можно подавать внешнее питание сюда вместо внешнего разъема). Особую функцию выполняет вывод, обозначенный, как AREF — в необходимых случаях на него подается отдельное опорное напряжение для АЦП, в обычном режиме на него никаких напряжений подавать нельзя. Назначение остальных выводов мы постепенно разберем при рассмотрении конкретных схем.





Платы **Arduino Nano** и **Mini** (см. рисунки вверху) в целом повторяют устройство Arduino Uno. Поэтому программы, отлаженные на Uno, обычно можно без переделок загружать в Nano и Mini. Только расположены эти версии Arduino на миниатюрных платах и используют самые малогабаритные корпуса контроллеров. Потому они не могут устанавливаться «в этажерку» стандартных Arduino-габаритов, а контроллер из них извлечь нельзя.

При этом Arduino Mini, самый миниатюрный из этих вариантов, лишен USB-разъема и может программироваться или через специальный отдельный адаптер USB-UART, или с помощью платы Arduino Uno (см. *Приложение 1*). Есть и другие мелкие отличия — так, у Mini отсутствует вывод AREF для подключения внешнего опорного напряжения АЦП, зато есть стабилизатор питания 3,3 вольта для внешних датчиков, как у Uno. У Nano, который почти идентичен Uno по всем остальным компонентам платы, такой стабилизатор отсутствует: видимый на рисунке вывод 3,3 вольта работает только при питании



от USB. Нет у этих плат и отдельного разъема для подключения внешнего питания — его следует подключать к контакту Vin. Как вариант, можно подавать от внешнего адаптера стабилизированное напряжение 5 вольт или ниже на вывод с таким же обозначением. Но в этом случае нельзя одновременно подключать плату к USB, потому лучше таким способом не пользоваться — Arduino прекрасно работает от 5-вольтового адаптера, включенного в качестве внешнего питания.

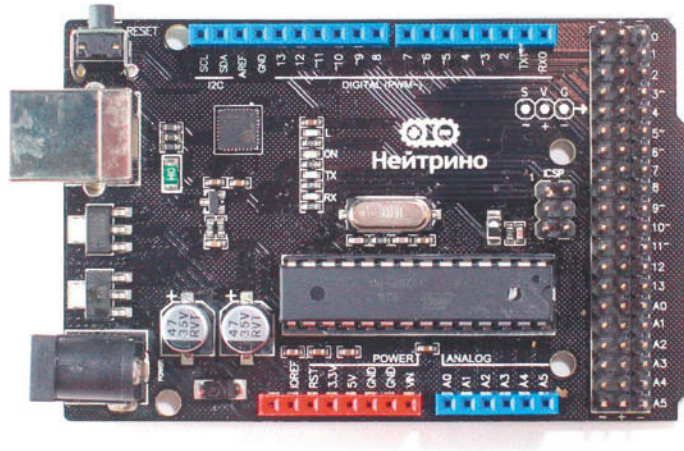
На платах Mini и Nano имеются дополнительные аналоговые входы A6 и A7, которые обусловлены бóльшим количеством контактов корпуса примененной миниатюрной версии контроллера ATmega328. В отличие от обычных аналоговых входов A0-A5, входы A6 и A7 не могут служить в качестве цифровой линии общего назначения.

В интернет-магазинах очень часто можно встретить так называемый **Arduino Pro Mini** (см. рисунок вверху справа). Плата разработана компанией SparkFun Electronics и отличается от обычного Mini наличием дополнительных контактов по коротким сторонам платы, большей частью повторяющих уже имеющиеся. По контактам на длинных сторонах платы Pro Mini и Mini полностью совместимы. Arduino Pro Mini обычно продается в виде набора — плата и отдельно штырьковые разъемы к ней. Разработчик должен сам запаять те разъемы, которые необходимо. Если все разъемы запаять заранее, плату Mini Pro может быть трудно установить на беспаечную макетную плату.

Платы Mini и Mini Pro в силу своих небольших размеров и отсутствия дополнительных микросхем, влияющих на потребление тока, очень хорошо подходят для устройств с автономным питанием на батарейках. Но у них есть серьезный недостаток в этом отношении: светодиод, сигнализирующий о включении питания. Он потребляет всего несколько миллиампер тока, но при питании экономичного прибора от батареек это может существенно сократить время работы. Поэтому при использовании плат Mini и Mini Pro в таком качестве этот светодиод лучше удалить, аккуратно отпаяв его с платы. Работе плат по прямому назначению это никак не помешает.

Как уже говорилось в предисловии, в продаже можно встретить много версий плат Arduino, выпускаемых различными фирмами: Freeduino, Seeeduino, CraftDuino и так далее. Их часто называют «клонами», что не совсем правильно: так называют точную копию, и истинные клоны Arduino выпускают многие китайские фирмы. А упомянутые версии в чем-то обязательно отличаются от оригинала — имеют больше контактов, более мощный источник питания и так далее.

В качестве примера приведем отечественную версию Uno под названием «Нейтрино» (рисунок внизу). Ее выпускает фирма RobotDyn по заказу ООО «Киберфизика». Плата «Нейтрино» очень хорошо подходит как раз к задачам нашей книги. У нее все сигнальные выводы контроллера дублированы в виде штыревого разъема на задней стороне платы, причем напротив каждого имеется по отдельному выводу питания и «земли». Поэтому к «Нейтрино» без всякой макетной платы удобно подключать различные датчики и исполнительные устройства.



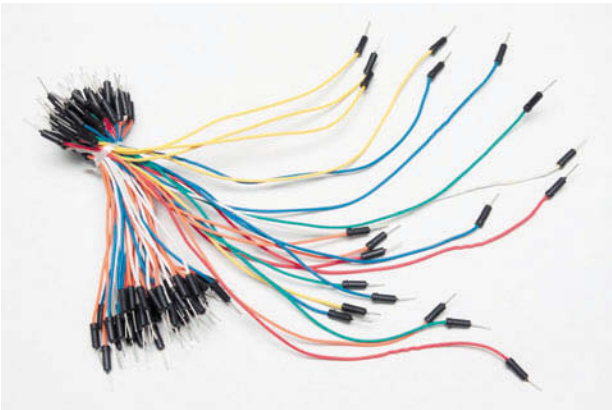
«Нейтрино» входит в комплект кибернетических конструкторов «Роботы и устройства», который производит компания «Киберфизика» (рисунок слева). Недорогой конструктор «Роботы и устройства. Старт», описание которого можно найти на сайте *cyberphysica.ru*, включает в себя многие компоненты и детали, необходимые для изучения примеров из этой книги — начиная прямо с контроллера, беспаечной макетной платы

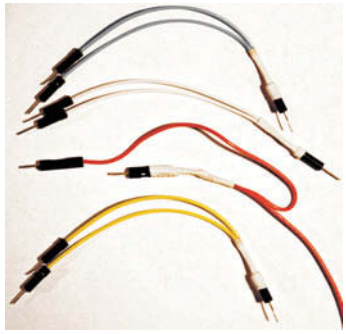
и набора соединительных проводов со штекерами (о проводах см. следующий раздел). Конечно, для целей этой книги только этого набора будет недостаточно, и придется приобретать многие компоненты отдельно. Но он может стать хорошей основой для начала, а остальное появится постепенно, по мере того, как вы будете осваивать материал.

Подключение датчиков и исполнительных устройств к платам Arduino

Для подключения различных датчиков и других устройств к Arduino к ним часто прилагаются многопроводные кабели с разъемами (см. рисунок внизу слева). Они обычно рассчитаны на подключение к какой-либо из упомянутых плат расширения (shield). На все случаи жизни таких плат, естественно, не напасешься, потому для подключения датчиков удобно иметь наборы проводов с разъемами. На рисунке внизу справа показан такой набор с разъемами «штырь»-«штырь», но они бывают разной конфигурации («штырь»-«гнездо», «гнездо»-«гнездо»).

Обычная проблема при этом заключается в недостатке контактов на платах Arduino для подключения нескольких датчиков или внешних устройств. Особенно этим отличаются датчики с интерфейсом TWI (см. *главу 4*), потому что он разрешает подключать к одним и тем же линиям в принципе сколько угодно датчиков. На современных платах Arduino Uno имеются дублирующие контакты для сигнальных линий этого интерфейса (SDA и SCL), но отсутствует дополнительное гнездо для подключения питания.





На монтажных схемах различных примеров использования датчиков для «размножения» контактов обычно показывают подключение через макетную плату, но в реальности это не очень удобно. Тут как раз и может пригодиться «Нейтрино», которая во многом заменяет дополнительные «шилды», хотя бы наличием большого количества контактов питания.

Иногда можно встретить в продаже шлейфы для датчиков, в которых провода разветвляются на два или более разъемов. Такие разветвители вполне можно изготовить самостоятельно, используя проводники с разъемами из наборов. Пример такого самодельного комплекта для подключения двух датчиков к одному интерфейсу TWI (см. *главу 4*) показан на рисунке вверху.

Среда программирования Arduino IDE

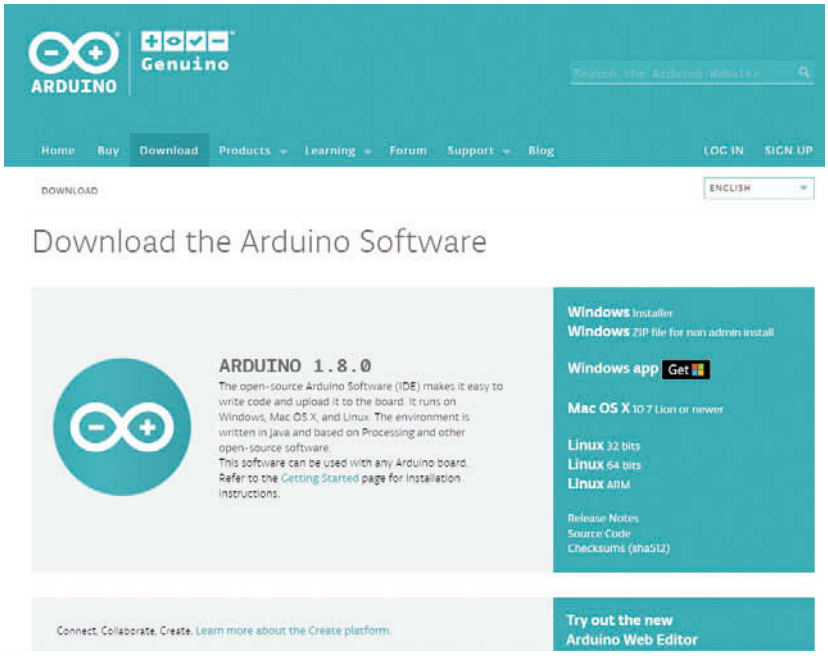
В настоящее время существуют две компании — американская и европейская, — официально выпускающие продукты под маркой Arduino. Они представлены сайтами ***arduino.cc*** и ***arduino.org***. Соответственно, какое-то время были два варианта среды программирования Arduino IDE, которые можно отличить по номерам версий. Номера версий с сайта ***arduino.cc*** начинались с цифр 1.6, а с сайта ***arduino.org*** — с 1.7. Эта книга уже была наполовину готова, когда компании, наконец, сумели объединить усилия, и последняя версия на обоих сайтах носит номер 1.8.0. На различных сайтах вы можете встретить примеры, сделанные под предыдущими версиями — они, как правило, пригодны для любой из версий среды Arduino.

Различия между версиями заключаются в основном в исправлении мелких ошибок и недоработок, а также в поддержке новых версий библиотек и плат, если они появляются. Для всех примеров из этой книги можно применять любую из старых версий с номерами 1.6 или 1.7, и в большинстве случаев даже более старые классические варианты с номерами, начинающимися с 1.0. Иллюстрации в этой главе сделаны в версии 1.8.0, примеры в остальной книге — в версии 1.6.9.

Заметьте, что мы предполагаем наличие у читателя ноутбука или настольного компьютера, работающего под Windows. Если у вас другие платформы (Mac OS или Linux), то вам придется разбираться с установкой среды самостоятельно, по инструкции, имеющейся на сайте ***arduino.cc***, а также на многих других сайтах, в том числе русскоязычных.

Сама среда программирования Arduino IDE (что расшифровывается, как Integrated Development Environment, интегрированная среда разработки) отличается от других подобных продуктов простотой и компактностью. Можно вообще ее не устанавливать автоматически — просто скачайте ZIP-архив с официального сайта (*http://www.arduino.cc/en/Main/Software*) и распакуйте его на компьютере в любую папку. На рисунке внизу страницы воспроизведена страница сайта ***arduino.cc*** с последней версией среды программирования, где видны ссылки на автоматический установщик (Windows Installer) и простой ZIP-архив для самостоятельной распаковки (non admin install).

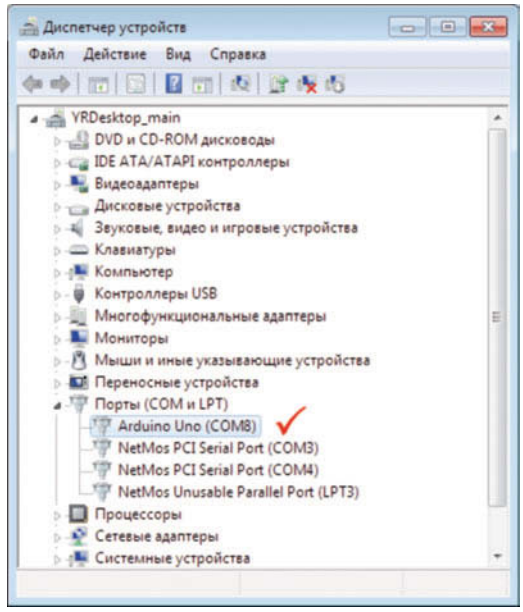
Проще, конечно, воспользоваться автоматическим установщиком, причем, если у вас русская версия Windows, то вы получите среду сразу на русском языке. Будут также установлены все нужные драйверы. Если у вас имелась старая версия, которая была установлена тем же «инсталлером», то ее лучше предварительно удалить (запустив *uninstall.exe* из папки *Arduino*). Перед этим сохраните копию папки с библиотеками (*Arduino\libraries*), которую вы неизбежно дополняли



сторонними библиотеками, и, возможно, папку с проектами (о ней далее).

Не исключено и сосуществование двух и более версий на одном компьютере. Только при этом надо учитывать, что файл предустановок и папка с проектами для всех версий 1.6 и далее окажутся общими. Кроме того, Windows может запутаться между версиями при автоматической загрузке программ из Проводника. Потому, если нет обоснованных причин для содержания нескольких версий на одном компьютере, лучше перед обновлением старую версию снести — обязательно с сохранением копии рабочих папок, как сказано выше.

В процессе установки вам, разумеется, предложат установить среду в системный каталог *Program Files* (или в *Program Files (x86)* для 64-разрядных Windows). Настоятельно советуем установить ее в отдельный каталог вне системных папок. В этом случае все вновь созданные программы и новые библиотеки можно хранить в том же месте, где саму среду Arduino. В противном случае они окажутся разбросанными по всему диску (частично в *Program Files*, частично в недрах пользовательских папок). Способ неудобный и опасный — высока вероятность потерять пользовательские папки при переустановке программы, и еще выше — при обновлении самой системы Windows. Кроме того, если вы захотите внести изменения в какие-то официальные библиотеки, входящие в Arduino по умолчанию, то изменять файлы внутри системной *Program Files* вам могут и не разрешить. Потому лучше заранее от таких трудностей застраховаться, разместив все, относящееся к Arduino, в одном месте (см. также далее о настройках среды Arduino).



По умолчанию папка с установленной средой будет называться *Arduino*. Можно сразу добавить к этому имени номер версии, чтобы потом в них не путаться (мы далее папку будем называть просто *Arduino*).

Для проверки работы драйвера сразу после установки подсоедините имеющуюся плату Arduino Uno или Nano к порту USB компьютера. Для этого, как мы уже говорили, потребуется либо обычный AB-кабель USB (плата Arduino Uno), либо кабель с разъемом USB mini (плата Arduino Nano). На плате должен при этом загореться зеленый или желтый светодиод ON. В Диспетчере устройств Windows (**Панель управления | Диспетчер устройств**) в разделе **Порты (COM и LPT)** появится название платы — например, **Arduino UNO (COM*xx*)**. На рисунке внизу предыдущей страницы показано окно Диспетчера устройств, где подключенная плата отмечена галочкой.

На случай, если при подключении плат Arduino начнутся какие-то недоразумения с драйверами, то нужный вам драйвер (файл *arduino.inf*) находится в каталоге *Arduino\Drivers* (будьте внимательны: именно в папке *Drivers*, а не в подпапке *FTDI USB Drivers*). Драйвер можно обновить прямо из **Диспетчера устройств** через пункт контекстного меню **Обновить драйвер**.

В процессе изучения Arduino обязательно возникнет необходимость установки дополнительных библиотек для работы с различными датчиками и исполнительными устройствами. В этой книге для каждого конкретного случая указывается ссылка, откуда можно скачать такую библиотеку, если она не входит в установочный комплект. Скачанная библиотека оказывается в архиве типа ZIP. Распаковать и установить ее в Arduino IDE можно через пункт меню **Скетч | Подключить библиотеку | Добавить ZIP-библиотеку**. Но не сложнее это сделать и вручную: просто скопировать из архива всю имеющуюся там папку библиотеки в каталог *Arduino\libraries* (не путать с каталогом *Arduino\LIB*!).

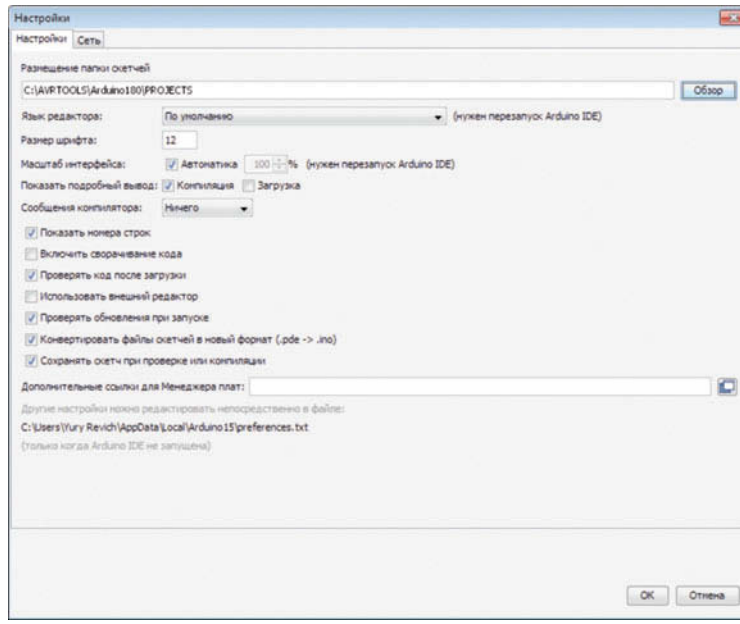
Заметим, что официально рекомендуемый вариант — вместо *Arduino\libraries* копировать новые библиотеки в папку *libraries*, которую Arduino IDE создает автоматически в папке с вашими проектами (см. далее). Никакой разницы в том или другом способе нет, кроме того, что придется рыться в поисках нужной библиотеки в двух местах вместо одного. Особенно это неудобно, когда папка с проектами расположена в пользовательских папках, а каталог Arduino — в *Program Files*, как это делается по умолчанию. Если новые библиотеки будут конфликтовать со старыми, то они будут это делать при размещении в любой папке, так что можете поступать по своему усмотрению. О том, как правильно

ссылаться на библиотеки в своих проектах, см. *Эксперимент 9*, раздел «*Управление четырехразрядным дисплеем*».

После перезапуска среды Arduino следует проверить доступность новой библиотеки, разыскав ее в списке меню **Скетч | Подключить библиотеку**. Отметим, что в этом списке новые установленные вами библиотеки будут отделены от имеющихся по умолчанию, что облегчает поиск.

Настройки Arduino IDE

Для запуска среды запустите файл *Arduino.exe* из папки, куда была установлена Ardino IDE (при автоматической установке возникнет соответствующий значок на Рабочем столе). После запуска Arduino IDE первым делом обратитесь к пункту **Файл | Настройки** (см. рисунок внизу). Там вы можете поменять язык самой программы (и, кстати, также и язык сообщений об ошибках), отказаться от проверки наличия обновлений (иначе при каждом запуске будете получать назойливые предложения сменить версию) и, главное, поменять размещение текстов ваших программ (скетчей), заданное по умолчанию. Для этого просто создайте внутри папки, содержащей *arduino.exe*, каталог с названием, например, *Projects*, и укажите его в самом первом пункте настроек через кнопку **Обзор**.

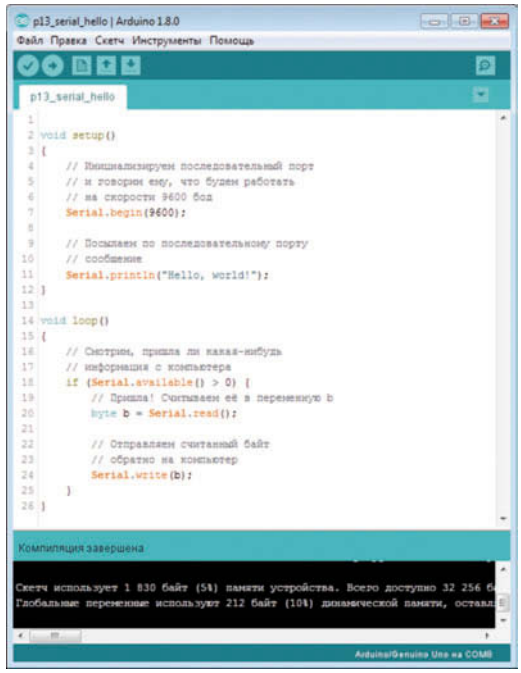


Еще раз повторим, что создание такого каталога возможно только при установке Arduino IDE вне системной папки *Program Files*, иначе придется пользоваться каталогом по умолчанию (или возиться с настройками доступа, что в Windows занятие крайне муторное).
Много разнообразных настроек доступны через файл *preferences.txt*. Его размещение указано внизу окна настроек, изменить его тоже в принципе можно, но не нужно, потому что обычно там менять ничего не требуется. Но иногда туда лезть все-таки приходится: так, обладателям большого монитора размер окна Arduino IDE по умолчанию покажется мелковатым, а запоминать размер среда умеет не во всех версиях. Для изменения этого параметра вручную следует отредактировать строки `editor.window.height.default` и `editor.window.width.default` (установив, например, 1000 и 800, соответственно). Только не забудьте, что перед внесением изменений в *preferences.txt* следует создать его резервную копию.

Пробная загрузка

Так как среда Arduino создавалась для любителей, то программы в ней носят несерьезное название «скетчи» (от англ. sketch — эскиз, набросок). Для пробы проще всего воспользоваться каким-то из готовых примеров, которые размещаются в папке *Arduino\Examples* (они также доступны через меню **Файл | Примеры**). Для большего разнообразия целесообразно дополнить их примерами, созданными компанией «Амперка». Архив с этими примерами можно скачать с сайта автора по адресу *http://revich.lib.ru/AVR/amperka-primery.zip*. Распакуйте все его содержимое в папку *Examples*, внутри которой теперь должна образоваться папка с именем *Amperka*, содержащая много дополнительных примеров из разных областей.

Попробуем что-нибудь загрузить в контроллер. Сразу подключать плату необязательно, сначала стоит проверить и при необходимости отладить скетч. Для загрузки примера войдите в меню **Файл | Примеры**, разыщите в списке пункт **Амперка** и, пройдя по стрелочке, загрузите пример под названием p13_serial_hello. В результате в окне появится текст примера (см. рисунок на следующей странице).
Следующим шагом следует текст превратить в загружаемую программу на языке, понятном контроллеру. Такая процедура называется *компиляцией*. При ее выполнении программа-компилятор, встроенная в Arduino IDE, заодно проверяет текст на наличие ошибок. Скомпилировать скетч можно через пункт меню **Скетч | Проверить/скомпилировать** (или через нажатие комбинации клавиш Ctrl+R). При успешном окончании



процесса в нижней части окна появится надпись «Компиляция завершена», а еще ниже на черном поле отобразятся сведения о количестве занимаемой памяти контроллера (см. рисунок). Если в тексте программы (скетча) будут ошибки, то компиляция прервется и сообщения о причинах появятся в том же черном окне внизу.

Затем подключите плату к компьютеру через USB-кабель. Можно проверить, что плата определилась и заодно узнать соответствующий ей номер COM-порта через Диспетчер устройств, как описано ранее. При последующих подключениях через то же самое гнездо USB компьютера номер порта будет тот же самый. Зайдите в меню **Инструменты | Порт** и выберите из списка нужный номер, после чего он появится в самом низу окна (в примере на рисунке это COM8). К сожалению, Ardiuno IDE не определяет подключенную плату автоматически, потому ее тип тоже *обязательно* нужно указать. Указание типа подключенной платы выполняется через пункт **Инструменты | Плата**, после чего название появится перед наименованием порта (в нашем случае это Arduino/Genuino Uno).

> Указывать платы следует потому, что их тип не всегда определяется верно (а при подключении, например, Arduino Mini через USB-адаптер не определяется совсем), и при загрузке программы могут возникнуть ошибки. Во избежание недоразумений в таких случаях следует привыкнуть всегда явно указывать тип платы перед загрузкой.

Вот теперь все подготовлено для загрузки программы. Обратитесь к меню **Скетч | Загрузка** (или нажмите сочетание клавиш Ctrl+U). Отметьте, что программа перед загрузкой будет компилироваться заново, так что ошибки не пройдут, даже если вы забыли проверить текст. Если все правильно, то по окончании процесса внизу окна с текстом появится надпись «Загрузка завершена». В некоторых версиях такое