



# Содержание

<b>Введение .....</b>	<b>5</b>
<b>Часть 1. Практика на базе набора Arduino Uno R3 с RFID-модулем .....</b>	<b>9</b>
Введение .....	10
Состав комплекта .....	12
Немного о макетной плате, резисторах и безопасности .....	17
Практическое занятие 1. Hello, world!.....	23
Практическое занятие 2. Эксперимент с мигающим светодиодом .....	25
Практическое занятие 3. Эксперимент с контролируемой потенциометром яркостью свечения светодиода через порт PWM .....	26
Практическое занятие 4. Светофорный эксперимент .....	27
Практическое занятие 5. Эксперимент с пищалкой .....	29
Практическое занятие 6. Эксперимент с датчиком наклона .....	30
Практическое занятие 7. Эксперимент с чистым входным сигналом .....	32
Практическое занятие 8. Эксперимент по управлению звуком и светом....	33
Практическое занятие 9. Эксперимент с вольтметром .....	35
Практическое занятие 10. Разноцветный термостат .....	36
Практическое занятие 11. Эксперимент с четырехразрядным цифровым светодиодным индикатором .....	38
Практическое занятие 12. Эксперимент со светодиодной матрицей.....	43
Практическое занятие 13. Эксперимент с трехцветным светодиодом .....	48
Практическое занятие 14. Эксперимент с модулем 74HC595.....	50
Практическое занятие 15. Часы реального времени DS1307.....	52
Практическое занятие 16. Эксперимент с датчиком уровня воды .....	54
Практическое занятие 17. Эксперимент с сенсором температуры и влажно- сти DHT11.....	56
Практическое занятие 18. Эксперимент с релейным модулем.....	57
Практическое занятие 19. Эксперимент с жидкокристаллическим монитором LCD1602A .....	59
Практическое занятие 20. Эксперимент с серводвигателем .....	60
Практическое занятие 21. Эксперимент с инфракрасным пультом дистанционного управления .....	62
Практическое занятие 22. Эксперимент с RFID-модулем RC522 .....	66
Практическое занятие 23. Эксперимент с системой контроля доступа .....	69
<b>Часть 2. Практика на Raspberry Pi 3 или 4 .....</b>	<b>71</b>
Введение.....	72
Подготовка .....	75

Первый проект – трехцветный светодиод .....	80
Второй проект – шпионский глаз .....	85
Третий проект – система мониторинга окружающей среды .....	98
Четвертый проект – система для управления освещением .....	105
Пятый проект – удаленная метеостанция .....	138
Шестой проект – работа с облачной платформой интернета вещей .....	156
Седьмой проект – система обнаружения проникновения в дом .....	180
Восьмой проект – голосовое управление умным домом .....	218
<b>Заключение .....</b>	<b>232</b>
<b>Список использованных источников .....</b>	<b>234</b>

# Введение

Интернет вещей – активно развивающееся направление, несмотря на отсутствие единых стандартов и разнообразие платформ и самих интернет-вещей, предлагающихся различными компаниями в виде готовых решений со своим дизайном. Существует огромное количество определений этого термина, однако в большинстве случаев под интернетом вещей понимается сбор и обмен данными между различными физическими устройствами (также называемыми умными устройствами, подключенными устройствами, интернет-вещами и т. д.) на основе определенной сети (здесь иногда добавляют фразу «там, где раньше это было невозможно»). Физическими устройствами могут быть автомобили, здания, камеры, бытовая техника, компьютерная техника и любые другие устройства, оснащенные электроникой, программным обеспечением, сенсорами, двигателями и модулями для подключения к сети Интернет, даже города. Данные, получаемые с физических устройств, как правило, собираются в определенном сетевом хранилище – дата-центре, облаке, сервере, репозитории и т. д., которые затем могут быть подвержены анализу методами data mining, machine learning, cloud computing и др. с целью решения тех или иных задач, стоящих перед разработчиками. Для хранения, обработки и визуализации этих данных, а также для предоставления различных сервисов для управления интернет-вещами и анализа данных существуют различные облачные платформы для интернета вещей.

Если взглянуть на кривую компании Гартнер (Gartner) по появляющимся технологиям (emerging technologies), то можно увидеть интернет вещей на пике в 2014–2015 годах, исчезновение IoT (Internet of Things, интернета вещей) в 2016–2017 годах, но замену его в этот период новой технологией – платформой интернета вещей (IoT Platform) – и затем перемещение платформы интернета вещей на кривой в 2018 году ближе к области спада и разочарования. Однако, как известно, реальные надежные технологии (второй версии) появляются только после прохождения через эту впадину избавления от иллюзий. Несмотря на исчезновение интернета вещей из области хайпа на кривой Гартнер в 2019 и последующих годах, область IoT активно развивается, о чем свидетельствуют статьи в открытом доступе той же компании Гартнер, например [1], и основанные на них статьи других авторов, например [2]. Согласно [2], например, к 2030 году прогнозируется количество IoT-устройств в мире в районе 29 млрд. Согласно цитате в этой статье от Forbes, любая современная промышленность сегодня – это промышленность с интернетом вещей. Поэтому знать основы этой технологии и уметь применять различные подходы к построению систем интернета вещей для различных задач должен уметь каждый уважающий себя программист или IT-специалист.

С интернетом вещей вплотную я познакомился в 2016 году. До этого я, конечно, слышал об этом направлении, или области, – ведь один из студентов, руководителем которого был Восков Л. С., еще в МИЭМе (ныне – в составе НИУ

ВШЭ) показывал мне свой проект по умной розетке, которая контролировалась удаленно через веб-интерфейс и зажигала включенную в нее настольную лампу. На слуху были разнообразные умные вещи – умный дом, умный автомобиль, умный холодильник, умная подставка для яиц в холодильнике и т. д. Безусловно, тогда это все было непросто – датчики и сенсоры, платы и прочее оборудование стоили немало. Но потом появилась потребность подготовить курс «Экосистемы интернета вещей», и... началось. Прежде всего было непонятно, что делать с практическими занятиями по этому курсу – что давать студентам? «Железок» никаких не было и в помине, в то же время очень хотелось, чтобы студенты работали на реальном оборудовании. Но на каком? Как построить практику? Времени оставалось все меньше, и тогда я решил попробовать купить комплект с Raspberry Pi 3 – тогда, в 2017 году, несмотря на то что операционная система Android Things была еще без официального релиза первой версии, по ней были довольно интересные примеры и проекты на официальном сайте разработчиков под Android Things, и не только там. Однако, начав разбираться с системой, я понял, что начинать надо не с этого, а с самых основ, с физики, схемотехники, которых у студентов-программистов, для кого предназначался курс, конечно же, не было. Поэтому вторым шагом была покупка комплекта с платой Arduino Uno – такого, в котором было бы максимально возможное количество «железок». Этот комплект заставил вспомнить основы физики, научиться терпению при определении ошибки в проекте – в схеме она, или в компонентах, или в коде, или в методичке, или где? – и многому другому, в том числе пришлось пропустить через себя методичку с огромным количеством ошибок на ломаном английском языке, которая была на сайте магазина, продававшего комплект. Так появилась первая часть этой книги. Тогда же, когда в моем распоряжении оказались Raspberry Pi 3, HDMI-минимонитор и Android Things, на книжной полке внезапно появилась и книга Android-разработчика Франческо Эззолы «Android Things Projects», которую пришлось печатать на заказ – недешевое удовольствие. И, несмотря на устаревший на 1 год код, английский язык и некоторые ошибки в заданиях или просто иногда пропуск некоторых важных деталей кода в книге, в 2018 году выросла вторая часть этой книги, посвященная нескольким проектам под Android Things на плате Raspberry Pi 3. Они были приведены к реалиям существующей на тот момент 1-й версии этой операционной системы и переработаны с целью добавления пользовательского интерфейса, которого в оригинале в некоторых заданиях просто нет. Осваивая книгу и пропуская задания через себя, я понимал, что вот оно – как это просто сделать, оказывается: контроль и полив растения удаленно через смартфон, контроль за удаленной квартирой с помощью ИК-датчика и камеры, пока ты находишься в отпуске, или получать всю информации о погоде за окном на твоём мониторе в комнате в виде Android Things-приложения, включая прогноз погоды.

Однако шло время, и программный код, библиотеки и даже операционная система начали устаревать. К сожалению, данное учебное пособие относится к очень быстроразвивающейся отрасли из области ИТ (Information Technologies, информационных технологий), в которой все устаревает с невероятной скоростью. Сегодня ты написал код в Android Studio Flamingo, а завтра уже вышла Android Studio Giraffe, где этот код не работает, и надо разбираться почему. Сначала, в мае 2019 года, компания Samsung объявила о закрытии облака Artik

Cloud, на котором был основан один из проектов второй части учебного пособия, для Raspberry Pi. Затем стали появляться некоторые нововведения, относящиеся к коду в Android Studio, который надо было писать для приложений под операционную систему Android Things. Новые версии Android Studio выходят дважды в год, и в следующий учебный год код, написанный и протестированный в предыдущем учебном году, конечно же, переставал работать, не говоря уже о поддержке нужных шаблонов для приложений интернета вещей. Затем в январе 2021 года случилось непоправимое – операционная система Android Things, на которой были основаны все проекты второй части учебного пособия, перестала поддерживаться, а позже – и существовать. И, несмотря на то что можно было по-прежнему разрабатывать проекты под эту ОС, так как существовал образ на SD-карте, последнее обновление этого образа было от 2019 года, когда был анонсирован переход платформы Android Things на коммерческие рельсы, и он больше не обновлялся – нельзя было скачать новую версию Android Things. Из Android Studio исчез шаблон приложения для интернета вещей. Позже появилась действительно серьезная проблема: в 2022–2023 учебном году у студентов в аудитории платы Raspberry Pi с установленной Android Things начали самопроизвольно перезагружаться из-за недостатка энергии, вызванного встроенным Wi-Fi-модулем. К сожалению, причину этой проблемы найти удалось уже ближе к завершению курса... и стало понятно, что пора уже переходить на новую операционную систему, которой незнакомы подобные проблемы. В качестве новой ОС была выбрана специально разработанная для Raspberry Pi ОС, которая теперь так и называется – Raspberry Pi (бывшая Raspbian OS). Но все приложения для Android Things пришлось, во-первых, переписать с Java на Python, одновременно изучив и выбрав некоторые библиотеки; а во-вторых, пришлось создавать с нуля интерфейс для этих приложений, также изучая и выбирая необходимые библиотеки, самой подходящей из которых оказалась PySimpleGUI. Таким образом, в новой версии учебного пособия, которую вы читаете, полностью переработаны все проекты для Raspberry Pi, а также добавлены еще три проекта, и все они написаны на связке Arduino IDE + Thonny (IDE в Raspberry Pi OS для программирования на Python) + Android Studio (Java) для мобильных приложений, где это необходимо, и работают под управлением свежих версий операционных систем Raspberry Pi и Android.

Первая часть этой книги предназначена для того, чтобы читатели (студенты) освоили или вспомнили основы схемотехники. Да-да, именно в этом и заключается предназначение комплекта модулей, датчиков, проводов, экранов, двигателей, джойстиков, индикаторов, карт и ключей, резисторов и светодиодов с кнопками, вместе с макетной платой и платой Arduino Uno – для того чтобы читатели освоили прежде всего схемы подключения, схемотехническую базу и вспомнили основы физики в части тока и напряжения, а также номиналы резисторов и как их посчитать, – на реальных устройствах, которые могут выйти из строя или работать неправильно при неправильном подключении. Задача же второй части данного учебного пособия – перейти от схемотехники к реальным проектам для интернета вещей с их типовой архитектурой, передачей данных от сенсоров к плате, от платы в облако, из облака в мобильное приложение-компаньон, позволяющее осуществить управление платой удаленно или визуализировать и проанализировать полученные данные и, глав-

ное, – воплотить в жизнь реальную архитектуру приложений для интернета вещей. В новой версии учебного пособия количество заданий первой части сокращено с 34 до 23, так как от читателя не требуется настолько подробно изучать схемотехнику, по моему мнению, и лучше сделать основной упор на вторую часть, посвященную именно проектам интернета вещей.

Практические эксперименты первой части учебного пособия построены по определенной схеме. Сначала перечисляются компоненты, необходимые для выполнения задания. Затем приводится описание задания – та часть, которую все обычно пропускают (особенно студенты). После этого приводится цветная схема с макетной платой, объясняющая, как соединить все компоненты заданий друг с другом. Принципиальные электрические схемы соединения компонентов я решил в этой новой версии учебного пособия не приводить, все равно большинство смотрит именно на вторую схему с макетной платой, так как она проще для восприятия, цветная и более интересная, чем принципиальная схема. Затем идет код программы (иногда их несколько), который необходимо скопировать в Arduino IDE и запустить.

Проекты из второй части построены немного по-другому – сначала идет небольшое введение, потом – перечень компонентов и их свойства и изображения. После этого приводится схема подключения компонентов с макетной платой. А затем идут этапы выполнения задания, в которых присутствуют как исходный код для копирования в Android Studio или Thonny, так и пояснения к этому коду, а также, при необходимости, – скриншоты различных ресурсов и платформ интернета вещей, используемых для выполнения задания, и скриншоты работающих в результате выполнения проектов приложений.

Отдельно хочется выразить слова благодарности студентам 3-го курса 2017–2018 и последующих учебных лет вплоть до текущего 2023–2024 учебного года образовательной программы «Программная инженерия» департамента программной инженерии факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики», на которых проходила апробация всех заданий из этого пособия. Они не только находили и исправляли ошибки и неточности в некоторых заданиях, но и предлагали разумные вещи для улучшения методических указаний, послуживших основой для этой книги. И конечно, нельзя не сказать эти слова бывшему руководителю упомянутого департамента – Авдошину С. М., который в свое время ошарашил меня курсом «Экосистемы интернета вещей», без которого не было бы никакого учебного пособия.

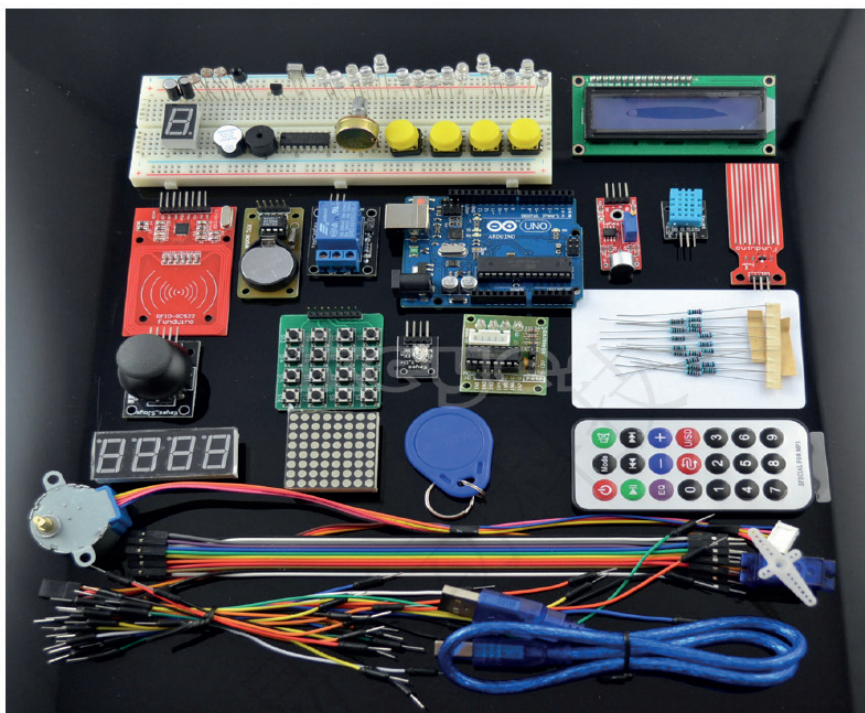
Для выполнения заданий из этого учебного пособия необходимо приобрести как минимум 2 комплекта, например такие, как в [3] (Arduino Uno R3) и [4] (Raspberry Pi 3 или 4). Также отдельно могут понадобиться дополнительные модули и датчики для выполнения заданий из 2-й части учебного пособия, а именно кардридер для чтения/записи microSD-карт, модуль с Raspberry Pi камерой v 2.1, PIR-сенсор, модуль Тройка Wi-Fi, датчик давления, влажности и температуры BME280 или BMP280, GPIO-адаптер для Raspberry Pi 3 или 4 (с ним удобнее) и модуль аналогового датчика света (освещенности). И конечно же, понадобится второй монитор для Raspberry Pi 3 или 4, и этот монитор должен быть с HDMI-входом.

При подготовке данного учебного пособия не только ни одна плата, но и ни один светодиод, модуль или датчик не пострадали.



# Часть 1

## Практика на базе набора Arduino Uno R3 с RFID-модулем





## ВВЕДЕНИЕ

Arduino – наиболее популярная платформа для разработки как простых, так и достаточно сложных проектов для интернета вещей. Популярность этой платформы обусловлена не только ее низкой ценой, но и огромным количеством обучающих материалов, примеров проектов в сети, в том числе и на таких ресурсах, как YouTube, различных форумах разработчиков, а также – хорошим официальным сайтом [5, 6], на котором Arduino Team постоянно предлагает ознакомиться с новыми проектами на базе одной из плат Arduino. Существует множество разновидностей плат Arduino, например: Uno, Leonardo, 101, Mega, Zero, Ethernet, Gemma, MKR FOX 1200 и т. д. Все платы на официальном сайте разделены на категории: начальный уровень (для обучения, к нему относится Uno), платы с расширенными функциями, платы для интернета вещей, платы для носимых устройств (в основном в тандеме с Lillypad, например для умной одежды). Несмотря на простоту среды Arduino IDE и ее недостатки, в ней можно разрабатывать интересные и сложные проекты; кроме того, существует официальный онлайн-аналог среды и многочисленные библиотеки, способные нарастить ее небольшой функционал. Именно поэтому для данного учебного пособия выбрана эта платформа, а именно Arduino Uno версии R3, и наиболее богатый и разнообразный в отношении количества различных модулей, сенсоров, датчиков и других компонентов комплект с этой платой – набор для начинающих на базе Arduino Uno версии R3 с RFID-модулем [3].

Arduino Uno Rev. 3 – лучшая плата для начинающих. Согласно официальному сайту эта плата является наиболее используемой и обладает наибольшим количеством документации из всех плат семейства Arduino. Arduino Uno – это плата на основе 8-битного микроконтроллера ATmega328P, см. рис. 1 – самая большая черная деталь на плате. Таким образом, по сути плата Arduino Uno является платой расширения или платой разработчика (developer board), сердцем которой является ATmega328P.

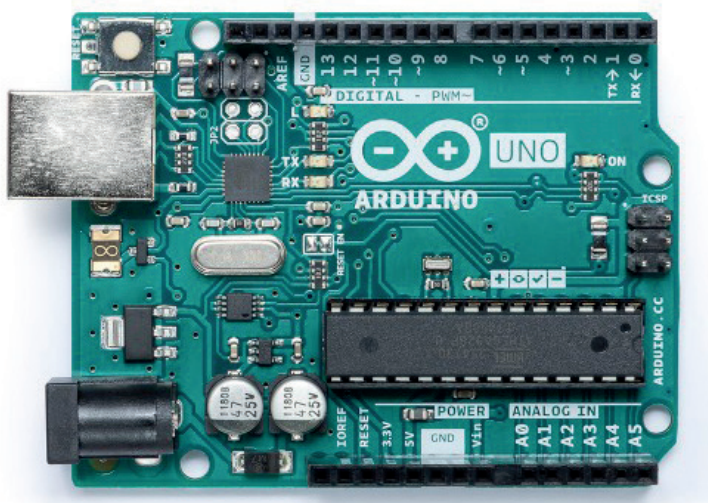


Рис. 1 ❖ Лицевая сторона оригинальной платы Arduino Uno

На плате есть и другой микроконтроллер – ATmega16U2, служащий для связи микроконтроллера ATmega328P и USB-порта платы (на рис. 1 – черный квадрат слева от TX и RX). У Arduino Uno есть 14 цифровых выходов (пинов) ввода/вывода, обозначенных на плате цифрами (из них 6 ШИМ (PWM, Pulse-Width Modulation), выходных пинов широтно-импульсной модуляции, обозначенных символом ~), 6 аналоговых входов – A0-A5 (с разрешением в 10 бит, т. е. 1024 различных значений), кварцевый кристалл-резонатор на 16 МГц, выход USB-Bf (на рис. 1 – слева сверху под кнопкой перезагрузки), выход для подключения питания от адаптера питания (7–12 В) или батарейки на 9 В (обычно в виде прямоугольного параллелепипеда; выход находится слева внизу на рис. 1), ICSP-разъем (In Circuit Serial Programming, программирование по последовательному протоколу чипа, уже подключенного в некоторую схему, или просто – программирование контроллера внутри схемы; на рис. 1 – посередине правого края) и кнопка перезагрузки (слева сверху, см. рис. 1). Кроме этого, плата располагает тремя пинами земли (GND), одним пином на 5 В, одним – на 3.3 В, Vin-пином для подключения внешнего источника питания или для получения напряжения, если плата подключена к внешнему адаптеру питания через разъем питания (через USB-соединение питание ограничивается 5 В), IOREF-пином (Input Output Reference – информация о напряжении микроконтроллера) и встроенным светодиодом L (или 13).

В некоторых наборах с Arduino Uno есть другая разновидность этой платы – китайский аналог, который ничем не хуже, но стоит в несколько раз меньше. Отличить его можно по микроконтроллеру слева от TX и RX (мосту к USB) – вместо ATmega16U2 на плате стоит микроконтроллер CH340G, который выглядит по-другому, см. рис. 2 (не квадратный, как ATmega16U2, а прямоугольный). Если Arduino Uno с ATmega16U2 сразу видна в системе как подключенная к порту COM3, так как работает с реальным портом COM, Arduino Uno с CH340G требует установки специального драйвера, который можно найти, например, в [7], потому что она эмулирует последовательный COM-порт в операционной системе. Набор из источника [3] – это набор именно с такой разновидностью платы Arduino Uno.

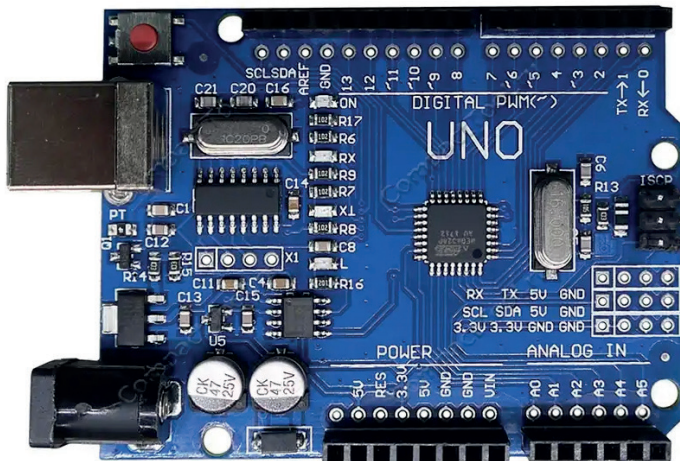


Рис. 2 ❖ Лицевая сторона платы-аналога Arduino Uno с микроконтроллером CH340G

Микроконтроллер ATmega328 на Arduino Uno поставляется уже с загрузчиком (bootloader), что позволяет загружать код без использования внешнего прогнатора. При желании можно обойти загрузчик и запрограммировать микроконтроллер через ICSP. ATmega328 обладает 32 Кб встроенной памяти (0.5 Кб из которых отведены загрузчику). В дополнение ко всему некоторые пины платы имеют несколько функций, например пины 2 и 3 могут использоваться для вызова внешних прерываний при некоторых событиях, например при событии смены высокого сигнала на пине на низкий (falling edge). Мы не будем углубляться в мельчайшие подробности характеристик платы – приведенной информации вполне достаточно для знакомства с платой и выполнения 23 практических заданий части 1 этого учебного пособия. Также здесь не будет приведена распиновка платы, так как, по сравнению с платой Raspberry Pi 3 или 4, с которой мы познакомимся в части 2 этого учебного пособия, для Arduino Uno делать распиновку нет смысла – все пины уже подписаны на плате, см. рис. 1.

Далее рассмотрим, что же входит, помимо самой платы Arduino Uno R3, в состав упомянутого набора на базе Arduino Uno версии R3 с RFID-модулем. Этот комплект был выбран также и потому, что позволяет сделать очень много практических заданий на основе различных компонентов и устройств, которые в него входят: конечно же, число возможных проектов на его основе далеко не ограничивается 23 экспериментами, приведенными в этой части учебного пособия, а ограничивается лишь фантазией разработчика. Практические задания предусмотрены почти для всех компонентов этого комплекта, чтобы познакомить читателя с большинством составных частей набора и их функциями. В следующем разделе приведены реальные фотографии компонентов комплекта, которые мы будем использовать, как они выглядят на самом деле, а не схематичные изображения или фотографии из интернета.

## СОСТАВ КОМПЛЕКТА

Набор Arduino Uno R3 с RFID-модулем [3] состоит из многих компонентов, но на рис. 3–3+0 далее показаны те, которые мы будем использовать в этой части учебного пособия.

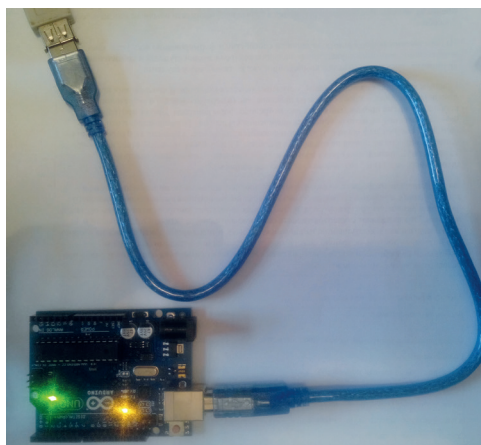


Рис. 3 ❖ Плата Arduino Uno R3 + USB-кабель (Аm-Bm)

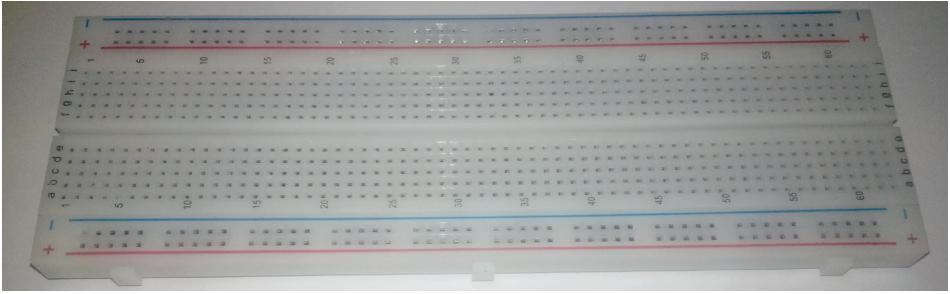


Рис. 4 ❖ Макетная плата (breadboard)

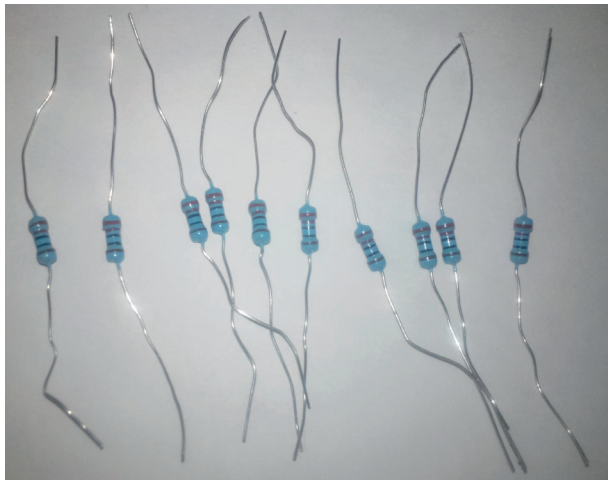


Рис. 5 ❖ Резисторы на 220 Ом, 10 шт.

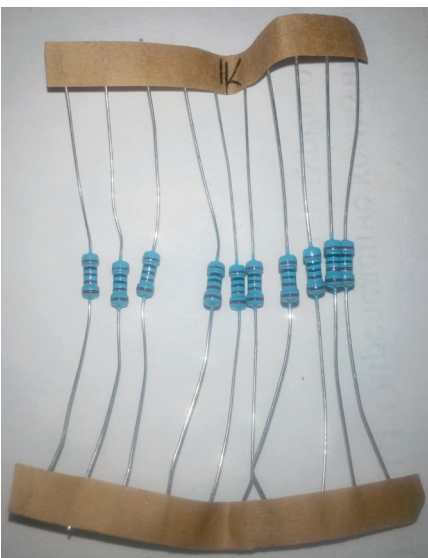


Рис. 6 ❖ Резисторы на 1 кОм, 10 шт.

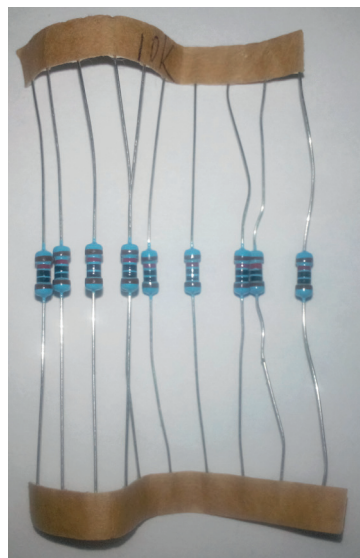


Рис. 7 ❖ Резисторы на 10 кОм, 10 шт.





Рис. 8 ❖ Светодиоды, 15 шт. (5 синих, 5 желтых, 5 красных)



Рис. 9 ❖ Кнопки, 4 шт.

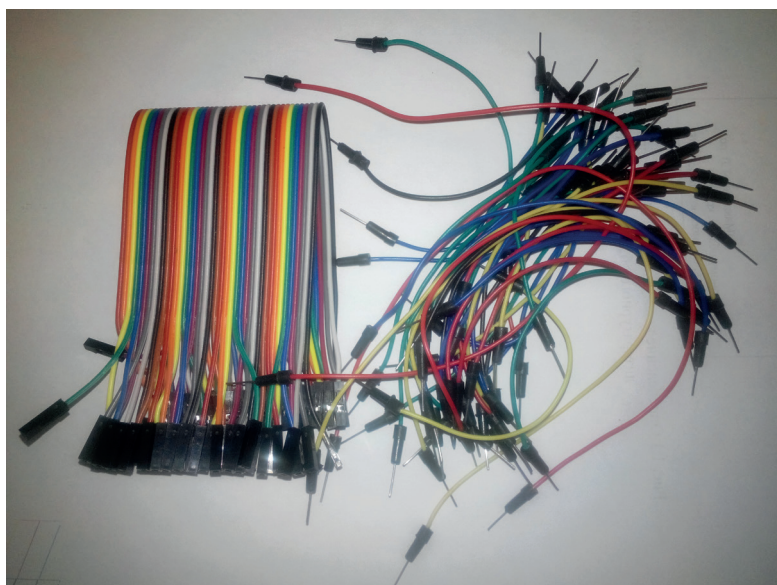


Рис. 10 ❖ Соединительные провода: m-m и f-f





Рис. 17. Четырехразрядный цифровой светодиодный индикатор

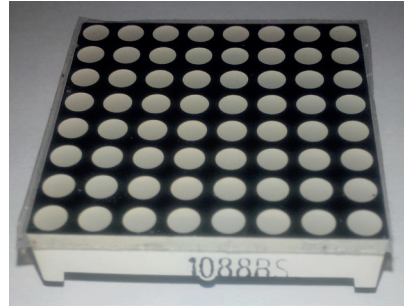


Рис. 18. Светодиодная матрица 8×8

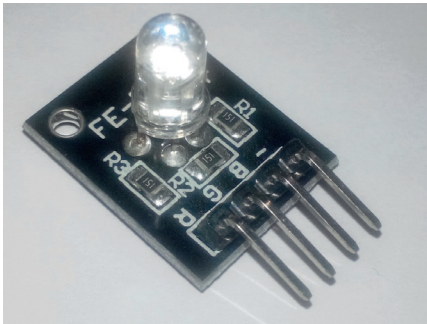


Рис. 19 ❖ Трехцветный светодиод с общим катодом (на модуле)

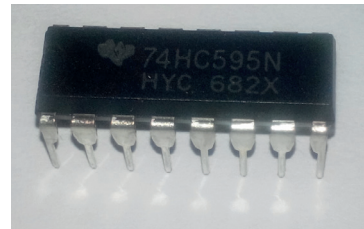


Рис. 20 ❖ Модуль 74HC595

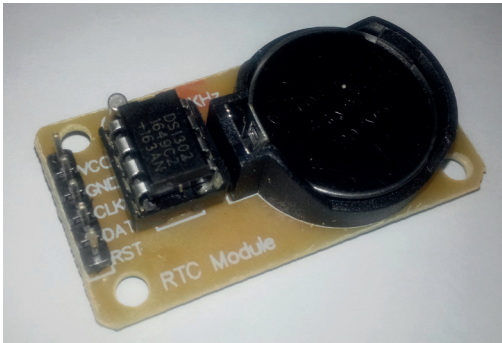


Рис. 21 ❖ Часы реального времени RTC DS1307

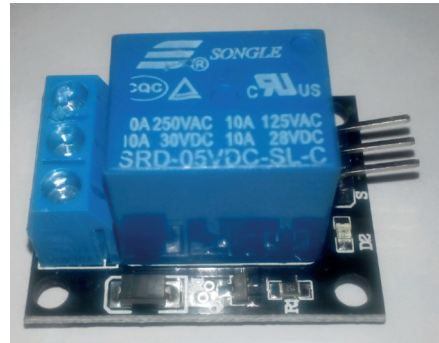


Рис. 22 ❖ Релейный модуль



Рис. 23 ❖ Датчик уровня воды (Water Sensor)

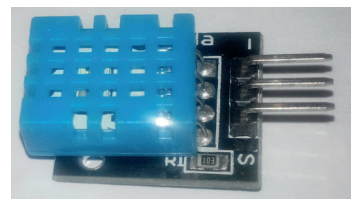


Рис. 24 ❖ Сенсор температуры и влажности DHT11





Рис. 25 ❖ Серводвигатель (сервопривод) с комплектом насадок



Рис. 26 ❖ RFID-ключ



Рис. 27 ❖ Инфракрасный пульт дистанционного управления NEC

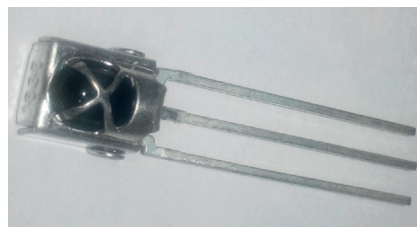


Рис. 28 ❖ Инфракрасный приемник управления

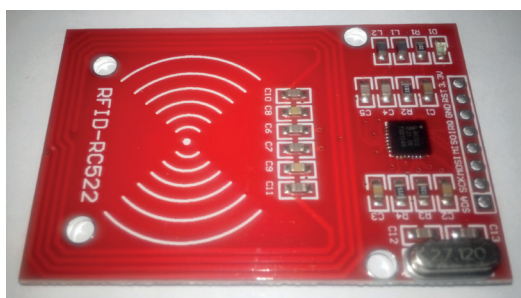


Рис. 29 ❖ RFID-модуль RC522



Рис. 30 ❖ RFID-карта

## Немного о макетной плате, резисторах и безопасности

Как в этой, так и во второй части книги вам понадобятся знания о том, что такое макетная плата, как включать в схему светодиод и как отличить один резистор от другого. Но сначала – о безопасности.

Главное правило обращения с электричеством, компонентами и модулями гласит: помните, что как вы можете повредить технику, так и она может нанести вам вред! Перед тем как выполнять задания, нужно помнить о простых

правилах работы с электронными компонентами и тем более – системами на модуле (SoM, System on Module), к которым относится Arduino Uno, платами и прочими электронными изделиями:

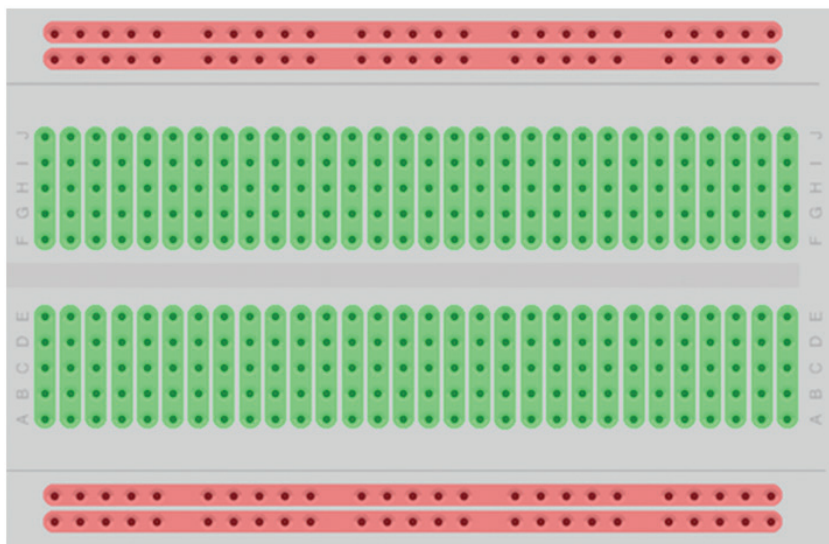
- собирать и разбирать/менять схему можно только при выключенном питании (отсоединенном USB-кабеле) – имейте терпение;
- светодиоды и другие чувствительные компоненты подключаются строго согласно схеме – через резисторы;
- не стоит путать питание с землей, плюс с минусом;
- если пошел дымок, например из потенциометра или макетной платы, или если вы чувствуете, что температурный сенсор, например, слишком нагрелся и показывает в мониторе порта Arduino IDE  $+326^{\circ}$  место  $25^{\circ}$  – немедленно отключите питание от платы и дальше разбирайтесь, что не так у вас в схеме;
- никогда не применяйте грубую физическую силу, если что-то не вставляется куда-то, например microSD-карта не вставляется в переходник или в саму плату Raspberry Pi, – проверьте, правильно ли вы это делаете, той ли стороной, и т. д.;
- никакого статического электричества! Если на вас свитер из синтетики или шерсти, если вы любите часто поправлять свои волосы, заземляйте свои руки перед тем, как дотрагиваться до электронных изделий (дотроньтесь до корпуса компьютера, железной ножки стола, батареи и т. д.)!

Макетная плата – удобное средство для соединения электрических компонентов в простые схемы и даже в схемы среднего уровня сложности. Макетные платы бывают разных типов, но в основном выделяют 2 типа: 1) с разрывом горизонтальных линий земли и питания сверху и снизу посередине и 2) без разрыва. В комплекте вам могут попасться оба типа. Макетная плата без разрыва горизонтальных линий земли и питания сверху и снизу показана на рис. 4. Если разрыва нет, это явно показывается синими и красными линиями: на рис. 4 линии идут непрерывно, значит, разрыва нет. В случае наличия разрыва красные и синие линии прерываются посередине платы.

Соединения макетной платы без разрыва линий земли и питания показаны на рис. 31. Соединения макетной платы с разрывом этих линий, соответственно, проходят снизу и сверху по горизонтали от краев только до середины макетной платы. Что же касается вертикальных соединений, у макетных плат обоих рассмотренных типов соединения прерываются 3 раза по вертикали: между зелеными разъемами верхнего и нижнего ряда на рис. 31 нет соединения, так же как и между зелеными и красными рядами.

Все схемы с макетной платой в этой книге нарисованы с помощью наиболее распространенной и популярной открытой библиотеки + редактора электронных компонентов и схем Fritzing [8].

Теперь – светодиод. У каждого светодиода есть короткий и длинный выходы (пины) – см. рис. 8. Они представляют, соответственно, катод и анод. На схеме у светодиодов эта особенность выражена следующим способом: более длинный пин (анод) изогнут у основания цветной колбы светодиода, более короткий же (катод) входит в колбу прямо, без изгиба (см., например, рисунок с макетной платой к практическому занятию 3 в этой части учебного пособия). При подключении светодиода надо помнить, что ток по нему может протекать



**Рис. 31** ❖ Соединения макетной платы без разрывов верхних и нижних горизонтальных линий питания и земли

только в одном направлении – от анода к катоду (светодиод – вид диода, который работает только в одном направлении), таким образом, анод (длинный пин) всегда подключается к источнику питания или управляющему сигналу, тогда как катод (короткий пин) обычно подключается к земле через сопротивление, ограничивающее ток, протекающий через светодиод. Светодиод нельзя подключать без сопротивления, иначе он может сгореть. Чем больше сопротивление в схеме со светодиодом (от 220 Ом до 10 кОм), тем меньше яркость свечения светодиода (меньший ток проходит через него). Подробнее о светодиодах можно почитать, например, в источнике [9].

И наконец, резисторы. На схемах резисторы обозначаются следующим образом (слева – в англоязычных источниках, справа – в русскоязычных):



**Рис. 32** ❖ Обозначение резисторов на принципиальных схемах [10]

У каждого резистора есть номинал: 220 Ом, 1 кОм и т. д. Резисторы, входящие в комплект с Arduino Uno и в другие комплекты с другими платами, обладают цветовыми насечками, см., например, рис. 5–7. Каждый цвет обозначает цифру, от 0 до 9, и цветовых насечек на резисторе несколько: таким образом, можно определить номинал резистора, пользуясь правилами, изображенными на рис. 33 [10]. Подобные резисторы, несмотря на их размер, все же встречаются в реальных схемах, используемых в промышленности, в бытовой технике и в других областях: например, управляющая плата холодильника Whirlpool собрана с помощью таких резисторов, поскольку холодильник большой, и делать миниатюрную плату с применением сверхточных технологий, которые

используются при производстве материнской платы для настольного компьютера, не имеет смысла.

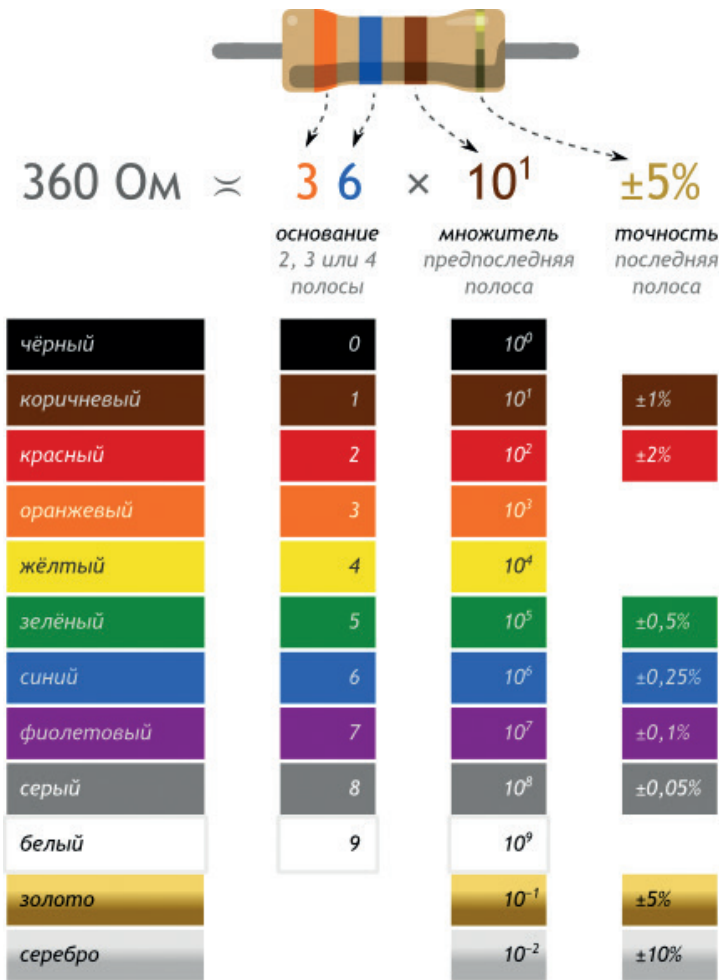


Рис. 33 ❖ Правила подсчета номинала резистора и его точности [10]

Обладая знаниями из таблицы, взятой с сайта [10] и изображенной на рис. 33, вы можете подсчитать номиналы и точность резисторов, входящих в комплект с Arduino Uno и показанных на рис. 5, 6 и 7. На рис. 34 показаны примеры резисторов и их номиналов:



Рис. 34 ❖ Примеры резисторов и их номиналов [10]

Однако есть и альтернативная картинка с правилами определения номиналов резисторов, и, скорее всего, мы столкнемся именно с такими резисторами, на которых не 4 цветные насечки, а 5, см. рис. 35. На рис. 5, 6 и 7 изображены

именно такие резисторы. Часто возникает вопрос: а с какой стороны считать номинал у резистора? Как определить, какая полоса – первая, а какая – последняя, относящаяся к точности? Во-первых, на самом деле все резисторы в свежкупленном наборе вдеты в бумажку, на которой, как правило, написан их номинал, как на рис. 5–7. Во-вторых, если внимательно присмотреться к резисторам, с одной стороны область утолщения плавно переходит в область с меньшим диаметром посередине, а с другой стороны резистора диаметр меняется резко. Если у вас такие резисторы, то начинать считать номинал надо с той стороны, где диаметр меняется резко с большего на меньший. Ну и, в конце концов, если у вас получилось что-то странное при подсчете номинала, чего не было заявлено в наборе, значит, вы начали не с той стороны.

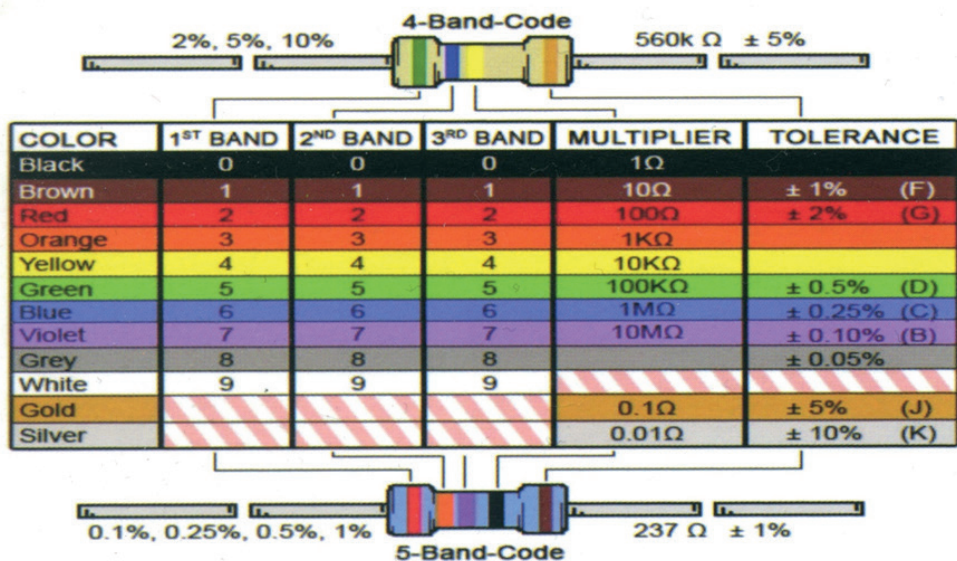


Рис. 35 ❖ Другие правила подсчета номинала резистора и его точности

Резисторы для схем бывают двух типов (здесь мы для простоты не говорим о специфических типах резисторов – потенциометрах, термисторах, варисторах, фоторезисторах и т. д., хотя некоторые из них встретятся нам в этой части книги): включенные последовательно в электрическую цепь (series resistors) по отношению к пинам платы, и параллельно, которые, в свою очередь, подразделяются на стягивающие и подтягивающие резисторы (pull-down и pull-up resistors). Значения последовательных резисторов обычно варьируются от 100 до 300 Ом, стягивающих и подтягивающих – от 1 до 10 кОм. Последовательные резисторы подключаются в электрическую цепь, чтобы защитить оборудование и, главное, – пины платы от больших значений тока: например, так подключается светодиод через резистор номиналом 220 Ом, чтобы светодиод не перегорел. Каждый пин платы обладает ограниченной способностью быть источником (высокое значение, логическая 1) или приемником (низкое значение, логический 0) электрического тока через электрическую схему, подключенную к нему. Периферия, которая в схеме потребляет большие значения



тока – больше, чем может позволить себе пин платы, даже если это происходит очень короткое время, может повредить пин на плате – именно поэтому в схеме последовательно включается ограничивающий ток резистор, например как показано на рис. 36 в схеме со светодиодом [11], где OUT – это пины платы, а Vcc – источник питания:

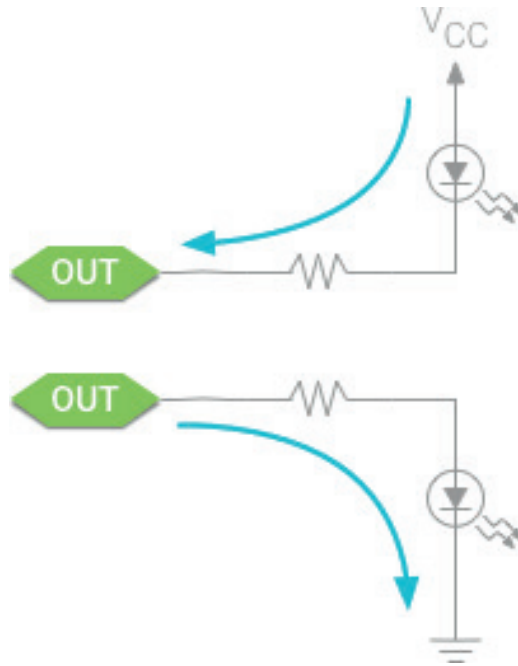
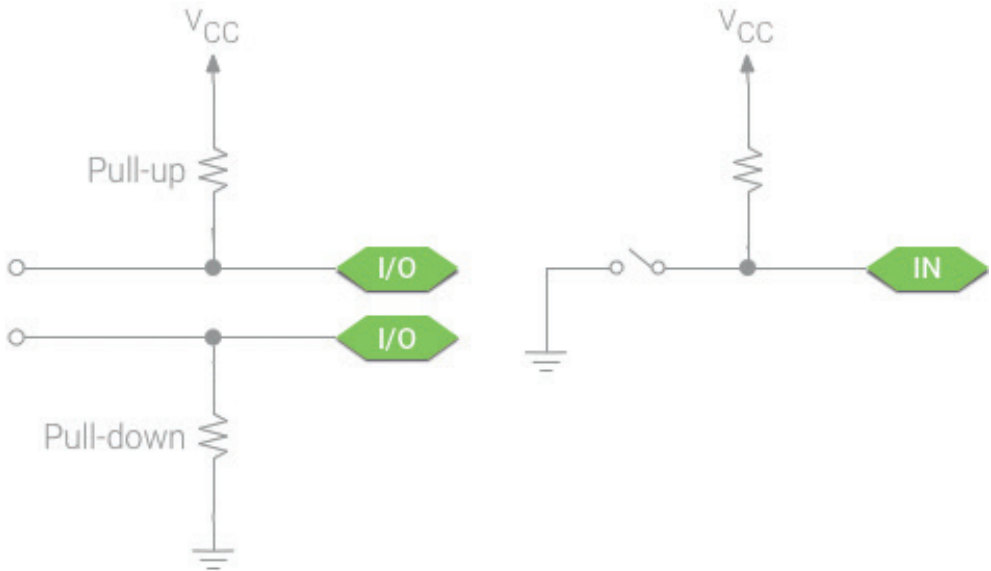


Рис. 36 ❖ Пример подключения последовательных резисторов [11]

Стягивающие и подтягивающие резисторы подключаются в схему параллельно по отношению к пинам и используются для того, чтобы, соответственно, «стягивать» значение напряжения на пине к низкому (обеспечивать стабильный сигнал логического 0) или «подтягивать» значение напряжения на пине к высокому (стабильная логическая 1). Делается это потому, что цифровые входы, не подключенные ни к какой нагрузке, являются «плавающими» (см. рис. 37, левую часть – представьте ее без включенных туда резисторов; I/O – пины платы): они подвержены различного рода помехам и искажениям, появляющимся из-за электромагнитных возмущений, которые влияют на значения, читаемые с пинов платы приложениями, и могут способствовать непредсказуемым изменениям этих значений. Стягивающие и подтягивающие резисторы заставляют пины показывать правильные значения 0 или 1, даже если к ним ничего не подключено [11]. В правой части рис. 37 изображен случай с переключателем: если в схеме не будет подтягивающего резистора, значение при открытом переключателе на входном пине IN платы, читающем значения, будет плавающим; если в схеме есть такой резистор, изображенный на рисунке, значение на пине IN будет точно соответствовать логической единице.



**Рис. 37** ❖ Пример подключения стягивающего (pull-down) и подтягивающего (pull-up) резисторов [11]

Еще один момент, о котором следует упомянуть, – это тот факт, что при подключении различных модулей и сенсоров надо также обращать внимание не только на схему, но и на надписи рядом с выходами (пинами) этих сенсоров. Сенсоры в наборах могут отличаться, хоть эта вероятность и мала, поэтому всегда проверяйте наличие информации/меток рядом с пинами устройств, используемых в практических заданиях, – это первично, а схема вторична, так как исходить надо всегда из реальности. Простой пример – задание 15 из этой части книги, где к конкретным пинам платы подключаются конкретные выходы модуля часов реального времени, обозначения которых можно увидеть на самом модуле – см. рис. 21. Некоторые обозначения: – (минус), GND, G – земля; + (плюс), VCC, VIN, +5V, 3.3V – питание; CLK, SCK – clock (время, частота), DAT, SDA – date/data (дата, данные), RST – reset (сброс настроек), R, G, B – цвета на трехцветном светодиоде; A0 (аналоговый), D0 (цифровой), SIG, S, VRx, VRy, SW – пин для передачи сигнала (данных).

Теперь, получив базовые знания об основных элементах, использующихся в практических экспериментах, и о технике безопасности, можно приступить к выполнению заданий.

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1. HELLO, WORLD!

В этом практическом занятии нам **понадобятся**:

- плата Arduino Uno;
- USB-кабель (Am-Bm).

Первое, что нужно сделать, – это выбрать, где вы будете работать: в среде Arduino IDE или с помощью online-системы Arduino Create (Arduino Web Editor) через веб-браузер. В первом случае необходимо скачать Arduino IDE ([5] -> Windows



Installer) и установить ее, включая установку драйверов для COM и USB-портов. Во втором случае нужно зарегистрироваться на сайте ([6] -> sign in/create one).

Далее необходимо пройти Setup Guide по следующему адресу: [12]. Это нужно для того, чтобы установить драйверы для платы, если они правильно не установились или нет прав администратора на компьютере, а также для того, чтобы правильно настроить среду и выбрать плату Arduino Uno и COM-порт.

После этого можно приступать к занятию. Подсоедините плату Arduino Uno к USB-порту компьютера (если вы еще этого не сделали). Если драйверы установлены правильно, плата должна определиться, ее название появится в панели уведомлений операционной системы. На плате есть встроенный мини-светодиод (miniLED), подключенный к 13-му цифровому порту. В этом занятии мы напишем код, который будет ожидать ввода через консоль буквы R, при ее вводе заставляя miniLED 13 загораться на полсекунды и гаснуть на полсекунды и писать в консоль фразу «Hello, World!». Так как miniLED является встроенным, никаких дополнительных схем создавать не надо. Для того чтобы открыть консоль, надо выбрать в пункте меню **Инструменты -> Монитор порта** или нажать комбинацию **Ctrl+Shift+M** (Arduino IDE) или выбрать пункт меню **Монитор порта** слева (Arduino Web Editor). Когда все готово, можно скопировать код ниже в среду и загрузить программу на плату с помощью кнопки ->.

В коде используется команда `Serial.begin(9600)`, означающая, что скорость/частота обмена данными платы с компьютером по USB-соединению составляет 9600 bps (bits per second, битов в секунду). В консоли можно увидеть, что есть и другие частоты, но для выполнения задания в консоли должна быть выставлена такая же частота (справа внизу).

Схема представлена на следующем рисунке:

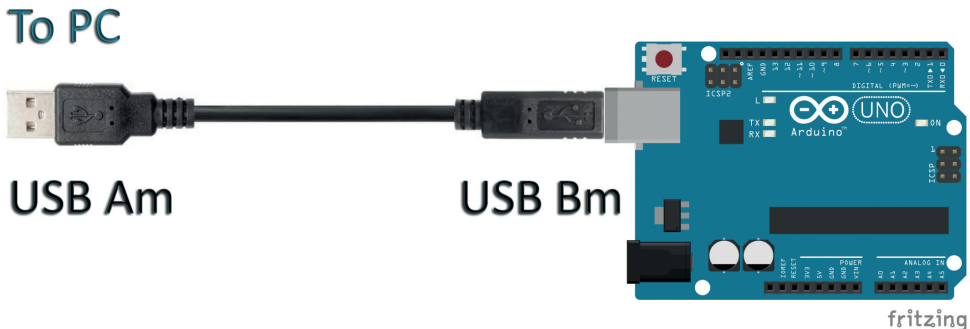


Рис. 38 ❖ Схема подключения для практического занятия 1

Код программы:

```
int val;
int ledpin = 13;
void setup ()
{
  Serial.begin (9600);
  pinMode (ledpin, OUTPUT);
}
void loop ()
```

```

{
  val = Serial.read ();
  if (val == 'R') {
    digitalWrite (ledpin, HIGH);
    delay (500);
    digitalWrite (ledpin, LOW);
    delay (500);
    Serial.println ("Hello World!");
  }
}

```

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2. ЭКСПЕРИМЕНТ С МИГАЮЩИМ СВЕТОДИОДОМ

В этом практическом занятии нам **понадобятся**:

- плата Arduino Uno;
- USB-кабель (Am-Bm).

В этом занятии мы все еще работаем со встроенным мини-светодиодом на плате (miniLED), подключенным к 13-му цифровому порту. Напишем код, который будет заставлять miniLED 13 загораться на секунду и гаснуть на секунду. Так как miniLED является встроенным, никаких дополнительных схем создавать не надо.

Схема представлена на следующем рисунке:

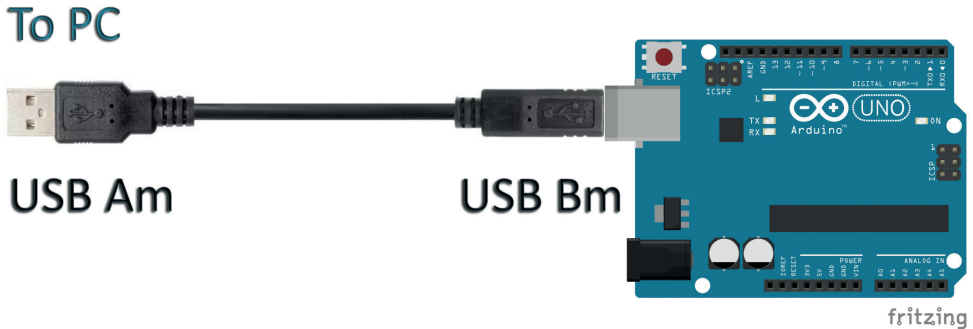


Рис. 39 ❖ Схема подключения для практического занятия 2

Код программы:

```

int ledpin = 13;
void setup ()
{
  pinMode (ledpin, OUTPUT);
}
void loop ()
{
  digitalWrite (ledpin, HIGH);
  delay (1000);
  digitalWrite (ledpin, LOW) ;
  delay (1000);
}

```

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3. ЭКСПЕРИМЕНТ С КОНТРОЛИРУЕМОЙ ПОТЕНЦИОМЕТРОМ ЯРКОСТЬЮ СВЕЧЕНИЯ СВЕТОДИОДА ЧЕРЕЗ ПОРТ PWM

В этом практическом занятии нам **понадобятся**:

- плата Arduino Uno;
- USB-кабель (Am-Bm);
- потенциометр;
- светодиод;
- резистор на 220 Ом;
- макетная плата;
- соединительные провода.

В этом занятии мы познакомимся с потенциометром и портом PWM. PWM (Pulse Width Modulation) – широтно-импульсная модуляция, или процесс управления мощностью, подводимой к нагрузке, путем изменения отношения периода импульса к длительности импульса при неизменной частоте. На плате Arduino Uno можно подавать значения от 0 до 255 на PWM-пин, что заставит плату выдавать PWM-сигнал в определенные моменты времени, соответствующие поданному входному значению. Другими словами, в терминах напряжения, подавая разные значения от 0 до 255 на PWM-пин, мы заставляем плату менять напряжение от 0 до 5 В. В коде используется функция `analogWrite(pin, value)`, с помощью которой можно менять напряжение, подаваемое на PWM-пин, задавая `value` от 0 до 255. Эта функция используется для регулировки скорости вращения мотора, яркости светодиода и т. д.

Arduino Uno располагает несколькими PWM-пинами, обозначенными символом ~ рядом с номером пина: 3, 5, 6, 9, 10, 11 (см. рис. 1). С помощью потенциометра, подключенного к одному из аналоговых портов платы (A0-A6), и светодиода, катод (короткий пин) которого подключен через резистор на 220 Ом к земле (всегда! к минусу) и анод (длинный пин) которого подключен к одному из цифровых пинов (0–13) платы (всегда! к плюсу), мы будем регулировать значение переменной `val` (крутя ручку потенциометра) и, соответственно, напряжение, которое выдает плата из этого цифрового пина.

**Схема** представлена на рис. 40.

**Код программы:**

```
int potpin = 0;
int ledpin = 11;
int val = 0;
void setup ()
{
  pinMode (ledpin, OUTPUT);
  Serial.begin (9600);
}
void loop ()
{
  val = analogRead (potpin);
  Serial.println (val);
  analogWrite (ledpin, val / 4);
  delay (10);
}
```

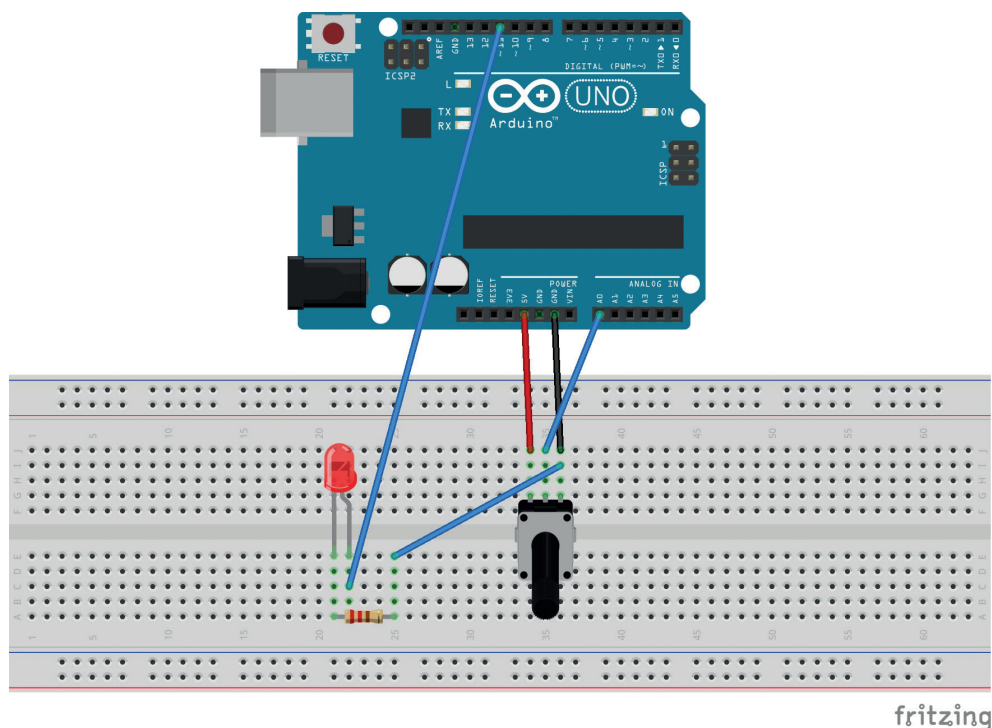


Рис. 40 ❖ Схема подключения с макетной платой для практического занятия 3

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4. СВЕТОФОРНЫЙ ЭКСПЕРИМЕНТ

В этом практическом занятии нам **понадобятся**:

- плата Arduino Uno;
- USB-кабель (Am-Bm);
- светодиоды – 3 шт. разных цветов;
- резистор на 220 Ом – 3 шт.;
- макетная плата;
- соединительные провода.

В этом занятии мы эмулируем работу светофора. В наборе не оказалось зеленого светодиода, поэтому будем использовать красный (цифровой пин 10 платы), желтый (пин 7) и синий (пин 4). Возможно, вам повезет больше, и у вас будет зеленый светодиод. Напишем код, который будет заставляя три светодиода переключаться примерно с той задержкой, которая принята в реальных светофорах (от красного к зеленому).

Схема представлена на рис. 41.

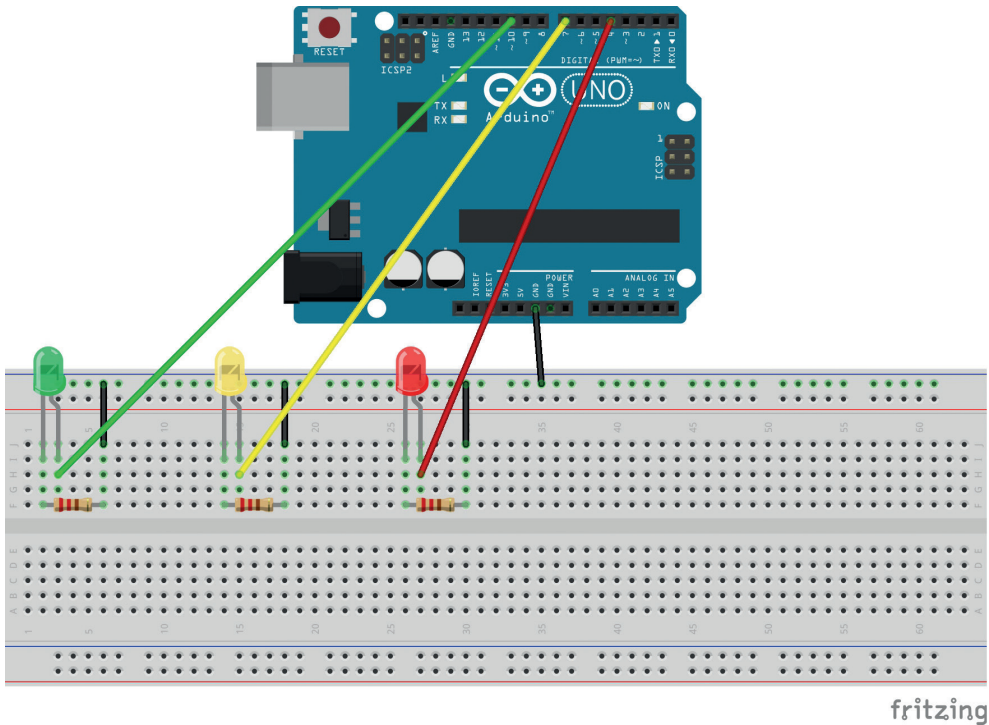


Рис. 41 ❖ Схема подключения с макетной платой для практического занятия 4

**Код программы:**

```
int redled = 10;
int yellowled = 7;
int greenled = 4;
void setup ()
{
  pinMode (redled, OUTPUT);
  pinMode (yellowled, OUTPUT);
  pinMode (greenled, OUTPUT);
}
void loop ()
{
  digitalWrite (redled, HIGH);
  delay (1000);
  digitalWrite (redled, LOW);
  digitalWrite (yellowled, HIGH);
  delay (200);
  digitalWrite (yellowled, LOW);
  digitalWrite (greenled, HIGH);
  delay (1000);
  digitalWrite (greenled, LOW);
}
```