ОСНОВНОЙ НАБОР ПОСТАВКИ СТАРТЕР ДЛЯ ООН

V1.0.19.7.24

Предисловие

О нас

Основанная в 2011 году, «Elegoo Inc.» – процветающая технологическая компания, специализирующаяся на исследовании, разработке, производстве и маркетинге открытого аппаратного обеспечения. Расположенная в Шеньчжэне – “Кремниевой долине” Китая – компания насчитывает более 150+ сотрудников и располагает заводом площадью в 1000+ квадратных метров.

Продуктовый ряд представляет собой вариацию, начиная с проводов DuPont и плат UNO R3 до полноценных стартовых наборов, рассчитанных на клиентов с любым уровнем предыдущего опыта и знаний об Arduino. Помимо этого, среди товаров также присутствуют аксессуаров для Raspberry Pi, такие как экраны 2.8’’ TFT touch и STM32 микроконтроллеры. В будущем компания планирует уделить больше внимания и, возможно, инвестировать в рынок 3D печати. Вся продукция «Elegoo Inc.» соответствует международным стандартам качества и высоко ценится покупателями со всего мира.

Официальный сайт: http://www.elegoo.com

Интернет-магазин на Amazon доступен для следующих стран:

US Amazon storefront: http://www.amazon.com/shops/A2WWHQ25ENKVJ1

CA Amazon storefront: http://www.amazon.ca/shops/A2WWHQ25ENKVJ1

UK Amazon storefront: http://www.amazon.co.uk/shops/AZF7WYXU5ZANW

DE Amazon storefront: http://www.amazon.de/shops/AZF7WYXU5ZANW

FR Amazon storefront: http://www.amazon.fr/shops/AZF7WYXU5ZANW

ES Amazon storefront: http://www.amazon.es/shops/AZF7WYXU5ZANW

IT Amazon storefront: http://www.amazon.it/shops/AZF7WYXU5ZANW

Наши уроки

Эти уроки рассчитаны на новичков. С их помощью вы сможете научиться работать с контроллером Arduino, датчиками и другими компонентами. Если вы хотите углубиться в изучение Arduino, мы рекомендуем прочитать «Arduino Cookbook» Майкла Марголиса.

Некоторые части кода в этих уроках редактировались Саймоном Монком. Саймон Монк – автор ряда книг об отрытом аппаратном обеспечении, которые доступны на Amazon: «Программируем Arduino», «30 проектов Arduino для злого гения (30 Arduino Projects for the Evil Genius)» и «Программируем Raspberry Pi (Programming the Raspberry Pi)».

Обслуживание клиентов

Как непрерывно и быстро растущая технологическая компания, наше стремление – оправдать все ваши ожидания, предоставляя продукцию и куслуги наивысшего качества. Вы можете связаться с нами, написав на service@elegoo.com или EUservice@elegoo.com.

Мы с нетерпением ждем ваши отзывы, комментарии или предложения.

Наши опытные инженеры ответят на любые ваши вопросы или помогут решить возникшие проблемы в течение 12 часов (в течение 24 часов, в праздничное время).

Packing list

www.elegoo.com

RGB LED
2PCS

Photoresistor
(photocell)
1PC

Resistor
120PCS

UNO R3
with USB
1PC

Tilt Ball Switch
1PC

LED
30PCS

Button(Small)
5PCS

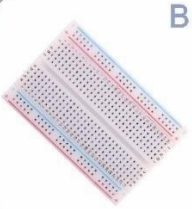
Active Buzzer
1PC

74HC595 IC
1PC

F-M Dupont Wire
5PCS

Breadboard
Jumper Wire
65PCS

Breadboard
1PC



Contact us : service@elegoo.com

Содержание

Урок 0: Установка среды IDE

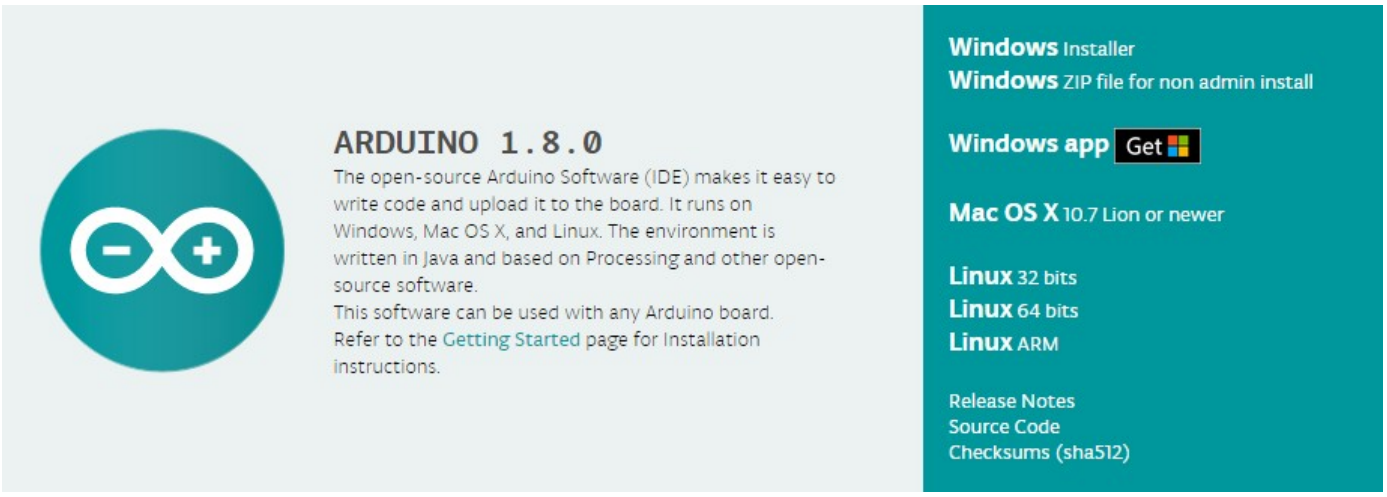
Вступление

Интегрированная среда разработки (Integrated Development Environment, IDE) Arduino – это программная сторона платформы Arduino.

В этом уроке рассказывается, как настроить ваш компьютер для использования Arduino, и как подготовиться к последующим урокам.

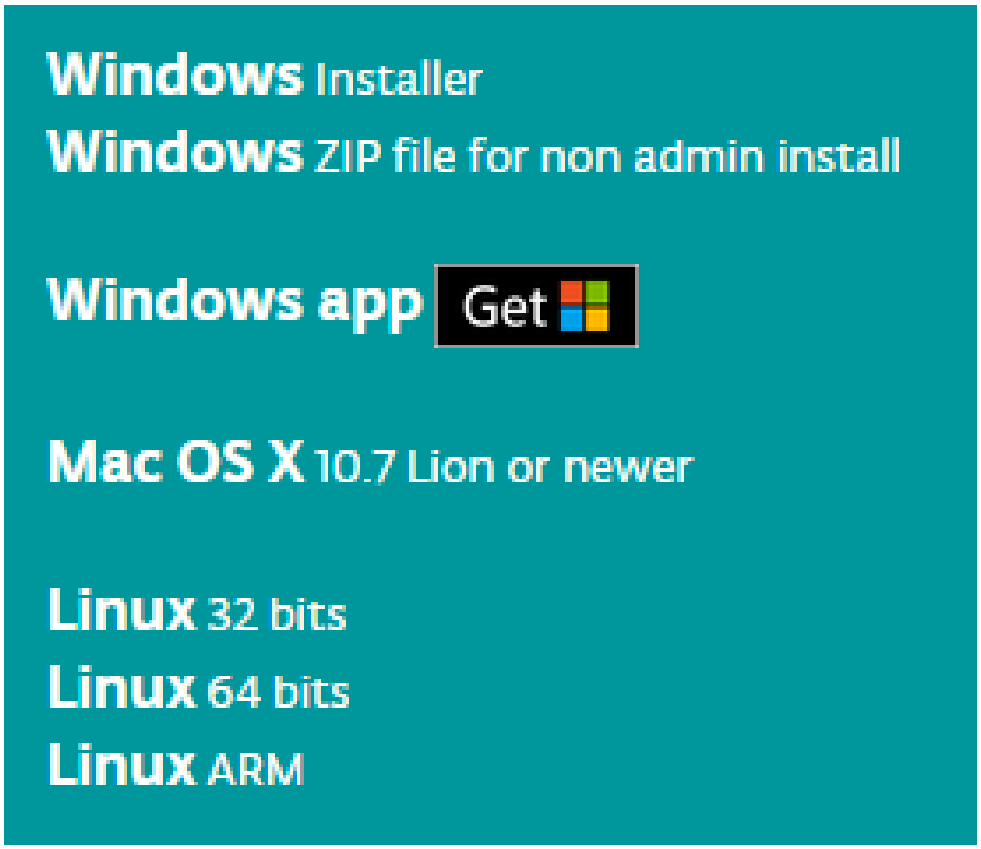
Программное обеспечение Arduino, которое вы будете использовать для программирование вашего контроллера Arduino, доступно для операционных систем Windows, Mac и Linux. Процесс установки отличается для разных операционных систем и, к сожалению, предполагает некоторую «ручную» работу.

ШАГ 1: Перейдите на https://www.arduino.cc/en/Main/Software и найдите страницу, показанную ниже.



Версия программы на этом сайте обычно является самой последней. К тому же, текущая версия может быть новее чем версия, представленная на рисунке.

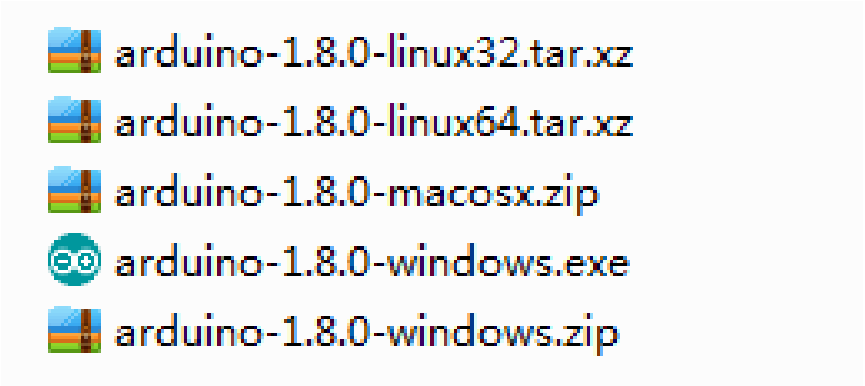
ШАГ 2: Загрузите среду разработки, совместимую с операционной системой вашего компьютера. Здесь как пример выбрана ОС Windows.



Выберите "*Windows Installer" (Установщик Windows)*.

Нажмите *"JUST DOWNLOAD" (ТОЛЬКО ЗАГРУЗИТЬ)*.

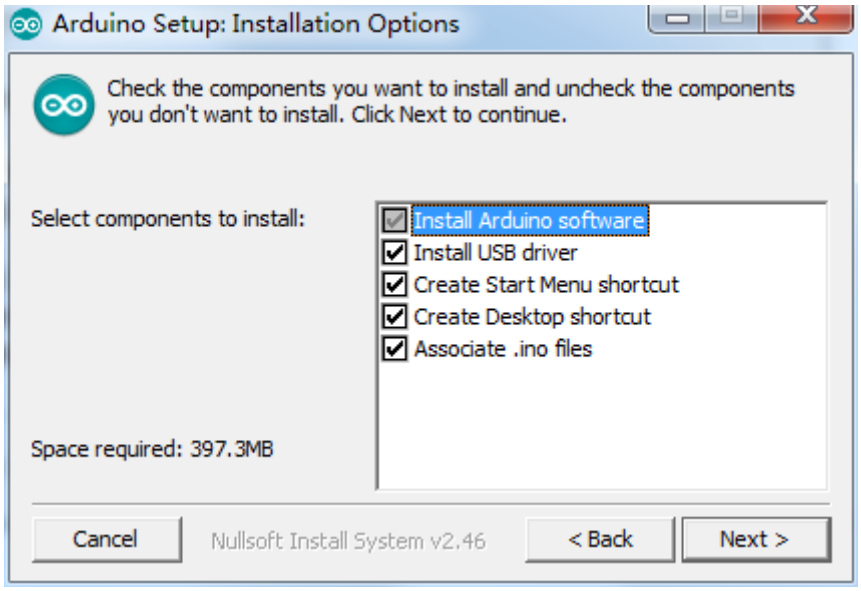
Версия программы 1.8.0 доступна вместе с другими материалами, которые мы предоставляем, и, на момент выпуска этого курса, является самой последней.



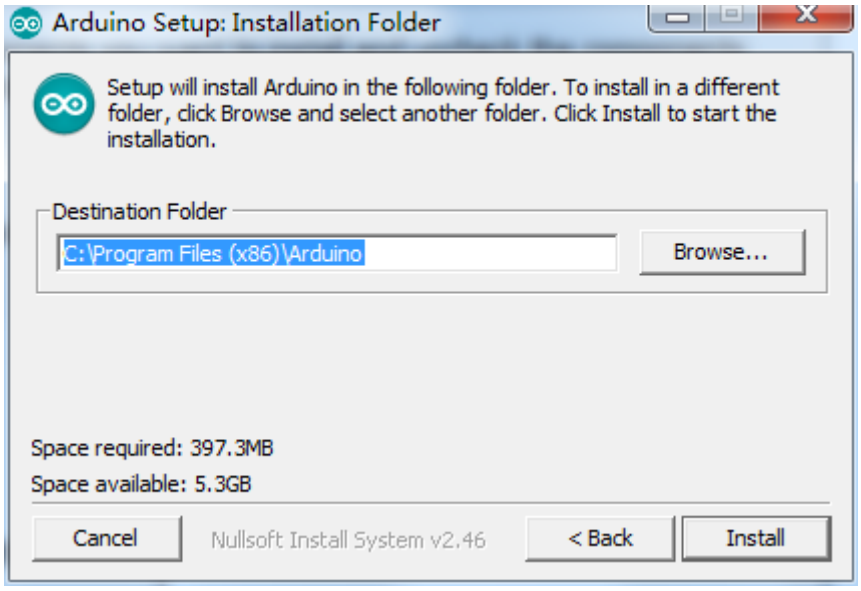
Установка Arduino (Windows)

Установить Arduino используя exe. Установочный пакет.

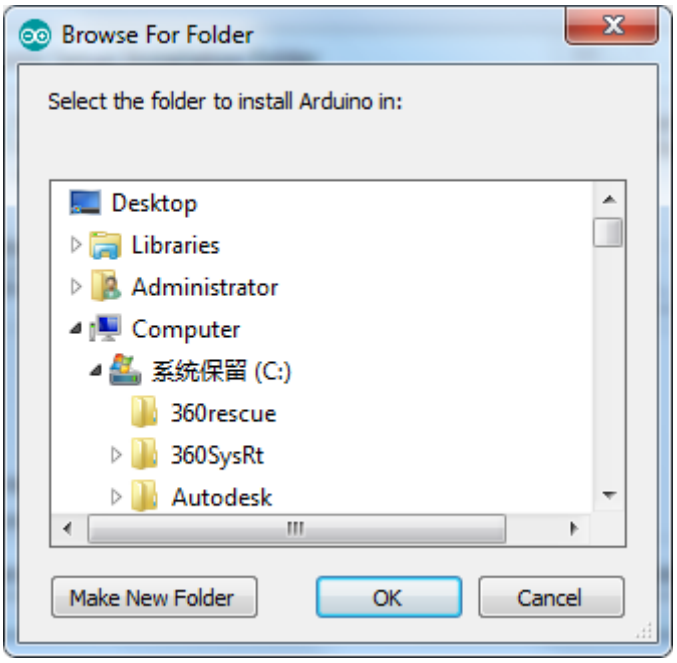
Выберите “*I Agree” (Я согласен)*, чтобы перейти к следующему интерфейсу



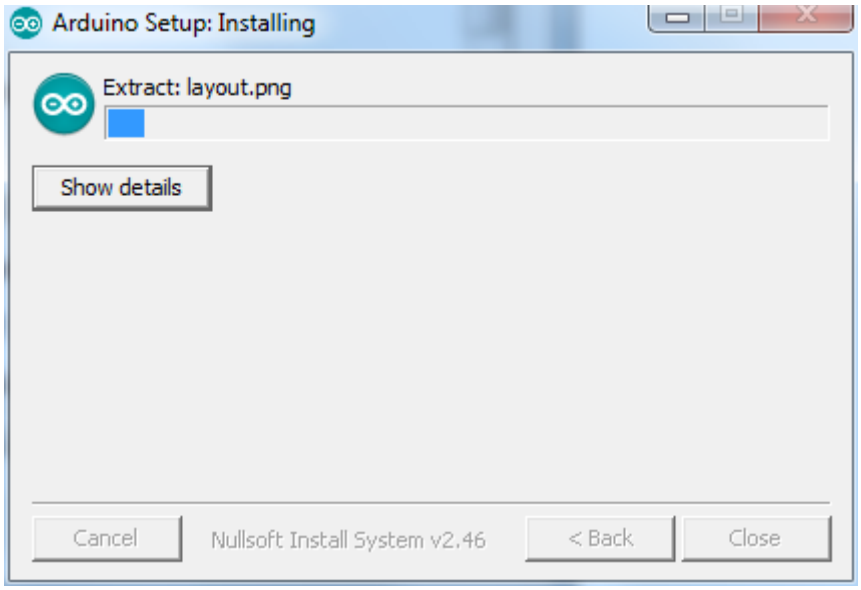
Нажмите *“Next” (Далее)*



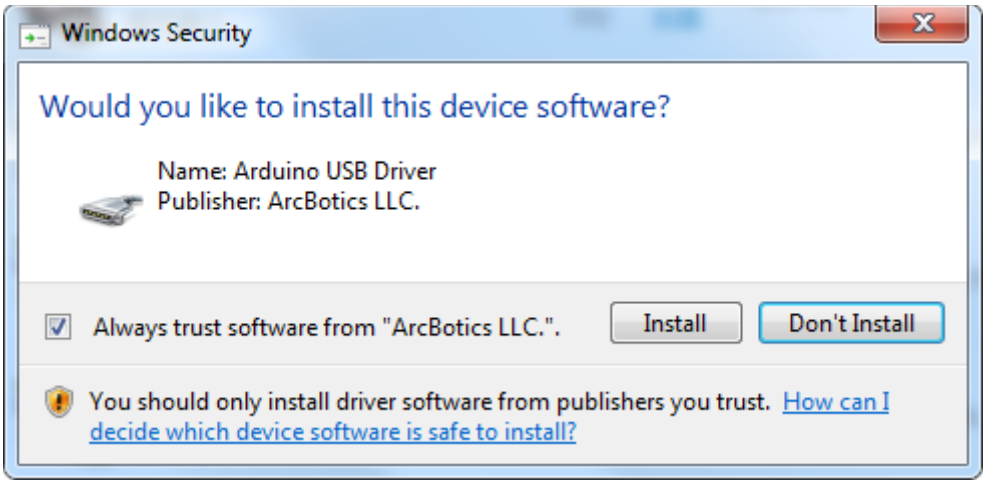
Вы можете выбрать “Browse…” для выбора пути установки или напрямую ввести необходимую вам директорию.



Выберите “*Install” (Установить), чтобы начать установку.*

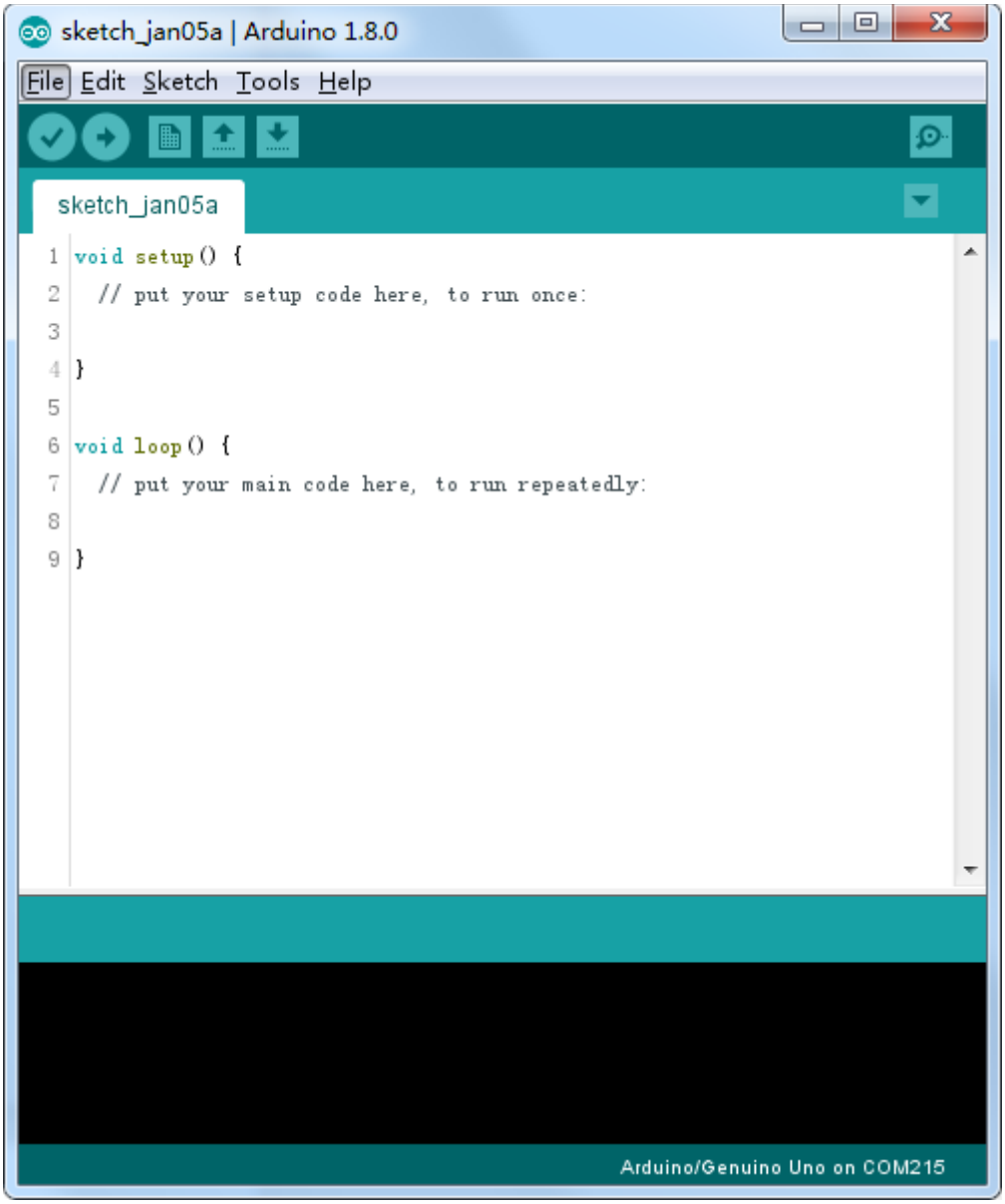


Наконец, когда вы увидите следующее окно, нажмите “*Install” (Установить),* чтобы завершить установку.



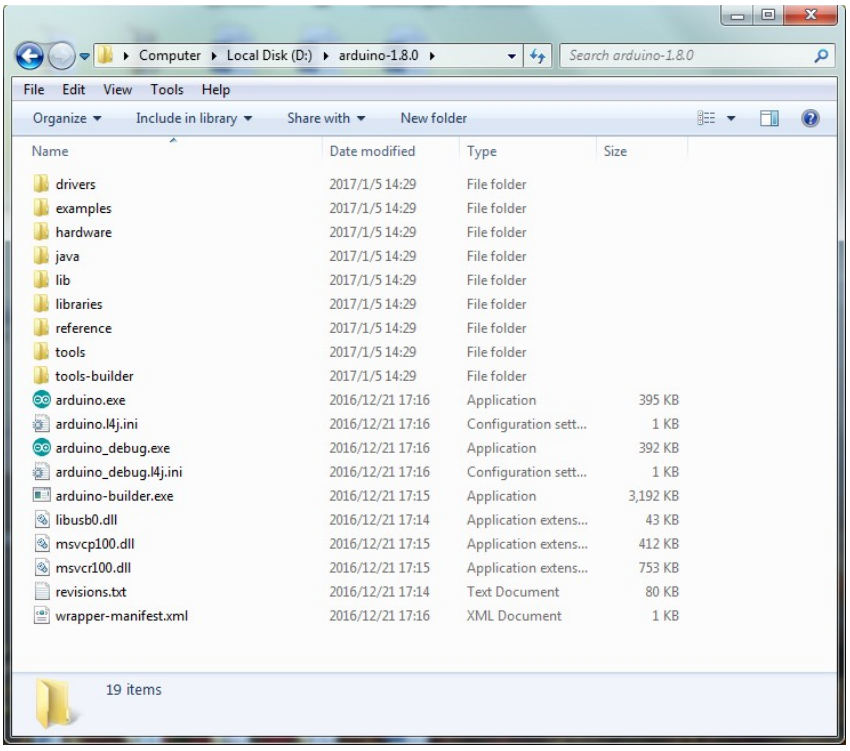
Вы увидите на рабочем столе следующий значок

Кликните дважды, чтобы выбрать желаемую среду разработки

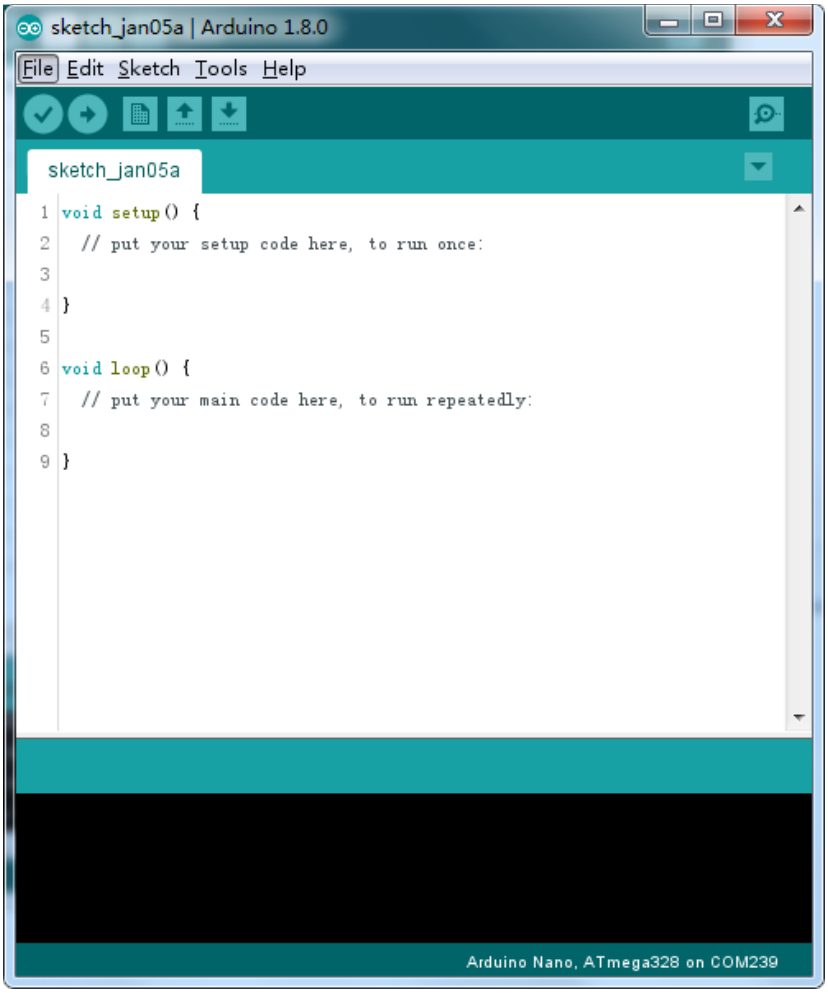


Вы можете остановить свой выбор на методе установки программы с использованием установочного пакета и пропустить материал, приведенный ниже, сразу перейдя к следующему уроку. Но если вам интересно узнать о других методах установки, пожалуйста, продолжайте читать.

Распакуйте загруженный zip файл, двойным нажатием открыть программу и ввести желаемую среду разработки


Computer ▸ Local Disk (D:) ▸ arduino-1.8.0 ▸
Search arduino-1.8.0
File Edit View Tools Help
Organize
Include in library
Share with
New folder
Name
Date modified
Type
Size
drivers 2017/1/5 14:29 File folder
examples 2017/1/5 14:29 File folder
hardware 2017/1/5 14:29 File folder
java 2017/1/5 14:29 File folder
lib 2017/1/5 14:29 File folder
libraries 2017/1/5 14:29 File folder
reference 2017/1/5 14:29 File folder
tools 2017/1/5 14:29 File folder
tools-builder 2017/1/5 14:29 File folder
arduino.exe 2016/12/21 17:16 Application 395 KB
arduino.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino_debug.exe 2016/12/21 17:16 Application 392 KB
arduino_debug.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino-builder.exe 2016/12/21 17:15 Application 3,192 KB
libusb0.dll 2016/12/21 17:14 Application extens... 43 KB
msvcp100.dll 2016/12/21 17:15 Application extens... 412 KB
msvcr100.dll 2016/12/21 17:15 Application extens... 753 KB
revisions.txt 2016/12/21 17:14 Text Document 80 KB
wrapper-manifest.xml 2016/12/21 17:16 XML Document 1 KB
19 items


arduino.exe


sketch_jan05a | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05a
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Arduino Nano, ATmega328 on COM239

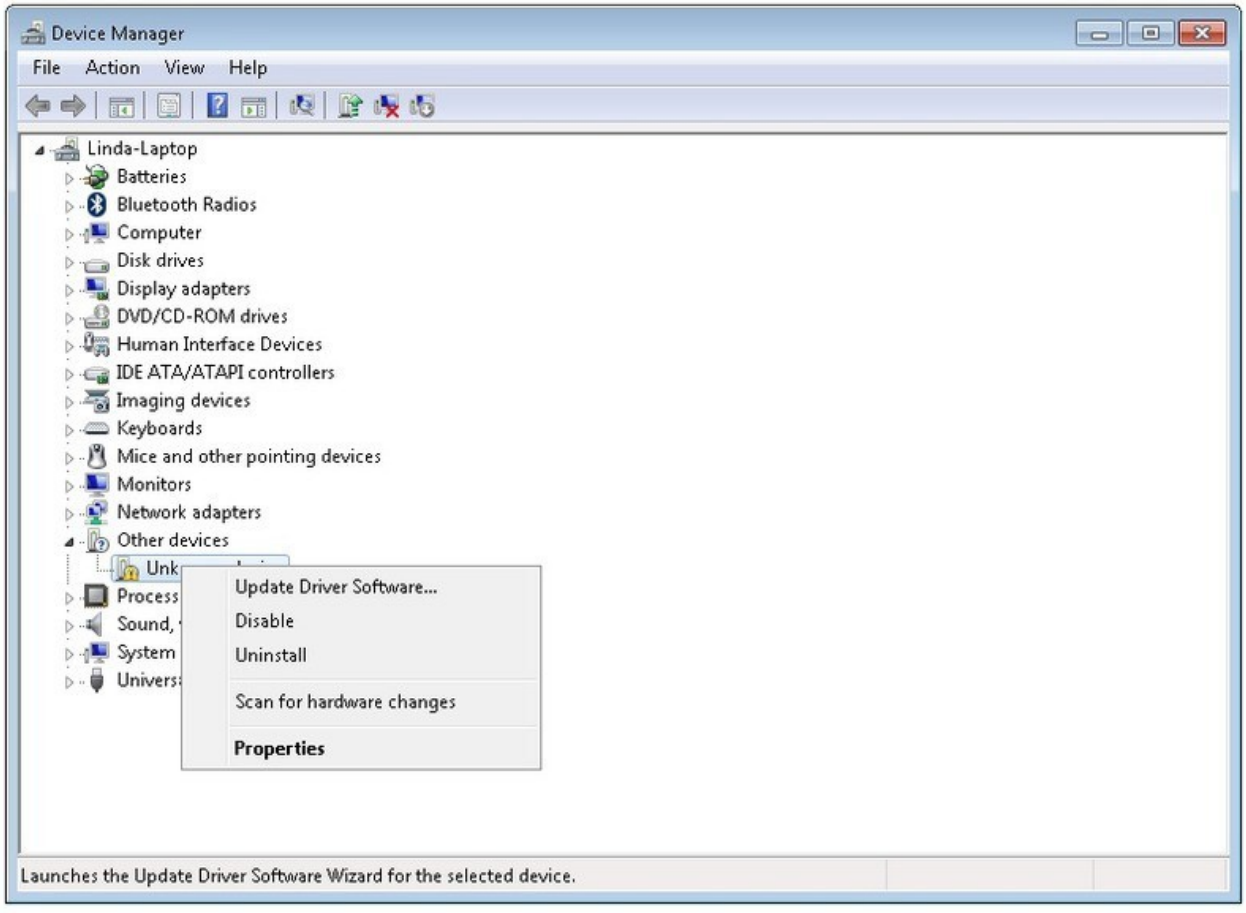
Данный метод установки требует отдельной установки драйвера.

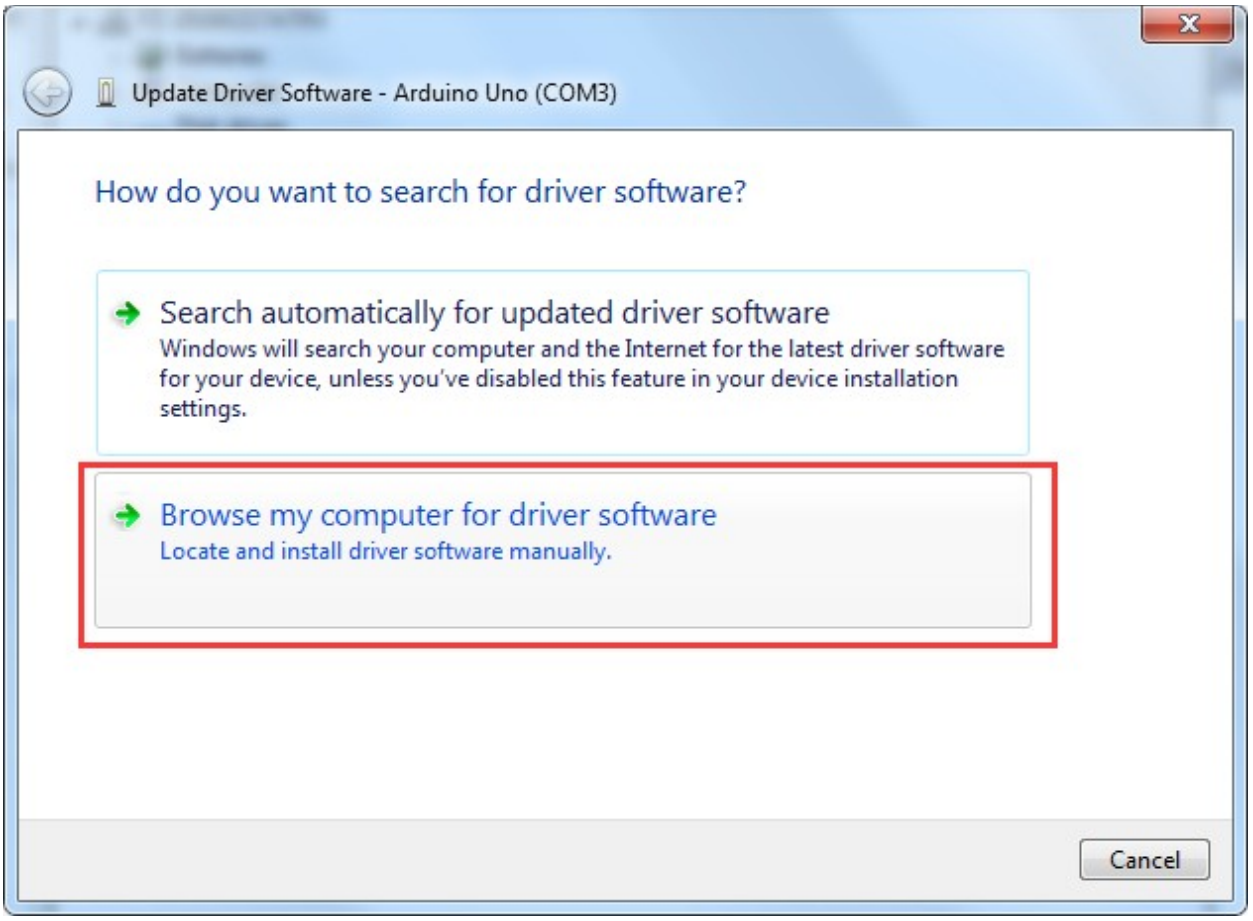
Папка Arduino содержит в себе как саму программу Arduino, так и драйверы, позволяющие подключение Arduino к вашему компьютеру через кабель USB. Перед запуском программного обеспечения Arduino, вы должны установить драйверы USB.

С помощью USB кабеля подключите плату Arduino к вашему компьютеру. Должен загореться светодиод, показывающий, что на плату поступает питание, и вы должны получить сообщение от Windows “Найдено новое оборудование“ ('Found New Hardware'*)*. Игнорируйте сообщение и отклоняйте любые попытки Windows произвести автоматическую установку драйверов.

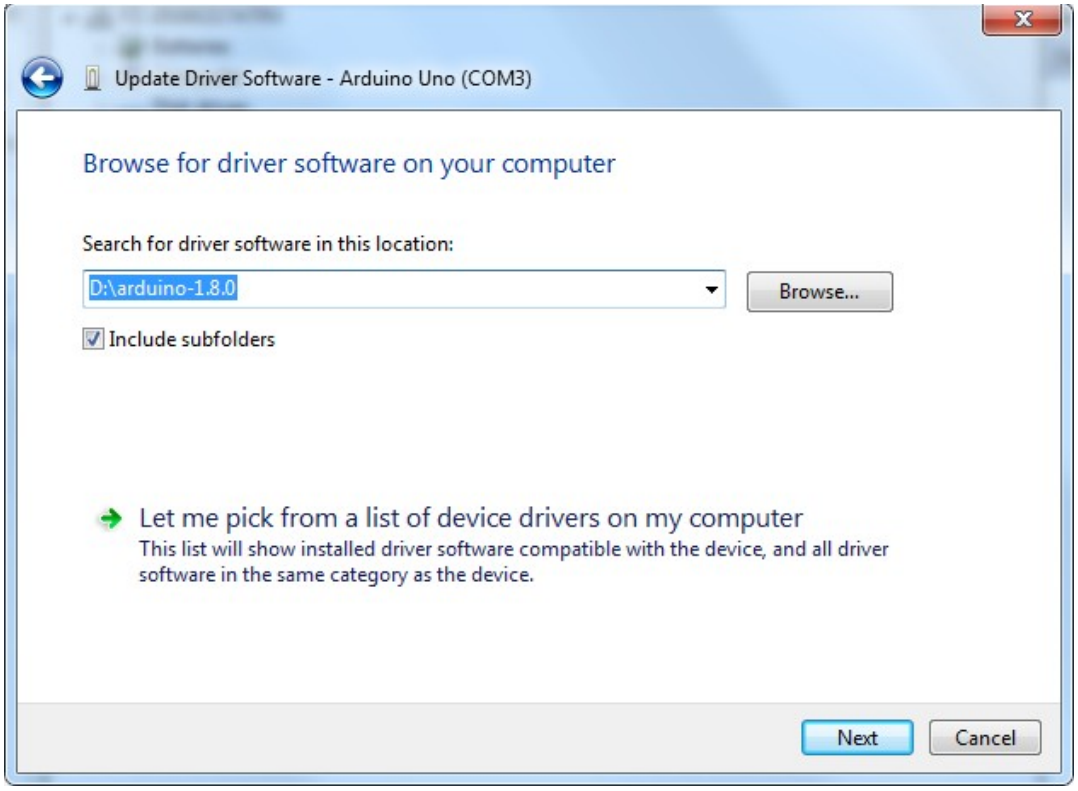
Самым проверенным способом установки USB драйверов является использование Диспетчера Устройств. Чтобы открыть Диспетчер Устройств в Windows 7, вы сначала должны открыть Панель Управления, затем выбрать опцию просмотра Значков, и найти Диспетчер Устройств в списке.

В “Другие Устройства” (‘Other Devices’), вы должны увидеть значок для “неизвестного устройства” с небольшим желтым предупреждающим треугольником. Это ваш Arduino.

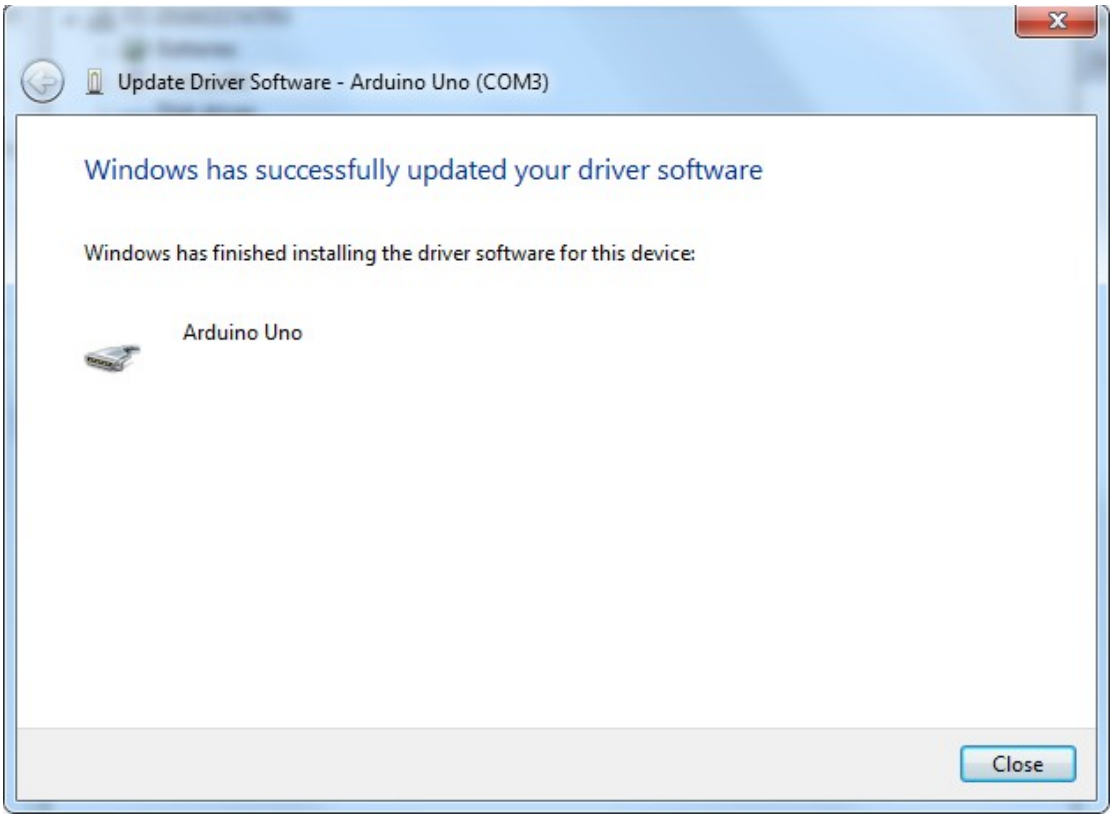




Щелкните правой кнопкой на устройство и выберите “Обновить Драйвер...”. Затем последует выбор либо “Автоматический поиск обновленного драйвера”, либо “Выполнить поиск драйверов на этом компьютере“. Выберите последнее и вручную укажите место размещения драйвера X\arduino1.8.0\drivers.



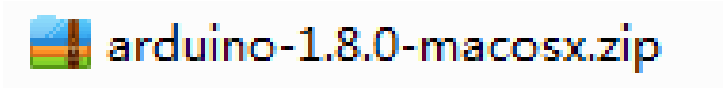
Выберите “Далее”, после чего может выскочить предупреждение системы безопасности. В таком случае, разрешите установку программного обеспечения. Как только установка будет завершена, вы получите сообщение с подтверждением.



Пользователи Windows могут пропустить указания для установки ПО на компьютеры с операционными системами Mac и Linux и сразу перейти к Уроку 1. Пользователи Mac и Linux могут продолжить чтение.

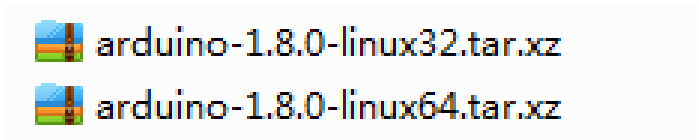
Установка Arduino (Mac OS X)

Загрузите и распакуйте zip файл. Кликните дважды на Arduino.app для ввода Arduino IDE. Система попросит вас установить библиотеку среды выполнения Java runtime library, если она еще не установлена на вашем компьютере. После окончания установки, вы можете запускать Arduino IDE.

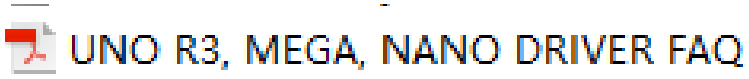


Установка Arduino (Linux)

Вам нужно будет использовать команду ‘make install’. Если вы пользуетесь системой Ubuntu, то рекомендуется устанавливать Arduino IDE из центра приложений Ubuntu.



ПОДСКАЗКА: Если вы столкнулись с проблемой в процессе установки драйверов, пожалуйста, обратитесь к UNO R3, MEGA, NANO DRIVER FAQ.



Урок 1: Установка библиотек и работа с серийным монитором

Установка дополнительных библиотек Arduino

Как только вы освоитесь с программным обеспечением Arduino и в совершенстве овладеете стандартными функциями, вам наверняка захочется расширить возможности вашего Arduino с помощью дополнительных библиотек.

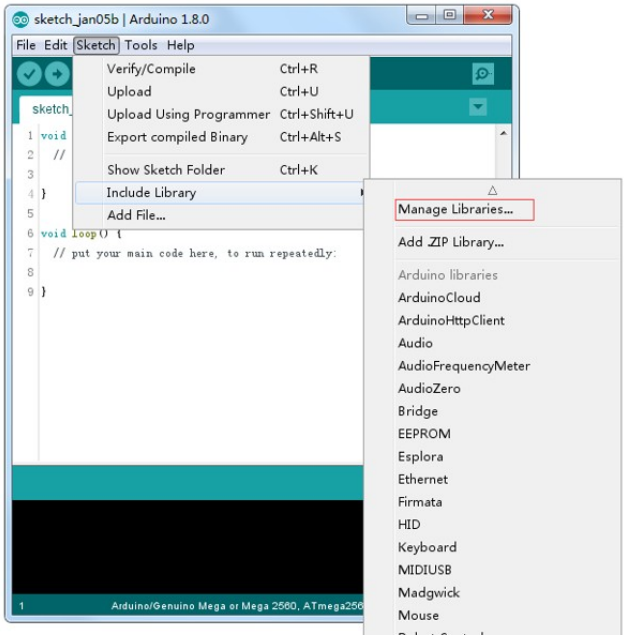
Что такое библиотека?

Библиотека - это набор функций, предназначенных для того, чтобы максимально упростить работу с различными датчиками, ЖК-экранами, модулями и т.д. Например, встроенная библиотека LiquidCrystal позволяет легко взаимодействовать с символьными LCD-экранами. Существуют сотни дополнительных библиотек, которые можно скачать в Интернете. Стандартные библиотеки Arduino и ряд наиболее часто используемых дополнительных библиотек перечислены в справке. Но перед тем, как использовать дополнительные библиотеки, необходимо сперва установить их.

Как установить библиотеку

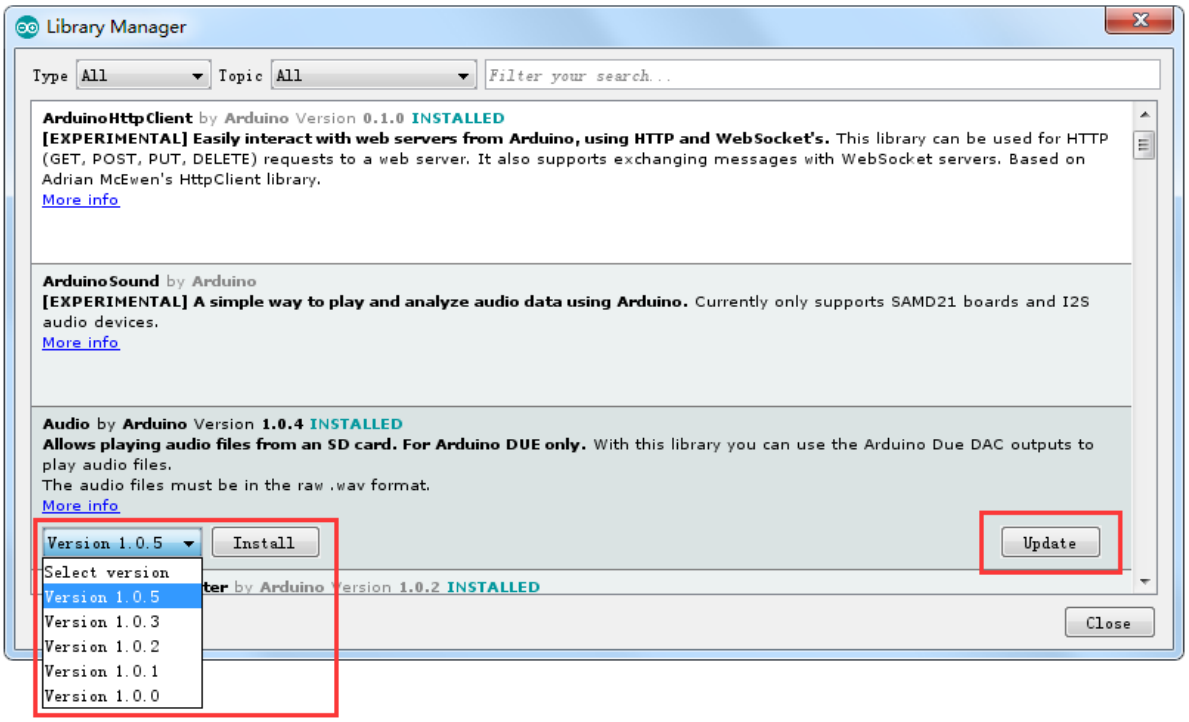
Используя Library Manager

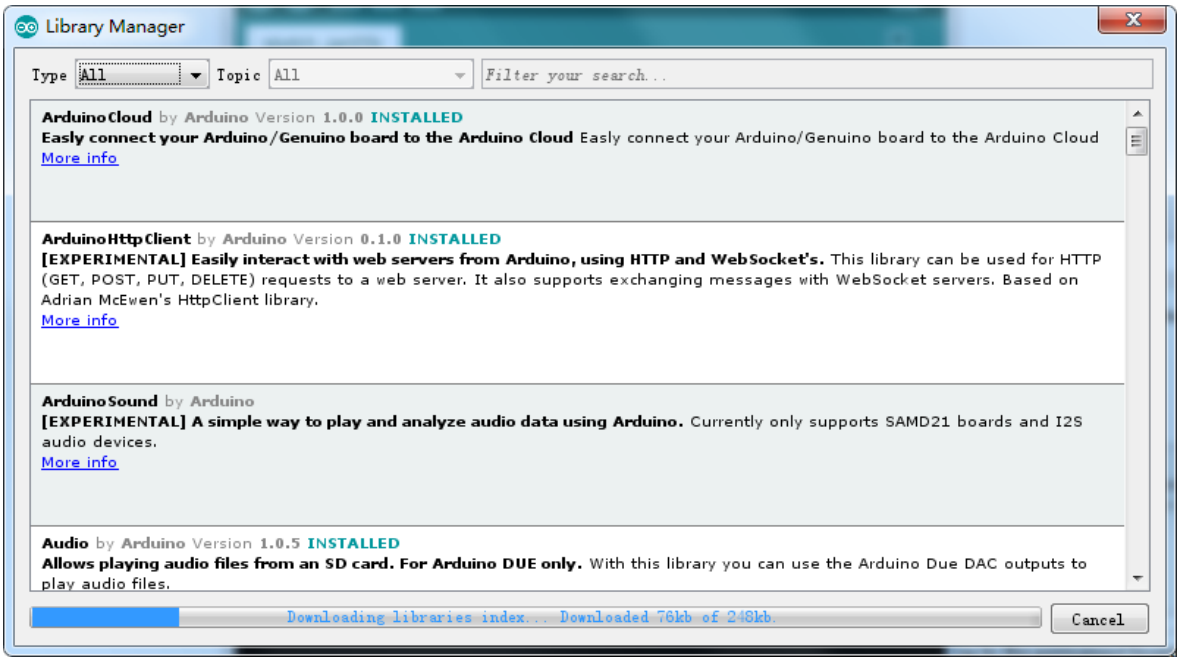
Для установки новой библиотеки в среде разработки Arduino вы можете использовать Library Manager (доступен для версий IDE 1.8.0 и новее). Откройте среду разработки и выберите *Sketch*, а затем *Include Library (Добавить библиотеку)* > *Manage Libraries (Управлять библиотеками)*.



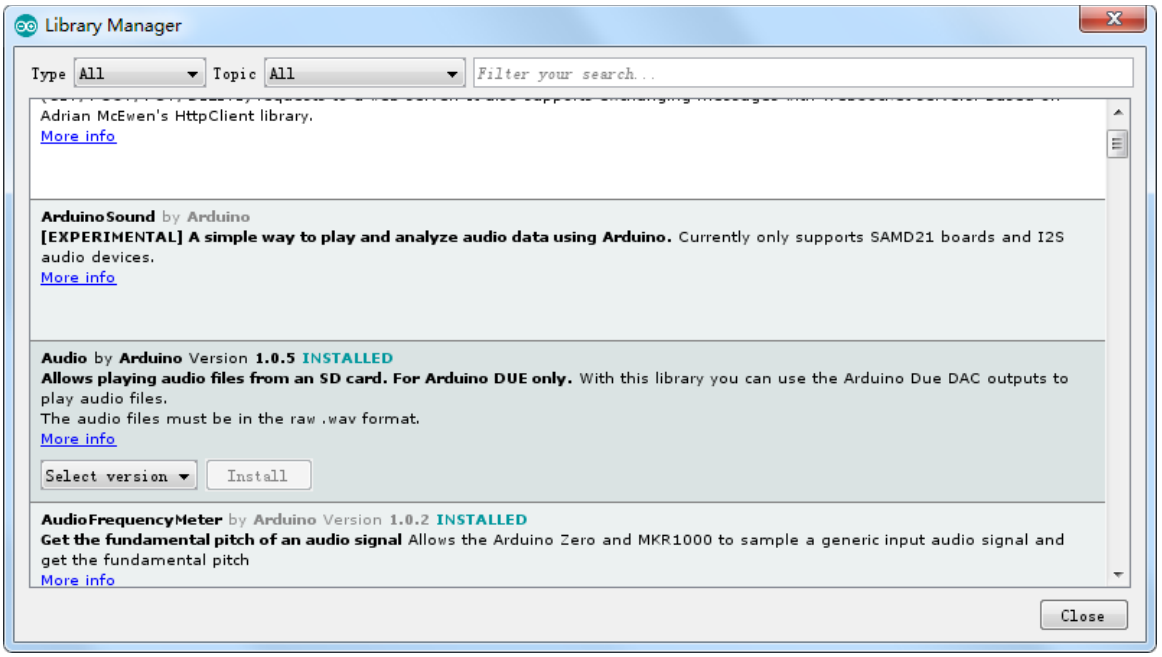
После чего откроется Library Manager, и вы сможете найти список библиотек, которые уже установлены, либо доступны для установки. В этом примере мы покажем, как установить библиотеку <Bridge>. Прокрутите список, чтобы найти ее, затем выберите необходимую вам версию библиотеки для установки. В некоторых случаях доступна только одна версия. Если меню выбора версий не появляется – не переживайте, это нормально.

Иногда приходится быть терпеливым, как показано ниже. В таком случае, пожалуйста, обновите страницу и ожидайте.





Наконец, запустите установку и ожидайте, пока среда разработки установит новую библиотеку. Загрузка может занять некоторое время, в зависимости от скорости вашего подключения к Интернету. По завершению установки, вы увидите тэг “Installed” рядом с названием библиотеки. Теперь можно закрыть Library Manager.

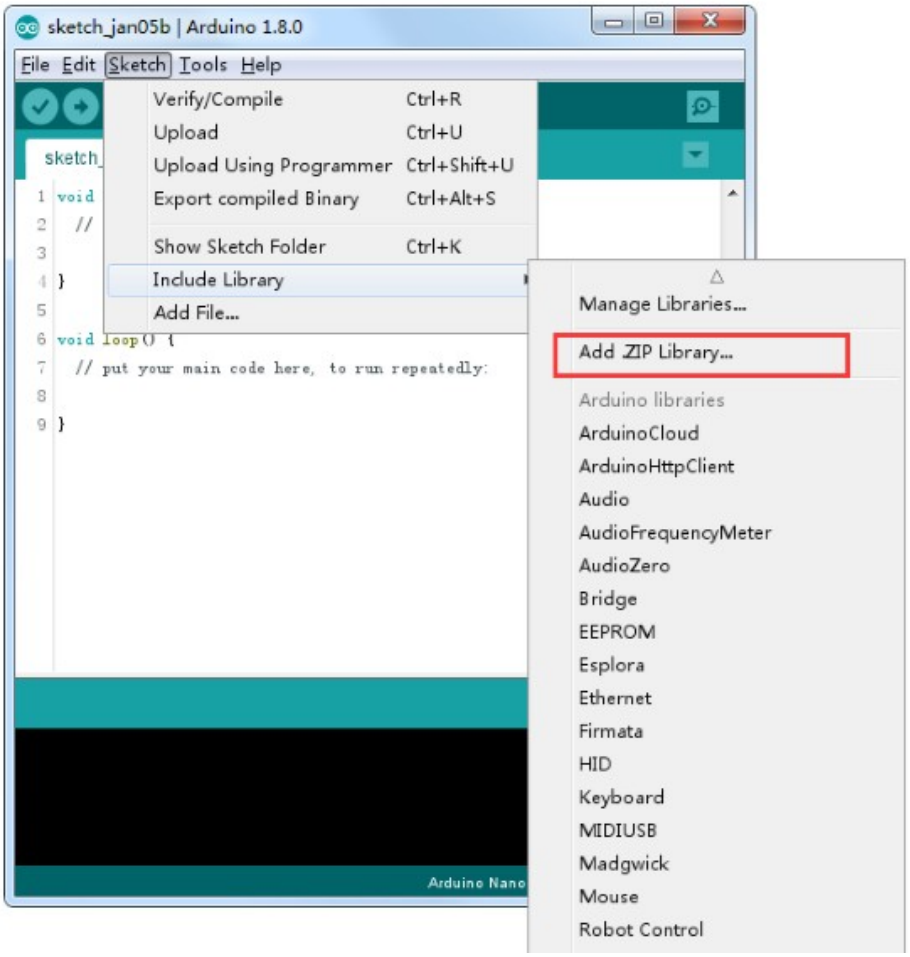


Новая библиотека теперь доступна в меню *Include Library*. Если вы хотите добавить вашу собственную библиотеку, откройте новую тему на Github.

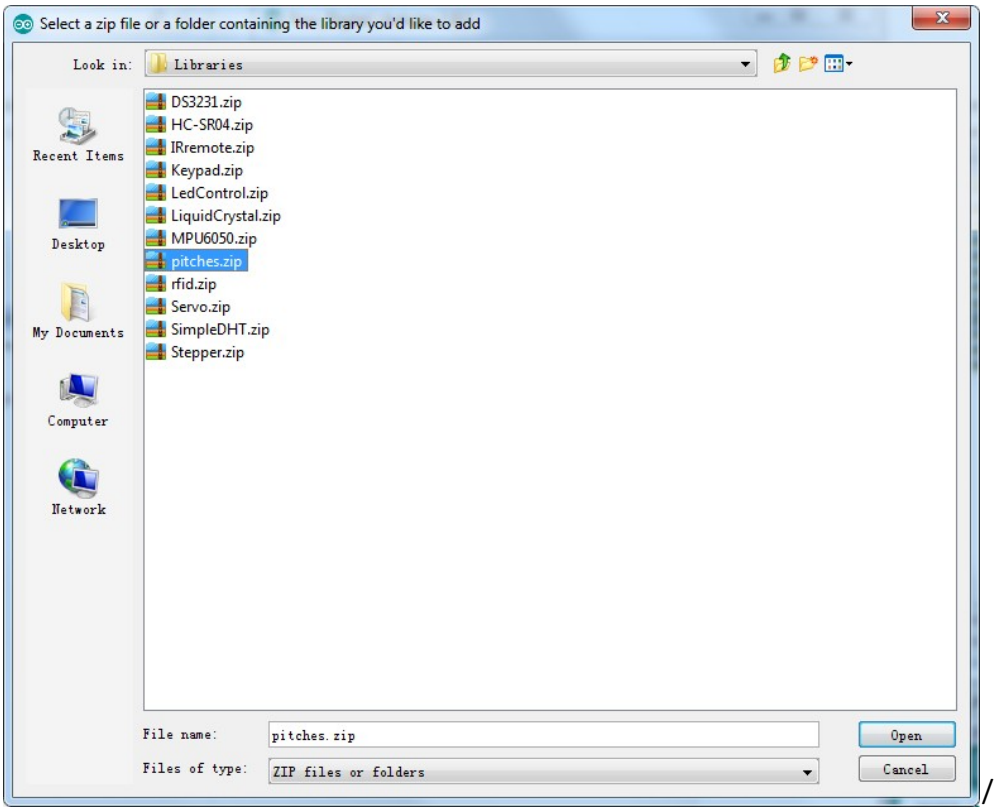
Импорт .zip библиотеки

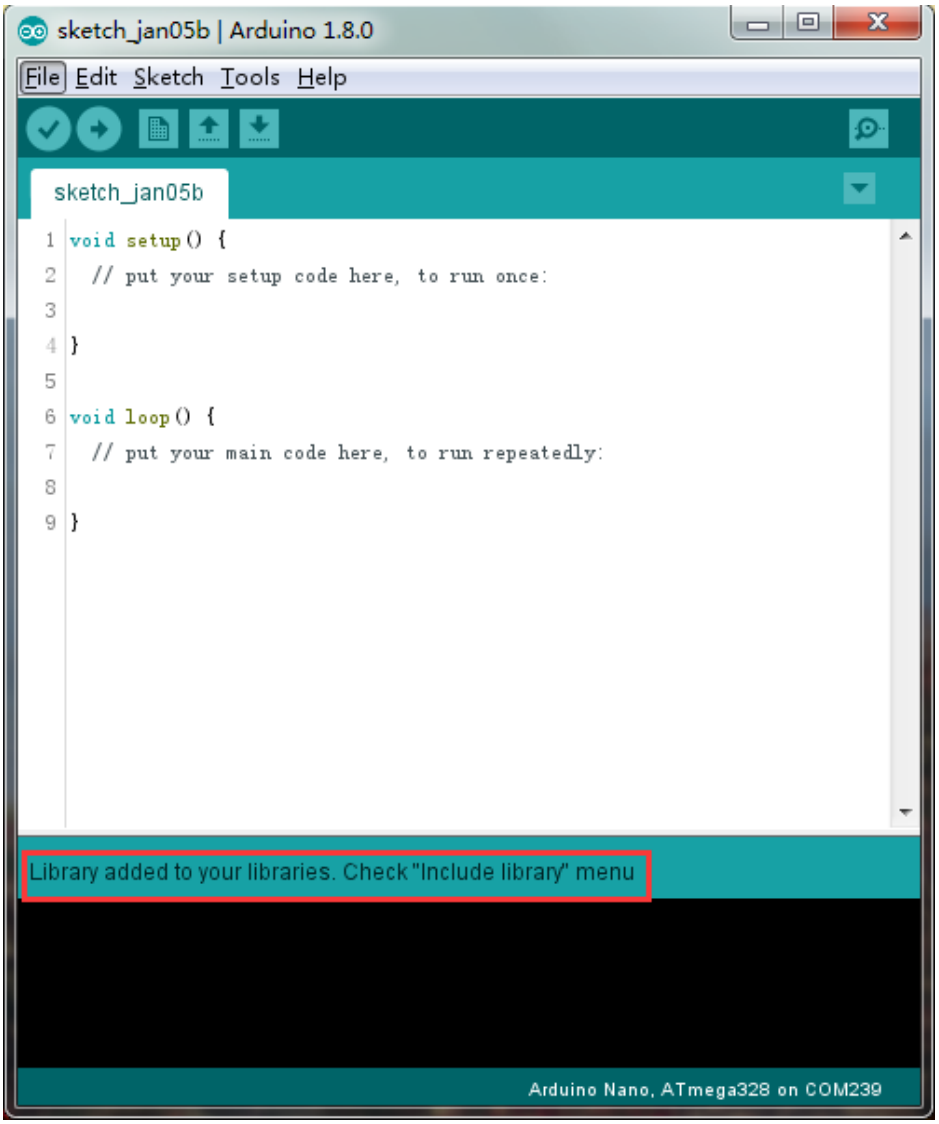
Чаще всего библиотеки выкладываются в виде ZIP-архива или просто папки. Название этой папки является названием библиотеки. Внутри папки будет файл с расширением *.cpp*, файл с расширением *.h*, а также текстовый файл *keywords.txt*, папка с примерами *examples* и другие файлы, требуемые библиотекой. Начиная с версии 1.0.5, устанавливать сторонние библиотеки можно прямо в среде разработки. Не распаковывайте скачанный архив с библиотекой - оставьте его, как есть.

В среде разработки Arduino выберите меню *Sketch > Import Library*. В самом верху выпадающего списка выберите пункт "Add .ZIP Library'' (Добавить .ZIP библиотеку).



Появится диалоговое окно, предлагающее вам выбрать библиотеку, которую вы хотели бы добавить. Перейдите к скачанному zip-файлу и откройте его.





Снова откройте меню *Sketch > Import Library*. Вы должны увидеть новую библиотеку в самом низу выпадающего списка. Теперь библиотеку можно использовать в программах. zip-файл уже распакован в директории *libraries* внутри вашей рабочей папки Arduino.

Примечание: после выполнения указанных действий библиотеку можно будет полноценно использовать в своих программах, однако примеры из установленной библиотеки появятся в меню *File > Examples* только после перезапуска среды разработки.

Это два самых распространенных подхода. Пользователи систем Mac и Linux могут поступать таким же образом. Установка вручную, которая приводится ниже в качестве альтернативного метода, используется пользователями намного реже и абсолютно добровольно, поэтому незаинтересованные читатели могут сразу переходить к следующему уроку.

Установка вручную

Перед установкой библиотеки закройте среду разработки Arduino. Затем распакуйте ZIP-архив с библиотекой. Допустим, вы устанавливаете библиотеку "ArduinoParty" - распакуйте файл *ArduinoParty.zip*. В нем должна быть папка *ArduinoParty* с файлами *ArduinoParty.cpp* и *ArduinoParty.h*. (Если файлы с расширением *.cpp* и *.h* лежат не в папке, то необходимо ее создать. В данном случае вам нужно создать папку с именем *ArduinoParty* и перенести в нее все файлы из ZIP-архива, например - *ArduinoParty.cpp* и *ArduinoParty.h*).

Перетащите папку *ArduinoParty* в директорию с библиотеками Arduino. В Windows она будет лежать примерно здесь: "My Documents\Arduino\libraries", у пользователей Mac - здесь: "Documents/Arduino/libraries", а в Linux-системах директория "*libraries*" будет внутри рабочей папки со скетчами.

После перемещения ваша директория с библиотеками должна выглядеть примерно так (под Windows):

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h

My Documents\Arduino\libraries\ArduinoParty\examples

....

или так (на Mac):

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h

Documents/Arduino/libraries/ArduinoParty/examples

...

Помимо файлов *.cpp* и *.h* здесь могут быть и другие файлы - просто убедитесь, что все они теперь лежат здесь. (Если файлы *.cpp* и *.h* расположены в корне папки "libraries" или вложены внутрь еще одной папки, то скачанная библиотека работать не будет. Например, Documents\Arduino\libraries\ArduinoParty.cpp и Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp – не рабочий пример.)

Перезапустите среду Arduino. Убедитесь, что новая библиотека появилась в меню *Sketch -> Import Library*. Вот и все! Вы установили библиотеку!

Работа с серийным монитором (Serial Monitor) (Windows, Mac, Linux)

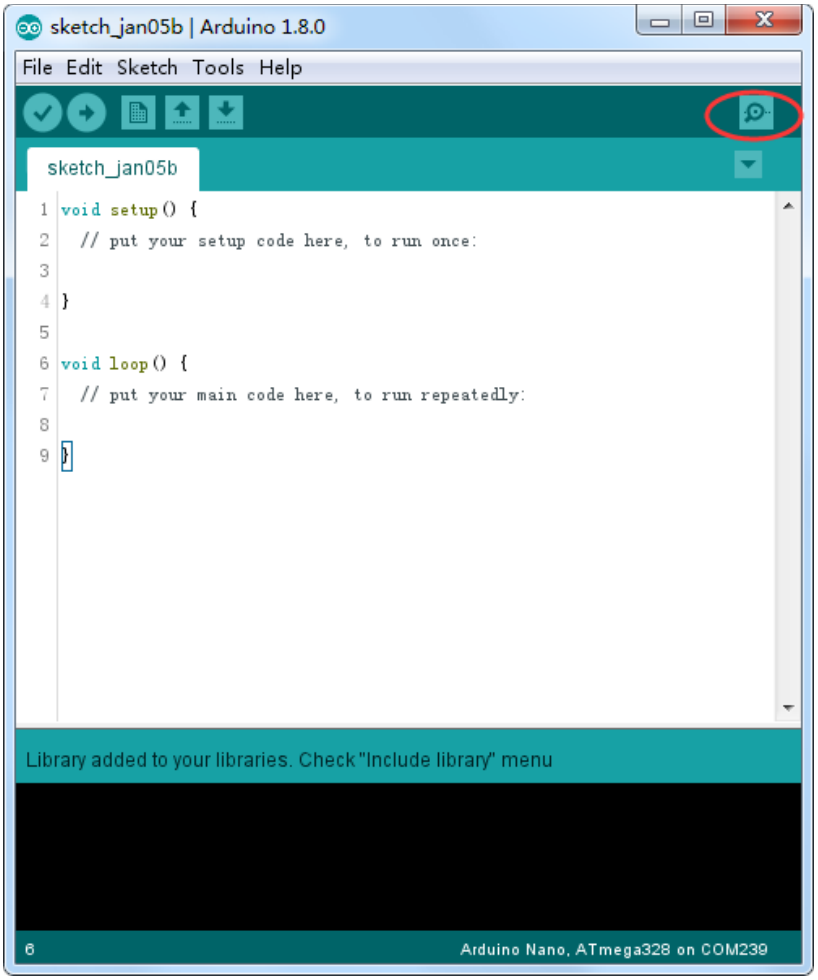
Среда разработки Arduino (IDE) – это программная сторона платформы Arduino.

Поскольку использования терминала представляет собой огромную часть

работы с платами Arduino и другими микроконтроллерами, было решено включить в программу терминал для отображения данных, переданных через последовательный порт. В среде Arduino он называется “серийный монитор”.

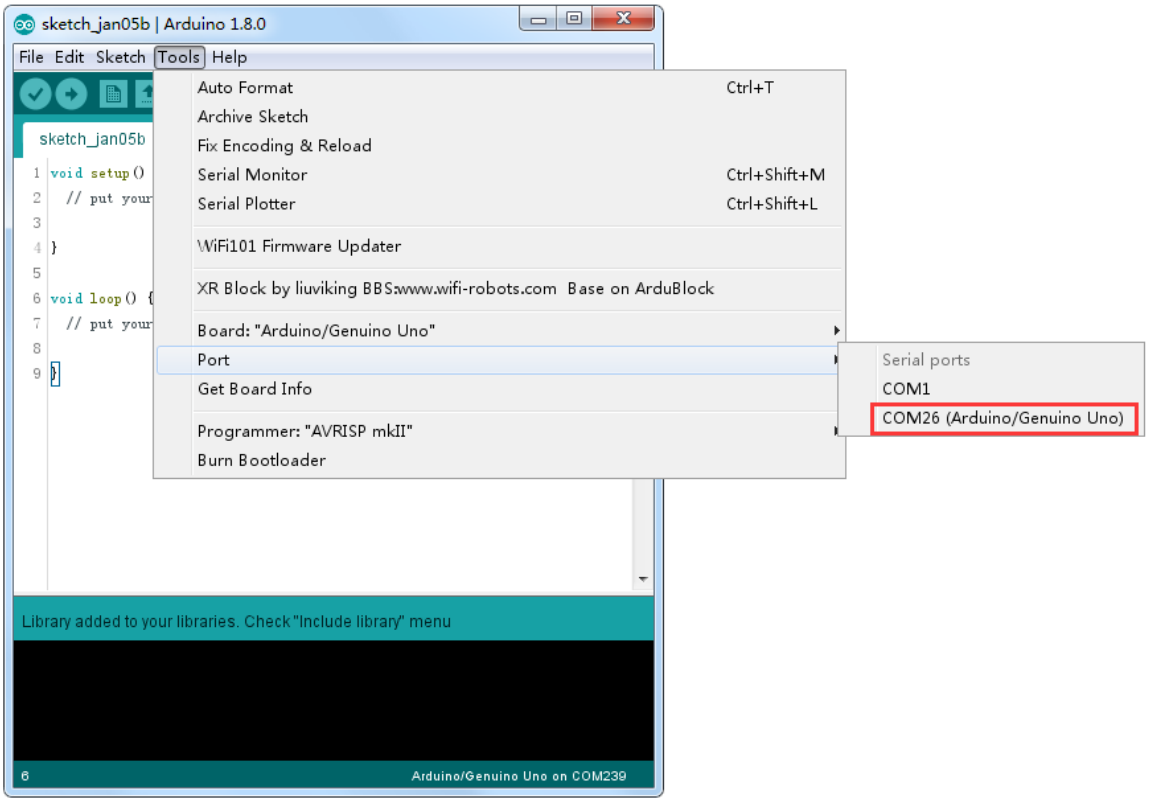
Подключение

Серийный монитор встроен во все версии среды разработки Arduino. Для его открытия, просто кликните на иконку Serial Monitor.

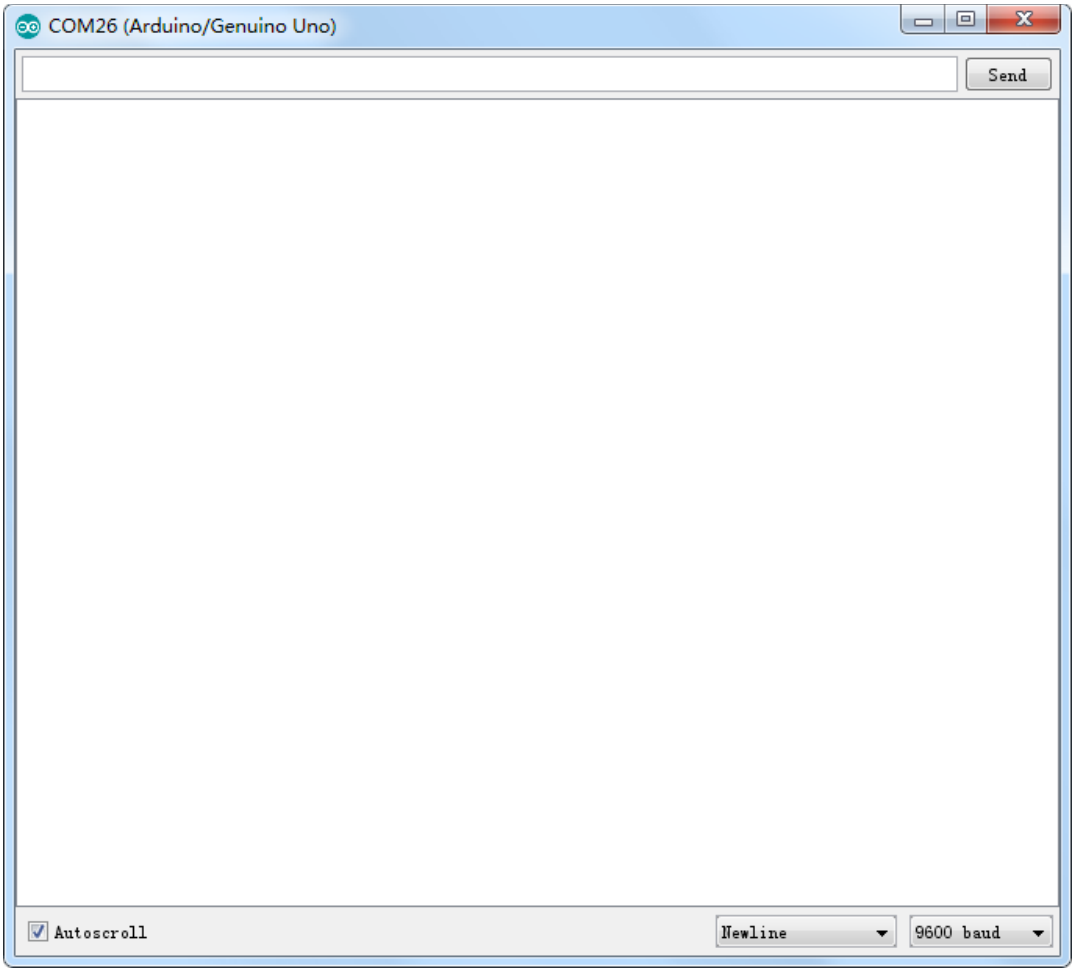


Выбор порта для отображения в серийном мониторе аналогично выбору порта для загрузки кода Arduino. Перейдите к Tools -> Serial Port и выберите необходимый порт серийного монитора.

Подсказка: Выберите COM-порт, который отображается в Диспетчере Устройств.

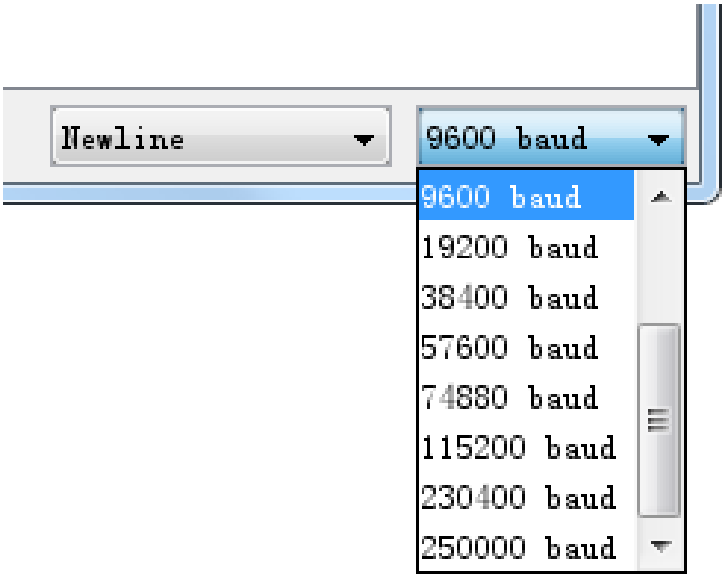


После открытия вы должны увидеть что-то, похожее на это:

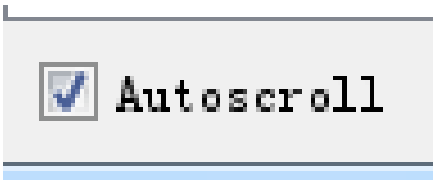


Настройки

Настройки серийного монитора довольно ограничены, но достаточны для удовлетворения большинства ваших требований к последовательной передаче данных. Первый регулируемый параметр – скорость передачи в бодах. Нажмите на выпадающее меню для выбора необходимой скорости (9600 бод).



Дополнительно, вы можете установить терминал на автопрокрутку (Autoscrol), установив или сняв флажок в нижнем левом углу.



Достоинства

Серийный монитор – это быстрый и простой способ установления последовательного подключения с вашим Arduino. Если вы уже работаете в среде разработки Arduino, то нет необходимости открывать отдельный терминал для отображения данных.

Недостатки

Количество регулируемых настроек серийного монитора оставляет желать лучшего. В случае нужды в более продвинутой последовательной передаче данных, он может не справиться с задачей.

Урок 2: Мигающий светодиод

Обзор

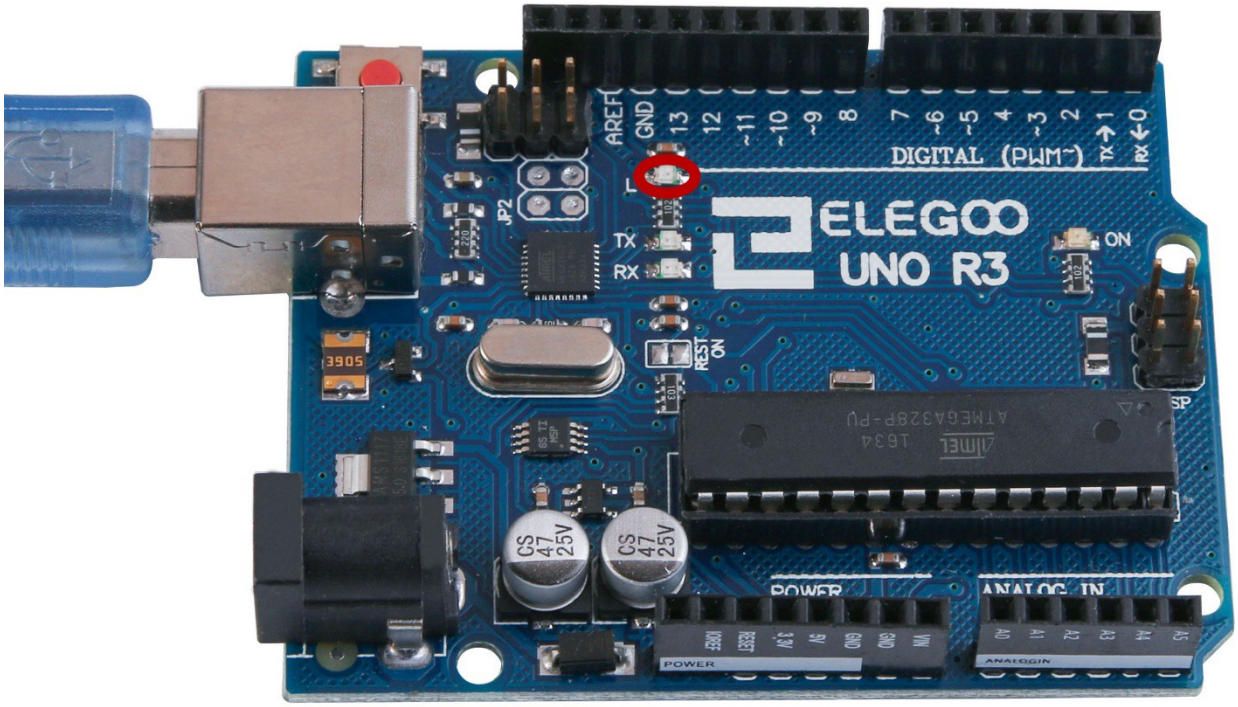
В этом уроке рассказывается, как заставить ваш UNO R3 контроллер приветственно помигать нам встроенным светодиодом, и как загружать программы шаг за шагом.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

Принцип

Плата UNO R3 имеет ряд разъемов с обеих сторон, используемых для подклю



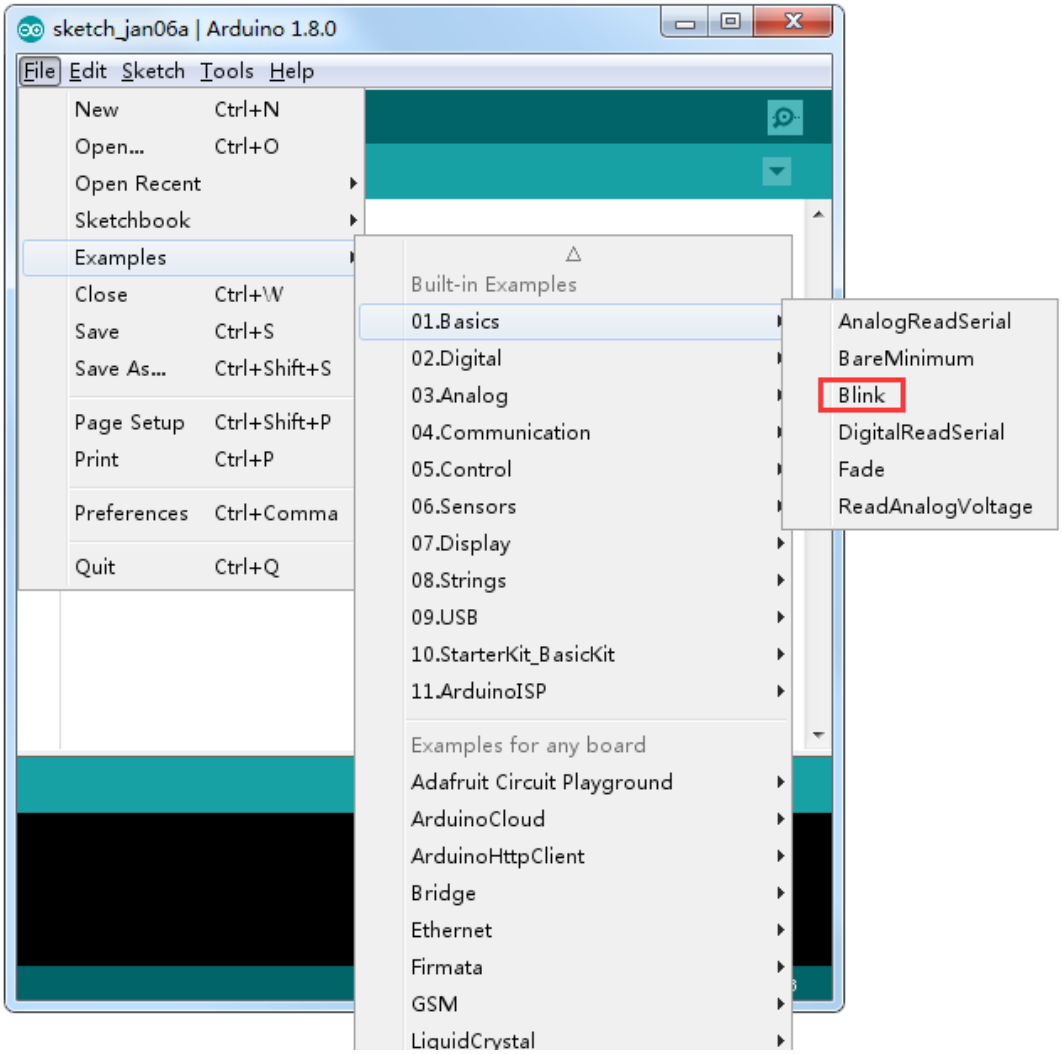
При подключении новой платы к персональному компьютеру, обратите внимание, что ваш светодиод начинает мигать, так как все платы от производителей поступают с уже “залитым” скетчем “Blink”.

На этом уроке мы перепрограммируем нашу UNO R3 плату, изменив частоту мигания светодиода.

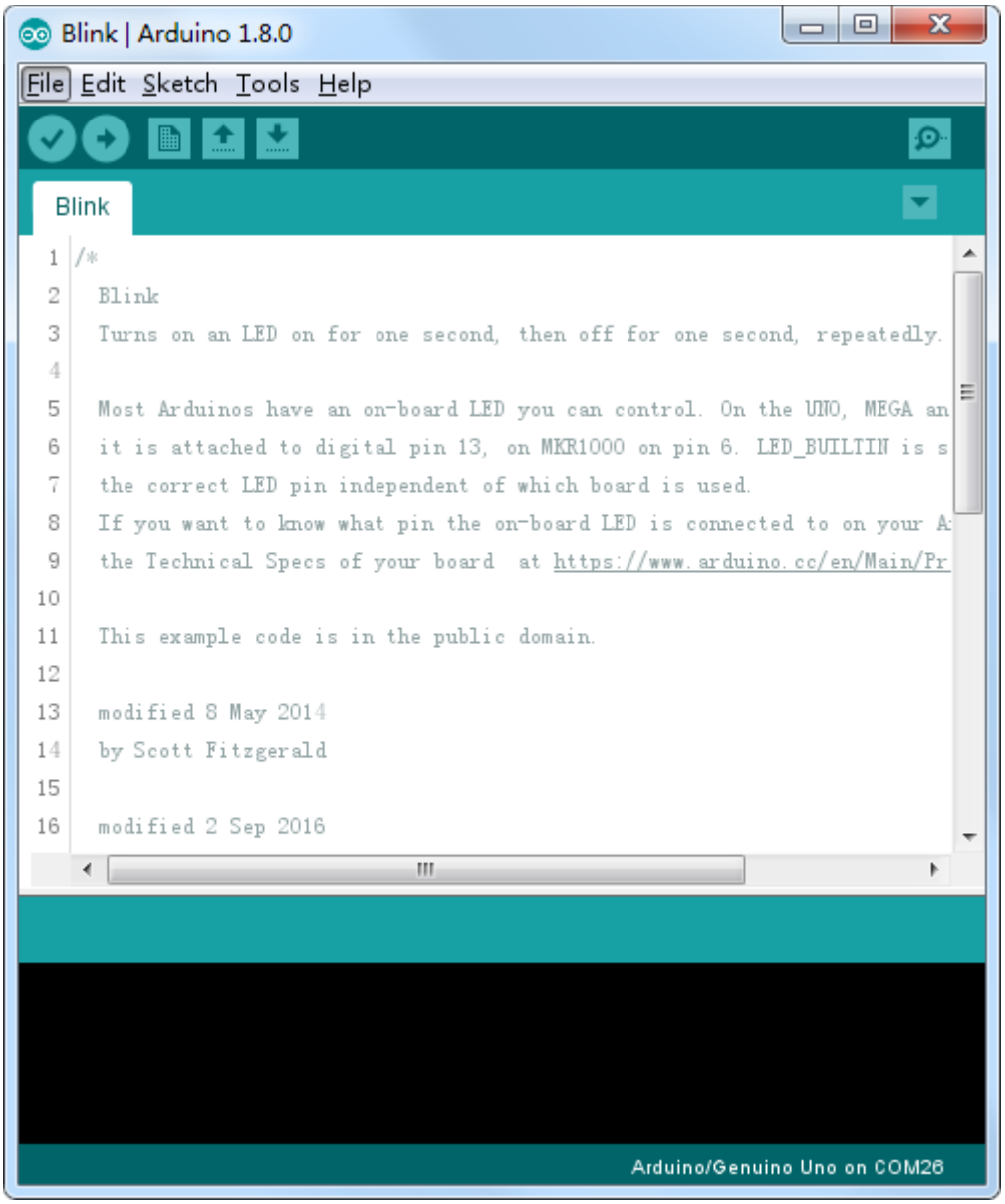
Пройдя урок 0, вы уже настроили оболочку Arduino IDE и убедились, что выбрали нужный серийный порт, по которому вы подключили вашу UNO R3 плату. Пришло время проверить ваше подключение и запрограммировать плату.

В оболочке Arduino IDE существует большая коллекция скетчей, которые уже готовы к использованию. Среди них находится и пример, который заставляет мигать “L” светодиод.

Откройте пример “Blink”, который находится в пункте меню File > Examples > 01.Basics



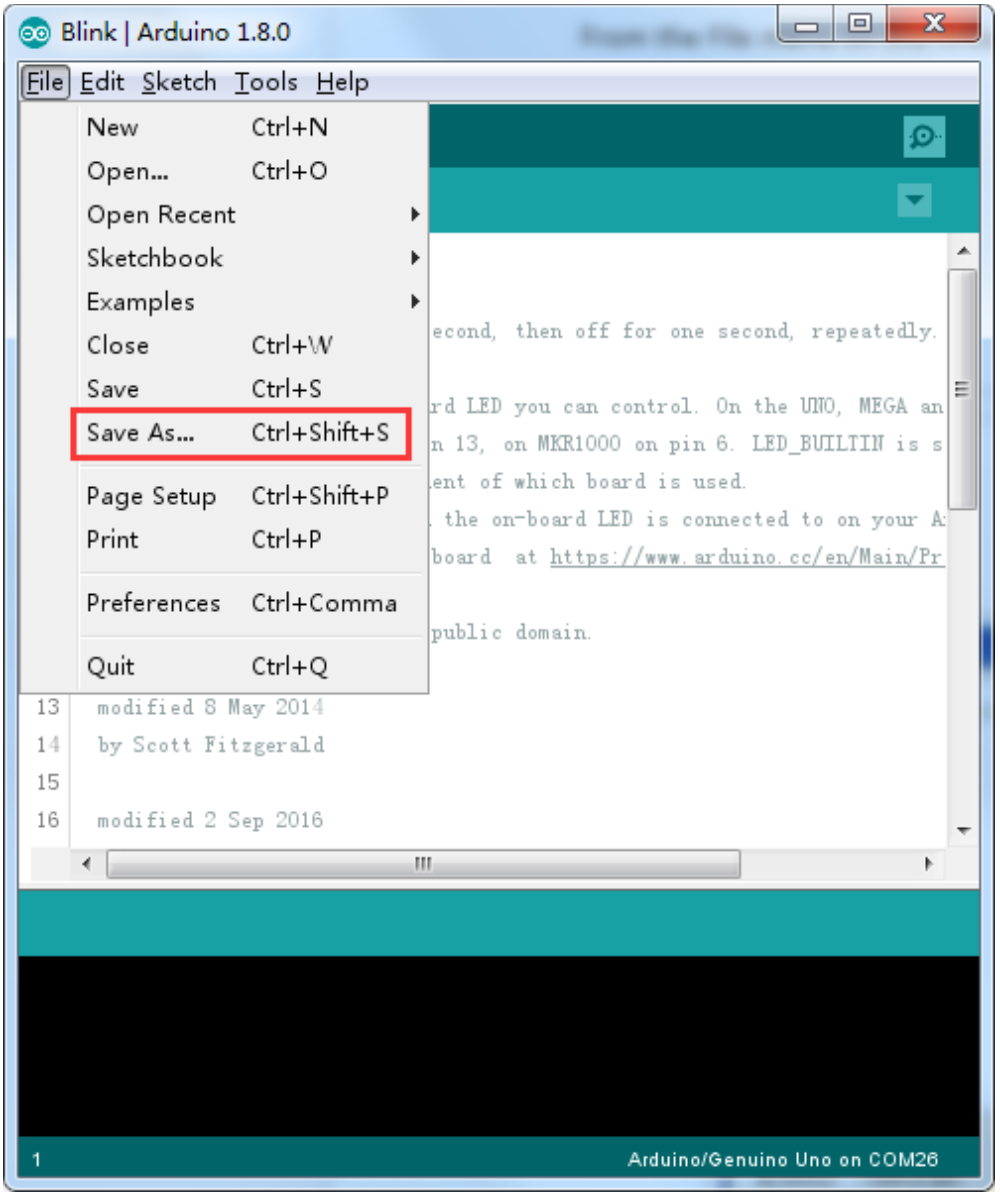
После открытия, расширьте окно оболочки Arduino IDE, чтобы Вы могли видеть весь скетч в одно окне.



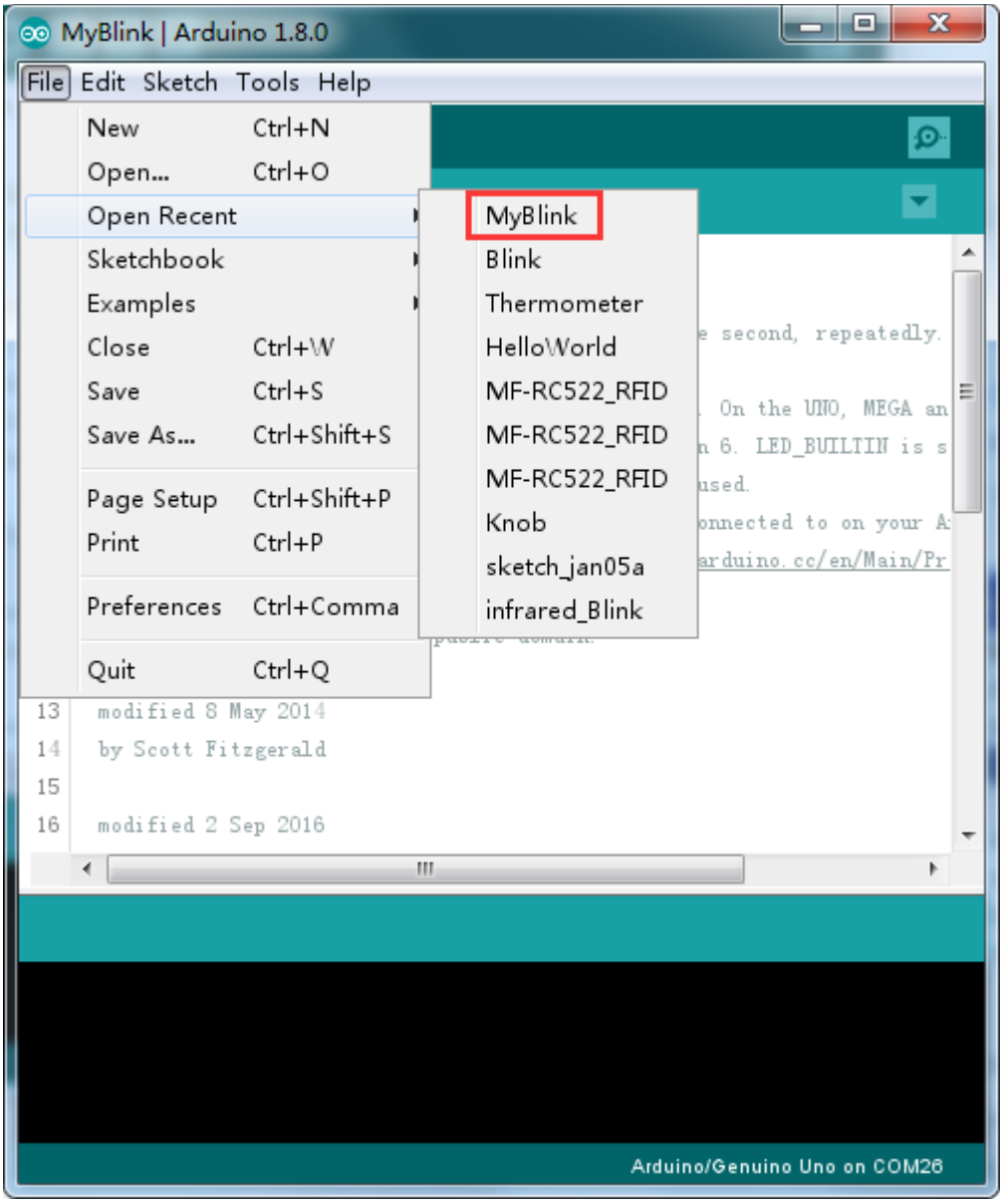
Скетчи из примеров, включенные в Arduino IDE, предусматривают режим “только чтение” (“read only”). То есть, загрузить их на плату UNO R3 вы сможете, но после изменения кода, вы не сможете их сохранить в том же файле.

Мы будем изменять скетч, так что, в первую очередь, вам необходимо сохранить собственную копию, которую вы сможете изменять.

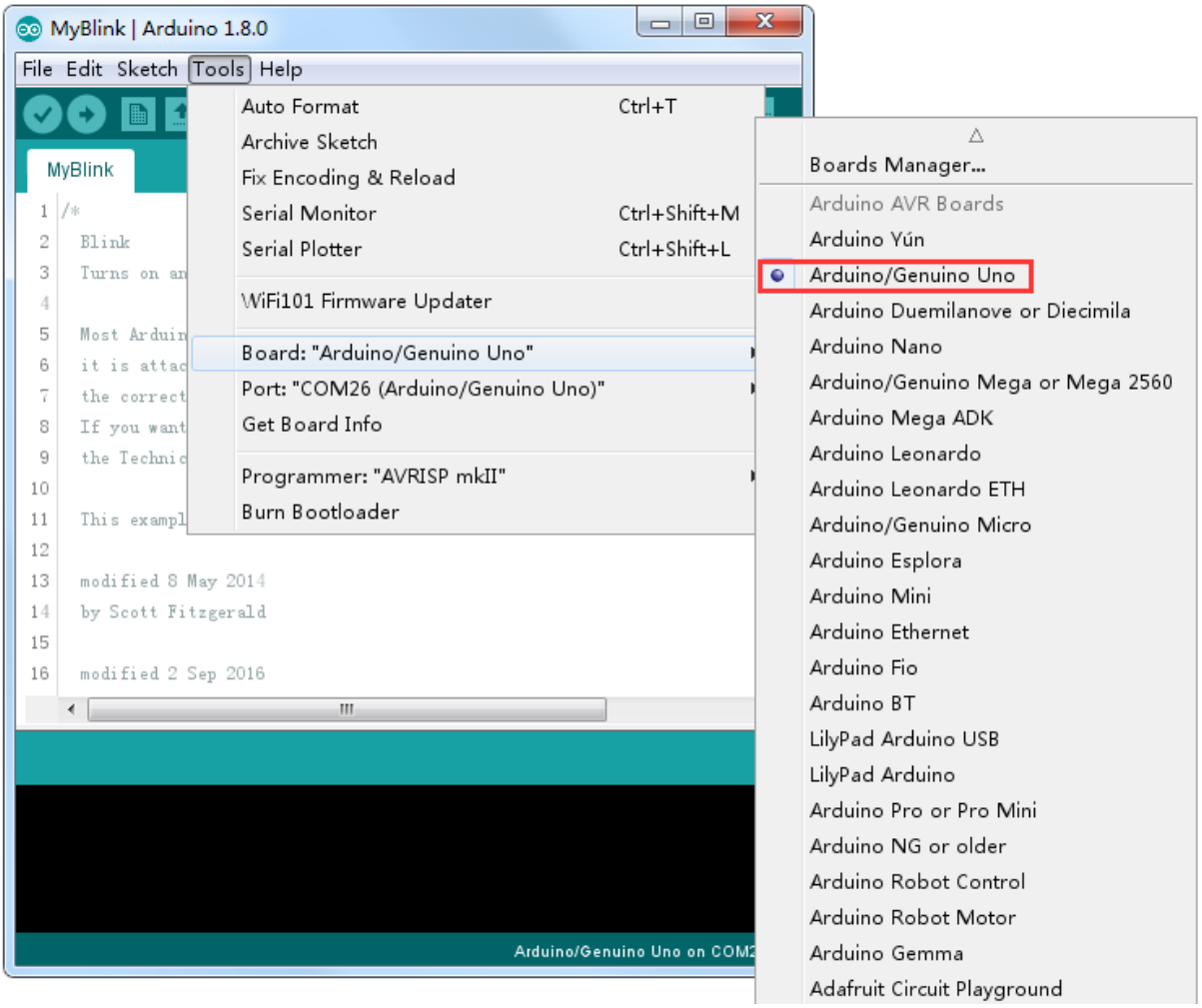
Из меню “File” выберите опцию “Сохранить как” (“Save As..”) и сохраните скетч под подходящим вам названием, например, “MyBlink”.

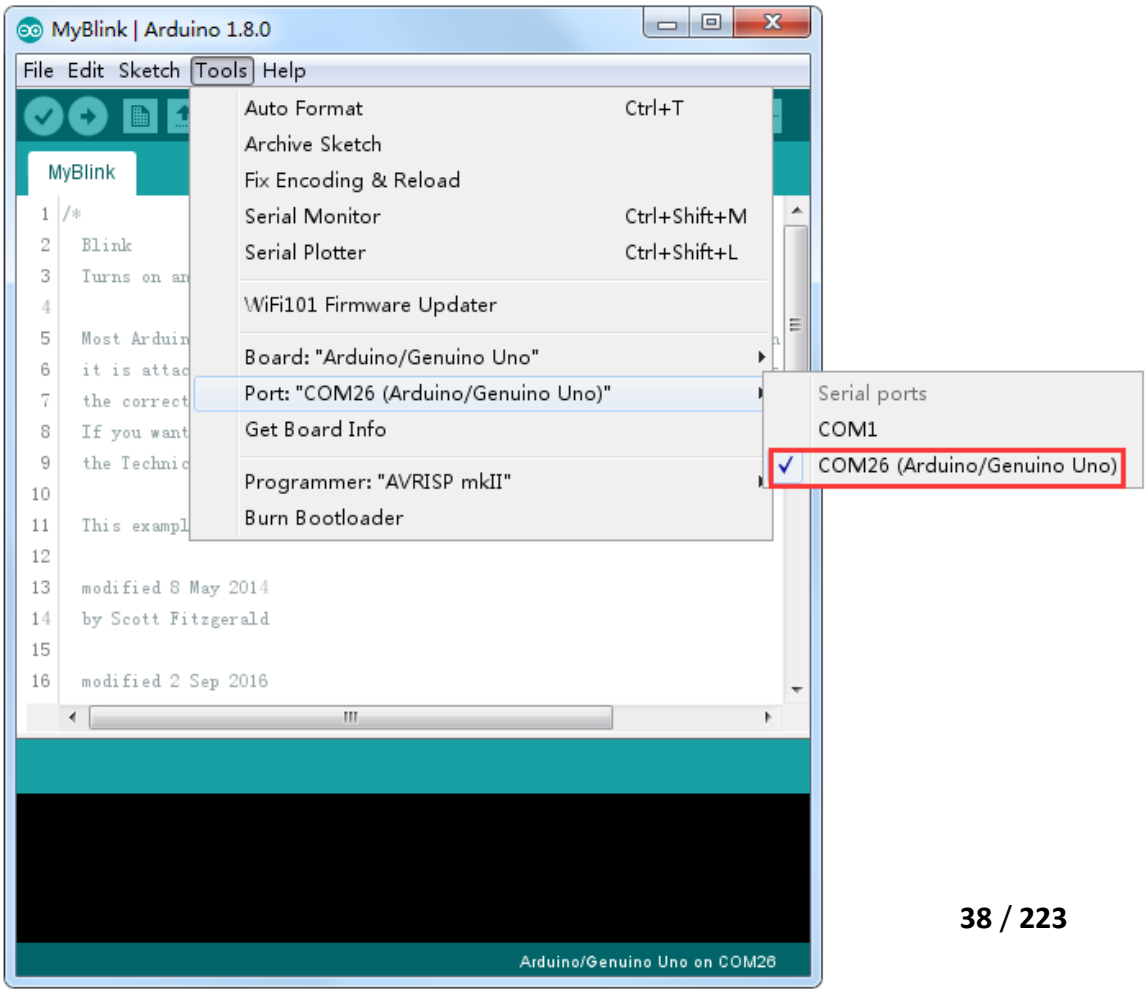

Blink | Arduino 1.8.0
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
13 modified 8 May 2014
14 by Scott Fitzgerald
15
16 modified 2 Sep 2016
1
Arduino/Genuino Uno on COM26

Вы сохранили копию скетча “Blink” в вашей библиотеке. Теперь открыть этот файл вы можете в любой момент, перейдя по вкладке File > Sketchbook.



Подключите свою плату Arduino к компьютеру с помощью USB и проверьте тип платы (“Board type”) и серийный порт (“Serial Port”), по которому она подключена.

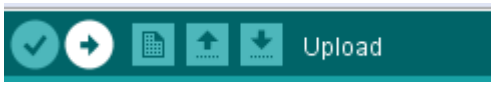

MyBlink | Arduino 1.8.0
File Edit Sketch Tools Help
MyBlink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM26 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino/Genuino Uno on COM


MyBlink | Arduino 1.8.0
File Edit Sketch Tools Help
MyBlink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM26 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM26 (Arduino/Genuino Uno)
Arduino/Genuino Uno on COM26

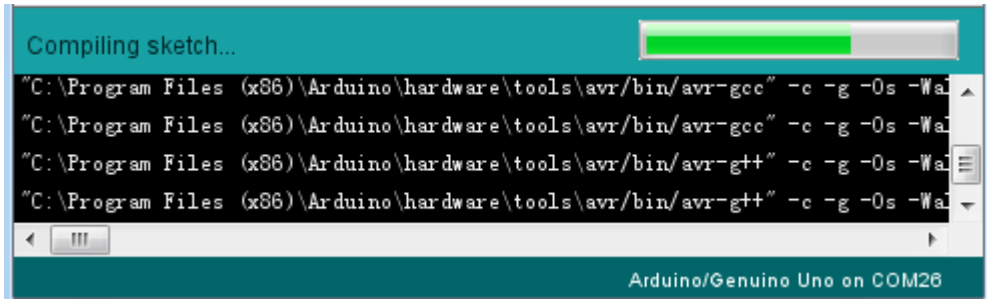
Примечание: Тип платы и серийный порт не обязательно соответствуют тем, которые показаны на изображении. Если вы используете плату 2560, то вам следует выбрать Mega 2560 в качестве типа платы. Оставшиеся варианты могут быть выбраны аналогично. Серийный порт также будет отличаться для каждого. Несмотря на то, что тут выбран COM 26, для вашего компьютера это может быть COM3 или COM4. Правильный COM-порт должен быть COMx (Arduino XXX), соответственно с сертификацией.

Текущие настройки отображаются внизу окна оболочки Arduino IDE

Кликните на кнопку “Загрузить” (“Upload”). Вторая кнопка слева на панели инструментов.



Во время загрузки в нижней части окна IDE появятся ползунок загрузки и сообщения. Вначале появляется фраза “Компилирование” (“Compiling sketch...”), что означает процесс конвертирования вашего скетча в формат, подходящий для загрузки на плату Arduino.



Дальше статус сменится на “Загрузка” (“Uploading”). В этот момент светодиоды на плате начнут мигать, так как начнется перенос скетча в микропроцессор.

В конце статус сменится на “Загрузка завершена” (“Done uploading”).

В сообщении, которое появится в текстовой строке отобразится информация о том, что загруженный скетч занимает 928 байтов из 32,256 доступных. Иногда при компиляции у вас может возникнуть подобная ошибка:

Причин может быть несколько: вы не подключили плату к компьютеру; вы не установили необходимые драйвера; вы выбрали некорректный серийный порт. Если вы столкнулись с одной из этих проблем, вернитесь к уроку 0 и проверьте ваши настройки. Если же загрузка прошла корректно, плата Arduino перезагрузится и “L” светодиод начнет мигать.

Откройте код

Обратите внимание, что в данном скетче множество “комментариев”. Комментарии не являются инструкцией по работе программы. Это исключительно пояснения отдельных функций и задач, которые выполняются на определенном этапе кода. Они приводятся здесь только для вас.

Все между символами /* и */ в верхней части скетча – это комментарии, в которых описаны задачи программы.

Так же есть комментарии, которые ограничиваются одной строкой. Они начинаются с символов // и заканчиваются по умолчанию в конце строки. Первая важная, по сути, часть данного кода — это строка:

```
int led = 13;
```

В комментариях над строкой указано, что мы присваиваем имя пину, к которому подключен светодиод. На большинстве плат Arduino это будет 13-ый пин, включая платы UNO и Leonardo.

Дальше используется функция “Setup”. Опять-таки, в комментариях указано, что функция срабатывает после нажатия кнопки “reset”. Также эта функция срабатывает, когда плата перезагрузится по каким-либо другим причинам. Например, подача питания или после загрузки скетча.

```
void setup() {
// инициализировать цифровой пин на вывод. pinMode(led, OUTPUT);
}
```

Каждый скетч Arduino обязан включать в себя функцию “setup” и часть, в которую можно добавлять собственные инструкции, заключенные между { и }. В нашем примере в функции присутствует только одна команда, в которой указано, что пин, который мы используем, настраивается на “вывод” (“Output”).

Также обязательным для любого скетча является функция цикла “Loop”. В отличие от функции “Setup”, которая отрабатывает один раз после перезагрузки, функция “Loop” после окончания работы команд, вновь запустится.

```
void loop() {
digitalWrite(led, HIGH);      // включить светодиод (HIGH)
delay(1000);      // подождите секунду
digitalWrite(led, LOW);      // выключить светодиод, понизив напряжение (LOW)
delay(1000);      // подождите секунду
}
```

В теле функции “Loop” светодиод включается (HIGH), данное значение задерживается на 1000 миллисекунд (1 секунда), светодиод отключается (LOW) и остается выключенным на 1 секунду, после чего цикл повторится.

Теперь мы повысим частоту мигания нашего светодиода. Как вы уже могли догадаться, для того, чтобы обеспечить более частое мигание светодиода, необходимо изменить параметр, указываемый в скобках ( ) в команде “delay”.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

Как уже было указано, период задержки указывается в миллисекундах. То есть, для того, чтобы заставить светодиод мигать в два раза чаще, необходимо изменить значение с 1000 на 500. В результате, пауза между включением/выключением светодиода составит половину секунды и светодиод будет мигать быстрее.

Для проверки, не забудьте загрузить измененный скетч на плату Arduino.

Урок 3: Светодиод

Обзор

В этом уроке вы научитесь изменять яркость светодиода, используя резисторы с разным сопротивлением.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x красный светодиод (5мм)

(1) x резистор (220 Ом)

(1) x резистор (1 кОм)

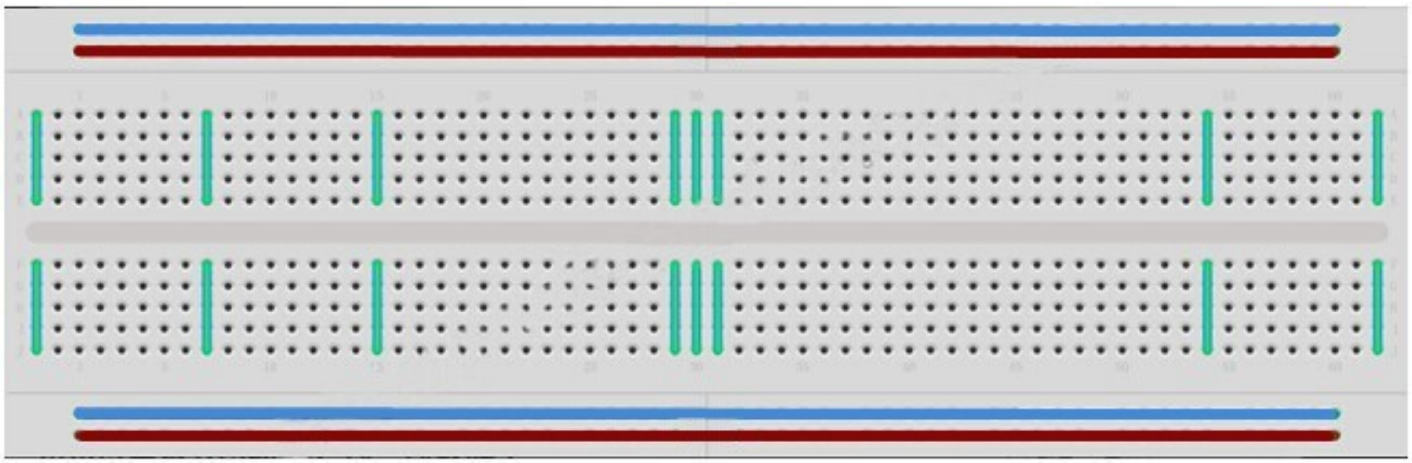
(1) x резистор (10 кОм)

(2) x перемычки (типа “папа”-“папа”)

Представление Компонентов

Макетная плата MB-102：

Макетная плата позволяет быстро и без использования паяльника создавать и моделировать прототипы электронных цепей. Ниже приведен пример.



Макетные платы бывают разных размеров и конфигураций. Простейшие представляют собой матрицу из отверстий-гнезд в пластиковом корпусе. Внутри проложены металлические полоски, обеспечивающие электрическое соединение между отверстиями. Например, вставив ножки двух разных компонентов в один ряд на плате, мы включаем их в цепь. Глубокая канавка, проходящая посередине платы, указывает на отсутствие металлических соединений в этом месте. Это значит, что вы можете воткнуть ножки чипа в отверстия по обе стороны от канавки, при этом не замыкая контакты. Некоторые платы имеют два ряда отверстий, нанесенных вдоль длинных сторон, отделенных от главной матрицы отверстий. Это шины питания (плюсовая «+» и минусовая «-»), они маркированы красной и синей линией вдоль контактных точек. Все точки шины электрически соединены между собой и, по сути представляют собой один проводник, но с множеством точек-разъёмов.

В то время, как макетные платы – прекрасный инструмент для создания прототипов, они, всё же, имеют некоторые ограничения. Поскольку соединения элементов на плате являются временными, они менее надежные чем распайка. Поэтому присутствие временных неисправностей в цепи вызвано, скорее всего, неплотными соединениями на макетной плате.

Светодиод:

Светодиоды отлично служат для разного рода индикации. Они потребляют мало электричества и при этом долговечны.

В данном примере мы используем самые распространенные светодиоды диаметром 5 мм. Также распространены светодиоды диаметром 3 миллиметра, ну и большие светодиоды диаметром 10 мм.

Подключать светодиод напрямую к батарейке или источнику напряжения не рекомендуется. Во-первых, надо сначала разобраться, где именно у светодиода отрицательная и положительная ноги. Ну а во-вторых, необходимо использовать токоограничивающие резисторы, иначе светодиод очень быстро перегорит!

Если вы не будете использовать резистор со светодиодом, последний очень быстро выйдет из строя, так как через него будет проходить слишком большое количество тока. В результате светодиод нагреется и контакт, генерирующий свет, разрушится.
Различить позитивную и негативную ноги светодиода можно двумя способами.
Первый – позитивная нога длиннее.
Второй – при входе в корпус самого диода на коннекторе негативной ноги есть плоская кромка.
Если вам попался светодиод, на котором плоская кромка на более длинной ноге, длинная нога все равно является позитивной.

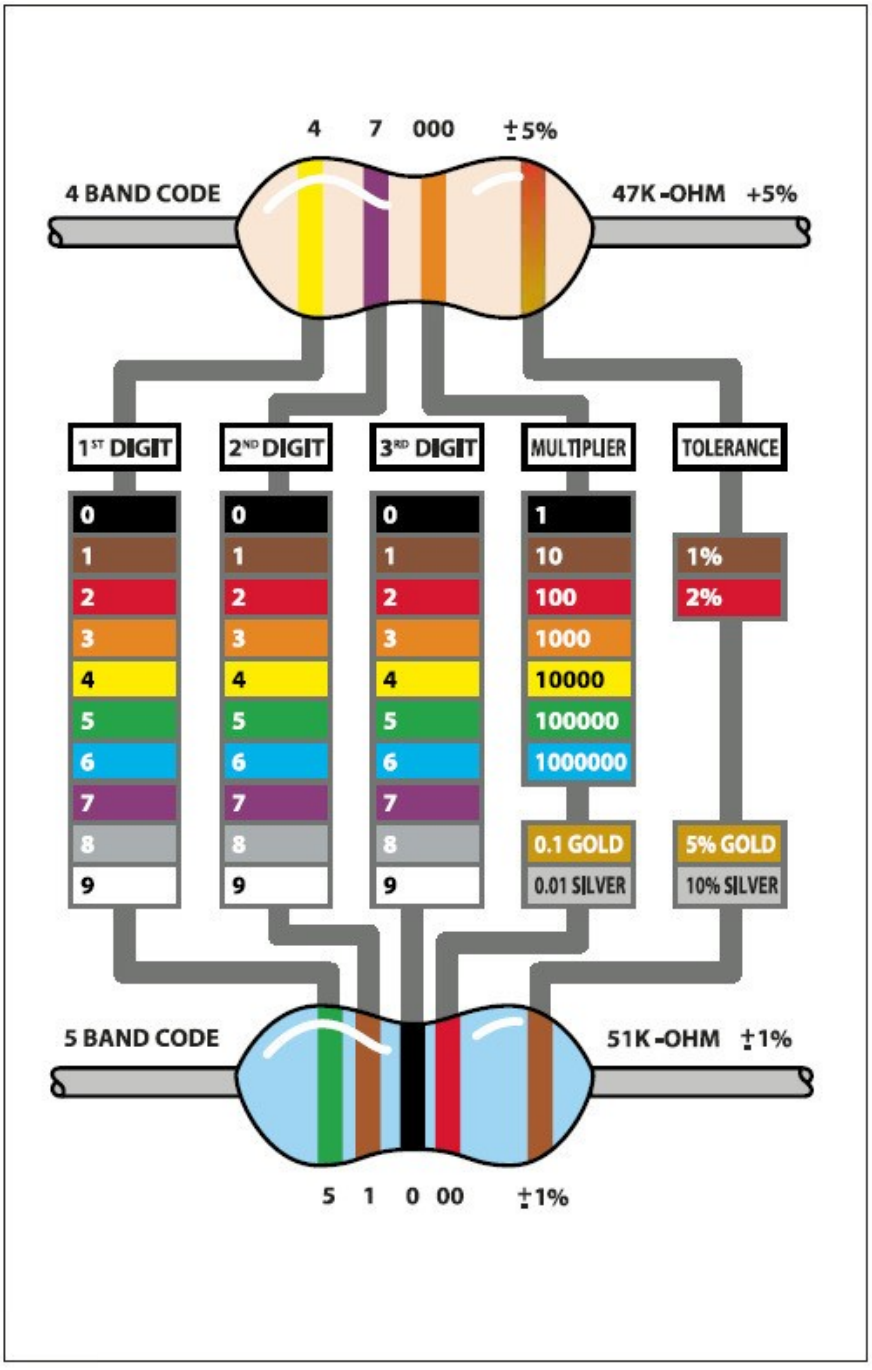
Резисторы:

Из названия можно догадаться, что резисторы сопротивляются потоку электричества. Чем больше номинал (Ом) резистора, тем больше сопротивление и тем меньше тока пройдет по цепи, в которой он установлен. Мы будем использовать это свойство резисторов для регулирования тока, который проходит через светодиод и, таким образом, его яркость.

Но сначала погорим немного о резисторах…
Единицы, в которых измеряется сопротивление – Ом, которые во многих источниках обозначаются греческой буквой Ω – Омега Так как Ом – маленькое значение сопротивления (практически незаметное в цепи), мы часто будем оперировать такими единицами как кОм - килоом (1000 Ом) и МОм мегаом (1000000 Ом).
В данном примере мы будем использовать резисторы с четырьмя различными номиналами: 220 Ом, 1 кОм и 10 кОм. Размеры этих резисторов одинаковы. Цвет

тоже. Единственное, что их различает – цветные полоски. Именно по этим полоскам визуально определяется номинал резисторов.

Цветовая маркировка резистора состоит из трех цветных полосок и четвертой золотой полоски на корпусе резистора.



В отличии от светодиодов, у резисторов нет положительной и отрицательной ног. Какой именно ногой подключать их к питанию/земле – неважно.

Если данный подход кажется вам слишком сложным, вы можете непосредственно посмотреть на цветовую маркировку на наших резисторах для определения их сопротивления. Вы также можете использовать цифровой мультиметр.

Подключение

Схема

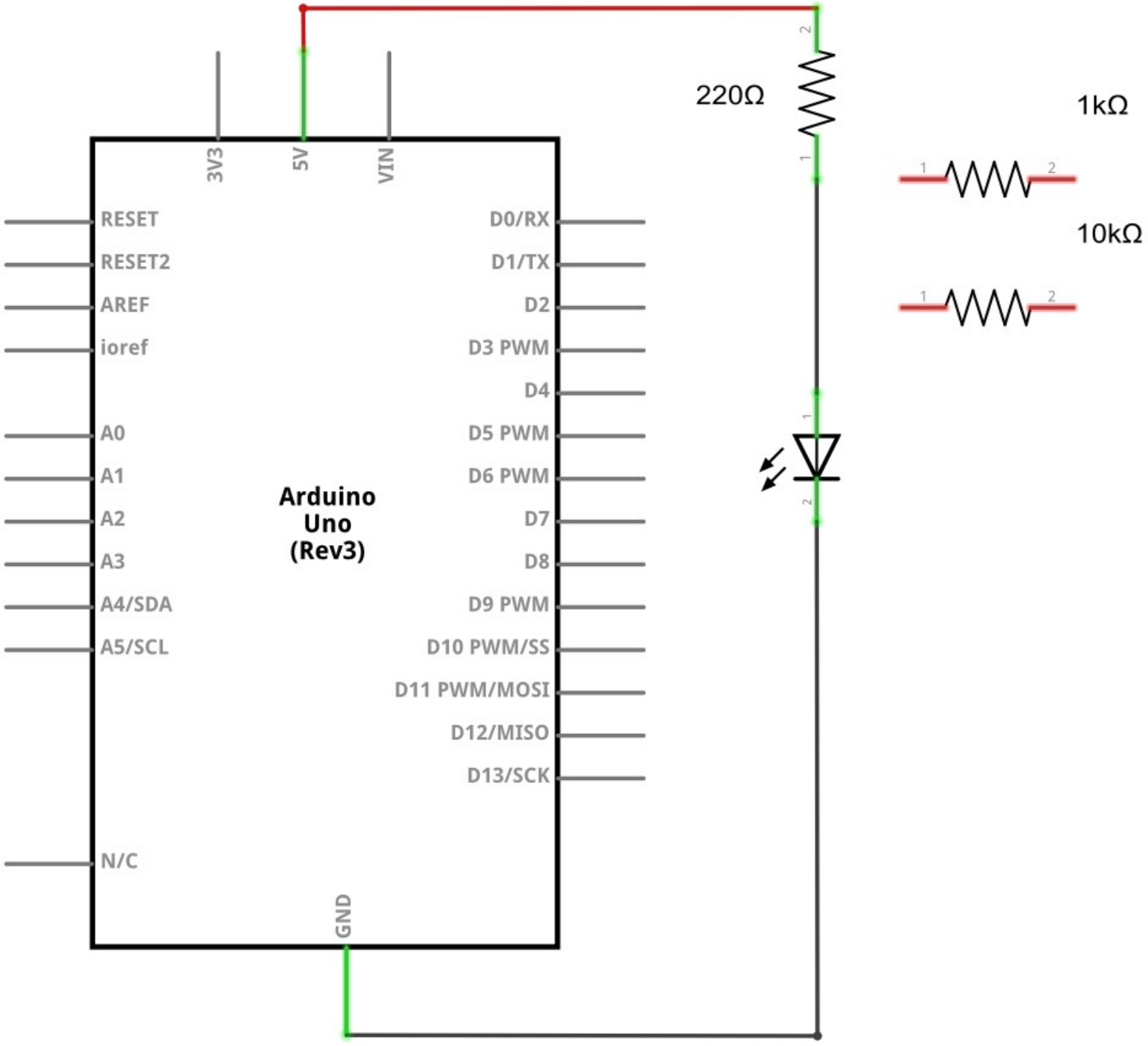
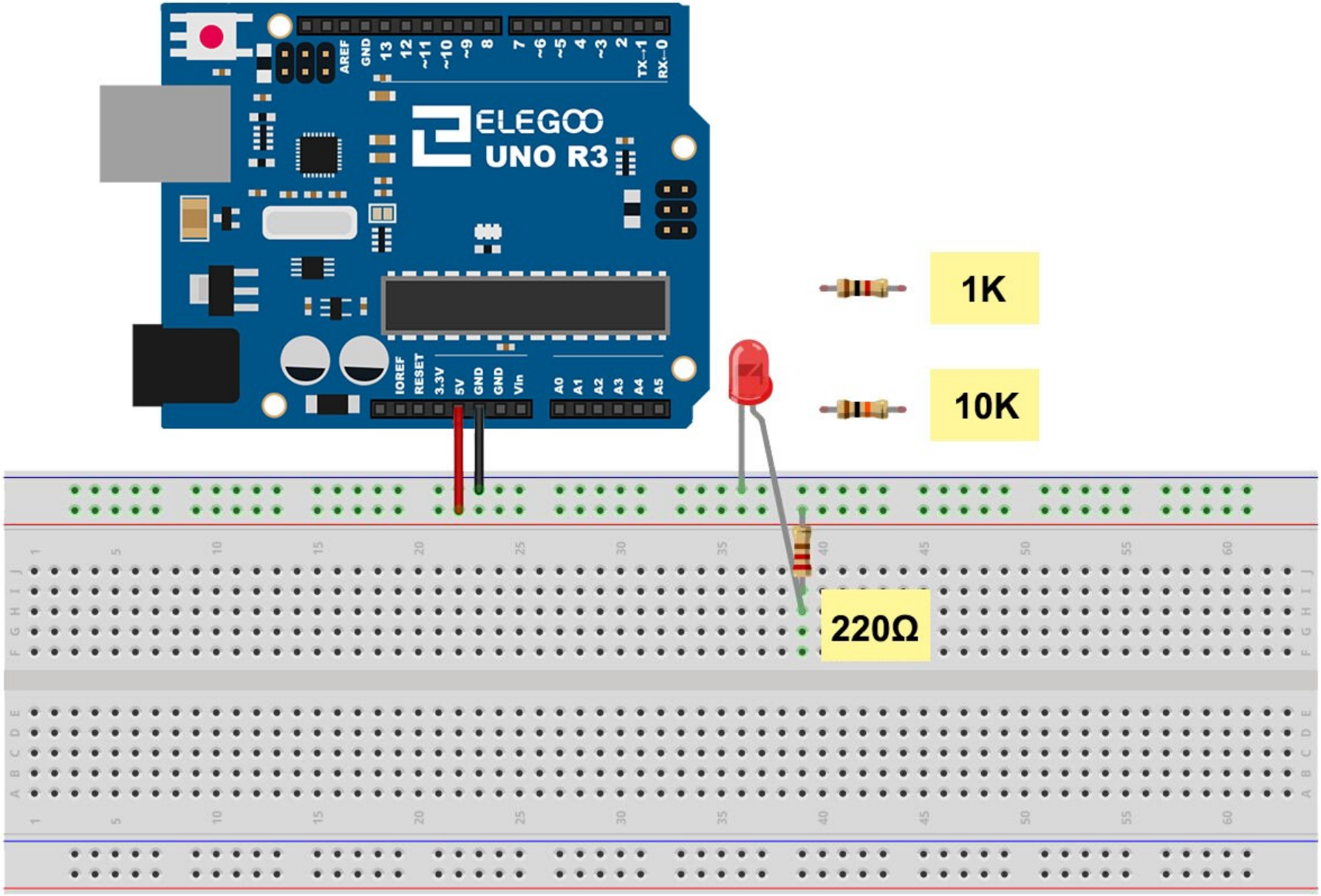


Схема монтажа

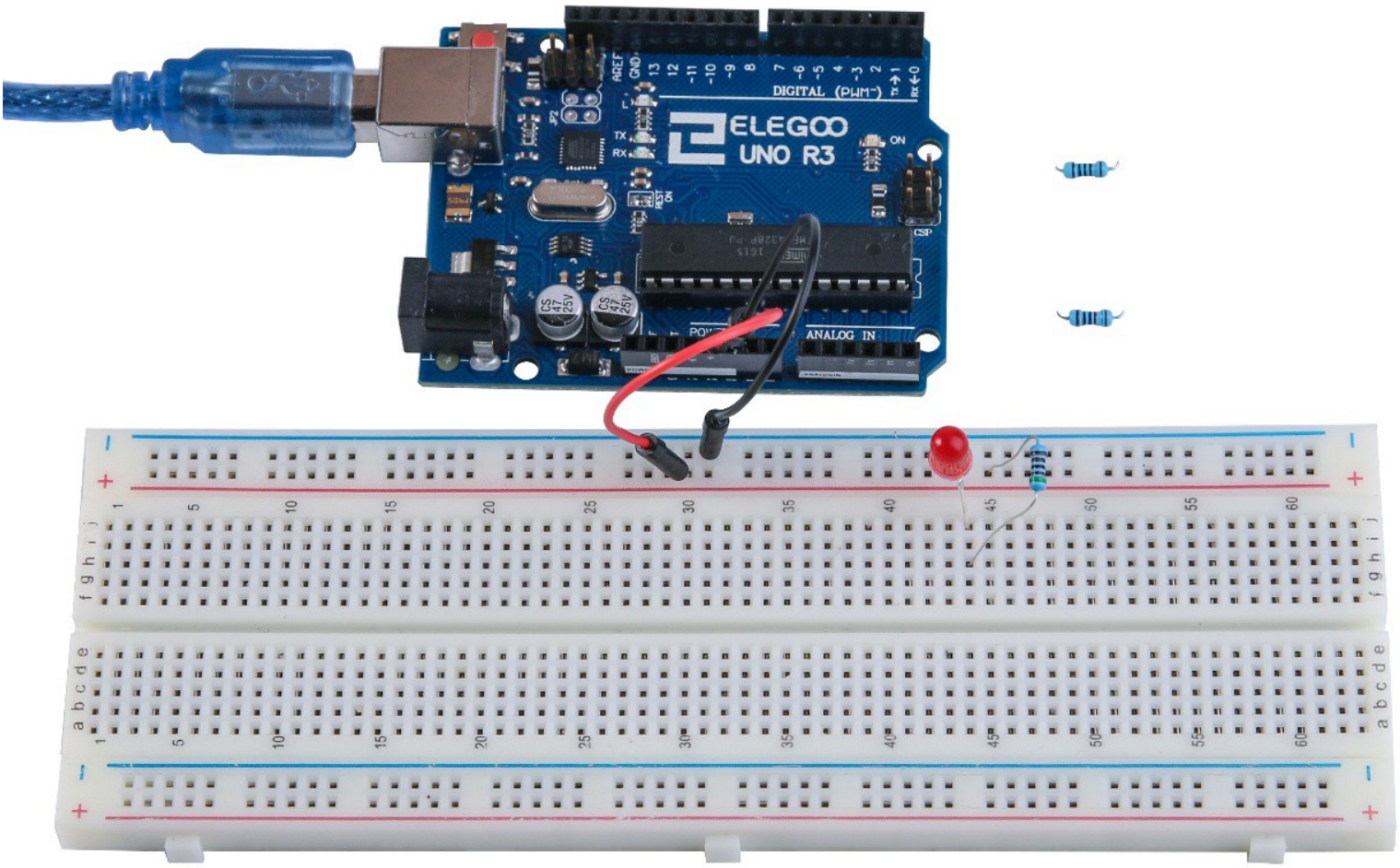


На Arduino есть пин на 5 В для питания периферийных устройств. Мы будем использовать его для питания светодиода и резистора. Больше вам от платы ничего не потребуется, только лишь подключить ее через USB к компьютеру.

С резистором на 220 Ом, светодиод должен гореть достаточно ярко. Если вы вместо резистора на 220 Ом установите резистор номиналом 1 кОм, светодиод будет гореть не так ярко. В конце концов, с резистором 10 кОм, светодиод будет еле виден. Вполне вероятно, чтобы увидеть разницу на последнем этапе вам придется вытянуть красный переходник, использовав его в качестве переключателя. Тогда вы сможете увидеть разницу в яркости.

В момент, когда к одной ноге резистора подключено 5 В, вторая нога резистора подключается к позитивной ноге светодиода, а вторая нога светодиода подключена к земле. Если мы переместим резистор так, что он будет располагаться за светодиодом, как показано ниже, светодиод все равно будет гореть. Вам, наверное, захочется вернуть резистор 220 Ом обратно на место. Не имеет значения, какой ногой подключать светодиод к плате.

Рисунок для примера



Урок 4: RGB светодиод

Обзор

RGB светодиоды – это простой и веселый способ добавить цветов в ваш проект. Поскольку внутри RGB светодиода размещается 3 обыкновенных светодиода, его использование и подключение не сильно отличается.

RGB светодиоды бывают 2х типов: с общим анодом и с общим катодом.

В случае подключения RGB светодиода с общим анодом, анод подключаем к "+5 В" на плате Arduino, а в случае подключения RGB светодиода с общим катодом – длинный вывод светодиода подключаем к GND платы.

Как и любой светодиод, подключать RGB светодиод необходимо через резистор (суммарно, 3 шт.), чтобы ограничить значения протекающих токов.

В нашем скетче мы начнем со светодиода, светящегося красным светом, затем переключимся на зеленый, затем на синий, а затем – снова на красный. Таким образом, мы пройдемся по всем цветовым вариантам.

Необходимые компоненты:

(1) х плата Elegoo Uno R3

(1) х макетная плата на 830 точек

(4) х перемычки (типа “папа”-“папа”)

(1) х RGB светодиод

(3) х резистора (220 Ом)

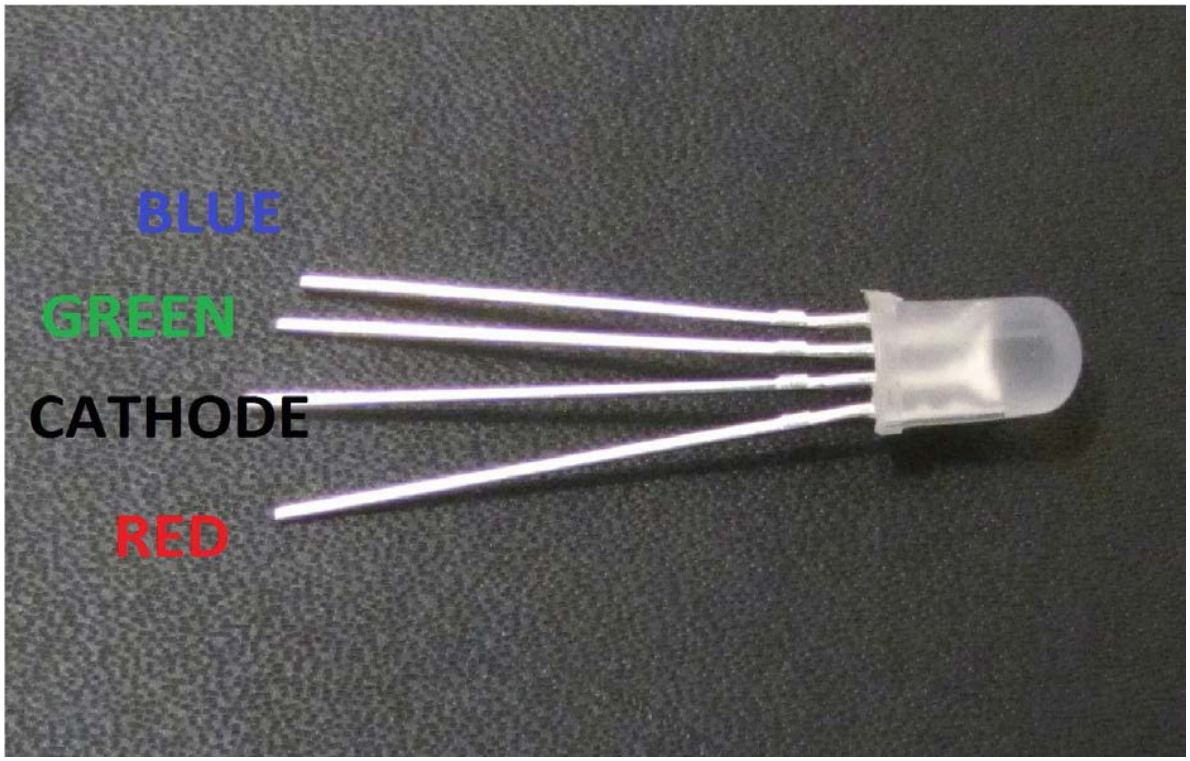
Представление компонентов

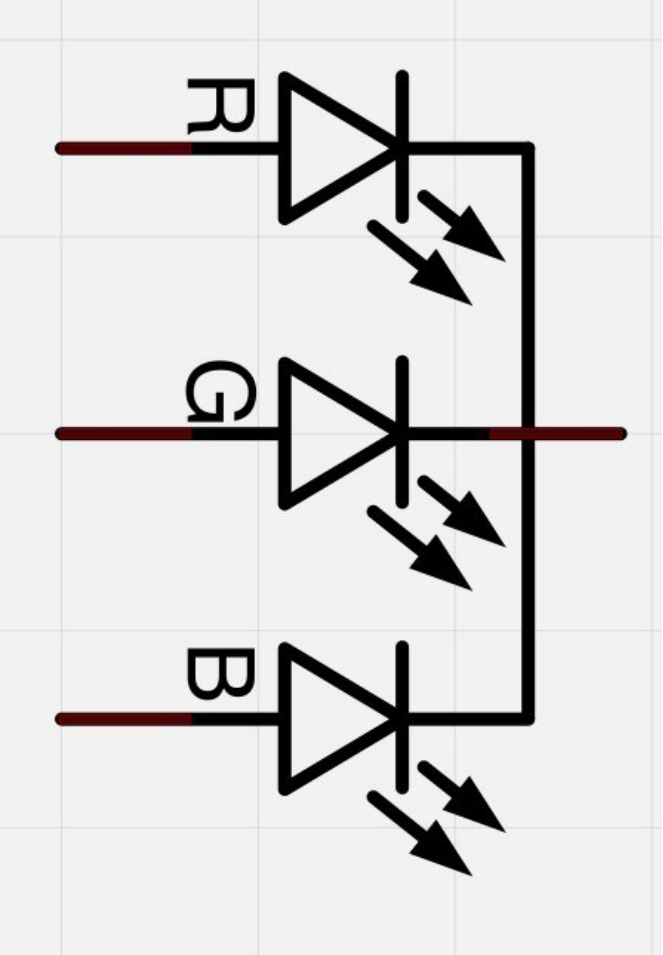
RGB светодиод:

На первый взгляд, RGB (Red, Green and Blue) светодиоды выглядят так же, как и обычные светодиоды, но на самом деле у них внутри установлено три светодиода: один красный, один зеленый и да, один синий. Управляя яркостью каждого из них, вы можете управлять цветом светодиода.

То есть, мы будем регулировать яркость каждого светодиода и получать нужный цвет на выходе, как будто это палитра художника. Для этого можно использовать переменные резисторы, как мы делали в Уроке 2, но в результате схема будет достаточно сложной! К счастью, Arduino предлагает нам функцию analogWrite. Если задействовать на плате контакты, отмеченные символом «~», мы можем регулировать напряжение, которое подается на соответствующий светодиод.

У RGB светодиода четыре ноги. По одному позитивному контакту на каждый светодиод и один общий контакт, к которому подключаются все отрицательные полюса светодиодов (аноды).





На фотографии показан светодиод с 4-мя контактами. Каждая отдельная ножка для зеленого, голубого или красного цвета называется анодом. Он всегда подключается к пину +5 В. Катод заземляется (GND). Если подключите наоборот – светодиод не загорится.

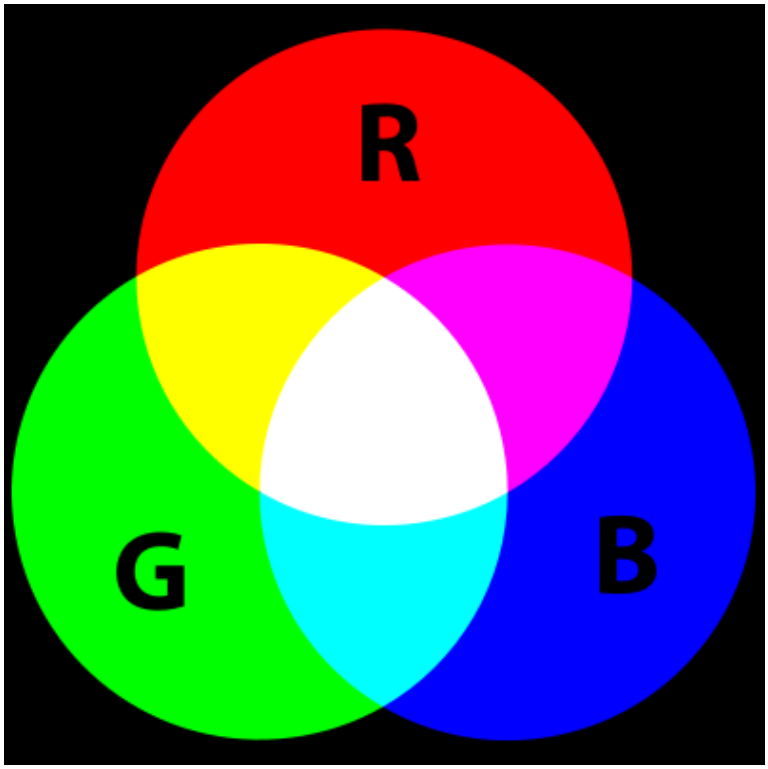
Общий анод на RGB светодиоде – это второй по счету, самый длинный контакт. Этот контакт мы подключим к земле (GND).

Для каждого светодиода нужен собственный резистор на 220 Ом, чтобы предотвратить возможность протекания чересчур больших токов. Эти резисторы устанавливаются в цепь между катодами (красный, зеленый и синий) и управляющими пинами на нашем UNO.

Цвета:

Мы можем смешивать три основных цвета и видеть новые оттенки, так как в наших глазах три типа рецепторов (для красного, зеленого и синего цветов). В результате ваш глаз и мозг обрабатывает информацию о насыщенности этих трех цветов и преобразовывает их в другие оттенки спектра.

То есть, используя одновременно три светодиода, мы словно обманываем наши глаза. Эта же идея используется в телевизорах, где жидкокристаллический дисплей состоит из маленьких точек красного, зеленого и синего цветов, которые расположены очень близко друг к другу и формируют отдельные пиксели.



Если мы настроим одинаковую яркость всех светодиодов, мы он будет светиться белым. Если мы отключим синий светодиод и будут гореть с одинаковой яркостью только красный и зеленый, мы получим желтый свет.

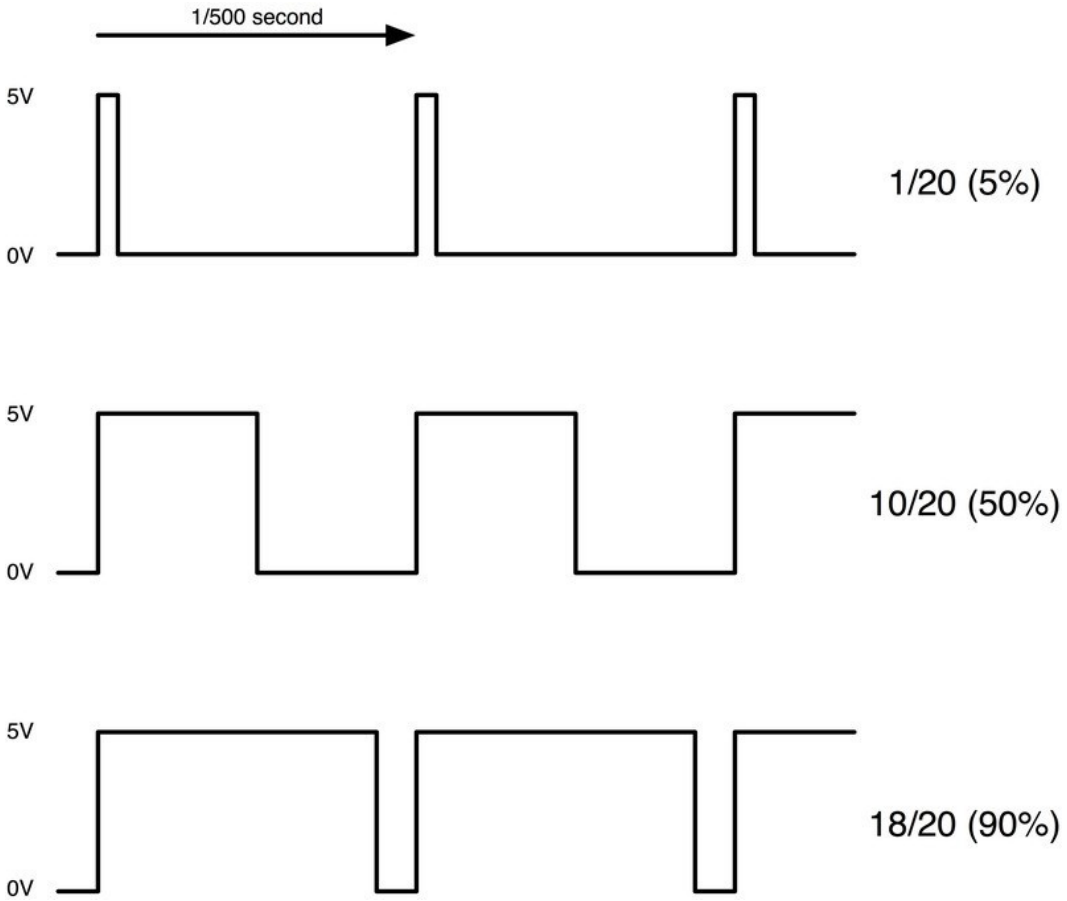
Мы можем управлять яркостью каждого светодиода отдельно, смешивая цвета как нам заблагорассудится.

Так как черный цвет не что иное, как отсутствие света, получить его не получится. Ближайший оттенок черного – это полностью выключенные светодиоды.

Теория (ШИМ)

Широтно-Импульсная Модуляция (ШИМ (PWM на английском)) – это один из методов управления питанием. В нашем случае ШИМ используется для управления яркостью каждого отдельного светодиода.

На рисунке ниже схематично изображен сигнал с одного из ШИМ пинов UNO.



Каждую 1/500 секунды ШИМ выход генерирует импульс. Длина этого импульса контролируется функцией 'analogWrite'. То есть, 'analogWrite(0)' не будет генерировать никакого импульса, а 'analogWrite(255)' сгенерирует сигнал, который будет длится до самого начала следующего. То есть, будет создаваться впечатление, что подается один непрерывный импульс.

Когда в пределах функции analogWrite мы указываем значение в диапазоне от 0 до 255, мы генерируем импульс определенной длительности. Если длина импульса составляет 5%, мы подадим на указанный выход Arduino 5% от максимально

доступного питания и создается впечатление, что светодиод горит не на максимальную яркость.

Однако, если длина импульса будет составлять 90%, тогда на указанный выход Arduino поступит 90% максимально доступного питания. На такой скорости мы не увидим, как светодиод загорается и гаснет, поэтому нам будет казаться, что происходит просто изменение яркости.

Подключение

Схема

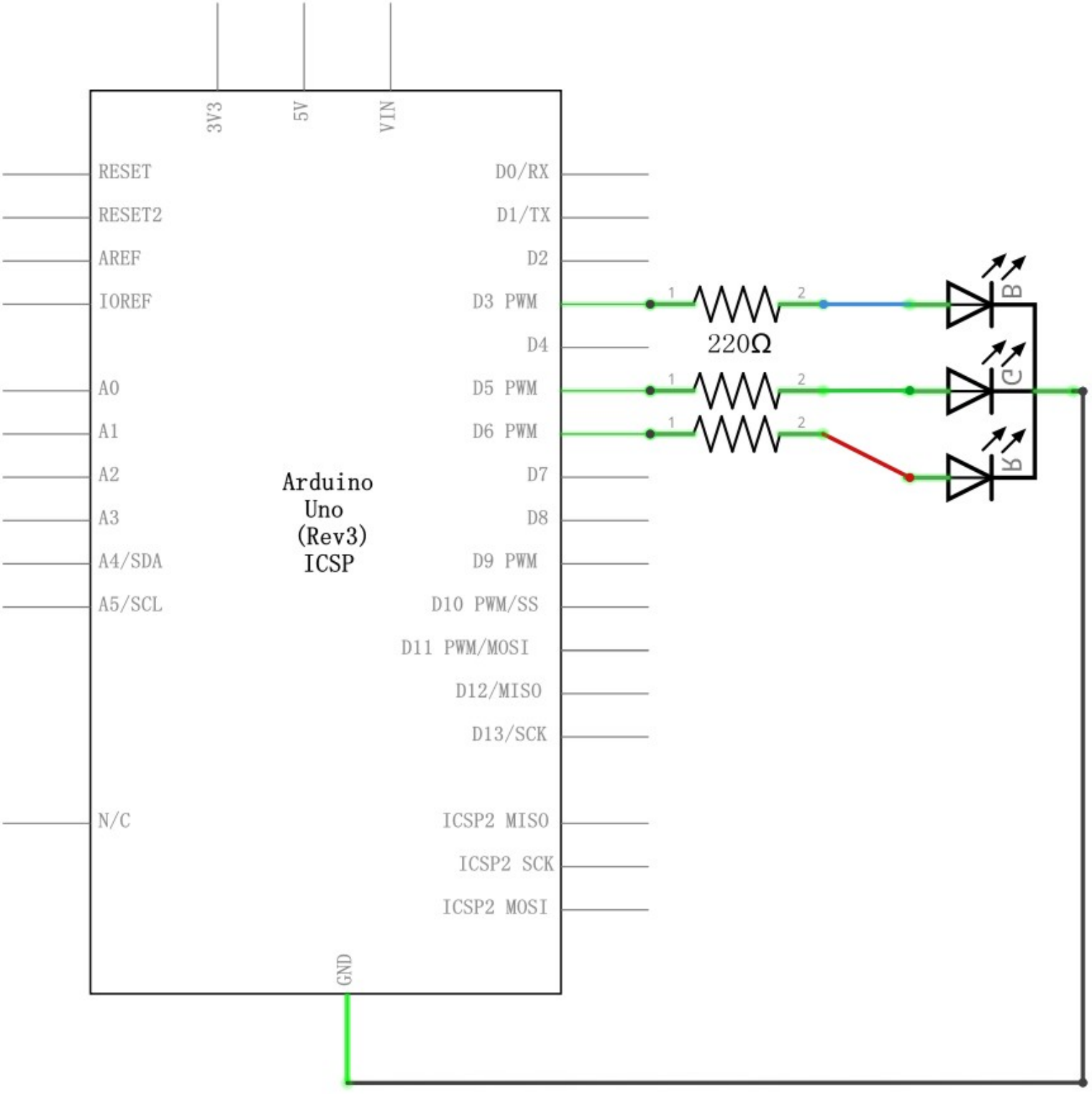
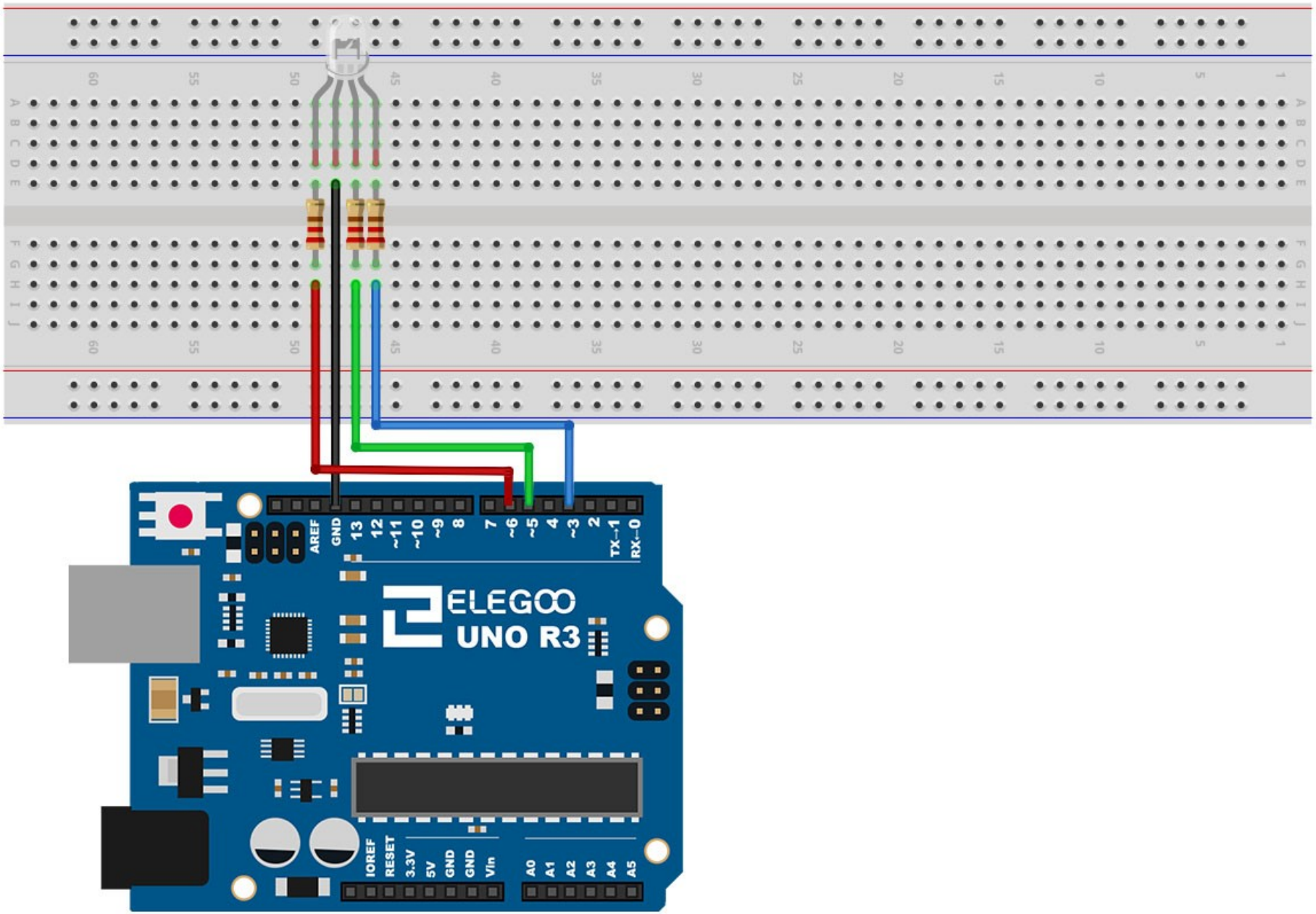


Схема монтажа



Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок4: RGB светодиод” (Lesson 4 RGB LED), и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

В нашем коде будет использоваться цикл FOR для перебора цветов. Первый цикл FOR для перехода от КРАСНОГО к ЗЕЛЕНОМУ.
Второй цикл FOR для перехода от ЗЕЛЕНОГО к СИНЕМУ.
Последний цикл FOR для перехода от СИНЕГО к КРАСНОМУ.
Попробуйте запустить этот скетч. Особенности скетча раскрыты ниже…

Скетч начинается с указания пинов, которые используются для каждого отдельного цвета:

```
// Инициализация пинов
#define BLUE 3
#define GREEN 5
#define RED 6
```

Следующий шаг – функция 'setup'. Эта функция отрабатывает один раз после запуска Arduino. В нашей программе в пределах этой функции мы инициализируем три пина, которые мы будем использовать в качестве выходов.

```
void setup()
{
pinMode(RED, OUTPUT); pinMode(GREEN, OUTPUT); pinMode(BLUE, OUTPUT);
digitalWrite(RED, HIGH); digitalWrite(GREEN, LOW); digitalWrite(BLUE, LOW);
}
```

Перед тем как рассмотреть функцию 'loop', давайте рассмотрим последнюю функцию в скетче.

Определенные переменные

redValue = 255; // выберите значение между 1 и 255 для смены цвета. greenValue = 0;

blueValue = 0;

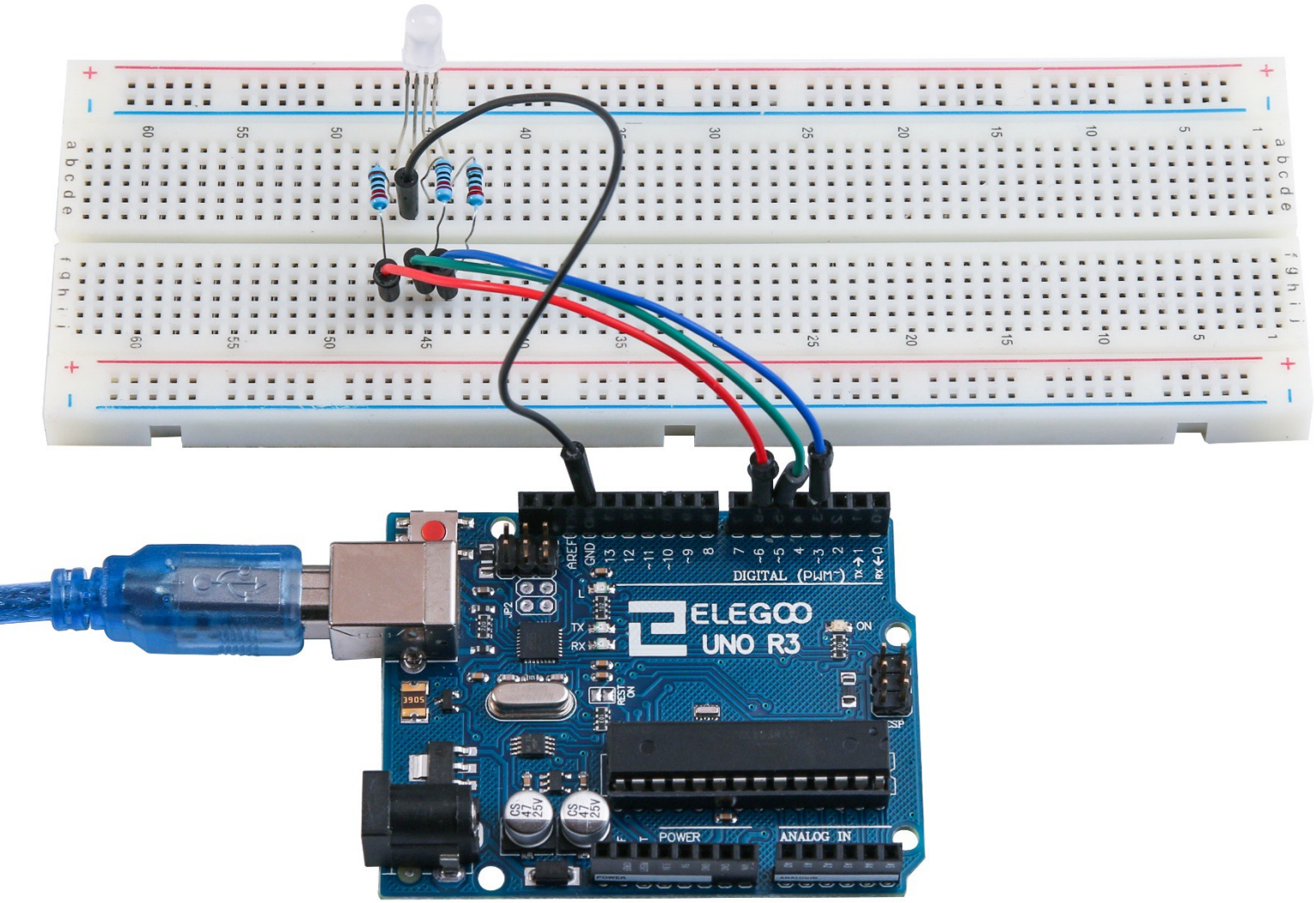
Эта функция принимает три аргумента. Один отвечает за яркость красного, зеленого и синего светодиодов. В каждом из них значение варьируется в диапазоне от 0 до 255, где 0 означает, что он полностью выключен, а 255 – максимальная яркость. После этого функция вызывает 'analogWrite', чтобы настроить яркость каждого светодиода.

Если вы взглянете на функцию 'loop', вы увидите, что в ее пределах мы настраиваем яркость красного, зеленого и синего оттенков, которые мы хотим отобразить и потом делаем паузу на 1 секунду, прежде чем перейти к другому цвету.

#define delayTime 10 // длительность «затухания» цвета Delay(delayTime);

Попробуйте добавить несколько собственных цветов в скетч.

Рисунок для примера



Урок 5: Цифровые порты (Подключение кнопки)

Обзор

В этом уроке вы научитесь использовать тактовые кнопки в подключении к цифровым портам для включения/выключения светодиода.

Нажатие на одну из кнопок будет включать светодиод, нажатие на другую – выключать.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x макетная плата на 830 точек

(1) x красный светодиод (5 мм)

(1) x резистор (220 Ом)

(2) x тактовые кнопки

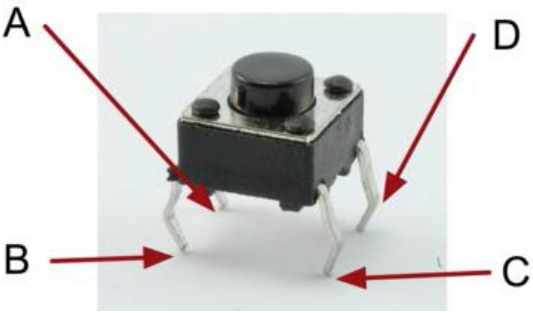
(7) x перемычки DuPont (типа “папа” - “папа”)

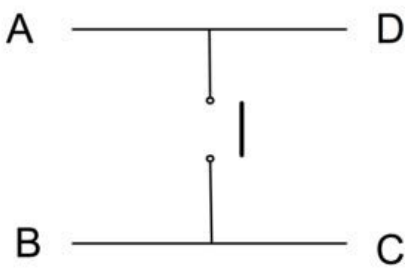
Представление компонентов

Переключатели:

Переключатель (также, ключ/выключатель) – очень простой электрический компонент. При нажатии на кнопку или щелчке рычагом, происходит замыкание электрической цепи, т.е. переключатель пропускает ток.

Маленькие тактовые кнопки, используемые в этом уроке, имеют 4 контакта, что может сбить с толку.





На самом деле, у кнопки всего два электрических контакта. Пины B и C соединены между собой, так же, как пины A и D внутри корпуса кнопки.

Подключение

Схема

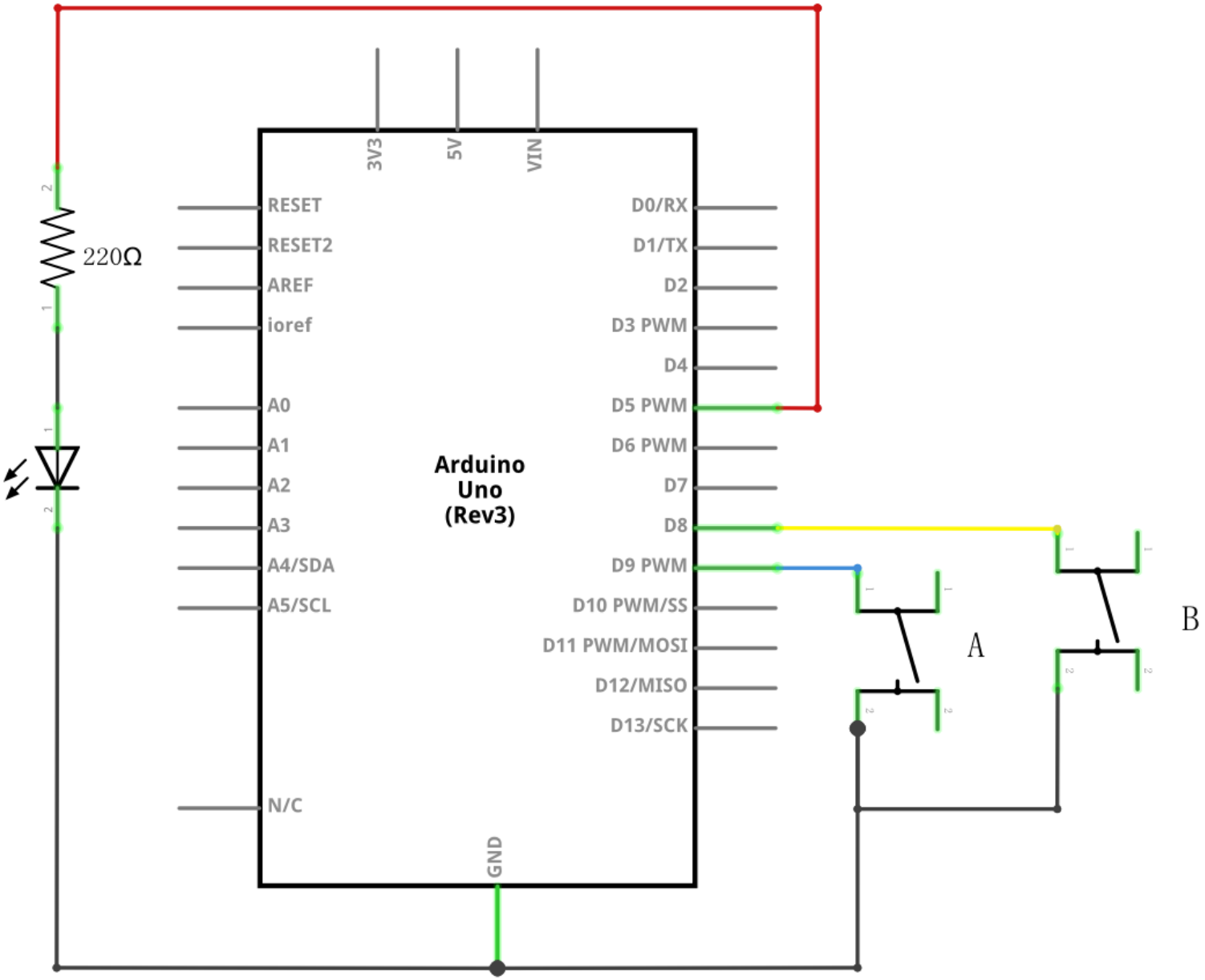
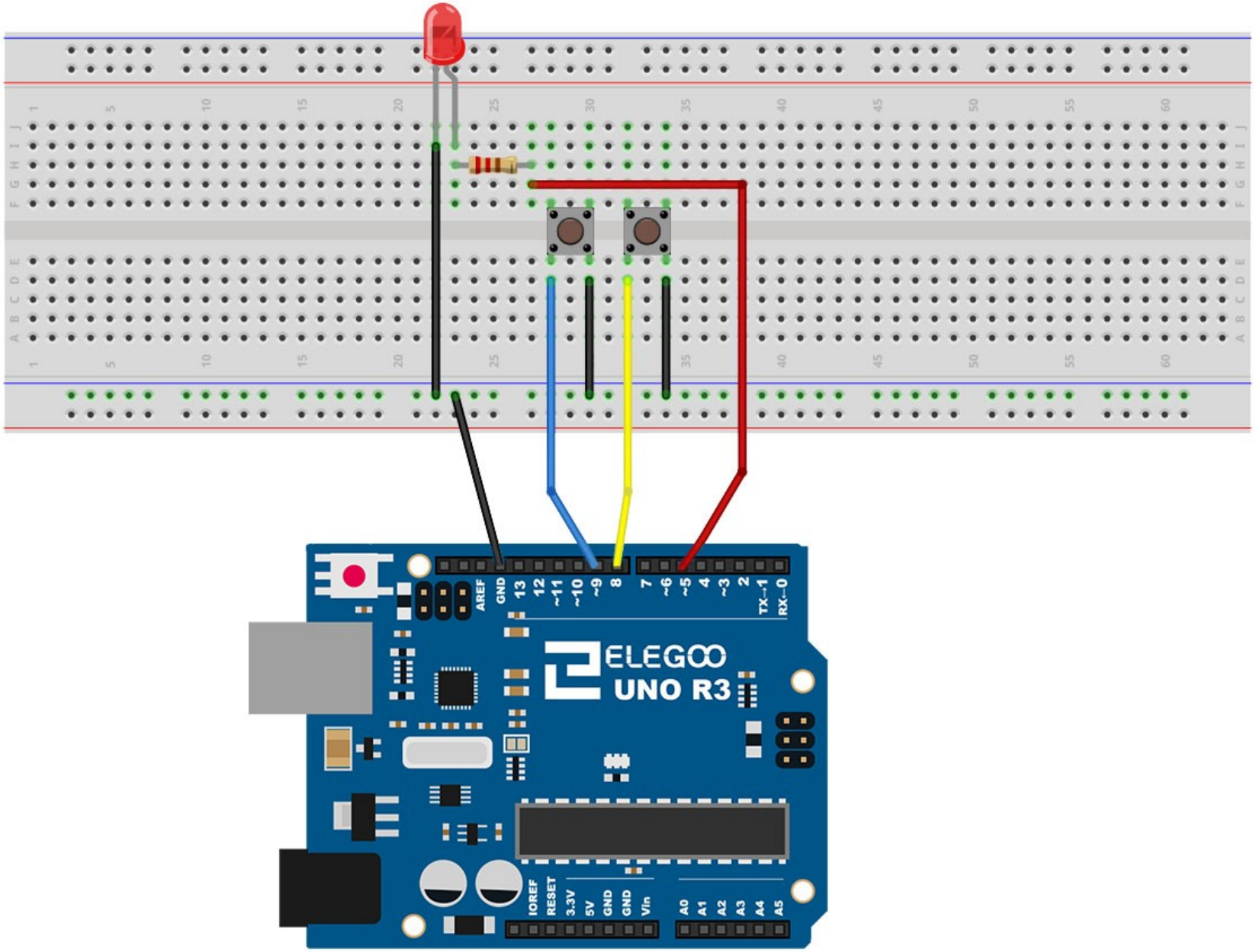


Схема монтажа



Обратите внимание, что контакты кнопки “вылезают” за пределы ее корпуса. Это значит, что контакты будут находится на достаточном расстоянии друг от друга только при их правильном расположении на макетной плате.

Помните, что короткий вывод светодиода – катод – подключается к общему проводу.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 5: Цифровые порты”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Загрузите скетч на вашу плату UNO. Нажатие левой кнопки будет включать светодиод, а нажатие правой кнопки – выключать.

Первая часть скетча инициализирует три переменные, соответствующие трем пинам, которые мы будем использовать. 'ledPin' – пин, подключенный к светодиоду как выход. 'buttonApin' – пин, подключенный к кнопке, находящейся ближе к верху макетной платы как вход, а 'buttonBpin' – пин, соответственно, подключенный ко второй кнопке.

Функция 'setup' обычно определяет ledPin как OUTPUT, но в данной ситуации у нас два пина, работающих на вход. В таком случае, мы устанавливаем pinMode на “INPUT_PULLUP“, как показано ниже:

```
pinMode(buttonApin, INPUT_PULLUP); pinMode(buttonBpin, INPUT_PULLUP);
```

“INPU T_PULLUP“ означает, что пин будет использоваться как вход со значением “HIGH”, устанавливаемым по умолчанию, до момента нажатия на кнопку, когда значение переключается на “LOW“.

Именно поэтому переключатели подключаются к общему проводу (GND). При нажатии кнопки, входной пин подключается к общему проводу (GND), и его значение больше не соответствует “HIGH”.

Поскольку вход обычно соответствует “HIGH” и переключается на “LOW“ только при нажатии на кнопку, логика должна быть противоположной. Мы будем использовать функцию “loop”.

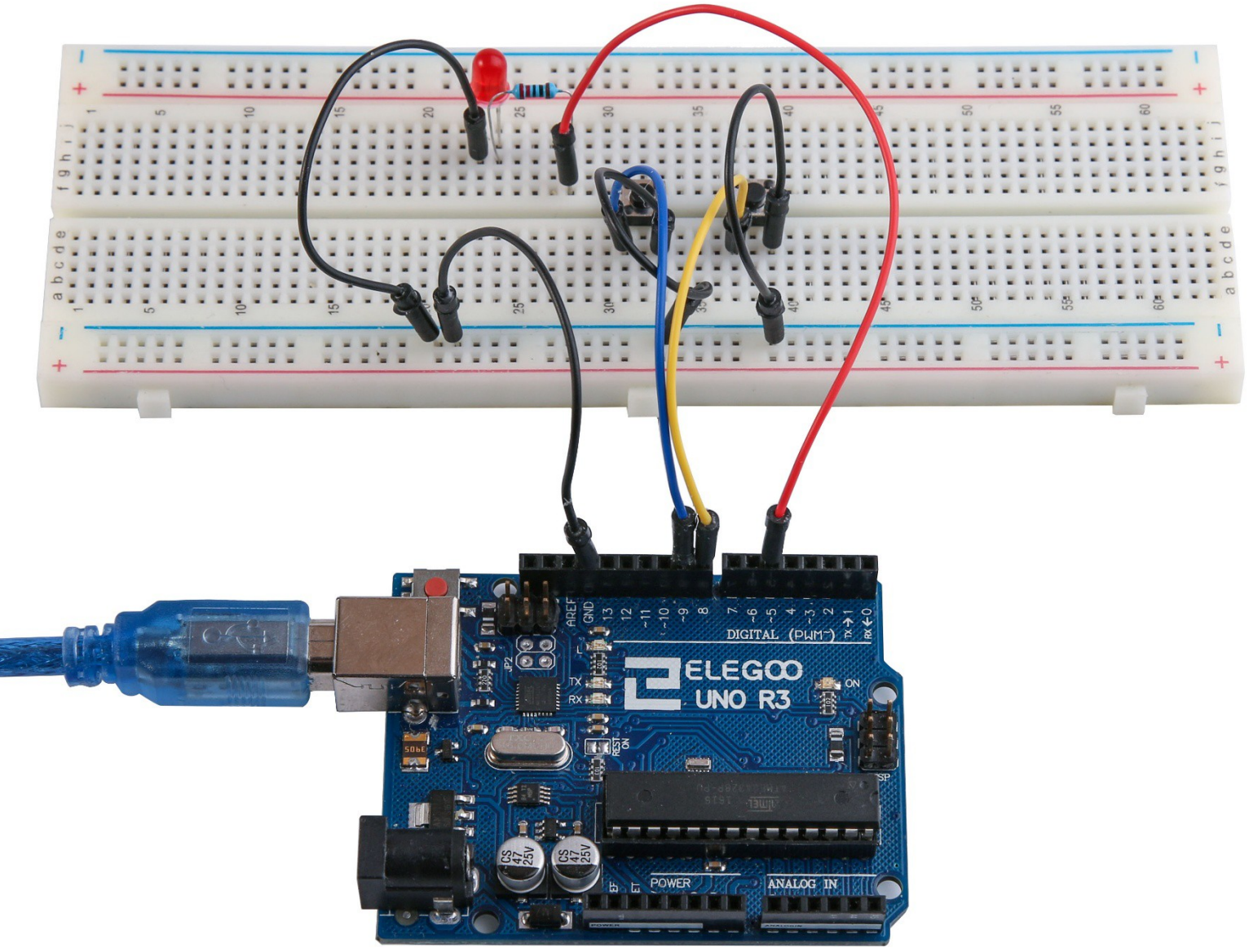
```
void loop()
{
if (digitalRead(buttonApin) == LOW)
{
digitalWrite(ledPin, HIGH);
}
if (digitalRead(buttonBpin) == LOW)

{
digitalWrite(ledPin, LOW);
}
}
```

Функция “loop” содержит два условия “if”. По одному на кнопку. Каждое из них выполняет “digitalRead” на соответствующем входе.

Помните, что если кнопка нажата, то соответствующий ввод будет “LOW”. Если кнопка A находится в состоянии “LOW”, то 'digitalWrite' на ledPin включит светодиод. Аналогично, если нажата кнопка B, то в ledPin запишется “LOW”.

Рисунок для примера



Урок 6: Пьезоизлучатель со встроенным генератором (активный зуммер)

Обзор

В этом уроке вы научитесь создавать звук, используя активный зуммер.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x активный зуммер

(2) x перемычки DuPont (типа “мама” - “папа”)

Представление компонентов

Зуммер:

Электрический зуммер – это сигнальное устройство, питающееся постоянным током и оснащенное интегральной схемой. Зуммеры широко используются в компьютерах, принтерах, ксероксах, будильниках, электронных игрушках, электронной аппаратуре подвижных объектов, телефонах, таймерах и другой электронике. Зуммеры бывают активными и пассивными. Чтобы понять, переверните два зуммера контактами вверх. Зуммер с зеленой печатной платой – пассивный, в то время как зуммер с прилагаемой черной изолентой – активный. Разница заключается в том, что в активном зуммере содержится генератор, генерирующий звук при включении в цепь. В пассивном зуммере нет генератора, поэтому он не будет пищать при подключении к источнику питания. Вместо этого необходимо использовать импульсы прямоугольной формы с частотой в приделах 2 – 5 кГц, чтобы зуммер начал «генерировать» звук. Зачастую, активный зуммер дороже, поскольку может содержать несколько встроенных колебательных контуров.

Подключение

Схема

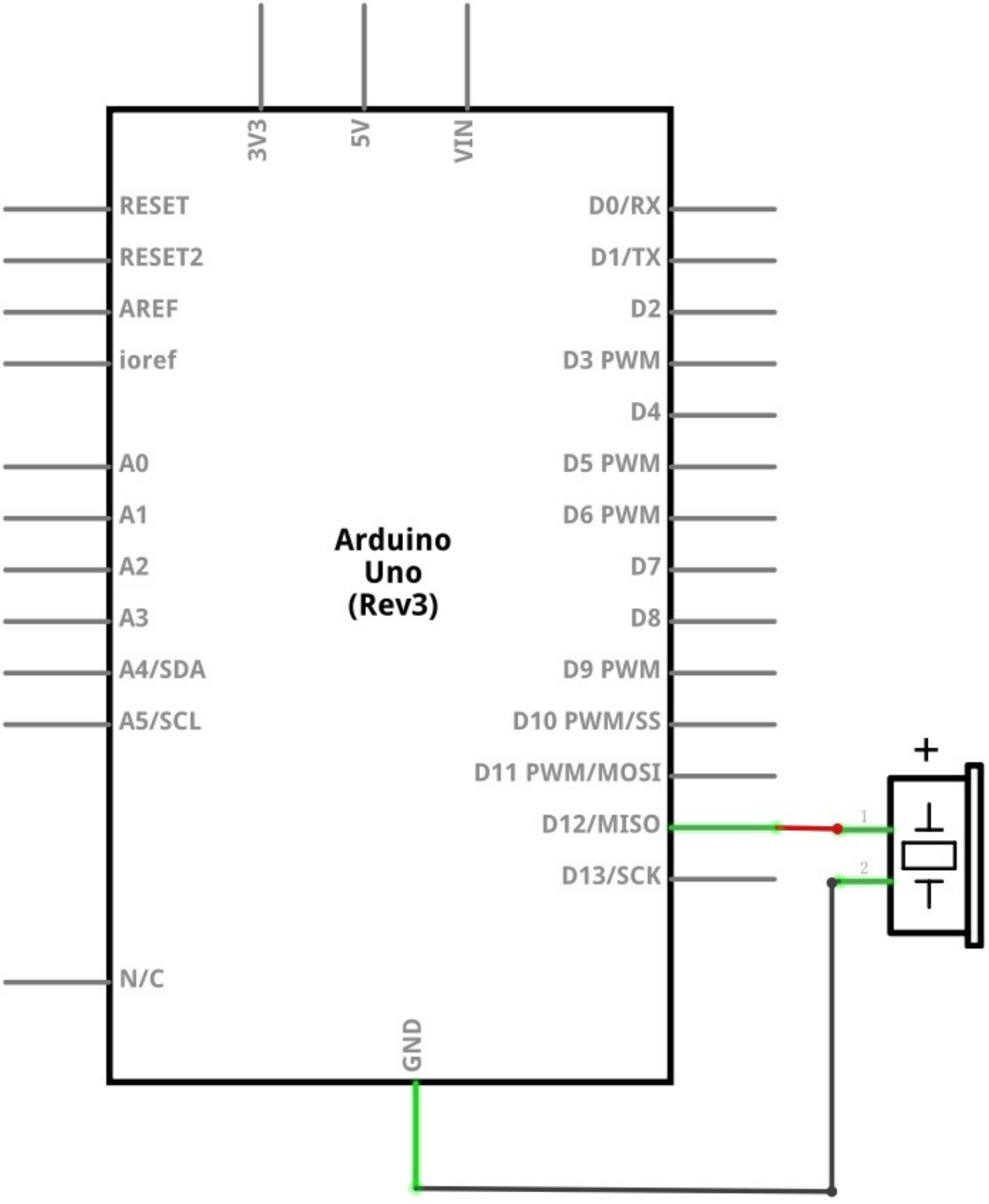
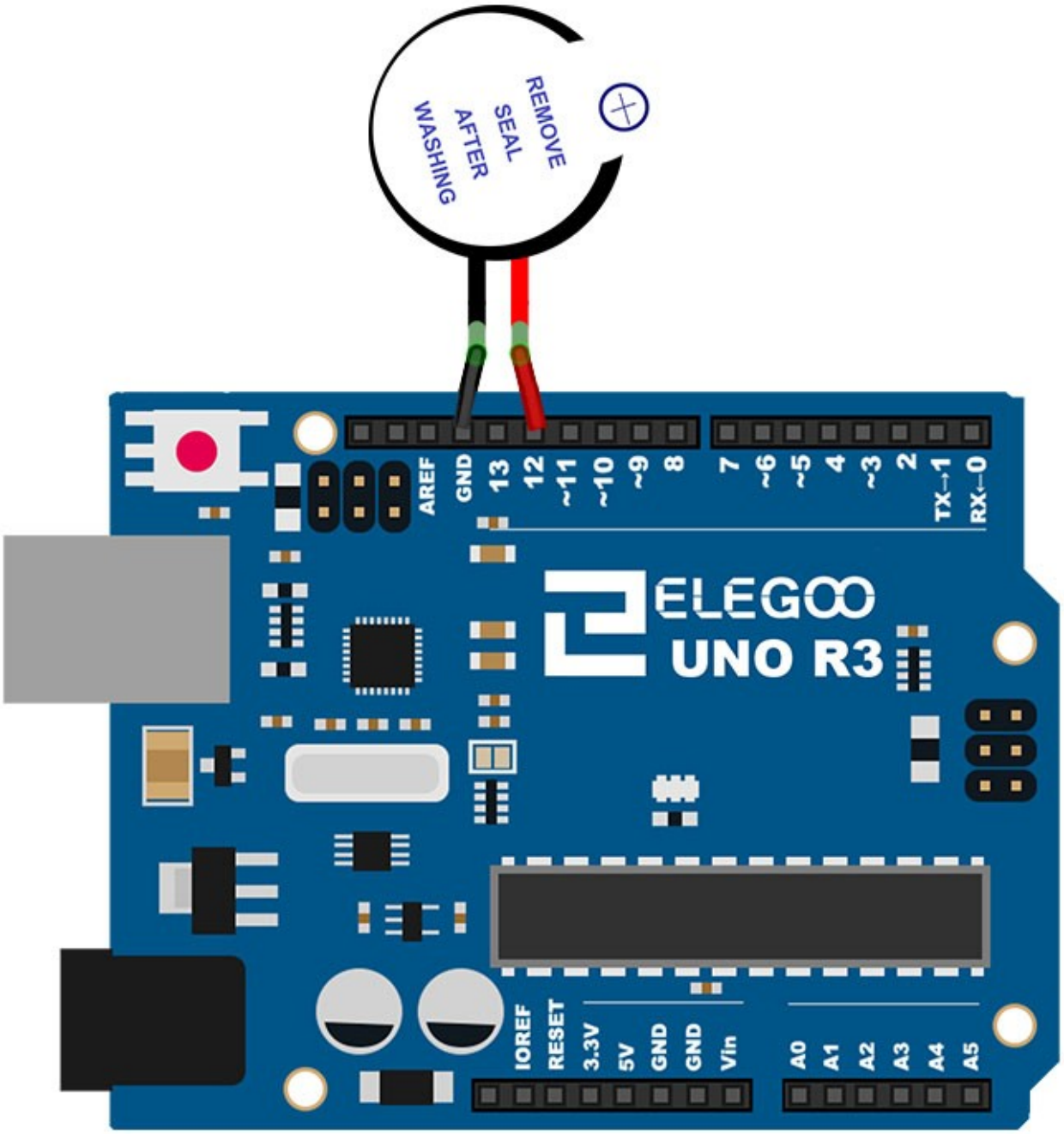


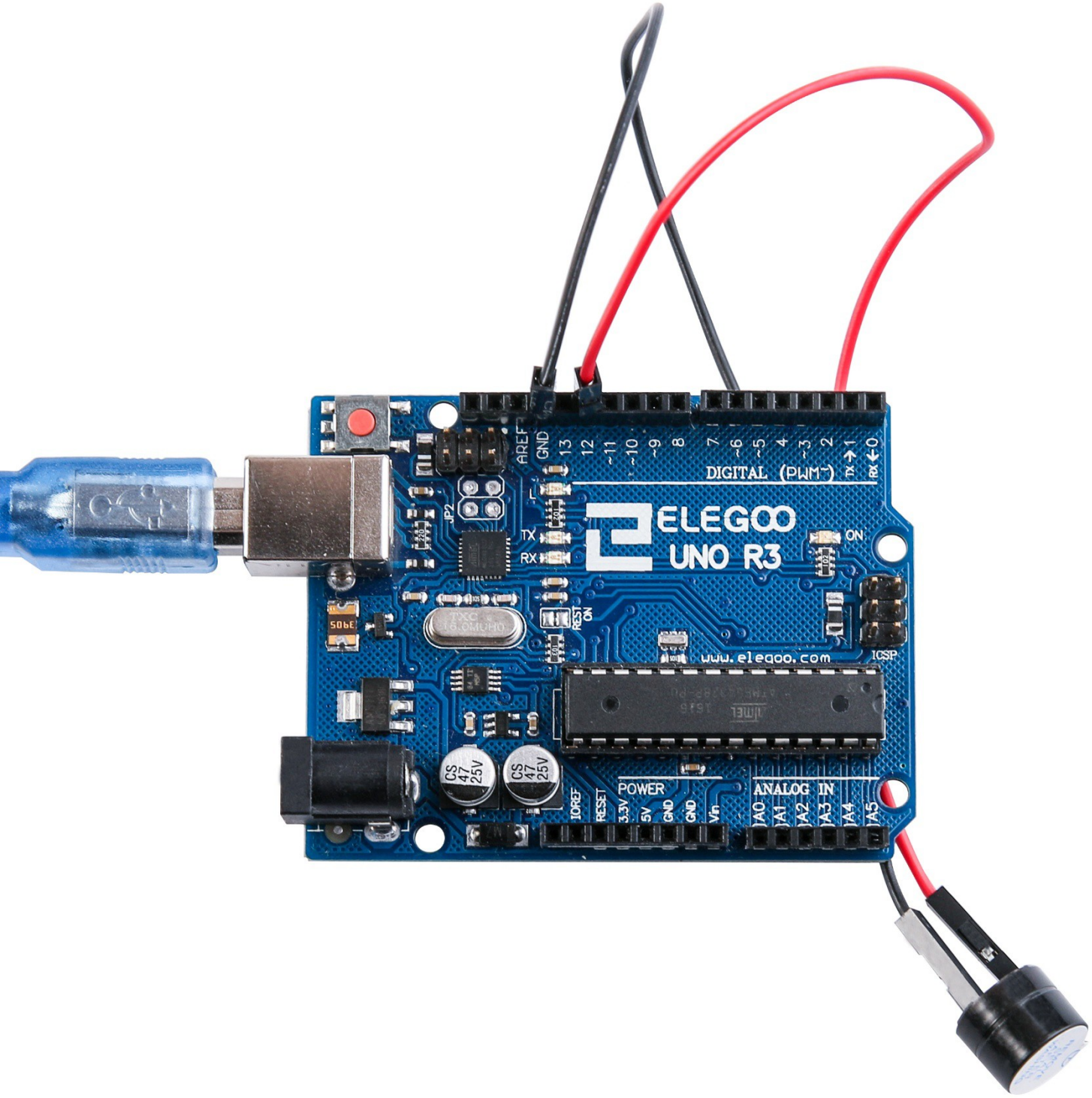
Схема монтажа



Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 6: Пьезоизлучатель со встроенным генератором (активный зуммер)”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Рисунок для примера



Урок 7: Датчик наклона

Обзор

В этом уроке вы научитесь, как использовать датчик наклона для определения малых углов наклона.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x датчик наклона

(2) x перемычки DuPont (типа “мама” - “папа”)

Представление компонентов

Датчик наклона:

Датчики наклона позволяют определять ориентацию или наклон. Они компактные, недорогие, низковольтные и просты в использовании. При правильной эксплуатации, они не выйдут из строя. Простота и неказистость датчиков наклона сделала их популярным выбором для игрушек, гаджетов и разных приборов. В литературе также встречается термин «ртутный переключатель».

Датчик наклона представляет собой герметичную колбу, как правило, цилиндрической формы, содержащую шарик ртути и два (или более) электрических контакта. Гравитация смещает каплю ртути до самой низкой точки в оболочке. Когда переключатель наклонен в соответствующем направлении, ртуть касается контактов, замыкая цепь. Наклон переключателя в противоположном направлении наоборот, приводит к размыканию контактов.

Не очень точный и универсальный, по сравнению с полноценным акселерометром, датчик наклона всё же хорош для определения движений или ориентации. Другим преимуществом есть то, что датчики большого размера могут сами замыкать цепь. Акселерометры, с другой стороны, на выдают цифровые или аналоговые данные, которые затем должны анализироваться с использованием дополнительных схем.

Подключение

Схема

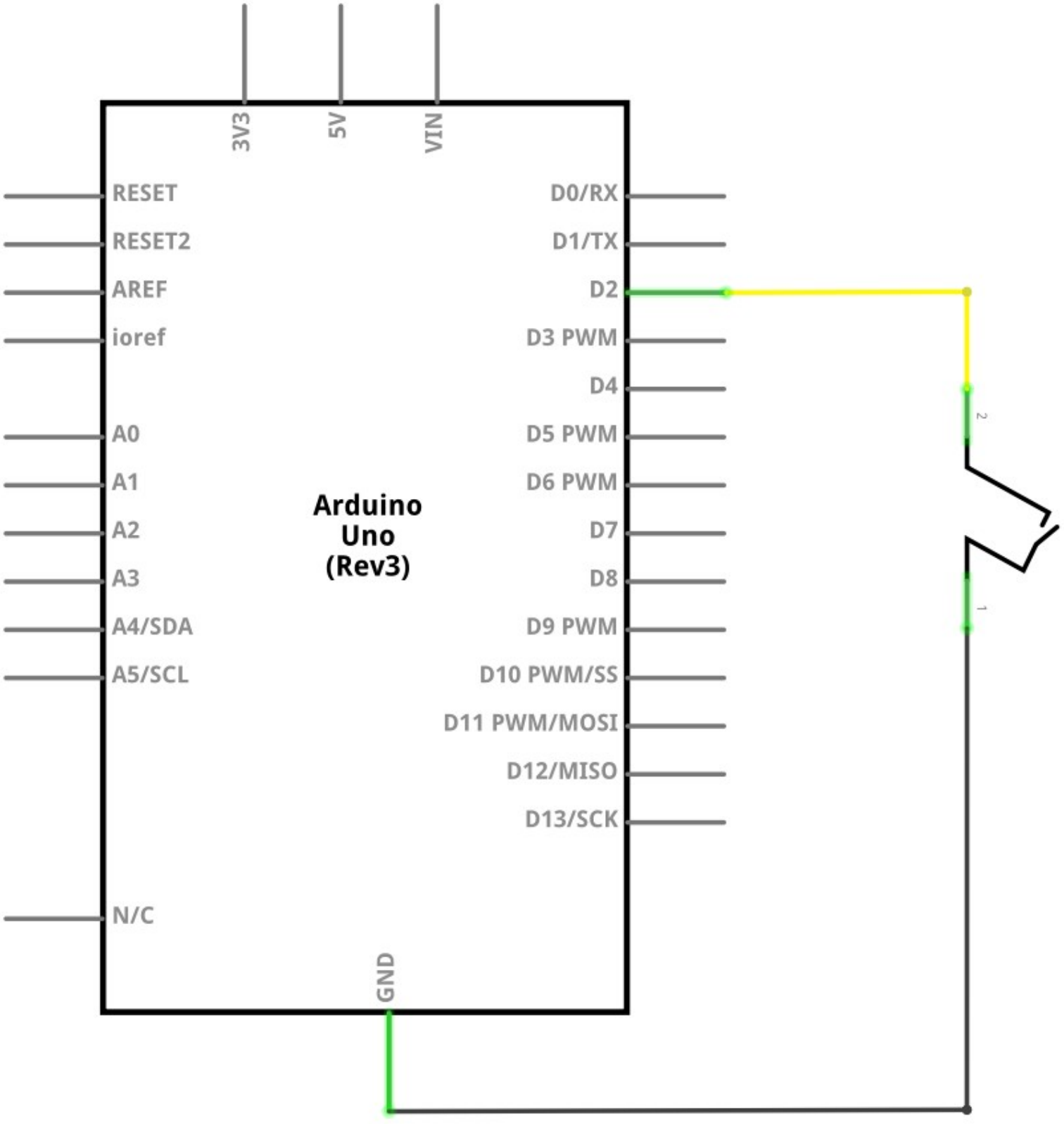
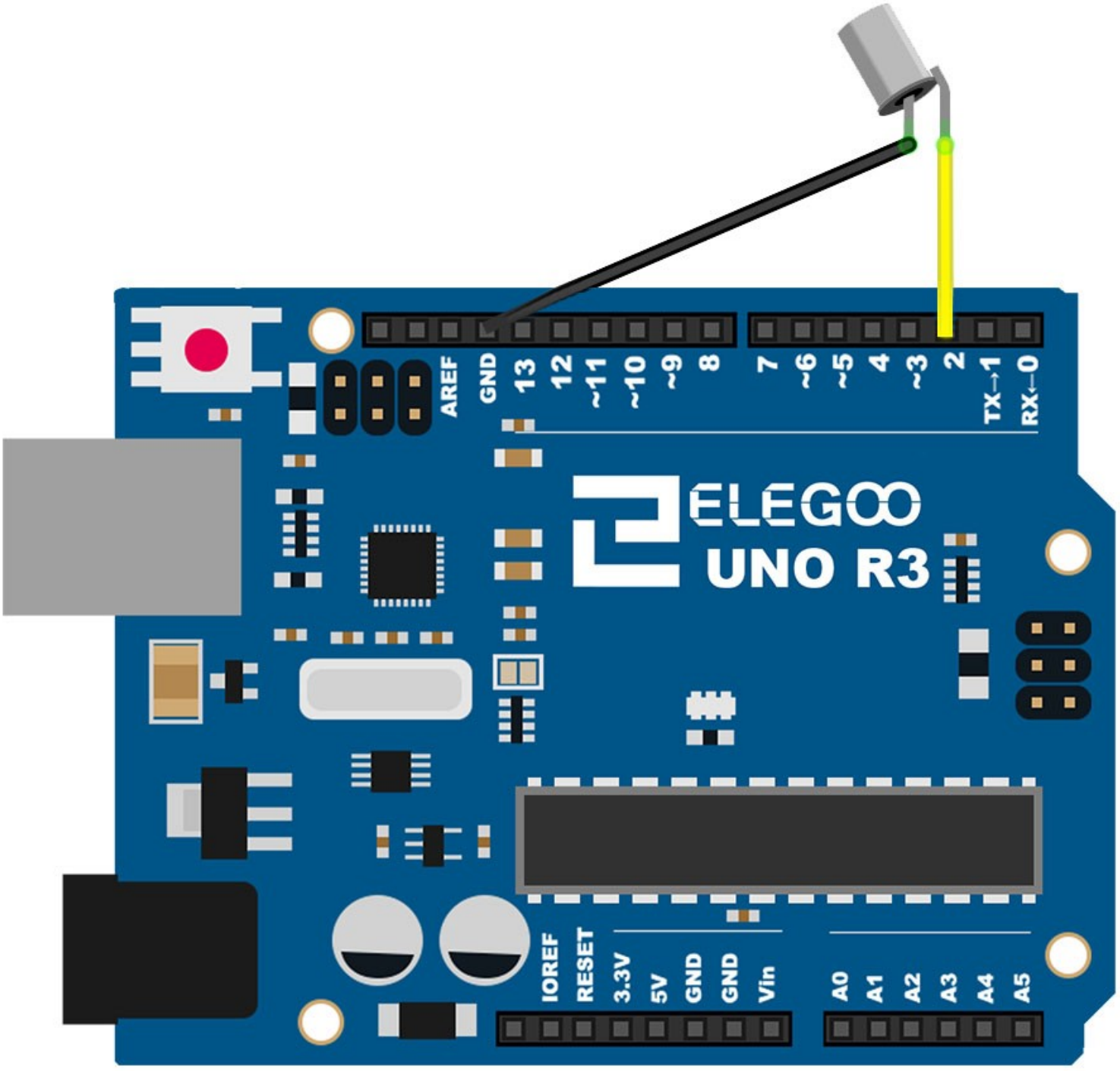


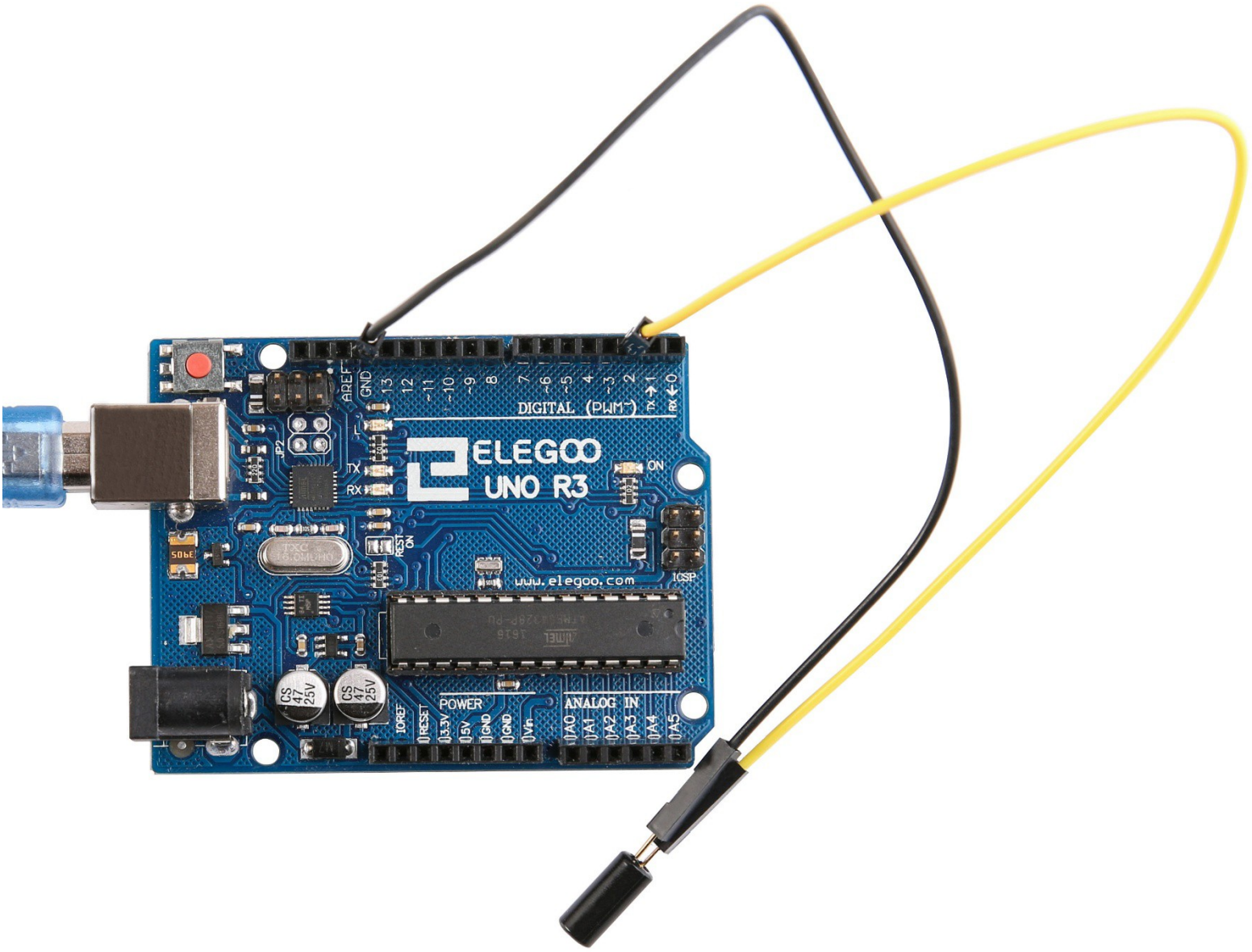
Схема монтажа



Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 8: Датчик наклона”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Рисунок для примера



Урок 8: Восемь светодиодов и сдвиговый регистр 74HC595

Обзор

В этом уроке вы научитесь использовать восемь крупных светодиодов с платой UNO без необходимости использовать все 8 пинов на вашей плате!

Конечно же, вы можете подсоединить восемь светодиодов по-отдельности к UNO, используя по резистору для каждого из них, но вы быстро заметите, что остались без свободных пинов на вашей плате. В таком подходе нет ничего плохого, но очень часто, помимо светодиодов, необходимо дополнительно подключать кнопки, сенсоры, серводвигатели и т.д. Поэтому в этом уроке мы будем использовать сдвиговый регистр 74HC595 («последовательный вход, параллельный выход»), который с помощью всего трех пинов контроллера поможет подключить до восьми устройств! Скорость управления сигналом, к сожалению, снижается до 500 000 раз в секунду вместо 8,000,000 в секунду, но это все равно очень быстро, и наш глаз не заметит разницы!

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x макетная плата на 830 точек

(8) x светодиодов

(8) x резистор (220 Ом)

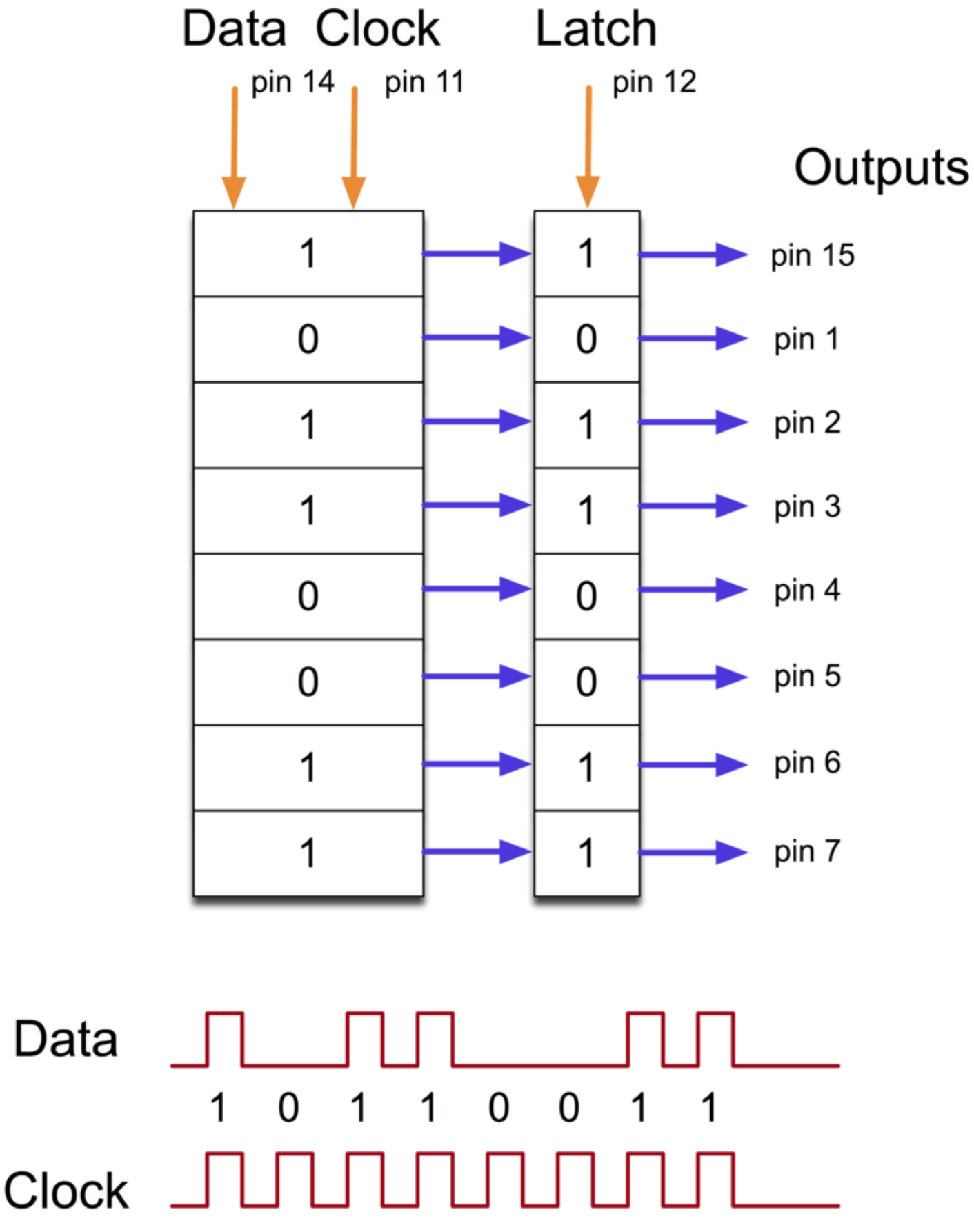
(1) x сдвиговый регистр 74HC595

(14) x перемычки (типа “папа” – “папа”)

Представление компонентов

Сдвиговый регистр 74HC595:

Сдвиговый регистр – это интегральная микросхема, которую можно представить в виде восьми ячеек памяти, каждая из которых содержит 1 либо 0. Для передачи этих значений мы используем пины микросхемы DS (Data) и SH_CP (Clock).


Data
Clock
Latch
pin 14
pin 11
pin 12
Outputs
1
0
1
1
0
0
1
1
1
0
1
1
0
0
1
1
pin 15
pin 1
pin 2
pin 3
pin 4
pin 5
pin 6
pin 7
Data
1 0 1 1 0 0 1 1
Clock

На пин SH_CP должно поступить восемь импульсов. Для каждого импульса, если значение на пине DS соответствует ‘HIGH’, то в сдвиговый регистр «проталкивается» 1, в противоположном случае – 0. После поступления всех восьми импульсов, пин ST_CP (Latch) копирует данные в регистр хранения. Этот шаг важен во избежание включения «неправильного» светодиода при поступлении данных в регистр сдвига.

На микросхеме также имеется пин OE (Output Enable), который предоставляет разрешение на вывод данных с внешних ячеек. Вы можете подключить его к одному из пинов платы UNO с поддержкой ШИМ и использовать 'analogWrite' для регулировки яркости светодиодов. Этот пин подключается к GND.

Подключение

Схема

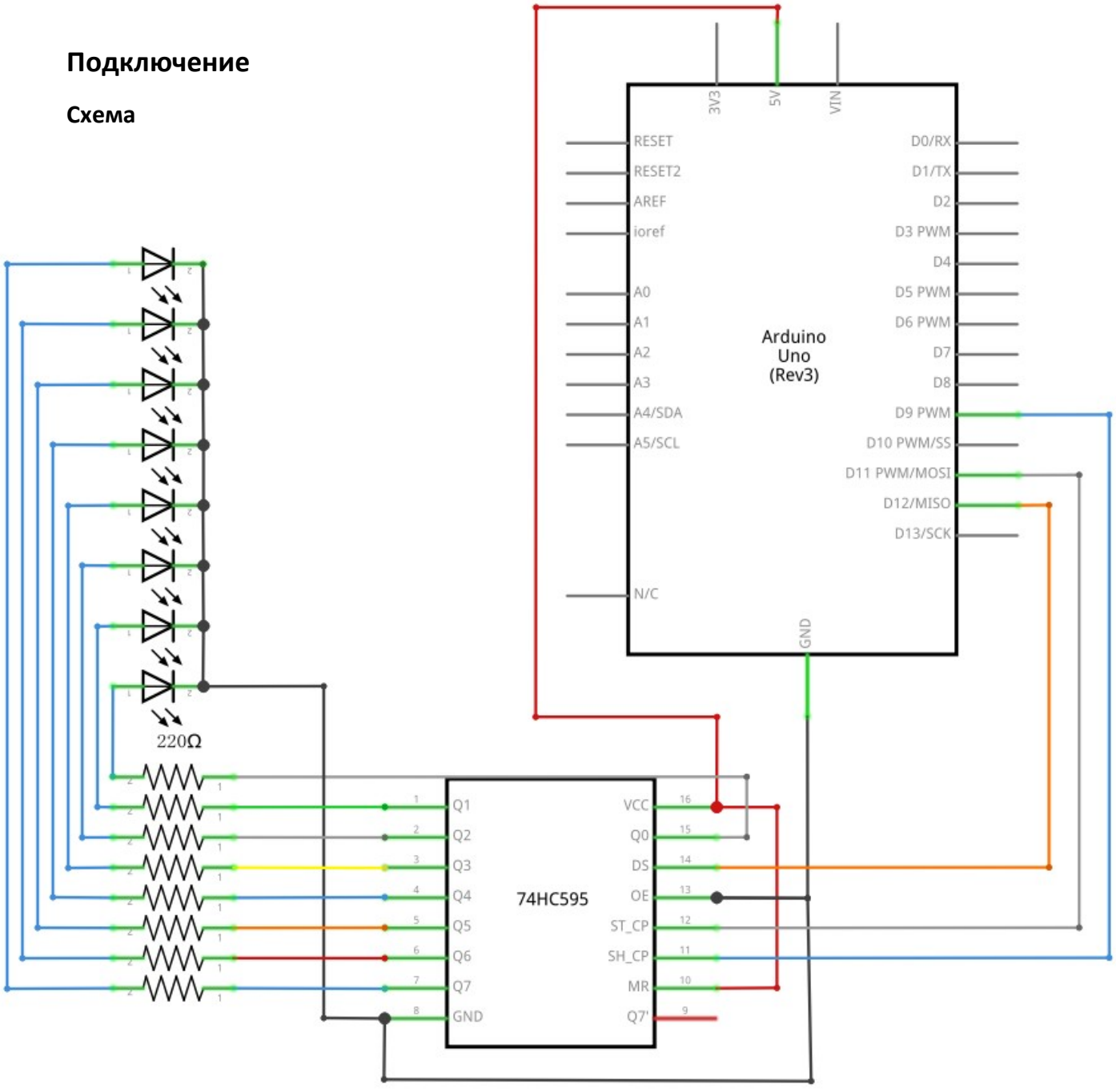
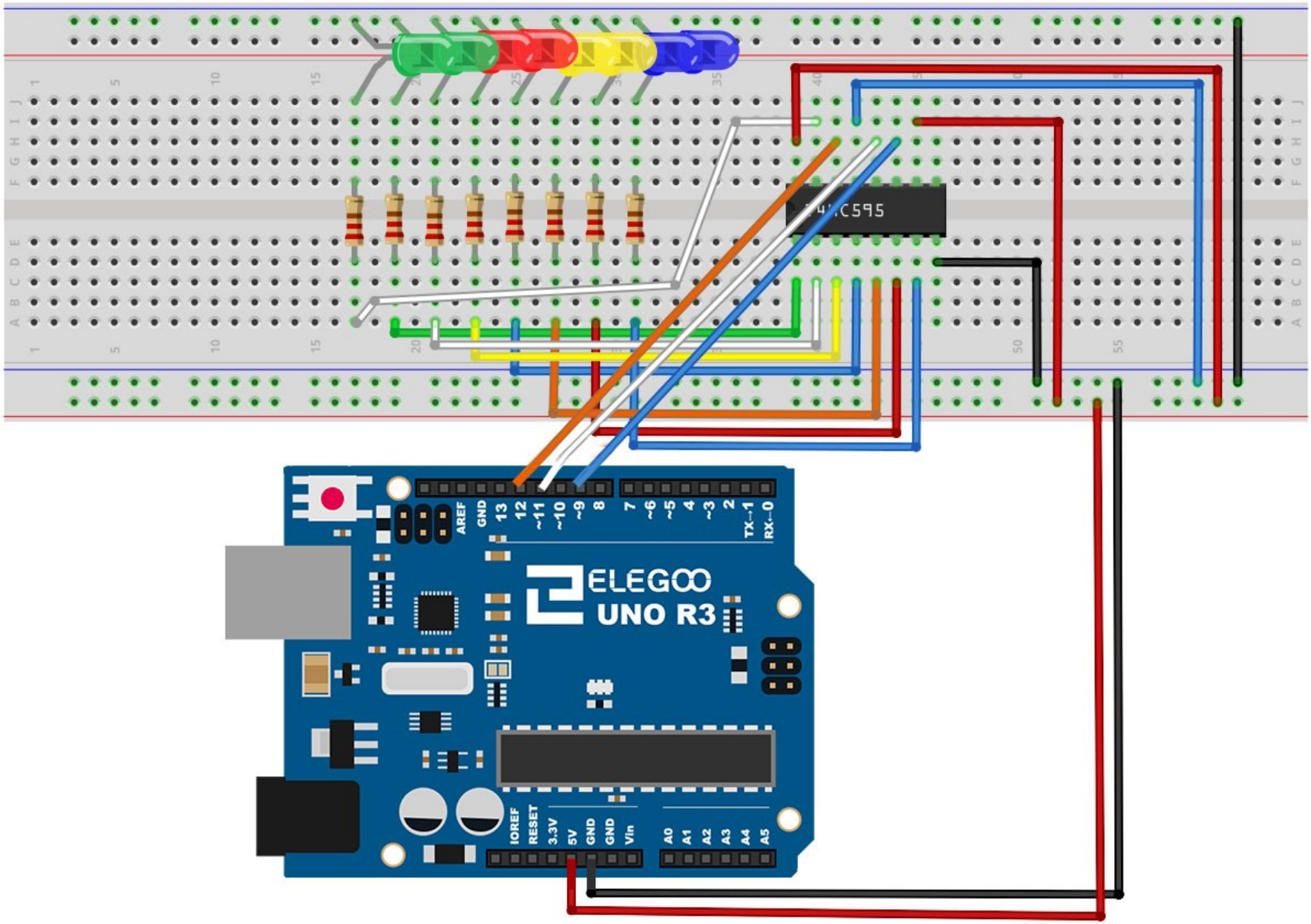


Схема монтажа



Поскольку мы используем восемь светодиодов и, соответственно, восемь резисторов, подключение всех элементов займет определенное количество времени. Наверное, проще будет начать с подключения микросхемы 74HC595, поскольку все остальное подключается к ней. Расположите чип на макетной плате так, чтобы маленькая U-образная зазубрина находилась в направлении верхнего края платы. Пин 1 микросхемы расположен слева от зазубрины.

Цифровой вывод 12 платы UNO подключается к пину 14 сдвигового регистра

Цифровой вывод 11 платы UNO подключается к пину 12 сдвигового регистра

Цифровой вывод 9 платы UNO подключается к пину 11 сдвигового регистра

Все, кроме одного, выходы микросхемы расположены слева. Таким образом, для упрощения подключения, подключим светодиоды туда же. После подключения микросхемы, перейдем к резисторам. Убедитесь, что ни один из контактов резисторов не соприкасается с другим. Перед подключением питания стоит проверить это еще раз. Если невозможно расположить резисторы так, чтобы избежать касания их контактов друг с другом, попробуйте укоротить их «ноги» так, чтобы они находились ближе к поверхности макетной платы.

Далее, подключим светодиоды длинным выводом к микросхеме. Присоедините перемычки, как показано на рисунке выше. Не забудьте о перемычке, подключающей пин 8 микросхемы к GND. Загрузите скетч и запустите его. Светодиоды будут включаться по очереди, пока все восемь на будут гореть. Затем они выключатся, и цикл повторится будет повторяться.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 24: Восемь светодиодов и сдвиговый регистр 74HC595”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Начинаем с инициализации пинов, которые мы будем использовать. Это три цифровых вывода платы UNO, которые подключены к пинам ST_CP, SH_CP и DS регистра 74HC595.

```
int latchPin = 11; int clockPin = 9; int dataPin = 12;
```

Далее определяем переменную 'leds'. Она будет хранить данные о текущих включенных и выключенных светодиодах. Данные типа 'byte' используются для 8-ми битных чисел. Каждый бит равен 1 или 0, поэтому идеально подходит для отслеживания включенных/выключенных светодиодов.

```
byte leds = 0;
```

Функция 'setup' просто устанавливает три пина, которые мы используем в качестве цифровых выводов.

```
void setup()
{
pinMode(latchPin, OUTPUT); pinMode(dataPin, OUTPUT); pinMode(clockPin, OUTPUT);
}
```

Функция 'loop' изначально выключает все светодиоды, передавая значение 0 переменной 'leds'. Затем она вызывает 'updateShiftRegister', которая передаст данные из 'leds' в регистр сдвига для выключения всех светодиодов. Мы будем рассматривать работу 'updateShiftRegister' позже.

Функция 'loop' останавливается на полсекунды, затем начинает вести счет от 0 до 7, используя цикл 'for' и переменную 'i'. Каждый раз для установки бита, контролирующего светодиод в переменной 'leds', используется функция 'bitSet'. Она также вызывает 'updateShiftRegister', чтобы светодиоды обновили свое состояние в соответствии с содержанием переменной 'leds'. Затем следует полусекундная задержка перед инкрементом переменной 'i', и загорается следующий светодиод.

```
void loop()
{
leds = 0; updateShiftRegister(); delay(500);
for (int i = 0; i < 8; i++)
{
bitSet(leds, i); updateShiftRegister(); delay(500);
}
}
```

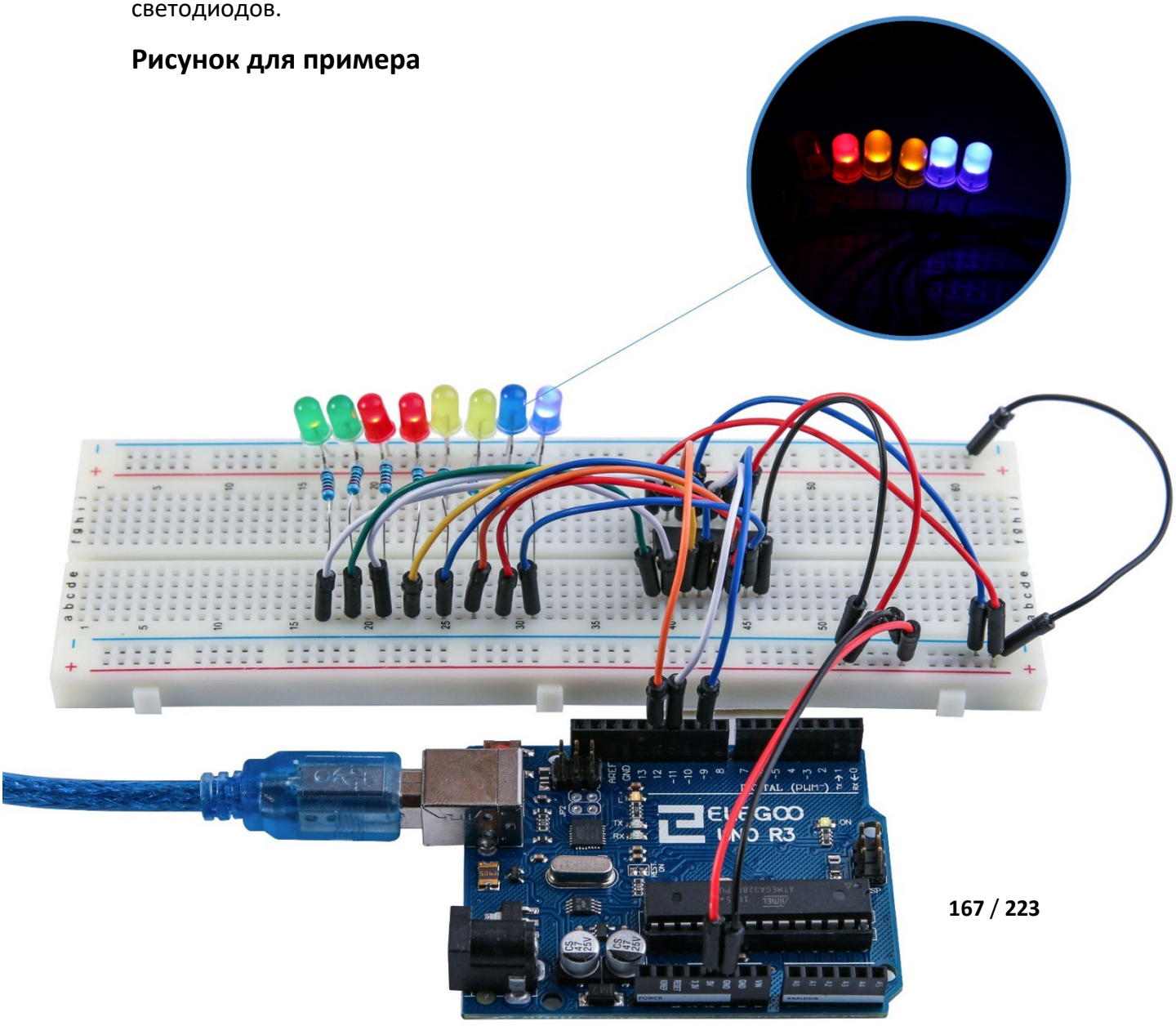
Функция 'updateShiftRegister', в первую очередь, устанавливает значение LOW в latchPin, затем вызывает функцию 'shiftOut' перед повторной установкой значения

HIGH в 'latchPin'. Эта процедура использует 4 параметра, из которых первые два – пины для DS и SH_CP. Третий параметр указывает, с какого конца данных вы хотите начать. Мы начнем со старшего правого бита – «наименее значащего бита» (Least Significant Bit). Последний параметр – это данные для сдвига в сдвиговом регистр – в нашем случае –'leds'.

```
void updateShiftRegister()
{
digitalWrite(latchPin, LOW); shiftOut(dataPin, clockPin, LSBFIRST, leds);
digitalWrite(latchPin, HIGH);
}
```

Если вы хотите, наоборот, выключать светодиоды, используйте аналогичную функцию (bitClear) с переменной 'leds'. Она установит в биты 'leds' значения 0, и вам затем надо будет вызывать 'updateShiftRegister' для «обновления» состояния светодиодов.

Рисунок для примера



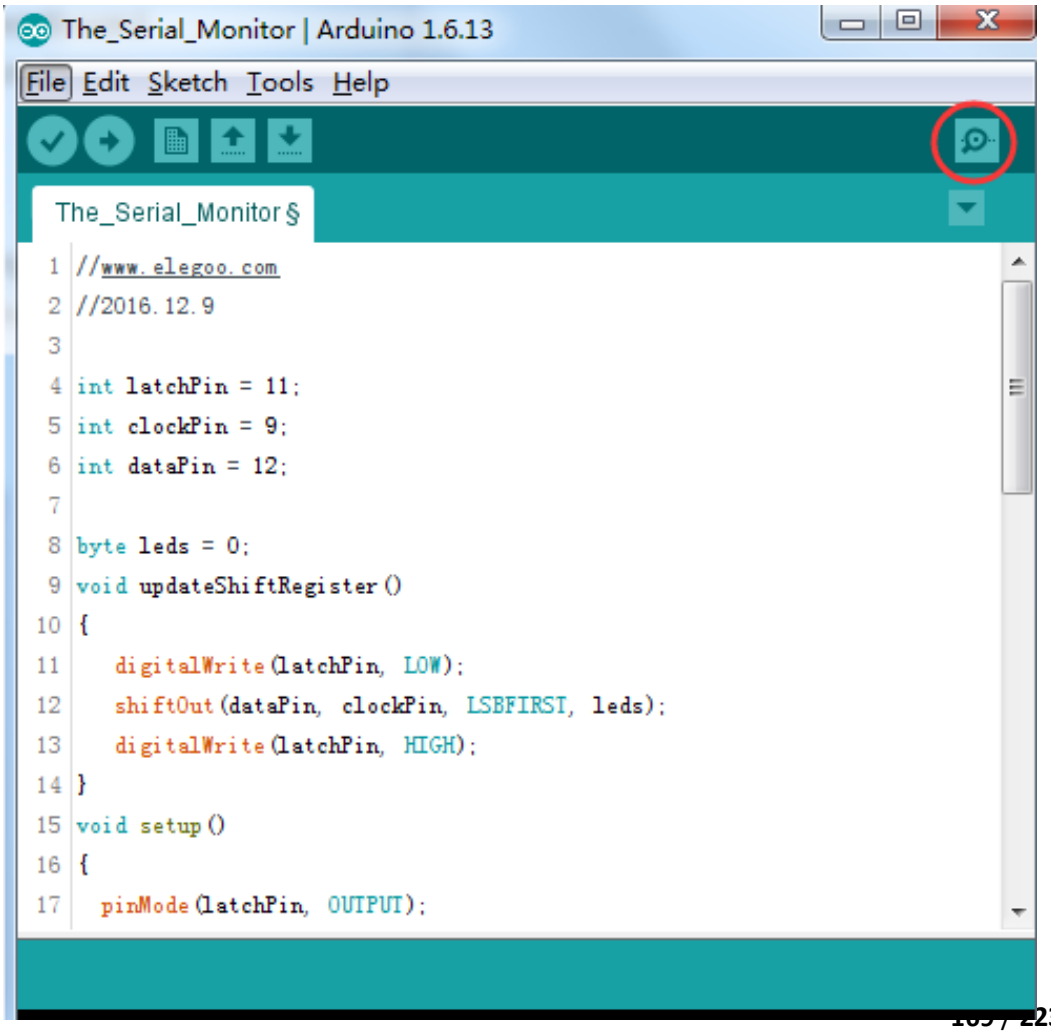
Урок 9: Серийный монитор

Обзор

Этот урок является продолжением Урока 8, но, в дополнение к вышеизложенному, вы также научитесь контролировать светодиоды напрямую с вашего компьютера, используя серийный монитор (Serial Monitor) Arduino. Серийный монитор – это «связывающее звено» между компьютером и платой UNO. Он позволяет вам отправлять и получать текстовые сообщения, удобен для отладки и позволяет вас контролировать вашу плату UNO, используя клавиатуру! Вы, например, сможете отправлять команды с вашего компьютера, чтобы включать светодиоды. В этом уроке вы будете использовать те же компоненты и макетную плату, что в Уроке Lesson 8. Если вы еще не ознакомились с материалом Урока 8, пожалуйста, выполните его перед началом этого урока.

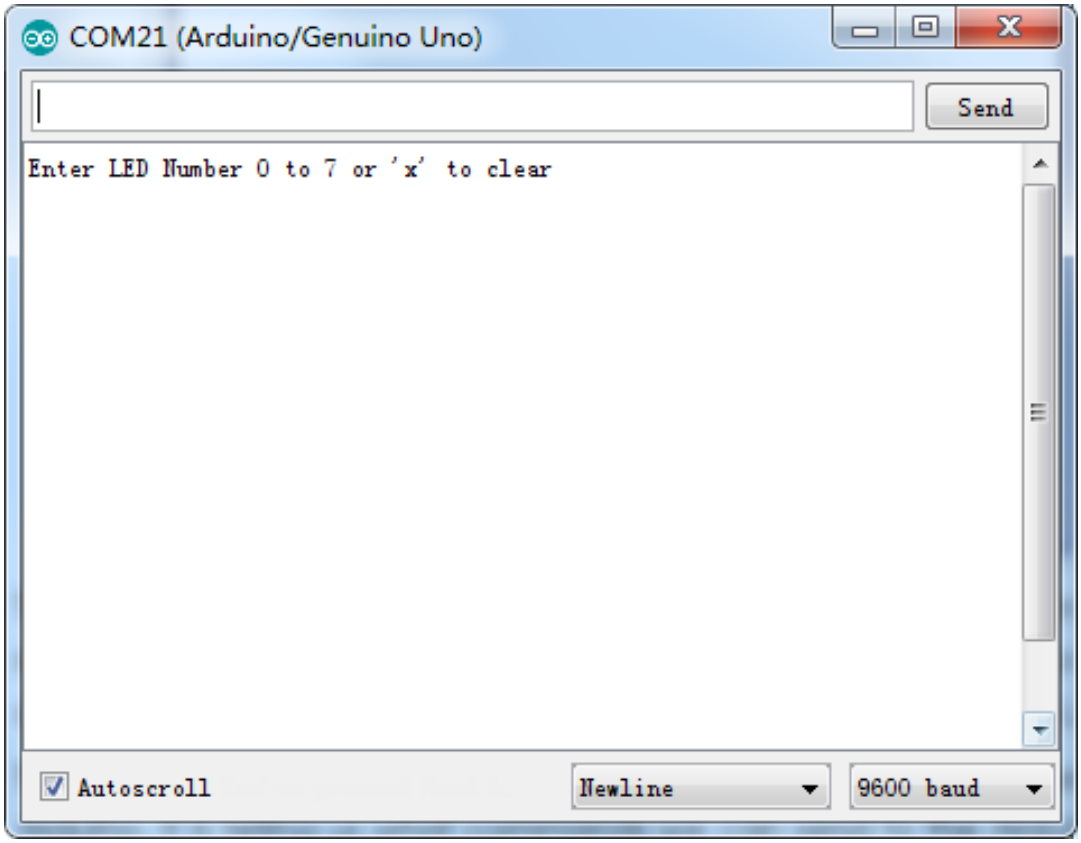
Необходимые шаги

После загрузки скетча на вашу UNO, нажмите на кнопку в правом верхнем углу окна среды разработки Arduino, как показано ниже.



Откроется следующее окно.

Нажмите на кнопку “Serial Monitor” для включения функции монитора. Основные сведения о серийном мониторе приведены в Уроке 1.

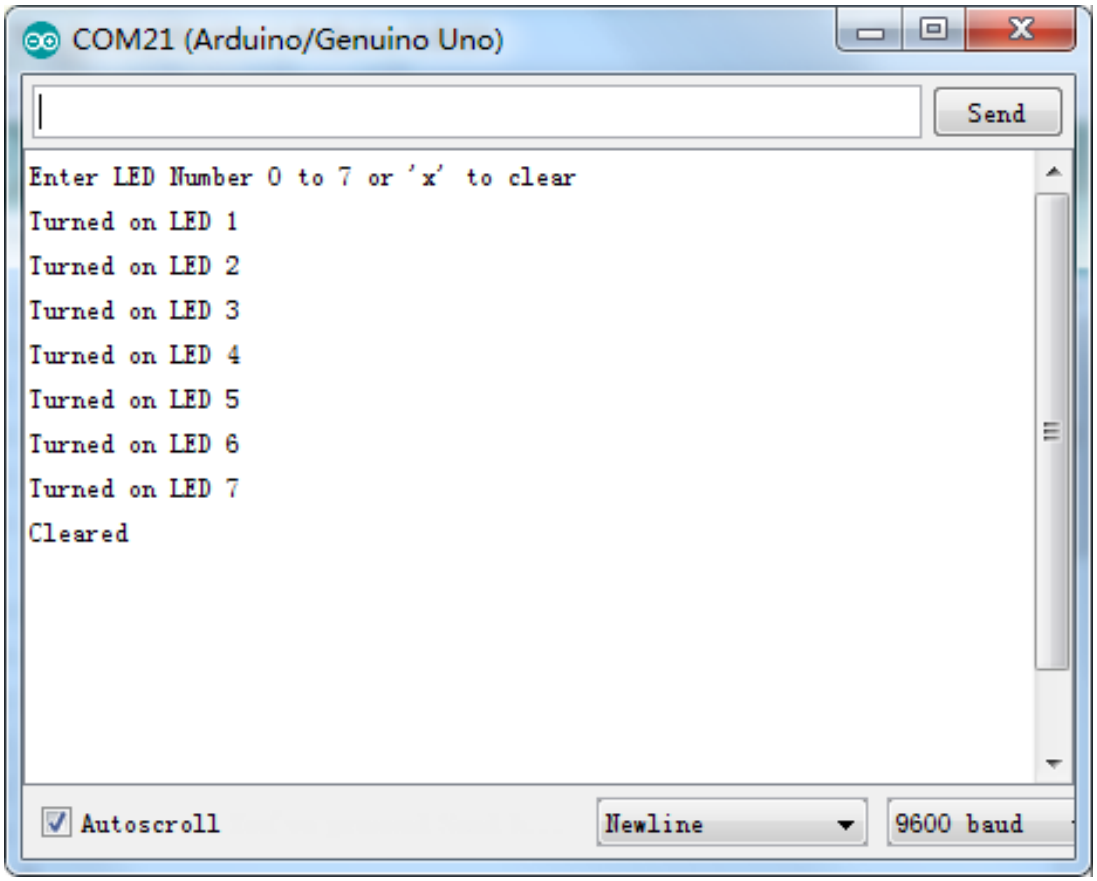


Это окно называется серийным монитором и является частью среды разработки Arduino. Задачи серийного монитора включают отправку сообщений с вашего компьютера на плату UNO (через USB-порт) и получение сообщений с платы.

Сообщение “Enter LED Number 0 to 7or 'x' to clear” (“Введите номер светодиода от 0 до 7 или 'x' для выключения”) было отправлено Arduino. Оно показывает, какие команды мы можем отдавать Arduino: либо отправить 'x' (чтобы выключить все светодиоды), либо номер светодиода, который вы хотите включить (где 0 – нижний светодиод, 1 – следующий за ним, 7 – самый верхний светодиод).

Попробуйте ввести следующие команды в строку с кнопкой “Отправить” ('Send') в верхней части Серийного Монитора. По очереди введите значения “x”, “0”, “3”, “5”, сопровождая каждый ввод нажатием кнопки “Отправить” ('Send').

При выключенных светодиодах, ввод “x” не даст результата, но при вводе номера, должен загореться соответствующий светодиод, и вы получите сообщение подтверждения с платы UNO board (как на рисунке ниже).



Введите “x” снова и нажмите “Отправить”, чтобы выключить все светодиоды.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 25: Серийный Монитор”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Этот скетч основывается на скетче, использованного в Уроке 24, поэтому мы остановимся только на моментах, которые были добавлены в код. Полезно будет иметь перед глазами полный текст скетча в вашей программе Arduino.

В функции 'setup' добавлено три строки:

```
void setup()
{
pinMode(latchPin, OUTPUT); pinMode(dataPin, OUTPUT); pinMode(clockPin, OUTPUT); updateShiftRegister(); Serial.begin(9600);
while (! Serial); // Wait until Serial is ready - Leonardo Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

Сначала имеем команду 'Serial.begin(9600)'. Она начинает последовательную передачу данных, чтобы UNO могла отправлять команды через USB-соединение. Число 9600 называется “скоростью в бодах”. Оно отображает, насколько быстро происходит передача данных. Скорость можно повысить, но вам так же придется выбрать соответствующее значение в Серийном Мониторе Arduino. Обсудим это позже. Пока оставим это значение, как есть – 9600.

Строка, начинающаяся с 'while', служит для проверки присутствия “кого-либо” на другом конце USB-подключения для передачи сообщений перед началом самой передачи. Иначе, сообщения будут отправляться, но не будут отображаться. Вообще, эта строка кода обязательна только для платы Arduino Leonardo, поскольку Arduino UNO совершает автоматический сброс при открытии Серийного Монитора.

Последняя добавленная строка в 'setup' отправляет сообщение, которое мы видим в верхней части Серийного Монитора.

Функция 'loop' – там, где происходит все самое интересное:

```
  void loop()
{
if (Serial.available())
{
char ch = Serial.read();
if (ch >= '0' && ch <= '7')
{
int led = ch - '0'; bitSet(leds, led); updateShiftRegister();
Serial.print("Turned on LED "); Serial.println(led);
}
if (ch == 'x')
{
leds = 0; updateShiftRegister();
```

```
Serial.println("Cleared");
}
}
}
```

Все, что происходит в функции, содержится в рамках условия 'if'. Поэтому, помимо случая, когда вызов встроенной функции Arduino 'Serial.available()' будет 'true', то ничего происходить не будет.

Serial.available() вернет 'true', если данные были отправлены на плату UNO и готовы к обработке. Входящие сообщения находятся в, так называемом, буфере данных, и, если буфер не пустой, функция Serial.available() возвращает 'true'.

Если сообщение было получено, то переходим к следующей строке кода:

```
char ch = Serial.read();
```

Так происходит считывание следующего символа в буфере и перенос из буфера. Эта функция также присваивает это значение переменной 'ch'. Переменная 'ch' – это переменная типа 'char', от 'character' (символ), и, как предполагает имя, содержит отдельный символ.

Если вы следовали инструкциям в строке верхней части Серийного Монитора, то этот символ будет либо одноразрядным числом от 0 до 7, либо буквой x.

Условие 'if' в следующей строке проверяет одноразрядность числа, проверяя, больше ли или равна переменная 'ch' символу '0' и меньше ли или равна переменная 'ch' символу '7'. Такое сравнение символов может показаться немного странным, но оно вполне допустимо. Каждый символ может быть представлен уникальным числовым кодом из таблицы кодировки ASCII. Поэтому, при сравнении символов с использованием операторов “<=” и “>=“, на самом деле происходит сравнение соответствующих им кодов ASCII.

После проверки переходим к следующей строке:

```
int led = ch – '0';
```

Теперь мы выполняем арифметические операции! Мы вычитаем число '0' от введенного числа. Поэтому, если вы ввели 0, то 0 – 0 будет равно 0. Если вы ввели 7, то 7 – 0 будет равно числу 7, поскольку для вычитания используются коды ASCII.

Поскольку нам известно количество светодиодов и какой из них мы хотим включить, нам нужно только проверить соответствующий бит в переменной 'leds' и обновить регистр сдвига.

```
bitSet(leds, led); updateShiftRegister();
```

Следующие две строки отправляют сообщение подтверждения Серийному Монитору.

```
Serial.print("Turned on LED ");
```

```
Serial.println(led);
```

В первой строке используется Serial.print вместо Serial.println. Разница между двумя в том, что Serial.print не совершает перевод строки после вывода своего параметра. Мы используем эту команду в первой строке, потому что мы выводим сообщение в двух частях. Сначала, 'Turned on LED '(“Включен светодиод “), а затем номер светодиода.

Номер светодиода содержится в переменной 'int', а не в текстовой строке. Serial.print может воспринимать либо тестовую строку, заключенную в двойные кавычки, либо 'int' (в принципе, любой тип переменной).

После первого условия 'if' имеем второе условие 'if', которое проверяет, является ли переменная 'ch' буквой “x“.

```
if (ch == 'x')
{
leds = 0; updateShiftRegister(); Serial.println("Cleared");
}
```

Если условие выполняется, то все светодиоды выключается и мы получаем подтверждающее сообщение.

Урок 10: Фоторезистор

Обзор

В этом уроке вы научитесь измерять интенсивность освещения, используя аналоговый ввод. Мы будем контролировать количество включаемых светодиодов, изменяя уровень освещенности. Фоторезистор находится в нижней части макетной платы, а переменный резистор – в верхней.

Необходимые компоненты:

(1) x плата Elegoo Uno R3

(1) x макетная плата на 830 точек

(8) x светодиоды

(8) x резистор (220 Ом)

(1) x резистор (1 кОм)

(1) x сдвиговый регистр 74HC595

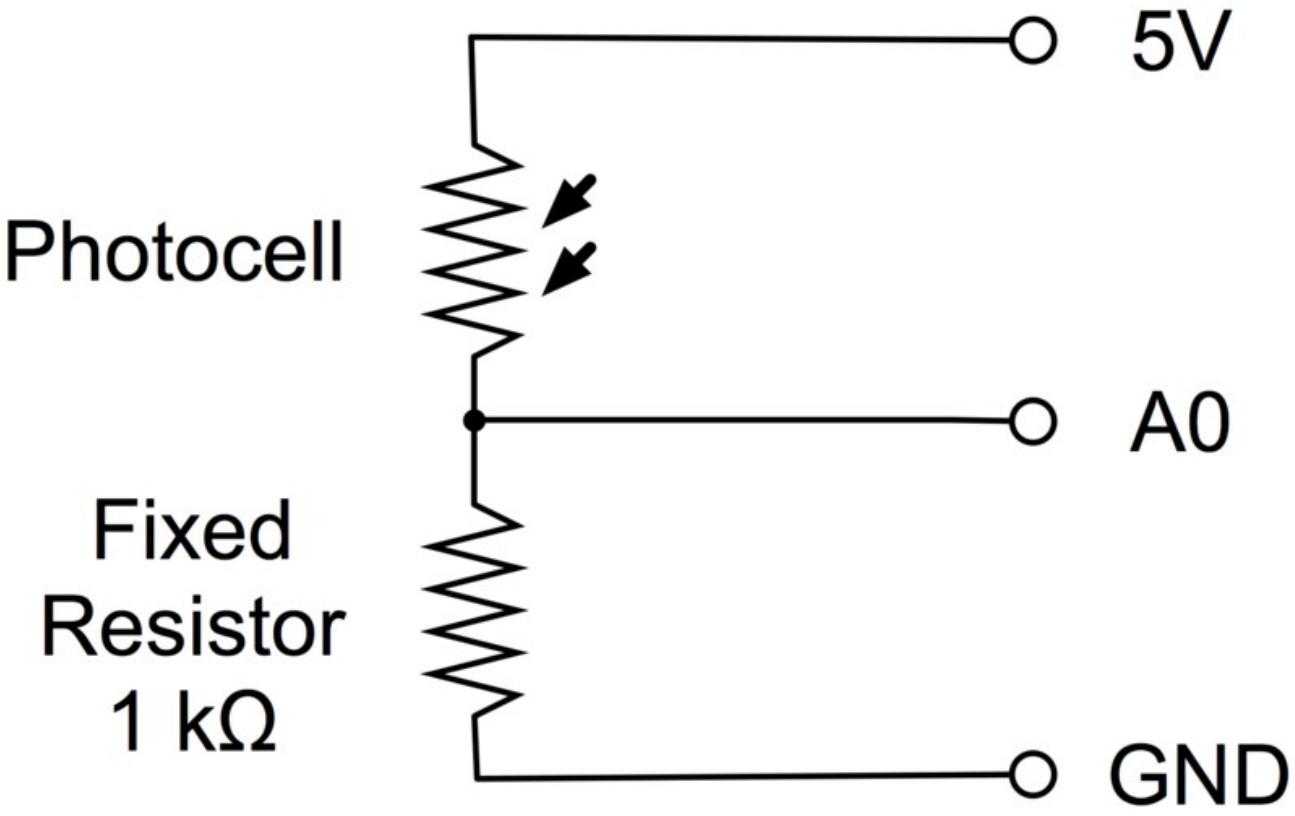
(1) x фоторезистор

(16) x перемычки (типа “папа” – “папа”)

Представление компонентов

Фоторезистор:

Фоторезистор – это резистор, сопротивление которого изменяется при облучении светом. Фоторезистор в нашем уроке имеет сопротивление около 50 кОм в темноте и 500 Ом при ярком освещении. Для преобразования переменного значения сопротивления в значения, которые мы можем “принять” аналоговым входом нашей платы UNO R3, необходимо это сопротивление перевести в напряжение. Самый простой способ реализации этого – последовательно подключить к нему обыкновенный резистор.



Резистор в подключении с фотоэлементом ведут себя как переменный резистор с подвижным электродом в точке их соединения (потенциометр). Когда яркость света очень высокая, сопротивление фоторезистора будет ниже чем сопротивление постоянного резистора, что соответствует перемещению движка переменного резистора в верхнюю точку (+5 В).

В темноте сопротивление фоторезистора превысит сопротивление постоянного резистора в 1 кОм, что равносильно перемещению движка переменного резистора к общему проводу (GND).

Загрузите скетч, представленный ниже, и попробуйте сначала закрыть фоторезистор пальцем, а затем поднести его к источнику света.

Подключение

Схема

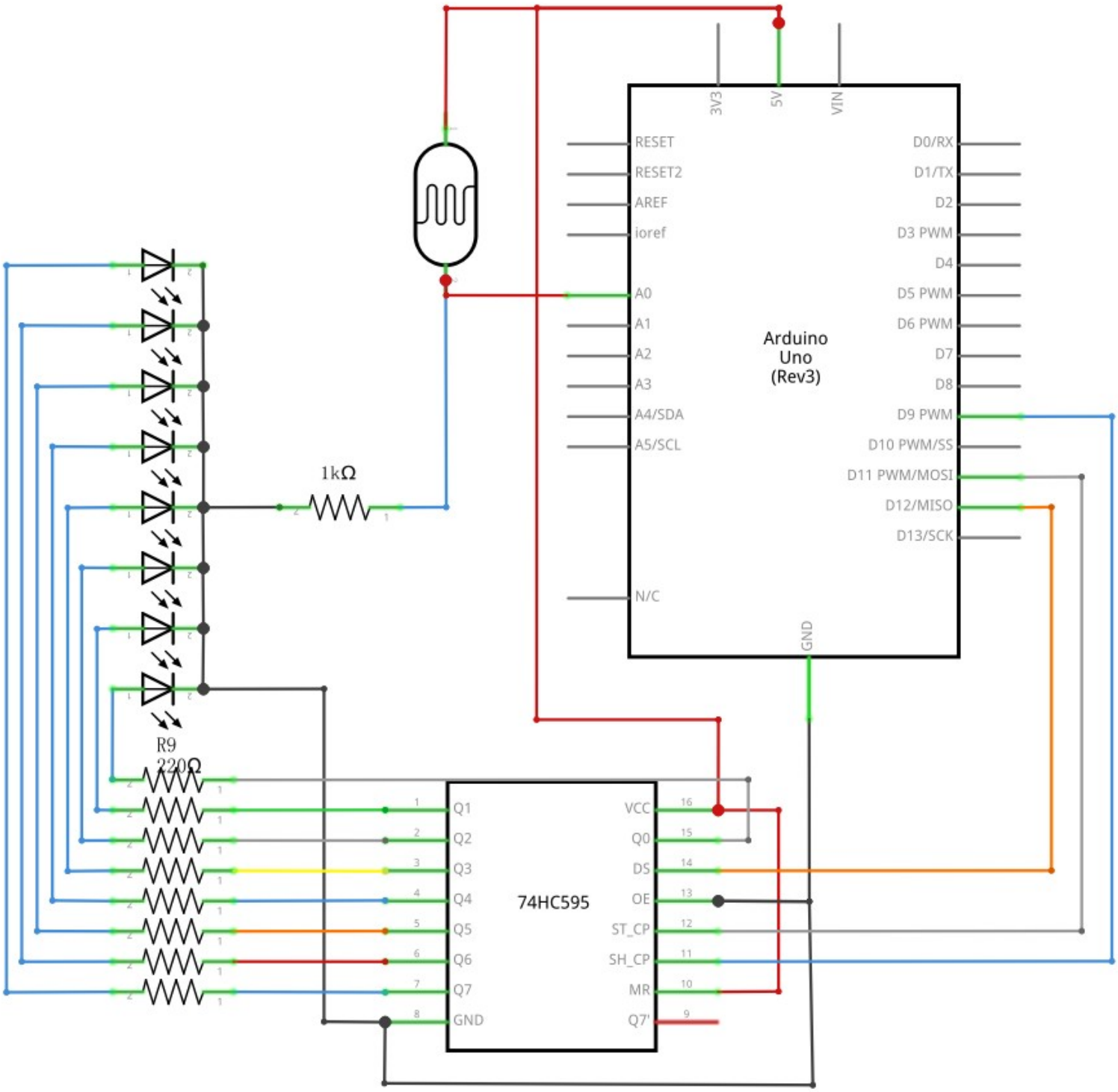
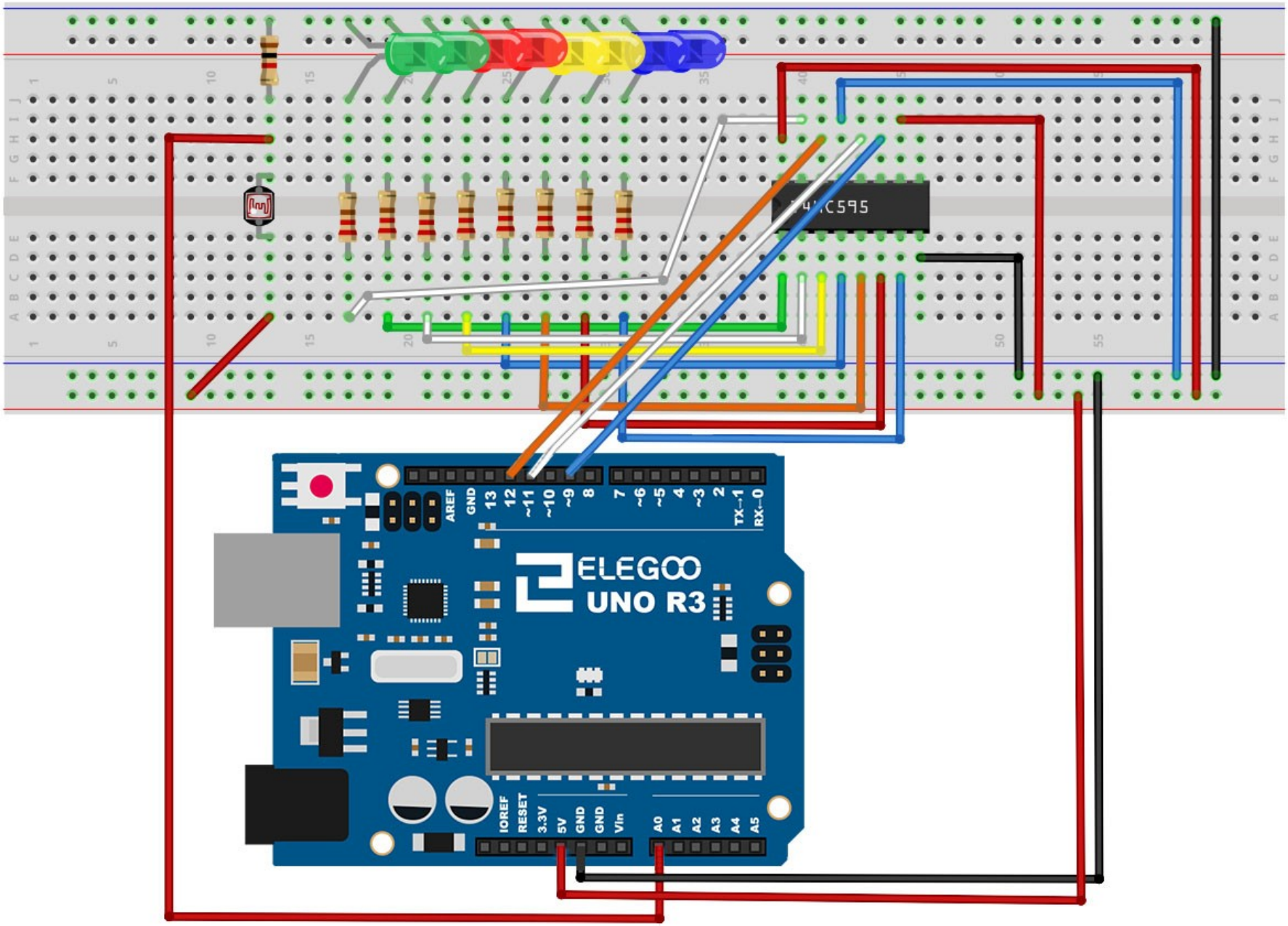


Схема монтажа



Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 26: Фоторезистор”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Первое, на что стоит обратить внимание – это то, что мы поменяли имя аналогового вывода на 'lightPin' вместо 'potPin', поскольку мы больше не работаем с переменным резистором. Единственным существенным изменением кода является строчка, рассчитывающая количество светодиодов, которые необходимо включить:

```
int numLEDSLit = reading / 57;        // all LEDs lit at 1k
```

В этот раз, мы делим «сырые» значения на 57, а не на 114. Другими словами, мы делим на половину от значения, которое мы использовали с переменным резистором для его разделения на 9 областей (начиная с включения 0 светодиодов до всех 8 светодиодов). Этот происходит за счет резистора в 1 кОм. Это значит, когда фоторезистор имеет такое же сопротивление, «сырым» значением будет 1023 / 2 = 511, что эквивалентно 8 включенным светодиодам и биту (numLEDSLit) равному 8.

Рисунок для примера