



Elegoo Super Starter Kit для UNO

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 аксессуары 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 открытом 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 услуги 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

Stepper Motor
1PC



Servo Motor
(SG90)
1PC



IR Receiver
Module
1PC



5V Relay
1PC



Uno R3
Controller
Board
1PC



LCD 1602 Module
(with pin header)
1PC



ULN2003 Stepper
Motor Driver
Module
1PC



Power Supply
Module
1PC



Prototype Expansion
Module
1PC



Contact us : service@elegoo.com

4 digit 7 - segment
Display
1PC



1 digit 7 -segment
Display
1PC



Tilt Ball Switch
1PC



Joystick Module
1PC



Fan Blade and
3-6V DC
Motor
(with wire)
each
1PC



DHT11 Temperature
and Humidity
Module
1PC



Ultrasonic Sensor
1PC



Button (Small)
5PCS



Potentiometer
1PC



Passive Buzzer
1PC



9V Battery with
Snap-on Connector
Clip 1PC



Active Buzzer
1PC



Remote
1PC



USB Cable
1PC



Female-to-male
Dupont wire
10PCS



Breadboard Jumper
Wire 65PCS



74HC595 IC
1PC



L293D
1PC



830 Tie-Points
Breadboard
1PC



Thermistor
1PC



Diode Rectifier
2PCS



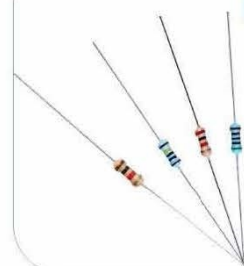
LED
25PCS



Photoresistor
(Photocell)
2PCS



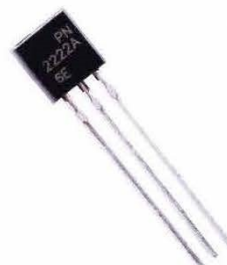
Resistor
120PCS



RGB
2PCS



NPN Transistor
PN2222
2PCS



Содержание

Урок 0: Установка среды IDE.....	8
Урок 1: Установка библиотек и работа с серийным монитором	23
Урок 2: Мигающий светодиод	32
Урок 3: Светодиод.....	7
Урок 4: RGB светодиод	50
Урок 5: Цифровые порты (Подключение кнопки)	61
Урок 6: Пьезоизлучатель со встроенным генератором (активный зуммер)	67
Урок 7: Пьезоизлучатель без встроенного генератора (пассивный зуммер)	71
Урок 8: Датчик наклона	72
Урок 9: Серводвигатель	76
Урок 10: Ультразвуковой дальномер	80
Урок 11: Датчик температуры и влажности DHT11.....	85
Урок 12: Аналоговый джойстик	91
Урок 13: ИК приемник (пульт).....	102
Урок 14: Жидкокристаллический индикатор (ЖКИ/LCD)	109
Урок15: Термометр	114
Урок 16: Восемь светодиодов и сдвиговый регистр 74HC595	162
Урок 17: Серийный монитор.....	169
Урок 18: Фоторезистор.....	175
Урок 19: Сегментный индикатор и 74HC595	180
Урок 20: Четырехразрядный семисегментный индикатор.....	186
Урок 21: Двигатель постоянного тока	191
Урок 22: Реле.....	3
Урок 23: Шаговый двигатель	8
Урок 24: Управление шаговым двигателем с помощью пульта	214

Урок 0: Установка среды IDE

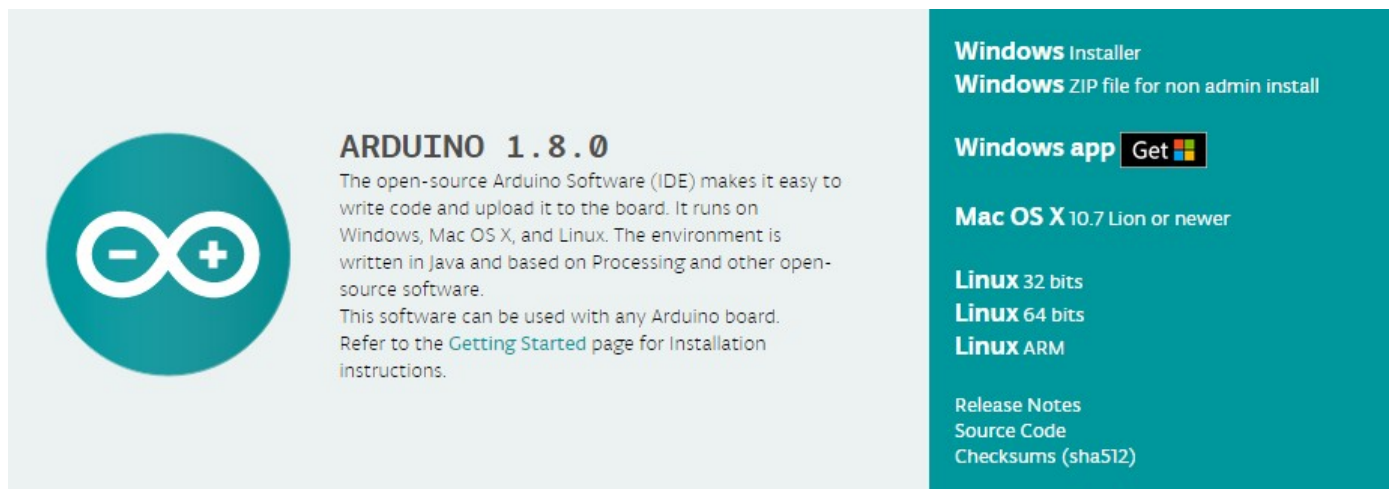
Вступление

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 программирования 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.





ARDUINO 1.8.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app 

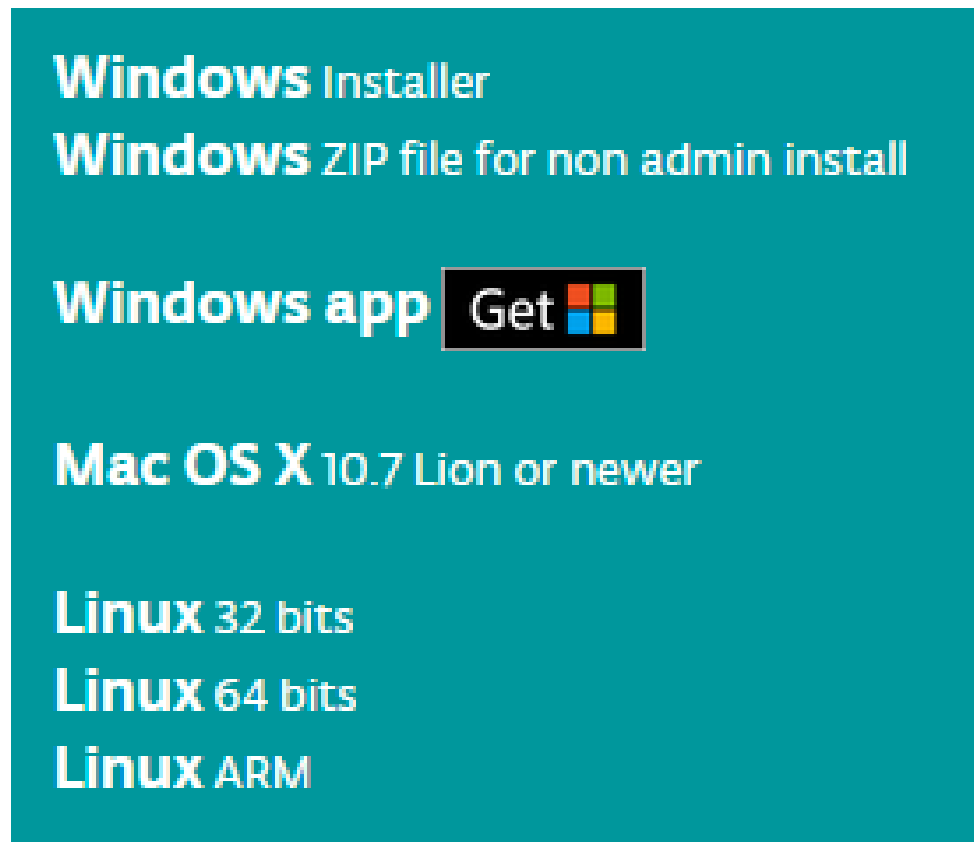
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

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)*.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

A light blue rectangular panel. On the left, there is a cartoon illustration of three characters: a square head, a rectangular head, and a circular head, each with a small antenna. To the right of the illustration, there is text: "SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 8,808,272 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!". Below this text, there are six circular buttons with the following labels: "\$3", "\$5", "\$10", "\$25", "\$50", and "OTHER". At the bottom of the panel, there are two buttons: "JUST DOWNLOAD" and "CONTRIBUTE & DOWNLOAD".

Нажмите *“JUST DOWNLOAD” (ТОЛЬКО ЗАГРУЗИТЬ)*.

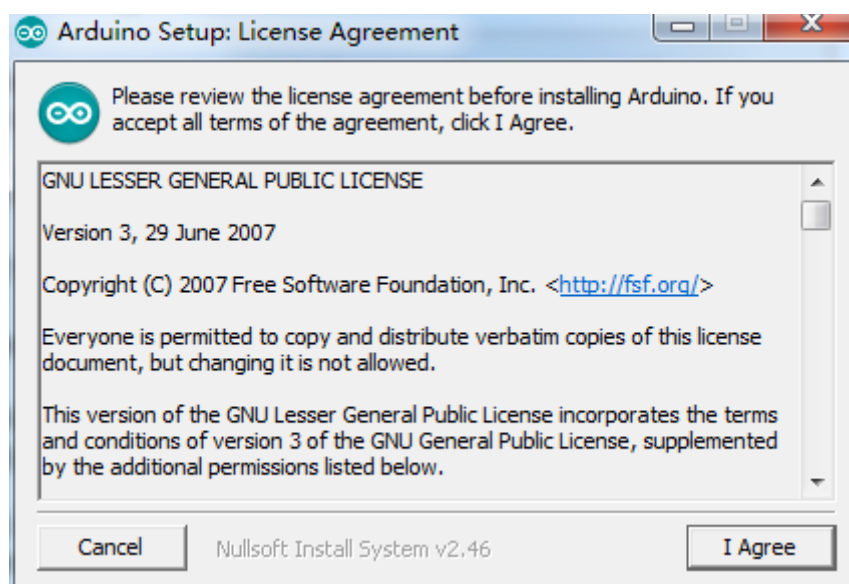
Версия программы 1.8.0 доступна вместе с другими материалами, которые мы предоставляем, и, на момент выпуска этого курса, является самой последней.

 `arduino-1.8.0-linux32.tar.xz`
 `arduino-1.8.0-linux64.tar.xz`
 `arduino-1.8.0-macosx.zip`
 `arduino-1.8.0-windows.exe`
 `arduino-1.8.0-windows.zip`

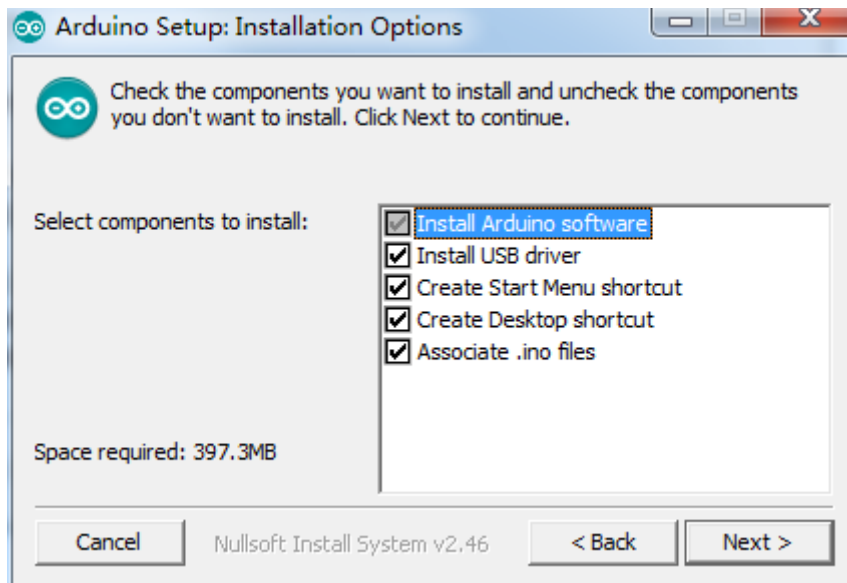
Установка Arduino (Windows)

Установить Arduino используя exe. Установочный пакет.

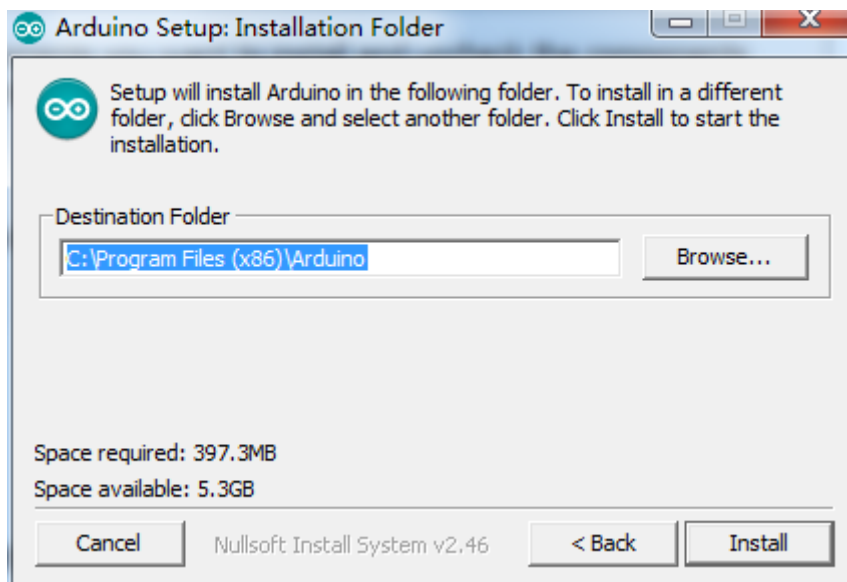
 `arduino-1.8.0-windows.exe`



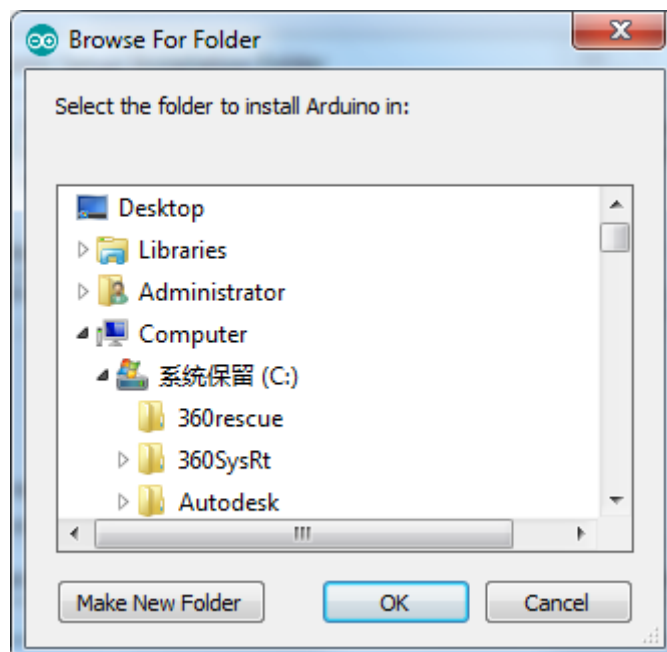
Выберите “I Agree” (Я согласен), чтобы перейти к следующему интерфейсу



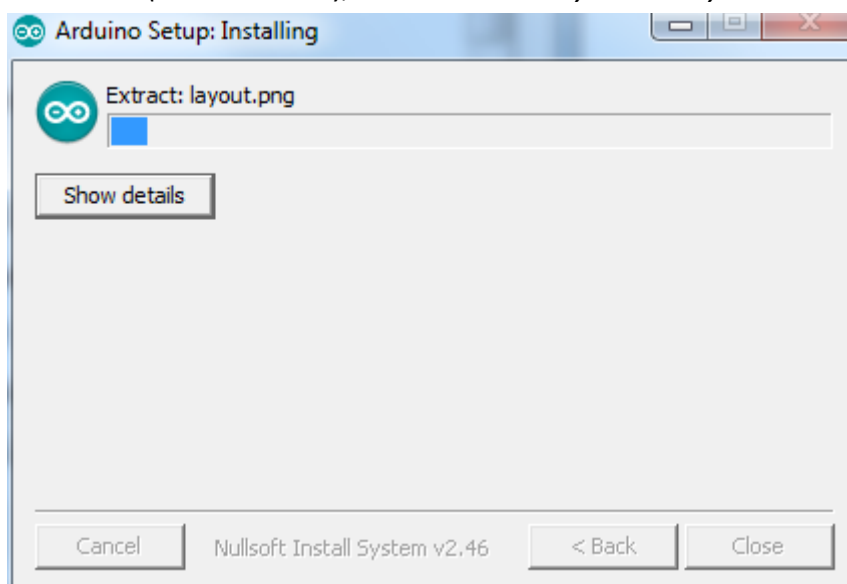
Нажмите “Next” (Далее)



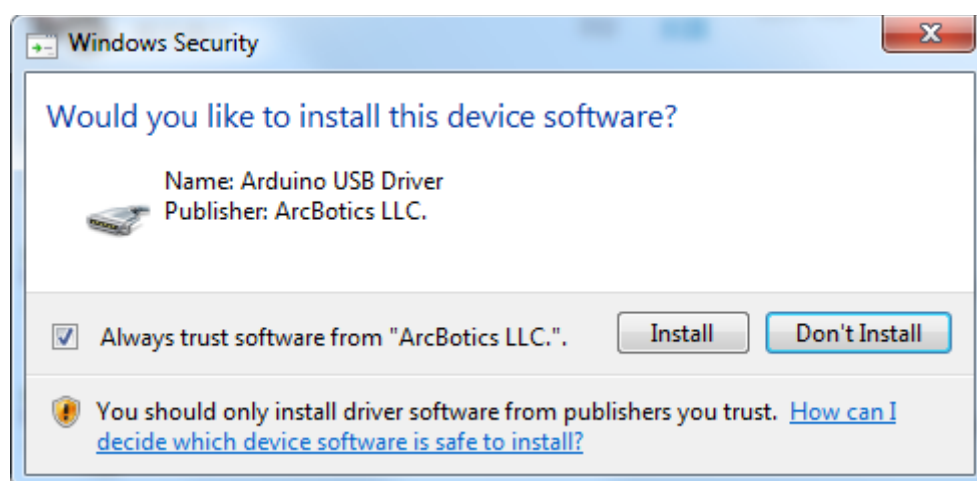
Вы можете выбрать “Browse...” для выбора пути установки или напрямую ввести необходимую вам директорию.



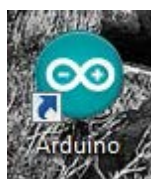
Выберите “Install” (Установить), чтобы начать установку.



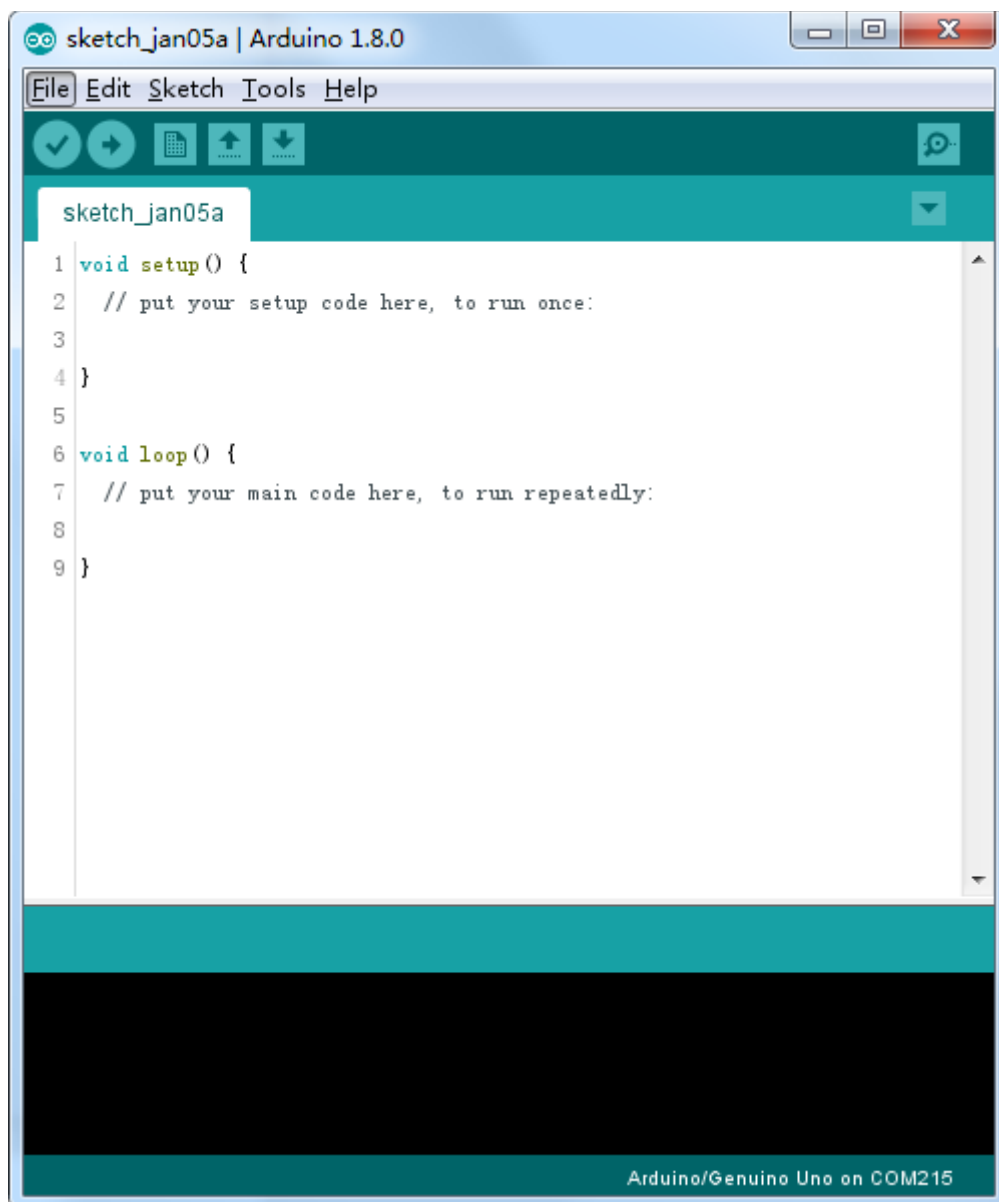
Наконец, когда вы увидите следующее окно, нажмите “Install” (Установить), чтобы завершить установку.



Вы увидите на рабочем столе следующий значок



Кликните дважды, чтобы выбрать желаемую среду разработки

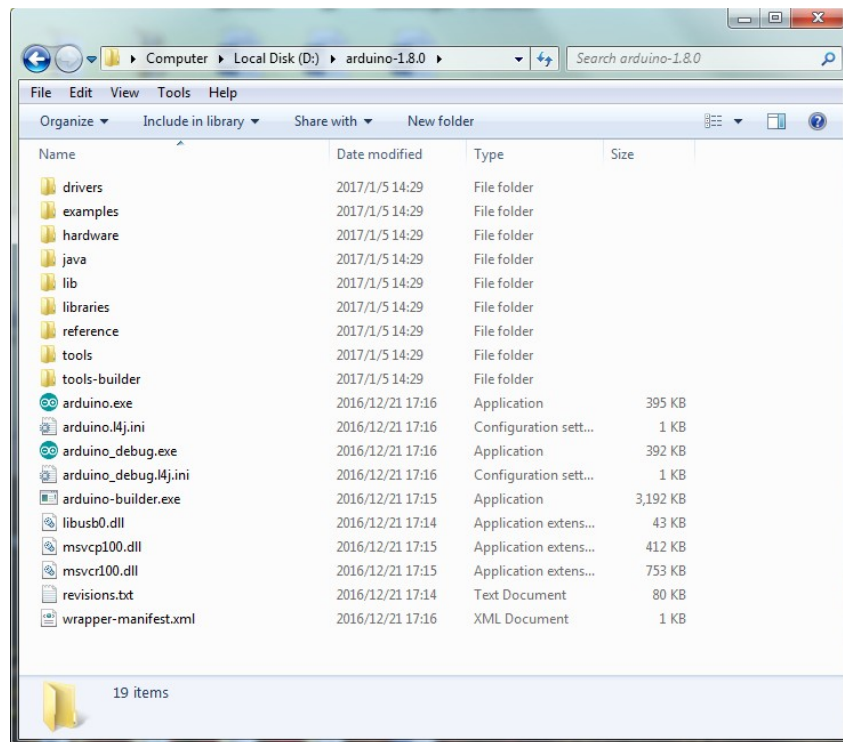


Вы можете остановить свой выбор на методе установки программы с использованием установочного пакета и пропустить материал, приведенный ниже, сразу перейдя к следующему уроку. Но если вам интересно узнать о других методах установки, пожалуйста, продолжайте читать.

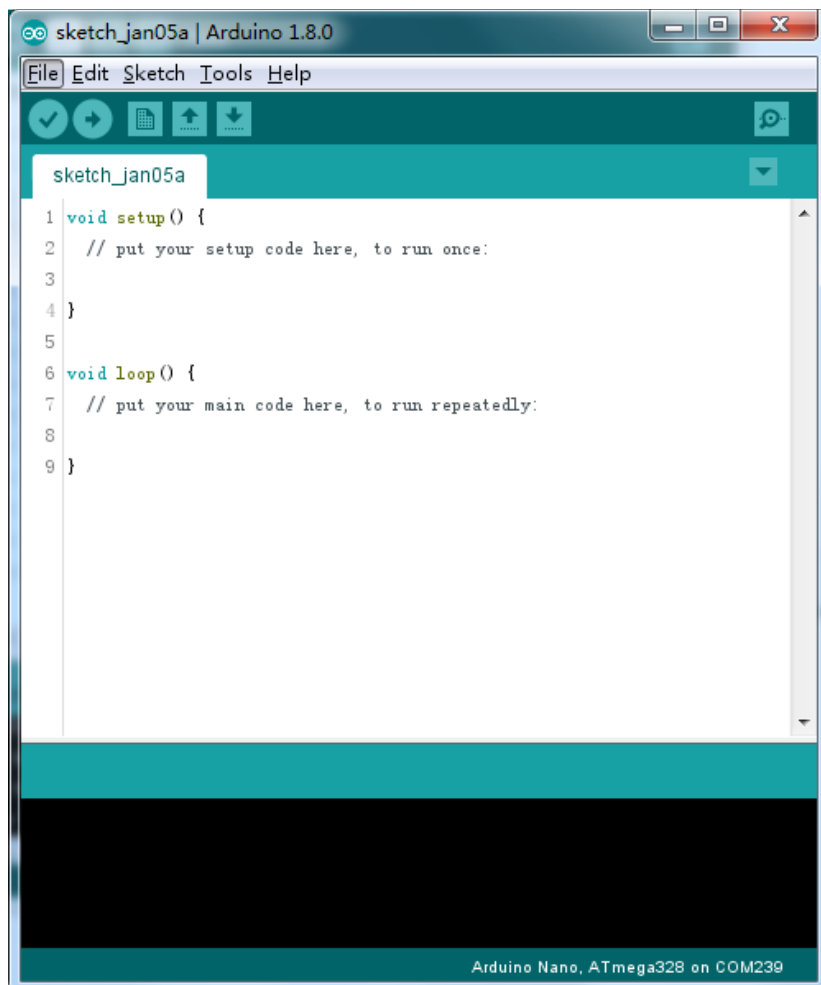
Распакуйте загруженный zip файл, двойным нажатием открыть программу и ввести желаемую среду разработки



arduino-1.8.0-windows.zip



 **arduino.exe**

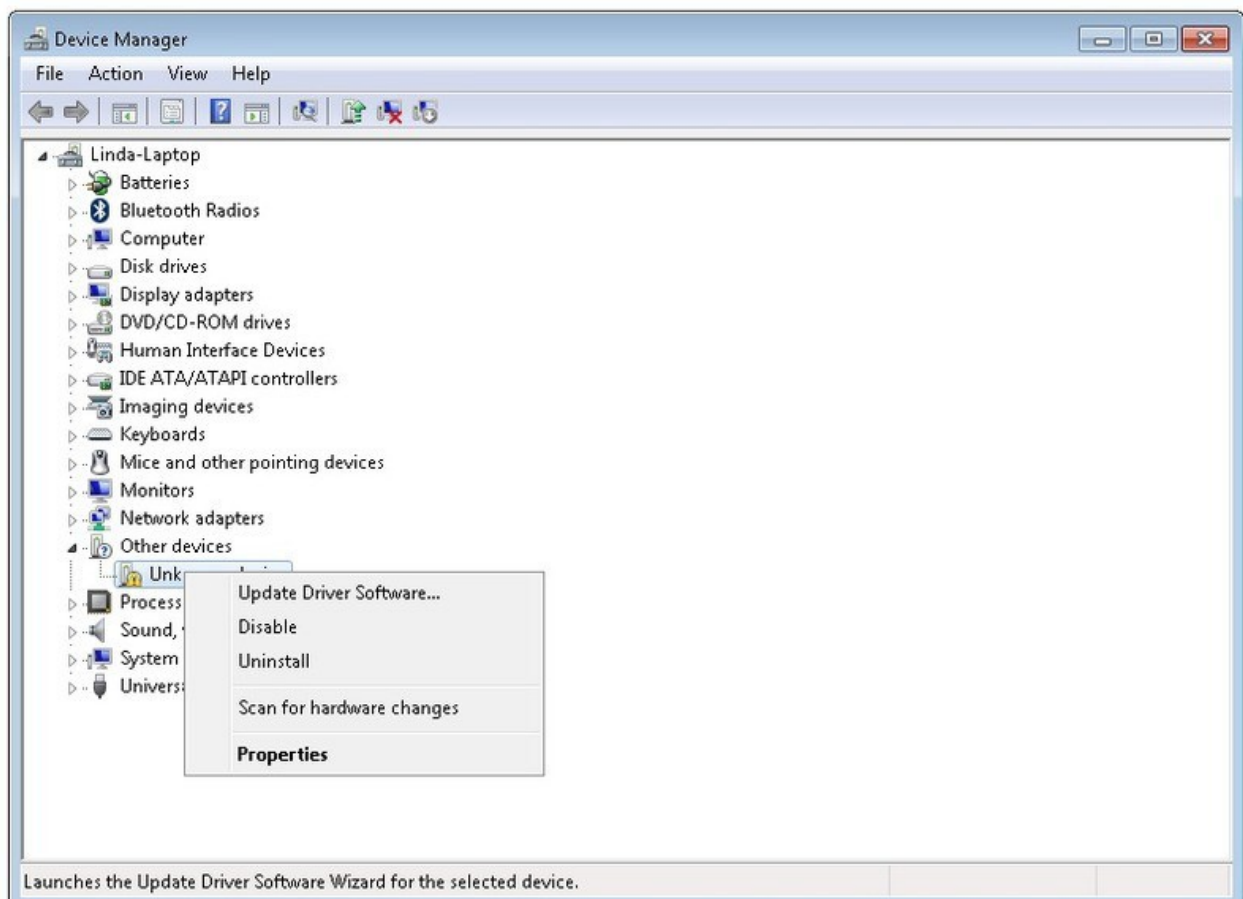


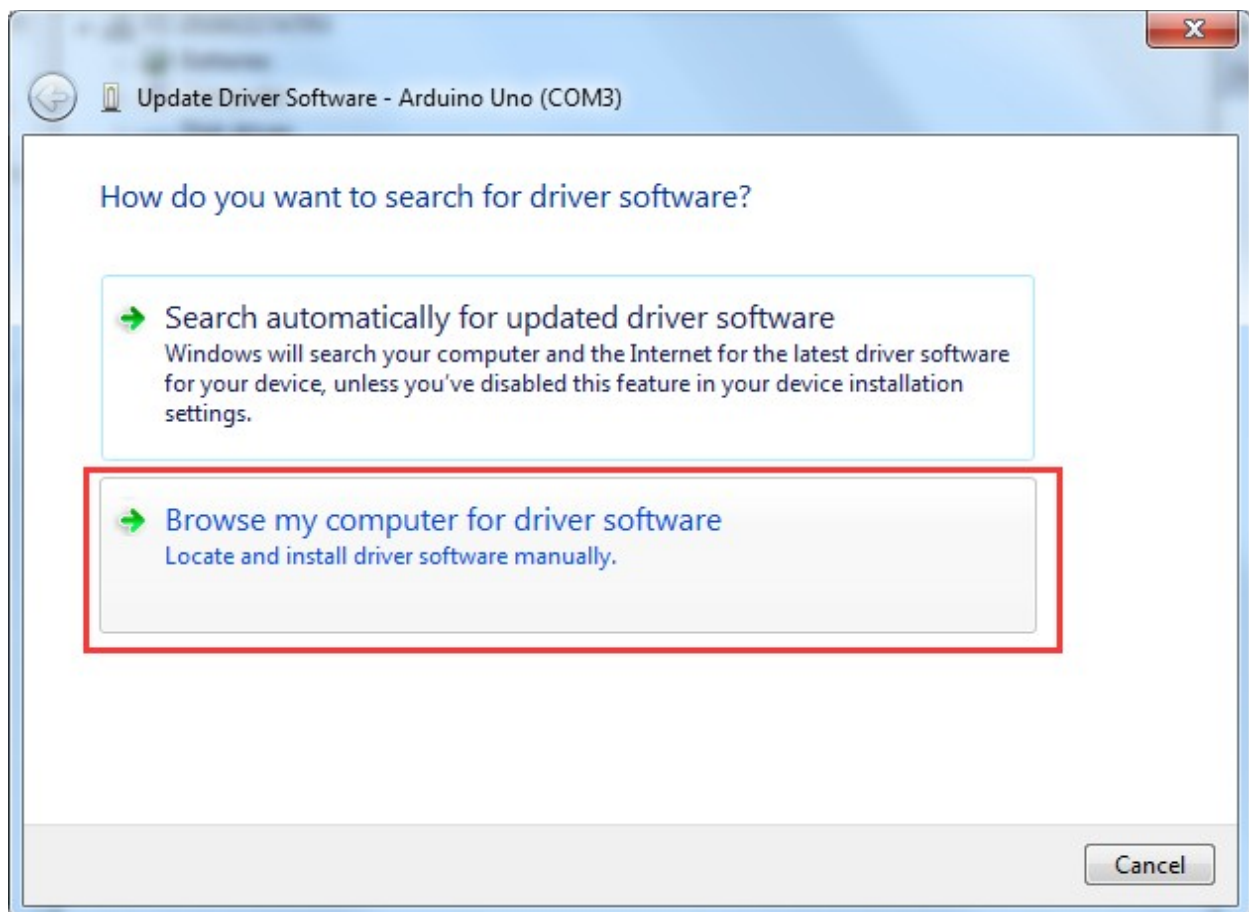
Данный метод установки требует отдельной установки драйвера.

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.

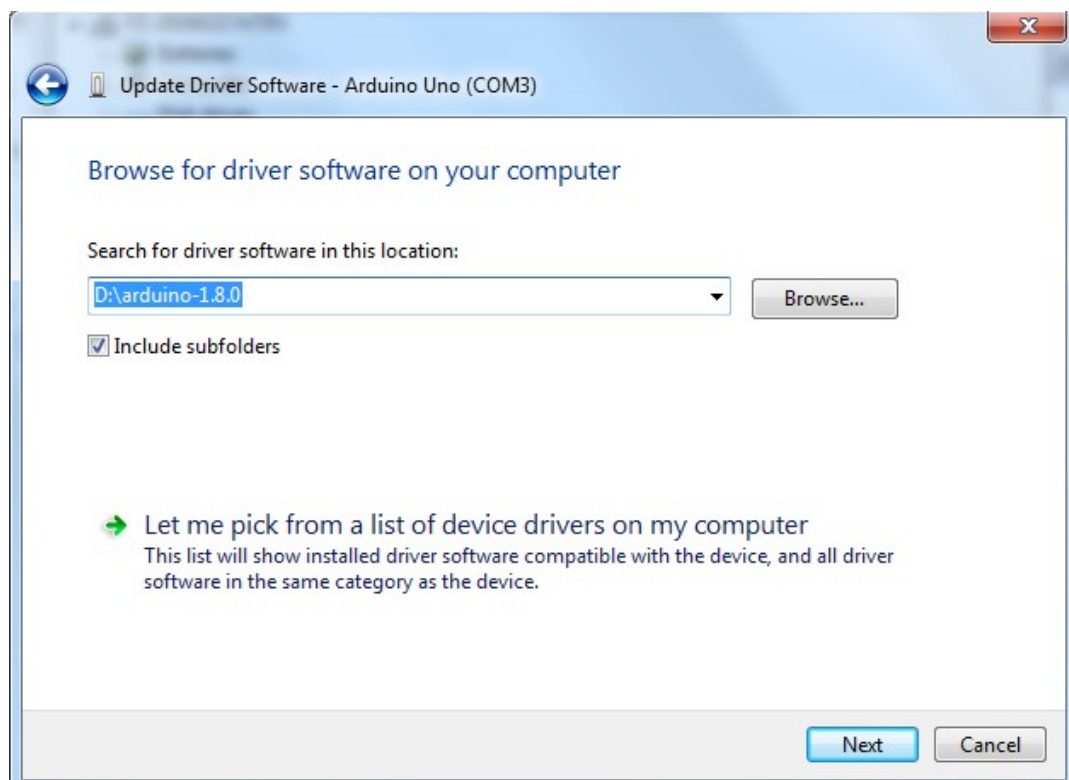
Самым проверенным способом установки USB драйверов является использование Диспетчера Устройств. Чтобы открыть Диспетчер Устройств в Windows 7, вы сначала должны открыть Панель Управления, затем выбрать опцию просмотра Значков, и найти Диспетчер Устройств в списке.

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

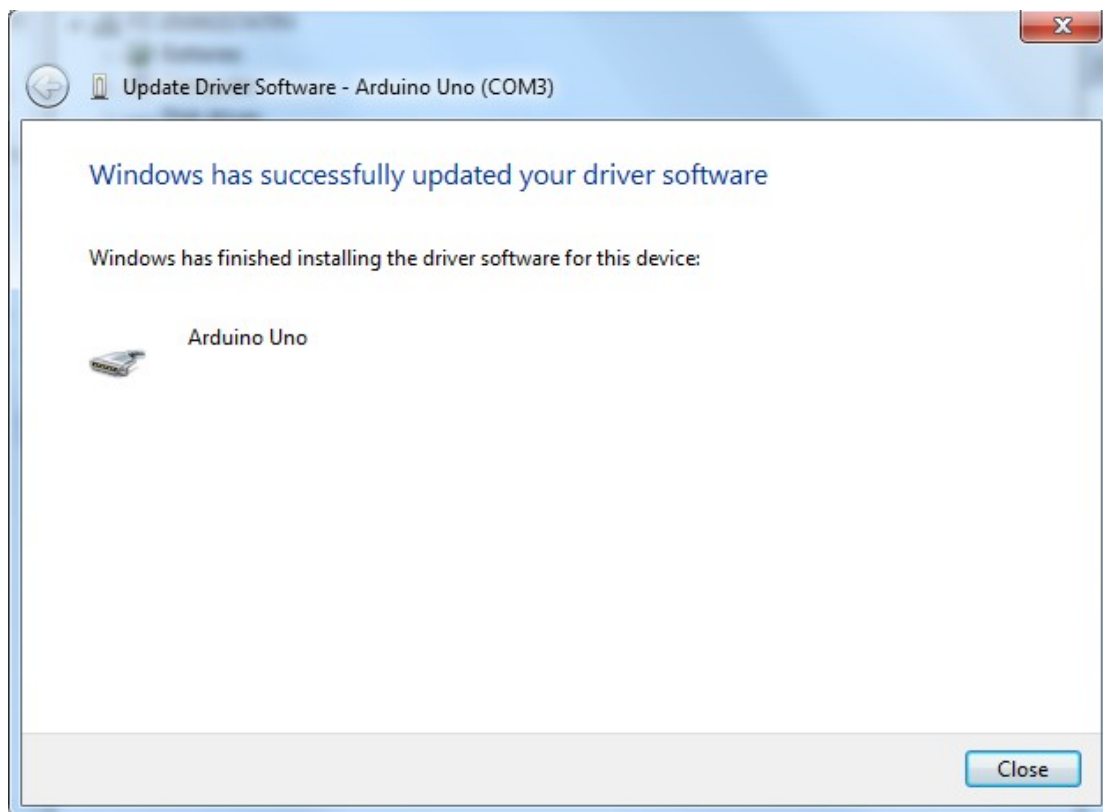




Щелкните правой кнопкой на устройство и выберите “Обновить Драйвер...”. Затем последует выбор либо “Автоматический поиск обновленного драйвера”, либо “Выполнить поиск драйверов на этом компьютере”. Выберите последнее и вручную укажите место размещения драйвера `X\arduino1.8.0\drivers`.



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.



arduino-1.8.0-macosx.zip

Установка Arduino (Linux)

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

 `arduino-1.8.0-linux32.tar.xz`

 `arduino-1.8.0-linux64.tar.xz`

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

 [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.

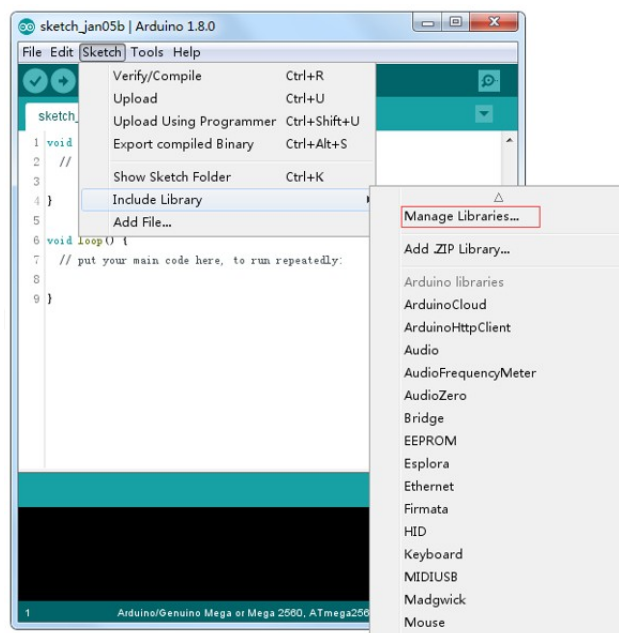
Что такое библиотека?

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

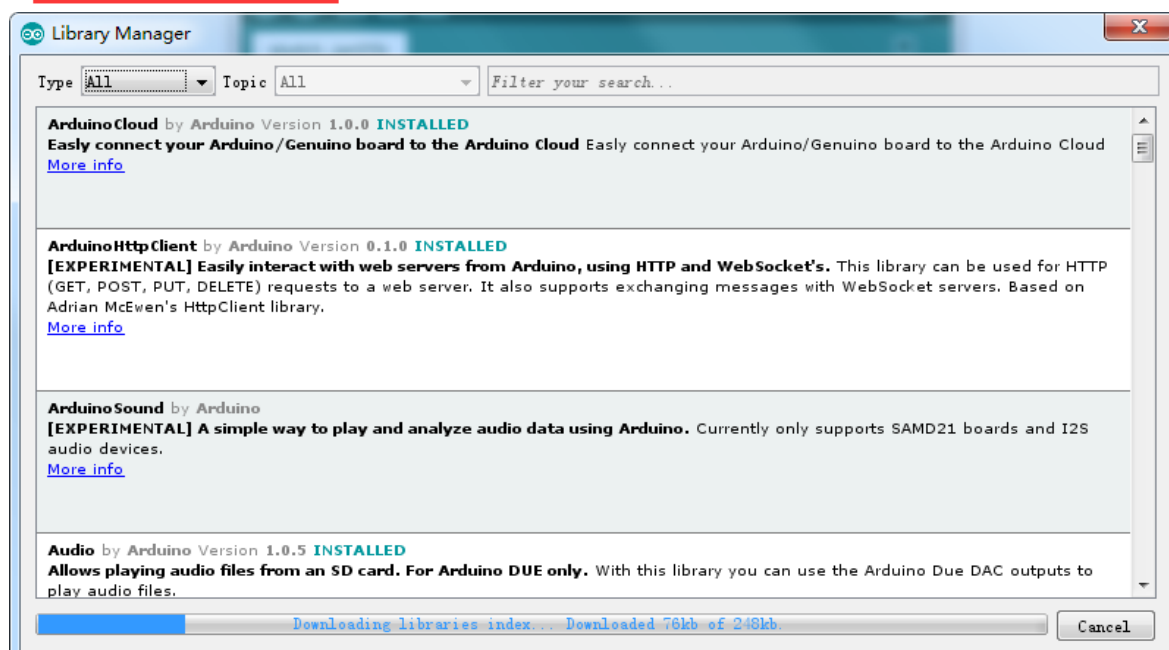
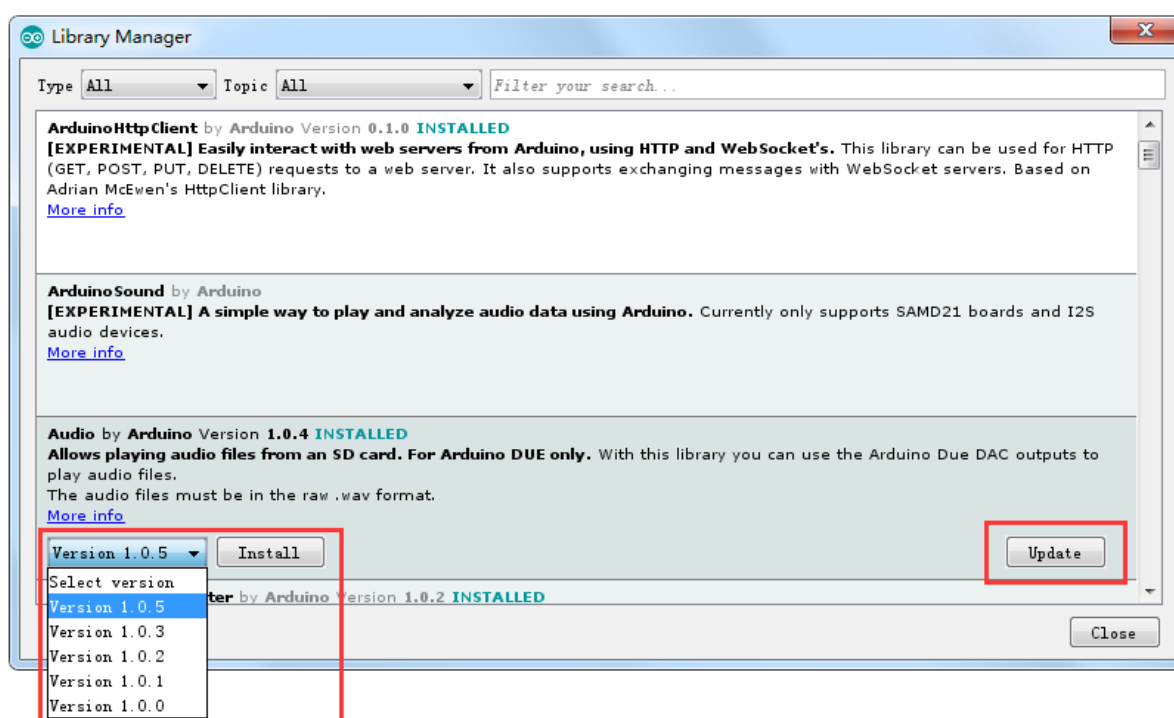
Используя Library Manager

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).

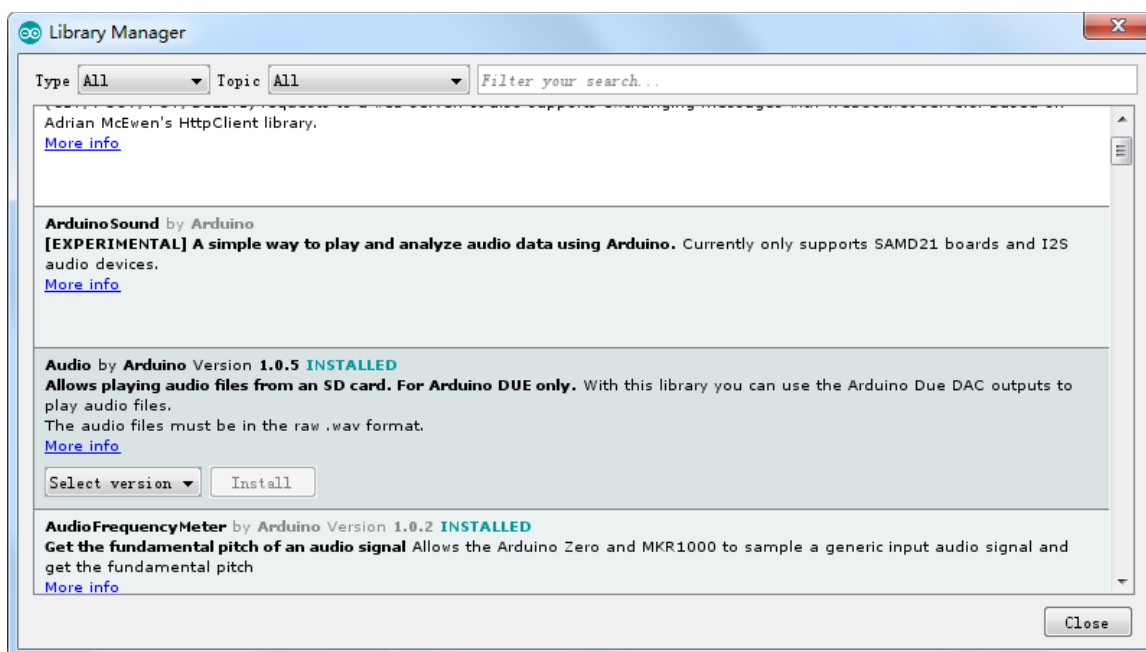


После чего откроется Library Manager, и вы сможете найти список библиотек, которые уже установлены, либо доступны для установки. В этом примере мы покажем, как установить библиотеку <Bridge>. Прокрутите список, чтобы найти ее, затем выберите необходимую вам версию библиотеки для установки. В некоторых 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.

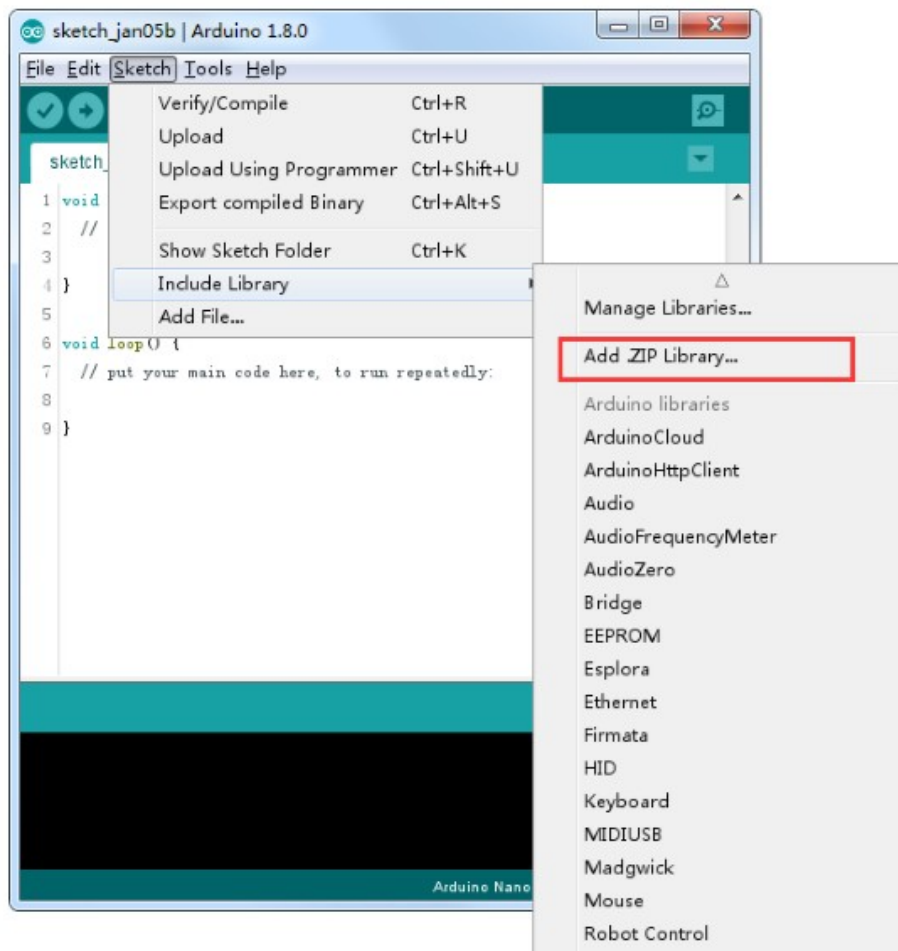


Новая библиотека теперь доступна в меню *Include Library*. Если вы хотите добавить вашу собственную библиотеку, откройте новую тему на [Github](#).

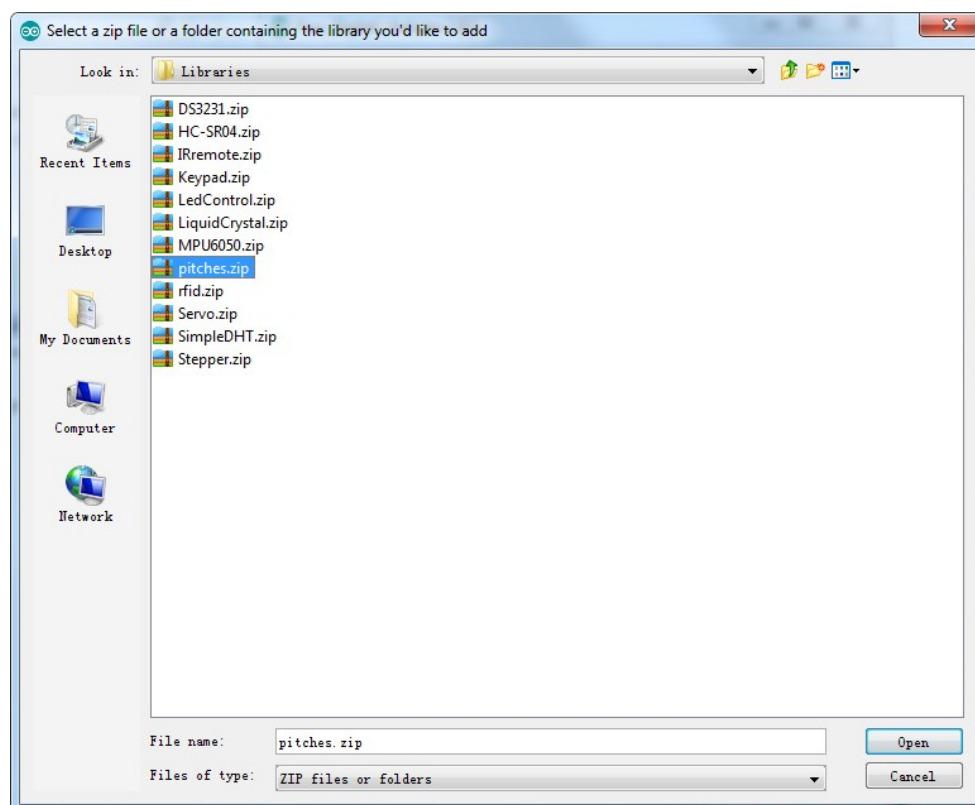
Импорт .zip библиотеки

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.

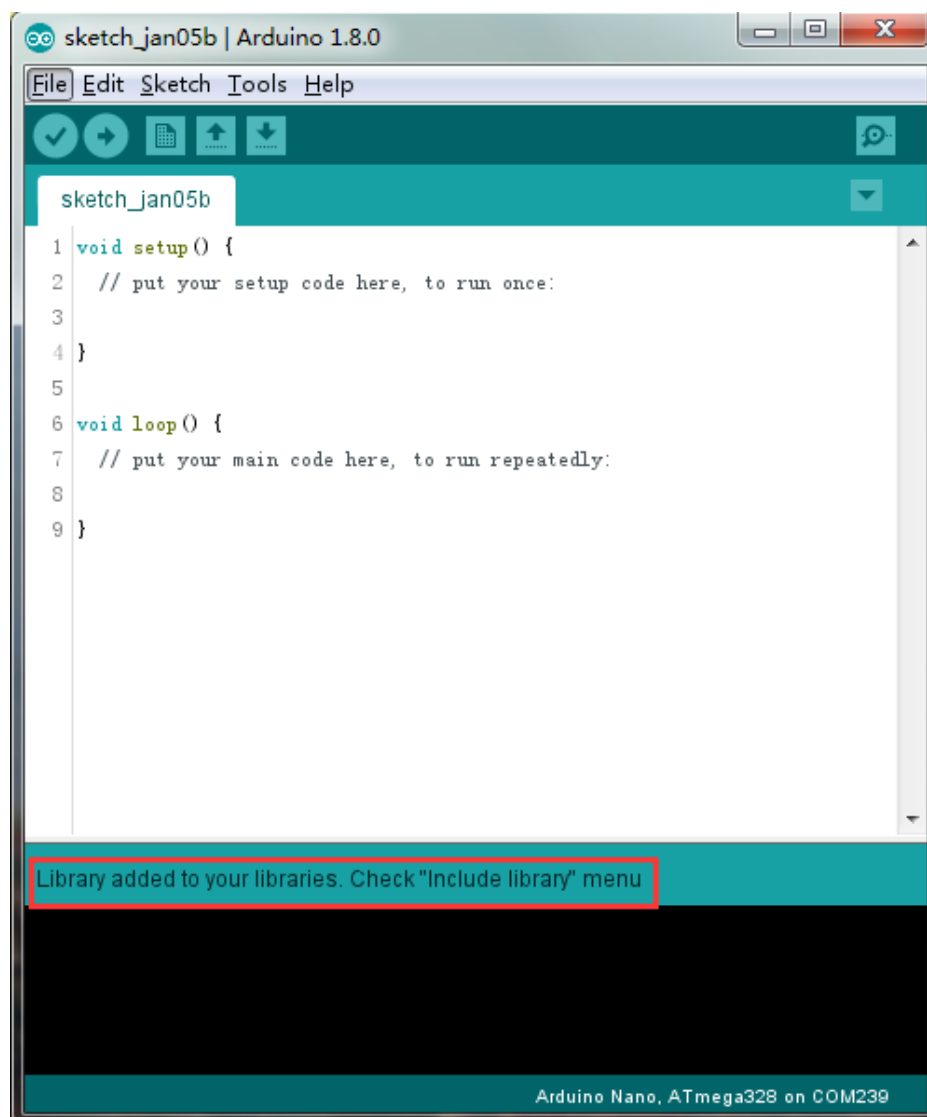
В среде разработки Arduino выберите меню *Sketch > Import Library*. В самом верху выпадающего списка выберите пункт "Add .ZIP Library" (Добавить .ZIP библиотеку).



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



/ 22



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)

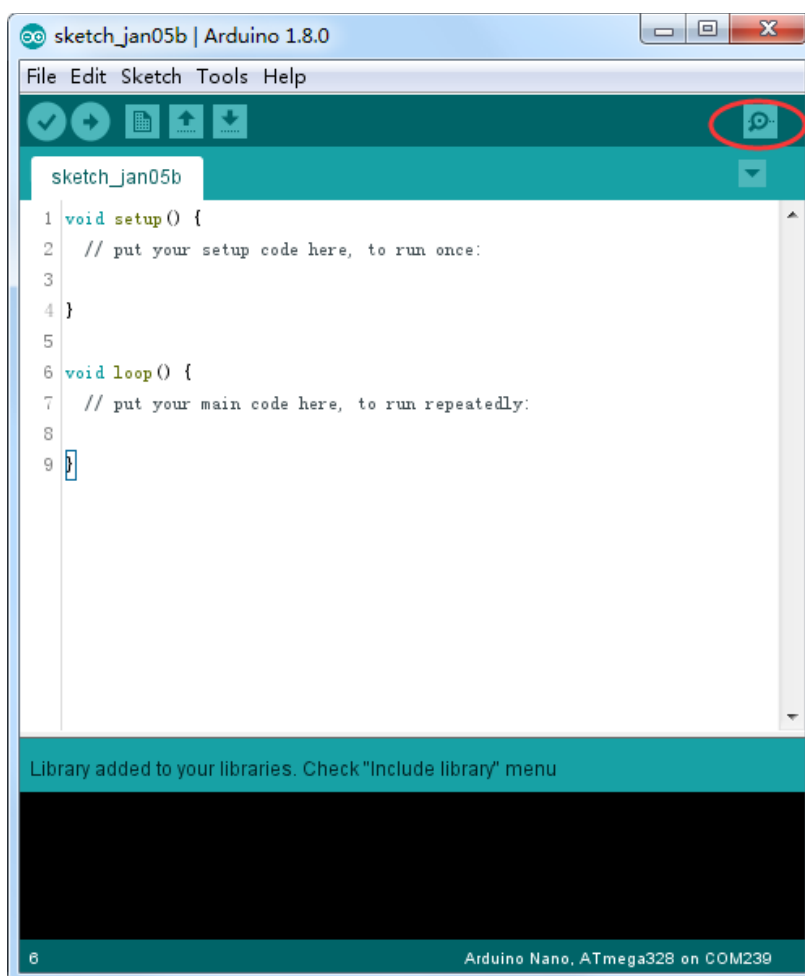
Среда разработки Arduino (IDE) – это программная сторона платформы Arduino.

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”.

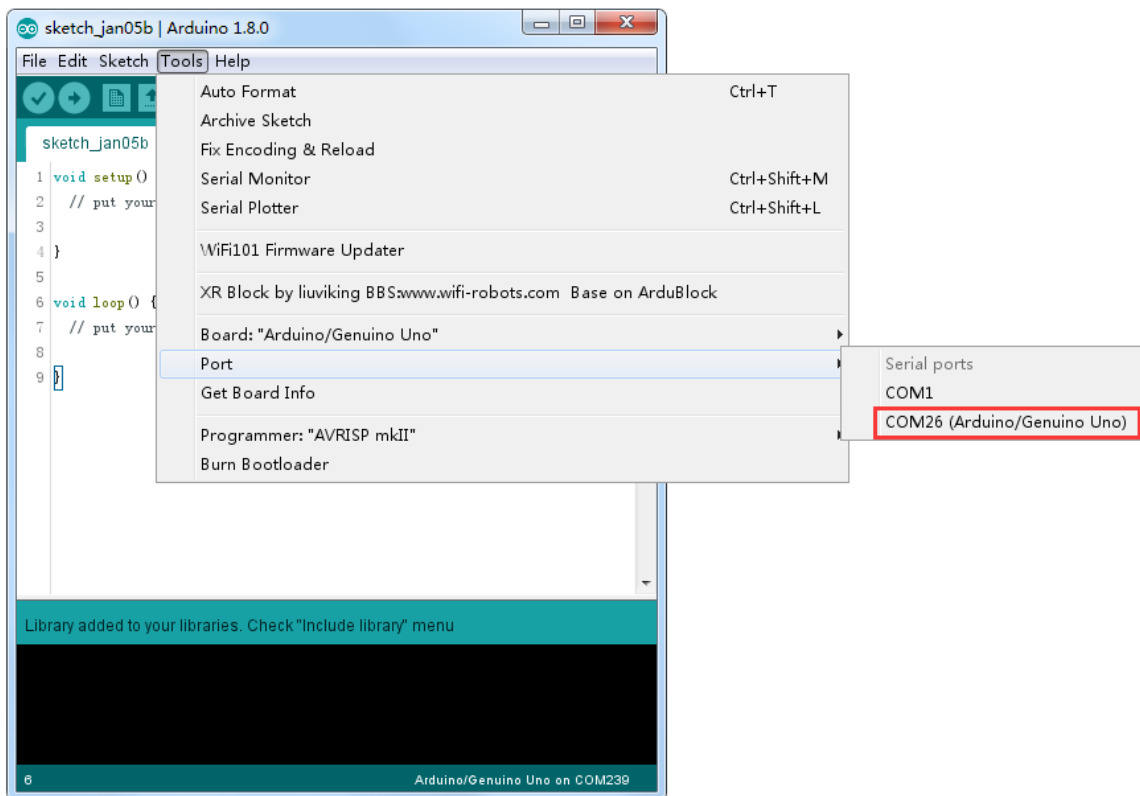
Подключение

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

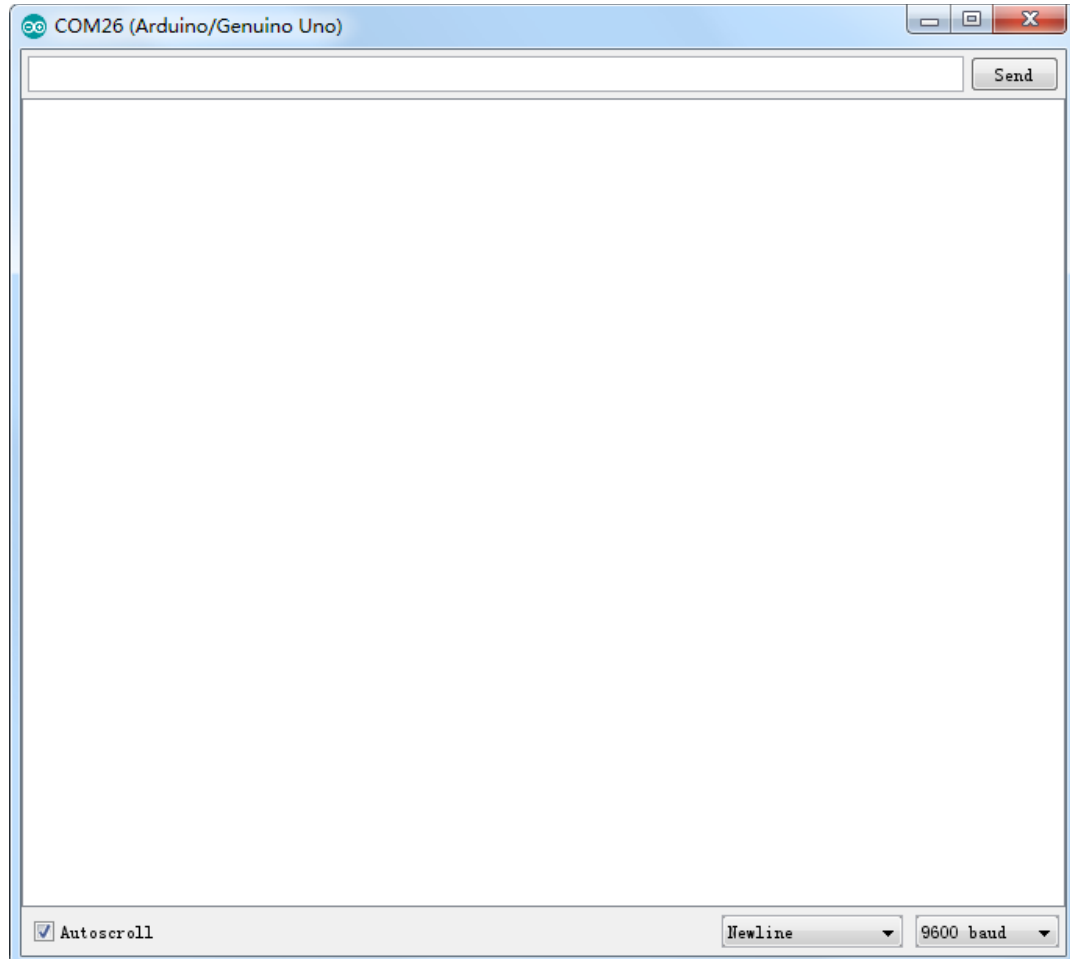


Выбор порта для отображения в серийном мониторе аналогично выбору порта для загрузки кода Arduino. Перейдите к Tools -> Serial Port и выберите необходимый порт серийного монитора.

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

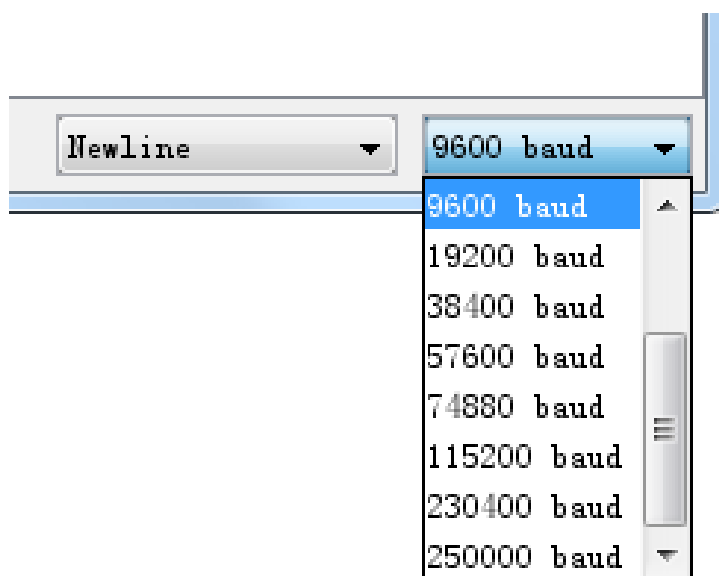


После открытия вы должны увидеть что-то, похожее на это:

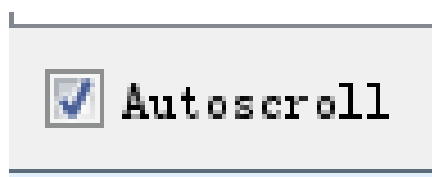


Настройки

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 (Autoscroll), 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

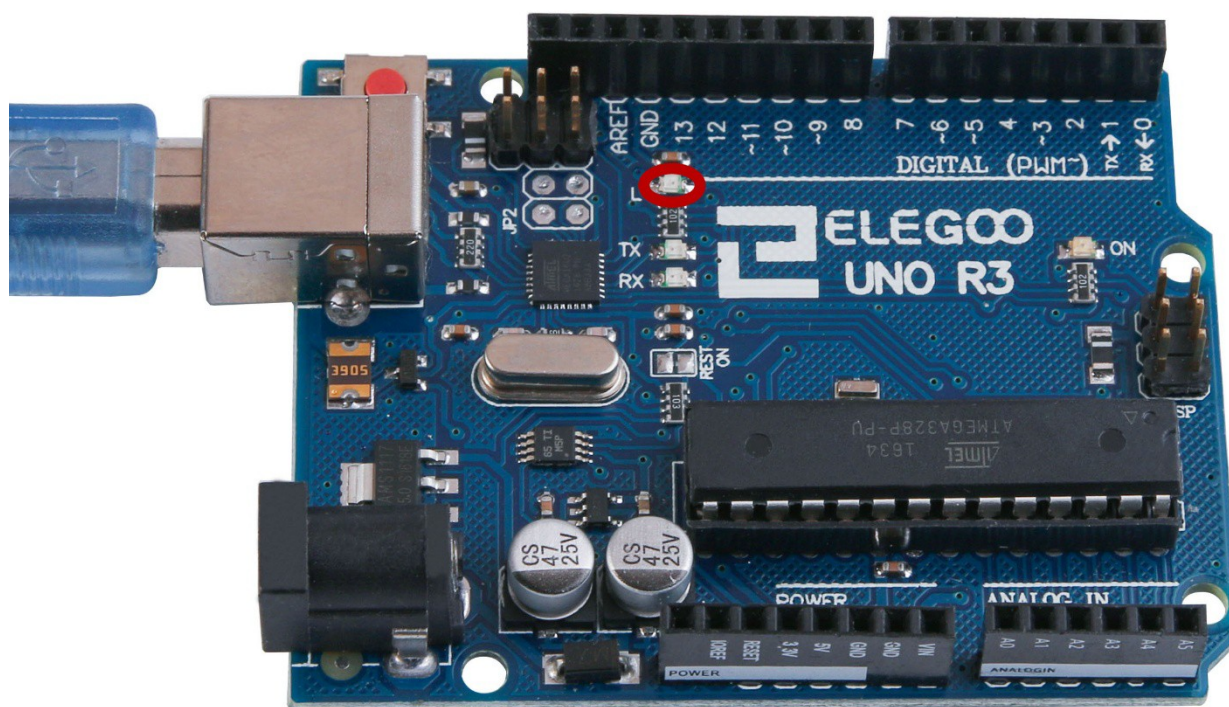
В этом уроке рассказывается, как заставить ваш UNO R3 контроллер приветственно помогать нам встроенным светодиодом, и как загружать программы шаг за шагом.

Необходимые компоненты:

- (1) х плата Elegoo Uno R3

Принцип

Плата UNO R3 имеет ряд разъемов с обеих сторон, используемых для подключения различных периферийных электронных устройств и подключаемых модулей, расширяющих операционные возможности платы – так называемых “шилдов” (“shields”). Кроме того, на плате присутствует встроенный светодиод (англ. – LED), которым Вы можете управлять с помощью скетчей. Этот встроенный светодиод условно назовем “L” светодиод, как это



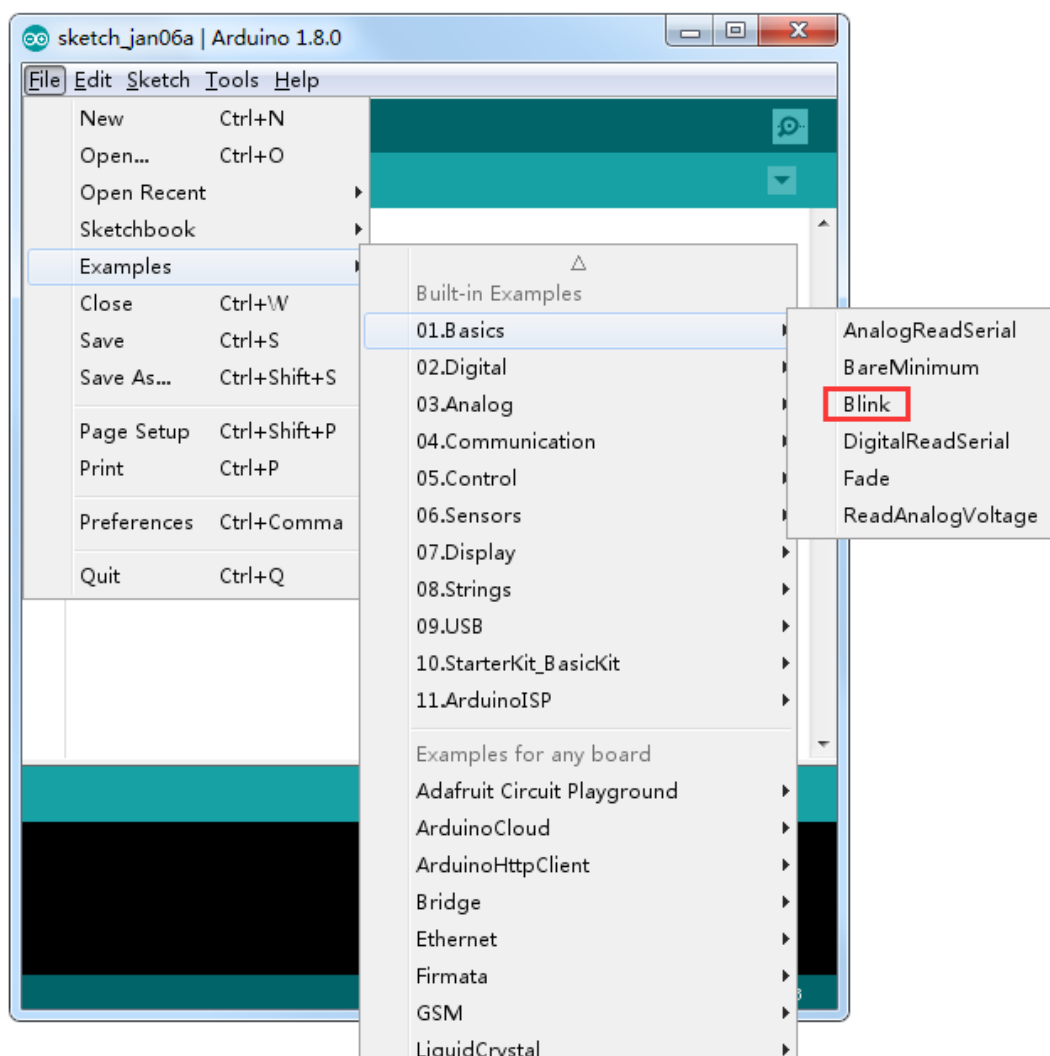
При подключении новой платы к персональному компьютеру, обратите внимание, что ваш светодиод начинает мигать, так как все платы от производителей поступают с уже “залитым” скетчем “Blink”.

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.

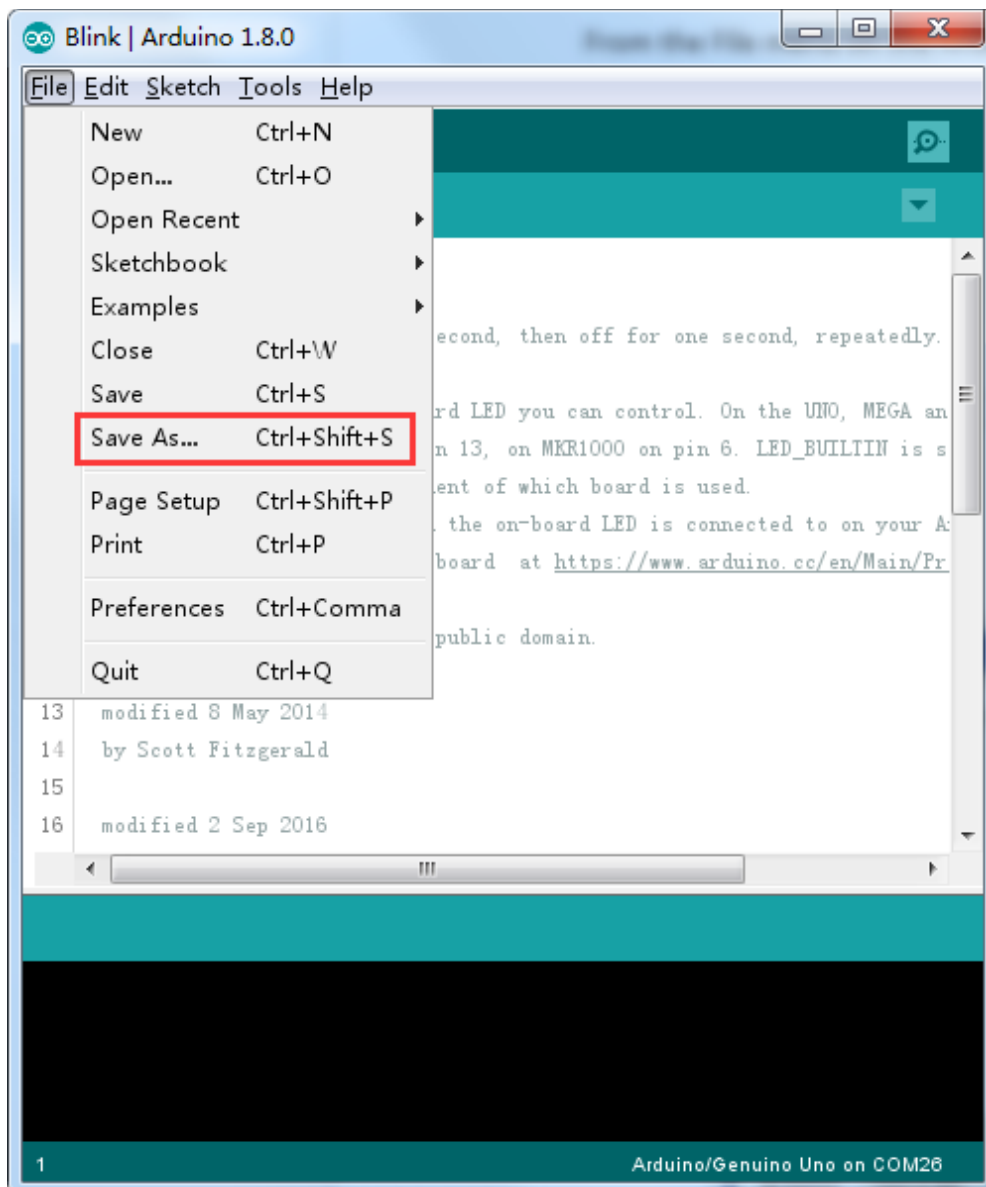
Откройте пример “Blink”, который находится в пункте меню File > Examples > 01.Basics



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

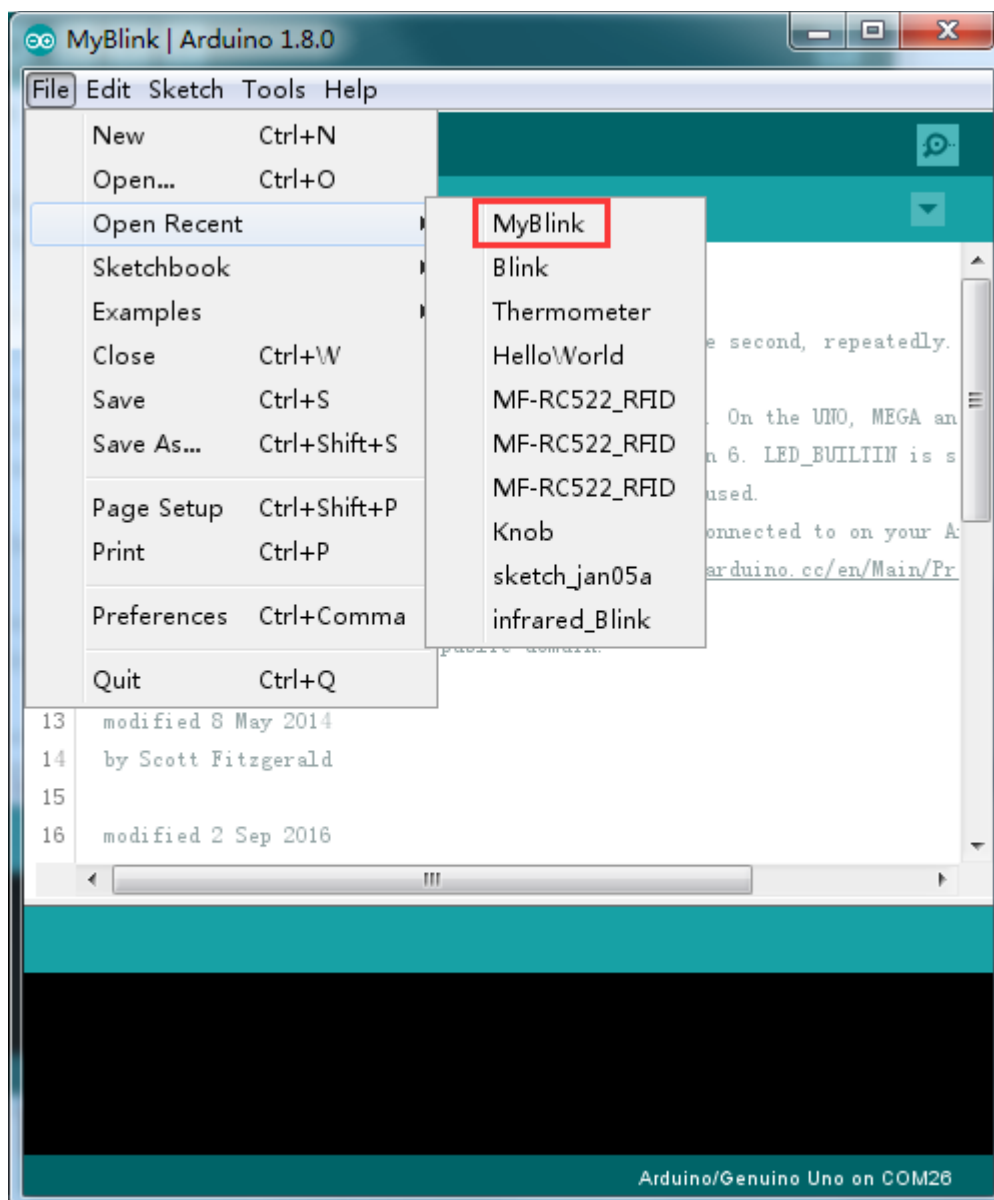


Скетчи из примеров, включенные в Arduino IDE, предусматривают режим “только чтение” (“read only”). То есть, загрузить их на плату UNO R3 вы сможете, но после изменения кода, вы не сможете их сохранить в том же файле. Мы будем изменять скетч, так что, в первую очередь, вам необходимо сохранить собственную копию, которую вы сможете изменять. Из меню “File” выберите опцию “Сохранить как” (“Save As..”) и сохраните скетч под подходящим вам названием, например, “MyBlink”.

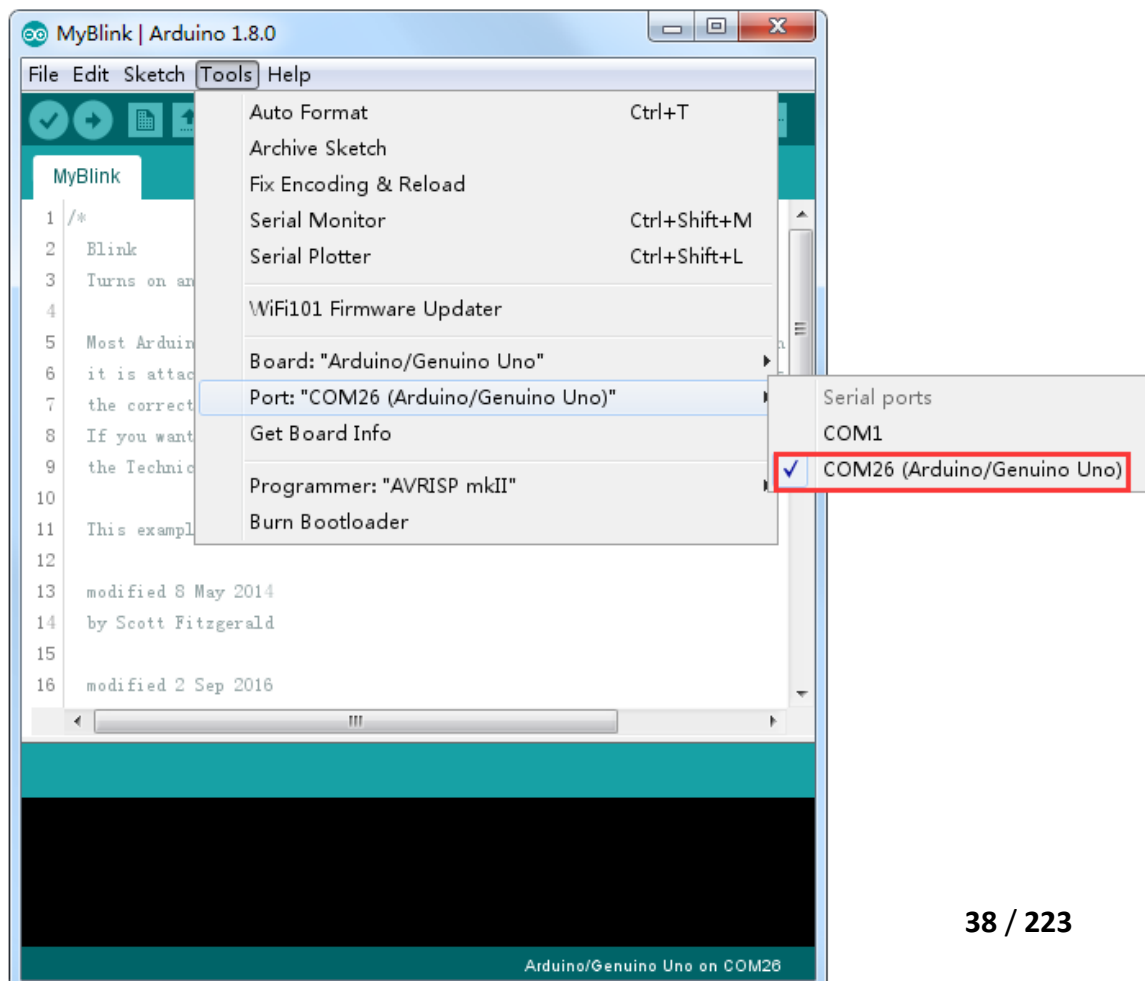
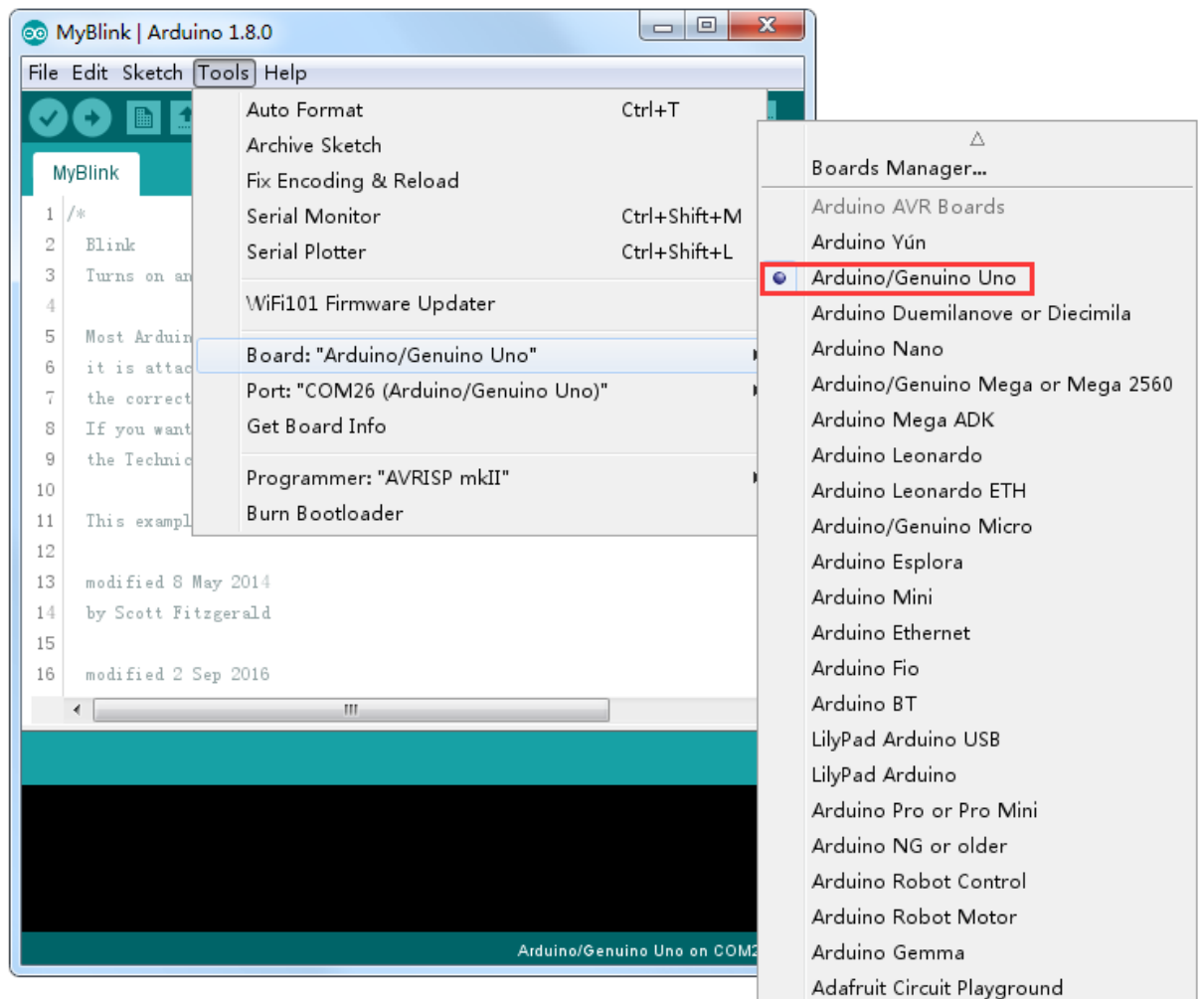




Вы сохранили копию скетча “Blink” в вашей библиотеке. Теперь открыть этот файл вы можете в любой момент, перейдя по вкладке File > Sketchbook.

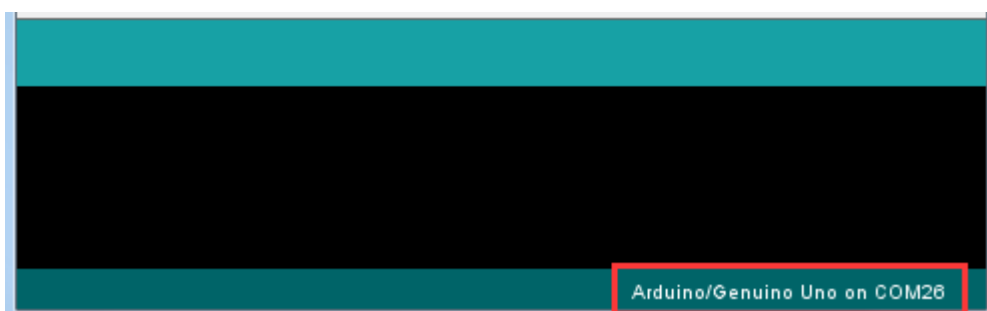


Подключите свою плату Arduino к компьютеру с помощью USB и проверьте тип платы ("Board type") и серийный порт ("Serial Port"), по которому она подключена.



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.

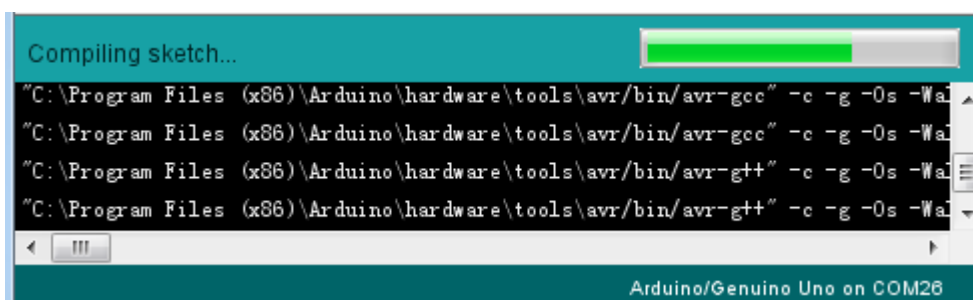
Текущие настройки отображаются внизу окна оболочки Arduino IDE



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



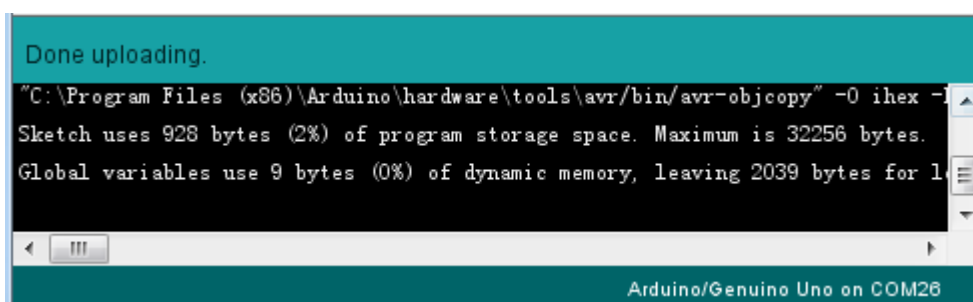
Во время загрузки в нижней части окна IDE появятся ползунок загрузки и сообщения. Вначале появляется фраза “Компилирование” (“Compiling sketch...”), что означает процесс конвертирования вашего скетча в формат, подходящий для загрузки на плату Arduino.



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.

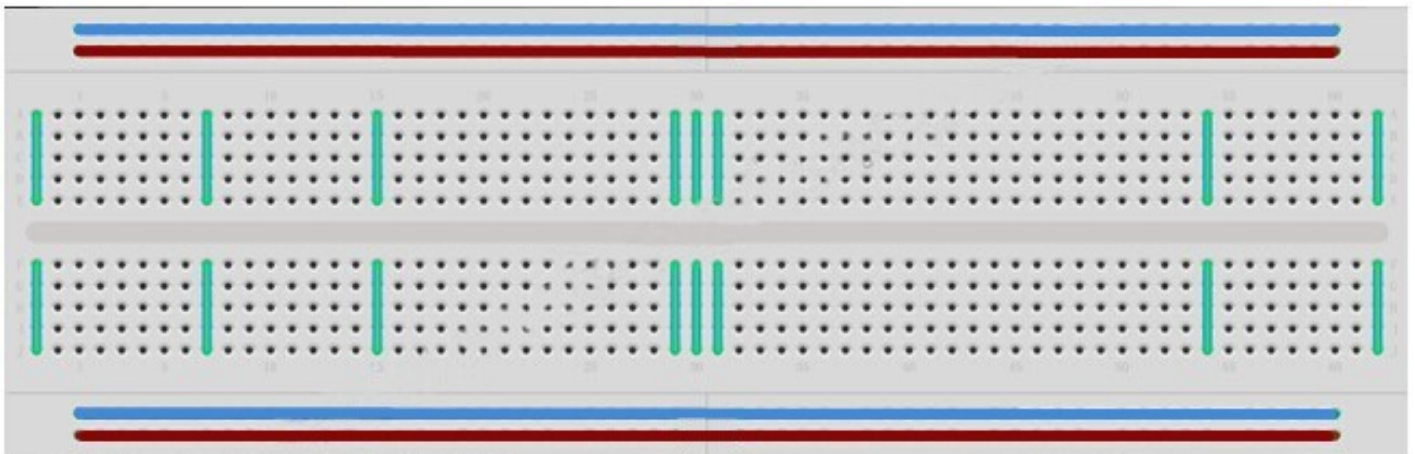
Необходимые компоненты:

- (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-разъемов.

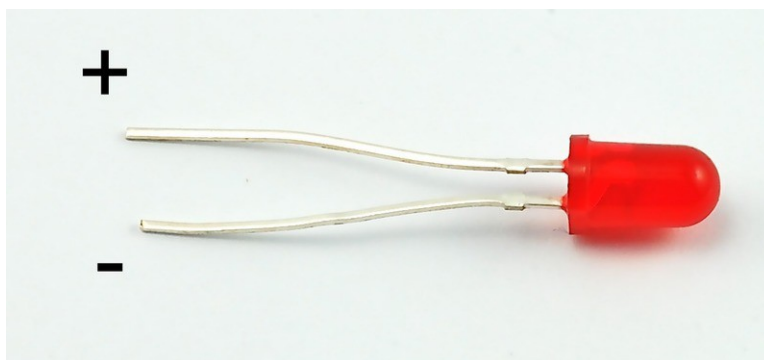
В то время, как макетные платы – прекрасный инструмент для создания прототипов, они, всё же, имеют некоторые ограничения. Поскольку соединения элементов на плате являются временными, они менее надежные чем распайка. Поэтому присутствие временных неисправностей в цепи вызвано, скорее всего, неплотными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



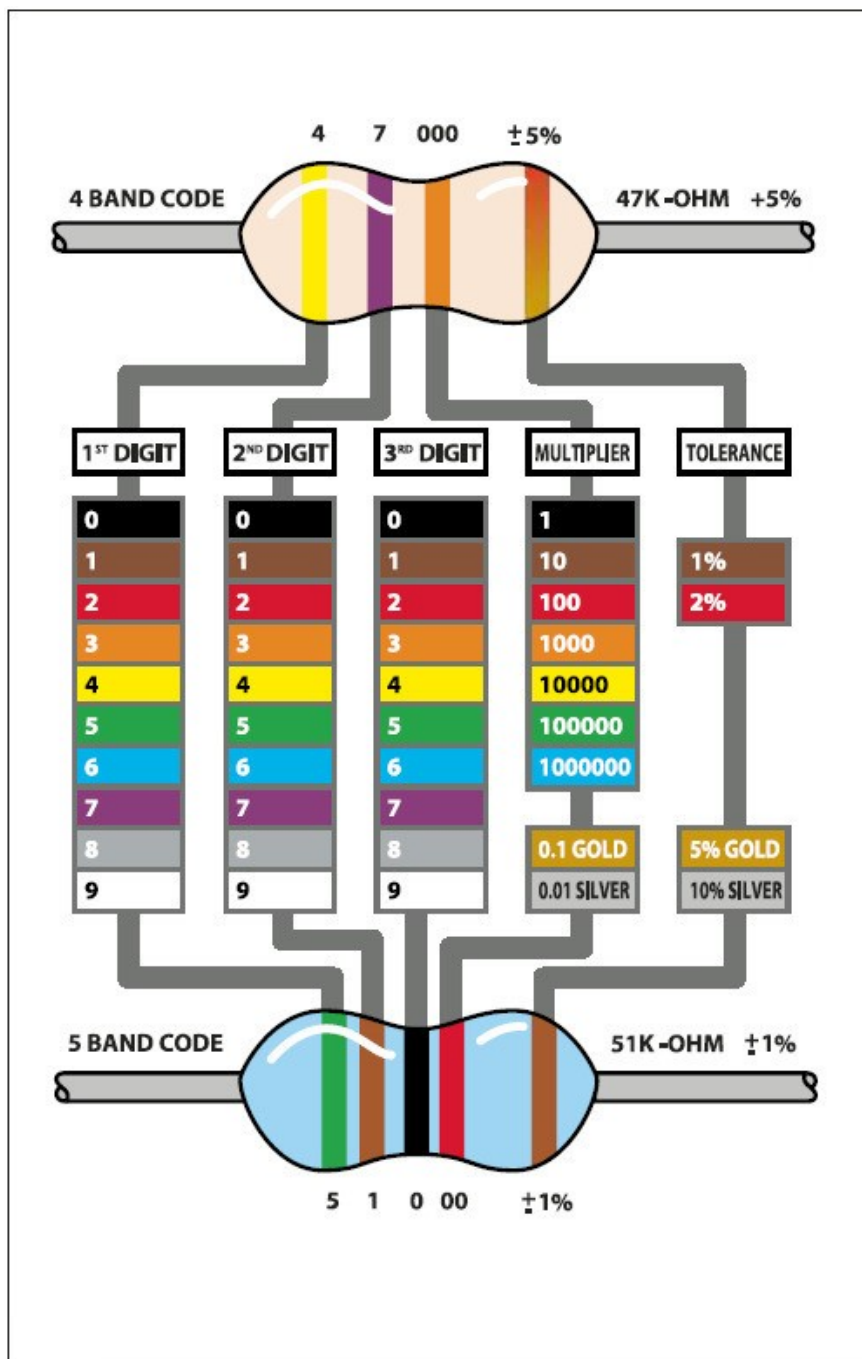
Но сначала погорим немного о резисторах...

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 полосы. Именно по этим полоскам визуально определяется номинал резисторов.

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



В отличие 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

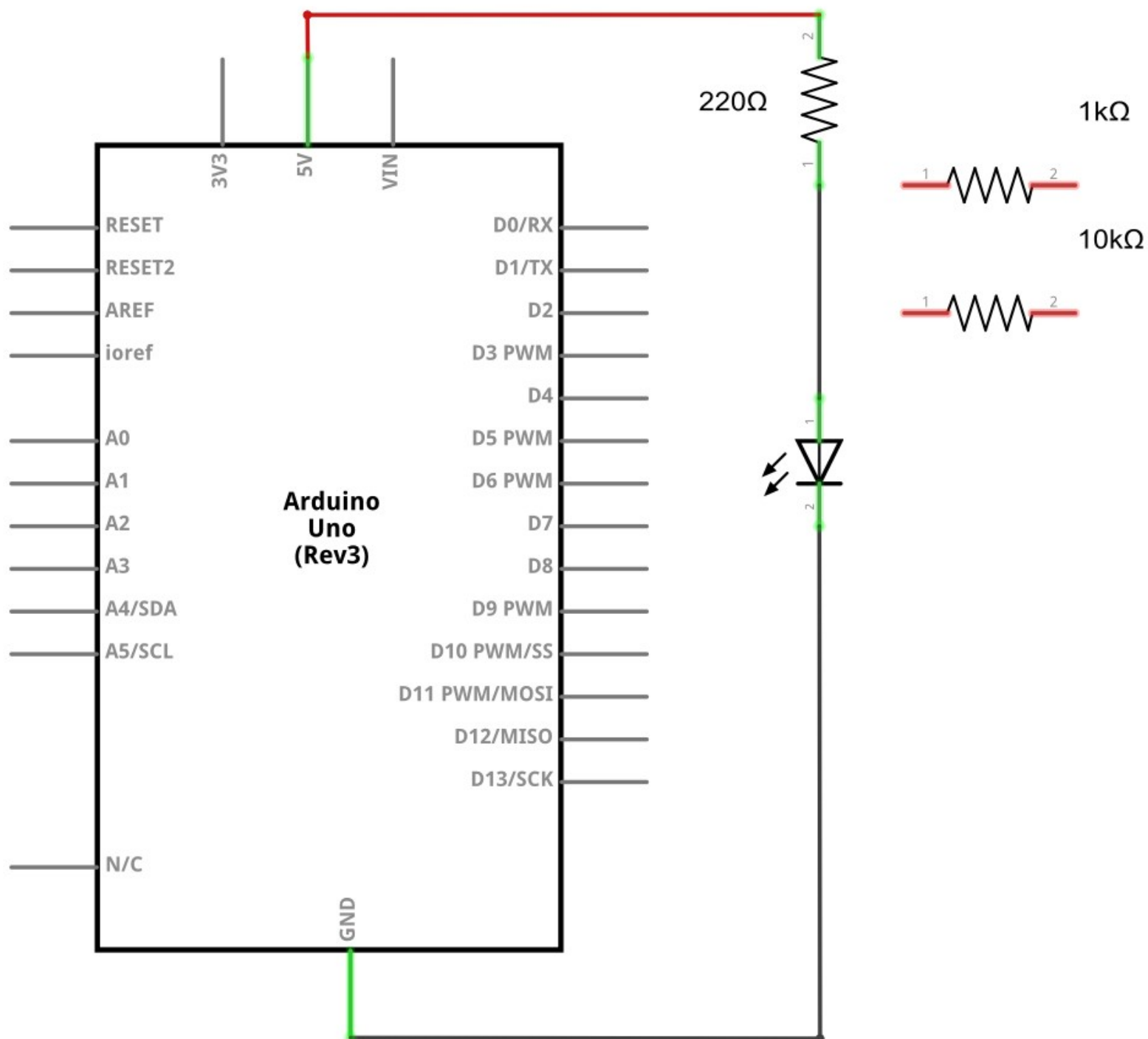
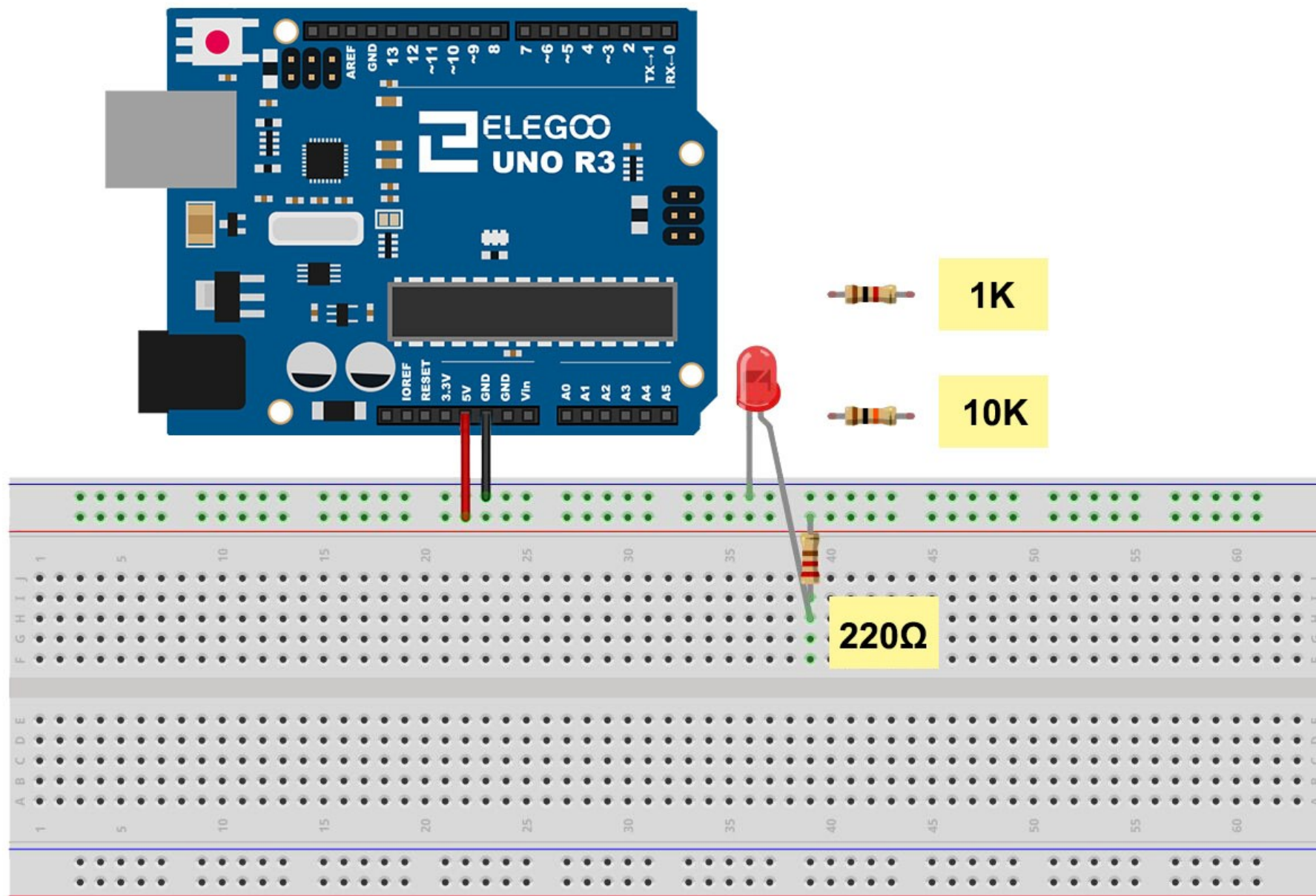


Схема монтажа

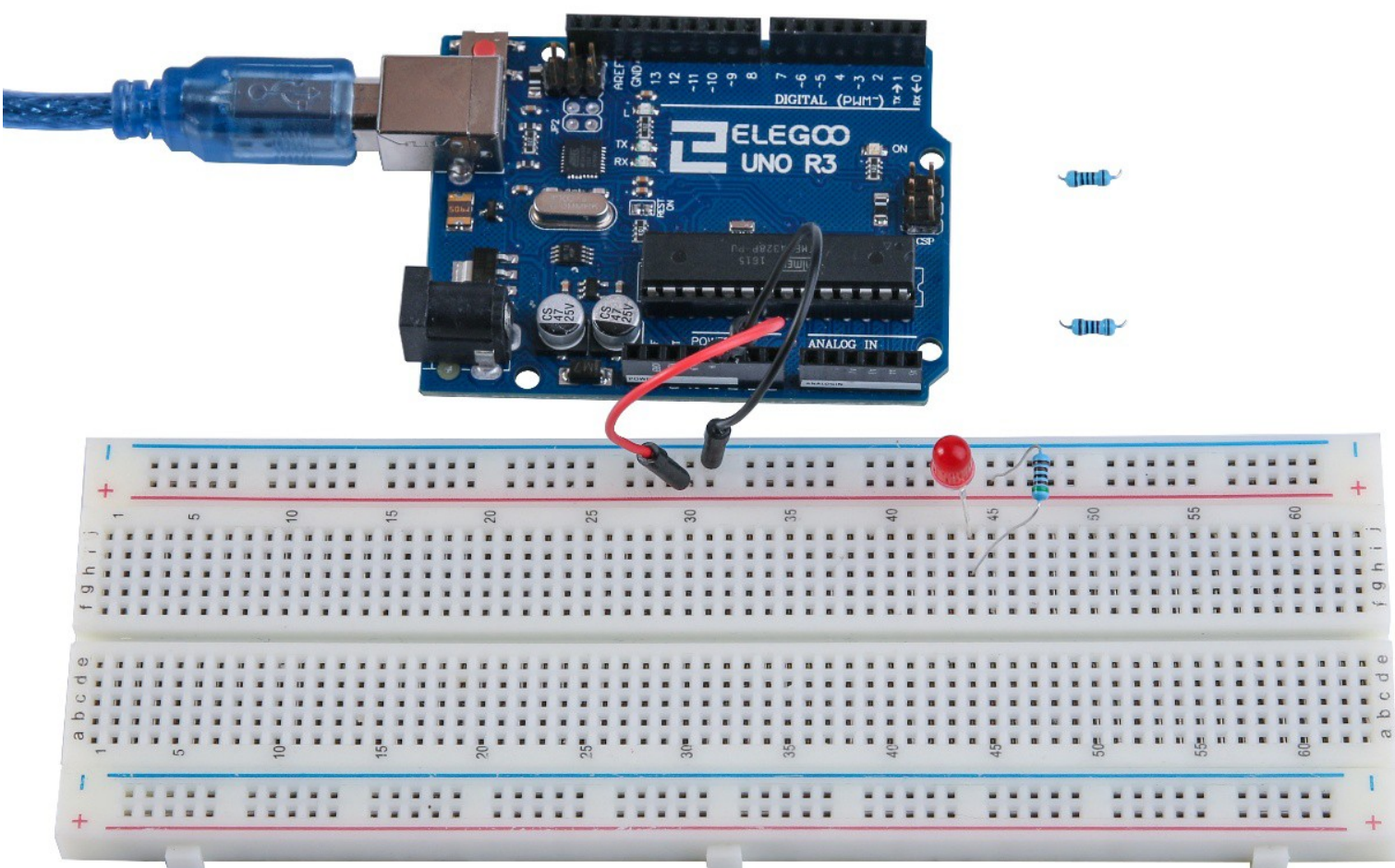


На Arduino есть пин на 5 В для питания периферийных устройств. Мы будем использовать его для питания светодиода и резистора. Больше вам от платы ничего не потребуется, только лишь подключить ее через USB к компьютеру.

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, используя 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) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (4) x перемычки (типа "папа"- "папа")
- (1) x RGB светодиод
- (3) x резистора (220 Ом)

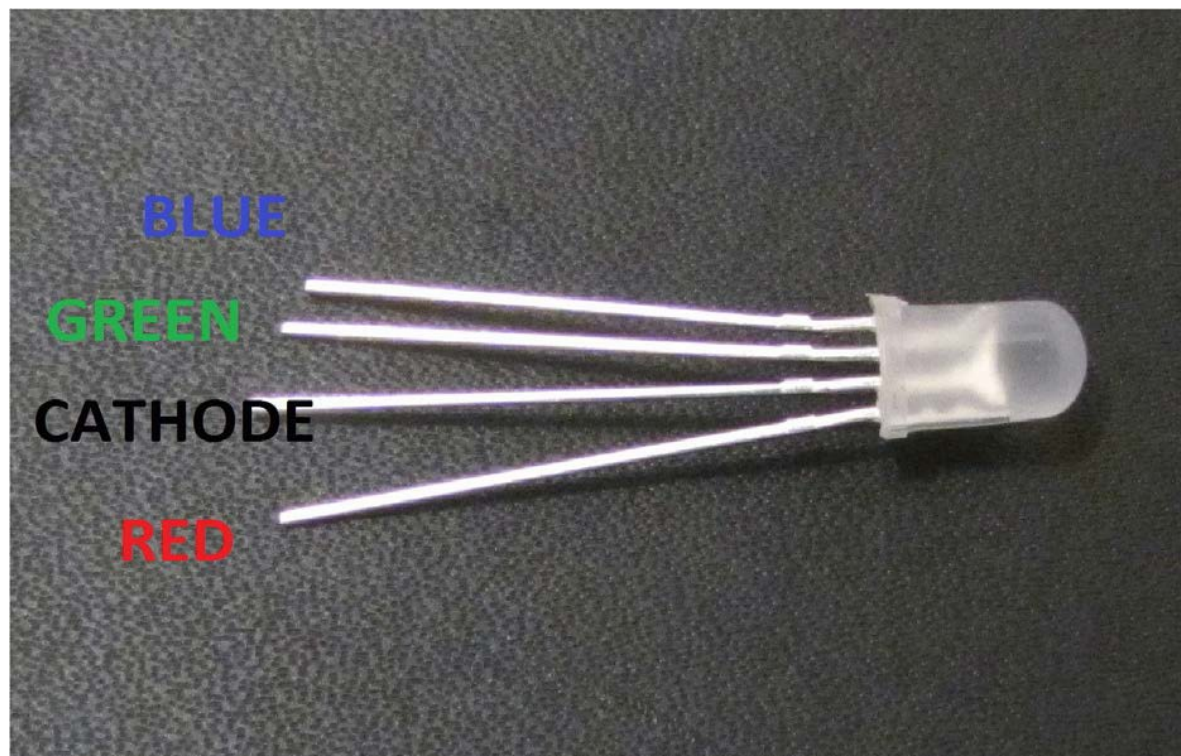
Представление компонентов

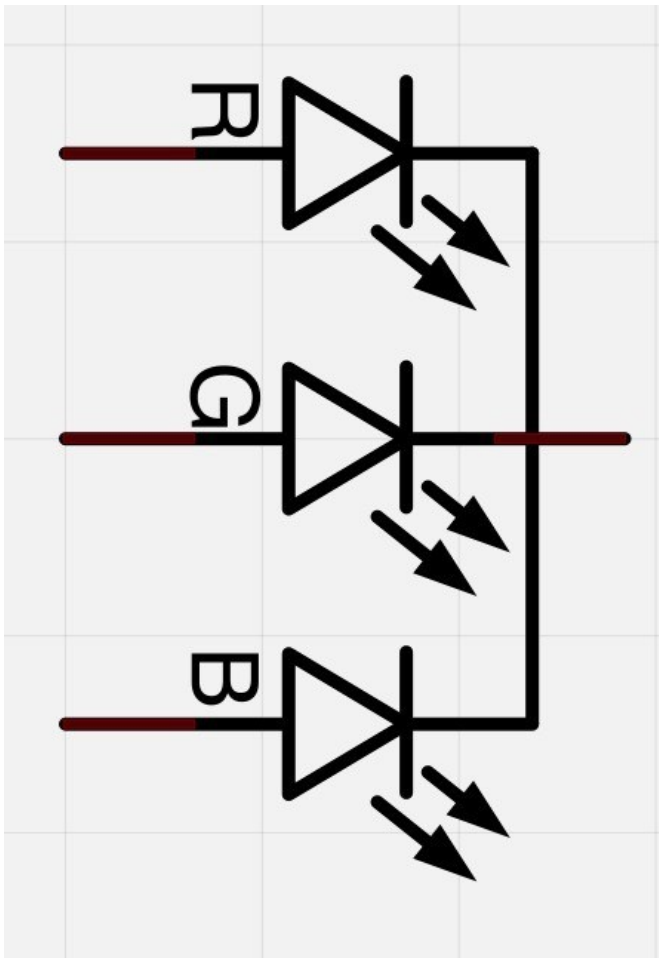
RGB светодиод:

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.

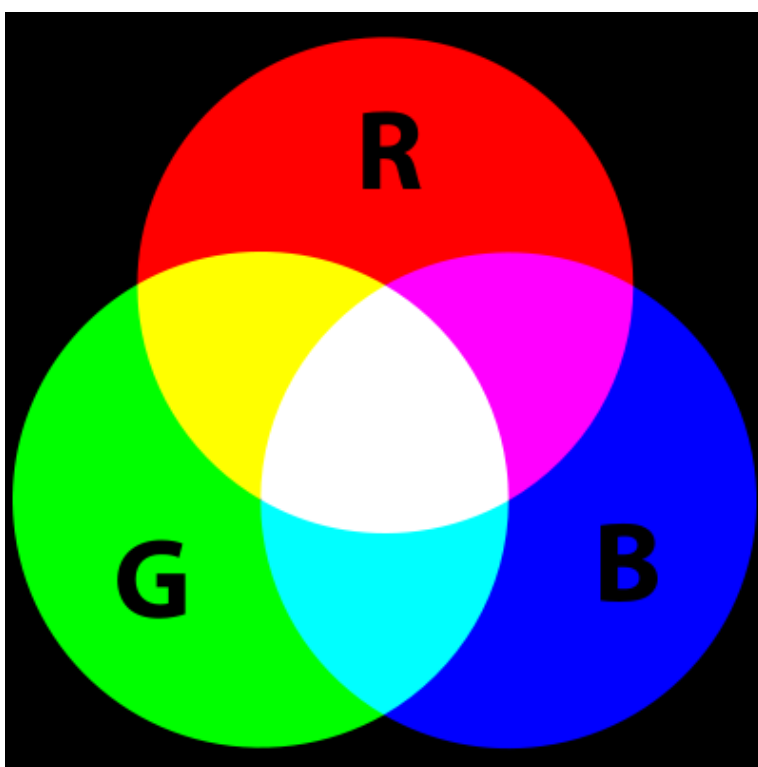
Общий анод на RGB светодиоде – это второй по счету, самый длинный контакт. Этот контакт мы подключим к земле (GND).

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.

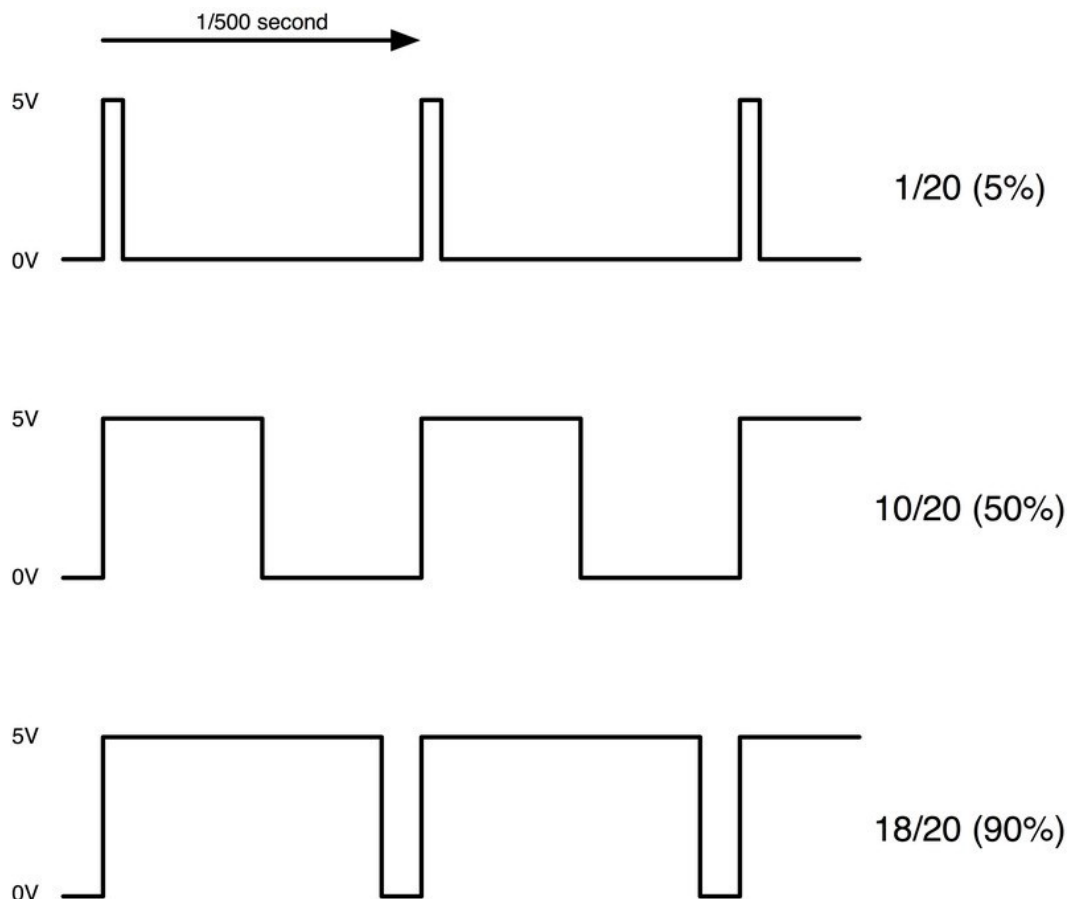
Мы можем управлять яркостью каждого светодиода отдельно, смешивая цвета как нам заблагорассудится.

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 длиться 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

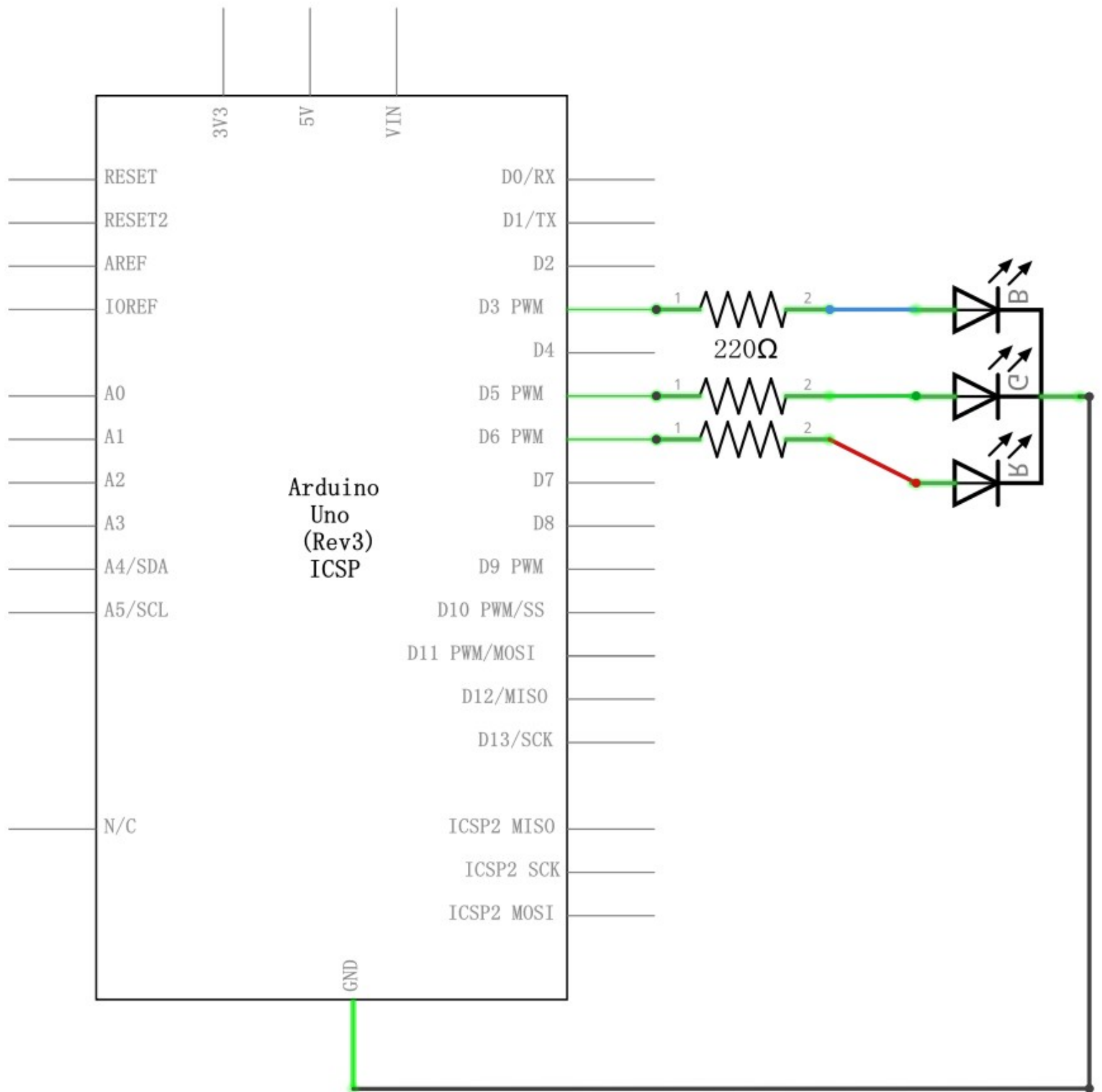
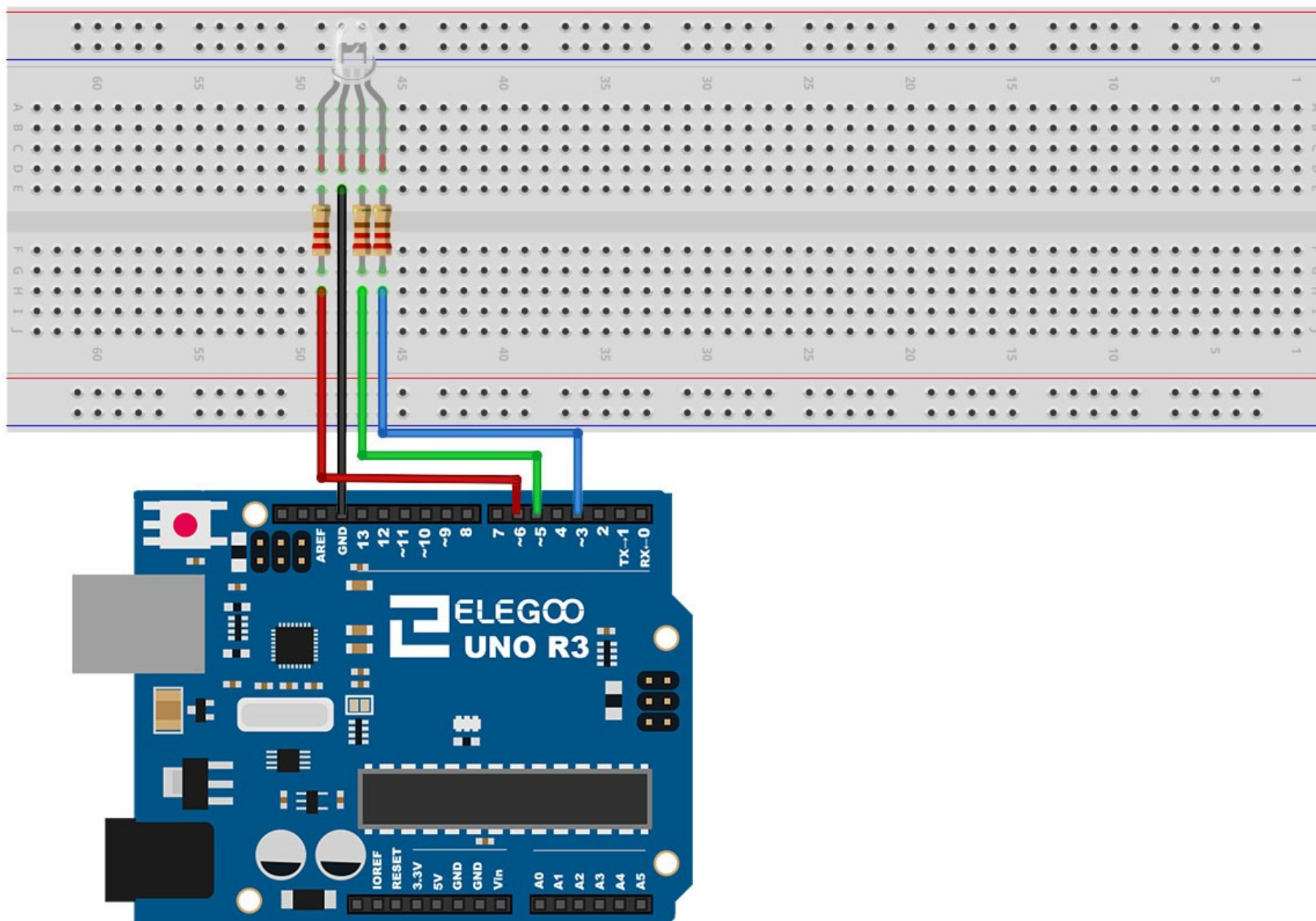
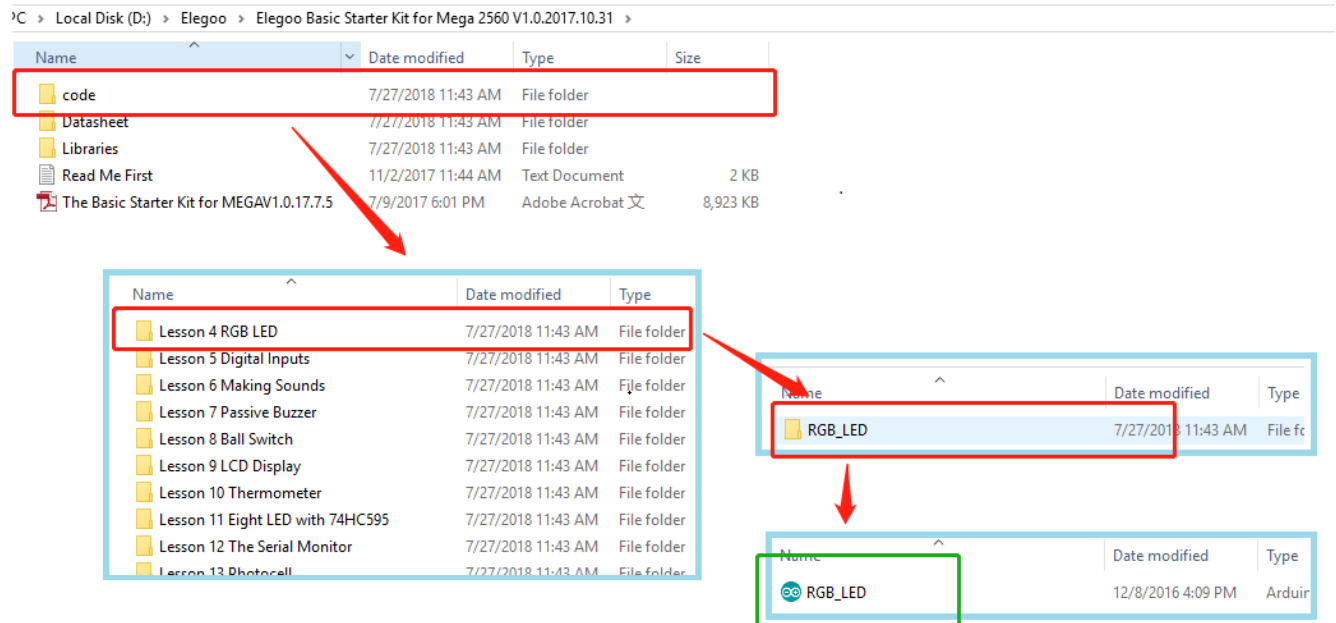


Схема монтажа



Code

After wiring, please open the Sketch in folder path: Tutorial > English > code > Lesson 4 RGB LED > RGB_LED, and click UPLOAD to upload the program.



See Lesson 2 for details about program uploading if there are any errors.

The sketch starts by specifying which pins are going to be used for each of the colors:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

The next step is to write the 'setup' function. As we have learnt in earlier lessons, the setup function runs just once after the Arduino has reset. In this case, all it has to do is define the three pins we are using as being outputs.

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

Определенные переменные

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

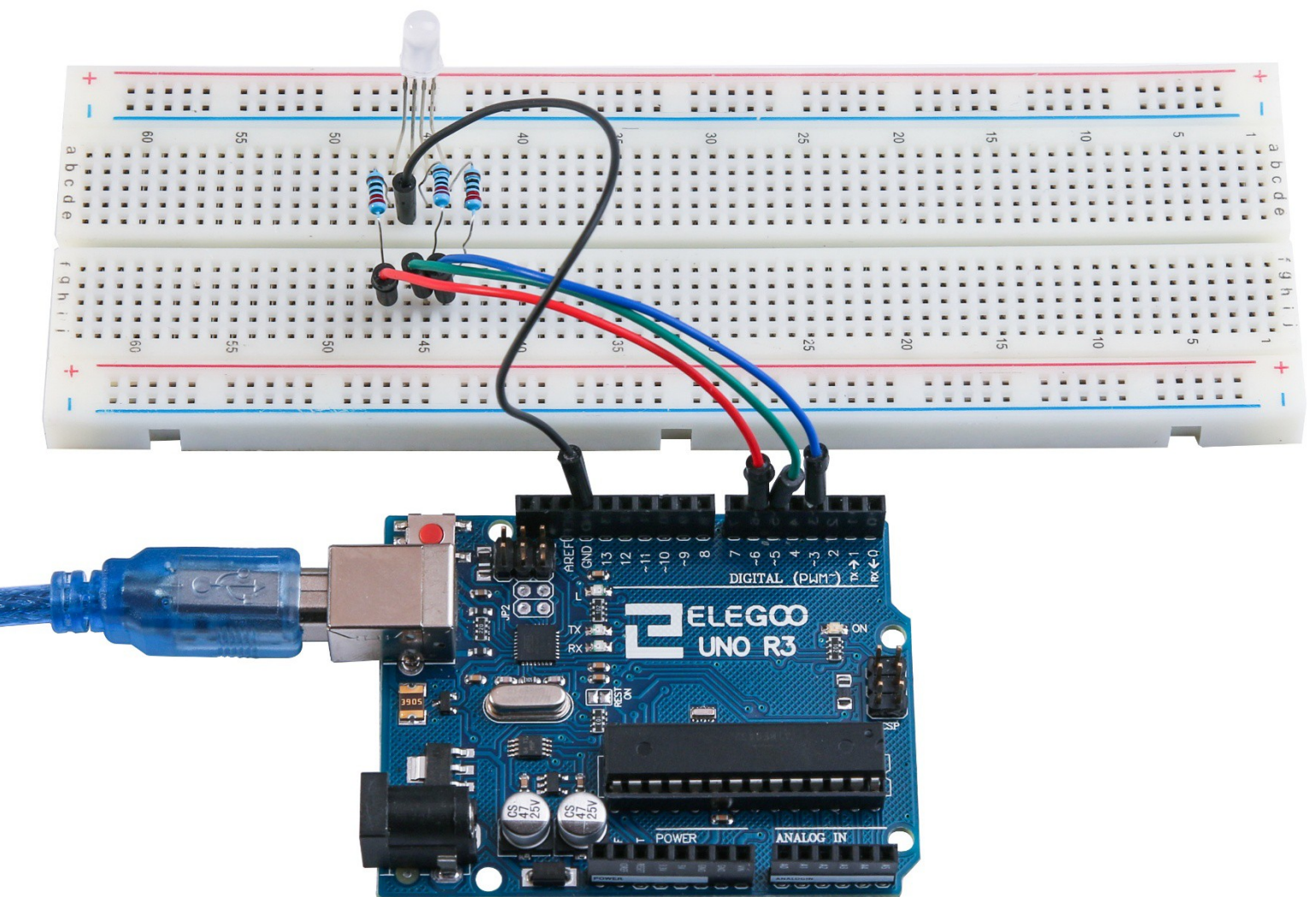
Эта функция принимает три аргумента. Один отвечает за яркость красного, зеленого и синего светодиодов. В каждом из них значение варьируется в диапазоне от 0 до 255, где 0 означает, что он полностью выключен, а 255 – максимальная яркость. После этого функция вызывает 'analogWrite', чтобы настроить яркость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

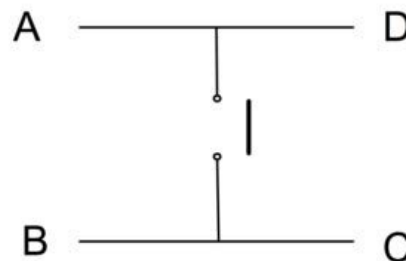
Представление компонентов

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 сбивать 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

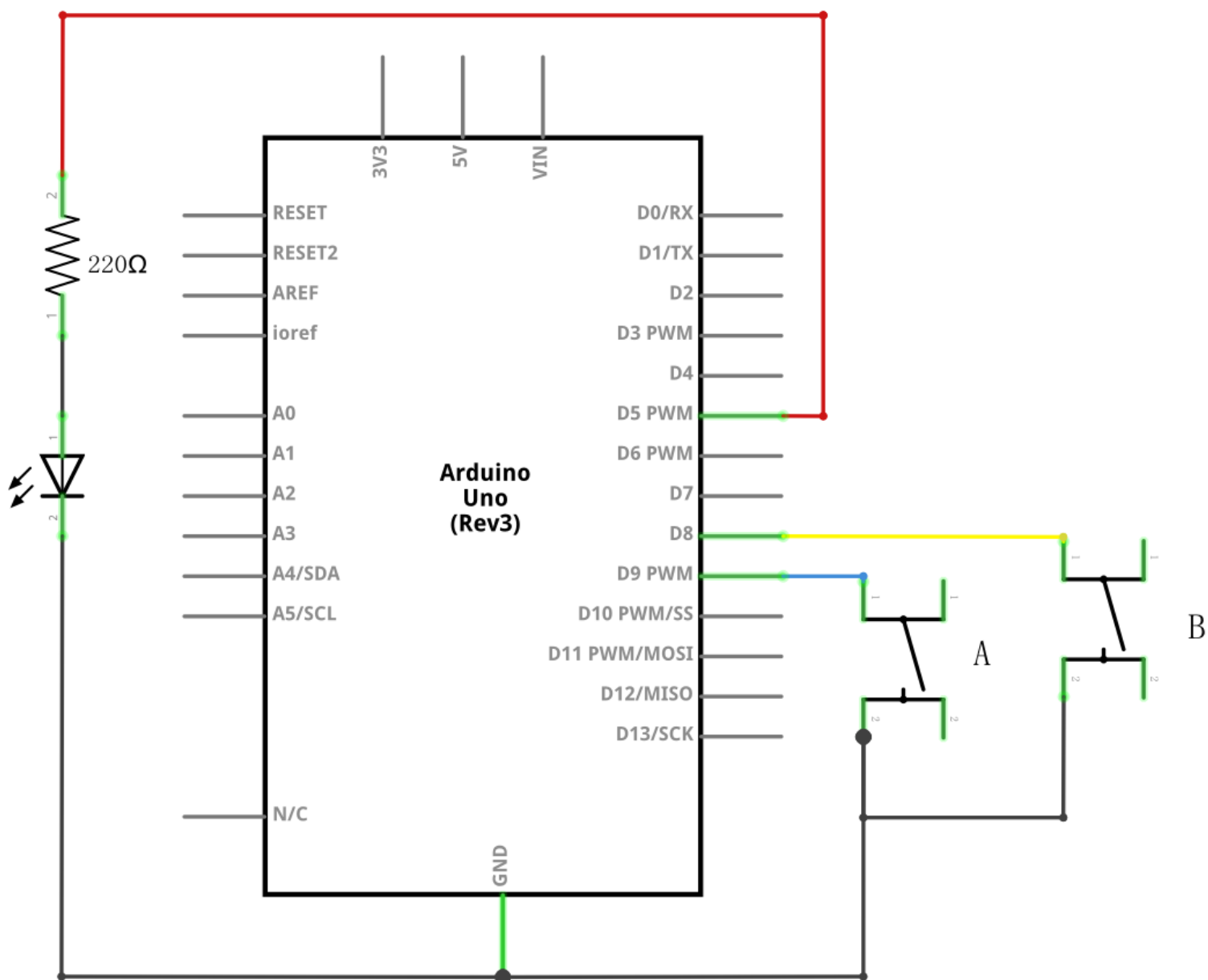
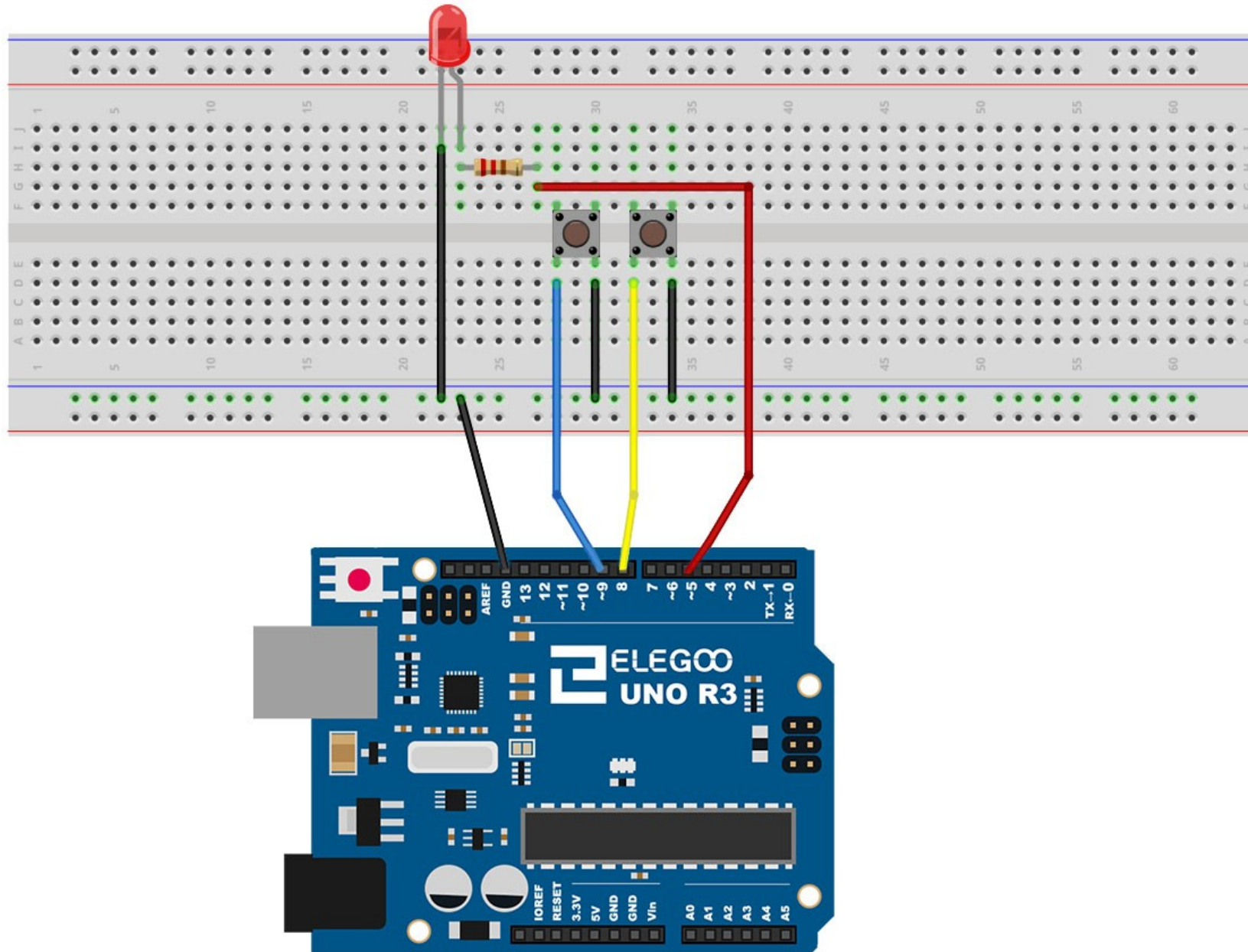


Схема монтажа



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);
```

“INPUT_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 А находится в состоянии “LOW”, то 'digitalWrite' на ledPin включит светодиод. Аналогично, если нажата кнопка В, то в ledPin запишется “LOW”.

A photograph of an Arduino Uno R3 microcontroller board connected to a breadboard. The breadboard contains a red LED, a resistor, and several jumper wires. A USB cable is plugged into the Arduino's USB port.

Урок 6: Пьезоизлучатель со встроенным генератором (активный зуммер)

Обзор

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

Необходимые компоненты:

- (1) х плата Elegoo Uno R3
- (1) х активный зуммер
- (2) х 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 изоляцией – 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 пределах 2 – 5 кГц, чтобы зуммер начал «генерировать» звук. Зачастую, активный зуммер дороже, поскольку может содержать несколько встроенных колебательных контуров.



Подключение

Схема

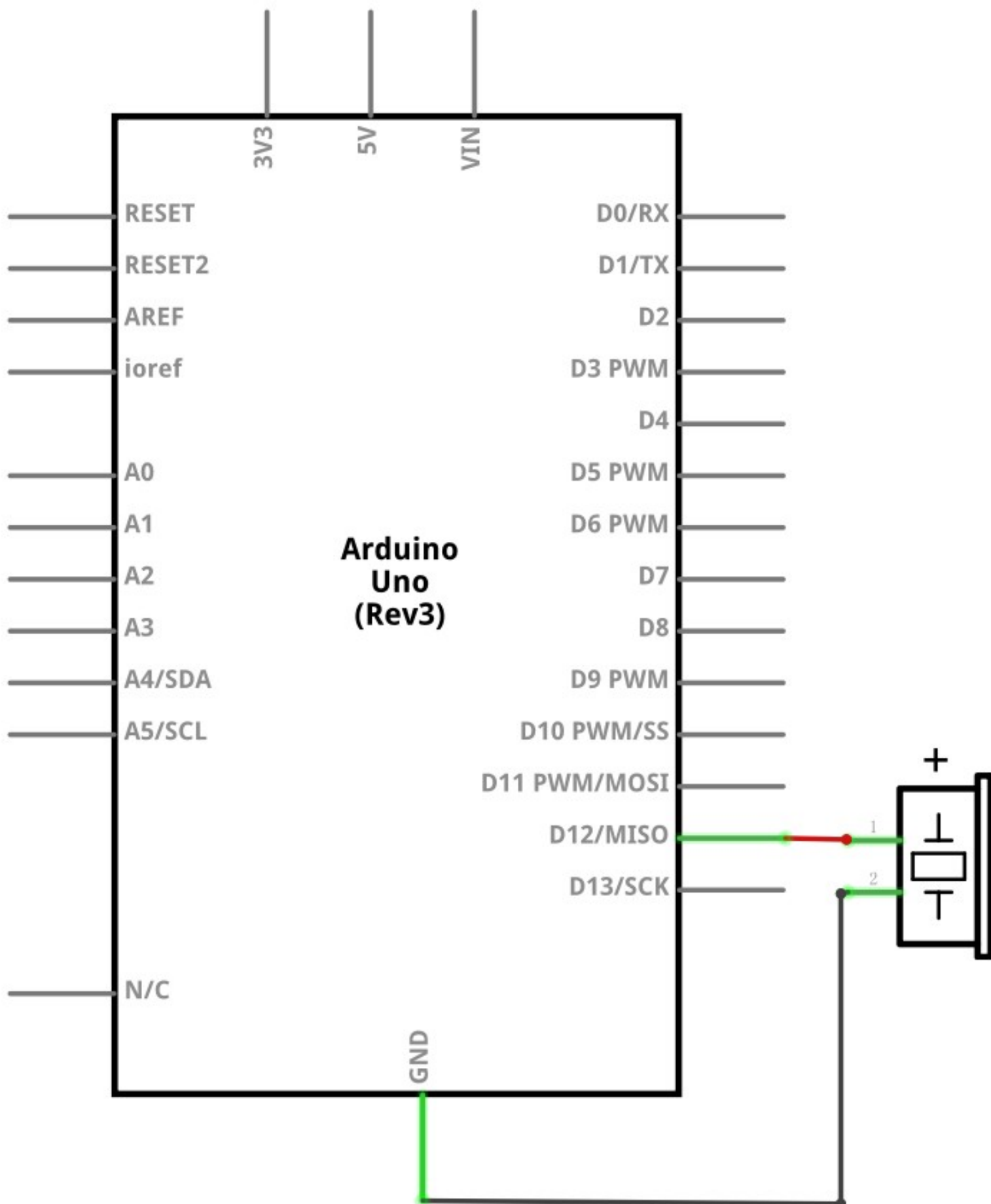
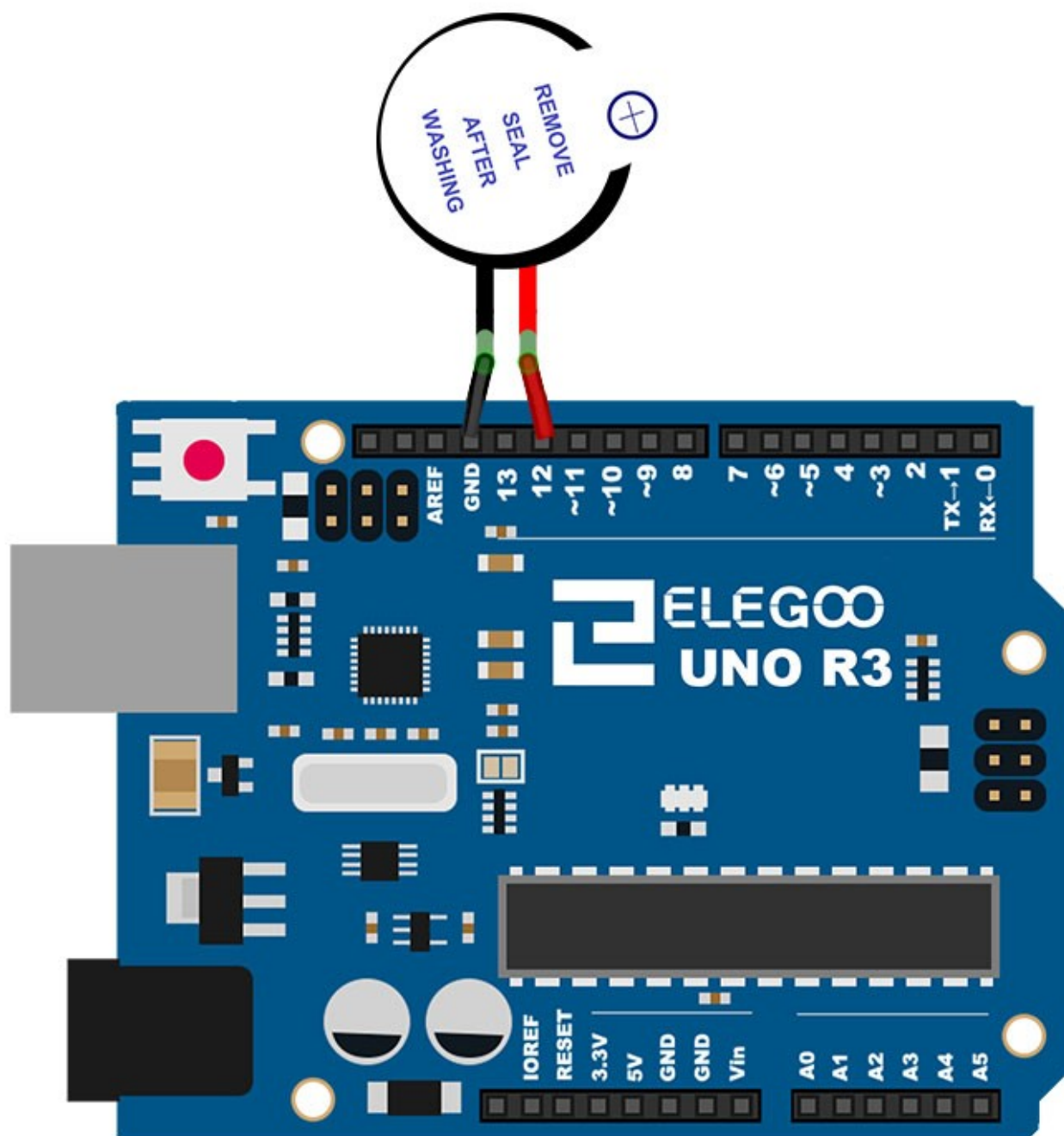


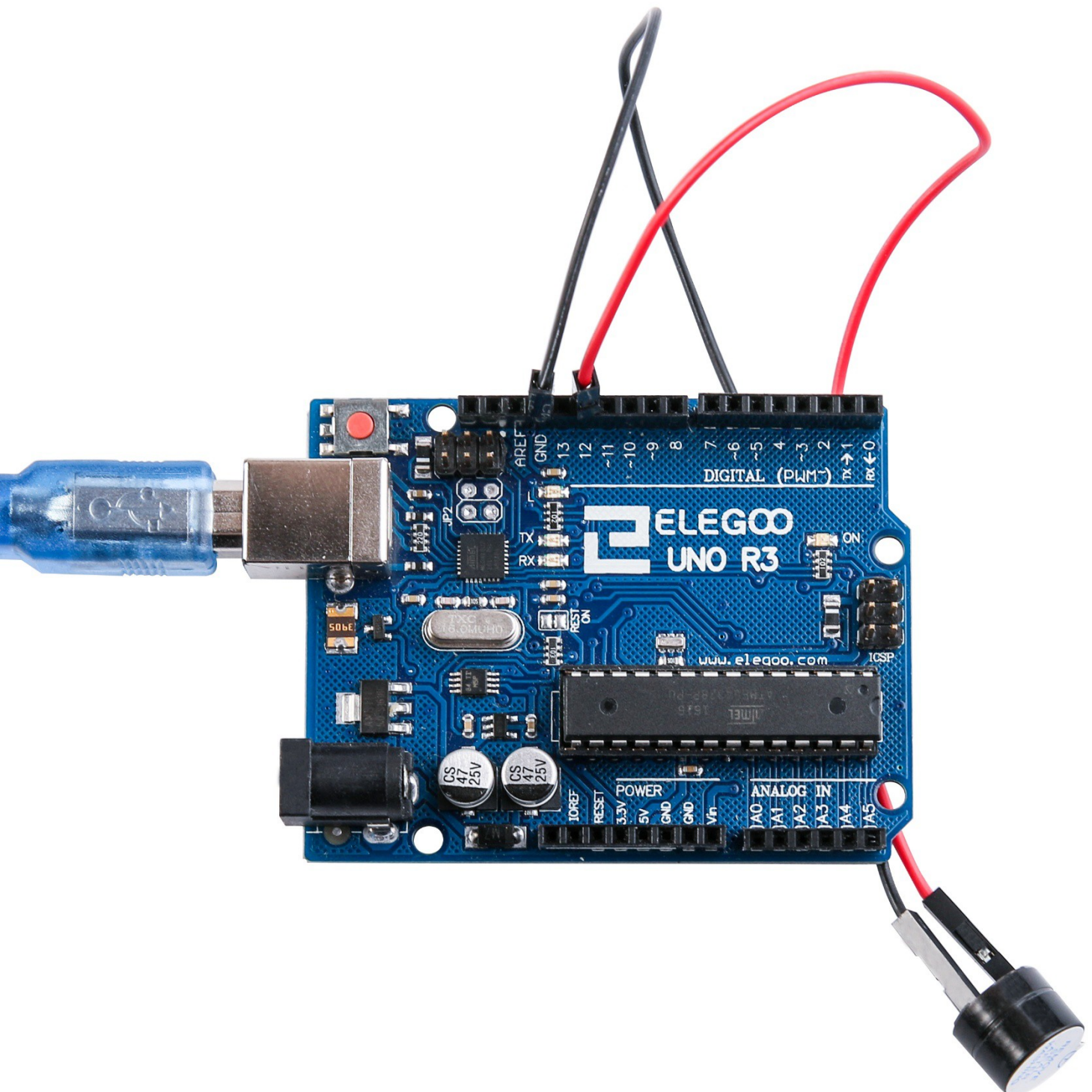
Схема монтажа



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: Пьезоизлучатель без встроенного генератора (пассивный зуммер)

Обзор

В этом уроке вы научитесь использовать пассивный зуммер.

Цель эксперимента – сгенерировать восемь различных звуков длительностью 0.5 секунд каждый: от Альт До (523 Гц), Ре (587 Гц), Ми (659 Гц), Фа (698 Гц), Соль (784 Гц), Ля (880 Гц), Си (988 Гц) до Дискант До (1047 Гц).

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x пассивный зуммер
- (2) x перемычки DuPont (типа “мама” - “папа”)



Представление компонентов

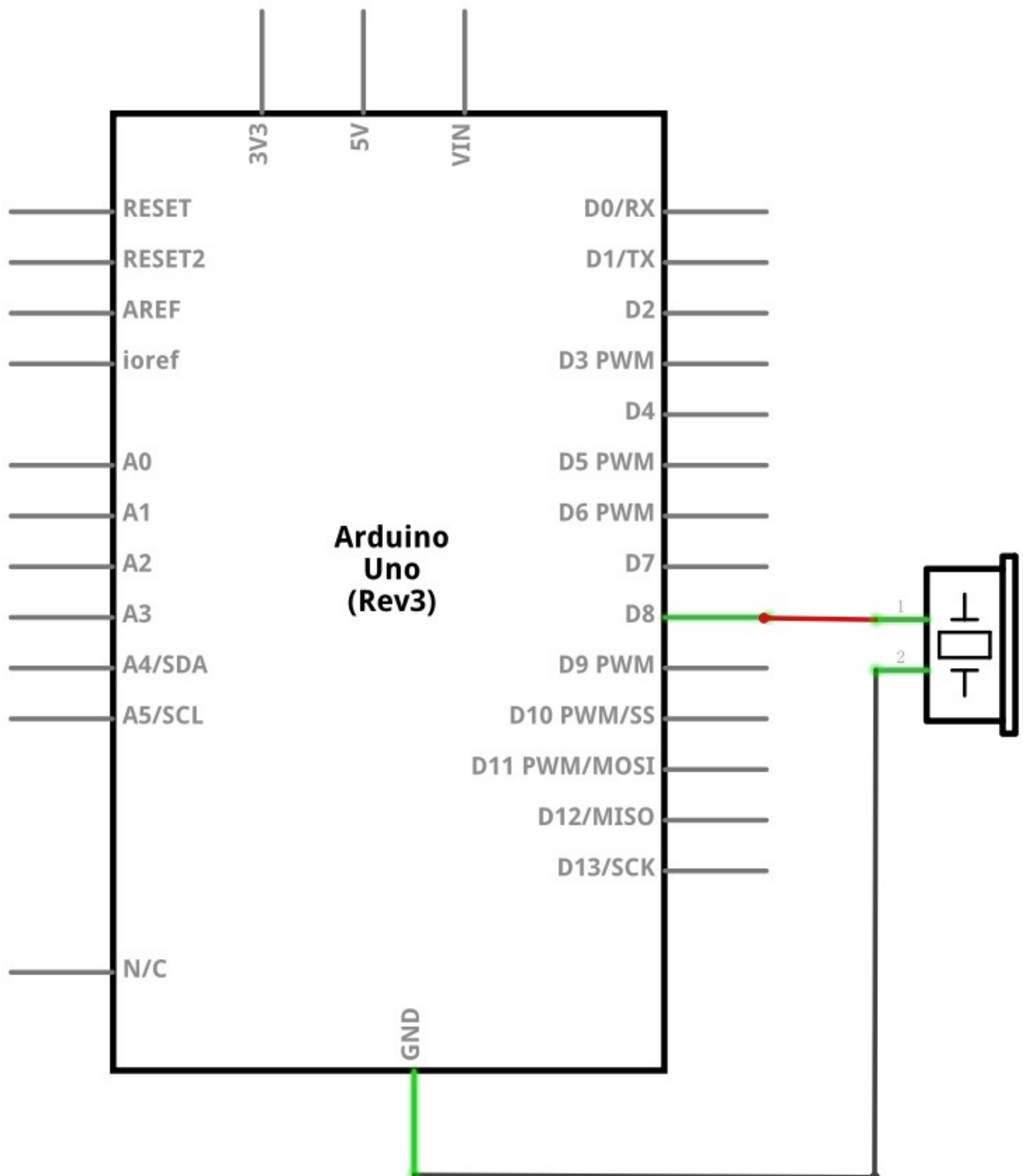
Пассивный зуммер:

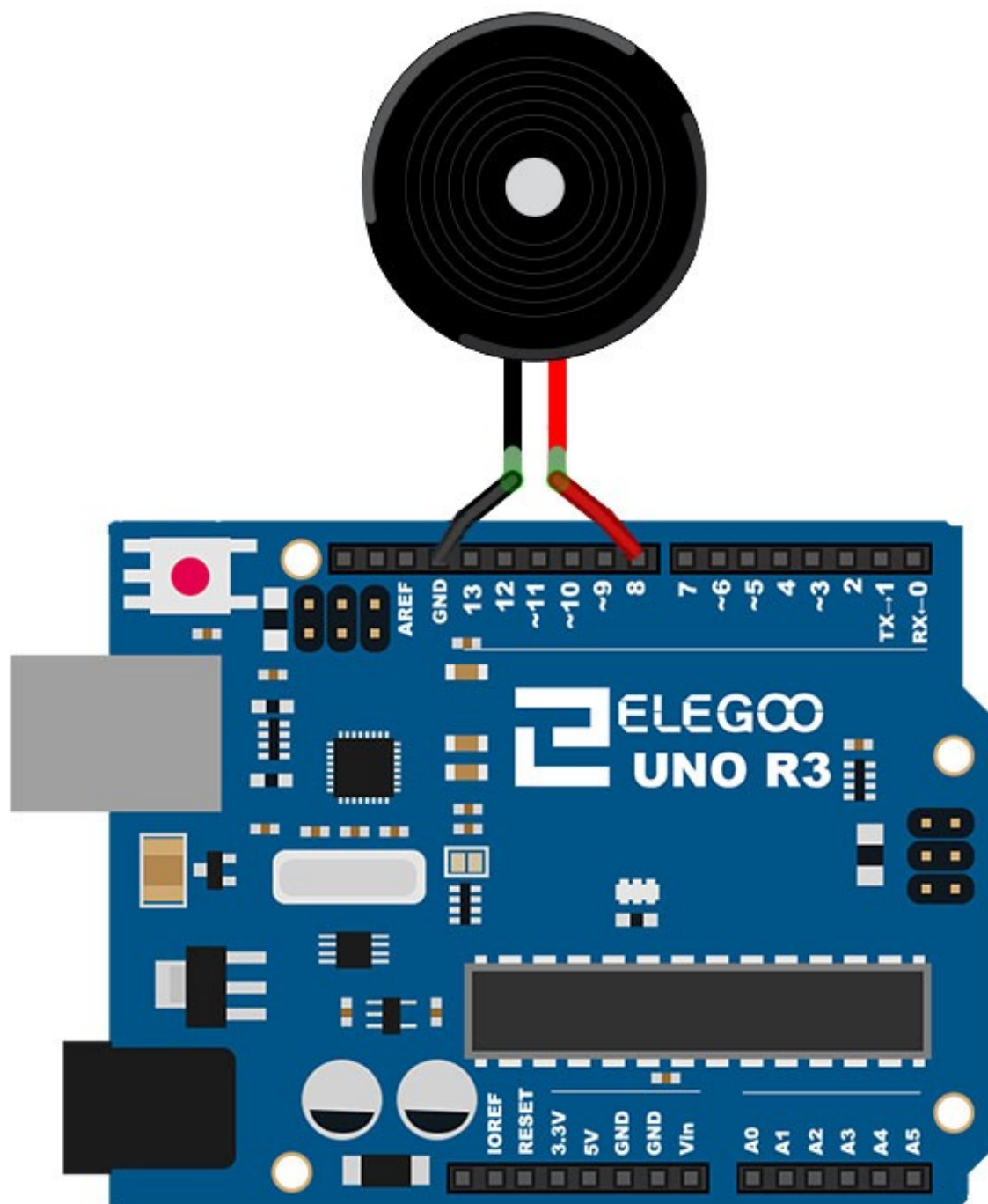
Пассивный зуммер (пьезоизлучатель, или пьезоэлектрический излучатель) – это электроакустическое устройство воспроизведения звука, использующие обратный пьезоэлектрический эффект. Принцип его действия основан на том, что под действием электрического поля возникает механическое движение мембраны, которое и вызывает слышимые нами звуковые волны. То есть, звук будет генерироваться с частотой, поданной с внешнего генератора (нашей платы UNO R3). Например, посылая на зуммер импульс частотой 523 Гц, мы можем получить Альт До; импульс частотой 587 Гц – Ре среднего регистра; импульс частотой 659 Гц – Ми среднего регистра, и т.д. Используя зуммер, мы можем играть музыку.

Мы должны быть внимательны и не использовать аналоговую функцию “Write ()” для генерации импульса на зуммер, поскольку эта функция генерирует импульс постоянного значения (500 Гц).

Соединение

Схема





Подключите зуммер к плате UNO R3: красный (положительный) провод – к пину 8, черный (общий провод) – к GND.

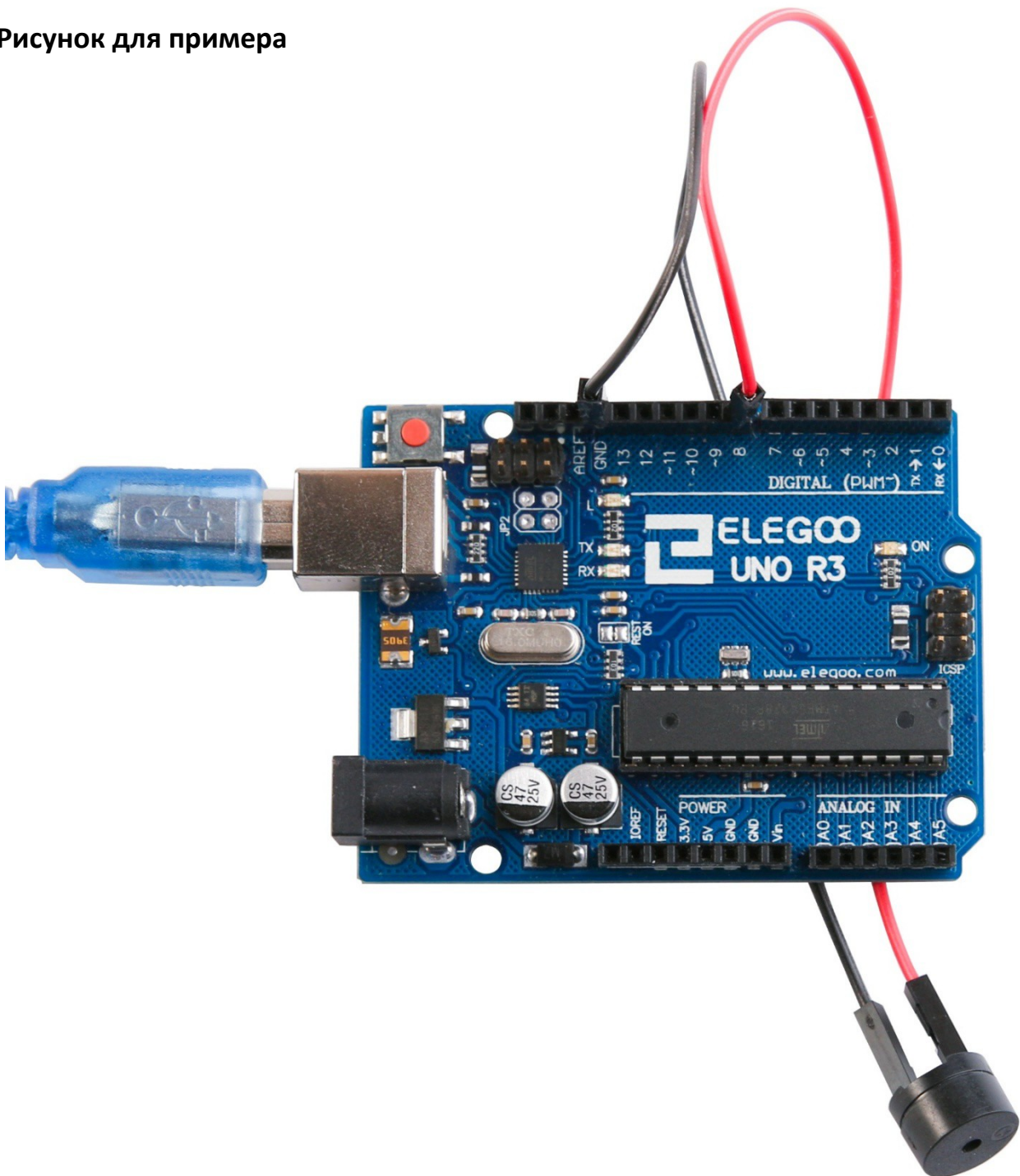
Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 7: Пьезоизлучатель без встроенного генератора (пассивный зуммер)”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Перед запуском, убедитесь, что вы установили библиотеку <pitch>. Переустановите, если необходимо, иначе, ваш код не будет работать.

Процедура добавления библиотек описана в Уроке 1.

Рисунок для примера



Урок 8: 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

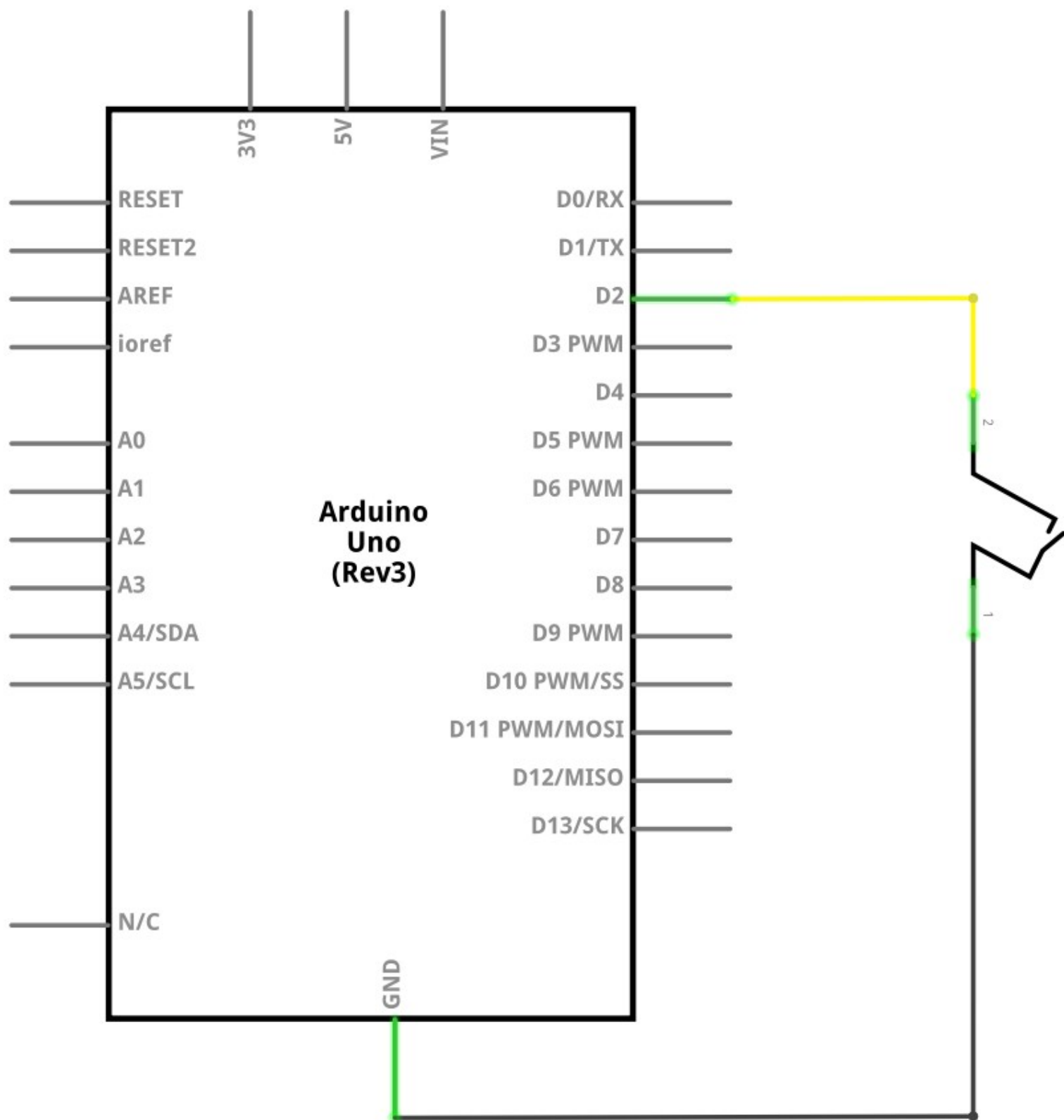
Датчик наклона:

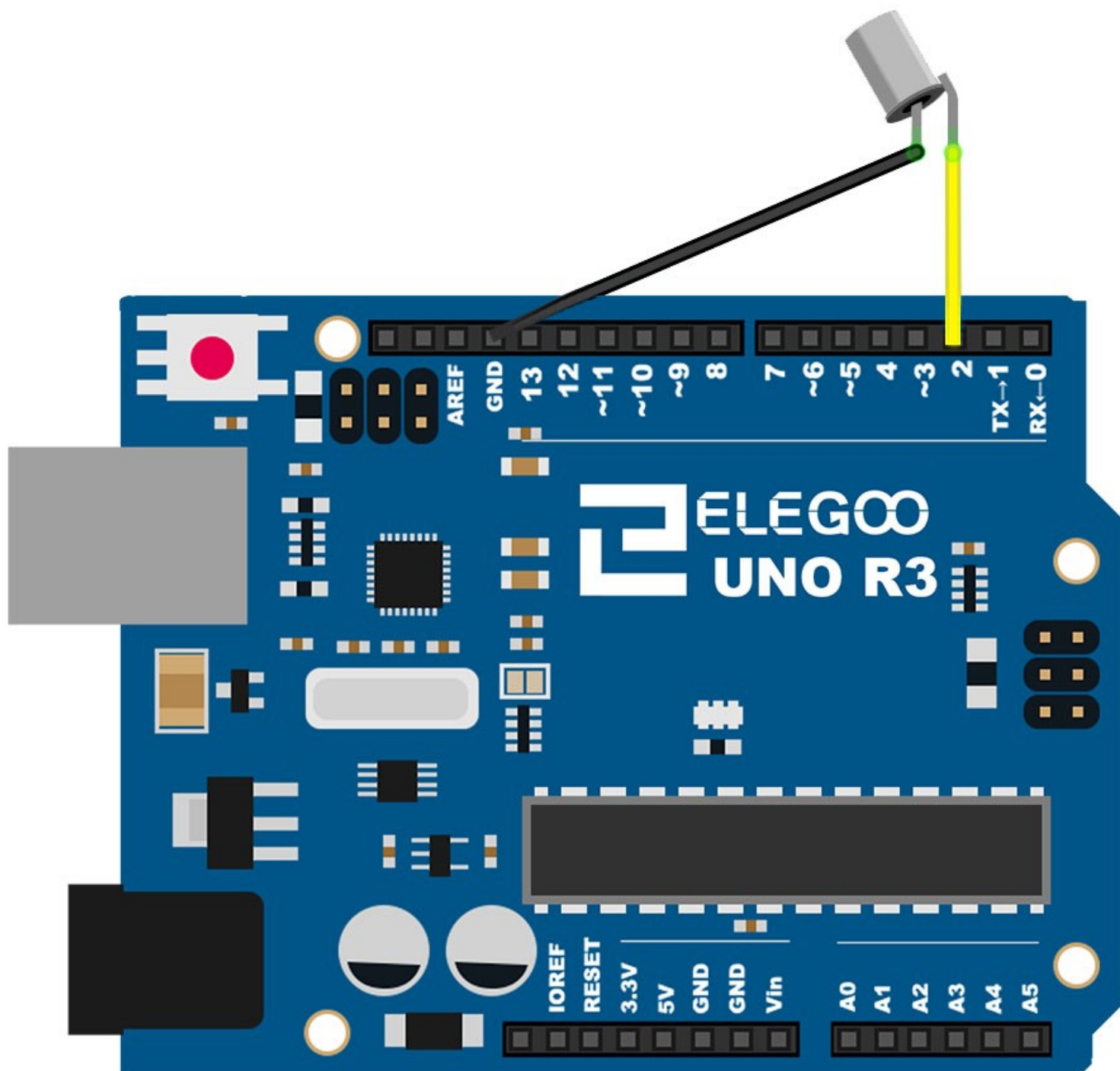
Датчики наклона позволяют определять ориентацию или наклон. Они компактные, недорогие, низковольтные и просты в использовании. При правильной эксплуатации, они не выйдут из строя. Простота и неказистость датчиков наклона сделала их популярным выбором для игрушек, гаджетов и разных приборов. В литературе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

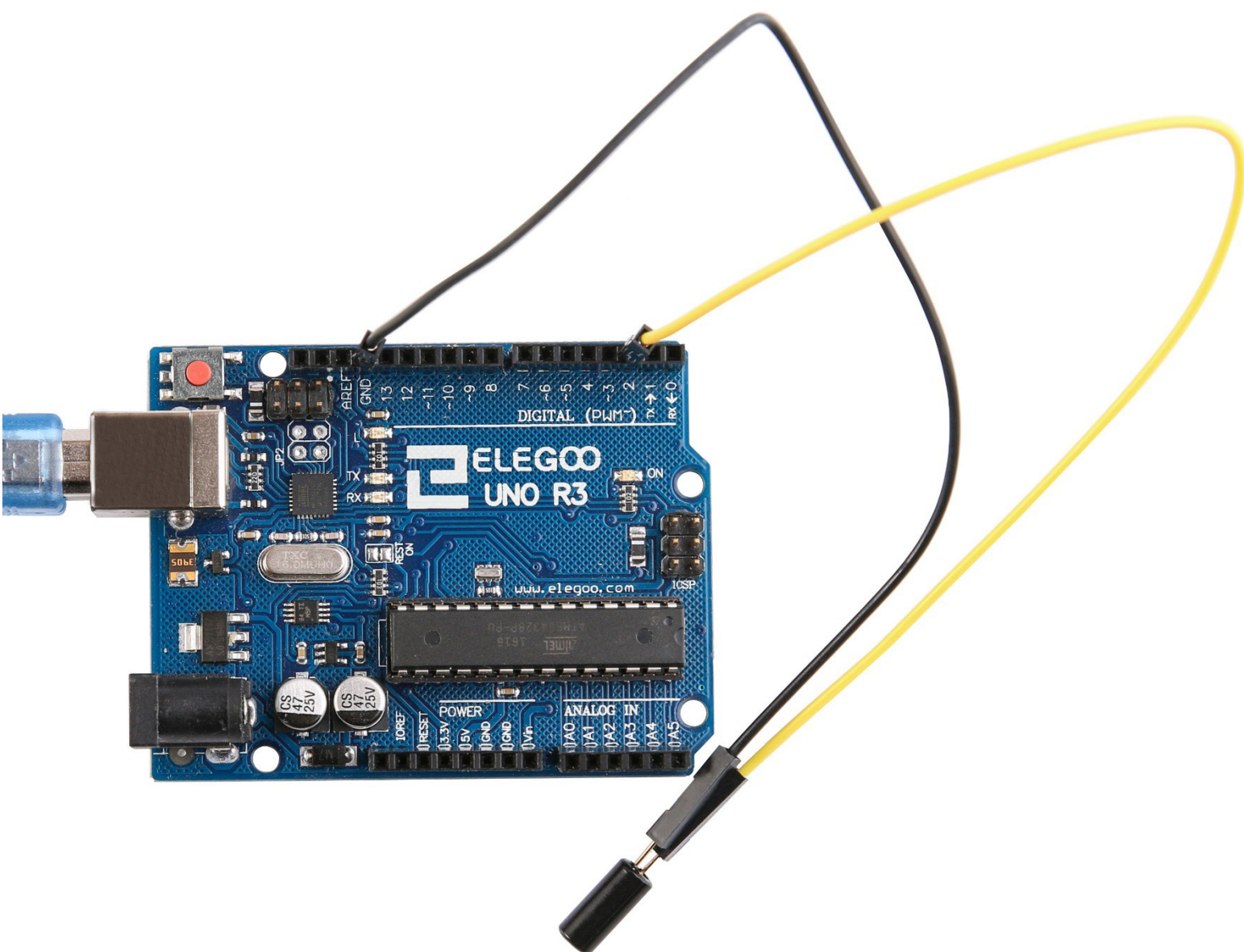




Код

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 9: Серводвигатель

Обзор

Сервопривод — это такой вид привода, который может повернуть свой вал только на определенный угол (180°C). Он контролируется подачей электрических сигналов с вашей платы UNO R3. Эти импульсы указывают сервоприводу, куда он должен двигаться. На сервомоторе 3 контакта. Красный — это всегда контакт 5 В. Контакт GND может быть черным или коричневым. Оставшийся третий контакт - это сигнал, который используется для управления положением ротора серводвигателя. Обычно он желтого или оранжевого цвета. Этот контакт мы подключаем к цифровому пину 9 на Arduino.

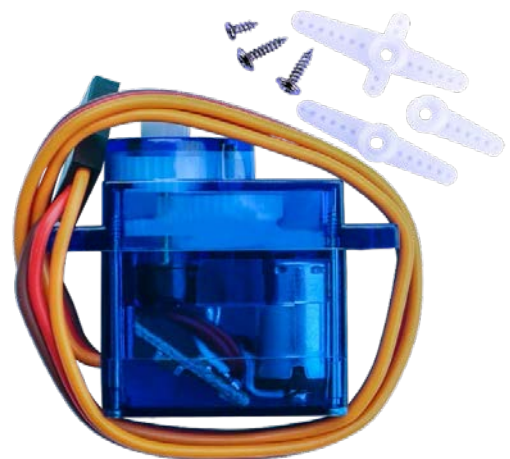
Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x серводвигатель (SG90)
- (3) x перемычки (типа “папа” - “папа”)

Представление компонентов

Технические характеристики серводвигателя SG90

- Универсальный для коннекторов JR и FP
- Длина кабеля: 25 см
- Скорость отработки команды: 0.12 с/ 60 град. (4.8 В), 0.10 с / 60 град. (6.0 В)
- Маленький крутящий момент (4.8 В): 1.6 кг/с
- Температура: -30~60°C
- Ширина полосы не пропускаения: 5мкс
- Рабочее напряжение: 3.5~6 В
- Размеры: 3,2 см x 3 см x 1,2 см
- Вес: 134 г



Подключение

Схема

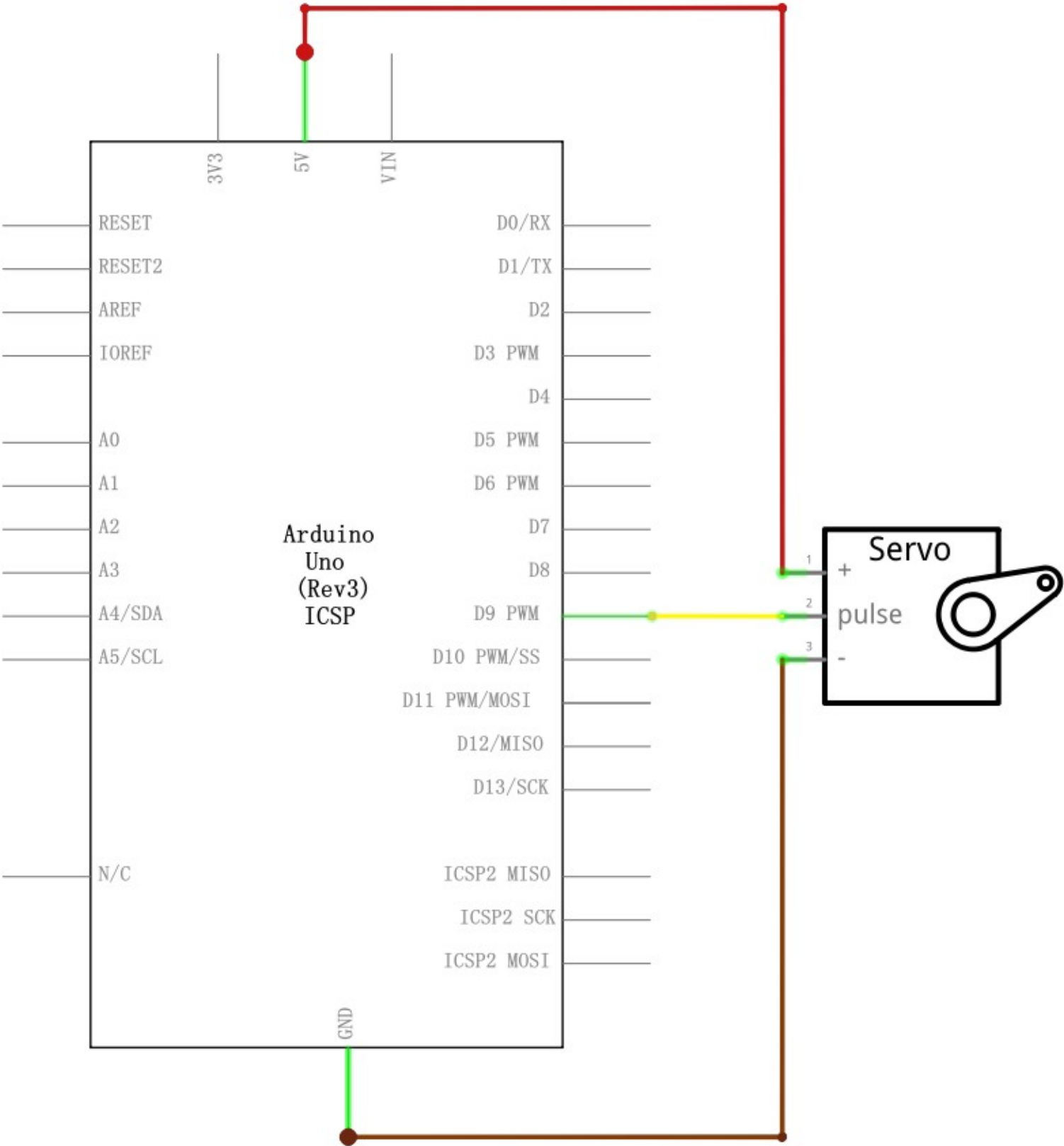
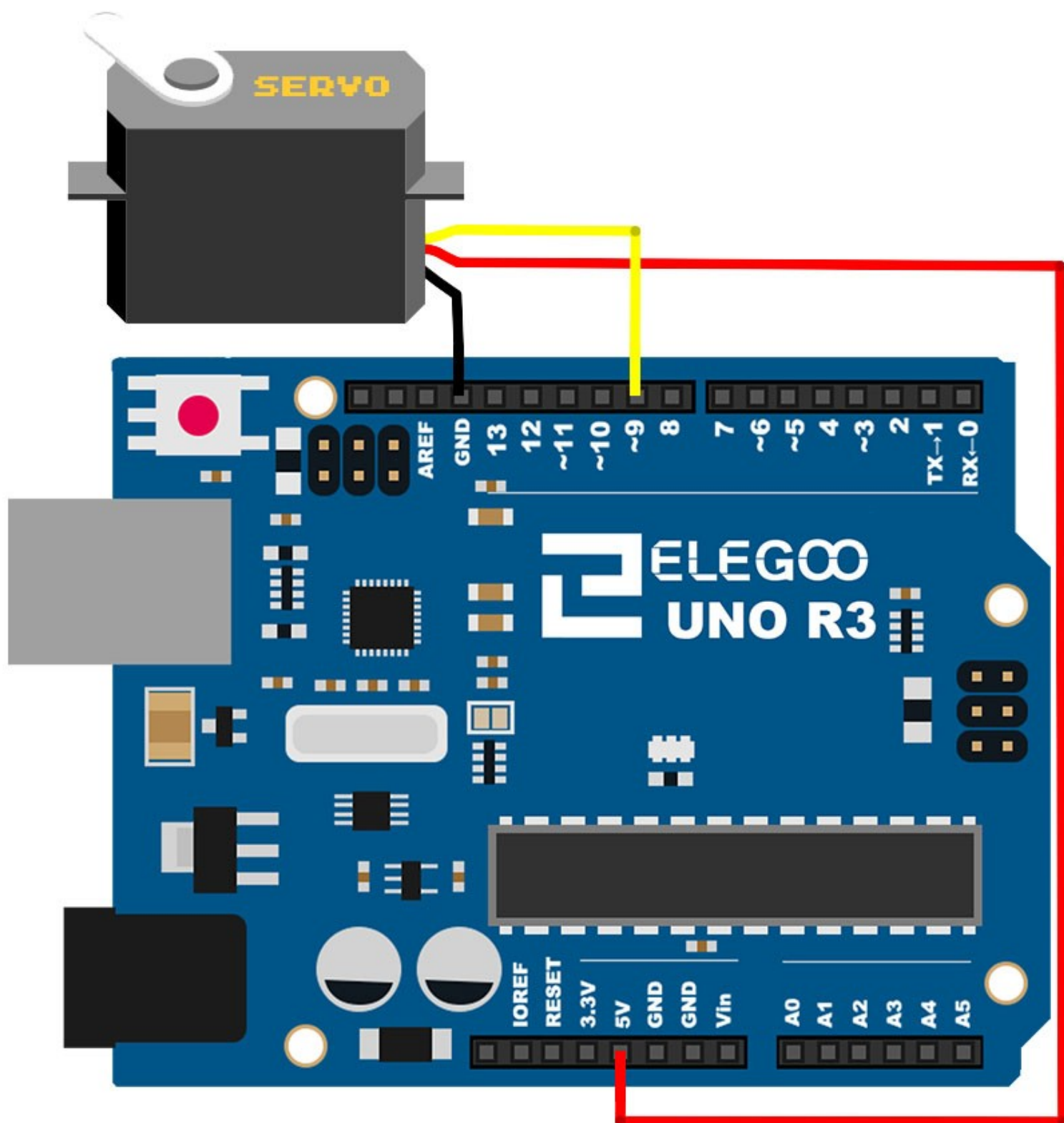


Схема монтажа



Код

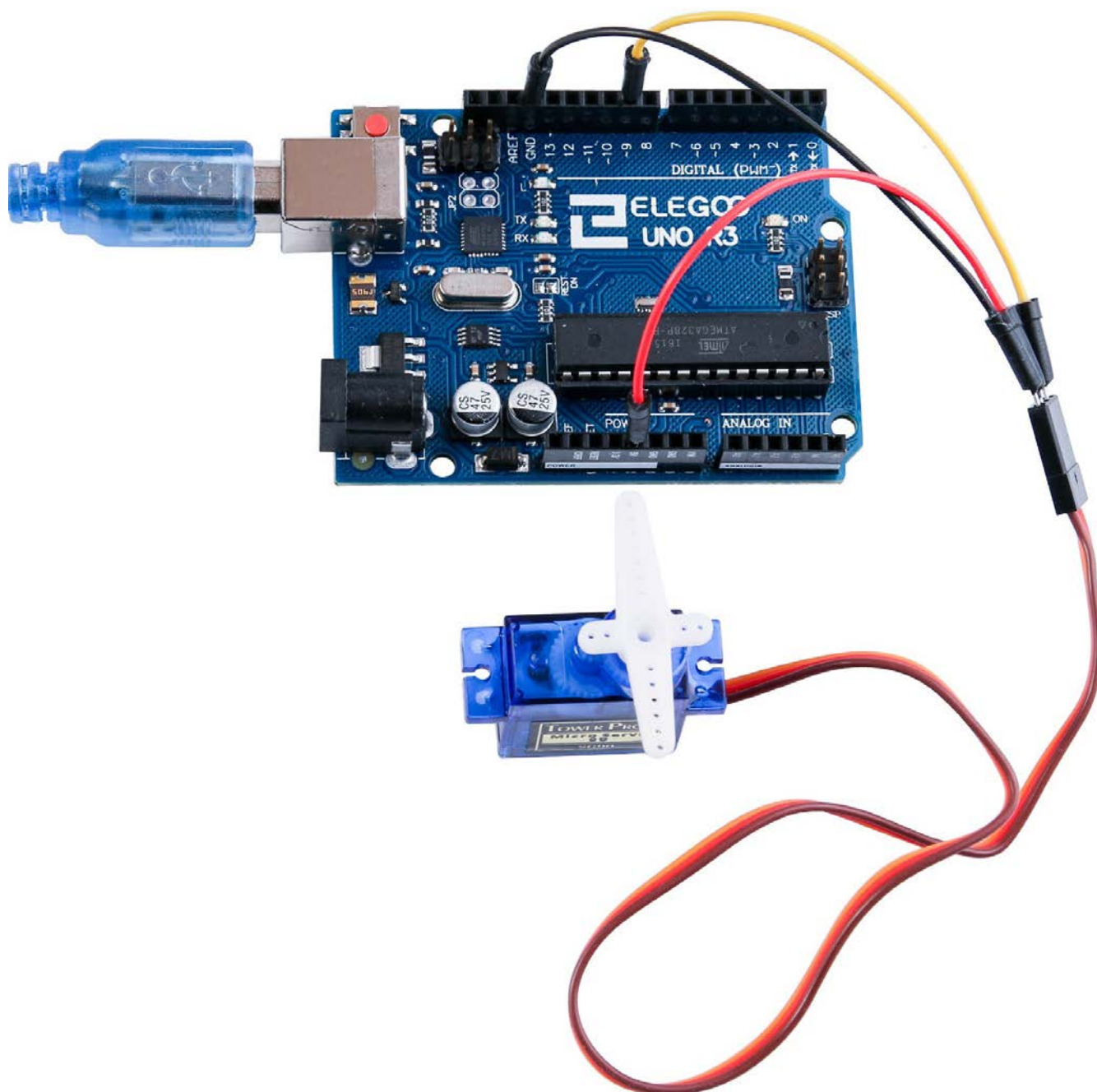
После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 9: Серводвигатель”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Перед запуском, убедитесь, что вы установили библиотеку <Servo>. Переустановите, если необходимо, иначе, ваш код не будет работать.

Процедура добавления библиотек описана в Уроке 1

Рисунок для примера

На этом рисунке коричневый провод серводвигателя подключен, используя черную перемычку (типа “папа” - “папа”), красный – красную перемычку (типа “папа” - “папа”), а оранжевый – желтую перемычку (типа “папа” - “папа”).



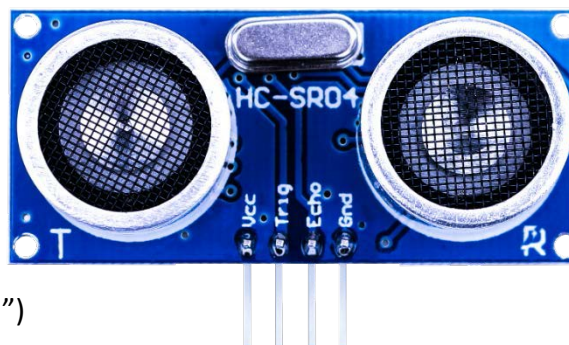
Урок 10: Ультразвуковой дальномер

Обзор

Ультразвуковой дальномер отлично подойдет для любых проектов, имеющих дело с измерением расстояния. Датчик HC-SR04 – недорогой и довольно простой в использовании. Последнее связано с тем, что мы будем использовать библиотеку, написанной специально для него.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x ультразвуковой дальномер
- (4) x перемычки DuPont (“мама” - “папа”)



Представление компонентов

Ультразвуковой дальномер

Ультразвуковой дальномер HC-SR04 является прибором бесконтактного типа, и обеспечивает высокоточное измерение и стабильность. Диапазон дальности его измерения составляет от 2 до 400 см. В комплект модуля с HC SR04 также входят ресивер и трансмиттер. Для получения данных, необходимо выполнить такую последовательность действий:

Подать на выход Trig импульс длительностью 10 мкс;

В ультразвуковом дальномере HC-SR04 произойдет преобразование сигнала в 8 импульсов с частотой 40 кГц, которые через излучатель будут посланы вперед;

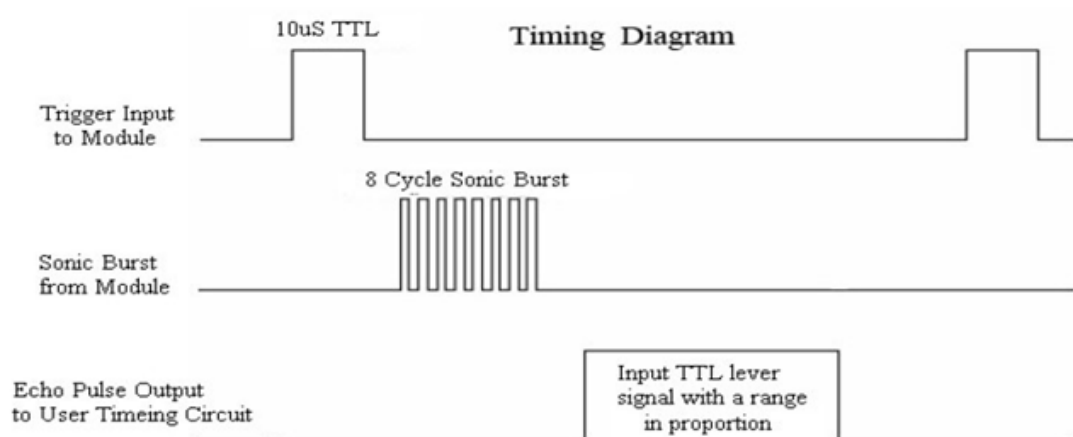
Когда импульсы дойдут до препятствия, они отразятся от него и будут приняты приемником R, что обеспечит наличие входного сигнала на выходе Echo;

На стороне контроллера полученный сигнал при помощи формул следует перевести в расстояние. Ниже представлена временная диаграмма.

Тестовое расстояние =

= (длительность полученного импульса × скорость звука (340 м/с) / 2

При делении ширины импульса на 58.2, получим данные в сантиметрах, при делении на 148 – в дюймах.



Подключение

Схема

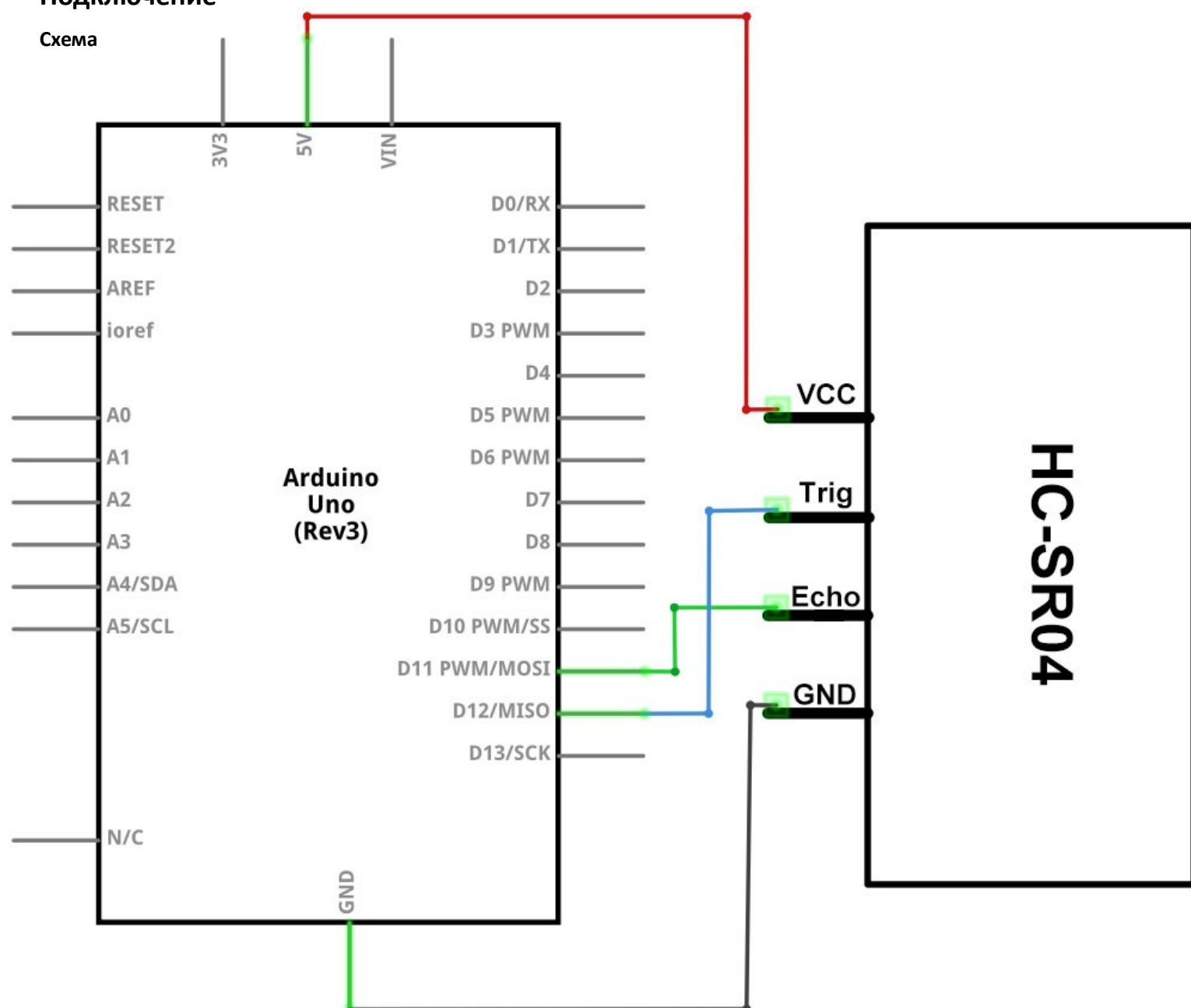
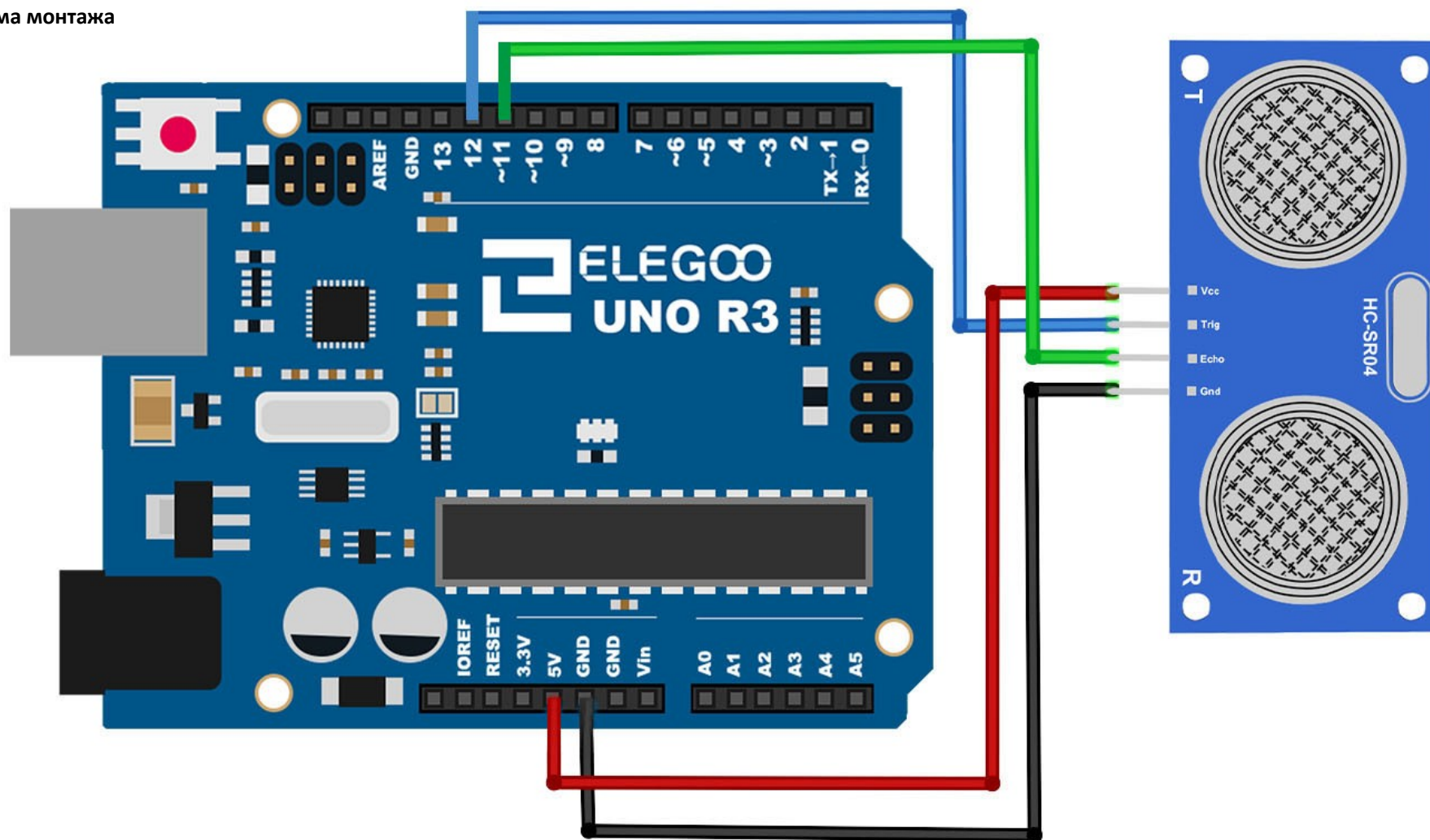


Схема монтажа



Код

За счет использования специально разработанной библиотеки, код будет коротким и простым. Мы включаем библиотеку в самом начале написания кода и далее, используя простые команды, мы контролируем поведение нашего сенсора.

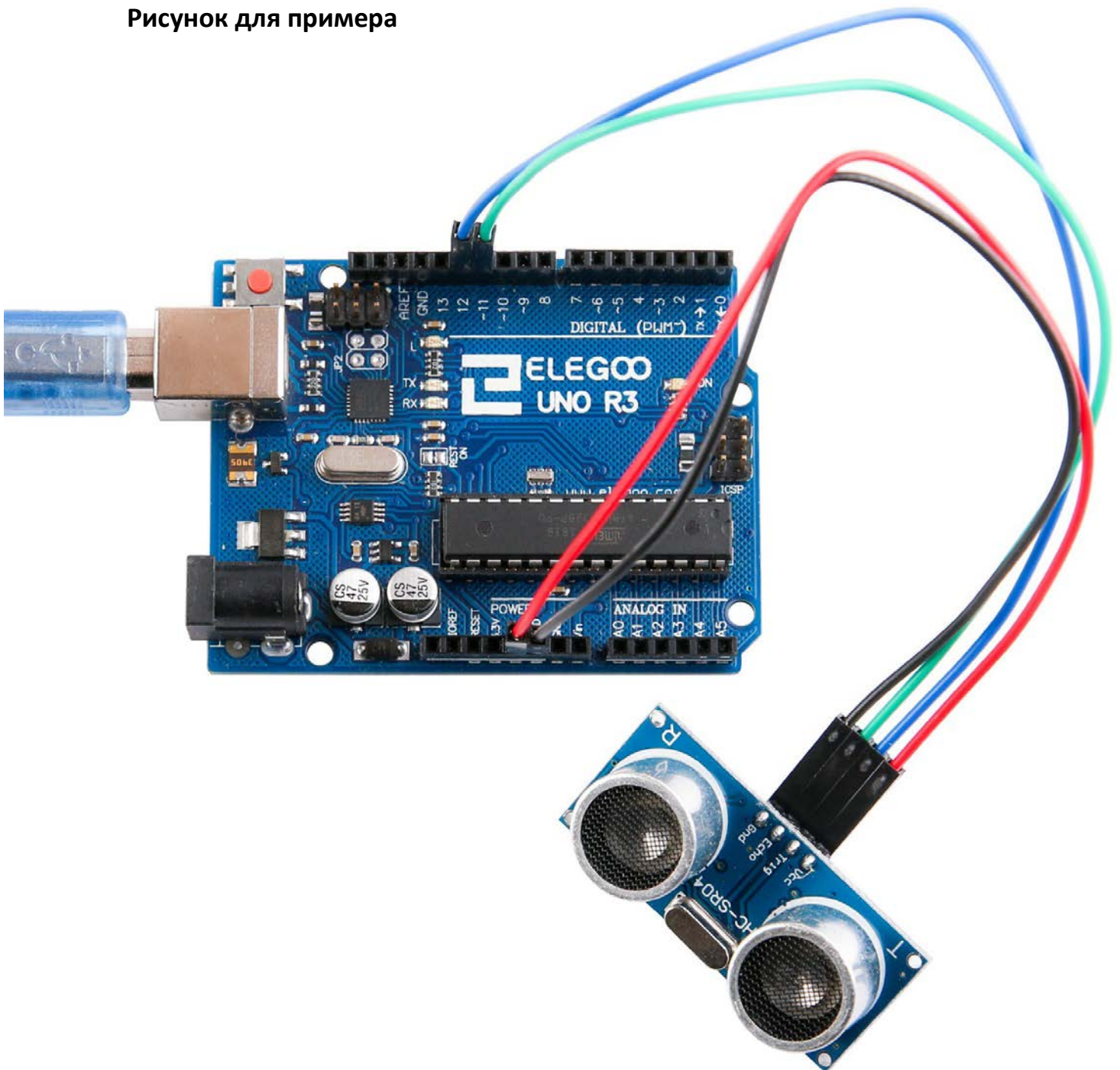
После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 10: Ультразвуковой дальномер”, и нажмите **UPLOAD** для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Перед запуском, убедитесь, что вы установили библиотеку <HC-SR04>.

Переустановите, если необходимо, иначе, ваш код не будет работать.

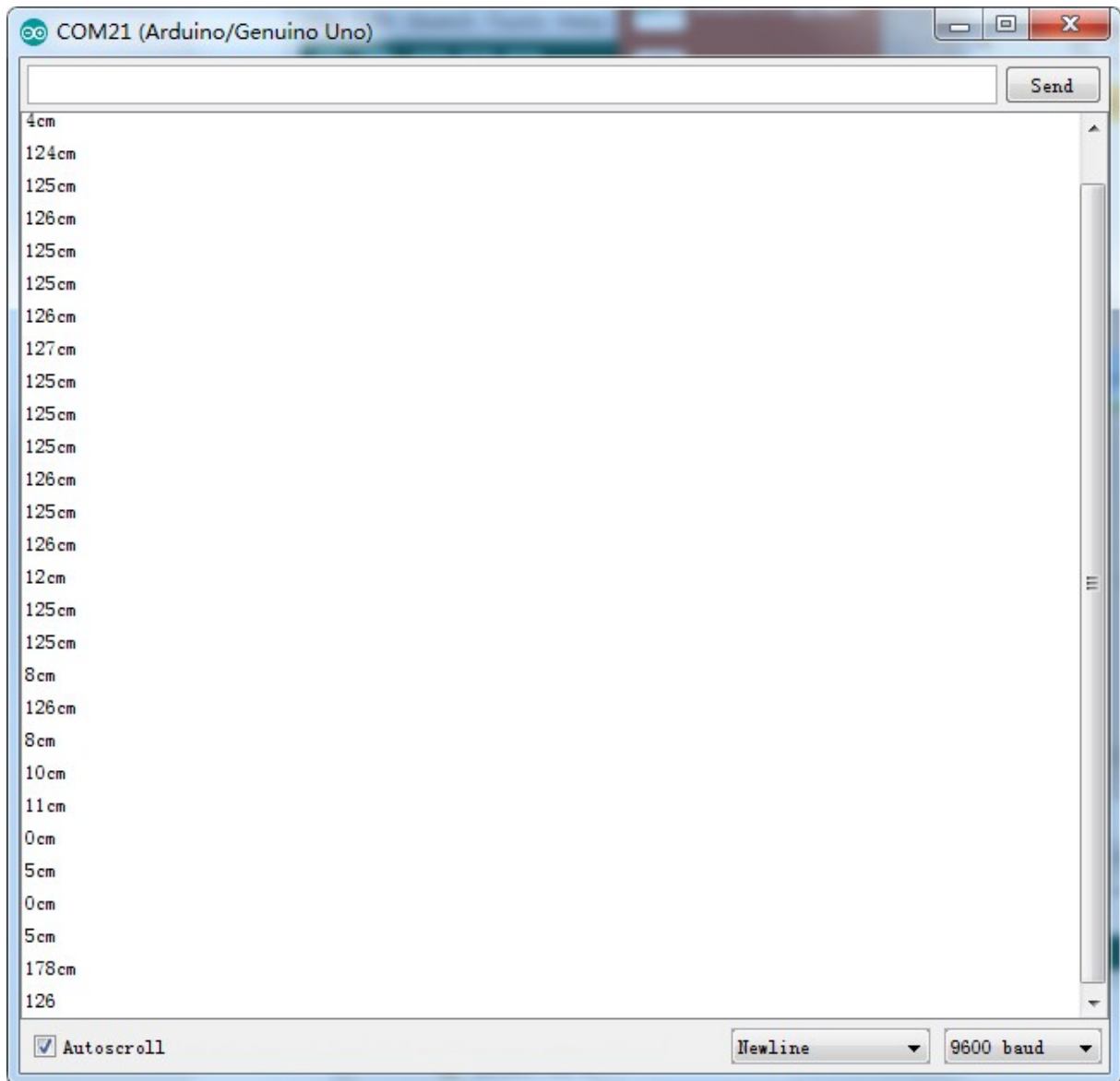
Процедура добавления библиотек описана в Уроке 1

Рисунок для примера



Откройте монитор, где будут отображены данные, как показано ниже:

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



Урок 11: Датчик температуры и влажности DHT11

Обзор

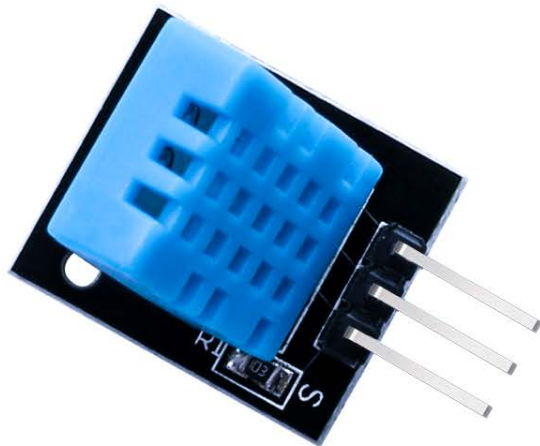
В этом уроке вы научитесь использовать датчик температуры и влажности DHT11. Он достаточно точный для использования в проектах, где необходимо производить измерения температуры и влажности. Опять же, мы будем использовать библиотеку, созданную специально для этого датчика, что значительно сократит и упростит написание кода.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x датчик температуры и влажности DHT11
- (3) x перемычки DuPont (типа “мама” – “папа”)

Представление компонентов

Датчик температуры и влажности:



Датчик температуры и влажности DHT11 состоит из двух основных частей: ёмкостного датчика влажности и термистора. Также в корпусе установлен простой чип для преобразования аналогового сигнала в цифровой. Считывать цифровой сигнал на выходе достаточно просто, можно использовать любой контроллер, не

обязательно Arduino. Продукт характеризуется высокой надежностью и длительностью использования.

Применения: отопление, вентиляция, кондиционирование, осушители воздуха, оборудование для контроля и инспекции, товары массового потребления, сфера автомобилестроения и автоматизации, регистрирующие устройства, метеорологические станции, бытовая техника, регулятор влажности, медицинское и другое оборудование для контроля и регулировки влажности.

Характеристики продукта

Измерение относительной влажности:

Разрешающая способность: 16 бит

Воспроизводимость: $\pm 1\%$

Точность: При температуре 25°C - $\pm 5\%$

Взаимозаменяемость: полная

Время реакции: $1/\tau$ (63%) при 25°C 6 сек

1 м/сек воздух 6сек

Гистерезис: $<\pm 0,3\%$

Стабильность: $<\pm 0,5\%$ /год

Измерение температуры:

Разрешающая способность: 16 бит

Воспроизводимость: $\pm 0,2^{\circ}\text{C}$

Диапазон: При 25°C $\pm 2^{\circ}\text{C}$

Время реакции: $1/\tau$ (63%) 10 сек

Электрические характеристики

Питание: постоянных 3,5~5,5 В

Ток питания: в режиме измерения – 0,3mA; в режиме ожидания – 60μA

Время выборки: более 2 сек

Описание контактов:

VDD – питание (3,5~5,5 В);

DATA – вывод данных;

NC – не используется;

GND – земля.

Подключение

Схема

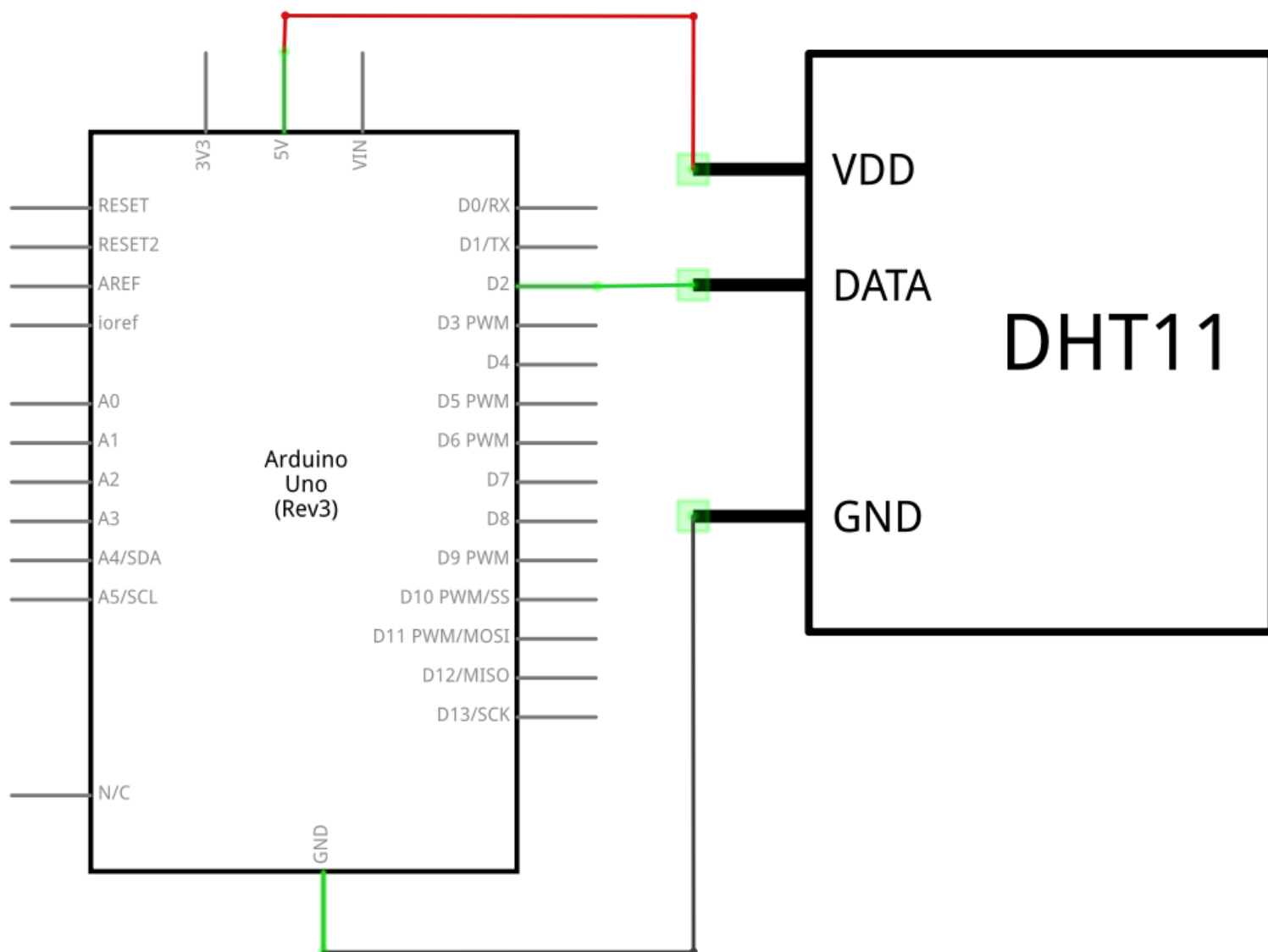
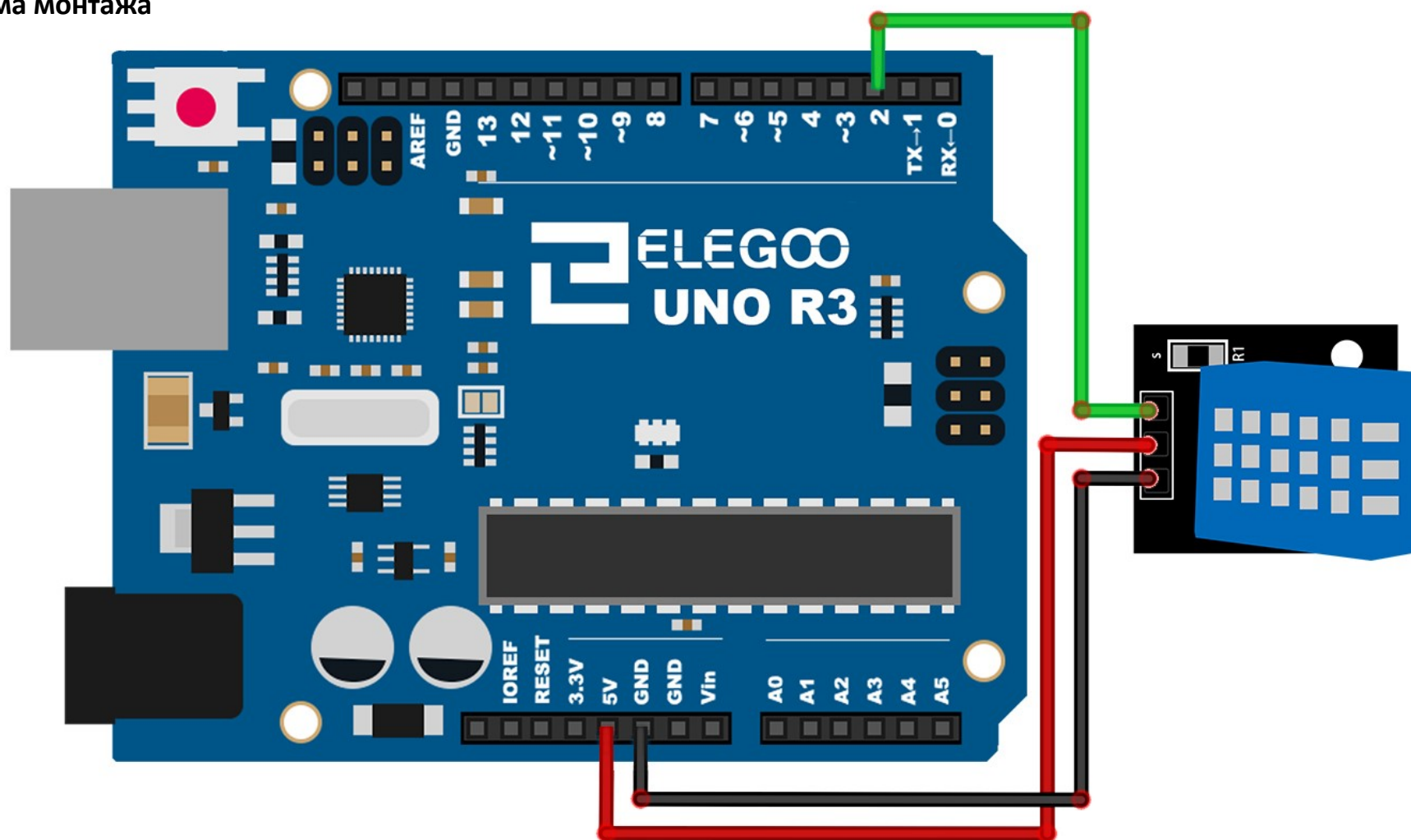


Схема монтажа



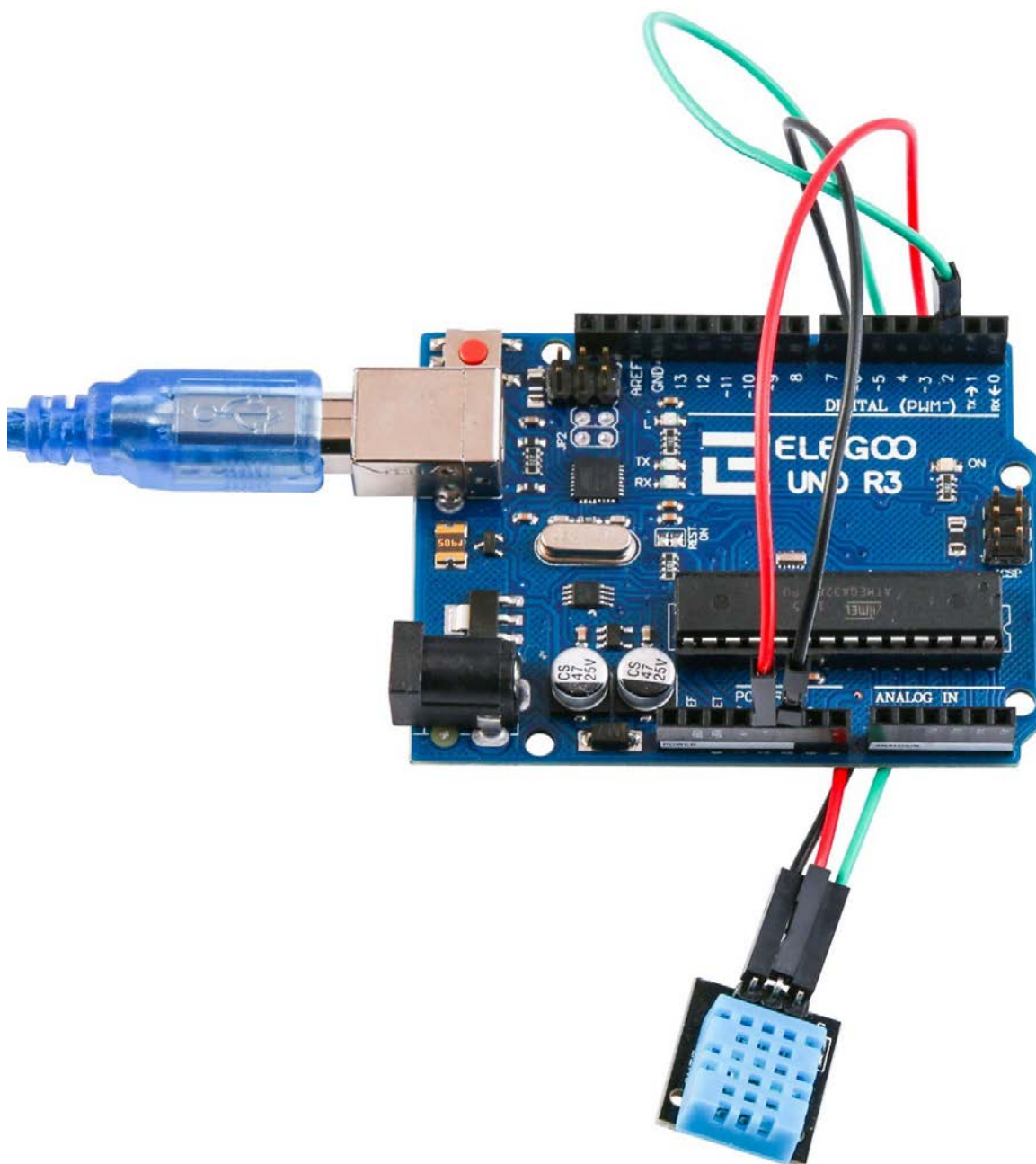
Как видите, необходимо подключить только 3 контакта, поскольку один из них не используется. Контакты VDD, DATA и GND датчика могут подключаться к любым пинам платы UNO.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 12: Датчик температуры и влажности DHT11”, и нажмите **UPLOAD** для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

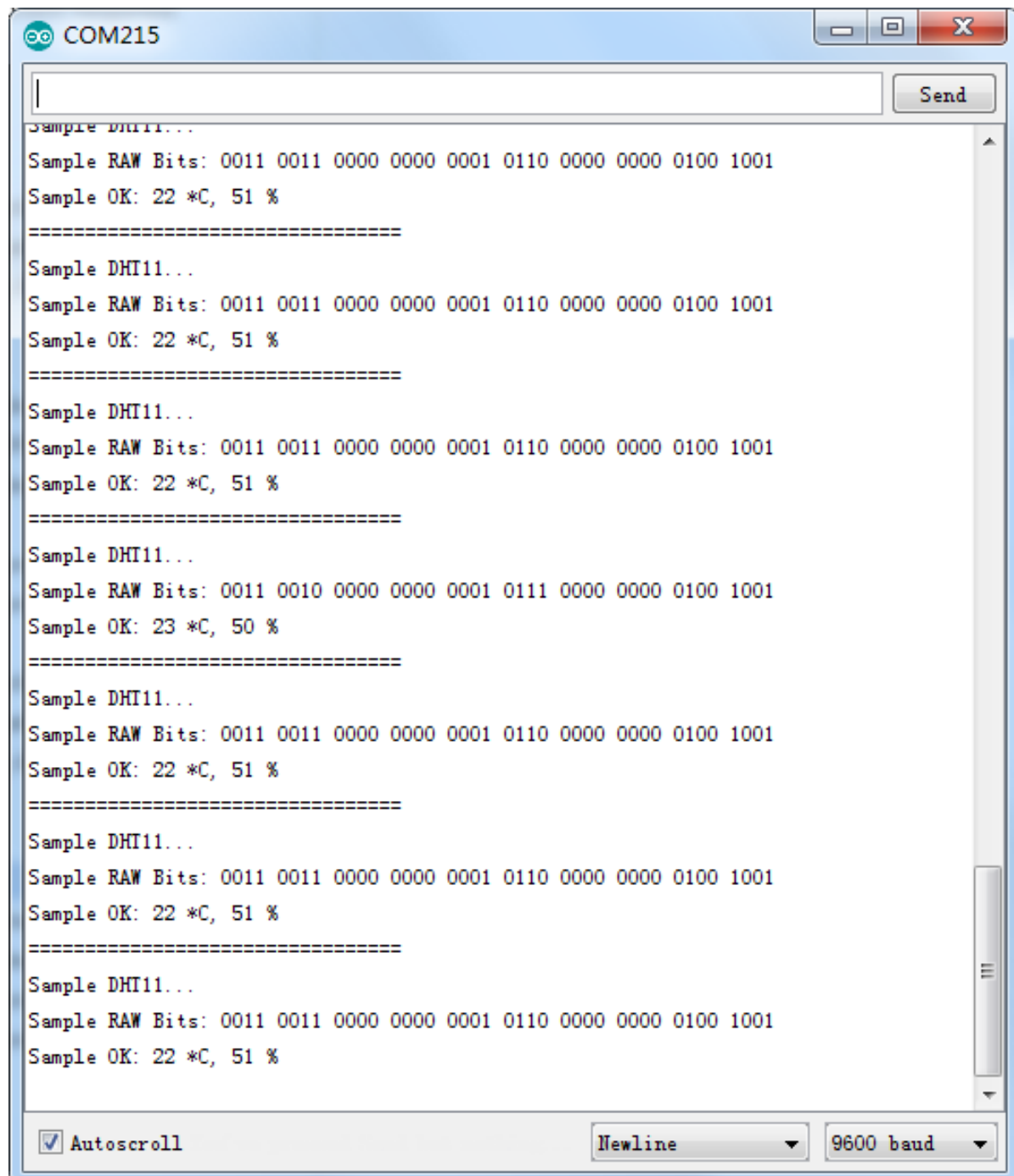
Перед запуском, убедитесь, что вы установили библиотеку < SimpleDHT >. Переустановите, если необходимо, иначе, ваш код не будет работать. Процедура добавления библиотек описана в Уроке 1.

Рисунок для примера



При открытии монитора вы должны увидеть данные, как показано ниже (должна отображаться температура окружающей среды – 22°C):

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



Урок 12: Аналоговый джойстик

Обзор

Аналоговый джойстик – прекрасный способ добавить управление в ваш проект. В этом уроке мы научимся использовать аналоговые джойстики в подключении к нашей плате.

Необходимые компоненты:

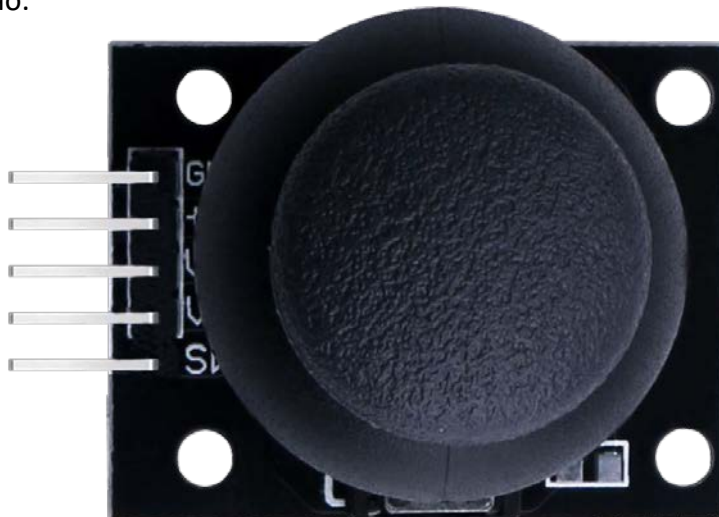
- (1) х плата Elegoo Uno R3
- (1) х джойстик
- (5) х перемычки DuPont (типа “мама” – “папа”)

Представление компонентов

Джойстик

На модуле 5 пинов: Vcc, Ground, X, Y, Key. Обратите внимание, что обозначения на вашем модуле могут отличаться. Это зависит от производителя. Джойстик аналоговый и обеспечивает более высокую точность, чем простые 'directional' джойстики, в которых используются кнопки и механические переключатели. Кроме того, на джойстик можно нажать (на моей модели прилагать для этого приличные усилия. Возможно, он просто еще не разработан). После нажатия отработает кнопка 'press to select'.

Для считывания данных с пинов X/Y надо использовать аналоговые выходы на Arduino. Пин Key замыкается землей при нажатии. В противоположном случае ни в какой цепи он не участвует. Для стабильного считывания данных с пинов Key/Select, они должны подключаться к источнику питания (Vcc) через подтягивающий резистор. Номинала встроенных на Arduino резисторов для этого будет вполне достаточно.



Подключение

Схема

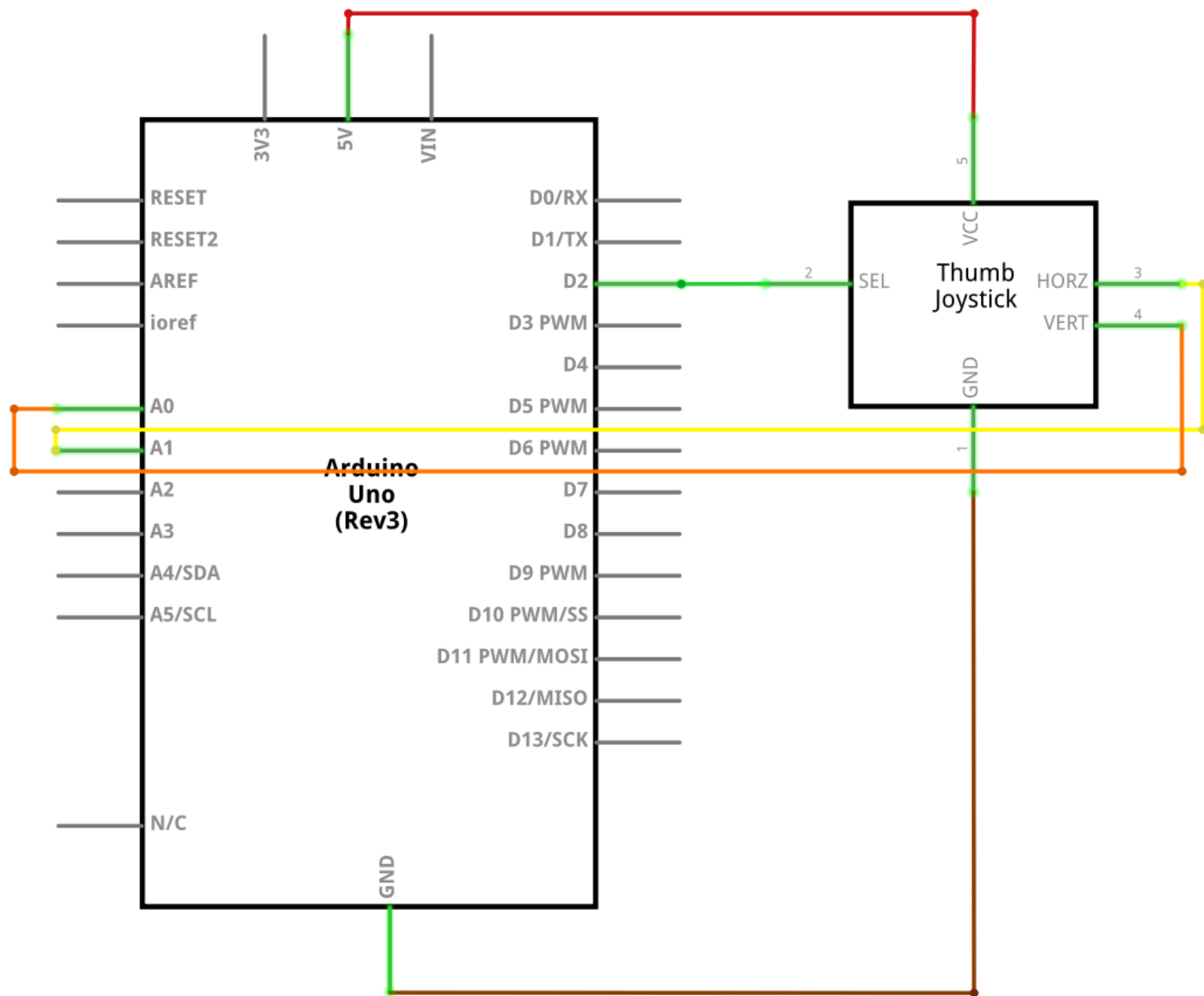
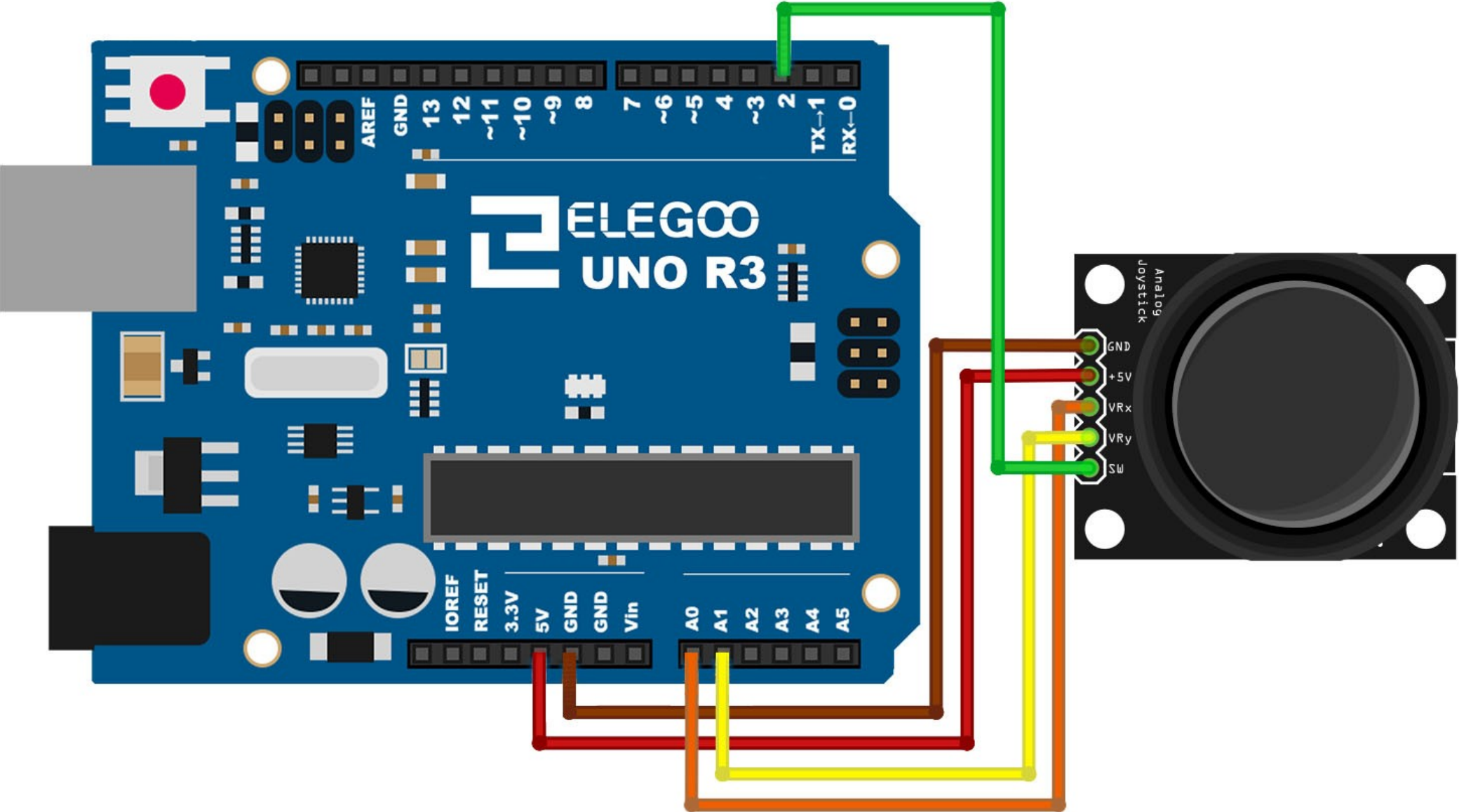


Схема монтажа



Нам понадобится совершить пять подключений: Key, Y, X, Vcc и Ground. Выводы “Y” и “X” – аналоговые, а “Key” - цифровой. Если вам не нужен переключатель, то можете использовать только 4 пина.

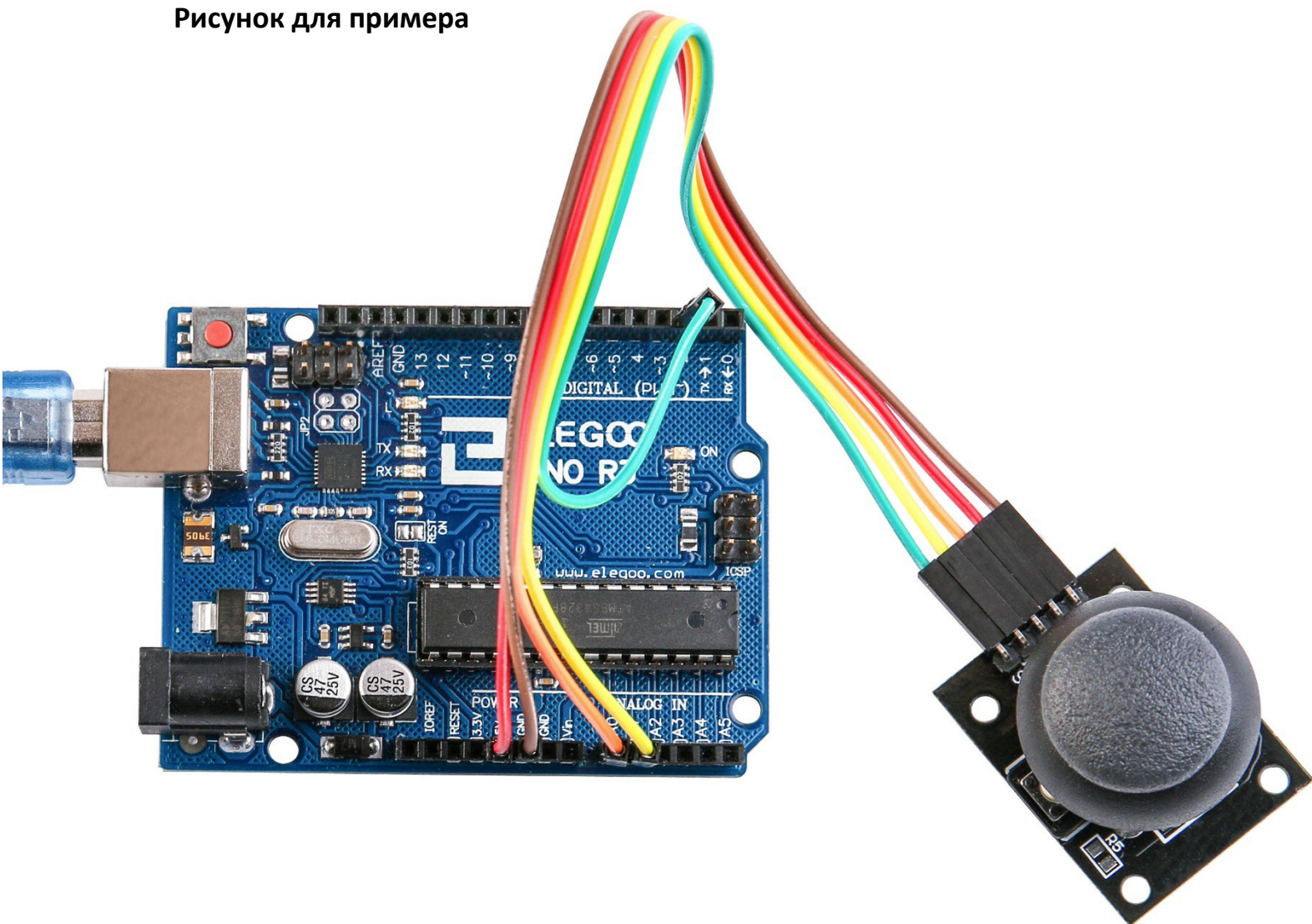
Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 13: Аналоговый джойстик”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Аналоговый джойстик – это, фактически, переменный резистор, поэтому он возвращает аналоговые значения.

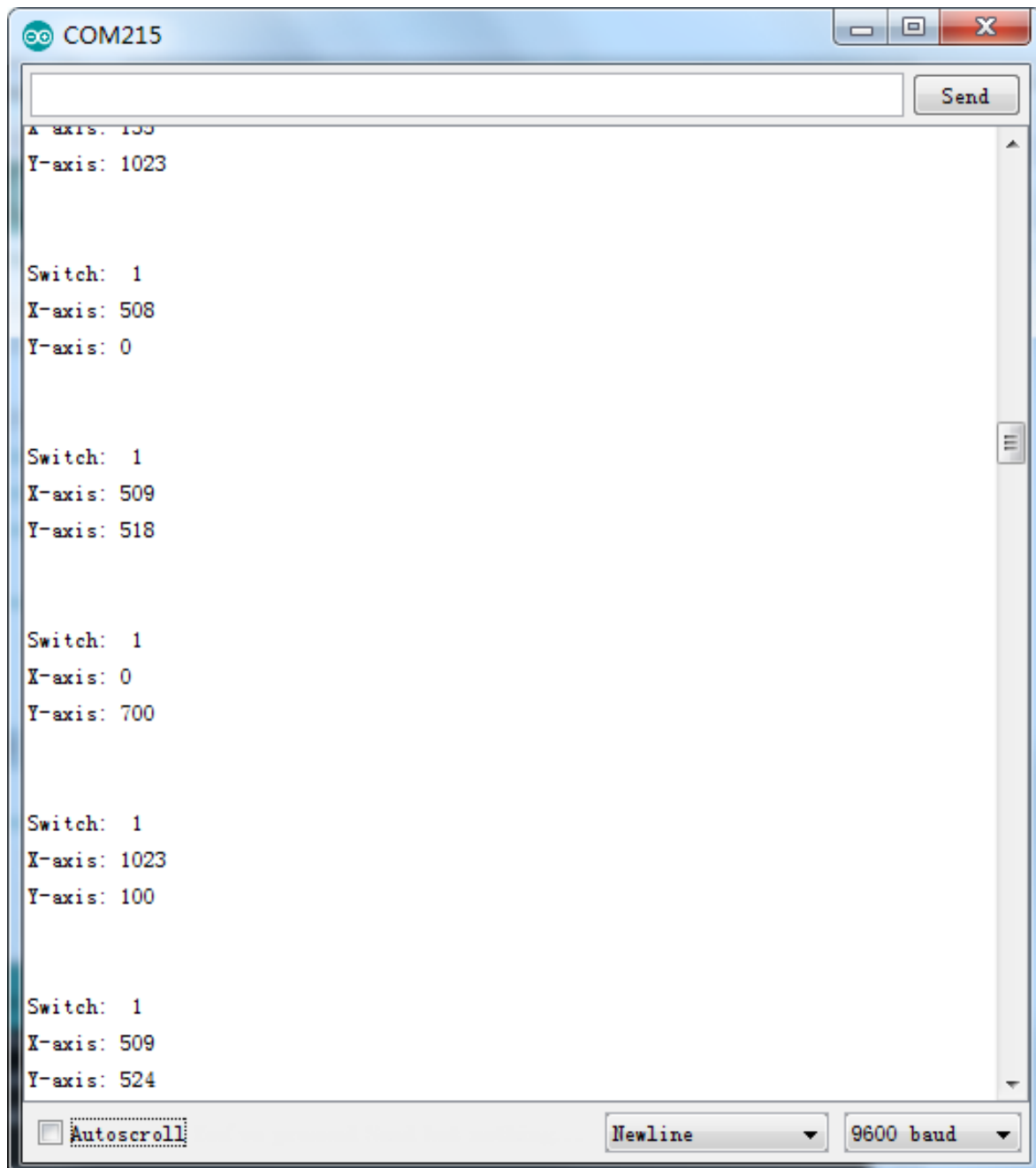
Значения в каждом из направлений будут в диапазоне от 0 до 1023, можно предположить, что центр будет находиться в диапазоне 511-512.

Рисунок для примера



При открытии монитора вы должны увидеть данные, как показано ниже:

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



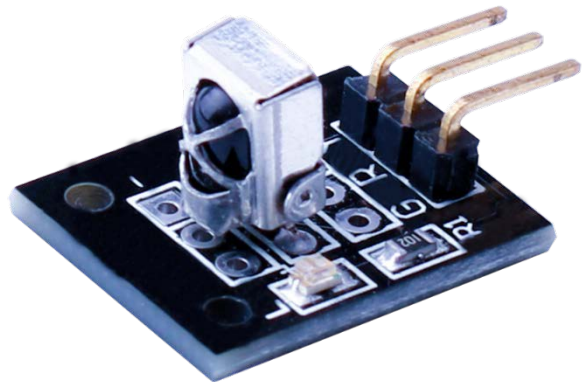
Урок 13: ИК приемник (пульт)

Обзор

Использование Инфракрасного (ИК) приемника с пультом дистанционного управления – это отличный способ добавить беспроводное и дистанционное управление электронными приборами в ваш проект. ИК пульт довольно простой в использовании. В этом уроке вы научитесь подключать ИК приемник к плате UNO и использовать библиотеку, созданную специально для этого датчика. В нашем скетче будут задействованы все шестнадцатеричные коды, соответствующие каждой кнопке на ИК пульте. Мы также будем детектировать распознавание кодов и нажатие на кнопку.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x модуль ИК приемника
- (1) x ИК пульт дистанционного управления
- (3) x перемычки DuPont (типа “мама” – “папа”)



Представление компонентов

ИК ПРИЕМНИК:

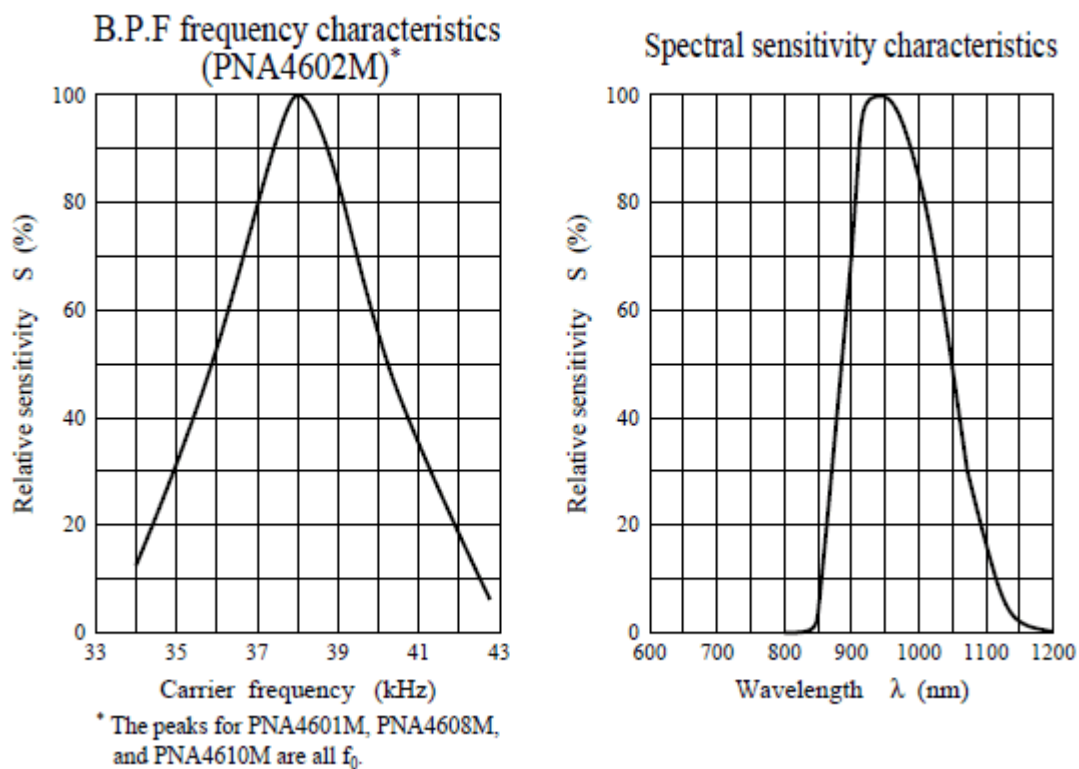
ИК приемник представляет собой микрочип со встроенным фоторезистором, чувствительным к инфракрасному излучению. Они, практически всегда, используются для детектирования дистанционного управления – каждый телевизор и DVD-проигрыватель оборудованы таким приемником, чтобы “слушать” ИК сигналы от пульта. Внутри пульта расположен ИК светодиод, испускающий ИК импульсы, “приказывающие” телевизору включиться/выключиться или сменить канал. ИК излучение – невидимое человеческому глазу, поэтому проверка установки занимает немного больше времени.

Существует некоторая разница между таким приемником и, например, фоторезистором на основе сульфида кадмия (CdS). ИК приемники специально настроены на прием ИК излучения, потому не подходят для работы в видимом диапазоне. Фоторезисторы, в свою очередь, хороши для работы с видимым излучением (зеленым/желтым), но не подходят для ИК диапазона. ИК приемники

оборудованы демодулятором (детектором), настроенным на прием ИК сигнала частотой 38 кГц. Простого свечения ИК светодиода будет недостаточно, сигнал должен быть ШИМ-мерцанием частотой 38 кГц. В фоторезисторах нет никакого демодулятора, потому они могут детектировать любую частоту в пределах скорости его реакции (приблизительно 1 кГц).

ИК приемники – это устройства цифрового вывода. При детектировании ИК сигнала частотой 38 кГц, на выходе приемника будет LOW (0 В), при отсутствии ИК сигнала – HIGH (5 В). Сопротивление фоторезистора меняется в зависимости от степени освещенности.

Что можно измерить



Из графиков видно, что пик детектирования соответствует частоте 38 кГц и излучению светодиода на 940 нм. Вы можете использовать диапазон частот 35 – 41 кГц, но это приведет к понижению чувствительности настолько, что на больших расстояниях прибор не будет детектировать сигнал. Точно так же, вы можете использовать светодиоды, излучающие в диапазоне 850 – 1100 нм, но они не будут детектироваться так же хорошо, как диапазон 900 – 1000 нм. Убедитесь, что вы используете соответствующие светодиоды!

Проверьте длины волн излучения ИК светодиода в его паспорте. Попробуйте выбрать светодиод, излучающий на длине волны 940 нм – помните, что это не видимый свет!

Соединение

Схема

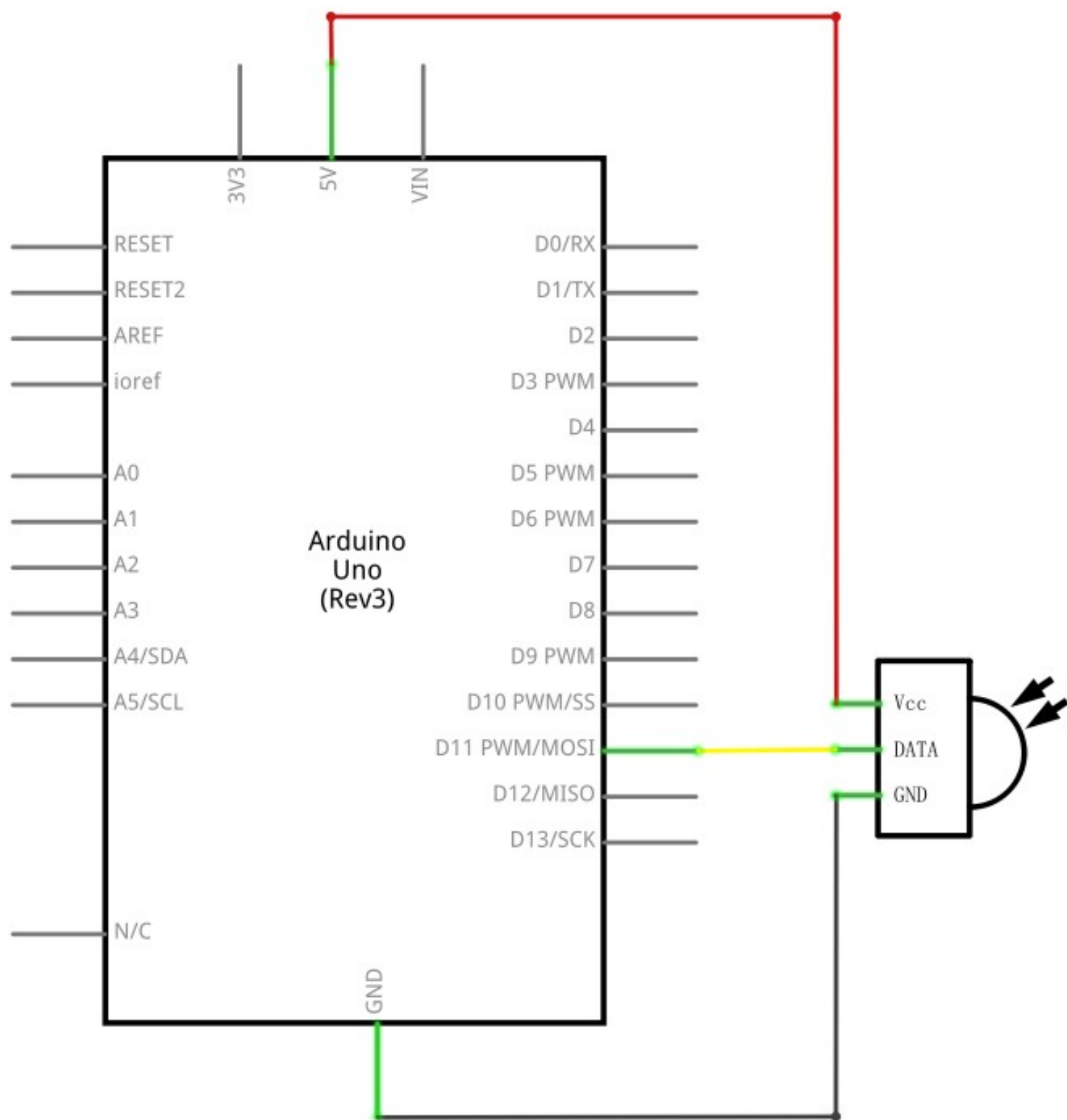
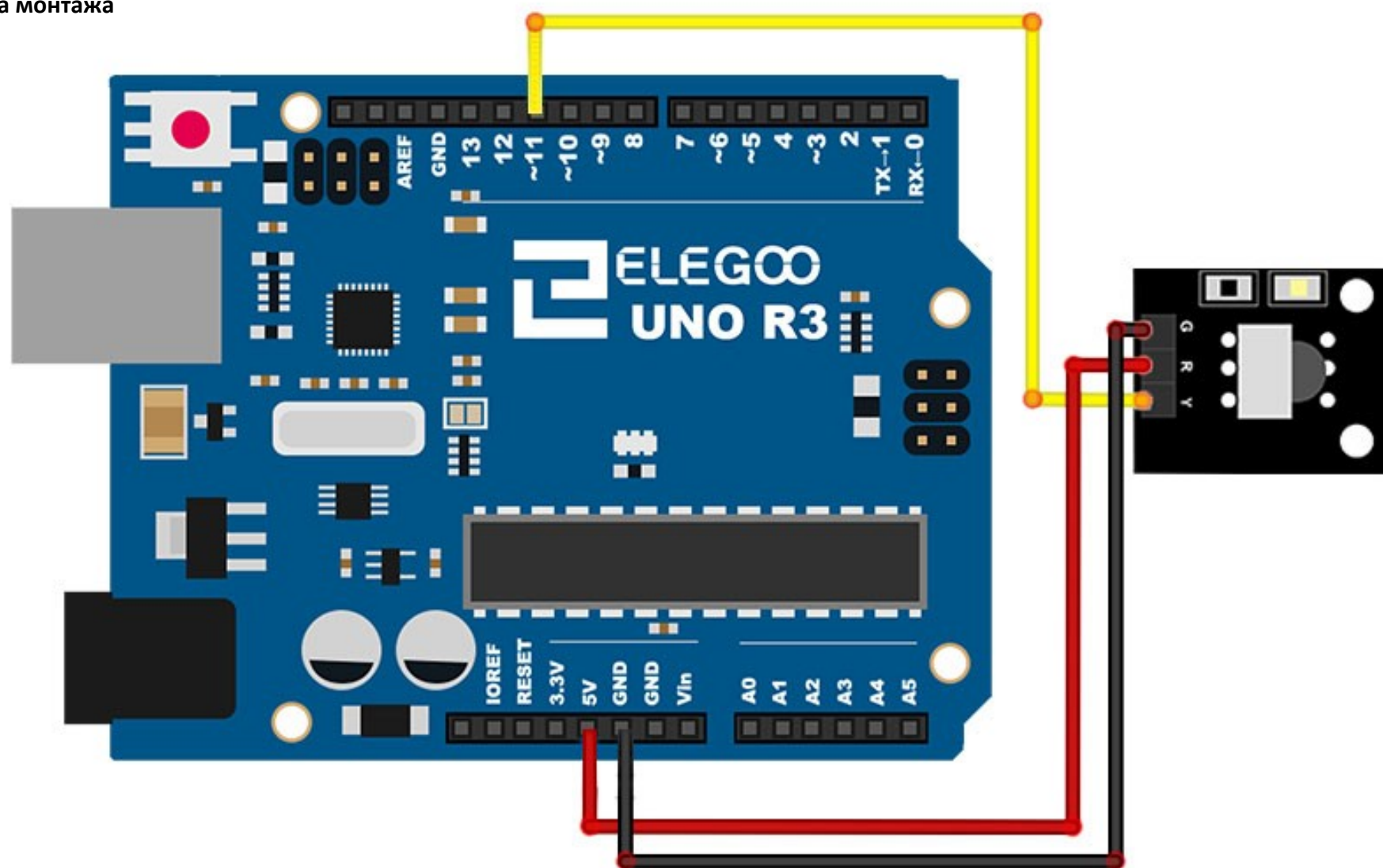


Схема монтажа



ИК приемник имеет три контакта: цифровой вывод (Signal), питание (Voltage) и земля (Ground). Обозначение “-” соответствует земле/общему проводу, “S” – сигналу/цифровому выводу, а средний контакт – питание (5 В).

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 14: ИК приемник (пульт)”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

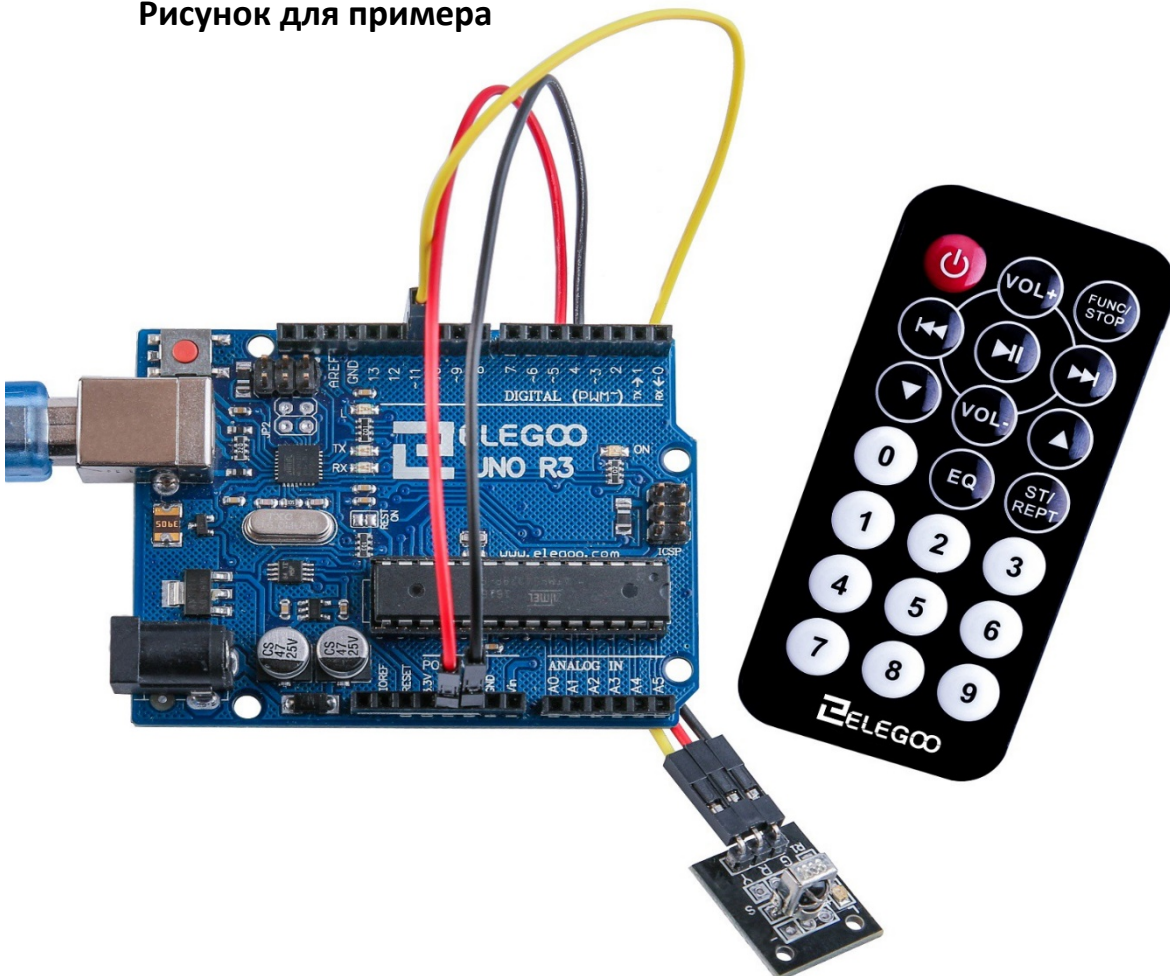
Перед запуском, убедитесь, что вы установили библиотеку < IRremote >. Переустановите, если необходимо, иначе, ваш код не будет работать.

Процедура добавления библиотек описана в Уроке 1.

Далее, мы должны переместить <RobotIRremote> из папки Library, потому что эта библиотека будет помехой для использования выбранной нами библиотеки. Вы можете вернуть ее обратно по окончании программирования микроконтроллера.

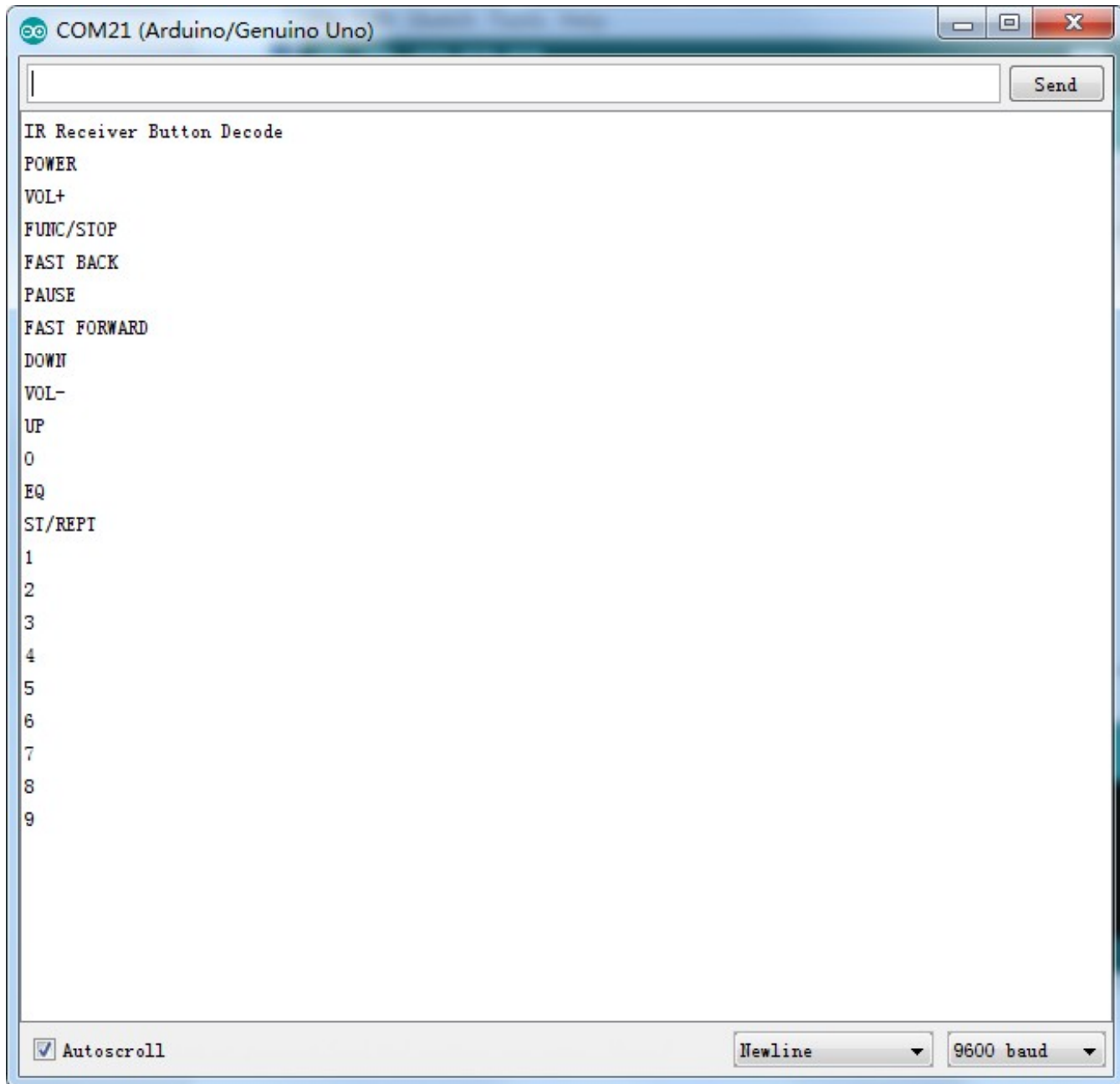
Перезапустите среду разработки после установки библиотеки.

Рисунок для примера



При открытии монитора вы должны увидеть данные, как показано ниже:

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



Урок 14: Жидкокристаллический индикатор (ЖКИ/LCD)

Обзор

В этом уроке вы научитесь подключать и использовать алфавитно-цифровой ЖКИ. Индикатор имеет светодиодную подсветку и может отображать строки длиной до 16 символов. Вы можете увидеть прямоугольники, предназначенные для каждого символа на индикаторе, а также пиксели, образующие символ. Символы высвечиваются белым цветом на синем фоне. Индикатор предназначен только для отображения текста. В этом уроке мы только запустим пример программы для Arduino, использующую библиотеку “LCD”, но в следующем уроке мы заставим наш индикатор отображать температуру, полученную датчиками.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x индикатор LCD1602
- (1) x потенциометр (10 кОм)
- (1) x макетная плата на 830 точек
- (16) x перемычки (типа “папа” – “папа”)



Представление компонентов

Индикатор LCD1602

Выводы индикатора LCD1602:

VSS: земля (общий провод);

VDD: питание +5 В;

VO: управление контрастностью индикатора LCD1602;

RS: выбор регистра для записи данных в памяти ЖКИ. Вы можете выбрать либо регистр данных, который удерживает данные, передаваемые на экран, либо регистр команд, который содержит инструкции для контроллера ЖКИ;

R/W: выбор режима чтения/записи;

E: при подаче низковольтного сигнала, дает команду ЖКИ для выполнения соответствующей инструкции;

D0-D7: отвечают за входящие/исходящие данные;

A и K: контролируют светодиодную подсветку.

Подключение

Схема

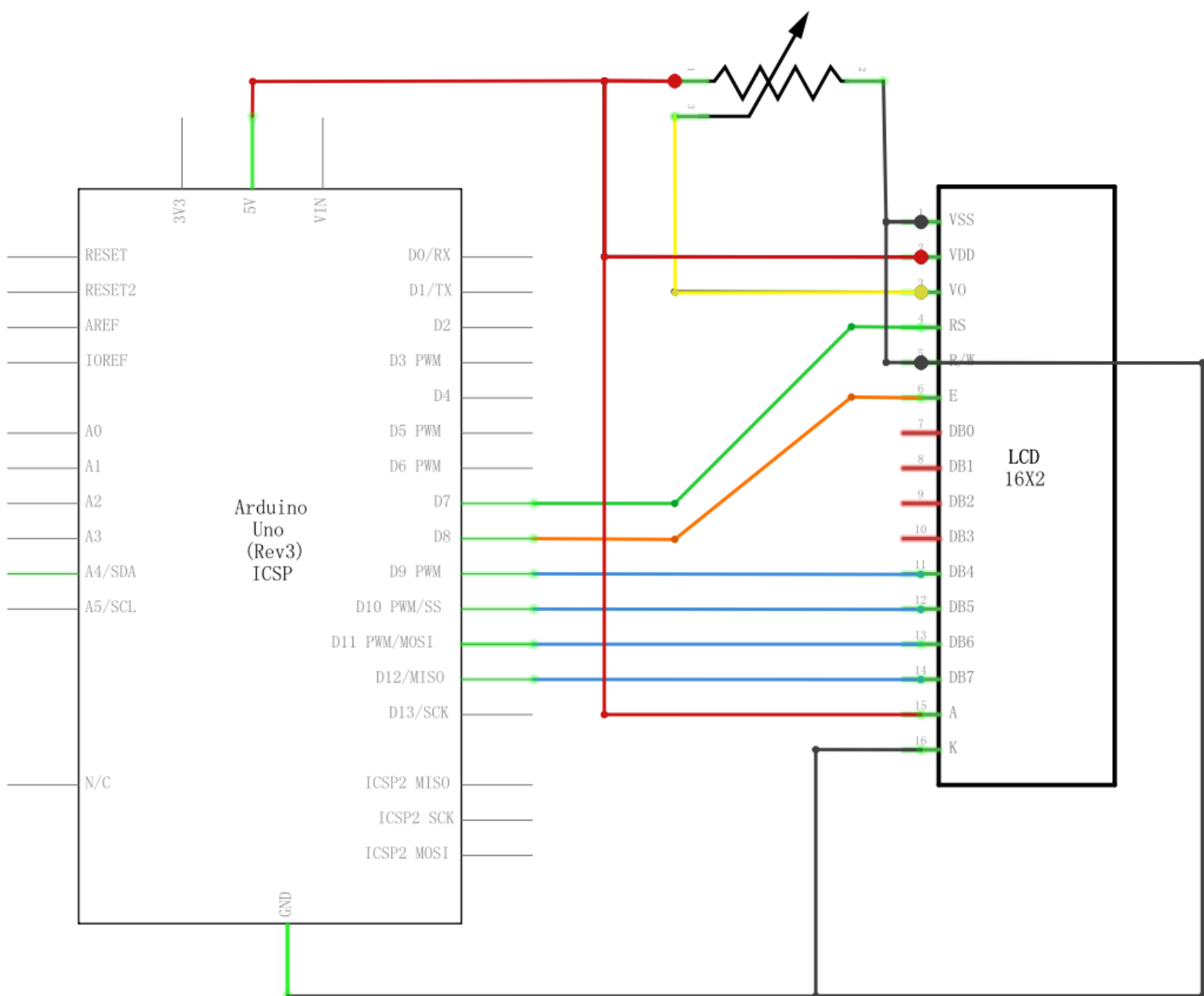
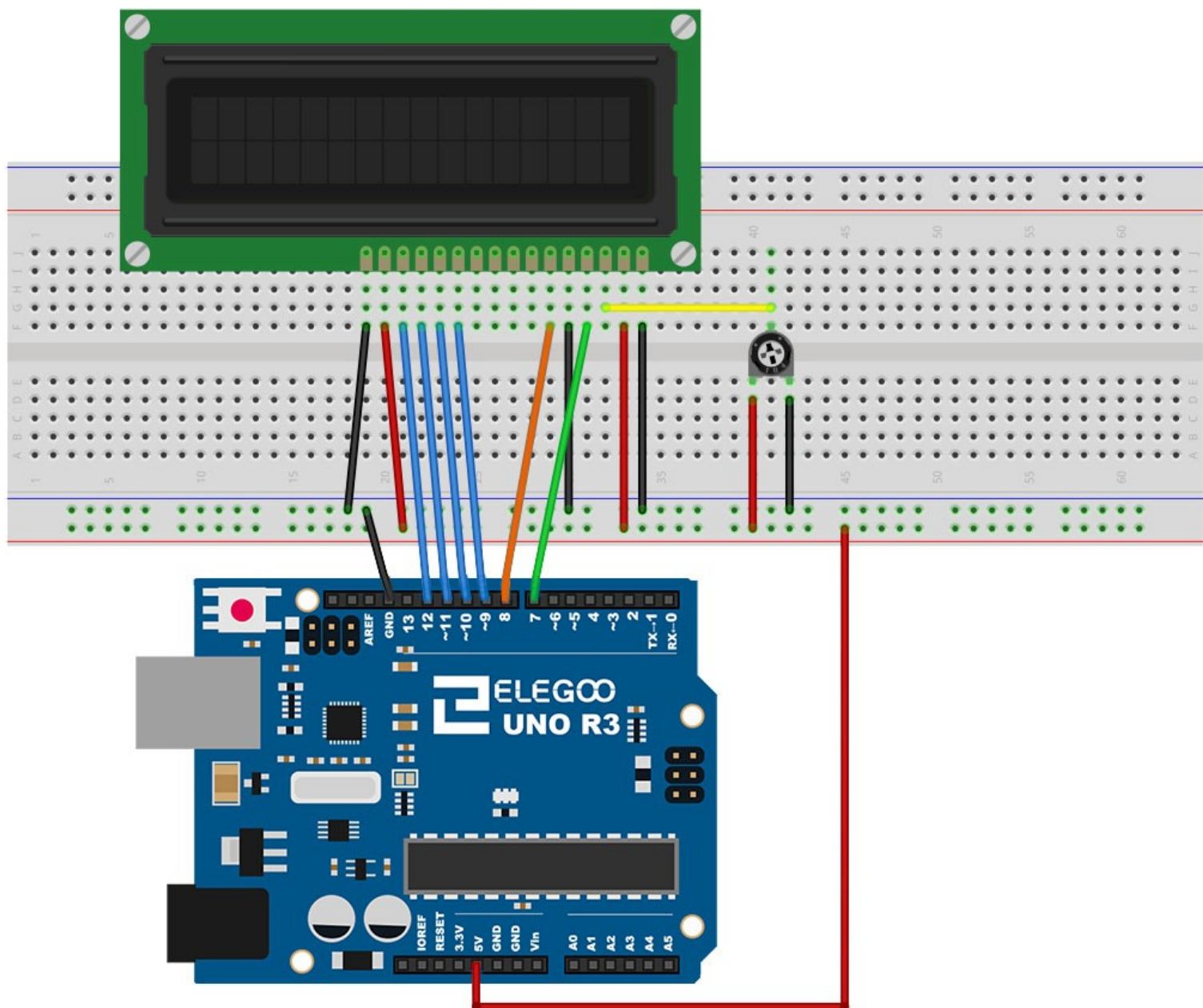


Схема монтажа



Подключать ЖКИ необходимо к 6 цифровым выводам на плате Arduino. Также необходимы контакты + 5 В и GND. Сопоставление ЖКИ с верхним краем вашей макетной платы поможет легче идентифицировать его контакты, **особенно**, если строки макетной платы пронумерованы, начиная с верхней строки 1. Не забудьте о длинном желтом кабеле для подключения потенциометра к выводу 3 ЖКИ. Потенциометр используется для регулировки контрастности ЖКИ. Если ваш ЖКИ не содержит распаянных коннекторов, следуйте инструкциям в следующем уроке.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 22: Жидкокристаллический индикатор (ЖКИ)”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры. Перед запуском, убедитесь, что вы установили библиотеку < LiquidCrystal >. Переустановите, если необходимо, иначе, ваш код не будет работать. Процедура добавления библиотек описана в Уроке 1.

Загрузите код на вашу плату Arduino. Вы должны увидеть сообщение 'hello, world', сопровождаемое отличным от нуля числом.

Первая строка в скетче:

```
#include <LiquidCrystal.h>
```

Это сообщает Arduino о том, что мы хотим использовать библиотеку “LiquidCrystal”. Далее следует строка, которую мы должны исправить. Она инициализирует пины Arduino, которые будут подключаться к ЖКИ.

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

После загрузки кода убедитесь, что светодиод работает и отрегулируйте контрастность ЖКИ с помощью потенциометра, пока вы не увидите тестовое сообщение.

В функции 'setup' имеем две команды: `lcd.begin(16, 2);`

```
lcd.print("Hello, World!");
```

Первая сообщает библиотеке о количестве строк и столбцов ЖКИ. Вторая команда отображает сообщение, которое мы видим на экране.

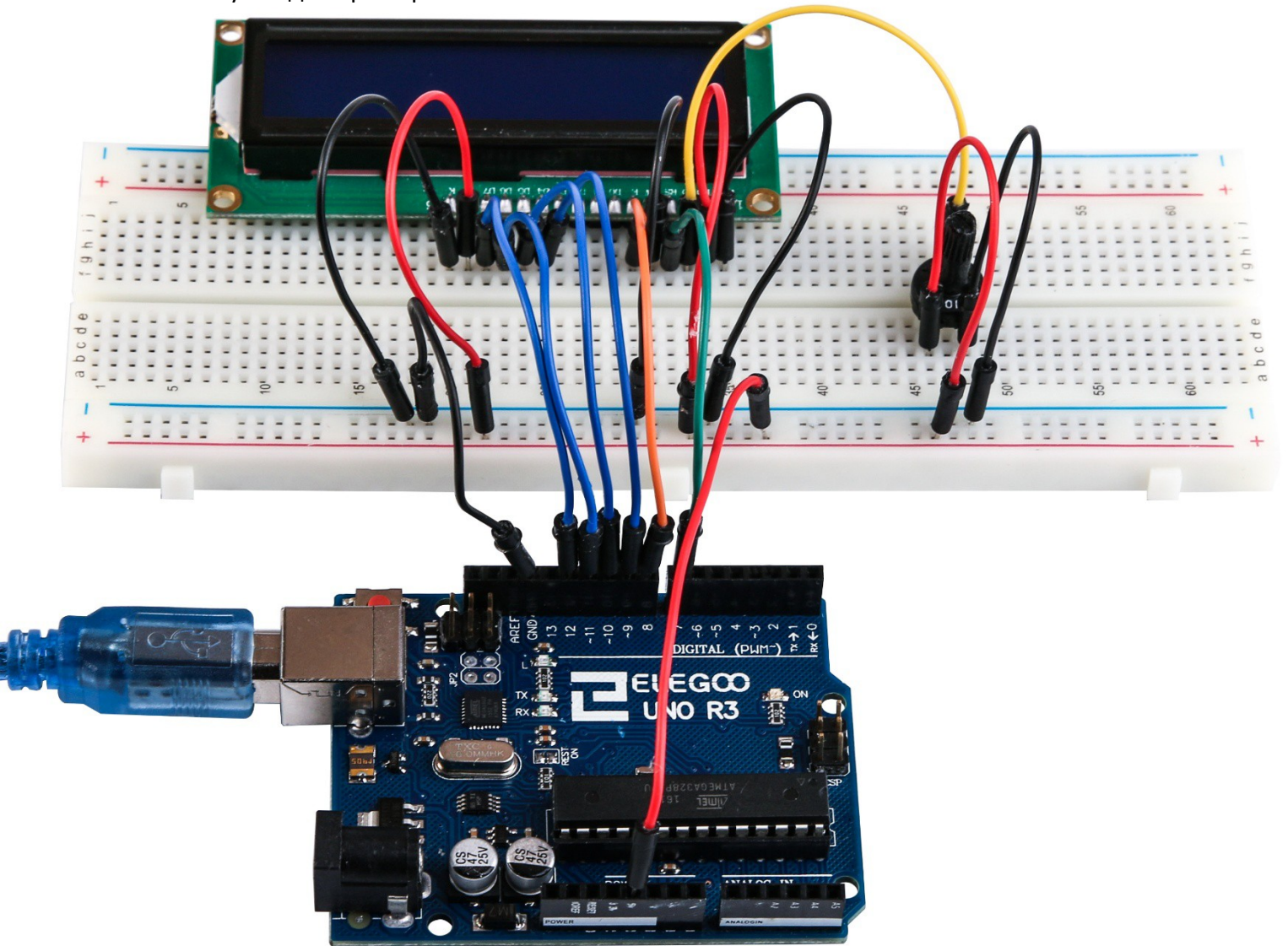
Функция 'loop' тоже содержит две команды: `lcd.setCursor(0, 1);`

```
lcd.print(millis()/1000);
```

Первая устанавливает положение курсора (где появится следующее текстовое сообщение) в столбец 0 и строку 1. Нумерация столбцов и строк начинается с 0.

Вторая отображает время в миллисекундах после сброса Arduino.

Рисунок для примера



Урок15: Термометр

Обзор

В этом уроке вы научитесь использовать ЖКИ для отображения температуры.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x Индикатор LCD1602
- (1) x резистор 10 кОм
- (1) x термистор
- (1) x потенциометр
- (1) x макетная плата на 830 точек
- (18) x перемычки (типа “папа” – “папа”)

Представление компонентов

Термистор

Термистор (терморезистор) - это резистор, который меняет свое сопротивление с изменением температуры. Технически все резисторы являются термисторами, так как их сопротивление меняется в зависимости от температуры. Но эти изменения очень незначительны и измерить их очень сложно. Термисторы изготавливаются таким образом, чтобы сопротивление изменялось на значительную величину в зависимости от температуры. Около 100 Ом и даже больше при изменении температуры на 1 градус по Цельсию!

Существуют два вида термисторов - с NTC (negative temperature coefficient - отрицательный температурный коэффициент) и с PTC (positive temperature coefficient - положительный температурный коэффициент). В большинстве случаев для измерения температуры используются NTC сенсоры. PTC часто используются в качестве предохранителей - с увеличением температуры растет сопротивление, это приводит к тому, что через них проходит большая сила тока, они нагреваются и срабатывают как предохранители. Достаточно удобно для предохранительных цепей!

Соединение

Схема

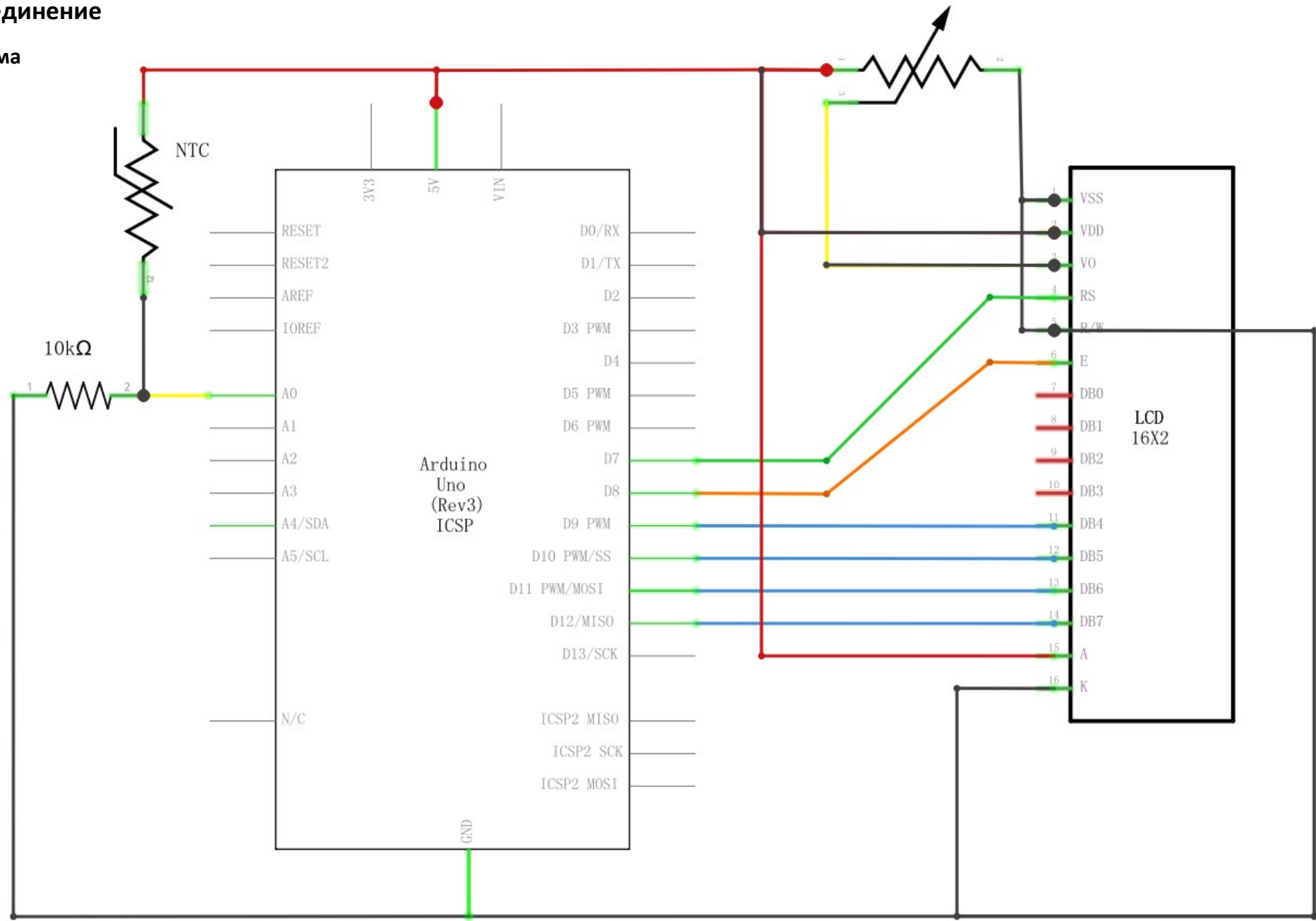


Схема монтажа

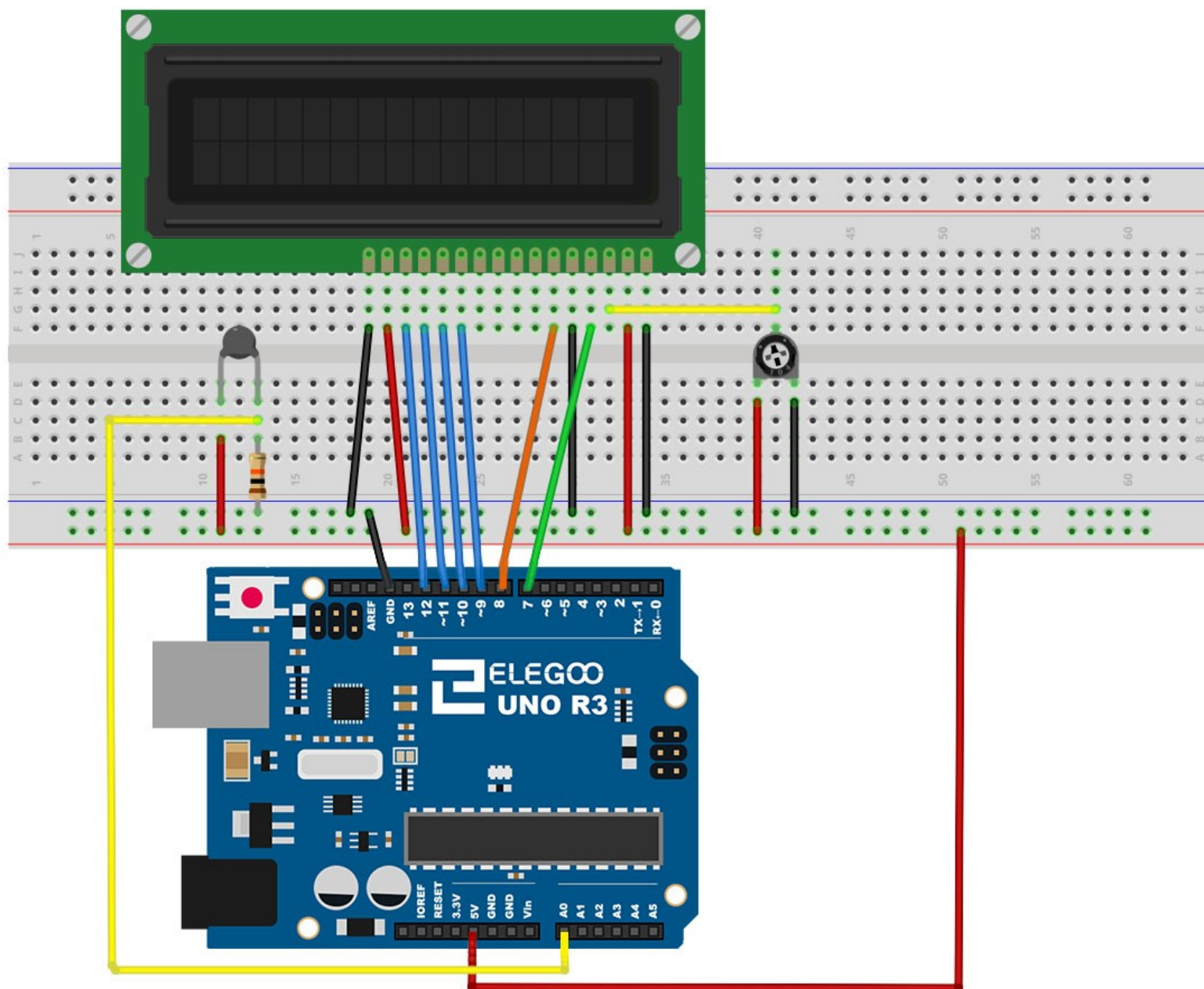


Схема подключения предусматривает использование той же макетной платы, что и в Уроке 22. Положение некоторых перемычек возле потенциометра было изменено. Также, добавлены резистор и термистор.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 23: Термометр”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Перед запуском, убедитесь, что вы установили библиотеку < LiquidCrystal >. Переустановите, если необходимо, иначе, ваш код не будет работать. Процедура добавления библиотек описана в Уроке 1.

Этот скетч основывается на программе, показанной в Уроке 22. Загрузите его на свою плату Arduino. Вы должны увидеть, что прикосновение пальца к датчику температуры меняет показания температуры на экране.

Полезно добавить комментарий над командой 'lcd'.

```
// BS      E      D4 D5      D6  D7  
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Это должно упростить ситуацию, если вы решите поменять используемые пины.

В функции 'loop' имеют место следующие интересные вещи. Во-первых, мы должны преобразовать аналоговые показания термистора в «реальные» значения температуры и, во-вторых, мы должны понять, как вывести их на дисплей.

Сначала посмотрим, как происходит расчет температуры.

```
int tempReading = analogRead(tempPin);  
double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));  
tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK ))  
* tempK );  
float tempC = tempK - 273.15;  
float tempF = (tempC * 9.0) / 5.0 + 32.0;
```

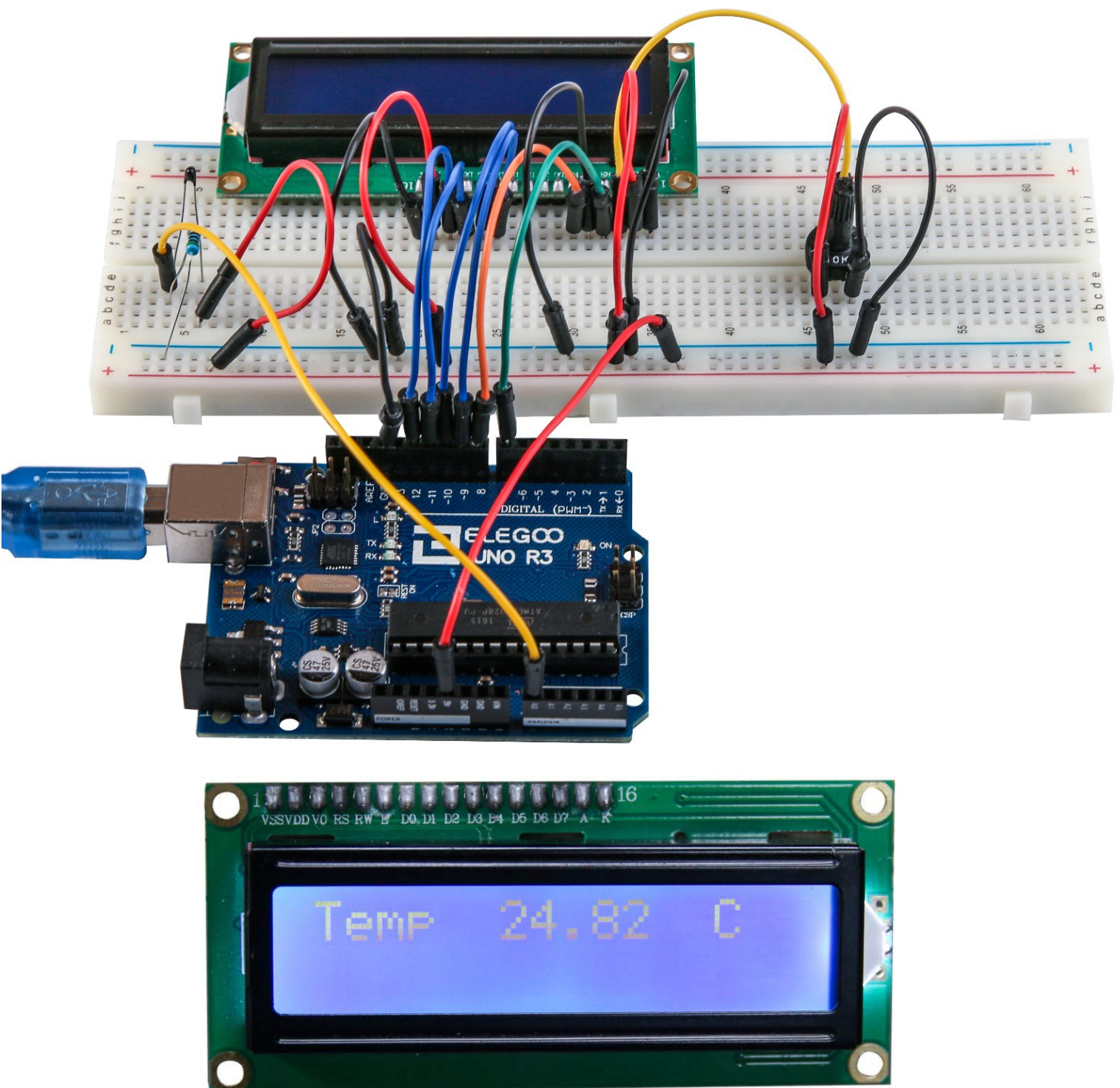
Вывод меняющихся данных на ЖКИ представляет определенные сложности. Главной проблемой является то, что значение температуры не всегда будет содержать одинаковое количество разрядов. Например, если температура понизилась с 101.50° до 99.00°, существует риск, что дополнительный разряд

предыдущего показания может остаться на экране. Во избежание этого, записывайте строку показания ЖКИ каждый раз для нового цикла.

```
lcd.setCursor(0, 0); lcd.print("Temp  C "); lcd.setCursor(6, 0); lcd.print(tempF);
```

Этот довольно странный комментарий служит напоминанием того, что ЖКИ содержит 16 столбцов. Вы можете вывести строку данных соответствующей длины с пробелами, в которых будут отображаться «реальные» показатели. Для заполнения пробелов установите курсор в положение, где должно отобразиться значение, а затем выведите его на экран.

Рисунок для примера



Урок 16: Восемь светодиодов и сдвиговый регистр 74HC595

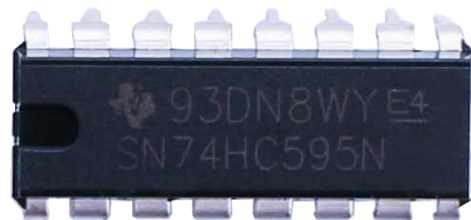
Обзор

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:

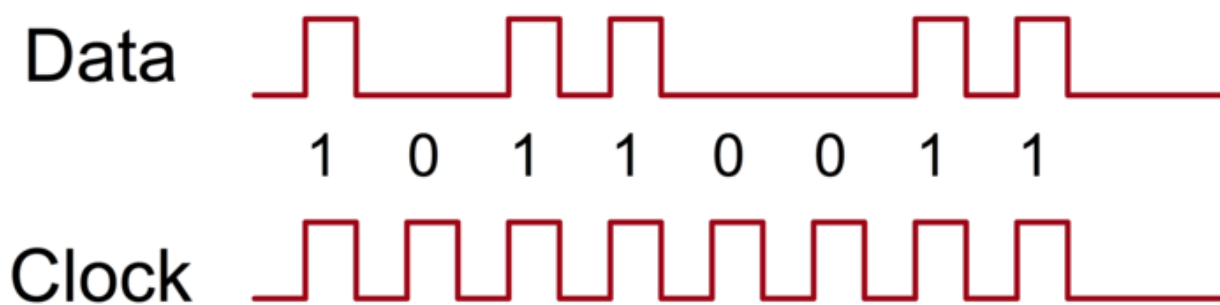
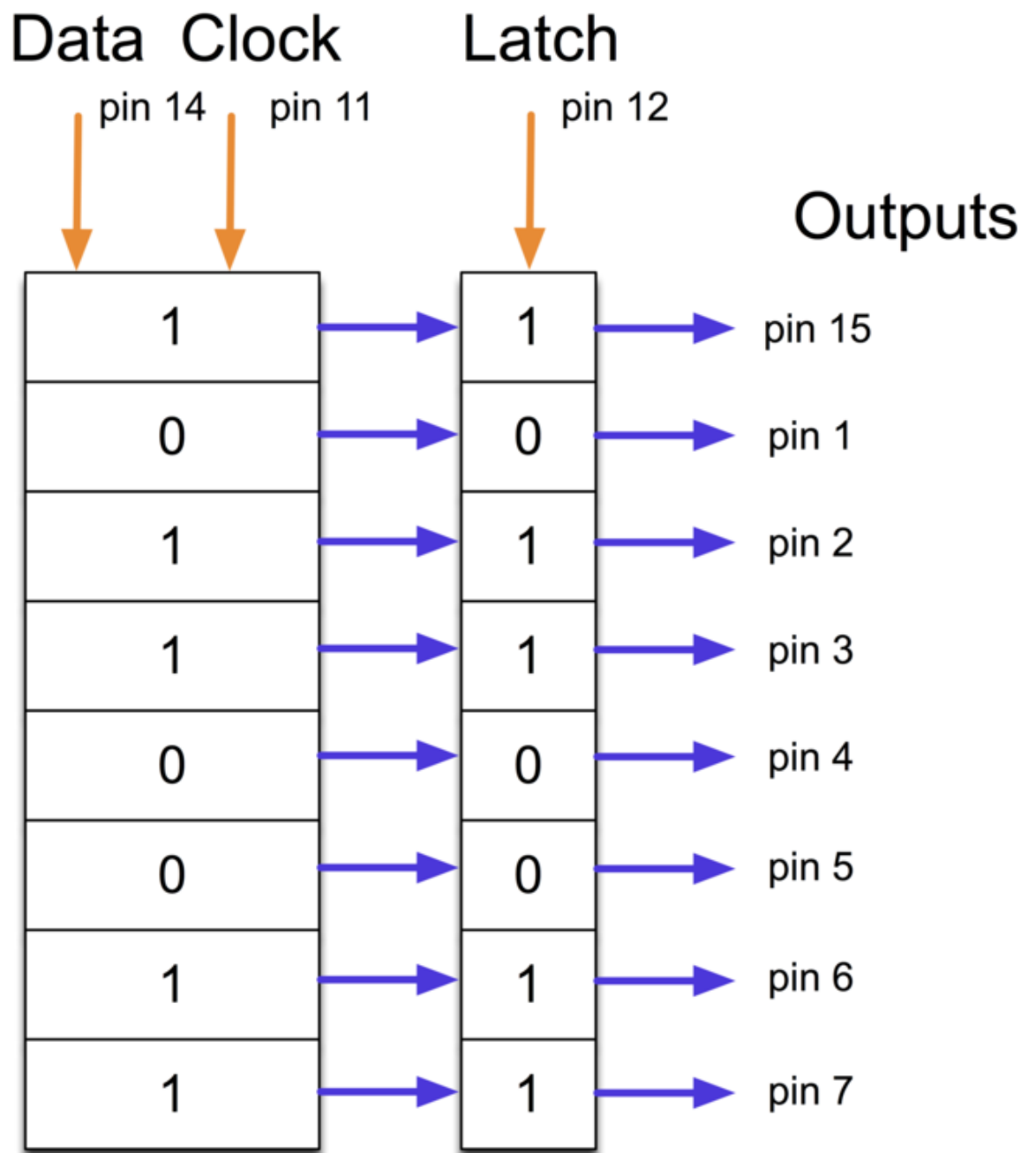
- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (8) x светодиодов
- (8) x резистор (220 Ом) |
- (1) x сдвиговый регистр 74HC595
- (14) x перемычки (типа “папа” – “папа”)



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).



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 противном 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

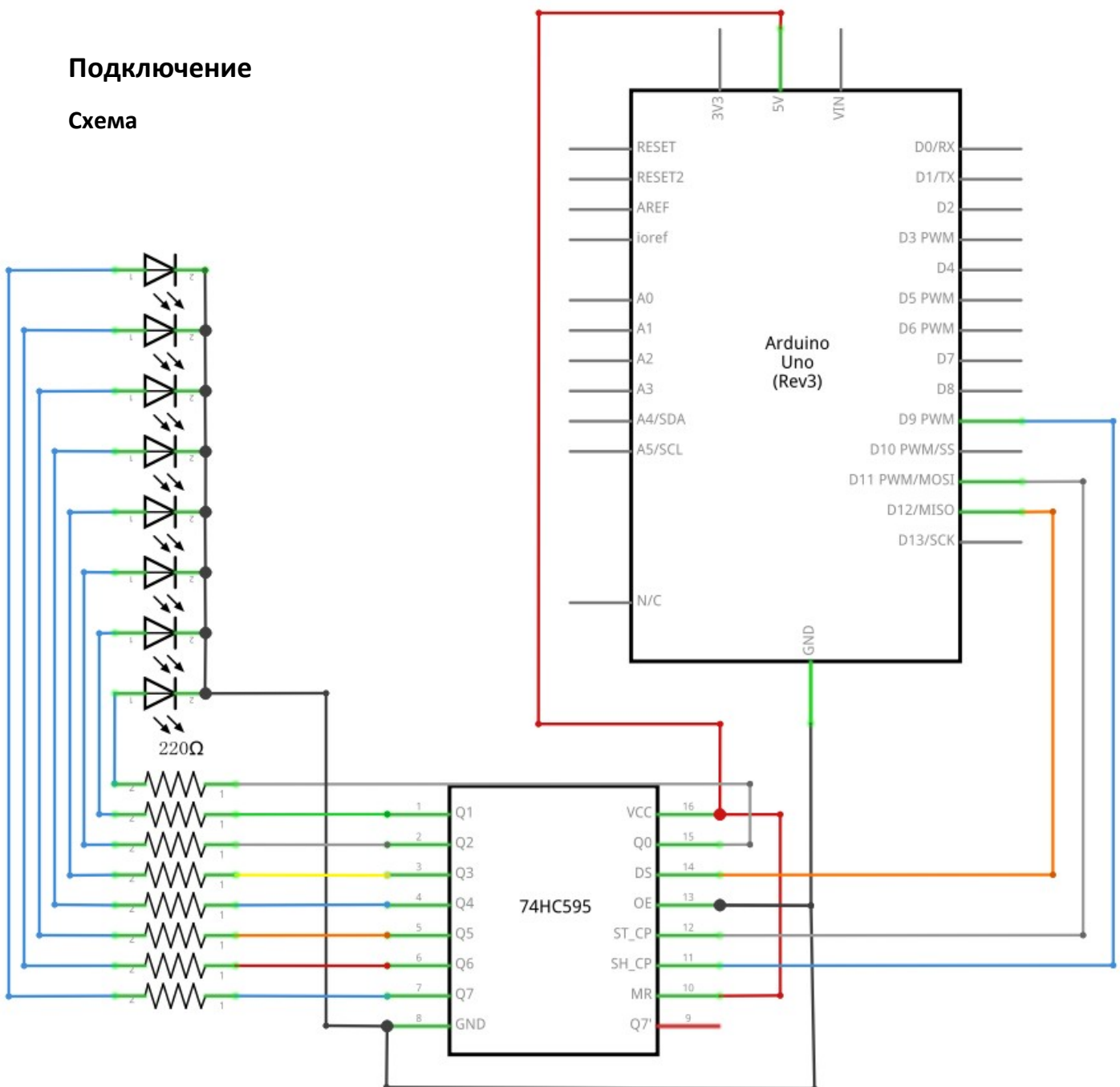
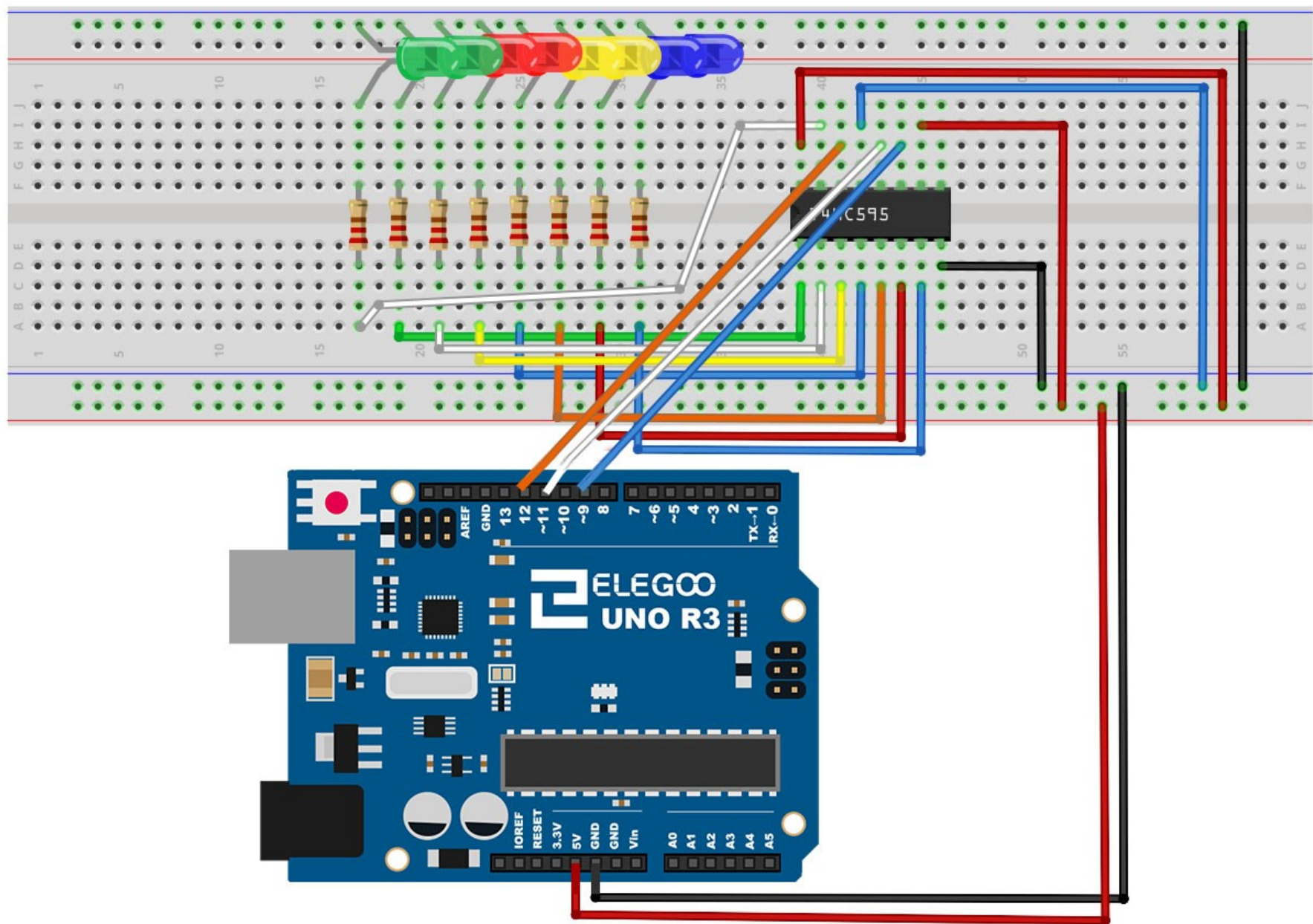


Схема монтажа



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 не 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);  
}  
}
```

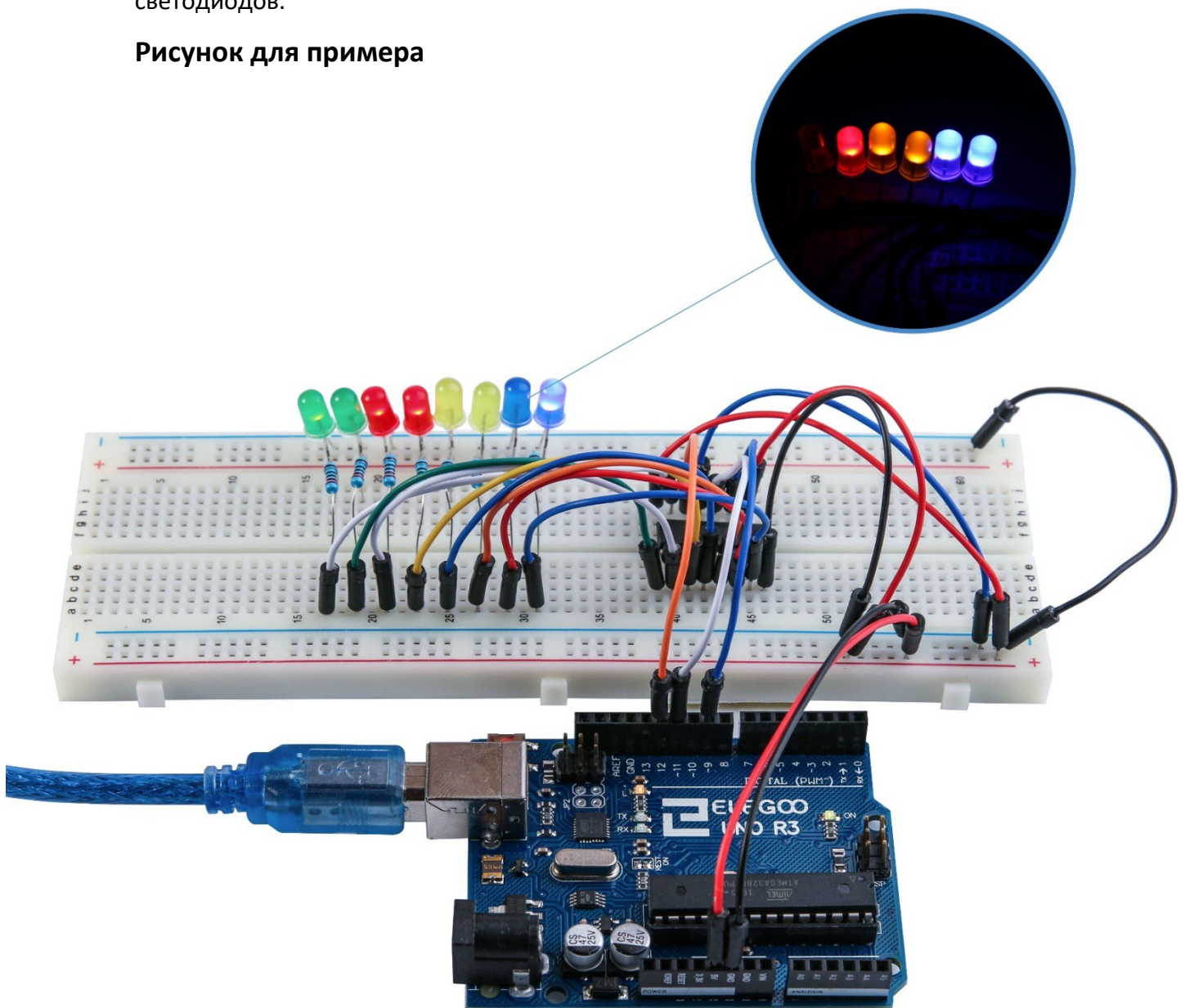
Функция 'updateShiftRegister', в первую очередь, устанавливает значение LOW в latchPin, затем вызывает функцию 'shiftOut' перед повторной установкой значения

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



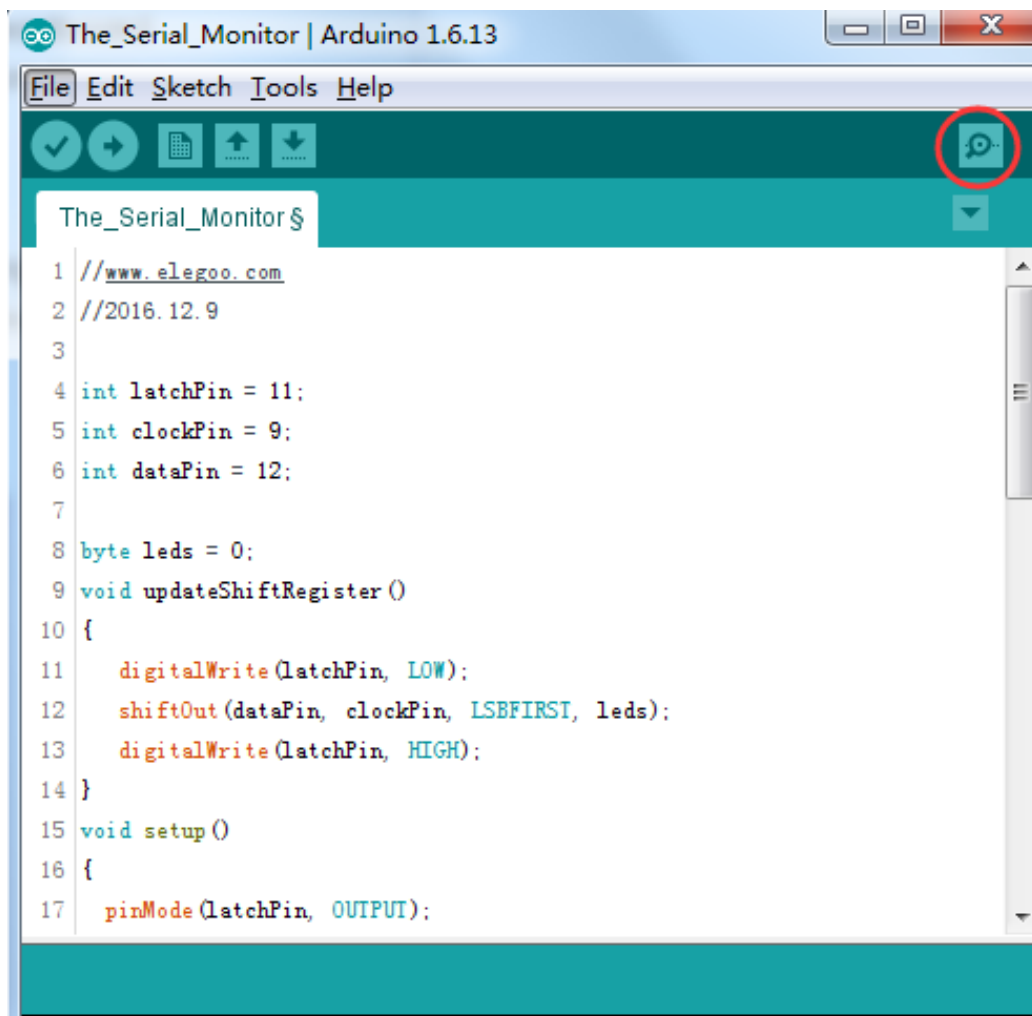
Урок 17: Серийный монитор

Обзор

Этот урок является продолжением Урока 24, 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 24. Если вы еще не ознакомились с материалом Урока 24, пожалуйста, выполните его перед началом этого урока.

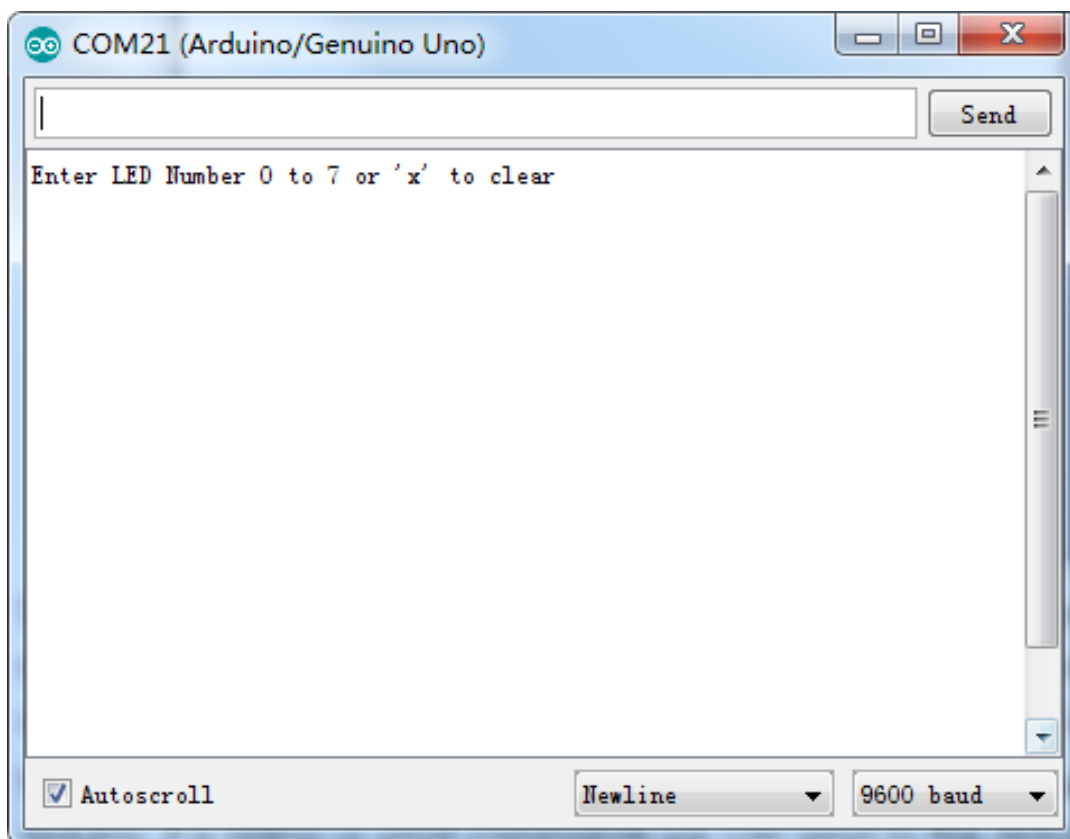
Необходимые шаги

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.

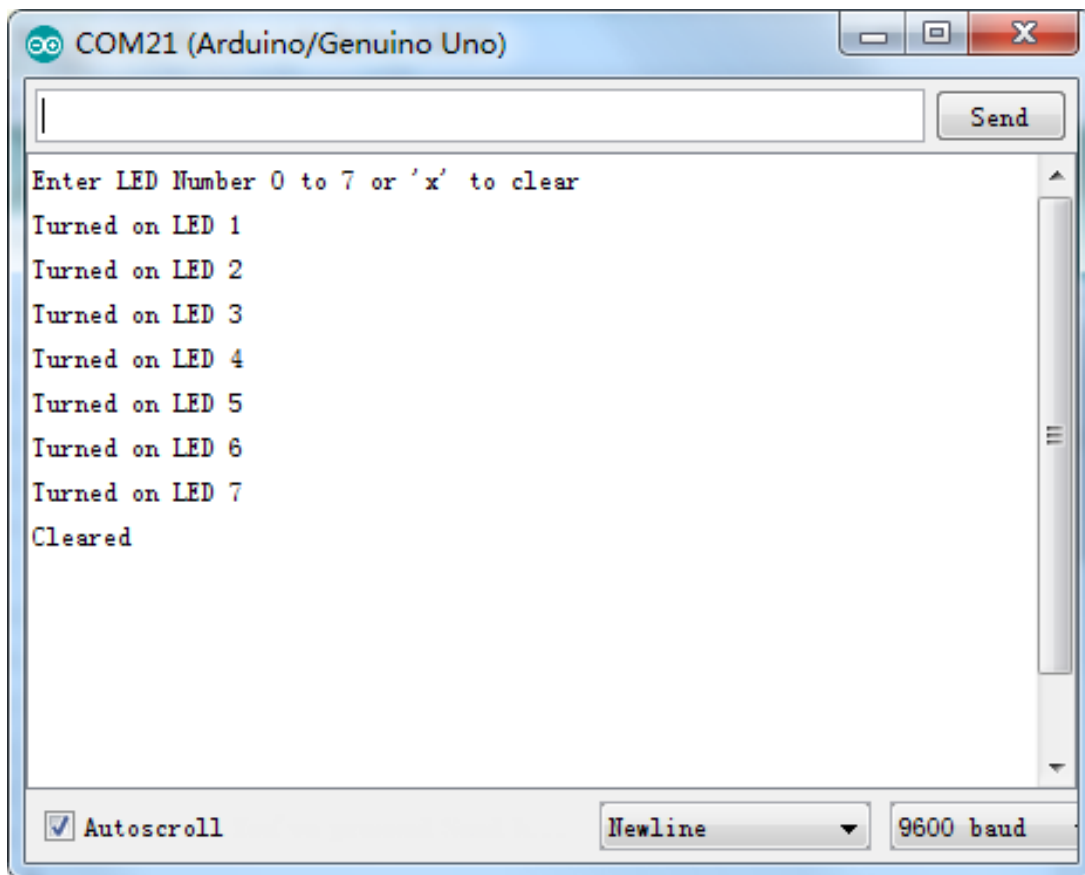


Это окно называется серийным монитором и является частью среды разработки Arduino. Задачи серийного монитора включают отправку сообщений с вашего компьютера на плату UNO (через USB-порт) и получение сообщений с платы.

Сообщение “Enter LED Number 0 to 7 or '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 18: 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:

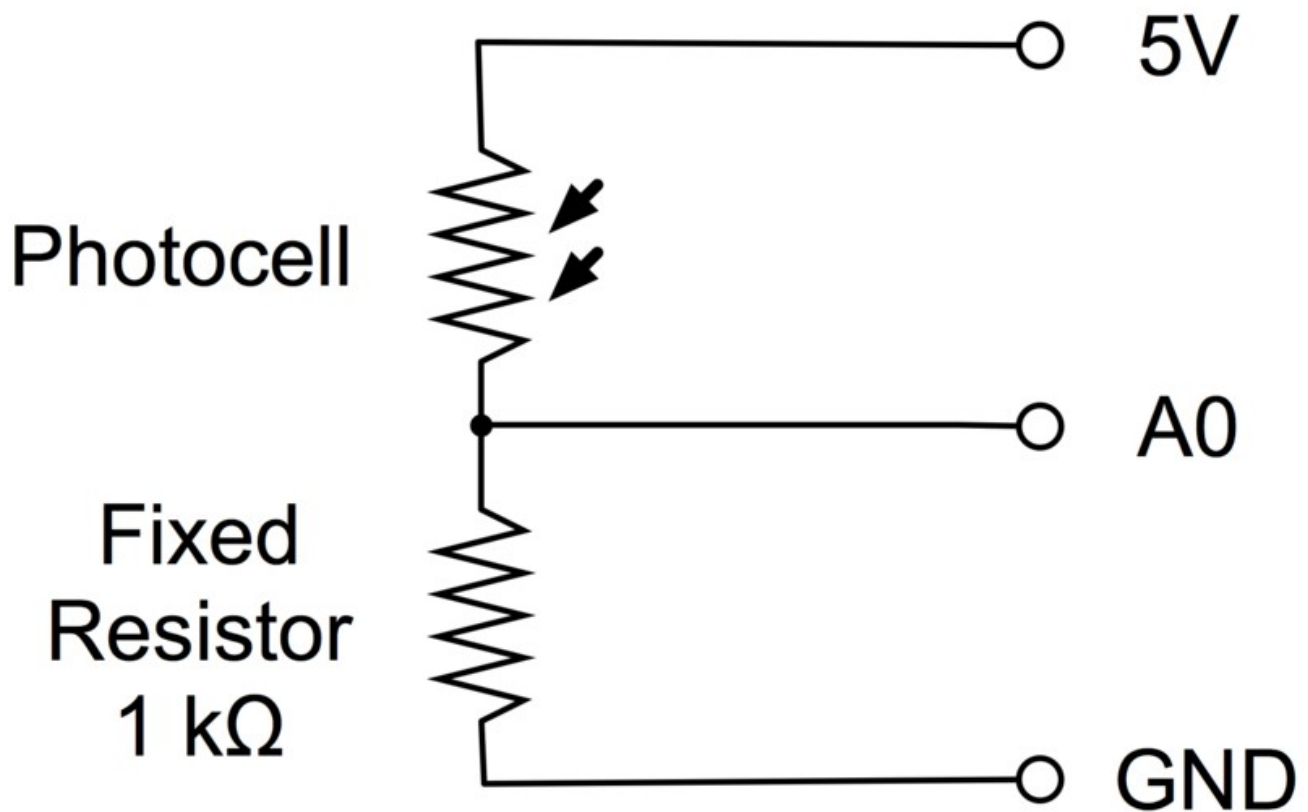
- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (8) x светодиоды
- (8) x резистор (220 Ом)
- (1) x резистор (1 кОм)
- (1) x сдвиговый регистр 74HC595
- (1) x фоторезистор
- (16) x перемычки (типа “папа” – “папа”)



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

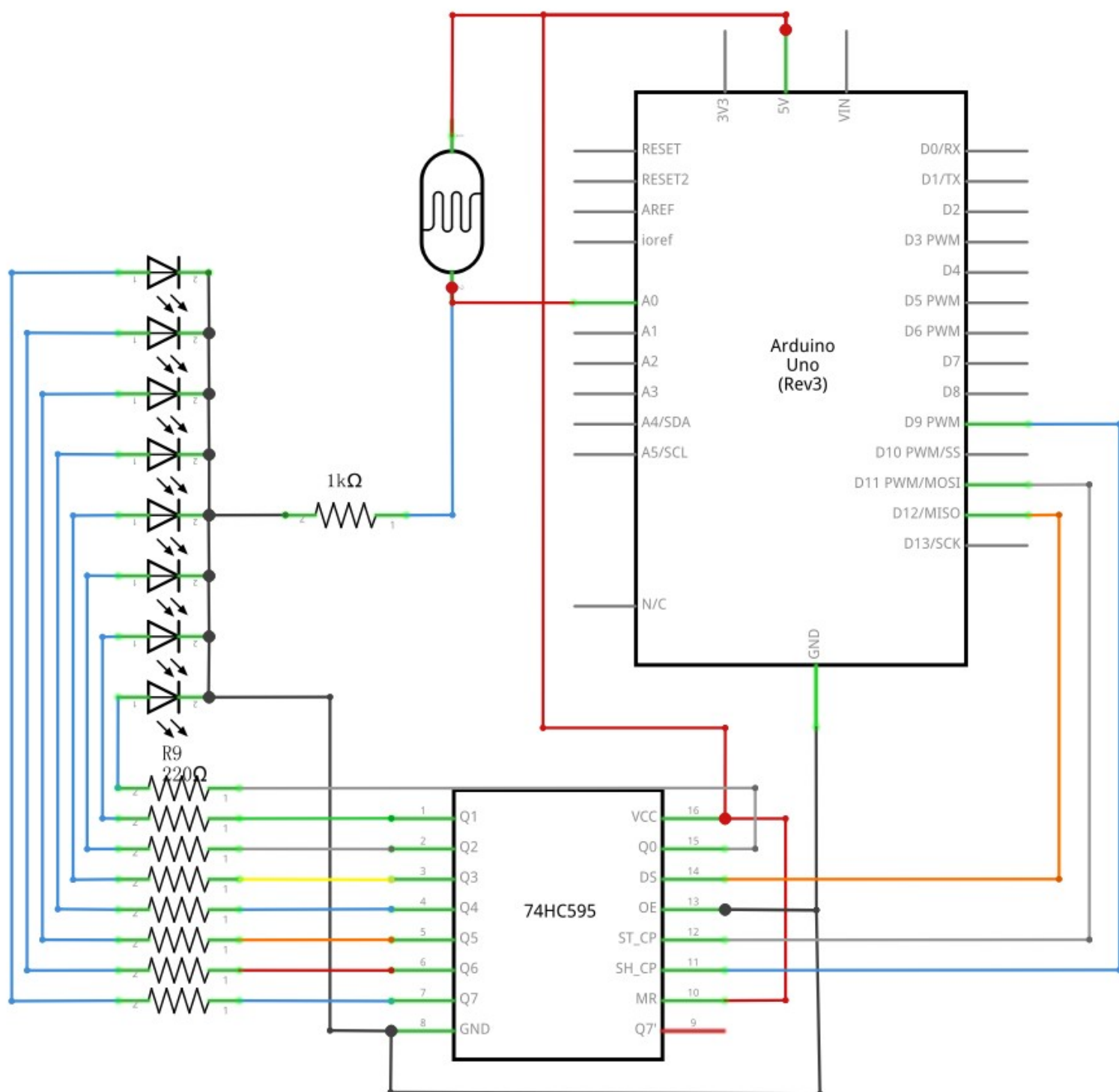
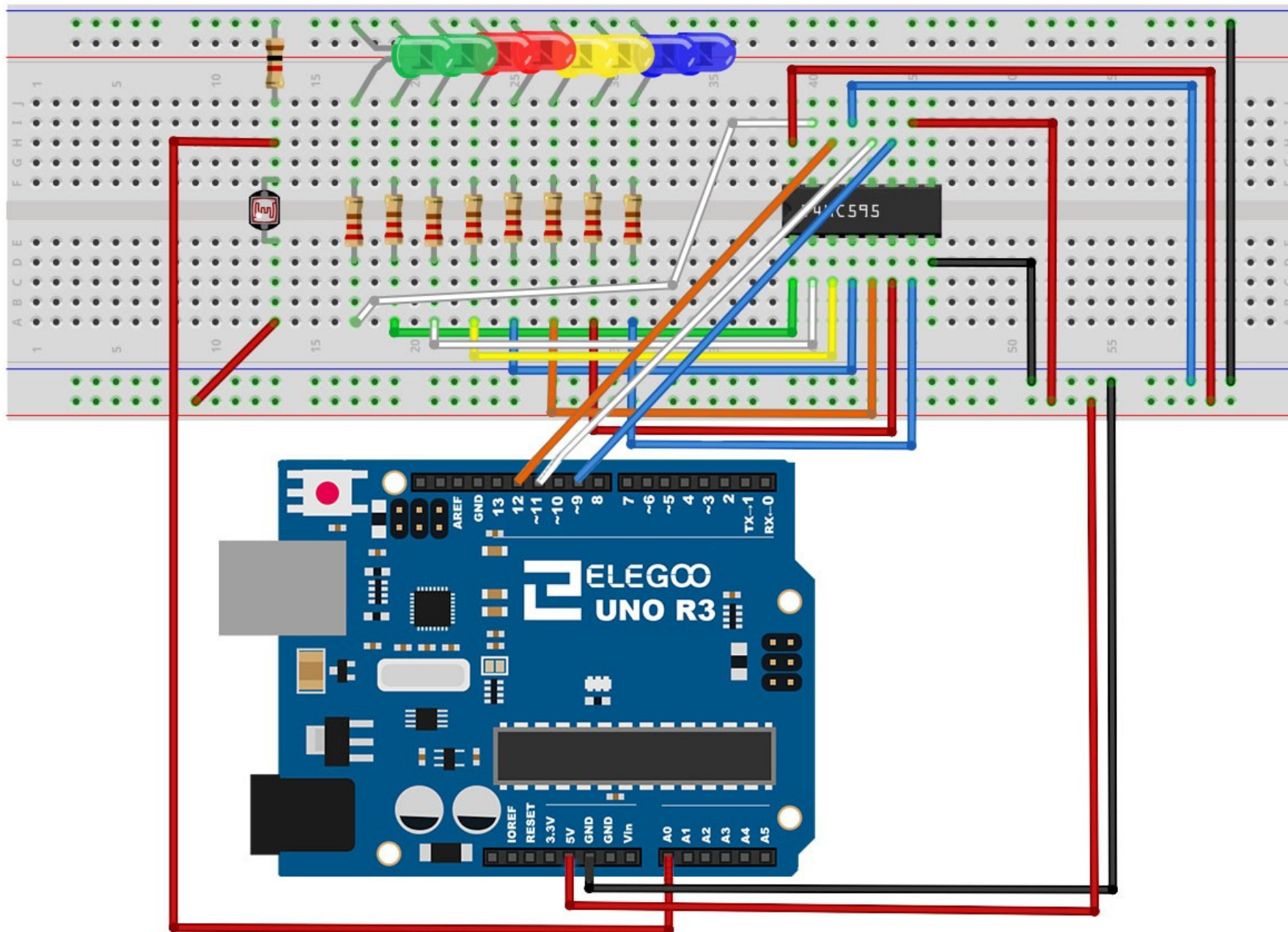


Схема монтажа



Код

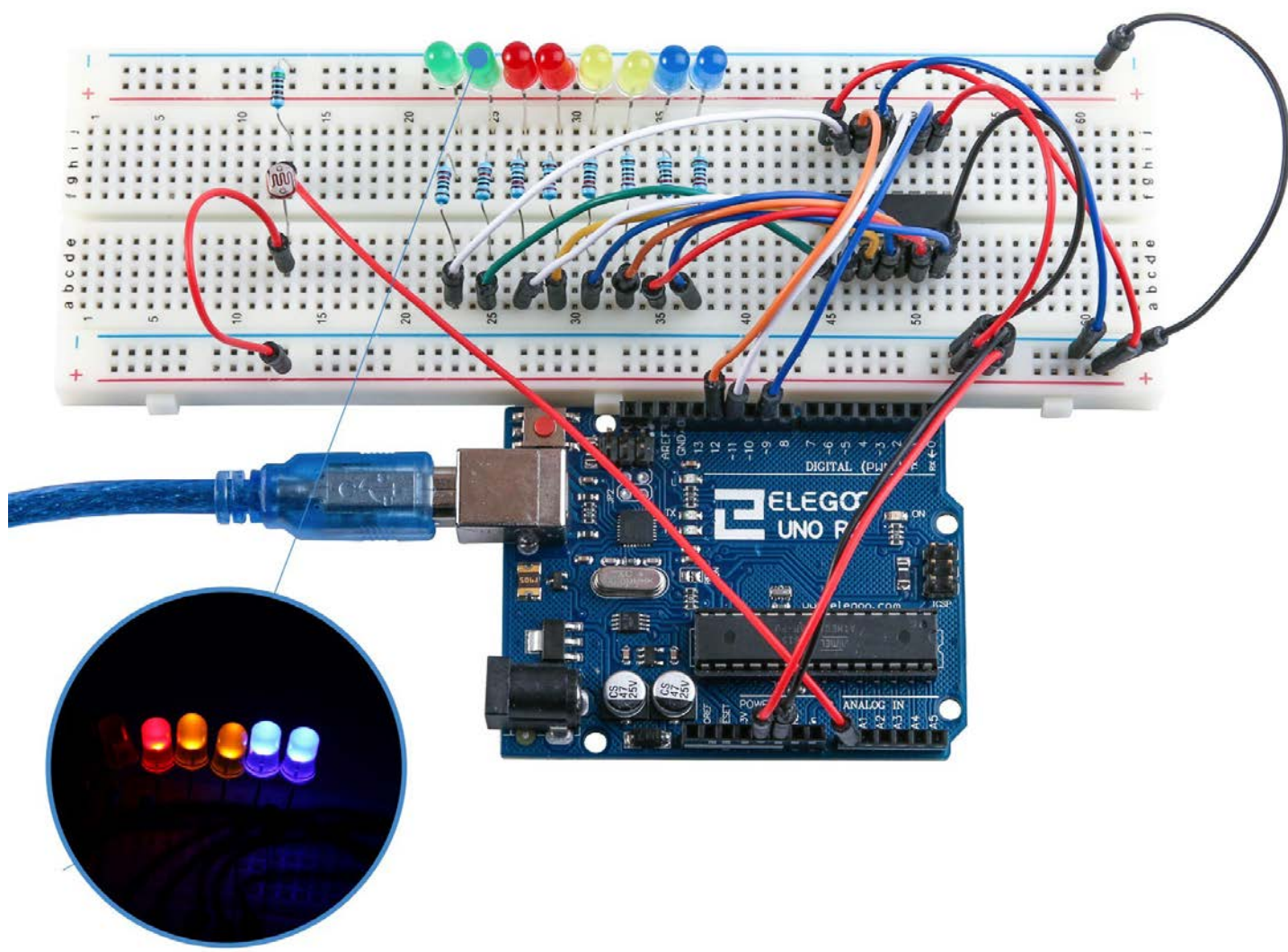
После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 26: Фоторезистор”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

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 строка, 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 деления на 9 областей (начиная с включения 0 светодиодов до всех 8 светодиодов). Это 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



Урок 19: Сегментный индикатор и 74HC595

Обзор

После изучения материала из Уроков 24, 25 и 26, мы будем использовать сдвиговый регистр для управления сегментным индикатором. Сегментный индикатор отображает числа от 0 до 9.

Необходимые компоненты:

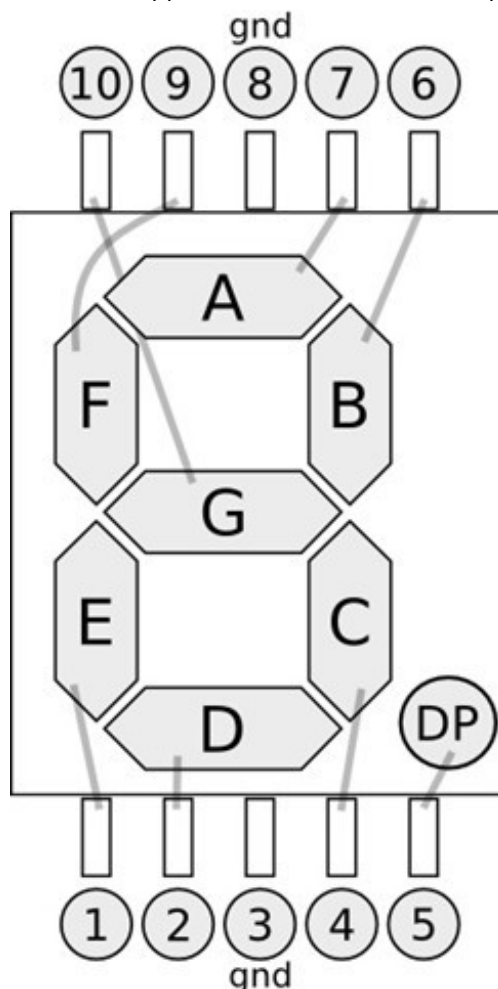
- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (1) x сдвиговый регистр 74HC595
- (1) x одnorазрядный семисегментный индикатор
- (8) x резистор (220 Ом)
- (26) x перемычки (типа “папа” – “папа”)



Представление компонентов

Семисегментный индикатор

Ниже представлена схема выводов семисегментного индикатора.



Соответствие чисел от 0 до 9 каждому сегменту индикатора показано в таблице (настоящая таблица представлена для семисегментного индикатора с общим катодом, если вы используете индикатор с общим анодом, каждые '1 0 0' должны быть заменены на '1'):

Число дисплея	надр	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

Подключение

Схема

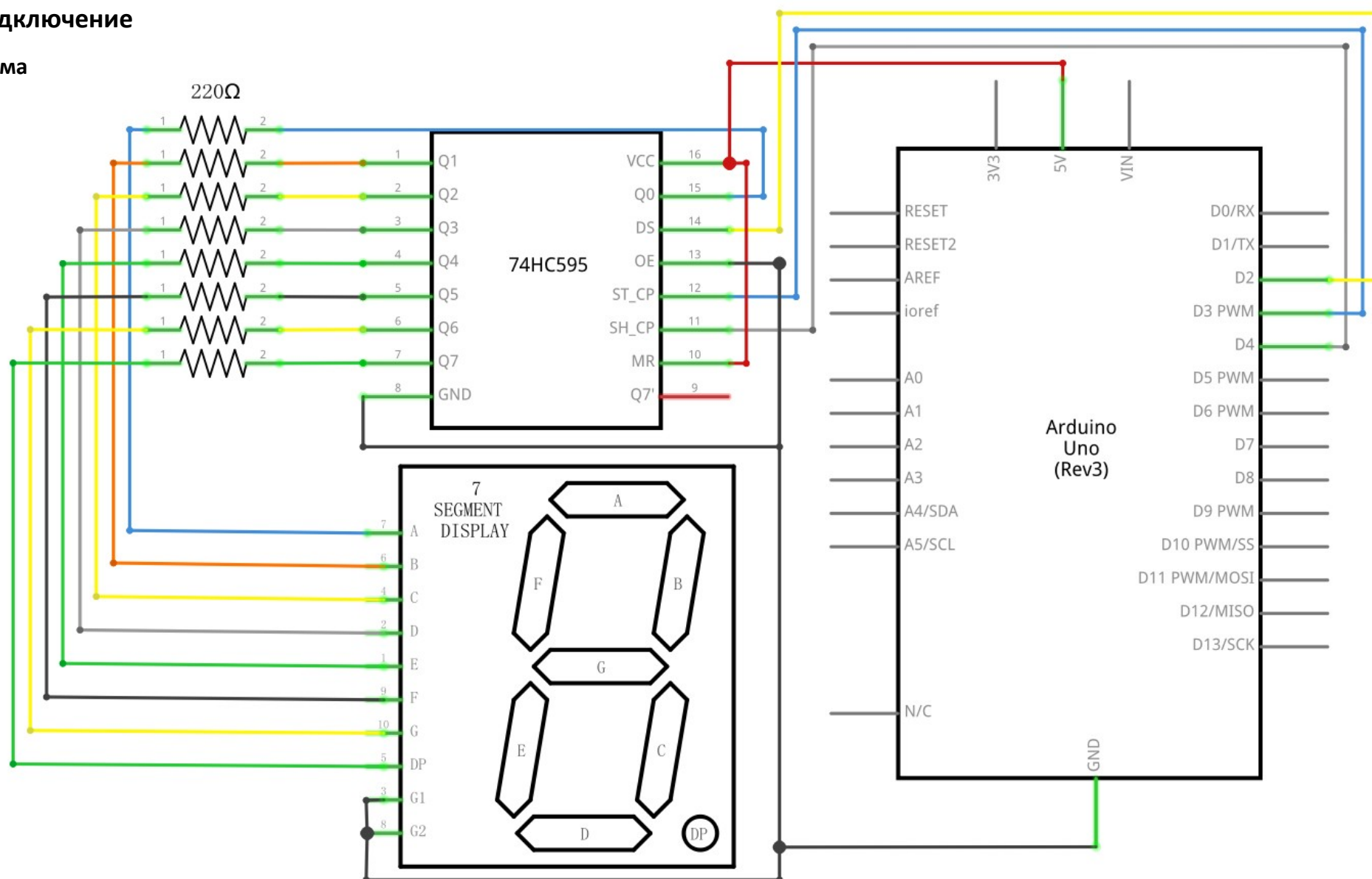


Схема монтажа

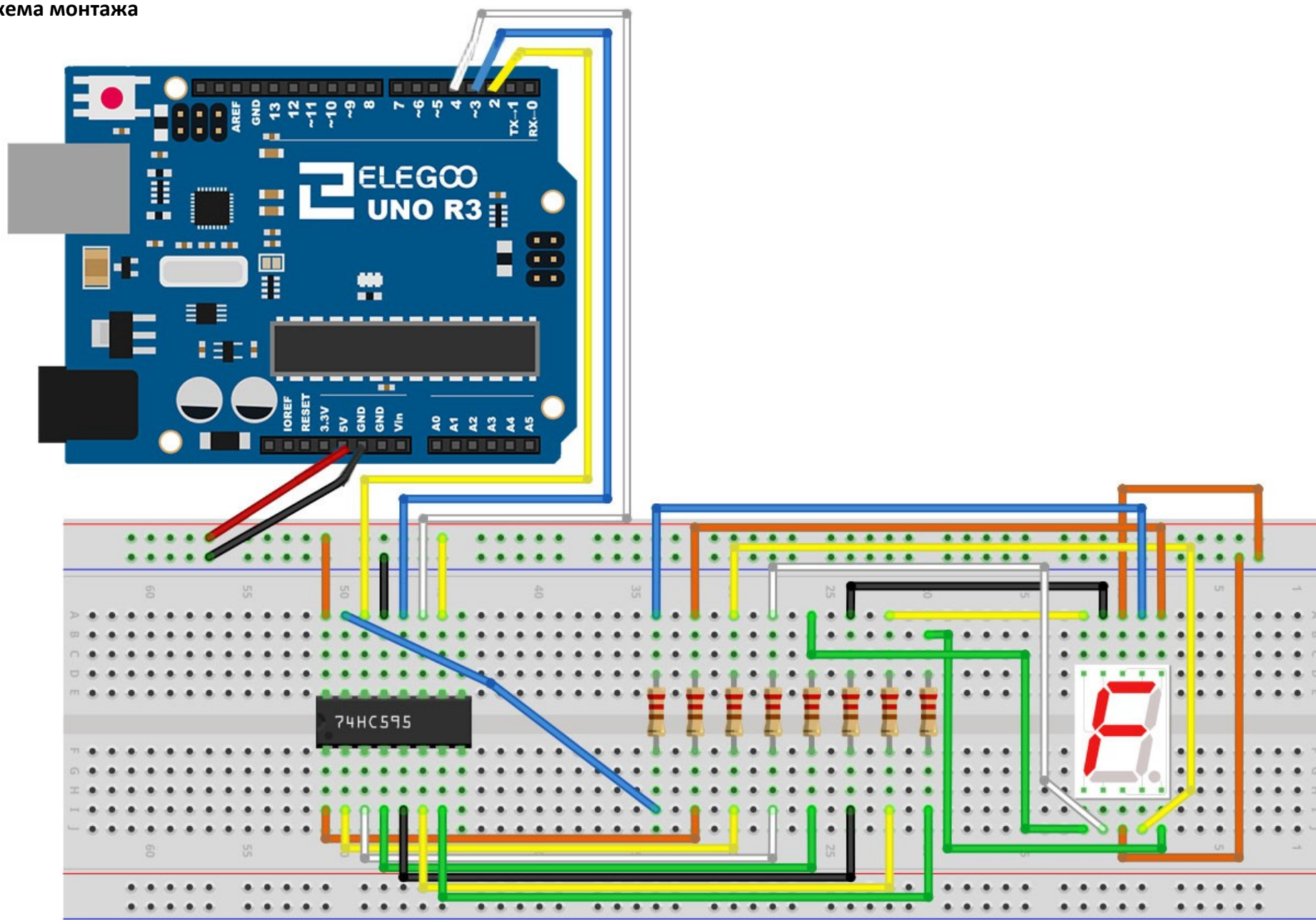


Таблица ниже показывает соответствие выводов сдвигового регистра 74HC595 и семисегментного индикатора для подключения:

Вывод 74HC595	Семисегментный индикатор
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

Шаг первый: подключите сдвиговый регистр 74HC595

Сначала, подключение выводов для питания и заземления:

VCC (вывод 16) и **MR** (вывод 10) подключены к + 5 В

GND (вывод 8) и **OE** (вывод 13) подключены к общему проводу

Выводы **DS**, **ST_CP** и **SH_CP**:

DS (вывод 14) подключен к выводу 2 платы UNO R3 (желтый цвет на рисунке ниже)

ST_CP (вывод 12) подключен к выводу 3 платы UNO R3 (синий цвет на рисунке ниже)

SH_CP (вывод 11) подключен к выводу 4 платы UNO R3 (белый цвет на рисунке ниже)

Шаг второй: подключите семисегментный индикатор

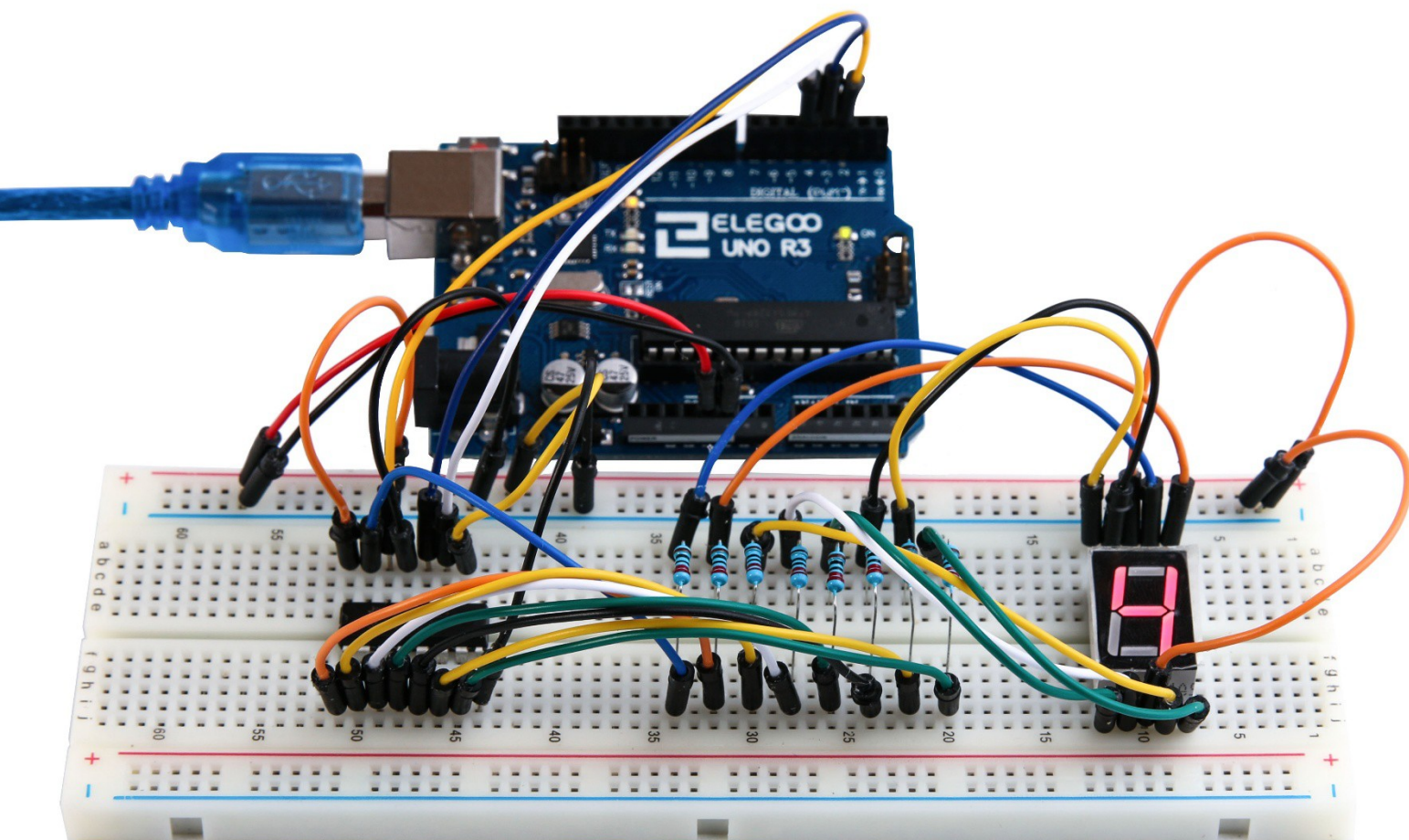
Индикатор подключается выводами 3, 8 к выводу GND платы UNO R3 (в случае с общим анодом – к выводу + 5 В).

Как показано в таблице выше, соедините выводы Q0 ~ Q7 регистра 74HC595 с соответствующими выводами индикатора (A ~ G и DP) последовательно и каждый через резистор в 220 Ом.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 27: Сегментный индикатор и 74НС595”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Рисунок для примера



Урок 20: Четырехразрядный семисегментный индикатор

Обзор

В этом уроке вы научитесь использовать четырехразрядный семисегментный индикатор.

Обратите внимание, что при использовании одноразрядного семисегментного индикатора, если он с общим анодом, то вывод общего анода подключается к источнику питания, если индикатор с общим катодом – к общему проводу (GND).

Используя четырехразрядный семисегментный индикатор, вывод общего анода или катода используется для управления отображающимся значением. Даже имея одно число на индикаторе, инерция зрения позволяет вам видеть все числа из-за высокой скорости сканирования.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (1) x сдвиговый регистр 74HC595
- (1) x четырехразрядный семисегментный индикатор
- (4) x резистор (220 Ом)
- (23) x перемычки (типа "папа" – "папа")



Four Digits Displays Series

Подключение

Схема

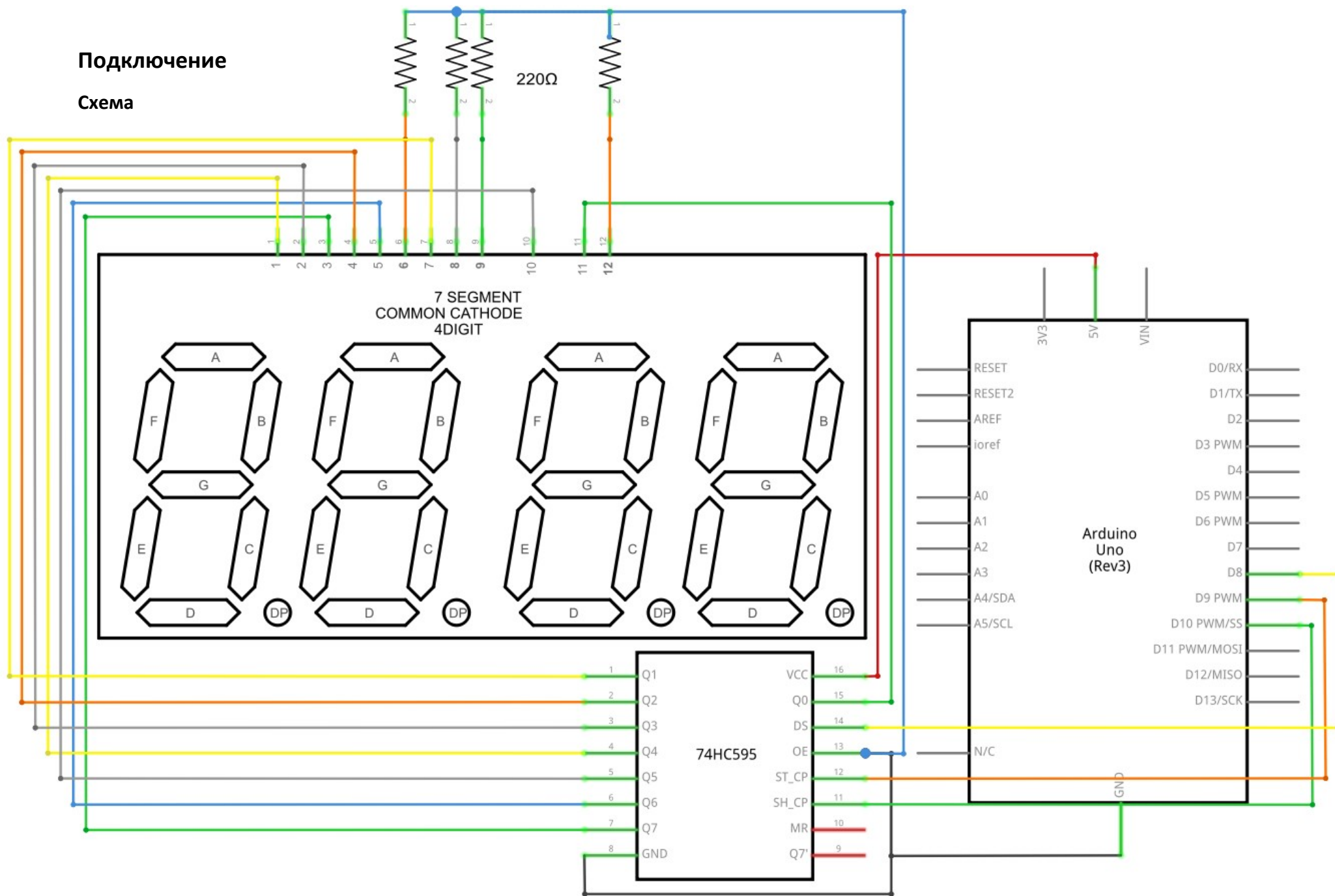
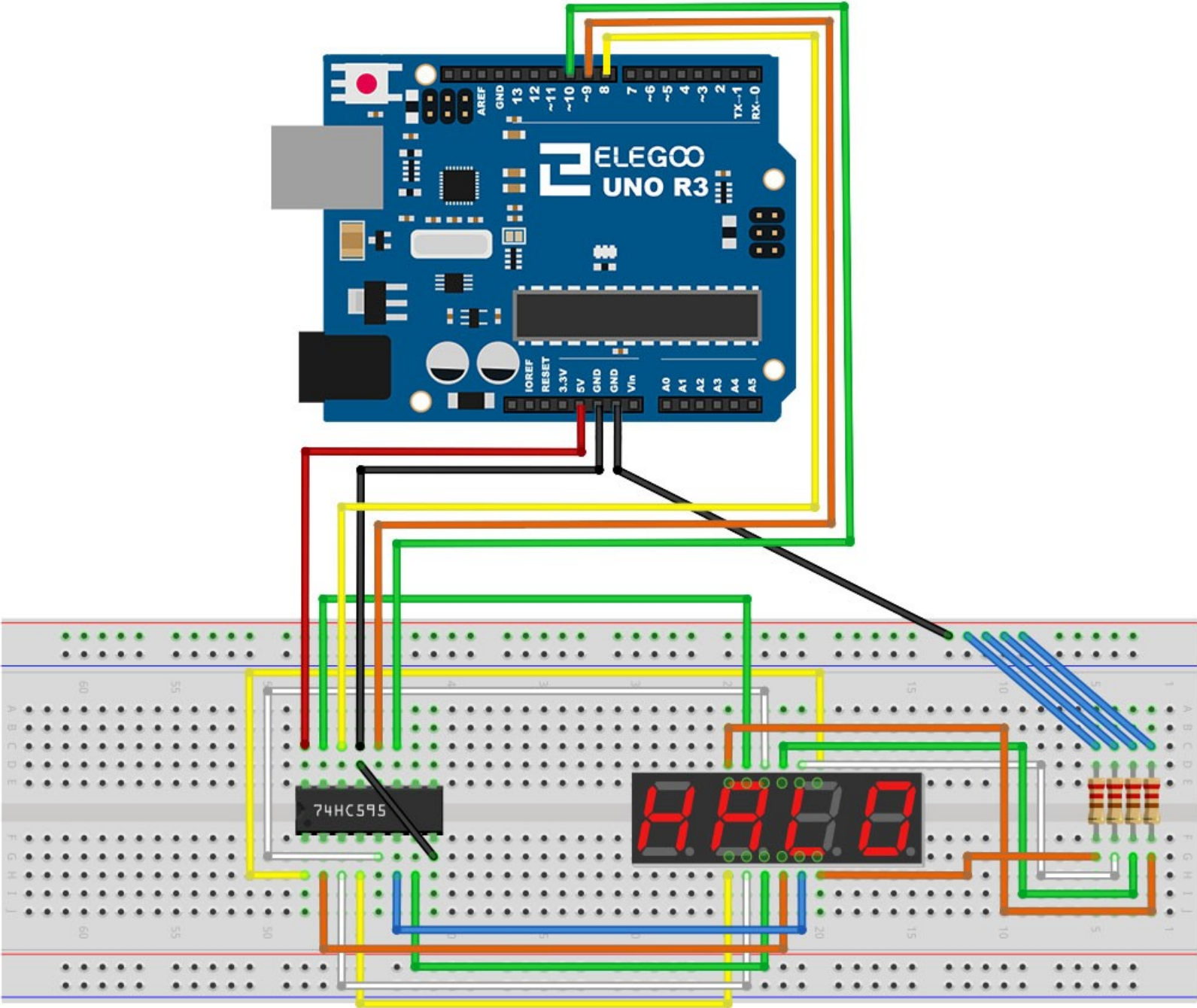


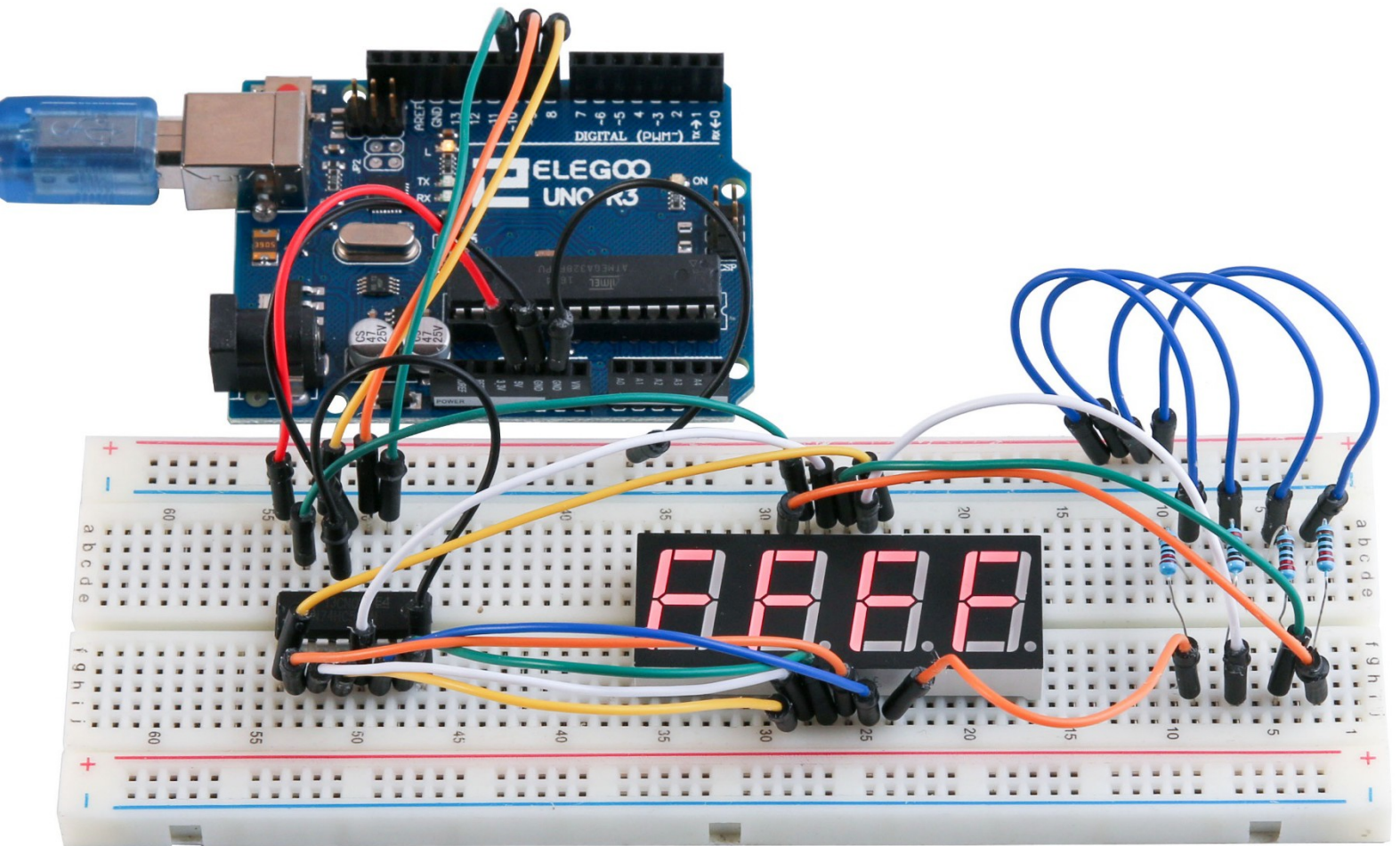
Схема монтажа



Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 28: : Четырехразрядный семисегментный индикатор”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Рисунок для примера



Урок 21: Двигатель постоянного тока

Обзор

В этом уроке вы научитесь контролировать небольшой двигатель постоянного тока, используя плату UNO R3 и транзистор.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (1) x микросхема L293D
- (1) x лопастной вентилятор и мотор постоянного тока (3-6 В)
- (5) x перемычки (типа “папа” – “папа”)
- (1) x модуль блока питания для макетной платы
- (1) x адаптер (9 В, 1 А)

Представление компонентов

Модуль блока питания для макетной платы

Маленькому мотору постоянного тока будет необходимая большая мощность питания, чем та, которую может выдержать цифровой вывод платы UNO R3 при прямом к нему подключении. Если мы подключим мотор напрямую к плате, то существует высокая вероятность ее повреждения. Поэтому мы будем использовать модуль блока питания.



Технические характеристики:

Переключатель Вкл./Выкл. с блокировкой

Светодиодный индикатор питания

Входное напряжение: 6,5 – 9 В через штыревой соединитель (5,5 мм x 2,1 мм)

Выходное напряжение: 3,3 В/5 В

Максимальный выходной ток: 700 мА

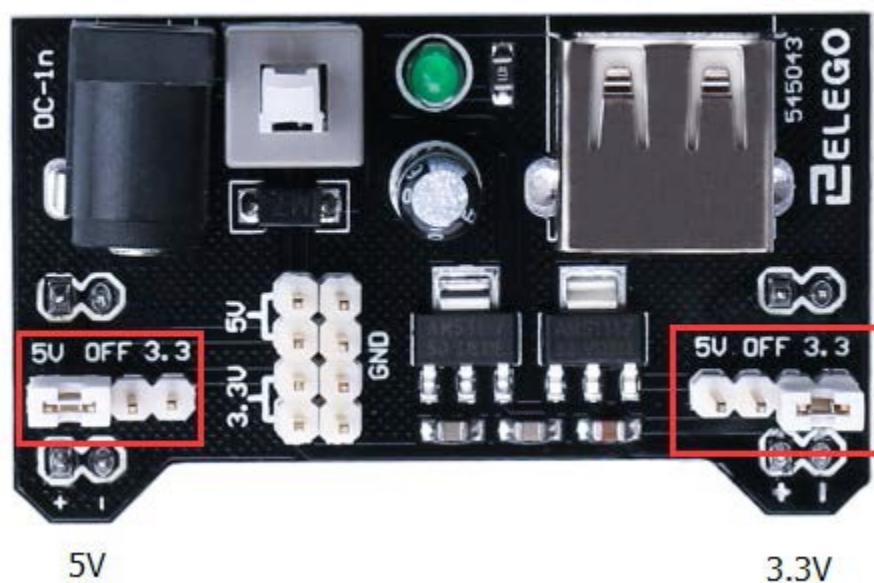
Селекторы выбора напряжения на шинах макетной платы 0 В, 3,3 В, 5 В

Дополнительные передние контакты для удобства использования

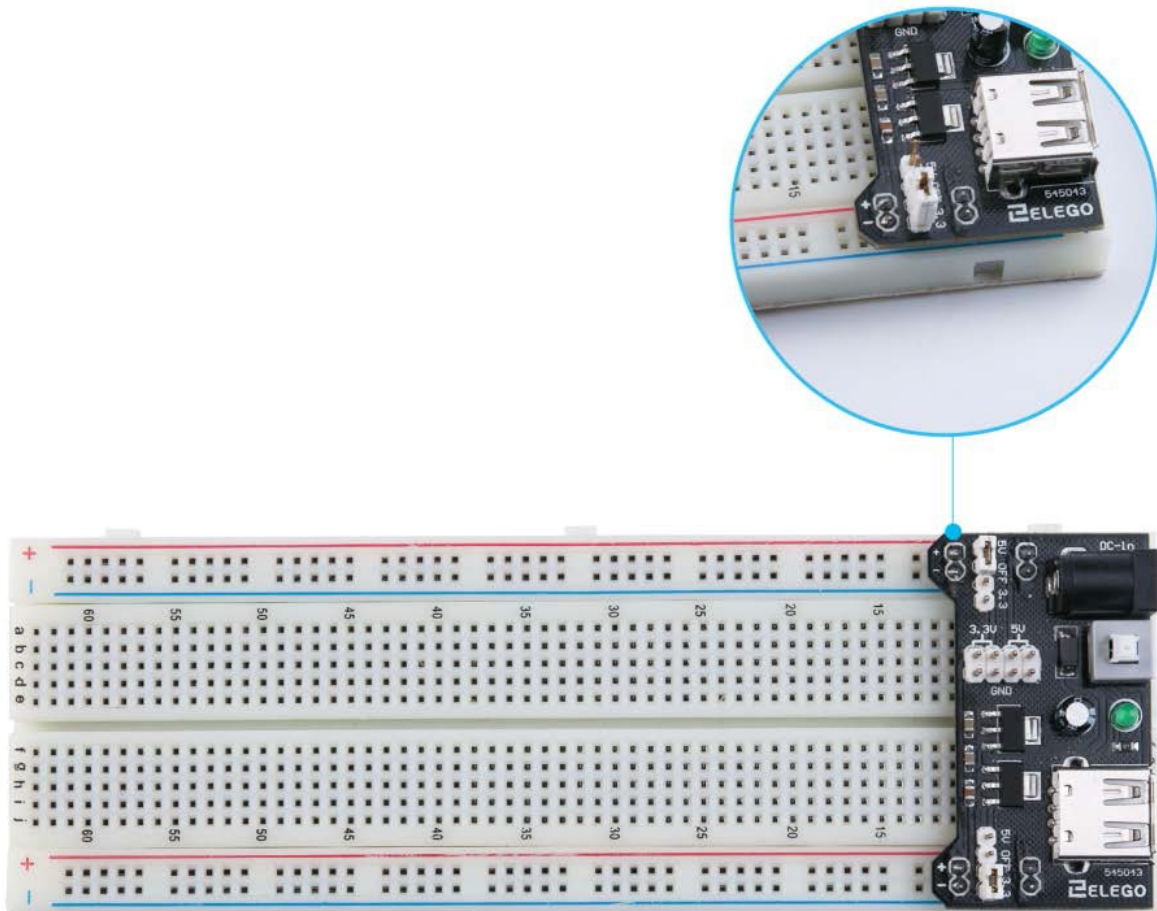
Размер: 5,5 см x 3,3 см

Выходной USB-разъем для питания внешних устройств

Настройка выходного напряжения:



Напряжение на левом и правом выходах можно настраивать по-отдельности. Выходное напряжение задается перемычками с соответствующей маркировкой. Примечание: если на обоих выходах перемычками выбрано отключение, то светодиод не будет светиться.

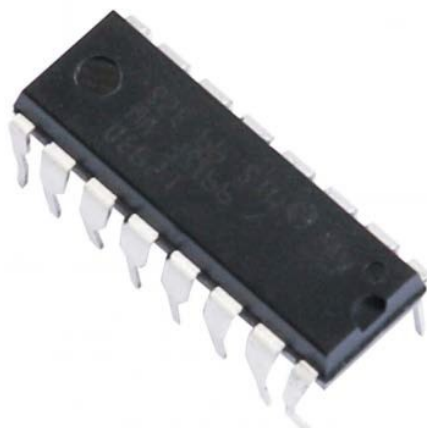


Важное примечание:

Убедитесь, что правильно установили модуль на макетную плату. На выступах платы модуля, возле контактных площадок штырей соединителей, обращенных вниз нанесена маркировка “+” и “-”. Установить надо так, чтобы знак “+” соответствовал красной полосе, а “-” синей. Так модуль питания MB102 обеспечивает одновременную подачу энергии на верхнюю и нижнюю пары шин питания макетной платы Arduino.

L293D

Это очень полезная микросхема. В общем, она позволяет управлять двумя независимыми двигателями. В этом уроке мы используем только одну сторону микросхемы, поскольку практически все контакты на правой стороне предназначены для второго мотора.



Технические характеристики:

Texas Instruments представляет две вариации микросхемы: L293 и L293D

Широкий диапазон напряжения питания: 4,5 – 36 В

Раздельное питание логической части

Защита от электростатических разрядов

Защита от перегрева

Высокая помехоустойчивость

Функционально напоминает SGS L293 и SGS L293D

Допустимый ток нагрузки – 1 А на каждый канал (600 мА для L293D)

Пиковый ток на выходе – 2 А на каждый канал (1,2 А для L293D)

Фиксирующий диод для подавления переходных процессов (L293D)



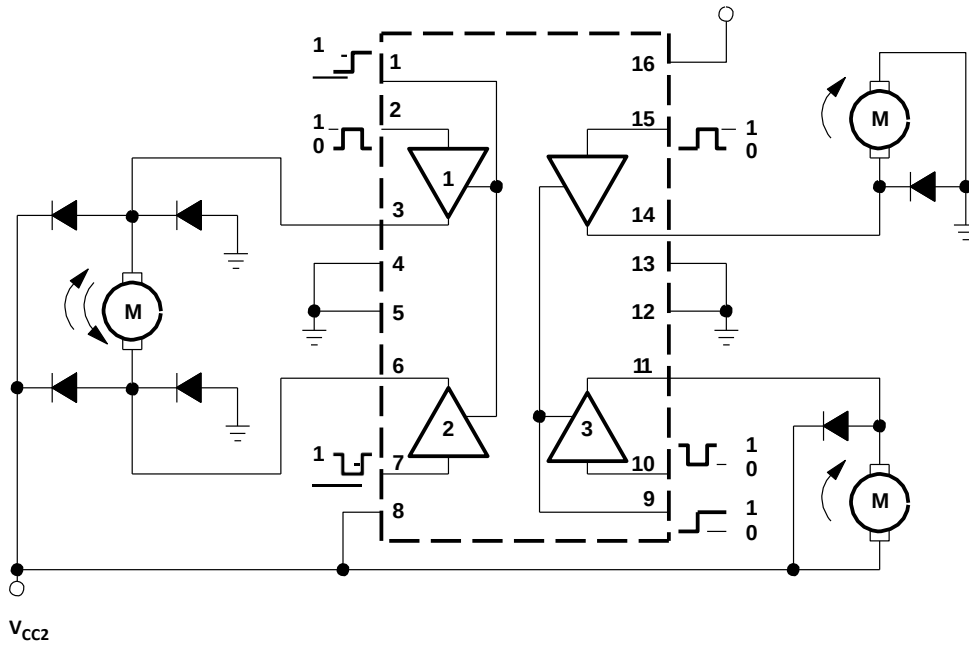
Описание

L293 и L293D – двухтактные четырехканальные микросхемы. Они могут использоваться для независимого управления двумя двигателями постоянного тока, с возможностью включения реверса. Также чип может быть использован для контроля одного биполярного шагового двигателя. Микросхема L293 предназначена для снабжения двунаправленными управляющими токами до 1 А при напряжениях от 4,5 В до 36 В. L293D предназначена для снабжения двунаправленными управляющими токами до 600 мА при напряжениях от 4,5 В до 36 В. Микросхемы совместимы по уровням TTL. Интегральная сборка основывается на нескольких составных транзисторах и, по сути, представляет собой четыре усилителя тока в общем корпусе. Выход усилителя поочередно подключается или к плюсу, или к минусу источника питания. Это

зависит от величины входного напряжения. По сути, мы имеем электронный ключ. То есть микросхему L293(D) можно определить, как четыре электронных ключа. Как видно на рисунке ниже, выходной каскад микросхемы состоит из комбинации схем Дарлингтона и Шиклаи. Верхняя часть схемы - это составной транзистор по схеме Шиклаи, а нижняя часть выполнена по схеме Дарлингтона.

Блок-схема

V_{CC1}



Мне надоели не поддающиеся расшифровке схемы, приводимые в паспортах с техническим описанием устройства, поэтому я создал свою собственную схему, предоставляющую, на мой взгляд, более существенную информацию.

Три провода подключаются к Arduino, два – к мотору и один провод подключается к источнику питания.

L293D

M1 PWM	1	16	Battery +ve
M1 direction 0/1	2	15	M2 direction 0/1
M1 +ve	3	14	M2 +ve
GND	4	13	GND
GND	5	12	GND
M1 -ve	6	11	M2 -ve
M1 direction 1/0	7	10	M2 direction 1/0
Battery +ve	8	9	M2 PWM

Motor 1

Motor 2

Для использования этой конфигурации выводов:

Левая сторона относится к первому мотору, а правая сторона – ко второму. Да, эту микросхему можно использовать только с одним подключенным мотором.

Подключения к Arduino

Подключите 'M1 PWM' к выводу Arduino с поддержкой ШИМ. На плате Uno они обозначены маркировкой. Например, пин 5. ОВыводит любое значение от 0 до 255, где 0 – это нерабочее состояние мотора, 128 – средняя скорость работы, а 255 – максимальная скорость.

Выводы 'M1 direction 0/1' и 'M1 direction 1/0' подключите к двум цифровым выводам Arduino. Если один из них имеет значение HIGH, а второй – LOW, мотор будет двигаться в одном направлении, а если наоборот – мотор начнет вращаться в противоположном направлении.

Подключение

Схема

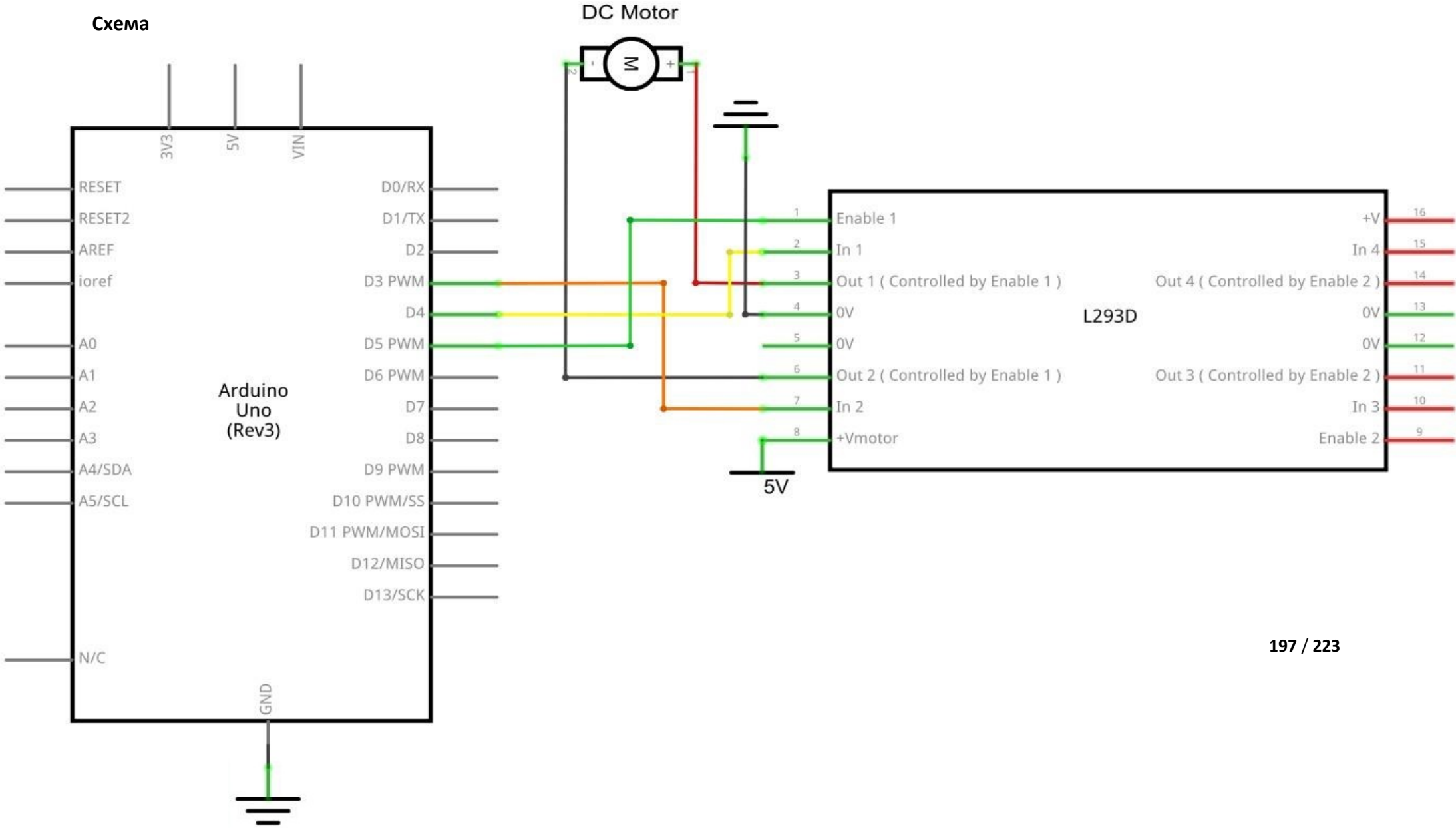
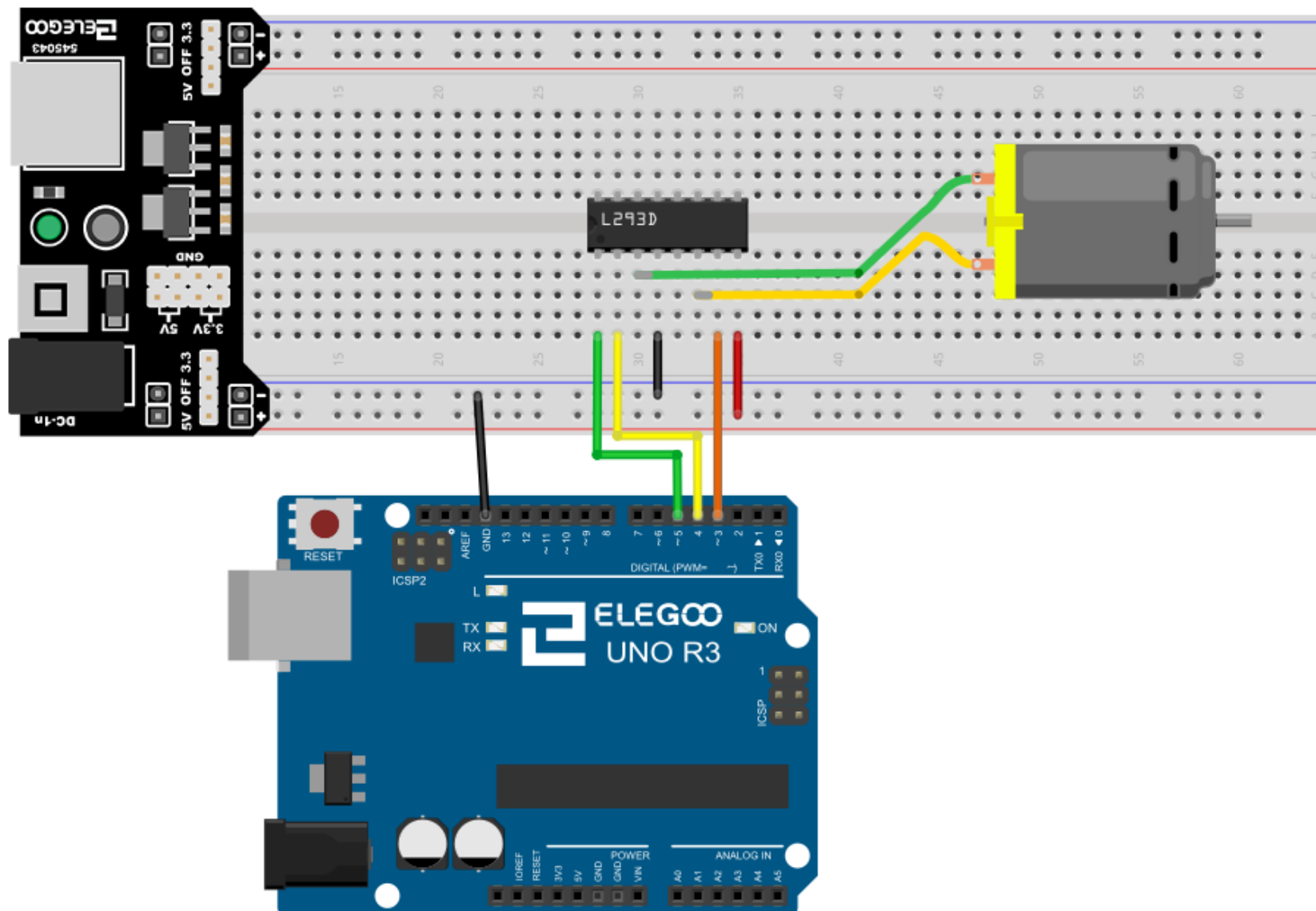
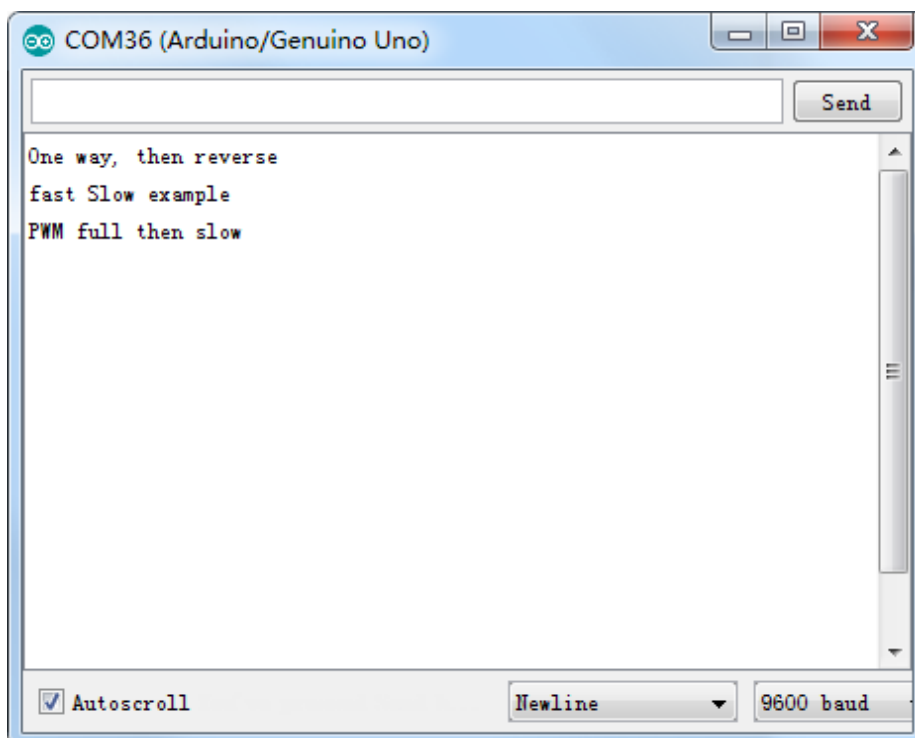


Схема монтажа



Никогда не следует напрямую подключать мотор к плате Arduino, поскольку при выключении мотора имеет место эффект обратной связи. В случае с маленьким мотором, вы повредите вашу Arduino, а в случае большого мотора – вы сможете наблюдать за интересным явлением возгорания и искрения.

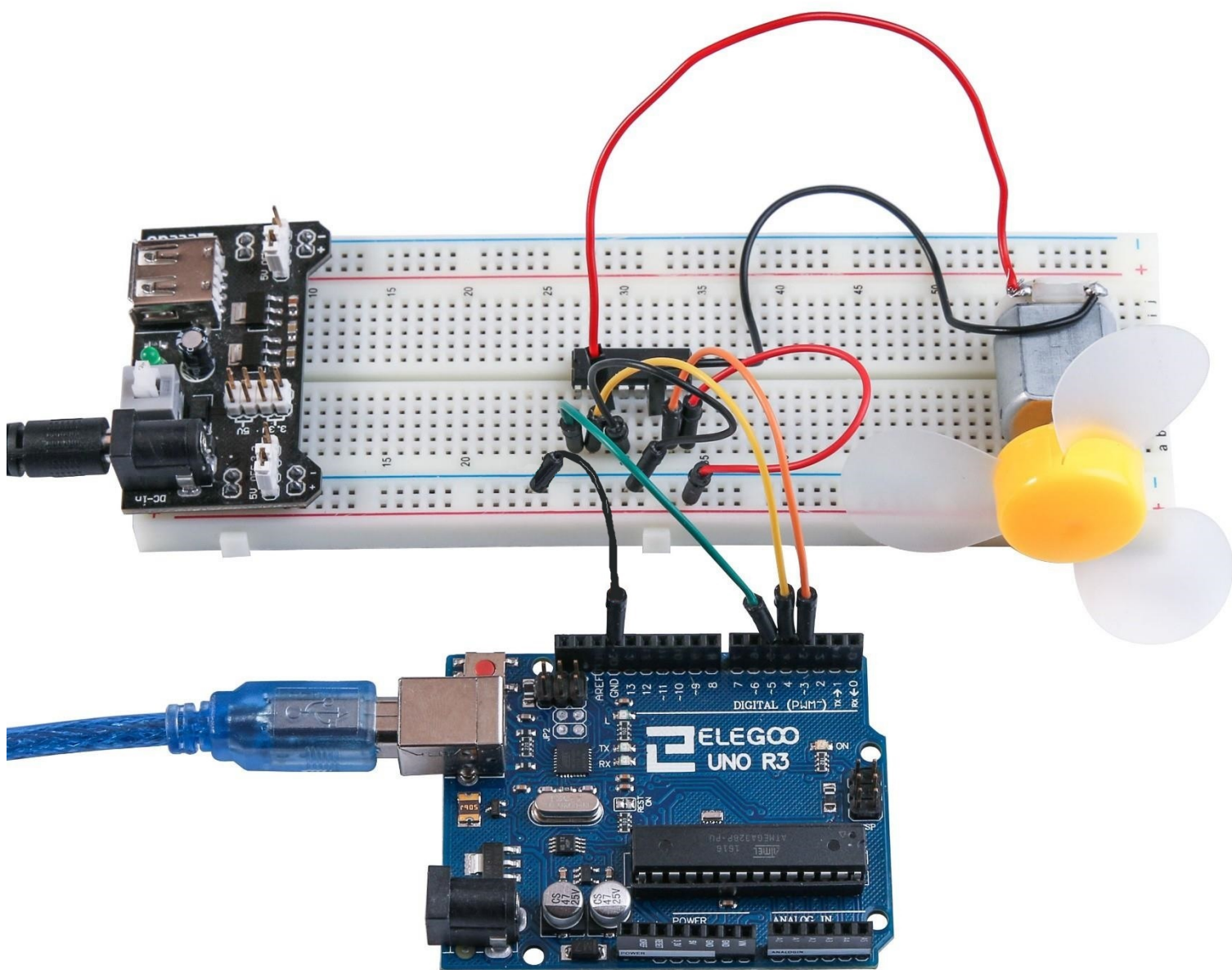


Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 21: Двигатель постоянного тока”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

После загрузки программы включите все переключатели питания. Мотор будет слегка проворачиваться по и против часового направления в течение 5 минут. Затем он начнет ускорять вращения по часовой стрелке. После короткой паузы, он начнет ускоренно вращаться против часовой стрелки. Затем контроллер отправит ШИМ-сигнал на мотор, чтобы он убавил обороты с максимальной скорости вращения до минимальной, а затем снова до максимальной. Затем следует 10-ти секундная пауза, после чего начинается следующий цикл.

Рисунок для примера



Урок 22: Реле

Обзор

В этом уроке вы научитесь использовать реле.

Необходимые условия:

- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (1) x лопастной вентилятор и мотор постоянного тока (3 – 6 В)
- (1) x микросхема L293D
- (1) x реле (5 В)
- (1) x модуль источника питания
- (1) x адаптер (9 В, 1 А)
- (8) x перемычки (типа “папа” – “папа”)



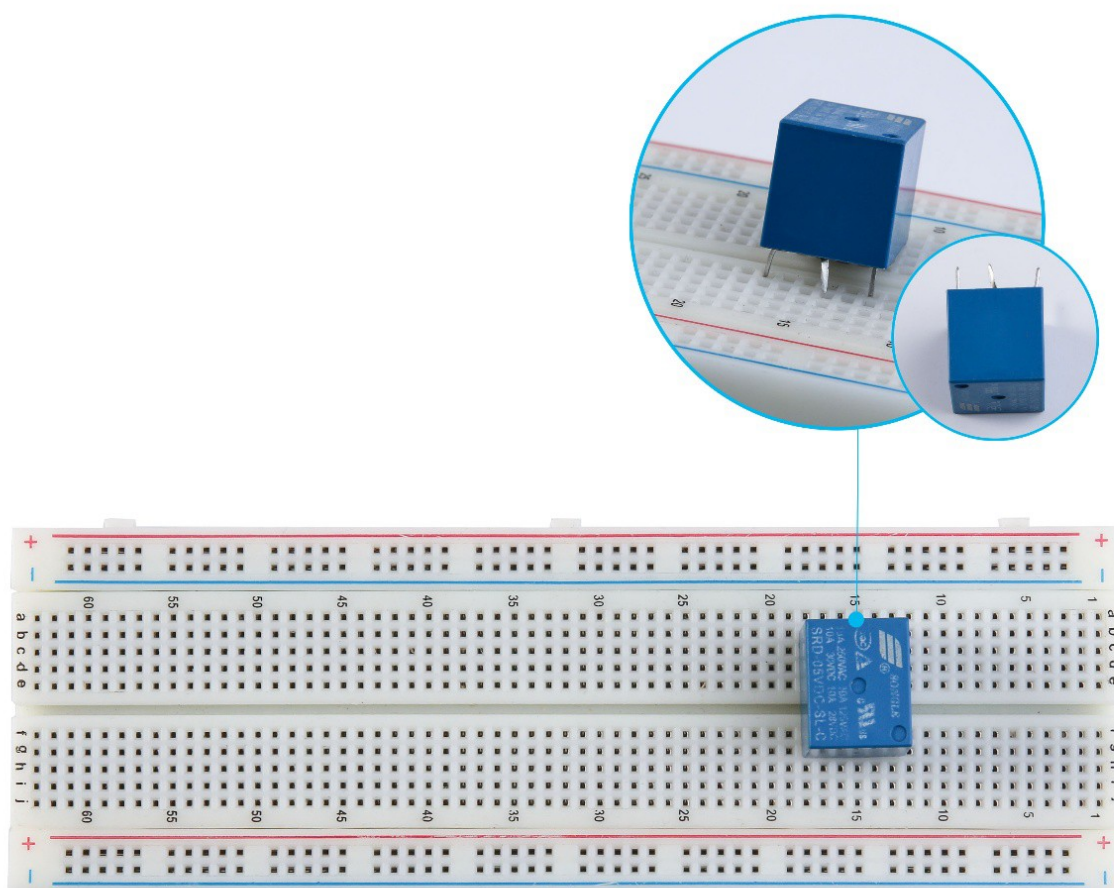
Представление компонентов

Реле:

Реле — это шлюз, который позволяет соединить вместе электрические цепи с совершенно разными параметрами. Реле включает или выключает внешние устройства, определенным образом замыкая или размыкая отдельную электрическую сеть, в которую они подключены. Реле можно представить себе в виде дистанционного пульта, с помощью которого мы выполняем нужные действия с помощью относительно «слабых» сигналов. В зависимости от типа этих внутренних размыкающих механизмов и особенностях устройства можно выделить две основные группы реле: электромеханические реле и твердотельные реле.

Электромагнитное реле — это электрическое устройство, которое механическим путем замыкает или размыкает цепь нагрузки при помощи магнита. состоит из электромагнита, подвижного якоря и переключателя. Твердотельные реле считаются хорошей альтернативой электромагнитным, они представляет собой модульное полупроводниковое устройство, которое производится по гибридной технологии. В составе реле имеются транзисторы, симисторы или тиристоры. Ниже показано подключение реле к плате Uno R3.

Вас может озадачить вопрос о том, как вставить реле в макетную плату. Как показано на



Схема

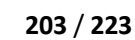
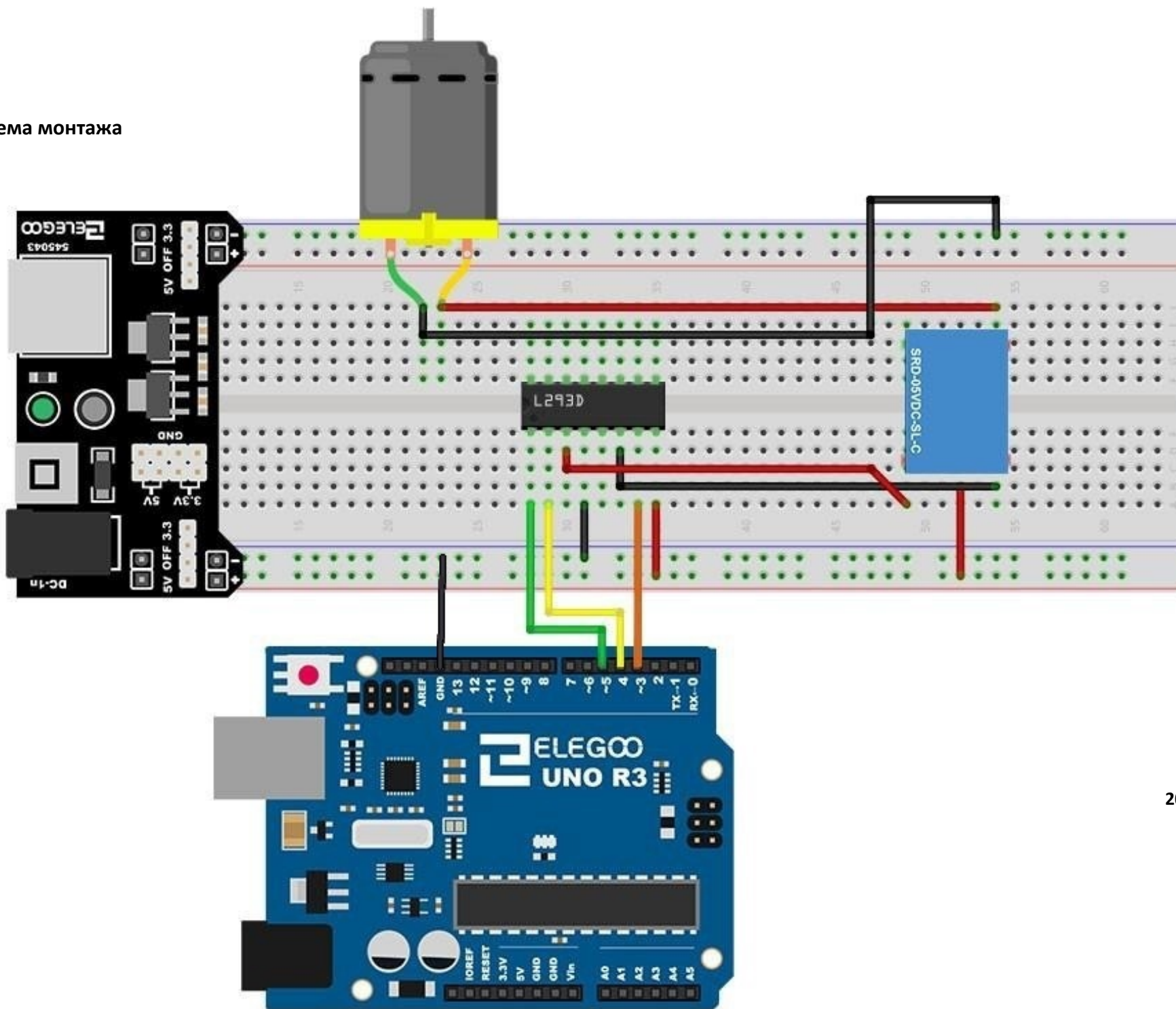


Схема монтажа

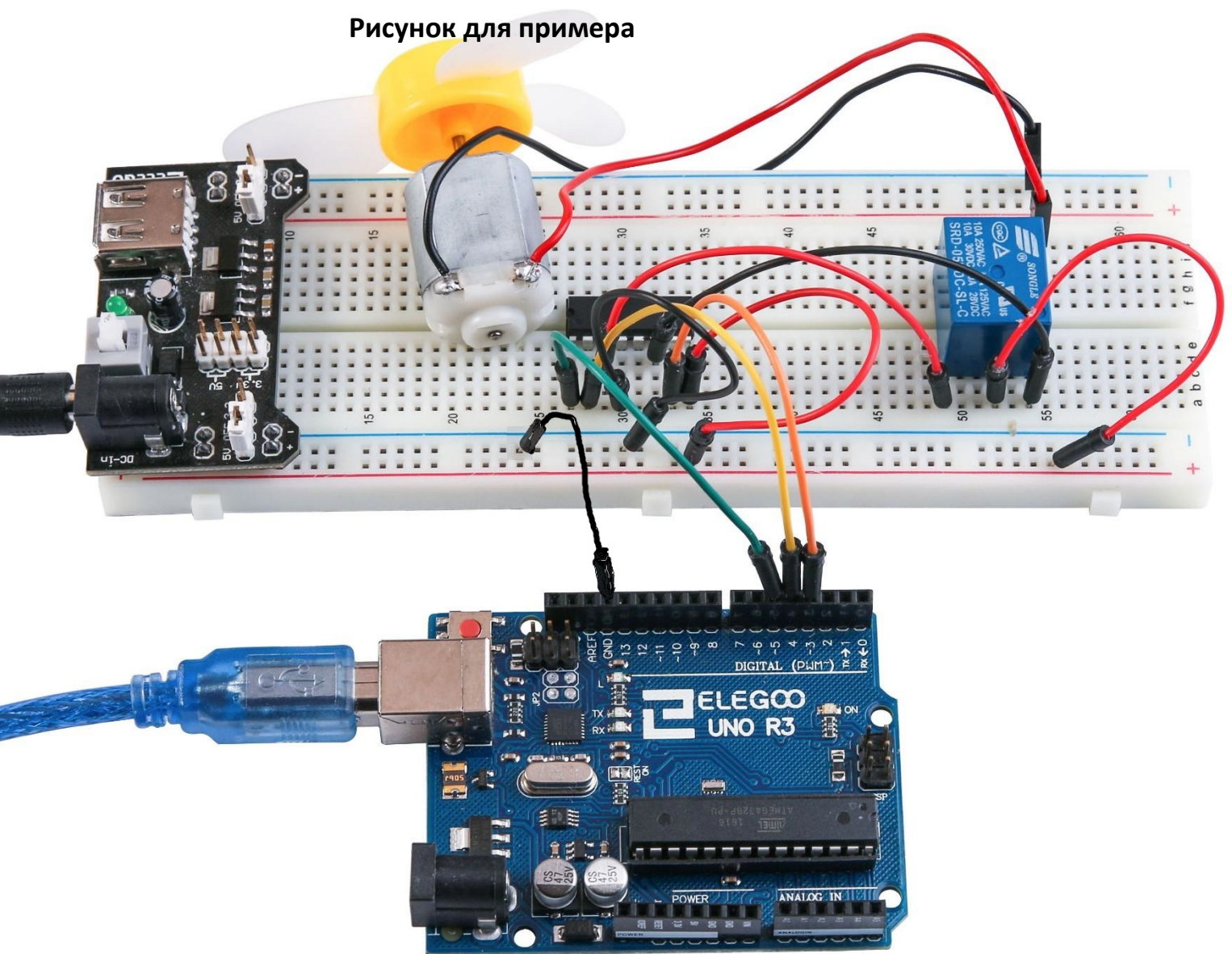


Code

After wiring, please open the program in the code folder- Lesson 30 Relay and click **UPLOAD** to upload the program. See Lesson 2 for details about program uploading if there are any errors.

After program loading, turn on all the power switches. The relay will pick up with a ringing sound. Then, the motor will rotate. After a period of time, the relay will be released, and the motor stops.

Рисунок для примера



Урок 23: Шаговый двигатель

Обзор

В этом уроке вы научитесь простому и занятному способу управления шагового двигателя.

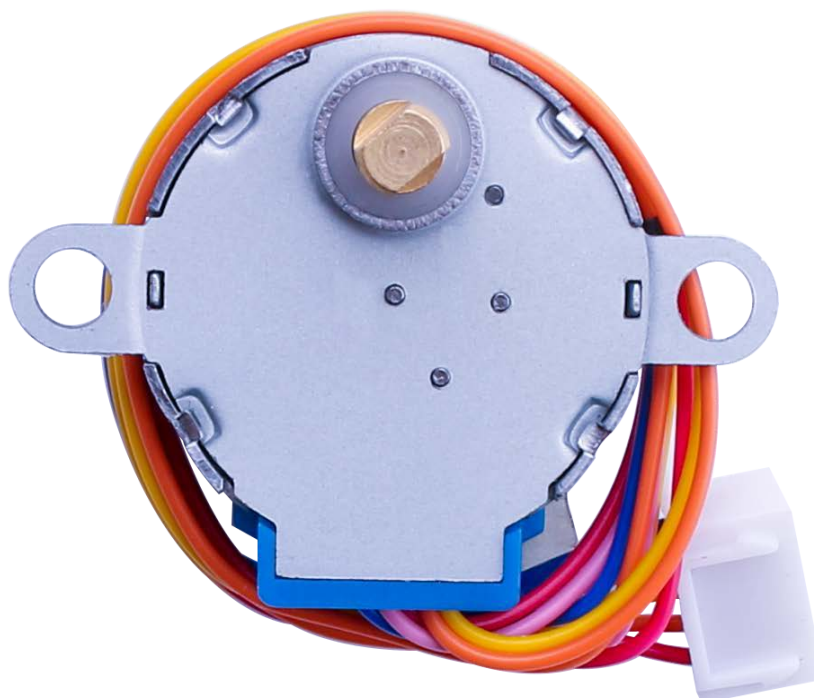
Шаговый двигатель, который мы будем использовать, поставляется со своим собственным драйвером, что облегчает его подключение к плате UNO.

Необходимые компоненты:

- (1) х плата Elegoo Uno R3
- (1) х макетная плата на 830 точек
- (1) х модуль драйвера для шагового двигателя ULN2003
- (1) х шаговый двигатель
- (1) х адаптер (9 В, 1 А)
- (1) х модуль блока питания для макетной платы
- (6) х перемычки DuPont (типа “мама” – “папа”)
- (1) х перемычки (типа “папа” – “папа”)

Представление компонентов

Шаговый двигатель



Шаговый двигатель – это мотор, перемещающий свой вал в зависимости от заданных в программе микроконтроллера шагов и направления. Подобные устройства чаще всего используются в робототехнике, принтерах, манипуляторах, различных станках и прочих электронных приборах. Большим преимуществом шаговых двигателей над двигателями постоянного вращения является обеспечение точного углового позиционирования ротора. Также в шаговых двигателях имеется возможность быстрого старта, остановки, реверса. Шаговый двигатель обеспечивает вращения ротора на заданный угол при соответствующем управляющем сигнале. Благодаря этому можно контролировать положение узлов механизмов и выходить в заданную позицию. Работа двигателя осуществляется следующим образом – в центральном вале имеется ряд магнитов и несколько катушек. При подаче питания создается магнитное поле, которое воздействует на магниты и заставляет вал вращаться. Такие параметры как угол поворота (шаги), направление движения задаются в программе для микроконтроллера.

Характеристики шагового двигателя 28BYJ-48

Модель: 28BYJ-48

Напряжение питания: 5 В (постоянный ток)

Количество фаз: 4

Количество шагов (на оборот): 64

Угол поворота: $5,625^\circ$

Частота: 100 Гц

Сопротивление постоянного тока: $50 \text{ Ом} \pm 7\%$ (25°C)

Частота холостого хода по часовой стрелке: $> 600 \text{ Гц}$

Частота холостого хода против часовой стрелки: $> 1000 \text{ Гц}$

Крутящий момент $> 34,3 \text{ мН*м}$ (120 Гц)

Момент самопозиционирования $> 34,3 \text{ мН*м}$

Стопорящий момент: $600\text{-}1200 \text{ г*см}$

Тяга: 300 г*см

Сопротивление изоляции $> 10 \text{ МОм}$ (500 В)

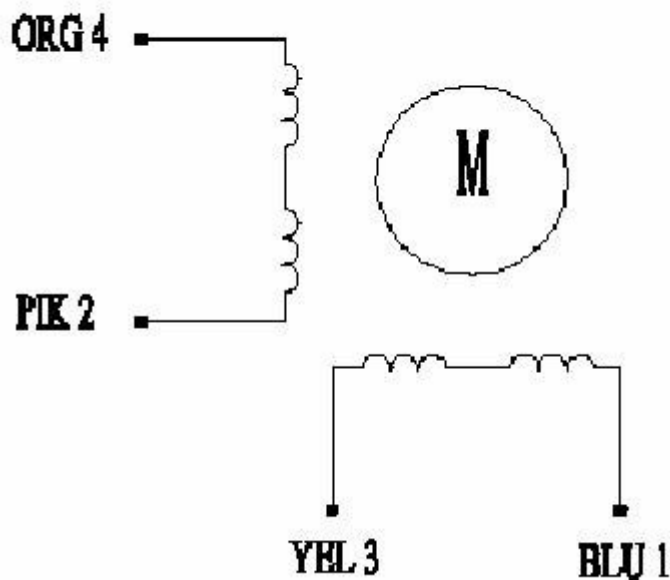
Класс электробезопасности: А

Температурный режим $< 40 \text{ К}$ (120 Гц)

Шум $< 35 \text{ дБ}$ (120 Гц, без нагрузки, 10 см)

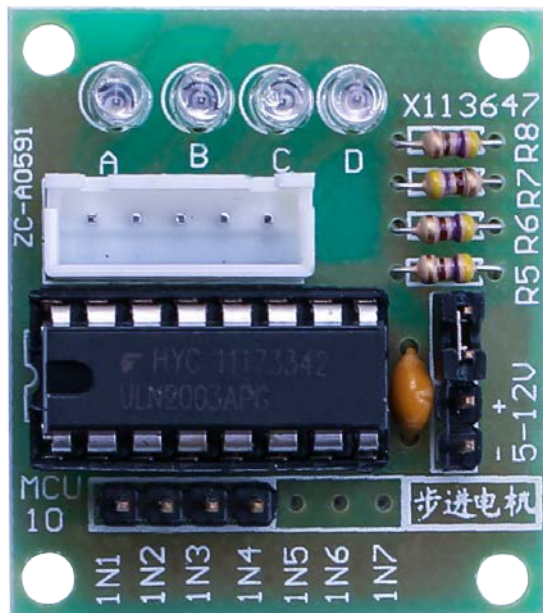
Схема интерфейса

WIRING DIAGRAM



Биполярные шаговые двигатели обычно имеют четыре контакта для подключения. В отличие от униполярных двигателей, у биполярных шаговых двигателей нет общего центрального вывода. Вместо этого они содержат два отдельных вывода на фазу, каждый со своей обмоткой. Униполярные шаговые двигатели имеют пять или шесть контактов для подключения и четыре электромагнитные катушки в корпусе (если быть более точными, то две катушки, разделенные на четыре). Центральные контакты катушек соединены вместе и используются для подачи питания на двигатель. Эти шаговые моторы называются униполярными, потому что питание всегда подается на один из этих полюсов.

Плата драйвера ULN2003



Технические характеристики

Размер: 42 мм x30 мм

Микросхема ULN2003, 500 мА

Светодиоды (А, В, С, D) для индикации четырех фаз работы двигателя

Белый пятиконтактный разъем – стандартный разъем для подключения к шаговому двигателю

Выводы питания разделены

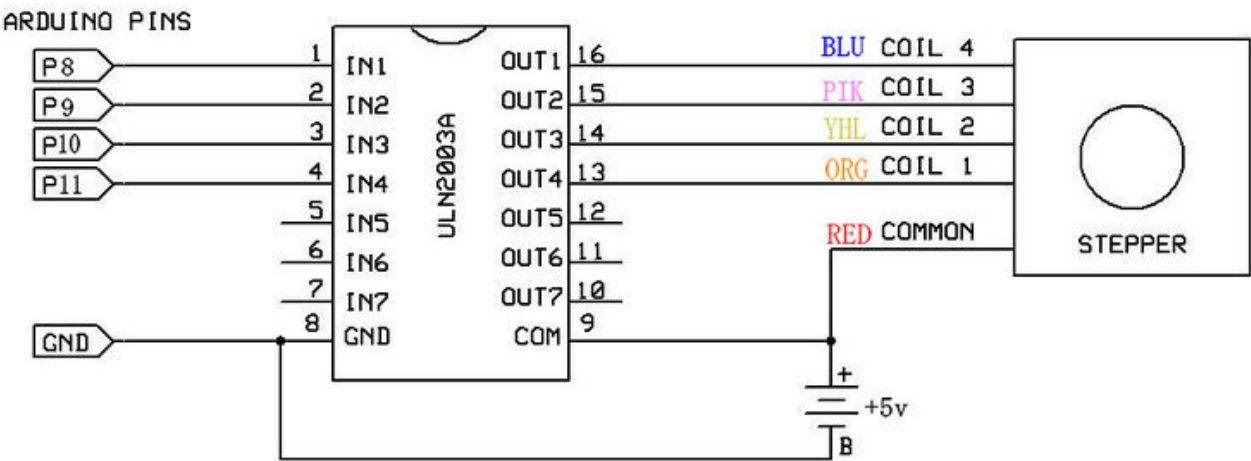
Оставшиеся выводы мы оставили для ваших будущих проектов.

Плата драйвера шагового двигателя на базе ULN2003, представляющей собой массив транзисторов, включенных по схеме Дарлингтона, позволяет достаточно просто управлять мотором, используя микроконтроллер. В нашем случае, в качестве управляющего микроконтроллера мы выберем микроконтроллер нашей платы Uno. Помимо самой микросхемы ULN2003, на плате имеется пятиконтактный разъем для подключения к двигателю и четыре светодиода, показывающих, какая из

обмоток запитана в текущий момент времени. Для управления шаговым двигателем запитывать обмотки необходимо в определенной последовательности, показанной в таблице ниже.

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Ниже приведена схема, показывающая подключение униполярного шагового двигателя к четырем выводам контроллера ULN2003A.



Подключение

Схема

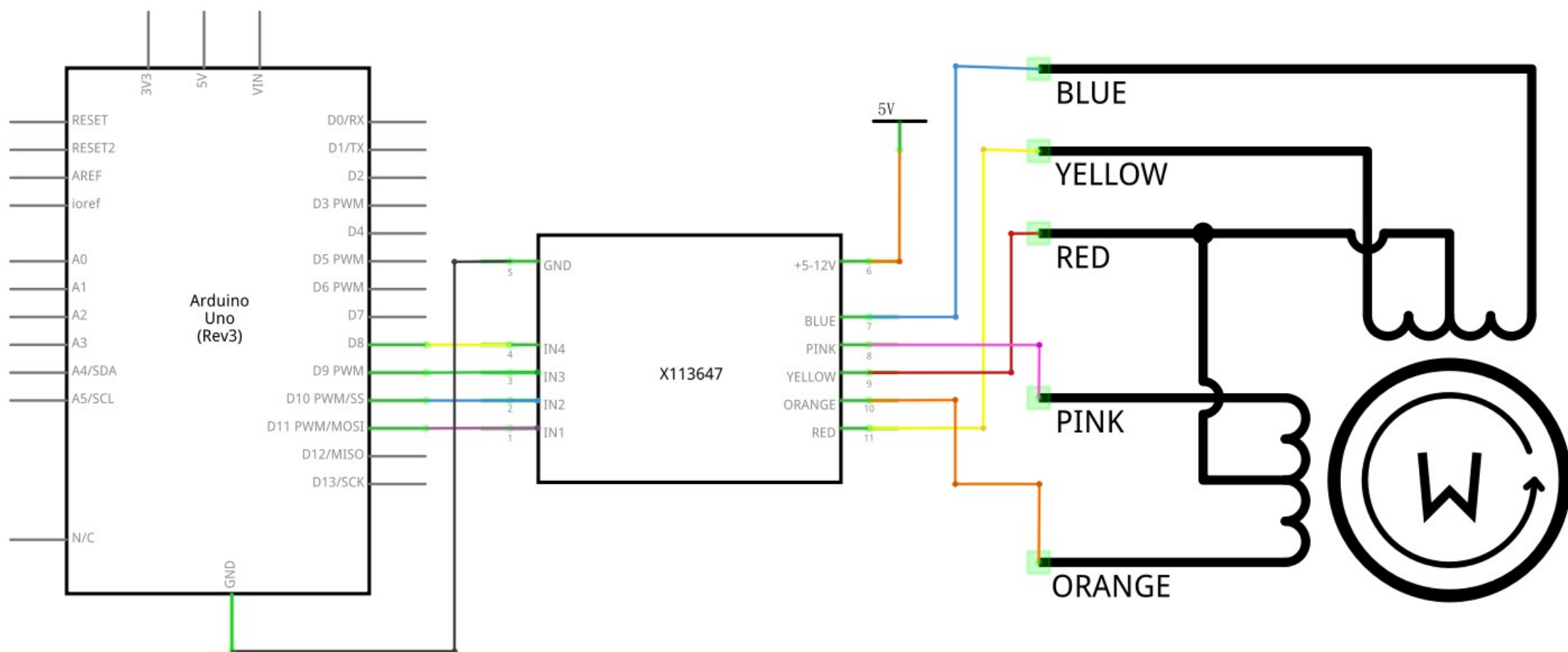
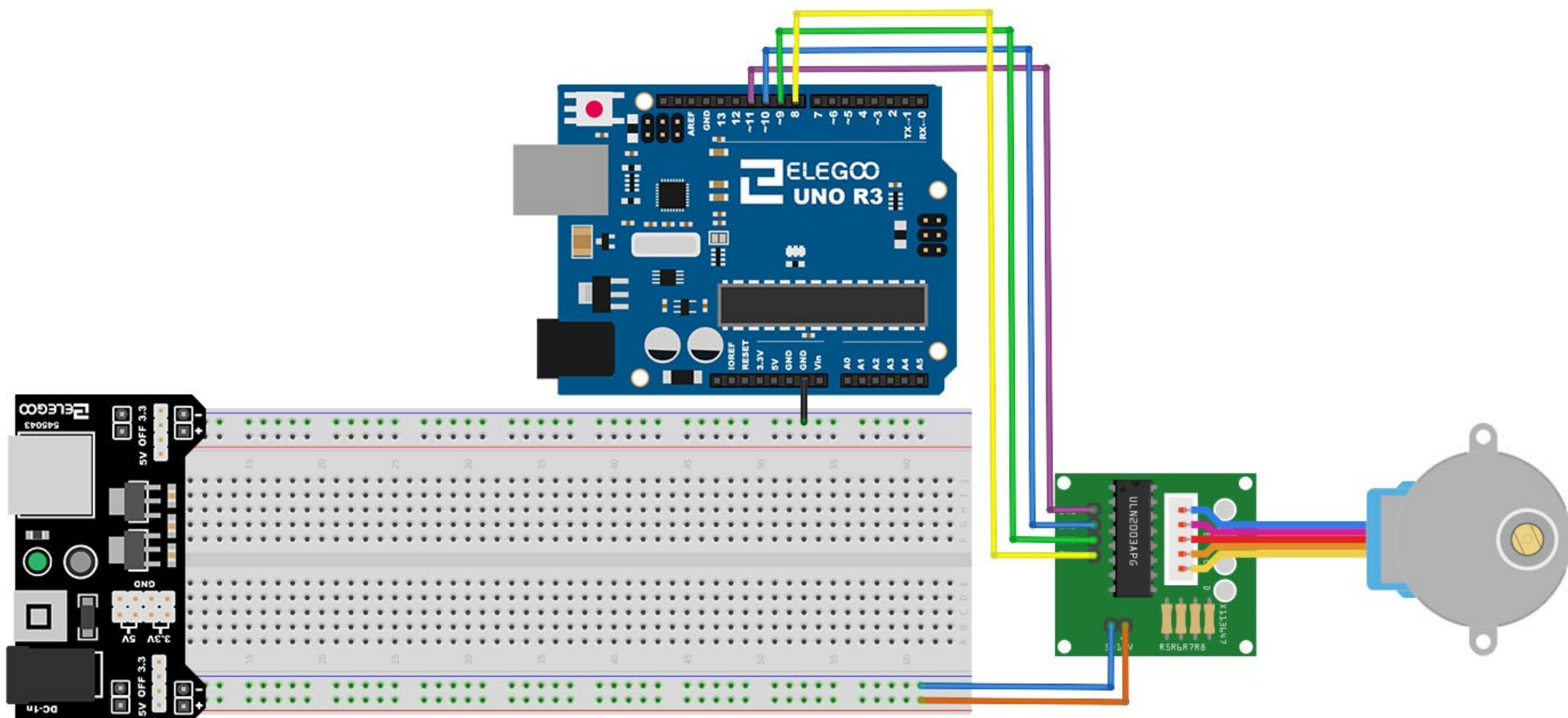


Схема монтажа



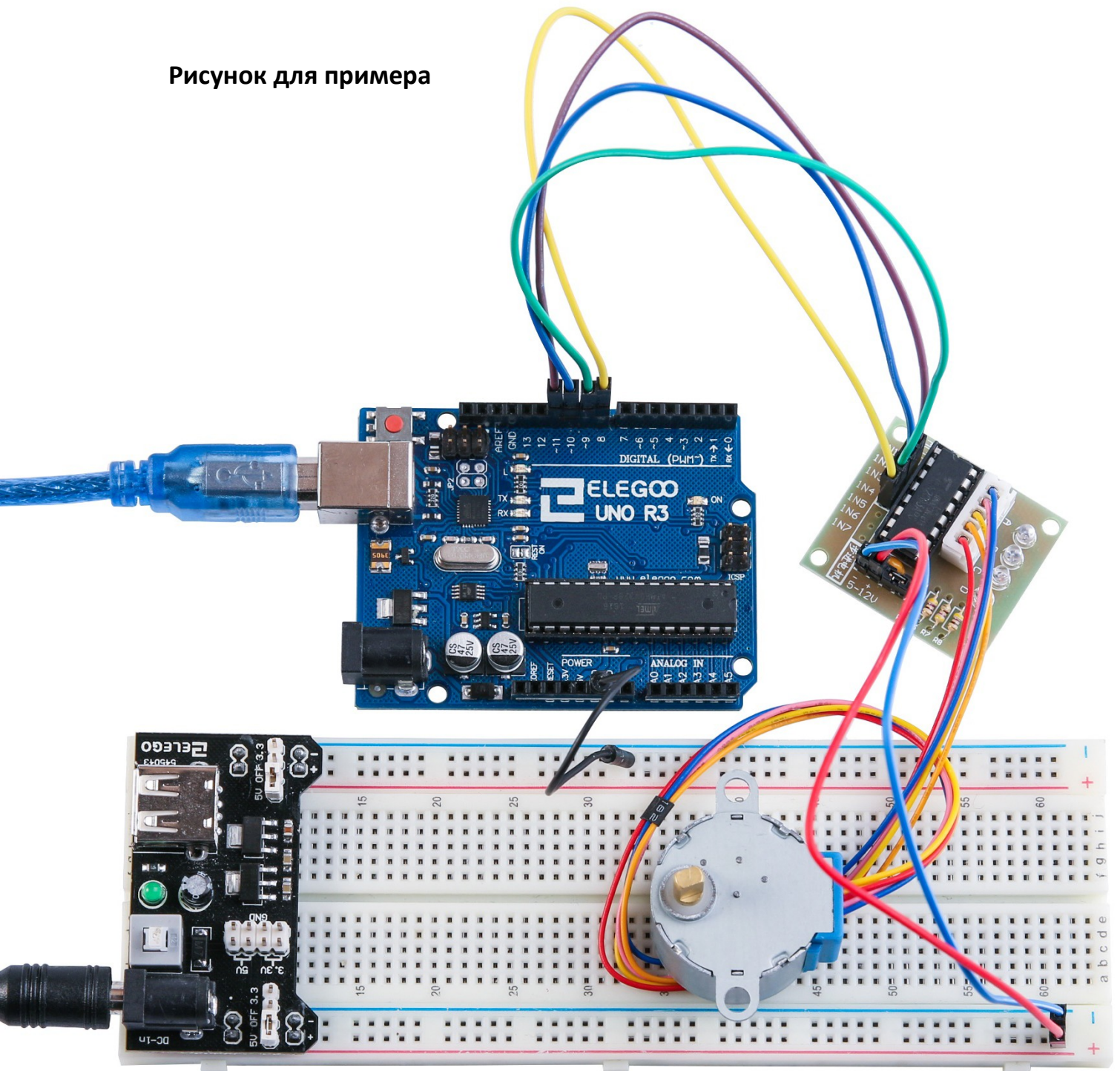
Для управления шаговым двигателем мы используем четыре вывода. Выводы 8-11 на плате UNO отвечают за управление мотором.

Если для питания UNO и мотора используются различные источники питания, то необходимо объединить выводы 'Ground' источников вместе.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 31: Шаговый двигатель”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры. Перед запуском, убедитесь, что вы установили библиотеку < Stepper >. Переустановите, если необходимо, иначе, ваш код не будет работать. Процедура добавления библиотек описана в Уроке 1.

Рисунок для примера



Урок 24: Управление шаговым двигателем с помощью пульта

Обзор

В этом уроке вы научитесь простому и интересному способу управления шаговым двигателем с помощью ИК-передатчика и пульта дистанционного управления. Шаговый двигатель, который мы будем использовать, поставляется со своим собственным драйвером, что облегчает его подключение к плате UNO.

Поскольку мы не хотим подавать питание на наш двигатель напрямую с платы UNO, мы будем использовать недорогой маленький модуль источника питания для макетной платы (9 В, 1 А).

ИК-передатчик подключается к UNO напрямую, поскольку он практически не потребляет энергии.

Необходимые компоненты:

- (1) x плата Elegoo Uno R3
- (1) x макетная плата на 830 точек
- (1) x модуль ИК-передатчика
- (1) x ИК-пульт дистанционного управления
- (1) x модуль управления шаговым двигателем ULN2003
- (1) x шаговый двигатель
- (1) x модуль источника питания
- (1) x адаптер (9 В, 1 А)
- (9) x перемычки DuPont (типа “мама” – “папа”)
- (1) x перемычки (типа “папа” – “папа”)

Подключение

Схема

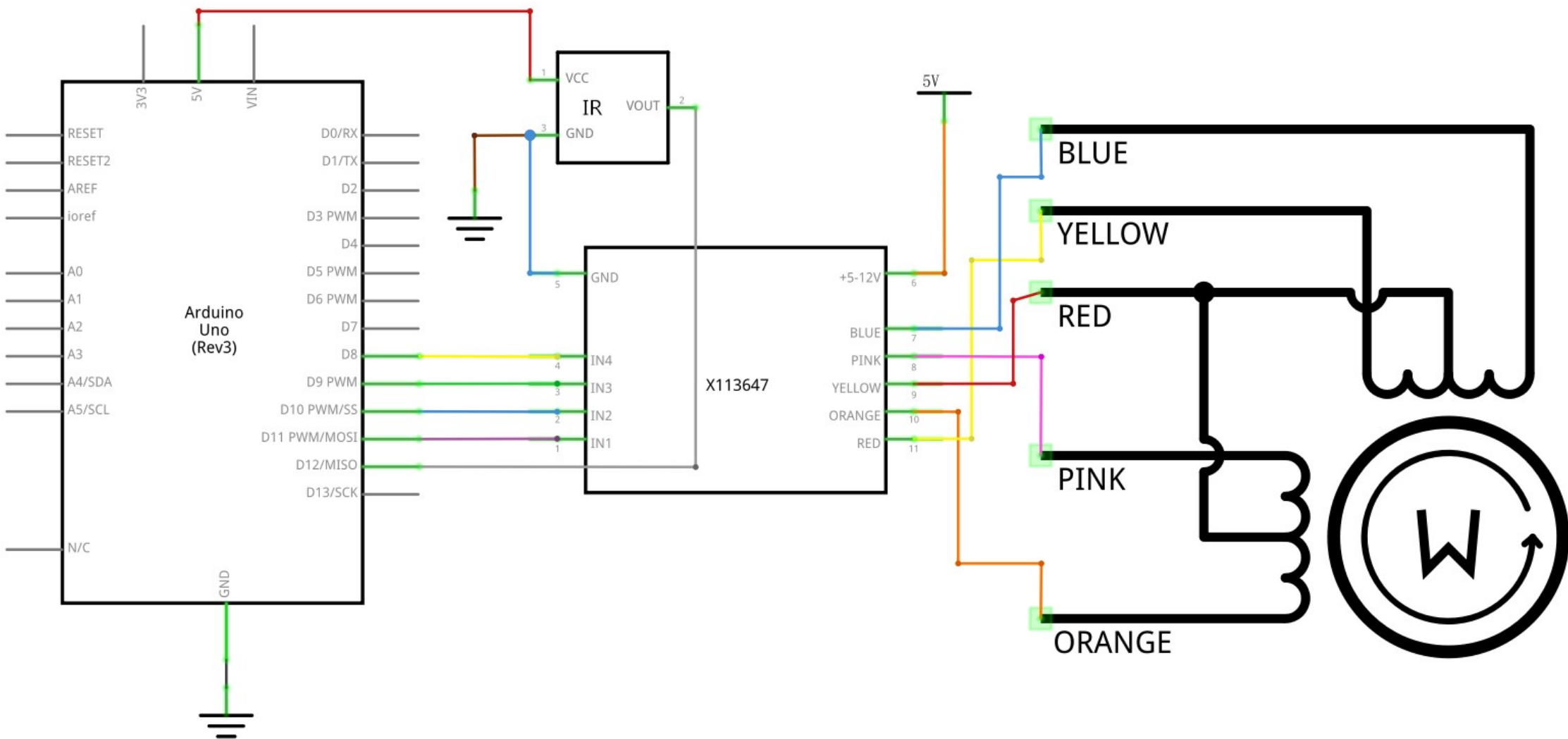
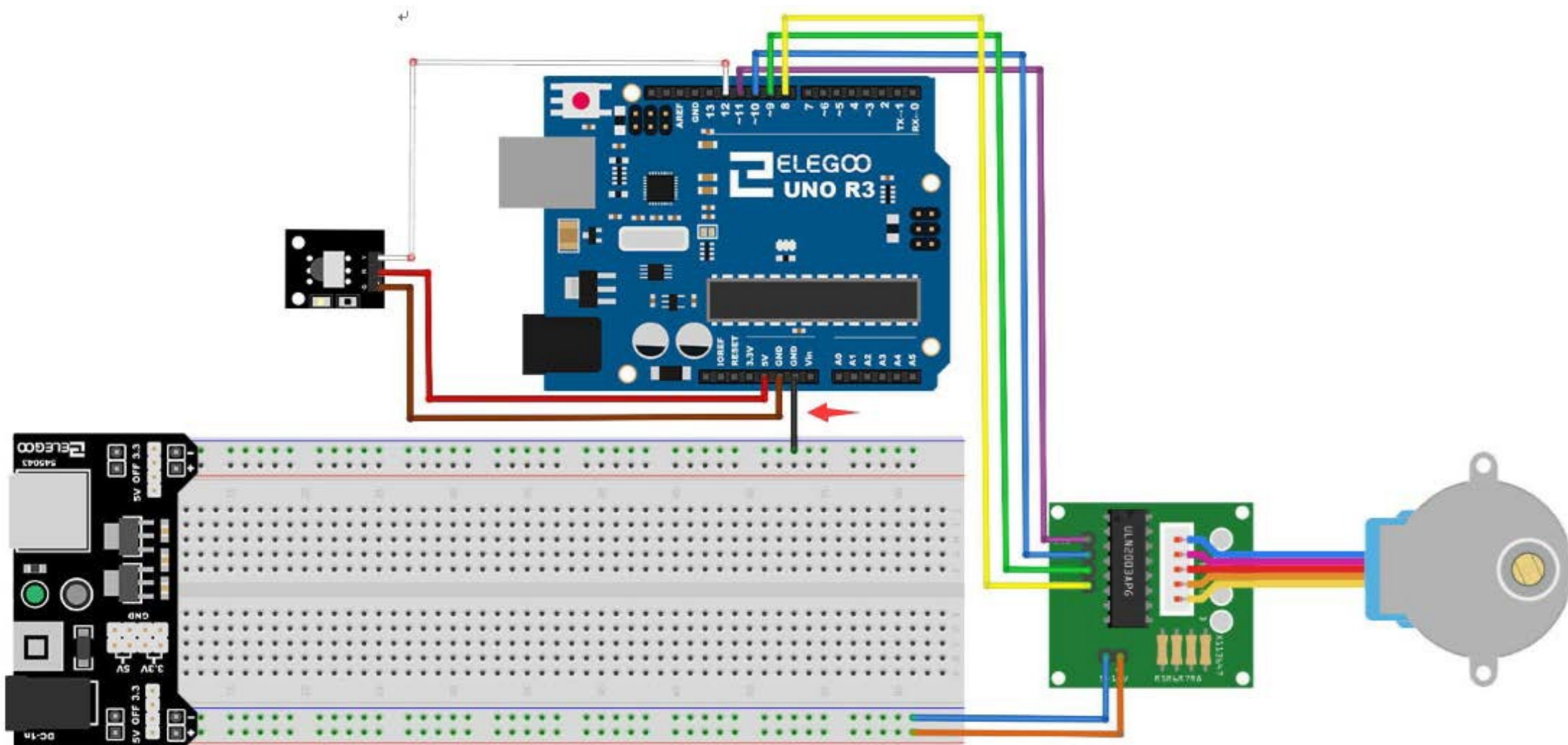


Схема монтажа



Мы используем четыре вывода для управления шаговым двигателем и один вывод для ИК-передатчика.

Выводы 8-11 управляют шаговым двигателем, а вывод 12 получает информацию с ИК-передатчика.

Мы подключаем выводы +5 В и Ground платы UNO к датчику. В качестве предосторожности, используйте источник питания макетной платы для шагового двигателя, поскольку мы не хотим повредить нашу плату UNO.

Код

После монтажа, пожалуйста, откройте программу в папке с кодом – “Урок 32: Управление шаговым двигателем с помощью пульта”, и нажмите UPLOAD для загрузки программы. Обратитесь к Уроку 2 для детального объяснения данной процедуры.

Перед запуском, убедитесь, что вы установили библиотеку < IRremote >. Переустановите, если необходимо, иначе, ваш код не будет работать.

Процедура добавления библиотек описана в Уроке 1.

Код распознает только два значения с ИК-передатчика: VOL+ и VOL-. Когда на пульте зажата VOL+, двигатель начнет совершать полные обороты по часовой стрелке. Кнопка VOL- вызывает вращения в противоположную сторону.

Рисунок для примера

