THE MOST COMPLETE STARTER KIT

TUTORIAL FOR MEGA 2560

V1.0.18.12.19

Preface

Our Company

Established in 2011, Elegoo Inc. is a thriving technology company dedicated to open-source hardware research & development, production and marketing. Located in Shenzhen, the Silicon Valley of China, we have grown to over 150+ employees with a 10,763+ square ft. factory.

Our product lines rang from DuPont wires, 2560 R3 boards to complete starter kits designed for customers of any level to learn Arduino knowledge. In addition, we also sell products of Raspberry Pi accessories like 2.8'' TFT touch and STM32. In the future we would devote more energy and investment to 3D printer products and so on. All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.

Official website:http://www.elegoo.com

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

Our Tutorial

This tutorial is designed for beginners. You will learn all the basic information about how to use Arduino controller board, sensors and components. If you want to study Arduino in more depth, we recommend that you read the Arduino Cookbook written by Michael Margolis.

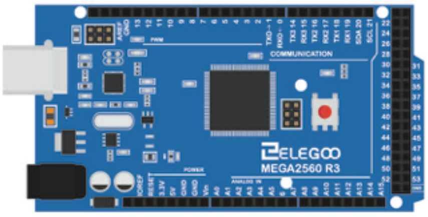
Some codes in this tutorial is edited by Simon Monk. Simon Monk is author of a number of books relating to Open Source Hardware. They are available in Amazon: Programming Arduino, 30 Arduino Projects for the Evil Genius and Programming the Raspberry Pi.

Customer Service

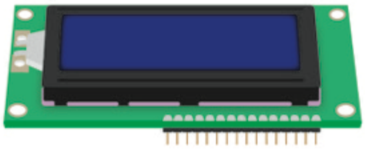
As a continuous and fast growing technology company we keep striving our best to offer you excellent products and quality service as to meet your expectation and you can reach out to us by simply drop a line at service@elegoo.com or EUservice@elegoo.com. We look forward to hearing from you and any of your critical comment or suggestion would be much valuable tous.

And any of problems and questions you have with our products will be promptly replied by our experienced engineers within 12 hours (24hrs during holiday)

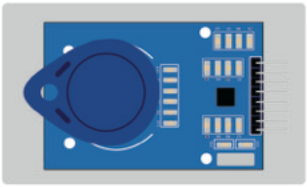
Packing list



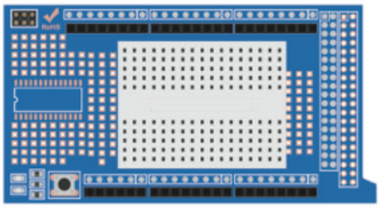
MEGA 2560 Controller Board
1PC



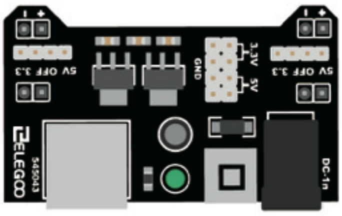
LCD1602 Module
(with pin header)
1PC



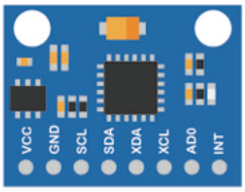
RC522 RFID Module
1PC



Prototype Expansion Module
1PC

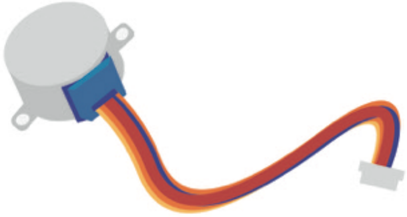


Power Supply Module
1PC

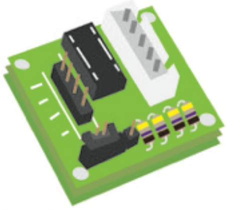


GY-521 Module
1PC

Servo Motor SG90
1PC

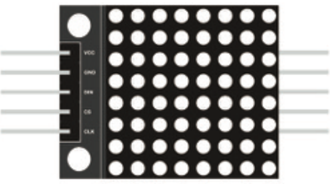


Stepper Motor
1PC



ULN2003 Stepper Motor
Driver Module
1PC

Remote Control
1PC



MAX7219 Module
1PC

1 Digit 7-Segment Display
1PC

4 Digit 7-Segment Display
1PC

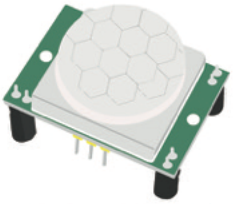
L293D
1PC

74HC595 IC
1PC

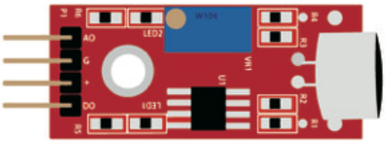
Active Buzzer
1PC

Passive Buzzer
1PC

Potentiometer 10K
2PCS



HC-SR501 PIR Motion Sensor Module
1PC

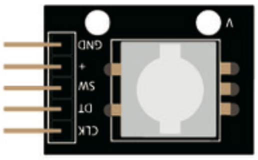


Sound Sensor Module
1PC

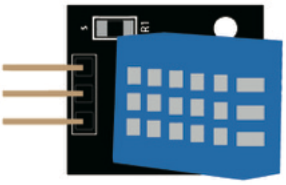
Water Level Detection Sensor Module
1PC

Ultrasonic Sensor
1PC

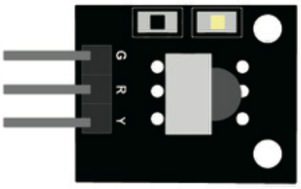
DS1307 RTC Module
1PC



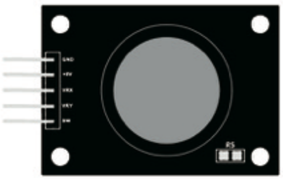
Rotary Encoder Module
1PC



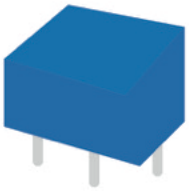
DHT11 Temperature and Humidity Module
1PC



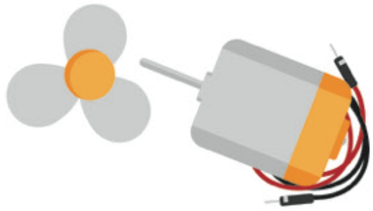
IR Receiver Module
1PC



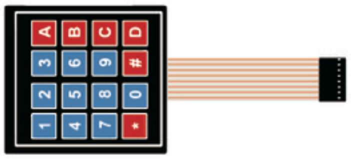
Joystick Module
1PC



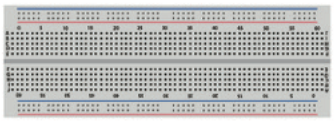
5V Relay
1PC



Fan Blade and 3-6V Motor
1PC

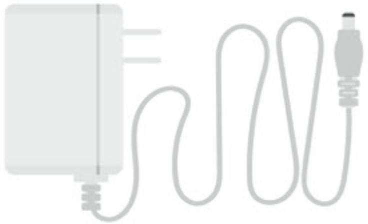


Membrane Switch Module
1PC



830 Tie-Points Breadboard
1PC

9V Battery with Snap-on Connector Clip
1PC

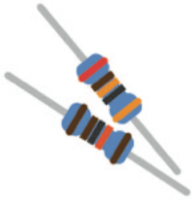


9V1A Adapter
1PC

Breadboard Jumper Wire
65PCS

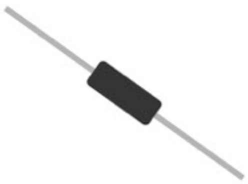
Female-to-Male Dupont Wire
20PCS

USB Cable
1PC

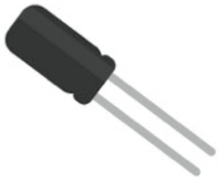


Resistor
120PCS

Thermistor
1PC

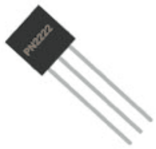


Diode Rectifier
5PCS

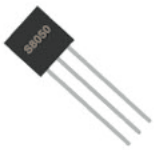


100uF Electrolytic Capacitor
2PCS

10uF Electrolytic Capacitor
2PCS

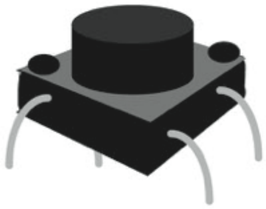


NPN Transistor PN2222
5PCS



NPN Transistor S8050
5PCS

Tilt Ball Switch
1PC



Button
5PCS

Red LED
5PCS

Yellow LED
5PCS

Blue LED
5PCS

Green LED
5PCS

White LED
5PCS

RGB LED
2PCS

104pF Ceramic Capacitor
5PCS

22pF Ceramic Capacitor
5PCS

Photoresistor(Photocell)
2PCS

Content

Lesson 4 RGB LED

Overview

RGB LEDs are a fun and easy way to add some color to your projects. Since they are like 3 regular LEDs in one, how to use and connect them is not much different.

They come mostly in 2 versions: Common Anode or Common Cathode.

Common Anode uses 5V on the common pin, while Common Cathode connects to ground.

As with any LED, we need to connect some resistors inline (3 total) so we can limit the current being drawn.

In our sketch, we will start with the LED in the Red color state, then fade to Green, then fade to Blue and finally back to the Red color. By doing this we will cycle through most of the color that can be achieved.

Component Required:

(1) x Elegoo Mega 2560 R3

(1) x 830 Tie Points Breadboard

(4) x M-M wires (Male to Male jumper wires)

(1) x RGB LED

(3) x 220 ohm resistors

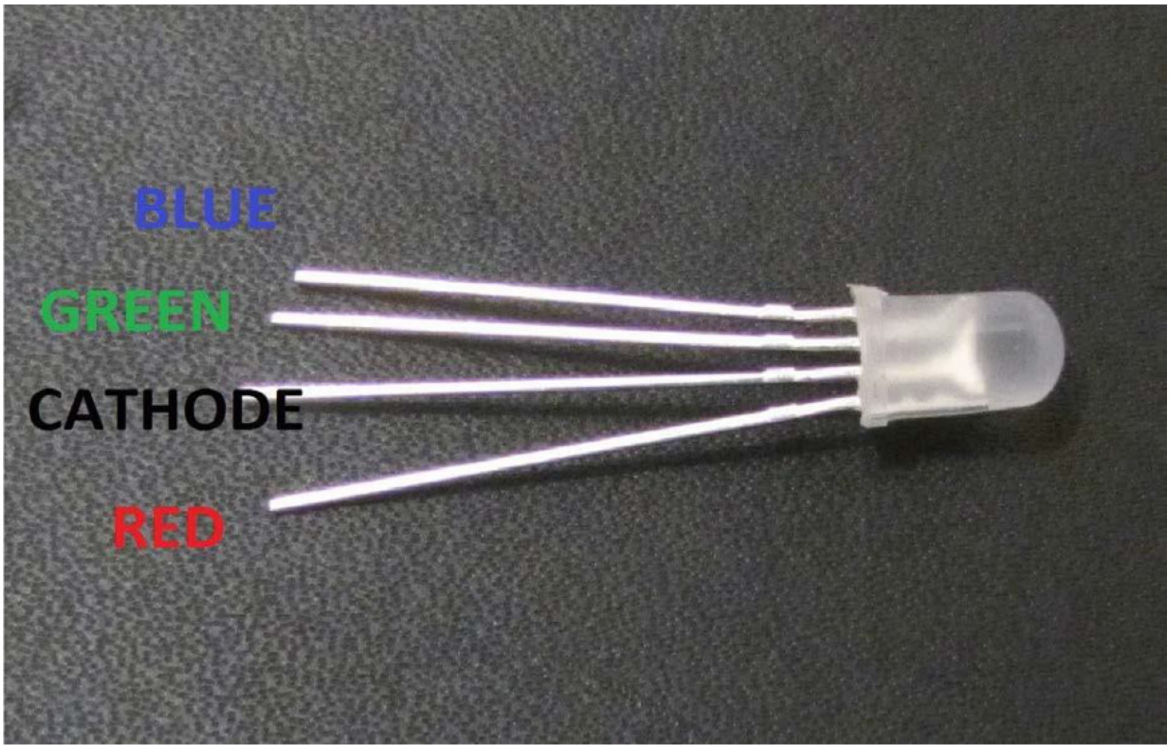
Component Introduction

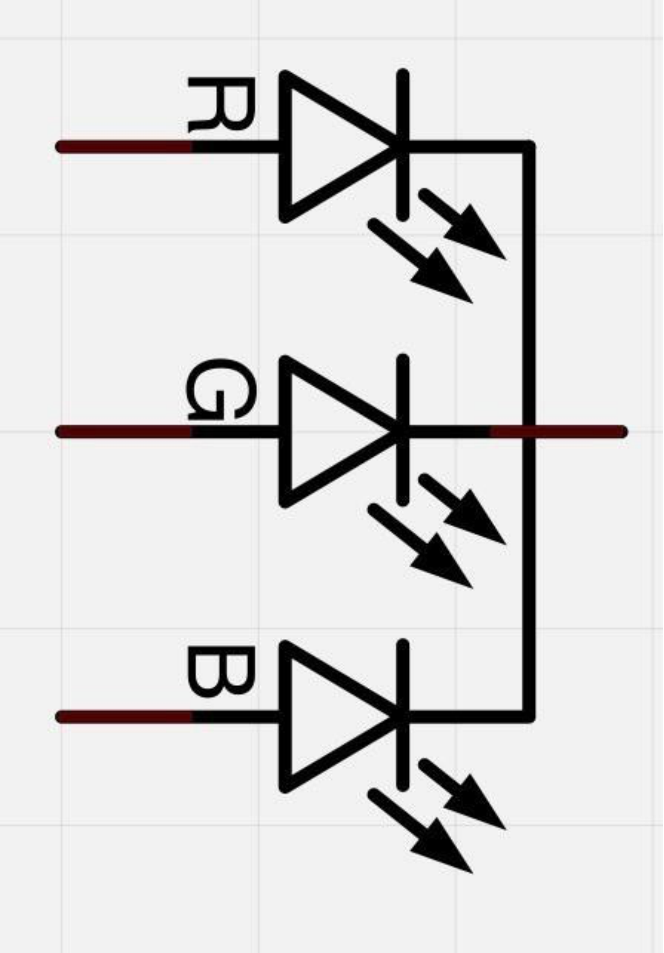
RGB:

At first glance, RGB (Red, Green and Blue) LEDs look just like regular LEDs. However, inside the usual LED package, there are actually three LEDs, one red, one green and yes, one blue. By controlling the brightness of each of the individual LEDs you can mix pretty much any color you want.

We mix colors the same way you would mix paint on a palette - by adjusting the brightness of each of the three LEDs. The hard way to do this would be to use different value resistors (or variable resistors) as we did with in Lesson 2, but that's a lot of work! Fortunately for us, MEGA 2560 R3 board has an analog Write function that you can use with pins marked with a ~ to output a variable amount of power to the appropriate LEDs.

The RGB LED has four leads. There is one lead going to the positive connection of each of the single LEDs within the package and a single lead that is connected to all three negative sides of the LEDs.





Here on the photographs you can see 4 electrode LED. Every separate pin for Green or Blue or Red color is called Anode. You will always connect “+” to it. Cathode goes to “-“ (ground). If you connect it other way round the LED will not light.

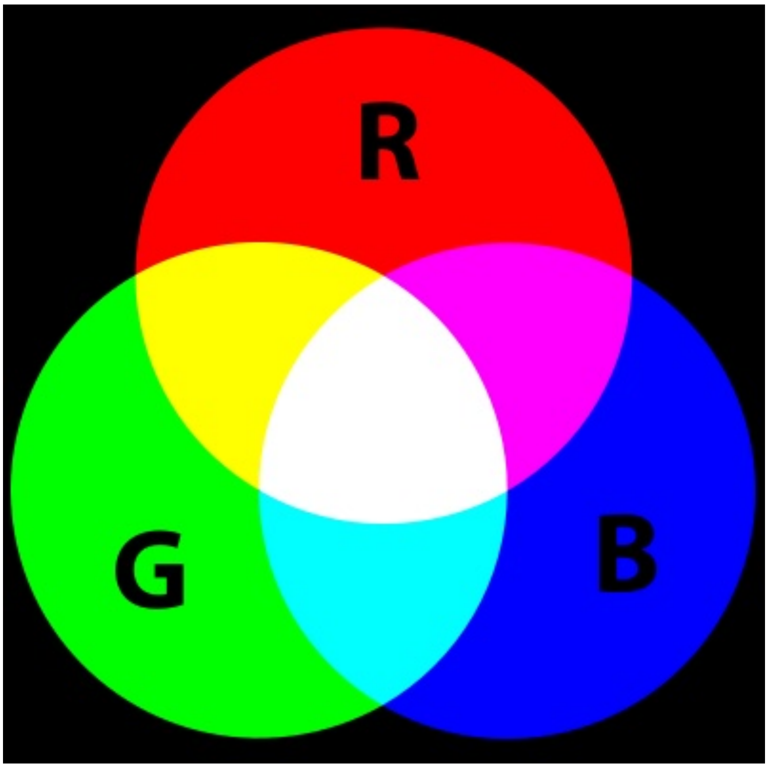
The common negative connection of the LED package is the second pin from the flat side. It is also the longest of the four leads and will be connected to the ground.

Each LED inside the package requires its own 220Ω resistor to prevent too much current flowing through it. The three positive leads of the LEDs (one red, one green and one blue) are connected to MEGA 2560 output pins using these resistors.

COLOR:

The reason that you can mix any color you like by varying the quantities of red, green and blue light is that your eye has three types of light receptor in it (red, green and blue). Your eye and brain process the amounts of red, green and blue and convert it into a color of the spectrum.

In a way, by using the three LEDs, we are playing a trick on the eye. This same idea is used in TVs, where the LCD has red, green and blue color dots next to each other making up each pixel.



If we set the brightness of all three LEDs to be the same, then the overall color of the light will be white. If we turn off the blue LED, so that just the red and green LEDs are the same brightness, then the light will appear yellow.

We can control the brightness of each of the red, green and blue parts of the LED separately, making it possible to mix any color we like.

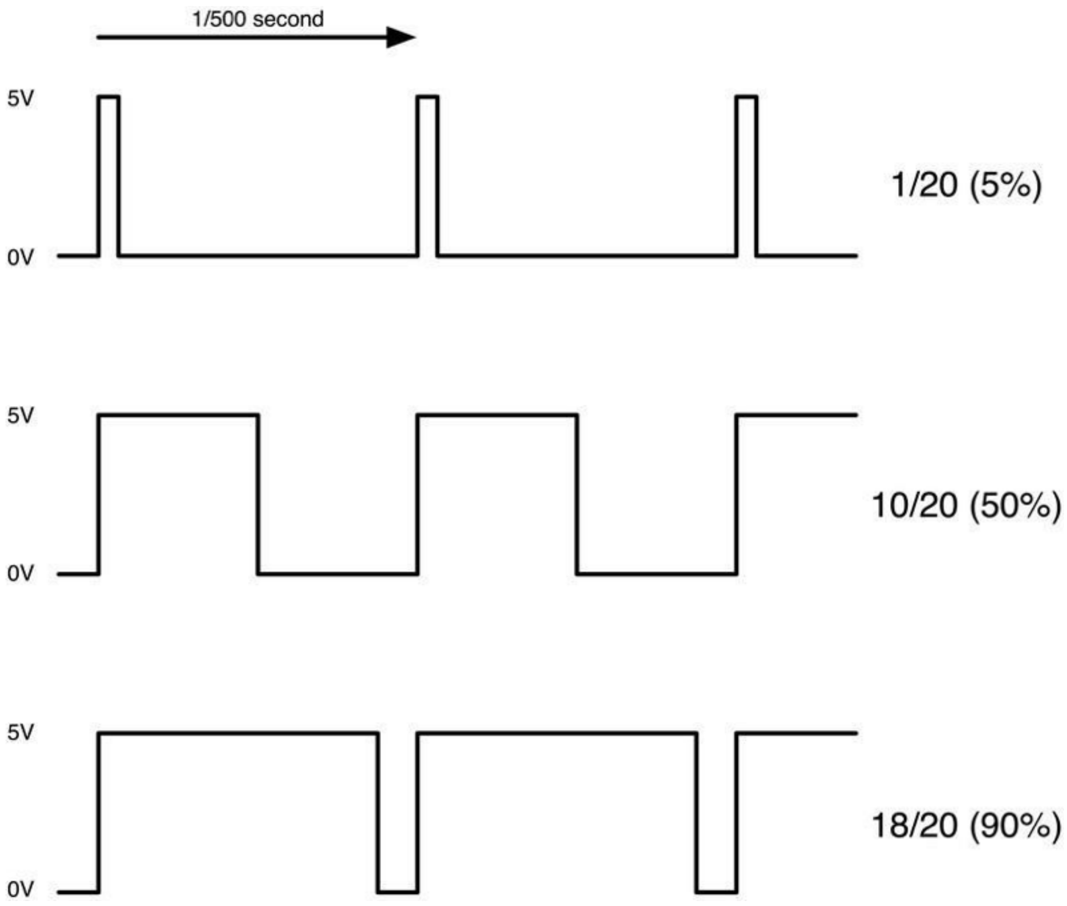
Black is not so much a color as an absence of light. Therefore, the closest we can come to black with our LED is to turn off all three colors.

Theory (PWM)

Pulse Width Modulation (PWM) is a technique for controlling power.

We also use it here to control the brightness of each of the LEDs.

The diagram below shows the signal from one of the PWM pins on the MEGA 2560.



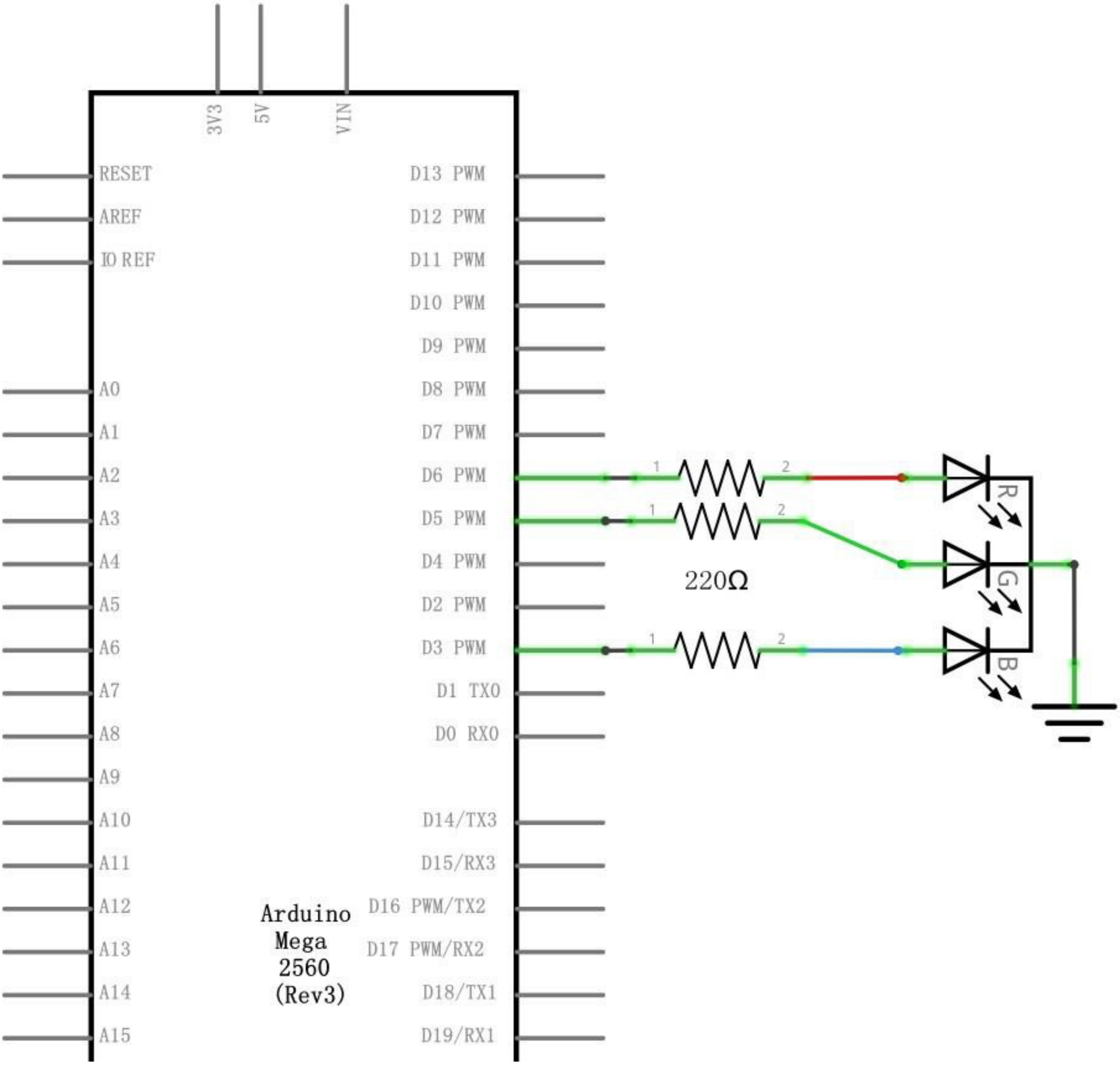
Roughly every 1/500 of a second, the PWM output will produce a pulse. The length of this pulse is controlled by the 'analog Write' function. So 'analog Write(0)' will not produce any pulse at all and 'analog Write(255)' will produce a pulse that lasts all the way until the next pulse is due, so that the output is actually on all the time.

If we specify a value in the analog Write that is somewhere in between 0 and 255, then we will produce a pulse. If the output pulse is only high for 5% of the time, then whatever we are driving will only receive 5% of full power.

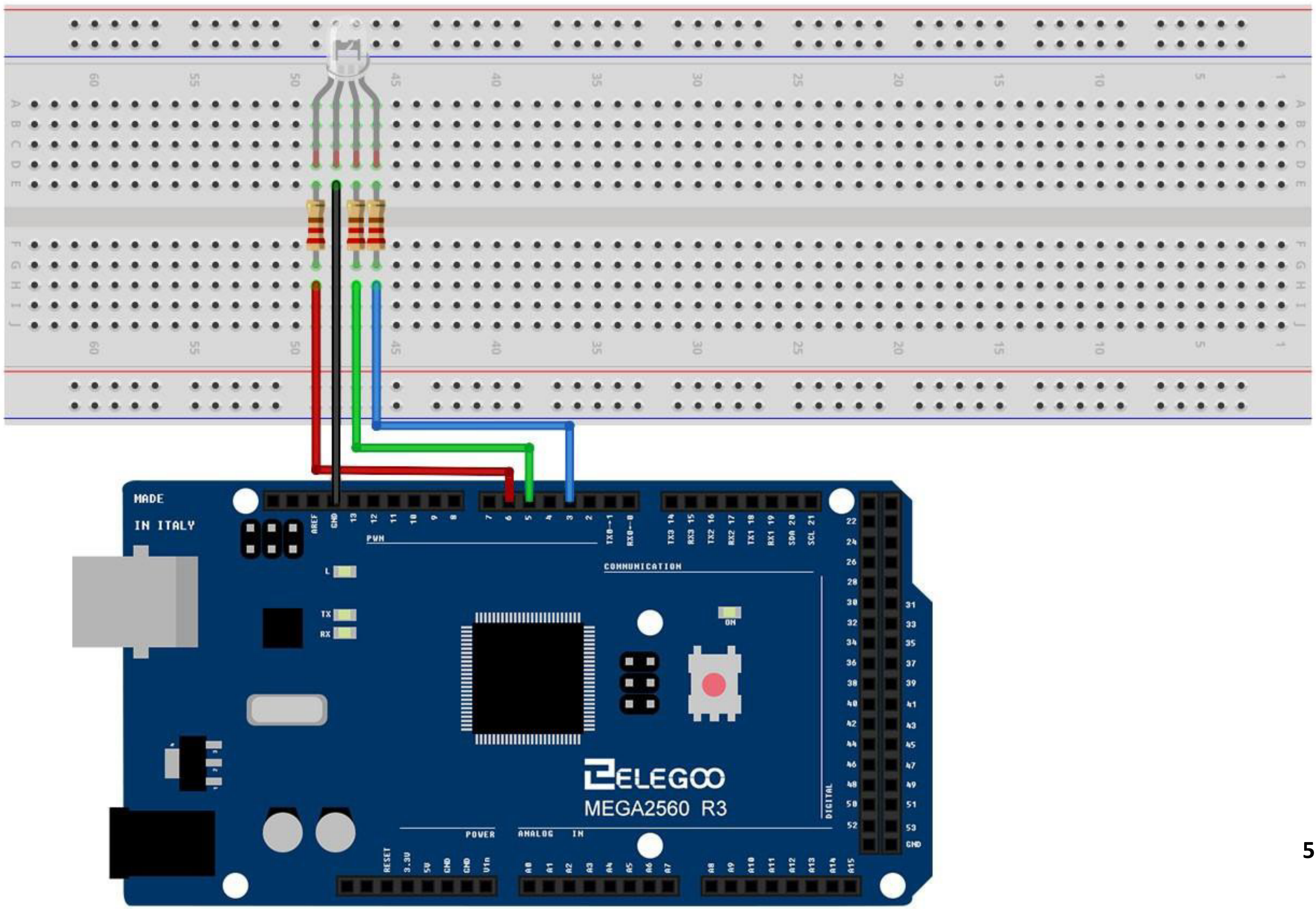
If, however, the output is at 5V for 90% of the time, then the load will get 90% of the power delivered to it. We cannot see the LEDs turning on and off at that speed, so to us, it just looks like the brightness ischanging.

Connection

Schematic

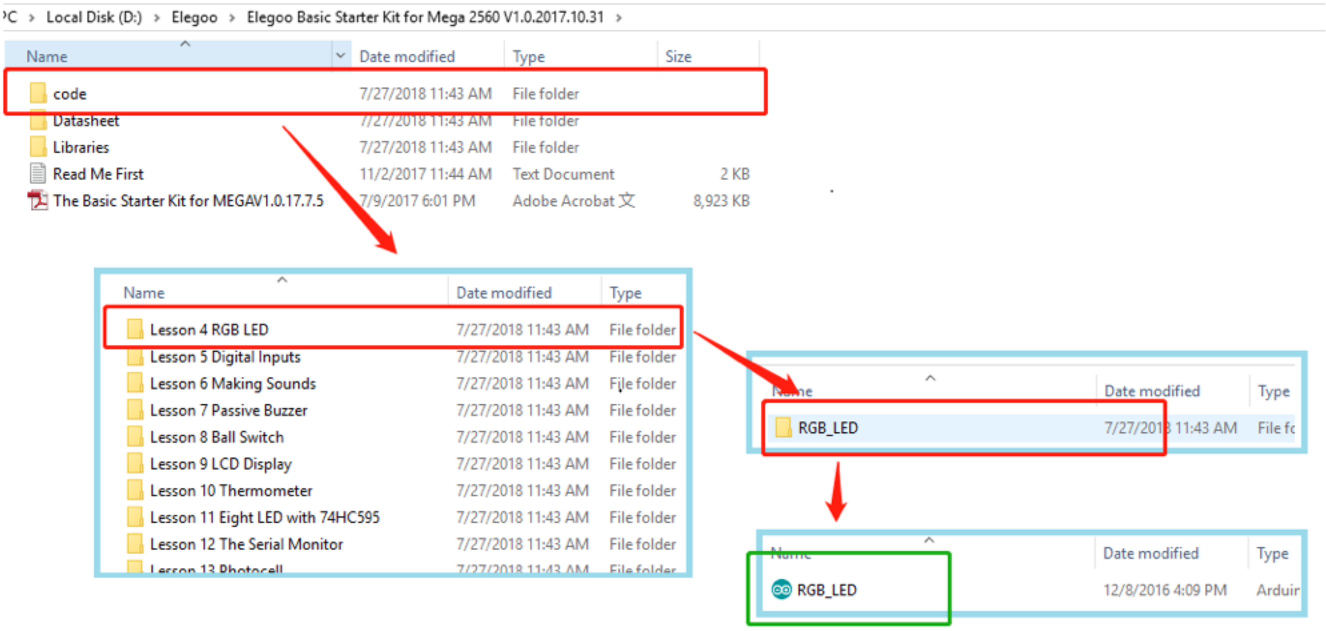


Wiring diagram



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.



›C › Local Disk (D:) › Elegoo › Elegoo Basic Starter Kit for Mega 2560 V1.0.2017.10.31 ›

Name	Date modified	Type	Size
code	7/27/2018 11:43 AM	File folder	
Datasheet	7/27/2018 11:43 AM	File folder	
Libraries	7/27/2018 11:43 AM	File folder	
Read Me First	11/2/2017 11:44 AM	Text Document	2 KB
The Basic Starter Kit for MEGAV1.0.17.7.5	7/9/2017 6:01 PM	Adobe Acrobat 文	8,923 KB

Name	Date modified	Type
Lesson 4 RGB LED	7/27/2018 11:43 AM	File folder
Lesson 5 Digital Inputs	7/27/2018 11:43 AM	File folder
Lesson 6 Making Sounds	7/27/2018 11:43 AM	File folder
Lesson 7 Passive Buzzer	7/27/2018 11:43 AM	File folder
Lesson 8 Ball Switch	7/27/2018 11:43 AM	File folder
Lesson 9 LCD Display	7/27/2018 11:43 AM	File folder
Lesson 10 Thermometer	7/27/2018 11:43 AM	File folder
Lesson 11 Eight LED with 74HC595	7/27/2018 11:43 AM	File folder
Lesson 12 The Serial Monitor	7/27/2018 11:43 AM	File folder
Lesson 13 Photocell	7/27/2018 11:43 AM	File folder

Name	Date modified	Type
RGB_LED	7/27/2018 11:43 AM	File fc

Name	Date modified	Type
RGB_LED	12/8/2016 4:09 PM	Arduir

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);
}
```

Before we take a look at the 'loop' function, let’s look at the last function in the sketch.

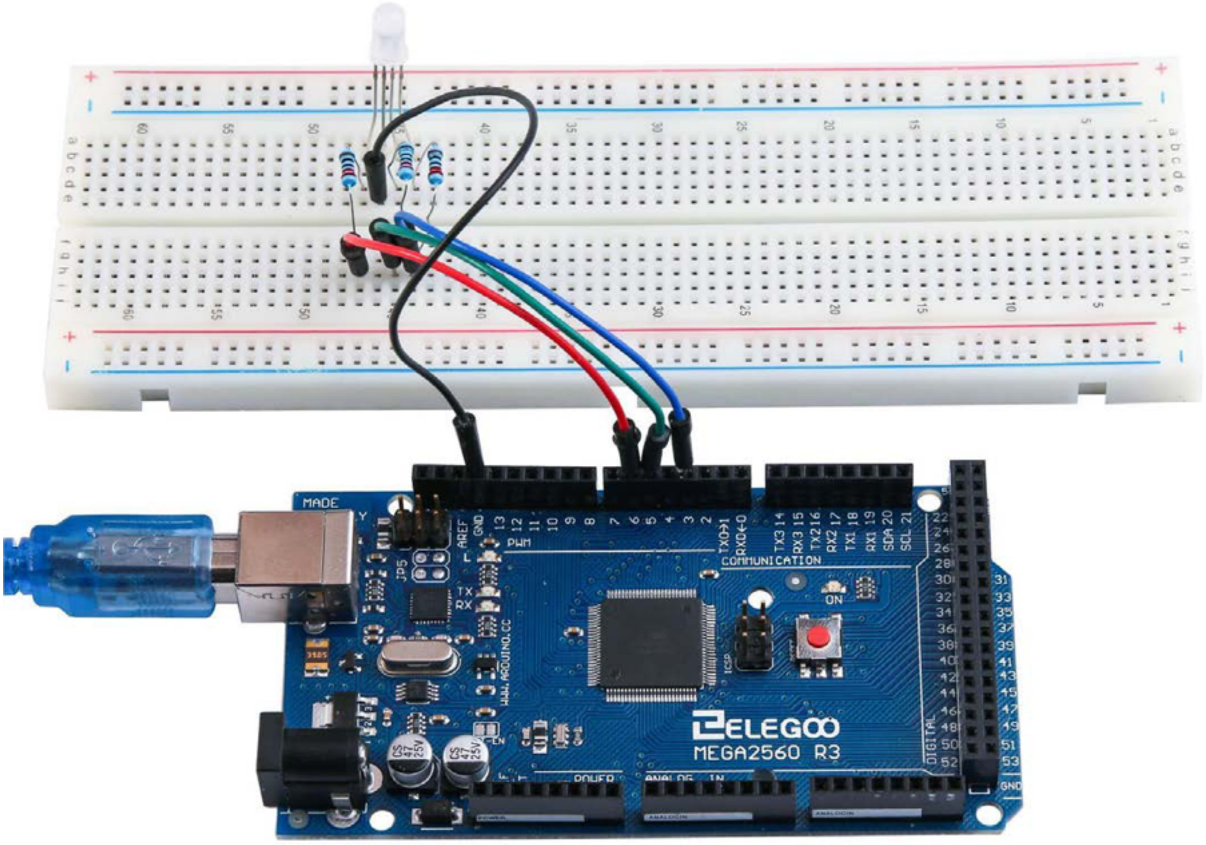
The define variables

```
redValue = 255; // choose a value between 1 and 255 to change thecolor.
greenValue = 0;
blueValue = 0;
```

This function takes three arguments, one for the brightness of the red, green and blue LEDs. In each case the number will be in the range 0 to 255, where 0 means off and 255 means maximum brightness. The function then calls 'analogWrite' to set the brightness of each LED.

Try adding a few colors of your own to the sketch and watch the effect on your LED.

Example picture



Lesson 6 Active buzzer

Overview

In this lesson, you will learn how to generate a sound with an active buzzer.

Component Required:

(1) x Elegoo Mega 2560 R3

(1) x Active buzzer

(2) x F-M wires (Female to Male DuPont wires)

Component Introduction

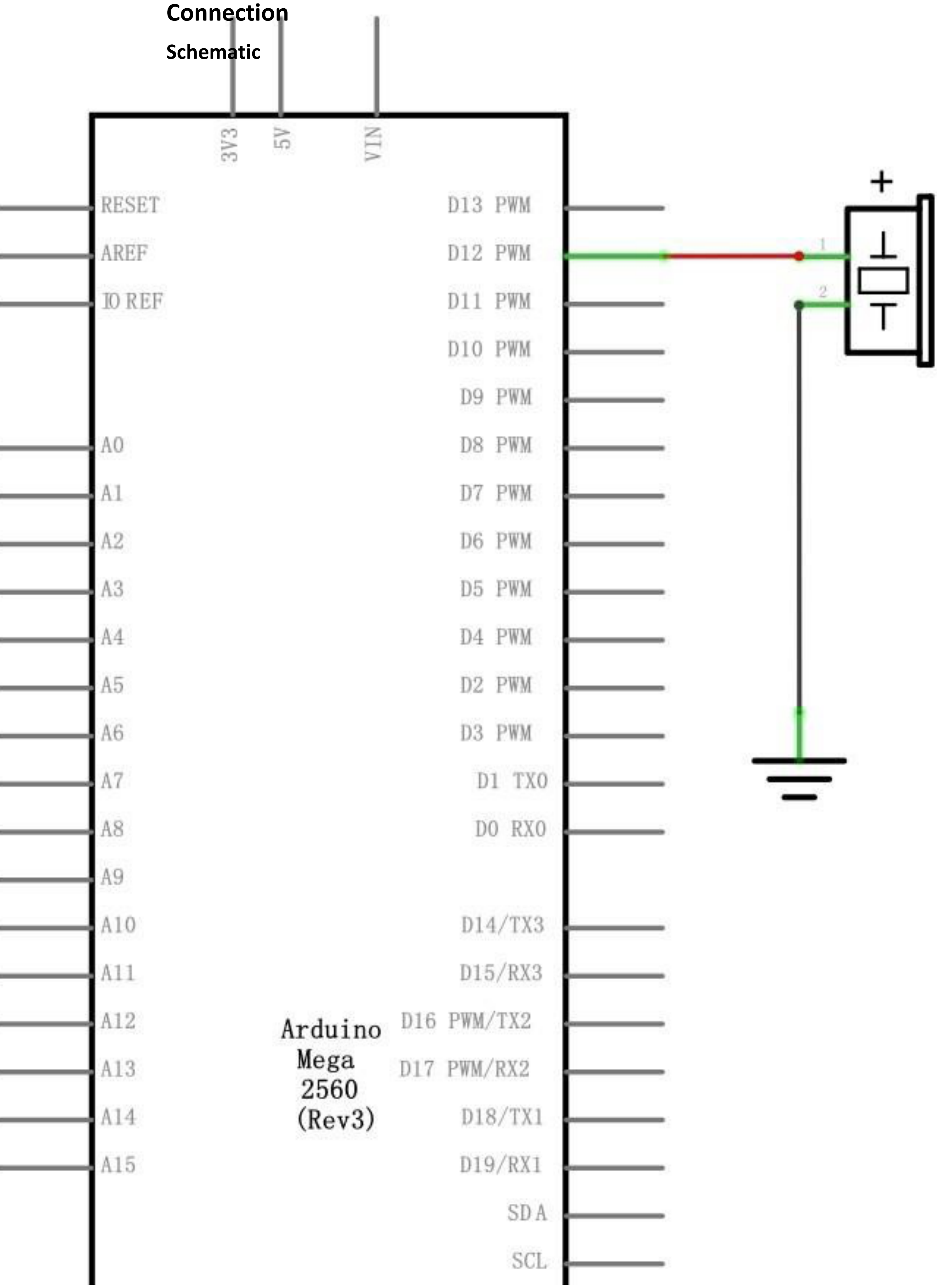
BUZZER:

Electronic buzzers are DC-powered and equipped with an integrated circuit. They are widely used in computers, printers, photocopiers, alarms, electronic toys, automotive electronic devices, telephones, timers and other electronic products for voice devices. Buzzers can be categorized as active and passive ones. Turn the pins of two buzzers face up. The one with a green circuit board is a passive buzzer, while the other enclosed with a black tape is an active one.

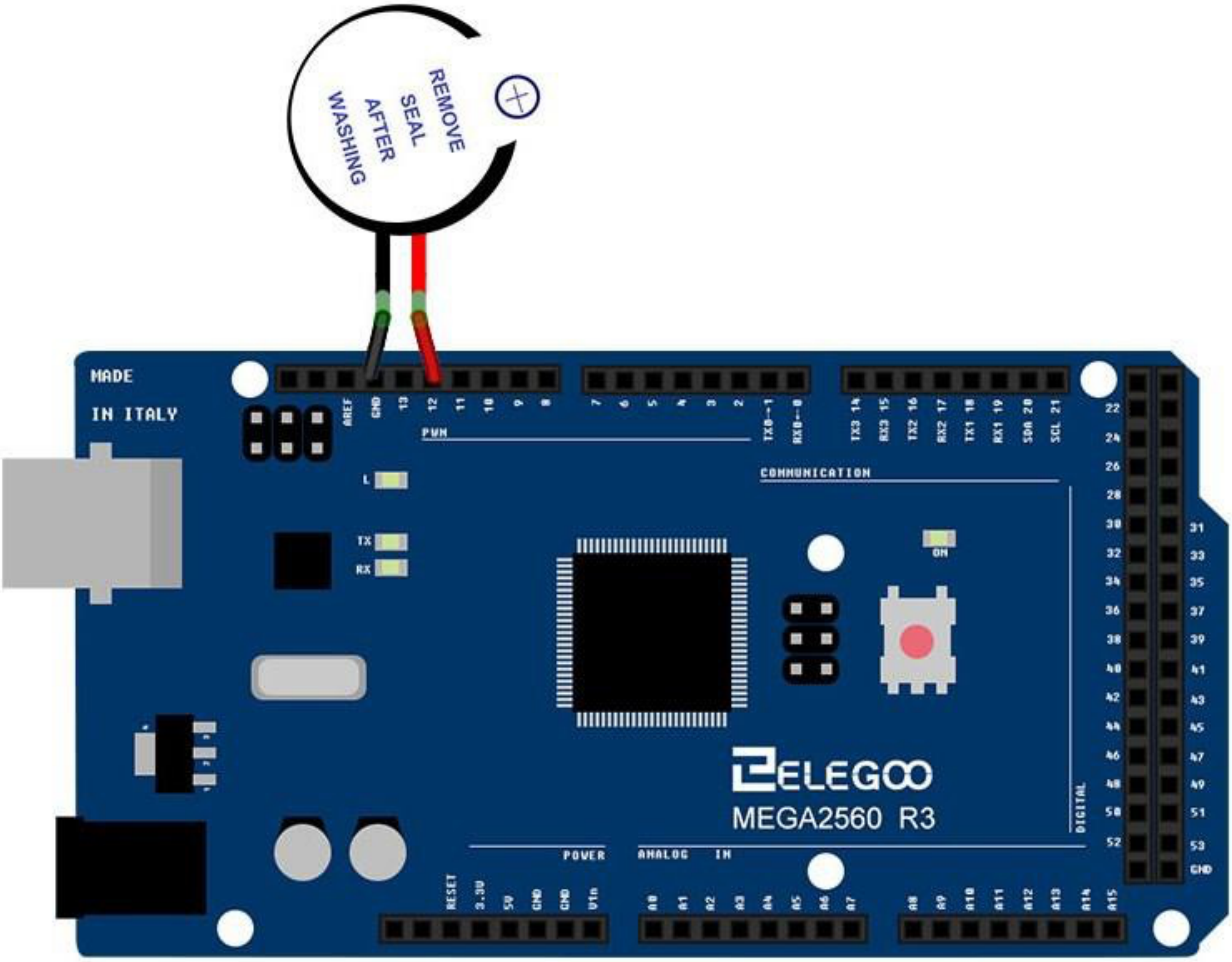
The difference between the two is that an active buzzer has a built-in oscillating source, so it will generate a sound when electrified. A passive buzzer does not have such a source so it will not tweet if DC signals are used; instead, you need to use square waves whose frequency is between 2K and 5K to drive it. The active buzzer is often more expensive than the passive one because of multiple built-in oscillating circuits.

Connection

Schematic



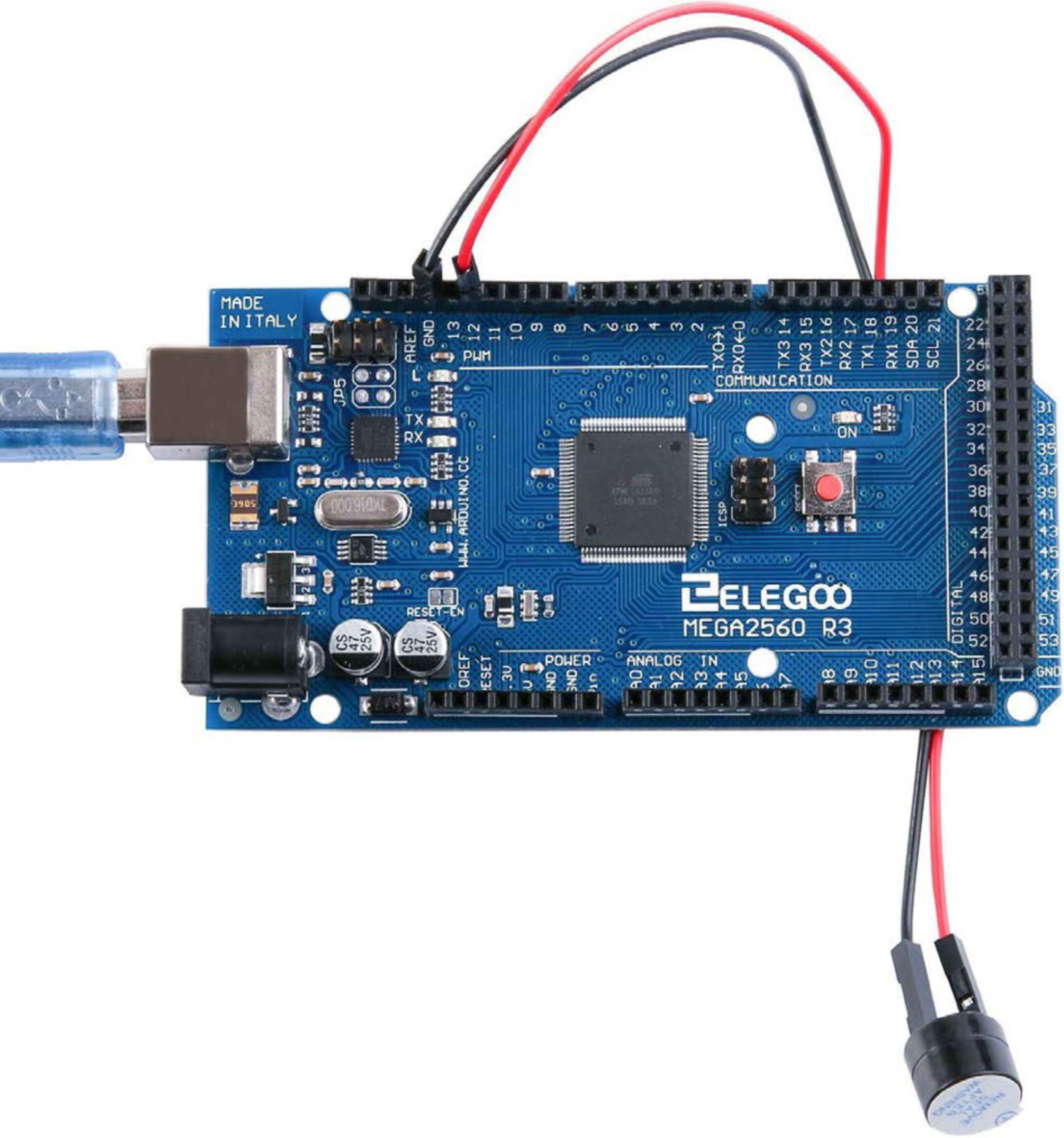
Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson 6 Making Sounds and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Example picture



Code

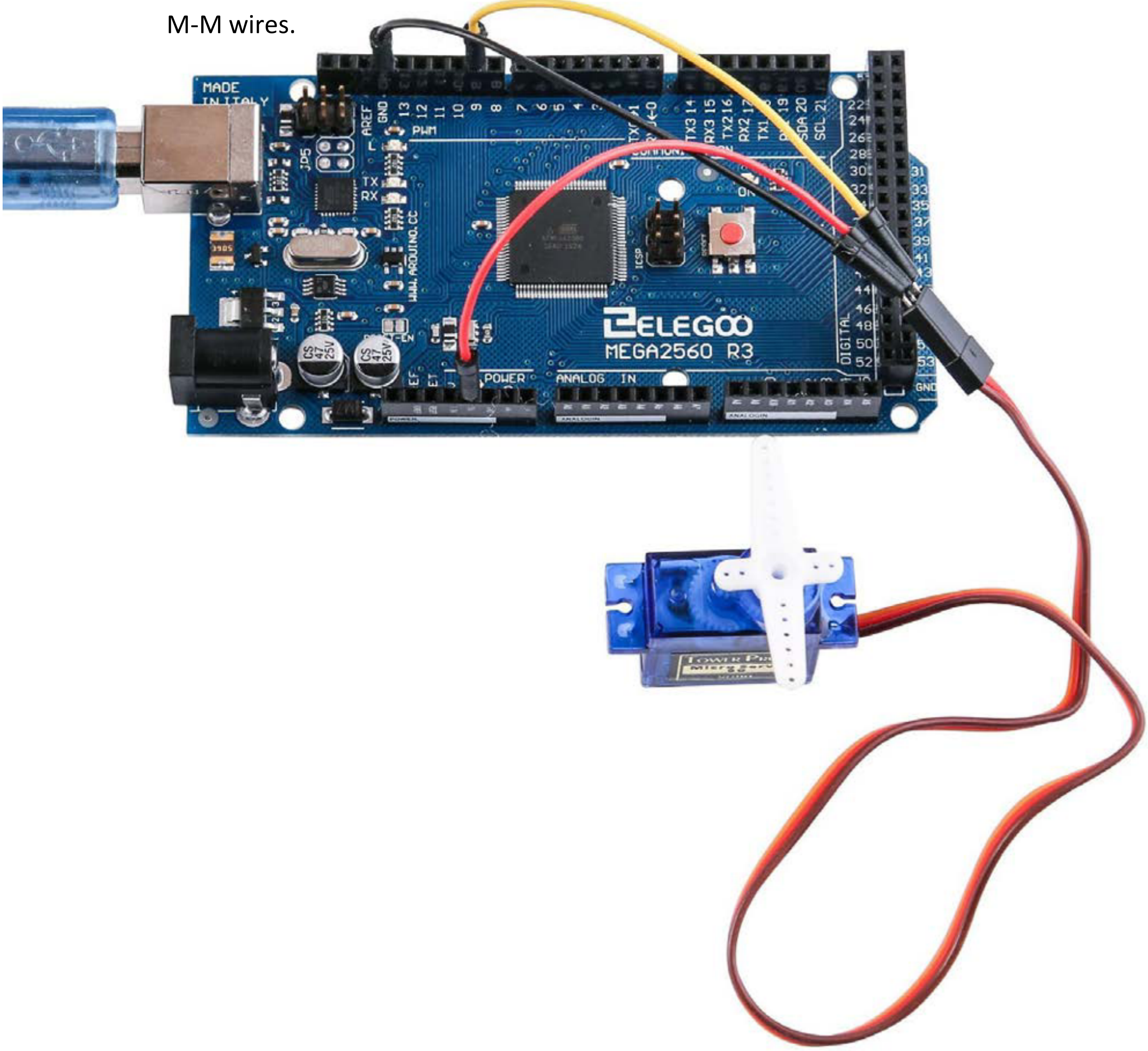
After wiring, please open the program in the code folder- Lesson 9 Servo and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are anyerrors.

Before you can run this, make sure that you have installed the < Servo> library or re-install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson1.

Example picture

In the picture, the brown wire of servo is adapted via the black M-M wires, the red one is adapted via the red M-M wires, and the orange one is adapted via the yellow M-M wires.



Lesson 10 Ultrasonic Sensor Module

Overview

Ultrasonic sensor is great for all kind of projects that need distance measurements, avoiding obstacles as examples.

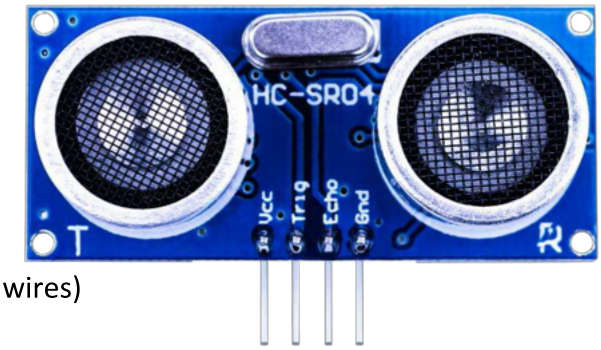
The HC-SR04 is inexpensive and easy to use since we will be using a Library specifically designed for these sensor.

Component Required:

(1) x Elegoo Mega 2560 R3

(1) x Ultrasonic sensor module

(4) x F-M wires (Female to Male DuPont wires)



Component Introduction

Ultrasonic sensor

Ultrasonic sensor module HC-SR04 provides 2cm-400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

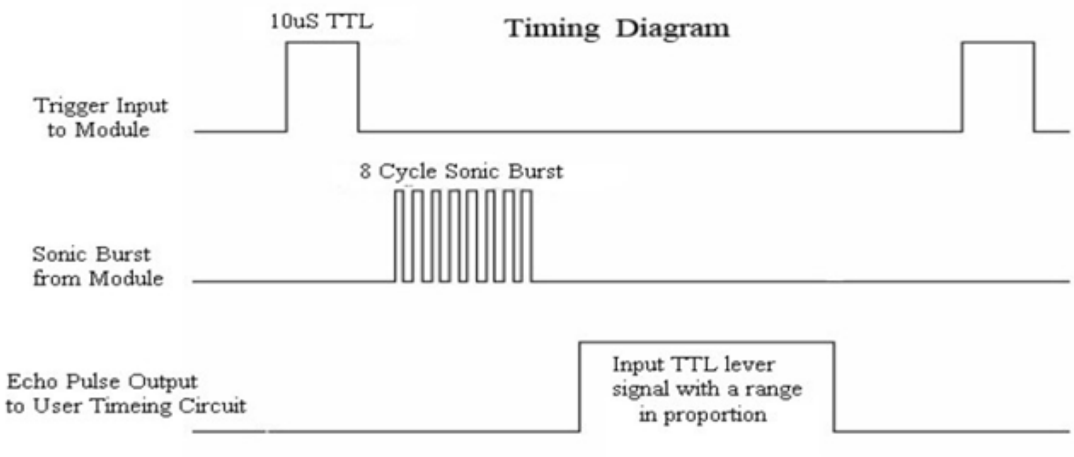
(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic tore turning.

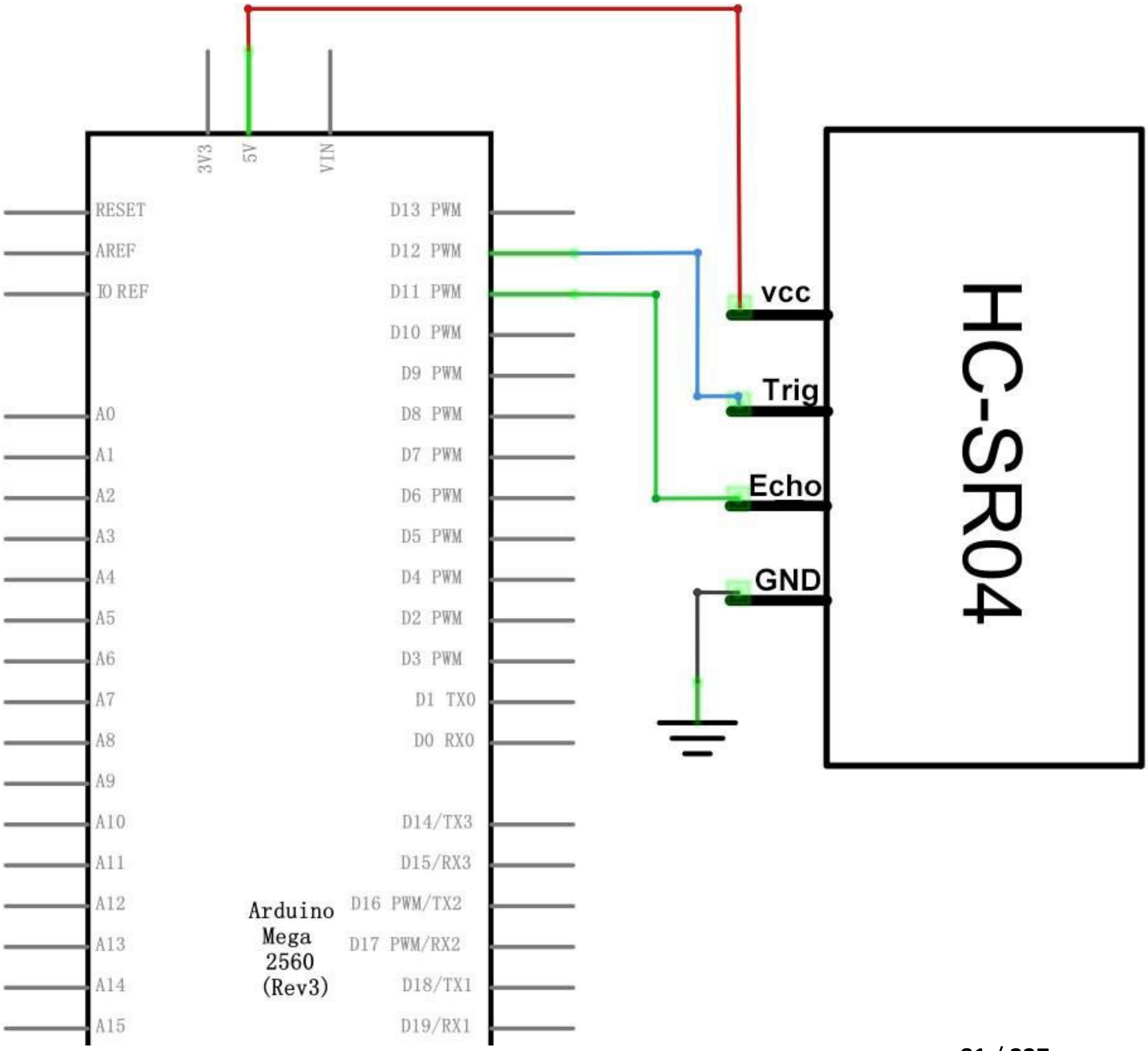
Test distance = (high level time × velocity of sound (340m/s) /2

The Timing diagram is shown below. You only need to supply a short 10us pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: us / 58 = centimeters or us / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

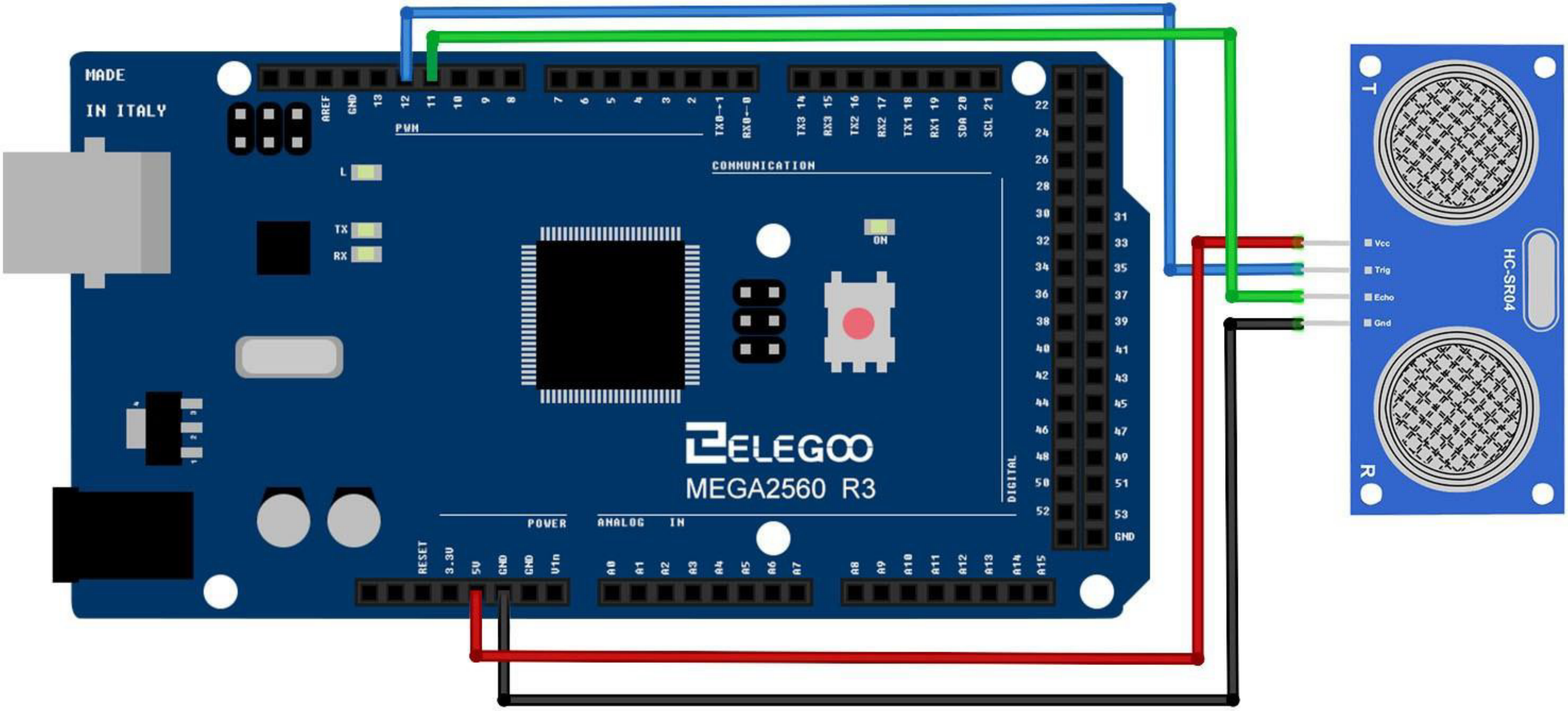

10uS TTL
Timing Diagram
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timeing Circuit
Input TTL lever signal with a range in proportion

Connection

Schematic



Wiring diagram



Code

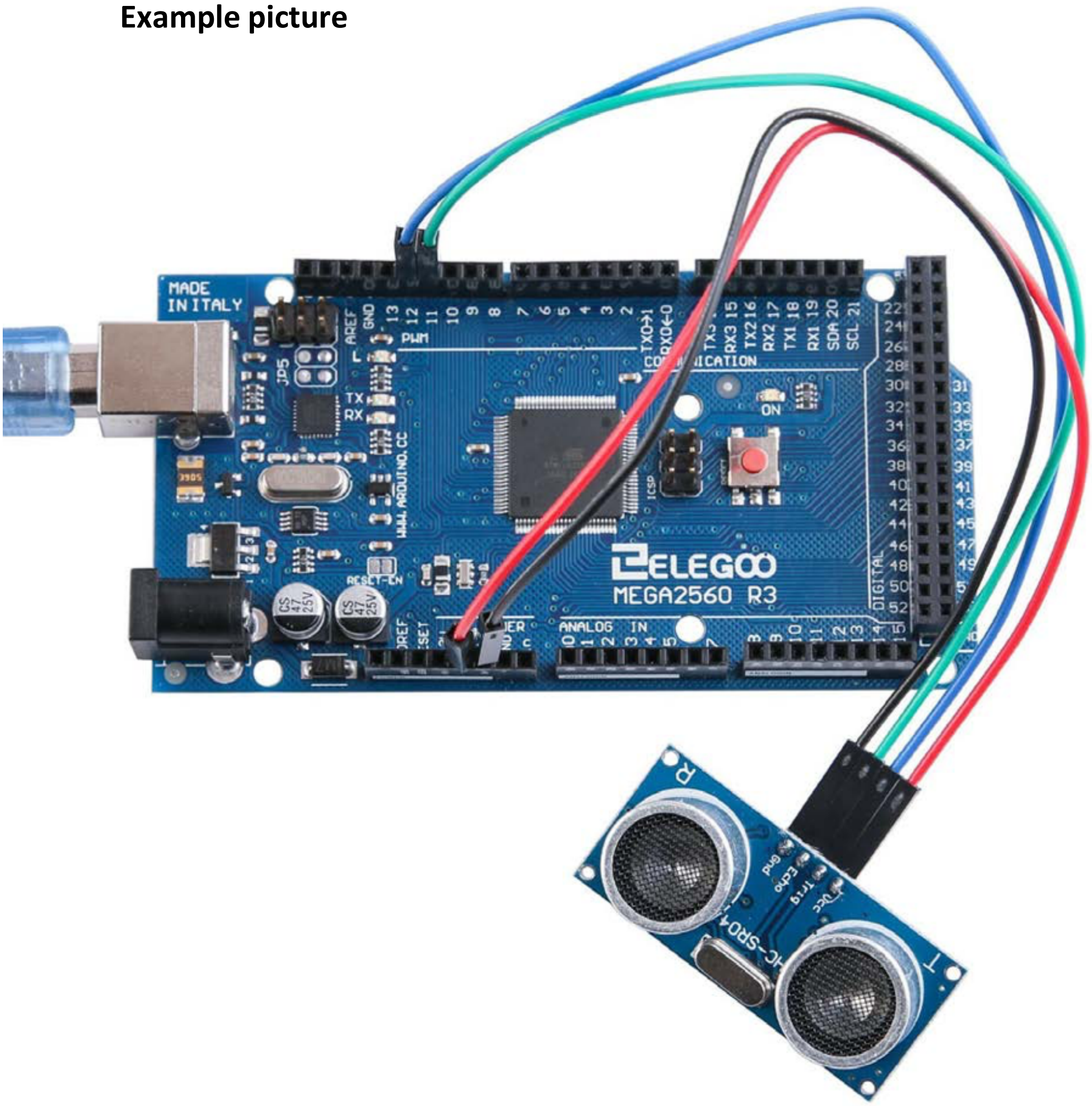
Using a Library designed for these sensors will make our code short and simple. We include the library at the beginning of our code, and then by using simple commands we can control the behavior of the sensor.

After wiring, please open the program in the code folder- Lesson 10 Ultrasonic Sensor Module and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the < HC-SR04> library or re-install it, if necessary. Otherwise, your code won't work.

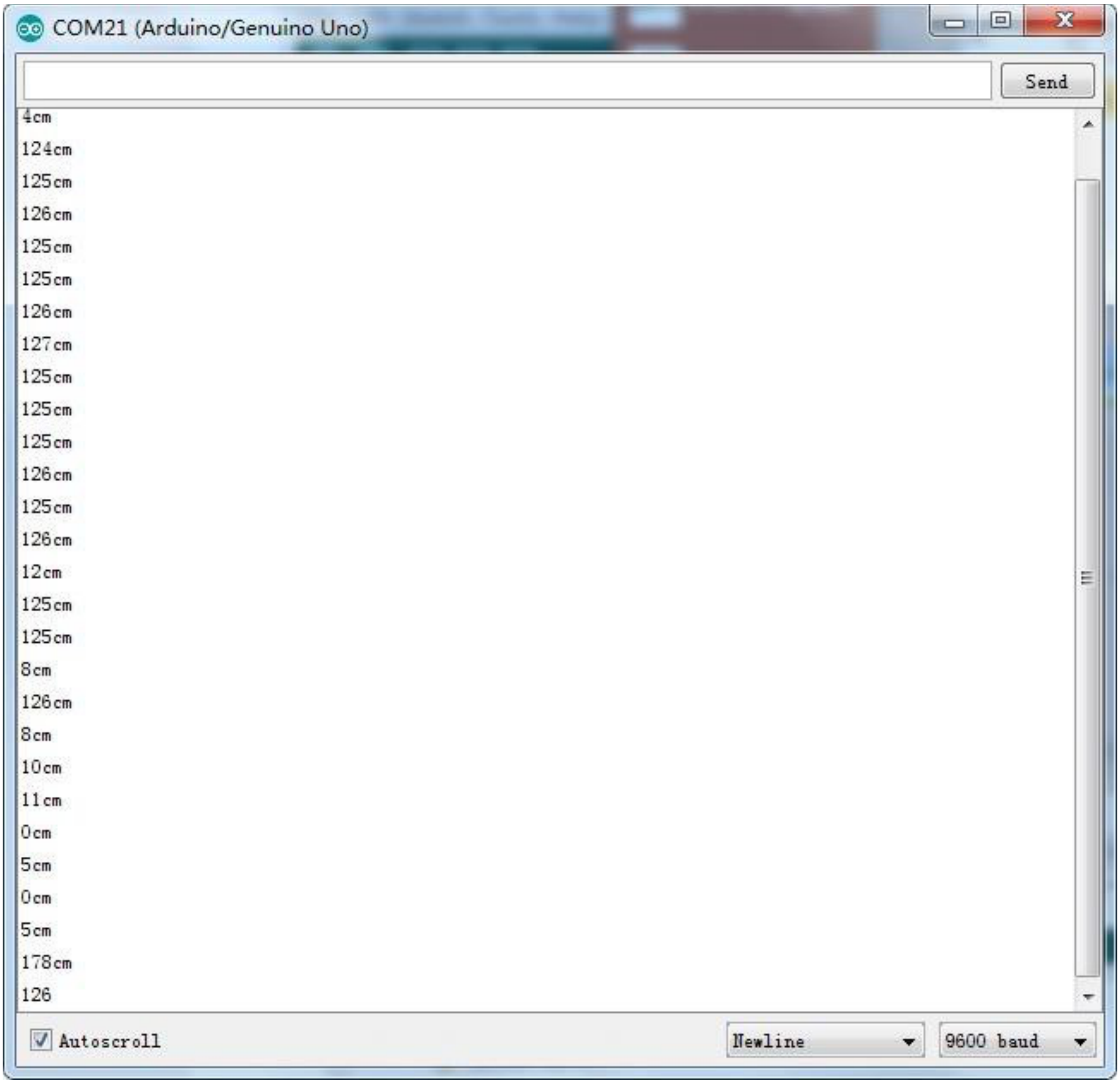
For details about loading the library file, see Lesson 1.

Example picture



Open the monitor then you can see the data asblow:

Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in Lesson1.



```
4cm
124cm
125cm
126cm
125cm
125cm
126cm
127cm
125cm
125cm
125cm
126cm
125cm
126cm
12cm
125cm
125cm
8cm
126cm
8cm
10cm
11cm
0cm
5cm
0cm
5cm
178cm
126
```

☑ Autoscroll   Newline   9600 baud

Lesson 19 Real Time Clock Module

Overview

In this lesson, you will learn how to use the RTC module, The DS1307 real-time clock is a low-power chip. Address and data are transferred serially through an I2C, which can be used unless being connected to UNO with only three data cables. DS1307 provides seconds, minutes, hours, day, date, month, and year information. Timekeeping operation continues while the part operates from the backup supply.

Component Required:

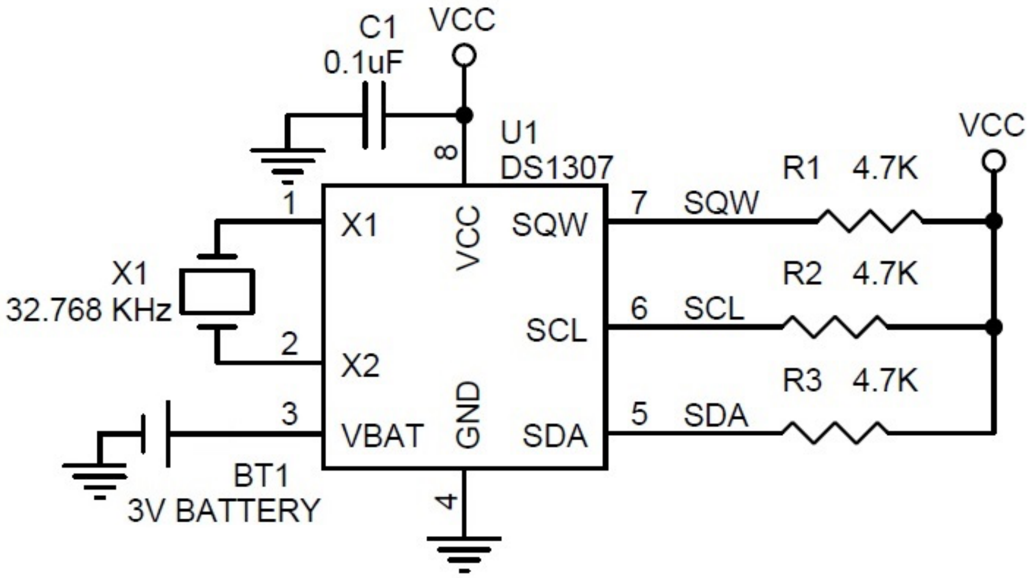
(1) x Elegoo Uno R3

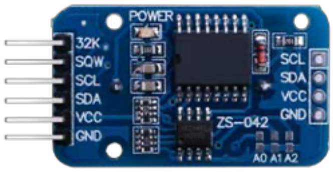
(1) x DS1307 RTC module

(4) x F-M wires (Female to Male DuPont wires)

Component Introduction

DS1307

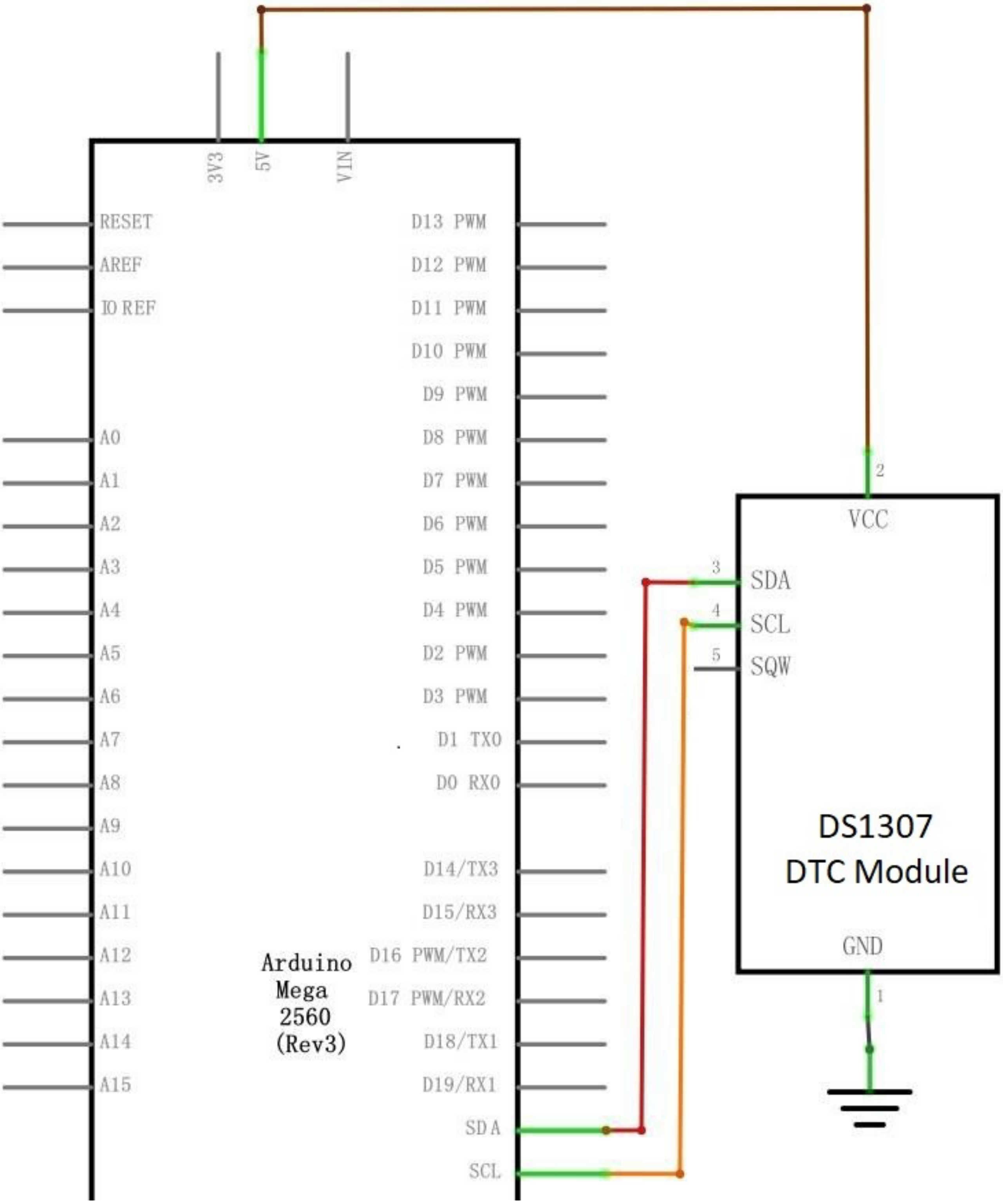




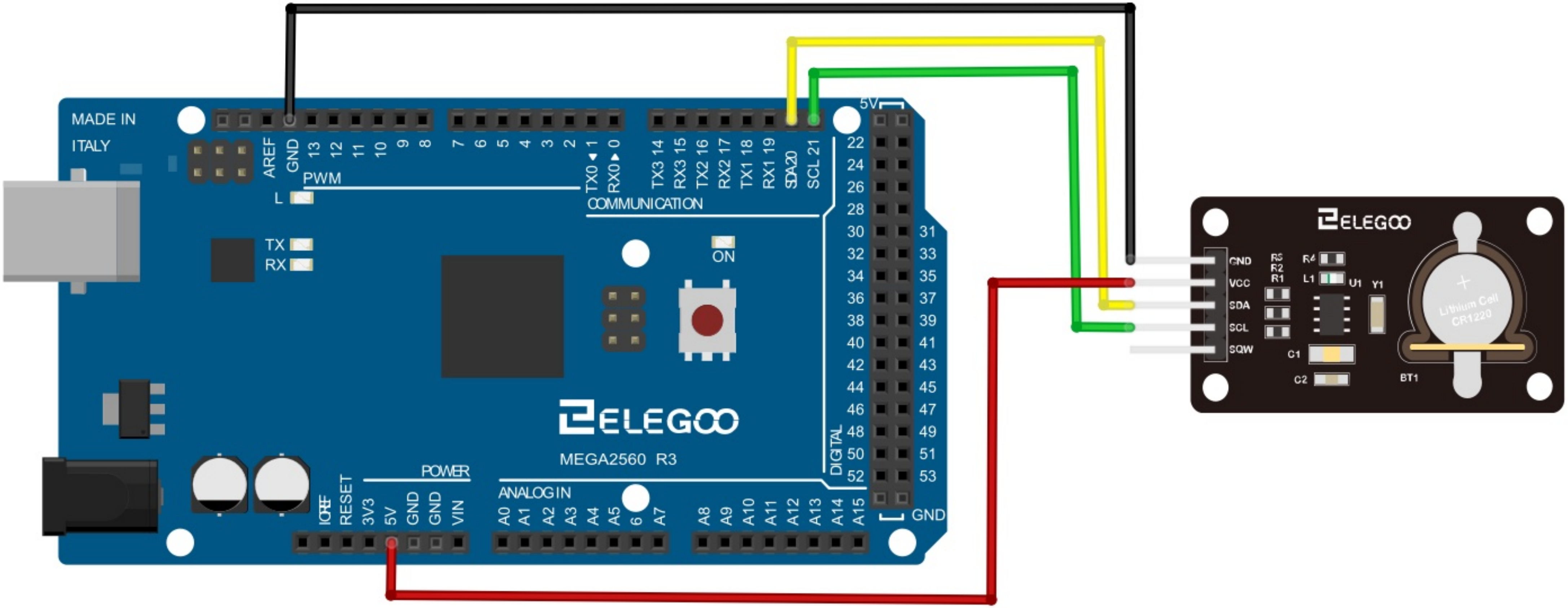
NOTICE: If you receive the old version shown on the left, please don't worry, its function and names of pins are the same as the new version.You could follow the wiring diagrams and sketch in the tutorial to make it work.

Connection

Schematic



Wiring diagram



Set up according to the following image.

Ignore the 32K and SQW pins; you will not need them. Plug the SCL pin into your UNO R3 board SCL port, and the SDA pin into the SDA port. The VCC pin plugs into the 5V port, and the GND plugs into the GND port.

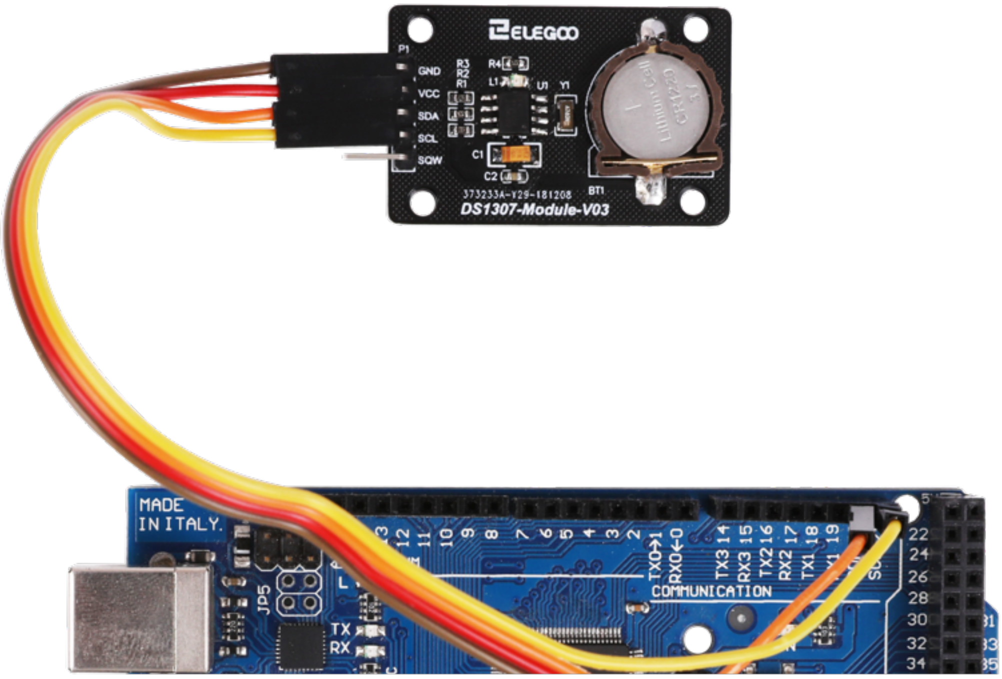
Code

After wiring, please open program in the code folder- Lesson 19 Real Time Clock Module and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the < DS1307 > library or re-install it, if necessary. Otherwise, your code won't work.

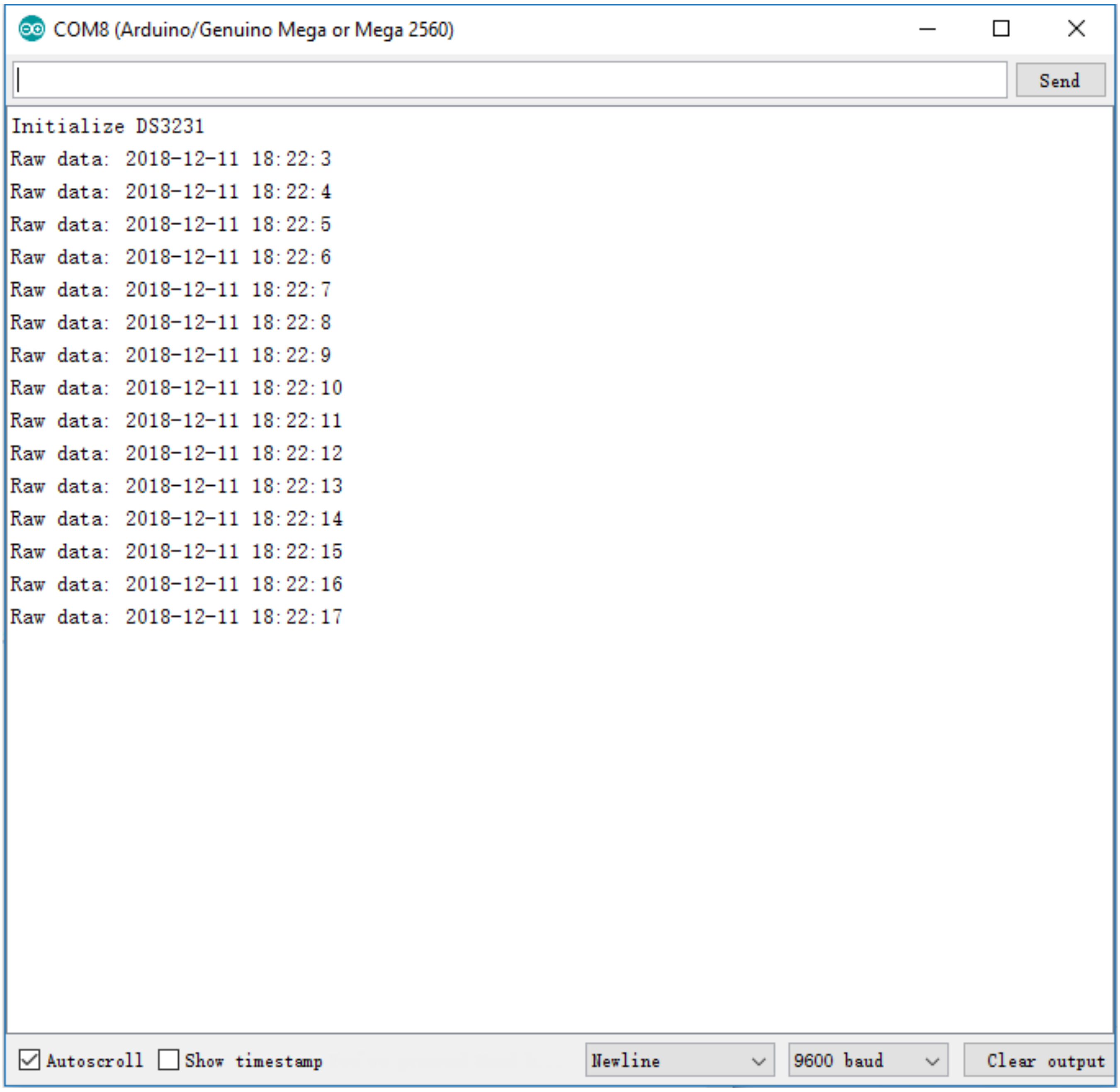
For details about loading the library file, see Lesson 1.

Example picture



Open the monitor then you can see the module can read the time as below:

Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details detail in Lesson 1.



```
Initialize DS3231
Raw data: 2018-12-11 18:22:3
Raw data: 2018-12-11 18:22:4
Raw data: 2018-12-11 18:22:5
Raw data: 2018-12-11 18:22:6
Raw data: 2018-12-11 18:22:7
Raw data: 2018-12-11 18:22:8
Raw data: 2018-12-11 18:22:9
Raw data: 2018-12-11 18:22:10
Raw data: 2018-12-11 18:22:11
Raw data: 2018-12-11 18:22:12
Raw data: 2018-12-11 18:22:13
Raw data: 2018-12-11 18:22:14
Raw data: 2018-12-11 18:22:15
Raw data: 2018-12-11 18:22:16
Raw data: 2018-12-11 18:22:17
```

Lesson 22 LCD Display

Overview

In this lesson, you will learn how to wire up and use an alphanumeric LCD display. The display has an LED backlight and can display two rows with up to 16 characters on each row. You can see the rectangles for each character on the display and the pixels that make up each character. The display is just white on blue and is intended for showing text.

In this lesson, we will run the Arduino example program for the LCD library, but in the next lesson, we will get our display to show the temperature, using sensors.

Component Required:

(1) x Elegoo Mega 2560 R3

(1) x LCD1602 module

(1) x Potentiometer (10k)

(1) x 830 tie-points Breadboard

(16) x M-M wires (Male to Male jumper wires)

Component Introduction

LCD1602

Introduction to the pins of LCD1602:

VSS: A pin that connects to ground

VDD: A pin that connects to a +5V power supply

VO: A pin that adjust the contrast of LCD1602

RS: A register select pin that controls where in the LCD's memory you are writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.

R/W: A Read/Write pin that selects reading mode or writing mode

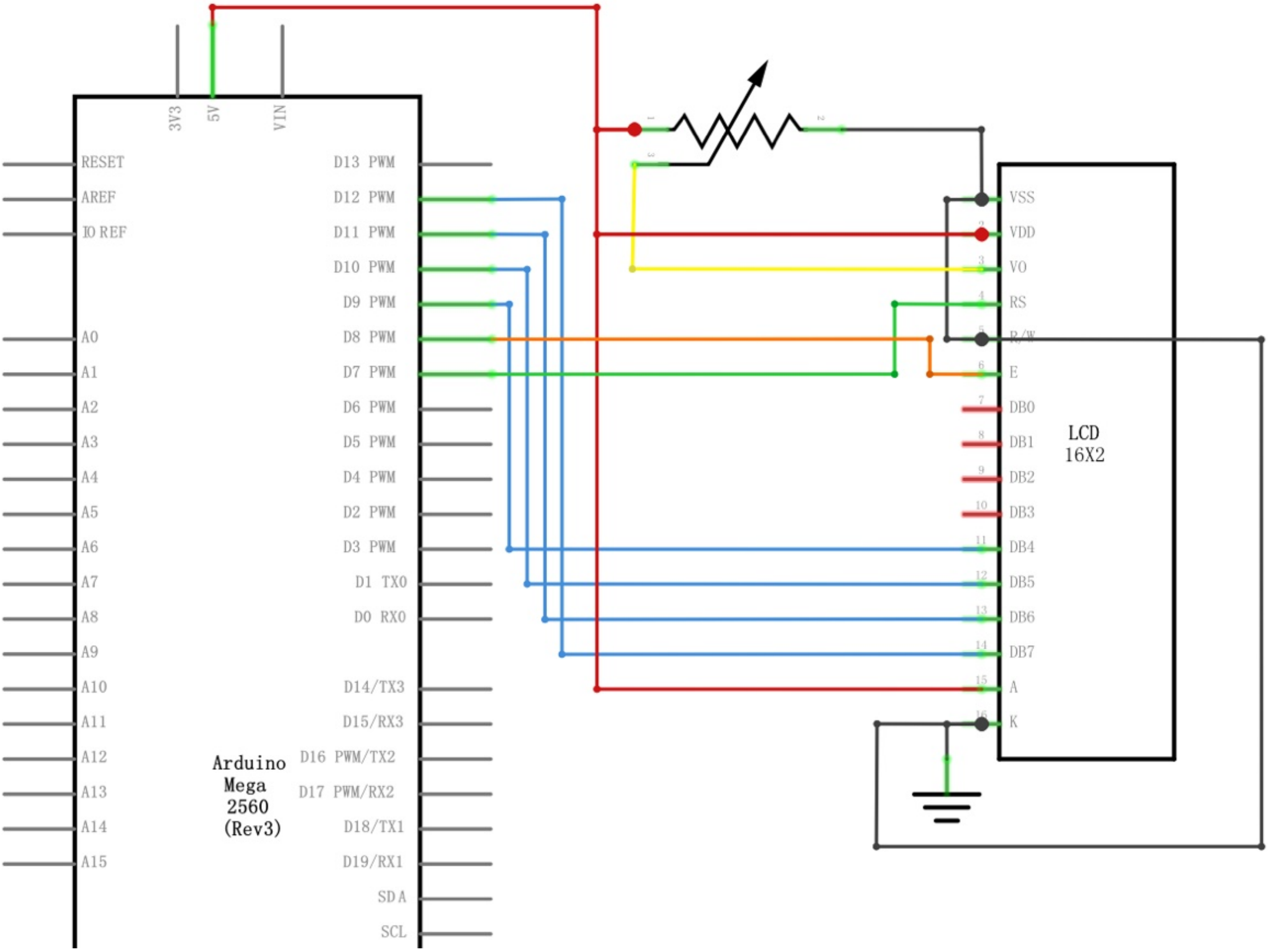
E: An enabling pin that, when supplied with low-level energy, causes the LDC module to execute relevant instructions.

D0-D7： Pins that read and write data

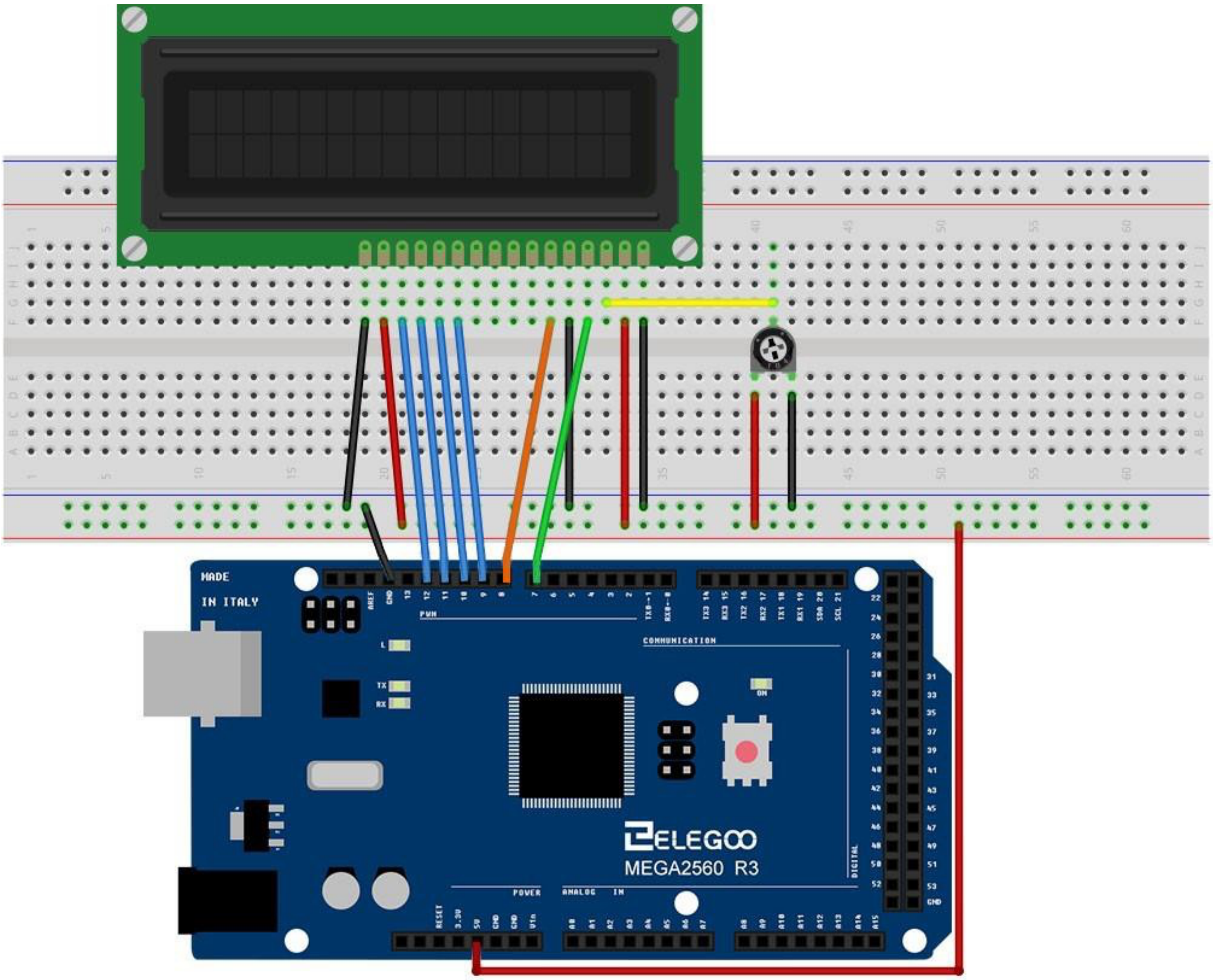
A and K: Pins that control the LED backlight

Connection

Schematic



Wiring diagram



The LCD display needs six Arduino pins, all set to be digital outputs. It also needs 5V and GND connections.

There are a number of connections to be made. Lining up the display with the top of the breadboard helps to identify its pins without too much counting, especially if the breadboard has its rows numbered with row 1 as the top row of the board. Do not forget, the long yellow lead that links the slider of the pot to pin 3 of the display. The 'pot' is used to control the contrast of the display.

You may find that your display is supplied without header pins attached to it. If so, follow the instructions in the next section.

Code

After wiring, please open the program in the code folder- Lesson 22 LCD Display and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the < Liquid Crystal > library or re-install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson 1.

Upload the code to your Arduino board and you should see the message 'hello, world' displayed, followed by a number that counts up from zero.

The first thing of note in the sketch is the line:

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

This tells Arduino that we wish to use the Liquid Crystal library.

Next we have the line that we had to modify. This defines which pins of the Arduino are to be connected to which pins of the display.

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

After uploading this code, make sure the backlight is lit up, and adjust the potentiometer all the way around until you see the text message

In the 'setup' function, we have two commands:

```
lcd.begin(16, 2);
lcd.print("Hello, World!");
```

The first tells the Liquid Crystal library how many columns and rows the display has.

The second line displays the message that we see on the first line of the screen.

In the 'loop' function, we a so have two commands:

```
lcd.setCursor(0, 1);
lcd.print(millis()/1000);
```

The first sets the cursor position (where the next text will appear) to column 0 & row 1. Both column and row numbers start at 0 rather than1.

The second line displays the number of milliseconds since the Arduino was reset.

Example picture