Preface

About SunFounder

SunFounder is a technology company focused on Raspberry Pi and Arduino open source community development. Committed to the promotion of open source culture, we strive to bring the fun of electronics making to people all around the world and enable everyone to be a maker. Our products include learning kits, development boards, robots, sensor modules and development tools. In addition to high quality products, SunFounder also offers video tutorials to help you build your own project. If you have interest in open source or making something cool, welcome to join us! Visit www.sunfounder.com for more!

About Super Kit

The super kit is suitable for the SunFounder Uno, SunFounder Mega 2560, SunFounder Duemilanove, and SunFounder Nano. All the code included in this kit works with these boards.

Our SunFounder board is fully compatible with Arduino.

With this kit, we will walk you through the know-how of using the SunFounder board in a hands-on way. Starting with the basics of electronics, you'll learn through building several creative projects. Including a selection of the most common and useful electronic components, this kit will help you "control" the physical world.

In this book, we will show you circuits with both realistic illustrations and schematic diagrams. You can go to our official website www.sunfounder.com to download related code by clicking **LEARN** -> **Get Tutorials** and watch videos under **VIDEO**.

Free Support

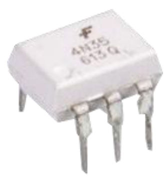
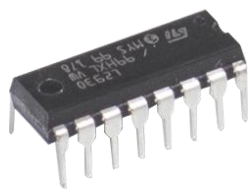
If you have any **TECHNICAL questions**, add a topic under **FORUM** section on our website and we'll reply as soon as possible.

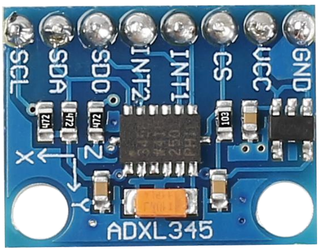
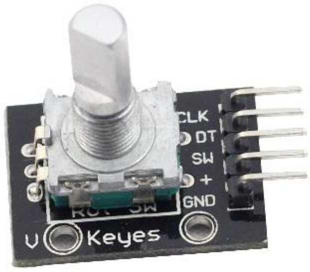
For **NON-TECH questions** like order and shipment issues, please **send an email to service@sunfounder.com**. You're also welcomed to share your projects on FORUM.

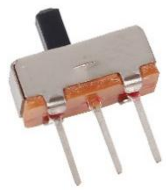
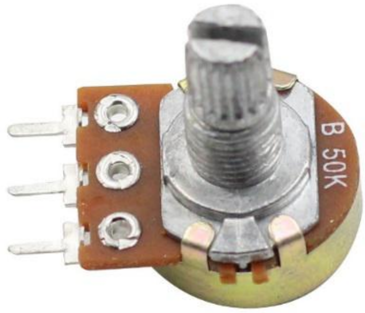
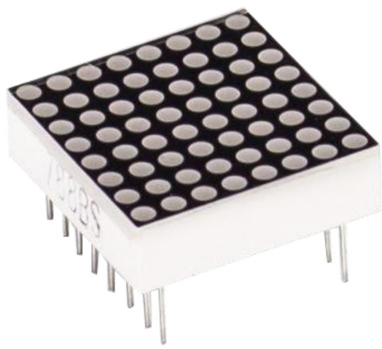
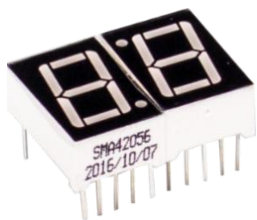
Rept 6.0

Contents

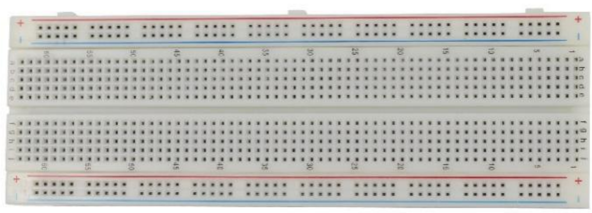
Components List

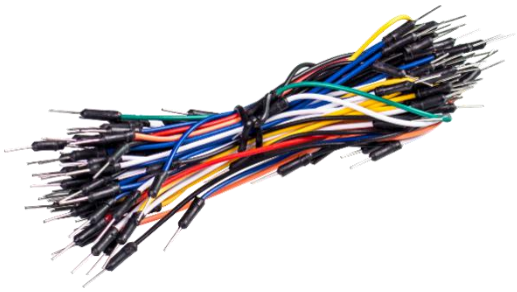
No.	Name	Qty.	Component
1	RGB LED	1	
2	555 Timer IC	1	
3	Optocoupler (4N35)	2	
4	Shift Register (74HC595)	2	
5	L293D	1	

6	Accelerometer ADXL345	1	
7	Rotary Encoder	1	
8	Push-Button (small)	5	
9	Resistor (220Ω)	8	 (red, red, black, black, brown)
10	Resistor (1kΩ)	4	 (brown, black, black, brown, brown)
11	Resistor (10kΩ)	4	 (brown, black, black, red, brown)
12	Resistor (1MΩ)	1	 (brown, black, green, gold)
13	Resistor (5.1MΩ)	1	 (green, brown, green, gold)

14	Switch	1	
15	Potentiometer (50kΩ)	1	
16	16*2 LCD Character Display	1	
17	8*8 Dot-Matrix Display	1	
18	7-Segment Character Display	2	

19	DC Motor	1	
20	LED (red)	16	
21	LED (white)	2	
22	LED (green)	2	
23	LED (yellow)	2	
24	NPN Transistor (S8050)	2	

25	PNP Transistor (S8550)	2	
26	Capacitor Ceramic 100nF	4	
27	Capacitor Ceramic 10nF	4	
28	Diode Rectifier (1N4007)	4	
29	Breadboard	1	
30	USB Cable	1	

31	Jumper Wire (Male to Male)	65	
32	Dupont Wire (Female to Male)	20	
33	Passive Buzzer	1	
34	Fan	1	

Note: After unpacking, please check that the number of components is correct and that all components are in good condition.

Get Started

Get the Code

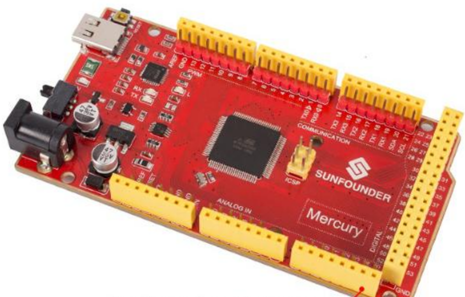
Before starting your own project, you must download the file **Super Kit V2.0 for Arduino.zip** on our official website by visiting **LEARN** -> **Get Tutorials** -> **Super Kit V2.0 for Arduino** and unzip it.

Arduino

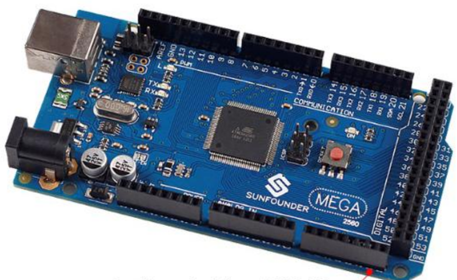
Description

Arduino is an open source platform with simple software and hardware. You can pick it up in short time even if you are a beginner. It provides an integrated development environment (IDE) for code compiling, compatible with multiple control boards. So you can just download the Arduino IDE, upload the sketches (i.e. the code files) to the board, and then you can see relative experimental phenomena. For more information, refer to http://www.arduino.cc.

SunFounder Arduino Board

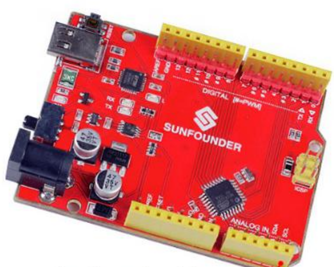


SunFounder Mercury



SunFounder Mega 2560 R3

SunFounder Nano



SunFounder Mars

SunFounder Uno R3

Now, SunFounder has five Arduino boards: SunFounder Uno R3 Board, Mega 2560 Board, Nano Board, SunFounder Mars Board and SunFounder Mercury Board. The Uno Board and the Mars Board share the same kind of pins at the same position. And the pins of the first half part on the Mega 2560 are the same as those on the Uno. The difference lies in that on the Mega 2560 board, the I2C pin (SDA) and pin SCL don't correspond to pin A4 and A5 but to pin 20 and 21. On the other hand, the Nano and the Uno also share the same kind of pins at the same position. Though the Nano is in a smaller size, making it applicable for projects.

The following is the parameters of the four boards:

Board / Parameters	Uno R3	Mars	Nano	Mega 2560	Mercury
Microcontroller	ATmega328P	ATmega328P	ATmega328P	ATmega2560	ATmega2560
USB connection	Type B	Type C	Mini-B	Type B	Type C
USB-to-Serial Chip	MEGA16U2	FT232R	PL2303	MEGA16U2	FT232R
Power jack?	Y	Y	N	Y	Y
Operating Voltage	5V	5V	5V	5V	5V
Input Voltage	7-12V	7-12V	7-12V	7-12V	7-12V
Digital I/O Pins	14 (of which 6 provide PWM output)	14 (of which 6 provide PWM output)	14 (of which 6 provide PWM output)	54 (of which 15 provide PWM output)	54 (of which 15 provide PWM output)
Analog Input Pins	6	6	8	16	16
DC Current per I/O Pin	20 mA	20mA	20 mA	20 mA	20 mA

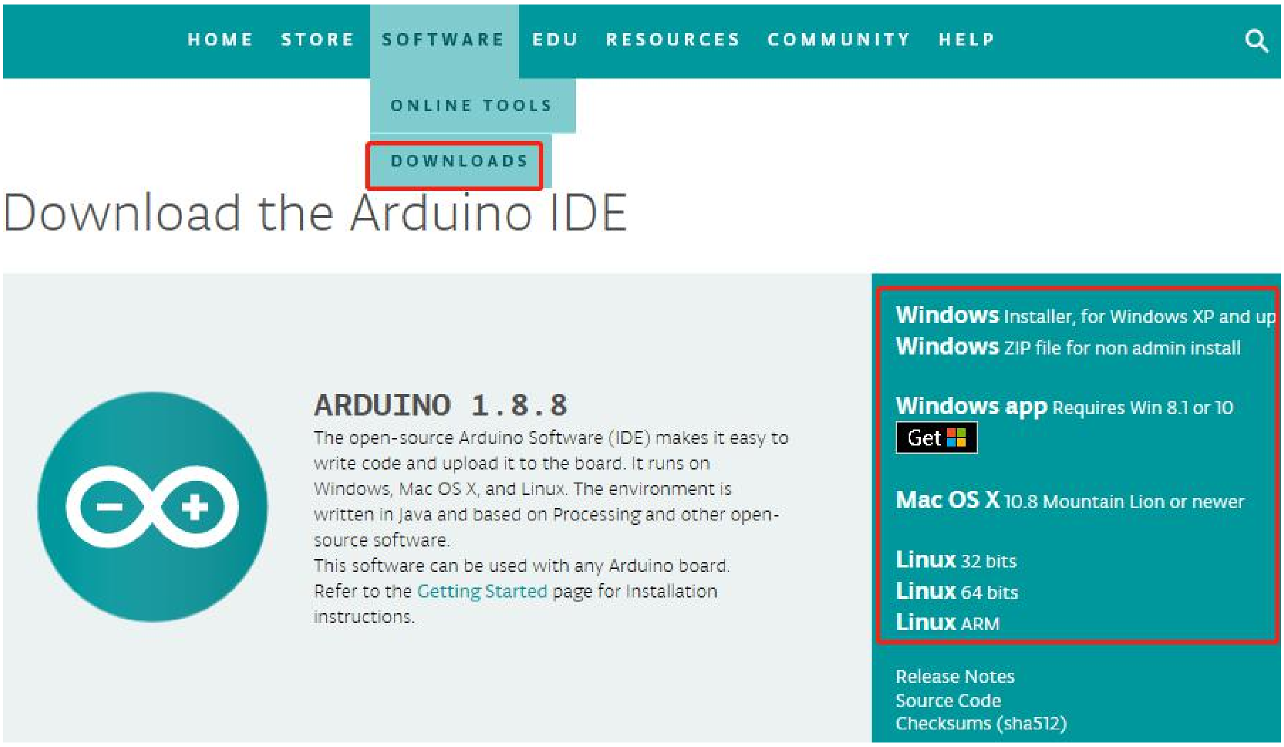
Install Arduino IDE

Here are the installation steps on the windows system.

For other systems, please refer to: **Install Arduinio IDE in different and FAQ.pdf**

The code in this kit is written based on Arduino, so you need to install the IDE first. Skip it if you have done this.

Now go to arduino.cc and click SOFTWARE -> DOWNLOADs. on the page, check the software list on the right side.

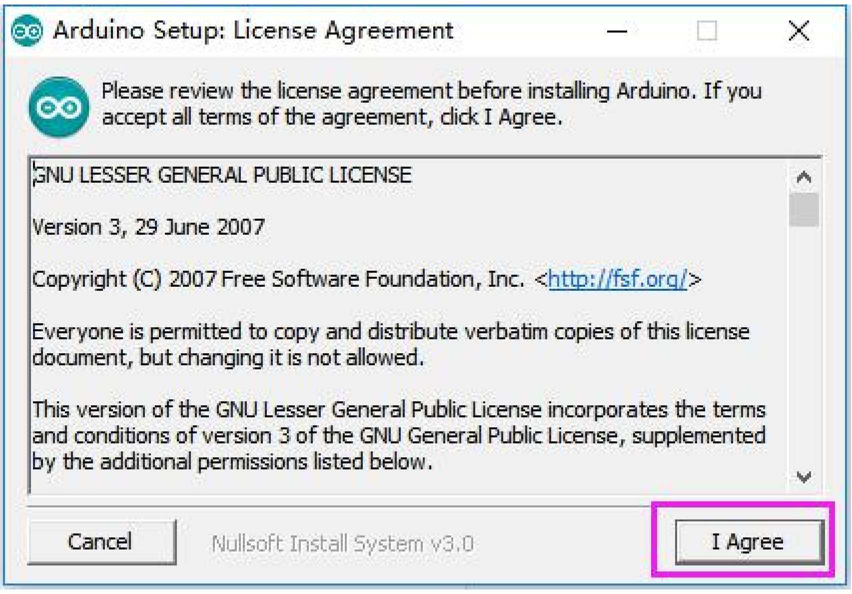


Find the one that suits your operation system and click to download. There are two versions of Arduino for Windows: Installer or ZIP file. You're recommended to download the former.

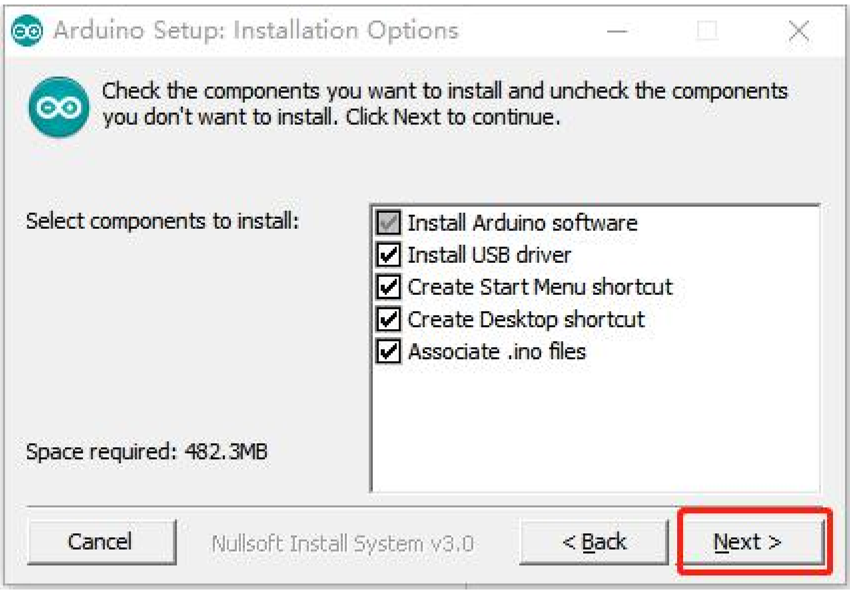
➢ For Installer File

Step 1: Find the .exe file just downloaded.

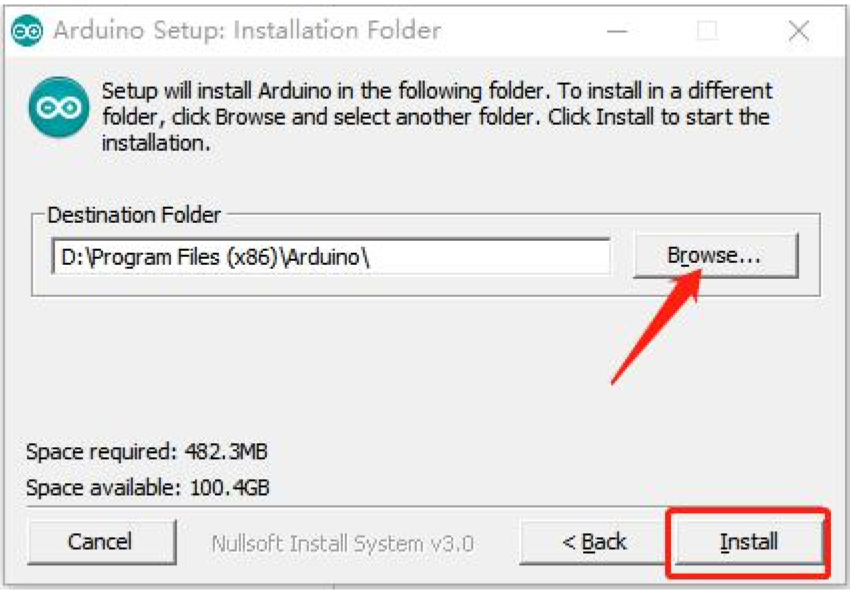
Step 2: Double click the file and a window will pop up as below. Click **I Agree**.



Step 3: Click **Next.**



Step 4: Select the path to install. By default, it's set in the C disk. You can click **Browse** and choose other paths. Click **OK**. Then click Install.



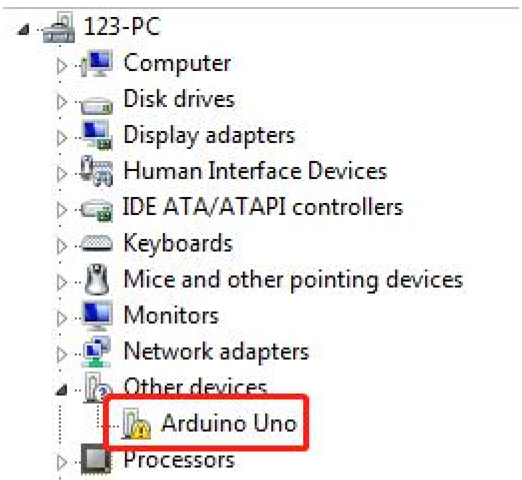
Step 5: Meanwhile, it will prompts install the needed drivers, please select the 'Always trust software from "Arduino LLC" '. After the installation is done, click **Close**.

➢ For ZIP File

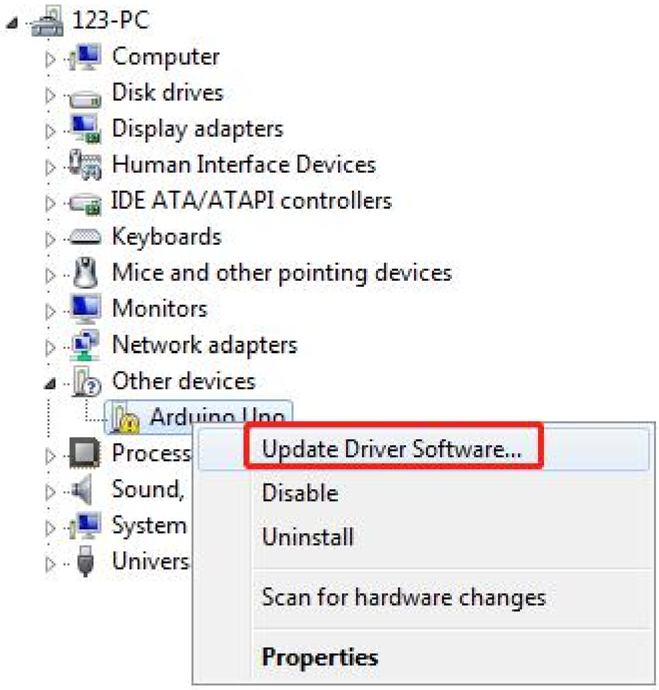
If you download the zip file before, when you connect the MCU to the computer, it may not be recognized. Then you need to install the driver manually. Take the following steps.

Step 1: Plug in the board to the computer with a 5V USB cable. After a while, a prompt message of failed installation will appear.

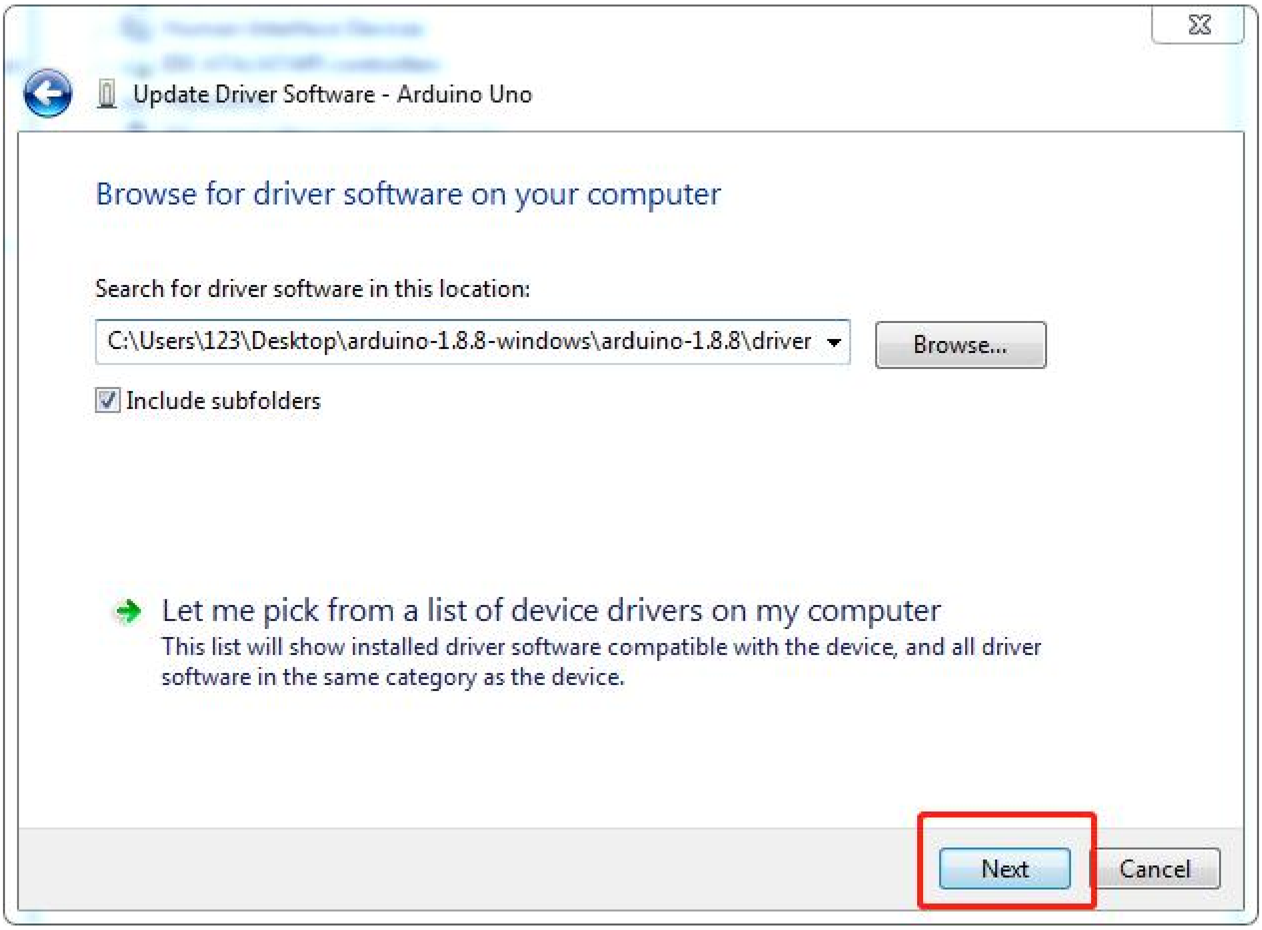
Step 2: Go to the **Device Manager**. You will find under other devices, Arduino Uno with an exclamation mark appear, which means the computer did not recognize the board.



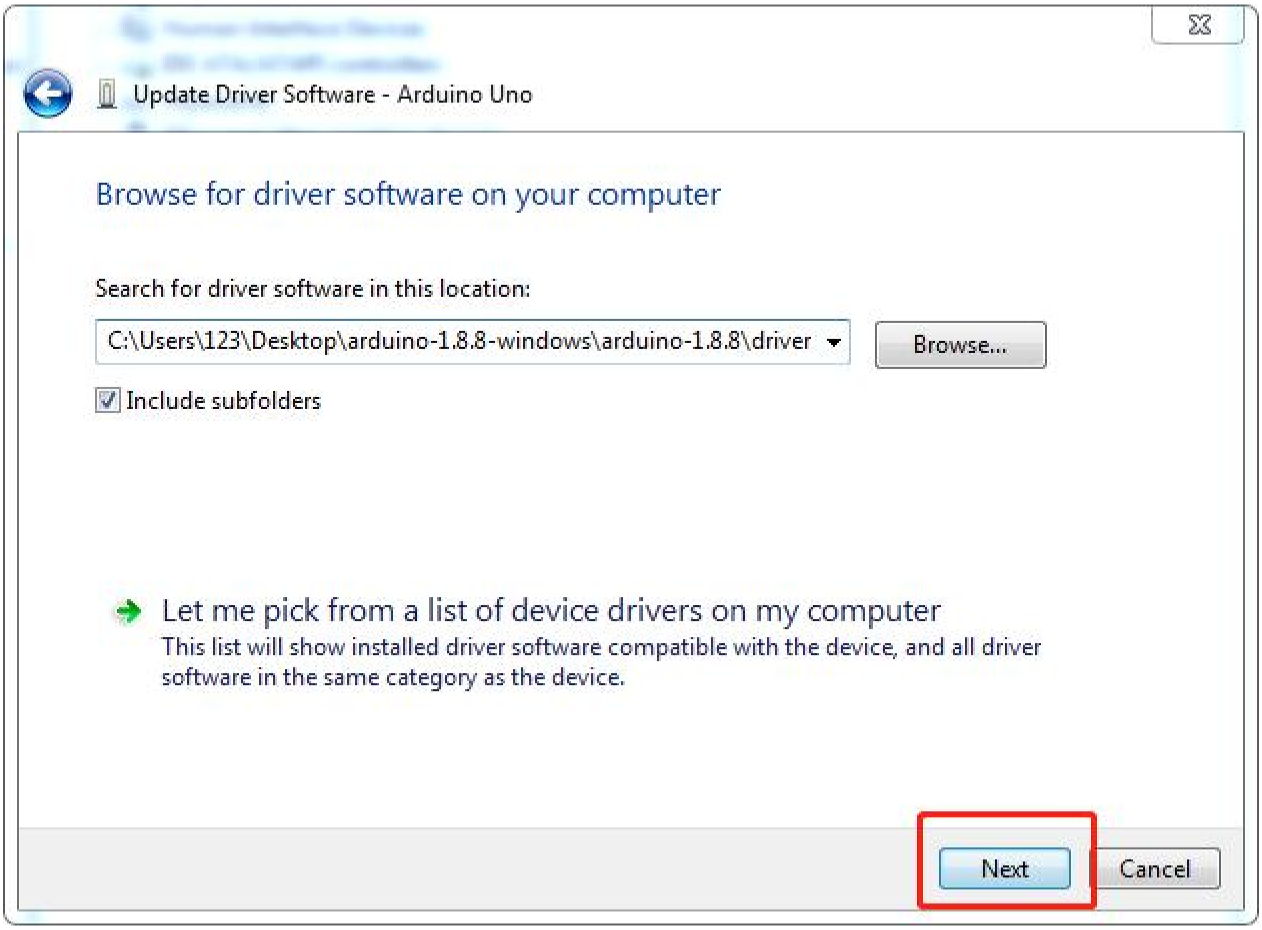
Step 3: Right click on **Arduino Uno** and select **Update Driver Software**.



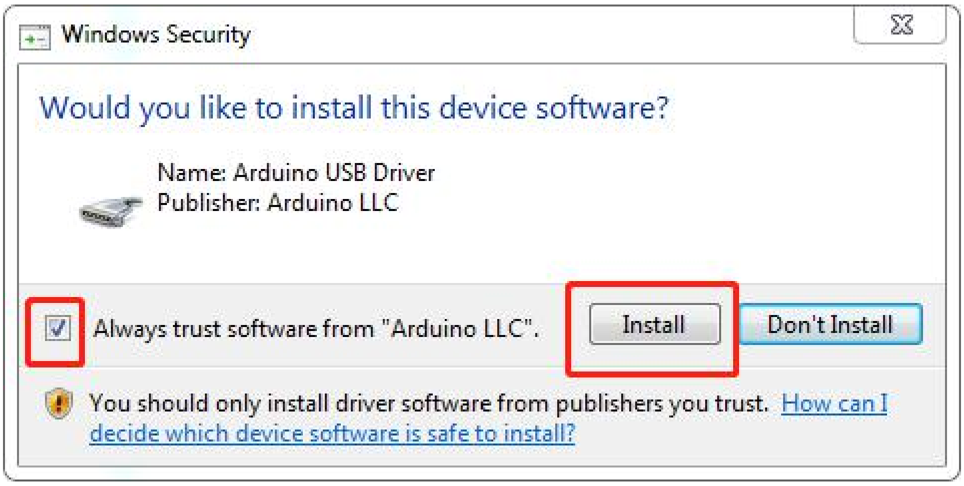
Step 4: Choose the second option, **Browse my computer for Driver software**.



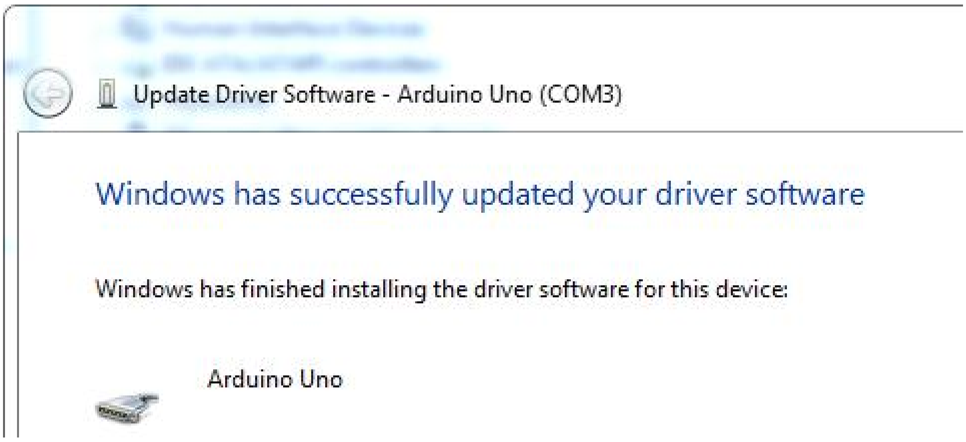
Step 5: A window pops up then. Click **Browse**. Then go to the folder where you just extracted the file. Go to the *drivers* folder and click **OK** -> **Next**.



Step 6: Select 'Always trust software from "Arduino LLC" ' then click Install.



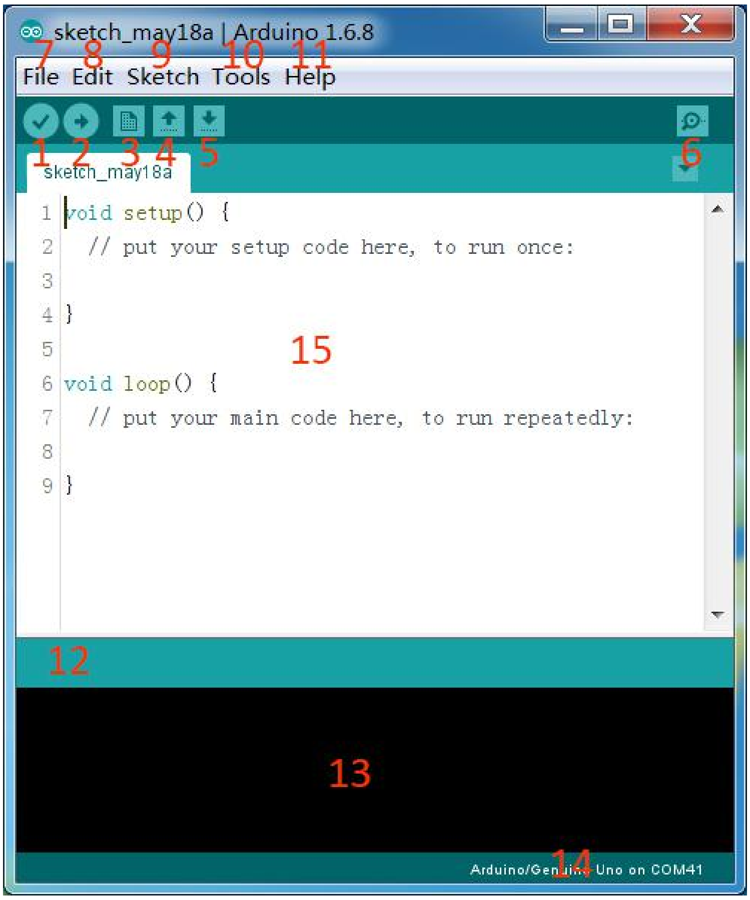
It may need a sec. Then the system prompts you the driver has been installed successfully. So the computer can recognize the board now. Click **Close**.



Open the IDE

Double-click the Arduino icon (arduino.exe) created by the installation process.

Then the Arduino IDE will appear. Let's check details of the software.



1. **Verify**: Compile your code. Any syntax problem will be prompted with errors.

2. **Upload**: Upload the code to your board. When you click the button, the RX and TX LEDs on the board will flicker fast and won't stop until the upload is done.

3. **New**: Create a new code editing window.

4. **Open**: Open an .ino sketch.

5. **Save**: Save the sketch.

6. Serial Monitor: Click the button and a window will appear. It receives the data sent from your control board. It is very useful for debugging.

7. **File**: Click the menu and a drop-down list will appear, including file creating, opening, saving, closing, some parameter configuring, etc.

8. **Edit**: Click the menu. On the drop-down list, there are some editing operations like Cut, Copy, Paste, Find, and so on, with their corresponding shortcuts.

9. **Sketch**: Includes operations like Verify, Upload, Add files, etc. More important function is Include Library – where you can add libraries.

10. **Tool**: Includes some tools – the most frequently used Board (the board you use) and Port (the port your board is at). Every time you want to upload the code, you need to select or check them.

11. **Help**: If you're a beginner, you may check the options under the menu and get the help you need, including operations in IDE, introduction information, troubleshooting, code explanation, etc.

12. In this message area, no matter when you compile or upload, the summary message will always appear.

13. Detailed messages during compile and upload. For example, the file used lies in which path, the details of error prompts.

14. **Board and Port**: Here you can preview the board and port selected for code upload. You can select them again by **Tools** -> **Board** / **Port** if any is incorrect.

15. The editing area of the IDE. You can write code here.

For more details about Arduino IDE, go to **Learning**->**Getting started**->**Foundation** on the arduino.cc and click **Arduino Software (IDE)** on the page http://www.arduino.cc/en/Guide/Environment

If your sketch fails upload, on the same page click **Troubleshooting** http://www.arduino.cc/en/Guide/Troubleshooting.

Notes:

- If your computer is running on the Windows XP system, the new version IDE will prompt errors when running the code. You are recommended to download the Arduino 1.0.5 or Arduino 1.0.6. Or you can also upgrade your Window system.
- All the experiments in this kit are done with SunFounder Uno R3 board, but they are also compatible with SunFounder Mega 2560, SunFounder Mars, SunFounder Nano and all official Arduino Boards. All the code included in this kit works with these boards.

  So what does COMPATIBLE mean here? It means you can use any of the three boards to do the same experiment with the same wiring. Simply put, if the wire is connected to Pin 12 of Uno in the user manual, likewise, you can connect it to Pin 12 on any other official Arduino boards you are using. Then open the corresponding sketch and upload them.

  Now let's begin!

Lesson 1 Blinking LED

Introduction

In this lesson, you will learn how to use the SunFounder Uno board by turning on an LED and making it blink once per second.

Components

- 1 * SunFounder Uno board
- 1 * USB cable
- 1 * Resistor (220Ω)
- 1 * 1 LED
- 1 * Breadboard
- Jumper wires

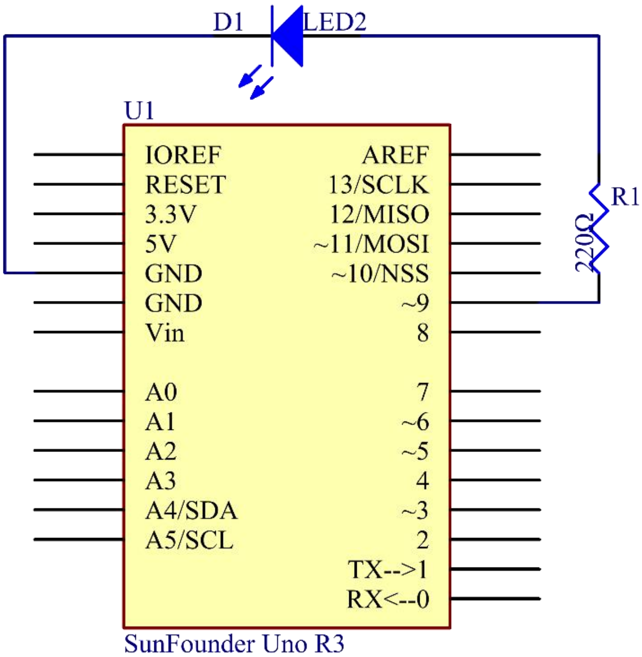
Principle

LED

A light-emitting diode (LED) is a semiconductor device which can turn electric energy into light energy via PN junctions. By wavelength, it can be categorized into laser diode, infrared light-emitting diode and visible light-emitting diode which is usually known as light-emitting diode (LED). LEDs are usually red, yellow, green, blue, or color-changing --- their color varies with different voltages.

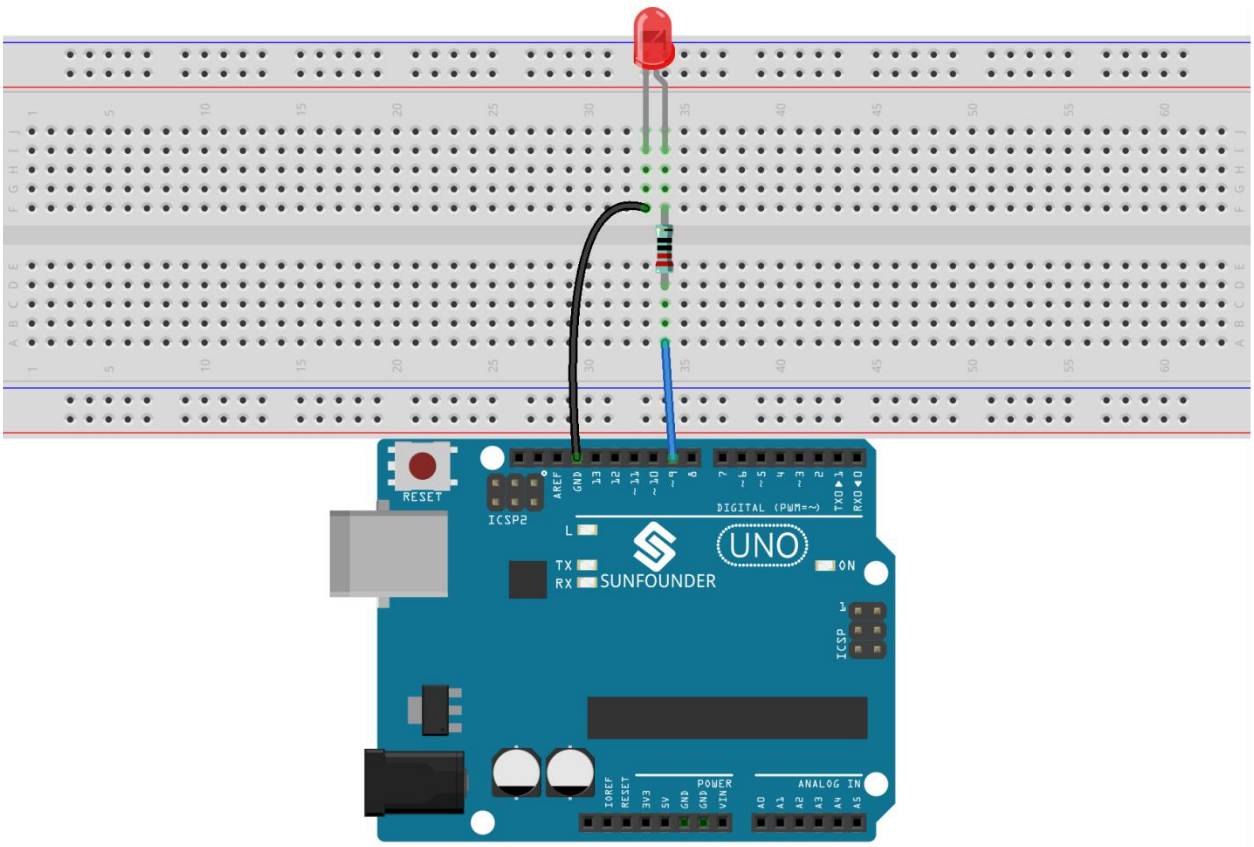
Before building any circuit, you should know the parameters of the components in the circuit, such as their operating voltage, operating circuit, etc. A current-limiting resistor should be connected when an LED is used; or else the LED can be burnt due to excessive currents. In this experiment, the operating voltage of the LED is between 1.5V and 2.0V and the operating current is 10mA - 20mA. The SunFounder Uno board can supply 5V power, the LED we choose works at 1.7V, 15mA. The current-limiting resistance equals total voltage subtracted by LED voltage, then divided by current. In this case, that would be (5-1.7)/0.015. Thus, the current-limiting resistance equals 220Ω.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.



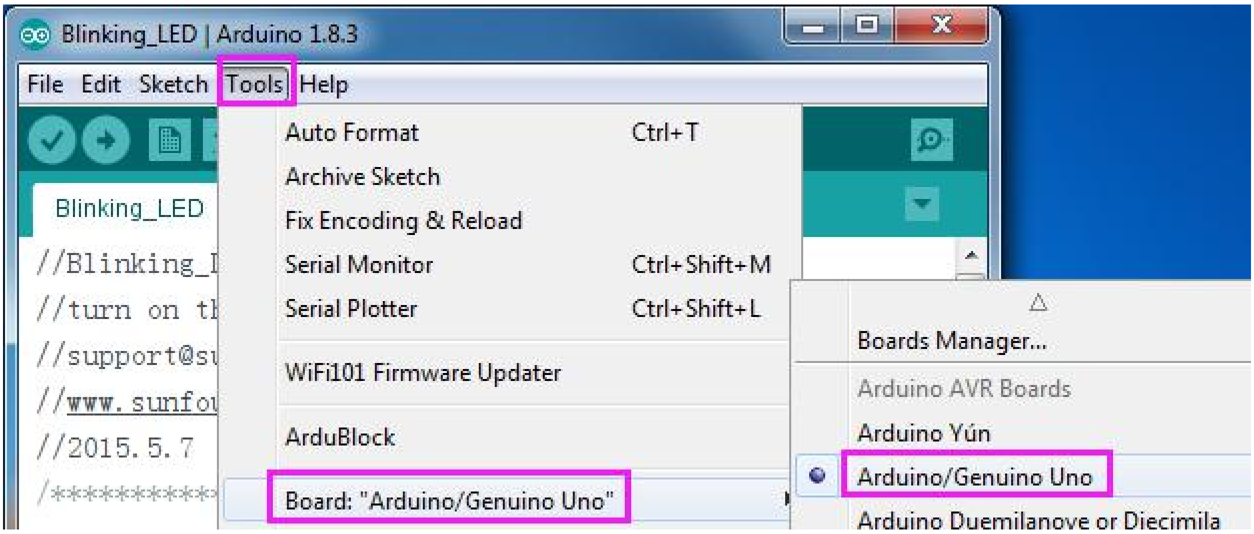
Step 2: Open the code file.

In the folder ***Super Kit V2.0 for Arduino*** you just downloaded and unzipped, you can see the folder for the corresponding lesson which includes two folders: *code* (the sketch) and *circuit* (Fritzing file). Go to the *code* folder and find the *.ino* file (.ino is the format of the sketch). Then double click to open it. You can also open an empty *.ino* file and type in the code we provide.

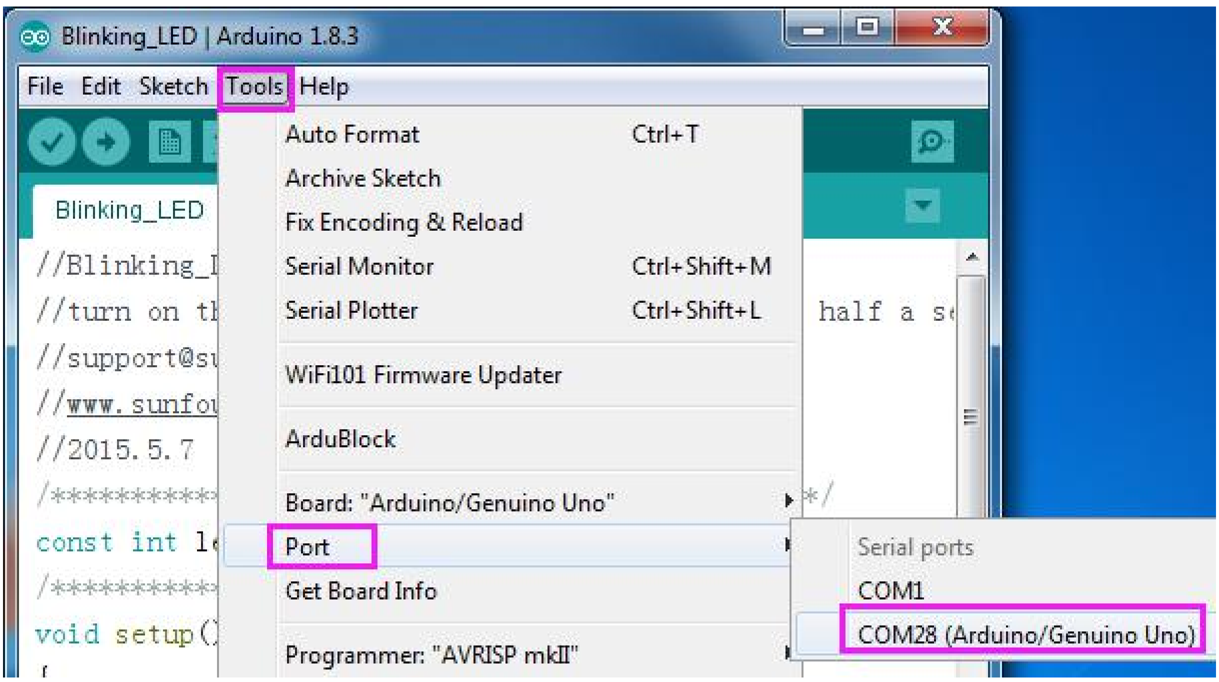
Step 3: Select the Board and Port.

Before uploading the code, you need to select the **Board** and **Port**.

Click **Tools** ->**Board** and select **Arduino/Genuino Uno**. If your board is Mega2560, then select **Arduino/Genuino Uno Mega** or **Mega2560**. If it's Nano, select **Arduino Nano**.

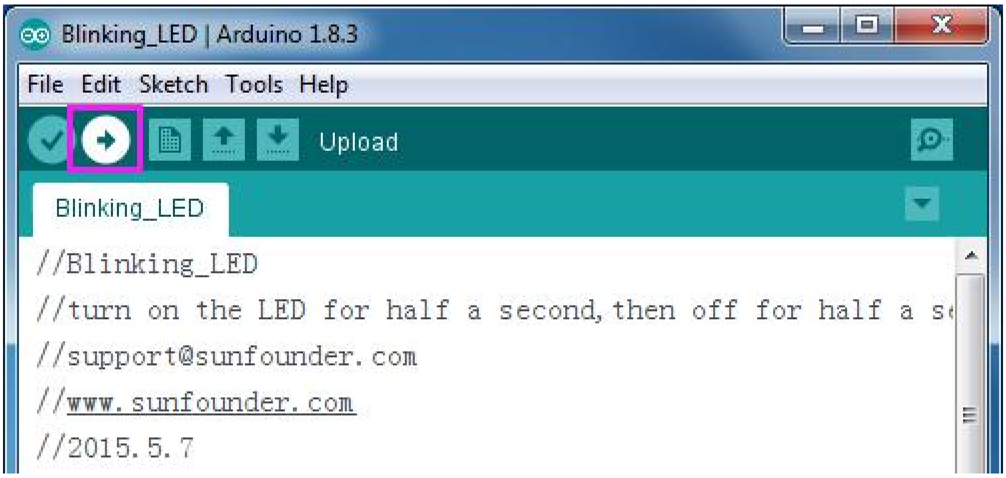


Then select **Tools** ->**Port**. Your port should be different from mine.

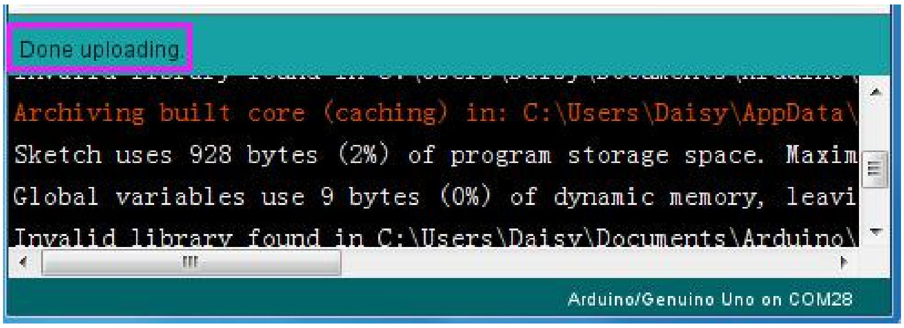


Step 4: Upload the sketch to the SunFounder Uno board.

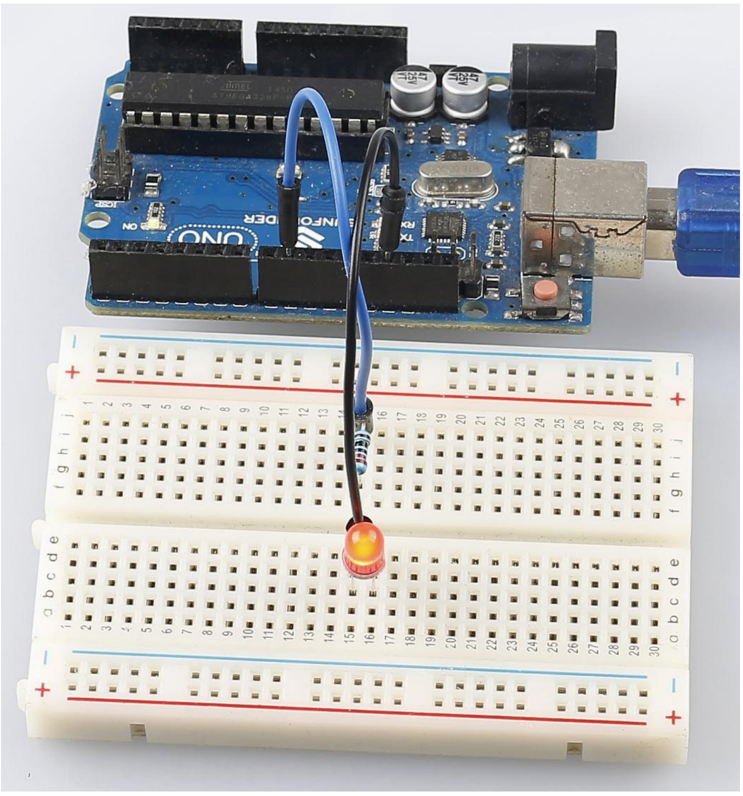
Click the **Upload** icon to upload the code to the control board.



If "Done uploading" appears at the bottom of the window, it means the sketch has been successfully uploaded.



You should now see the LED blinking.



Experimental Summary

Through this experiment, you have learned how to turn on an LED. You can also change the blinking frequency of the LED by changing the num value in the delay function delay (num). For example, change it to delay (250) and you will find that the LED blinks more quickly.

Tips: For any **TECHNICAL** questions, add a topic under **FORUM** section on our website www.sunfounder.com and we'll reply as soon as possible. For **NON-TECH** questions like order issues, please **email** service@sunfounder.com.

Lesson 2 Controlling an LED by Button

Introduction

In this experiment, you will learn how to turn on/off an LED by using an I/O port and a button. The "I/O port" refers to the INPUT and OUTPUT port. Here the INPUT port of the SunFounder Uno board is used to read the output of an external device. Since the board itself has an LED (connected to Pin 13), you can use this LED to do this experiment for convenience.

Components

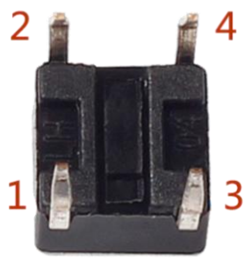
- 1 * SunFounder Uno board
- 1 * USB cable
- 1 * Button
- 1 * Resistor (10kΩ)
- 1 * Breadboard
- Jumper wires

Principle

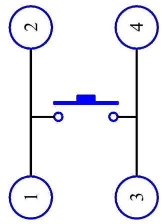
Button

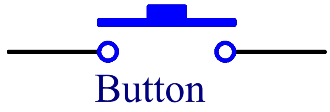
Buttons are a common component used to control electronic devices. They are usually used as switches to connect or break circuits. Although buttons come in a variety of sizes and shapes, the one used here is a 6mm mini-button as shown in the following pictures.

Pin 1 is connected to pin 2 and pin 3 to pin 4. So you just need to connect either of pin 1 and pin 2 to pin 3 or pin 4.



The following is the internal structure of a button. The symbol on the right below is usually used to represent a button in circuits.



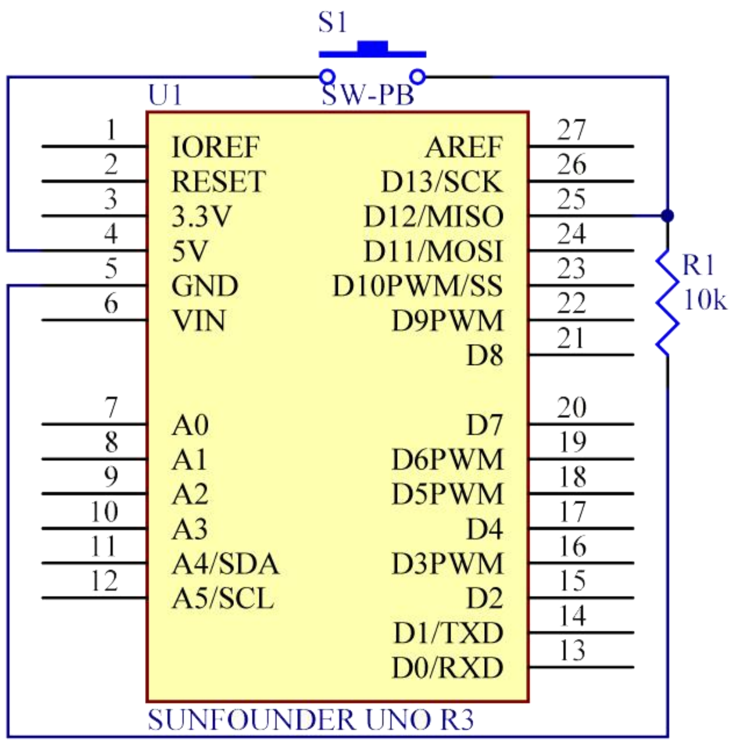


Since the pin 1 is connected to pin 2, and pin 3 to pin 4, when the button is pressed, the 4 pins are connected, thus closing the circuit.

Generally, the button can be connected directly to the LED in a circuit to turn on or off the LED, which is comparatively simple. However, sometimes the LED will brighten automatically without any button pressed, which is caused by various kinds of external interference. In order to avoid this interference, a pull-down resistor is used – usually connect a 1K–10KΩ resistor between the button and GND. It can be connected to GND to consume the interference when the button is off.

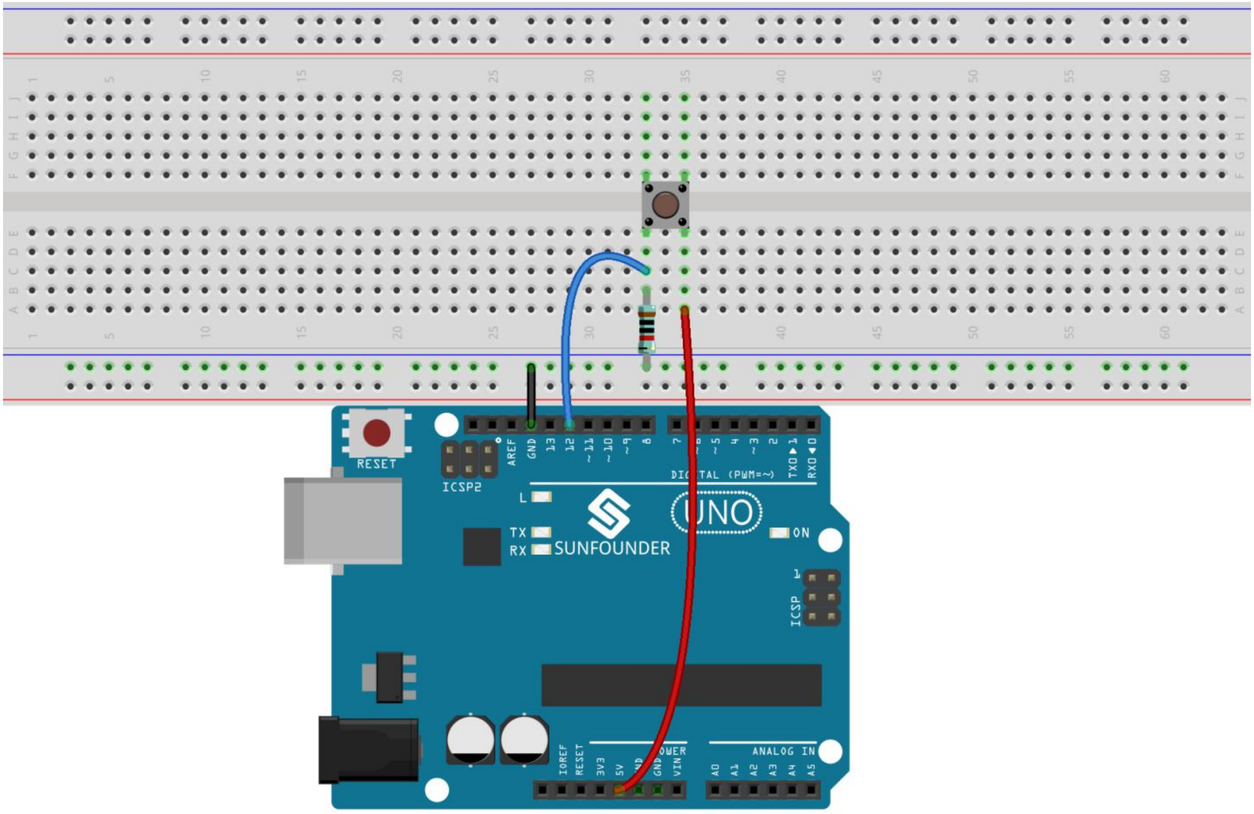
This circuit connection is widely used in numerous circuits and electronic devices. We may use the button to control a circuit later in many experiments (in or outside this kit maybe), so you might get its principle, which is very simple, and application at the beginning of your study.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

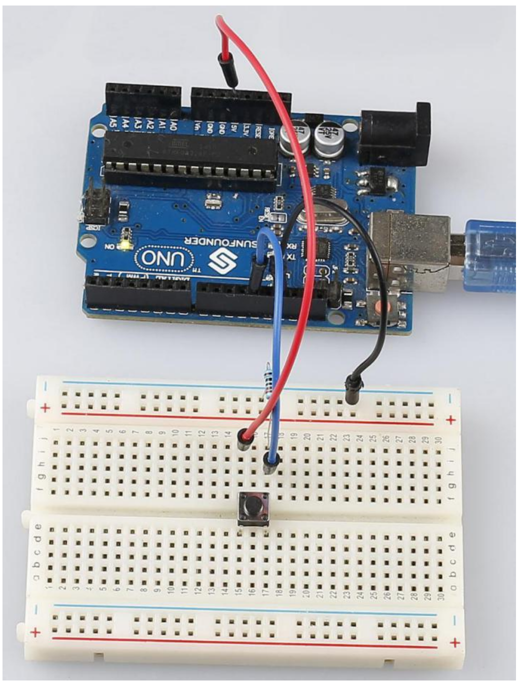


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Now, press the button, and the LED on the SunFounder Uno board will light up.



Lesson 3 Controlling an LED by PWM

Introduction

In this lesson, let's try something interesting – gradually changing the luminance of an LED through programming. Since the pulsing light looks like breathing, we give it a magical name - breathing LED. We'll accomplish this effect with pulse width modulation (PWM).

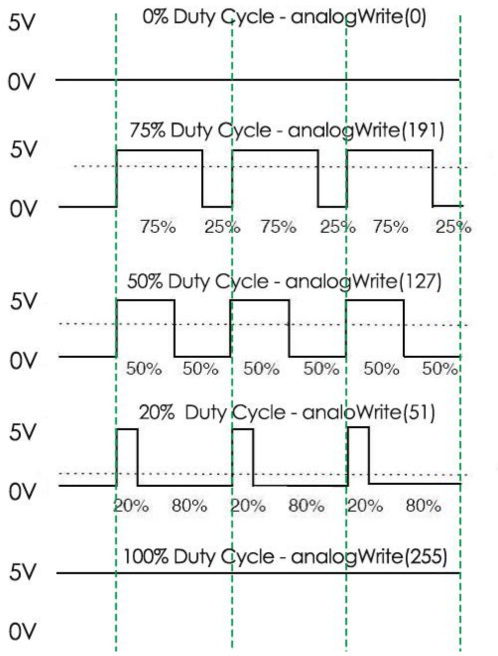
Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * LED
- 1 * Resistor (220Ω)
- 1 * USB cable
- Jumper wires

Principle

Pulse width modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called pulse width. To get varying analog values, you change, or modulate, that width. If you repeat this on-off pattern fast enough with some device, an LED for example, it would be like this: the signal is a steady voltage between 0 and 5V controlling the brightness of the LED. (See the PWM description on the official website of Arduino).

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each.

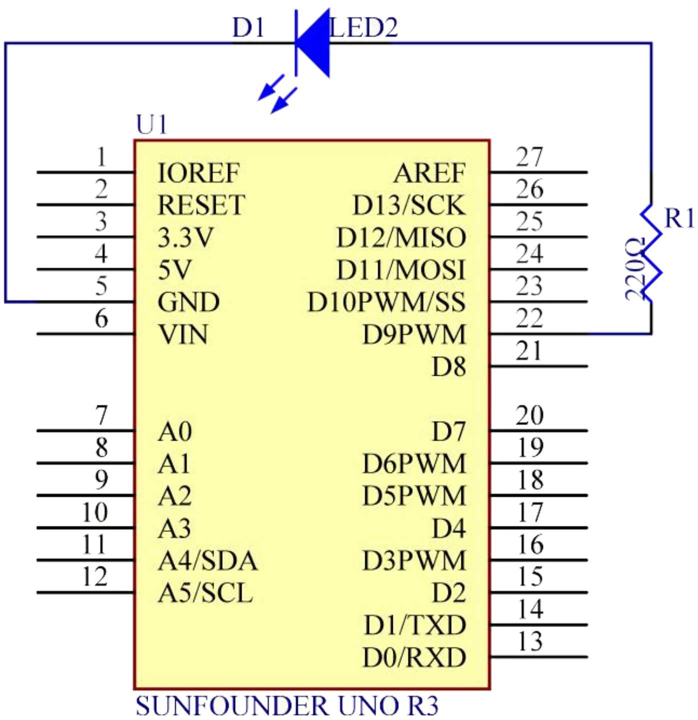


A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

You will find that the smaller the PWM value is, the smaller the value will be after being converted into voltage. Then the LED becomes dimmer accordingly. Therefore, we can control the brightness of the LED by controlling the PWM value.

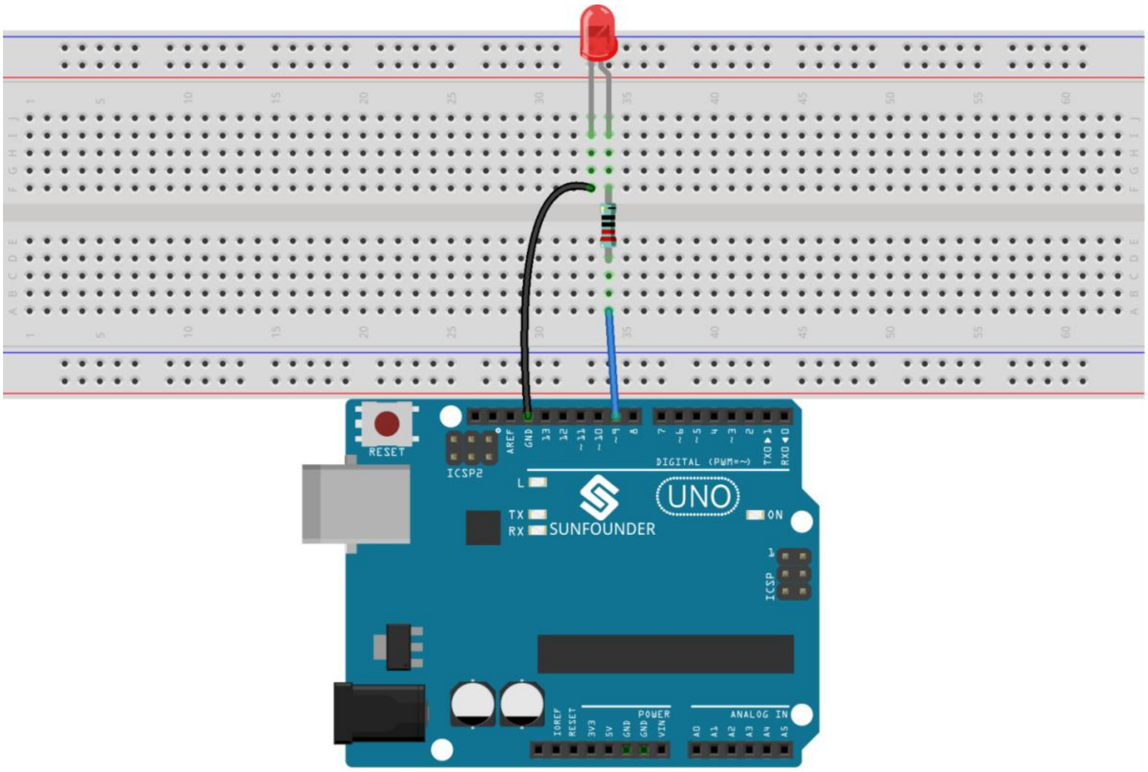
In this experiment, we use this technology to make the LED brighten and dim slowly so it looks like our breath.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

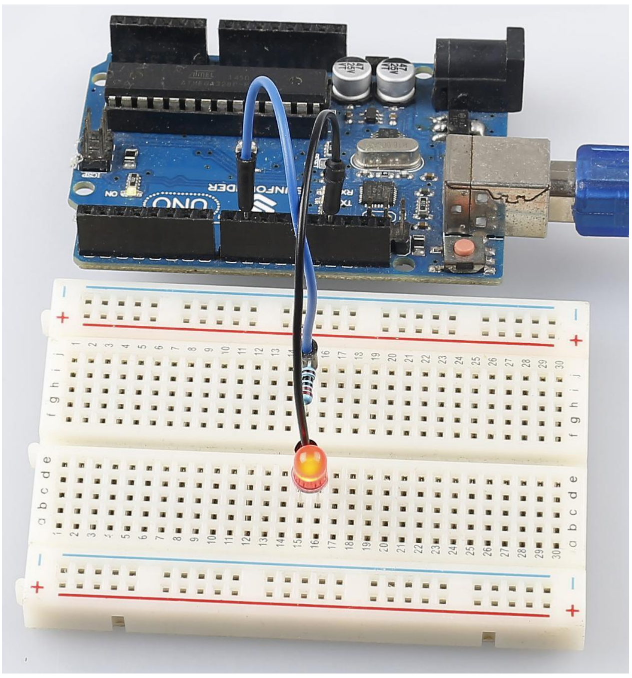


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Here you should see the LED gets brighter and brighter, then slowly dimmer, and again brighter and dimmer repeatedly, just like breathing.



Lesson 4 Controlling an LED by Potentiometer

Introduction

In the previous experiment, you have learned how to control an LED by PWM programming, which is interesting though sounds slightly abstract. In this lesson, let's see how to change the luminance of an LED by a potentiometer.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * Resistor (220Ω)
- 1 * LED
- 1 * Potentiometer
- 1 * USB cable
- Jumper wires

Principle

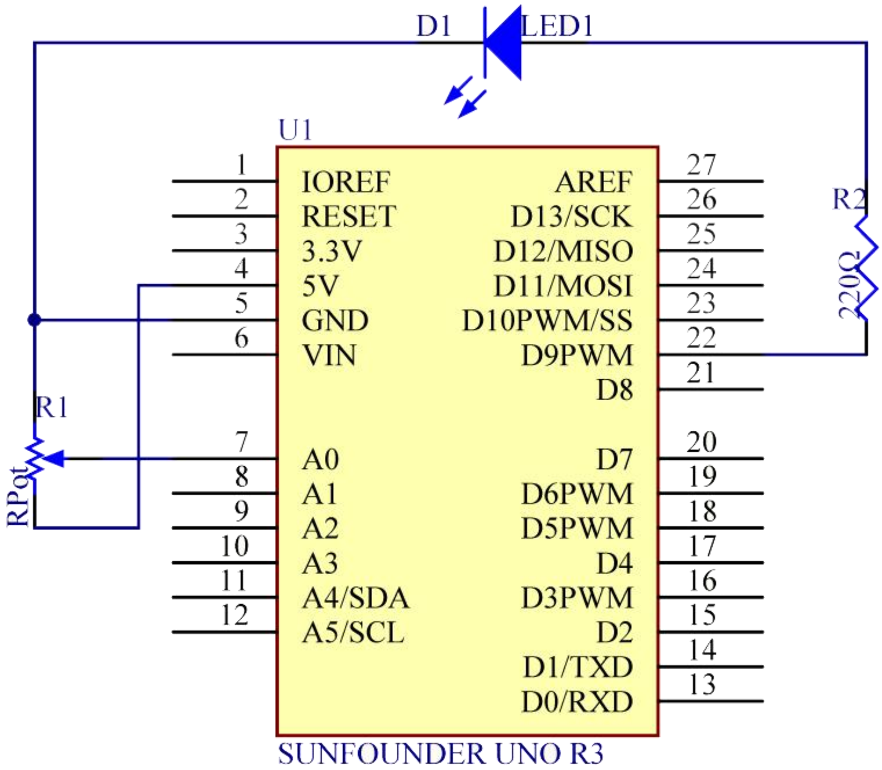
Analog V.S. Digital

A linear potentiometer is an analog electronic component. So what's the difference between an analog value and a digital one? Simply put, digital means on/off, high/low level with just two states, i.e. either 0 or 1. But the data state of analog signals is linear, for example, from 1 to 1000; the signal value changes over time instead of indicating an exact number. Analog signals include those of light intensity, humidity, temperature, and so on.

In this experiment, a potentiometer, or pot, is used to change the current in the circuit so the luminance of the LED will change accordingly. And since the pot is an analog device, the current change is smooth, thus the LED will gradually get brighter or dimmer instead of going through an obvious stepwise process.

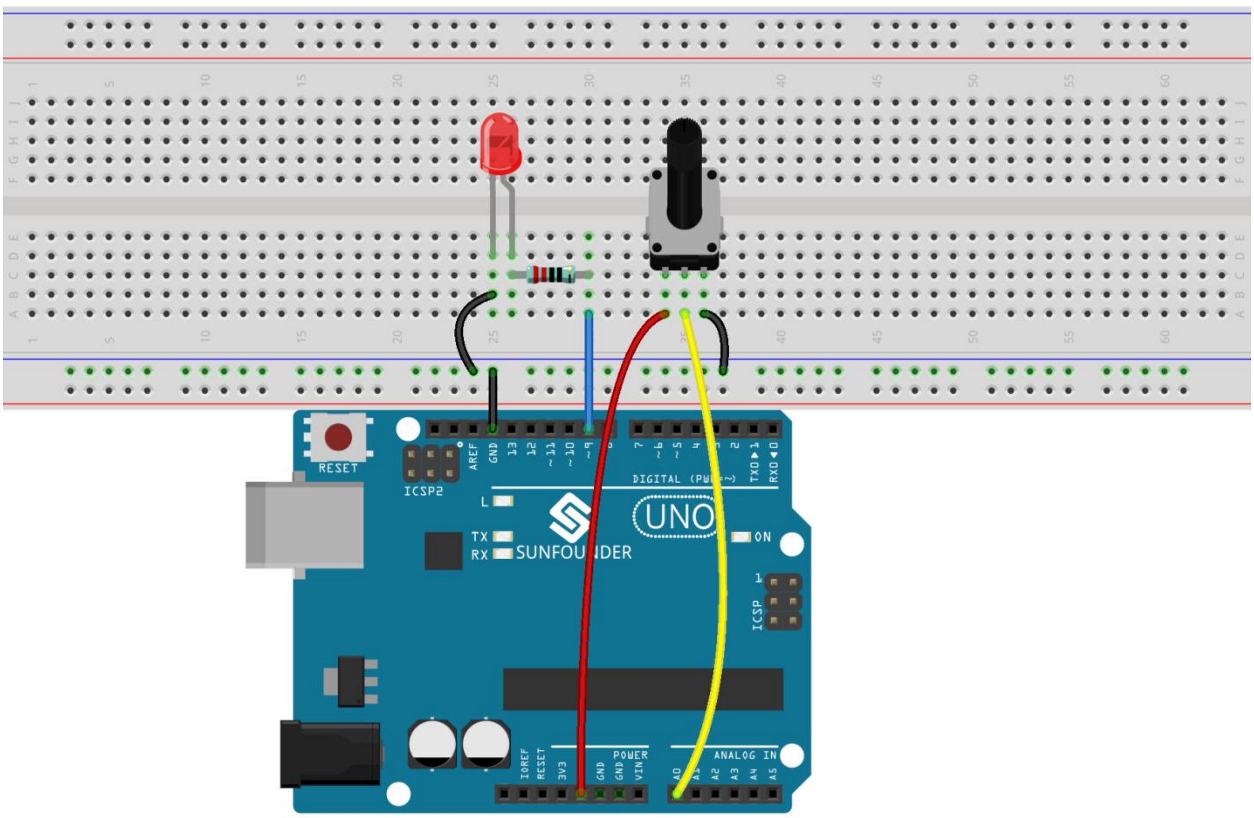
What we mean by PWM here is the digitalization of analog signals, which is a process of approaching analog signals. Since the potentiometer inputs analog signals, it should be connected to analog ports, i.e. A0-A5 of the Uno board, instead of digital ports.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.



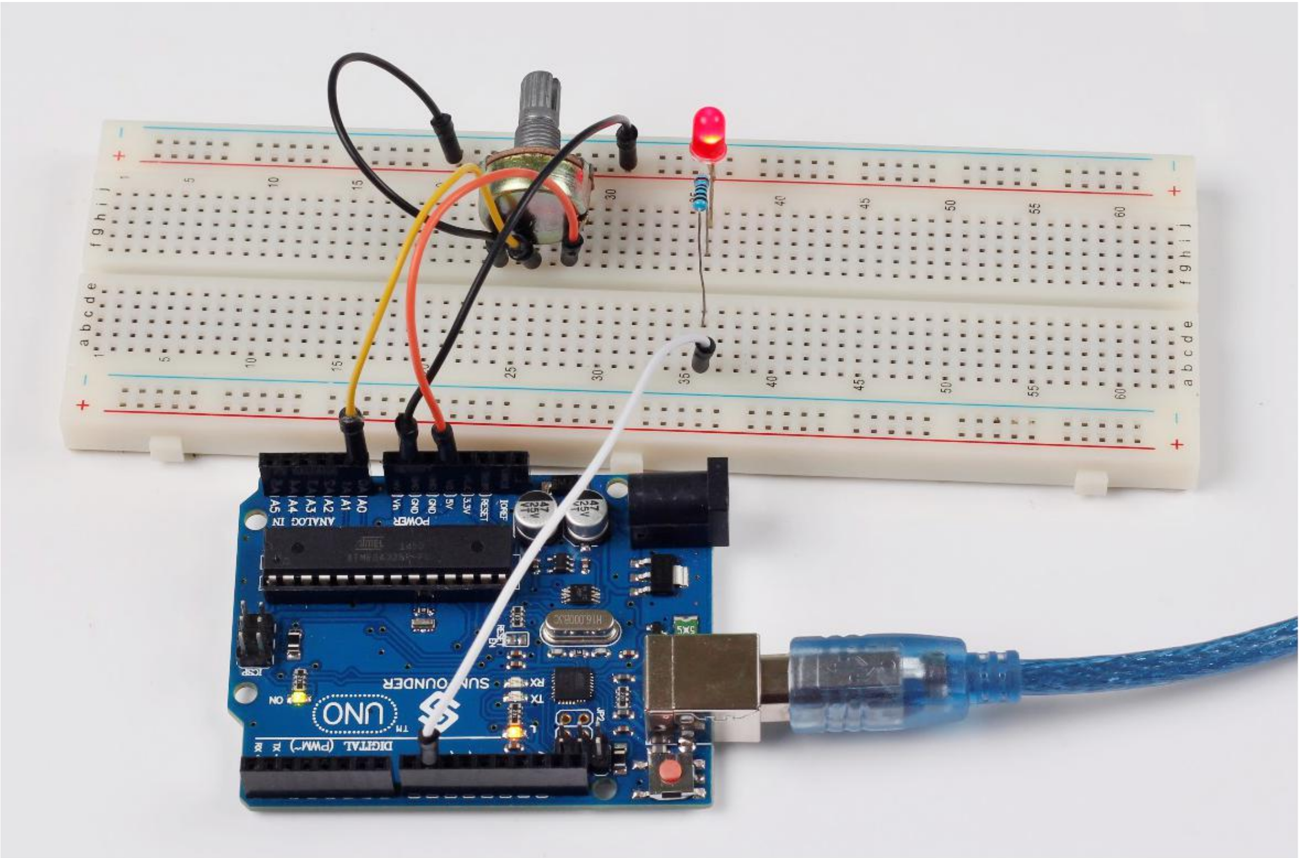
As you can see, the potentiometer is connected to pin A0 of the SunFounder Uno board, which can measure voltages from 0V to 5V. The corresponding value returned is from 0 to 1024. The measurement accuracy for voltage change is relatively high.

Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Spin the shaft of the potentiometer and you should see the luminance of the LED change.



Lesson 5 Flowing LED Lights

Introduction

In this lesson, we will conduct a simple yet interesting experiment – using LEDs to create flowing LED lights. As the name suggests, these eight LEDs in a row successively light up and dim one after another, just like flowing water.

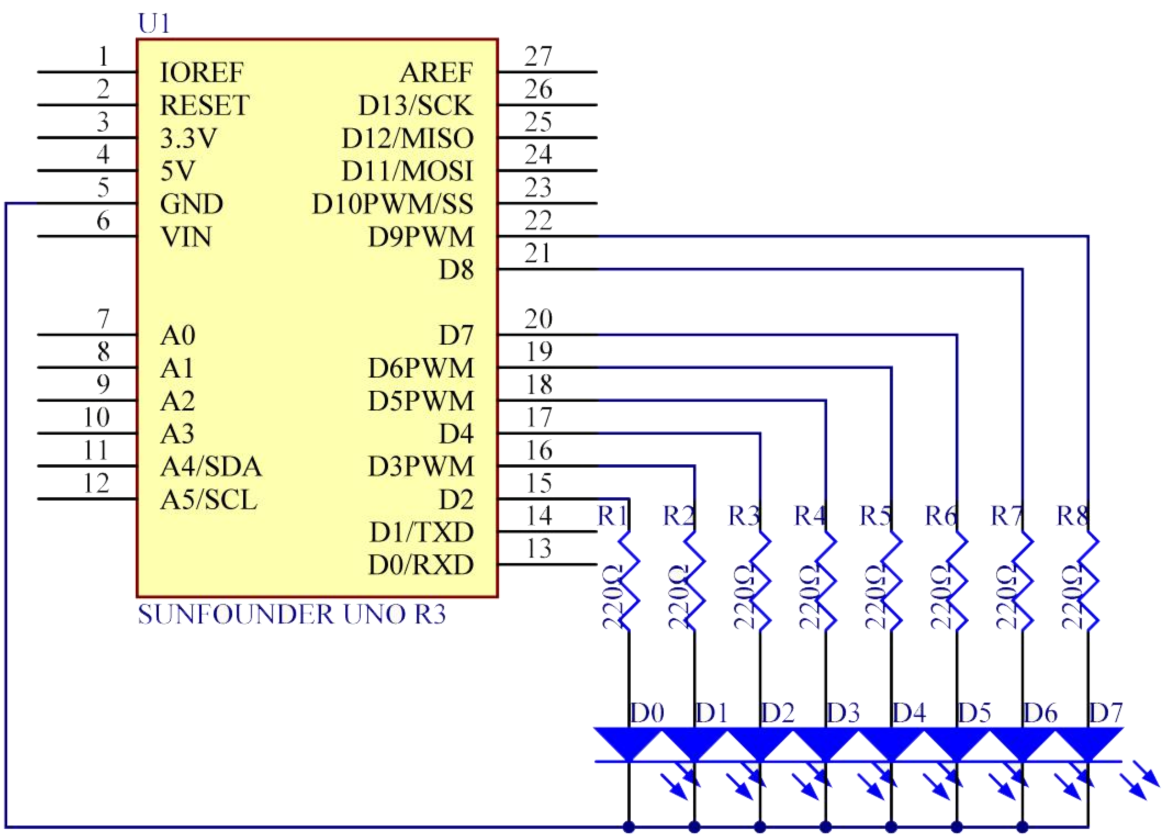
Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 8 * LED
- 8 * Resistor (220Ω)
- 1 * USB cable
- Jumper wires

Principle

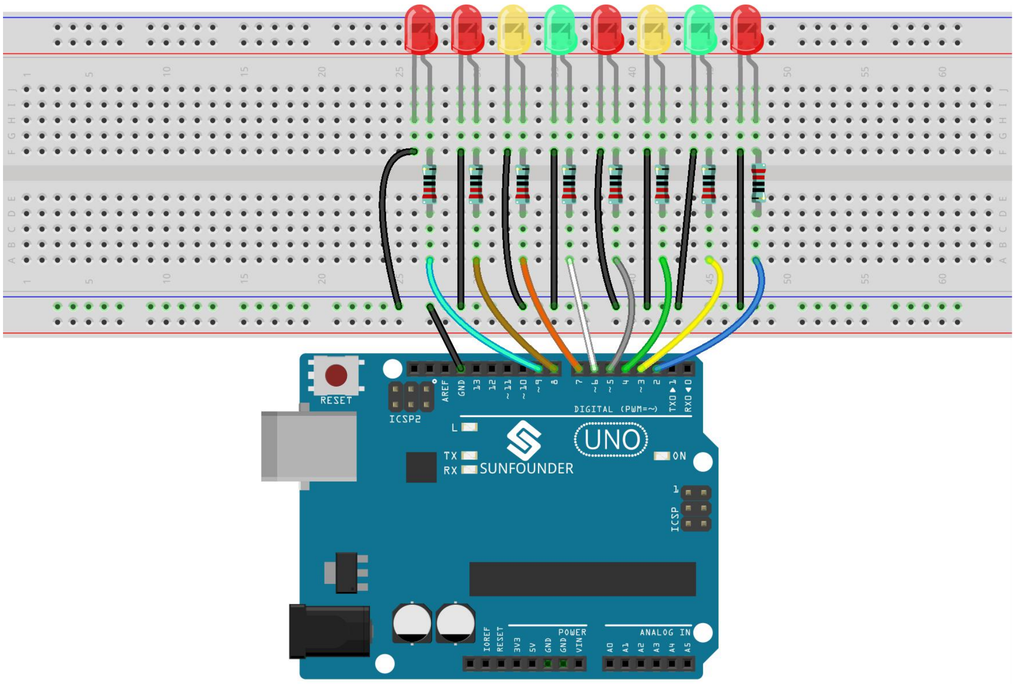
The principle of this experiment is simply to turn on eight LEDs in turn.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

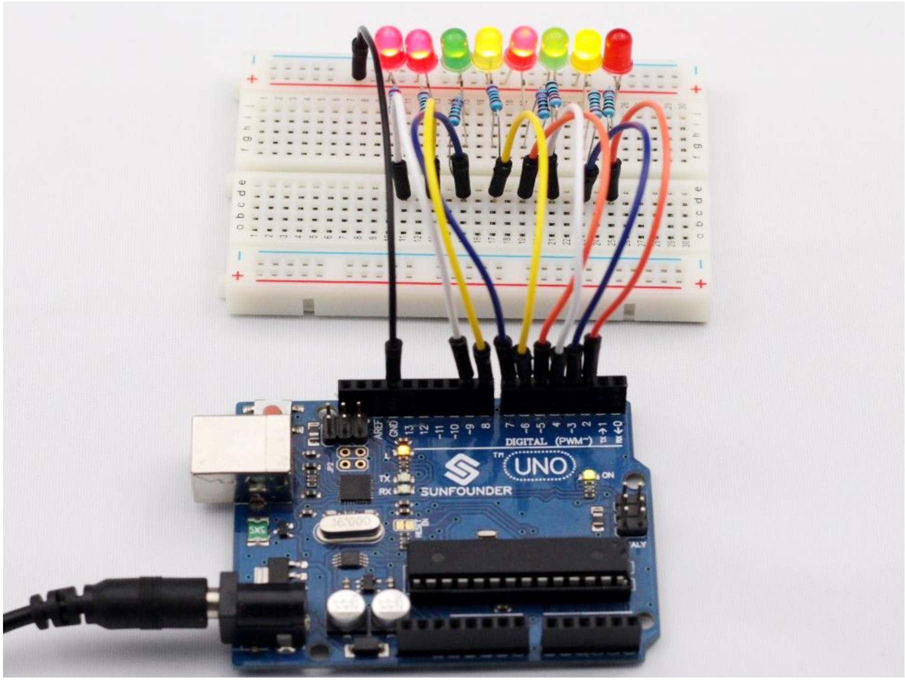


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Now, you should see eight LEDs brighten one by one from left to right, and then dim in turn from right to left. After that, the LEDs will light up from right to left and dim from left to right. This whole process will repeat until the circuit is power off.



Lesson 6 RGB LED

Introduction

Previously we've used the PWM technology to control an LED brighten and dim. In this lesson, we will use it to control an RGB LED to flash various kinds of color.

Components

- 1 * RGB LED
- 3 * Resistor (220Ω)
- 1 * Breadboard
- 1 * SunFounder Uno board
- 1 * USB cable
- Jumper wires

Principle

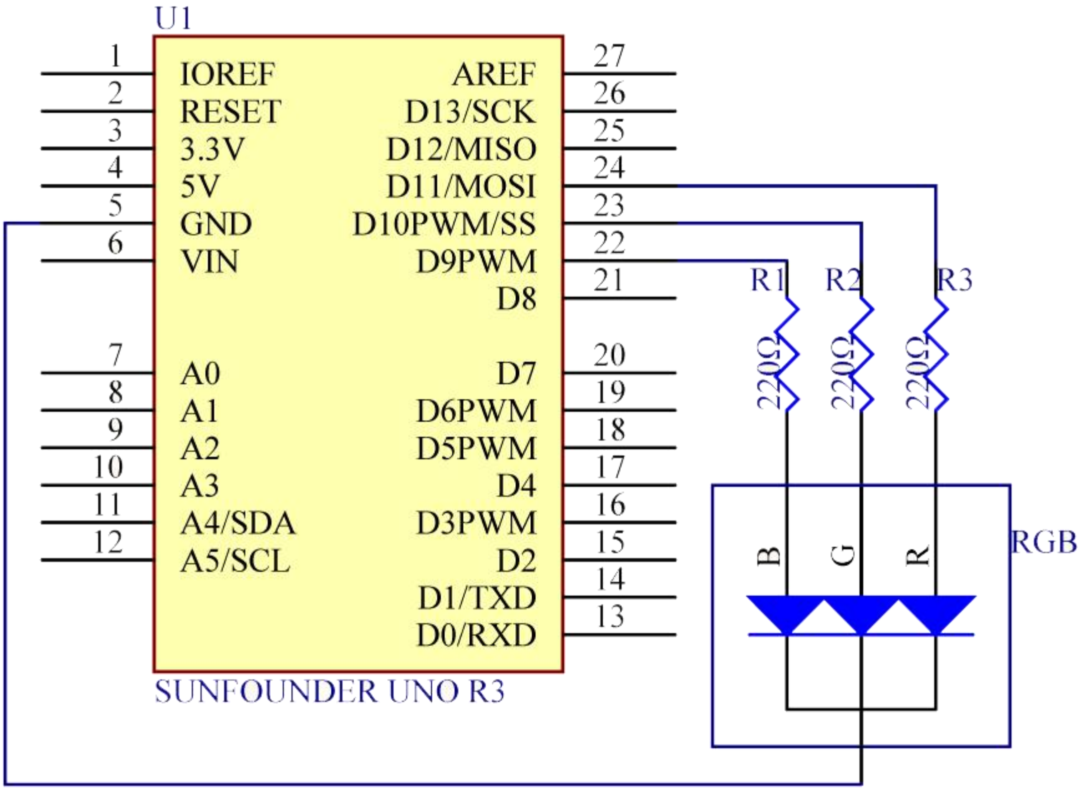
RGB

RGB stands for the red, green, and blue color channels and is an industry color standard. RGB displays various new colors by changing the three channels and superimposing them, which, according to statistics, can create 16,777,216 different colors.

Each of the three color channels has 255 stages of brightness. When the three primary colors are all 0, it is the least bright, thus turning it off. When the three colors are all 255, which is the brightest, the LEDs will brighten. When the light emitted of the three colors are mixed together, the colors will be mixed too. However, the brightness is equal to the sum of all brightness. And the more you mix, the brighter the LED gets. This process is known as additive mixing.

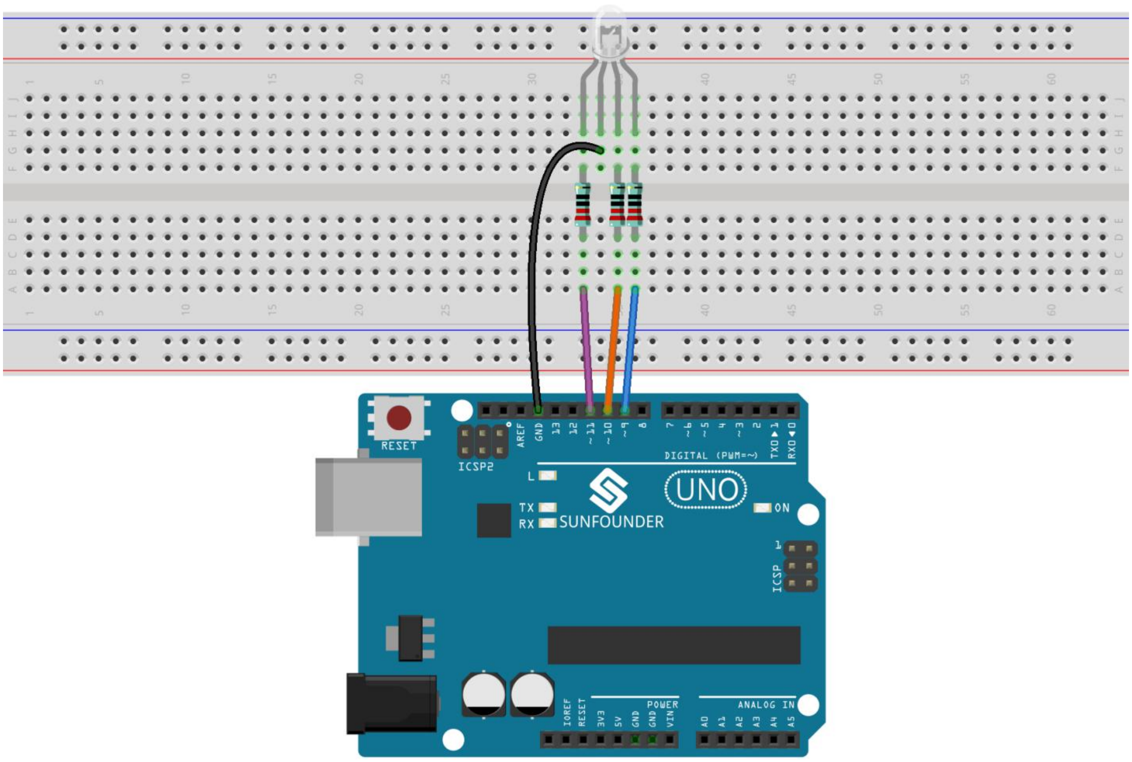
In this experiment, we will also use PWM which, if you've followed the lessons thus far, you already have a basic understanding of. Here we input a value between 0 and 255 to the three pins of the RGB LED to make it display different colors.

The Schematic Diagram



Experimental Procedure

Step 1: Build the circuit.

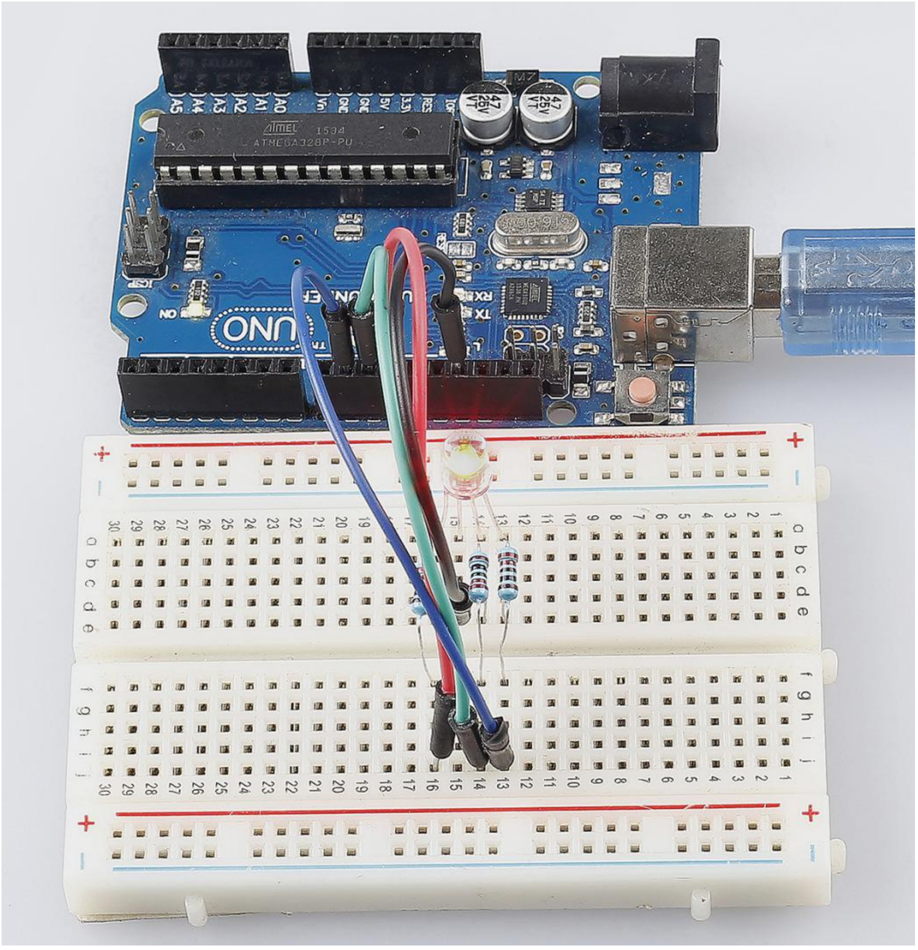


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Here you should see the RGB LED flash circularly red, green, and blue first, then red, orange, yellow, green, blue, indigo, and purple.



Lesson 7 DC Motor Control

Introduction

In this experiment, we will learn how to control the direction and speed of a small-sized DC motor by a driver chip L293D and the SunFounder Uno board. Making simple experiments, we will just make the motor rotate left and right, and accelerate or decelerate automatically.

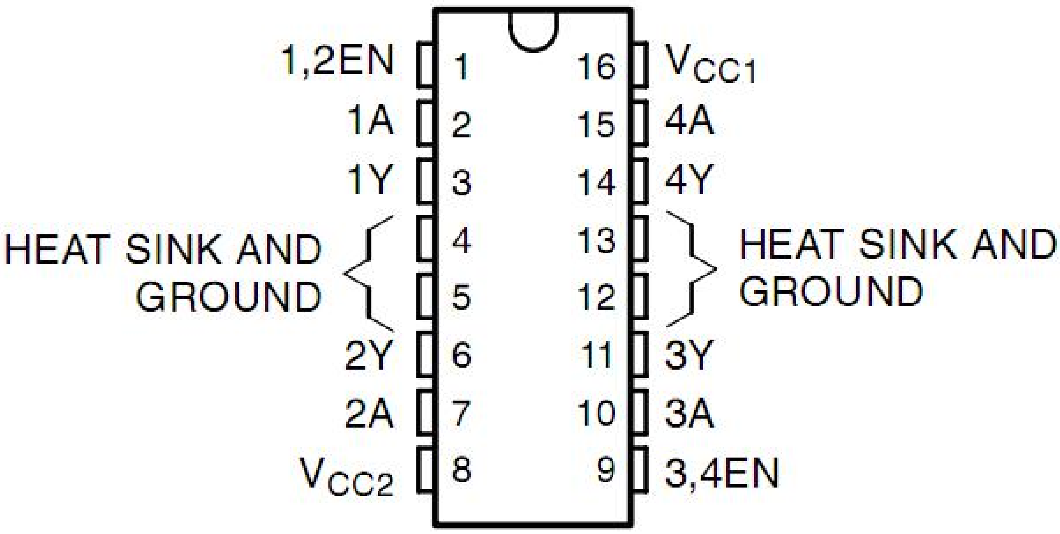
Components

- 1 * Small-sized DC motor
- 1 * L293D
- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- Jumper wires

Principle

L293D

This is a very practical chip that can independently control two DC motors. In this experiment, just about half of the chip is used – most pins on the right side of the chip are used to control a second motor, which is not needed since we apply only one motor here.



L293D has two pins (Vcc1 and Vcc2) for power supply. Vcc2 is used to supply power for the motor, while Vcc1 to supply for the chip. Since a small-sized DC motor is used here, connect both pins to +5V. If you use a higher power motor, you need to connect Vcc2 to an external power supply. At the same time, the GND of L293D should be connected to that of the SunFounder Uno board.

DC Motor Specifications

Voltage: 3-6V

Main Size: length 25mm, thickness 15mm, width 20mm

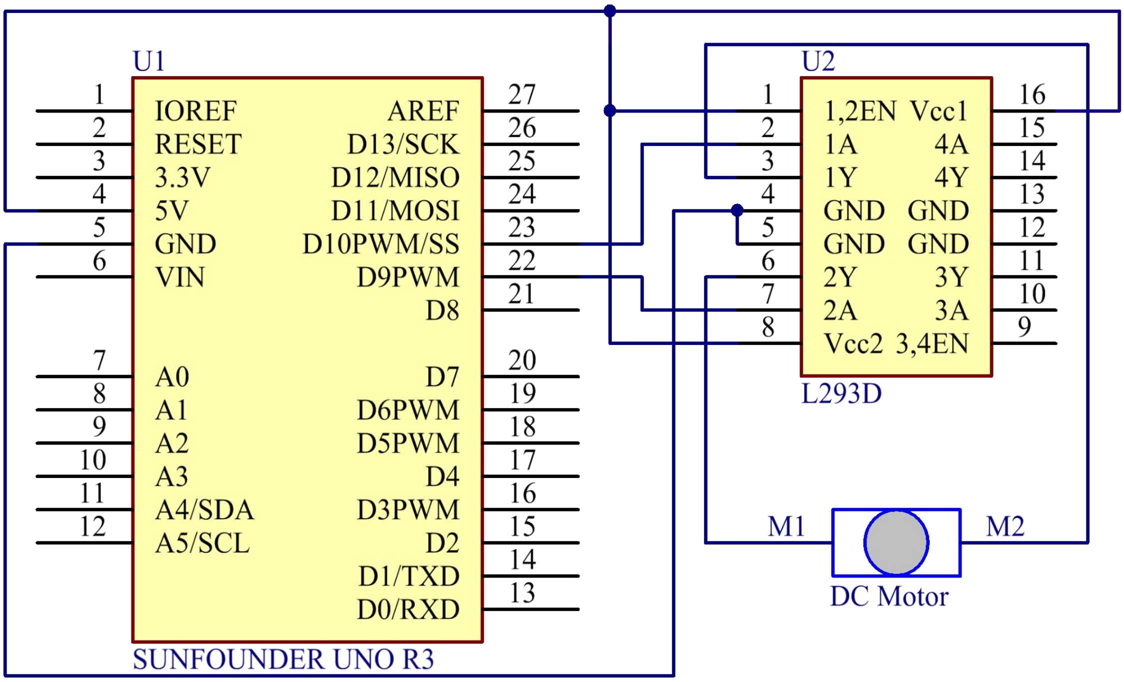
Motor Shaft Length: 9mm, Shaft Diameter 2mm

Rated Voltage: 3v

Reference Current: 0.35-0.4A

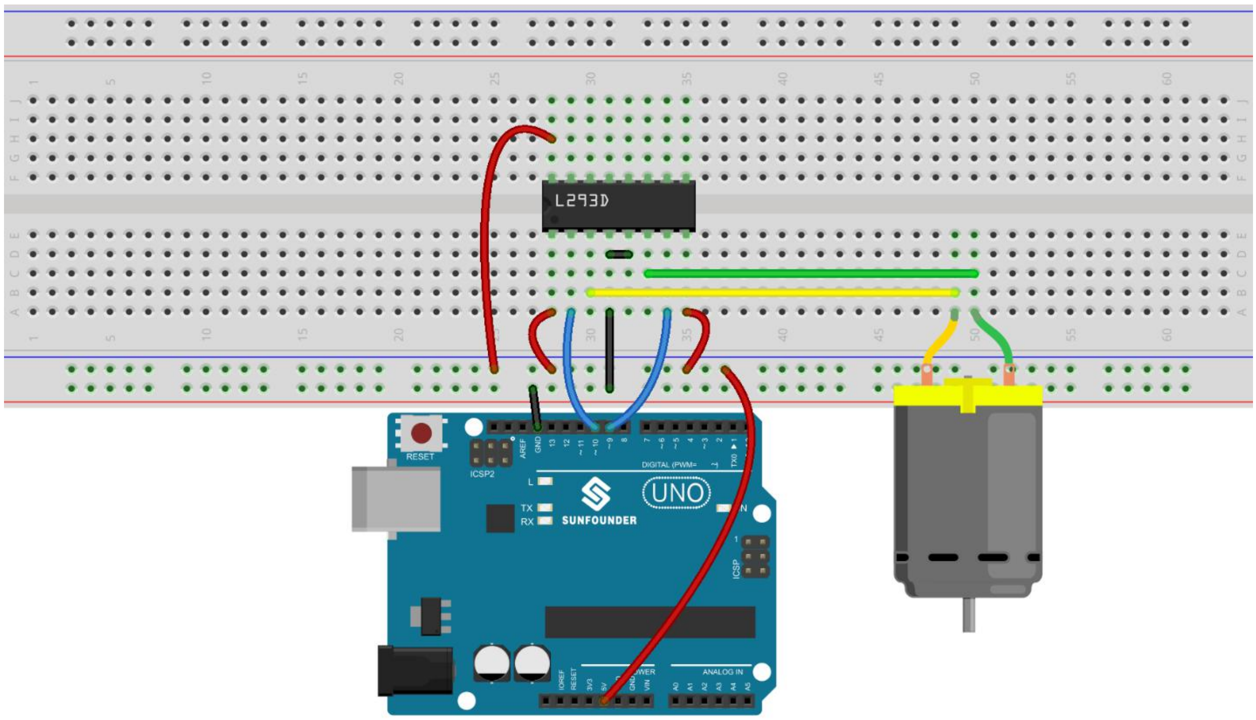
3v Rotating Speed: 13000 RPM

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

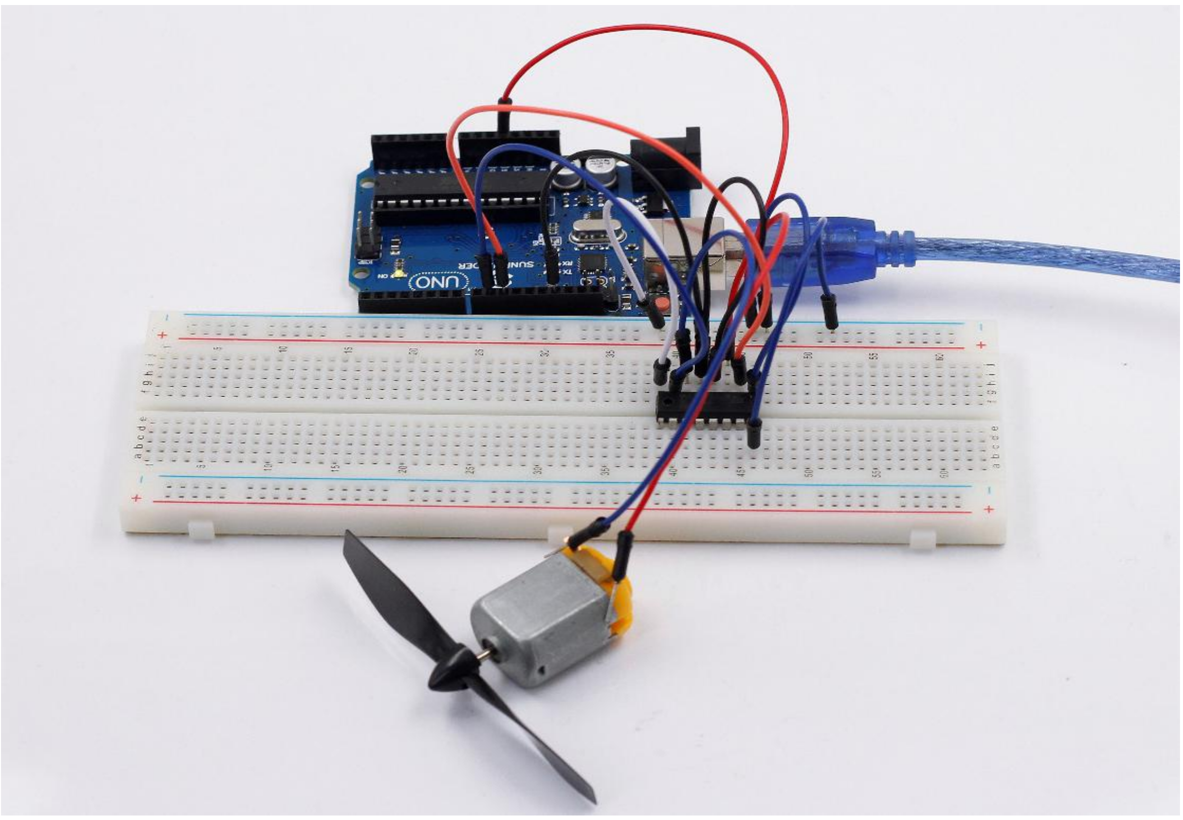


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

The blade of the DC motor will begin rotating left and right, in a speed that varies accordingly.



Lesson 8 LCD1602

Introduction

In this experiment, the SunFounder Uno board is used to directly drive LCD1602 to display characters.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * LCD1602
- 1 * Potentiometer (50kΩ)
- 1 * USB cable
- Jumper wires

Principle

Generally, LCD1602 has parallel ports, that is, it would control several pins at the same time. LCD1602 can be categorized into eight-port and four-port connections. If the eight-port connection is used, then all the digital ports of the SunFounder Uno board are almost completely occupied. If you want to connect more sensors, there will be no ports available. Therefore, the four-port connection is used here for better application.

Pins of LCD1602 and Their Functions

VSS: connected to ground

VDD: connected to a +5V power supply

VO: to adjust the contrast

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 to select between reading and writing mode

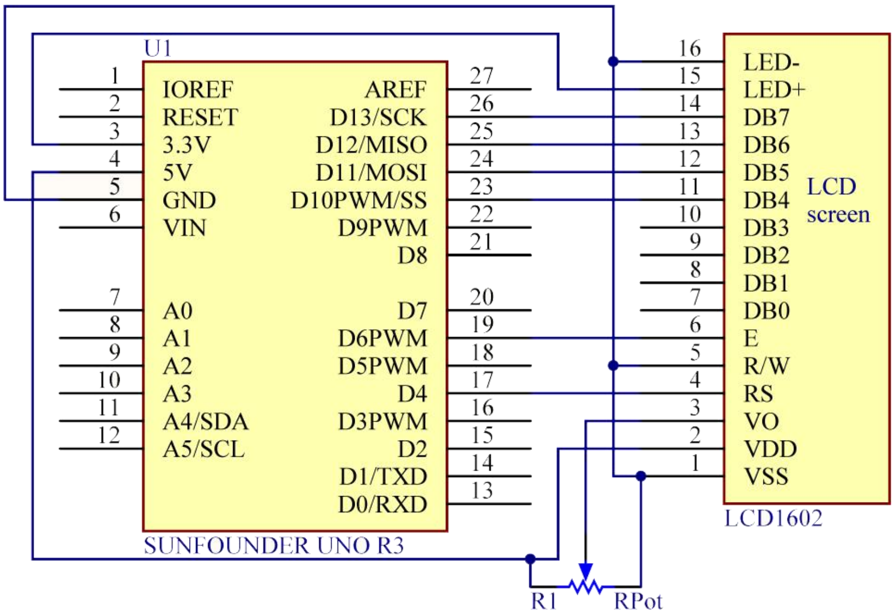
E: An enabling pin that, when low-level energy is supplied, causes the LDC module to execute relevant instructions.

D0-D7: to read and write data

A and K: Pins that control the LCD backlight

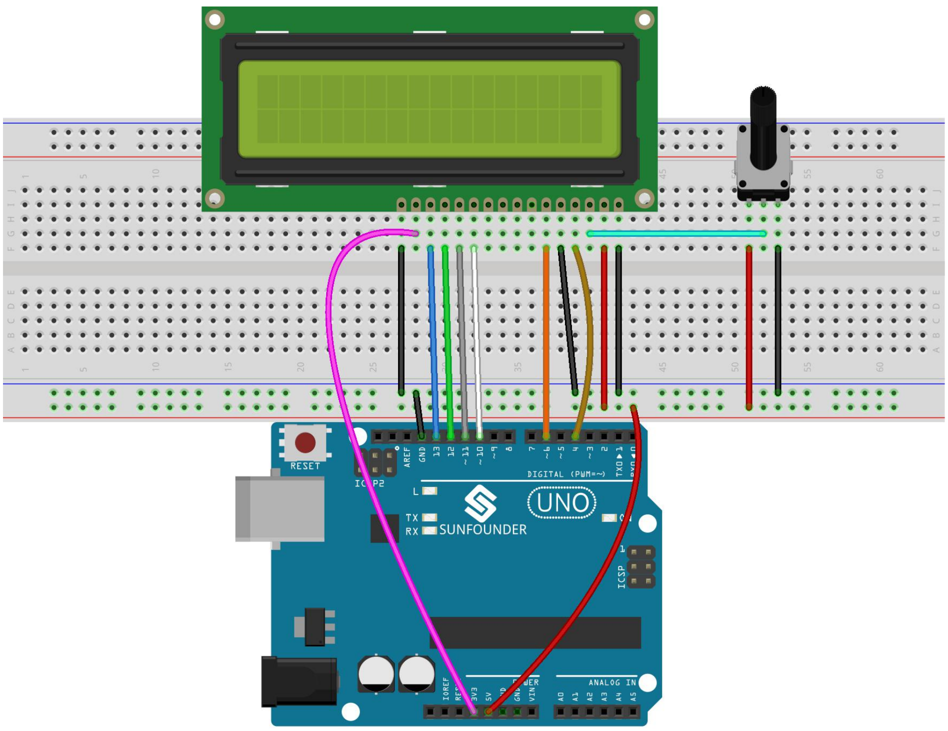
In this experiment, a 50KΩ potentiometer is used to adjust the contrast of LCD1602 to display characters or figures as you want. For programming, it is optimized by calling function libraries.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit (please be sure the pins are connected correctly. Otherwise, characters will not be displayed properly):

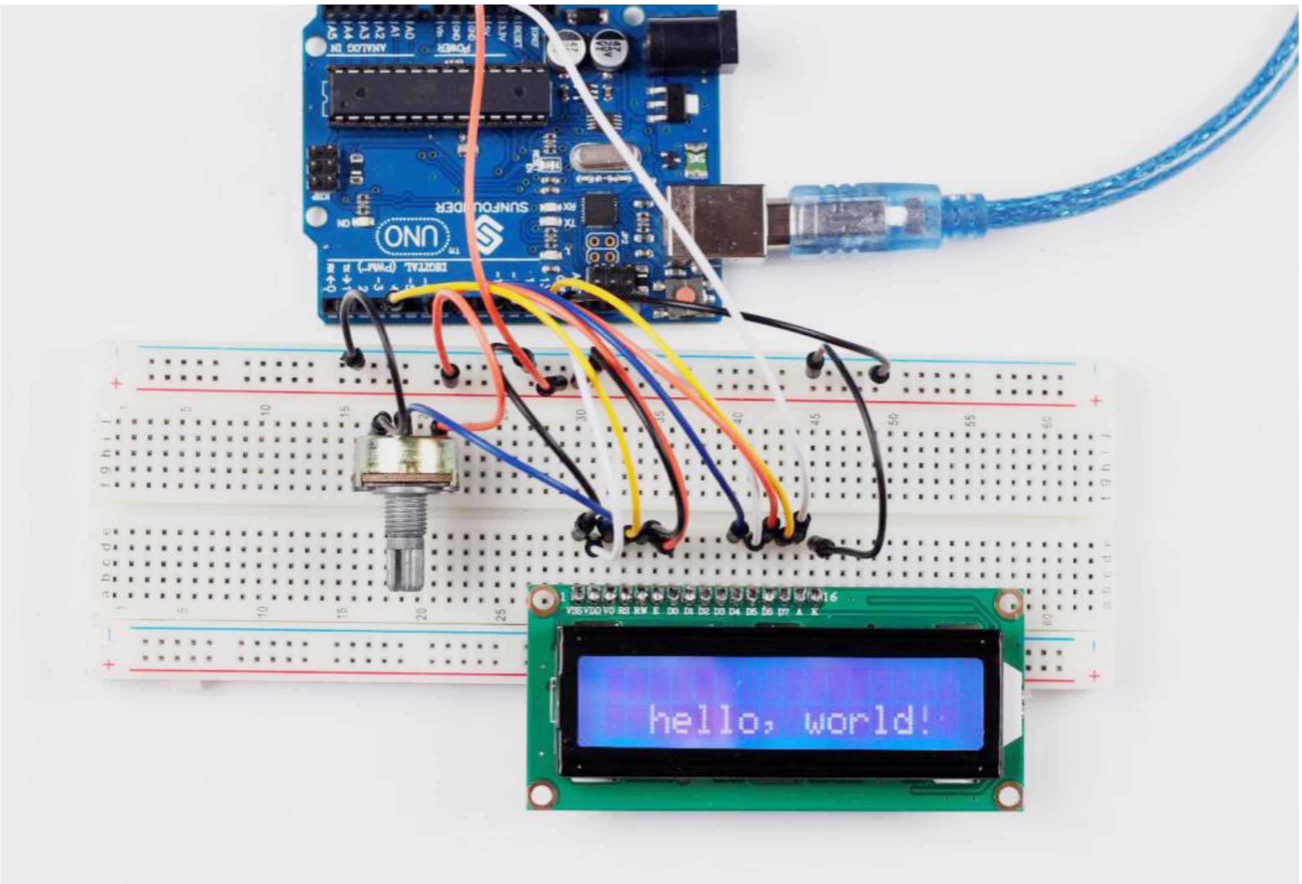


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

You should now see the characters "**SunFounder**" and "**hello, world**" rolling on the LCD. You may need to adjust the potentiometer on the LCD1602 until it can display clearly.



Lesson 9 Serial Monitor

Introduction

In this experiment, you will learn how to turn on or off LEDs through a computer and the Serial Monitor. Serial Monitor is used for communication between the Uno board and a computer or other devices. It is a built-in software in the Arduino environment and you can click the button on the upper right corner to open it. You can send and receive data via the serial port on the Uno board and control the board by input from the keyboard.

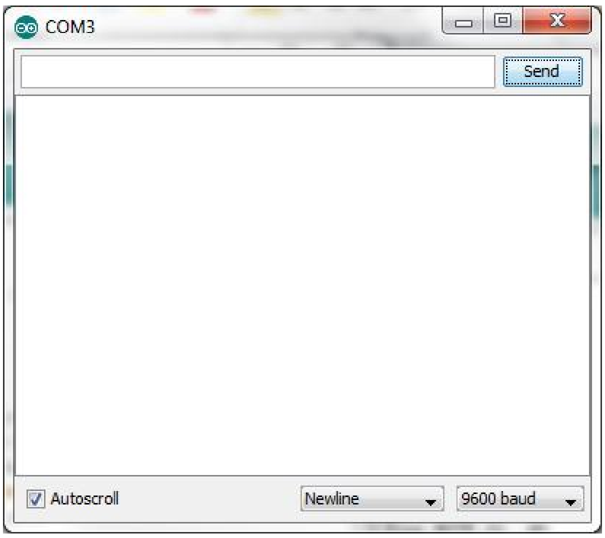
In this experiment, since we use colored LEDs as loads, you can enter a color among red, green, and blue on Serial Monitor in the IDE. The corresponding LED on the SunFounder Uno board will then light up.

Components

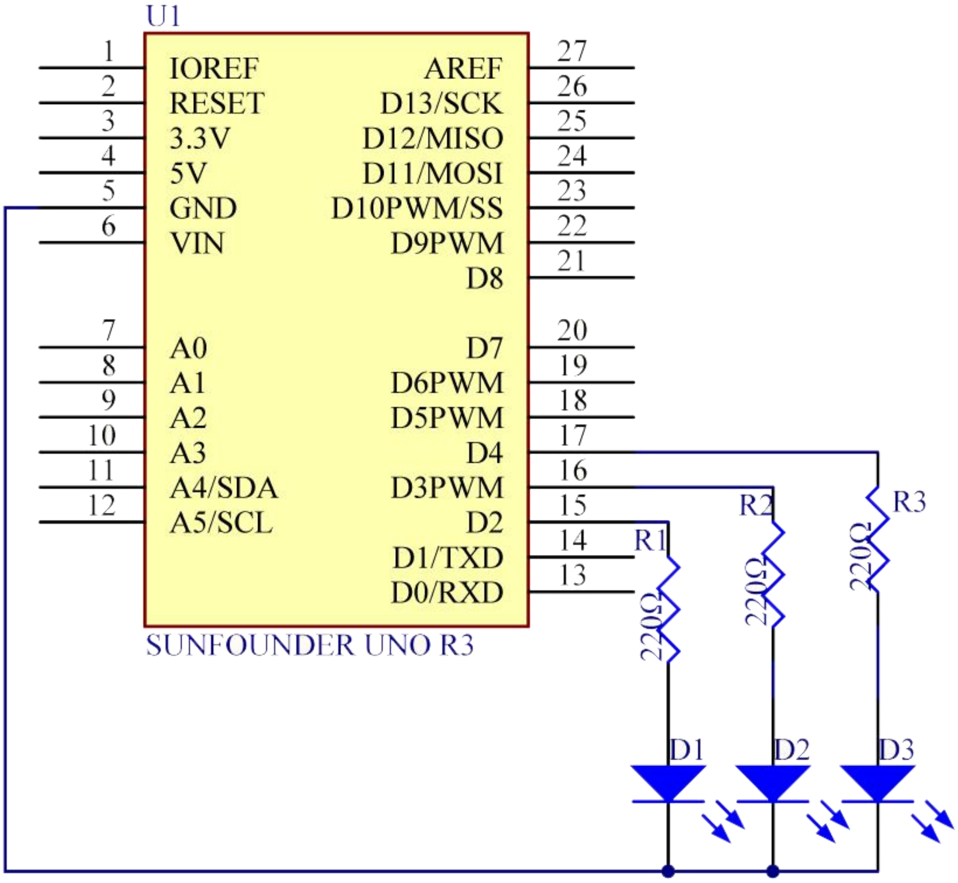
- 1 * SunFounder Uno board
- 1 * Breadboard
- 3 * LED
- 3 * Resistor (220Ω)
- 1 * USB cable
- Jumper wires

Principle

Here, the Serial Monitor serves as a transfer station for communication between your computer and the SunFounder Uno board. First, the computer transfers data to the Serial Monitor, and then the data is read by the Uno board. Finally, the Uno will perform related operations.

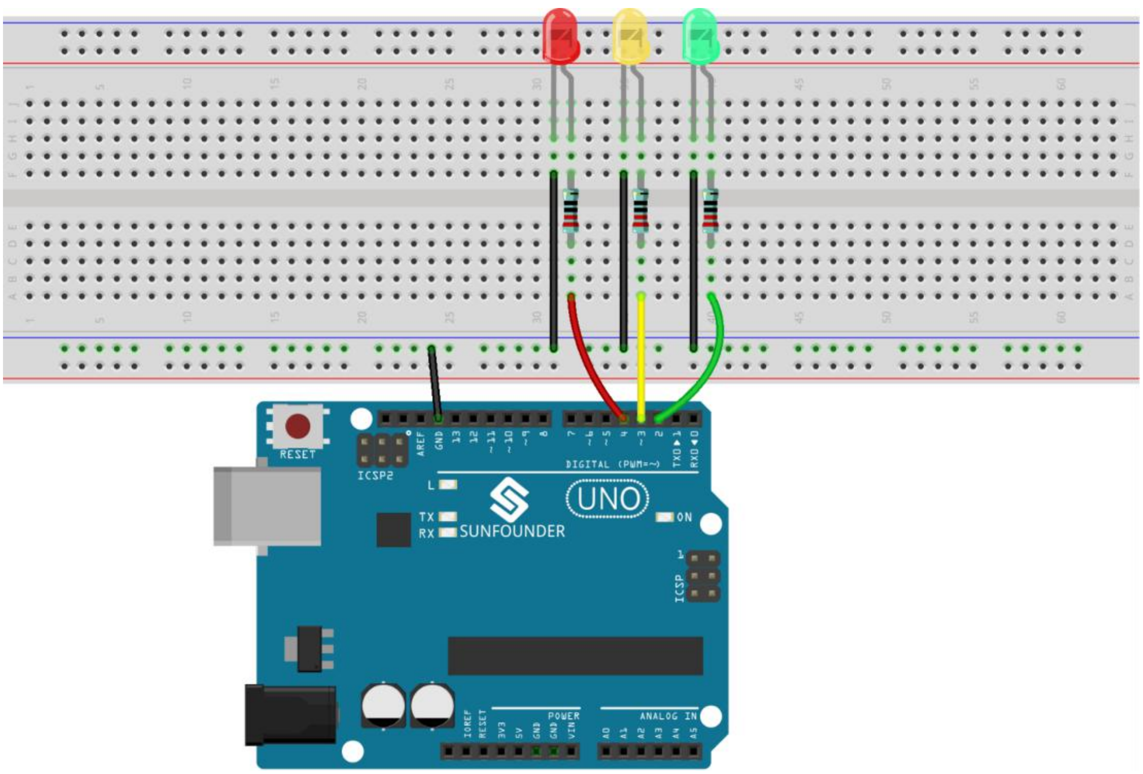


The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

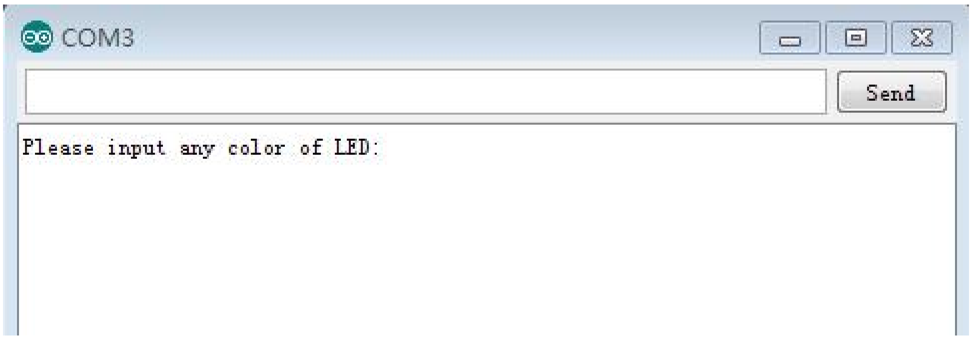


Step 2: Open the code file.

Step 3: Select correct Board and Port.

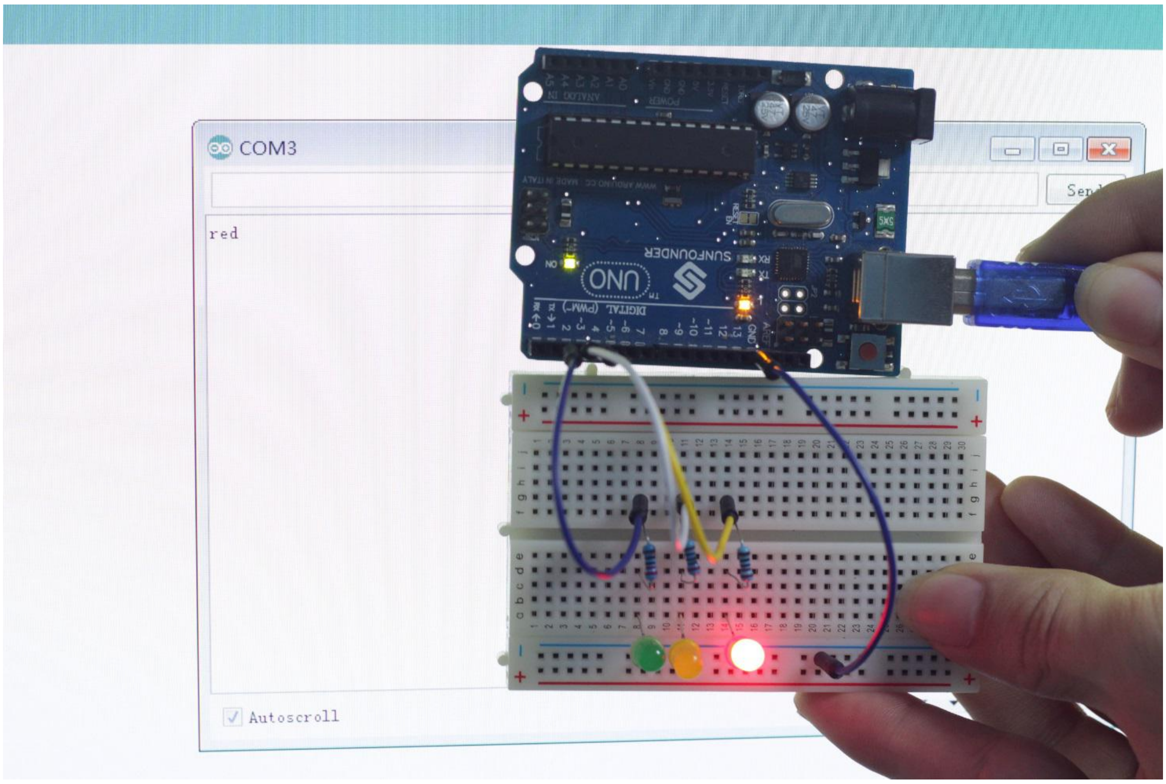
Step 4: Upload the sketch to the SunFounder Uno board.

Now, click the Serial Monitor button at the upper right corner in the IDE. Then the Serial Monitor window will pop up as shown below.



With this window, you can not only send data from your computer to the SunFounder Uno board, but also receive data from the board and display it on the screen. When you open the window, it will display **"Please input any color of LED:**". You can input a color here. If you enter red, green, or blue, click **Send**, then the corresponding LED on the breadboard will light up. However, if you enter any other colors, no LEDs will brighten.

For example, if you input red, you will see the red LED light up.



Lesson 10 7-Segment Display

Introduction

A 7-segment display is a device that can display numerals and letters. It's made up of seven LEDs connected in parallel. Different letters/numbers can be shown by connecting pins on the display to the power source and enabling the related pins, thus turning on the corresponding LED segments. In this lesson we will learn how to use it to display specific characters.

Components

- 1 * SunFounder Uno board
- 1 * 7-segment display (Common Cathode)
- 8 * Resistor (220Ω)
- 1 * USB cable
- 1 * Breadboard
- Jumper wires

Principle

7-Segment Display

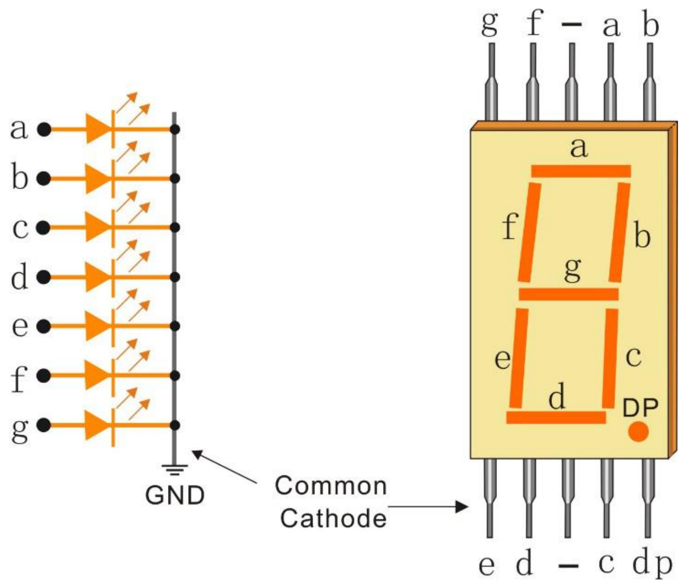
A 7-segment display is an 8-shaped component which packages 7 LEDs. Each LED is called a segment – when energized, one segment forms part of a numeral (both decimal and hexadecimal) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each of the LEDs in the display is given a positional segment with one of its connection pins led out from the rectangular plastic package. These LED pins are labeled from "a" through to "g" representing each individual LED. The other LED pins are connected together forming a common pin. So by forward biasing the appropriate pins of the LED segments in a particular order, some segments will brighten and others stay dim, thus showing the corresponding character on the display.

The common pin of the display generally tells its type. There are two types of pin connection: a pin of connected cathodes and one of connected anodes, indicating Common Cathode (CC) and Common Anode (CA). As the name suggests, a CC display has all the cathodes of the 7 LEDs connected when a CA display has all the anodes of the 7 segments connected.

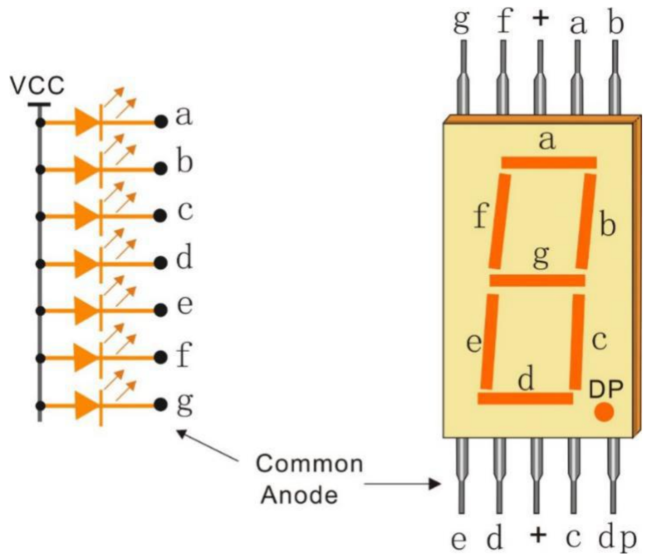
Common Cathode 7-Segment Display

In a common cathode display, the cathodes of all the LED segments are connected to the logic "0" or ground. Then an individual segment (a-g) is energized by a "HIGH", or logic "1" signal via a current limiting resistor to forward bias the anode of the segment.

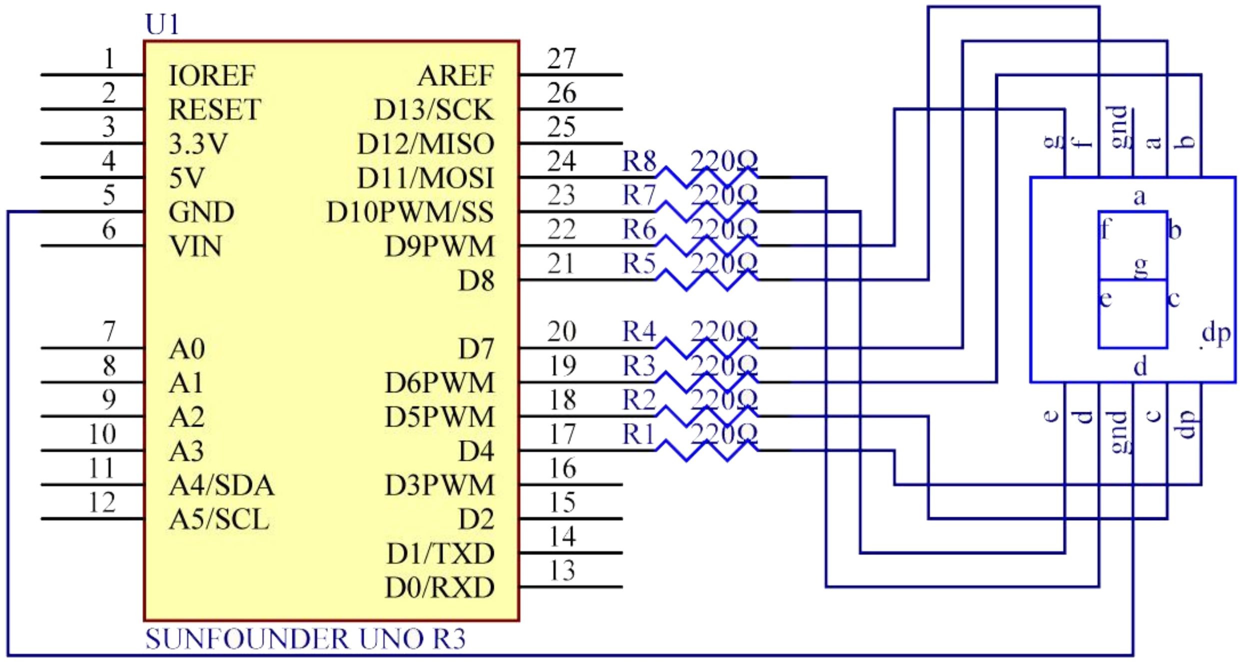


Common Anode 7-Segment Display

In a common anode display, the anodes of all the LED segments are connected to the logic "1". Then an individual segment (a-g) is energized by a ground, logic "0" or "LOW" signal via a current limiting resistor to the cathode of the segment.



The Schematic Diagram



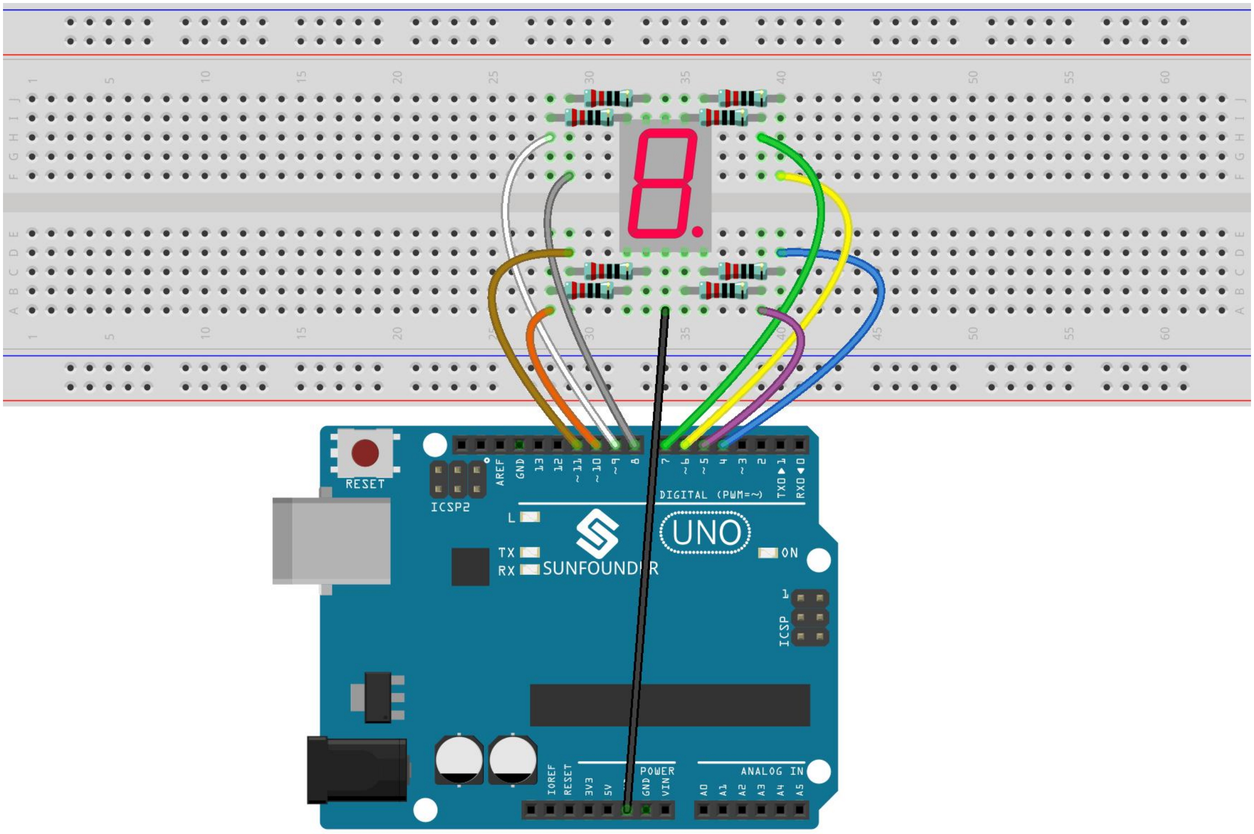
Experimental Procedures

Step 1: Build the circuit (here a common cathode 7-segment display is used).

The wiring between the 7-segment display and the SunFounder Uno board is as shown below:

7-Segment Display	SunFounder Uno Board
a	7
b	6
c	5
d	11

e	10
f	8
g	9
dp	4
-	GND

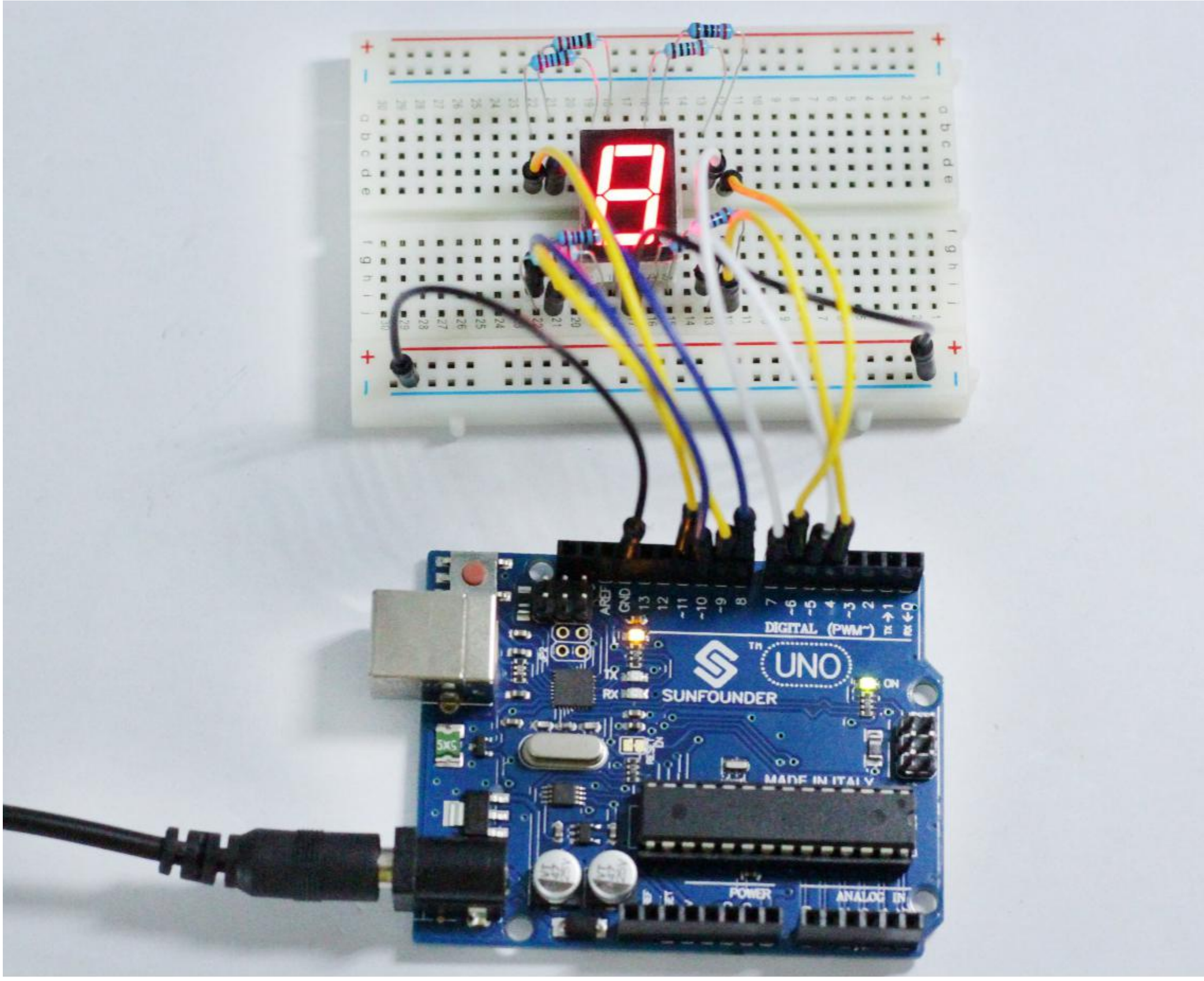


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

You should now see the 7-segment display from 0 to 9 and then A to F, back and forth.


SUNFOUNDER
UNO
DIGITAL (PWM~)
MADE IN ITALY
POWER
ANALOG IN

Lesson 11 74HC595

Introduction

Generally, there are two ways to drive a single 7-segment display. One way is to connect its 8 pins directly to eight ports on the SunFounder Uno board, which we have done previously. Or you can connect the 74HC595 to three ports of the Uno board and then the 7-segment display to the 74HC595. In this experiment, we will use the latter. By this way, we can save five ports – considering the Uno board's limited ports, this is very important. Now let's get started!

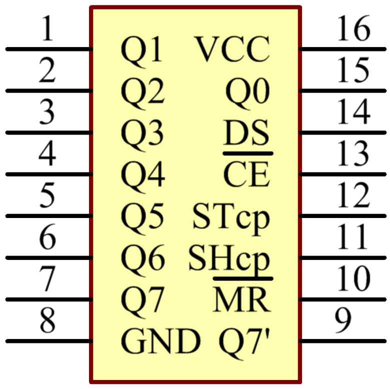
Components

- 8 * Resistor (220Ω)
- 1 * 74HC595
- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- 1 * 7-segment display
- Jumper wires

Principle

74HC595

The 74HC595 consists of an 8–bit shift register and a storage register with three–state parallel outputs. It converts serial input into parallel output so that you can save IO ports of an MCU. The 74HC595 is widely used to indicate multipath LEDs and drive multi-bit segment displays. "Three-state" mentioned above refers to the fact that you can set the output pins as either high, low or high impedance. With data latching, the instant output will not be affected during the shifting; with data output, you can cascade 74HC595s more easily.



Pins of 74HC595 and Their Functions

Q0-Q7: 8-bit parallel data output pins, able to control 8 LEDs or 8 pins of 7-segment display directly.

Q7': Series output pin, connected to DS of another 74HC595 to connect multiple 74HC595s in series

MR: Reset pin, active at low level; here it is directly connected to 5V.

SH: Time sequence input of shift register. On the rising edge, the data in shift register moves successively one bit, i.e. data in Q1 moves to Q2, and so forth. While on the falling edge, the data in shift register remain unchanged.

ST: Time sequence input of storage register. On the rising edge, data in the shift register moves into memory register.

OE: Output enable pin, active at low level, connected to GND.

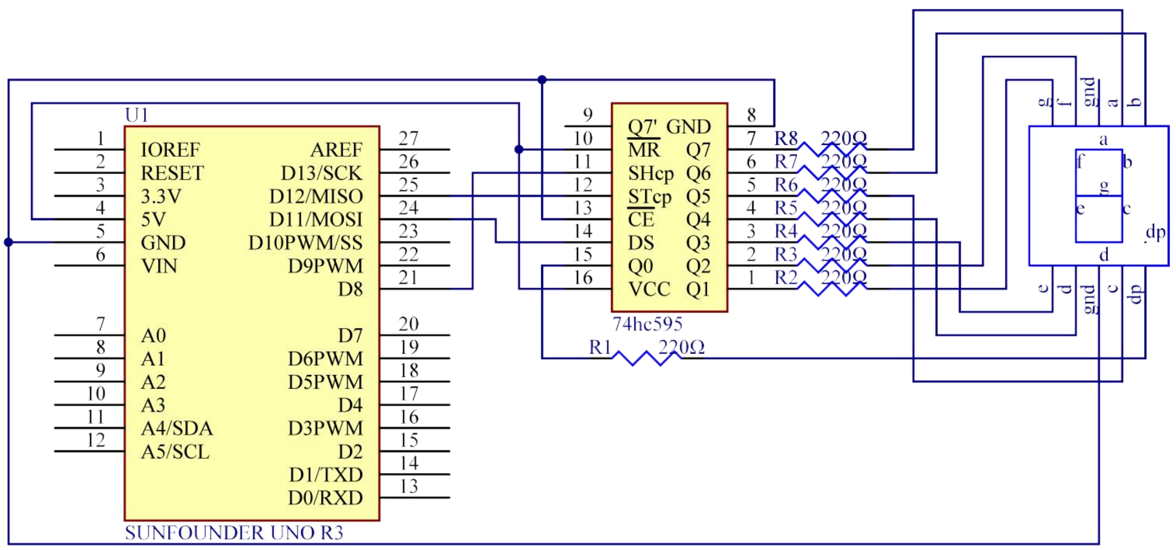
DS: Serial data input pin

VCC: Positive supply voltage

GND: Ground

Here a function *shiftout()* is used, which comes with the Arduino IDE. Simply input a number between 0 and 255 and the storage register can convert it into an 8-bit binary number and output it in parallel. This allows you to easily control the 8 pins of the 7-segment display and create any patterns you want.

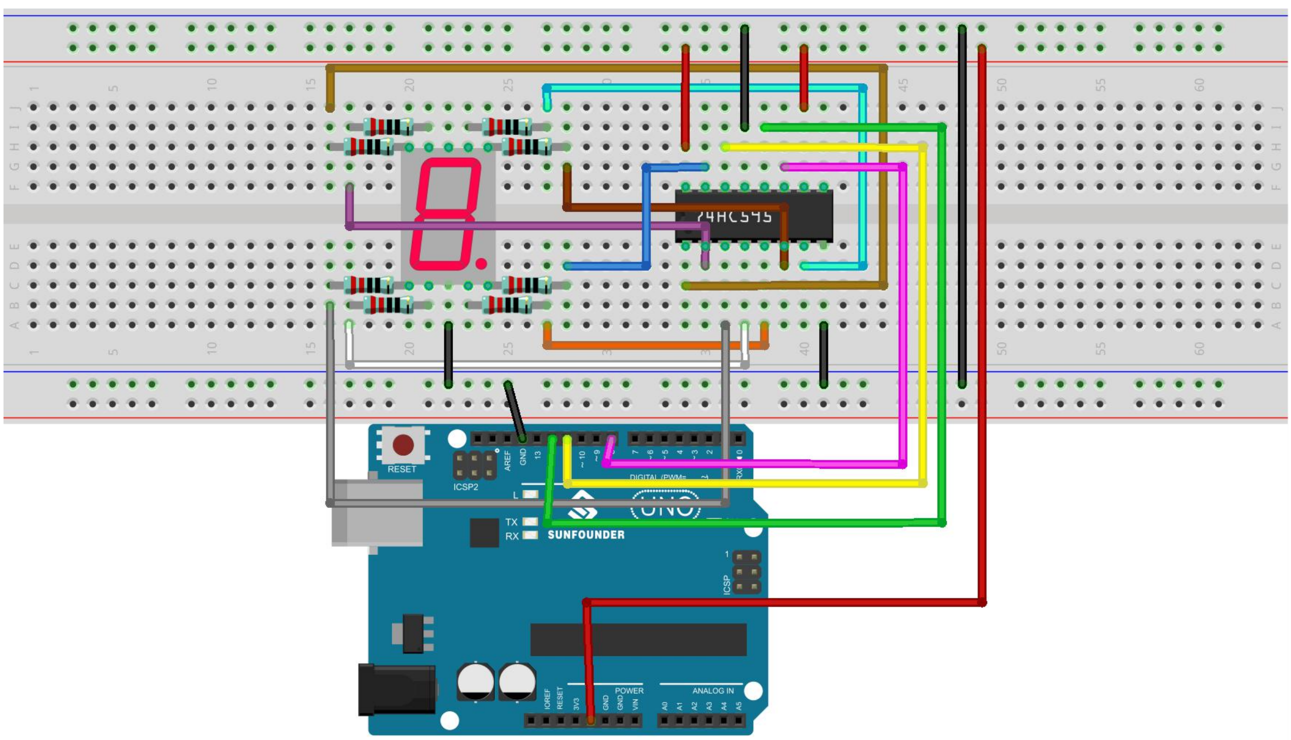
The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

7-Segment Display	74HC595	SunFounder Uno R3
a	Q7	
b	Q6	
c	Q5	
d	Q4	
e	Q3	
f	Q2	
g	Q1	
DP	Q0	
	VCC	5V
	DS	11
	CE	GND
	ST	12
	SH	8
	MR	5V
	Q7'	N/C
	GND	GND
-		GND

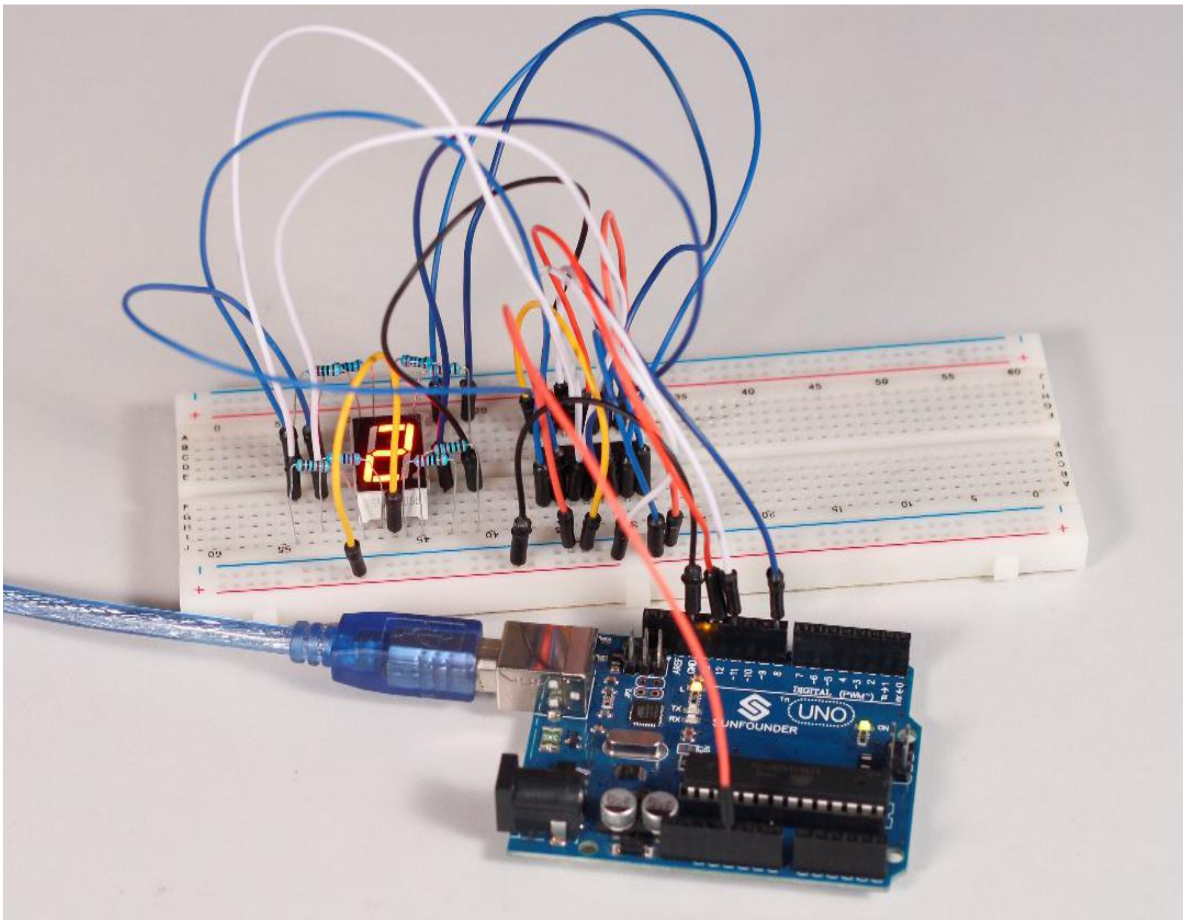


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

You should now see the 7-segment display from 0 to 9 and A to F.



Lesson 12 Dot Matrix LED Display

Introduction

With low-voltage scanning, dot matrix LED displays have advantages such as power saving, long service life, low cost, high brightness, a wide angle of view, long visual range, waterproofness, and so on. They can meet the needs of different applications and thus have a broad development prospect. In this experiment, we will make it display some simple characters like numbers and letters to experience its charm from the beginning.

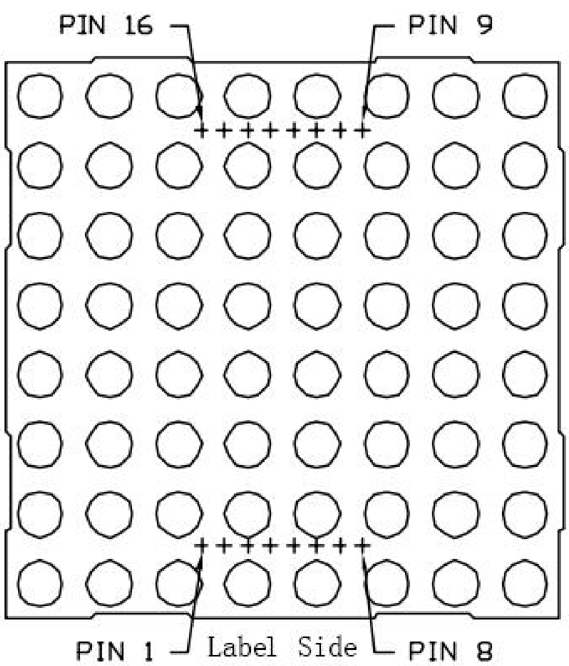
Components

- 1 * SunFounder Uno board
- 1 * Dot-matrix (8*8)
- 8 * Resistor (220Ω)
- 1 * Breadboard
- 2 * 74HC595
- 1 * USB cable
- Jumper wires

Principle

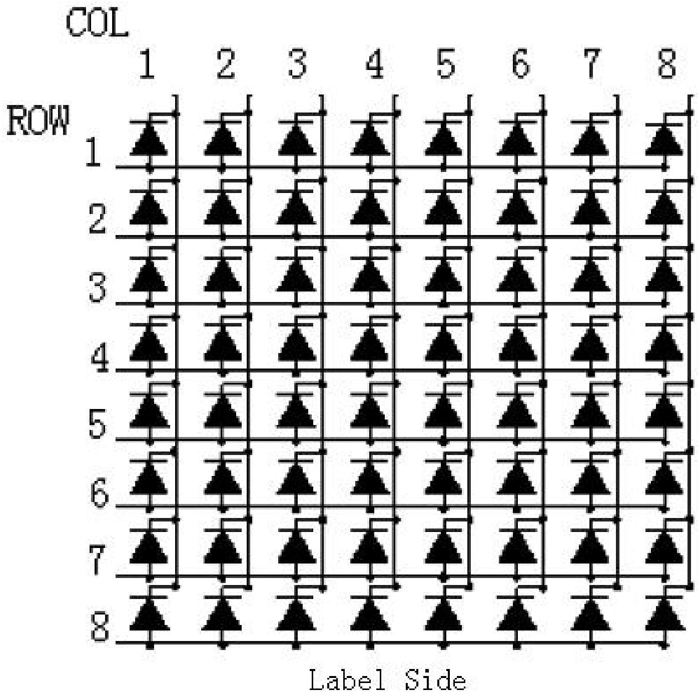
Dot Matrix

The external view:



Pin Definition

Define the row and column numbering at first (only for a dot matrix whose model number ends with BS)



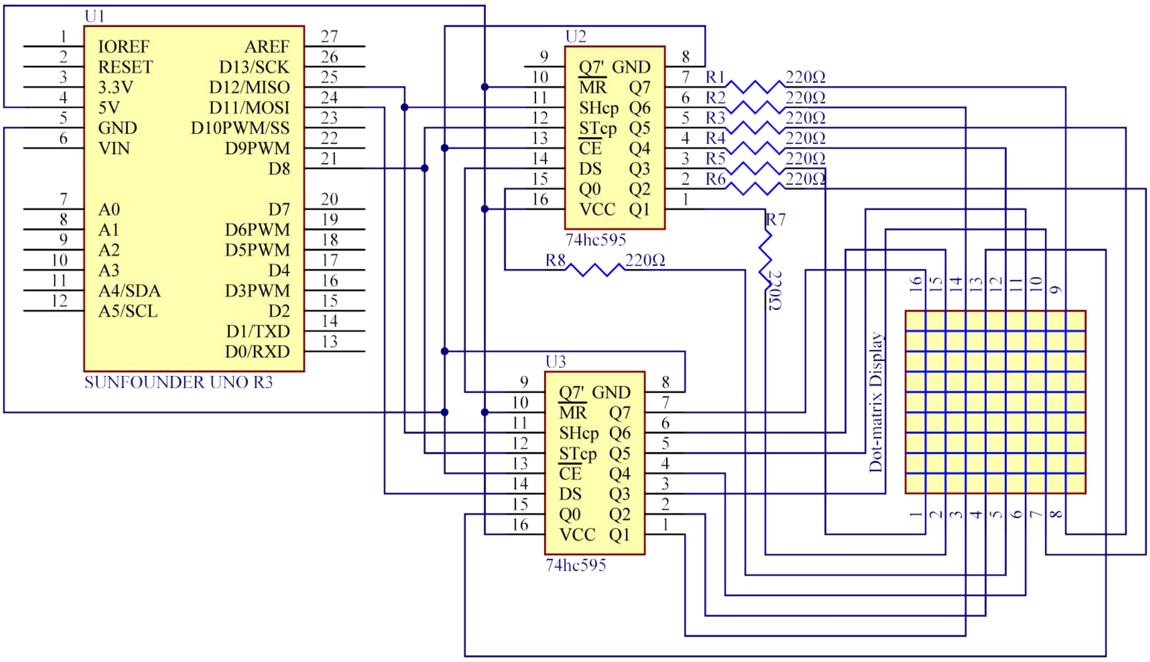
Pin numbering corresponding to the above rows and columns:

COL	1	2	3	4	5	6	7	8
Pin No.	13	3	4	10	6	11	15	16
ROW	1	2	3	4	5	6	7	8
Pin No.	9	14	8	12	1	7	2	5

The 8*8 dot matrix is made up of sixty-four LEDs and each LED is placed at the cross point of a row and a column. When the electrical level of a certain row is High and the electrical level of a certain column is Low, the corresponding LED at their cross point will light up. For example, to turn on the LED at the first dot, you should set ROW 1 to high level and COL 1 to low, so the LED at the first dot brightens; to turn on all the LEDs on the first row, set ROW 1 to high level and COL 1-8 to low, and then all the LEDs on the first row will light up; similarly, set COL 1 to low level and ROW 1-8 to high level, and all the LEDs on the first column will light up.

The principle of 74HC595 has been previously illustrated. One chip is used to control the rows of the dot matrix while the other is to control the columns.

The Schematic Diagram



Experimental Procedures

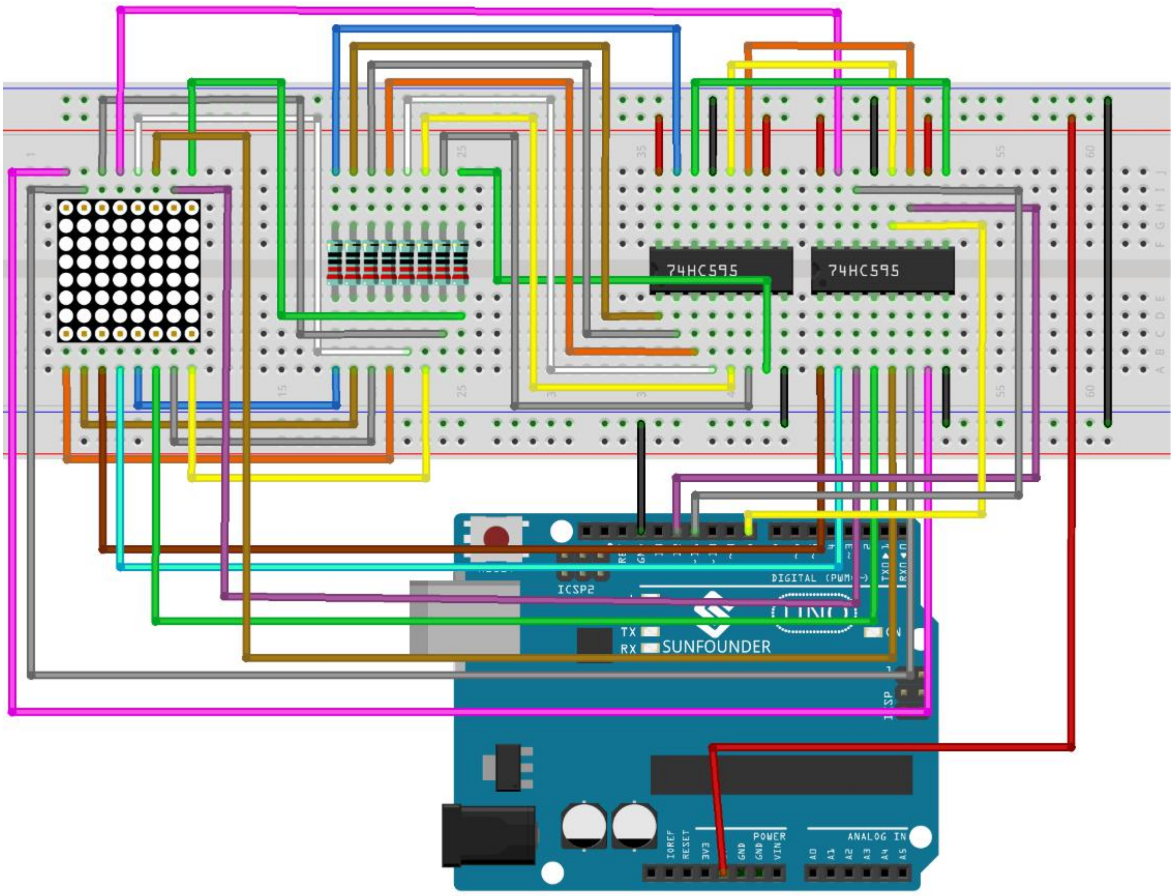
Step 1: Build the circuit.

Connect 74HC595 (u2), Dot Matrix and SunFounder Uno board:

74HC595 (u2)	Dot Matrix	SunFounder Uno
Q1 (Pin1)	2	
Q2 (Pin2)	7	
Q3 (Pin3)	1	
Q4 (Pin4)	12	
Q5 (Pin5)	8	
Q6 (Pin6)	14	
Q7 (Pin7)	9	
GND (Pin8)		GND
Q7' (Pin9)	N/O	N/O
MR (Pin10)		5V
SH_cp (Pin11)		12
ST_cp (Pin12)		8
CE (Pin13)		GND
DS (Pin14) to Q7' of 74HC595 (u3)	N/O	N/O
Q0 (Pin15)	5	
VCC (Pin16)		5V

Connect the other 74HC595 (u3), Dot Matrix and SunFounder Uno board

74HC595 (u3)	Dot-Matrix	SunFounder Uno
Q1 (Pin1)	3	
Q2 (Pin2)	4	
Q3 (Pin3)	10	
Q4 (Pin4)	6	
Q5 (Pin5)	11	
Q6 (Pin6)	15	
Q7 (Pin7)	16	
GND (Pin8)		GND
Q7′ (Pin9) to DS of 74HC595 (u2)	N/O	N/O
MR (Pin10)		5V
SH_cp (Pin11)		12
ST_cp (Pin12)		8
CE (Pin13)		GND
DS (Pin14)		11
Q0 (Pin15)	13	
VCC (Pin16)		5V

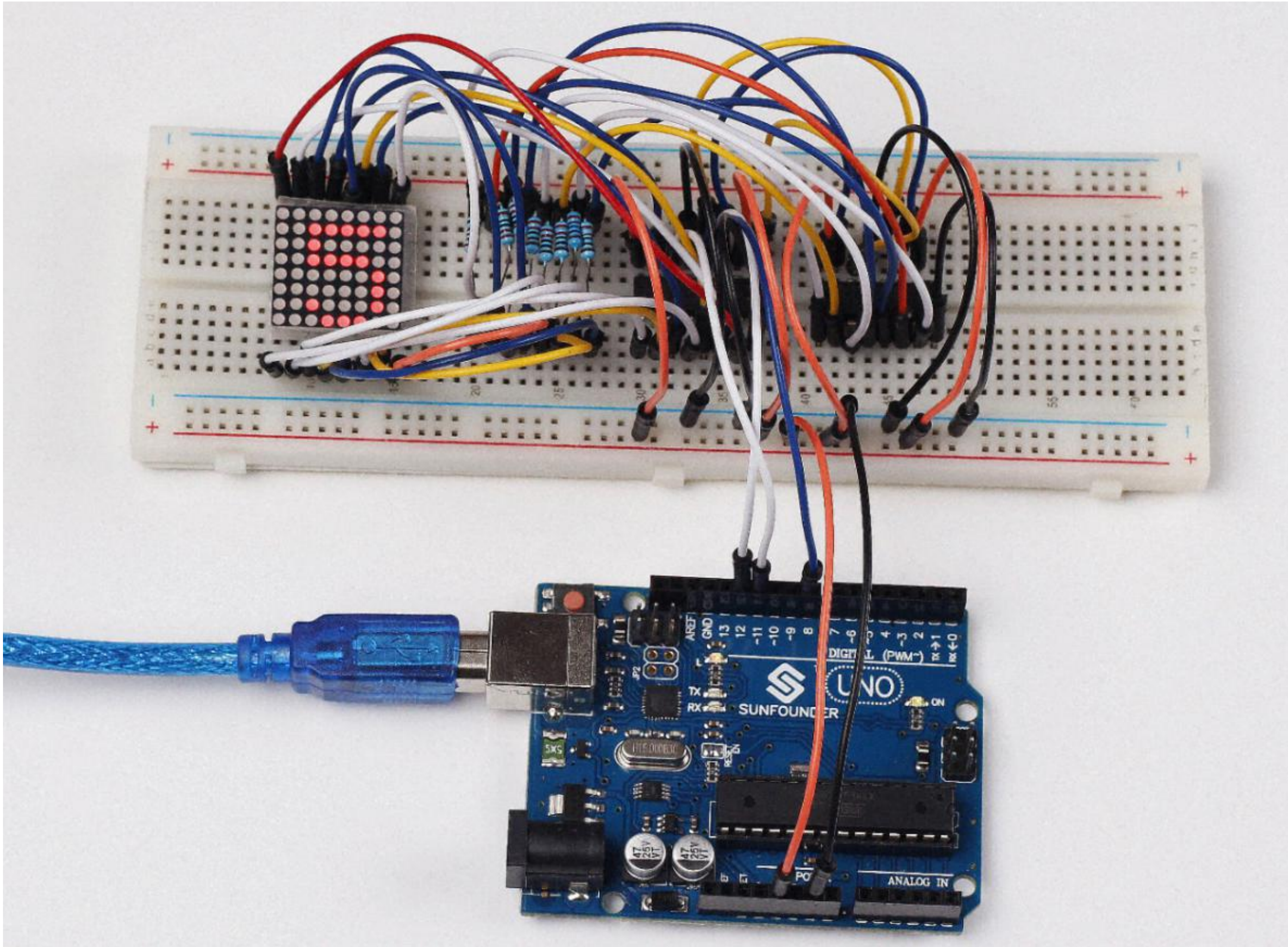


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Here you should see the dot-matrix display 0 to 9 and then A to F circularly.



Lesson 13 NE555 Timer

Introduction

The NE555 Timer, a mixed circuit composed of analog and digital circuits, integrates analog and logical functions into an independent IC, thus tremendously expanding the applications of analog integrated circuits. It is widely used in various timers, pulse generators, and oscillators. In this experiment, the SunFounder Uno board is used to test the frequencies of square waves generated by the 555 oscillating circuit and show them on Serial Monitor.

Components

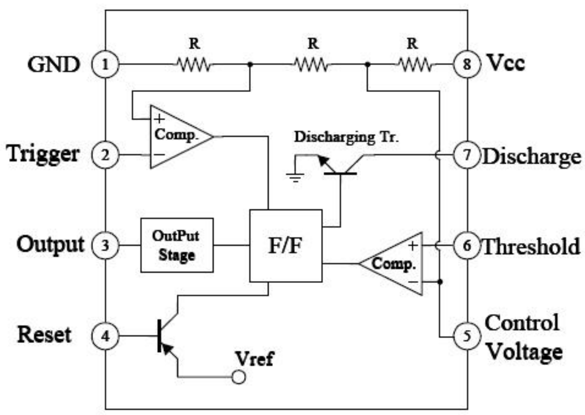
- 1 * SunFounder Uno board
- 1 * USB cable
- 1 * Breadboard
- 1 * NE555
- 2 * 104 ceramic capacitor
- 1 * Potentiometer (50KΩ)
- 1 * Resistor (10KΩ)
- Jumper wires

Principle

555 IC

The 555 IC was originally used as a timer, hence the name 555 time base circuit. It is now widely used in various electronic products because of its reliability, convenience, and low price. The 555 is a complex hybrid circuit with dozens of components such as a divider, comparator, basic R-S trigger, discharge tube, and buffer.

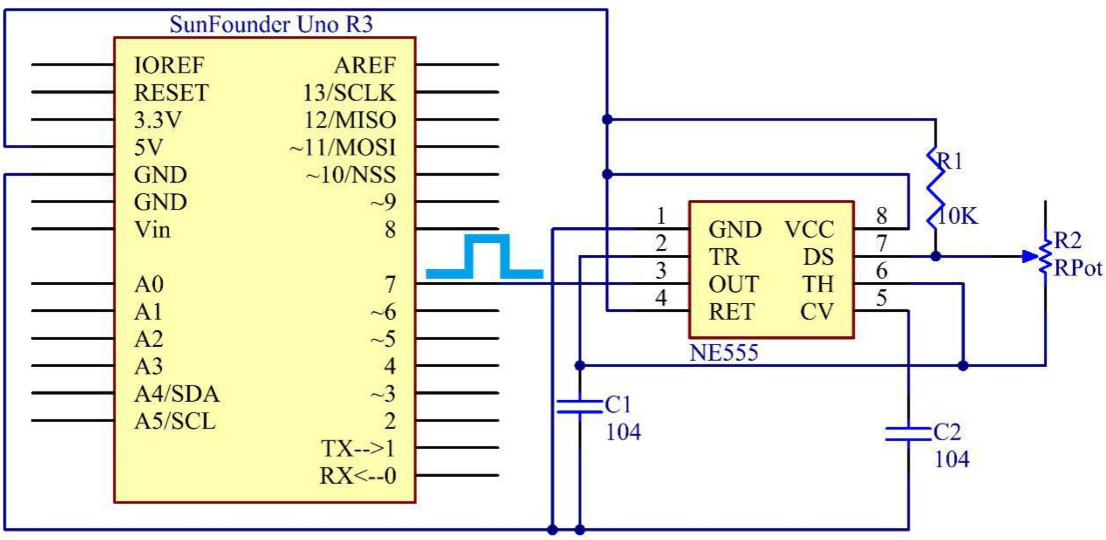
Pins and Their Functions



As shown in the picture, the pins are set dual in-line with the 8-pin package.

- Pin 1 (**GND**): the ground
- Pin 2 (**TRIGGER**): when the voltage at the pin reduces to 1/3 of the VCC (or the threshold defined by the control board), the output terminal sends out a High level

  Pin 3 (**OUTPUT**): outputs High or Low, two states 0 and 1 decided by the input electrical level; maximum output current approx. 200mA at High
- Pin 4 (**RESET**): when a Low level is received at the pin, the timer will be reset and the output will return to Low level; usually connected to positive pole or neglected
- Pin 5 (**CONTROL VOLTAGE**): to control the threshold voltage of the chip (if it skips connection, by default, the threshold voltage is 1/3 VCC and 2/3 VCC)
- Pin 6 (**THRESHOLD)**: when the voltage at the pin increases to 2/3 VCC (or the threshold defined by the control board), the output terminal sends out a High level
- Pin 7 (**DISCHARGE)**: output synchronized with Pin 3, with the same logical level; but this pin does not output current, so pin 3 is the real High (or Low) when pin 7 is the virtual High (or Low); connected to the open collector (OC) inside to discharge the capacitor
- Pin 8 (**VCC**): positive terminal for the NE555 timer IC, ranging +4.5V to +16V
- The NE555 timer works under the monostable, astable and bistable modes. In this experiment, apply it under the astable mode, which means it works as an oscillator, as shown below:



Connect a resistor R1 between the VCC and the discharging pin DS, another resistor between pin DS and the trigger pin TR which is connected to the threshold pin TH and then to the capacitor C1. Connect the RET (pin 4) to GND, CV (pin 5) to another capacitor C2 and then to the ground.

Working Process

The oscillator starts to shake once the circuit is power on. Upon the energizing, since the voltage at C1 cannot change abruptly, which means pin 2 is Low level initially, set the timer to 1, so pin 3 is High level. The capacitor C1 charges via R1 and R2, in a time span:

$$Tc=0.693(R1+R2)$$

When the voltage at C1 reaches the threshold 2/3Vcc, the timer is reset and pin 3 is Low level. Then C1 discharges via R2 till 2/3Vcc, in a time span:

$$Td=0.693(R2)$$

Then the capacitor is recharged and the output voltage flips again:

$$\text{Duty cycle } D=Tc/(Tc+Td)$$

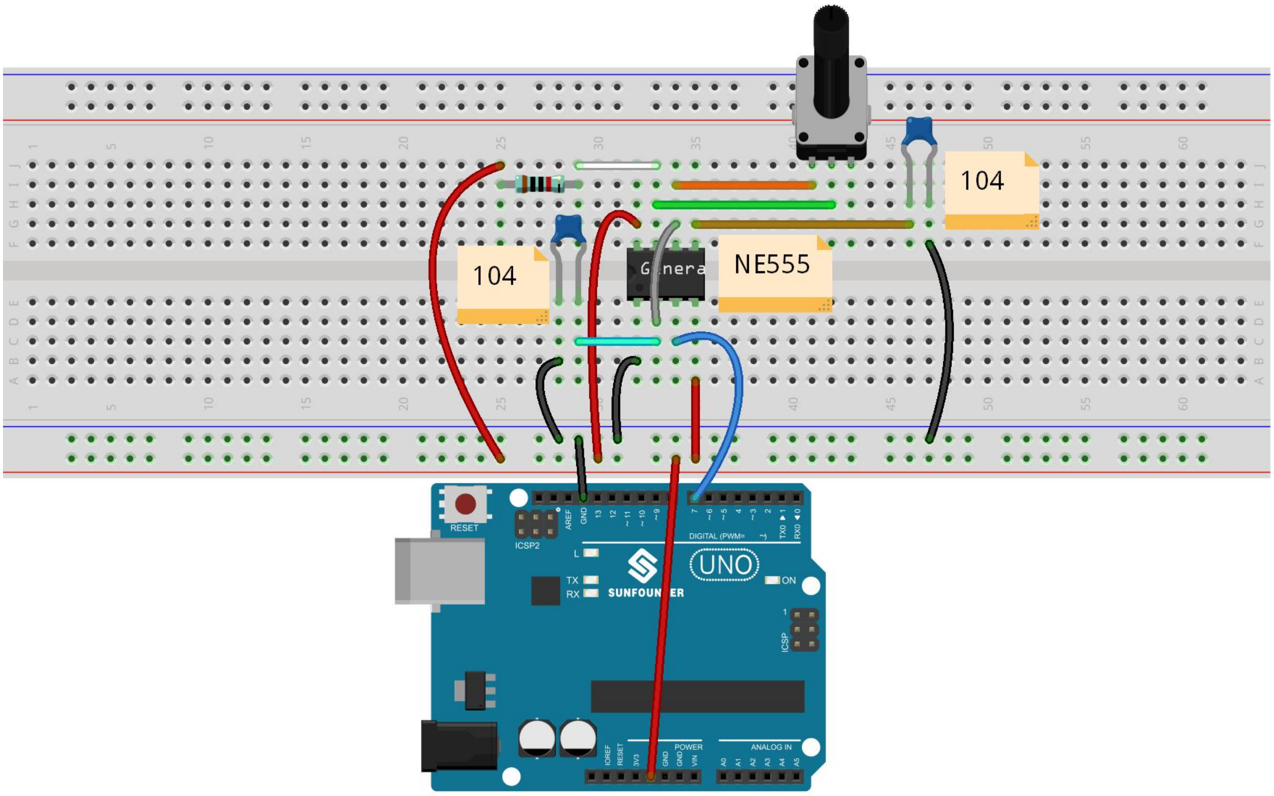
Since a potentiometer is used for resistor, we can output square wave signals with different duty cycles by adjusting its resistance. But R1 is a 10K resistor and R2 is 0k-50k, so the range of the ideal duty cycle is 0.545%-100%. If you want another else, you need to change the resistance of R1 and R2.

$$Dmin=(0.693(10K+0K))/(0.693(10K+0K)+0.693x0k)\ x100\%=100\%$$

$$Dmax=(0.693(10K+50K))/(0.693(10K+50K)+0.693x50k)\ x100\%=54.54\%$$

Experimental Procedures

Step 1: Build the circuit.

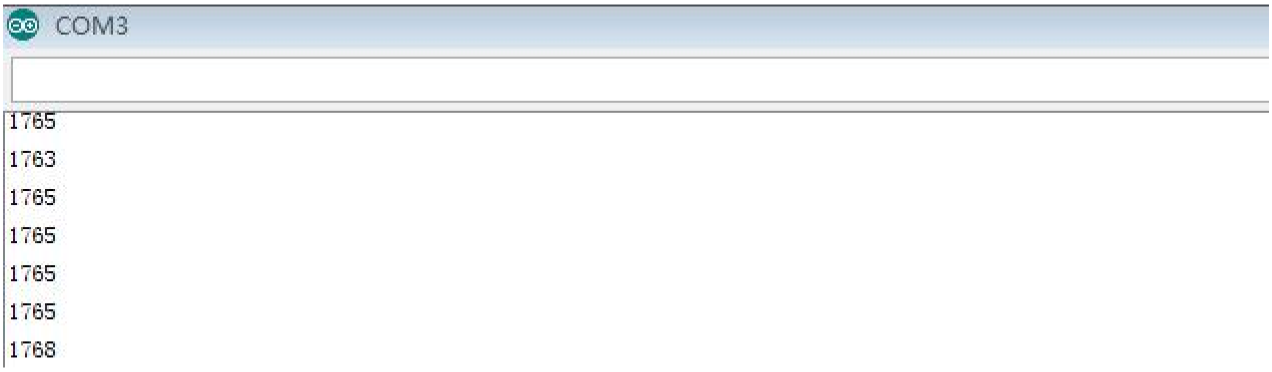


Step 2: Open the code file.

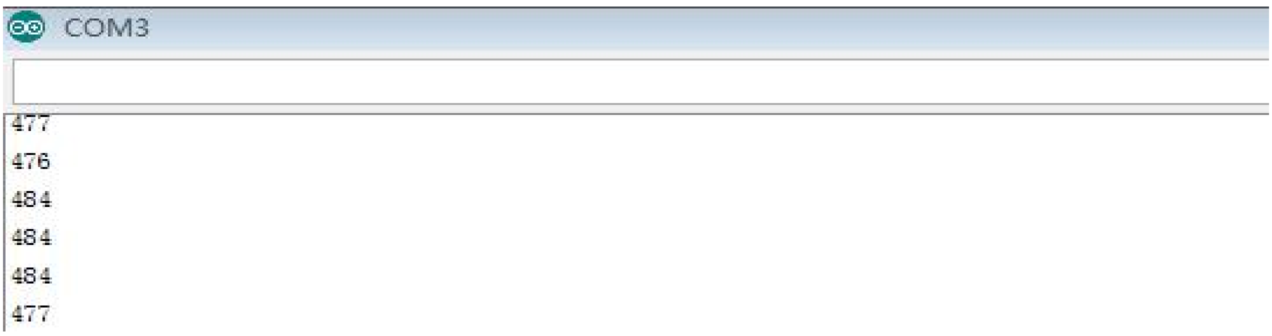
Step 3: Select correct Board and Port.

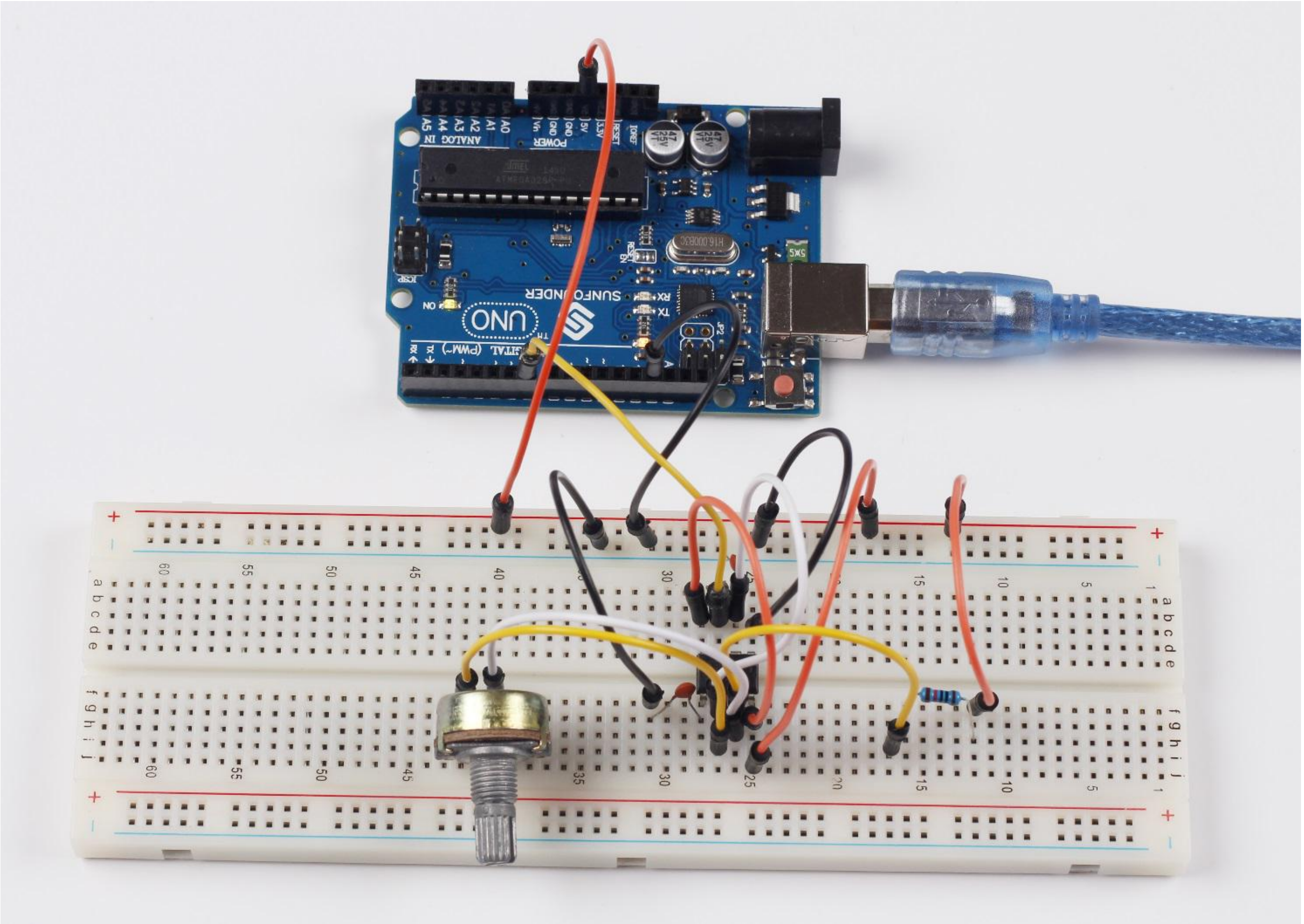
Step 4: Upload the sketch to the SunFounder Uno board.

After uploading, open the Serial Monitor and you will see the following window.



Adjust the potentiometer, and the length of the pulse (in microsecond) displayed will change accordingly.





Lesson 14 Rotary Encoder

Introduction

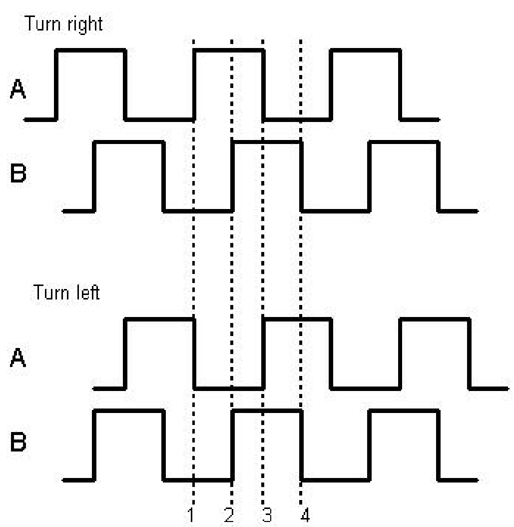
A rotary encoder is a type of electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. In this experiment, we will learn how to apply one.

Components

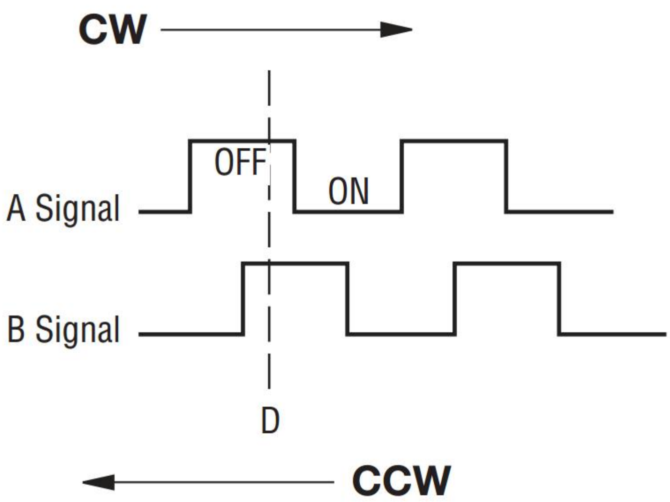
- 1 * SunFounder Uno board
- 1 * USB cable
- 1 * Breadboard
- 1 * Rotary encoder module
- Jumper wires

Principle

There are mainly two types of rotary encoders: absolute and incremental (relative) encoders. The output of absolute encoders indicates the current position of the shaft, making them angle transducers. The output of incremental encoders provides information about the motion of the shaft, which is typically further processed elsewhere into information such as speed, distance, and position.
In this experiment, an incremental encoder is used. It is intended to turn rotational displacement into a series of digital pulse signals which are then used to control the angular displacement. It generates two-phase square waves whose phase difference is 90°. Usually the two-phase square waves are called channel A and channel B as shown below:

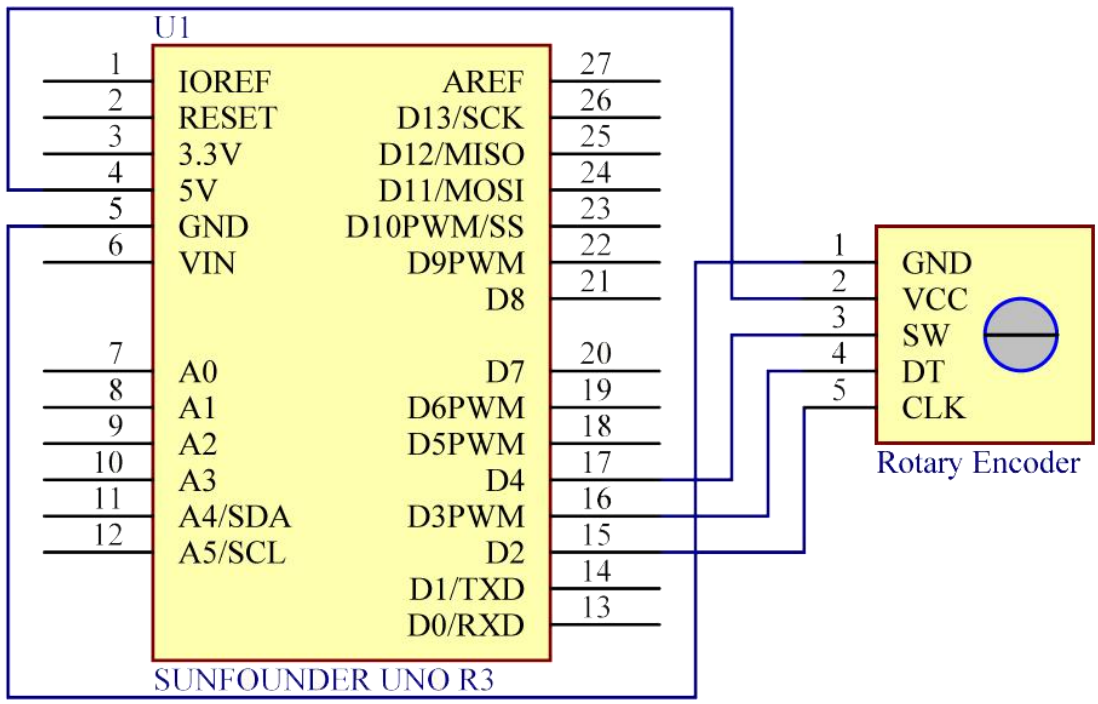


It's difficult to distinguish between turning left and right of a switch during SCM programming. However, when using an oscilloscope to observe the situation of the switch, you will find that a phase difference exists between the signals of the two output pins, shown as follows:



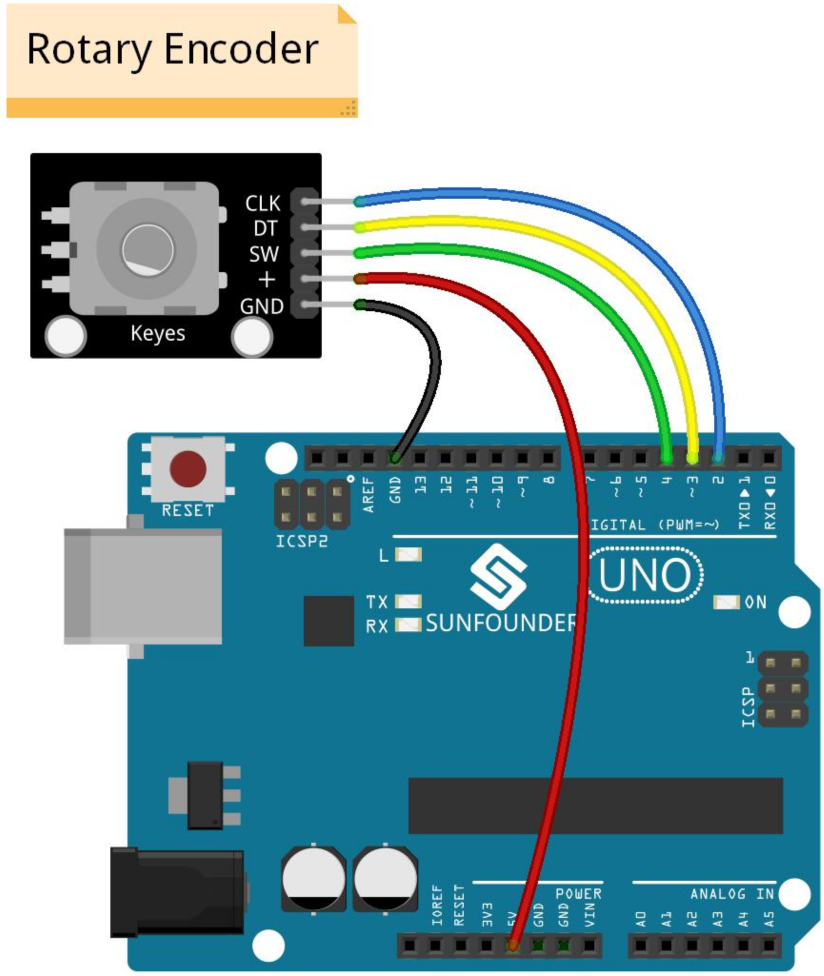
If both channel A and channel B are high, it indicates the switch is spun clockwise (CW); if channel A is high while channel B is low, it implies it is spun counterclockwise (CCW). As a result, if channel A is high, you can tell whether the rotary encoder is turned left or right as long as you know the state of channel B.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.



For the convenience of wiring, a rotary encoder module is used.

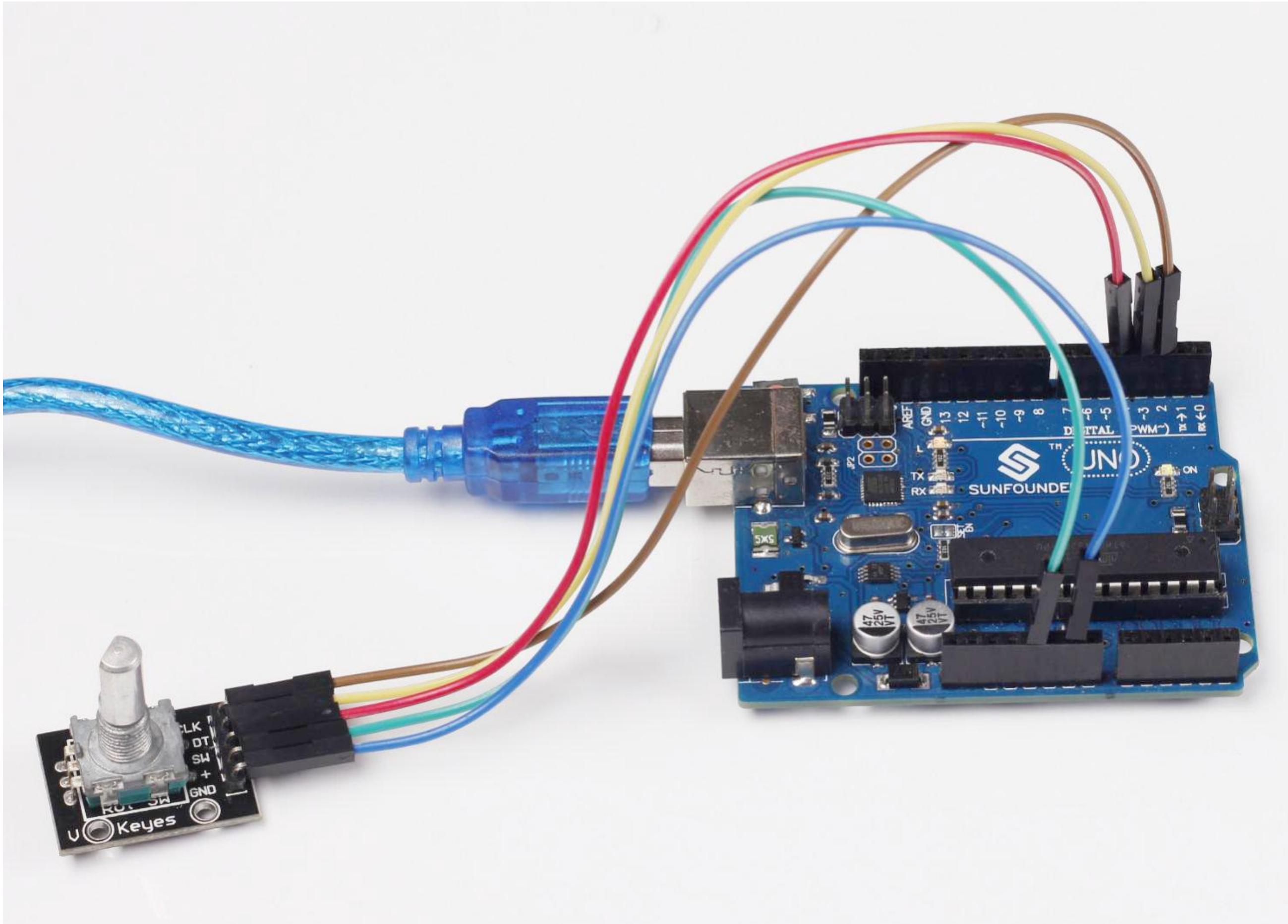
Rotary Encoder Module	SunFounder Uno
+	5V
GND	GND
CLK	Pin 2
DT	Pin 3
SW	Pin 4

Step 2: Open the code file.

Step 3: Select correct Board and Port.

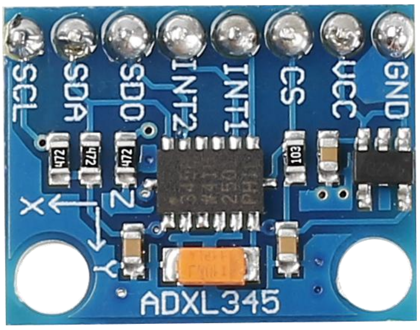
Step 4: Upload the sketch to the SunFounder Uno board.

Open Serial Monitor and you will see the angular displacement of the rotary encoder printed on the window. Spin the shaft of the rotary encoder clockwise, and the angular displacement will increase; spin it counterclockwise, the value decreases. Press it down, and the value will be reset and restore to the initial state.



Lesson 15 ADXL345

Introduction



In this lesson, we will learn how to use the acceleration sensor ADXL345.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * ADXL345 module
- 1 * USB cable
- Jumper wires

Principle

Accelerometer

An accelerometer is used to measure the force generated during the acceleration. The most fundamental is the commonly-known acceleration of gravity which is 1g.

By measuring the acceleration caused by gravity, you can calculate the tilt angle of the device to the level surface. Through analyzing the dynamic acceleration, you can tell the way how the device is moving. For example, self-balancing board or hoverboard applies the acceleration sensor and gyroscope for Kalman filter and posture correction.

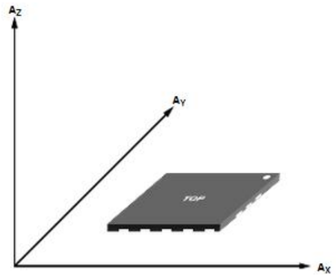
ADXL345

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit two's complement and is accessible through either an SPI (3- or 4-wire) or I2C digital interface. In this experiment, the I2C digital interface is used.

It is well suited to measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables the inclination change measurement by less than 1.0°. And the excellent sensitivity (3.9mg/LSB @2g) provides a high-precision output of up to ±16g.

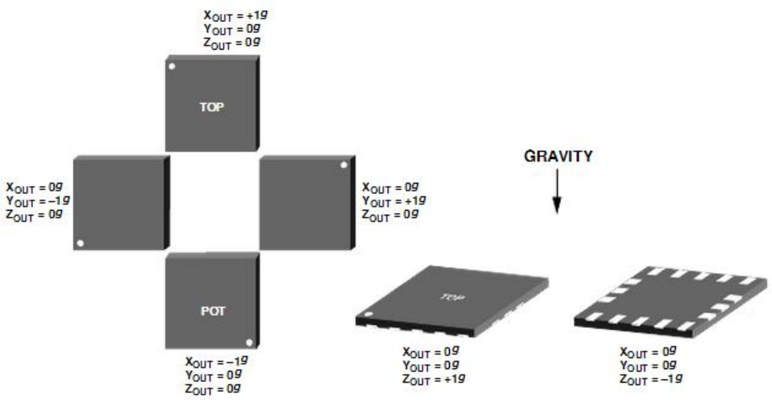
How ADXL345 Works

The ADXL345 detects the acceleration with the sensing component at the front, and then the electric signal sensing component changes it into electric signal, which is analog. Next, the AD adapter integrated on the module will convert the analog signal into digital one.



Detection Axes of ADXL345

The X_OUT, Y_OUT and Z_OUT are the values at the X, Y, and Z axis respectively. Place the module face up: Z_OUT can reach +1g at most, the minimum of X_OUT is -1g toward the Ax direction, and the minimum of Y_OUT is -1g toward the Ay direction. On the other hand, turn the module upside down: the minimum of Z_OUT is -1g, the maximum of X_OUT is +1g toward the Ax direction, and the maximum of Y_OUT is +1g toward the Ay direction. , as shown below. Rotate the ADXL345 module and you'll see the change of three values.



Relationship between output and gravity direction

Pin Function of ADXL345 Module

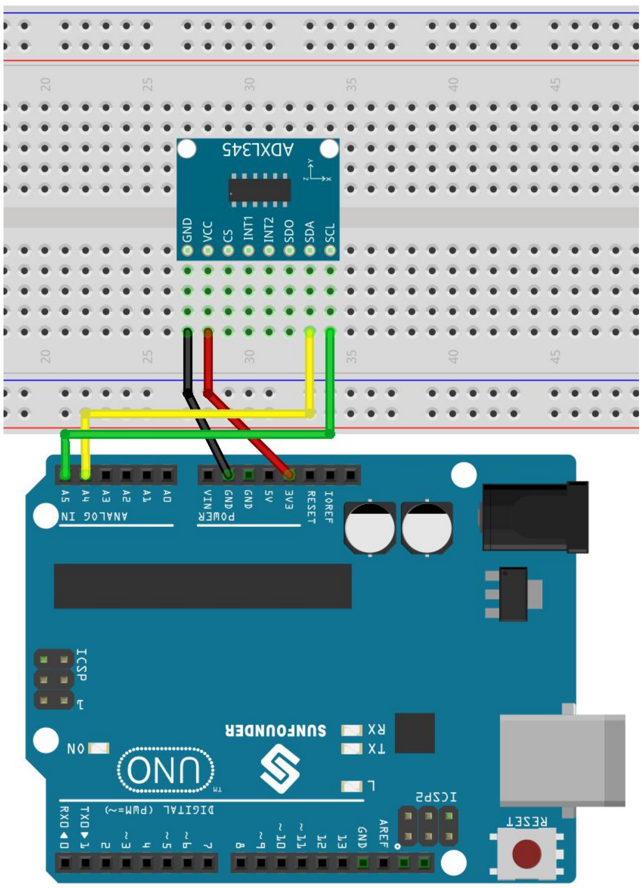
Name	Description
GND	GND
VCC	Supply Voltage，3.3V/5V.
CS	Chip Select I2C mode is enabled if the CS pin is tied high to VDD I/O (VDD I/O = 1.8V), so the I2C mode is set as default for the module.
INT1	Interrupt 1 Output
INT2	Interrupt 2 Output
SDA	Serial Data (I2C), Serial Data In (SPI 4-Wire), Serial Data In/Out (SPI 3-Wire)
SCL	Serial Communications Clock

Since SDO has been connected to GND, the I2C address of the ADXL345 is 0x53, 0xA6 for write, 0xA7 for read.

Experimental Procedures

Step 1: Build the circuit.

ADXL345	Uno/Mega2560
3.3V	3.3V
GND	GND
SCL	A5 Uno/ pin 21 Mega2560
SDA	A4 Uno/pin 20 Mega2560



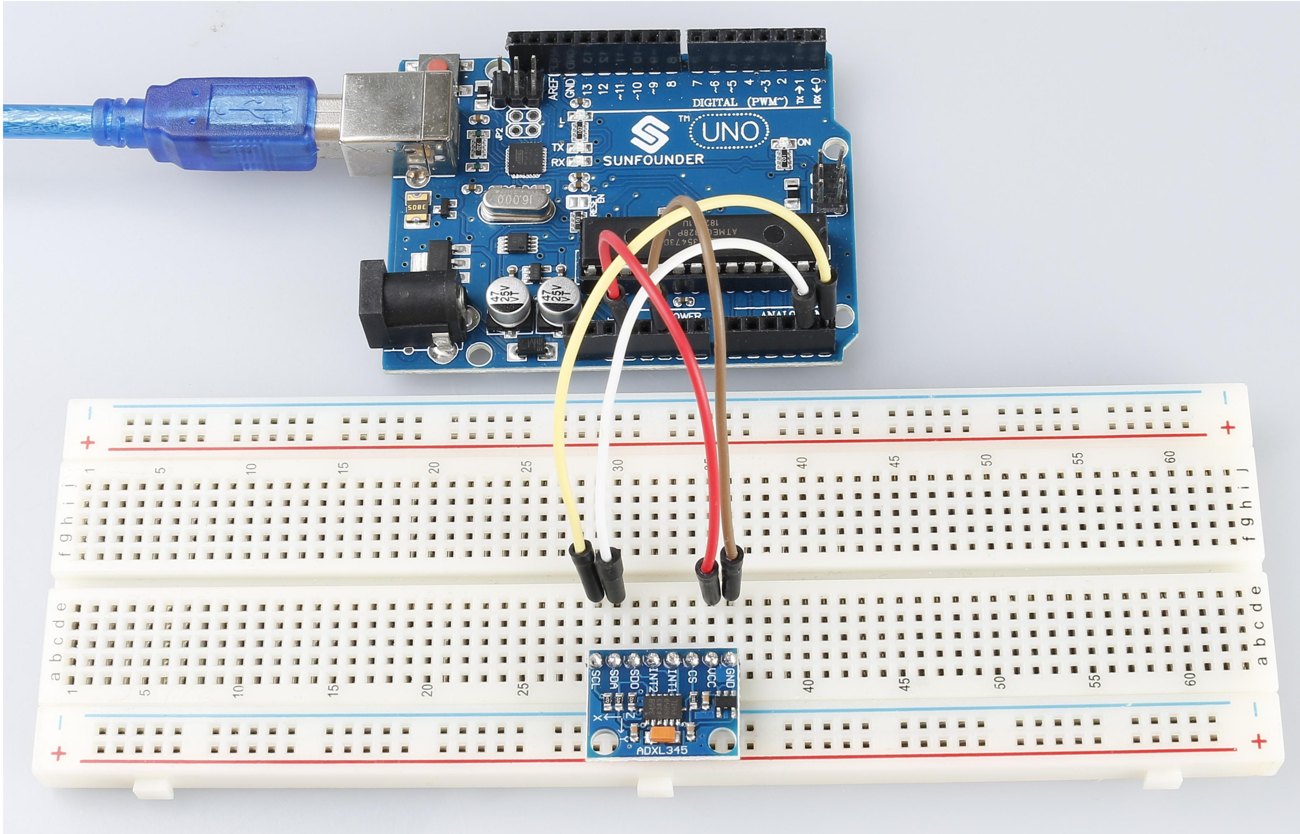
Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

After uploading, open Serial Monitor, where you can see the data detected.

When the acceleration of the module changes, the figure will change accordingly on the window.



Lesson 16 Simple Creation - Light Alarm

Introduction

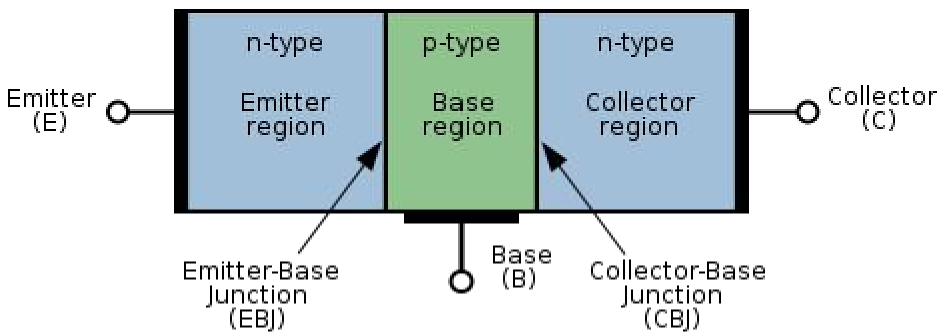
This experiment is really interesting – to apply a DIY phototransistor. DIY phototransistors use the glow effect and photoelectric effect of LEDs – they will generate weak currents when some light is shined on it. And we use a transistor to amplify the currents generated, so the SunFounder Uno board can detect them.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- 1 * Passive buzzer
- 1 * Resistor (10KΩ)
- 1 * LED
- 1 * NPN Transistor S8050
- Jumper wires

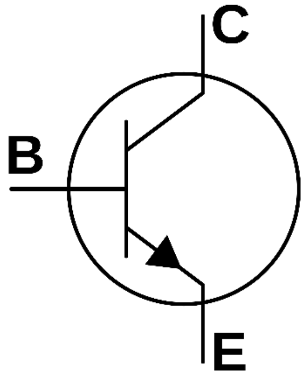
Principle

With the photoelectric effect, LEDs generate weak currents when exposed to light waves.



NPN consists of a layer of P-doped semiconductor (the "base") between two N-doped layers (see the picture above). A small current entering the base is amplified to produce a large collector and emitter current. That is, when there is a positive potential difference measured from the emitter of an NPN transistor to its base (i.e., when the base is high relative to the emitter) as well as positive potential difference measured from the base to the collector, the transistor becomes active. In this "on" state, current flows between the collector and emitter of the transistor.

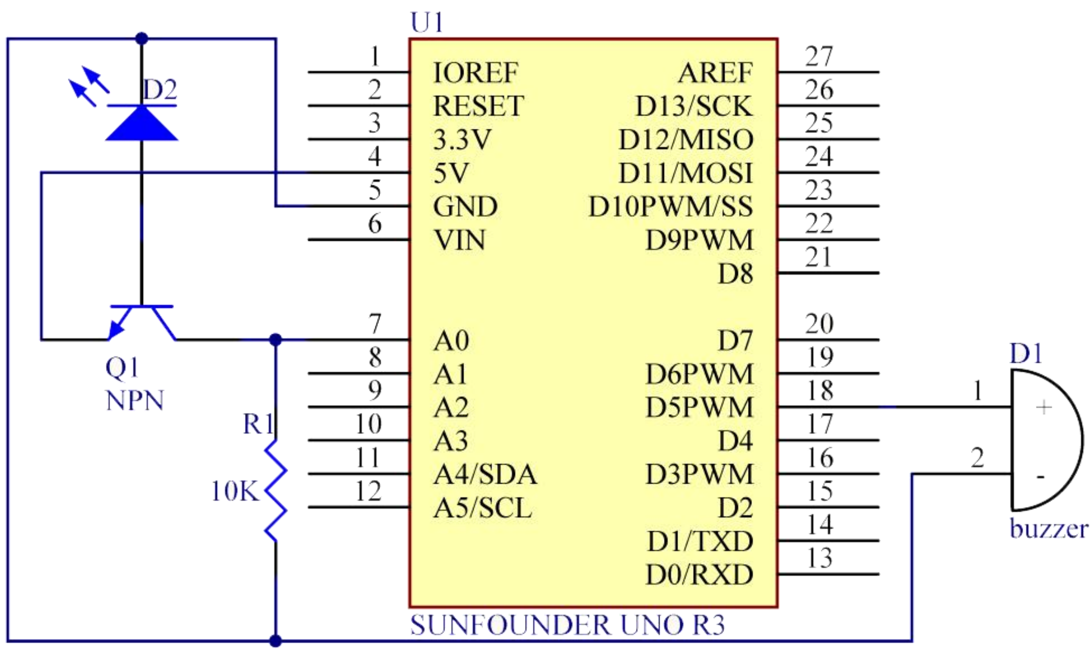
There are three poles for the regions: base (b), emitter (e) and collector (c). They form two P-N junctions, namely the base-emitter junction and collector-base junction. The arrows in the schematic symbol of an NPN (see the figure below) indicates the direction of the base-emitter junction.



We can see the two PN junctions with unilateral conductivity inside the transistor, which enables it a switch component.

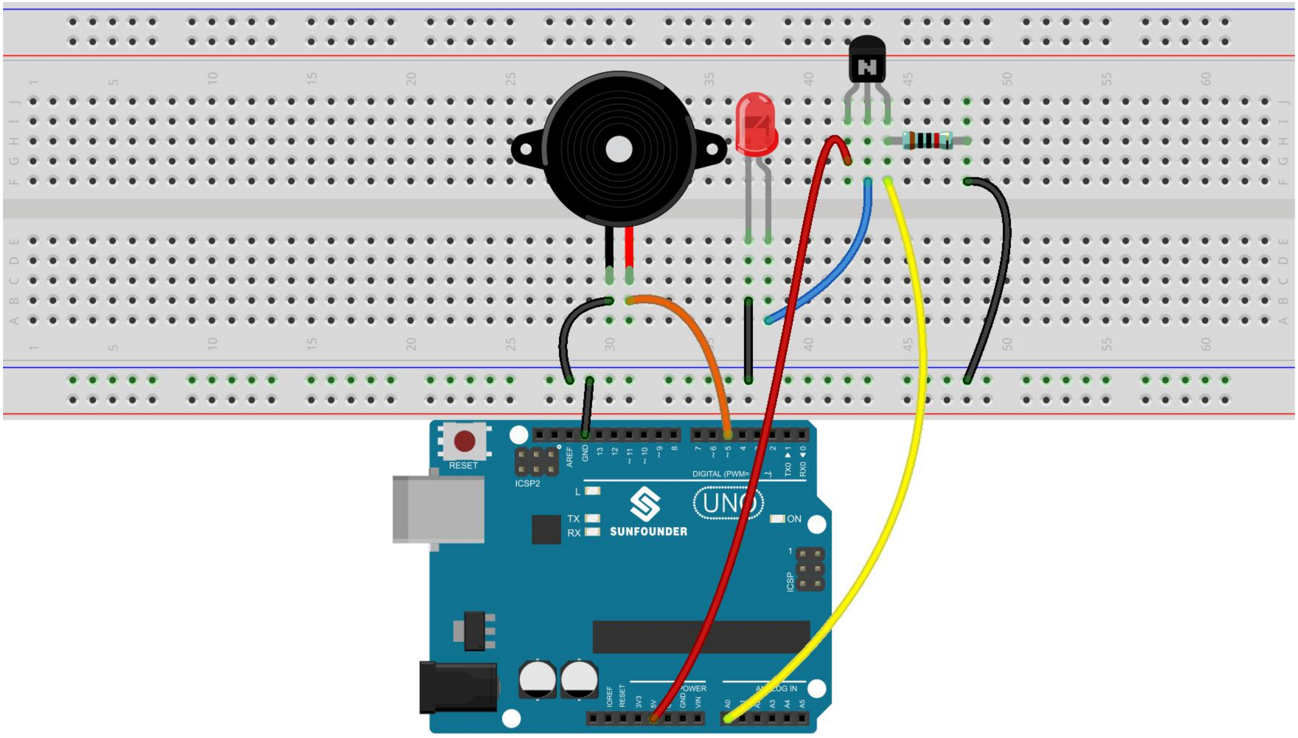
A 10kΩ pull-down resistor is attached to the transistor output stage in order to avoid analog port suspending to interfere with signals and cause misjudgment.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

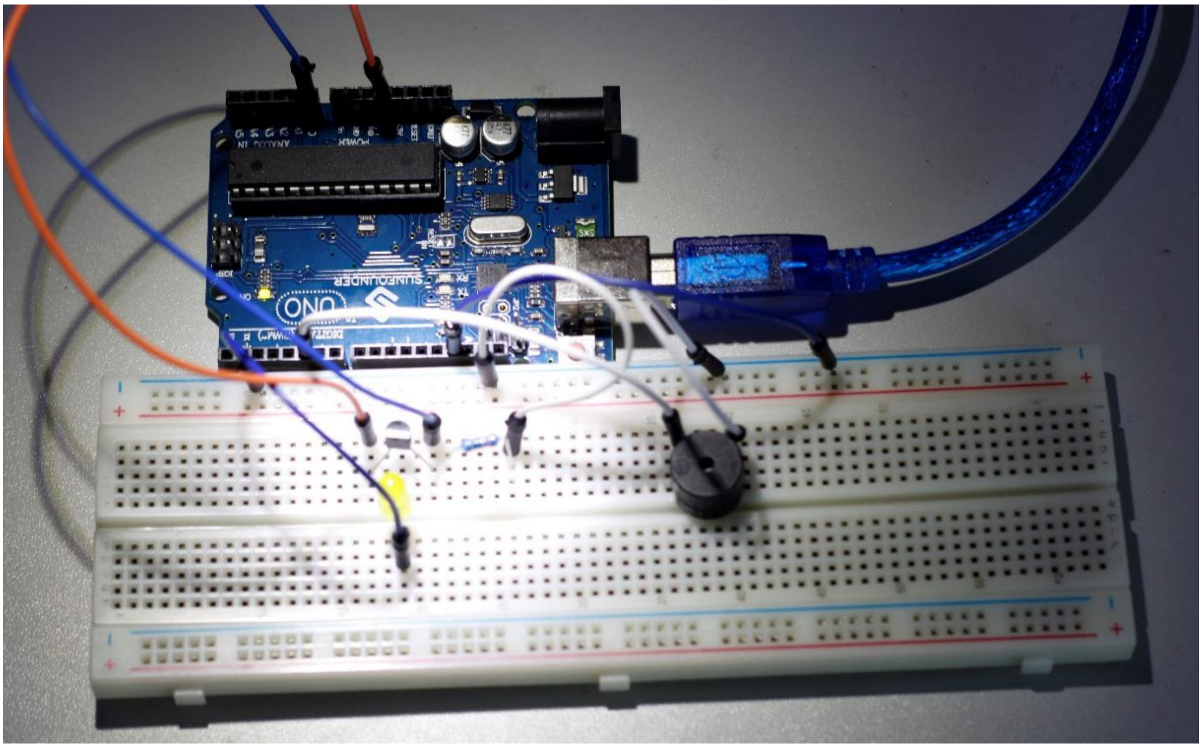


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Now, you can hear the buzzer make sounds when the LED is shined.



Lesson 17 Simple Creation - Traffic Light

Introduction

In this lesson, we will continue to make amusing experiment – simulate a traffic light. The colored lights of a traffic light are typically red for stop, green for go, and yellow for proceed with caution. We'll use LEDs for the indicators. Let's get started!

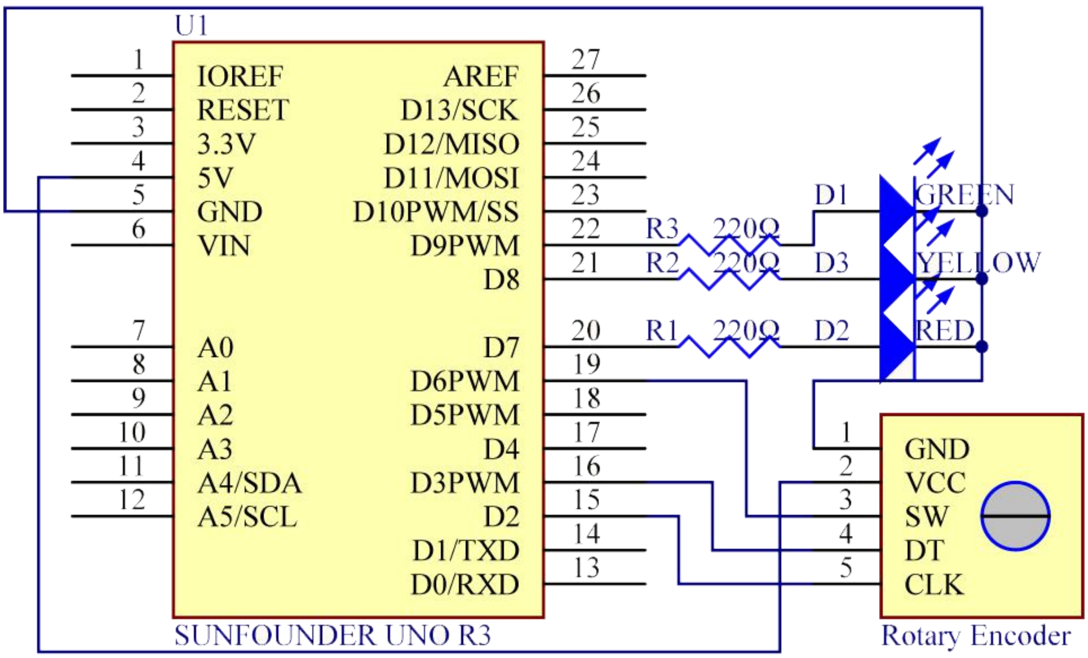
Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- 1 * Rotary encoder
- 3 * LED
- 3 * Resistor (220Ω)
- Jumper wires

Principle

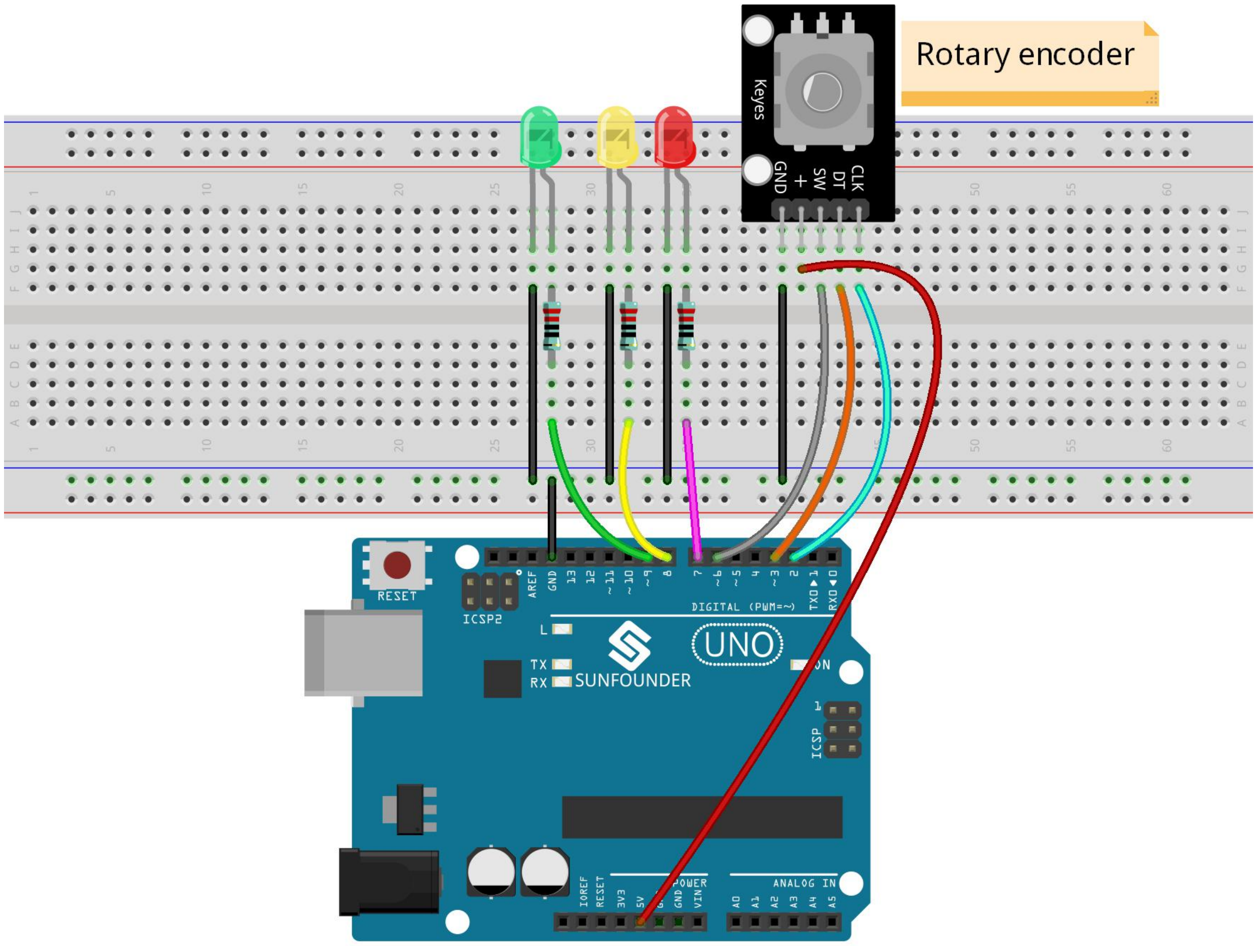
For a real traffic light, the time length for displaying red and green is much longer than yellow. As a result, we define two cycles with code: a short and a long one. In the short cycle, the traffic light changes its order at a rate of roughly once per second. On the other hand, the long cycle is changed by the rotary encoder which determines the time length of red and green.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

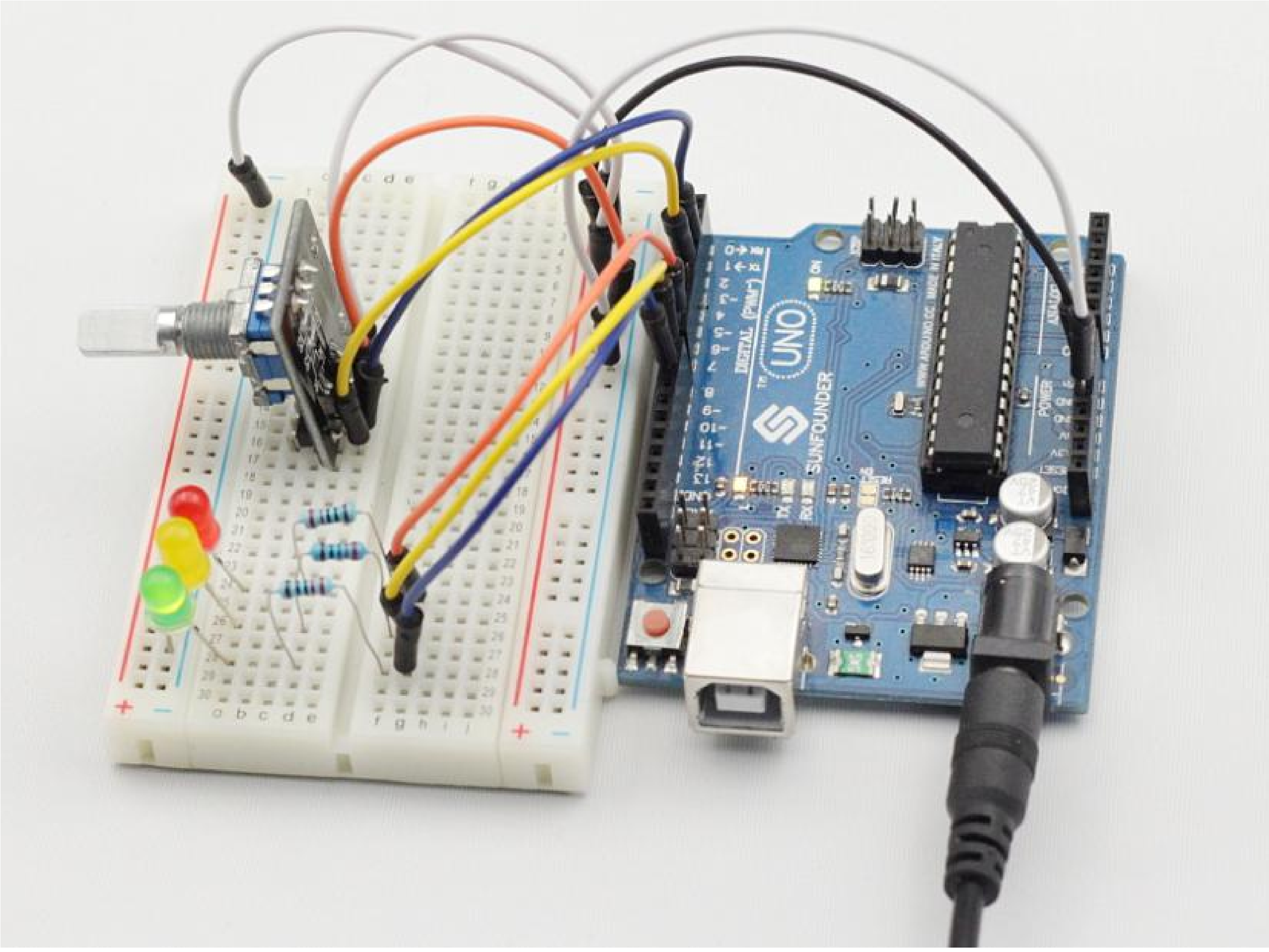


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

Here you should see the red LED light up first, then the red LED and yellow LED, followed by the green LED, and finally the yellow LED again.



Lesson 18 Simple Creation - Digital Dice

Introduction

In previous experiments, we learned how to use a 7-segment display and control LEDs by a button. In this lesson, we will use a 7-segment display and a button together to create a simple digital dice.

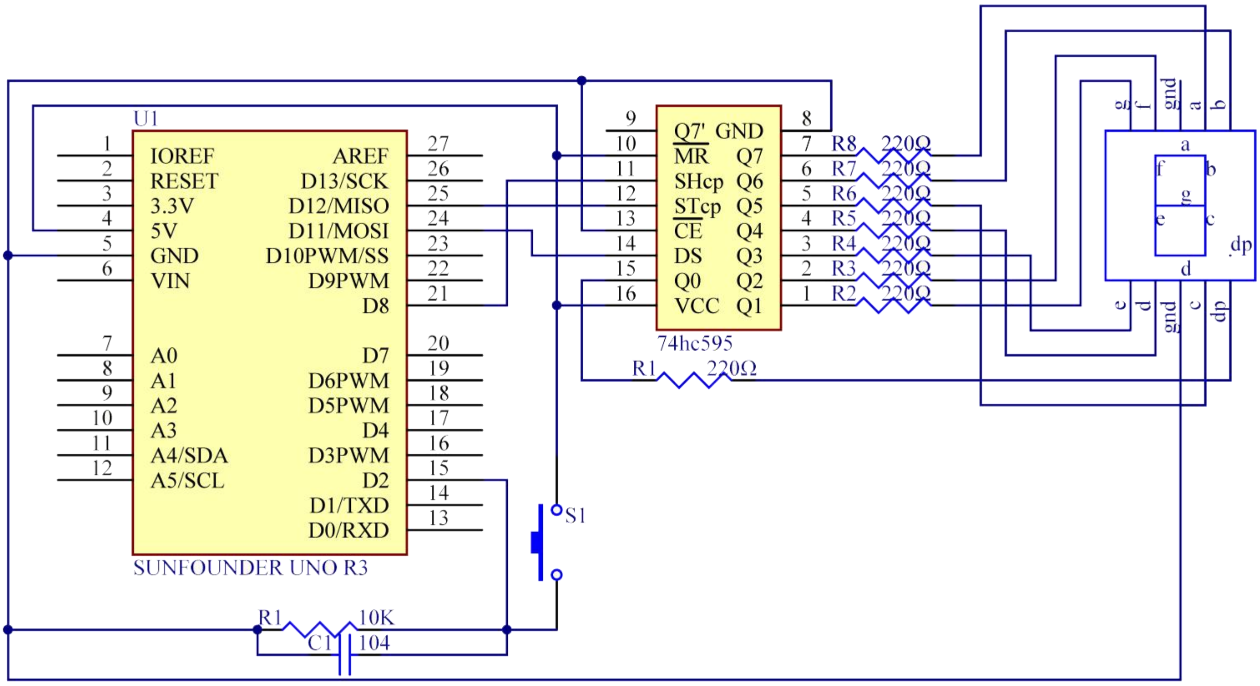
Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- 1 * Button
- 1 * Resistor (10KΩ)
- 8 * Resistor (220Ω)
- 1 * 7-segment display
- 1 * 104 ceramic capacitor
- 1 * 74HC595
- Jumper wires

Principle

The idea behind a digital dice is very simple: a 7-segment display circularly jumps from 1 to 6 rapidly. When the button is pressed, the jumping will slow down until it stops on a number three seconds later. When the button is pressed again, the process will repeat.

The Schematic Diagram

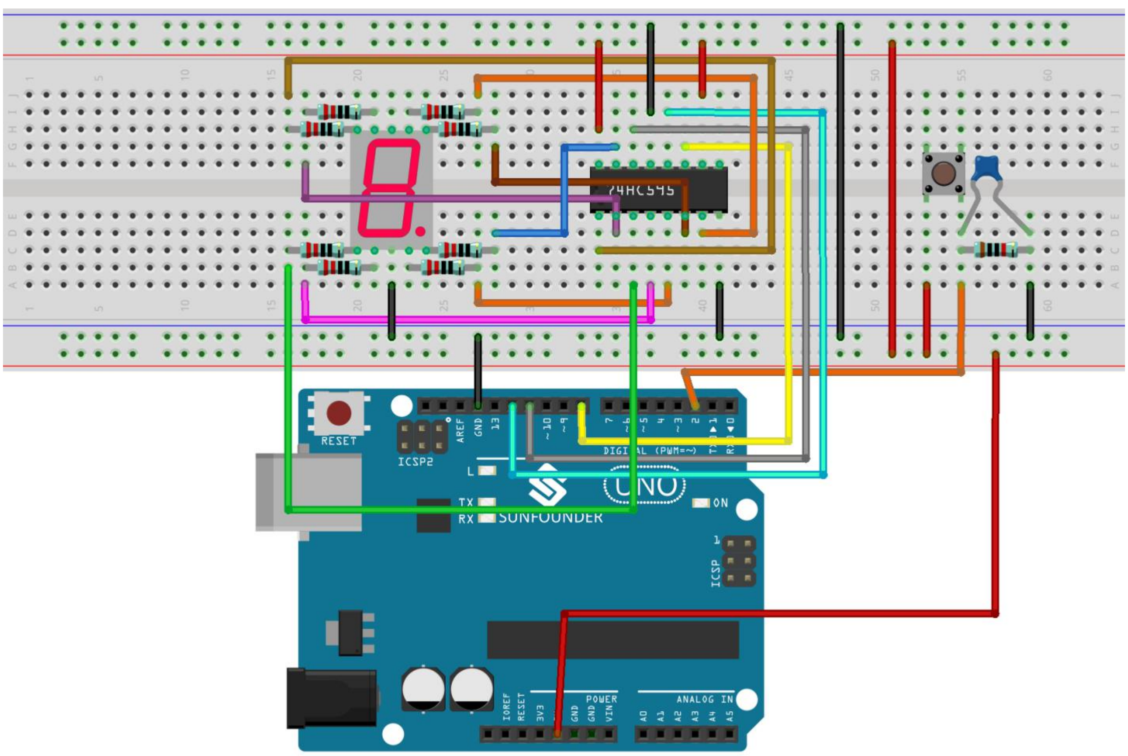


Experimental Procedures

Step 1: Build the circuit.

For the wiring between 74HC595, 7-segment and SunFounder Uno, please refer to Lesson 11 74HC595.

For the button, connect one end of the button to 5V, the other to one pin of the 10 KΩ resistor and the capacitor, and then connect the other terminal of the resistor and capacitor to GND.

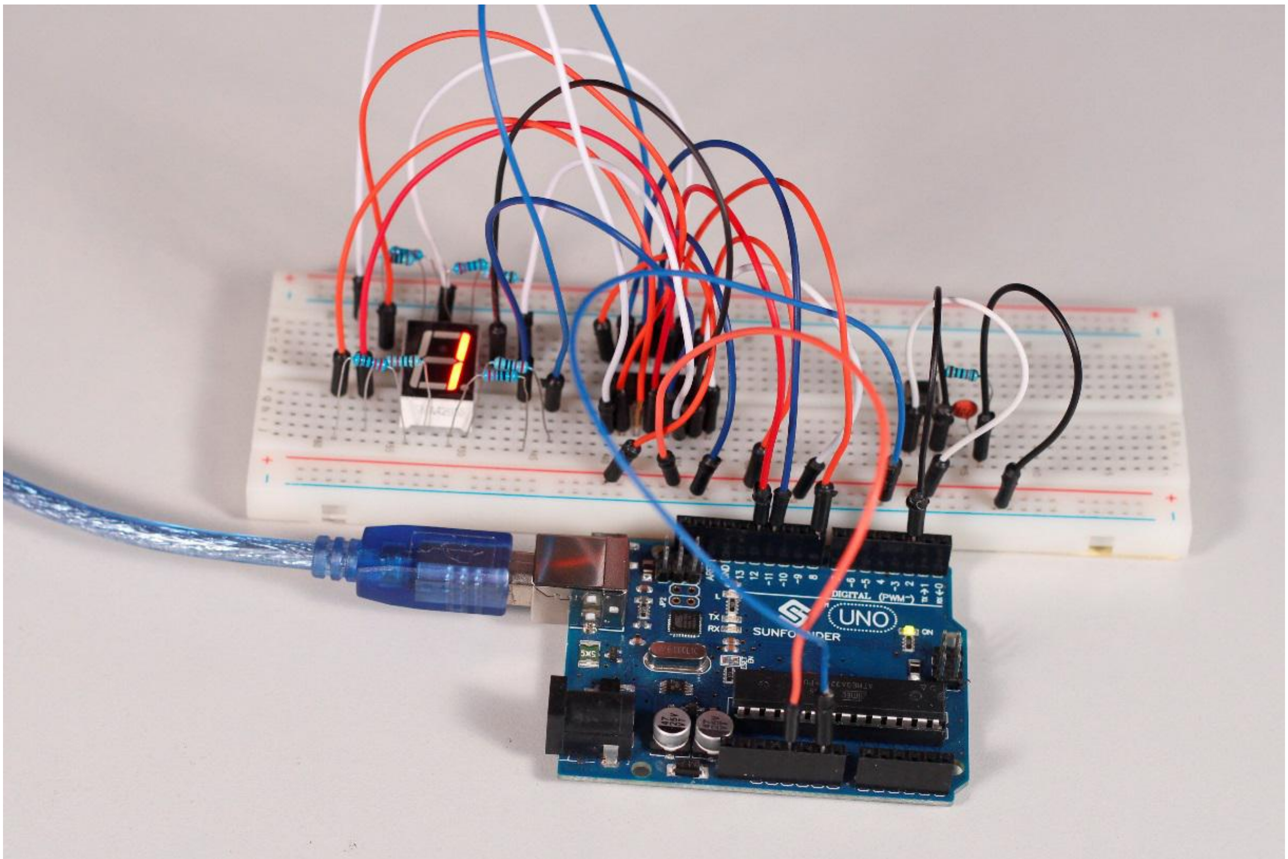


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

You can now see the 7-segment display jump between numbers from 1 to 6. Press the button, and the jumping will slow down until it stops three seconds later. Press the button again, and the process will repeat.



Lesson 19 Simple Creation - Small Fan

Introduction

In this experiment, we will use a button and a DC motor together to assemble a small fan. With the button, we can shift the rotational speed or gear selection of the fan.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * USB cable
- 1 * DC motor
- 1 * Resistor (10KΩ)
- 1 * Button
- 1 * Motor drive L293D
- 1 * 104 Ceramic Capacitor
- 1 * Fan
- Jumper wires

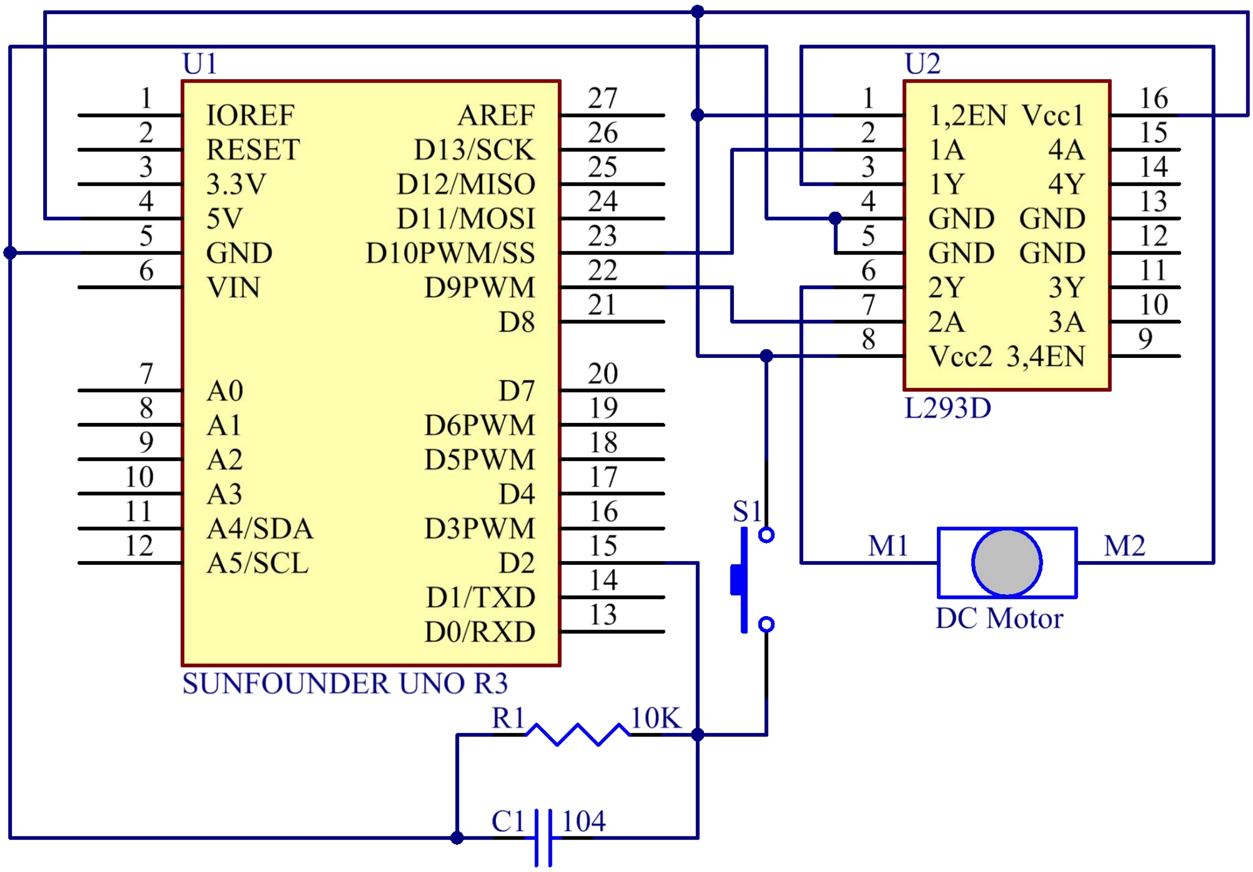
Principle

This experiment is to control the rotational speed of the motor by reading the times the button is pressed.

The chip L293D for motor drive, many details have been provide in Lesson 7. You may check the principle if necessary. The capacitor and resistor here connected to the button in parallel is to eliminate jitters and output a steady low level, so as to protect the motor.

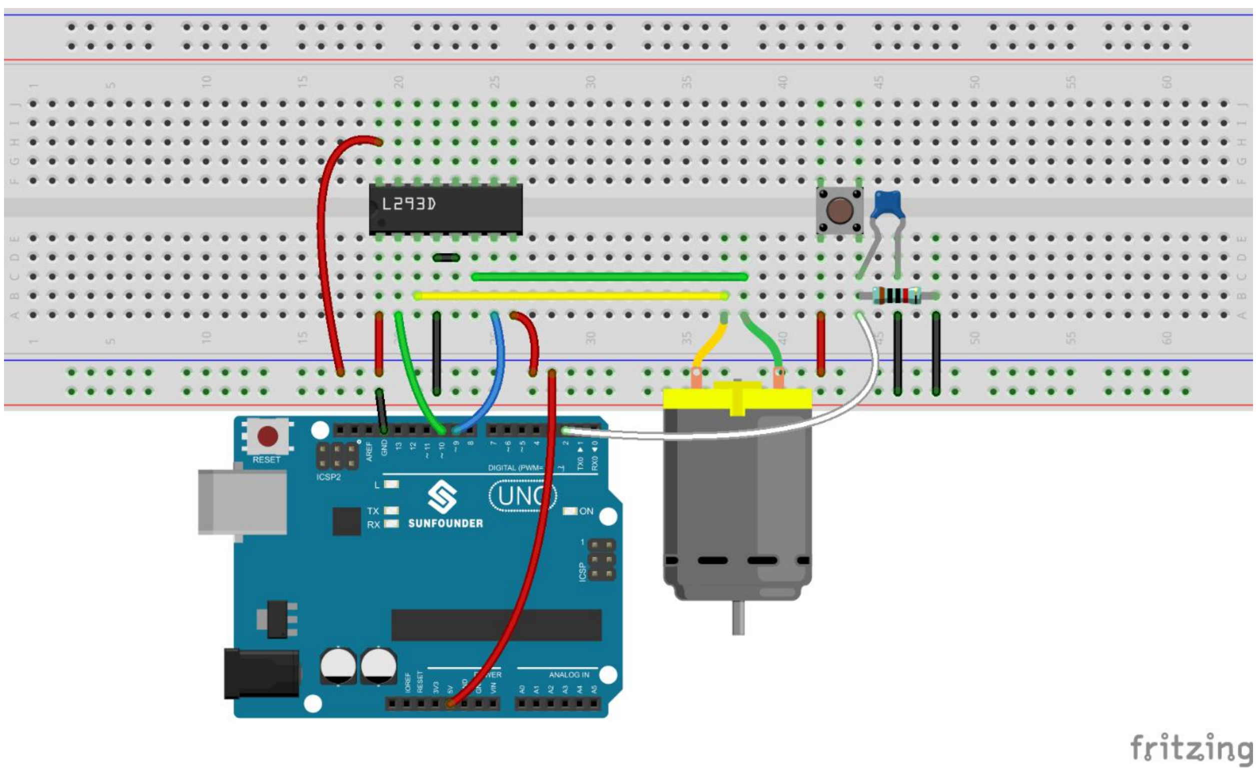
The fan has 4 gears. After startup, the default gear is 0 and the fan will not rotate. If the button is pressed once, the small fan will get into first gear (low speed) and rotate slowly; for pressing twice, it will get into the second gear (medium speed) and rotate at a medium pace; for pressing three times, it will get into third gear (high speed) and rotate rapidly. If the button is pressed four times, it will return to 0 gear and stop.

The Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

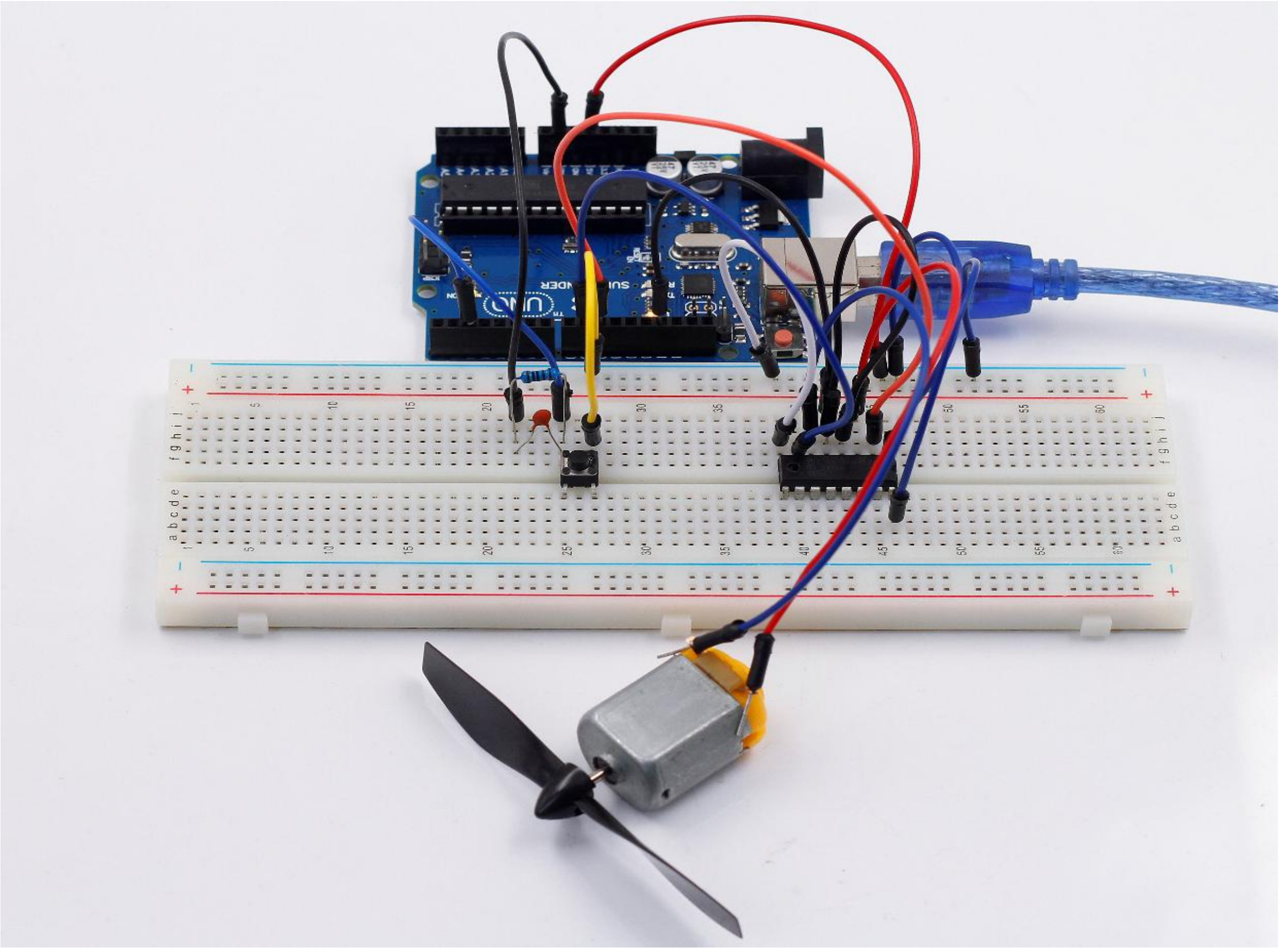


Step 2: Open the code file.

Step 3: Select correct Board and Port.

Step 4: Upload the sketch to the SunFounder Uno board.

As mentioned above, the times you press the button will change the rotational speed of the fan. Pressing it once will cause it to rotate slowly, while three times will cause it to rotate rather quickly, and pressing it four times will cause it to stop.



Copyright Notice