LROBRUYA

LTARK-46

Content

Preface

Company Profile

Founded in 2014, Shenzhen Lonten Technology Co., Ltd. focuses on the design, research production of Electronics Module for robotics related products. Consisting of professional researchers and skilled engineers, our R&D team constantly strives for creative function and excellent user experience. The company's R&D investments on arduino kits raspberry pi kits, as well as 3D printer and robots that back up STEAM education.

Customer Service

Our self-owned factory is certificated with BSCI and SO, covering an area of 5,000 square meters, and achieving an annual production capacity of over 10,000 units. Our products are all certified to CE, FCC, and ROHS standards, have exported to more than 100 countries including, but not limited to France, the United States of America, Australia, Russia, the United Kingdom, Germany, Singapore, Egypt, and India, bringing technological innovation to all walks of life.

Install Arduino IDE

Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.

In this Project, you will learn how to setup your computer to use Arduino and how to set about the Projects that follow.

The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to https://www.arduino.cc/en/software.

The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

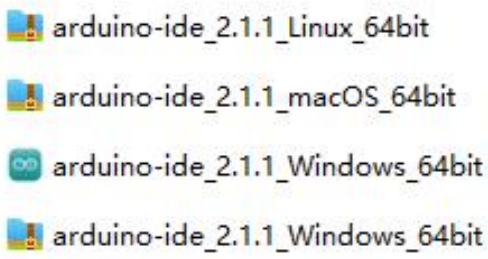
STEP2：Download the development software that is compatible with the operating.

system of your computer. Take Windows as an example here.

Click Windows Win 10 and newer,64 bits.

Click JUST DOWNLOAD.

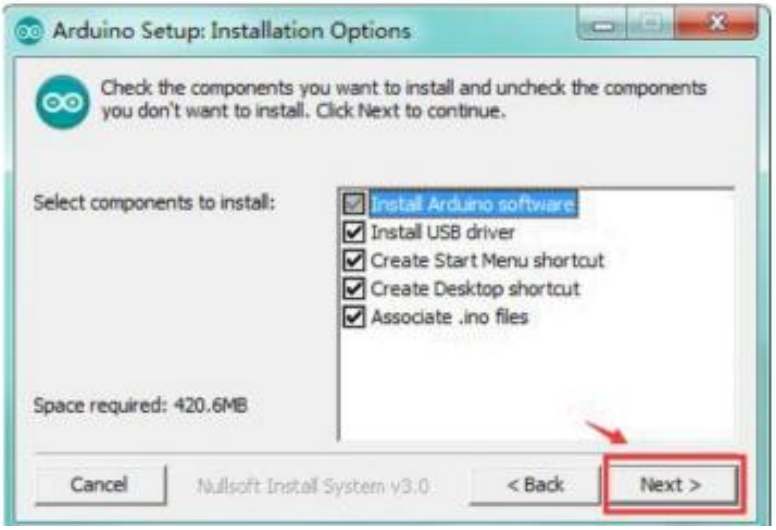
Also version 2.1.1 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.



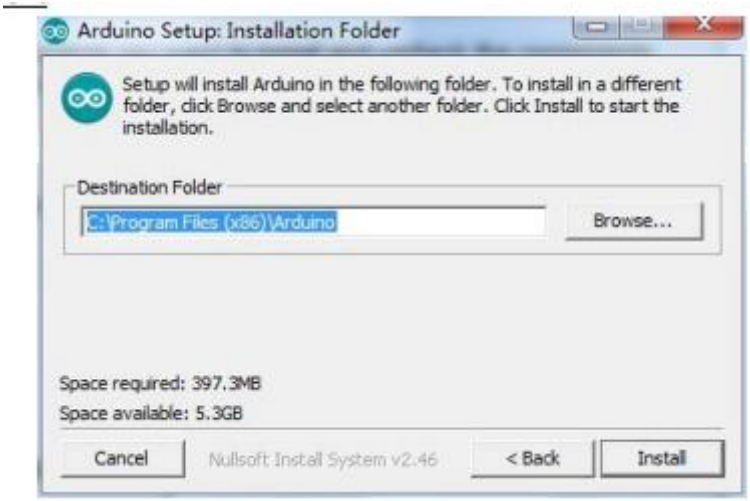
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

Click I Agree to see the following interface.

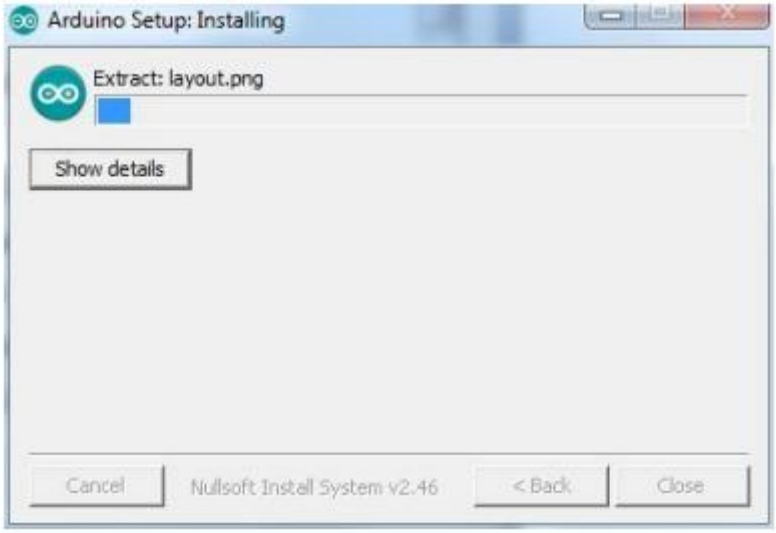


Click Next



You can press Browse… to choose an installation path or directly type in the directory you want.

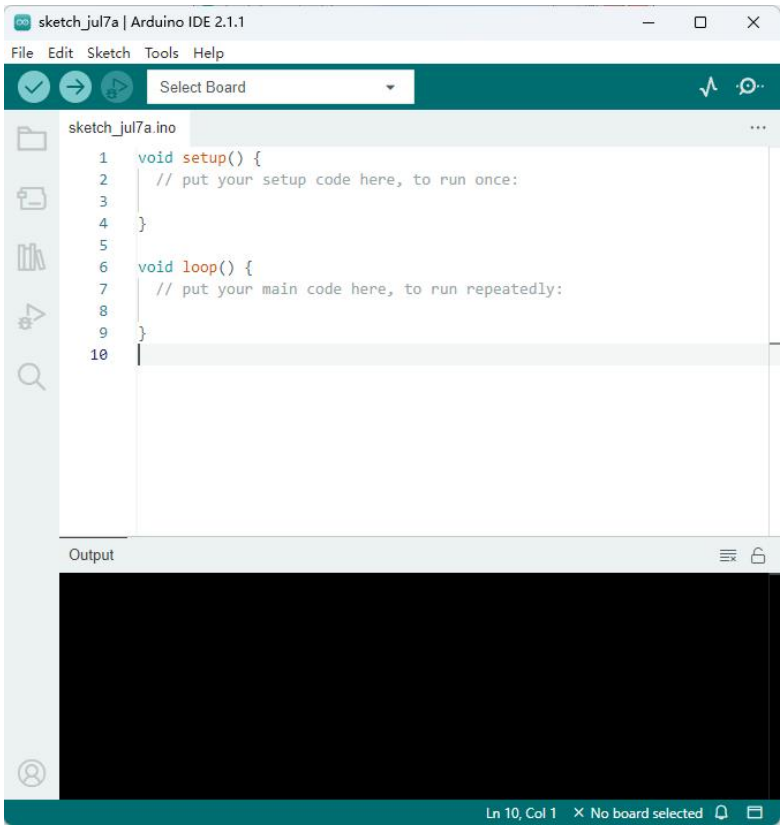
Click Install to initiate installation



Finally, the following interface appears, click Install to finish the installation.

Next, the following icon appears on the desktop

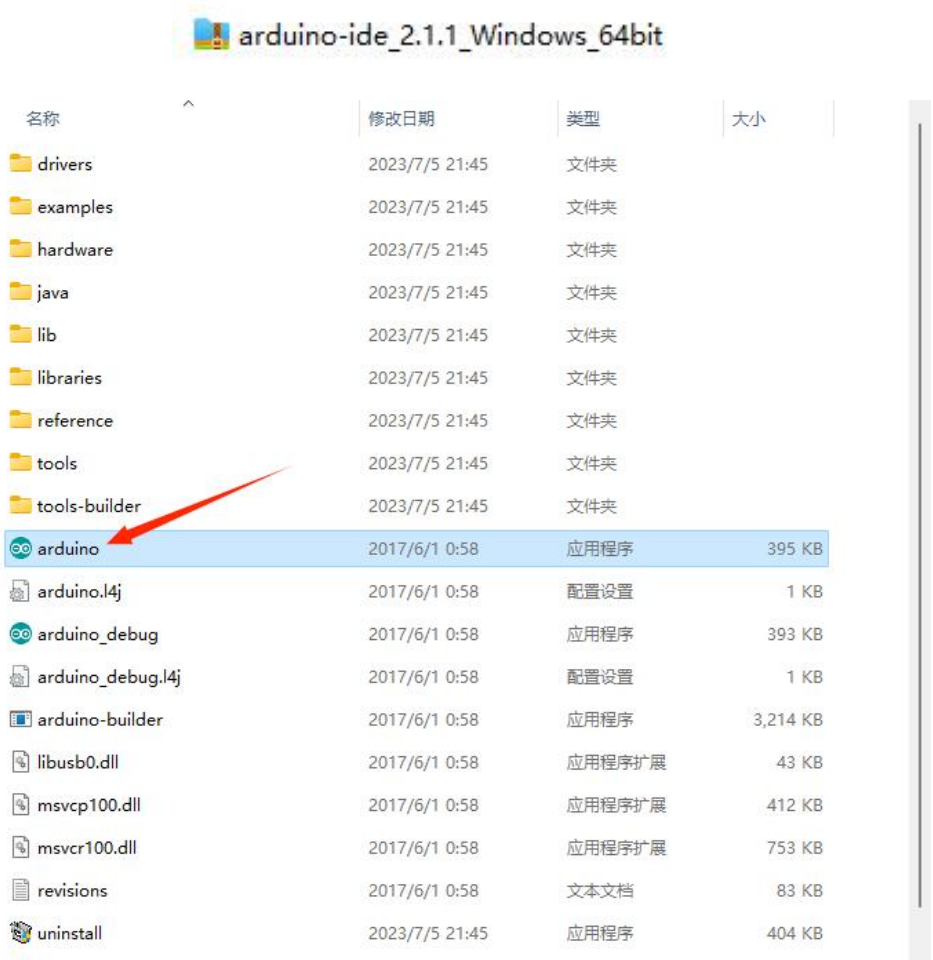
Double-click to enter the desired development environment

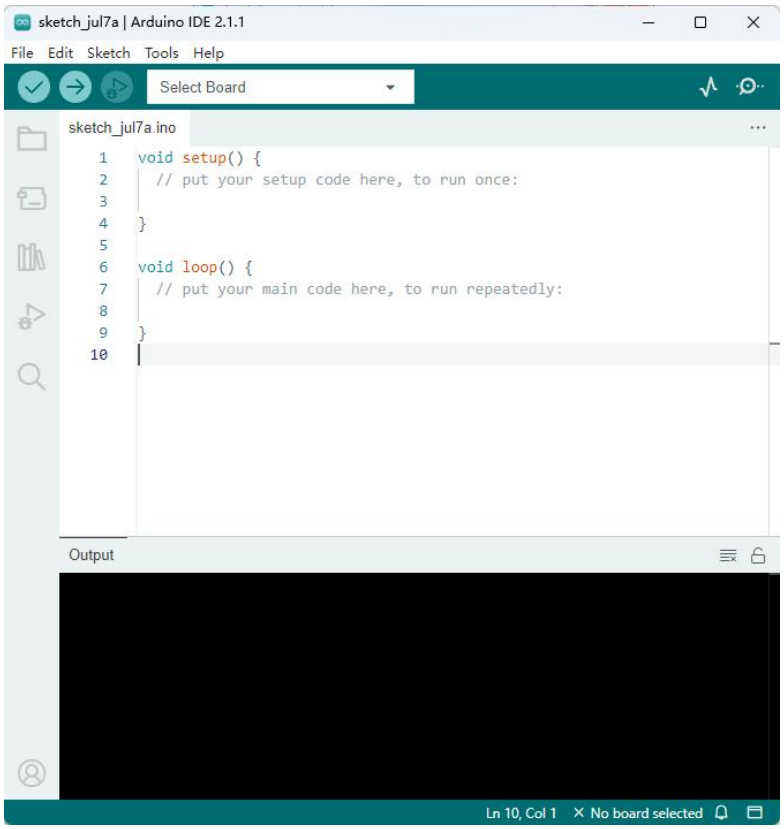


You may directly choose the installation package for installation and

skip the contents below and jump to the next section. But if you want to learn some methods other than the installation package, please continue to read the section.

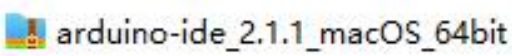
Unzip the zip file downloaded, Double-click to open the program and enter the desired development environment.





Installing Arduino (Mac OS X)

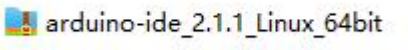
Download and Unzip the zip file, double click the Arduino.app to enter Arduino IDE; the system will ask you to install Java runtime library if you don't have it in your computer. Once the installation is complete you can run the Arduino IDE.



Installing Arduino (Linux)

You will have to use the make install command. If you are using the Ubuntu system, it is recommended to install Arduino IDE from the software center of Ubuntu.

LROBRUYA



Add Libraries and Open Serial Monitor

Installing Additional Arduino Libraries

Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

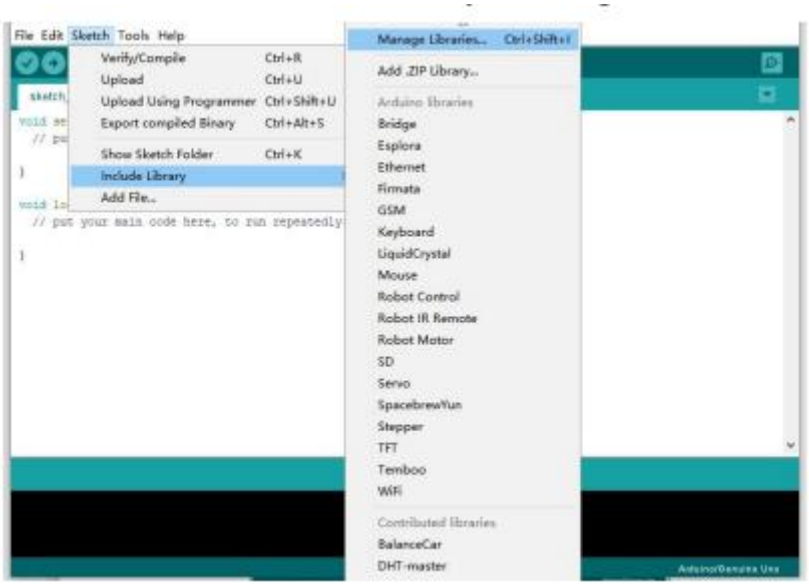
What are Libraries?

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in Liquid Crystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

How to Install a Library

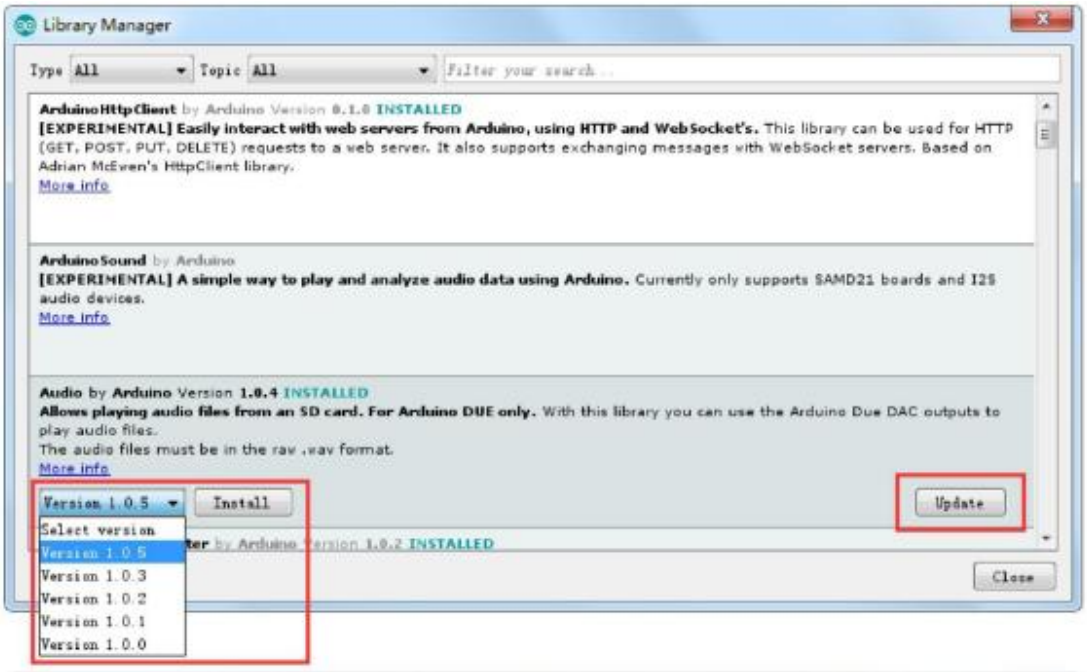
Using the Library Manager

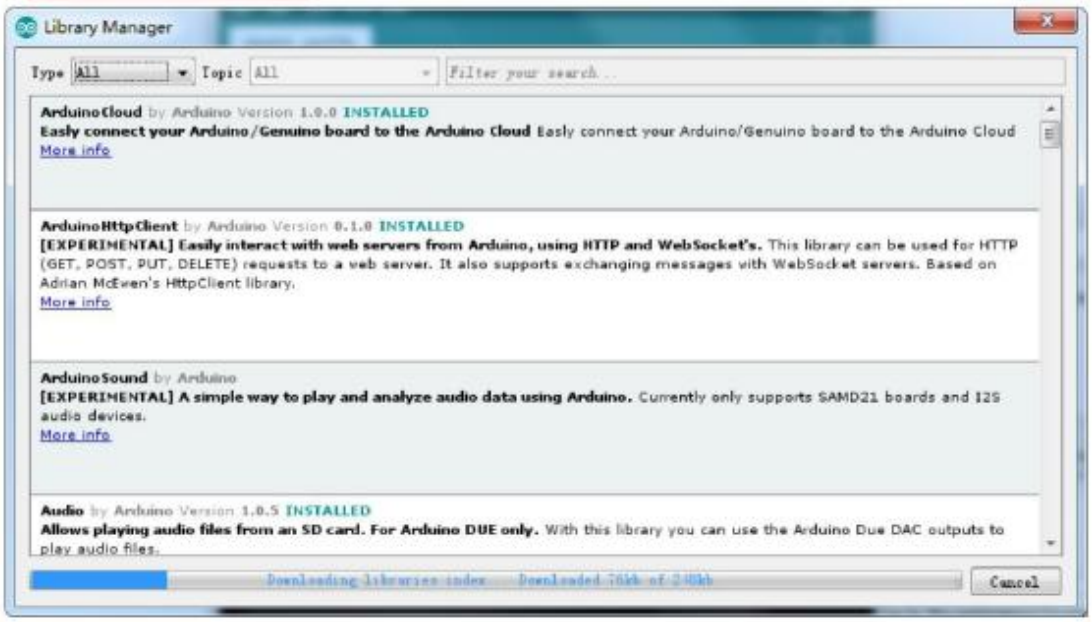
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.0). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



Then the library manager will open and you will find a list of libraries that are already installed or ready for installation. In this example we will install the Bridge library. Scroll the list to find it, then select the version of the library you want to install. Sometimes only one version of the library is available. If the version selection menu does not appear, don't worry: it is normal.

There are times you have to be patient with it, just as shown in the figure. Please refresh it and wait.

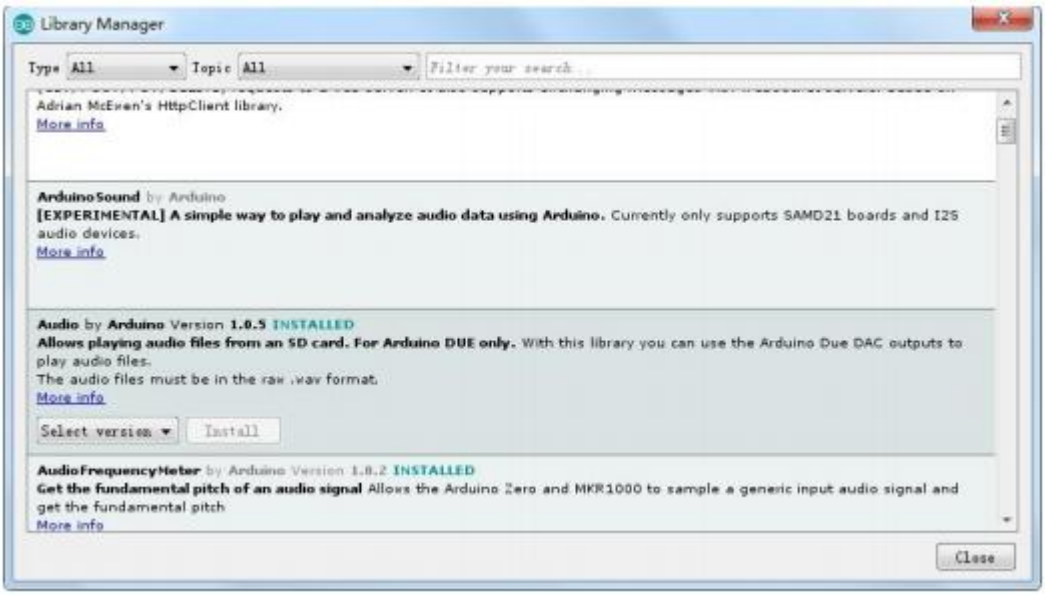




Finally click on install and wait for the IDE to install the new library.

Downloading may take time depending on your connection speed. Once it has finished, an Installed tag should appear next to the Bridge library.

You can close the library manager.



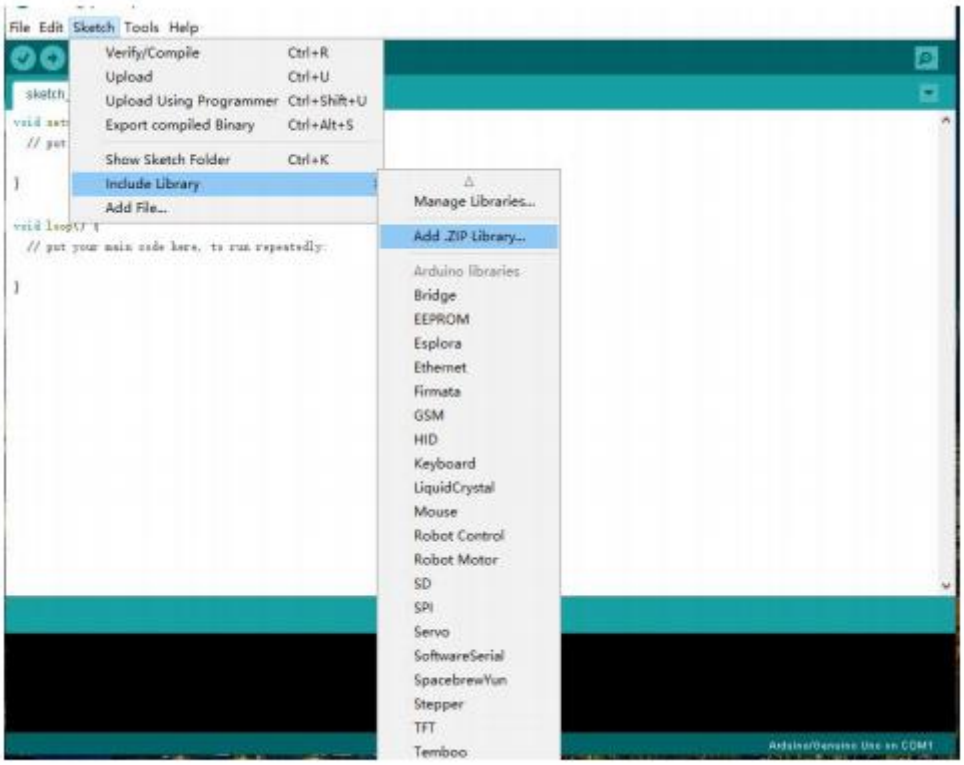
You can now find the new library available in the Include Library menu.

If you want to add your own library open a new issue on Github.

Importing a .zip Library

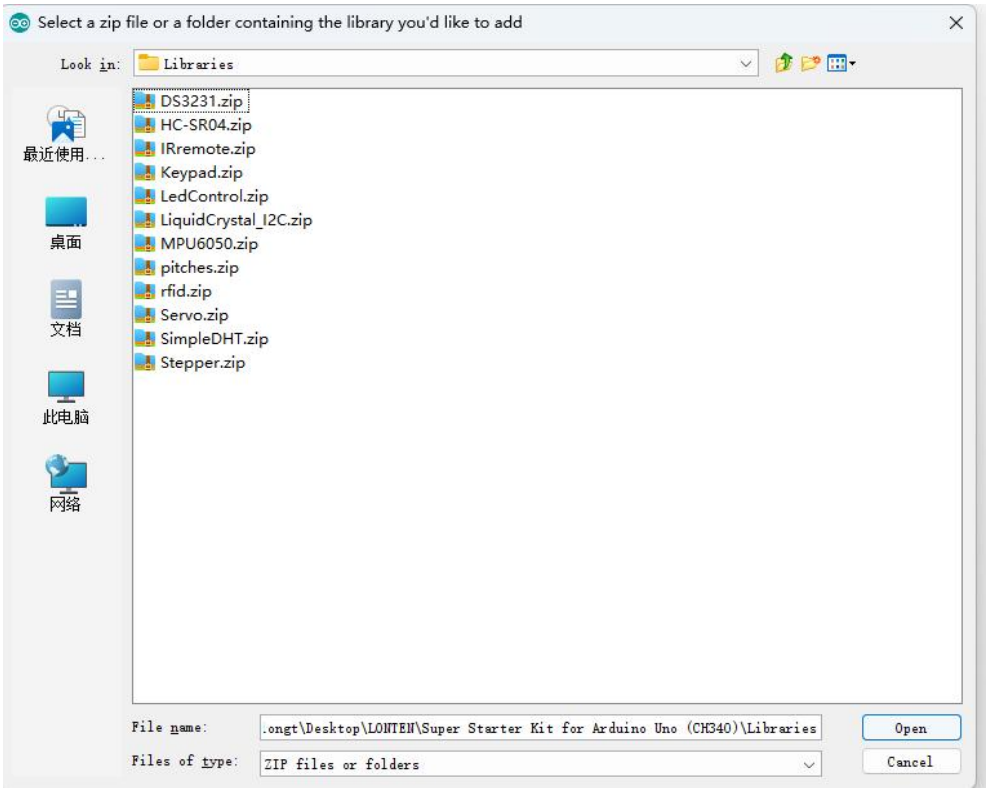
Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. Starting with version 1.0.5, you can install 3rd party libraries in the IDE. Do not unzip the downloaded library, leave it as is.

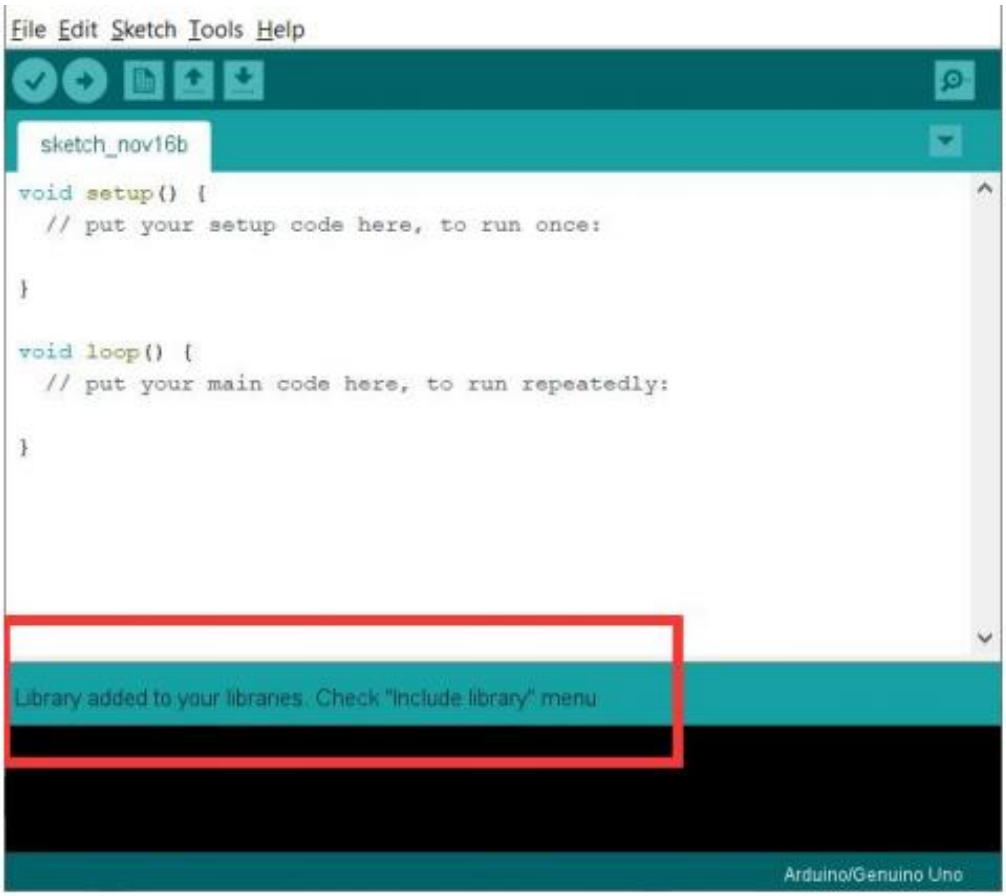
In the Arduino IDE, navigate to Sketch > Include Library. At the top of the drop down list, select the option to "Add .ZIP Library".



You will be prompted to select the library you would like to add.

Navigate to the .zip file's location and open it.


Select a zip file or a folder containing the library you'd like to add
Look in: Libraries
DS3231.zip
HC-SR04.zip
IRremote.zip
Keypad.zip
LedControl.zip
LiquidCrystal_I2C.zip
MPU6050.zip
pitches.zip
rfid.zip
Servo.zip
SimpleDHT.zip
Stepper.zip
最近使用...
桌面
文档
此电脑
网络
File name: .ongt\Desktop\LOUTEN\Super Starter Kit for Arduino Uno (CH340)\Libraries
Files of type: ZIP files or folders
Open
Cancel


File Edit Sketch Tools Help
sketch_nov16b
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
Library added to your libraries. Check "include library" menu
Arduino/Genuino Uno

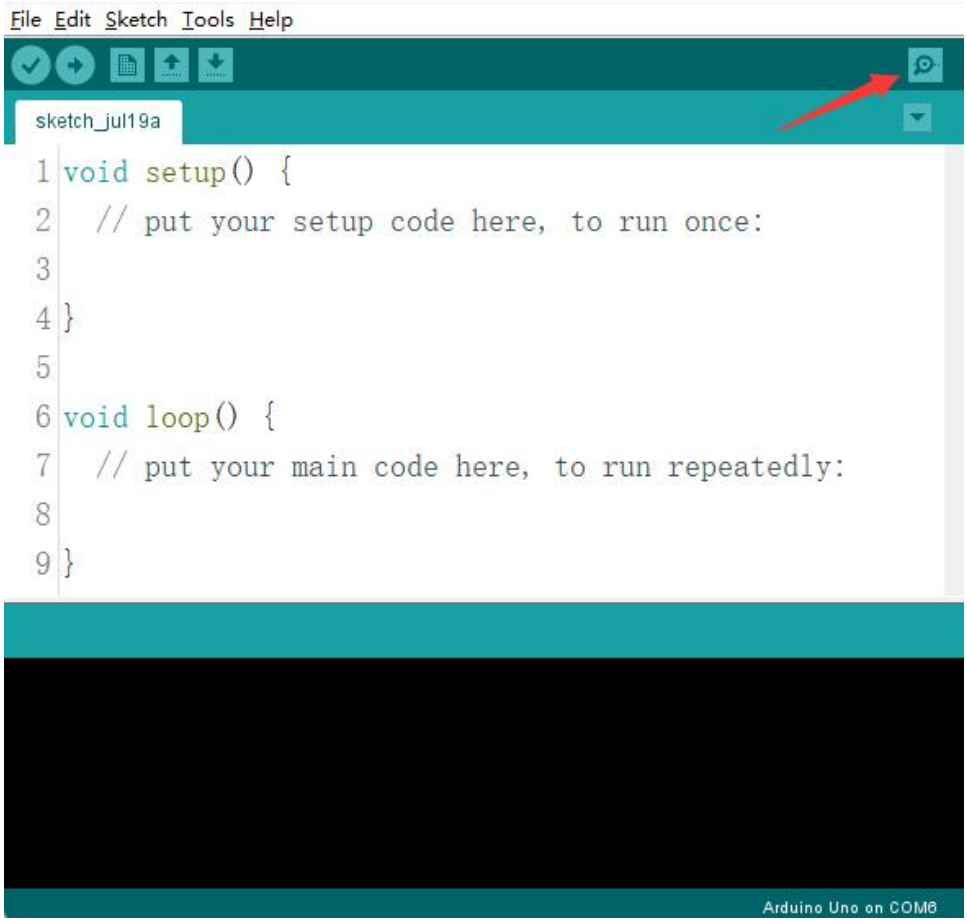
Return to the Sketch > Import Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory. NB: the Library will be available to use in sketches, but examples for the library will not be exposed in the File > Examples until after the IDE has restarted.

Arduino Serial Monitor (Windows, Mac, Linux)

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform. And, because using a terminal is such a big part of working withArduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is called the Serial Monitor.

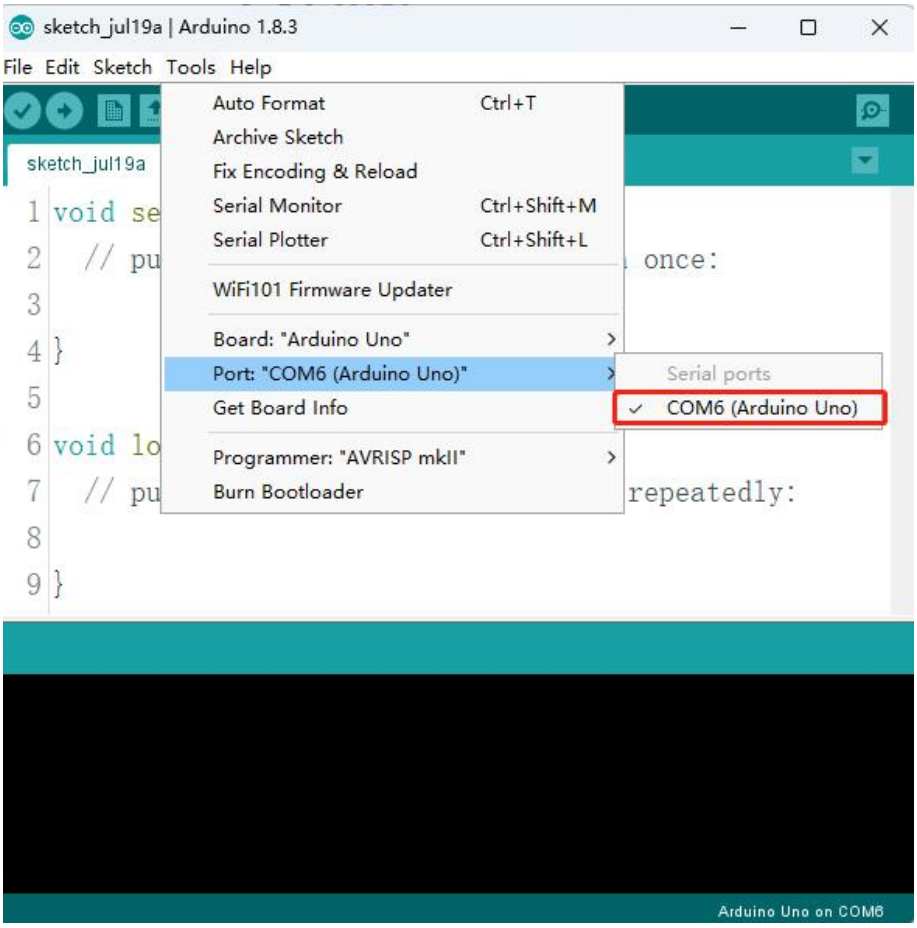
Making a Connection

Serial monitor comes with any and all version of the Arduino IDE. To open it, simply click the Serial Monitor icon.

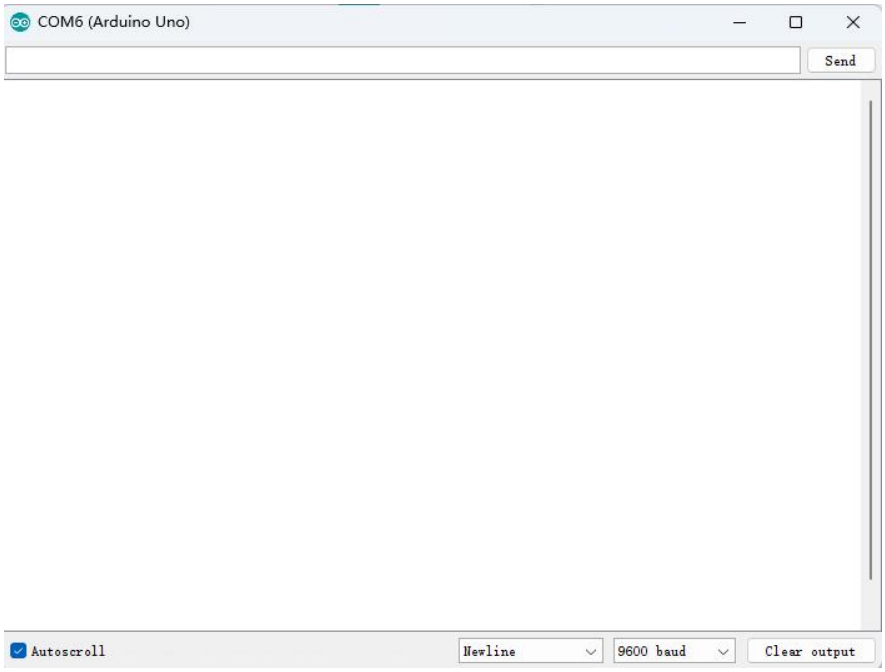


Selecting which port to open in the Serial Monitor is the same as selecting a port for uploading Arduino code. Go to Tools -> Serial Port, and select the correct port.

Tips: Choose the same COM port that you have in Device Manager.

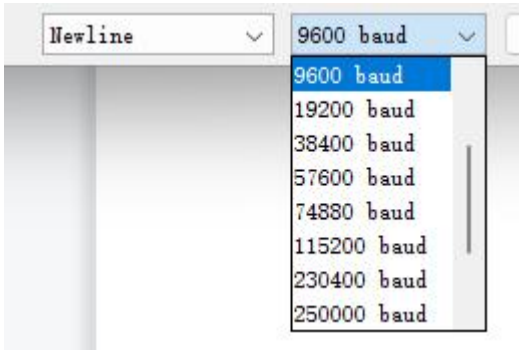


Once open, you should see something like this:



Settings

The Serial Monitor has limited settings, but enough to handle most of your serial communication needs. The first setting you can alter is the baud rate. Click on the baud rate drop-down menu to select the correct baud rate. (9600 baud)



Last, you can set the terminal to Autoscroll or not by checking the box in the bottom left corner.

Pros

The Serial Monitor is a great quick and easy way to establish a serial connection with your Arduino. If you're already working in the Arduino IDE, there's really no need to open up a separate terminal to display data.

Cons

The lack of settings leaves much to be desired in the Serial Monitor, and, for advanced serial communications, it may not do the trick.

Blink Test

Overview

In this Project, you will learn how to program your UNO R3 controller board to blink the Arduino's built-in LED, and how to download programs by basic steps.

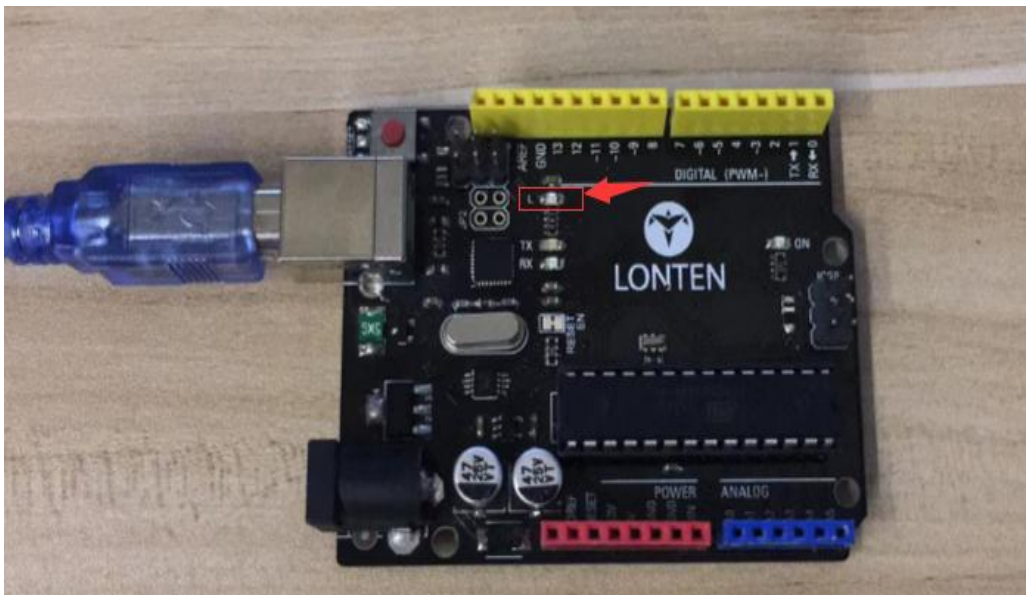
Component Required:

LONTEN Uno R3 Board* 1

Principle

The UNO R3 board has rows of connectors along both sides that are used to connect to several electronic devices and plug-in 'shields' that extends its capability.

It also has a single LED that you can control from your sketches. This LED is built onto the UNO R3 board and is often referred to as the 'L' LED as this is how it is labeled on the board.

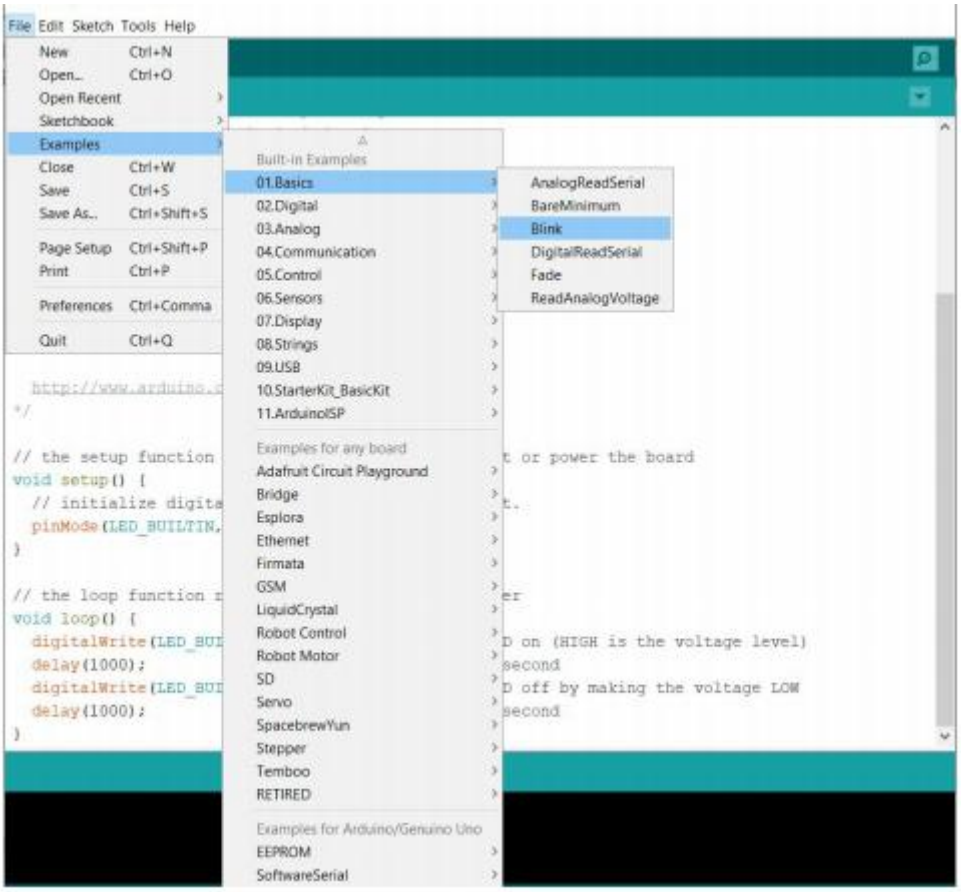


In this Project, we will reprogram the UNO board with our own Blink sketch and then change the rate at which it blinks.

In the previous chapter-How to install Arduino IDE, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO board. The time has now come to put that connection to the test and program your UNO board.

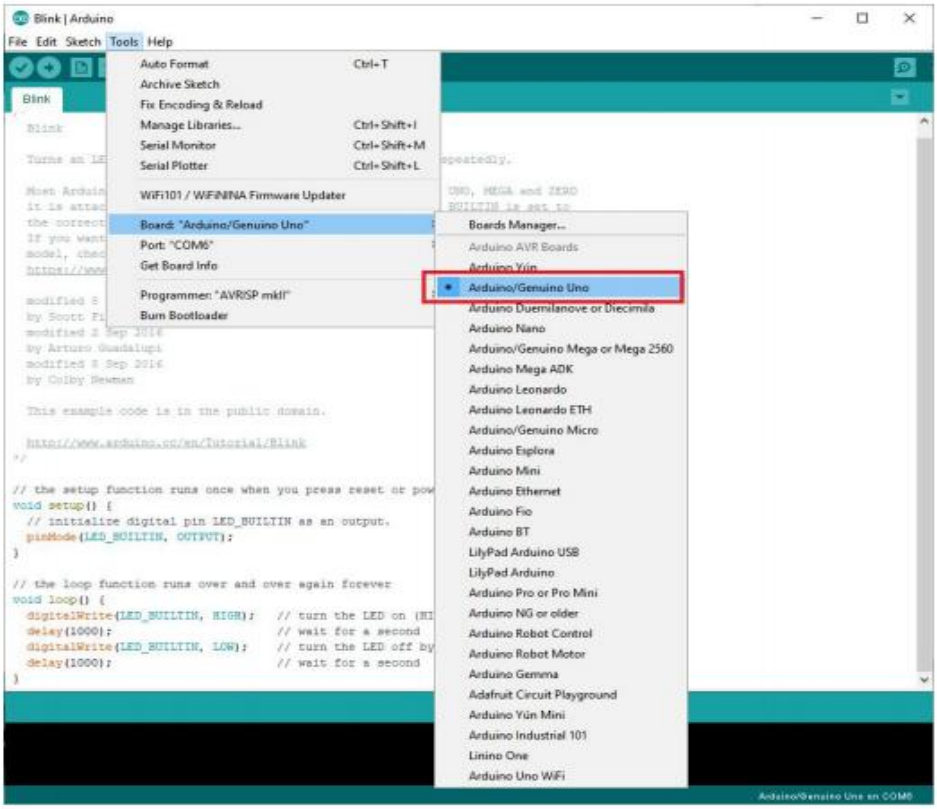
The Arduino IDE includes a large collection of example sketches that you can load up and use. This includes an example sketch for making the 'L' LED blink.

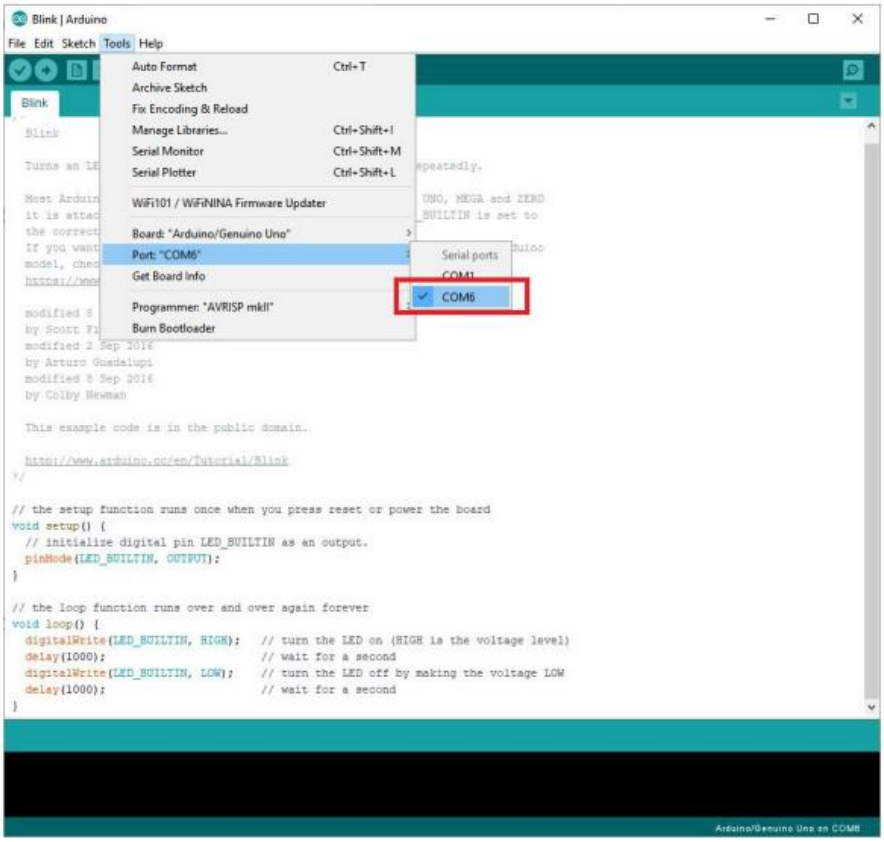
Load the 'Blink' sketch that you will find in the IDE's menu system under File > Examples > 01.Basics > Blink

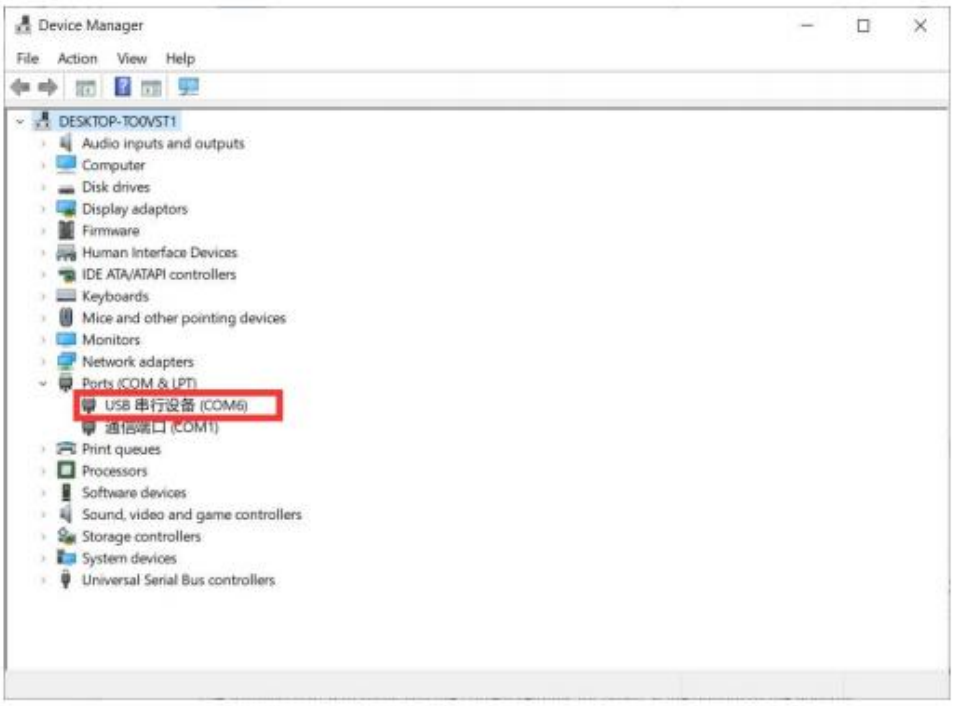


When the sketch window opens, enlarge it so that you can see the entire sketch in the window.

Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.


Blink | Arduino
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi

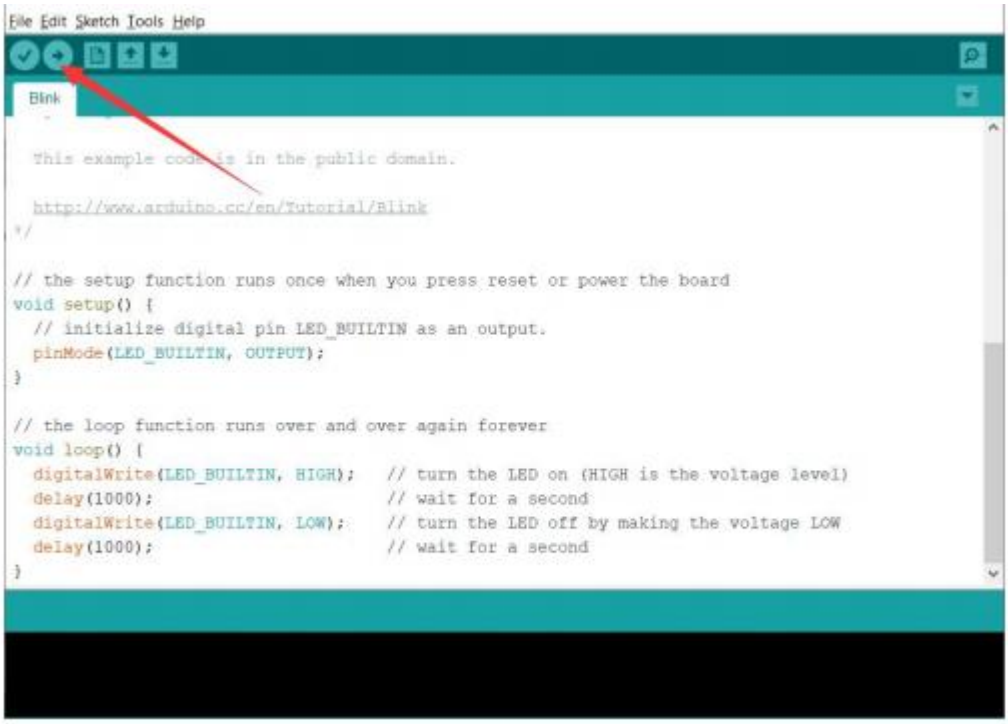

Blink | Arduino
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM6
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}



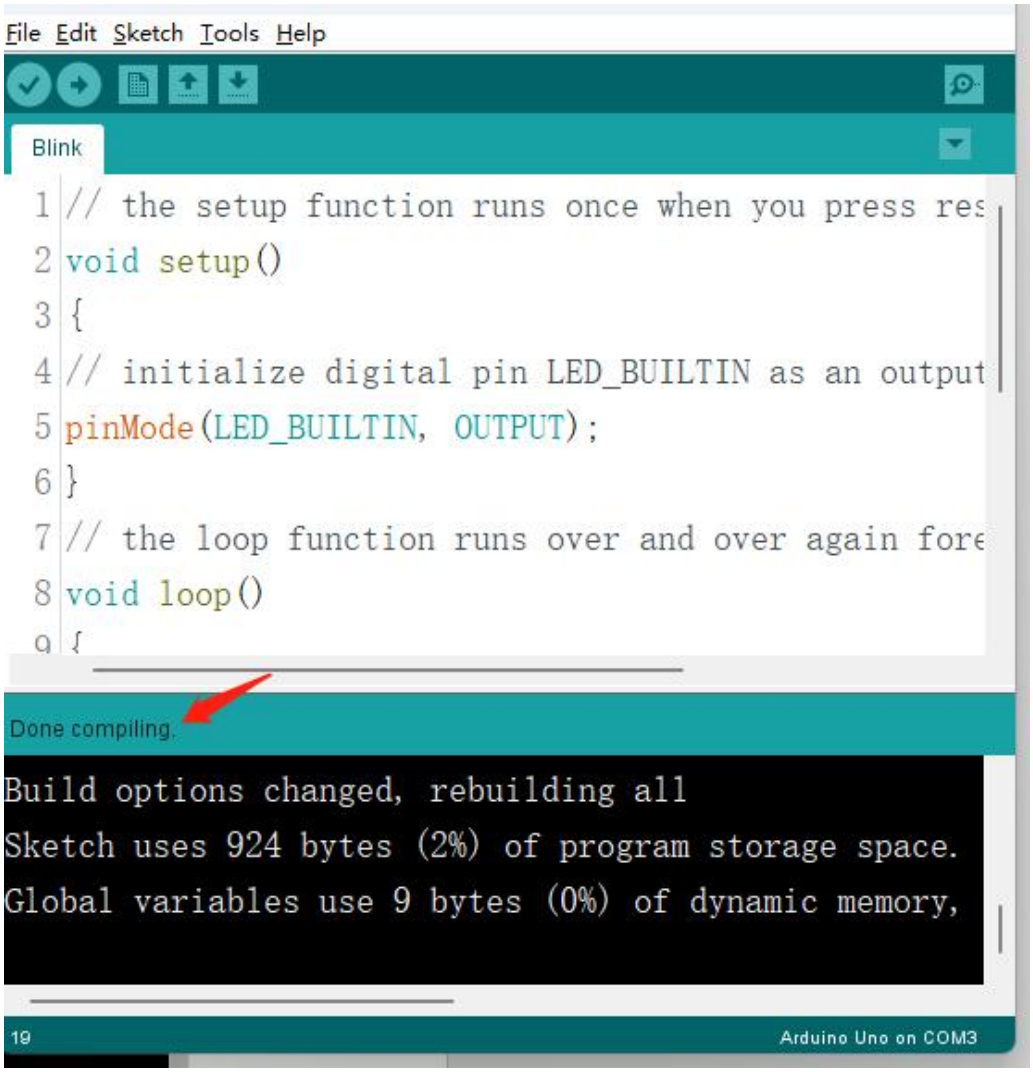
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using UNO, then you will have to choose Arduino UNO as the Board Type, other choices can be made in the same manner. And the Serial Port displayed for everyone is different, despite COM 6 chosen here, it could be COM3 or COM4 on your computer. A right COM port is supposed to be COMX (arduino XXX), which is by the certification criteria.

The Arduino IDE will show you the current settings for board at the bottom of the window.

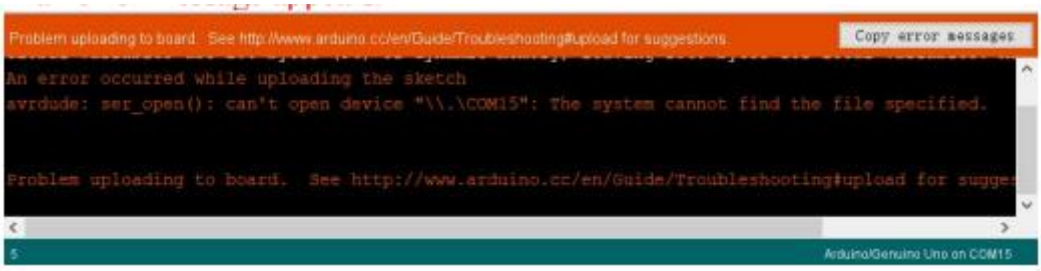
Click on the 'Upload' button. The second button from the left on the toolbar.



When the status bar prompts "Done uploading", it means the code upload is successful



If an error message appears.

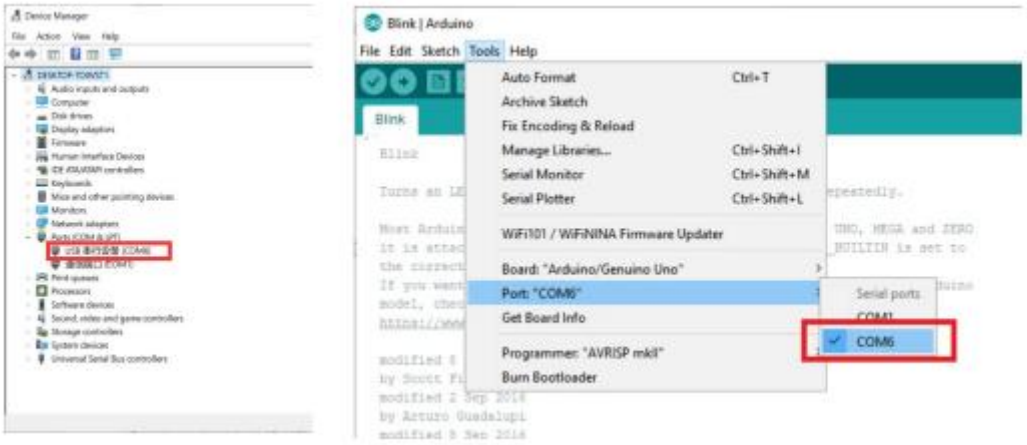


There can be several reasons:

1. The arduino uno driver software is not installed successfully, please refer to the course for the installation steps: "How to Install Arduino

Driver".

2. The communication serial port selection of arduino uno is wrong; you can check the communication port COMx of your arduino uno in the computer in the device manager.



3. If your Arduino uno is connected to a Bluetooth module, it will occupy the communication serial port. You need to remove the Bluetooth module connection before uploading the code.

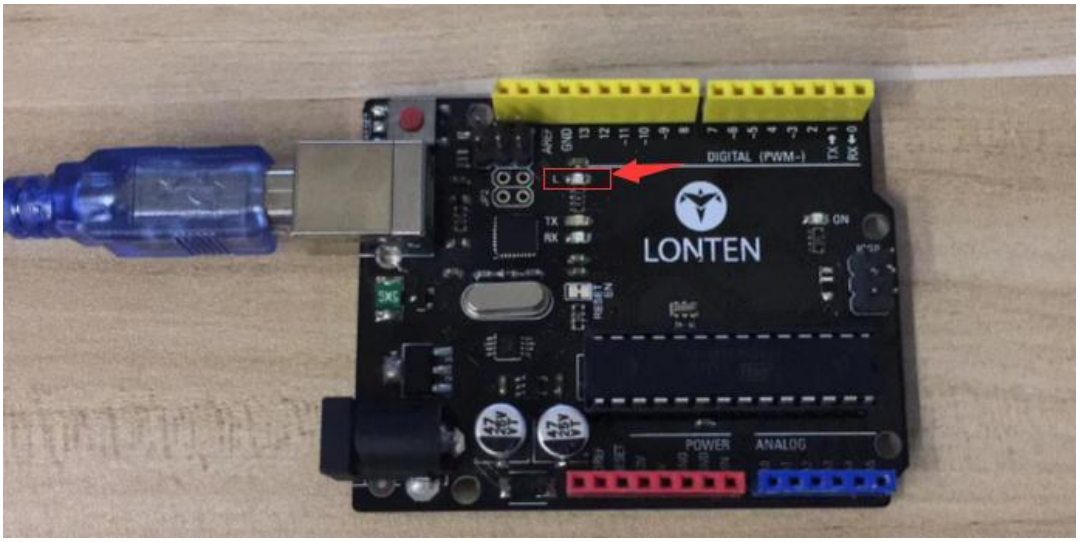
4. The USB data cable is not firmly connected.

Check if there are any of the above problems. After correcting, follow the previous steps to re-operate.

Sample Program

```
// the setup function runs once when you press reset or power the board
void setup()
{
```

```
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop()
{
digitalWrite(LED_BUILTIN, HIGH);
// turn the LED on (HIGH is the voltage level)
delay(1000);
// wait for a second
digitalWrite(LED_BUILTIN, LOW);
// turn the LED off by making the voltage LOW
delay(1000);
// wait for a second
}
```



After the code is successfully uploaded, the "L" character LED will flash once per second. So far, you have completed the testing process of your first program.

Lesson 1 LED

Overview

In this lesson, you will learn how to change the brightness of an LED by using different values of resistor.

Component Required:

(1) x LONTEN Uno R3 Board

(1) x 5mm red LED

(1) x 220 ohm resistor

(1) x 1k ohm resistor

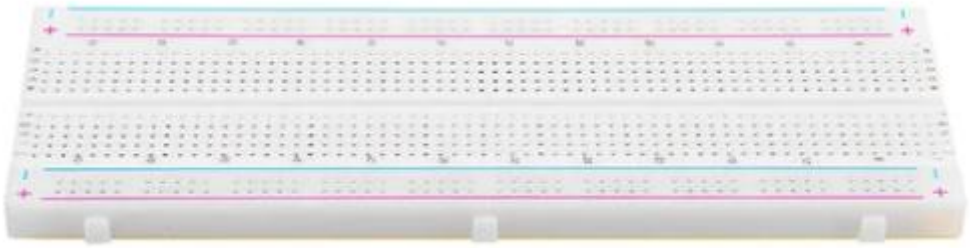
(1) x 10k ohm resistor

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

Component Introduction

BREADBOARD MB-102:

A breadboard enables you to prototype circuits quickly, without having to solder the connections. Below is an example.



Breadboards come in various sizes and configurations. The simplest kind is just a grid of holes in a plastic block. Inside are strips of metal that provide electrical connection between holes in the shorter rows. Pushing the legs of two different components into the same row joins them together electrically. A deep channel running down the middle indicates that there is a break in connections there, meaning, you can push a chip in with the legs at either side of the channel without connecting them together. Some breadboards have two strips of holes running along the long edges of the board that are separated from the main grid. These have strips running down the length of the board inside and provide a way to connect a common voltage. They are usually in pairs for +5 volts and

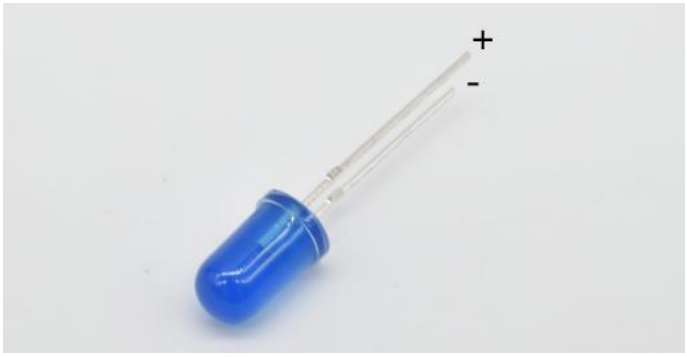
ground. These strips are referred to as rails and they enable you to connect power to many components or points in the board.

While breadboards are great for prototyping, they have some limitations. Because the connections are push-fit and temporary, they are not as reliable as soldered connections. If you are having intermittent problems with a circuit, it could be due to a poor connection on a breadboard.

LED:

LEDs make great indicator lights. They use very little electricity and they pretty much last forever.

In this Lesson, you will use perhaps the most common of all LEDs: a 5mm red LED. 5mm refers to the diameter of the LED. Other common sizes are 3mm and 10mm. You cannot directly connect an LED to a battery or voltage source because 1) the LED has a positive and a negative lead and will not light if placed the wrong way and 2) an LED must be used with a resistor to limit or 'choke' the amount of current flowing through it; otherwise, it will burn out!



If you do not use a resistor with an LED, then it may well be destroyed. almost immediately, as too much current will flow through, heating it and destroying the ‘junction’ where the light is produced.

There are two ways to tell which is the positive lead of the LED and which the negative.

Firstly, the positive lead is longer.

Secondly, where the negative lead enters the body of the LED, there is a flat edge to the case of the LED.

If you happen to have an LED that has a flat side next to the longer lead, you should assume that the longer lead is positive.

RESISTORS:

As the name suggests, resistors resist the flow of electricity. The higher the value of the resistor, the more it resists and the less electrical current. will flow through it. We are going to use this to control how much electricity flows through the LED and therefore, how brightly it shines.

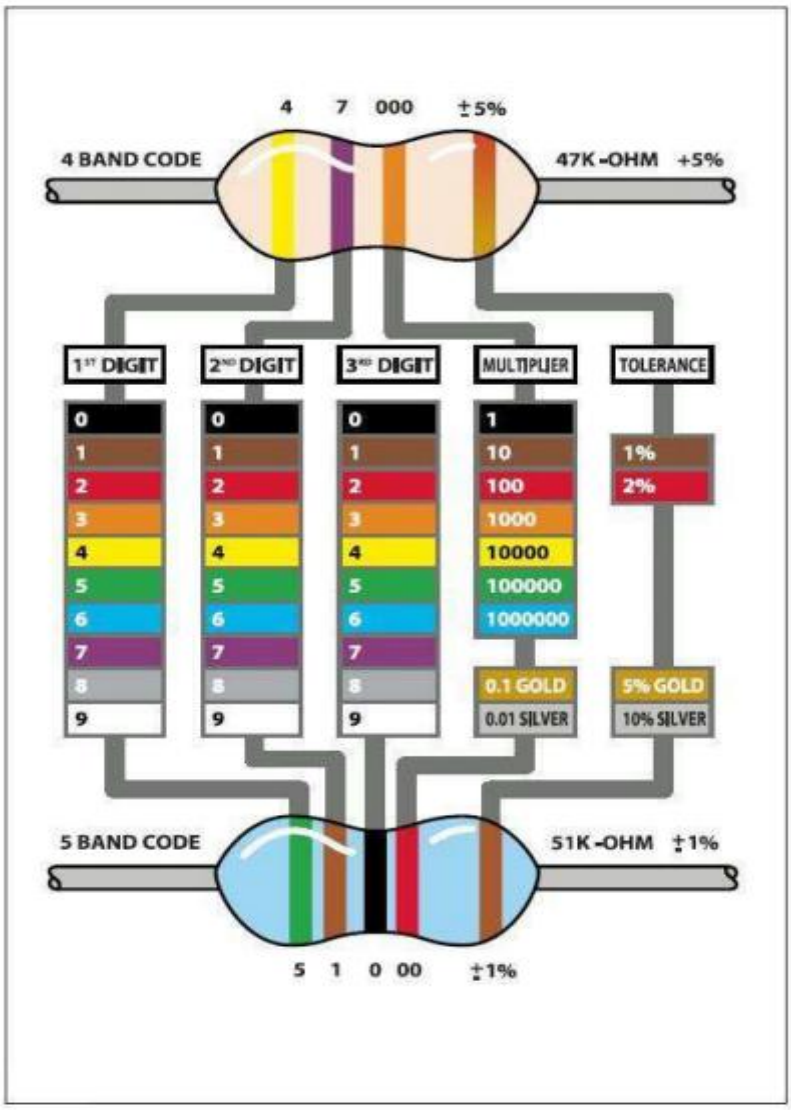
But first, more about resistors…

The unit of resistance is called the Ohm, which is usually shortened to Ω the Greek letter Omega. Because an Ohm is a low value of resistance (it doesn’t resist much at all), we also denote the values of resistors in Kω (1,000 Ω) and MΩ (1,000,000 Ω).

These are called kilo-ohms and mega-ohms.

In this learning kit, we are going to use three different values of resistor: 220Ω, 1Kω and 10Kω. These resistors all look the same, except that they have different colored stripes on them. These stripes tell you the value of the resistor.

The resistor color code has three colored stripes and then a gold stripe at one end.



Unlike LEDs, resistors do not have a positive and negative lead. They can be connected either way around.

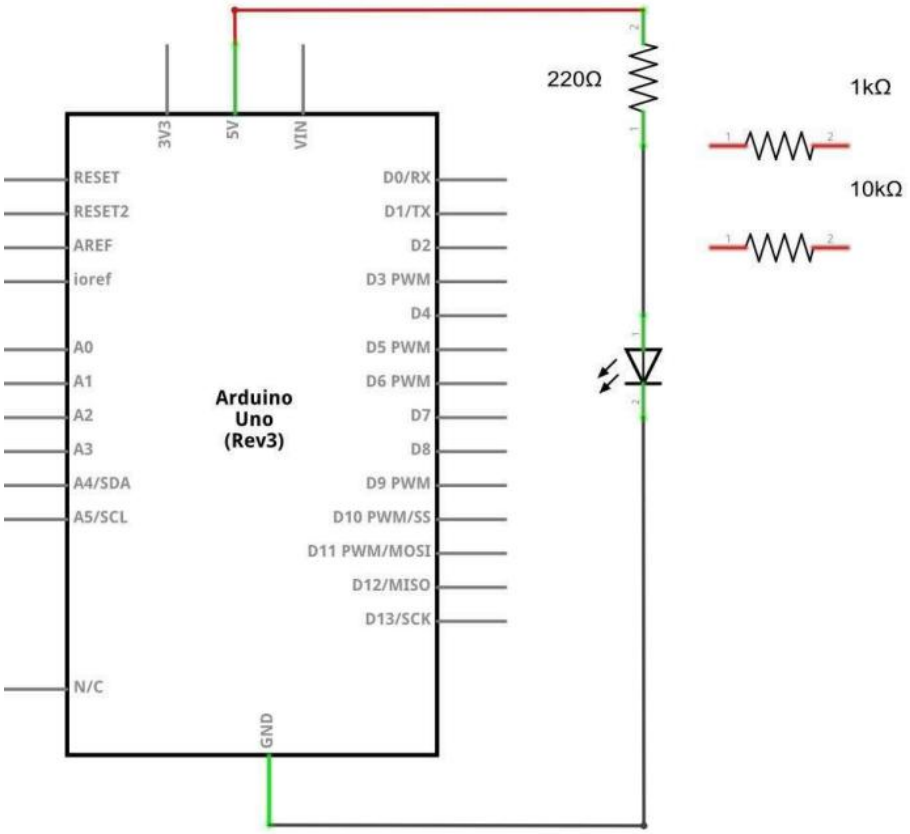
If you find this approach method too complicated, you can read the color ring flag on our resistors directly to determine its resistance value. Or you may use a digital multimeter instead.

We follow below diagram from the experimental schematic link. Here we

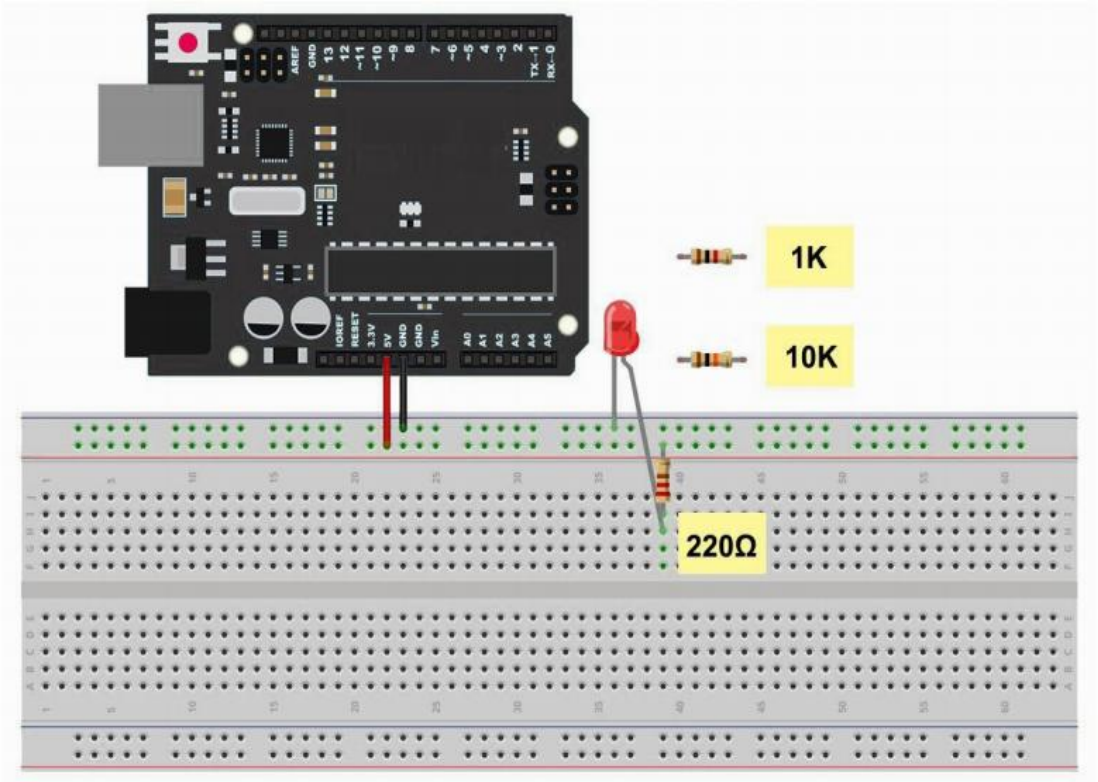
use digital pin 10. We connect LED to a 220-ohm resistor to avoid high current damaging the LED.

Connection

Schematic



Circuit Connection



The UNO is a convenient source of 5 volts, which we will use to provide power to the LED and the resistor. You do not need to do anything with your UNO, except to plug it into a USB cable.

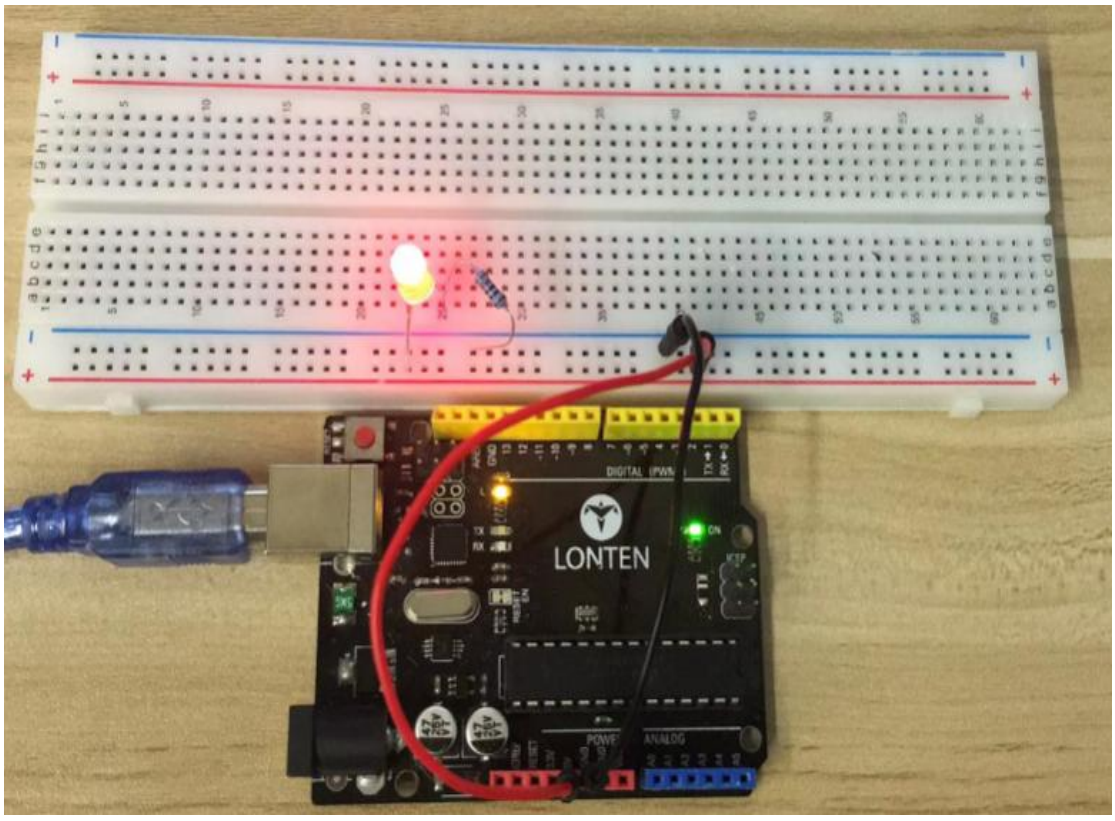
With the 220 Ω resistor in place, the LED should be quite bright. If you swap out the 220 resistor for the 1k Ω resistor, then the LED will appear a little dimmer. Finally, with the 10 k Ω resistor in place, the LED will be just about visible. Pull the red jumper lead out of the breadboard and touch it into the hole and remove it, so that it acts like a switch. You should just be able to notice the difference.

At the moment, you have 5V going to one leg of the resistor, the other leg of the resistor going to the positive side of the LED and the other side of

the LED going to GND. However, if we moved the resistor so that it came after the LED, as shown below, the LED will still light.

You will probably want to put the 220 Ω resistor back in place. It does not matter which side of the LED we put the resistor, as long as it is there somewhere.

Example picture



LROBRUYA

Lesson 2 Digital Inputs

Overview

In this lesson, you will learn to use push buttons with digital inputs to turn an LED on and off.

Pressing the button will turn the LED on; pressing the other button will turn the LED off.

Component Required:

(1) x LONTEN Uno R3

(1) x 830 Tie-pointsBreadboard

(1) x 5mm red LED

(1) x 220 ohm resistor

(2) x push switches

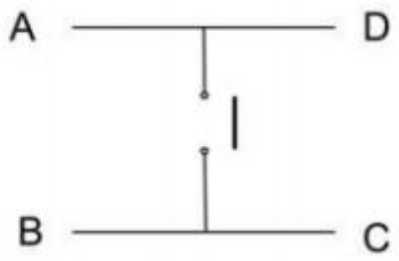
(7) x M-M wires (Male to Male jumperwires)

Component Introduction

PUSH SWITCHES:

Switches are really simple components. When you press a button or flip a lever,they connect two contacts together so that electricity can flow through them.

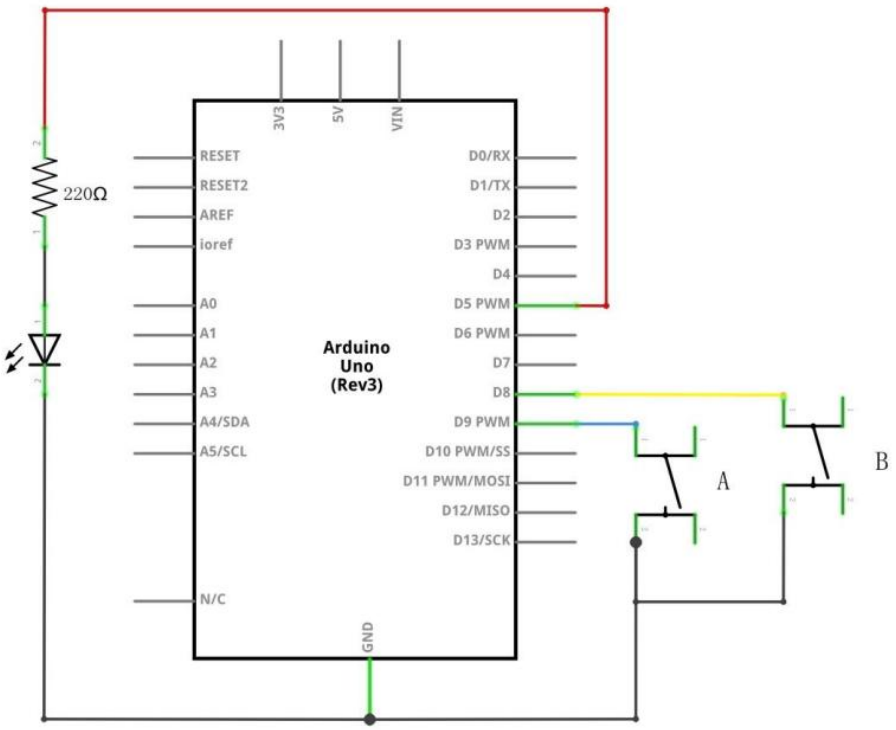
The little tactile switches that are used in this lesson have four connections, which can be a little confusing.



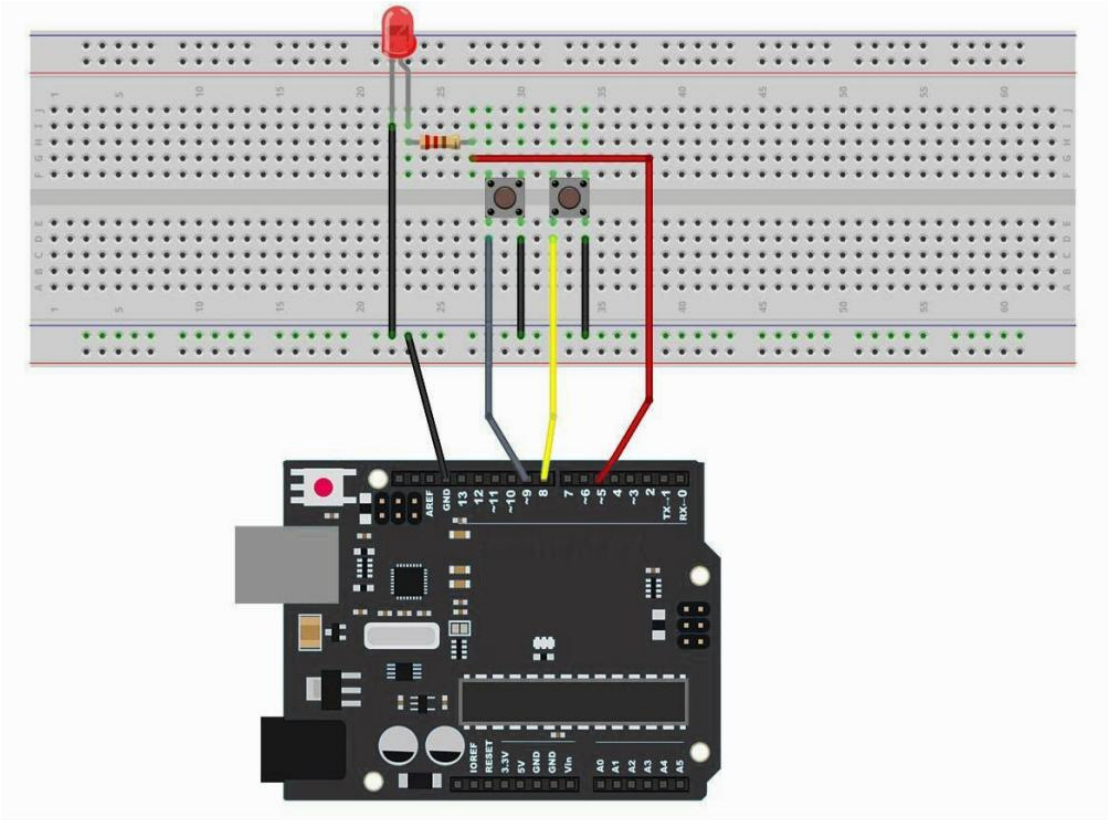
Actually, there are only really two electrical connections. Inside the switch package,pins B and C are connected together, as are A and D.

Connection

Schematic



Circuit Connection



Although the bodies of the switches are square, the pins protrude from opposite sides of the switch. This means that the pins will only be far enough apart when they are placed correctly on the breadboard.

Remember that the LED has to have the shorter negative lead to the left.

Code

After wiring,please open program in the code folder- Lesson 2 Digital Inputs, and press UPLOAD to upload the program.

Load the sketch onto your UNO board. Pressing the left button will turn the LED on while pressing the right button will turn it off.

The first part of the sketch defines three variables for the three pins that are to be used. The 'ledPin' is the output pin and 'buttonApin' will refer to the switch nearer the top of the breadboard and 'buttonBpin' to the other switch.

The 'setup' function defines the ledPin as being an OUTPUT as normal, but now we have the two inputs to deal with. In this case, we use the set the pinMode to be 'INPUT_PULLUP' like this:

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

The pin mode of INPUT_PULLUP means that the pin is to be used as an input, but that if nothing else is connected to the input, it should be 'pulled up' to HIGH. In other words, the default value for the input is HIGH, unless it is pulled LOW by the action of pressing the button.

This is why the switches are connected to GND. When a switch is pressed, it connects the input pin to GND, so that it is no longer HIGH.

Since the input is normally HIGH and only goes LOW when the button is pressed, the logic is a little upside down. We will handle this in the 'loop' function.

```
void loop()
{
```

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

In the 'loop' function there are two 'if' statements. One for each button. Each does an 'digitalRead' on the appropriate input.

Remember that if the button is pressed, the corresponding input will be LOW, if button A is low, then a 'digitalWrite' on the ledPin turns it on.

Similarly, if button B is pressed, a LOW is written to the ledPin.

Lesson 3 Active Buzzer

Overview

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

Component Required:

(1) x LONTEN Uno R3

(1) x Active buzzer

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

Component Introduction

BUZZER:

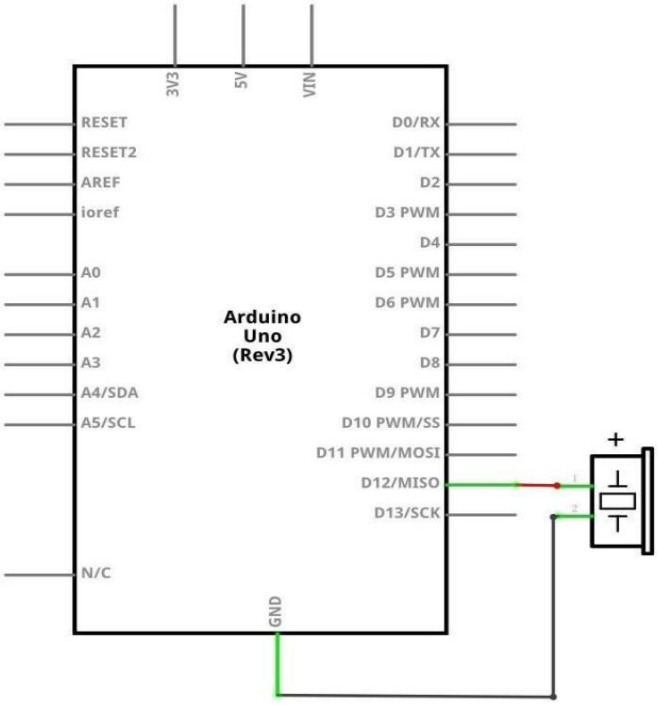
Active buzzer is widely used on computer, printer, alarm, electronic toy, telephone, timer etc.. It has an inner vibration source. Simply connect it with 5V power supply, it can buzz continuously.

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.

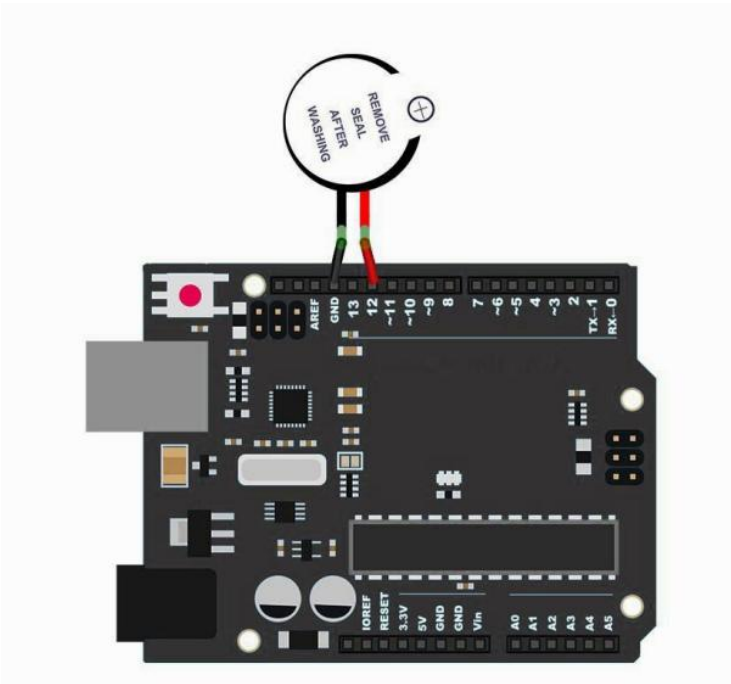
The difference between the two is that an active buzzer has a built-in oscillating source, so it will generate a sound when electrified. A passive buzzer does not have such a source so it will not tweet if DC signals are used; instead, you need to use square waves whose frequency is between 2K and 5K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

Connection

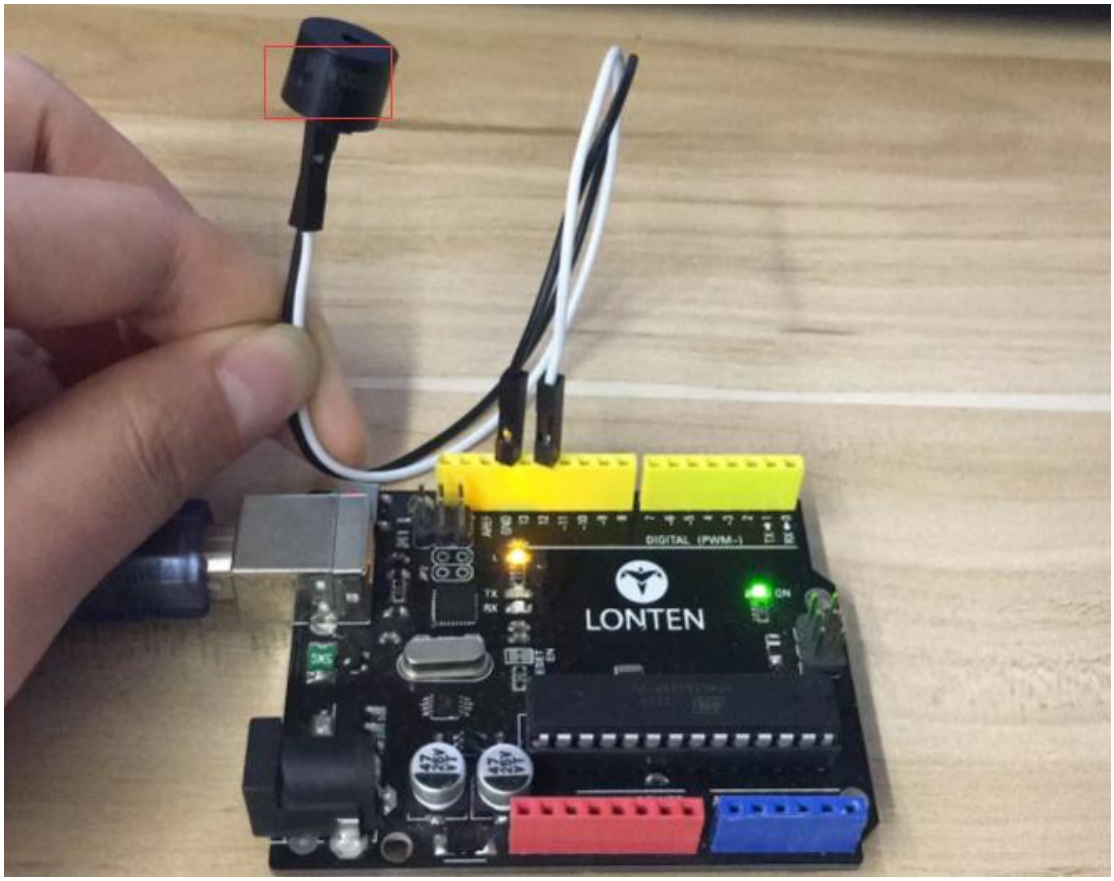
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 3 Making Sounds and click UPLOAD to upload the program.

Lesson 4 Passive Buzzer

Overview

In this lesson, you will learn how to use a passive buzzer.

The purpose of the experiment is to generate eight different sounds, each sound lasting 0.5 seconds: from Alto Do (523Hz), Re (587Hz), Mi

(659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz) to Treble Do (1047Hz).

Component Required

(1) x LONTEN Uno R3

(1) x Passive buzzer

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

Component Introduction

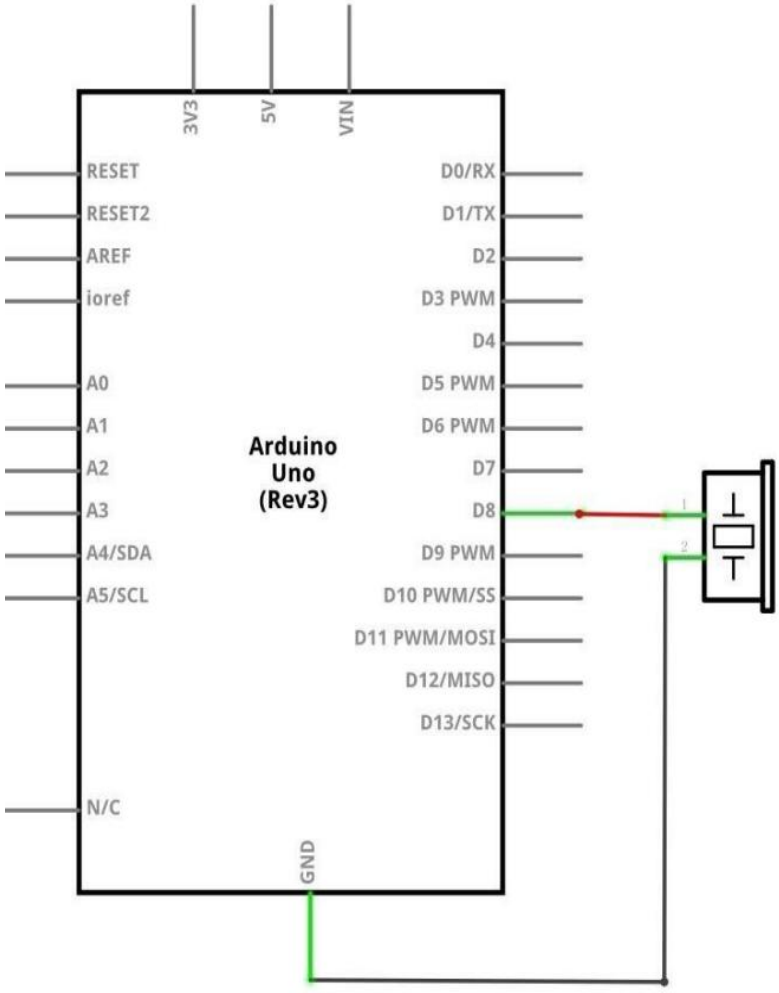
Passive Buzzer:

The working principle of passive buzzer is using PWM generating audio to make the air to vibrate. Appropriately changed as long as the vibration frequency, it can generate different sounds. For example, sending a pulse of 523Hz, it can generate Alto Do, pulse of 587Hz, it can generate midrange Re, pulse of 659Hz, it can produce midrange Mi. By the buzzer, you can play a song.

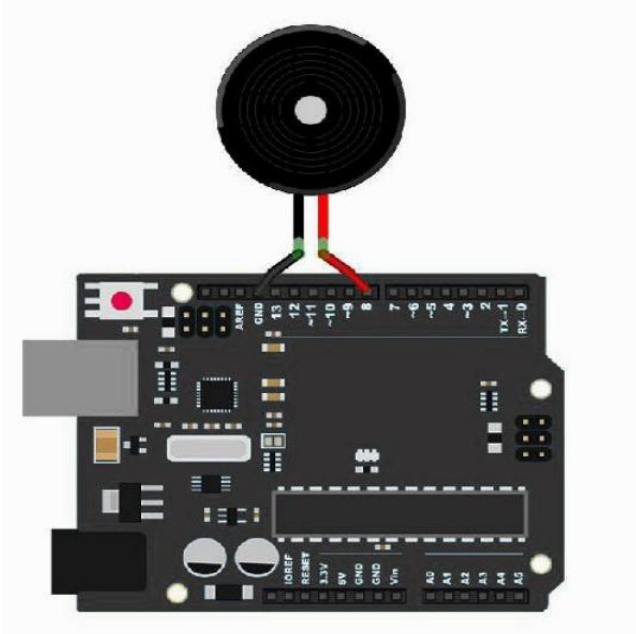
We should be careful not to use the UNO R3 board analog Write () function to generate a pulse to the buzzer, because the pulse output of analog Write () is fixed (500Hz).

Connection

Schematic

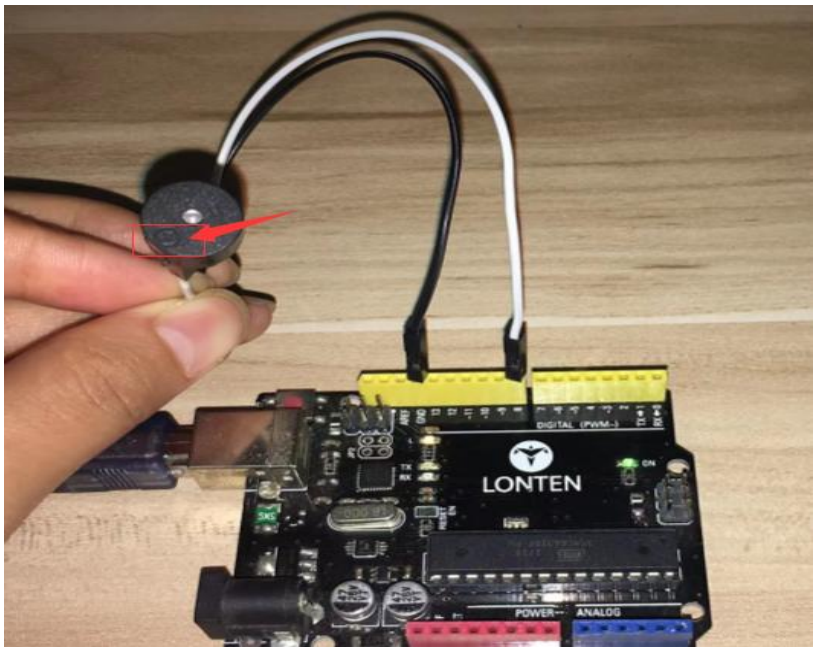


Circuit Connection



Wiring the buzzer connected to the UNO R3 board, the red (positive) to the pin8, black wire (negative) to the GND.

Example picture



Code

After wiring, please open the program in the code folder- Lesson 4 Passive Buzzer and click UPLOAD to upload the program.

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

Lesson 5 Tilt Ball Switch

Overview

In this lesson, you will learn how to use a tilt ball switch in order to detect small angle of inclination.

Component Required

(1) x LONTEN Uno R3

(1) x Tilt Ball switch

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

Component Introduction

Tilt sensor:

Tilt sensors (tilt ball switch) allow you to detect orientation or inclination. They are small, inexpensive, low-power and easy-to-use. If used properly, they will not wear out. Their simplicity makes them popular for toys, gadgets and appliances. Sometimes, they are referred to as "mercury switches", "tilt switches" or "rolling ball sensors" for obvious reasons.

They are usually made up of a cavity of some sort (cylindrical is popular, although not always) with a conductive free mass inside, such as a blob of mercury or rolling ball.

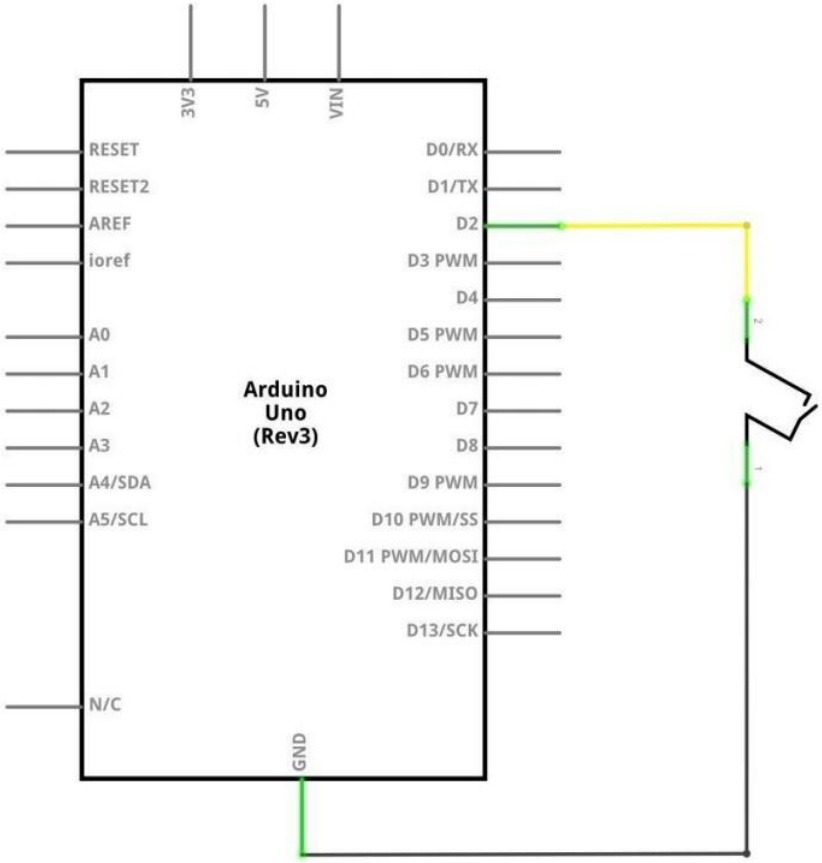
One end of the cavity has two conductive elements (poles). When the sensor is oriented so that that end is downwards, the mass rolls onto the poles and shorts them, acting as a switch throw.

While not as precise or flexible as a full accelerometer, tilt switches can detect motion or orientation. Another benefit is that the big ones can switch power on their own.

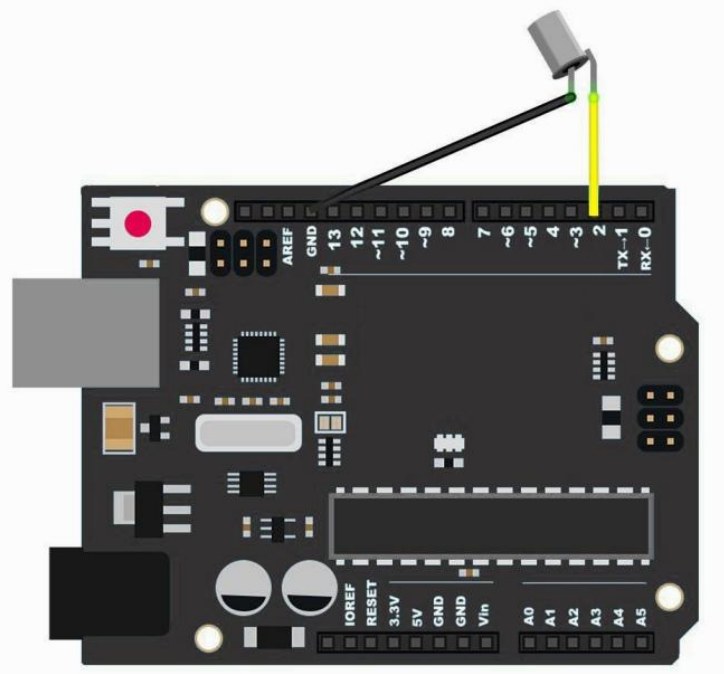
Accelerometers, on the other hand, output digital or analog voltage that must then be analyzed using extra circuitry.

Connection

Schematic



Circuit Connection



Code

After wiring, please open the program in the code folder- Lesson 5 Ball Switch and click UPLOAD to upload the program.

Lesson 6 Analog Value Reading

Overview

In this experiment, we will begin the learning of analog I/O interfaces. On an Arduino, there are 6 analog interfaces numbered from A0 to A5. These 6 interfaces can also be used as digital ones numbered as D14-D19. After a brief introduction, let's begin our project. Potentiometer used here is a typical output component of analog value that is familiar to us.

Component Required

(1) x LONTEN Uno R3

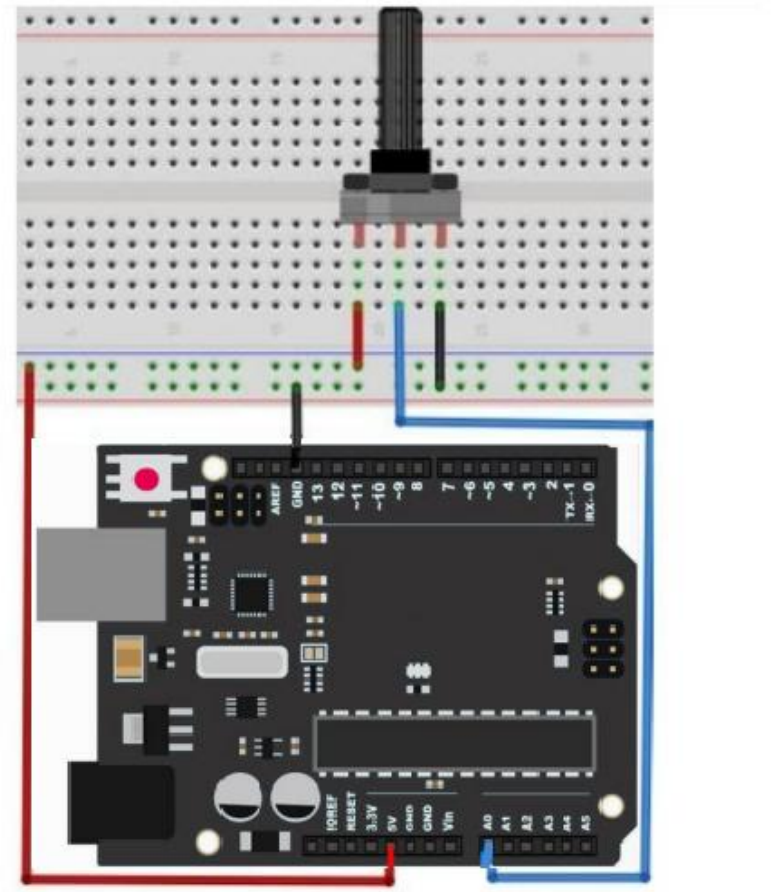
(1) x Potentiometer

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

Circuit Connection

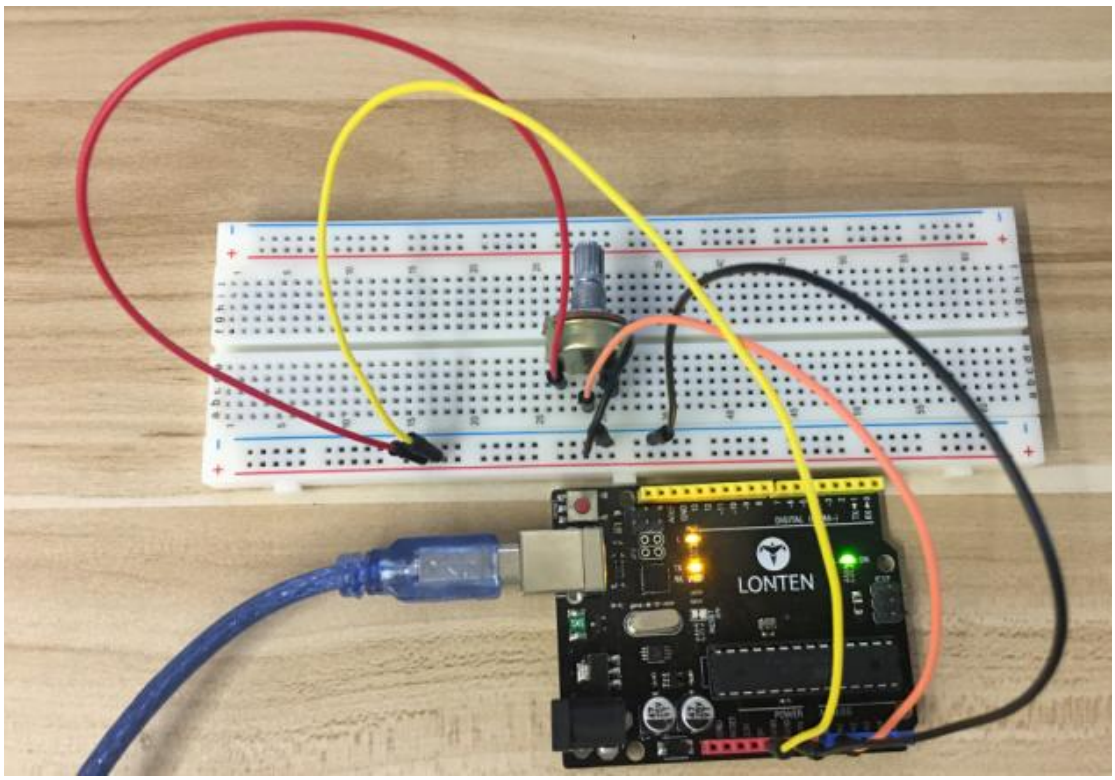
In this experiment, we will convert the resistance value of the potentiometer to analog ones and display it on the screen. This is an application we need to master well for our future experiments.

Connection circuit as below:



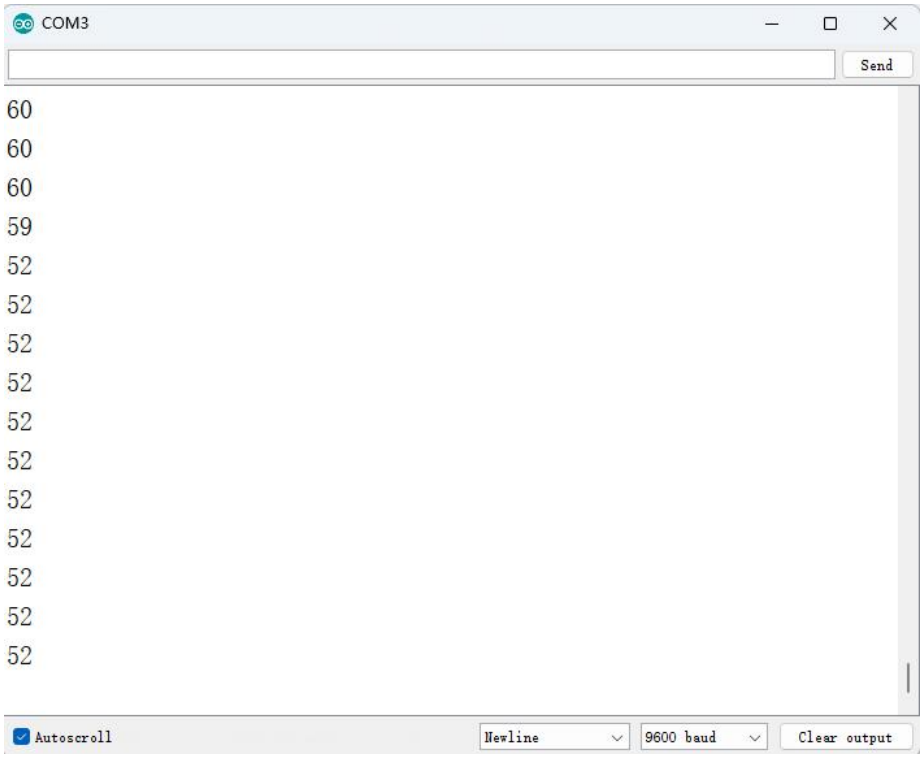
The analog interface we use here is interface A0.

Example picture



Code

The Sample Program uses the built-in LED connected to pin 13. Each time the device reads a value, the LED blinks. Below is the analog value it reads.



When you rotate the potentiometer knob, you can see the displayed value changes. The reading of analog value is a very common function since most sensors output analog value. After calculation, we can have the corresponding value we need.

Lesson 7 Servo

Overview

Servo is a type of geared motor that can only rotate 180 degrees. It is controlled by sending electrical pulses from your UNO R3 board. These pulses tell the servo what position it should move to. The Servo has three wires, of which the brown one is the ground wire and should be

connected to the GND port of UNO, the red one is the power wire and should be connected to the 5v port, and the orange one is the signal wire and should be connected to the Dig #9 port.

Component Required

(1) x LONTEN Uno R3

(1) x Servo (SG90)

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

Component Introduction

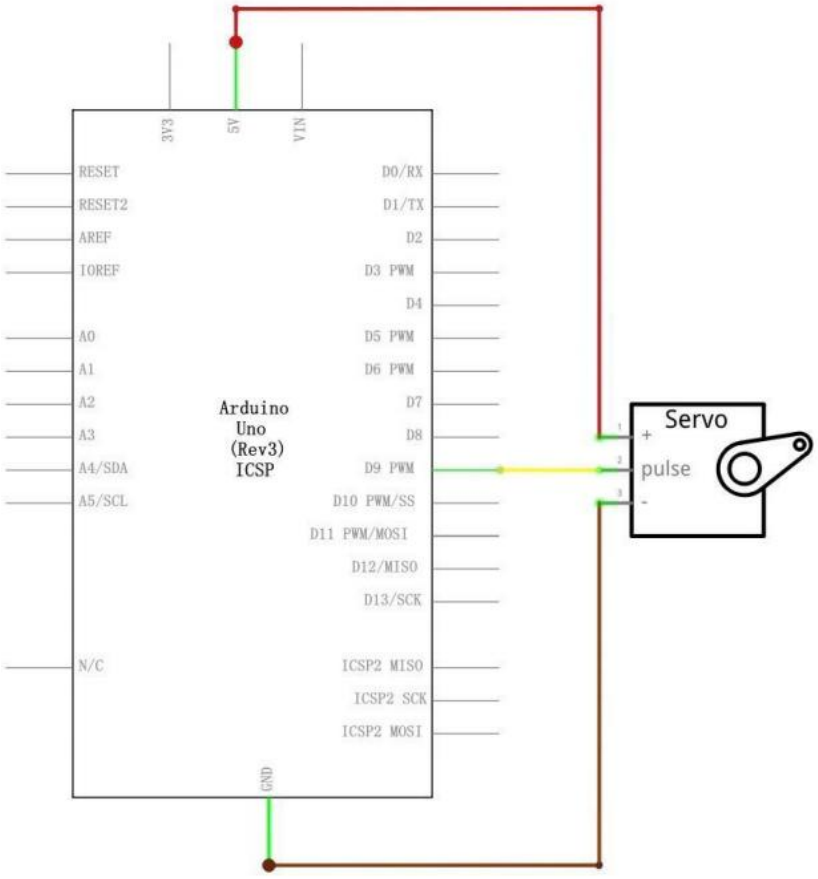
SG90

- Universal for JR and FP connector
- Cable length : 25cm
- No load; Operating speed: 0.12 sec / 60 degree (4.8V), 0.10 sec / 60 degree (6.0V)
- Stall torque (4.8V): 1.6kg/cm
- Temperature : -30~60'C

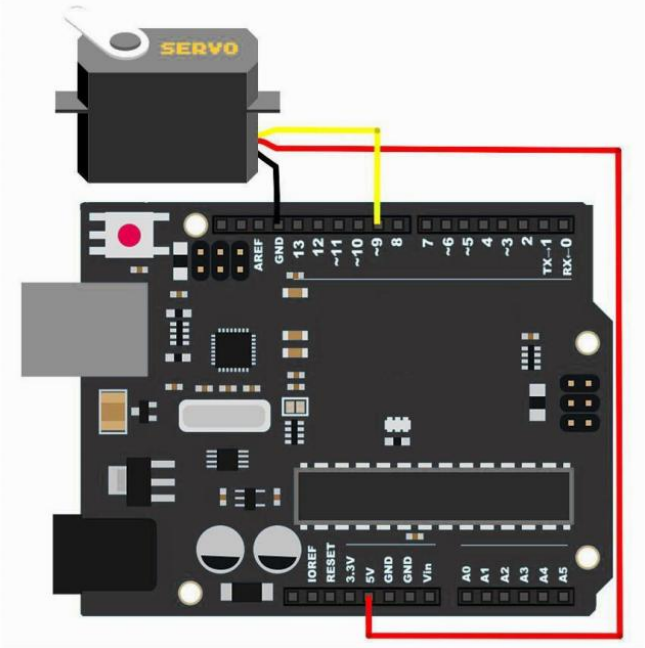
- Dead band width: 5us Working voltage: 3.5~6V
- Dimension : 1.26 in x 1.18 in x 0.47 in (3.2 cm x 3 cm x 1.2 cm)
- Weight : 4.73 oz (134 g)

Connection

Schematic

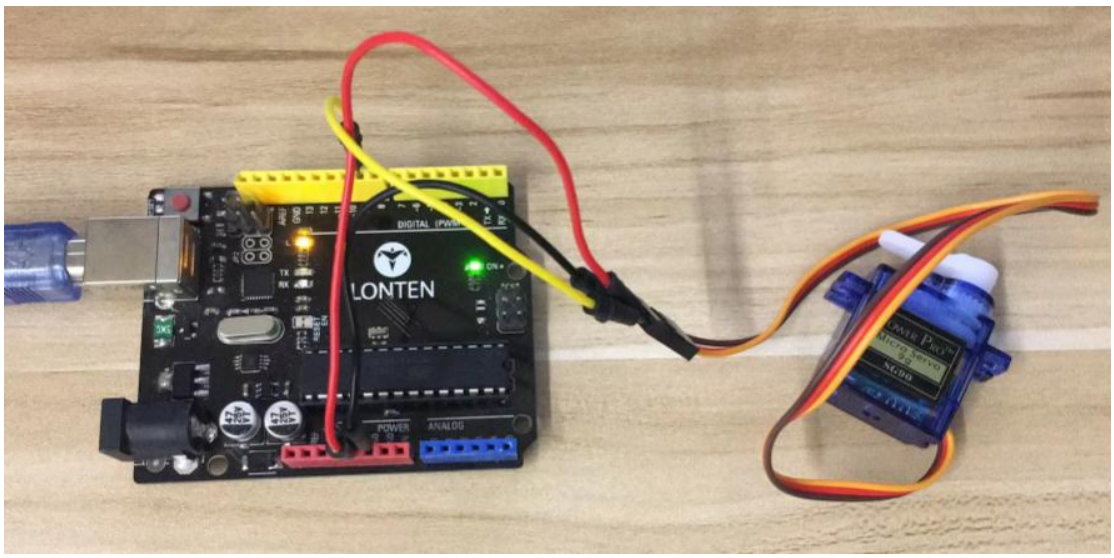


Circuit Connection



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.



Code

After wiring, please open the program in the code folder- Lesson 7 Servo and click UPLOAD to upload the program.

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.

Lesson 8 Analog Joystick Module

Overview

Analog joysticks are a great way to add some control in your projects. In this tutorial we will learn how to use the analog joystick module.

Component

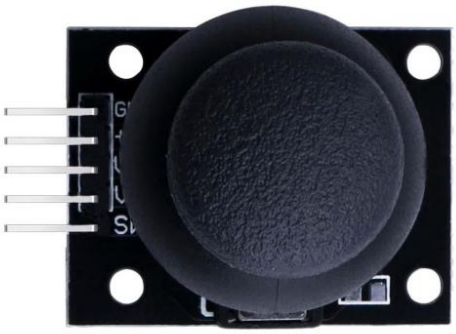
(1) x LONTEN Uno R3

(1) x Joystick module

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

Component Introduction

Joystick

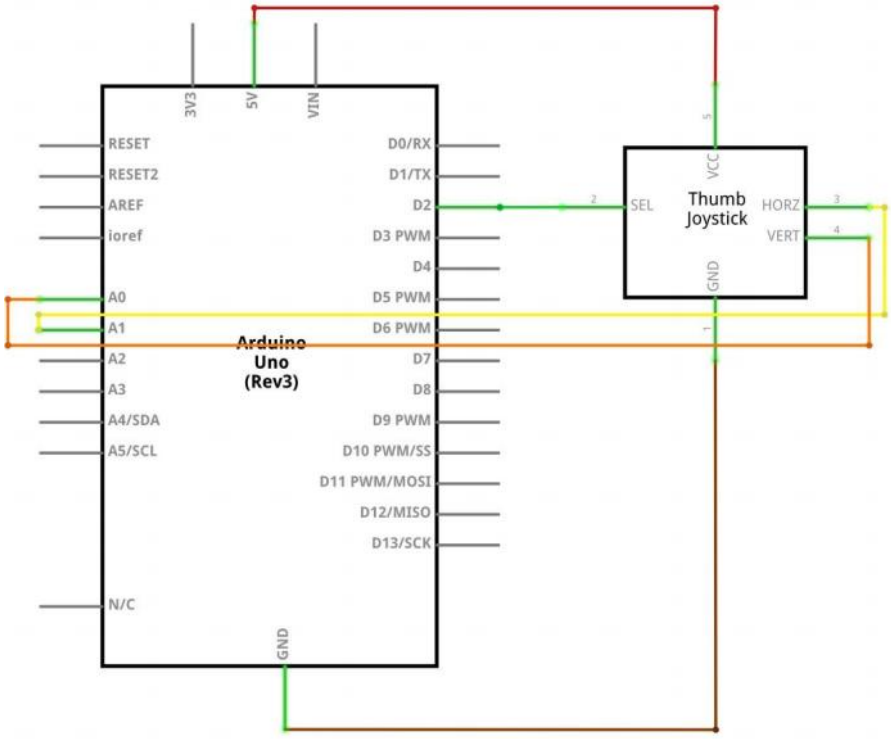


The module has 5 pins: VCC, Ground, X, Y, Key. Note that the labels on yours may be slightly different, depending on where you got the module from. The thumb stick is analog and should provide more accurate readings than simple ‘directional’ joysticks tact use some forms of buttons, or mechanical switches. Additionally, you can press the joystick down (rather hard on mine) to activate a ‘press to select’ push-button.

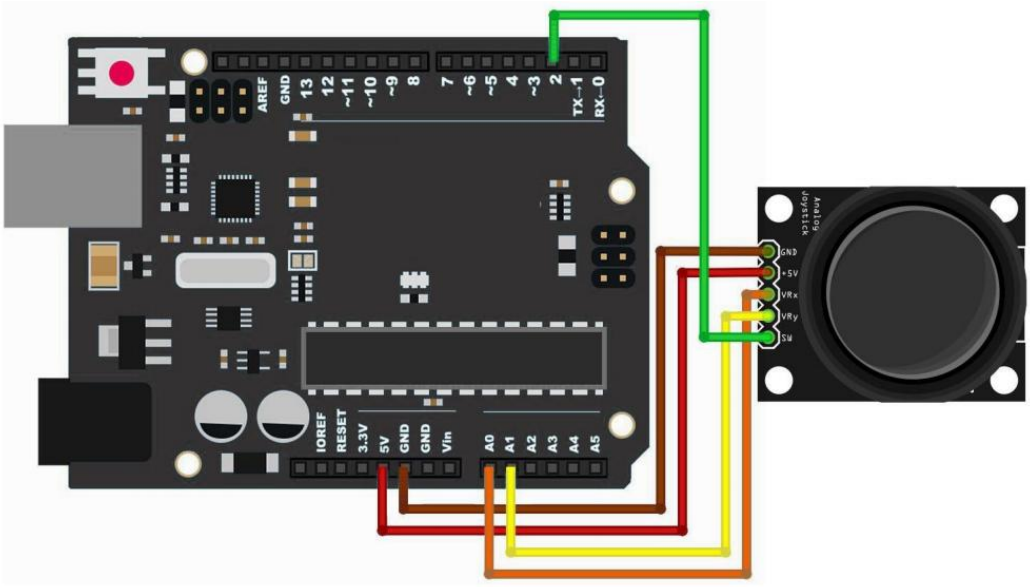
We have to use analog Arduino pins to read the data from the X/Y pins, and a digital pin to read the button. The Key pin is connected to ground, when the joystick is pressed down, and is floating otherwise. To get stable readings from the Key /Select pin, it needs to be connected to VCC via a pull-up resistor. The built in resistors on the Arduino digital pins can be used. For a tutorial on how to activate the pull-up resistors for Arduino pins, configured as inputs.

Connection

Schematic



Circuit Connection

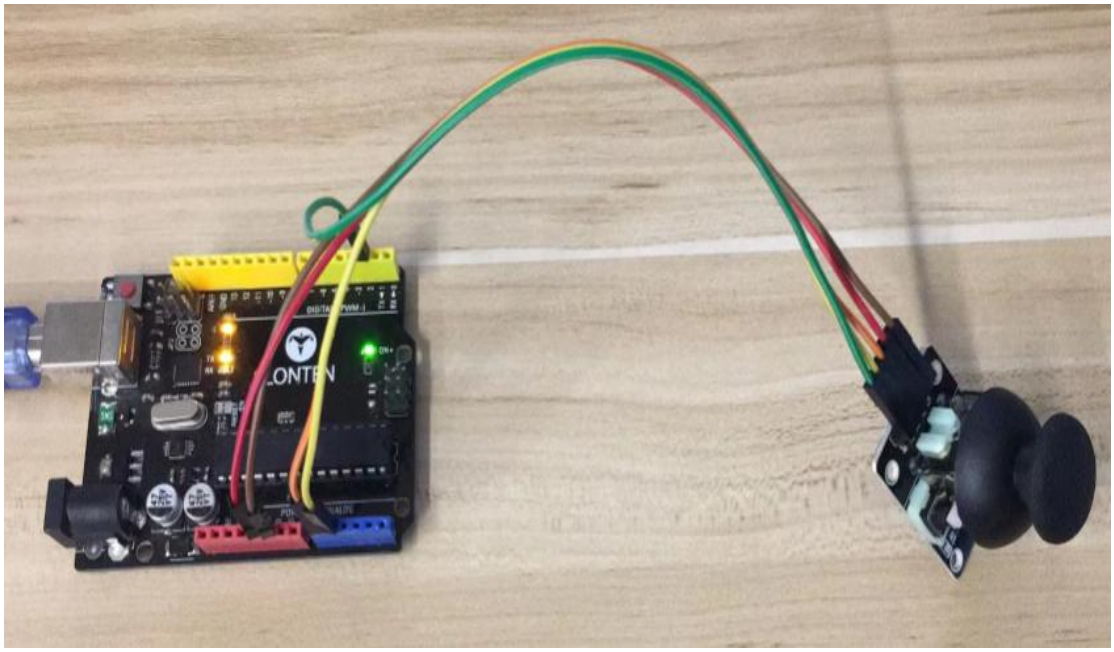


We need 5 connections to the joystick.

The connections are: Key, Y, X, Voltage and Ground.

“Y and X” are Analog and “Key” is Digital. If you don’t need the switch then you can use only 4 pins.

Example picture



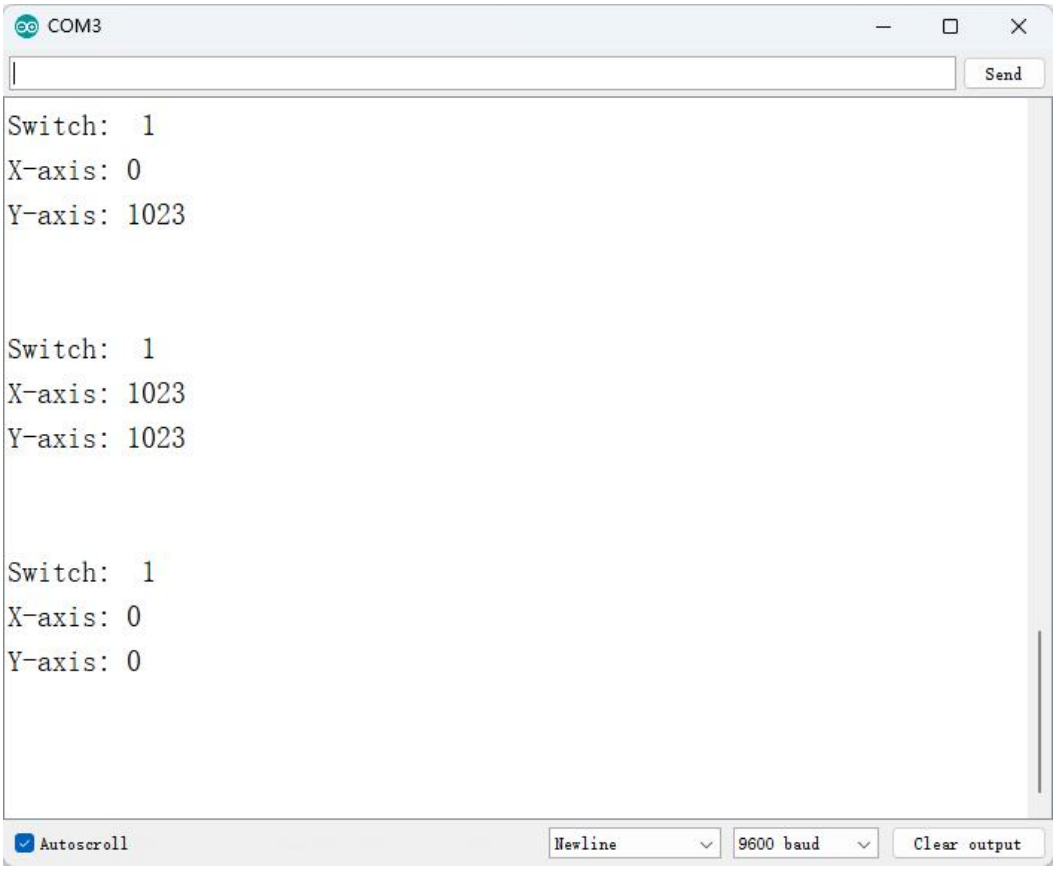
Code

After wiring, please open the program in the code folder- Lesson 8 Analog Joystick Module and click UPLOAD to upload the program.

Analog joysticks are basically potentiometers so they return analog values.

When the joystick is in the resting position or middle, it should return a value of about 512.

The range of values goes from 0 to 1024.



Lesson 9 Water Level Detection Sensor Module

Overview

In this lesson, you will learn how to use a water level detection sensor module. This module can perceive the depth of water and the core component is an amplifying circuit which is made up of a transistor and several pectinate PCB routings. When put into the water, these routings will present a resistor that can change along with the change of the water's depth. Then, the signal of water's depth is converted into the

electrical signal, and we can know the change of water's depth through the ADC function of UNO R3.

Component Required

(1) x LONTEN Uno R3

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

(1) x Water lever detection sensor module

Component Introduction

Water sensor:

A water sensor brick is designed for water detection, which can be widely used in sensing the rainfall, water level, even the liquate leakage. The brick is mainly composed of three parts: an electronic brick connector, a 1 MΩ resistor, and several lines of bare conducting wires. You can use it with the analog pins to detect the amount of water induced contact between the grounded and sensor traces. This item can judge the water level through with a series of exposed parallel wires stitch to measure the water droplet/water size. It can easily change the water size to analog signal, and output analog value can directly be used in the program

function, then to achieve the function of water level alarm. It has low power consumption, and high sensitivity.

Features:

1、Working voltage: 5V

2、Working Current: <20ma

3、Interface: Analog

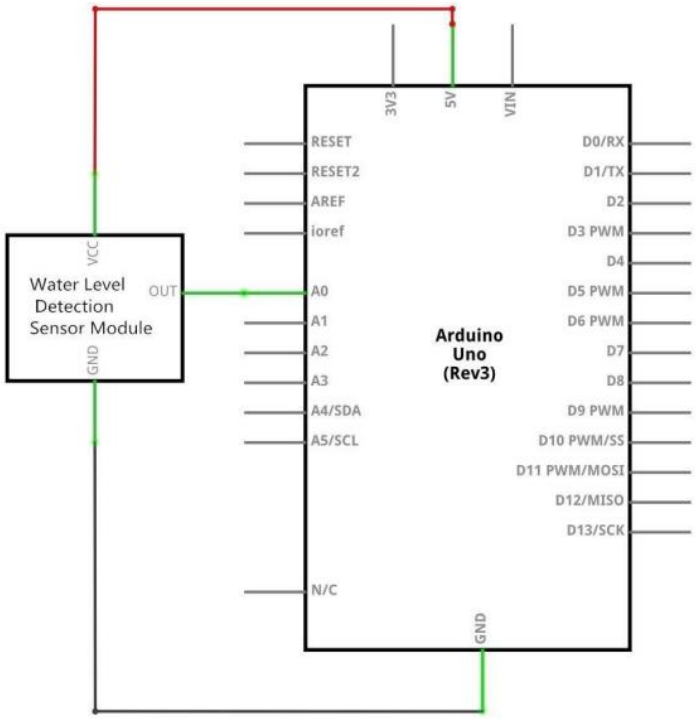
4、Width of detection: 40mm×16mm

5、Working Temperature: 10℃~30℃

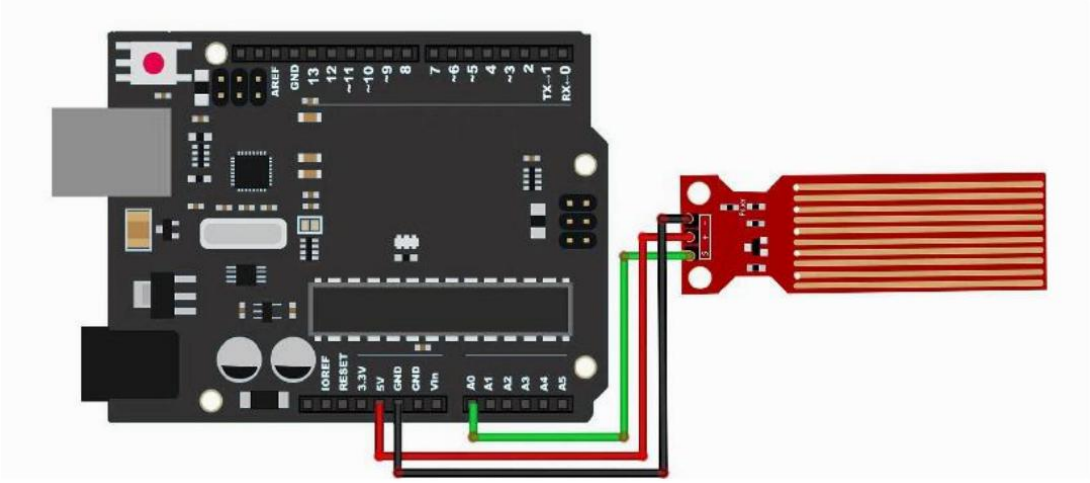
6、Output voltage signal: 0~4.2V

Connection

Schematic

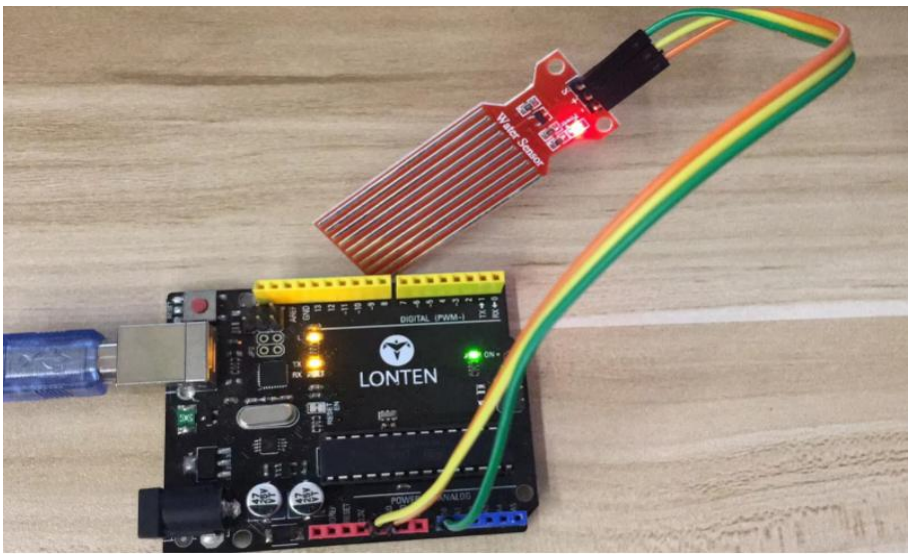


Circuit Connection



Wiring tips: Power supply (+) is connected to 5V of UNO R3 board, ground electrode (-) is connected to GND. Signal output (S) is connected to the ports (A0-A5) which have function of inputting analog signal in UNO R3 board, random one is OK, but it should define the same demo code as the routine.

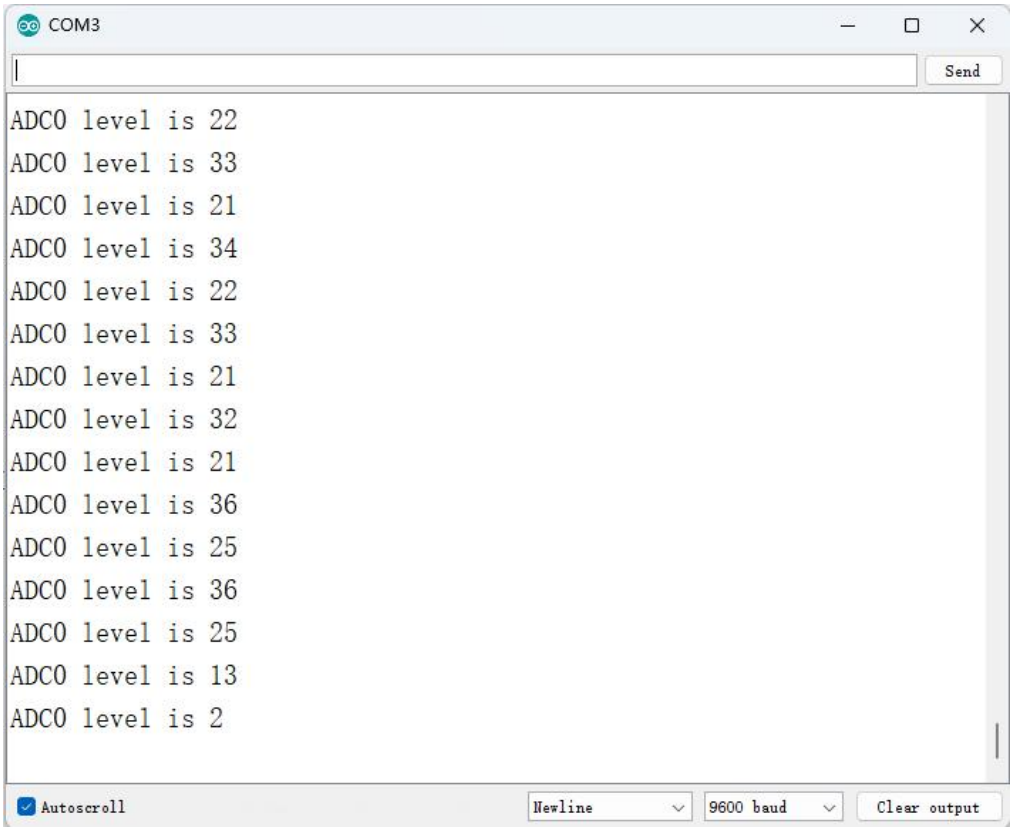
Example picture



Code

After wiring, please open the program in the code folder- Lesson 9 Water Level Detection Sensor Module and click UPLOAD to upload the program.

Open the monitor then you can see the data as below:



Lesson 10 IR Receiver Module

Overview

Using an IR Remote is a great way to have wireless control of your project. Infrared remotes are simple and easy to use. In this tutorial we

will be connecting the IR receiver to the UNO, and then use a Library that was designed for this particular sensor.

In our sketch we will have all the IR Hexadecimal codes that are available on this remote, we will also detect if the code was recognized and also if we are holding down a key.

Component Required:

(1) x LONTEN Uno R3

(1) x IR receiver module

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

Component Introduction

IR RECEIVER SENSOR:

IR detectors are little microchips with a photocell that are tuned to listen to infrared light. They are almost always used for remote control detection - every TV and DVD player has one of these in the front to listen for the IR signal from the clicker. Inside the remote control is a

matching IR LED, which emits IR pulses to tell the TV to turn on, off or change channels. IR light is not visible to the human eye, which means it takes a little more work to test a setup.

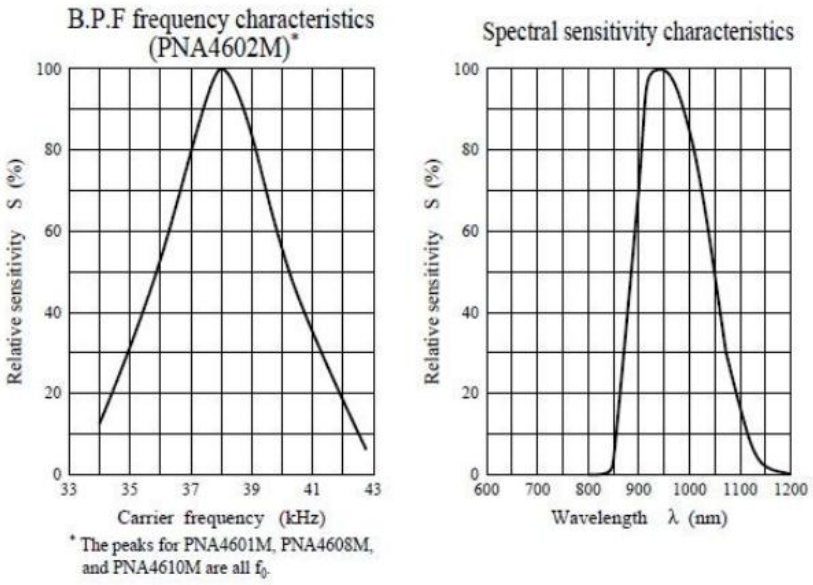
There are a few difference between these and say a CdS Photocells:

IR detectors are specially filtered for IR light, they are not good at detecting visible light. On the other hand, photocells are good at detecting yellow/green visible light, and are not good at IR light.

IR detectors have a demodulator inside that looks for modulated IR at 38 KHz. Just shining an IR LED won't be detected, it has to be PWM blinking at 38KHz.

Photocells do not have any sort of demodulator and can detect any frequency (including DC) within the response speed of the photocell (which is about 1KHz)IR detectors are digital out - either they detect 38KHz IR signal and output low (0V) or they do not detect any and output high (5V). Photocells act like resistors, the resistance changes depending on how much light they are exposed to.

What You Can Measure



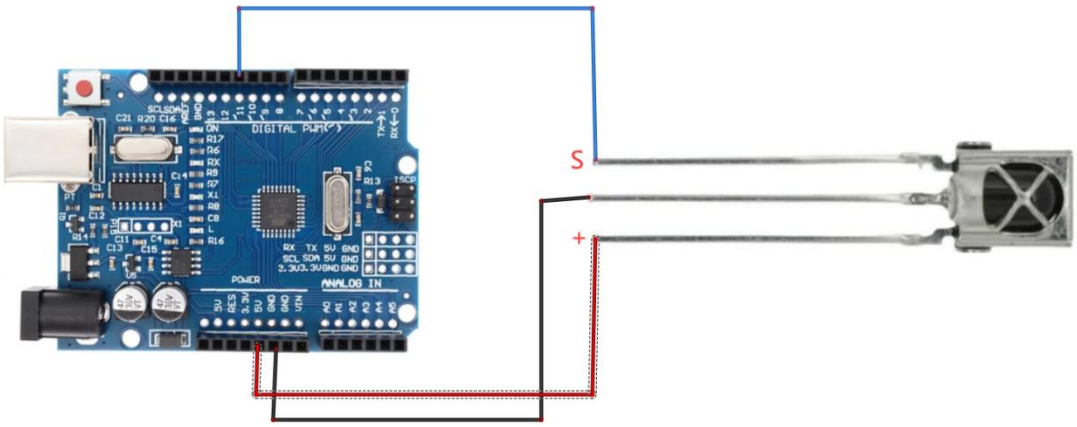
As you can see from these datasheet graphs, the peak frequency detection is at 38 KHz and the peak LED color is 940 nm. You can use from about 35 KHz to 41 KHz but the sensitivity will drop off so that it won't detect as well from afar. Likewise, you can use 850 to 1100 nm LEDs but they won't work as well as 900 to 1000nm so make sure to get matching LEDs! Check the datasheet for your IR LED to verify the wavelength.

Try to get a 940nm - remember that 940nm is not visible light!

Remote control code:

Attention:The remote control in this kit does not include the CR1220 battery, which needs to be purchased locally.

Circuit Connection



There are 3 connections to the IR Receiver.

The connections are: Signal, Voltage and Ground.

The "-" is the Ground, "S" is signal, and middle pin is Voltage 5V.

Code

After wiring, please open the program in the code folder- Lesson 10 IR Receiver Module and click UPLOAD to upload the program.

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

Next we will move the <RobotIRremote> out of the Library folder, we do this because that library conflicts with the one we will be using. You can just drag it back inside the library folder once you are done programming your microcontroller. Once you have installed the Library, just go ahead and restart your IDE Software.

Lesson 11 Eight LED with 74HC595

Overview

In this lesson, you will learn how to use eight large red LEDs with an UNO without needing to give up 8 output pins!

Although you could wire up eight LEDs each with a resistor to an UNO pin you would rapidly start to run out of pins on your UNO. If you don't have a lot of stuff connected to your UNO. It's OK to do so - but often times we want buttons, sensors, servos, etc. and before you know it you've got no pins left. So, instead of doing that, you are going to use a chip called the 74HC595 Serial to Parallel Converter. This chip has eight

outputs (perfect) and three inputs that you use to feed data into it a bit at a time.

This chip makes it a little slower to drive the LEDs (you can only change the LEDs about 500,000 times a second instead of 8,000,000 a second) but it's still really fast, way faster than humans can detect, so it's worth it!

Component Required:

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(8) x leds

(8) x 220 ohm resistors

(1) x 74hc595 IC

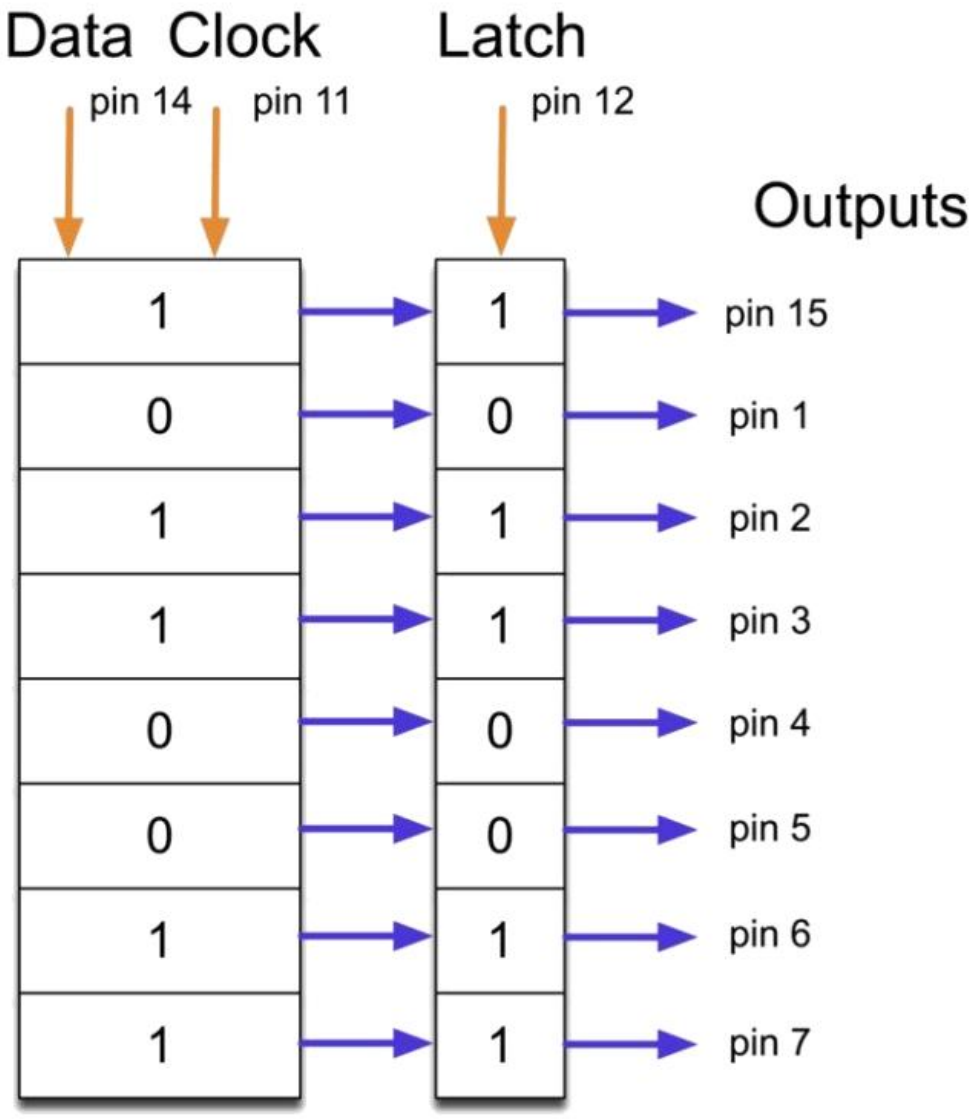
(14) x M-M wires (Male to Male jumper wires)

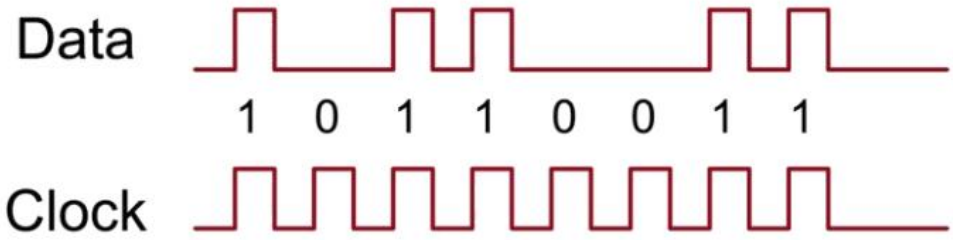
Component Introduction

74HC595 Shift Register:

The shift register is a type of chip that holds what can be thought of as eight memory locations, each of which can either be a 1 or a 0. To set

each of these values on or off, we feed in the data using the 'Data' and 'Clock' pins of the chip.



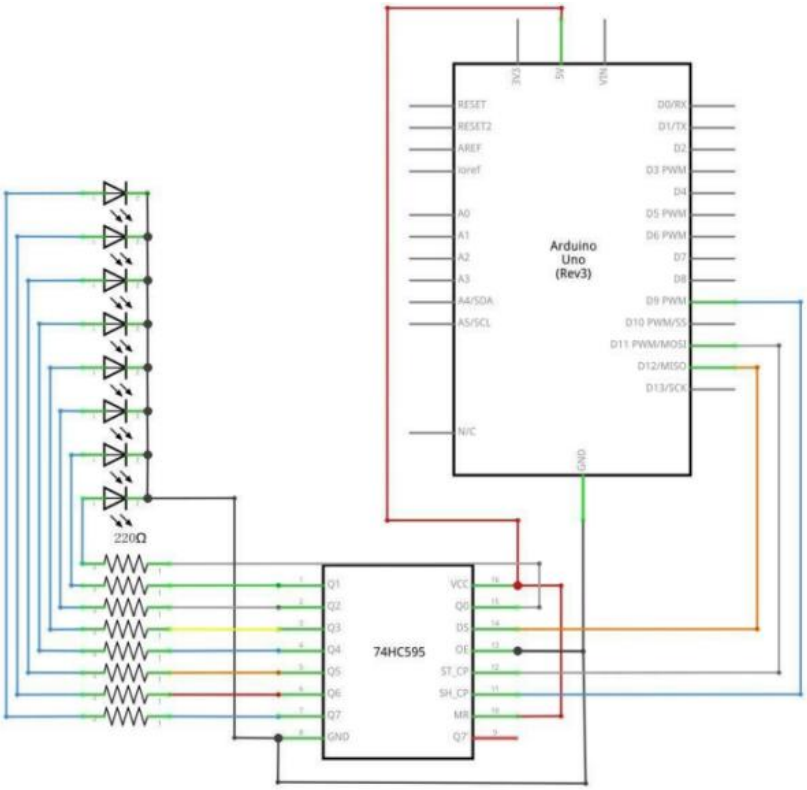


The clock pin needs to receive eight pulses. At each pulse, if the data pin is high, then a 1 gets pushed into the shift register; otherwise, a 0. When all eight pulses have been received, enabling the 'Latch' pin copies those eight values to the latch register. This is necessary; otherwise, the wrong LEDs would flicker as the data is being loaded into the shift register.

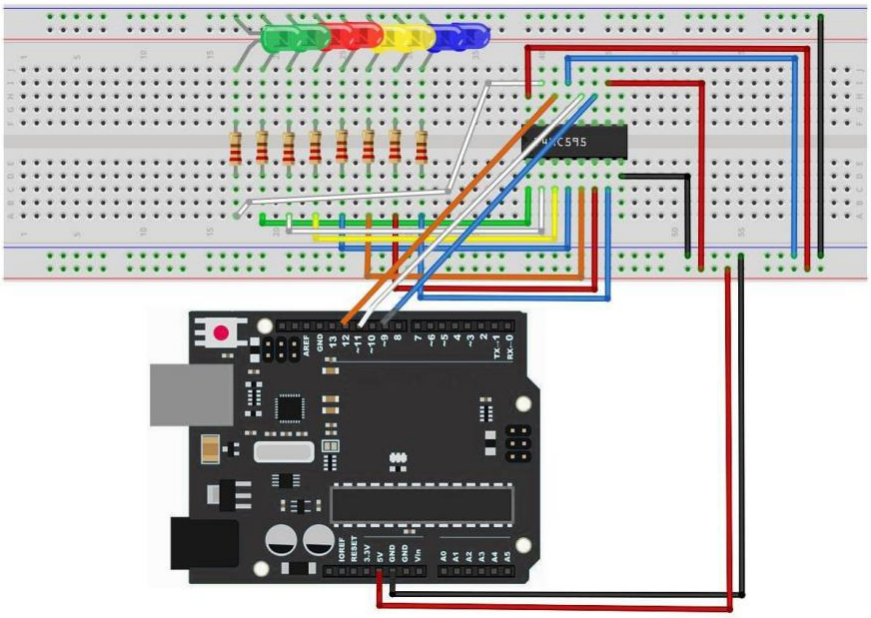
The chip also has an output enable (OE) pin, which is used to enable or disable the outputs all at once. You could attach this to a PWM-capable UNO pin and use 'analogWrite' to control the brightness of the LEDs. This pin is active low, so we tie it to GND.

Connection

Schematic



Circuit Connection



As we have eight LEDs and eight resistors to connect, there are actually quite a few connections to be made.

It is probably easiest to put the 74HC595 chip in first, as pretty much everything else connects to it. Put it so that the little U-shaped notch is towards the top of the breadboard. Pin 1 of the chip is to the left of this notch.

Digital 12 from the UNO goes to pin #14 of the shift register

Digital 11 from the UNO goes to pin #12 of the shift register

Digital 9 from the UNO goes to pin #11 of the shift register

All but one of the outputs from the IC is on the left side of the chip. Hence, for ease of connection, that is where the LEDs are, too.

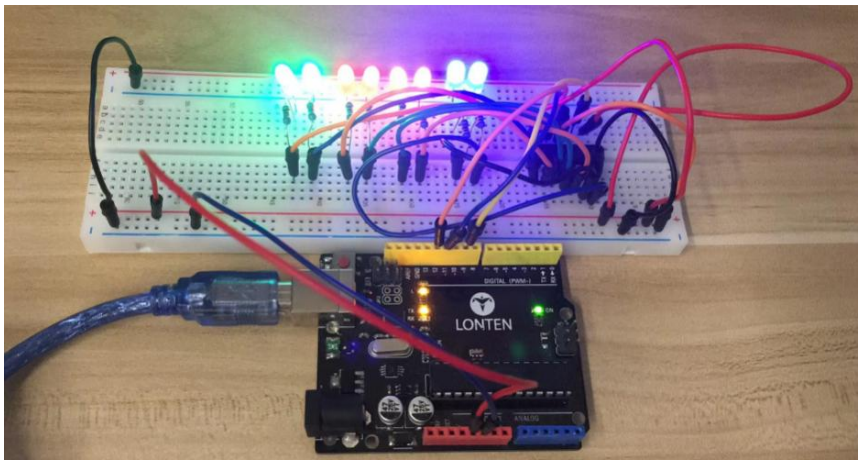
After the chip, put the resistors in place. You need to be careful that none of the leads of the resistors are touching each other. You should check this again before you connect the power to your UNO. If you find it difficult to arrange the resistors without their leads touching, then it helps to shorten the leads so that they are lying closer to the surface of the breadboard.

Next, place the LEDs on the breadboard. The longer positive LED leads must all be towards the chip, whichever side of the breadboard they are on.

Attach the jumper leads as shown above. Do not forget the one that goes from pin 8 of the IC to the GND column of the breadboard.

Load up the sketch listed a bit later and try it out. Each LED should light in turn until all the LEDs are on, and then they all go off and the cycle repeats.

Example picture



Code

After wiring, please open the program in the code folder- Lesson 11 Eight LED with 74HC595 and click UPLOAD to upload the program.

The first thing we do is define the three pins we are going to use. These are the UNO digital outputs that will be connected to the latch, clock and data pins of the 74HC595.

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

Next, a variable called 'leds' is defined. This will be used to hold the pattern of which LEDs are currently turned on or off. Data of type 'byte'

represents numbers using eight bits. Each bit can be either on or off, so this is perfect for keeping track of which of our eight LEDs are on or off.

```
byte leds = 0;
```

The 'setup' function just sets the three pins we are using to be digital outputs.

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

The 'loop' function initially turns all the LEDs off, by giving the variable 'leds' the value 0. It then calls 'updateShiftRegister' that will send the 'leds' pattern to the shift register so that all the LEDs turn off. We will deal with how 'updateShiftRegister' works later.

The loop function pauses for half a second and then begins to count from 0 to 7 using the 'for' loop and the variable 'i'. Each time, it uses the Arduino function 'bitSet' to set the bit that controls that LED in the variable 'leds'. It then also calls 'updateShiftRegister' so that the leds update to reflect what is in the variable 'leds'.

There is then a half second delay before 'i' is incremented and the next LED is lit.

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

The function 'updateShiftRegister', first of all sets the latchPin to low, then calls the UNO function 'shiftOut' before putting the 'latchPin' high again. This takes four parameters, the first two are the pins to use for Data and Clock respectively.

The third parameter specifies which end of the data you want to start at. We are going to start with the right most bit, which is referred to as the 'Least Significant Bit' (LSB).

The last parameter is the actual data to be shifted into the shift register, which in this case is 'leds'.

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

If you wanted to turn one of the LEDs off rather than on, you would call a similar Arduino function (bitClear) with the 'leds' variable. This will set that bit of 'leds' to be 0 and you would then just need to follow it with a call to 'updateShiftRegister' to update the actual LEDs.

Lesson 12 Photocell

Overview

In this lesson, you will learn how to measure light intensity using an Analog Input.

You will build on lesson 17 and use the level of light to control the number of LEDs to be lit.

The photocell is at the bottom of the breadboard, where the pot was above.

Component Required:

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(8) x leds

(8) x 220 ohm resistors

(1) x 1k ohm resistor

(1) x 74hc595 IC

(1) x Photoresistor (Photocell)

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

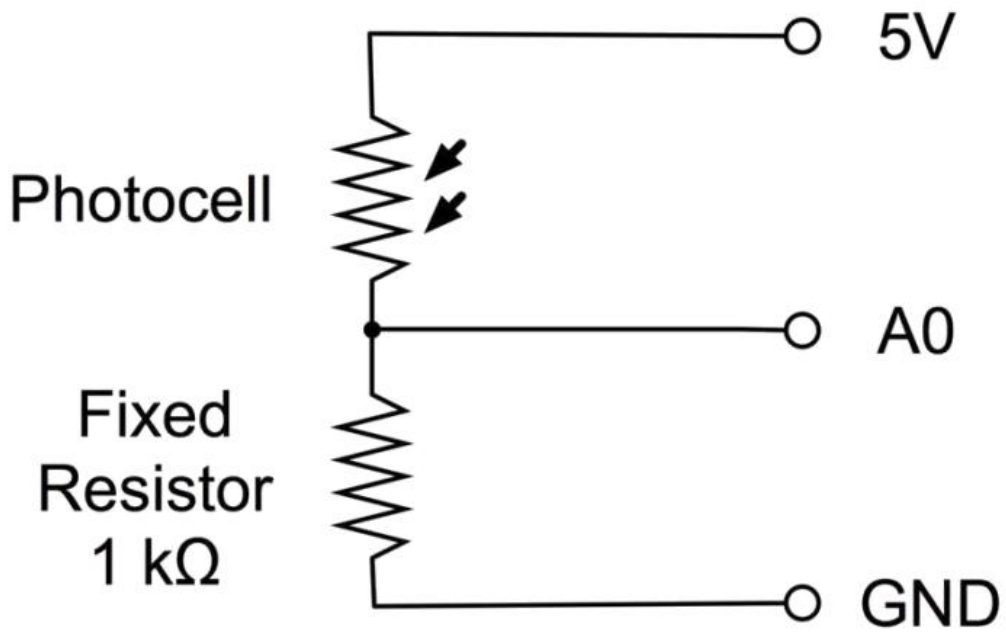
Component Introduction

PHOTOCELL:

The photocell used is of a type called a light dependent resistor, sometimes called an LDR. As the name suggests, these components act just like a resistor, except that the resistance changes in response to how much light is falling on them.

This one has a resistance of about 50 kΩ in near darkness and 500 Ω in bright light.

To convert this varying value of resistance into something we can measure on an UNO R3 board's analog input, it needs to be converted into a voltage. The simplest way to do that is to combine it with a fixed resistor.



The resistor and photocell together behave like a pot. When the light is very bright, then the resistance of the photocell is very low compared

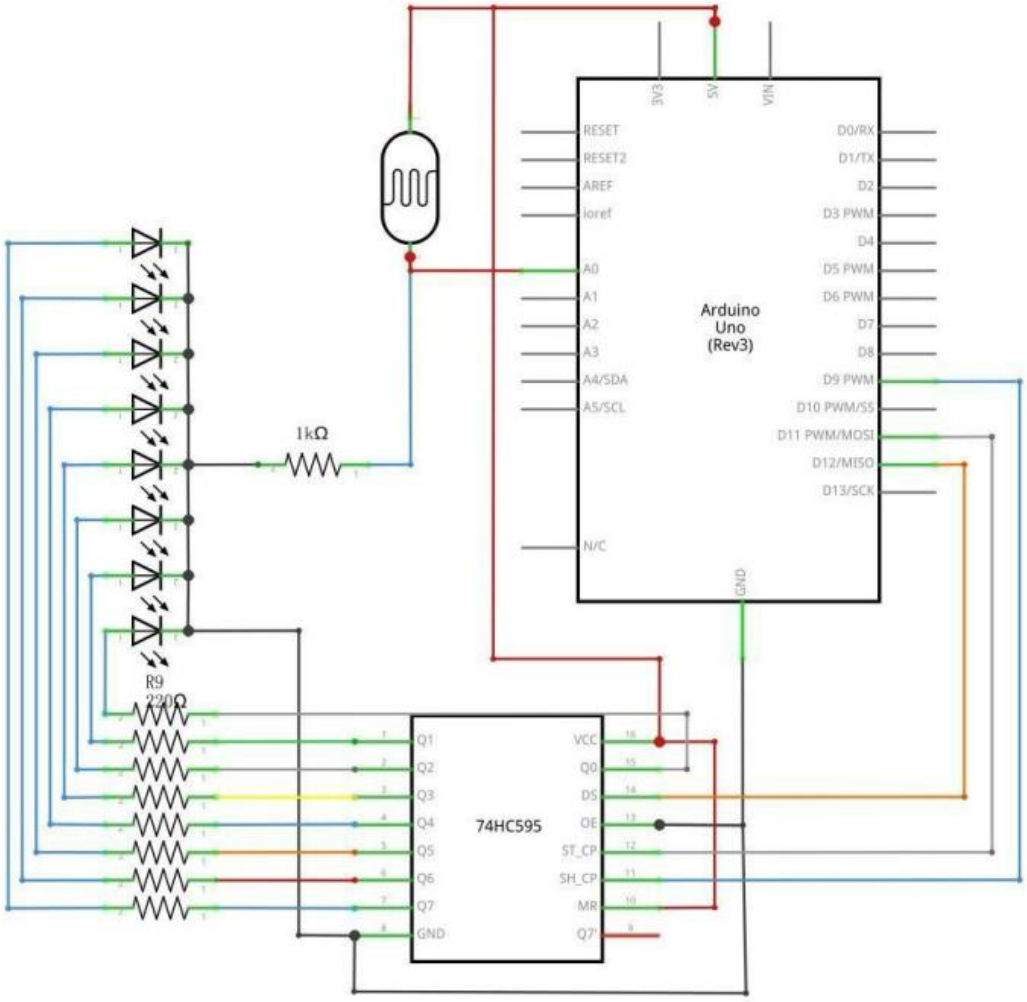
with the fixed value resistor, and so it is as if the pot were turned to maximum.

When the photocell is in dull light, the resistance becomes greater than the fixed 1 kΩ resistor and it is as if the pot were being turned towards GND.

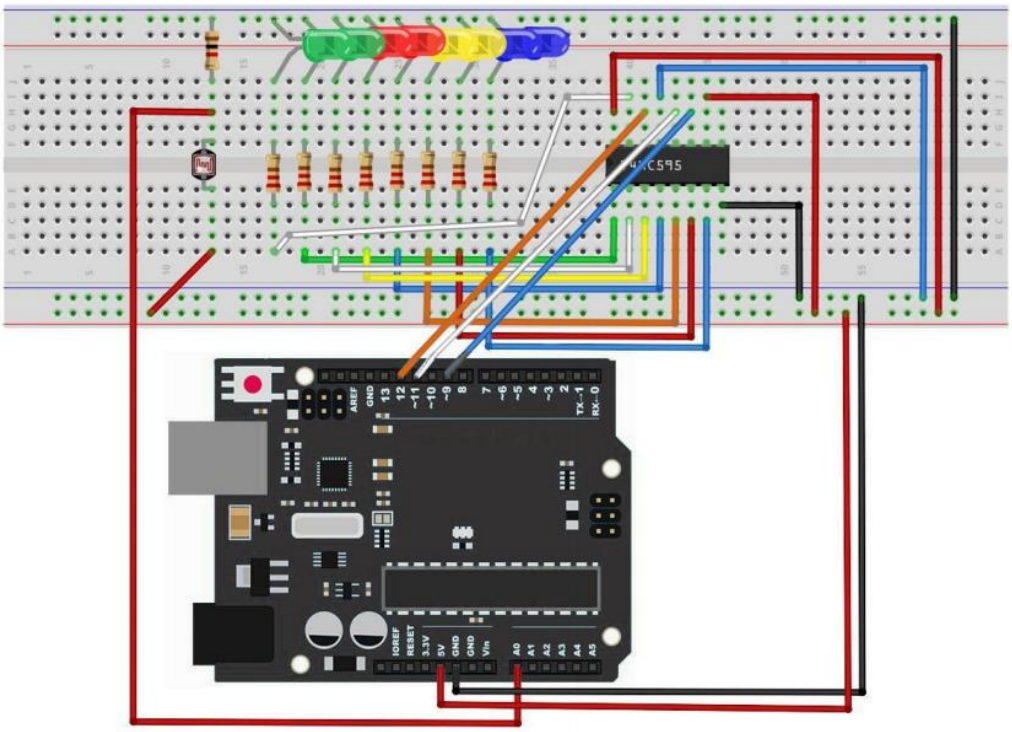
Load up the sketch given in the next section and try covering the photocell with your finger, and then holding it near a light source.

Connection

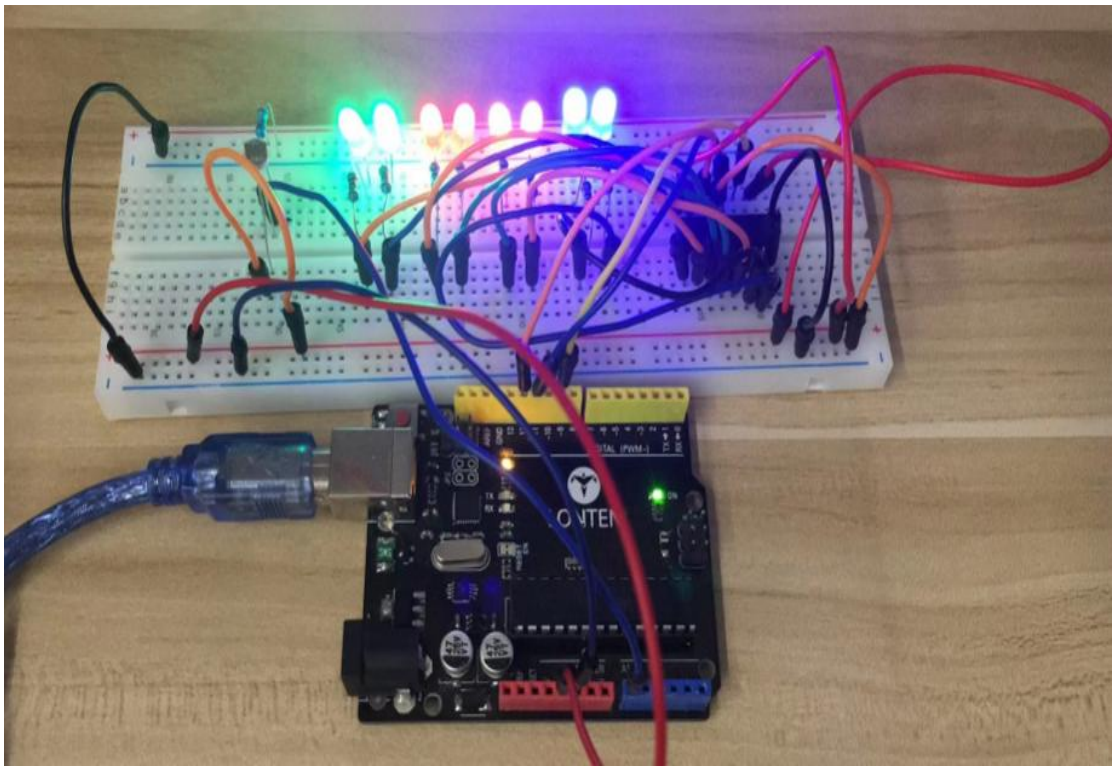
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 12 Photocell and click UPLOAD to upload the program.

The first thing to note is that we have changed the name of the analog pin to be 'lightPin' rather than 'potPin' since we no longer have a pot connected.

The only other substantial change to the sketch is the line that calculates how many of the LEDs to light:

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

This time, we divide the raw reading by 57 rather than 114. In other words, we divide it by half as much as we did with the pot to split it into nine zones, from no LEDs lit to all eight lit. This extra factor is to account for the fixed 1 kΩ resistor. This means that when the photocell has a resistance of 1 kΩ (the same as the fixed resistor), the raw reading will be 1023 / 2 = 511. This will equate to all the LEDs being lit and then a bit (numLEDSLit) will be 8.

Lesson 13 74HC595 And Segment Display

Overview

In learning Lesson, we will use the 74HC595 shift register to control the segment display. The segment display will show number from 9-0.

Component Required

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(1) x 74HC595 IC

(1) x 1 Digit 7-Segment Display

(8) x 220 ohm resistors

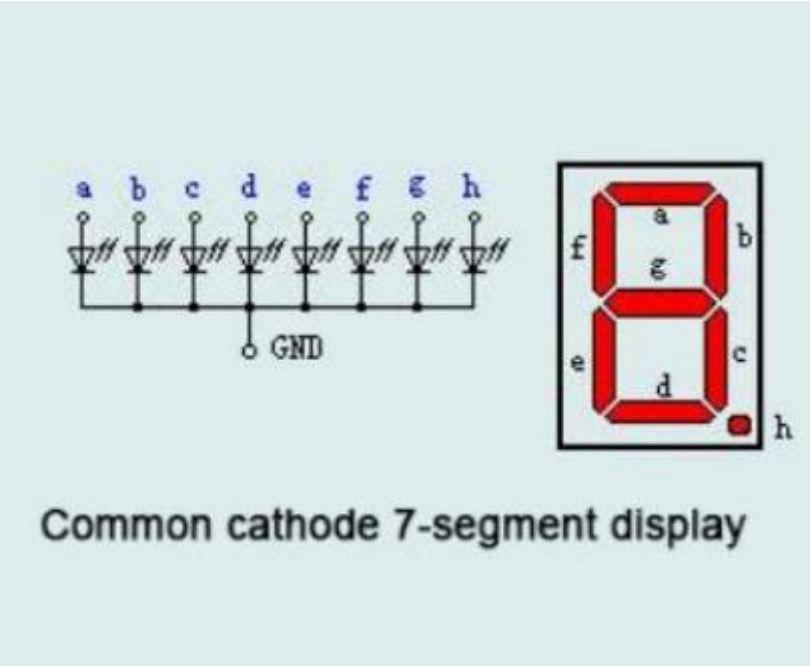
(26) x M-M wires (Male to Male jumper wires)

Component Introduction

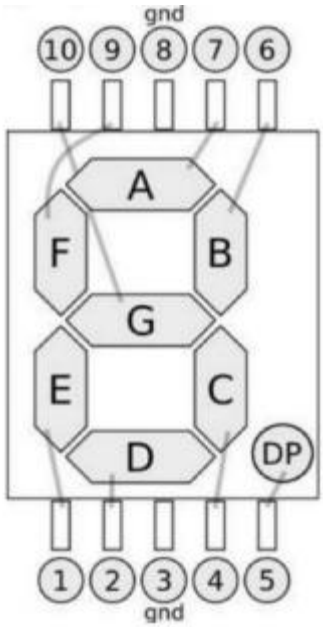
Seven segment display

LED segment displays are common for displaying numerical information. It's widely applied on displays of electromagnetic oven, full automatic washing machine, water temperature display, electronic clock etc. It is necessary that we learn how it works. LED segment display is a semiconductor light-emitting device. Its basic unit is a light-emitting diode (LED). LED segment display can be divided into 7-segment display and 8-segment display according to the number of segments.

8-segment display has one more LED unit (for decimal point display) than 7-segment one. In this experiment, we use a 8-segment display.



According to the wiring method of LED units, LED segment displays can be divided into display with common anode and display with common cathode. Common anode display refers to the one that combine all the anodes of LED units into one common anode (COM).For the common anode display, connect the common anode (COM) to +5V. When the cathode level of a certain segment is low, the segment is on; when the cathode level of a certain segment is high, the segment is off. For the common cathode display, connect the common cathode (COM) to GND. When the anode level of a certain segment is high, the segment is on; when the anode level of a certain segment is low, the segment is off.



Each segment of the display consists of an LED. So when you use it, you also need use a current-limiting resistor. Otherwise, LED will be burnt out. In this experiment, we use a common cathode display. As we mentioned above, for common cathode display, connect the common cathode (COM) to GND. When the anode level of a certain segment is high, the segment is on; when the anode level of a certain segment is low, the segment is off.

74HC595

The following table shows the seven-segment display 74HC595 pin correspondence table:

74HC595 pin	Seven shows remarkable control pin (stroke)

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

Step one: Connect 74HC595

First, the wiring is connected to power and ground:

VCC (pin 16) and MR (pin 10) connected to 5V

GND (pin 8) and OE (pin 13) to ground

Connection DS, ST_CP and SH_CP

pin:

DS (pin 14) connected to UNO R3 board pin 2 (the figure below the yellow line)

ST_CP (pin 12, latch pin) connected to UNO R3 board pin 3 (FIG blue line below)

SH_CP (pin 11, clock pin) connected to UNO R3 board pin 4 (the figure below the white line)

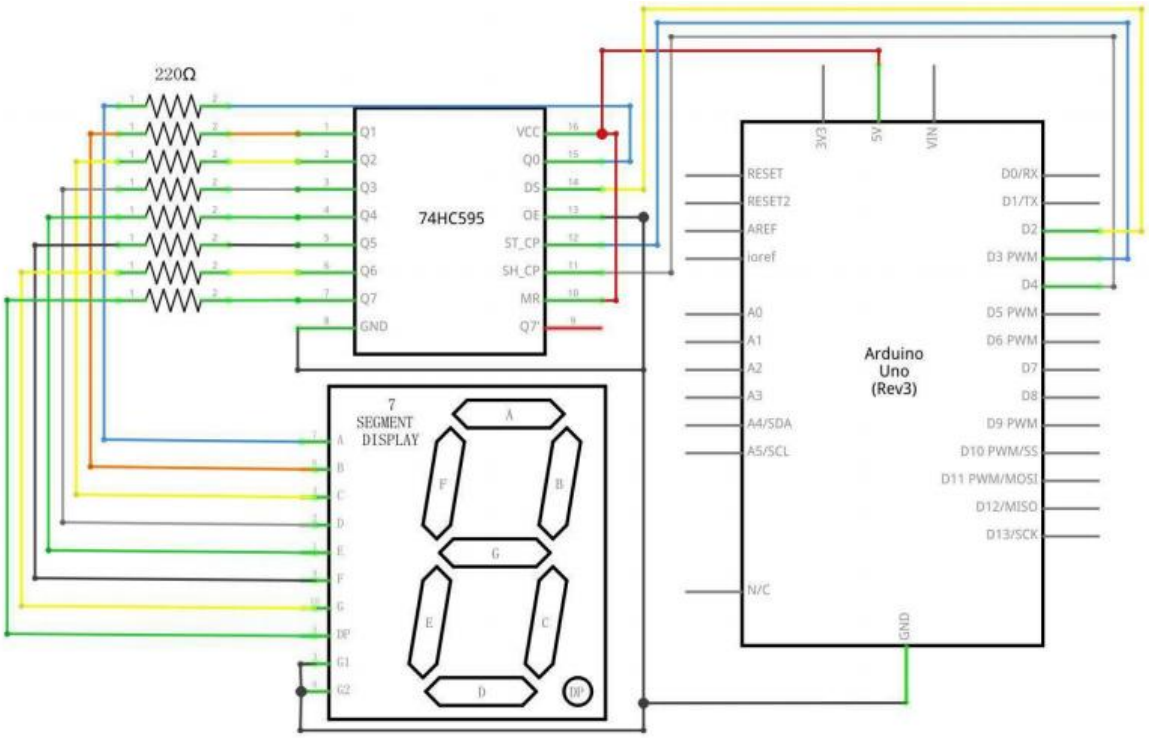
Step two: Connect the seven segment display

The seven-segment display 3, 8 pin to UNO R3 board GND (This example uses the common cathode, if you use the common anode, please connect the 3, 8 pin to UNO R3 board + 5V)

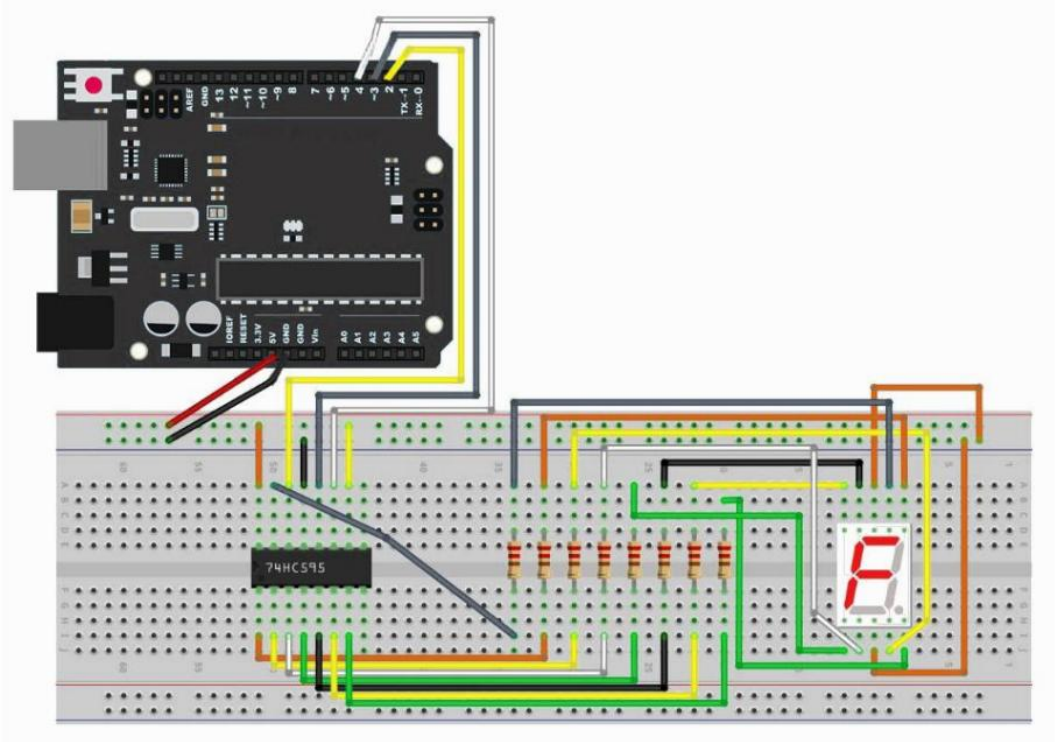
According to the table above, connect the 74HC595 Q0 ~ Q7 to seven-segment display corresponding pin (A ~ G and DP), and then each foot in a 220 ohm resistor in series.

Connection

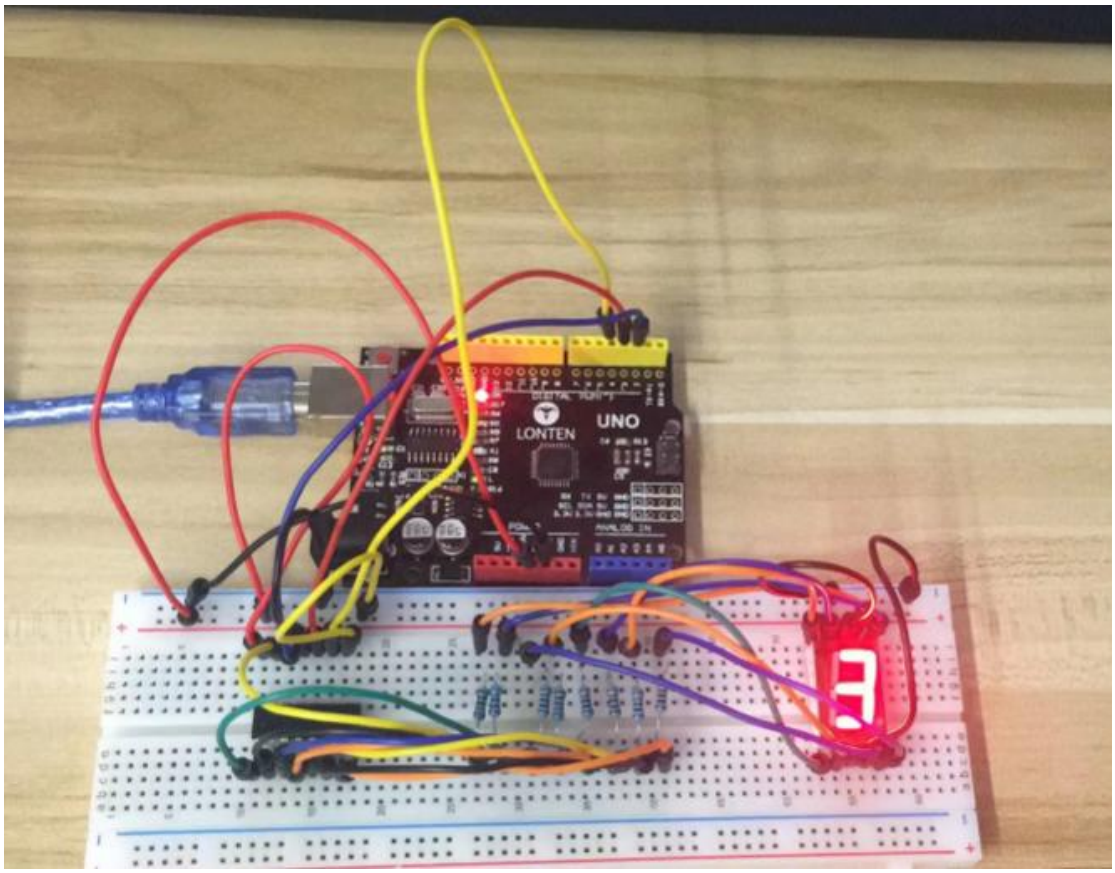
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 13 74HC595 And Segment Display and click UPLOAD to upload the program.

LED segment display repeat displays number 9 to 0.

Lesson 14 Four Digital Seven Segment Display

Overview

In this lesson, you will learn how to use a 4-digit 7-segment display.

When using 1-digit 7-segment display, please notice that if it is common anode, the common anode pin connects to the power source; if it is common cathode, the common cathode pin connects to the GND.

When using 4-digit 7-segment display, the common anode or common cathode pin is used to control which digit is displayed. Even though there is only one digit working, the principle of Persistence of Vision enables you to see all numbers displayed because each the scanning speed is so fast that you hardly notice the intervals.

Component Required:

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(1) x 74HC595 IC

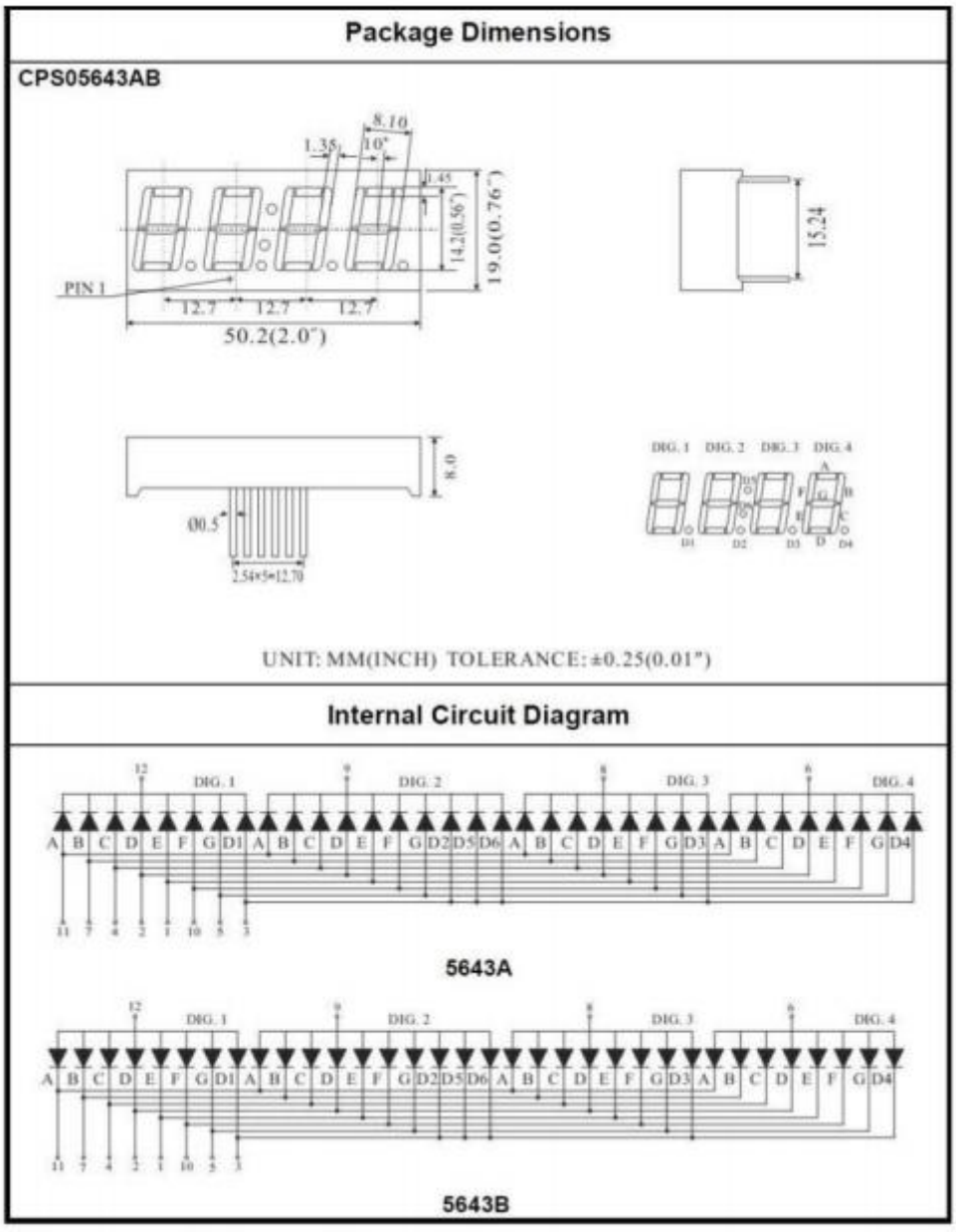
(1) x 4 Digit 7-Segment Display

(4) x 220 ohm resistors

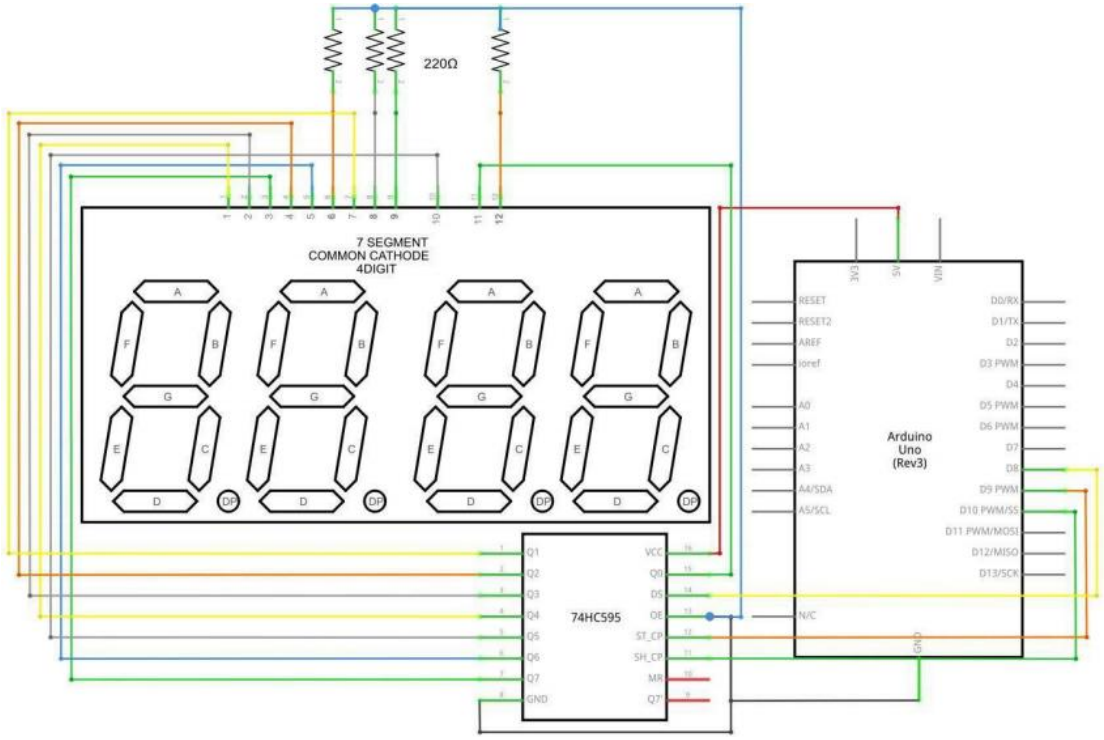
(23) x M-M wires (Male to Male jumper wires)

Component Introduction

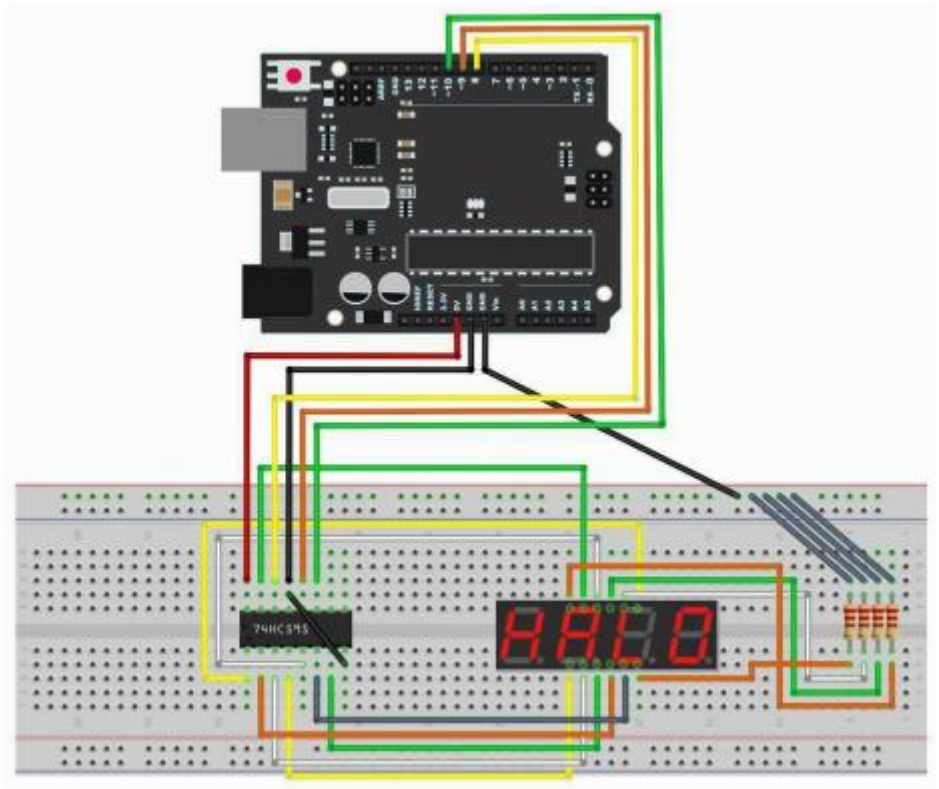
Four Digital Seven segment display



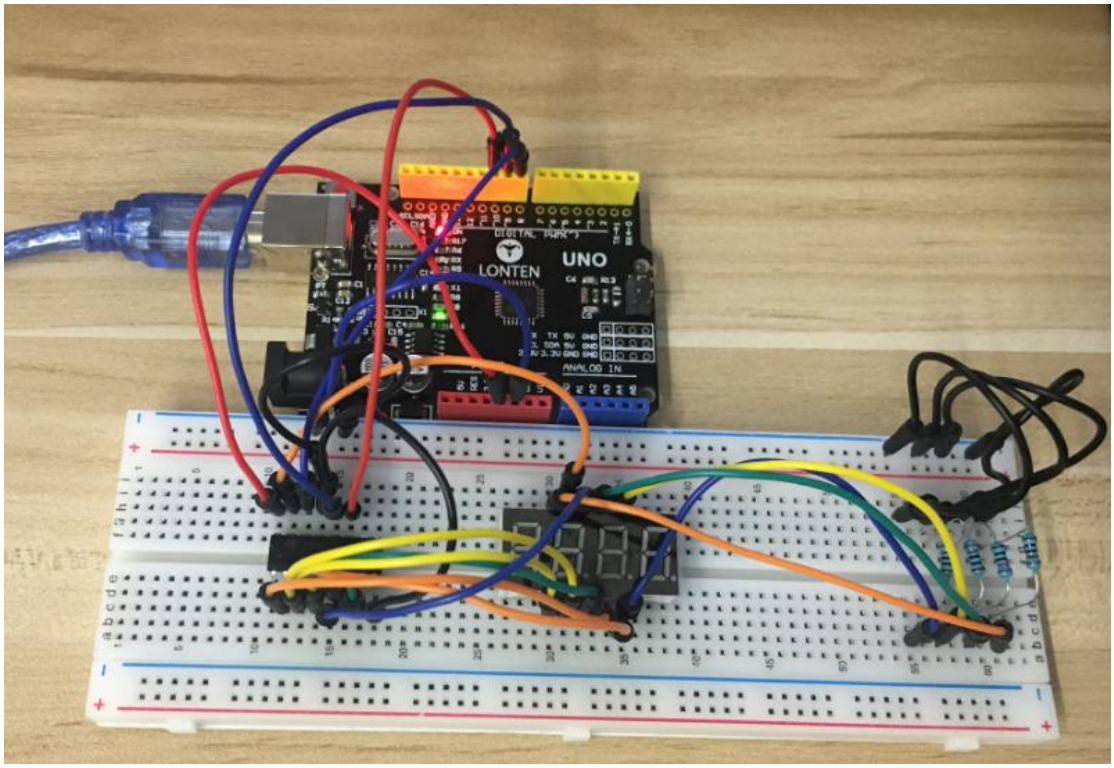
Connection
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 14 Four Digital Seven Segment Display and click UPLOAD to upload the program.

Lesson 15 8x8 LED Dot Matrix Module

Overview

LED dot matrix screens are composed of LED light-emitting diodes, which display text, images, animations, videos, etc. by controlling the

LED to turn on or off. They are widely used in public places for information display, such as advertising screens, bulletin boards, etc. LED dot matrix screens can be divided into monochrome, bicolor, tricolor lights, etc. according to the LED emitting color, and can display red, yellow, green, and even true color. Divided into 4 based on the number of LEDs × 4. 8 × 8. 16 × 16 different types. The process requirements for multi-color dot matrix screens are relatively high, and it is necessary to consider the impact of color mixing on colors. Here we use monochrome 8 × Use an 8-point matrix screen to understand its principle.

Component Required:

(1) x LONTEN Uno R3

(1) x 8x8 LED Dot Matrix module

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

Component Introduction

8x8 LED Dot Matrix Module

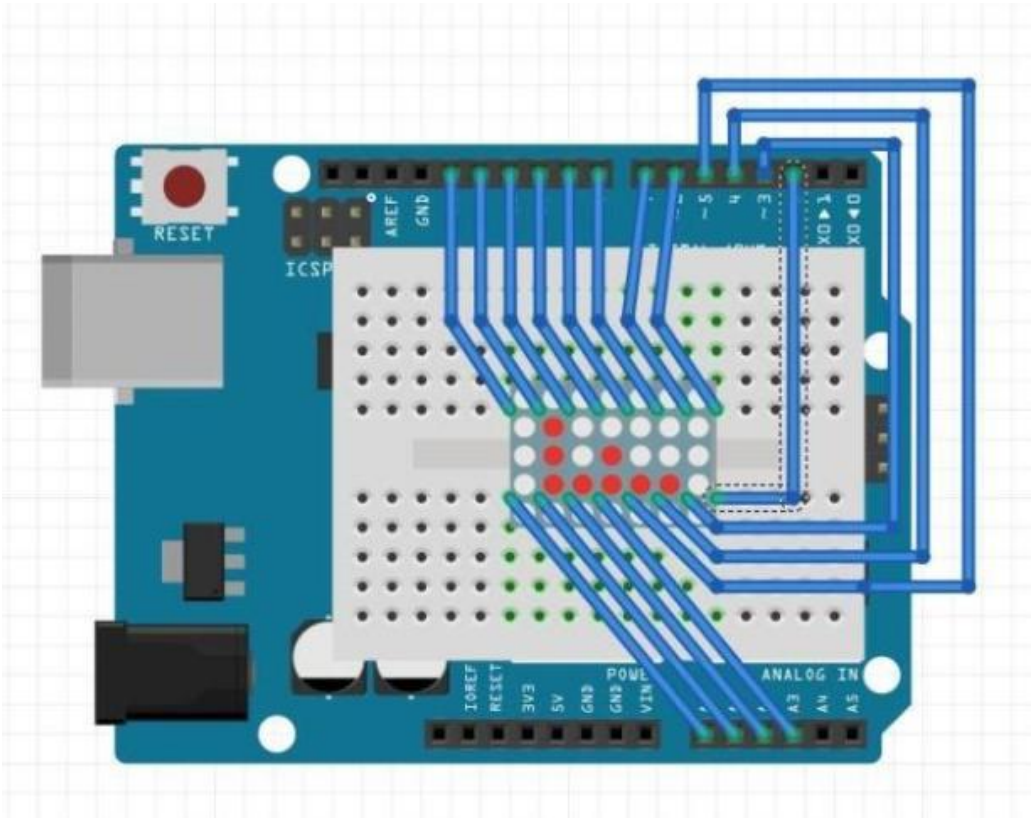
8 × 8-point matrix screen has 16 pins, with the side with silk screen facing downwards and numbered counterclockwise as 1.8 or 9.16.

Circuit Connection

According to the definition of dot matrix screen pins, [9, 14, 8, 12, 1, 7, 2, 5] of the dot matrix screen are respectively connected to [6, 11, 5, 9, 14, 4, 15, 2] of the development board, and these 8 pins are the positive pole of the LED;

The [13, 3, 4, 10, 6, 11, 15, 16] of the dot matrix screen are respectively connected to the [10, 16, 17, 7, 3, 8, 12, 13] of the development board, and these 8 pins are the negative pole of the LED.

It should be noted that A0A5 of Uno R3 development board can also be used for regular GPIO, with numbers 1419.



Code

The LED dot matrix screen lights up and then goes out, and then lights up column by column and row by row.

Lesson 16 RC522 RFID Module

Overview

In this lesson, you will learn to how to apply the RC522 RFID Reader Module on UNO R3. This module uses the Serial Peripheral Interface (SPI) bus to communicate with controllers such as Arduino, Raspberry Pi, beagle board, etc.

Component Required:

(1) x LONTEN Uno R3

(1) x RC522 RFID module

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

Component Introduction

RC522

The MFRC522 is a highly integrated reader/writer for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO 14443A / MIFARE® mode.

The MFRC522's internal transmitter part is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443A/MIFARE® cards and transponders without additional active circuitry. The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO/IEC 14443A/MIFARE® compatible cards and transponders. The digital part handles the complete ISO/IEC 14443A framing and error detection (Parity & CRC).The MFRC522 supports MIFARE®Classic (e.g. MIFARE® Standard) products. The MFRC522 supports contactless communication using MIFARE® higher transfer speeds up to 848 kbit/s in both directions.

Various host interfaces are implemented:

• SPI interface

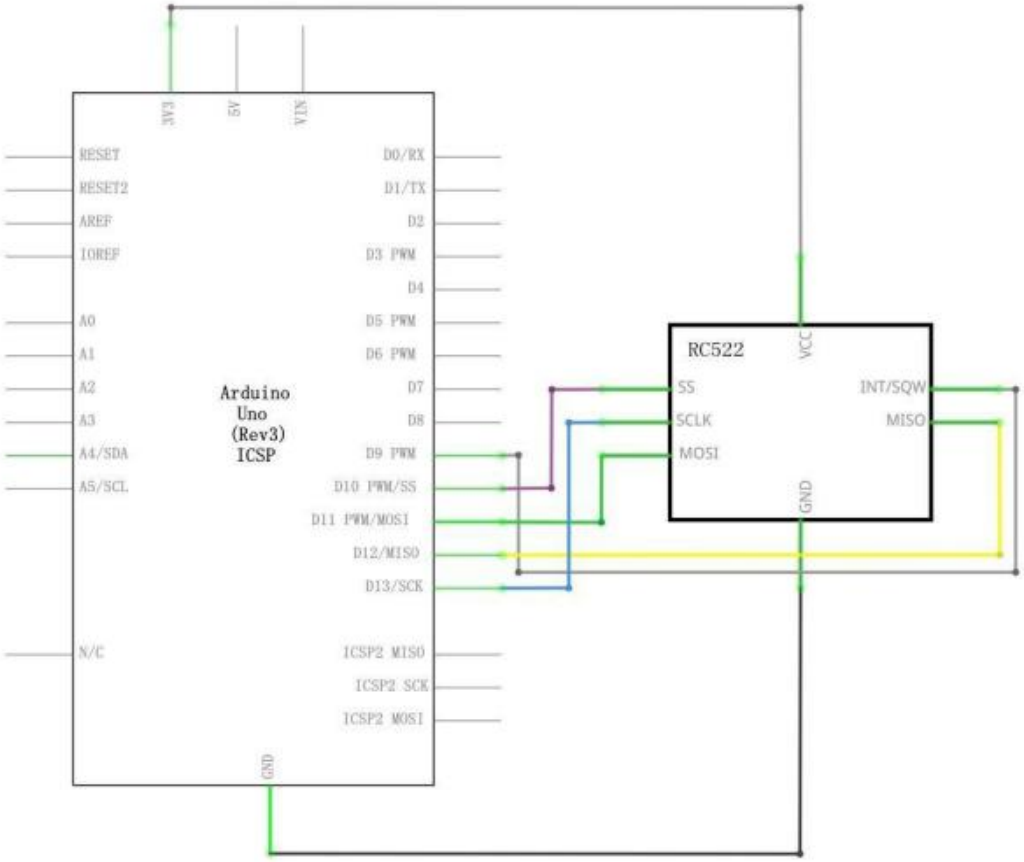
• Serial UART (similar to RS232 with voltage levels according pad voltage supply)

• I2C interface.

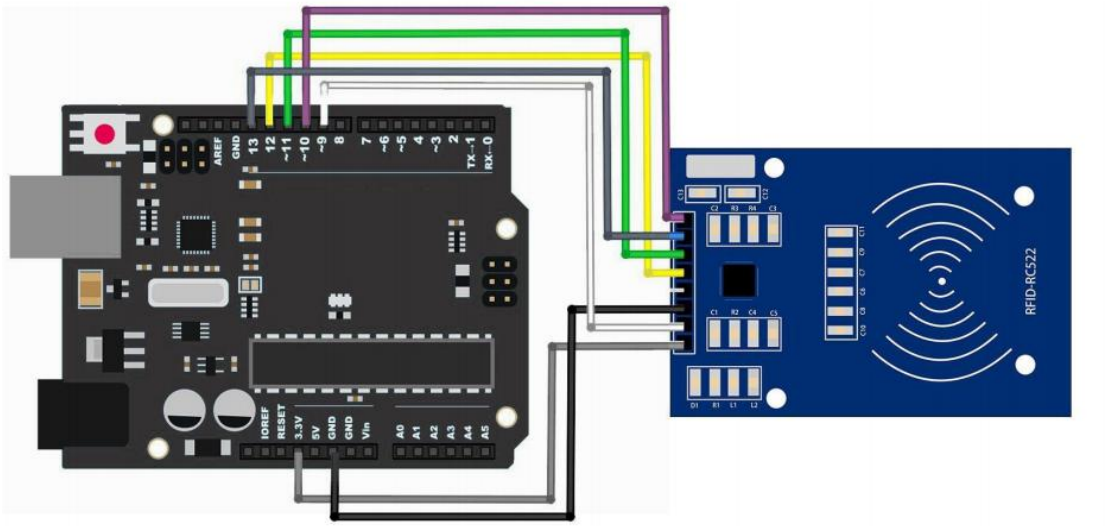
The figure below shows a typical circuit diagram, using a complementary antenna connection to the MFRC522.

Connection

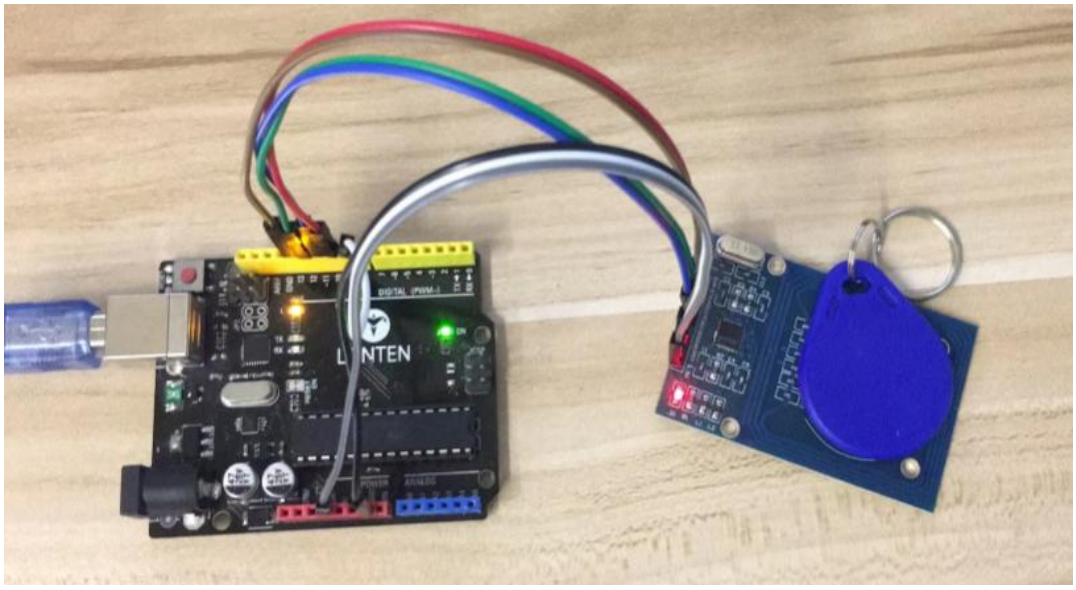
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 16 RC522 RFID Module and press UPLOAD to upload the program.

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

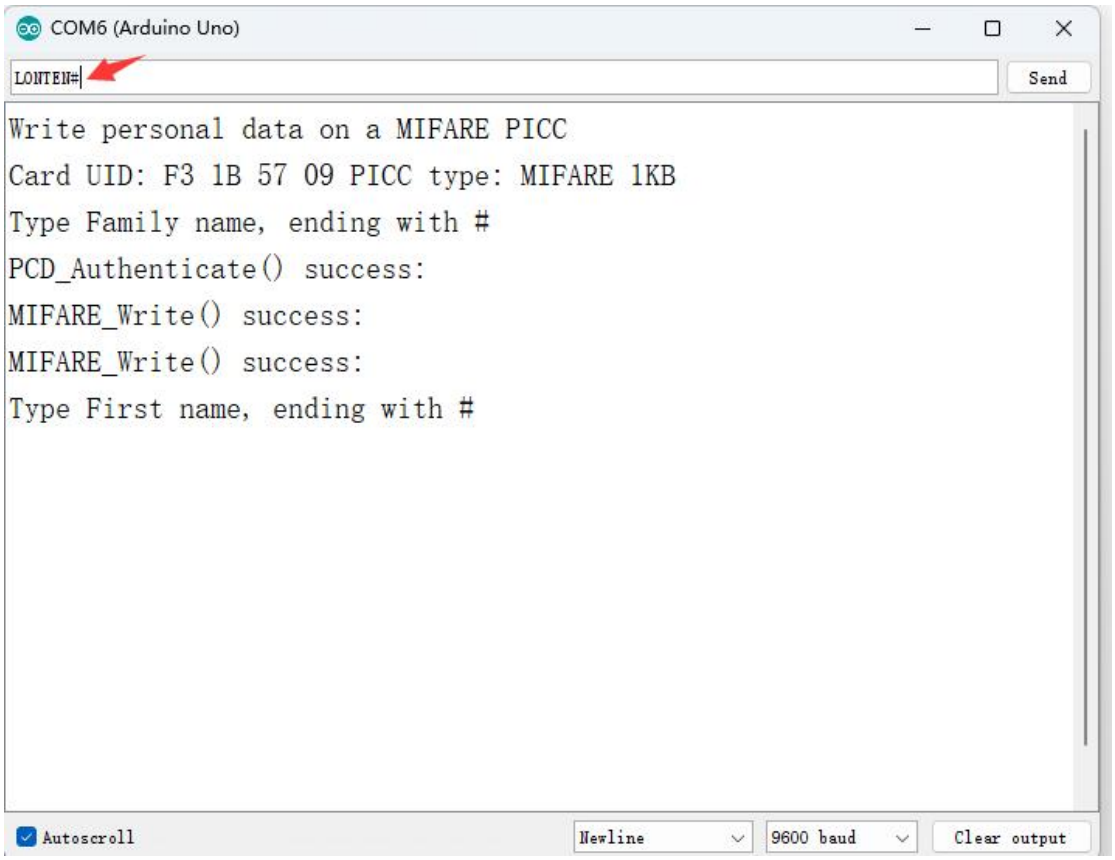
#define RST_PIN 9 // Configurable, see typical pin layout above

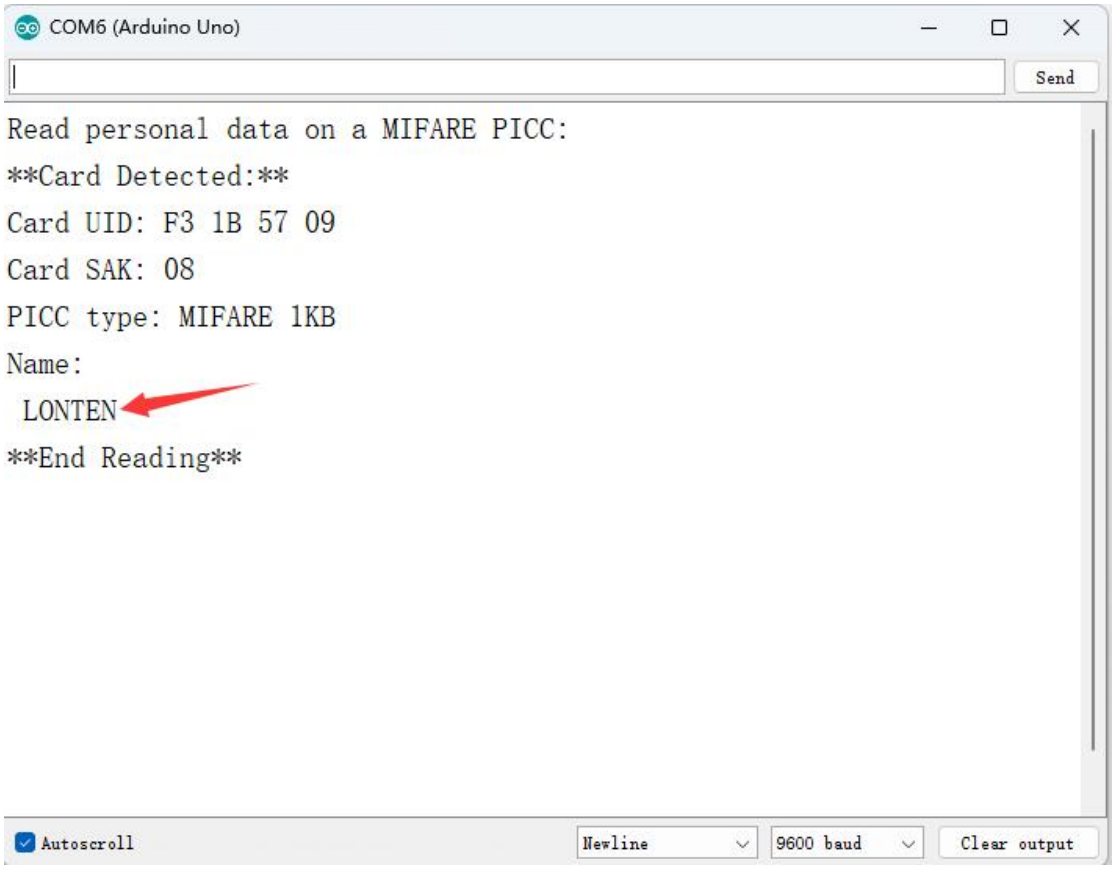
#define SS_PIN 10 // Configurable, see typical pin layout above

The locations of SPI pins vary with different chips, and you have to make a minor modification of the function.

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

Enter information here, ending with #





Lesson 17 LCD1602 I2C Module

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 I2C library, but in the next lesson, we will get our display to show the temperature, using sensors.

Component Required:

(1) x LONTEN Uno R3

(1) x LCD1602 I2C module

(1) x Potentiometer (10k)

(1) x 830 tie-points Breadboard

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

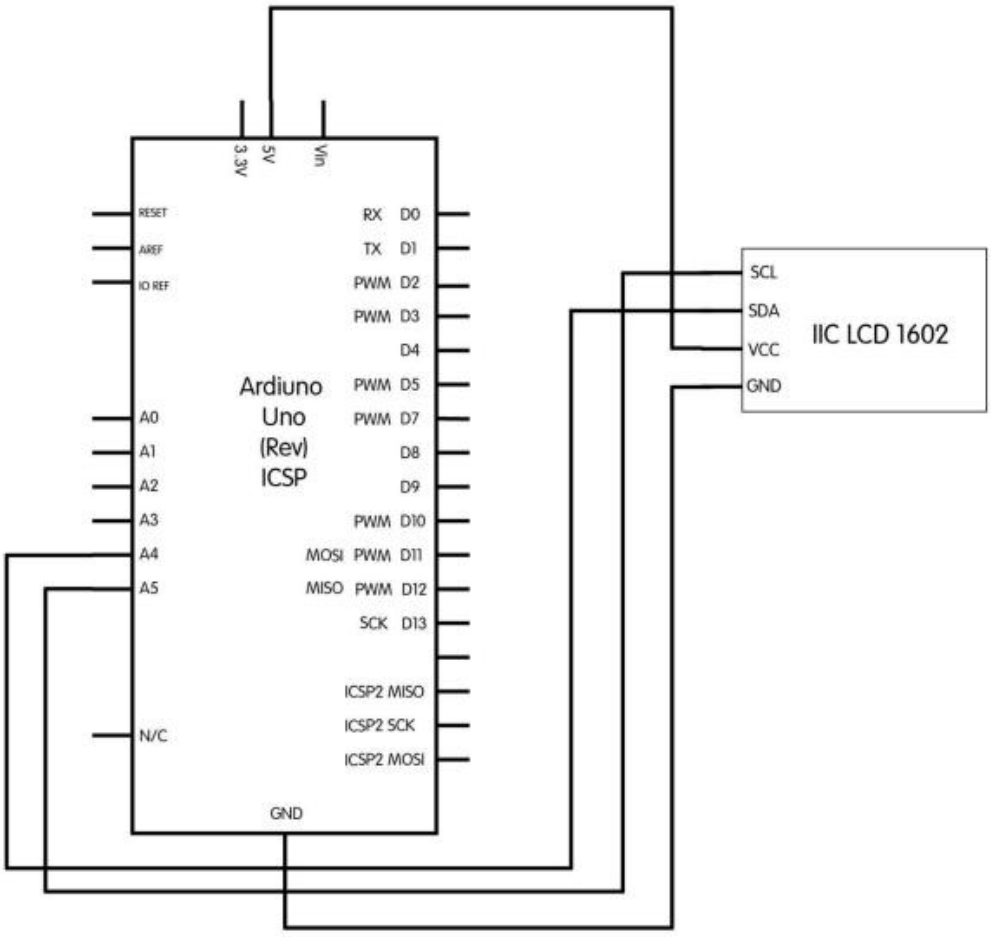
Component Introduction:

This is great LCD display compatible with arduino. With limited pin resources, your project will quickly run out of resources using normal LCDs. With this I2C interface LCD module, you only need 2 lines (I2C)to display the information.If you already have I2C devices in your

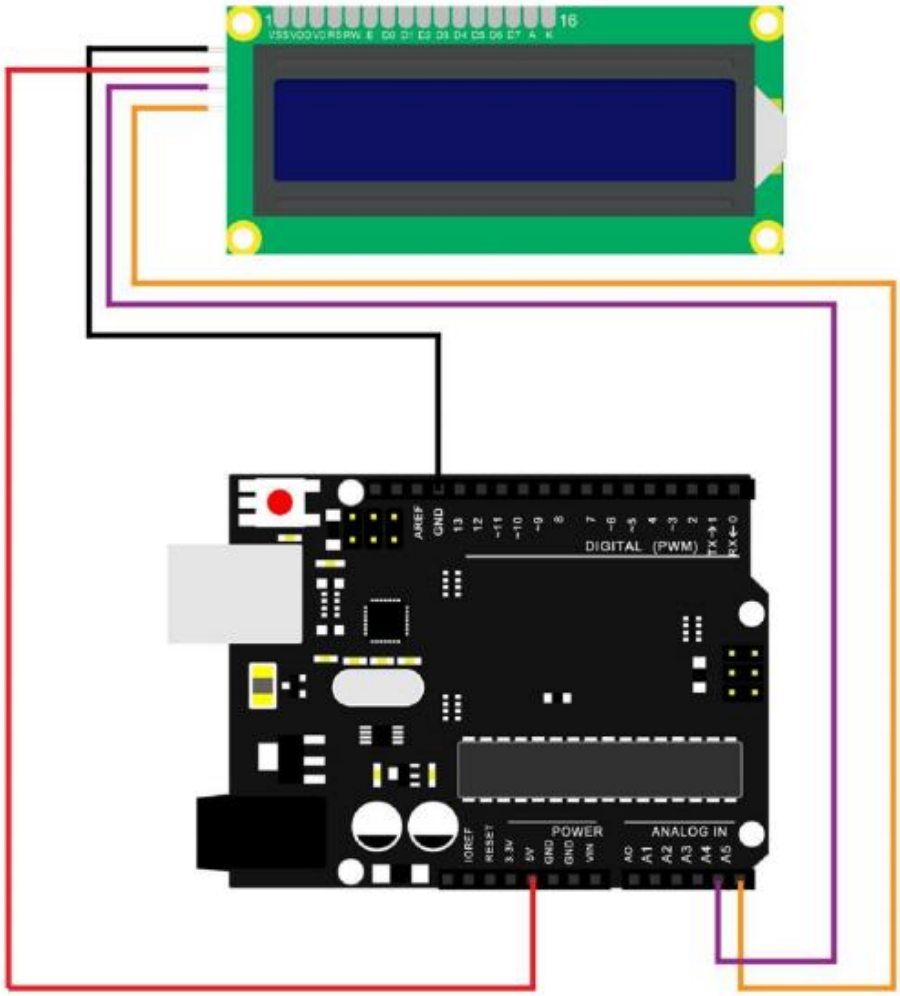
project, this LCD module actually cost no more resources at all. The address can be set 0x27.

Connection

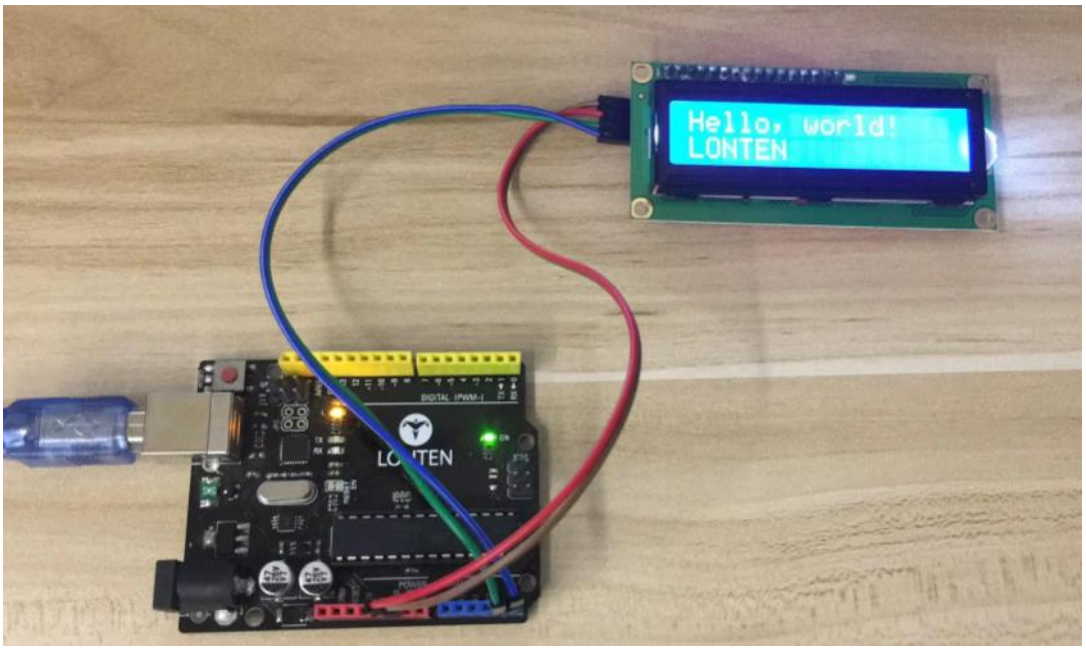
Schematic



Circuit Connection



Example picture



Code

After wiring, please open the program in the code folder- Lesson 17 LCD Display and click UPLOAD to upload the program.

Before you can run this, make sure that you have installed the <LiquidCrystal_I2C > library or re-install it, if necessary. Otherwise, your code won't work.

Upload the code to your Arduino board and you should see the message " Hello, world! LONTEN" displayed.

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

#include <LiquidCrystal_I2C.h>

This tells Arduino that we wish to use the LiquidCrystal_I2C library.

Next set the LCD address to 0x27 for a 16 chars and 2 line display

LiquidCrystal_I2C lcd(0x27,16,2);

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 four commands:

lcd.setCursor(0,0);

lcd.print("Hello, world!");

lcd.setCursor(0,1);

lcd.print("LONTEN");

The lcd.setCursor(0,0) tells the Begin displaying the next line of code in the first row and first column.

The lcd.setCursor(0,1) tells the Begin displaying the next line of code in the second row and first column.

lcd.print("Hello, world!");

lcd.print("LONTEN");

Tell screen display

“ Hello, world! LONTEN”

Lesson 18 Relay

Overview

In this lesson, you will learn how to use a relay.

Component Required:

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(1) x Fan blade and 3-6v dc motor

(1) x 5v Relay

(1) x LED

(6) x wires

Component Introduction

Relay:

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used as in solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete

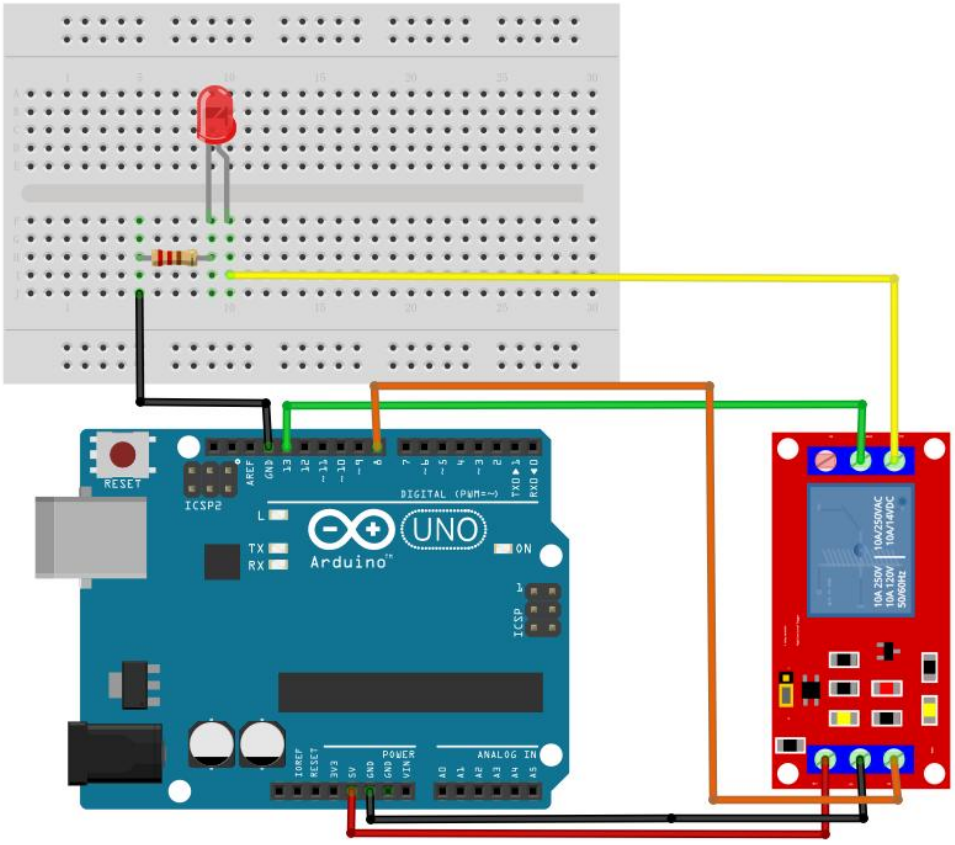
electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long-distance telegraph circuits as amplifiers. They repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform the switching.

Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults. In modern electric power systems, these functions are performed by digital instruments called "protective relays".

Below is the schematic of how to drive relay with Arduino.

Circuit Connection



Code

After wiring, please open the program in the code folder- Lesson 18 Relay and click UPLOAD to upload the program.

This relay module is active HIGH level.

Wire it up well, powered up, then upload the above code to the board.

You will see the relay is turned on（ON connected, NC disconnected）for two seconds, then turned off for two seconds（NC closed,ON disconnected）, repeatedly and circularly.

When the relay is turned on, external LED is on. If relay is turned off, external LED is off.

Lesson 19 Stepper Motor

Overview

In this lesson, you will learn a fun and easy way to drive a stepper motor. The stepper we are using comes with its own driver board making it easy to connect to our UNO.

Component Required:

(1) x LONTEN Uno R3

(1) x 830 tie-points breadboard

(1) x ULN2003 stepper motor driver module

(1) x Stepper motor

(1) x 9V1A Adapter

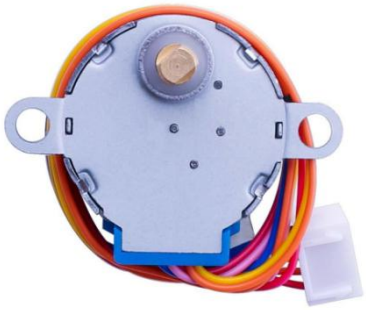
(6) x F-M wires (Female to Male DuPont wires)

(1) x M-M wire (Male to Male jumper wire)

Component Introduction

Stepper Motor

LROBRUYA



A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied. One of the most significant advantages of a stepper motor is its ability to be accurately controlled in an open loop system. Open loop control means no feedback information about position is needed. This type of control eliminates the need for expensive sensing and feedback devices such as optical encoders. Your position is known simply by keeping track of the input step pulses.

Stepper motor 28BYJ-48 Parameters

Model: 28BYJ-48

Rated voltage: 5VDC

Number of Phase: 4

Speed Variation Ratio: 1/64

Stride Angle: 5.625° /64

Frequency: 100Hz

DC resistance: 50Ω±7%(25°C)

Idle In-traction Frequency: > 600Hz

Idle Out-traction Frequency: > 1000Hz

In-traction Torque >34.3mN.m(120Hz)

Self-positioning Torque >34.3mN.m

Friction torque: 600-1200 gf.cm

Pull in torque: 300 gf.cm

Insulated resistance >10MΩ(500V)

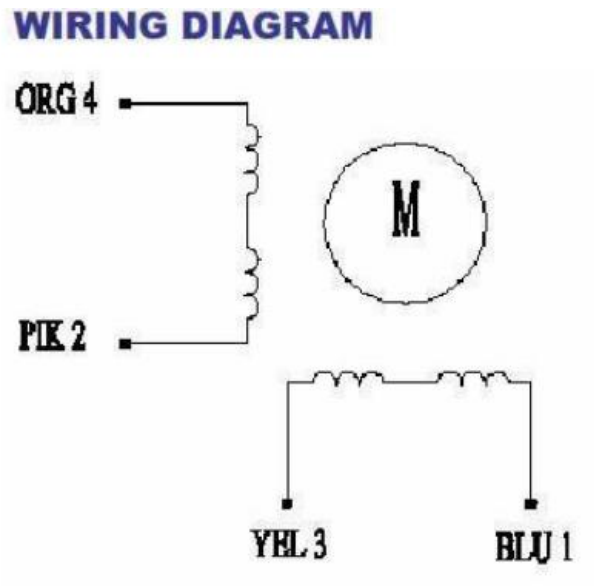
Insulated electricity power：600VAC/1mA/1s

Insulation grade：A

Rise in Temperature <40K(120Hz)

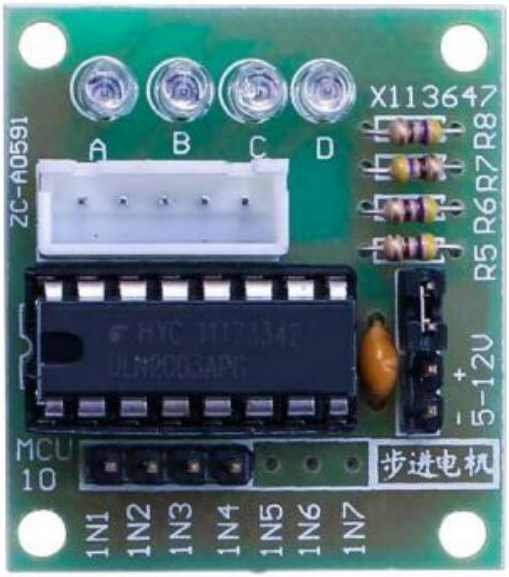
Noise <35dB(120Hz,No load,10cm)

Interfacing circuits



The bipolar stepper motor usually has four wires coming out of it. Unlike unipolar steppers, bipolar steppers have no common center connection. They have two independent sets of coils instead. You can distinguish them from unipolar steppers by measuring the resistance between the wires. You should find two pairs of wires with equal resistance. If you've got the leads of your meter connected to two wires that are not connected (i.e. not attached to the same coil), you should see infinite resistance (or no continuity).

ULN2003 Driver Board



Product Description

o Size: 42mmx30mm

o Use ULN2003 driver chip, 500mA

o A. B. C. D LED indicating the four phase stepper motor working condition.

o White jack is the four phase stepper motor standard jack.

o Power pins are separated

o We kept the rest pins of the ULN2003 chip for your further prototyping.

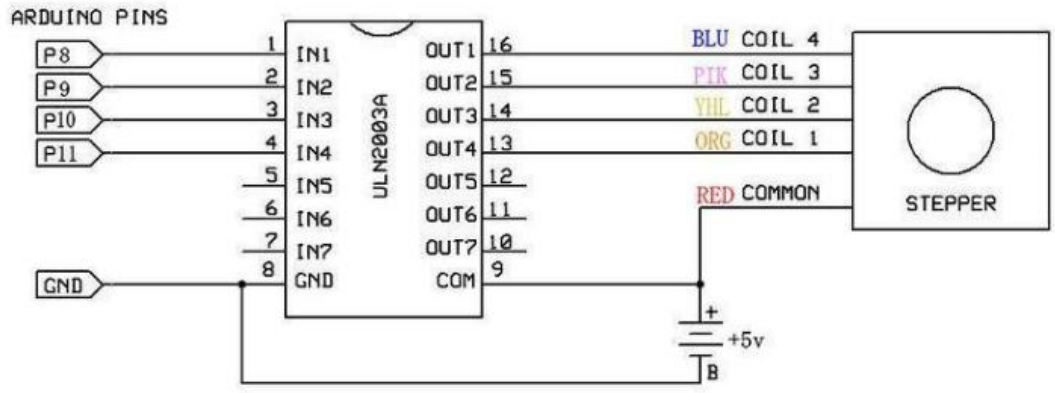
The simplest way of interfacing a unipolar stepper to Arduino is to use a breakout for ULN2003A transistor array chip. The ULN2003A contains seven Darlington transistor drivers and is somewhat like having seven TIP120 transistors all in one package. The ULN2003A can pass up to 500

mA per channel and has an internal voltage drop of about 1V when on. It also contains internal clamp diodes to dissipate voltage spikes when driving inductive loads. To control the stepper, apply voltage to each of the coils in a specific sequence.

The sequence would go like this:

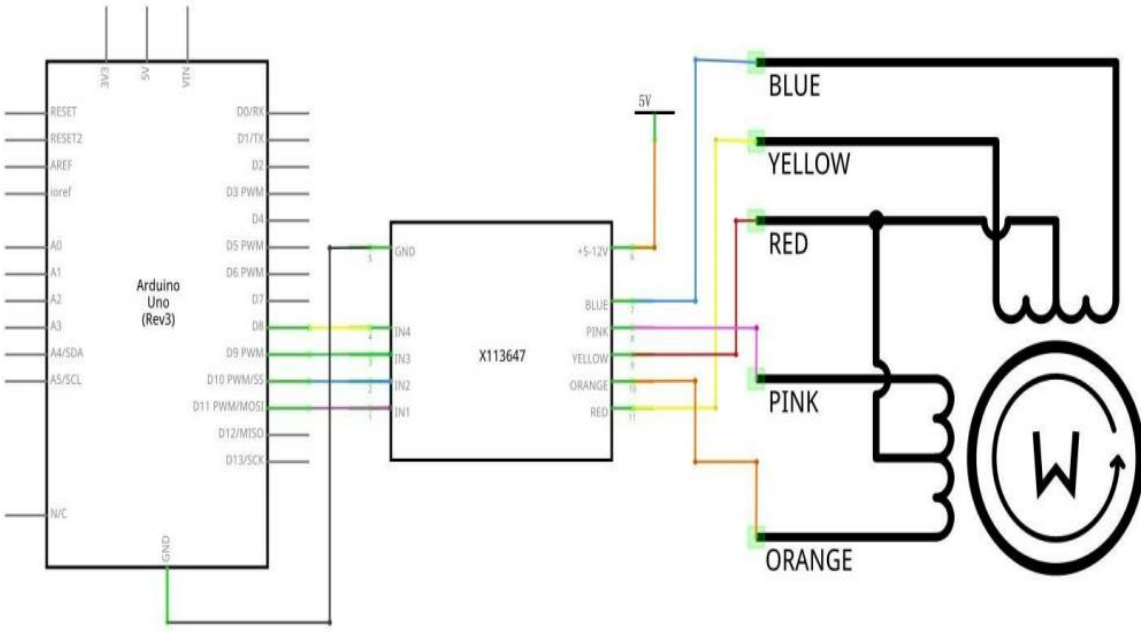
Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Here are schematics showing how to interface a unipolar stepper motor to four controller pins using a ULN2003A, and showing how to interface using four com.

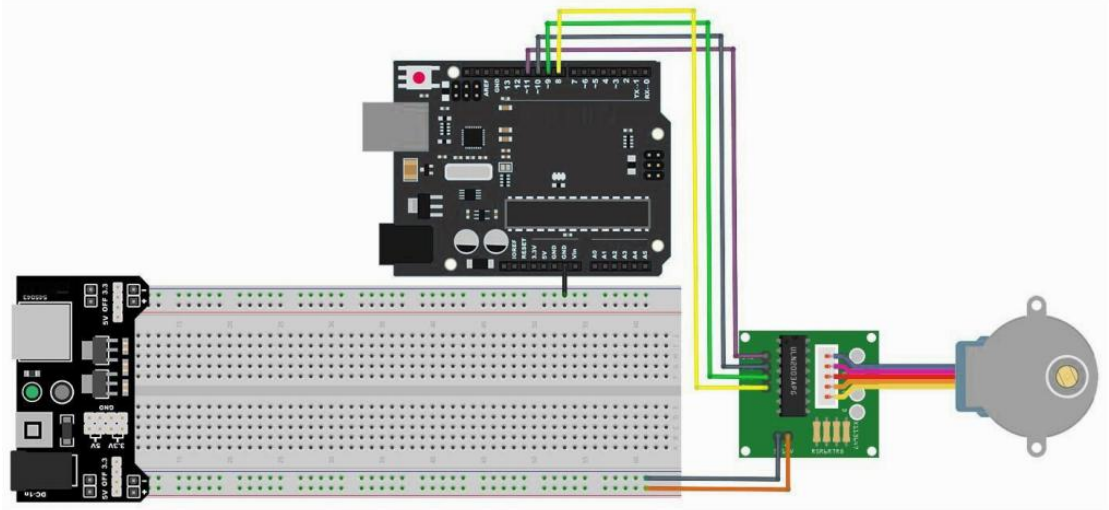


Connection

Schematic



Circuit Connection



We are using 4 pins to control the Stepper.

Pin 8-11 are controlling the Stepper motor.

We connect the Ground from to UNO to the Stepper motor.

Code

After wiring, please open the program in the code folder- Lesson 19 Stepper Motor and click UPLOAD to upload the program.

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

Lesson 20 DHT11 Temperature and Humidity Sensor

Overview

This DHT11 Temperature and Humidity Sensor features calibrated digital signal output with the temperature and humidity sensor complex. Its technology ensures high reliability and excellent long-term stability. Ahigh-performance 8-bit microcontroller is connected. This sensor includes a resistive element and a sense of wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high cost performance advantages.

Component Required

(1) x LONTEN Uno R3

(1) x DHT11 Temperature and Humidity module

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

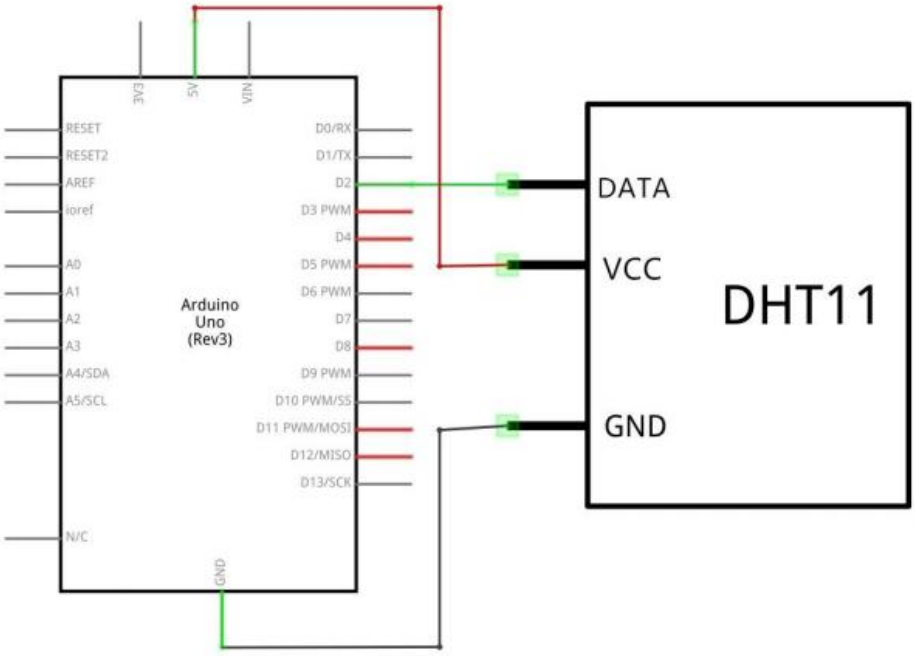
Component Introduction

Temp and humidity sensor:

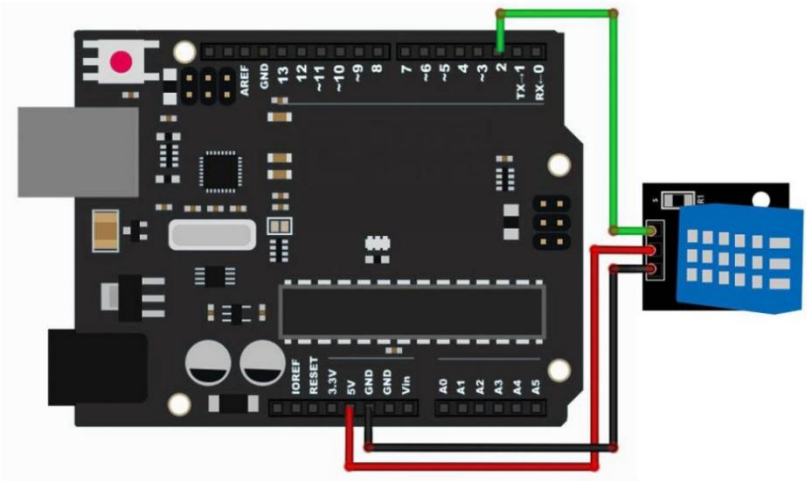
Each DHT11 sensor features extremely accurate calibration data of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, and we should call these calibration coefficients. The single-wire serial interface system is integrated to make it quick and easy. Qualities of small size, low power, and 20-meter signal transmission distance make it a wide applied application and even the most demanding one. Convenient connection, special packages can be provided according to users need.

Connection

Schematic



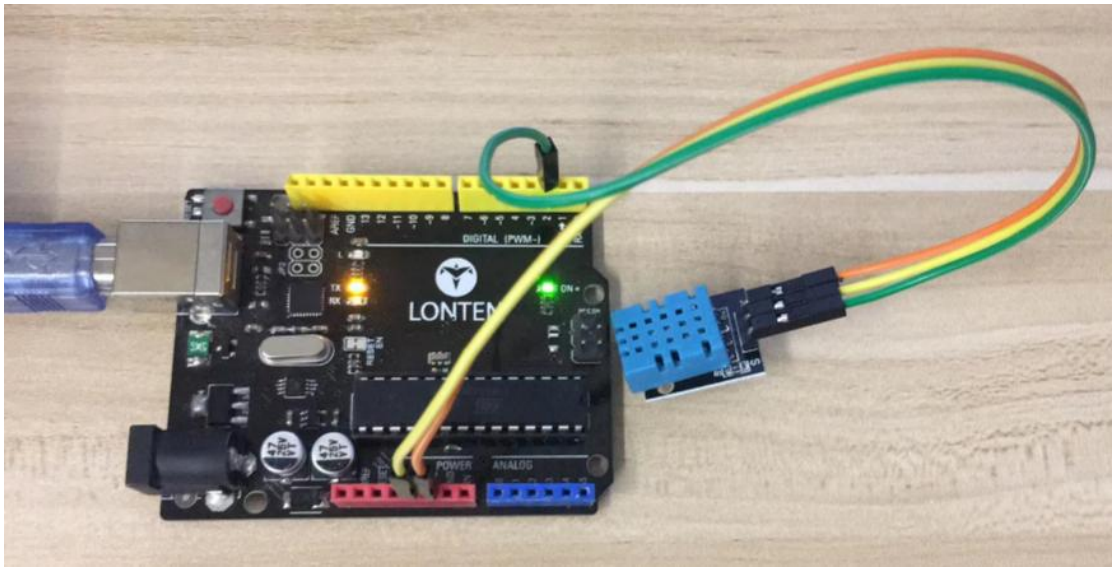
Circuit Connection



As you can see we only need 3 connections to the sensor, since one of the pin is not used.

The connections are: Voltage, Ground and Signal which can be connected to any Pin on our UNO.

Example picture



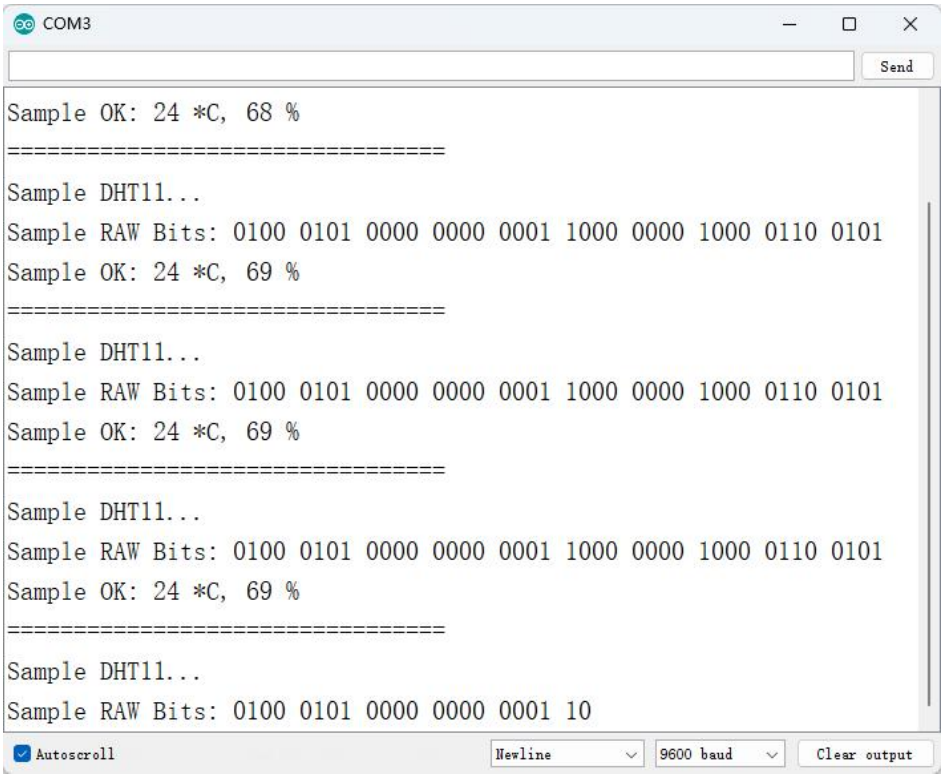
Code

After wiring, please open the program in the code folder- Lesson 20 DHT11 Temperature and Humidity Sensor and click UPLOAD to upload the program.

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

Upload the program then open the monitor, we can see the data as below:

(It shows the temperature of the environment, we can see it is 24 degree)



Lesson 21 Sound Sensor Module

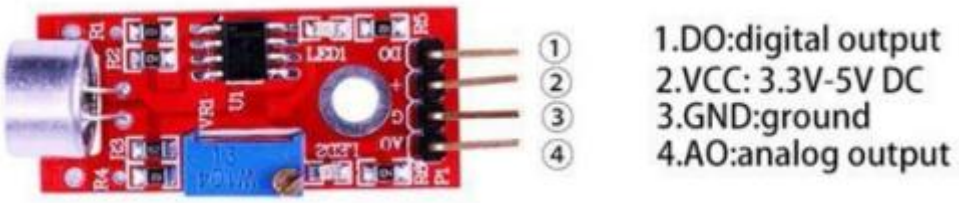
Overview

In this lesson, you will learn how to use a sound sensor module. This module has two outputs:

AO: analog output, real-time output voltage signal of microphone

DO: when the intensity of the sound reaches a certain threshold, the output is a high or low level signal. The threshold sensitivity can be achieved by adjusting the potentiometer.

To make sure the microphone can detect your voice normally, please try to change its sensitivity by turning the blue precise potentiometer on the module.



Component Required

(1) x LONTEN Uno R3

(1) x Sound sensor module

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

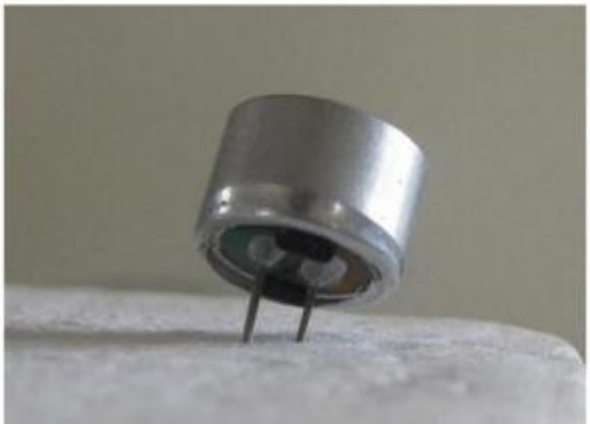
Component Introduction

Ultrasonic sensor

Microphone

Transducers are devices which convert energy from one form to other. A microphone is a transducer which converts sound energy to electrical signals. It works opposite to a speaker. Microphones are available in different shapes and sizes. Depending on the application, a microphone may use different technologies to convert sound to electrical signals. Here, we are going to discuss about the electret condenser microphone which is widely used in mobile phones, laptops, etc.

As the name suggests, the electret condenser microphone is a parallel plate capacitor and works on the principle of a variable capacitance. It consists of two plates, one fixed (called the back plate) and the other moveable (called the diaphragm) with a small gap between them. An electric potential charges the plate. When sound strikes the diaphragm it starts moving, thereby changing the capacitance between the plates which in turn results in a variable electric current to flow.

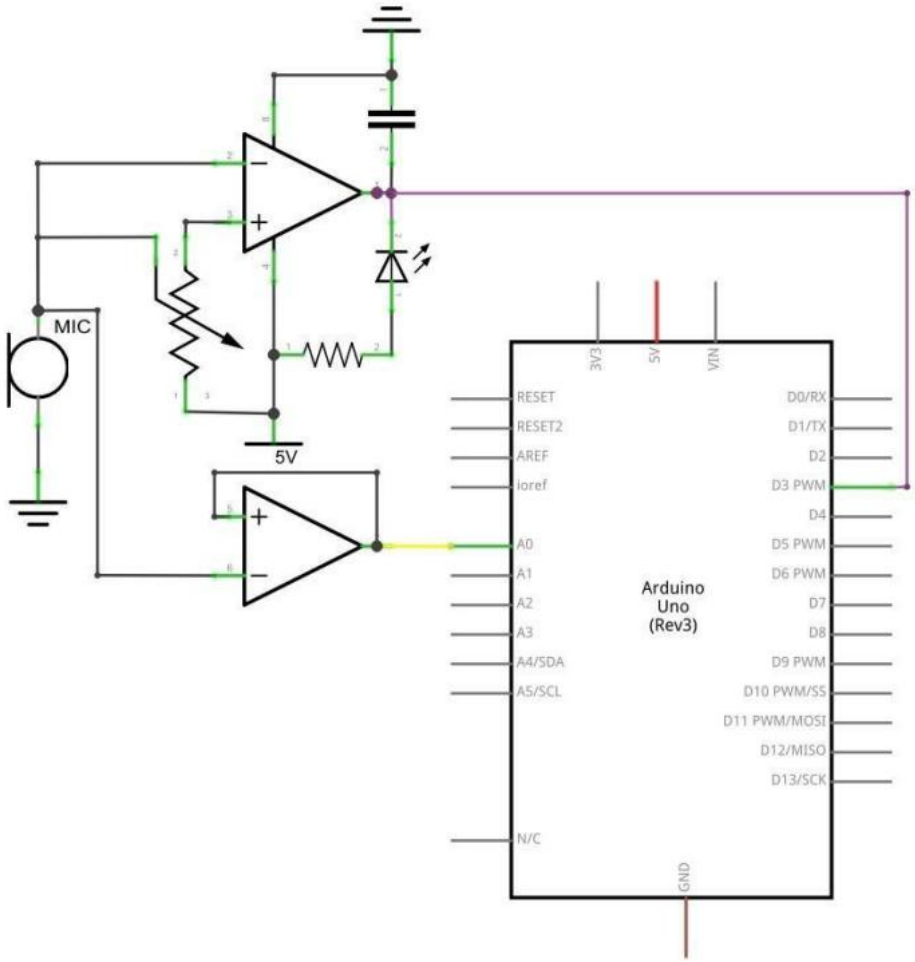


These microphones are widely used in electronic circuits to detect minor sounds or air vibrations which in turn are converted to electrical signals for further use. The two legs as shown in the image above are used to make electrical connection with the circuit.

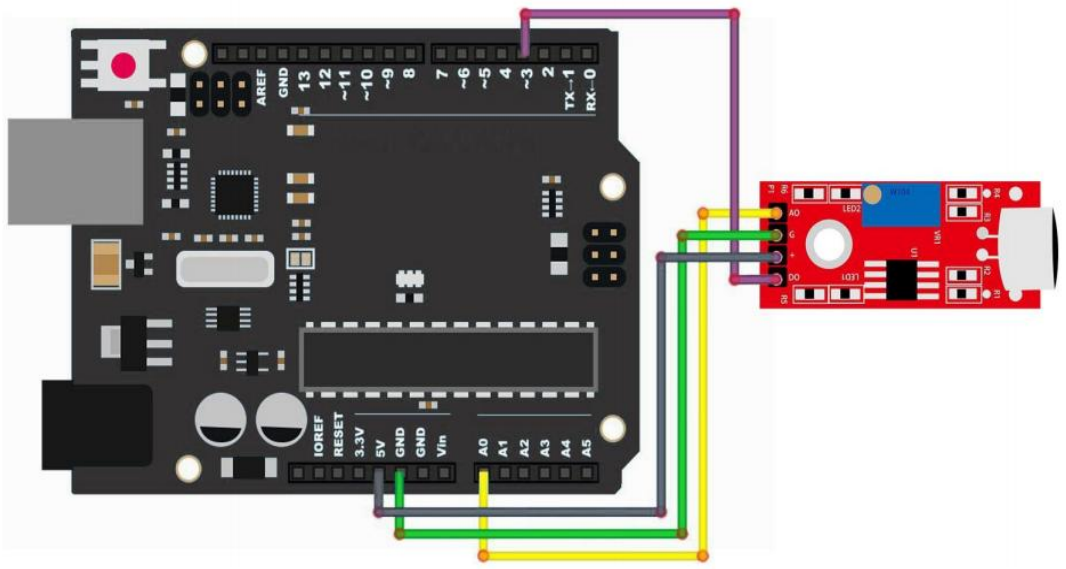
A solid conducting metal body encapsulates the various parts of the microphone. The top face is covered with a porous material with the help of glue. It acts as a filter for the dust particles. The sound signals/air vibrations passes through the porous material and falls on the diaphragm through the hole shown in the image above.

Connection

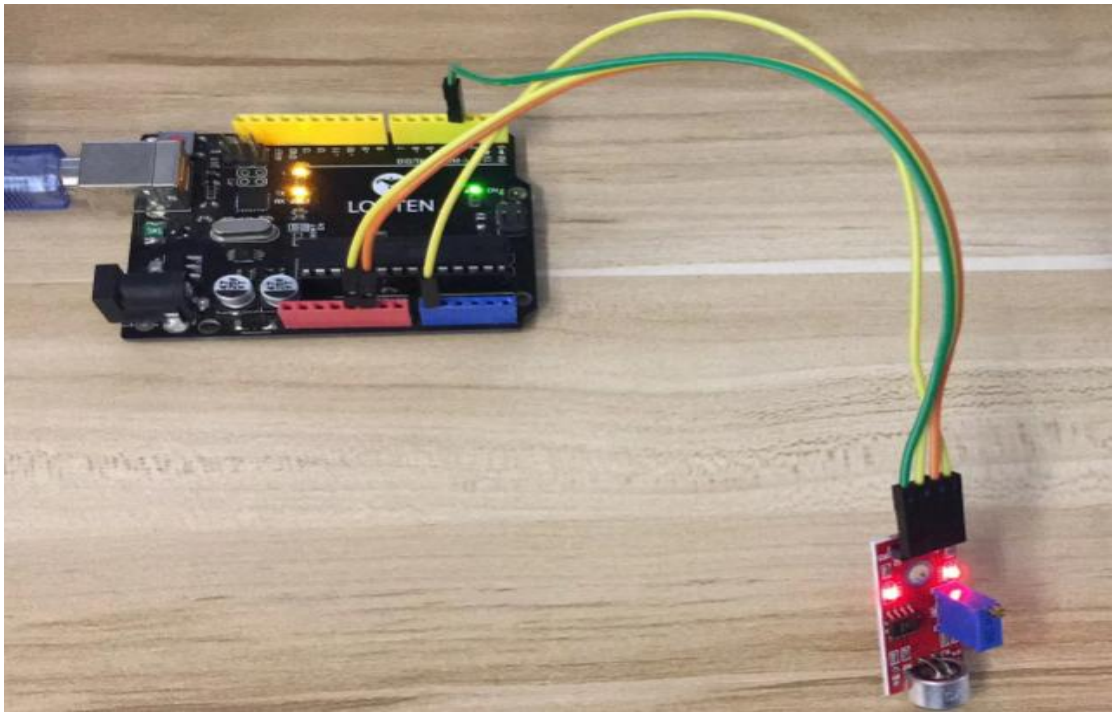
Schematic



Circuit Connection



Example picture



Code

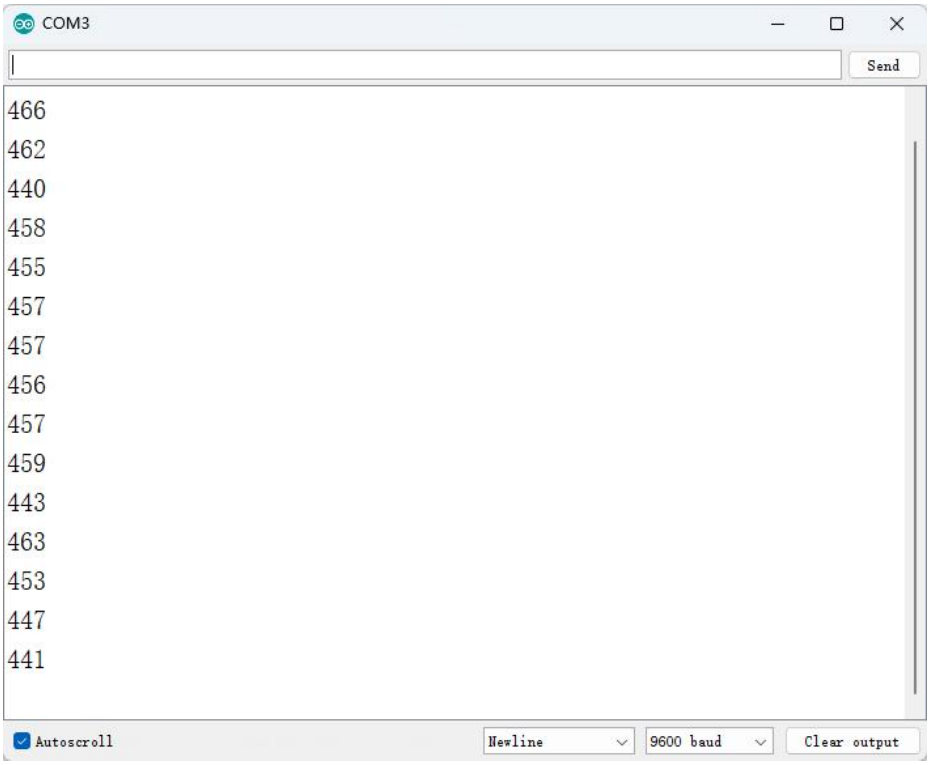
After wiring, please open the program in the code folder- Lesson 21 Sound Sensor Module and click UPLOAD to upload the program.

This module provides two signal output modes, for which we wrote two codes:

digital_signal_output and analog_signal_output. The code of digital_signal_output works when the voice reaches a certain value, it will trigger a digital signal and the dig #11 pin on Arduino will output a high level and the indicator L will be lit up at the same time. This

triggering value may be changed according to the sensitivity adjustment method mentioned above. The code of analog_signal_output will read the analog value of the module and directly display it on the serial monitor, likewise, this value can also be changed according to the sensitivity adjustment method mentioned above.

Open the monitor then you can see the data as below:



Lesson 22 RGB Module

Overview

In this experiment, we will learn how to use RGB module.

RGB module are made from a patch of full-color LED. By adjusting the voltage input of R, G, B pins, we can adjust the strength of the three primary colors (red/blue/green) so to implementation result of full color effect.

Component Required:

(1) x LONTEN UNO Board

(1) x USB Cable

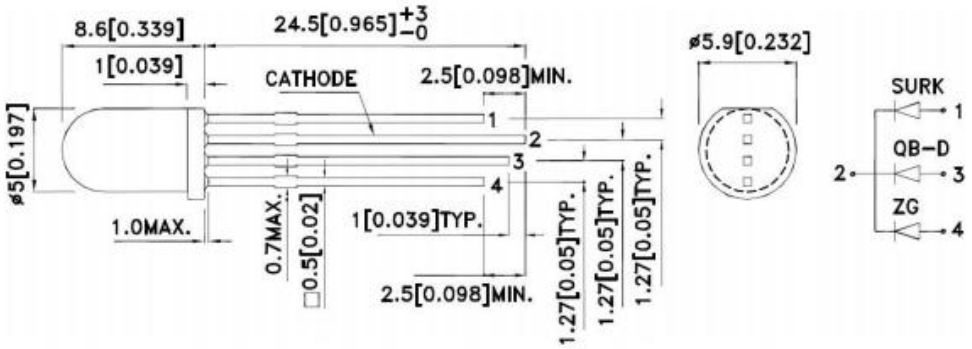
(1) x RGB module

(1) x Breadboard

(4) x Breadboard Jumper Wires

Component Introduction

RGB:



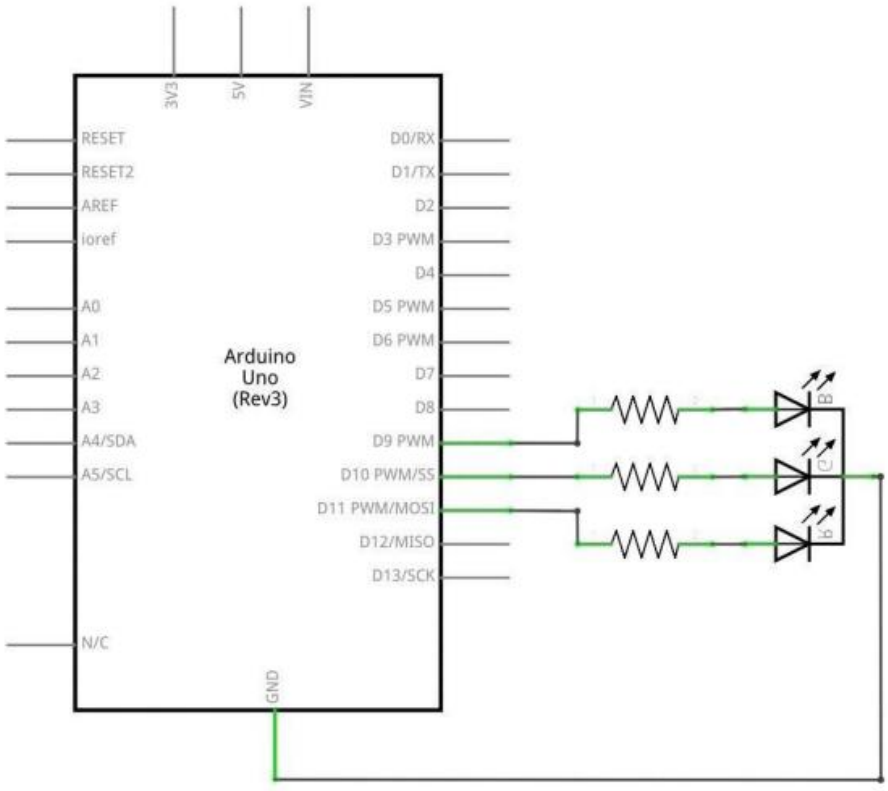
Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Hyper Red Blue Green	650 468 515		nm	IF=20mA
λD [1]	Dominant Wavelength	Hyper Red Blue Green	630 470 525		nm	IF=20mA
Δλ1/2	Spectral Line Half-width	Hyper Red Blue Green	28 25 30		nm	IF=20mA
C	Capacitance	Hyper Red Blue Green	35 100 45		pF	VF=0V;f=1MHz
VF [2]	Forward Voltage	Hyper Red Blue Green	1.95 3.3 3.3	2.5 4 4.1	V	IF=20mA
IR	Reverse Current	Hyper Red Blue Green		10 50 50	uA	VR=5V

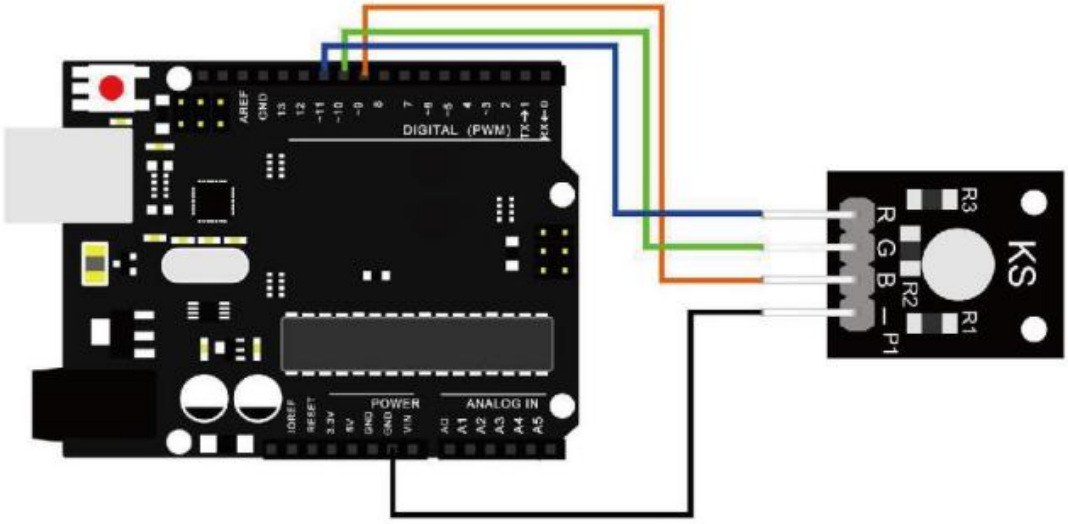
Notes:
1.Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

Connection

Schematic



wiring diagram



Result

After we connect the circuit as the picture, we upload the program of each module. We can see the module changing their color as the code set. If you want to make it change the color in different way, you can revise the code.

Lesson 23 DS1302 Clock Sensor

Introduction

DS1302, a trickle charge clock chip rolled out by DALLAS, is inclusive of a real-time clock/calendar and 31 bytes of static RAM. It communicates with the microcontroller through the serial interface.

The real-time clock/calendar circuit provides information about seconds, minutes, hours, days, weeks, months, and years. In addition, the number of days in a month and ones of a leap year can be adjusted automatically.

The clock operation uses 24 or 12 hour format through AM/PM indication.

The DS1302 can communicate with the single-chip microcomputer through the synchronous serial way and only three ports needed which are RST reset , I/O data line and SCLK serial clock.

The read and write data of the clock/RAM is communicated in a byte or over 31 bytes. Its power consumption is very low when working, thereby, the power is less than 1mW when keeping data and clock information.

Meanwhile, a positioning hole of the module contributes to fix it on the other devices.

Specification:

Working voltage: DC 5V

Working current: 60mA

Maximum power: 0.3A

Working temperature: -25℃-65℃

Interface: 5pin header interface with 2.54mm pitch

Positioning hole: 3mm in diameter

Size: 50*28mm

Weight: 5.8g

The backup battery is CR1220 and non-rechargeable

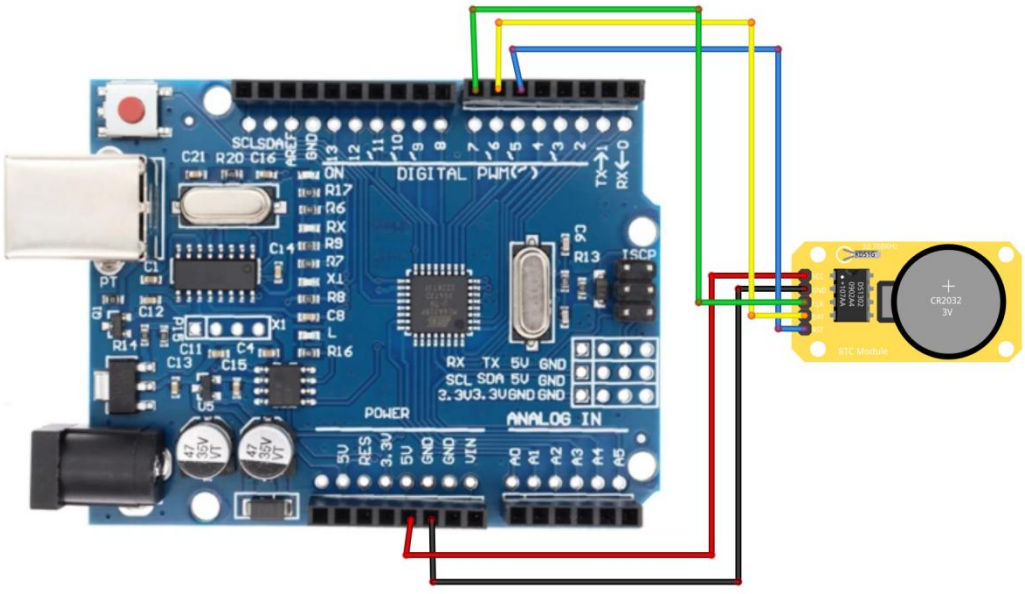
Voltage:3V

Current: 40mAh

The theoretical data can be retained for more than 1 year

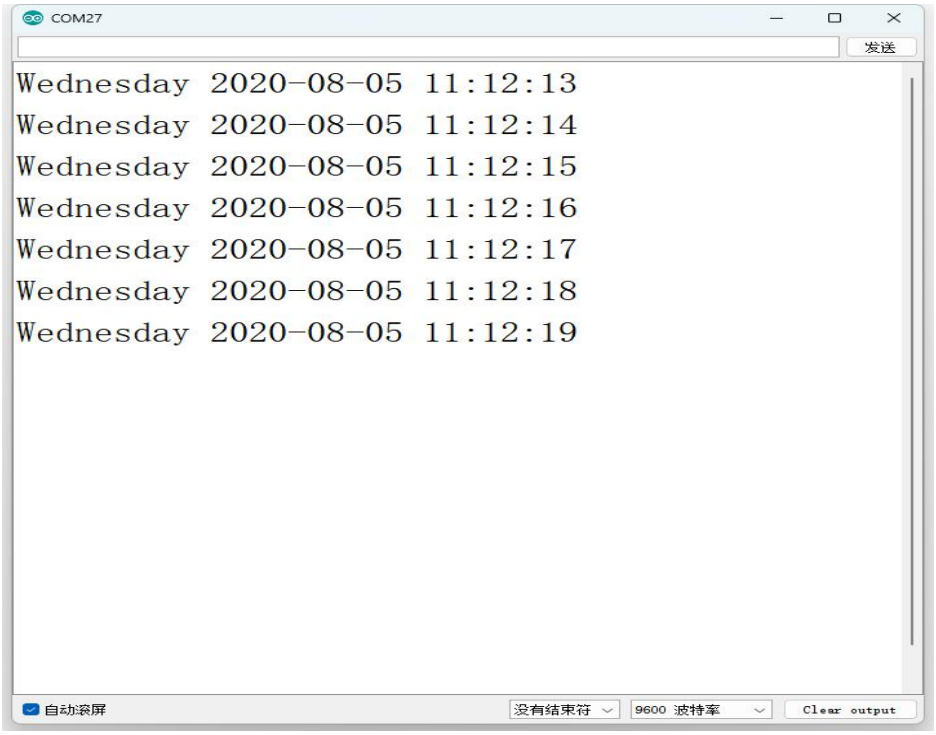
Attention:The DS1302 Clock Sensor in this kit does not include the CR1220 battery, which needs to be purchased locally.

wiring diagram



Test Result

Wire up the components according to the above figure, burn the code and power on. The current time and date are displayed on the serial monitor, as shown below.



Lesson 24 4*4 Button Module

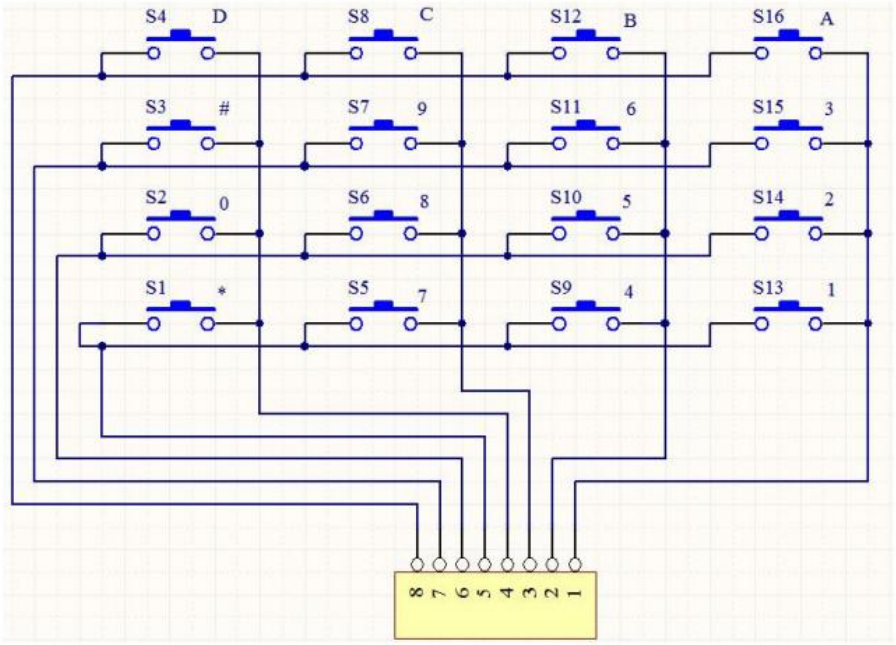
Introduction

In the application system of microcontroller, keyboard is essential in man-machine dialogue. When you are short of a button, you can connect one to the I/O port of the comtroller; but when you need a lot of buttons with limited I/O port resources, this 4*4 Matrix Keypad is no doubt your best choice.

4*4 matrix keypad is the most applied keypad form. We need to master its keypad identification technology as entry to microcontroller world.

Here, we will use an examples to illustrate the identification method of 4*4 matrix keypad. The key layout is in matrix form, so with only eight I/O ports, we can identify 16 buttons, saving lots of I/O port resources.

Pin layout for 4*4 Large Button module:



Hardware Required

(1) x LONTEN UNO Board

(1) x USB cable

(1) x 4*4 Button Module

(1) x Active Buzzer

(1) x Red M5 LED

(1) x 220Ω resistor

(1) x Breadboard

(12) x Breadboard jumper wire

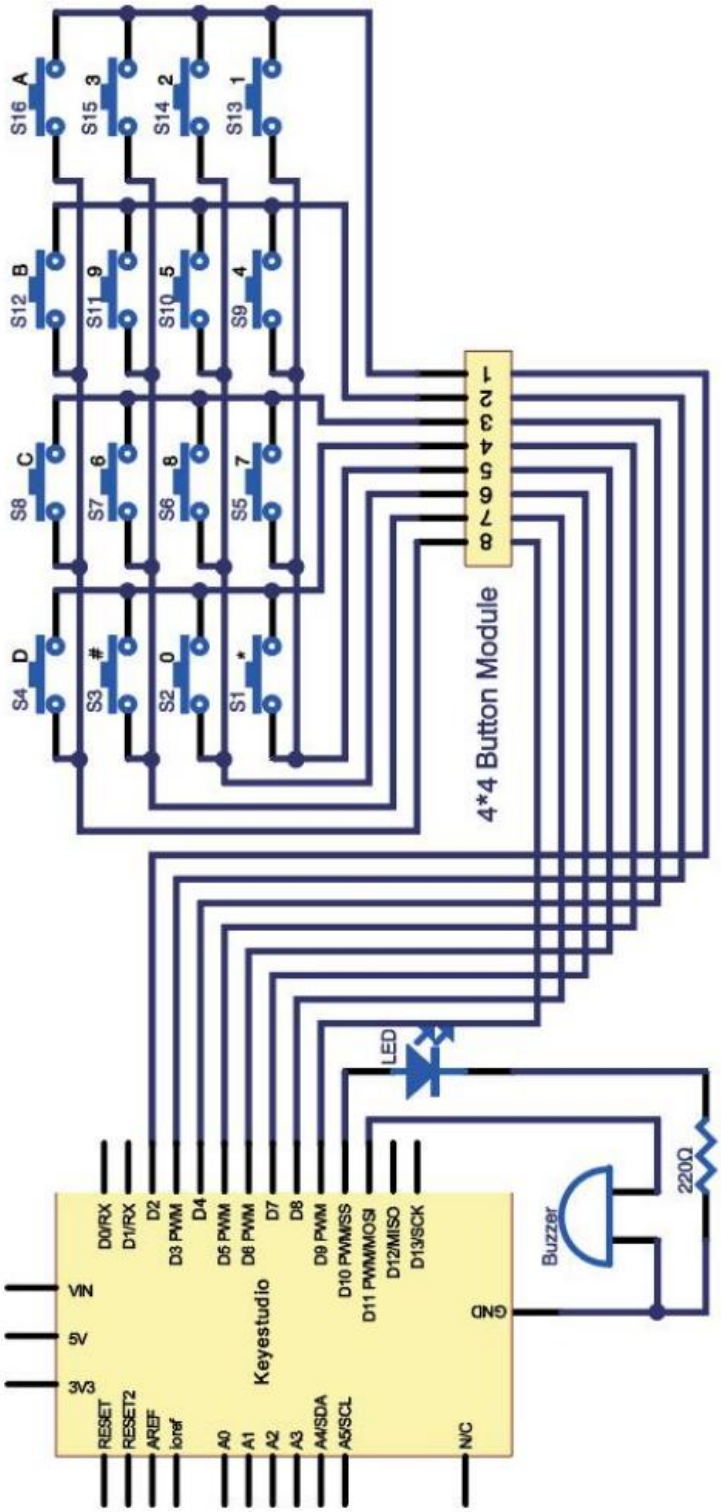
4*4 Button Module

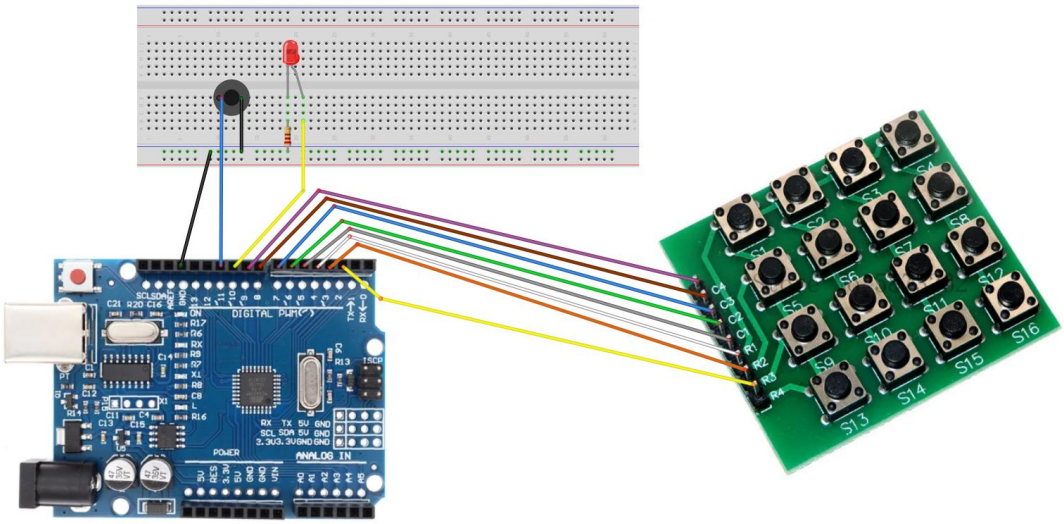
Working voltage: 3.3-5V

Rated power: 50mA.12V DC

Working temperature: -30℃to +85℃

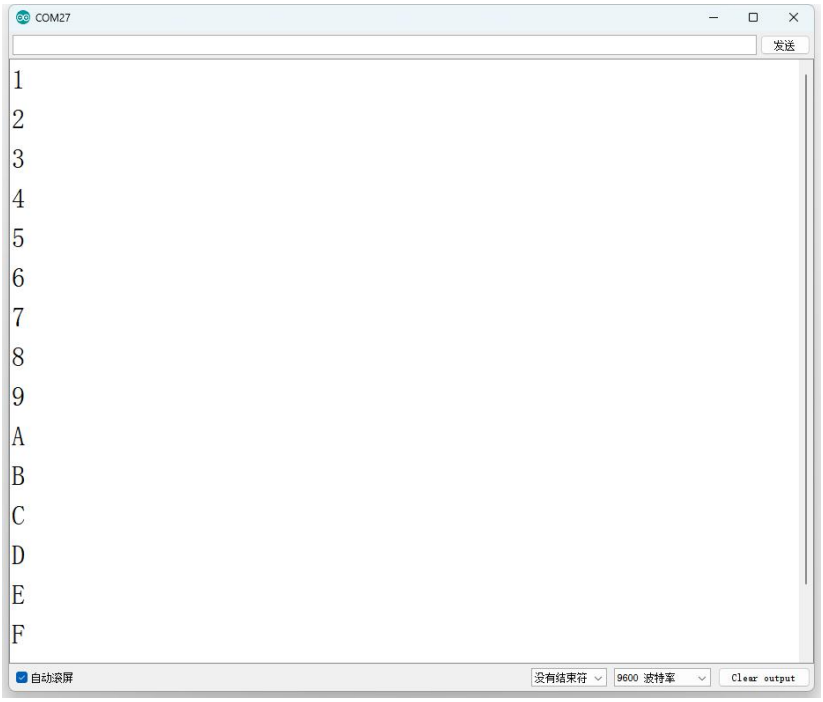
Connection Diagram


S16 A
S15 3
S14 2
S13 1
S12 B
S11 9
S10 5
S9 4
S8 C
S7 6
S6 8
S5 7
S4 D
S3 #
S2 0
S1 *
1
2
3
4
5
6
7
8
4*4 Button Module
LED
220Ω
Buzzer
D0/RX
D1/RX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
VIN
5V
3V3
RESET
RESET2
AREF
Ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
Keyestudio



Test Result

Upload the program to the board, open serial monitor; press certain button on the module, it will display corresponding value as below picture shown:



When serial monitor prints 1, LED lights up; when it printing 2, the buzzer rings 1S.