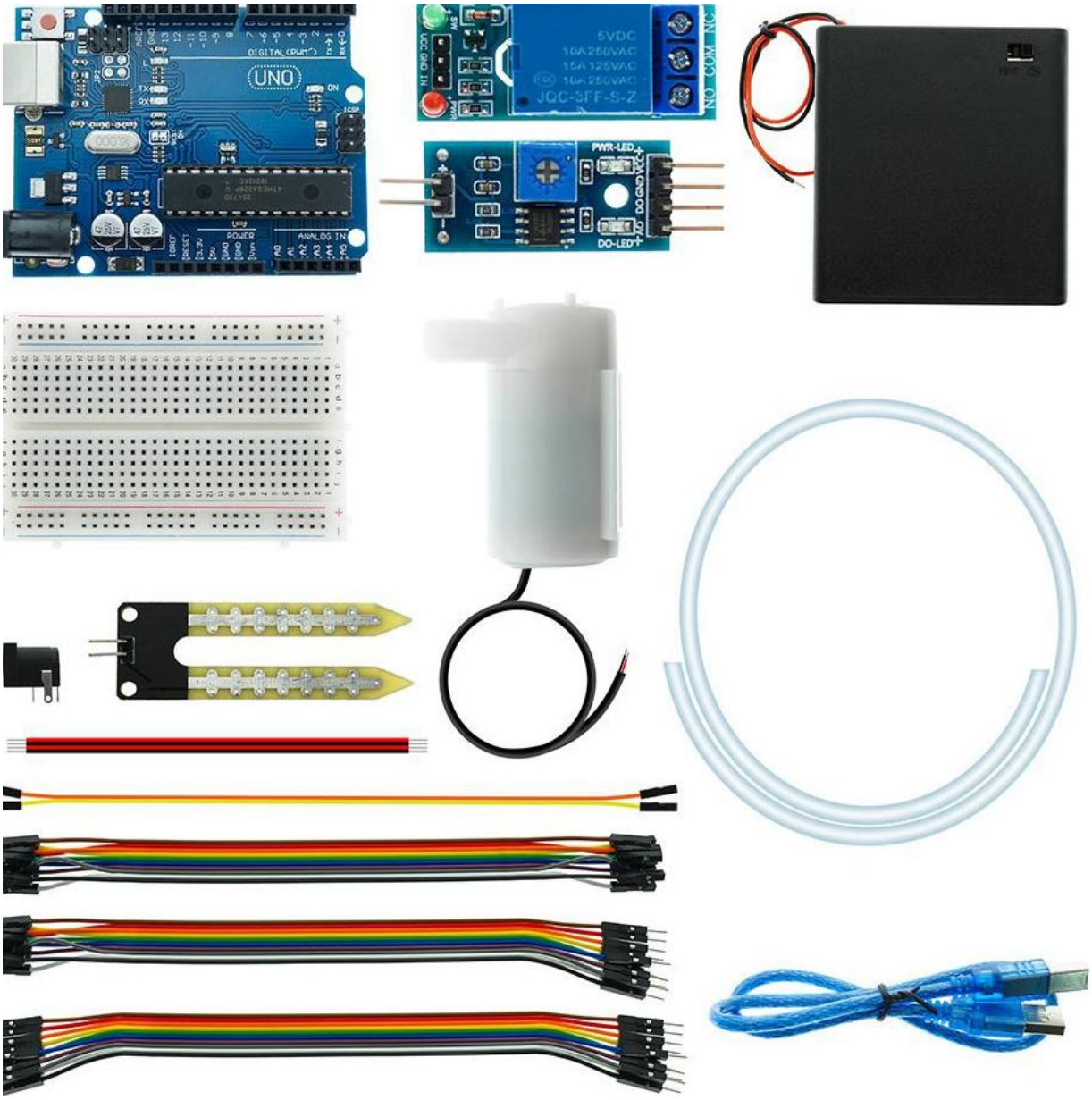
Based on UNO R3 watering system kit tutorial

V1.0.20.5.27

Product List



Preface

The kit is based on the popular open source electronic platform Arduino. You can share and exchange your

Experience and design ideas with fans all over the world. All components in the kit are individually packaged, including all the electronic components required for these. At the same time, there are detailed assembly and debugging instructions in this tutorial. To ensure that you can assemble and run smoothly during the learning process, if you encounter difficult technical problems, you can contact us at any time, and we will provide you with high-quality technical support services for free as soon as possible.

The content of this tutorial can ensure that novices without any professional skills can accomplish the goal. in case

If you are interested in Arduino and want to learn how to program and build circuits, please visit our

Store or contact us to purchase an Arduino learning starter kit specially prepared for beginners.

Content

Lesson 1 Arduino

Arduino

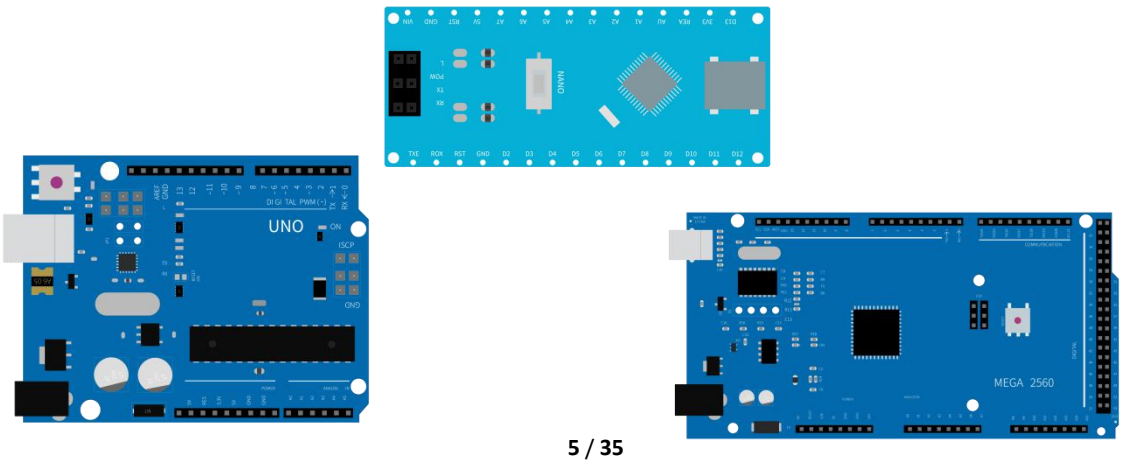
Arduino is an open source electronic platform based on easy-to-use hardware and software.

Suitable for anyone working on interactive projects. Generally, an Arduino project consists of circuits and codes.

Arduino board

The Arduino board is a circuit board that integrates a microcontroller, input and output interfaces, etc. The Arduino board can use sensors to sense the environment and receive user operations to control LEDs, motor rotation, and so on. We just need to assemble the circuit and write the code.

Currently, there are several types of Arduino development boards, and the codes between different types of development boards are common (due to different hardware, some development boards may not be fully compatible). Popular main control boards include:



Lesson 2 Installing IDE

Introduction

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

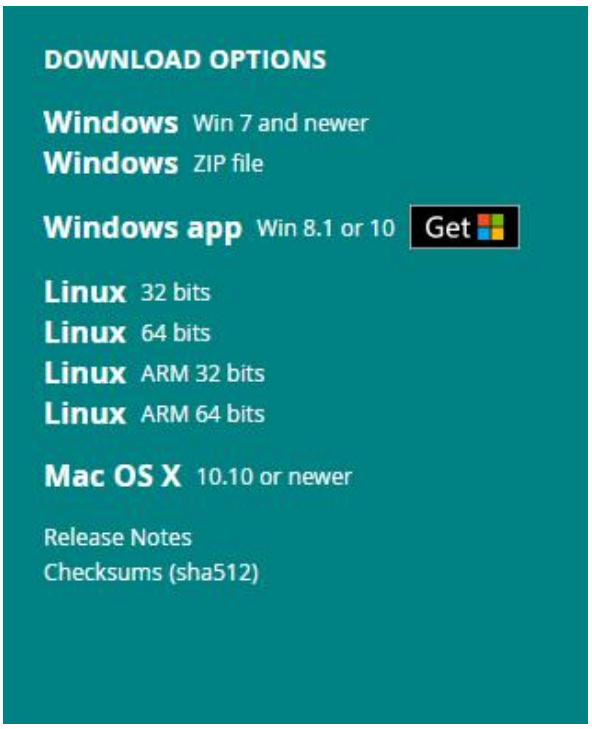
In this lesson, you will learn how to setup your computer to use Arduino and how to set about the lessons that follow.

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/Main/Software and find below page.

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.

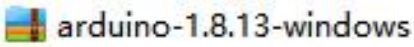


Click Windows Installer.

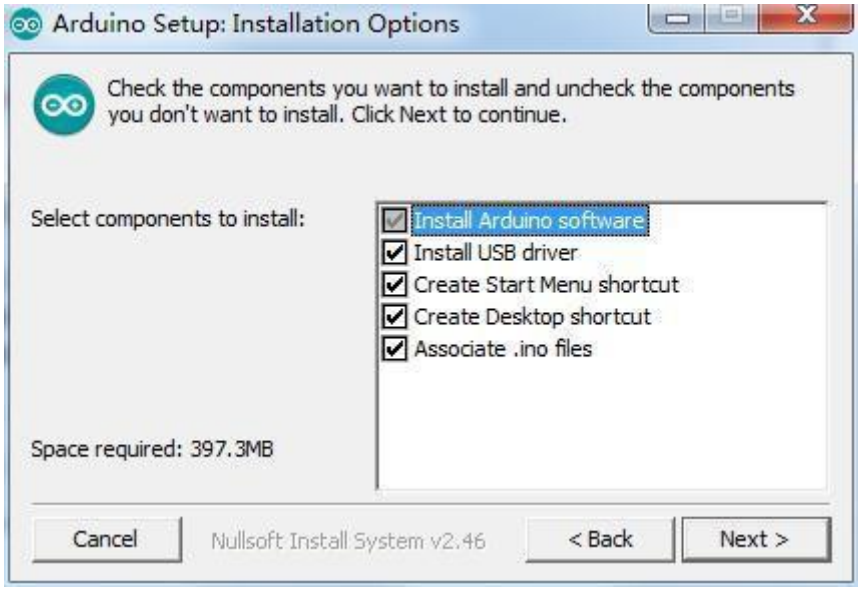
Click JUST DOWNLOAD.

Also version 1.8.13is available in the material we provided, and the versions of our materials are the latest versions when this course was made.

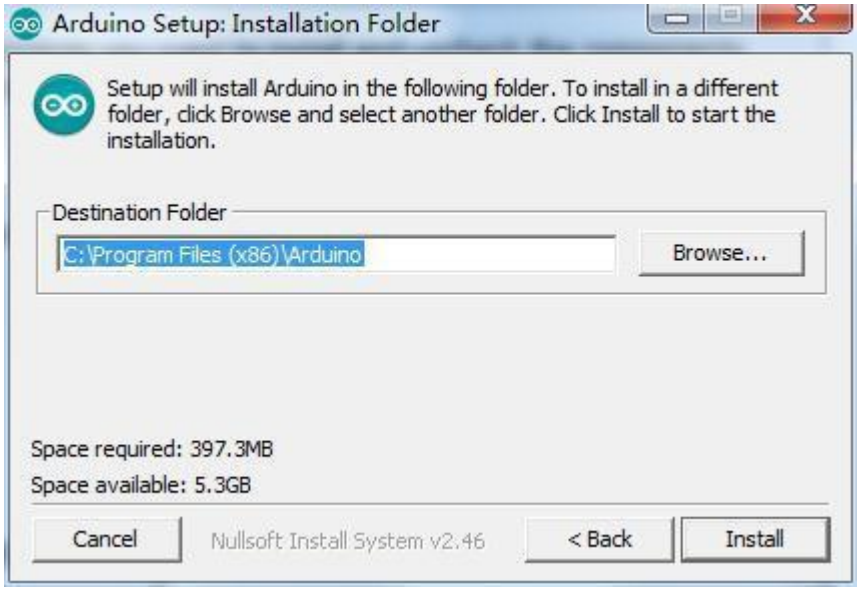
Installing Arduino (Windows)



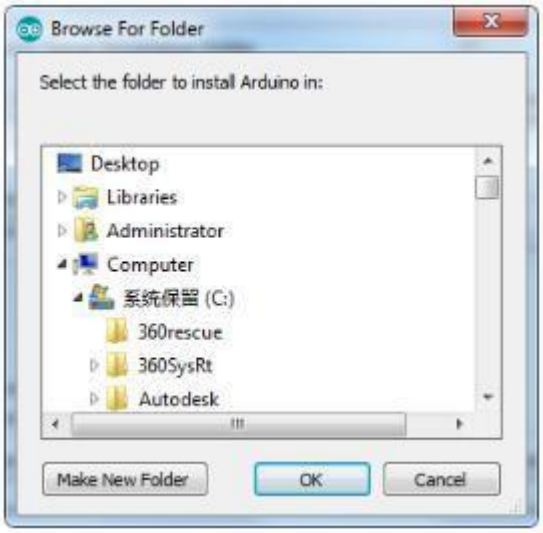
Click I Agree to see the following interface



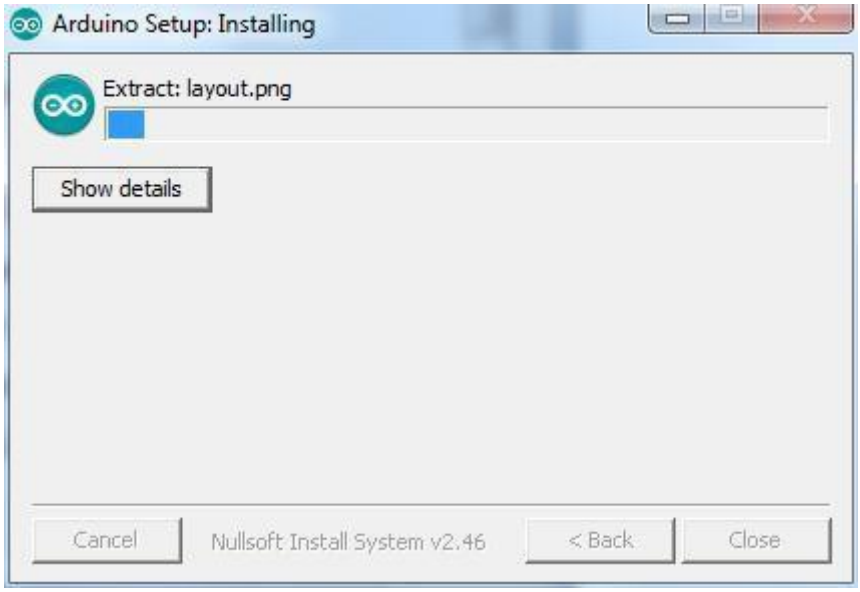
Click Next



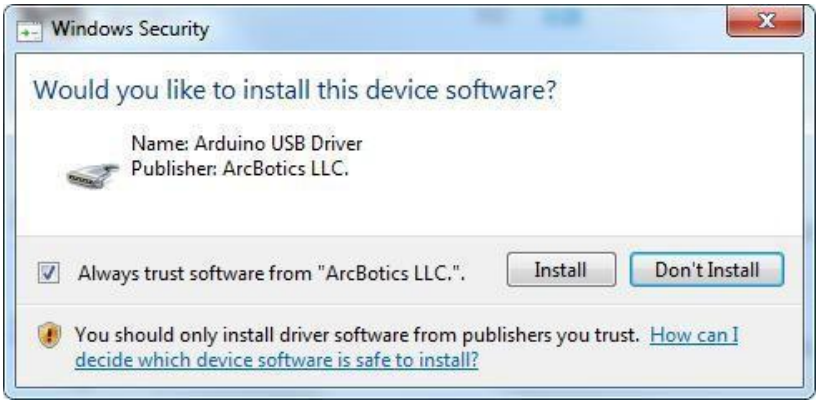
You can press Browse··· to choose an installation path or directly type in the directory you want.



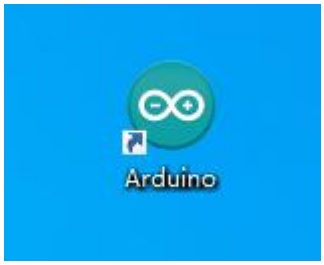
Click Install to initiate installation



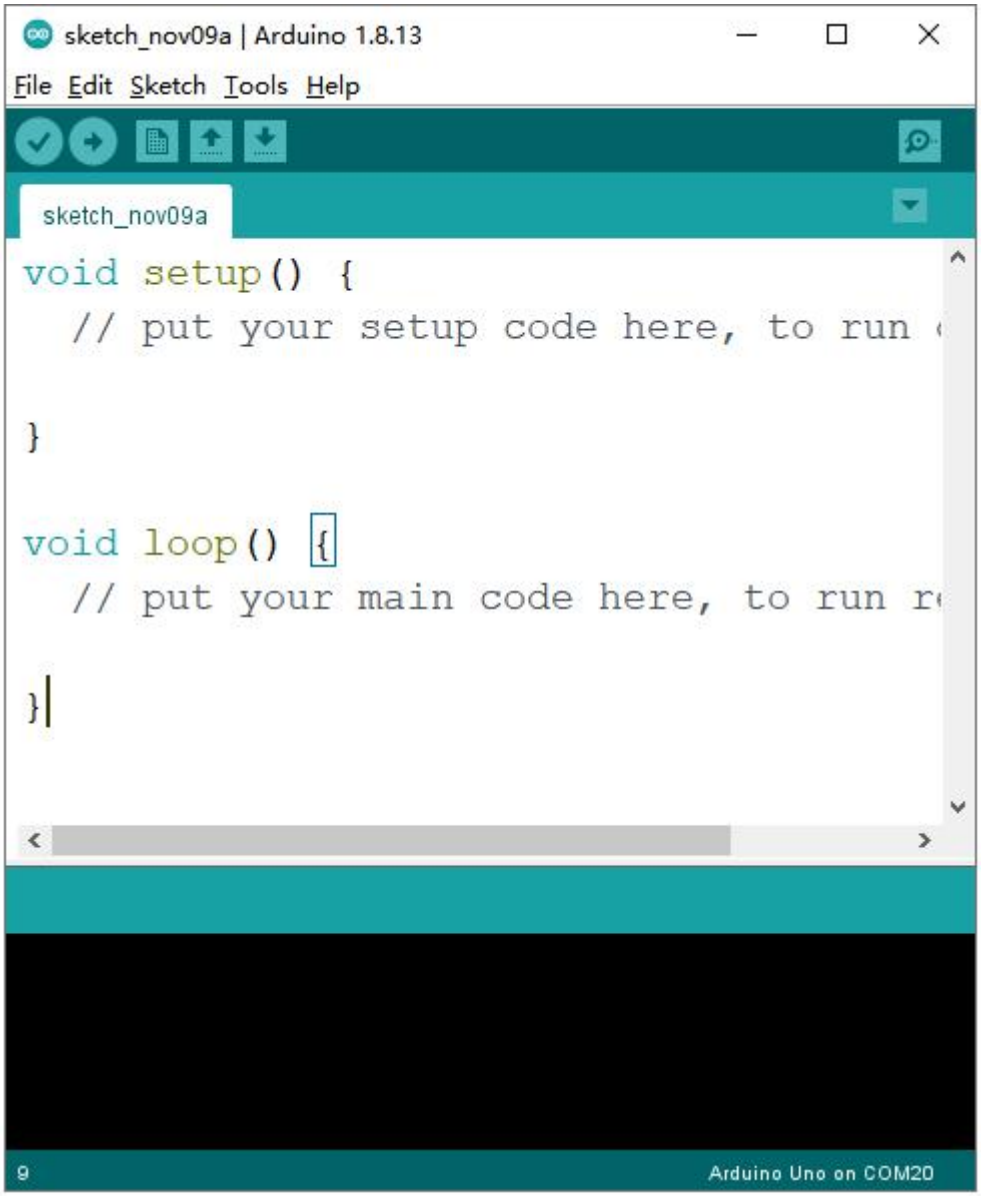
Finally, the following interface appears, click Install to finish the installation.



Next, the following icon appears on the desktop



Double-click to enter the desired development environment



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.

Lesson 3 Add Libraries

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.

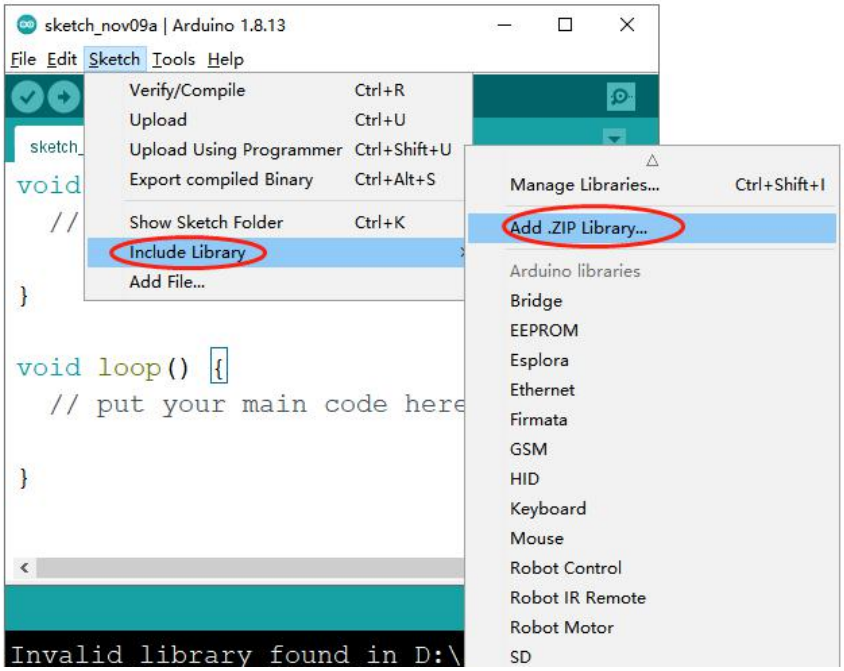
What are Libraries?

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 LiquidCrystal 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.

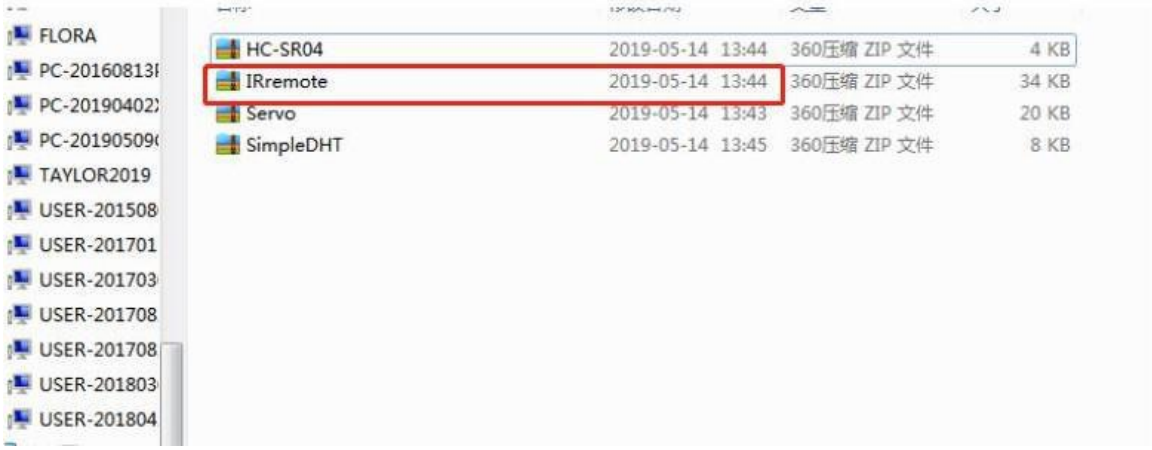
How to Install a Library

Using the Library Manager

To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.9). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



Then we check to see if the library is installed correctly.



Example: IRromote

Open arduino software - project - load library - add a .zip library

Add method two:

Copy the library folder to the Libraries folder in the Arduino installation directory. Restart Arduino and the added library will take effect.

Lesson 4 Blink

Overview:

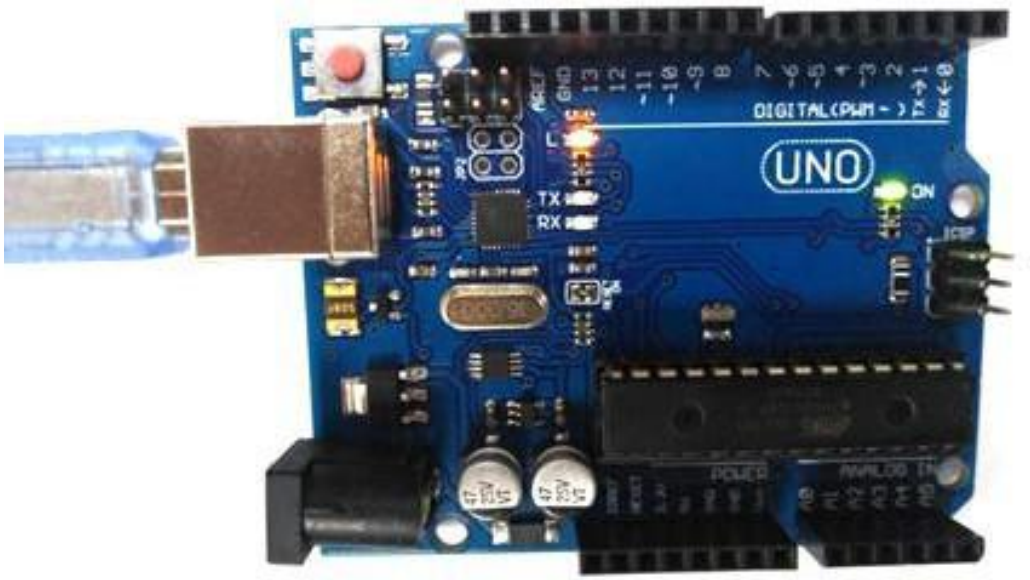
In this lesson, you will learn how to program your UNO R3 controller board to blink the Arduino’s built-in LED, and how to download programs by basic steps.

Component Required:

1x UNO R3

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



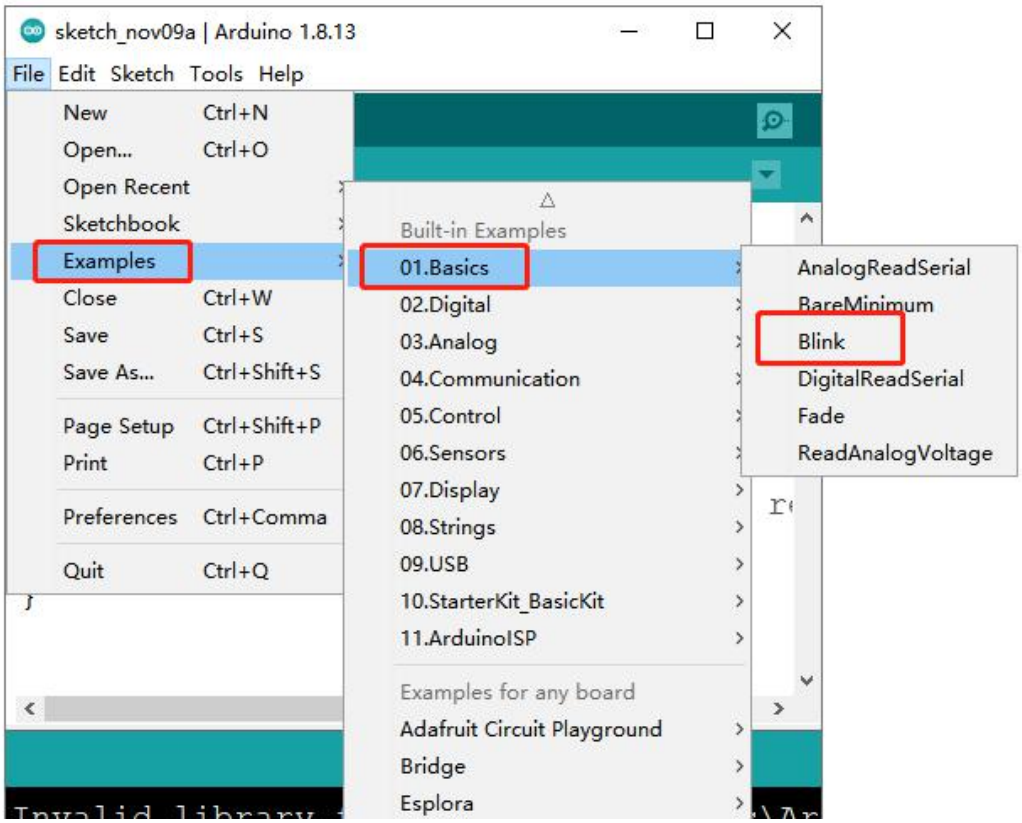
You may find that your UNO R3 board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

In this lesson, we will reprogram the UNO R3 board with our own Blink sketch and then change the rate at which it blinks.

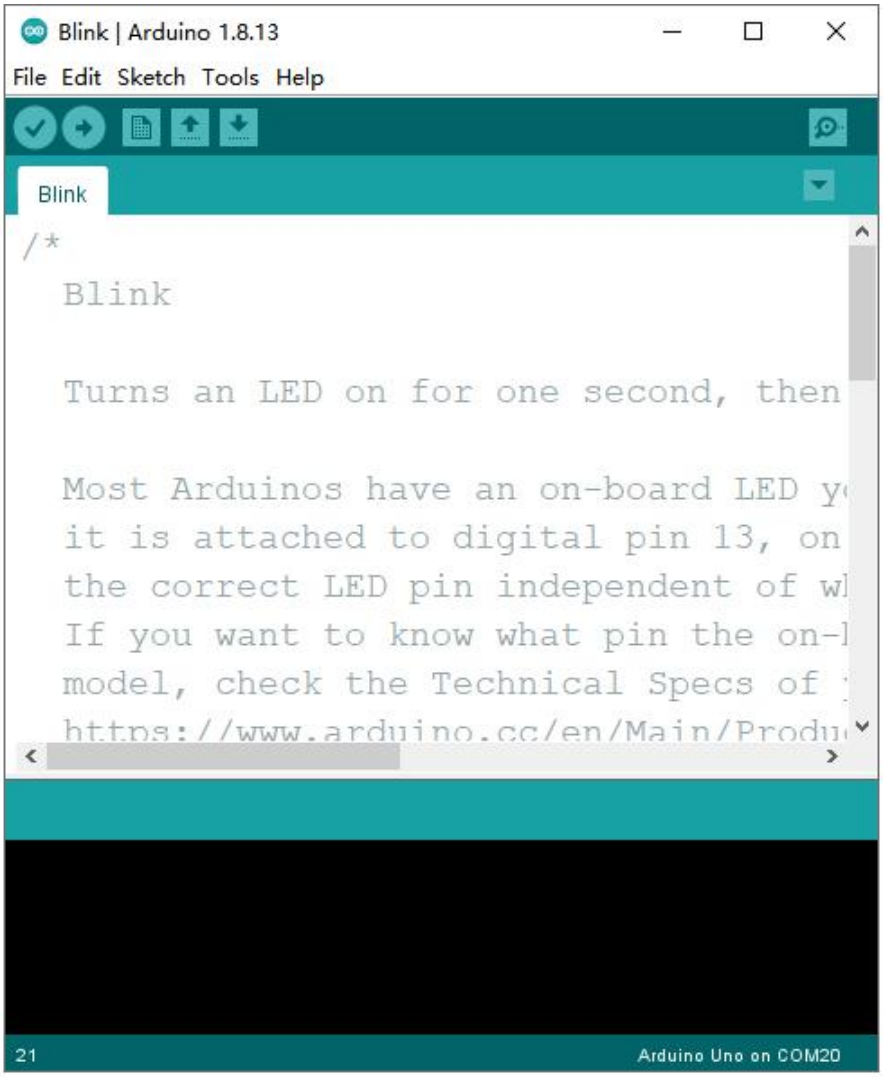
In Lesson 1, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO R3 board. The time has now come to put that connection to the test and program your UNO R3 board.

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.

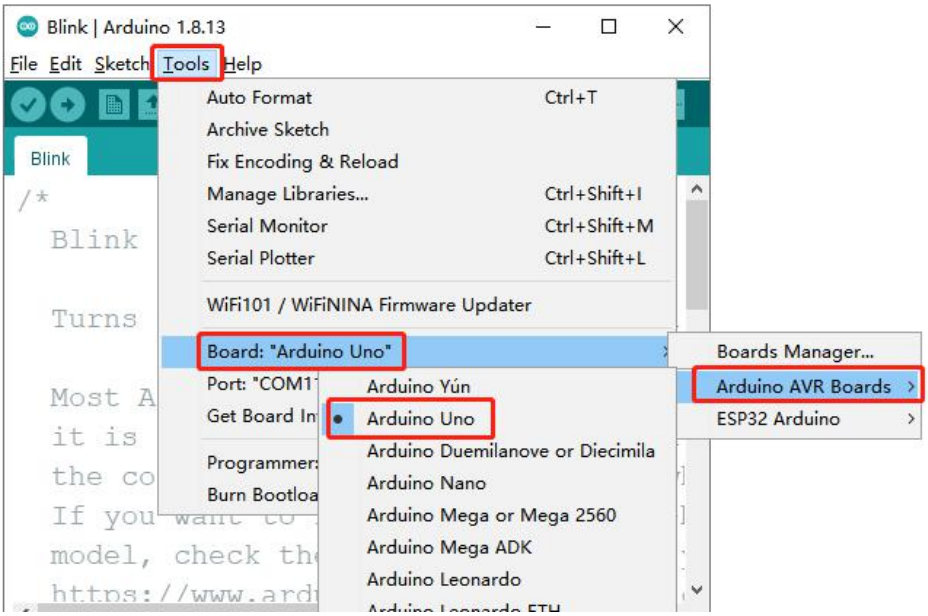
Load the 'Blink' sketch that you will find in the IDE's menu system under File > Examples > 01.Basics



Open as shown below:

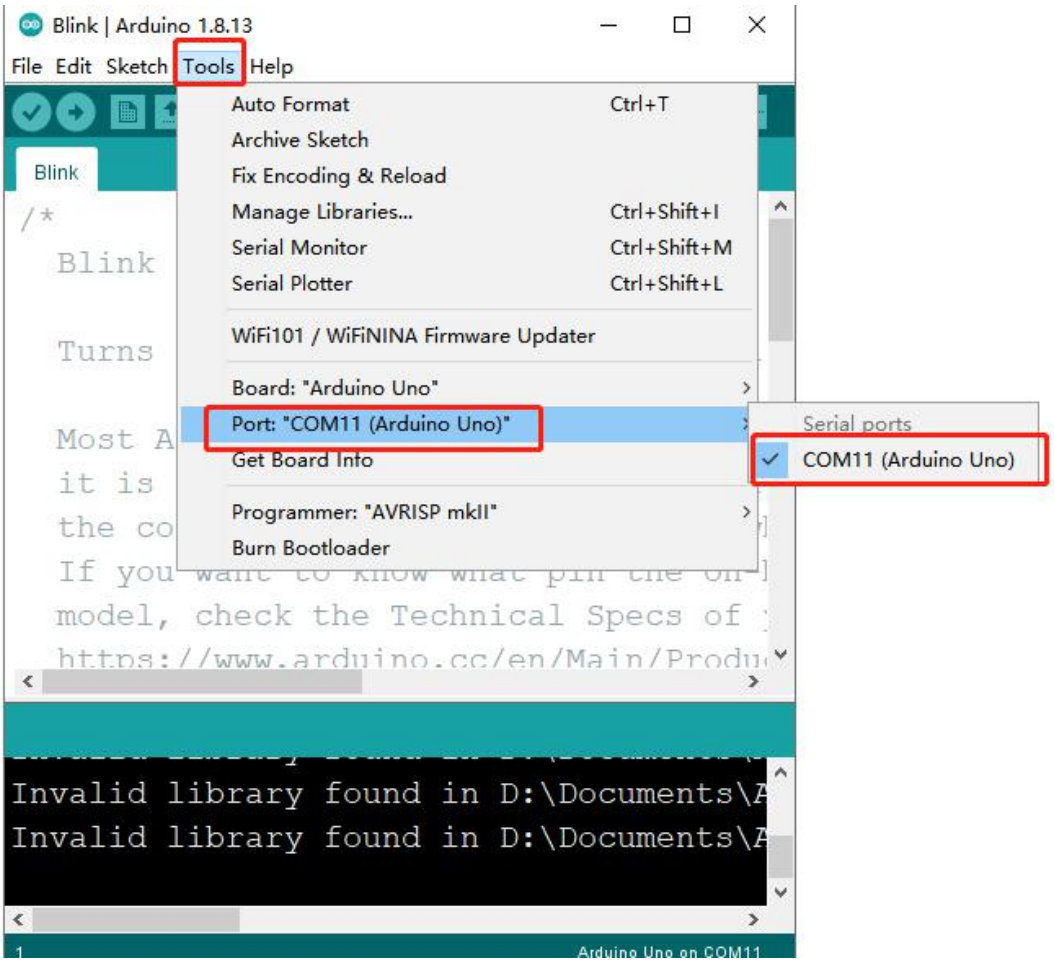


Upload code needs to select the board model Arduino/Genuino Uno



The upload code also needs to select the board port number. Our computer displays the port number of arduino as: "COM11 Arduio / Genuino Uno", you may be the other serial port number.

Note: A correct COM port should be COMX (arduino) XXX), which is certified by the standard.



Note that a huge part of this sketch is composed of comments.

These are not actual program instructions; rather, they just explain

how the program works. They are there for your benefit.

Everything between /* and */ at the top of the sketch is a block

comment; it explains what the sketch is for.

Single line comments start with

// and everything up until the end of that line is considered a comment.

The first line of code is:

int led = 13;

As the comment above it explains, this is giving a name to the pin that the LED is attached to. This is 13 on most Arduinos, including the UNO and Leonardo.

Next, we have the 'setup' function. Again, as the comment says, this is executed when the reset button is pressed. It is also executed whenever the board resets for any reason, such as power first being applied to it, or after a sketch has been Uploaded.

```
void setup() {

  // initialize digital pin LED_BUILTIN as an output.

  pinMode(LED_BUILTIN, OUTPUT);

}
```

Every Arduino sketch must have a 'setup' function, and the place where you might want to add instructions of your own is between the { and the }.

In this case, there is just one command there, which, as the comment states tells the Arduino board that we are going to use the LED pin as an output.

It is also mandatory for a sketch to have a 'loop' function. Unlike the 'setup' function that only runs once, after a reset, the 'loop' function will, after it has finished running its commands, immediately start again.

```
void loop() {

digitalWrite(led, HIGH); // turn the LED on

(HIGH is the voltage level)

delay(1000);// wait for a second
```

digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
delay(1000) // wait for a second
}

Inside the loop function, the commands first of all turn the LED pin on (HIGH), then 'delay' for 1000 milliseconds (1 second), then turn the LED pin off and pause for another second.

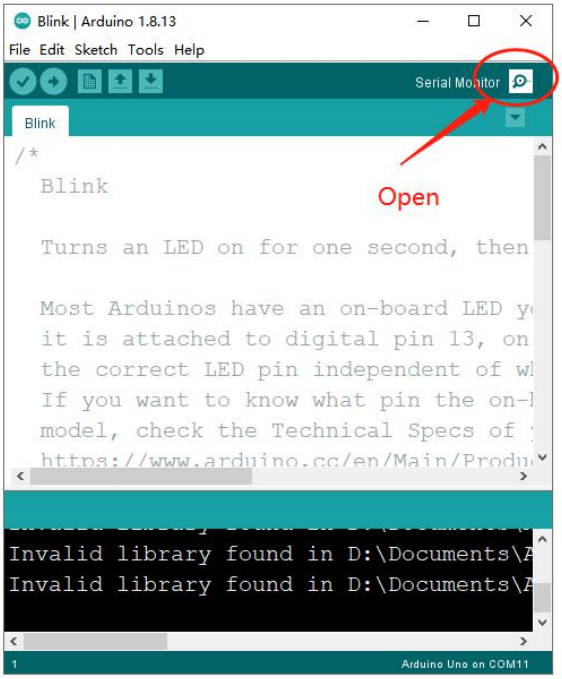
You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

```
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

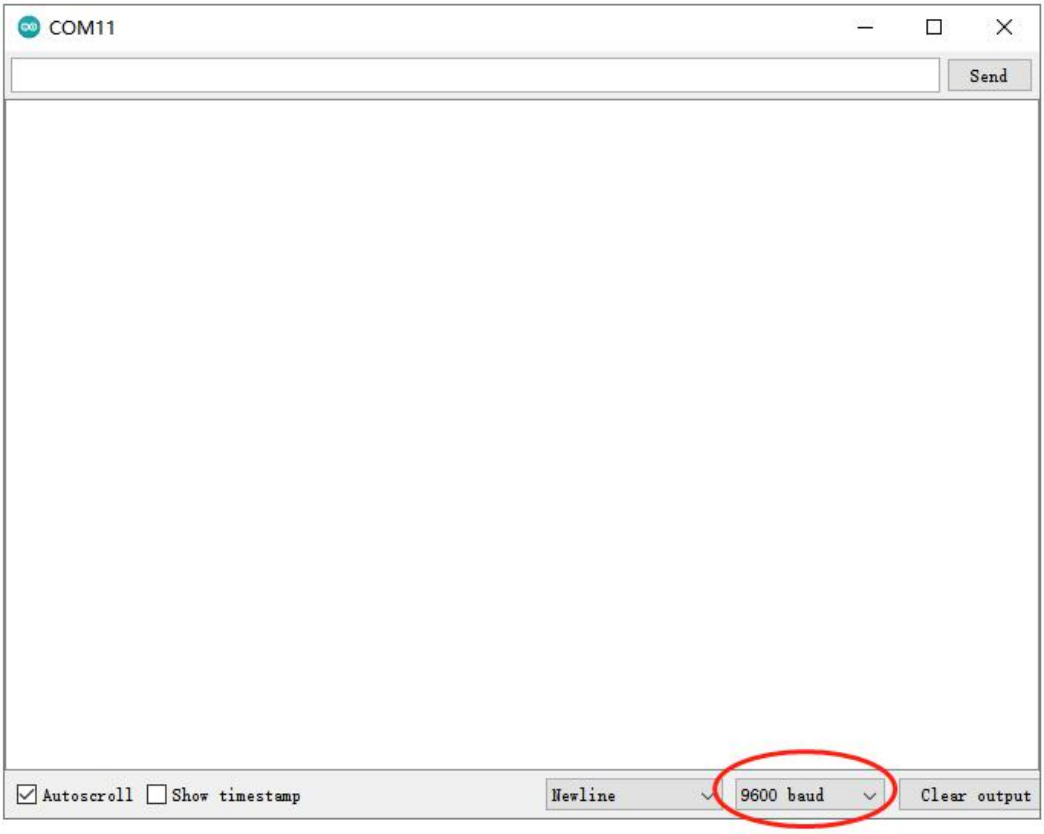
This delay period is in milliseconds, so if you want the LED to blink twice as fast, change the value from 1000 to 500. This would then pause for half a second eachdelay rather than a whole second.

Upload the sketch again and you should see the LED start to blink more quickly.

Note: I want to add some information about how to open the serial monitor here.



Serial port is generally selected 9600HZ

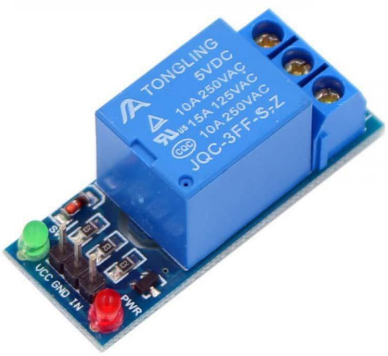


Lesson 5 Relay Module

Overview:

In this course you will learn how to use relay modules.

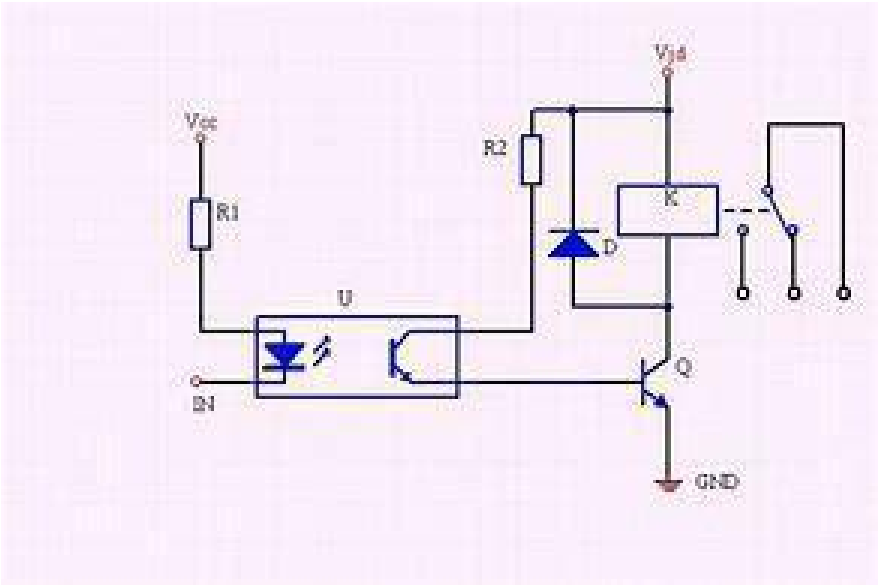
When the input variable (excitation) becomes a specified requirement, the relay is a component in which the amount of charge of the output circuit occurs due to a step change of an electric appliance. The company's relay modules can handle a wide range of other electrical components from 28V to 240V AC or DC power supplies. The MCU can be used to achieve the goal of timing control switches. Can be used in burglar alarm, toys, construction and other fields. A relay is an electrical control device. It has an interaction between a control system (also called an input circuit) and a control system (also known as an output circuit). Typically used in automatic control circuits, it is actually performed in the current operation of the "automatic control circuit".



Component Required:
1 *Uno R3
1 * 5v Relay module
1*400 tie-points breadboard
1*6v submersible pump
3 x M-M wires (Male to Male DuPont wires)
3 x F-M wires (Female to Male DuPont wires)

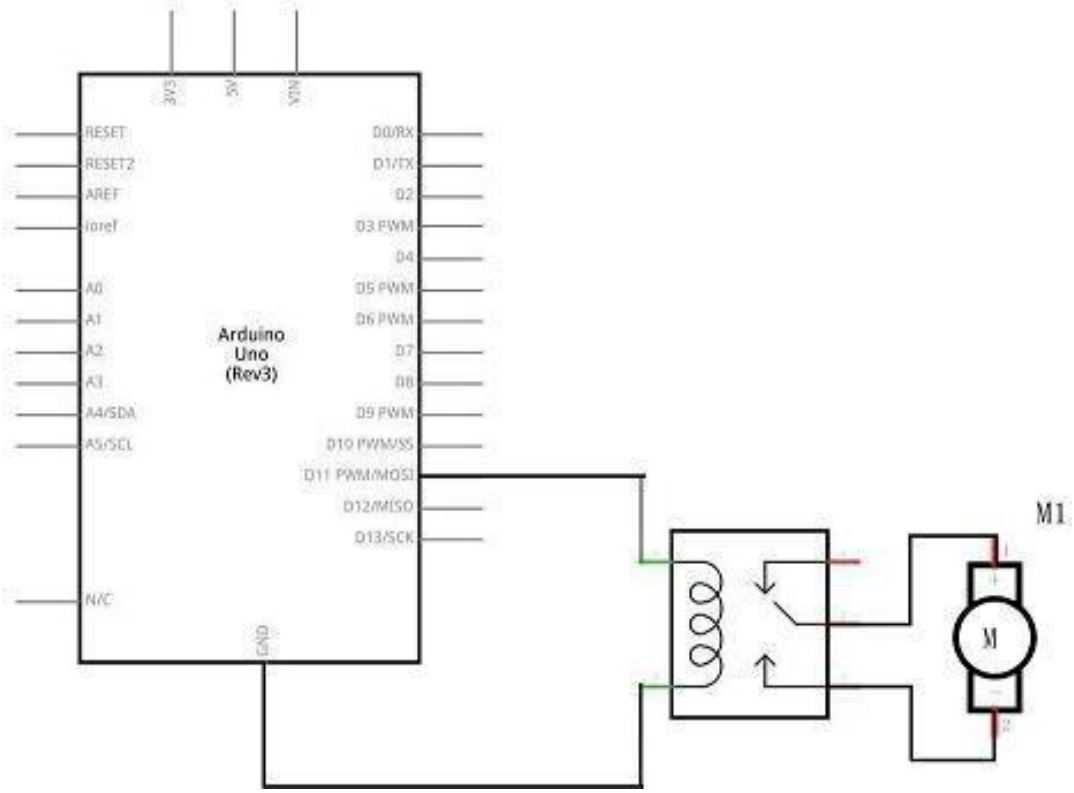
Component Introduction Relay:

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".

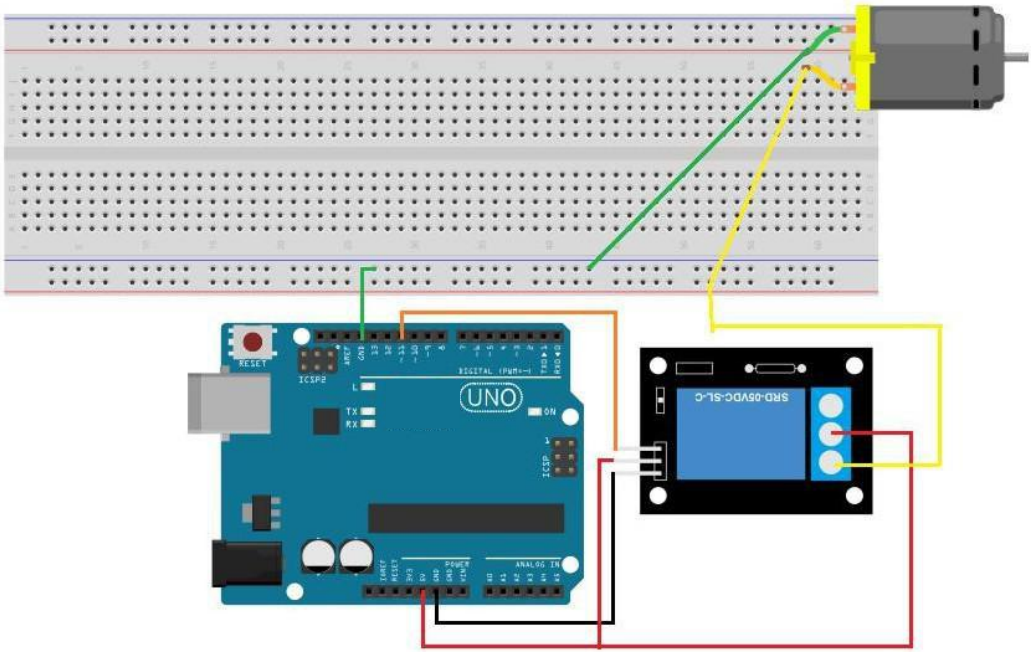


DC 3-6V 120L/H Mini Submersible Water Pump:
echnical Specifications
DC Voltage: 2.5-6V
Maximum lift: 40-110cm / 15.75”-43.4” Flow rate: 80-120L/H
Outside diameter of water outlet: 7.5mm / 0.3” Inside diameter of water outlet: 5mm / 0.2” Diameter: Approx. 24mm / 0.95”
Length: Approx. 45mm / 1.8” Height: Approx. 30mm / 1.2” Material: engineering plastic

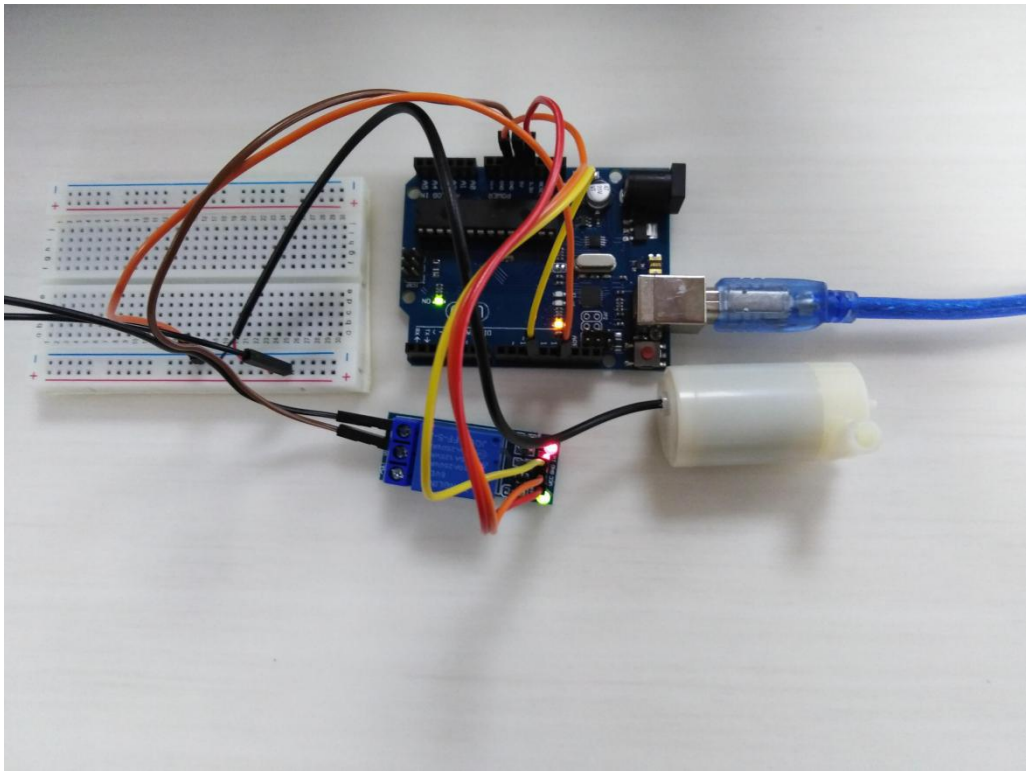
Connection Diagram:



Wiring schematic :



Physical wiring diagram:



Code:

```
void setup()
{
pinMode(11,OUTPUT);
}
void loop()
{

digitalWrite(11,HIGH); delay(4000);
digitalWrite(11, LOW); delay(4000);

}
```

For the Arduino syntax, you can refer to the following link:
https://www.arduino.cc/reference/en/

Lesson 6 Soil moisture detection module

Overview:

In this course you will learn how to use Soil moisture detection modules. This is a summary of the soil moisture sensor can be used to detect moisture, when the soil is dry, the module outputs a high level, whereas output low. Using this sensor make an automatic watering system, so that your garden plants without people to manage.

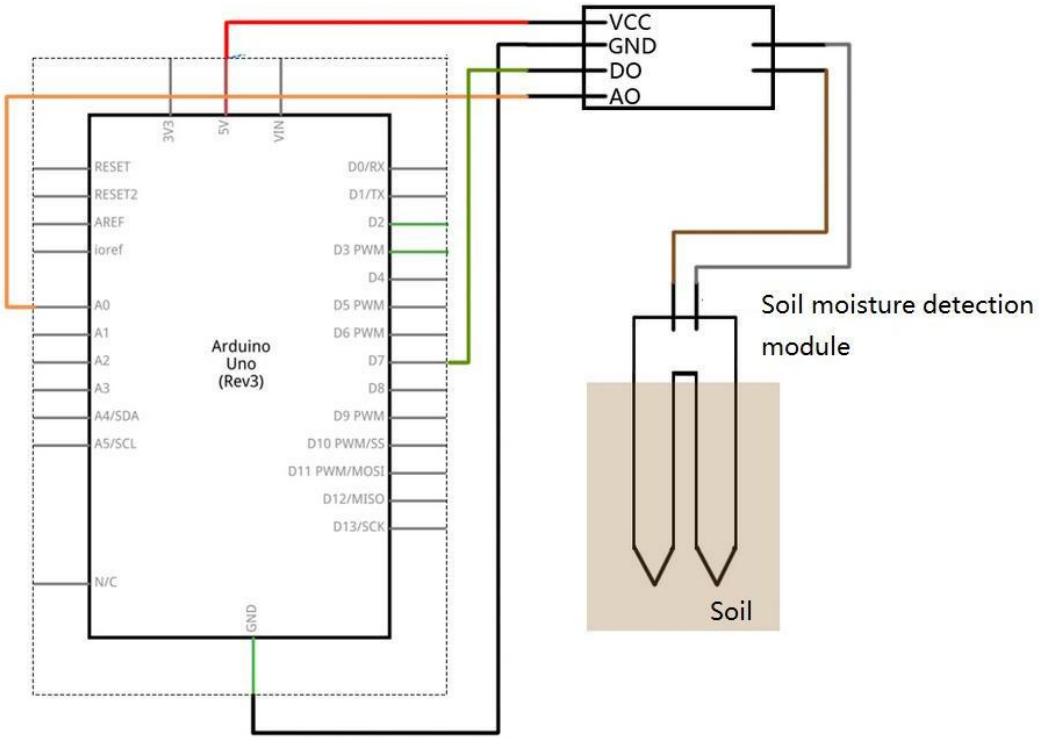
Component Required:

1 x Arduino Uno R3
1x Soil moisture detector module
2x F-F wires (Female to Female DuPont wires)
4 x F-M wires (Female to Male DuPont wires)

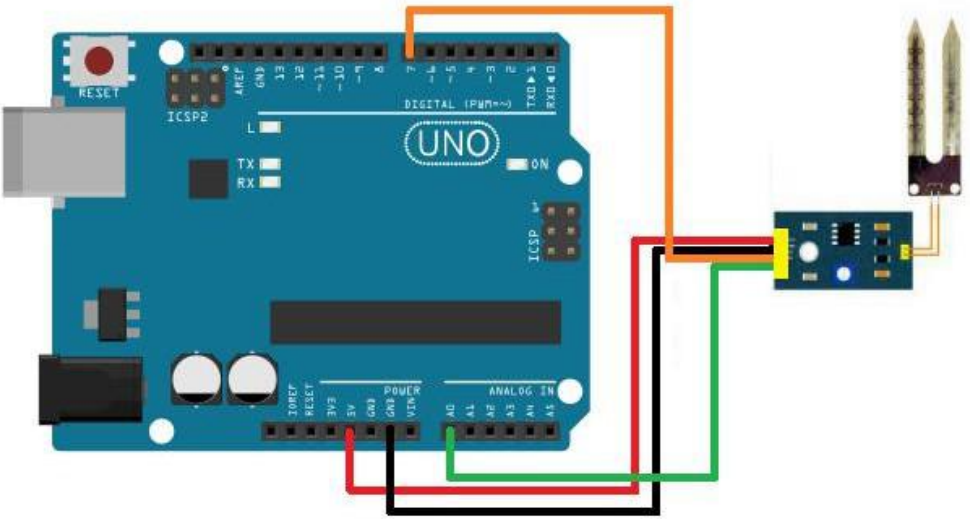
Component Introduction:

Soil Moisture Sensor Features:
The working voltage of 3.3 V to 5 V.
Simple digital output, g + v directly by single chip microcomputer. The sensitivity is adjustable using digital potentiometer (blue).
A fixed bolt hole, convenient installation.
The power indicator light (red) and digital switch output indicator light (green). The comparator uses the LM393 chips, work stability.

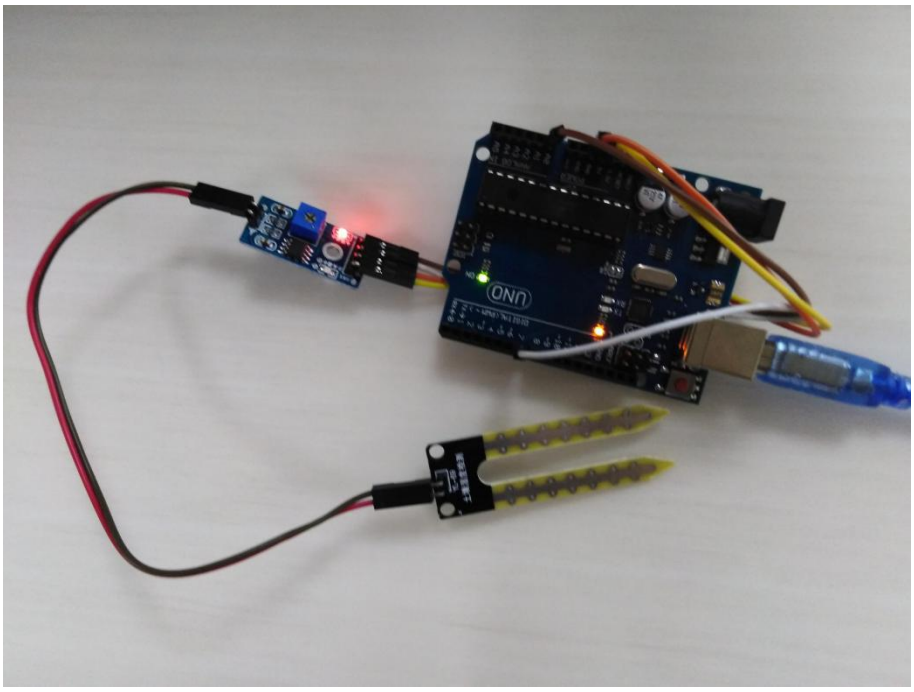
Connection Diagram:



Wiring schematic :

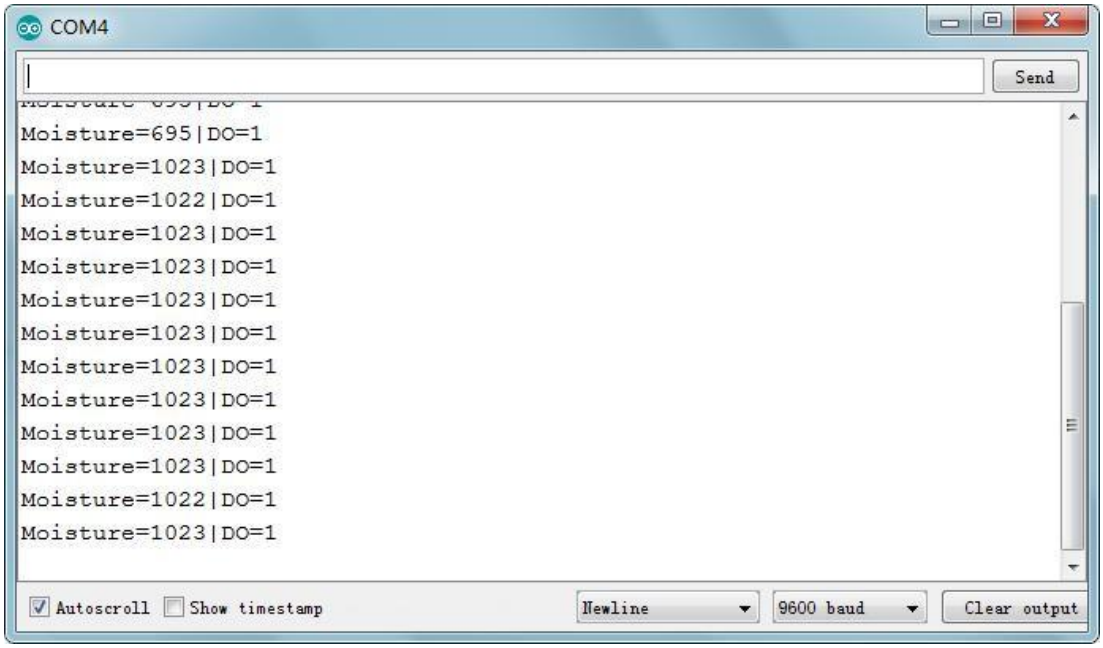


Physical wiring diagram:

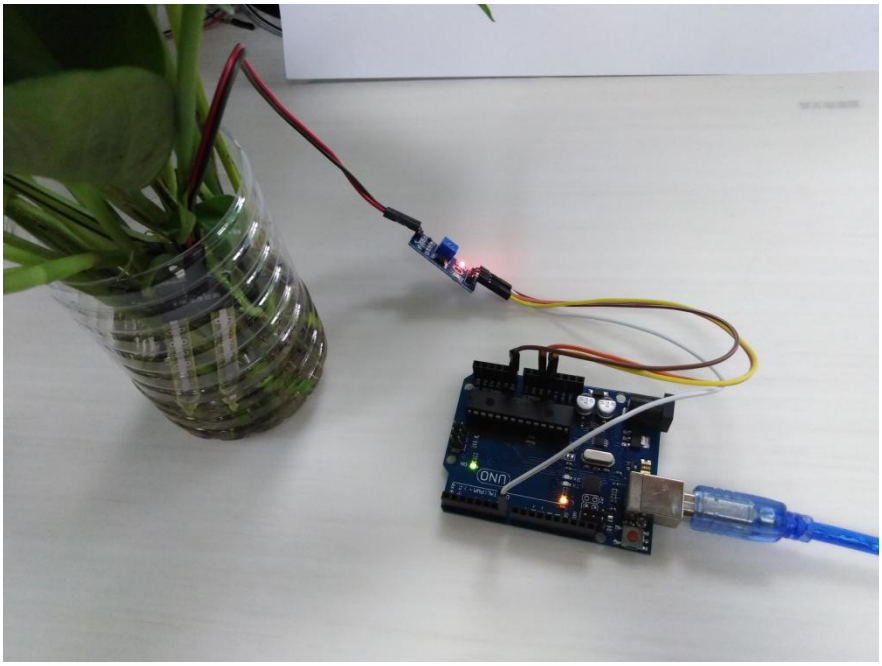


Note: When operating at 5V, the maximum value of AO reading in air is 1023, and the minimum value of immersion in water is 245. If it is working at 3.3V, the maximum value of AO reading in air is 695, and the minimum value of immersion in water is 245;

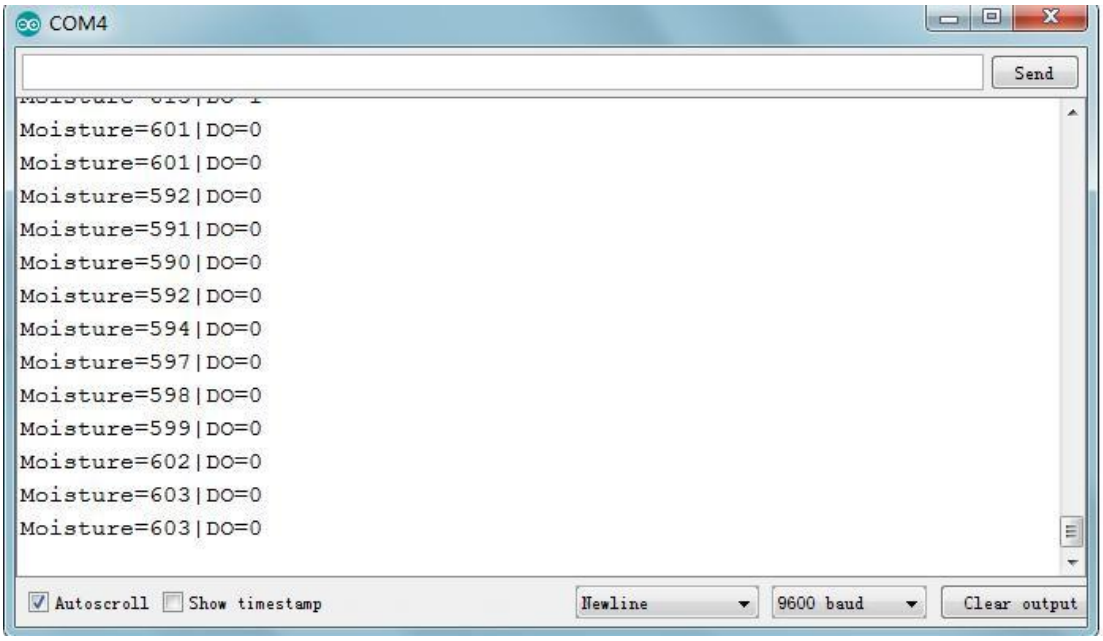
Open the serial monitor as shown:



When the soil moisture module is exposed to water, the detection humidity is relatively large, as shown in the following figure:



Sensors soaked in water and lifted data changes,The smaller the Moisture value, the higher the humidity and vice versa.



Code:

```
//LingShun Lab
#define Moisture A0  //Define AO pin as IO-A0
#define DO 7       //The DO pin is defined as io-7

void setup() {
pinMode(Moisture, INPUT); //Define A0 as the input mode
pinMode(DO, INPUT);
Serial.begin(9600);

}
void loop() {
//The serial port returns the measured data
Serial.print("Moisture=");
Serial.print(analogRead(Moisture)); //Read the value of the AO
Serial.print("|DO=");
Serial.println(digitalRead(DO)); //Read the value of DO
delay(1000);
}
```

Lesson 7 Based on UNO R3 DIY automatic watering system

Overview:

After studying the above lessons, we are fully capable of making our own DIY automatic watering system through this lesson.

Component Required:
1 x Uno R3 1x 6V battery
1 x 5v Relay module
1 x Soil moisture detector module
1 x F-F wires (Female to Female DuPont wires)
7 x F-M wires (Female to Male DuPont wires)
1xRubber hose 1m

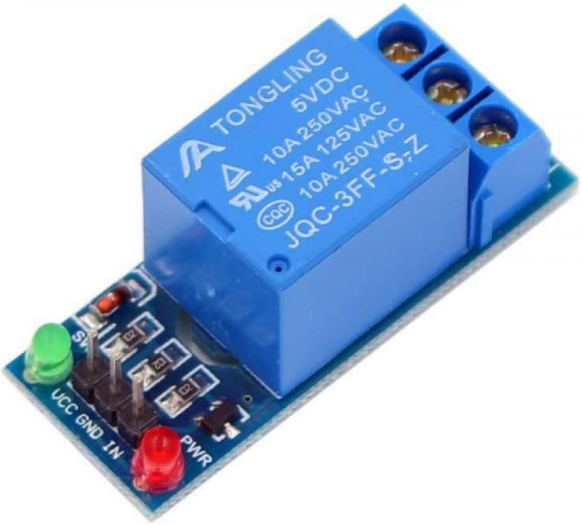
Component Introduction:

Soil Moisture Sensor Features:

The working voltage of 3.3 V to 5 V.
Simple digital output, g + v directly by single chip microcomputer. The sensitivity is adjustable using digital potentiometer (blue).
A fixed bolt hole, convenient installation.
The power indicator light (red) and digital switch output indicator light (green). The comparator uses the LM393 chips, work stability.

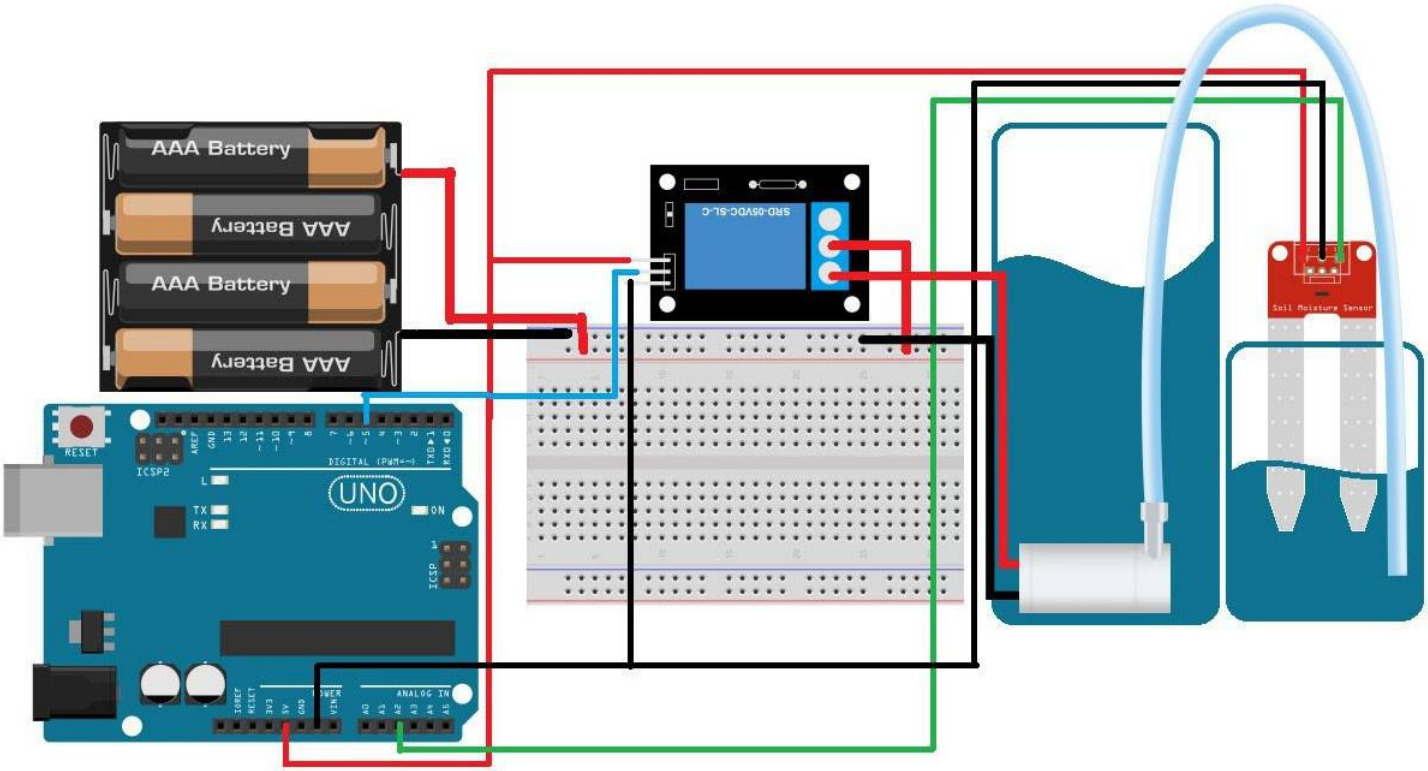
5v relay module:

Switch AC up to 125/250VAC @ 10A
Switch DC up to 15VDC @ 10A
Normally Open (NO) and Normally Closed (NC)
contacts LED indicator when relay is energized
5V logic compatible

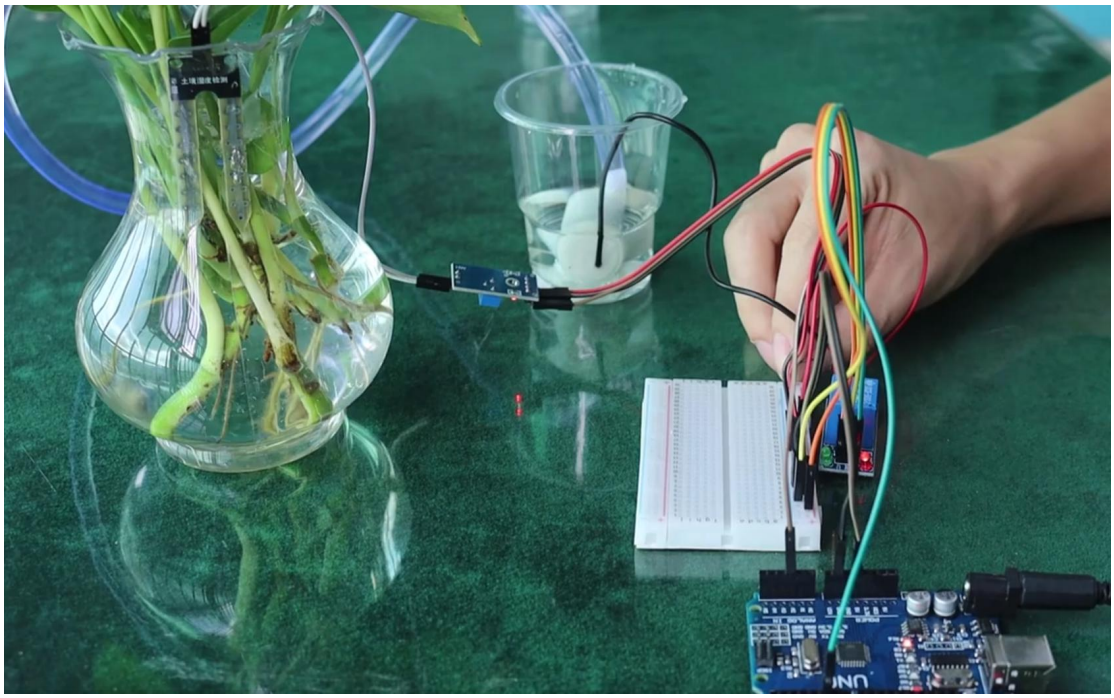


DC 3-6V 120L/H Mini Submersible Water Pump:
echnical Specifications
DC Voltage: 2.5-6V
Maximum lift: 40-110cm / 15.75”-43.4” Flow rate: 80-120L/H
Outside diameter of water outlet: 7.5mm / 0.3” Inside diameter of water outlet: 5mm / 0.2” Diameter: Approx. 24mm / 0.95”
Length: Approx. 45mm / 1.8” Height: Approx. 30mm / 1.2” Material: engineering plastic

Wiring schematic :



Physical wiring diagram:



Code:

```
#define MOISTURE_PIN    A2    //Soil moisture sensor pins

int soilHumidity;//soil moisture content
int setHumidity = 50; //Set the humidity
void setup() {
   //Pump control pin initialization
   pinMode(5, OUTPUT);
   digitalWrite(5, HIGH);

}

void loop() {
   soilHumidity = map(analogRead(MOISTURE_PIN), 0, 1023, 0, 100);//Convert the soil moisture
sensor to 0-100%
   if (soilHumidity < setHumidity)
   {
      pumpOn();
   } else {
      pumpOff();
```

```
  }
}
//Open pump
void pumpOn() {
  digitalWrite(5, LOW);

}
//The pump shut down
void pumpOff() {
  digitalWrite(5, HIGH);

}
```