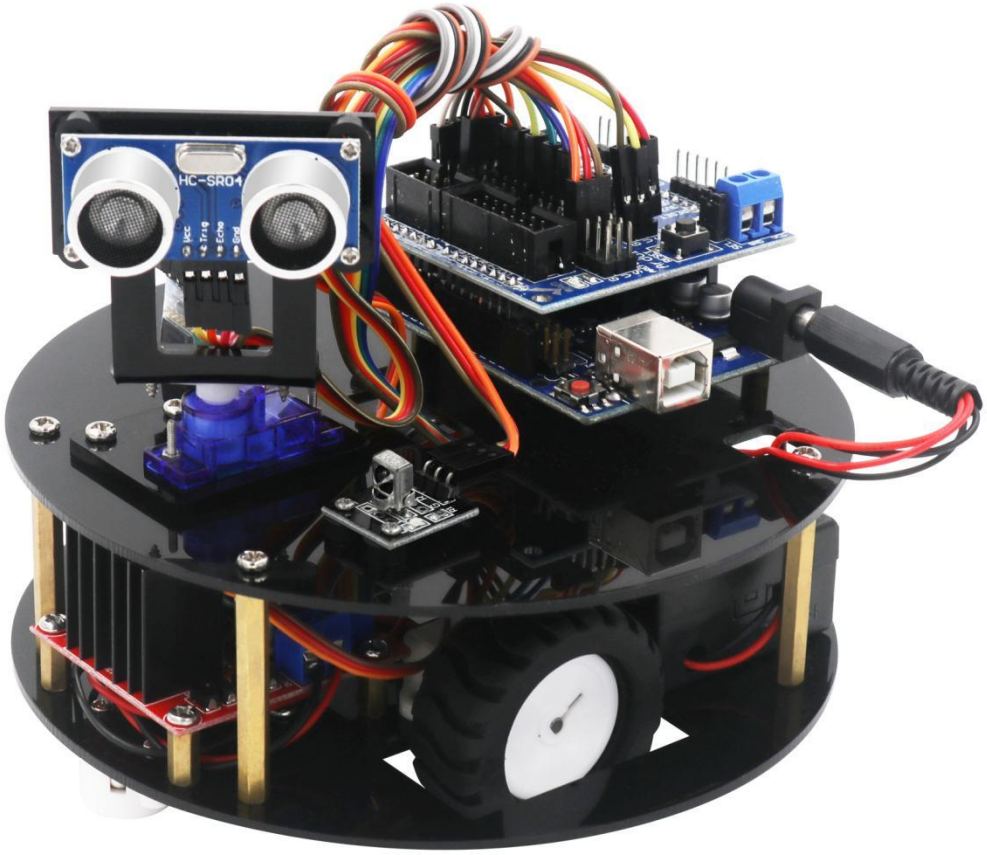
Smart Turtle Robot Car Kit


HC-SR04
Vcc
Trig
Echo
Gnd

Content

Company Profile

Established in 2011, lafvin is a manufacturer and trader specialized in research,development and production of 2560 uno,nano boards,and all kinds of accessories or sensors use for arduino,rasperrry.We also complete starter kits designed for interested lovers of any levels to learn Arduino or Raspberry.We are located in Shenzhen,China.All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.

Customer Service

We are cooperating with a lot of companies from diffirent countries.Also help them to purchase electronic component products in china,and became the biggest supplier of them. We look forward to build cooperate with more companies in future.

By the way,We also look forward to hearing from you and any of your critical comment or suggestions.Pls email us by lafvin_service@163.com if you have any questions or suggestions.

As a continuous and fast growing company. We keep striving our best to offer you excellent products and quality service.

Our Store

Aliexpress store: https://www.aliexpress.com/store/1942043

Brand in Amazon:LAFVIN

Product Catalog

https://drive.google.com/drive/folders/0BwvEeRN9dKllblZING00TkhYbGs?usp=sharing

Tutorial

This tutorial include codes,labraries and lessons. It is designed for beginners. It will teach every users how to assembly the smart robot car kit and use Arduino controller board, sensors and modules.

Packing list

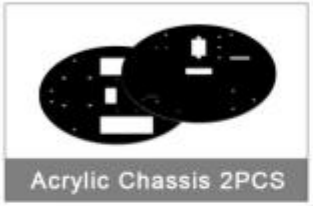

UNO R3 with Cable 1PCS


V5 Expansion Board 1PCS


L298N Motor Driver Board 1PCS


Ultrasonic Sensor 1PCS


Bluetooth Module 1PCS

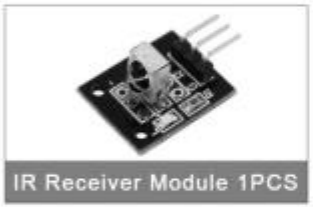

Acrylic Chassis 2PCS


Servo Motor(SG90) 1PCS


Cell Box 1PCS


F-F Dupont Wire 1PCS


Remote Control 1PCS


IR Receiver Module 1PCS


Tires 2PCS


DC Motor + Holder 2PCS


Line Tracking Module 3PCS


Universal Wheel 2PCS


M3*10mm 6PCS
M3*6mm 26PCS


M2*8mm 4PCS
M1.6*12mm 8PCS


Ultrasonic Holder 1PCS

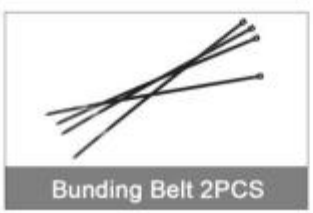

Plastic Shim 9PCS

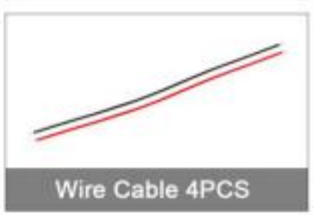

Nut 22PCS


Copper Cylinder 11PCS


Black Tape 1PCS


Screwdriver 1PCS


Bunding Belt 2PCS


Wire Cable 4PCS

Lesson 1 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.](https://www.arduino.cc/en/Main/Software)

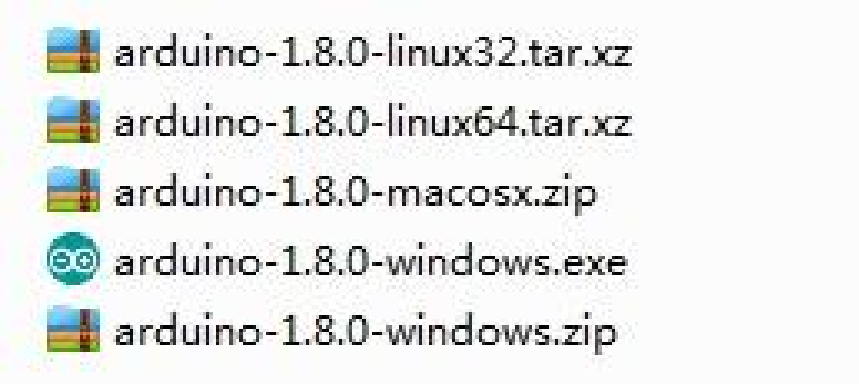
The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

STEP2：Download the development software that is compatible with the operating system of your computer. <u>Take Windows as an example here.</u>

Click Windows Installer.

Click JUST DOWNLOAD.

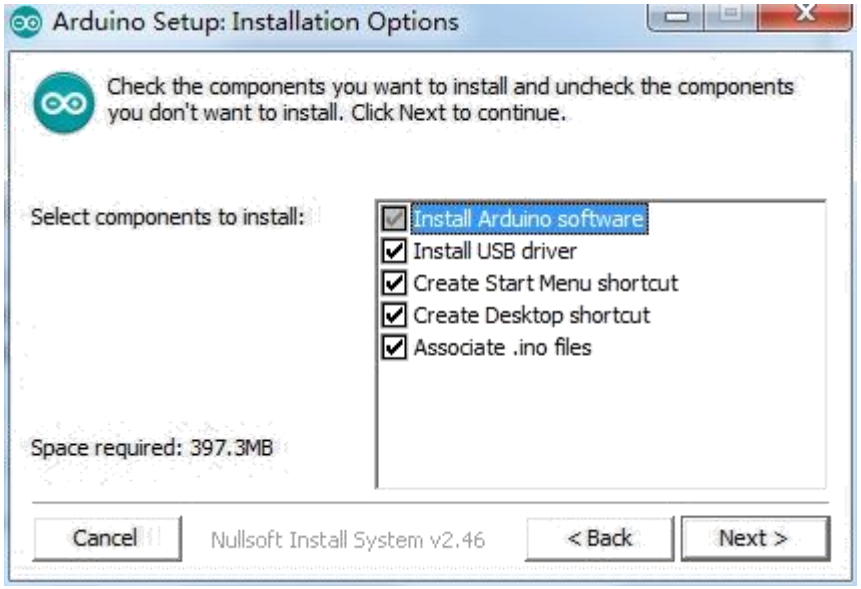
Also version 1.8.0 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.



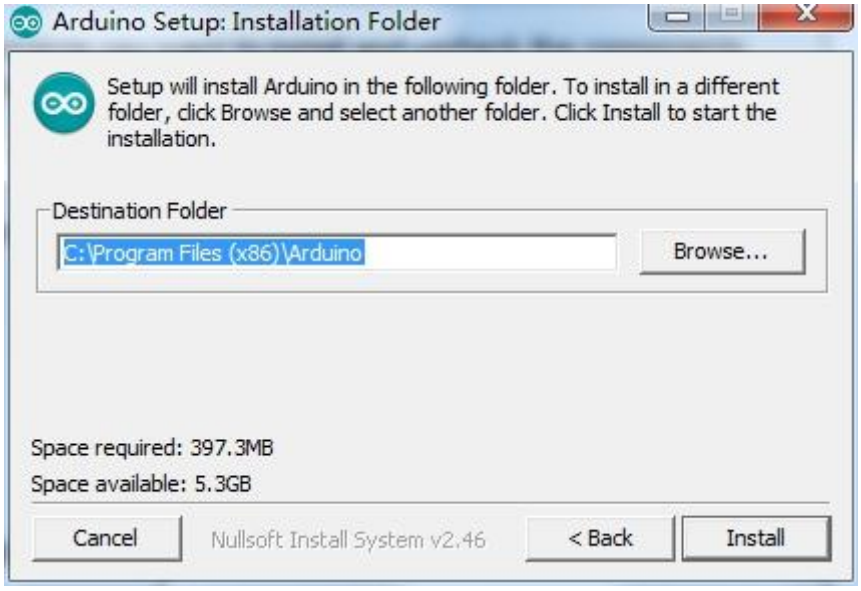
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

Click I Agree to see the following interface

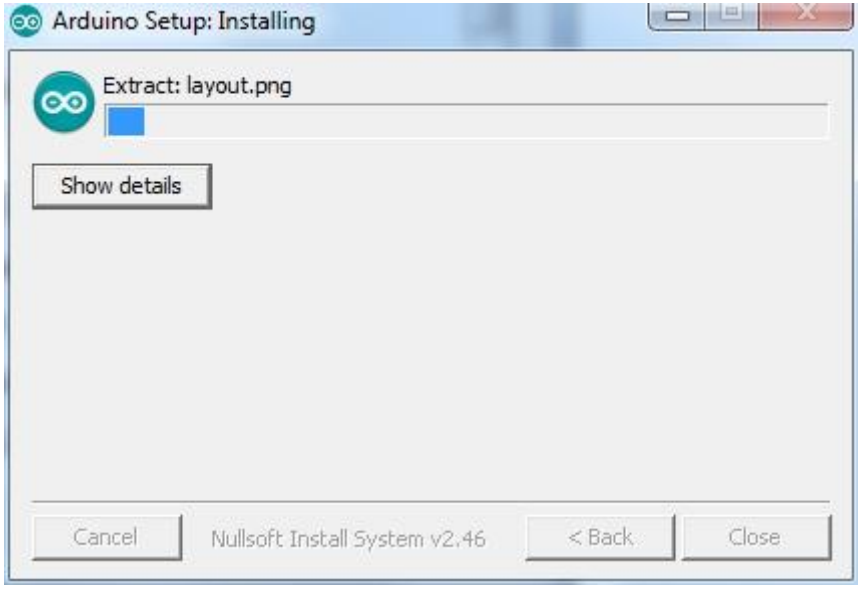


Click Next



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

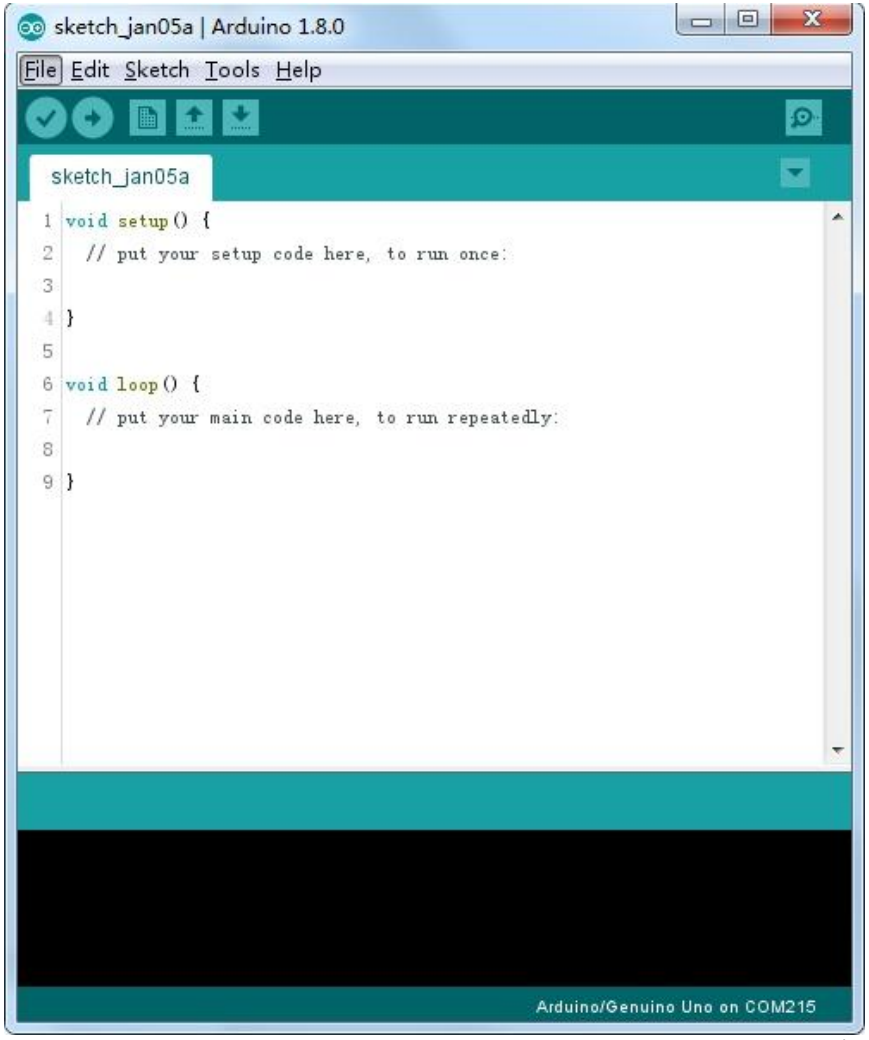
Click Install to initiate installation



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

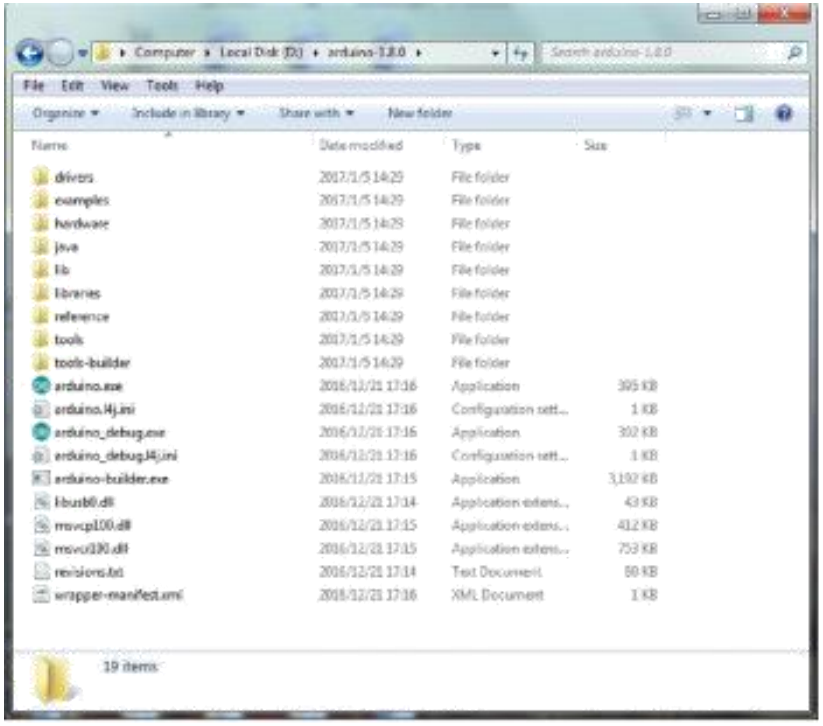
Next, the following icon appears on the desktop

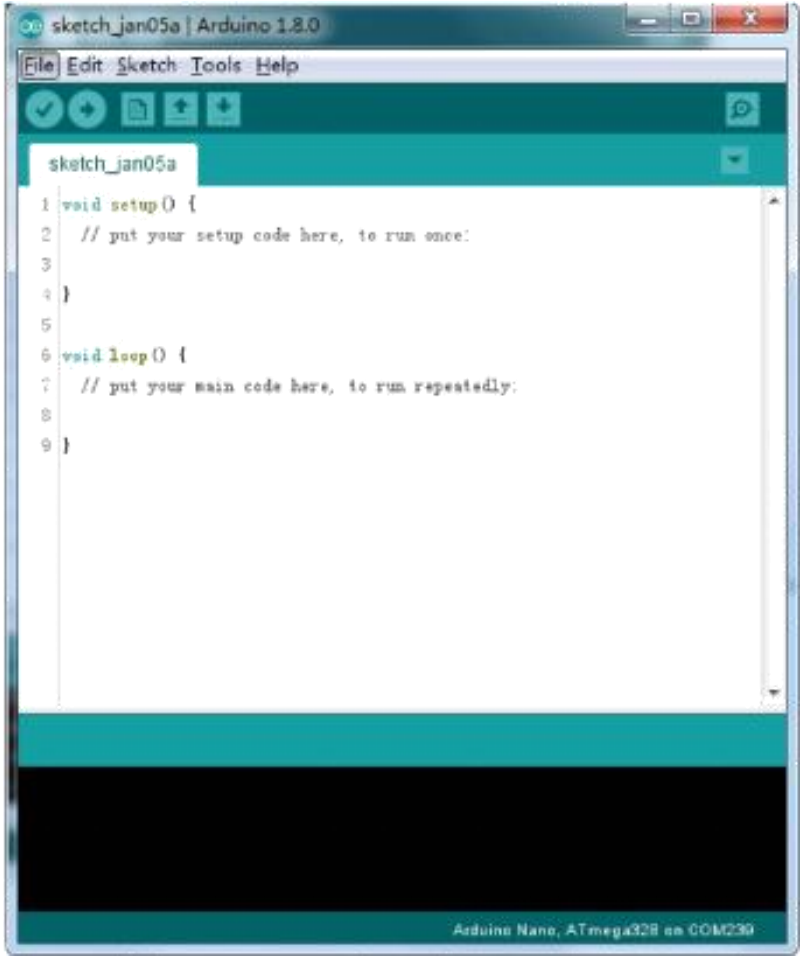
Double-click to enter the desired development environment



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


Computer ▸ Local Disk (D:) ▸ arduino-1.8.0 ▸
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Name Date modified Type Size
drivers 2017/1/5 14:29 File folder
examples 2017/1/5 14:29 File folder
hardware 2017/1/5 14:29 File folder
java 2017/1/5 14:29 File folder
lib 2017/1/5 14:29 File folder
libraries 2017/1/5 14:29 File folder
reference 2017/1/5 14:29 File folder
tools 2017/1/5 14:29 File folder
tools-builder 2017/1/5 14:29 File folder
arduino.exe 2016/12/21 17:16 Application 395 KB
arduino.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino_debug.exe 2016/12/21 17:16 Application 392 KB
arduino_debug.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino-builder.exe 2016/12/21 17:15 Application 3,192 KB
libusb0.dll 2016/12/21 17:14 Application extens... 43 KB
msvcp100.dll 2016/12/21 17:15 Application extens... 412 KB
msvcr100.dll 2016/12/21 17:15 Application extens... 753 KB
revisions.txt 2016/12/21 17:14 Text Document 80 KB
wrapper-manifest.xml 2016/12/21 17:16 XML Document 1 KB
19 items


sketch_jan05a | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05a
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Arduino Nano, ATmega328 on COM239

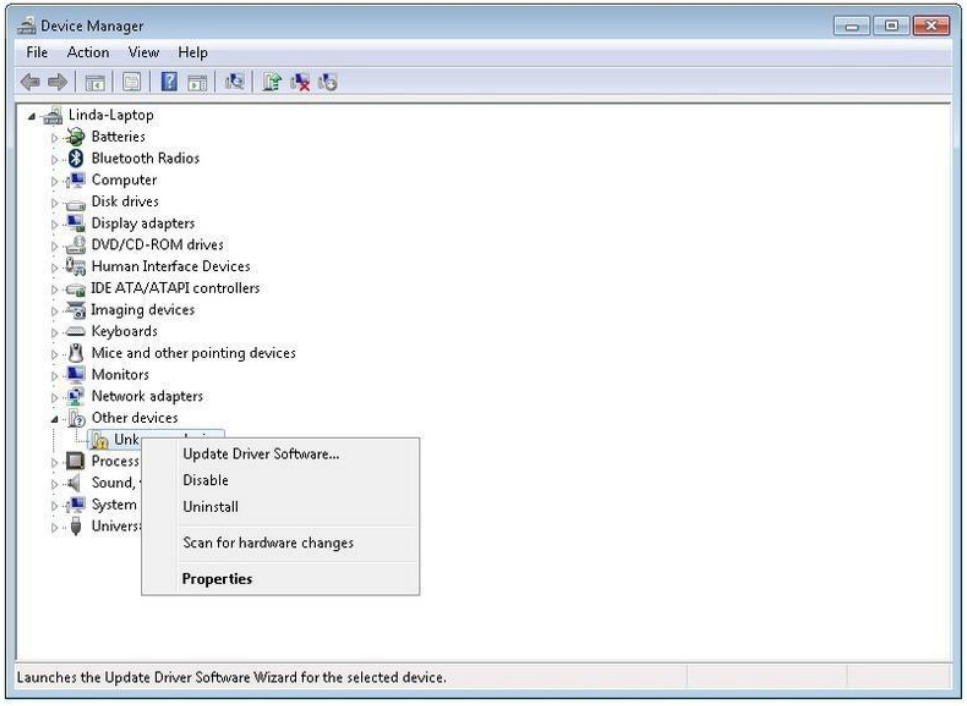
However, this installation method needs separate installation of driver.

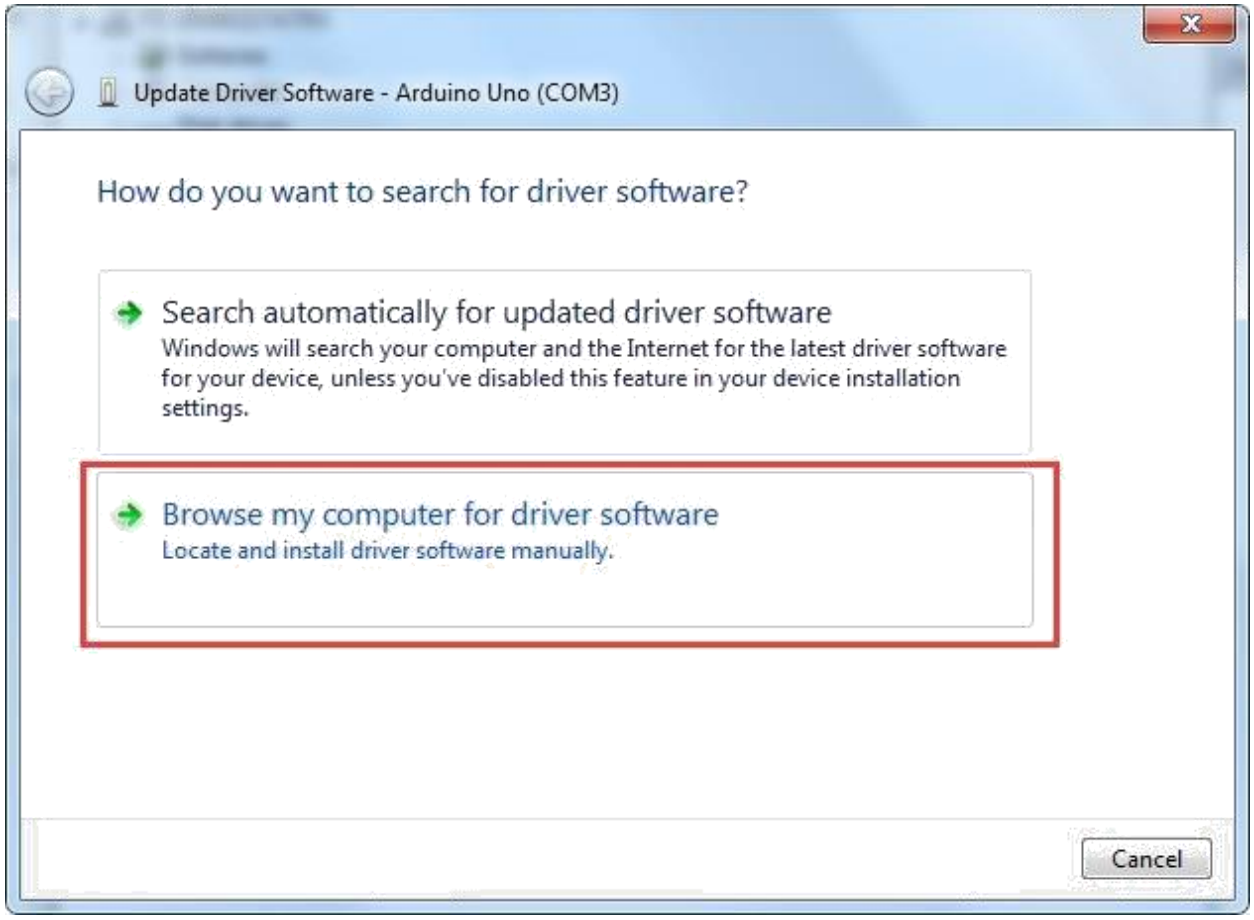
The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer. The power light on the LED will light up and you may get a 'Found New Hardware' message from Windows. Ignore this message and cancel any attempts that Windows makes to try and install drivers automatically for you.

The most reliable method of installing the USB drivers is to use the Device Manager. This is accessed in different ways depending on your version of Windows. In Windows 7, you first have to open the Control Panel, then select the option to view Icons, and you should find the Device Manager in the list.

Under ‘Other Devices’, you should see an icon for ‘unknown device’ with a little yellow warning triangle next to it. This is your Arduino.





Right-click on the device and select the top menu option (Update Driver Software...).

You will then be prompted to either ‘Search Automatically for updated driver software’ or ‘Browse my computer for driver software’. Select the option to browse and navigate to the X\arduino1.8.0\drivers.

Click 'Next' and you may get a security warning, if so, allow the software to be installed. Once the software has been installed, you will get a confirmation message.

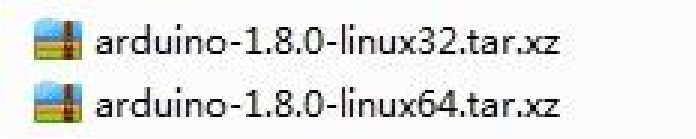
Windows users may skip the installation directions for Mac and Linux systems and jump to Lesson 1. Mac and Linux users may continue to read this section.

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 2 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.

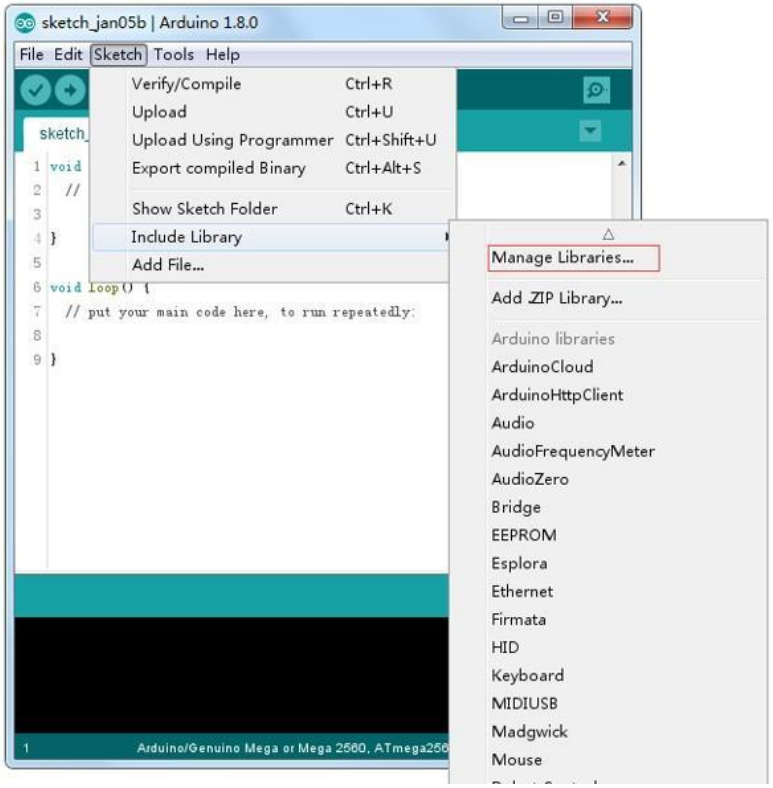
What are Libraries?

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 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

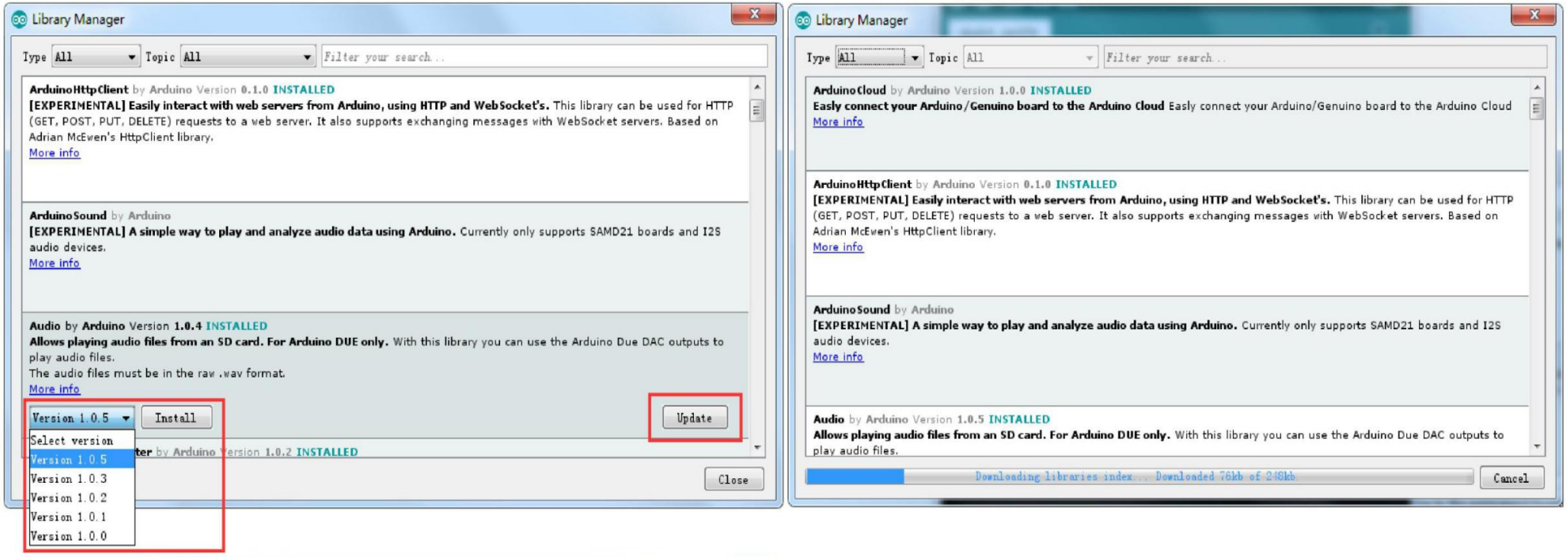
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.0). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



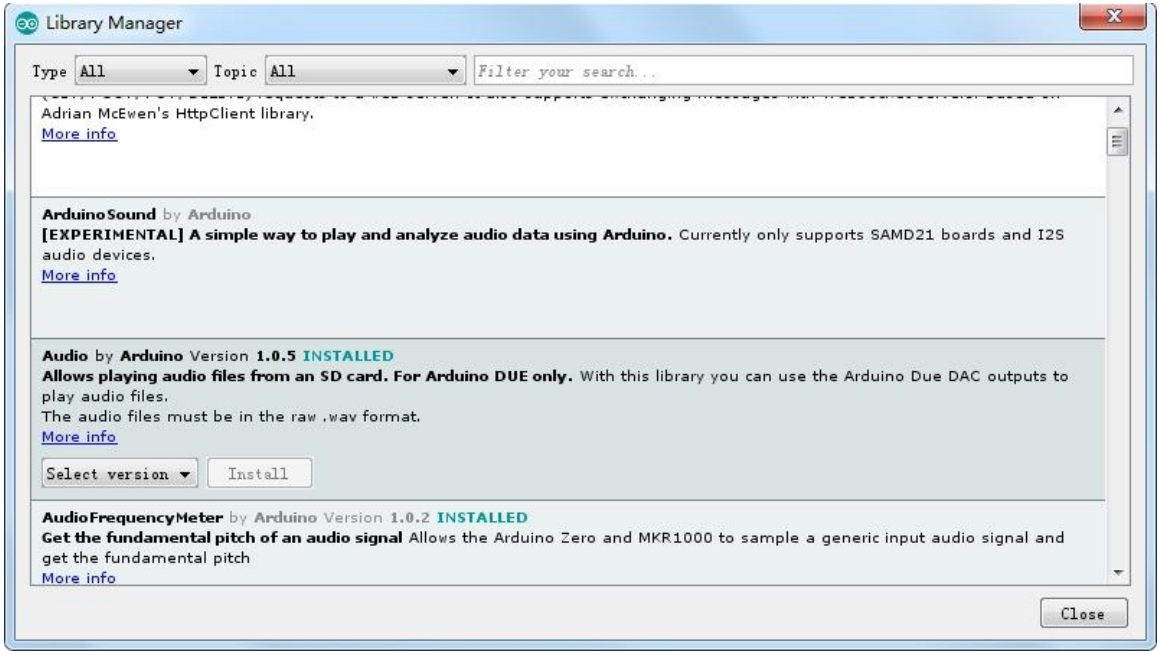
Then the library manager will open and you will find a list of libraries that are already installed or ready for installation. In this example we will install the Bridge library. Scroll the list to find it, then select the version of the library you want to install. Sometimes only one version of the 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.



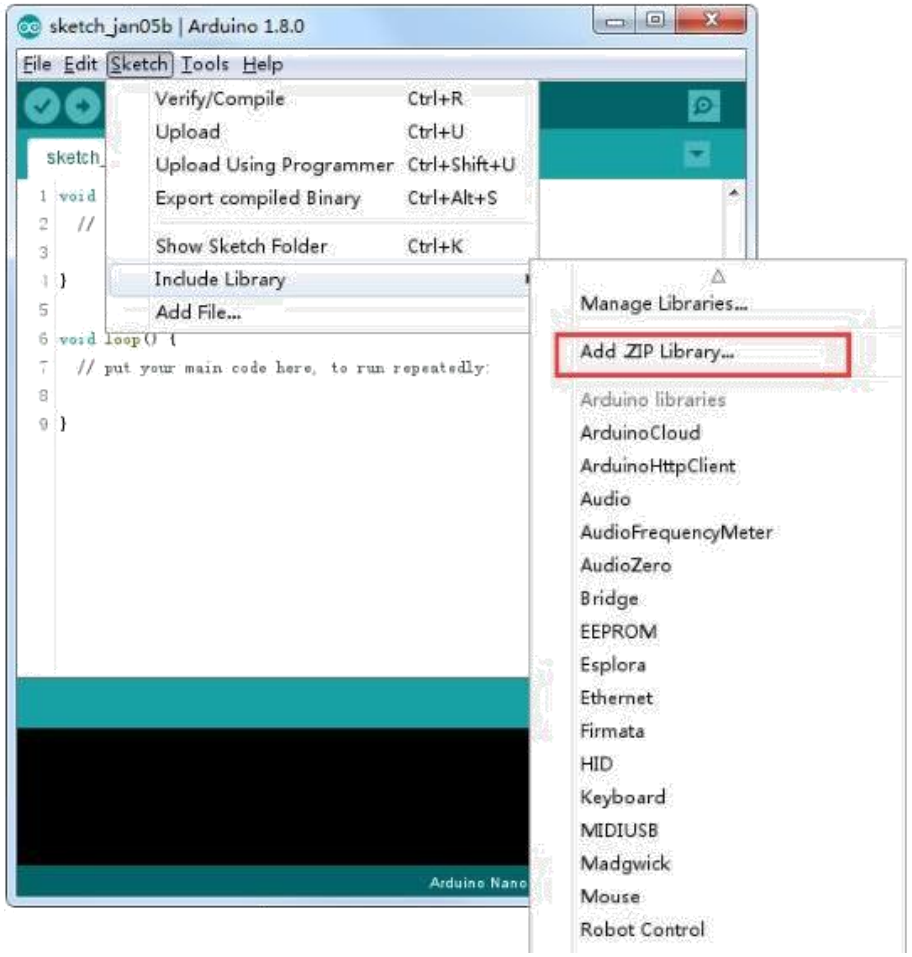
You can now find the new library available in the Include Library menu. If you want to add your own library open a new issue on Github.

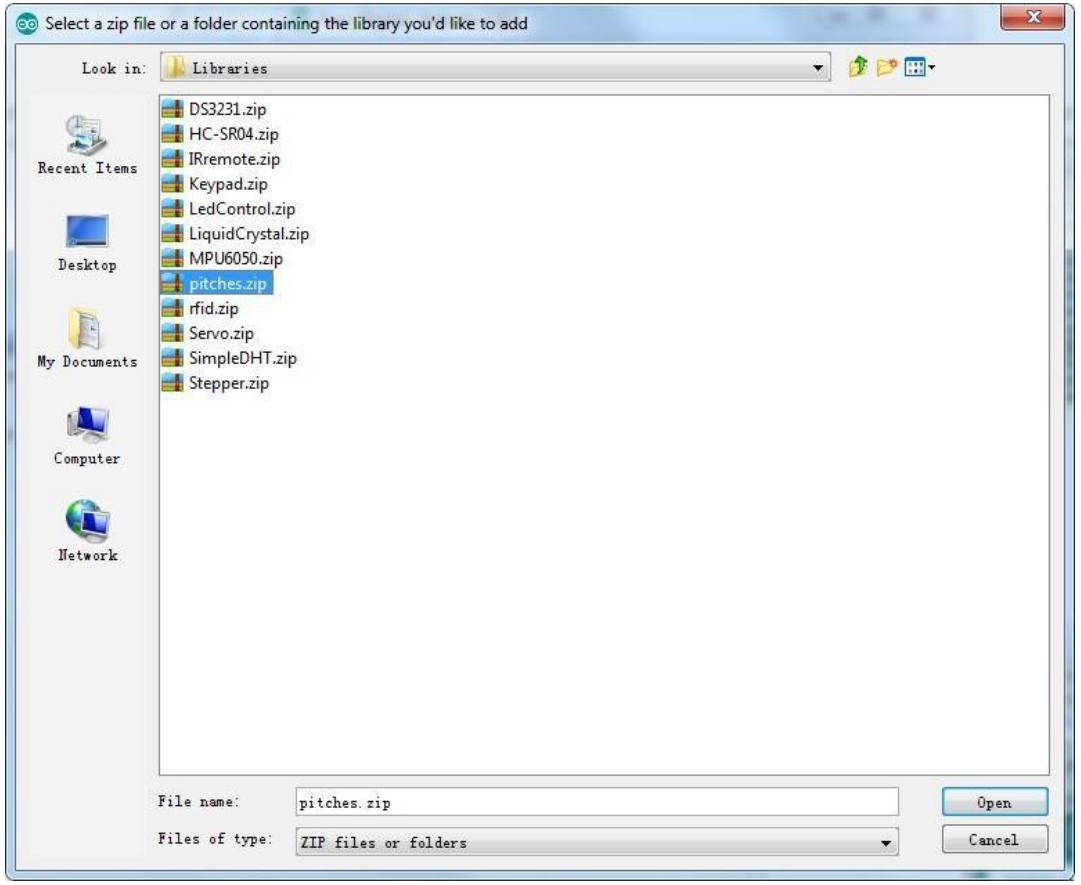
Importing a .zip Library

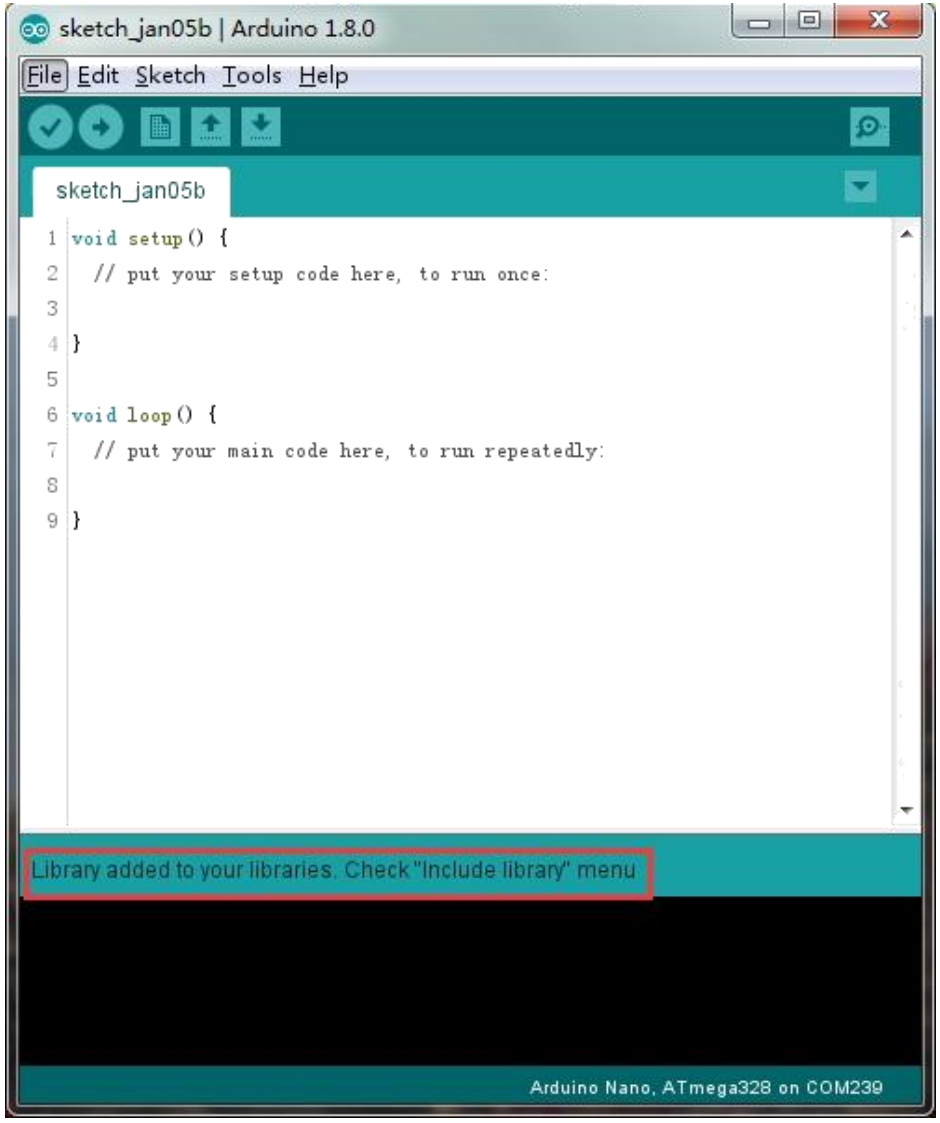
Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. Starting with version 1.0.5, you can install 3rd party 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.






sketch_jan05b | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05b
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Library added to your libraries. Check "Include library" menu
Arduino Nano, ATmega328 on COM239

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.

Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

Manual installation

To install the library, first quit the Arduino application. Then uncompress the ZIP file containing the library. For example, if you're installing a library called

"ArduinoParty", uncompress ArduinoParty.zip. It should contain a folder calledArduinoParty, with files like ArduinoParty.cpp and ArduinoParty.h inside. (If the .cpp and .h files aren't in a folder, you'll need to create one. In this case, you'd make a folder called "ArduinoParty" and move into it all the files that were in the ZIP file, like ArduinoParty.cpp and ArduinoParty.h.)

Drag the ArduinoParty folder into this folder (your libraries folder). Under Windows, it will likely be called "My Documents\Arduino\libraries". For Mac users, it will likely be called "Documents/Arduino/libraries". On Linux, it will be the "libraries" folder in your sketchbook.

Your Arduino library folder should now look like this (on Windows): My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h My Documents\Arduino\libraries\ArduinoParty\examples

or like this (on Mac and Linux): Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h Documents/Arduino/libraries/ArduinoParty/examples ....

There may be more files than just the .cpp and .h files, just make sure they're all there. (The library won't work if you put the .cpp and .h files directly into the libraries folder or if they're nested in an extra folder. For example: Documents\Arduino\libraries\ArduinoParty.cpp and Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp won't work.)

Restart the Arduino application. Make sure the new library appears in the Sketch->Import Library menu item of the software. That's it! You've installed a library!

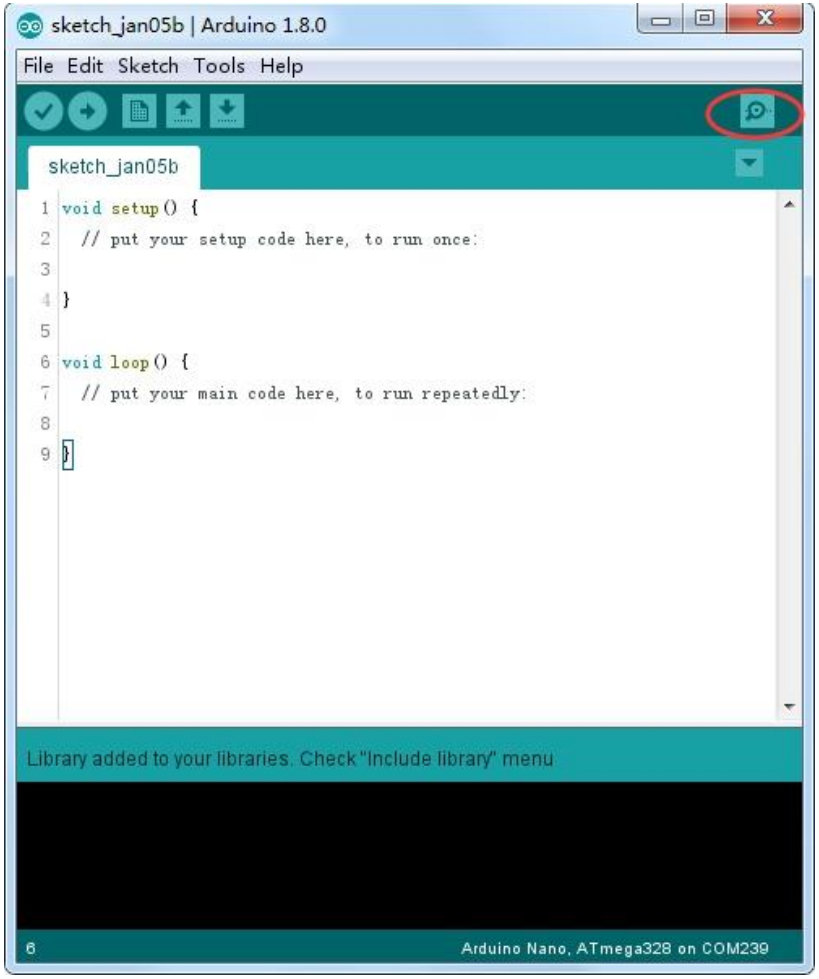
Arduino Serial Monitor (Windows, Mac, Linux)

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 with

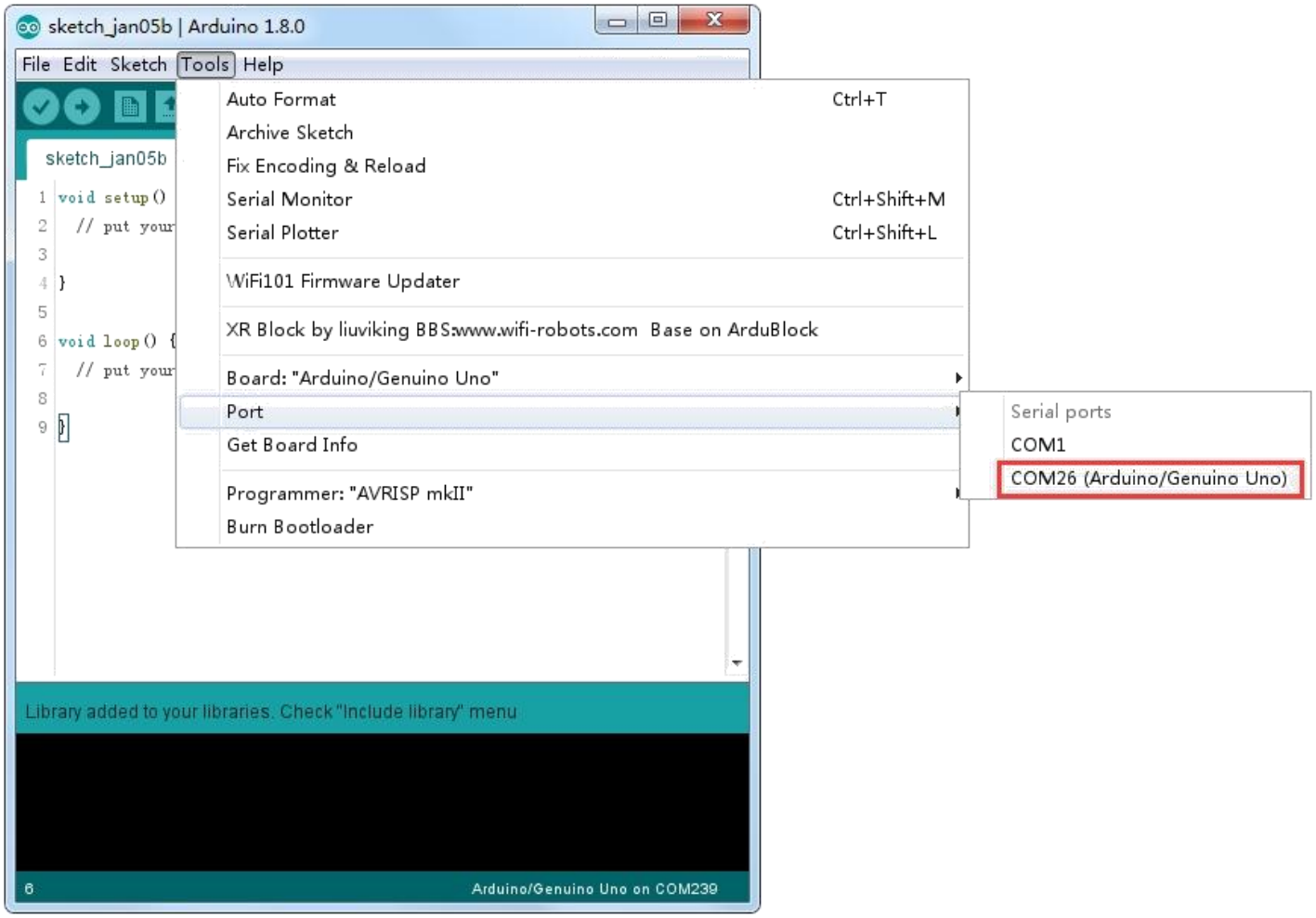
Arduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is called the Serial Monitor.

Making a Connection

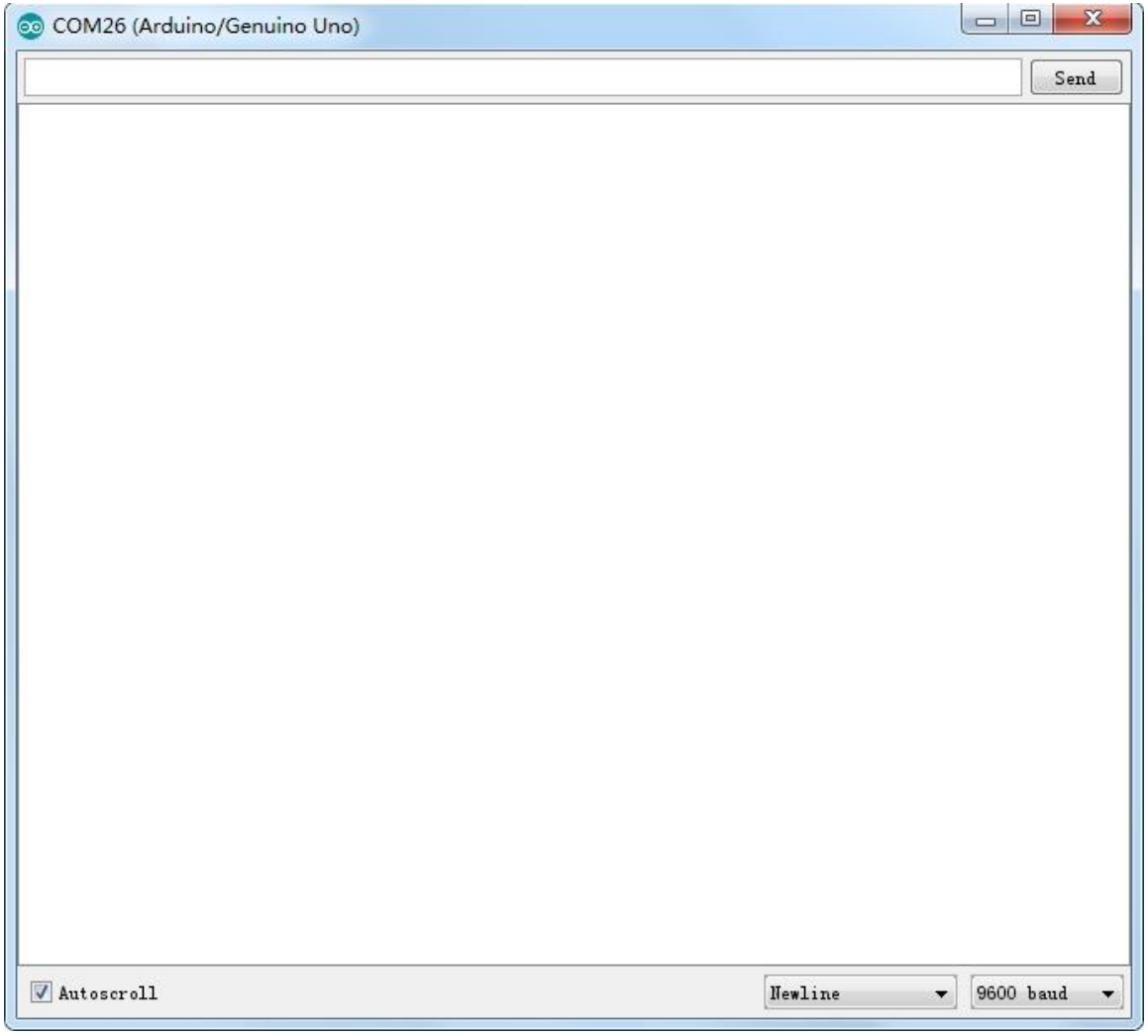
Serial monitor comes with any and all version of the Arduino IDE. To open it, simply click the Serial Monitor icon.



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.

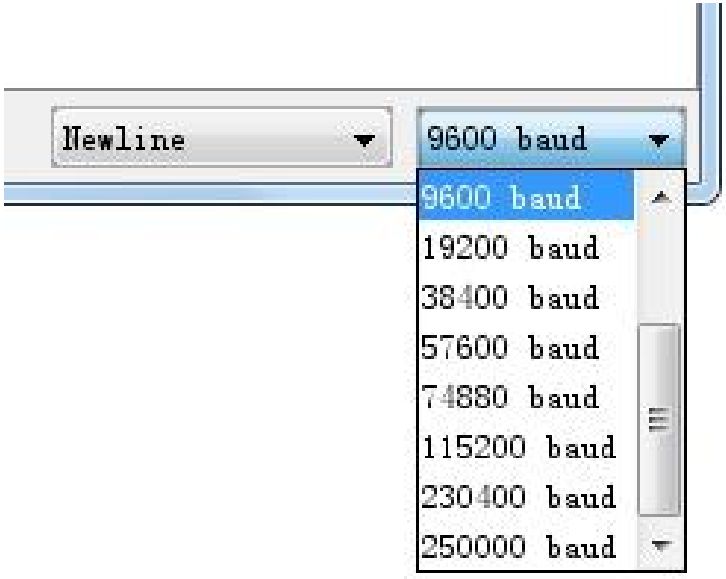


Once open, you should see something like this:



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.

Lesson 3 Blink

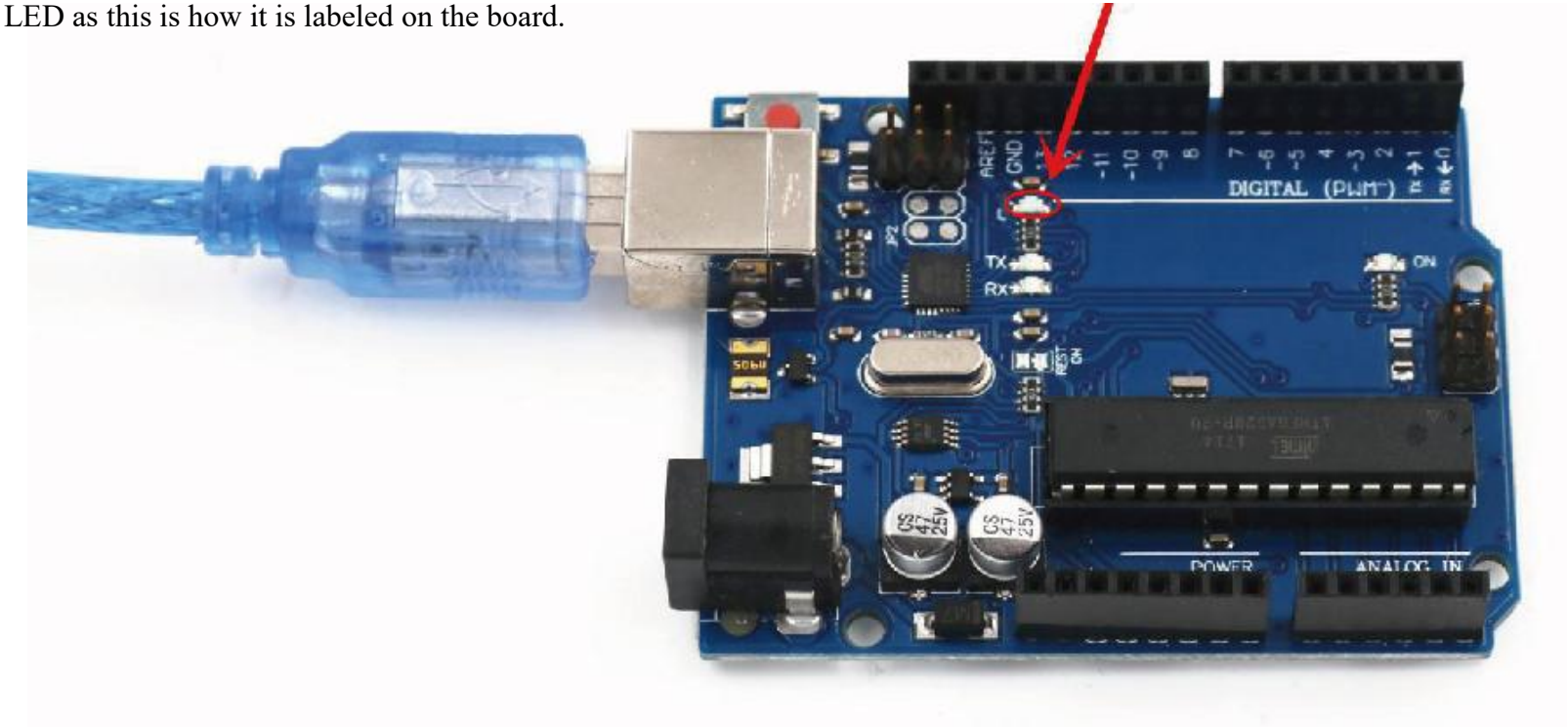
About this lesson:

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.

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.



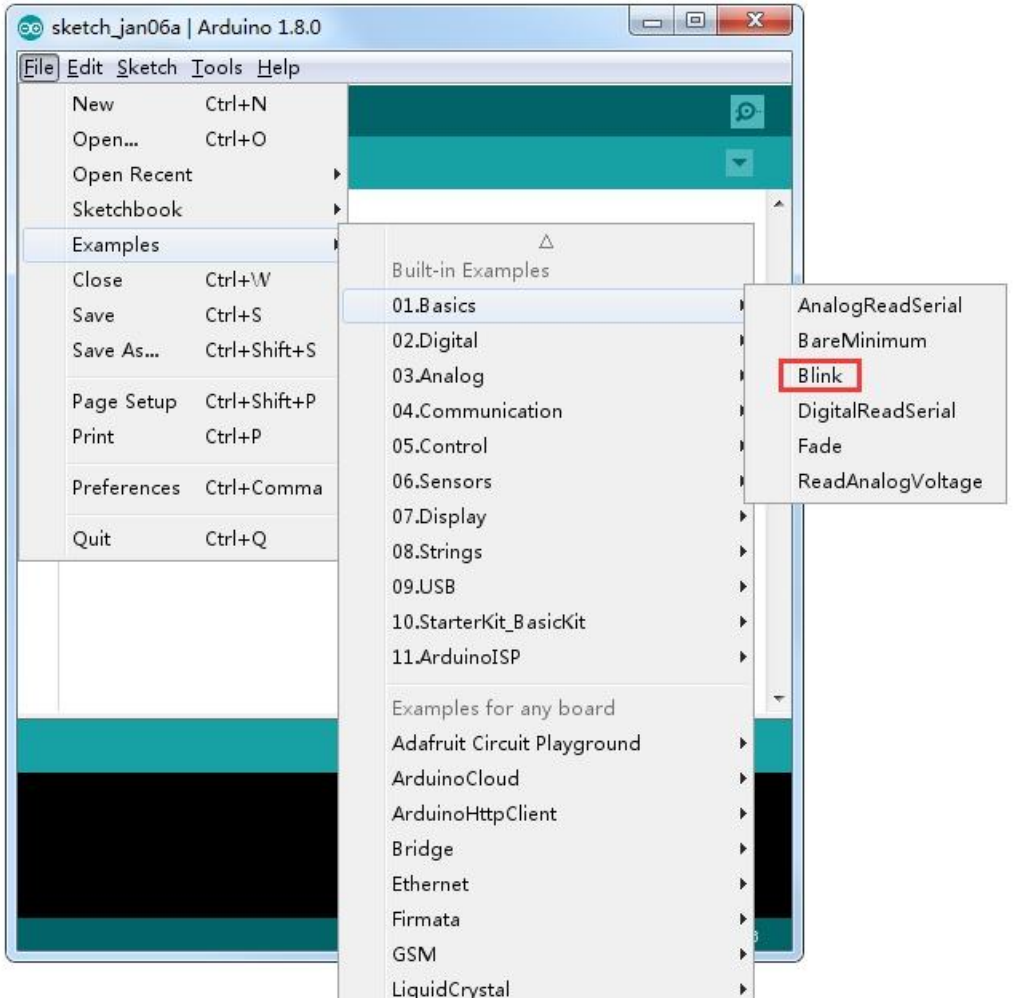
You may find that your UNO R3 board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

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

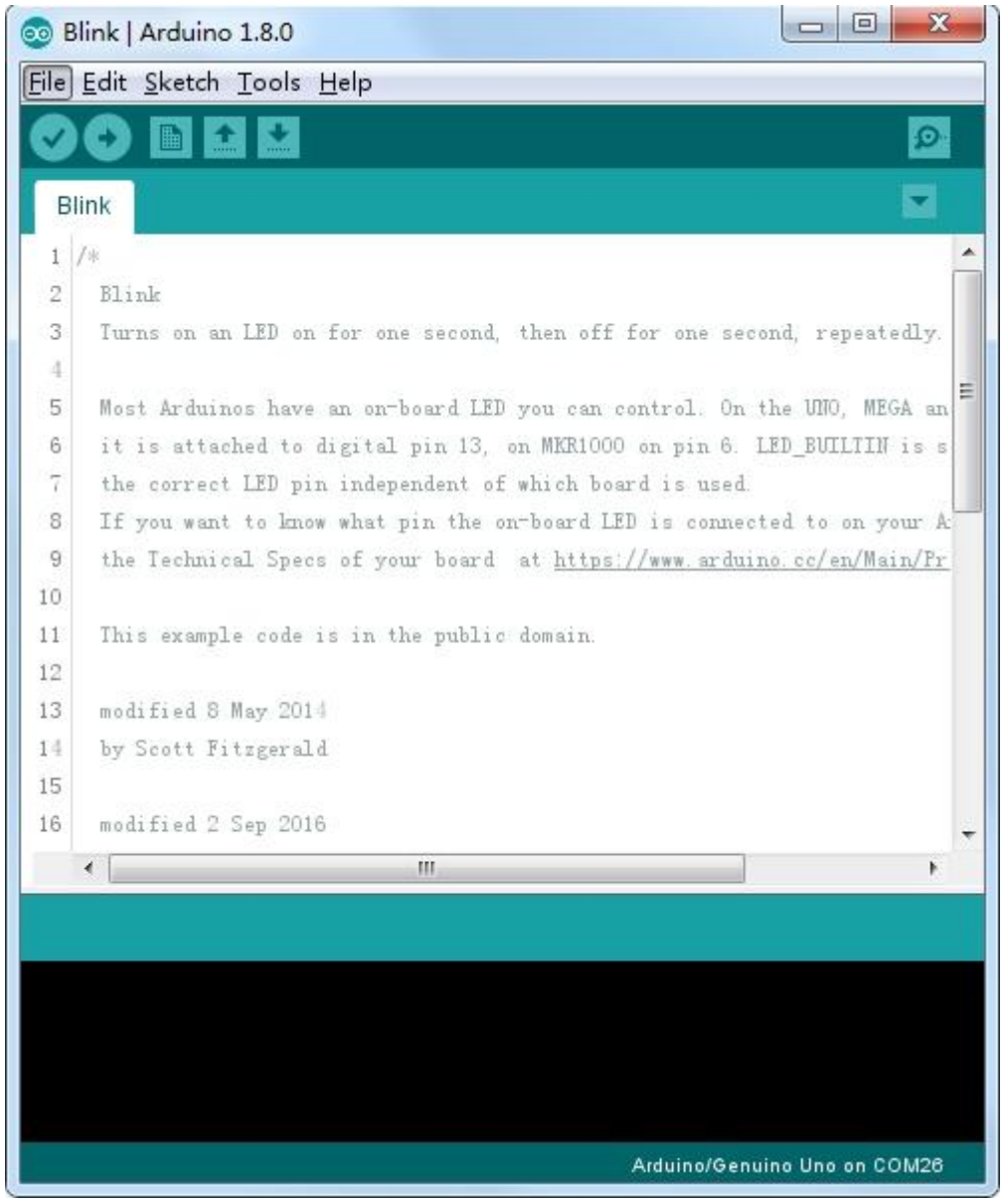
In Lesson 0, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO R3 board. The time has now come to put that connection to the test and program your UNO R3 board.

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



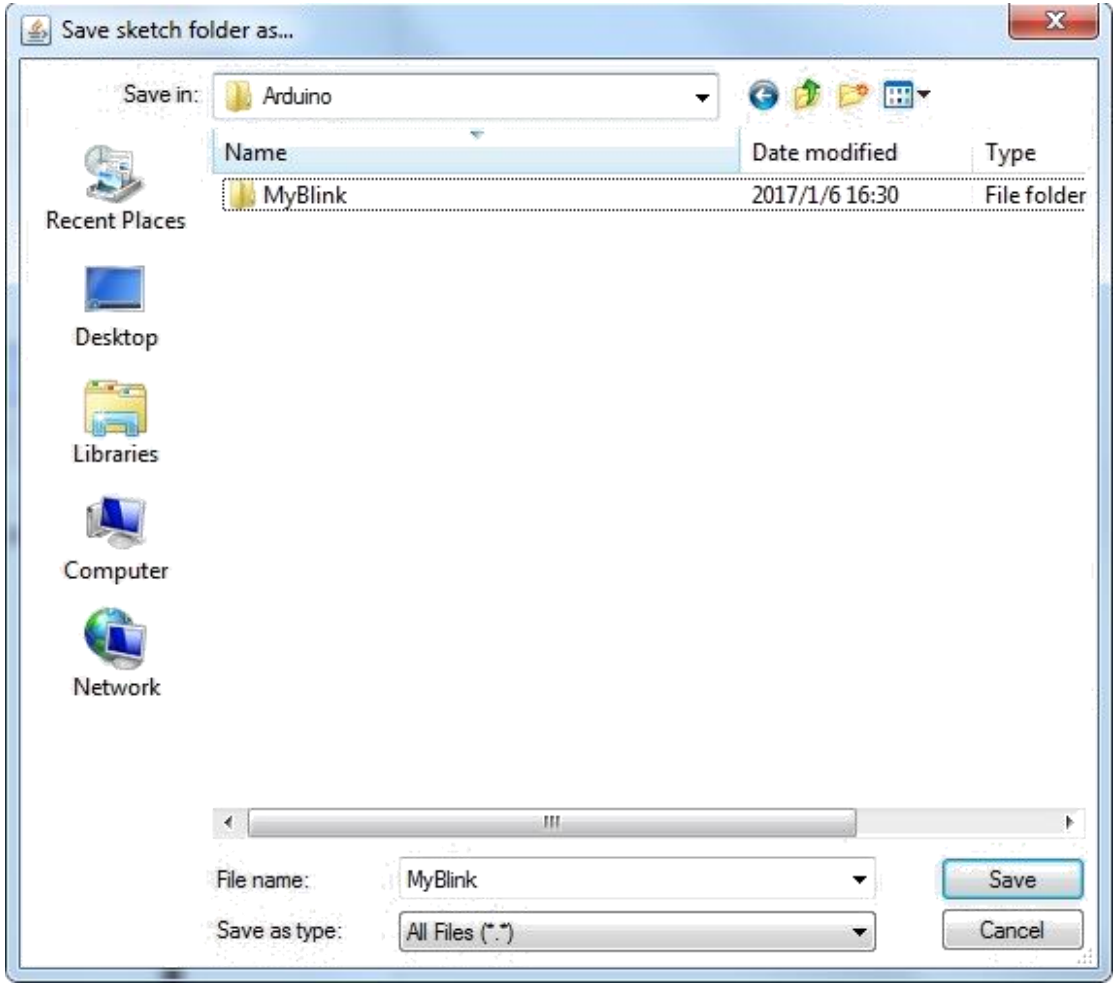
When the sketch window opens, enlarge it so that you can see the entire sketch in the window.



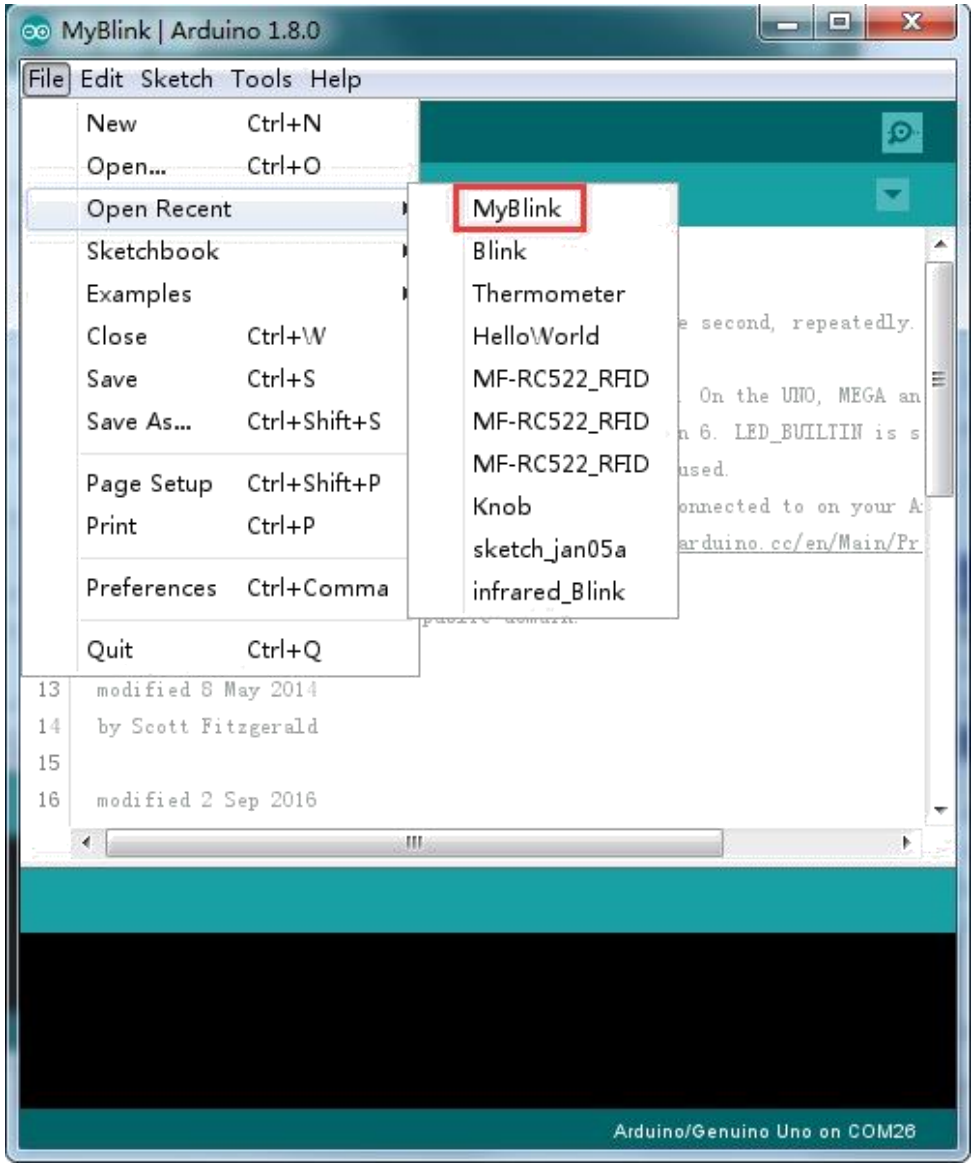
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

Since we are going to change this sketch, the first thing you need to do is save your own copy.

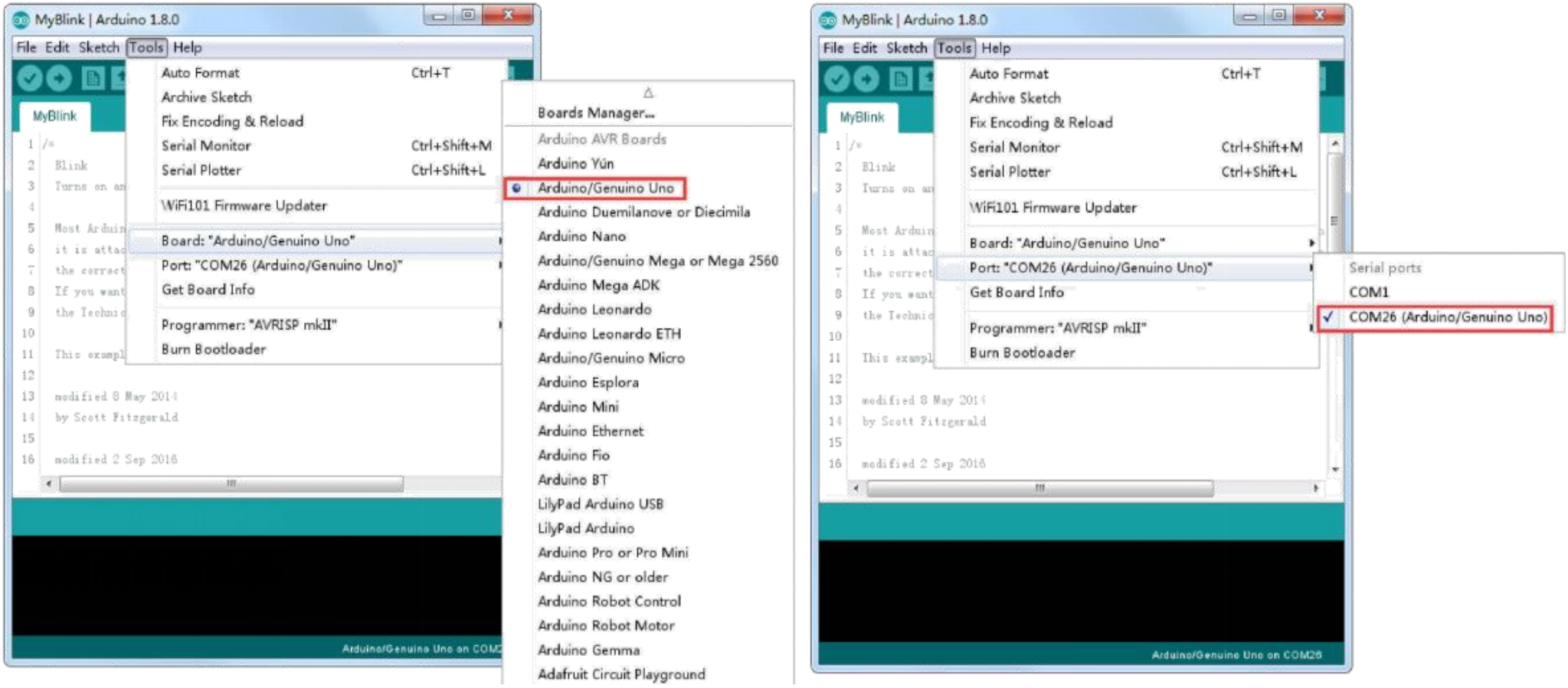
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.



You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



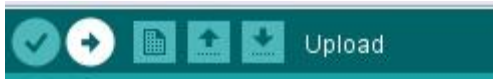
Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.



Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using 2560, then you will have to choose Mega 2560 as the Board Type, other choices can be made in the same manner. And the Serial Port displayed for everyone is different, despite COM 26 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.

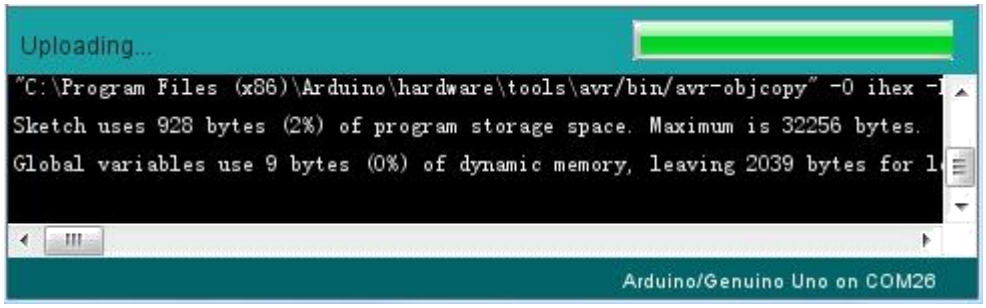
The Arduino IDE will show you the current settings for board at the bottom of the window.

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

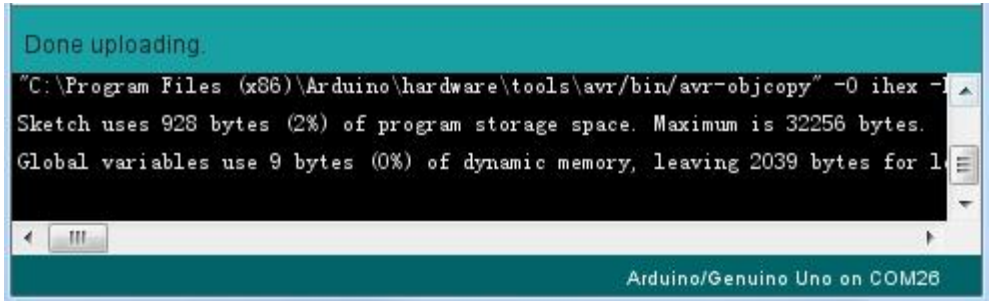


If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the staus will change to 'Done'.



The other message tells us that the sketch is using 928 bytes of the 32,256 bytes available.After the 'Compiling Sketch..' stage you could get the following error message:

It can mean that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.

If you encounter this, go back to Lesson 0 and check your installation.

Once the upload has completed, the board should restart and start blinking. Open the code

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.

gle 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 the digital pin as an output.
  pinMode(led, 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); delay(1000); digitalWrite(led, LOW);
   delay(1000);                    // turn the LED on (HIGH is the voltage
}                                  level) // 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.

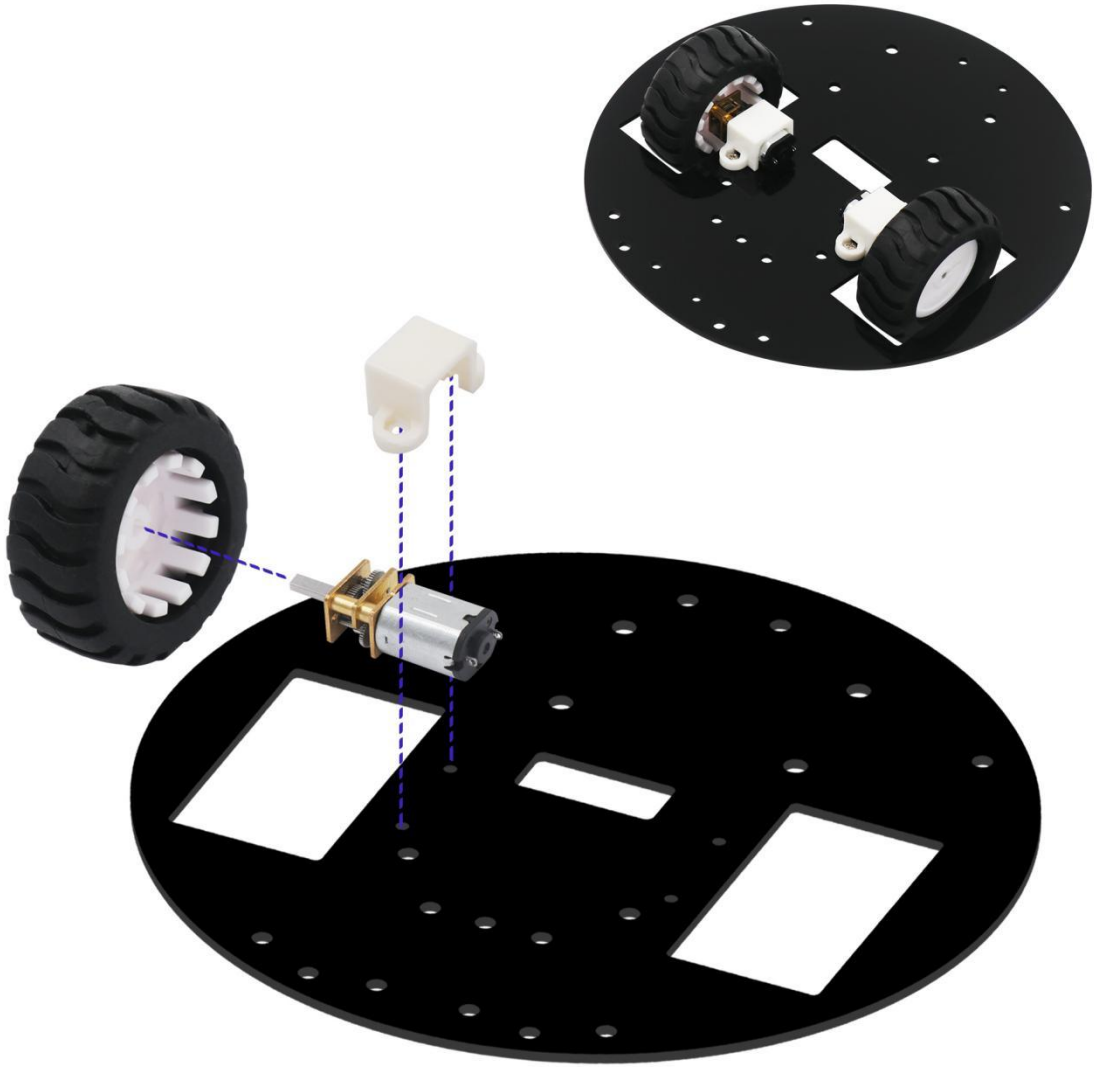
You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

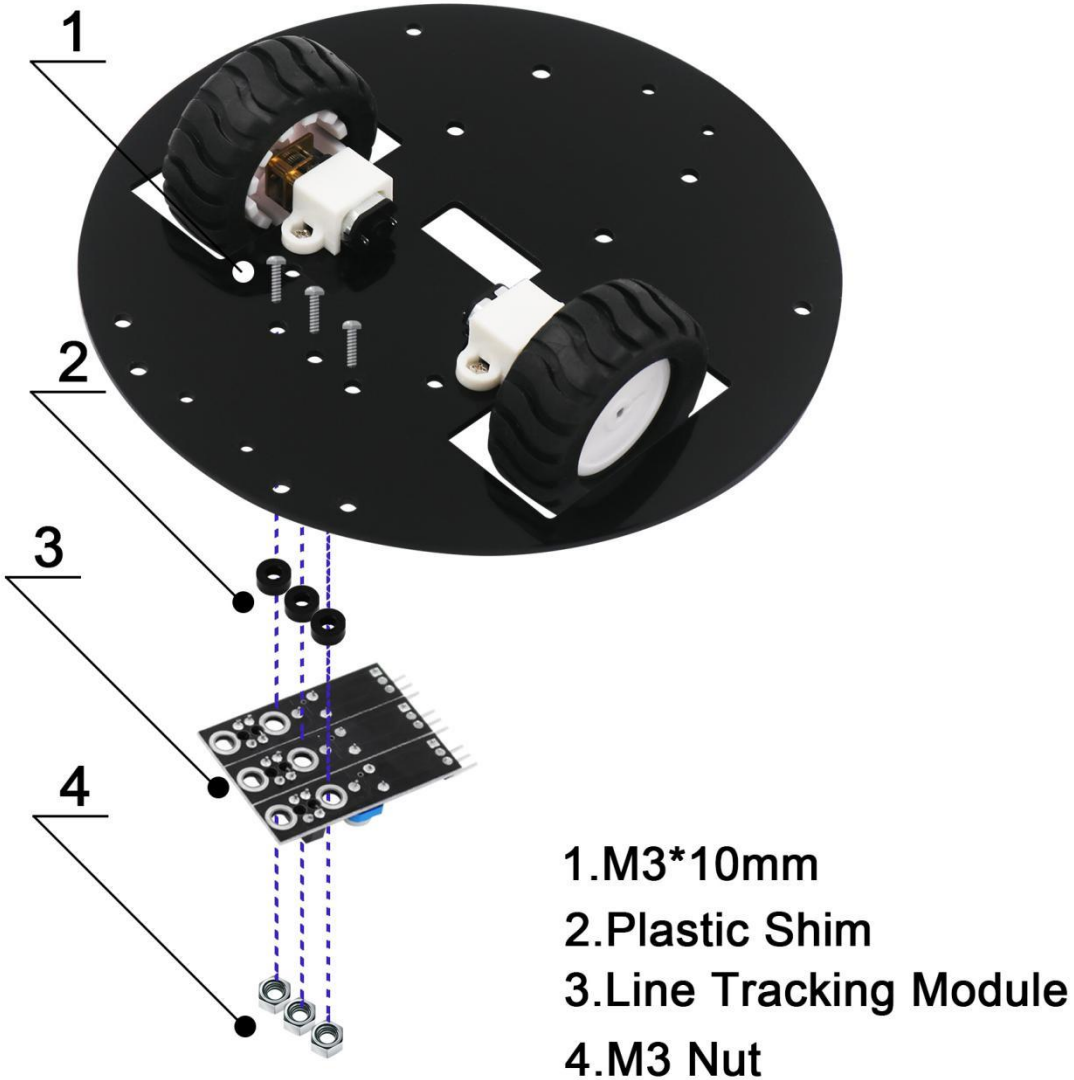
```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500);                        // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500);                        // wait for a second
36 }
```

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 each delay rather than a whole second.

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

Lesson 4 Installation Method



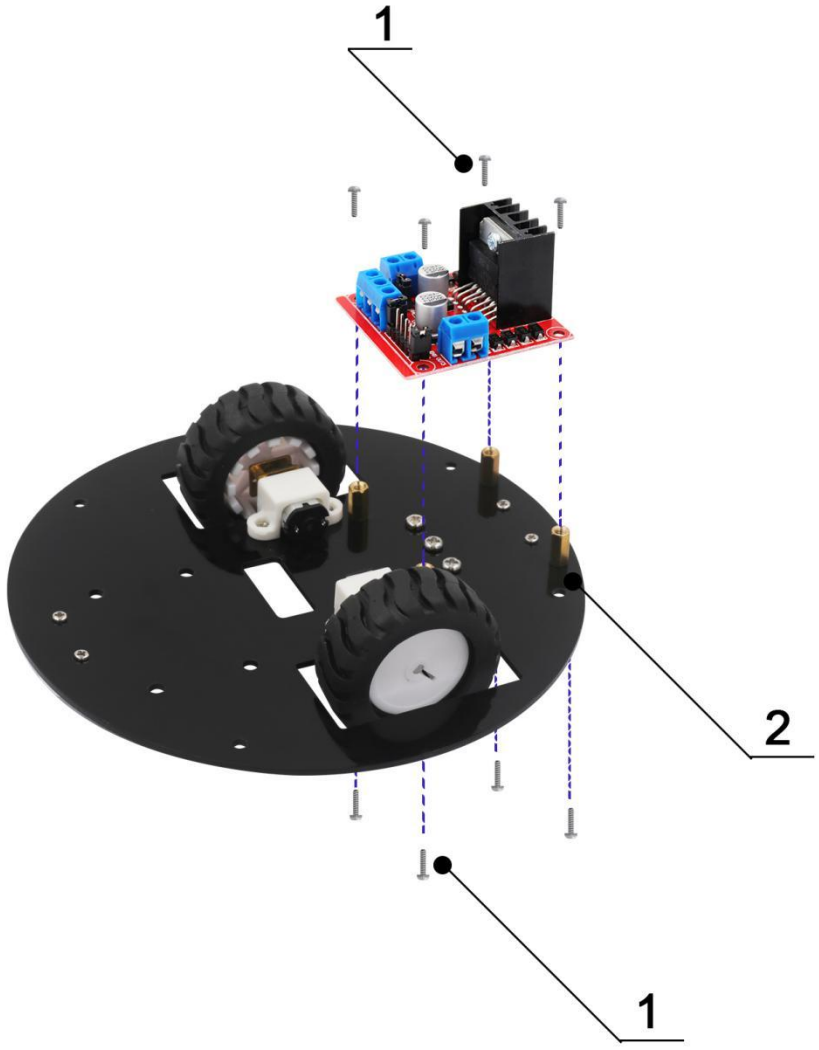


1.M3*10mm
2.Plastic Shim
3.Line Tracking Module
4.M3 Nut

1.M2*12mm

1.M3*6mm

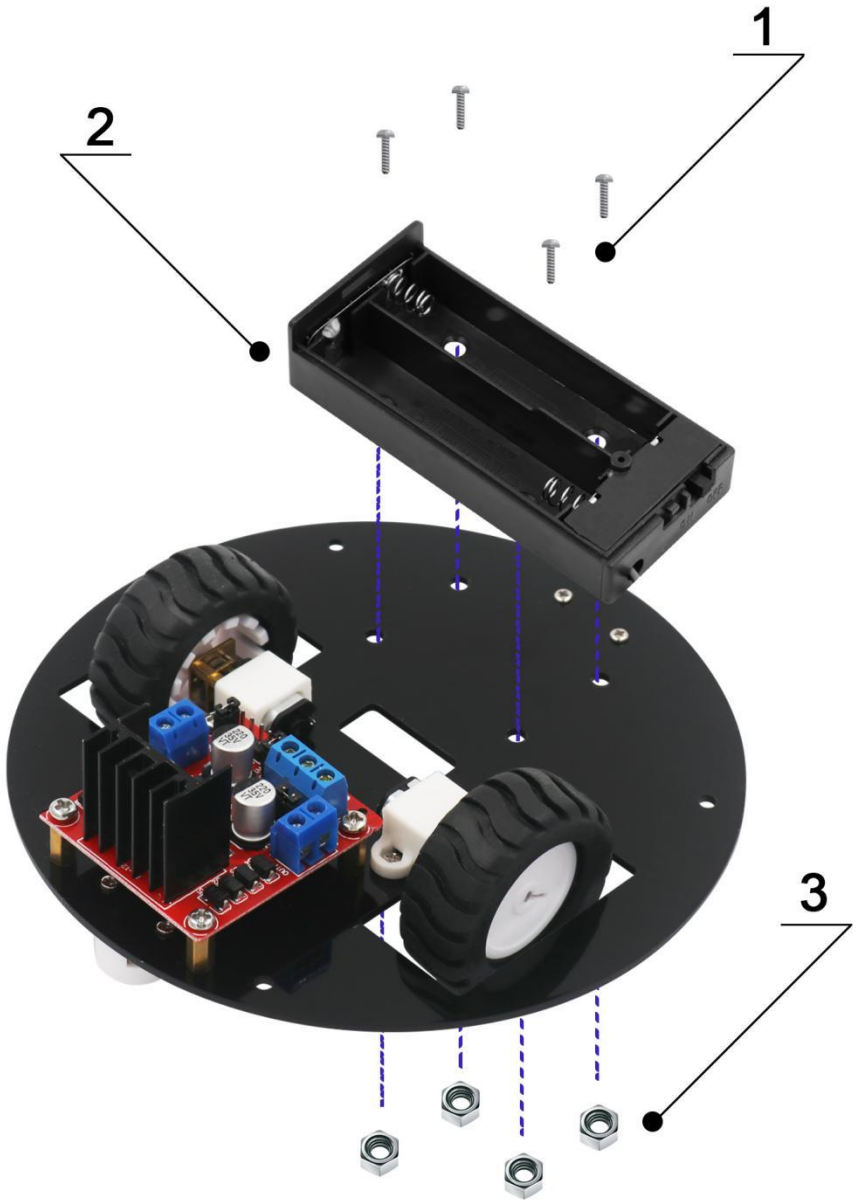
2.Copper Cylinder M3*8MM

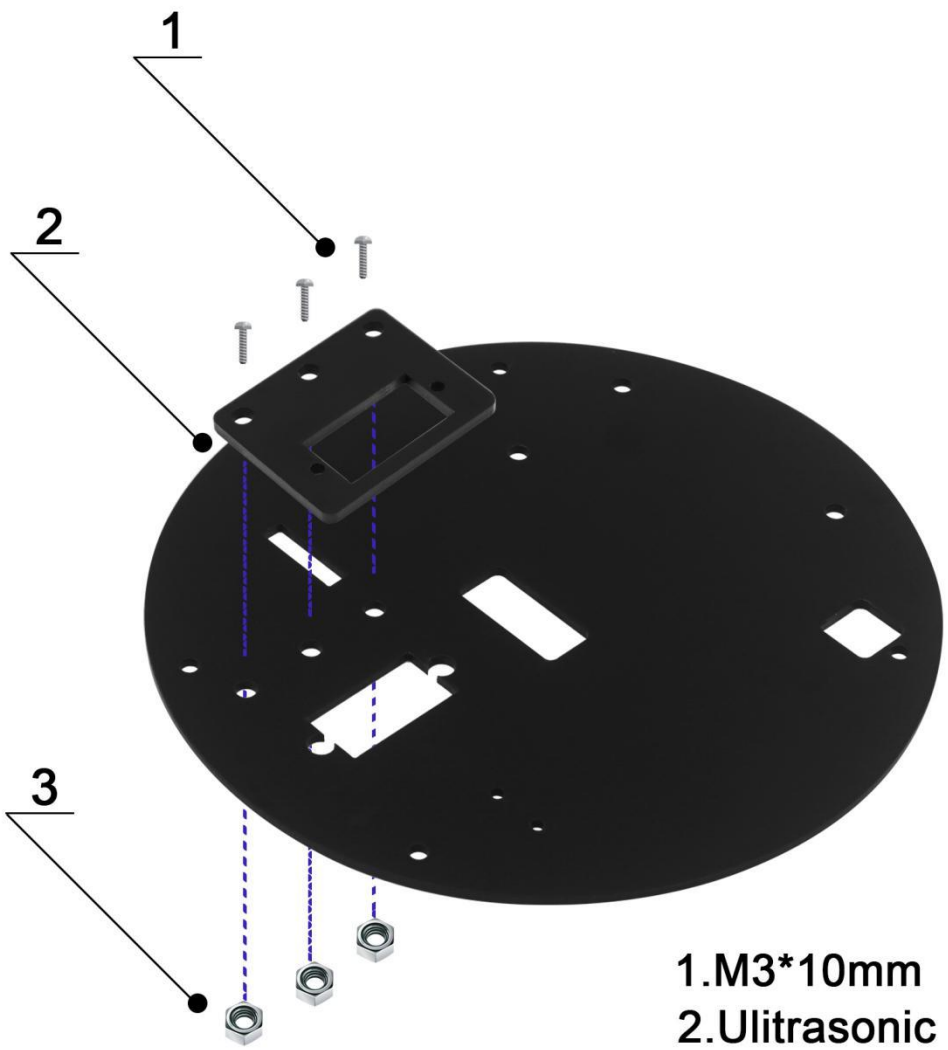


1.M3*6mm

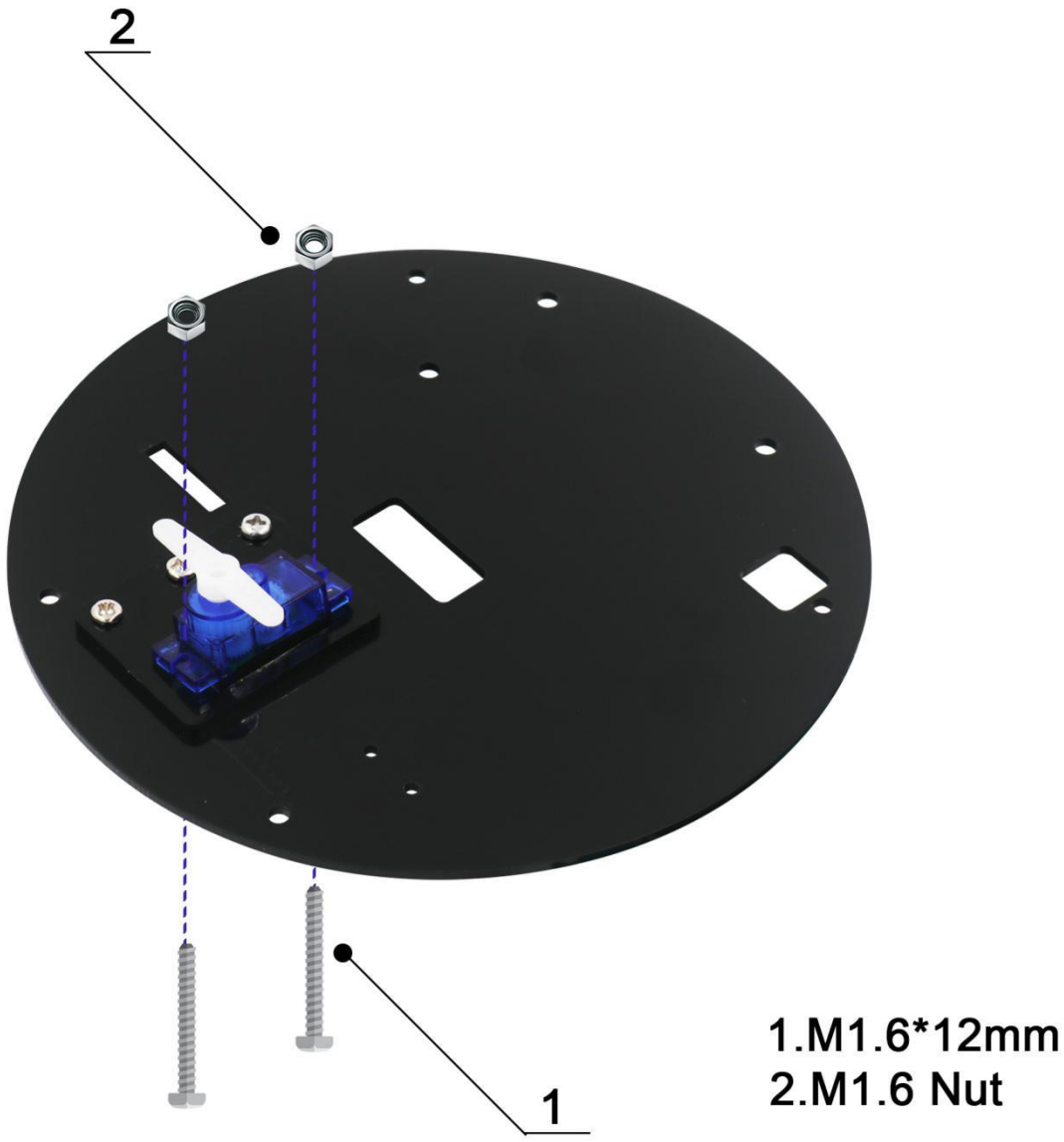
2.Cell Box

3.M3 Nut

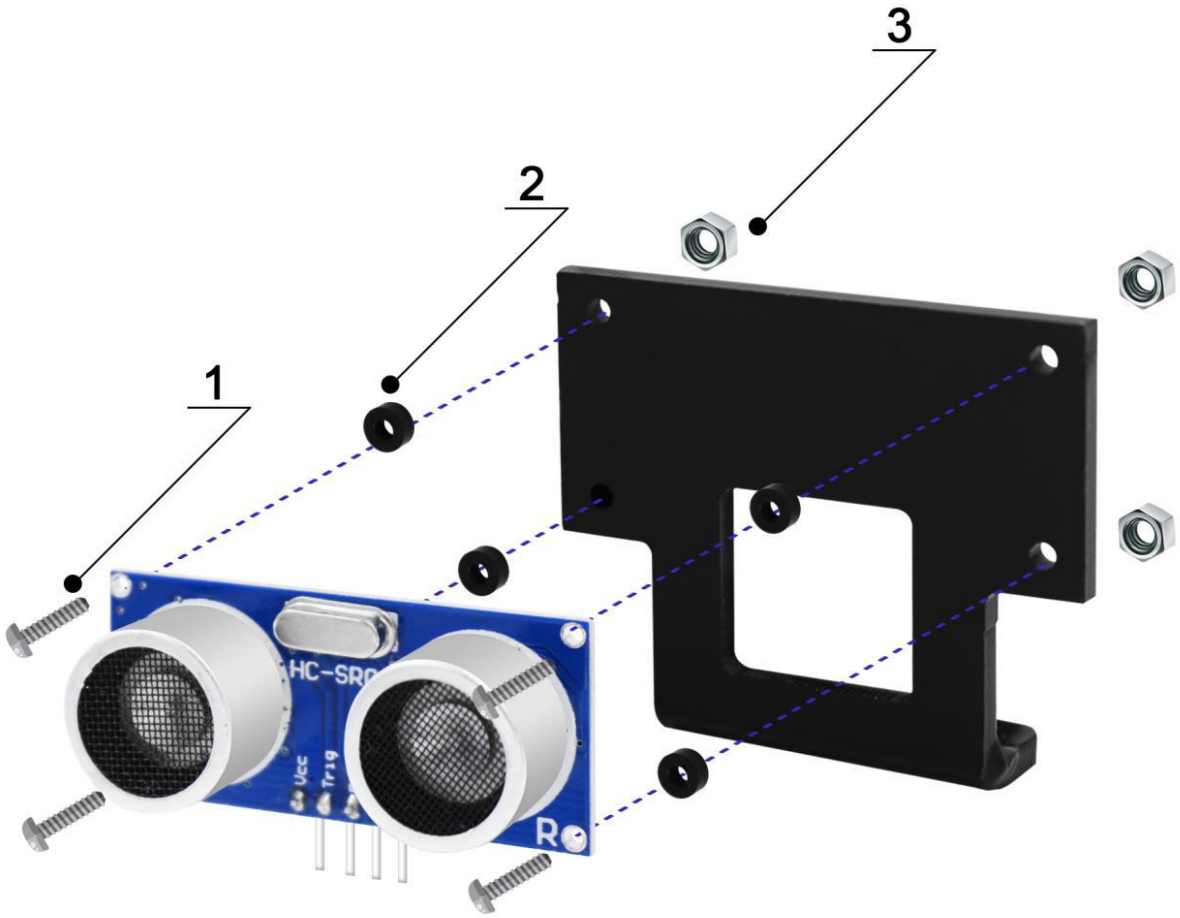




1.M3*10mm
2.Ulitrasonic Holder
3.M3 Nut



1.M1.6*12mm
2.M1.6 Nut



1.M1.6*12mm

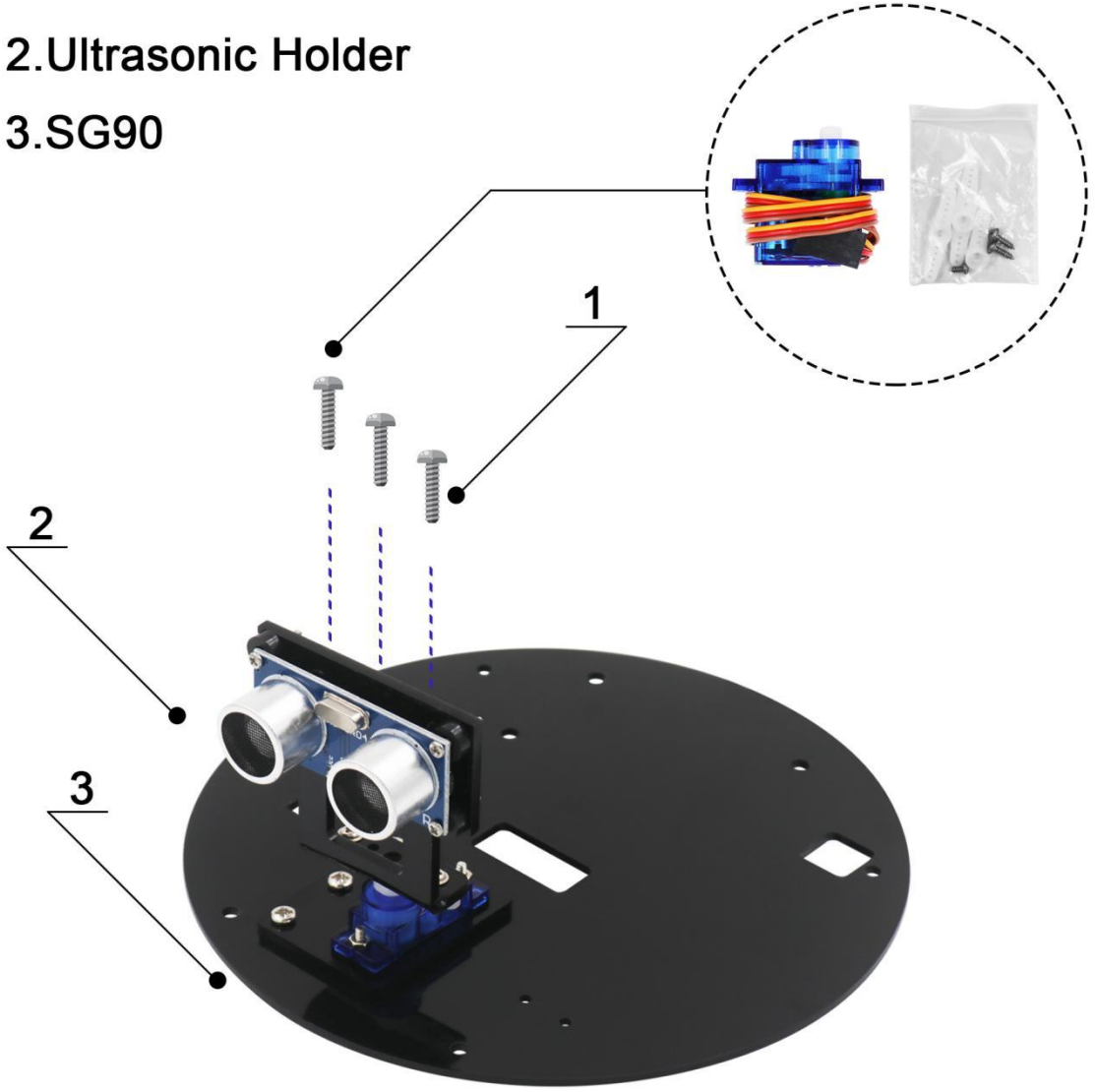
2.Plastic Shim

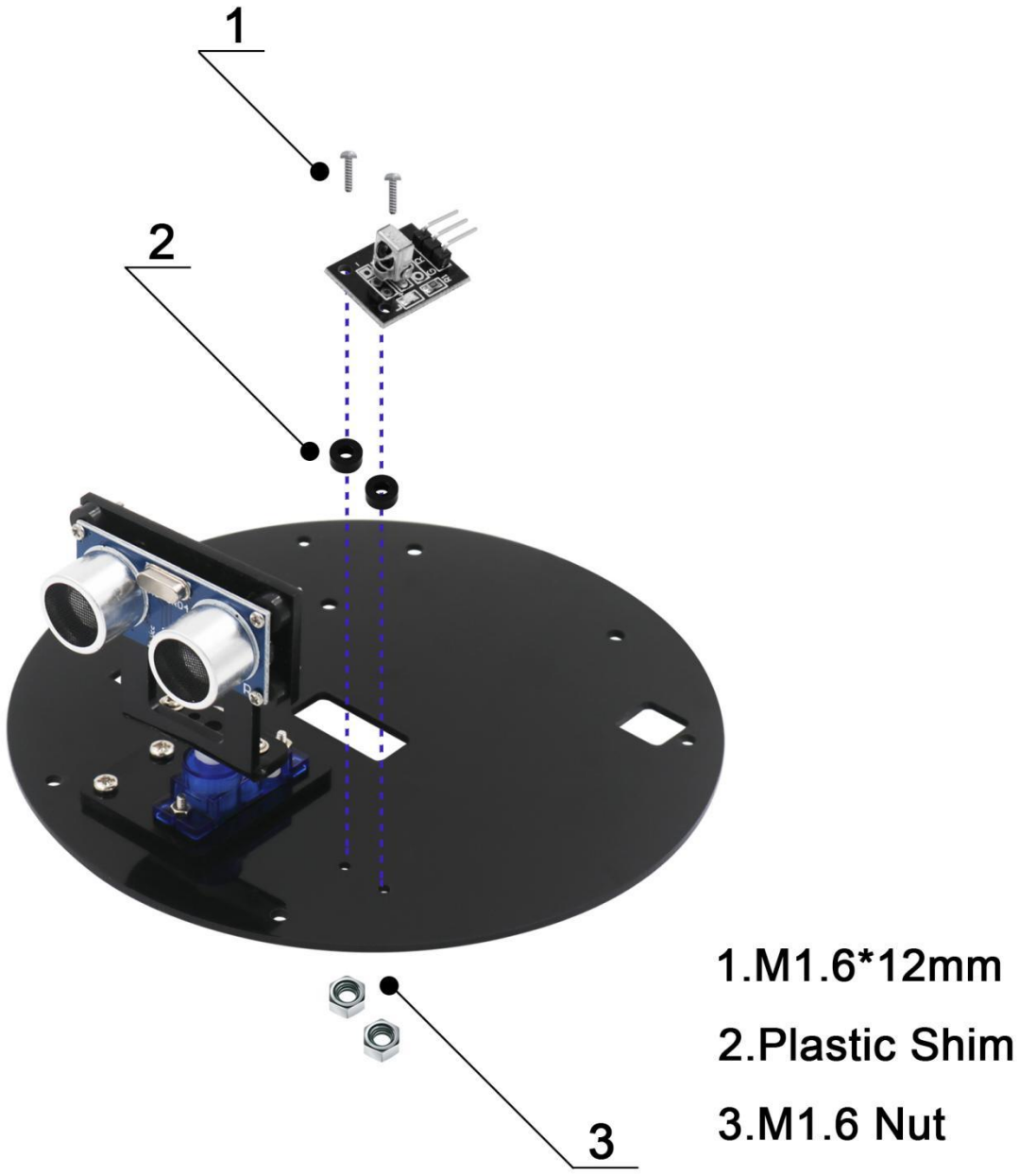
3.M1.6 Nut

1.SG90 Screws

2.Ultrasonic Holder

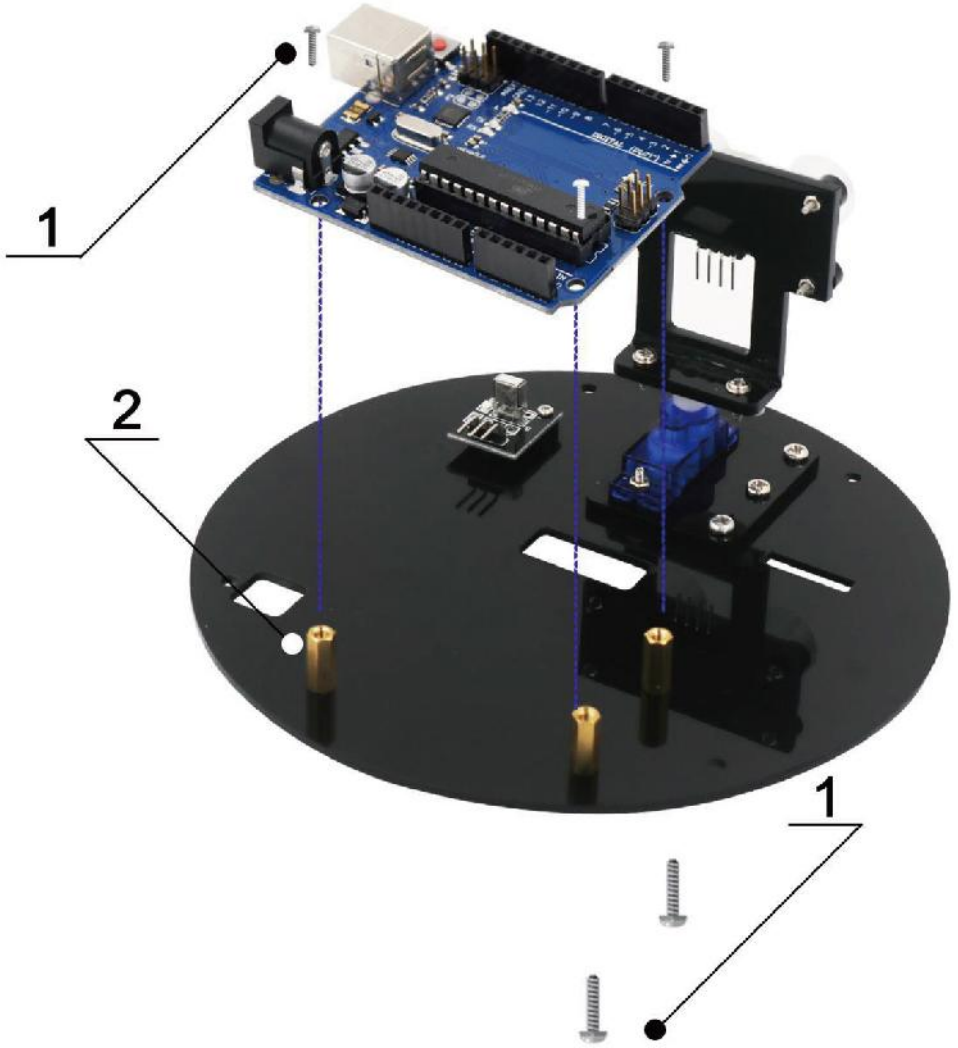
3.SG90




1
2
3
1.M1.6*12mm
2.Plastic Shim
3.M1.6 Nut

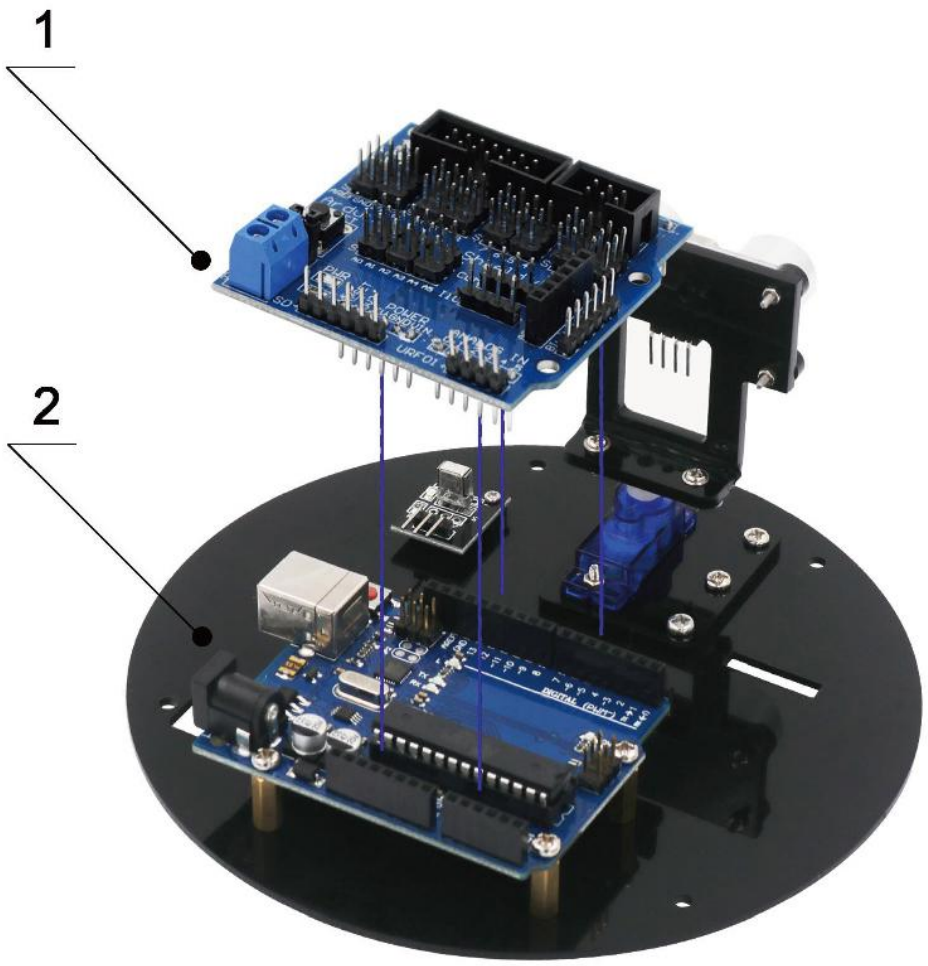
1.M3*6mm

2.Copper Cylinder M3*8mm



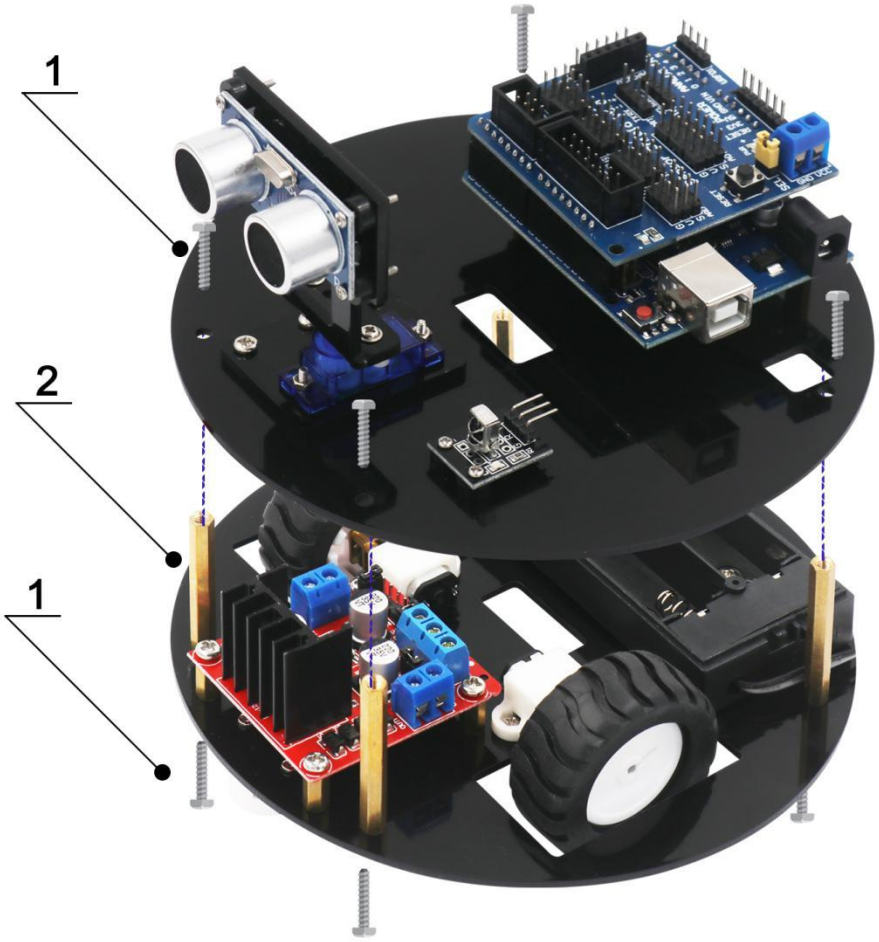
1.V5 Expansion Board

2.UNO R3



1.M3*6mm

2.Copper Cylinder M3*40mm



Lesson 5 Servo

About this lesson:

In this lesson, you will learn how to control a servo motor using LAFVIN UNO R3.

The servo motor has three leads. The color of the leads varies between servo motors, but the red lead is always 5V and GND will either be brown.The red one is the power wire and should be connected to the 5v port and this is usually orange. This control lead is connected to digital pin 9.

Introduction

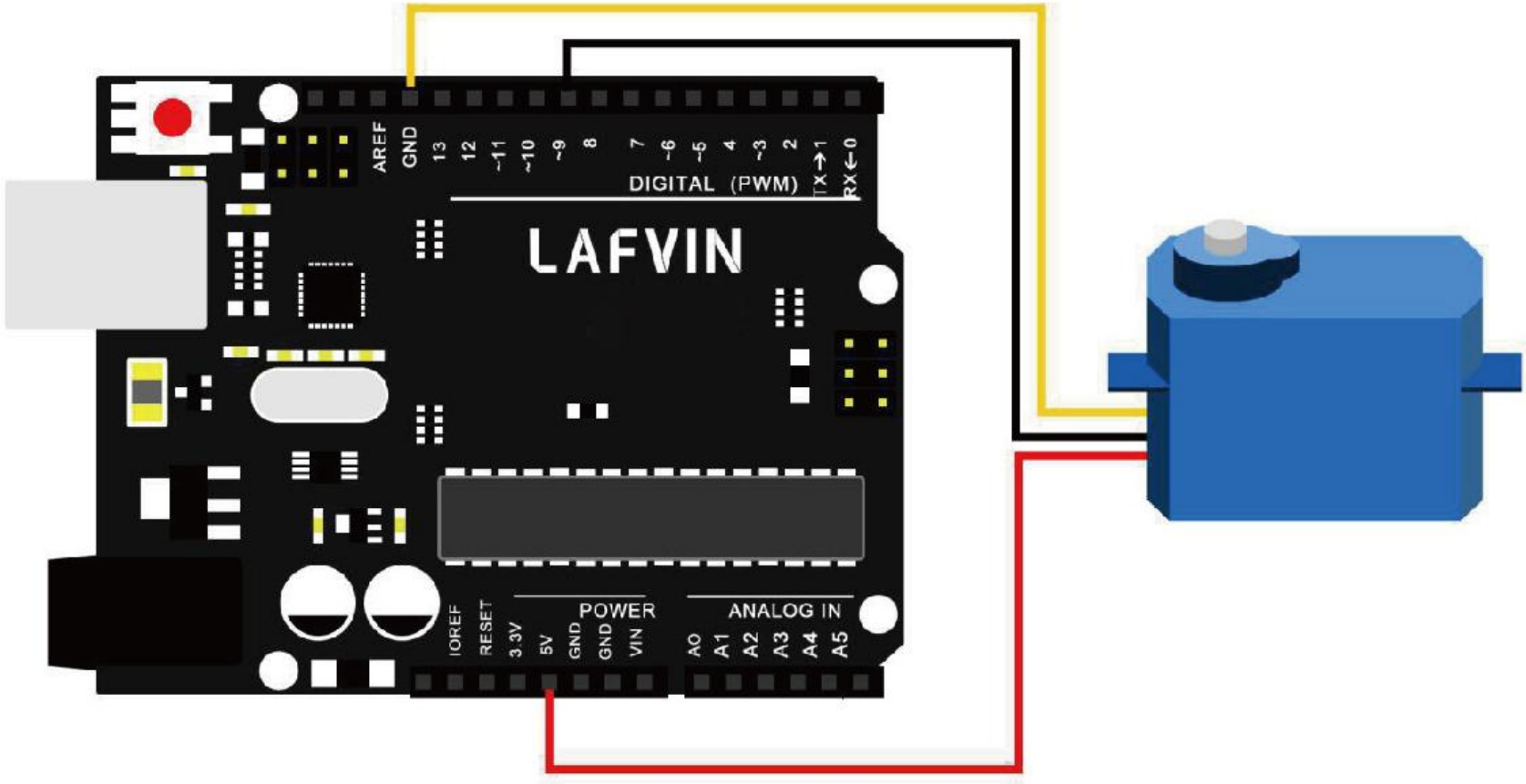
Servo motors are great devices that can turn to a specified position.

Usually, they have a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position and it will go there. As simple as that!

Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the flaps on a RC plane. With time, they found their uses in robotics, automation, and of course, the Arduino world.

There are two ways to control a servomotor with Arduino. One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor. Another way is to directly use the Servo function of the Arduino to control the motor. In this way, the program will be easier but it can only control two-contact motor because for the servo function, only digital pin 9 ang 10 can be used. The Arduino drive capacity is limited. So if you need to control more than one motor, you will need external power.

Connection diagram



Code

After connecting,please open the the program and load up the code - Lesson 5 Servo onto your Arduino board. See Lesson 3 for details about program uploading if there are any errors.

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

For details about loading the library file, see Lesson 2.

Lesson 6 Ultrasonic Sensor Module

About this lesson:

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

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

Introduction:

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

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

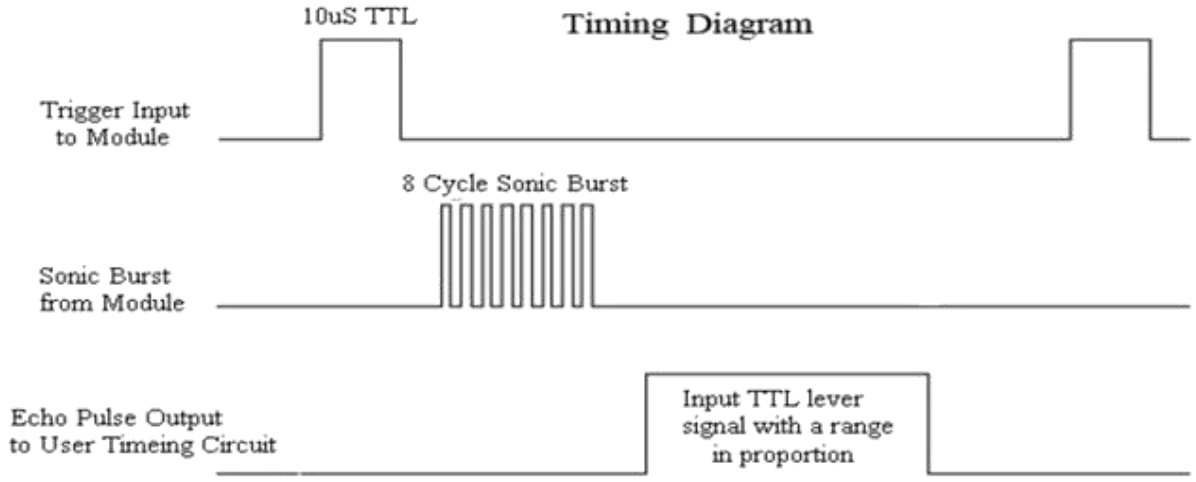
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

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

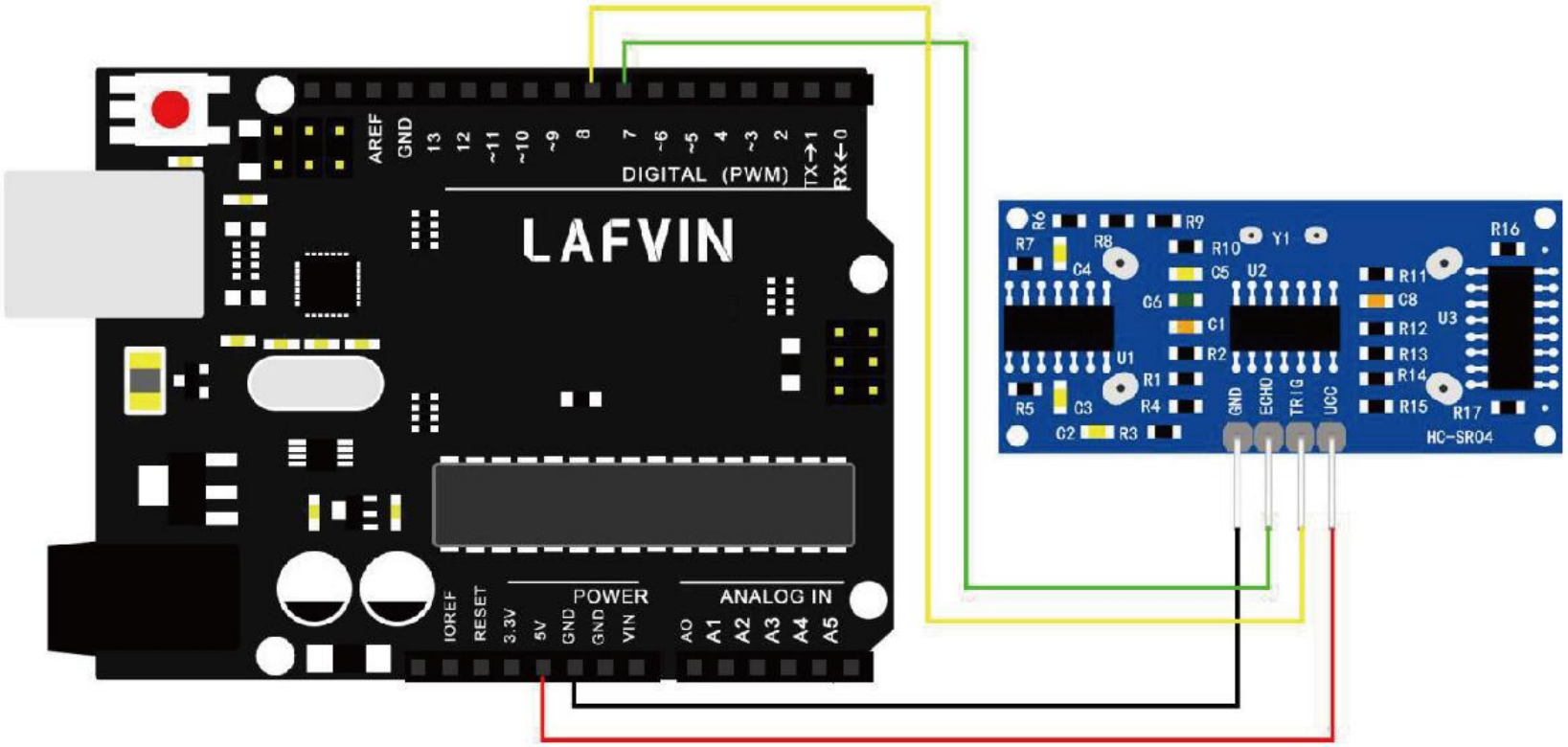
Test distance = (high level time × velocity of sound (340m/s) /2

The Timing diagram is shown below. You only need to supply a short 10us pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the

range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: us / 58 = centimeters or us / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Wiring diagram



Code

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

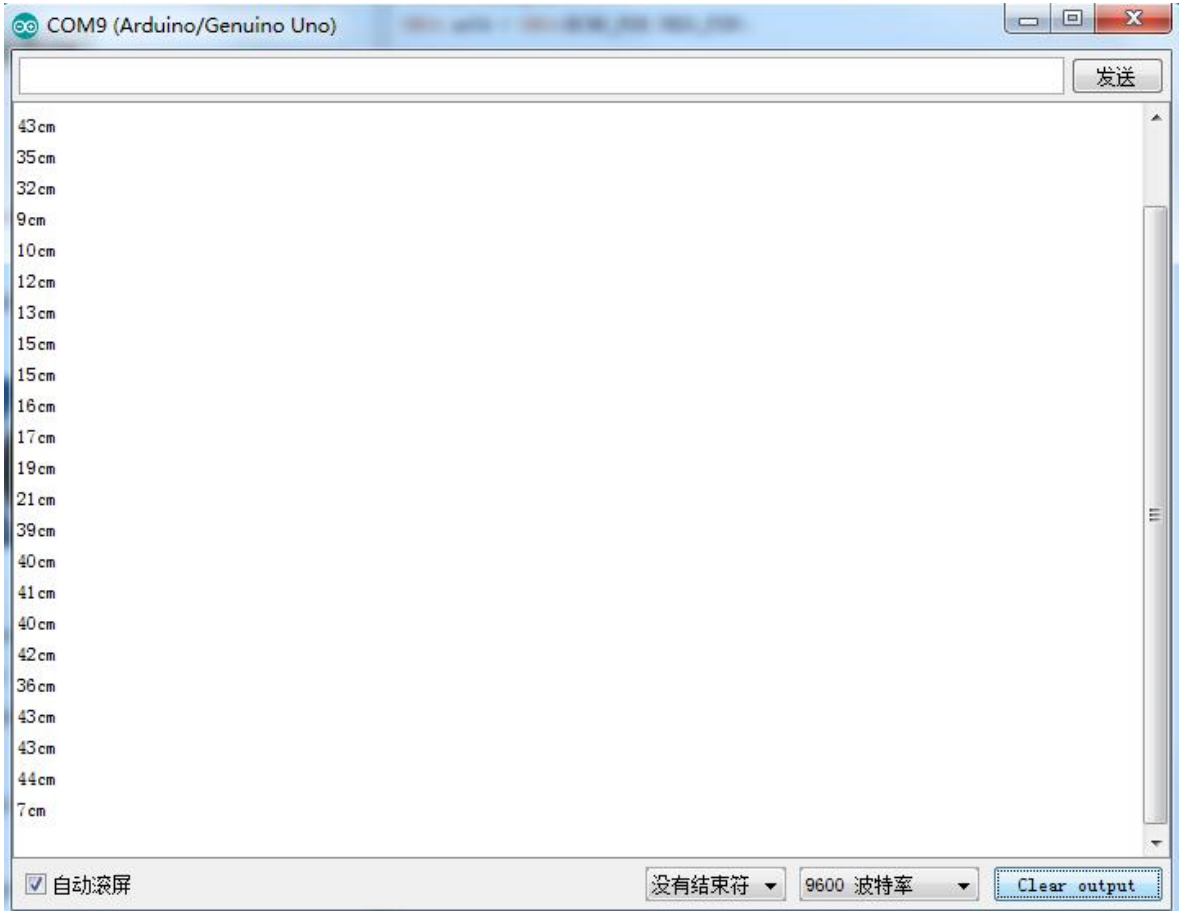
After wiring, please open the program in the code folder- Lesson 6 Ultrasonic Sensor Module and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

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

For details about loading the library file, see Lesson 2.

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

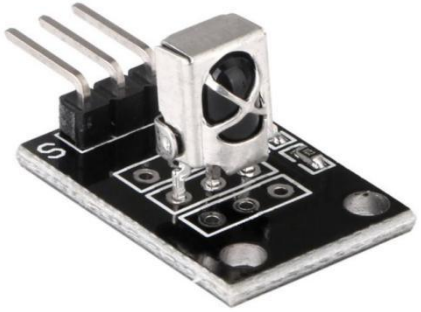
Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in Lesson 2.



Lesson 7 IR Receiver Module

About this lesson:

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.

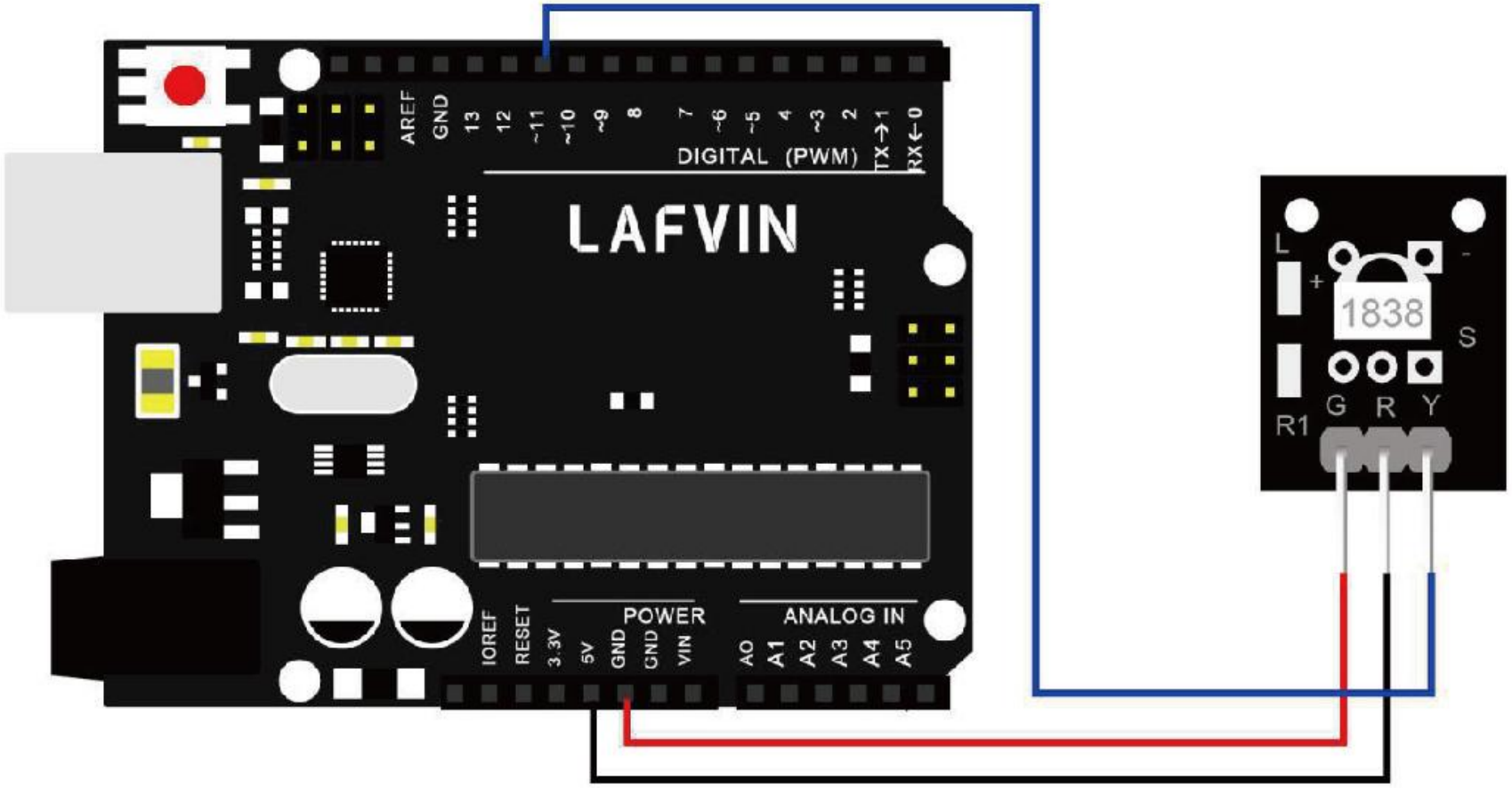


Introduction

IR is widely used in remote control. With this IR receiver, Arduino project is able to receive command from any IR remoter controller if you have the right decoder. Well, it will be also easy to make your own IR controller using IR transmitter.

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.

Wiring diagram



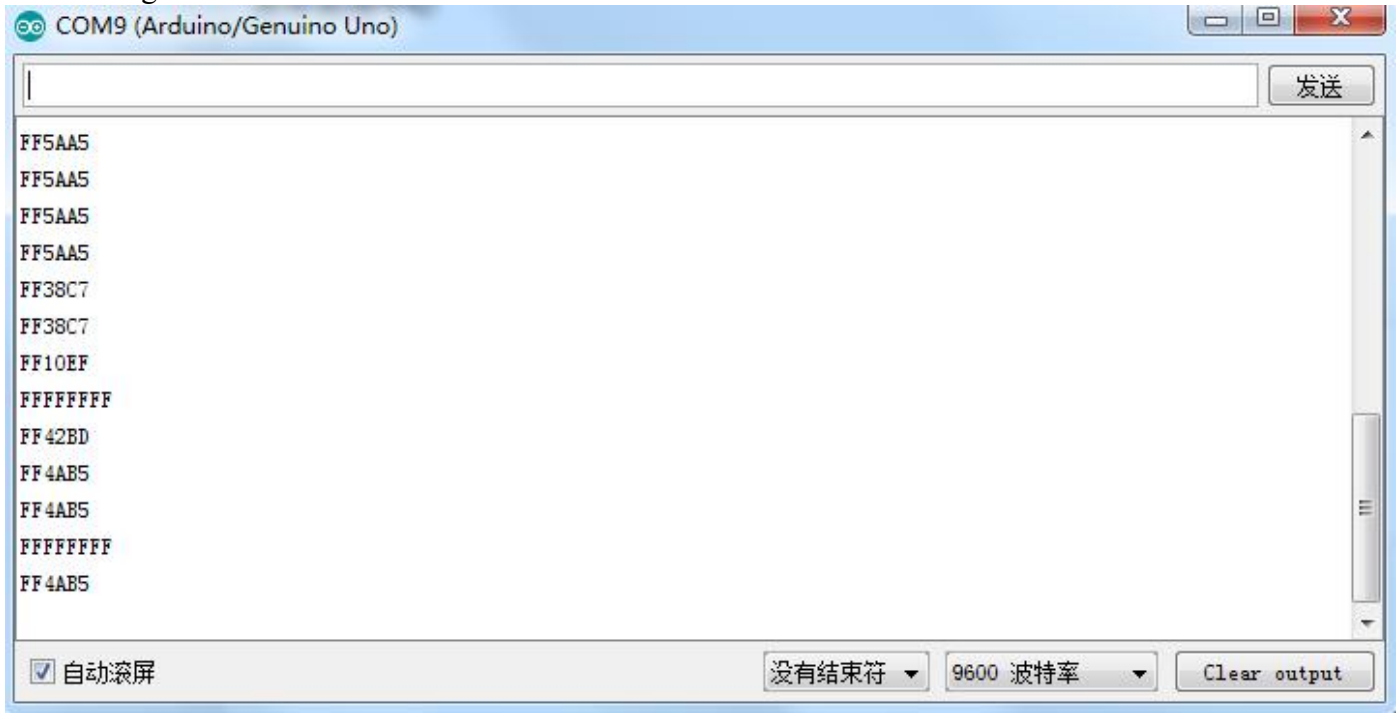
Code

After wiring, please open the program in the code folder- Lesson 7 IR Receiver Module and click UPLOAD to upload the program. See Lesson3 for details about program uploading if there are any errors.

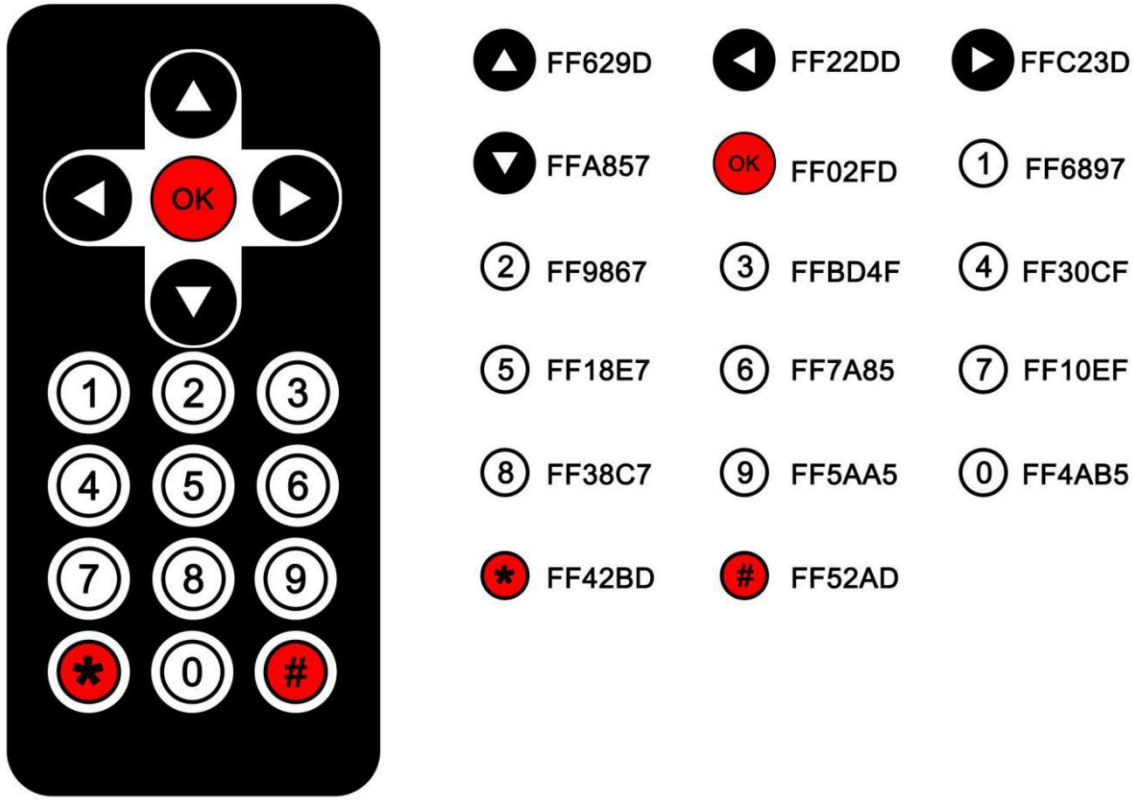
Before you can run this, make sure that you have installed the < IRremote > library or re-install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson 3.

In this lessson, we need to use a IR remote control which has 17 functional key and its launching distance is 8 meters at most, proper to control various devices indoors. This project is actually to decode remote control signal. After connection and uploading codes, aim at IR receiving module and press the key, finally you can see corresponding codes. If you press the key too long, it will show messy codes easily as shown in bellow figure.



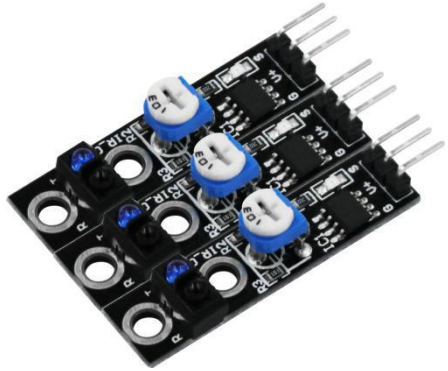
Remote control code:



Lesson 8 Tracking Sensor

About this lesson:

In this lesson, you will learn how to use a Tracking Sensor.we will use an obstacle avoidance sensor module and an LED attached to pin 13 of the LAFVIN Uno board to build a simple circuit to make a tracking light.



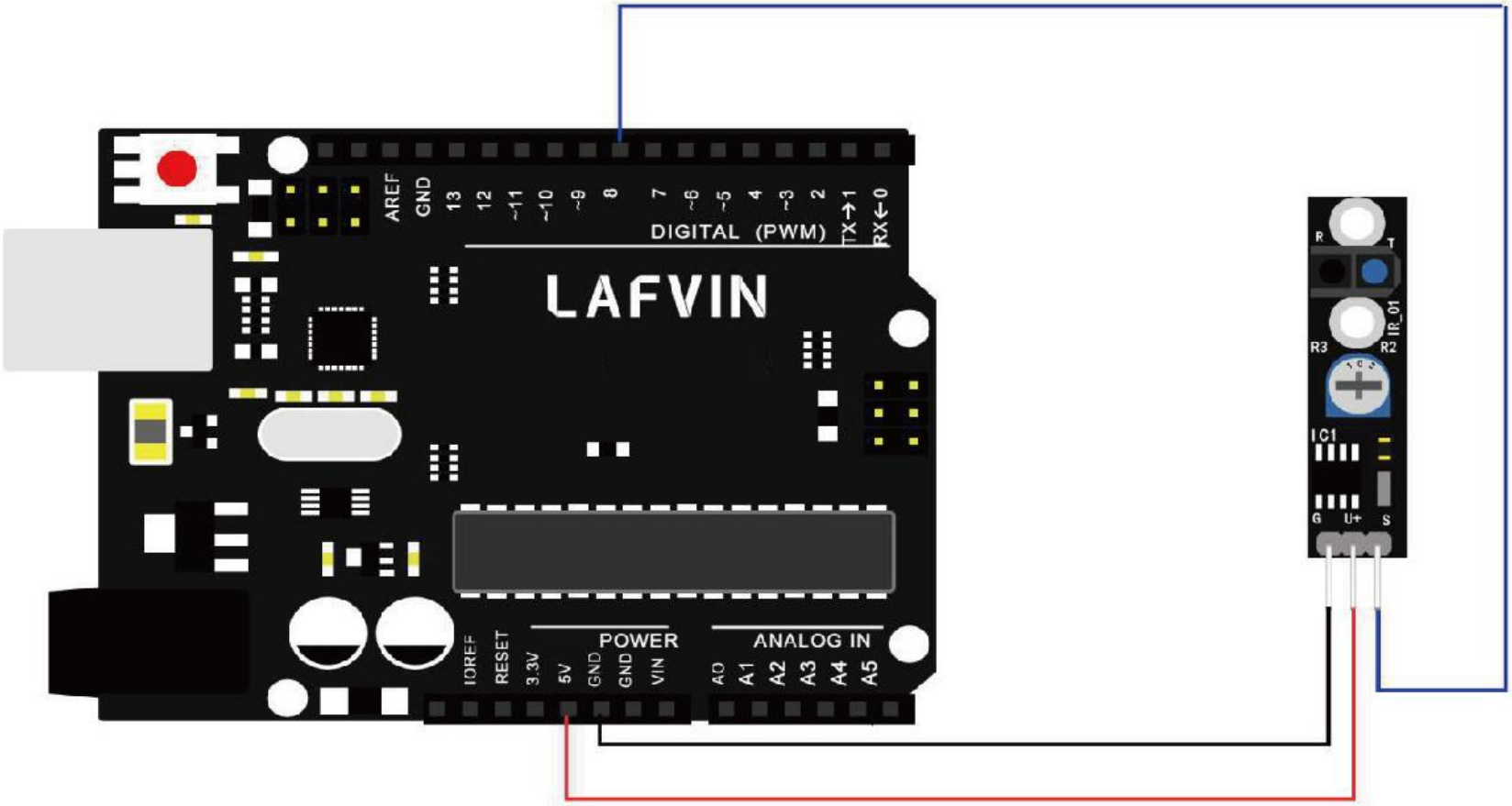
Component Introduction

This Line Tracking Sensor can detect white lines in black and black lines in white. The single line-tracking signal provides a stable output signal TTL for a more accurate and more stable line. Multi-channel option can be easily achieved by installing required line-tracking robot sensors.

Specification:

Power Supply: +5V Operating Current: <10mA
Operating Temperature Range: 0°C ~ + 50 °C
Output Interface: 3-wire interface (1 - signal, 2 - power, 3 - power supply negative) Output Level: TTL level

Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson 8 Tracking Sensor and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

When the infrared transmitter emits rays to a piece of paper, if the rays shine on a white surface, they will be reflected and received by the receiver, and pin S will output low level; If the rays encounter black lines, they will be absorbed, thus the receiver gets nothing, and pin S will output high level.

Since an LED has been attached to pin 13, connect the pin out to D8 of the LAFVIN Uno board. When the tracking sensor detects reflection signals (white), the LED will be on. Otherwise, it will be off (black line).

Lesson 9 Bluetooth Module

About this lesson:

In this lesson,we will learn how to use the Bluetooth Module.

Introduction:

The HC06 is a Serial port Bluetooth module which having fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The HC-06 Bluetooth module to LAFVIN UNO R3:

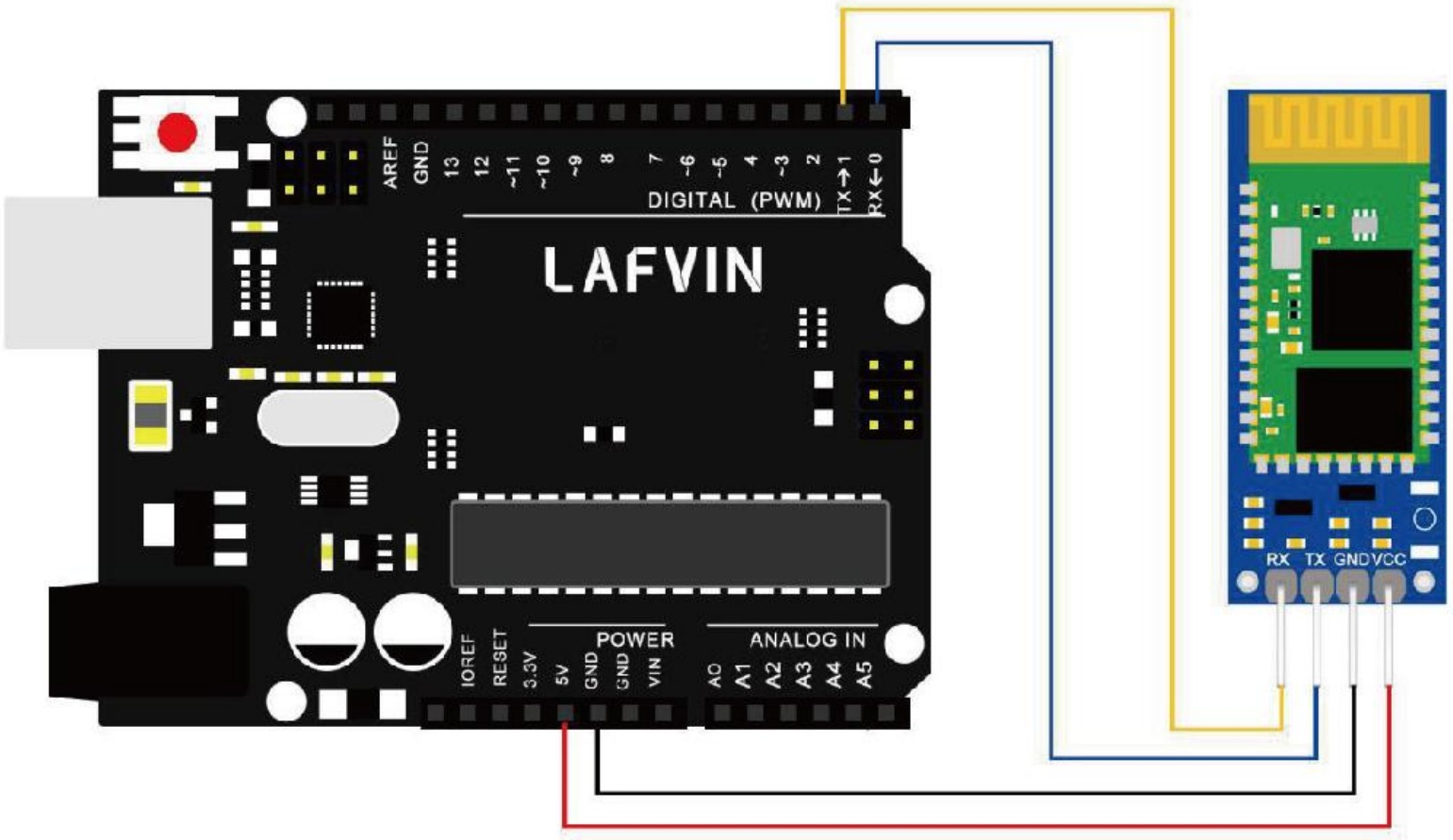
VCC>>>>5v

GND>>>>GND

TXD>>>>D0

RXD>>>>D1

Wiring diagram

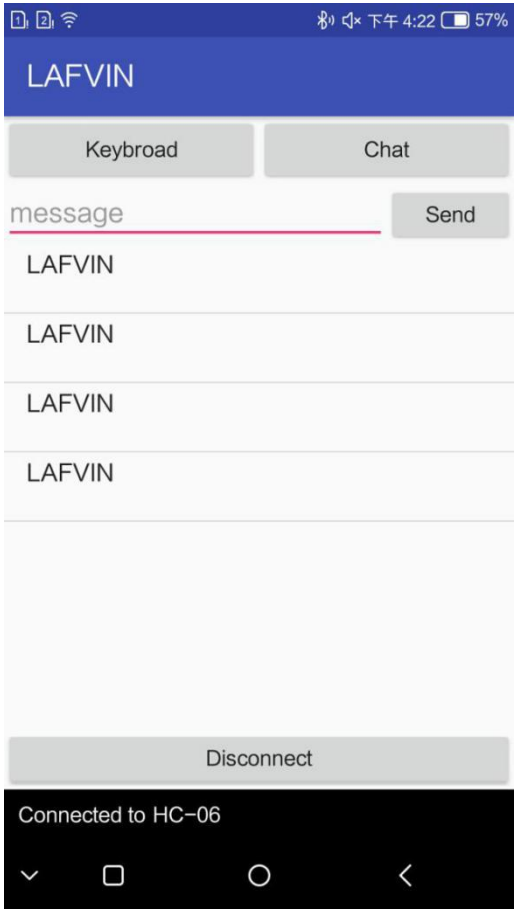


Code

After wiring, please open the program in the code folder- Lesson 9 Bluetooth Module and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

Atterntion:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.

Download LAFVIN Bluetooth APP or other APP—Bluetooth terminal application.After power-on, power indicator D1 is on. Pair your phone with HC-06. for doing this go to Settings->Bluetooth->Scan device->select HC-06 and pair it. Pass code to pair is '1234'.Open Bluetooth Terminal software, go to options and select 'connect a device - secure' option. It it ask for pass code enter 1234.
If you send 'L' it will show 'LAFVIN' as shown in Figure.



Lesson 10 L298N Motor Driver

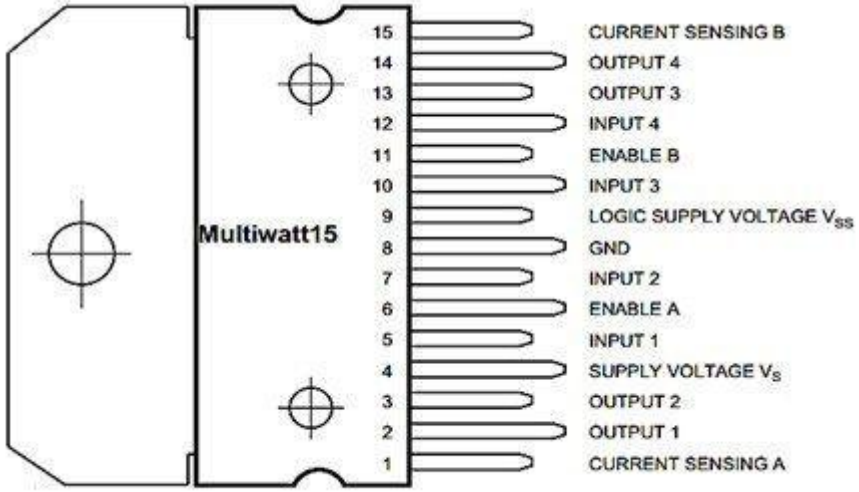
About this lesson:

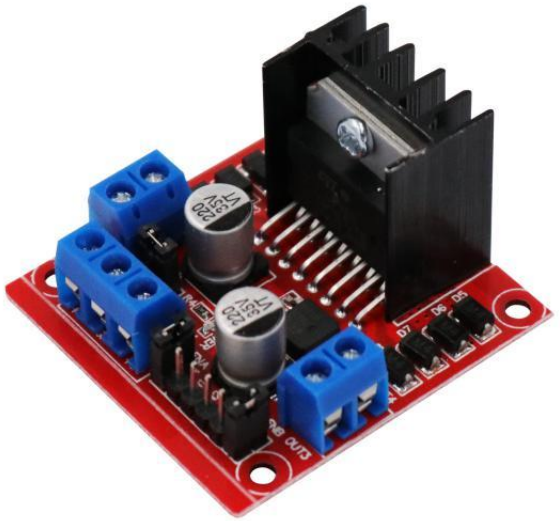
In this lesson, you will learn how to use a water level detection sensor module.

Component Introduction

The L298N actually contains two complete H-Bridge circuits, so it is capable of driving a pair of DC motors. This makes it ideal for robotic projects, as most robots have either two or four powered wheels. The L298N can also be used to drive a single stepper motor, however we won't cover that configuration in this article.

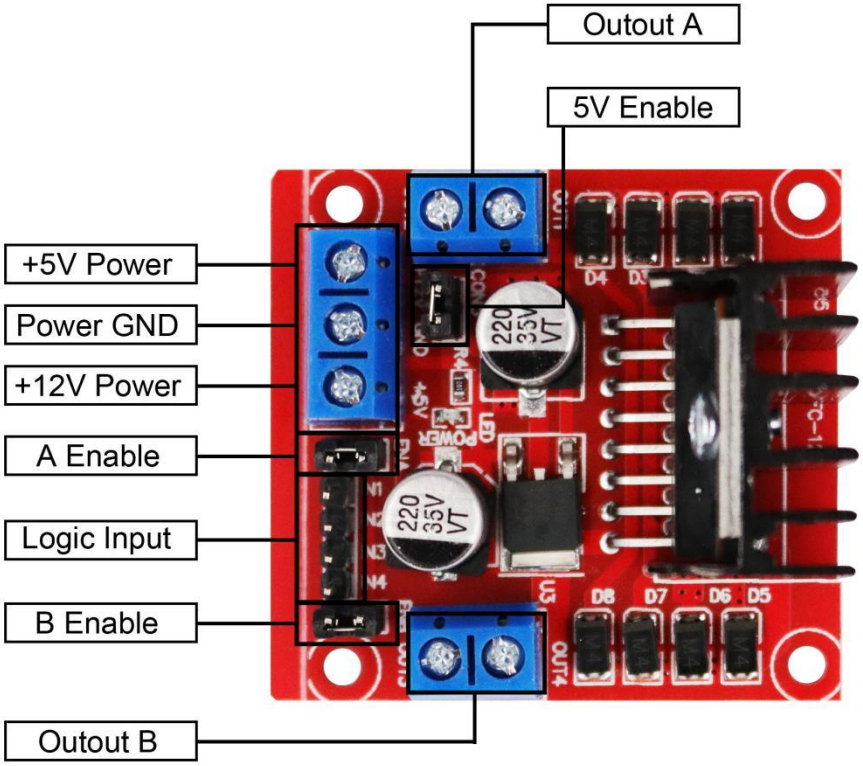
Here is a diagram of the pinouts of an L298N integrated circuit:



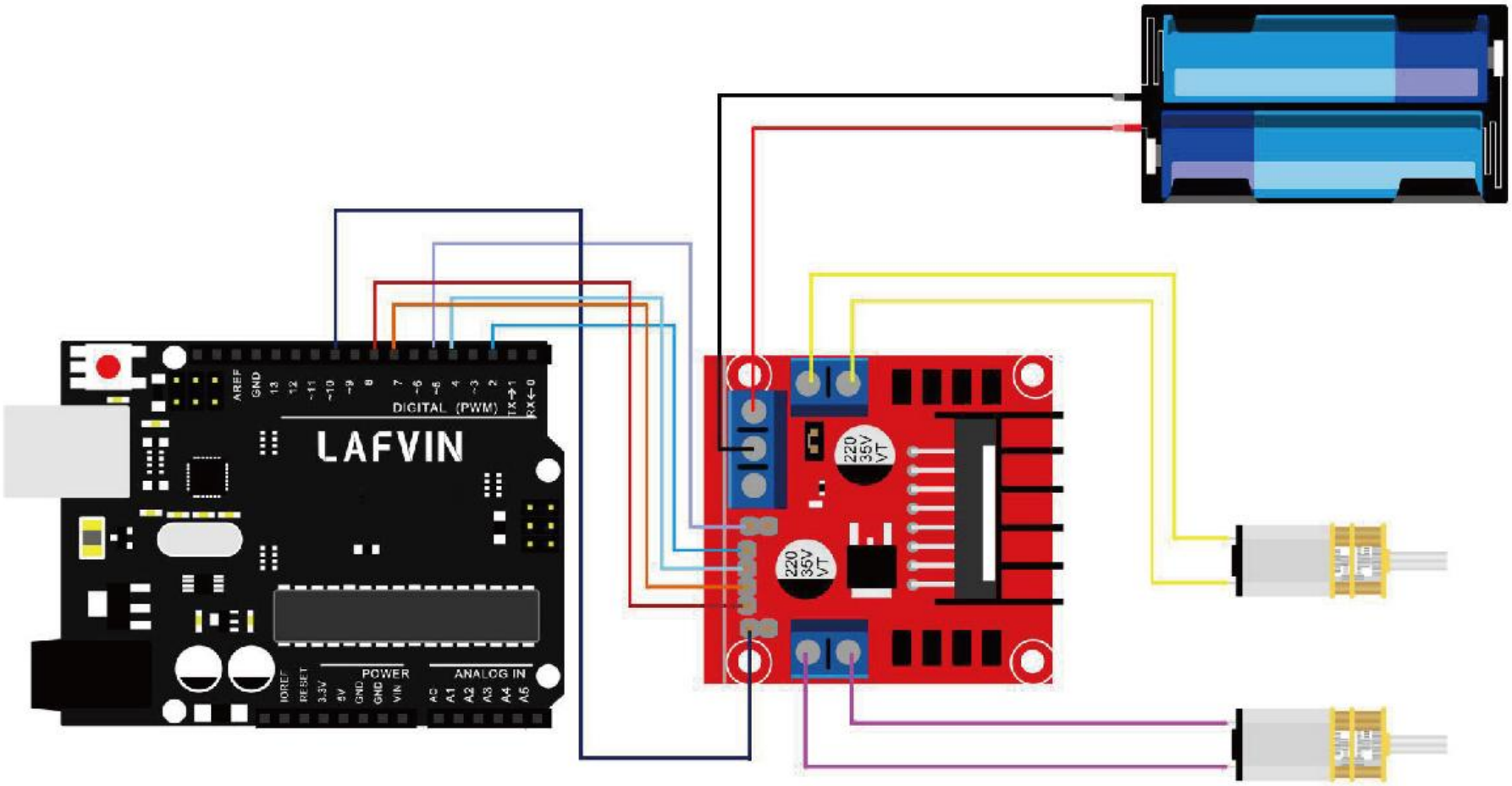


Using L298N made by ST Company as the control chip, the module has characteristics of strong driving ability, low calorific value and strong anti-interference ability.

This module can use built-in 78M05 for electric work via a driving power supply part. But to avoid the damage of the voltage stabilizing chip, please use an external 5V logic supply when using more than 12V driving voltage.
Using large capacity filter capacitor, this module can follow current to protect diodes, and improve reliability.



Wiring diagram



Code

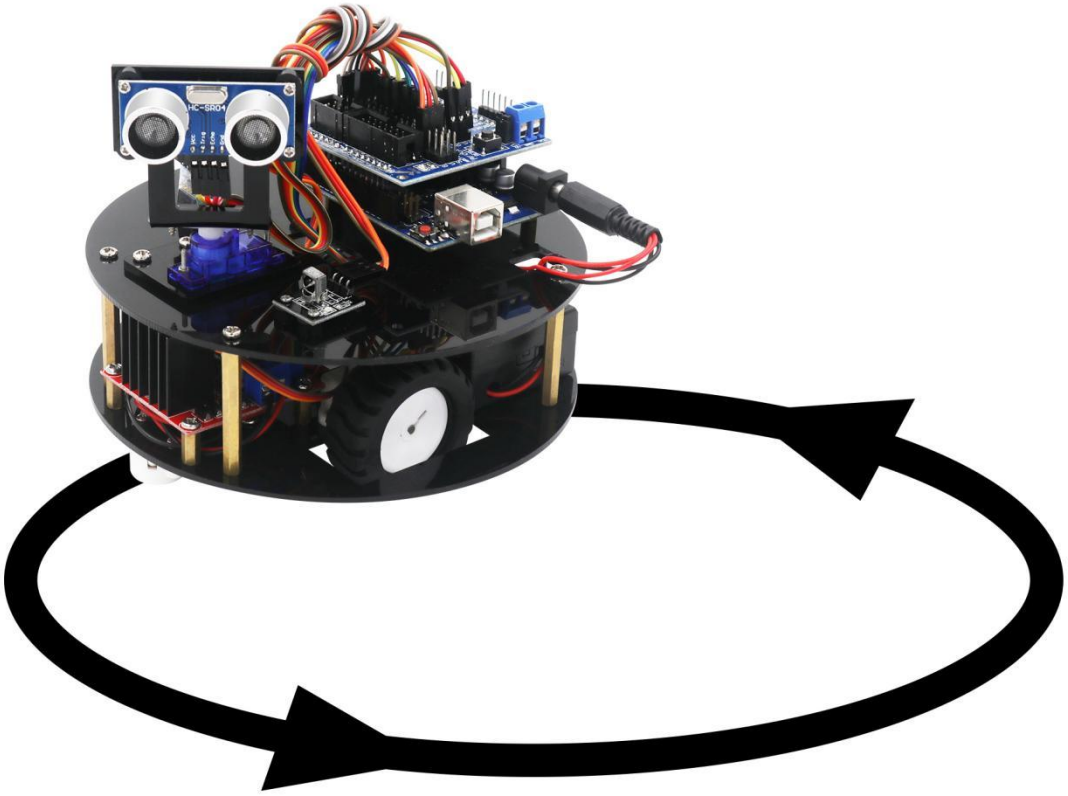
After wiring, please open the program in the code folder- Lesson 10 L298N Motor Driver and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

After connection and power-on, two motors rotate clockwise for 2 second at a speed of 200 (PWM value is 200) and then stop for 2 second; two motors rotate anticlockwise for 2 second at a speed of 100 (PWM value is 100) and then stop for 2second; circulating like this.

Lesson 11 Line Tracking Car

About this lesson:

In this lesson,we will learn a simple and automatic line tracking system of a car.



Step 1: Prepare a black track on white ground. (the width of the black track is more than 20mm and less than 30mm).
Please note, the bend angle of the track can't be larger than 90 degree. If the angle is too large, the car will move out of the track.

Step 2: Adjust the sensitivity of tracking sensor modules.
Turn on and hold the car to adjust the potentiometer on the tracking sensor with Phillips screwdriver until you get the best sensitivity status: the signal indicate LED light will turn on when sensor is above white ground, and the signal LED
will turn off when the sensor is above black track.
Signal Indicate LED ON: White Ground
Signal Indicate LED OFF: Black Track

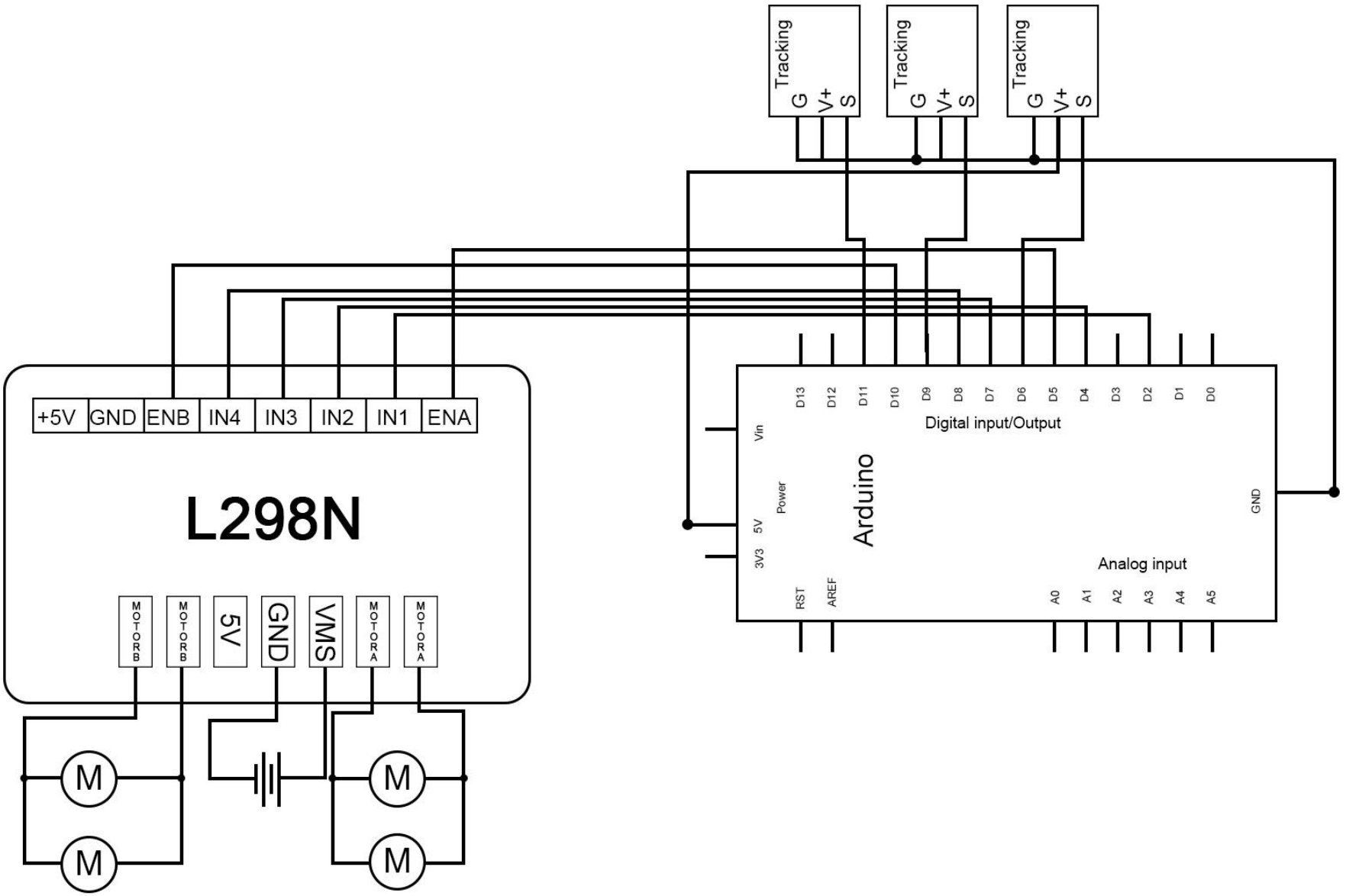
Step 3: Turn on the car and put the car over the black track, then the car will move along the black track.
If the car can't move, please check the following:
If adjusted well the sensibility of the tracking sensor

Car tracking flow chart

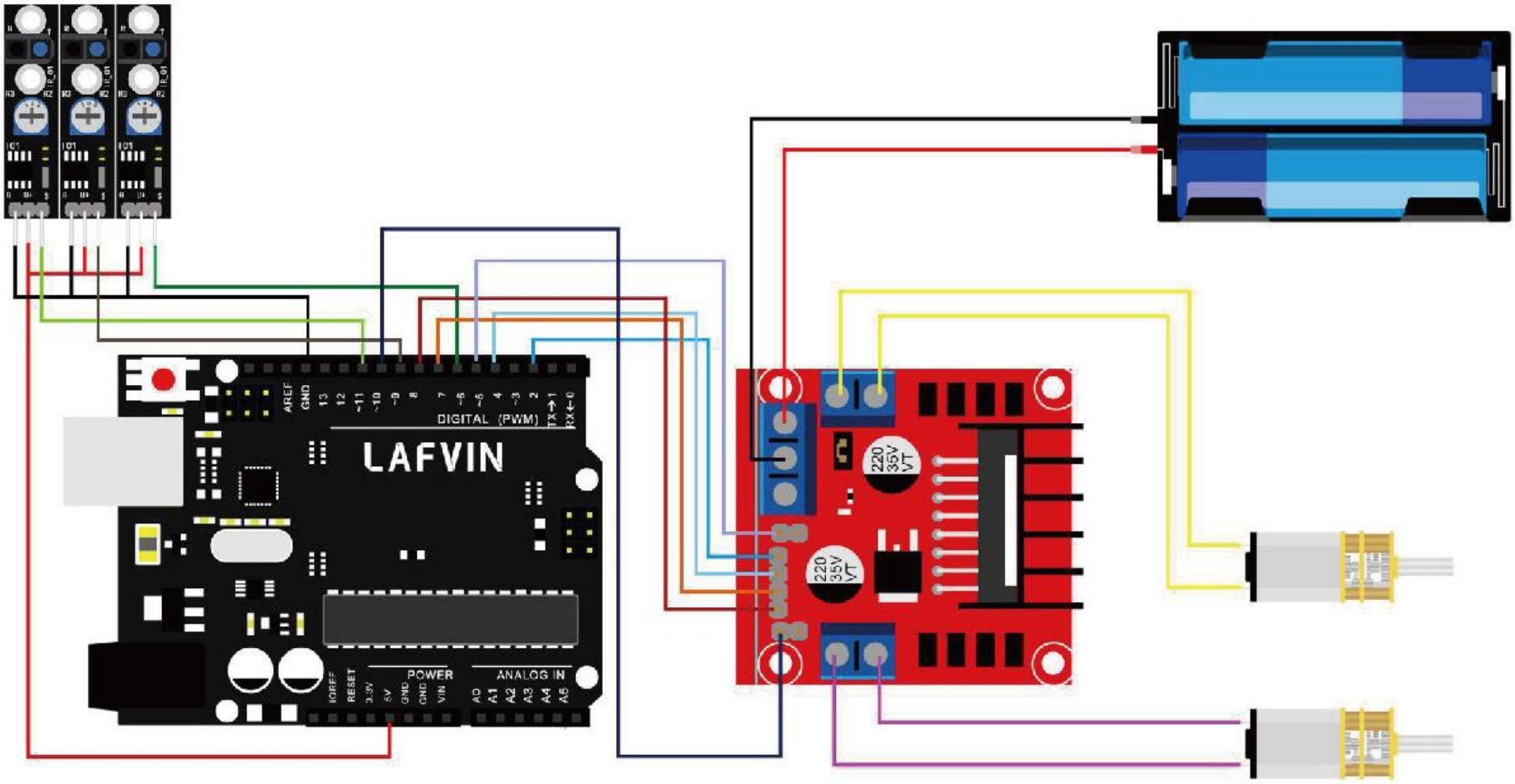
The car entered the tracking mode, namely began constantly scanning and detector connected to the I/O port of the SCM, once detected a signal of a I/O port, enter judgment processing procedures, to determine which one of 3 detectors detect the black line.

Connection

Schematic



Wiring diagram



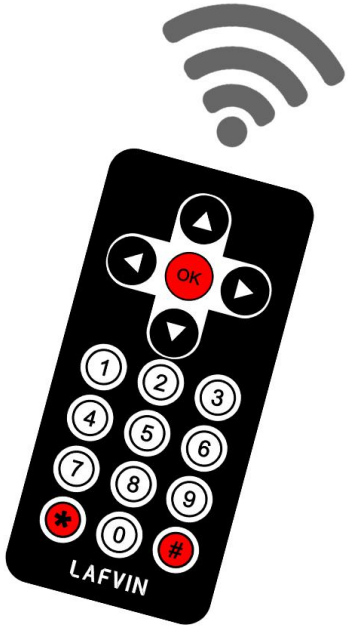
Code

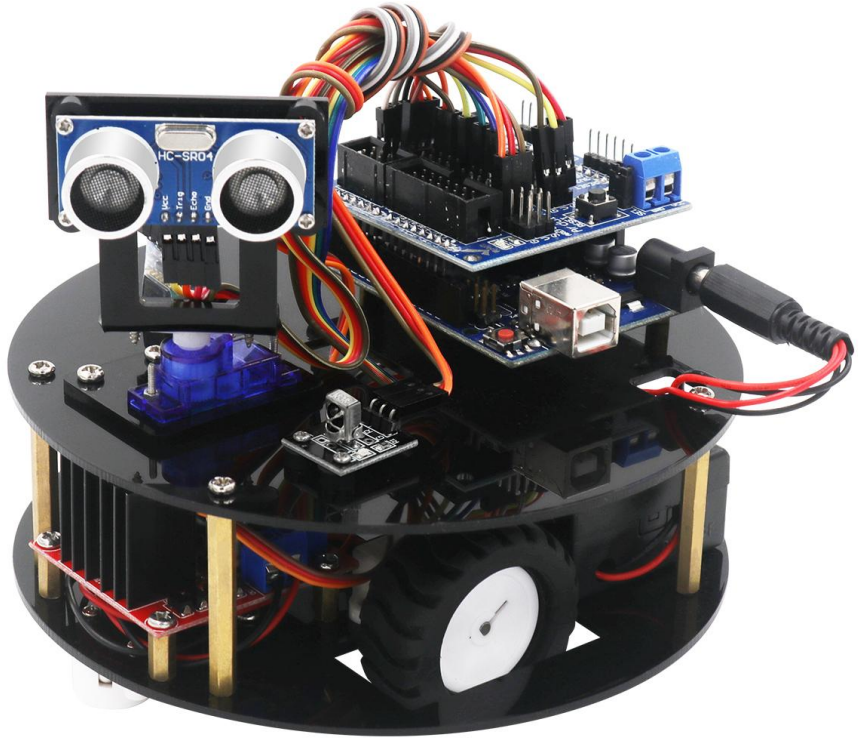
After wiring, please open the program in the code folder- Lesson 11 Line Tracking Car and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

Lesson 12 IR Remote Control Car

About this lesson:

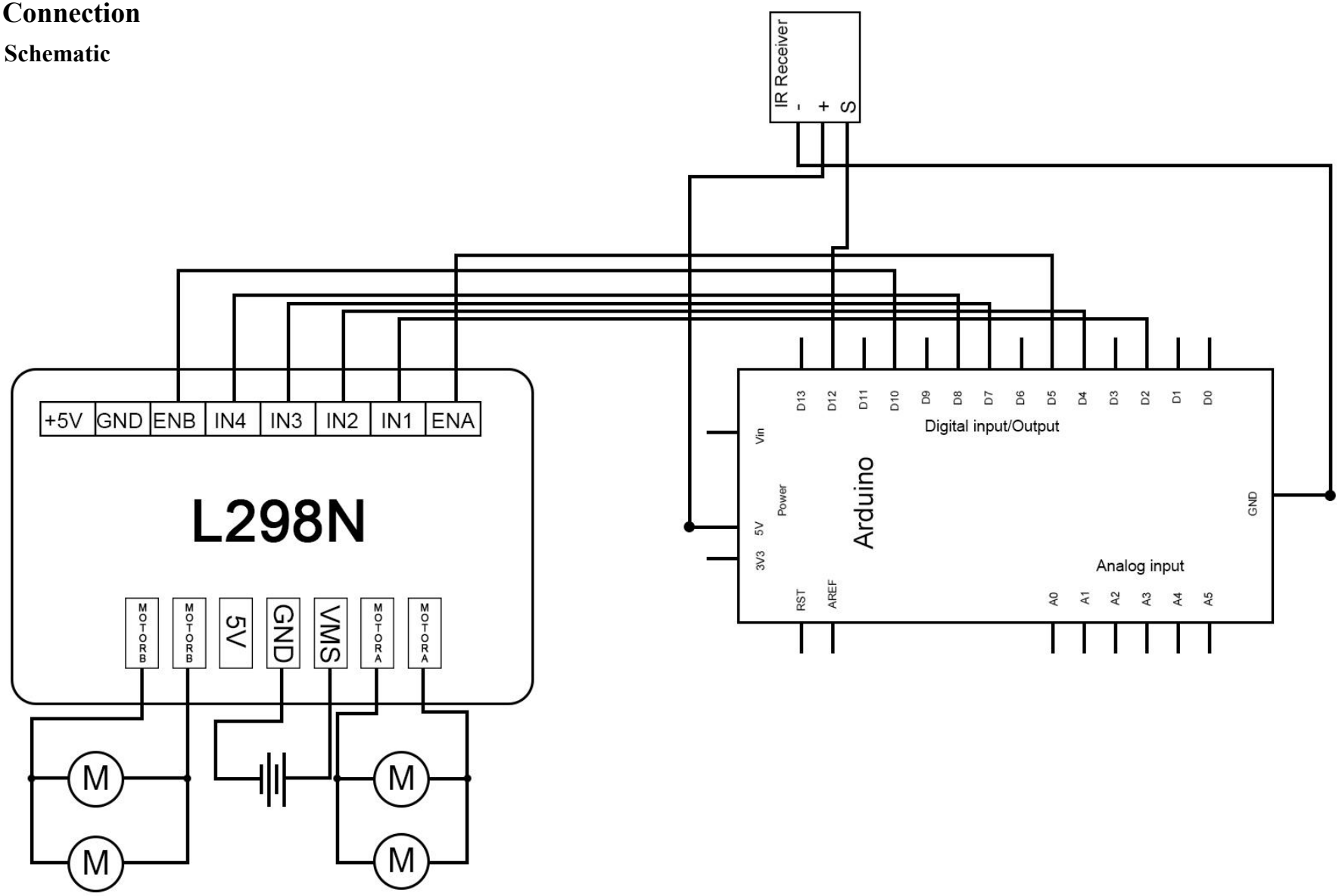
This project ,regarding Arduino microcontroller as main control, uses IR module to receive IR remote signal and send the signal to Arduino. Arduino will analyses the signal and then control the driver motor and the motion of the car with IR remote control. In addition, you can observe the state of the car through 1602 I2C Module.



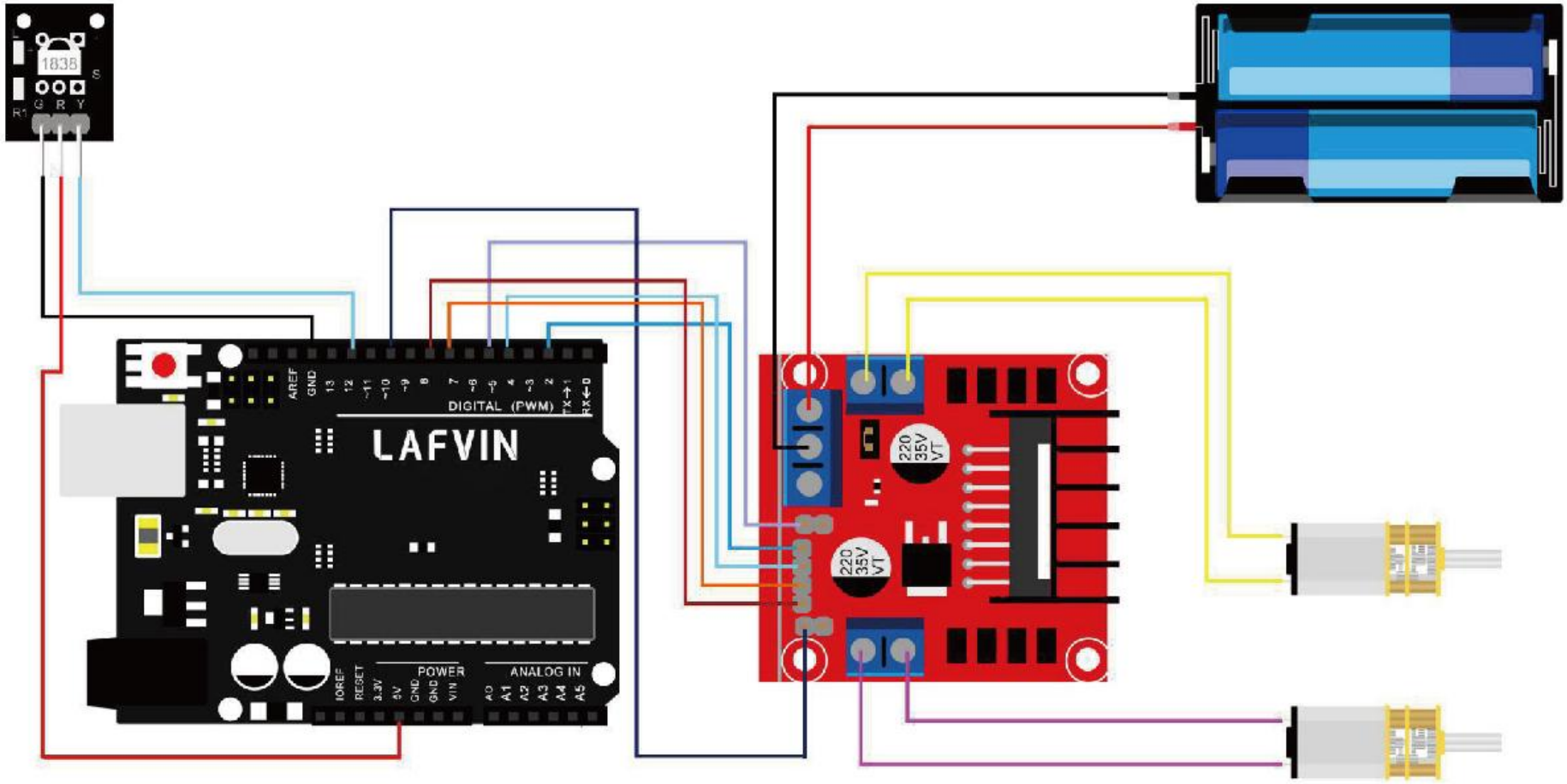


Connection

Schematic



Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson 12 IR Remote Control Car and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.

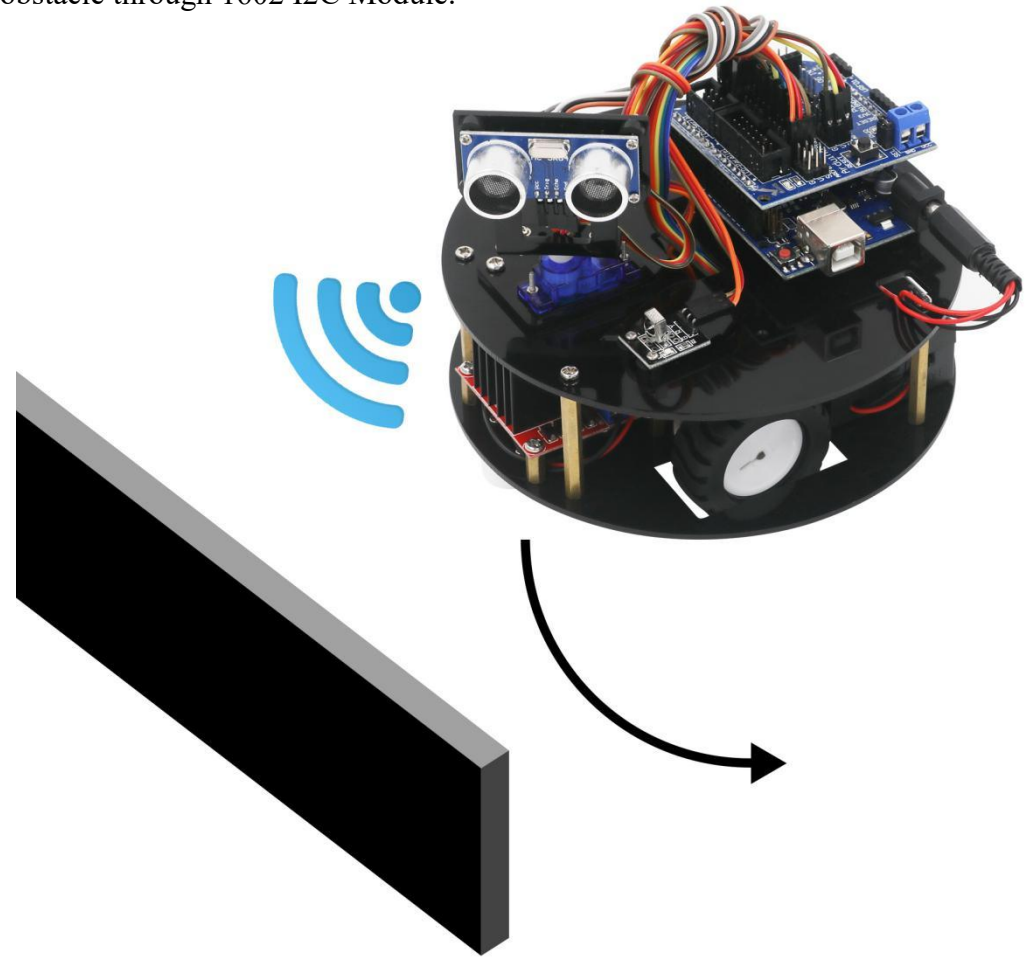
Before you can run this, make sure that you have installed the < IRremote > library or re-install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson 2.

Lesson 13 Obstacle Avoidance

About this lesson:

This lessom ,regarding Arduino as main control, detect front obstacle by ultrasonic sensor and platform motor, and send the feedback to Arduino. Arduino will analyses the feedback signal and then control the driver motor to adjust the car diversion. Finally the car is able to avoid obstacle automatically and keep going. In addition, you can observe the state and speed of the car , the angle of motor , and the distance between car and obstacle through 1602 I2C Module.



Principle:

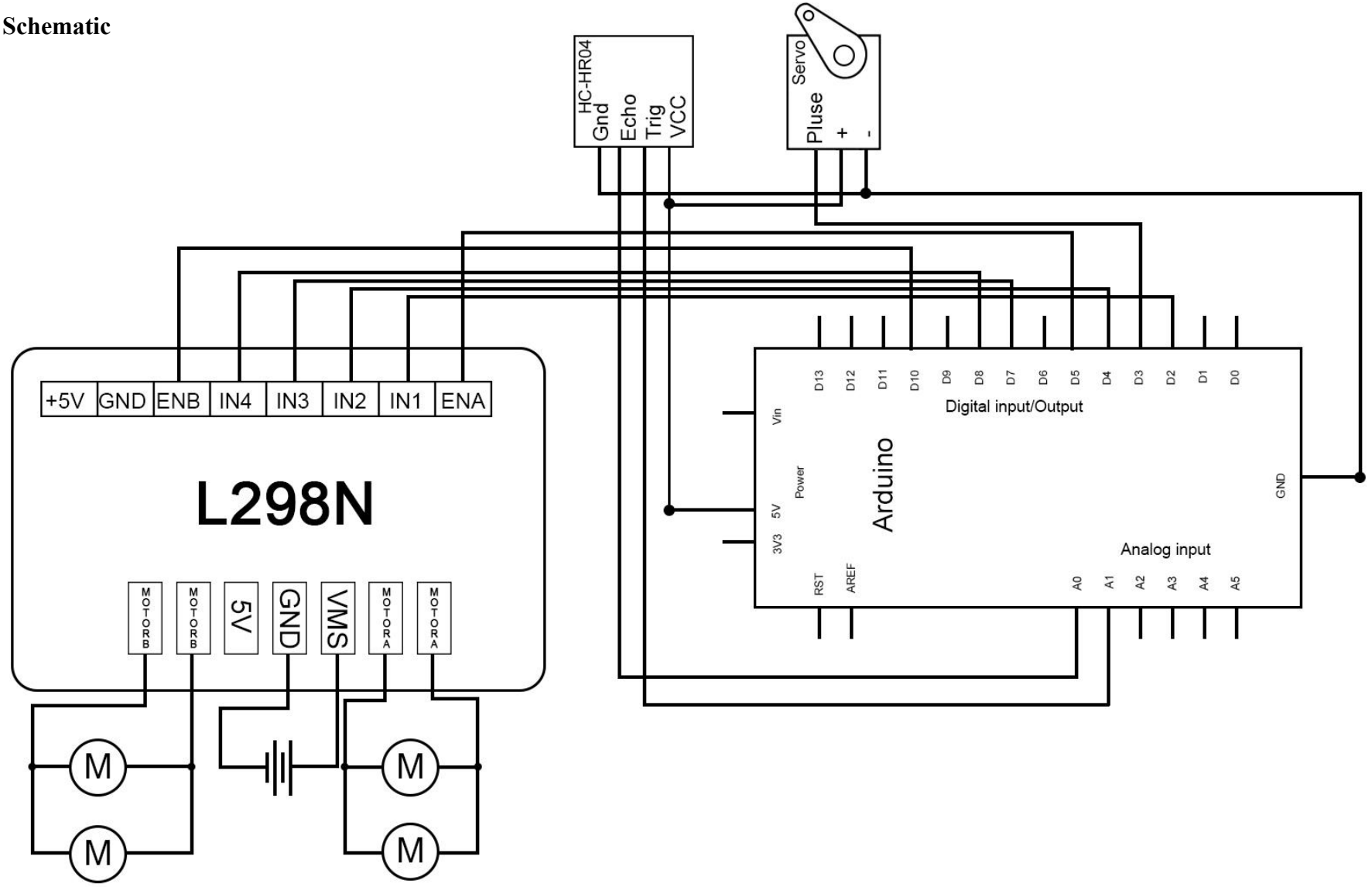
1.Ultrasonic detecting distance: one port emits high level more than 10 US. Once it outputting level, open potentiometer to time. When the port becomes low level, read out current value. Use the time of detecting distance to calculate distance.

2.Use ultrasonic to detect the distance between obstacle and car, so that control the motion of the car according to the data.

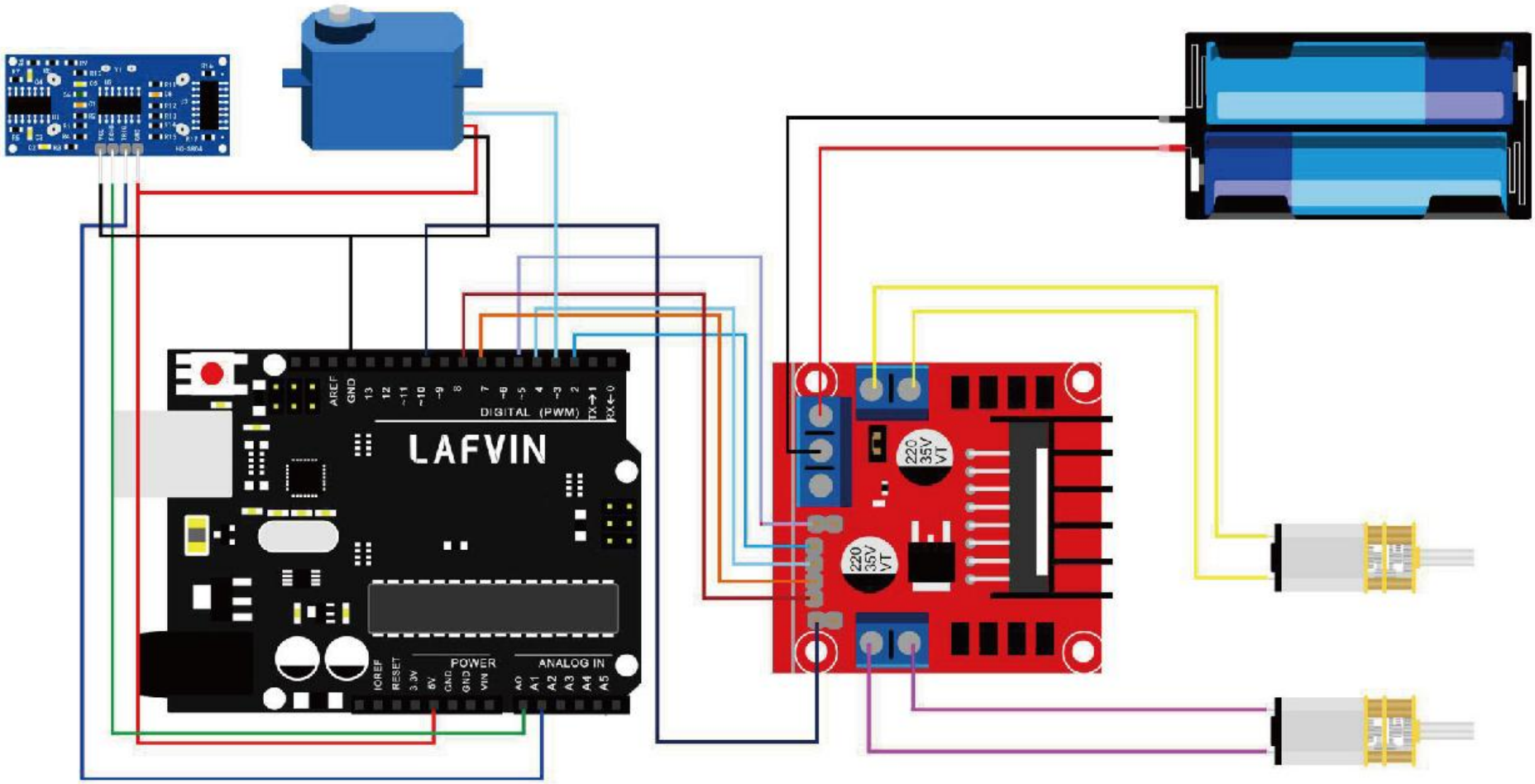
3.If the distance between the car and obstacle is less than 35 cm, the car goes backward; if the distance is no less than 10 cm, the car goes forwards; if the distance is less than 60 cm , the motor turns to detect the distance between car and left obstacle or right obstacle; if the distance between car and left obstacle, the distance between car and right obstacle are less than 35 cm, the car goes backward; if the distance between car and left obstacle is larger , the car turns left; if the distance between car and left obstacle is less than or equal to the distance between car and right obstacle, the car turns right.

Connection

Schematic



Wiring diagram



Code

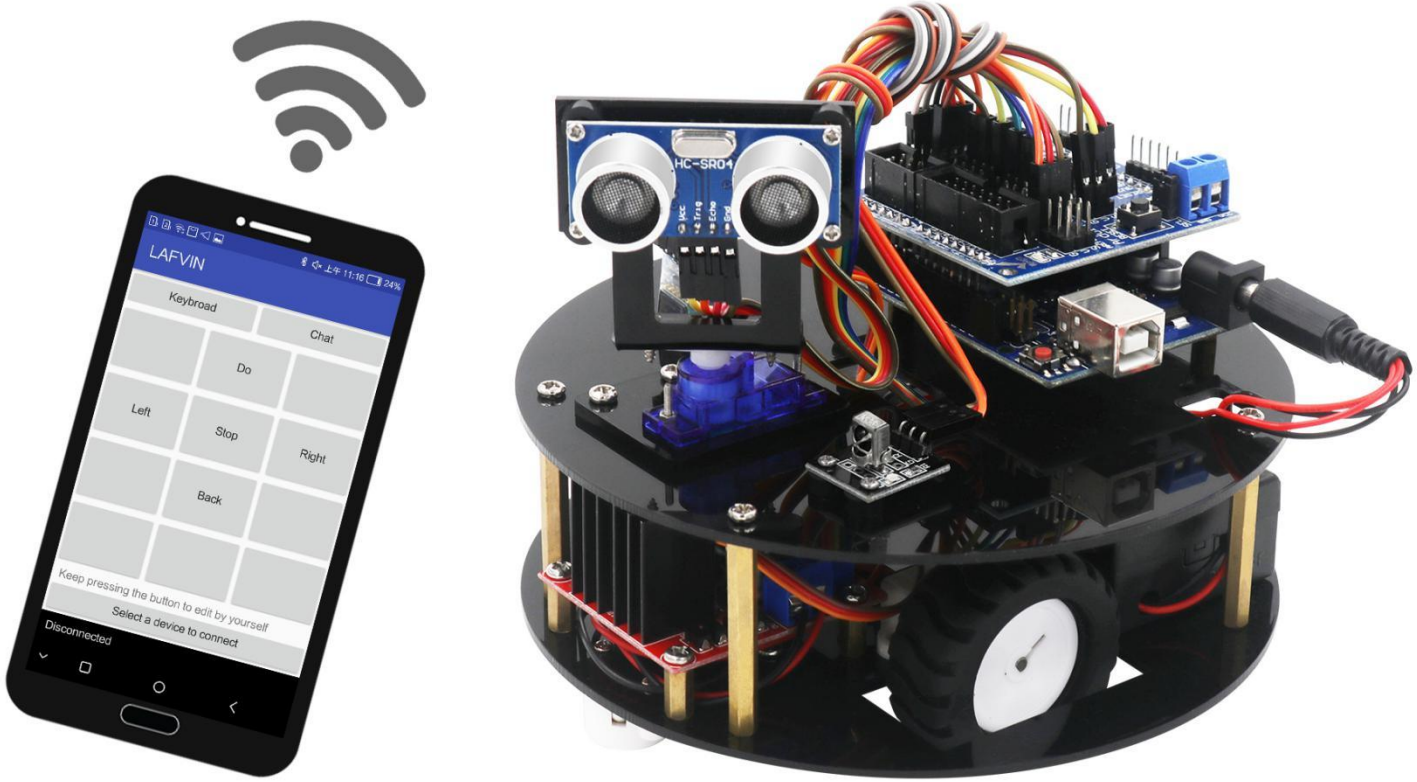
After wiring, please open the program in the code folder- Lesson 13 Obstacle Avoidance with 1602 Car and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the < Servo> < HC-SR04> library or re-install it, if necessary. Otherwise, your code won't work.
For details about loading the library file, see Lesson 2.

Lesson 14 Bluetooth Remote Control Car

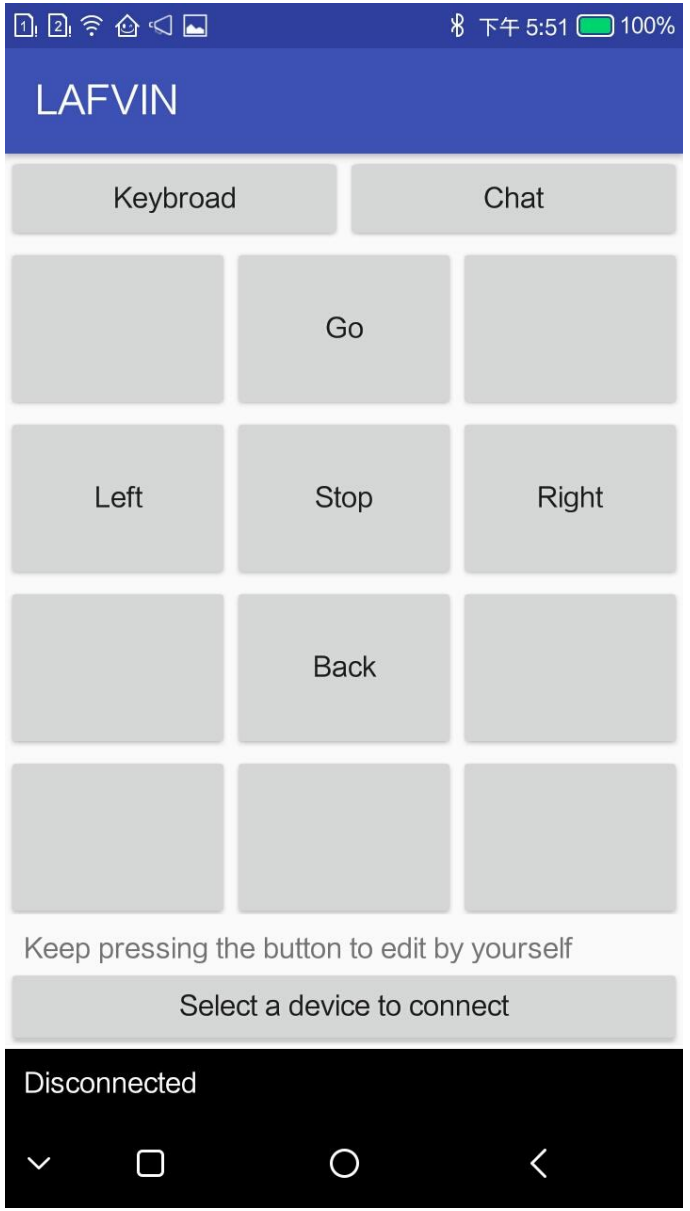
About this lesson:

This lesson , regarding Arduino as main control, use Bluetooth module to receive signal from mobile phone, and send the signal to Arduino. Arduino will analyses the signal and then control the driver motor to control the motion of the car. In addition, you can observe the state and speed of the car , the angle of motor , and the distance between car and obstacle through keyestudio 1602 I2C Module.



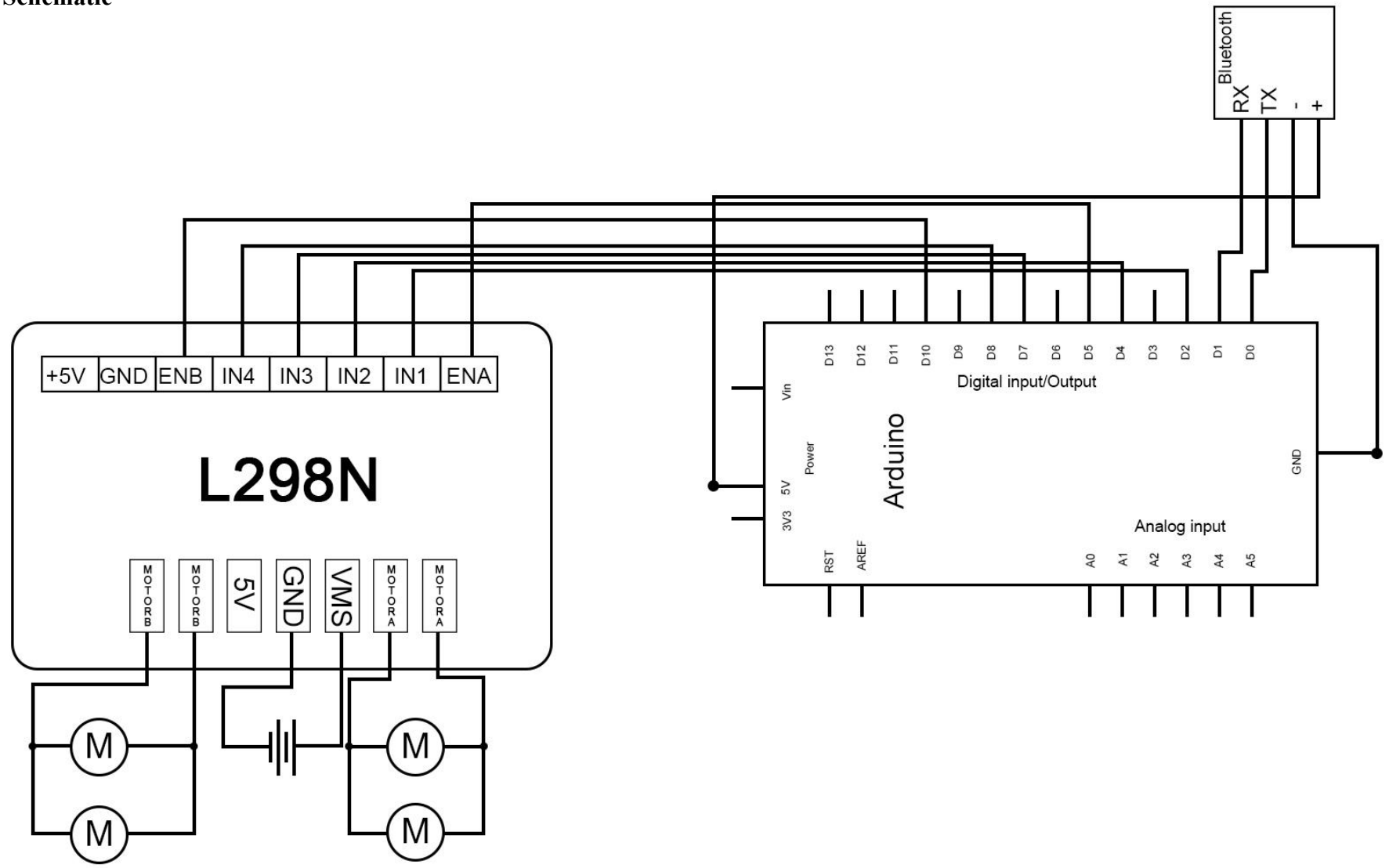
Introduction

1.Connecting Aduino to Bluetooth module , this module communicating with APP on the mobile phone
2.APP sending “U”“D”“L”“R”“S” to Bluetooth module
3.Bluetooth module will send the received massage to Arduino, and Arduino controls the car correspondingly.
4.When Arduino receiving “U”, the car goes forward; when Arduino receiving “D”, it goes backward; when Arduino receiving “R”, it turns right; when receiving “S”, it stops moving.

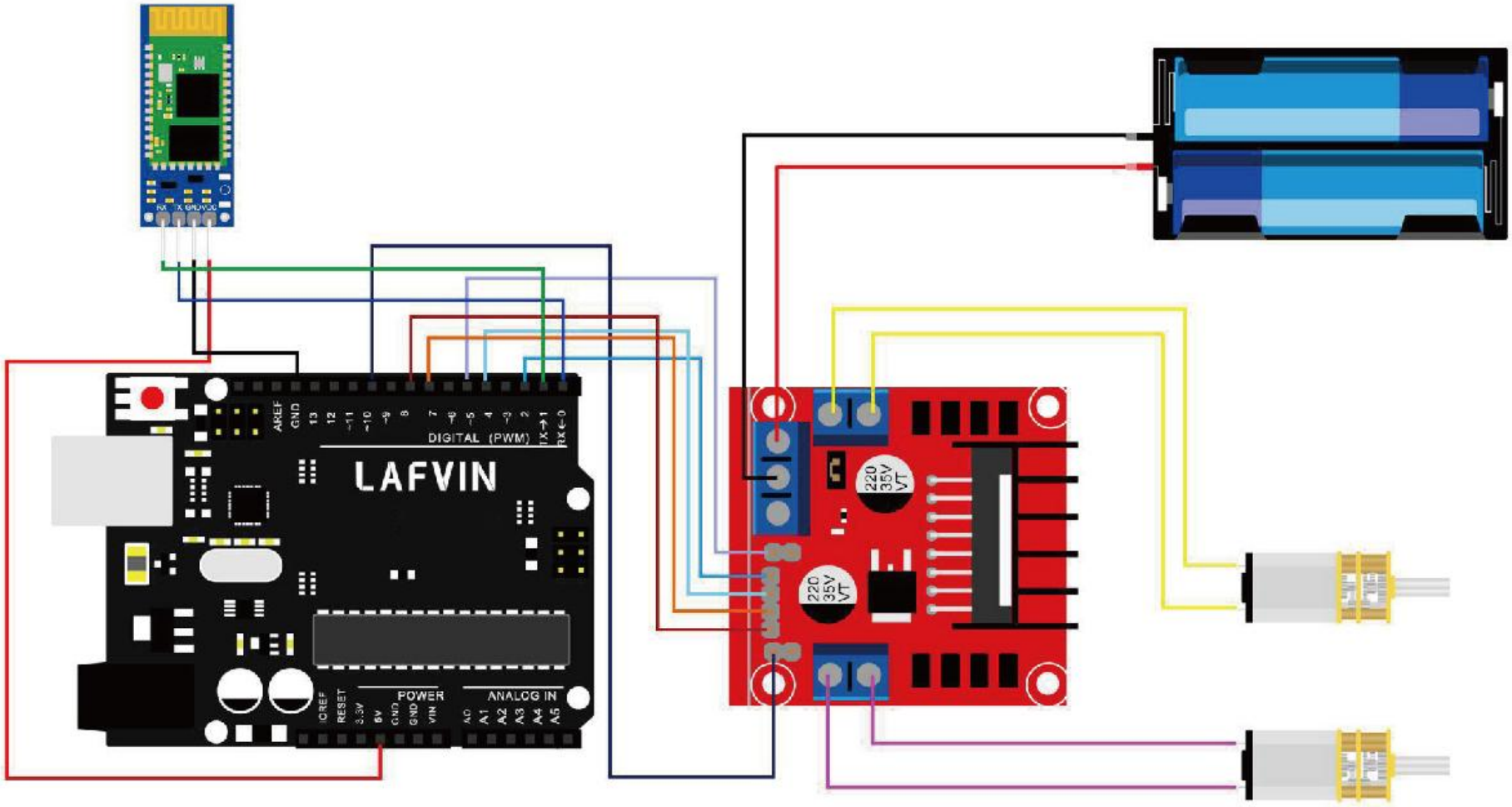


Connection

Schematic



Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson 14 Bluetooth Remote Control Car and click UPLOAD to upload the program. See Lesson 2 for details about program uploading if there are any errors.

Atterntion:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.