LAFVIN

Obstacle Avoidance Smart Car Kit

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 different 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

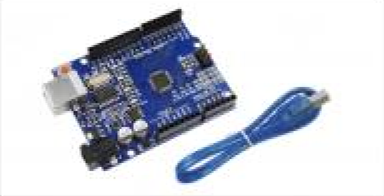
Product Catalog

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

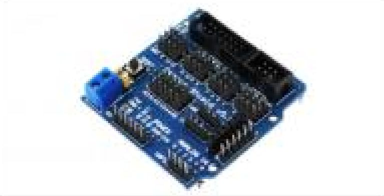
Tutorial

This tutorial include codes,libraries 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.

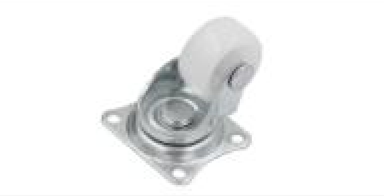
Packing list



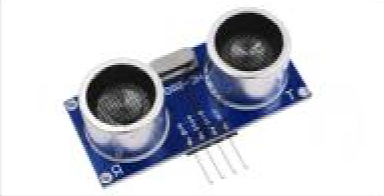
UNO R3 with Cable 1PCS



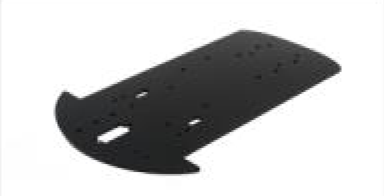
V5 Expansion Board 1PCS



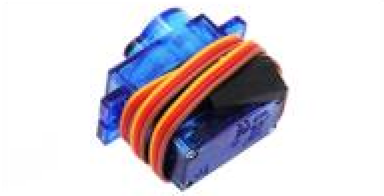
Universal Wheel 1PCS



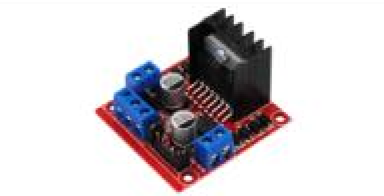
Ultrasonic Sensor 1PCS



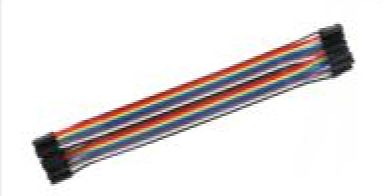
Acrylic Chassis 1PCS



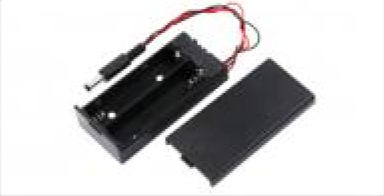
Servo Motor(SG90) 1PCS



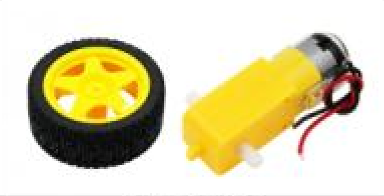
L298N Motor Driver Board 1PCS



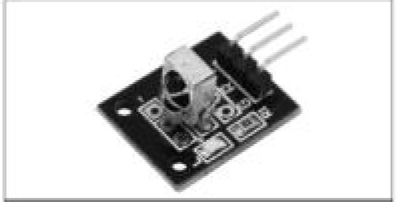
F-F Dupont Wire 20PIN



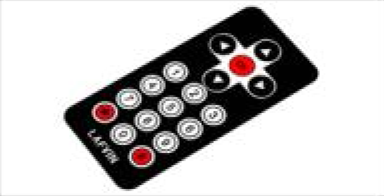
Cell Box 1PCS



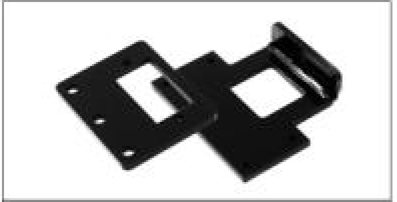
Tires 2PCS
DC Motor 2PCS



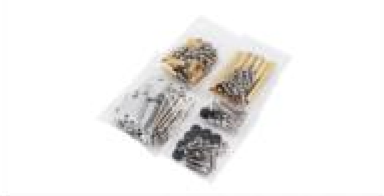
IR Receiver Module 1PCS



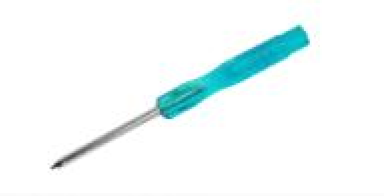
Remote Control 1PCS



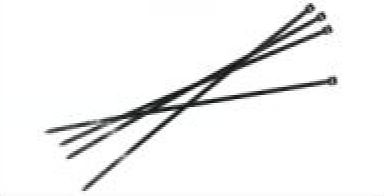
Ultrasonic Holder 1PCS



Screw Kit 1 Set



Screwdriver 1PCS



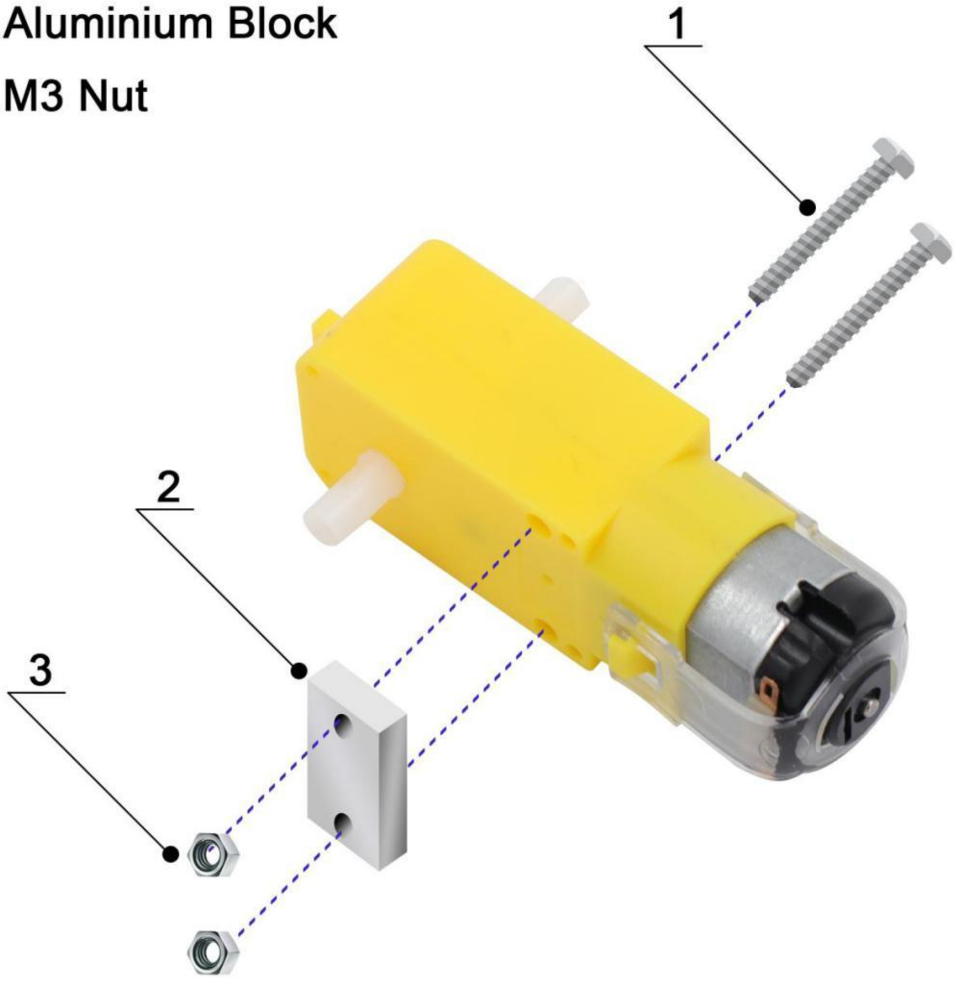
Bunding Belt 2PCS

Assembly Guide

1.M3*30mm

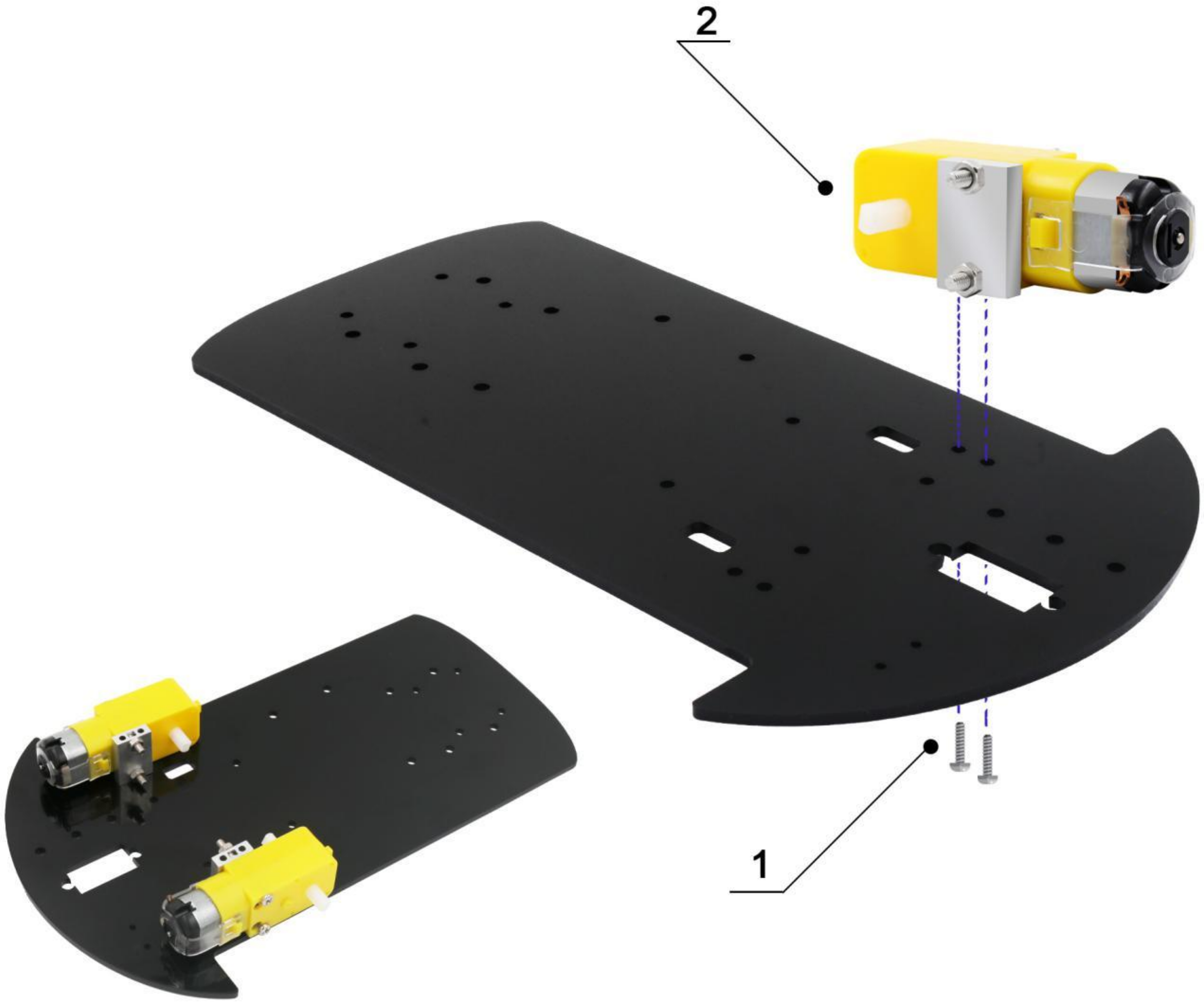
2.Aluminium Block

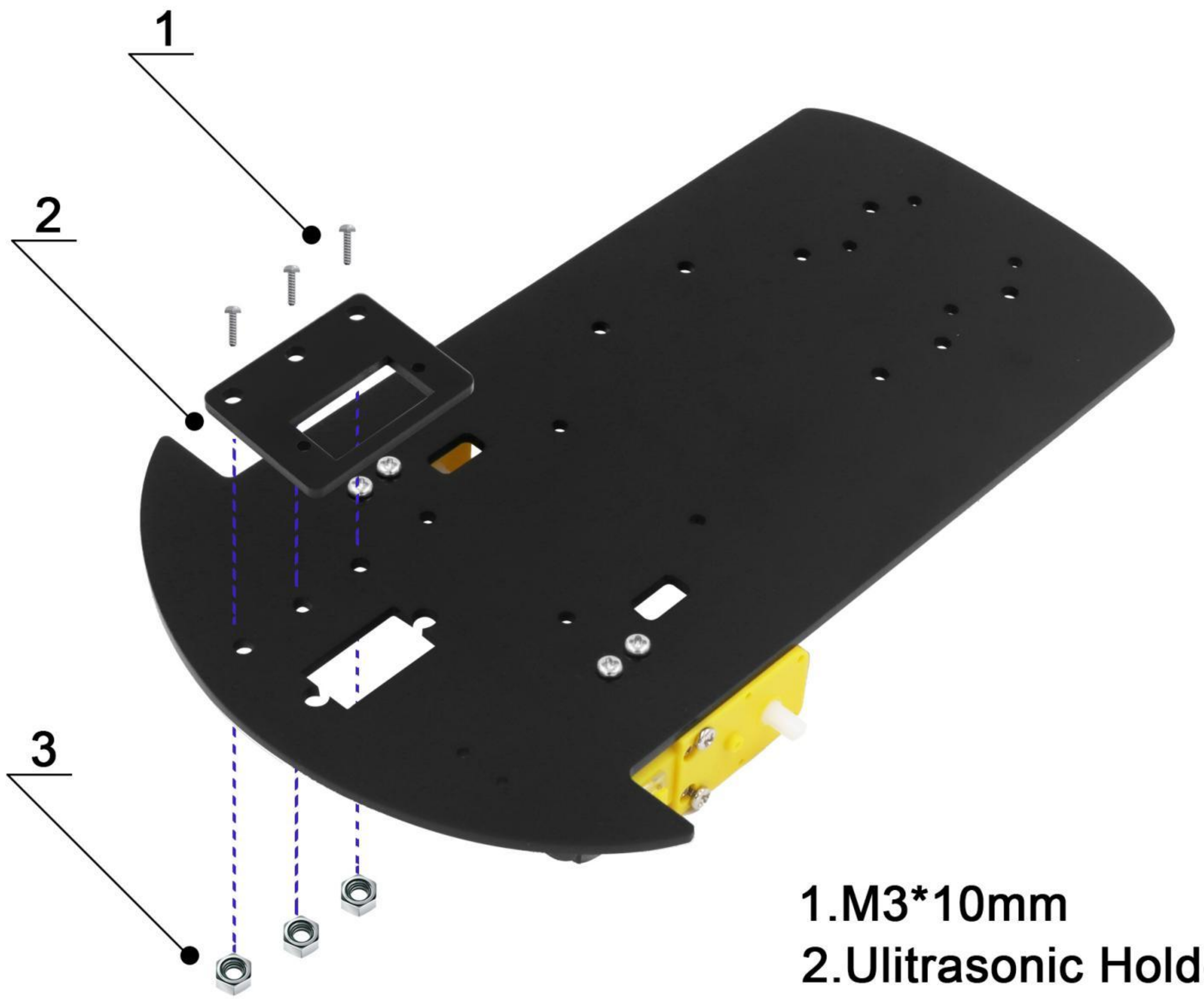
3.M3 Nut



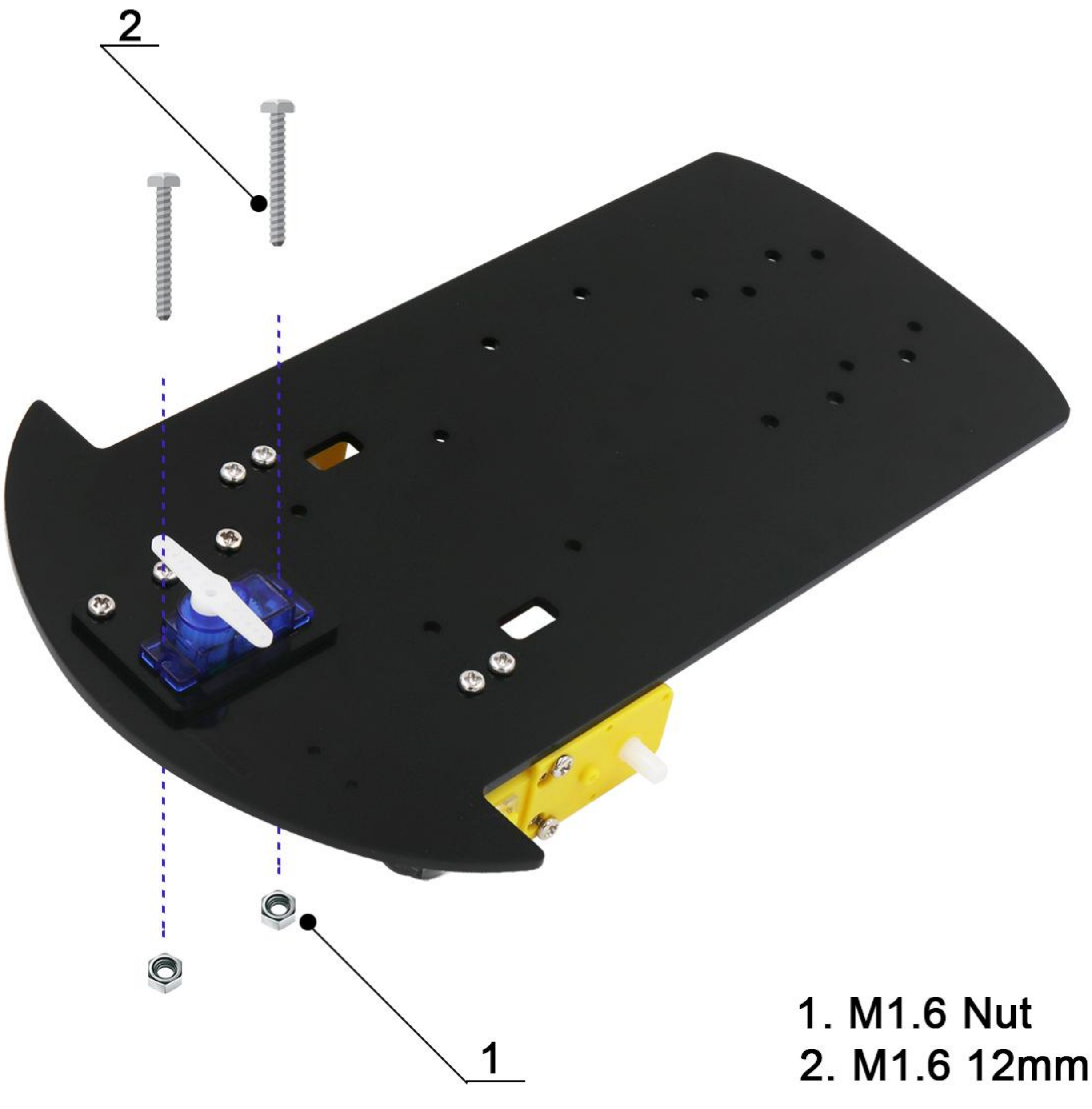
1.M3*6MM

2.Motor with Aluminium Block

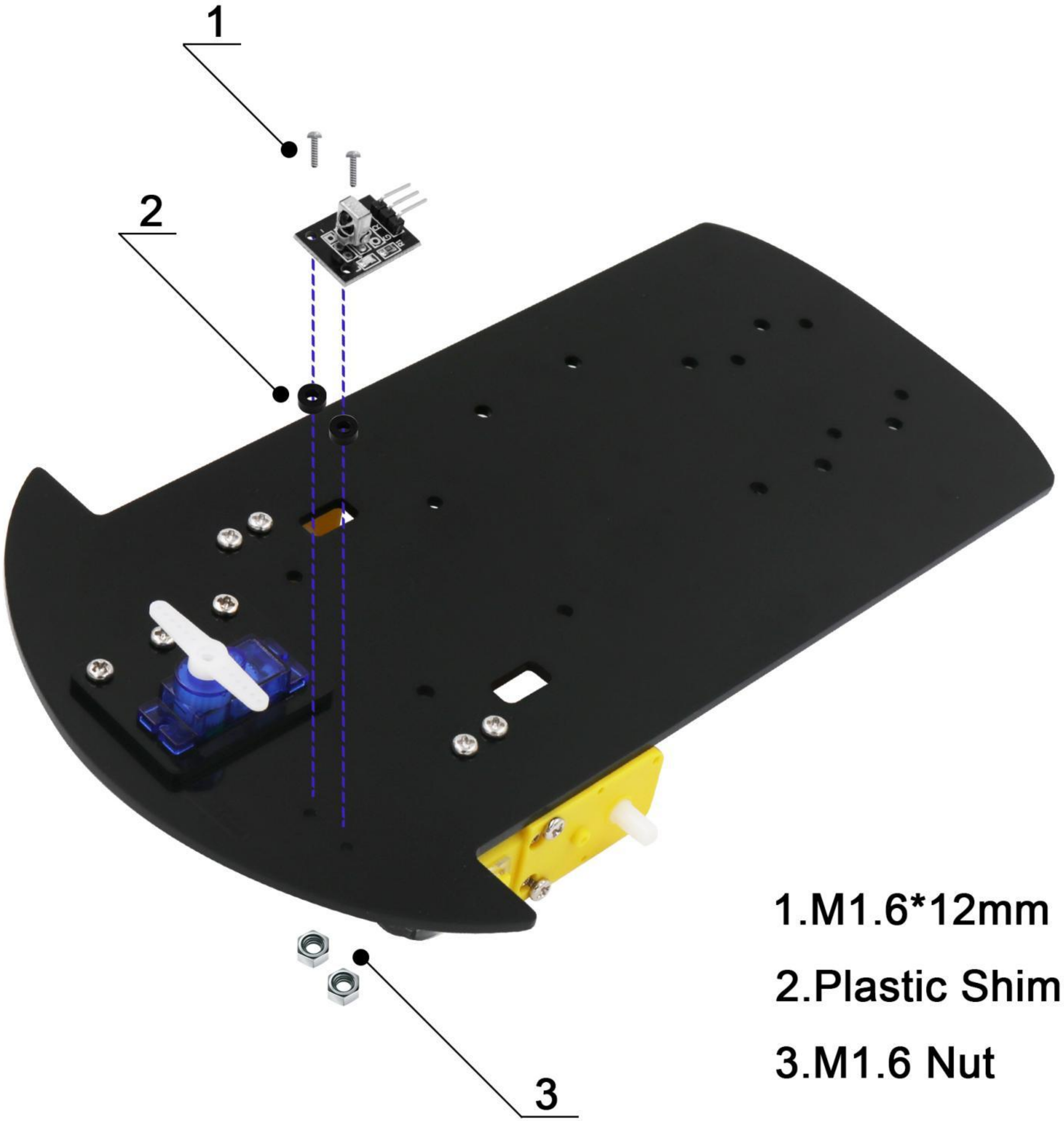




1.M3*10mm
2.Ulitrasonic Holder
3.M3 Nut



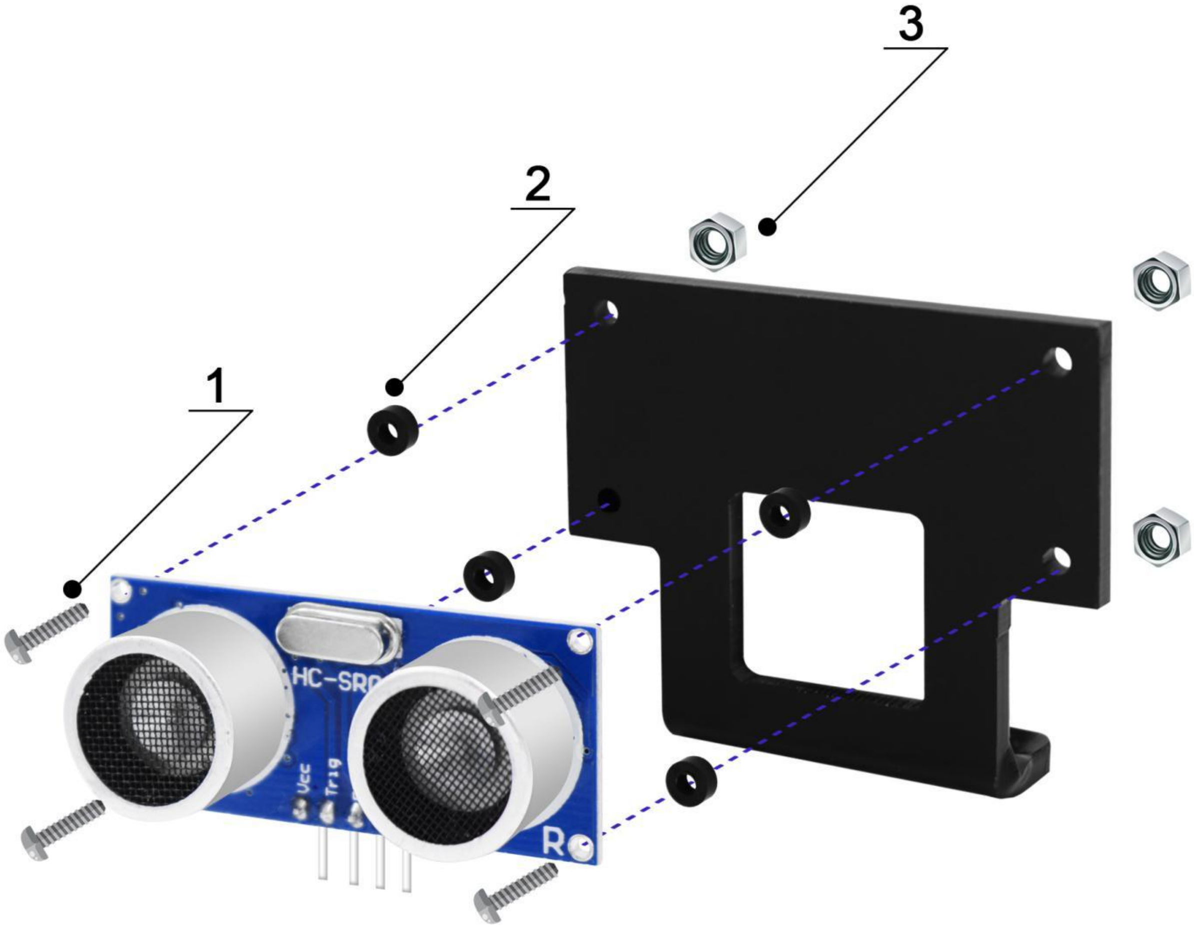
1. M1.6 Nut
2. M1.6 12mm



1.M1.6*12mm

2.Plastic Shim

3.M1.6 Nut



1.M1.6*12mm

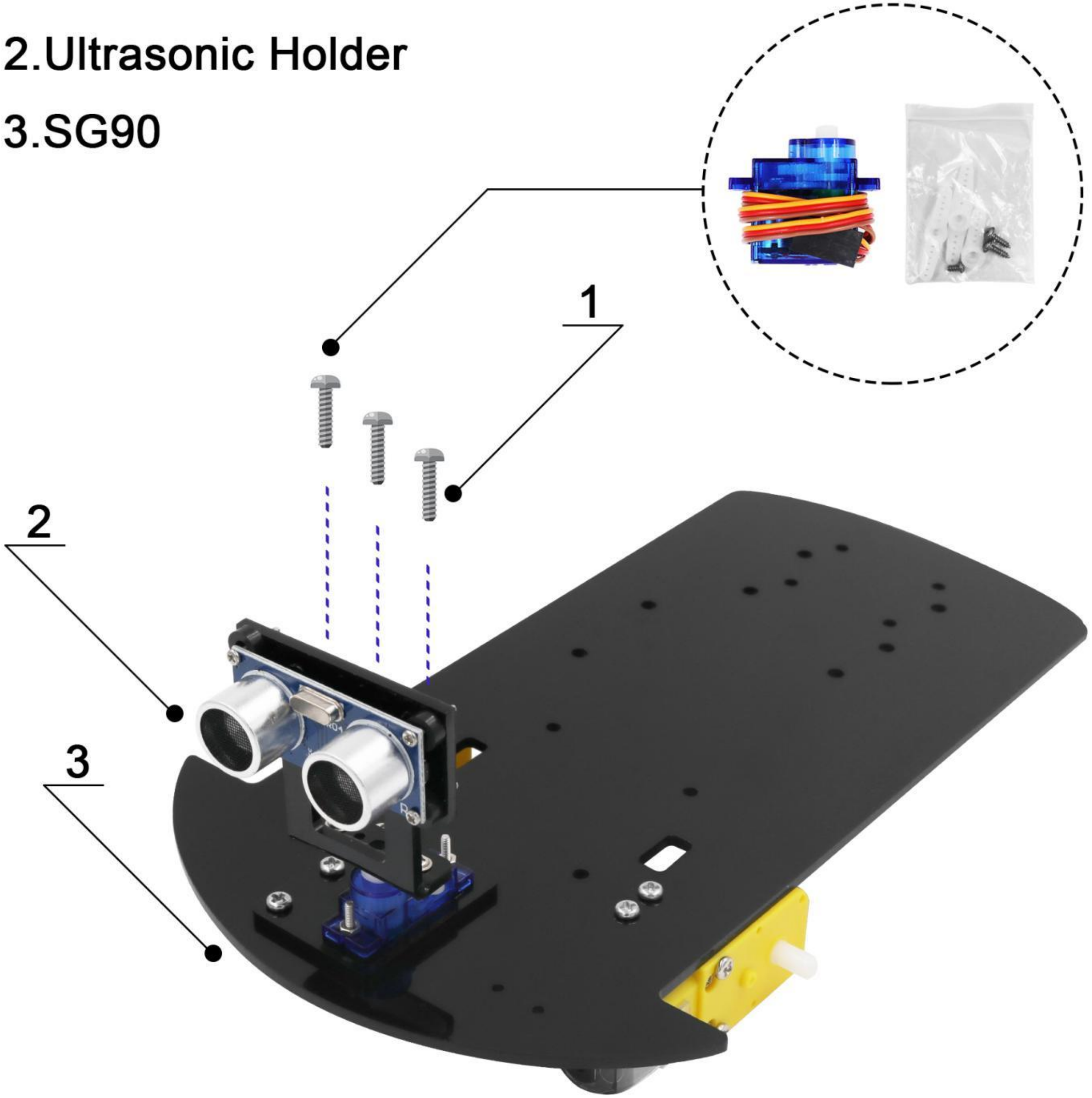
2.Plastic Shim

3.M1.6 Nut

1.SG90 Screws

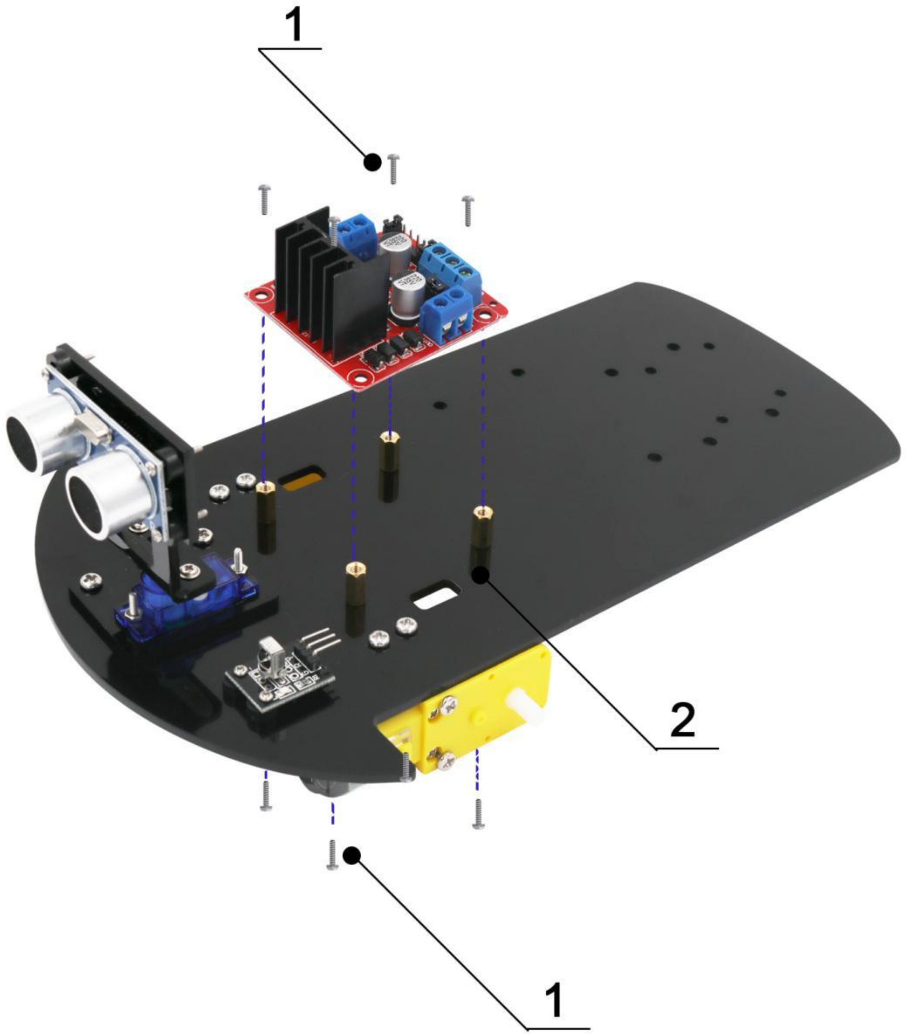
2.Ultrasonic Holder

3.SG90



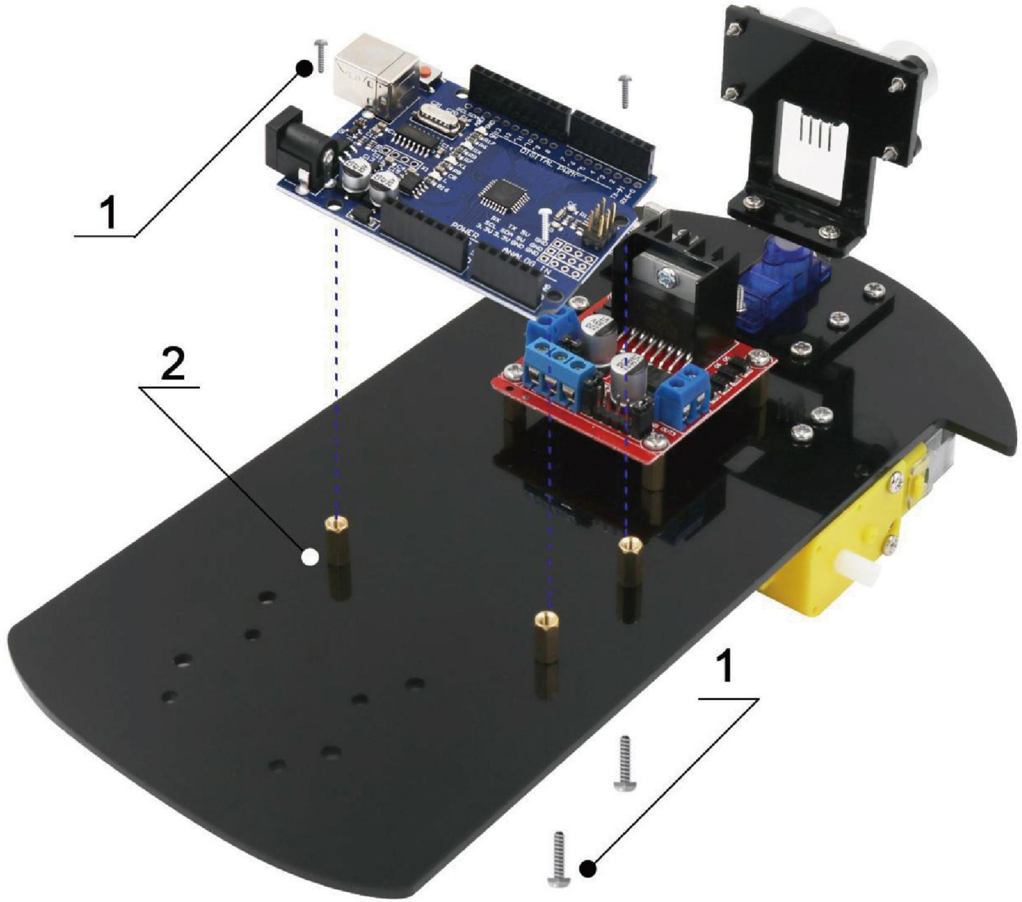
1.M3*6mm

2.Copper Cylinder M3*8mm

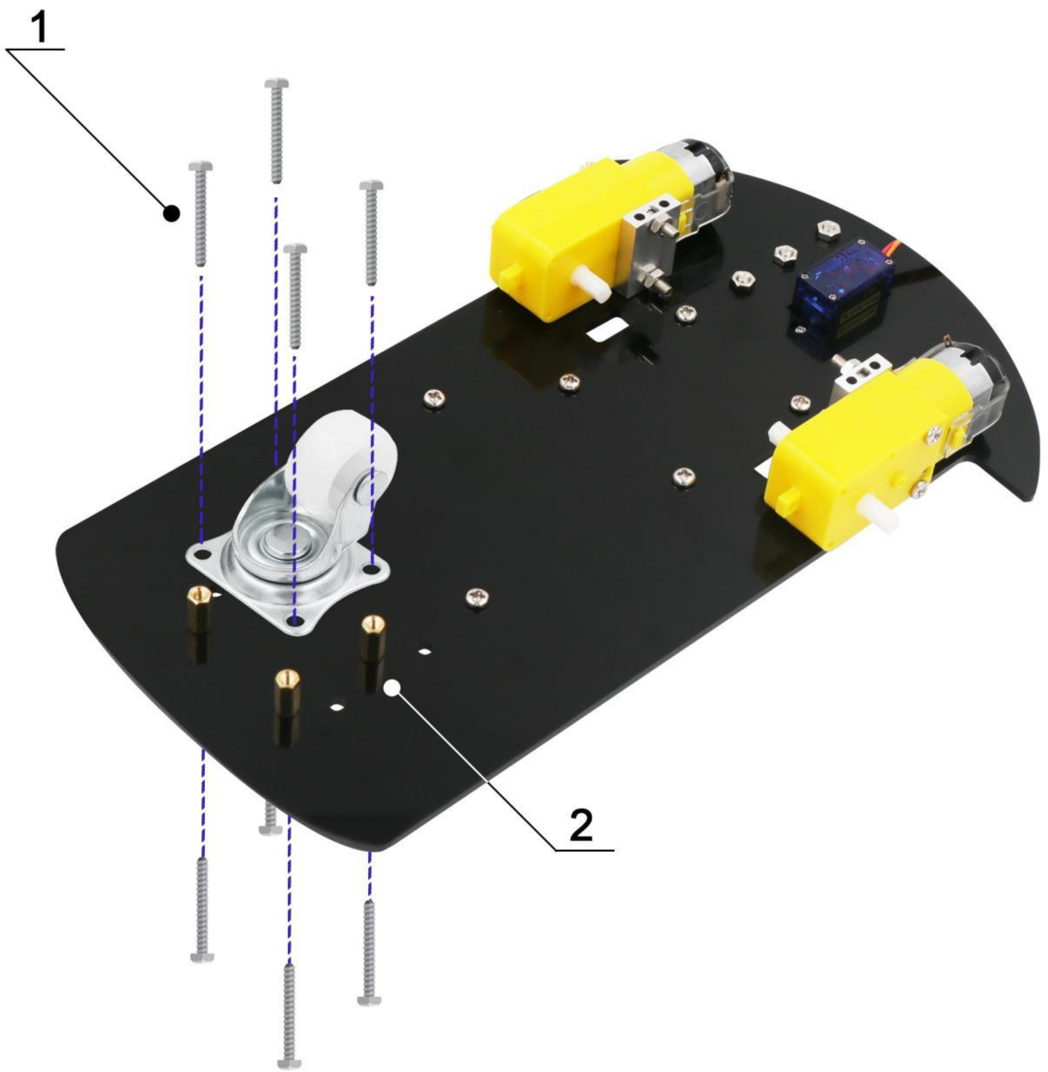


1.M3*6mm

2.Copper Cylinder M3*8mm

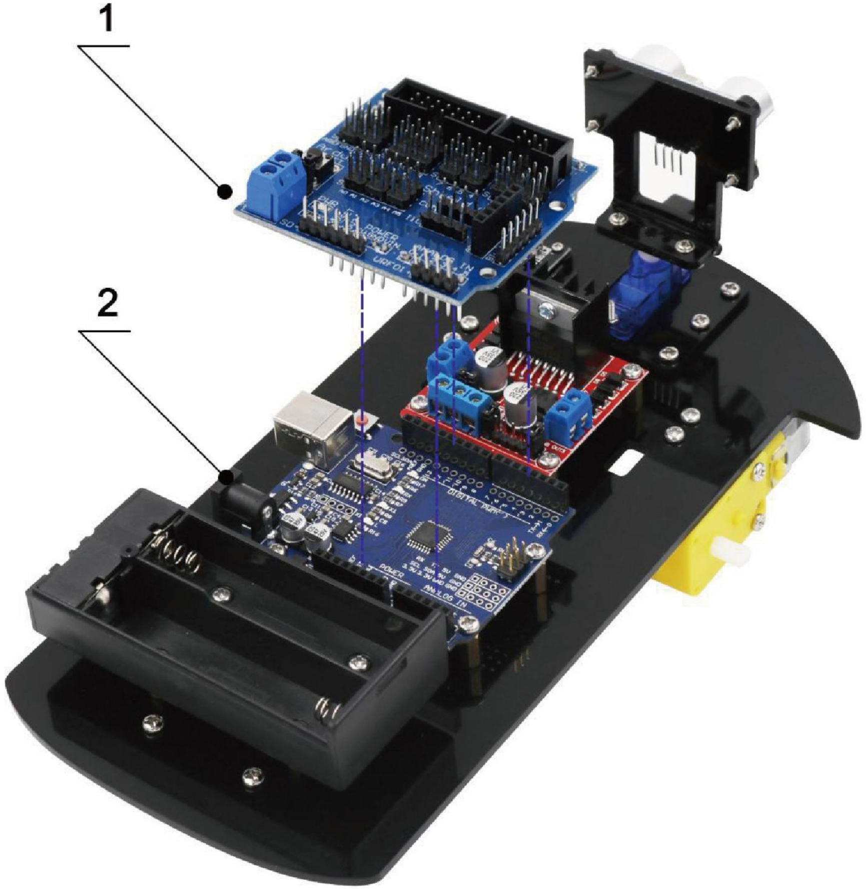


1.M2*6mm
2.Copper Cylinder M3*8mm



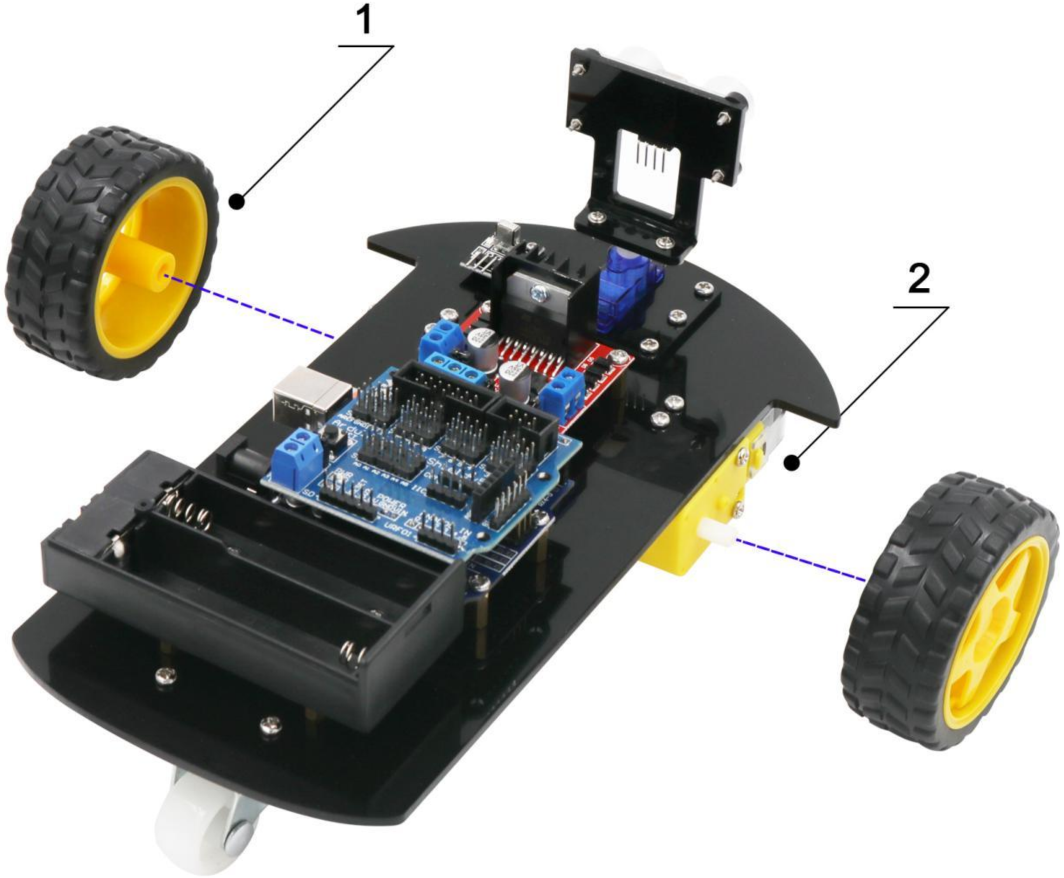
1.V5 Expansion Board

2.UNO R3



1.Tire

2.DC Motor

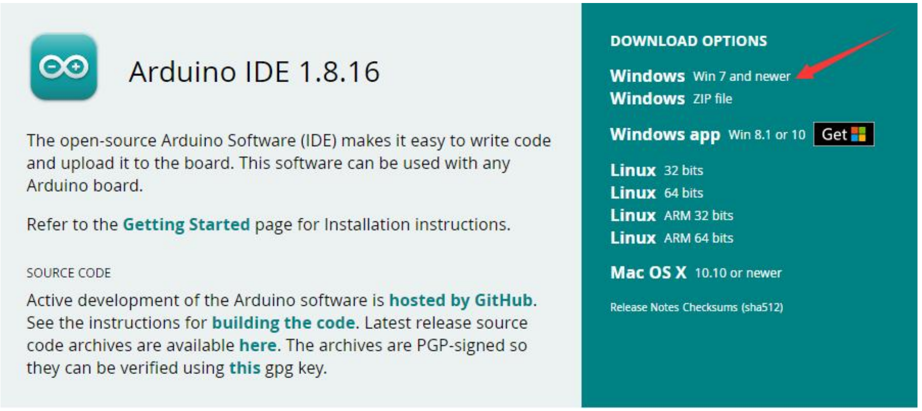


Lesson 1 How to Install Arduino IDE

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.

Go to https://www.arduino.cc/en/Main/Software and find below page.



Arduino IDE 1.8.16

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 Get

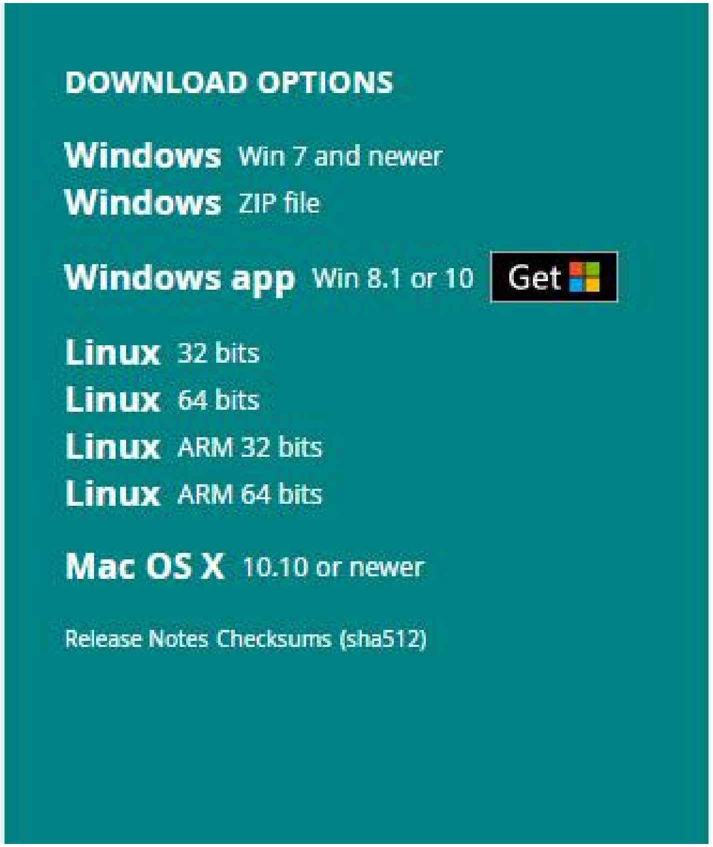
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes Checksums (sha512)

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

Download the development software that is compatible with the operating system of your computer.
Take Windows as an example here.



DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

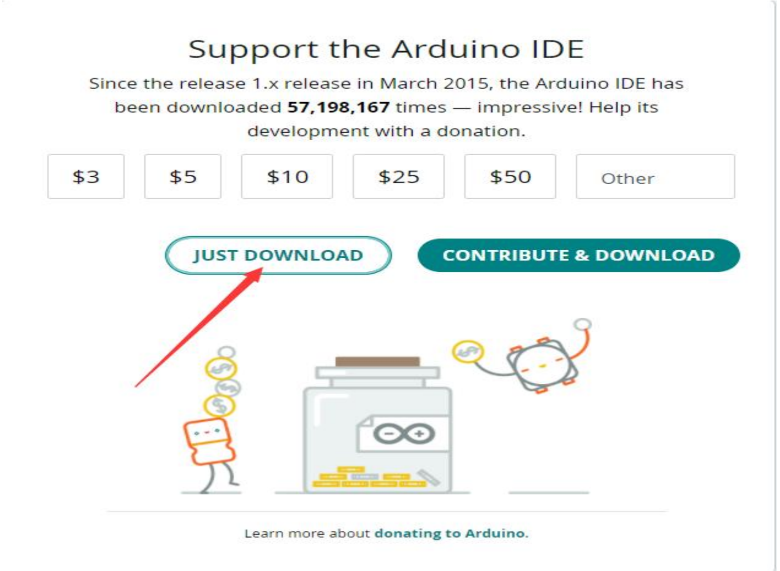
Windows app Win 8.1 or 10 Get

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes Checksums (sha512)

Click Windows Win 7 and newer.



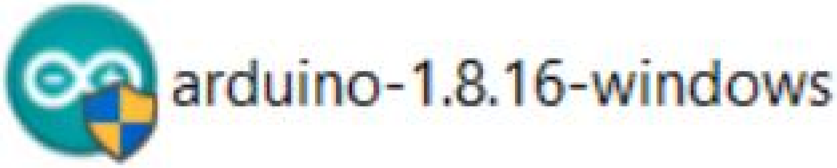
Support the Arduino IDE

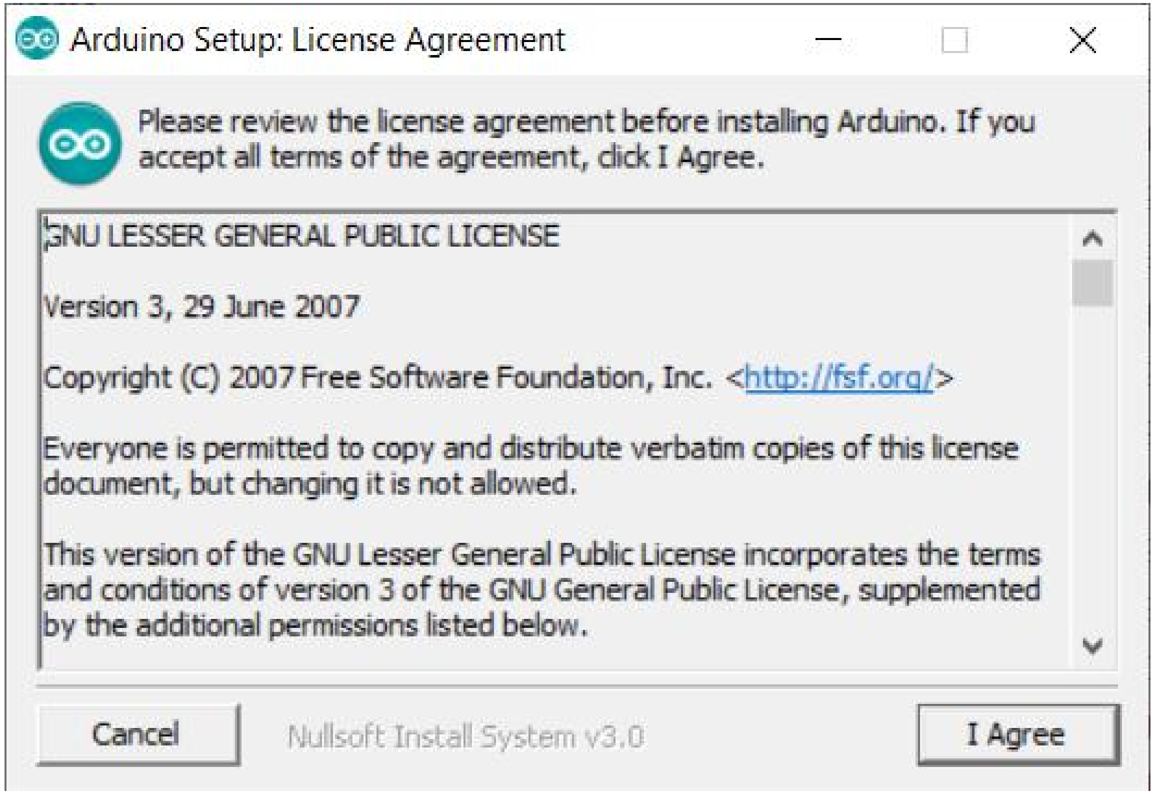
Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **57,198,167** times — impressive! Help its development with a donation.

Learn more about **donating to Arduino**.

Click JUST DOWNLOAD.
Choose the path you want to save.

After the download is complete.Double click “arduino-1.8.16-windows.exe”




Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

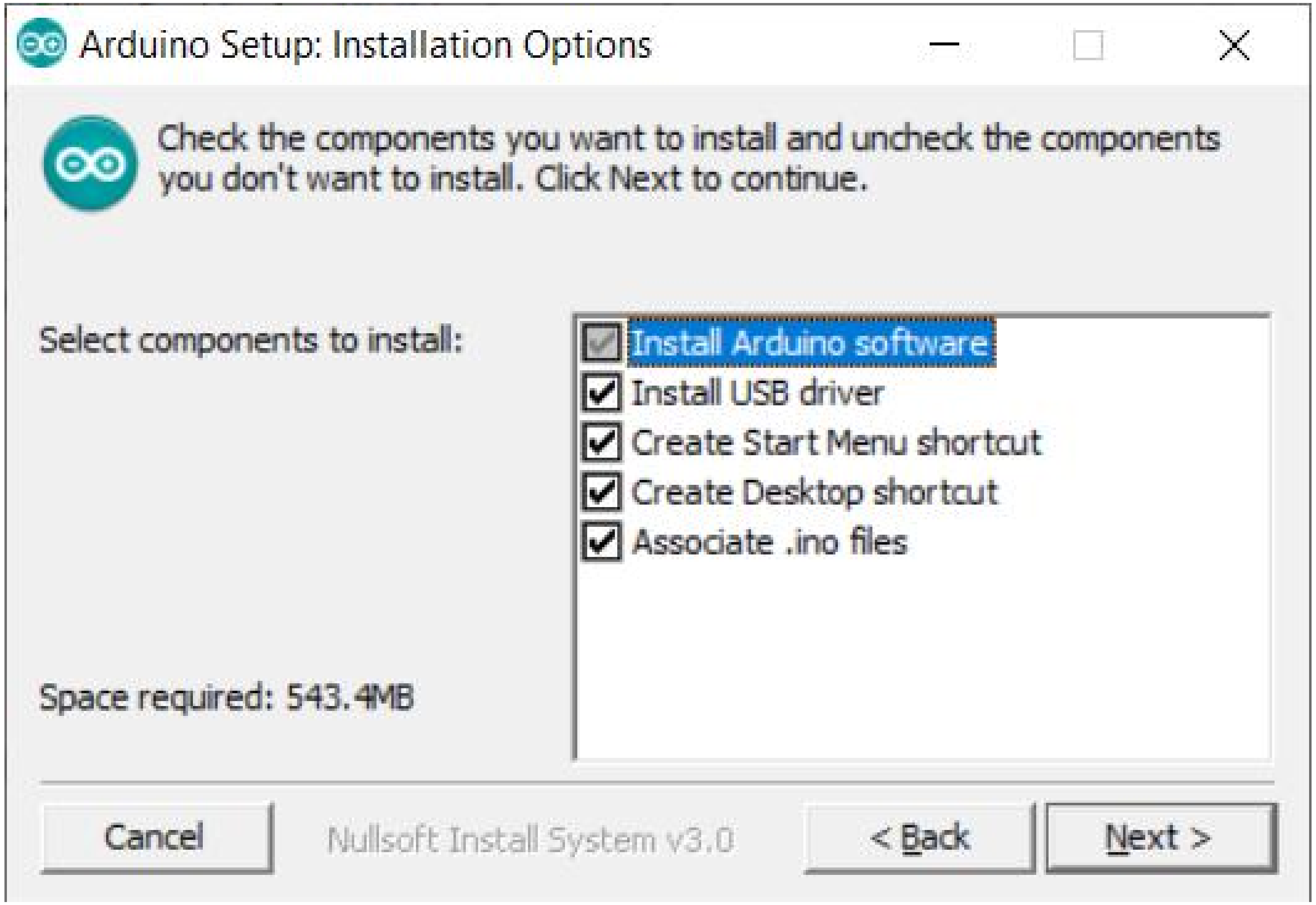
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Click I Agree to see the following interface



Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Install Arduino software

☑ Install USB driver

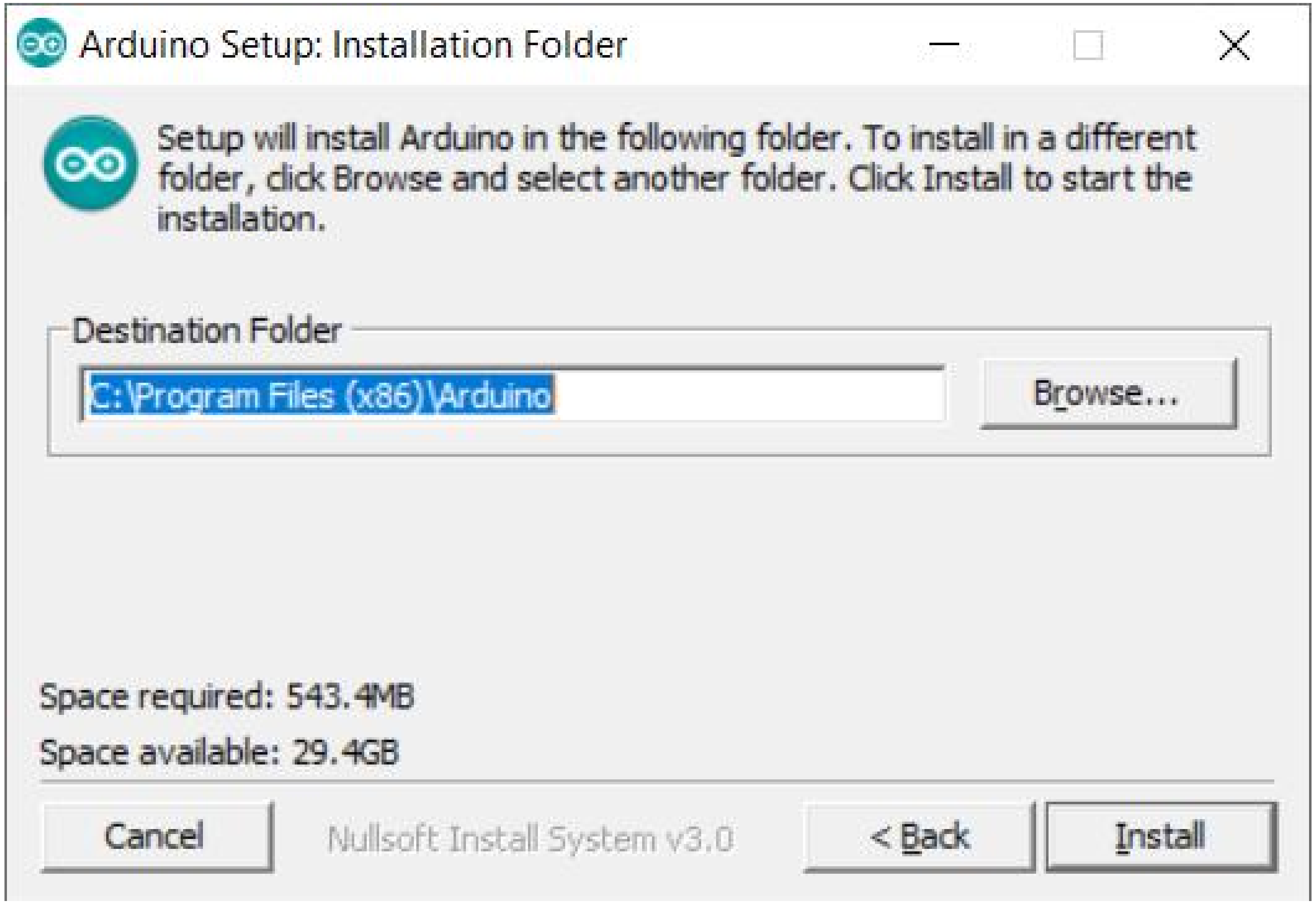
☑ Create Start Menu shortcut

☑ Create Desktop shortcut

☑ Associate .ino files

Space required: 543.4MB

Click Next


Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

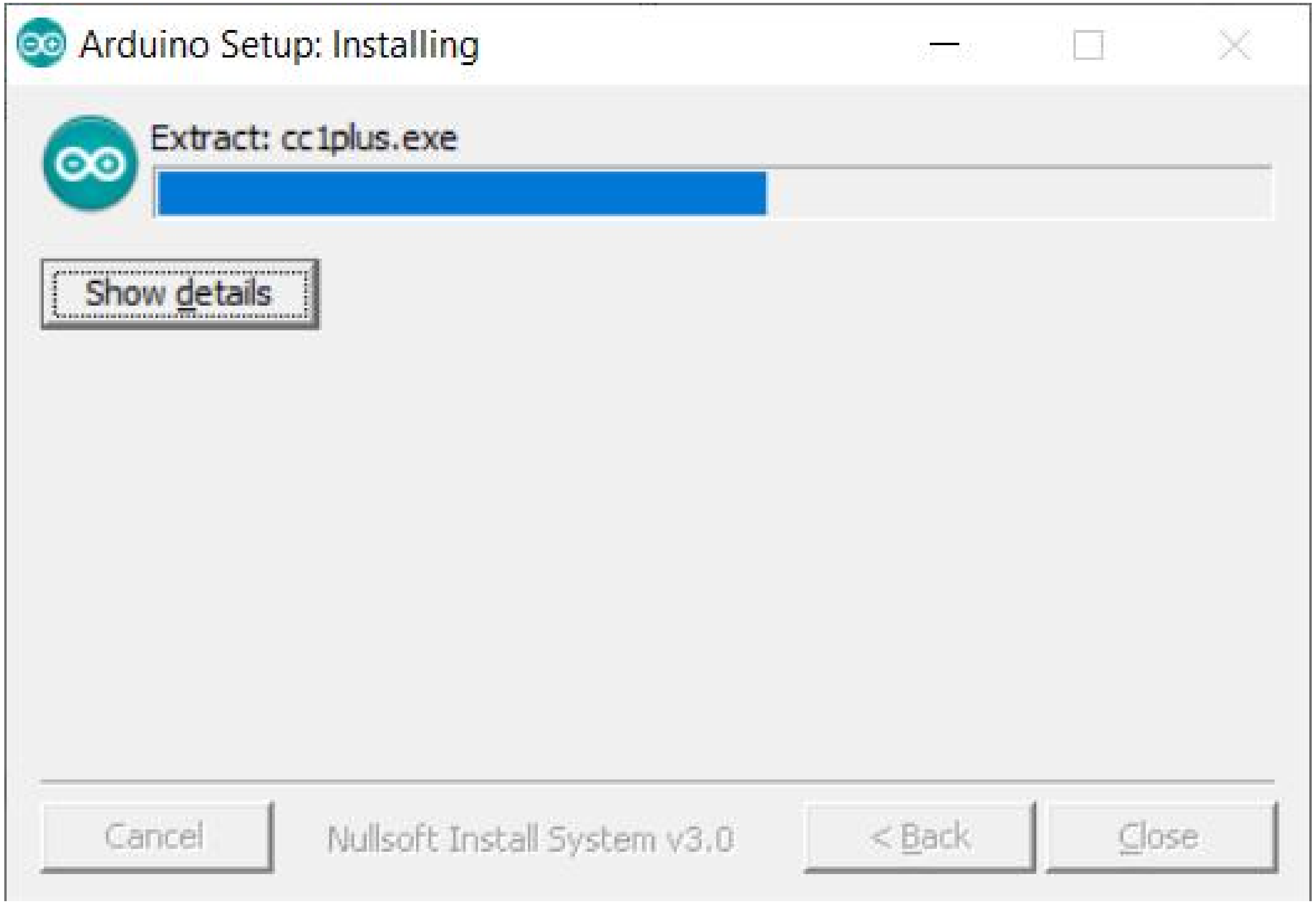
C:\Program Files (x86)\Arduino

Space required: 543.4MB

Space available: 29.4GB

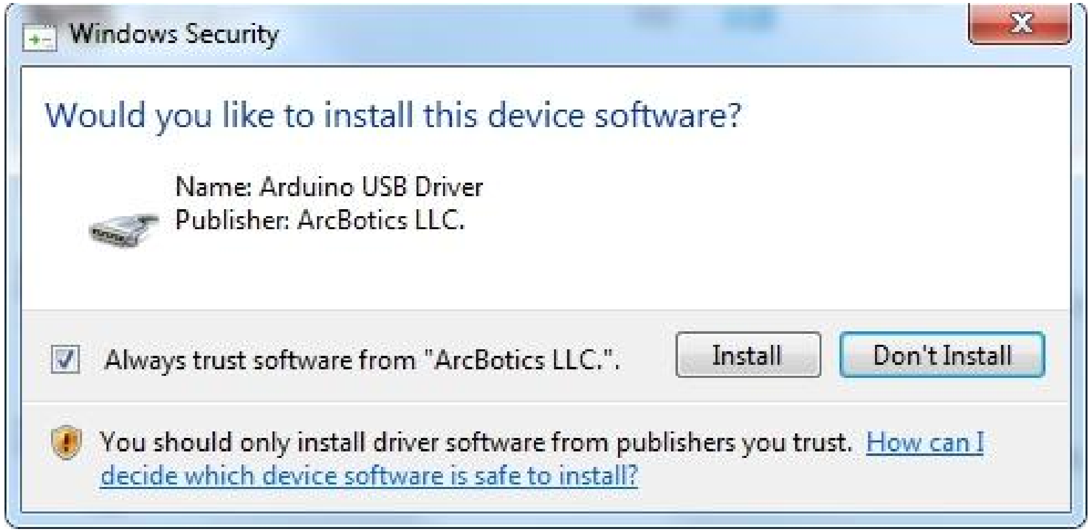
Cancel
Nullsoft Install System v3.0
< Back
Install

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



Extract: cc1plus.exe

Wait for the installing process, then click Close.
if appear the interface of Window Security, just continue to click
Install to finish the installation.



Windows Security

Would you like to install this device software?

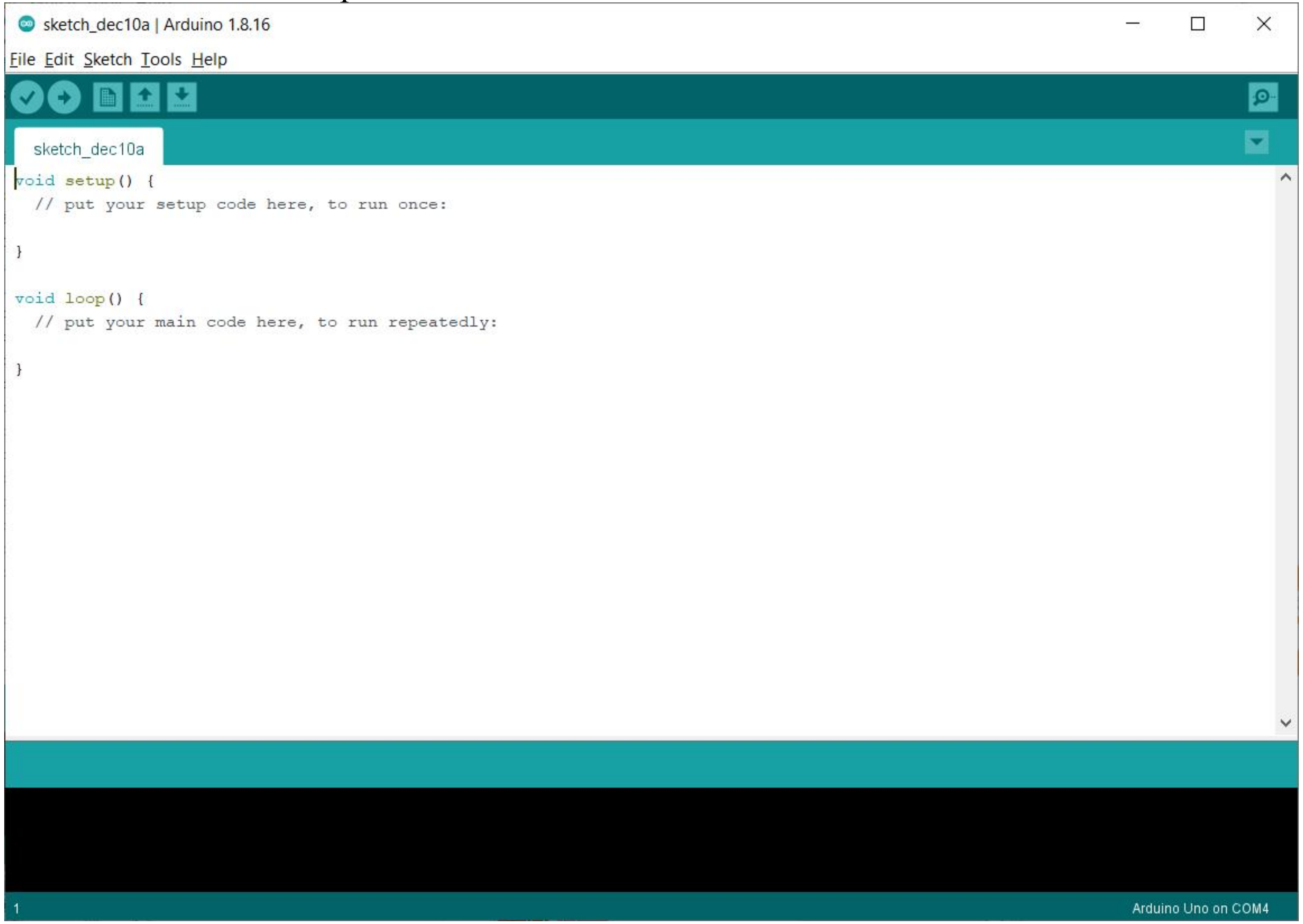
Name: Arduino USB Driver
Publisher: ArcBotics LLC.

☑ Always trust software from "ArcBotics LLC.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Next, the following icon appears on the desktop

Double-click to enter the desired development environment



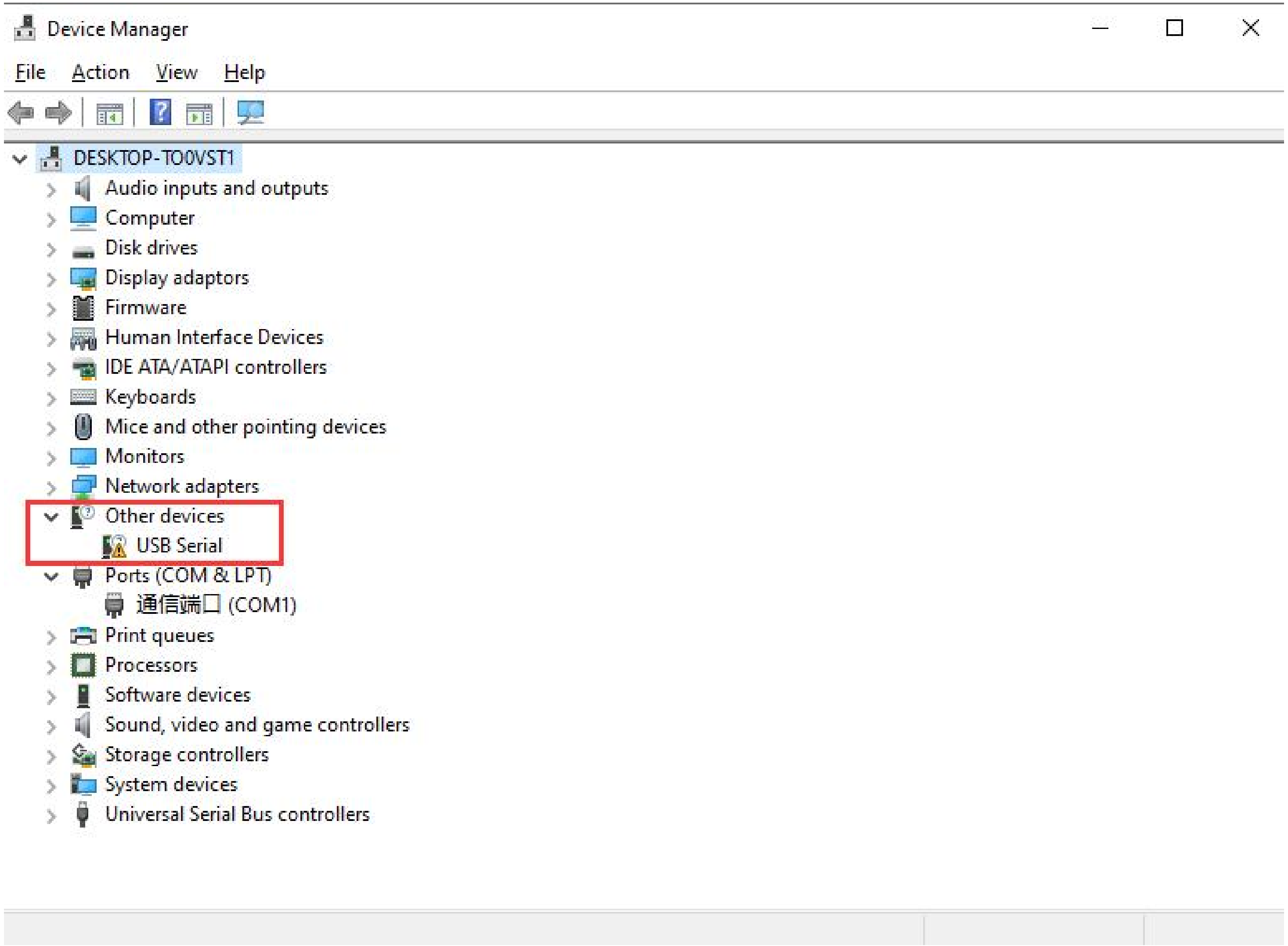
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Lesson 2 How to Install Arduino Driver

Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let’s move on to the driver installation in the Window system.

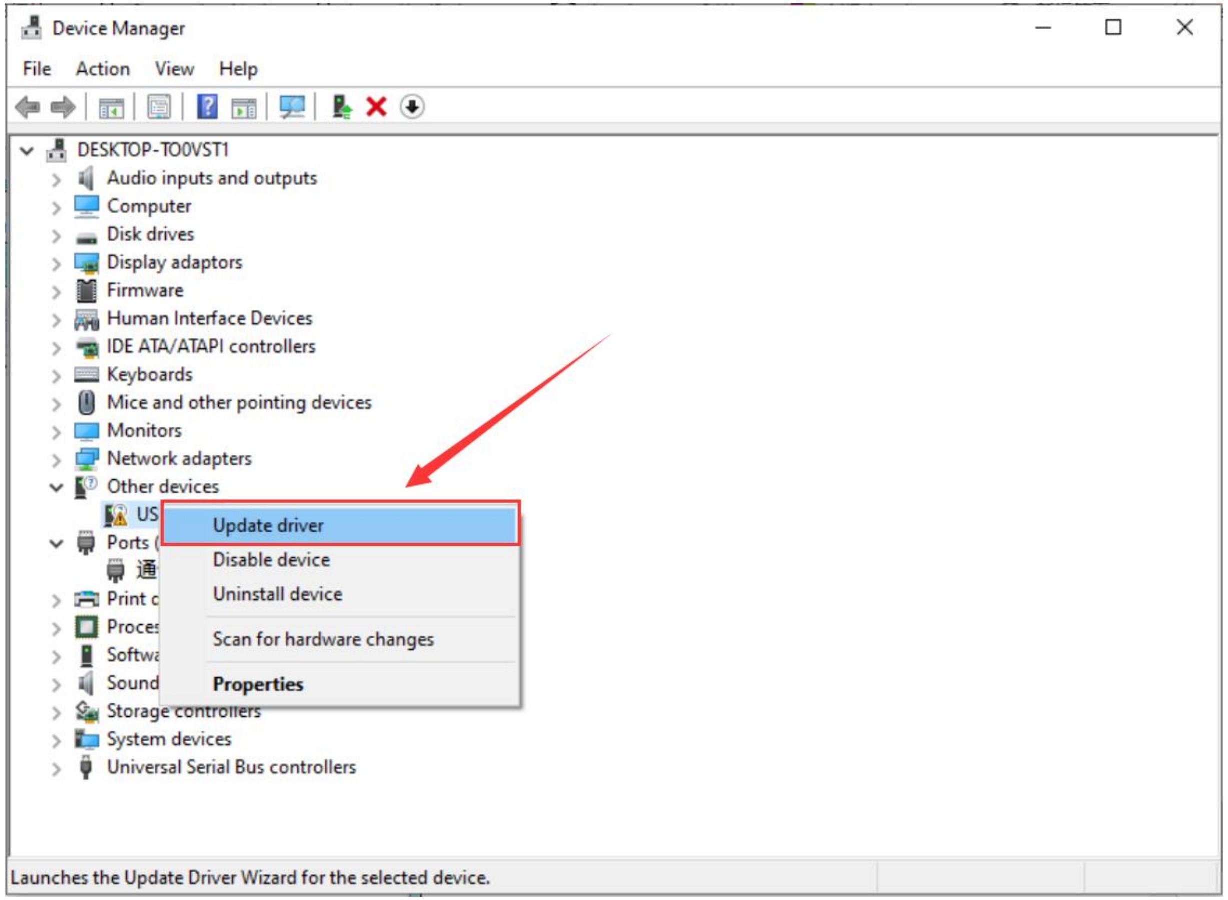
The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

When you connect UNO board to your computer at the first time, right click the icon of your
“Computer” —>for “Properties”—> click the “Device manager” ,
under “Other Devices”or“USB-Serial”, you should see an icon for “Unknown device” with a little yellow warning triangle next to it. This is your Arduino.Or you can search for "device" in your computer, or you can open the device manager of your computer.



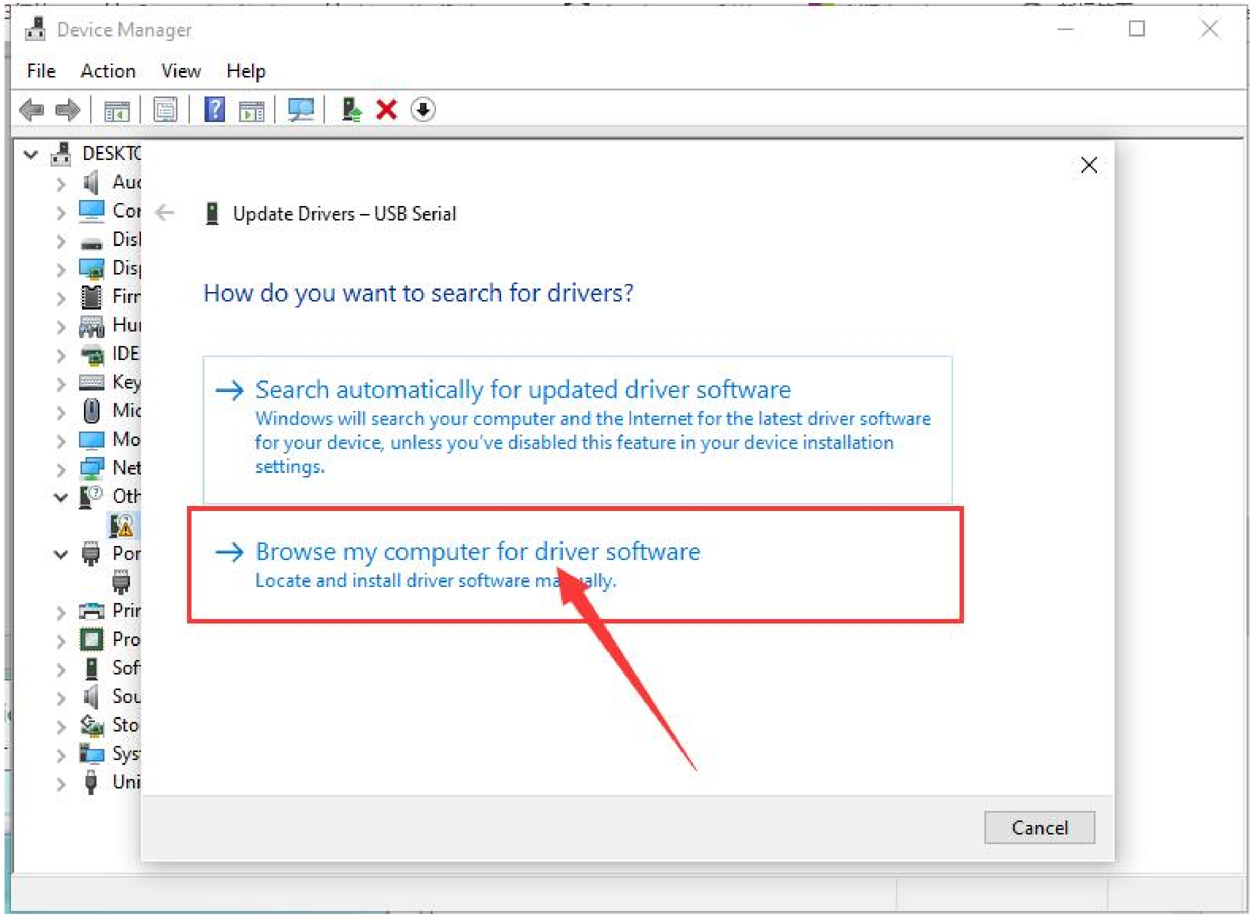
- DESKTOP-TO0VST1
  - Audio inputs and outputs
  - Computer
  - Disk drives
  - Display adaptors
  - Firmware
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
    - USB Serial
  - Ports (COM & LPT)
    - 通信端口 (COM1)
  - Print queues
  - Processors
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



Launches the Update Driver Wizard for the selected device.

Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



Update Drivers – USB Serial

How do you want to search for drivers?

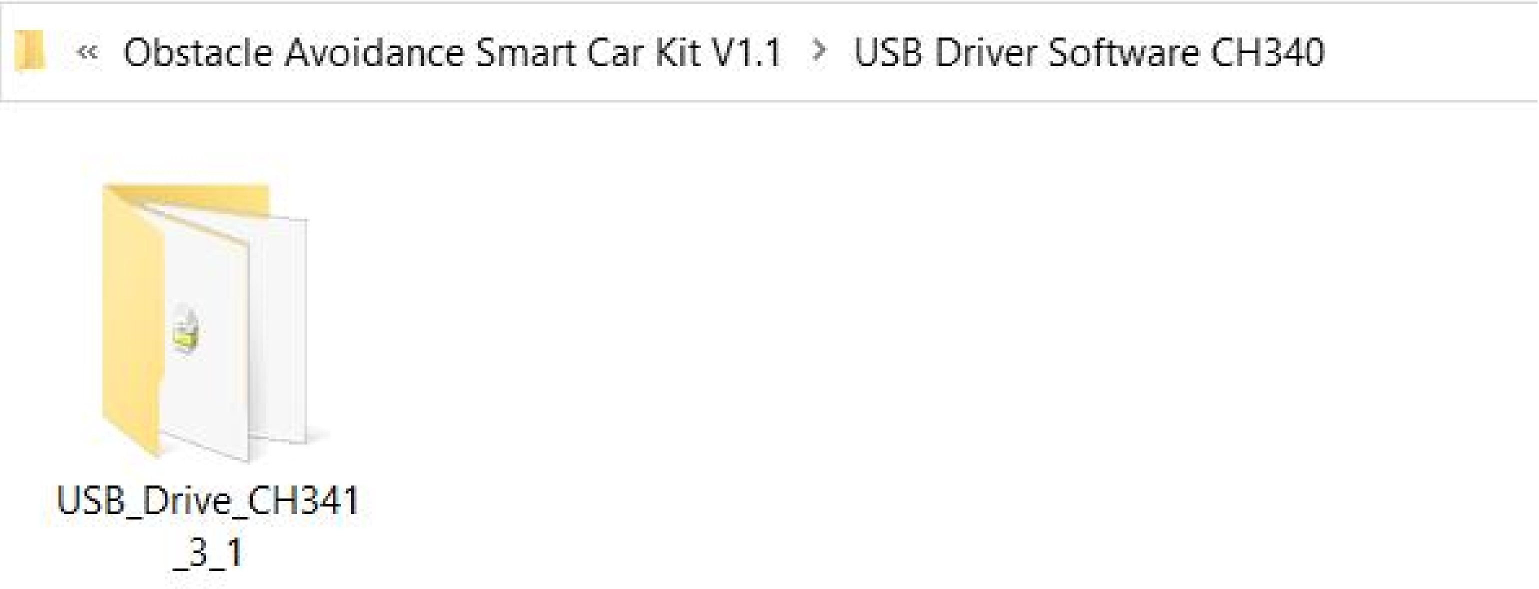
→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software
Locate and install driver software manually.

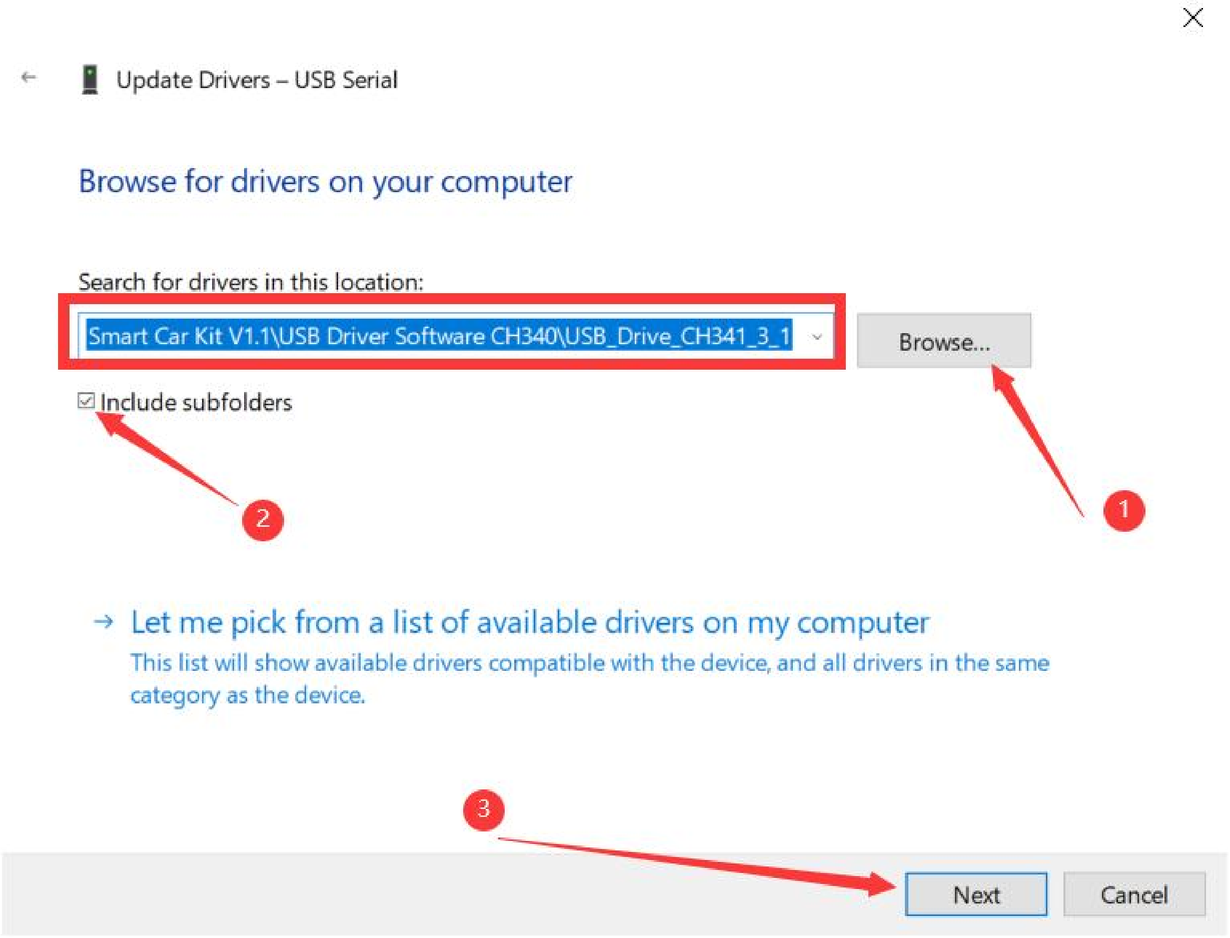
Right-click on the device and select the top menu option (Update Driver Software...).

You will then be prompted to either ‘Search Automatically for updated driver software’ or ‘Browse my computer for driver software’.

The CH340 driver software file is provided in the CD. I believe you have copied or downloaded the reference file we provide to your computer. The path needs to select the path where the "USB_Drive_CH341_3_1" folder is stored on your personal computer.


« Obstacle Avoidance Smart Car Kit V1.1 › USB Driver Software CH340
USB_Drive_CH341_3_1

Select the option to browse and navigate to the :**E:\Obstacle Avoidance Smart Car Kit V1.1\USB Driver Software CH340\USB_Drive_CH341_3_1.**(For example, I copied the reference file “Obstacle Avoidance Smart Car Kit V1.1” of the CD to Disk E, so the driver search location is E:\\Obstacle Avoidance Smart Car Kit V1.1\USB Driver Software CH340\USB_Drive_CH341_3_1)



Update Drivers – USB Serial

Browse for drivers on your computer

Search for drivers in this location:

Smart Car Kit V1.1\USB Driver Software CH340\USB_Drive_CH341_3_1

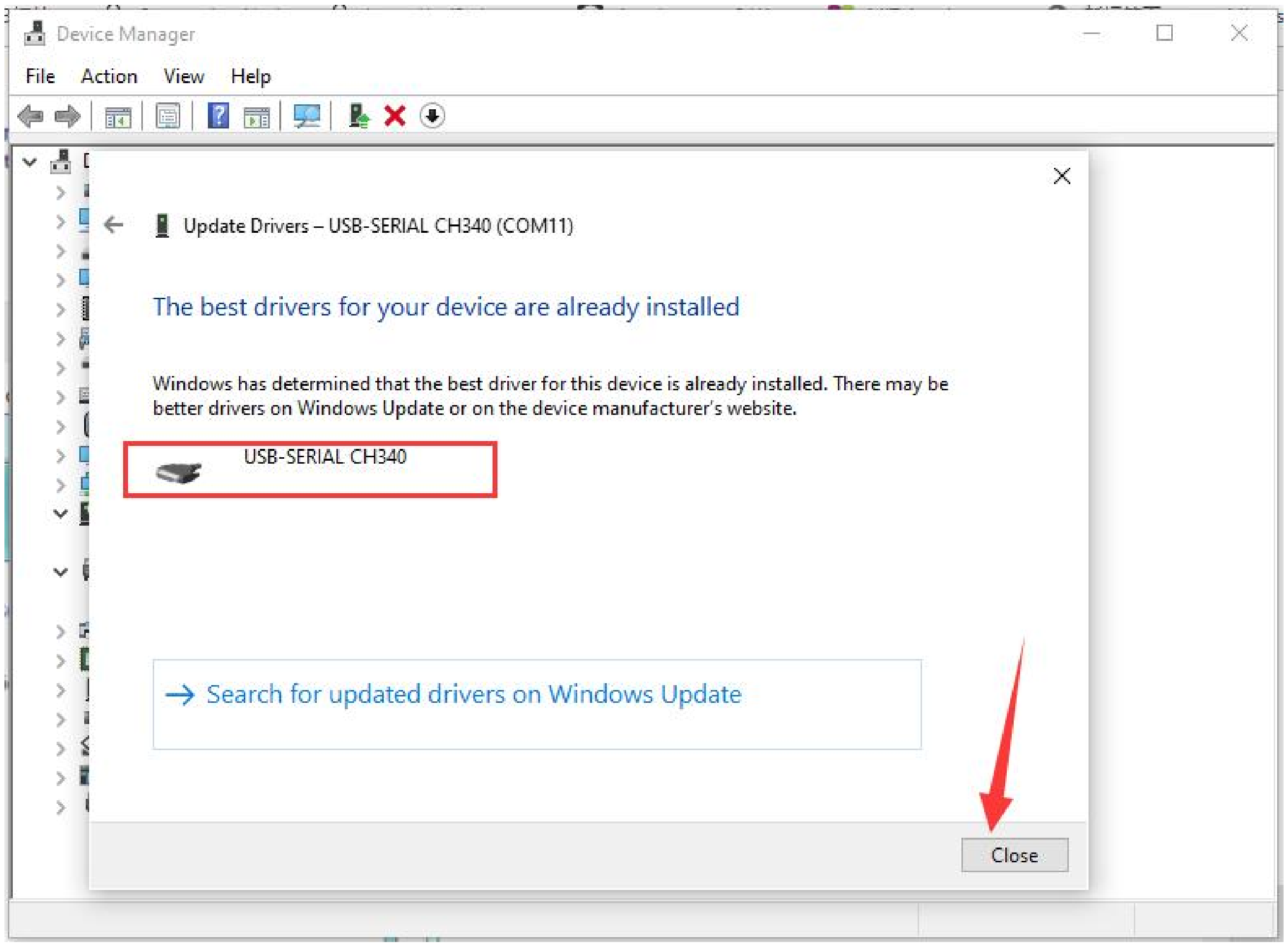
☑ Include subfolders

→ Let me pick from a list of available drivers on my computer

This list will show available drivers compatible with the device, and all drivers in the same category as the device.

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. Installation completed,click “Close”.



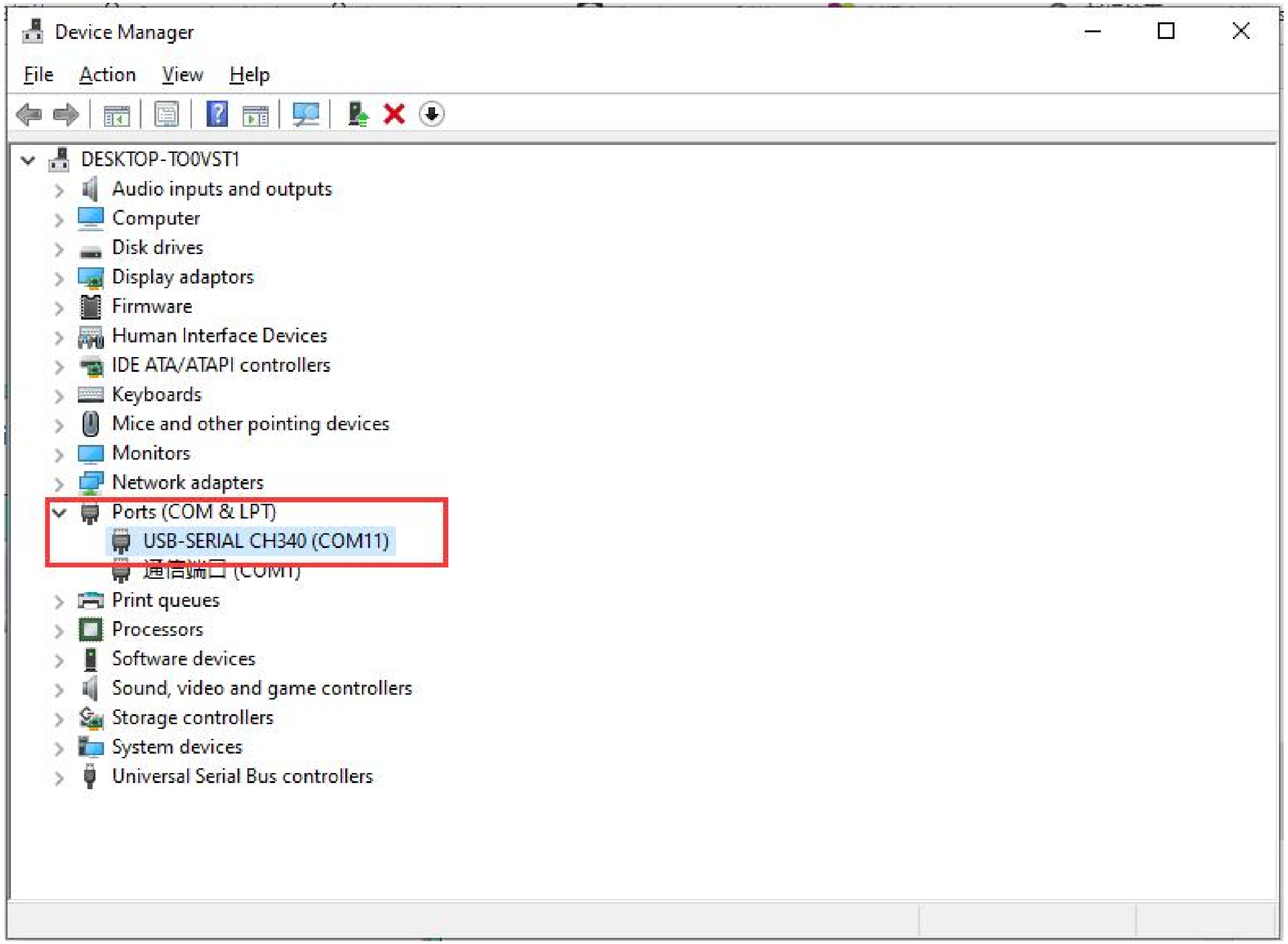
Update Drivers – USB-SERIAL CH340 (COM11)

The best drivers for your device are already installed

Windows has determined that the best driver for this device is already installed. There may be better drivers on Windows Update or on the device manufacturer's website.

USB-SERIAL CH340

Up to now, the driver is installed well. Then you can right click “Computer”—>“Properties”—>“Device manager” , you should see the device as the figure shown below.



DESKTOP-TO0VST1
- Audio inputs and outputs
- Computer
- Disk drives
- Display adaptors
- Firmware
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - USB-SERIAL CH340 (COM11)
  - 通信端口 (COM1)
- Print queues
- Processors
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

Lesson 3 How to 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?

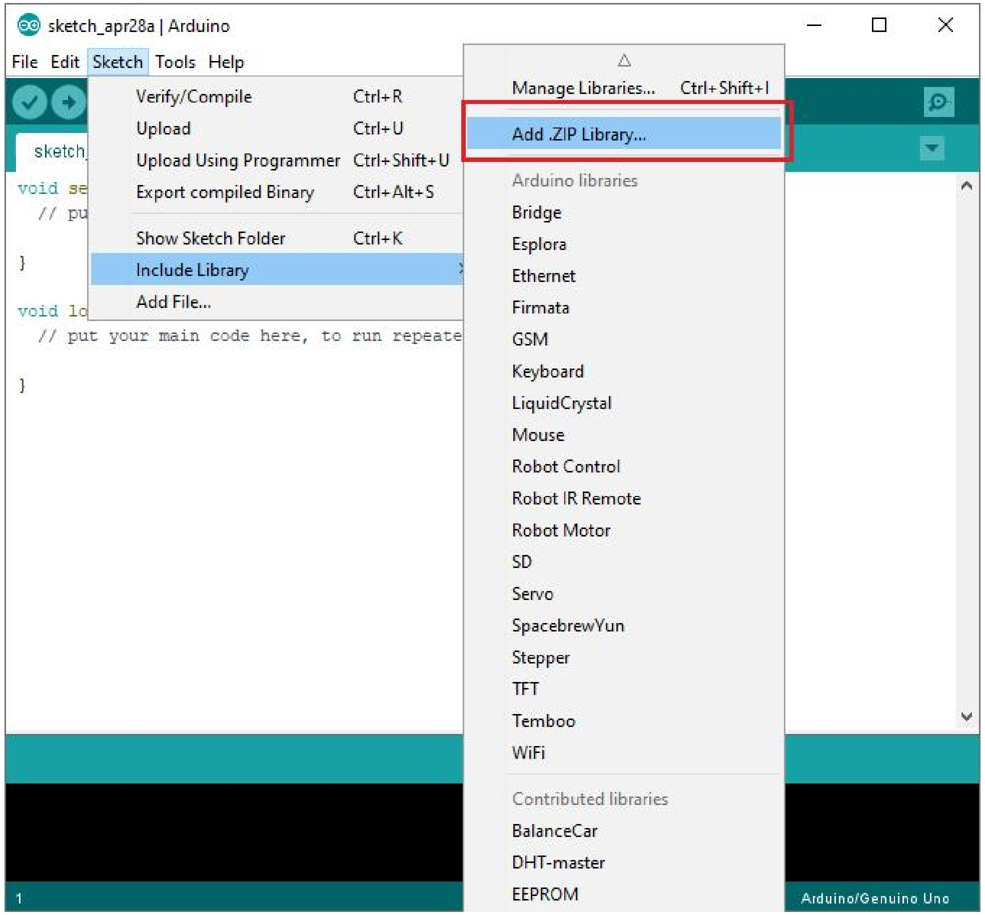
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in 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

Importing a .zip Library

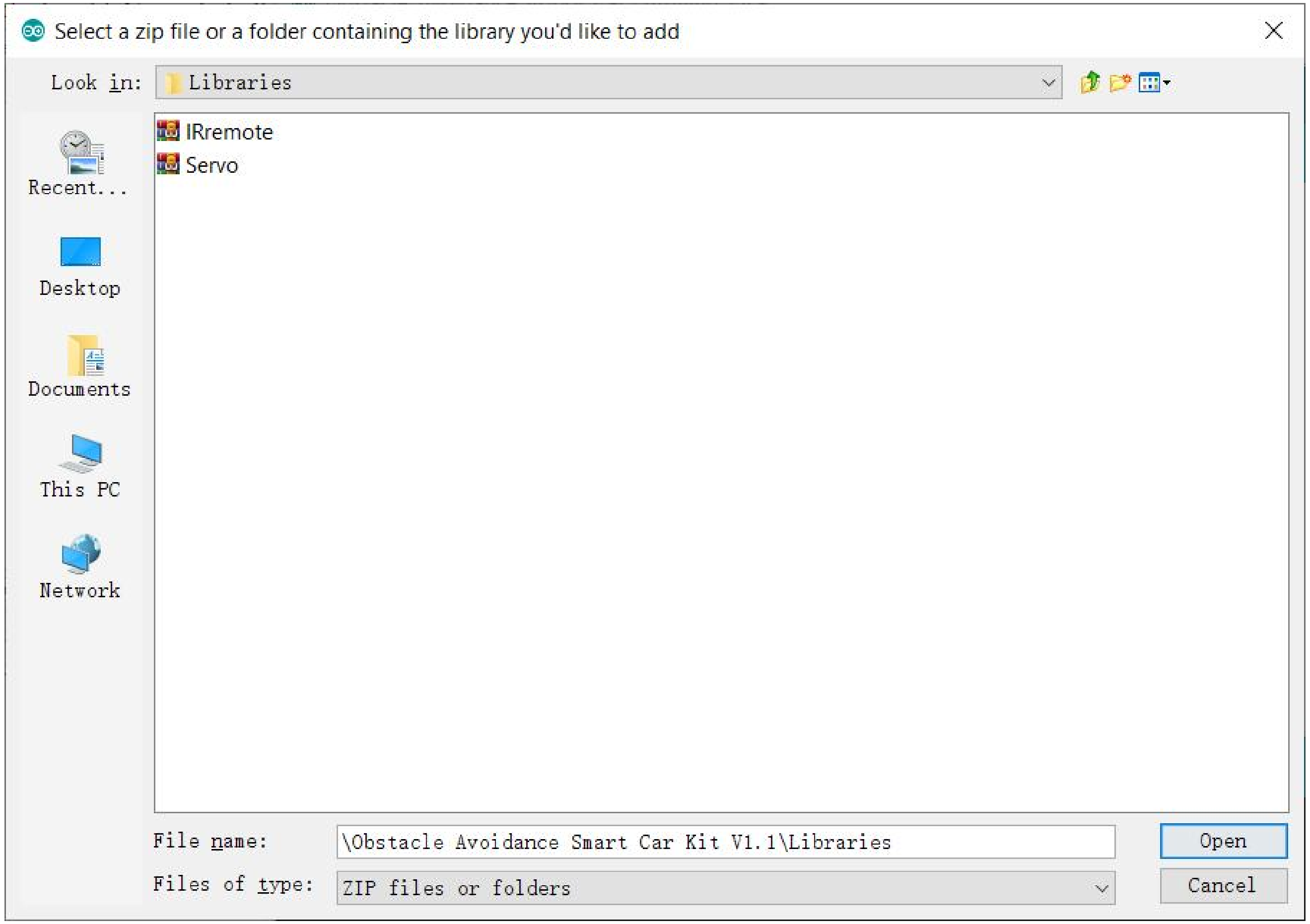
Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. you can install 3rd party libraries in the IDE. Do not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



```
void se
  // pu

}

void lo
  // put your main code here, to run repeate

}
```

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

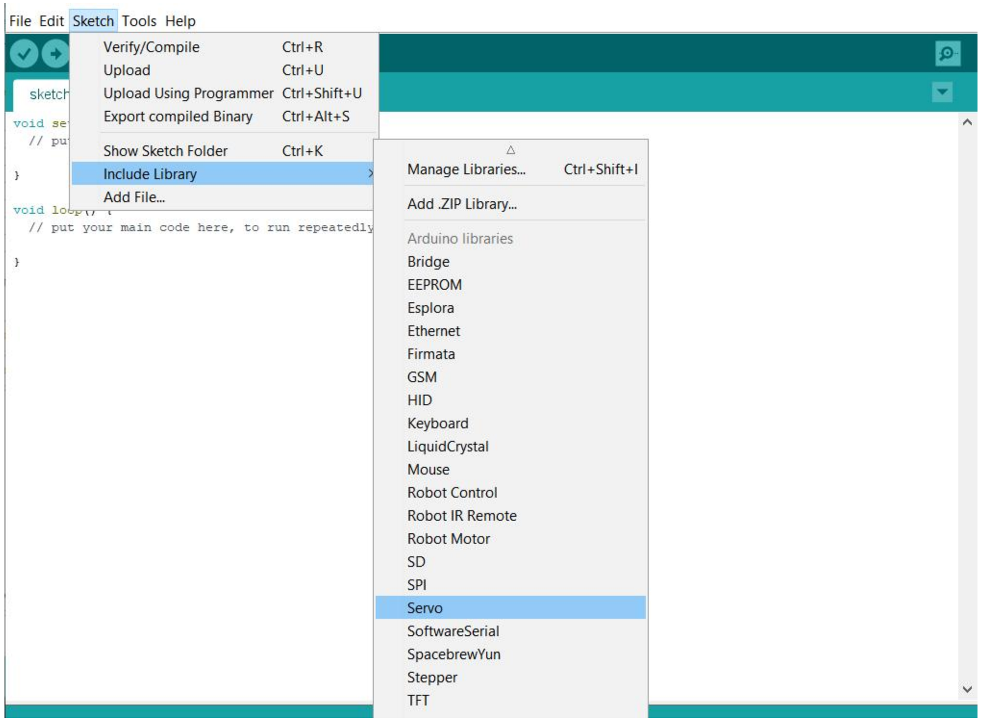
File name: \Obstacle Avoidance Smart Car Kit V1.1\Libraries

Files of type: ZIP files or folders

(Note: This is just to demonstrate how to add a zip library file, whether you need to add a library file depends on your actual program needs)

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.



```
void se
  // pu

}

void loop() {
  // put your main code here, to run repeatedly

}
```

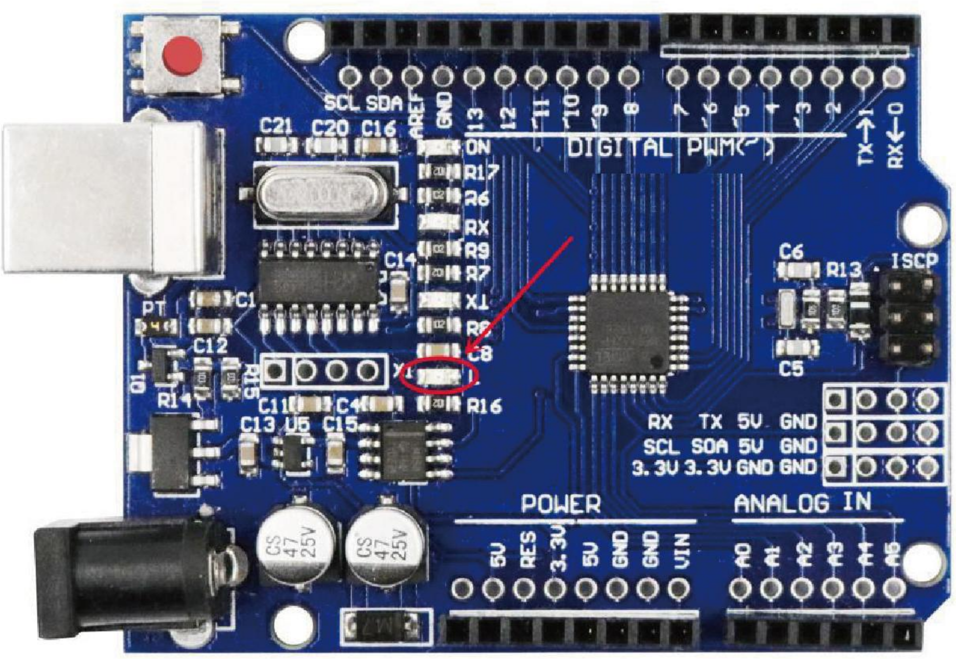
Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

Lesson 4 Blink Test(Test your first program)

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

The UNO 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 board and is often referred to as the 'L' LED as this is how it is labeled on the board.



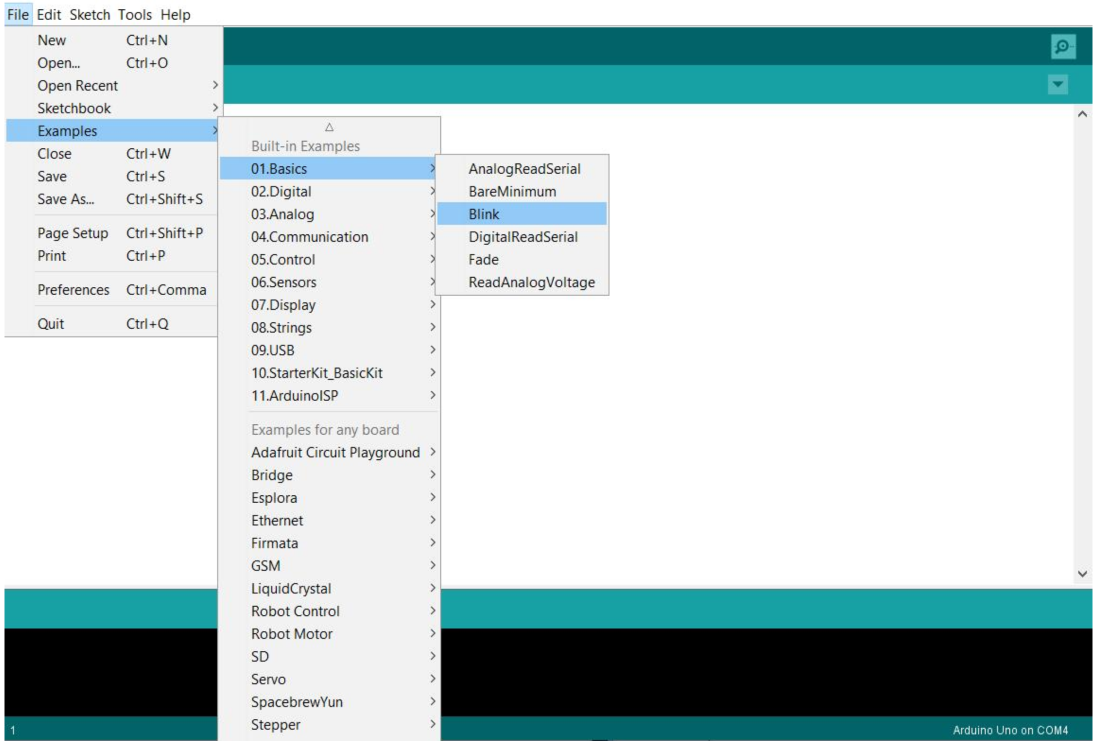
You may find that your UNO board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

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

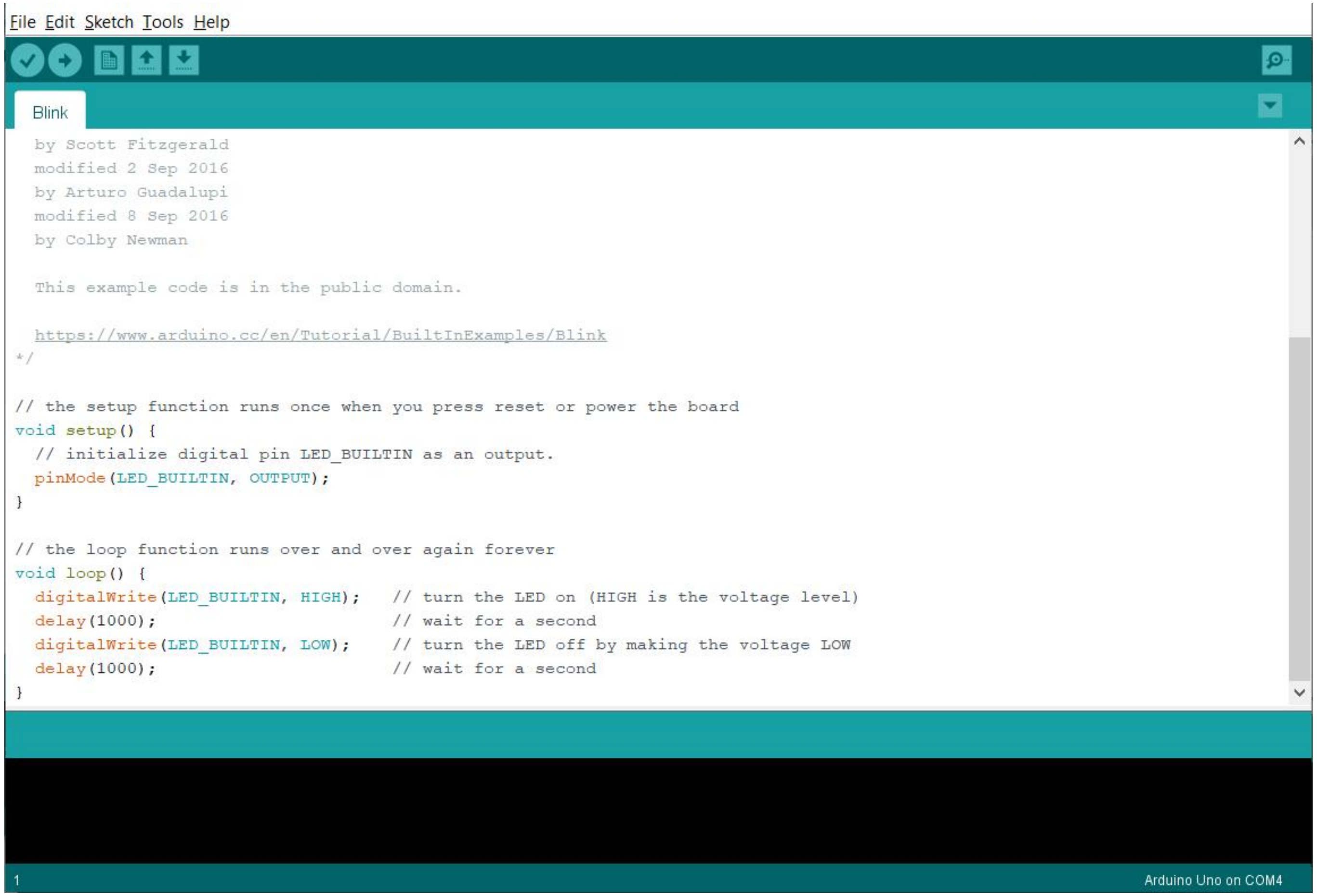
In the previous chapter-How to install Arduino IDE, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO board. The time has now come to put that connection to the test and program your UNO board.

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



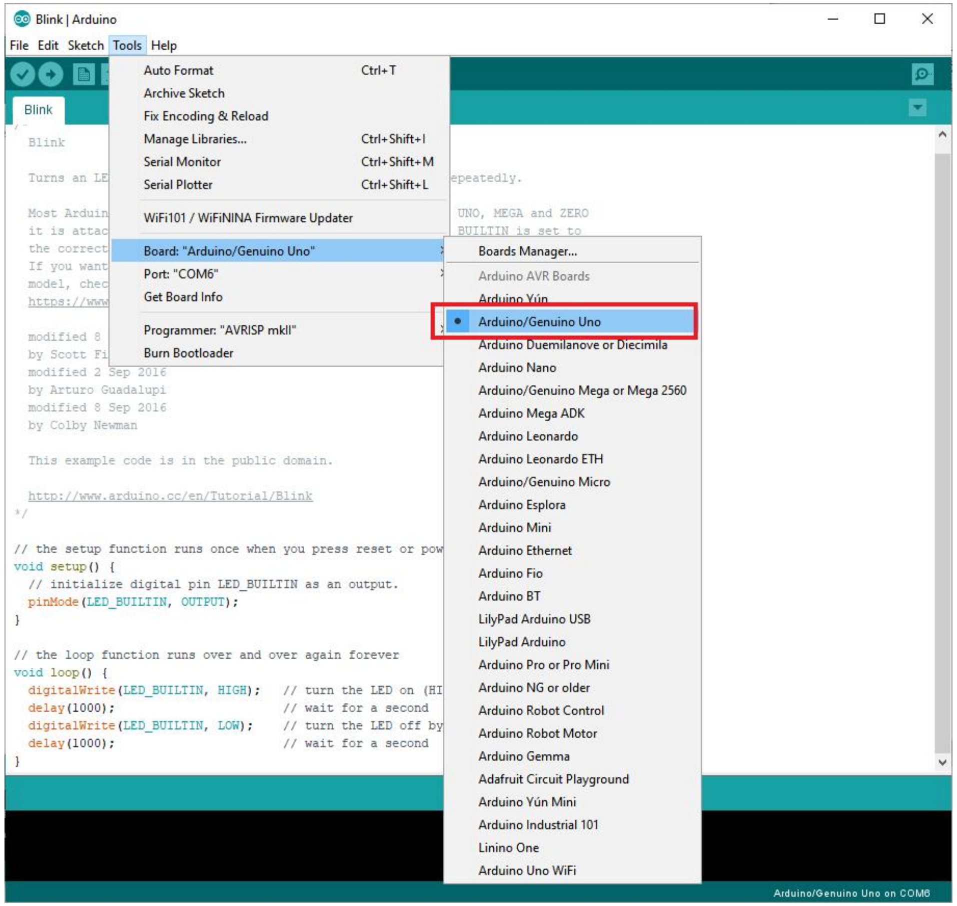
When the sketch window opens, enlarge it so that you can see the entire sketch in the window.



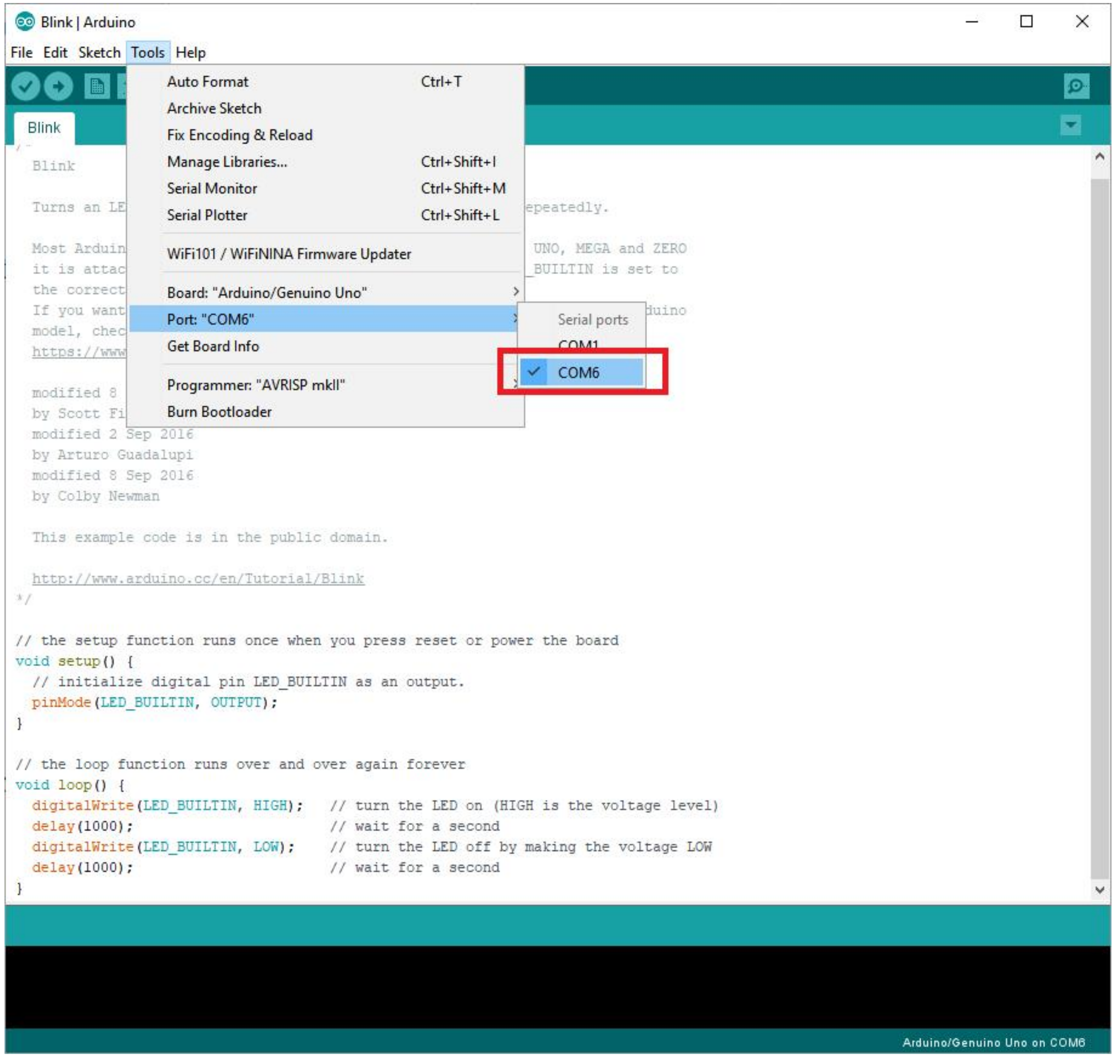
```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// 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
}
```

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.

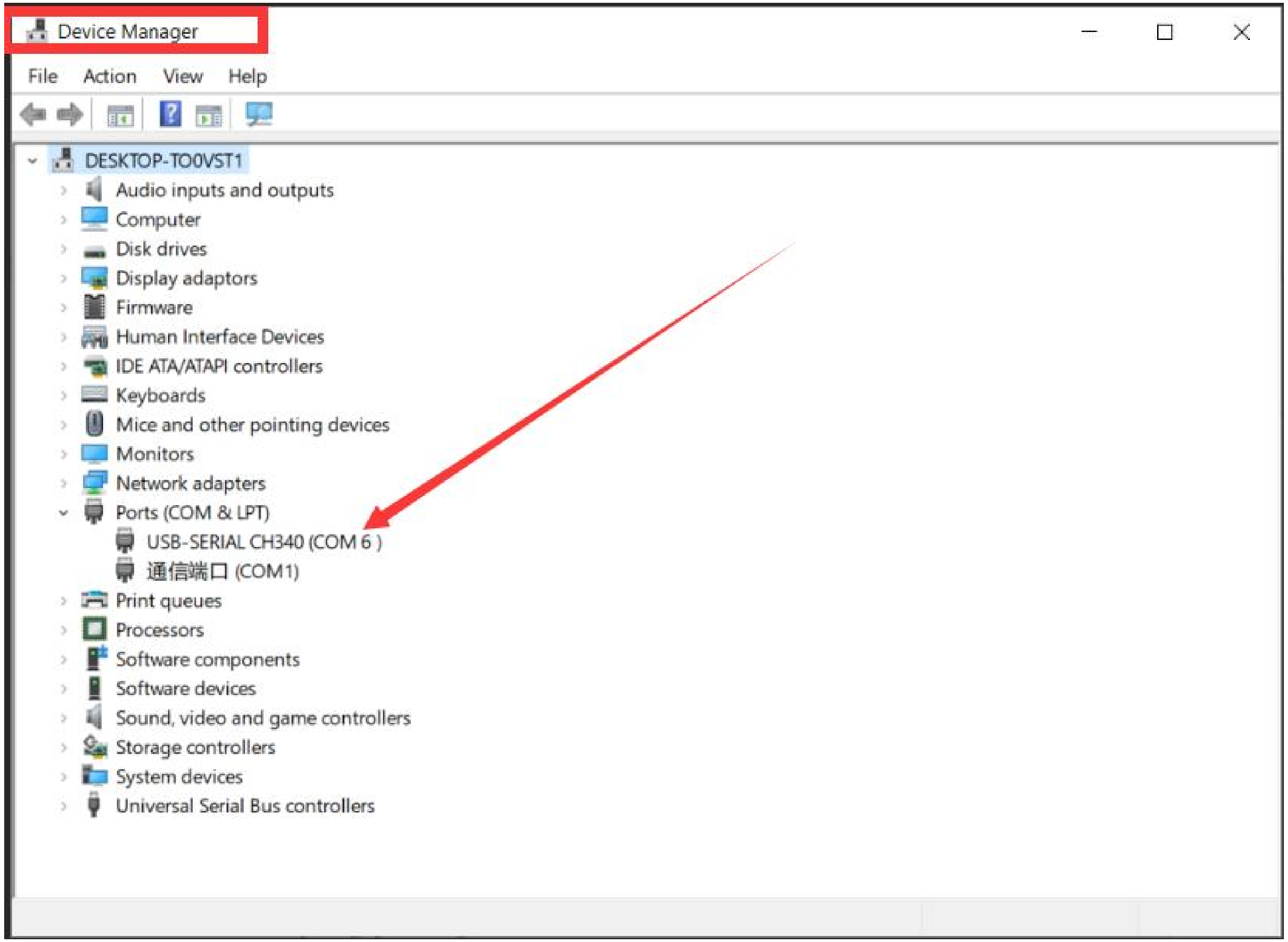
Since we are going to change this sketch, the first thing you need to do is save your own copy.
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.
Attach your Arduino board to your computer with the USB cable and check that the
'Board Type' and 'Serial Port' are set correctly.



```
/*
  Blink

  Turns an LE                                     epeatedly.

  Most Arduin                                      UNO, MEGA and ZERO
  it is attac                                      BUILTIN is set to
  the correct
  If you want
  model, chec
  https://www

  modified 8
  by Scott Fi
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or pow
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 (HI
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by
  delay(1000);                       // wait for a second
}
```


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
Blink
Arduino/Genuino Uno on COM6

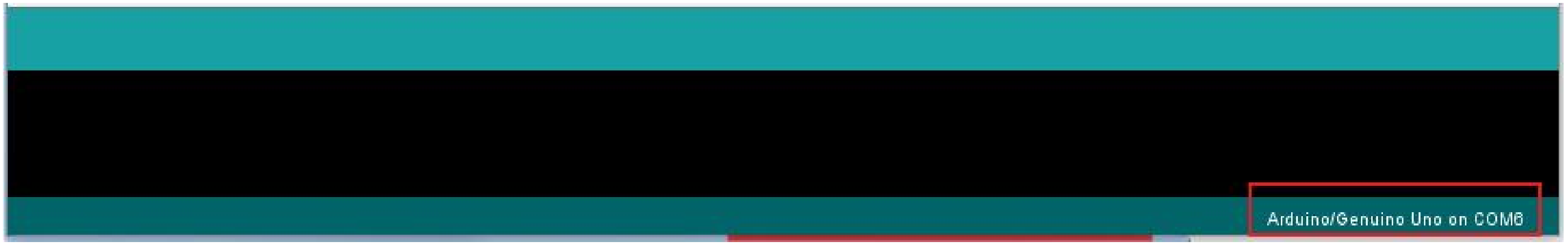
```
/*
  Blink

  Turns an LE                                   epeatedly.

  Most Arduin                                    UNO, MEGA and ZERO
  it is attac                                    _BUILTIN is set to
  the correct
  If you want                                                duino
  model, chec
  https://www

  modified 8
  by Scott Fi
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// 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
}
```



```
DESKTOP-TO0VST1
  Audio inputs and outputs
  Computer
  Disk drives
  Display adaptors
  Firmware
  Human Interface Devices
  IDE ATA/ATAPI controllers
  Keyboards
  Mice and other pointing devices
  Monitors
  Network adapters
  Ports (COM & LPT)
    USB-SERIAL CH340 (COM 6 )
    通信端口 (COM1)
  Print queues
  Processors
  Software components
  Software devices
  Sound, video and game controllers
  Storage controllers
  System devices
  Universal Serial Bus controllers
```

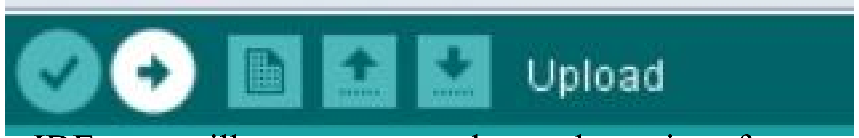
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using UNO, then you will have to choose Arduino/Genuino Uuo as the Board Type, other choices can be made in the same manner. And the Serial Port displayed for everyone is different, despite COM 6 chosen 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.You can check the serial number of your Arduino uno development board in your device manager

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



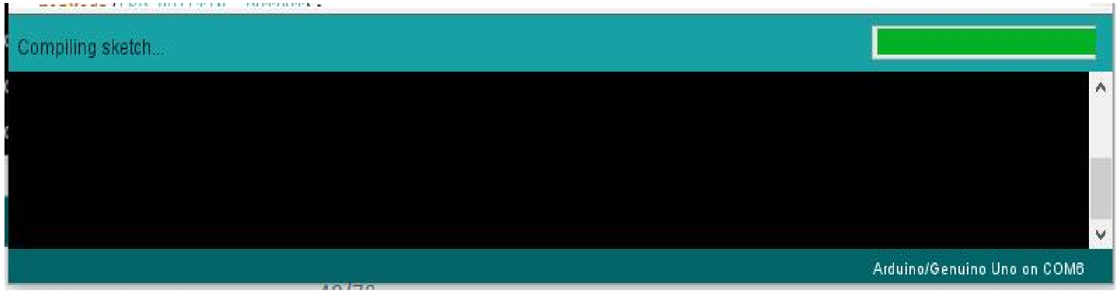
Arduino/Genuino Uno on COM6

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.

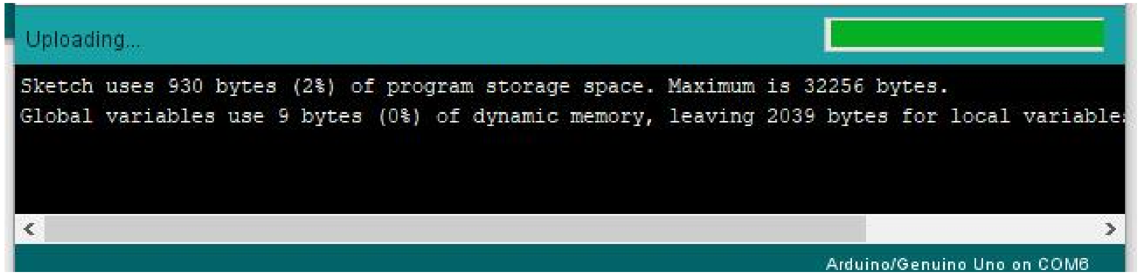
]



Compiling sketch...

Arduino/Genuino Uno on COM6

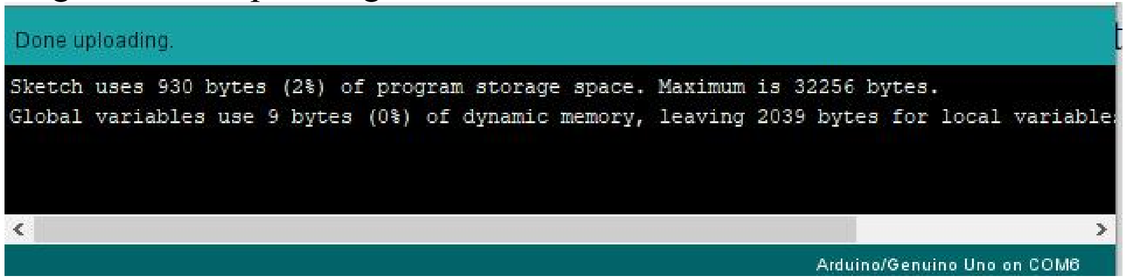
Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Uploading...

```
Sketch uses 930 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variable:
```

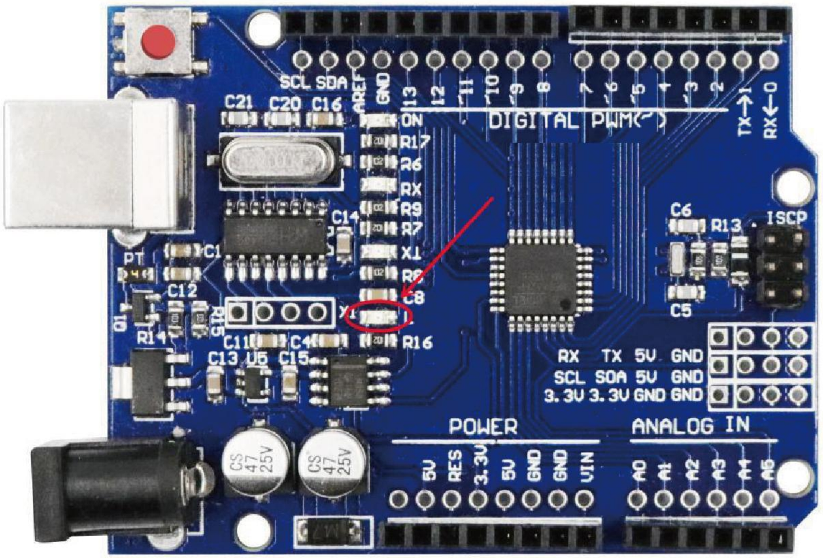
Finally, the status will change to 'Done uploading'.

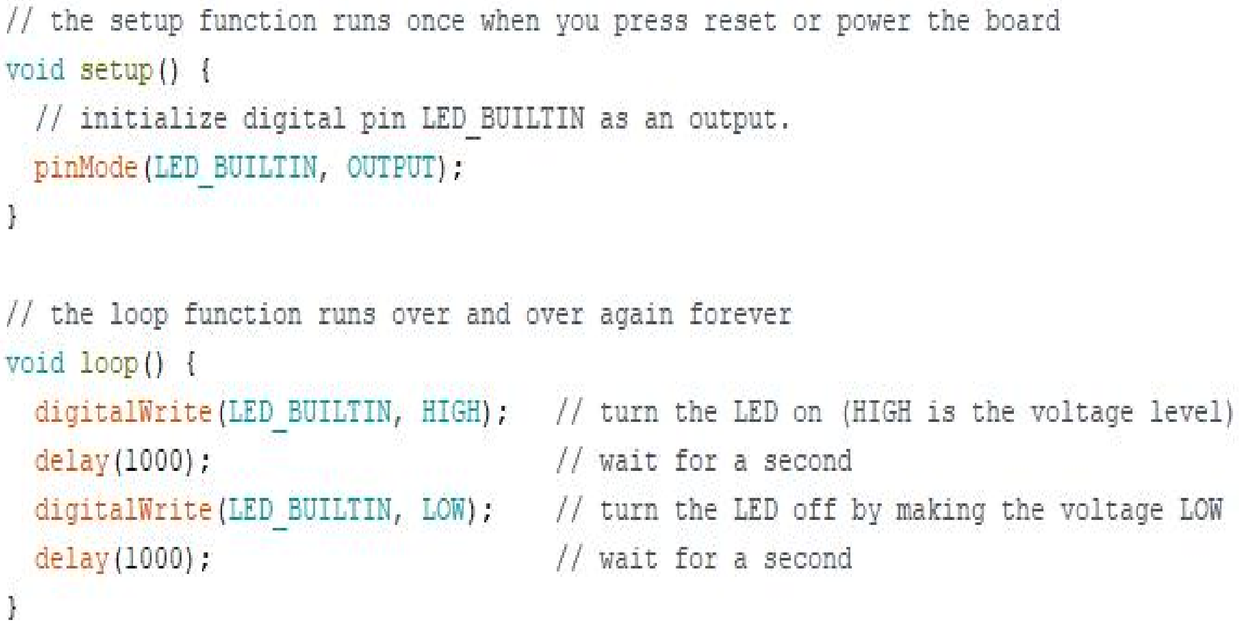


Done uploading.

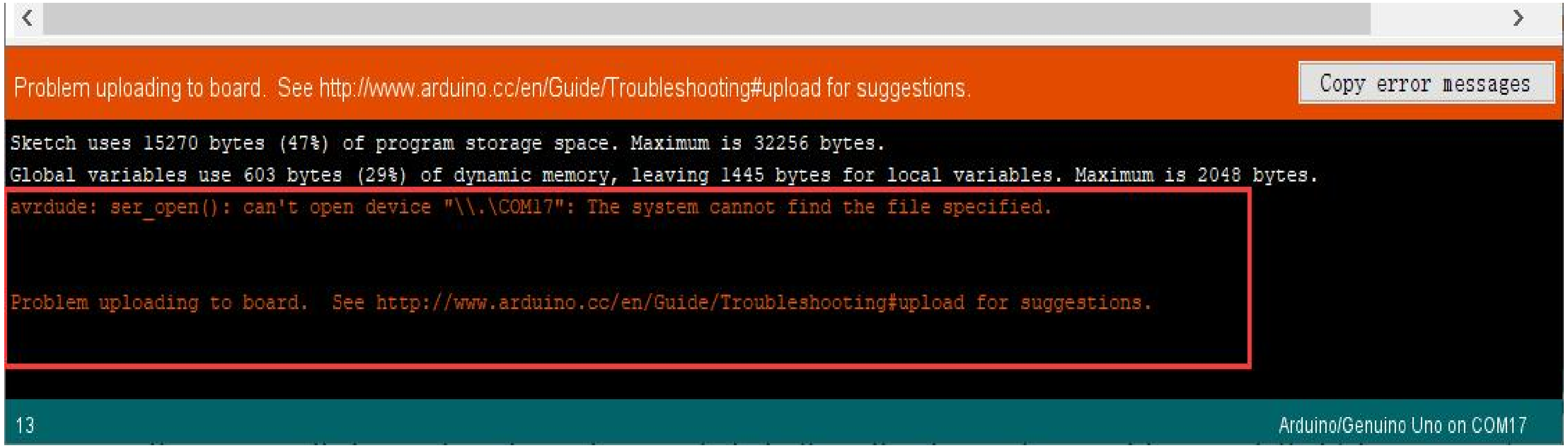
```
Sketch uses 930 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variable:
```

After uploading the code, you will see that the "L" LED on the Arduino UNO motherboard will light up for 1 second, and then turn off for 1 second, and the process will continue to loop.




```
// 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
}
```

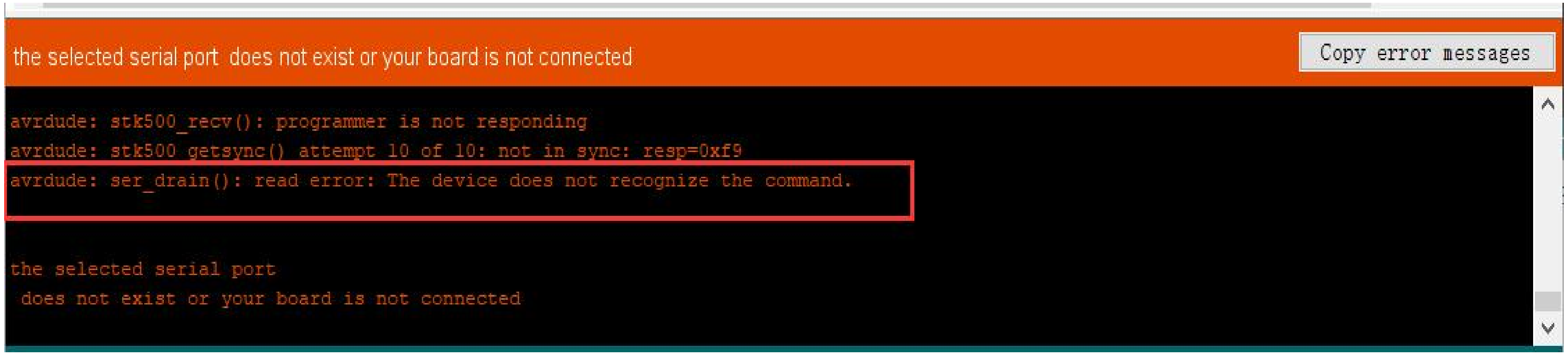
If you are missing some steps, Arduino IDE may report some errors. You can refer to the following solutions.



Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.

```
Sketch uses 15270 bytes (47%) of program storage space. Maximum is 32256 bytes.
Global variables use 603 bytes (29%) of dynamic memory, leaving 1445 bytes for local variables. Maximum is 2048 bytes.
avrdude: ser_open(): can't open device "\\.\COM17": The system cannot find the file specified.


Problem uploading to board.  See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
```

(1) It means 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.How to Install Arduino Driver

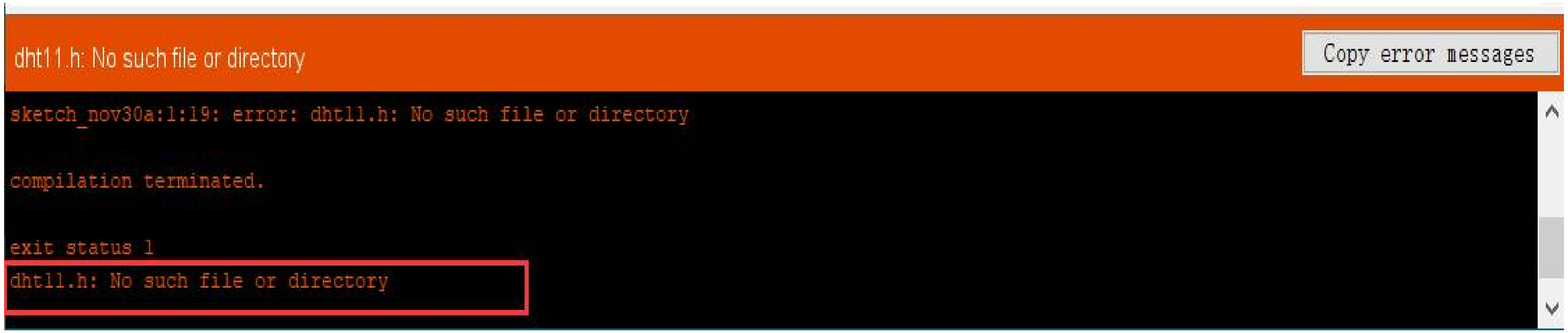


the selected serial port does not exist or your board is not connected

```
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0xf9
avrdude: ser_drain(): read error: The device does not recognize the command.


the selected serial port
 does not exist or your board is not connected
```

(2)It means that the communication serial port for uploading code is occupied. If you connect the Bluetooth module when uploading the code, you need to disconnect the Bluetooth module from the Arduino UNO expansion board, and then reinstall the Bluetooth module after uploading.

Another possibility is that the USB interface of the Arduino UNO is insufficiently powered, and you need to turn on the external power switch of the expansion board.



dht11.h: No such file or directory

```
sketch_nov30a:1:19: error: dht11.h: No such file or directory

compilation terminated.

exit status 1
dht11.h: No such file or directory
```

(3)If you need to use library files in some project programs, you may encounter this error.

It means that some library files are used in the code, but you did not add the corresponding library files to the Arduino IDE before uploading the code.How to Add Libraries.

Lesson 5 Servo

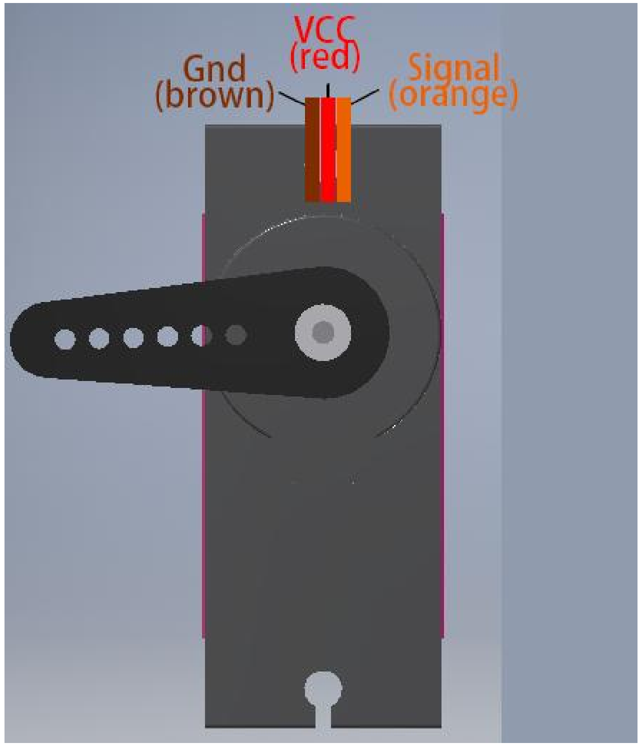
About this lesson:

In this lesson, you will learn how to control a servo motor using Arduino 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 A2.

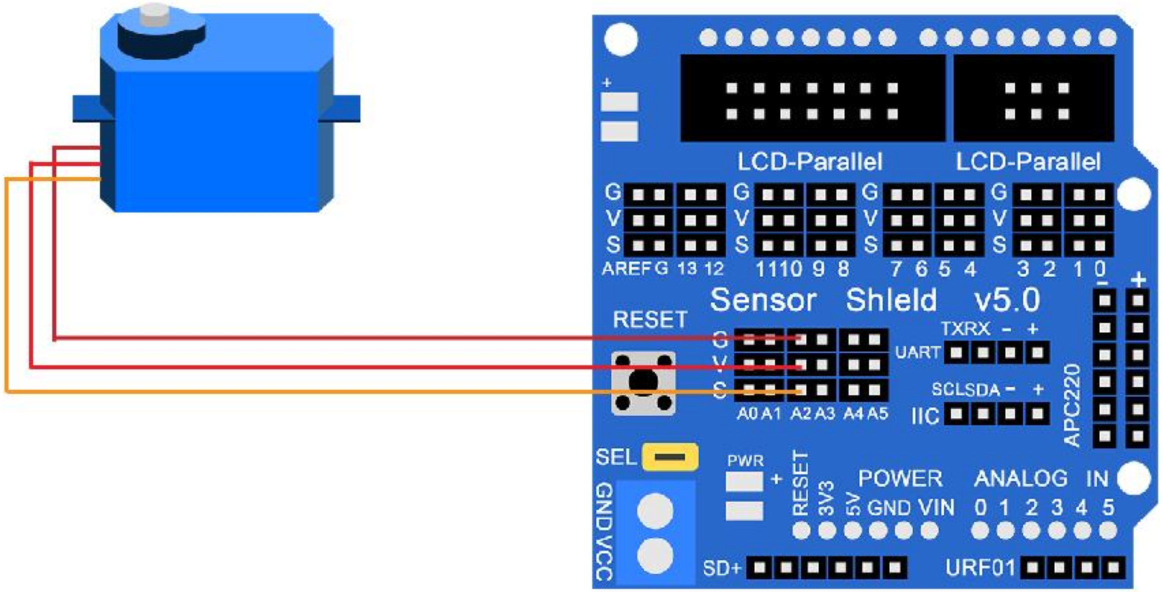
Introduction

Servo motor is a position control rotary actuator. It mainly consists of a housing, circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCUs or receivers and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and obtain the voltage difference output.When the motor speed is constant, the potentiometer is driven to rotate through the cascade reduction gear, which leads that the voltage difference is 0, and the motor stops rotating. Generally, the angle range of servo rotation is 0° --180 °.



In general, servo has three lines in brown, red and orange. The brown wire is grounded, the red one is a positive pole line and the orange one is a signal line.

Connection diagram



Code

Open this source program in Code>Lesson 5 Servo>Servo.ino.After uploading the code. The servo goes from 0° to 180 ° , and then from180 ° to 0°. Repeat this process in a loop.

See Lesson 4 Blink Test(Test your first program) 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 3 How to Add Libraries .

Lesson 6 Ultrasonic Sensor Module

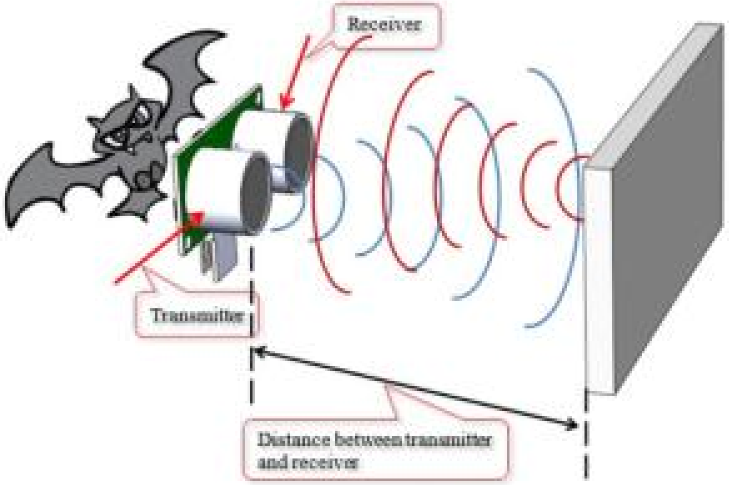
About this lesson:

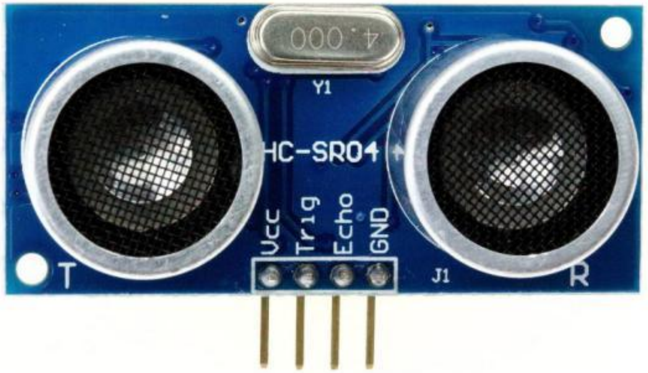
Ultrasonic sensor is great for all kind of projects that need distance measurements, avoiding obstacles as examples.

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

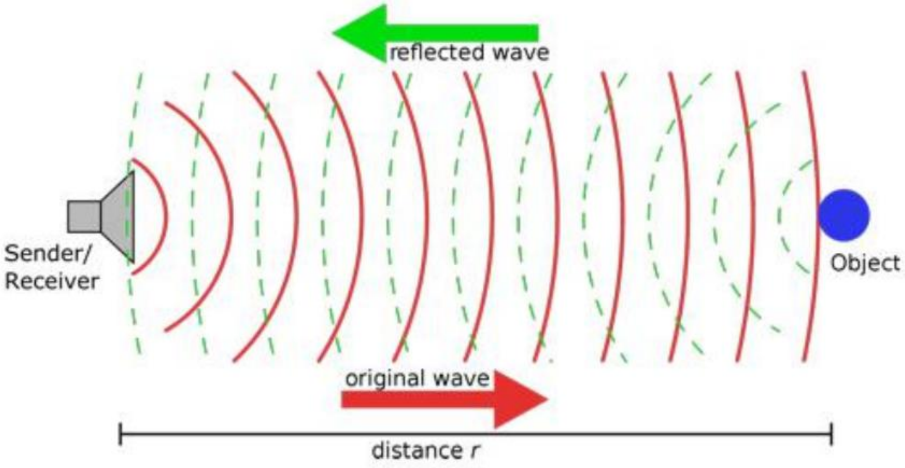
What is an ultrasonic sensor

Review the ultrasonic sensor from the previous lesson. It works like a bat's eye. Determine the distance of obstacles in front after receiving and receiving high-frequency sound waves.
As the following picture shown, it is our ultrasonic module. It has two something like eyes.
One is transmitting end, the other is receiving end.





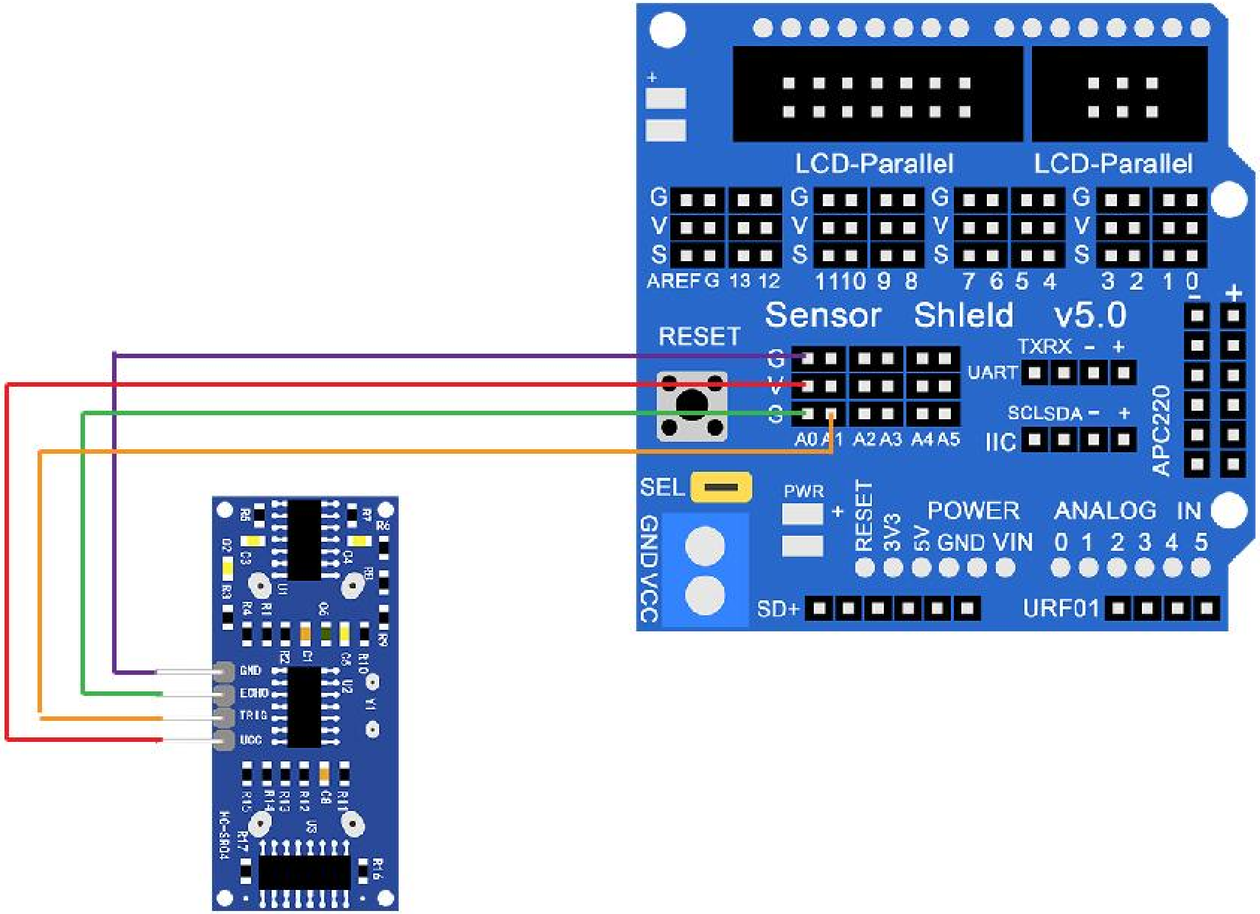
The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like what bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.The HC-SR04 or the ultrasonic sensor is being used in a wide range of electronics projects for creating obstacle detection and distance measuring application as well as various other applications. Here we have brought the simple method to measure the distance with arduino and ultrasonic sensor and how to use ultrasonic sensor with arduino.



Specification

Power Supply :+5V DC
Quiescent Current : <2mA
Working Current: 15mA
Effectual Angle: <15° 92
Ranging Distance : 2cm – 400 cm
Resolution : 0.3 cm
Measuring Angle: 30 degree
Trigger Input Pulse width: 10uS

Wiring diagram

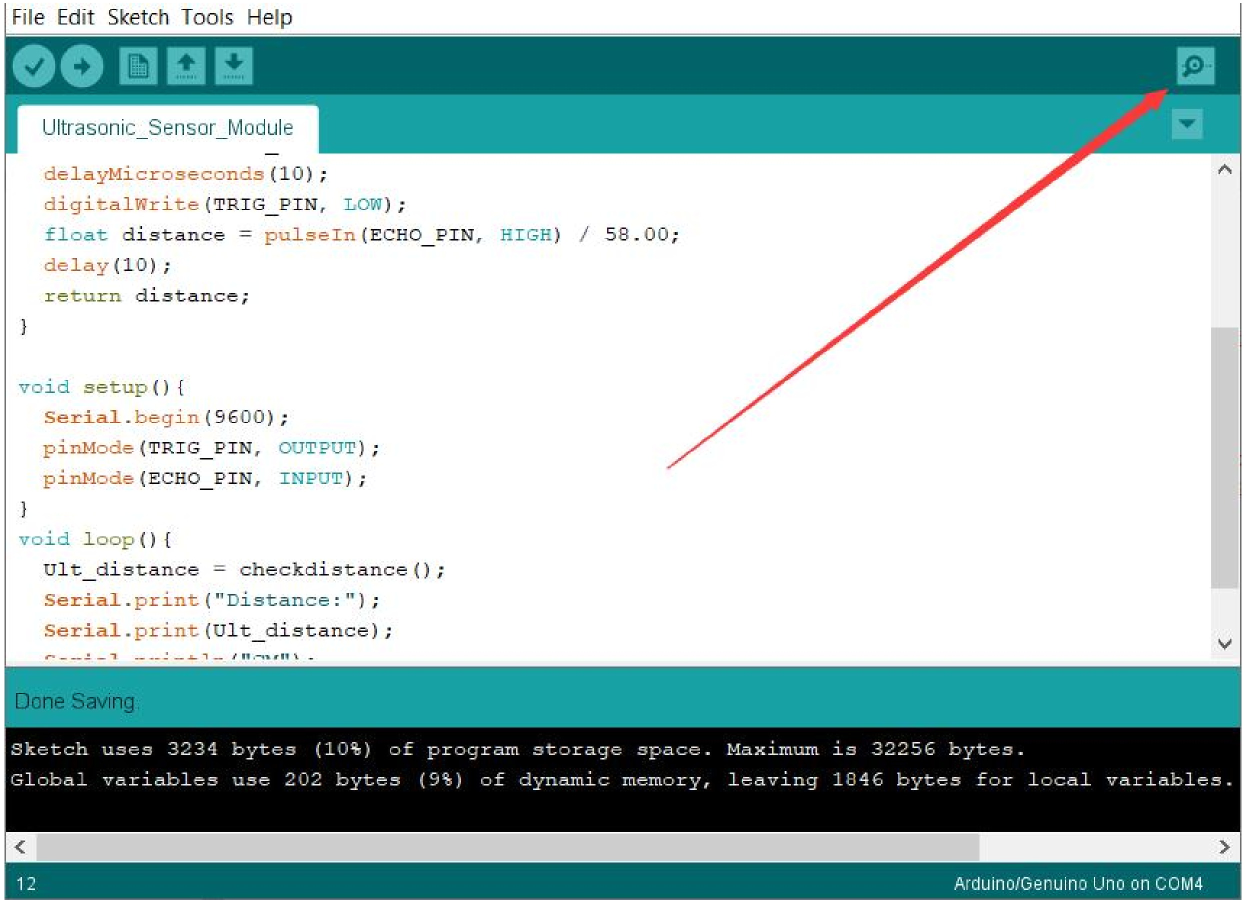


Code

Open this source program in Code>Lesson 6 Ultrasonic Sensor Module>Ultrasonic Sensor Module.ino.Upload the code to Arduino UNO.

See Lesson 4 Blink Test(Test your first program) for details about program uploading if there are any errors.

After uploading the code. Click the Serial Monitor button to turn on the serial monitor.

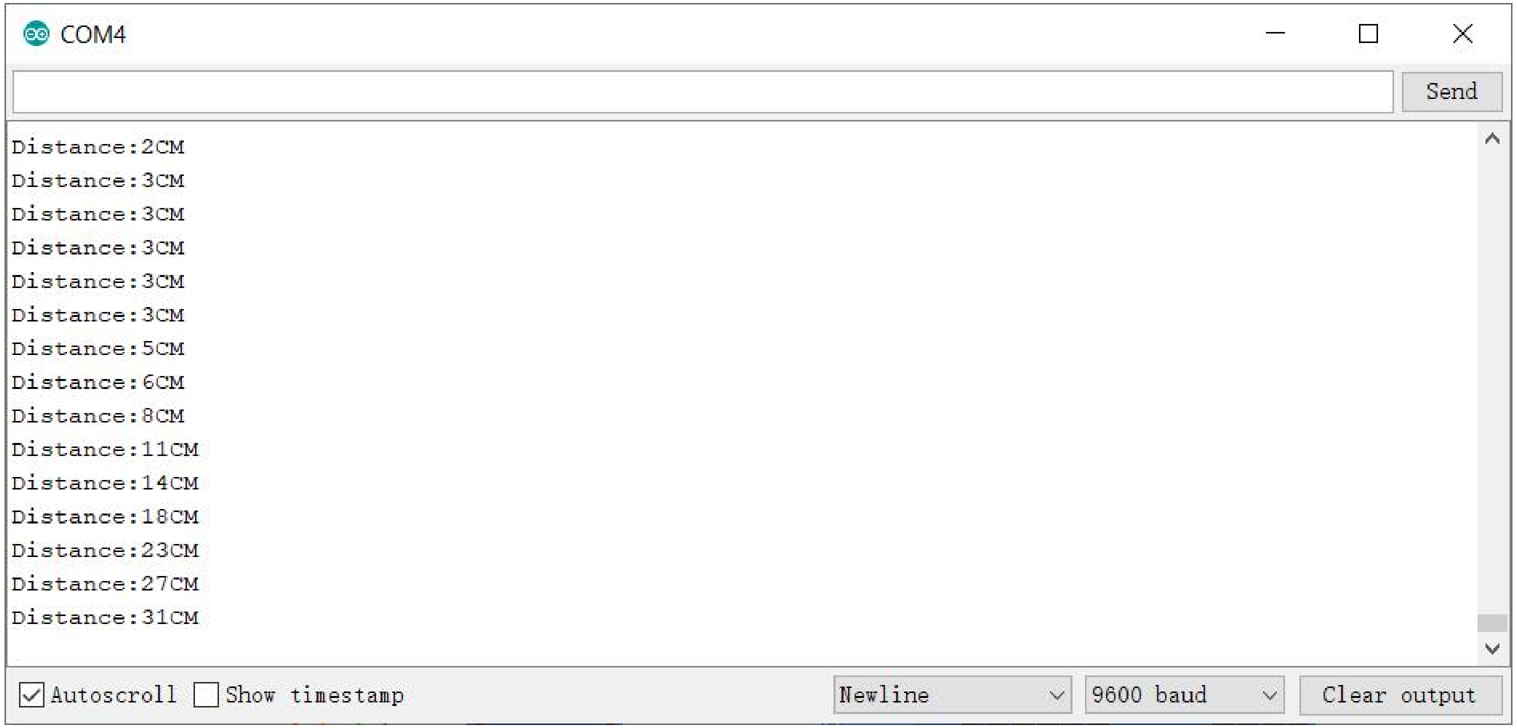


```
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  float distance = pulseIn(ECHO_PIN, HIGH) / 58.00;
  delay(10);
  return distance;
}

void setup(){
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}
void loop(){
  Ult_distance = checkdistance();
  Serial.print("Distance:");
  Serial.print(Ult_distance);
  Serial.println("CM");
```

Done Saving.

```
Sketch uses 3234 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 202 bytes (9%) of dynamic memory, leaving 1846 bytes for local variables.
```

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



```
Distance:2CM
Distance:3CM
Distance:3CM
Distance:3CM
Distance:3CM
Distance:3CM
Distance:5CM
Distance:6CM
Distance:8CM
Distance:11CM
Distance:14CM
Distance:18CM
Distance:23CM
Distance:27CM
Distance:31CM
```

☑ Autoscroll □ Show timestamp Newline 9600 baud Clear output

Lesson 7 IR Receiver Module

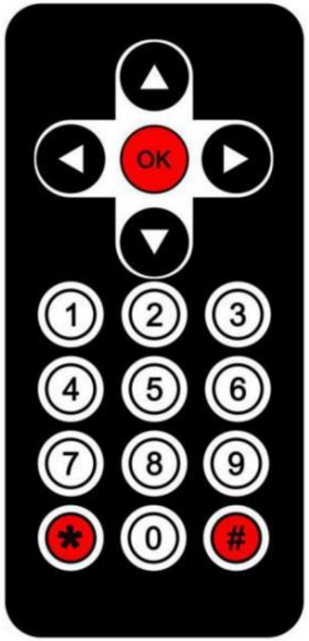
About this lesson:

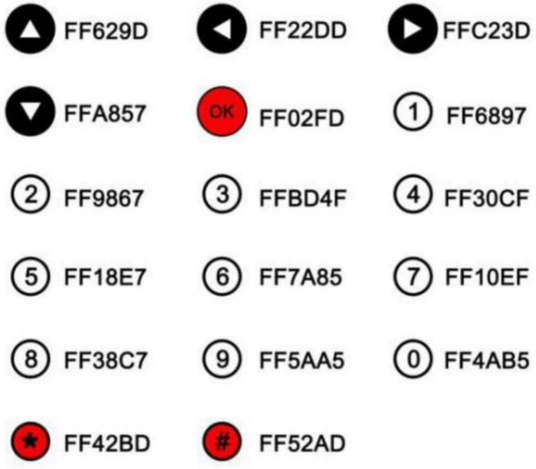
Using an IR Remote is a great way to have wireless control of your project. Infrared remotes are simple and easy to use. In this tutorial we will be connecting the IR receiver to the UNO, and then use a Library that was designed for this particular sensor.

What is an infrared remote control

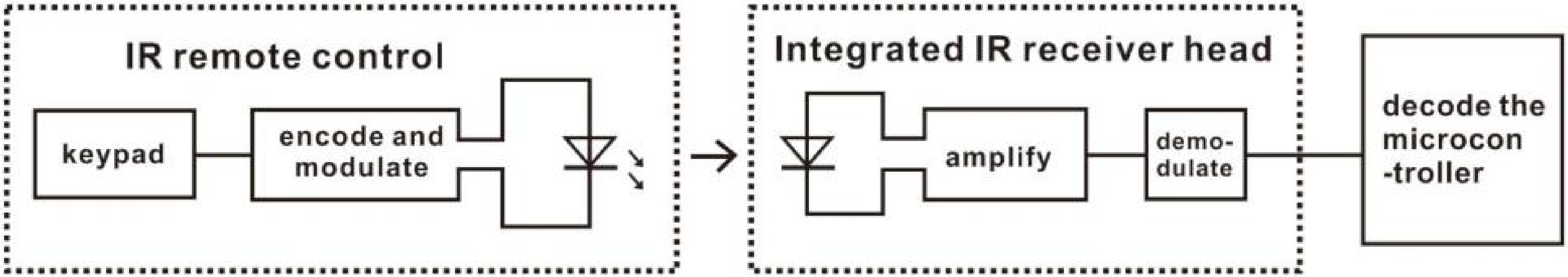
There is no doubt that infrared remote control is commonly seen in our daily life. It's hard to imagine our world without it.An infrared remote control can be used to control a wide range of home appliances such as television, audio, video recorders and satellite signal receivers.Well, in the following let's get a better understanding of the infrared remote control.Infrared remote control is composed of infrared transmitting and infrared receiving systems.That is, consist of an infrared remote control, an infrared receiver module and a microcontroller that can decode.You can refer to the figure below.

Remote control key value





▲	FF629D	◀	FF22DD	▶	FFC23D
▼	FFA857	OK	FF02FD	1	FF6897
2	FF9867	3	FFBD4F	4	FF30CF
5	FF18E7	6	FF7A85	7	FF10EF
8	FF38C7	9	FF5AA5	0	FF4AB5
*	FF42BD	#	FF52AD		

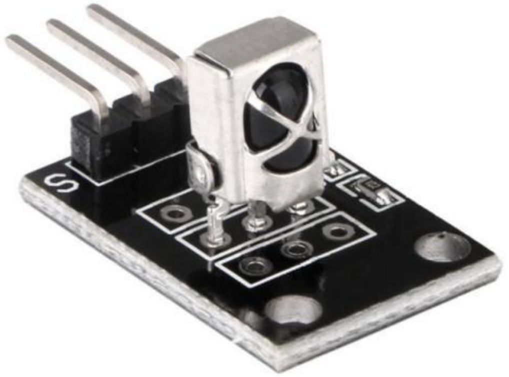


The 38K infrared carrier signal transmitted by an infrared remote controller is encoded by an encoding chip inside the remote controller. It is composed of a pilot code, user code, data code, and data inversion code.The time interval between pulses is used to distinguish whether it is a signal 0 or 1. (when the ratio of high level to low level is about 1:1, considered as signal 0.)

And the encoding is just well composed of signal 0 and 1.The user code of the same button on remote controller is unchanged. Using difference data distinguish the key pressed on the remote control. When press down a button on the remote control, it will send out an infrared carrier signal. And when infrared receiver receives that signal, its program will decode the carrier signal, and through different data codes, thus can judge which key is pressed.The microcontroller is decoded by an received signal 0 or 1 to determine which key is pressed by the remote control.

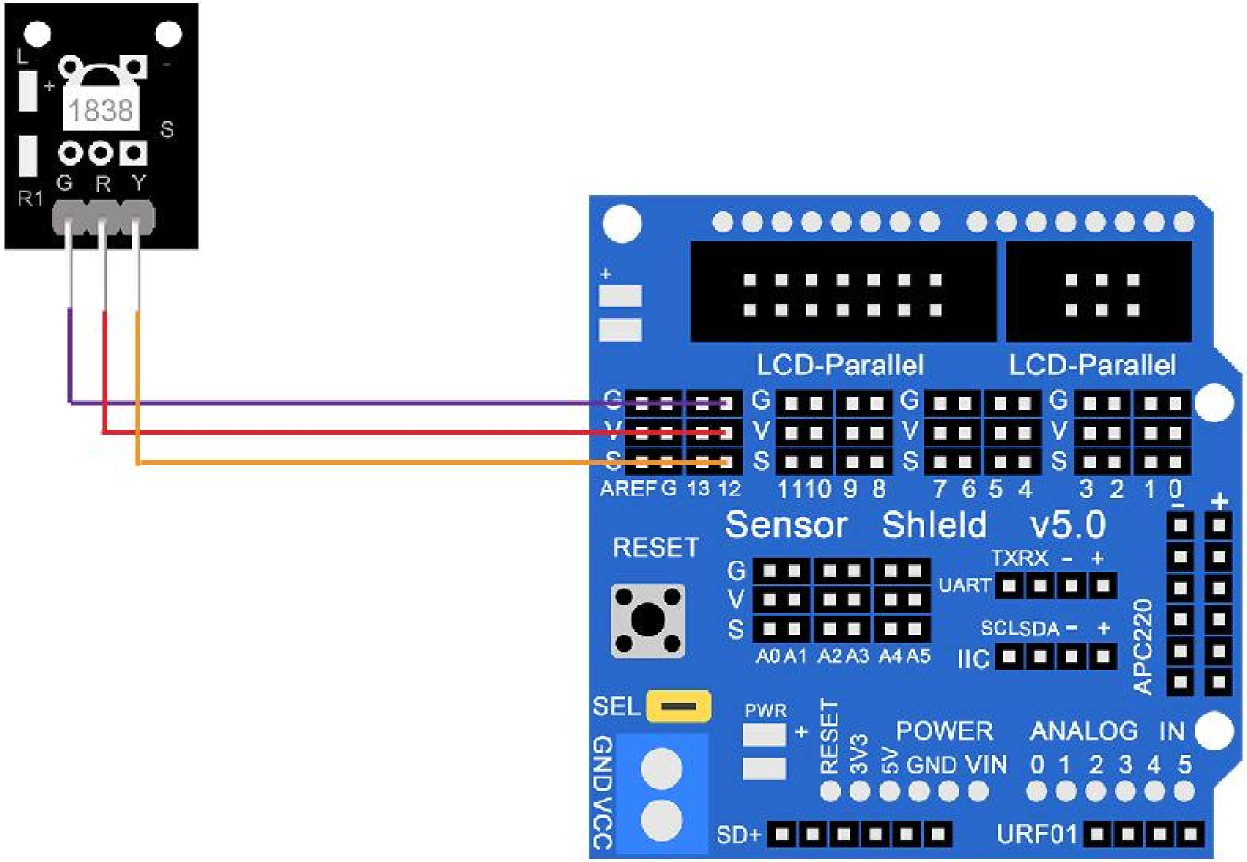
What is an infrared receiver

The robot shield comes with infrared receiver module. It is mainly composed of an infrared receiving head. This device integrates with reception, amplification and demodulation.Its inter IC has been demodulated, outputting Digital signal. Suitable for IR remote control and infrared data transmission.



IR receiver Pin:
The “-” is the Ground,middle pin is Voltage 5V, and “S” is signal.
The “G” is the Ground, “R” is Voltage 5V, and “Y”pin is signa.

Wiring diagram



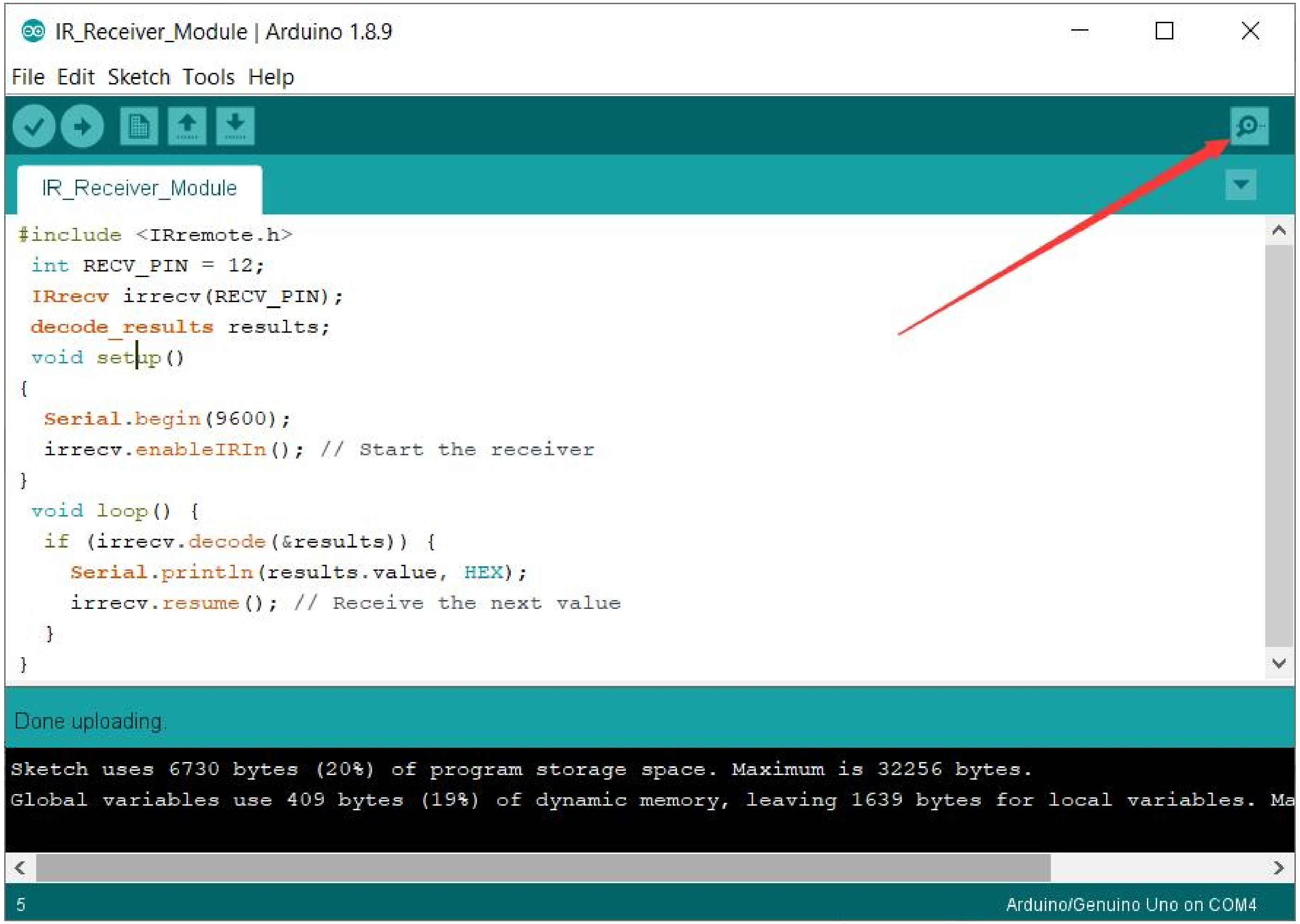
Code

Open this source program in Code>Lesson 7 IR_Receiver_Module>IR_Receiver_Module.ino.Upload the code to Arduino UNO.

See Lesson 4 Blink Test(Test your first program) 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 3 How to Add Libraries .

After uploading the code. Click the Serial Monitor button to turn on the serial monitor.



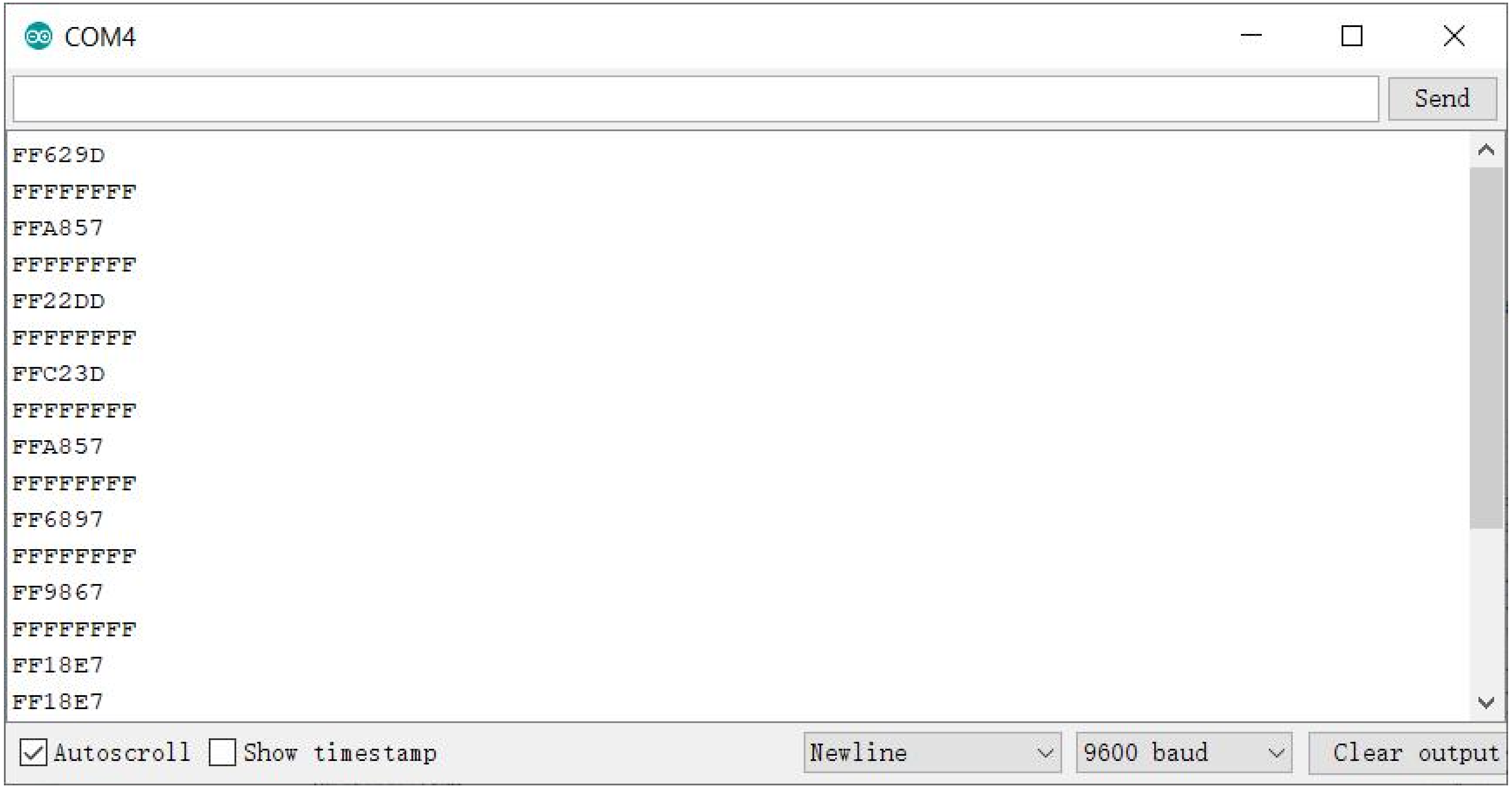
IR_Receiver_Module | Arduino 1.8.9
File Edit Sketch Tools Help
IR_Receiver_Module

```
#include <IRremote.h>
 int RECV_PIN = 12;
 IRrecv irrecv(RECV_PIN);
 decode_results results;
 void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
 void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Done uploading.

```
Sketch uses 6730 bytes (20%) of program storage space. Maximum is 32256 bytes.
Global variables use 409 bytes (19%) of dynamic memory, leaving 1639 bytes for local variables. Ma
```

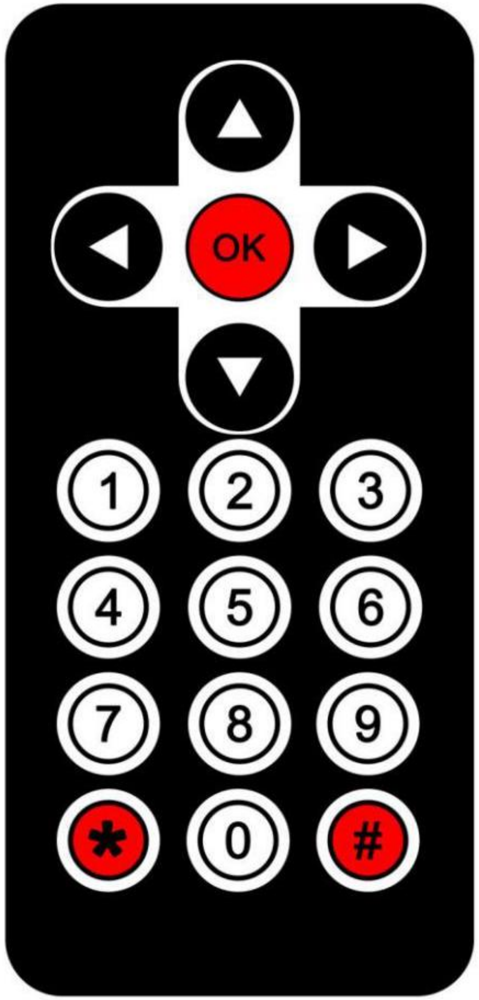
After connection and uploading codes, aim at IR receiving module and press the key, finally you can see corresponding codes. Open the monitor then you can see the data as blow:



```
FF629D
FFFFFFFF
FFA857
FFFFFFFF
FF22DD
FFFFFFFF
FFC23D
FFFFFFFF
FFA857
FFFFFFFF
FF6897
FFFFFFFF
FF9867
FFFFFFFF
FF18E7
FF18E7
```

☑ Autoscroll □ Show timestamp Newline 9600 baud

Remote control code:



▲	FF629D	◀	FF22DD	▶	FFC23D
▼	FFA857	OK	FF02FD	1	FF6897
2	FF9867	3	FFBD4F	4	FF30CF
5	FF18E7	6	FF7A85	7	FF10EF
8	FF38C7	9	FF5AA5	0	FF4AB5
*	FF42BD	#	FF52AD		

Lesson 8 L298N Motor Driver

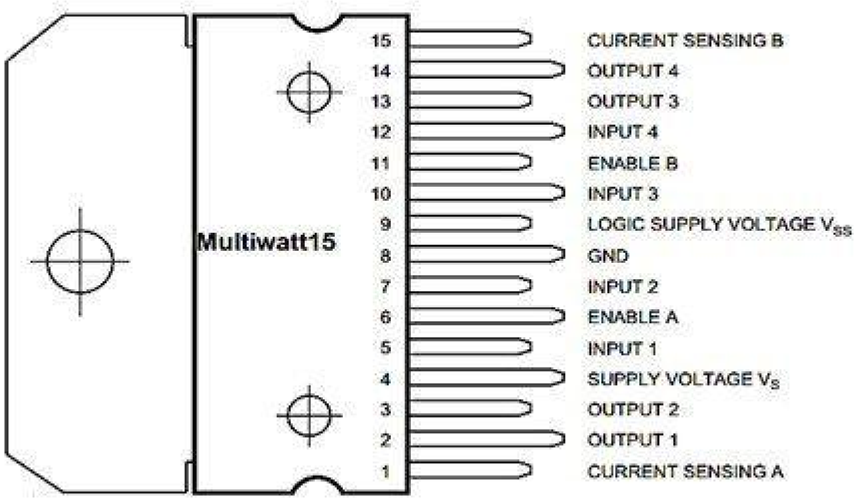
About this lesson:

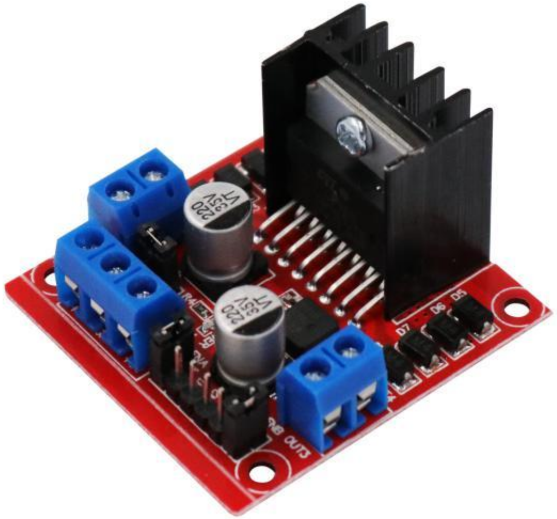
In this lesson, you will learn how to use a L298N Motor Driver module.

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:



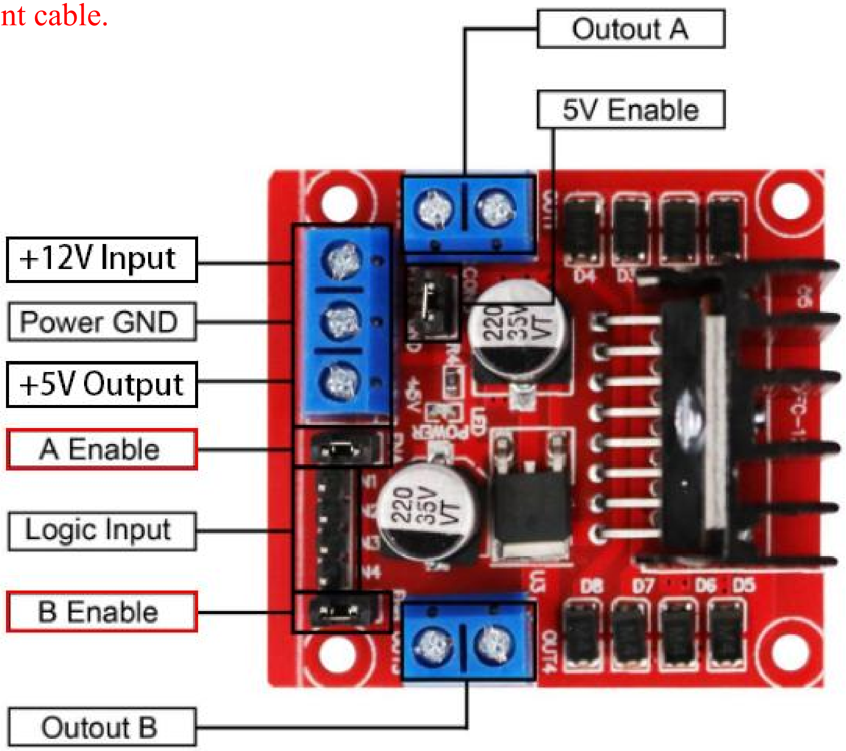


Using L298N made by ST Company as the control chip, the module has characteristics of strong driving ability, low calorific value and strong anti-interference ability.

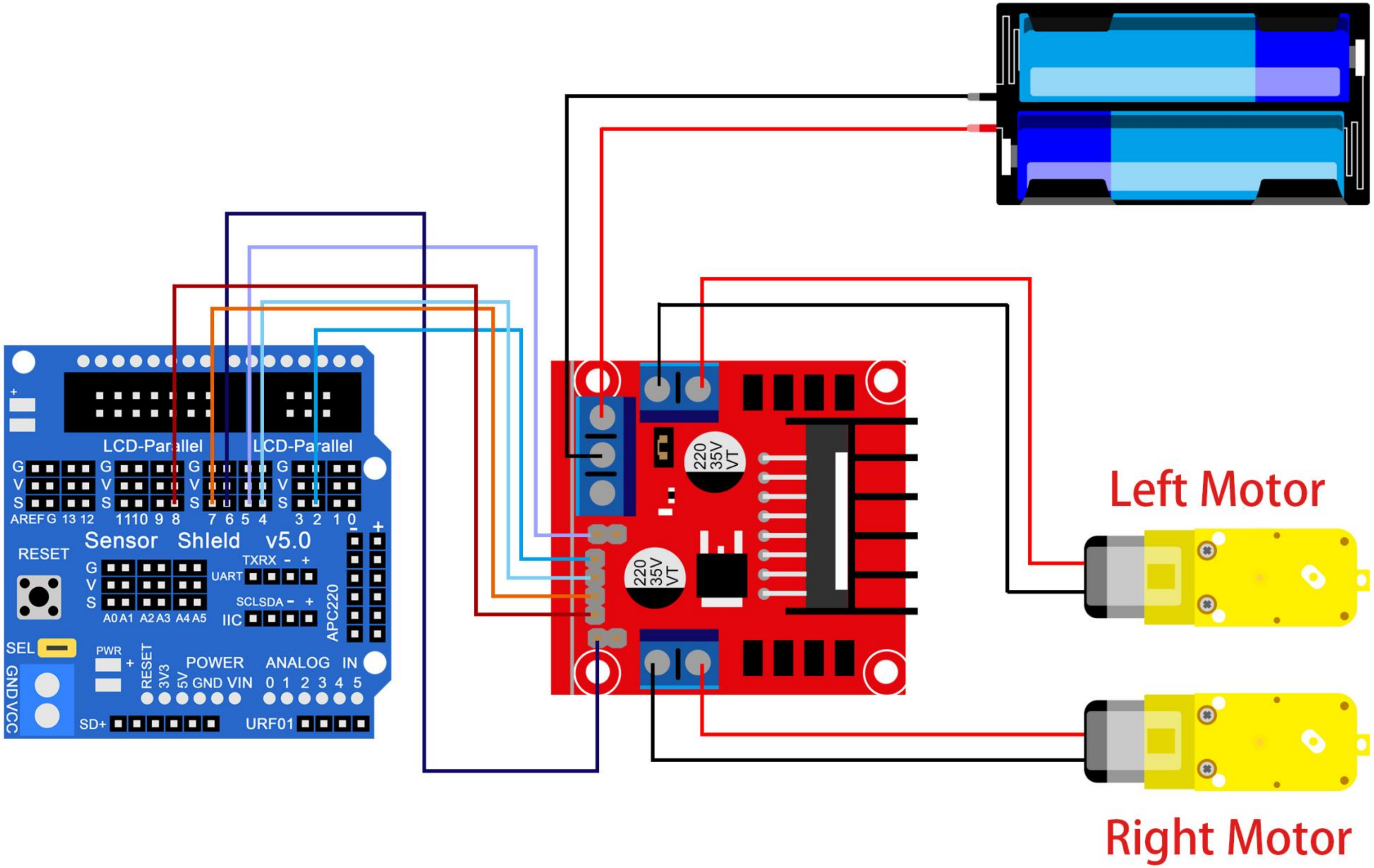
This module can use built-in 78M05 for electric work via a driving power supply part. But to avoid the damage of the voltage stabilizing chip, please use an external 5V logic supply when using more than 12V driving voltage.

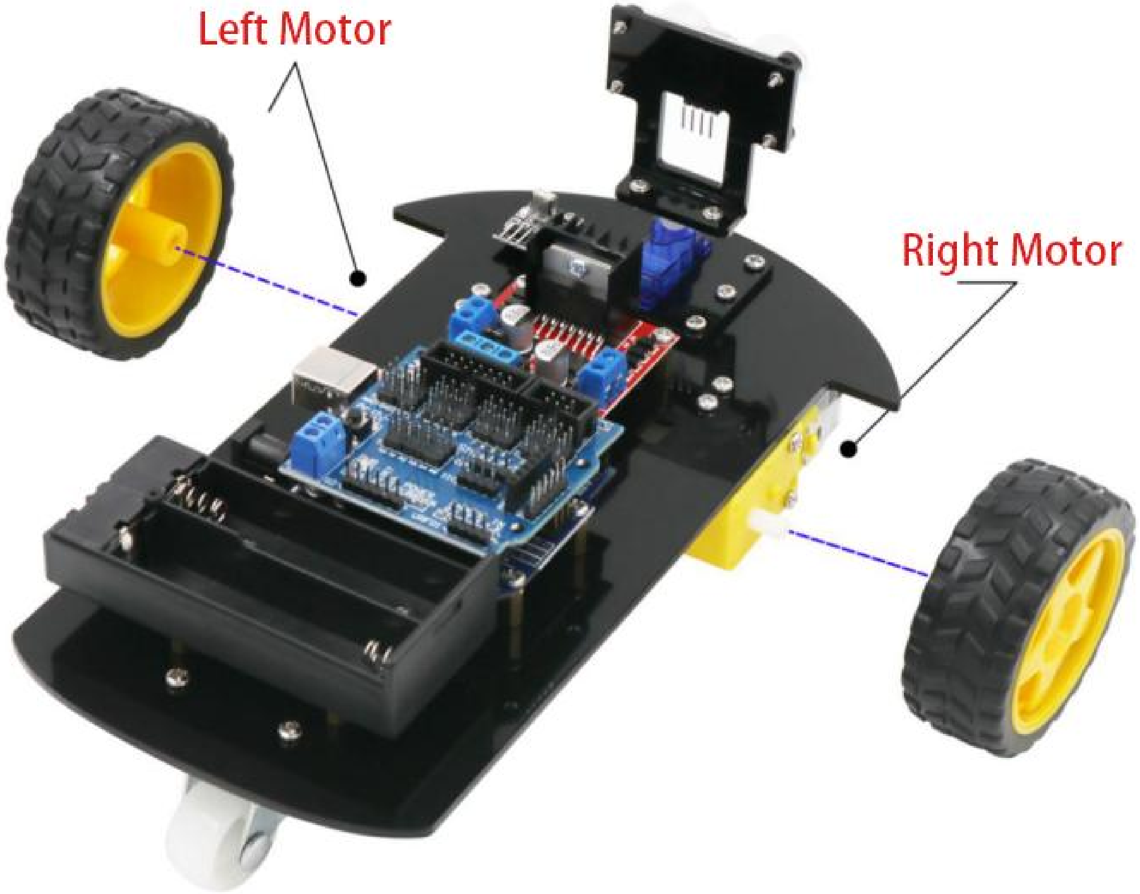
Using large capacity filter capacitor, this module can follow current to protect diodes, and improve reliability.

In order to be able to adjust the speed of motor A and motor B, unplug the short-circuit jumper caps of A Enable and B Enable. Connect to the IO interface of arduino uno with a Dupont cable.



Wiring diagram




Left Motor
Right Motor

Code

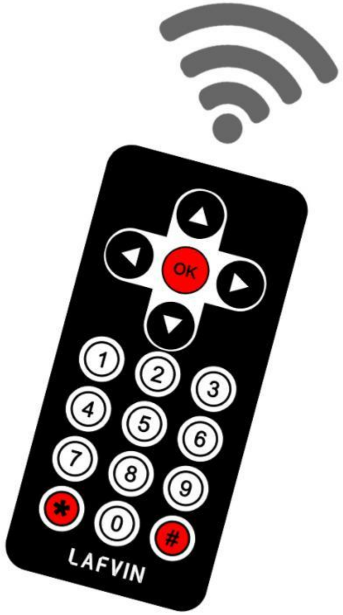
Open this source program in Code>Lesson 8 L298N Motor Driver>L298N_Motor_Driver.ino.Upload the code to Arduino UNO.After uploading the code, the robot car will walk forward for 2 seconds, go backward for 2 seconds, turn left for 2 seconds, turn right for 2 seconds, and then stop for 2 seconds. Repeat this process in a loop.

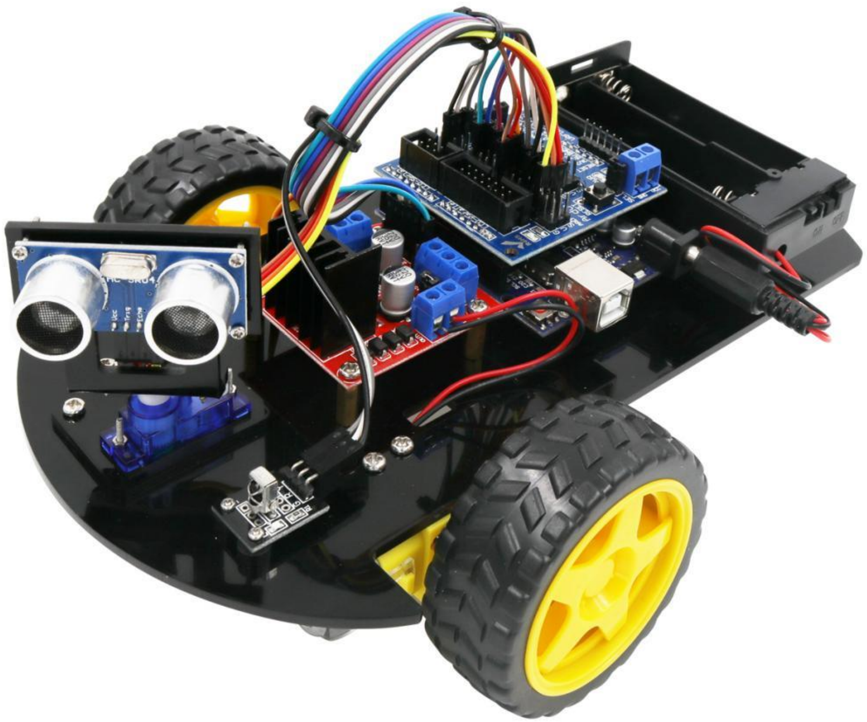
See Lesson 4 Blink Test(Test your first program) for details about program uploading if there are any errors.

Lesson 9 IR Remote Control Car

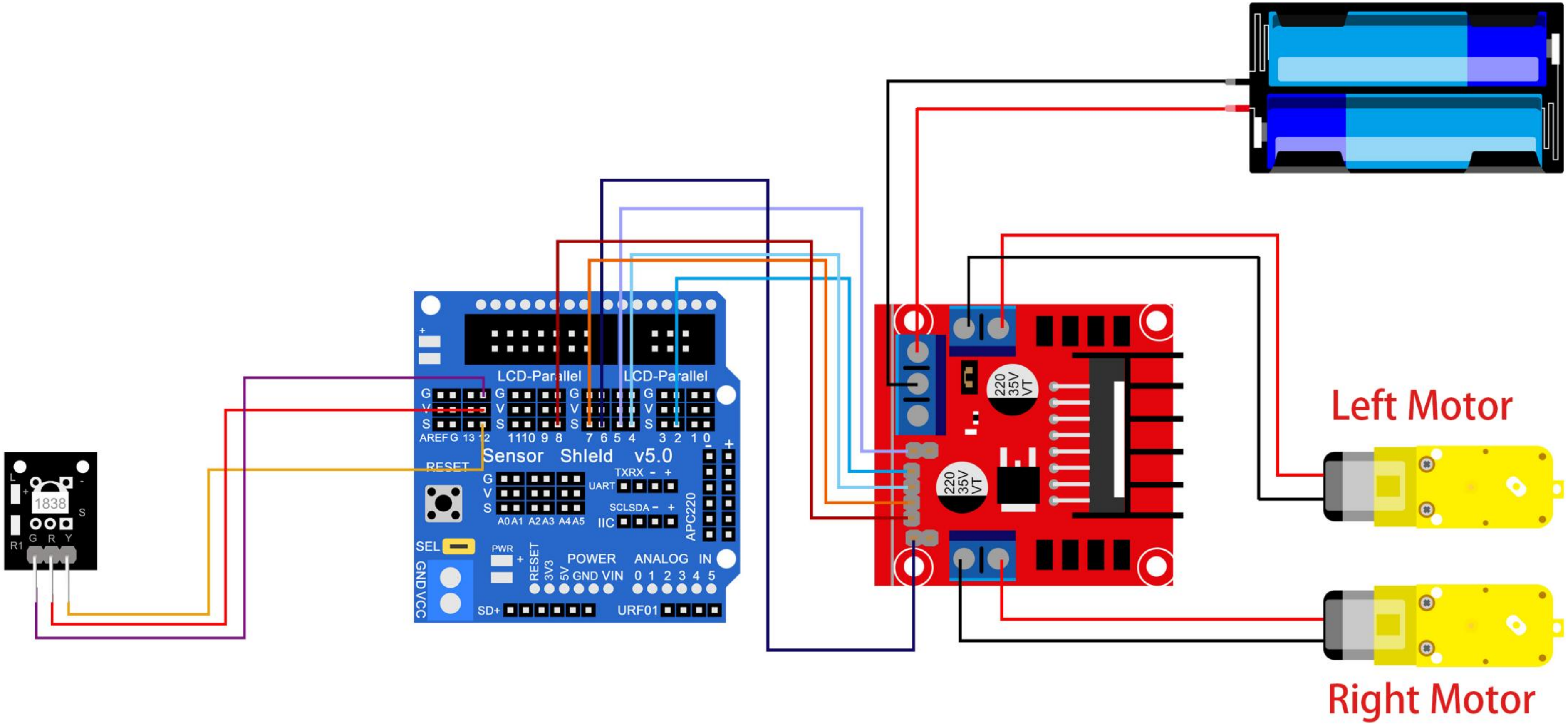
About this lesson:

This lesson ,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.





Wiring diagram



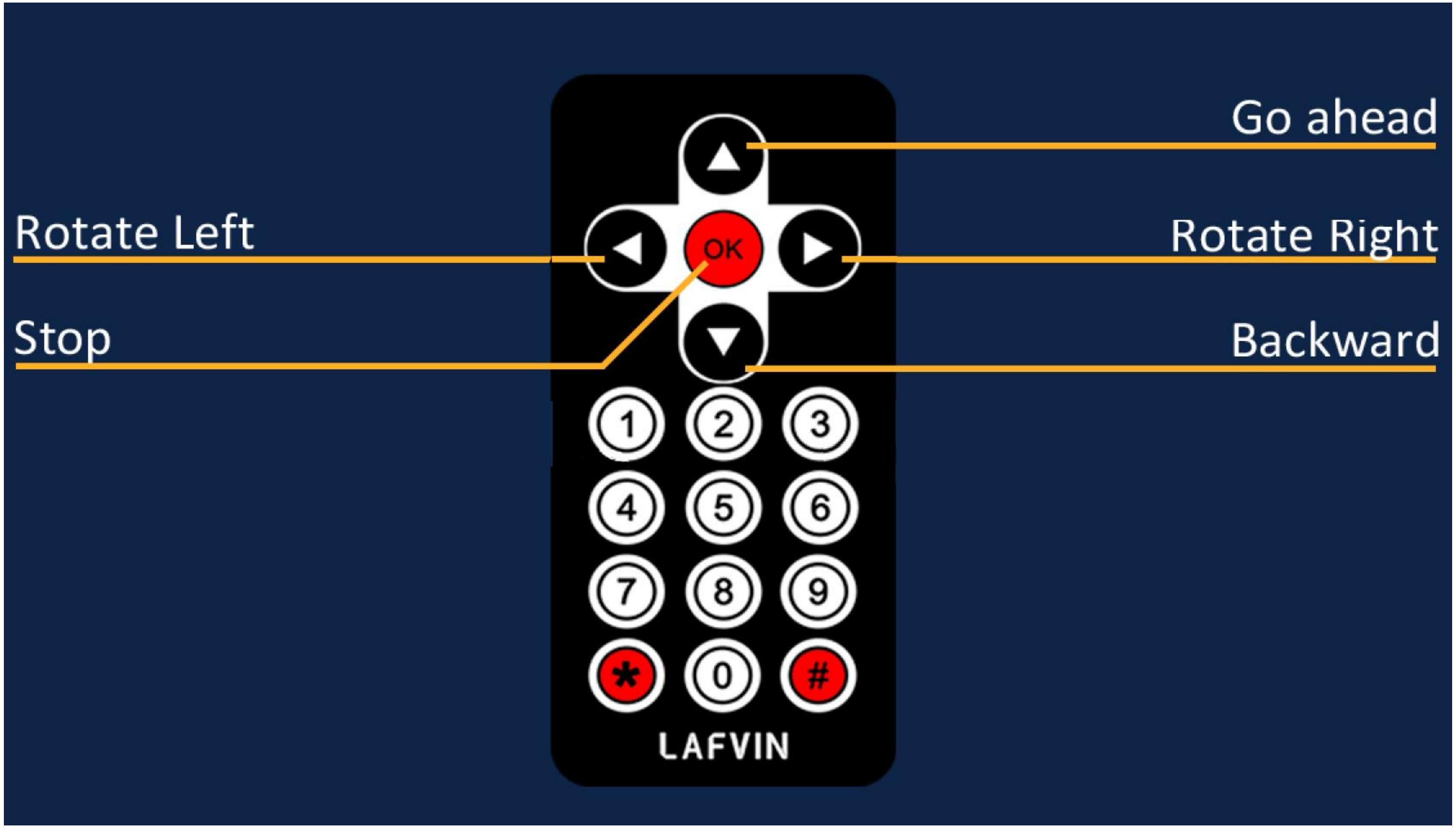
Code

Open this source program in Code>Lesson 9 IR Remote Control Car>IR_Remote_Control_Car.ino.Upload the code to Arduino UNO.

See Lesson 4 Blink Test(Test your first program) 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 3 How to Add Libraries .

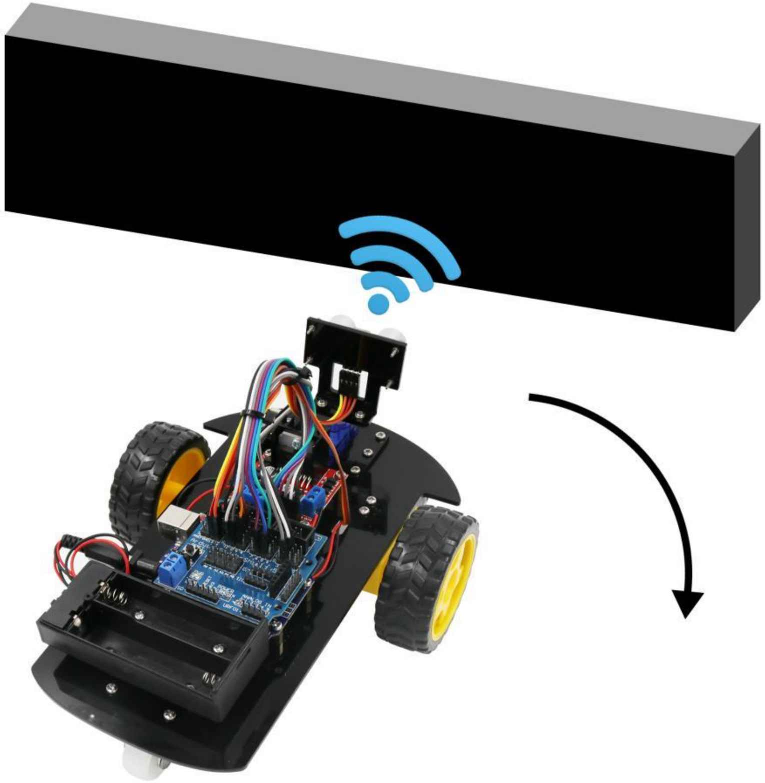
Use the remote control to control the direction of the robot car



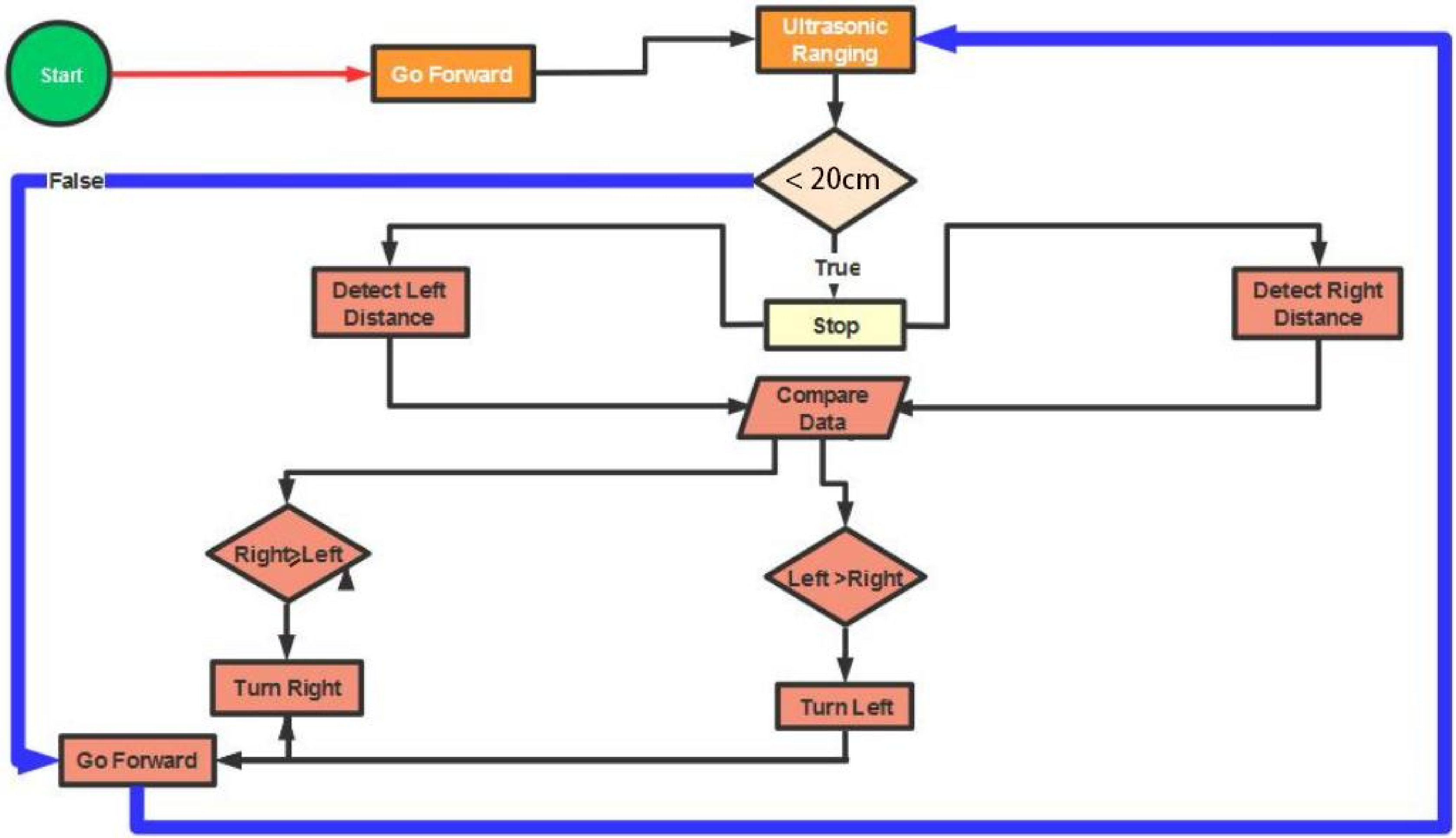
Lesson 10 Obstacle Avoidance Robot Car

About this lesson:

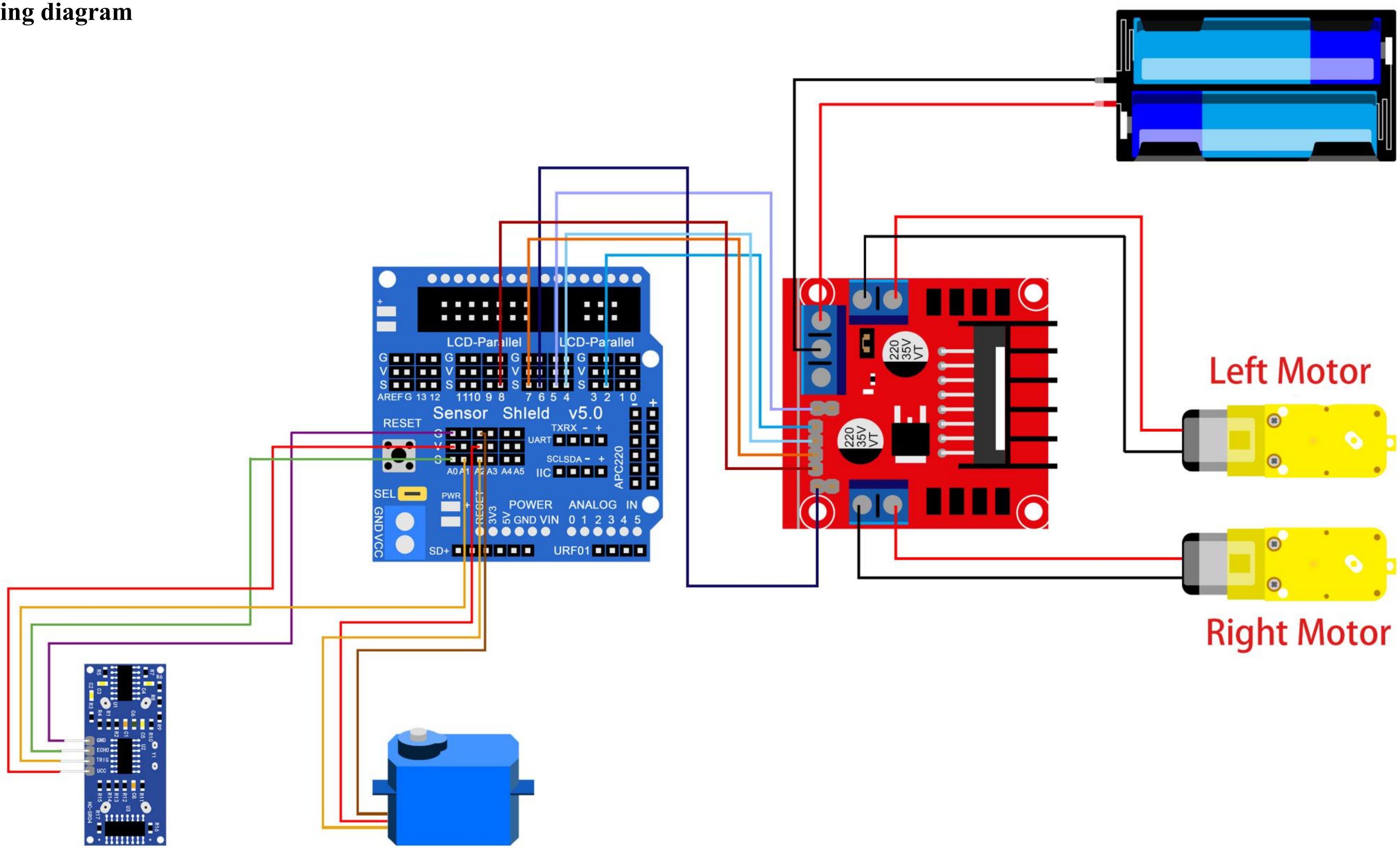
This lesson ,regarding Arduino as main control, detect front obstacle by ultrasonic sensor, and send the distance data to Arduino. Arduino will analyses the feedback signal and then control the driver motor to adjust the car diversion. Finally the car is able to avoid obstacle automatically and keep going.



Flow Chart


Start
Go Forward
Ultrasonic Ranging
< 20cm
False
True
Stop
Detect Left Distance
Detect Right Distance
Compare Data
Right>Left
Left >Right
Turn Right
Turn Left
Go Forward

Wiring diagram



Code

Open this source program in Code>Lesson 10 Obstacle_Avoidance_Robot_Car>Obstacle_Avoidance_Robot_Car.ino.

Upload the code to Arduino UNO.

The robot car judges whether the obstacle is on the left or the right through the ultrasonic module, and then turns to avoid the obstacle. Tip: The height of the obstacle should be greater than 15cm.

See Lesson 4 Blink Test(Test your first program) 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 3 How to Add Libraries .