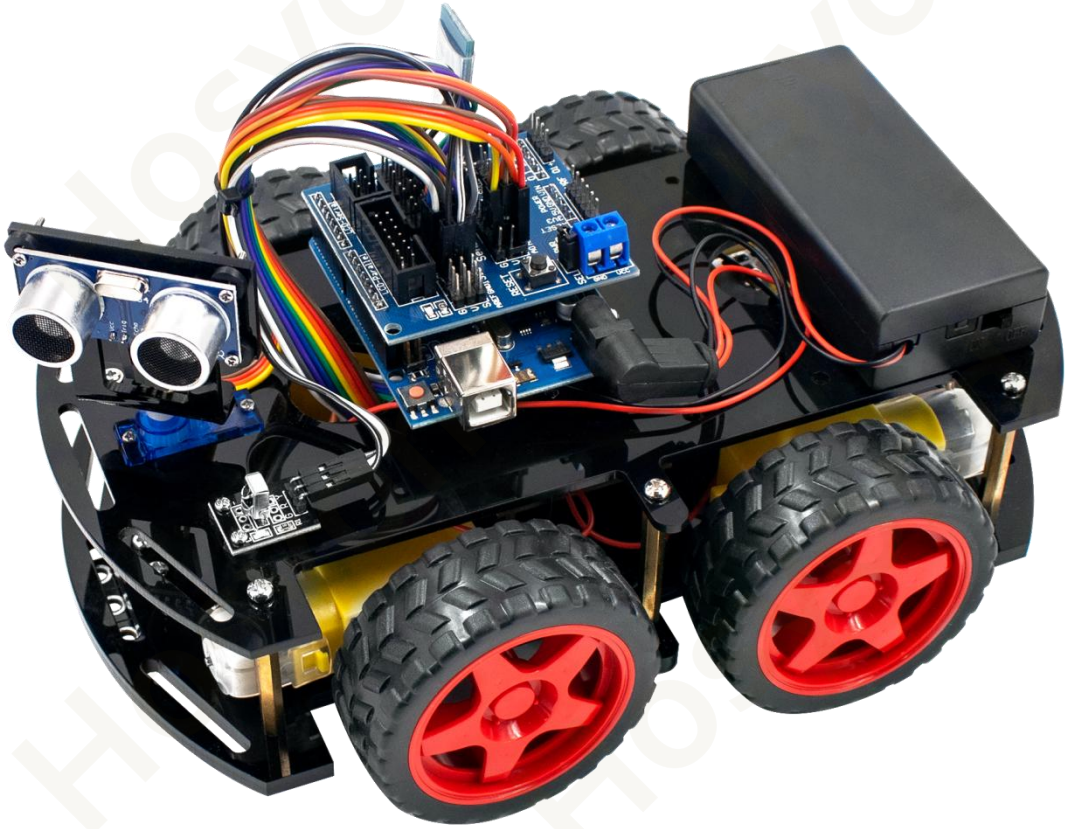
4WD Smart Robot Car Kit



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.Please 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

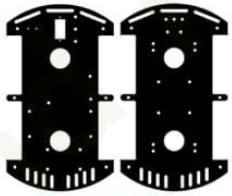
This tutorial include codes,libraries and lessons. It is designed for beginners. It will teach every users how to assembly the smart robot car kit and use Arduino controller board, sensors and modules.

Packing list

Hosyond

Smart Car Kit

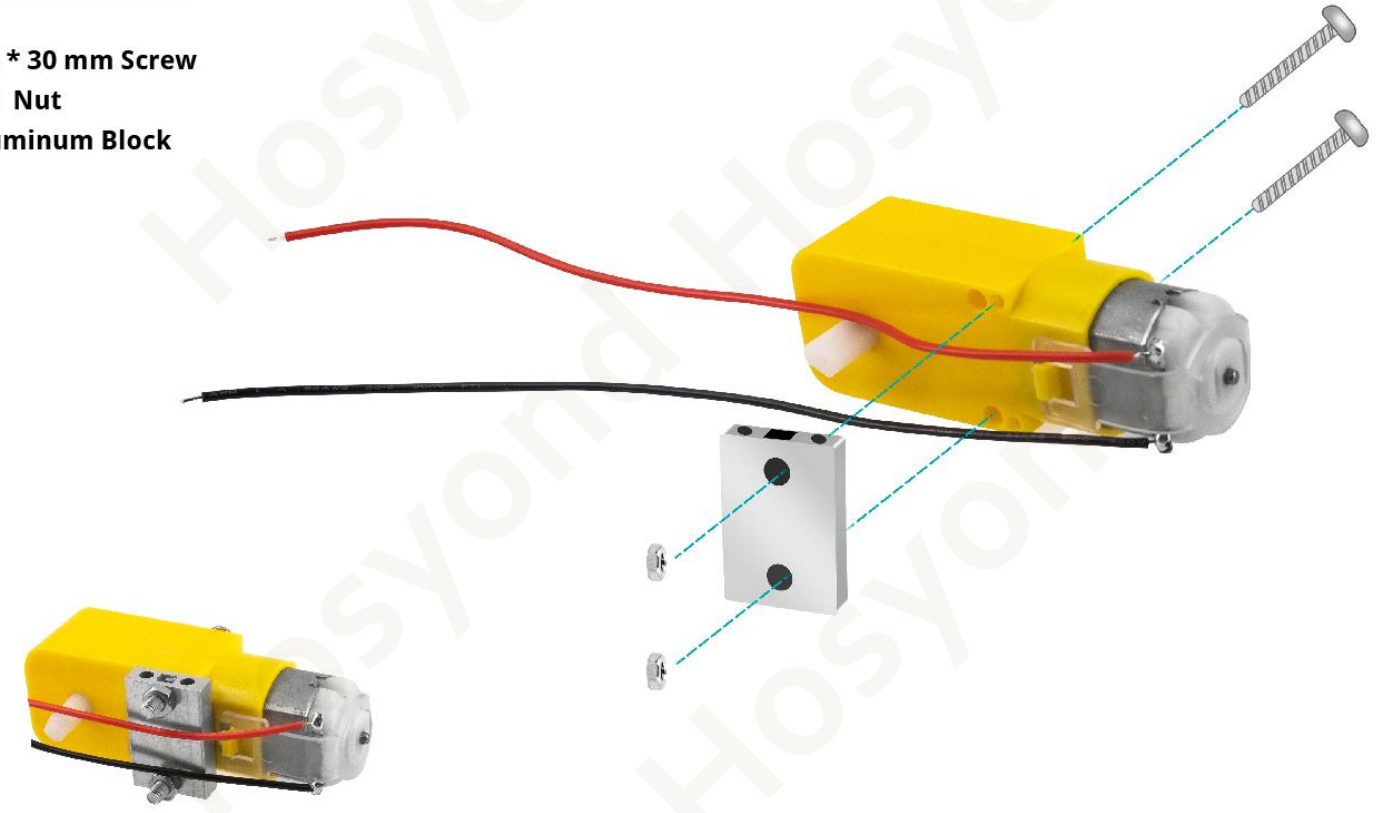
hosyond_service@163.com

 1pcs R3 board	 1pcs V5 Expanding Board	 1pcs L298N Motor Driver Board	 1pcs Ultrasound Module	 1pcs SG90 Servo
 3pcs Line Tracking Module	 1pcs Bluetooth Module	 2pcs Acrylic Chassic	 1 pcs Cell box	 1pcs IR Receiver
 1pcs Black Tape	 1pcs Remote	 1pcs Ultrasound Module Bracket	 4pcs Tire	 4pcs DC Motor
 40pins F-F DuPont Cable	 1pcs USB Cable	 1pcs Screwdriver & 4pcs Bunding Belt	 Screw & Nut & Copper Cylinder	

Lesson 1 Installation Method

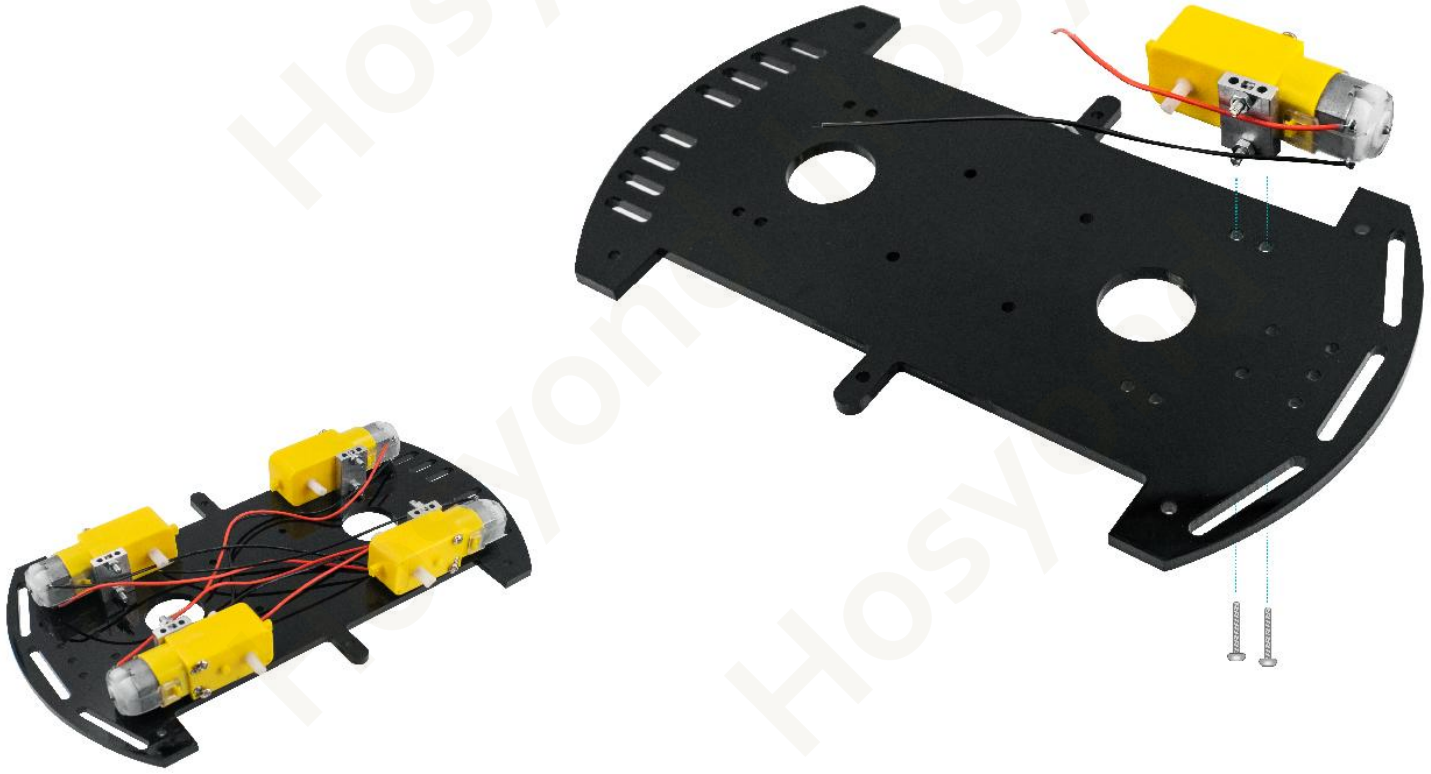
Step 01

1. M3 * 30 mm Screw
2. M3 Nut
3. Aluminum Block



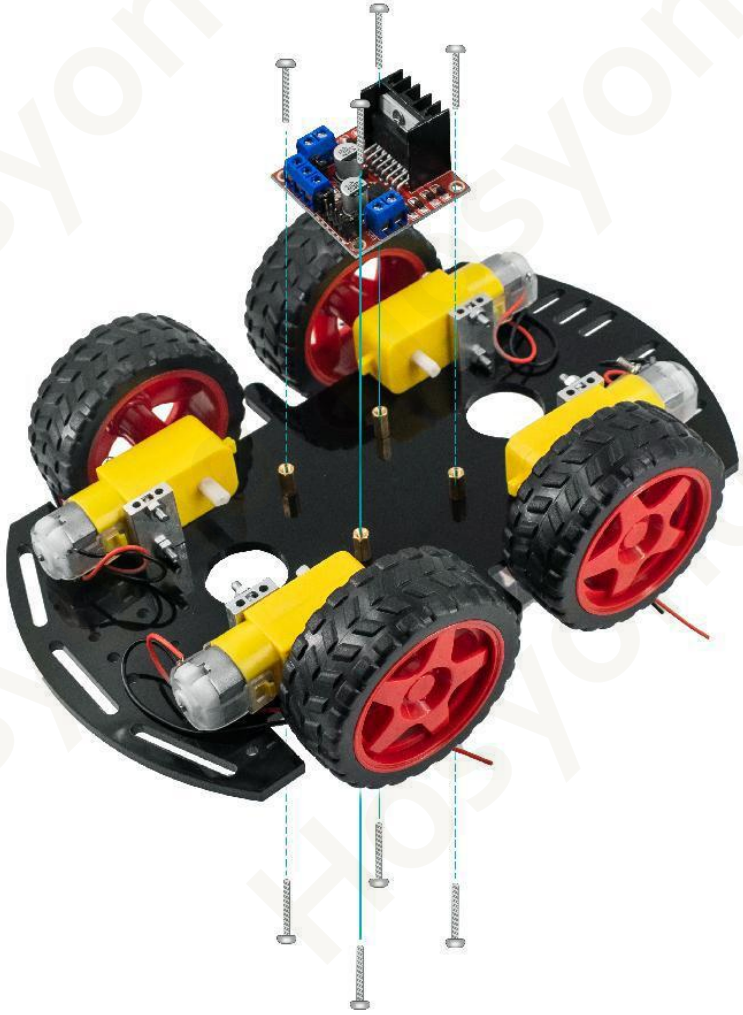
Step 02

1. M3 * 8 mm Screw



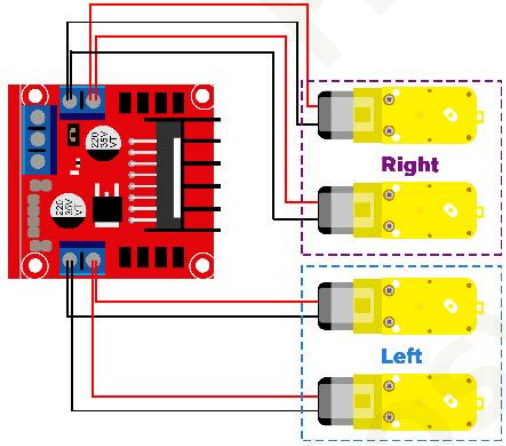
Step 03

1. M3 * 10 mm Copper Cylinder
2. M3 * 8mm Screw
3. Tire

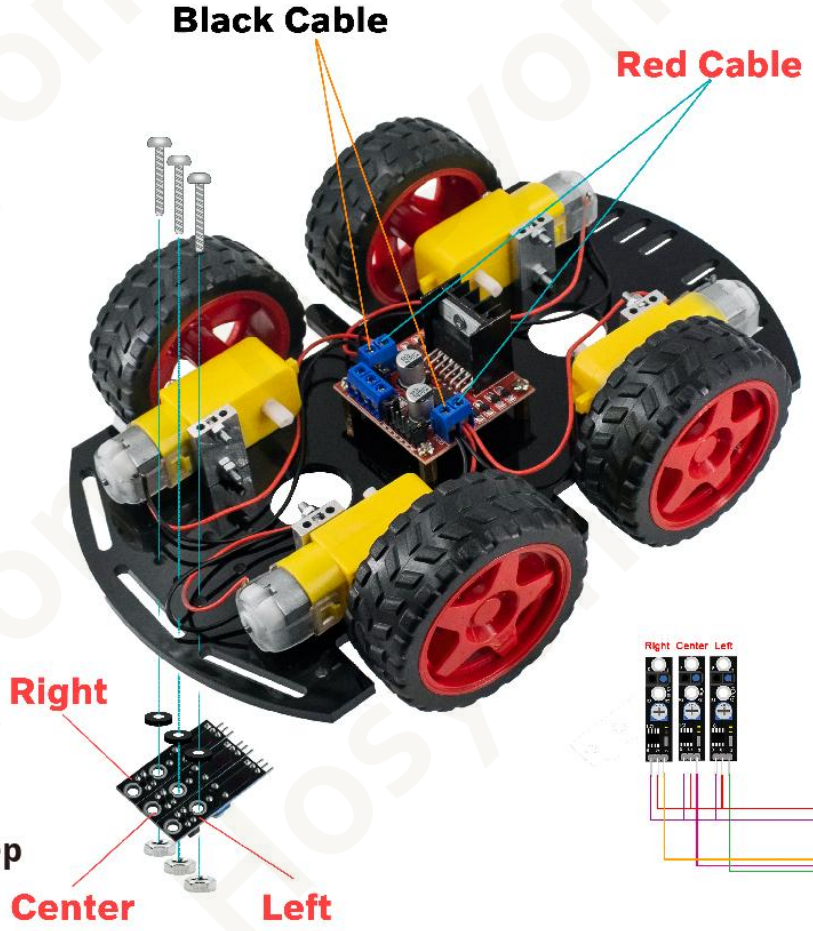


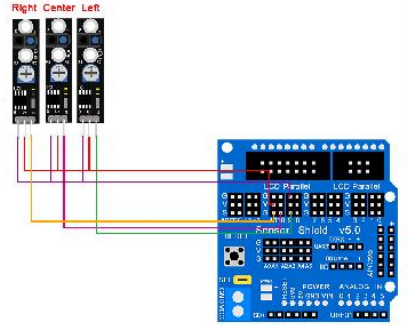
Step 04

1. M3* 12 mm Screw
2. M3 Nut



Connect Motor to L298N on this step



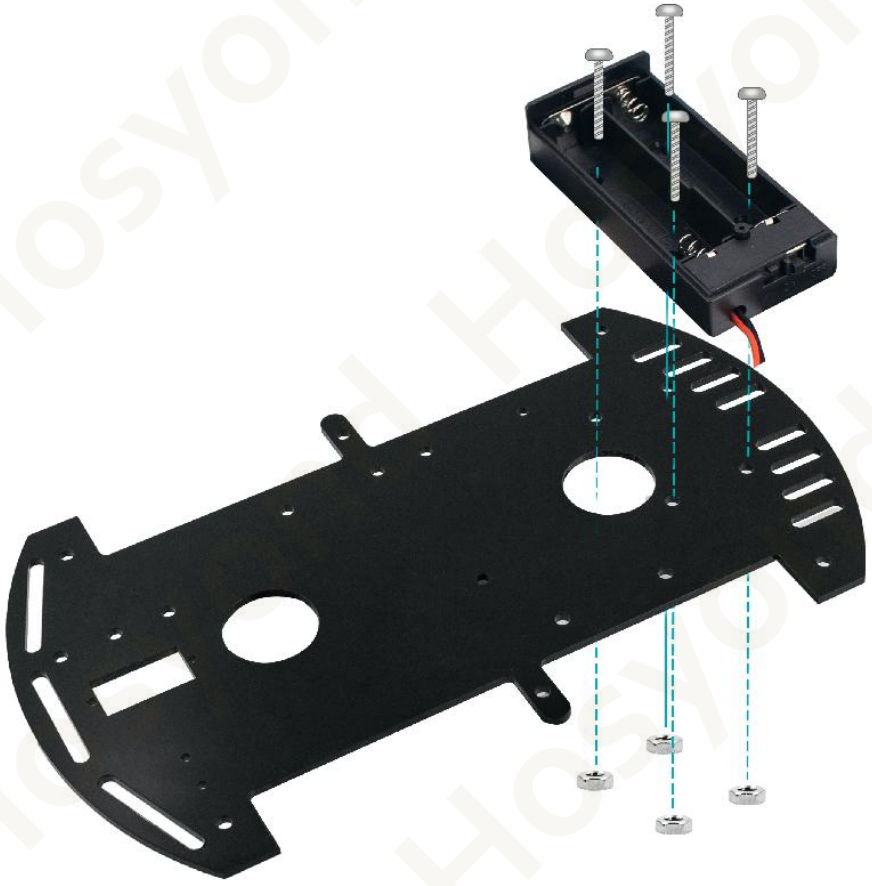


Step 05

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

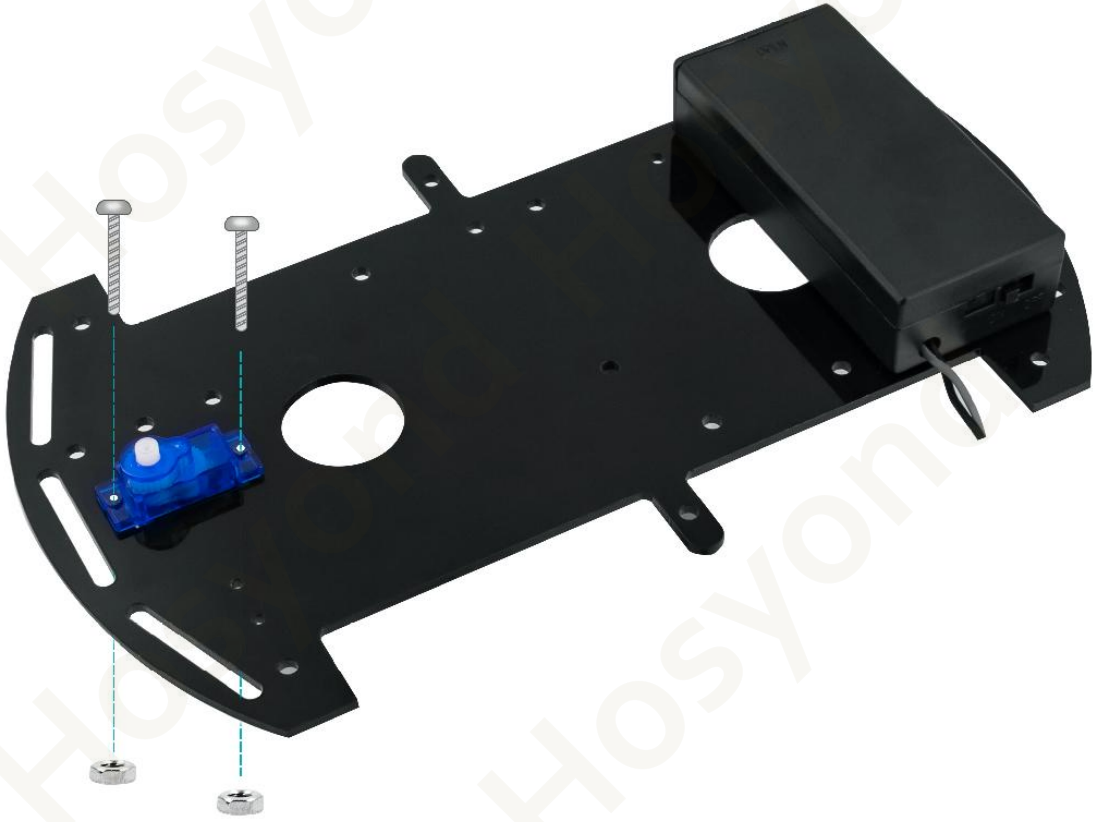
Step 06

1. M3 * 8 mm Screw
2. M3 Nut



Step 07

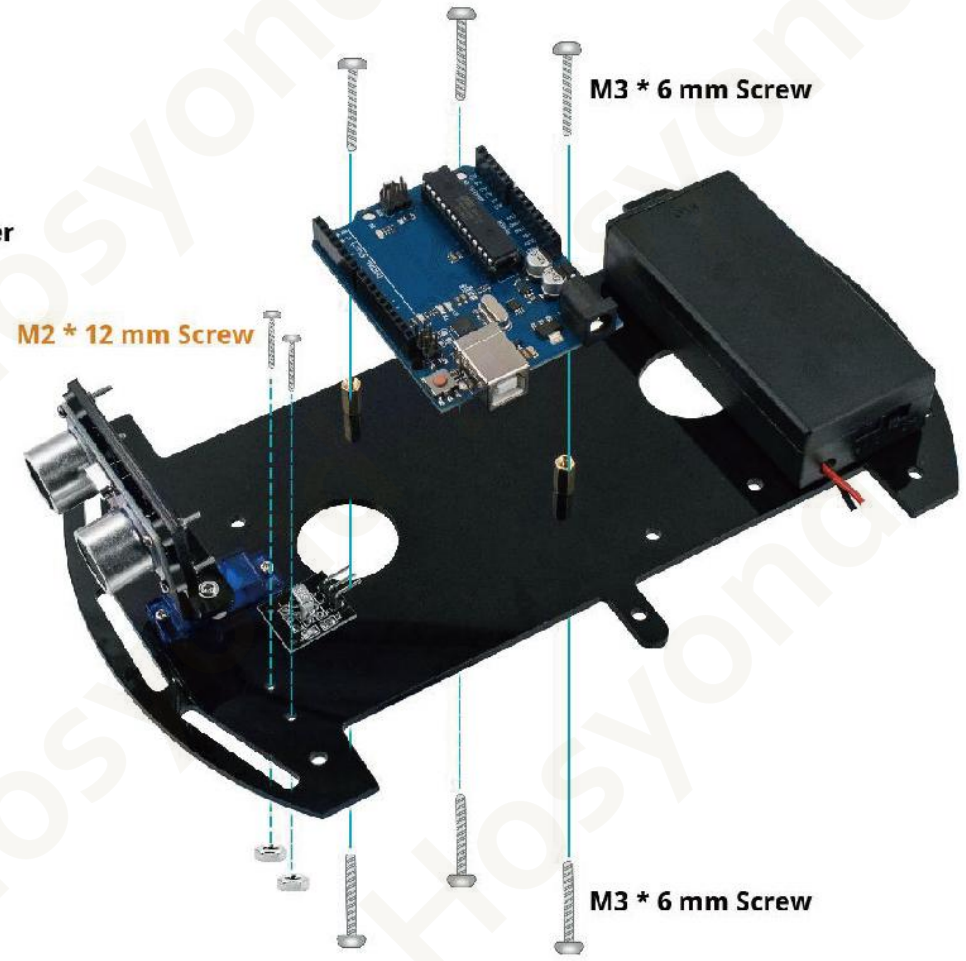
1. M2 * 12 mm
2. M2 Nut



Step 08

Step 09

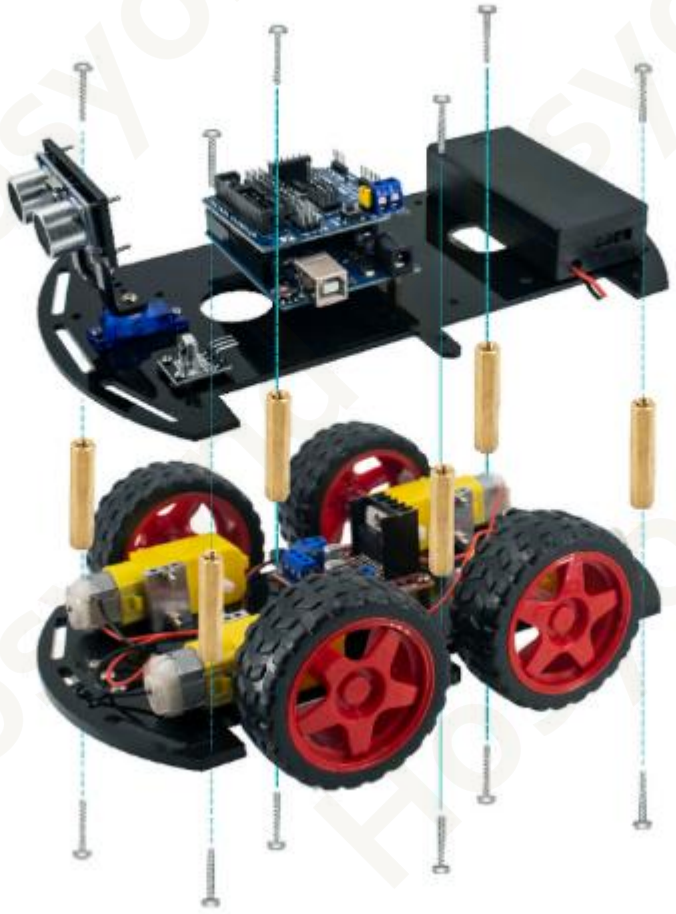
1. M3 * 6mm Screw
2. M2 * 12mm Screw
3. M2 Nut
4. M3 * 10mm Copper Cylinder

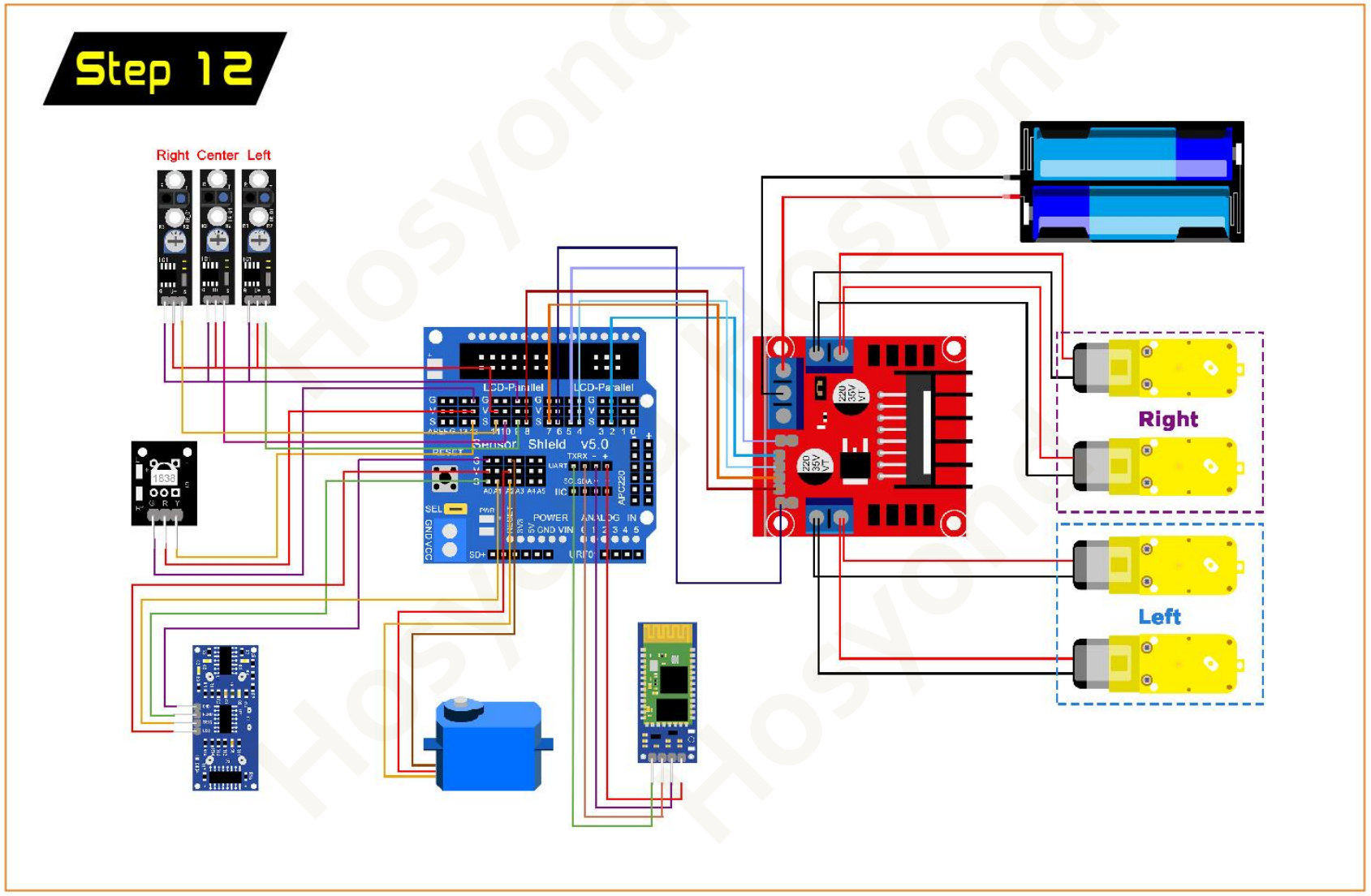


Step 10

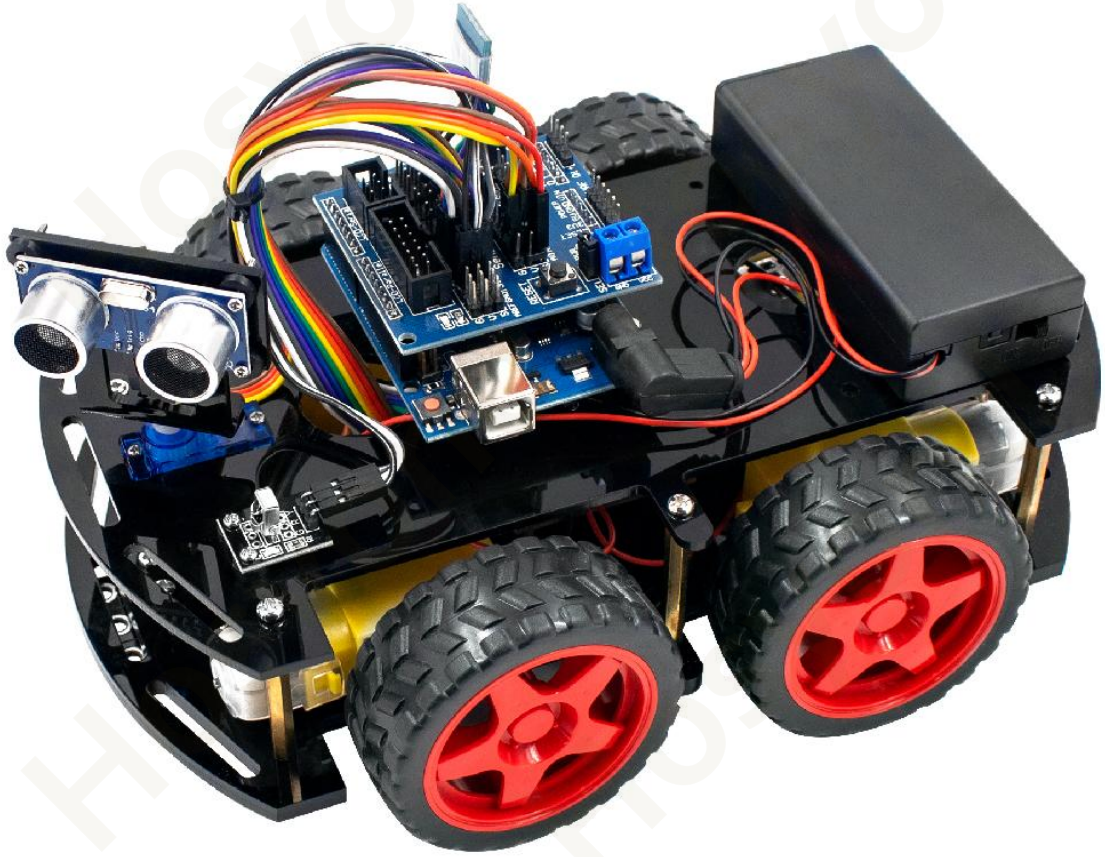
Step 11

1. M3 *10 mm Screw
2. M3 * 40 mm Copper Cylinder




Step 12
Right Center Left
Right
Left

Final



Lesson 2 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.

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.
Take Windows as an example here.

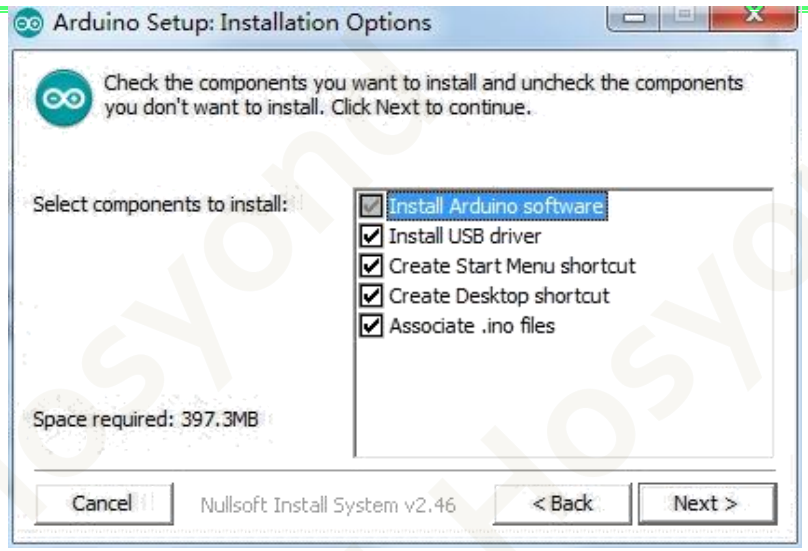
Click Windows Installer.

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.

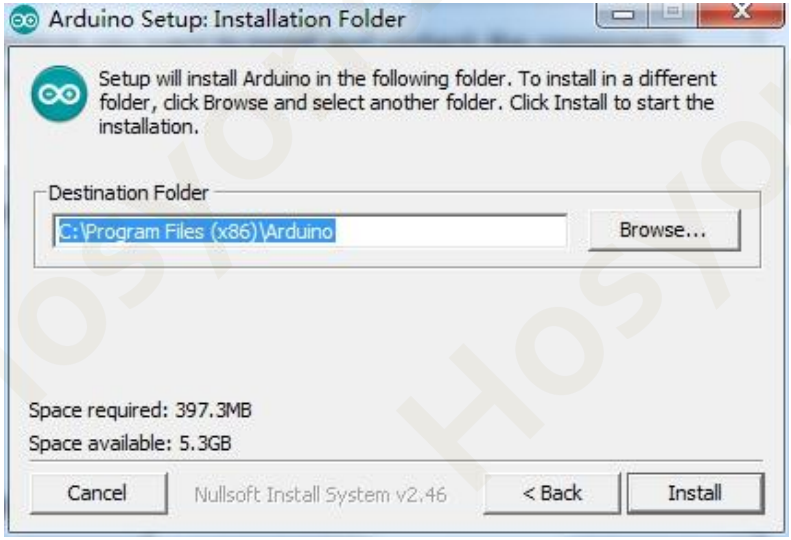
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

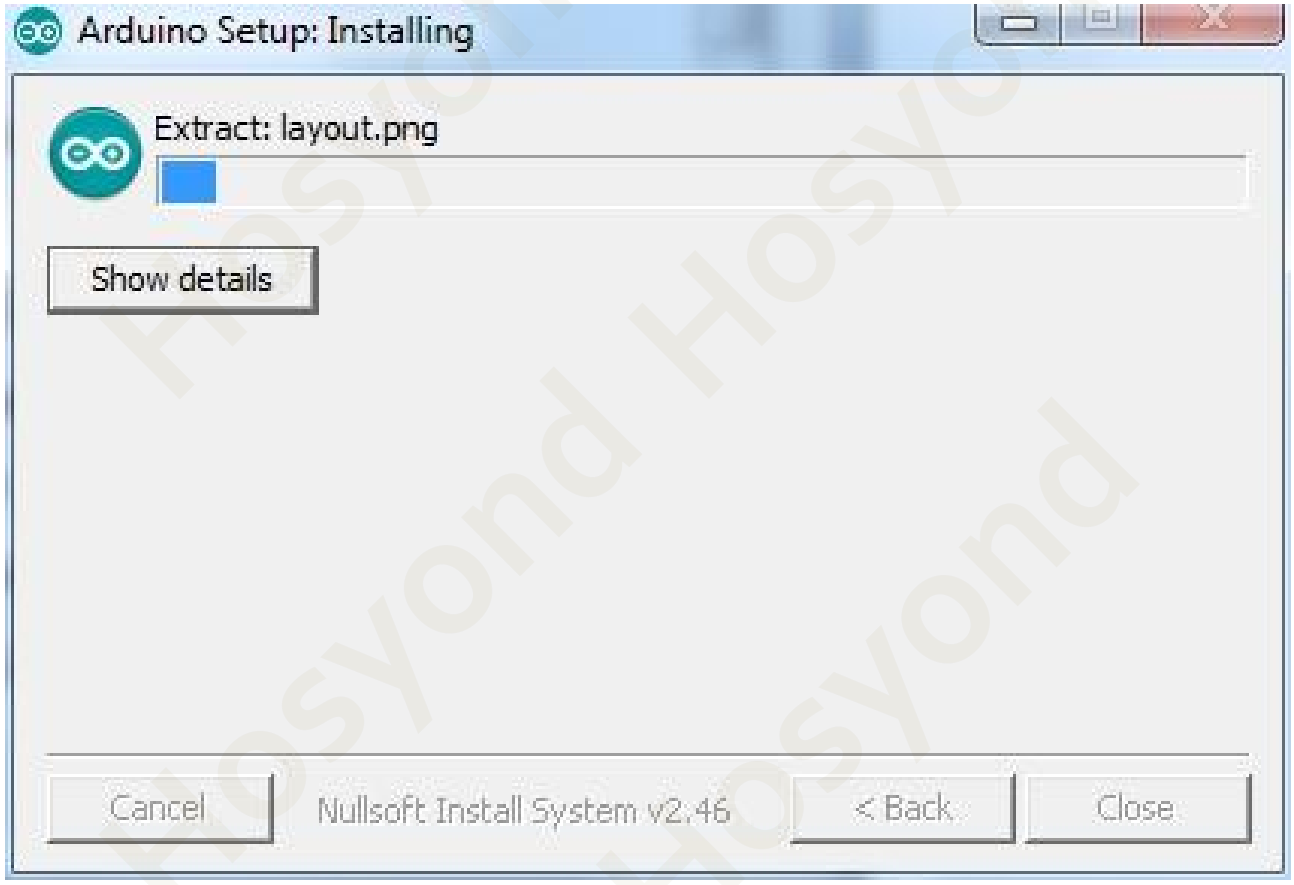
Click I Agree to see the following interface


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: 397.3MB
Cancel
Nullsoft Install System v2.46
< Back
Next >

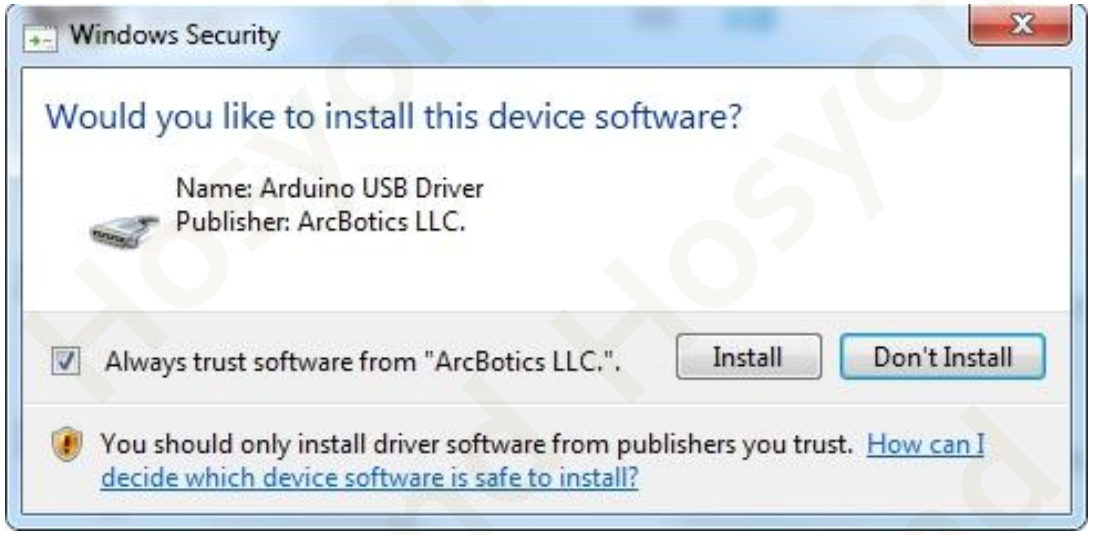
Click Next


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
Browse...
Space required: 397.3MB
Space available: 5.3GB
Cancel
Nullsoft Install System v2.46
< Back
Install

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

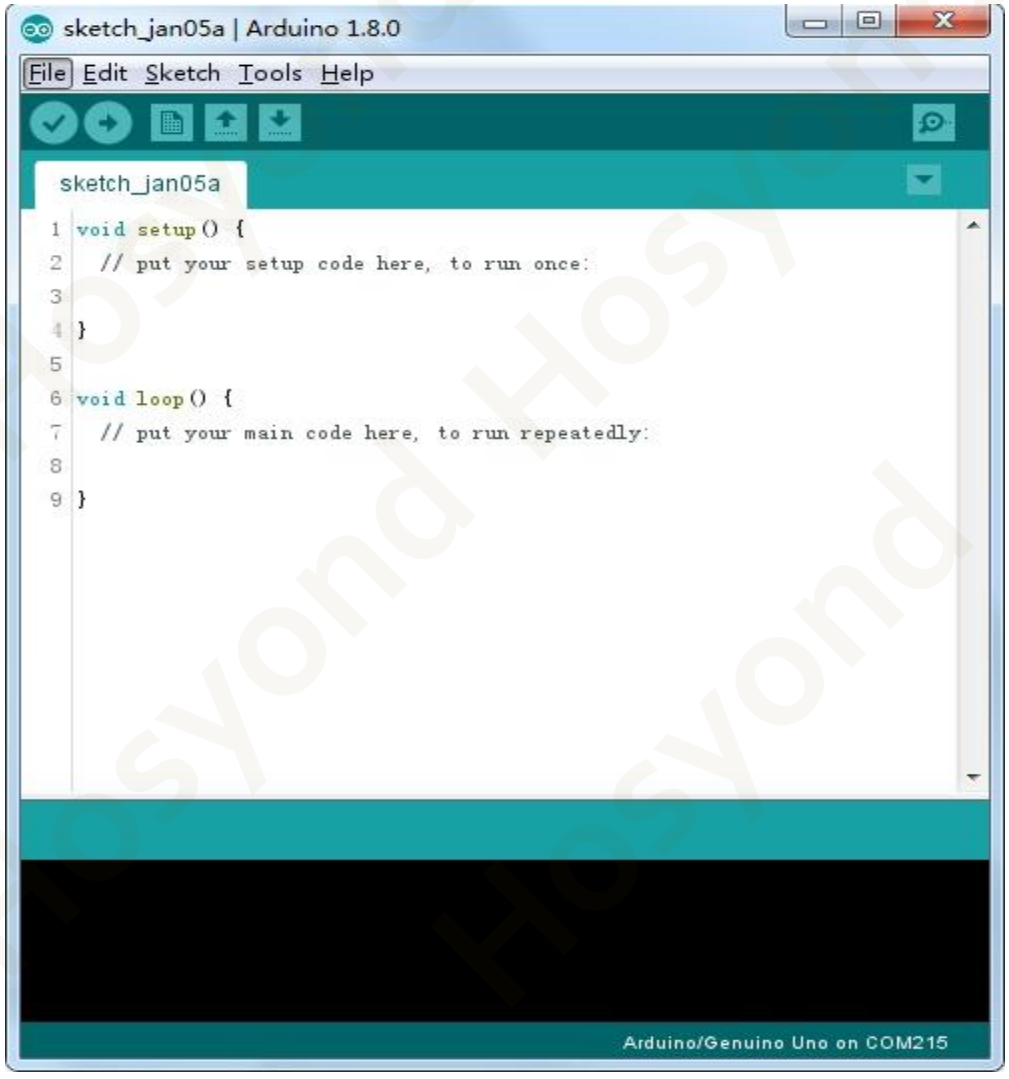


Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



Next, the following icon appears on the desktop

Double-click to enter the desired development environment

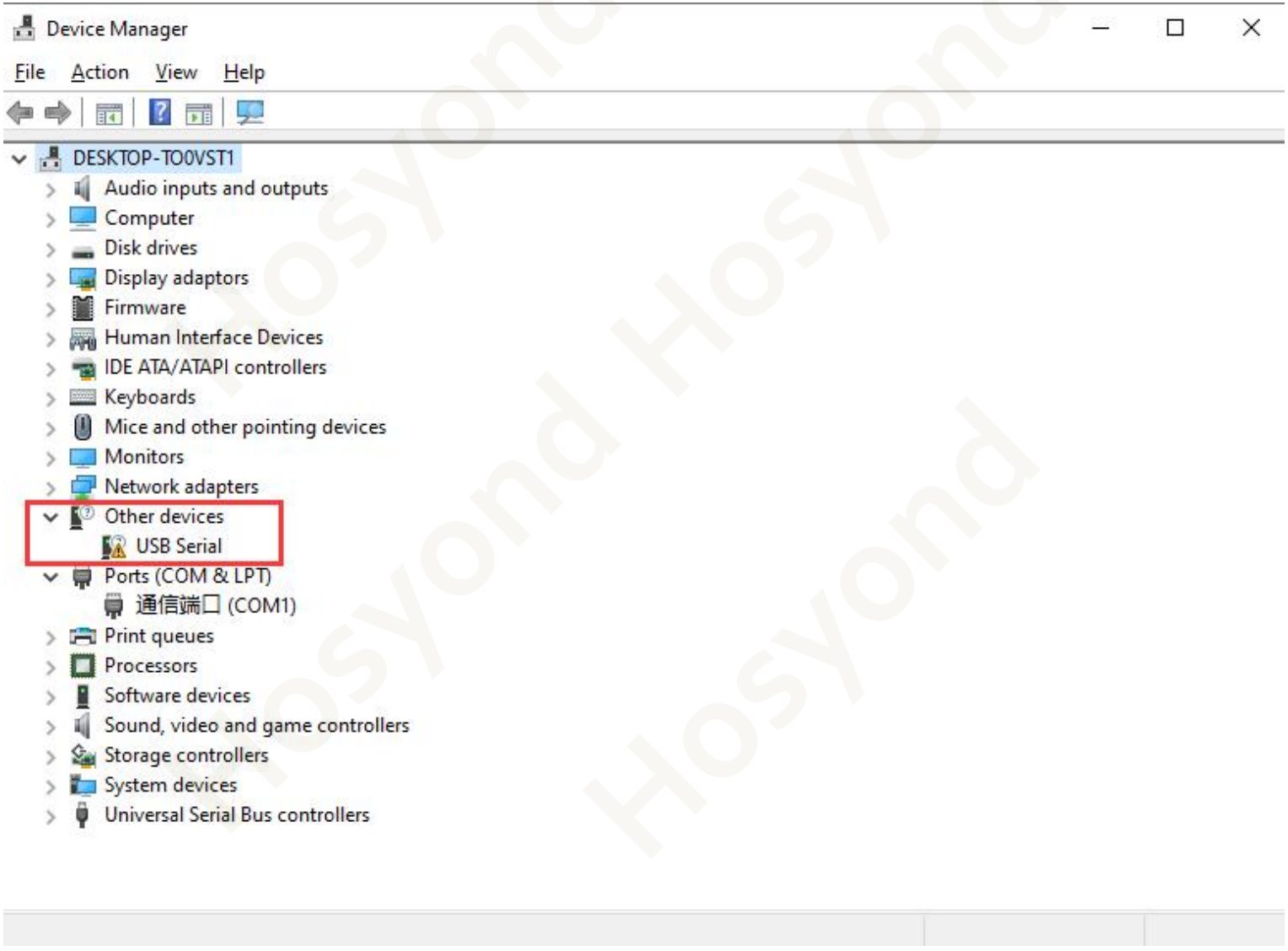


Lesson 3 How to Install Arduino Driver

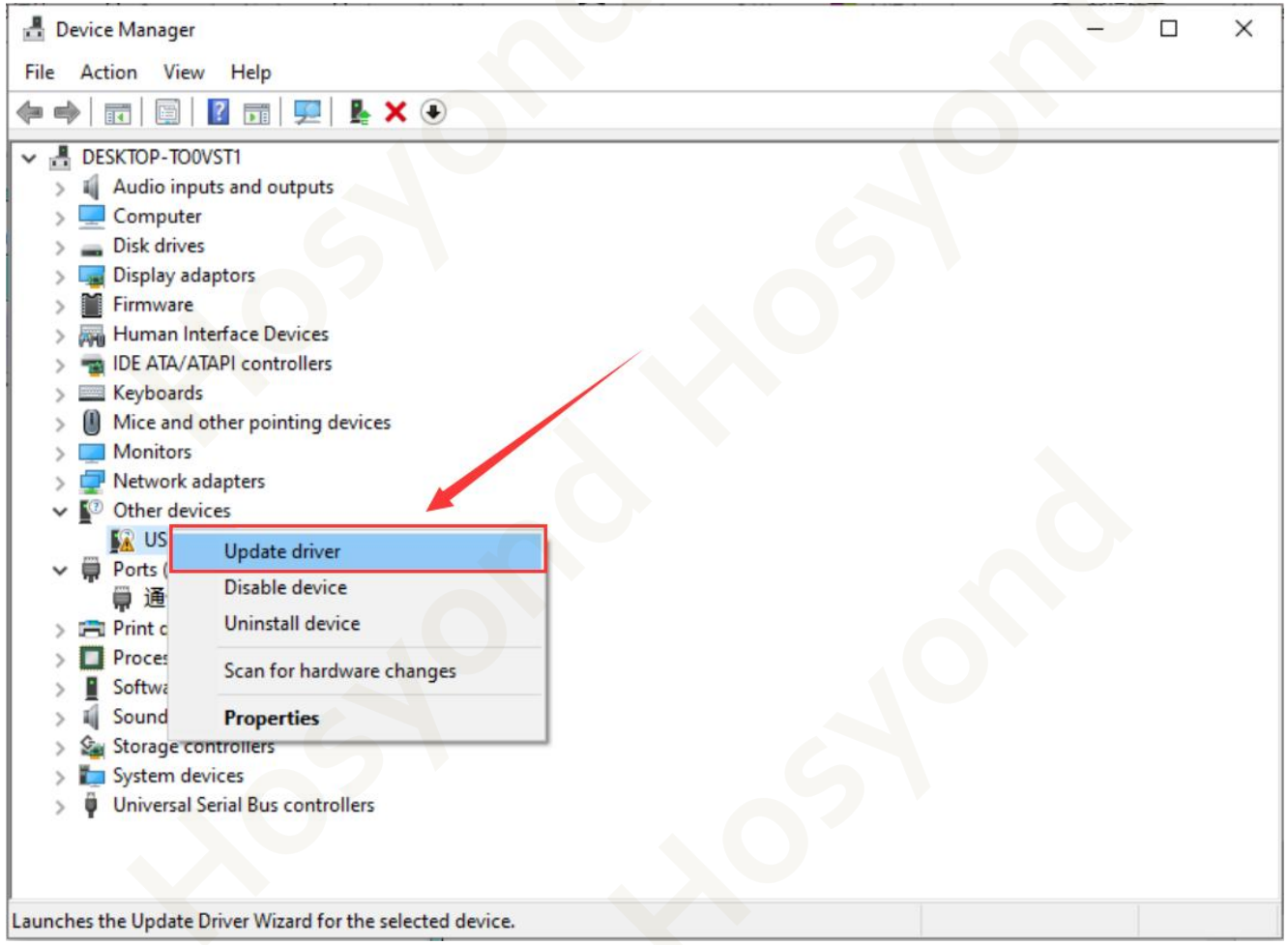
Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the Window system.

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.

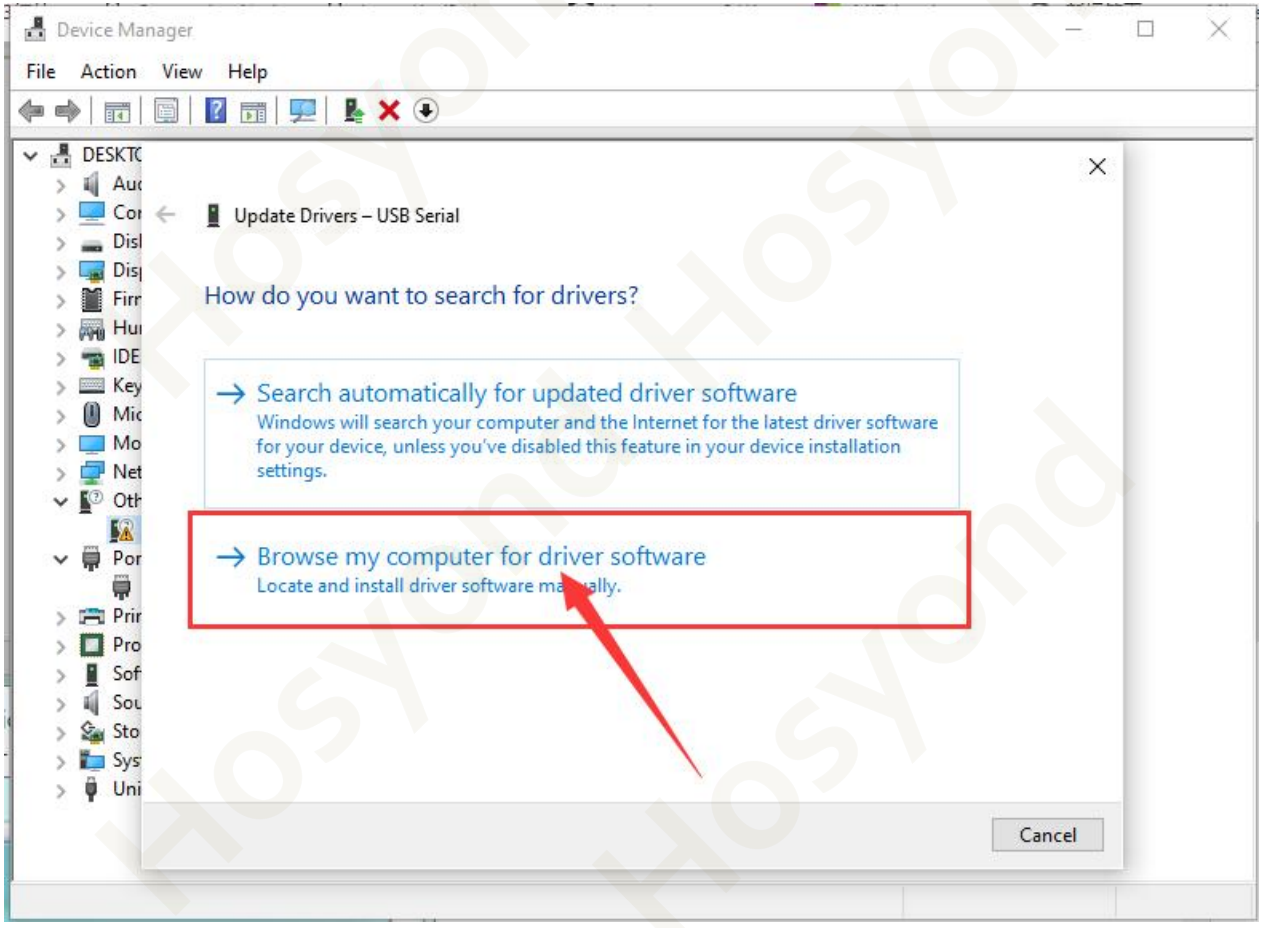
When you connect UNO board to your computer at the first time, right click the icon of your "Computer" —>for "Properties"—> click the "Device manager" , under "Other Devices"or"USB-Serial", you should see an icon for "Unknown device" with a little yellow warning triangle next to it. This is your Arduino.Or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


Device Manager
File Action View Help
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

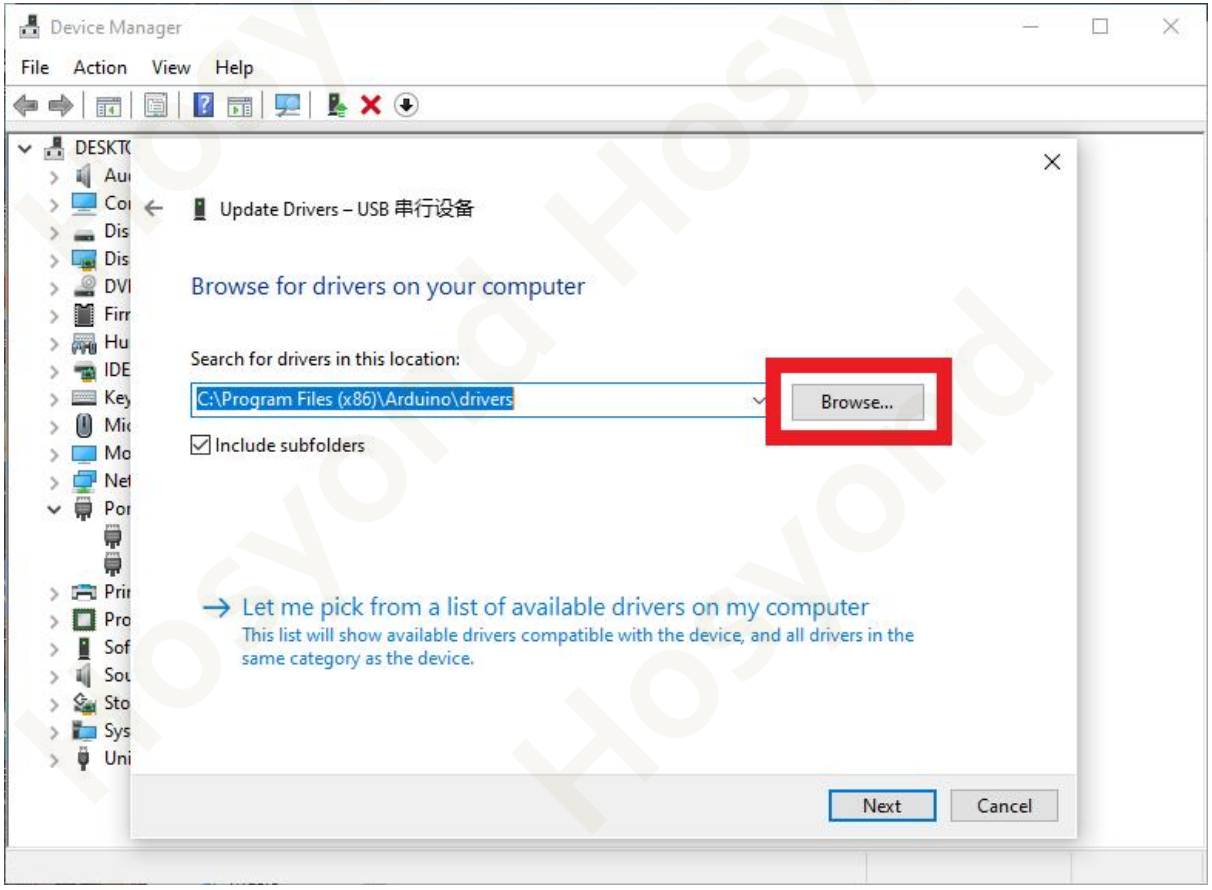
Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



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”.

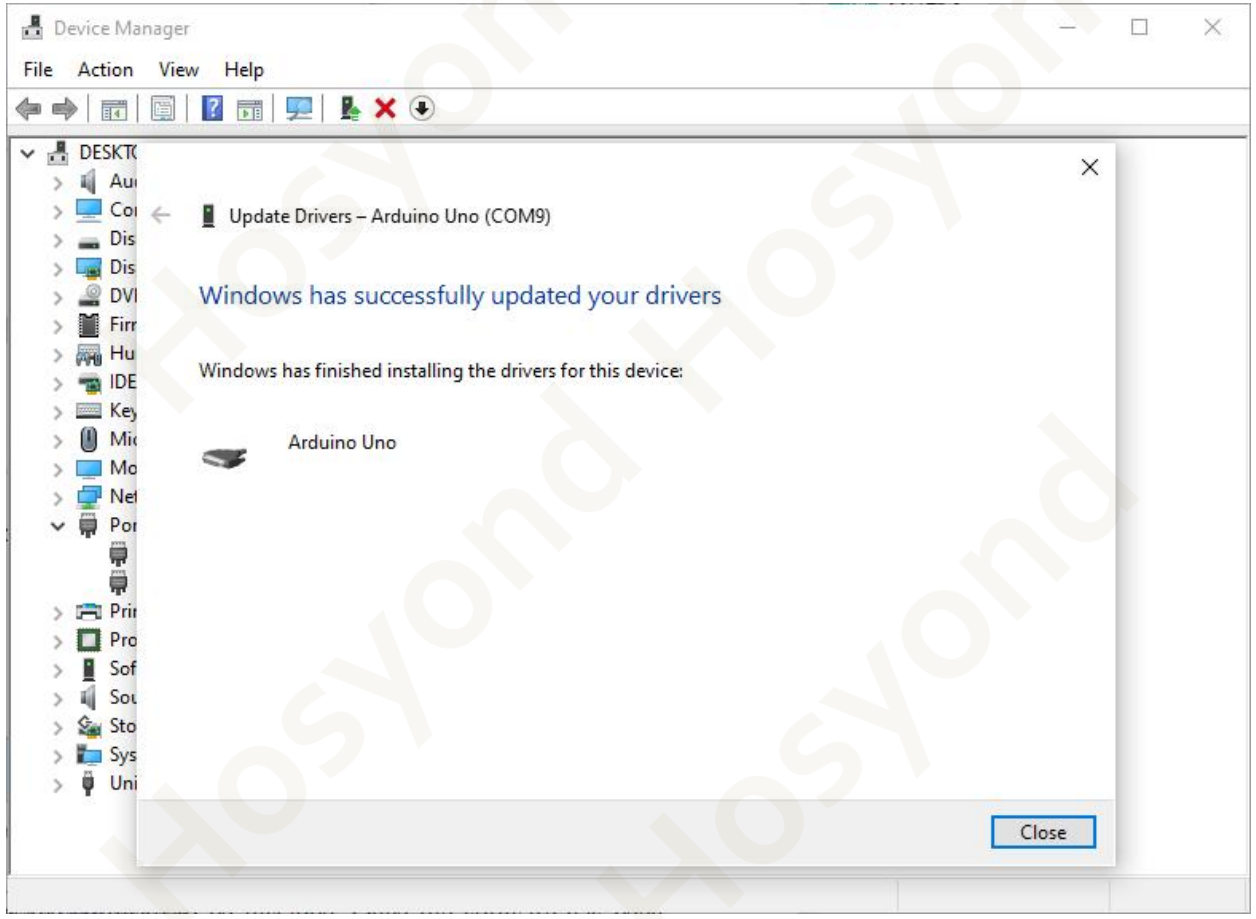


You will then be prompted to either ‘Search Automatically for updated driver software’ or ‘Browse my computer for driver software’. Select the option to browse and navigate to the :**C\Program Files(x86)\Arduino\drivers.**(Note: Here is the path you choose to install arduino IDE. The path chosen in the installation tutorial in the previous section is that, so the path I chose is**C\Program Files(x86)\Arduino\drivers**)

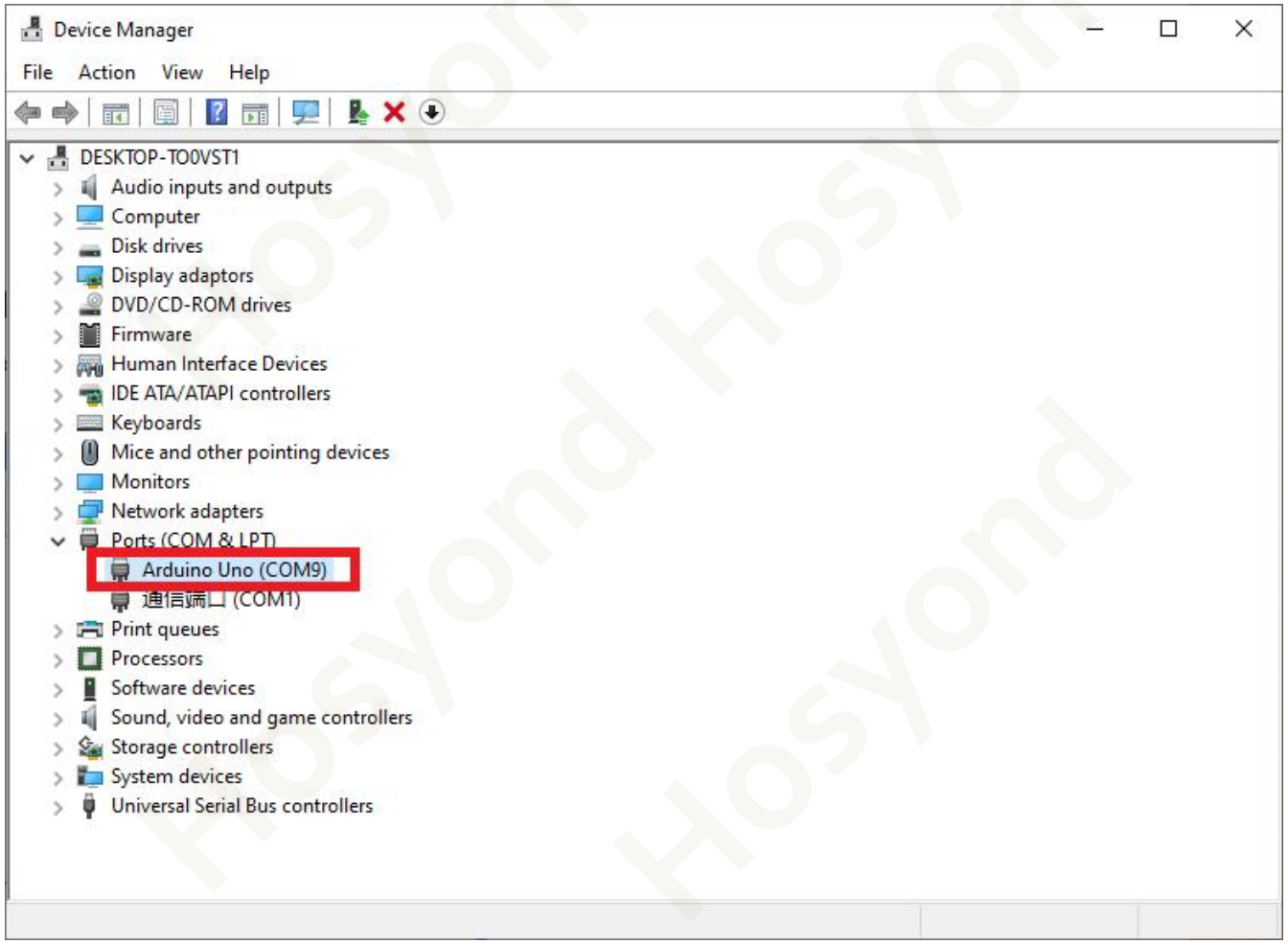


Click “Next” and you may get a security warning, if so, allow the software to be installed.

Once the software has been installed, you will get a confirmation message. Installation completed,click “Close”.



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.



Lesson 4 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.

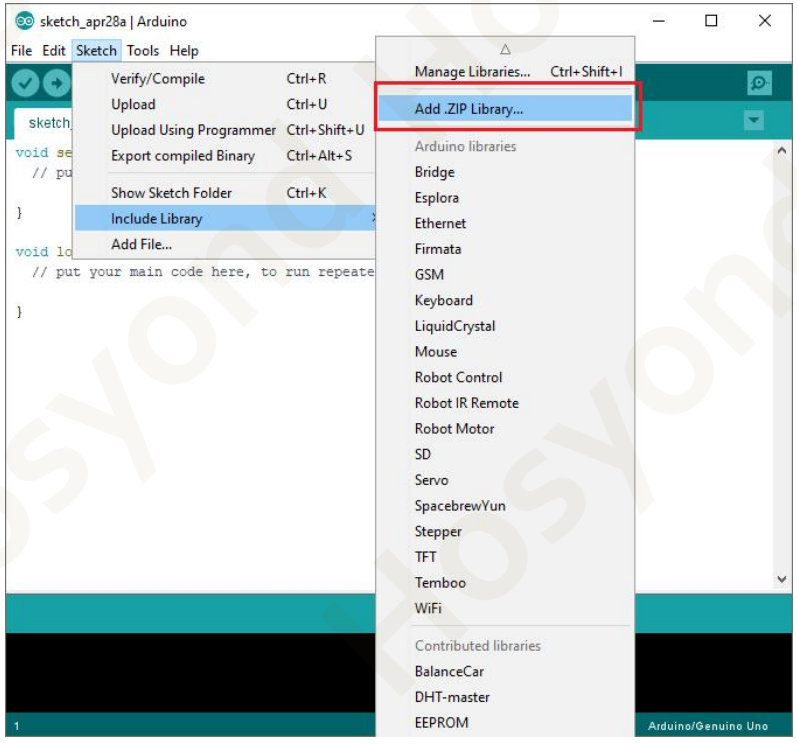
What are Libraries?

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.

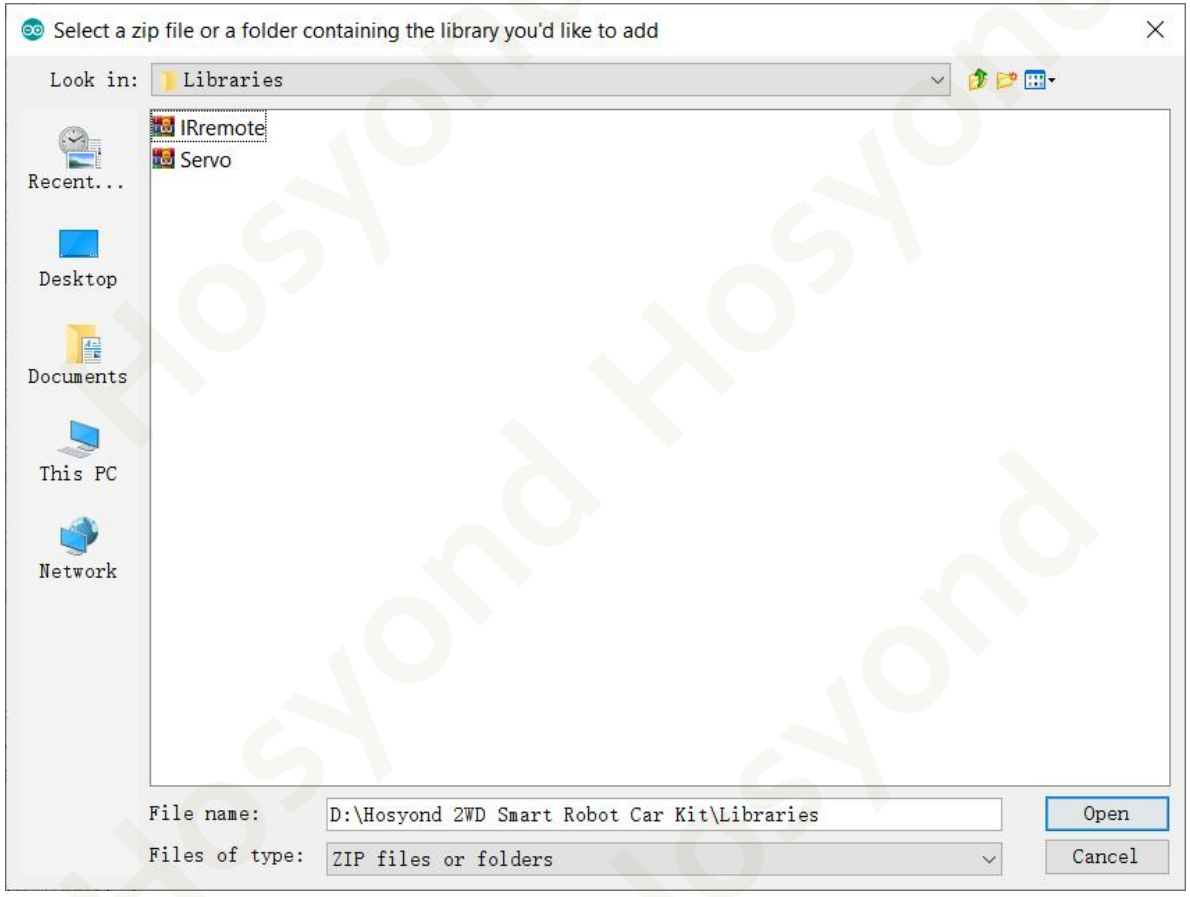
Importing a .zip Library

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".

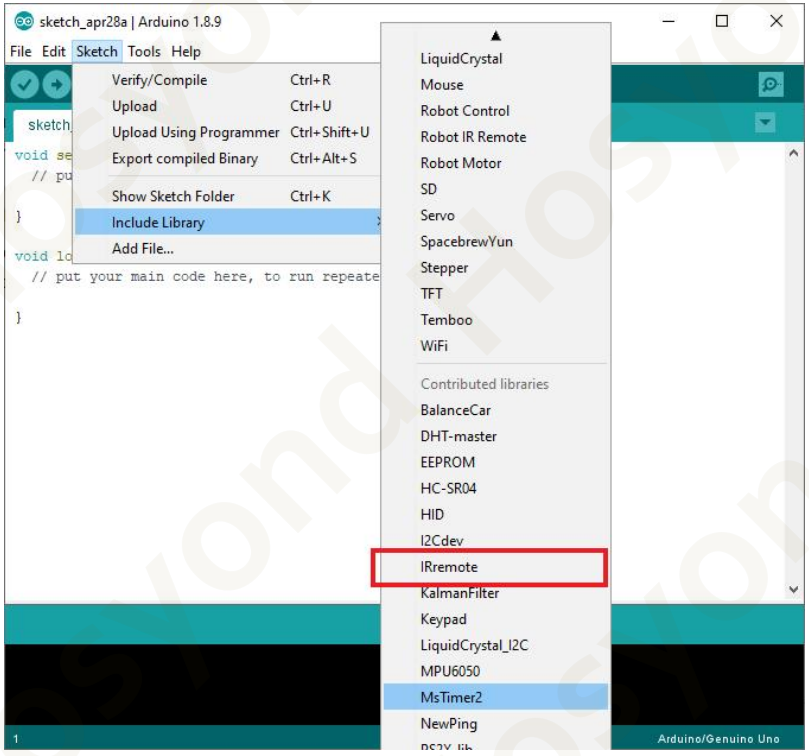


You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.



(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.



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 5 Blink Test

Overview

In this Project, 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.

Component Required:

1 x Uno Board

Principle

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.

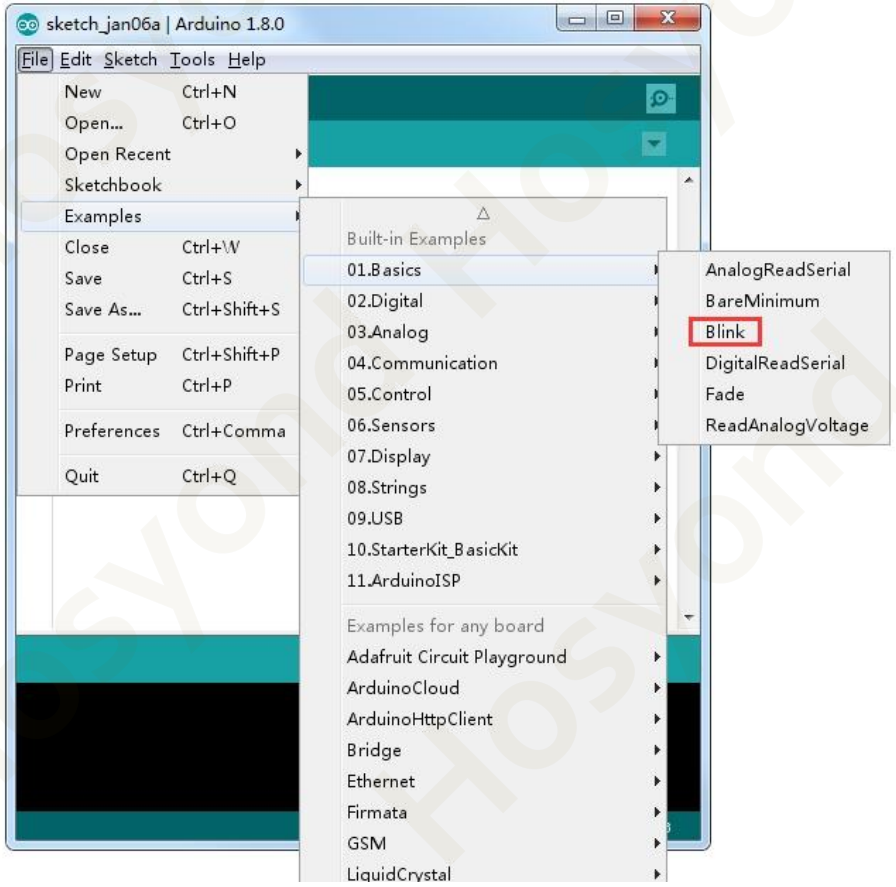
You may find that your UNO board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

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

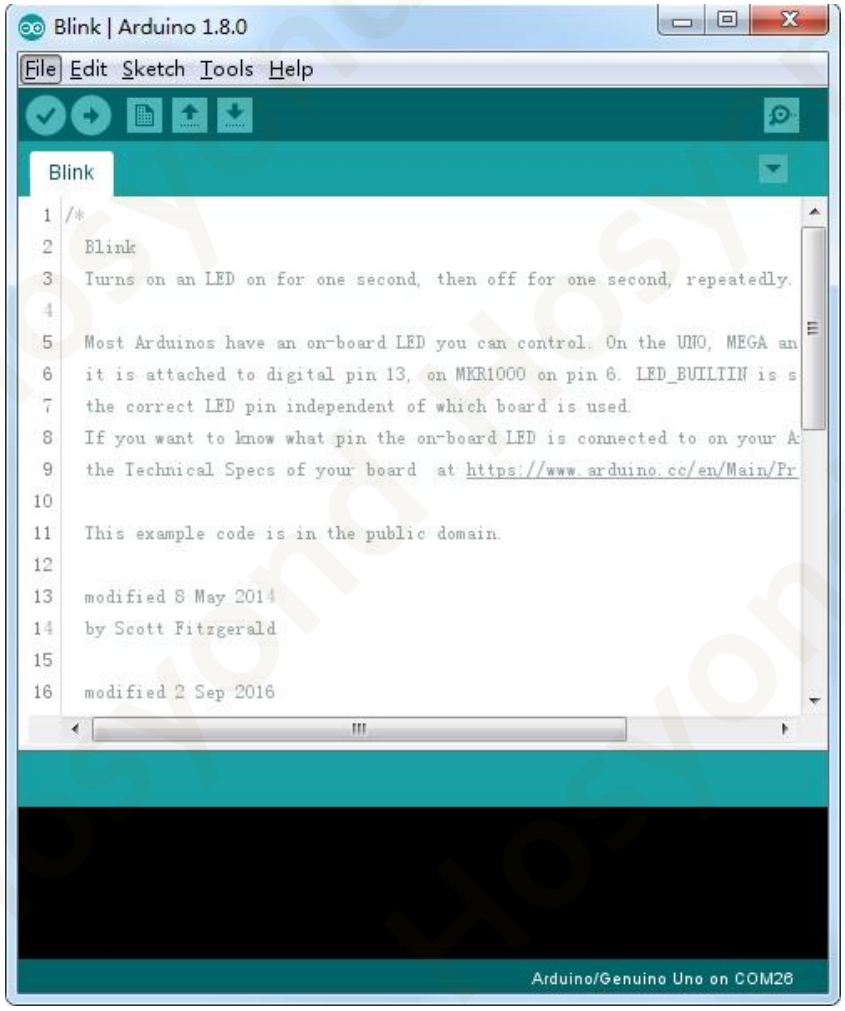
In the previous chapter-How to install Arduino IDE, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO board. The time has now come to put that connection to the test and program your UNO board.

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



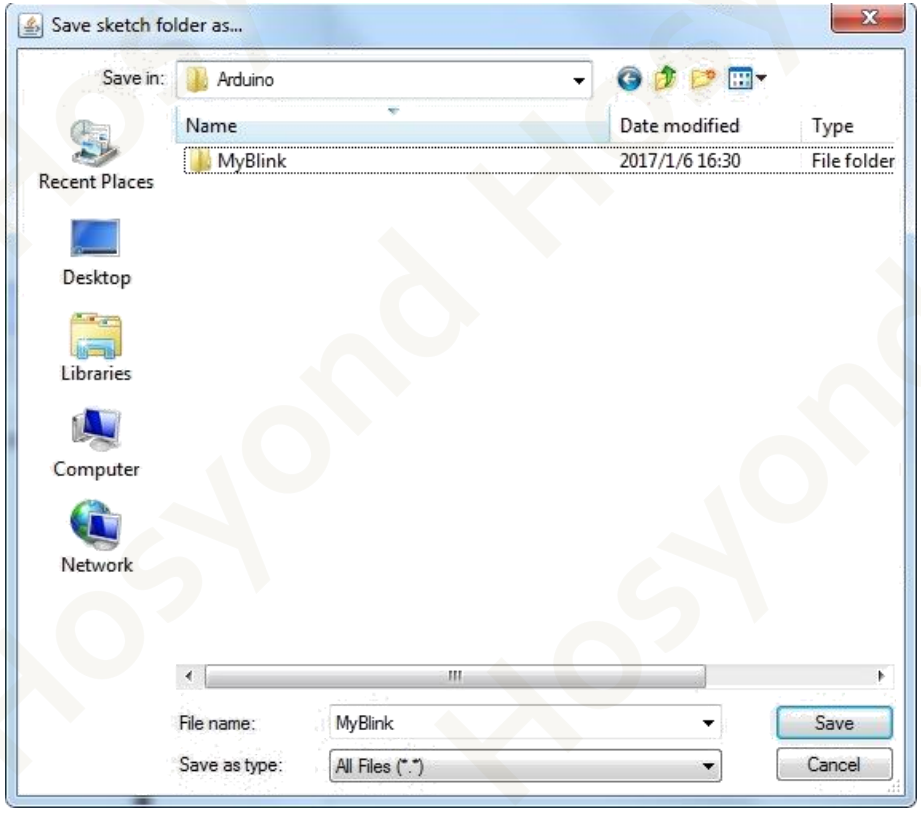
When the sketch window opens, enlarge it so that you can see the entire sketch in the window.



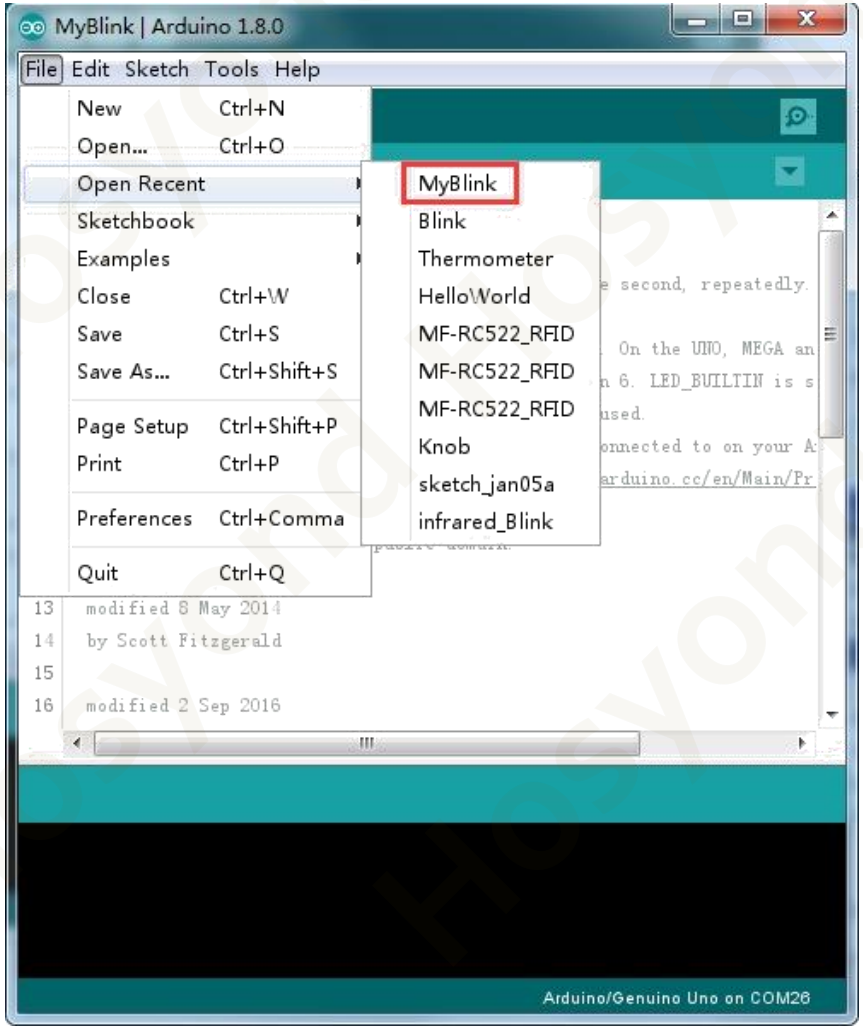
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

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

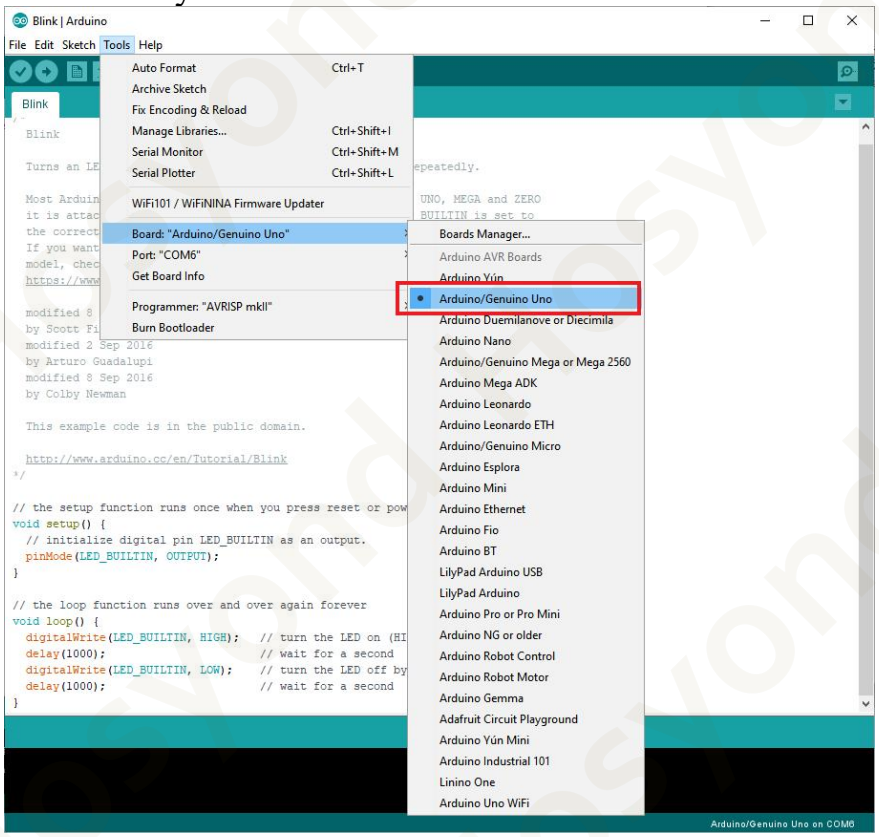
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.

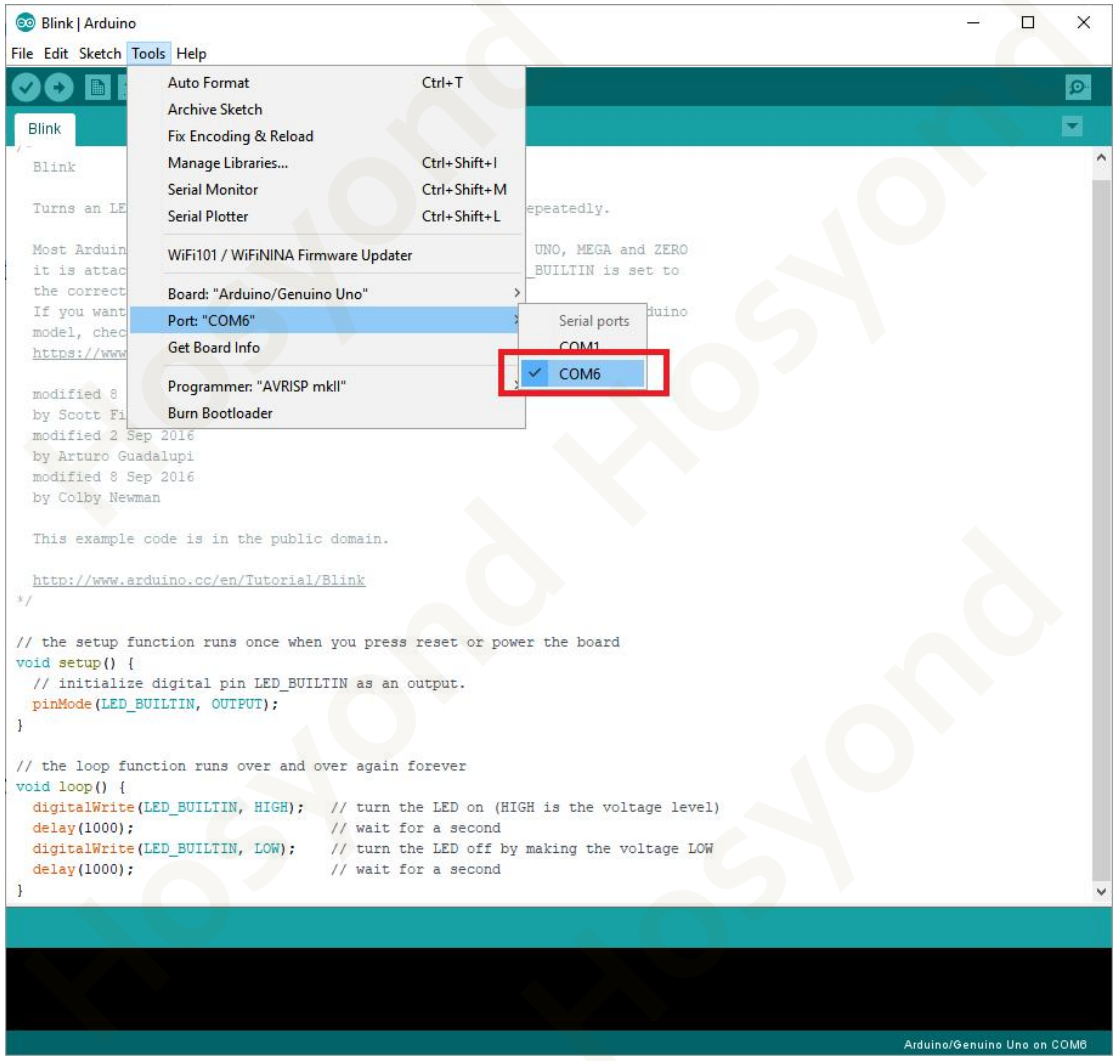


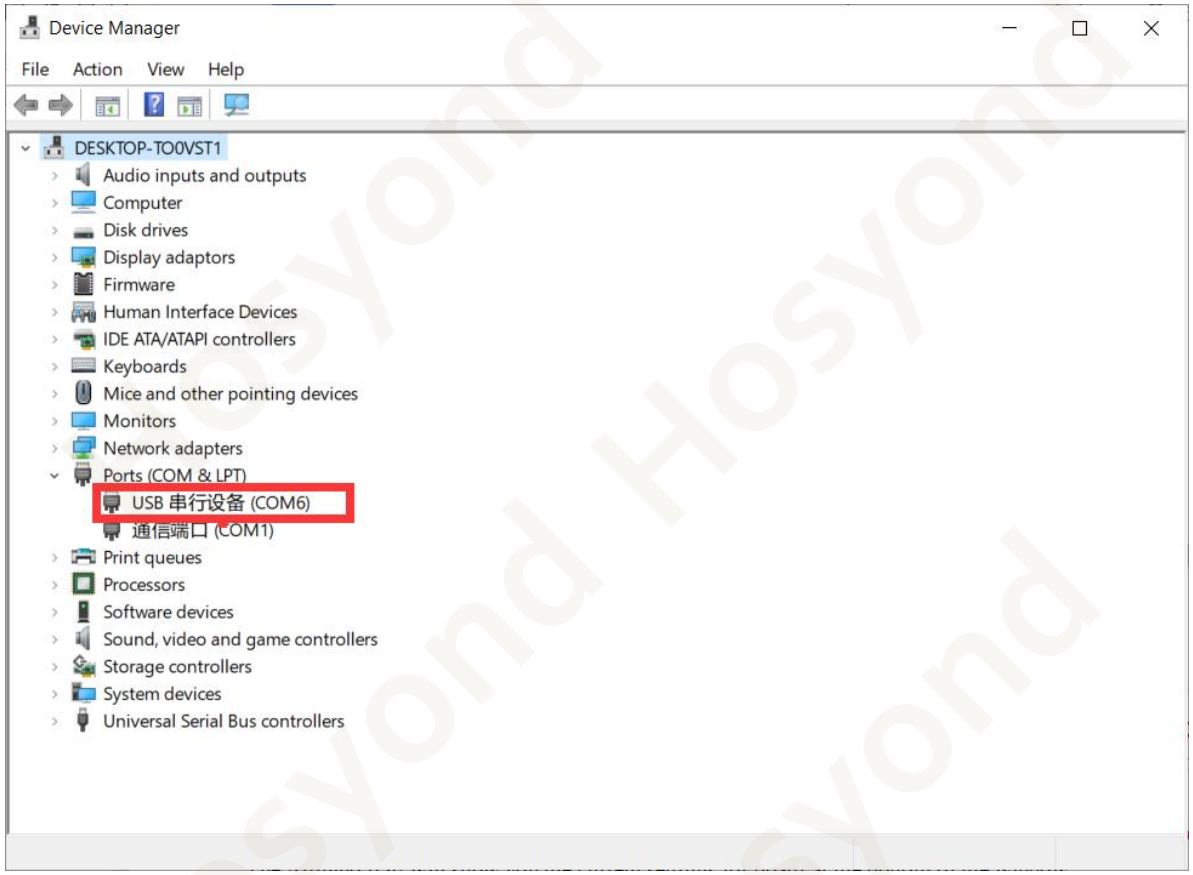
You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



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



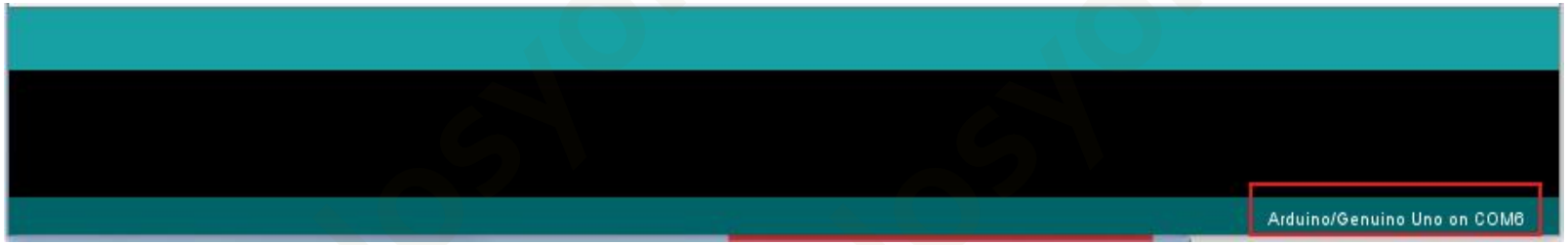

Blink | Arduino
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM6
Blink
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
}
Arduino/Genuino Uno on COM6



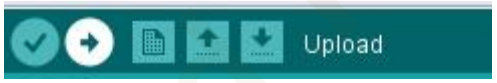
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using 2560, then you will have to choose Mega 2560 as the Board Type, other choices can be made in the same manner. And the

Serial Port displayed for everyone is different, despite COM 6 chosen here, it could be COM3 or COM4 on your computer. A right COM port is supposed to be COMX (arduino XXX), which is by the certification criteria.

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

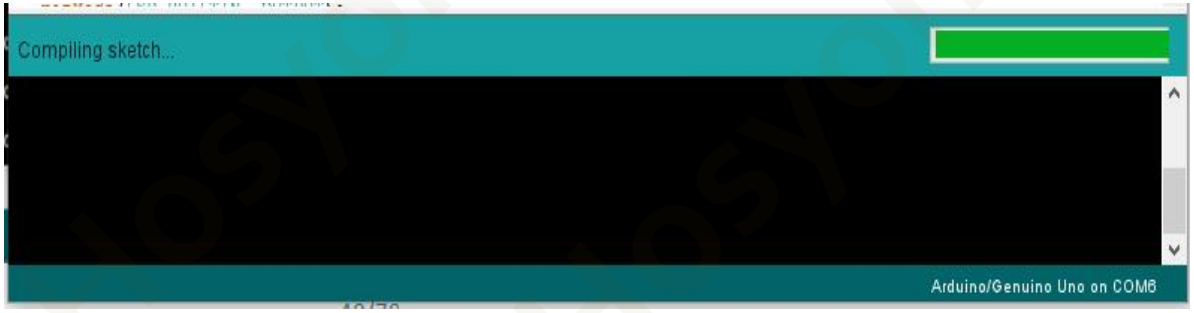


Click on the 'Upload' button. The second button from the left on the toolbar.

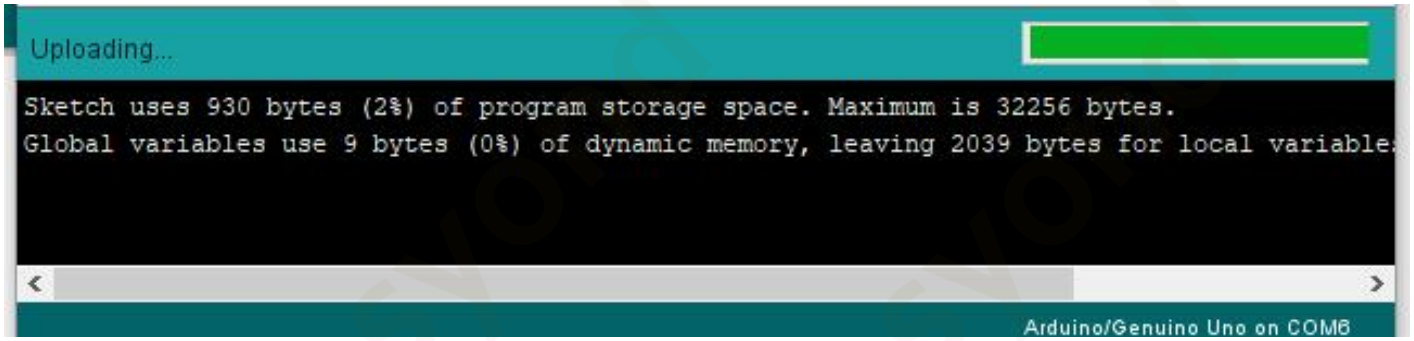


If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

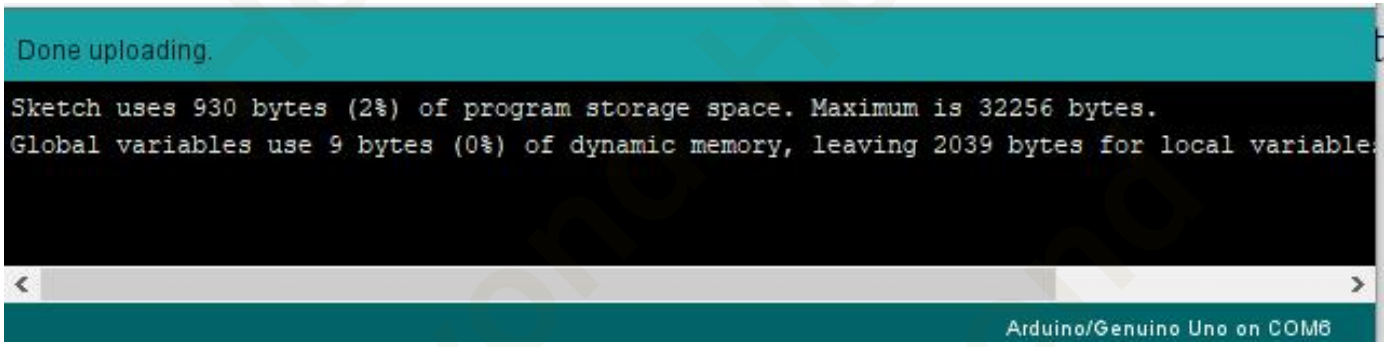
]

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 Project 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.

Single line comments start with // and everything up until the end of that line is considered a comment.

The first line of code is: int led = 13;

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

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

```
void setup() {
// initialize 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);
}
```

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.

// turn the LED off (LOW is the voltage level) // wait for a second
// turn the LED on (HIGH is the voltage level) // wait for a second

```
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 6 Servo

About this lesson:

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

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

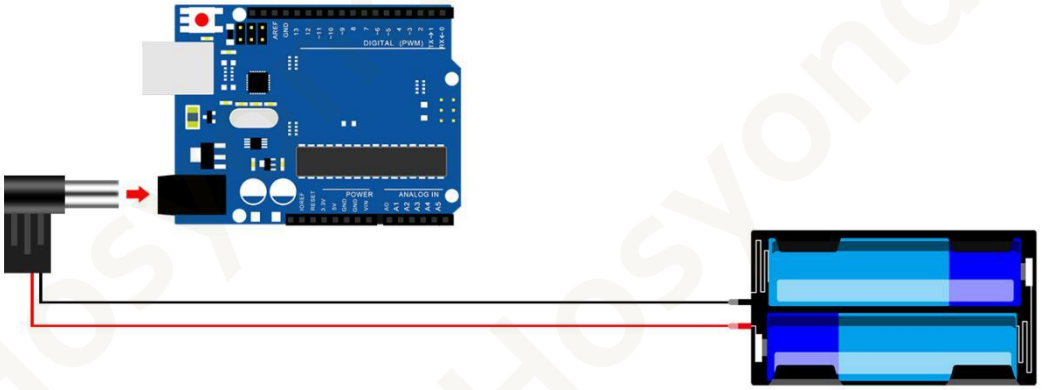
Servo motors are great devices that can turn to a specified position.

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!

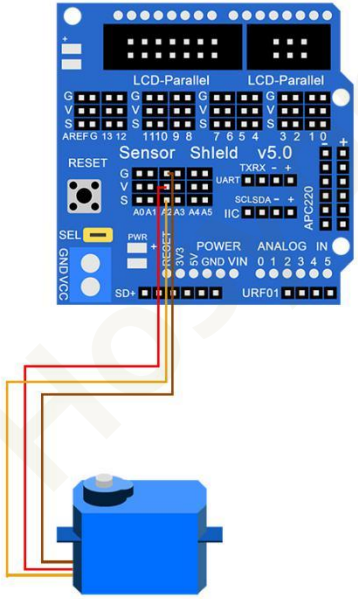
Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the flaps on a RC plane. With time, they found their uses in robotics, automation, and of course, the Arduino world.

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 and 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



Place V5 Expanding Board on UNO



Code

After uploading the code, the servo motor rotates from 0 degrees to 180 degrees, 1 degree at a time. Then rotate from 180 degrees to 0 degrees, one degree at a time.

After connecting,please open the the program and load up the code - Lesson_6_Servo onto your Arduino board. See Lesson 5 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 4.

Lesson 7 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:

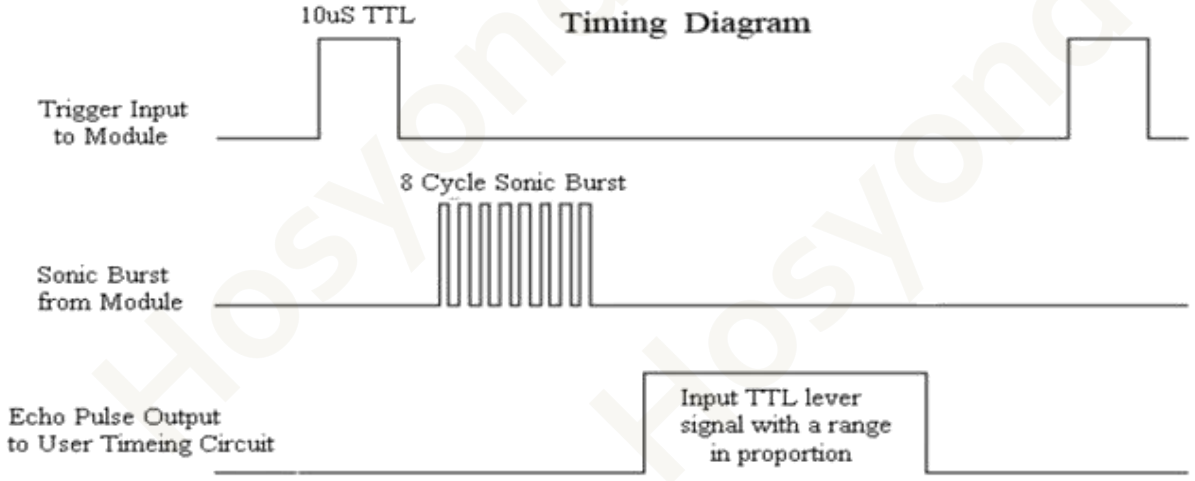
(1) Using IO trigger for at least 10us high level signal,

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

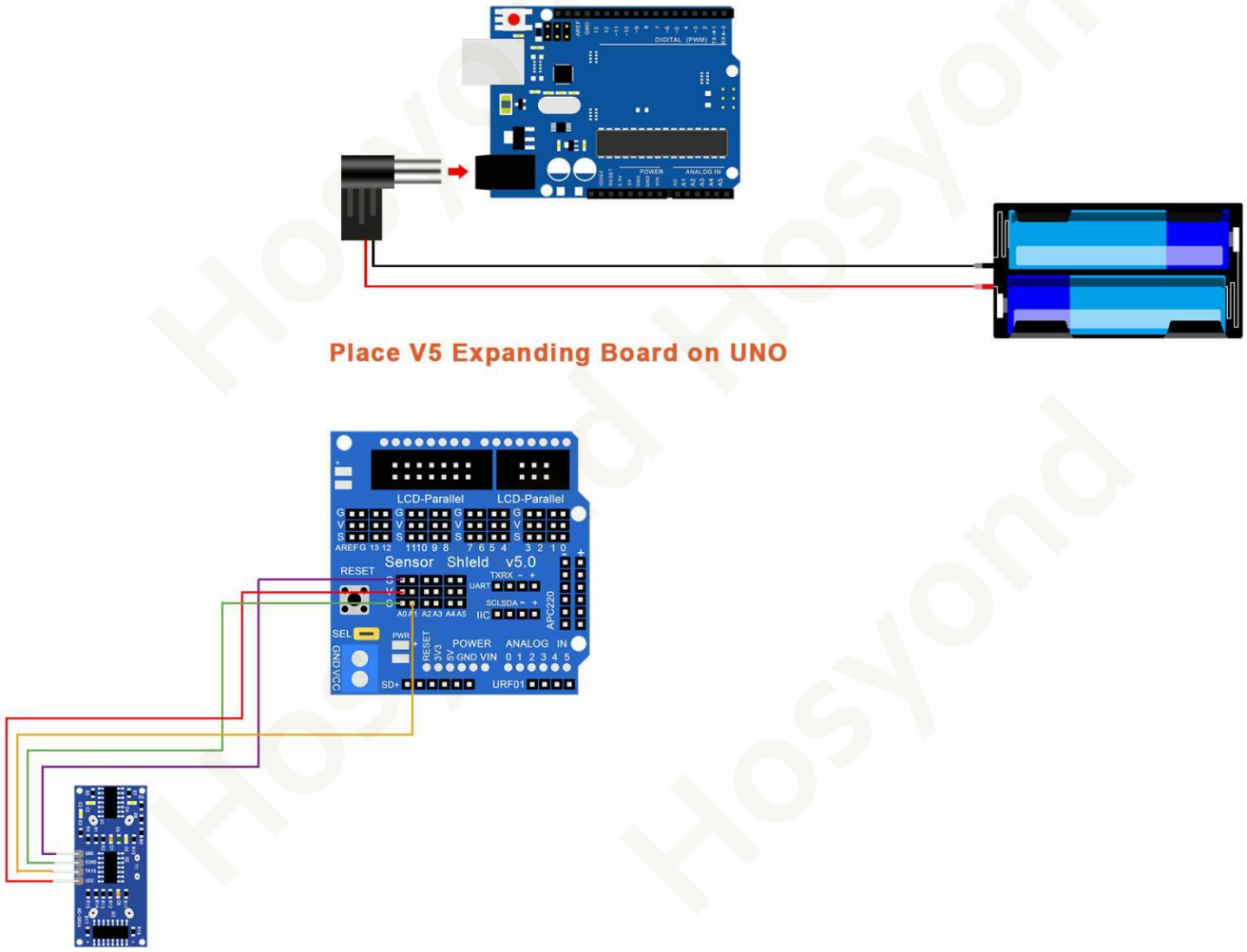
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic tore turning.

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



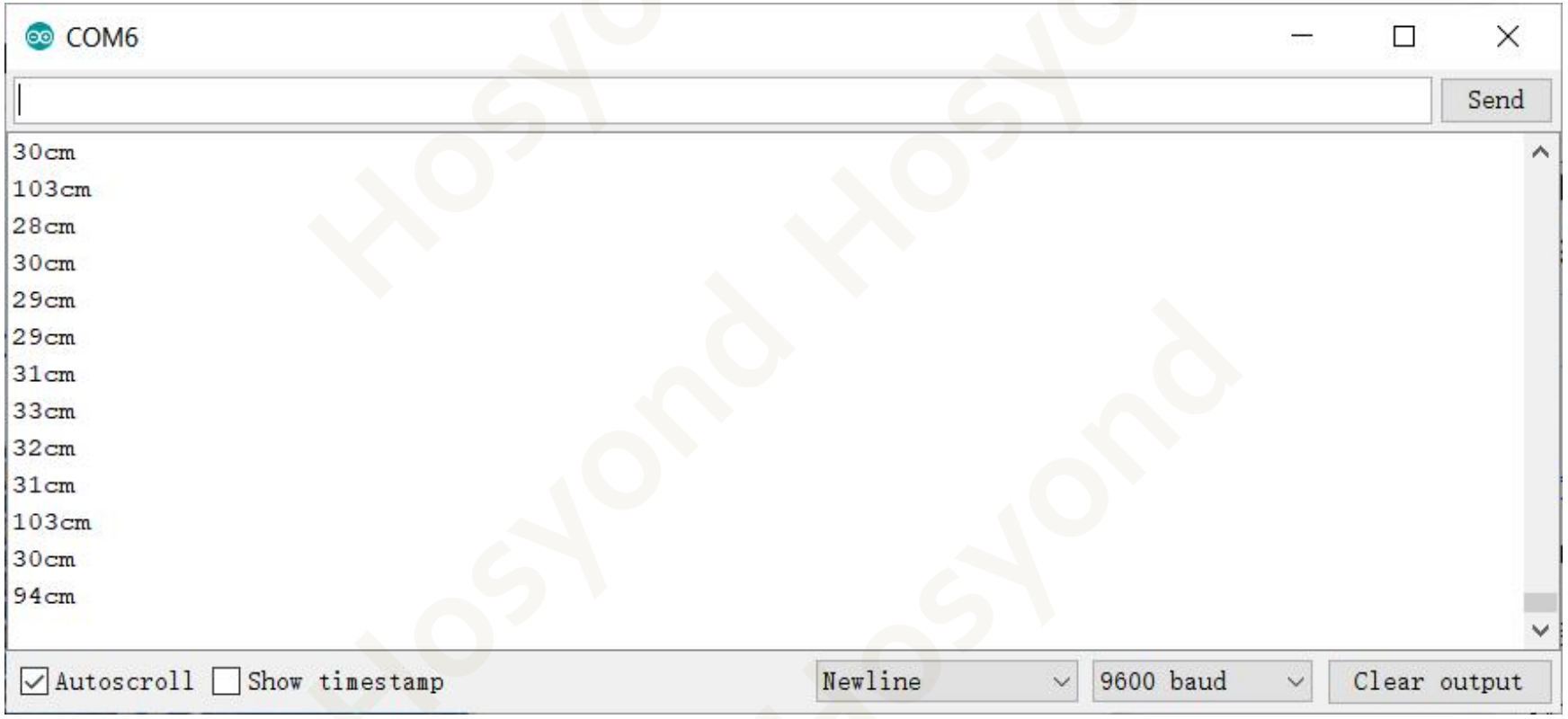
Code

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_7_Ultrasonic_Sensor_Module and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

After uploading the code, click the button in the upper right corner to open the serial monitor to view the measured distance

Then you can see the data as blow:



Lesson 8 Line-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 of the Arduino 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.The tracking sensor is actually an infrared sensor. The component used here is the TCRT5000 infrared tube.Its working principle is to use the different reflectivity of infrared light to the color, then convert the strength of the reflected signal into a current signal.During the process of detection, black is active at HIGH level, but white is active at LOW level. The detection height is 0-3 cm.By rotating the adjustable potentiometer on the sensor, it can adjust the detection sensitivity of the sensor.

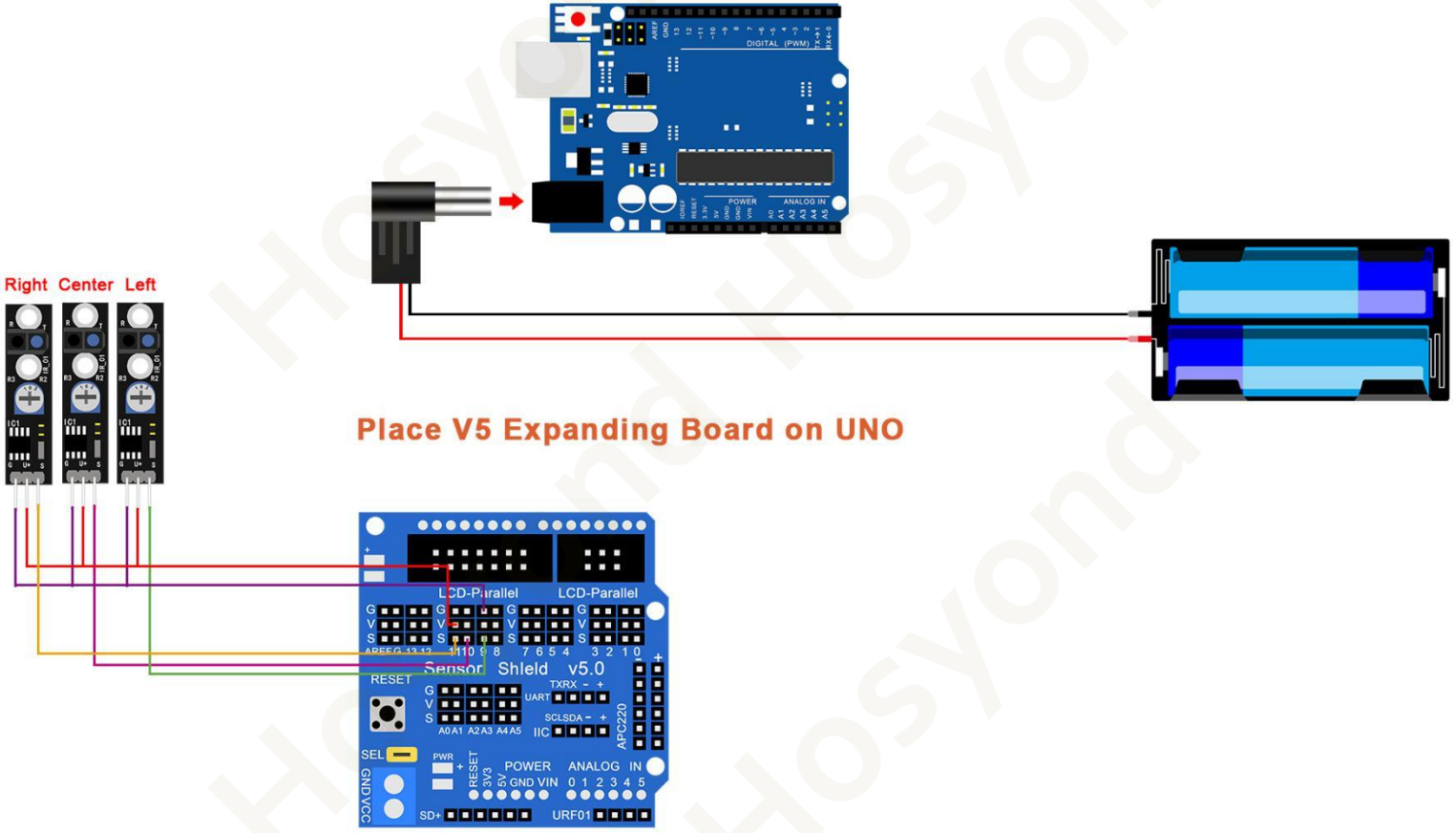
Specification:

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

How to use the Line-Tracking sensor

we read the signal level of Line-Tracking sensor to judge whether detect black or not.
When detects black, sensor's signal pin outputs HIGH (display 1).The red indicator light will turn off ; otherwise, output LOW (display 0),the red indicator light will turn on.
Show the result on the serial monitor.

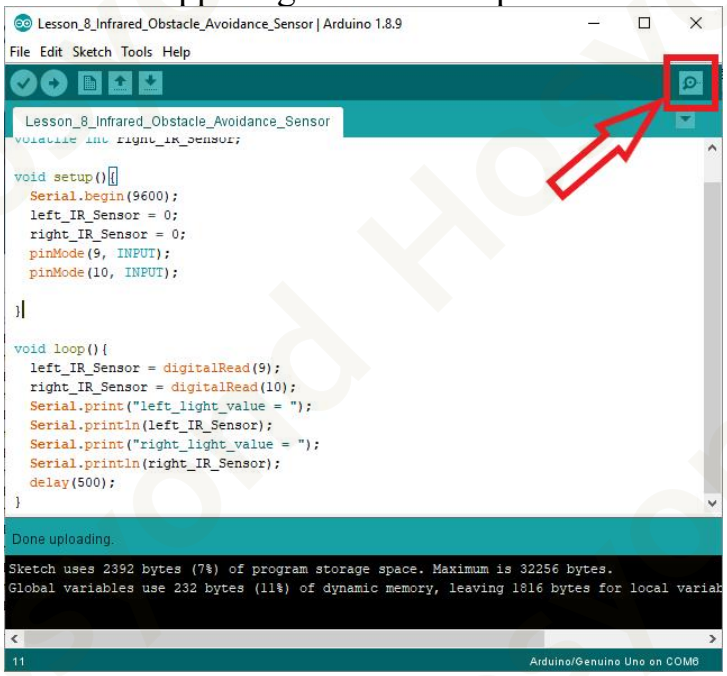
Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson_8_Line_Tracking_sensor and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

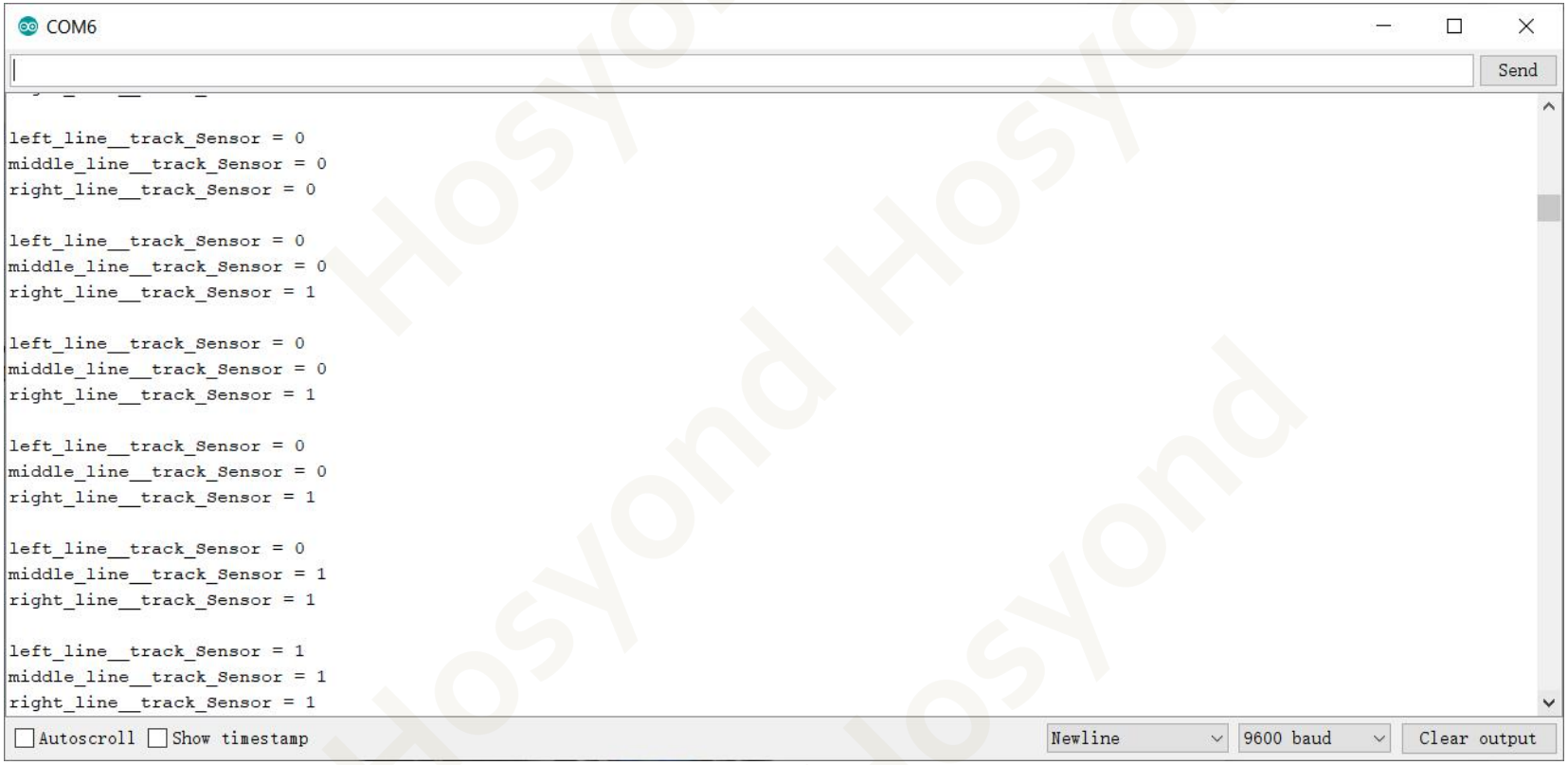
After uploading the code, click the button in the upper right corner to open the serial monitor to view the measured distance



Then you can see the data as blow:

When the line patrol sensor detects black, the serial monitor will receive "1", and the digital port will receive a high level, otherwise the serial monitor will receive "0", and the digital port will receive a low level. If the three sensors on the left and

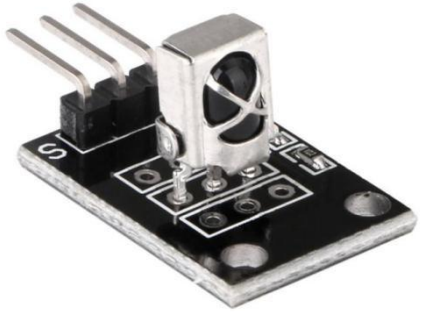
right and the middle detect black at the same time, the serial monitor receives three "1"s. If the left sensor detects black and the other two sensors detect non-black, the signals received by the serial monitor are "1" "0" "0".



Lesson 9 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.

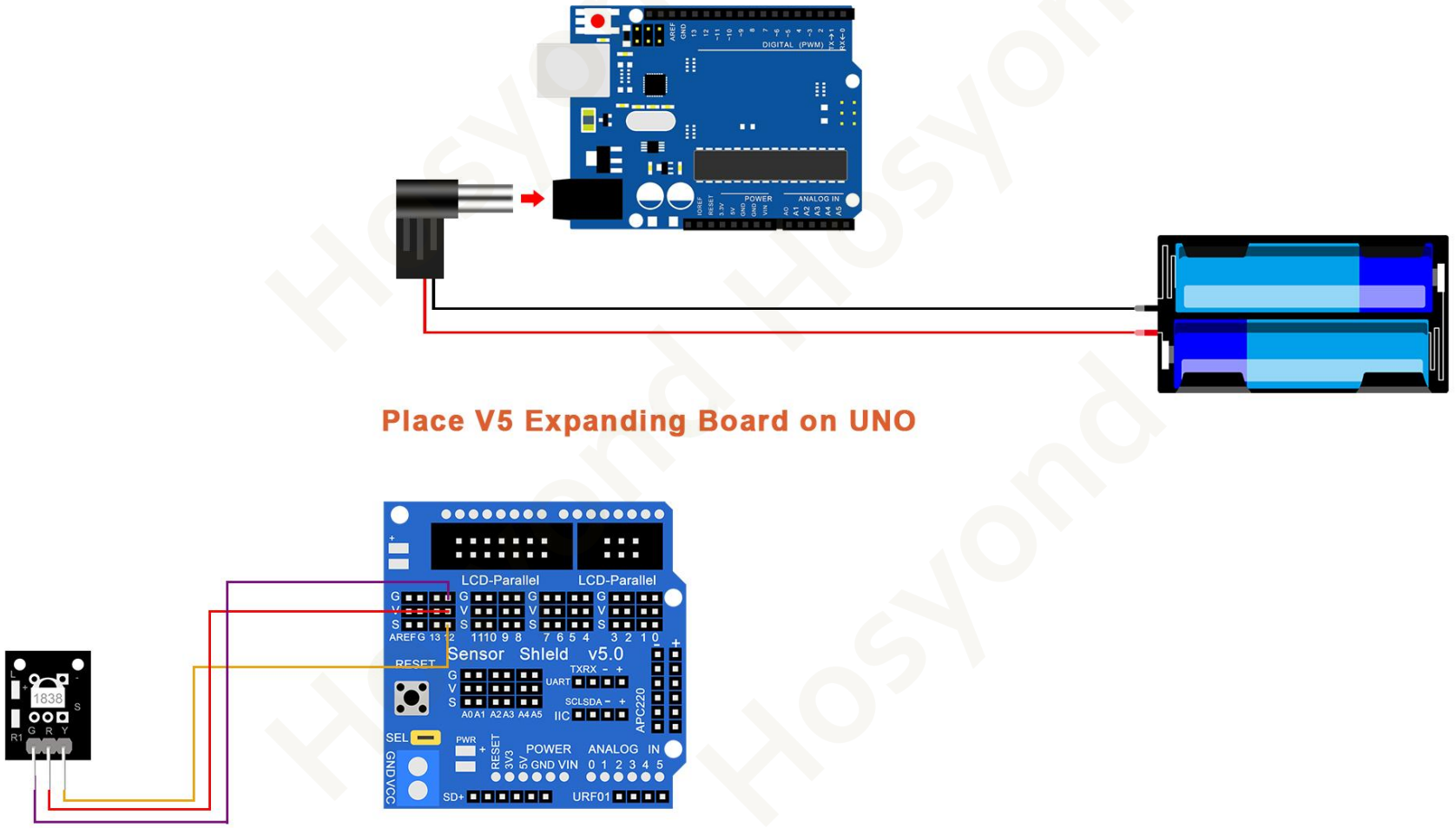


Introduction

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.
The "G" is the Ground, "R" is signal, and "Y"pin is Voltage 5V

Wiring diagram

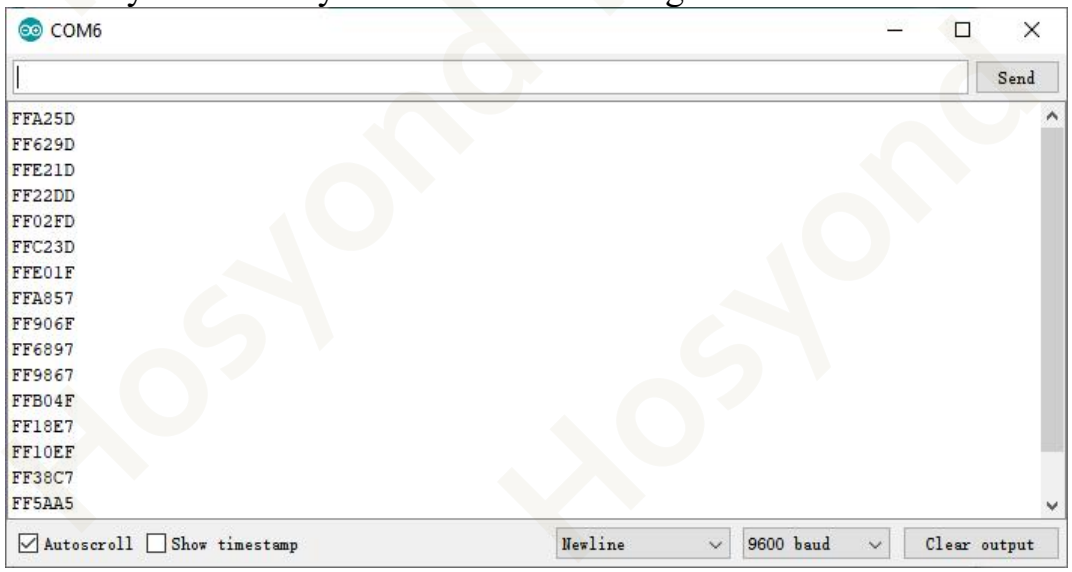


Code

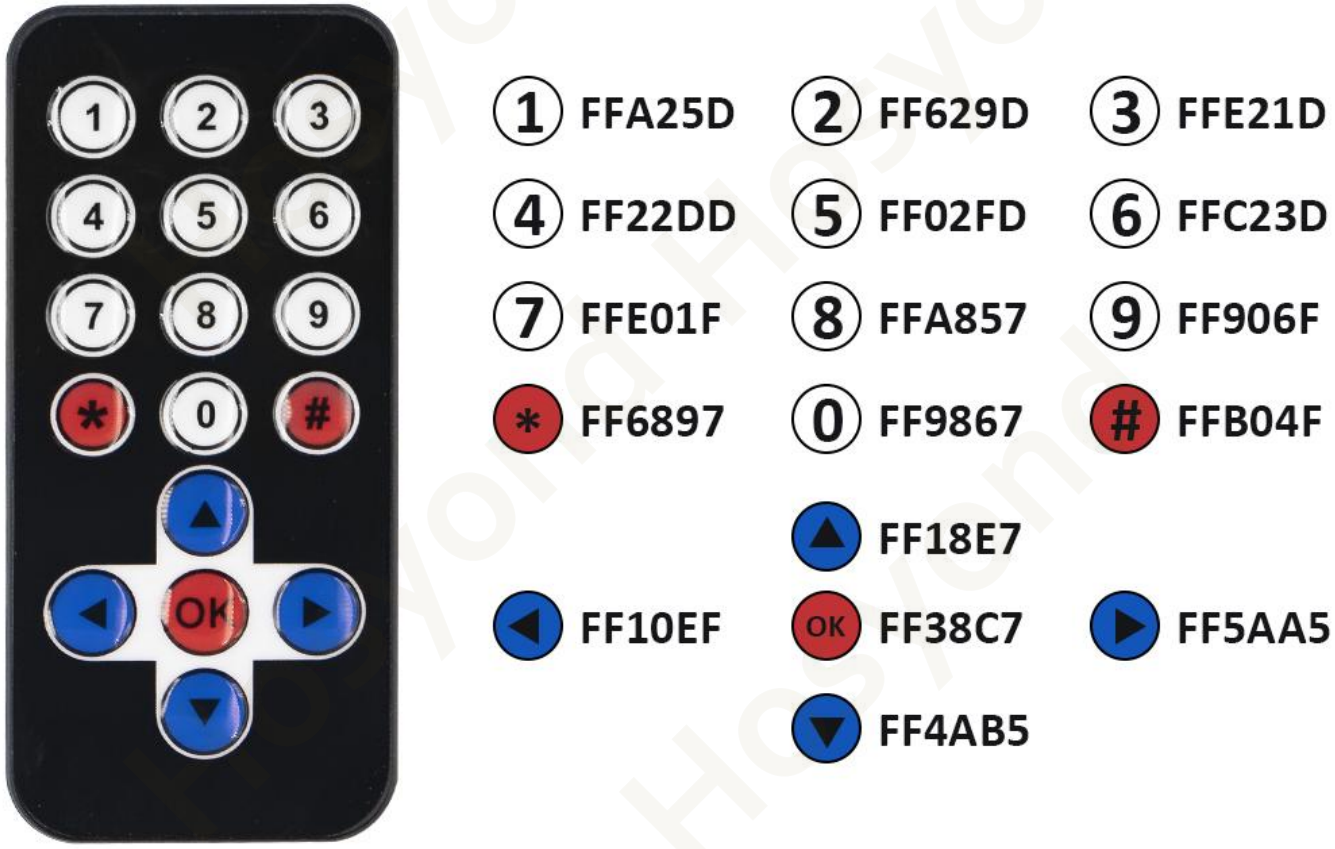
After wiring, please open the program in the code folder- Lesson_9_IR_Receiver_Module and click UPLOAD to upload the program. See Lesson5 for details about program uploading if there are any errors.

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 4.

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 10 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).

We use the serial port communication function of the Bluetooth module, use the app to control the Bluetooth connection of the mobile phone and the HC-06 Bluetooth module, and then the mobile phone app sends data, and the HC-06 Bluetooth module transmits the received data to the arduino uno through the serial port. The default communication baud rate of the HC-06 Bluetooth module is 9600.

The HC-06 Bluetooth module to LAFVIN UNO R3:

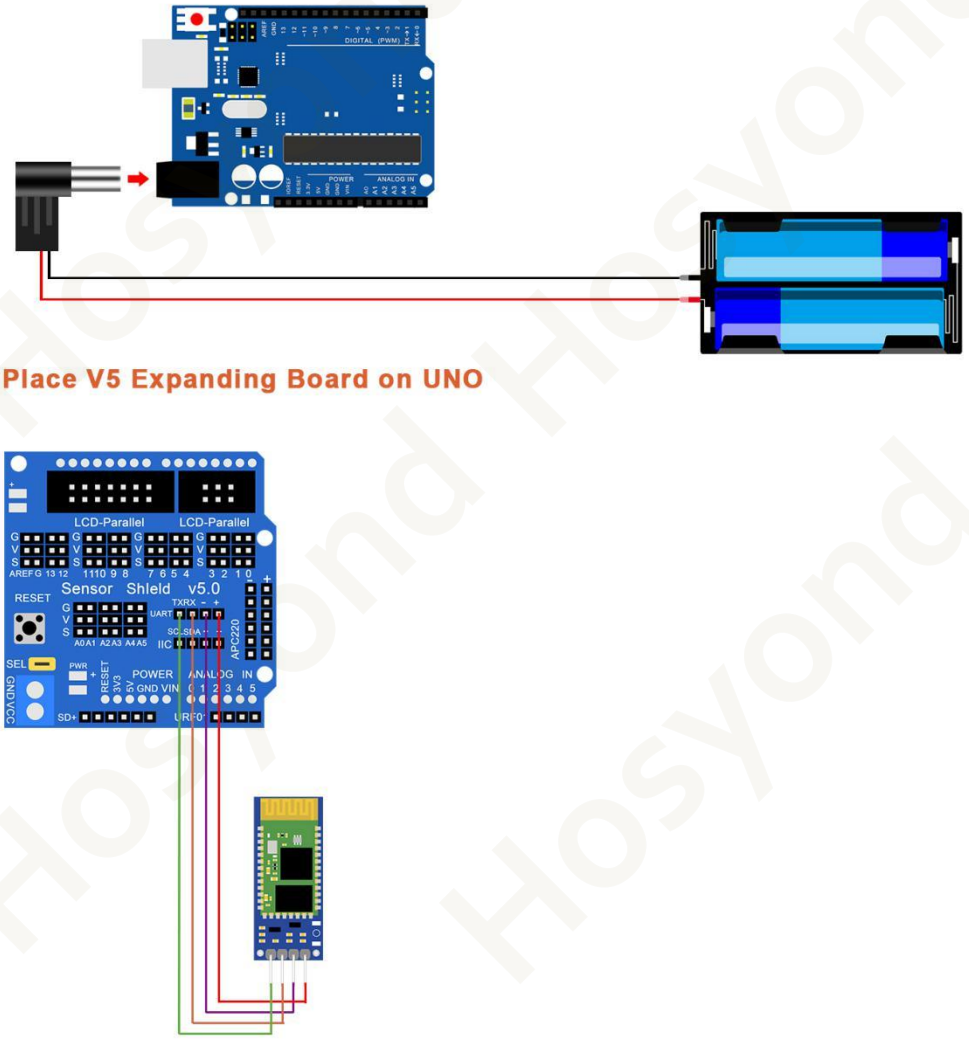
VCC>>>> +

GND>>>> -

TXD>>>>RX

RXD>>>>TX

Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson_10_Bluetooth_Module and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

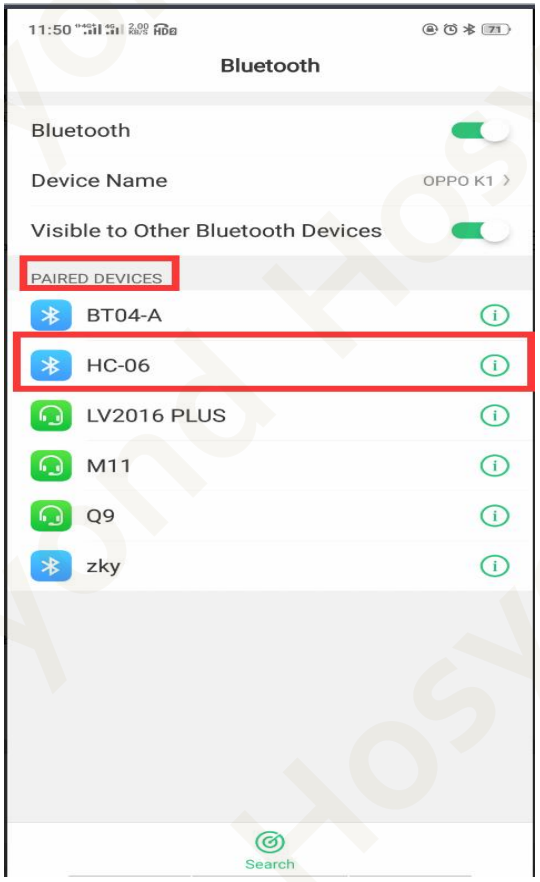
Attention:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.When uploading the code, CANNOT connect the Bluetooth module first; otherwise uploading fails! You are supposed to upload the code to control board, then connect the Bluetooth module.

Instructions for the use of app

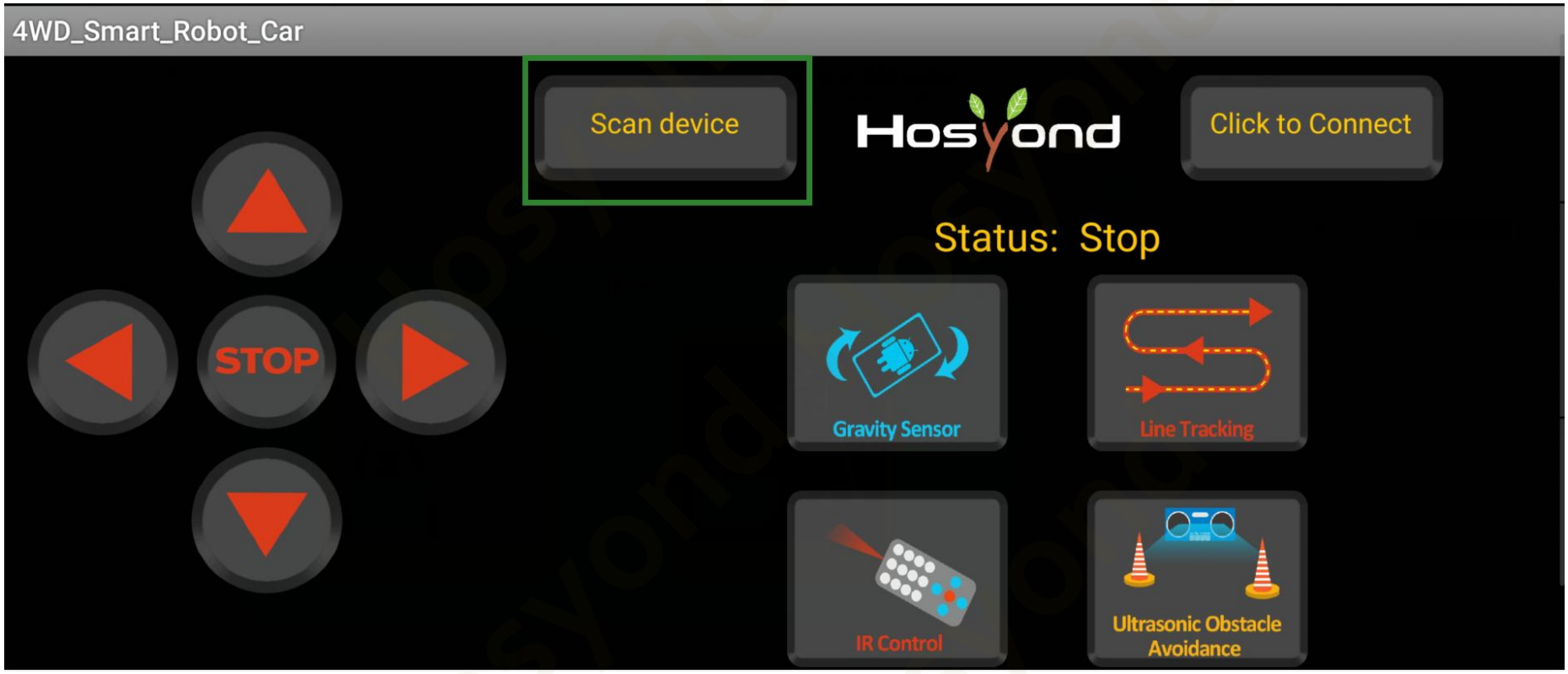
Firstly, download the "**4WD_Smart_Robot_Car_V1.apk**" file from the folder to your mobile phone and install it into an application software.

Then make sure the Bluetooth module is connected. 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 your phone is connected to the

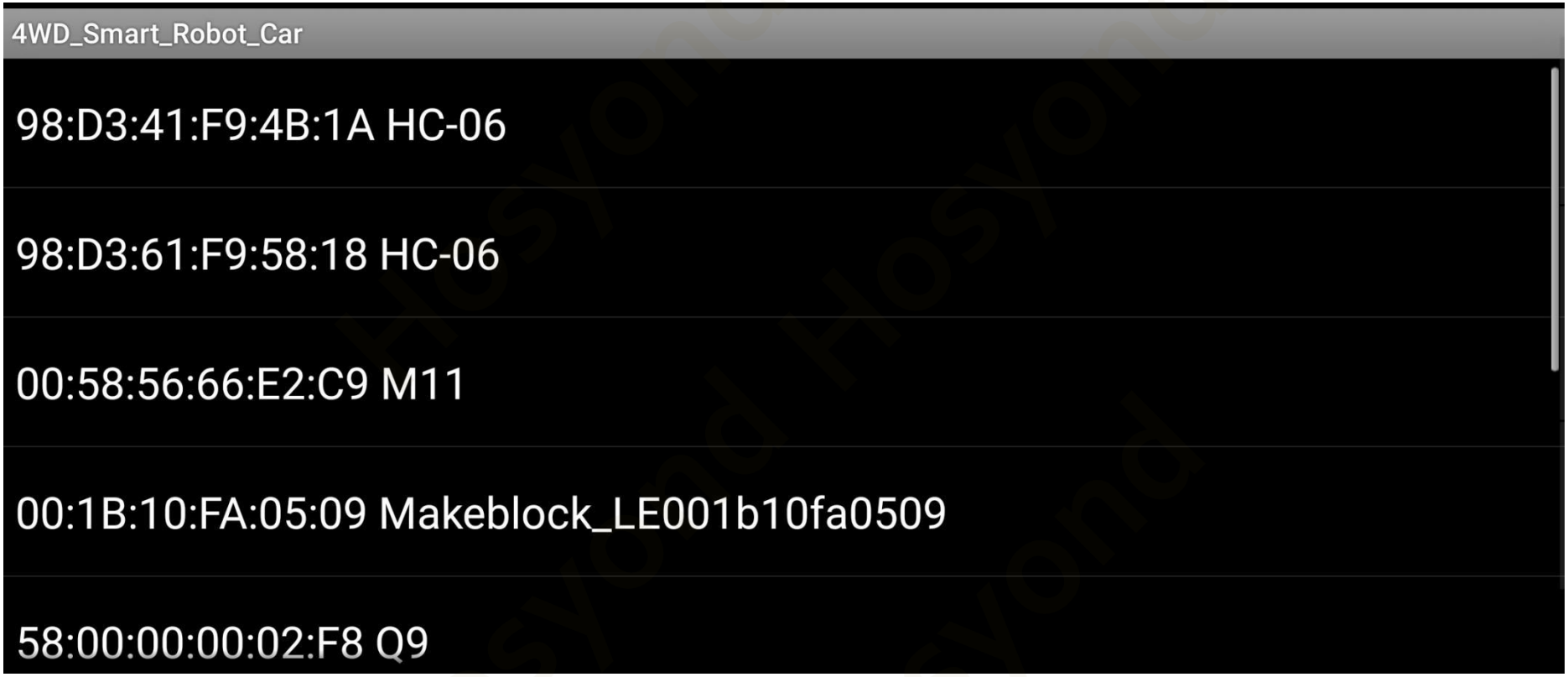
Bluetooth module, you will see a usable device called HC-06 on the PAIRED DEVICES(As shown below).If the HC-06 does not appear on the PAIRED DEVICES, reoperate the above steps.



After the above steps are complete, we open the **4WD_Smart_Robot_Car app**.



Click the Connect Bluetooth icon button"Scan device" . the HC-06 will appear in our scan results.Select HC-06.

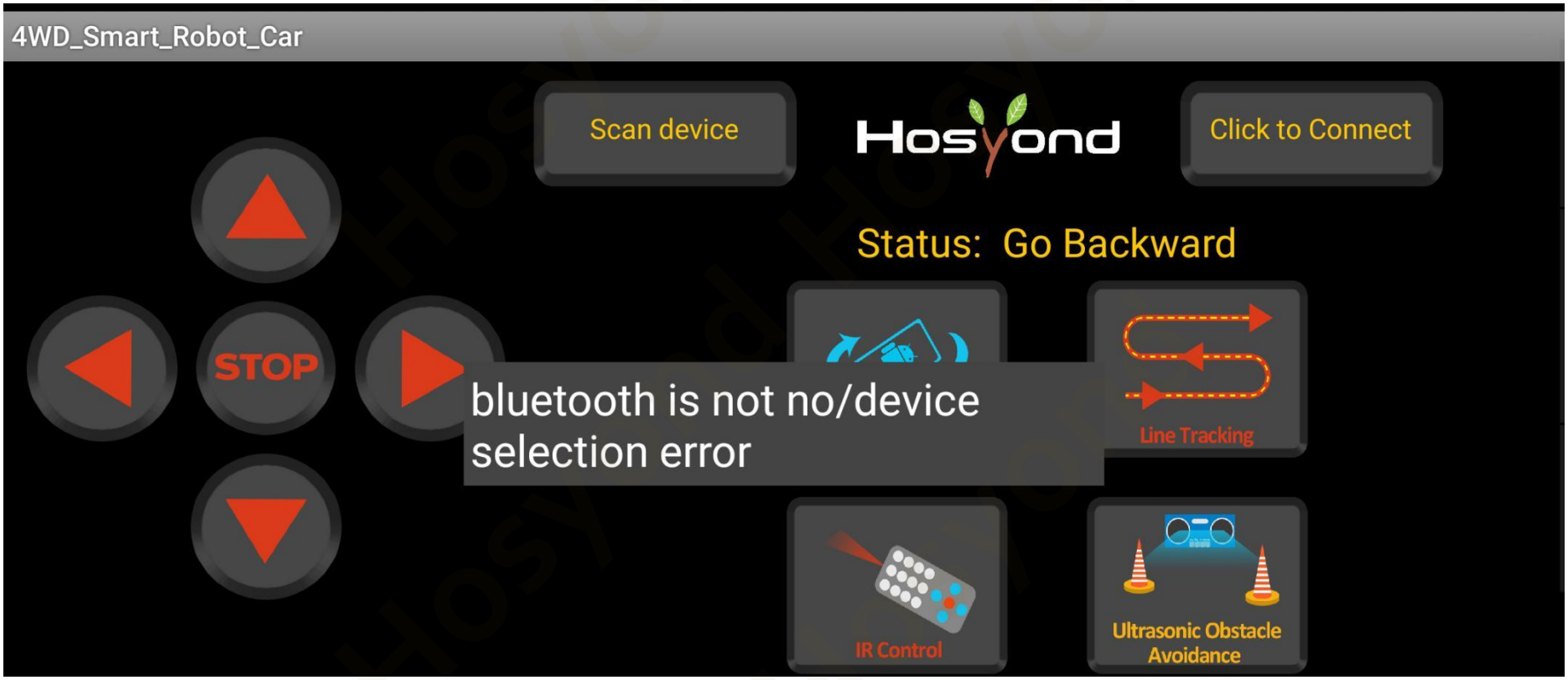


After selecting the hc-06 device, click button **Click to connect** .Note: You must select the correct device before clicking the "**Click to connect**"button

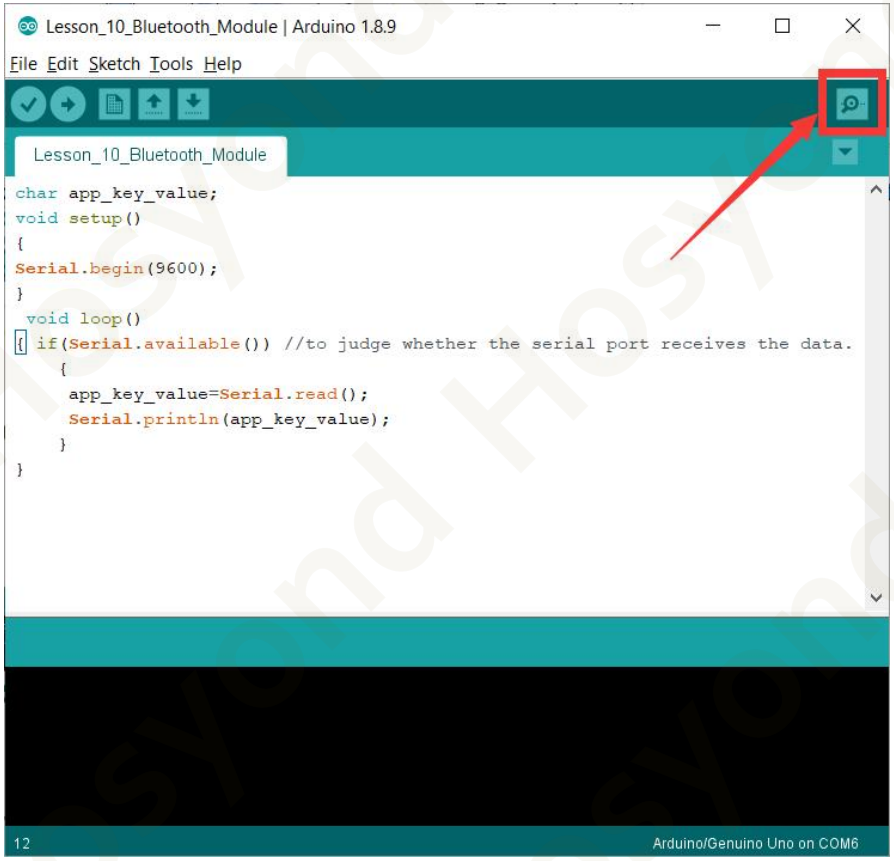
After click on the button **Click to connect** .wait 2 seconds,If the connection is successful, "Connected" will be displayed

(Note: When the Bluetooth module is not successfully connected, the red LED light will continue to flash. When the connection is successful, the red LED light will be remain on.)

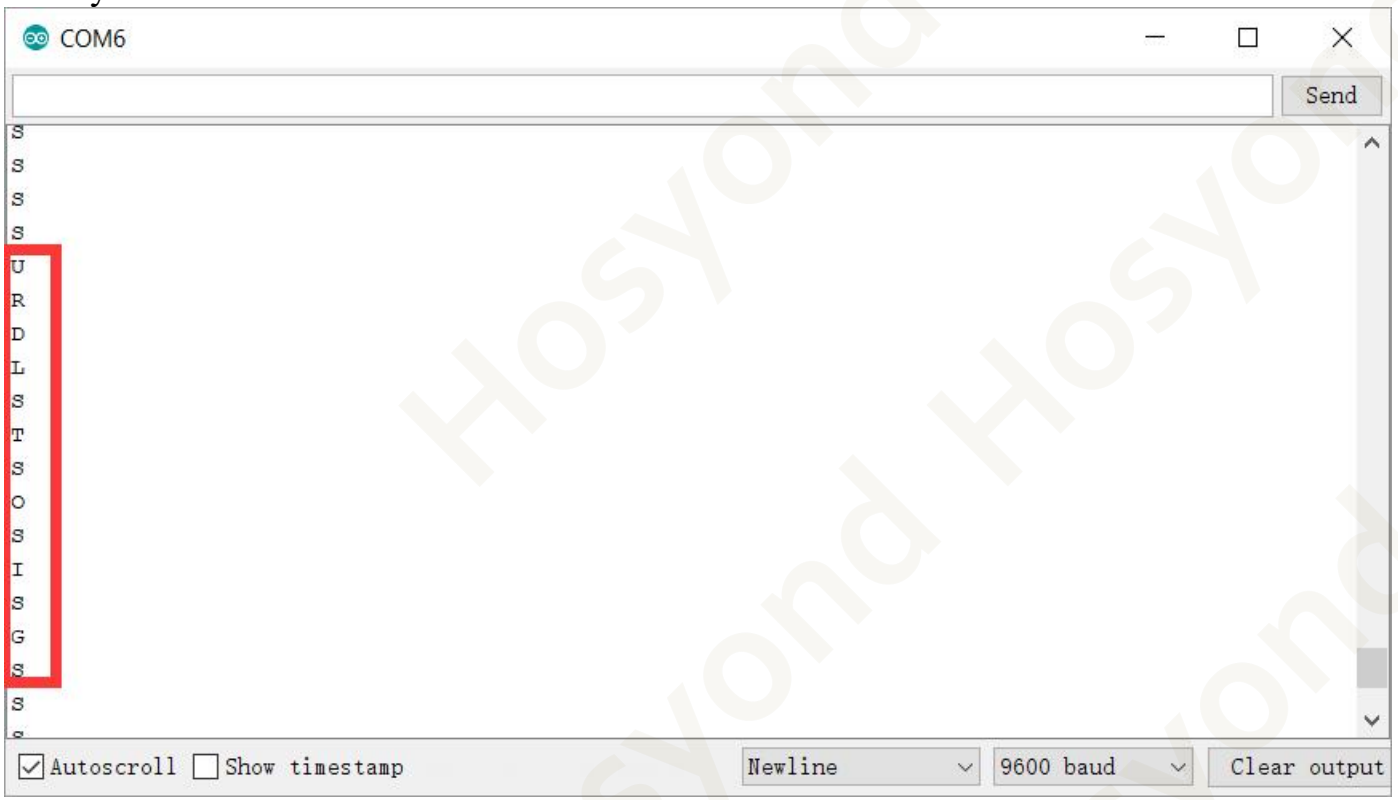
If there is a warning in the interface, it means that Bluetooth is not turned on normally or the selected device is wrong. Follow the steps above to reconnect



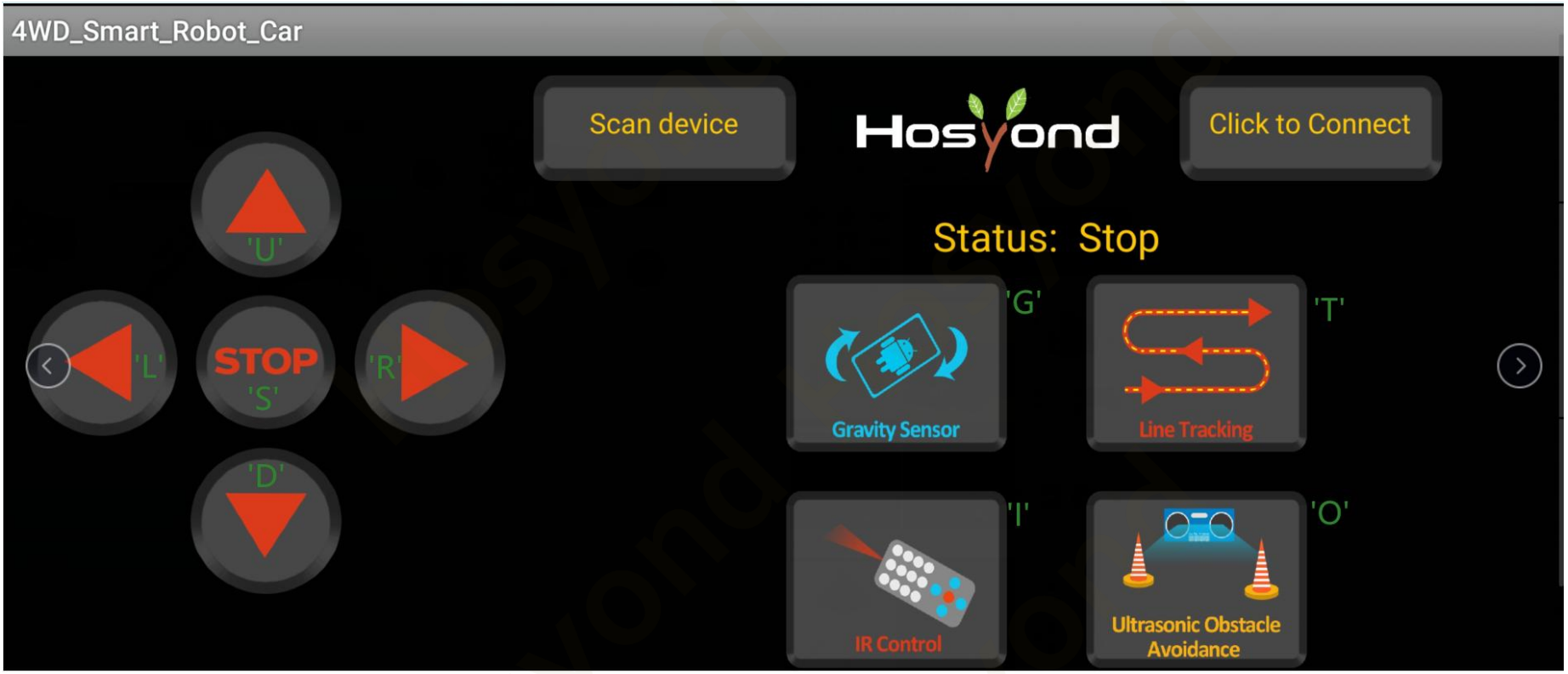
After uploading the code, click the button in the upper right corner to open the serial monitor to view the measured distance



Then you can see the data as blow:



For example,If the Line Tracking mode button is pressed,



the mobile phone Bluetooth sends the character "T" to the Robot car Bluetooth module.

Set the baud rate of the communication between the Bluetooth module and the mobile phone's Bluetooth to 9600.

Lesson 11 L298N Motor Driver

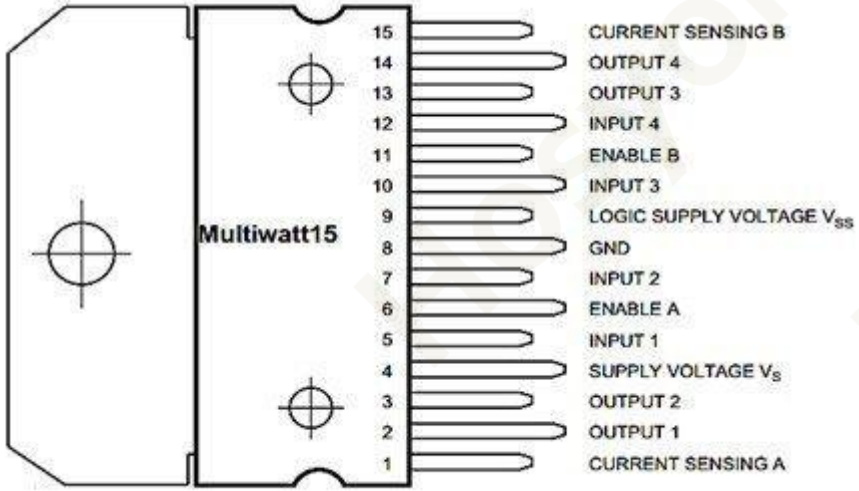
About this lesson:

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.

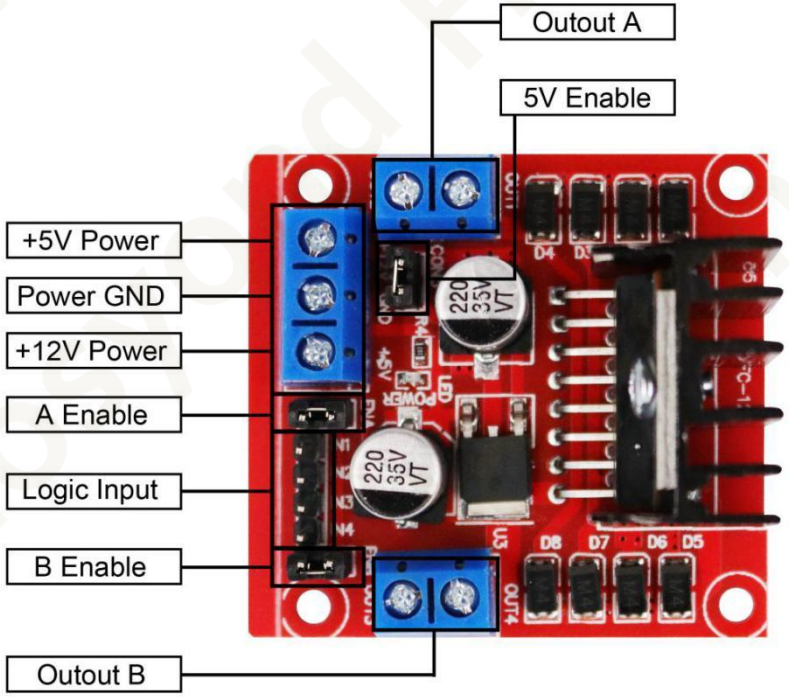
Here is a diagram of the pin outs of an L298N integrated circuit:



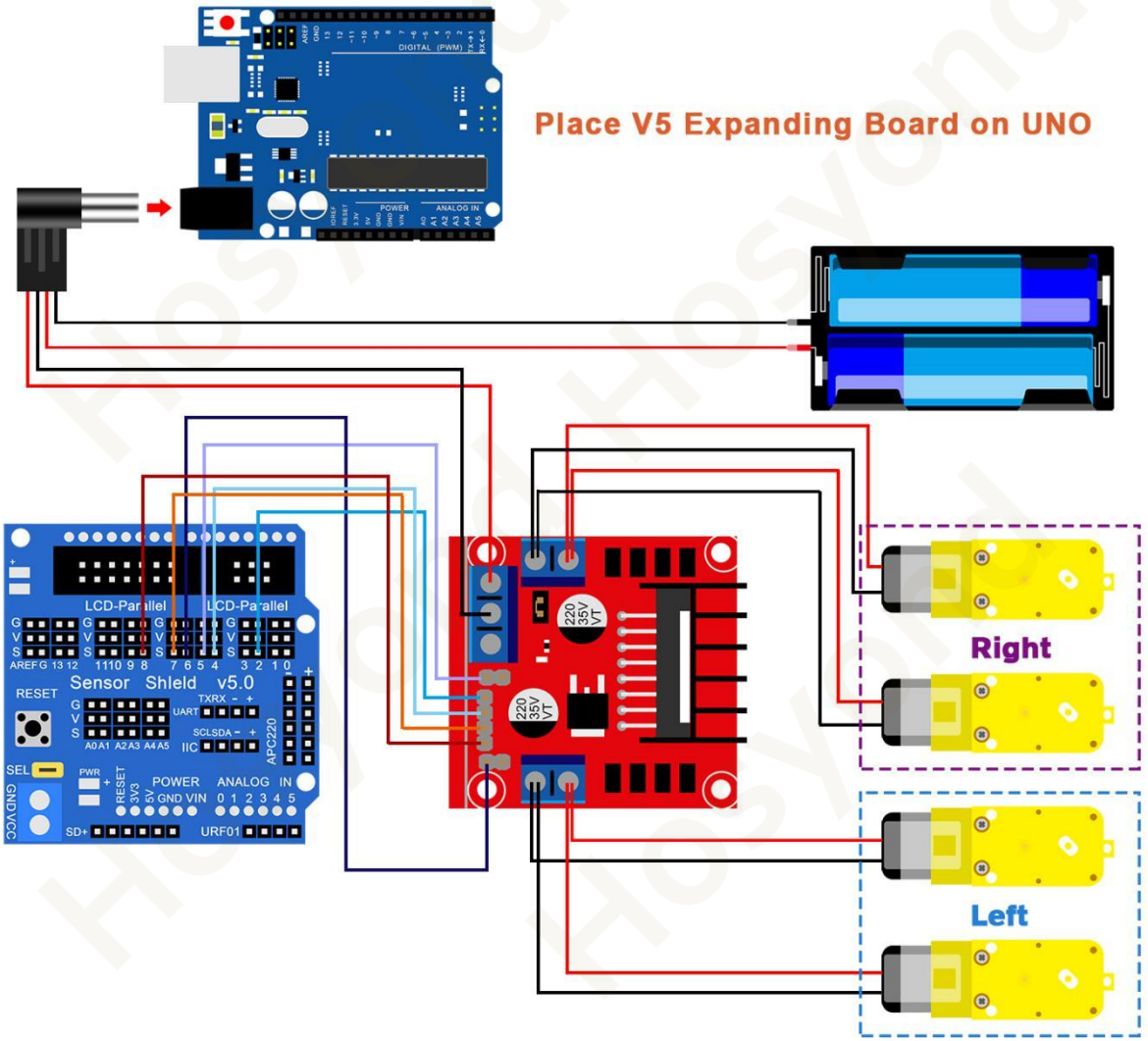
Using L298N made by ST Company as the control chip, the module has characteristics of strong driving ability, low calorific value and strong anti-interference ability.

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

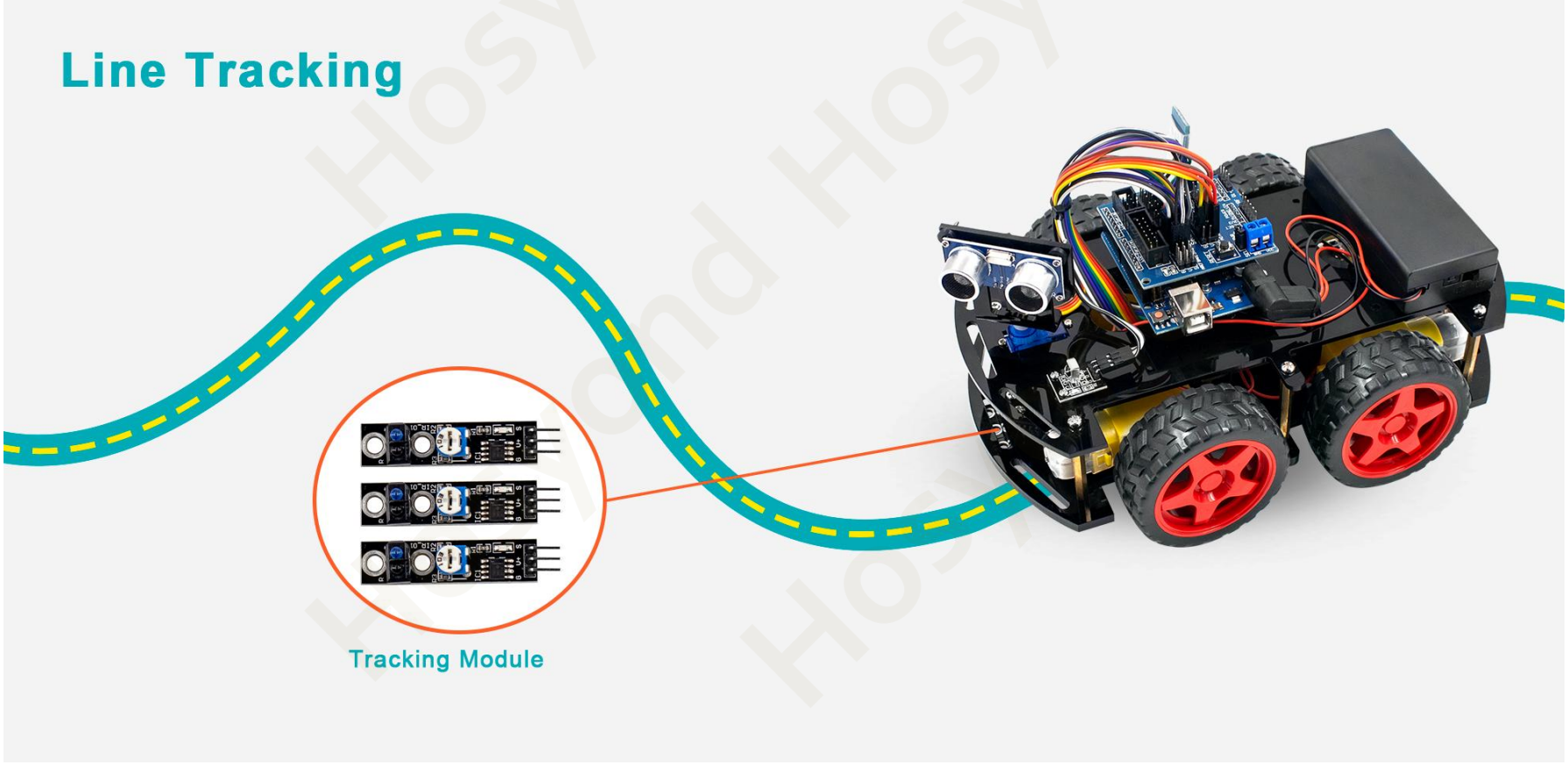
After wiring, please open the program in the code folder- Lesson_11_L298N_Motor_Driver and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

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 200 (PWM value is 200) and then stop for 2second; circulating like this.

Lesson 12 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.

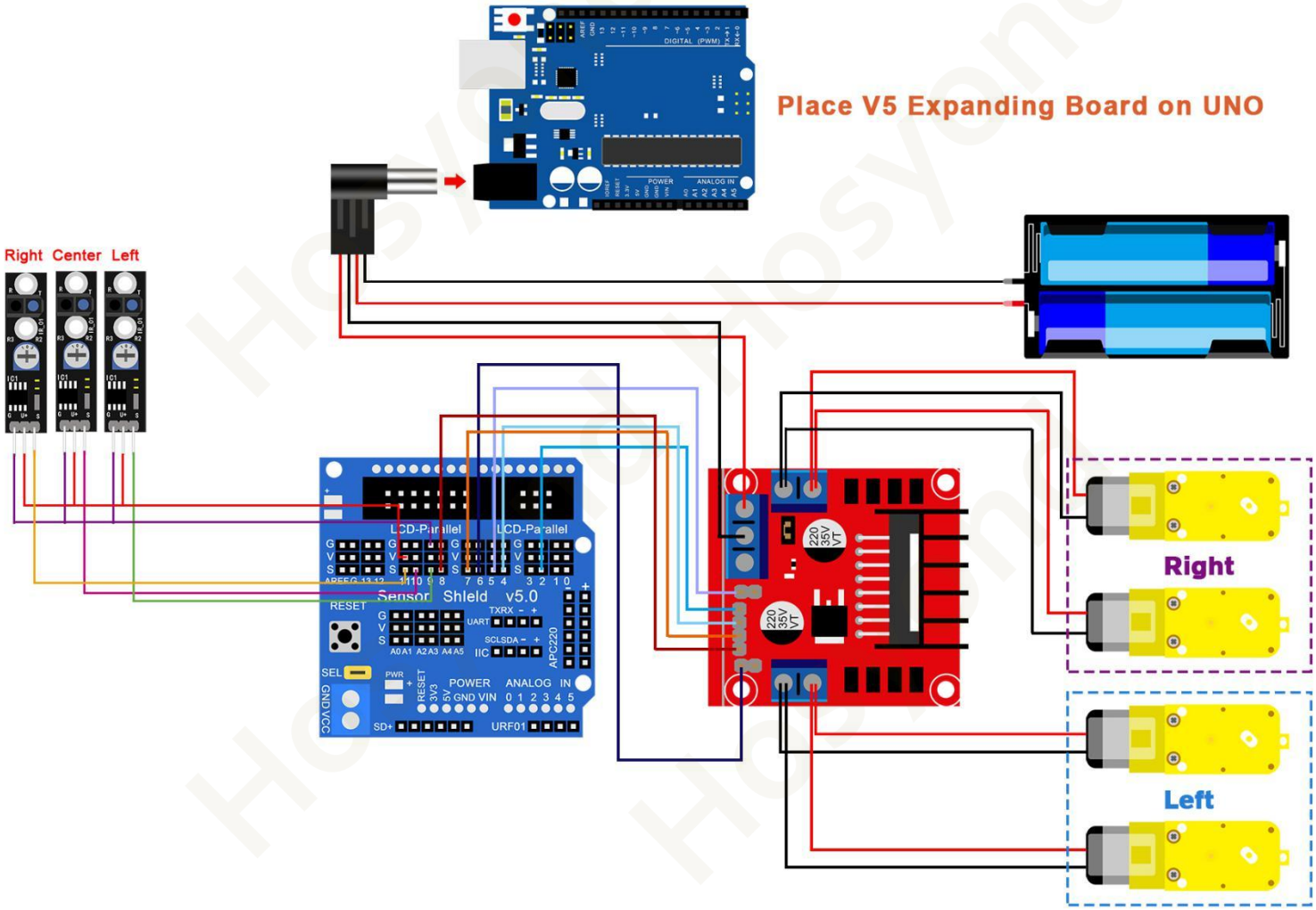
If the car can’t move, please check the following:

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.

Wiring diagram



Code

After wiring, please open the program in the code folder- Lesson_12_Line_Tracking_Car and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

Turn the POWER switch ON.The robot car will move forward along the black track.

Note: In order to make the car better implement the line following function, the width of the black line should be greater than the parallel width of the three line following sensors.

Lesson 13 Ultrasound Obstacle Avoidance Car

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.

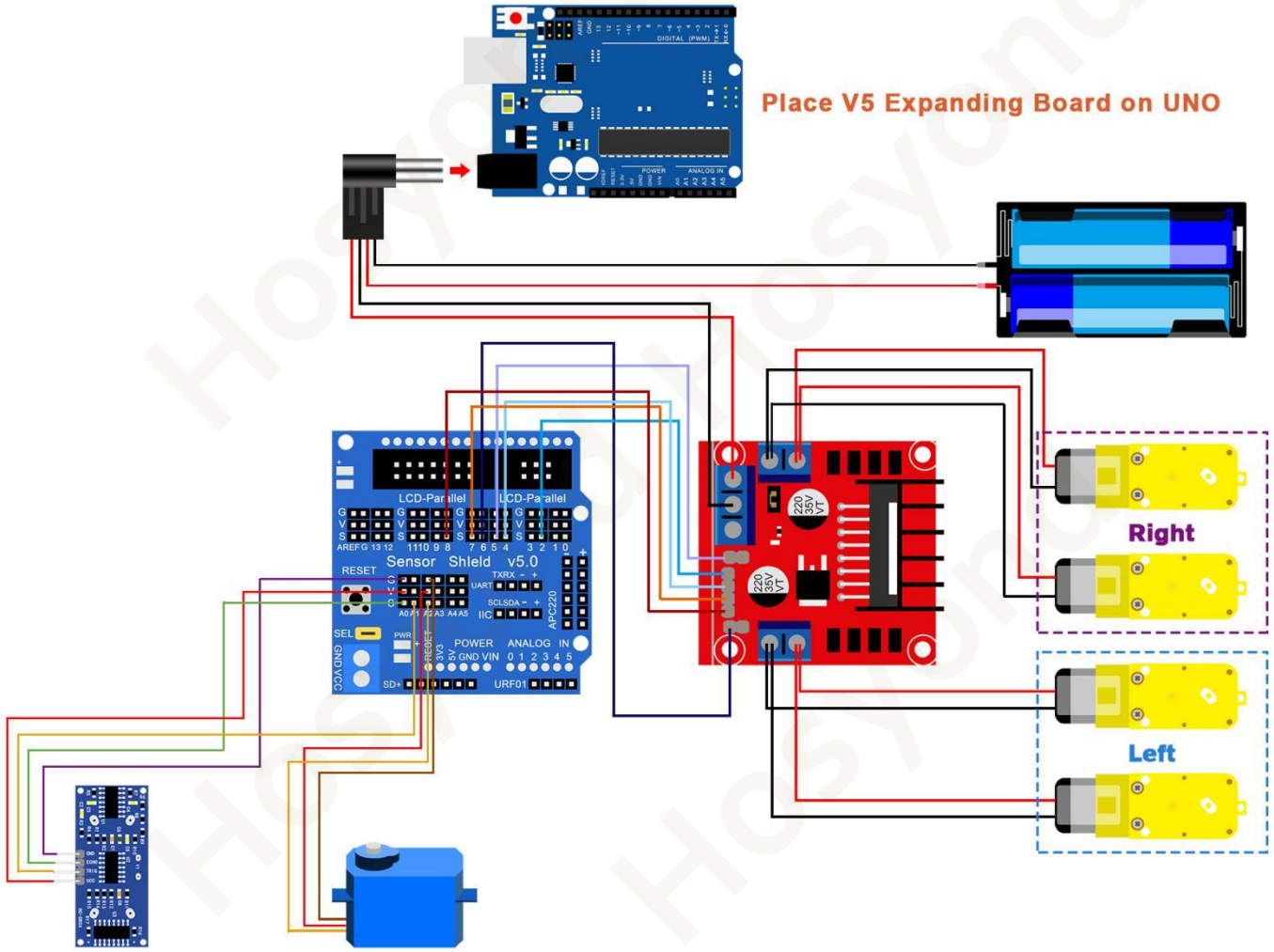
Principle:

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.When there is no obstacle in front of the robot car, the tank keeps walking straight. When the distance of the obstacle in front of the robot car is less than 20cm, the robot car stops, then detects whether there is an obstacle in the left front and right front, and then turns in the opposite direction.

Wiring diagram



Code

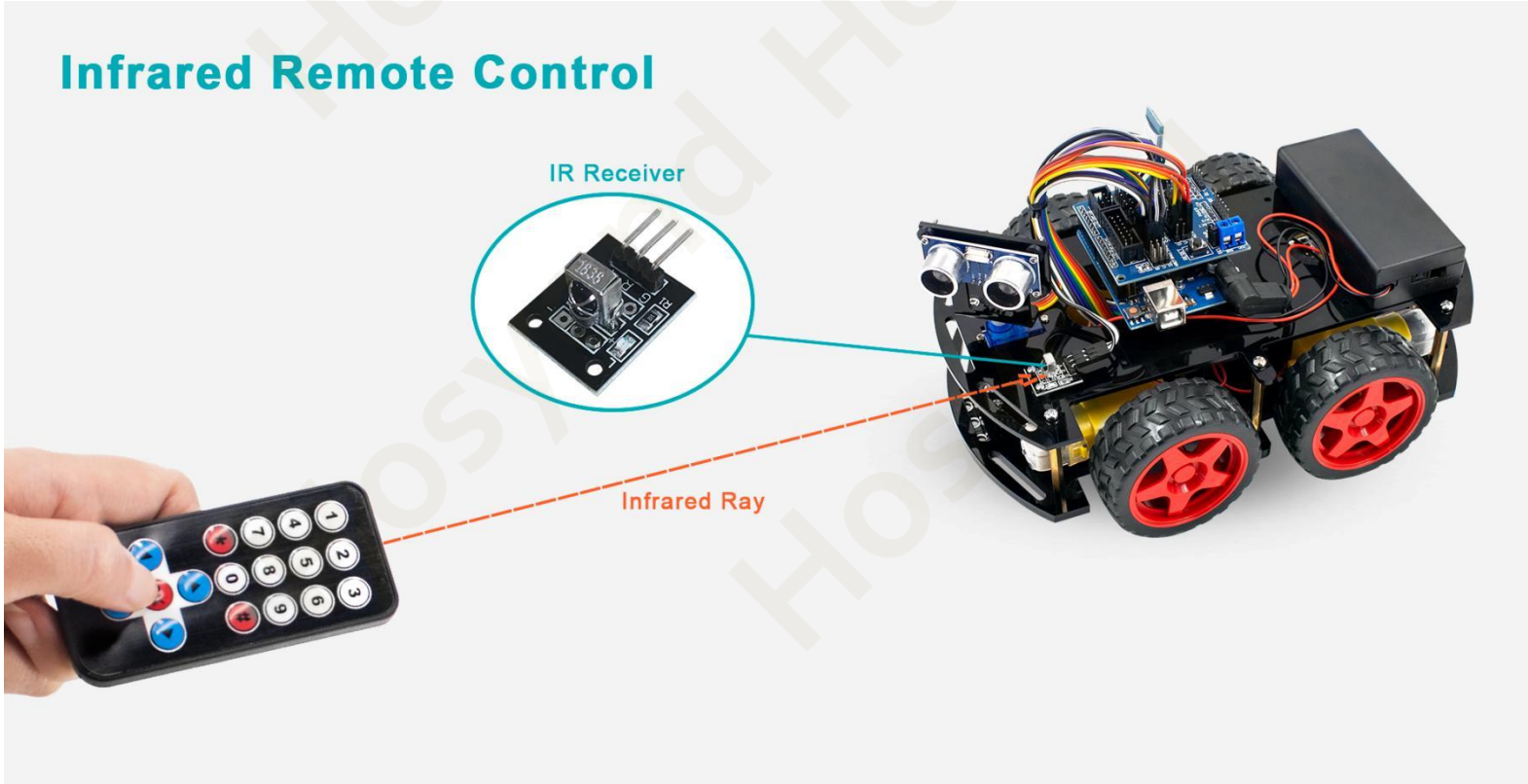
After wiring, please open the program in the code folder- Lesson_13_Ultrasound_Obstacle_Avoidance_Car and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

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 4.

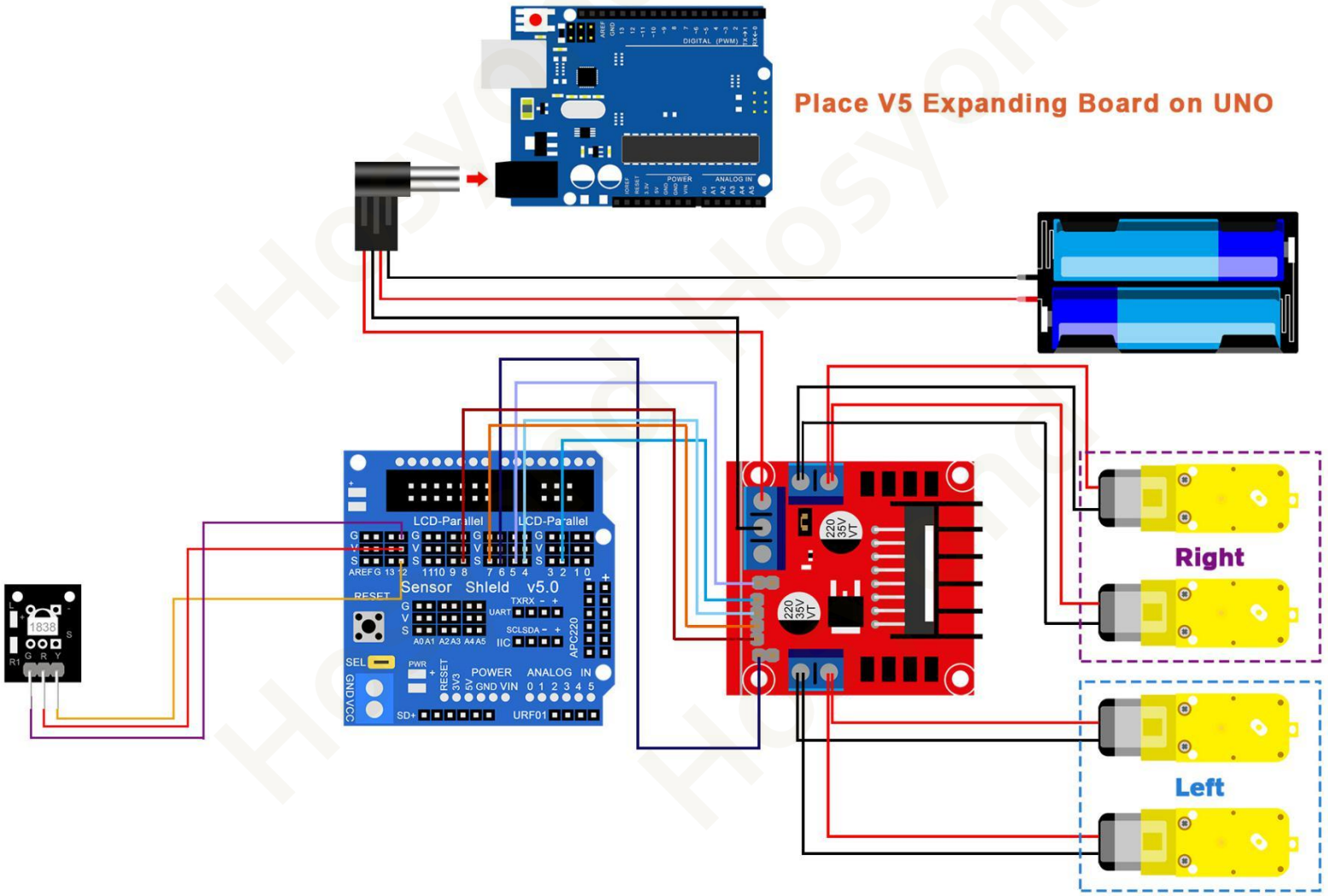
Lesson 14 IR Remote Control Car

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

After wiring, please open the program in the code folder- Lesson_14_IR_Remote_Control_Car and click UPLOAD to upload the program. See Lesson 5 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 4.

How to control the Robot Car

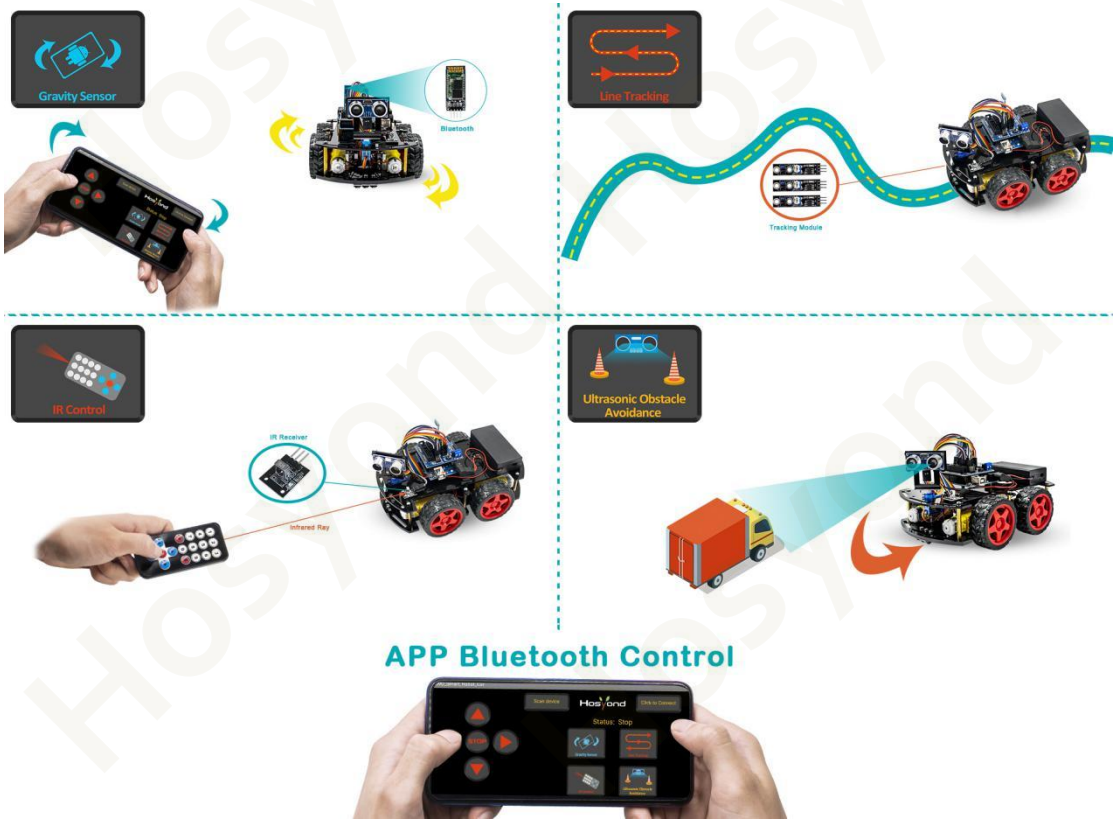
Key on IR remote control					
Robot status	Go forward	Go backward	Turn left	Turn right	Stop

1 FFA25D	2 FF629D	3 FFE21D
4 FF22DD	5 FF02FD	6 FFC23D
7 FFE01F	8 FFA857	9 FF906F
* FF6897	0 FF9867	# FFB04F
	▲ FF18E7	
◀ FF10EF	OK FF38C7	▶ FF5AA5
	▼ FF4AB5	

Lesson 15 Bluetooth Multifunctional Robot Car

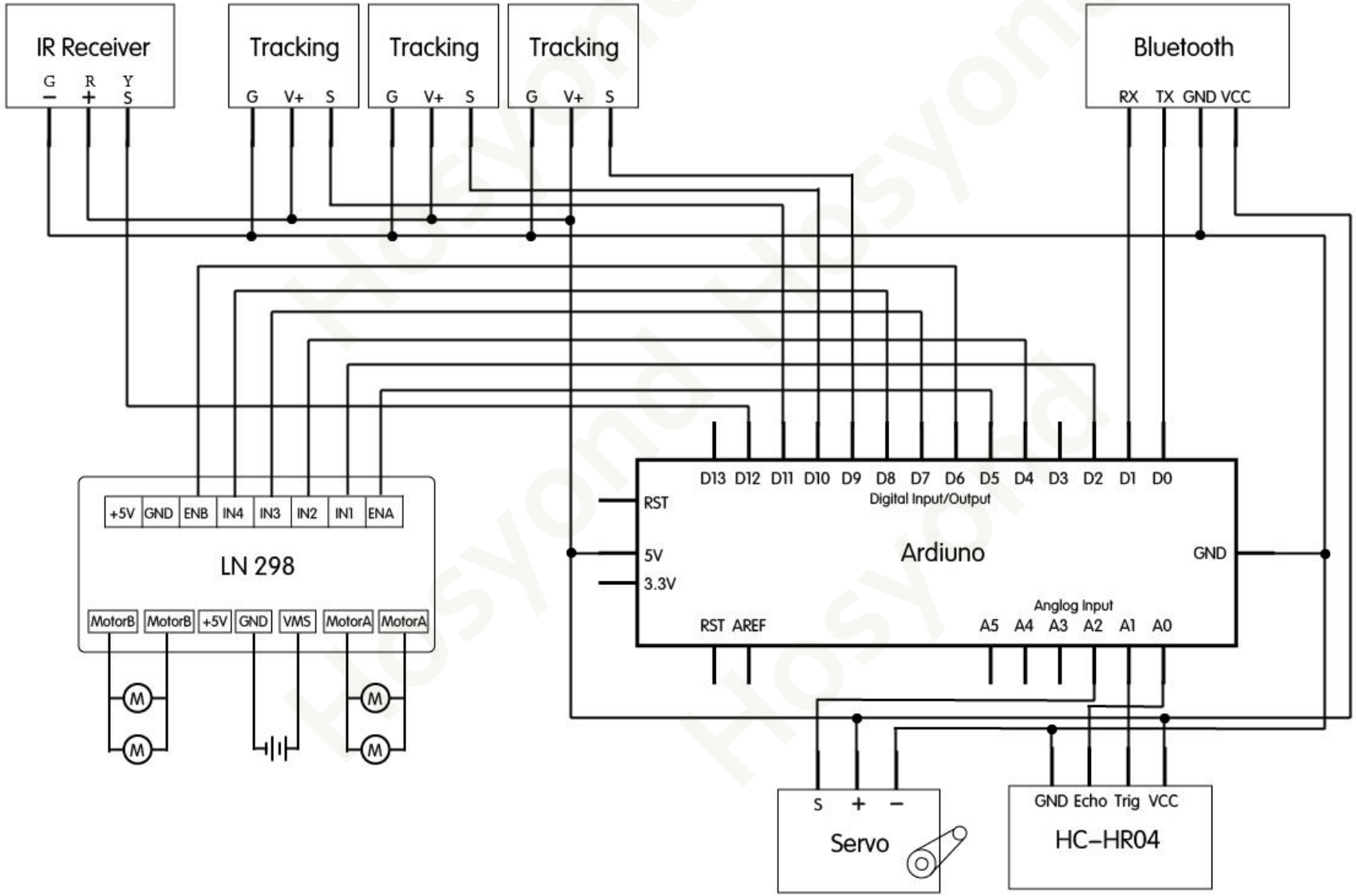
About this lesson:

In this project, we will put four functions, namely line tracking, obstacle avoidance, Gravity sensor control,Bluetooth and IR remote control, together into one to realize the working mode of the car.

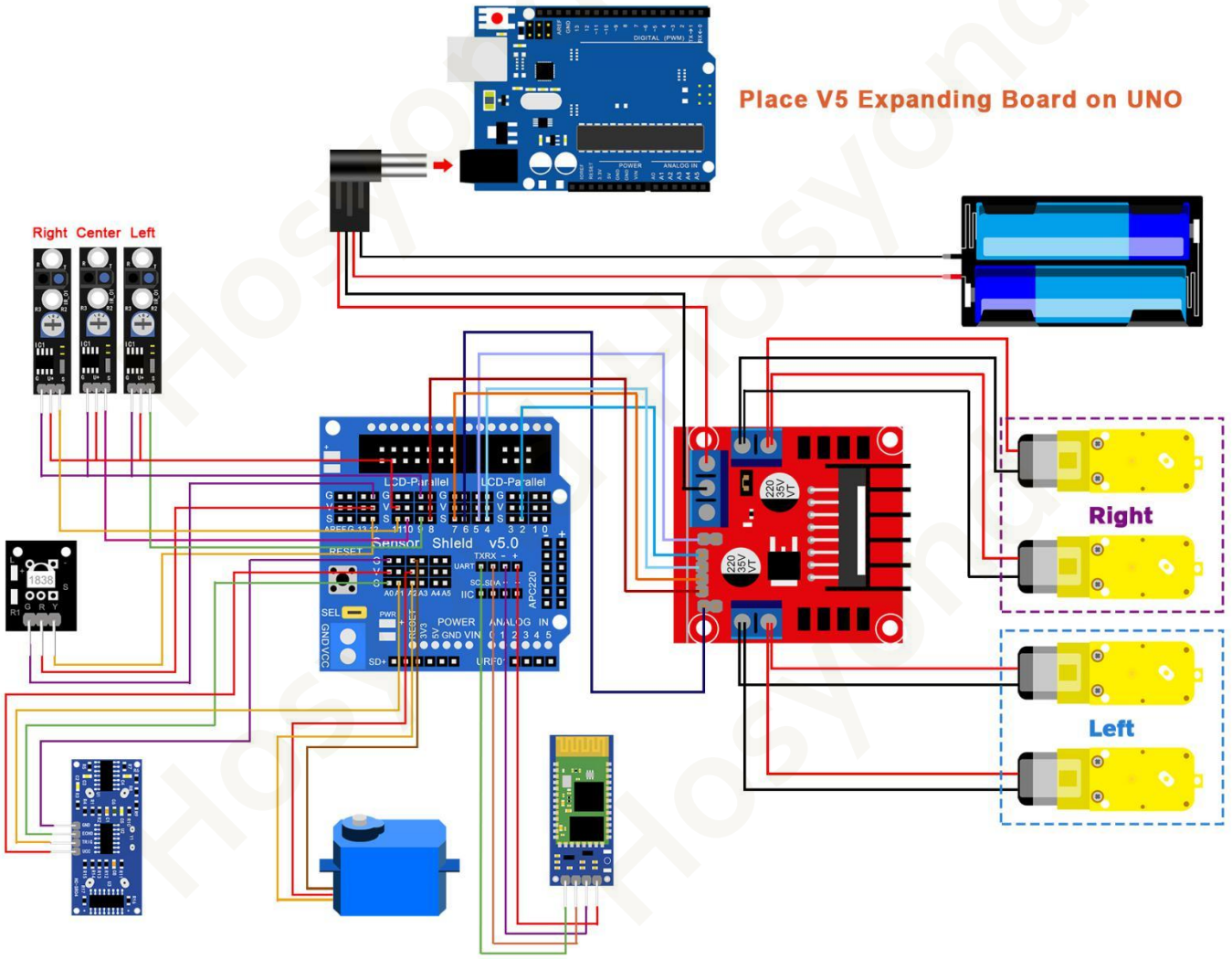


Connection

Schematic



Wiring diagram



Code

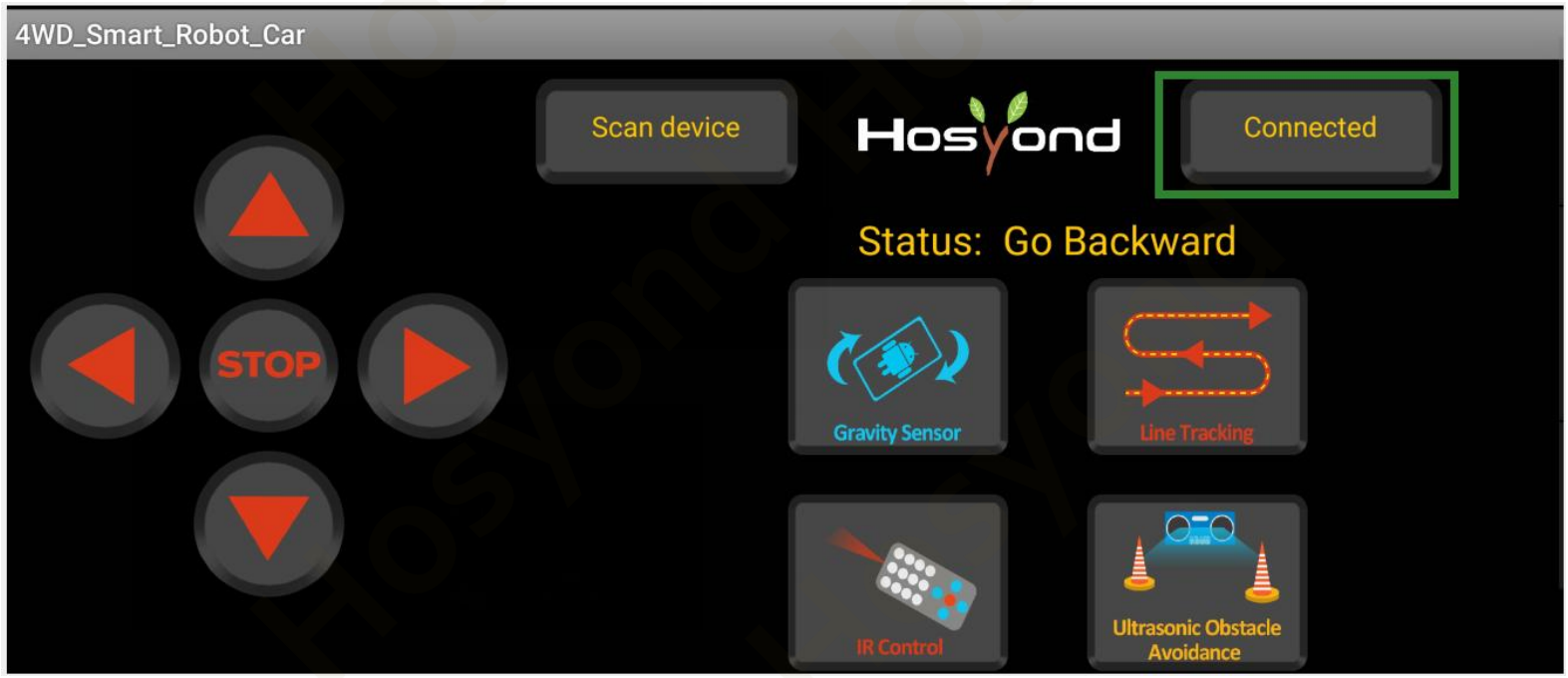
After wiring, please open the program in the code folder- Lesson_15_Multifunctional_Car and click UPLOAD to upload the program. See Lesson 5 for details about program uploading if there are any errors.

Attention:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.When uploading the code, CANNOT connect the Bluetooth module first; otherwise uploading fails! You are supposed to upload the code to control board, then connect the Bluetooth module.

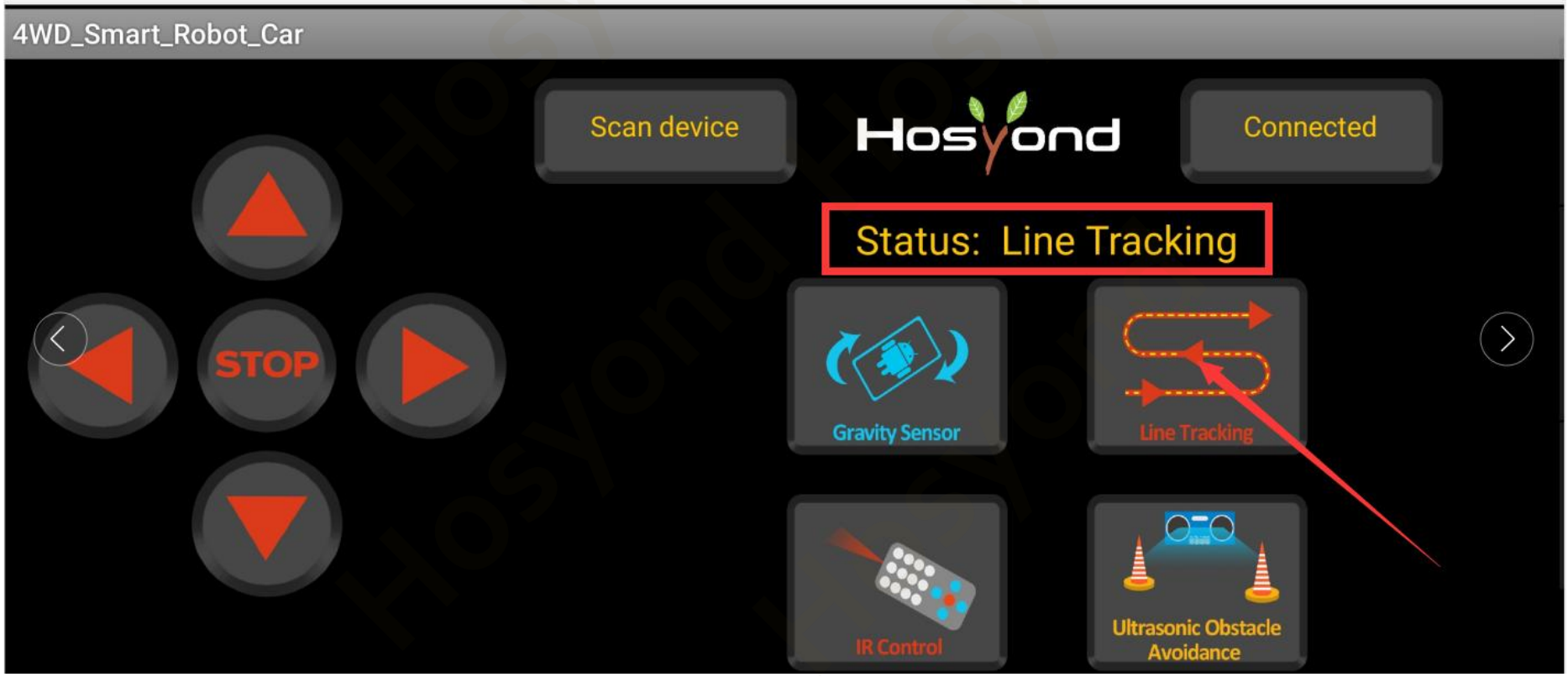
How to use the app to control the robot car

After completing the program upload, plug in the Bluetooth module again, and then open the app software to complete the connection of the Bluetooth device (refer to Lesson 10 Bluetooth Module for specific steps).

The interface after the app software and Bluetooth module are successfully connected is as follows.

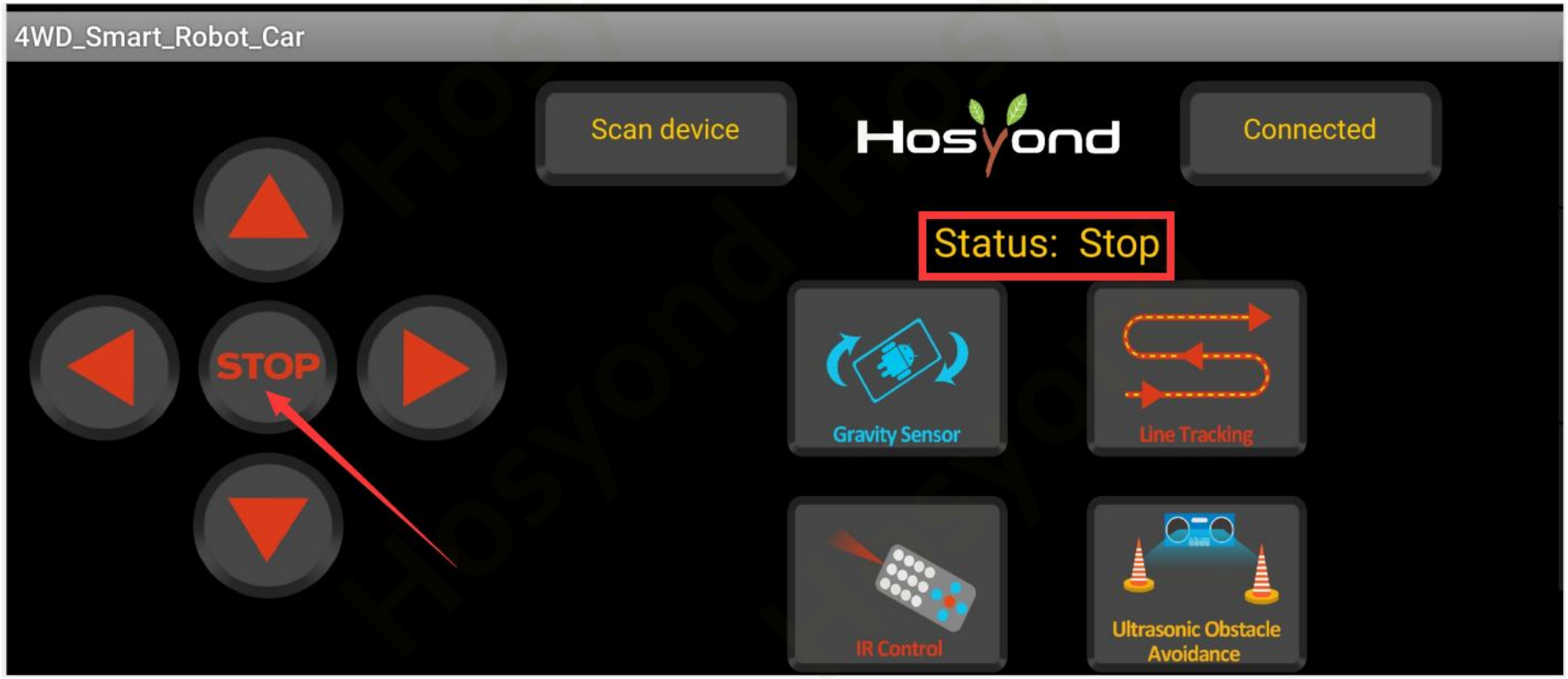


For example, if you want to control the car to enter the Line tracking mode, you click the line patrol function button, and then the status bar in the app interface will display: status: Line Tracking.After the robot car receives the signal, it will enter the line patrol state. At this time, you need to provide a black track for the robot car.



Note: If you want to switch to other modes, you need to click the STOP button first, and then click the function button you want to switch. It means that you need to be in status: stop to switch to other modes.

Step 1:



Step 2：：


4WD_Smart_Robot_Car
Scan device
Hosyond
Connected
Status: IR Control
STOP
Gravity Sensor
Line Tracking
IR Control
Ultrasonic Obstacle Avoidance

The app can control the robot car to complete all functions, and can switch freely. The functions you can choose are Line Tracking Mode, Gravity Sense Control Mode, IR Remote Control Mode, Ultrasonic Obstacle Avoidance Mode.