Lesson 1 - Make The Car Move

Points of this section

Learning part：

- Learn how to use Arduino IDE
- Make the car move by uploading program

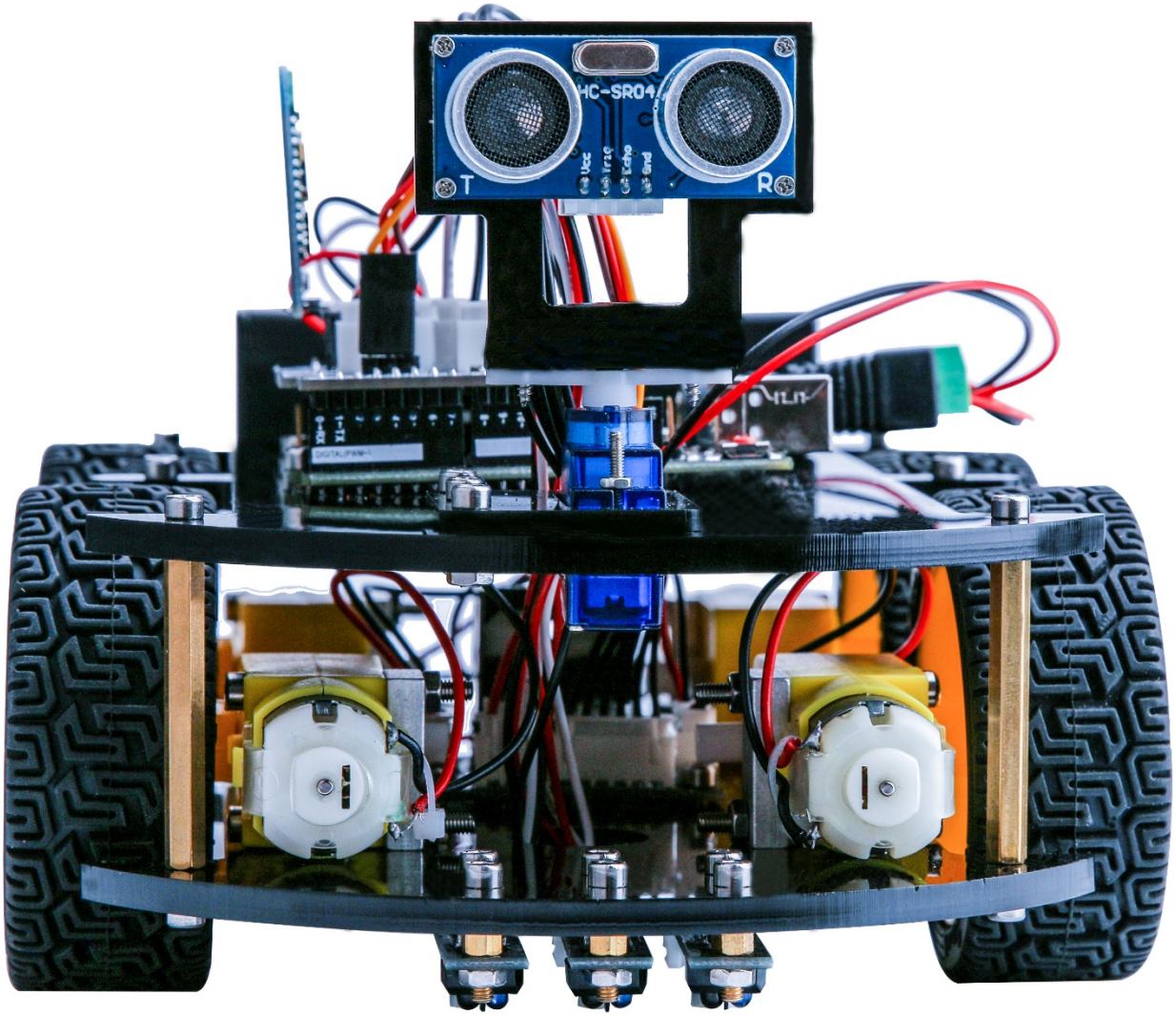
Preparations:

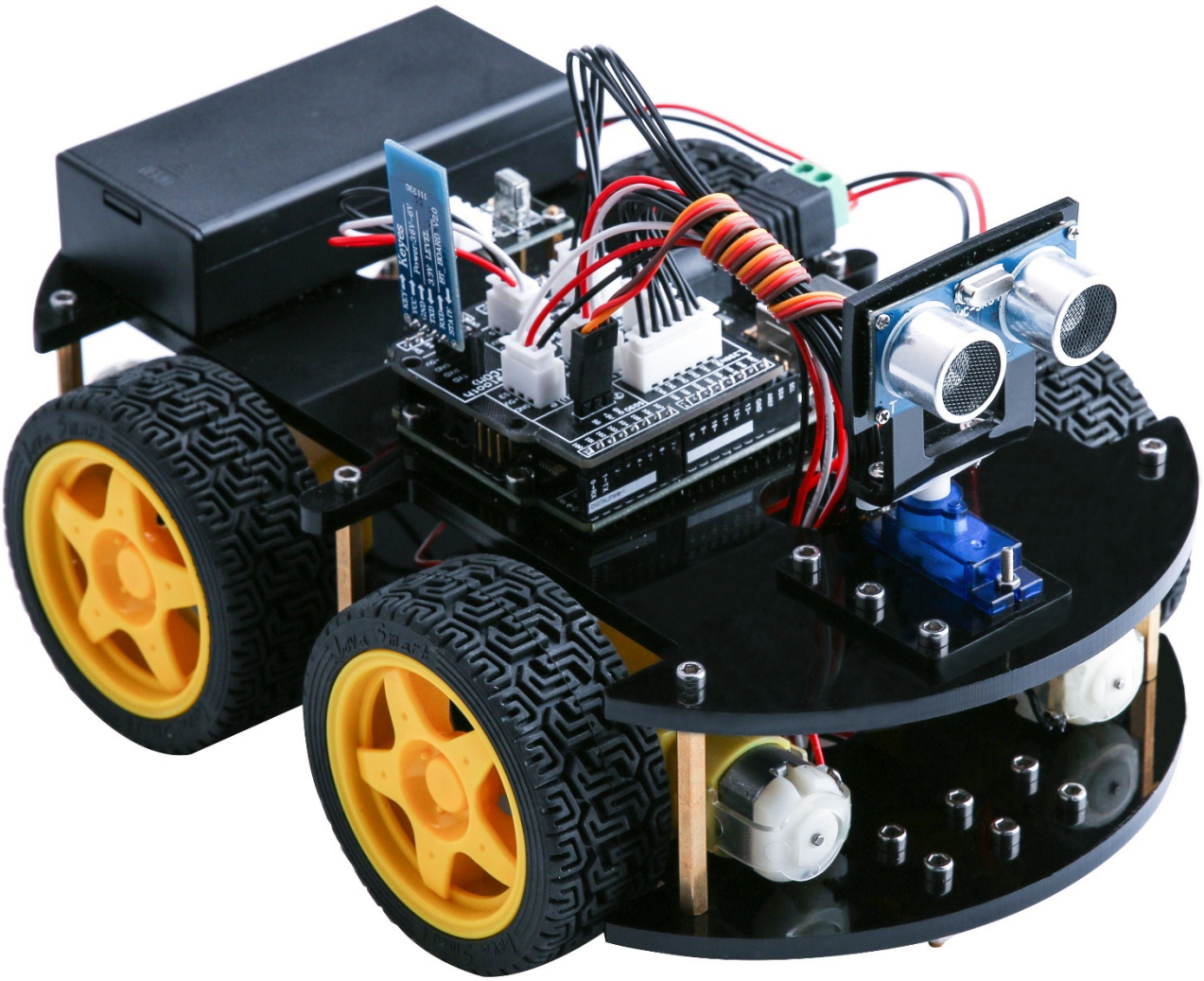
- One car (with a battery)
- One USB cable

Ⅰ. Introduction of the car

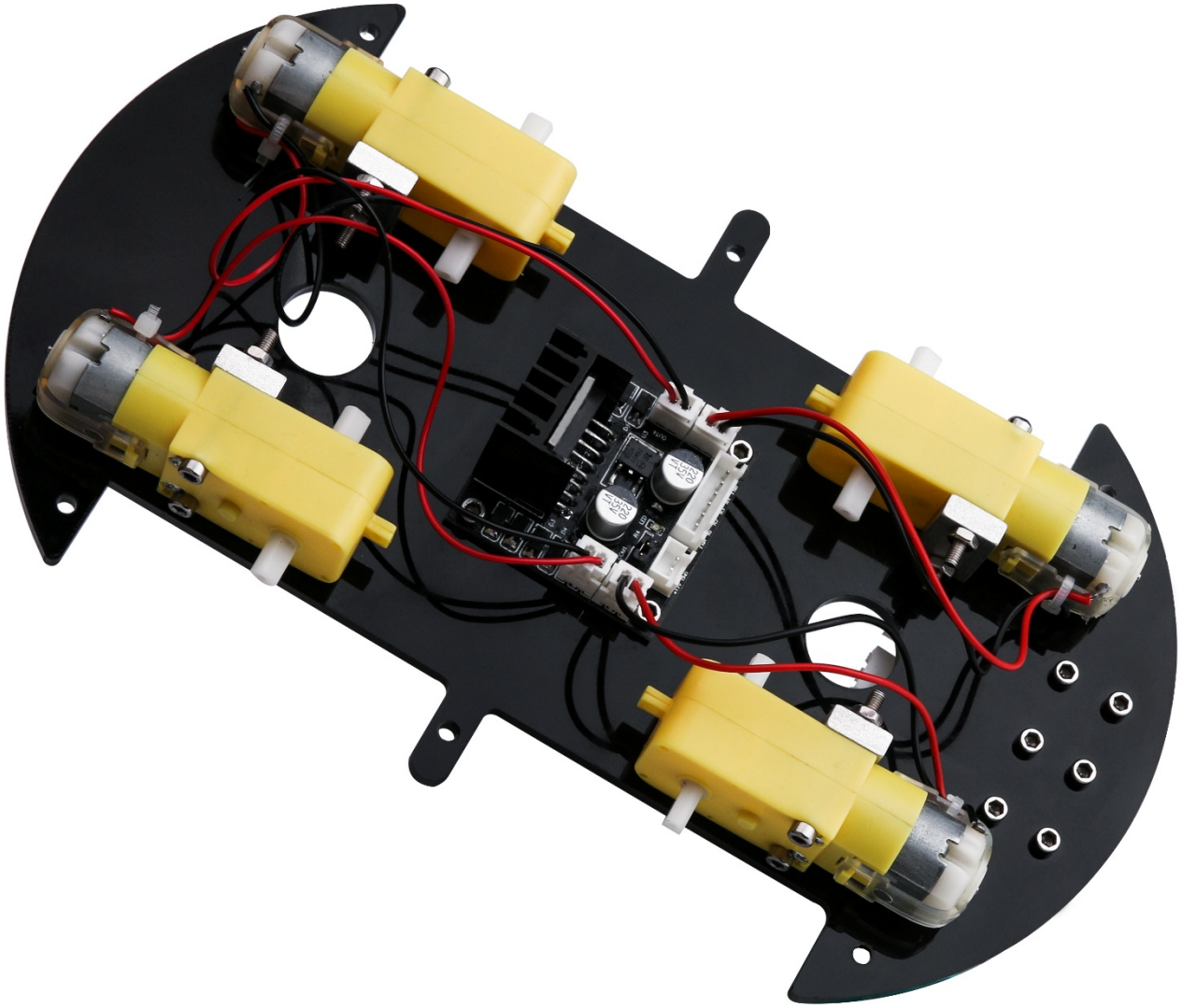
This kit is an extremely flexible vehicular kit particularly designed for education, competition and entertainment purposes. The upper panel of the kit is directly compatible with 9-gram steering engine. It also carries supersonic sensor, battery and other fixed holes to facilitate installation of various sensors. This is a very funny and versatile robot that meets learning and production purposes. With it, you can implement diverse interesting ideas, such as Bluetooth and infrared remote control, automatic avoidance of obstacles, and line inspection.

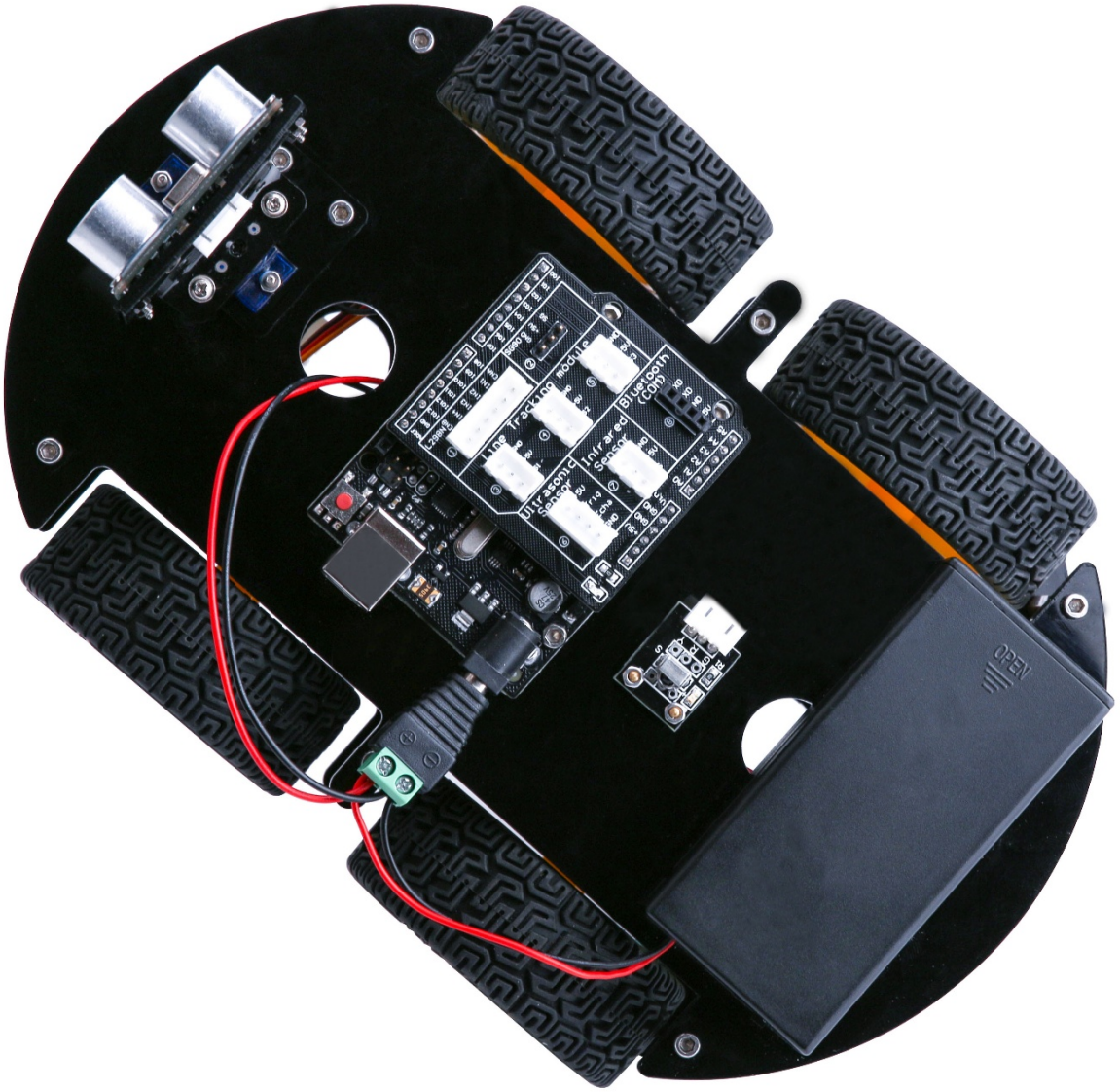
Let’s describe the small vehicle that will accompany us for a long time in the future.


HC-SR04
T
R



Each parts of the car is as below:




Line tracking module
Bluetooth (COM)
Infrared Sensor
Ultrasonic Sensor
OPEN

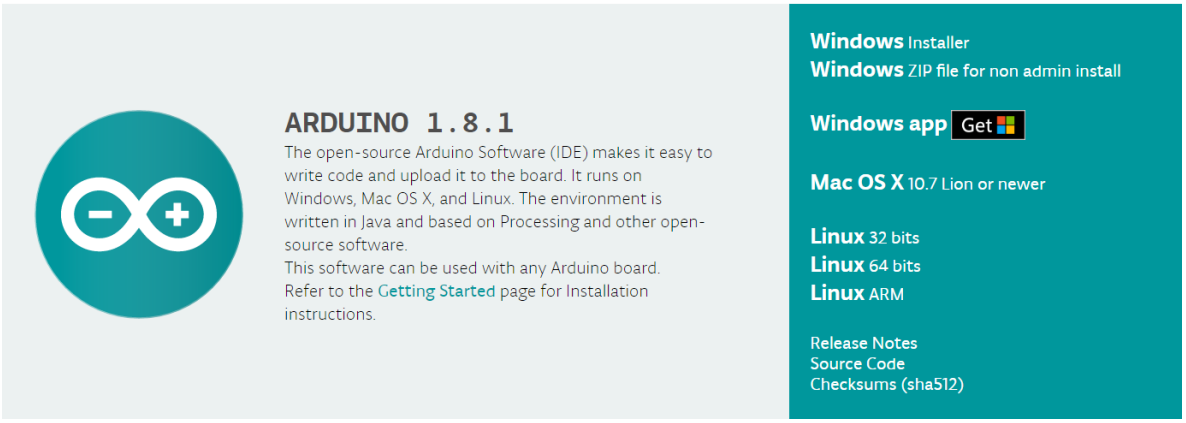
Function of each part：

1. Battery holder with a switch: provide power supply for the vehicle
2. Electric motor + wheel: drive the vehicle to move
3. acrylic plate: the frame of the car
4. L298N motor driving board: drive the motor to rotate
5. UNO controller board: the brain of the car, controls all the parts
6. V5 sensor expansion board: combined with the UNO, make the connection become more easier
7. Servo and cloud platform: enable the GP2Y0A21 distance sensor to rotate 180 degrees
8. Ultrasonic sensor module: distance measurement and obstacle avoidance
9. Line tracking module: black and white sensor for recognition of the white and black lanes
10. Infrared receiver and remote control: provide the infrared remote control function
11. Bluetooth module: provide the Bluetooth control function

Ⅱ. Upload program

Each movement of the vehicle is controlled by the program so it's necessary to get the program installed and set up correctly.

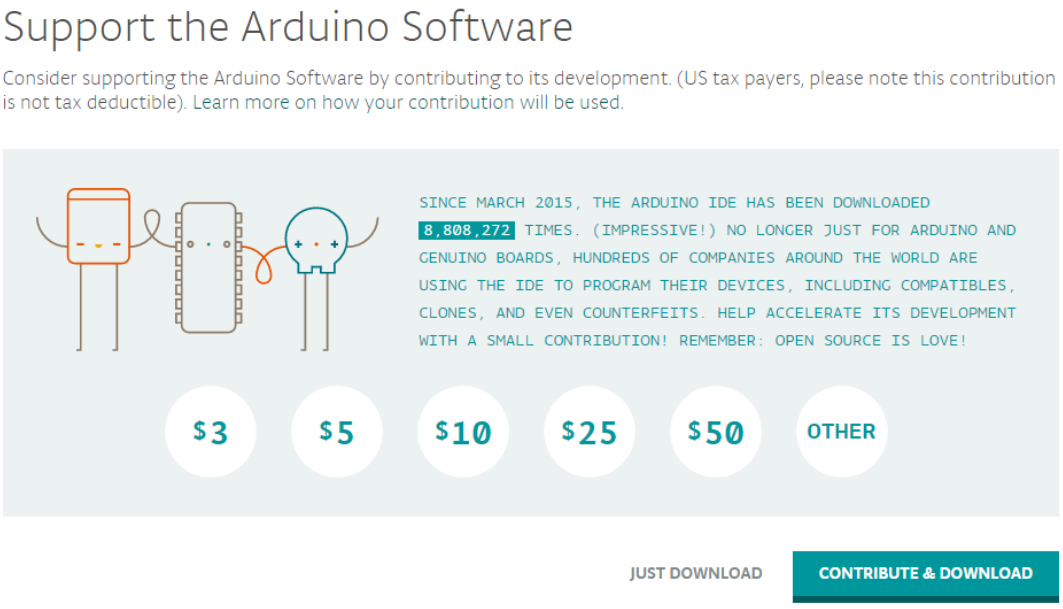
STEP 1: Go to https://www.arduino.cc/en/Main/Software and find below page.



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

STEP2：Download the development software that is suited for the operating system of your computer. Take Windows as an example here.

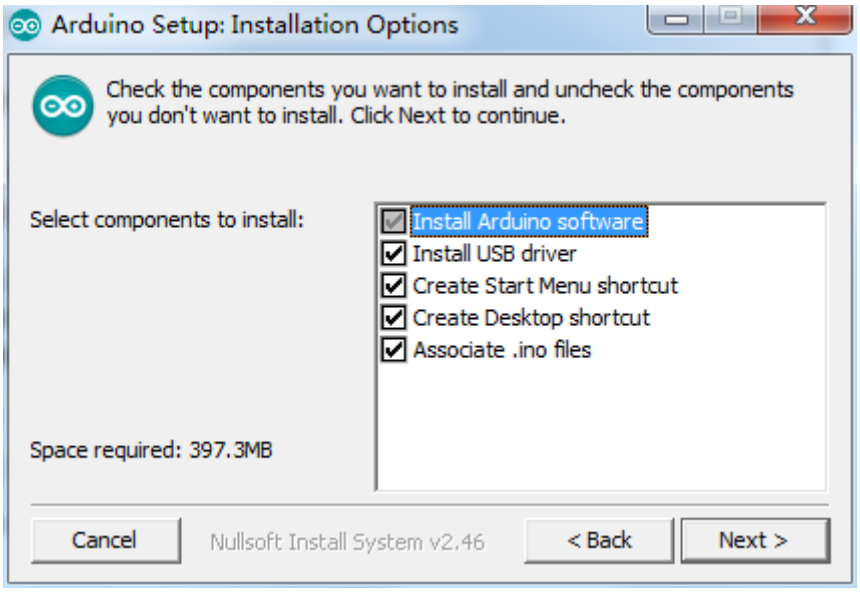
You can install it using the EXE installation package or the green package.



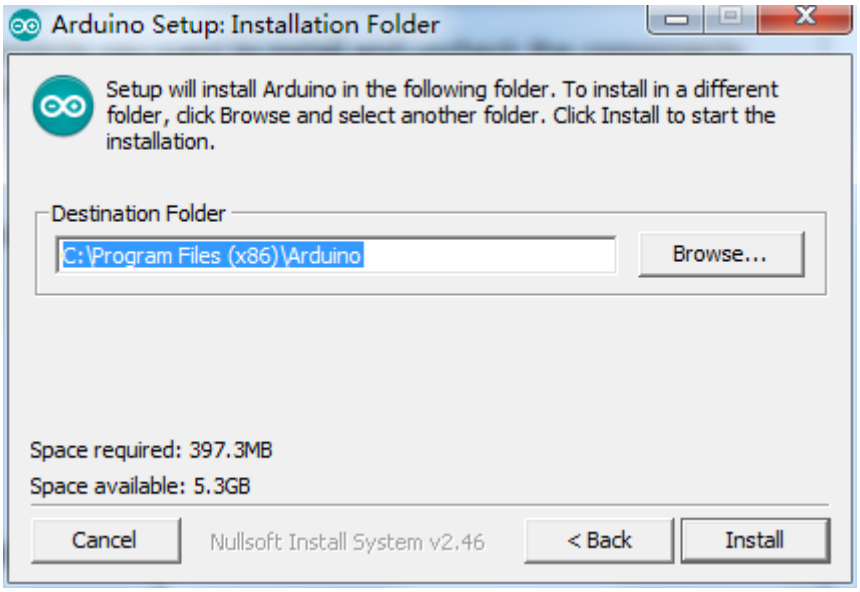
Press the button "JUST DOWNLOAD" to download the software.

These are available in the materials we provide, and the versions of our materials are the latest versions when this course was made.

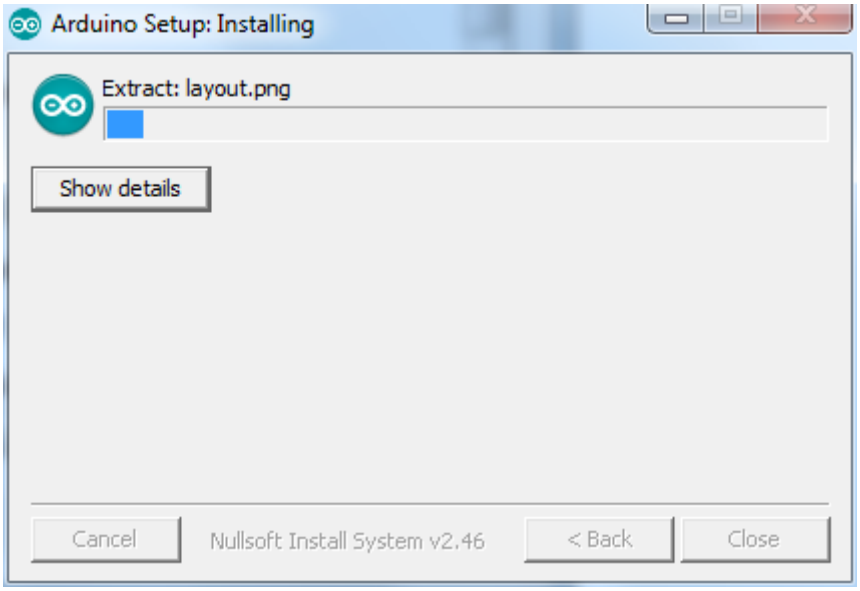
Choose I Agree to see the following interface



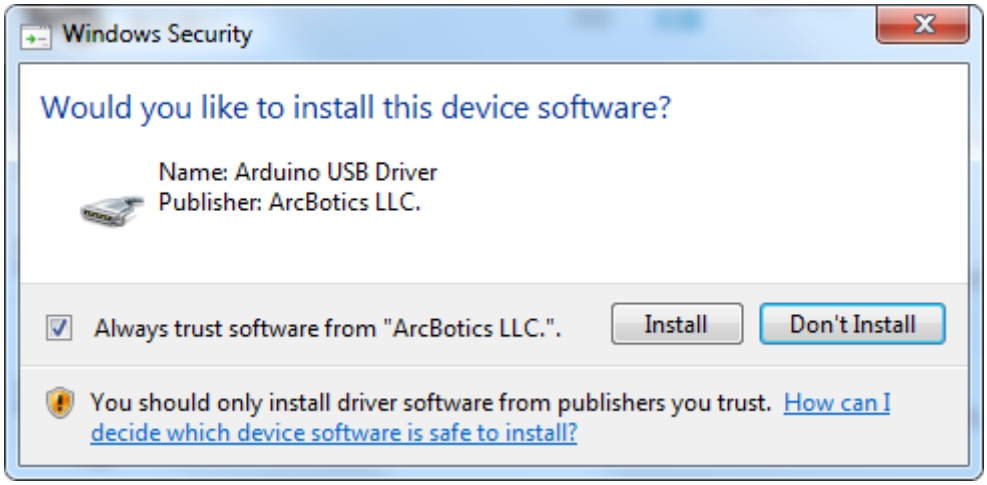
Choose Next



Press Install to initiate installation

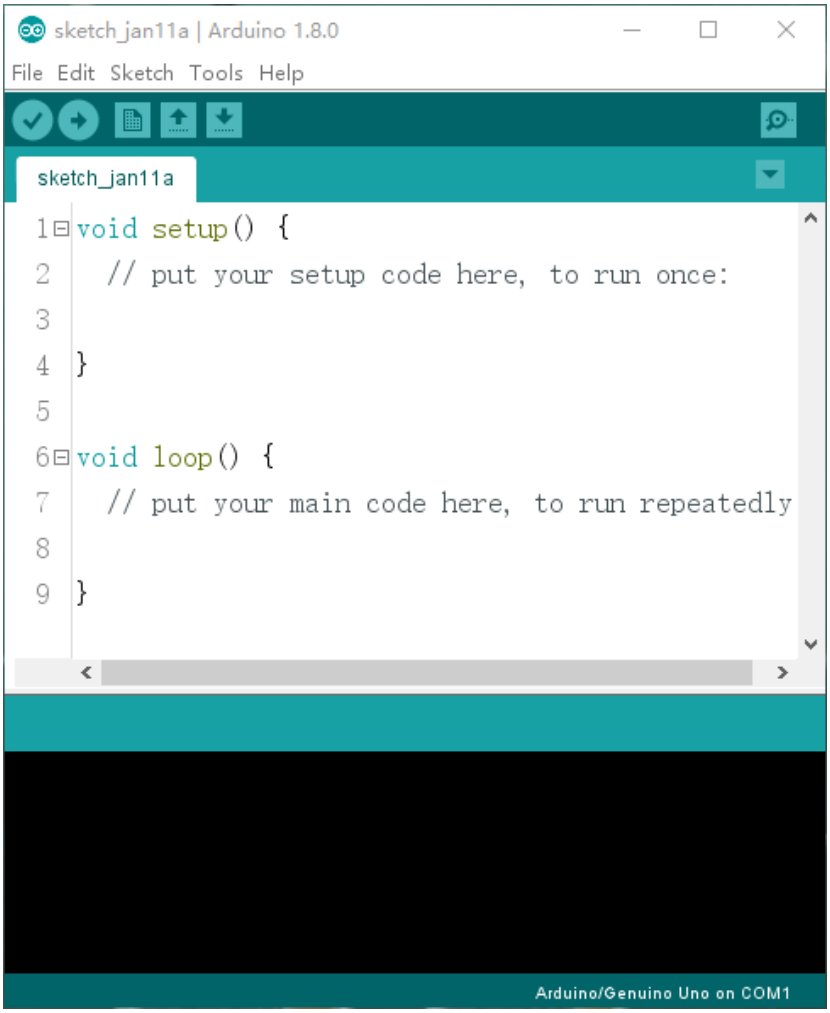


Finally, the following interface appears, you should choose Install to ensure correctness of development

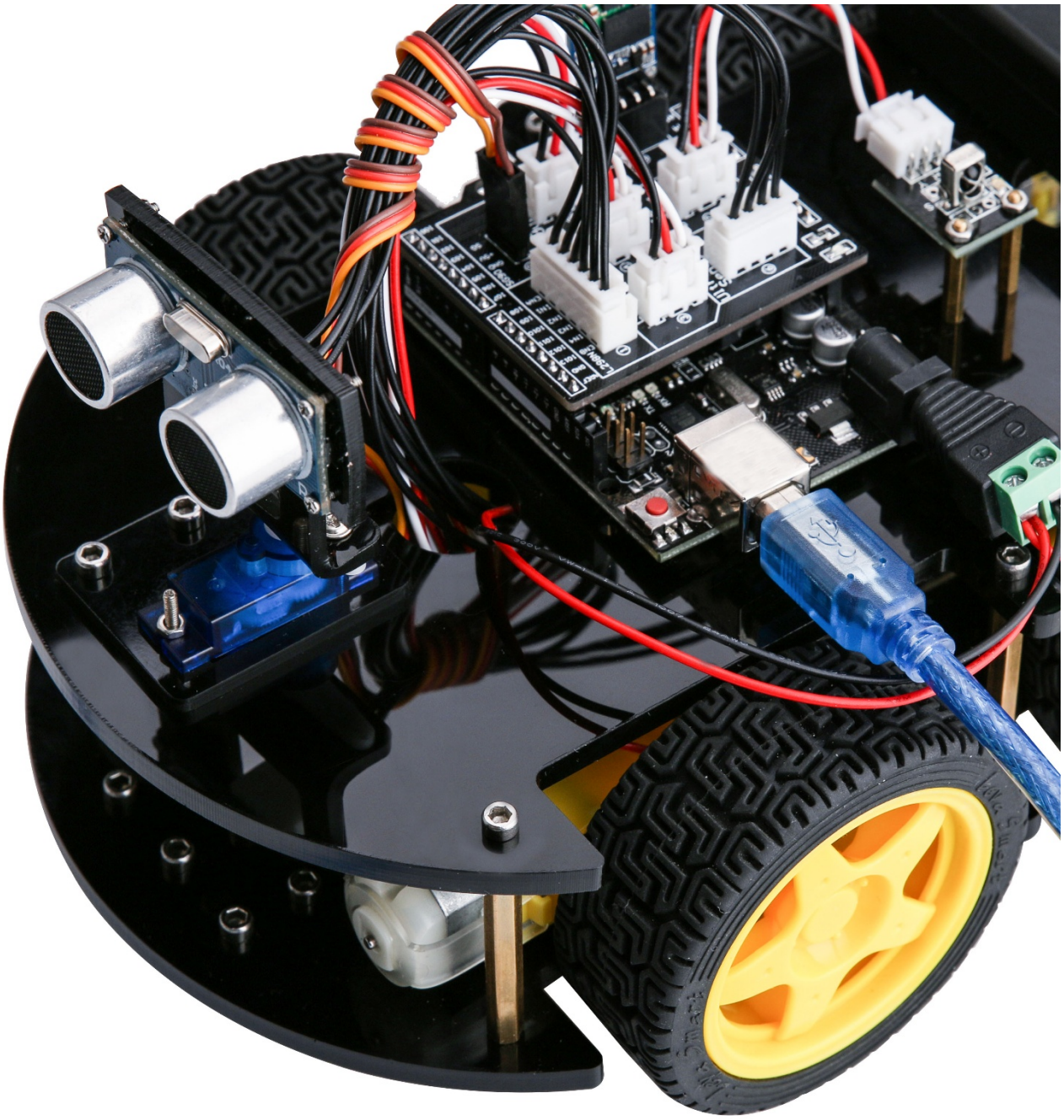


Next, the following icon appears on the desktop

Double-click to enter the desired development environment

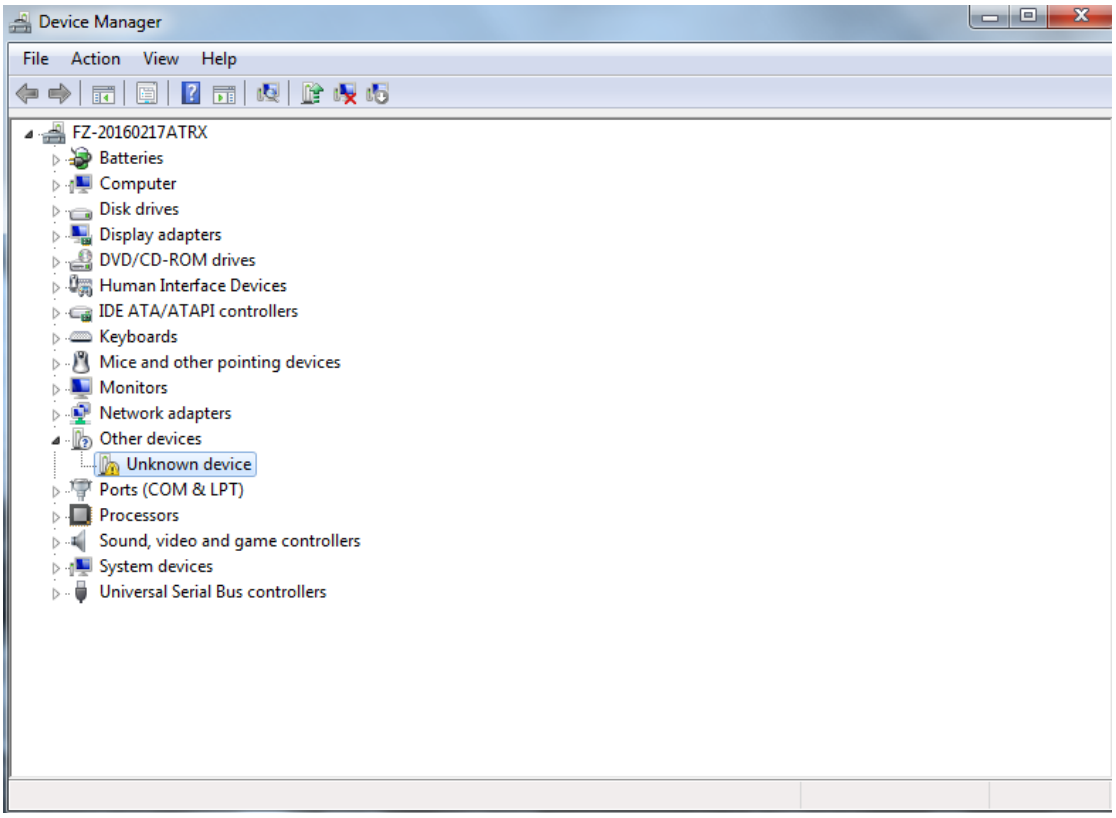


STEP3：Connect the car to the computer.

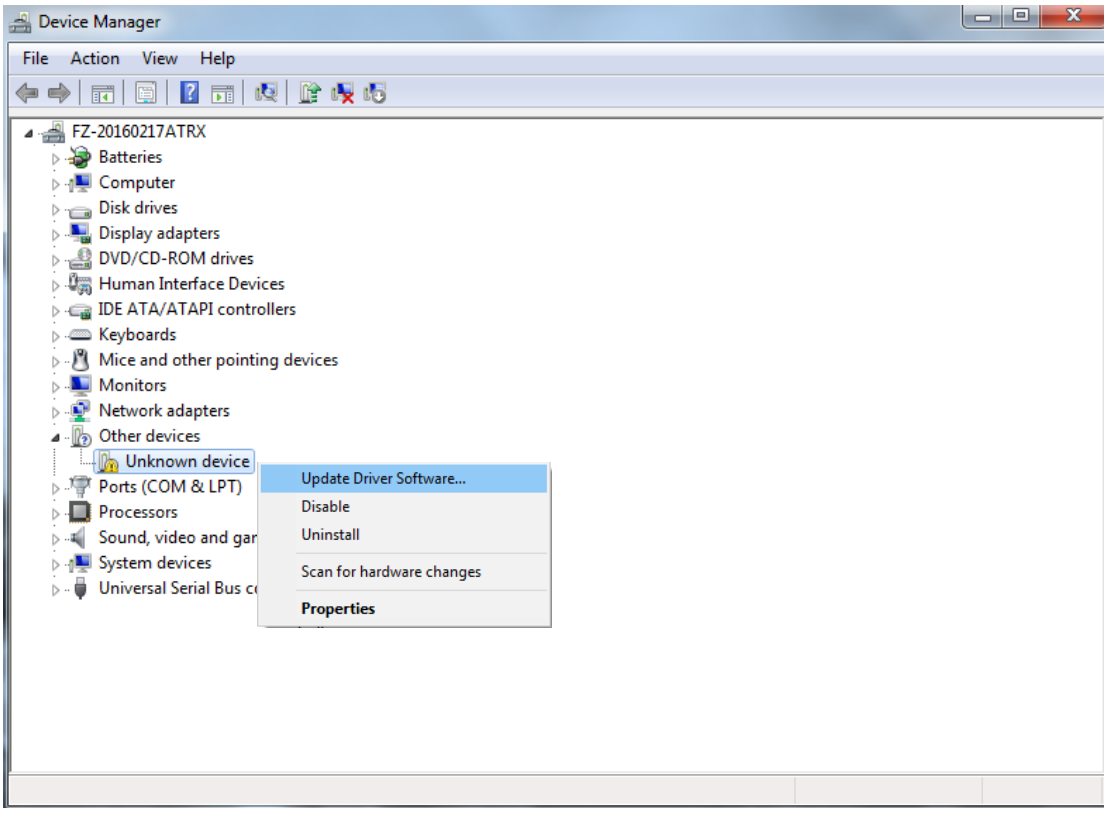


STEP 4: Open IDE——Tool——Port. If you see the right port, it means that the vehicle has been connected correctly to the computer. In this case, you can jump to STEP 5 directly. Otherwise, you need to install the driver in the following way.

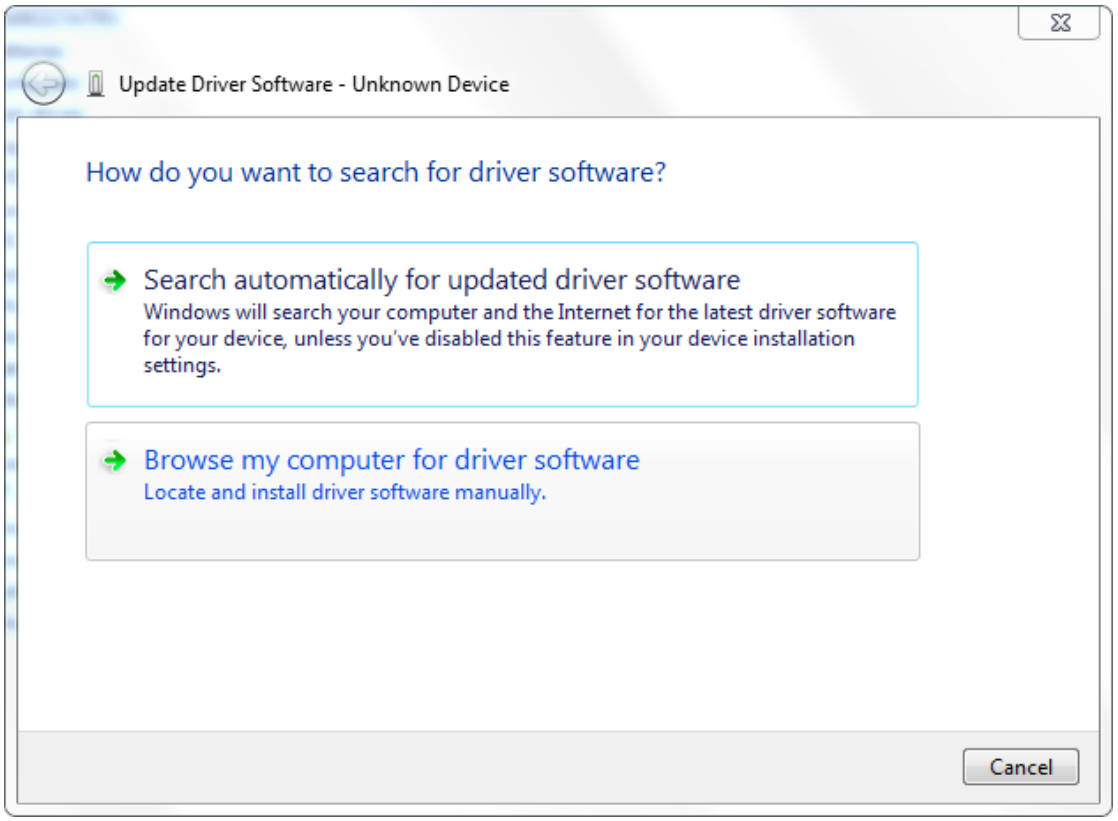
Open Device Manager by right clicking My Computer——Management——Device Manager



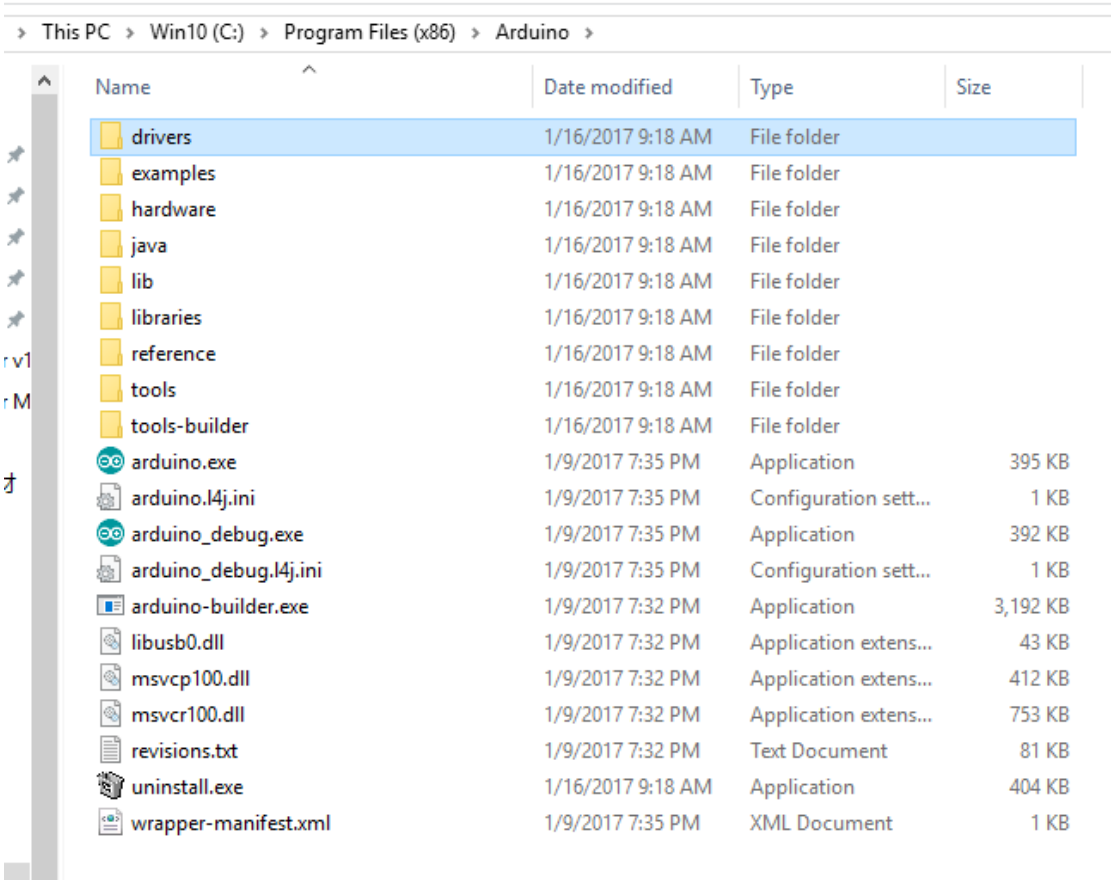
Right click unknown device-----update device software



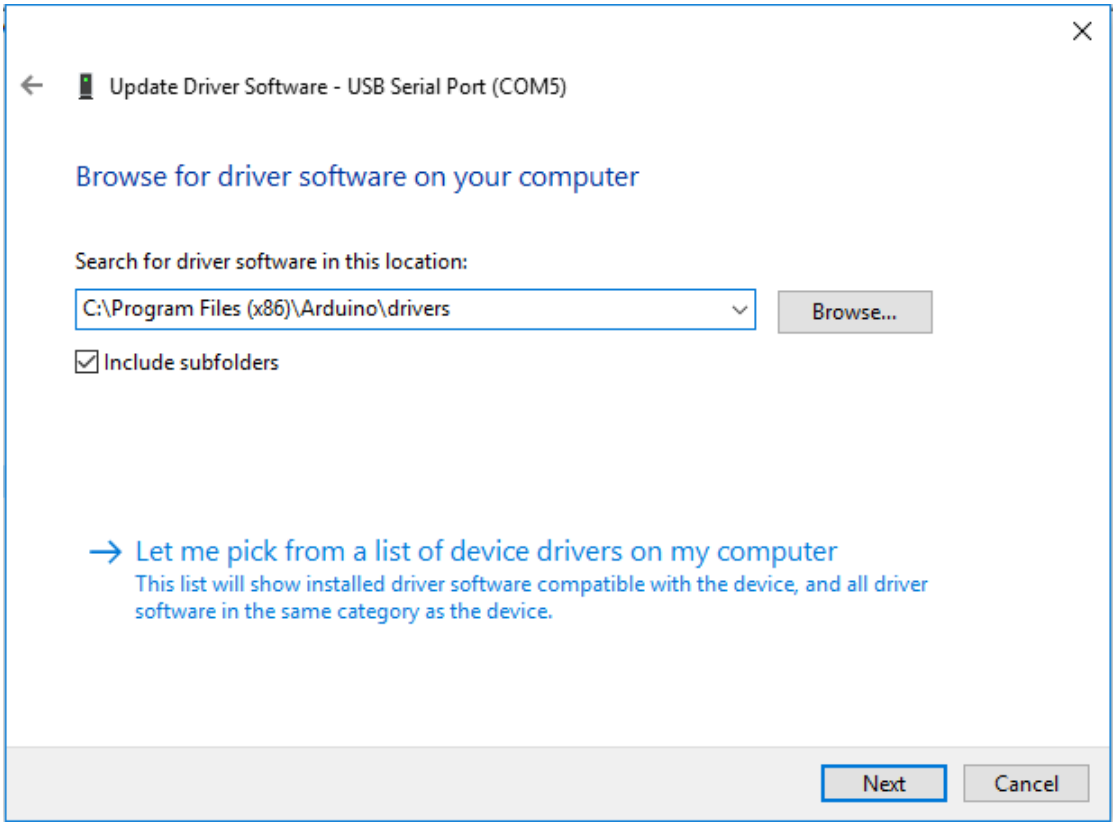
It shows that the driver has not been installed, and you need to click Browse my computer for driver software to find the drivers. The drives is in the Arduino folder. Normally you will install the folder in C:\Program Files (x86)\Arduino.



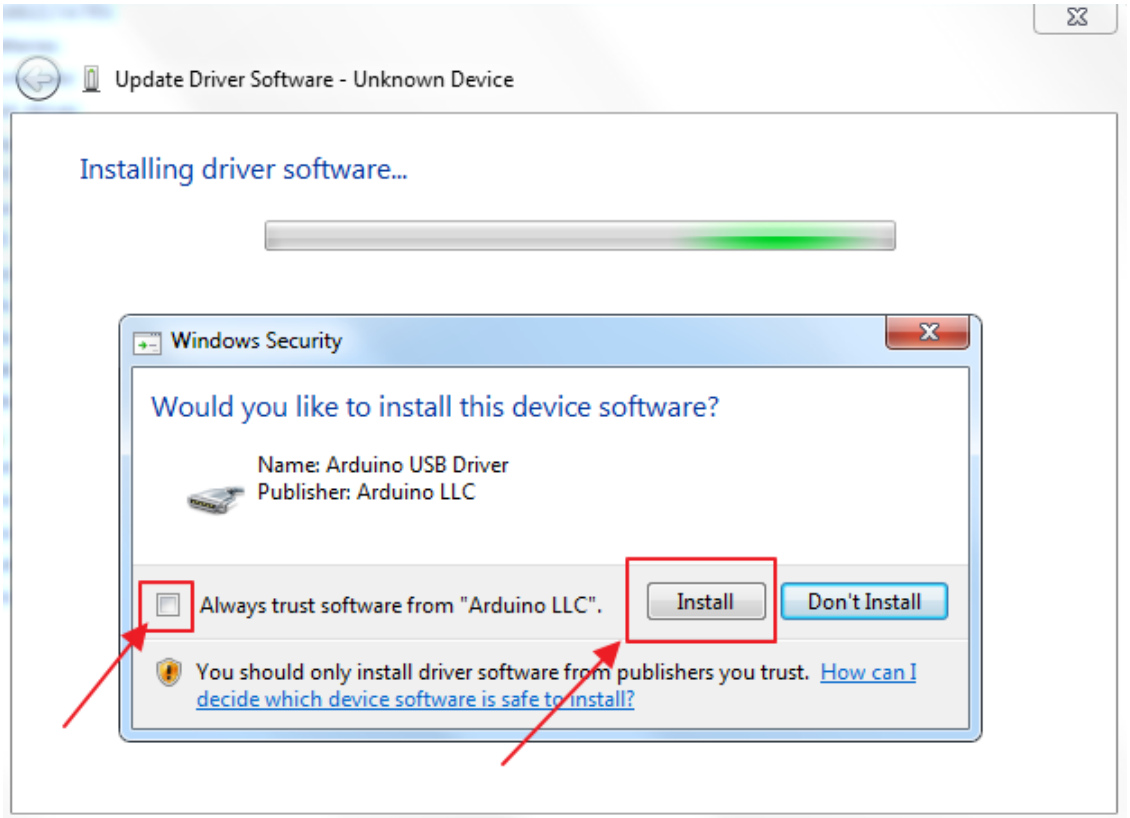
Arduino install folder

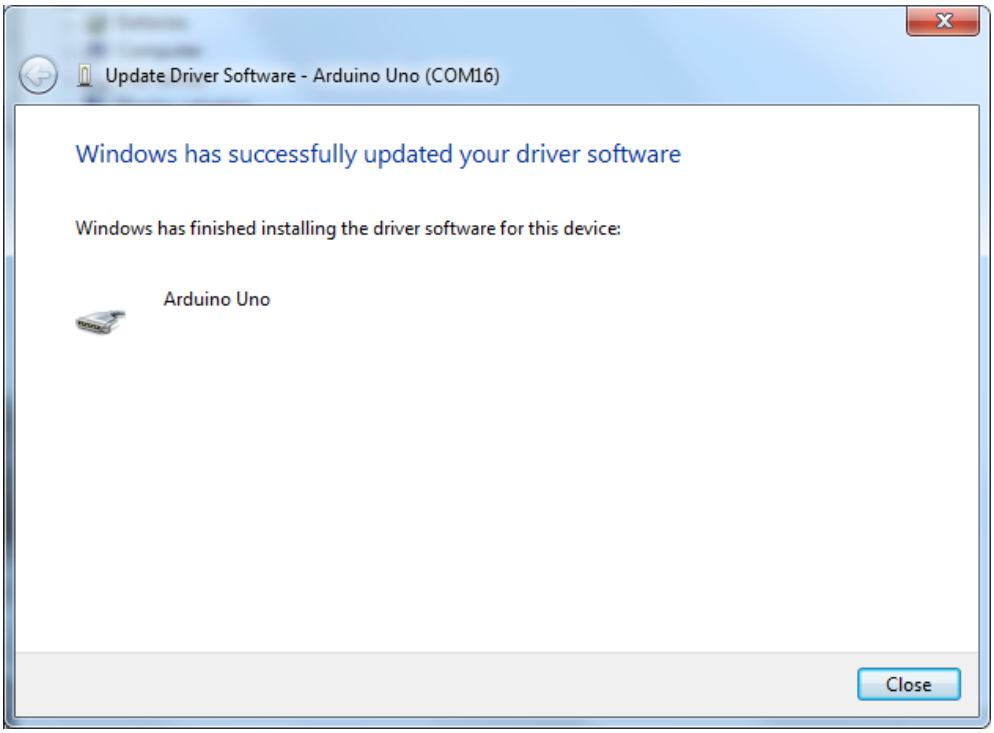


Select the Arduino driver folder

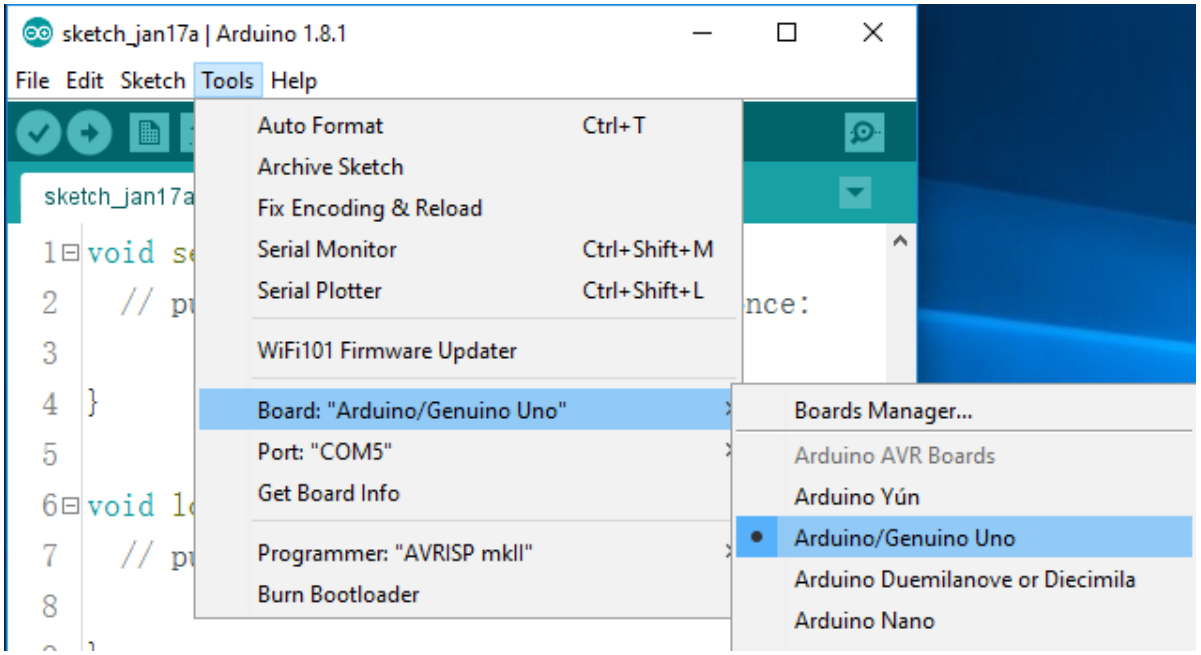


Install Arduino USB device

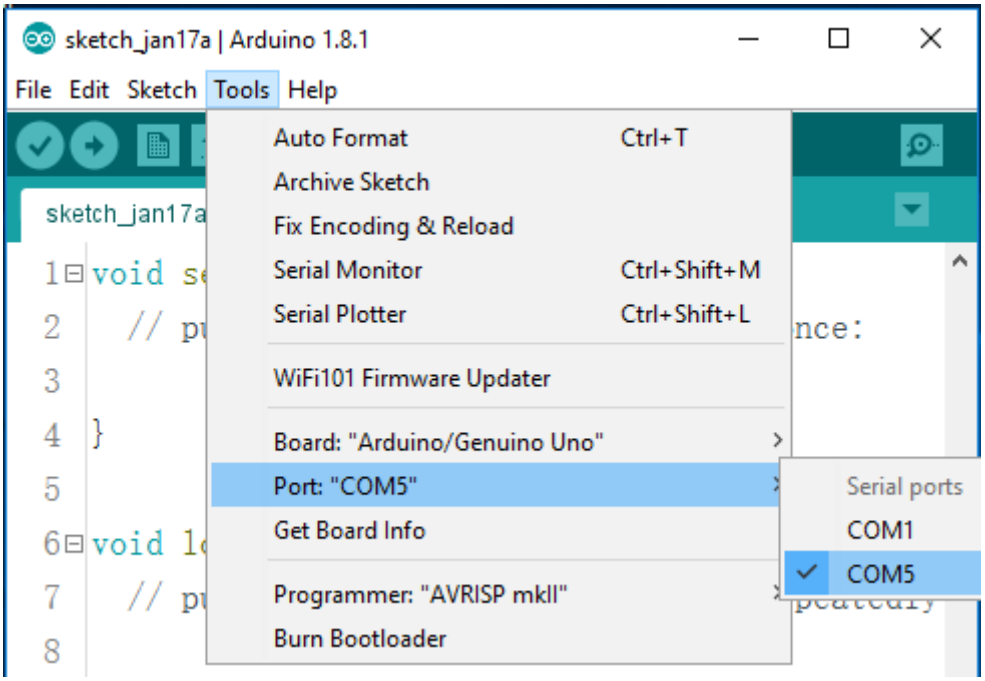




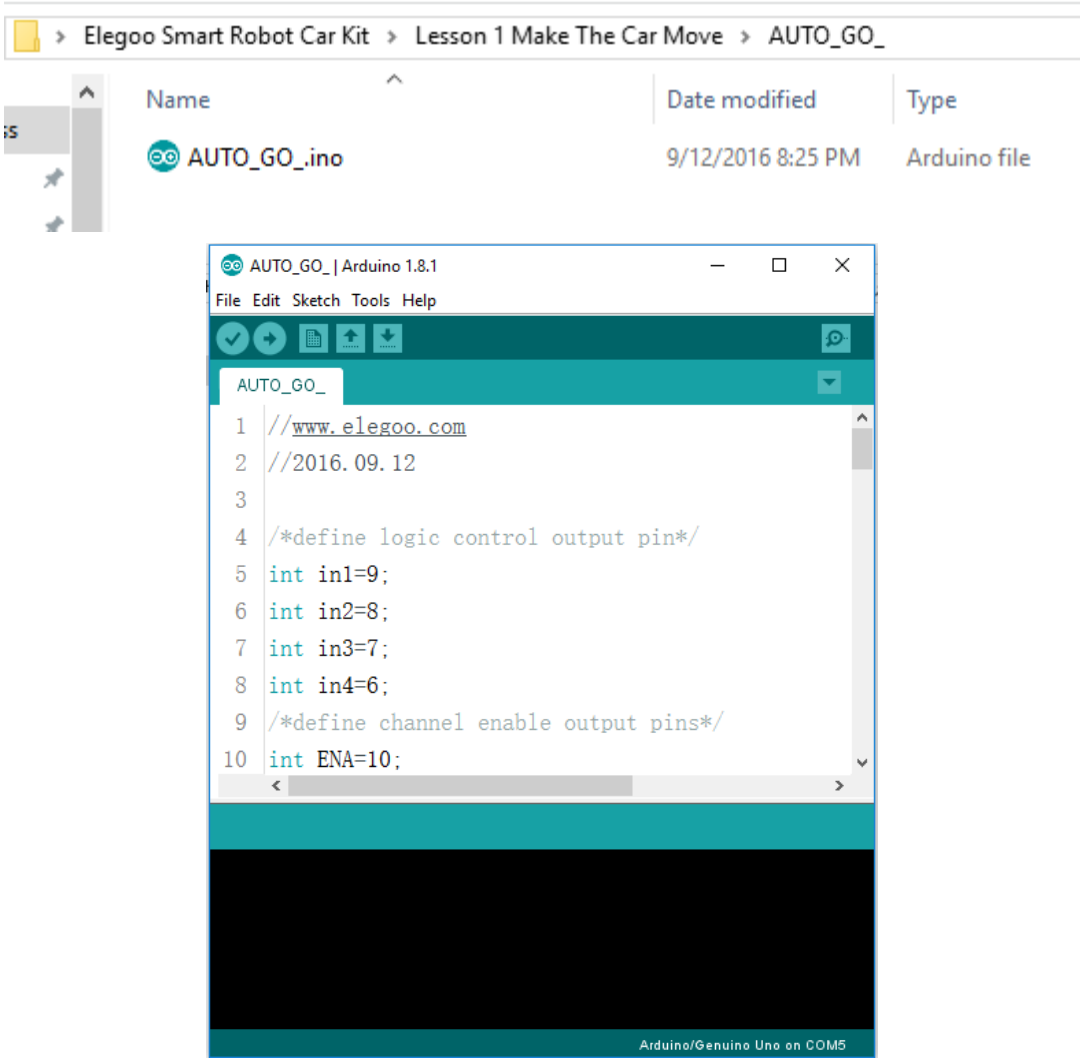
STEP5: After the driver is installed, please open the IDE and then click Tools---Board---Arduino/Genuino Uno



STEP6: Click Tools---Port---COMx (Arduino/Genuino Uno)



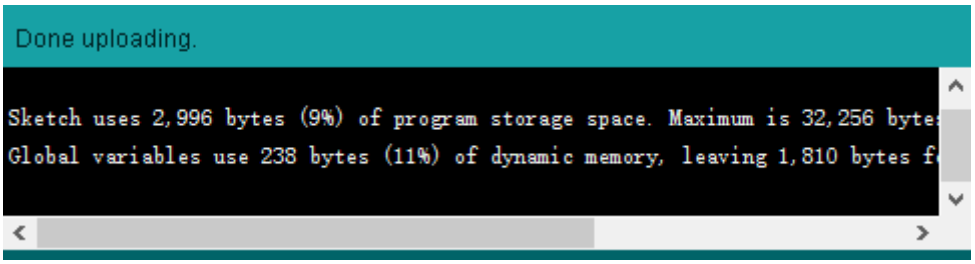
STEP7： Open the file AUTO_GO_\AUTO_GO_.ino and upload to the UNO controller board



Push the Button with arrow to upload program.

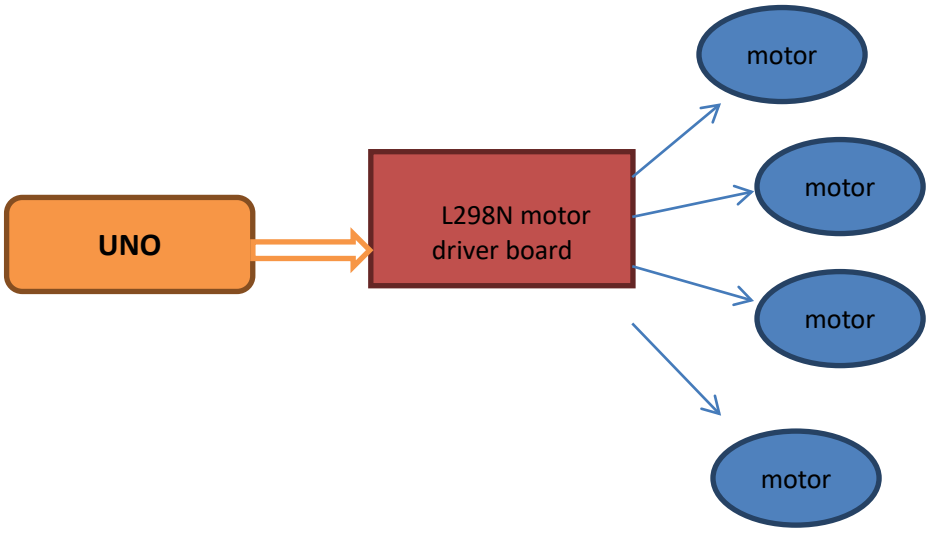
(TIPS: The bluetooth module should be pulled out when you upload the program every time, or it will be failed to upload the program.)

The picture below shows that it is uploaded successfully.



STEP8：Let's have a look at the results. Upload the program to the UNO controller board. After disconnecting the car to the computer, you can turn on the power switch and put the car on the ground. Then you will see the car moving.

Ⅲ. Description of Principles



How to use L298N motor driver board

Definition of the connection ports on L298N board have been marked above. The motors should be connected to the L298N board as the picture above, and if you find the rotational direction of one of the motors is opposite, please change the connecting position of its black and red wires.

L298N GND is connected to battery box GND;
L298N VCC is connected to battery box VCC;
UNO board is also connected to battery box.
L298N 5V here cannot be connected to UNO 5V;
ENA and ENB control the speed of right motor and speed of left motor separately by PWM.
IN1, IN2, IN3, IN4 : IN1 and IN2 are used to control right motor, IN3 and IN4 are used to control left motor. About the principle, please look at the sheet below: (We take right motor for example)

ENA	IN1	IN2	DC MOTOR STATUS
O	X	X	STOP
1	0	O	BRAKING
1	0	1	FORWARD
1	1	0	BACKWARD
1	1	1	BARKING

IV. Make The Car Move

The first step: Drive the motor

We will try to move the motor without speed controlling. Because it is easy to write program without speed controlling.

First of all, let's see the connection of the motor the L298N board, we will use Arduino 5, 6, 7, 8, 9, 10 pins to control the car. 6 and 7 pins control the right wheel. 8 and 9 pins control the left wheel. 5 and 10 pins control ENA and ENB.

So the connection is as below:

L298N	V5 expansion board
ENA	5
IN1	6
IN2	7
IN3	8
IN4	9
ENB	11

Based on the sheet given above, we first design a simple program to make the right wheel turn 0.5s in positive direction, stop 0.5s, turn 0.5s in negative direction and stop 0.5s. And the wheel will repeat the reaction.

Connect the UNO controller board to the computer, Open the file right_wheel_rotation\ right_wheel_rotation.ino

Code is as follow:

```
/*In1 connected to the 9 pin,
  In2 connected to the 8 pin, ENA pin 10,*/
int ENA=5;
int IN1=6;
int IN2=7;
void setup()
{
   pinMode(IN1,OUTPUT);
   pinMode(IN2,OUTPUT);
   pinMode(ENA,OUTPUT);
```

```
  digitalWrite(ENA,HIGH);
}
void loop()
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);      //Right wheel forward
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);                //Right wheel stop
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);               //Right wheel back
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);                //Right wheel stop
  delay(500);
}
```

Upload the program to the UNO board, disconnect it from the computer, and then switch on the car's power supply. You will see that the right wheel moves as you expected.

If the car is not moving, press the reset button on the UNO board.

If the moving direction of the motor is different from the direction you set, you can change the connection of black and red lines from the motor to L298N board.

Then, we make the left wheel rotate in the same way.

Connect the UNO controller board to the computer, Open the file Left_wheel_rotation\ Left_wheel_rotation.ino

Code is as follow:

```
/*In3 connected to the 7 pin,
  In4 connected to the 6 pin, ENB pin 5,*/
int ENB=11;
int IN3=8;
int IN4=9;
void setup()
```

```
{
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(ENB,OUTPUT);
  digitalWrite(ENB,HIGH);
}
void loop()
{
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);              //Left wheel forward
  delay(500);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);               //Left wheel stop
  delay(500);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);               //Left wheel back
  delay(500);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);               //Left wheel stop
  delay(500);
}
```

Upload the program to the UNO board, disconnect it from the computer, and then switch on the car's power supply. You will see that the right wheel moves as you expected.

The second step: Move forward and backward

After finishing debugging the car, you can write programs to make the car move.

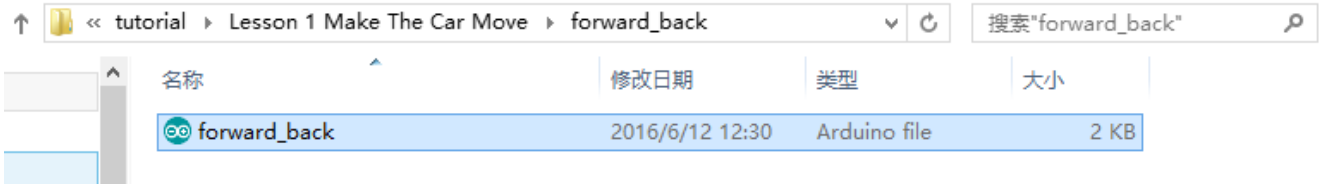
Below is the way how car moves:

CAR	forward	back	stop
Left wheel	Forward	back	stop
Right wheel	Forward	back	stop

CAR	Turn left	Turn right	stop
Left wheel	back	Forward	Stop
Right wheel	forward	back	stop

Next, we will write a simple program to make the car go forward 0.5s , then stop 0.5s, then back up 0.5s and then stop 0.5s.

Connect the UNO controller board to the computer, Open the file forward_back\ forward_back.ino



Code is as follow:

```
int ENA=5;
int IN1=6;
int IN2=7;
int ENB=11;
int IN3=8;
int IN4=9;
void setup()
{
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);

}
void loop()
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);              // left wheel goes forward
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);             // right wheel goes forward
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);              //left wheel holds still
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);              // right wheel holds still
  delay(500);
```

```
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);              //left wheel is back up
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);              // right wheel is back up
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);               // left wheel holds still
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);               // right wheel holds still
  delay(500);
}
```

Upload the program to the UNO board, disconnect it from the computer, and then switch on the car's power supply. You will see that the right wheel moves as you expected.

The third step: Write the program

It may be a difficult for you to write the whole program to make the car move automatically. So we separate the movements into different function, for example moving forward and turning left. And when we write the program in the final step, we can call the function.
Next, we begin to write programs for each movement:

Code is as follow:

```
/*********************************************
****  Forward sub function
functions：  Move forward


*********************************************
***/  void forward( )
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);               //Left wheel forward
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);          //Right wheel forward
}

/*********************************************
```

```
****  Forward sub function
functions：Move backward
*********************************************
***/  void back( )
{
   digitalWrite(IN1,LOW);
   digitalWrite(IN2,HIGH);              //Left wheel back
   digitalWrite(IN3,HIGH);
   digitalWrite(IN4,LOW);               //Right wheel back
```

```
}
/*********************************************
****  turnLeftsub function
functions：Turn left
*********************************************
***/  void turnLeft( )
{
   digitalWrite(IN1,HIGH);
   digitalWrite(IN2,LOW);              //Left wheel back
   digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);           //Right wheel forward
}
/*********************************************
****  turn Right sub function
functions：Turn right
*********************************************
***/  void turnRight( )
{
   digitalWrite(IN1,LOW);
   digitalWrite(IN2,HIGH);             //Left wheel forward
   digitalWrite(IN3,LOW);
   digitalWrite(IN4,HIGH);             //Right wheel back
}
/*********************************************
****  stop sub function
functions：Stop
*********************************************
***/  void _stop()
{
   digitalWrite(IN1,LOW);
   digitalWrite(IN2,LOW);              //Left wheel stop
   digitalWrite(IN3,LOW);
   digitalWrite(IN4,LOW);              //Right wheel stop
}
```

The fourth step: Move automatically

Finally, we start to write program to make the car move automatically: go forward 0.4s - back up 0.4s - turn left 0.4s - turn right 0.4s.

Connect the UNO controller board to the computer, Open the file AUTO_GO_\ AUTO_GO_.ino

The code is as below:

```
/*define logic control output pin*/
int in1=6;
int in2=7;
int in3=8;
int in4=9;
/*define channel enable output pins*/
int ENA=5;
int ENB=11;
/*define forward function*/
void _mForward()
{
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(in1,HIGH);//digital output
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
  Serial.println("Forward");
}
/*define back function*/
void _mBack()
```

```
{
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  Serial.println("Back");
}
/*define left function*/
void _mleft()
{
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  Serial.println("Left");
}
/*define right function*/
void _mright()
{
  digitalWrite(ENA,HIGH);
  digitalWrite(ENB,HIGH);
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
  Serial.println("Right");
}
/*put your setup code here, to run once*/
void setup() {
```

```
  Serial.begin(9600); //Open the serial port and set the baud rate to 9600
/*Set the defined pins to the output*/
    pinMode(in1,OUTPUT);
    pinMode(in2,OUTPUT);
    pinMode(in3,OUTPUT);
    pinMode(in4,OUTPUT);
    pinMode(ENA,OUTPUT);
    pinMode(ENB,OUTPUT);
}
/*put your main code here, to run repeatedly*/
void loop() {
_mForward();
delay(1000);
_mBack();
delay(1000);
_mleft();
delay(1000);
_mright();
delay(1000);
}
```

Upload the program to the UNO board, disconnect it from the computer, and then switch on the car's power supply. You will see that the right wheel moves as you expected.