



Module_ Test

Module_Test



Module_Test

i. Points of This Section

ii. Learning Objectives:

-  ***1. Learn how to use Arduino IDE***
-  ***2. Make the car move by uploading program***

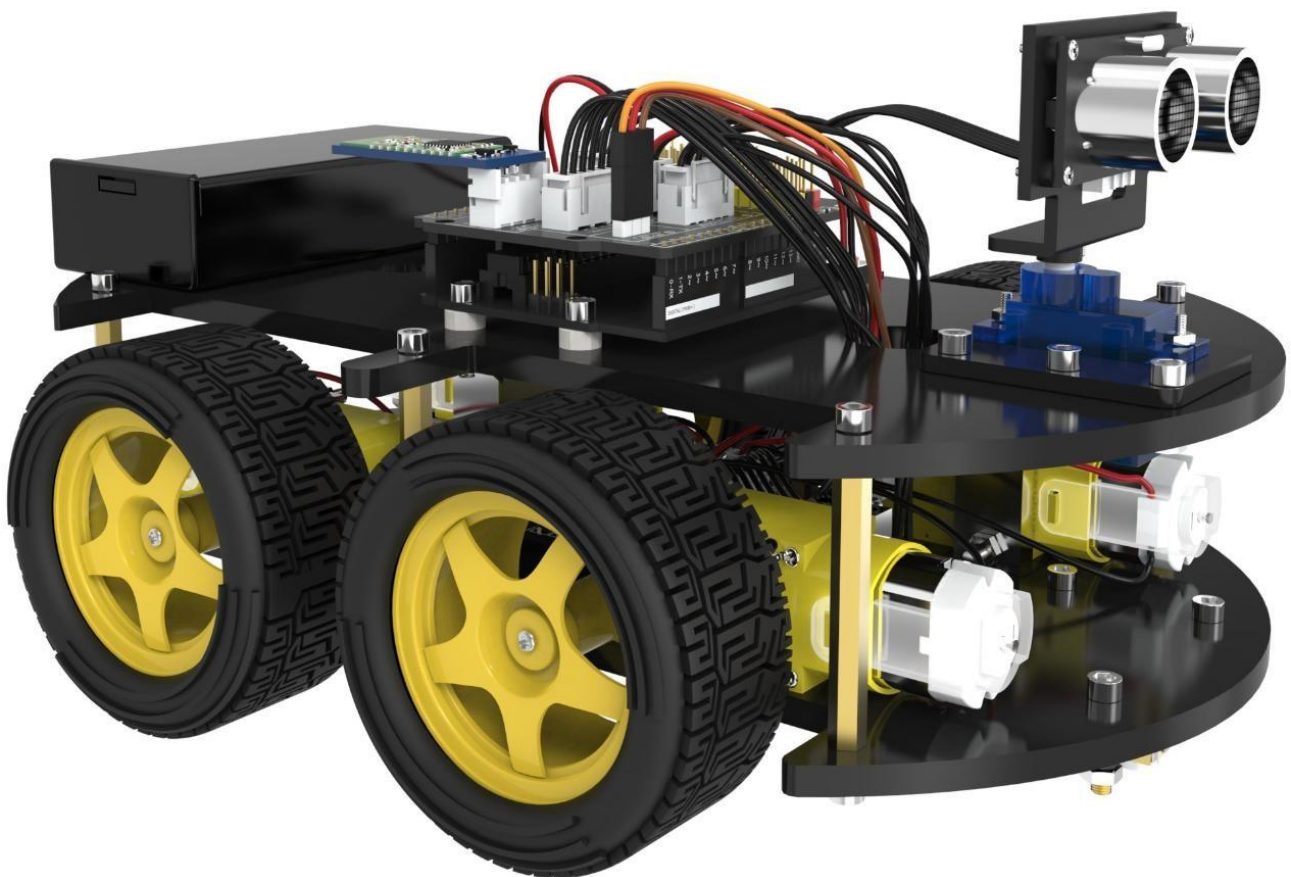
iii. Preparations:

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

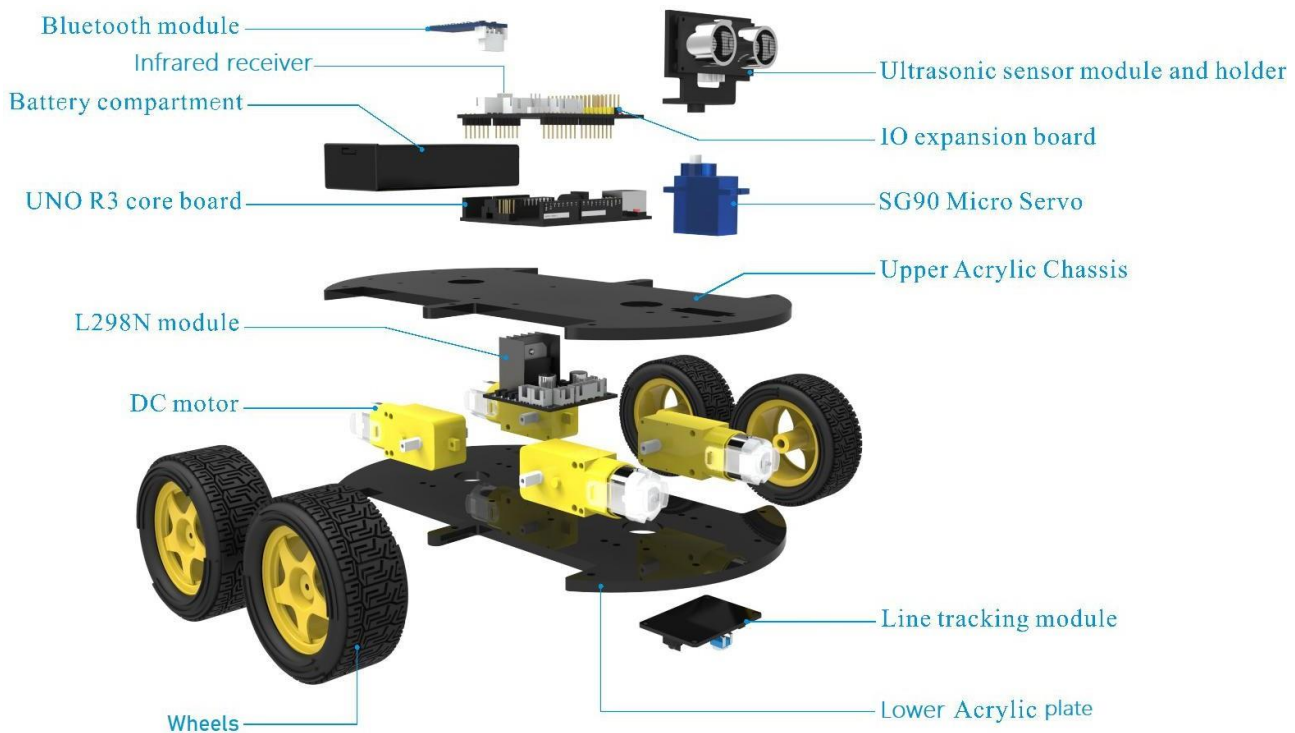
I . Introduction of the car

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 introduce the small vehicle that will accompany us for a long time in the future.



Each parts of the car is as below:



Function of each part:

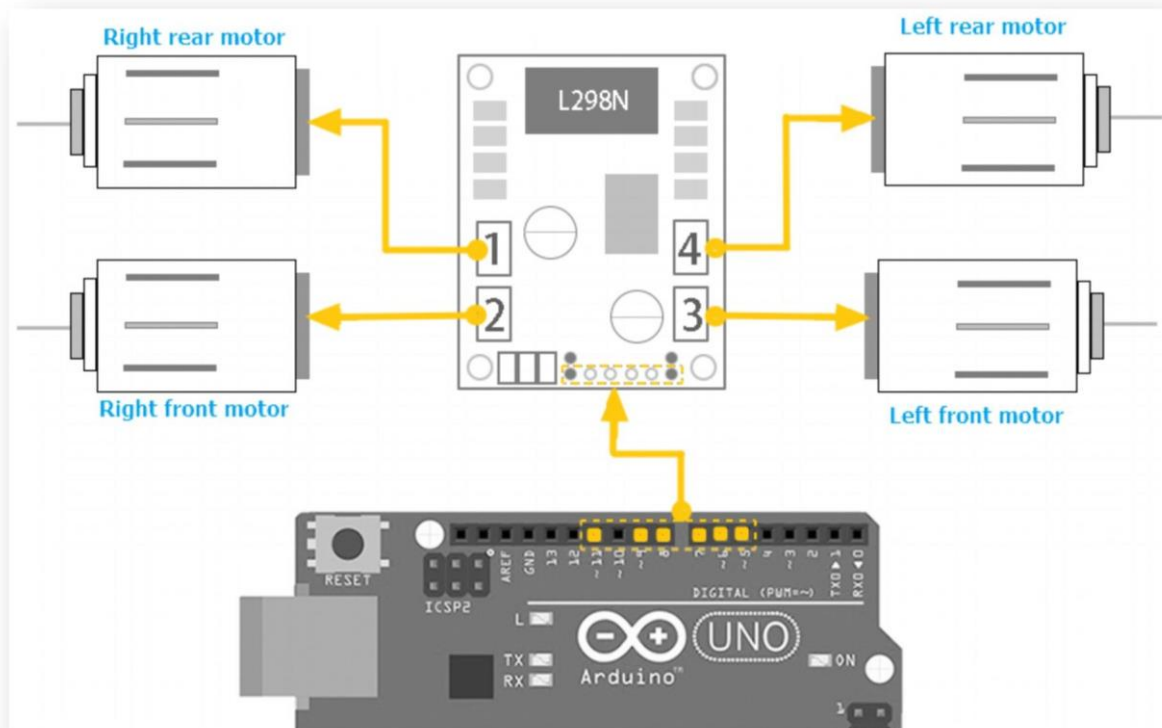
1. Battery compartment (with a switch): supply power for the vehicle
2. DC motor + wheels: drive the vehicle to move
3. Upper and under Acrylic Chassis: the frame of the car
4. L298N motor driving board: drive the motor to rotate
5. UNO R3 core board: the brain of the car, controls all the parts
6. IO expansion board: combined with the UNO, make connection become more easier
7. SG90 Micro Servo: enable the GP2Y0A21 distance sensor to rotate 180 degrees
8. Ultrasonic sensor module and holder: 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

Because we have uploaded the program to the car in Lesson 0, now, you can turn on the power switch and put the car on the ground. Then you will see the car moving.

Tips: Before turning on the power switch, check whether the battery is fully charged. If the battery is low, charge it in time. In the charging process, the charger shows a red LED indicates that the battery is not fully charged, the charger shows a blue LED indicates that it is fully charged.

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 shown in 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 and left motor separately by PWM.

IN1, IN2, IN3 and IN4: IN1 and IN2 are used to control left motor, IN3 and IN4 are used to control right motor. About the principle, please look at the sheet below: (We take left motor for example)

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

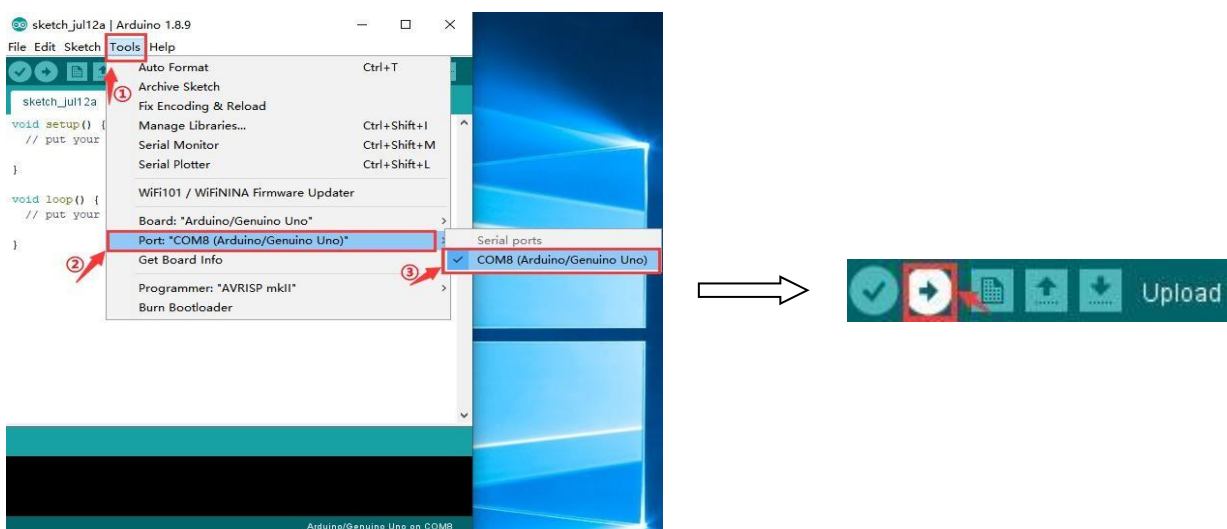
IV. Make the Car Move

TIPS:

When uploading codes, please remove the Bluetooth module from the IO expansion board
(Because the serial port for uploading codes and Bluetooth communication is the same one and there will be conflicts.)

You can mount the Bluetooth module after the upload.

Please make sure your com is right before click on upload to upload the program.



The first step: drive the motor

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

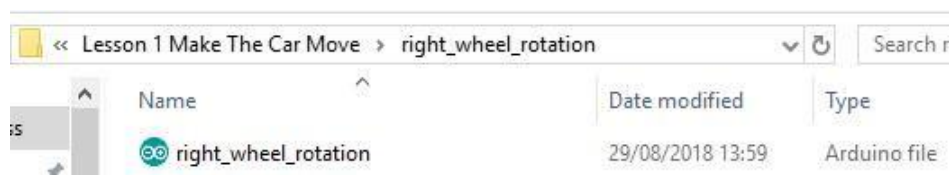
First of all, let's see the connection of the motor and the L298N board, we will use Arduino 5, 6, 7, 8, 9, 11 pins to control the car. 9 and 11 pins control the right wheel. 7 and 8 pins control the left wheel. 5 and 6 pins control ENA and ENB respectively.

So the connection is as below:

L298N	V5 expansion board
ENA	5
ENB	6
IN1	7
IN2	8
IN3	9
IN4	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 code file in the path "**\Lesson 1 Make The Car Move\right_wheel_rotation\ right_wheel_rotation.ino**". 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 the L298N board.

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

Connect the UNO controller board to the computer, open the code file in the path **“Lesson 1 Make The Car Move\Left_wheel_rotation\ Left_wheel_rotation.ino”**. 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 code file in the path **“Lesson 1 Make The Car Move\forward_back\forward_back.ino”**. Upload the program to the UNO board.



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 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:

The fourth step: move automatically

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 code file in the directory **"Lesson 1 Make The Car Move\AUTO_GO_\AUTO_GO.ino"**. Upload the program to the UNO board.

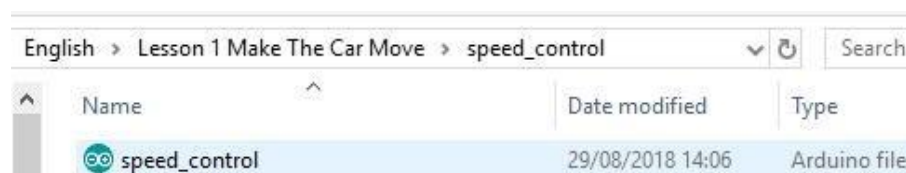


Disconnect it from the computer, and then switch on the car's power supply. You will see that the wheel moves as you expected.

The fifth step: speed_control

The code to achieve the function is to control the speed of the car: go forward and reduce the speed → stop 1s → running back and accelerate → stop 2s.

Connect the UNO controller board to the computer, open the code file in the directory **"Lesson 1 Make The Car Move\speed_control\ speed_control.ino"**. Upload the program to the UNO board.



The sixth step: car_control

The code to achieve the function is to control the speed of the car and you can set the speed you like to change the value of **CAR_SPEED**.

Connect the UNO controller board to the computer, open the code file in the directory "**Lesson 1 Make The Car Move\speed_control\car_control.ino**". Upload the program to the UNO board.

English > Lesson 1 Make The Car Move > car_control				Search car...
	Name	Date modified	Type	
	 car_control	16/07/2019 10:47	Arduino file	