# **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:

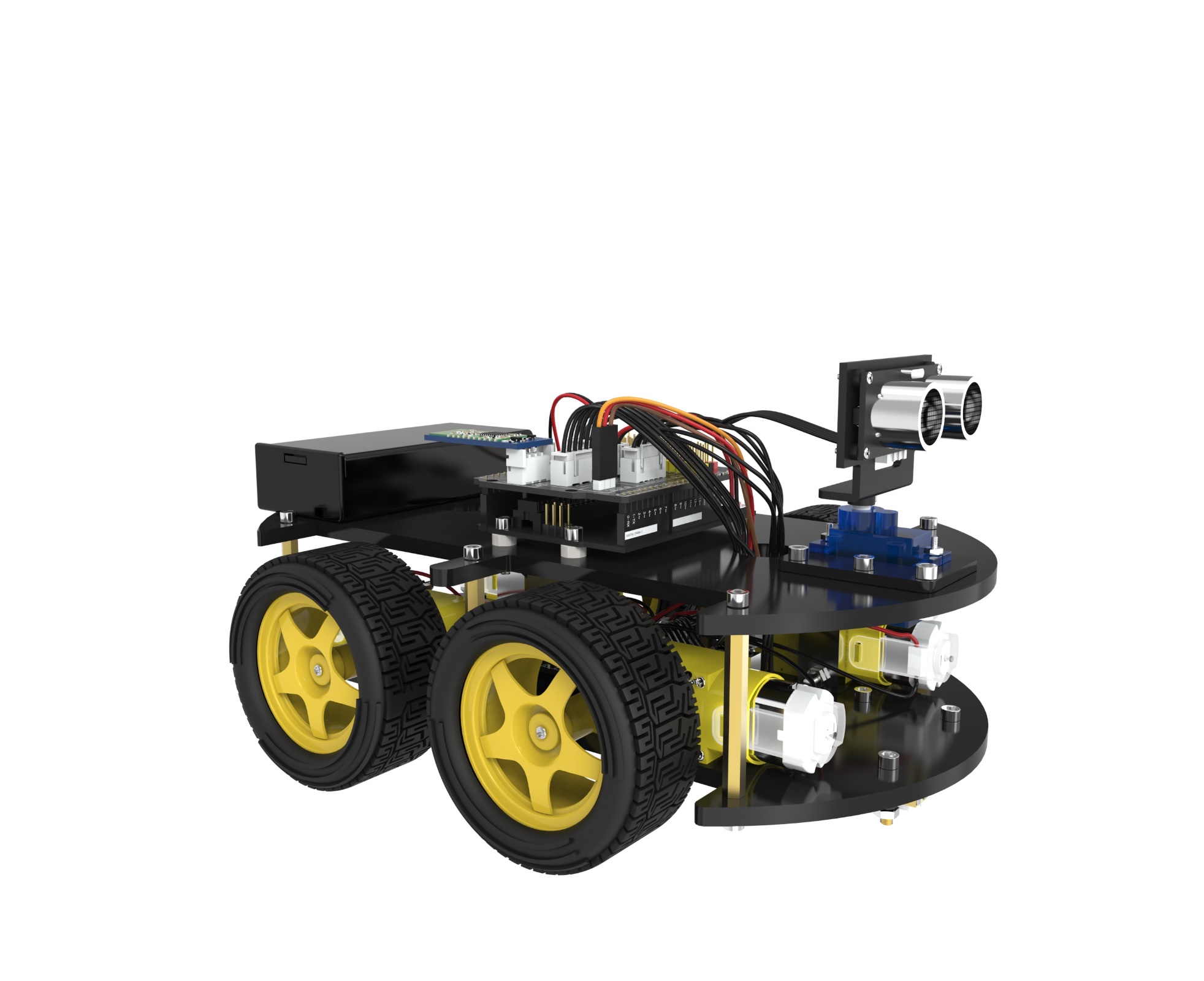
One car (with a battery)

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.



### Each parts of the car is as below:

### 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. We will use the Arduino Software IDE (Integrated Development Environment) as a programming tool.

### 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.

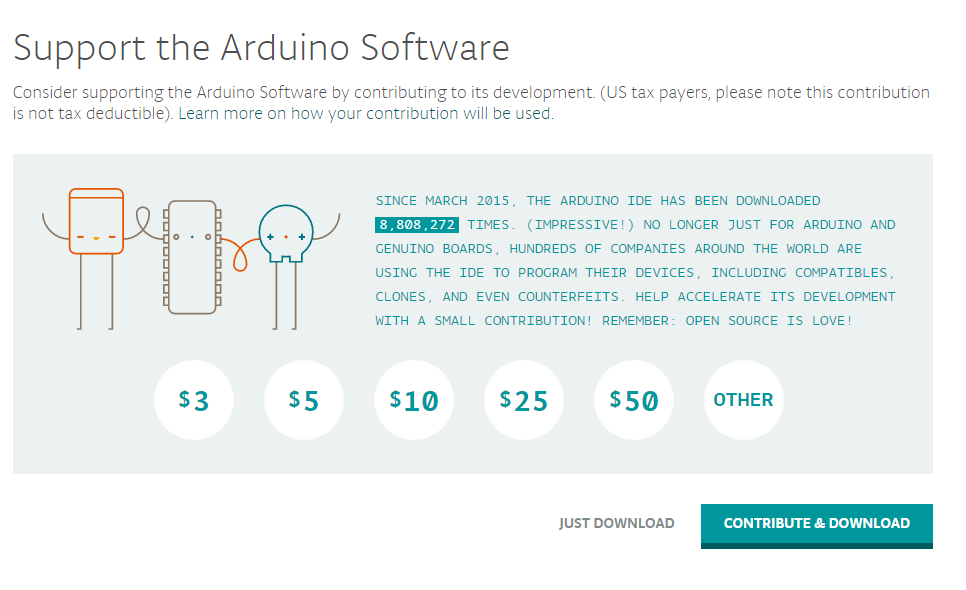


The following is the exe implementation of the installation procedures.

Press the char “Windows Installer”



Press the button “JUST DOWNLOAD” to download the software.





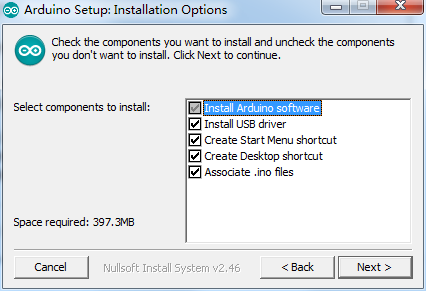
The download file



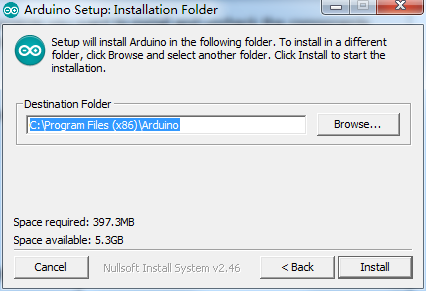
These are available in the materials we provide, and the versions of our materials are the latest versions when this course was made.



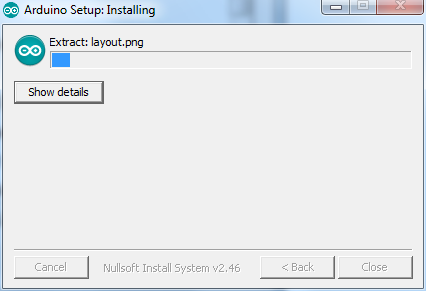
Choose I Agree to see the following interface



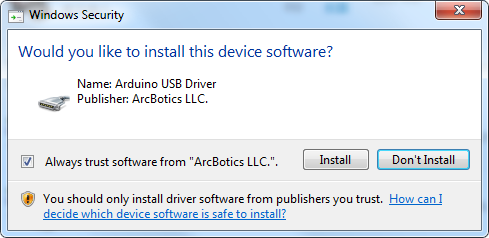
Choose Next



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



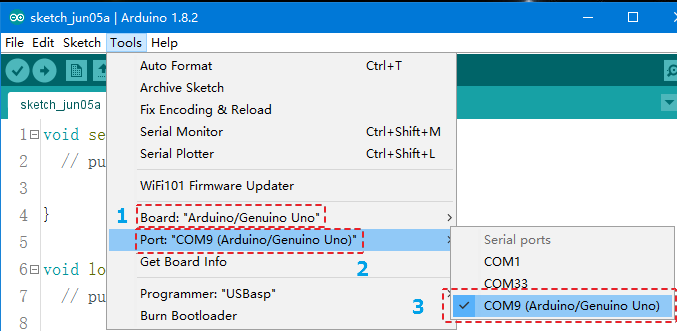
Double-click to enter the desired development environment



### STEP3：Connect the car to the computer.

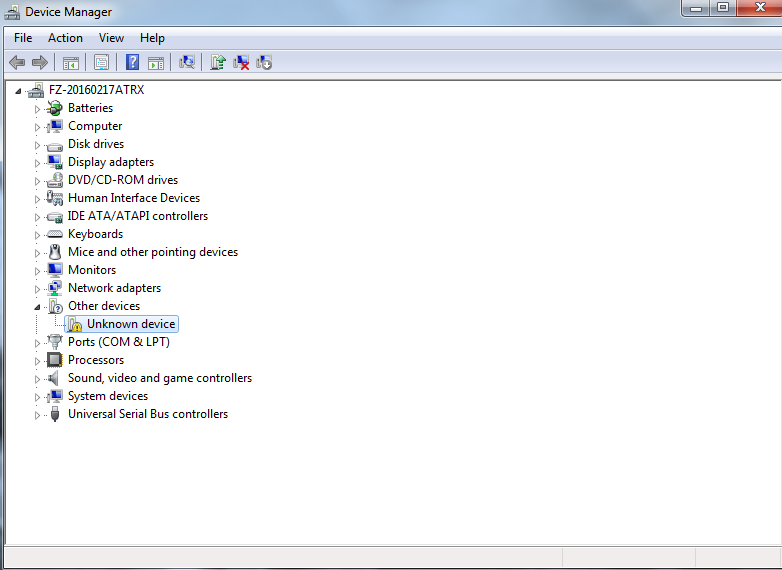
### STEP 4: Open the Arduino IDE. Select “Tool” 🡪 “Board:” 🡪 ”Arduino/Genuino Uno”. Select “Tool” 🡪”Port:”🡪”COM (Arduino/Genuino Uno)”.

**Each Arduino Uno board has a different COM number on the same computer and usually the COM number with a suffix name “(Arduino/Genuino Uno)” in Arduino 1.8.2. You should choose the COM number of the actual display.**

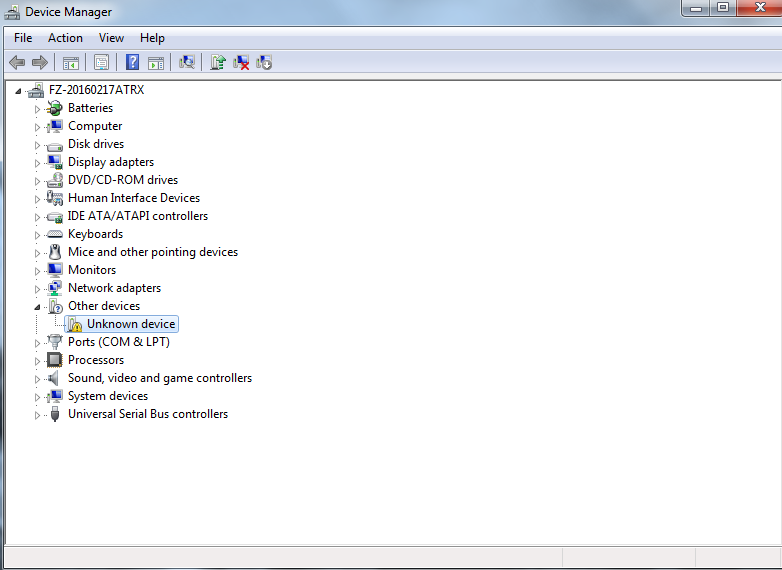


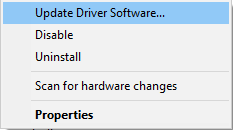
**If you see the port “COM (Arduino/Genuino Uno)”, it means that the vehicle has been connected correctly to the computer. In this case, you can jump to STEP 5 directly. Otherwise, you need to install the driver in the following way.**

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

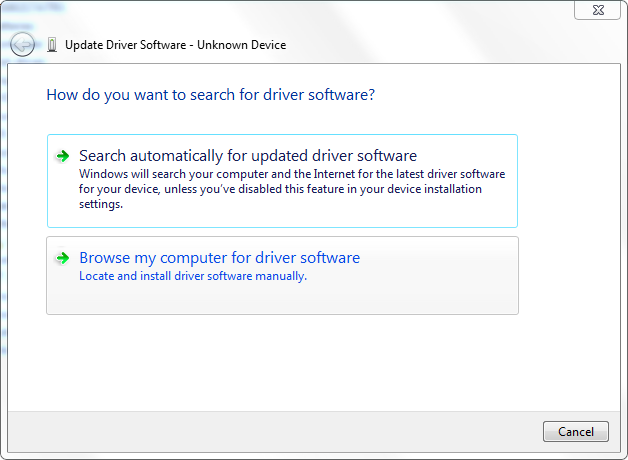


Right click unknown device-----update device software

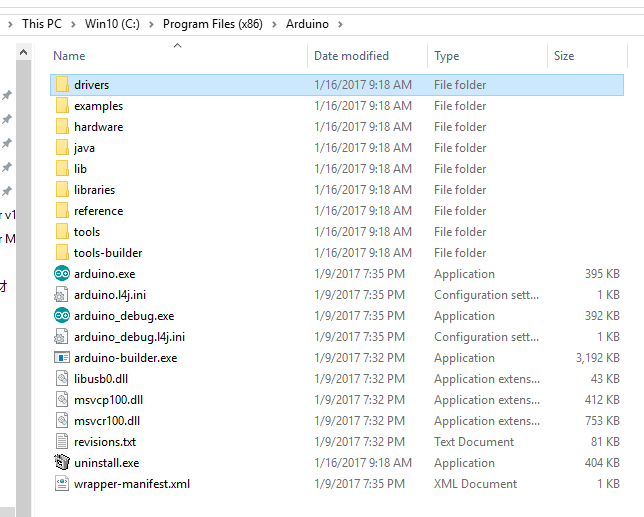




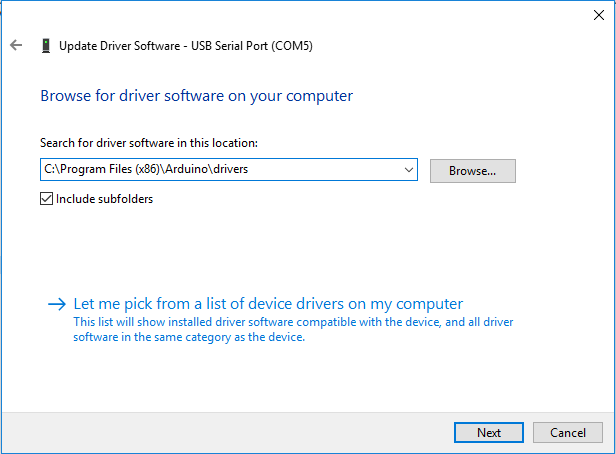
It shows that the driver has not been installed, and you need to click Browse my computer for driver software to find the drivers. The drives is in the Arduino folder. Normally you will install the folder in C:\Program Files (x86)\Arduino.



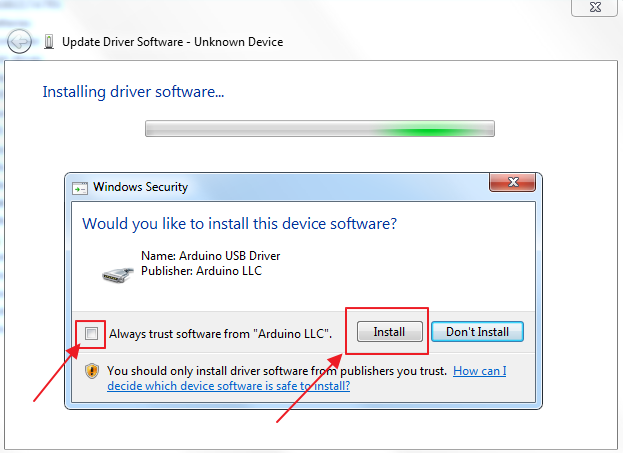
Arduino install folder

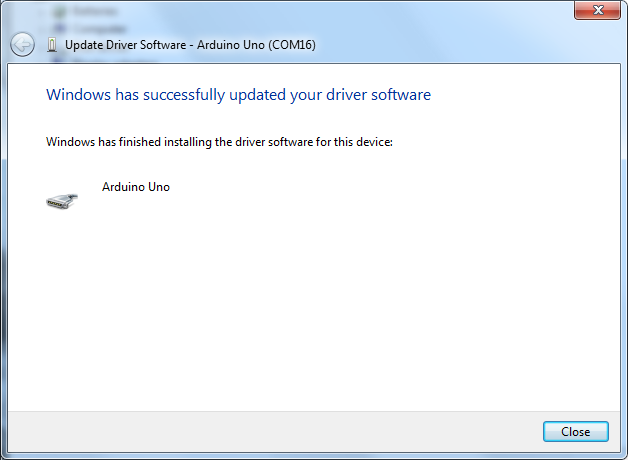


Select the Arduino driver folder

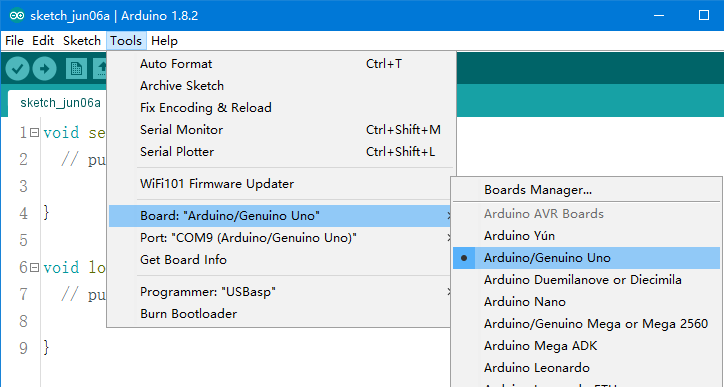


Install Arduino USB device





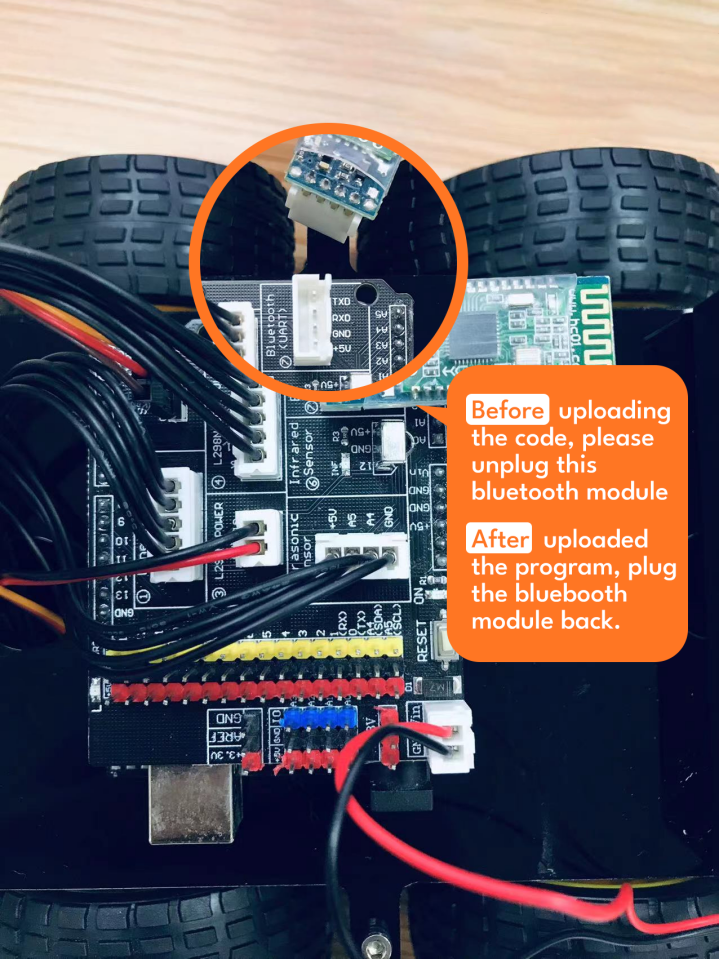
### **STEP5: After the driver is installed, please open the IDE and then click “Tools”🡪”Board”🡪** “**Arduino/Genuino Uno”.**

****

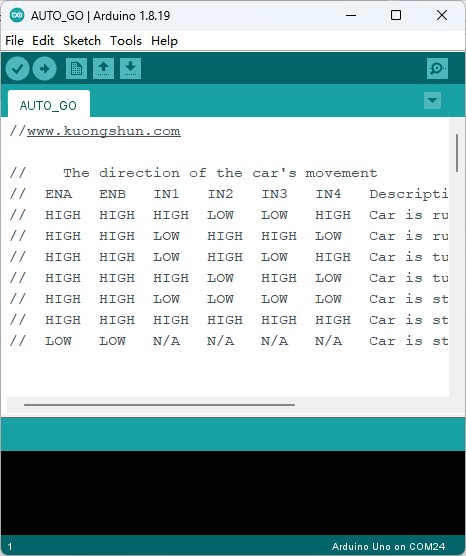
### STEP6: Click “Tools”🡪”Port”🡪COM.

### STEP7：Open the code file in the directory “\Lesson 1 Make The Car Move \AUTO\_GO\_\AUTO\_GO\_.ino” and upload to the UNO controller board.

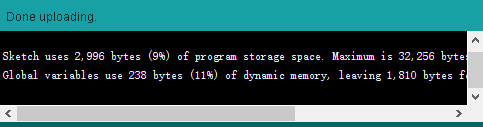
**TIPS: The bluetooth module should be pulled out when you upload the program every time, or it will be failed to upload the program.**

****







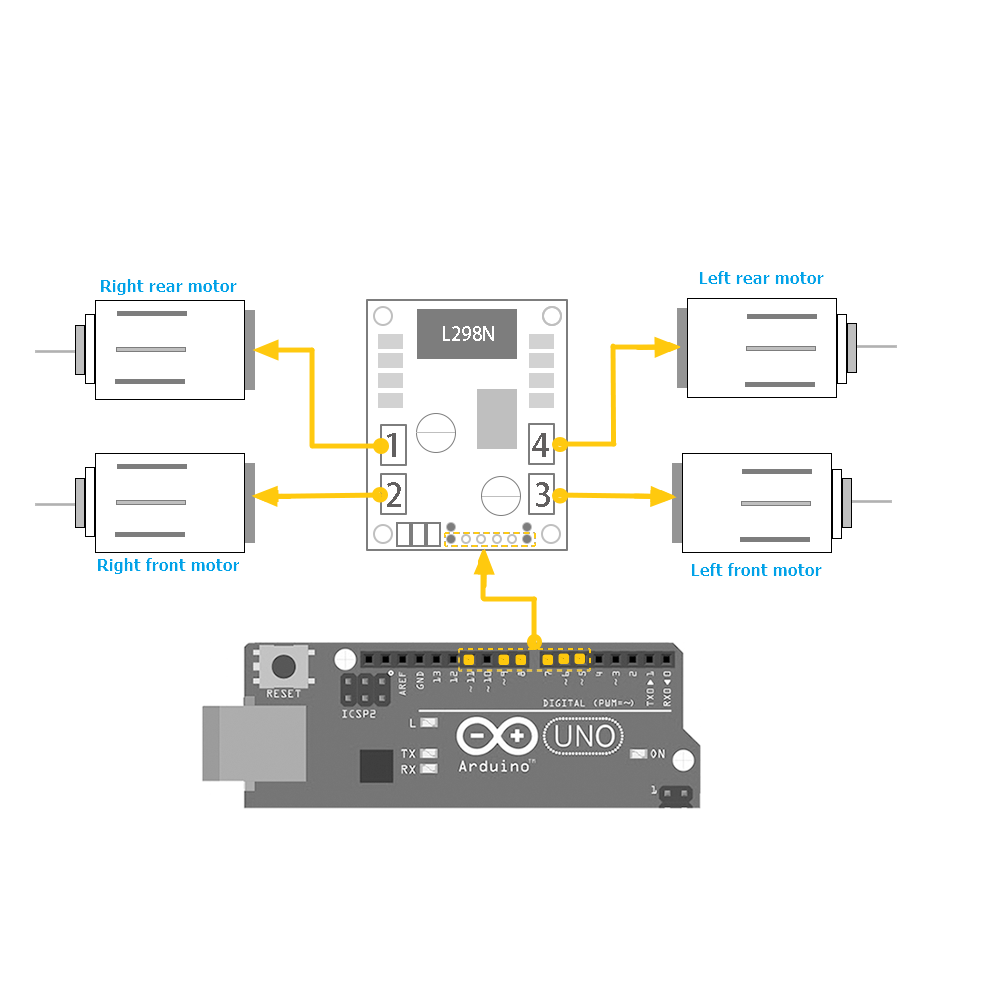


**The picture above shows that it is uploaded successfully.**

### 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.

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 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 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)

|  |  |  |  |
| --- | --- | --- | --- |
| **ENB** | **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

### 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, 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.

So the connection is as below:

|  |  |
| --- | --- |
| L298N | V5 expansion board |
| ENB | 5 |
| ENA | 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.**

****

**Code preview:**

*//www.kuongshun.com*

*//     Right motor truth table*

*//Here are some handy tables to show the various modes of operation.*

*//  ENB         IN3             IN4         Description*

*//  LOW   Not Applicable   Not Applicable   Motor is off*

*//  HIGH        LOW             LOW         Motor is stopped (brakes)*

*//  HIGH        LOW             HIGH        Motor is on and turning forwards*

*//  HIGH        HIGH            LOW         Motor is on and turning backwards*

*//  HIGH        HIGH            HIGH        Motor is stopped (brakes)*

*// define IO pin*

#define ENB 6

#define IN3 9

#define IN4 11

*//init the car*

void setup() {

  pinMode(IN3, OUTPUT); *//set IO pin mode OUTPUT*

  pinMode(IN4, OUTPUT);

  pinMode(ENB, OUTPUT);

  digitalWrite(ENB, HIGH); *//Enable right motor*

}

*//mian loop*

void loop() {

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, HIGH);*//Right wheel turning forwards*

  delay(500); *//delay 500ms*

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, LOW); *//Right wheel stoped*

  delay(500);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, LOW); *//Right wheel turning backwards*

  delay(500);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, HIGH); *//Right wheel stoped*

  delay(500);

}

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 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.**

******

**Code preview:**

*//www.kuongshun.com*

*//     Left motor truth table*

*//Here are some handy tables to show the various modes of operation.*

*//  ENA         IN1               IN2         Description*

*//  LOW   Not Applicable    Not Applicable    Motor is off*

*//  HIGH        LOW               LOW         Motor is stopped (brakes)*

*//  HIGH        HIGH              LOW         Motor is on and turning forwards*

*//  HIGH        LOW               HIGH        Motor is on and turning backwards*

*//  HIGH        HIGH              HIGH        Motor is stopped (brakes)*

*// define IO pin*

#define ENA 5

#define IN1 7

#define IN2 8

*//init the car*

void setup() {

  pinMode(IN1, OUTPUT); *//set IO pin mode OUTPUT*

  pinMode(IN2, OUTPUT);

  pinMode(ENA, OUTPUT);

  digitalWrite(ENA, HIGH);*//Enable left motor*

}

*//mian loop*

void loop() {

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW); *//Right wheel turning forwards*

  delay(500); *//delay 500ms*

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW); *//Right wheel stoped*

  delay(500);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, HIGH);*//Right wheel turning backwards*

  delay(500);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, HIGH); *//Right wheel stoped*

  delay(500);

}

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.

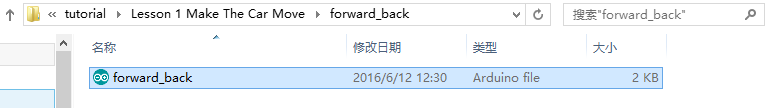
Below is the way how car moves:

|  |  |  |  |
| --- | --- | --- | --- |
| **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.**

******

**Code preview:**

*//www.kuongshun.com*

*//    Left motor truth table*

*//  ENA         IN1               IN2         Description*

*//  LOW   Not Applicable    Not Applicable    Motor is off*

*//  HIGH        LOW               LOW         Motor is stopped (brakes)*

*//  HIGH        HIGH              LOW         Motor is on and turning forwards*

*//  HIGH        LOW               HIGH        Motor is on and turning backwards*

*//  HIGH        HIGH              HIGH        Motor is stopped (brakes)*

*//    Right motor truth table*

*//  ENB         IN3             IN4         Description*

*//  LOW   Not Applicable   Not Applicable   Motor is off*

*//  HIGH        LOW             LOW         Motor is stopped (brakes)*

*//  HIGH        LOW             HIGH        Motor is on and turning forwards*

*//  HIGH        HIGH            LOW         Motor is on and turning backwards*

*//  HIGH        HIGH            HIGH        Motor is stopped (brakes)*

*//    The direction of the car's movement*

*//  Left motor    Right motor     Description*

*//  stop(off)     stop(off)       Car is stopped*

*//  forward       forward         Car is running forwards*

*//  forward       backward        Car is turning right*

*//  backward      forward         Car is turning left*

*//  backward      backward        Car is running backwards*

*//define the L298n IO pin*

#define ENA 5

#define ENB 6

#define IN1 7

#define IN2 8

#define IN3 9

#define IN4 11

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);

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, HIGH); *//go forward*

  delay(500);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW);

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, LOW); *//stop*

  delay(500);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, HIGH);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, LOW); *//go back*

  delay(500);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW);

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, HIGH); *//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 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 preview:**

void forward(){

  digitalWrite(ENA,HIGH); *//enable L298n A channel*

  digitalWrite(ENB,HIGH); *//enable L298n B channel*

  digitalWrite(IN1,HIGH); *//set IN1 hight level*

  digitalWrite(IN2,LOW); *//set IN2 low level*

  digitalWrite(IN3,LOW); *//set IN3 low level*

  digitalWrite(IN4,HIGH); *//set IN4 hight level*

  Serial.println("Forward");*//send message to serial monitor*

}

void back(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

  Serial.println("Back");

}

void left(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,HIGH);

  Serial.println("Left");

}

void right(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,HIGH);

  digitalWrite(IN2,LOW);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

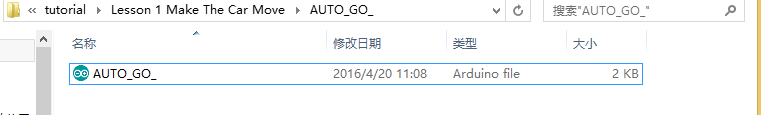
  Serial.println("Right");

}

### 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.**

****

**Code preview:**

*//www.kuongshun.com*

*//    The direction of the car's movement*

*//  ENA   ENB   IN1   IN2   IN3   IN4   Description*

*//  HIGH  HIGH  HIGH  LOW   LOW   HIGH  Car is runing forward*

*//  HIGH  HIGH  LOW   HIGH  HIGH  LOW   Car is runing back*

*//  HIGH  HIGH  LOW   HIGH  LOW   HIGH  Car is turning left*

*//  HIGH  HIGH  HIGH  LOW   HIGH  LOW   Car is turning right*

*//  HIGH  HIGH  LOW   LOW   LOW   LOW   Car is stoped*

*//  HIGH  HIGH  HIGH  HIGH  HIGH  HIGH  Car is stoped*

*//  LOW   LOW   N/A   N/A   N/A   N/A   Car is stoped*

*//define L298n module IO Pin*

#define ENA 5

#define ENB 6

#define IN1 7

#define IN2 8

#define IN3 9

#define IN4 11

void forward(){

  digitalWrite(ENA,HIGH); *//enable L298n A channel*

  digitalWrite(ENB,HIGH); *//enable L298n B channel*

  digitalWrite(IN1,HIGH); *//set IN1 hight level*

  digitalWrite(IN2,LOW); *//set IN2 low level*

  digitalWrite(IN3,LOW); *//set IN3 low level*

  digitalWrite(IN4,HIGH); *//set IN4 hight level*

  Serial.println("Forward");*//send message to serial monitor*

}

void back(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

  Serial.println("Back");

}

void left(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,HIGH);

  Serial.println("Left");

}

void right(){

  digitalWrite(ENA,HIGH);

  digitalWrite(ENB,HIGH);

  digitalWrite(IN1,HIGH);

  digitalWrite(IN2,LOW);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

  Serial.println("Right");

}

*//before execute loop() function,*

*//setup() function will execute first and only execute once*

void setup() {

  Serial.begin(9600);*//open serial and set the baudrate*

  pinMode(IN1,OUTPUT);*//before useing io pin, pin mode must be set first*

  pinMode(IN2,OUTPUT);

  pinMode(IN3,OUTPUT);

  pinMode(IN4,OUTPUT);

  pinMode(ENA,OUTPUT);

  pinMode(ENB,OUTPUT);

}

*//Repeat execution*

void loop() {

  forward(); *//go forward*

  delay(1000);*//delay 1000 ms*

  back(); *//go back*

  delay(1000);

  left(); *//turning left*

  delay(1000);

  right(); *//turning right*

  delay(1000);

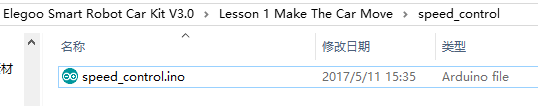
}

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.**

****

**Code preview:**

*//www.kuongshun.com*

#define ENA 5

#define ENB 6

#define IN1 7

#define IN2 8

#define IN3 9

#define IN4 11

void setup() {

  pinMode(IN1,OUTPUT);

  pinMode(IN2,OUTPUT);

  pinMode(IN3,OUTPUT);

  pinMode(IN4,OUTPUT);

  pinMode(ENA,OUTPUT);

  pinMode(ENB,OUTPUT);

}

void loop() {

*//go forward*

  digitalWrite(IN1,HIGH);

  digitalWrite(IN2,LOW);

  digitalWrite(IN3,LOW);

  digitalWrite(IN4,HIGH);

*//reduce the speed*

  for(int i = 255; i >= 0; i--){

    analogWrite(ENB,i);

    analogWrite(ENA,i);

    delay(20);

  }

*//stop*

  analogWrite(ENB,0); *//speed = 0*

  analogWrite(ENA,0);

  delay(1000);

*//runing back*

  digitalWrite(IN1,LOW);

  digitalWrite(IN2,HIGH);

  digitalWrite(IN3,HIGH);

  digitalWrite(IN4,LOW);

*//accelerate*

  for(int i = 0; i <= 255; i++){

    analogWrite(ENB,i);

    analogWrite(ENA,i);

    delay(20);

  }

*//stop*

  digitalWrite(ENB,LOW); *//Motor is off*

  digitalWrite(ENA,LOW);

  delay(2000);

}