Basic course ---2.Car advance

Note:

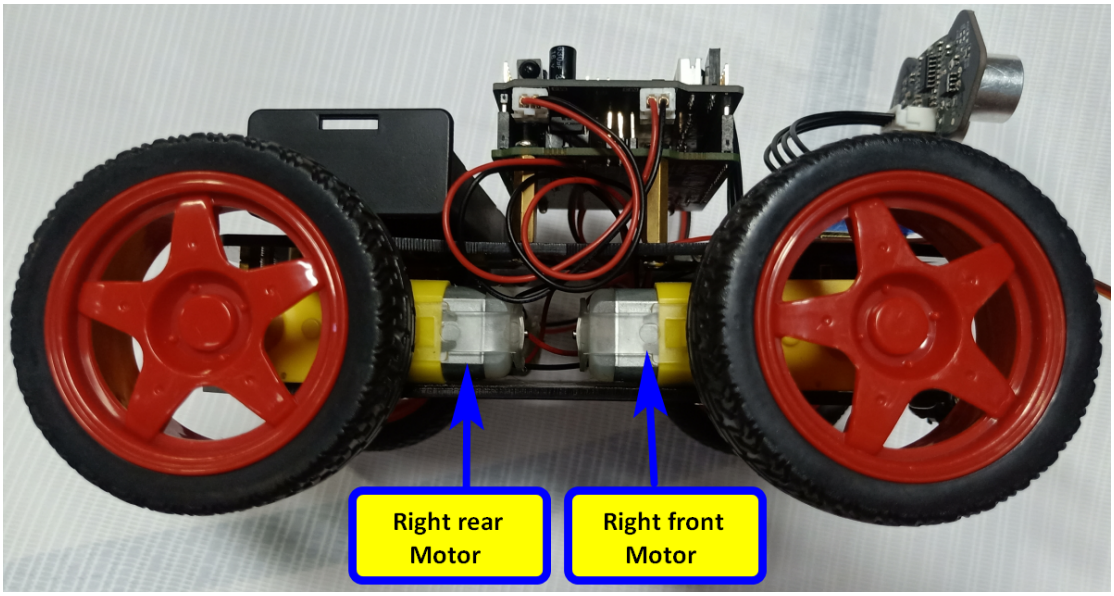
Because the battery has over-current protection, the speed of the car should not exceed 150 during using, it is fast enough.

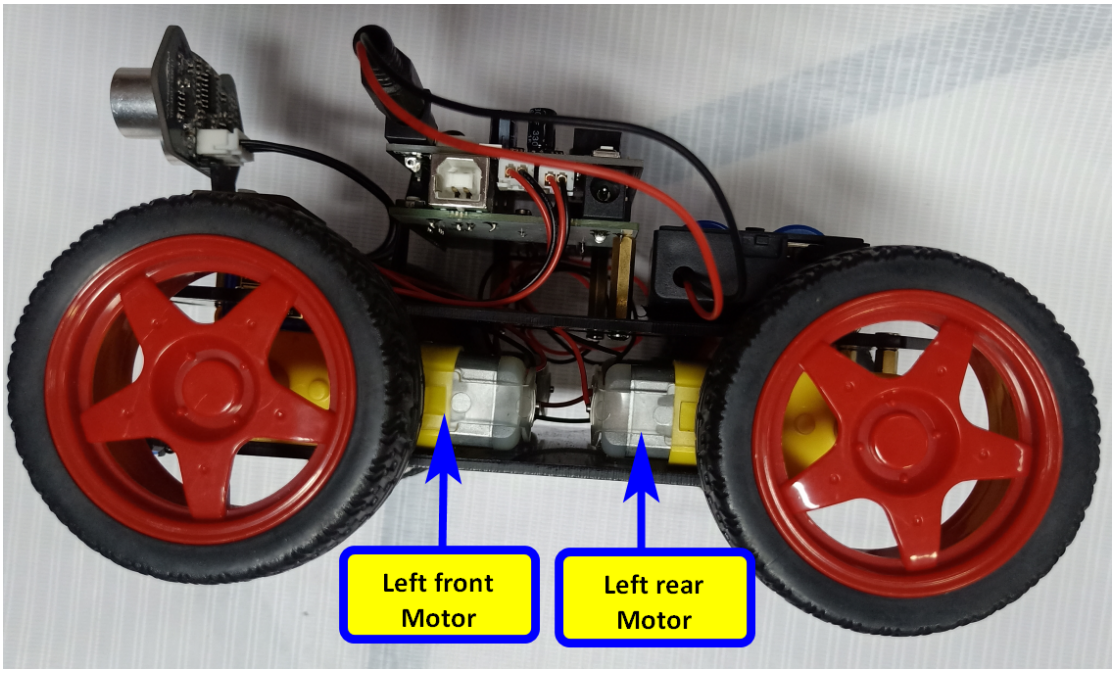
1. Learning goal

In this lesson, we will learn how to control the motor of the car.

2. Preparation

2.1 The position of the motor on the robot car. As shown below.





2.2 The pin of UNO board is connected the motor.

3. Principle of experimental

From the hardware interface manual, we can know that four motor of the robot car are driven by PWM by PCA9685 chip.

Classification	Function	The number of Drive chip PCA9685	Drive Method	Connection with CPU	Uno board
Left Motor	Left front motor forward	LINB(13)	PCA9685	I2C_SDA/I2C_SCL	A4/A5
	Left front motor reverse	LINA(12)			
	Left rear motor forward	RINB(15)			
	Left rear motor reverse	RINA(14)			
Right Motor	Right front motor forward	LED10			
	Right front motor reverse	LED11			
	Right rear motor forward	LED8			
	Right rear motor reverse	LED9			
Servo	Control S1	LED0			
	Control S2	LED1			
	Control S3	LED2			
	Control S4	S1(3)			
LOGO light	Control bluelight	LED7			
Tarcking sensor	Left tracking sensor	Uno board drive directly			A0
	Middle tracking sensor				A1
	Right tracking sensor				A2
Ultrasonic sensor	Ultrasonic Echo				12
	Ultrasonic RGB light				11
Key	K1				7
IR	IR control				A3
Bluetooth interface	RX				0
	TX				1
On board RGB Light	RGB Light on expansion board				6
Buzzer	Control buzzer				10

PCA9685 chip and UNO board adopt I2C communication method. I2C address is 0x40.

connection as shown below.

PCA9685 chip	Right front Motor
LED10	Forward
LED11	Reverse

PCA9685 chip	Right rear Motor
LED8	Forward
LED9	Reverse

PCA9685 chip	Left front Motor
LINB(13)	Forward
LINA(12)	Reverse

PCA9685 chip	Left rear Motor

RINB(15)	Forward
RINA(14)	Reverse

4. About code

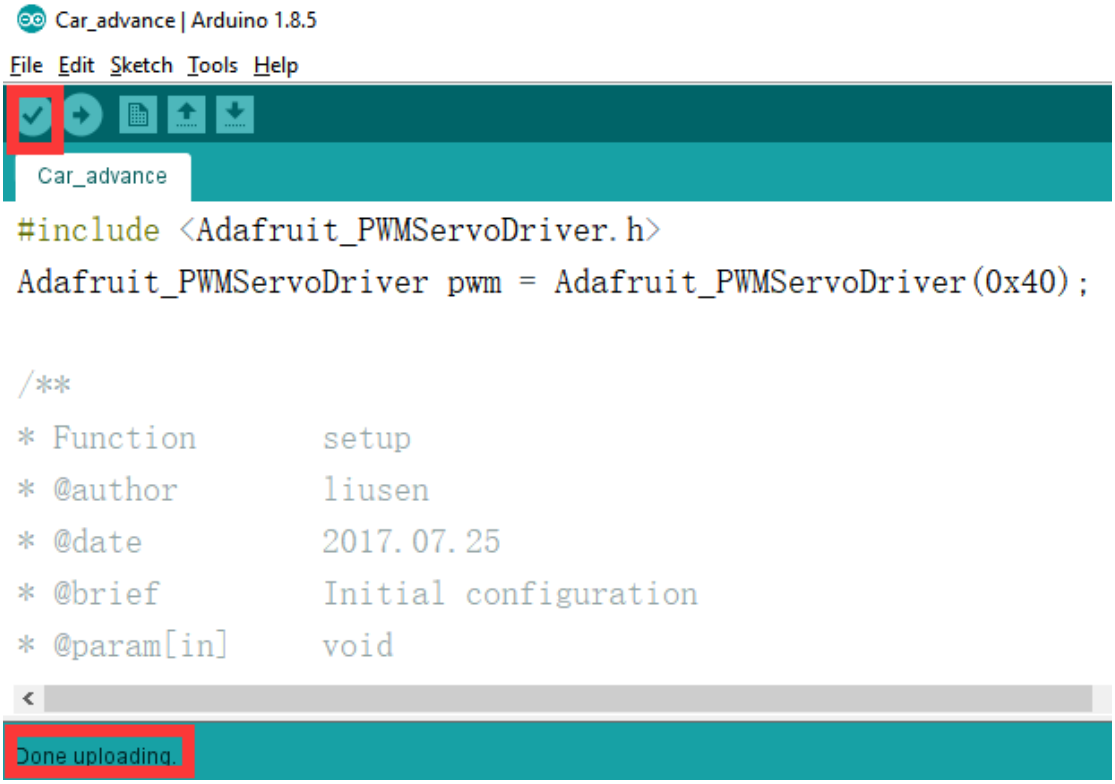
For the code of this course, please refer to: **Car_advance.ino** in the **Car_advance** folder

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

void setup()
{
    pwm.begin();
    pwm.setPWMFreq(60);    // Analog servos run at ~60 Hz updates
    LOGO_breathing_light(255, 40, 5);     //Gradually light the blue light of the
Yhaboom_LOGO
}

void LOGO_breathing_light(int brightness, int time, int increament)
{
  if (brightness < 0)
  {
    brightness = 0;
  }
  if (brightness > 255)
  {
    brightness = 255;
  }
  for (int b = 0; b < brightness; b += increament)
  {
    int newb = map(b, 0, 255, 0, 4095);
    pwm.setPWM(7, 0, newb);
    delay(time);
  }
}

void run(int Speed)
{
  Speed = map(Speed, 0, 255, 0, 4095);
  pwm.setPWM(10, 0, Speed); //Right front wheel Forward
  pwm.setPWM(11, 0, 0);
  pwm.setPWM(8, 0, Speed); //Right rear wheel Forward
  pwm.setPWM(9, 0, 0);
```

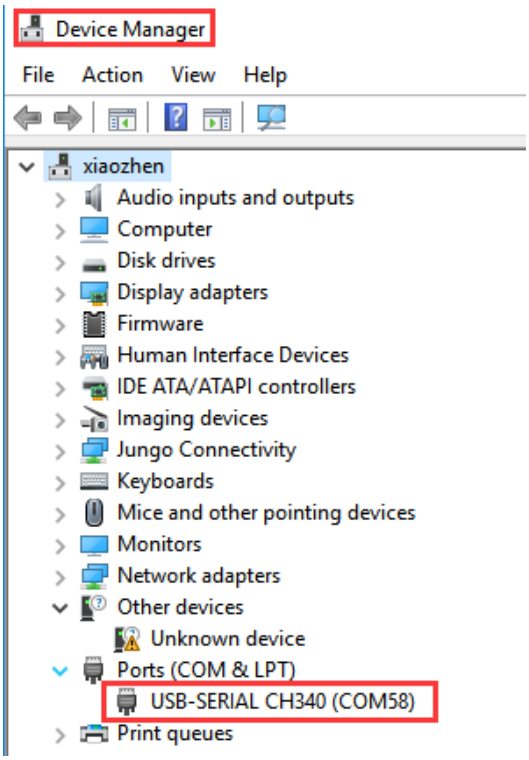
```
    pwm.setPWM(13, 0, Speed); //Left front wheel Forward
    pwm.setPWM(12, 0, 0);
    pwm.setPWM(15, 0, Speed); //Left rear wheel Forward
    pwm.setPWM(14, 0, 0);
}

void loop()
{
      delay(1000);
      while(1)
      {
          run(150);       //This speed can be adjusted by yourslef
      }
}
```

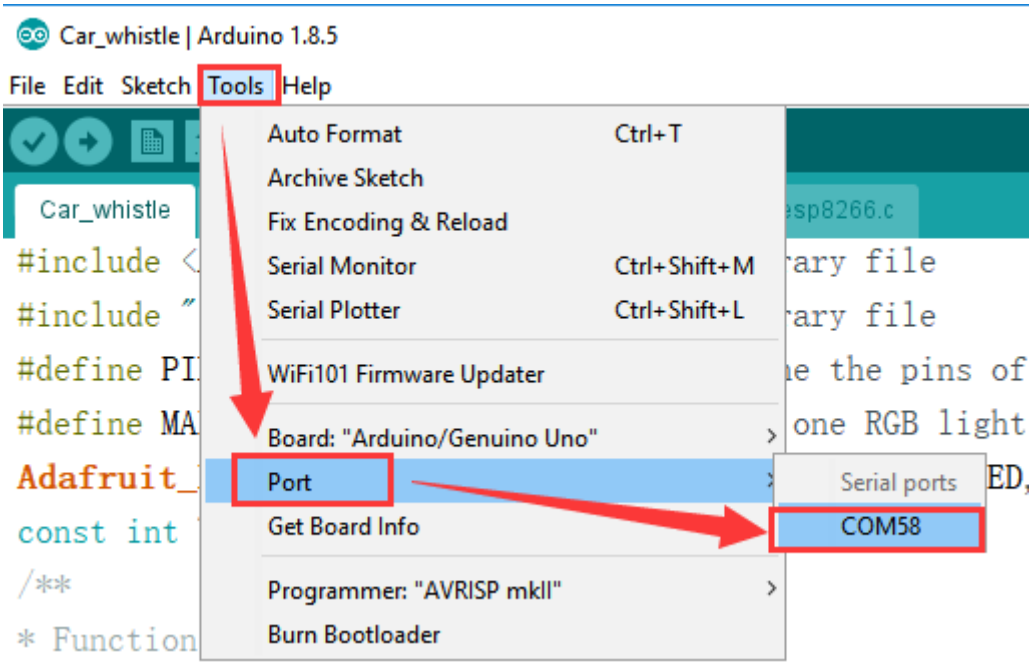
5. Compiling and downloading code

5.1 We need to open the **Car_advance.ino** file by Arduino IDE software. Then click"√"under the menu bar to compile the code, and wait for the word "**Done compiling** " in the lower right corner, as shown in the figure below.

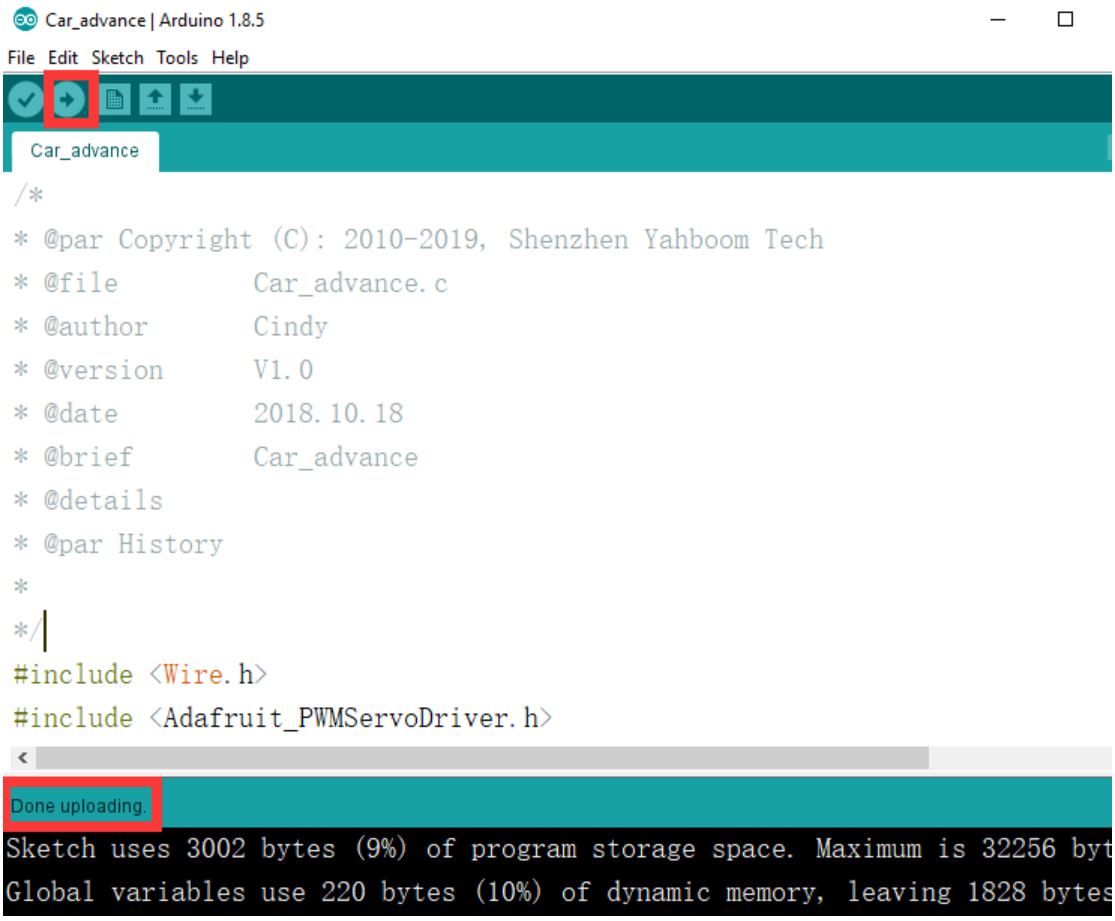


5.2 In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】---selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





5.3 After the selection is completed, you need to click "→"under the menu bar to upload the code to the UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the UNO board, as shown in the figure below.



6. Experimental phenomena

After the program is downloaded, after 1s, we can see that the car will advance and keep looping in this state.