2019

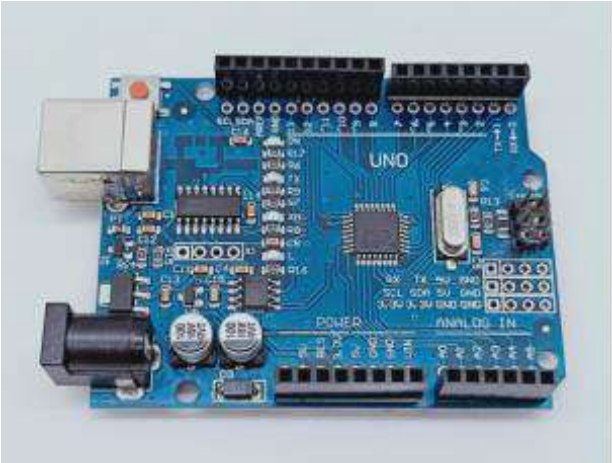
Yahboom Arduino 4WD Robot

Arduino IDE Programming Tutorials

C programming

1- Color LED

1) Preparation



1-1 Arduino UNO board

1-2 RGB module

2) Purpose of Experimental

After the code upload is completed, the delay is 2s, and the lights of 7 different colors are displayed cyclically.

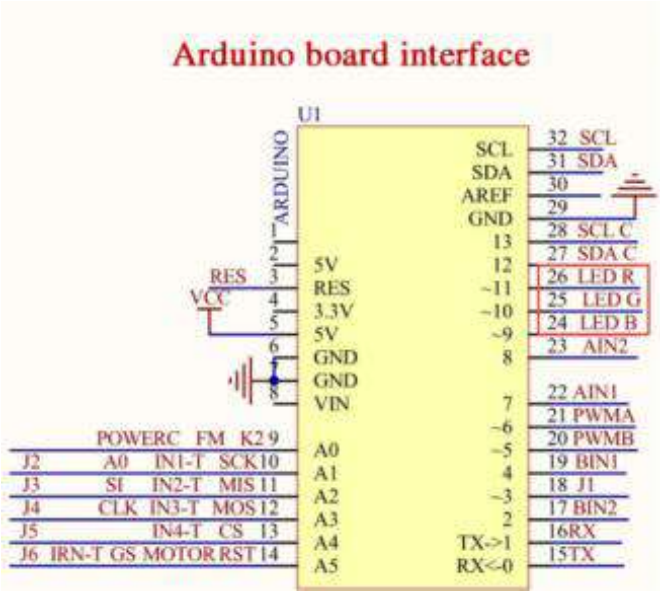
3) Principle of experimental

3 LEDs (red, green, blue) are packaged in the RGB lamp module. We can mix different colors(256*256*256) by controlling the brightness of the three LEDs.

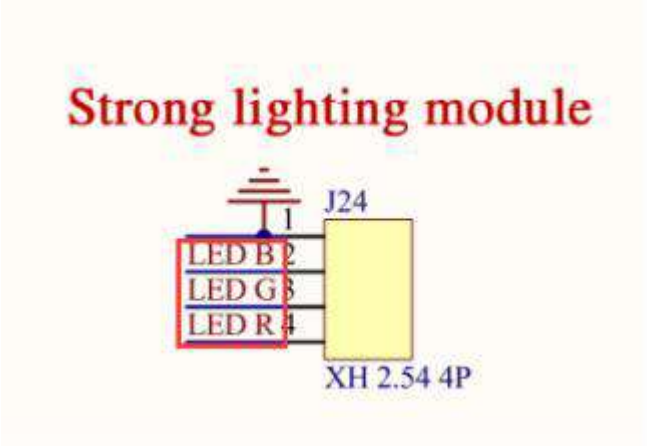
According to the circuit schematic, the RGB lamp is a common cathode LED, one pin is connect to GND, and the remaining three pins are respectively connected to the port 11, 10, 9 on the Arduino UNO board. Each LED needs to be connected in series with a 220Ω resistor as the current limiting resistor. We can control the LED by controlling the corresponding pin to be high level of Arduino UNO board.

4) Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 RGB module interface circuit diagram

4-2 According to the circuit schematic:

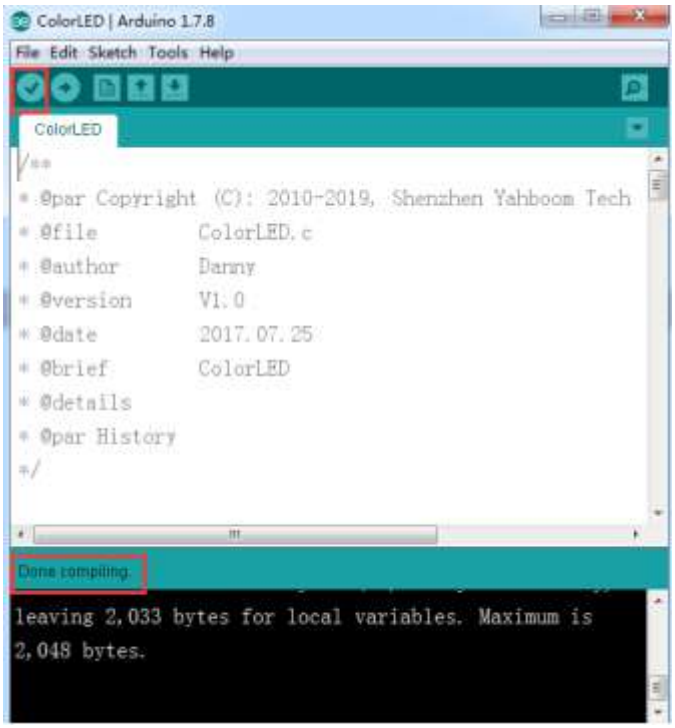
LED_R-----11(Arduino UNO)

LED_G-----10(Arduino UNO)

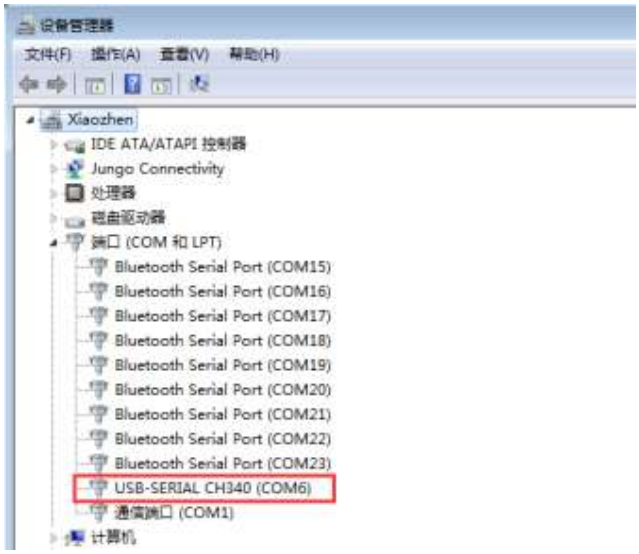
LED_B-----9(Arduino UNO)

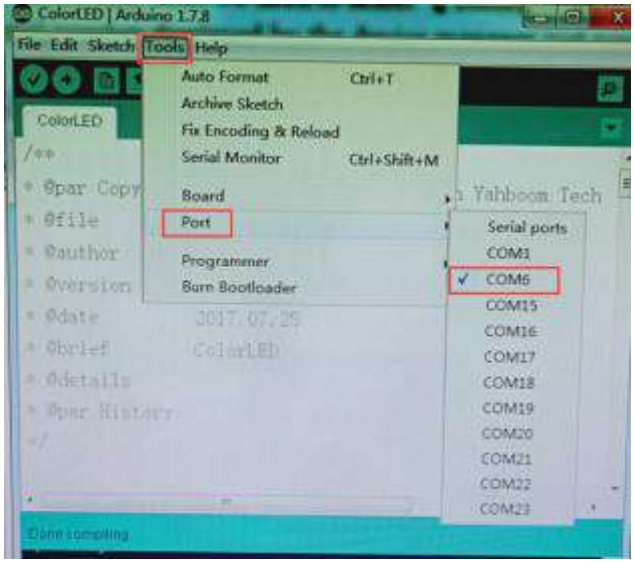
4-3 About the code

1. We need to open the code of this experiment: **ColorLED.ino**, 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.

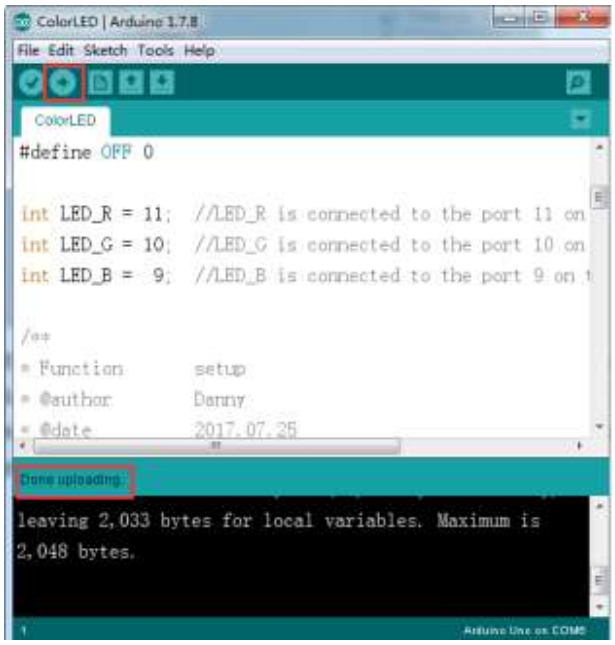


2. In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



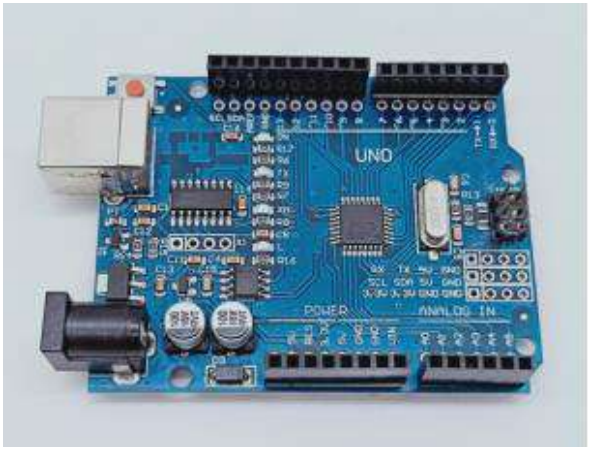


3. After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



2- Advance

1) Preparation



1-1 Arduino UNO board

1-2 4 motors

2) Purpose of Experimental

After the code upload is completed, the delay is 2s, advance 1s.

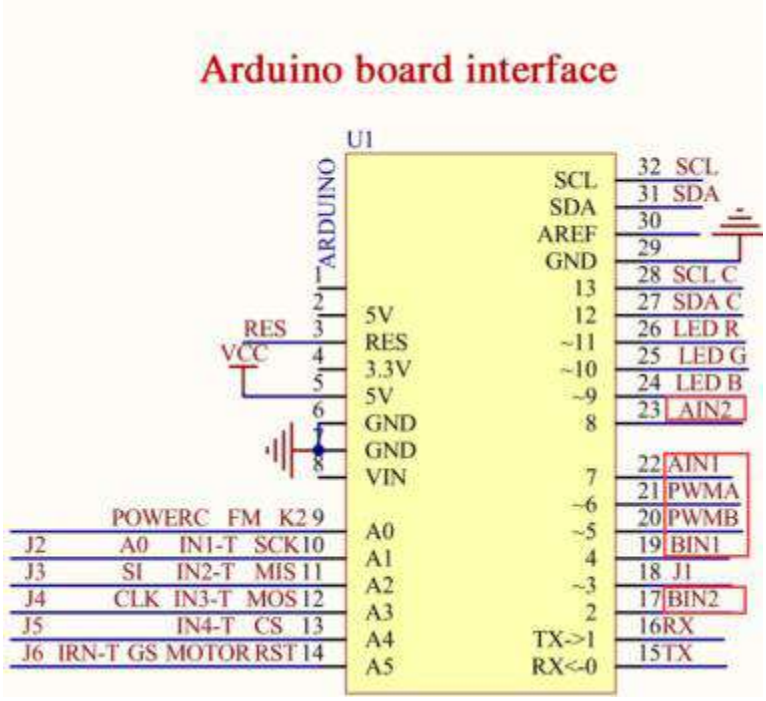
3) Principle of experimental

We use the TB6612FNG driver chip to drive the motor. Control the forward, reverse, and stop of the motor by controlling the level of AIN1, AIN2, BIN1, BIN2, PWMA, and PWMB of the driver chip.

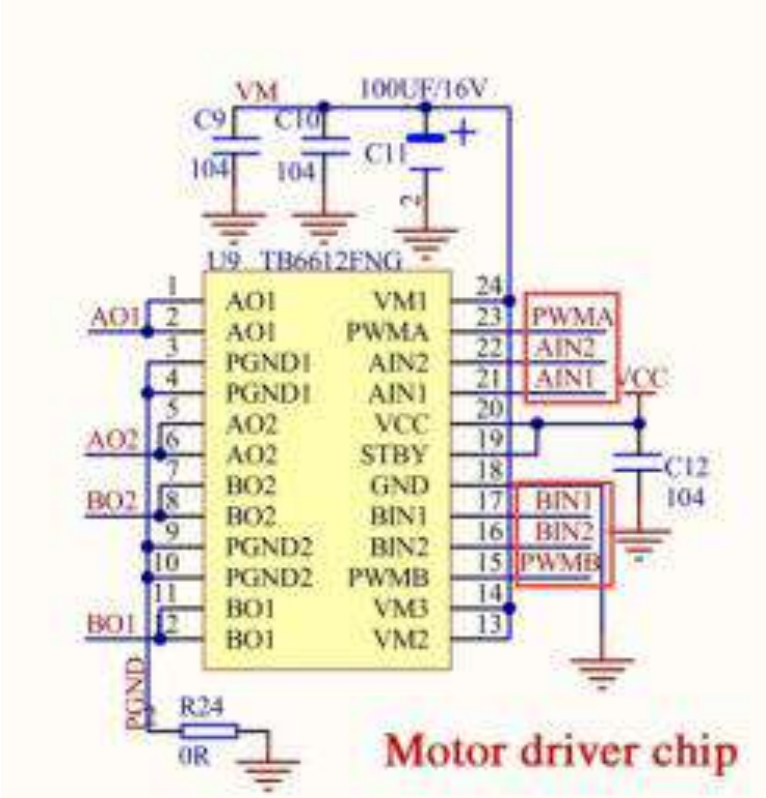
In this experiment, we control robot car advance by controlling AIN1 to be high level, AIN2 is low level, BIN1 is high level, BIN2 is low level. And we control speed of the robot car by controlling PWMA, PWMB(0-255). One-channel PWM control the speed of the motor on one side of the robot car.

4) Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 Motor drive chip---TB6612FNG

4-2 According to the circuit schematic:

AIN1-----7(Arduino UNO)

AIN2-----8(Arduino UNO)

PWMA-----6(Arduino UNO)

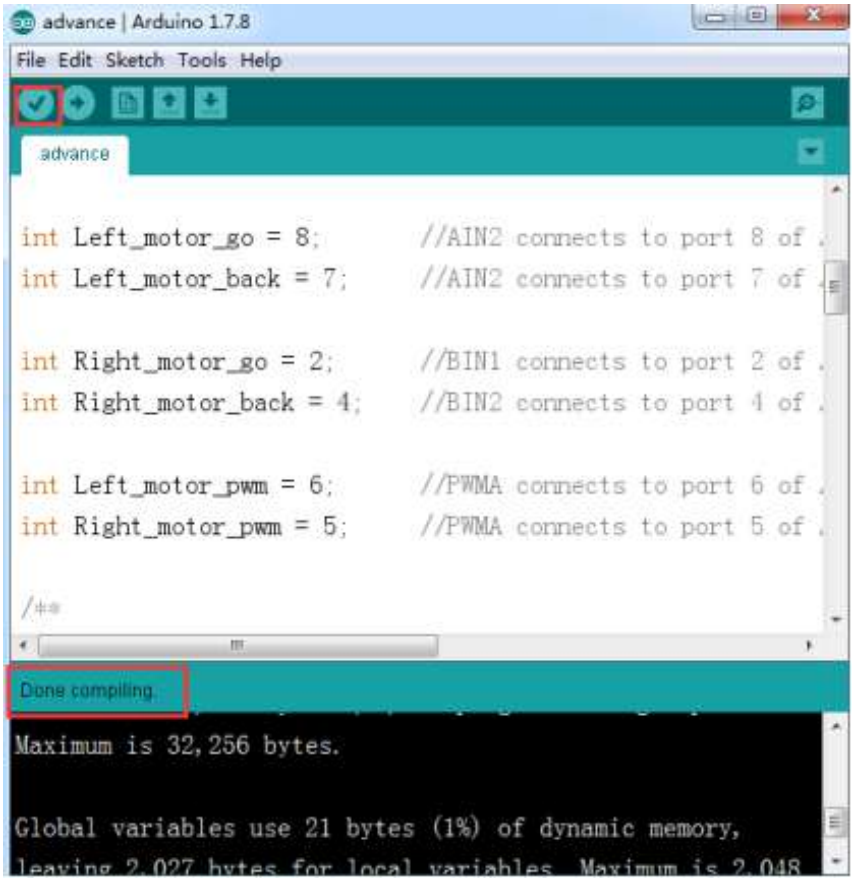
BIN1-----4(Arduino UNO)

BIN2-----2(Arduino UNO)

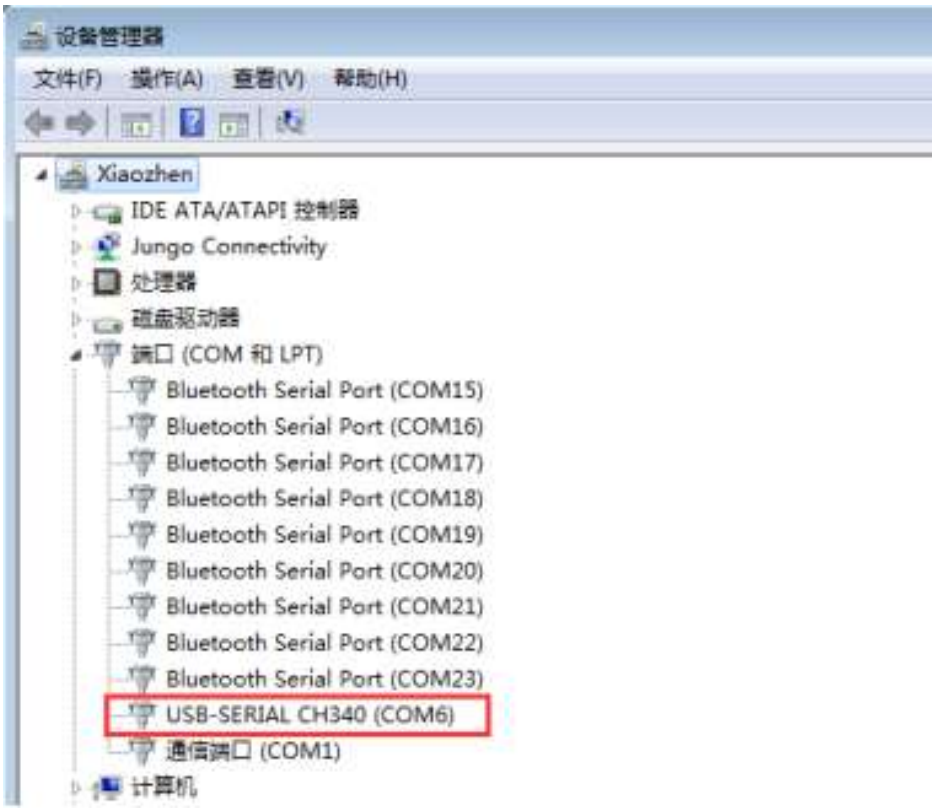
PWMB-----5(Arduino UNO)

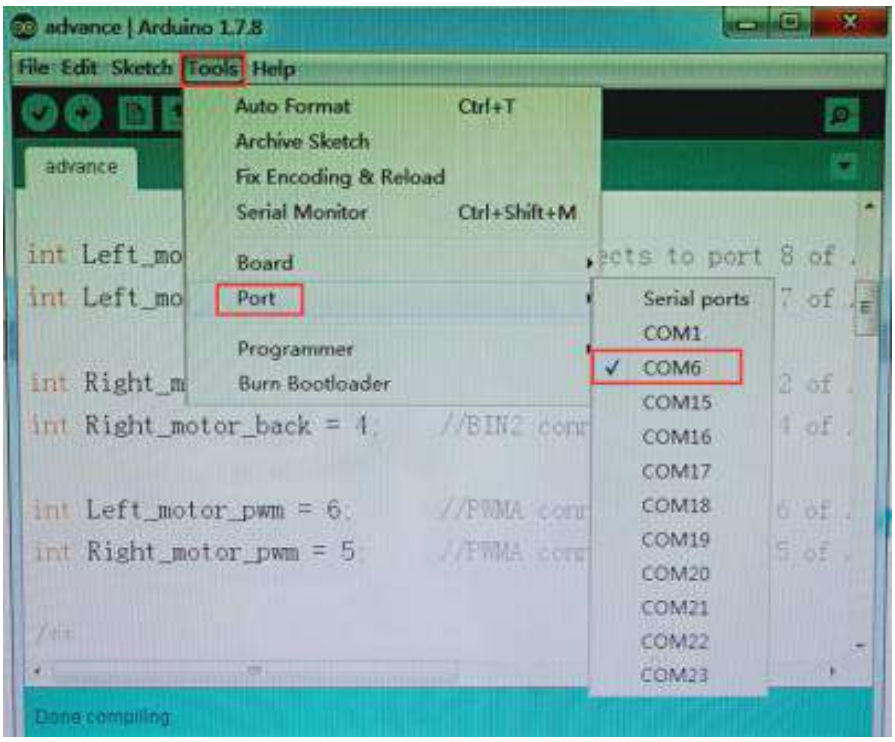
4-3 About the code

1. We need to open the code of this experiment: **advance.ino**, 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.

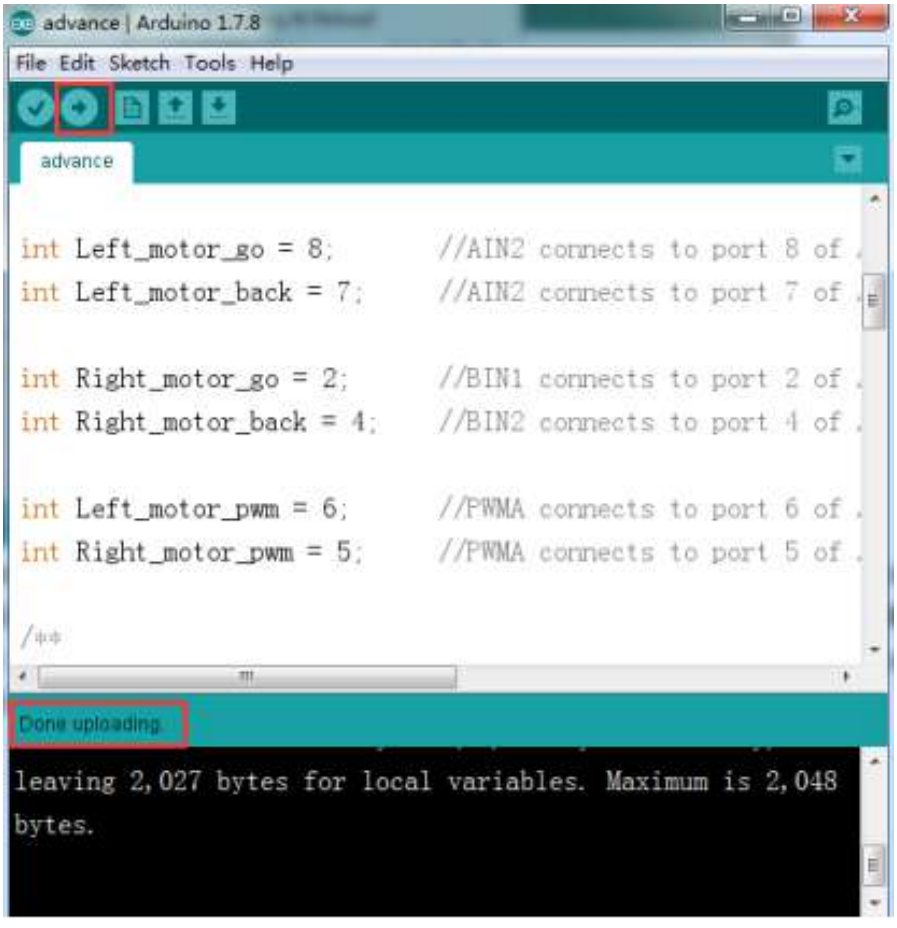


2. In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



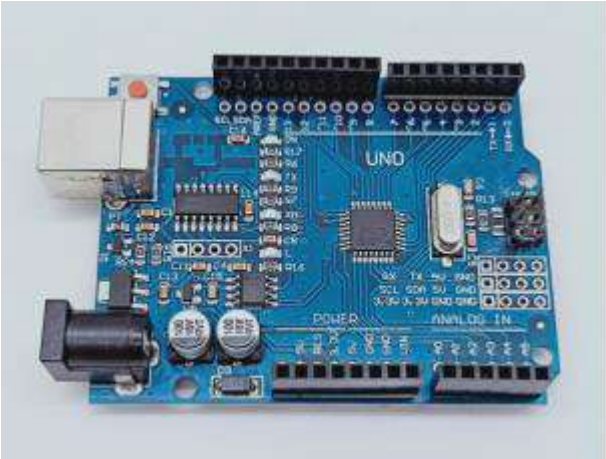


3. After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



3- CarRun

1) Preparation



1-1 Arduino UNO board

1-2 4 motors

2) Purpose of Experimental

After the code upload is completed, the robot car will delay is 2s, advance 1s, back 1s, turn left 2s, turn right 2s, turn left in place 3s, turn right in place 3s, stop 0.5s. And it goes on and on.

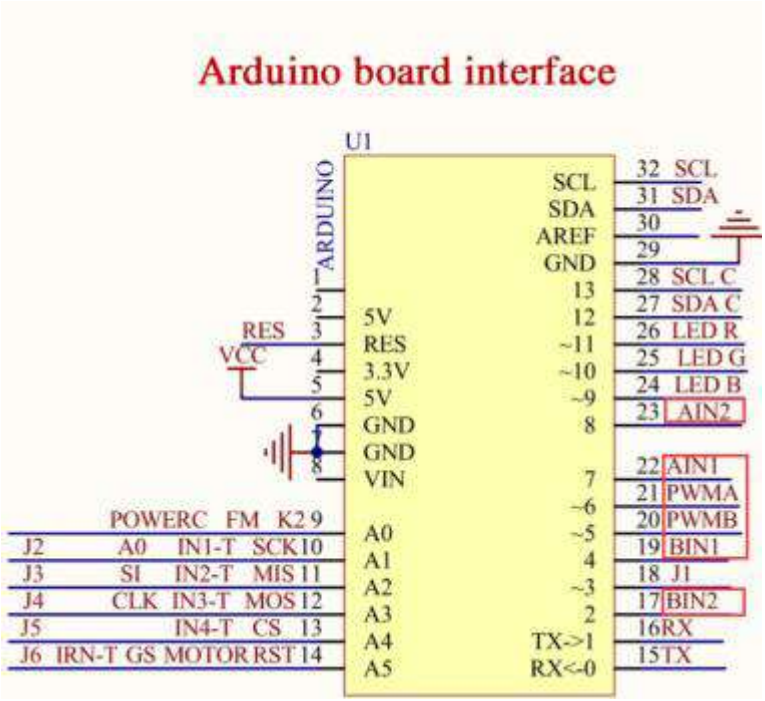
3) Principle of experimental

We use the TB6612FNG driver chip to drive the motor. Control the forward, reverse, and stop of the motor by controlling the level of AIN1, AIN2, BIN1, BIN2, PWMA, and PWMB of the driver chip.

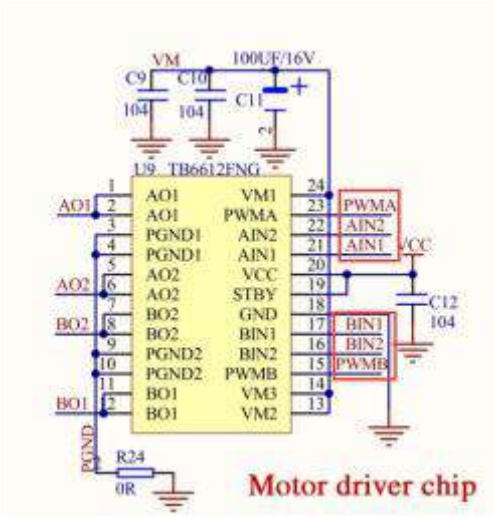
In this experiment, we control robot car advance by controlling AIN1 to be high level, AIN2 is low level, BIN1 is high level, BIN2 is low level. And we control speed of the robot car by controlling PWMA, PWMB(0-255). One-channel PWM control the speed of the motor on one side of the robot car.

4) Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 Motor drive chip---TB6612FNG

4-2 According to the circuit schematic:

AIN1-----7(Arduino UNO)

AIN2-----8(Arduino UNO)

PWMA-----6(Arduino UNO)

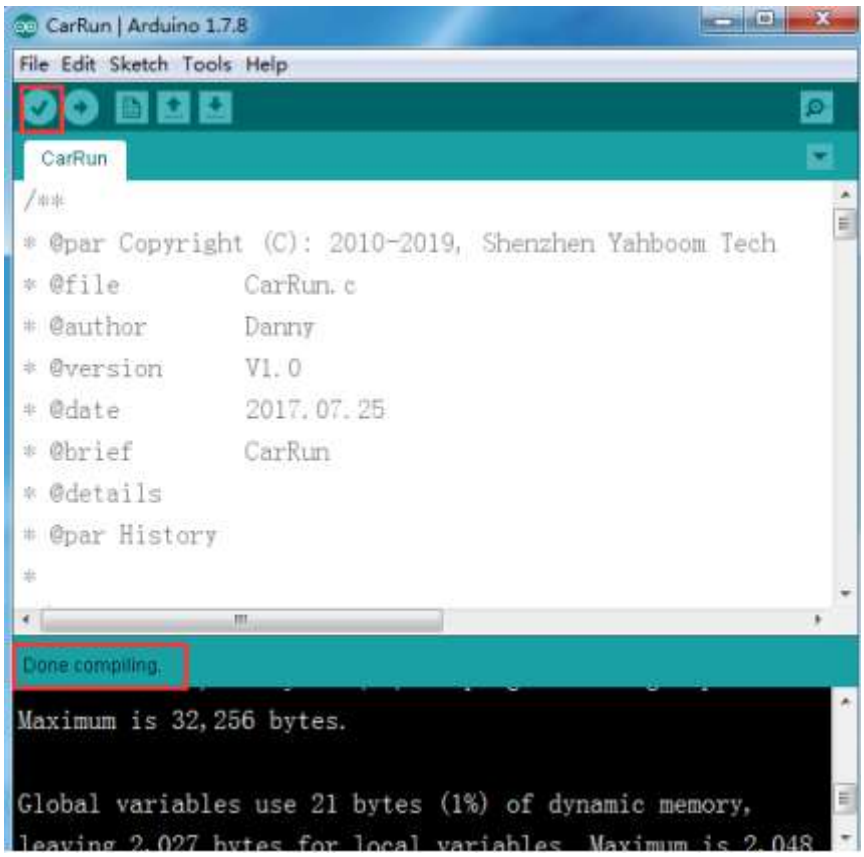
BIN1-----4(Arduino UNO)

BIN2-----2(Arduino UNO)

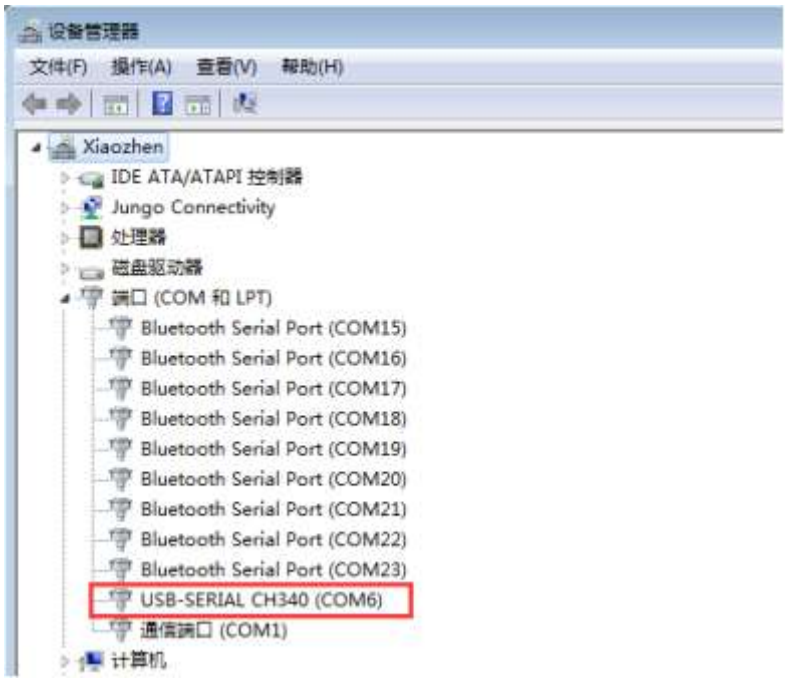
PWMB-----5(Arduino UNO)

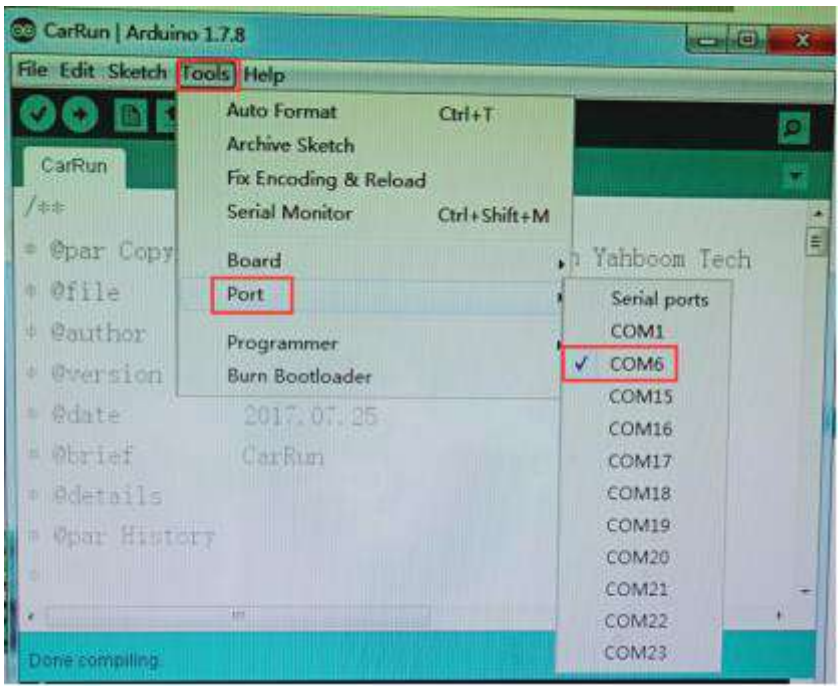
4-3 About the code

1. We need to open the code of this experiment: **CarRun.ino**, 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.

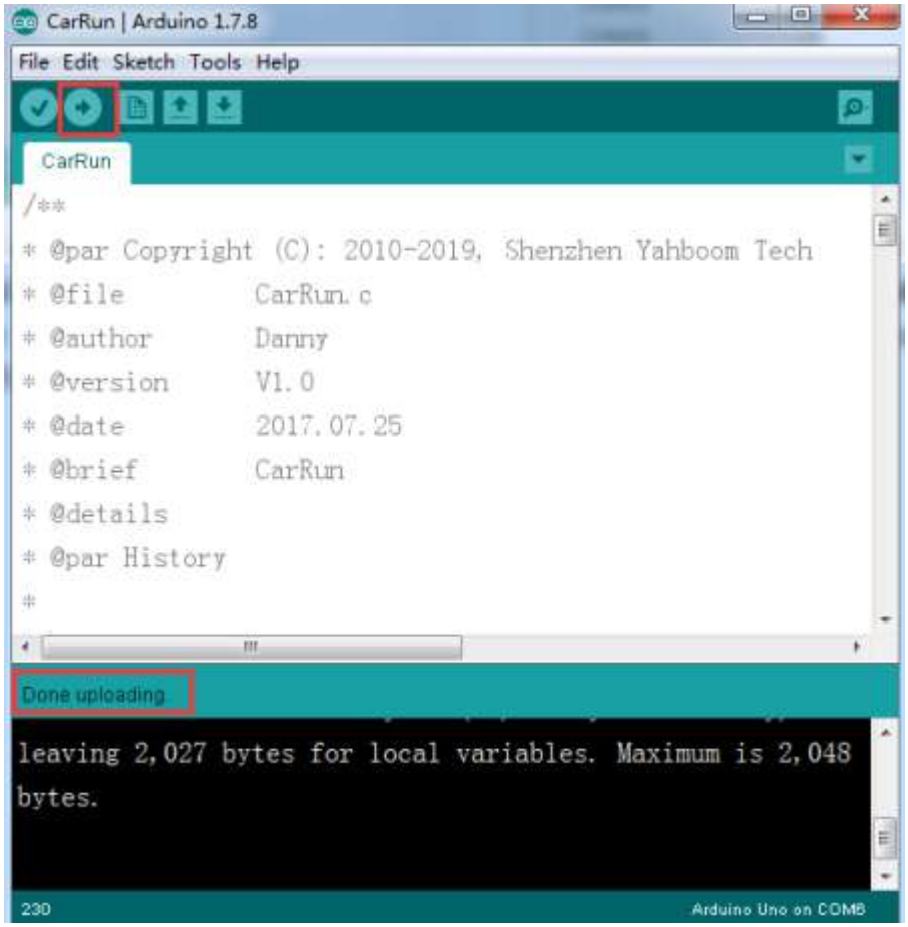


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



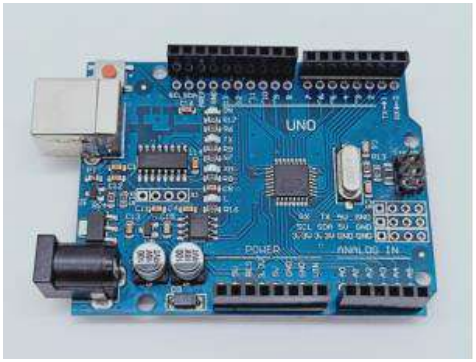


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



4- ServoControlColor

1) Preparation



1-1 Arduino UNO board

1-2 SG90 servo

1-3 RGB module

2)Purpose of Experimental

After the code upload is completed, the car will still for 0.5 s. The servo starts to turn and lights up different colors when turning to different angles.

3)Principle of experimental

The working principle of the servo: the control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor.

Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

0.5ms----------0°

1.0ms----------45°

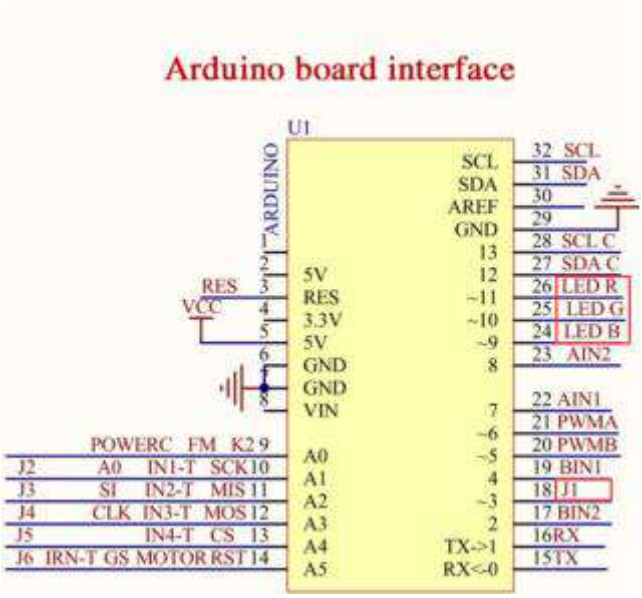
1.5ms----------90°

2.0ms----------135°

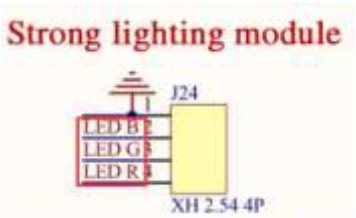
2.5ms----------180°

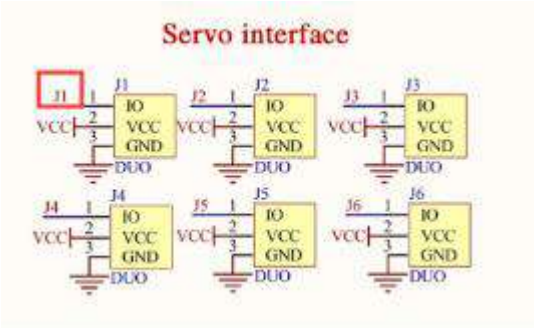
4)Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram





4-2 RGB module interface 4-3 Servo interface

4-2 According to the circuit schematic:

LED_R-----11(Arduino UNO)

LED_G-----10(Arduino UNO)

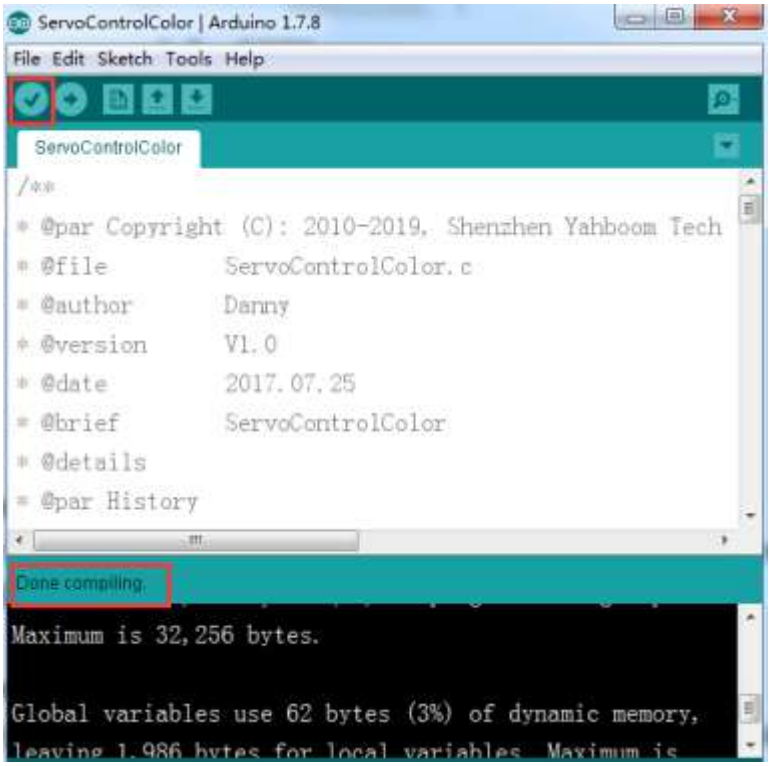
LED_B-----9(Arduino UNO)

J1---3(Arduino UNO)

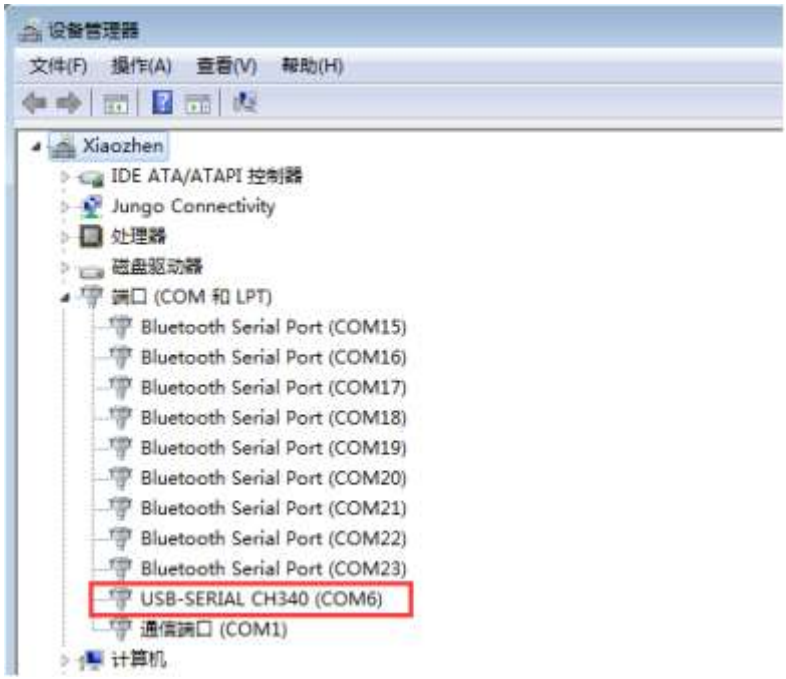
(Note: SG90 Servo is be connected to steering gear interface J1)

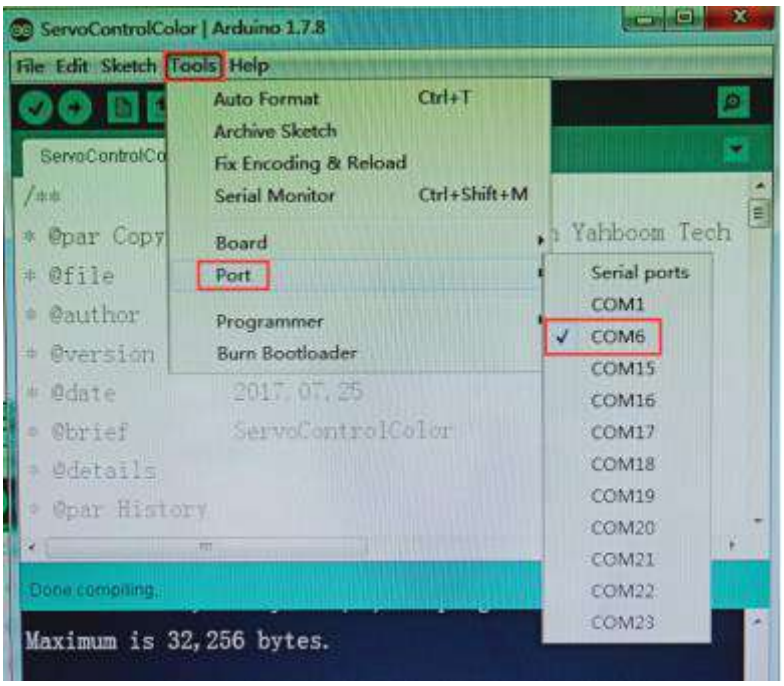
4-3 About the code

1. We need to open the code of this experiment: **ServoControlColor.ino**, 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.

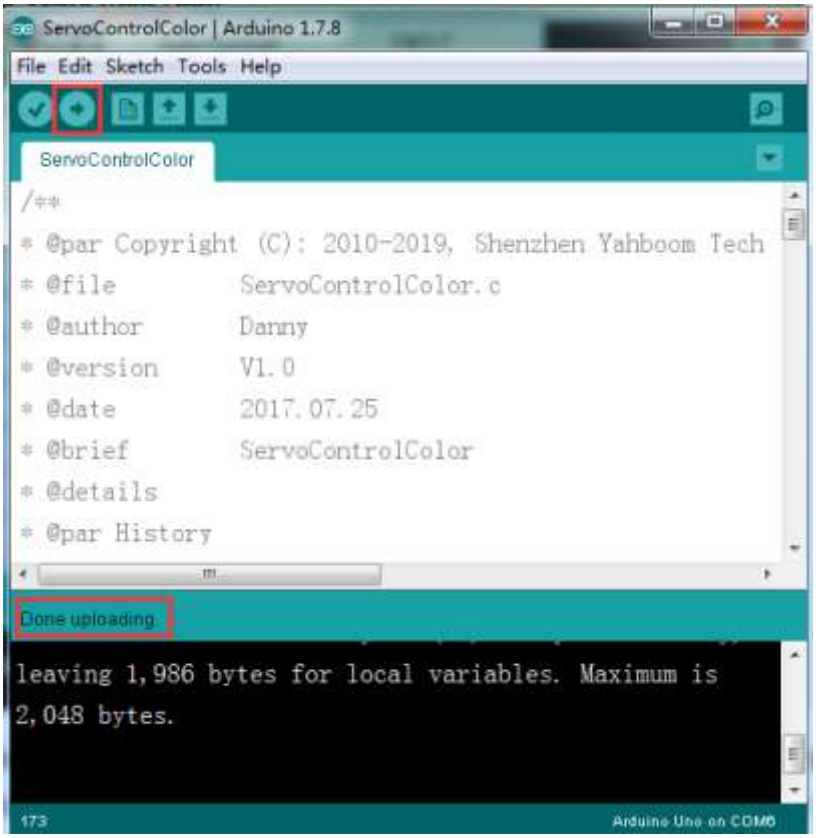


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



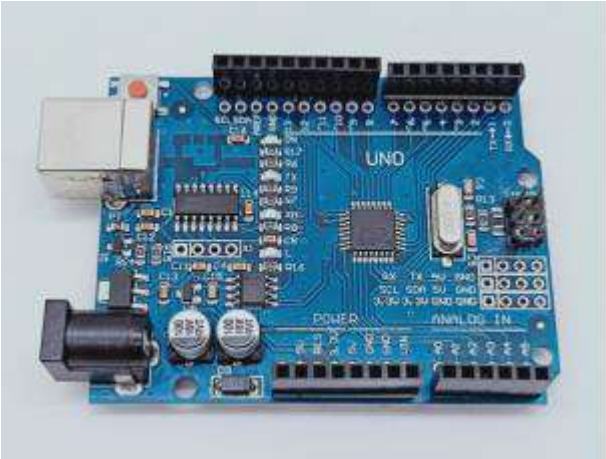


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



5- KeyScanStart

1) Preparation



1-1 Arduino UNO board

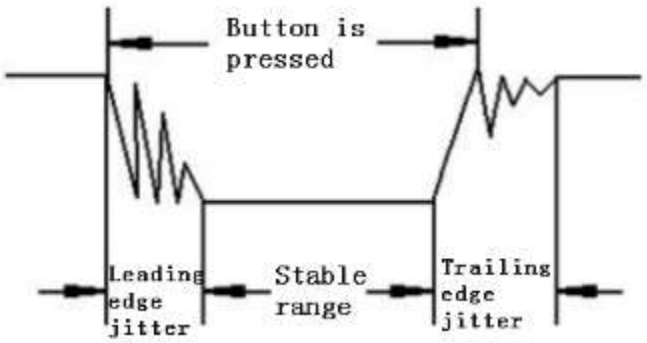
1-2 Key

2)Purpose of Experimental

After the code upload is completed, You need to press the KEY to start the car, the car will advance 1 s ,back 1 s,turn left 2 s,turn right 2 s, turn left in place 3 s, turn right in place 3 s, stop 0.5s.

3）Principle of experimental

Generally, our button switches are mechanical elastic switches. When the mechanical contacts are opened and closed, due to the elastic action of the mechanical contactor, switchs will not be able to be connected immediately when closed and it will not be disconnected at once.



3-1 Button jitter state diagram

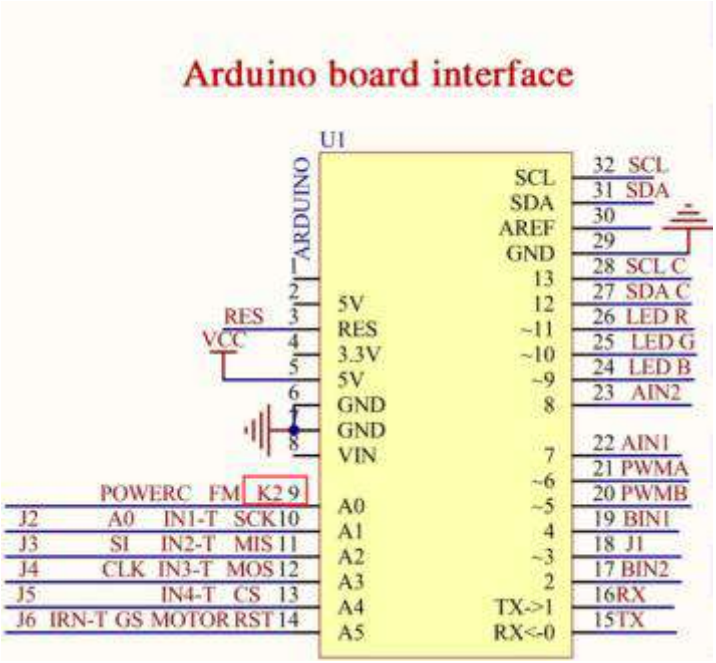
The jitter time is usually within 10ms. The button must be eliminated jitter toensure that the program only responds once after the button is closed once.

In this experiment, we took the software delay eliminated jitter. After detecting that the button is closed, delay codes is executed to generate a delay of 5ms to 10ms, and

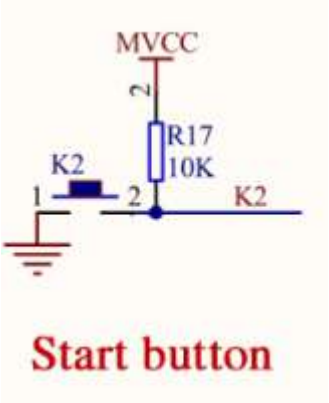
the state of the button is detected again after the leading edge jitter disappears. When it is detected that the button is released, it also needs to delay 5ms~10ms.

4)Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



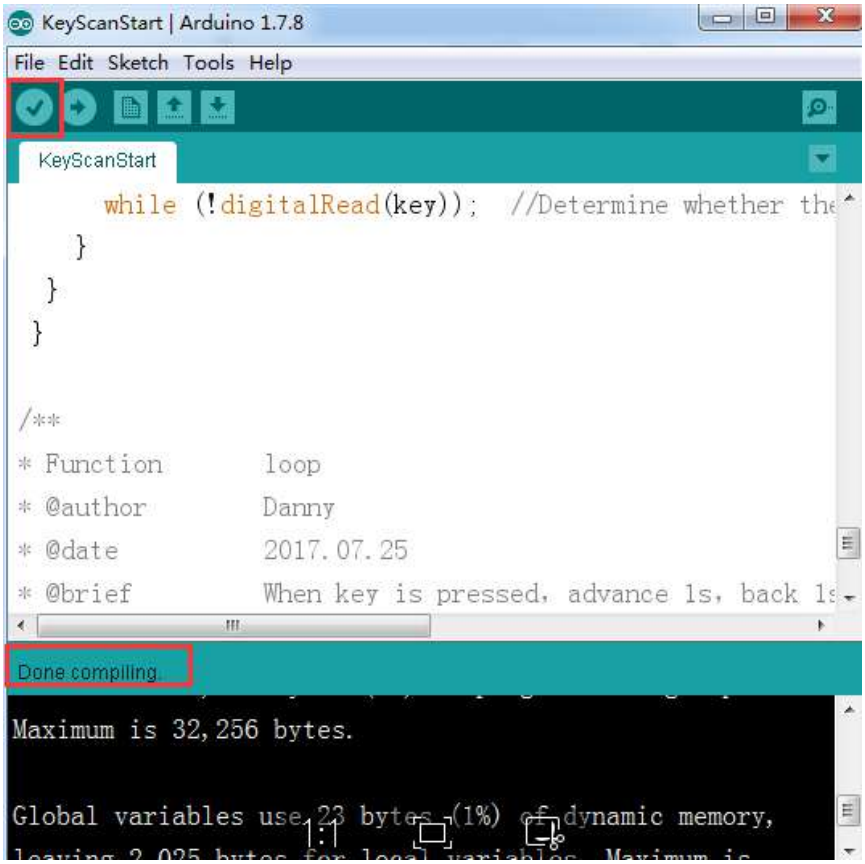
4-2 key circuit diagram

4-2 According to the circuit schematic:

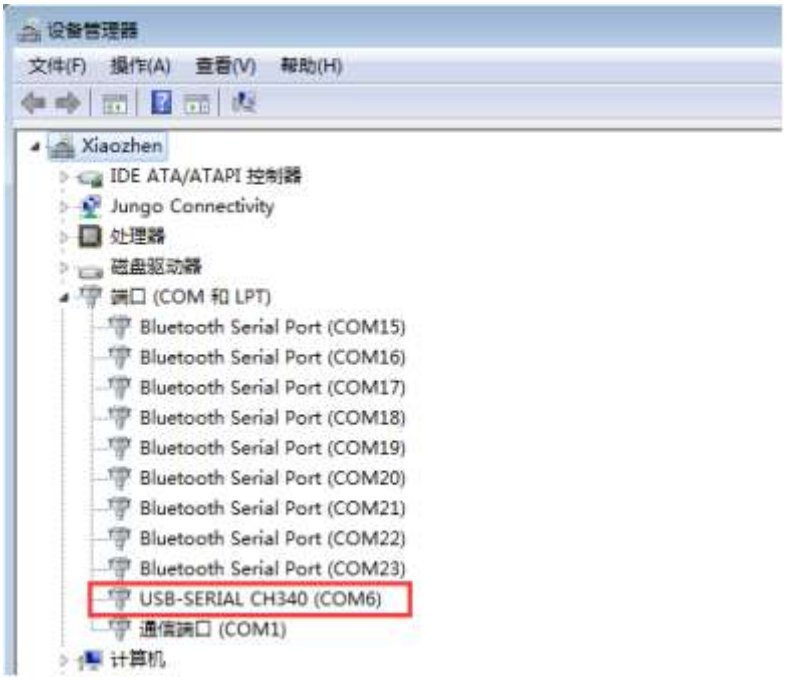
Key-----A0(Arduino UNO)

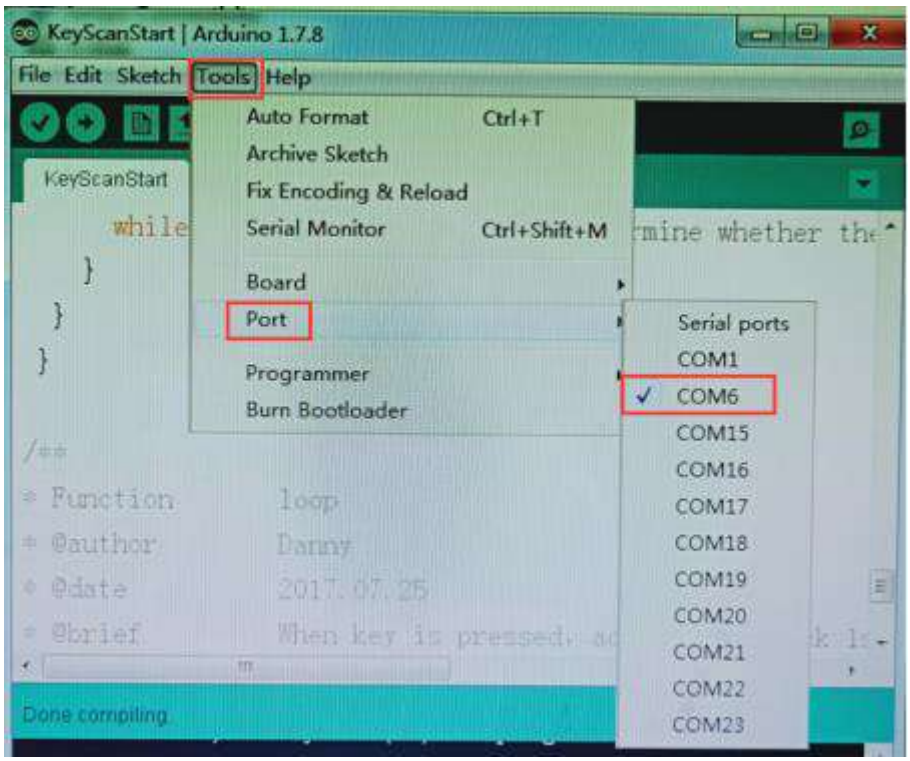
4-3 About the code

1. We need to open the code of this experiment:**KeyScanStart.ino**, 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 blew.

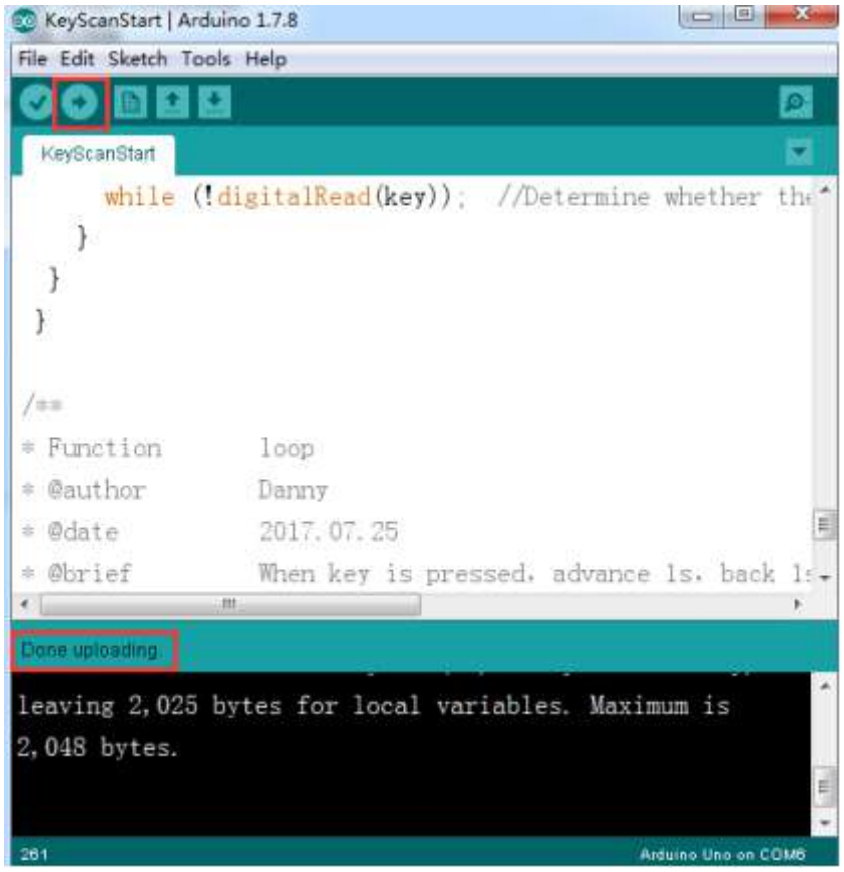


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



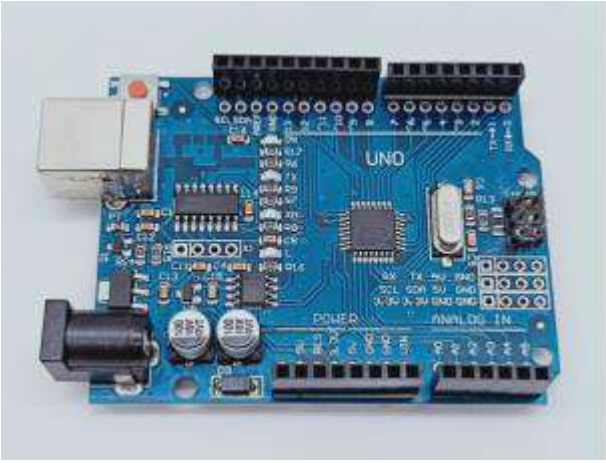


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



6- Infrared_avoid

1)Preparation



1-1 Arduino UNO board

1-2 Infrared obstacle avoidance module

2)Purpose of Experimental

After the code upload is completed. You need to press the K2 to start the car, and the infrared obstacle avoidance function is started. When there is an obstacle in front, the car can avoid the obstacle automatically.

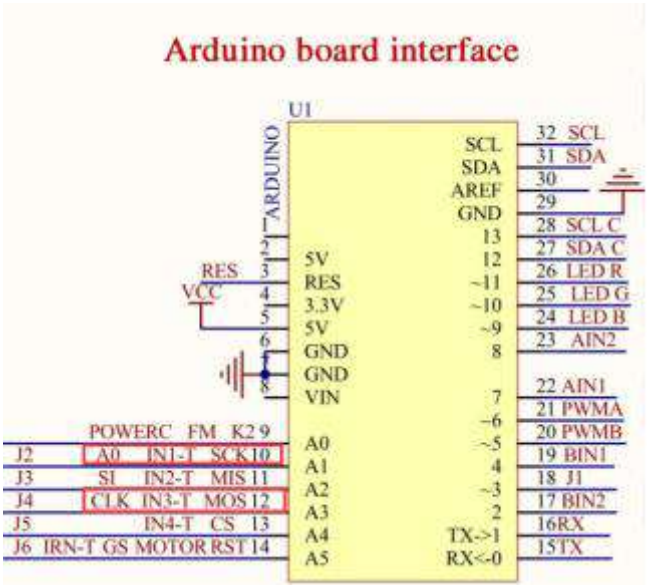
3)Principle of experimental

The basic principle of the infrared sensor to avoid obstacles is to use the reflective nature of the object.

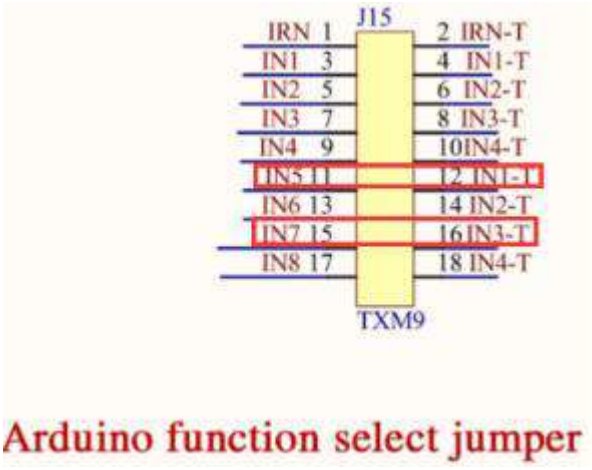
Within a certain range, if there is an obstacle, the infrared rays will encounter obstacle and will be reflected to reach the sensor receiving pin. In this experiment, we used 2 infrared sensors connected to the Raspberry Pi board to detect the obstacles by detecting the electrical level of the two ports, and robot car will make corresponding obstacle avoidance actions.

4)Experimental Steps

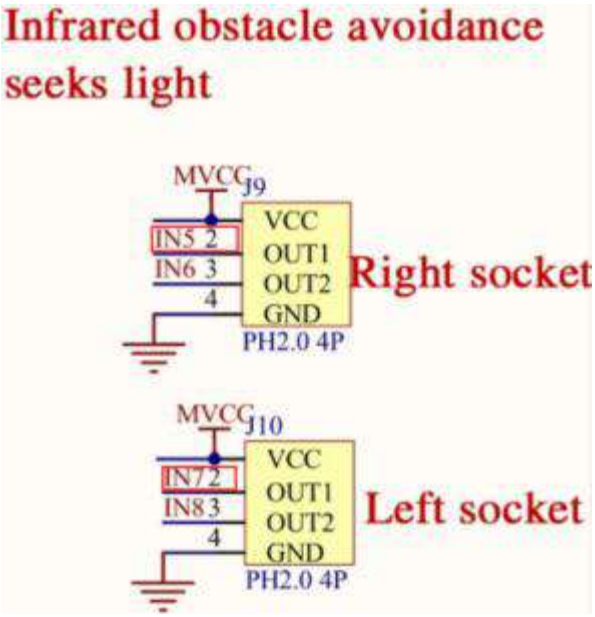
4-1 About the schematic



4-1 Raspberry Pi interface circuit diagram



4-2 Arduino funtion select jumper



4-2 Left and right infrared sensor interface

4-2 According to the circuit schematic:

Left infrared sensor----- A3

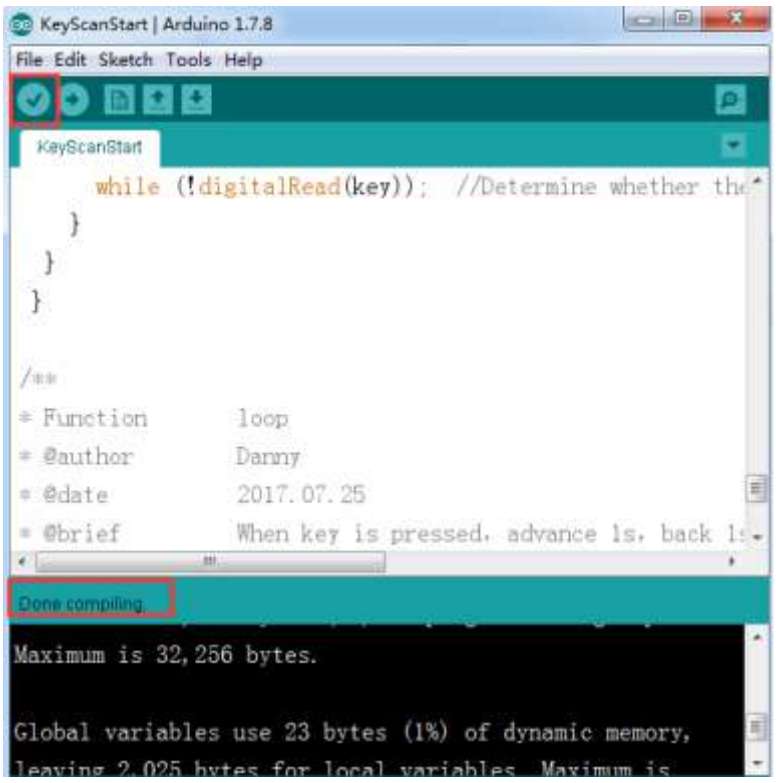
Right infrared sensor-----A1

(Note: We use the wiringPi library to write code.)

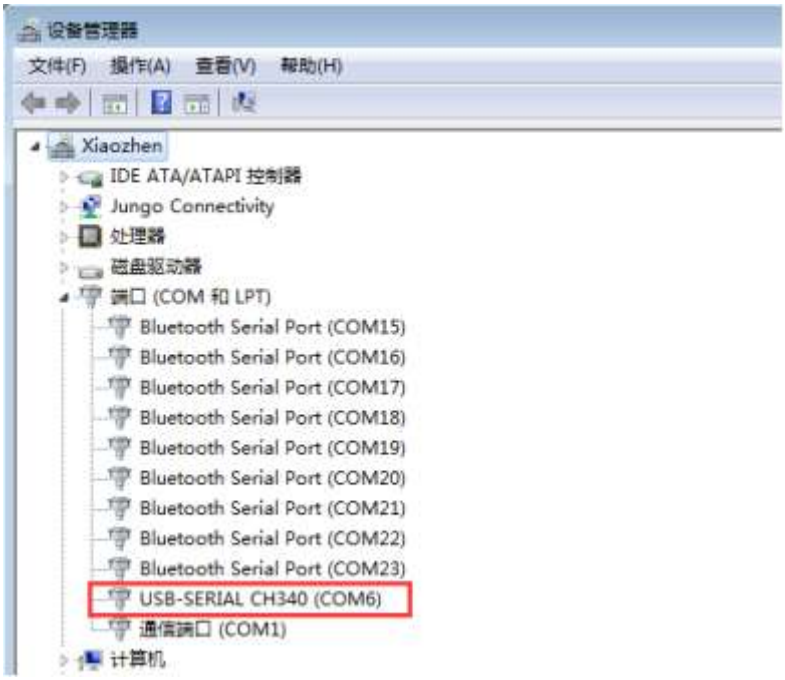
（Note: In this experiment, we can adjust the sensitivity of the infrared obstacle avoidance module by rotating the potentiometer on the infrared module to achieve better experimental results.）

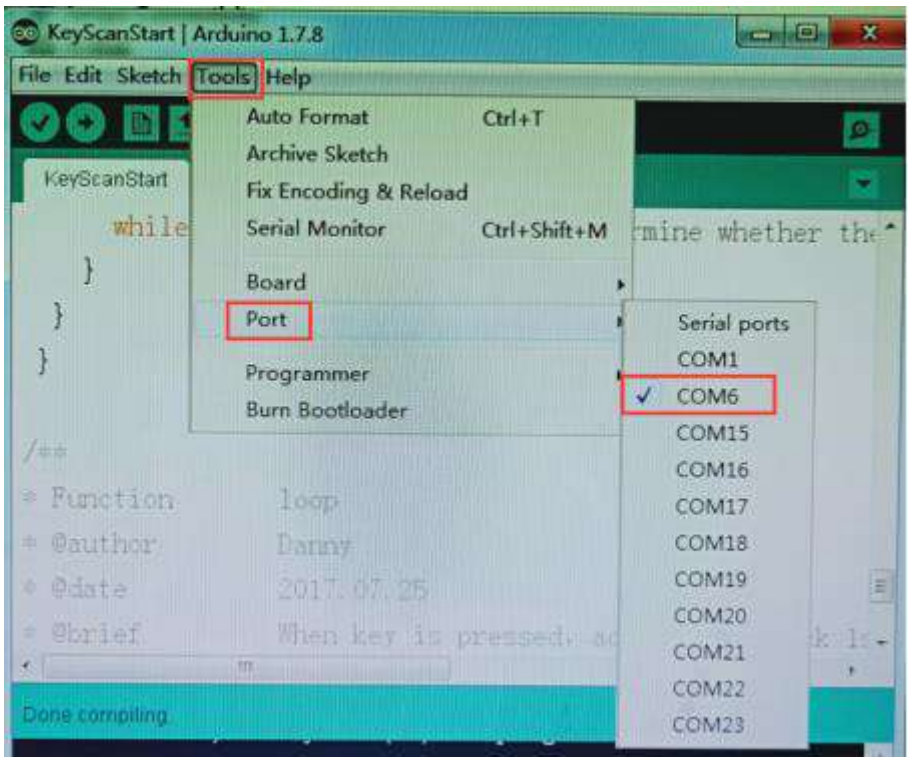
4-3 About the code

1. We need to open the code of this experiment:**infrared_avoid.ino**, 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.

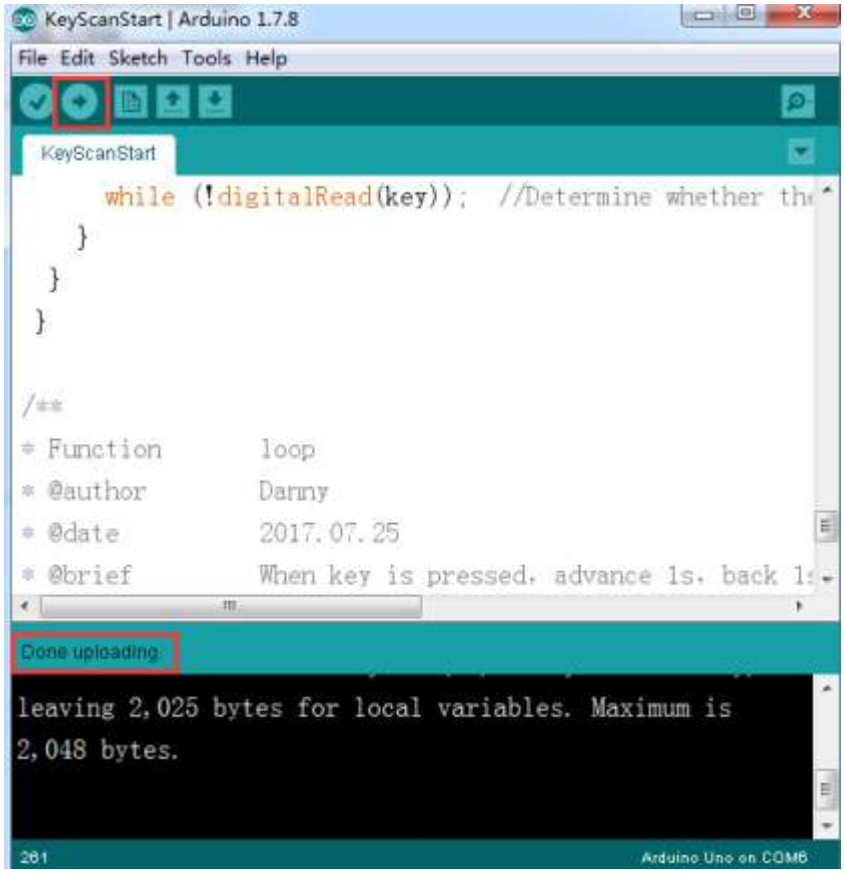


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



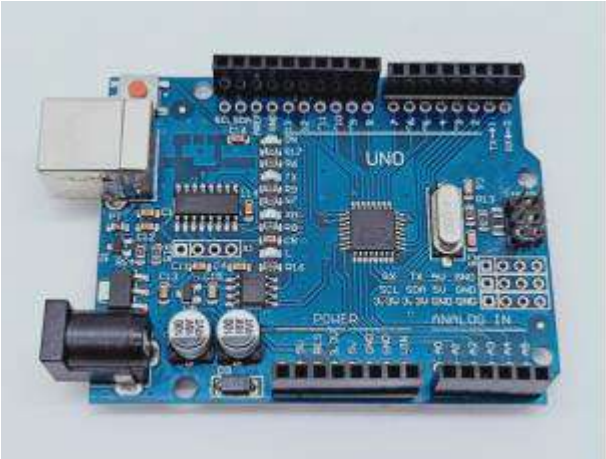


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



7- Infrared_follow

1)Preparation



1-1 Arduino UNO board

1-2 Infrared obstacle avoidance module

2)Purpose of Experimental

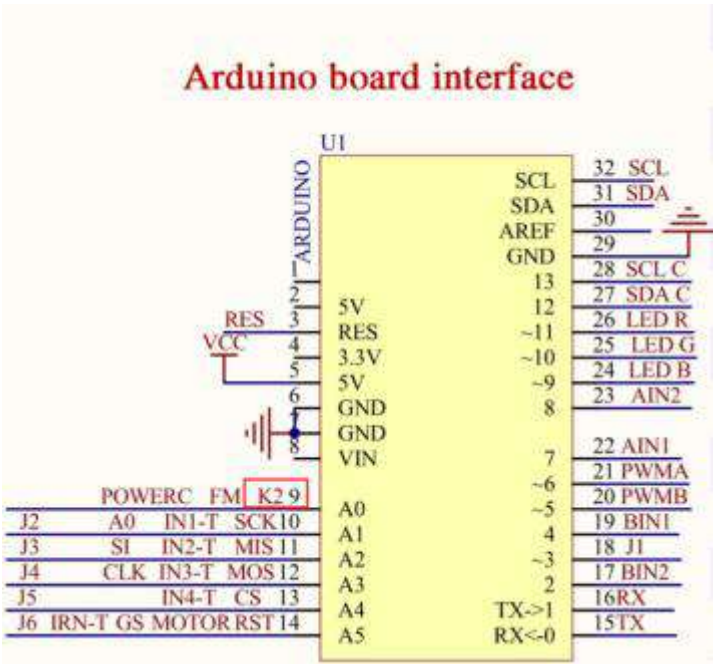
After the code upload is completed, You need to press the K2 to start the car, and the infrared follow function is started. When there is an obstacle in front, the car can follow the obstacle automatically.

3)Principle of experimental

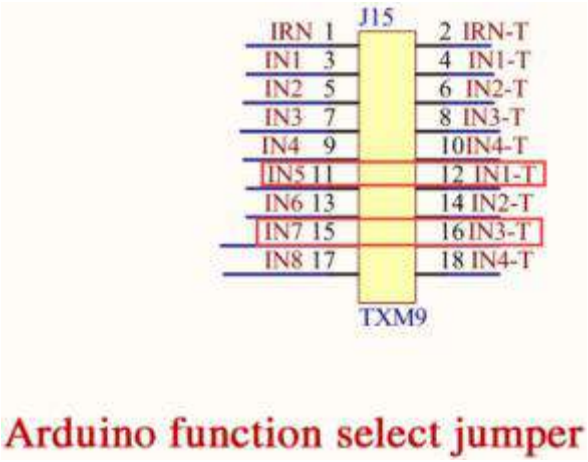
The basic principle of the infrared sensor is to use the reflective nature of the object. Within a certain range, if there is an obstacle, the infrared rays will encounter obstacle and will be reflected to reach the sensor receiving pin. In this experiment, we used 2 infrared sensors connected to the Raspberry Pi board to detect the obstacles by detecting the electrical level of the two ports, and robot car will make corresponding follow actions.

4)Experimental Steps

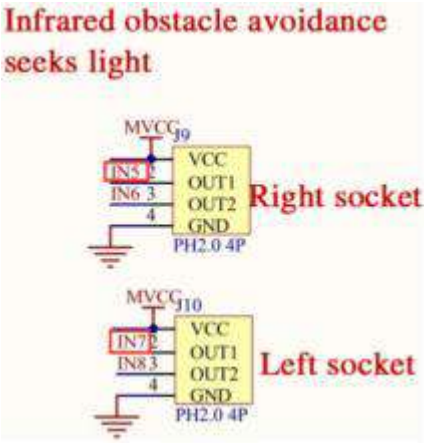
4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 Arduino function select jumper



4-3 Left and right infrared sensor interface

4-2 According to the circuit schematic:

Left infrared sensor-----A3

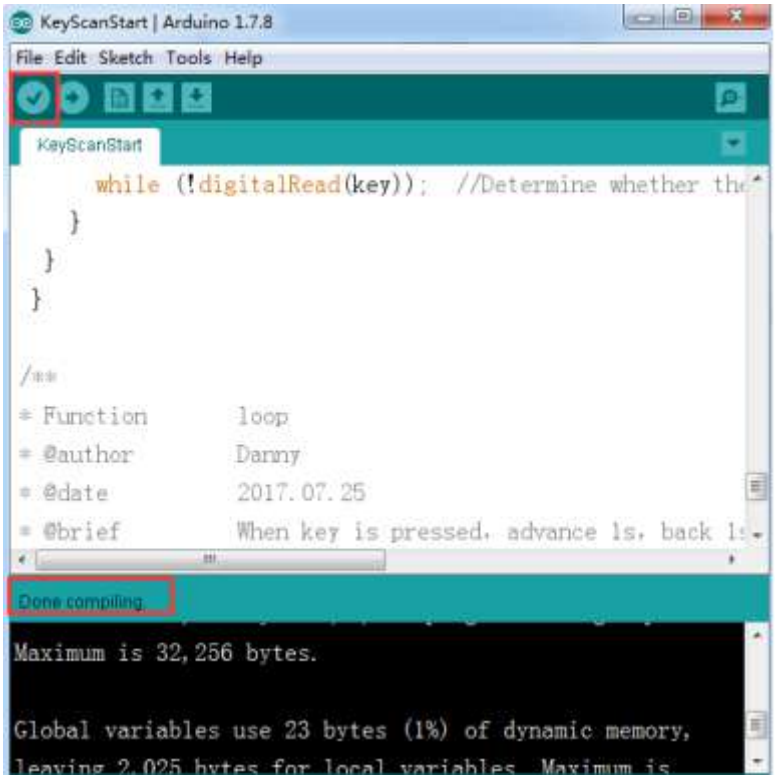
Right infrared sensor-----A1

(Note: We use the wiringPi library to write code.)

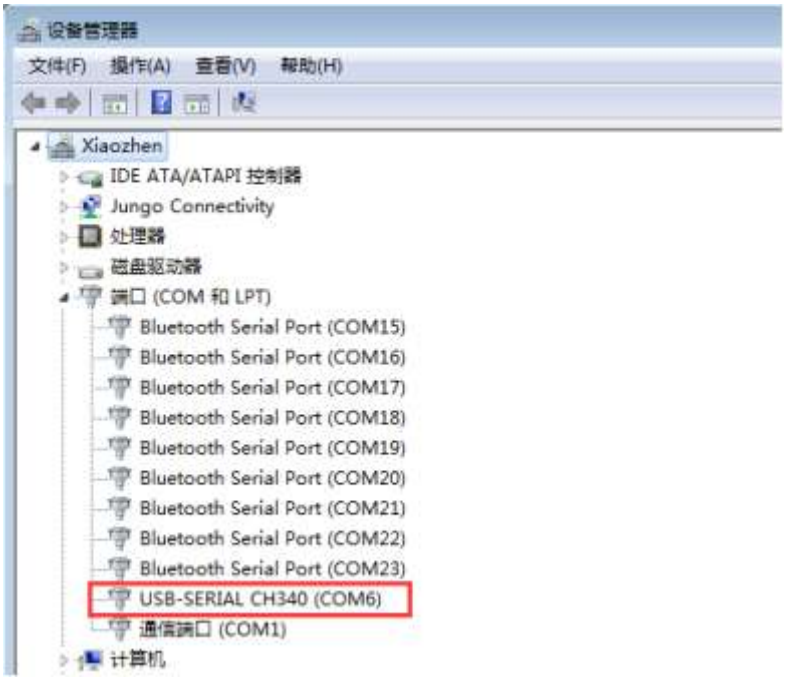
（Note: In this experiment, we can adjust the sensitivity of the infrared follow module by rotating the potentiometer on the infrared module to achieve better experimental results.）

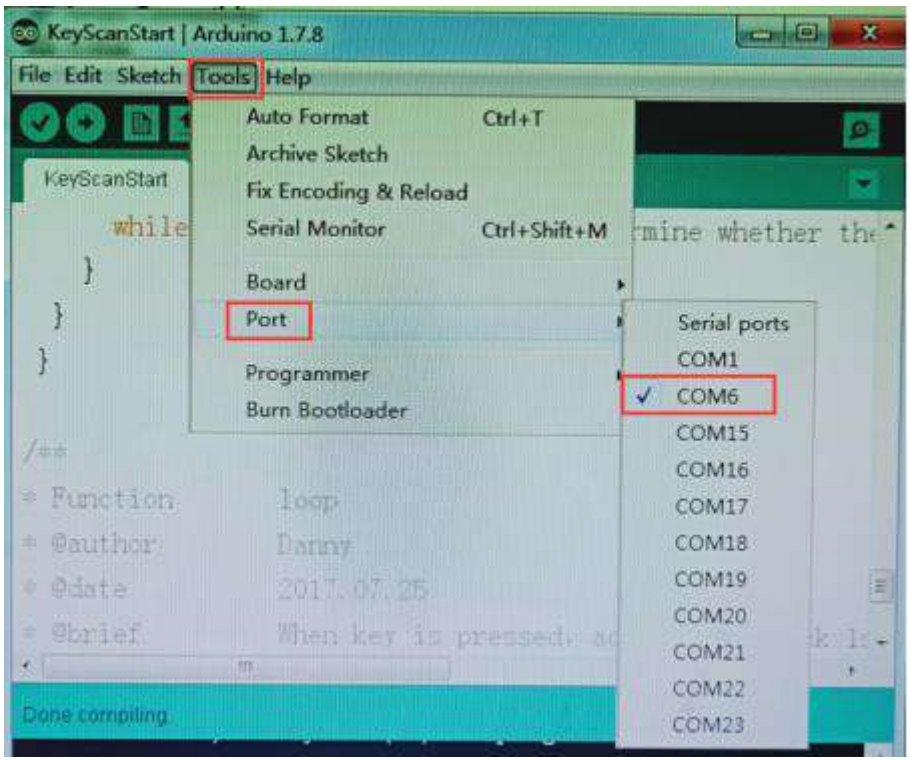
4-3 About the code

1. We need to open the code of this experiment:**infrared_follow.ino**, 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.

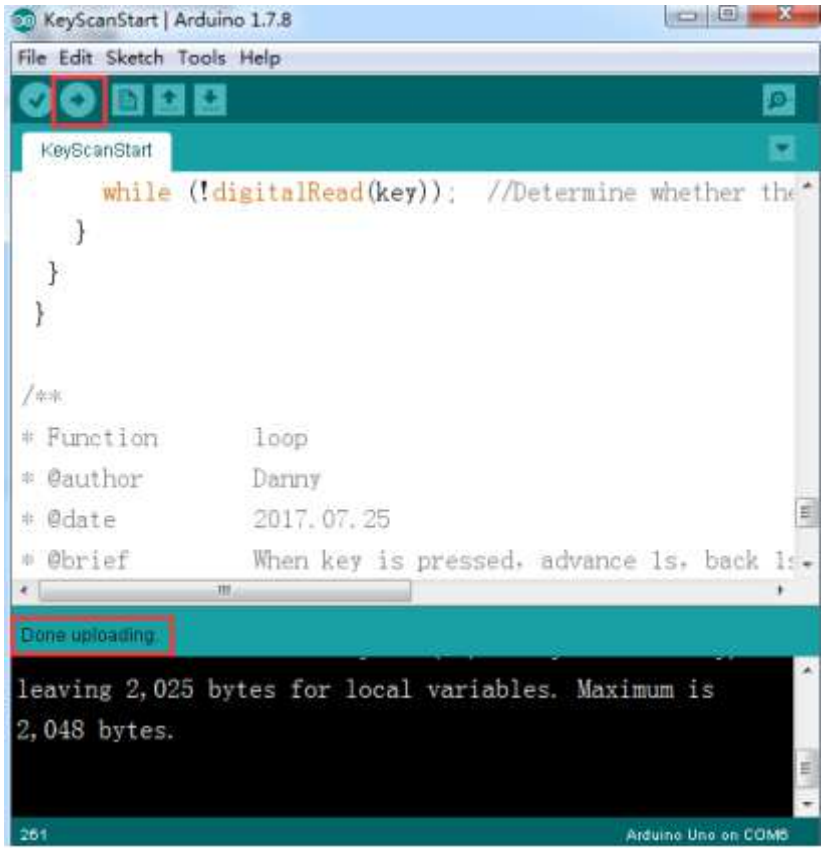


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



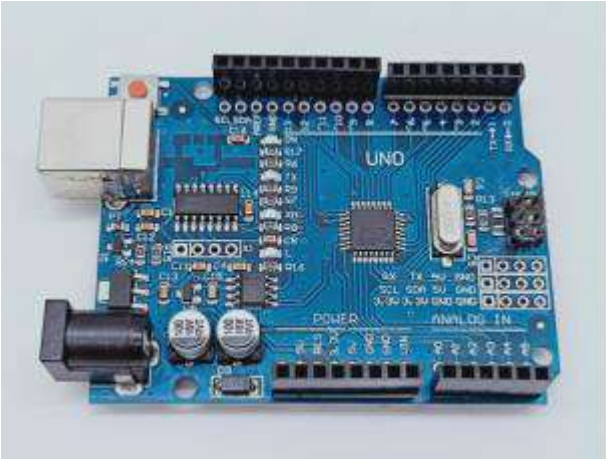


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



8- Light_follow

1) Preparation



1-1 Arduino UNO board

1-2 Infrared obstacle avoidance module

2)Purpose of Experimental

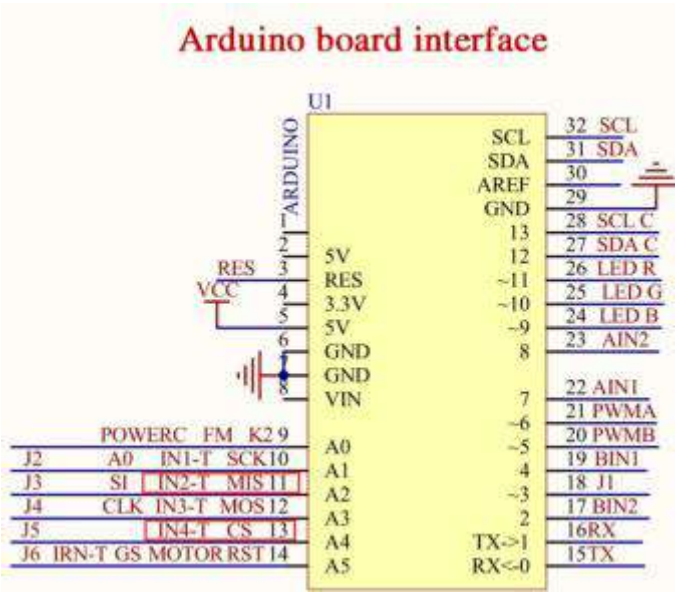
After the code upload is completed, you need to press the K2 to start the car, and the light follow function is started. When both light-sensitive resistors detect light, the car advance; when there is light detected on the left side, the car turn left; when light is detected on the right side, the car turn right; when no light is detected on the left and right sides, the car stopped.

3)Principle of experimental

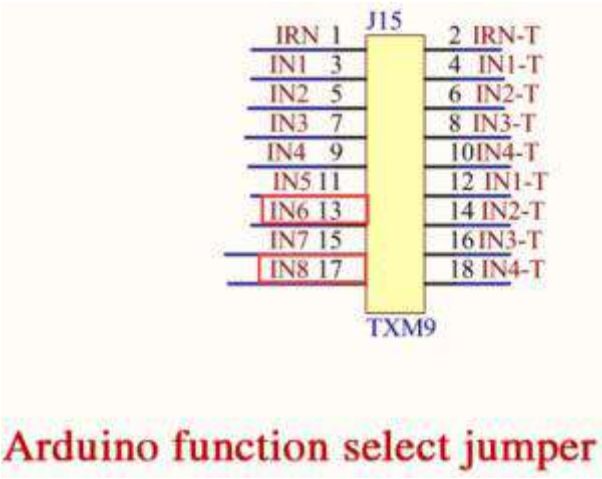
The photoresistor is a resistor made by utilizing the photoconductivity of the semiconductor to change the resistance value according to the intensity of the incident light. The incident light intensity, the resistance is reduced, the incident light is weak, and the resistance is increased. If there is light, the level of the pin connected to the photoresistor becomes high level.

4)Experimental Steps

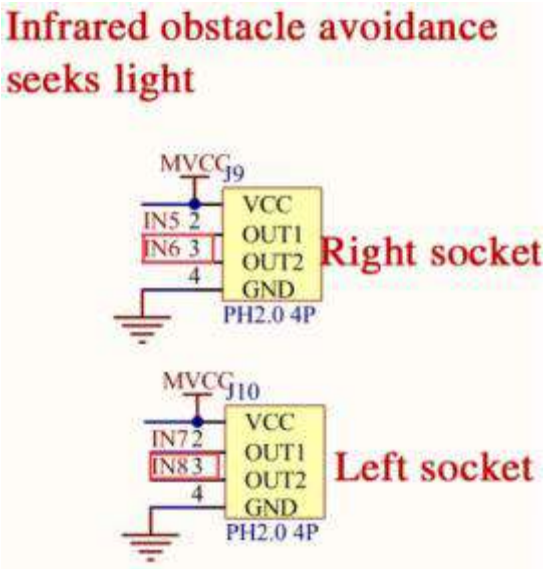
4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 Arduino function select jumper



4-3 Left and right infrared sensor interface

4-2 According to the circuit schematic:

Left infrared sensor----- A4

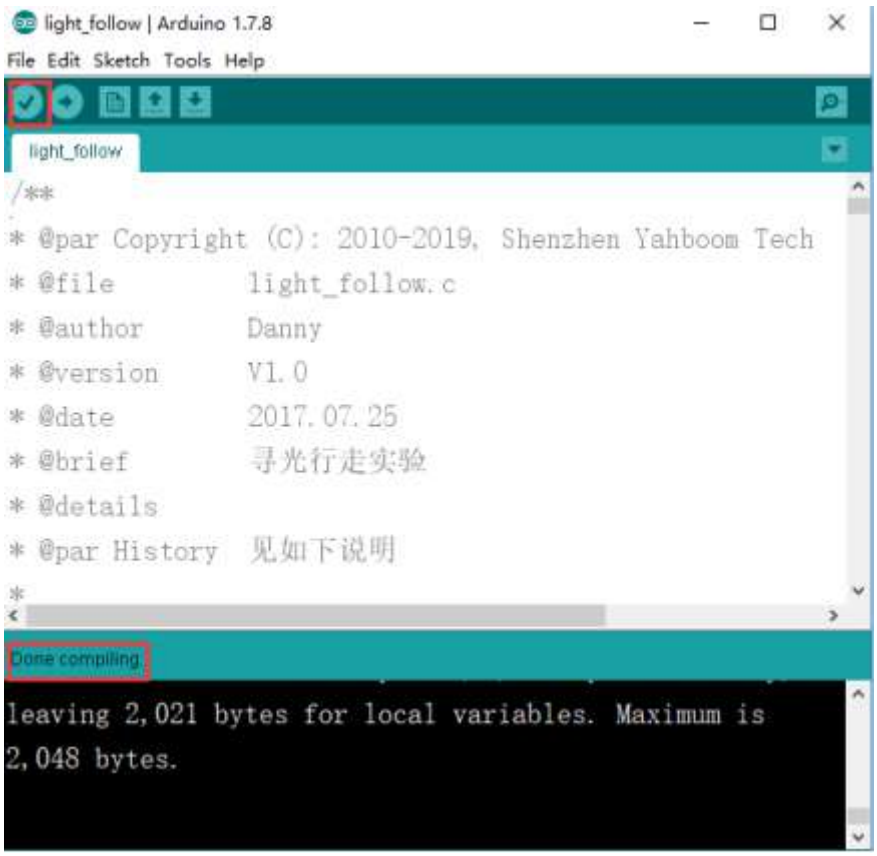
Right infrared sensor-----A2

(Note: We use the wiringPi library to write code.)

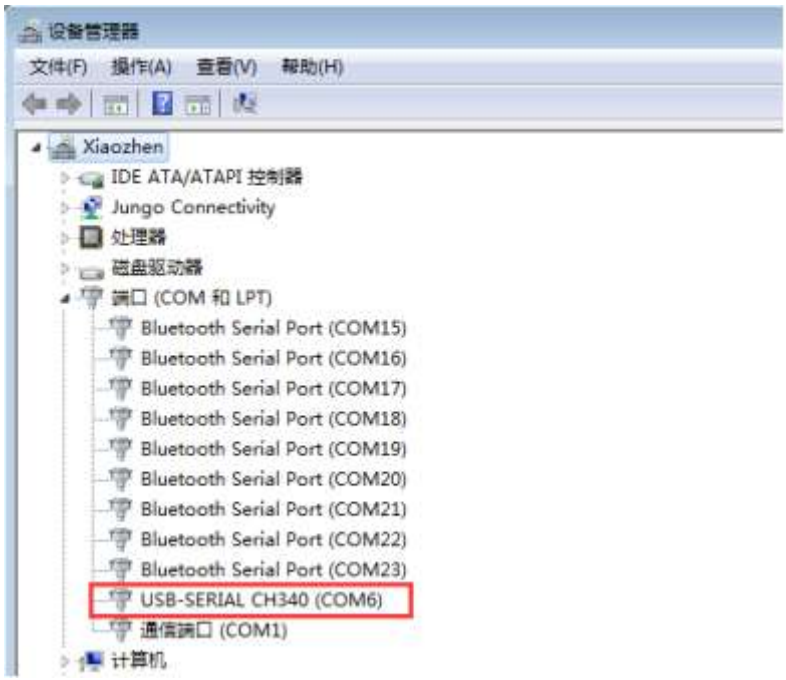
（Note: In this experiment, we can adjust the sensitivity of the light follow module by rotating the potentiometer on the infrared module to achieve better experimental results.）

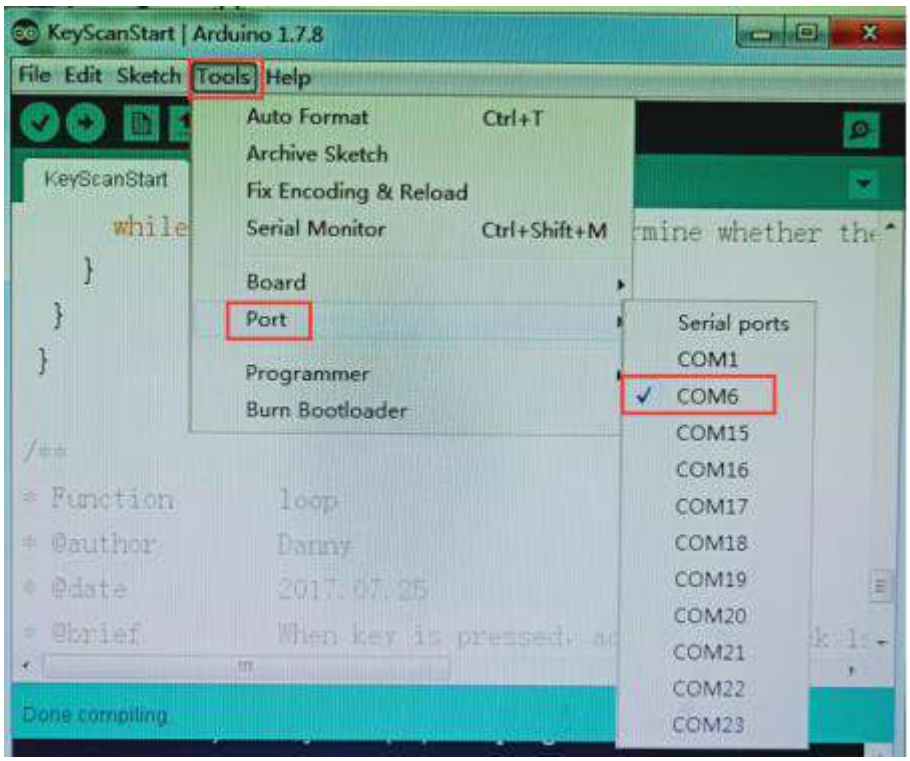
4-3 About the code

1. We need to open the code of this experiment:**light_follow.ino**, 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.

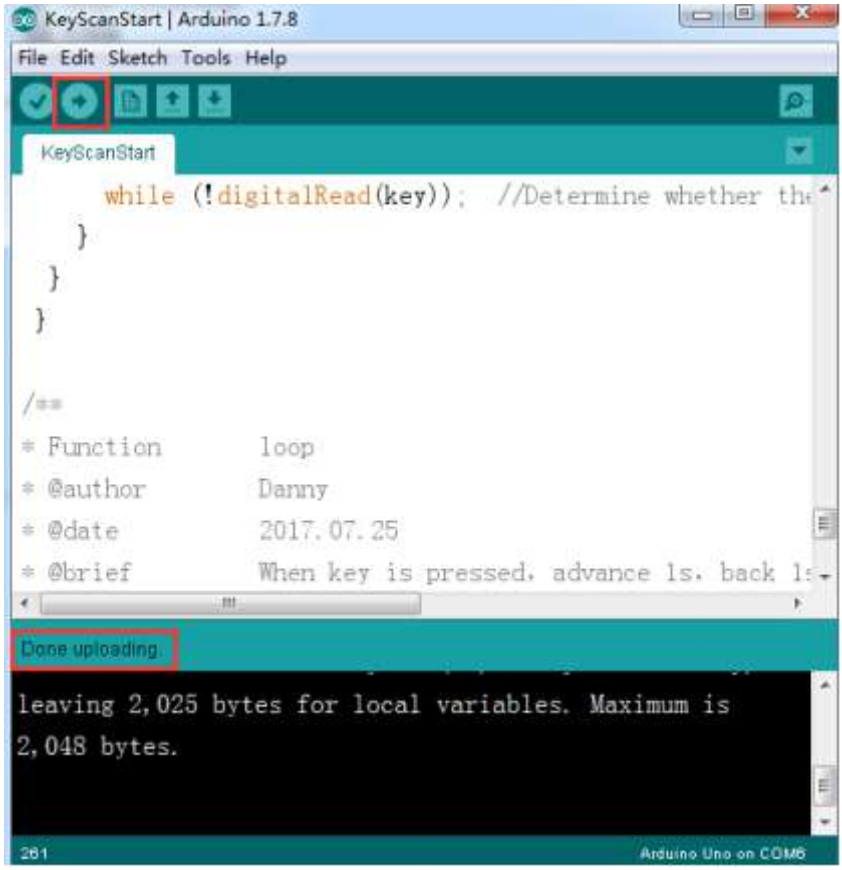


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



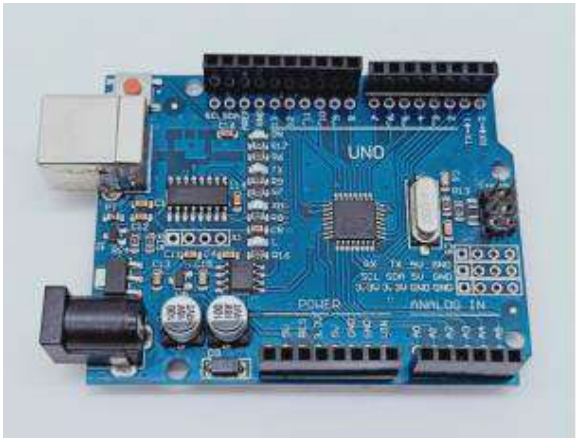


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



9- Avoid_ultrasonic

1) Preparation



1-1 Arduino UNO board

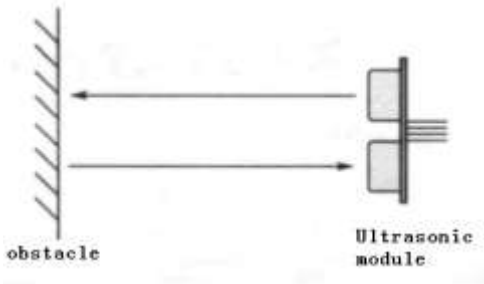
1-2 Ultrasonic module

2)Purpose of Experimental

After the code upload is completed, you need to press the K2 to start the car, and the ultrasonic obstacle avoidance function is started. When there is an obstacle in front, the car can avoid the obstacle automatically.

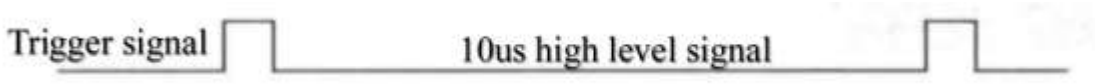
3)Principle of experimental

The ultrasonic module is a sensor that uses ultrasonic characteristics to detect the distance. It has two ultrasonic probes for transmitting and receiving ultrasonic waves. The range of measurement is 3-450 cm.



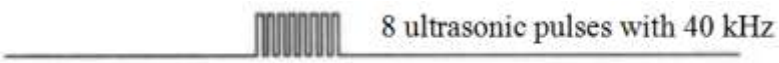
3-1 Ultrasonic emission and reception schematic

(1) You need to input a high level signal of at least 10us to the Trig pin to trigger the ranging function of the ultrasonic module.



3-2 Ultrasonic module sends trigger signal

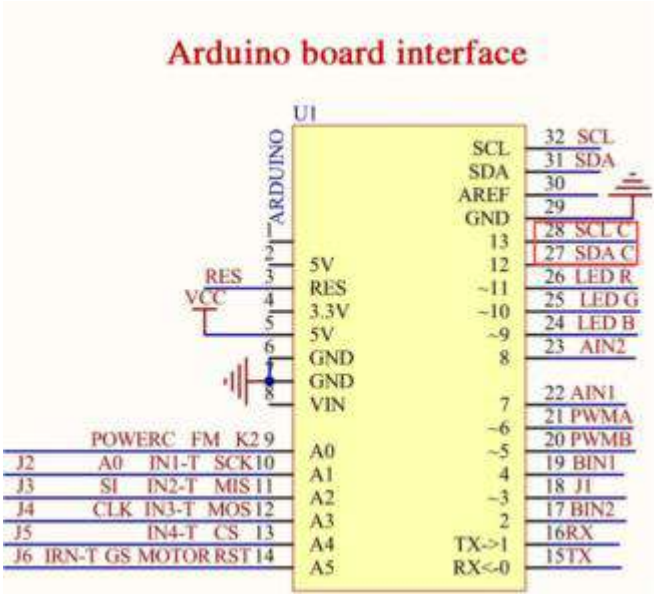
(2) After the ranging function is triggered, the module will automatically send out 8 ultrasonic pulses with 40 kHz and automatically detect whether there is a signal return. This step is done internally by the module.



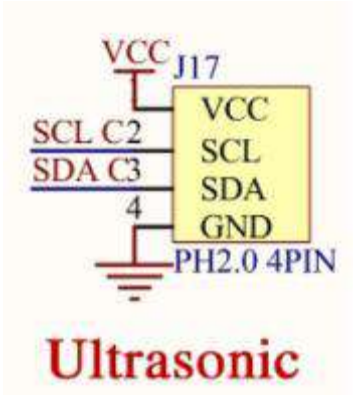
(3) When the module detects an echo signal, the ECHO pin will output a high level. The high level duration is the time from when the ultrasonic wave is sent to when it returns. You can calculate the distance by using the time function to calculate the high level duration. Formula: Distance = High level duration * Speed of sound(340M/S)/2.

4)Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



4-2 ultrasonic interface

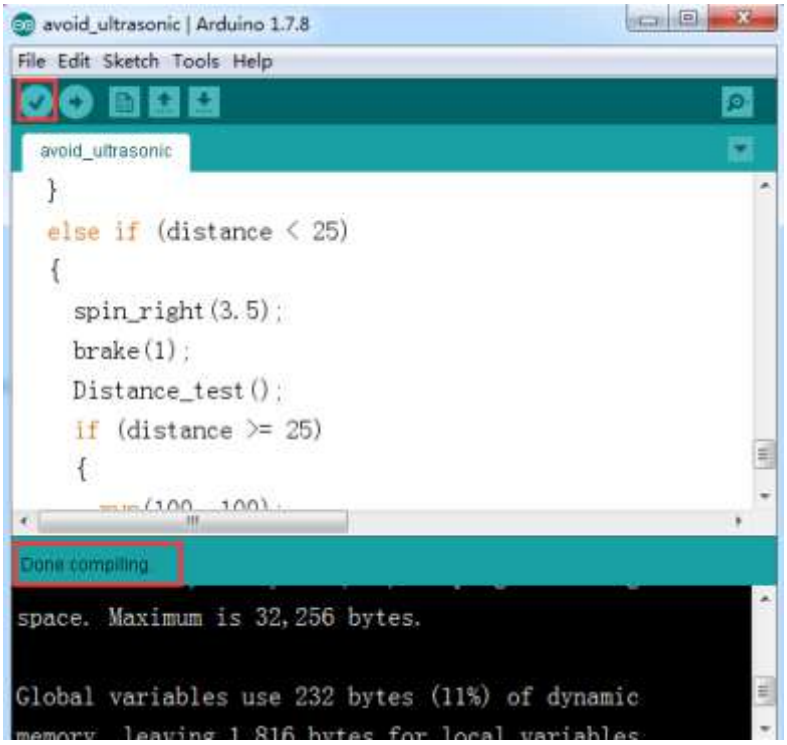
4-2 According to the circuit schematic:

Trig-----SCL----- 13(Arduino UNO)

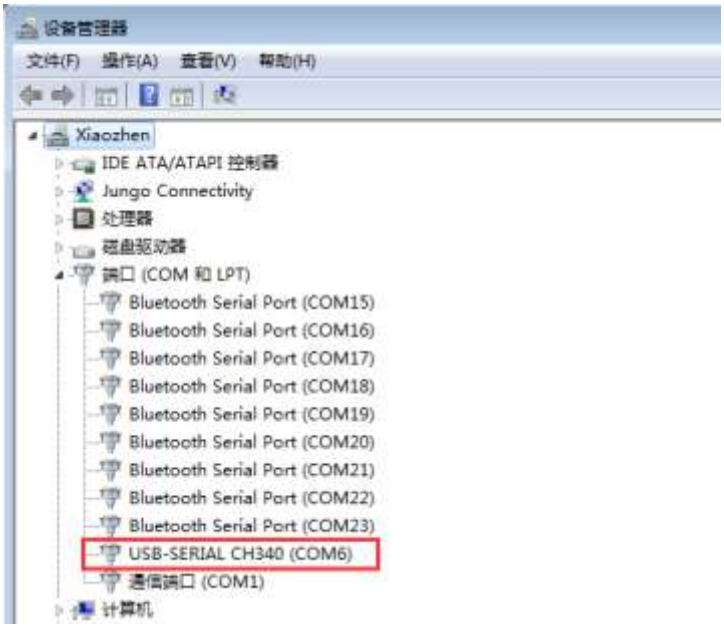
Echo-----SDA-----12(Arduino UNO)

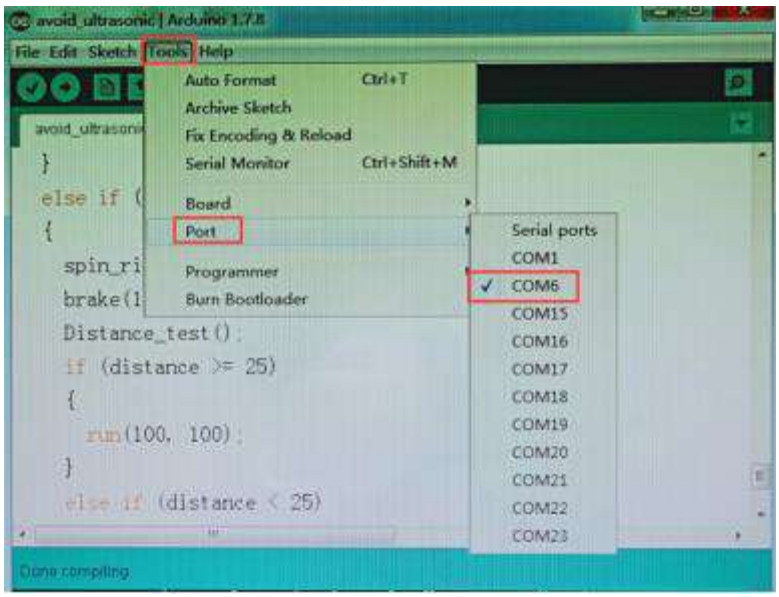
4-3 About the code

1. We need to open the code of this experiment:**avoid_ultrasonic.ino**, 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.

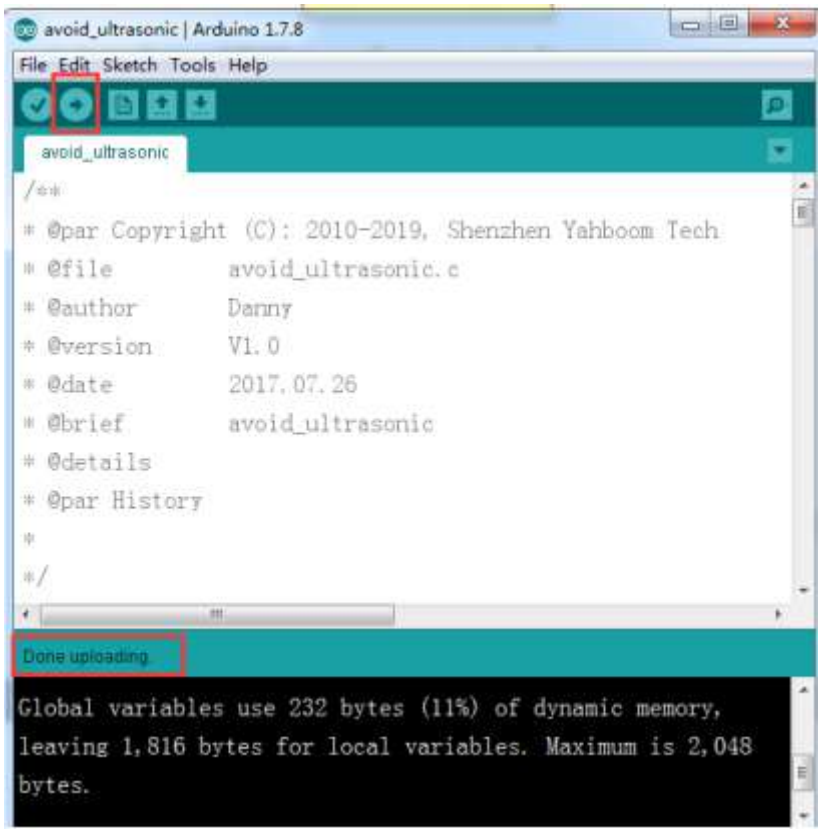


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



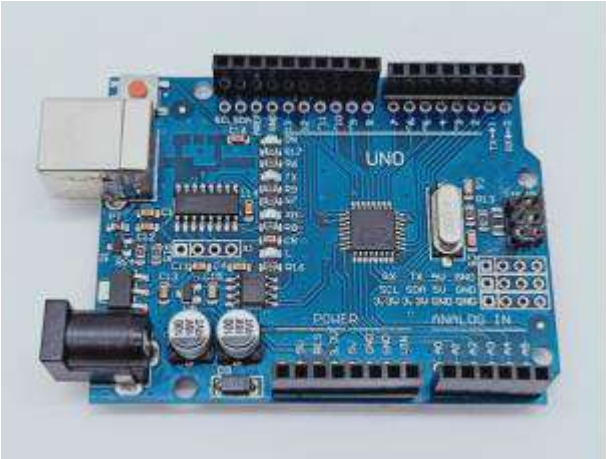


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



10- Color_recognition

1) Preparation



1-1 Arduino UNO board

1-2 color_recognition module

2)Purpose of Experimental

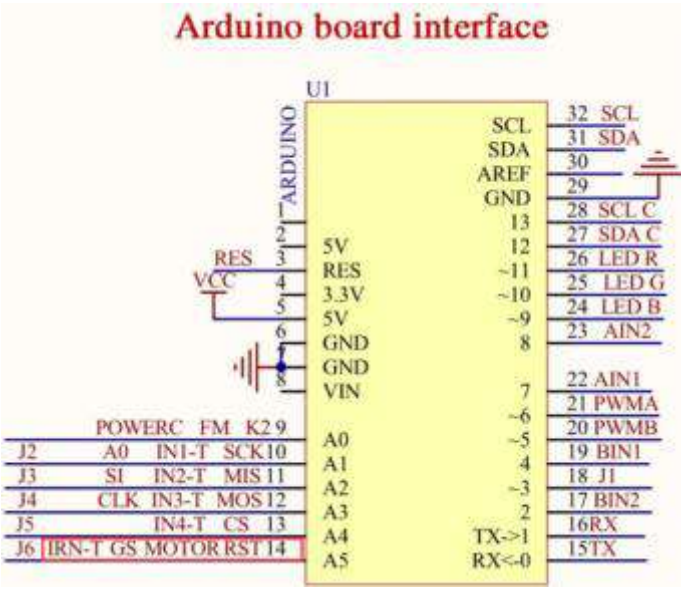
After the code upload is completed. The color recognition module under the car is turned on automatically, and then the serial monitor of the Arduino IDE is opened to see that the photoresistor reads the illuminance value corresponding to the different colors.

3)Principle of experimental

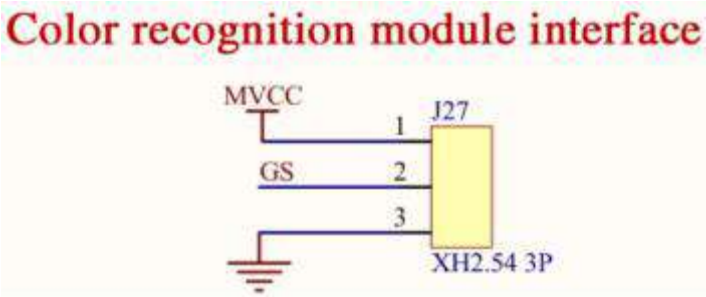
The sensor emits white visible light. Reflected by the reflective surface, the photoresistor can sense the reflected intensity to output different voltages. Through the processing of the voltage signal, the gradation change of the reflecting surface can be analyzed.

4)Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



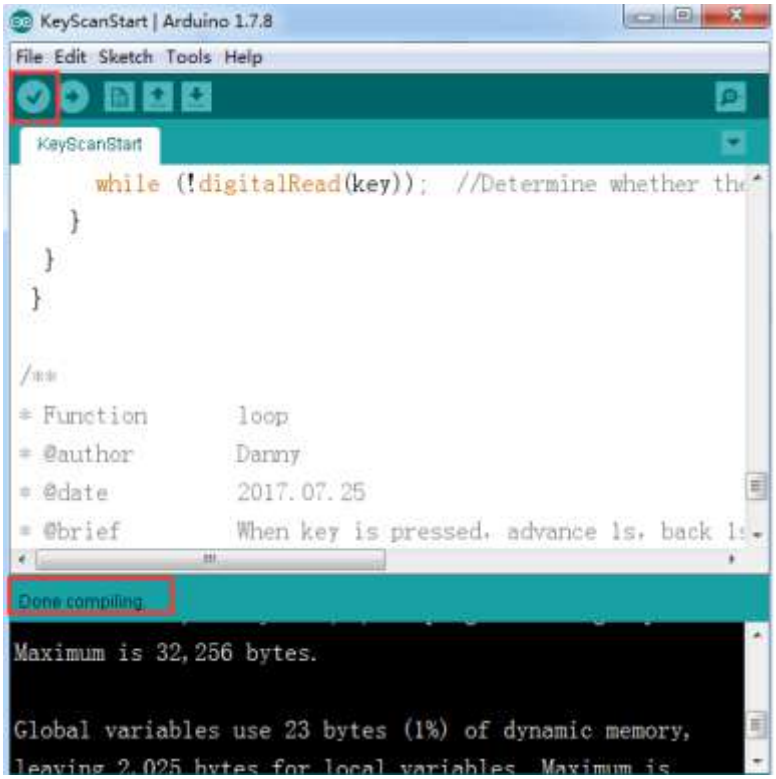
4-2 Color recognition module interface

4-2 According to the circuit schematic:

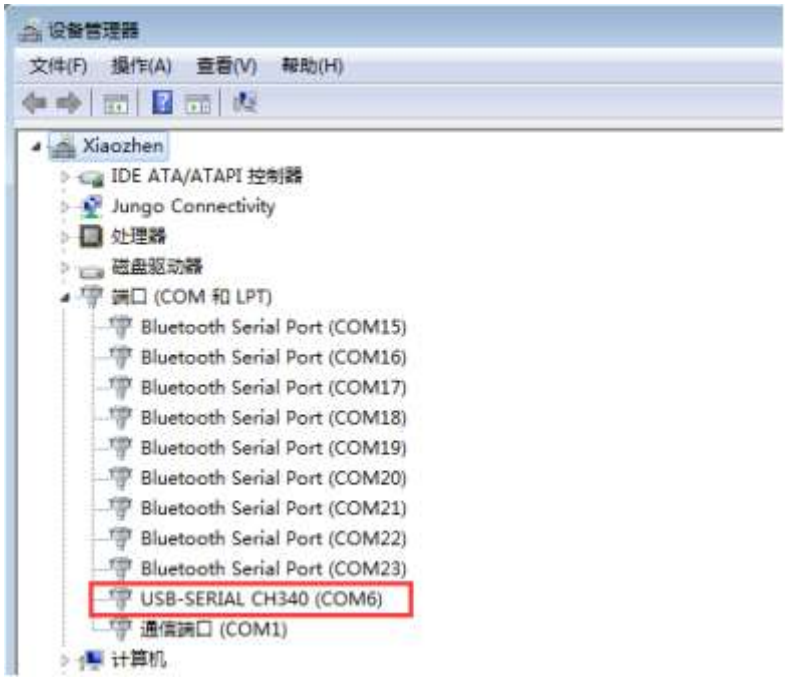
GS-----A5(Arduino UNO)

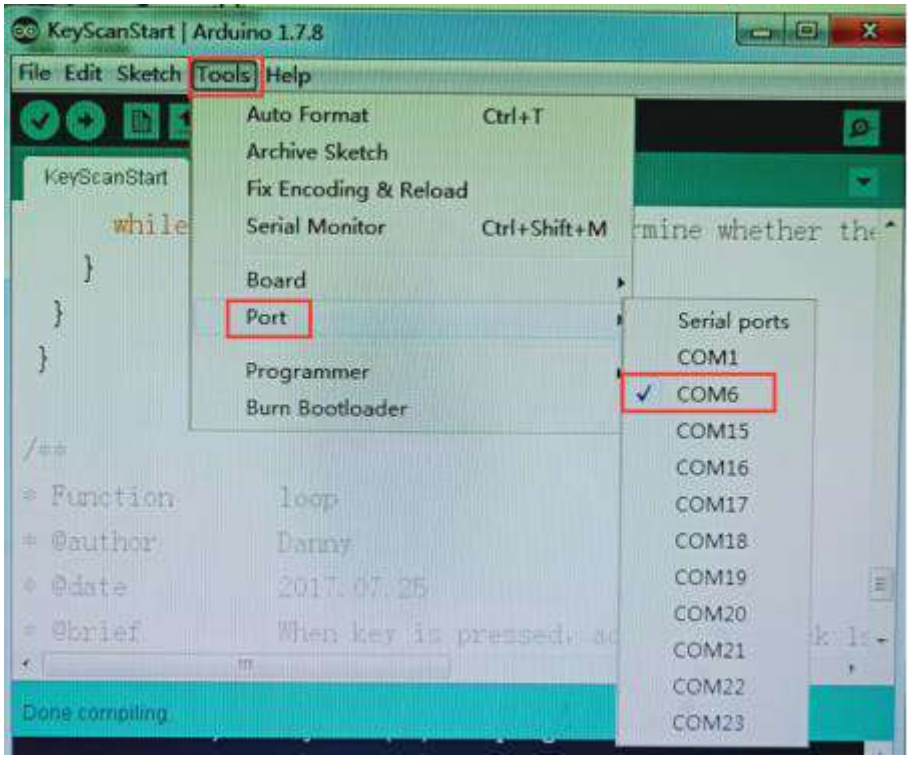
4-3 About the code

1. We need to open the code of this experiment:**color_recognition.ino**, 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.

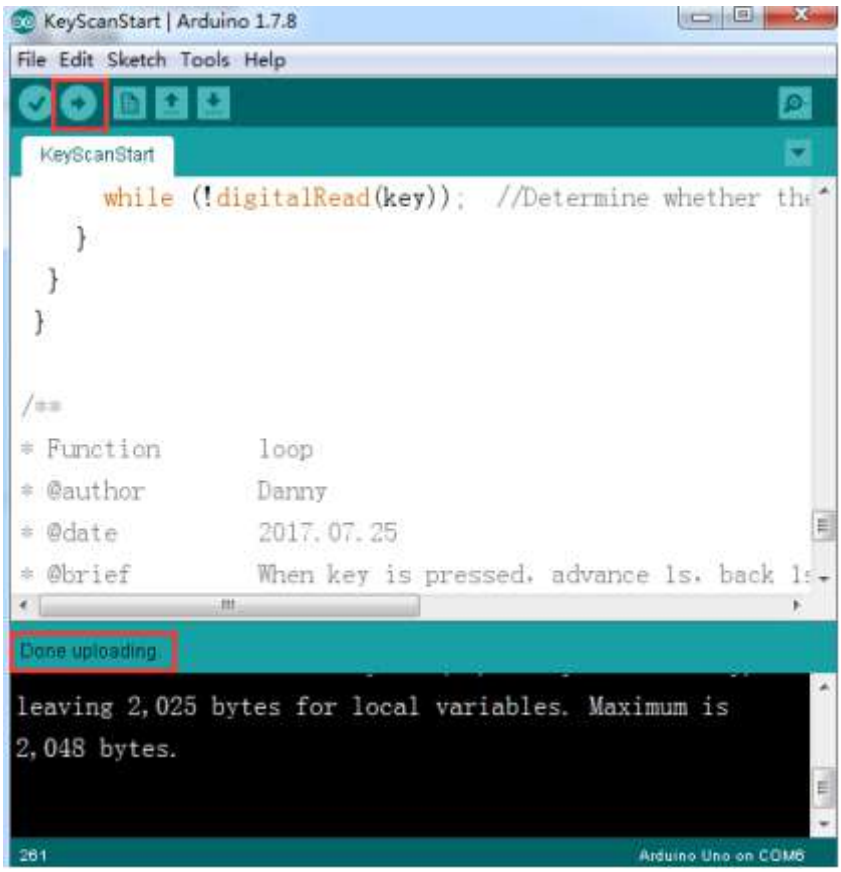


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.

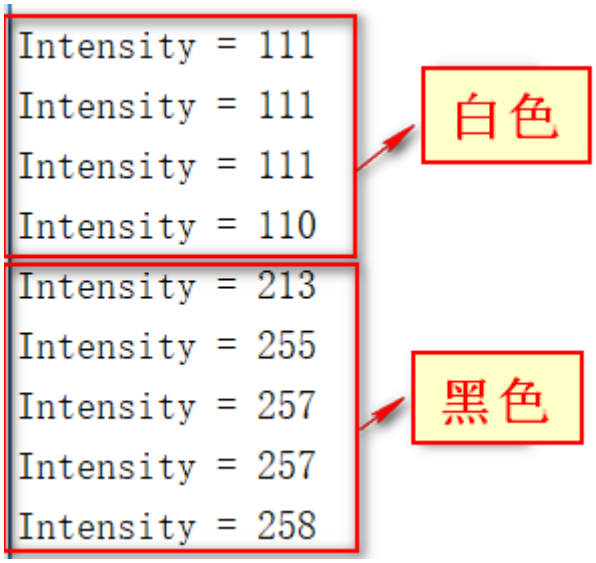


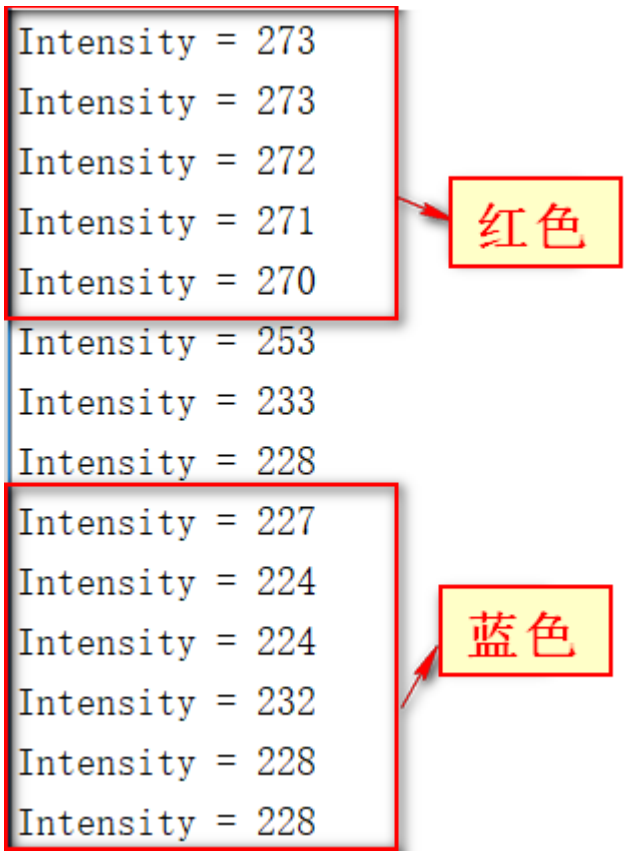


3.After the selection is completed, you need to click "→"under the menu bar to upload the code to the Arduino UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



The following is the different data read by the serial port by testing 4 different colors. (The data measured this time is for reference only, the actual test environment shall prevail)




Intensity = 273
Intensity = 273
Intensity = 272
Intensity = 271
Intensity = 270
Intensity = 253
Intensity = 233
Intensity = 228
Intensity = 227
Intensity = 224
Intensity = 224
Intensity = 232
Intensity = 228
Intensity = 228
红色
蓝色

11- Voltage_detection

1) Preparation

1-1 Arduino UNO board

2)Purpose of Experimental

After the code upload is completed. You can see the battery voltage of the car by the serial monitor of the Arduino IDE.

3)Principle of experimental

The analog port of Arduino can detect 0-5V voltage, corresponding to the value of 0-1023.

According to principle of equal current, we can obtain: Vm/（R14+R15）= Vad/R15. According to detecting by Arduino UNO,we can obtain:Vad= (AD/1023)*5.

Vm/（R14+R15）= Vad/R15 ----------①

Vad= (AD/1023)*5 ----------②

From ①、②, we can obtain this formula:

Voltage = (Voltage / 1023) * 5 *（R14 + R15）/ R15-0.35

Note: 5----Standard voltage of the Arduino UNO

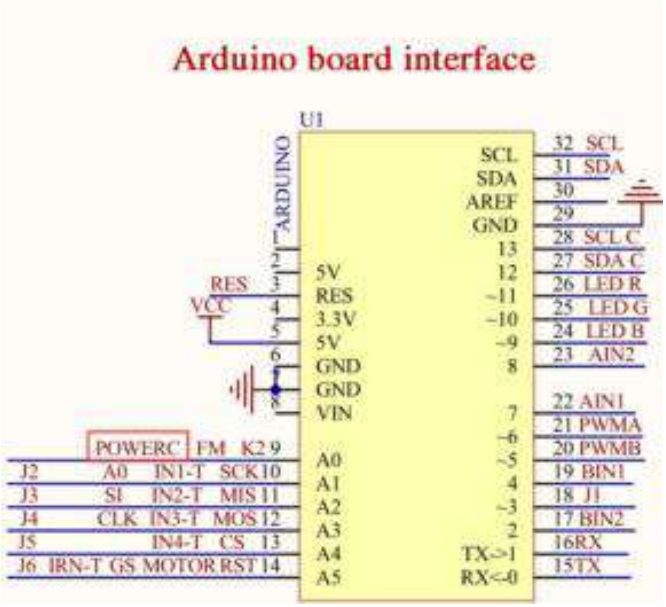
Voltage----The AD value collected by port A0(0-1023)

3----（R14+R15）/R15 (R14=20K,R15=10K)

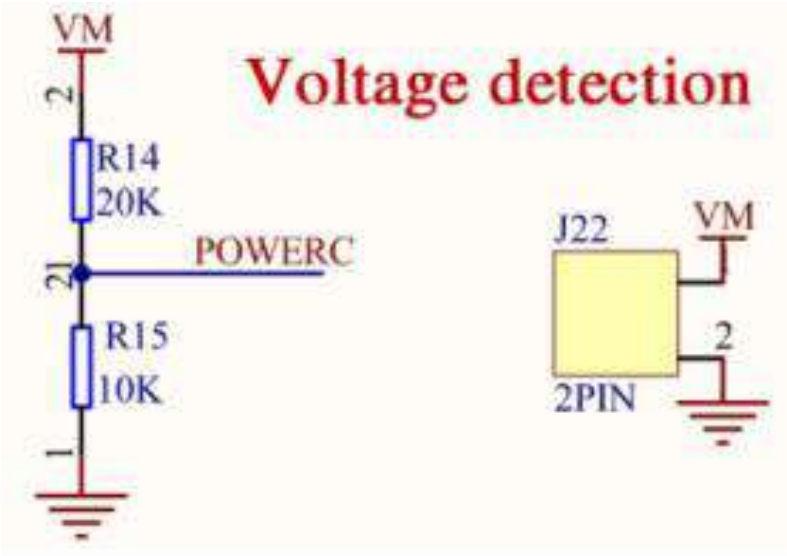
0.35----An adjustment value due to the accuracy of the resistor.

4)Experimental Steps

4-1 About the schematic



4-1 Arduino UNO interface circuit diagram



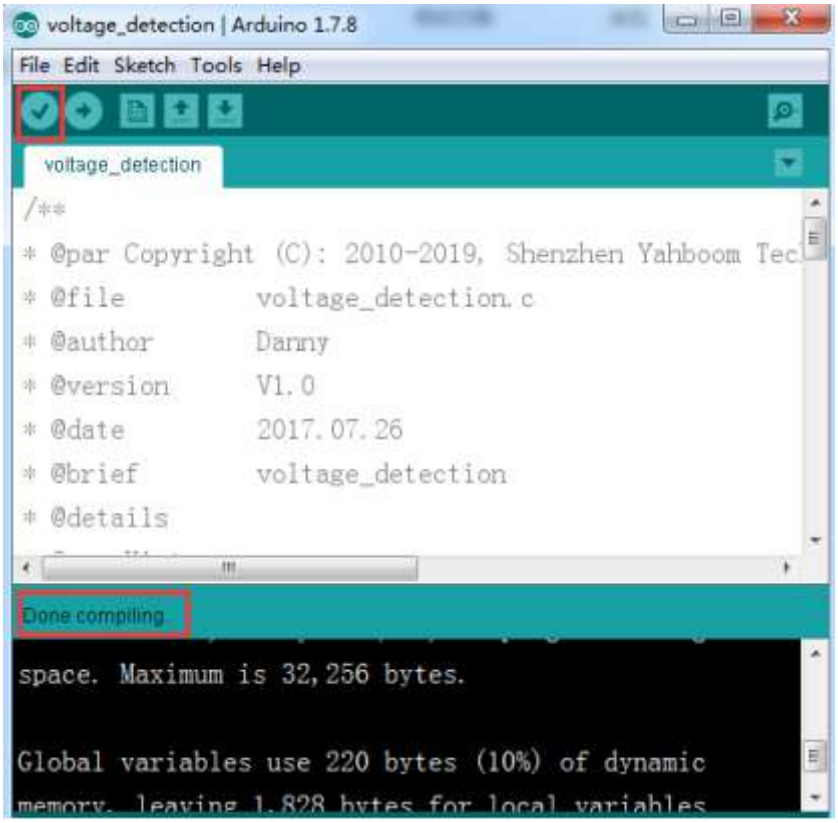
4-2 Battery voltage detection circuit

4-2 According to the circuit schematic:

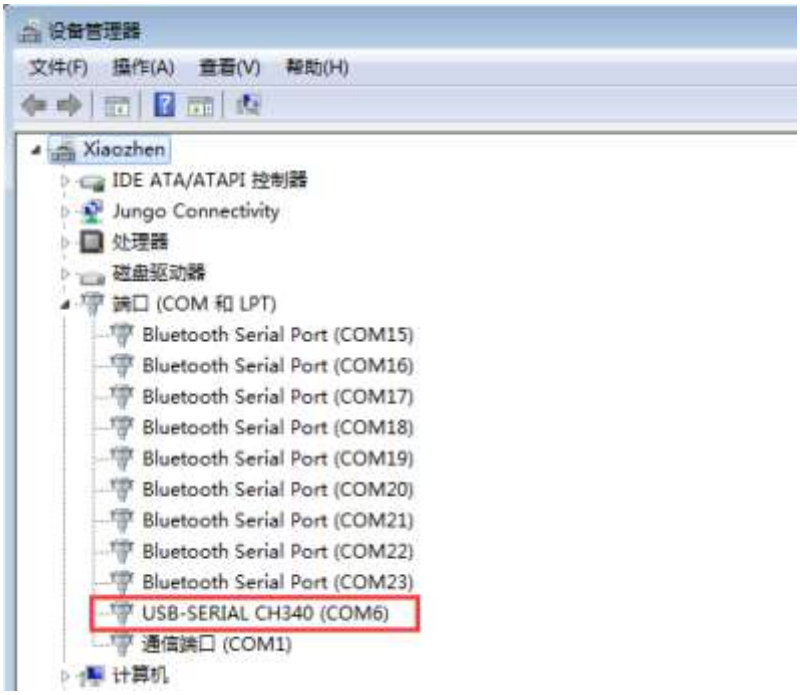
POWERC-----A0(Arduino UNO)

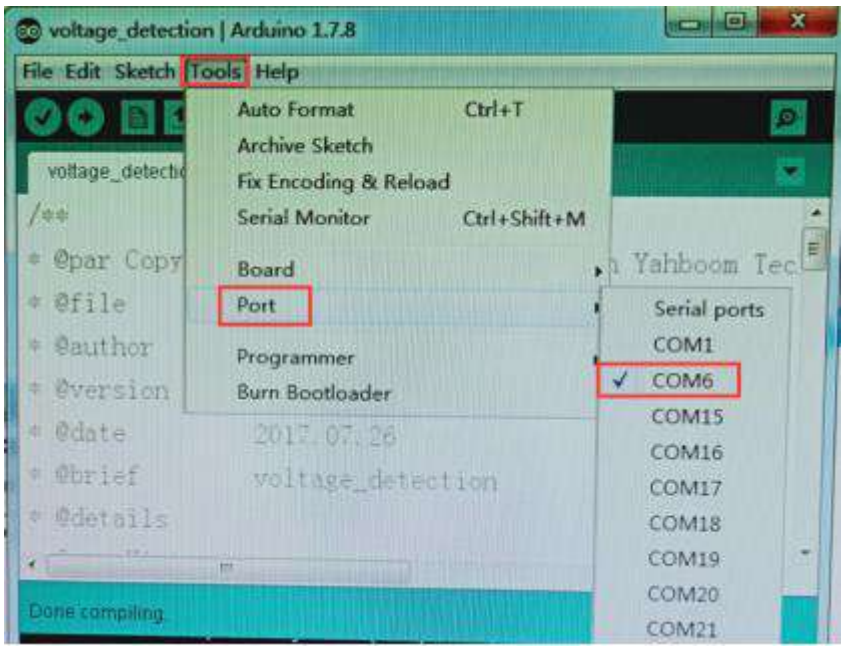
4-3 About the code

1. We need to open the code of this experiment:**voltage_detection.ino**, 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.

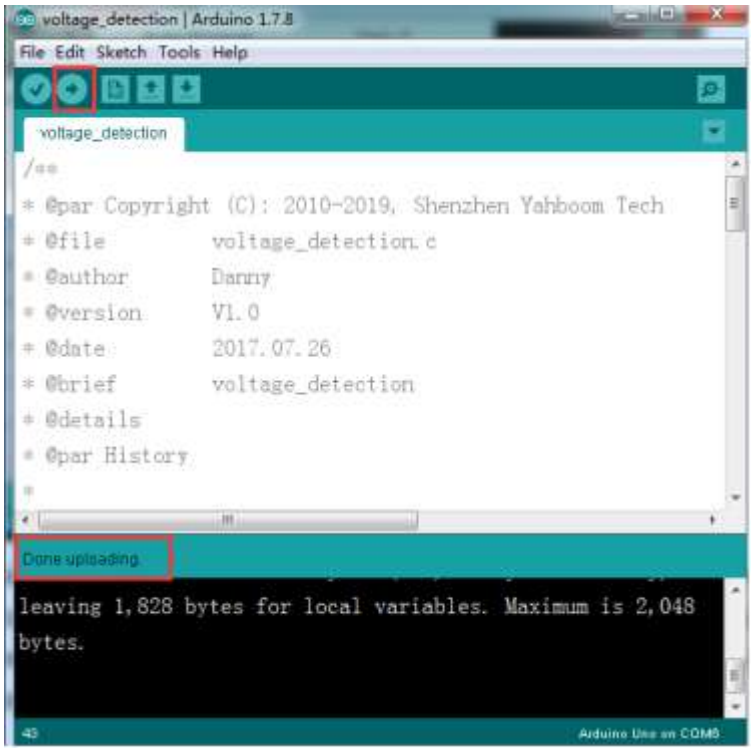


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





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



4.After the code is uploaded. You need to open the serial port monitor on the top right corner of Arduino IDE, and Set the serial port baud rate to 9600. You will see that the current voltage value of the car is printed.

12- Servo_ultrasonic_avoid

1) Preparation

1-1 Arduino UNO board

1-2 Ultrasonic module

1-3 RGB module

1-4 SG90 servo

2)Purpose of Experimental

After the code upload is completed,you need to press the K2 to start the car, and the ultrasonic obstacle avoidance with servo function is started.

When the distance on the front is less than 35cm, the colorful light module is red, then servo turn to 0 degree, ranging and recording, servo turn to 180 degree, ranging and recording, servo turn to 90 degree, ranging and recording. And the robot car will compare the left and right distance to determine whether to avoid the obstacle to the left or right. When the distance in three directions is less than 25, the robot car will turn round.

3)Principle of experimental

The working principle of the servo: the control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor.

Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in

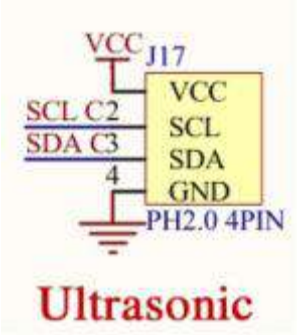
fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

0.5ms----------0°
1.0ms----------45°
1.5ms----------90°
2.0ms----------135°
2.5ms----------180°

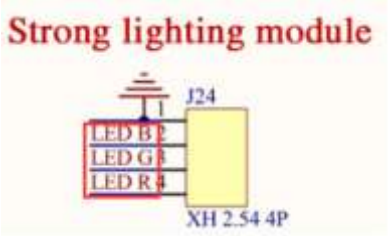
4)Experimental Steps

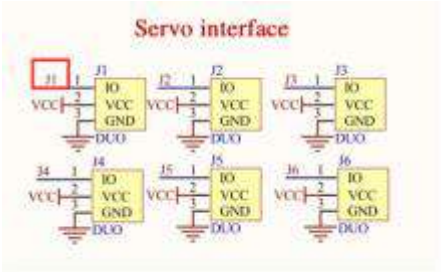
4-1 About the schematic

4-1 Arduino UNO interface circuit diagram



4-2 ultrasonic interface





4-3 RGB module interface 4-4 Servo interface

4-2 According to the circuit schematic:

LED_R-----11(Arduino UNO)

LED_G-----10(Arduino UNO)

LED_B-----9(Arduino UNO)

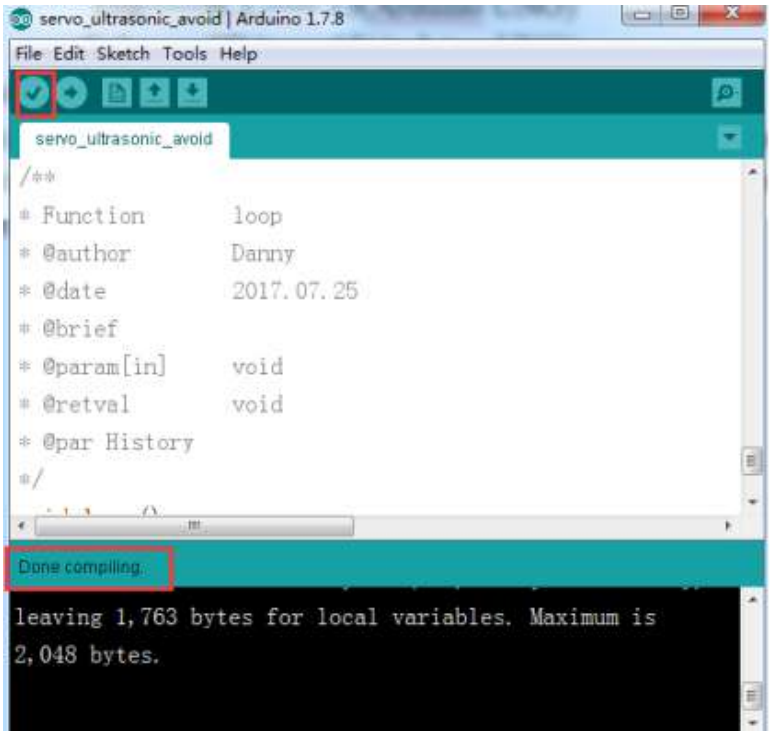
J1---3(Arduino UNO)

Trig-----SCL----- 13(Arduino UNO)

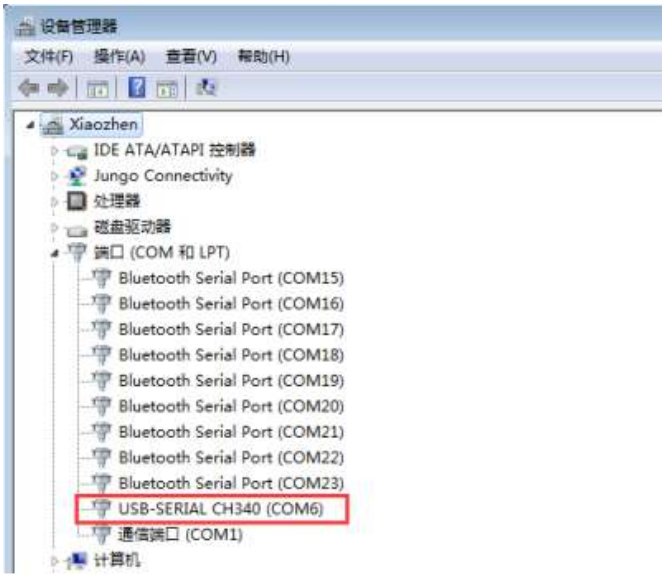
Echo-----SDA-----12(Arduino UNO)

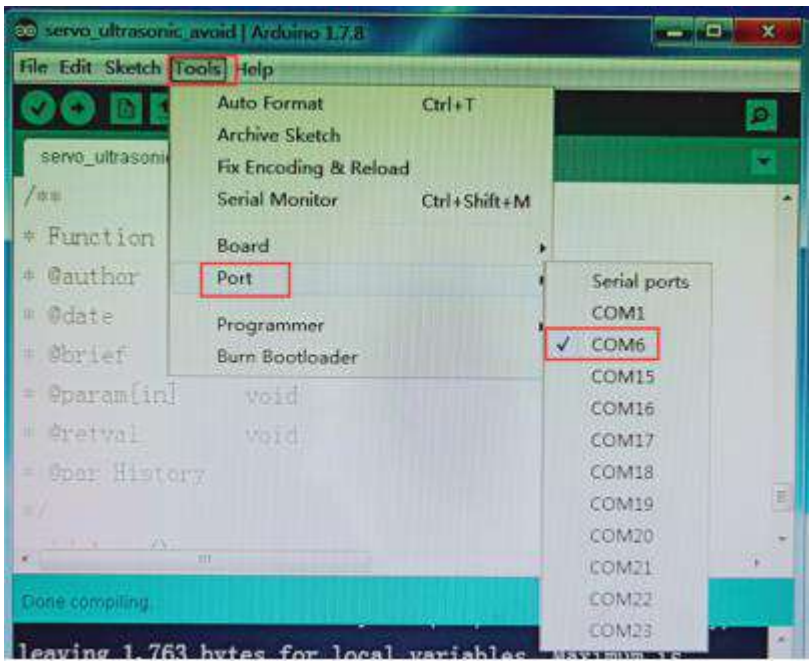
4-3 About the code

1. We need to open the code of this experiment:**servo_ultrasonic_avoid.ino**, 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.

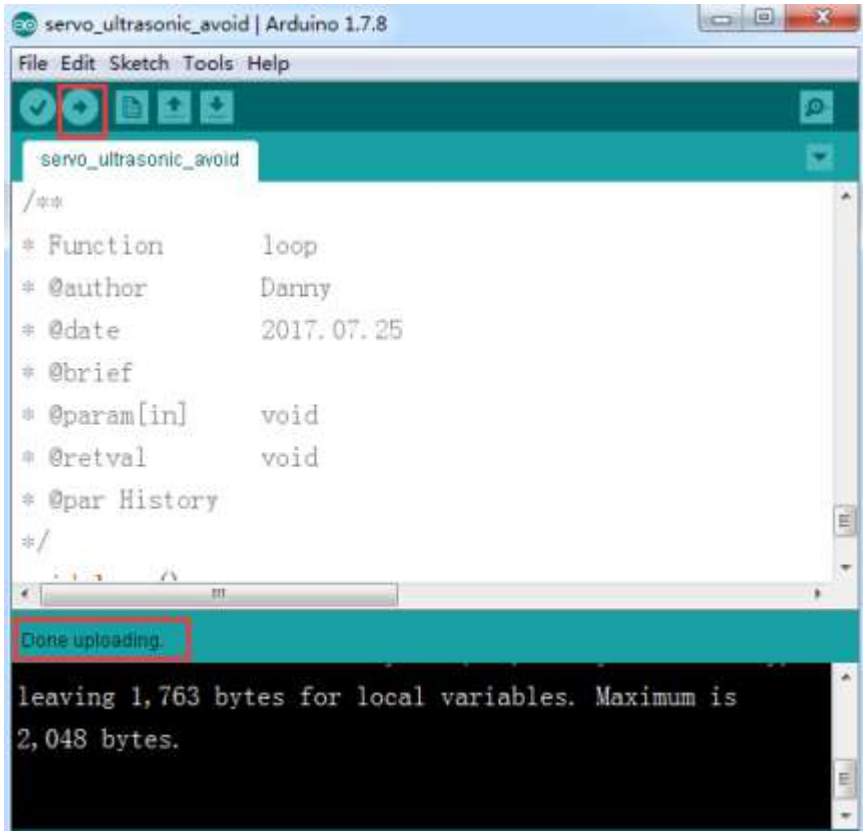


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





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



13- Bluetooth_control

1)Introduction of experimental

In this experiment, we control car by Bluetooth App by Android Mobile phone. The mobile phone sends commands through the serial port to control theadvance, backward, turn left, turn right , stop, any angle control of the servo, outfire, whistle, speed of robot car.

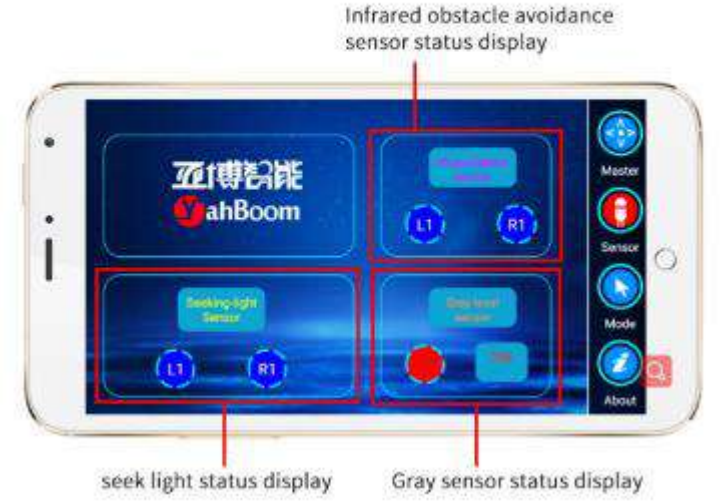
At the same time, the status of various sensors on the robot car and the distance measured by the ultrasonic wave are displayed in real time on the Bluetooth APP interface by the serial port.

2)Experimental Steps

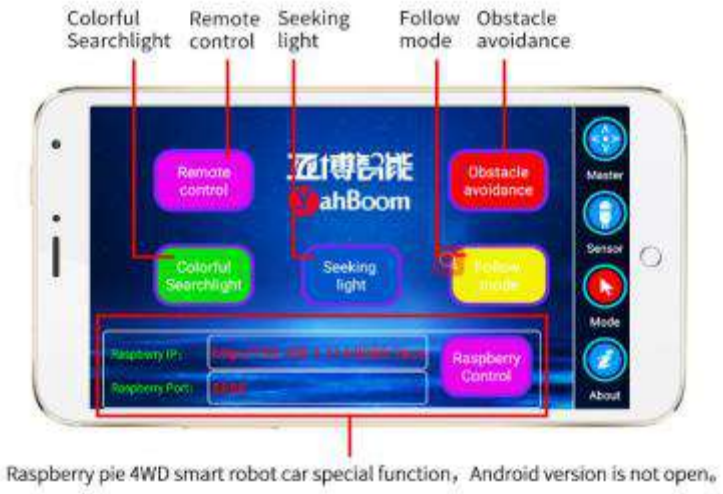
(1) You need to install the 【Yahboom 4WD Controller】 software on yourAndroid mobile phone, turn on the power switch of the car(the red indicating light of the Bluetooth module is flashing). You should open the Bluetooth on yourAndroid mobile phone and open the 【Yahboom 4WD Controller】 software. Bluetooth is automatically connected when the Mobile phone is near the robot.

(2) After successful connection,enter the main control page

Sensor display

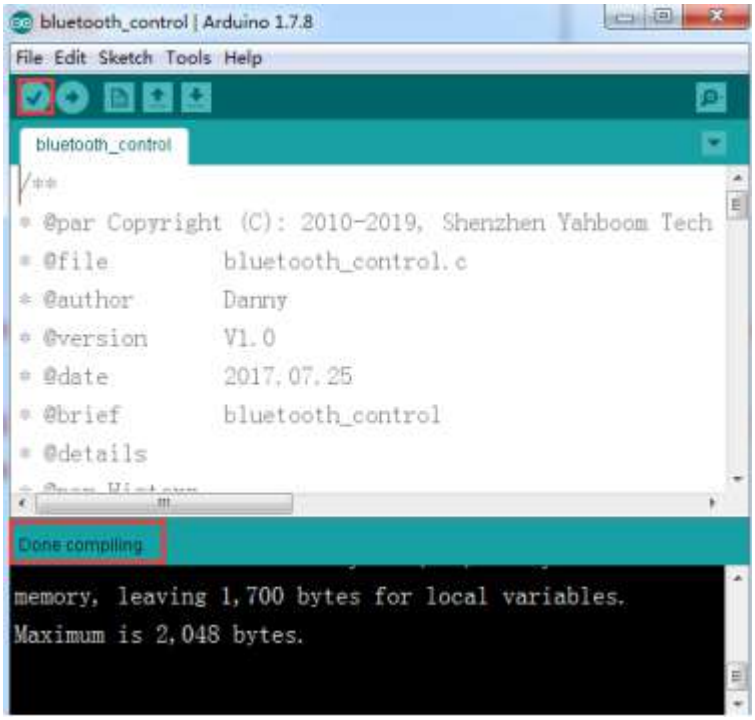


Mode Choice

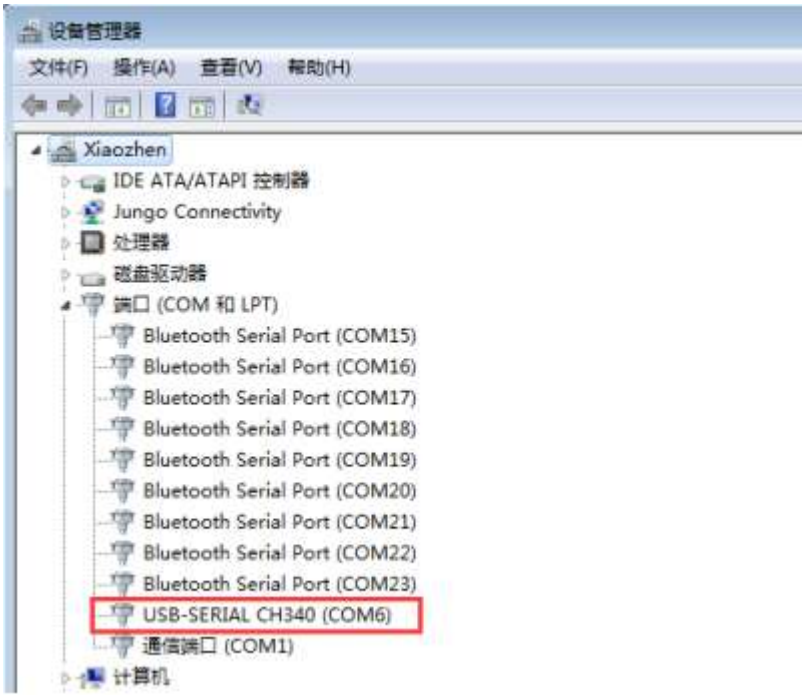


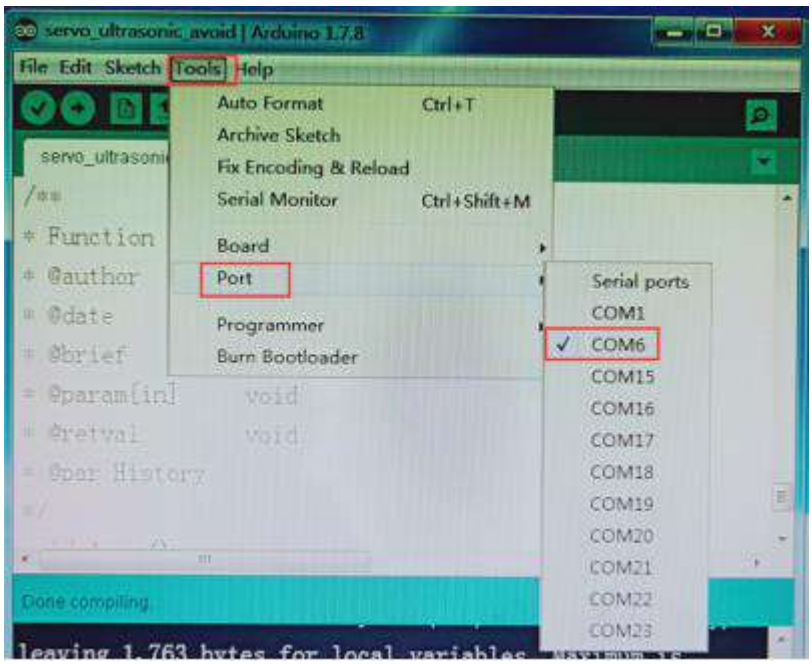
3) About the code

1. We need to open the code of this experiment:**bluetooth_control.ino**, 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.

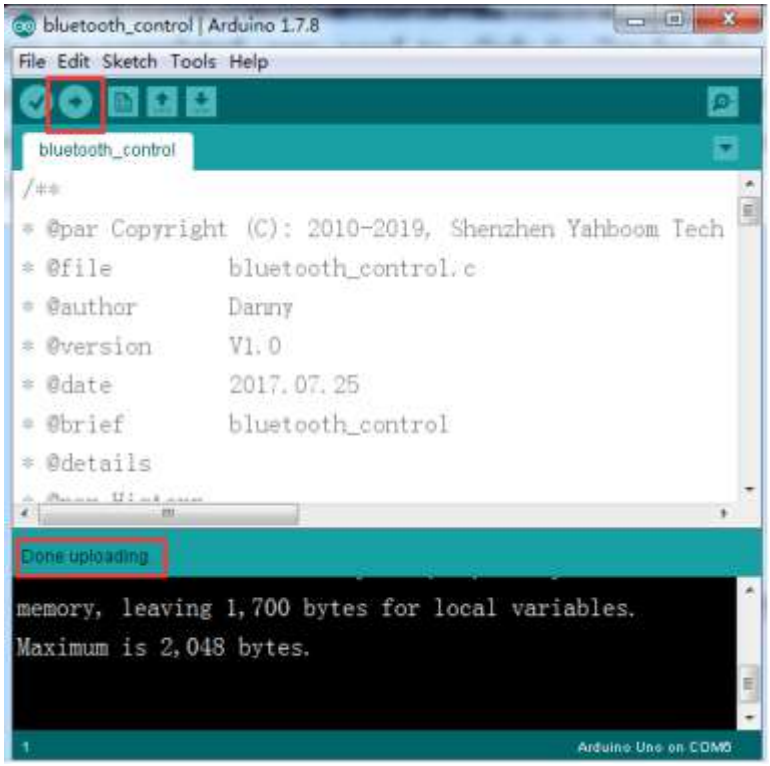


2.In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





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



!!!Note:

1. The Bluetooth module needs to be properly inserted into the expansion board of the robot car.

2. 51/Arduino Download Switch on the expansion board must be set to [OFF].

3. The ultrasonic module must be inserted.