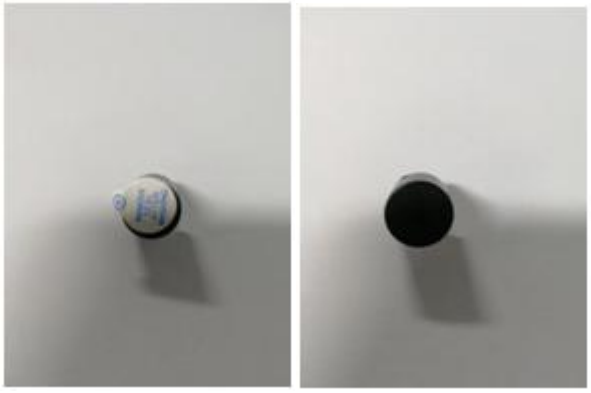
4. keysacnStartCar

The purpose of the experiment:

As with the previous experimental project, upload the program keyssacnStart Ca.ino of this lesson, put the smart car in a spacious area, turn on the power s witch, and the car is still. After pressing the start button, the car starts the spe cified fancy action with a short whistle.

Precautions:

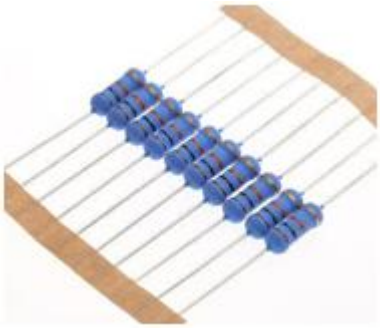
1.A is the active buzzer B is a passive buzzer. This experiment uses an active buzzer for experiments.



A B

List of components required for the experiment:

Arduino Smart Car* 1
USB cable* 1
Active buzzer* 1
DuPont Line* 11
Breadboard* 1
Button * 1
10K resistor * 1



Experimental code analysis:

```
//============================yahboom=======================
====
//   Intelligent car   key star and buzzer alarm experiment
//============================================================
=====
int Left_motor_back=9;        //(IN1)
int Left_motor_go=5;          //(IN2)
int Right_motor_go=6;         //(IN3)
int Right_motor_back=10;      //(IN4)
int key=A0;//Define the key A0 interface
int beep=A1;//Define the buzzer A1 interface
```

```
void setup()
{
  //Initialize the motor drive IO for output mode
  pinMode(Left_motor_go,OUTPUT); // PIN 5 (PWM)
  pinMode(Left_motor_back,OUTPUT); // PIN 9 (PWM)
  pinMode(Right_motor_go,OUTPUT);// PIN 6 (PWM)
  pinMode(Right_motor_back,OUTPUT);// PIN 10 (PWM)
  pinMode(key,INPUT);//Define the key interface for the input interface
  pinMode(beep,OUTPUT);
}
void run(int time)     //car run
{
  digitalWrite(Right_motor_go,HIGH);
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,200);//PWM ratio 0~255 speed control,
                          //the difference of left and right wheel slightly increase or
decrease
  analogWrite(Right_motor_back,0);
  digitalWrite(Left_motor_go,HIGH);
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,200);//PWM ratio 0~255 speed control,
                          //the difference of left and right wheel slightly increase or
decrease
  analogWrite(Left_motor_back,0);
  delay(time * 100);    //execution time, can be adjusted
}

void brake(int time)        //car stop
{
  digitalWrite(Right_motor_go,LOW);
  digitalWrite(Right_motor_back,LOW);
  digitalWrite(Left_motor_go,LOW);
  digitalWrite(Left_motor_back,LOW);
  delay(time * 100);//execution time, can be adjusted
}
void left(int time)         //turn left(left wheel stop，right wheel go)
```

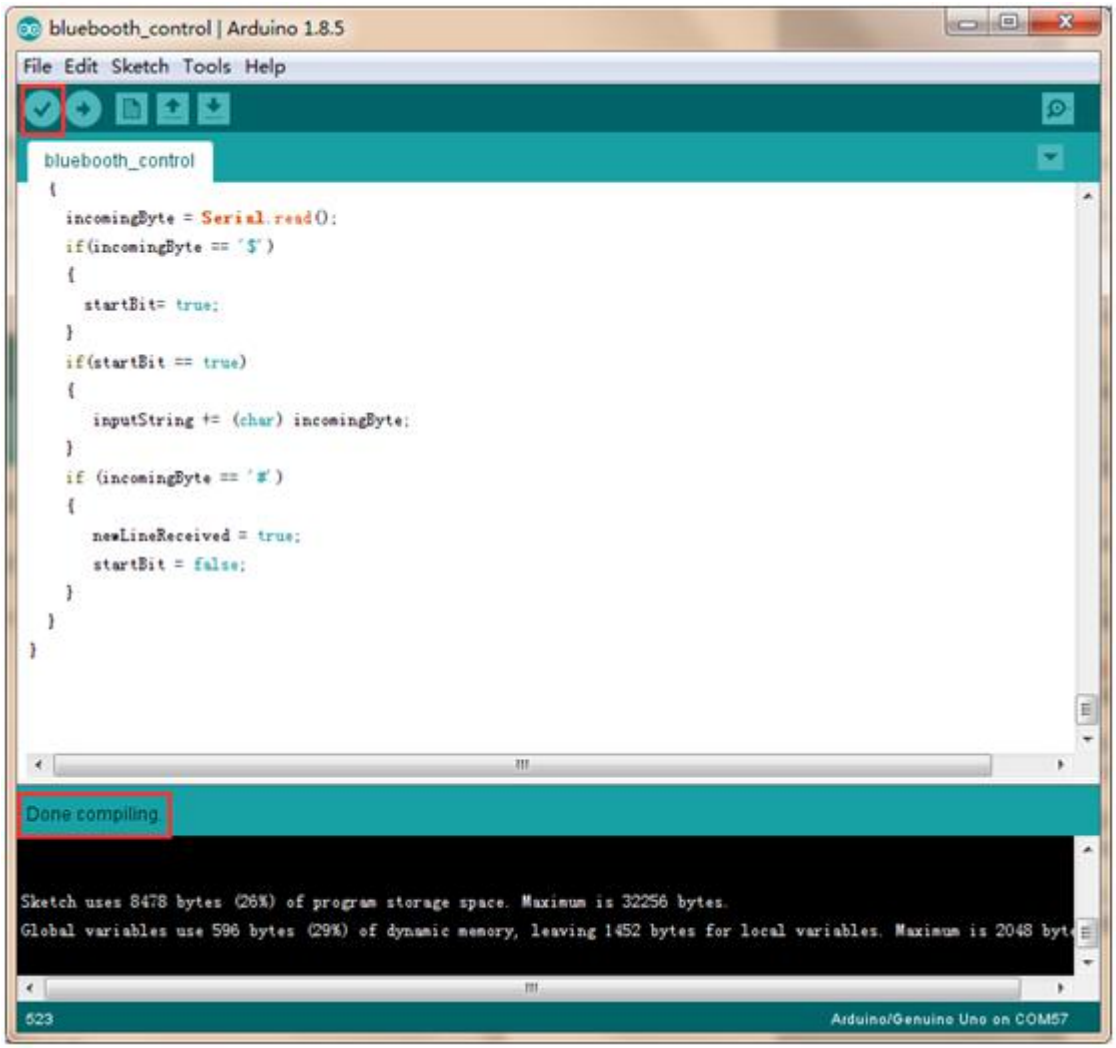
```
{
  digitalWrite(Right_motor_go,HIGH);    // right motor go
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,200);
  analogWrite(Right_motor_back,0);      //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,LOW);      //left wheel back
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,0);       //PWM ratio 0~255 speed control
  delay(time * 100);    //execution time, can be adjusted
}
void spin_left(int time)          //left rotation(left wheel back，right wheel go)
{
  digitalWrite(Right_motor_go,HIGH);    // right motor go
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,200);
  analogWrite(Right_motor_back,0);       //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,LOW);       //left wheel back
  digitalWrite(Left_motor_back,HIGH);
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,200);     //PWM ratio 0~255 speed control
  delay(time * 100);             //execution time, can be adjusted
}
void right(int time)         //turn right(right wheel stop，left wheel go)
{
  digitalWrite(Right_motor_go,LOW);     //right motor back
  digitalWrite(Right_motor_back,LOW);
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,0);      //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,HIGH);     //left motor go
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,200);
  analogWrite(Left_motor_back,0);      //PWM ratio 0~255 speed control
  delay(time * 100);         //execution time, can be adjusted
}
void spin_right(int time)         // right rotation(right wheel back，left wheel go)
```

```
{
  digitalWrite(Right_motor_go,LOW);    //right motor back
  digitalWrite(Right_motor_back,HIGH);
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,200);   //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,HIGH);    //left motor go
  digitalWrite(Left_motor_back,LOW);
  analogWrite(Left_motor_go,200);
  analogWrite(Left_motor_back,0);      //PWM ratio 0~255 speed control
  delay(time * 100);            //execution time, can be adjusted
}
void back(int time)          //back
{
  digitalWrite(Right_motor_go,LOW);   //right wheel back
  digitalWrite(Right_motor_back,HIGH);
  analogWrite(Right_motor_go,0);
  analogWrite(Right_motor_back,150); //PWM ratio 0~255 speed control
  digitalWrite(Left_motor_go,LOW);    //left wheel back
  digitalWrite(Left_motor_back,HIGH);
  analogWrite(Left_motor_go,0);
  analogWrite(Left_motor_back,150);   //PWM ratio 0~255 speed control
  delay(time * 100);        //execution time, can be adjusted
}
void keysacn()
{
  int val;
  val=digitalRead(key);//Read the value of the port 7 level to the val
  while(!digitalRead(key))//When the key is not pressed, circulate all the time
  {
    val=digitalRead(key);//This sentence can be omitted and the circulate can
run away.
  }
  while(digitalRead(key))//When the key is pressed
  {
    delay(10);
    val=digitalRead(key);//Read the value of the port 7 level to the val
```

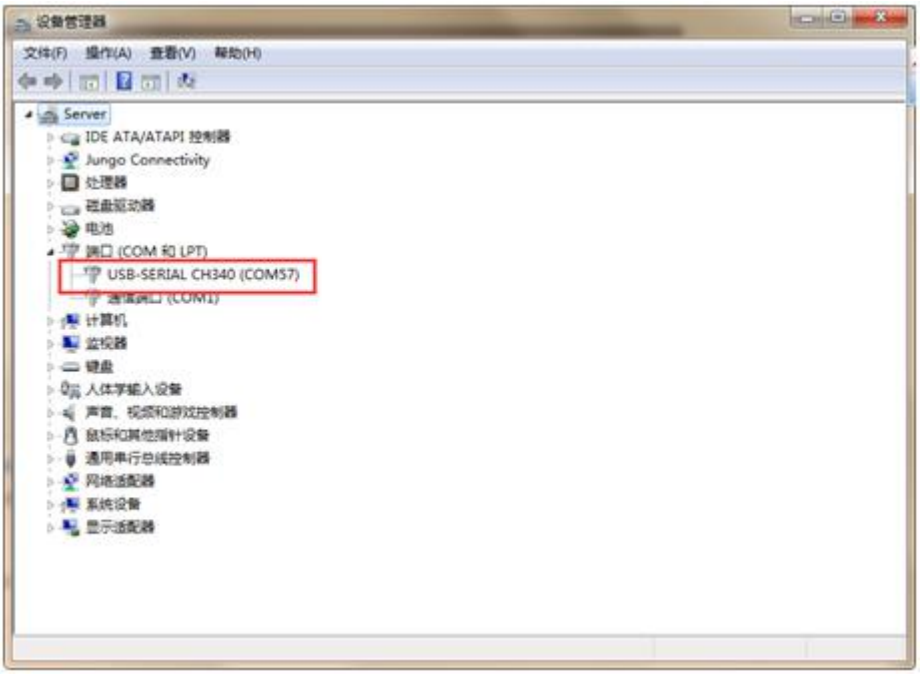
```
    if(val==HIGH)   //Judge whether the key is pressed again
    {
      digitalWrite(beep,HIGH);          //buzzer sound
      while(!digitalRead(key))             //Judge whether the key isreleased
        digitalWrite(beep,LOW);             //buzzer no sound
    }
    else
      digitalWrite(beep,LOW);           //buzzer no sound
  }
}
void loop()
{
  keysacn();       //Call key scan function
  back(10);        //back 1s
  brake(5);        //stop 0.5s
  run(10);         //run 1s
  brake(5);        //stop 0.5s
  left(10);        //turn left 1s
  right(10);       //turn right 1s
  spin_left(20);   //left rotation 2s
  spin_right(20); //right rotation2s
  brake(5);         //stop
}
```

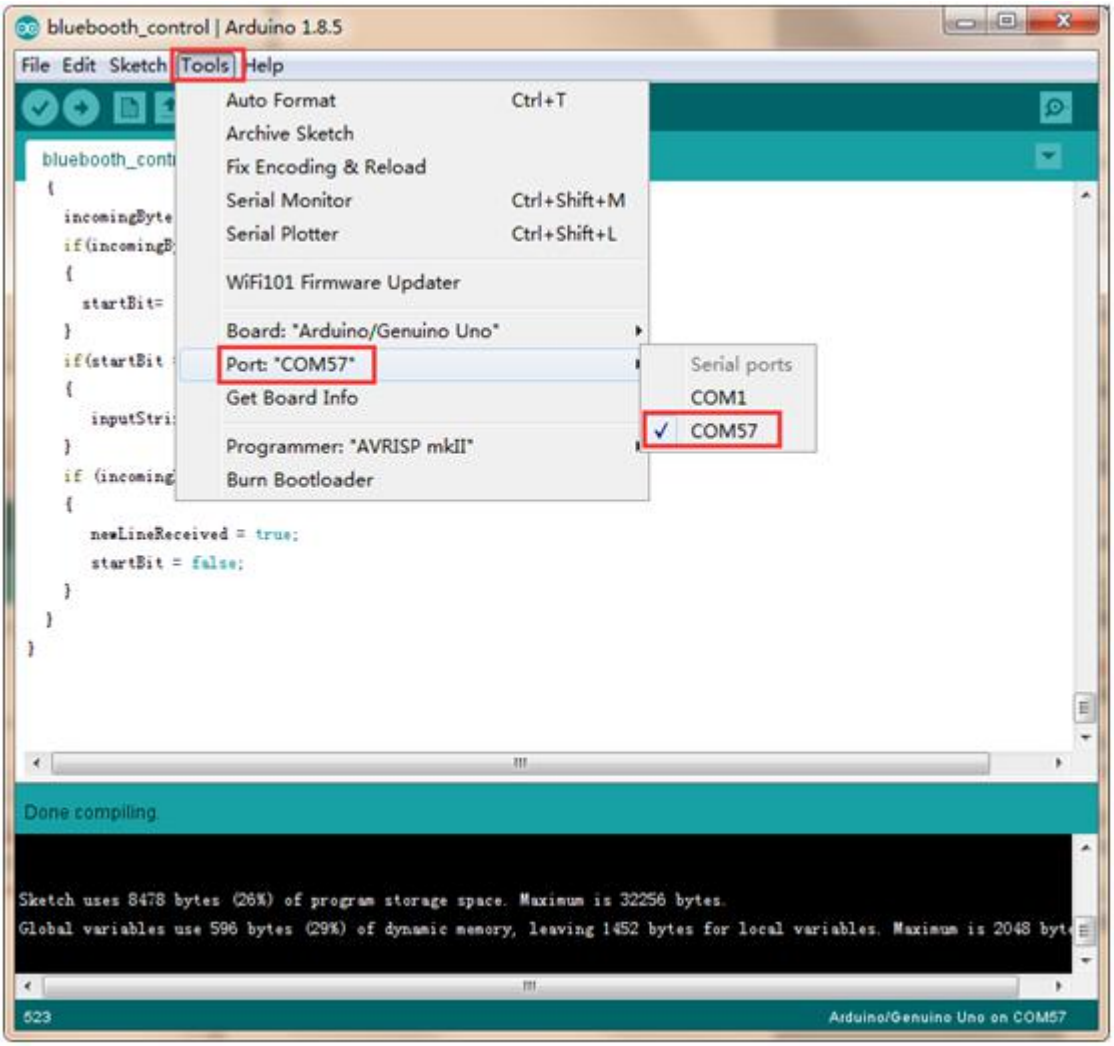
Experimental steps:

1. We need to open the code of this experiment: **keysacnStartCar.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.

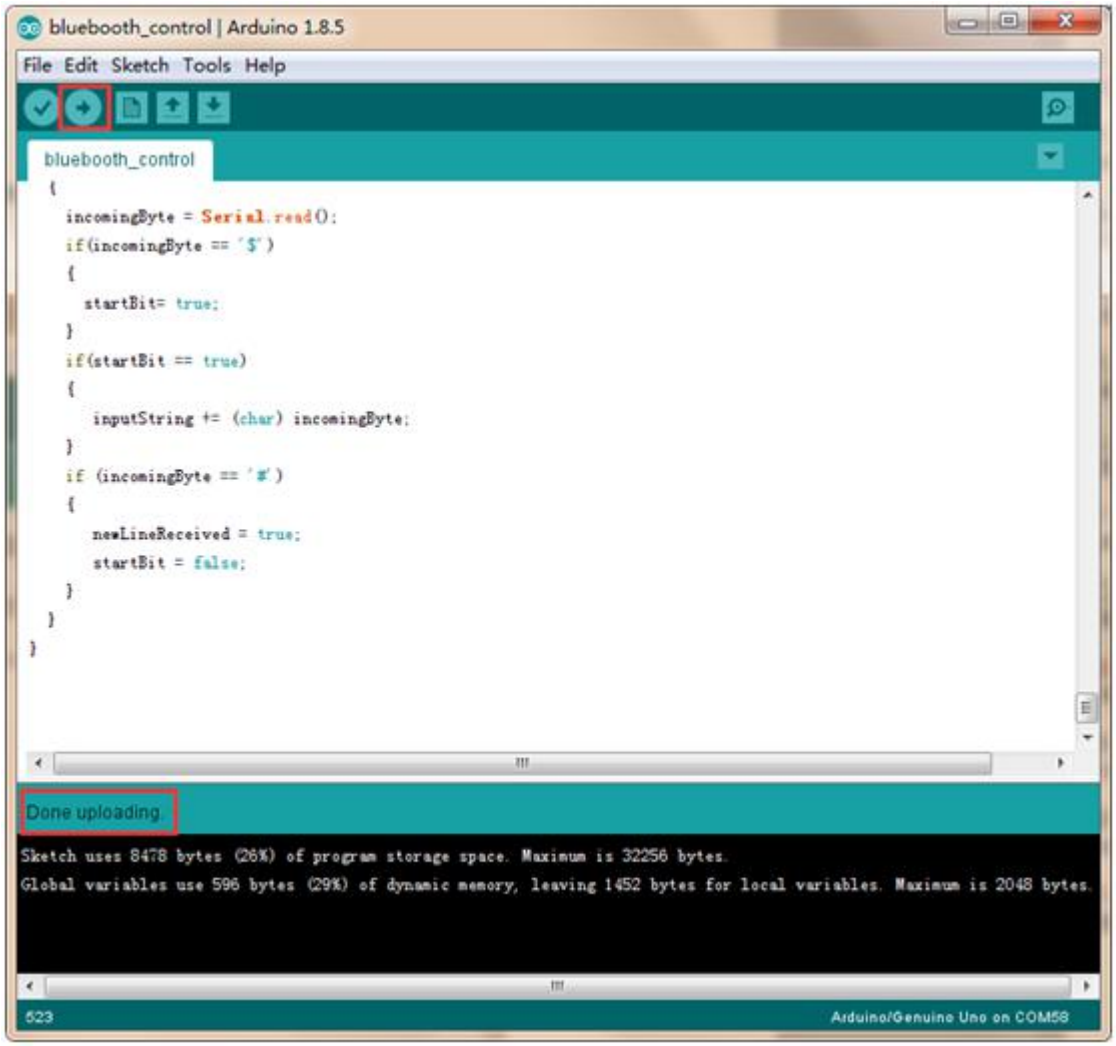


2. In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】---selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.


设备管理器
文件(F) 操作(A) 查看(V) 帮助(H)
Server
IDE ATA/ATAPI 控制器
Jungo Connectivity
处理器
磁盘驱动器
电池
端口 (COM 和 LPT)
USB-SERIAL CH340 (COM57)
通信端口 (COM1)
计算机
监视器
键盘
人体学输入设备
声音、视频和游戏控制器
鼠标和其他指针设备
通用串行总线控制器
网络适配器
系统设备
显示适配器

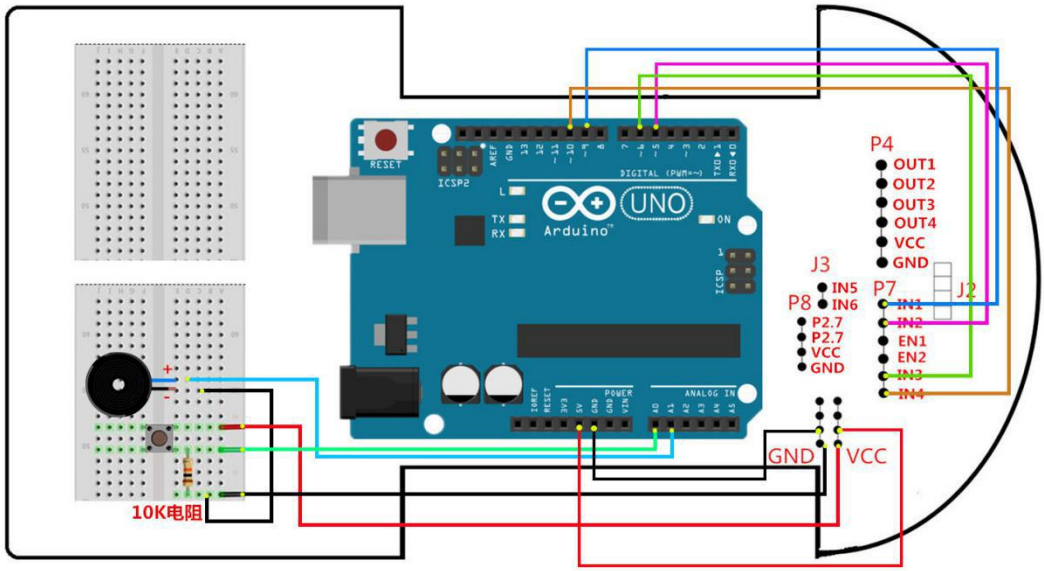

bluebooth_control | Arduino 1.8.5
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM57"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM57
bluebooth_conti
{
incomingByte
if(incomingB
{
startBit=
}
if(startBit
{
inputStri
}
if (incoming
{
newLineReceived = true;
startBit = false;
}
}
}
Done compiling.
Sketch uses 8478 bytes (26%) of program storage space. Maximum is 32256 bytes.
Global variables use 596 bytes (29%) of dynamic memory, leaving 1452 bytes for local variables. Maximum is 2048 byt
523
Arduino/Genuino Uno on COM57

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.Please wire the Smart Car as shown below.

keysacn start car



5.Put the smart car in a spacious area, turn on the power switch, and the car is stationary. After pressing the start button as shown below, the car starts the specified fancy action with a short whistle.