

Lesson 2 Bluetooth Car



Points of this section

It is very important and so cool to control your car wirelessly in a certain space when we learn the Arduino, so in the lesson, we will teach you how to control a car by Bluetooth.

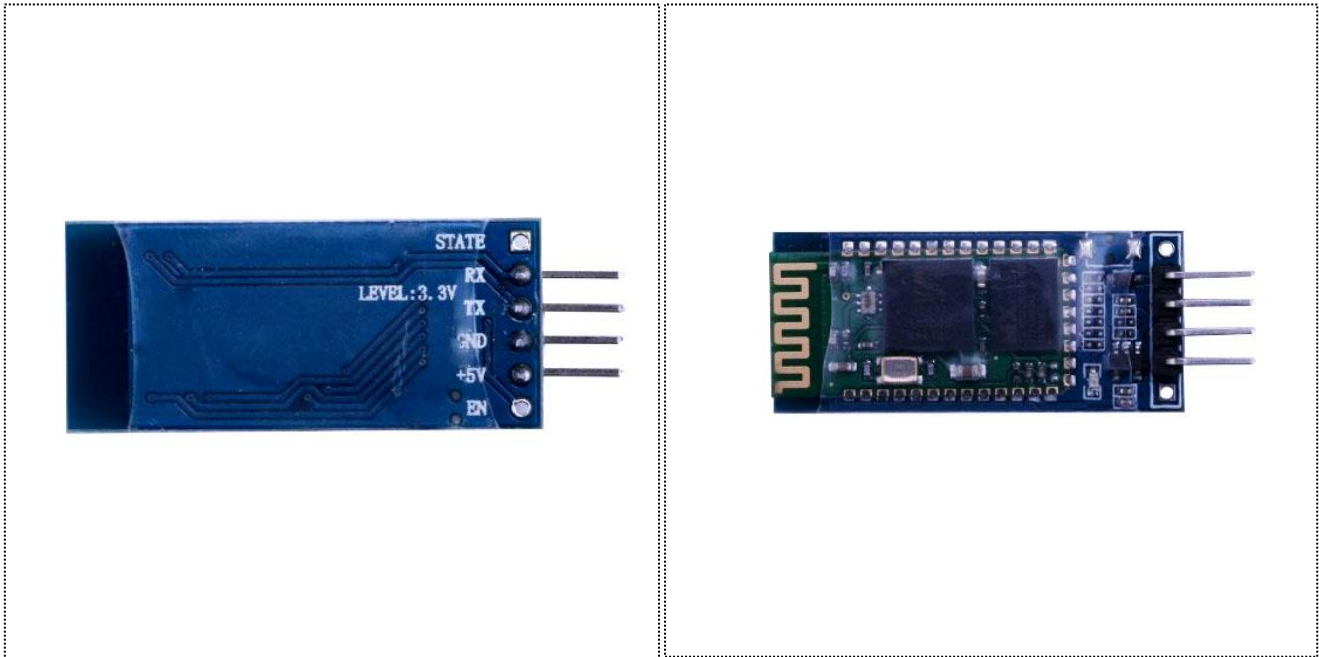
Learning Parts:

- ◆ Learn how to use the HC-06 Bluetooth module and the Bluebooth APP
- ◆ Learn how to control the vehicle via Bluebooth
- ◆ Write programs to implement this function

Preparations:

- ◆ A vehicle (equipped with battery)
- ◆ A USB cable
- ◆ A HC-06 Bluetooth module
- ◆ An Android mobile phone or tablet

I . Bluetooth module



The description of HC-06 module:

1. Adopt mainstream Bluetooth chip of CSR, protocol standard of BluetoothV2.0
2. Analog working voltage of serial port is 3.3V
3. Users can set baud rate 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
4. Dimension of key module is 28mm x 15 mm x 2.35mm
5. Working electric current: 40mA
6. Dormancy electric current: less than 1mA
7. Being used for GPS navigation system, hydroelectric gas reading system, industrial field mining control system
8. Can be connected to Bluetooth laptop, computer with Bluetooth adapter, PDA, etc.

This is the schematic diagram of Bluetooth module connected to UNO controller board:



In the experiment we will connect it to UNO board via expansion board V5.



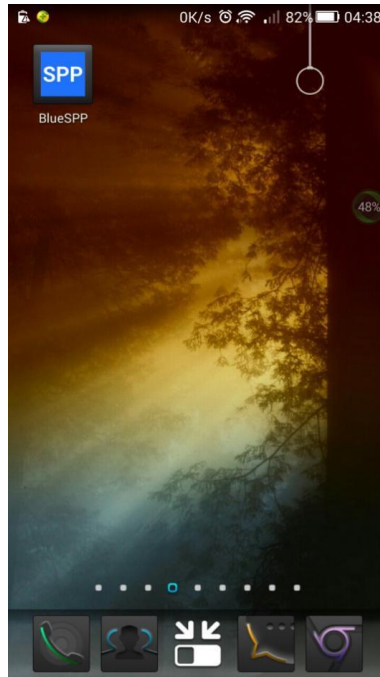
① bluetooth module

II . Bluetooth APP

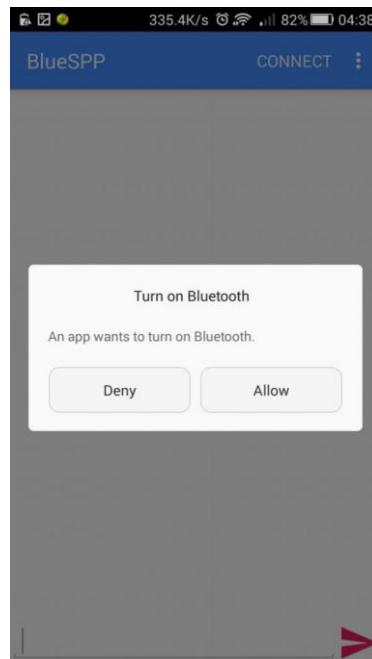
We offer you the APK software for android phone. You can open like below:

	Bluetooth Car.docx	2016/1/28 0:01	Microsoft Word ...	0 KB
	com.shenyao.cn.android.BlueSPP.apk	2016/1/20 9:14	APK 文件	1,876 KB

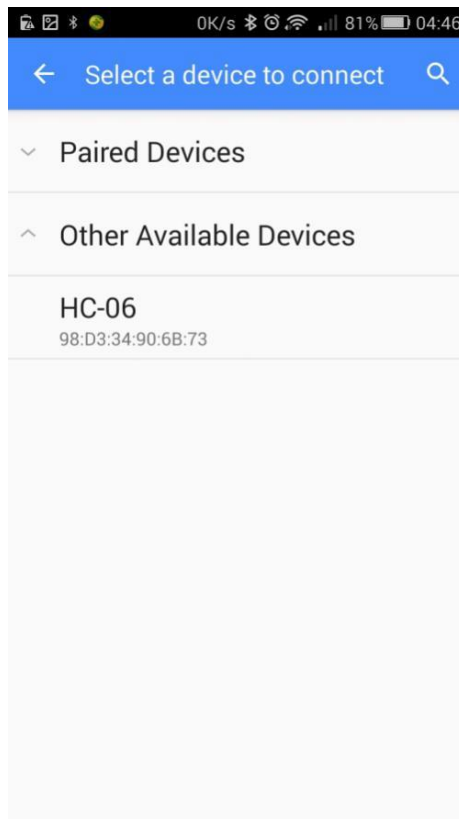
Install the APK and then run it on your phone.



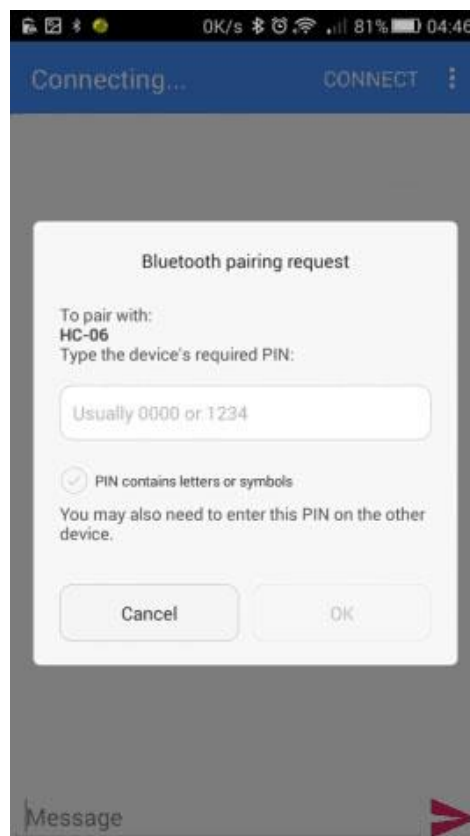
It will show as blow and advise you to turn on Bluetooth function when you run it.



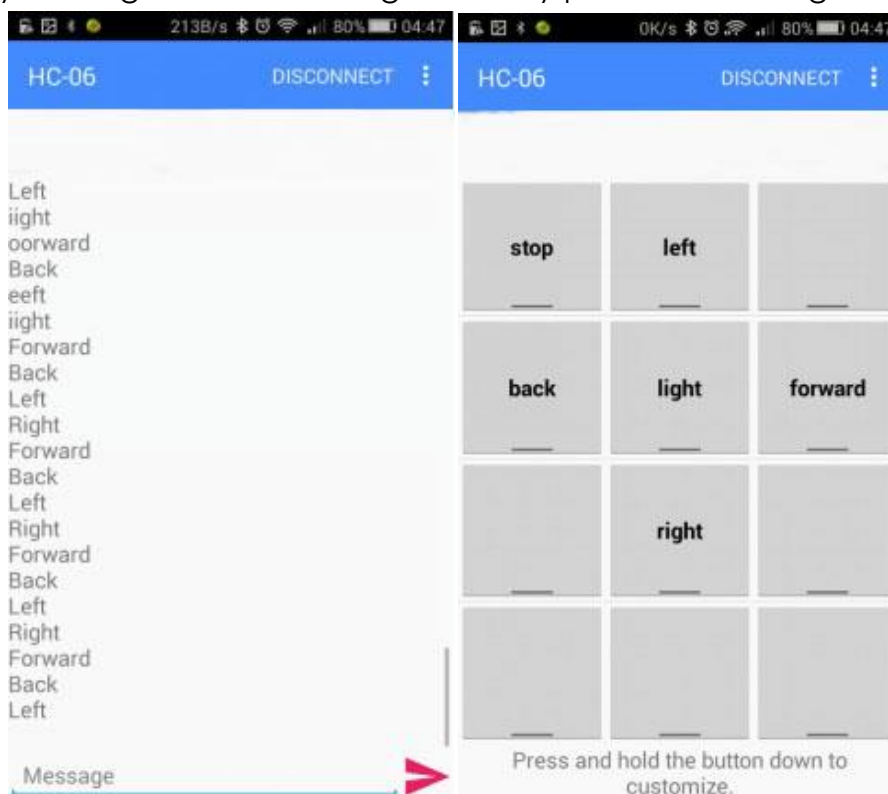
Then you phone will search Bluetooth equipment nearby.



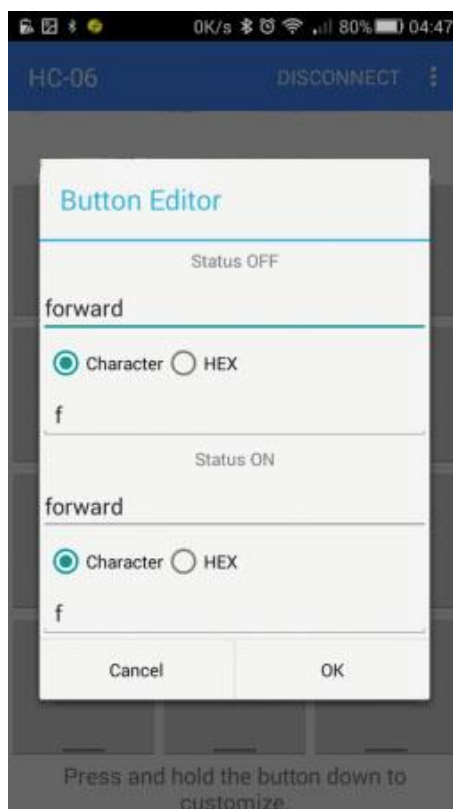
Click HC-06, it will show "password is needed", and it will work when you enter 0000 or 1234.



The app interface will be shown as the left one below, then we can slide the screen to the left side by our finger and we can get the key pattern as the right one below:



Finally, we set the definition of each button, we will take "goforward" for example, please see below, and the rest key-values are set in the same manner.



III. Testing

Testing program is as below:

```
int LED=13;//Define 13 pin of LED
volatile int state = LOW;//The initial state of function is defined as a low level
char getstr;    //Defines function that receives Bluetooth character
void setup()
{
    pinMode(LED, OUTPUT);
    Serial.begin(9600);
}
/*Control LED sub function*/
void stateChange()
{
    state = !state;
    digitalWrite(LED, state);
}
void loop() {
    getstr=Serial.read();//Bluetooth serial port receives data in function
    if(getstr=='A')
    {
        stateChange();
    }
}
```

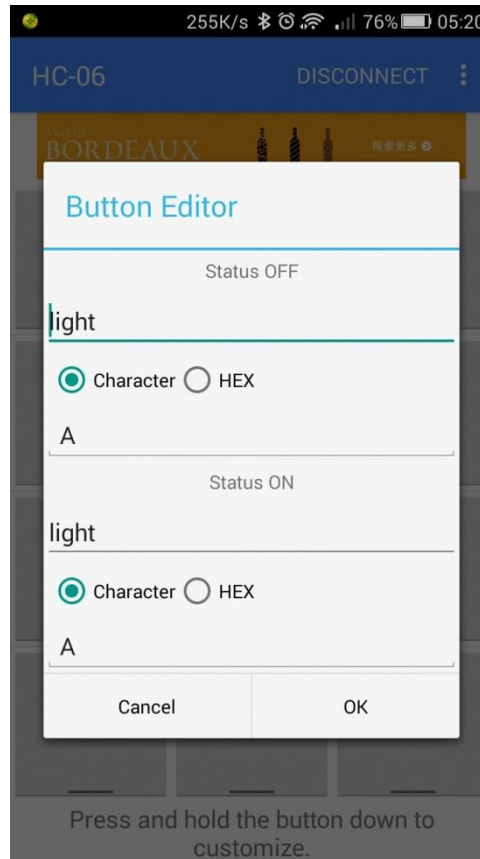
Open the file bluetooth_blink\bluetooth_blink.ino



Upload the program to the UNO board, disconnect it from the computer, and then switch on the car's power supply. (TIPS: The bluetooth module should be pulled out when you upload the program, or it will be failed to upload the program.)

Open APP

After connecting the phone to the car through Bluetooth, we set data as below:



After set-up, press this button. You will find that L light on the UNO board changes with the switch.

The code

```
Serial.begin(9600);
```

The purpose of this block of code is to set the baud rate of the UNO control board as 9600 and open the serial port. In this way, they can communicate with each other, because the original baud rate of the Bluetooth module is 9600.

```
getstr=Serial.read();//The Bluetooth serial port to receive the data in the function  
if(getstr=='A'){  
  stateChange();
```

This function is executed repeatedly within the circulating function. It will first read data from the serial port and then check the data. If it meets the condition, it will

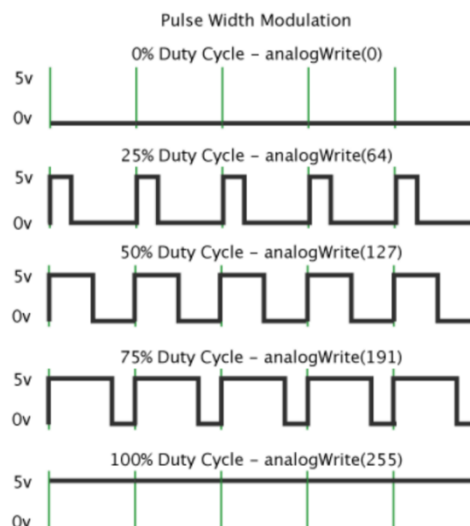
execute the corresponding sub-function. For example, if it reads the letter A from the serial port, it will execute the sub-function responsible for switching on/off the LED light.

IV. Make a Bluetooth Car

When the car turns left or right, it's not necessary to set the speed too fast. On the contrary, we need to control the speed of car. But how to control?

The answer is PWM.

PWM is the abbreviation of "Pulse Width Modulation", is called pulse modulation in short, is an effective technology to control analog circuit with digital output of microprocessor, car is used to change speed of motor by altering duty cycle of a square wave. In other words, connect and break circuit between two sides of motor constantly, is switch of holding motor work, motor will not be off when power is off because of the fast speed. So we can control speed of car if we control specific value of power on time and power off time. The speed of car will be max when circuit is holding still. The speed of car will be minimum if circuit is holding off. The speed of car will be median in half time. PWM is a technology to get analog quantity through digital method. A square wave is formed by digital control, square wave signal only have two state of on and off (That is high-low of digital pins). Simulate voltage changing from 0 to 5V by controlling specific value of duration on and off time. Occupied time of on (That is high level in academy) is called pulse width, so PWM is also called pulse width modulation. Let's learn about PWM through five square waves below.



Green vertical line above represent a period of square wave. The value written into every `analogWrite(value)` corresponds to the percentage, the percentage is also called Duty Cycle, refer to the percentage gotten from specific value between duration high level and low level time in a period. In figure, from top to bottom, the first square wave, duty cycle is 0%, corresponding value is 0. Output circuit current is minimum, motor hold still. The longer duration time is, the bigger circuit current motor gets, the faster the speed is. So, the final one's duty cycle is 100%, corresponding value is 255, motor rotates in full speed. 50% is medium hyponastic rotate speed, 25% is relatively slower, even can't start (The circuit current is relatively big to start motor because of static friction). PWM is mostly used to adjust light of LED and rotate speed of motor, wheel speed controlled by motor is easily be controlled. The advantage of PWM can be more reflected when you play with some Arduino cars.

```
analogWrite(pin,value);
```

`analogWrite()` is used to write analog value of 0 to 255 for PWM ports. What you need to note is that, `analogWrite()` is only used to digital pins with function of PWM. Pins with function of PWM in UNO are only digital pins of 3, 5, 6, 9, 10, 11.

Our car's speed is controlled by connecting pin5 and pin10 of ENA and ENB. The program below, have set a digital function `int ABS=135;`

The speed is controlled in below program, so you can control the speed on your own.

```
analogWrite(ENA,ABS);
```

```
analogWrite(ENB,ABS);
```

After learning the basic knowledge, we will upload the program as below to the car, open the file `bluetooth_car\bluetooth_car.ino`.

```
int LED=13;
```

```
volatile int state = LOW;
```

```
char getstr;
```

```
int in1=9;
```

```
int in2=8;
```

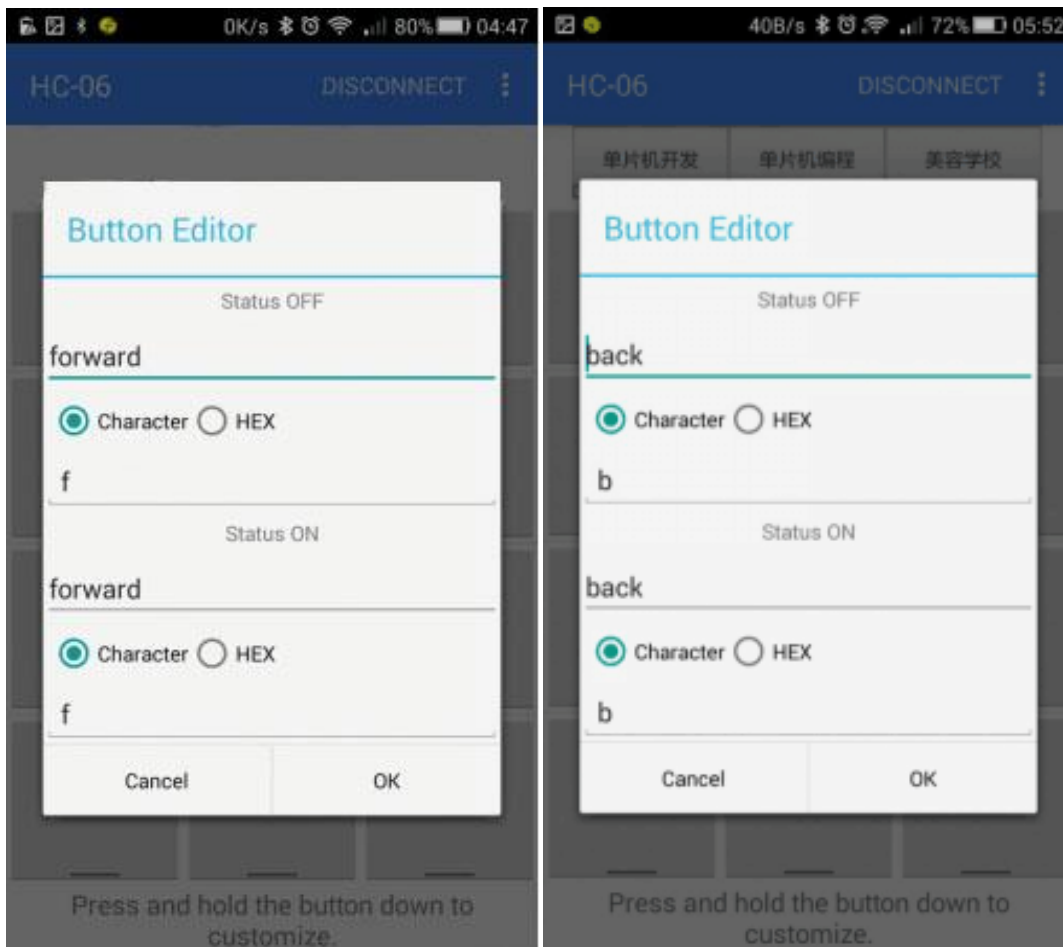
```
int in3=7;
int in4=6;
int ENA=10;
int ENB=5;
int ABS=135;
void _mForward()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("go forward!");
}
void _mBack()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("go back!");
}
void _mleft()
{
    analogWrite(ENA,ABS);
    analogWrite(ENB,ABS);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
```

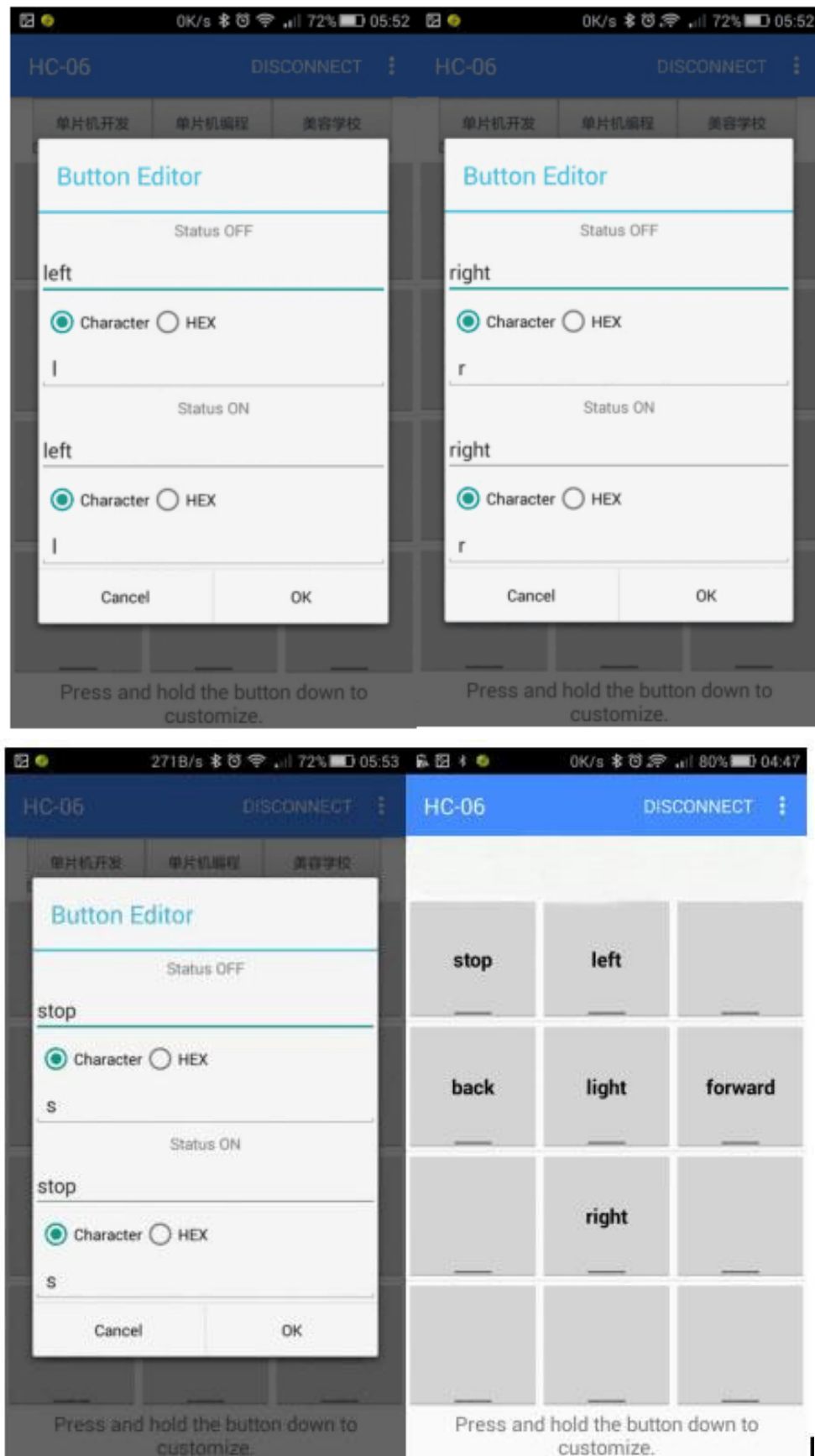
```
    Serial.println("go left!");  
}  
void _mright()  
{  
    analogWrite(ENA,ABS);  
    analogWrite(ENB,ABS);  
    digitalWrite(in1,HIGH);  
    digitalWrite(in2,LOW);  
    digitalWrite(in3,LOW);  
    digitalWrite(in4,HIGH);  
    Serial.println("go right!");  
}  
void _mStop()  
{  
    digitalWrite(ENA,LOW);  
    digitalWrite(ENB,LOW);  
    Serial.println("Stop!");  
}  
void stateChange()  
{  
    state = !state;  
    digitalWrite(LED, state);  
}  
void setup()  
{  
    pinMode(LED, OUTPUT);  
    Serial.begin(9600);  
    pinMode(in1,OUTPUT);  
    pinMode(in2,OUTPUT);  
    pinMode(in3,OUTPUT);  
    pinMode(in4,OUTPUT);  
    pinMode(ENA,OUTPUT);  
    pinMode(ENB,OUTPUT);
```

```
_mStop();  
}  
void loop()  
{  
  getstr=Serial.read();  
  if(getstr=='f')  
  {  
    _mForward();  
  }  
  else if(getstr=='b')  
  {  
    _mBack();  
    delay(200);  
  }  
  else if(getstr=='l')  
  {  
    _mleft();  
    delay(200);  
  }  
  else if(getstr=='r')  
  {  
    _mright();  
    delay(200);  
  }  
  else if(getstr=='s')  
  {  
    _mStop();  
  }  
  else if(getstr=='A')  
  {  
    stateChange();  
  }  
}
```

Upload the program to the UNO control board, switch on the power supply of the vehicle and put it on the ground.

Open the mobile APP, and set up parameters as follows.





Voila, now we can control the car by Bluetooth and play with it.