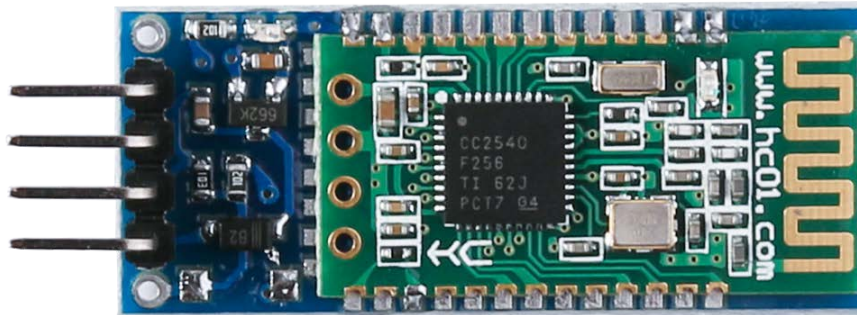


Lesson 2 Bluetooth Car



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

Preparations:

- ◆ A vehicle (equipped with battery)
- ◆ A USB cable
- ◆ A Bluetooth module
- ◆ An iPhone or tablet

I . Bluetooth module

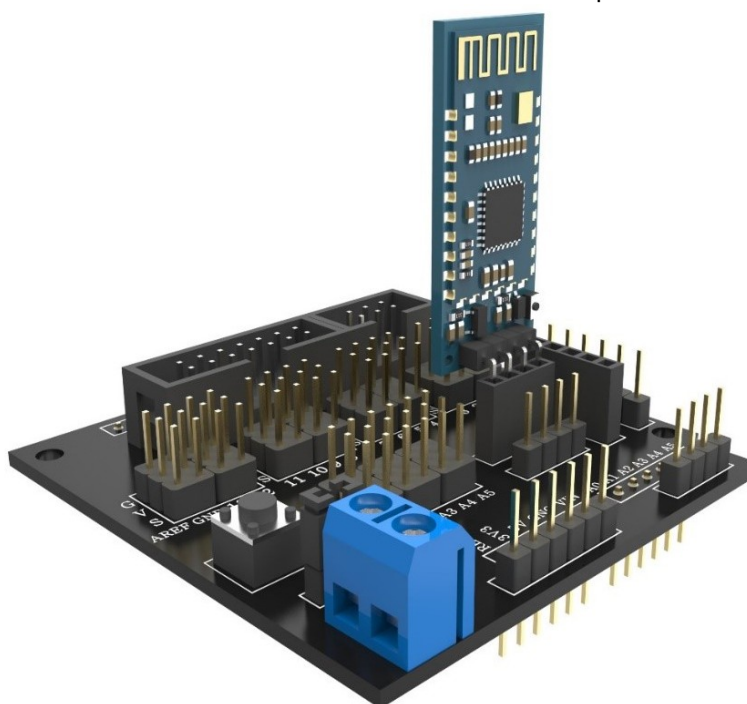


The description of Bluetooth module:

1. Adopt mainstream Bluetooth chip of TI, protocol standard of BluetoothV4.0
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.



II. Getting Started with the Bluetooth APP

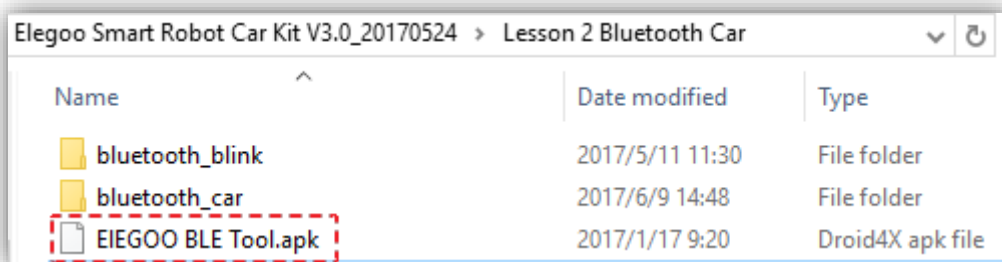
Before beginning, connect the HC-08 Bluetooth module to the expansion board and turn on the power.

STEP1: Install the application.

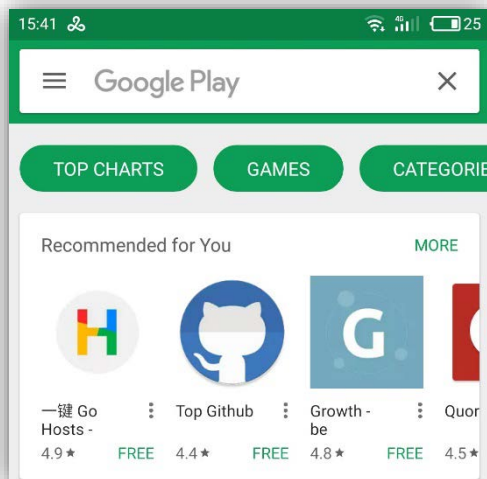
For Android

There are two ways to install the application.

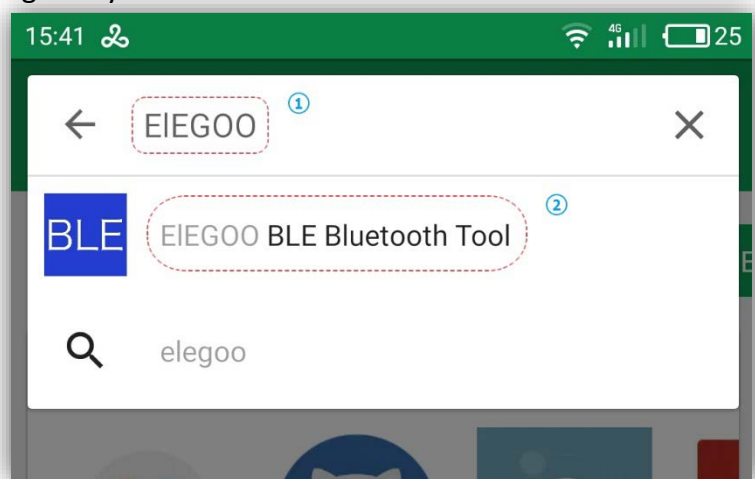
1. Copy the “EIEGOO BLE Tool.apk” file to the Android products and install it.



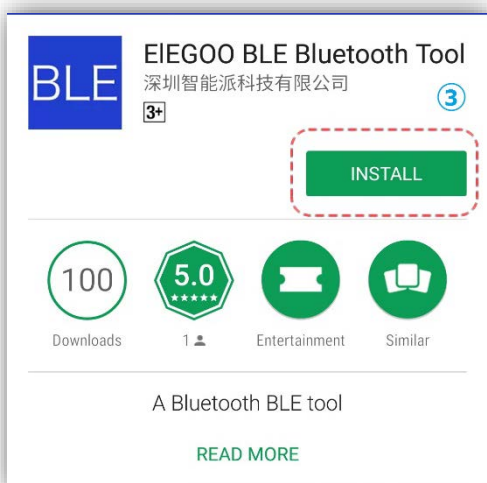
2. Search “EIEGOO BLE Bluetooth Tool” in Google Play Store and install it.



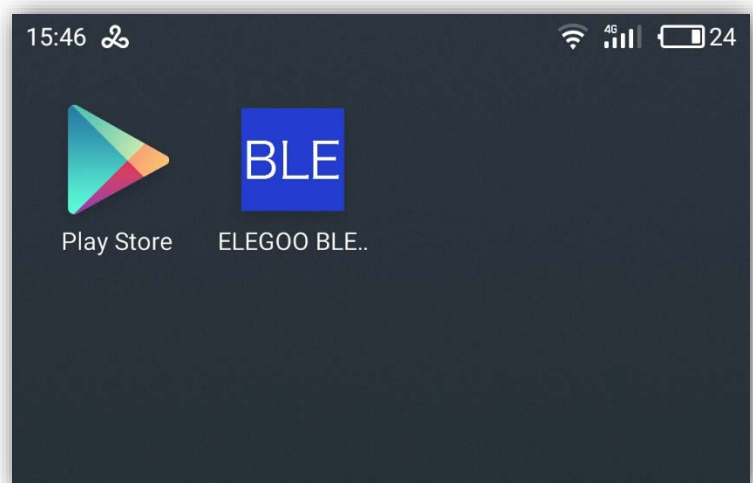
1. Launch Google Play



2. Search “EIEGOO BLE Bluetooth Tool”



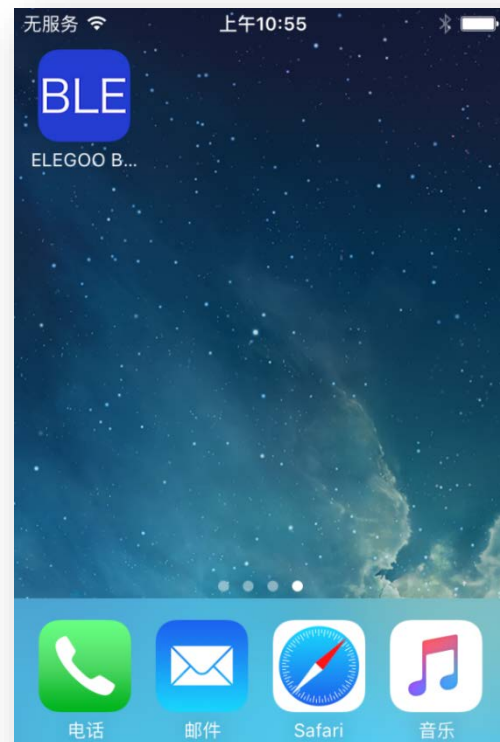
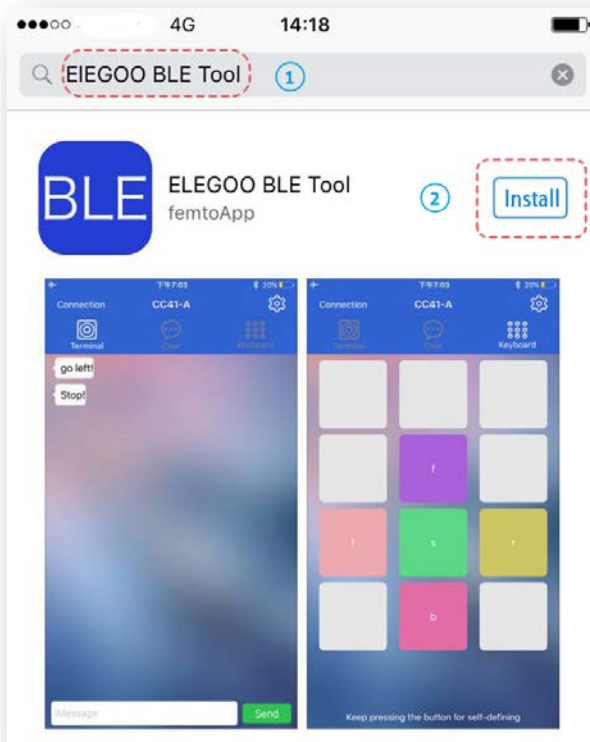
3. INSTSL



4. Finish the

For iOS

Search “ELEGOO BLE Tool” in Apple APP Store and install it.

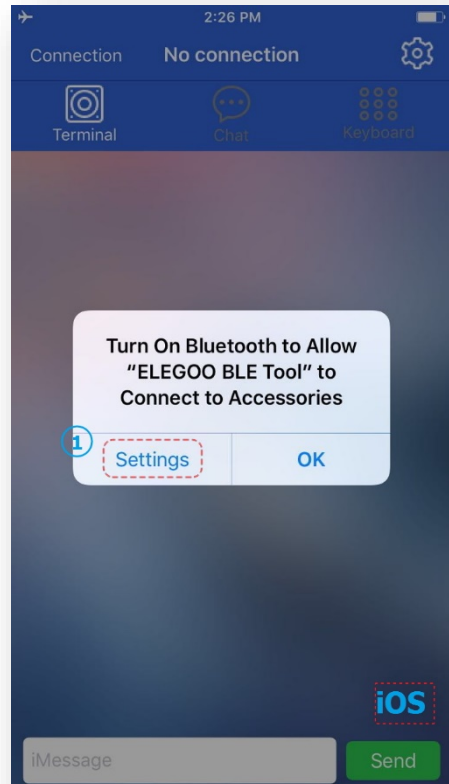
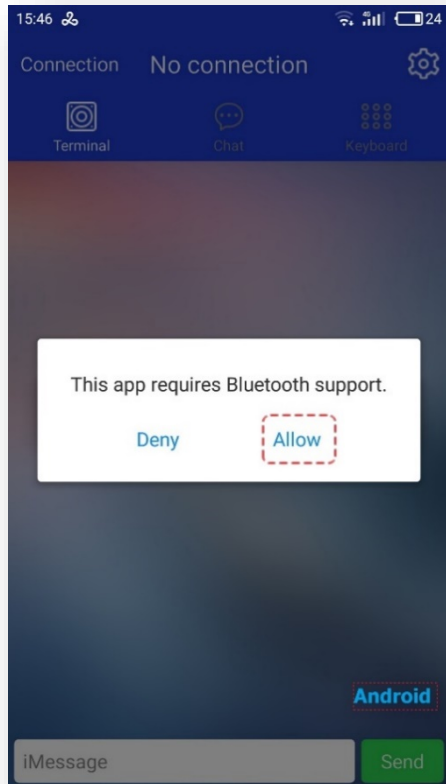


STEP2: Application Settings.

Launching the application.



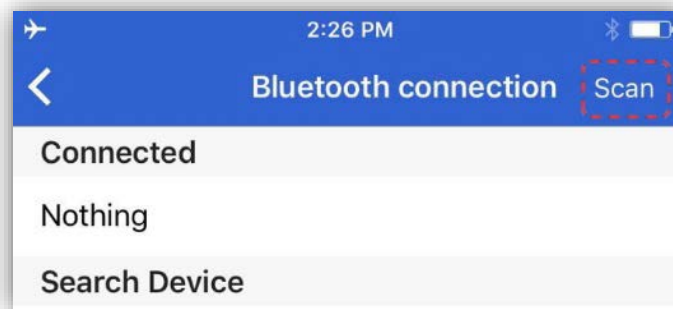
If you don't open your Bluetooth, it will show as blow and advise you to turn on Bluetooth function. For Android, click “Allow” that will open the bluetooth automatically. For iOS, click “Settings” that will jump to bluetooth setup interface, you should turn on the bluetooth and then return to the “ELEGOO BLE Tool” APP interface.



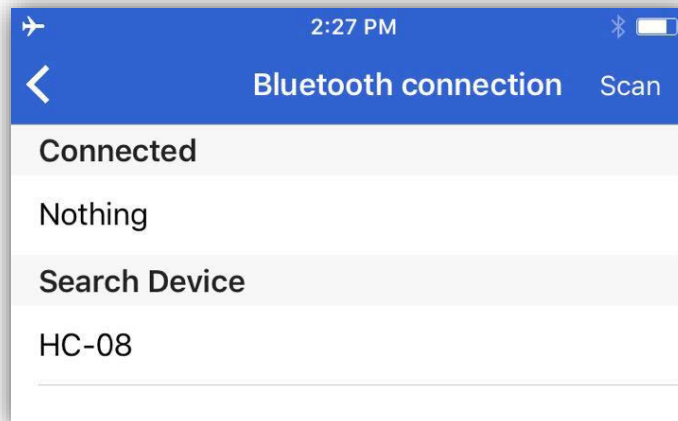
After turn on Bluetooth, the bluetooth icon appears on the APP interface. Click “Connection”.



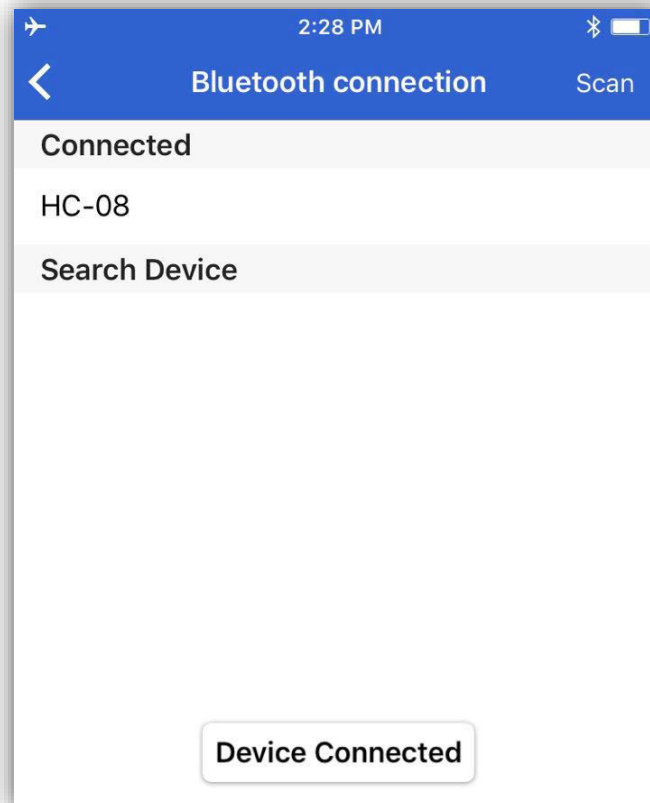
Scan the Bluetooth signal.



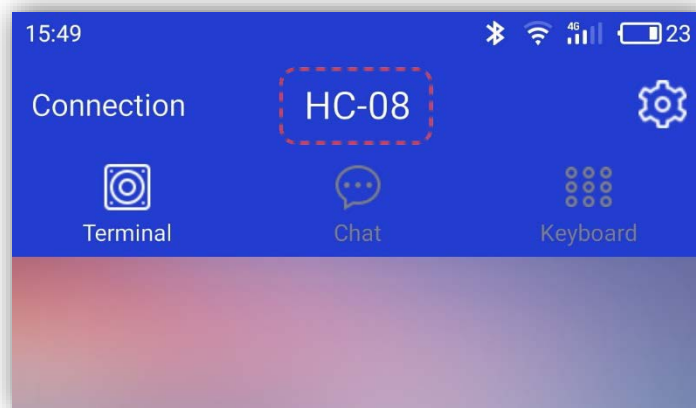
Then your phone will search Bluetooth equipment nearby. HC-08 device appears.



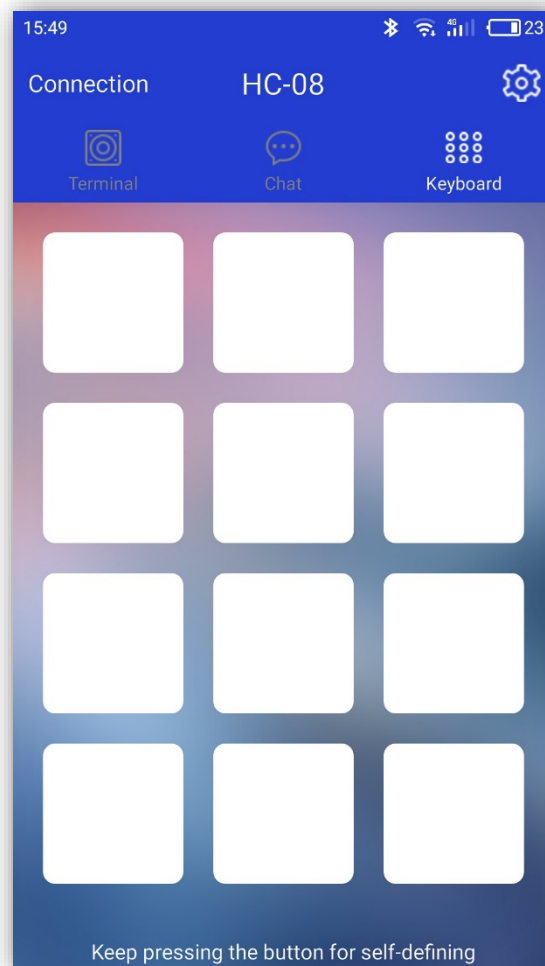
Click the Bluetooth name "HC-08", when the connection is successful, the screen will be displayed "Device Connected".



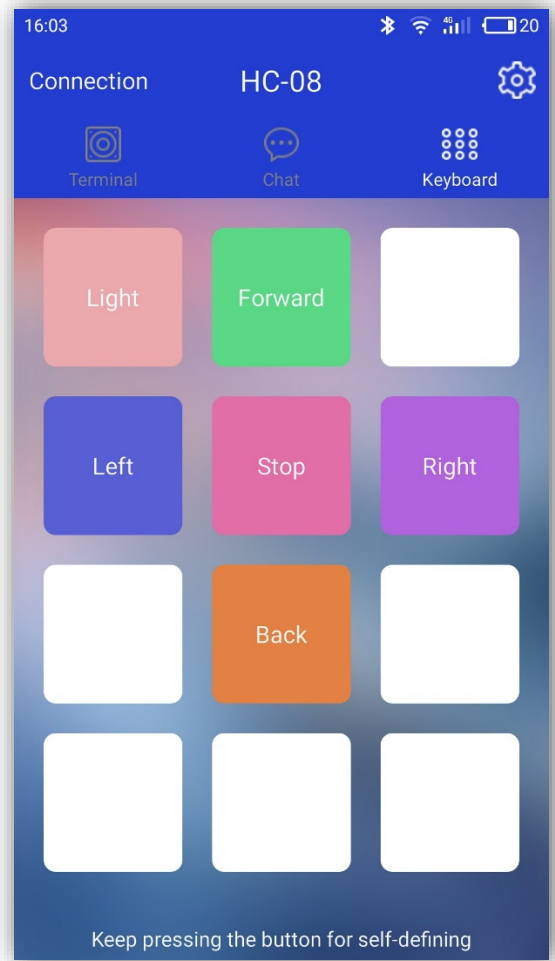
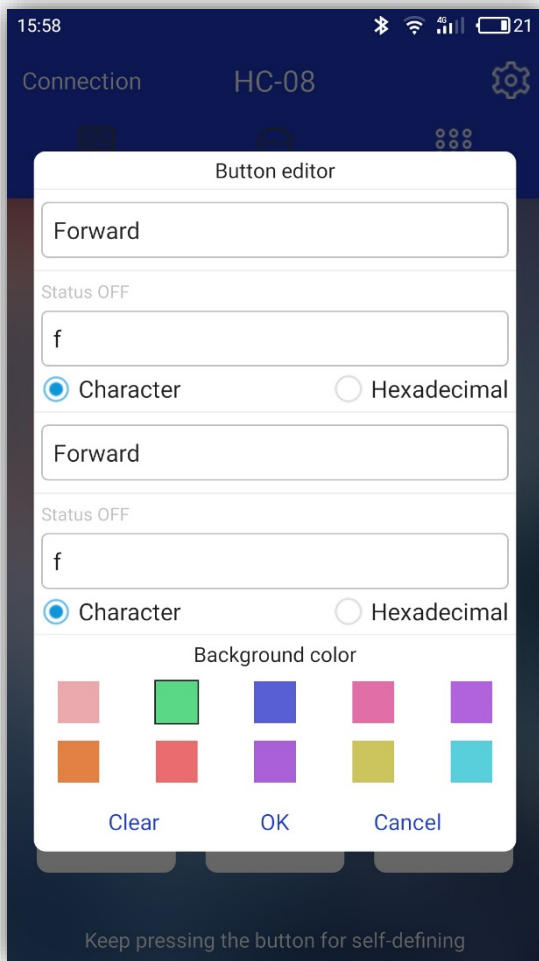
Returns the main interface of the application. After the Bluetooth connection, the Bluetooth name will be displayed on the screen.



Then we can slide the screen to the right side by our finger and we can get the key pattern as below:



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

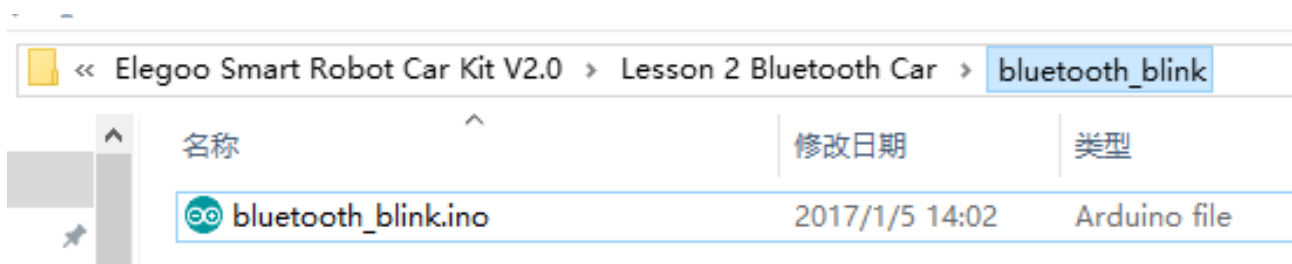


III. Testing

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:

The screenshot shows a mobile application titled "Button editor". It contains two sections for configuring a button's behavior. The first section is labeled "Status OFF" and the second is labeled "Status ON". Each section has a text input field containing the letter "A", followed by two radio button options: "Character" (which is selected) and "Hexadecimal". Below these sections is a "Background color" section displaying a grid of ten colored squares. At the bottom of the screen are three buttons: "Clear", "OK", and "Cancel".

Button editor

LED

Status OFF

A

☒ Character ☐ Hexadecimal

A

Status ON

A

☒ Character ☐ Hexadecimal

Background color

Clear OK Cancel

After set-up, press this button. You will find that 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();
```

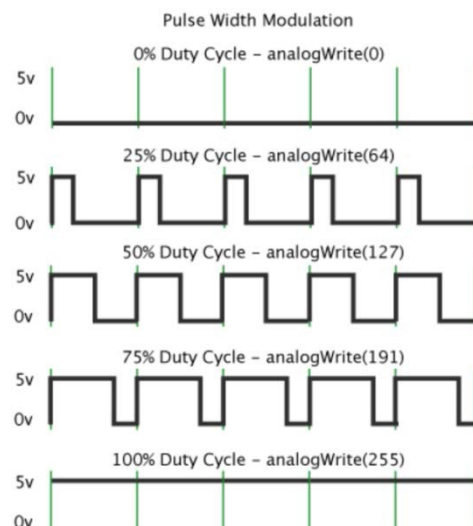
This function is executed repeatedly within the circulating function. It will first read data from the serial port and then check the data. If it meets the condition, it will execute the corresponding sub-function. For example, if it reads the letter A from the serial port, it will execute the sub-function 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.

Button editor

Forward

Status OFF

f

☒Character

☐Hexadecimal

Forward

Status ON

f

☒Character

☐Hexadecimal

Background color

Clear

OK

Cancel

Button editor

back

Status OFF

b

☒ Character ☐ Hexadecimal

back

Status ON

b

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

Left

Status OFF

I

☒ Character ☐ Hexadecimal

Left

Status ON

I

☒ Character ☐ Hexadecimal

Background color

Clear

OK

Cancel

Button editor

Right

Status OFF

r

☒ Character ☐ Hexadecimal

Right

Status ON

r

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

Stop

Status OFF

s

☒ Character ☐ Hexadecimal

Stop

Status ON

s

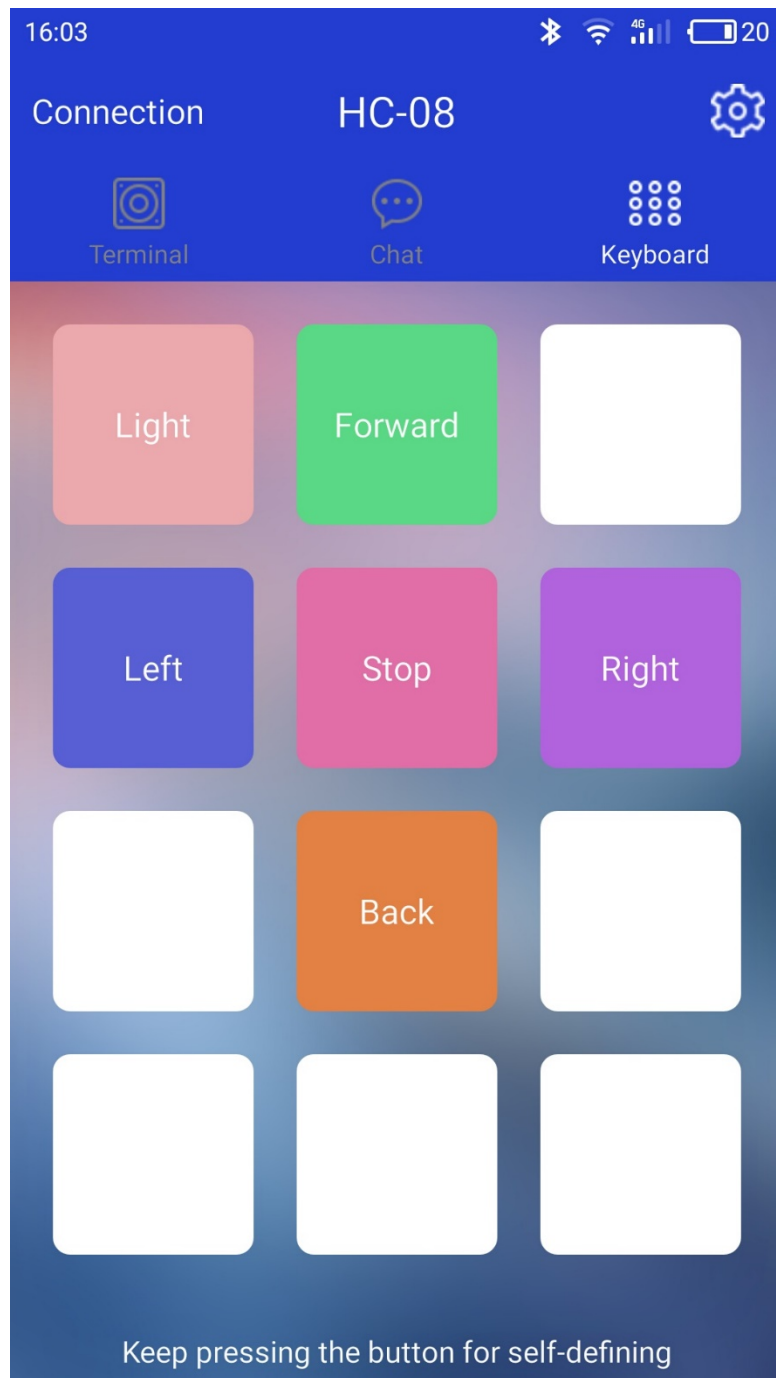
☒ Character ☐ Hexadecimal

Background color

Clear

OK

Cancel



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