

## Lesson 5 Bluetooth Car

---

### ***Points of this section***

*In the previous lessons, we learned how to make those components on the car work alone. that's good, but not enough, in this lesson, we'll do something cooler, put all these features in one single App, and do more DIY things.*

### ***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:***

- ◆ ***Smart Car (with battery)***
- ◆ ***USB cable***
- ◆ ***Bluetooth module***
- ◆ ***Smartphone (iOS or Android)***

## I . Bluetooth module

The Bluetooth is a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wave UHF radio waves in the industrial, scientific, and medical radio bands (2.400 to 2.485 GHz), and building personal area networks (PANs). There are also RF protocols such as ZigBee and Wi-Fi.



In Smart Car Kit, we use the Bluetooth module model "HC-08", it can send serial data to other devices via Bluetooth

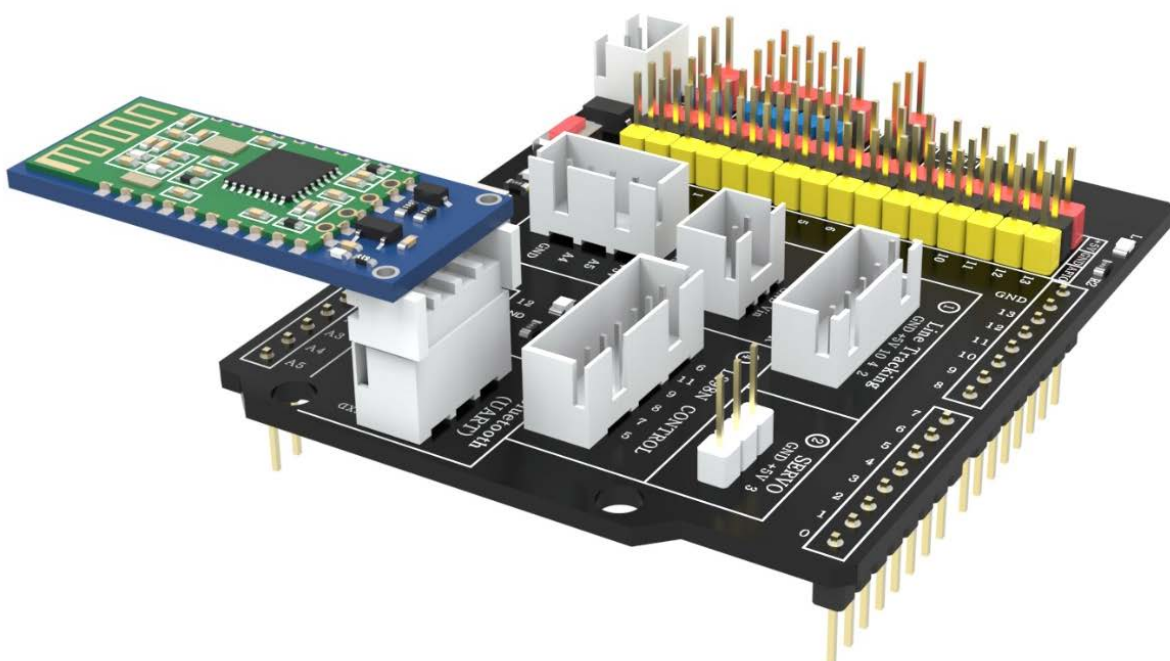


HC-08 communicates with UNO through the **RX/TX** pin on the shield.

**\*WARNING:**

The RX/TX pin will be occupied when the sketch is uploading.

Please unplug the Bluetooth module until done uploading.

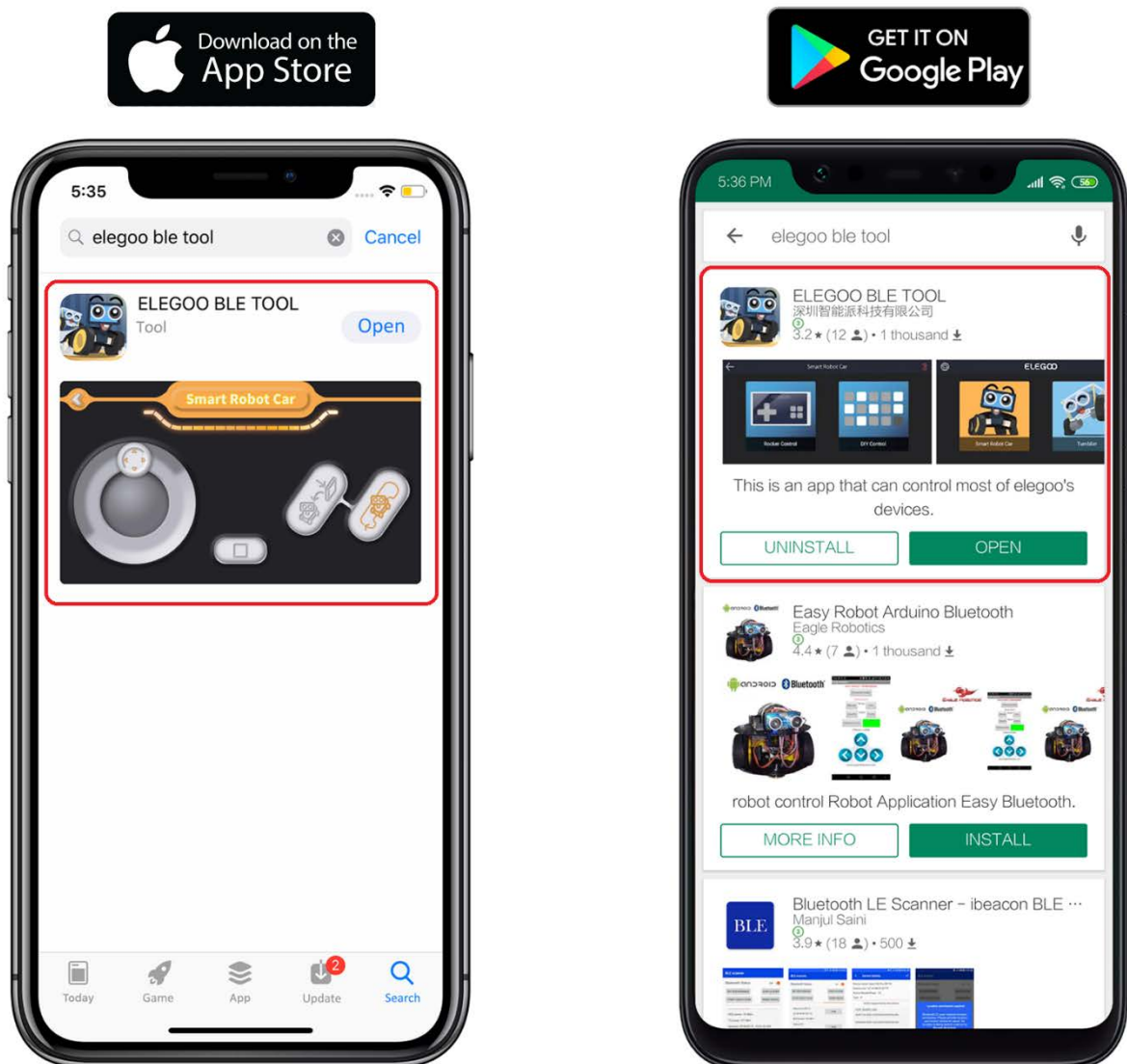


## II. Getting Started with the Bluetooth APP

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

### STEP1: Install the application.

You can download the latest version of the "ELEGOO BLE TOOL" app on the [App Store](#) and [Google Play](#).



## STEP2: Application Settings.

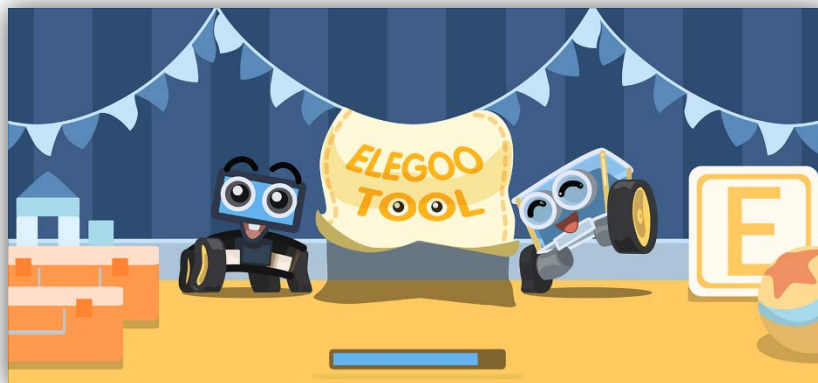
First of all, Turn on your cellphone's Bluetooth function.



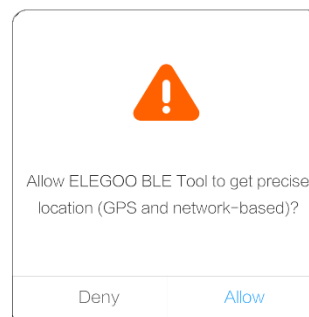
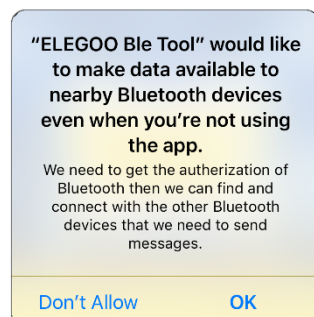
**Please note:**

**The Smart Cars CANNOT be paired in settings page, it can only be connected with "Elegoo BLE Tool" App.**

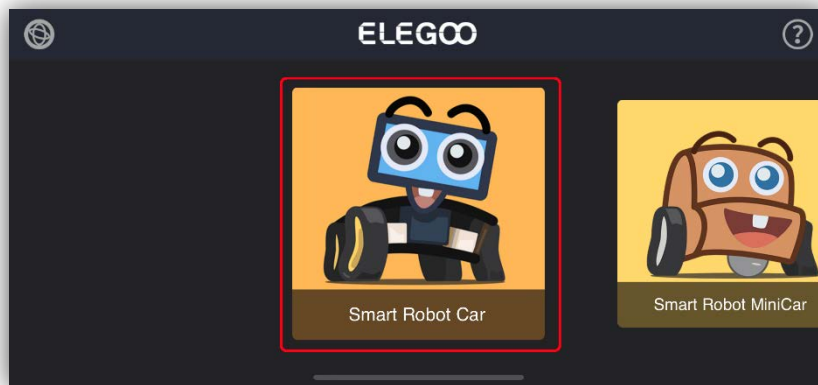
❖ Open the "Elegoo BLE Tool" App.



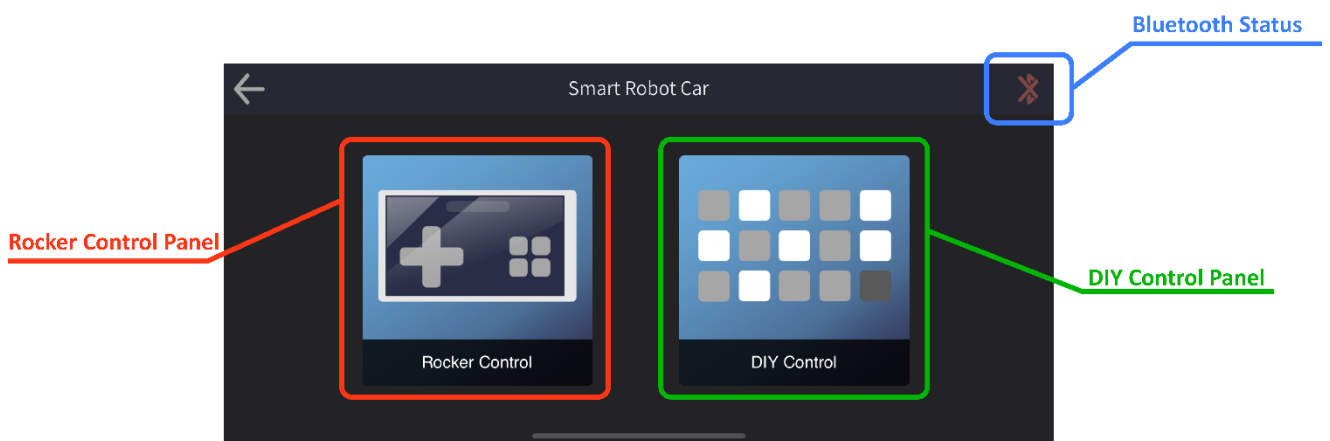
**The first time you open the app, the system will pop up a permission request. Please select the "Allow" option to ensure the app works.**




- ❖ Select the “Smart Robot Car”.



- ❖ The function of each icon in the Smart Robot Car homepage:

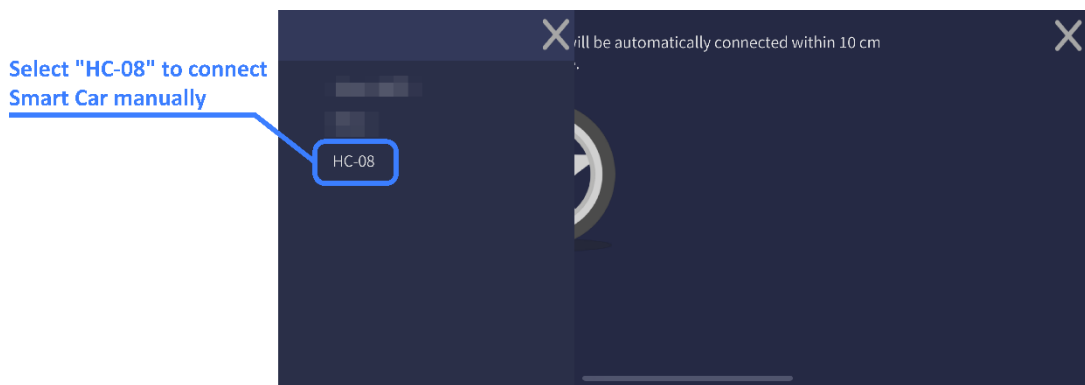


Tap the “” icon to enter the Bluetooth searching interface.

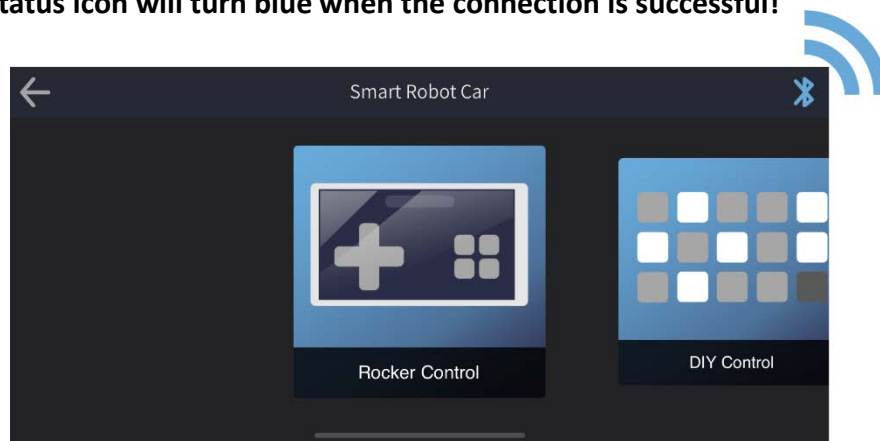
Put your phone close to the car(within 10cm), the app will recognize the Smart Car and automatically connect with it



You can also open the Bluetooth device list by tapping the menu icon “” in the upper left corner  
And select “HC-08” to connect the car manually.

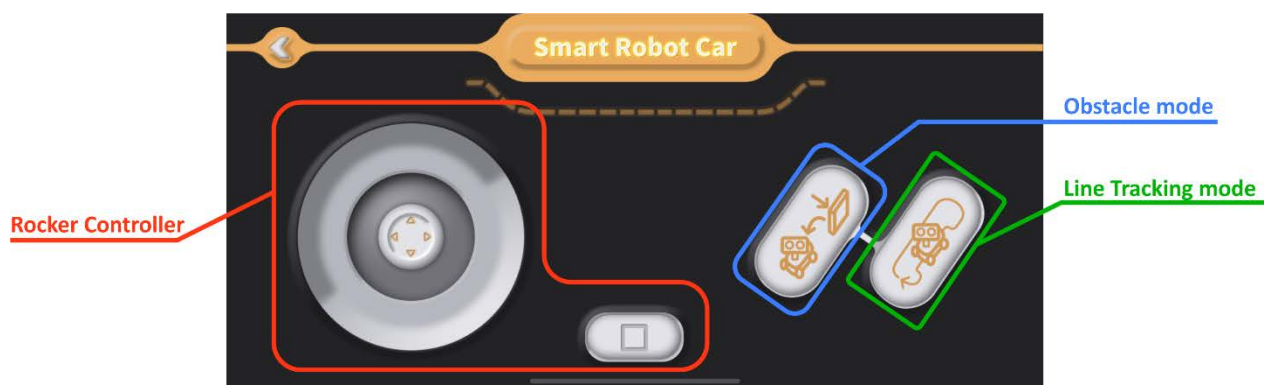


- ❖ The Bluetooth status icon will turn blue when the connection is successful!



Great, now the Smart Car is already connected to the App, we can control it in two different ways.

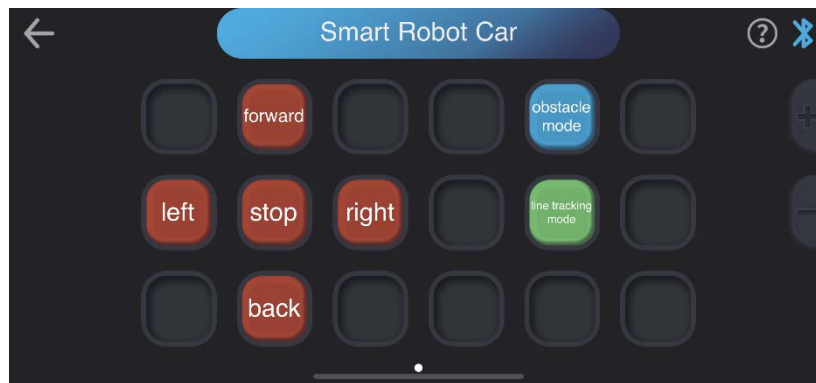
- ❖ The Rocker Control panel of the “Elegoo BLE Tool” App.



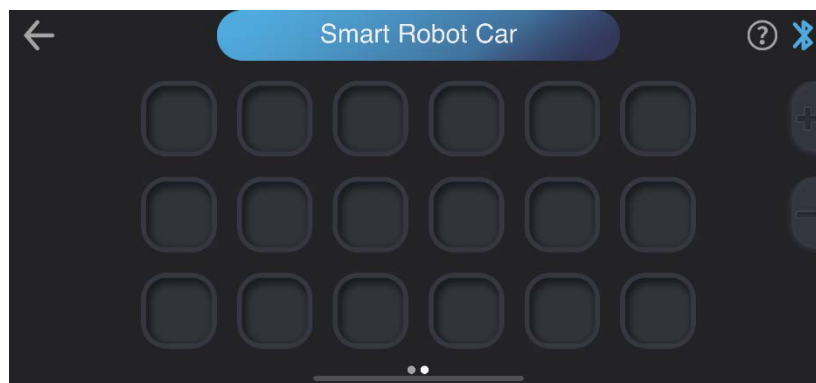
The main functions in the Rocker Control panel are divided into three parts:

- Rocker controller:** You can freely control the movement of the Smart Car, press the square button to stop the car
- Obstacle mode:** The car will turn into the obstacle avoidance mode, which is the same as the function in Lesson2.
- Line tracking mode:** The car will turn into the line tracking mode, which is the same as the function in Lesson 3

### ❖ The DIY control panel of the “Elegoo BLE Tool” App.

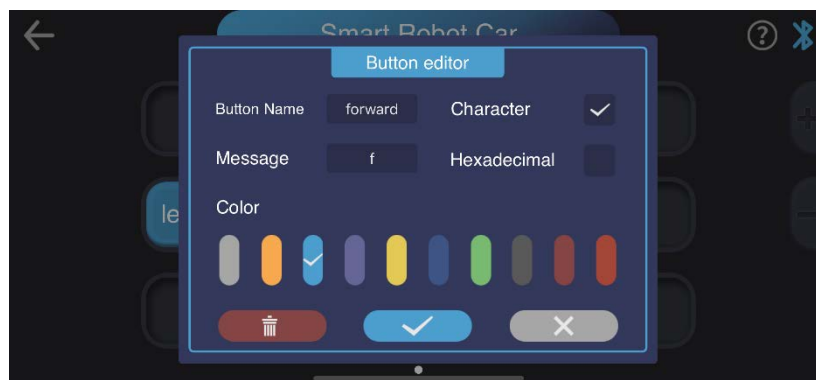


In the default settings, the DIY interface has only a few blank grids, we need to set their name, message and color to create buttons.



**Long press** the button you want to set, a “Button editor” option box will pop up as shown below. You need to fill in the “**Button Name**”, “**Message**” and select the **color** of the button in this page

(All preset Messages are of the character type, so you only need to check the "Character" option.).



### ❖ The comparison table of default Message and the Function is below.

function	FORWARD	BACK	LEFT	RIGHT	STOP	Line Tracking mode	Obstacles mode
Message	f	b	l	r	s	1	2

\*Case sensitive



### III. Smart Car core code

In this lesson, we used all the features of a smart car, so we need to put the code from all the previous lessons in one single sketch, which we call SmartCar\_Core.



You can open this Sketch in the path:

"\Elegoo Smart Robot Car Kit V3.0\Lesson 5 Bluetooth Cat\SmartCar\_Core"

A large part of the code has been explained in the previous Lessons, here we will explain the Bluetooth-related part of the entire Sketch.

#### Bluetooth Data capture code:

```
void getBTData() {
  if(Serial.available()) {
    switch(Serial.read()) {
      case 'f': func_mode = Bluetooth; mov_mode = FORWARD; break;
      case 'b': func_mode = Bluetooth; mov_mode = BACK; break;
      case 'l': func_mode = Bluetooth; mov_mode = LEFT; break;
      case 'r': func_mode = Bluetooth; mov_mode = RIGHT; break;
      case 's': func_mode = Bluetooth; mov_mode = STOP; break;
      case '1': func_mode = LineTeacking; break;
      case '2': func_mode = ObstaclesAvoidance; break;
      default: break;
    }
  }
}
```

This code block define a function "getBTData()". When the data is captured in UNO's serial port, this function will run and recognize the contents of this data and convert it into the control command of the next level function.

**For example:** When we push the rocker forward in the Rocker control panel of the APP, you phone will send a letter 'f' to the Smart Car through the Bluetooth. After receiving the letter, UNO wakes up the



“getBTData()” function and passes the “getBTData()” function. The letter 'f' is converted to the movement\_mode : “FORWARD”.

#### Bluetooth command code:

```
void bluetooth_mode() {
  if(func_mode == Bluetooth){
    switch(mov_mode){
      case FORWARD: forward(); break;
      case BACK:    back();    break;
      case LEFT:    left();    break;
      case RIGHT:   right();   break;
      case STOP:    stop();    break;
      default: break;
    }
  }
}
```

After the previous function, the Bluetooth signal has been converted to a direct control command, which will be received by “bluetooth\_mode()”, it determine which motion functions will be executed.

**For example:** In the previous example, The letter 'f' is converted to “FORWARD”, the bluetooth\_mode() will turn this command to the motion control function “forward()”.

#### Motion control code:

```
void forward(bool debug = false){
  analogWrite(ENA, carSpeed);
  analogWrite(ENB, carSpeed);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  if(debug) Serial.println("Go forward!");
}

//ect...

void stop(bool debug = false){
  digitalWrite(ENA, LOW);
  digitalWrite(ENB, LOW);
  if(debug) Serial.println("Stop!");
}
```

---

At last, those are Smart Car motion control functions that control the UNO's output, which will drive the motor to work.