

## *Lesson 5 Bluetooth Car*

---

### *Points of this section*

前のレッスンでロボット各部分の使用方法を紹介しました。

このレッスンで各部分を一つのアプリで制御することや DIY 方法を学びます。

### *Learning Parts:*

- ◆ Bluetooth モジュールと Bluetooth APP の使用方法を学ぶ
- ◆ Bluetooth 経由でカーを制御する方法を学ぶ
- ◆ この関数を実装するプログラムを書く

### *Preparations:*

- ◆ Smart Car (バッテリー搭載)
- ◆ USB ケーブル
- ◆ Bluetooth モジュール
- ◆ スマホ (IOS または Android)

## I . Bluetooth module

Bluetooth は、工業、科学、および医療に応用するワイヤレステクノロジー標準で、無線帯域（2.400～2.485 GHz）の短波 UHF 電波を使用して固定デバイス間で近距離でデータを交換し、パーソナルエリアネットワーク (PAN) を構築することができます。ZigBee や Wi-Fi などの RF プロトコルもそうです。



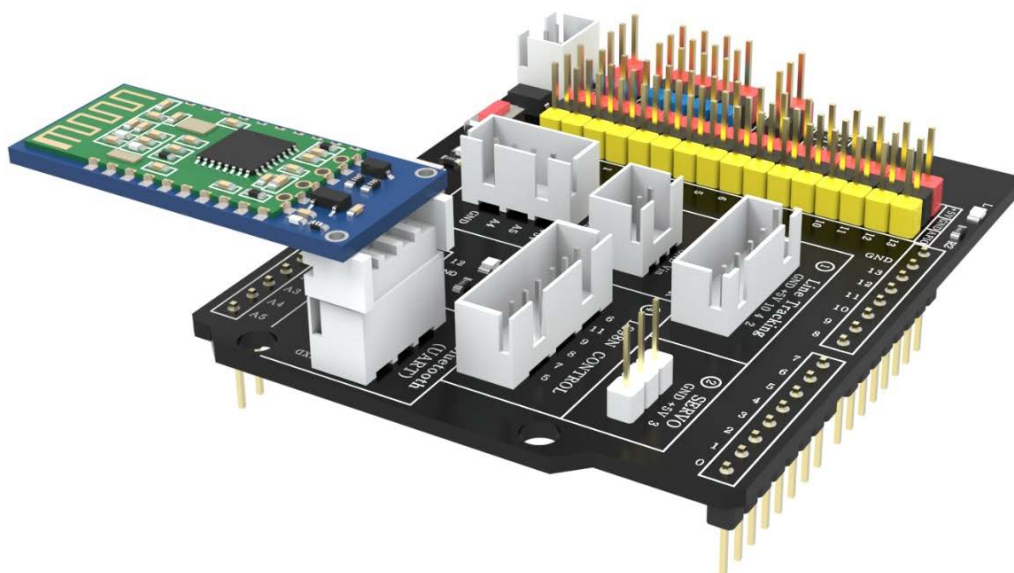
スマートカーキットでは、Bluetooth モジュール「HC-08」を利用して、Bluetooth 経由で他のデバイスにシリアルデータを送信します。



HC-08 はシールドの **RX/TX** ピンを通して UNO と通信します。

### 注意:

スケッチを書き込み中、RX/TX ピンが使用されますので、書き込み完了するまで Bluetooth モジュールを取り付けしないでください。

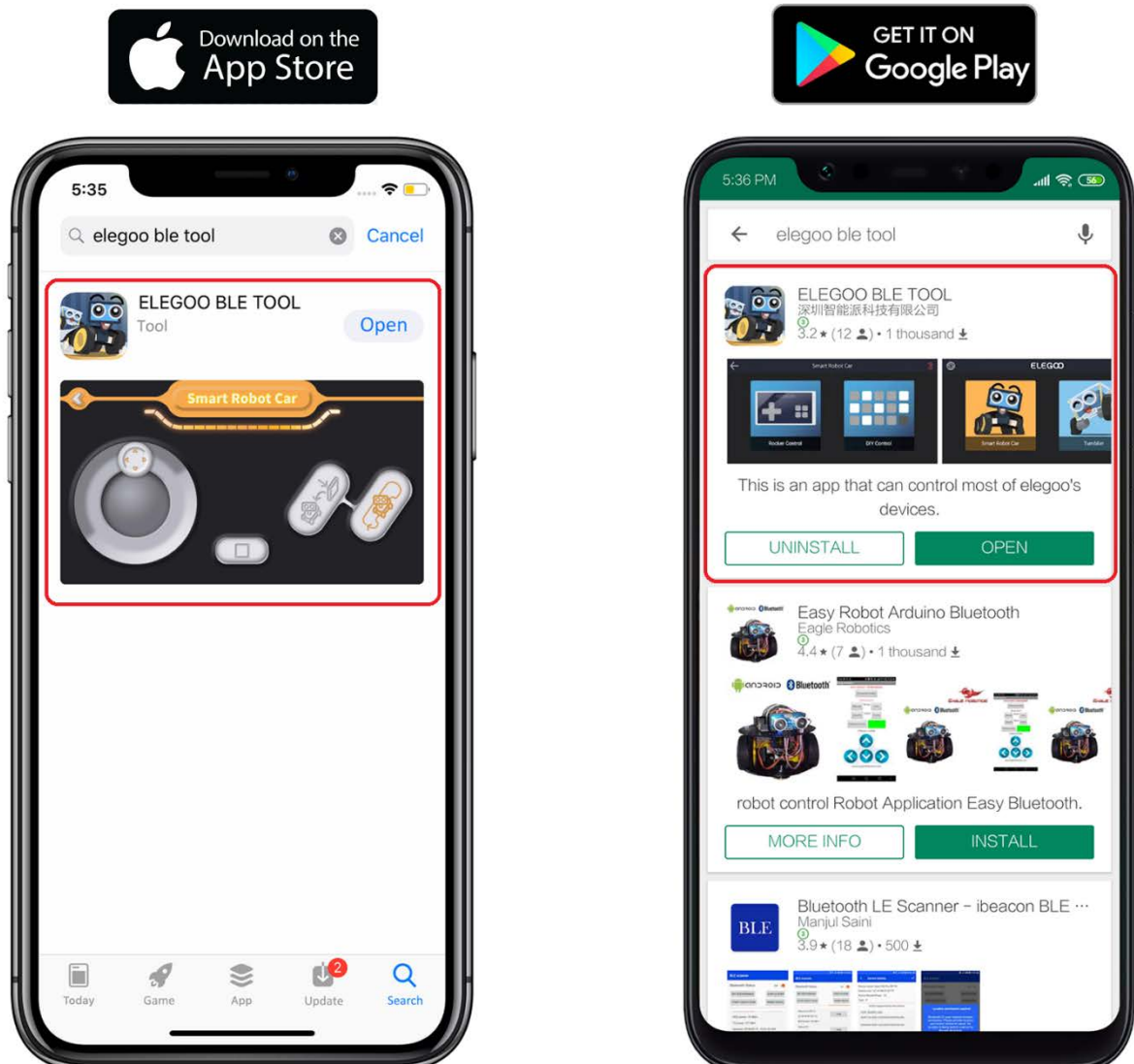


## II . Bluetooth APP 入門

始める前に、HC-08 Bluetooth モジュールを拡張ボードに接続し、電源をオンにしてください。

### STEP1: Install the application.

[App Store](#) や [Google Play](#) で最新バージョンのアプリ「**ELEGOO BLE TOOL**」をダウンロードしてください。



## STEP2: Application Settings.

まず、Bluetooth 機能をオンにしてください。



**注意:**

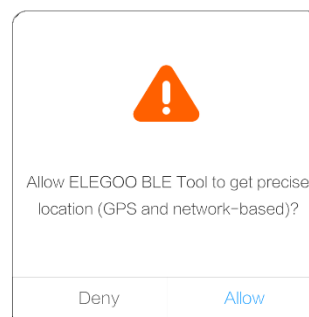
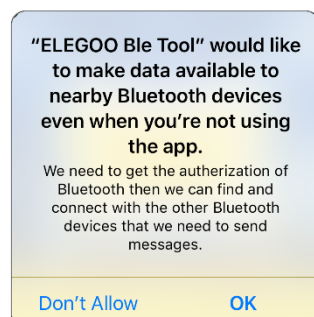
設定ページにロボットカーが見つかりません。

ロボットカーが「Elegoo BLE Tool」アプリだけで接続できます。

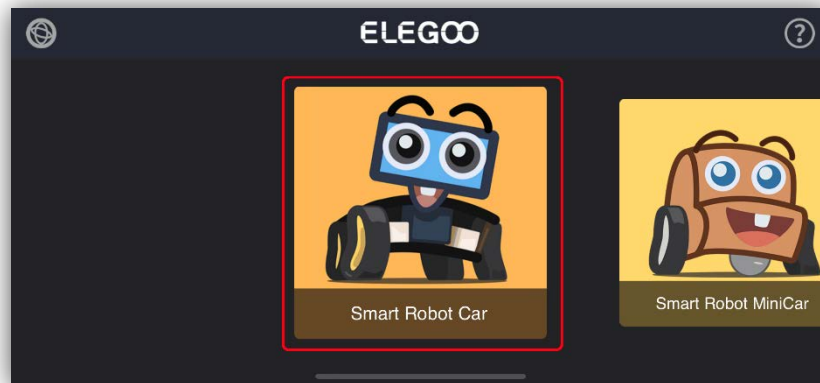
❖ 「Elegoo BLE Tool」アプリを開いてください。



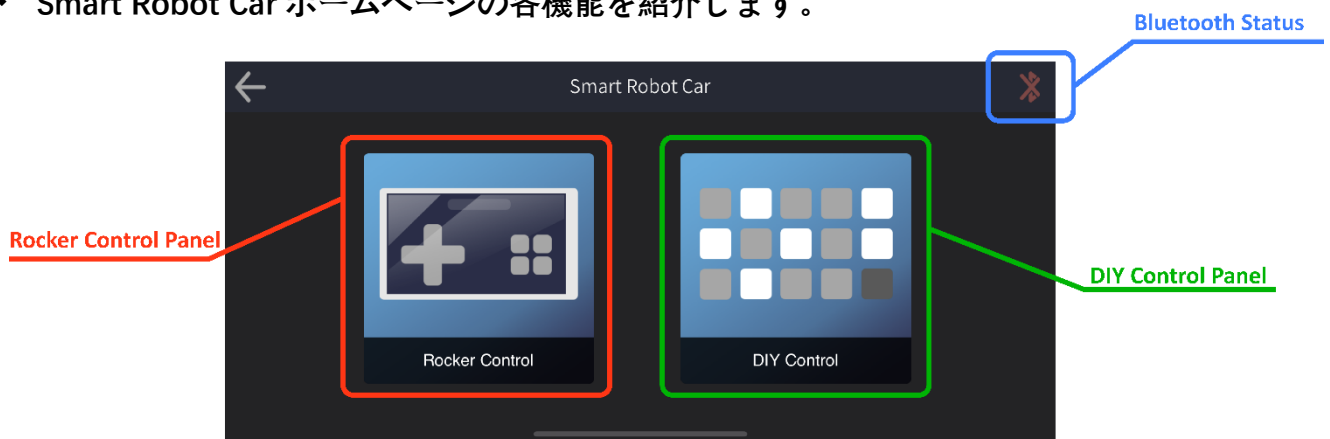
The first time you open the app, the system will pop up a permission request. Please select the "Allow" option to ensure the app works.



- ❖ 「Smart Robot Car」を選びます

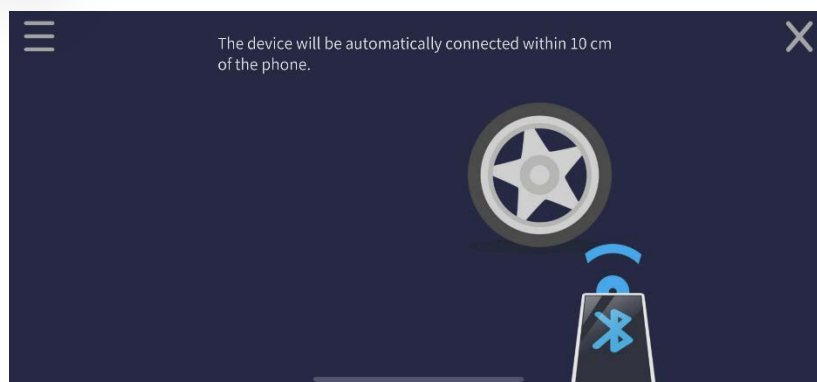


- ❖ Smart Robot Car ホームページの各機能を紹介します。



アイコンをクリックして検索インターフェースに入ります。

スマホをカーの近く（10cm 以内）に置くと、アプリが自動的にスマートカーに接続します



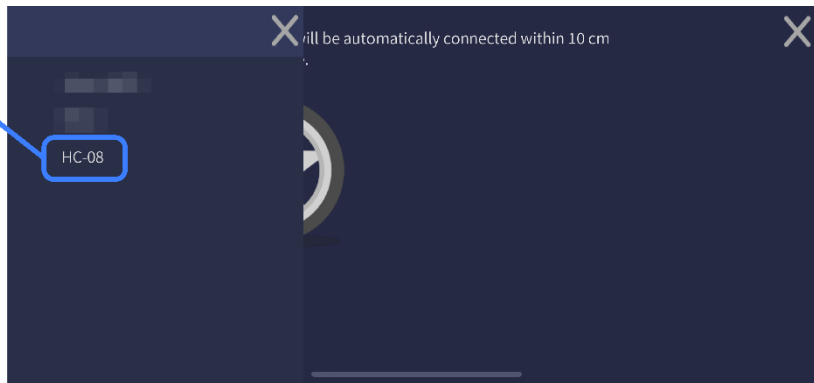
また、左上隅にあるメニューアイコン”



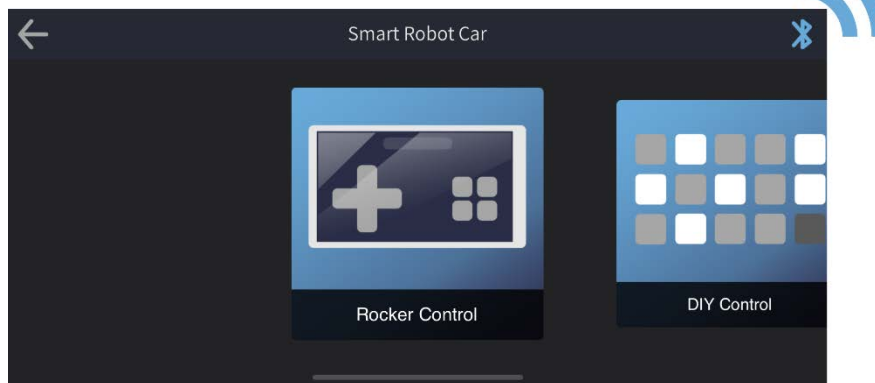
”をクリックしてデバイスリストを開くこともできます。

「HC-08」を選択してスマートカーに接続します。

Select "HC-08" to connect Smart Car manually

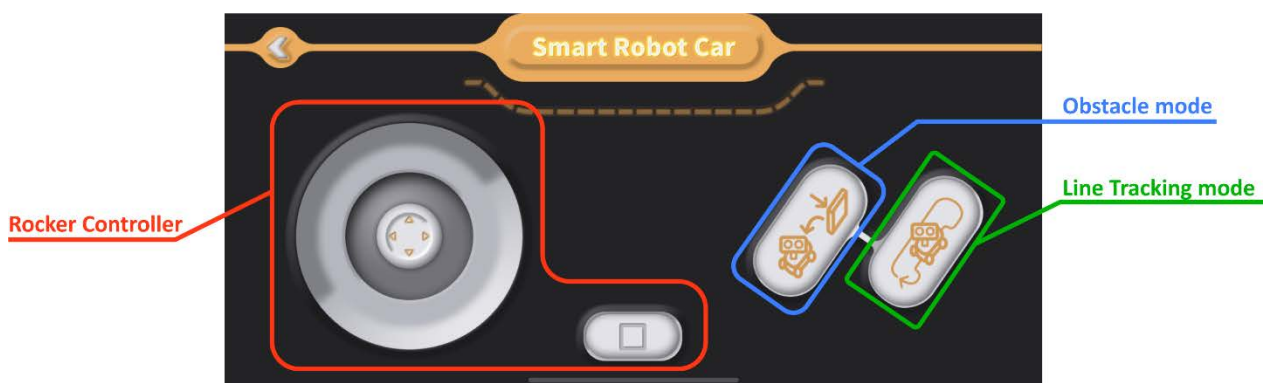


❖ 成功に接続すると、Bluetooth アイコンが青になります。



スマートカーもう既にアプリに接続しました。制御する方法は二つあります。

❖ 「Elegoo BLE Tool」アプリのジョイスティックでコントロールインターフェースです



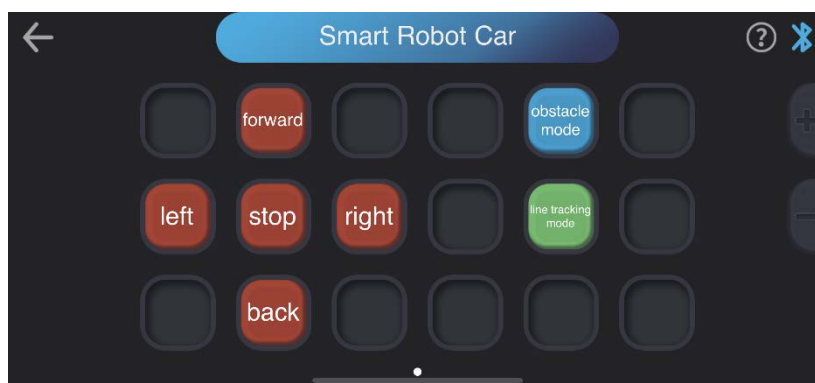
ジョイスティックでコントロールインターフェースは三つの機能があります：

**Rocker controller:** ロボットカーを制御することができ、ボタンを押してカーを停止します。

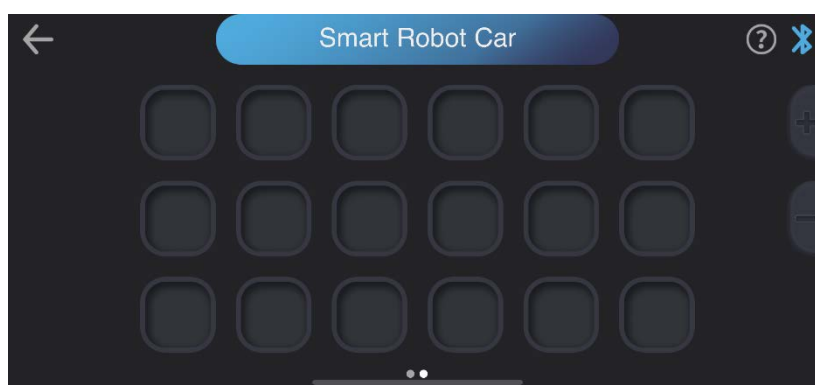
**Obstacle mode:** 障害回避モードに入り、Lesson2 をご参考してください。

**Line tracking mode:** ライントラッキングモードに入り、Lesson3 をご参考してください。

- ❖ 「Elegoo BLE Tool」アプリのカスタムページです。

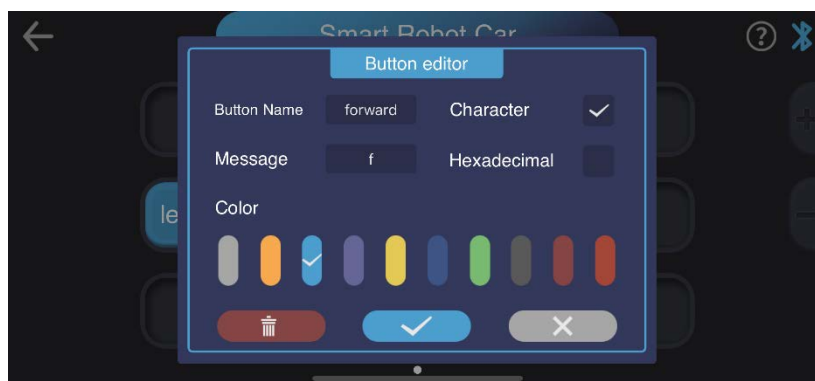


写真のような空白ブランクに名前、メッセージや色のボタンを設定する必要があります。



設定したいボタンを**長押し**、ボタンエディタでボタンの色や名前を入力してください。

(すべてのプリセットメッセージは Character Type なので、"Character"オプションをチェックするだけでOKです。)



- ❖ 各機能に対応するメッセージは下記のようにご参考ください:

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

このレッスンでスマートカー全ての機能を利用しますので、この前学んだコードを「SmartCar\_Core」というスケッチに入れてください。



下記のルートにスケッチがあります。

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

スケッチの Bluetooth について説明します。

**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;
    }
  }
}
```

このコードは、関数「getBTData ()」を定義します。データが UNO のシリアルポートに取り込まれると、この関数は実行されてデータの内容を認識し、それを次の制御コマンドに変更します。

**例:** ジョイスティックを前に推すと、Bluetooth でスマートカーに文字 'f'を送信します。そして UNO は「getBTData ()」関数を起こして送ります。文字 'f' は movement\_mode: “ FORWARD”に変換されます。



## 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;
    }
  }
}
```

この後、Bluetooth 信号は「`bluetooth_mode ()`」によって受け取られて直接制御コマンドに変換されます。

例：前の例では、文字 'f'は“ FORWARD”に変換され、`bluetooth_mode ()` はこの命令を動作コントロール機能“ `forward ()` ”に変えました。

## 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!");
}
```

最後、これらは UNO の出力を制御するスマートカーの動作コントロール機能で、モーターを動かします。