

Leçon 2 - Bluetooth (Bluetooth Car)



Points clés de la leçon

Il est très intéressant de contrôler un robot sans fils et tout particulièrement en utilisant la technologie Bluetooth. Cette leçon vous apprend à maîtriser cette technologie pour contrôler votre robot.

Sommaire:

- Utiliser le module "HC-06 Bluetooth" et l'application "Bluetooth SPP"

- Contrôler le robot par Bluetooth

- Ecrire le programme contenant les fonctions de mouvement du robot

Préparatifs:

- Le robot

- Un câble USB (fourni)

- Une tablette ou un téléphone sous Android (non fournis)

I . Module Bluetooth

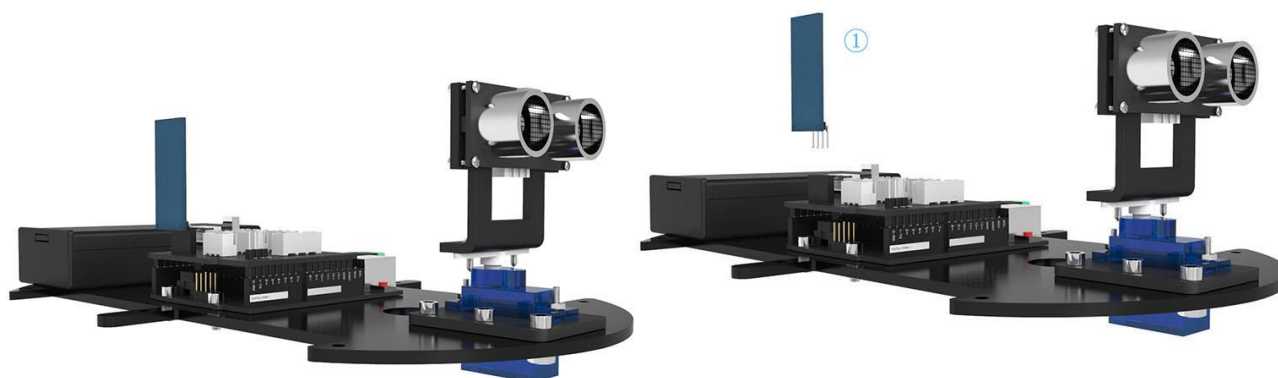


Description du module HC-06:

1. Utilise la carte et le protocole standards Bluetooth
2. Alimentation 3.3V
3. Compatible Baud rate 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
4. Dimensions : 28mm x 15 mm x 2.35mm
5. Courant : 40mA
6. Courant en veille : <1mA
7. Utilisable pour les fonctions : GPS, système de lecture de gaz...
8. Peut-être connecté à un ordinateur, PDA ...

Connection sur le robot



|



① bluetooth module

II. Application Bluetooth

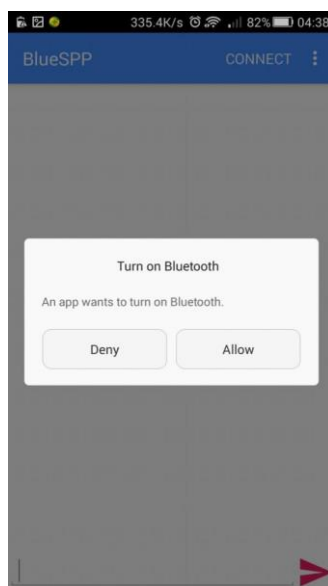
Nous vous offrons l'application Android Blue SPP qui permet une communication entre un téléphone Android et le module Bluetooth:

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

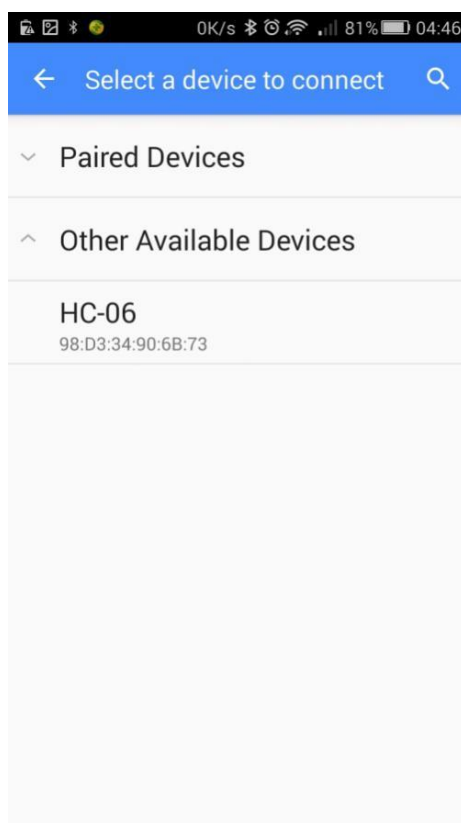
Installez l'APK sur votre appareil:



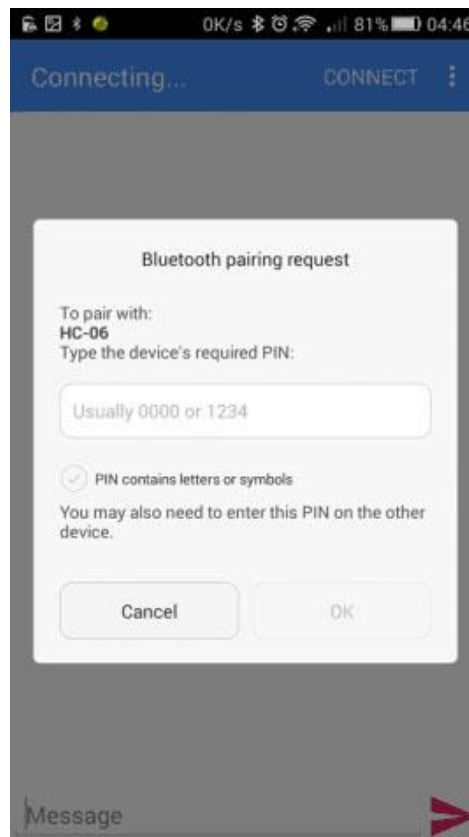
Vous devrez activer le Bluetooth sur votre appareil pour lancer l'application correctement.



Vous pouvez maintenant chercher les appareils Bluetooth à proximité:

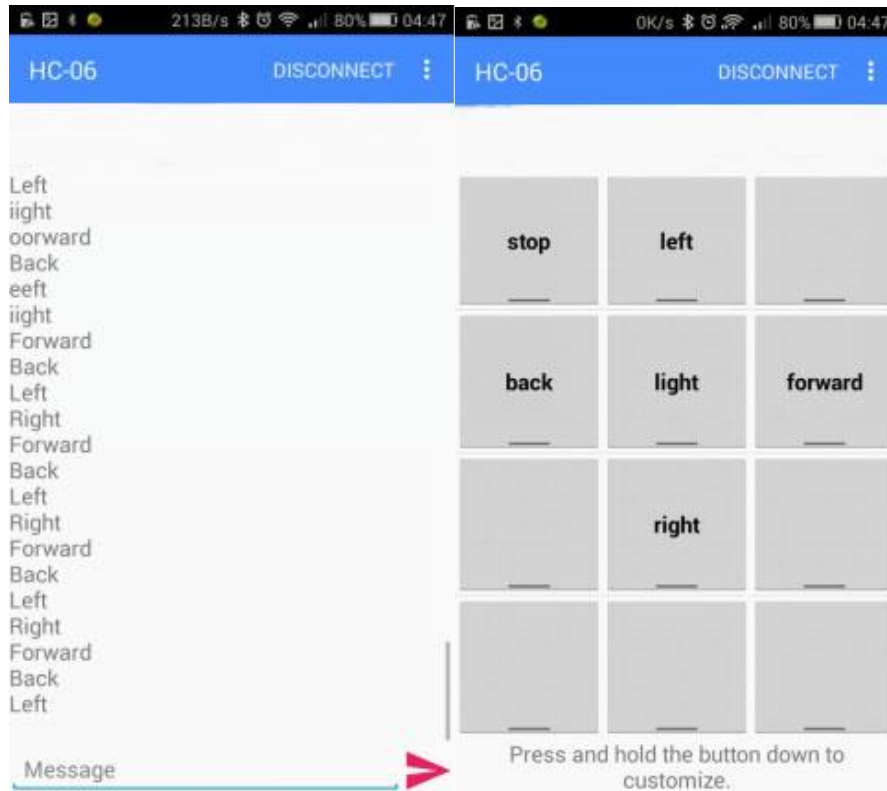


Cliquez sur HC-06, un mot de passe est demandé : saisissez 0000 ou 1234.

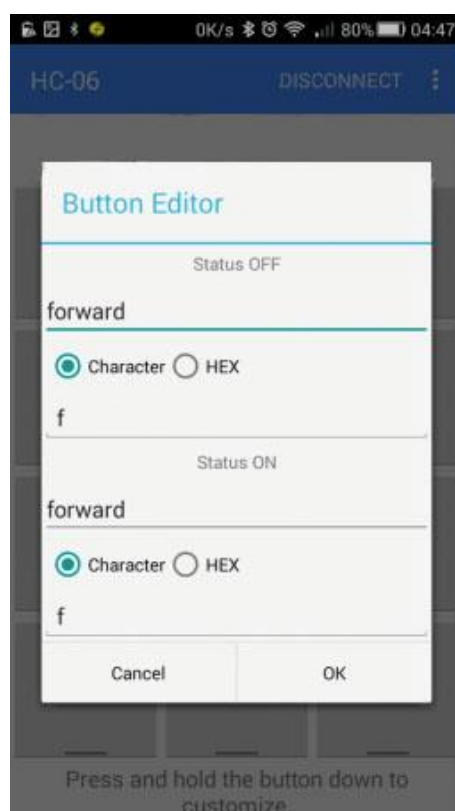


L'interface de l'application est soit sous forme de chat, soit sous forme de clavier virtuel selon votre envie.

Vous pouvez donc envoyer une commande en la tapant ou en paramétrant les boutons du clavier:



Ecran permettant d'affecter une commande à une touche du clavier:



III. Tester le fonctionnement Bluetooth

Testons le programme suivant qui permet d'allumer/ éteindre une LED connectée sur la PIN 13 de la carte Elegoo UNO

```
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();
  }
}
```

Ouvrez le fichier "bluetooth_blink\bluetooth_blink.ino"

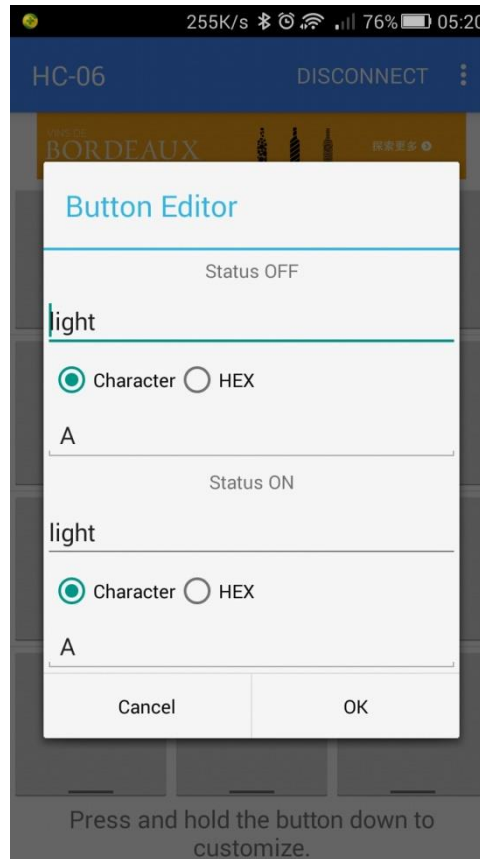
U深度U盘 (H:) > 工作项目 > 4WD car tutorial and code > code > 2 , Bluetooth Car			
名称	修改日期	类型	
Bluetooth_Blink	2016/1/24 23:32	文件夹	

Déversez le programme sur la carte Elegoo UNO

ATTENTION : lorsque le module BLUETOOTH est branché, il empêche le déversement de code sur la carte Elegoo UNO, il est donc impératif de le retirer temporairement.

Ouvrez l'application

Après connexion, nous paramétrons une touche du clavier de la manière suivante:



La pression de la touche "A" du clavier virtuel allume ou éteint la LED.

Explication du code:

Serial.begin(9600);

Cette instruction initialise une connexion série.

getstr=Serial.read();//The Bluetooth serial port to receive the data in the function

if(getstr=='A'){

stateChange();

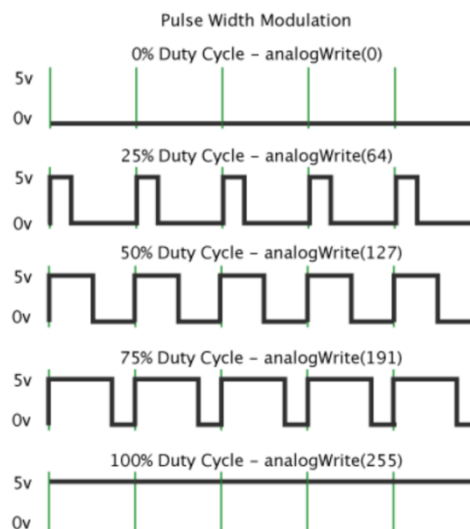
Cette suite d'instructions lit le port série dans l'attente d'un message. La chaîne de caractères reçue est ensuite comparée à la lettre "A". Si le résultat est "est égal", la fonction stateChange() est appelée.

IV. Robot contrôlé par Bluetooth

PWM :

Lorsque le robot tourne à gauche ou à droite, il n'est pas nécessaire d'avoir une vitesse élevée. Plus la vitesse est élevée et moins le contrôle est précis. Mais comment pouvons-nous contrôler la vitesse des moteurs?

La réponse est PWM. De l'anglais "Pulse Width Modulation" ou Modulation de Largeur d'Impulsion en français (MLI). C'est une technologie efficace pour contrôler des circuits analogiques via une sortie numérique. Elle permet de définir dans une plage de temps donnée, la durée pendant laquelle la sortie va effectivement émettre du courant et ne pas en émettre. Plus la durée d'émission de courant est grande, plus la quantité d'énergie transmise sera importante. En d'autres termes, cela permet de contrôler la quantité d'énergie reçue par le système que l'on veut piloter et donc dans le cas de nos moteurs, cela permet d'en faire varier la vitesse de fonctionnement.



Les lignes vertes permettent de matérialiser une période de temps (de l'une à la suivante).

A la valeur transmise par "analogWrite(value)" correspond un pourcentage. Ce pourcentage est appelé "Duty Cycle" ou temps durant lequel l'état de la sortie est "Haut".

Sur la première figure la valeur est 0. Dans ce cas, les moteurs restent immobiles.

Sur la dernière figure, la valeur est 255. Dans ce cas, les moteurs tournent à la vitesse maximum.

Les trois figures centrales représentent 25% de la puissance, 50% et enfin 75%.

Les sorties PWM sont aussi utilisées pour ajuster l'intensité d'une LED ou d'une ampoule par exemple.

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

analogWrite() est utilisé pour définir une valeur analogique sur une sortie compatible PWM (valeur de 0 à 255).

Les ports compatibles PWM sont les ports 3, 5, 6, 9, 10, 11.

La vitesse de notre robot est contrôlée ainsi : pin5 et pin10 sur ENA et ENB.

ABS est la valeur analogique définie comme suit : int ABS=135;

Vous pouvez personnaliser cette valeur pour augmenter, baisser la vitesse de votre robot.

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

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

Maintenant que nous avons vu les connaissances de base, nous allons déverser le code suivant sur la carte Elegoo UNO.

Ouvrez le fichier "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!");
```

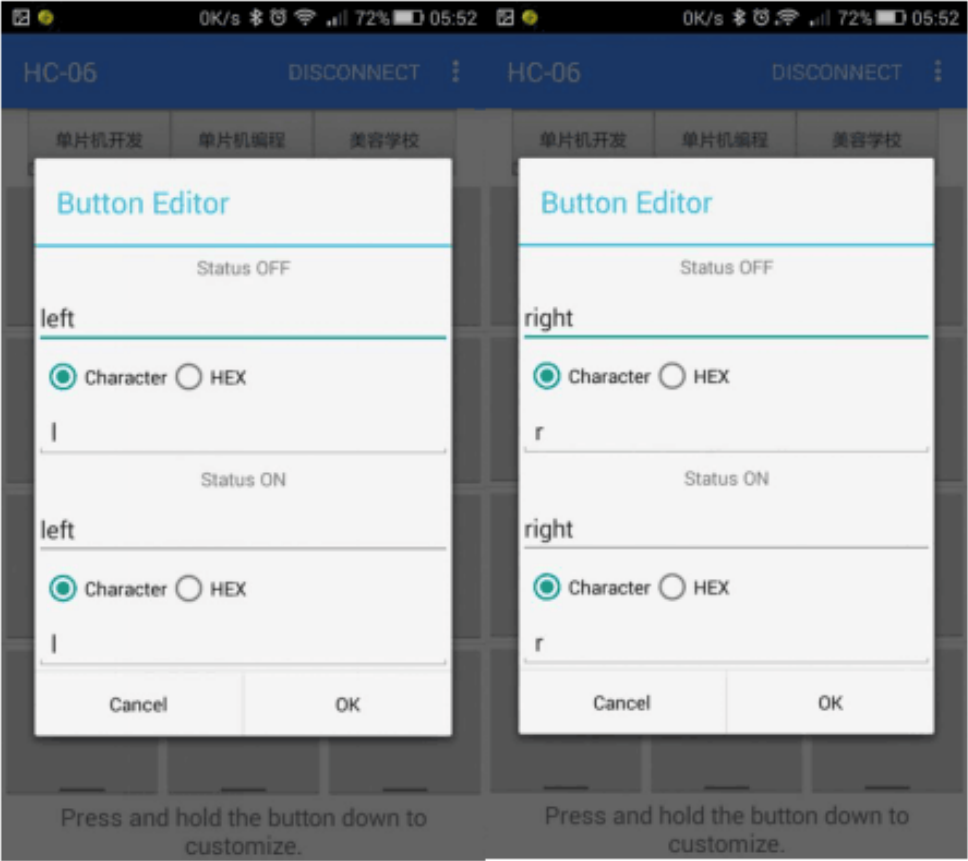
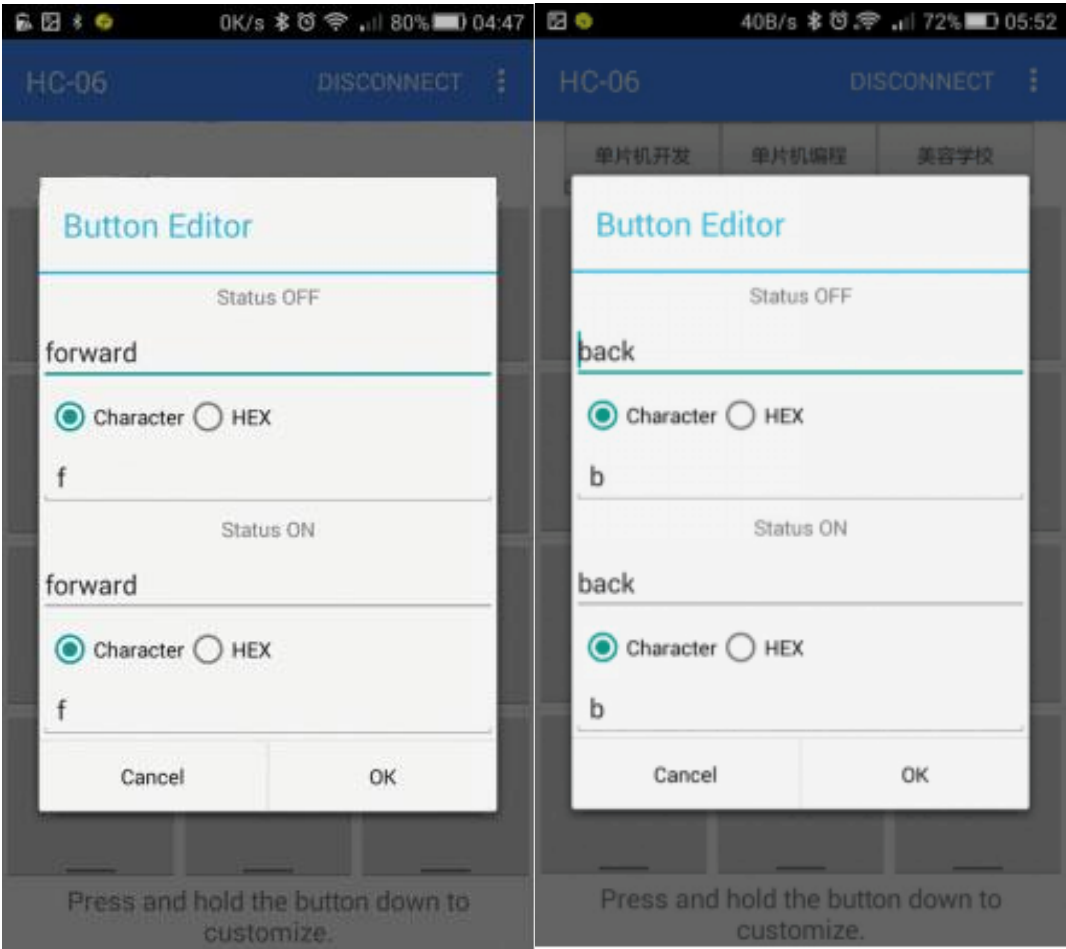
```
}  
void _mStop()  
{  
    digitalWrite(ENA,LOW);  
    digitalWrite(ENB,LOW);  
    Serial.println("Stop!");  
}  
void stateChange()  
{  
    state = !state;  
    digitalWrite(LED, state);  
}  
void setup()  
{  
    pinMode(LED, OUTPUT);  
    Serial.begin(9600);  
    pinMode(in1,OUTPUT);  
    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);  
    }  
}
```

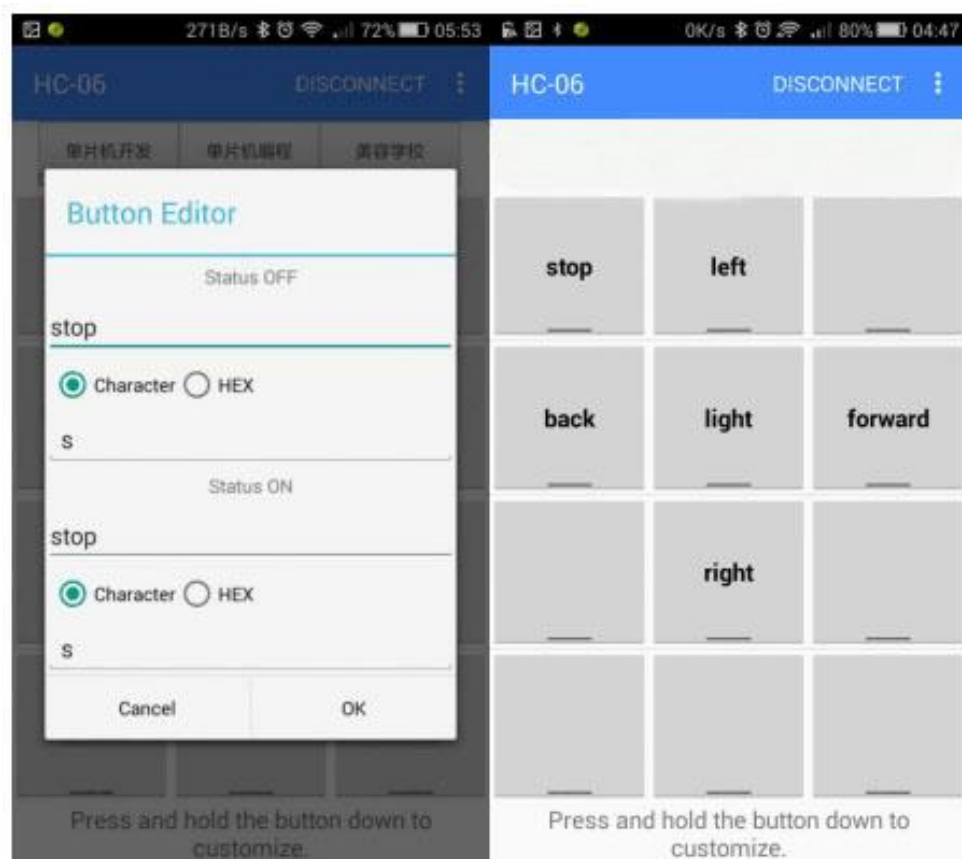
```
else if(getstr=='l')
{
    _mleft();
    delay(200);
}
else if(getstr=='r')
{
    _mright();
    delay(200);
}
else if(getstr=='s')
{
    _mStop();
}
else if(getstr=='A')
{
    stateChange();
}
}
```

Déversez le code sur la carte Elegoo UNO (n'oubliez pas de retirer le module Bluetooth durant cette opération), puis débranchez le câble USB.

Posez le robot au sol et mettez-le en marche.

Ouvrez l'application Android et définissez les paramètres suivants:





Voilà, il est possible maintenant de contrôler le robot via la technologie Bluetooth.