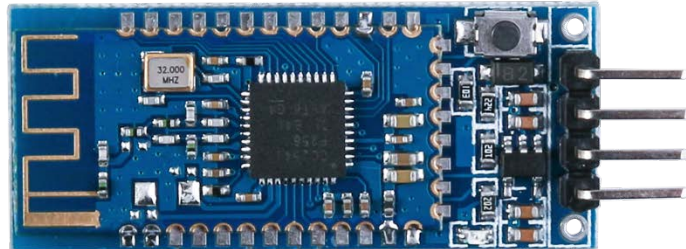


## Lesson 2 Bluetooth Car



### このセクションのポイント

Arduino を学ぶとき、車をワイヤレスかつ特定の空間で制御することは非常に重要です。したがって、レッスンでは、Bluetooth で車を制御する方法を学習します。

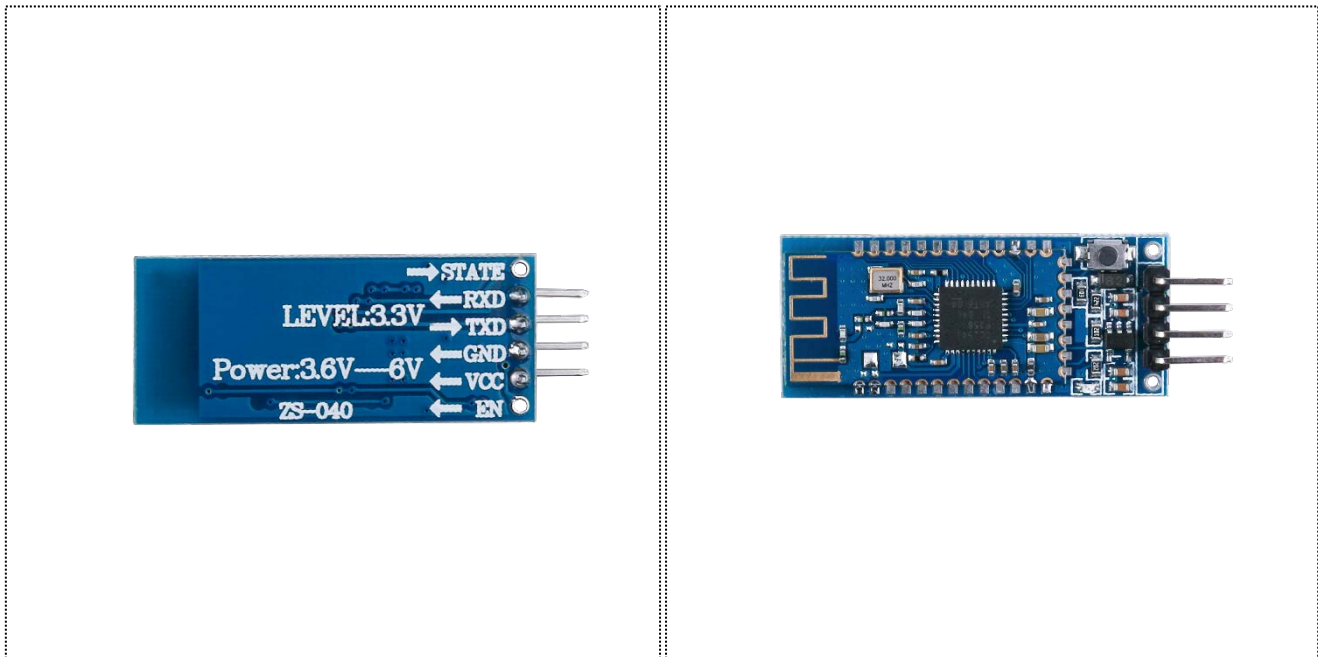
学習パーツ:

- ◆ CC41-A Bluetooth モジュールと Bluetooth APP の使用方法を学ぶ
- ◆ Bluetooth 経由で車両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

準備:

- ◆ 車両（バッテリー搭載）
- ◆ USB ケーブル
- ◆ CC41-A Bluetooth モジュール
- ◆ iPhone またはタブレット

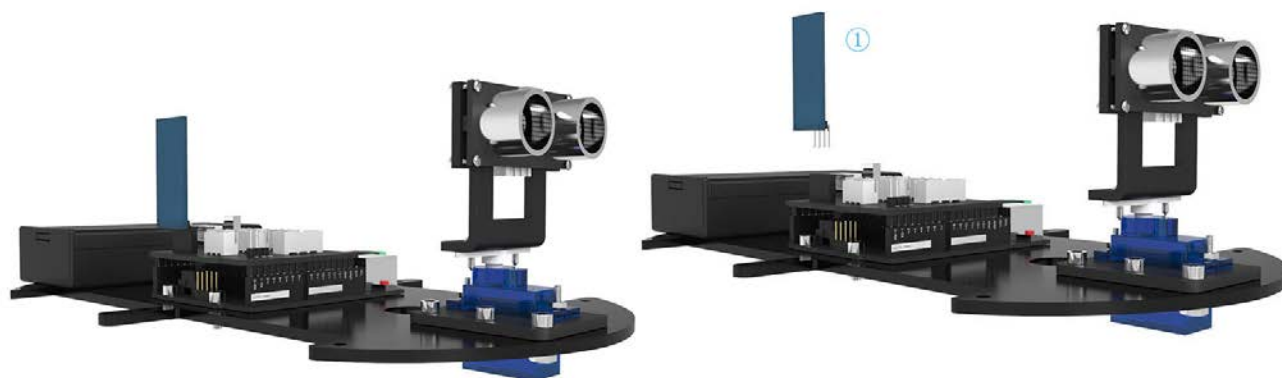
## I. Bluetooth モジュール



### CC41-A モジュールの説明:

1. CSR の主流 Bluetooth チップ、BluetoothV2.0 のプロトコル標準を採用  
シリアルポートのアナログ動作電圧は 3.3V です。
  3. ユーザーはボーレート 1200、2400、4800、9600、19200、38400、57600、115200 を設定できます  
キーモジュールの寸法は、28mm × 15mm × 2.35mm
  5. 動作電流：40mA
  6. 休眠電流：1mA 未満
  7. GPS ナビゲーションシステム、水力ガスリーディングシステム、産業フィールドマイニング制御システムに使用されている
  8. Bluetooth ノートパソコン、Bluetooth アダプタ、PDA などのコンピュータに接続することができます。
- これは、**UNO** コントローラボードに接続された **Bluetooth** モジュールの回路図です:

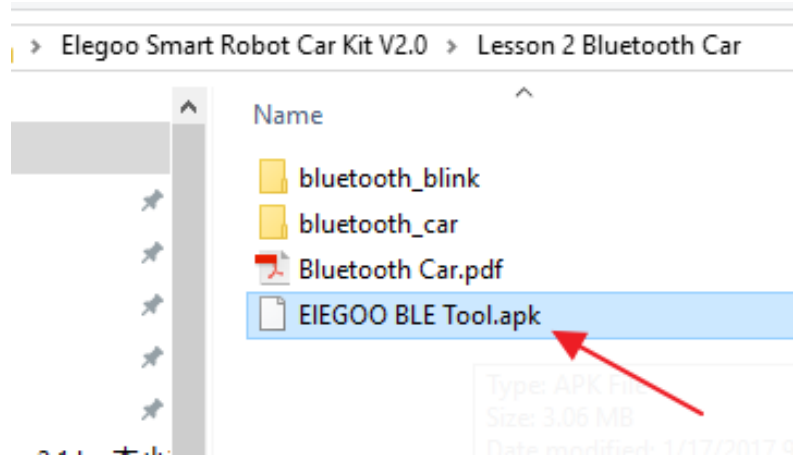
In the experiment we will connect it to UNO board via expansion board V5.



① bluetooth module

## II. Bluetooth APP

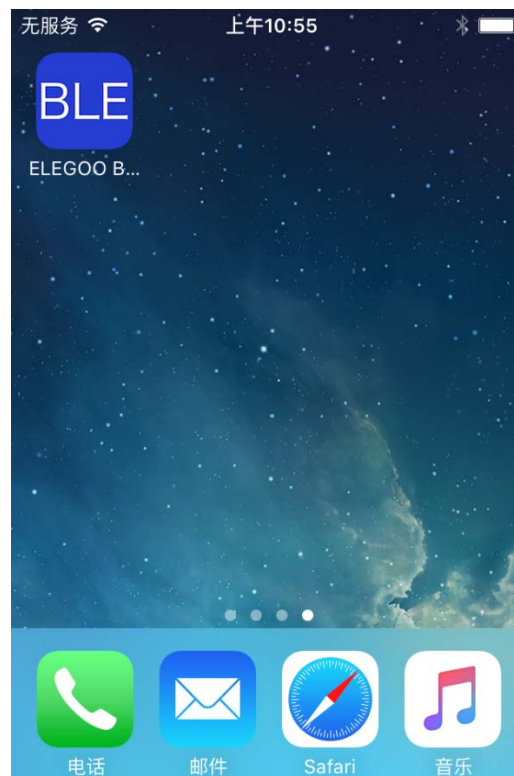
Android システムの場合は、APK ファイルを Android 製品にコピーしてインストールします。



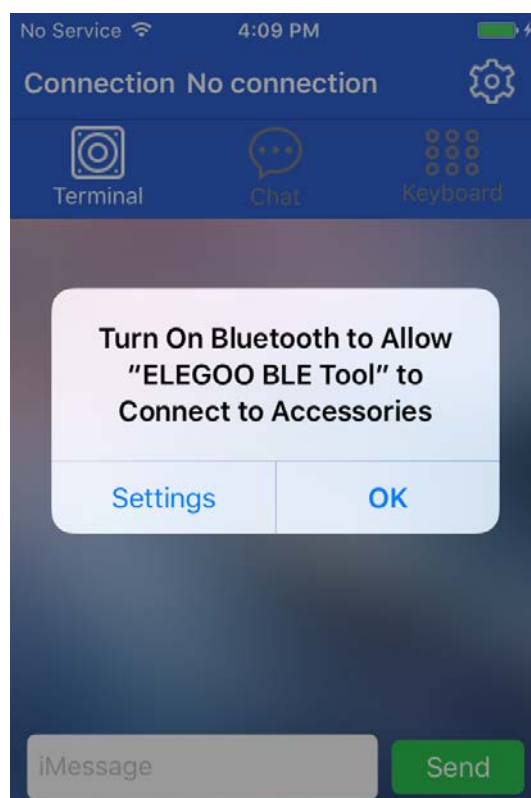
以下では、Apple の携帯電話やタブレットを例にとります：

Apple Store で「ELEGOO BLE ツール」を検索してください。

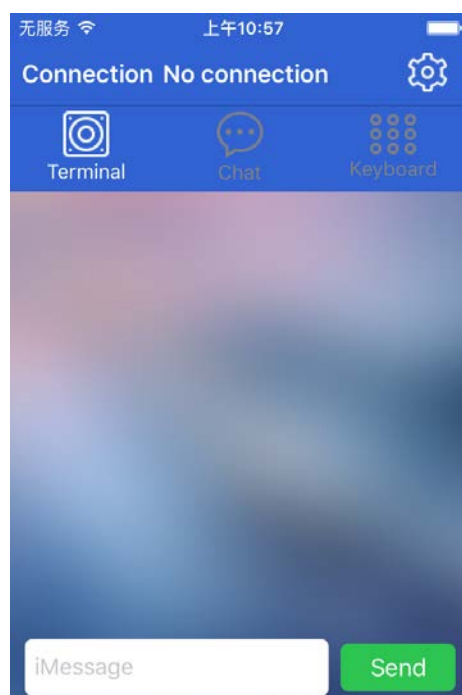
APP をインストールし、携帯電話で実行します。



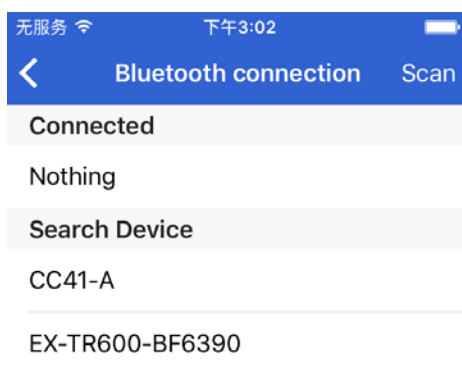
以下のように表示され、実行時に Bluetooth 機能をオンにするように指示されます。



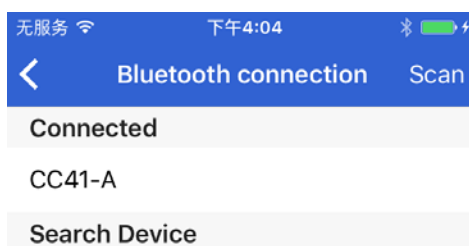
Connection を押してください



携帯電話が近くの Bluetooth 機器を検索します。

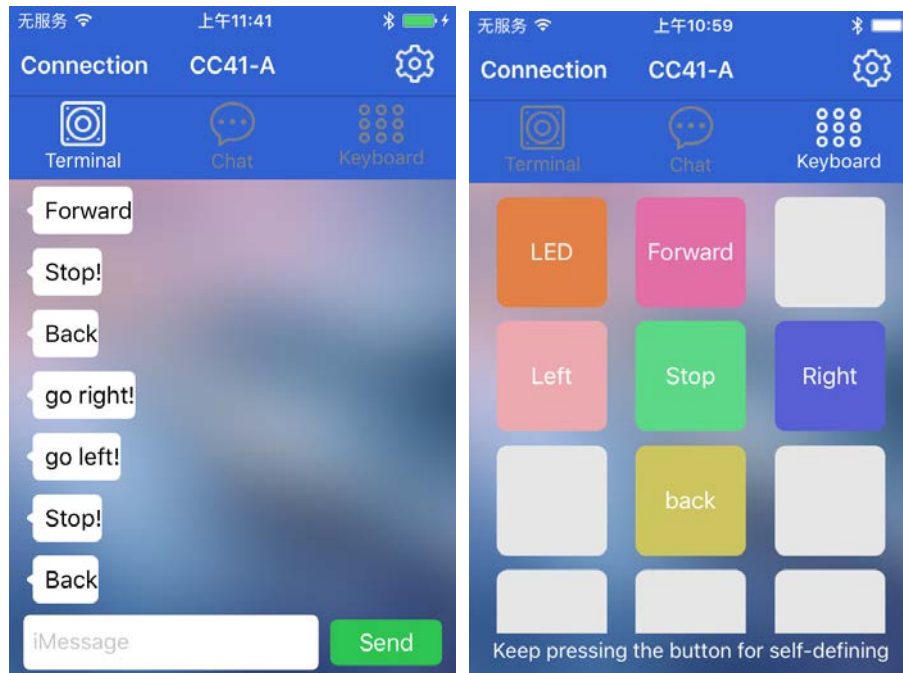


CC41-A をクリックし接続が成功すると、画面にデバイス接続が表示されます

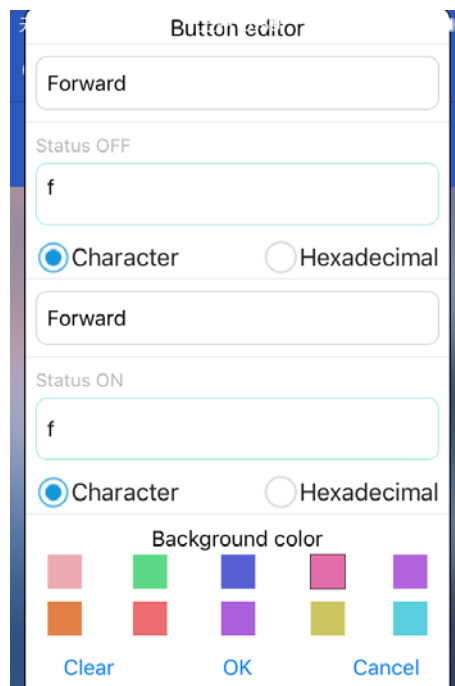


Device Connected

アプリのインターフェースは左のように表示されます。次に、指で画面を左にスライドさせると、右のようなキーパターンが得られます：



最後に、各ボタンの定義を設定します。たとえば、「go forward」とします。以下を参照してください。残りのキー値も同様に設定されます。

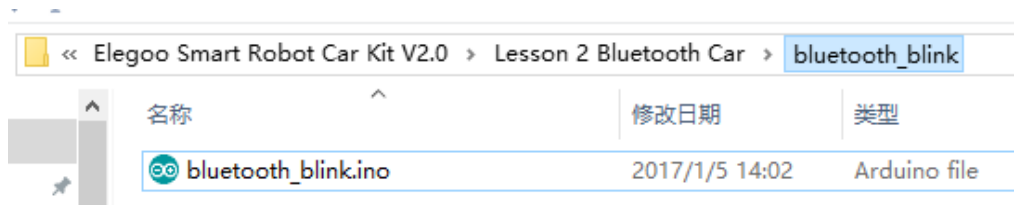


### III. Testing

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();
    }
}
```

ファイルを開きます : ¥ bluetooth\_blink ¥ bluetooth\_blink.ino

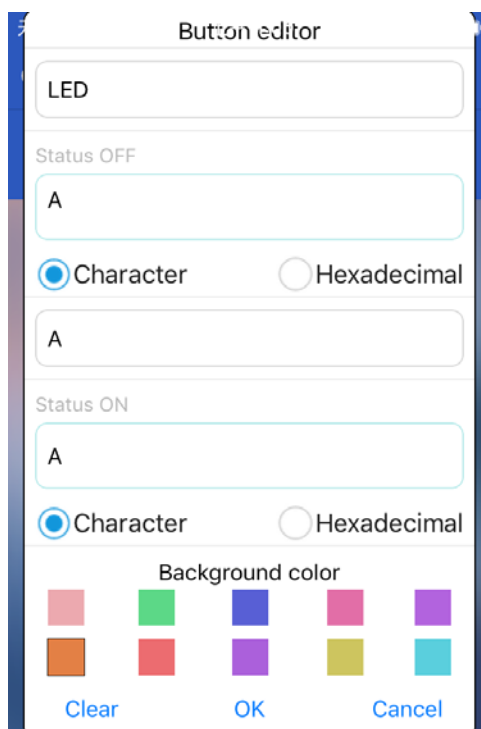


プログラムを UNO ボードにアップロードし、コンピュータから切り離してから、車の電源を入れます。（ヒント：あなたがプログラムをアップロードするときに Bluetooth モジュールを引き出す必要があります。そうしないと、プログラムのアップロードに失敗します）。



## APP を開く

Bluetooth を使用して電話を車に接続した後、次のようにデータを設定します：



セットアップ後、このボタンを押してください。 UNO ボードのライトがスイッチで変わることがわかります。

## コード

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

このコードブロックの目的は、UNO コントロールボードのボーレートを 9600 に設定し、シリアルポートを開くことです。このようにして、Bluetooth モジュールの元のボーレートが 9600 であるため、互いに通信することができます。

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

```
if(getstr=='A'){
```

```
stateChange();
```

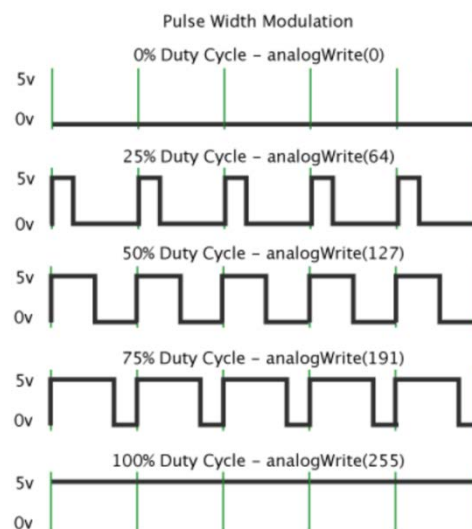
この機能は、循環機能内で繰り返し実行されます。 まず、シリアルポートからデータを読み込み、データをチェックします。 条件を満たす場合は、対応するサブ関数を実行します。 たとえば、シリアルポートから文字 A を読み取ると、LED ライトをオン/オフするサブファンクションが実行されます。

## IV. Bluetooth カーを作る

車が左右に回転させるとき、速度をあまり速く設定する必要はありません。逆に、我々は車の速度を制御する必要があります。しかし、どのように制御するのでしょうか？

答えは PWM です。

PWM は「Pulse Width Modulation」の略でパルス変調と呼ばれ、マイクロプロセッサのデジタル出力でアナログ回路を制御するのに有効な技術であり、矩形波のデューティサイクルを変更することによりモータの速度を変化させます。言い換えれば、モータと電源間を絶えず接続/遮断し、モータを動作させるスイッチです。この時、スイッチング速度が速いため電源がオフになってもモータがオフにはなりません。したがって、パワーオンタイムとパワーオフタイムの特定の値を制御すると、車の速度を制御できます。回路がオンのままのときは車の速度は最大になります。回路がオフしていると車の速度は最小になります。車の速度は、オンとオフの時間が半分の時間の時で中央値になります。PWM はデジタル方式でアナログ量を得る技術です。矩形波はデジタル制御で形成され、矩形波信号はオンとオフの 2 つの状態（デジタルピンの高低）のみを有する。持続時間のオン/オフ時間の特定の値を制御することにより、電圧を 0V 5V に変化させる。占有時間（アカデミーでは高レベル）はパルス幅と呼ばれるので、PWM はパルス幅変調とも呼ばれます。下の 5 つの方形波による PWM について学びましょう。



上の緑の縦線は方形波の周期を表します。すべての `analogWrite` (値) に書き込まれる値はパーセンテージに対応し、パーセンテージはデューティサイクルとも呼ばれ、期間内のハイレベルとローレベルの間の特定の値から得られたパーセンテージを参照します。図の上から順に、最初の矩形波、デューティ・サイクルは 0%、対応する値は 0 です。出力回路電流は最小であり、モーターは静止しています。継続時間が長いほど、回路電流が大きいほど速くなります。最終的なデューティサイクルは 100% で、対応する値は 255 で、モータは最高速度で回転します。50% は中低速回転速度で、25% は比較的遅く、始動できません（静的摩擦のために回路電流が比較的大きくなります）。PWM は主に LED の光を調整したりモータの回転速度を調整するために使用され、モータによって制御される車輪速度は容易に制御されます。PWM の利点は、あなたが Arduino の車で遊ぶときに、より実感できることです。

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

analogWrite () は、PWM ポートに対して 0?255 のアナログ値を書き込むために使用されます。あなたが注意する必要があるのは、analogWrite () は PWM の機能を持つデジタルピンにのみ使用されるということです。 UNO で PWM 機能を持つピンは 3, 5, 6, 9, 10, 11 のデジタルピンのみです。

私たちの車の速度は、ENA と ENB のピン 5 とピン 10 を接続することによって制御されます。以下のプログラムは、デジタル関数 int ABS = 135 を設定しています。

速度はプログラムの下で制御されるので、あなた自身でスピードをコントロールすることができます。

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

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

基本的な知識を習得したら、以下のようにプログラムを車にアップロードし、bluetooth\_car ¥ bluetooth\_car.ino ファイルを開きます。

```
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();
}
}
```

UNO 制御基板にプログラムをアップロードし、車両の電源を入れて地上に置きます。  
モバイル APP を開き、次のようにパラメータを設定します。

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

Clear OK Cancel

Button editor

Left

Status OFF

l

☒ Character
 ☐ Hexadecimal

Left

Status ON

l

☒ Character
 ☐ Hexadecimal

Background color

Clear OK Cancel

Button editor

Right

Status OFF

r

☒ Character
 ☐ Hexadecimal

Right

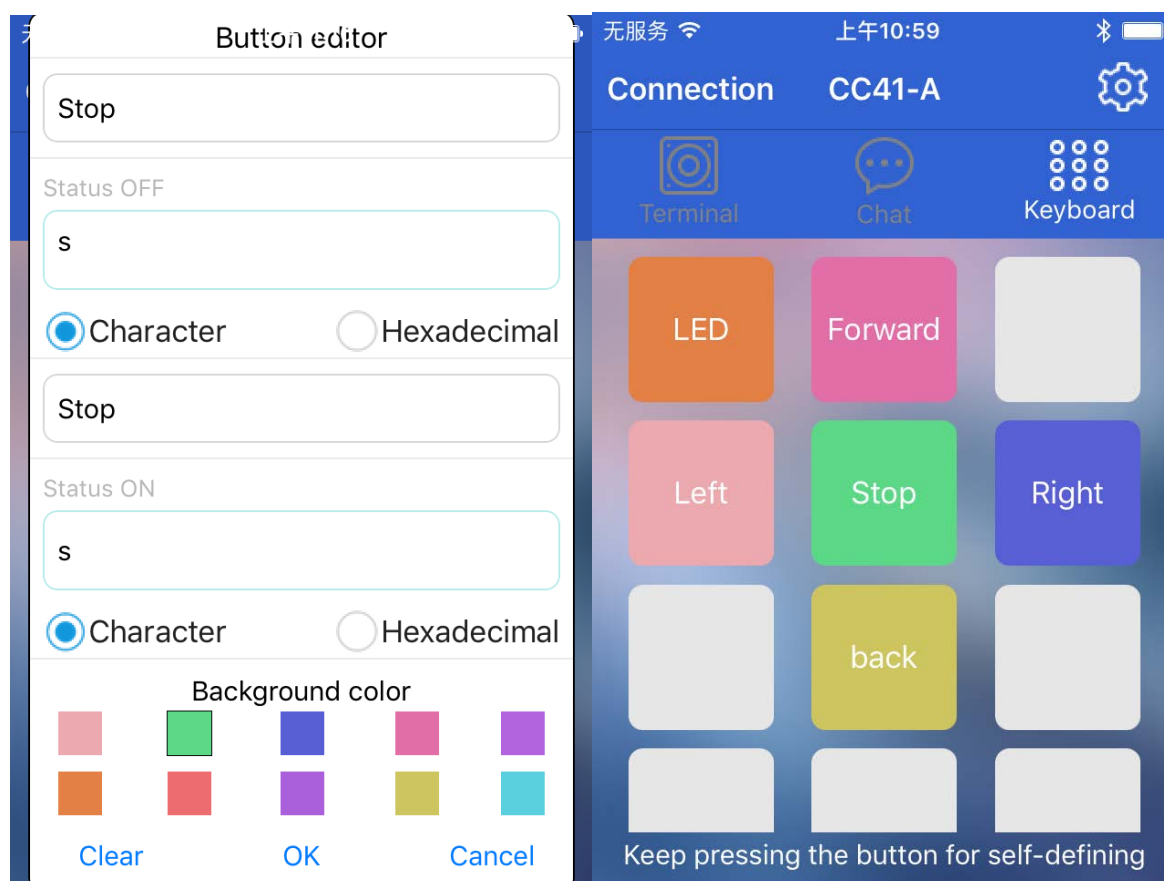
Status ON

r

☒ Character
 ☐ Hexadecimal

Background color

Clear OK Cancel



これで Bluetooth で車を制御し、遊ぶことができます。