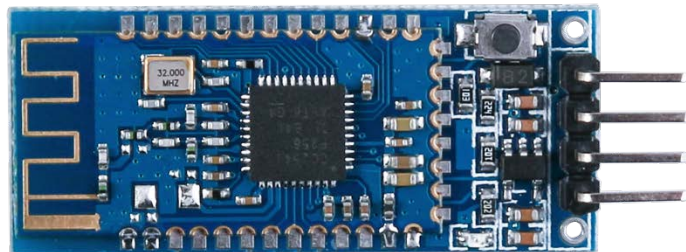


## Lección 2 coche Bluetooth



### Puntos de esta sección

Es muy importante y tan divertido controlar su coche de forma inalámbrica en un cierto espacio cuando aprendemos el Arduino, por lo que en esta lección, le enseñaremos cómo controlar un coche por Bluetooth.

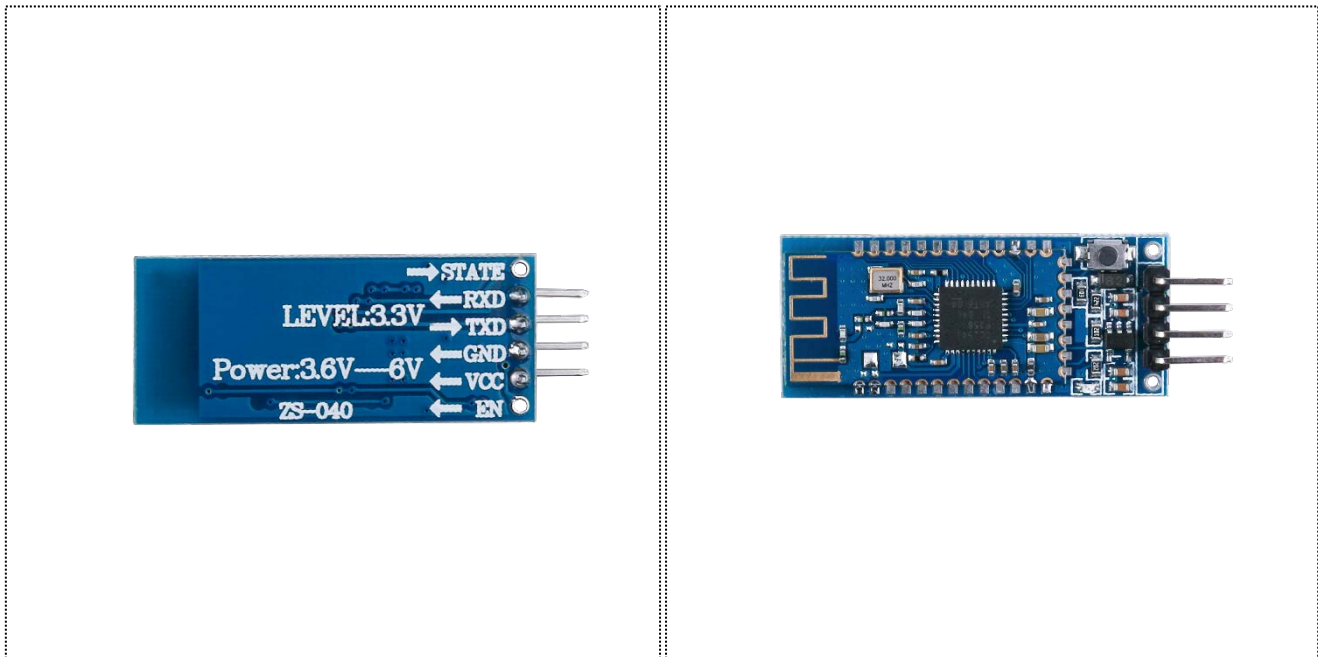
Aprendiendo piezas:

- ◆ Aprenda a utilizar el módulo Bluetooth CC41-A y la aplicación Bluetooth
- ◆ Aprenda cómo controlar el vehículo a través de Bluetooth
- ◆ Escribir programas para implementar esta función

Preparativos:

- ◆ Un vehículo (equipado con batería)
- ◆ Un cable USB
- ◆ Un modulo bluetooth CC41-A Bluetooth
- ◆ Un iPhone o tablet

## I . Modulo Bluetooth

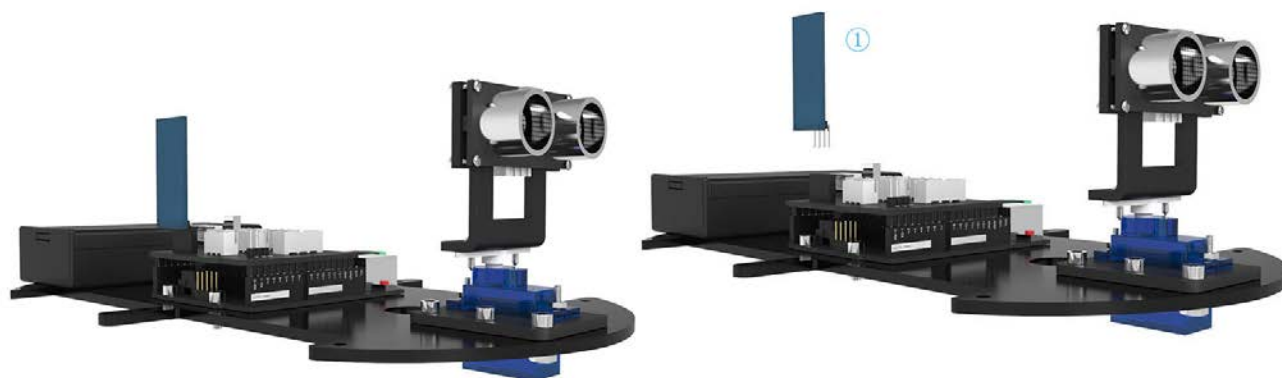


### La descripción del módulo CC41-A:

1. Adopte el microprocesador principal de Bluetooth de CSR, estándar del protocolo de BluetoothV2.0
2. El voltaje de trabajo análogo del puerto serial es 3.3V
3. Los usuarios pueden establecer la velocidad en baudios 1200, 2400, 4800, 9600, 19200, 38400, 57600, 1150200
4. La dimensión del módulo principal es 28mm x 15 milímetros x 2.35mm
5. Corriente eléctrica de trabajo: 40MA
6. Corriente eléctrica de la inactividad: menos de 1MA
7. Usándose para el sistema de navegación del GPS, sistema hidroeléctrico de la lectura del gas, sistema de control industrial de la mina del campo
8. Puede ser conectado al ordenador portátil de Bluetooth, computadora con el adaptador de Bluetooth, PDA, etc.

**Este es el diagrama esquemático del módulo Bluetooth conectado a la tarjeta de controlador UNO:**

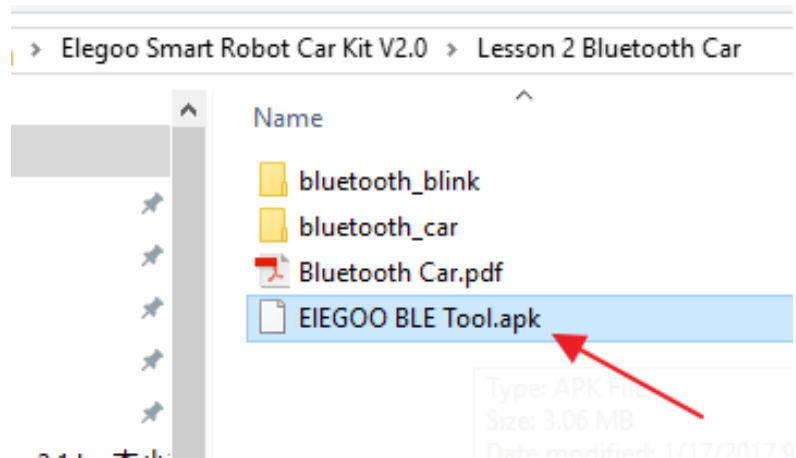
En el experimento lo conectaremos a la placa UNO a través de la tarjeta de expansión V5.



① bluetooth module

## II. Bluetooth APP

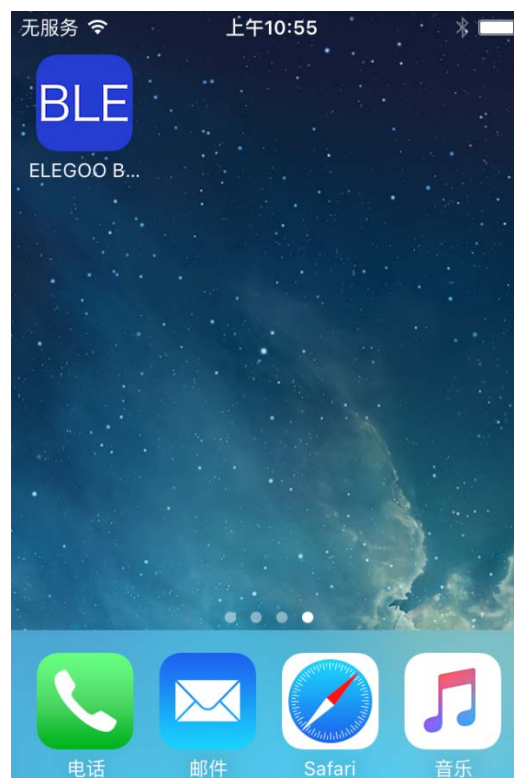
Para el sistema Android, copie el archivo APK a los productos Android e instálelo.



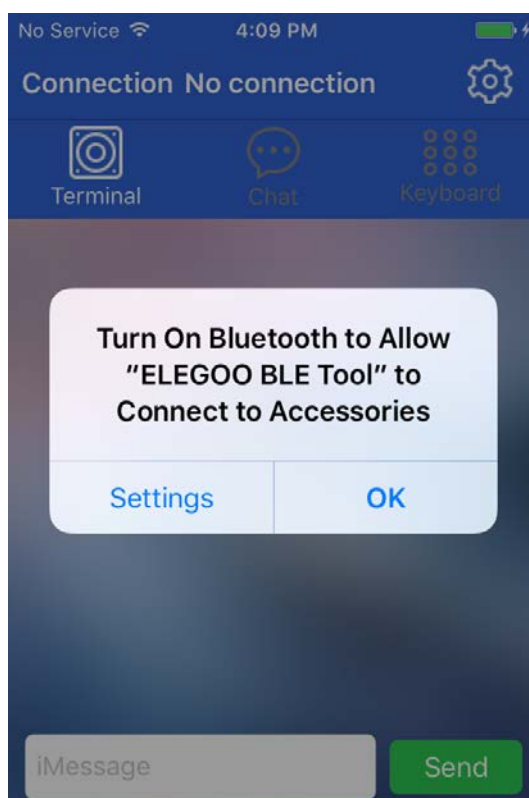
A continuación, tome el teléfono móvil o la tableta de Apple, por ejemplo:

Búsqueda "ELEGOO BLE Tool" en Apple Store.

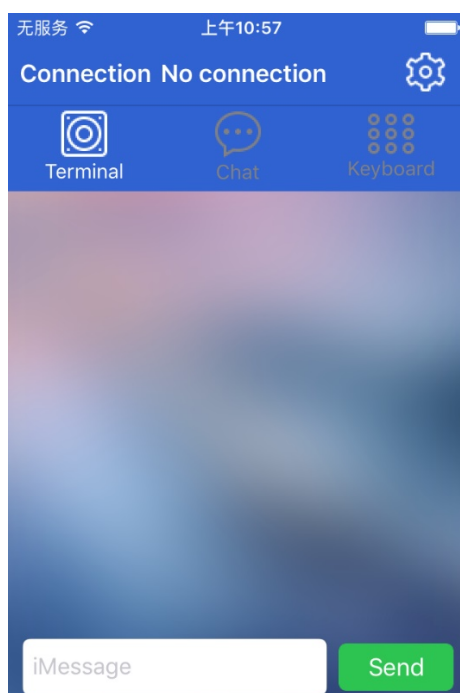
Instale la APP y luego ejecútela en su teléfono.



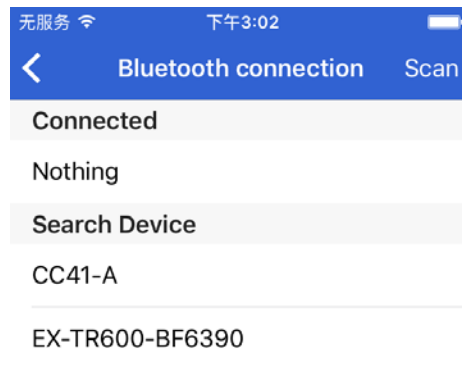
Se mostrará como golpe y le aconsejará que active la función Bluetooth cuando la ejecute.



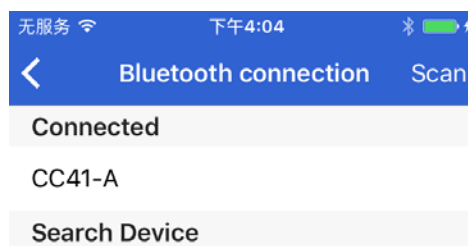
Haga clic en Conexión.



A continuación, el teléfono buscará equipos Bluetooth cercanos.

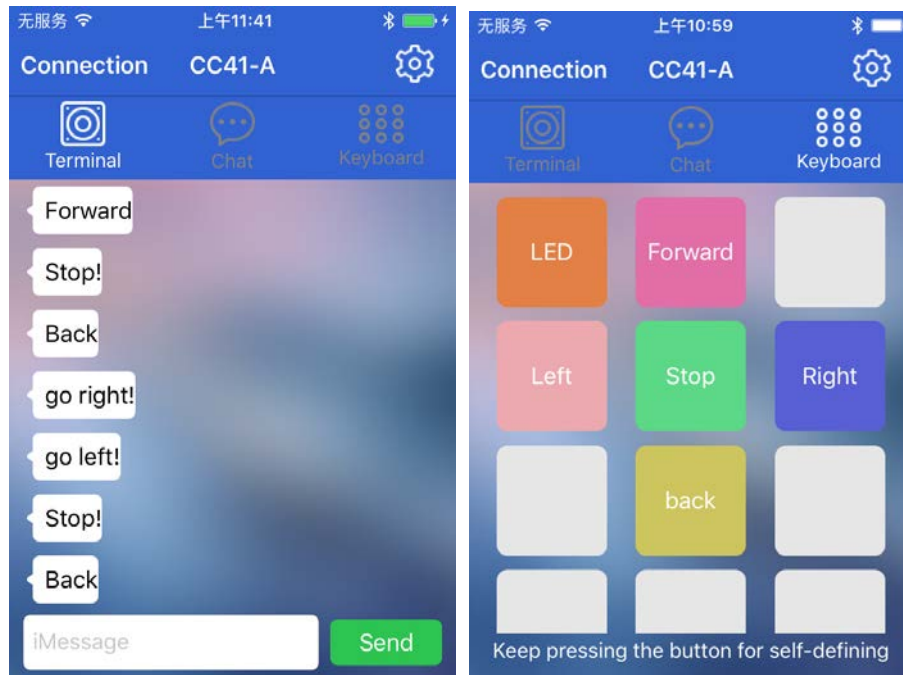


Haga clic en CC41-A, cuando la conexión sea correcta, se mostrará la pantalla Device Connected

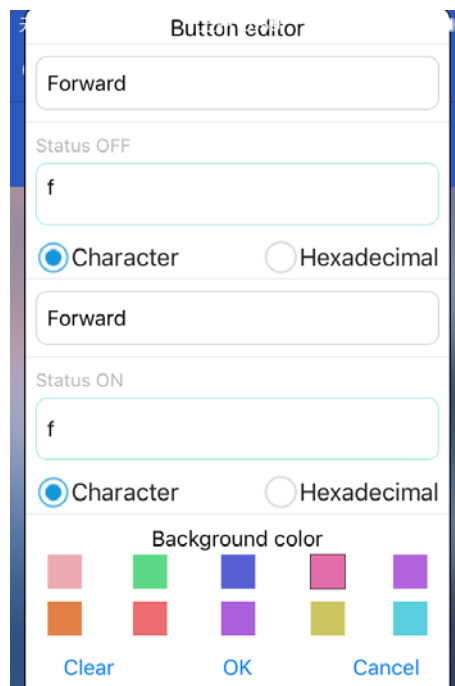


Device Connected

La interfaz de la aplicación se mostrará como la izquierda, a continuación, podemos deslizar la pantalla hacia el lado izquierdo por nuestro dedo y podemos obtener el patrón de la clave como la derecha de abajo:



Por último, establecemos la definición de cada botón, vamos a tomar "goforward" por ejemplo, por favor ver más abajo, y el resto valores-clave se establecen de la misma manera.

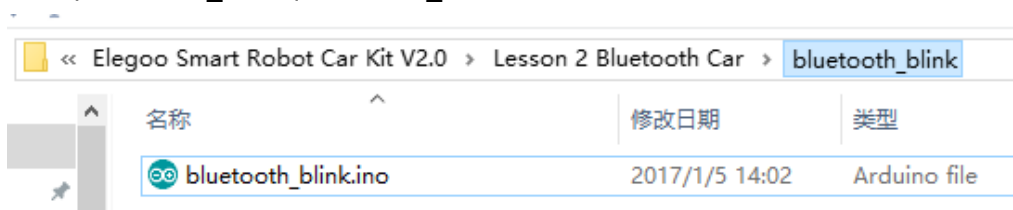


### III. Pruebas

El programa de pruebas es el siguiente:

```
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();
    }
}
```

Abra el fichero : \bluetooth\_blink\bluetooth\_blink.ino

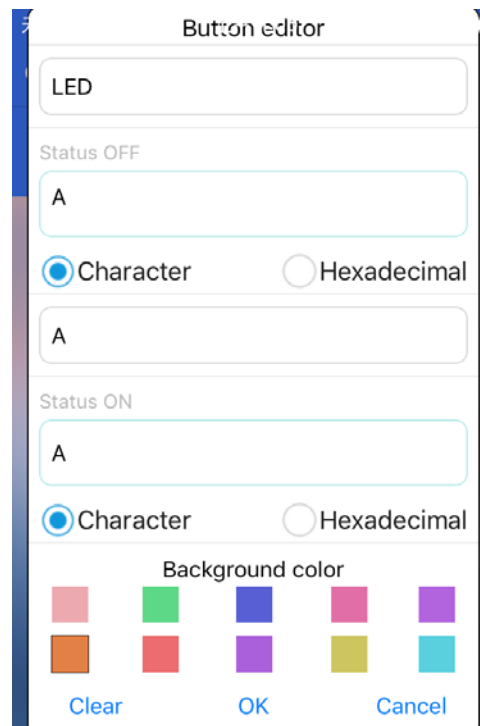


Cargue el programa en la placa UNO, desconéctelo del PC y luego encienda la fuente de alimentación del coche. (CONSEJOS: El módulo Bluetooth se debe extraer cuando cargue el programa, o no podrá cargar el programa).



## Abrir APP

Después de conectar el teléfono al coche con Bluetooth, fijamos datos como abajo:



Después de la configuración, pulse este botón. Usted encontrará que la luz en el tablero UNO cambia con el interruptor.

## El código

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

El propósito de este bloque de código es establecer la velocidad en baudios de la placa de control UNO como 9600 y abrir el puerto serie. De esta manera, pueden comunicarse entre sí, porque la velocidad de transmisión original del módulo Bluetooth es de 9600.

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

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

```
stateChange();
```

Esta función se ejecuta repetidamente dentro de la función de circulación. Primero leerá los datos del puerto serie y comprobará los datos. Si cumple con la condición, ejecutará la subfunción correspondiente. Por ejemplo, si lee la letra A del puerto serie, ejecutará la subfunción responsable de encender / apagar la luz LED.

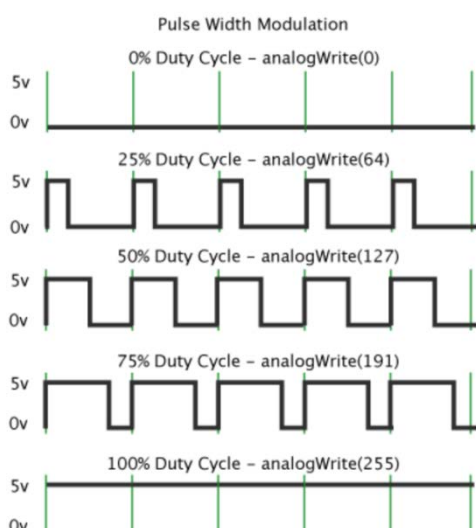


## IV. Hacer un coche Bluetooth

Cuando el coche gira a la izquierda o a la derecha, no es necesario ajustar la velocidad demasiado rápido. Por el contrario, tenemos que controlar la velocidad del coche. Pero, ¿cómo controlarlo?

La respuesta es PWM.

PWM es la abreviatura de "Pulse Width Modulation", se llama modulación de impulsos en resumen, es una tecnología eficaz para controlar el circuito analógico con salida digital de microprocesador, coche se utiliza para cambiar la velocidad del motor mediante la alteración del ciclo de trabajo de una onda cuadrada. En otras palabras, conectar y romper el circuito entre los dos lados del motor constantemente, es el interruptor de mantenimiento del motor de trabajo, el motor no se apagará cuando la energía está apagado debido a la velocidad rápida. Así que podemos controlar la velocidad del coche si controlamos el valor específico de la potencia en el tiempo y el tiempo de apagado. La velocidad del coche será máxima cuando el circuito se mantenga inmóvil. La velocidad del coche será mínima si el circuito se mantiene apagado. La velocidad del coche será mediana en el medio tiempo. PWM es una tecnología para obtener la cantidad analógica a través del método digital. Una onda cuadrada está formada por control digital, la señal de onda cuadrada sólo tiene dos estado de encendido y apagado (Que es alto-bajo de pines digitales) .Simulación de voltaje cambiando de 0 a 5V controlando el valor específico de la duración de tiempo de encendido y apagado. Tiempo ocupado de (es decir, alto nivel en la academia) se llama pulso de ancho, por lo que PWM también se llama modulación de ancho de pulso. Aprendamos sobre PWM a través de cinco ondas cuadradas abajo.



La línea vertical verde arriba representa un período de la onda cuadrada. El valor escrito en cada `analogWrite` (valor) corresponde al porcentaje, el porcentaje también se denomina Ciclo de trabajo, se refiere al porcentaje obtenido a partir del valor específico entre el nivel alto de duración y el nivel bajo

en un período. En la figura, de arriba a abajo, la primera onda cuadrada, el ciclo de trabajo es 0%, el valor correspondiente es 0. La corriente del circuito de salida es mínima, el motor se mantiene inmóvil. Cuanto mayor sea el tiempo de duración, mayor será el motor de corriente de circuito, más rápida será la velocidad. Así, el ciclo de trabajo final es 100%, el valor correspondiente es 255, el motor gira a toda velocidad. El 50% es la velocidad de rotación hiponástica media, el 25% es relativamente más lento, incluso no puede comenzar (la corriente del circuito es relativamente grande para arrancar el motor debido a la fricción estática). PWM se utiliza sobre todo para ajustar la luz del LED y girar la velocidad del motor, la velocidad de la rueda controlada por el motor es fácilmente controlable. La ventaja de PWM puede reflejarse más cuando juegas con algunos coches Arduino.

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

`analogWrite()` Se utiliza para escribir un valor analógico de 0 a 255 para puertos PWM. Lo que hay que tener en cuenta es que, `analogWrite()` Sólo se utiliza para los pines digitales con función de PWM. Los pasadores con función de PWM en UNO son sólo clavijas digitales de 3,5,6,9,10,11.

La velocidad de nuestro vehículo se controla mediante la conexión de los pin5 y pin10 de ENA y ENB. El programa siguiente, han establecido una función digital en ABS = 135;

La velocidad se controla con el programa de abajo, por lo que puede controlar la velocidad por su cuenta.

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

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

Después de aprender los conocimientos básicos, subiremos el programa de abajo al coche, abrimos el archivo `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();
  }
}
```

Cargue el programa en la placa de control UNO, encienda la fuente de alimentación del vehículo y póngala en el suelo.

Abra la aplicación móvil y configure los parámetros de la siguiente manera.

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

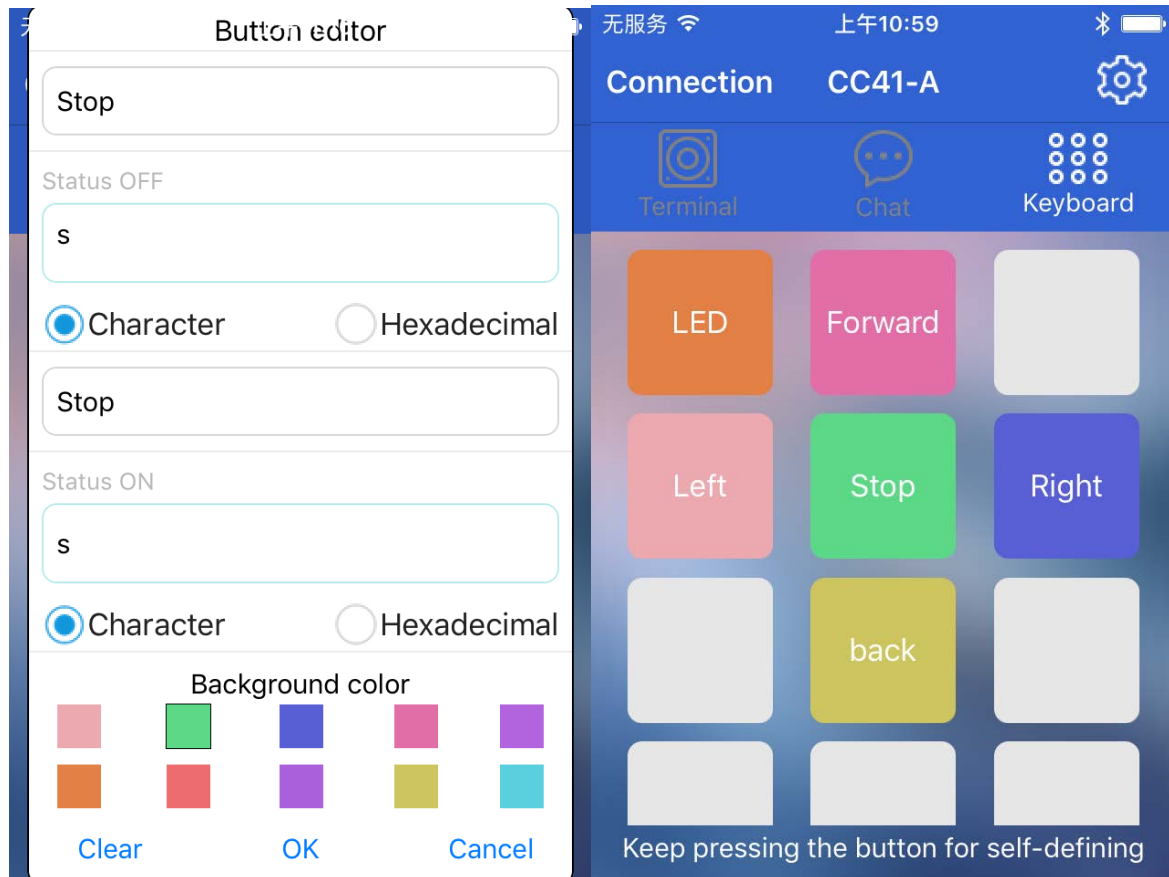
r

☒ Character
 ☐ Hexadecimal

Background color

Clear OK Cancel





Voila, ahora podemos controlar el coche por Bluetooth y jugar con él.