

A background image showing a top-down view of an orange robot chassis with a black wheel and a tangle of colorful wires (red, yellow, blue, black) protruding from the top. A semi-transparent teal rectangle is overlaid on the center of the image, containing the project title.

PROYECTO N° 7

Controla el Code&Drive
vía Bluetooth con una App

PROYECTO N° 7: Controla el Code&Drive vía Bluetooth con una App

Crea un programa para controlar el Code&Drive desde tu *smartphone* o tableta Android vía Bluetooth, utilizando una App.

NIVEL DE DIFICULTAD: Avanzado.

DURACIÓN DEL EJERCICIO: 60 min.

MATERIALES:

- 1 Code&Drive
- 1 Cable USB - Micro USB
- Ordenador
- 1 Smartphone o tableta Android con la App utilizada para el proyecto (Arduino Bluetooth RC Car)

El kit Code&Drive deberá estar montado de acuerdo a las instrucciones indicadas en el manual.

¿Cómo funciona la placa Build&Code 4in1 en el Code&Drive?

Para realizar este proyecto utilizarás los pins digitales, los pins analógicos, conexiones para motores DC y la comunicación Bluetooth del Code&Drive.

- **Pins digitales:** están situados en la parte izquierda de la placa Build&Code 4in1. El código de color es Negro (GND), Rojo (VCC) y Amarillo (Digital I/O Signal). Encontrarás 12 conectores numerados del 2 al 13, de los cuales 7 pueden suministrar señal de salida PWM, indicados con el símbolo ~ junto al número.
- **Pins analógicos:** están situados en la parte derecha de la placa Build&Code 4in1. El código de color es Negro (GND), Rojo (VCC) y Azul (*Analog Input*). Encontrarás 5 conectores numerados del A0 al A5.
- **Transmisión de datos entre el PC y la placa Build&Code 4in1:** conecta la placa Build&Code 4in1 con el PC debes utilizando el cable USB-Micro USB. El interruptor BLT/USB debe estar en posición de USB.
- **Transmisión de datos entre la placa Build&Code 4in1 y el smartphone Android:** coloca el interruptor BLT/USB en posición BLT para permitir la comunicación entre la placa Build&Code 4in1 y el dispositivo Android.
- **Configuración del Bluetooth en el *smartphone* o tableta Android:** alimenta y enciende la placa Build&Code 4in1. Enciende el Bluetooth de tu smartphone, busca la señal llamada "HC-06" y selecciónala para establecer la conexión. Se te pedirá una contraseña que es 1234 o 0000.
- **Alimentación de la placa Build&Code 4in1:** puedes alimentar la placa Build&Code 4in1 desde el mismo cable USB conectado en un PC. Verás que se encenderá de forma inmediata y tendrás alimentación para los sensores y el Bluetooth, pero no para los motores DC. También puedes alimentar la placa Build&Code 4in1 vía un conector Jack de 5.5mm de diámetro, utilizando el adaptador externo conectado a una toma de corriente de 12VDC cómo máximo.
- **Alimentación de los motores DC:** debes usar la alimentación del porta pilas, al cual va conectado el Jack de alimentación. Para encender la placa Build&Code 4in1 hay que poner en ON el interruptor ON/OFF situado en la parte superior de la misma placa.

¿Cómo funcionan los sensores del Code&Drive?

- **LEDs azul y verde:** un LED es un Diodo Emisor de Luz (Light Emitting Diode). En este caso, los colores que emiten son azul y verde. Puedes encender y apagar estos componentes utilizando los botones de la App correspondientes a la luz delantera y la luz trasera. Puedes

conectar los LEDs en cualquiera de los pines digitales de la placa Build&Code 4in1 excepto los nº 4, 5, 6 y 7, ya que estos están reservados para los motores. Para programar su funcionamiento hay que configurar los pines digitales donde vayan conectados como salida (*Output*).

- **Buzzer:** es un timbre eléctrico que produce un sonido o zumbido continuo o intermitente de un mismo tono, generalmente agudo. Puedes controlar este componente con el botón de la bocina de la App y puedes conectarlo en cualquiera de los pines digitales de la placa Build&Code 4in1 excepto los nº 4, 5, 6 y 7, que están reservados para los motores. Para programar su funcionamiento debes configurar los pines digitales donde vayan conectados como salida (*Output*).
- **Motores DC:** son máquinas que convierten energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción de un campo magnético. Puedes controlar el sentido del giro utilizando las flechas de la App, y controlar la velocidad utilizando la barra de control. Para que los motores funcionen correctamente deberás alimentar la placa Build&Code 4in1 con el portapilas y poner en ON los 4 interruptores de color blanco que se encuentran en el interruptor DIP.

Para la programación de los motores DC utilizarás los puertos digitales 4, 5, 6 y 7.

Los puertos digitales 4 y 7 son para controlar el sentido de giro de los motores y se programan como salidas de señal digital 1 o 0, o *HIGH* y *LOW*.

Los puertos digitales 5 y 6 son para controlar la velocidad de los motores y se programan como salidas de señal PWM con valores de 0 a 255.

CONEXIONES:

1. Realiza las conexiones de los motores a los pines digitales como se indica en la siguiente tabla:

Motores DC	Pin Digital
Motor A (Derecha)	4 (Sentido de giro), 5 (Velocidad)
Motor B (Izquierda)	7 (Sentido de giro), 6 (Velocidad)

1. Conecta los sensores a la placa Build&Code 4in1 como se indica a continuación:

Sensor/Actuador	Pin Digital
1 LED azul	9
1 LED verde	10
1 zumbador	8

Motores DC	Conexiones
Motor A (Derecha)	A01 (Cable rojo), A02 (Cable negro)
Motor B (Izquierda)	B01 (Cable rojo), B02 (Cable negro)

Controlar el Code&Drive con la App Arduino Bluetooth RC Car

La aplicación que recomendamos para controlar el Code&Drive es Arduino Bluetooth RC Car y [puedes descargarla en Google Play](#).



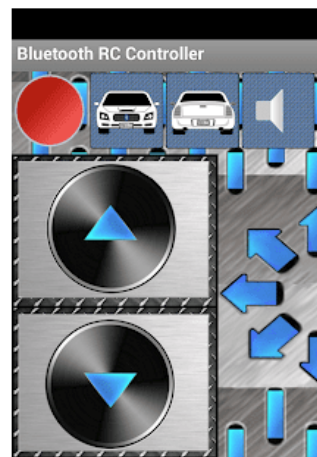
Arduino Bluetooth RC Car

Andi.Co Educación

★★★★★ 1,899

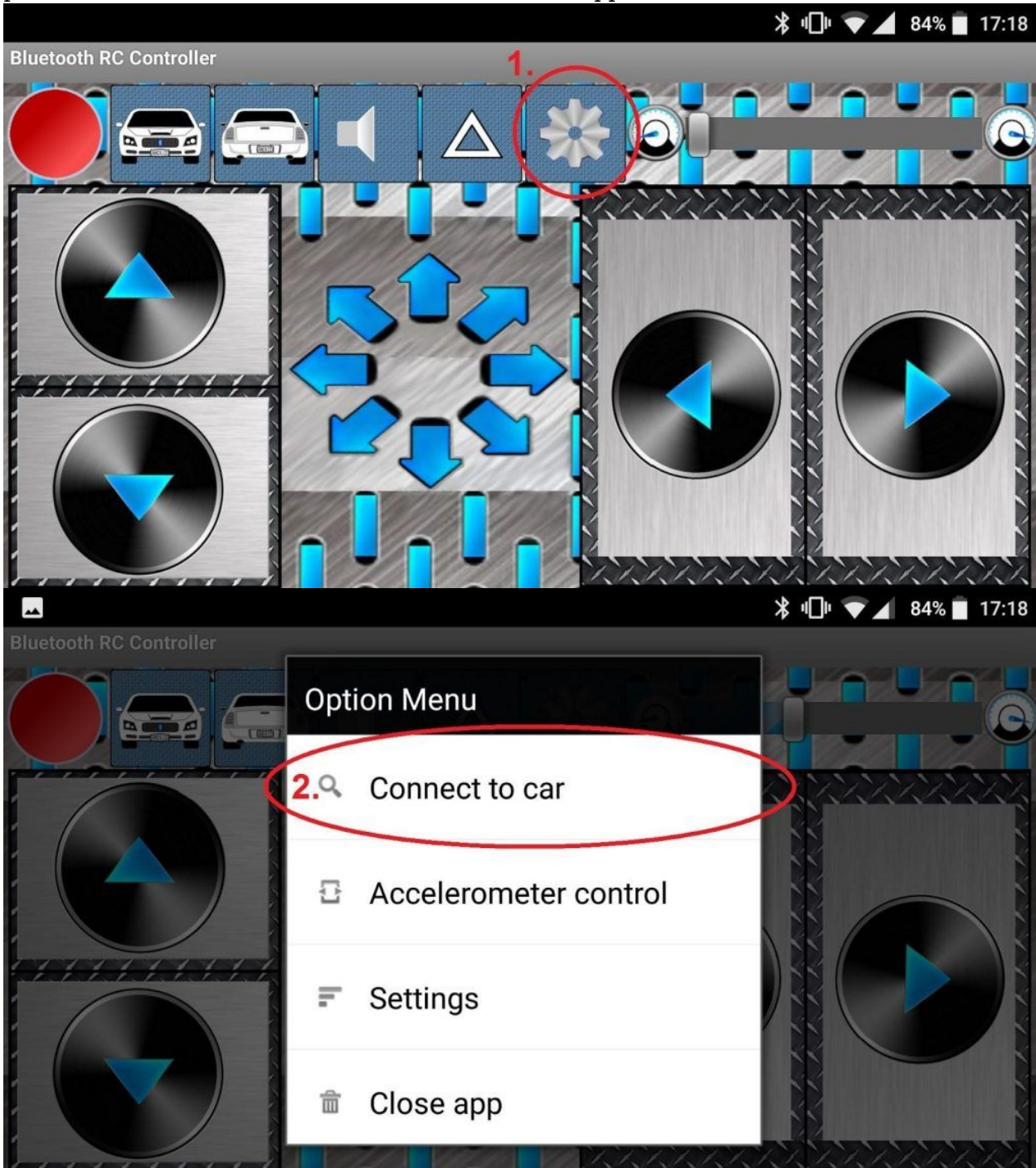
PEGI 3

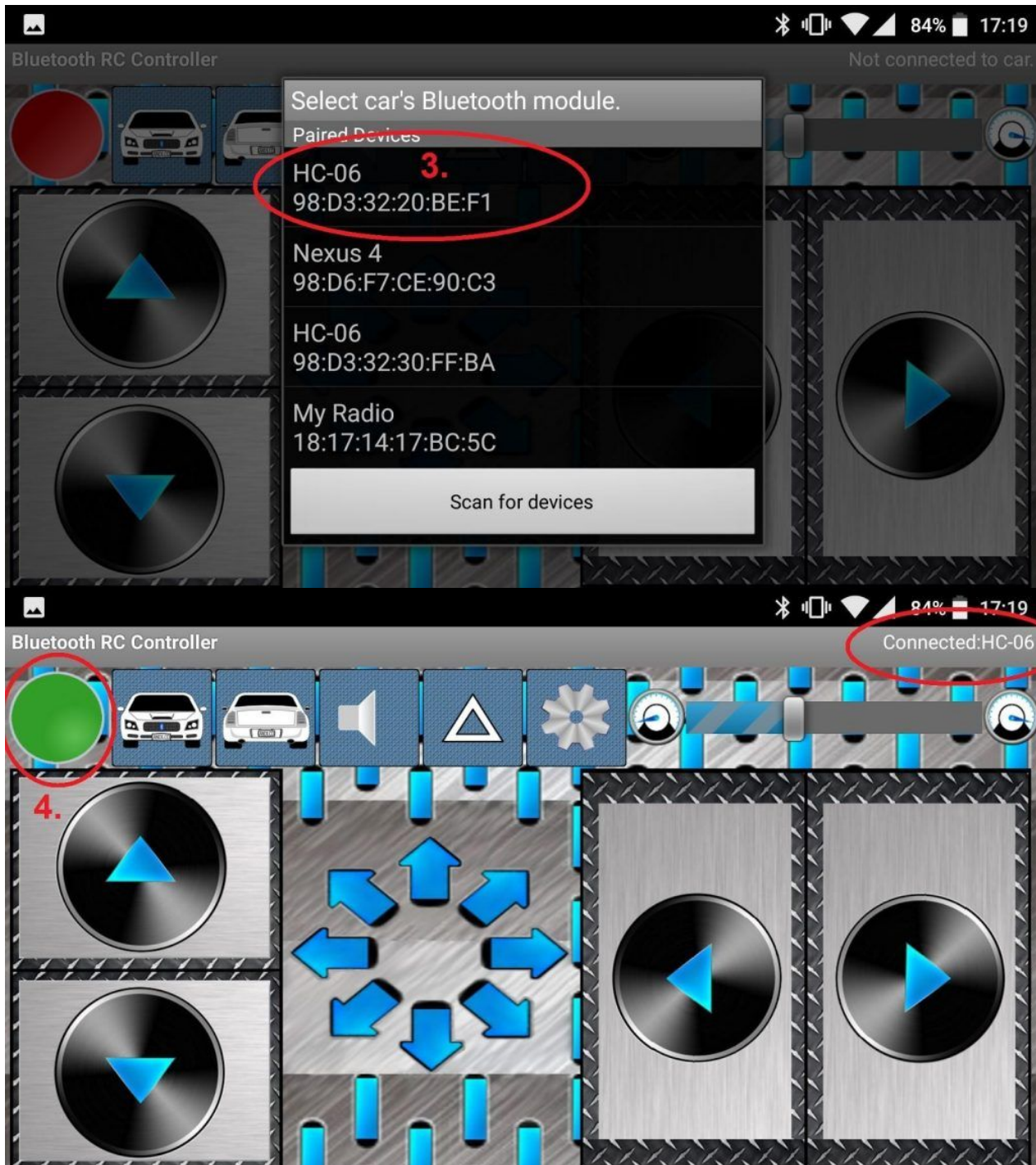
Instalar



Para configurar el Bluetooth ve a "Option Menu" → "Connect to car" y busca en la lista el HC-06 que has vinculado con tu *smartphone* previamente. Cuando esté correctamente conectado, el indicador rojo parpadeante se pondrá de color verde, indicando una correcta conexión con el Code&Drive. Ya

podrás comenzar a controlar tu Code&Drive con la App.





CÓDIGO DE PROGRAMACIÓN

Puedes realizar esta actividad utilizando el *software* Arduino. A continuación encontrarás el código de programación necesario.

Código Arduino

1. [Descarga el software Arduino](#) y realiza el proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente programa:

```
char data = 0; // VARIABLE DE LECTURA BLUETOOTH;
int directionA = 4, speedA = 5, directionB = 7, speedB = 6, Speed ;
//PINS DE VELOCIDAD Y DIRECCIÓN DE LOS MOTORES A Y B;
int LedV = 9, LedB = 10; // PIN DE LED VERDE Y AZUL;
int pinBuzzer = 8; // PIN DEL BUZZER;

void setup()
{
  Serial.begin(9600);
  // CONFIGURACIÓN DE LOS PINES DEL BUIL&CODE 4IN1
  pinMode ( directionA, OUTPUT);
  pinMode ( speedA, OUTPUT);
  pinMode ( directionB, OUTPUT);
  pinMode ( speedB, OUTPUT);
  pinMode ( LedV, OUTPUT);
  pinMode ( LedB, OUTPUT);
}

void loop()
{
  if(Serial.available() > 0)
  {
    data = Serial.read(); //LECTURA DE ENTRADA DE DATOS
    BLUETOOTH
    // INTERPRETACIÓN DE LOS DATOS DE ENTRADA VIA BLUETOOTH
    switch (data)
    {
      case 'F': // ADELANTE;
        //MOTOR A
        analogWrite (speedA, Speed);
        digitalWrite ( directionA,HIGH);

        //MOTOR B
        analogWrite (speedB, Speed);
        digitalWrite ( directionB,LOW);
        break;

      case 'B': // ATRAS;
        //MOTOR A
        analogWrite (speedA, Speed);
        digitalWrite ( directionA,LOW);
    }
  }
}
```

```
//MOTOR B
analogWrite (speedB, Speed);
digitalWrite ( directionB,HIGH);
break;

case 'S': // REPOSO;
//MOTOR A
analogWrite (speedA, 0);
digitalWrite ( directionA,LOW);

//MOTOR B
analogWrite (speedB, 0);
digitalWrite ( directionB,HIGH);
break;

case 'L': // IZQUIERDA; LEFT
//MOTOR A
analogWrite (speedA, Speed);
digitalWrite ( directionA,HIGH);
break;

case 'R': // DERECHA; RIGHT
//MOTOR B
analogWrite (speedB, Speed);
digitalWrite ( directionB,LOW);
break;

// VELOCIDAD VARIABLE
case '1':
Speed = 25;
break;

case '2':
Speed = 25*2;
break;

case '3':
Speed = 25*3;
break;

case '4':
Speed = 25*4;
break;

case '5':
Speed = 25*5;
```



```
break;

case '6':
Speed = 25*6;
break;

case '7':
Speed = 25*7;
break;

case '8':
Speed = 25*8;
break;

case '9':
Speed = 25*9;
break;

case 'q':
Speed = 255;
break;

// LUCES DELANTERAS
case 'W':
digitalWrite (LedV, HIGH);
break;

case 'w':
digitalWrite (LedV, LOW);
break;

// LUCES TRASERAS; BACK LIGHT
case 'U':
digitalWrite (LedB, HIGH);
break;

case 'u':
digitalWrite (LedB, LOW);
break;

// BOCINA
case 'V':
tone(pinBuzzer,523.25,0);
break;

case 'v':
```

```
        tone(pinBuzzer,0,100);  
        break;  
    }  
}  
}
```

3. Configura y carga el código, siguiendo las instrucciones indicadas en la guía de Primeros Pasos del kit Code&Drive.
4. Revisa que el interruptor BLT/USB de la placa controladora Build&Code 4in1 esté en posición USB, para una correcta carga del código.

RESULTADO DEL EJERCICIO

Con este programa controlarás el Code&Drive desde la App, utilizando tu *smartphone* o tableta. Con las flechas podrás controlar la dirección, con la barra de control variarás la velocidad, con los botones de luces delanteras y traseras encenderás/apagarás los LEDs, y con el botón de la bocina harás sonar la bocina.