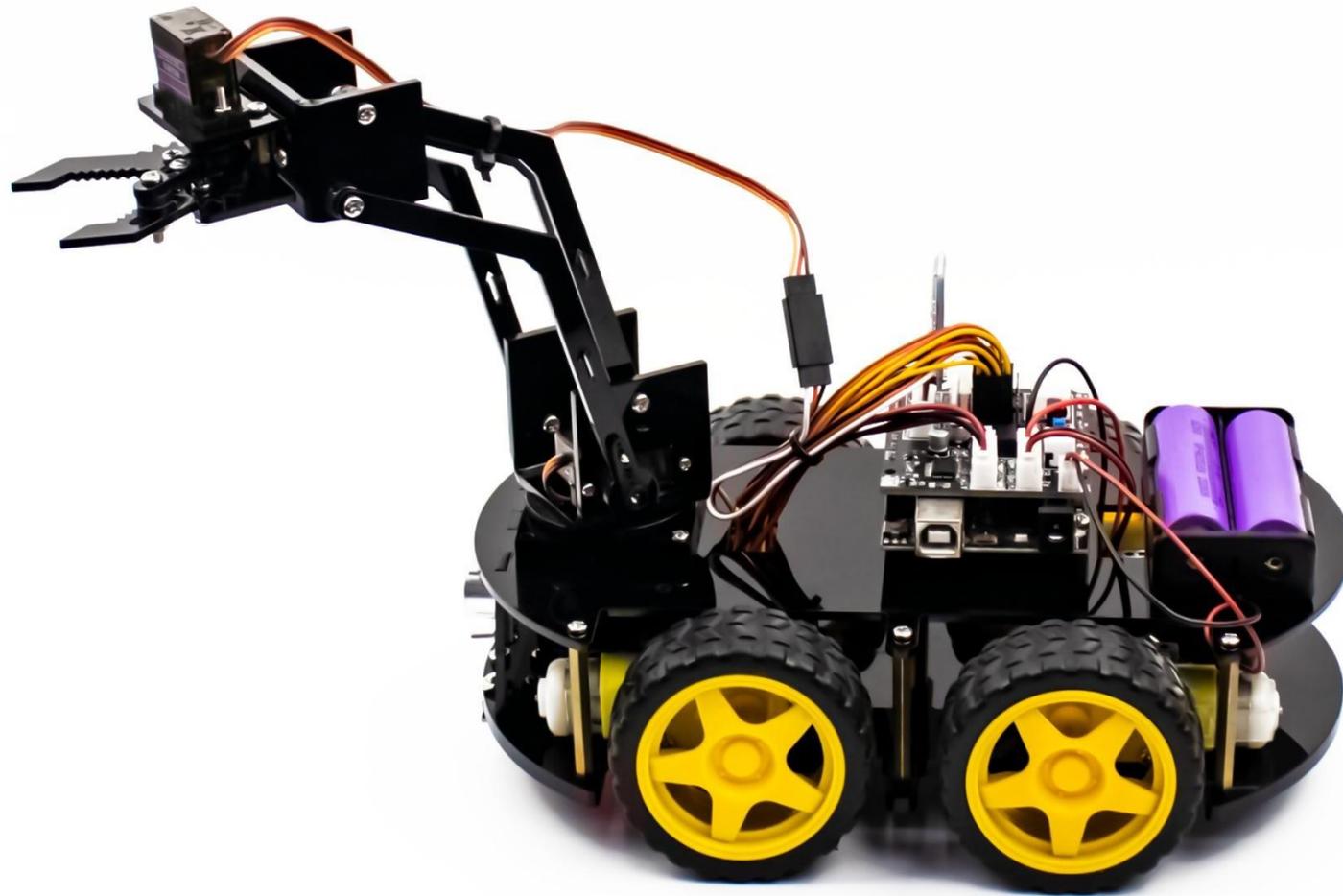




Kit Vehículo Inteligente 4WD con Brazo Robótico



Contenido

Introducción al Kit Vehículo Inteligente 4WD con Brazo Robótico	2
Lección 1 Introducción al IDE de Arduino	6
Cómo Instalar Arduino IDE	6
Cómo instalar el controlador de Arduino	14
Cómo agregar bibliotecas	21
Prueba de parpadeo (prueba tu primer programa)	25
Lección 2 Servocontrol y calibración del ángulo de instalación.....	38
Lección 3 Control de velocidad y dirección del motor	49
Lección 4 Seguimiento de líneas Smart Car.....	56
Lección 5 Vehículo robot anticaída.....	70
Lección 6 Vehículo robot ultrasónico para evitar obstáculos	76
Lección 7 Vehículo Robot con seguimiento Ultrasónico	86
Lección 8 Vehículo Robot de control remoto infrarrojo.....	93
Lección 9 7-en-1 Brazo robótico multifunción Smart Car	103

Perfil de la empresa

Fundada en 2011, LAFVIN es un fabricante y comerciante especializado en investigación, desarrollo y producción de Mega2560, UNO, Nano placas, y todo tipo de accesorios o sensores de uso para Arduino, Raspberry. También completamos kits de inicio diseñados para los amantes interesados de cualquier nivel para aprender Arduino o Raspberry. Estamos ubicados en Shenzhen, China. Todos nuestros productos cumplen con los estándares internacionales de calidad y son muy apreciados en una variedad de diferentes mercados en todo el mundo.

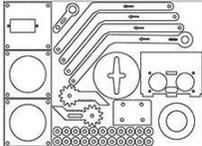
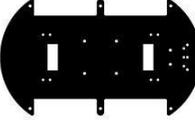
Servicio al Cliente

Estamos cooperando con muchas empresas de diferentes países. También ayudarlos a comprar productos de componentes electrónicos en China, y se convirtió en el mayor proveedor de ellos. Esperamos construir cooperar con más empresas en el futuro. Por cierto, también esperamos escuchar de usted y cualquiera de sus comentarios críticos o sugerencias. Como una empresa continua y de rápido crecimiento. Seguimos esforzándonos al máximo para ofrecerle excelentes productos y un servicio de calidad.

Tutoriales

Este tutorial incluye códigos, bibliotecas, lecciones y video de la guía de instalación. Está diseñado para principiantes. Enseñará a todos los usuarios cómo ensamblar el automóvil robot y usar la placa controladora Arduino UNO, sensores, servo y módulo Bluetooth. Proporcione un tutorial de instalación dinámica 3D para construir rápidamente su automóvil robot. La fuente de alimentación utiliza dos baterías de litio 18650 con batería de larga duración. El software de control APP recién creado, las 7 funciones principales del kit de coche robot están completamente controladas por la APP, y el modo de función se puede cambiar libremente.

LISTADO DE PARTES

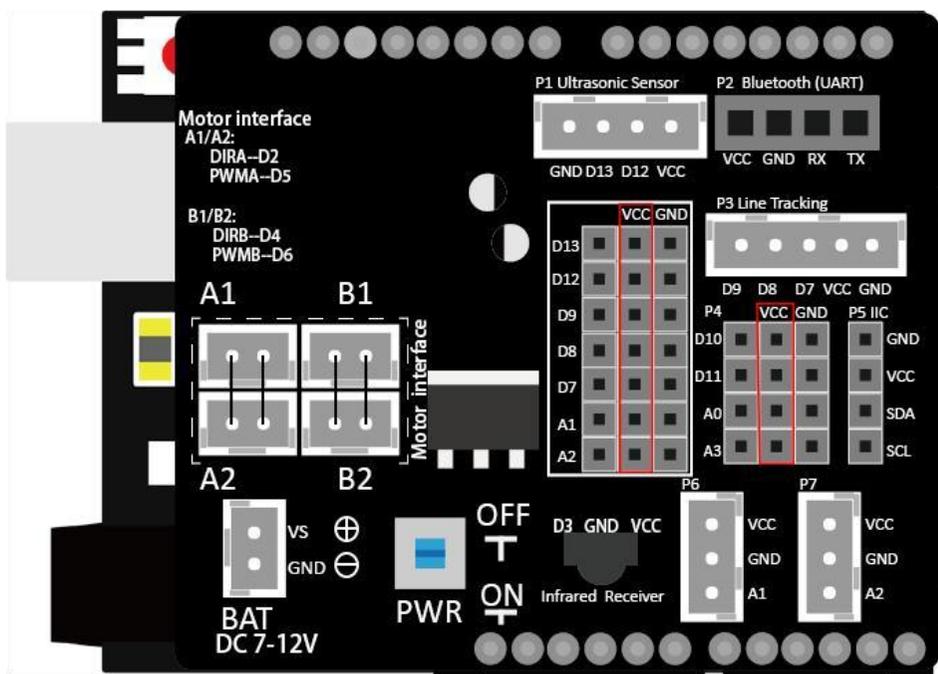
 <p>R3 x1</p>	 <p>Expanding Board x1</p>	 <p>Remote x1</p>	 <p>Ultrasonic x1</p>	 <p>MG90S Servo x3</p>	 <p>Bluetooth x1</p>
 <p>Acrylic A1 x1</p>	 <p>Line Tracking Module x3</p>	 <p>Motor x4</p>	 <p>Tire x4</p>	 <p>Acrylic A2 x1</p>	 <p>Acrylic A3 x1</p>
 <p>Acrylic A4 x1</p>	 <p>USB Cable x1</p>	 <p>Battery Case x1</p>	 <p>Bottom Acrylic Layer x1</p>	 <p>Upper Acrylic Layer x1</p>	 <p>Adhesive Tape x1</p>
 <p>AXK Plain Bearing x1</p>	 <p>Cylinder Screwdriver x1</p>	 <p>Tie x3</p>	 <p>Screw bag</p>	 <p>Screwdriver x1</p>	 <p>CD Tutorial x1</p>
 <p>4pin XH2.54 to F Dupont Wire x1</p>	 <p>3pin Servo Extension Cable x1</p>	 <p>3pin F-F Dupont Wire x3</p>			

Introducción al kit de Vehículo Inteligente 4WD con Brazo Robótico

4WD Robot Arm Smart Car Kit se fabrica principalmente utilizando el control principal Arduino UNO R3 y la placa de expansión de accionamiento del motor TB6612, dos motores de reducción, chasis de marco de acrílico y algunos sensores.

El Kit de Vehículo inteligente 4WD con brazo robótico tiene las siguientes ventajas:

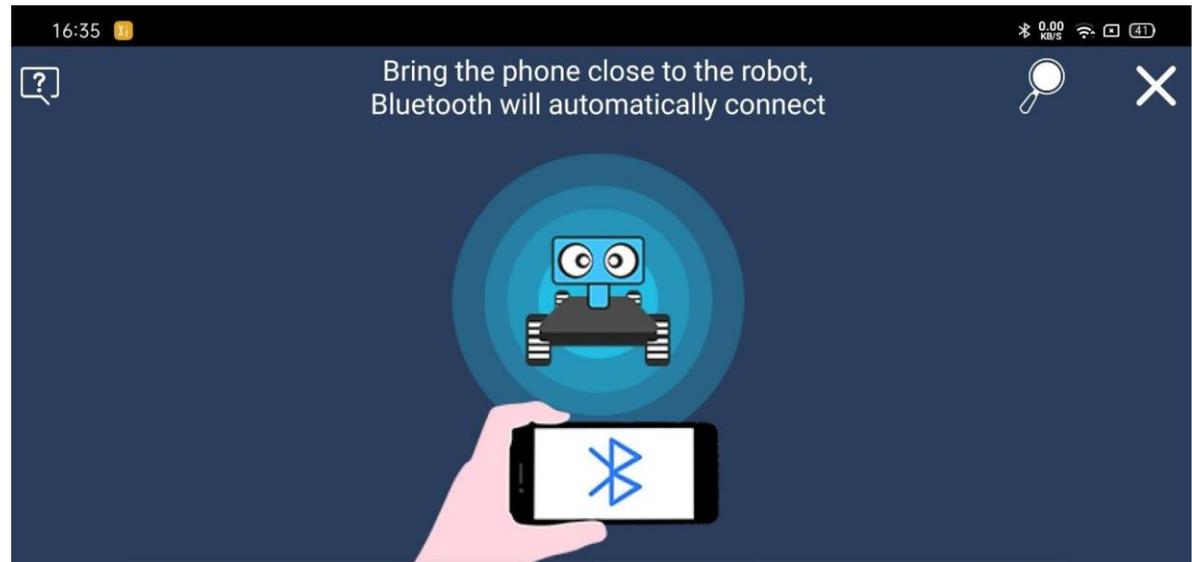
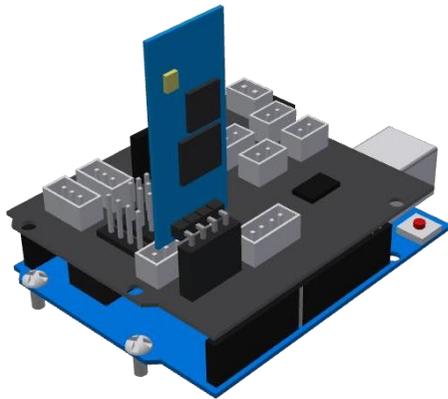
1) La tarjeta de accionamiento del motor Arduino integra el chip del controlador, eliminando el cableado complejo tradicional y el espacio de instalación. El escudo utiliza cableado de interfaz estándar, simple y fácil de entender, cableado de instalación rápida.



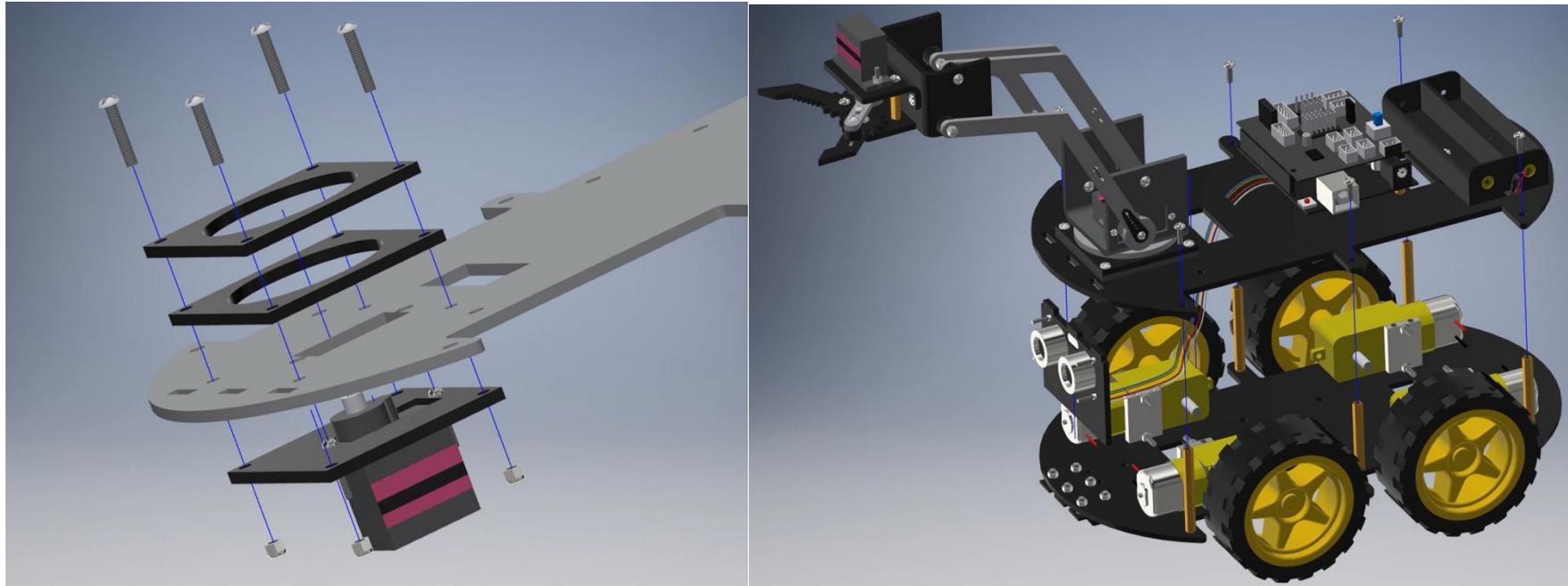
2) El software de control APP recientemente creado, las 7 funciones principales del kit de coche robot están completamente controladas por la APP, y el modo de función se puede cambiar libremente. Haga clic para 7-en-1 multifunción brazo robótico coche inteligente.



3) El módulo Bluetooth de nuevo diseño está conectado al circuito, y el programa de carga no requiere la desconexión manual del módulo Bluetooth. Ayude a los estudiantes principiantes a evitar con éxito la trampa de cargar fallas en el programa. Conecte la alimentación al módulo Bluetooth, luego abra la aplicación de control remoto de la aplicación y acerque el teléfono al módulo Bluetooth y se conectará automáticamente. Ya no necesita el engorroso proceso de conexión anterior.



4) Proporcione un tutorial de instalación dinámica 3D para construir rápidamente su automóvil robot. La fuente de alimentación utiliza dos baterías de litio 18650 con batería de larga duración.



Lección 1 Comenzando la instalación de Arduino IDE

Introducción

Como instalar Arduino IDE

El Entorno de Desarrollo Integrado Arduino (IDE) corresponde al software de la plataforma Arduino. En este proyecto, aprenderá cómo configurar su computadora para usar Arduino y cómo establecer los proyectos que siguen. El software de Arduino que utilizará para programar su Arduino está disponible para Windows, Mac y Linux. El proceso de instalación es diferente para las tres plataformas y desafortunadamente hay una cierta cantidad de trabajo manual para instalar el software.

PASO 1: Ir a <https://www.arduino.cc/en/Main/Software> and find below page.



La versión disponible en este sitio web suele ser la última versión, y la versión real puede ser más reciente que la versión de la imagen.

PASO 2: Descargue el software de desarrollo que sea compatible con el sistema operativo de su computadora. Tomaremos Windows como ejemplo aquí.



Click **Windows Installer**.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **8,808,272** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

Click JUST DOWNLOAD.

También la versión 1.8.0 está disponible en el material que proporcionamos, y las versiones de nuestros materiales son las últimas versiones cuando se realizó este curso.



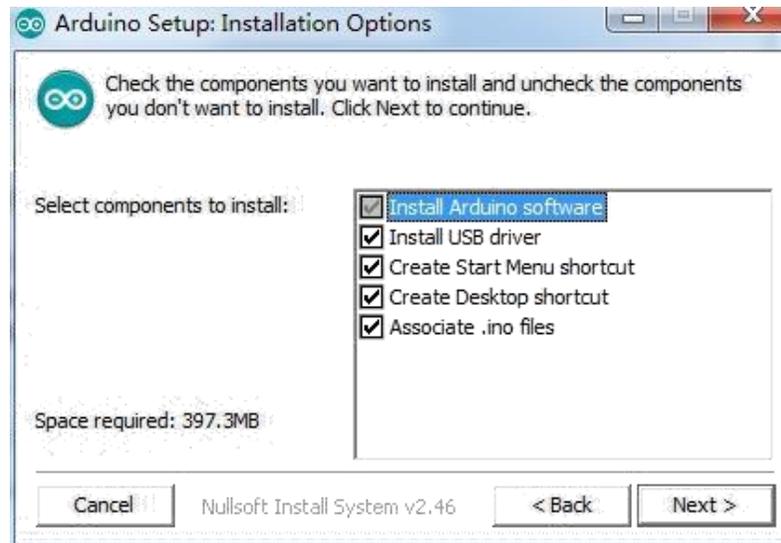
-  `arduino-1.8.0-linux32.tar.xz`
-  `arduino-1.8.0-linux64.tar.xz`
-  `arduino-1.8.0-macosx.zip`
-  `arduino-1.8.0-windows.exe`
-  `arduino-1.8.0-windows.zip`

Instalando Arduino (Windows)

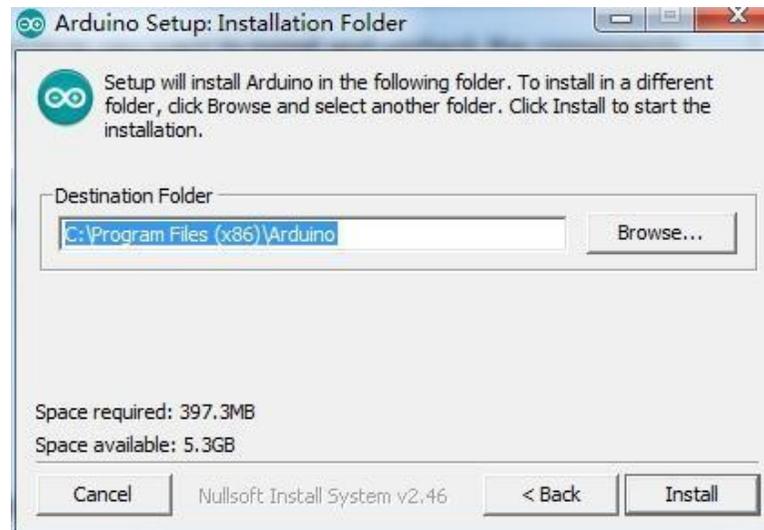
Instalando Arduino con el paquete de instalación exe.



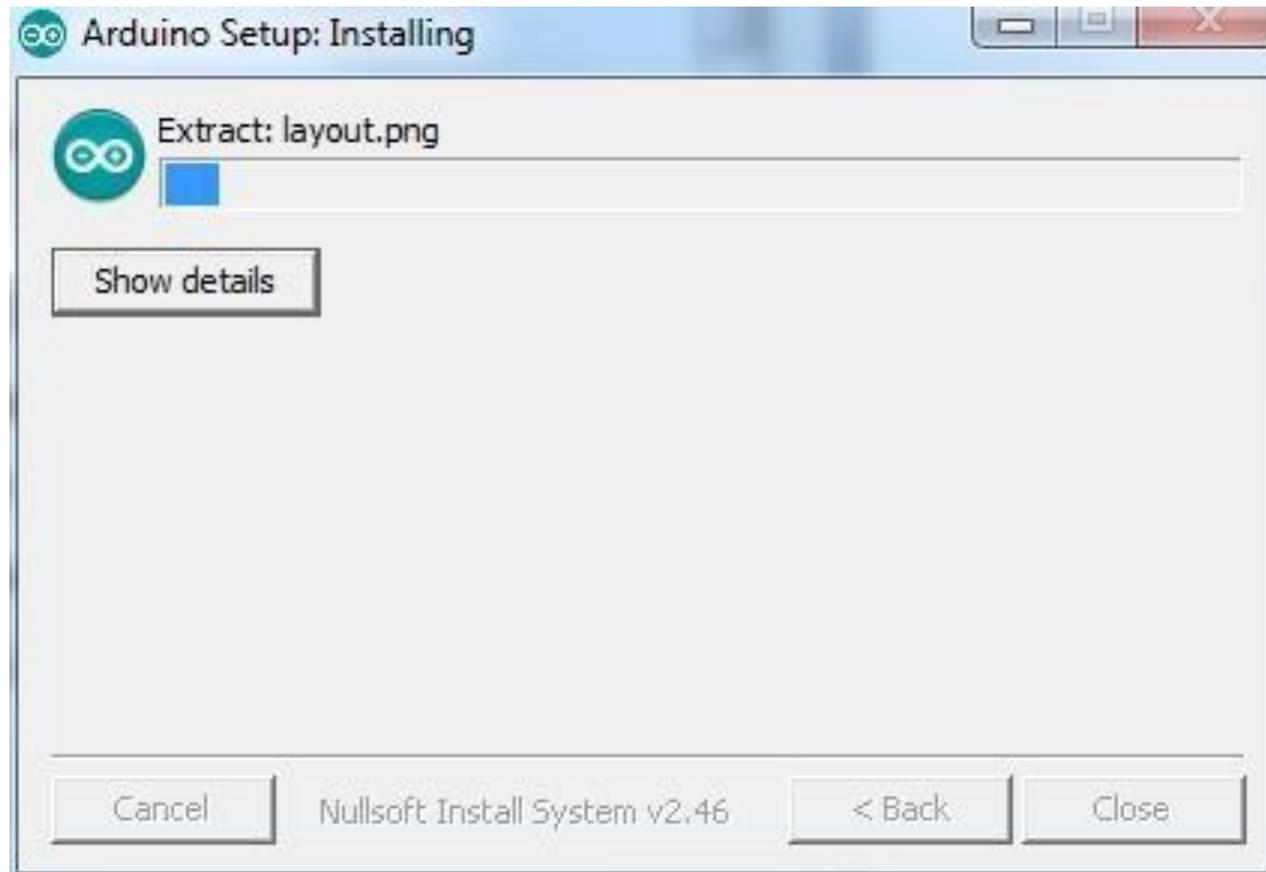
Click **I Agree** para ver la siguiente interfaz



Click Next



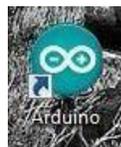
Puede presionar Examinar... para elegir una ruta de instalación o directamente escriba el directorio que desee. Haga clic en Instalar para iniciar la instalación



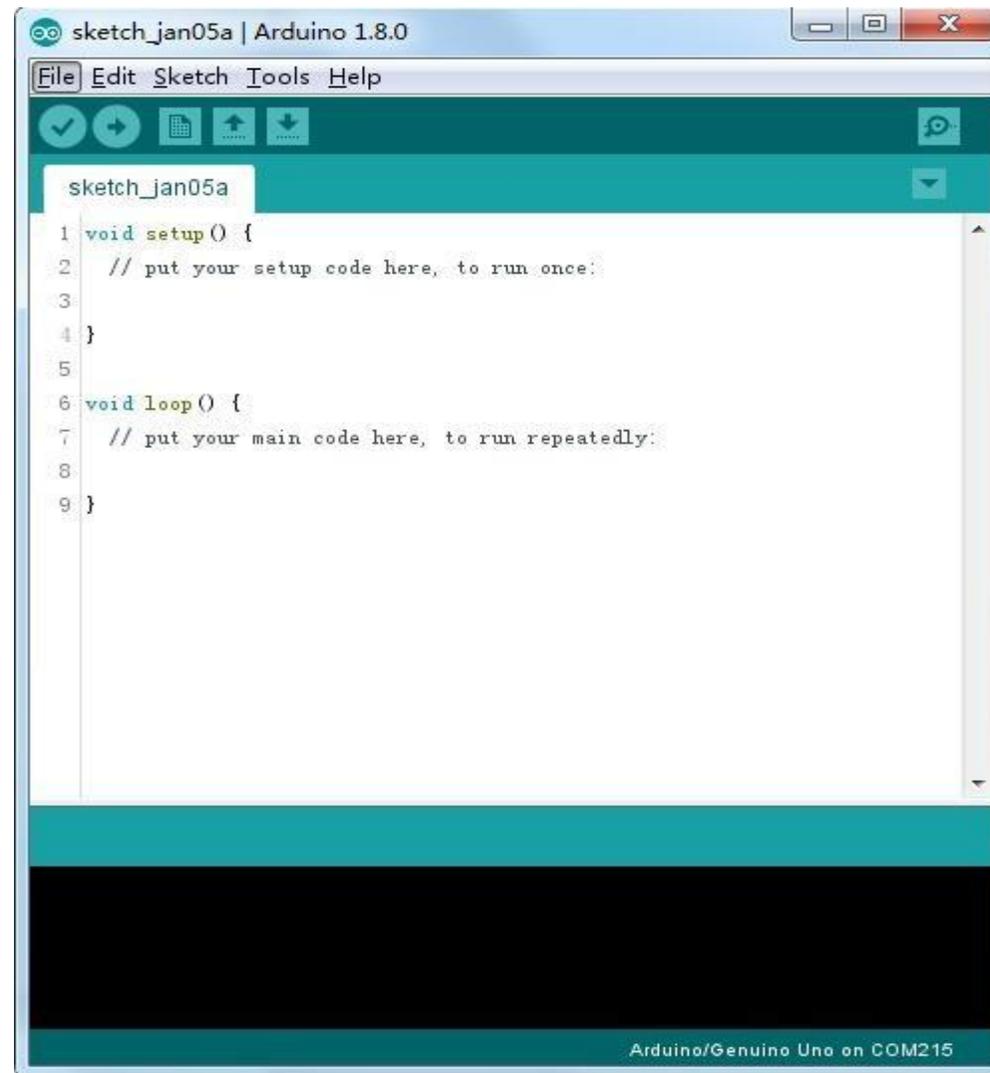
Espere el proceso de instalación, si aparece la interfaz de Seguridad de ventanas, simplemente continúe haciendo clic en Instalar para finalizar la instalación.



A continuación, aparece el siguiente icono en el escritorio.



Haga doble clic para entrar en el entorno de desarrollo deseado.



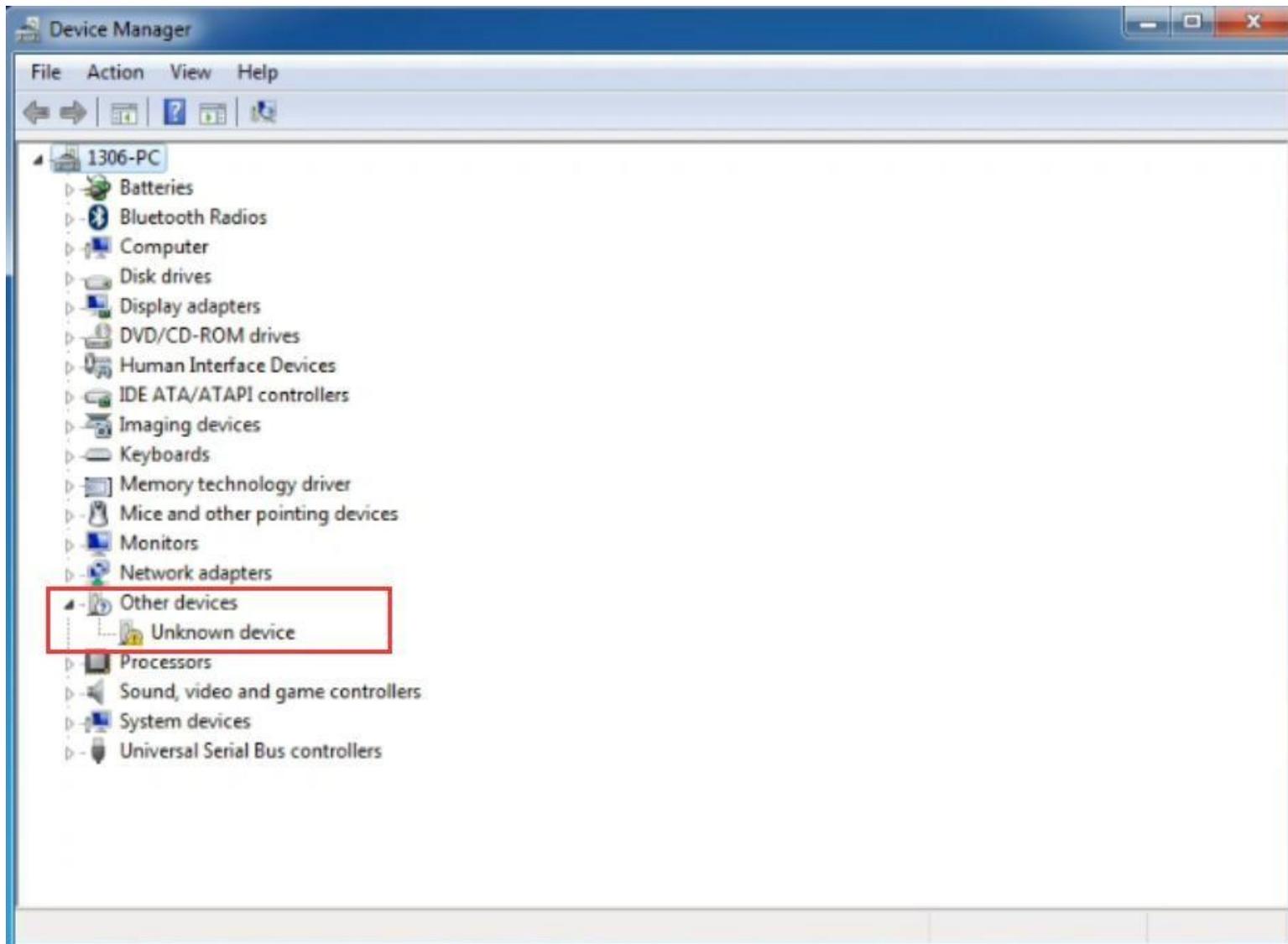
Cómo instalar el controlador de Arduino

A continuación, presentaremos la instalación del controlador de la placa de desarrollo UNO R3. La instalación del controlador puede tener ligeras diferencias en diferentes sistemas informáticos. Entonces, en lo siguiente, pasemos a la instalación del controlador en el sistema Windows.

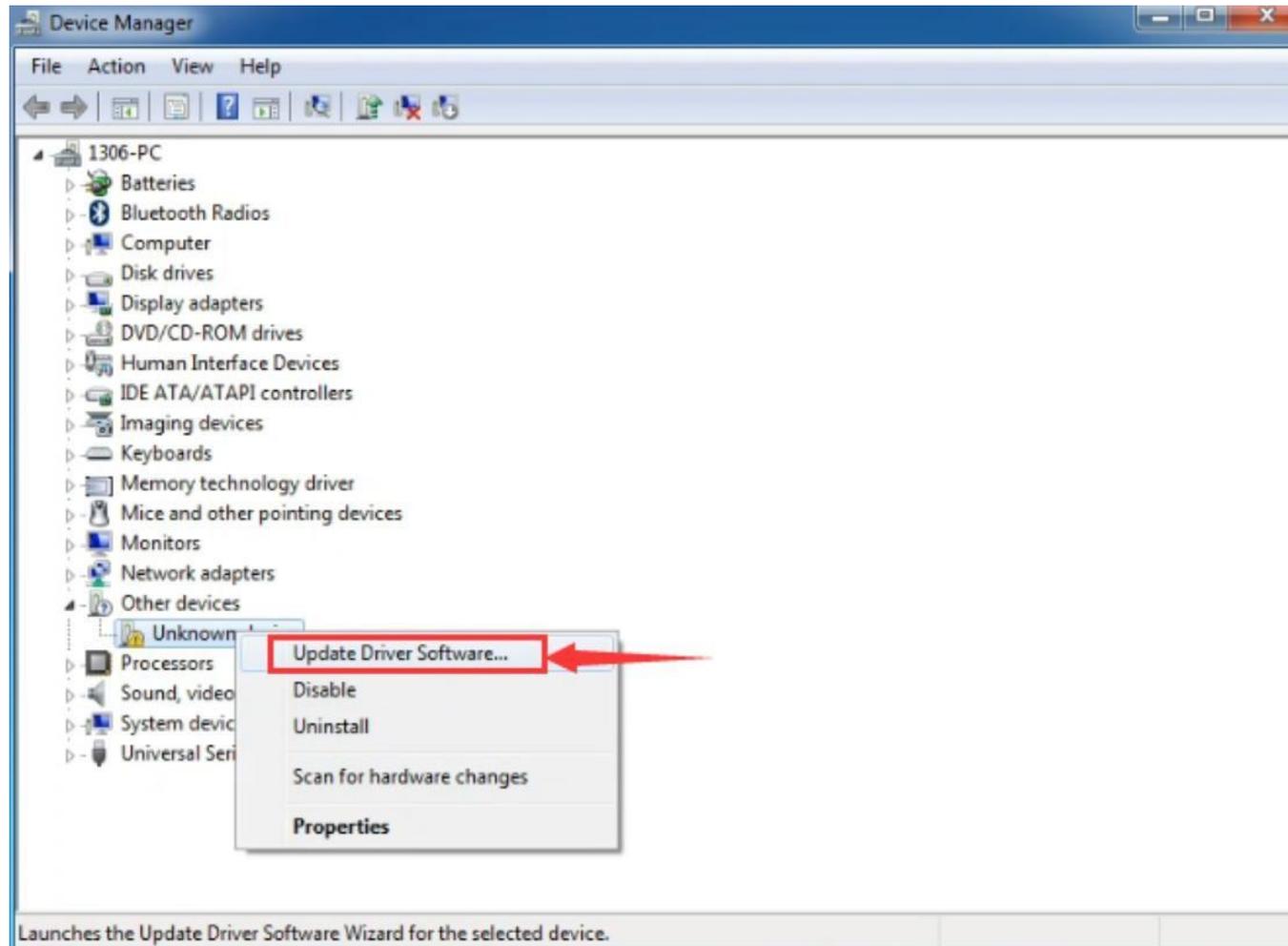
La carpeta Arduino contiene tanto el programa Arduino en sí como los controladores que permiten que el Arduino se conecte a su computadora mediante un cable USB. Antes de lanzar el software Arduino, va a instalar los controladores USB.

Cuando conectes la tarjeta UNO a tu computador la primera vez, haz click derecho en el ícono de: **“Computer”** —> **“Properties”**—> click en **“Device manager”**

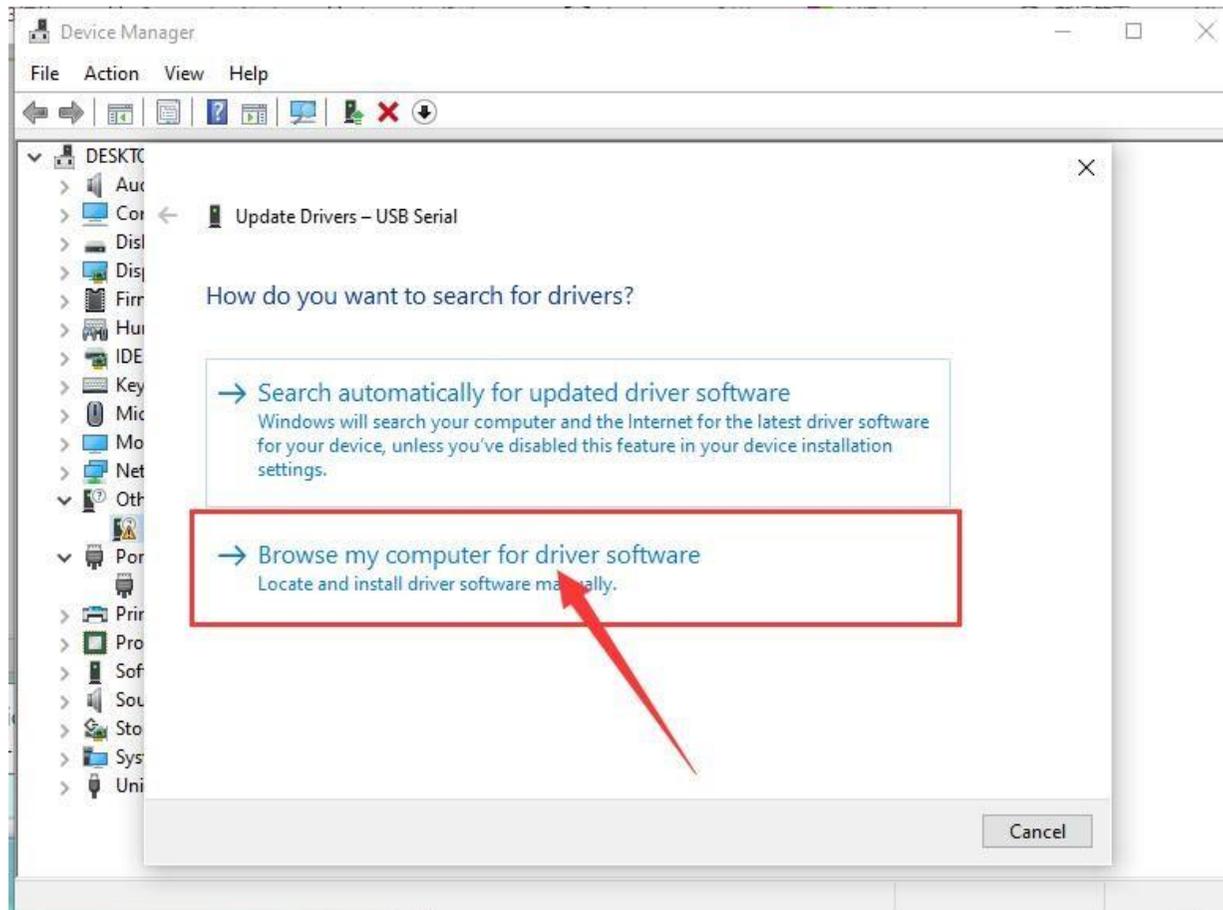
Por debajo **“Other Devices”** o **“USB-Serial”**, Debería ver un icono de **“Unknown device”** con un pequeño triángulo de advertencia amarillo al lado. Este es su Arduino. O puede buscar **"dispositivo"** en su computadora, o puede abrir el administrador de dispositivos de su computadora.



Luego haga clic derecho en el dispositivo y seleccione la opción de menú superior (Actualizar software de controlador...) que se muestra como la figura a continuación.

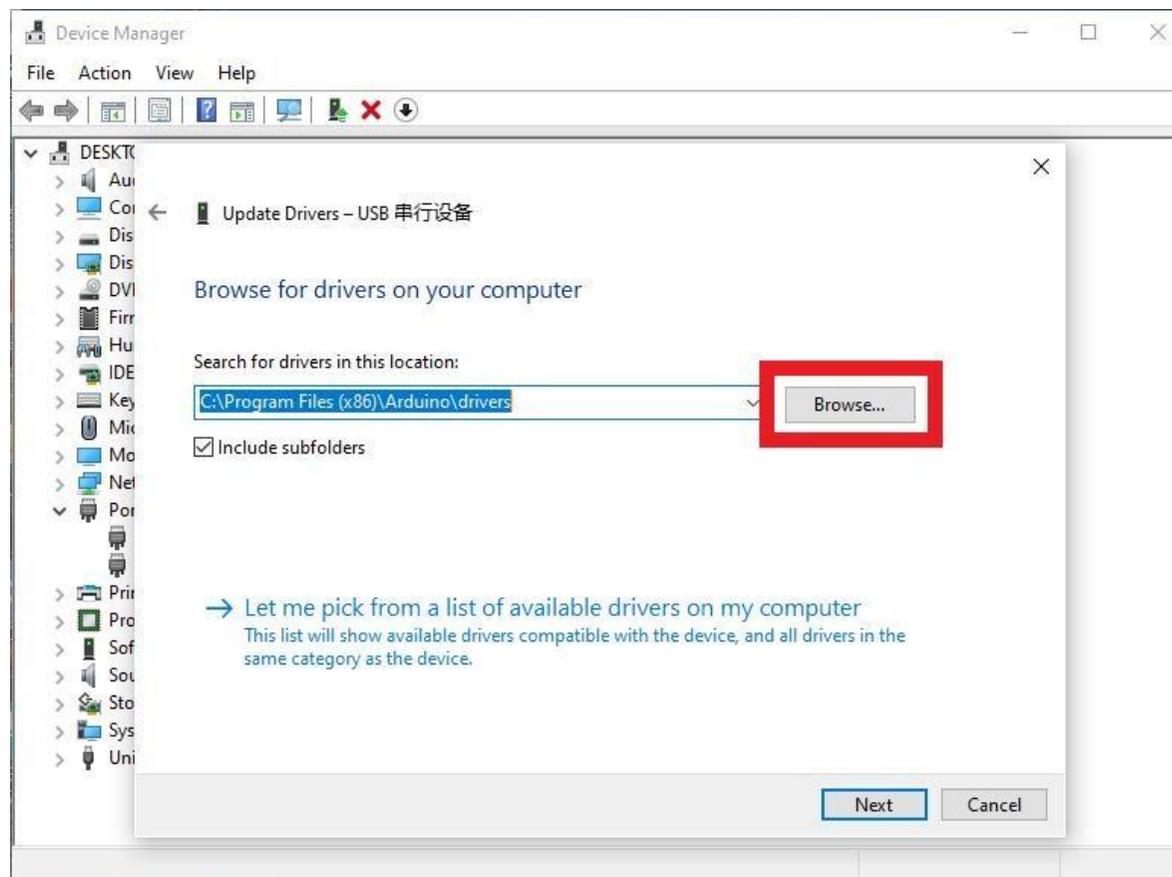


A continuación, se le pedirá que “Search Automatically for updated driver software” o “Browse my computer for driver software”. Como se muestra abajo. En esta página, seleccione **“Browse my computer for driver software”**.



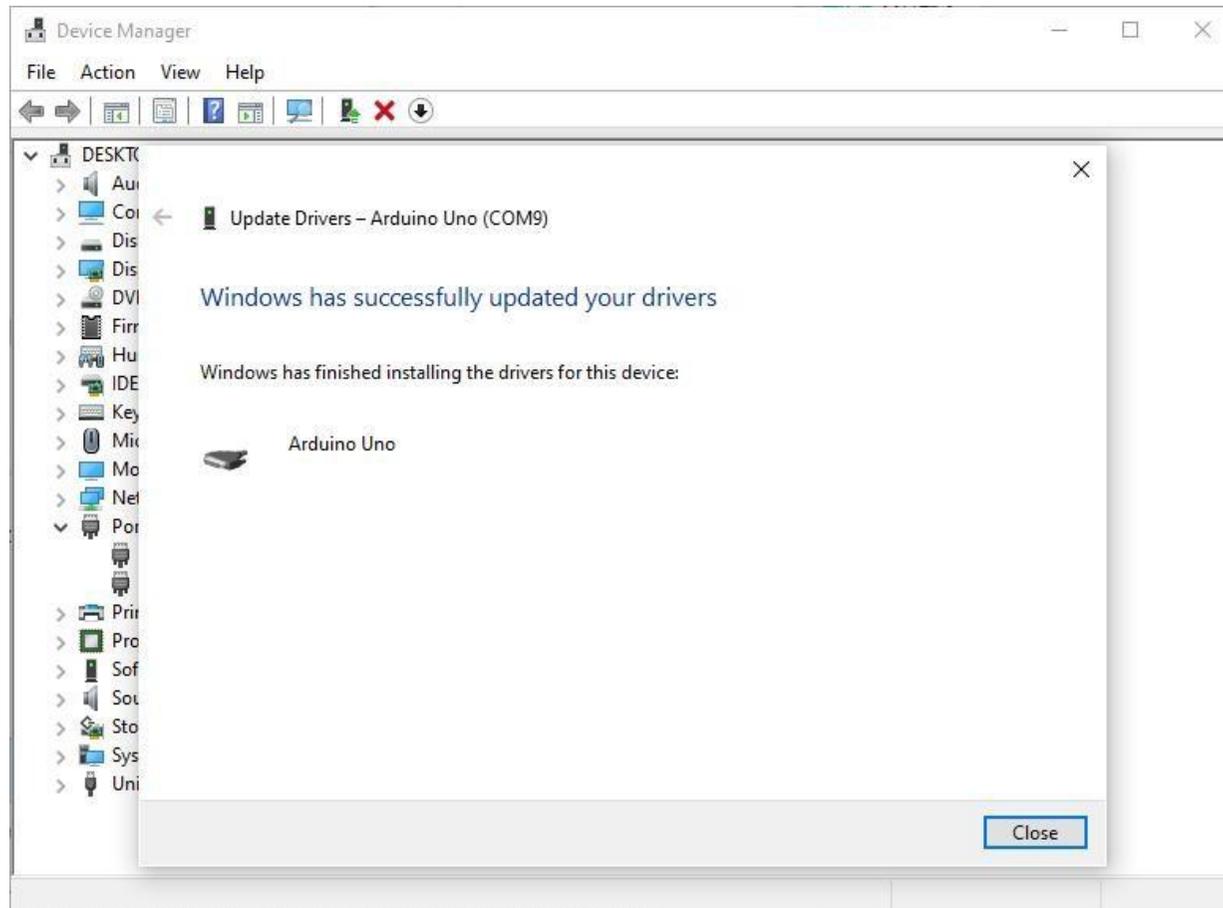
Haga clic derecho en el dispositivo y seleccione la opción de menú superior (Actualizar software de controlador ...).

A continuación, se le pedirá que: ‘Search Automatically for updated driver software’ o ‘Browse my computer for driver software’. Seleccione la opción para navegar y navegar hasta el: **C:\Program Files(x86)\Arduino\drivers**. **(Nota: Aquí está la ruta que elige para instalar el IDE de arduino. La ruta elegida en el tutorial de instalación en la sección anterior es esa, por lo que la ruta que elegí es: C:\Program Files(x86)\Arduino\drivers)**

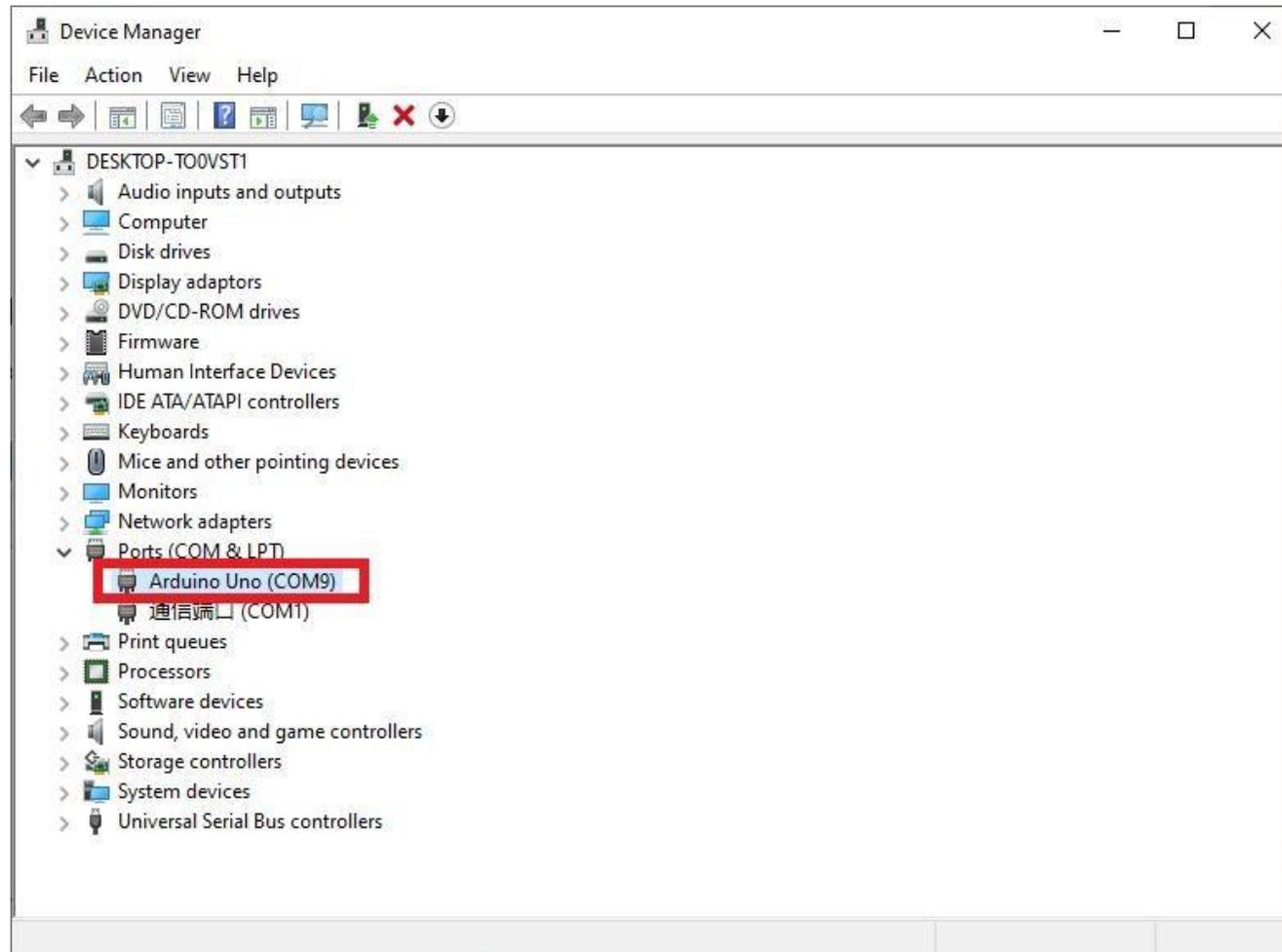


Click **“Next”** y puede recibir una advertencia de seguridad, si es así, permita que se instale el software.

Una vez que se haya instalado el software, recibirá un mensaje de confirmación. Instalación completada, haga clic en **“Close”**.



Hasta ahora, el controlador está bien instalado. Luego puedes hacer clic derecho “**Computer**”—> “**Properties**”—> “**Device manager**”, Debería ver el dispositivo como la figura que se muestra a continuación.



Como Agregar Librerías

Instalación de librerías adicionales de Arduino

Una vez que se sienta cómodo con el software Arduino y utilizando las funciones integradas, es posible que desee ampliar la capacidad de su Arduino con bibliotecas adicionales.

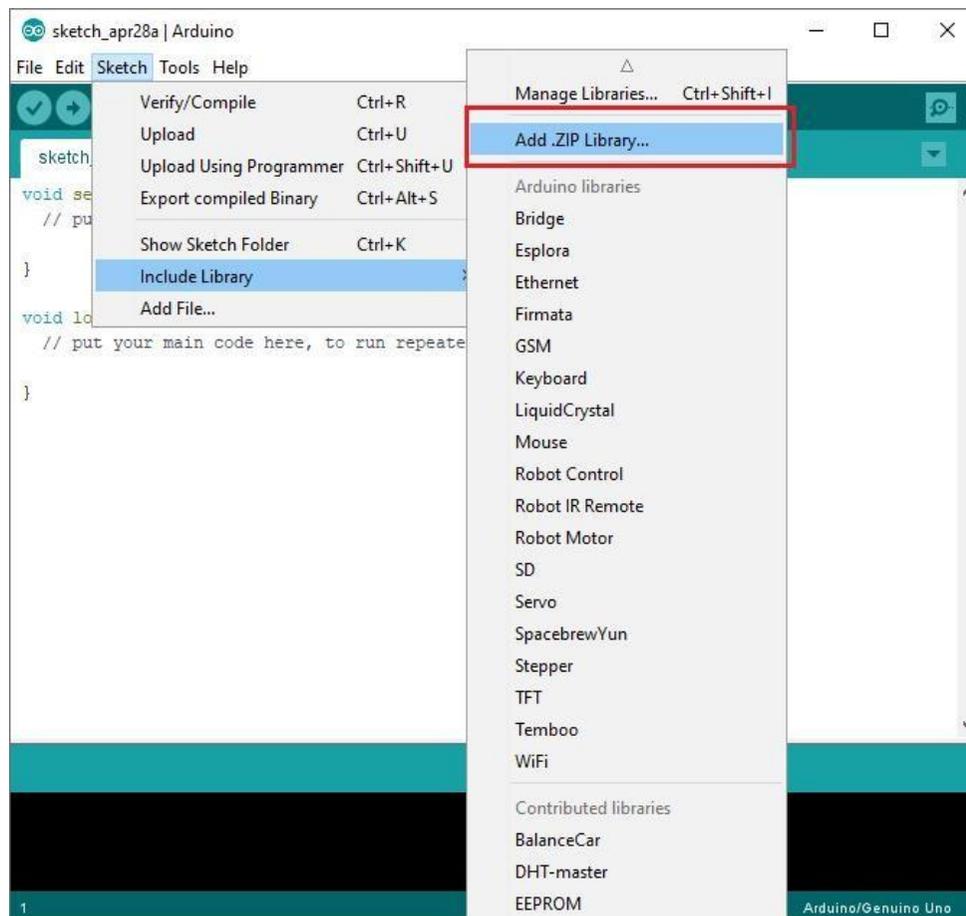
Que son las Librerías o Bibliotecas?

Las Bibliotecas son una colección de códigos que facilita la conexión a un sensor, pantalla, módulo, etc. Por ejemplo, la biblioteca incorporada de LiquidCrystal facilita la conversación con pantallas LCD de caracteres. Hay cientos de bibliotecas adicionales disponibles en Internet para descargar. Las bibliotecas integradas y algunas de estas bibliotecas adicionales se enumeran en la referencia. Para usar las bibliotecas adicionales, deberá instalarlas.

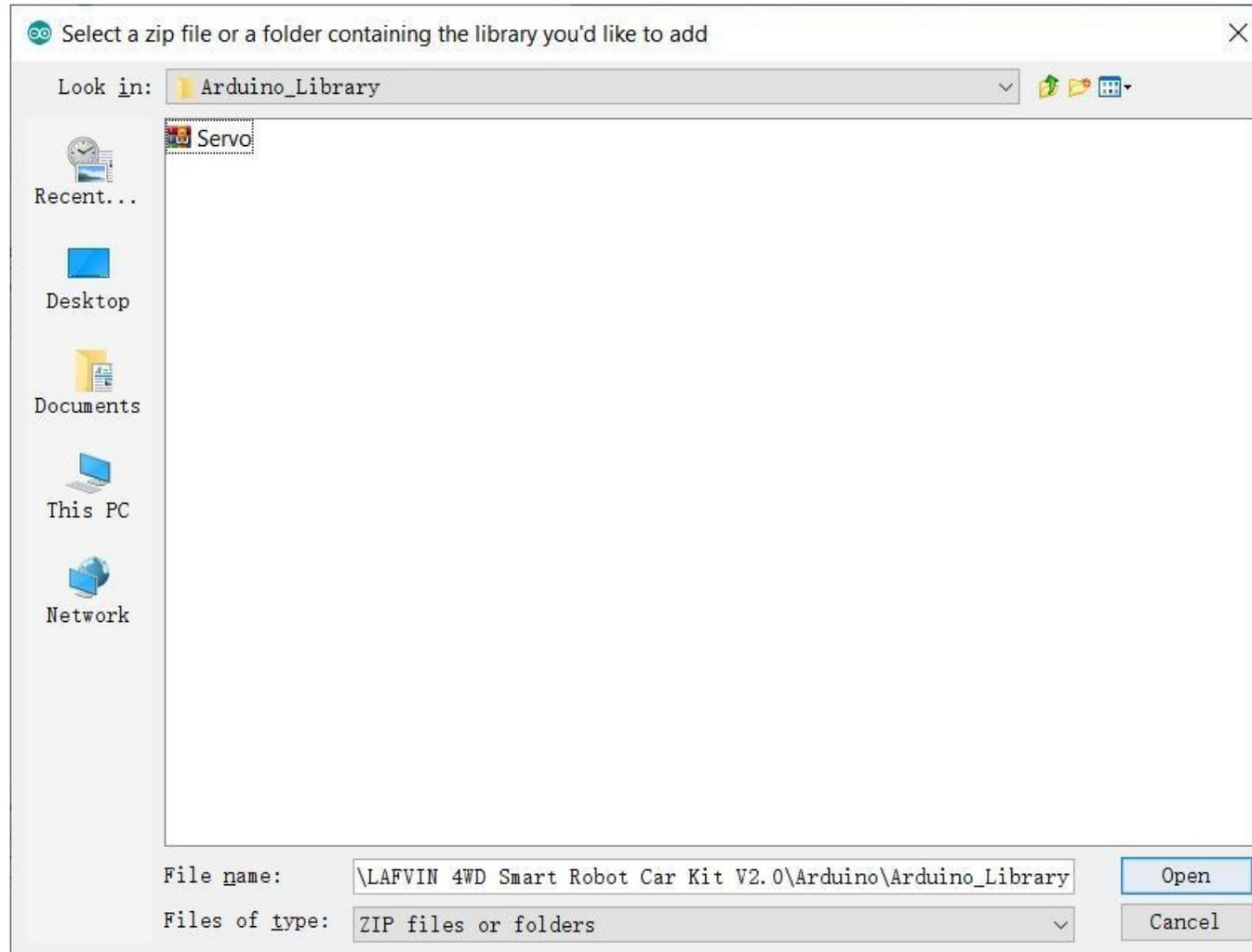
Como Instalar una biblioteca?

Importando una Librería .zip

Las bibliotecas a menudo se distribuyen como un archivo o carpeta ZIP. El nombre de la carpeta es el nombre de la biblioteca. Dentro de la carpeta habrá un archivo .cpp, un archivo .h y, a menudo, un archivo de palabras clave.txt un archivo de ejemplos, una carpeta de ejemplos y otros archivos requeridos por la biblioteca. puede instalar bibliotecas de 3rd party en el IDE. No descomprima la biblioteca descargada, déjela como está. En el Arduino IDE, navegar a **Sketch > Include Library**. En la parte superior de la lista desplegable, seleccione la opción de **"Add .ZIP Library"**.



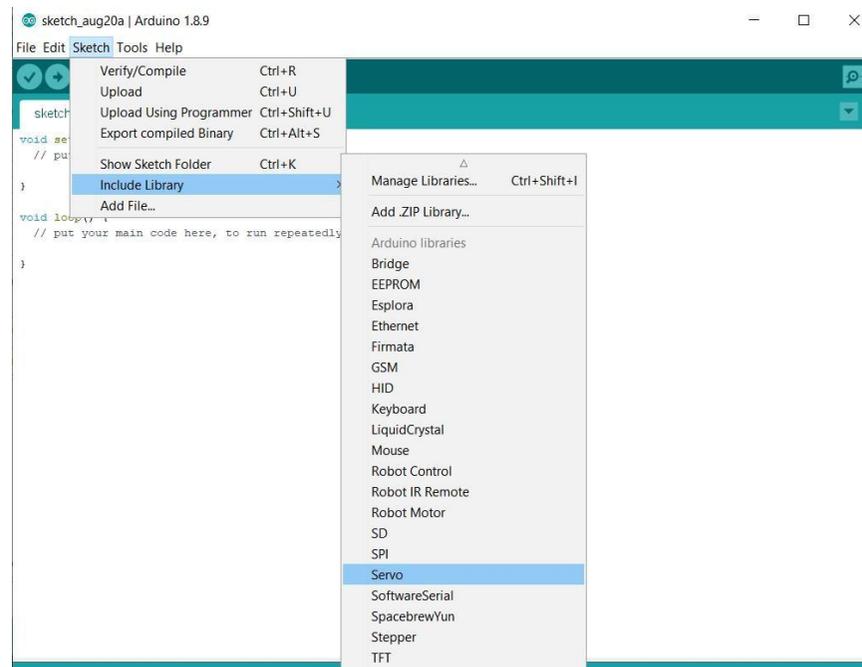
Se le pedirá que seleccione la biblioteca que desea agregar. Navegue hasta la ubicación del archivo .zip y ábralo.



(Nota: Esto es solo para demostrar cómo agregar un archivo de biblioteca zip, si necesita agregar un archivo de biblioteca depende de las necesidades reales de su programa)

Regresar al menú **Sketch > Import Library**. Ahora debería ver la biblioteca en la parte inferior del menú desplegable. Está listo para ser utilizado en su boceto (Sketch). El archivo zip se habrá expandido en la carpeta de bibliotecas en su directorio de bocetos de Arduino.

NB: La Biblioteca estará disponible para su uso en bocetos, pero los ejemplos de la biblioteca no se expondrán en el **File > Examples** después que el IDE se encuentre reiniciado.

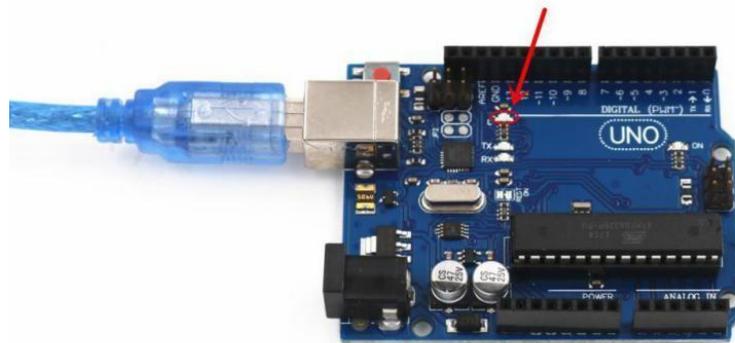


Esos dos son los enfoques más comunes. Los sistemas MAC y Linux se pueden manejar de la misma manera. La instalación manual que se presentará a continuación como alternativa puede ser raramente utilizada y los usuarios sin necesidad pueden omitirla.

Prueba de parpadeo (prueba tu primer programa)

Aprenderá cómo programar su placa controladora UNO para parpadear el LED incorporado del Arduino y cómo descargar programas mediante pasos básicos. La placa UNO tiene filas de conectores a lo largo de ambos lados que se utilizan para conectarse a varios dispositivos electrónicos y 'escudos' (Shield) enchufables que amplían su capacidad.

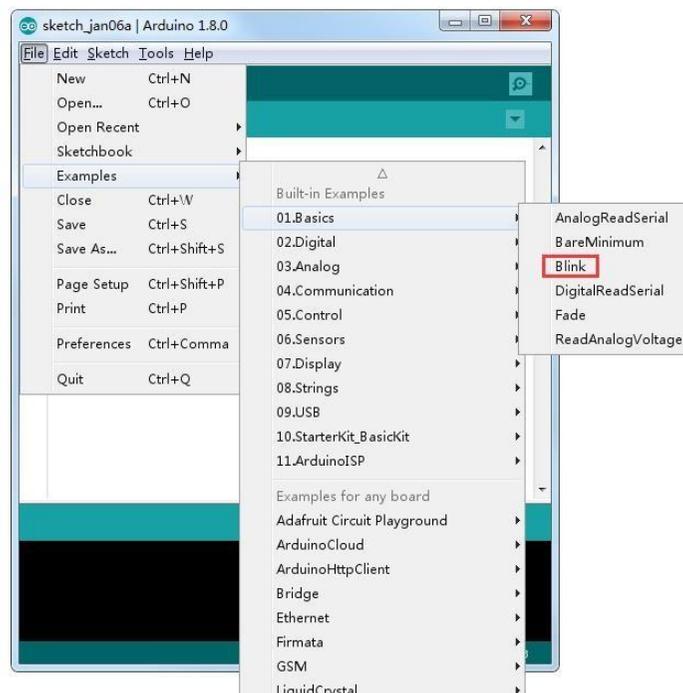
También tiene un solo LED que puedes controlar desde tus bocetos (Sketch). Este LED está integrado en la placa UNO y a menudo se conoce como el LED 'L', ya que así es como se etiqueta en la placa.



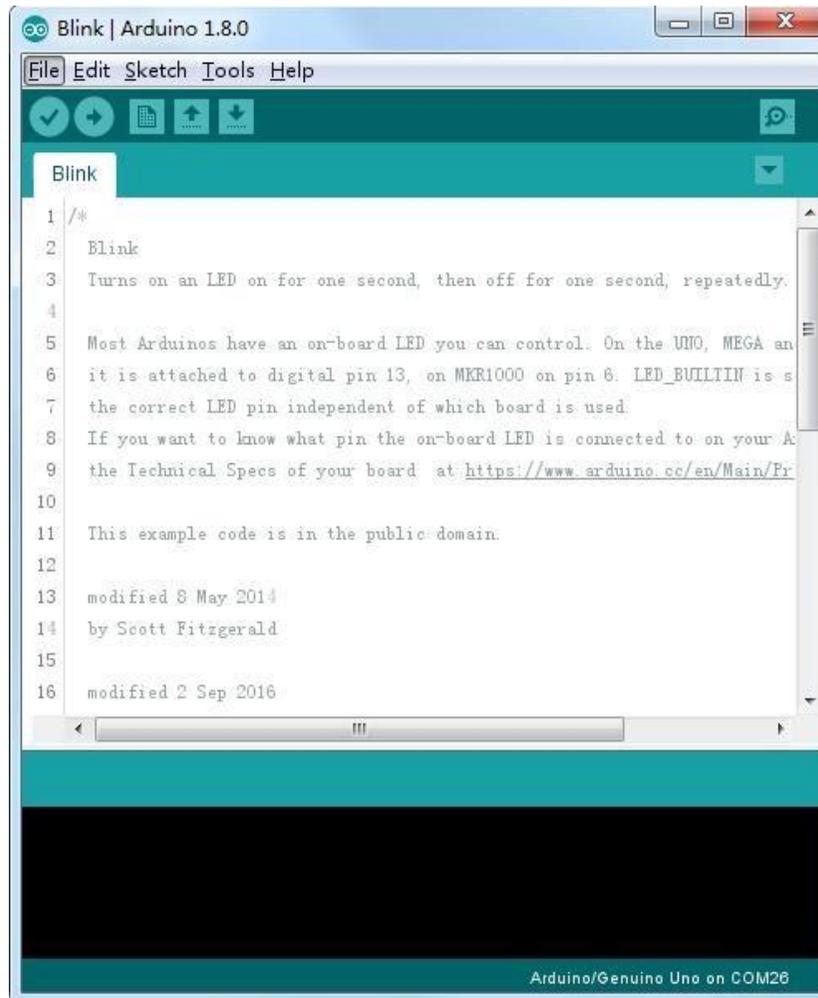
Es posible que el LED 'L' de su placa UNO ya parpadee cuando lo conecta a un enchufe USB. Esto se debe a que las placas generalmente se envían con el boceto 'Blink' preinstalado. En este proyecto, reprogramaremos el tablero UNO con nuestro propio boceto de parpadeo y luego cambiaremos la velocidad a la que parpadea. En el capítulo anterior-Cómo instalar el IDE de Arduino, configuró su IDE de Arduino y se aseguró de poder encontrar el puerto serie adecuado para que se conecte a su placa UNO. Ha llegado el momento de poner a prueba esa conexión y programar su placa UNO.

El IDE de Arduino incluye una gran colección de bocetos de ejemplo que puede cargar y usar. Esto incluye un boceto de ejemplo para hacer que el LED 'L' parpadee.

Cargue el boceto 'Parpadear' que encontrará en el sistema de menús del IDE en **File > Examples > 01.Basics**.



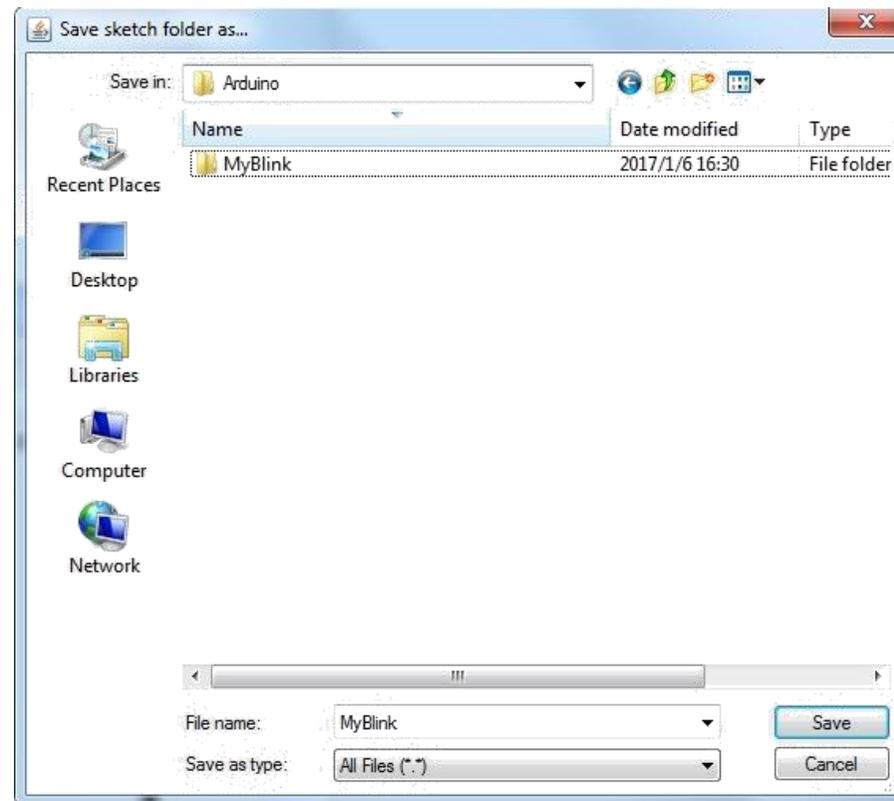
Cuando se abra la ventana de boceto, ampliela para que pueda ver todo el boceto en la ventana.



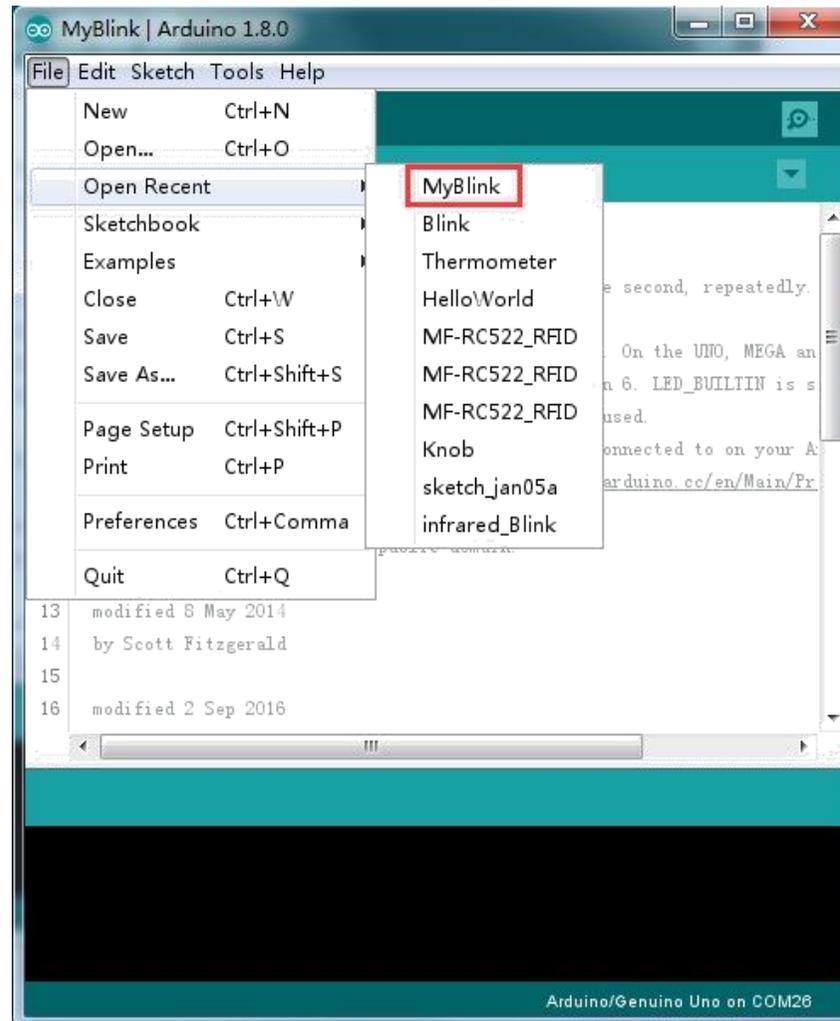
Los bocetos de ejemplo incluidos con el IDE de Arduino son de "solo lectura". Es decir, puede subirlos a una placa UNO R3, pero si los cambia, no puede guardarlos como el mismo archivo.

Ya que vamos a cambiar este boceto, lo primero que debe hacer es guardar su propia copia.

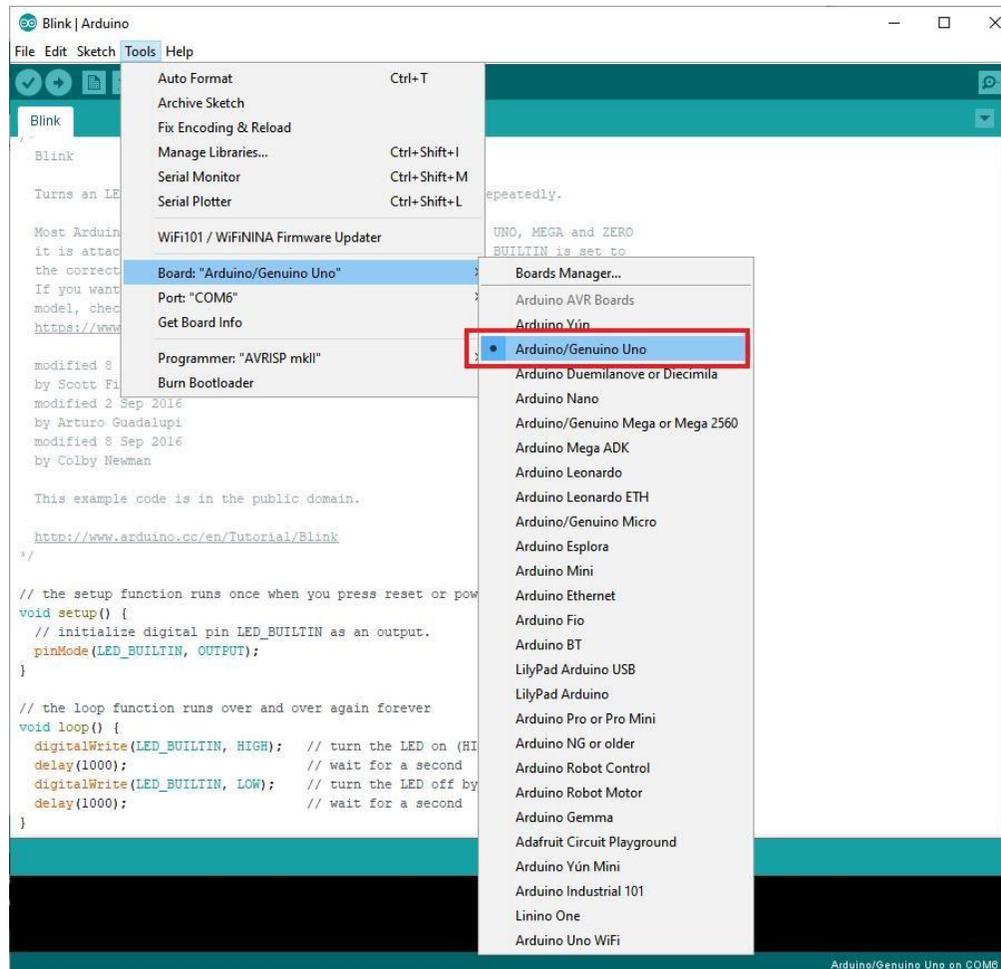
Desde el menú Archivo en el IDE de Arduino, seleccione 'Guardar como' y luego guarde el boceto con el nombre 'MyBlink'.

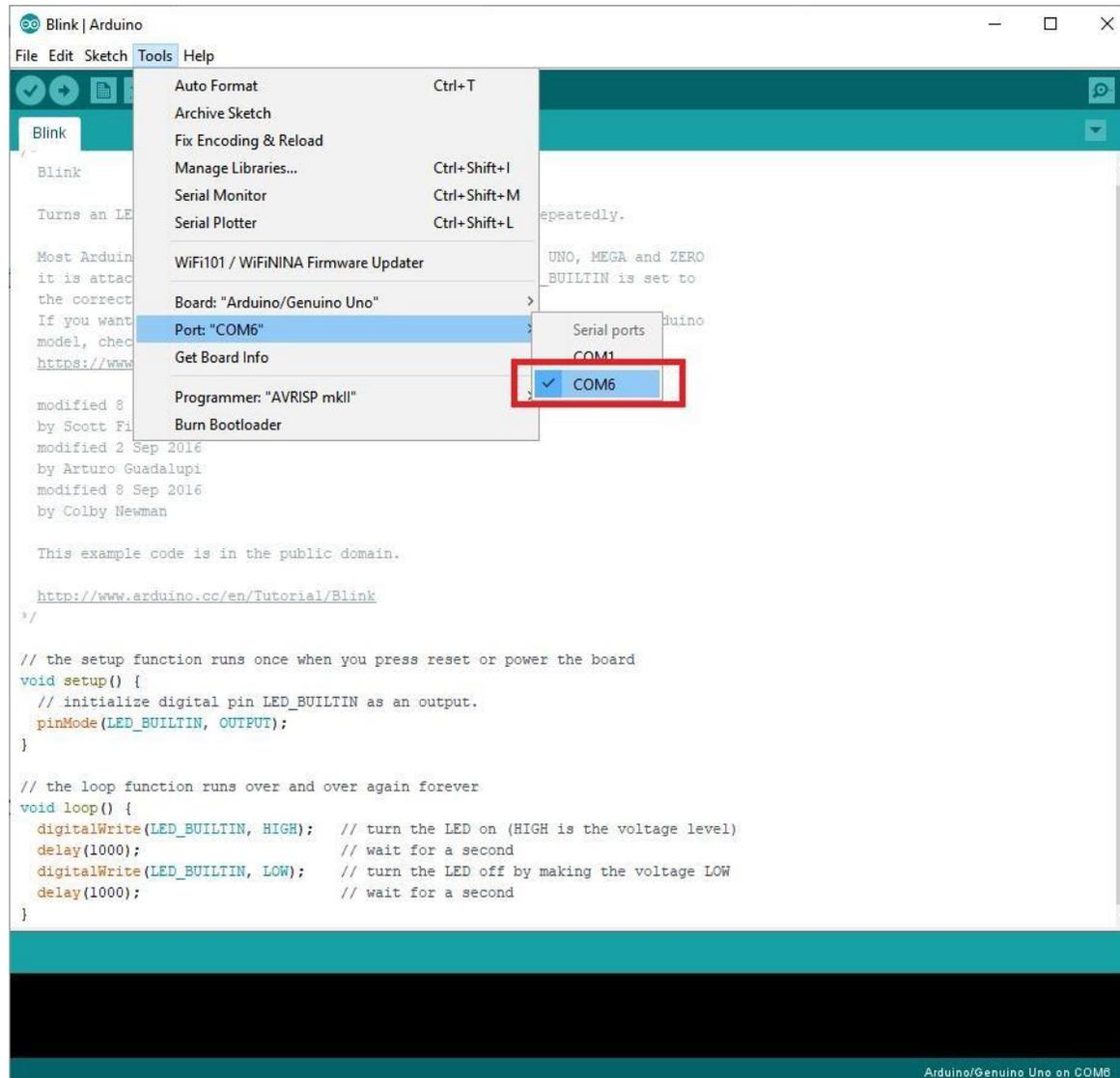


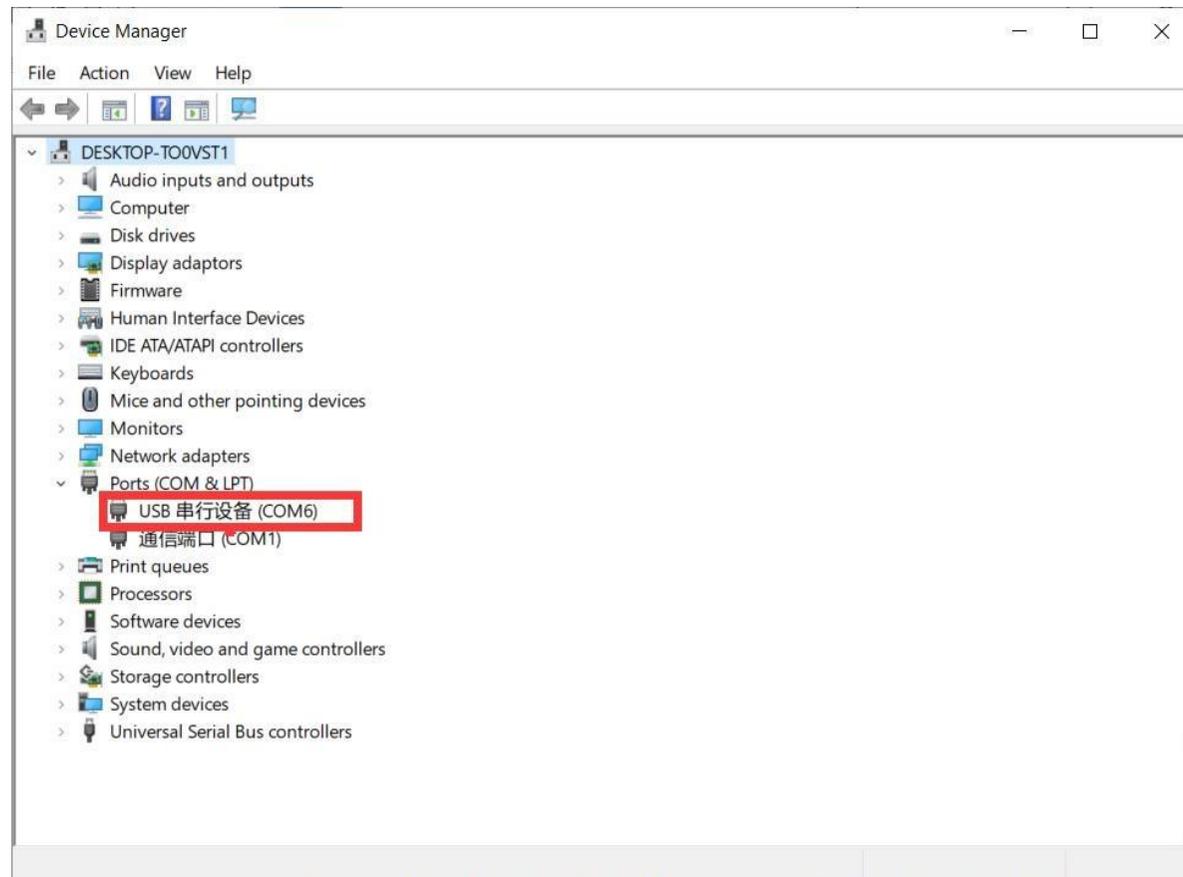
Has guardado tu copia de 'Blink' en tu cuaderno de bocetos. Esto significa que, si alguna vez desea encontrarlo de nuevo, puede abrirlo usando la opción de menú **Archivo > Cuaderno de bocetos**.



Conecte su placa Arduino a su computadora con el cable USB y verifique que el 'Tipo de placa' y 'Puerto serie' estén configurados correctamente.



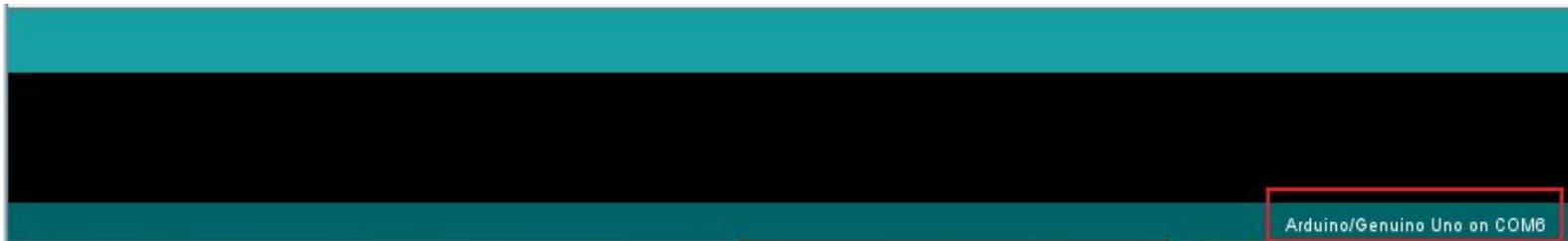




Nota: El tipo de placa y el puerto serie aquí no son necesariamente los mismos que se muestran en la imagen. Si está utilizando UNO, deberá elegir Arduino UNO como el tipo de placa, se pueden hacer otras elecciones de la misma manera.

Y el puerto serie que se muestra para todos es diferente, a pesar de que COM 6 elegido aquí, podría ser COM3 o COM4 en su computadora. Se supone que un puerto COM correcto es COMX (arduino XXX), que es según los criterios de certificación.

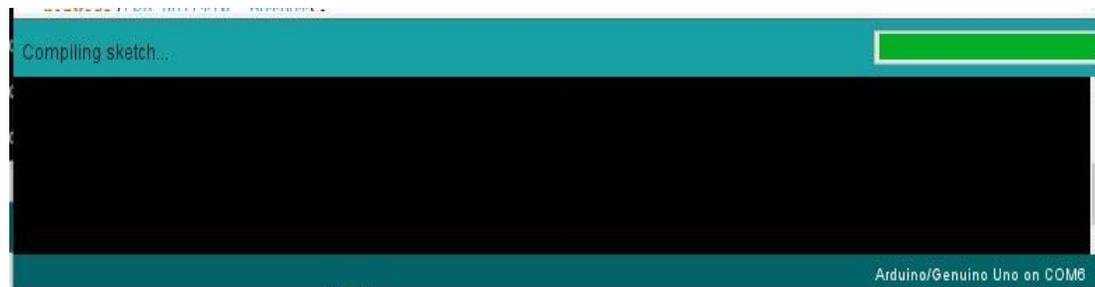
El IDE de Arduino le mostrará la configuración actual de la placa en la parte inferior de la ventana.



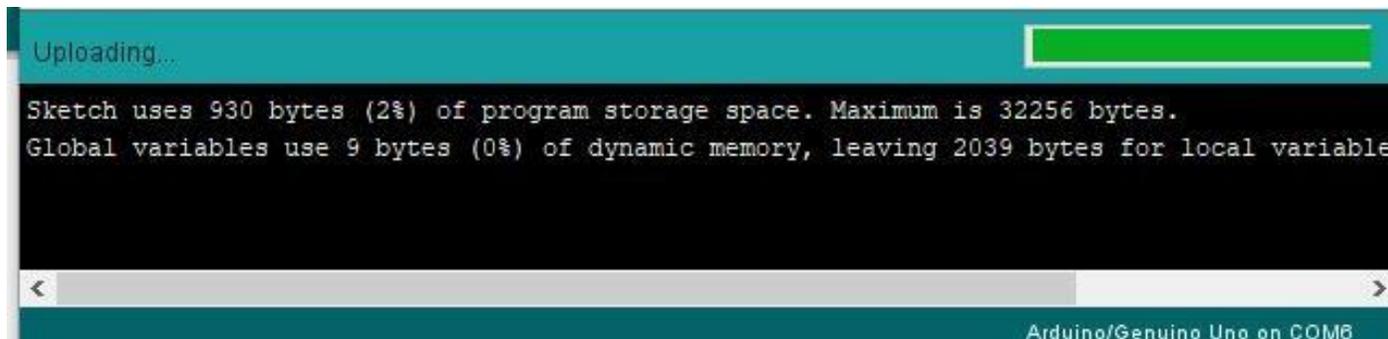
Click en el botón 'Upload'. El segundo botón de la izquierda en la barra de herramientas.



Si observa el área de estado del IDE, verá una barra de progreso y una serie de mensajes. Al principio, dirá 'Compilando boceto ...'. Esto convierte el boceto en un formato adecuado para cargar en el tablero.

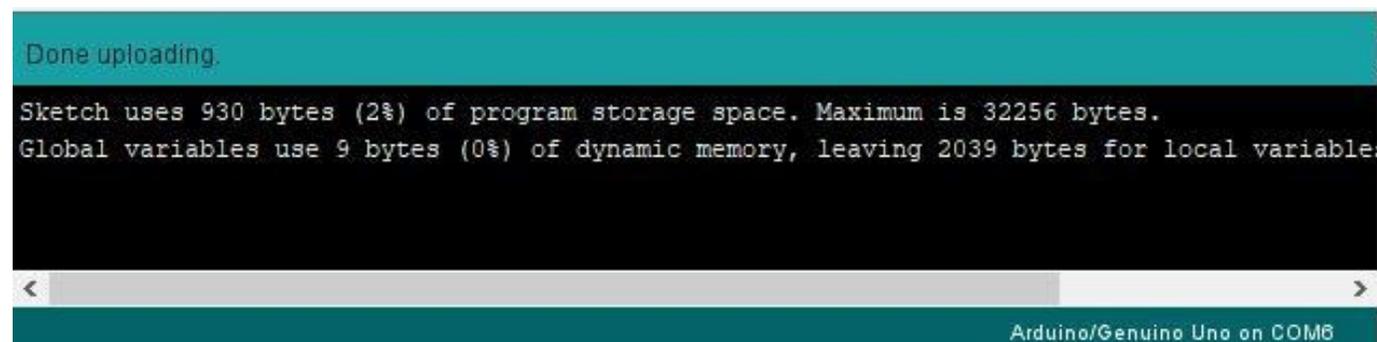


A continuación, el estado cambiará a '**Cargando**'. En este punto, los LED en el Arduino deberían comenzar a parpadear a medida que se transfiere el boceto.



```
Uploading...  
Sketch uses 930 bytes (2%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.  
< >  
Arduino/Genuino Uno on COM8
```

Finalmente, el status cambiará a **'Done'**.



```
Done uploading.  
Sketch uses 930 bytes (2%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.  
< >  
Arduino/Genuino Uno on COM8
```

El otro mensaje nos dice que el boceto está usando 928 bytes de los 32.256 bytes disponibles. Después de la etapa **'Compilando boceto'** podría obtener el siguiente mensaje de error:

```
Problem uploading to board. See http://www.arduino.cc/en/... Copy error messages
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x22
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting
Arduino/Genuino Uno on COM1
```

Puede significar que su placa no está conectada en absoluto, o que los controladores no se han instalado (si es necesario) o que se ha seleccionado el puerto serie incorrecto. Si encuentra esto, vuelva a Project 0 y compruebe su instalación. Una vez que se haya completado la carga, la placa debe reiniciarse y comenzar a parpadear.

Tenga en cuenta que una gran parte de este boceto se compone de comentarios. Estas no son instrucciones reales del programa; más bien, solo explican cómo funciona el programa. Están ahí para su beneficio.

Todo lo que se encuentre entre `/*` y `*/` En la parte superior del boceto hay un comentario en bloque; Explica para qué sirve el boceto.

Los comentarios de una sola línea comienzan con `//` y todo hasta el final de esa línea se considera un comentario.

La primera línea de código es: `int led = 13;`

Como explica el comentario anterior, esto le está dando un nombre al pin al que está conectado el LED. Esto es 13 en la mayoría de los Arduinos, incluyendo la UNO y LEONARDO.

A continuación, tenemos la función '**setup**'. Una vez más, como dice el comentario, esto se ejecuta cuando se presiona el botón de reinicio. También se ejecuta cada vez que la placa se reinicia por cualquier motivo, como la primera vez que se le aplica potencia, o después de que se haya cargado un boceto.

```
void setup() {
```

```
// Inicializar el pin digital como salida. pinMode(led, OUTPUT);
```

```
}
```

Cada boceto de Arduino debe tener una función de '**configuración**', y el lugar donde es posible que desee agregar sus propias instrucciones es entre paréntesis { }.

En este caso, solo hay un comando allí, que, como dice el comentario, le dice a la placa Arduino que vamos a usar el pin LED como salida.

También es obligatorio que un boceto tenga una función de '**bucle**'. A diferencia de la función '**setup**' que solo se ejecuta una vez, después de un reinicio, la función '**loop**', después de que haya terminado de ejecutar sus comandos, se iniciará inmediatamente de nuevo.

```
void loop()
```

```
{ digitalWrite(led, HIGH); delay(1000);
```

```
digitalWrite(led, LOW); delay(1000);
```

```
}
```

Dentro de la función de bucle, los comandos primero encienden el pin LED (HIGH), luego '**delay**' durante 1000 milisegundos (1 segundo), luego apagan el pin LED y hacen una pausa por otro segundo.

Ahora vas a hacer que tu LED parpadee más rápido. Como habrás adivinado, la clave para esto radica en cambiar el parámetro en () para el comando '**delay**'.

// turn the LED off (LOW is the voltage level) // Esperar por un segundo
// turn the LED on (HIGH is the voltage level) // Esperar por un segundo

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the volt
33   delay(500) // wait for a second
34   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the vo
35   delay(500) // wait for a second
36 }
```

Este período de retraso es en milisegundos, por lo que, si desea que el LED parpadee dos veces más rápido, cambie el valor de 1000 a 500. Esto se detendría durante medio segundo cada retraso en lugar de un segundo entero. Cargue el boceto nuevamente y debería ver que el LED comienza a parpadear más rápido.

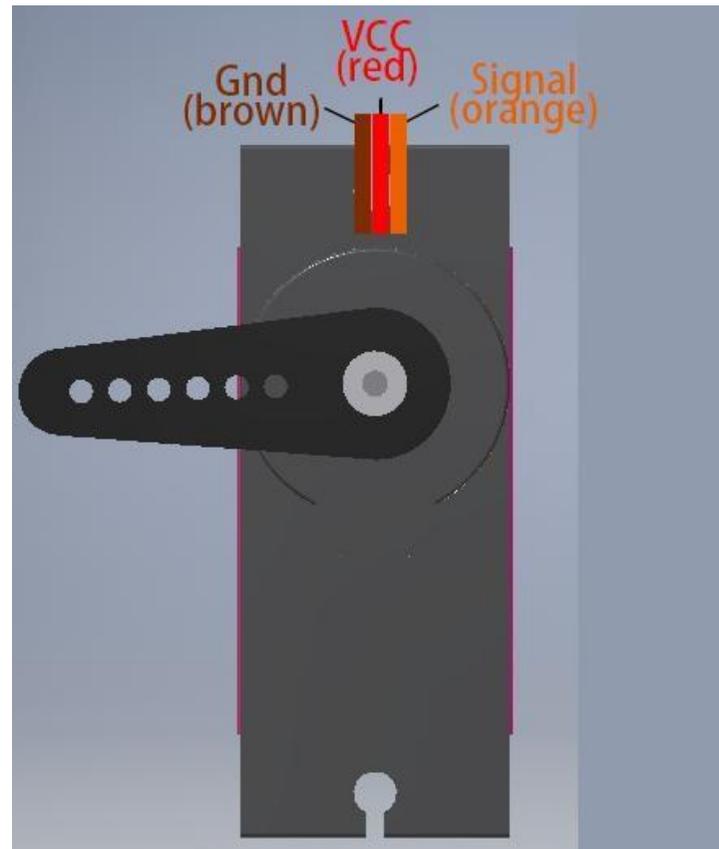
Lección 2 Servocontrol y calibración del ángulo de instalación

Resumen

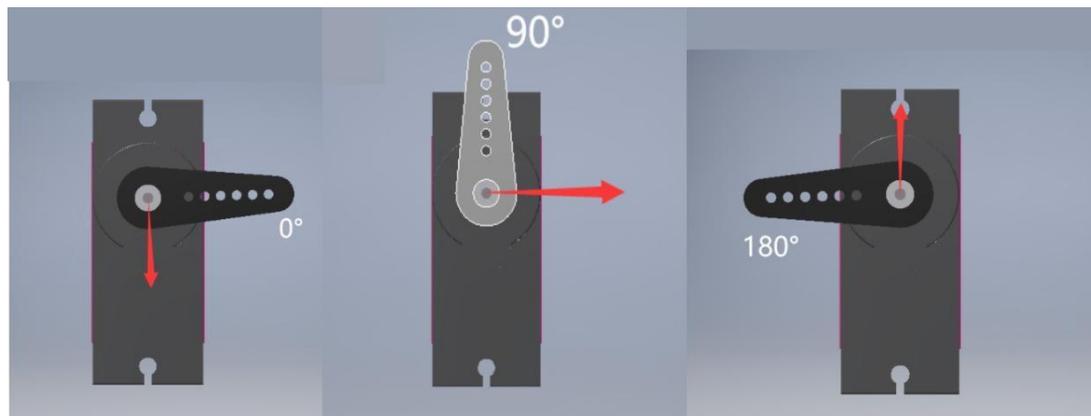
En esta lección, aprendemos cómo controlar la rotación del servo. Cada articulación del brazo robótico está controlada por un servo.

Que es un Servo

El servomotor es un actuador rotativo de control de posición. Consiste principalmente en una carcasa, placa de circuito, motor sin núcleo, engranaje y sensor de posición. Su principio de funcionamiento es que el servo recibe la señal enviada por MCU o receptores y produce una señal de referencia con un período de 20ms y un ancho de 1.5ms, luego compara el voltaje de polarización de CC adquirido con el voltaje del potenciómetro y obtiene la salida de diferencia de voltaje. Cuando la velocidad del motor es constante, el potenciómetro se acciona para girar a través del engranaje de reducción en cascada, lo que lleva a que la diferencia de voltaje sea 0 y el motor deje de girar. Generalmente, el rango de ángulo de rotación del servo es de 0° -- 180° . El ángulo de rotación del servomotor se controla regulando el ciclo de trabajo de la señal PWM (modulación de ancho de pulso). El ciclo estándar de la señal PWM es de 20 ms (50Hz). Teóricamente, el ancho se distribuye entre 1ms-2ms, pero de hecho, está entre 0.5ms-2.5ms. El ancho corresponde al ángulo de rotación de 0° a 180° . Pero tenga en cuenta que para motores de diferentes marcas, la misma señal puede tener un ángulo de rotación diferente.



En general, el servo tiene tres líneas en marrón, rojo y naranja. El cable marrón está conectado a tierra, el rojo es una línea de polo positivo y el naranja es una línea de señal.



Los ángulos servo correspondientes se muestran a continuación:

High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

Especificaciones:

Voltaje de trabajo: DC 4.8V ~ 6V

Rango de ángulo de operación: about 180 ° (at 500 → 2500 μseg)

Rango de ancho de pulso: 500 → 2500 μseg

Sin velocidad de carga: 0.12 ± 0.01 seg / 60 (DC 4.8V) 0.1 ± 0.01 seg / 60 (DC 6V)

Corriente sin carga: 200 ± 20 mA (DC 4.8V) 220 ± 20 mA (DC 6V)

Par de parada: 1.3 ± 0.01 kg · cm (DC 4.8V) 1.5 ± 0.1 kg · cm (DC 6V)

Corriente de Parada: $\cong 850$ mA (DC 4.8V) $\cong 1000$ mA (DC 6V)

Corriente de espera: 3 ± 1 mA (DC 4.8V) 4 ± 1 mA (DC 6V)

Nota:

Si ha instalado el brazo robótico y ha encontrado que el servo genera calor durante el uso, significa que el ángulo inicial del servo no está calibrado.

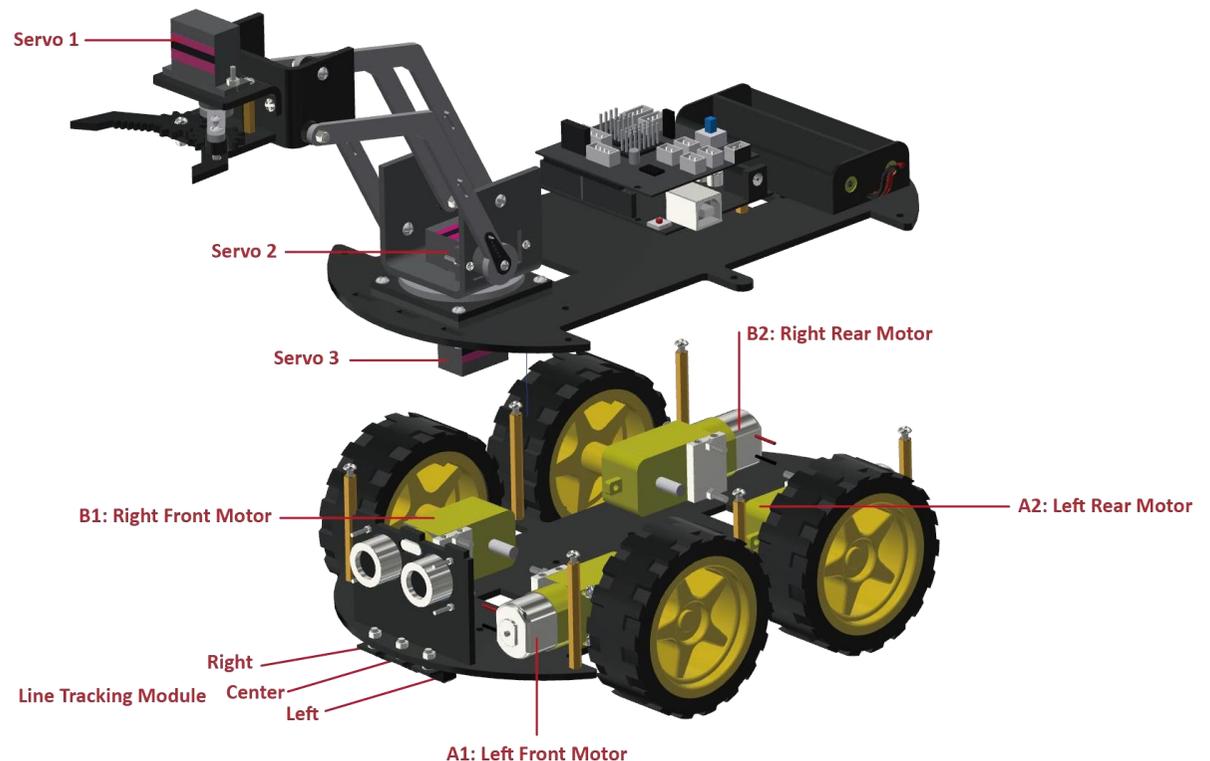
Antes de instalar el servo, debe establecer la posición inicial de los tres servomotores. Solo después de establecer el ángulo, puede instalarlo de acuerdo con la dirección vertical o paralela indicada por el video.

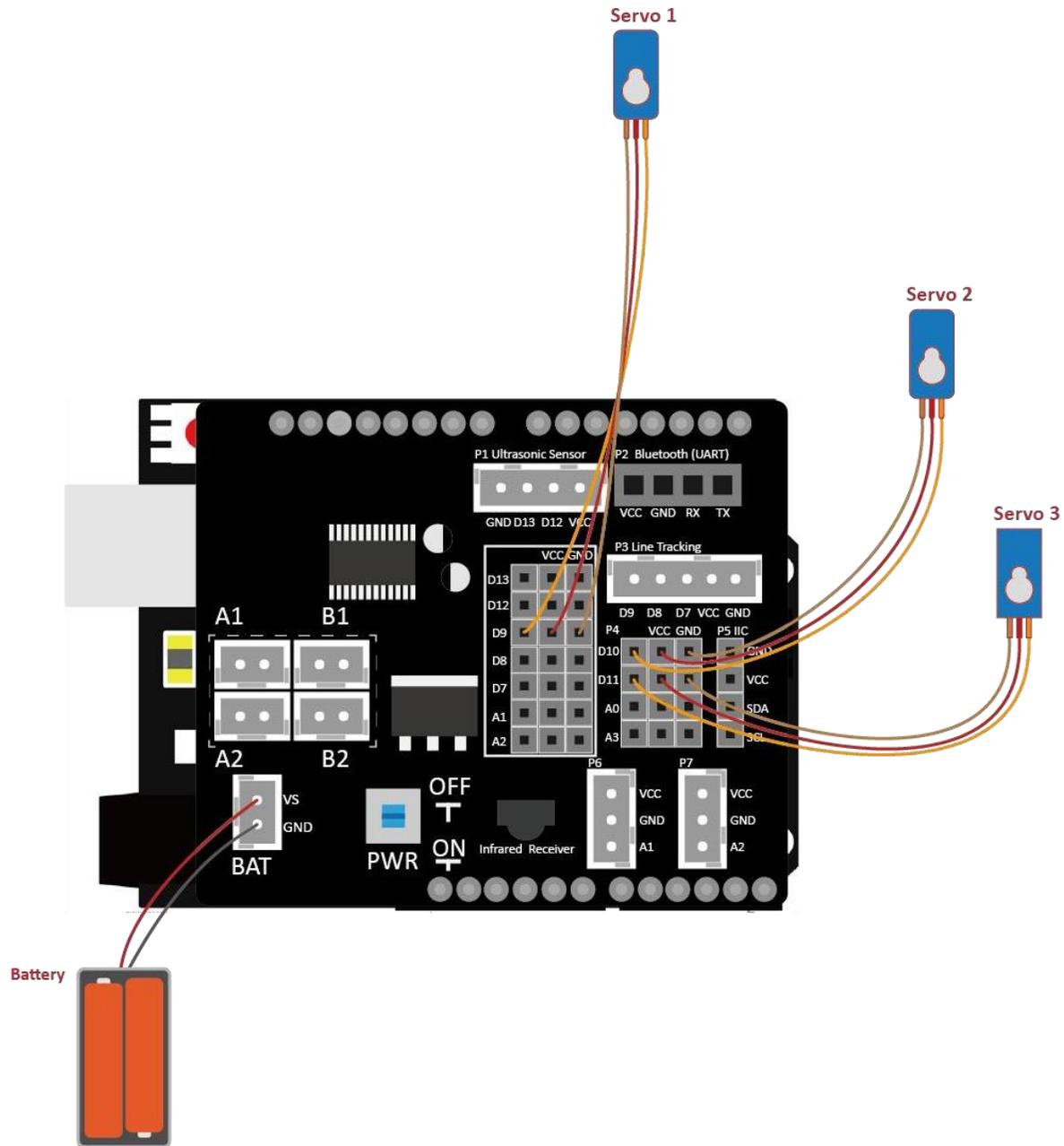
El método de calibración del ángulo del servomotor es el siguiente:

Garra-->Servo1-->D9--> 90°

Brazo-->Servo2-->D10-->135°

Base-->Servo3-->D11-->90°





El siguiente código es inicializar el valor de ángulo del servo, lo que facilita la instalación y el ajuste. Abra este programa fuente en **Arduino_Code>Lesson_2>Servo_Installation_Angle_Initialization.ino**.

```
#include <Servo.h>
```

```
int claw_degrees;
```

```
int arm_degrees;
```

```
int base_degrees;
```

```
Servo myservo1;//servo de Garra
```

```
Servo myservo2;//servo de Brazo
```

```
Servo myservo3;//servo de Base
```

```
void setup()
```

```
{
```

```
claw_degrees = 90;// Inicializar el valor de ángulo del servo 1 de Garra.  
arm_degrees = 135;// Inicializar el valor de ángulo del servo 2 del Brazo.  
base_degrees = 90;// Inicializar el valor de ángulo del servo 3 de la Base.  
servo 3myservo1.attach(9);// Servo 1 Garra está conectado a D9  
myservo2.attach(10);// Servo 2 Brazo está conectado a D10  
myservo3.attach(11);// Servo 3 Base está conectado a D11  
}  
void loop()  
{  
myservo1.write(claw_degrees);// Hace que el Servo 1 de Garra gire a 90°  
delay(200);  
myservo2.write(arm_degrees);// Hace que el Servo 2 del brazo gire a 135°  
delay(200);
```

```
myservo3.write(base_degrees);// Hace que el Servo 3 de Base gire a 90°
```

```
delay(200);
```

```
}
```

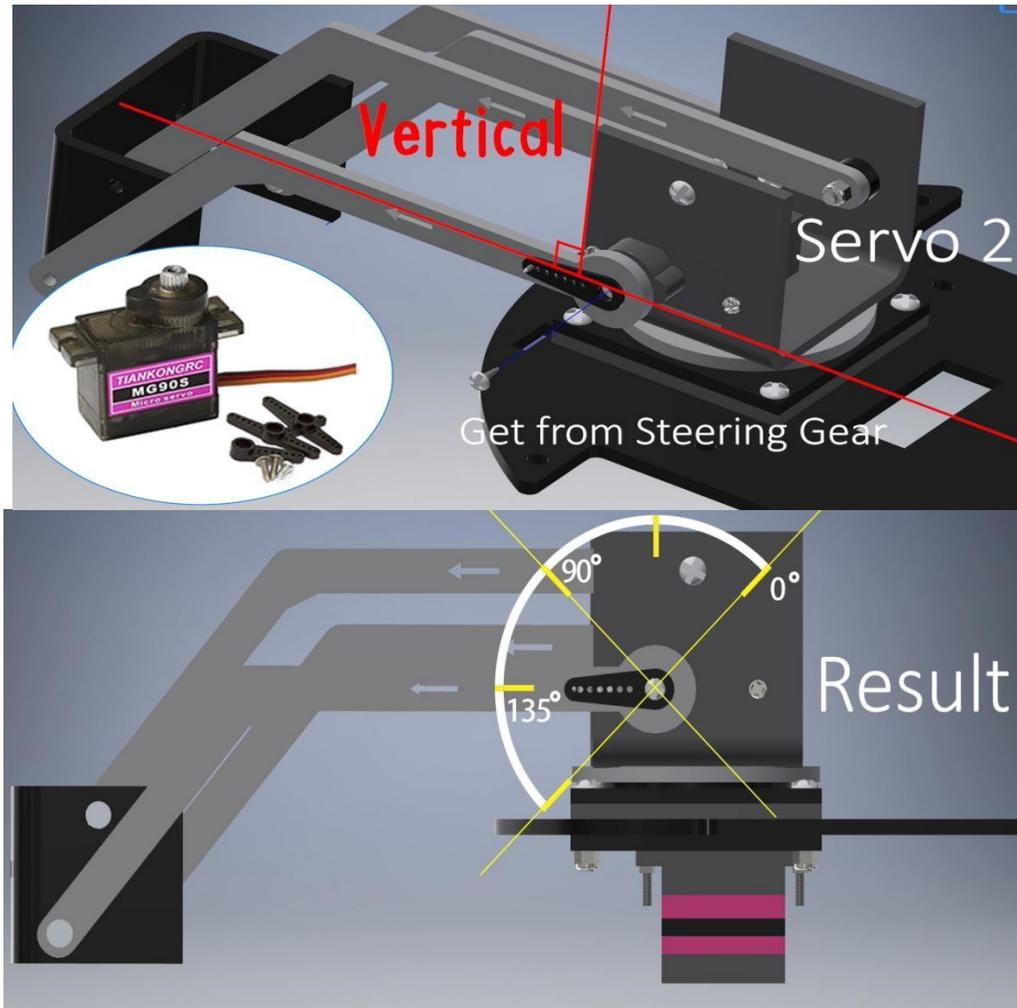
Sube este Código a Arduino UNO, enciende la alimentación. Servo 1 gira a 90° , servo 2 gira a 135° , y servo 3 gira 90° .

Inserte el servobrazo oscilante de plástico en el engranaje del servo de acuerdo con las direcciones paralelas y verticales marcadas en el video.

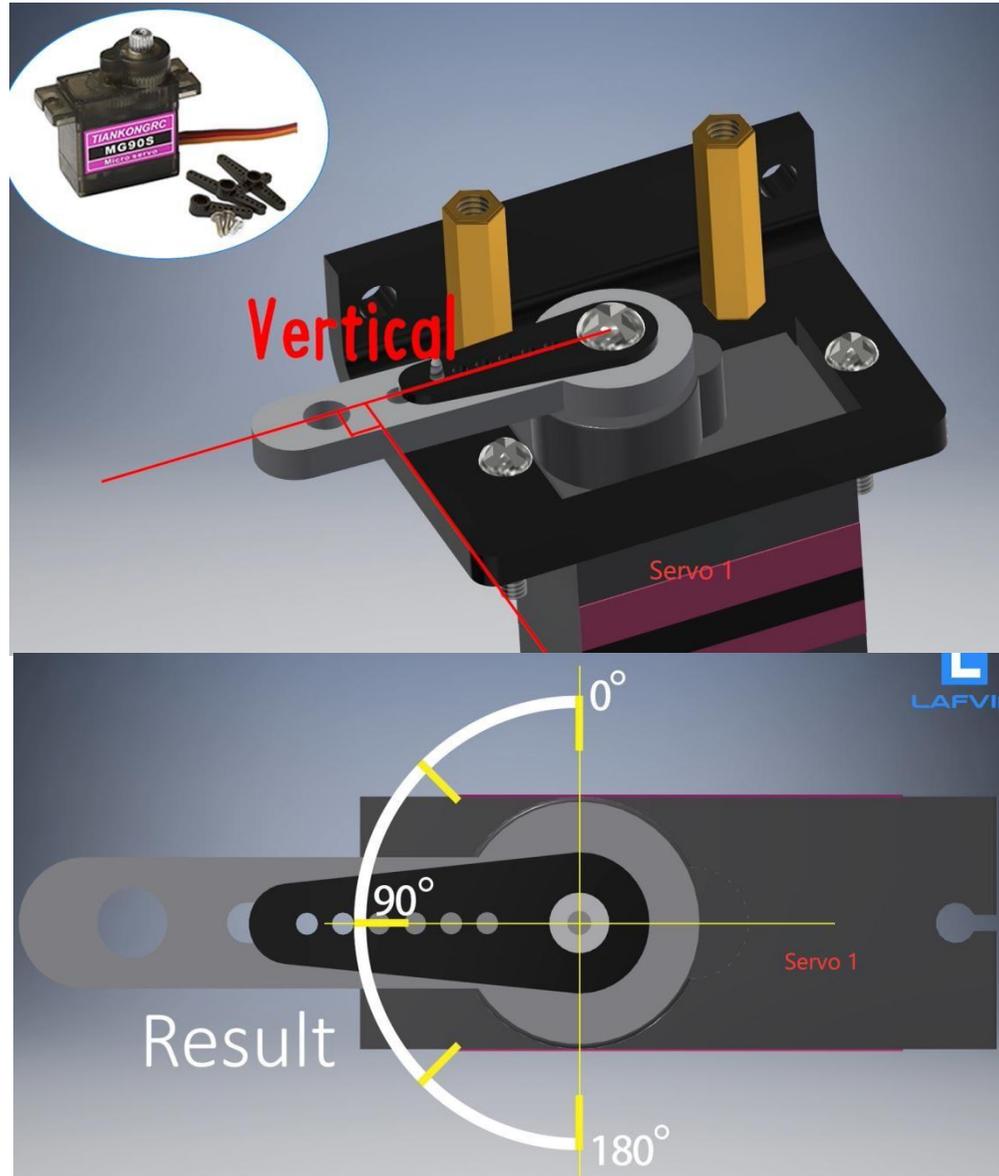
Base-->Servo3-->D11--> 90°



Brazo-->Servo2-->D10-->135°



Garra-->Servo1-->D9--> 90°



Lección 3 Control de velocidad y dirección del motor

Resumen

En esta lección aprenderemos cómo controlar la dirección y el control de velocidad del coche robot.

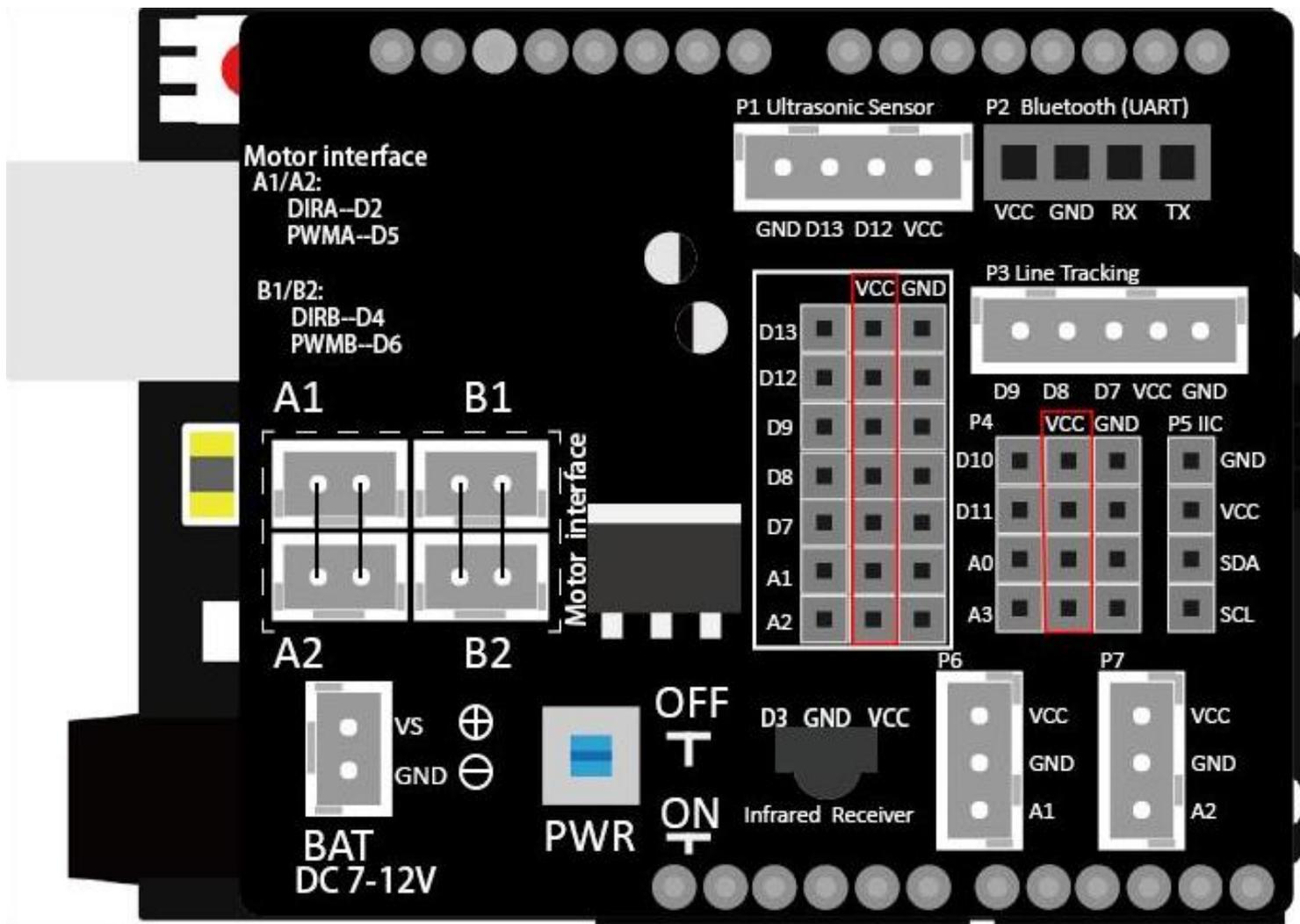
Motor Driver

La placa de expansión tiene integrado el chip de accionamiento del motor. El dispositivo de accionamiento del motor actual, que tiene una gran estructura de unión de puente MOSFET-H de corriente, salida de circuito de doble canal, puede impulsar 2 motores al mismo tiempo. Emite una potencia de accionamiento continuo de hasta 1 A por canal de corriente, iniciando una corriente máxima de hasta 2A / 3A (pulso continuo / pulso único (punzón)); 4 modos de control del motor: avance / retroceso / freno / parada Final;

Especificaciones:

Voltaje de motor recomendado (VMOT): 7.4– 13.5 V

- Voltaje Lógico (VCC): 2.7 – 5.5 V
- Máxima corriente de salida: 3 A por canal
- Corriente continua de salida: 1 A por canal (se puede conectar en paralelo para entregar 2 A continuos)



A1 A2 B1 B2 del motor Arduino shield es la interfaz del motor.

Los motores conectados a la interfaz A1 y a la interfaz A2 tienen la misma velocidad y la misma dirección.

Los motores conectados a los puertos B1 y B2 tienen la misma velocidad y la misma dirección.

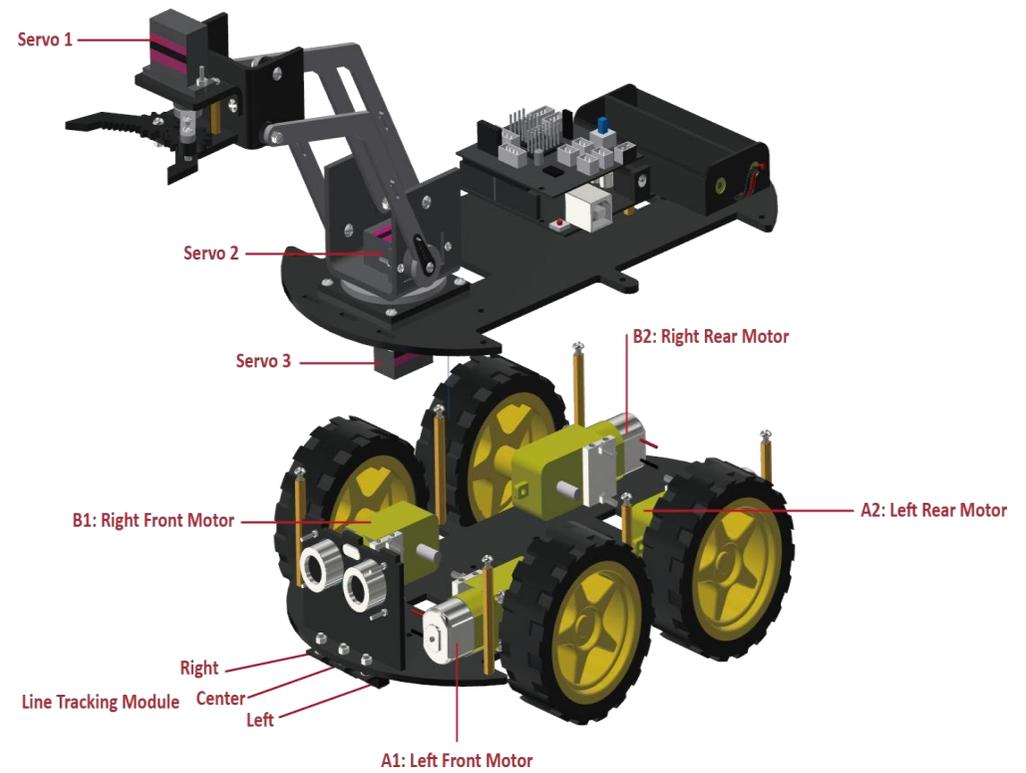
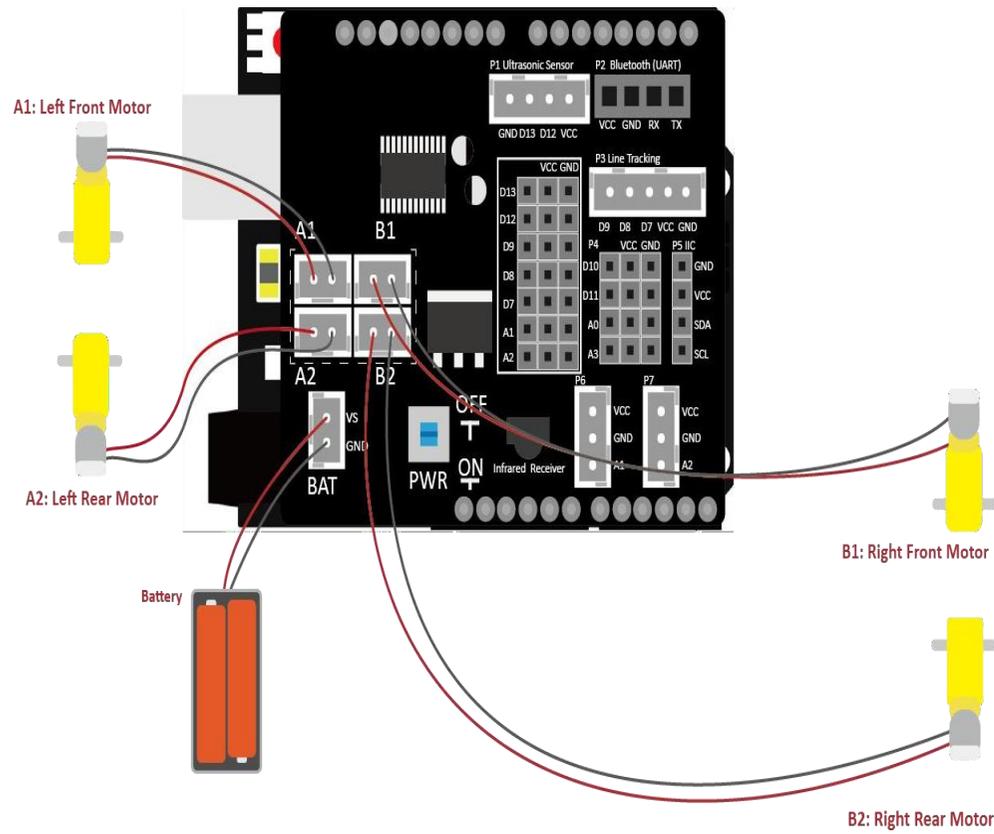
El puerto de E/S digital D2 controla la dirección del motor del puerto A, y el puerto de E/S digital D5 emite la señal PWM para controlar la velocidad del motor del puerto A.

El puerto de E/S digital D4 controla la dirección del motor de la interfaz B, y el puerto de E/S digital D6 emite la señal PWM para controlar la velocidad del motor de la interfaz B.

El valor de PWM está en el rango de 0-255. Cuanto mayor sea el valor, más rápido girarán los motores. Cuando PWM=0, significa detenerse, cuando PWM=255 significa velocidad máxima.

4WD Robot	D2	D5(PWM)	D4	D6(PWM)
Adelante	HIGH	0-255	LOW	0-255
Atrás	LOW	0-255	HIGH	0-255
Girar a la Izquierda	LOW	0-255	LOW	0-255
Girar a la Derecha	HIGH	0-255	HIGH	0-255
Parar	/	0	/	0

Como conectar el circuito



Vamos a programar

Prueba 1--Control de velocidad y dirección del motor

En la prueba experimental 1, escribiremos código para controlar el coche robot para que se mueva hacia adelante, hacia atrás, gire a la izquierda, gire a la derecha, se detenga.

Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_3>Test_1_Motor_Speed_and_Direction_Control.ino.

Pensando en la Programación

/ Defina una subfunción, la función de la subfunción es controlar el automóvil robot para avanzar, conveniente llamar a*

*otras funciones o repetir bucles */*

void Move_Forward(int speed) // Definir la función de avance de la velocidad de entrada

{

digitalWrite(2,HIGH); //D2 digital I/O el puerto controla la dirección del motor de la interfaz A

```
analogWrite(5,speed); //D5 digital I/O puerto salida de señal PWM para controlar la velocidad del motor del puerto A.  
digitalWrite(4,LOW); //D4 digital I/O el puerto controla la dirección del motor de la interface B  
analogWrite(6,speed); //D6 digital I/O puerto de salida de señal PWM para controlar la velocidad de la interface B motor.  
}
```

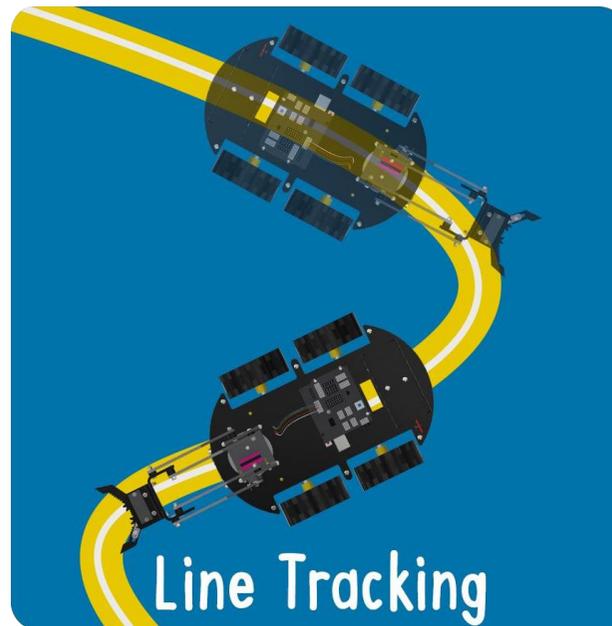
¿Qué verás?

Conéctese por diagrama de conexión, cargue el código y encienda, el automóvil inteligente avanza y retrocede durante 2s, gira a la izquierda y a la derecha durante 2s y se detiene durante 2s alternativamente.

Lección 4 Seguimiento de líneas Smart Car

Resumen

En esta lección, completaremos la prueba de dos códigos experimentales. En la prueba experimental 1, aprendimos a usar el sensor infrarrojo de seguimiento de línea y observamos los resultados devueltos por el sensor para distinguir objetos blancos y negros. En la prueba experimental 2, aprendimos a combinar el sensor infrarrojo de seguimiento de línea con el motor para controlar el coche robot para completar la función de seguimiento de línea.



¿Qué es el sensor de seguimiento de línea?

El sensor de seguimiento es en realidad un sensor infrarrojo. El componente utilizado aquí es el tubo infrarrojo TCRT5000. Su principio de funcionamiento es utilizar la diferente reflectividad de la luz infrarroja al color, luego convertir la intensidad de la señal reflejada en una señal de corriente. Durante el proceso de detección, el negro está activo en el nivel ALTO, pero el blanco está activo en el nivel BAJO. La altura de detección es de 0-3 cm. Al girar el potenciómetro ajustable en el sensor, puede ajustar la sensibilidad de detección del sensor.

Cómo utilizar el sensor de seguimiento de línea

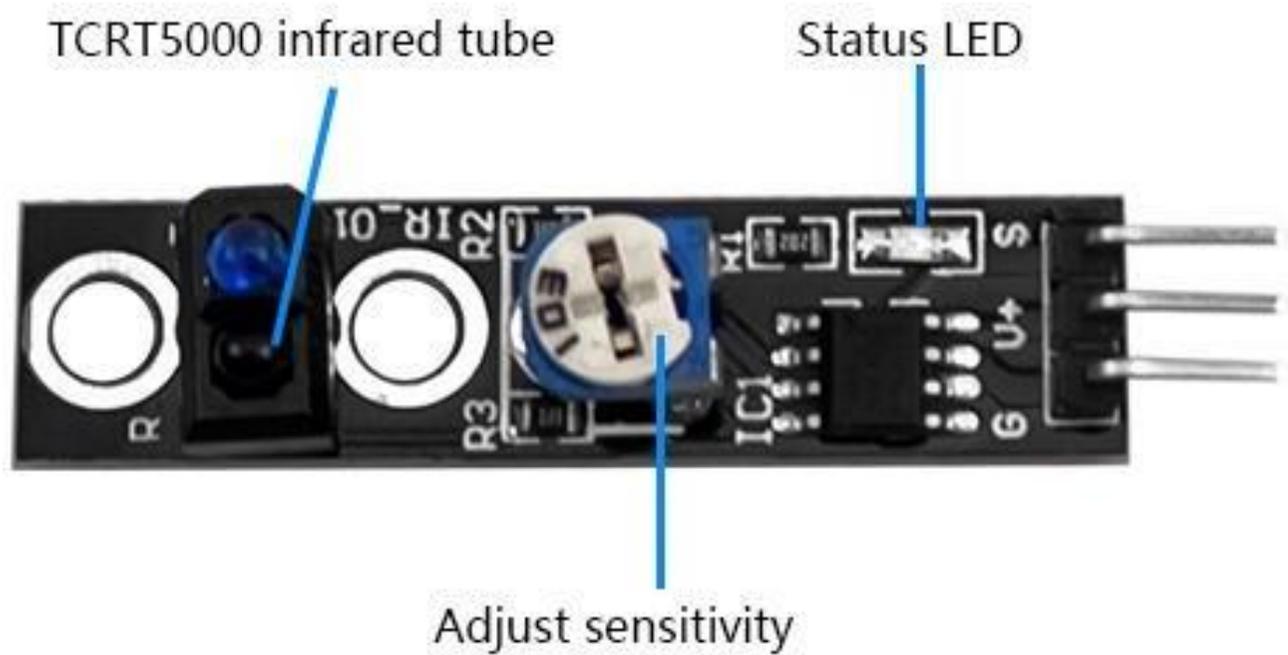
Voltaje de Operación: 3.3-5V (DC)

Interface: G(GND) V+(VCC) S(Signal)

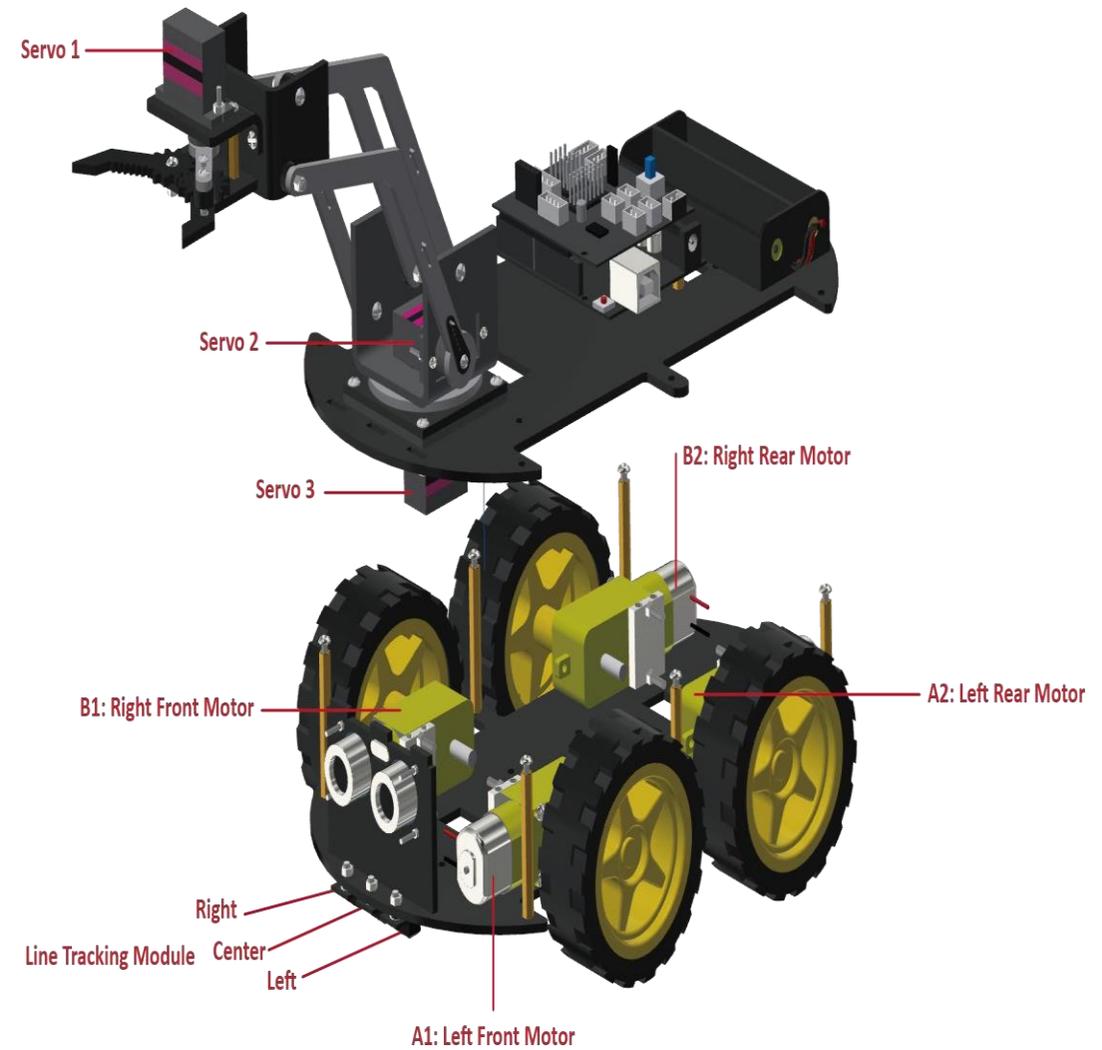
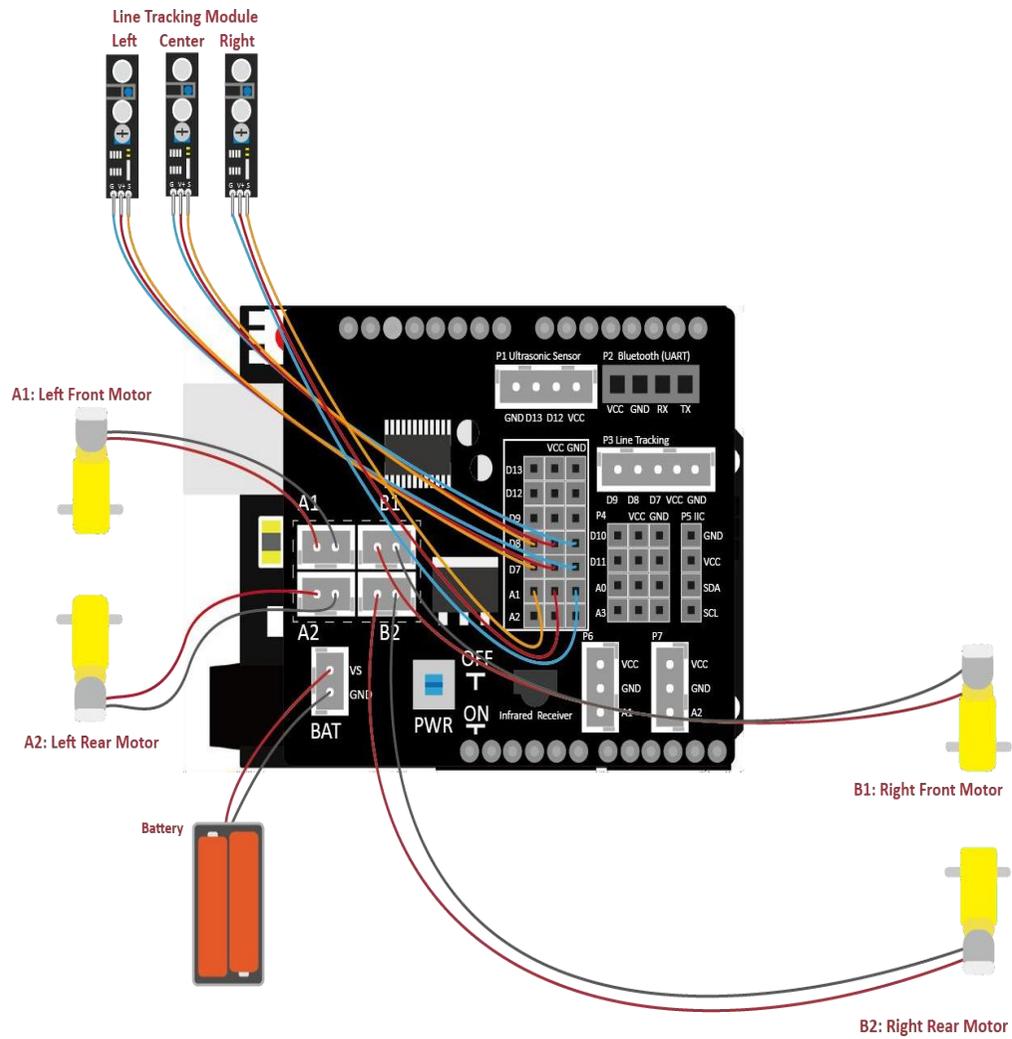
Señal de salida: Digital signal

Detección Altura: 0-3 cm

Nota especial: antes de probar, gire el potenciómetro en el sensor para ajustar la sensibilidad de detección. Cuando ajuste el LED en el umbral entre ON y OFF, la sensibilidad es la mejor.



Como conectar el circuito.



Vamos al programa

Prueba 1--Sensor de seguimiento de línea

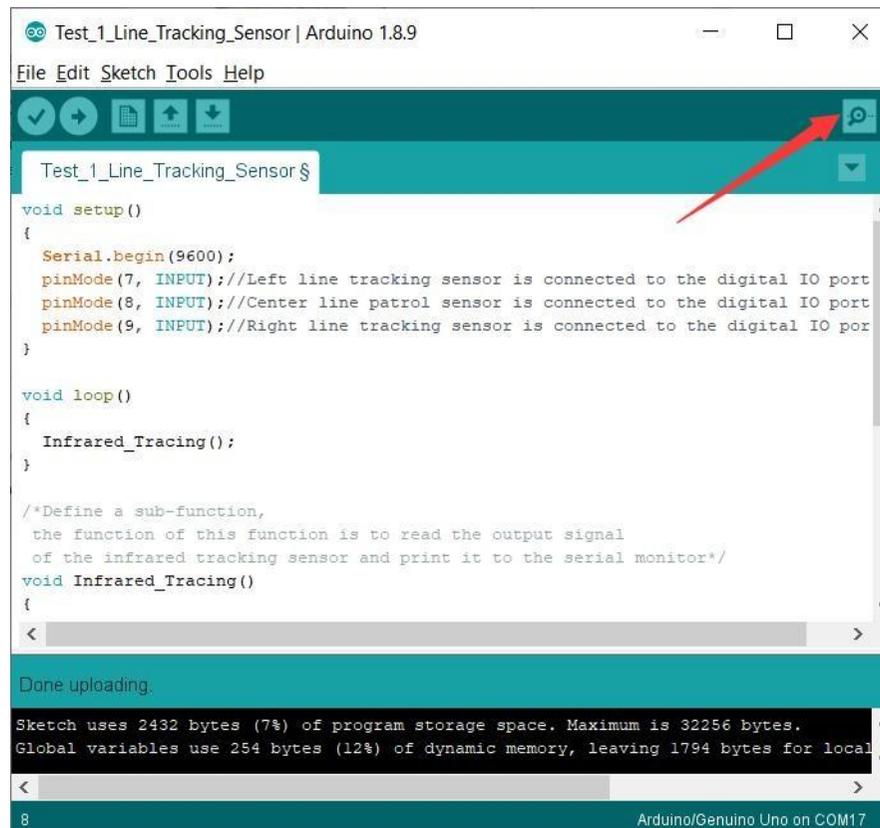
El objetivo principal del experimento de prueba es leer la señal de retorno del sensor de seguimiento de línea e imprimirla en el monitor del puerto serie. Cuando detecta papel blanco, el pin de señal del sensor emite LOW (pantalla 0) y el LED de estado está encendido; Cuando detecta negro, el pin de señal del sensor emite HIGH (pantalla 1) y el LED de estado está apagado.

Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_4>**Test_1_Line_Tracking_Sensor.ino**.

Después de cargar el código, haga clic en el botón en la esquina superior derecha para abrir el monitor serie y ver la distancia medida



```
Test_1_Line_Tracking_Sensor | Arduino 1.8.9
File Edit Sketch Tools Help
Test_1_Line_Tracking_Sensor $
void setup()
{
  Serial.begin(9600);
  pinMode(7, INPUT); //Left line tracking sensor is connected to the digital IO port
  pinMode(8, INPUT); //Center line patrol sensor is connected to the digital IO port
  pinMode(9, INPUT); //Right line tracking sensor is connected to the digital IO port
}

void loop()
{
  Infrared_Tracing();
}

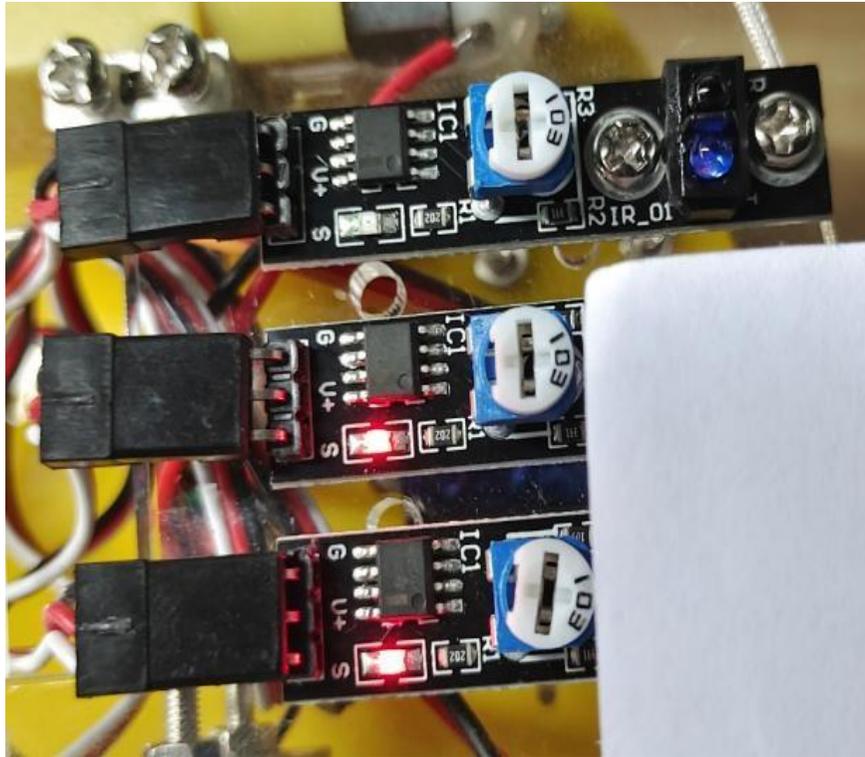
/*Define a sub-function,
the function of this function is to read the output signal
of the infrared tracking sensor and print it to the serial monitor*/
void Infrared_Tracing()
{
}

Done uploading.
Sketch uses 2432 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 254 bytes (12%) of dynamic memory, leaving 1794 bytes for local
8 Arduino/Genuino Uno on COM17
```

Luego puede ver los datos como lista:

Cuando detecta papel blanco, el pin de señal del sensor emite LOW (pantalla 0) y el LED de estado está encendido;

Cuando detecta negro, el pin de señal del sensor emite HIGH (pantalla 1) y el LED de estado está apagado.

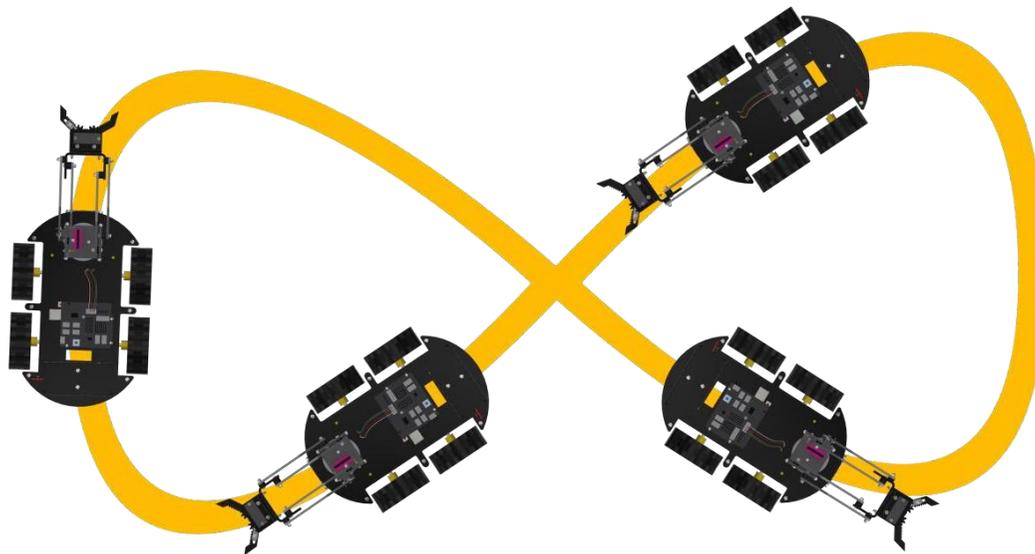


```
COM17
Center Tracking value:1
Right Tracking value:0
Left Tracking value:0
Center Tracking value:0
Right Tracking value:1
Left Tracking value:1
Center Tracking value:1
Right Tracking value:1
Left Tracking value:0
Center Tracking value:0
Right Tracking value:1
Left Tracking value:0
Center Tracking value:1
Right Tracking value:0
Left Tracking value:0
Center Tracking value:0
```

Autoscroll Show timestamp Newline

Prueba 2 líneas de seguimiento Smart Car

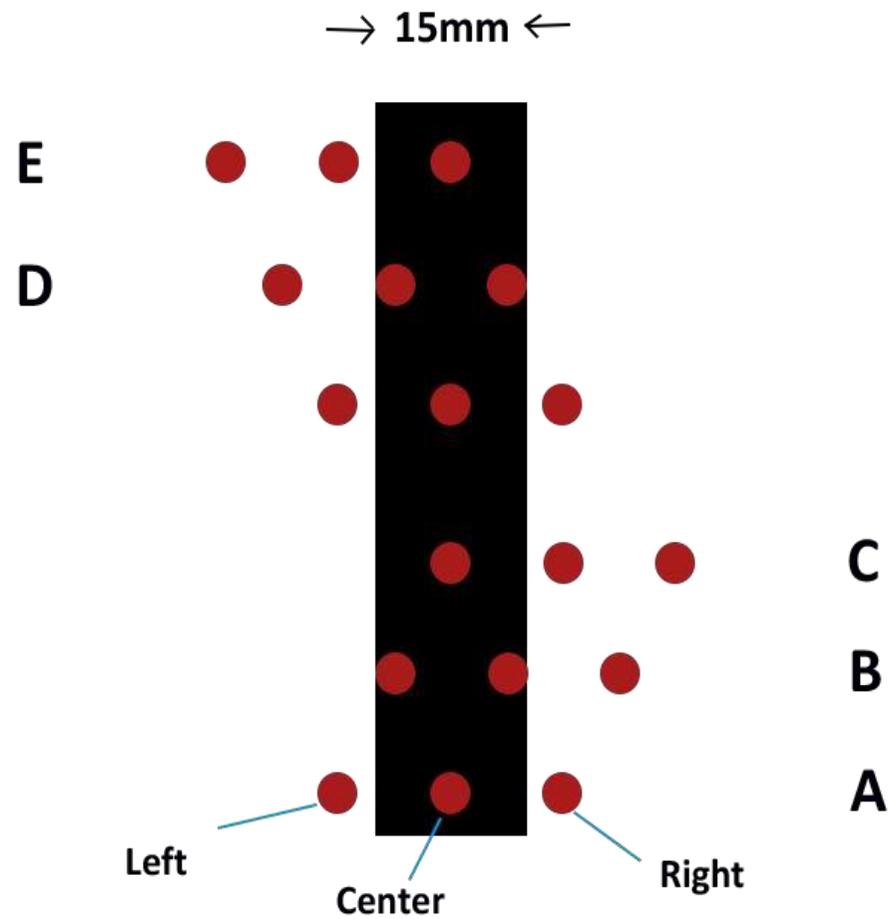
En la prueba experimental 2, aprendimos a combinar el sensor infrarrojo de seguimiento de línea con el motor para controlar el automóvil robot para completar la función de seguimiento de línea.



Pensando en la programación

La parte de inspección de línea del coche robot incluye tres sensores de seguimiento de línea infrarroja, a saber, el sensor de seguimiento de línea izquierda, el sensor de seguimiento de línea media y el sensor de seguimiento de línea derecha.

Se incluye un rollo de cinta aislante negra en las piezas del kit. El ancho de la cinta es de 15 mm, puede usarlo para planificar la trayectoria del automóvil. Cuando el robot completa la función de seguimiento de línea, pueden ocurrir las siguientes situaciones.



Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_4>**Test_2_Line_Tracking_Smart_Car.ino.**

Análisis de Programa

Cuando está en el **Estado A**, solo el sensor de seguimiento de la línea media detecta la línea negra, y el automóvil robot se mueve directamente a una velocidad de 105.

```
if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))  
{  
    Move_Forward(105);  
}
```

Cuando está en el **Estado B**, el sensor de seguimiento de línea izquierda y el sensor de seguimiento de línea central detectan la línea negra, y el automóvil robot gira hacia la izquierda a una velocidad de 95.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))
```

```
{  
    Rotate_Left(95);  
}
```

Cuando está en el **Estado D**, el sensor de seguimiento de línea derecha y el sensor de seguimiento de línea central detectan la línea negra, y el automóvil robot gira hacia la derecha a una velocidad de 95.

```
else if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
```

```
{  
    Rotate_Right(95);  
}
```

Cuando está en el **Estado C**, el sensor de seguimiento de la línea izquierda detecta la línea negra y el robot gira hacia la izquierda a una velocidad de 120.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value != Black && Right_Tra_Value != Black))
```

```
{  
  Rotate_Left(120);  
}
```

Cuando está en el **Estado E**, el sensor de seguimiento de la línea derecha detecta la línea negra, y el coche robot gira hacia la derecha a una velocidad de 120%.

```
else if (Left_Tra_Value != Black && (Center_Tra_Value != Black && Right_Tra_Value == Black))  
{  
  Rotate_Right(120);  
}
```

Cuando los tres sensores de seguimiento de línea detectan la línea negra al mismo tiempo, el coche robot se detiene.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
```

```
{
```

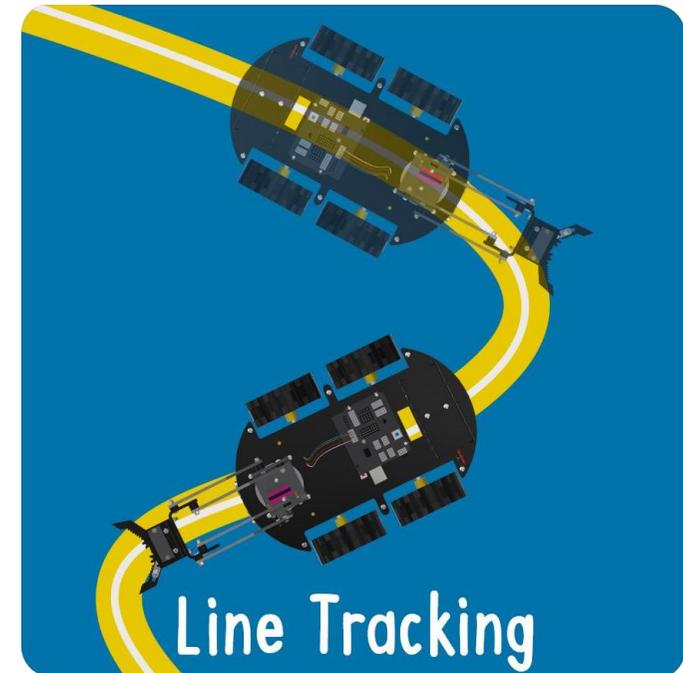
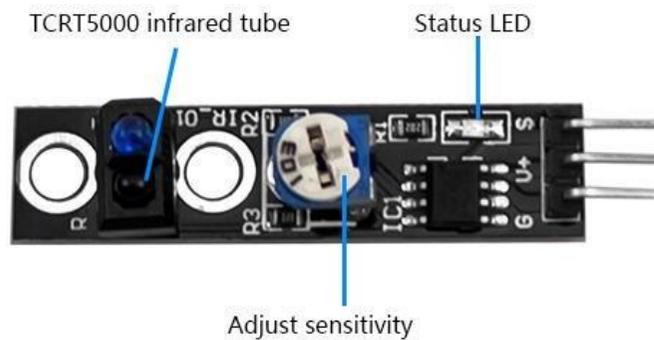
```
Stop();
```

```
}
```

¿Qué verás?

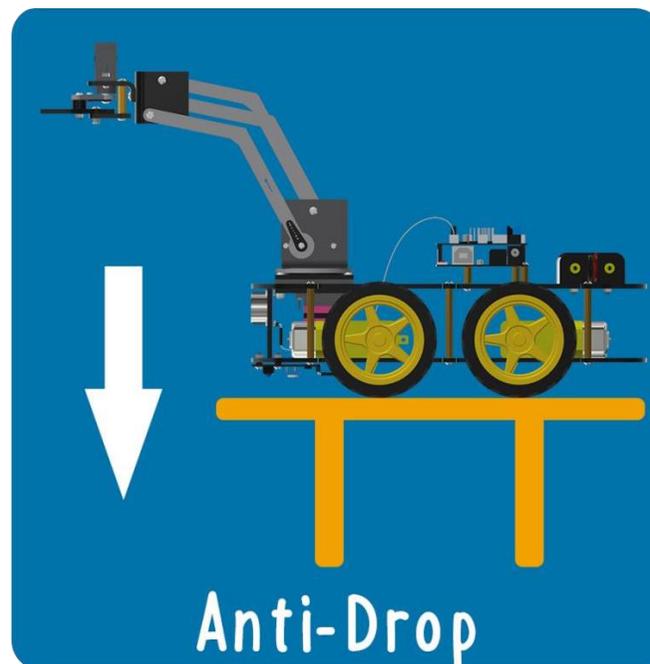
Cargue el código de prueba en la placa de control UNO R3, encienda el interruptor POWER. Entonces el coche inteligente se moverá a lo largo de la línea negra.

Nota: Los pisos de diferentes materiales en el hogar tienen diferentes grados de reflexión de la luz. Puede ajustar el potenciómetro en el sensor de seguimiento de línea para cambiar la sensibilidad de respuesta. Esto puede hacer que el automóvil siga la línea negra más suavemente.



Lección 5 Coche robot anticaída

Si prueba el sensor de línea infrarroja más profundamente, encontrará que el rango de detección está entre 0 y 3 cm. Cuando el sensor detecta un objeto que no es negro, emite una señal "0" y la luz indicadora del sensor se enciende; Cuando el sensor no detecta un objeto, emite una señal "1" y la luz indicadora del sensor se apaga. Si el coche robot se coloca en el escritorio para moverse, el sensor infrarrojo de seguimiento de línea puede ayudar al coche robot a determinar si ha llegado al borde del escritorio. Si es así, el coche robot retrocederá y girará para evitar caerse de la mesa.



¿Qué es el sensor de seguimiento de línea?

El sensor de seguimiento es en realidad un sensor infrarrojo. El componente utilizado aquí es el tubo infrarrojo TCRT5000. Su principio de funcionamiento es utilizar la diferente reflectividad de la luz infrarroja al color, luego convertir la intensidad de la señal reflejada en una señal de corriente. Durante el proceso de detección, el negro está activo en el nivel ALTO, pero el blanco está activo en el nivel BAJO.

La altura de detección es de 0-3 cm.

Al girar el potenciómetro ajustable en el sensor, puede ajustar la sensibilidad de detección del sensor.

Cómo utilizar el sensor de seguimiento de línea.

Voltaje de Operación: 3.3-5V (DC)

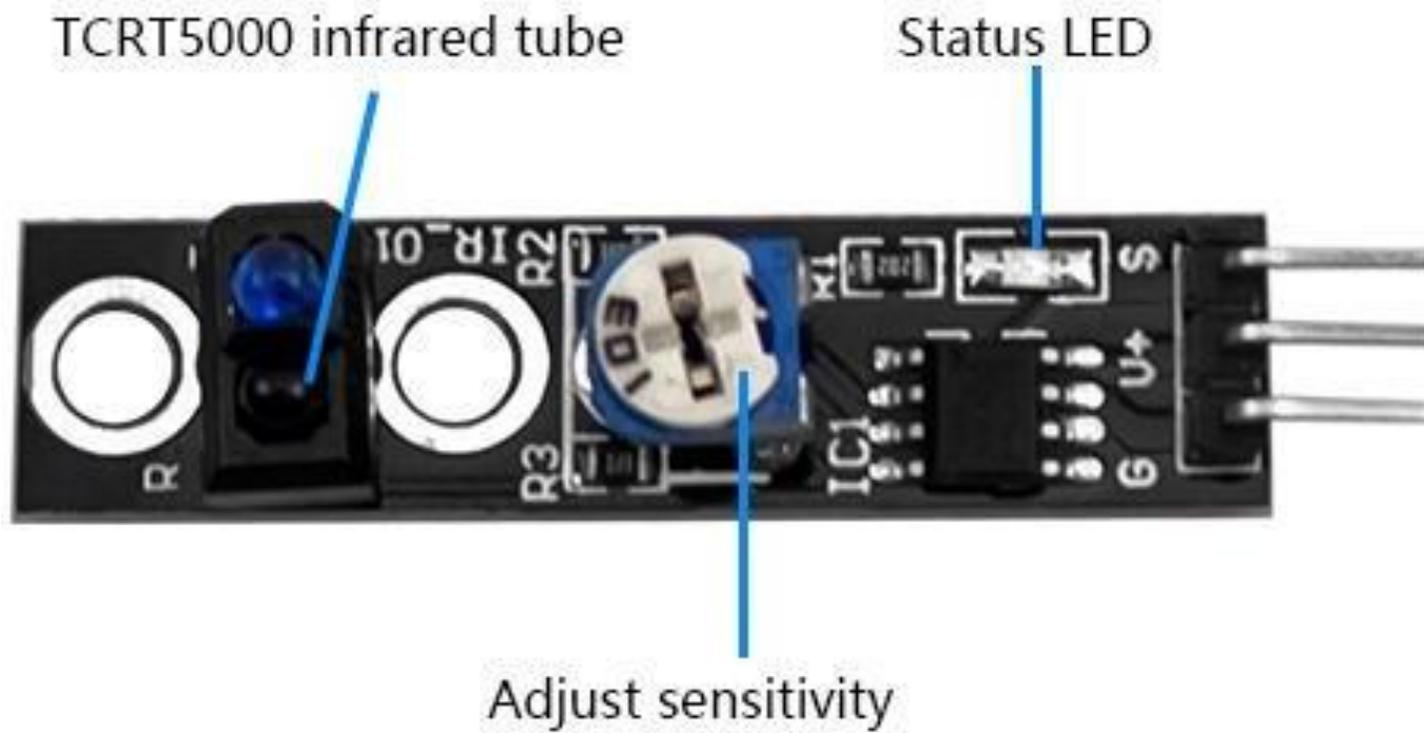
Interface: G(GND) V+(VCC) S(Signal)

Señal de salida: Señal Digital

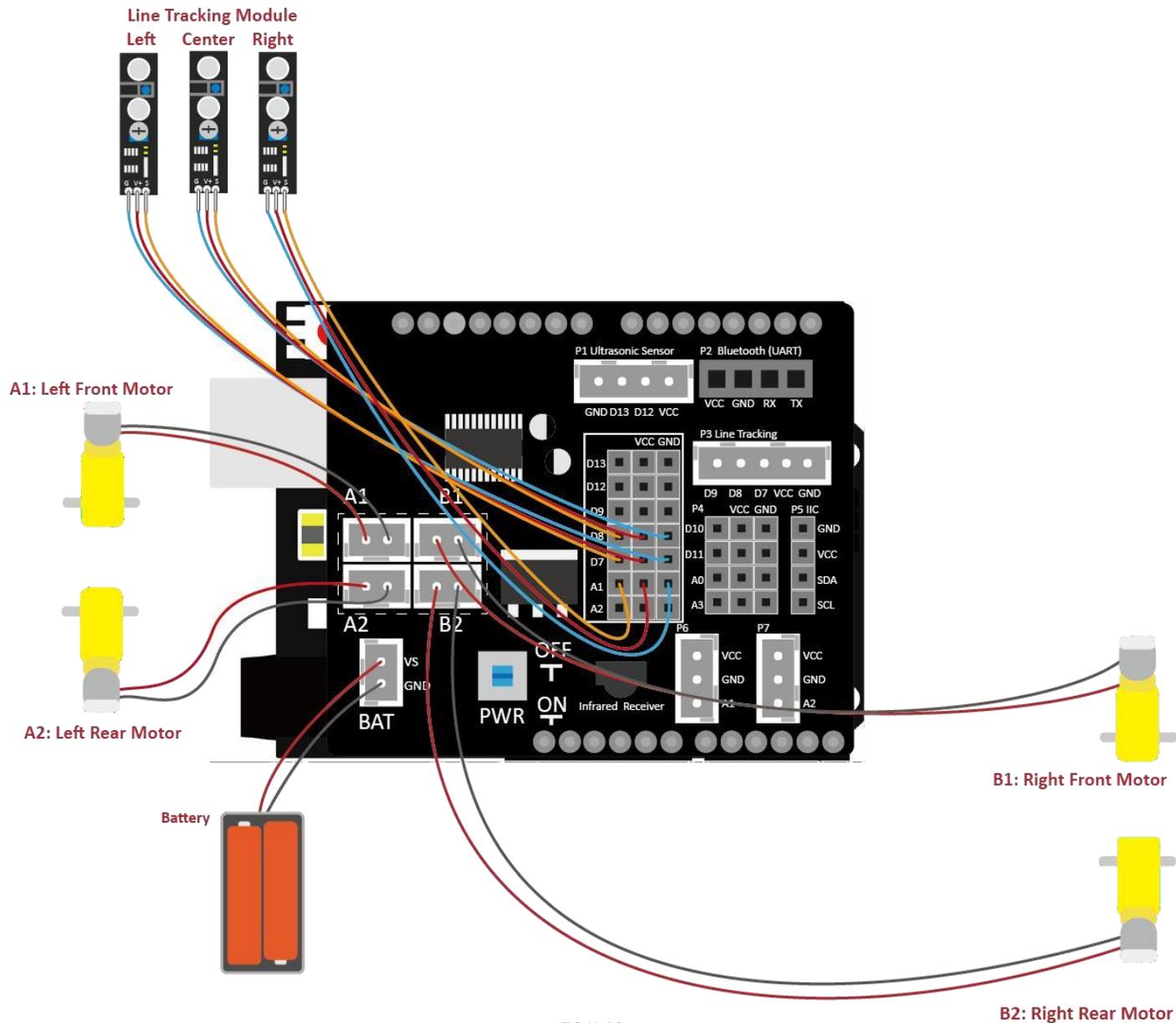
Altura de detección: 0-3 cm

Nota especial: Antes de probar, gire el potenciómetro en el sensor para ajustar la sensibilidad de detección. Cuando ajuste el LED en

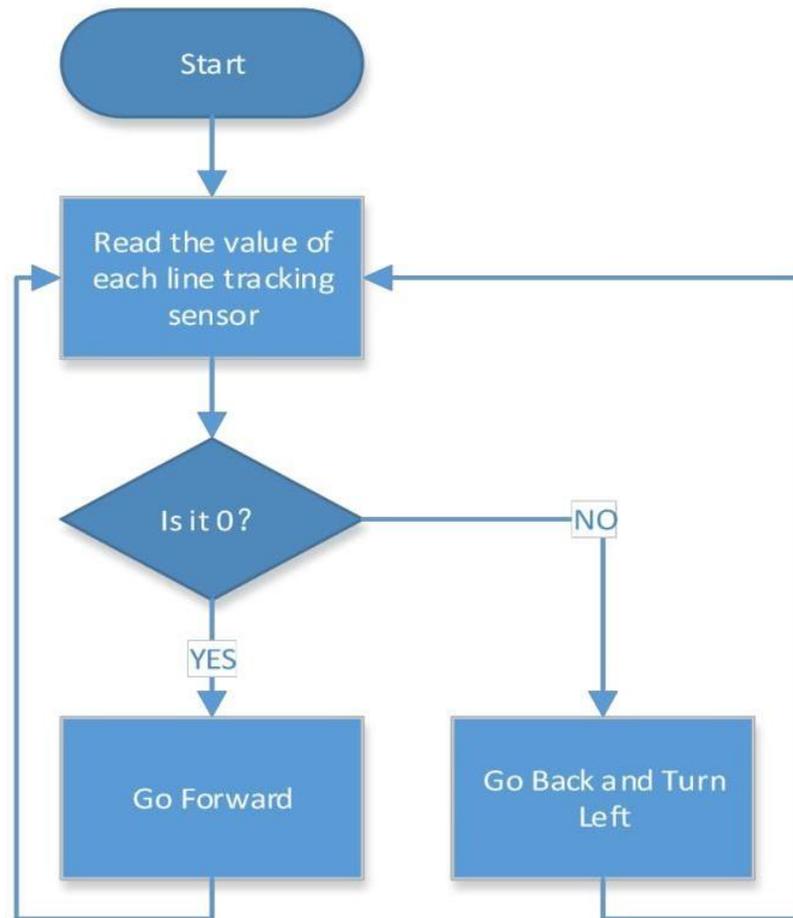
el umbral entre ON y OFF, la sensibilidad es la mejor.



Como conectar el circuito.



Pensando en la programación.



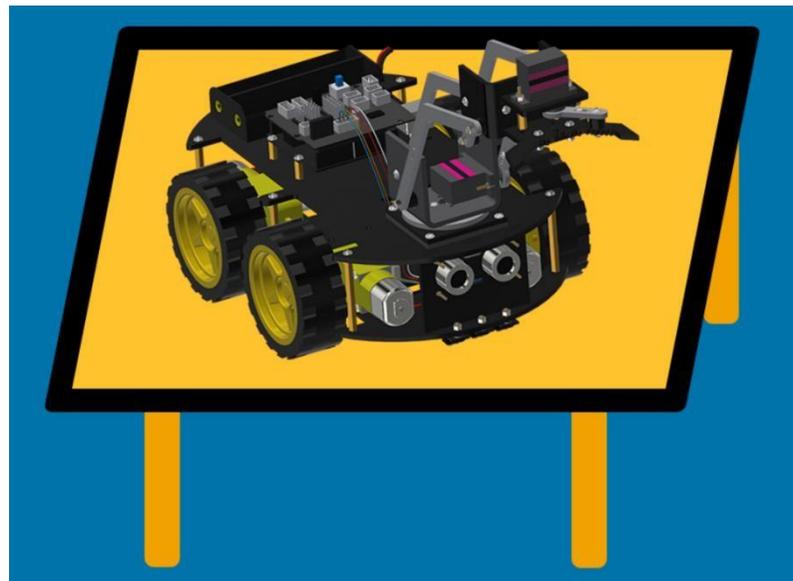
Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_5>**Test_1_Anti_Drop_Robot_Car.ino**.

¿Qué verás?

Cargue el código de prueba en la placa de control UNO R3. Use cinta negra para dibujar un área fija en la mesa, y el automóvil robot se moverá en el área fija. El coche inteligente retrocederá y girará a la izquierda una vez que esté cerca del borde del escritorio.



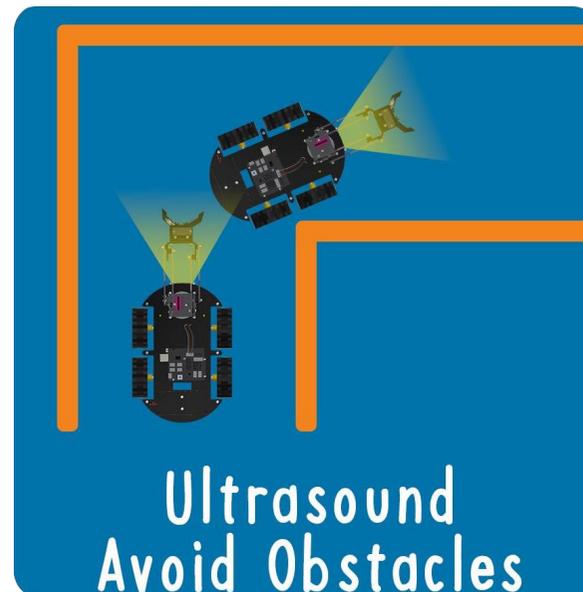
Lección 6 Vehículo robot ultrasónico para evitar obstáculos.

Resumen

En esta lección, completaremos la prueba de 2 códigos experimentales.

En la segunda prueba experimental 1, aprenderemos a usar el módulo ultrasónico para medir la distancia.

En la prueba experimental 2, el sensor ultrasónico se utiliza para medir la distancia del obstáculo directamente en frente y ayudar al coche robot a evitar el obstáculo.

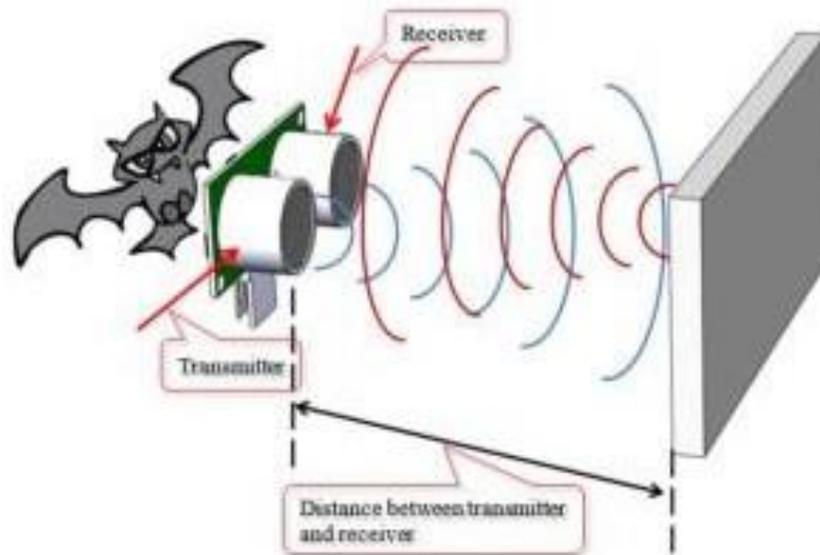


¿Qué es un sensor ultrasónico?

Revise el sensor ultrasónico de la lección anterior. Funciona como el ojo de un murciélago.

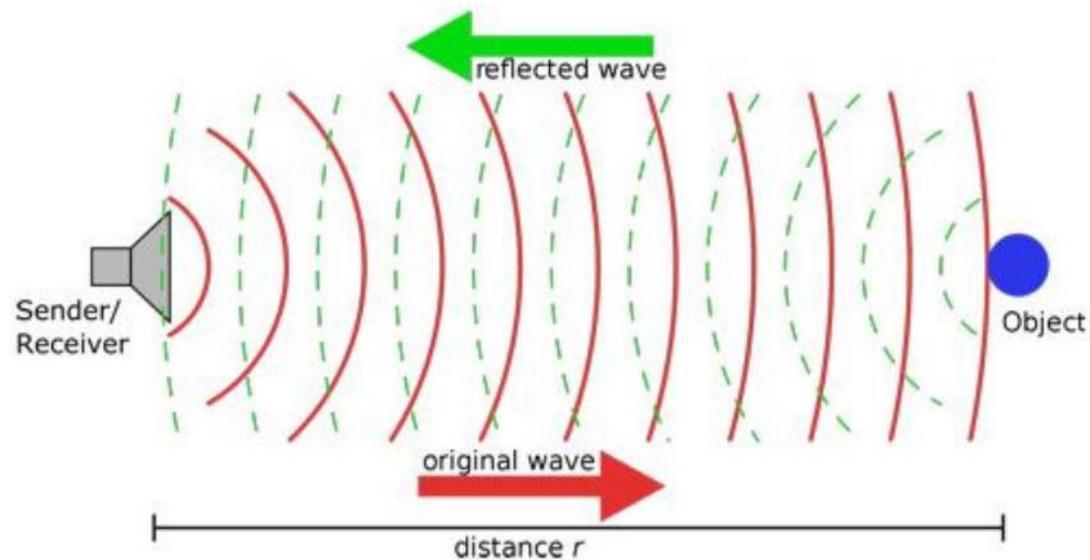
Determine la distancia de los obstáculos en frente después de recibir y recibir ondas de sonido de alta frecuencia.

Como se muestra en la siguiente imagen, es nuestro módulo ultrasónico. Tiene dos algo así como ojos. Uno es el extremo transmisor, el otro es el extremo receptor.



El sensor ultrasónico HC-SR04 utiliza un sonar para determinar la distancia a un objeto como lo hacen los murciélagos. Ofrece una excelente detección de rango sin contacto con alta precisión y lecturas estables en un paquete fácil de usar. Viene completo con módulos transmisores y receptores ultrasónicos. El HC-SR04 o el sensor ultrasónico se está utilizando en una amplia gama de proyectos electrónicos para crear aplicaciones de detección de obstáculos y medición de distancia, así como en otras aplicaciones.

Aquí hemos traído el método simple para medir la distancia con arduino y sensor ultrasónico y cómo usar el sensor ultrasónico con arduino.



Especificaciones

Fuente de alimentación: +5V

DC Corriente de reposo: <2mA

Corriente de Trabajo: 15mA

Ángulo Efectivo: 15° 92

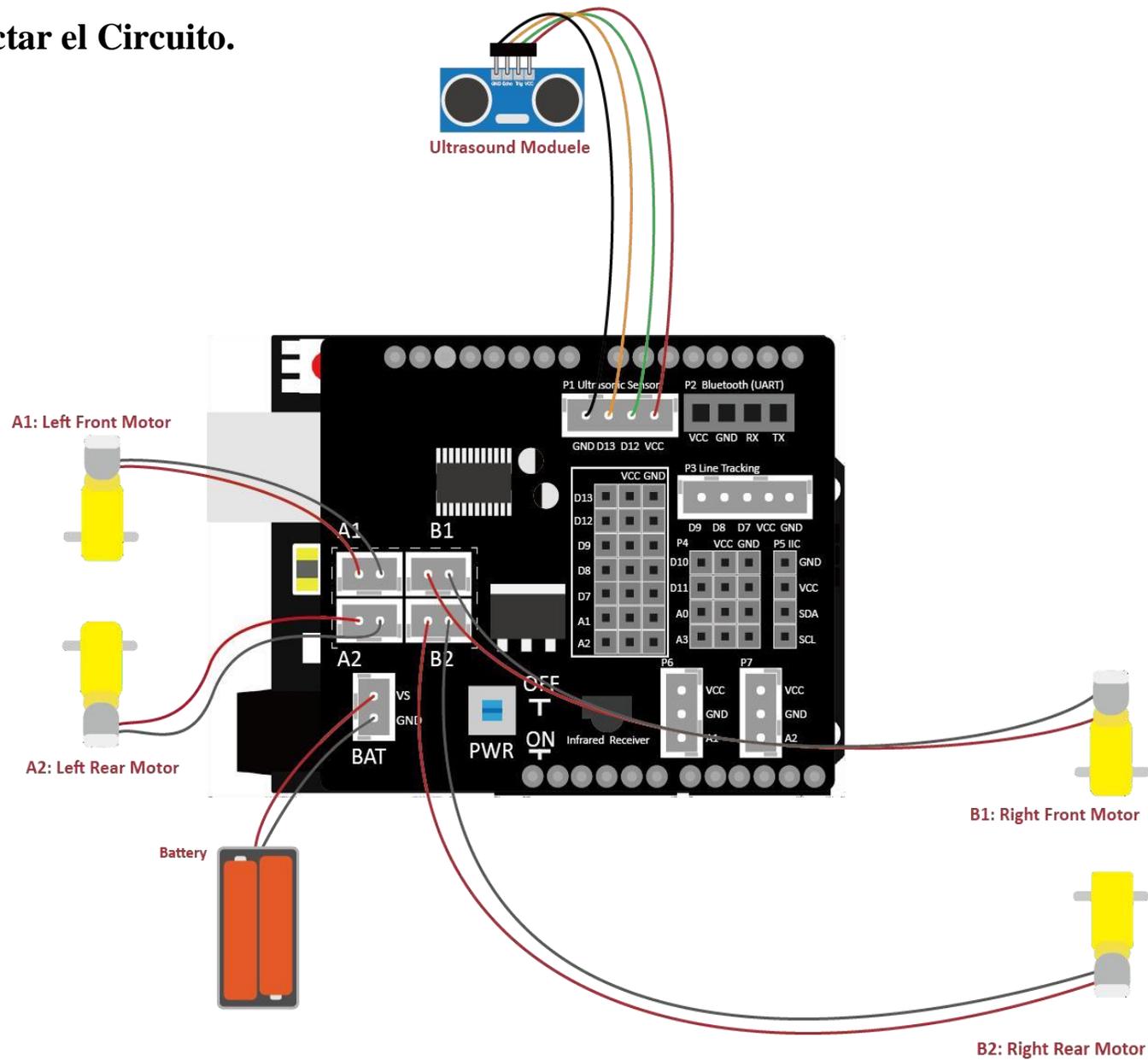
Rango de distancia: 2cm - 400 cm

Resolución: 0.3 cm

Ángulo de Medición: 30°

Ancho de pulso de entrada de disparo: 10uS

Como Conectar el Circuito.



Vamos al programa

Prueba 1—Módulo Sensor Ultrasónico

En la Prueba experimental 1, aprenderemos cómo controlar el sensor ultrasónico y mostrar la distancia medida por el sensor ultrasónico en el monitor serie.

Código Arduino

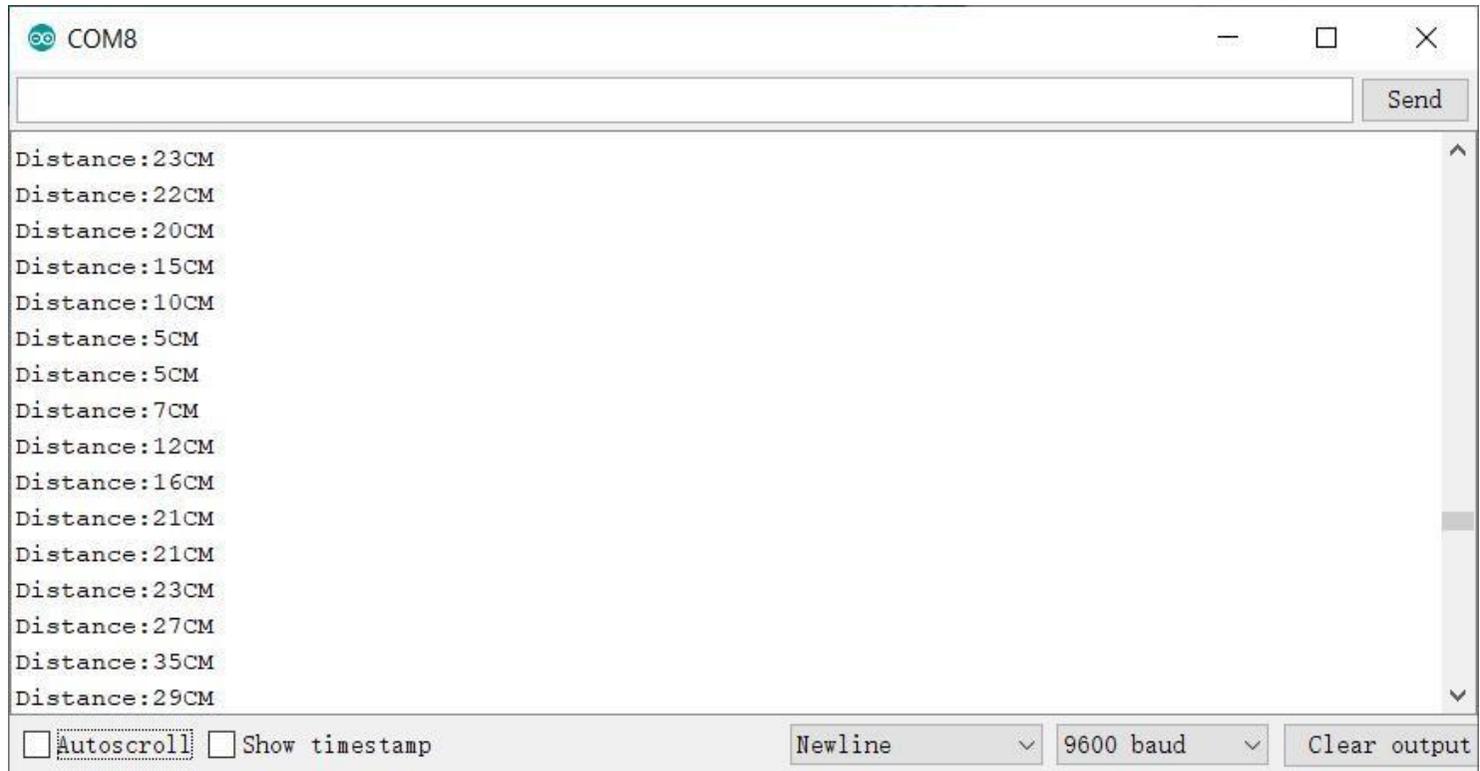
Si desea consultar el programa que proporcionamos. Abra el software IDE de Arduino y abra este programa fuente en

Arduino_Code>Lesson_6>**Test_1_Ultrasonic_Sensor_Module.ino**

Después de cargar el código, haga clic en el botón en la esquina superior derecha para abrir el monitor serie y ver la distancia medida.

```
Test_1_Ultrasonic_Sensor_Module | Arduino 1.8.9
File Edit Sketch Tools Help
Test_1_Ultrasonic_Sensor_Module
float distance = pulseIn(13, HIGH) / 58.00;
delay(10);
return distance;
}
void Ultrasonic_Sensor_Module() {
  int Distance = 0;
  Distance = checkdistance();
  Serial.print("Distance:");
  Serial.print(Distance);
  Serial.println("CM");
  delay(100);
}
void setup() {
  Serial.begin(9600);
  pinMode(12, OUTPUT);
  pinMode(13, INPUT);
}
void loop() {
  Ultrasonic_Sensor_Module();
}
Done uploading.
Sketch uses 3222 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 200 bytes (9%) of dynamic memory, leaving 1848 bytes for local variables. Maximum is 2048 bytes.
18 Arduino/Genuino Uno on COM8
```

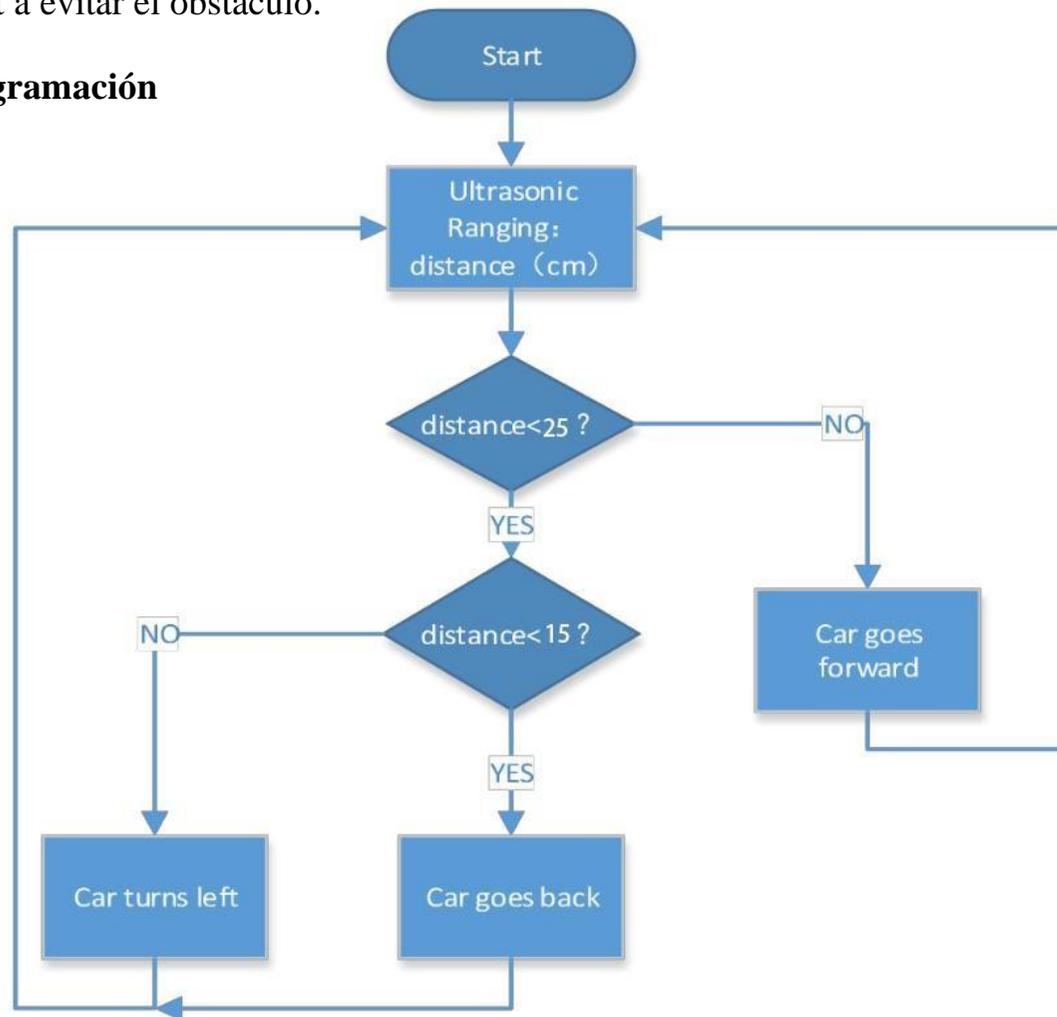
Entonces puedes ver los datos como lista:



Prueba 2—Vehículo Robot Evita Obstáculos Ultrasonico

En la prueba experimental 2, el sensor ultrasónico se utiliza para medir la distancia del obstáculo directamente en frente y ayudar al coche robot a evitar el obstáculo.

Pensando en la Programación



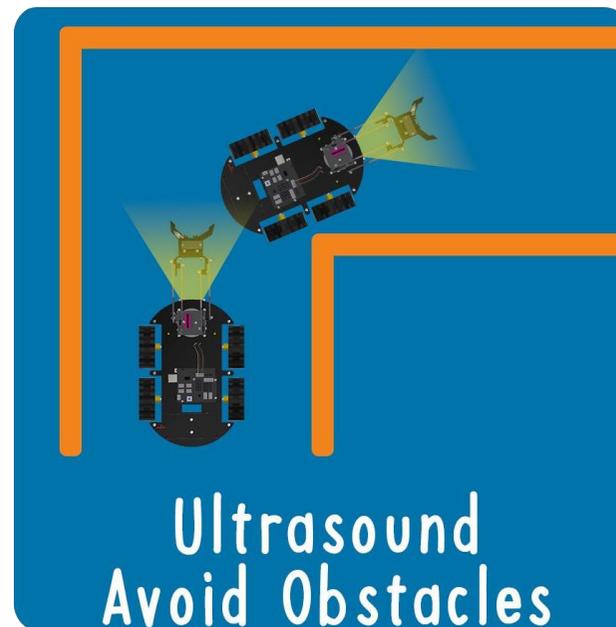
Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_6>Test_2_Ultrasonic_Infrared_Obstacle_Avoidance_Robot_Car.ino.

¿Qué verás?

Cargue el código en la placa de control UNO R3 y encienda el interruptor POWER. El coche inteligente evitará el obstáculo detectado.



Lección 7 Vehículo Robot de Seguimiento Ultrasónico

Resumen

En el último curso, aprendimos a usar el módulo ultrasónico y el módulo infrarrojo para evitar obstáculos. Del mismo modo, use el módulo ultrasónico para ayudar al automóvil robot a seguir los objetos en movimiento por delante.

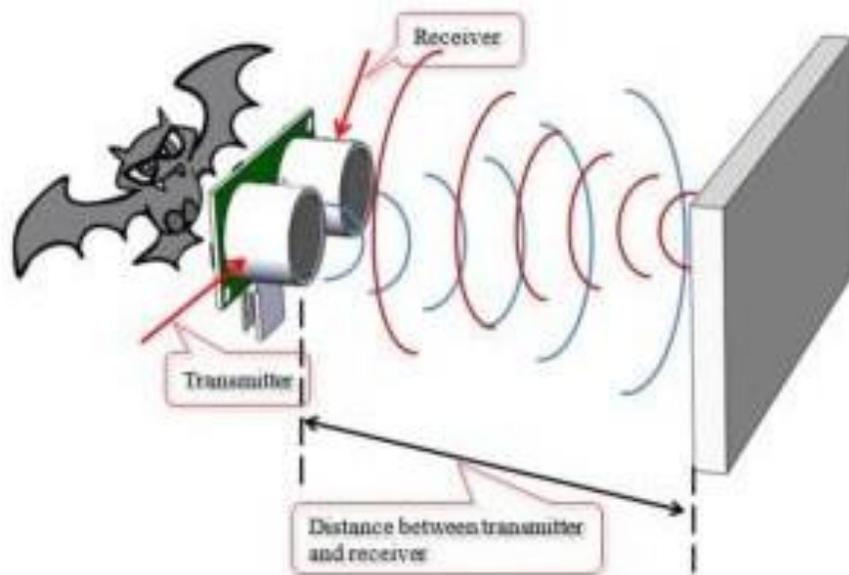
Cuando el objeto guiado está directamente en frente del coche robot, el sensor ultrasónico puede detectar la distancia entre el coche robot y el objeto guía. Cuando la distancia sea superior a 20mm, controla el coche robot para que se acerque al guía;

Cuando la distancia es inferior a 15 mm, controle el coche robot para que se mueva hacia atrás y mantenga una distancia adecuada, $15 \text{ mm} < < 20 \text{ mm}$.



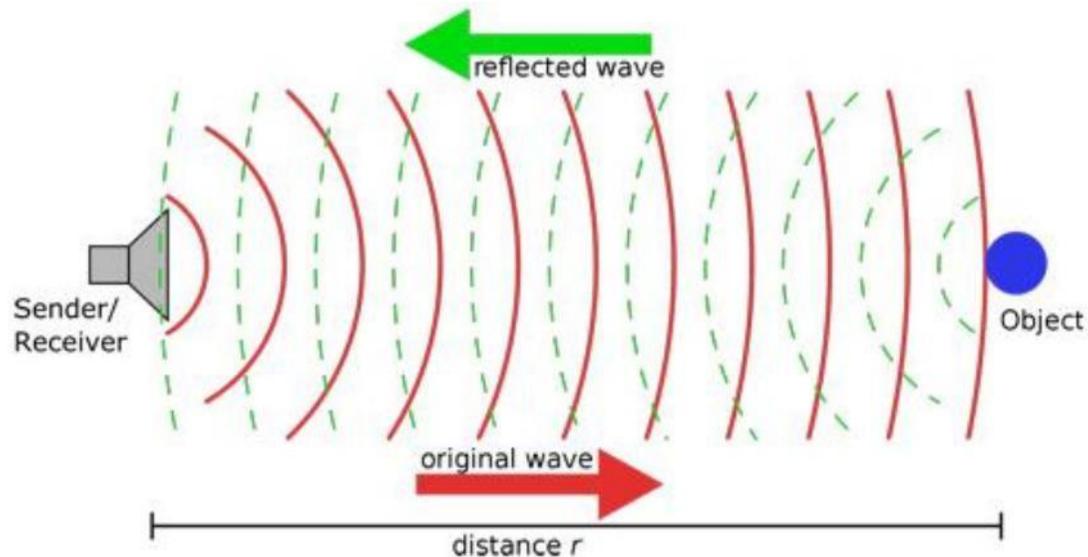
¿Qué es un sensor ultrasónico?

Revise el sensor ultrasónico de la lección anterior. Funciona como el ojo de un murciélago. Determine la distancia de los obstáculos en frente después de recibir y recibir ondas de sonido de alta frecuencia. Como se muestra en la siguiente imagen, es nuestro módulo ultrasónico. Tiene dos algo así como ojos. Uno es el extremo transmisor, el otro es el extremo receptor.

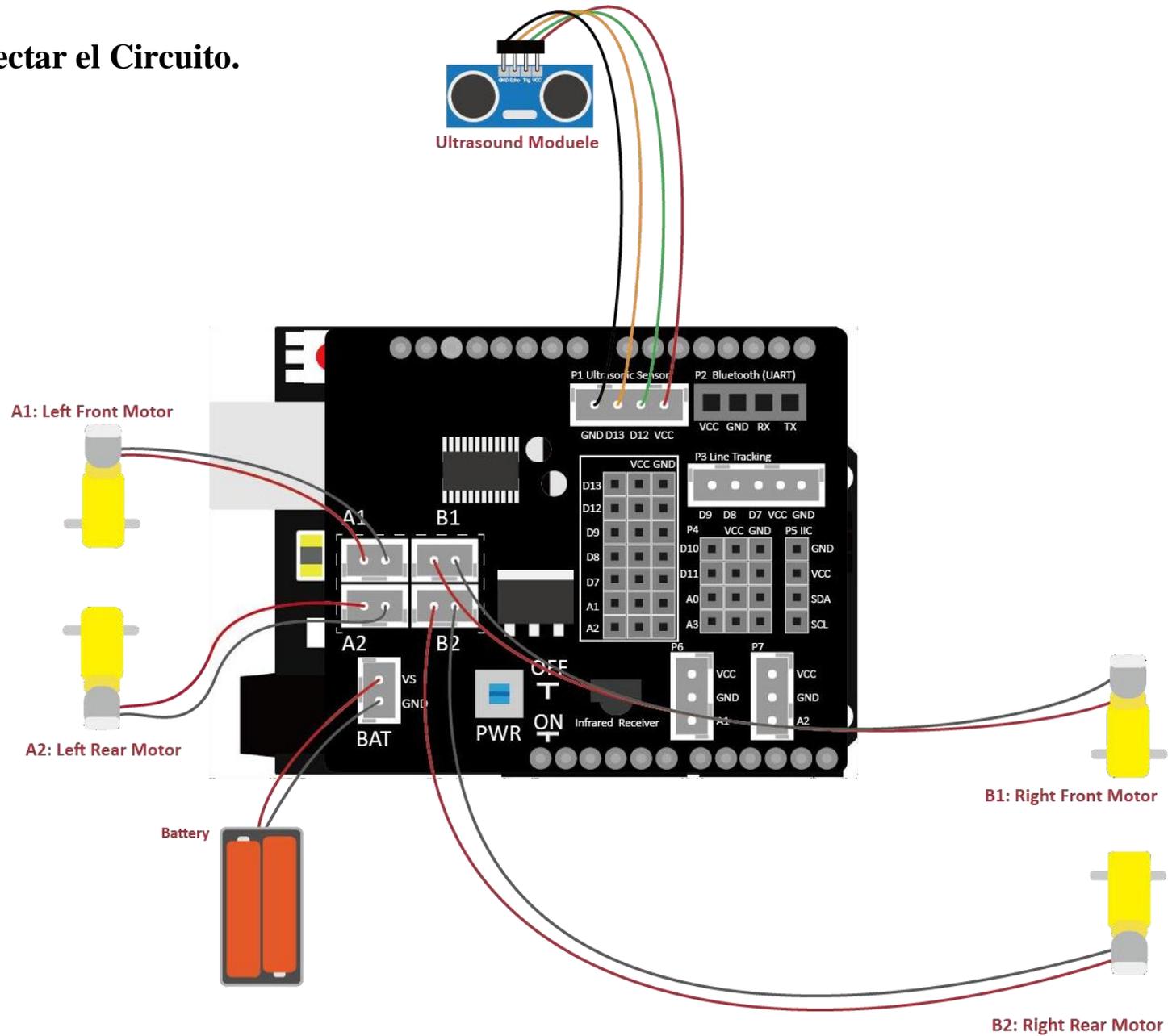


El sensor ultrasónico HC-SR04 utiliza un sonar para determinar la distancia a un objeto como lo hacen los murciélagos. Ofrece una excelente detección de rango sin contacto con alta precisión y lecturas estables en un paquete fácil de usar. Viene completo con módulos transmisores y receptores ultrasónicos. El HC-SR04 o el sensor ultrasónico se está utilizando en una amplia gama de proyectos electrónicos para crear aplicaciones de detección de obstáculos y medición de distancia, así como en otras aplicaciones.

Aquí hemos traído el método simple para medir la distancia con arduino y sensor ultrasónico y cómo usar el sensor ultrasónico con arduino.



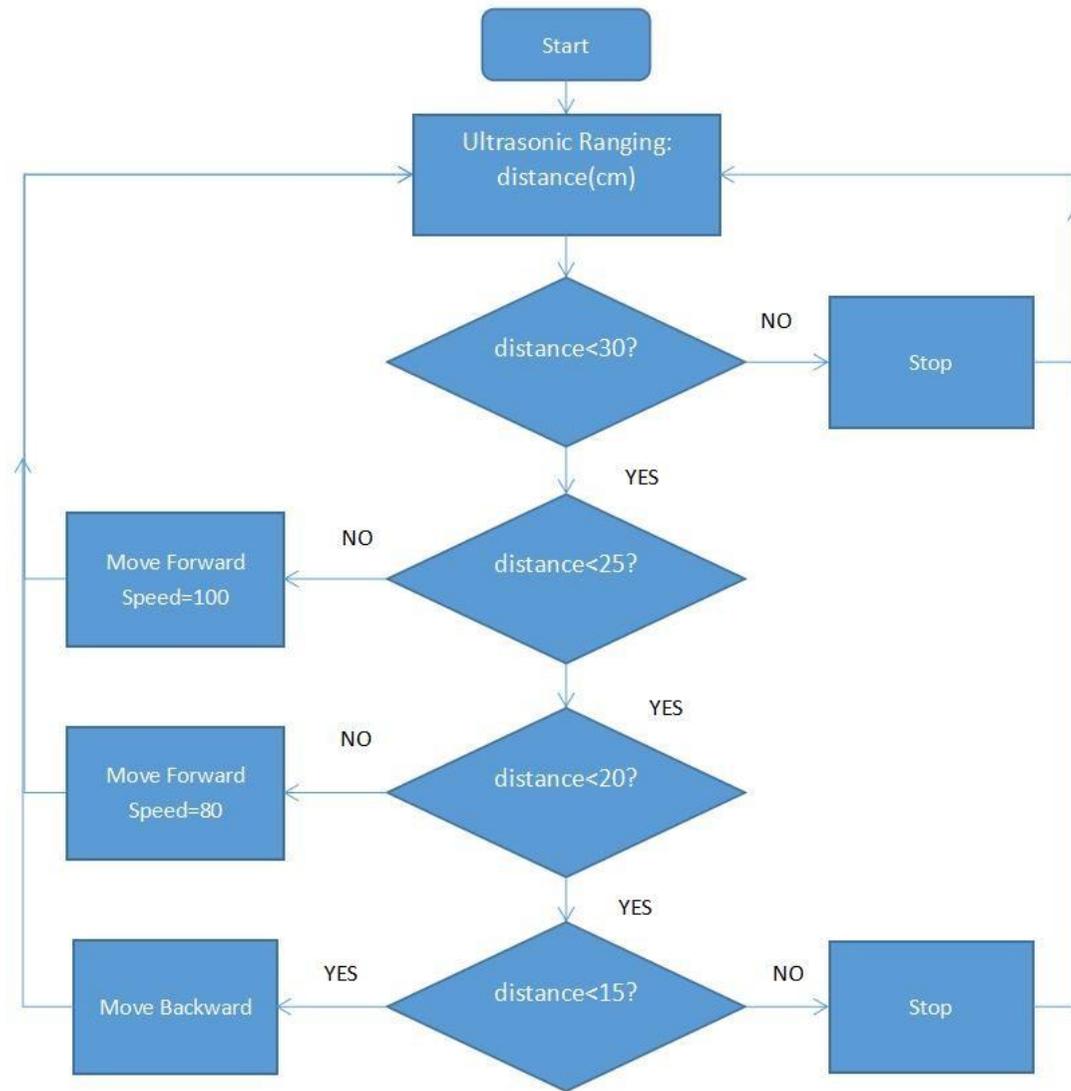
Como Conectar el Circuito.



Vamos al programa

Prueba 1—Vehículo Robot seguimiento Ultrasónico.

Cuando el objeto guiado está directamente en frente del coche robot, el sensor ultrasónico puede detectar la distancia entre el coche robot y el objeto guía. Cuando la distancia sea superior a 20 mm, controle el coche robot para acercarse al guía; Cuando la distancia es inferior a 15 mm, controle el coche robot para que se mueva hacia atrás y mantenga una distancia adecuada, $15 \text{ mm} < < 20 \text{ mm}$.



Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en
Arduino_Code>Lesson_7>Test_1_Ultrasonic_Follow_Robot_Car.ino.

¿Qué verás?

Cargue el código en la placa de control UNO R3 y encienda el interruptor POWER.



El coche inteligente seguirá el obstáculo para moverse a lo largo de la línea recta, pero no puede hacer un giro.

Lección 8 Vehículo robot de control remoto infrarrojo

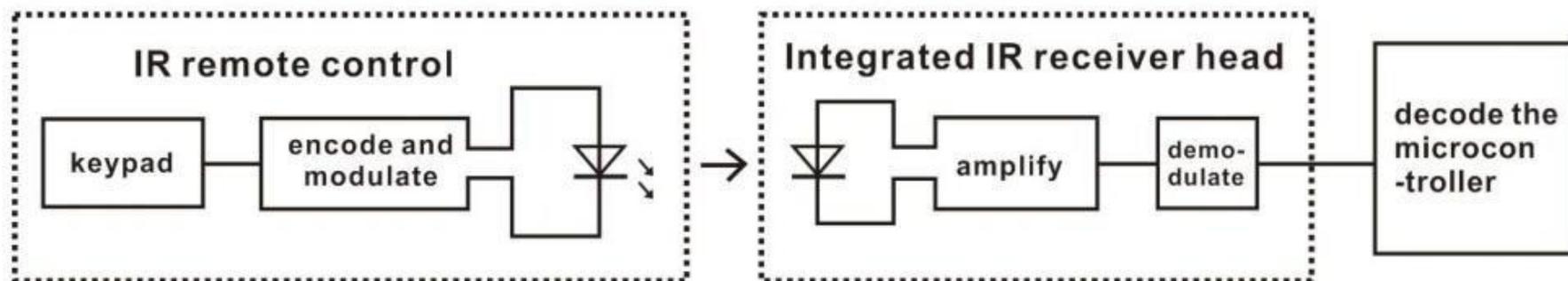
Resumen

En esta lección, aprenderemos el control remoto infrarrojo y usaremos el control remoto infrarrojo para controlar el automóvil robot hacia adelante, hacia atrás, girará a la izquierda, girará a la derecha. Al mismo tiempo, el control remoto infrarrojo puede controlar el brazo robótico.



¿Qué es un mando a distancia por infrarrojos?

No hay duda de que el control remoto infrarrojo se ve comúnmente en nuestra vida diaria. Es difícil imaginar nuestro mundo sin ella. Un control remoto infrarrojo se puede utilizar para controlar una amplia gama de electrodomésticos como televisión, audio, grabadoras de video y receptores de señal satelital. Bueno, a continuación, vamos a obtener una mejor comprensión del control remoto infrarrojo. El control remoto por infrarrojos se compone de sistemas de transmisión y recepción de infrarrojos. Es decir, consisten en un mando a distancia por infrarrojos, un módulo receptor de infrarrojos y un microcontrolador que puede decodificar. Puede consultar la siguiente figura.



La señal portadora infrarroja de 38K transmitida por un control remoto infrarrojo está codificada por un chip de codificación dentro del control remoto. Se compone de un código piloto, código de usuario, código de datos y código de inversión de datos. El intervalo de tiempo entre pulsos se utiliza para distinguir si se trata de una señal 0 o 1. (cuando la relación entre el nivel alto y el nivel bajo es de aproximadamente 1:1, considerado como señal 0.)

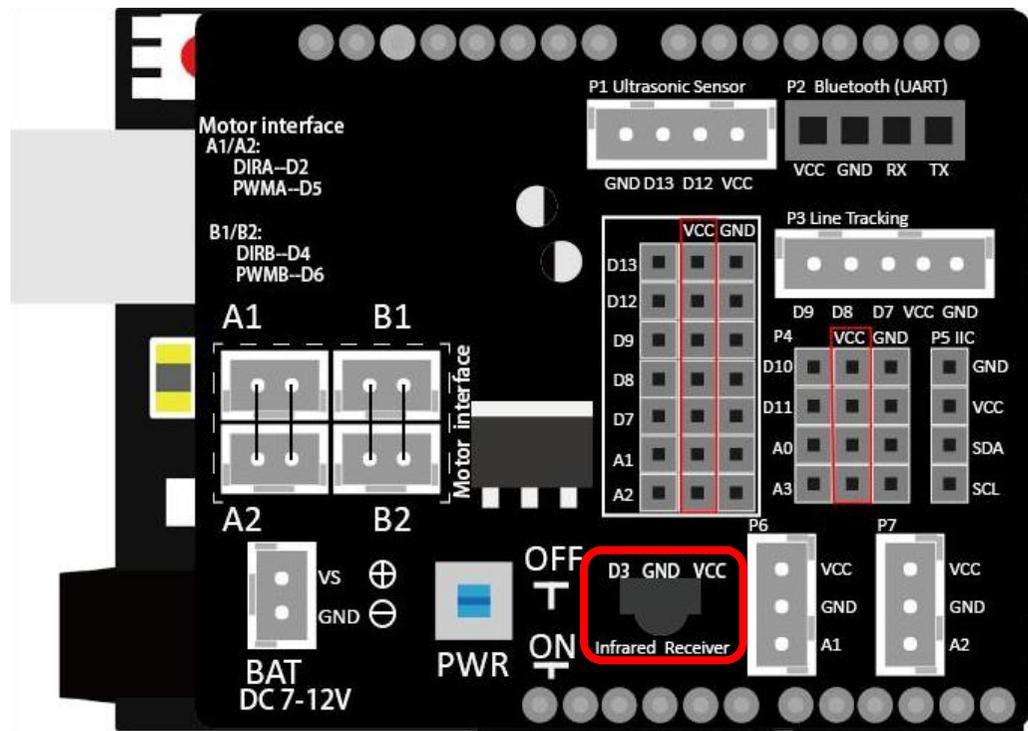
Y la codificación está bien compuesta de señal 0 y 1. El código de usuario del mismo botón en el control remoto no cambia. Usando datos de diferencia, distingue la tecla presionada en el control remoto. Cuando presiona un botón en el control remoto, enviará una señal portadora de infrarrojos. Y cuando el receptor infrarrojo recibe esa señal, su programa decodificará la señal portadora, y a través de diferentes códigos de datos, así puede juzgar qué tecla se presiona.

El microcontrolador es decodificado por una señal recibida 0 o 1 para determinar qué tecla es presionada por el control remoto.

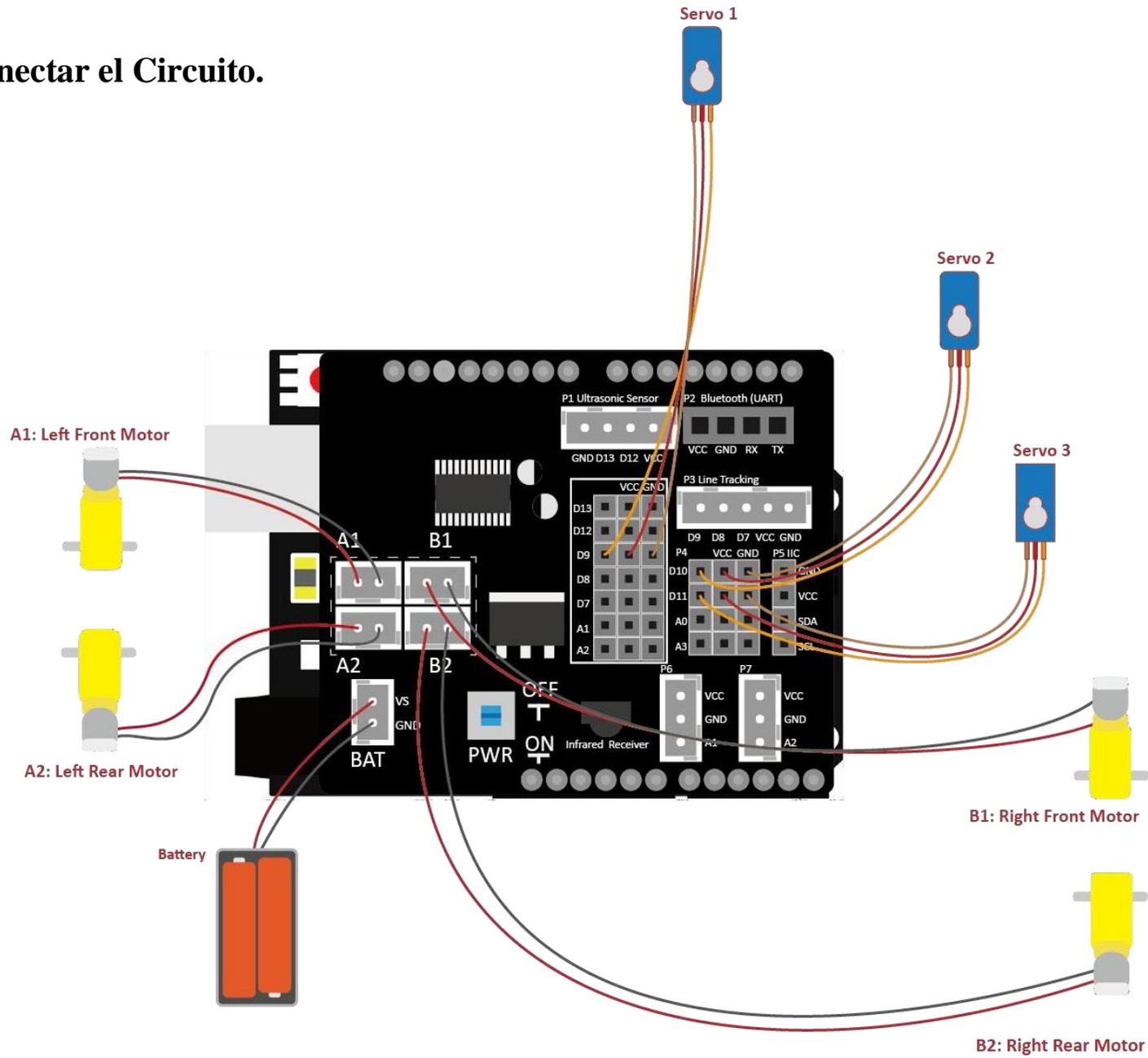
¿Qué es un receptor de infrarrojos?

El escudo del robot viene con un módulo receptor de infrarrojos. Se compone principalmente de un cabezal receptor infrarrojo. Este dispositivo se integra con recepción, amplificación y demodulación. Su inter IC ha sido demodulado, emitiendo señal digital. Adecuado para control remoto IR y transmisión de datos por infrarrojos.

La interfaz de datos del receptor de infrarrojos está conectada al puerto de E/S digital D3.



Como Conectar el Circuito.



Vamos al programa

Prueba 1-- Brazo robótico de control remoto infrarrojo.

En la prueba experimental 1, aprenda a recibir señales de control remoto infrarrojo y distinga el valor de la clave de control remoto correspondiente, y finalmente realice el automóvil robot de control remoto infrarrojo.

Pensando en la programación

Introduzca el archivo de biblioteca de control remoto por infrarrojos y el archivo de biblioteca de valores clave, de lo contrario, la compilación informará de un error.

```
#include "IR_remote.h"
```

```
#include "keymap.h"
```

Este bloque de programa es para inicializar el receptor de infrarrojos, y el número de puerto se llena con "3".

La interfaz de datos del receptor de infrarrojos está conectada al puerto de E/S digital D3.

```
IRremote ir(3);
```

Esta oración juzga si presionar el botón de ajuste, si se presiona el botón "UP", controlar el automóvil para avanzar. Si desea establecer otros botones, solo necesita modificar el valor del botón "IR_KEYCODE_UP"

```
if (ir.getIrKey(ir.getCode(),1) == IR_KEYCODE_UP)
{
    Move_Forward(100);
    delay(300);
    Stop();
}
```

Código clave de botones del mando a distancia:

Key					
Key value	IR_KEYCODE_UP	IR_KEYCODE_DOWN	IR_KEYCODE_LEFT	IR_KEYCODE_RIGHT	IR_KEYCODE_OK
Key					
Key value	IR_KEYCODE_1	IR_KEYCODE_2	IR_KEYCODE_3	IR_KEYCODE_4	IR_KEYCODE_5
Key					
Key value	IR_KEYCODE_6	IR_KEYCODE_7	IR_KEYCODE_8	IR_KEYCODE_9	IR_KEYCODE_STAR
Key					
Key value	IR_KEYCODE_0	IR_KEYCODE_POUND			

Código Arduino

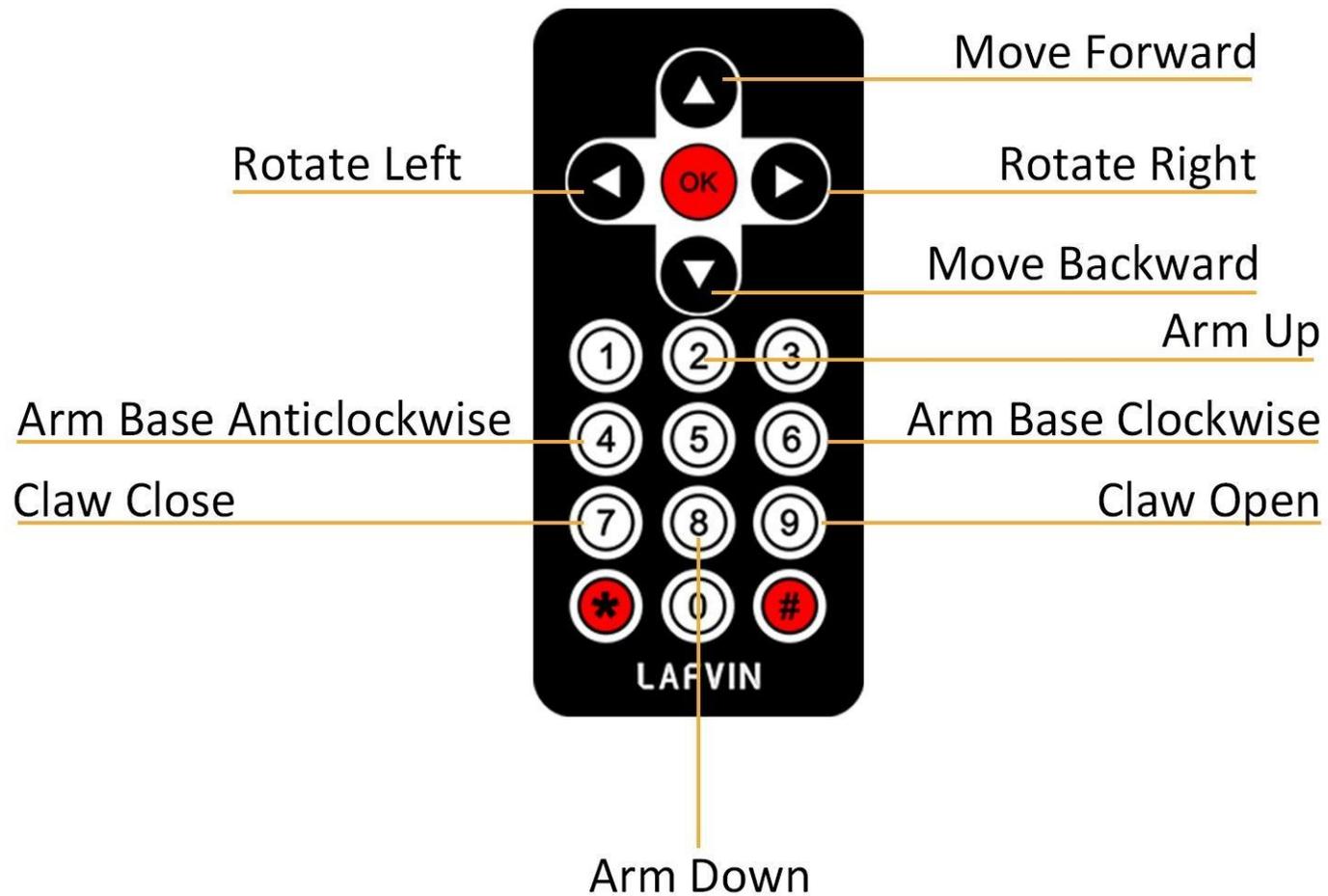
Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_8>**Test_1_Infrared_Remote_Control_Robot_Arm_Smart_Car.ino.**

¿Qué verás?

Cargue el código en el tablero de control UNO R3 y encienda el interruptor POWER. Utilice el mando a distancia para controlar la dirección del coche robot y el servo del brazo robótico.

Nota: Debe mantener presionado el botón del control remoto infrarrojo para mantener el motor en movimiento. El motor dejará de moverse cuando se suelte el botón.



Lección 9 7-en-1 Brazo robótico multifunción Vehículo Inteligente

Resumen

Si desea experimentar todas las funciones del coche robot directamente, puede omitir las lecciones anteriores y consultar directamente los pasos de esta lección.

En esta lección, estudiamos principalmente cómo usar la aplicación para controlar el automóvil inteligente robot.

En la prueba 1, aprendemos la comunicación entre el módulo Bluetooth y el teléfono móvil.

En la prueba 2, podemos cambiar libremente los diversos modos funcionales a través de la aplicación del teléfono móvil, como seguimiento de línea, infrarrojo ultrasónico evitar obstáculos, control remoto infrarrojo, búsqueda de luz, sígueme, control remoto, control remoto del sensor de gravedad.

Follow Me

Line Tracking

Gravity sensor

Auto Carry

Ultrasound Avoid Obstacles

Anti-Drop

Developed by LAFVIN
only support android

4WD Robot Arm Smart Car

Actions

Speed

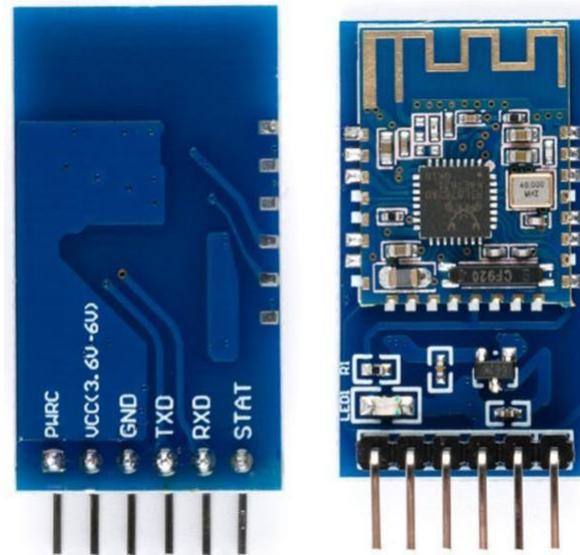
Depth Sensor Line Tracking Anti-Drop Line Tracking Following

¿Qué es un módulo Bluetooth?

Módulo Bluetooth JDY-16

El módulo de transmisión transparente JDY-16 se basa en el estándar de protocolo 4.2 compatible con Bluetooth, la banda de frecuencia de trabajo es el rango de 2.4GHZ, el método de modulación es GFSK, la potencia máxima de transmisión es 0db, la distancia máxima de transmisión es de 60 metros, y adopta el diseño de chip original importado y admite a los usuarios modificar el dispositivo a través de comandos AT Nombre, UUID de servicio, potencia de transmisión, contraseña de emparejamiento y otras instrucciones son convenientes y flexibles de usar.

El módulo compatible con Bluetooth JDY-16 puede realizar la transmisión de datos entre el módulo y el teléfono móvil o entre el módulo y el módulo. El modo de comunicación de UART o IIC se puede seleccionar a través de IO, y el compatible con Bluetooth se puede utilizar rápidamente para la aplicación del producto a través de una configuración simple. Haga que la aplicación de BLE en productos sea más rápida y conveniente.



Parámetros del Producto:

Modelo: JDY-16

Frecuencia de Trabajo: 2.4G

Poder de transmisión: 0db (máximo)

Interface de comunicación: UART o IIC

Voltaje de trabajo: 1.8V-3.6V

Temperatura de Trabajo: -40°C-80°C

Antena: Construir-en PCB antena

Sensitividad de recepción: -97dbm

Distancia de transmisión: 60 metros

Tamaño de módulo: 19.6mm * 14.94 *2.6

Bluetooth-compatible versión: BLE 4.2 (compatible con BLE4.0, BLE4.1)

Velocidad de transmisión transparente: 115200 bps/s

Estado de activación actual: 4mA (Con transmisión)

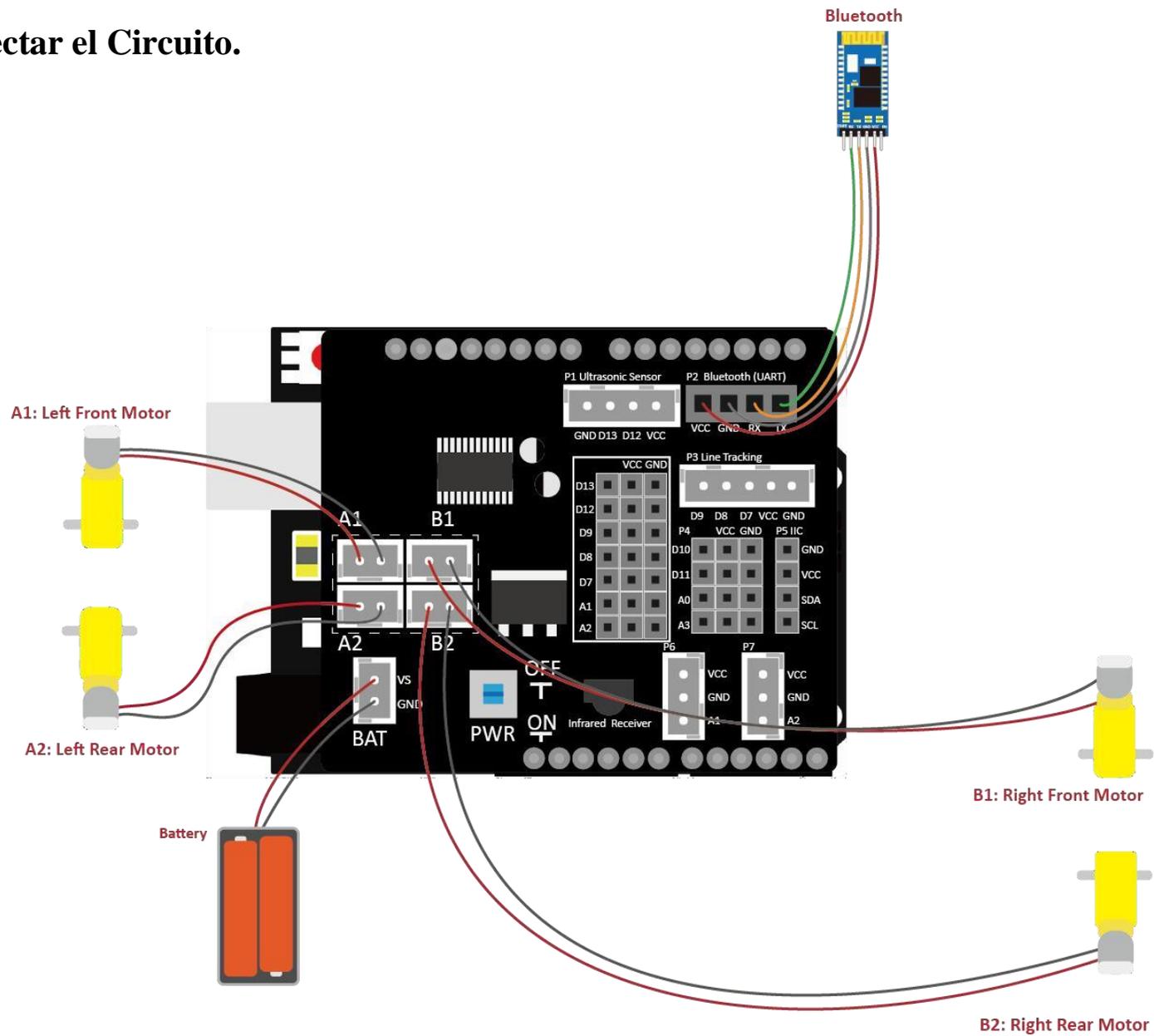
Corriente estado de sueño ligero: <300uA (Con transmisión)

Corriente de sueño profundo: 1.8uA (Sin transmisión)

Guardado de parámetros de comando: los datos de configuración de parámetros se guardan después del apagado

STM Temperatura de soldadura: <300°C

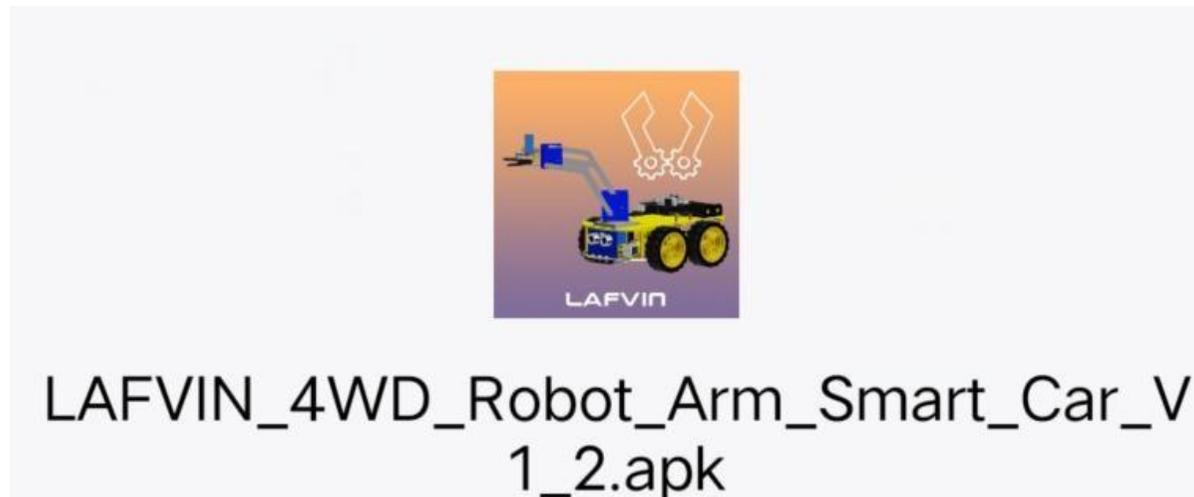
Como Conectar el Circuito.



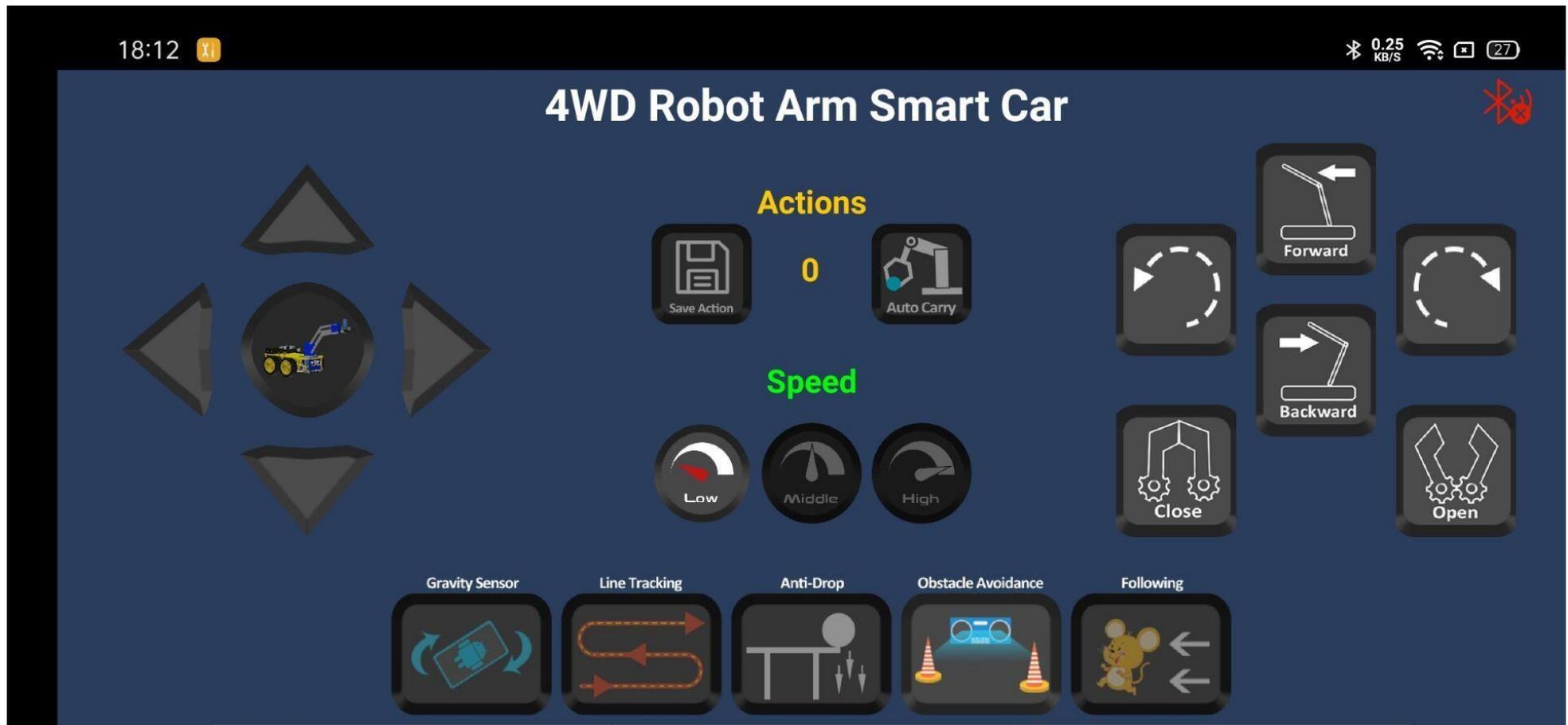
Cómo conectar la aplicación con el módulo Bluetooth.

En primer lugar, copie el archivo "**LAFVIN_4WD_Robot_Arm_Smart_Car_V1_2.apk**" de la carpeta APP en CD a su teléfono móvil e instálelo en un software de aplicación.

El módulo Bluetooth incorporado del teléfono móvil debe ser compatible con Bluetooth 4.0 o superior.

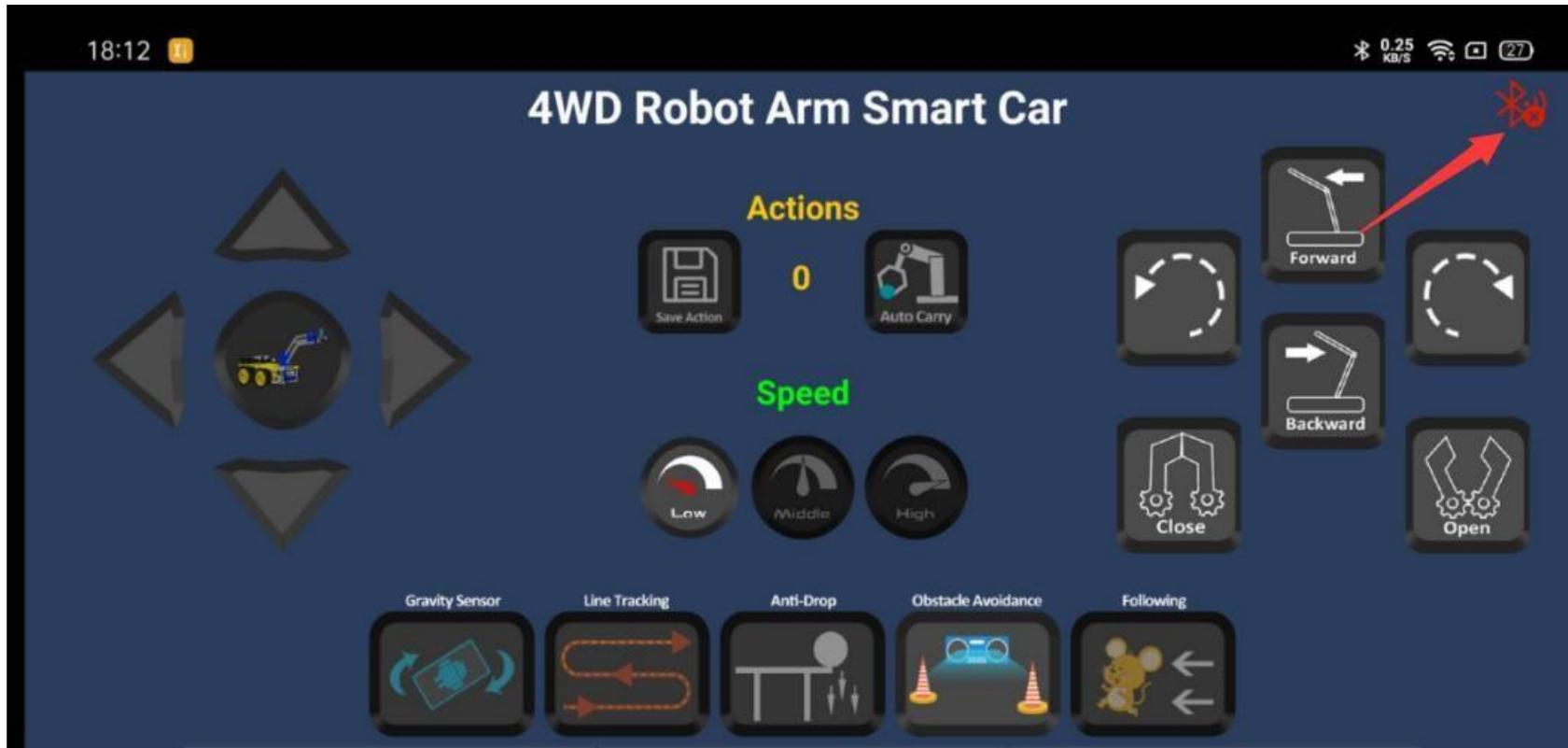


Abra la aplicación, verá la siguiente interfaz de control.



Instale el módulo Bluetooth JDY-16 en el arduino uno Motor Driver Shield, encienda el interruptor de encendido y la luz indicadora del módulo Bluetooth comienza a parpadear una vez en 1.5s, lo que significa que está en un estado conectable.

Haga clic en el icono de Bluetooth  para entrar en la página de emparejamiento automático Bluetooth.



16:35

0.00 KB/S

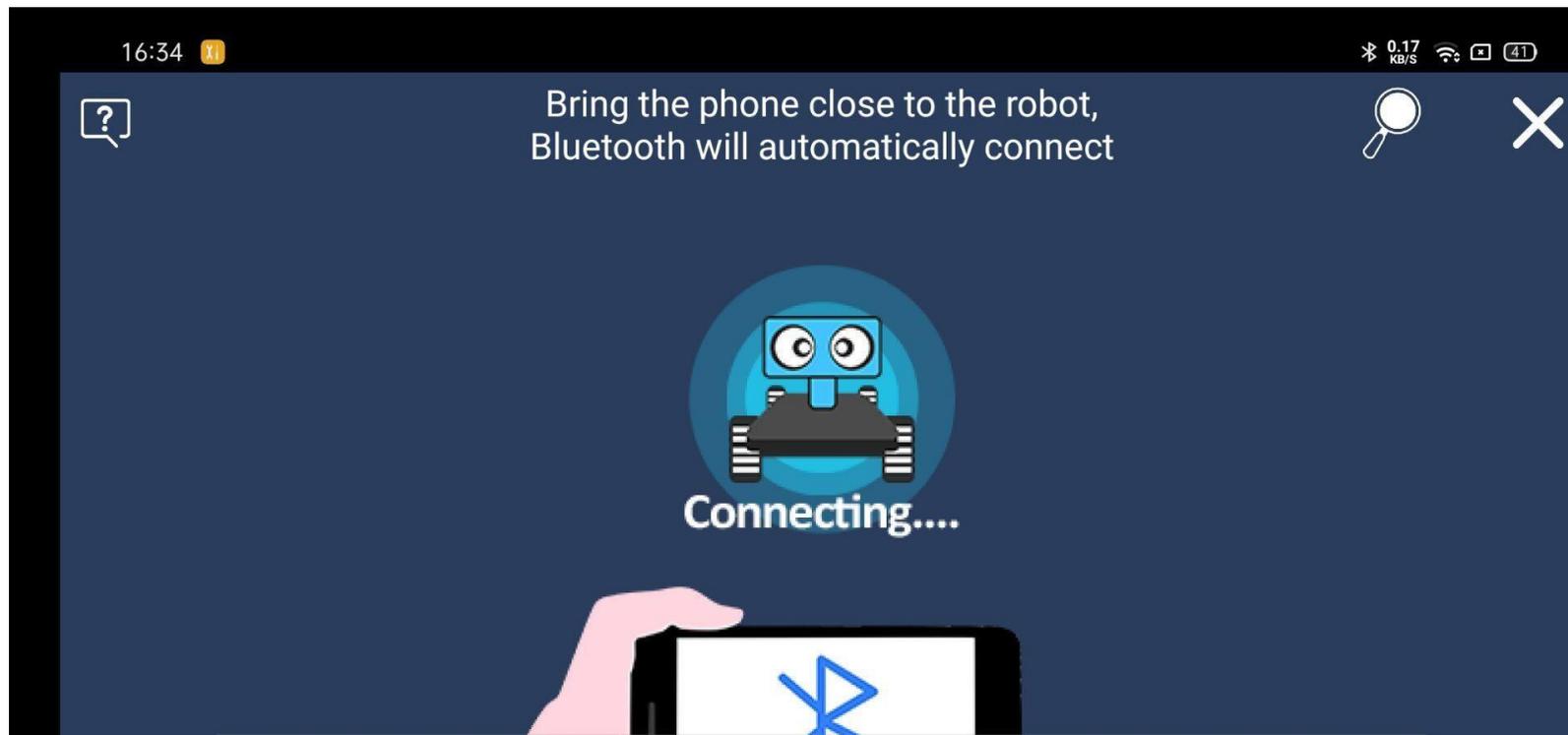


Bring the phone close to the robot,
Bluetooth will automatically connect

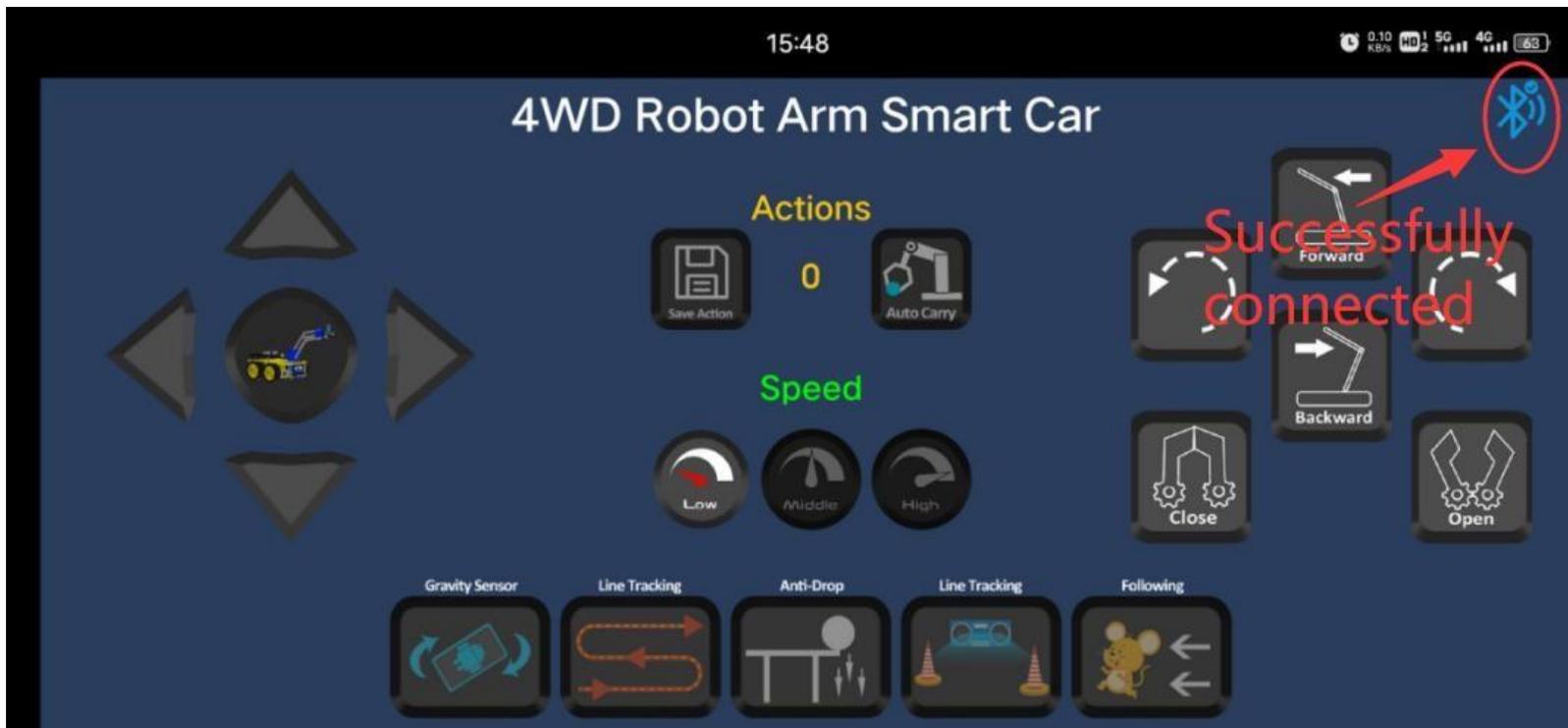


Acerque el teléfono al robot, espere 5 segundos, Bluetooth se conectará automáticamente.

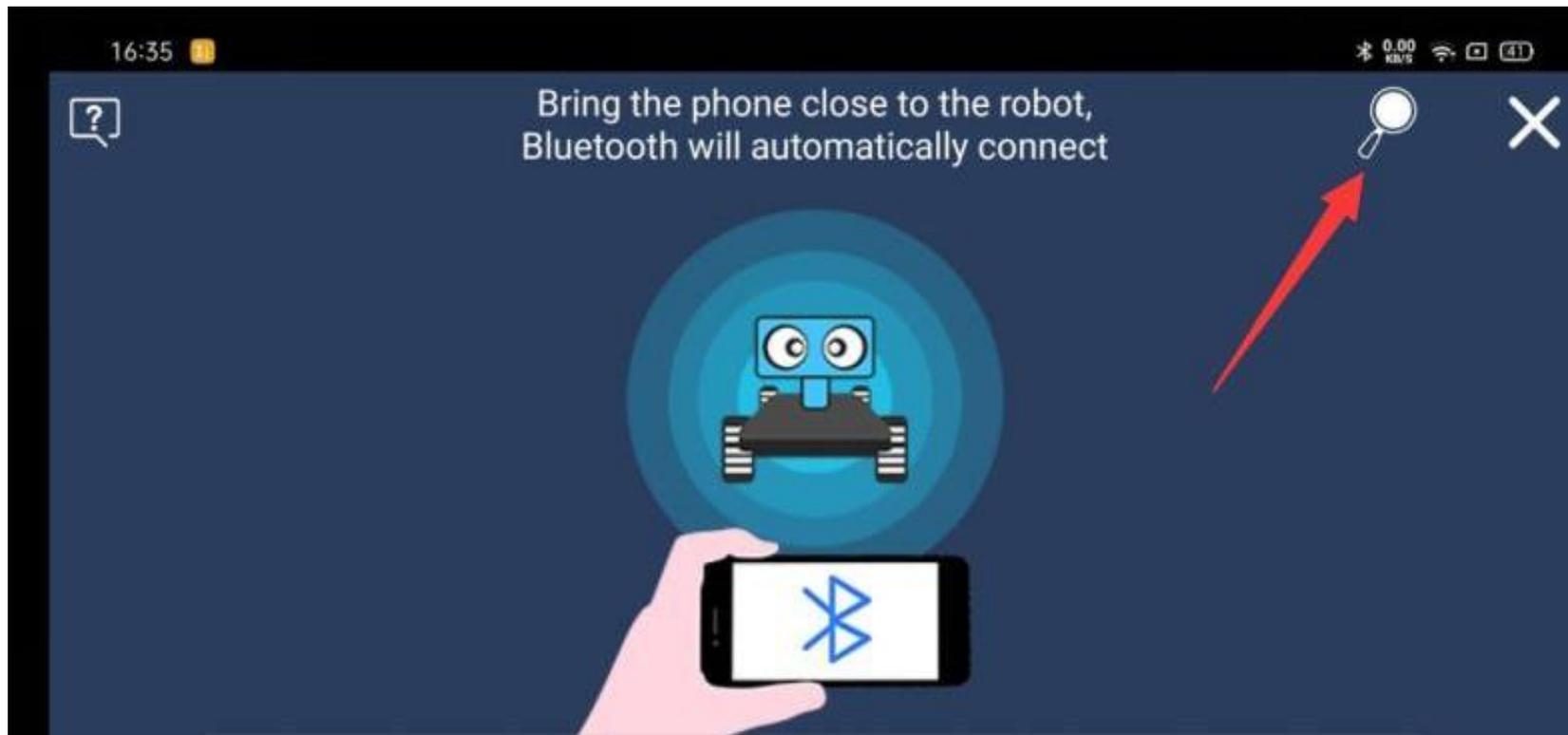
Nota: Si hay varios dispositivos al mismo tiempo, el Bluetooth del teléfono móvil se conectará automáticamente al coche robot que esté más cerca de él. Por lo tanto, para garantizar una conexión más rápida y precisa, se recomienda mover el teléfono móvil cerca del coche robot al que desea conectarse.



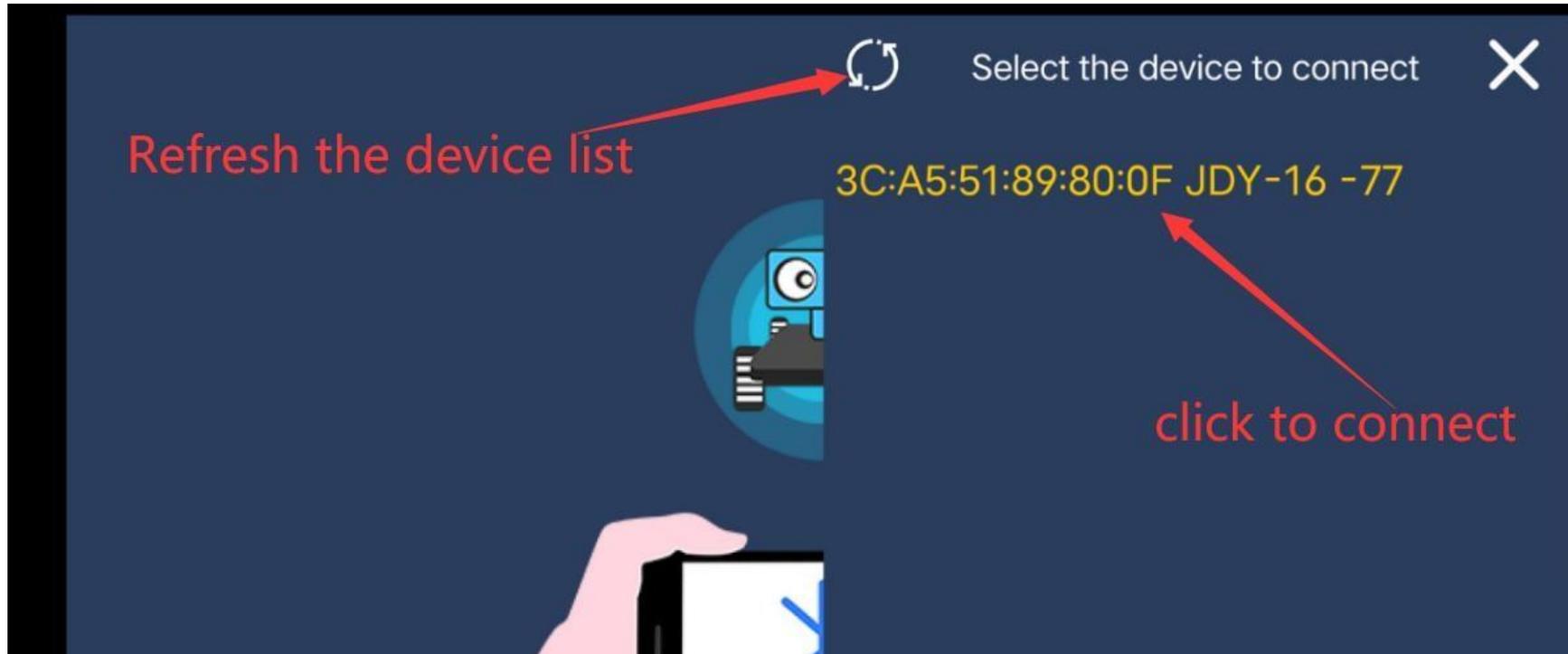
Cuando Bluetooth se conecta correctamente, volverá automáticamente a la interfaz de control de la función inicial.



Si la conexión automática no se conecta, haga clic en el icono de búsqueda en la esquina superior derecha para abrir la lista de dispositivos Bluetooth e intentar conectarse manualmente.



Puede intentar actualizar la lista de dispositivos Bluetooth y hacer clic en el nombre del dispositivo para conectarse manualmente.



Si no puede conectarse automáticamente y no hay Bluetooth en la lista de Bluetooth, extraiga el módulo Bluetooth de Arduino Shield y vuelva a enchufarlo. El indicador LED del módulo bluetooth parpadea una vez en 1,5 segundos, esperando el estado de la conexión, e inténtelo de nuevo.

Vamos al programa

Prueba 1-- El módulo Bluetooth recibe información.

En la Prueba Experimental 1, aprendimos cómo recibir la información enviada por la aplicación del teléfono móvil al módulo Bluetooth JDY-16.

Código Arduino

Si desea consultar el programa que ofrecemos. Abra este programa fuente en

Arduino_Code>Lesson_9>Test_1_Bluetooth_Module_Receives_Information.ino.

Pensando en la Programación

```
String BLE_value;
```

```
void setup(){
```

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

```
BLE_value = "";
```

```
}
```

Establezca la velocidad en baudios para la comunicación serie en 9600. La velocidad de comunicación entre el puerto serie de la placa de control principal de Arduino UNO y el puerto serie de Bluetooth debe ser la misma. La velocidad en baudios de comunicación del módulo Bluetooth JDY-16 es de 9600 por defecto.

```
void loop(){
```

```
    // Si el puerto serie ha recibido nueva información, no se detendrá hasta que se reciban los nuevos datos.
```

```
    while (Serial.available() > 0) {
```

```
        // Combinar los caracteres finales en una cadena
```

```
        BLE_value = BLE_value + ((char)(Serial.read()));
```

```
        // Necesidad de retrasar la espera a que se complete la recepción
```

```
    delay(2);
```

```
    }
```

Si el puerto serie ha recibido nueva información, no se detendrá hasta que se reciban los nuevos datos.

Combina los caracteres finales en una cadena. Necesidad de retrasar la espera a que se complete la recepción.

```
if (0 < String(BLE_val).length() && 2 >= String(BLE_val).length()) { // Si se recibe una nueva cadena de caracteres

    // Imprima la cadena recibida en el monitor serie

    Serial.println(BLE_value);

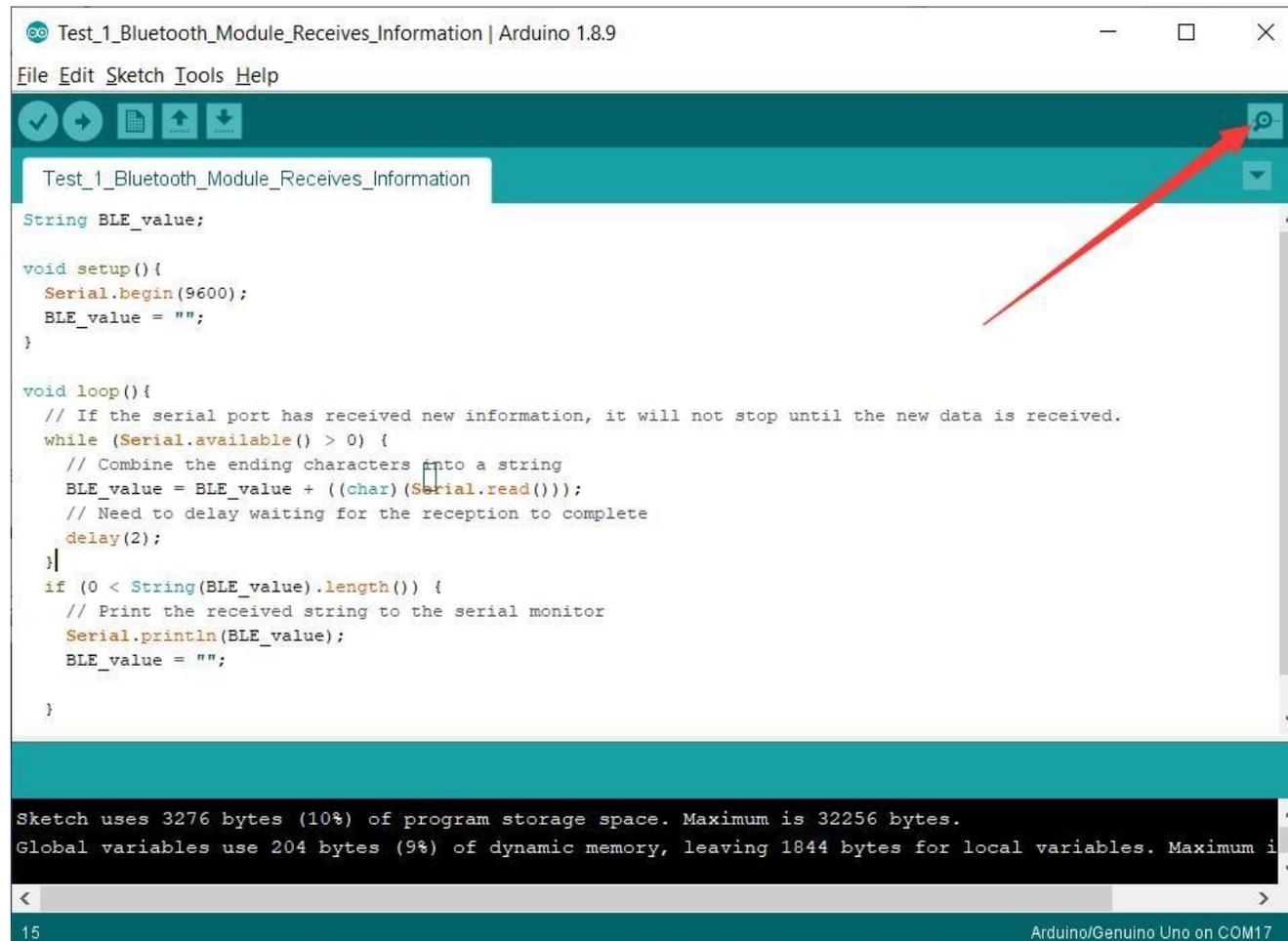
    BLE_value = ""; // Borrar los últimos datos sin prepararse para la próxima recepción.

}

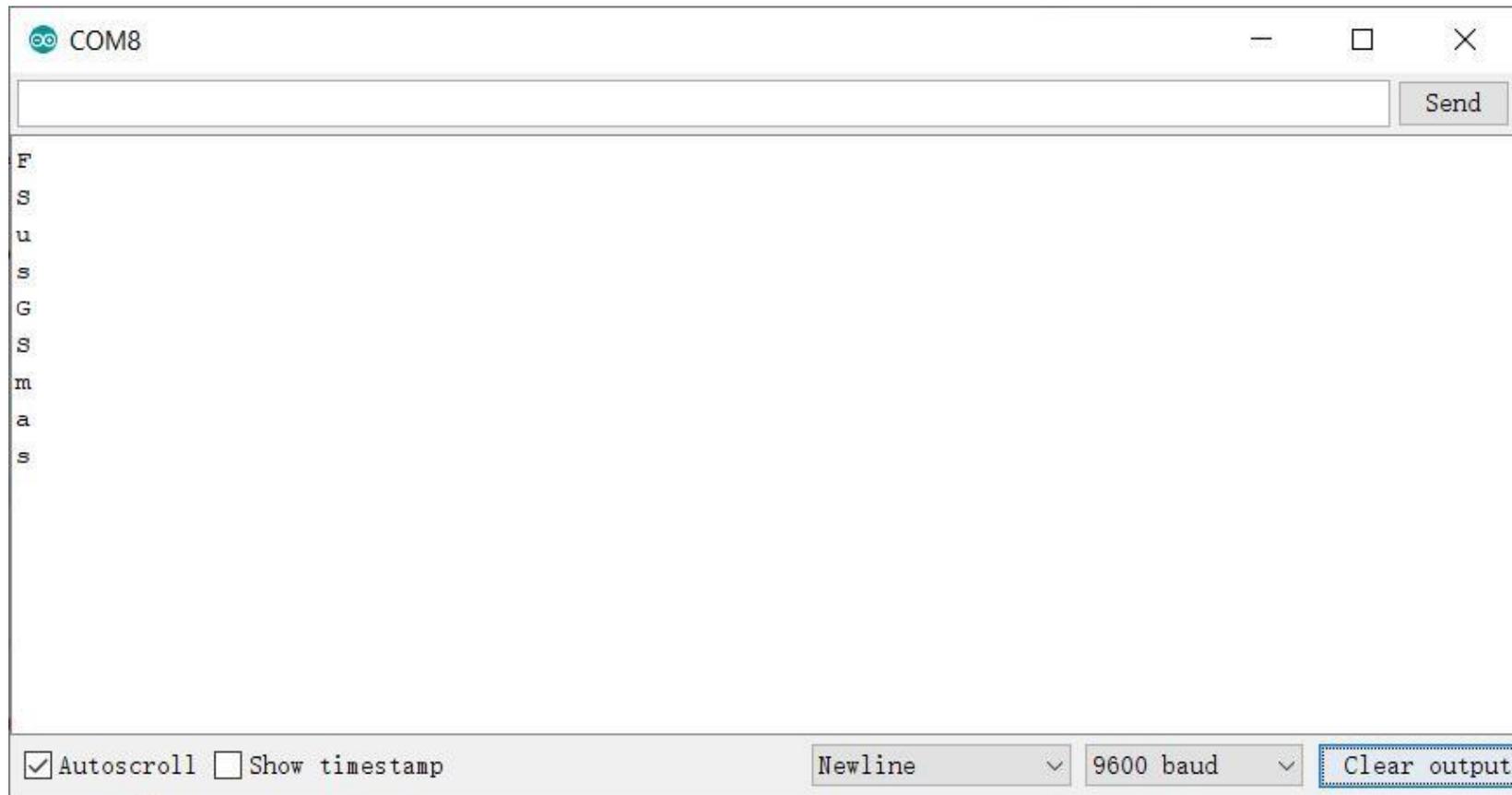
}
```

Si se recibe una nueva cadena de caracteres, imprima la cadena de caracteres recibida en el monitor serie y borre los últimos datos sin prepararse para la siguiente recepción.

Después de completar la carga del programa, abra el monitor serie para ver la información recibida por el módulo Bluetooth.



Entonces puedes ver los datos como lista:



4WD Robot Arm Smart Car

release button:S
long click:

Actions

second click:s

Speed

release button:s
long click:

Gravity Sensor (G)

Line Tracking (T)

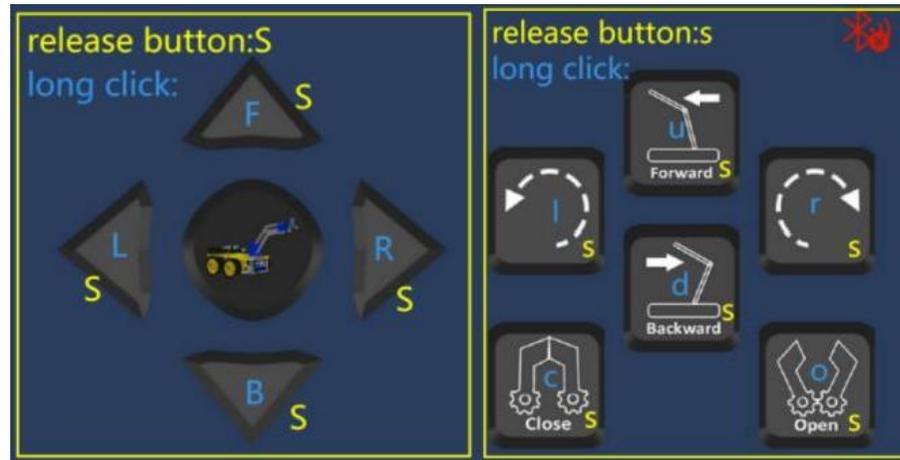
Anti-Drop (D)

Obstade Avoidance (A)

Following (W)

longclick:
second click:S

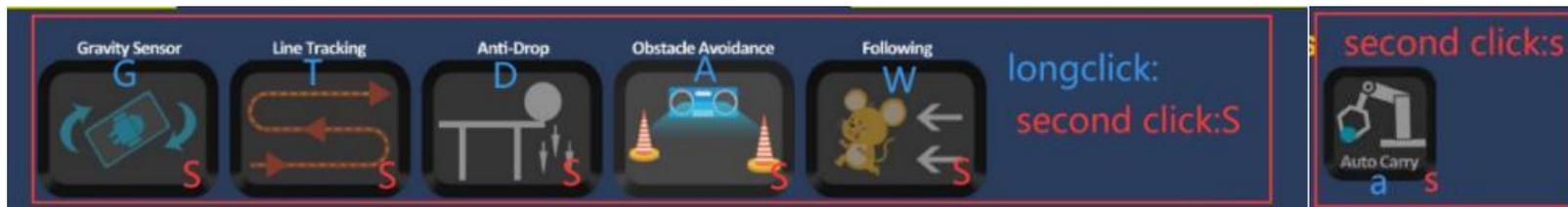
El botón en el cuadro amarillo debe presionarse y el comando es el carácter marcado en azul, y el comando cuando se suelta el botón es el carácter marcado en amarillo.



Por ejemplo, cuando presiona el botón  de la aplicación del teléfono móvil, el comando 'F' se enviará al módulo Bluetooth del coche robot, y cuando el botón  es soltado, el comando 'S' se enviará al módulo Bluetooth del coche robot.

Cuando el botón  de la aplicación móvil es presionado, el comando 'U' se enviará al módulo Bluetooth del coche robot, y cuando el botón  es soltado, el comando 'S' se enviará al módulo Bluetooth del coche robot.

Los botones en el cuadro rojo tienen la máxima autoridad. Cuando se presionan los botones en el cuadro rojo, otros botones se bloquearán y no se podrán usar. Debe hacer clic en el botón nuevamente para desbloquear otros botones. El comando enviado cuando se hace clic en el botón del cuadro rojo por primera vez son los caracteres marcados en azul, y el comando enviado cuando se hace clic en el botón del cuadro rojo por segunda vez sigue siendo los caracteres marcados en rojo.



Por ejemplo, cuando el botón  de la APP es presionado, el comando 'G' enviará al módulo Bluetooth del vehículo Robot. Cuando en botón  se hace clic por segunda vez, el comando 'S' se enviará al módulo Bluetooth del coche robot.

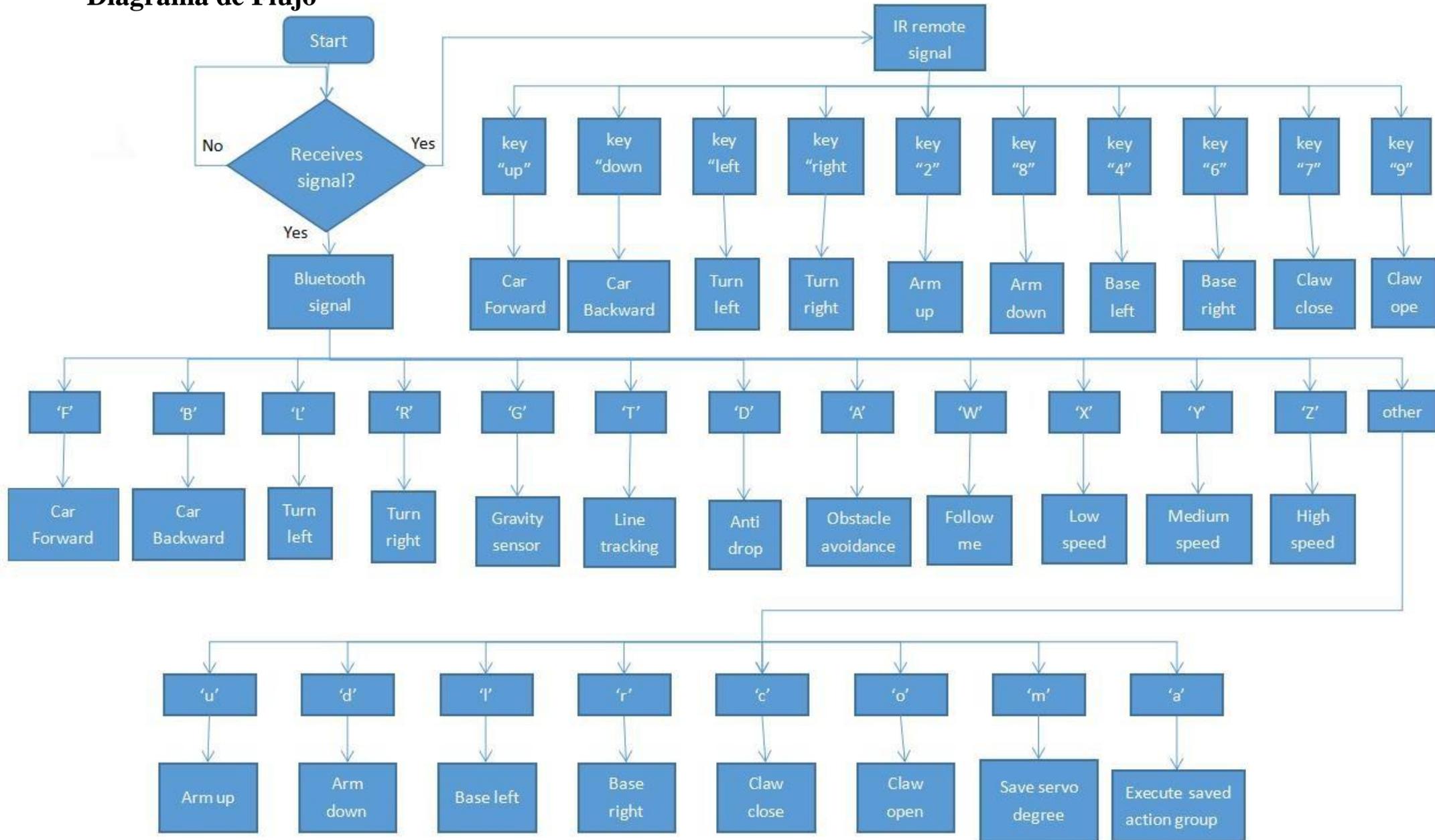
Por ejemplo, cuando el botón  de la aplicación del teléfono móvil se hace clic, el comando 'A' se enviará al módulo Bluetooth del coche robot. Cuando el botón  se hace clic por segunda vez, el comando 'S' se enviará al módulo Bluetooth del coche robot.

7-en-1 Brazo robótico multifunción Smart Car.

En esta prueba experimental, todas las funciones se combinan en el mismo programa, puede usar la aplicación para cambiar libremente entre diferentes funciones. Al mismo tiempo, el control remoto infrarrojo también puede controlar la dirección del coche robot y el grado del servo del brazo robótico.



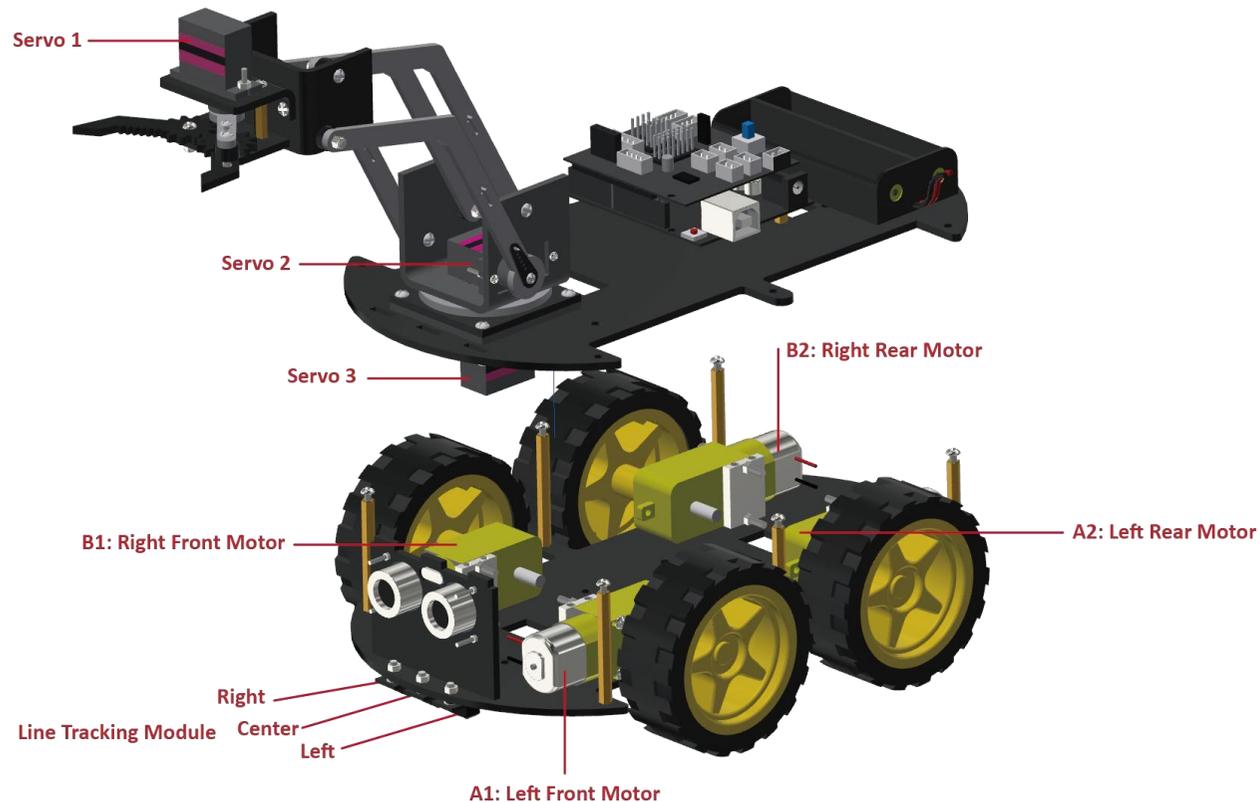
Diagrama de Flujo

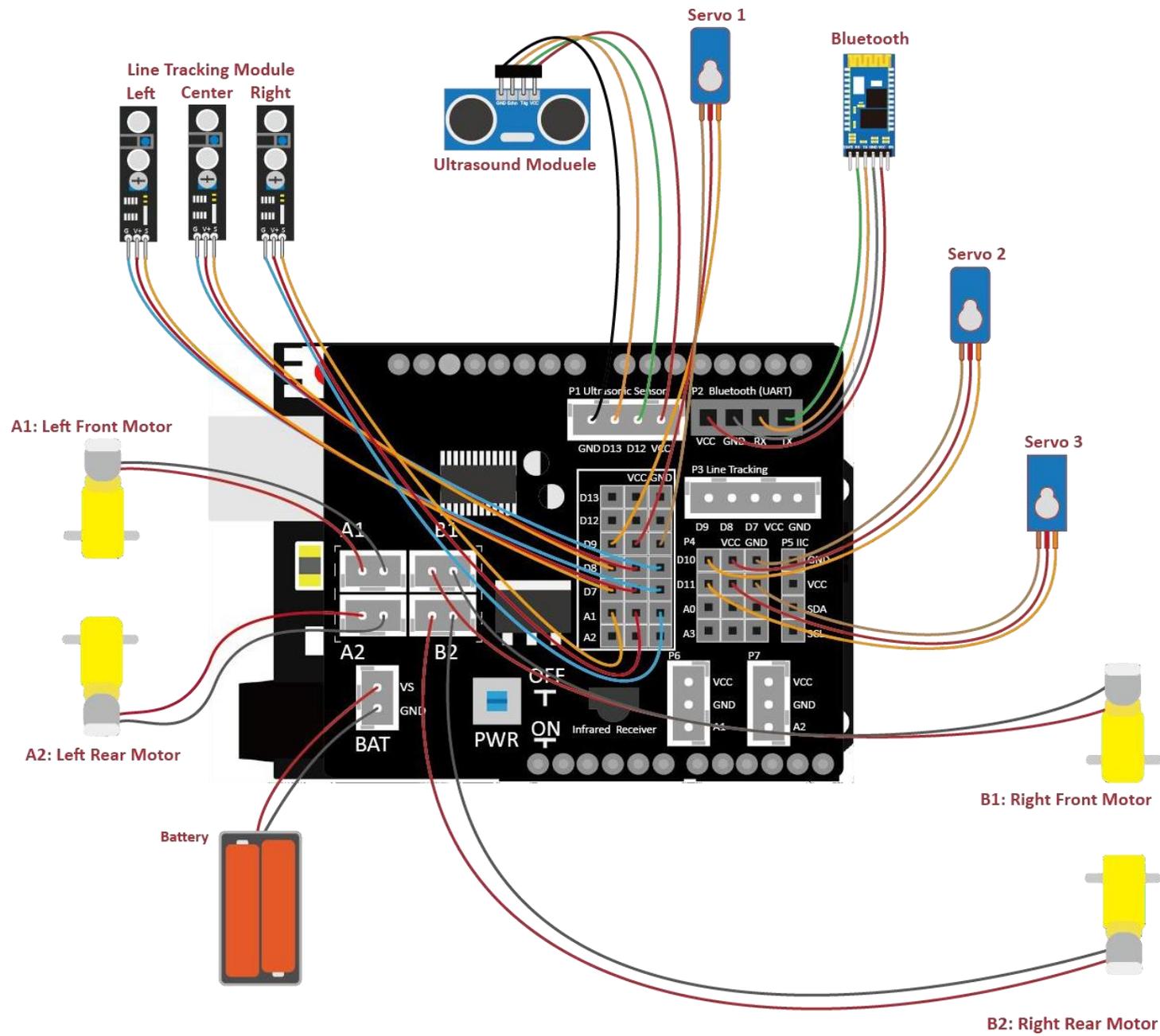


Cómo utilizar la función 7 en 1

Paso 1: Cableado del Circuito

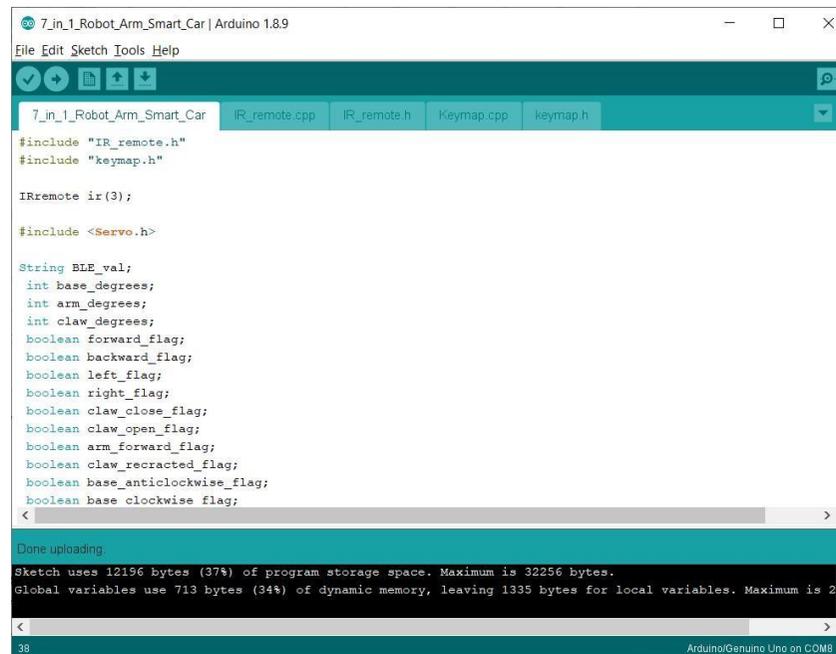
Conecte el cableado de acuerdo con el diagrama de cableado, prestando atención a los polos positivo y negativo de la fuente de alimentación. Distinguir correctamente la posición de instalación del módulo sensor, como el módulo de seguimiento de línea izquierda, central y derecha. A1: Motor delantero izquierdo B1: Motor delantero derecho.





Paso 2: Subir código

Abra este programa fuente en Arduino_Code>Lesson_9>7_in_1_Multi_Function_Robot_Arm_Smart_Car.ino. Sube el programa a arduino UNO. Si informa de un error durante la carga del programa, puede consultar [Lesson 1 Getting Started with Arduino IDE](#).



```
7_in_1_Robot_Arm_Smart_Car | Arduino 1.8.9
File Edit Sketch Tools Help
7_in_1_Robot_Arm_Smart_Car IR_remote.cpp IR_remote.h Keymap.cpp keymap.h
#include "IR_remote.h"
#include "keymap.h"

IRremote ir(3);

#include <Servo.h>

String BLE_val;
int base_degrees;
int arm_degrees;
int claw_degrees;
boolean forward_flag;
boolean backward_flag;
boolean left_flag;
boolean right_flag;
boolean claw_close_flag;
boolean claw_open_flag;
boolean arm_forward_flag;
boolean claw_retracted_flag;
boolean base_anticlockwise_flag;
boolean base_clockwise_flag;
```

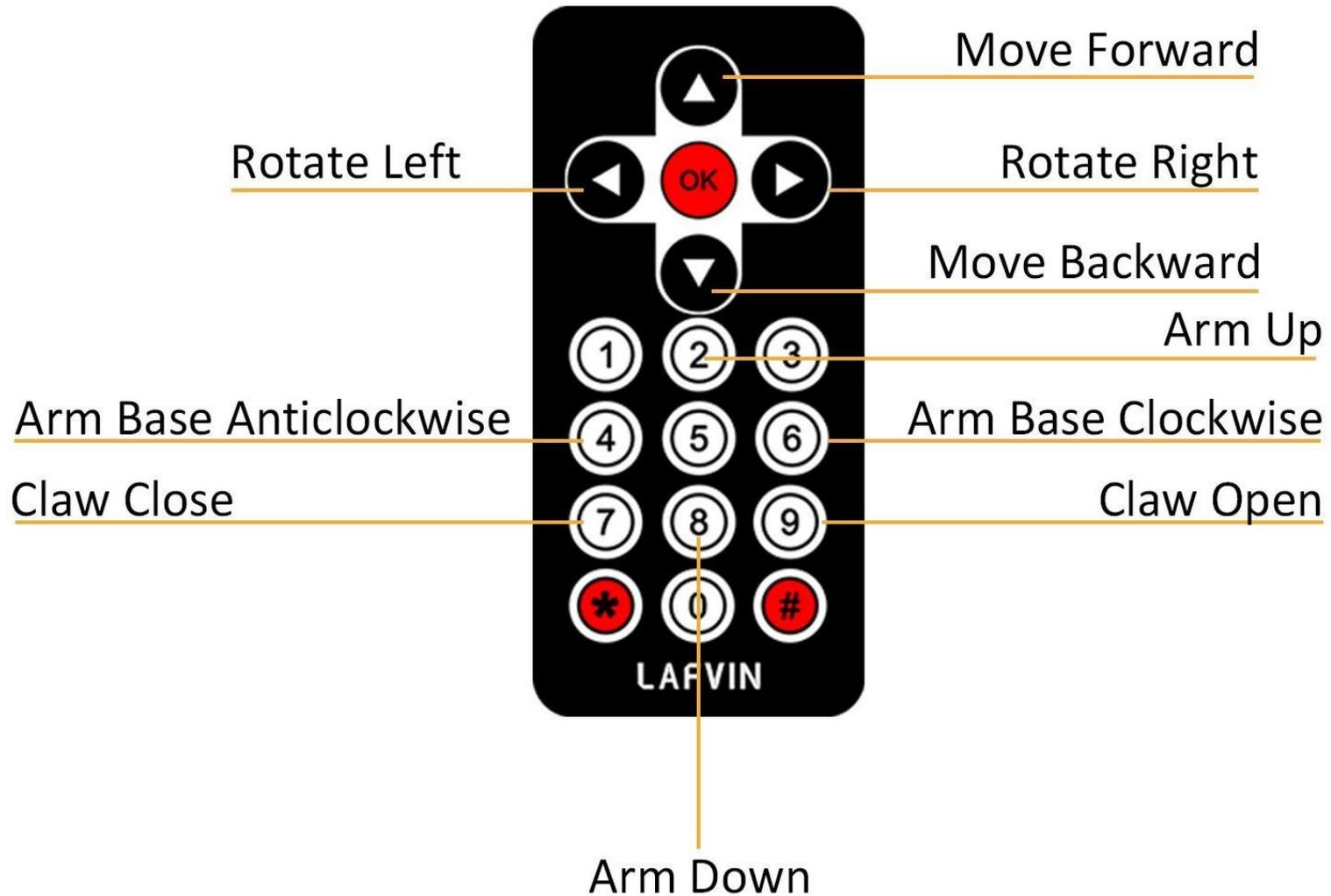
Done uploading.
Sketch uses 12196 bytes (37%) of program storage space. Maximum is 32256 bytes.
Global variables use 713 bytes (34%) of dynamic memory, leaving 1335 bytes for local variables. Maximum is 2048 bytes.

38 Arduino/Genuino Uno on COM8

NOTA: Antes de poder ejecutar esto, asegúrese de que ha instalado la biblioteca <Servo> o vuelva a instalarla, si es necesario. De lo contrario, el código no funcionará. Para obtener más información sobre cómo cargar el archivo de biblioteca, vea [Lección sobre cómo agregar bibliotecas](#).

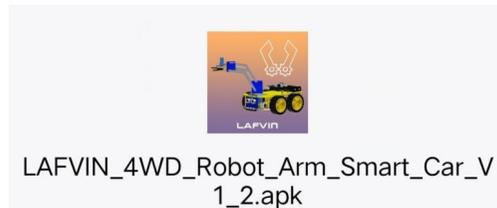
Paso 3: Control Remoto Infrarrojo

Después de cargar con éxito el código, puede usar directamente el control remoto infrarrojo para controlar el automóvil inteligente robot.



Paso 4: Instalar la APP

En primer lugar, copie el archivo "LAFVIN_4WD_Robot_Arm_Smart_Car_V1_2.apk" de la carpeta APP en CD a su teléfono móvil e instálelo en un software de aplicación. **El módulo Bluetooth incorporado del teléfono móvil debe ser compatible con Bluetooth 4.0 o superior.**



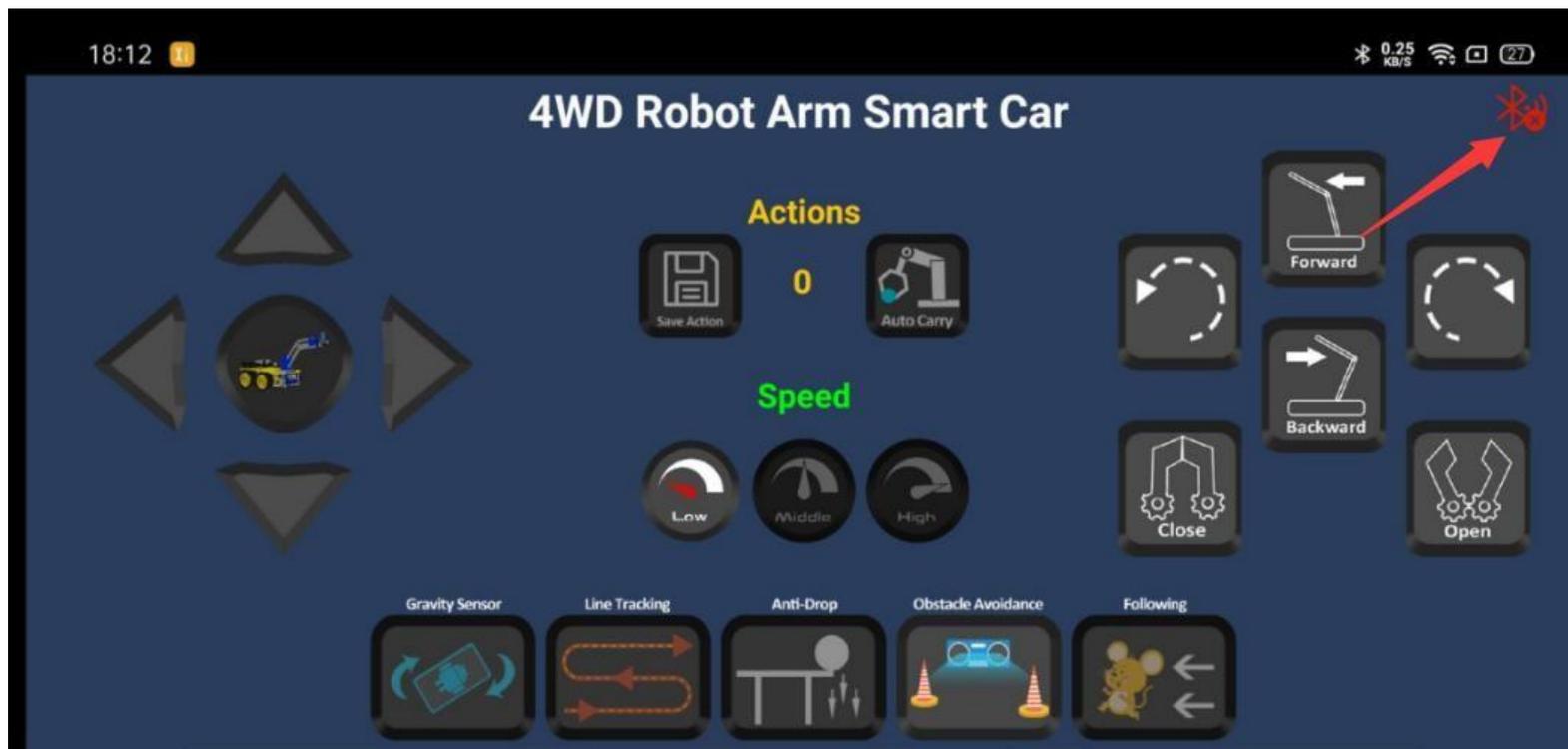
Abra la aplicación, verá la siguiente interfaz de control.

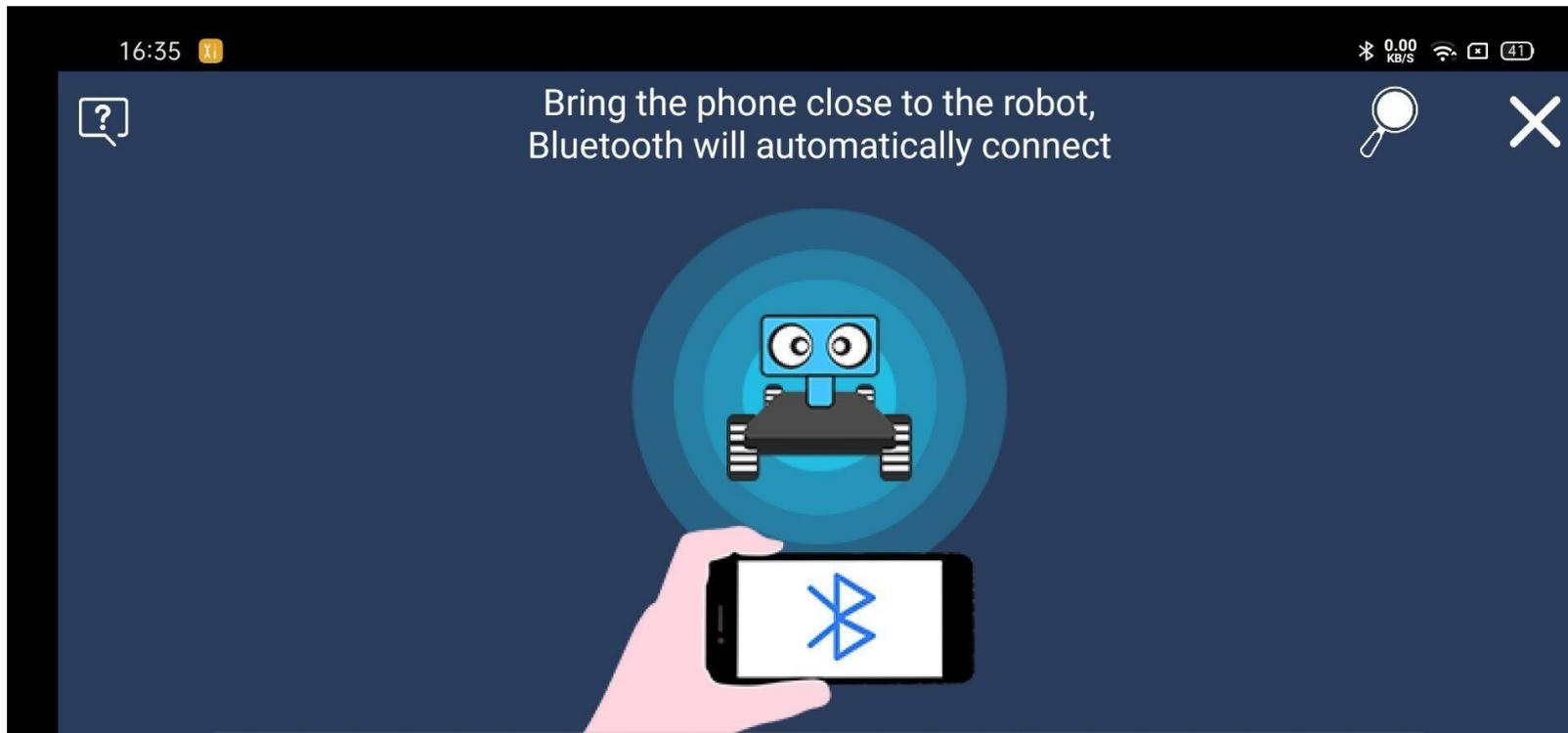


Paso 5: Conexión Bluetooth

Instale el módulo Bluetooth JDY-16 en el Arduino uno Motor Driver Shield, encienda el interruptor de encendido y la luz indicadora del módulo Bluetooth comenzará a parpadear una vez en 1.5s, lo que significa que está en un estado conectable.

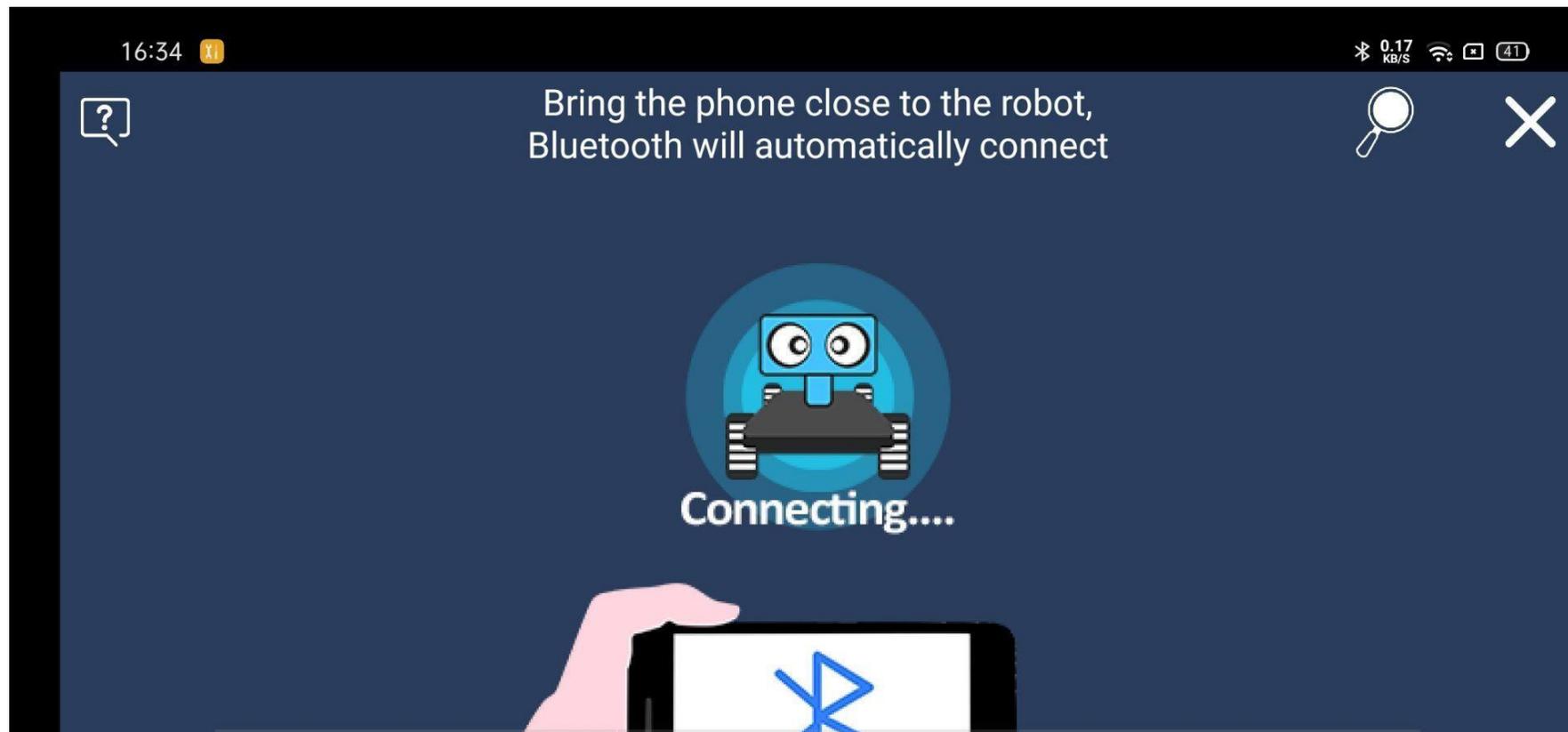
Haga clic en el icono de Bluetooth  para entrar en la página de emparejamiento automático Bluetooth.





Nota: Si hay varios dispositivos al mismo tiempo, el Bluetooth del teléfono móvil se conectará automáticamente al coche robot que esté más cerca de él. Por lo tanto, para garantizar una conexión más rápida y precisa, se recomienda mover el teléfono móvil cerca del coche robot al que desea conectarse.

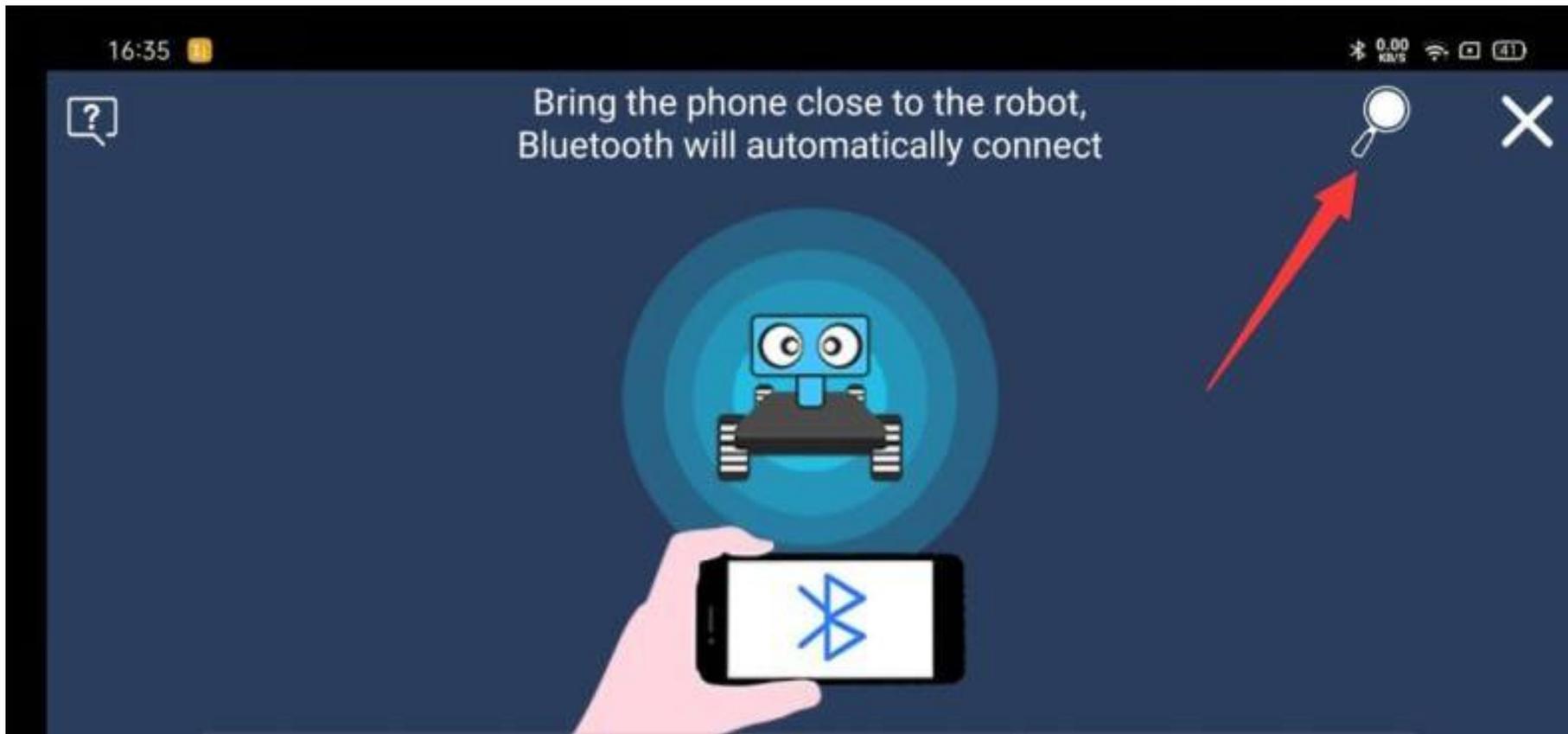
Acerque el teléfono al robot, espere 5 segundos, Bluetooth se conectará automáticamente.



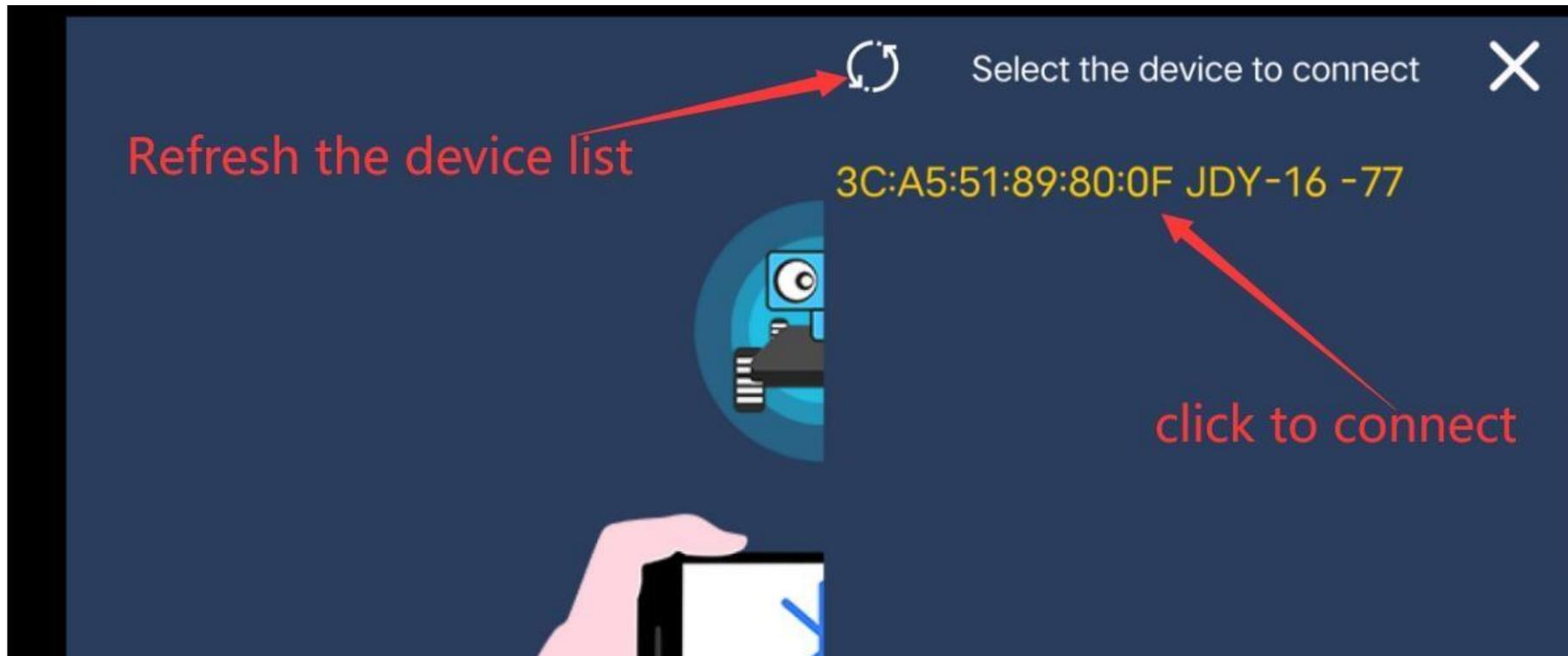
Cuando el Bluetooth se conecte correctamente, volverá automáticamente a la interfaz de control de la función inicial.



Si la conexión automática no se conecta, haga clic en el icono de búsqueda en la esquina superior derecha para abrir la lista de dispositivos Bluetooth e intentar conectarse manualmente.



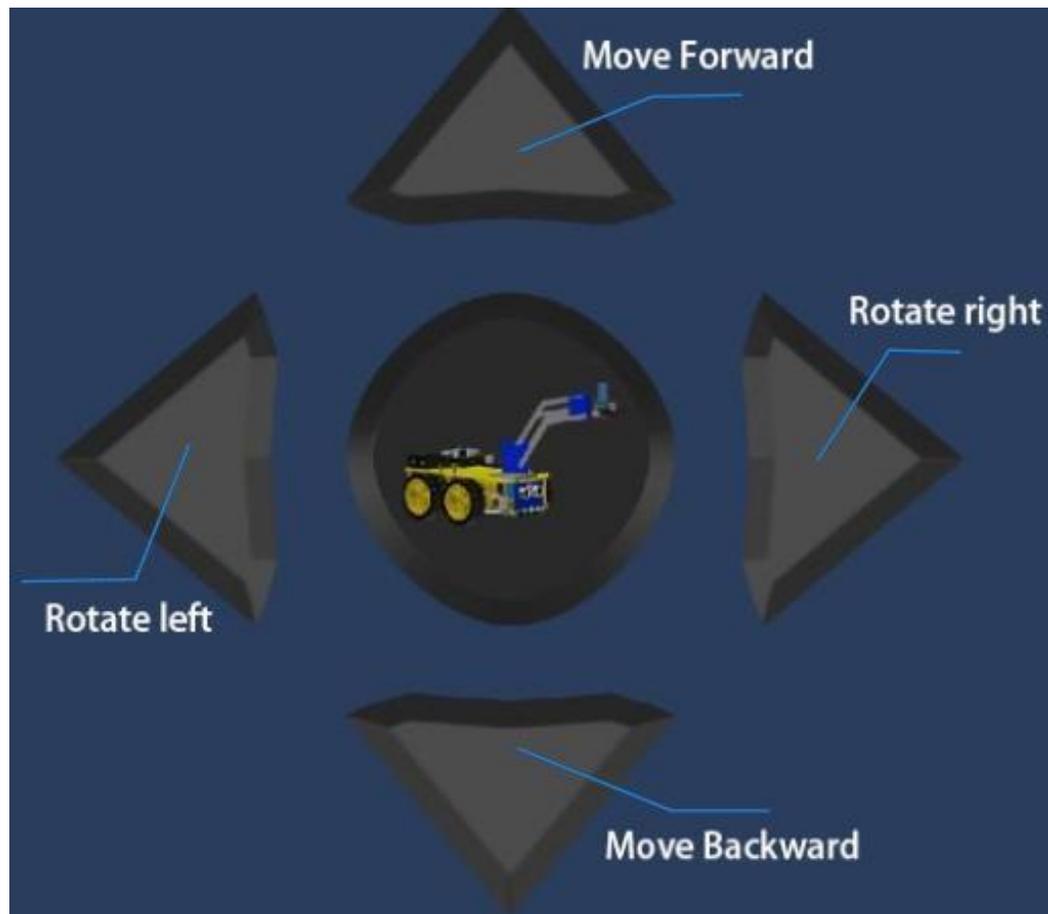
Puede intentar actualizar la lista de dispositivos Bluetooth y hacer clic en el nombre del dispositivo para conectarse manualmente.



Si no puede conectarse automáticamente y no hay Bluetooth en la lista de Bluetooth, extraiga el módulo Bluetooth de Arduino Shield y vuelva a enchufarlo. El indicador LED del módulo bluetooth parpadea una vez en 1,5 segundos, esperando el estado de la conexión, e inténtelo de nuevo.

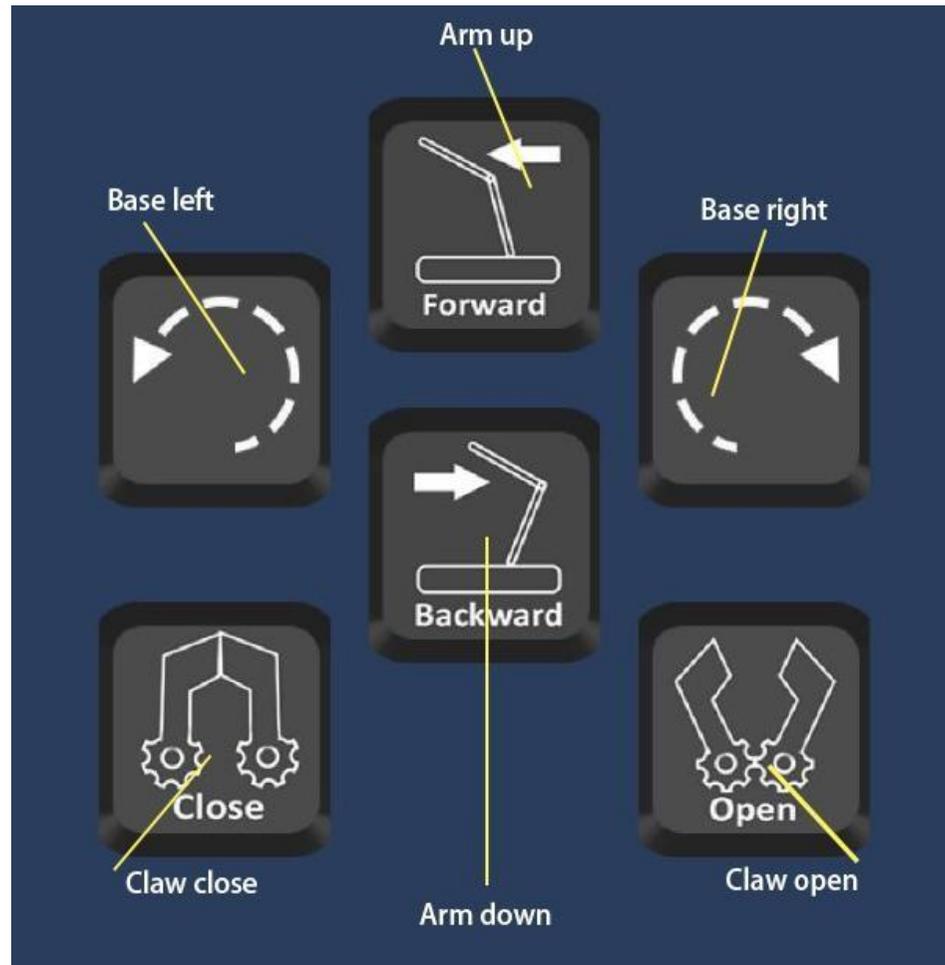
Paso 6: Interface de Control App

Botón de dirección del coche robot: Presione para activar, suelte para detener.



Botones de control del brazo robótico:

Presione para activar, suelte para detener.



Botón de modo multifunción:

Haga clic en el botón de modo para activar la función. Otros botones están bloqueados y no se pueden utilizar. Debe hacer clic nuevamente para salir del modo de función. Los otros botones se desbloquean y se reanuda el uso.



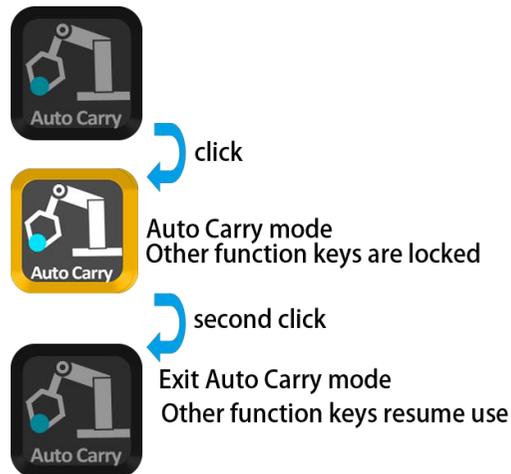
Botón de memoria de acción:



: Haga clic para guardar la acción actual del brazo robótico



: Muestra el número de acciones guardadas, el número máximo de acciones guardadas es 20.



: Haga clic para ejecutar todas las acciones guardadas. Cuando se hace clic en el botón, entrará en el modo de transporte automático. Otros botones de otras aplicaciones están bloqueados y no se pueden usar. Debe hacer clic nuevamente para salir del modo de ejecución automática, y los otros botones se desbloquearán para reanudar el uso.

Botón de control de velocidad del coche robot:

Haga clic en el botón para cambiar la velocidad

