



ASIGNATURA:

Organización y diseño de computadora

TEMA:

Microcomputadora Arduino & Raspberry Pi

TRABAJO A REALIZAR:

Creación de un proyecto con una microcomputadora

con Arduino o Raspberry Pi.

Deben entregar documentación en PDF que incluya (material, proceso de creación del proyecto)

El proyecto será evaluado en el aula.

Se calificará: Funcionalidad y presentación.

Se evaluará a cada integrante con preguntas relacionadas con el funcionamiento del proyecto, los componentes y sus características.

NOMBRES DEL EQUIPO:

1.- PÉREZ BARAHONA PEDRO LUIS

2.- Salvador Carrion Angel Alberto

3.- JAVIER ADRIAN ONOFRE CHABLE

4.- Hernandez Chavarria Jesus Emiliano

5.-ALEJANDRO SALES MENDOZA

6.- HAU PUC ERICK ANTONIO

7.- BRYAN YAHIR COLORADO SALAZAR

8.- ARAM ELIACID MENDEZ MARTINEZ

9.- NAHUAT CEME JESUS RAFAEL

10.- VICENTE ALAMILLA ANDRES

11.- JAVIER ISAAC ROMERO CAMPO

12.- ISAI LOPEZ BROCA

14.- NAHUAT PALOMERA JORGE LUIS

PROGRAMA EDUCATIVO:

(INGENIERÍA EN DATOS E INTELIGENCIA ORGANIZACIONAL)

PRESENTADO A: ISMAEL JIMÉNEZ SÁNCHEZ

"El proyecto final tiene como objetivo el uso y la comprensión de una microcomputadora, en este caso, se utilizó un Arduino básico, modelo R3, para facilitar su uso en un carro robot. Su función es seguir trayectorias marcadas y utilizar un sensor ultrasónico para detectar objetos a una distancia de aproximadamente 25 cm. Esto permite tomar decisiones sobre rutas alternativas más adecuadas para avanzar de la mejor manera posible.

Aunque el proyecto es sencillo en su concepción, su nivel de complejidad radica en la ejecución del código, tanto para el control remoto, el control desde un celular como en su funcionamiento independiente con el sensor ultrasónico. Dejaré un enlace al DRIVE que contiene la documentación."

Robot kit arduino con componentes Arduino,modulo tracking line, modulo Bluetooth,Modulo infrarrojo,servomotors de las llantas,sensor ultrasonico, (4) cables, conectores:\$3999

Bateria recargable de 3.7 volt \$100 cada una

Cargador de baterias \$100

Tipo de proyecto.

Introducción

Tarjeta Arduino utilizada.

Componentes del Arduino.

Propósito y modo de empleo.

Sensor hipersónico.

Control mediante control remoto y celular.

Funcionamiento del control automático.

Las microcomputadoras Arduino y Raspberry Pi son plataformas de hardware diseñadas para proyectos de electrónica y computación en el ámbito de la informática física y el desarrollo de sistemas embebidos. Ambas son herramientas populares en la comunidad de aficionados, educadores y profesionales que trabajan en el campo de la electrónica y la programación.

Arduino:

Arduino es una plataforma de desarrollo de hardware de código abierto que utiliza microcontroladores para realizar tareas específicas.

Está orientado a principiantes y usuarios intermedios que desean crear proyectos interactivos y dispositivos electrónicos. Utiliza un lenguaje de programación basado en C/C++ y un entorno de desarrollo integrado (IDE) fácil de usar.

Los proyectos de Arduino pueden involucrar sensores, actuadores y otros dispositivos electrónicos para interactuar con el entorno.

Raspberry Pi:

Raspberry Pi es una computadora de placa única (SBC) del tamaño de una tarjeta de crédito.

A diferencia de Arduino, Raspberry Pi es una computadora completa con un procesador, memoria, puertos de entrada/salida y sistema operativo.

Puede ejecutar sistemas operativos completos, como Linux, y admitir una variedad de aplicaciones y tareas de computación.

Es versátil y se utiliza para proyectos que van desde servidores pequeños hasta estaciones de medios, y también en proyectos de IoT (Internet de las cosas).

Arduino es más adecuado para proyectos de electrónica y control específicos, mientras que Raspberry Pi es más poderoso y versátil, funcionando como una computadora completa en un formato compacto. Ambas plataformas son excelentes opciones según las necesidades específicas del proyecto.

Introducción

Carro Robot inteligente autónomo con sensor ultrasónico

En este trabajo se presenta el diseño y la construcción de un carro robot inteligente autónomo que utiliza una placa Arduino UNO, un sensor ultrasónico, un módulo Bluetooth, un control IR y un sistema de seguimiento de línea. El objetivo de este proyecto es crear un vehículo capaz de desplazarse por diferentes superficies y evitando obstáculos, y siguiendo una línea, todo mientras está siendo controlado a distancia mediante un control remoto o un celular.

El Arduino UNO es una placa electrónica de código abierto que permite programar y controlar diferentes componentes, como motores, sensores, luces, etc. El sensor ultrasónico es un dispositivo que emite y recibe ondas de sonido de alta frecuencia, y que permite medir la distancia a un objeto. El módulo Bluetooth es un transceptor inalámbrico que permite la comunicación entre el Arduino y el celular. El control IR es un mando a distancia que envía señales infrarrojas al Arduino para controlar el movimiento del carro.

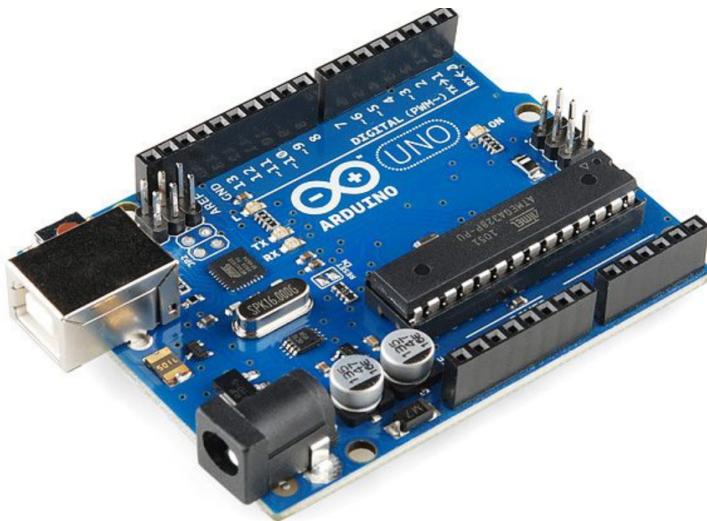
El propósito de este proyecto es aplicar los conocimientos adquiridos durante clase, como la programación, para así desarrollar la creatividad, el ingenio y el trabajo en equipo. El modo de empleo del carro robot es el siguiente: se conecta la batería al Arduino, se enciende el carro, se selecciona el modo de funcionamiento (automático, control remoto o celular) y se observa el comportamiento del carro.

Este trabajo se organiza de la siguiente manera: en el capítulo 1 se describe el marco teórico del proyecto, en el que se explican los conceptos básicos de la placa Arduino, el sensor ultrasónico, el módulo Bluetooth, el control IR y el sistema de seguimiento de línea. En el capítulo 2 se detalla el diseño y la construcción del carro robot, en el que se muestran los materiales, los componentes, el circuito, el código y el montaje. En el capítulo 3 se presentan los resultados y las pruebas realizadas al carro robot, en las que se evalúa el rendimiento, la precisión y la eficiencia del vehículo. En el capítulo 4 se discuten las conclusiones y las recomendaciones del proyecto, en las que se resaltan los logros, las dificultades, las mejoras y las aplicaciones del carro robot.

Carro Robot inteligente autónomo con sensor ultrasónico

El proyecto se utilizará

- 1.-Placa arduino UNO
- 2.- Sensor ultrasónico
- 3.-Módulo Bluetooth
- 4.-Ir control
- 5.-Line Tracking



El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.¹² La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.³ Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios. También es similar al Arduino Nano y Leonardo.⁴⁵ El diseño de referencia de hardware se distribuye bajo una licencia Creative Commons Attribution Share-Alike 2.5 y está disponible en el sitio web de Arduino. Los archivos de diseño y producción para algunas versiones del hardware también están disponibles.

La palabra "uno" significa en italiano lo mismo que en español, y se eligió para marcar el lanzamiento inicial del software Arduino. La placa Uno es la primera de una serie de placas Arduino basadas en USB,² y la versión 1.0 del Arduino IDE fueron las versiones de referencia de Arduino, ahora evolucionadas a nuevas versiones.³ El ATmega328 en la placa viene preprogramado con un cargador de arranque que le permite cargar un nuevo código sin el uso de un programador de hardware externo.²

Mientras que el Uno se comunica utilizando el protocolo STK500 original, difiere de todas las placas anteriores en que no utiliza el chip de controlador USB a serie FTDI. En cambio, usa el Atmega16U2 (Atmega8U2 hasta la versión R2) programado como un adaptador USB

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.¹² La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.³ Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios. También es similar al Arduino Nano y Leonardo.⁴⁵ El diseño de referencia de hardware se distribuye bajo una licencia Creative Commons Attribution Share-Alike 2.5 y está disponible en el sitio web de Arduino. Los archivos de diseño y producción para algunas versiones del hardware también están disponibles.

Arduino UNO R3



La placa Arduino UNO R3, en versiones normal, R1, R2, R3 y SMD es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines digitales de entrada/salida (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un conector ICSP y un botón de reset.

Las características adicionales que vienen con la versión Arduino UNO R3 son:

ATmega16U2 en lugar de 8U2 como convertidor de USB a serie.

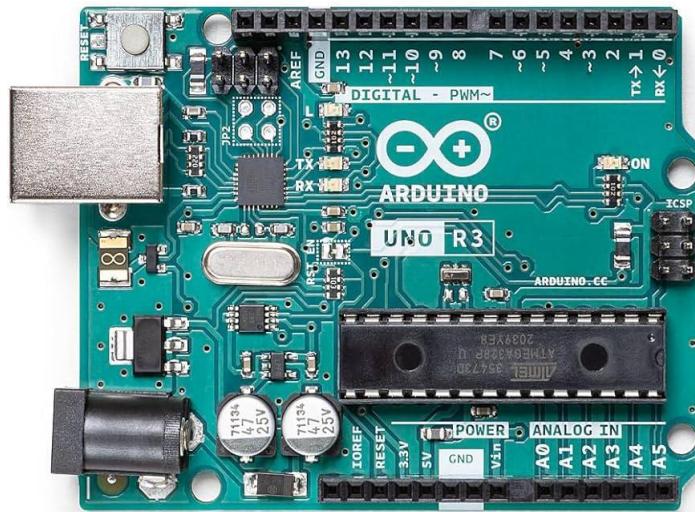
Añadidos pines SDA y SCL para comunicación TWI colocados cerca del pin AREF.

Dos pines nuevos colocados cerca del pin RESET, el IOREF que permite que las placas de expansión (shields) se adapten al voltaje proporcionado por la placa y el segundo es un pin no conectado, que está reservado para propósitos futuros.

Circuito de RESET más fuerte.

La placa Arduino UNO R3 marca el precedente del próximo lanzamiento de Arduino 1.0. El UNO y la versión 1.0 serán las versiones de referencia para las futuras placas Arduino. Es lo último de una serie de nuevas placas Arduino USB, y el modelo de referencia para la plataforma Arduino.

La placa Arduino UNO difiere de todas las placas precedentes en que no usa el chip de controlador USB de la serie FTDI. En cambio, presenta el Atmega16U2 (Atmega8U2 hasta la versión R2) programado como un adaptador de USB a puerto serie.



Características de la Arduino UNO R3

Microcontrolador: ATmega328P

Voltaje de funcionamiento: 5 V. Rango de entrada recomendado entre 7-12 V. Límite máximo entre 6-20 V

Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)

Pines de entrada analógica: 6 pines

Corriente continua por pin: 20 mA

Corriente CC para 3.3V por pin: 50 mA

Memoria Flash: 32 KB (ATmega328P) de la cual 0.5 KB utilizada por el gestor de arranque (bootloader)

SRAM: 2 KB

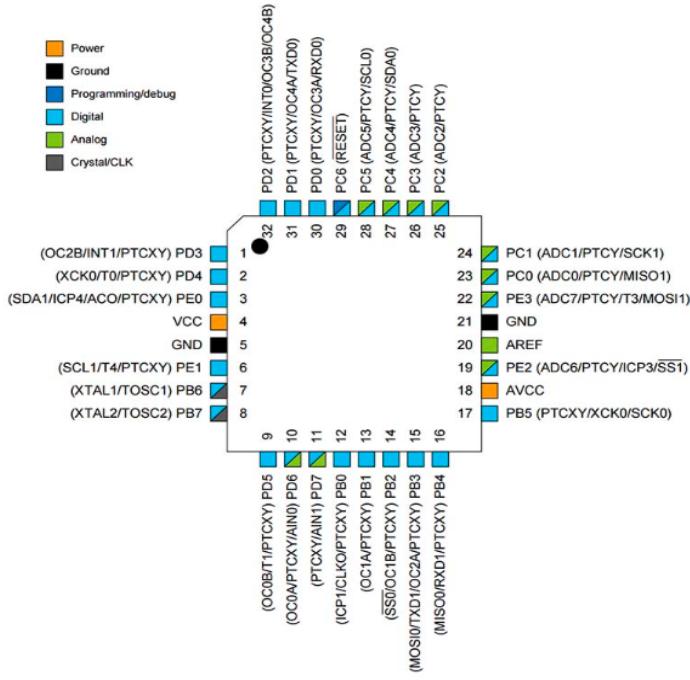
EEPROM: 1 KB

Velocidad del reloj: 16 MHz

Longitud: 68.6 mm

Ancho: 53.4 mm

Peso: 25 g



Microcontrolador Atmel Atmega 328P

El Atmega328p es un microcontrolador de alto rendimiento de 8 bits, basado en tecnología AVR RISC, que combina una memoria flash ISP de 32 KB con capacidades de lectura y escritura, EEPROM de 1KB, y SRAM de 2 KB.

Tiene 23 líneas de E/S de propósito general, 32 registros de trabajo de propósito general, tres temporizadores/contadores flexibles con modos de comparación, interrupciones internas y externas, USART programable, una interfaz I2C, puerto en serie SPI, un convertidor A/D de 6 canales y 10 bits (8 canales en encapsulados TQFP y QFN/MLF), watchdog programable con oscilador interno y cinco modos de ahorro de energía seleccionables por software.

ATMega328P and Arduino Uno Pin Mapping

Arduino function		Arduino function
reset	(PCINT14/RESET) PC6	1
digital pin 0 (RX)	(PCINT16/RxD) PD0	2
digital pin 1 (TX)	(PCINT17/TxD) PD1	3
digital pin 2	(PCINT18/INT0) PD2	4
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5
digital pin 4	(PCINT20/XCK/T0) PD4	6
VCC	VCC	7
GND	GND	8
crystal	(PCINT6/XTAL1/TOSC1) PB6	9
crystal	(PCINT7/XTAL2/TOSC2) PB7	10
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12
digital pin 7	(PCINT23/AIN1) PD7	13
digital pin 8	(PCINT0/CLK0/ICP1) PB0	14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega16 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Atmega 328p con encapsulado DIP28

Alimentación de la Arduino UNO R3

La placa Arduino UNO R3 puede alimentarse a través de la conexión USB o con una fuente de alimentación externa. La fuente de poder se selecciona automáticamente.

La alimentación externa (no USB) puede provenir de un adaptador de CA a CC o batería.

El adaptador se puede conectar al enchufar un conector 2.1 mm de alimentación de la placa, con centro positivo. Los cables de una batería se pueden insertar en los pines GND y Vin del conector POWER.

La placa puede operar con un suministro externo de 6 a 20 V. Sin embargo, si se suministra con menos de 7 V, el pin de 5 V puede suministrar menos de cinco voltios y la placa puede volverse inestable. Si usa más de 12 V, el regulador de voltaje puede sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 V.

Los pines de alimentación son los siguientes:

Vin: El voltaje de entrada a la placa Arduino UNO R3 cuando está usando una fuente de alimentación externa (a diferencia de los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Puede suministrar voltaje a través de este pin o, si suministra voltaje a través del conector de alimentación, acceda a través de este pin.

5V: Este pin produce un voltaje de 5V regulados desde el regulador en la placa. La placa Arduino UNO R3 se puede alimentar con la toma de alimentación de CC (7-12 V), el conector USB (5V) o el pin VIN de la placa (7-12 V). El suministro de voltaje a través de los pines de 5V o 3.3V evita el regulador y puede dañar la placa. Úselos con precaución.

3V3: Una fuente de 3.3 voltios generada por el regulador en la placa. El consumo máximo de corriente es de 50mA.

GND: Pines de tierra

IOREF: Este pin en la placa Arduino UNO R3 proporciona la referencia de voltaje con la que funciona el microcontrolador. Una placa de expansión (shield) correctamente configurada puede leer el voltaje del pin IOREF y seleccionar la fuente de alimentación apropiada o habilitar los adaptadores de nivel de voltaje en las salidas para trabajar con 5V o 3.3V.

Memoria de la Arduino UNO R3

La placa Arduino UNO R3 usa el microcontrolador ATmega328P tiene 32 KB de memoria FLASH (con 0.5 KB ocupados por el gestor de arranque). También tiene 2 KB de SRAM y 1 KB de EEPROM (que se pueden leer y escribir con la librería EEPROM).

Entradas y salidas de la Arduino UNO R3

Cada uno de los 14 pines digitales de la placa Arduino UNO R3 se pueden usar como entrada o salida, usando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Funcionan a 5V. Cada pin puede proporcionar o recibir 20 mA como condición de operación recomendada y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50 kohm. Un máximo de 40 mA es el valor que no debe excederse en ningún pin de E/S para evitar daños permanentes al microcontrolador.

Serial: pines 0 (RX) y 1 (TX). Se usa para recibir (RX) y transmitir (TX) datos en el puerto serie en TTL. Estos pines están conectados a los pines correspondientes del chip serie ATmega8U2 USB-to-TTL.

Interrupciones externas: pines 2 y 3. Estas clavijas se pueden configurar para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Vea la función `attachInterrupt()` para más detalles.

PWM: pines 3, 5, 6, 9, 10 y 11. Proporcionan salida PWM de 8 bits con la función `analogWrite()`.

SPI: pines 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines admiten comunicación SPI utilizando la librería SPI.

LED: pin 13. Hay un LED integrado accionado por el pin digital 13. Cuando el pin tiene un valor ALTO, el LED está encendido, cuando el pin está BAJO, está apagado.

TWI: pin A4 o SDA y pin A5 o SCL. Soporte de comunicación con bus I2C (TWI) utilizando la librería Wire.

Diagrama de pines de la Arduino UNO R3

La placa Arduino UNO R3 tiene 6 entradas analógicas, etiquetadas de A0 a A5, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, miden desde tierra (GND) a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y la función `analogReference()`.

Hay un par de otros pines en la placa:

AREF. Voltaje de referencia para las entradas analógicas. Usado con `analogReference()`.

Reset: Es un pin de entrada. Llevar este pin a tierra (GND) reiniciará el microcontrolador. Normalmente se usa para agregar un botón de reinicio a las placas de expansión (shields).

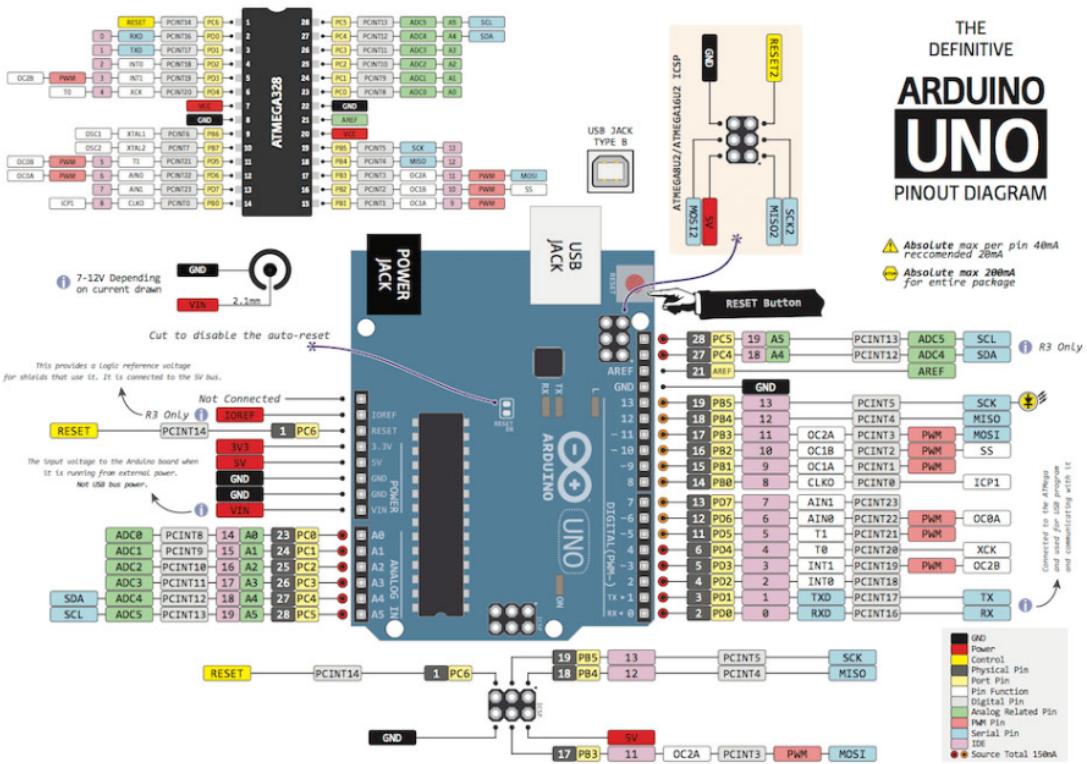


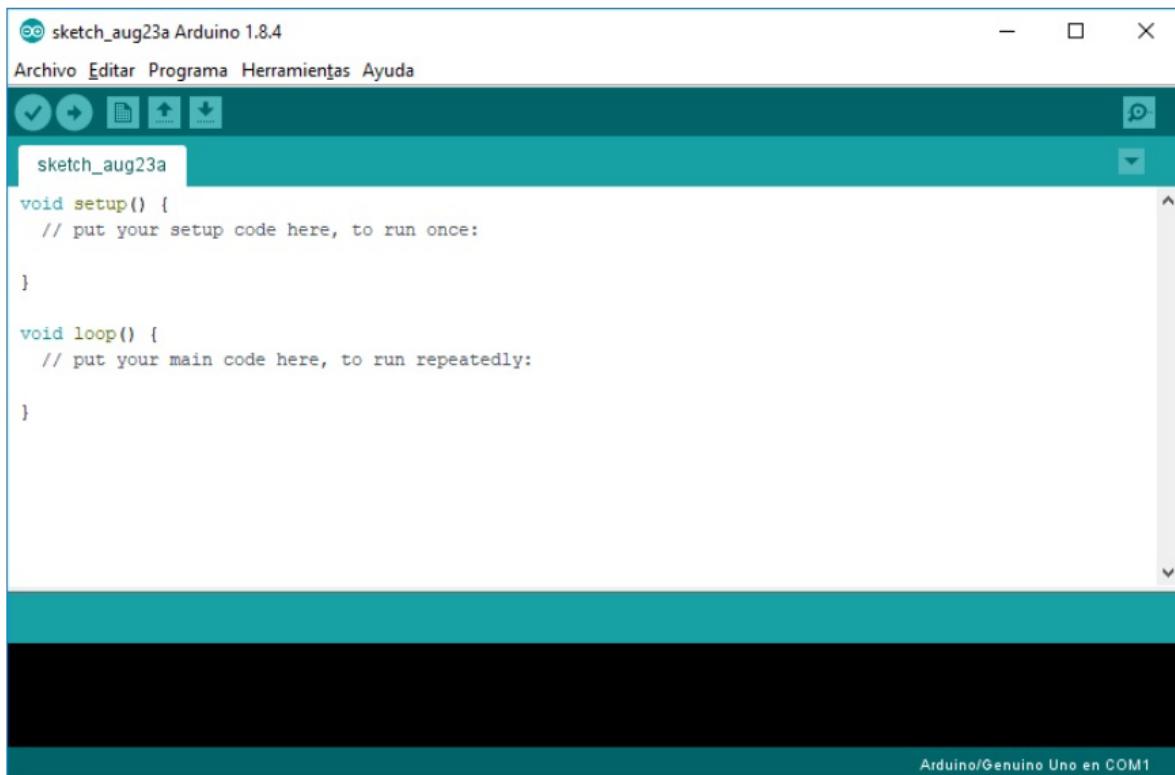
Diagrama de pines de la Arduino UNO R3

Comunicaciones en la Arduino UNO R3

La placa Arduino UNO R3 tiene varias facilidades para comunicarse con un ordenador, otra placa Arduino u otros microcontroladores. El ATmega328P proporciona comunicación en serie UART TTL (5V), que está disponible en los pines digitales 0 (RX) y 1 (TX).

Un ATmega16U2 en la placa canaliza esta comunicación en serie a través del conector USB y aparece como un puerto virtual para el software en la computadora. El firmware 16U2 usa los controladores COM USB estándar y no se necesita ningún controlador externo. Sin embargo, en Windows, se requiere un archivo .inf.

Programación de la Arduino UNO R3



```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

Arduino/Genuino Uno en COM1

Arduino IDE

La placa Arduino UNO se puede programar con el entorno de desarrollo Arduino IDE.

Arduino IDE

El ATmega328P en la placa Arduino UNO R3 viene pre-programado con un gestor de arranque (bootloader) que le permite cargar un nuevo programa sin el uso de un programador de hardware externo. Se comunica utilizando el protocolo original STK500.

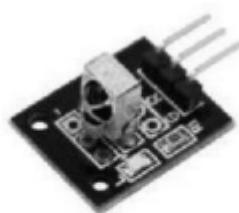
Programador AVR STK500

También puede omitir el gestor de arranque y programar el microcontrolador a través del conector ICSP (In-Circuit Serial Programming – Programación serial en circuito) usando Arduino ISP o similar. El código fuente del firmware ATmega16U2 (o 8U2 en las placas rev1 y rev2) está disponible en el repositorio de Arduino. El ATmega16U2/8U2 está cargado con un gestor de arranque DFU, que puede ser activado por:

En placas Rev1: conectando el puente de soldadura en la parte posterior de la placa (cerca del mapa de Italia) y luego buscando el 8U2.

En las placas Rev2 o posteriores: hay una resistencia que tira de la línea HUB 8U2/16U2 a tierra, lo que facilita poner el modo DFU. A continuación, puede usar el software FLIP de Atmel (Windows) o el programador DFU (Mac OS X y Linux) para cargar un nuevo firmware. O puede usar el conector ISP con un programador externo (sobrescribiendo el gestor de arranque DFU).

COMPONENTES DE LA PLACA ARDUINO UNO R3



IR Receiver Module 1PCS

Módulo IR receiver

Es un componente electrónico que es capaz de recibir señales infrarrojas (IR). Estos módulos son comúnmente utilizados en dispositivos electrónicos para la comunicación a través de señales infrarrojas. Los controles remotos de televisores, reproductores de DVD, aires acondicionados y otros dispositivos utilizan tecnología infrarroja para enviar comandos a sus receptores correspondientes.

El módulo IR receiver está diseñado para detectar y decodificar señales infrarrojas enviadas por un control remoto u otro dispositivo emisor de infrarrojos. Cuando un control remoto emite un comando, este se transmite a través de señales infrarrojas que son captadas por el módulo IR receiver. Luego, el módulo convierte estas señales infrarrojas en señales eléctricas que pueden ser interpretadas por el dispositivo al que está conectado. Esto permite que el dispositivo responda a los comandos enviados a través del control remoto.



Line Tracking Module 3PCS

Line tracking module

Es un módulo de seguimiento de línea. Estos módulos son comúnmente utilizados en proyectos de robótica para permitir que un robot siga una línea trazada en el suelo. El módulo suele estar equipado con sensores infrarrojos que son capaces de

detectar la diferencia entre la superficie de la línea y el área circundante. Esto permite al robot ajustar su dirección y seguir la trayectoria de la línea.

A menudo, un "line tracking module" incluye varios sensores distribuidos a lo largo del chasis del robot. Estos sensores pueden emitir luz infrarroja hacia el suelo y luego medir la cantidad de luz reflejada. La variación en la cantidad de luz reflejada permite al módulo determinar si el sensor está sobre la línea o fuera de ella. En función de esta información, el robot puede realizar ajustes en sus motores o ruedas para mantenerse en la línea.



El sensor ultrasónico HC-SR04.

Este sensor es fácil de usar y es asequible. Aquí hay una breve descripción de cómo funciona y cómo puedes usarlo con Arduino:

Principio de funcionamiento:

El sensor emite un pulso ultrasónico.

Mide el tiempo que tarda el pulso en rebotar en un objeto y regresar al sensor.

Conexiones:

El sensor tiene cuatro pines: VCC (alimentación), Trig (disparo), Echo (eco), y GND (tierra).

Conéctalos al Arduino de la siguiente manera: VCC al pin de 5V, Trig a un pin digital, Echo a otro pin digital, y GND a tierra.

Este código enviará un pulso ultrasónico, medirá el tiempo que tarda en regresar y calculará la distancia en centímetros, que se imprimirá en el puerto serial.

Código de ejemplo:

Código de ejemplo en Arduino para medir la distancia con el sensor ultrasónico:

```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);
}
```

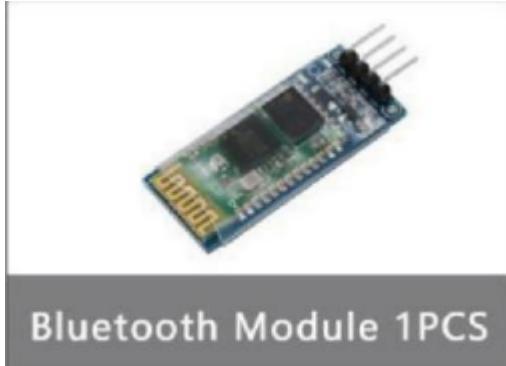
```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);
}
```



Módulo Bluetooth para Arduino

Es un dispositivo que permite la comunicación inalámbrica entre un Arduino y otros dispositivos, como smartphones, tablets o computadoras, que también admitan Bluetooth. Estos módulos Bluetooth permiten la transferencia de datos entre el Arduino y otros dispositivos, lo que amplía las posibilidades de interacción y control remoto en proyectos electrónicos.

Uno de los módulos Bluetooth más comunes para Arduino es el HC-05 o HC-06. A continuación, te doy una breve guía sobre cómo conectar y utilizar el módulo Bluetooth HC-05 con Arduino:

Conexiones físicas:

El módulo HC-05 generalmente tiene cuatro pines principales: VCC (alimentación), GND (tierra), TX (transmisión), y RX (recepción).

Conéctalos al Arduino de la siguiente manera: VCC al pin de 5V, GND a tierra, TX a un pin digital (por ejemplo, D2), y RX a otro pin digital (por ejemplo, D3).

Recuerda configurar el módulo Bluetooth con la velocidad de baudios correcta (en este ejemplo, 9600 bps) y también asegúrate de que el módulo esté configurado correctamente (por ejemplo, en modo esclavo) según las especificaciones del fabricante. Ten en cuenta que hay otros módulos Bluetooth disponibles, y las conexiones y configuraciones específicas pueden variar según el modelo. Siempre consulta la documentación del módulo específico que estás utilizando.

Código de ejemplo:

Aquí hay un ejemplo sencillo de código para Arduino que utiliza el módulo HC-05 para recibir datos a través de Bluetooth y mostrarlos en el puerto serial:

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(2, 3); // RX, TX

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
}

void loop() {
  if (bluetooth.available() > 0) {
    char data = bluetooth.read();
    Serial.print(data);
  }
}
```

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(2, 3); // RX, TX

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
}

void loop() {
  if (bluetooth.available() > 0) {
    char data = bluetooth.read();
    Serial.print(data);
  }
}
```



Servo Motor(SG90) 1PCS

Servo motor SG90

Es un tipo específico de motor eléctrico utilizado en aplicaciones donde se requiere un control preciso del ángulo. Los servos SG90 son pequeños y ligeros, y son ampliamente utilizados en proyectos de electrónica, robótica y modelismo.

Control de Posición: Los servomotores SG90 están diseñados para controlar la posición angular con precisión. Pueden girar un eje a una posición específica, y puedes controlar la posición deseada enviando señales de control.

Rango de Giro Limitado: A diferencia de los motores de corriente continua (DC) que pueden girar continuamente, los servo motores están diseñados para tener un rango de giro limitado, generalmente alrededor de 180 grados. Esto los hace ideales para aplicaciones donde se necesita un movimiento angular preciso.

Retroalimentación: Los servo motores a menudo incorporan algún tipo de sistema de retroalimentación, como un potenciómetro, que proporciona información sobre la posición actual del eje. Esta retroalimentación permite al controlador ajustar la posición del servo para que coincida con la posición deseada.

Señal de Control: Los servomotores SG90 son controlados mediante señales de pulso de duración variable. La anchura del pulso determina la posición angular del servo.

Alimentación: Suelen operar con voltajes bajos, como 4.8V o 6V, lo que los hace compatibles con muchas fuentes de alimentación utilizadas en proyectos electrónicos.

Debido a su tamaño compacto y su capacidad para proporcionar un control preciso del movimiento angular, los servomotores SG90 son muy populares en proyectos de hobby, como robots pequeños, modelismo, mecanismos de dirección en vehículos teledirigidos y otras aplicaciones donde se requiere un control de posición preciso y compacto.



DC Motor 4PCS

Un "DC motor 4pcs para Arduino" es un conjunto de cuatro motores de corriente continua (DC motors) diseñados para ser utilizados con placas Arduino o microcontroladores similares. Los motores de corriente continua son dispositivos electromecánicos que convierten la energía eléctrica en movimiento rotativo.

Características clave de los motores de corriente continua y cómo se podrían utilizar en un contexto de Arduino:

Funcionamiento básico: Los motores de corriente continua funcionan mediante la aplicación de voltaje a sus terminales. Cuando se aplica voltaje, la corriente fluye a través del motor, generando un campo magnético que interactúa con el campo magnético del motor, lo que produce el movimiento rotativo del eje del motor.

Control de velocidad y dirección: Para controlar la velocidad y dirección de un motor de corriente continua, se ajusta la cantidad de voltaje aplicado y la polaridad de la conexión. En un contexto de Arduino, esto se lograría utilizando puentes H o módulos de control de motores que permiten al Arduino controlar la velocidad y dirección de cada motor de manera independiente.

Conexiones a Arduino: Cada motor DC generalmente se conecta a través de un módulo de control de motor o puente H. Estos módulos permiten al Arduino enviar señales de control para ajustar la velocidad y dirección de cada motor. Además, los motores generalmente requieren una fuente de alimentación externa para funcionar.

Proyectos típicos: Los motores de corriente continua son fundamentales en muchos proyectos de robótica y electrónica. Con cuatro motores, puedes implementar sistemas de tracción diferencial en robots móviles, crear vehículos teledirigidos, construir mecanismos de control de movimiento y mucho más. Este código hace que el motor gire en una dirección, luego en la dirección opuesta, y finalmente se detiene por un segundo antes de repetir el proceso.

Aquí hay un ejemplo simple de código de Arduino para controlar un motor DC utilizando un puente H (por ejemplo, L298N):

```
const int motorPin1 = 2; // Conectar a IN1 del puente H
const int motorPin2 = 3; // Conectar a IN2 del puente H

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
}

void loop() {
    // Girar en una dirección durante 2 segundos
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    delay(2000);

    // Girar en la dirección opuesta durante 2 segundos
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    delay(2000);

    // Detener el motor durante 1 segundo
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    delay(1000);
}
```

```
const int motorPin1 = 2; // Conectar a IN1 del puente H

const int motorPin2 = 3; // Conectar a IN2 del puente H

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
}

void loop() {
    // Girar en una dirección durante 2 segundos
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    delay(2000);

    // Girar en la dirección opuesta durante 2 segundos
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    delay(2000);

    // Detener el motor durante 1 segundo
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    delay(1000);
```



L298N Motor Board

}

El L298N es un puente H de doble sentido que permite controlar la dirección y la velocidad de dos motores de corriente continua (DC motors) de manera independiente. Este dispositivo es comúnmente utilizado con placas Arduino y otros microcontroladores para el control de motores en proyectos de robótica y automatización.

Aquí hay una descripción básica del módulo L298N y cómo se utiliza con Arduino:

Puente H: El L298N es un tipo de circuito conocido como un puente H. Un puente H permite controlar la dirección del flujo de corriente a través de un motor, lo que a su vez controla la dirección del giro del motor.

Control de Dos Motores: El L298N es capaz de controlar dos motores de corriente continua de manera simultánea. Cada motor tiene dos conexiones, una para el terminal positivo y otra para el terminal negativo.

Conexiones del Motor: Cada motor se conecta a las salidas del L298N, llamadas OUT1, OUT2, OUT3 y OUT4. La polaridad de estas conexiones determina la dirección en la que girará el motor.

Control de Velocidad: El L298N permite controlar la velocidad de los motores mediante la variación de la señal de PWM (modulación por ancho de pulso) aplicada a las conexiones ENA y ENB. Al ajustar el ciclo de trabajo del PWM, puedes controlar la velocidad de giro del motor.

Conexión a Arduino: El L298N se conecta a las salidas digitales de Arduino para controlar la dirección y velocidad de los motores. Los pines

de control suelen ser IN1, IN2, IN3 e IN4 para la dirección, y ENA y ENB para la velocidad mediante PWM.

Este código controla un motor conectado al L298N, girando en una dirección a velocidad media, luego en la dirección opuesta a velocidad baja, y finalmente se detiene antes de repetir el proceso. Recuerda ajustar las conexiones y pines según tu configuración específica.

```
const int motor1A = 2; // Conectar a IN1 del L298N
const int motor1B = 3; // Conectar a IN2 del L298N
const int enable1 = 9; // Conectar a ENA del L298N

void setup() {
    pinMode(motor1A, OUTPUT);
    pinMode(motor1B, OUTPUT);
    pinMode(enable1, OUTPUT);
}

void loop() {
    // Girar en una dirección a velocidad media
    digitalWrite(motor1A, HIGH);
    digitalWrite(motor1B, LOW);
    analogWrite(enable1, 128);

    delay(2000);

    // Girar en la dirección opuesta a velocidad baja
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, HIGH);
    analogWrite(enable1, 64);

    delay(2000);

    // Detener el motor
    digitalWrite(motor1A, LOW);
    digitalWrite(motor1B, LOW);
    analogWrite(enable1, 0);

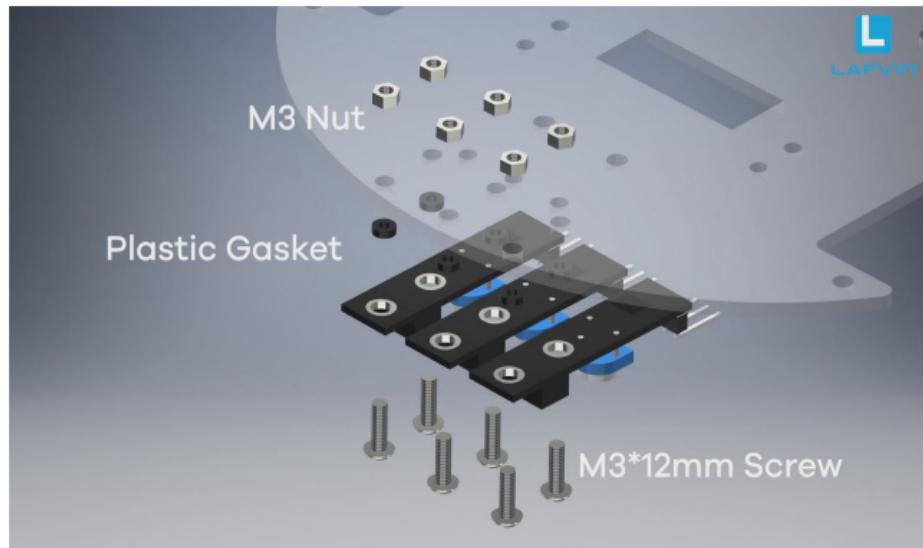
    delay(1000);
}
```

arduino

Copy code

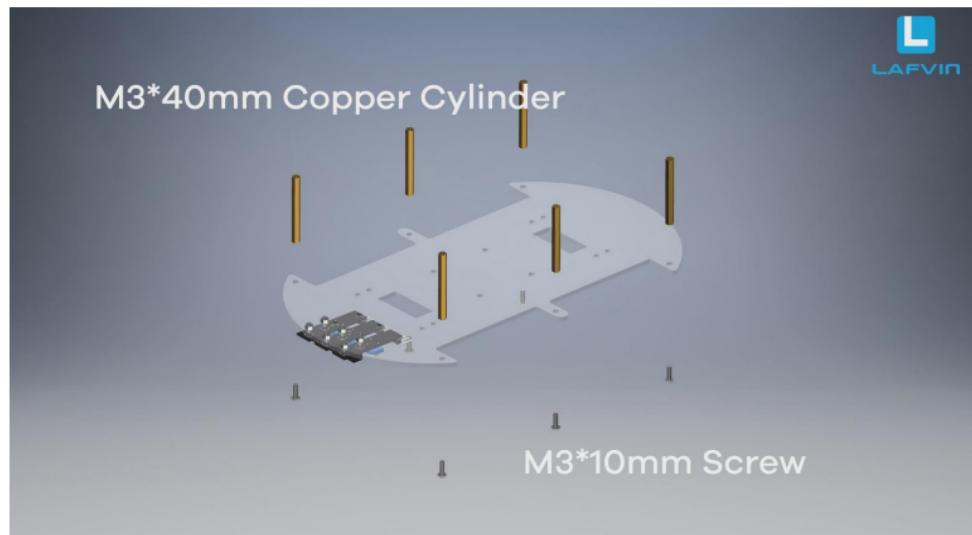
```
const int motor1A = 2; // Conectar a IN1 del L298N
```

```
const int motor1B = 3; // Conectar a IN2 del L298N  
const int enable1 = 9; // Conectar a ENA del L298N  
  
void setup() {  
    pinMode(motor1A, OUTPUT);  
    pinMode(motor1B, OUTPUT);  
    pinMode(enable1, OUTPUT);  
}  
  
void loop() {  
    // Girar en una dirección a velocidad media  
    digitalWrite(motor1A, HIGH);  
    digitalWrite(motor1B, LOW);  
    analogWrite(enable1, 128);  
  
    delay(2000);  
  
    // Girar en la dirección opuesta a velocidad baja  
    digitalWrite(motor1A, LOW);  
    digitalWrite(motor1B, HIGH);  
    analogWrite(enable1, 64);  
  
    delay(2000);  
  
    // Detener el motor  
    digitalWrite(motor1A, LOW);  
    digitalWrite(motor1B, LOW);  
    analogWrite(enable1, 0);  
    delay(1000);  
}  
}
```

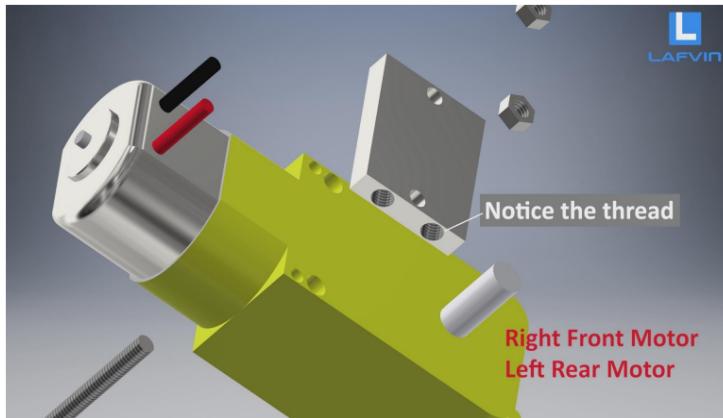


En esta etapa pasaremos en la parte del ensamblaje del vehículo:

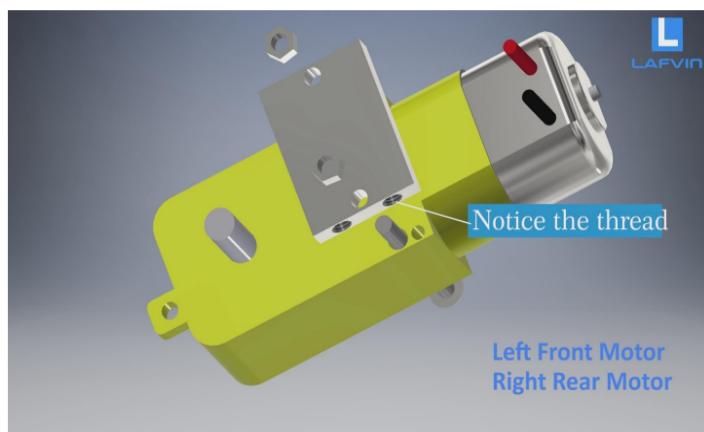
Primer paso ,en la base conectar el plástico Gasket a la base en cual se parte esencial de el ensamblaje del carro robot.



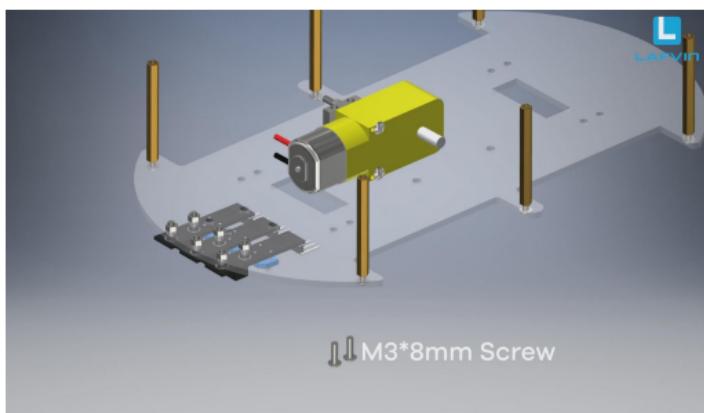
Montar los cilindros Copper a la base, que sera poste de colocación de la otra base superior.



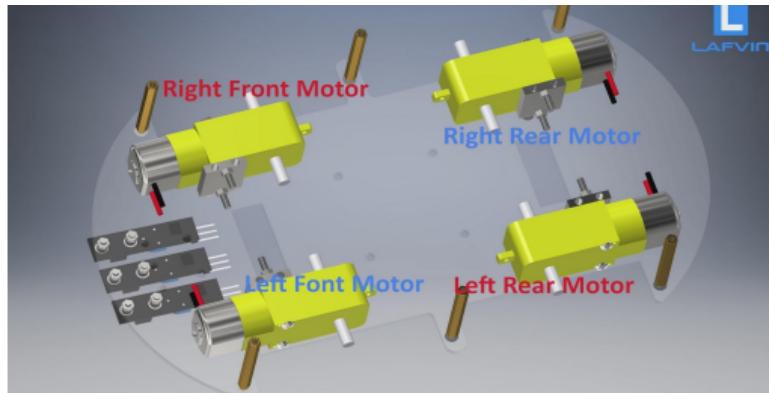
Colocar los componente de los servo motores de cada rueda, en cual se podra anclar en la base del robot



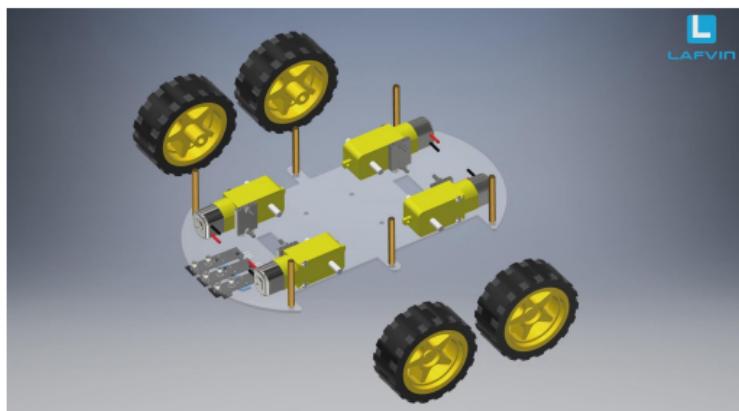
Colocar el mismo procedimiento en las otras partes correspondientes de los motores, de las ruedas para poder colocar a la base principal.



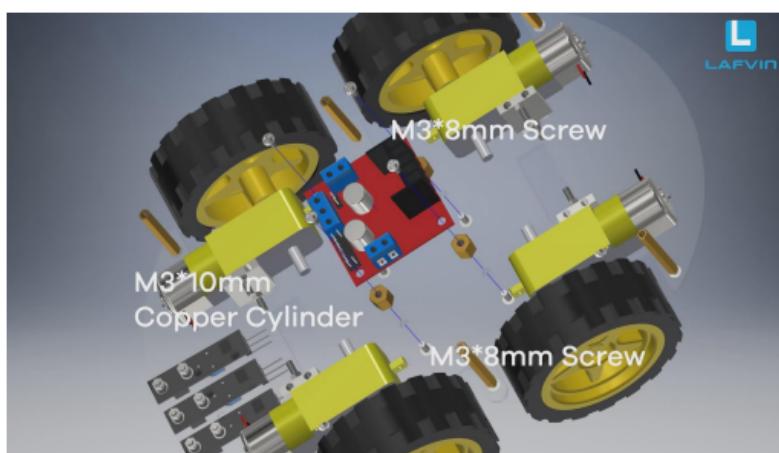
Se coloca el motor en la parte correspondiente (derecha-izquierda) a la base, con sus tornillos que vienen en la caja..



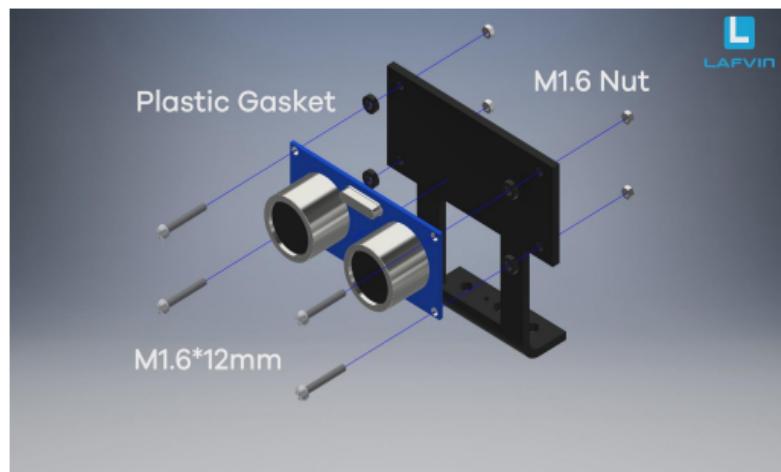
Verificar que los 4 motores, estén instalado adecuadamente en cada parte , fijarlo bien con los tornillos, y estén conectado correctamente



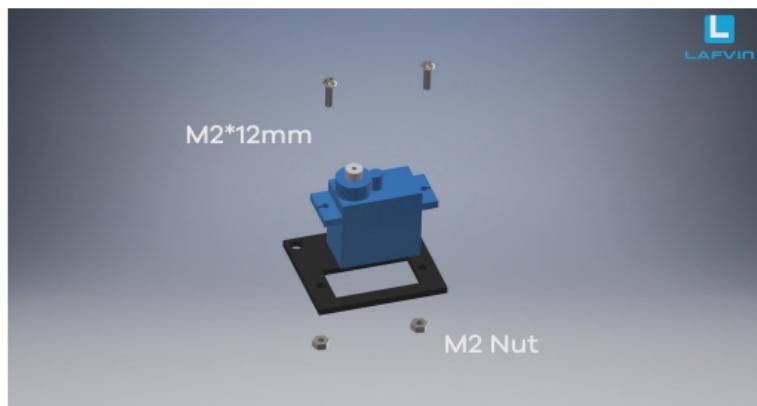
Es hora de colocar las llantas a los motores correspondiente y verificar su movilidad de cada una manualmente.



Colocar el componente regulador de los motores a la placa en la parte superior.



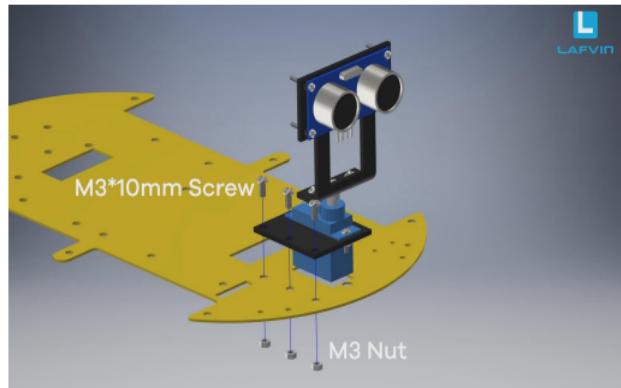
Colocar el sensor hipersónico en su base, para luego colocarlo en la base del carro robot.



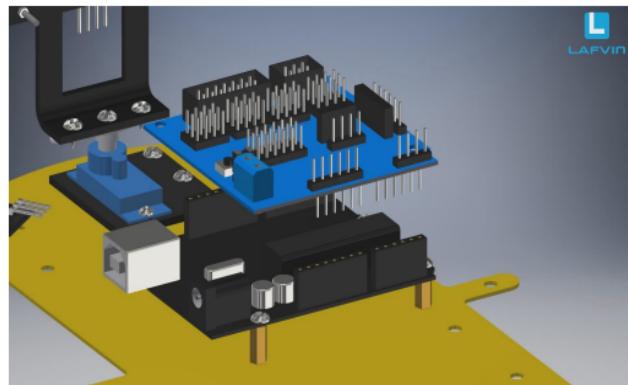
Colocar el interruptor en la base correspondiente



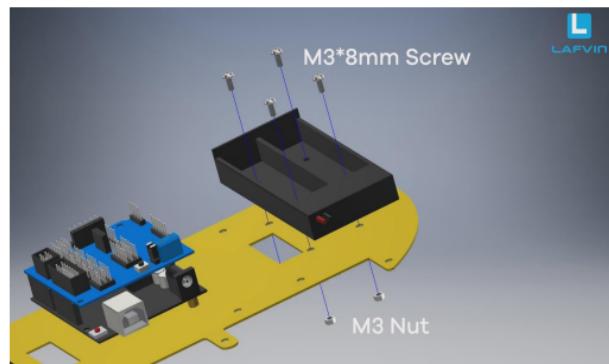
Ahora, se colocará el servomotor que es la parte esencial del movimiento de los motores en la placa arduino, es el componente principal para el movimientos de los motores.



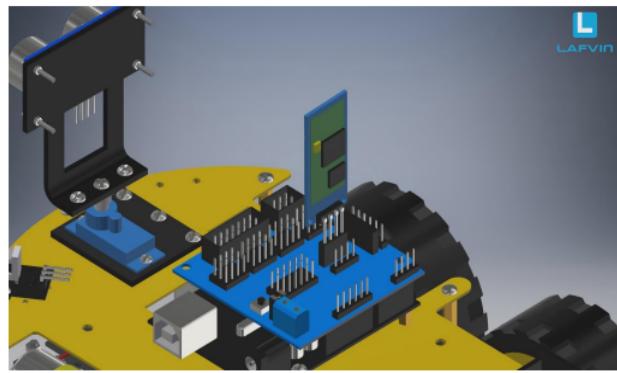
Se coloca el sensor supersónico ya armado a la base, en la parte posterior, se ancla a la base.



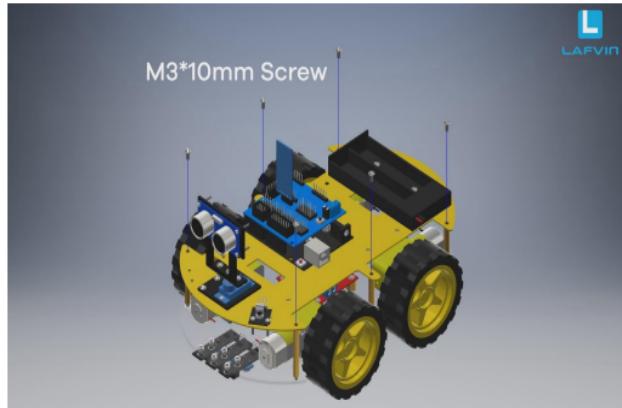
Se coloca la placa arduino en la base fijándolo en sus espacio para colocar el tornillo correspondiente.



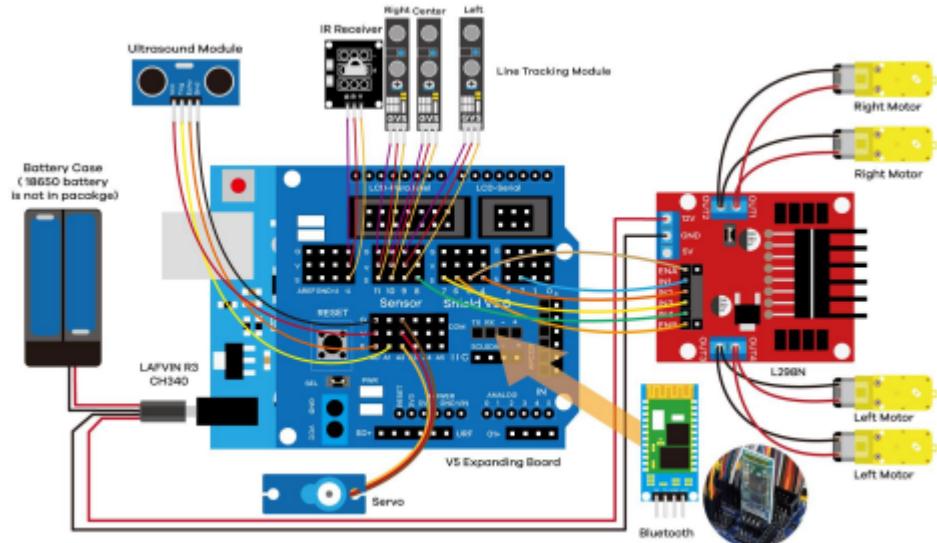
Colocar la base para batería en la base correspondiente del carro robot.



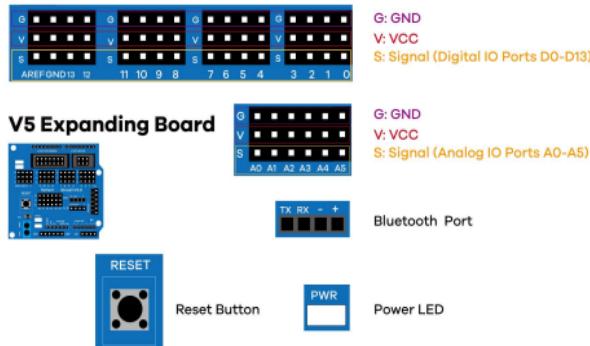
Se coloca el receptor bluetooth a la placa arduino .



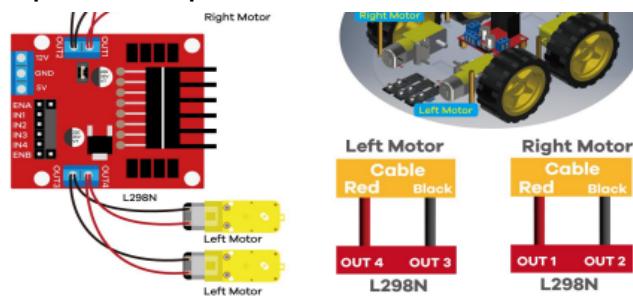
Verificar que todos los componentes estén colocados adecuadamente y atornillados para su fijación en la base.



Es una diagrama visual de la colocación de los pines en los conectores correspondiente, verificar bien su colocación de cada cable para evitarse algún cortocircuito por las polaridades que puede ser.

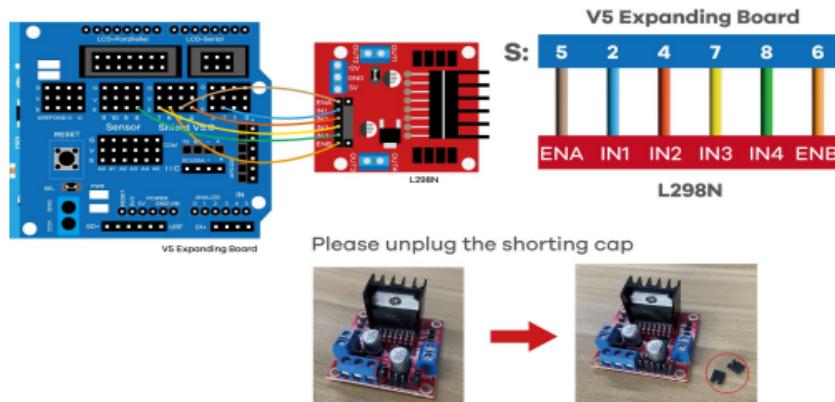


En la tarjeta v5 Expanding , verificar bien la colocación de los pines en el lugar correcto para su óptimo funcionamiento.

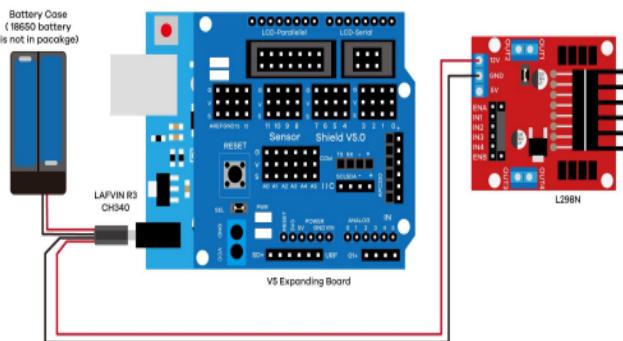


24/83

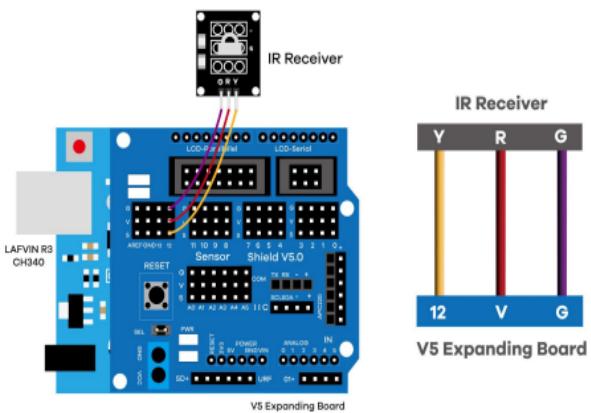
Verificar el conector del servomotor que esté conectado a su tarjeta , con la conexión de cada motor de las llantas esten conectado en su lugar correspondiente y poder seguir en el ensamblaje del vehículo



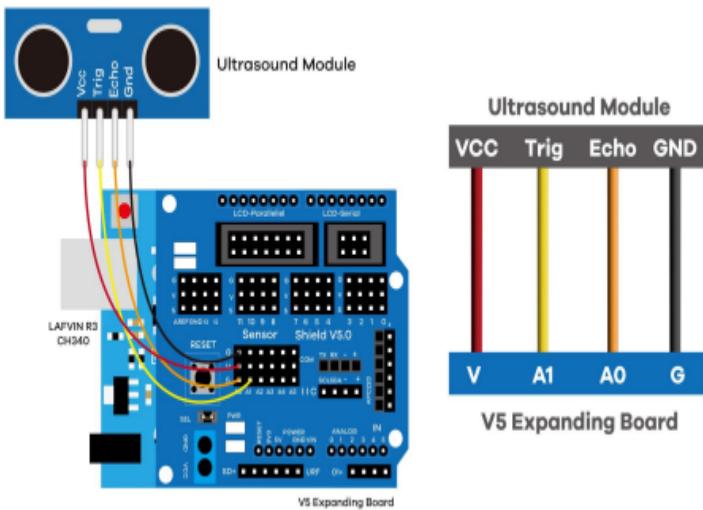
Verificar la conexión correcta de pines del expanding board.



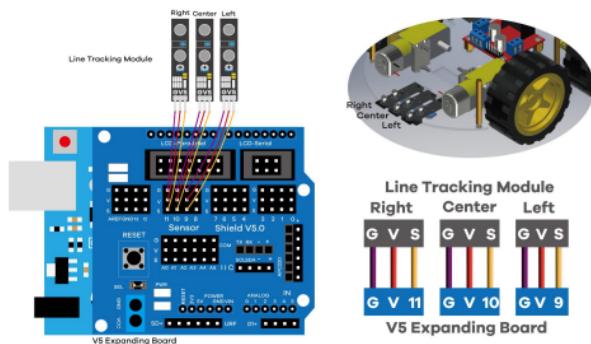
Colocar la conexión correcta de la fuente de energía a las placas para su funcionamiento correcto.



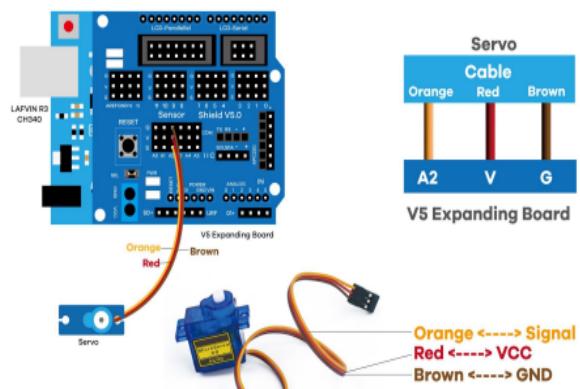
Colocar el dispositivo IR receiver en la placa arduino en su parte correspondiente conectando los pines de manera correcta.



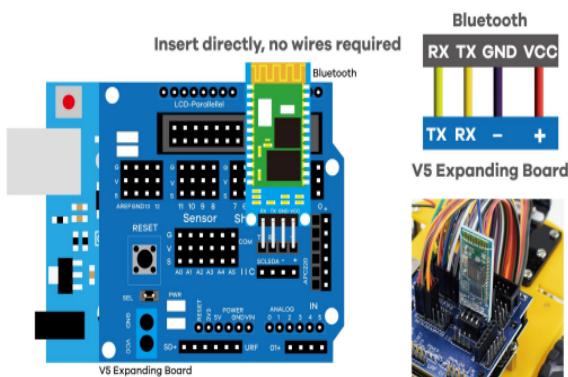
Conectar el módulo ultrasónico en la placa arduino con la colocación de los pines adecuadamente.



Es tiempo de colocar el Módulo line tracking en la placa arduino ,verificando que estén colocados correctamente los pines, para su óptimo funcionamiento.



En este paso se colocará el módulo servo a la placa V5 expanding, para poder ser alimentado con energía y recibir señales de orden de la placa arduino para el uso de las ruedas.



Esta sera la ultima parte de colocación de los componentes,en la placa arduino,el receptor bluetooth ,(colocarlo bien en sus pines)

INSTRUCCIONES:

[Software | Arduino](#)

① Conecte su placa Arduino UNO. Navegue hasta el Administrador de dispositivos de Windows

(Inicio > escriba "administrador de dispositivos" para buscar > haga doble clic en el primer resultado para iniciar el Administrador de dispositivos). Vaya al puerto, si aparece "Arduino UNO (COMX)" en la lista de puertos, significa que el controlador Arduino Uno

ha sido instalado exitosamente. puedes saltarte este paso

Downloads

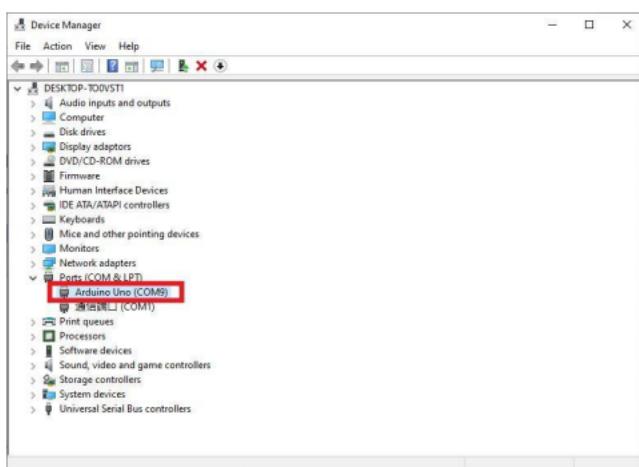


[Instale el controlador Arduino UNO](#)

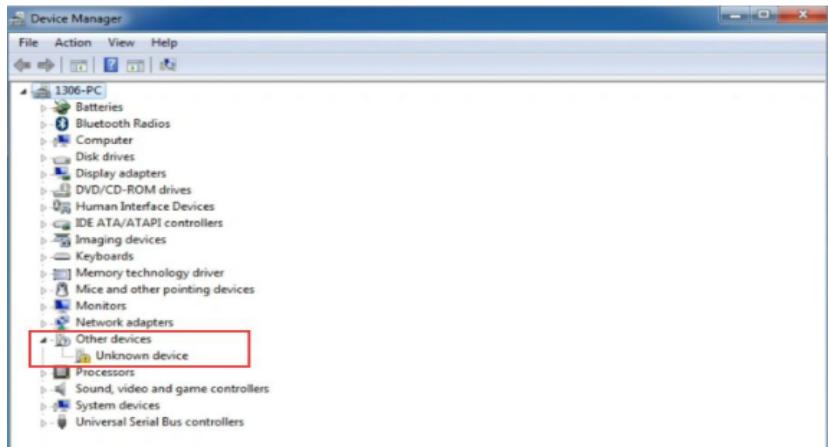
Conecte su placa Arduino UNO. Navegue hasta el Administrador de dispositivos de Windows

(Inicio > escriba "administrador de dispositivos" para buscar > haga doble clic en el primer resultado para iniciar el Administrador de dispositivos). Vaya al puerto, si aparece "Arduino UNO (COMX)" en la lista de puertos, significa que el controlador Arduino Uno

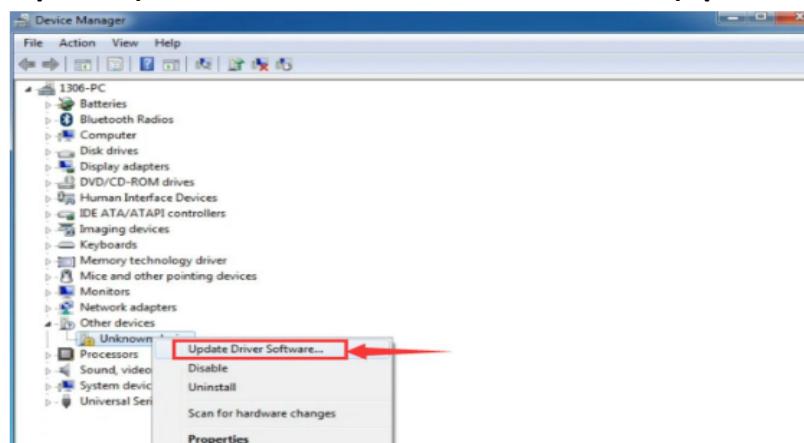
ha sido instalado exitosamente. puedes omitir este paso:



② Si no puede encontrar el puerto Arduino UNO (COM), vaya a "Otros dispositivos" y busque un "Dispositivo desconocido".

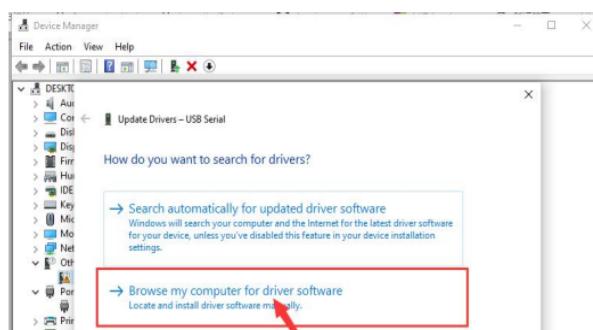


③ Luego haga clic derecho en el dispositivo y seleccione la opción del menú superior (Actualizar software del controlador...) que se muestra como la figura:



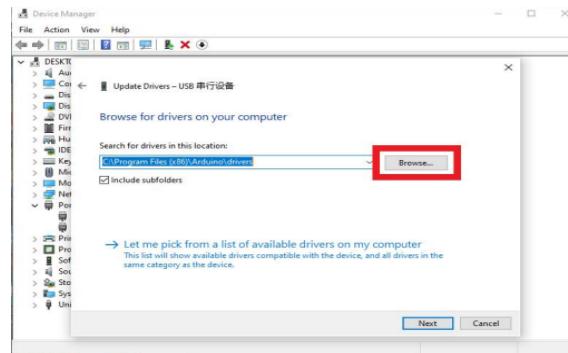
④ Luego se le solicitará “Buscar automáticamente el software del controlador actualizado” o “Buscar en mi computadora para el software del controlador”. Se muestra a continuación. En esta página, seleccione “Buscar controladores en mi computadora”.

④ Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.

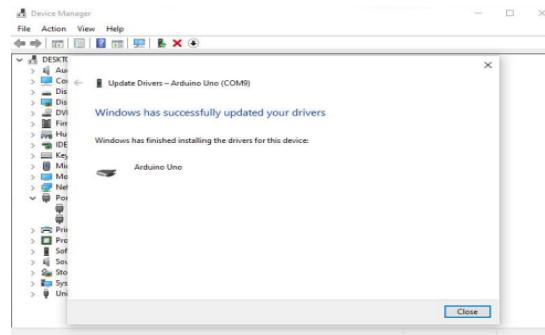


⑤Haga clic derecho en el dispositivo y seleccione la opción del menú superior (Actualizar software del controlador...). Seleccione la opción para explorar y navegar hasta: C\Program Files(x86)\Arduino\drivers. (Nota: Aquí está el ruta que elija para instalar Arduino IDE. Desde que instalé el IDE de Arduino en el disco C, la ruta de ubicación

Elegí C\Program Files(x86)\Arduino\drivers)

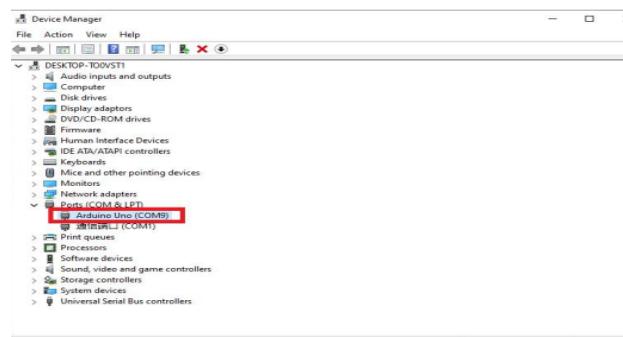


⑥Haga clic en "Siguiente" y es posible que reciba una advertencia de seguridad; de ser así, permita que se instale el software. Una vez que se haya instalado el software, recibirá un mensaje de confirmación. Instalación completada, haga clic cerca.



Hasta ahora, el controlador está bien instalado. Luego puede hacer clic derecho en “Equipo”→“Propiedades”→“Dispositivo

manager”, debería ver el dispositivo como el:



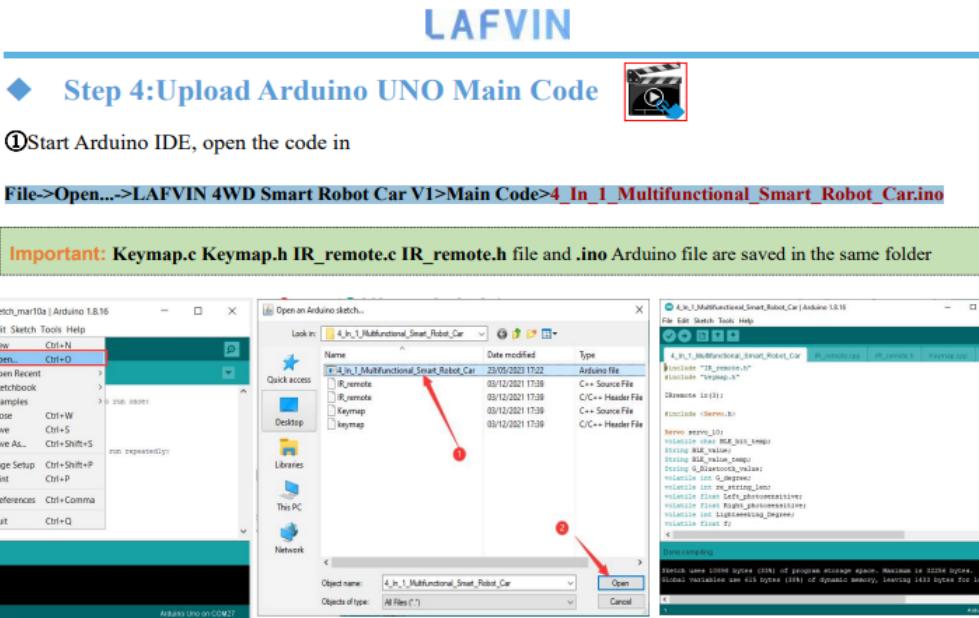
④ Paso 4: Cargue el código principal de Arduino UNO

① Inicie Arduino IDE, abra el código en

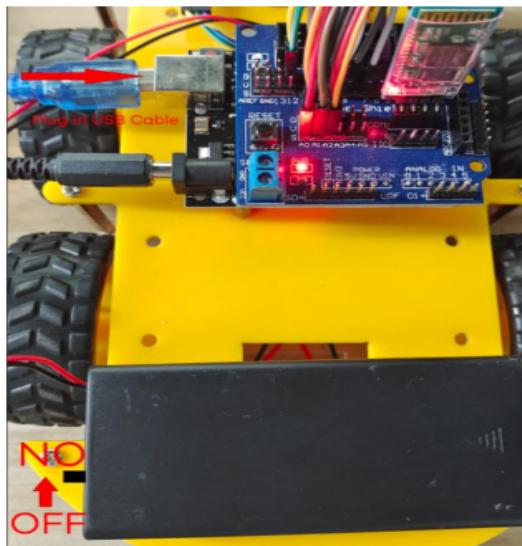
Archivo->Abrir...->LAFVIN 4WD Smart Robot Car V1>Código principal>4_In_1_Multifunction_Smart_Robot_Car.ino

File->Open...->LAFVIN 4WD Smart Robot Car V1>Main Code>4_In_1_Multifunctional_Smart_Robot_Car.ino

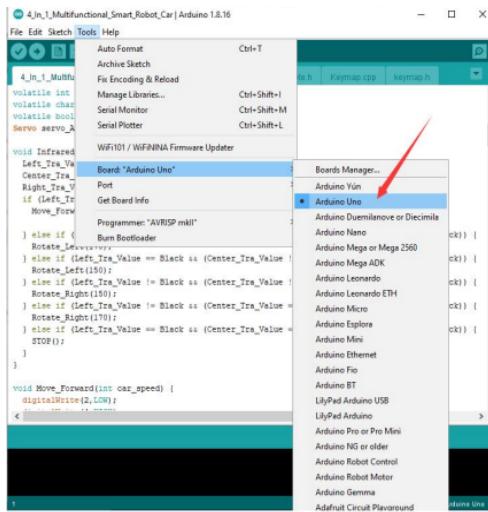
[upload main core.mp4 \(dropbox.com\)](#)



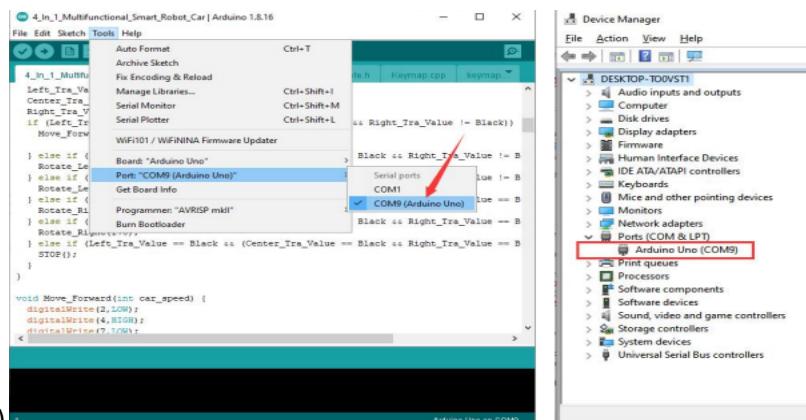
② Conecte la placa Arduino UNO a su computadora. (Puede ser que la fuente de alimentación de la interfaz USB de la computadora es insuficiente, puedes encender el interruptor de encendido del coche robot al mismo tiempo.)



③ Seleccione su placa en Herramientas > menú Placa>Arduino UNO

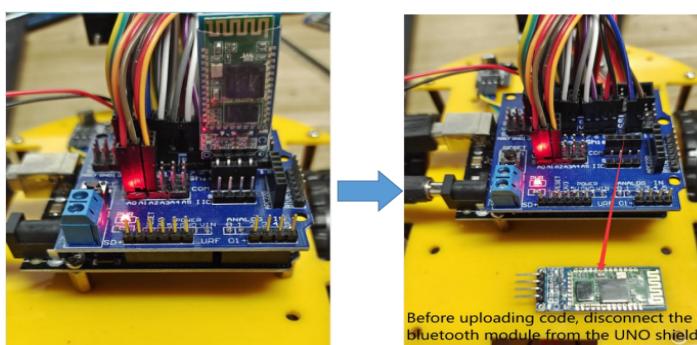


④ Seleccione el puerto (si no ve el puerto COM en su IDE de Arduino, debe instalar Arduino UNO)

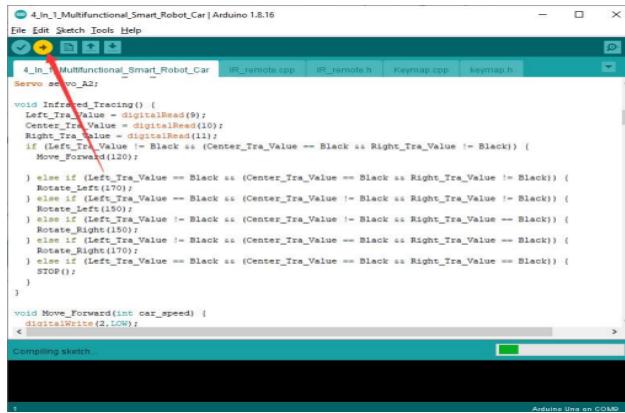


Conductores)

⑤ Importante: antes de cargar el código, desconecte el módulo Bluetooth del escudo V5.
Después de cargar correctamente el código, instale el módulo Bluetooth en la placa de expansión V5. De lo contrario, el módulo Bluetooth ocupa la interfaz de comunicación (RX TX) para cargar código, y Arduino IDE informará un error que indica que la carga falló.



⑥Haga clic en el botón Cargar en el IDE de Arduino. Espere unos segundos mientras el código se compila y se carga en tu tablero.



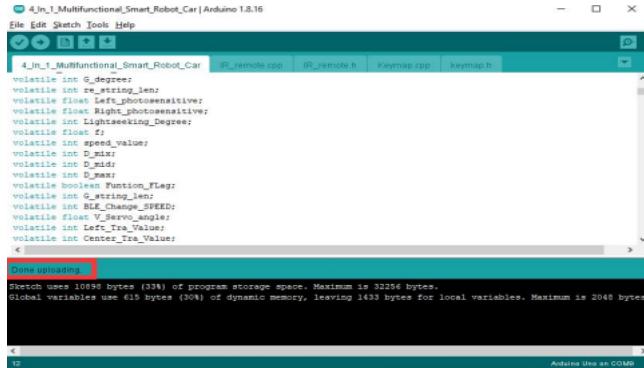
Si todo salió como se esperaba, debería ver el mensaje "Carga finalizada". mensaje. Si el IDE de Arduino informa, errores tal vez te perdiste algunos pasos. La guía de introducción a Arduino es la siguiente [Cargar un boceto en Arduino IDE Errores al cargar un boceto](#)

Primeros pasos

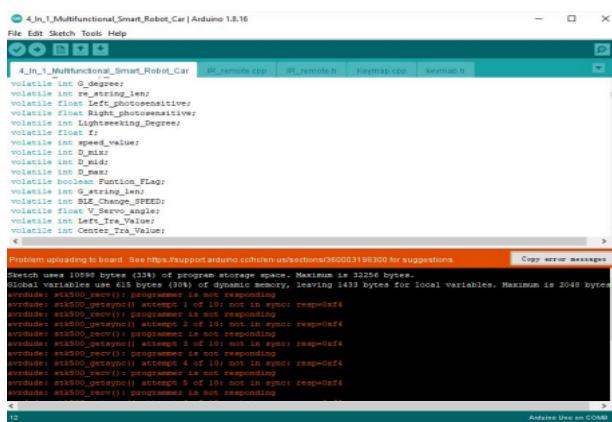
[Upload a sketch in Arduino IDE – Arduino Help Center](#)

[Errors when uploading a sketch – Arduino Help Center](#)

[Getting Started with Arduino products | Arduino](#)



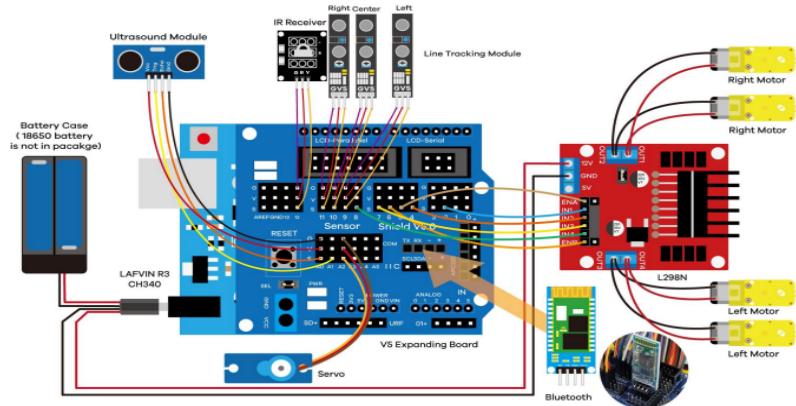
Si se informa el error avrdude: stk500_recv(): el programador no responde. La razón frecuente es que el RX TX del puerto del dispositivo UNO está ocupado, debe realizar los pasos ⑤.



⑤ Paso 5: Coche robot de control remoto por infrarrojos

Verifique nuevamente si el cableado del circuito es correcto, preste atención a las marcas positivas y negativas de cada interfaz. Encienda el interruptor de alimentación.

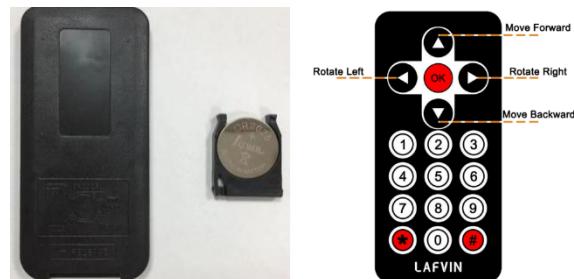
[IR Control.mp4 \(dropbox.com\)](#)



Seleccione la función de control remoto por infrarrojos después de cargar con éxito el código de Arduino UNO.

Nota: ① Debido a los requisitos del transporte aéreo, es posible que el control remoto por infrarrojos no contenga la batería CR2025 3V. Debe preparar la batería usted mismo.

② El transmisor y el receptor del control remoto por infrarrojos están en la misma línea recta, lo que mejorará efecto de control.



¿El coche robot no se mueve? Puede haber las siguientes razones:

① Debe funcionar con baterías con un voltaje >7,4 V. Encienda la energía de la batería

Interruptor. Solo la fuente de alimentación USB de 5V no puede accionar el motor. Debido al transporte aéreo, es posible que la batería no esté incluida en el kit, deberás comprarla tú mismo.

② Cuando el módulo receptor de infrarrojos recibe la instrucción, la luz indicadora de señal debería parpadear, si no, la posible razón es que el cableado del módulo receptor de infrarrojos está equivocado.

[1.gif \(750x1333\) \(shopify.com\)](#)

③ El cableado del módulo L298N del controlador del motor es incorrecto o está suelto.

④ La carga del código falló o se cargó un código incorrecto

[Errors when uploading a sketch – Arduino Help Center](#)

The robot car is not moving? There may be the following reasons:

- ① It must be powered by batteries with a voltage >7.4V. Turn on the battery power switch. Only 5V USB power supply cannot drive the motor. Due to air transportation, the battery may not be included in the kit, you need to buy the battery yourself.
- ② When the infrared receiving module receives the instruction, [the signal indicator light should flash](#), if not, the possible reason is that the [wiring of the infrared receiving module is wrong](#).
- ③ The [wiring of the motor driver L298N module is wrong](#) or the wiring is loose.
- ④ The code [upload failed](#) or [an incorrect code was uploaded](#)

Paso 6: la aplicación se conecta al módulo Bluetooth

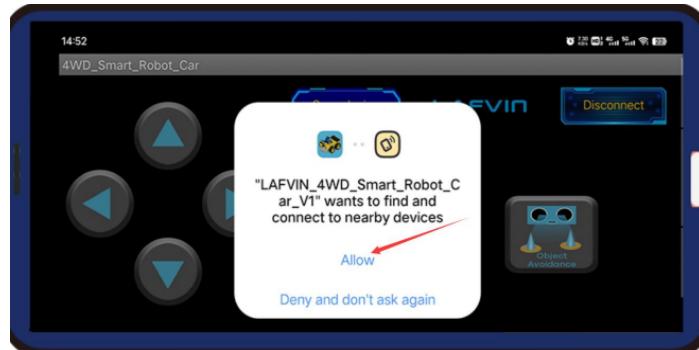
① Instale el archivo .apk en su teléfono (solo es compatible con el sistema Android)

[app connect to bluetooth.mp4 \(dropbox.com\)](#)

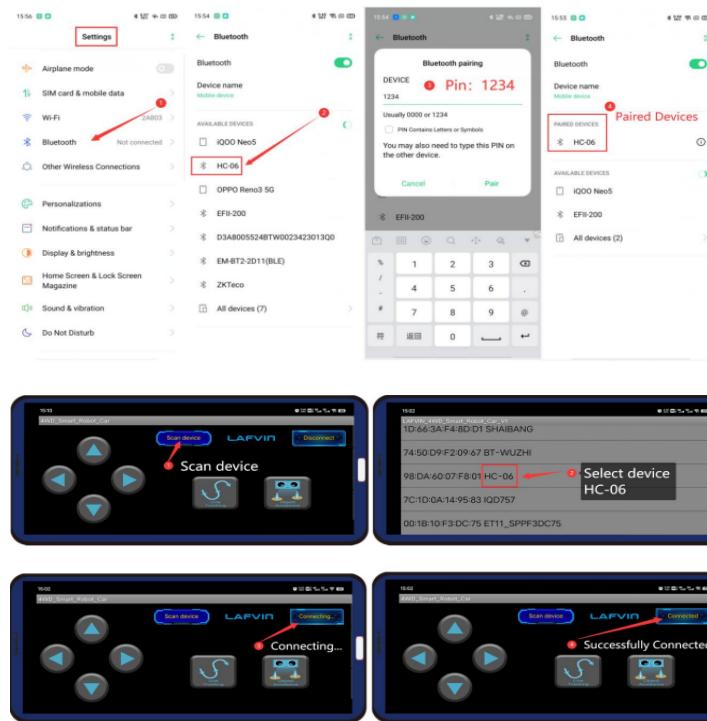


②Permita que la aplicación busque y se conecte a dispositivos cercanos

Consejo: Si la versión de Android del teléfono > 12 o la versión del SDK de Android > 31, la aplicación debe obtener permiso para acceder a dispositivos cercanos; de lo contrario, la función de comunicación Bluetooth no se puede



③La aplicación se conecta al módulo Bluetooth HC-06

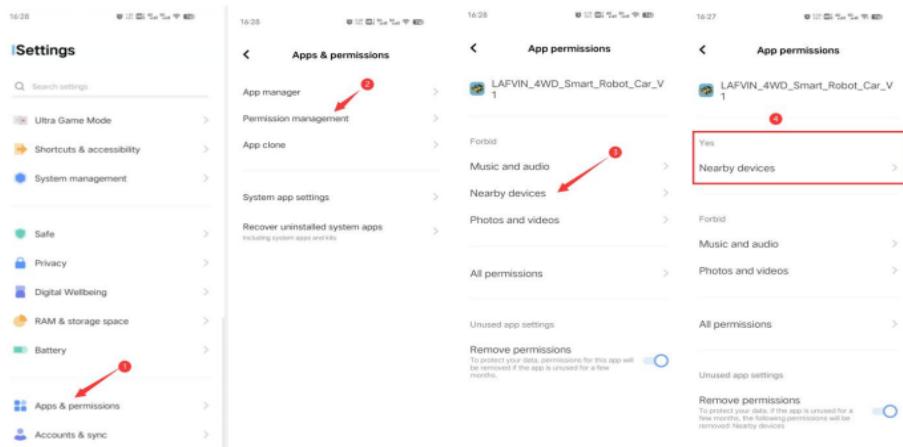


Consejo: Si informa Error 507: No se puede conectar, ¿está encendido el dispositivo?

La posible razón es que ha seleccionado el dispositivo bluetooth incorrecto o que el HC-06 no está disponible estado. Intente reiniciar la fuente de alimentación cuando la luz LED integrada del módulo HC-06 parpadee rápidamente (Estado disponible). Repita el paso anterior.



Consejo: Si hace clic para buscar dispositivos, la lista de dispositivos no tendrá ningún dispositivo seleccionable. La aplicación ,necesita permiso para acceder a dispositivos cercanos.



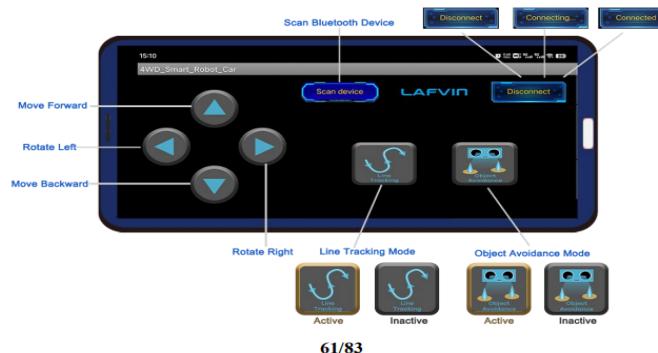
② Paso 7: Modo multifunción de control remoto de la aplicación

☞ Modo de control de dirección

☞ Modo de seguimiento de línea

☞ Modo de evitación de objetos

[App control.mp4 \(dropbox.com\)](#)



Direction Control Mode:

[App control.mp4 \(dropbox.com\)](#)

A screenshot of the mobile application interface in Direction Control Mode. It shows a 4x4 grid of control buttons: Move Forward (top), Rotate Left (left), Move Backward (bottom), and Rotate Right (right). Above the grid are buttons for 'Scan Bluetooth Device', 'Disconnect', 'Connecting...', and 'Connected'. Below the grid are icons for 'Line Tracking Mode' (Active and Inactive) and 'Object Avoidance Mode' (Active and Inactive).

The robot car is not moving? There may be the following reasons:

- ① It must be powered by batteries with a voltage >7.4V. Turn on the battery power switch. Only 5V USB power supply cannot drive the motor. Due to air transportation, the battery may not be included in the kit, you need to buy the battery yourself.
- ② The [wiring of the motor driver L298N module is wrong](#) or the wiring is loose.
- ③ The code [upload failed](#) or [an incorrect code was uploaded](#)

¿El coche robot no se mueve? Puede haber las siguientes razones:

① Debe funcionar con baterías con un voltaje >7,4 V. Encienda la energía de la batería

Interruptor. Solo la fuente de alimentación USB de 5V no puede accionar el motor. Debido al transporte aéreo, el

Es posible que la batería no esté incluida en el kit, deberás comprarla tú mismo.

② El cableado del módulo L298N del controlador del motor es incorrecto o está suelto.

③ La carga del código falló o se cargó un código incorrecto

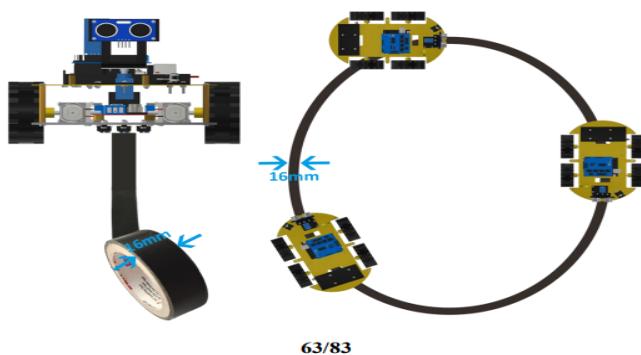
Modo de seguimiento de línea

[App control.mp4 \(dropbox.com\)](#)

Primero coloque el automóvil en la vía manual. Cuando se presiona la tecla del control remoto de la aplicación, el automóvil

activar el modo de seguimiento de línea. En este modo, el coche circulará por la pista. Al presionar la tecla

Del control remoto de la aplicación u otras teclas, saldrá del modo de seguimiento de línea.



¿No funciona perfectamente?

① Primero verifique si el cableado del sensor de seguimiento de línea es correcto.

S(Izquierda)-->D7 S(centro)-->D8 S(derecha)-->D9

② Luego verifique las luces LED de estado

Cada sensor debería funcionar normalmente de la siguiente manera:

blanco detectado-->LED de estado ENCENDIDO Negro detectado-->LED de estado APAGADO

③ Finalmente ajuste la sensibilidad

Dado que los sensores del módulo de seguimiento de línea se ven muy afectados por el medio ambiente, si el automóvil no Funciona bien bajo el módulo de seguimiento de línea, puede usar el potenciómetro para ajustar los umbrales del sensor para que

que puede permitir que el coche funcione

[Line Follow Module Test1.gif \(dropbox.com\)](#)

Not working perfectly?

①First check if the wiring of the line-tracing sensor is correct?

S(Left)-->D7 S(center)-->D8 S(right)-->D9

②Then check Status LED Lights

Each sensor should work normally as follows:



white detected-->Status LED ON Black detected-->Status LED OFF

③Finally adjust the sensitivity

Since the sensors on the line-tracking module are greatly affected by the environment, if the car does not work well under the line-tracking module, you can use the potentiometer to adjust the sensor thresholds so that it can enable the car to operate at its optimum level.

Modo de evitación de objetos

El automóvil entrará en el modo para evitar objetos al presionar la tecla del control remoto de la aplicación.
El automóvil

avanzará automáticamente hasta encontrar obstáculos y girará automáticamente en la dirección sin obstáculos para seguir avanzando al encontrar obstáculos. Al pulsar la tecla del control de la aplicación remota u otras teclas, saldrá del modo de evitación de objetos.

Modo de evitación de objetos

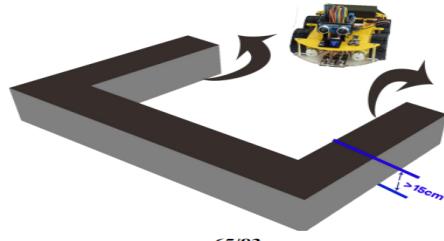
[App control.mp4 \(dropbox.com\)](#)

> Object-Avoidance Mode



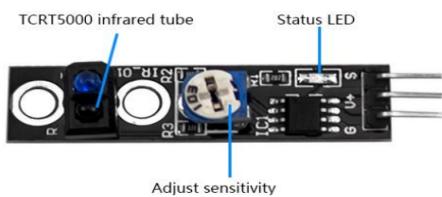
The car will enter the object-avoidance mode when pressing the key of the APP remote control. The car will move forward automatically until encountering obstacles. And it will automatically turn to the direction without obstacles to continue moving forwards when encountering obstacles. When pressing the key of the APP remote control or other keys, it will exit the object-avoidance mode.

Tip: Can't work correctly? Try to restart the battery power. Or test the ultrasonic sensor module separately.



65/83

Sensor de seguimiento de línea



El sensor de seguimiento de línea tiene tres pines: uno para tierra (G), uno para Vcc (V+) y otro para la salida de señal (S). El azul

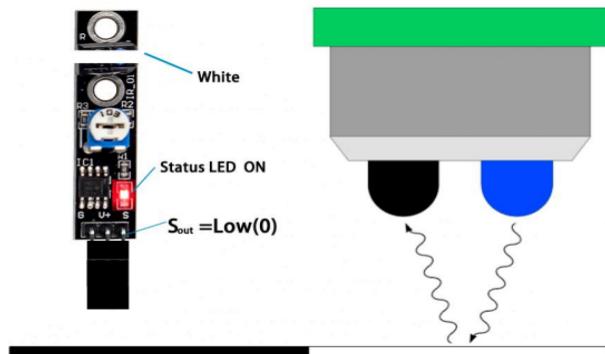
El LED es el LED de transmisión de infrarrojos y el LED negro es el LED de recepción de infrarrojos. El sensor de seguimiento de línea tiene un alcance de 0,5 cm.

a 1,5 cm. El IC de la placa es un comparador LM393, que convierte la señal analógica del LED receptor en una señal digital.

que se envía al pin de salida de señal del sensor

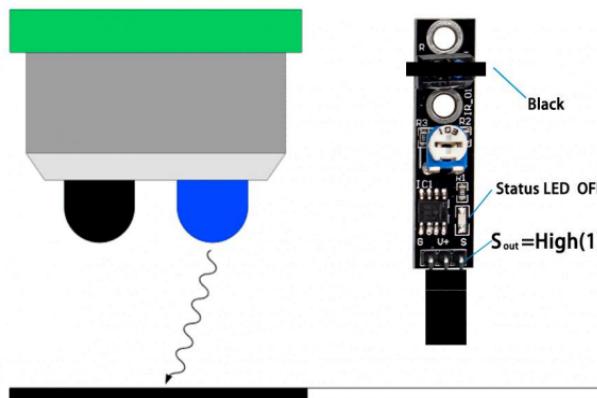
Cómo funcionan los sensores de seguimiento

Las superficies blancas reflejan la luz IR y las superficies oscuras absorben la luz IR. Cuando el sensor de seguimiento se coloca sobre una superficie blanca, la luz IR se refleja en la superficie. Se detecta la luz IR reflejada



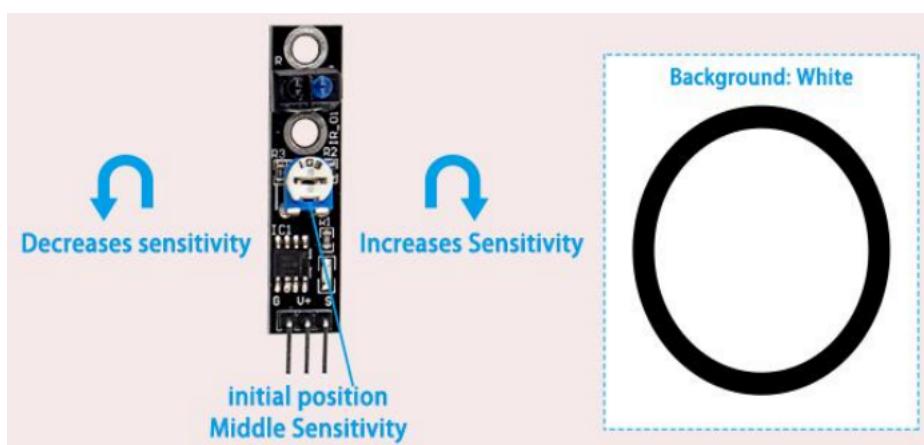
Pero si el sensor de seguimiento se coloca sobre una superficie de color oscuro, la superficie absorbe la luz IR. Entonces la recepción

El LED no detectará ninguna luz IR.



Ajustar la sensibilidad

La posición inicial del potenciómetro es de sensibilidad media y la línea negra puede lograr el mejor efecto sobre el blanco.fondo. Si el color de fondo no es blanco, es posible que necesite aumentar la sensibilidad para



Controlador de motor L298N

Hay muchas formas de controlar un motor de CC. El método que usaremos aquí es adecuado para la mayoría de los motores de aficionados, que requieren

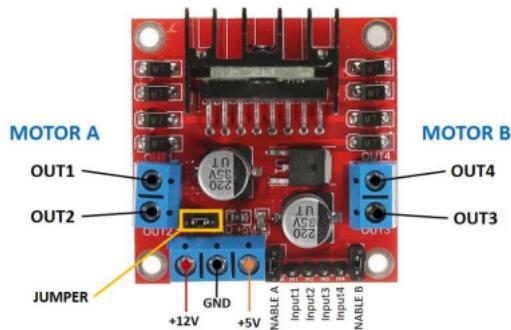
6V o 12V para operar. Usaremos el controlador de motor L298N que puede manejar hasta 3A a 35V. Además, nos permite conducir dos DC.

motores simultáneamente, lo cual es perfecto para construir un robot. El controlador del motor L298N se muestra en la siguiente figura:



Configuración de pines del controlador del motor L298N

Echemos un vistazo a la distribución de pines del controlador del motor L298N y veamos cómo funciona.



- OUT1: DC motor A + terminal
- OUT2: DC motor A - terminal
- OUT3: DC motor B + terminal
- OUT4: DC motor B - terminal

El controlador del motor tiene un bloque de dos terminales en cada lado para cada motor.

En la parte inferior tiene un bloque de tres terminales con +12 V, GND y +5 V. El bloque de terminales de +12 V se utiliza para encender el

motores. El terminal +5V se utiliza para encender el chip L298N. Sin embargo, si el puente está en su lugar, el chip se alimenta utilizando el

fuente de alimentación del motor y no es necesario suministrar 5 V a través del terminal +5 V. Nota: si suministra más de 12 V, deberá quitar el puente y suministrar 5 V al terminal +5 V. Es importante tener en cuenta que a pesar del nombre del terminal de +12 V, con la configuración que usaremos aquí (con el puente en su lugar) puedes

Suministre cualquier voltaje entre 6V y 12V. En este tutorial usaremos 4 baterías AA de 1.5V que combinadas producen aproximadamente

6V, pero puedes usar cualquier otra fuente de alimentación adecuada. En resumen: ☐ +12V: El terminal de +12V es donde debes conectar tu fuente de alimentación.

☐ GND: fuente de alimentación GND

☐ +5V: proporciona 5V si se quita el puente. Actúa como una salida de 5 V si el puente está colocado

☐ Puente: puente en su lugar: utiliza la fuente de alimentación del motor para encender el chip. Puente quitado: necesitas proporcionar 5V

al terminal +5V. Si suministra más de 12V, deberá quitar el puente.

En la parte inferior derecha tienes cuatro pines de entrada y dos terminales de habilitación. Los pines de entrada se utilizan para controlar la dirección de su

Los motores de CC y los pines de habilitación se utilizan para controlar la velocidad de cada motor.

- IN1: Input 1 for Motor A
- IN2: Input 2 for Motor A
- IN3: Input 1 for Motor B
- IN4: Input 2 for Motor B
- EN1: Enable pin for Motor A
- EN2: Enable pin for Motor B

There are jumper caps on the enable pins by default. You need to remove those jumper caps to control the speed of your motors.

At the bottom you have a three terminal block with **+12V**, **GND**, and **+5V**. The **+12V** terminal block is used to power up the motors. The **-5V** terminal is used to power up the L298N chip. However, if the jumper is in place, the chip is powered using the motor's power supply and you don't need to supply 5V through the **+5V** terminal.

Note: if you supply more than 12V, you need to remove the jumper and supply 5V to the +5V terminal.

It's important to note that despite the +12V terminal name, with the setup we'll use here (with the jumper in place) you can supply any voltage between 6V and 12V. In this tutorial will be using 4 AA 1.5V batteries that combined output approximately 6V, but you can use any other suitable power supply.

In summary:

- **+12V**: The +12V terminal is where you should connect your power supply
- **GND**: power supply GND
- **+5V**: provide 5V if jumper is removed. Acts as a 5V output if jumper is in place
- **Jumper**: jumper in place – uses the motors power supply to power up the chip. Jumper removed: you need to provide 5V to the +5V terminal. If you supply more than 12V, you should remove the jumper

At the bottom right you have four input pins and two enable terminals. The input pins are used to control the direction of your DC motors, and the enable pins are used to control the speed of each motor.

- IN1: Input 1 for Motor A
- IN2: Input 2 for Motor A
- IN3: Input 1 for Motor B
- IN4: Input 2 for Motor B
- EN1: Enable pin for Motor A
- EN2: Enable pin for Motor B

There are jumper caps on the enable pins by default. You need to remove those jumper caps to control the speed of your motors.

Controle motores de CC con el L298N

Ahora que está familiarizado con el controlador de motor L298N, veamos cómo usarlo para controlar sus motores de CC. Habilitar pines

Los pines de habilitación son como un interruptor de ENCENDIDO y APAGADO para sus motores. Por ejemplo: ☐ Si envía una señal ALTA al pin enable 1, el motor A está listo para ser controlado y a la velocidad máxima; ☐ Si envía una señal BAJA al pin de habilitación 1, el motor A se apaga; ☐ Si envía una señal PWM, puede controlar la velocidad del motor. La velocidad del motor es proporcional al ciclo de trabajo. Sin embargo, tenga en cuenta que para ciclos de trabajo pequeños, es posible que los motores no giren y emitan un zumbido continuo.

SIGNAL ON THE ENABLE PIN	MOTOR STATE
HIGH	Motor enabled
LOW	Motor not enabled
PWM	Motor enabled: speed proportional to duty cycle

Pines de entrada

Los pines de entrada controlan la dirección en la que giran los motores. Las entradas 1 y 2 controlan el motor A, y las entradas 3 y 4 controlan

motor B. ☐ Si aplica BAJO a la entrada 1 y ALTO a la entrada 2, el motor girará hacia adelante;

☐ Si aplica energía al revés: ALTA a la entrada 1 y BAJA a la entrada 2, el motor girará hacia atrás. Motor B

se puede controlar usando el mismo método pero aplicando ALTO o BAJO a la entrada 3 y a la entrada 4. Control de 2 motores CC: ideal para construir un robot

Si desea construir un automóvil robot con 2 motores de CC, estos deben girar en direcciones específicas para que el robot vaya hacia la izquierda, hacia la derecha, hacia adelante o hacia atrás.

Por ejemplo, si desea que su robot avance, ambos motores deben girar hacia adelante. Para hacerlo retroceder, ambos

debe girar hacia atrás. Para girar el robot en una dirección, es necesario hacer girar el motor opuesto más rápido. Por ejemplo, para hacer que el robot gire a la derecha, habilite

el motor de la izquierda y desactive el motor de la derecha. La siguiente tabla muestra las combinaciones de estados de los pines de entrada para el

robot

DIRECTION	IN 1(D2)	IN 2(D4)	IN 3 (D7)	IN 4(D8)
Forward	0	1	0	1
Backward	1	0	1	0
Rotate_Right	1	0	0	1
Rotate_Left	0	1	1	0
Stop	1	1	1	1

Módulo de sensor ultrasónico

Si la función para evitar obstáculos no puede funcionar correctamente, puede probar el módulo del sensor ultrasónico por separado. (No en orden

Para verse afectado por otros factores, es necesario desconectar la conexión de otros módulos. Por ejemplo, módulo de infrarrojos, línea

módulo de seguimiento, módulo de accionamiento del motor L298N)

Descripción

El sensor ultrasónico HC-SR04 utiliza un sonar para determinar la distancia a un objeto. Este sensor lee desde 2 cm hasta 400 cm.

(0,8 pulgadas a 157 pulgadas) con una precisión de 0,3 cm (0,1 pulgadas), lo cual es bueno para la mayoría de los proyectos de aficionados. Además, este

El módulo en particular viene con módulos transmisores y receptores ultrasónicos. La siguiente imagen muestra el sensor ultrasónico HC-SR04.



Características

Aquí hay una lista de algunas de las características y especificaciones del sensor ultrasónico HC-SR04:

- Fuente de alimentación: +5 V CC

- Corriente de reposo: <2 mA

- Corriente de trabajo: 15 mA

- Ángulo efectivo: <15°

- Distancia de alcance: 2 cm – 400 cm/1" – 13 pies ■ Resolución: 0,3 cm

- Ángulo de medición: 30 grados
- Ancho de pulso de entrada del disparador: pulso TTL de 10uS
- Señal de salida de eco: pulso TTL proporcional al rango de distancia
- Dimensión: 45 mm x 20 mm x 15 mm

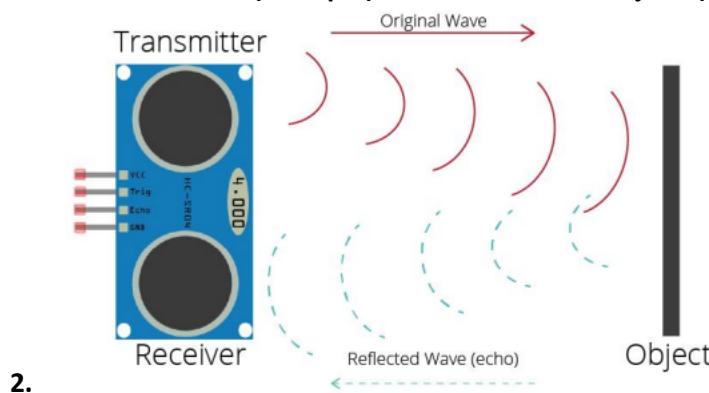
Features

Here's a list of some of the HC-SR04 ultrasonic sensor features and specs:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1" – 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS TTL pulse
- Echo Output Signal: TTL pulse proportional to the distance range
- Dimension: 45mm x 20mm x 15mm

¿Como funciona? El sensor ultrasónico utiliza un sonar para determinar la distancia a un objeto.
Esto es lo que sucede:

- El transmisor de ultrasonido (pin de activación) emite un sonido de alta frecuencia (40 kHz).
- El sonido viaja por el aire. Si encuentra un objeto, regresa al módulo.
- El receptor de ultrasonidos (echo pin) recibe el sonido reflejado (eco).



El tiempo entre la transmisión y recepción de la señal nos permite calcular la distancia a un objeto.
Esto es

possible porque conocemos la velocidad del sonido en el aire. Aquí está la fórmula:

$$\text{distance} = (\text{traveltime}/2) \times \text{speed of sound}$$

The speed of sound is: 343m/s = 0.0343 cm/uS = 1/29.1 cm/uS

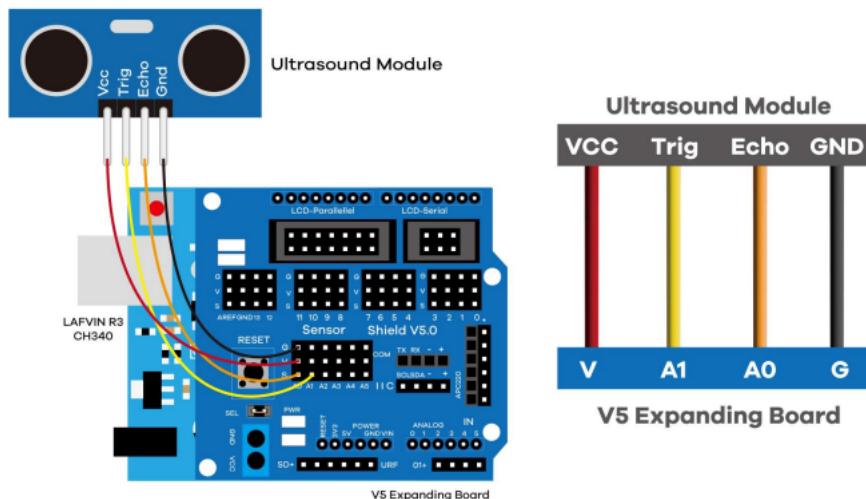
speed of sound in the air at 20°C (68°F) = 343m/s

distancia = (tiempo de viaje/2) x velocidad del sonido

La velocidad del sonido es: 343 m/s = 0,0343 cm/us = 1/29,1 cm/us

velocidad del sonido en el aire a 20°C (68°F) = 343m/s

Arduino Shield V5.0 con cableado de sensor HC-SR04



código

Haga clic en el ícono Archivo>>Abrir para abrir el archivo Test Code\Ultrasonic_Sensor_Module.ino.

Click the File>>Open icon to open Test Code\Ultrasonic_Sensor_Module.ino file.

```

Ultrasonic_Sensor_Module | Arduino 1.8.16
File Edit Sketch Tools Help
Ultrasonic_Sensor_Module §
digitalWrite(A1, HIGH);
delayMicroseconds(10);
digitalWrite(A1, LOW);
float distance = pulseIn(A0, HIGH) / 58.00;
delay(10);
return distance;

void setup() {
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode (A1, OUTPUT);
  pinMode (A0, INPUT);
}

void loop() {
  distance = checkdistance();
  Serial.print(distance);
  Serial.println("cm");
  delay(1000);
}

```

Haga clic, suba el boceto a Arduino UNO

```

Ultrasonic_Sensor_Module | Arduino 1.8.16
File Edit Sketch Tools Help
Upload
Ultrasonic_Sensor_Module
Trig : Trigger (INPUT) - A1
Echo: Echo (OUTPUT) - A0
GND: GND
*/
long distance;
float checkdistance()
{
  digitalWrite(A1, LOW);
  delayMicroseconds(2);
  digitalWrite(A1, HIGH);
  delayMicroseconds(10);
  digitalWrite(A1, LOW);
  float distance = pulseIn(A0, HIGH) / 58.00;
  delay(10);
  return distance;
}
Done uploading
Sketch uses 3098 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 190 bytes (9%) of dynamic memory, leaving 1658 bytes for local variables. Maxi

```

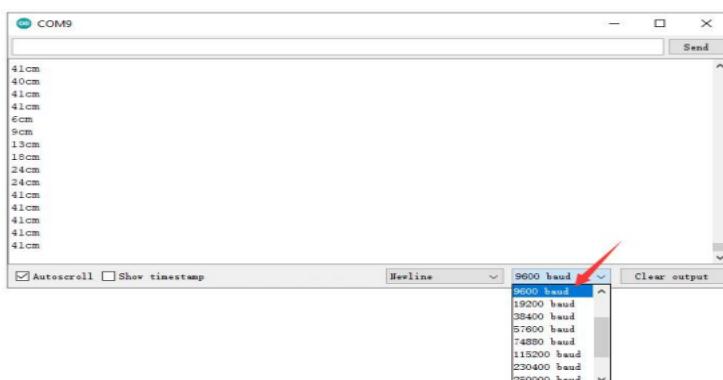
Haga clic en , abra el Monitor serie a una velocidad de 9600 baudios. La distancia al objeto más cercano se imprime en la ventana del Monitor serie. Cuando el objeto se mueve, el

La distancia del monitor impreso cambiará.

Then Click  open the Serial Monitor at a baud rate of 9600.



The distance to the nearest object is printed in the Serial Monitor window. When the object moves, the distance of the monitor printed will change.

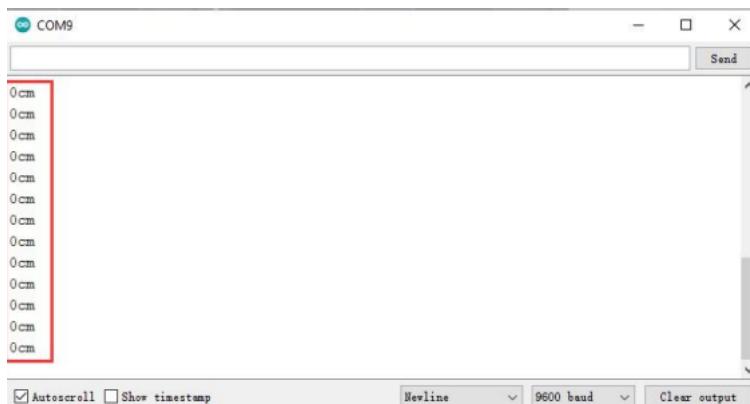


82/83

Consejo: Si el valor de impresión es 0 cm, el motivo del error puede ser:

①Cableado incorrecto. ②El voltaje proporcionado por el cable de datos USB es insuficiente y necesita usar la batería para suministrar

poder al mismo tiempo.



83/83

Sensor ultrasónico

Un sensor ultrasónico es un dispositivo que utiliza ondas de sonido de alta frecuencia para medir la distancia a un objeto. Estas ondas son inaudibles para el oído humano y se reflejan en el objeto, permitiendo al sensor calcular el tiempo que tarda en recibir el eco.

Un sensor ultrasónico tiene ventajas sobre otros tipos de sensores, como los infrarrojos (IR), ya que puede detectar objetos de cualquier color, transparencia o superficie, y puede funcionar en ambientes con polvo, humedad o niebla. Un sensor ultrasónico tiene un solo elemento que emite y recibe el sonido, mientras que un sensor IR tiene un transmisor y un receptor de luz infrarroja.

Durante este proyecto, usamos el sensor ultrasónico 1PCS, que es un dispositivo electrónico que como mencionamos anteriormente, emplea ondas sonoras de alta frecuencia para detectar la presencia de objetos en su entorno.

El sensor ultrasónico 1PCS se compone de dos partes principales: un emisor y un receptor. El emisor contiene un cristal piezoelectrico que al recibir una señal eléctrica vibra y genera una onda de sonido que viaja por el aire.

Por su parte, el receptor también contiene un cristal piezoelectrico que al recibir la onda de sonido reflejada la convierte en una señal eléctrica. El sensor también cuenta con un circuito eléctrico que procesa la señal y la envía a la placa Arduino UNO que controla el carro. De igual manera, cuenta con varias ventajas sobre otros tipos de sensores, como los ópticos o los infrarrojos. El sensor ultrasónico puede detectar objetos de cualquier color, transparencia, superficie o brillo, ya que solo depende del sonido.

Debido a todo lo dicho anteriormente, el sensor ultrasónico 1PCS es un componente esencial para el funcionamiento del carro robot inteligente autónomo, ya que le permite desplazarse por diferentes superficies, adaptarse a diferentes condiciones ambientales y ser controlado a distancia. Este sensor, es un ejemplo claro de cómo la electrónica, la programación y la robótica se combinan para crear proyectos innovadores y creativos, como este.

Ejemplos de código:

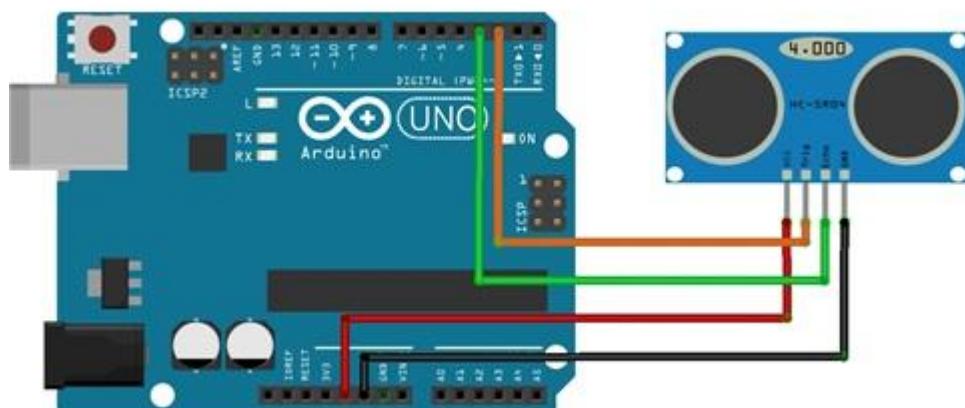
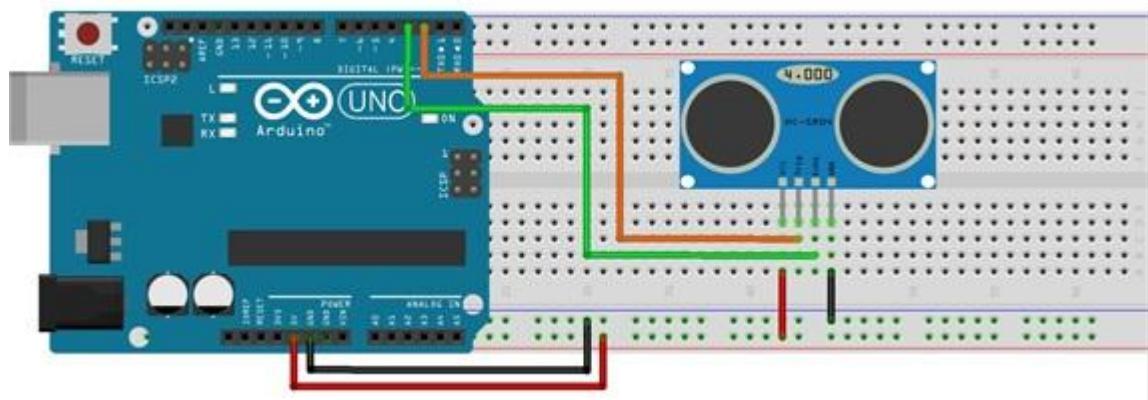
Este código lee la distancia medida por el sensor y la muestra en el monitor serie. Necesitas conectar el pin TRIG del sensor al pin 2 del Arduino y el pin ECHO al pin 3. El código no utiliza ninguna biblioteca externa.

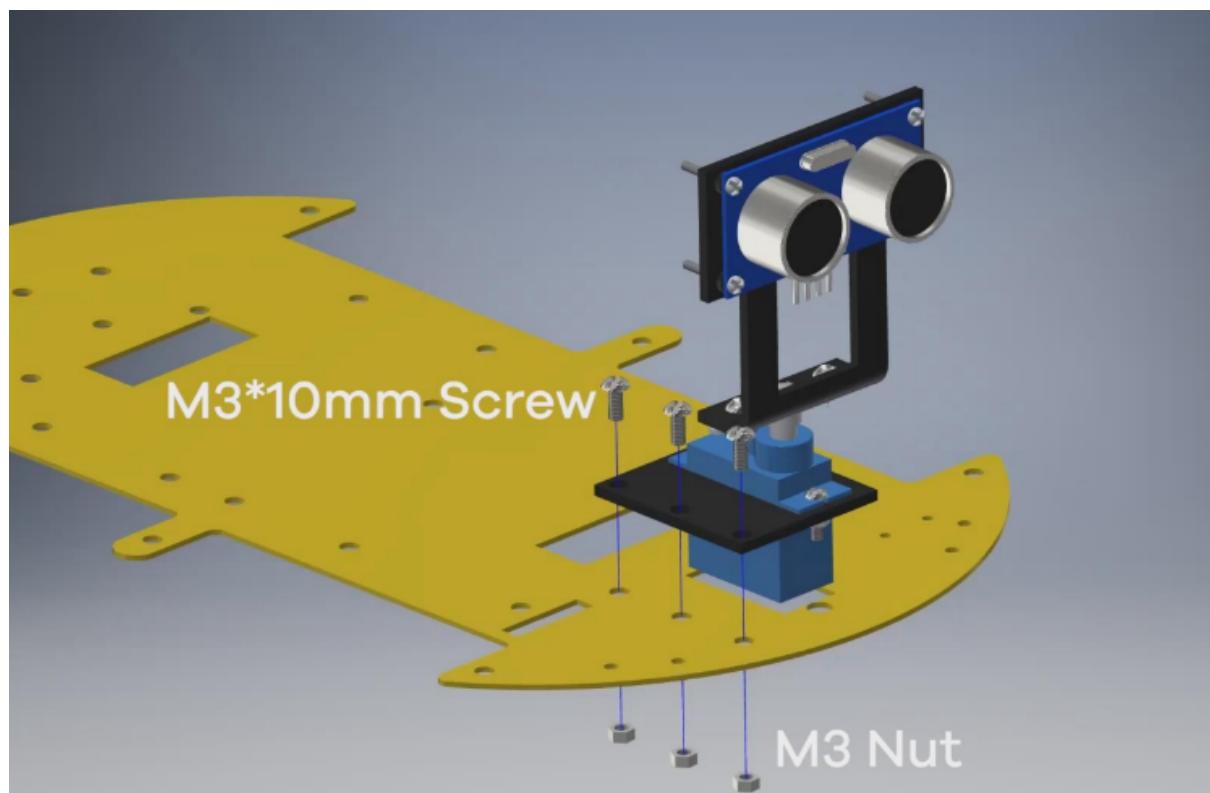
```
#define TRIG 2 // Define el pin del TRIG
#define ECHO 3 // Define el pin del ECHO
long Tiempo; // Variable para almacenar el tiempo
float Distancia; // Variable para almacenar la distancia

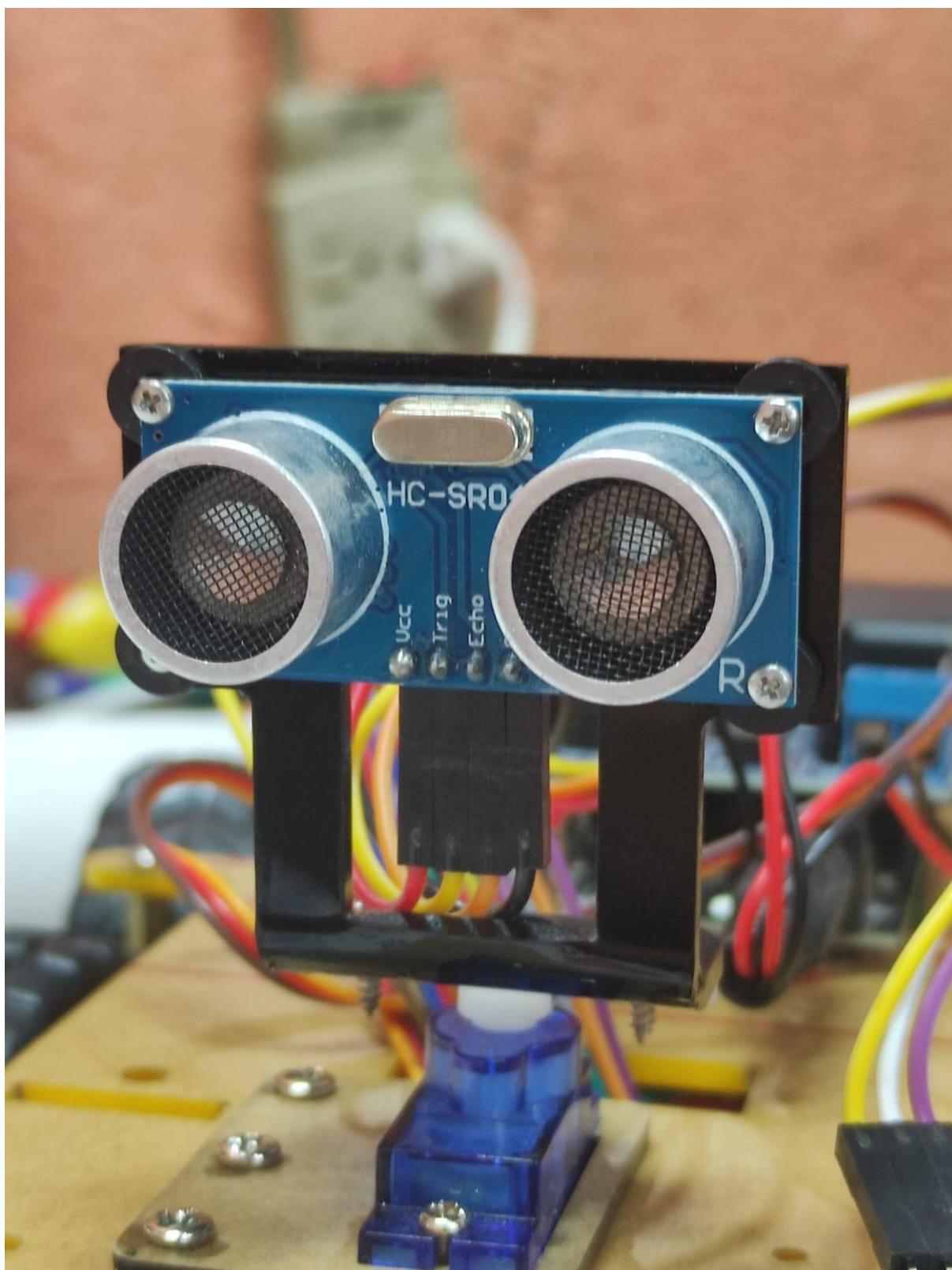
void setup () {
    Serial.begin (9600); // Inicia la comunicación serie
    pinMode(TRIG, OUTPUT); // Configura el pin del TRIG como salida
    pinMode(ECHO, INPUT); // Configura el pin del ECHO como entrada
}

void loop() {
    digitalWrite(TRIG, LOW); // Apaga el pulso del TRIG
    delayMicroseconds(2); // Espera 2 microsegundos
    digitalWrite(TRIG, HIGH); // Enciende el pulso del TRIG
    delayMicroseconds(10); // Espera 10 microsegundos
    digitalWrite(TRIG, LOW); // Apaga el pulso del TRIG
    Tiempo = pulseIn(ECHO, HIGH); // Lee el tiempo que tarda en volver el eco
    Distancia = (Tiempo/2) * 0.0343; // Calcula la distancia en cm
    Serial.print("Distancia: ");
    Serial.print(Distancia);
    Serial.println(" cm");
    delay(1000); // Espera un segundo
}
```

Imágenes







Control mediante control remoto y celular

El control mediante control remoto y celular sobre 4WD Smart Robot Car es realizado gracias a una placa Arduino UNO, que se encarga de procesar las señales y comandos que recibe el carro y de controlar los motores y los sensores. Arduino es una plataforma de hardware y software libre que permite crear proyectos de electrónica de forma fácil y rápida.

Antes de configurar el carro, se necesita un cable USB para conectar la placa Arduino al ordenador y cargar el código. Un entorno de desarrollo integrado (IDE) de Arduino para programar la placa Arduino con el lenguaje C++, el IDE de Arduino se puede descargar gratuitamente desde su página web oficial.

De igual manera para esto, se ocupa un teléfono celular con sistema operativo Android que tenga instalada la aplicación Archy Control para celular, que permite controlar el carro mediante bluetooth con una interfaz gráfica de usuario. Esta aplicación se puede descargar gratuitamente desde Google Play.

Una vez obtenido todo, este es el procedimiento para configurar el funcionamiento del carro una vez ensamblado:

- Conectar la placa Arduino al ordenador con el cable USB y abrir el IDE de Arduino.
- Descargar el código fuente del proyecto. El código fuente es un archivo con extensión .ino que contiene las instrucciones que se ejecutan en la placa Arduino. El código fuente se compone de varias partes, como las variables, las constantes, las funciones, los bucles, las condiciones, etc. El código fuente se puede modificar y personalizar según las preferencias y necesidades del usuario.
- Cargar el código fuente en la placa Arduino usando el IDE de Arduino. Para ello, se debe seleccionar el tipo de placa Arduino y el puerto de comunicación en el menú Herramientas, y luego presionar el botón Cargar. El IDE de Arduino verificará el código fuente y lo transferirá a la placa Arduino. Si el proceso es exitoso, se mostrará un mensaje de “Carga completa” en la parte inferior del IDE de Arduino.
- Desconectar la placa Arduino del ordenador y conectarla a la batería del carro.

- Encender el carro y el módulo bluetooth. El módulo bluetooth tiene un LED que parpadea cuando está encendido y listo para emparejarse con otro dispositivo.
- Abrir la aplicación Archy Control para celular en el teléfono celular y emparejarlo con el módulo bluetooth del carro. Para ello, se debe activar el bluetooth del teléfono celular y buscar el dispositivo con el nombre “HC-06” o similar. Luego, se debe introducir el código “1234” o “0000” para establecer la conexión.
- Una vez conectado, el LED del módulo bluetooth dejará de parpadear y se quedará fijo.
- Usar la aplicación Archy Control para celular para controlar el carro mediante bluetooth con los motores y los sensores del carro.
- También se puede usar el control remoto para controlar el carro mediante infrarrojos con los botones de dirección, velocidad, sonido y luces. El control remoto tiene un emisor de infrarrojos que envía señales al módulo receptor de infrarrojos del carro, que las transmite a la placa Arduino, que las interpreta y ejecuta las acciones correspondientes en los motores y los sensores del carro.

Conclusión

El carro robótico puede realizar diferentes funciones y modos de operación, como evitar obstáculos, seguir líneas, emitir sonidos y luces, etc. Se puede modificar y personalizar el código fuente para añadir más funciones y características al carro. Realmente cuenta con un gran funcionamiento una vez configurado.

Ejemplos de código

A continuación se muestran algunos ejemplos de código:

Esta es la parte del código fuente que define las variables y las constantes que se usan en el proyecto:

```
// Definir las variables y las constantes
int motor1A = 3; // Pin de control del motor 1A
int motor1B = 4; // Pin de control del motor 1B
int motor2A = 5; // Pin de control del motor 2A
int motor2B = 6; // Pin de control del motor 2B
int motor3A = 7; // Pin de control del motor 3A
int motor3B = 8; // Pin de control del motor 3B
```

```

int motor4A = 9; // Pin de control del motor 4A
int motor4B = 10; // Pin de control del motor 4B
int buzzer = 11; // Pin de control del zumbador
int led = 12; // Pin de control del LED
int ir = 2; // Pin de entrada del receptor de infrarrojos
int trig = 13; // Pin de salida del sensor ultrasónico
int echo = A0; // Pin de entrada del sensor ultrasónico
int track1 = A1; // Pin de entrada del sensor de seguimiento de línea 1
int track2 = A2; // Pin de entrada del sensor de seguimiento de línea 2
int track3 = A3; // Pin de entrada del sensor de seguimiento de línea 3
int track4 = A4; // Pin de entrada del sensor de seguimiento de línea 4
int speed = 255; // Velocidad de los motores (0-255)
char command; // Comando recibido por bluetooth o infrarrojos
long duration; // Duración del pulso del sensor ultrasónico
int distance; // Distancia medida por el sensor ultrasónico
int mode = 0; // Modo de operación del carro (0: manual, 1: evita obstáculos, 2: sigue
líneas)

```

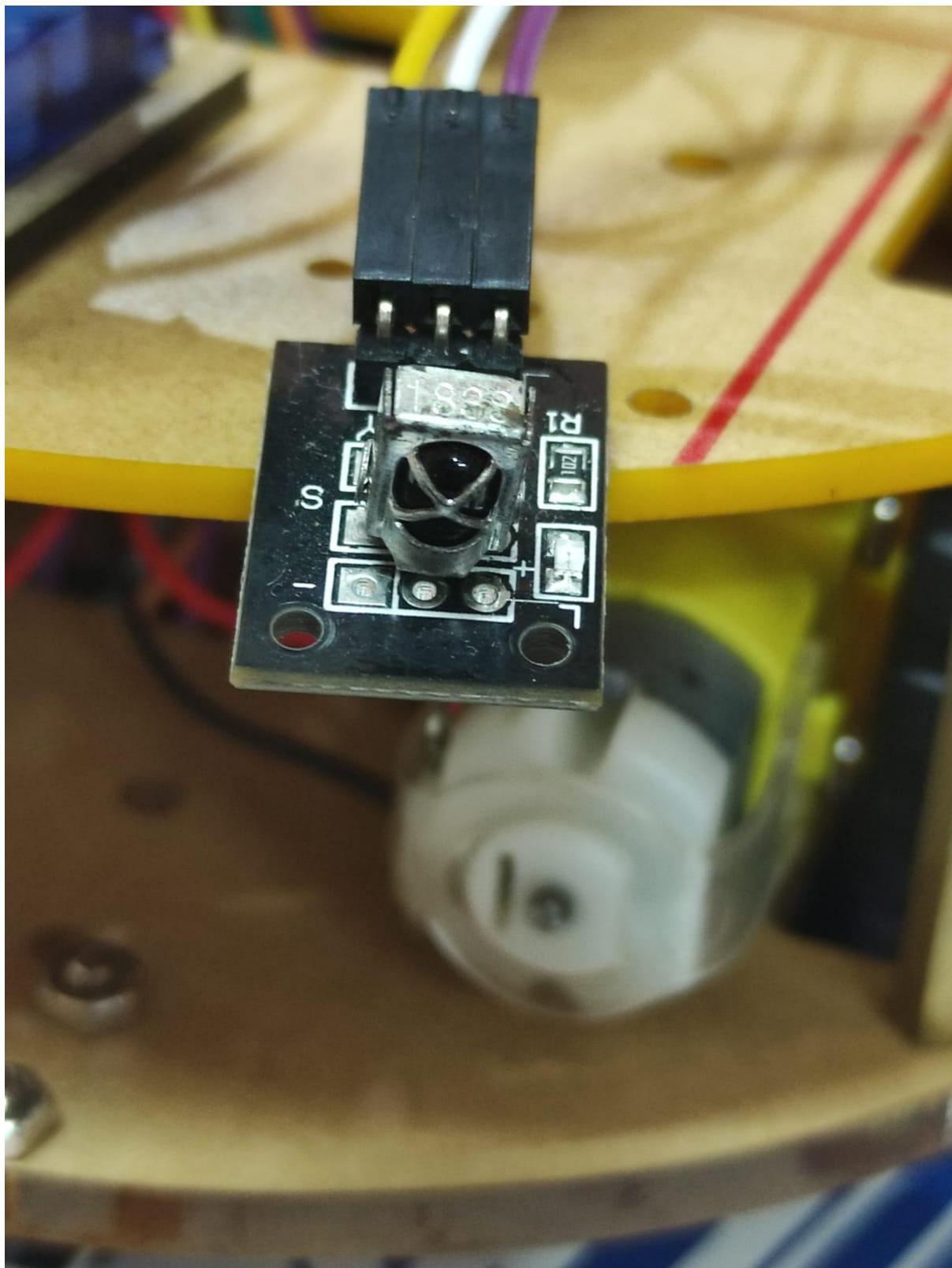
Esta es la parte del código fuente que configura los pines como entrada o salida y establece la comunicación serial con el módulo bluetooth:

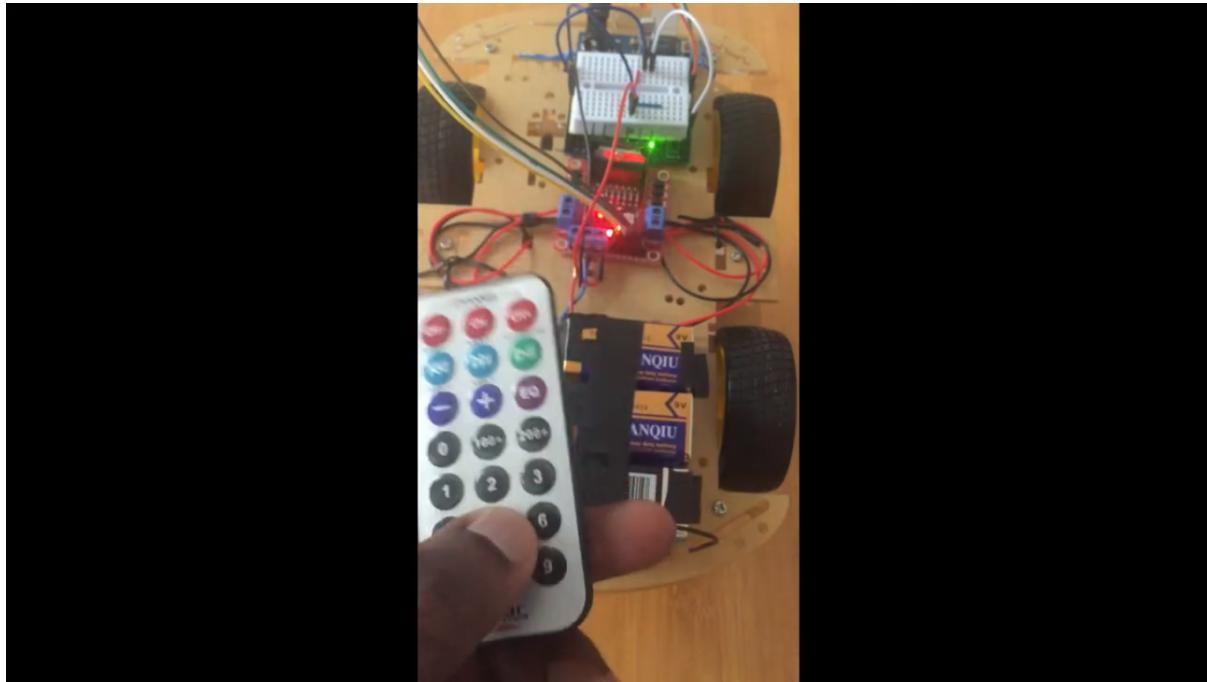
```

// Configurar los pines como entrada o salida
void setup() {
    pinMode(motor1A, OUTPUT);
    pinMode(motor1B, OUTPUT);
    pinMode(motor2A, OUTPUT);
    pinMode(motor2B, OUTPUT);
    pinMode(motor3A, OUTPUT);
    pinMode(motor3B, OUTPUT);
    pinMode(motor4A, OUTPUT);
    pinMode(motor4B, OUTPUT);
    pinMode(buzzer, OUTPUT)
}

```

Imágenes:





Referencias Bibliográficas:

https://naylampmechatronics.com/blog/10_tutorial-de-arduino-y-sensor-ultrasonico-hc-sr04.html

<https://www.mecatronicalatam.com/es/tutoriales/proyectos-con-arduino/sensor-ultrasonico-arduino/>

<https://www.youtube.com/watch?v=9soDTEJTnal>

https://es.wikipedia.org/wiki/Arduino_Uno

<https://proyectoarduino.com/arduino-uno-r3/>