

**Universidad Autónoma de Nuevo León**

**Facultad de Ingeniería Mecánica y Eléctrica**



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



**FIME**

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

<b>Unidad de aprendizaje:</b>	<b>COMPUTO INTEGRADO</b>		
<b>Grupo:</b>	<b>001</b>	<b>EQUIPO: 08</b>	<b>GRUPO:02</b>
<b>Periodo escolar:</b>	AGOSTO-DICIEMBRE 2025		
<b>Docente:</b>	HECTOR HUGO FLORES MORENO		
<b>Actividad:</b>	<b>MANUAL TECNICO</b>		

<b>Matricula</b>	<b>Apellidos</b>	<b>Nombre(s)</b>	<b>PE</b>
2073830	Hernandez Juárez	Salvador	ITS
2132048	Naranjo Rojas	Horacio	ITS
2131952	Meneses Garcia	Elisa del Carmen	ITS
2073443	Ibarra Alejo	Iván Alexis	ITS
2109620	Martínez Peralta	Jesús Rubelio	ITS

## MANUAL TECNICO

Hola [Administrador] Bienvenido al Manual Técnico de nuestro proyecto [[Robot multifuncional con Arduino](#)]. Este documento ha sido elaborado con el propósito de servir como una guía integral para la implementación, configuración, programación y mantenimiento del sistema robótico desarrollado. Nuestro objetivo es proporcionar toda la información técnica necesaria para comprender la estructura, el funcionamiento y las mejores prácticas de uso de este robot educativo.

El Robot multifuncional con Arduino es un sistema móvil inteligente diseñado para operar bajo tres modos de control principales:

- Evitación automática de obstáculos
- Control remoto por Bluetooth
- Control por comandos de voz

Este proyecto combina electrónica, programación embebida y control de movimiento, demostrando la integración efectiva entre hardware y software en entornos de robótica.

**REQUISITOS PREVIOS:** Antes de iniciar con la implementación, ensamblaje, programación o mantenimiento del robot multifuncional con Arduino, es indispensable asegurarse de contar con los requisitos técnicos esenciales, tanto en el ámbito de hardware (componentes electrónicos y materiales de construcción) como de software (herramientas de desarrollo y comunicación). Estos requisitos garantizan un entorno de trabajo estable, funcional y seguro, facilitando la correcta ejecución de pruebas, calibraciones y operaciones del sistema robótico.

### HARDWARE NECESARIO

Para garantizar un funcionamiento óptimo durante el desarrollo y las pruebas del robot, se debe contar con los siguientes **componentes electrónicos y materiales**:

COMPONENTE	DESCRIPCION TECNICA	FOTO
Placa Arduino UNO	Microcontrolador principal basado en ATmega328P. Gestiona la lectura de sensores, el control de motores y la comunicación serial con el módulo Bluetooth.	
Controlador de motor L293D	Módulo "puente H dual" para controlar el sentido y velocidad de los motores DC. Permite manejo independiente de dos canales de motor.	

Sensor ultrasónico HC-SR04	Sensor de medición de distancia por ultrasonido. Utilizado en el modo de evitación de obstáculos (rango aproximado: 2–400 cm).	
Módulo Bluetooth HC-05	Dispositivo de comunicación inalámbrica por protocolo Bluetooth clásico. Permite recibir comandos desde un smartphone en modo control remoto o por voz.	
Servomotor SG90 o equivalente	Micro servo de rotación limitada (0°–180°). Se utiliza para orientar el sensor ultrasónico hacia distintos ángulos durante el escaneo.	
Motores DC de engranajes (5V)	Motores de tracción con reducción interna. Proveen fuerza y control de movimiento del chasis.	
Ruedas (Tapas de botellón recicladas)	Ruedas plásticas acoplables a los ejes de los motores DC. Permiten el desplazamiento del robot.	

Soporte para baterías Li-ion	Porta baterías tipo 18650. Suministra energía a los motores y a la placa Arduino según el diseño energético.	
Baterías de ion de litio (18650)	Baterías recargables de alto rendimiento. Fuente de alimentación principal del sistema.	
Cables puente (jumpers)	Cables para realizar las conexiones eléctricas entre módulos y el Arduino. Incluye tipos M-M, M-H y H-H.	
Cartón rígido	Material auxiliar para crear la estructura del carro y su base	

## SOFTWARE NECESARIO

Para llevar a cabo la programación, configuración y pruebas del Robot multifuncional con Arduino, es fundamental disponer del siguiente conjunto de herramientas y plataformas de software. Estas permiten compilar el código, cargar el programa en la placa Arduino, establecer comunicación con el módulo Bluetooth y validar los distintos modos de operación del robot.

COMPONENTE	DESCRIPCION	VERSION RECOMENDADA
Arduino IDE	Entorno de desarrollo integrado	IDE 2.0 o superior

	programación en C/C++ y carga del código en la placa Arduino. Incluye herramientas de depuración básica y monitor serial.	
Driver USB Arduino	Controlador necesario para permitir la comunicación entre la computadora y la placa Arduino UNO mediante puerto USB.	Última versión disponible
Librería Servo.h	Librería estándar de Arduino utilizada para controlar el servomotor que orienta el sensor ultrasónico.	Incluida por defecto
AFMotor.h	Librería propia del <i>Adafruit Motor Shield</i> , utilizada para controlar los motores DC mediante el integrado L293D.	Última versión disponible
Aplicación Android para control Bluetooth	Aplicación móvil utilizada para enviar comandos manuales o comandos de voz al robot: <ul style="list-style-type: none"> <li>• Bluetooth RC Controller</li> <li>• SriTu Voice App</li> </ul> Ambientes compatibles con módulos HC-05/HC-06	Última versión en Google Play

## CODIGO UTILIZADO

```
/*obstacle avoiding, Bluetooth control, voice control robot car.
```

```
Home Page
```

```
*/
```

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

```
#include <AFMotor.h>
```

```
#define Echo A5
```

```
#define Trig A4
```

```
#define motor 10
```

```
#define Speed 250
```

```
#define SpeedTurn 150
```

```
#define spoint 103
```

```
char value;
```

```
int distance;
```

```
int Left;
```

```
int Right;
```

```
int L = 0;
```

```
int R = 0;
```

```
int L1 = 0;
```

```
int R1 = 0;
```

```
Servo servo;
```

```
AF_DCMotor M1(1);
```

```
AF_DCMotor M2(2);
```

```
AF_DCMotor M3(3);
```

```
AF_DCMotor M4(4);
```

```
void setup() {
```

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

```
  pinMode(Trig, OUTPUT);
```

```
  pinMode(Echo, INPUT);
```

```
  servo.attach(motor);
```

```
  M1.setSpeed(Speed);
```

```
  M2.setSpeed(Speed);
```

```
  M3.setSpeed(Speed);
```

```
  M4.setSpeed(Speed);
```

```
}
```

```
void loop() {
```

```
//Obstacle();
Bluetoothcontrol();
//voicecontrol();
}

void Bluetoothcontrol() {
    if (Serial.available() > 0) {
        value = Serial.read();
        Serial.println(value);
    }
    if (value == 'U') {
        forward();
    } else if (value == 'D') {
        backward();
    } else if (value == 'L') {
        left();
    } else if (value == 'R') {
        right();
    } else if (value == 'S') {
        Stop();
    }
}

void Obstacle() {
    distance = ultrasonic();
    Serial.print("Distancia detectada: ");
    Serial.print(distance);
    Serial.println(" cm");

    if (distance <= 15 && distance >= 2) {
        Serial.println("¡Obstáculo detectado! Deteniendo y retrocediendo...");
    }
}
```

```
Stop();
backward();
delay(150);
Stop();

Serial.println("Escaneando izquierda...");
L = leftsee();
Serial.print("Distancia izquierda: ");
Serial.print(L);
Serial.println(" cm");
servo.write(spoint);
delay(800);

Serial.println("Escaneando derecha...");
R = rightsee();
Serial.print("Distancia derecha: ");
Serial.print(R);
Serial.println(" cm");
servo.write(spoint);

if (L < R) {
    Serial.println("Más espacio a la derecha. Girando a la derecha...");
    right();
    delay(600);
    Stop();
    delay(200);

} else if (L > R) {
    Serial.println("Más espacio a la izquierda. Girando a la izquierda...");
    left();
    delay(600);
```

```
Stop();
delay(200);
} else {
Serial.println("Espacio similar en ambos lados. Manteniéndose detenido.");
Stop();
}
} else {
Serial.println("Camino libre. Avanzando...");
forward();
}
}

void voicecontrol() {
if (Serial.available() > 0) {
value = Serial.read();
Serial.println(value);

if (value == '^') {
forward();
} else if (value == '_') {
backward();
} else if (value == '<') {
L = leftsee();
servo.write(spoint);
if (L >= 10 ) {
left();
delay(500);
Stop();
} else if (L < 10) {
Stop();
}
}
```

```
 } else if (value == '>') {  
 R = rightsee();  
 servo.write(spoint);  
 if (R >= 10 ) {  
 right();  
 delay(500);  
 Stop();  
 } else if (R < 10) {  
 Stop();  
 }  
 } else if (value == '*') {  
 Stop();  
 }  
 }  
  
// Ultrasonic sensor distance reading function
```

```
int ultrasonic() {  
 digitalWrite(Trig, LOW);  
 delayMicroseconds(4);  
 digitalWrite(Trig, HIGH);  
 delayMicroseconds(10);  
 digitalWrite(Trig, LOW);  
 long t = pulseIn(Echo, HIGH);  
 long cm = t / 29 / 2; //time convert distance  
 return cm;  
 }
```

```
void forward() {  
 M1.run(FORWARD);  
 M2.run(FORWARD);
```

```
M3.run(FORWARD);  
M4.run(FORWARD);  
}  
  
void backward() {
```

```
M1.run(BACKWARD);  
M2.run(BACKWARD);  
M3.run(BACKWARD);  
M4.run(BACKWARD);  
}
```

```
void right() {  
M1.run(BACKWARD);  
M2.run(BACKWARD);  
M3.run(FORWARD);  
M4.run(FORWARD);  
}
```

```
void left() {  
M1.run(FORWARD);  
M2.run(FORWARD);  
M3.run(BACKWARD);  
M4.run(BACKWARD);  
}
```

```
void Stop() {  
M1.run(RELEASE);  
M2.run(RELEASE);  
M3.run(RELEASE);  
M4.run(RELEASE);  
}
```

```
int rightsee() {  
    servo.write(20);  
    delay(800);  
    Left = ultrasonic();  
    return Left;  
}  
  
int leftsee() {  
    servo.write(180);  
    delay(800);  
    Right = ultrasonic();  
    return Right;  
}
```

## DIAGRAMA DE CONEXIONES

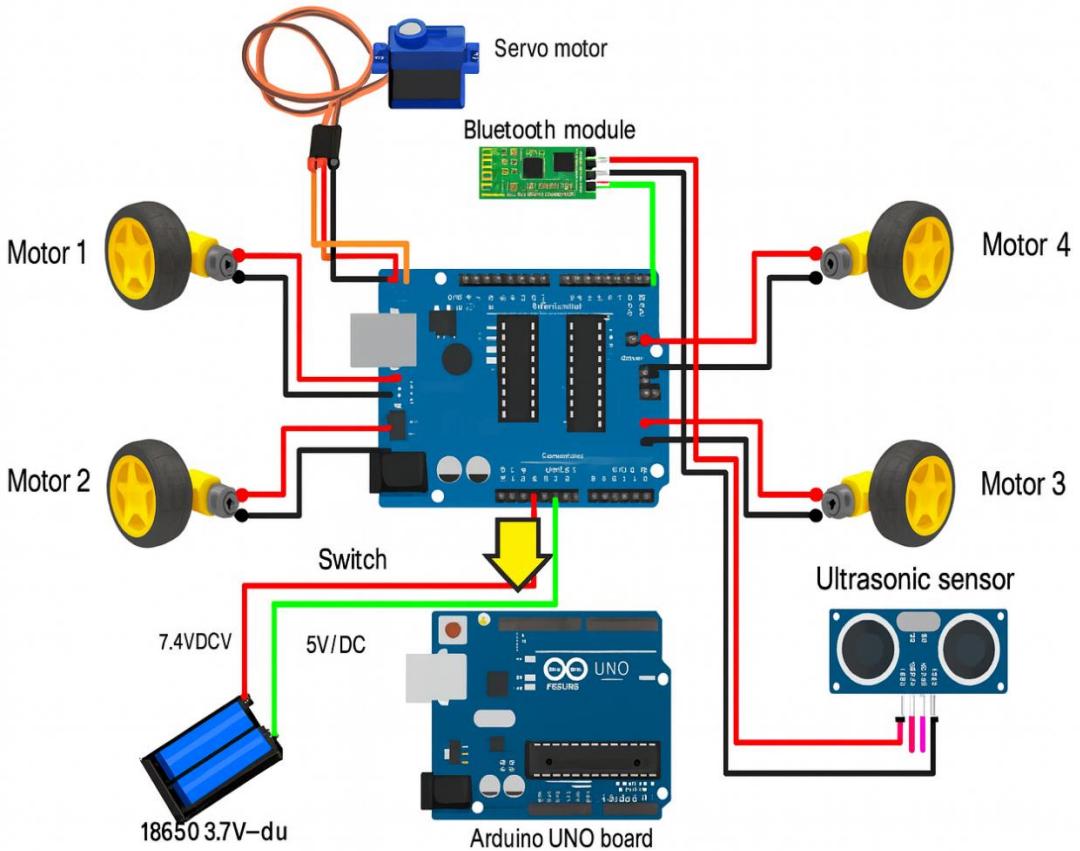
La imagen presentada corresponde al diagrama completo de cableado del robot multifuncional basado en Arduino. En ella se muestra la interconexión entre el controlador principal, los módulos sensores, actuadores, motores y el sistema de alimentación. A continuación, se detalla cada uno de los elementos y sus respectivas conexiones.

### 1. Placa Arduino UNO (Controlador principal)

La placa Arduino UNO se encuentra situada en la parte inferior del diagrama, actuando como la unidad central de procesamiento. Desde ella se distribuyen las señales de control hacia:

- **Servomotor** (PWM – Pin 10)
- **Sensor ultrasónico HC-SR04** (A4 para *Trig* y A5 para *Echo*)
- **Módulo Bluetooth HC-05** (TX/RX)
- **Motor Shield o Controlador L293D** (mediante conexión por pines superiores)

También recibe energía desde el sistema de baterías.



## 2. Controlador de motores (Motor Shield o L293D)

Ubicado sobre el Arduino UNO, este módulo se encarga de suministrar corriente y controlar los cuatro motores DC.

Las salidas se distribuyen de la siguiente forma:

- **Motor 1** Salida superior izquierda
- **Motor 2** Salida inferior izquierda
- **Motor 3** Salida inferior derecha
- **Motor 4** Salida superior derecha

Los cables **rojos** representan la alimentación positiva ( $V_{mot}$ ), y los cables **negros** la línea negativa (GND).

## 3. Motores DC (Motor 1, 2, 3 y 4)

Cada motor incluye dos cables:

- **Rojo:** alimentación positiva
- **Negro:** alimentación negativa

Todos están conectados directamente al Motor Shield. Estos motores permiten el desplazamiento del robot en todas las direcciones.

#### 4. Servomotor (orientación del sensor ultrasónico)

Situado en la parte superior izquierda del esquema:

- Cable **marrón/negro** GND
- Cable **rojo** 5V
- Cable **naranja** Pin digital 10 (control PWM)

El servomotor permite girar el sensor ultrasónico para escanear el entorno.

#### 5. Módulo Bluetooth HC-05

Localizado en la parte superior central del diagrama. Conexiones:

- **VCC** 5V del Arduino
- **GND** GND común
- **TX** Pin RX del Arduino (D0)
- **RX** Pin TX del Arduino (D1)

Se emplea para control remoto mediante aplicación móvil y comandos de voz.

#### 6. Sensor ultrasónico HC-SR04

Situado en la parte inferior derecha del diagrama.

Conexiones:

- **VCC** 5V
- **GND** GND
- **Trig** Pin A4
- **Echo** Pin A5

Este sensor mide la distancia frontal y lateral al combinarse con el servomotor.

#### 7. Sistema de alimentación (baterías 18650)

En la parte inferior izquierda aparece un módulo doble de baterías **18650 de 3.7V**, conectado en serie para proporcionar aproximadamente **7.4V DC**.

Las conexiones son:

- **Rojo (+)** Interruptor principal
- **Interruptor Entrada VIN del Motor Shield**
- **Negro (-)** GND del sistema

El Arduino UNO recibe energía a través del Motor Shield o mediante su entrada VIN/Jack.

## 8. Interruptor principal (Switch)

Colocado entre el portabaterías y el Motor Shield.

- Permite encender o apagar todo el sistema.
- Controla el suministro de 7.4 V a los motores y, por extensión, al Arduino.

## 9. Código de colores en el cableado

COLOR DEL CABLE	FUNCION
Rojo	Alimentación positiva (VCC, VMOT)
Negro	Tierra (GND)
Verde / Amarillo / Naranja	Señales de control y comunicación
Morado / Rosa	Señales de sensor ultrasónico (Triggers y Echo)

## MODO DE OPERACION

El robot multifuncional dispone de tres modos principales de operación, cada uno diseñado para cumplir funciones específicas de movilidad y control remoto. La activación de los modos depende del programa cargado en Arduino, así como de la disponibilidad del módulo Bluetooth y del sensor ultrasónico. A continuación se describen los modos de funcionamiento, su lógica interna y las condiciones operativas necesarias para un desempeño óptimo.

### 9.1 MODO DE EVITACION DE OBSTACULOS

Este modo permite que el robot navegue de manera autónoma utilizando el sensor ultrasónico HC-SR04 montado sobre un servomotor. El objetivo es detectar objetos en la trayectoria y tomar decisiones de movimiento para evitar colisiones.

- El Arduino realiza mediciones constantes de distancia mediante el sensor ultrasónico.
- Cuando se detecta un obstáculo entre 2 cm y 15 cm.
- El robot compara las distancias laterales.
- Una vez libre el camino, continúa avanzando.

## 9.2 MODO DE CONTROL BLUETOOTH

En este modo, el usuario controla directamente el movimiento del robot mediante una aplicación móvil compatible con el módulo HC-05.



El dispositivo envía caracteres individuales al Arduino, el cual interpreta dichos comandos para ejecutar movimientos específicos.

- El módulo HC-05 establece una conexión inalámbrica con un dispositivo Android.
- Cada botón presionado en la aplicación envía un carácter vía Bluetooth.
- Arduino recibe el comando por el puerto serial.
- El robot ejecuta la instrucción correspondiente mediante sus cuatro motores.

## 9.3 MODO DE CONTROL DE VOZ

Este modo permite controlar el robot mediante comandos de voz enviados desde una aplicación móvil, la cual traduce palabras clave a caracteres especiales transmitidos por Bluetooth.

- El usuario pronuncia un comando en la aplicación.
- La aplicación interpreta la voz y envía un carácter específico al HC-05.
- El Arduino recibe el carácter por Serial.
- Se ejecuta la acción correspondiente, verificando primero si existe espacio lateral en los giros.

El desarrollo del presente Manual Técnico permitió documentar de manera detallada la estructura, el funcionamiento y los procedimientos necesarios para la implementación del **Robot Multifuncional con Arduino**, un sistema capaz de operar en modos de **evitación de obstáculos, control remoto por Bluetooth y control por comandos de voz**.