****

**Categoría**

**Futuros Ingenieros**

**Equipo**

**Edinor**

**Integrantes**

**Andrea Moreno**

**Cristian Villalba**

**Rafael Pelletier**

**Año lectivo**

**2024**

**Introducción**

Armar un carrito utilizando Arduino puede ser un proyecto emocionante y educativo. El procedimiento general para construir y programar un carrito incluye varias etapas clave: planificación, ensamblaje de hardware, y programación. A continuación, se presenta una introducción sobre cada uno de estos pasos y cómo se integran los códigos en Arduino.

**Procedimiento de Construcción del Carrito**

1. **Planificación y Diseño**

El primer paso en la creación del carrito es la planificación. Esto incluye definir las funcionalidades del carrito, seleccionar los componentes necesarios y diseñar un esquema de cómo se conectarán estos componentes. Entre los elementos básicos se encuentran el chasis, los motores, las ruedas, una placa Arduino, una batería y sensores opcionales.

El primer comportamiento que usamos para hacer el vehículo o el carrito, fue Chasis que es la estructura física del carrito. Sirve como soporte para todo los componentes y mantiene todo en su lugar de manera segura y estable.

Seguro fueron las Llamadas,el Motor y el Conjunto de Dirección de Eje , son piezas claves para el movimiento del carrito.

Ya tercera sería lo más importante para el funcionamiento que es Arduino Uno que sirve para gestionar las operaciones del carrito. Ahí entraría el sensor, Recibe entrada de los sensores y, Basándose en el código programado, controla los motores y otros actuadores.

Encima del Arduino pusimos el Controlador de Motor L293D, es un puente que permite controlar la dirección y velocidad de los motores DC. También agregamos servo Motor MG99R, es un tipo de servo Motor conocido por su alta resistencia y precisión.Para q se moviera para la derecha o para la izquierda.

Igual con el Motor Micro Servo GG, utilizado para realizar movimientos precisos y controlados. Estos servos son populares en aplicaciones de robótica debido a su tamaño compacto, ligereza y capacidad para proporcionar control de posición exacto. Se utilizan comúnmente para dirigir ruedas, controlar brazos robóticos pequeños o mover componentes a ángulos específicos. Arriba del servo Motor estaría el sensor Ultrasonico, utilizados para medir distancia y detectar obstáculos permitiendo al carrito evitar colisiones.

También agregamos soporte para el carrito, ahí entrarían los espaciadores de Latón.

Y para que funcionará, le agregamos una batería de 9 v y el Switch para apagar y encender

**Recolección de Componentes:**

**Chasis**

El chasis es la estructura física del carrito. Sirve como soporte para todos los componentes y mantiene todo en su lugar de manera segura y estable.

* Importancia:

Proporciona la base sobre la cual se montan los motores, la placa Arduino y otros componentes.

Garantiza que el carrito mantenga su forma y funcionalidad durante el movimiento.

Permite la distribución equilibrada del peso, lo cual es crucial para la estabilidad y el rendimiento del carrito.

**Motor**

El Motor es un dispositivo que proporciona el movimiento al carrito. En la mayoría de los casos, se utilizan motores DC (corriente continua) o servomotores. La función del motor es convertir la energía eléctrica en movimiento mecánico, permitiendo que el carrito avance, retroceda o gire. El motor se conecta a las ruedas del carrito para transmitirles el movimiento.

**Placa Arduino**

La placa Arduino es el cerebro del carrito. Es una microcontroladora que se programa para gestionar las operaciones del carrito. Recibe entradas de los sensores y, basándose en el código programado, controla los motores y otros actuadores. Es la pieza central que coordina todas las funciones del carrito mediante el procesamiento de señales y ejecución de instrucciones.

**Controlador de Motor L293D**

El controlador de motor L293D es un puente H que permite controlar la dirección y velocidad de los motores DC. Debido a que los pines de Arduino no pueden proporcionar suficiente corriente para los motores, el L293D actúa como un intermediario. Recibe señales de baja corriente desde el Arduino y las amplifica para controlar los motores. Este controlador permite cambiar la polaridad de los motores, lo que es necesario para mover el carrito hacia adelante y hacia atrás.

**Batería**

La batería proporciona la energía necesaria para que todos los componentes del carrito funcionen. Debe suministrar el voltaje y la corriente adecuados tanto para los motores como para la placa Arduino y otros componentes electrónicos. Una batería comúnmente usada en estos proyectos es una de 9V o un pack de baterías recargables, que ofrece una buena duración y rendimiento.

**Sensor**

Los sensores permiten que el carrito interactúe con su entorno. En este caso usamos el Sensor Ultrasónico.

* Sensor Ultrasónico: Utilizados para medir distancias y detectar obstáculos, permitiendo al carrito evitar colisiones.

**Parachoques de esponja**

Un parachoques de esponja es un tipo de amortiguador hecho de material esponjoso, diseñado para absorber impactos y proteger superficies de daños. Estos parachoques son suaves y flexibles, lo que los hace ideales para aplicaciones donde es importante minimizar el riesgo de daños a los objetos o superficies en contacto. Se utilizan en una variedad de entornos, desde juguetes y robots hasta equipos electrónicos y muebles, proporcionando una capa de protección contra golpes y caídas.

**Servo Motor MG996R**

El MG996R es un tipo de servo motor conocido por su alta resistencia y precisión. Este servo motor es popular en proyectos de robótica y modelismo debido a su capacidad para manejar cargas pesadas y proporcionar un control preciso de posición. Algunas características clave del MG996R incluyen:

* Par máximo: Aproximadamente 10 kg·cm a 6V.
* Velocidad: 0.17 segundos por 60 grados a 6V.
* Ángulo de operación: Típicamente 0 a 180 grados.
* Voltaje de operación: Generalmente entre 4.8V y 7.2V.
* **Aplicaciones**: Control de dirección, actuadores en robots, sistemas de control de movimiento en modelos a escala, entre otros.

**Conjunto de Dirección de Eje**

Un conjunto de dirección de eje es un sistema mecánico utilizado para controlar la dirección de las ruedas en vehículos y otros dispositivos móviles. Este conjunto generalmente incluye varios componentes clave, como el eje de dirección, los brazos de dirección, las juntas universales y los rodamientos. El objetivo principal del conjunto de dirección de eje es transmitir los movimientos del volante o del sistema de control a las ruedas, permitiendo así al operador dirigir el vehículo. Este tipo de sistema se utiliza para maniobrar las ruedas y controlar la dirección del movimiento del robot de manera precisa y eficiente.

**Cable Macho y Hembra**

* **Cable Macho:** En robótica, un cable macho tiene conectores que se insertan en receptáculos o conectores hembra. Estos cables suelen tener pines metálicos sobresalientes en uno o ambos extremos. Se utilizan para hacer conexiones rápidas y fáciles entre diferentes componentes electrónicos, como sensores, motores, y microcontroladores.
* **Cable Hembra:** Un cable hembra tiene conectores diseñados para recibir los pines de un conector macho. Los extremos tienen receptáculos o agujeros en los que se insertan los pines del cable macho. Son esenciales para conectar de manera segura y ordenada varios componentes en un sistema robótico.

**Motor Micro Servo GG**

Motor Micro Servo GG: En el contexto de un carrito robótico, un motor micro servo GG es un pequeño motor servo utilizado para realizar movimientos precisos y controlados. Estos servos son populares en aplicaciones de robótica debido a su tamaño compacto, ligereza y capacidad para proporcionar control de posición exacto. Se utilizan comúnmente para dirigir ruedas, controlar brazos robóticos pequeños o mover componentes a ángulos específicos.

**Llantas**

Las llantas son las ruedas de un carrito robótico. Son esenciales para la movilidad del vehículo y están diseñadas para proporcionar tracción y estabilidad. En robótica, las llantas pueden variar en tamaño y material, dependiendo del terreno sobre el cual el carrito va a operar. Las llantas pueden estar hechas de goma, plástico u otros materiales que aseguren un agarre adecuado.

**Soportes para Robótica Carrito**

Los soportes en un carrito robótico son estructuras o componentes diseñados para montar y asegurar diferentes partes del vehículo, como motores, sensores, microcontroladores y otros componentes electrónicos. Estos soportes ayudan a mantener el diseño compacto y organizado del carrito, asegurando que todos los componentes estén bien fijados y alineados correctamente para su funcionamiento. Los soportes pueden estar hechos de materiales como plástico, metal o acrílico y son cruciales para la integridad estructural del robot.

**Switch**

En un vehículo robótico, los switches se usan para encender y apagar el carrito.

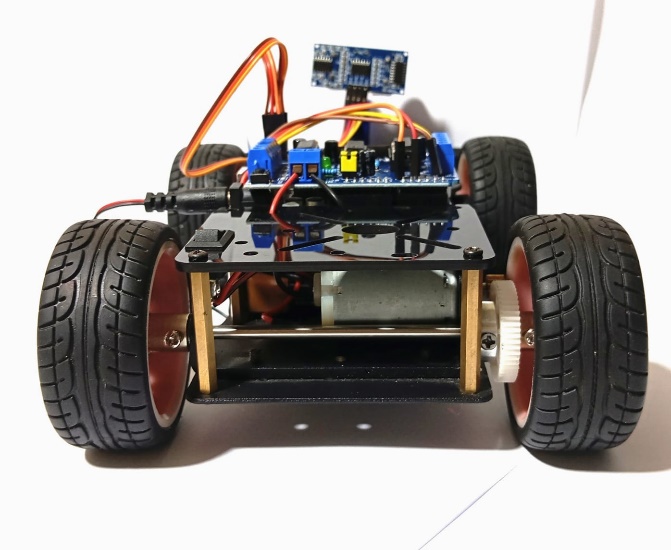
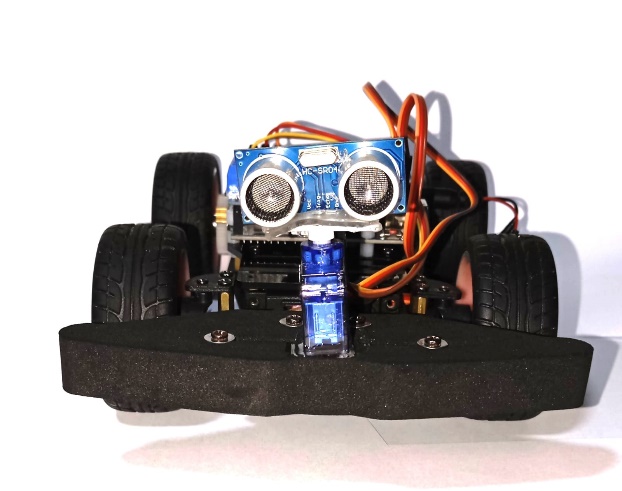
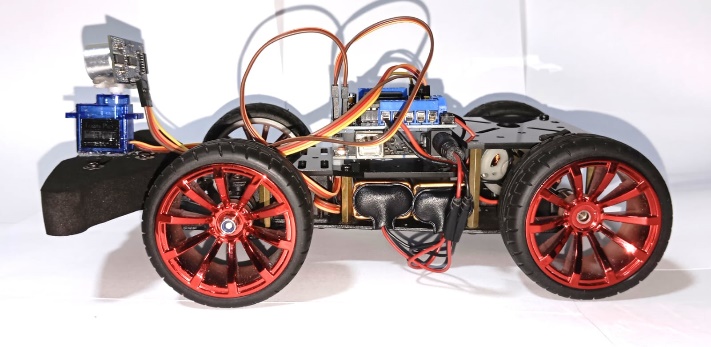
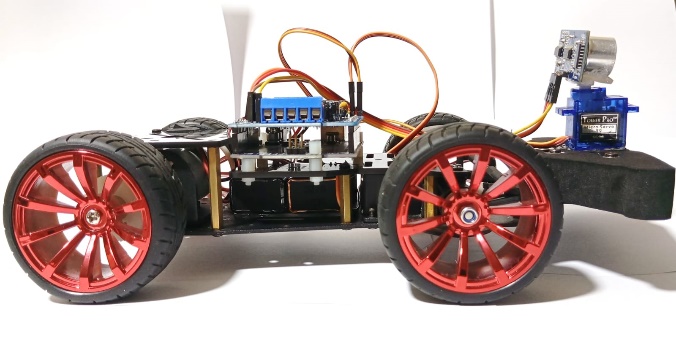
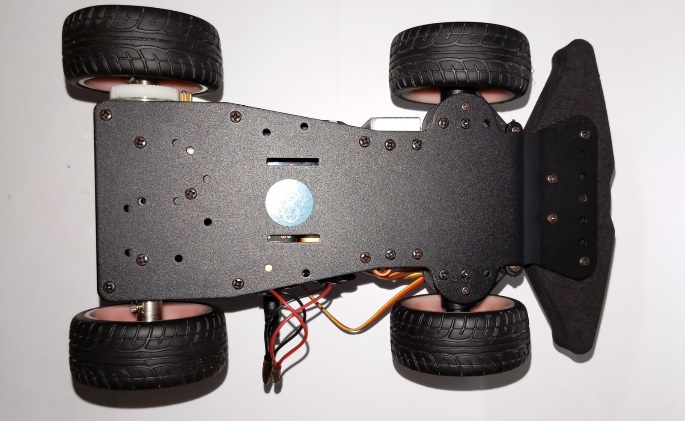
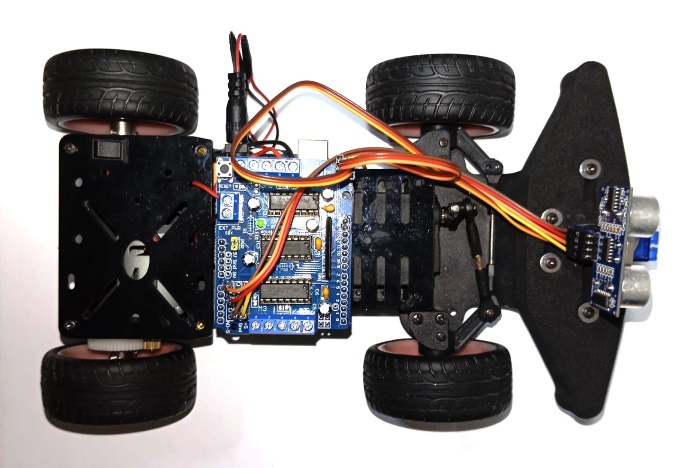
**Espaciadores de Latón**

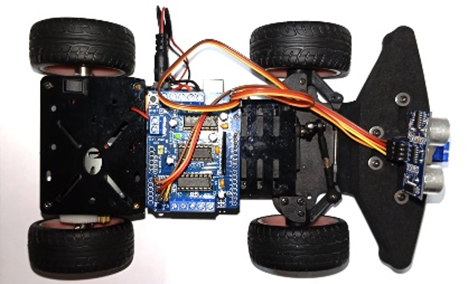
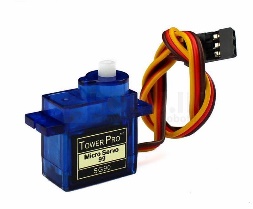
Los espaciadores de latón son pequeños componentes cilíndricos utilizados en robótica para separar, elevar y asegurar partes del robot. Estos espaciadores proporcionan un montaje rígido y estable entre las placas del chasis, componentes electrónicos, motores y otras partes del robot. El latón es un material preferido debido a su resistencia a la corrosión y durabilidad.

**Batería de 9V**

Una batería de 9 voltios es una fuente de alimentación portátil que proporciona 9 voltios de corriente continua (DC). Es comúnmente utilizada para alimentar pequeños dispositivos electrónicos y componentes de robots.

En un carrito robótico, una batería de 9V puede ser utilizada para alimentar los motores, el controlador de motor, el microcontrolador (como un Arduino), y otros componentes electrónicos del vehículo.

**Fotos del Carrito**

**Componentes** 

**Cable Macho y Hembra (14u)**

**Bateria 9 V (2u)**

**Control de Motor L293D (1u)**

**Espaciadores de Laton (6u)**

**Y soportes (1u)**

**Chasis (1u)**

**Conjunto de Dirección de Eje (1u)**

**Motor (1u)**

**Switch (1u)**

**Motor Micro servo GG (1u)**

**Parachoques de Esponja (1u)**

**Placa Arduino(1u)**

**Sensor (1u)**

**Servo Motor mg996r (1u)**

**Llantas (4u)**

**Carrito**

**4.** **Programación en Arduino**

**Parachoques de Esponja (1u)**

**Switch(1u)**

**Motor (1u)**

**Sensor (1u)**

**Servo Motor mg996r (1u)**

**Bateria 9 V (2u)**

**Cable Macho y Hembra (14u)**

**Control de Motor L293D (1u)**

**Motor Micro servo GG (1u)**

**Placa Arduino(1u)**

**Soportes**

**(6u)**

**Chasis**

**(1u)**

**Conjunto de Dirección de Eje**

**(1u)**

**Llantas**

**(4u)**

#include <Servo.h> // Load the Servo library

#include <Ultrasonic.h> // Load the Ultrasonic library

// Define pins for servo motors and ultrasonic sensor

const int servo1Pin = 22; // Pin for servo1 (mini servo GG)

const int servo2Pin = 24; // Pin for servo2 (MG996R)

const int echoPin = 8; // Pin for ultrasonic sensor echo

const int triggerPin = 9; // Pin for ultrasonic sensor trigger

const int m1Pin = 2; // Pin for motor impulse control (M1)

// Create servo objects

Servo servo1;

Servo servo2;

// Variables for ultrasonic sensor

long distance;

long time;

void setup() {

Serial.begin(9600);

pinMode(echoPin, INPUT);

pinMode(triggerPin, OUTPUT);

servo1.attach(servo1Pin, 2, 178); // Attach servo1 to pin 22 with limits 2-178

servo2.attach(servo2Pin); // Attach servo2 to pin 24

pinMode(m1Pin, OUTPUT); // Set M1 pin as output

}

void loop() {

// Read distance from ultrasonic sensor

digitalWrite(triggerPin, LOW);

delayMicroseconds(5);

digitalWrite(triggerPin, HIGH);

delayMicroseconds(10);

time = pulseIn(echoPin, HIGH);

distance = int(0.01715 \* time);

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Control servo1 movement

if (distance < 100) {

servo1.write(90); // Stop servo1

delay(100);

// Reverse direction and move backwards

servo2.write(180);

digitalWrite(m1Pin, HIGH);

delay(500);

digitalWrite(m1Pin, LOW);

delay(500);

servo2.write(0);

} else {

// Move servo1 from 2 to 178 degrees

for (int i = 2; i <= 178; i++) {

servo1.write(i);

delay(10);

}

for (int i = 178; i >= 2; i--) {

servo1.write(i);

delay(10);

}

}

}

1. **Diagrama de Flujo del Carrito (corto)**

* Primera prueba. Inicio

|

V

Avanzar

|

V

Detectar Pared

|

|----> ¿Pared Detectada?

| |

No Sí

| |

V V

Continuar Detener

Avanzando |

V

Retroceder

|

V

Girar a la

derecha

|

V

Avanzar Recto

|

V

Fin

Segunda Prueba.

Inicio

|

V

Avanzar -----> Sensor Activo?

| |

V V

No Sí

| |

V V

Continuar Avanzando Esquivar Obstáculo

| |

V V

-----> Volver a Avanzar

|

V

Fin