

Seminario de Robótica

Corporación Universitaria Remington.
Facultad de Ingeniería.
Ingeniería de Sistemas.

Jhoan Sebastián Palacio Martínez.
Julián Piedrahita Trujillo.
Tutor: Jonathan Stick Campos Núñez.
Trabajo para Seminario como opción de grado.
2024.

Dedicatoria

Con gran afecto este proyecto va dedicado a cada una de las personas que hacen parte del vinculo familiar (padres, esposa, novia, hijos, abuelos, tíos, primos) por haber hecho parte de este camino tan largo y lleno de esfuerzos.

Agradecimientos

Este agradecimiento va primero para Dios que nos da la vida, salud, sabiduría y entendimiento día a día, en segundo lugar, a nuestras familias que han estado y acompañando en cada uno de los momentos durante estos procesos, en tercer lugar, a cada uno de los tutores por brindarnos sus conocimientos, ayudarnos a ser mejores seres y formar profesionales competentes.

Tabla de contenido

Resumen	5
Palabras claves	5
Pregunta orientadora de la búsqueda.....	6
Objetivo general	6
Objetivos específicos	6
Metodología de la búsqueda de la información	7
Sustentación teórica de la pregunta	8
1. Desarrollo de la robótica.....	8
1.1 ¿Cómo se define la robótica?.....	8
1.2 ¿De dónde proviene la palabra robótica?.....	8
1.3 ¿En que momento se fabricó el primer robot?	9
1.4 ¿Cuáles son las aplicaciones más importantes de la robótica?	9
1.5 ¿Qué ventajas puede ofrecer el uso de la robótica?	9
2. Construcción de robots	10
2.1 Controlador	10
2.2 Esqueleto del robot	10
2.3 Actuadores	10
2.4 Manipulador	11
2.5 Sistema de control.....	11
3. Arduino.....	11
4. Controlador	12
4.1 Conexión y funcionamiento	13
5. Protoboard.....	14
6. Servomotor.....	14
7. Informe de la actividad	15
Conclusiones.....	16
Referencias	17
Anexos	18

Índice de anexos

Anexo 1: Diagrama de conexiones	18
Anexo 2: Lista de materiales	18
Anexo 3: Códigos de funcionamiento.....	19
Anexo 4: Evidencia fotográfica de la practica	25

Resumen

A lo largo de las décadas se ha evidenciado la importancia que contiene los robots en la vida del ser humano tanto en la parte laboral como en su día a día, estos pueden desempeñar diversas actividades desde las mas sencillas hasta las mas complejas, aquellos robots que cuentan con un sistema de inteligencia artificial, esto se puede llevar a cabo de acuerdo a los diversos sensores de movimiento, servomotores, algoritmos, lo cual permite que los robots actúen en conjunto de acuerdo a lo que perciben en su entorno. El énfasis que se realiza especificando la importancia que posee la robótica su historia, brinda una perspectiva mas explicita sobre esta ciencia, como aplicarla y hacer mejoras a la calidad de vida.

En este informe como opción de grado se documenta y da a conocer el procedimiento para la creación de un robot básico (auto) y que este siga órdenes para su desplazamiento utilizando un servomotor y un sensor de movimiento los cuales van conectados al Arduino, también se conectan dos motores al controlador L298n y este se conectó al Arduino al cual se le realizo la lógica de programación

Palabras claves

Algoritmo, Arduino, controlador, robot, robótica, sensor de movimiento, servomotor.

Pregunta orientadora de la búsqueda

Incógnitas que surgen en el proceso de la creación y construcción de robots

¿Qué se necesita para crear un robot?

¿Qué sensores utilizar para su construcción?

¿Cómo se puede construir un robot con Arduino?

¿Qué utilidad puede tener un robot detector de obstáculos?

Objetivo general

- Identificar como actúan los robots dependiendo sus sensores, servomotores, controladores de acuerdo a la programación que se les proporciona y determinar las utilidades que pueden tener y su posible progreso y mejora

Objetivos específicos

- Conocer la conducta del robot.
- Detectar que ajustes son necesarios para saber que mejoras aplicar en su funcionamiento.
- Verificar la codificación pertinente para el robot.
- Verificar el correcto funcionamiento del robot.
- Documentar el procedimiento de la creación del robot en este caso (un auto).

Metodología de la búsqueda de la información

La metodología empleada para la investigación del presente informe se utilizó Google, Google académico, libros en pdf donde se consultó sobre historia de la robótica, Arduino, servomotores, sensores de movimiento,

Sustentación teórica de la pregunta

1. Desarrollo de la robótica

1.1 ¿Cómo se define la robótica?

La robótica es la ciencia que combina varias disciplinas o ramas de la tecnología para diseñar máquinas programadas para realizar tareas o para simular el comportamiento humano o animal. A simple vista, un robot es una computadora capaz de moverse sin entrar en detalles.

La robótica es la creación de artefactos que satisfacen el deseo humano de crear seres a su semejanza para satisfacer algunas de sus necesidades más complejas y liberarse de trabajos tediosos o peligrosos. El término "robótica" se refiere a la creación, construcción y manejo de robots. Es una industria que ofrece una amplia gama de productos.. (Robótica - Ferrovial, 2024)

1.2 ¿De dónde proviene la palabra robótica?

El término "robot" proviene de la palabra checa "robota". Karel Čapek la utilizó por primera vez en su obra dramática R. U. R. (Robots Universales Rossum), publicada en 1920, para referirse a unas máquinas pensantes que se rebelan y acaban por matar a su creador. La palabra checa "robota" fue traducida al inglés como "robot".

Isaac Asimov inventó el término robótica y lo definió como la disciplina que estudia a los robots. Asimov también inventó las tres leyes de la robótica. En la ciencia ficción, los humanos han imaginado a los robots explorando nuevos mundos, adquiriendo poder o simplemente aliviándolos de las tareas domésticas. En cualquier caso, se deben cumplir las tres reglas de la robótica establecidas por Isaac Asimov: Un robot debe ser inofensivo para los humanos, obedecer sus órdenes y proteger su integridad sin faltar a lo previo. (Robótica - Ferrovial, 2024)

1.3 ¿En qué momento se fabricó el primer robot?

Algunas fuentes indican que los primeros robots surgieron alrededor del siglo III a.C., pero las primeras máquinas automatizadas eran más mecánicas que robóticas.

La Westinghouse creó el primer robot humanoide del mundo llamado Elektro. Era una máquina que podía caminar y tenía una grabación de 700 palabras para simular

una conversación. Entre 1939 y 1940, participó en la Exposición Universal.. (Robótica - Ferroviario, 2024)

1.4; Cuáles son las aplicaciones más importantes de la robótica?

Las aplicaciones de la robótica en los sectores profesionales incluyen:

- Traslado de materiales.
- Configuración.

Corte mediante métodos mecánicos, ajustes, rebabas y pulido.

- Pintura.
- Procesar materiales como plásticos.
- Trabajos peligrosos como soldar, trabajar con materiales pesados y inhalar sustancias peligrosas
- Atención médica.
- Reutilización.

Actualmente, los robots son comunes en la industria y son necesarios para la mayoría de los procedimientos de fabricación. Por ejemplo, Ferroviario inició el proyecto ZRR para residuos municipales en España el 5 de octubre de 2018. Durante 21 meses, el proyecto investigará el uso de IA y robótica en la clasificación y selección de desechos urbanos municipales. Para este proyecto, Ferrocarril colabora con Zenrobotics, una empresa emergente que se especializa en estas tecnologías innovadoras... (Robótica - Ferroviario, 2024)

1.5; Qué ventajas puede ofrecer el uso de la robótica?

- Mayor precisión.
- Ayudar emocionalmente a las personas.
- Realizar tareas peligrosas.
- Realidad ampliada.

- Mayor velocidad.
- Reducción de costos.
- Ir a donde el humano no puede.
- Hacer tareas que para el ser humano serían mortales.

2. Construcción de robots.

En el desarrollo de la construcción de robots interceden diversos factores y componentes, hablando de robots industriales estos deberán ser construidos de materiales resistentes; en este proyecto se fabricara un robot básico tanto en sus componentes como en su programación; las partes de las cuales está conformado un robot son:

2.1 Controlador:

Este utiliza una pequeña computadora para controlar todas las funciones, movimientos, cálculos y procesamientos de datos. Este pequeño computadora tiene una unidad central, memoria, dispositivos de alimentación e interfaces que le permiten usar comandos de fuera. En nuestro caso, utilizaremos una placa Arduino uno.

Los parámetros que controlan tienen la capacidad de cambiar la función de los controladores. Aunque pueden ser fundamentalmente de posición, cinemática, dinámica o adaptativa.

2.2 Esqueleto del robot.

El esqueleto sostiene las partes restantes de un robot. Este debe ser específico para el propósito del robot, como ser grande, fuerte, pequeño o rápido. En nuestro caso, será un chasis de carrito que soportará al robot y le permitirá llevar llantas para moverse.

Otro factor importante es el material del que estará hecho el esqueleto. Puede ser duro, pesado, flexible o todo al mismo tiempo, dependiendo del objetivo del robot. Hay robots con la misma estructura, ya sea papel mache, madera, hierro o plástico.

2.3 Actuadores:

Los actuadores están en el corazón del robo industrial, ya que son los encargados de impulsar el movimiento de la maquinaria.

Se trata de elementos electrónicos que pueden realizar acciones físicas dentro del entorno del robot.

Hay dos grupos principales de actuadores: motores.

Se trata de elementos esenciales para los robots, ya que permiten que el robot se desplace de un lugar a otro, mueva el brazo del robot y opere la pinza.

Otros actuadores incluyen pantallas LCD, visualizadores, parlantes y sincronizadores de audio.

Sensores Para ser completo, un robot debe reaccionar a su entorno y ser completamente autónomo.

Los sensores brindan esta capacidad según el propósito del robot y deben ubicarse estratégicamente según su funcionalidad.

Los sensores más conocidos son los de luz, sonido, gravedad, temperatura, humedad, presión, velocidad, magnetismo y posición.

También hay sensores de proximidad, sensores de distancia, cámaras de vídeo, etc.

2.4 Manipulador.

El manipulador es la parte mecánica más grande del robot. La pieza en sí es una pieza estructural formada por elementos sólidos o uniones que están conectadas entre sí mediante uniones y permiten moverse. También conocido como cuerpo, brazos, manos y extremidades.

2.5 Sistema de control.

Un sistema de control es una de las partes de un robot que incluye software y hardware que dirige y controla el movimiento del robot para realizar una tarea determinada, en nuestro sistema de control se realizará un algoritmo para que el robot interactúe con algunos sensores, realice un desplazamiento y efectúe algunas actividades.

Aquí se explica cómo preprogramar o hacer que su robot sea autónomo.

Los robots preprogramados realizan tareas repetitivas repetidamente.

Los robots autónomos pueden leer y reaccionar a los cambios en su entorno a través de sensores.

En resumen, para que un robot haga lo que usted quiere que haga, es muy importante alinear el esqueleto, los sensores y los actuadores con este objetivo.

En particular, el cerebro del robot debe programarse para realizar funciones clave. Cada pieza tiene un propósito y es una adición esencial a esta máquina. (*Esneca, 2023*).

3. Arduino

Arduino es una plataforma de creación de prototipos electrónicos de código abierto basada en hardware y software flexible y fácil de usar.

Destinado a artistas, diseñadores y cualquier persona interesada en crear objetos y entornos interactivos como pasatiempo. Arduino detecta su entorno al recibir información de una variedad de sensores y puede influir en su entorno controlando luces, motores y otros dispositivos. El microcontrolador de la placa se programa utilizando el lenguaje de programación Arduino (basado en cables) y el entorno de desarrollo Arduino (basado en procesamiento). Los proyectos Arduino pueden ser autónomos o comunicarse con el software que se ejecuta en su computadora (Flash, Processing, MaxMSP, etc).

Los paneles se pueden montar a mano o pedir premontados. El software se puede descargar gratis. Los diseños de referencia de hardware (archivos CAD) están disponibles bajo una licencia de código abierto, por lo que puede personalizarlos libremente según sus necesidades. Arduino ganó una mención de honor en la categoría Comunidad Digital en el Gran Premio Ars Electronica en 2006.

HARDWARE, Existen múltiples versiones de la placa Arduino. La mayoría usan el ATmega168 de Atmel, mientras que las placas más antiguas usan el ATmega8. Nota: Los diseños de referencia para Arduino se distribuyen bajo licencia Creative Commons Attribution-ShareAlike 2.5 (Herrador, 2009)

4. Controlador

El módulo de control de motor puente H L298N permite controlar la velocidad y dirección de dos motores DC o motores paso a paso de una forma muy sencilla gracias a los dos puentes H integrados.

Como ya hemos mencionado, básicamente un puente H o puente H es un componente formado por cuatro transistores que permite invertir el sentido de la corriente, y de esta forma se puede invertir el sentido de giro del motor. Este módulo funciona en un rango de voltaje de 3 V a 35 V con una resistencia máxima de 2 A. Respecto a la alimentación, hay que tener en cuenta que la electrónica del módulo consume unos 3V, por lo que el motor estará a 3V. Es inferior al voltaje suministrado al módulo. Además, el L298N tiene un regulador de voltaje que te permite obtener 5V del módulo, perfecto para alimentar tu Arduino. Por supuesto, este regulador sólo funcionará si suministras al módulo una tensión máxima de 12V. Módulo muy utilizado en proyectos de robótica por su facilidad de uso y bajo coste. (prometec, 2023)

4.1 conexión y funcionamiento

La entrada de voltaje V_{in} admite voltajes entre 3V y 35V, y justo a la derecha de ella en la imagen tenemos el pin que debemos conectar a GND.

La tercera conexión de ese grupo lógico V puede funcionar de dos maneras:

- Si el jumper del regulador está cerrado activaremos el regulador de voltaje L298N, y en lógica V tendremos una salida de 5V, que podremos usar para lo que queramos, por ejemplo para alimentar una placa Arduino.
- Si quitamos el jumper desactivaremos el regulador, necesitaremos alimentar la parte lógica del módulo, por lo que tendremos que introducir una tensión de 5V por la conexión V lógica para que el módulo funcione.
- ¡Ten cuidado! Si introducimos corriente por V lógica con el jumper de regulación puesto podríamos dañar el módulo.
- Además, el regulador sólo funciona con voltajes de hasta 12V en V_{in} , por encima de este valor tendremos que quitar el jumper y alimentar la parte lógica del módulo desde otra fuente.

El resto de las conexiones se utilizan de una forma u otra dependiendo de si vamos a manejar dos motores DC o un paso a paso. motor. En esta sección trabajaremos en el control de motores de CC.

La salida de los motores A y B proporciona la potencia necesaria para mover los motores. Recuerda la polaridad al conectar para que podamos seguir adelante y hacerlo correctamente. Si no, simplemente cambie los enlaces.

Los pines IN1 e IN2 controlan la dirección de rotación del motor A y los pines IN3 e IN4 controlan la dirección del motor B. Funcionan si IN1 está en HIGH y IN2 en LOW, el motor A girará en un sentido, y si IN1 está en LOW y IN2 en HIGH, lo hará en el otro sentido. Lo mismo con los pines IN3 e IN4 y el motor B.

Para controlar la velocidad de rotación del motor, retire los puentes y use los pines ENA y ENB. Lo conectamos a las dos salidas PWM del Arduino para enviar un valor entre 0 y 255 que controla la velocidad de rotación. Si tenemos saltos, los motores siempre girarán a la misma velocidad.

El diagrama de configuración que utilizaremos será el siguiente, aunque podrás utilizar los pines que quieras siempre y cuando respetemos el PWM de los pines que conectemos a la ENA y al ENB. (prometec, 2023)

5. Protoboard

Se llamará soporte protoboard a aquellos elementos que cumplirán la función en la vivienda de ser inamovibles; así mismo contendrán todas las instalaciones y a la vez serán la base de la flexibilidad de la vivienda. Con el sistema de encajes para los muros, las instalaciones de estos se realizarán de manera fácil, rápida y sin altos

costos, logrando personalizar el recinto según el usuario que lo habite. (ESTRADA, 2016)

6. Servomotor

El módulo de control de motor puente H L298N permite controlar la velocidad y dirección de dos motores DC o motores paso a paso de una forma muy sencilla gracias a los dos puentes H integrados. Como ya hemos mencionado, básicamente un puente H o puente H es un componente formado por cuatro transistores que permite invertir el sentido de la corriente, y de esta forma se puede invertir el sentido de giro del motor.. (Aguayo, 2009)

Este módulo funciona en un rango de voltaje de 3 V a 35 V con una resistencia máxima de 2 A. Respecto a la alimentación, hay que tener en cuenta que la electrónica del módulo consume unos 3V, por lo que el motor estará a 3V. Es inferior al voltaje suministrado al módulo. Además, el L298N tiene un regulador de voltaje que te permite obtener 5V del módulo, perfecto para alimentar tu Arduino. Por supuesto, este regulador sólo funcionará si suministras al módulo una tensión máxima de 12V. Módulo muy utilizado en proyectos de robótica por su facilidad de uso y bajo coste. (Aguayo, 2009)

La tarjeta Arduino Duemilanove tiene seis salidas de PWM, Modulación de Ancho de Pulso y Modulación de Ancho de Pulso en inglés, que usaremos para controlar los motores.. (Aguayo, 2009)

7. Informe de la actividad

Este subproyecto forma parte de la construcción de nuestro robot autónomo, usaremos el driver L298n para gobernar 2 motores de CC que realizarán la parte mecánica de nuestro robot móvil. Nuestro robot irá recto y cuando encuentre un obstáculo a menos de 5 cm girará en sentido horario hasta que esquive el obstáculo y siga su camino.

Se elabora código para lectura de los sensores y motores para que ejecuten las actividades establecidas, se verifica el comportamiento que va teniendo el robot a

medida que se codifica y de esta manera establecer las correcciones que necesite hasta llegar a su optimo funcionamiento.

El carro (robot), se forma con un chasis al cual se le ensamblaron 2 motores, 2 llantas, el controlador L298n, una protoboard, 1 arduino, el cableado y las pilas.

Conclusiones

- Se comprueba que por medio de diferentes actores (motores, sensores, códigos) se puede hacer la construcción de un robot funcional.
- Se evidencia que por medio de códigos se puede dar ordenes a un robot para que esquive obstáculos.
- Con diversas pruebas de codificación se evidencia que el comportamiento del robot puede que no sea totalmente eficiente y de dicha forma evidenciar que correcciones aplicar para que ya sea optimo su funcionamiento.
- A pesar de la complejidad que pueda la codificación y calibración de su movilidad se puede llegar a que sean muy cercanas en términos de tiempo y movilidad.
- Queda la certeza que el mundo actual esta enfatizado en el uso de la tecnología y la creación de robots es un gran avance para alivianar las tareas del día, con lo cual al adquirir cada día nuevos conocimientos de cómo construir y ejecutar un equipo funcional nos ayudara a crecer y avanzar como sociedad.

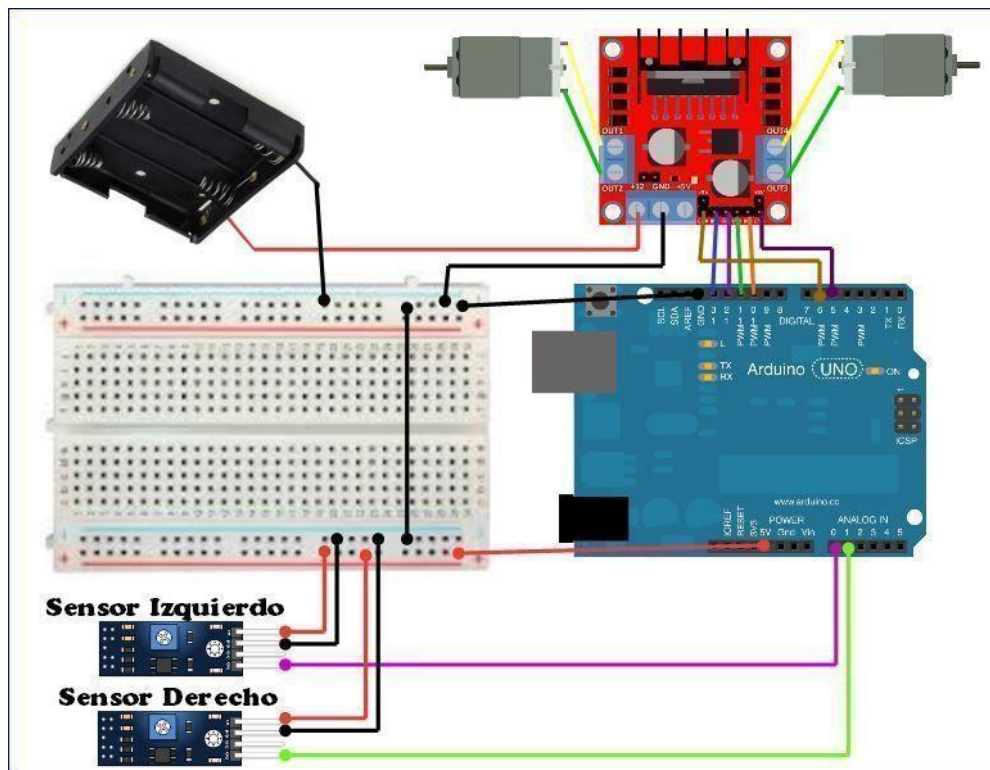
Referencias

Referencias

- Aguayo, P. (Diciembre de 2009). *Servo motor con Arduino*. Obtenido de arduino.cl:
<https://arduino.cl/servo-motor-con-arduino/>
- ESTRADA, S. B. (2016). *SISTEMA PROTOBOARD Y MURO*. Obtenido de
<https://repository.upb.edu.co/>
<https://repository.upb.edu.co/bitstream/handle/20.500.11912/5005/Sistema%20protoboard%20y%20muro-mueble.pdf?sequence=1&isAllowed=y>
- Herrador, R. E. (13 de Noviembre de 2009). *Guía de usuario arduino*. Obtenido de
www.uco.es: https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf
- prometec. (2023). *prometec*. Obtenido de www.prometec.com:
<https://www.prometec.net/1298n/>
- Robótica - Ferrovial. (2024). *Ferrovial*. Obtenido de www.Ferrovial.com.

Anexos

Anexo 1: Diagrama de conexiones



Anexo 2: lista de materiales

- 1 Kit chasis de carro con tres ruedas
- 1 Arduino UNO Compatible con Cable USB
- 1 cable micro USB
- 1 baterías de respaldo 9V 2600mAh
- 1 Puente H
- 1 Protoboard
- Separadores de Nylon M3x8
- 6 tornillos M3x6
- 6 tuercas
- 2 motores de corriente continua.
- 2 ruedas fijas.
- 1 rueda móvil.
- 1 sensores Ultrasónico HC-SR04
- 12 cables dupont macho a macho de 10cm
- 7 cables dupont macho a macho 20cm
- 7 cables dupont macho a macho 30cm

Anexo 3: Códigos de funcionamiento

```

/*
  Nombre: Robot esquiwaobstáculos
  Autor: Jhoan sebastian palacios y Julian Piedrahita
  Fecha: Junio 2024
  Funcionalidad: Este subproyecto forma parte de la construcción de nuestro
  robot autónomo.
  En este subproyecto usaremos el driver L298n para gobernar 2 motores de CC
  que realizarán
  la parte mecánica de nuestro robot móvil.
  En este subproyecto nuestro robot irá recto y cuando encuentre un
  obstáculo a menos de 5 cm
  girará en sentido horario hasta que esquive el obstáculo y siga su camino
  */
#include <Servo.h>

// Definición de variables y constantes relacionadas con el motor izquierdo
const int IN1 = 13; // Pin digital 13 para controlar sentido giro motor
izquierdo
const int IN2 = 12; // Pin digital 12 para controlar sentido giro motor
izquierdo

// Definición de variables y constantes relacionadas con el motor derecho
const int IN3 = 11; // Pin digital 11 para controlar sentido giro motor
izquierdo
const int IN4 = 10; // Pin digital 10 para controlar sentido giro motor
izquierdo

// Este programa usará el sensor de ultrasonidos HCSR04
// para medir la distancia a la que se encuentran
// diferentes obstáculos en centímetros y mostrarlo por el puerto serie.
const int triggerEmisor = 3;
const int echoReceptor = 2;
const int valorUmbral = 20;
long tiempoEntrada; // Almacena el tiempo de respuesta del sensor de
entrada
float distanciaEntrada; // Almacena la distancia en cm a la que se
encuentra el objeto

int distancia = 0;
int distanciaD = 0;
int distanciaIz = 0;
int distanciaDerecha = 0;

```

```

int distanciaIzquierda = 0;

Servo servo1;      // crea objeto

int PINSERVO = 4;  // pin conectado a señal del servo
int PULSOMIN = 1000; // pulso minimo en microsegundos
int PULSOMAX = 2000; // pulso maximo en microsegundos

// Función que se ejecuta una sola vez al cargar el programa
void setup()
{
    servo1.attach(PINSERVO, PULSOMIN, PULSOMAX); // inicializacion de servo

    // Se declaran todos los pines como salidas
    pinMode (IN1, OUTPUT);
    pinMode (IN2, OUTPUT);
    pinMode (IN3, OUTPUT);
    pinMode (IN4, OUTPUT);

    pinMode(triggerEmisor,OUTPUT); // El emisor emite por lo que es
    configurado como salida
    pinMode(echoReceptor,INPUT); // El receptor recibe por lo que es
    configurado como entrada
    Serial.begin(9600); // Inicia el puerto de comunicaciones en serie
}
// Función que se repite de manera periódica
void loop()
{
    distancia = medirDistancia();
    if(distanciaEntrada > 30){
        robotAvance();
        Serial.println("MIRANDO HACIA DELANTE CON DISTANCIA DE:"+
String(distancia));
        delay(200);
    }
    if (distanciaEntrada < 39){
        robotParar();
        delay(250);
        robotRetroceso();
        delay(500);
        robotParar();
        delay(250);
        distanciaD = mirarDerecha();
    }
}

```

```

    delay(1000);
    distanciaIz = mirarIzquierda();
    delay(1000);

    if(distanciaD > distanciaIz)
    {
        robotDerecha();
        Serial.println("GIRO HACIA LA DERECHA");
        delay(200);
        robotParar();
    }
    else if(distanciaIz > distanciaD)
    {
        robotIzquierda();
        Serial.println("GIRO HACIA LA IZQUIERDA");
        delay(200);
        robotParar();
    }
}
}

int mirarDerecha()
{
    servo1.write(0);
    digitalWrite(triggerEmisor,LOW); // Para estabilizar
    delayMicroseconds(10);
    delay(700);
    int distanciaD = distanciaDerecha;
    servo1.write(90);

    // Comenzamos las mediciones
    // Se envía una señal activando la salida trigger durante 10
microsegundos
    digitalWrite(triggerEmisor, HIGH); // envío del pulso ultrasónico
    delayMicroseconds(10);
    tiempoEntrada=pulseIn(echoReceptor, HIGH);
    distanciaDerecha = int(0.017*tiempoEntrada); // Fórmula para calcular la
distancia en cm
    Serial.println("El valor de la distancia a la derecha es: "+
String(distanciaDerecha));
    delay(200);
    return distanciaD;
}

```

```
int mirarIzquierda()
{
    servo1.write(200);
    digitalWrite(triggerEmisor,LOW); // Para estabilizar
    delayMicroseconds(10);
    delay(700);
    int distanciaIz =distanciaIzquierda;
    servo1.write(90);

    // Comenzamos las mediciones
    // Se envía una señal activando la salida trigger durante 10
microsegundos
    digitalWrite(triggerEmisor, HIGH); // envío del pulso ultrasónico
    delayMicroseconds(100);
    tiempoEntrada=pulseIn(echoReceptor, HIGH);
    distanciaIzquierda = int(0.017*tiempoEntrada); // Fórmula para calcular
la distancia en cm
    Serial.println("El valor de la distancia a la izquierda es: "+
String(distanciaIzquierda));
    delay(200);
    return distanciaIz;
}

int medirDistancia()
{
    digitalWrite(triggerEmisor,LOW); // Para estabilizar
    delayMicroseconds(10);

    // Comenzamos las mediciones
    // Se envía una señal activando la salida trigger durante 10
microsegundos
    digitalWrite(triggerEmisor, HIGH); // envío del pulso ultrasónico
    delayMicroseconds(100);
    tiempoEntrada=pulseIn(echoReceptor, HIGH);
    distanciaEntrada= int(0.017*tiempoEntrada); // Fórmula para calcular la
distancia en cm
    Serial.println("El valor de la distancia al frente es: "+
String(distanciaEntrada));
    delay(200);

}

/*
```

```

    Función robotAvance: esta función hará que ambos motores se activen a
máxima potencia
    por lo que el robot avanzará hacia delante
    */
void robotAvance()
{
    // Motor izquierdo
    // Al mantener un pin HIGH y el otro LOW el motor gira en un sentido
digitalWrite (IN1, HIGH);
digitalWrite (IN2, LOW);
    // Motor derecho
    // Al mantener un pin HIGH y el otro LOW el motor gira en un sentido
digitalWrite (IN3, HIGH);
digitalWrite (IN4, LOW);

    Serial.println("IR A DELANTE");
}
/*
    Función robotRetrosceso: esta función hará que ambos motores se activen a
máxima potencia
    en sentido contrario al anterior por lo que el robot avanzará hacia
atrás
    */
void robotRetrosceso()
{
    // Motor izquierdo
    // Al mantener un pin LOW y el otro HIGH el motor gira en sentido
contrario al anterior
digitalWrite (IN1, LOW);
digitalWrite (IN2, HIGH);
    // Motor derecho
    // Al mantener un pin LOW y el otro HIGH el motor gira en sentido
contrario al anterior
digitalWrite (IN3, LOW);
digitalWrite (IN4, HIGH);

    Serial.println("REVERSA");
}
/*
    Función robotDerecha: esta función accionará el motor izquierdo y
parará el derecho
    por lo que el coche girará hacia la derecha (sentido horario)
    */
void robotDerecha()
{

```

```

// Motor izquierdo
// Se activa el motor izquierdo
digitalWrite (IN1, HIGH);
digitalWrite (IN2, LOW);
// Motor derecho
// Se para el motor derecho
digitalWrite (IN3, LOW);
digitalWrite (IN4, LOW);

Serial.println("GIRAR HACIA LA DERECHA");
delay(400);

}
/*
Función robotIzquierda: esta función accionará el motor derecho y
parará el izquierdo
por lo que el coche girará hacia la izquierda (sentido antihorario)
*/
void robotIzquierda ()
{
// Motor izquierdo
// Se para el motor izquierdo
digitalWrite (IN1, LOW);
digitalWrite (IN2, LOW);
// Motor derecho
// Se activa el motor derecho
digitalWrite (IN3, HIGH);
digitalWrite (IN4, LOW);

Serial.println("GIRAR HACIA LA IZQUIERDA");
delay(400);

}
/*
Función robotParar: esta función parará ambos motores
por lo que el robot se parará.
*/
void robotParar()
{
// Motor izquierdo
// Se para el motor izquierdo
digitalWrite (IN1, LOW);
digitalWrite (IN2, LOW);
// Motor derecho
// Se para el motor derecho

```

```
digitalWrite (IN3, LOW);  
digitalWrite (IN4, LOW);  
  
Serial.println("FRENANDO");  
}
```

Anexo 4: Evidencia fotográfica de la practica

