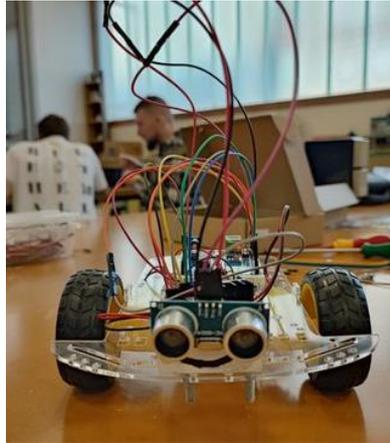


ROBOT ESQUIVA OBSTÁCULOS CON SENSOR FIJO

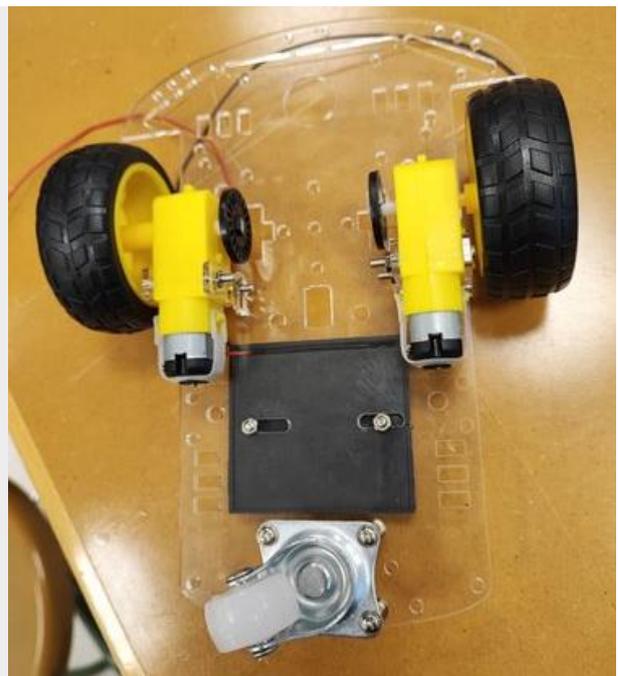
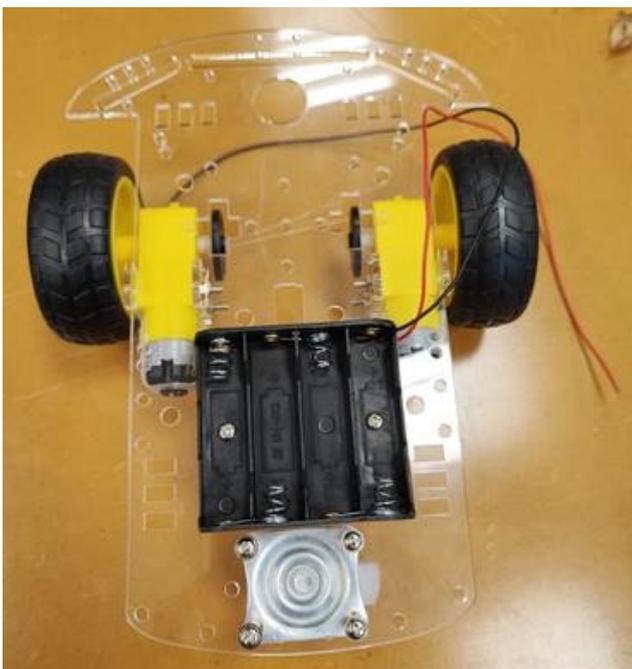


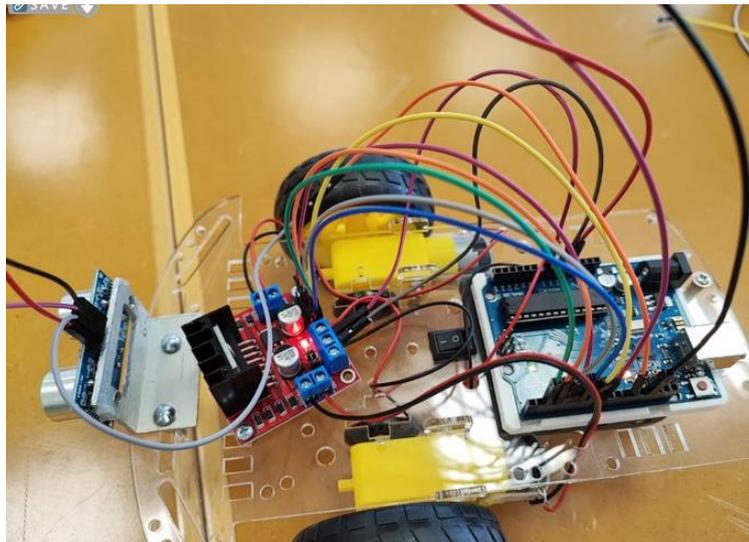
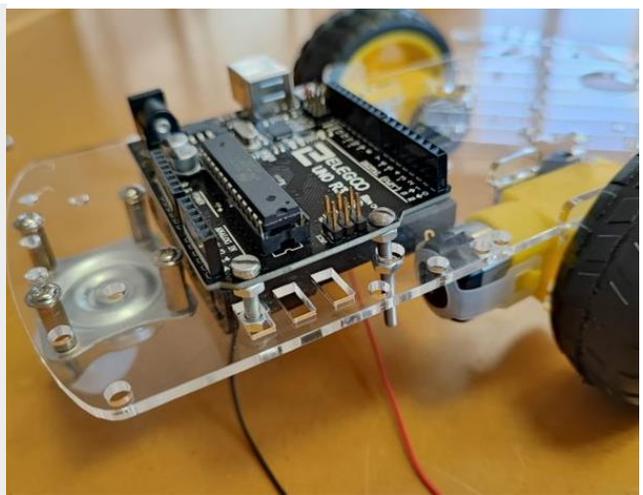
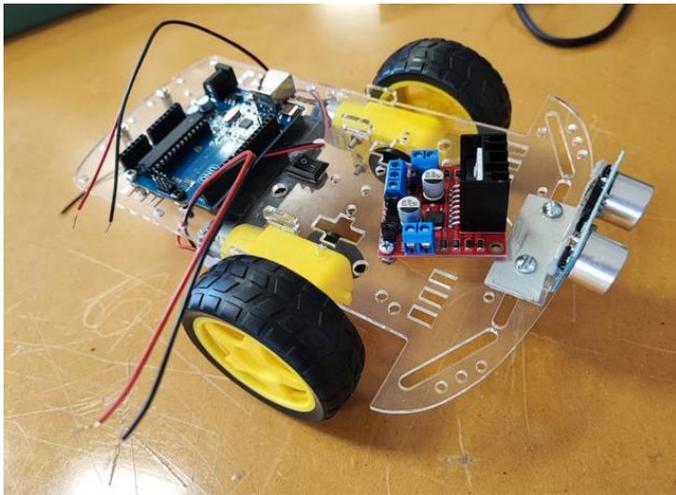
En este proyecto nos marcamos el objetivo de desarrollar un robot capaz de esquivar objetos. Un robot que avanzará en línea recta mientras que no detecte ningún obstáculo; en caso de detectar algún estorbo, girará sobre sí mismo hasta que esquive el objeto. Vamos a utilizar para ello una placa de Arduino UNO, un controlador de motores y un sensor de distancia.

Material necesario:

- Placa Arduino
- Módulo controlador de motores L298N.
- Módulo HC-SR04, sensor de ultrasonido y soporte.
- Chasis robot Arduino 2WD.
- 2 ruedas.
- Rueda loca.
- 2 motores reductores TT de doble eje.
- Interruptor ON/OFF de 2 pines.
- Portapilas para la alimentación de la placa y 4 pilas AA.

Montaje





Los conectores INA e INB se utilizarán para controlar la velocidad de cada uno de los motores, realizaremos la conexión usando los pines PWM (Pulse-Width Modulation) que aparecen etiquetados con el símbolo ~ (son los pines digitales 11, 10, 9, 6, 5, y 3). IN1, IN2, IN3 e IN4 son los responsables del giro del motor, cambiando HIGH y LOW podemos cambiar el sentido de giro

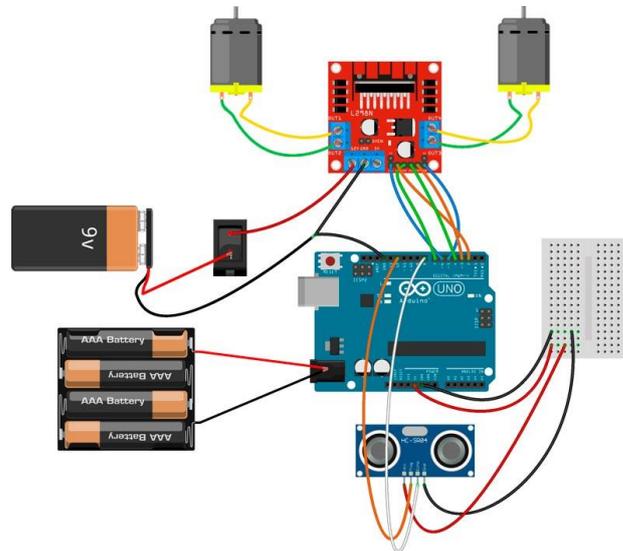
- ENA con el pin 6
- IN1 con el pin 2
- IN2 con el pin 7
- IN3 con el pin 4
- IN4 con el pin 3
- ENB con el pin 5

A continuación, conectaremos los motores con el módulo L298N, podemos encontrar 4 conexiones etiquetadas como OUT1, OUT2, OUT3 y OUT4. En este montaje se han conectado los cables del **motor derecho** en las conexiones **OUT1 (negro) y OUT2 (rojo)**. Los cables del **motor izquierdo** se han colocado en **OUT3 (rojo) y OUT4 (negro)**.

El **interruptor** que se colocó en la parte central se va a utilizar para cortar la alimentación que suministra la pila de 9V a los motores y al controlador. Por tanto, tendremos que conectar el **cable rojo** al interruptor. Unimos ese cable al interruptor y desde la otra patilla del interruptor pasaremos un cable rojo a la alimentación del controlador. El **cable negro**, de toma de tierra, debe introducirse en el **conector central** de los 3 conectores de alimentación del controlador. Pero además, hay que tener en cuenta que

la toma de tierra de controlador L298N **debe conectarse también a una de las conexiones de toma de tierra (GND) de la placa Arduino.**

Para el cableado del sensor de ultrasonidos, conectaremos GND e Vcc a la placa Arduino, y uniremos **Echo** al pin 8 de la placa y, por último, **Trig**, al pin digital 12 de Arduino.



* PARA ESTA VERSIÓN CON EL PORTAPILAS ES SUFICIENTE PAR ALIMENTAR LA PLACA ARDUINO Y EL MÓDULO CONTROLADOR DE MOTORES L298N.

Código Arduino

Antes de usar el programa completo, hay que probar que las ruedas funcionan correctamente, es decir, que giran correctamente. Para ello utilizaremos el programa:: *"prueba_ruedas.ino"*

```
[code]
// MOTOR 1
int IN1 = 2;
int IN2 = 7;
int ENA = 6;
// MOTOR 2
int IN3 = 4;
int IN4 = 3;
int ENB = 5;

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
}

void loop() {
  //avanza();
  //retrocede();
  //derecha();
  izquierda();
}
```

```

void parada(uint16_t tiempo) {
  parar();
  delay(tiempo);
}
void avanza(){
  // MOTOR 1
  analogWrite (ENA, 150);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  // MOTOR 2
  analogWrite(ENB, 150);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
void retrocede(){
  // MOTOR 1
  digitalWrite(ENA, 150);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  // MOTOR 2
  digitalWrite(ENB, 150);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void derecha(){
  // MOTOR 1
  analogWrite (ENA, 150);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  // MOTOR 2
  analogWrite (ENB, 150);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
void izquierda(){
  // MOTOR 1
  analogWrite (ENA, 150);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  // MOTOR 2
  analogWrite (ENB, 150);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void parar(){
  // MOTOR 1
  digitalWrite(ENA, 0);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  // MOTOR 2
  digitalWrite(ENB, 0);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
[/code]

```

Si al probar esto no giran correctamente, puede que haya que cambiar el cableado de los motores (cambiar el sentido de giro).

Una vez comprobado esto, cargaremos el programa completo: ***“robot_esquivaobst_sensor_fijo.ino”***

Para este programa se ha utilizado la biblioteca "New Ping". Esta biblioteca no viene por defecto en el programa de Arduino por lo que hay que descargarla y añadirla antes de subir la programación a la placa.

```
#include <NewPing.h>

#define PIN_TRIG 12 // Pin del Arduino conectado al pin Trigger del sensor de ultrasonidos
#define PIN_ECHO 8 // Pin del Arduino conectado al pin Echo del sensor de ultrasonidos
#define MAX_DISTANCIA 100 // Distancia máxima a detectar en cm.

NewPing sonar(PIN_TRIG, PIN_ECHO, MAX_DISTANCIA);

// MOTOR 1
int IN1 = 2;
int IN2 = 7;
int ENA = 6;

// MOTOR 2
int IN3 = 4;
int IN4 = 3;
int ENB = 5;

long randomNumber;

void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  randomSeed(analogRead(A0));
}

void loop() {
  delay(1000);
  int tiempo = sonar.ping_median();
  int distancia = tiempo / US_ROUNDTRIP_CM;
  // Imprimir el tiempo medido en la consola
  Serial.print("Tiempo: ");
  Serial.print(tiempo);
  Serial.println(" microsegundos");
  // Imprimir la distancia medida en la consola
  Serial.print("Distancia: ");
```

```
// US_ROUNDTRIP_CM constante para determinar la distancia. Convertir el tiempo en distancia (0 = indica fuera de rango)
```

```
Serial.print(distancia);
```

```
Serial.println(" cm");
```

```
/* */
```

```
if (distancia > 0){ // hay espacio
```

```
  if(distancia < 30){
```

```
    //parar();
```

```
    if(distancia > 20){
```

```
      retrocede();
```

```
      delay(200);
```

```
      parada(100);
```

```
    }else{
```

```
      randomNumber = random(1,3);
```

```
      Serial.print("El numero aleatorio es = ");
```

```
      Serial.println(randomNumber);
```

```
      if (randomNumber == 1){
```

```
        izquierda();
```

```
        delay(400);
```

```
        parada(500);
```

```
      }else{
```

```
        derecha();
```

```
        delay(400);
```

```
        parada(500);
```

```
      }
```

```
    }
```

```
  }else{
```

```
    avanza();
```

```
    //delay(200);
```

```
    //parada(500);
```

```
  }
```

```
  }else{
```

```
    avanza();
```

```
  }
```

```
}
```

```
void parada(uint16_t tiempo) {
```

```
  parar(); // Para los motores
```

```
    delay(tiempo); // Espera el tiempo que se le indique.
}

void avanza(){
    // MOTOR 1
    analogWrite (ENA, 150);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    // MOTOR 2
    analogWrite(ENB, 150);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void retrocede(){
    // MOTOR 1
    digitalWrite(ENA, HIGH);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    // MOTOR 2
    digitalWrite(ENB, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void derecha(){
    // MOTOR 1
    analogWrite (ENA, 150);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    // MOTOR 2
    analogWrite (ENB, 150);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void izquierda(){
    // MOTOR 1
    analogWrite (ENA, 150);
    digitalWrite(IN1, LOW);
```

```
digitalWrite(IN2, HIGH);  
  
// MOTOR 2  
  
analogWrite (ENB, 150);  
  
digitalWrite(IN3, HIGH);  
  
digitalWrite(IN4, LOW);  
  
}  
  
void parar(){  
  
// MOTOR 1  
  
digitalWrite(ENA, 0);  
  
digitalWrite(IN1, LOW);  
  
digitalWrite(IN2, LOW);  
  
// MOTOR 2  
  
digitalWrite(ENB, 0);  
  
digitalWrite(IN3, LOW);  
  
digitalWrite(IN4, LOW);
```