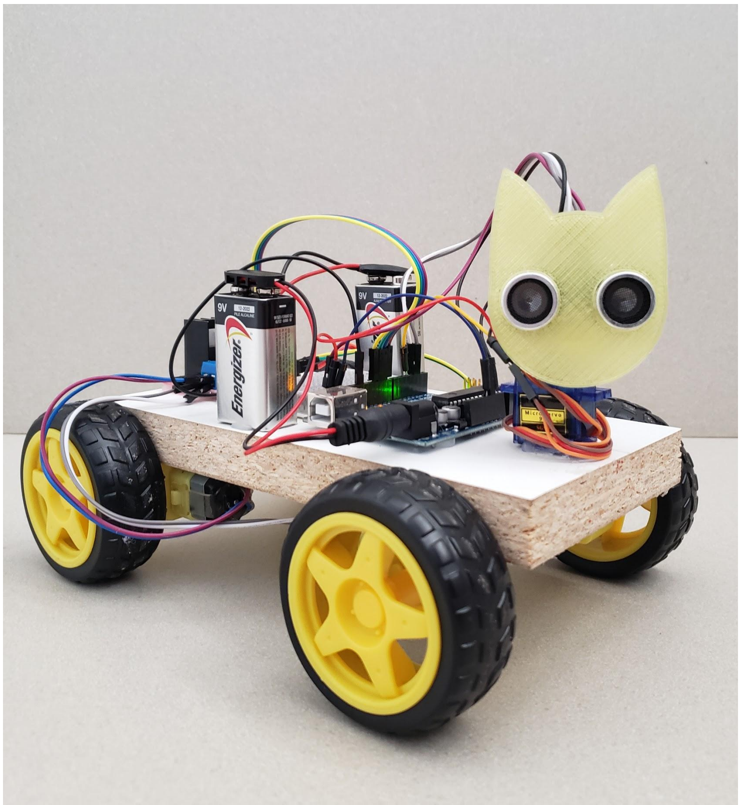
K-CiMiRO

(Casimiro)

Obstacle Avoiding Arduino Car



Manufactured by *R*^{2}

Table of Contents

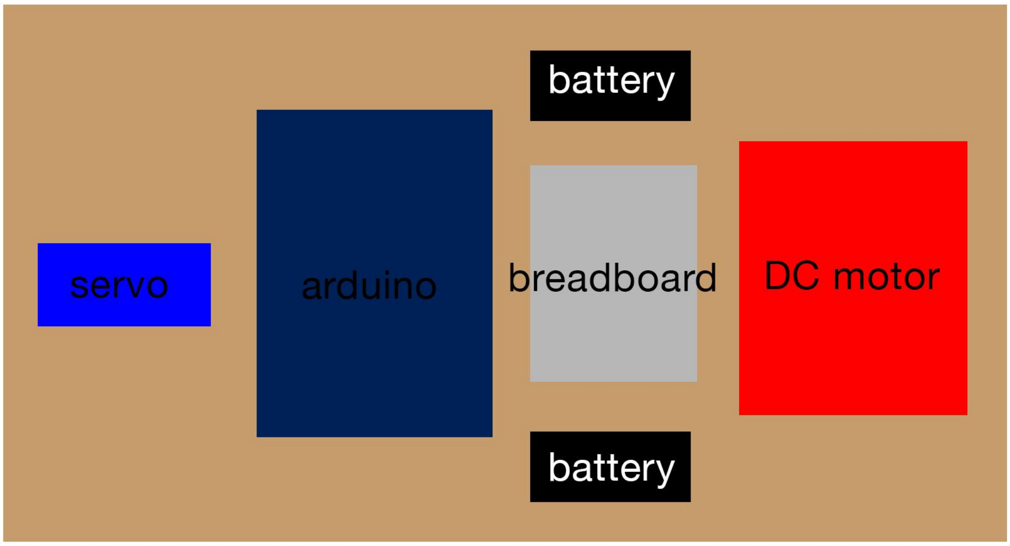
By: Nathalia Dias Regiani & Victoria Andersag Rojas

Materials

1. Arduino UNO Board
2. HC-SR04 Ultrasonic Sensor
3. Servo Motor SG90
4. Arduino DC Motor Control - L298N
5. Small Breadboard
6. Base for the car (wooden piece 19.5 cm x 9.5 cm)
7. (4) motors with wheels
8. A pack of male & female wires
9. (2) battery connectors (one with a DC Adaptor)
10. (2) 9 volt batteries
11. Hot glue gun
12. Screw driver
13. Ultrasonic sensor mount

Assembly Instructions

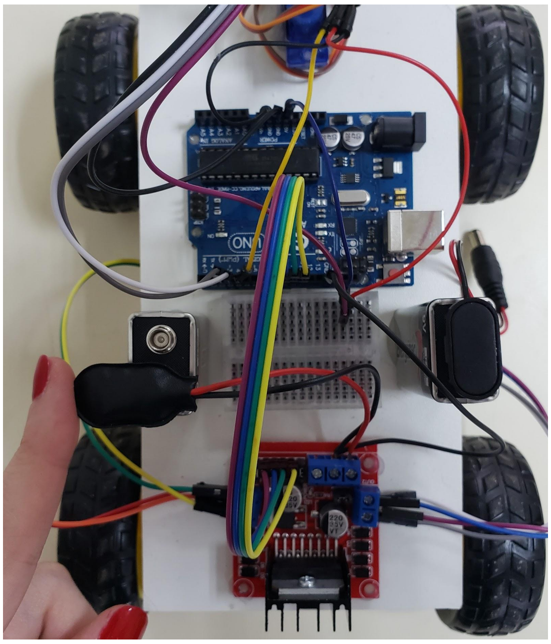
1. Attach the wheels to the motors.
2. With the hot glue gun, glue the motors to each corner of the base.
3. Solder two wires to each motor.
4. Glue the servo, Arduino, DC motor, breadboard, and two batteries on the top of the base following the picture below.

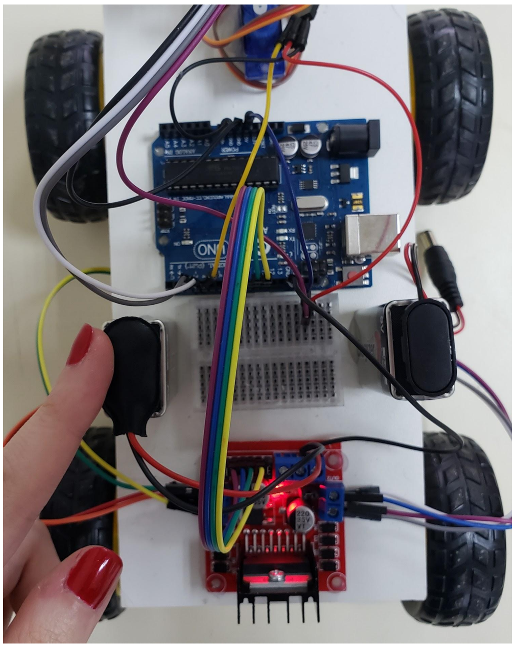


5. Screw the wires of the motors into the DC motor following the order of the circuit diagram found on page 6.
6. Connect the rest of the components with wires following the order of the circuit diagram.
7. Screw the ultrasonic sensor mount into the servo.
8. Attach the ultrasonic sensor to the mount.
9. Attach the battery connectors to both batteries.
    a. The one of the top should have the DC adaptor and the one on the bottom the normal one.
    b. For the one on the bottom, attach only one pin instead of the two.
10. Upload the code found on page 7 to the Arduino using the Arduino software which can be found for free on its website.

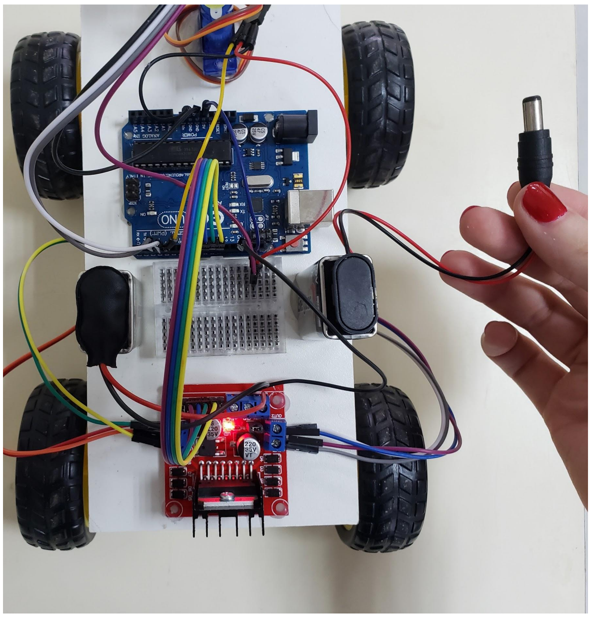
How to Turn On/Off

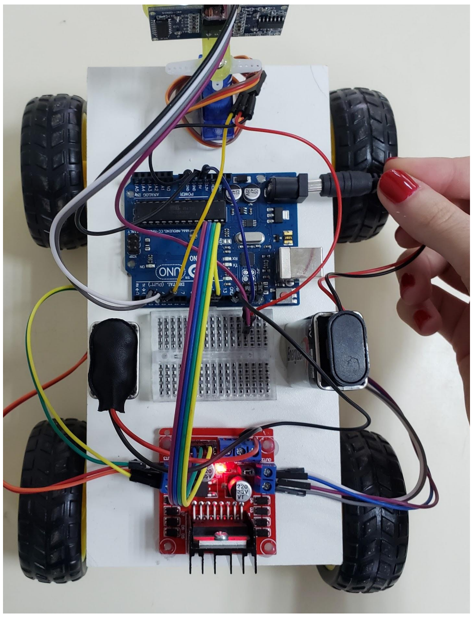
Since there is not an on and off switch on the robot, to turn on first completely connect both pins to the battery on the left.





Then connect the right battery with the DC adaptor to the Arduino.





To turn off, just disconnect the battery with the DC motor first and then plug off one the the pins from the battery on the left.

How It Works

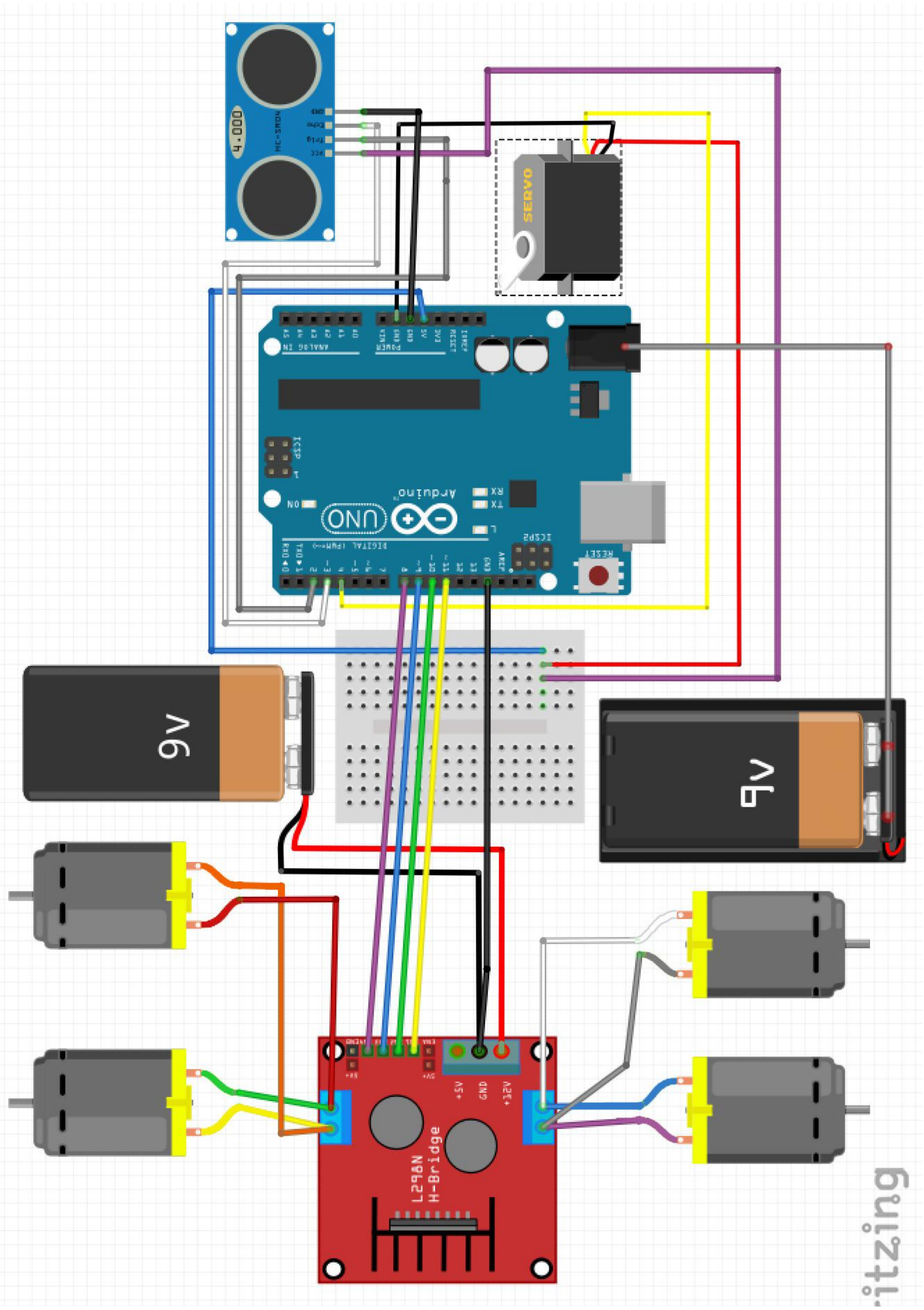
The main element of the car is the HC-SR04 Ultrasonic Sensor. The sensor is a 4 pin module, whose pins are Vcc (powers the sensor, typically with +5V), Trigger (input pin), Echo (output pin), and Ground (pin connected to the Ground of the system) respectively. The module has two eyes on the front which form the ultrasonic transmitter and receiver. The sensor works with the simple mathematical formula:

$$\textbf{Distance} = \textbf{Speed} \times \textbf{Time}$$

To calculate the distance using the formula above, we need the speed and time. The ultrasonic wave uses the universal speed at room conditions of 330m/s. The circuitry inbuilt on the module will calculate the time taken for the wave sent to come back. The echo pin turns on “high” for that particular amount of time, this way we can also know the time taken. Now the sensor calculates the distance using its microcontroller or microprocessor. The waves can be compared to the ones dolphin’s sound waves (where when they make a sound, waves are sent out and bounce on objects, making said waves come back and tell the dolphins about nearby objects).

Arduino: Microcontrollers to build digital devices.

Circuit Diagram



Arduino’s Code

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

#define ECHO 3 // Pin that receives the waves (and where the sensor connects to the arudino)
#define TRIGGER 2 // Pin that sends the waves (and where the sensor connects to the arudino)


int obstamax =35; // Maximum obstacle detection in centimeters
int obstamin= 5; // Minimum obstacle detection in centimeters
int i=0; // All numbers are real numbers

//////////////////////////

int ledPin0 =  11; // pins that connect the H-Bridge to the Arduino and creates them as a variables
int ledPin1 = 10;
int ledPin2 =  8;
int ledPin3 =  9;

//////////////////////////

int estado = 0;
int tt = 0;
int duracion;
int distancia;
int CMD; // Variable for cm to the right
int CMC; // Variable for cm forward
int CMI; // Variable for cm to the left
int CM; // Variables will be in centimeters

int angizq = 170; // How many degrees turned to the left of the servo
int angdere = 10; // How many degrees turned to the right of the servo
int angcent = 90; // Middle position of the servo
int a = 0;
int velocidad = 200; // How fast wheels turn

//////////////////////////

void setup()
{

myservo.attach(4); // Pins to connect servo to arduino
  pinMode(ECHO, INPUT); // Received the waves for sensor
  pinMode(TRIGGER, OUTPUT); // Sends the waves for sensor

pinMode(ledPin0, OUTPUT); // The H-Bridge transfers the energy of the wheels to arduino
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
myservo.write(angcent);
delay(3000); // 3 seconds of looking forward before reacting
motor_stop();
}

void loop()
{

        leer_obstaculo();// Reads and processes the obstacles

}

int DISTACM() // Starts the distance from zero so new screening can take place

  digitalWrite(TRIGGER, LOW);
  delayMicroseconds(2);

  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);

  digitalWrite(TRIGGER, LOW);
  duracion = pulseIn(ECHO, HIGH); // Waves are received by the sensor
  distancia =  (float (duracion))/53;
  delay(5);

  return distancia;


}
////////////////////////

void motor_stop() // How to treat the wheels if the car has to stop
{
  analogWrite(ledPin0,0);
  analogWrite(ledPin1,0);
  analogWrite(ledPin2,0);
  analogWrite(ledPin3,0);
}

////////////////////////

void motor_atras() // Which wheels to use if going backwards
{
   analogWrite(ledPin0,0);
   analogWrite(ledPin1,velocidad);
   analogWrite(ledPin2,0);
   analogWrite(ledPin3,velocidad);
}

////////////////////////
```

```
void motor_adelante() // Which wheels to use if going forward
{
  analogWrite(ledPin0,velocidad);
  analogWrite(ledPin1,0);
  analogWrite(ledPin2,velocidad);
  analogWrite(ledPin3,0);
}

/////////////////////////

void motor_izquierda() // Which wheels to use if going left

        analogWrite(ledPin0,0);
        analogWrite(ledPin1,velocidad);
        analogWrite(ledPin2,velocidad);
        analogWrite(ledPin3,0);
}

/////////////////////////

void motor_derecha() // Which wheels to use if going right

        analogWrite(ledPin0,velocidad);
        analogWrite(ledPin1,0);
        analogWrite(ledPin2,0);
        analogWrite(ledPin3,velocidad);

}
/////////////////////////

void leer_obstaculo()
{

 CM = DISTACM();// sets function for centimeters

if(CM <= obstamax && CM >=obstamin ) //checks if there are obstacles at 35cm of distance

        {
         motor_stop();

         myservo.write(angdere); // Servomotor right
         delay(500); // Waits 5 seconds

         CMD = DISTACM();// Checks how many cm are permitted

         myservo.write(angizq);//servomotor right
         delay(1000);

         CMI = DISTACM();// Checks how many cm are permitted

          if(CMI > CMD)// if there's an obstacle to the right
```

```
     {
      myservo.write(angcent); // Servo 180 degrees to the left
      delay(1000);
      motor_derecha(); // car moves to the left
      delay(3000);
     }

     if(CMI < CMD) // If there's an obstacle to the left

     {
      myservo.write(angcent); // Servo 180 degrees to the right
      delay(1000);
      motor_izquierda(); // Car moves to the right
      delay(3000);
     }
    }

//////////////////////////

 else

  { motor_adelante(); // If there's nothing to the sides, then car continues forward  }

 }
```

Batteries Disclaimer

There are two batteries on the robot. One is for the robot’s wheels while the other one is for the sensor and servo. The battery for the wheels can run for about 3 hours, while the other one can last for 9 hours. If the car starts to become slower or its response becomes erratic, check the batteries.