

Practica 2: cochecito

En la práctica tenemos que montar un cochecito y que realice las siguientes instrucciones:

1. Gestionar la lectura de dos sensores de ultrasonido desde un Arduino (mostrando los valores de distancia por pantalla – monitor serie--)
2. Uso del microcontrolador Arduino y el circuito (L298N, 2 puentes H) para el control de los 4 motores de corriente continua de un robot. Lograr que se ejecute la siguiente secuencia de movimientos (el robot se mueve hacia adelante 2 segundos, se para 1 segundo, se mueve hacia atrás 2 segundos, se para 1 segundo, gira hacia la derecha 2 segundos, se para 1 segundo, gira hacia la izquierda dos segundos, se para 1 segundo).
3. Enlazar las dos primeras prácticas. Hacer que el robot se mueve hacia adelante, cuando detecta un obstáculo cercano (usando los dos sensores de ultrasonido, por debajo de 0.5 metros, aprox., se detiene, se mueve hacia atrás 1 segundos, gira hacia la derecha 2 segundos, y vuelve a moverse hacia adelante).

El código lo dividí en tres archivos: las librerías para control de los motores y otra para el control de los sensores de ultrasonidos y el ultimo archivo es el “control_robot.ino” que es el encargado de contener el código de control del cochecito y las funciones principales del Arduino.

-Ultrasonidos.cpp:

/*

Libreria para el control de sensores de ultrasonidos

Creada por Manuel Fernandez Uceira el 06/02/2025

Ultima modificacion: 06/02/2025 12:47

*/

//Librerias necesarias

#include <arduino.h>

class ultrasonidos{

public:

ultrasonidos(){

trigger = 5;

echo = 4;

}

ultrasonidos(int trig, int Echo){

trigger = trig;

echo = Echo;

}

void iniciar()

{

pinMode(trigger, OUTPUT);

pinMode(echo, INPUT);

}

long medicion(){

long tiempo;

long distancia;

digitalWrite(trigger, LOW);

```
    delayMicroseconds(5);  
    digitalWrite(trigger, HIGH);  
    delayMicroseconds(15);  
    digitalWrite(trigger, LOW);  
  
    tiempo = pulseIn(echo, HIGH);  
    distancia = tiempo * 0.01657;  
  
    return (distancia);  
}  
private:  
int trigger, echo;  
};
```

-motores.cpp:

/*

Libreria para el control de motores con diferentes puentes H

Creada por Manuel Fernandez Uceira el 06/02/2025

Ultima modificacion: 06/02/2025 12:47

*/

#include <arduino.h>

class l298N{

private:

public:

int enableA, enableB, in1, in2, in3, in4;

l298N(){}

l298N(int EnableA, int EnableB, int IN1, int IN2, int IN3, int IN4){

enableA = EnableA;

enableB = EnableB;

in1 = IN1;

in2 = IN2;

in3 = IN3;

in4 = IN4;

}

void initMotor1()

{

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(enableA, OUTPUT);

```

    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enableA, 0);
}

void initMotor2()
{
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(enableB, OUTPUT);

    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    analogWrite(enableB, 0);
}
};

class coche : public l298N
{
public:
    coche(){}
    coche(int EnableA, int EnableB, int IN1, int IN2, int IN3, int IN4)
    {
        enableA = EnableA;
        enableB = EnableB;
        in1 = IN1;
        in2 = IN2;
        in3 = IN3;
        in4 = IN4;
    }
    //Funcion para iniciar los motores del coche

```

```
void iniciarCoche()
```

```
{
```

```
    initMotor1();
```

```
    initMotor2();
```

```
}
```

```
//Funciones de movimiento
```

```
void moverAdelante(int velocidad)
```

```
{
```

```
    digitalWrite(in1, HIGH);
```

```
    digitalWrite(in2, LOW);
```

```
    digitalWrite(in3, HIGH);
```

```
    digitalWrite(in4, LOW);
```

```
    analogWrite(enableA, velocidad);
```

```
    analogWrite(enableB, velocidad);
```

```
}
```

```
void moverAtras(int velocidad)
```

```
{
```

```
    digitalWrite(in1, LOW);
```

```
    digitalWrite(in2, HIGH);
```

```
    digitalWrite(in3, LOW);
```

```
    digitalWrite(in4, HIGH);
```

```
    analogWrite(enableA, velocidad);
```

```
    analogWrite(enableB, velocidad);
```

```
}
```

```
void moverDerecha(int velocidad)
```

```
{
```

```
    digitalWrite(in1, LOW);
```

```
    digitalWrite(in2, HIGH);
```

```
    digitalWrite(in3, HIGH);
```

```

        digitalWrite(in4,LOW);

        analogWrite(enableA, velocidad);
        analogWrite(enableB, velocidad);
    }
    void moverIzquierda(int velocidad)
    {
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4,HIGH);
        analogWrite(enableA, velocidad);
        analogWrite(enableB, velocidad);
    }

    //Funcion para parar el coche
    void parar()
    {
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        analogWrite(enableA, 0);
        analogWrite(enableB, 0);
    }
    private:

};

```

-control_robot.ino:

```
#include "ultrasonidos.cpp"
```

```
#include "motores.cpp"
```

```
//Inicializamos las variables necesarias para el funcionamiento del coche
```

```
int trig1 = 2, trig2 = 4, echo1 = 3, echo2 = 5;
```

```
int enableA = 6, in1 = 7, in2 = 8;
```

```
int enableB = 9, in3 = 10, in4 = 11;
```

```
int velocidad_recta = 150, velocidad_curva = 125;
```

```
//Inicializamos los sensores
```

```
ultrasonidos ultrasonidos1(trig1, echo1);
```

```
ultrasonidos ultrasonidos2(trig2, echo2);
```

```
//Inicializamos el coche
```

```
coche robot(enableA, enableB, in1, in2, in3, in4);
```

```
//Funcion para la demostracion del coche
```

```
void demostracion()
```

```
{
```

```
    robot.moverAdelante(velocidad_recta);
```

```
    delay(4000);
```

```
    robot.parar();
```

```
    delay(1000);
```

```
    robot.moverAtras(velocidad_recta);
```

```
    delay(4000);
```

```
    robot.parar();
```

```
    delay(1000);
```

```
    robot.moverDerecha(velocidad_curva);
```

```
    delay(4000);
```

```
    robot.parar();
```

```
    delay(1000);
```



```
robot.moverIzquierda(velocidad_curva);  
delay(4000);  
robot.parar();  
delay(1000);  
}
```

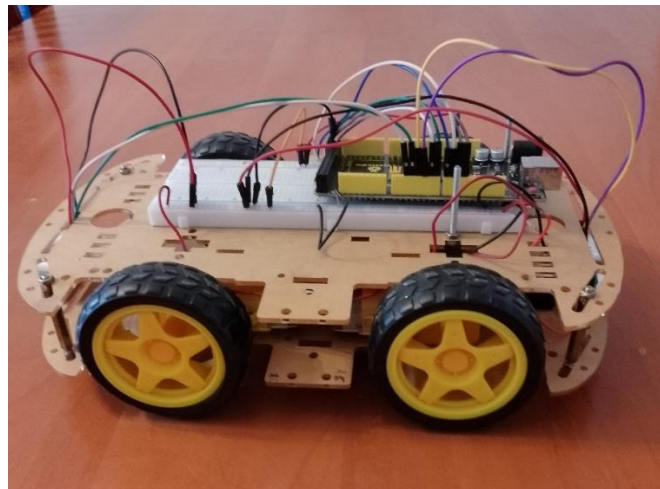
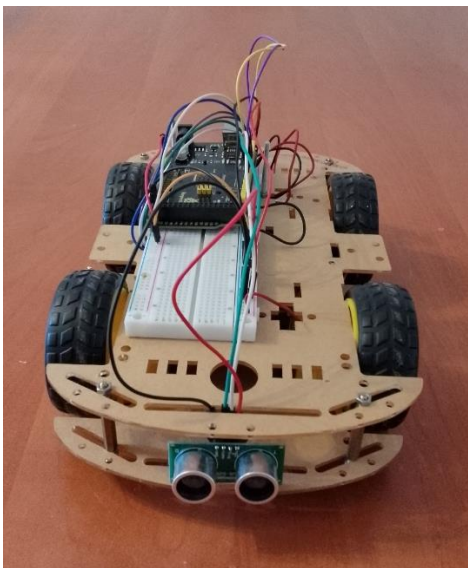
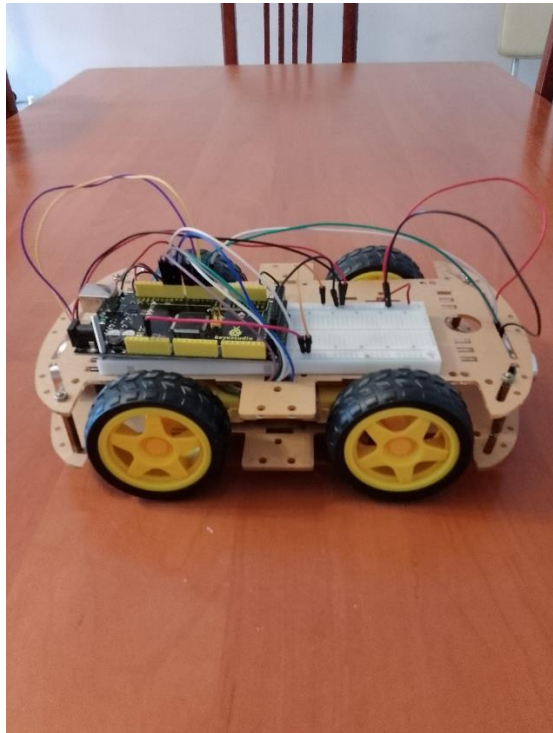
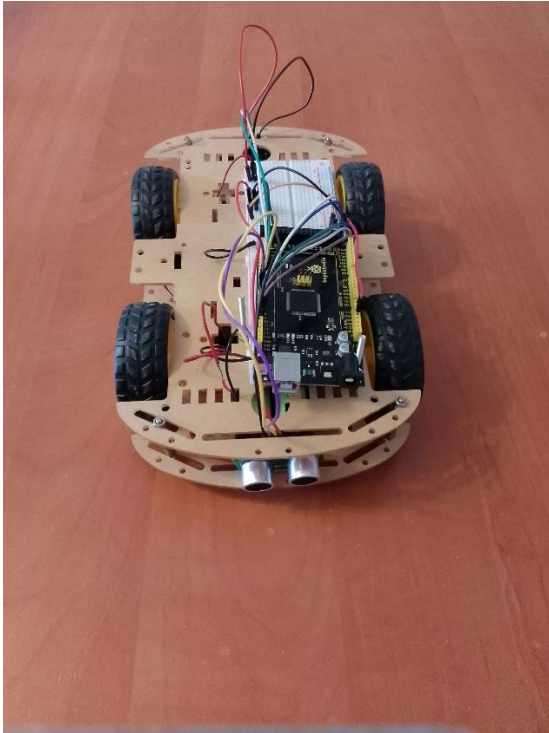
//Programa principal del arduino

```
void setup() {  
    // Inicializamos la conexion serie y las clases de control del robot  
    Serial.begin(9600);  
    ultrasonidos1.iniciar();  
    ultrasonidos2.iniciar();  
    robot.iniciarCoche();  
    //Esta es una espera para poder colocar el robot  
    delay(5000);  
    //Esta funcion se encarga de mover el robot de la forma indicada en el apartado 2 de la practica  
    demostracion();  
}
```

```
void loop() {  
    // Realizamos las mediciones delantera y trasera del robot  
    long medicionAdelante = ultrasonidos2.medicion();  
    long medicionAtras = ultrasonidos1.medicion();  
    //Mostramos por el monitor serie los resultados  
    Serial.print("Medicion de distancia delantera: "); Serial.println(medicionAdelante);  
    Serial.print("Medicion de distancia trasera: "); Serial.println(medicionAtras);  
    //Movemos el robot como se indica en el apartado 3 de la practica.  
    if ((medicionAdelante > 50)){  
        robot.moverAdelante(velocidad_recta);  
    }else if(medicionAtras > 50)
```

```
{  
  robot.parar();  
  delay(500);  
  robot.moverAtras(velocidad_recta);  
  delay(1000);  
  robot.parar();  
  delay(500);  
  robot.moverDerecha(velocidad_curva);  
  delay(2000);  
}else robot.parar();  
  
}
```

Imágenes del montaje del cochecito:



Enlaces al video de demostración y github con todo el código:

El video se encuentra dentro del repositorio de github.

https://github.com/mans199876/proyectos1_practica2.git