



# PROYECTO N° 4

## Esquiva obstáculos

## PROYECTO N° 4: Esquivar obstáculos

Aprende a crear un programa para que cuando el sensor de ultrasonidos del Code&Drive detecte un objeto o superficie plana a menos de 25 cm delante de sí, encienda los LEDs, se mueva hacia atrás y gire para esquivarlo. Una vez que haya esquivado el objeto, seguirá su recorrido en línea recta.

**NIVEL DE DIFICULTAD:** Intermedio.

**DURACIÓN DEL EJERCICIO:** 40 min.

**MATERIALES:**

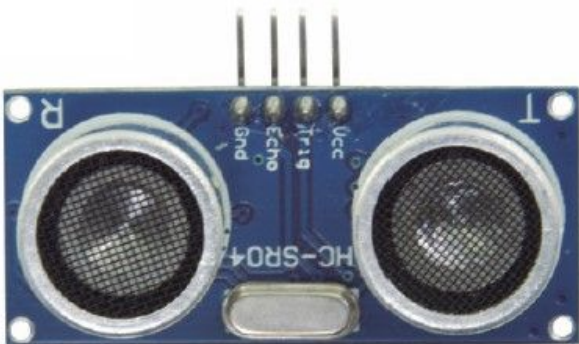
- 1 LED Verde.
- 1 LED Rojo.
- 1 Sensor de Ultrasonido.
- 2 motores con sus ruedas.
- 1 Cable USB - micro USB
- Ordenador

El kit Code&Drive deberá estar montado de acuerdo a las instrucciones indicadas en el manual.

### ¿Qué es un sensor de ultrasonido?

El sensor de ultrasonido es un dispositivo para medir la distancia. Su funcionamiento consiste en enviar un pulso de sonido a alta frecuencia, no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia el sensor, que dispone de un micrófono adecuado para esta frecuencia.

Midiendo el tiempo entre pulsos y conociendo la velocidad del sonido, podemos estimar la distancia del objeto, en cuya superficie impactó el impulso de ultrasonido.



**Sensor de ultrasonido**

### CONEXIONES:

1. Conecta los LEDs a los pines digitales "9" y "10".
2. Conecta el sensor de ultrasonido a los pines digitales "12" y "13".
3. Conecta los motores DC al conector gris de la placa Build&Code 4in1.

Recuerda que el cable rojo debe ir en las conexiones "A01" y "B01", y el negro en las conexiones "A02" y "B02".

### CÓDIGO DE PROGRAMACIÓN

Puedes realizar esta actividad utilizando los *software* Arduino y Bitbloq, además de otros *software* de programación por bloques compatibles. A continuación encontrarás el código de programación

necesario.

## Código Arduino

1. [Descarga el software Arduino](#) y realiza en proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente programa:

```
int PinSpeedMA = 5, PinSpeedMB = 6; // PIN DIGITAL PARA LA
VELOCIDAD DE LOS MOTORES
int PinTurnMA = 4, PinTurnMB = 7; // PIN DIGITAL PARA SENTIDO DE
GIRO DE LOS MOTORES
int TrigPin = 13; // PINES DEL SENSOR ULTRASONIDOS
int EchoPin = 12;
float SSound = 0.0343; //VELOCIDAD DEL SONIDO EN cm/us
long Lengh, Distance ; // VARIABLES PARA CALCULAR LA DISTANCIA EN
cm
int PinLED1 = 9, PinLED2 = 10; // PIN DIGITAL LED1 Y LED2

void setup() {
  // put your setup code here, to run once:
  //CONFIGURACIÓN DE LOS PUERTOS DIGITALES
  pinMode(PinSpeedMA, OUTPUT);
  pinMode(PinSpeedMB, OUTPUT);
  pinMode(PinTurnMA, OUTPUT);
  pinMode(PinTurnMB, OUTPUT);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
  pinMode(PinLED1, OUTPUT);
  pinMode(PinLED2, OUTPUT);
  // VELOCIDAD DE LOS MOTORES DE 100 A 255
  analogWrite(PinSpeedMA, 175);
  analogWrite(PinSpeedMB, 175);
}

void loop() {
  // put your main code here, to run repeatedly:
  DistanceCM(); // LLAMAR A LA FUNCIÓN PARA CALCULAR LA DISTANCIA
  if (( Distance < 25) && ( Distance > 1)) // SI LA DISTANCIA ES
  ENTRE 1 Y 25cm
  {
    digitalWrite(PinLED1,HIGH);//LED1 Y LED2 = ON
    digitalWrite(PinLED2,HIGH);
    digitalWrite(PinTurnMA, HIGH);// CODE&DRIVE HACIA ATRÁS
    digitalWrite(PinTurnMB, LOW);
    delay(1000);
  }
}
```

```
        digitalWrite(PinTurnMA, HIGH); // GIRO DEL CODE&DRIVE
        digitalWrite(PinTurnMB, HIGH);
        delay(1000);
    }
    else
    {
        digitalWrite(PinLED1, LOW); // LED1 Y LED2 = OFF
        digitalWrite(PinLED2, LOW);
        digitalWrite(PinTurnMA, LOW); // CODE&DRIVE HACIA DELANTE
        digitalWrite(PinTurnMB, HIGH);
    }
}

void DistanceCM() // FUNCIÓN DE CÁLCULO DE DISTANCIA
{
    // CALCUL DE LA DISTANCIA EN cm
    digitalWrite(TrigPin, LOW); // Nos aseguramos de que el
    trigger está desactivado
    delayMicroseconds(4); // Para asegurarnos de que el
    trigger está LOW
    digitalWrite(TrigPin, HIGH); // Activamos el pulso de
    salida
    delayMicroseconds(14); // Esperamos 10µs. El pulso
    sigue active este tiempo
    digitalWrite(TrigPin, LOW); // Cortamos el pulso y a
    esperar el ECHO
    Lengh = pulseIn(EchoPin, HIGH) ; //pulseIn mide el tiempo que
    pasa entre que el pin declarado (echoPin) cambia de estado bajo a
    alto (de 0 a 1)
    Distance = SSound* Lengh / 2; // CÁLCULO DE LA DISTANCIA
}
```

### Código para software de programación por bloques compatible

1. [Descarga el software](#) y realiza en proceso de instalación.
2. Abre el programa y, una vez en él, copia el siguiente código:

```

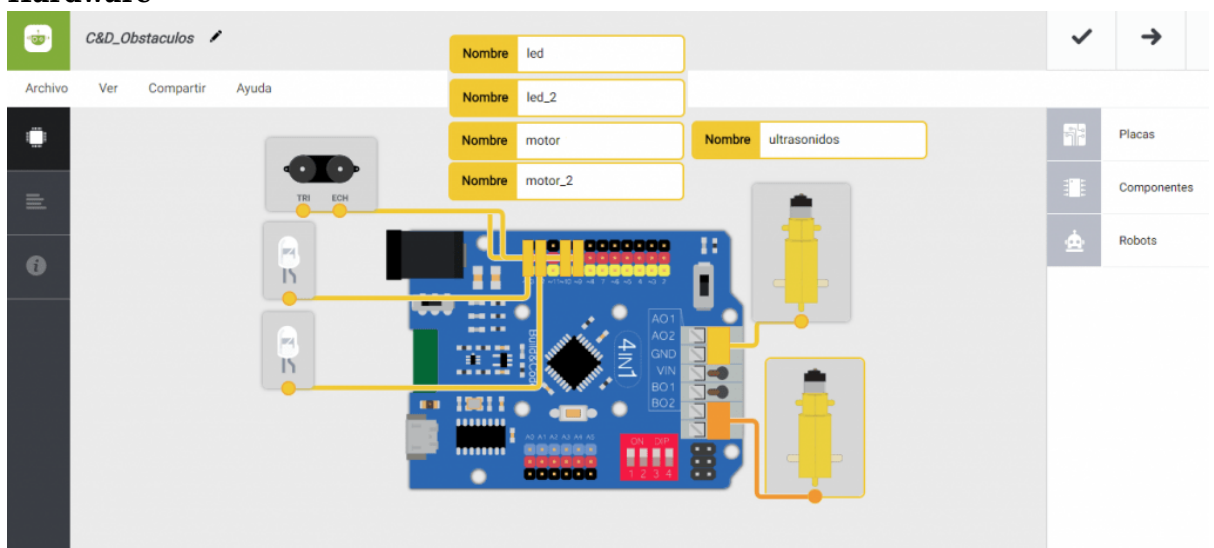
Programa de Arduino
fijar pin PWM 6 a 100 ▶ VELOCIDAD DE LOS MOTORES A 100 DE 255
fijar pin PWM 5 a 100
por siempre
si lee el sensor ultrasónico trig pin 13 echo pin 12 < 25 y lee el sensor ultrasónico trig pin 13 echo pin 12 > 1 entonces
  fijar salida pin digital 9 a ALTO ▶ Encender Leds
  fijar salida pin digital 10 a ALTO
  fijar salida pin digital 7 a BAJO ▶ CODE&DRIVE HACIA ATRAS
  fijar salida pin digital 4 a ALTO
  esperar 1 segundos ▶ ESPERA DE 1SEGUNDO
  fijar salida pin digital 7 a ALTO ▶ CODE&DRIVE GIRANDO
  fijar salida pin digital 4 a ALTO
  esperar 1 segundos ▶ ESPERA DE 1SEGUNDO
si no
  fijar salida pin digital 9 a BAJO ▶ Apagar los Leds.
  fijar salida pin digital 10 a BAJO
  fijar salida pin digital 7 a ALTO ▶ CODE&DRIVE HACIA DELANTE
  fijar salida pin digital 4 a BAJO
  
```

3. Configura y carga el código, siguiendo las instrucciones indicadas en la [guía de Primeros Pasos del Code&Drive](#).
4. Revisa que el interruptor BLT/USB de la placa controladora Build&Code 4in1 está en posición USB, para una correcta carga del código.

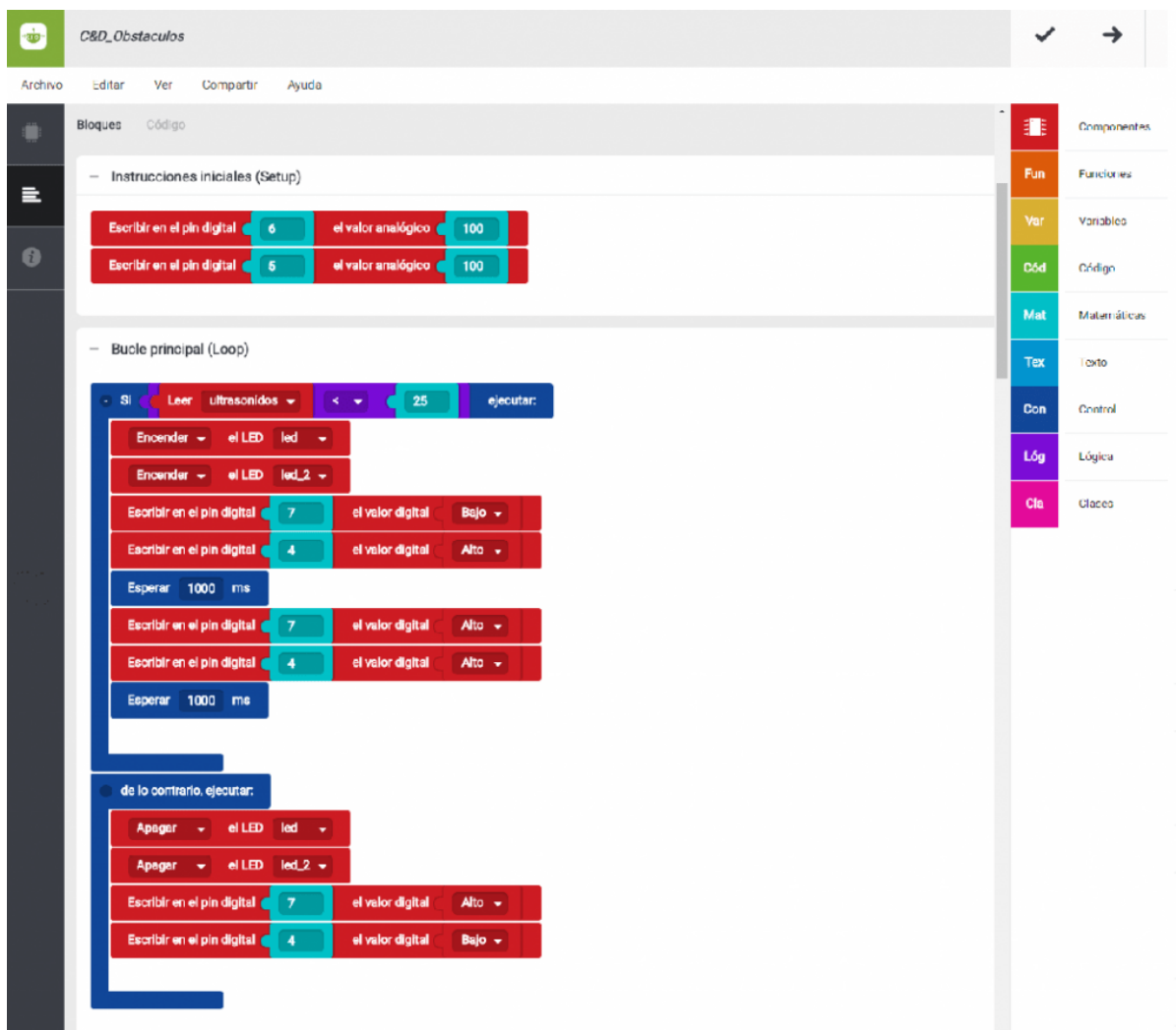
### Código Bitbloq

1. [Descarga el software Bitbloq](#) y realiza el proceso de instalación.
2. Abre el programa mBlock y, una vez en él, copia el siguiente código:

- o **Hardware**



- o **Software**



3. Configura y carga el código, siguiendo las instrucciones indicadas en la [guía de Primeros Pasos del Code&Drive](#).
4. Revisa que el interruptor BLT/USB de la placa controladora Build&Code 4in1 esté en posición USB, para una correcta carga del código.

### RESULTADO DEL EJERCICIO

El Code&Drive se moverá hacia delante y cuando el sensor de ultrasonido detecte un objeto a menos de 25 cm, encenderá los LEDs, retrocederá durante 1 segundo y dará la vuelta durante 1 segundo. Luego, comprobará que no haya ningún obstáculo, apagará los LEDs y seguirá su recorrido hacia delante.