

PROYECTO N° 3: Radar 180°

Aprende a construir y programar un radar de 180° utilizando los componentes del Maker Control Kit y el Maker Kit 3.

El sistema detectará cualquier objeto que se encuentre dentro de los 180° de alcance del radar y calculará su distancia. Si la distancia es menor a 20 cm, el LED rojo se encenderá. Si la distancia es mayor a 20 cm, el LED verde se encenderá.

NIVEL DE DIFICULTAD: Principiante.

DURACIÓN DEL EJERCICIO: 60 min.

MATERIALES:

- 1 Servomotor
- 1 Sensor de distancia por ultrasonidos
- 1 LED rojo
- 1 LED verde
- 1 Caja de cartón
- 1 Placa controladora Build&Code 4in1
- 1 Cable USB - Micro USB
- 1 Ordenador
- Material para crear el soporte del sensor de distancia por ultrasonidos
- Adhesivo o cinta adhesiva

¿Qué es un sensor de distancia por ultrasonidos?

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.

CONSTRUCCIÓN DE LA ESTRUCTURA:

Para el montaje del radar usarás una caja de cartón y para la estructura del sensor de distancia por ultrasonidos usarás láminas de cartón pluma.

[Descarga la guía rápida de montaje](#) y sigue los pasos indicados:

1. Para la estructura del sensor de distancia por ultrasonidos recorta dos trozos de cartón pluma. En un trozo haz dos agujeros para que los atraviesen el receptor y el emisor de ultrasonidos. Une las dos piezas de forma perpendicular y sujeta todo el conjunto con el eje del servomotor.
2. En el exterior de la caja pon el servomotor con el sensor de distancia por ultrasonidos, el LED rojo y el LED verde.
3. En el interior de la caja introduce la placa controladora Build&Code 4in1 y conecta todos los componentes electrónicos y el cable USB. Sigue las indicaciones del apartado de conexiones que se encuentra a continuación.

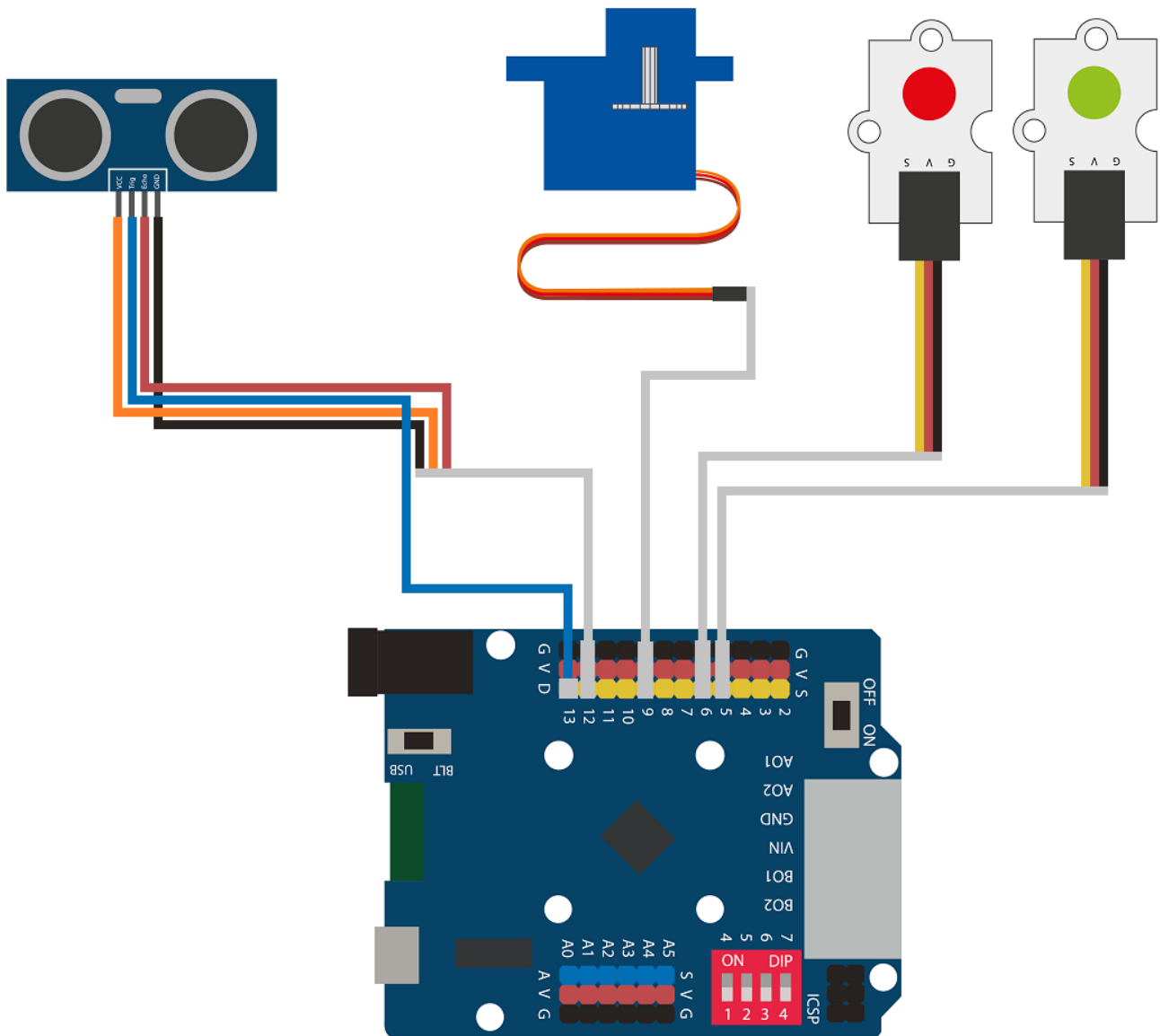
CONEXIONES:

- Conecta el LED verde al puerto digital 5 de la placa controladora Build&Code 4in1.
- Conecta el LED rojo al puerto digital 6 de la placa controladora Build&Code 4in1.
- Conecta el servomotor al puerto digital 9 de la placa controladora Build&Code 4in1.

Conecta el sensor de distancia por ultrasonidos al puerto digital 12 y 13 de la placa controladora Build&Code 4in1.

| Puerto digital 12 | Sensor de distancia por ultrasonidos |
|--------------------------|---------------------------------------------|
| V | VCC |
| G | GND |
| S / D | Echo |
| Puerto digital 13 | Sensor de distancia por ultrasonidos |
| S / D | Trig |

Para guiarte, mira los colores de los cables y los colores de los terminales de la placa controladora Build&Code 4in1. Cada cable debe ir conectado a su color:



CÓDIGO DE PROGRAMACIÓN:

El programa está compuesto por las siguientes partes:

1. Movimiento del servomotor 180° hacia la derecha y hacia la izquierda.
2. Durante el movimiento del servomotor, se mide la distancia con el sensor de distancia por ultrasonidos:
 - Si la distancia medida es menor a 20 cm, se enciende el LED rojo.
 - Si la distancia medida es mayor a 20 cm, se enciende el LED verde.

Esta secuencia se irá repitiendo mientras todo el conjunto tenga alimentación eléctrica.

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 para cada *software*.

Código Arduino

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

```
#include <Servo.h>
Servo motor1; // motor1 = Nombre del servomotor
int degreeM1; // Variables de los grados del servomotor
int LEDRed = 6, LEDGreen = 5; // LED rojo conectado puerto digital
5, LED verde conectado puerto digital 6
int TrigPin = 13; // Puertos de conexión del sensor de
ultrasonidos
int EchoPin = 12;
float SSound = 0.0343; //VELOCIDAD DEL SONIDO EN cm/us
long Lengh, Distance ; // VARIABLES PARA CALCULAR LA DISTANCIA EN
CM

void setup() {
  // Put your setup code here, to run once:
  motor1.attach (9); // Servomotor conectado puerto digital 9
  pinMode(TrigPin, OUTPUT); // Configuración puertos del sensor de
ultrasonidos
  pinMode(EchoPin, INPUT);
  pinMode(LEDRed, OUTPUT); // Configuración puertos de los LEDs
  pinMode(LEDGreen, OUTPUT);
}

void loop() {
  // Put your main code here, to run repeatedly:
  for (degreeM1 = 15; degreeM1 < 181; degreeM1 = degreeM1+5) //
Incrementar la variable degreeM1 +5 hasta ser más grande que 181
  {
    DistanceCM(); // Calcular la distancia en cm
    if (Distance < 20) // Si la distancia es más pequeña que 20 cm
    {
      digitalWrite (LEDRed, HIGH); // LED Rojo = ON
      digitalWrite (LEDGreen, LOW); // LED Verde = OFF
    }
    else // Si la distancia es mayor que 20 cm
    {
      digitalWrite (LEDRed, LOW); // LED rojo = OFF
    }
  }
}
```

```

        digitalWrite (LEDGreen, HIGH); // LED verde = ON
    }
    motor1.write(degreeM1); // Mover el servomotor a los grados de
degreeM1
    delay(200); // Espera de 200 ms
}
for (degreeM1 = 180; degreeM1 > 15; degreeM1 = degreeM1-5) //
Disminuir la variable degreeM1 -5 hasta ser más pequeño que 15
{
    DistanceCM(); // Calcular la distancia en cm
    if (Distance < 20) // Si la distancia más pequeña de 20 cm
    {
        digitalWrite (LEDRed, HIGH); // LED Rojo = ON
        digitalWrite (LEDGreen, LOW); // LED Verde = OFF
    }
    else // Si la distancia es mayor de 20 cm
    {
        digitalWrite (LEDRed, LOW); // LED rojo = OFF
        digitalWrite (LEDGreen, HIGH); // LED verde = ON
    }
    motor1.write(degreeM1); // Mover servomotor a los grados de
degreeM1
    delay(200); // Espera de 200 ms
}
}

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 10 µs. 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
}

```

3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

Código para *software* de programación por bloques compatible

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

```

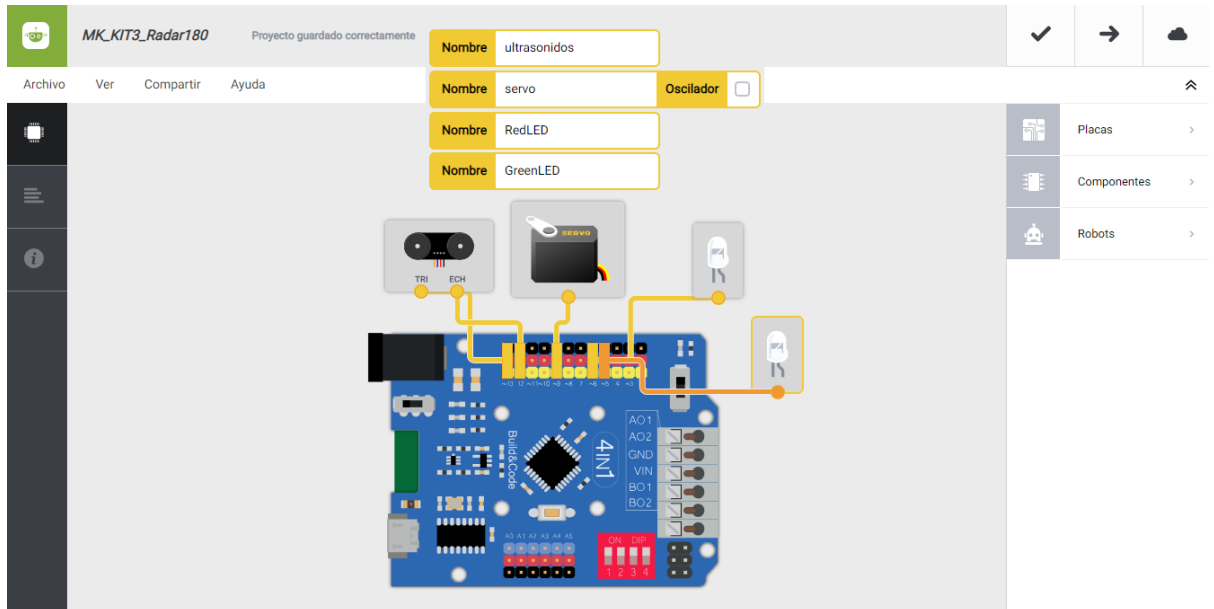
Programa de Arduino
fijar degree a 15
por siempre
  repetir hasta que degree > 181
    fijar ángulo del pin 9 del servo a degree
    cambiar degree por 5
    fijar distance a lee el sensor ultrasónico trig pin 13 echo pin 12
    si distance < 20 entonces
      fijar salida pin digital 6 a ALTO
      fijar salida pin digital 5 a BAJO
    si no
      fijar salida pin digital 6 a BAJO
      fijar salida pin digital 5 a ALTO
    esperar 0.2 segundos
  repetir hasta que degree < 15
    fijar ángulo del pin 9 del servo a degree
    cambiar degree por -5
    fijar distance a lee el sensor ultrasónico trig pin 13 echo pin 12
    si distance < 20 entonces
      fijar salida pin digital 6 a ALTO
      fijar salida pin digital 5 a BAJO
    si no
      fijar salida pin digital 6 a BAJO
      fijar salida pin digital 5 a ALTO
    esperar 0.2 segundos
  
```

3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

Código BitBloq

1. Accede al software [Bitbloq](#).
2. Abre el programa Bitbloq y, una vez en él, copia el siguiente código:

- **Hardware**



- **Software**

The screenshot displays the ebotics software interface for a project named 'MK_KIT3_Radar180'. The interface is divided into several sections:

- Variables globales y funciones:** A 'Declarar variable' block is used to declare a variable named 'degreeM1' and set its value to 0.
- Instrucciones iniciales (Setup):** A 'Mover servo' block is configured to move the servo to 15 degrees.
- Bucle principal (Loop):** The main loop consists of two 'Contar con' (for) loops:
 - The first loop counts from 15 to 181, summing by 5. Inside this loop, there is an 'SI' (if) block that reads ultrasonic sensors. If the reading is less than 20, it triggers a sequence of actions: turning on the RedLED, turning off the GreenLED, moving the servo to the current value of 'degreeM1', and waiting for 200ms.
 - The second loop counts from 181 to 15, subtracting by 5. It contains an identical sequence of actions as the first loop.

3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

RESULTADO DEL EJERCICIO

Con el servomotor y el sensor de distancia por ultrasonidos has fabricado el sistema para medir la distancia con un movimiento giratorio de 180°. Los LEDs servirán como indicadores: si ha detectado un objeto, se encenderá el LED rojo y si no ha detectado ningún objeto, se encenderá el LED verde. ¡Has creado un radar de 180° de movimiento giratorio!