

PROJECT Nº 3: 180° radar

Learn to build and program a 180° radar using the components of the Maker Control Kit and Maker Kit 3.

The system will detect any object that it finds within 180° of the radar's scope and calculate its distance. If the distance is less than 20 cm, the red LED will turn on. If the distance is greater than 20 cm, the green LED will turn on.

DIFFICULTY LEVEL: Beginner.

DURATION OF THE EXERCISE: 60 min.

MATERIALS:

- 1 Servomotor

- 1 Ultrasonic distance sensor
- 1 Red LED
- 1 Green LED
- 1 Cardboard box
- 1 4in1 Build&Code board
- 1 USB - Micro USB cable
- 1 Computer
- Material to create the support for the ultrasonic distance sensor
- Adhesive or tape

What is an ultrasonic distance sensor?

The ultrasonic sensor is a device used to measure distance. It operates by emitting a high-frequency sound pulse, not audible to the human ear. This pulse bounces off of nearby objects and is reflected toward the sensor, which has a microphone capable of picking up this frequency.

By measuring the time between pulses and knowing the speed of sound, we can estimate the distance of an object whose surface was hit by the ultrasonic pulse.

BUILDING THE STRUCTURE:

To assemble the radar, you will use a cardboard box, and for the structure of the ultrasonic distance sensor, you will use sheets of poster board.

[Download the quick assembly guide](#) and follow the steps shown:

1. For the structure of the ultrasonic distance sensor, cut out two pieces of poster board. On one piece make two holes so that the ultrasonic receptor and transmitter can go through them. Join the pieces perpendicularly and fasten the whole set to the servomotor shaft.
2. Put the servomotor with the ultrasonic distance sensor, the red LED, and the green LED on the outside of the box.
3. Put the 4in1 Build&Code board inside the box and connect all the electronic components and the USB cable. Follow the indications in the section on connections found below.

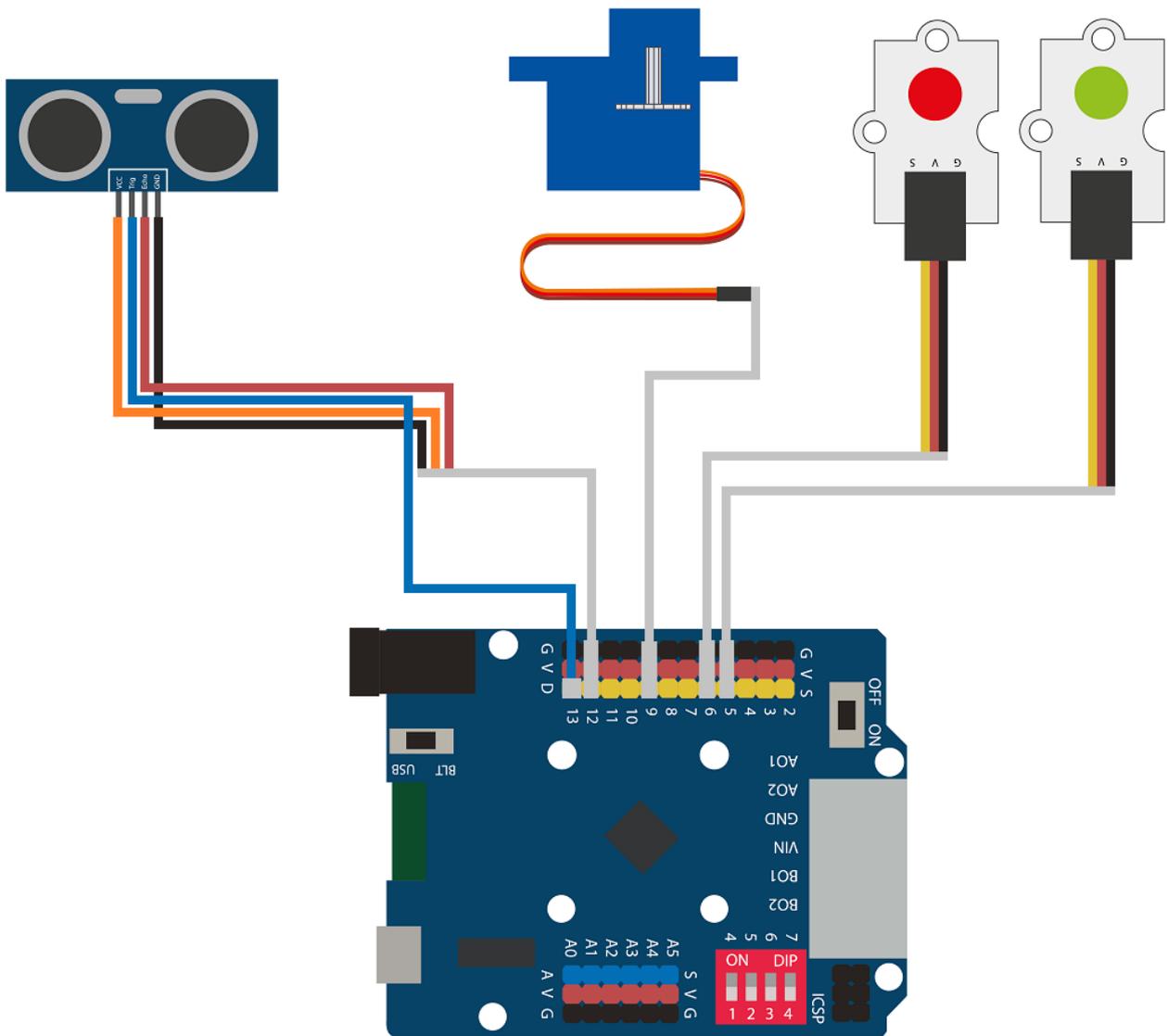
CONNECTIONS:

- Connect the green LED to digital port 5 on the 4in1 Build&Code board.

- Connect the red LED to digital port 6 on the 4in1 Build&Code board.
- Connect the servomotor to digital port 9 on the 4in1 Build&Code board.
- Connect the ultrasonic distance sensor to digital ports 12 and 13 on the 4in1 Build&Code board.

Digital Port 12	Ultrasonic distance sensor
V	VCC
G	GND
S / D	Echo
Digital Port 13	Ultrasonic distance sensor
S / D	Trig

Look at the colors of the cables and the colors of the terminals on the 4in1 Build&Code board to guide you. Each cable should be connected to its color:



PROGRAMMING CODE:

The program is comprised of the following parts:

1. Moving the servomotor 180° to the right and the left.
2. While the servomotor is moving, the distance is measured with the ultrasonic distance sensor:
 - If the distance measured is less than 20 cm, the red LED turns on.
 - If the distance measured is greater than 20 cm, the green LED turns on.

This sequence will continue to repeat as long as the setup has a power supply.

You can do this activity by using Arduino and Bitbloq software, as well as other compatible block programming software. Below you will find the programming code needed for each software.

Arduino Code

1. Download the Arduino software and go through the installation process.
2. Open the Arduino program, and once in, copy the following program:

```
#include <Servo.h>
Servo motor1; // motor1 = Name of the servomotor
int degreeM1; // Variables of the angles of the servomotor
int LEDRed = 6, LEDGreen = 5; // Red LED connected to digital port 5, green LED connected to digital port 6
int TrigPin = 13; // Connection ports of the ultrasonic sensor
int EchoPin = 12;
float SSound = 0.0343; // SPEED OF SOUND IN cm/us
long Lengh, Distance ; // VARIABLES TO CALCULATE THE DISTANCE IN CM

void setup() {
  // Put your setup code here, to run once:
  motor1.attach (9); // Servomotor connected to digital port 9
  pinMode(TrigPin, OUTPUT); // Configuration ports of the ultrasonic sensor
  pinMode(EchoPin, INPUT);
  pinMode(LEDRed, OUTPUT); // Configuration port of the LEDs
  pinMode(LEDGreen, OUTPUT);
}

void loop() {
  // Put your main code here, to run repeatedly:
  for (degreeM1 = 15; degreeM1 < 181; degreeM1 = degreeM1+5) // Add the variable degreeM1 +5 until it is larger than 181
  {
    DistanceCM(); // Calculate the distance in cm
    if (Distance < 20) // If the distance is less than 20 cm
    {
      digitalWrite (LEDRed, HIGH); // Red LED = ON
      digitalWrite (LEDGreen, LOW); // Green LED = OFF
    }
  }
}
```

```

else // If the distance is greater than 20 cm
{
    digitalWrite (LEDRed, LOW); // Red LED = OFF
    digitalWrite (LEDGreen, HIGH); // Green LED =
ON
}
    motor1.write(degreeM1); // Move the servomotor to
the angle of degreeM1
    delay(200); // Wait 200 ms
}
    for (degreeM1 = 180; degreeM1 > 15; degreeM1 =
degreeM1-5) // Reduce the variable degreeM1 -5 until
it is smaller than 15
    {
        DistanceCM(); // Calculate the distance in cm
        if (Distance < 20) // If the distance is less
than 20 cm
        {
            digitalWrite (LEDRed, HIGH); // Red LED = ON
            digitalWrite (LEDGreen, LOW); // Green LED =
OFF
        }
        else // If the distance is greater than 20 cm
        {
            digitalWrite (LEDRed, LOW); // Red LED = OFF
            digitalWrite (LEDGreen, HIGH); // Green LED =
ON
        }
        motor1.write(degreeM1); // Move the servomotor to
the angle of degreeM1
        delay(200); // Wait 200 ms
    }
}

void DistanceCM()// FUNCTION TO CALCULATE DISTANCE
{
    // DISTANCE CALCULATION IN cm
    digitalWrite(TrigPin, LOW); // We check that
the trigger is deactivated
    delayMicroseconds(4); // To make sure
that the trigger is LOW
    digitalWrite(TrigPin, HIGH); // We activate
the output pulse
    delayMicroseconds(14); // We wait 10

```

```
μs. The pulse is still active this time
    digitalWrite(TrigPin, LOW);          // We cut the
pulse and wait for ECHO
    Length = pulseIn(EchoPin, HIGH) ; //pulseIn measures
the time that elapses from when the declared pin
(echoPin) changes from low status to high (from 0 to
1)
    Distance = SSound* Length / 2;  // CALCULATION OF
THE DISTANCE
}
```

3. Configure and load the code, following the instructions provided in the document “How to add a library to program your RGB LED.”

Code for compatible block programming software

1. Download the software and go through the installation process.
2. Open the program, and once in, copy the following code:

```

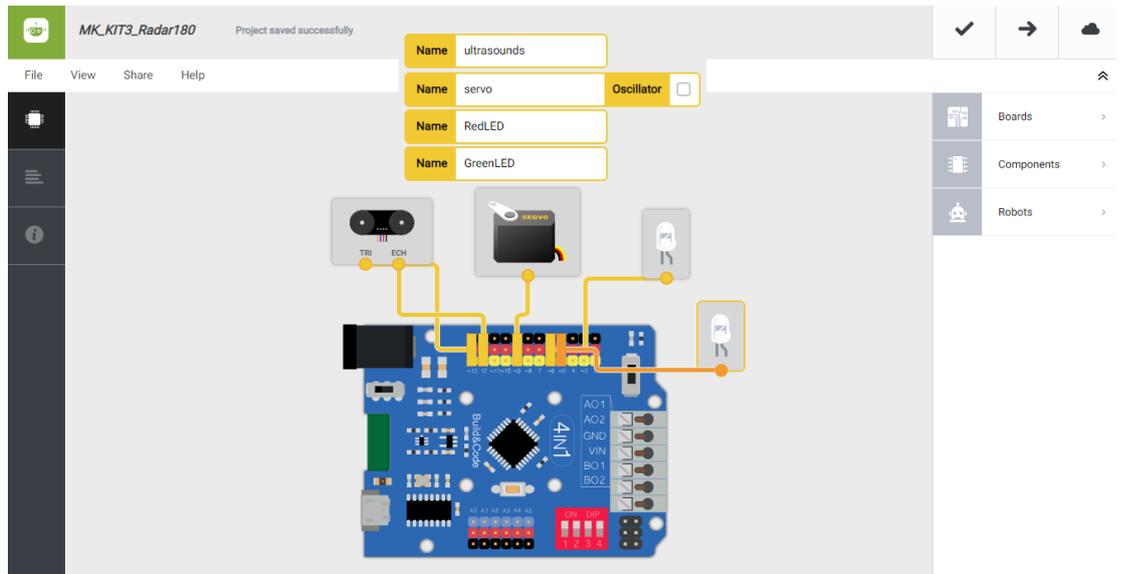
Arduino Program
set degree to 15
forever
  repeat until degree > 181
    set servo pin 9 angle as degree
    change degree by 5
    set distance to read ultrasonic sensor trig pin 13 echo pin 12
    if distance < 20 then
      set digital pin 6 output as HIGH
      set digital pin 5 output as LOW
    else
      set digital pin 6 output as LOW
      set digital pin 5 output as HIGH
    wait 0.2 secs
  repeat until degree < 15
    set servo pin 9 angle as degree
    change degree by -5
    set distance to read ultrasonic sensor trig pin 13 echo pin 12
    if distance < 20 then
      set digital pin 6 output as HIGH
      set digital pin 5 output as LOW
    else
      set digital pin 6 output as LOW
      set digital pin 5 output as HIGH
    wait 0.2 secs

```

3. Configure and load the code, following the instructions given in the document First Steps for the 4in1 Build&Code board.

BitBlox Code

1. Get the BitBlox software.
2. Open the BitBlox program, and once in, copy the following code:
 - **Hardware**



- **Software**



3. Configure and load the code, following the instructions given in the document First Steps for the 4in1 Build&Code board.

RESULT OF THE EXERCISE:

With the servomotor and the ultrasonic distance sensor, you have made the system to measure distance with a turning motion of 180°. The LEDs will serve as indicators: if an object has been detected, the red LED will turn on, and if no object has been detected, the green LED will turn on. You have created a radar with 180° of turning motion!