

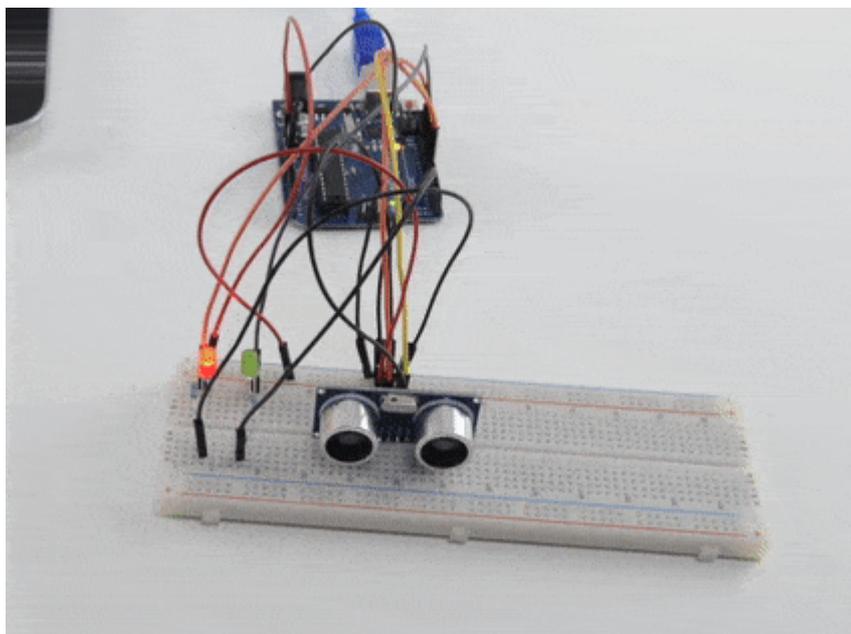
Build&Code Plus

Project 3

LED CONTROLLED BY DISTANCE SENSOR

PROJECT N° 3: Control an LED with a distance sensor

Build and code a circuit to control the color of an LED using a distance sensor. If an object is less than 20 cm close to the distance sensor, the LED will be red, but if the object is over 20 cm far from the distance sensor then the LED will be green.



DIFFICULTY LEVEL: Beginner.

DURATION OF THE EXERCISE: 30 min.

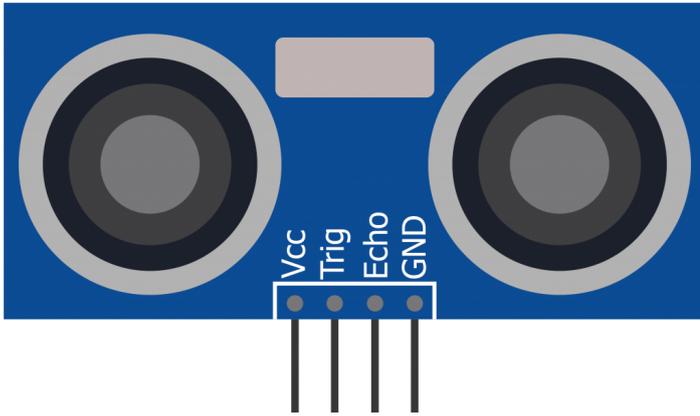
MATERIALS:

- 2 LEDs (1 green and 1 red)
- 2 Resistances 220
- 1 Ultrasonic sensor (distance sensor)
- 1 Build&Code UNO board
- 1 Protoboard

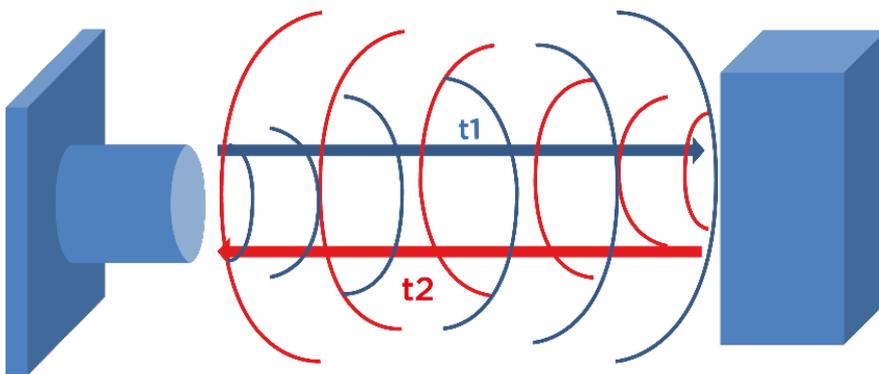
What is an ultrasonic distance sensor?

A distance sensor functions exactly as bat when it flies in the night without hitting any object. The bat sends ultrasonic waves while flying, and if this waves bounce back it means that there is an object near.

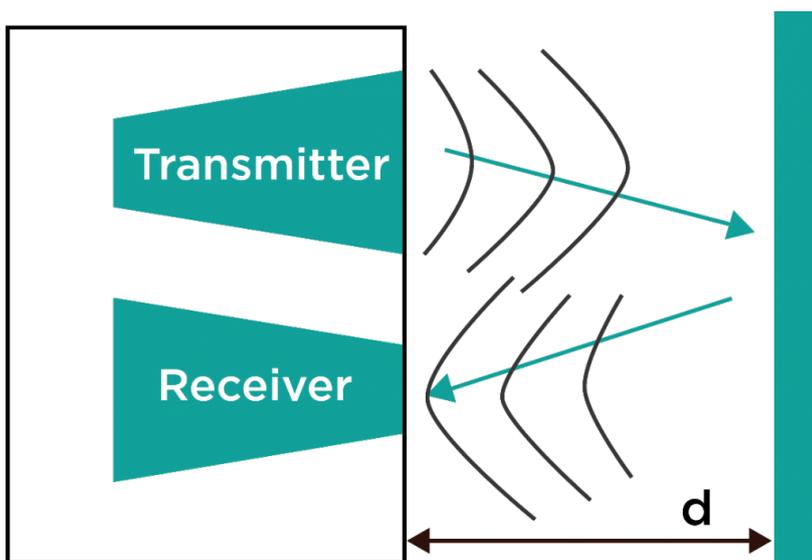
The ultrasonic sensor does the same thing. It sends ultrasonic waves and if they bounce back then it recognizes that there is an object near. The distance to the object is measured by the time it takes for the ultrasonic wave to bounce back.



The ultrasonic sensor is composed of two different modules: emitter and receiver. The emitter sends ultrasonic waves, which are sound waves that humans cannot hear (same waves that the bats send). These waves go forward until they hit an object. Then, they bounce back to the sensor, when the ultrasound is detected by the receiver.



$$T = t_1 + t_2$$



Because the sound waves always travel at the same speed when they move through the air: 343 m/s, it is easy to know the distance of the detected object.

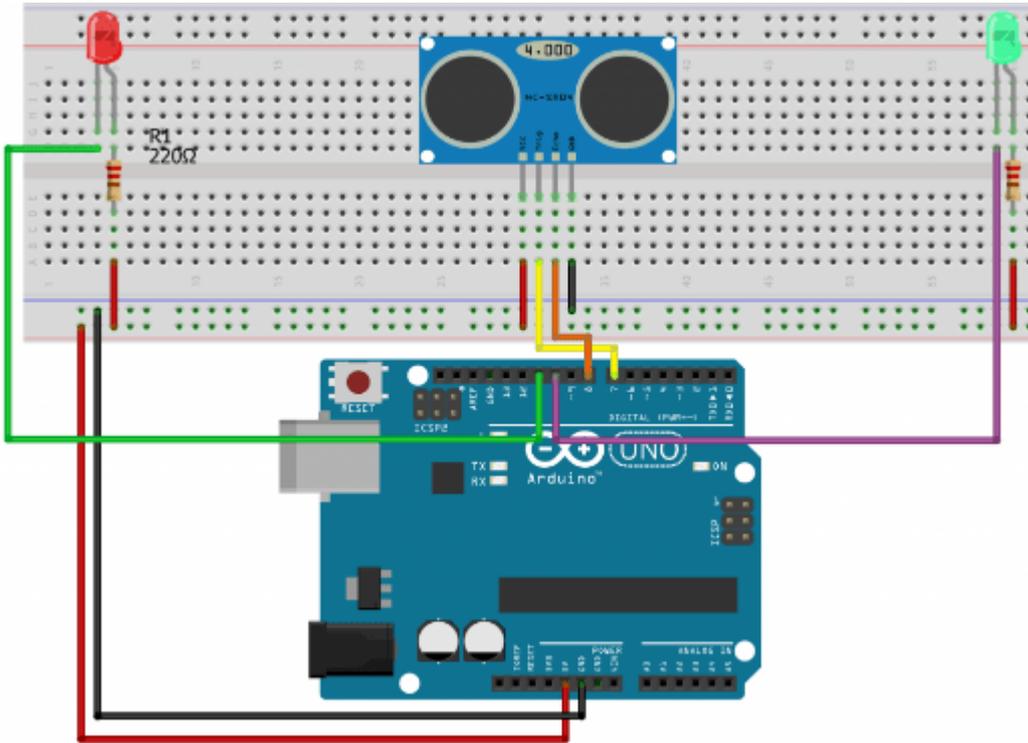
First, you must measure the time it takes the sound wave to go and come back to the sensor. Then, you divide this time by 2, so you know exactly how long it took the wave to hit the object. When you know the travel time (to hit the object), you have to multiply it by the speed the waves move through air (343 m/s). The result will be the distance where the object is.

$$\mathbf{Distance} = 343\text{m/s} * \text{time (seconds)}.$$

If the ultrasonic waves doesn't come back, it means that there is no object and the sensor will inform that nothing is in front of it.

CONNECTIONS:

1. The protoboard receives the electricity from the Build&Code UNO 5V Pin. And goes back to the Build&Code UNO through the GND Pin. All grounds of the circuit must be connected to each other so they have the same GND value. In the image, the GND is represented with a **black** cable, where all the components are connected to each other and to the GND board. The **red** cable represents the 5V, that supply electricity the ultrasonic distance sensor.
2. To activate the green and red LED you have to connect them to a digital pin. These pins will be the ones that send electricity to the LEDs. In the image you will see that the red LED is connected to the digital pin 11 with a **green** cable and the green LED is connected to the digital pin 10 with a **purple**.
3. The ultrasonic sensor has 4 pins. You have already connected 2 of them (5V and GND). The 2 other pins are ECHO and TRIG, which 2 digital pins, these pins are the ones that will send the information to the Build&Code UNO board. In the image you will see that the TRIG pin is connected to the digital pin 7 with a **yellow** cable, and the ECHO pin is connected to the digital pin 8 with an **orange** cable.



PROGRAMMING CODE

You can do this project using the Arduino program or a visual programming software by blocks compatible. Below you will find the necessary code.

Arduino Code

You will write a program that is constantly reading information from the ultrasonic sensor. According to the distance measured by the sensor the red or the green LED will turn on.

For this to happen you will use the next condition: If the distance is less than 20 cm, the red LED will be on and the green LED will be off; but if the distance is more than 20 cm, the green LED will be on and the red LED will be off.

To measure the distance you have to define TRIG Pin as an output and ECHO as an input. TRIG will send the ultrasonic waves forward and ECHO will be waiting for the waves to come back.

Then, each time a wave comes back, you have to measure the distance of the object. Remember that the speed of sound in the air is always: 343,2 m/s.

So if you know the total time of the wave to go and come back you will have to calculate the distance. For that, you have to divide by 2 the total time the wave

goes and comes back and then you multiply it by the speed of sound in the air:

$$\mathbf{Distance} = 343\text{m/s} * \text{time (seconds)}$$

To work with a better resolution, you will work with cm/us (centimeters / microseconds) using the next conversion:

$$\frac{343\text{m}}{\text{s}} * \frac{100\text{cm}}{1\text{m}} * \frac{1\text{s}}{1000000\text{us}} = \frac{0,0343\text{cm}}{\text{us}} = 0,0343\text{cm/us}$$

Once you have calculated the distance you will need to turn the red or the green light. For that to happen, you have to determine as output both pins where you have the connected the two LEDs. And when you want to turn on one of them you have to give it a HIGH value (1) and when you want it to be off you have to give it a LOW value (0).

1. Download and install the Arduino IDE program. It is available for Windows, Mac OS and Linux.
2. Open the Arduino program and copy the following program in it:

```
int trigPin = 7; //Define the pins that you will
work with
int echoPin = 8;
int LEDR = 10;
int LEDV = 11;
float Speed = 0.0343; // Sound speed at cm/us
long duration, distance;

void setup()
{
    pinMode(trigPin, OUTPUT); //Define digital pin 7
as an output

    pinMode(echoPin, INPUT); //Define digital pin 8
as an input

    pinMode(LEDR, OUTPUT); //Define digital pin 10
as an output

    pinMode(LEDV, OUTPUT); //Define digital pin 11
as an output
```

```

    digitalWrite (LEDR , LOW); // Define digital pin
10 in a low status

    digitalWrite (LEDV , LOW); /Define digital pin 11
in a low status
}
void loop()
{
    digitalWrite(trigPin, LOW);           // Make sure
that the TRIG is deactivated

    delayMicroseconds(2);                 // Make sure
that the TRIG is in LOW

    digitalWrite(trigPin, HIGH);         // Activate the
output pulse
    delayMicroseconds(10);                // Wait for
10µs, the pulse remains active during this time

    digitalWrite(trigPin, LOW);          //Stop the
pulse and wait for ECHO

    duration = pulseIn(echoPin, HIGH) ; // pulseIn
measures the time since the defined pin (echoPin)
changes its status from low to high (from 0 to 1)

    distance = Speed* duration / 2;      //Divide by 2
because we want to have only the "go" time, not the
"go and back" time

                                           // and divide
by 29,1 because 1 is divided by the sound speed
(1/SpeedSound) at cm/us

    if ( distance < 20){
        digitalWrite (LEDR , HIGH);     //If the
sensor detects a distances less than 20 cm the red
LED turns on

        digitalWrite (LEDV , LOW);      //and turns
off the green LED

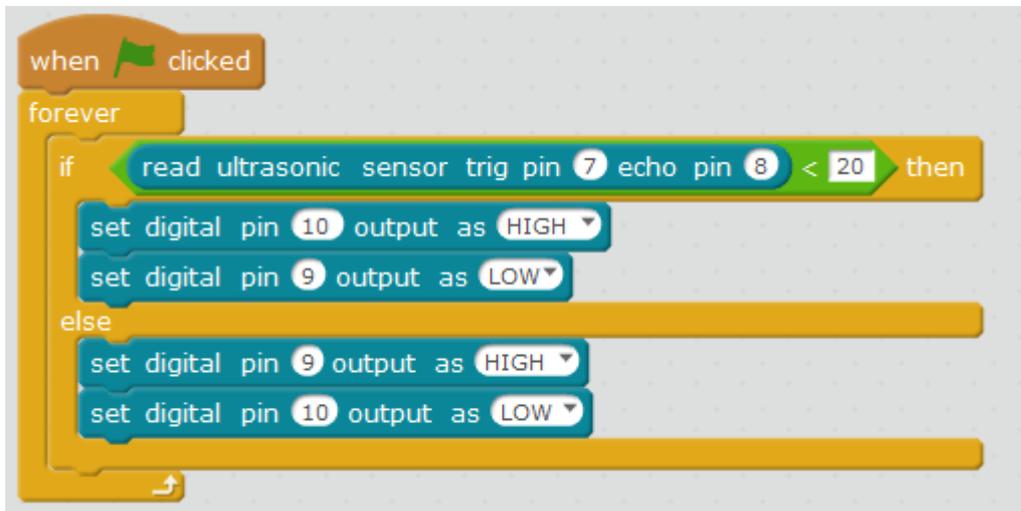
    }
    else{ // otherwise
        digitalWrite (LEDR , LOW);     // turn off the

```

```
red LED
    digitalWrite (LEDV , HIGH); //turn on the
green LED }
}
```

Code for the visual programming software by blocks compatible

1. Download and install the program.
 - 1.1 Open the software.
 - 1.2 Configure the program to save code into the Build&Code UNO board. You will find the instructions in the Arm Robot First Steps guide.
2. Open the program and copy the following code. Use the following image as a guide:



RESULT OF THE EXERCISE

When the sensor detects an object less than 20 cm away, the red LED will turn on and when the distance is higher than 20 cm, the green LED will turn on.