Build an Arduino Self-Driving Car

Lesson 4: Ultrasonic Distance Sensor

Estimated time: 45 minutes

Introduction

In this lesson students will learn to use an ultrasonic distance sensor to measure the distance to a nearby object. They will also test the sensor under "real world" conditions by testing objects with different sizes, shapes, and textures.

Learning Objectives

- Use an ultrasonic distance sensor with the Arduino
- Explore how the sensor works on different objects
- Program the Arduino to light up an LED when an object gets close to the sensor

Preparation

- Try this activity yourself before you try it with students. You can keep a working circuit (including the LED) assembled for a demonstration.
- Have an assortment of objects available for students to test with the sensors. Try to have objects with different shapes, sizes, and textures, since they will all reflect sound differently.

Materials

In addition to the materials from previous lessons, each group will need:

- HC-SR04 ultrasonic distance sensor
- Male-female jumper wires (4)
- Assorted male-male jumper wires
- LED
- 220Ω resistor

Procedure

Walk your students through the following steps or have them watch the video and follow along.

Step	Timestamp	Description
1	0:00	Demonstrate the working circuit for students.
2	0:31	Connect the sensor to the Arduino (see Appendix 4A).
3	3:09	Look at the "Timing Diagram" section of the sensor's datasheet. Explain how the sensor operates.
4	4:41	Read through the commented example code (Appendix 4A) and enter it in the Arduino IDE.
5	8:56	Upload the code to the Arduino and open the serial monitor (***Tools*** **>** ***Serial Monitor***). Wave your hand back and forth in front of the sensor to confirm that it works.
6	11:19	Test the sensor with objects of different shapes, sizes, and textures. Use a ruler to measure the actual distance from the sensor to the object. How does it compare to the distance shown in the serial monitor? Do some objects work better than others?
7	12:36	Add an LED to the circuit. Edit the code so the LED turns on when an object is closer than a certain distance to the sensor (see Appendix 4B, but let students try this on their own first).

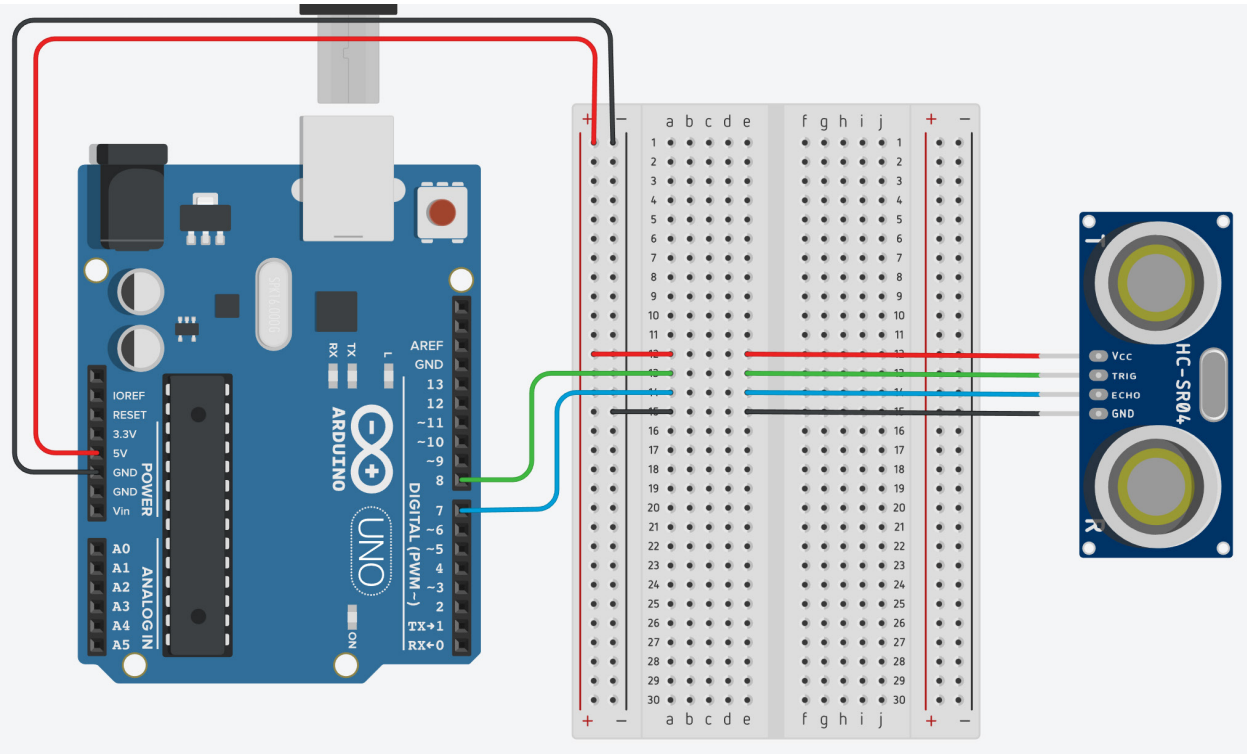
Assessment

As you walk around the class:

- Confirm that students can get the sensor and example code working.
- Observe how students test other objects in front of the sensor.
- Make sure students can alter the program to make an LED light up when an object is close enough to the sensor.

Appendix 4A: Breadboard Diagram and Code for HC-SR04 Sensor

The following diagram shows how to connect the ultrasonic sensor to the Arduino. For clarity, the H-bridge is omitted from this diagram, but you can connect the sensor to a breadboard with an existing H-bridge circuit. Make sure you power the sensor (the VCC pin) with 5V from the Arduino and *not* 6V from the 4xAA battery pack.



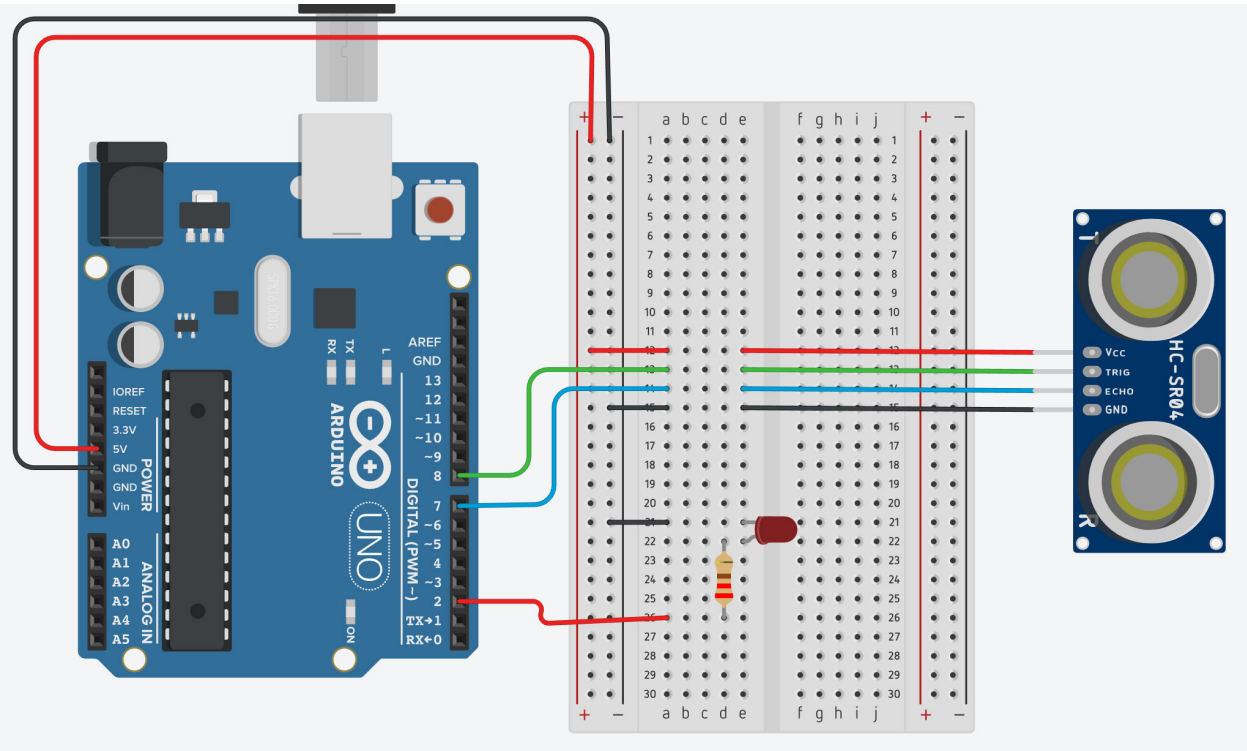
The following code will print the sensor reading to the serial monitor.

```
/*
  HC-SR04 demonstration code
  based on Ping example code available from
  File > Examples > 06.Sensors > Ping
  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping
*/

// define variables for sensor pins as constants because they
// won't change

const int triggerPin = 8;
const int echoPin = 7;

// define variables for duration of the ping and distance
long duration;  // time for echo to return in microseconds
long cm;        // distance to target in centimeters

void setup() { // code that only runs once
  Serial.begin(9600);           // initialize serial communication:
  pinMode(triggerPin,OUTPUT);   // set trigger pin as output
  pinMode(echoPin,INPUT);       // set echo pin as input
}
```

```
void loop() {

  // The HC-SR04 is triggered by a HIGH pulse of 10 microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  // use pulseIn command to measure the duration of the echo pulse
  // in microseconds
  duration = pulseIn(echoPin, HIGH);

  // convert the time into a distance in centimeters
  cm = duration/58;

  // print measurement to serial monitor
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);  // delay before next measurement
}
```

Appendix 4B: Breadboard Diagram and Code to Light Up LED

The following diagram shows an LED connected to the circuit with a 220Ω resistor.



The following code will turn on the LED when an object is detected closer to the sensor than the threshold distance.

```
/*
  HC-SR04 demonstration code
  based on Ping example code available from
  File > Examples > 06.Sensors > Ping
  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping
*/

// define constant variables that won't change
const int triggerPin = 8;
const int echoPin = 7;
const int ledPin = 2;
const int threshold = 15;

// define variables for duration of the ping and distance
long duration;  // time for echo to return in microseconds
long cm;        // distance to target in centimeters

void setup() { // code that only runs once
  Serial.begin(9600);           // initialize serial communication:
  pinMode(triggerPin,OUTPUT);   // set trigger pin as output
  pinMode(echoPin,INPUT);       // set echo pin as input
  pinMode(ledPin,OUTPUT);       // set LED pin as output
}
```

```
void loop() {
  // The HC-SR04 is triggered by a HIGH pulse of 10 microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  // use pulseIn command to measure the duration of the echo pulse
  // in microseconds
  duration = pulseIn(echoPin, HIGH);

  // convert the time into a distance in centimeters
  cm = duration/58;


  if(cm < threshold){ // if object is closer than threshold
    digitalWrite(ledPin,HIGH);  // turn the LED on
  }
  else{  // else, object is farther away than threshold
    digitalWrite(ledPin,LOW);  // turn the LED off
  }

  // print measurement to serial monitor
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);  // delay before next measurement
}
```