Lesson 20- Distance Sensors in Our City (Ultrasonic Sensor)

LESSON PLAN 20

LESSON 20 DISTANCE SENSORS IN OUR CITY
1.5 HOURS

LESSON FOCUS AND GOALS

- Explain how ultrasonic sensors are used in smart cities
- Identify inputs and outputs in a smart system
- Build and program an ultrasonic distance sensor
- Display live distance data using the Serial Monitor
- Design a smart city feature that uses distance data

MATERIALS NEEDED

- Arduino Uno (or compatible)
- Ultrasonic sensor (HC-SR04)
- LED or buzzer
- Breadboard
- 4 M-M Jumper wires

Doc with the codes

BACKGROUND

City Challenge Overview

Cities use sensors to make spaces safer, more efficient, and more accessible. In this lesson, you will design a Smart City sensor system using an ultrasonic distance sensor. Your system will measure how close an object is and use that information to control an output—just like real smart city technology. By the end of the lesson, you will have a working sensor that you can creatively place somewhere in your city.

Vocabulary

- Smart City: A city that uses technology to improve daily life
- Sensor (Input): Collects information from the environment
- Output: A response or action taken by a system
- Ultrasonic Sensor: Measures distance using sound waves
- Automation: A system that responds without human control
- Serial Monitor: A tool that displays data sent from the Arduino

Real-World Smart City Examples (5 minutes)

Explain to students:

“Cities use distance sensors to control traffic lights, manage parking, open doors, improve safety, and assist people with disabilities.”

Examples include:

- Parking sensors that show open spaces
- Automatic crosswalk signals
- Smart trash cans that detect when they are full
- Train platform safety systems

Ask:

- Where in your city would knowing distance be useful?

BACKGROUND CONTINUED

The Ultrasonic sensor module HC-SR04 provides 2cm-400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The sensor works by sending an ultrasonic sound that we can not hear. It waits for the sound to bounce back off any object in front of it then it will record the time that it takes for the sound to return(echo). Then it will convert that into distance using the speed of sound and divide that answer by 2. Why does it divide by two? because the sound travels to the object then back and we only want the distance one way.
By doing this the ultrasonic sensor can measure the distance between the sensor and any object in front of it but only up from 2cm to 400cm away.
Formula used:
Test distance = (high level time × velocity of sound (340m/s) /2

How the Ultrasonic Sensor Works (5–7 minutes)

- The sensor sends out a sound wave.
- The sound hits an object and bounces back.
- The sensor measures how long it takes to return.
- The Arduino calculates the distance.

Check for Understanding:

- Is the ultrasonic sensor an input or an output?
- Answer: Input

How the Sensor Fits into a Smart City (5–7 minutes)

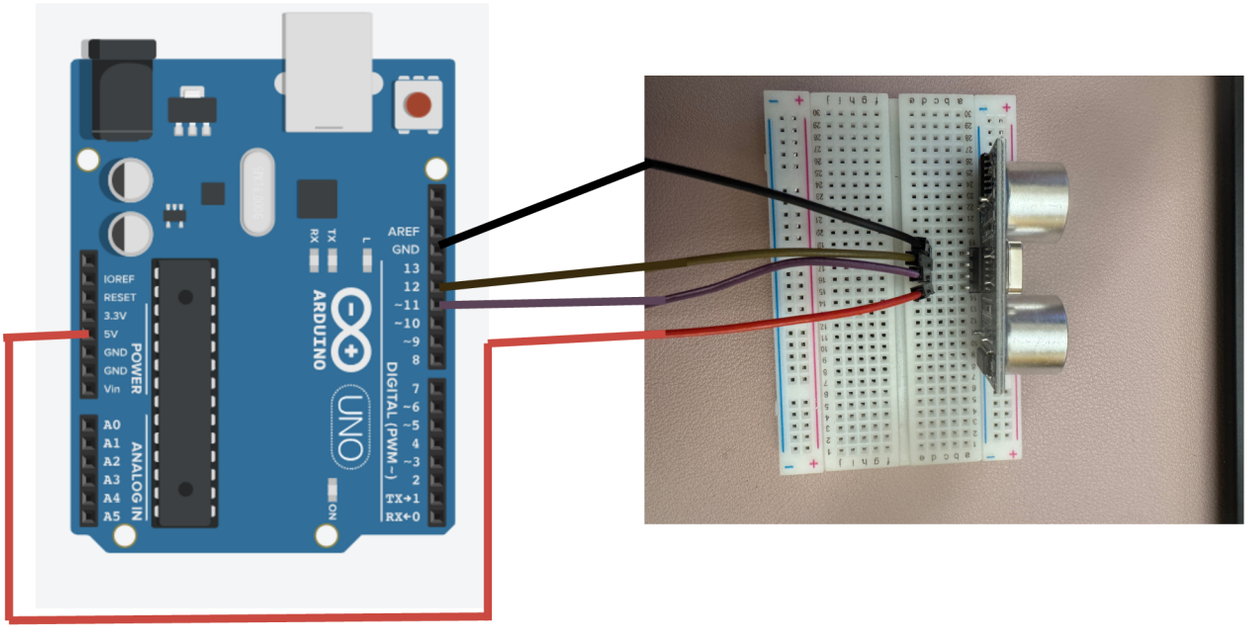
- Explain:
- The ultrasonic sensor sends out sound waves.
- The waves bounce off objects like cars or people.
- The Arduino calculates how far away the object is.
- The city system decides what action to take.

Smart City Thinking Question:

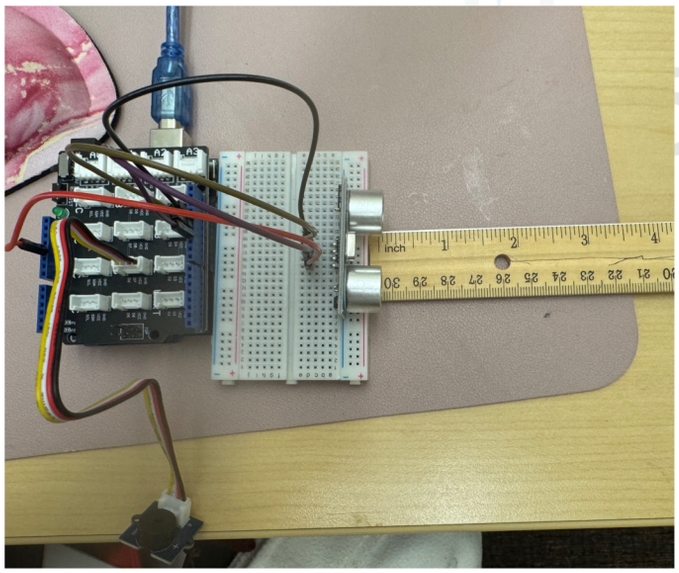
- What might happen in a city if something gets too close?

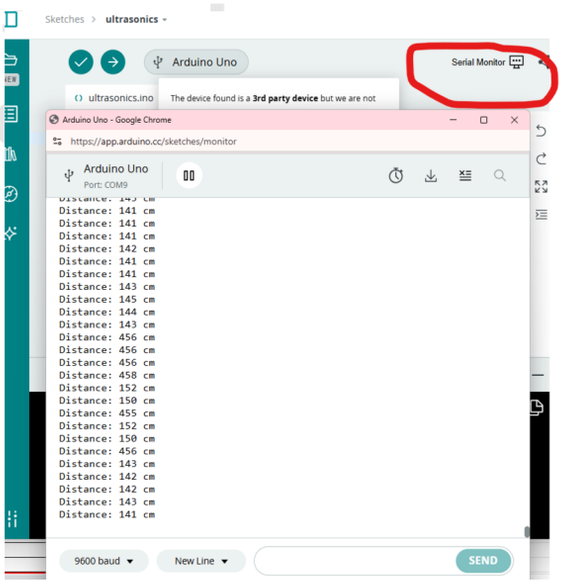
ACTIVITY 1

Step 1: Connect the base shield to the arduino. Note: this photo shows it without the base shield. You need the base shield.
Step 2: Using four jumper wires attach the ultrasonic sensor to the Arduino base shield. Be careful to make sure the wires are in the correct slots since the base shield is not labeled. It is helpful to count the slots. See wiring diagram on next page but note that the base shield is not attached.



Step 3: Copy and paste the code for this lesson to the arduino IDE and upload it to the Arduino.
Step 4. Open the serial monitor located in the top right corner of the arduino app.
Step 5: After you have opened the serial monitor, take ruler and line it up in front of the sensor. Next, place an object or you hand in front of the sensor at different measurements and compare it to the serial monitor readings
Step 6: Plug in the buzzer into pin D6. Place you hand at different distances on the ruler and see what happens with the buzzer.





```
const int trigPin = 11;
const int echoPin = 12;
const int outputPin = 6;

long duration;
int distance;

void setup()
 {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(outputPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance < 20)
 {
    digitalWrite(outputPin, HIGH);
  }
 else
 {
    digitalWrite(outputPin, LOW);
  }
  delay(200);
}
```

CODING CHALLENGE

The buzzer is being triggered when your hand is placed at a certain distance away from the sensor. What is that distance(s)? How can we change the distance if we needed to?Have students examine the code and look for where in the code to change the distance that triggers the buzzer.

DESIGN CHALLENGE

Student Task

Decide where this sensor will live in your city model.

Choose one:

- Smart parking space
- Crosswalk safety alert
- Automatic building door
- Train or bus platform warning
- Trash can fullness detector

Students must:

- Name their system
- Describe what it senses
- Explain what the output does
- Justify why it improves the city

Encourage students to name their system (e.g., “SafeWalk Sensor,” “SmartPark Alert”). This reinforces ownership and real-world engineering thinking. Have students try to combine more sensors or outputs.

LESSON SUMMARY

In this lesson, students explore how distance sensors are used in Smart Cities to make urban spaces safer, more efficient, and more responsive. Using an ultrasonic distance sensor, students collect real-time measurements and view the data in the Arduino Serial Monitor, just like city engineers monitoring traffic, parking, and public safety systems.

Students then connect the sensor to an output device (such as an LED, buzzer, or motor) to simulate a Smart City feature—like an automatic door, traffic alert, or pedestrian safety system—that responds based on distance. Through hands-on building and testing, students learn how sensors, data, and outputs work together to create intelligent city systems.

By the end of the lesson, students understand how sensing, data visualization, and automated responses are essential components of Smart City technology and how engineers use these tools to solve real-world problems.