Practical 8

Aim: - To interface **UltraSonic Distance Sensor** with Arduino.

Components Required :

Arduino Board, Bread Board, Ultrasonic Distance Sensor, Resistors, Connecting wires.

Theory:

The Ultrasonic sensor is used to measure the distance of the object using SONAR.

It emits the Ultrasound at a frequency of **40KHZ or 40000 Hz**. The frequency travels through the air and strikes the object on its path. The rays bounce back from the object and reach back to the module.

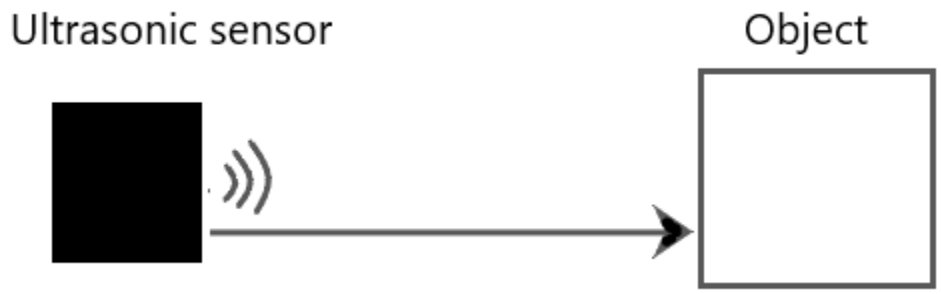
The four terminals of sensor are VCC, TRIG, ECHO, and GND. The voltage supply or VCC is +5V. We can connect the ECHO and TRIG terminal to any of the digital I/O pin on the specific Arduino board.

The Ultrasonic sensors work best for medium ranges.

How does Ultrasonic sensor work?

Let's understand how the sensor works.

- It sends ultra-high frequency samples.

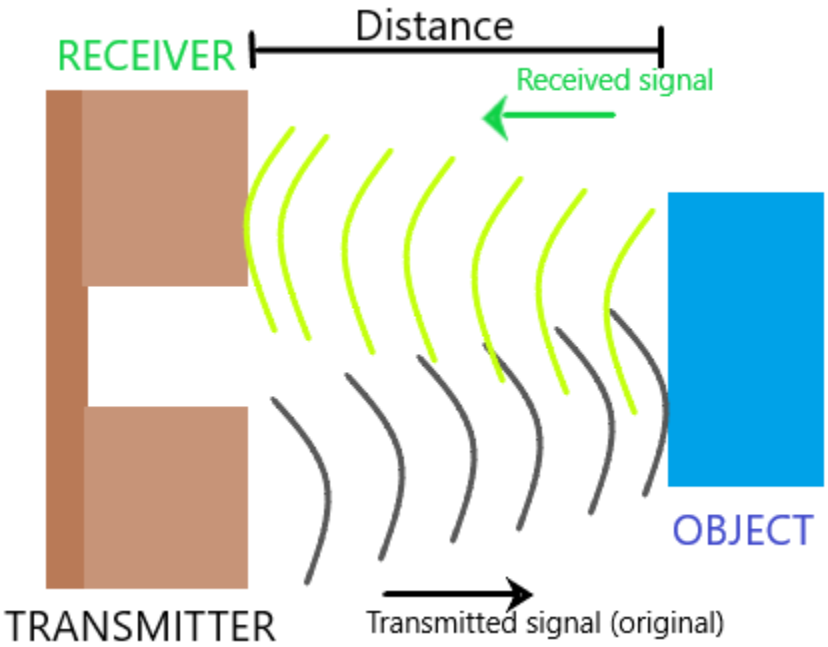


- When samples strike the object, it bounces back from the object.

- The distance sensor reports the time it takes between the sending and receiving of the samples.

Principle

We need to first set the **TRIG** (triggered) pin at **HIGH**. It will send out the burst of 8 cycles called the sonic burst, which will travel at the sound speed. It will be further received by the ECHO pin. The time traveled by the sound wave is considered the ECHO pin's output time in microseconds.



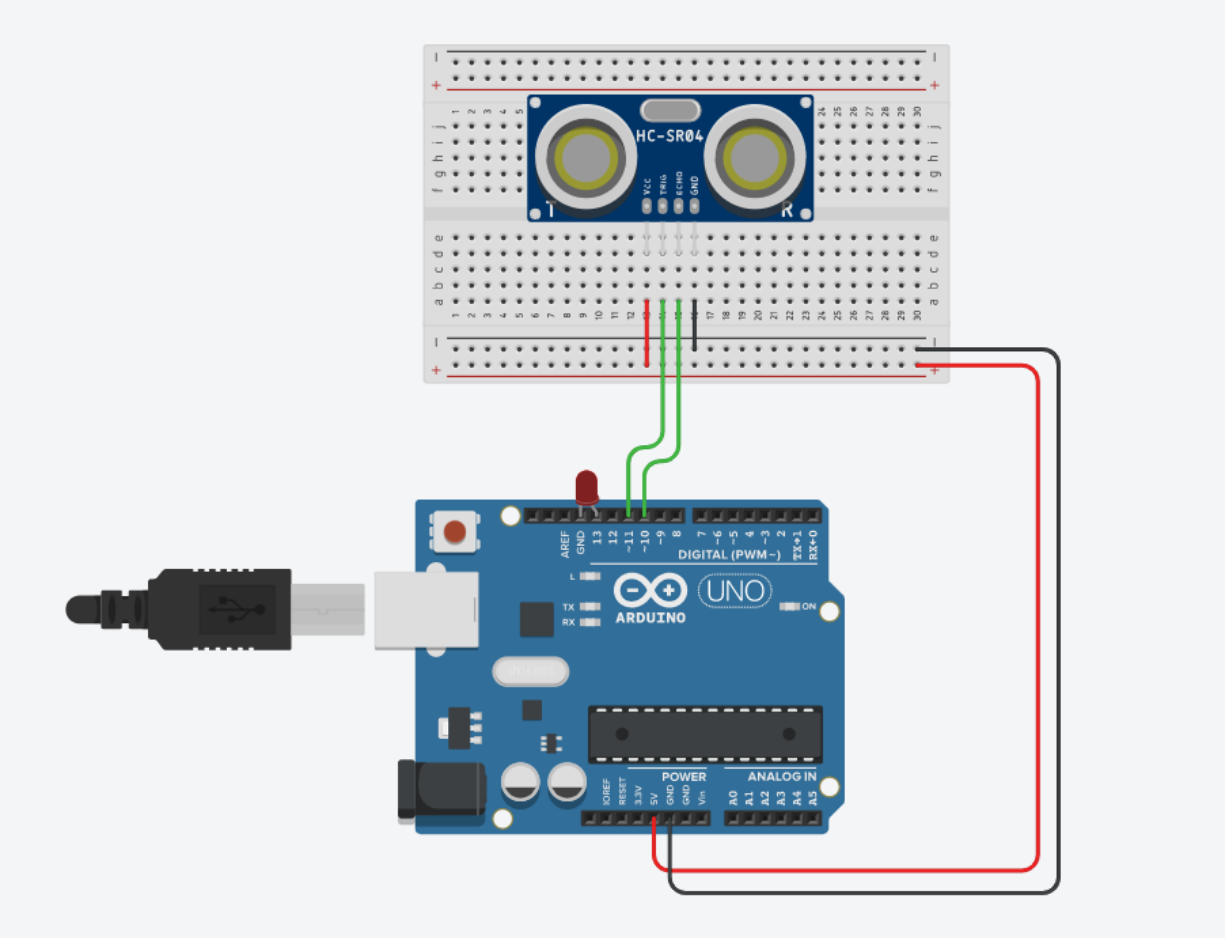
We will use the PulseIn() function to read the time from the output of the ECHO pin. It will wait for the specified pin to go HIGH and LOW. The function would return the timing at the end.

Implementation :

The steps to set up the connection are listed below:

- Connect the VCC pin of sensor to 5V of the Arduino board.
- Connect the GND pin of sensor to GND of the Arduino board.
- Connect the TRIG pin of sensor to pin 6 of the Arduino board.
- Connect the ECHO pin of sensor to pin 5 of the Arduino board.

Circuit Diagram:



Code:

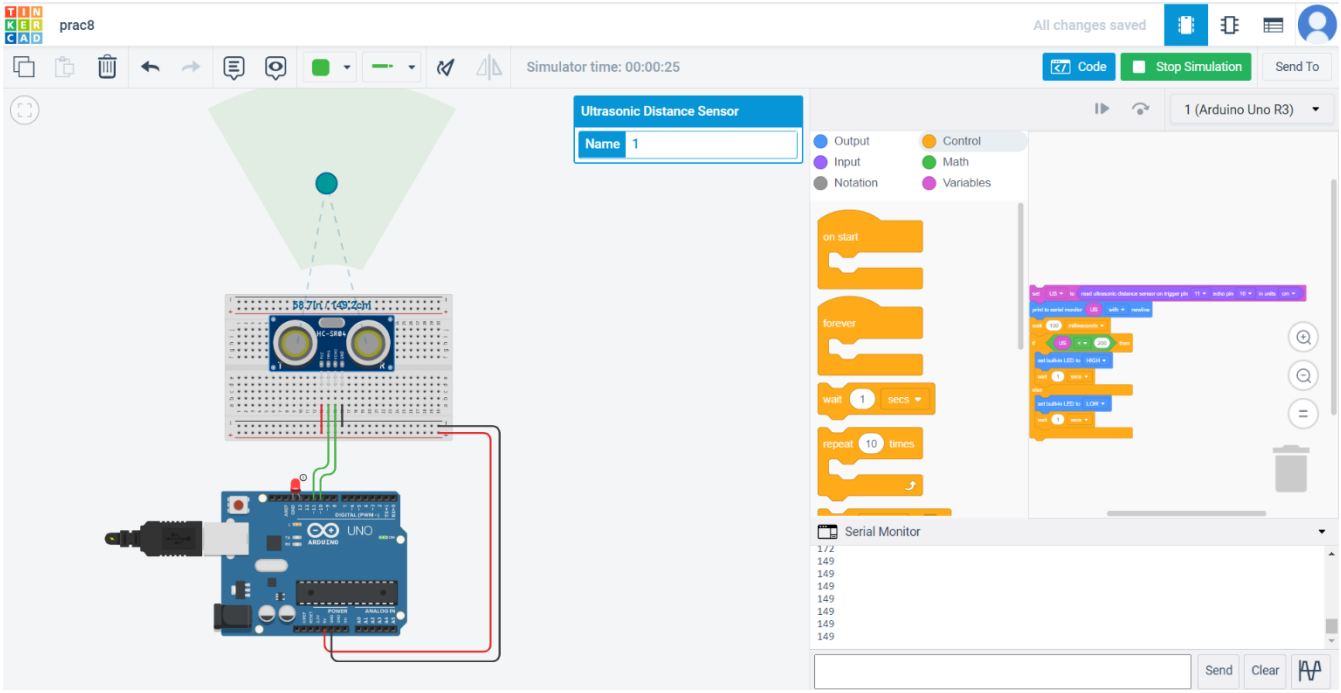
```
// C++ code
//
int US = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  US = 0.01723 * readUltrasonicDistance(11, 10);
  Serial.println(US);
  delay(100); // Wait for 100 millisecond(s)
  if (US < 200) {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
  } else {
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  }
}
```

Output :



Conclusion: Thus we studied the interfacing of Light Dependent resistor and how the resistance changes depending on the light.