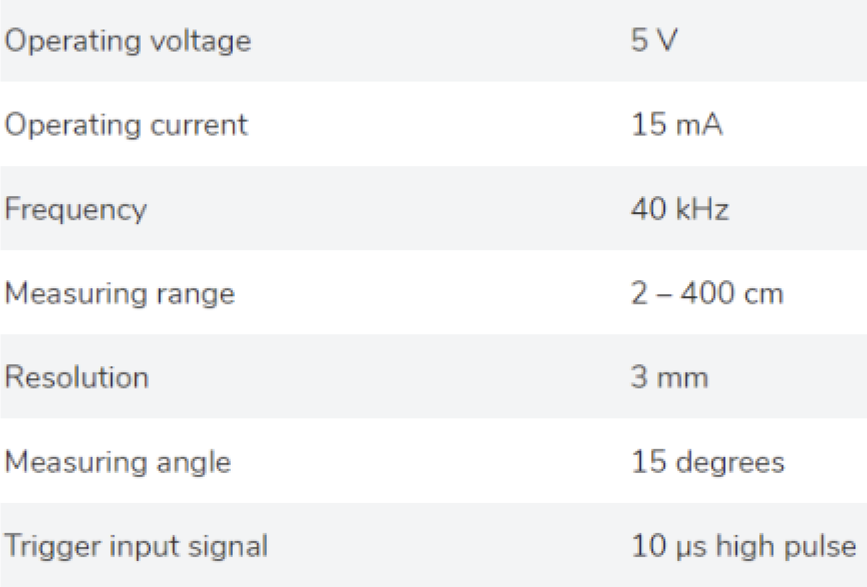
Interfacing an Ultrasonic Sensor with Arduino

Circuit Components

1. **Arduino Uno R3** – A microcontroller board that processes data and controls components. The #Arduino is a micro-controller board that is based on the ATmega328P, it consists of digital and analog input/output pins that are interfaced with the ultrasonic sensor (used to send and receive data). the Arduino Uno IDE consists of a Serial Monitor that displays the output.
2. **Ultrasonic Sensor (HC-SR04)** – Measures distance using ultrasonic waves. Is the Ultrasonic Sensor used. An ultrasonic sensor contains two circular structures: one is the #transmitter and the other is a #receiver. The transmitter transmits the ultrasonic sound while the receiver receives the reflected signal. It has:
    - **VCC** – Connected to Arduino's **5V** for power.
    - **Trig (Trigger Pin)** – Connected to **Pin 10**, sends ultrasonic pulses.
    - **Echo Pin** – Connected to **Pin 9**, receives the reflected signal.
    - **GND** – Connected to **Ground (GND)** of the Arduino.
    - 

Operating voltage	5 V
Operating current	15 mA
Frequency	40 kHz
Measuring range	2 – 400 cm
Resolution	3 mm
Measuring angle	15 degrees
Trigger input signal	10 µs high pulse

Working Principle of Ultrasonic Sensor

The **ultrasonic sensor** has two **cylindrical transducers**:

- **Transmitter (T)** – Emits ultrasonic waves.
- **Receiver (R)** – Detects the reflected waves.

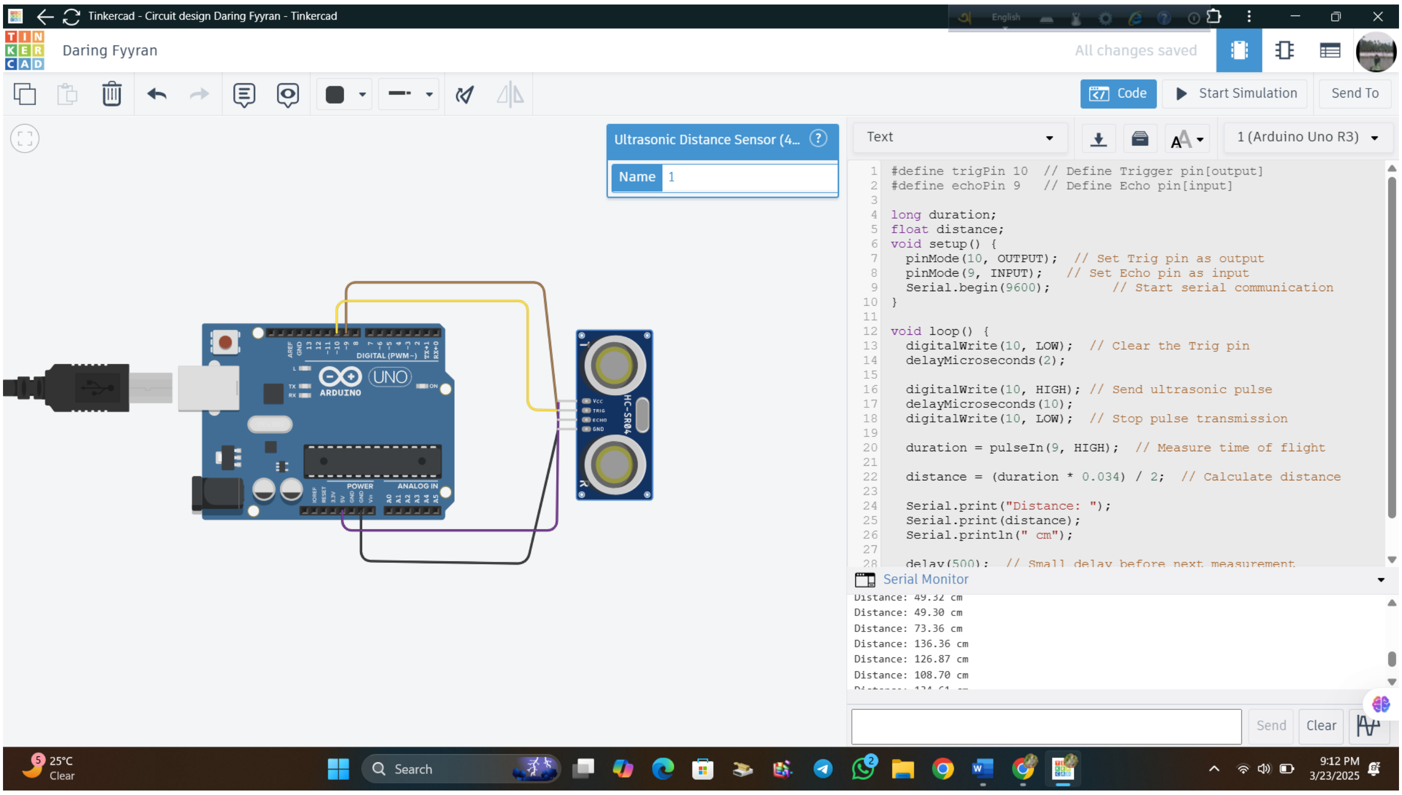
How It Works:

1. The sensor emits an **8-cycle ultrasound burst at 40kHz**.
2. The wave **bounces off an object** and returns to the receiver.
3. The **Echo pin** measures the **time taken (T)** for the wave to return.
4. Using the **speed of sound (C = 0.034 cm/µs)**, distance is calculated.

Distance Calculation:

- **General Formula:** Distance=Time×Speed of Sound
- Since the wave travels **to and from** the object, the actual distance (D) is: D=T×C/2
- This gives the **object's proximity** to the sensor.

Circuit:



```
#define trigPin 10  // Define Trigger pin[output]
#define echoPin 9   // Define Echo pin[input]

long duration;
float distance;
void setup() {
  pinMode(10, OUTPUT);  // Set Trig pin as output
  pinMode(9, INPUT);   // Set Echo pin as input
  Serial.begin(9600);         // Start serial communication
}

void loop() {
  digitalWrite(10, LOW);  // Clear the Trig pin
  delayMicroseconds(2);

  digitalWrite(10, HIGH); // Send ultrasonic pulse
  delayMicroseconds(10);
  digitalWrite(10, LOW);  // Stop pulse transmission

  duration = pulseIn(9, HIGH);  // Measure time of flight

  distance = (duration * 0.034) / 2;  // Calculate distance

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

  delay(500);  // Small delay before next measurement
```

```
Distance: 49.32 cm
Distance: 49.30 cm
Distance: 73.36 cm
Distance: 136.36 cm
Distance: 126.87 cm
Distance: 108.70 cm
Distance: 124.61 cm
```

CODE:

```
#define trigPin 10  // Define Trigger pin

#define echoPin 9   // Define Echo pin

long duration;

float distance;

void setup() {

  pinMode(10, OUTPUT);  // Set Trig pin as output

  pinMode(9, INPUT);   // Set Echo pin as input

  Serial.begin(9600);        // Start serial communication}

void loop() {

  digitalWrite(10, LOW);  // Clear the Trig pin

  delayMicroseconds(2);

  digitalWrite(10, HIGH); // Send ultrasonic pulse

  delayMicroseconds(10);

  digitalWrite(10, LOW);  // Stop pulse transmission
  duration = pulseIn(9, HIGH);  // Measure time of flight
  distance = (duration * 0.034) / 2;  // Calculate distance
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);  // Small delay before next measurement}
```

Code Explanation

1. **Variable Declaration**
   - trigPin = 10, echoPin = 9 (Assign pin numbers).
   - time and distance variables are declared for measurement.
2. **Setup Function (void setup())**
   - pinMode(trigPin, OUTPUT); → Set trigger pin as output.
   - pinMode(echoPin, INPUT); → Set echo pin as input.
   - Serial.begin(9600); → Begin serial communication.
3. **Main Loop (void loop())**
   - **Trigger Signal:**
     - Set trigPin **LOW** (clear signal).
     - Wait **2µs**, then set **HIGH** for **10µs** to send ultrasonic pulse.
     - Set trigPin **LOW** to stop signal.
   - **Time Calculation:**
     - Measure time taken for sound wave to return using pulseIn(echoPin, HIGH).
   - **Distance Calculation:**
     - Formula: distance = (time * 0.034) / 2
     - 0.034 is the speed of sound in air (cm/µs).
     - Divide by 2 since the signal travels to the object and back.
   - **Display Distance:**
     - Serial.println(distance); prints the result in **Serial Monitor**.

Simulation & Output

- The **Serial Monitor** displays distance values.
- Moving an object **closer** → distance **decreases**.
- Moving an object **farther** → distance **increases**.