Experiment 1: Distance Measurement Using Ultrasonic Sensor

* Aim:

To measure the distance of an object using an HC-SR04 ultrasonic sensor connected to an Arduino and trigger an LED when the object is detected within a certain range.

* Requirements:

    * Arduino UNO
    * Ultrasonic Sensor (HC-SR04)
    * Breadboard
    * Jumper wires
    * LED
    * 220Ω resistor
    * USB cable and Arduino IDE

* Theory:

The HC-SR04 ultrasonic sensor works by emitting ultrasonic waves (sound waves with frequency above human hearing range) to detect the distance of objects. It has two main parts: a transmitter (Trigger) and a receiver (Echo). When the Trigger pin is activated with a HIGH signal for 10 microseconds, the sensor sends out an ultrasonic wave at 40 kHz. If the wave hits an object, it reflects back to the Echo pin.

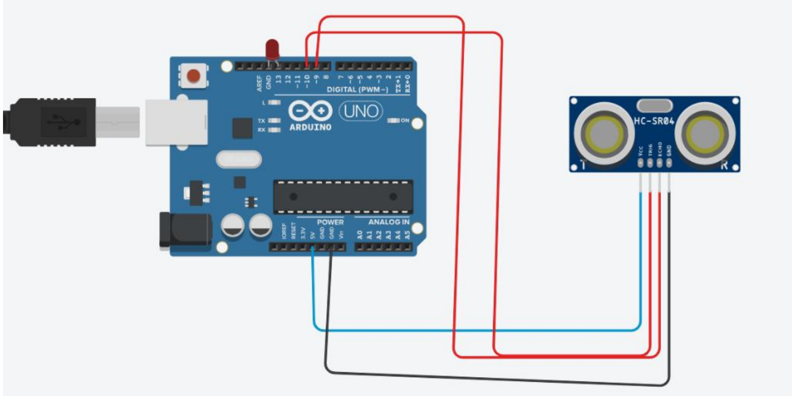
The Arduino measures the time it takes for the wave to return. Since the speed of sound in air is approximately 343 meters per second (or 0.0343 cm/µs), the distance can be calculated using the formula:

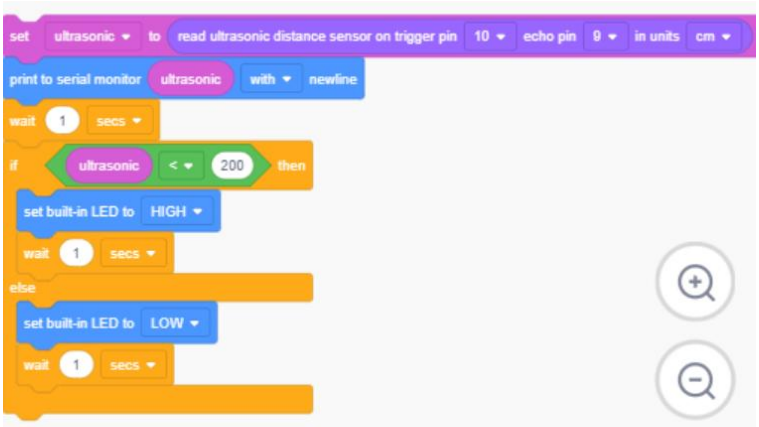
Distance (cm) = (Time × 0.0343) / 2

The division by 2 accounts for the wave traveling to the object and then back to the sensor. The sensor can detect objects ranging from 2 cm to 400 cm with good accuracy.

* Connections:

    * VCC → 5V on Arduino
    * GND → GND
    * Trig → Pin 9
    * Echo → Pin 10
    * LED Anode (long leg) → Pin 13 (through 220Ω resistor)
    * LED Cathode → GND





```
set ultrasonic to read ultrasonic distance sensor on trigger pin 10 echo pin 9 in units cm
print to serial monitor ultrasonic with newline
wait 1 secs
if ultrasonic < 200 then
  set built-in LED to HIGH
  wait 1 secs
else
  set built-in LED to LOW
  wait 1 secs
```

* Procedure:

1. Assemble the circuit as per the connections mentioned.
2. Write/upload a program that:

   * Sends a 10µs pulse to the Trig pin
   * Measures the pulse duration from the Echo pin
   * Calculates the distance
   * Turns ON the LED if the object is within a certain distance (e.g., <10 cm)
3. Open the Serial Monitor to view distance readings.
4. Test by placing your hand or an object in front of the sensor and observe LED behavior.

* Result:

The ultrasonic sensor successfully detected objects and turned the LED ON when the object was within the threshold distance.

Experiment 2: Motion Detection Using PIR Sensor

* Aim:

To detect human motion using a PIR sensor and display the detection result on the Serial Monitor.

* Requirements:

* Arduino UNO
* PIR Motion Sensor
* Jumper wires
* Arduino IDE with Serial Monitor

* Theory:

A PIR (Passive Infrared) sensor detects changes in infrared radiation emitted by objects in its field of view. Humans and animals emit infrared radiation due to body heat. When a person moves in front of the sensor, the variation in infrared radiation is detected and converted to a digital signal.

The PIR sensor has three pins:

* Power (VCC)
* Ground (GND)
* Signal (OUT)

When motion is detected, the output pin sends a HIGH signal; otherwise, it stays LOW.

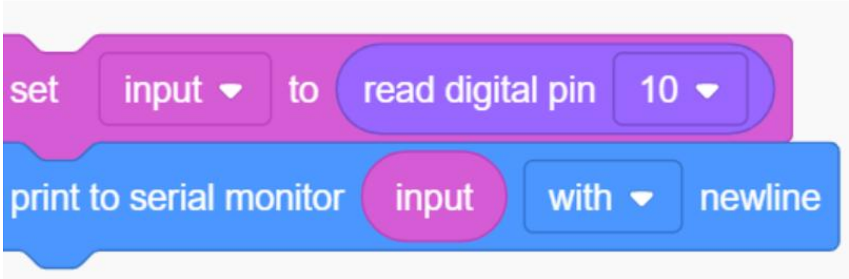
* Connections:

  * VCC → 5V
  * GND → GND
  * OUT (Signal) → Digital Pin 10

* Procedure:

  1. Connect the PIR sensor to the Arduino as described.
  2. Open the Arduino IDE, create a variable to hold input.
  3. Use blocks/code:

     * Input: Read digital pin 10 and store value in a variable.
     * Output: Use Serial.print() to print the value or a message.
  4. Upload the code and open Serial Monitor.
  5. Move in front of the sensor and observe the messages printed.



```
set input ▾ to read digital pin 10 ▾
print to serial monitor input with ▾ newline
```

* Result:
  The Serial Monitor showed "Motion Detected" whenever movement was detected, confirming proper working of the sensor.

Experiment 3: Temperature Measurement Using LM35 Sensor

* Aim:
  To measure ambient temperature using a temperature sensor and display it using Arduino.

* Requirements:

  * Arduino UNO
  * Jumper wires
  * Arduino IDE

* Theory:
  The LM35 temperature sensor provides an analog voltage that varies linearly with temperature. It outputs 10 millivolts per degree Celsius (e.g., at 25°C, the output is 250mV). This analog voltage is read using Arduino’s analog pin and converted to temperature.

  Arduino's analog input reads values between 0 and 1023 (10-bit resolution). The voltage is calculated as:

  Voltage = (analogReading × 5.0) / 1023

  Temperature = Voltage × 100

  (Because 10mV = 0.01V for every 1°C, so we multiply by 100)

* Connections:

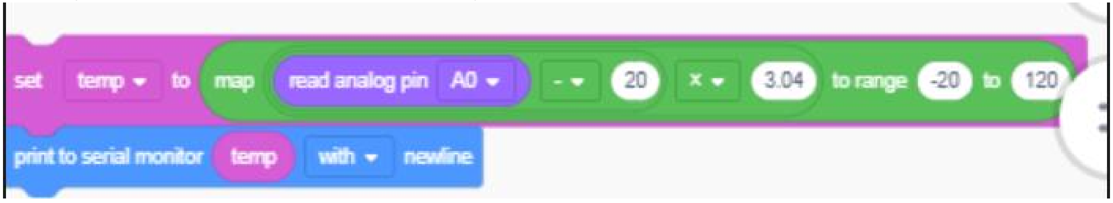
* VCC (left pin) → 5V
* Output (middle pin) → A0
* GND (right pin) → GND

* Procedure:

1. Place the sensor on the breadboard and make connections.
2. Write/upload the code that:

* Reads analog value from A0
* Converts it to voltage
* Converts voltage to temperature
* Prints result to Serial Monitor

3. Open Serial Monitor and view live temperature data.



```
set temp ▾ to map read analog pin A0 ▾ - ▾ 20 × ▾ 3.04 to range -20 to 120
print to serial monitor temp with ▾ newline
```

* Result:
The current temperature was successfully read and displayed in the Serial Monitor.

Experiment 4: Smoke Detection Using Gas Sensor (MQ-2)

* Aim:
To detect smoke/gas presence using MQ-2 sensor and alert via an LED connected to Arduino.

* Requirements:

* Arduino UNO
* MQ-2 Gas Sensor
* Breadboard (small)
* Jumper wires
* LED
* 220Ω resistor
* 10kΩ resistor

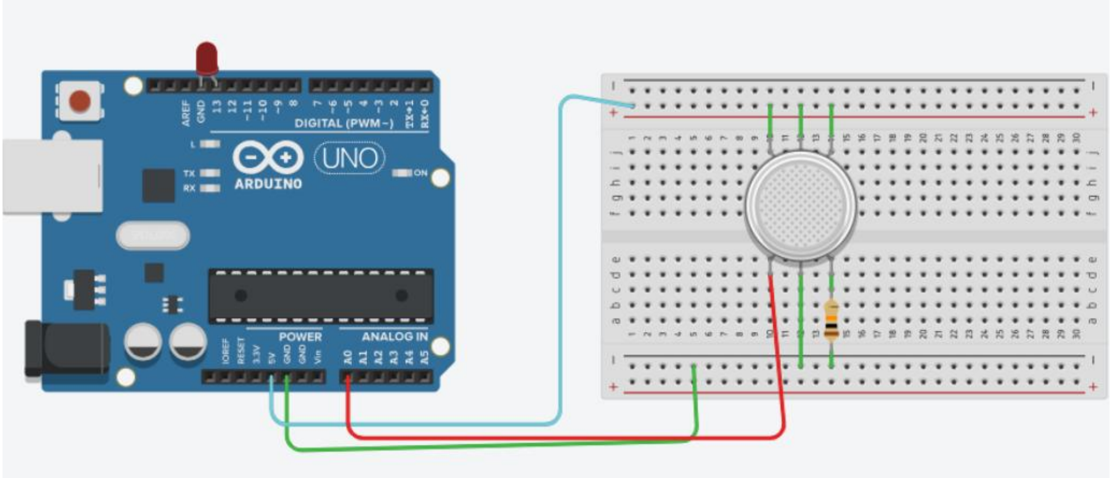
* Theory:
The MQ-2 gas sensor is used to detect gases like methane, propane, LPG, and smoke. It contains a heating element and a sensing layer. When exposed to a gas, the sensor's resistance changes, altering the voltage at the analog output. This analog value is read by Arduino.

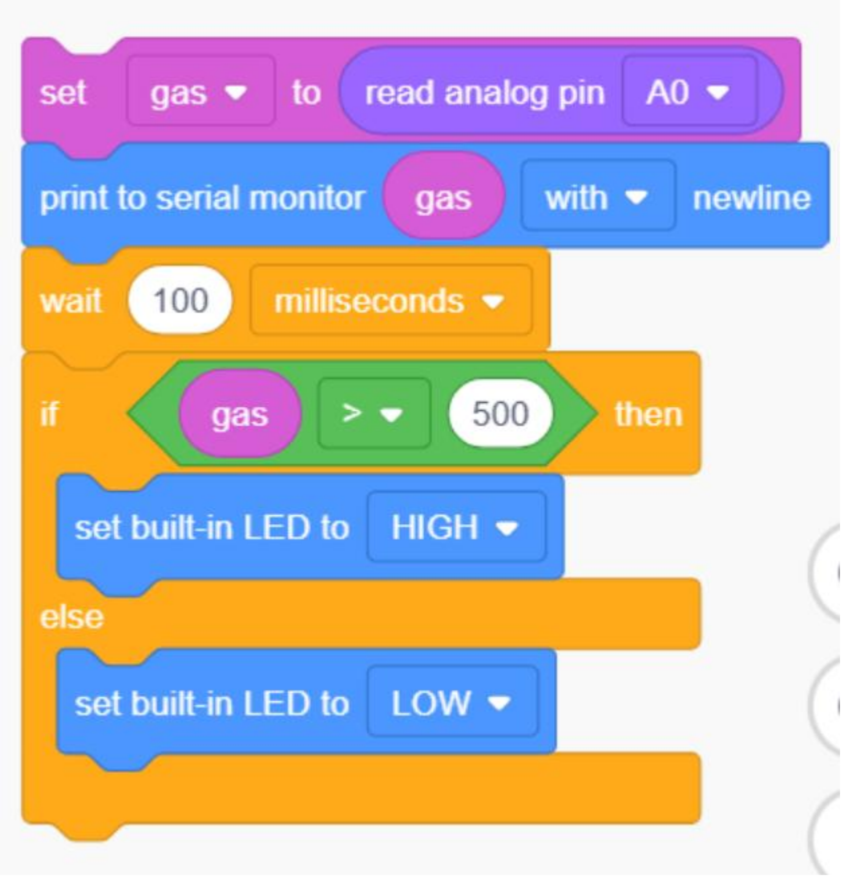
If the gas level crosses a certain threshold, a digital decision can be made — for instance, turning on a LED for visual alert.

* Connections:

* A1, H1, A2 → Connect together and then to 5V (Power)
* H2 → GND
* B1 → A0 (Analog signal)
* B2 → 10kΩ resistor → GND
* LED:

* Anode (long leg) → Pin 13 (through 220Ω resistor)
* Cathode → GND





```
set gas ▾ to read analog pin A0 ▾
print to serial monitor gas with ▾ newline
wait 100 milliseconds ▾
if gas > ▾ 500 then
    set built-in LED to HIGH ▾
else
    set built-in LED to LOW ▾
```

* Procedure:

1. Assemble the gas sensor circuit as described.
2. Connect an LED to pin 13 to indicate gas presence.
3. Write/upload code that:

    * Reads analog value from A0
    * Compares the value to a threshold
    * If above threshold, turns ON the LED
4. Bring smoke near the sensor and observe LED and readings.

* Result: