# EXP No.8: ULTRASONIC SENSOR

**OBJECTIVE:** Interface Ultrasonic sensor with ESP8266 to measure the distance from target and display it in Blynk IoT app

**MATERIALS REQUIRED:**

- ESP8266 NodeMCU board
- Ultrasonic sensor
- Breadboard
- Jumper wires
- USB cable
- Computer with Arduino IDE installed
- Blynk app installed on a smartphone or tablet

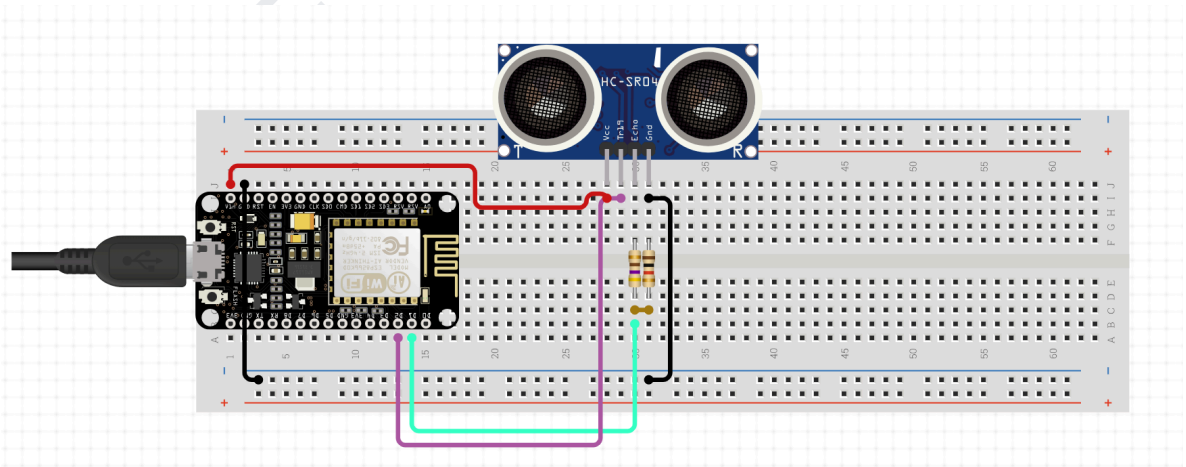
**THEORY:**

This project involves using an ultrasonic sensor with an ESP8266 microcontroller to measure distances and control devices via the Blynk app.

The ultrasonic sensor emits sound waves and calculates the distance to an object based on the time it takes for the waves to bounce back.

The ESP8266 processes this data and sends it to the Blynk app over Wi-Fi, where users can monitor the distance in real-time and trigger actions, such as turning on alarms or other devices, based on preset conditions. This integration enables the creation of responsive, smart systems for various applications.

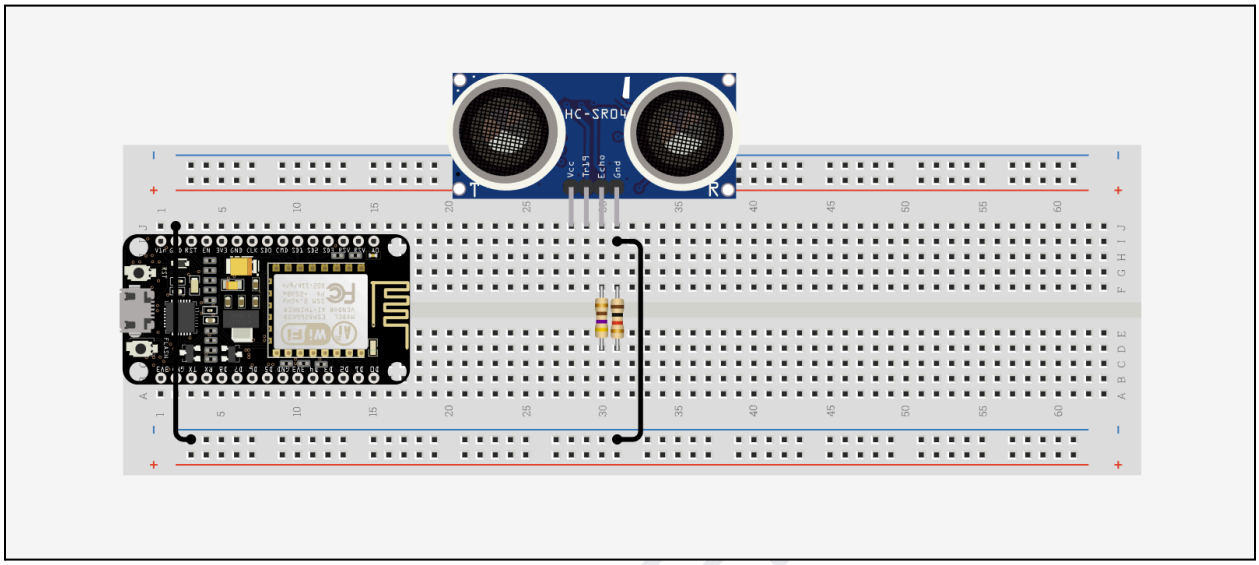
**CIRCUIT DIAGRAM:**



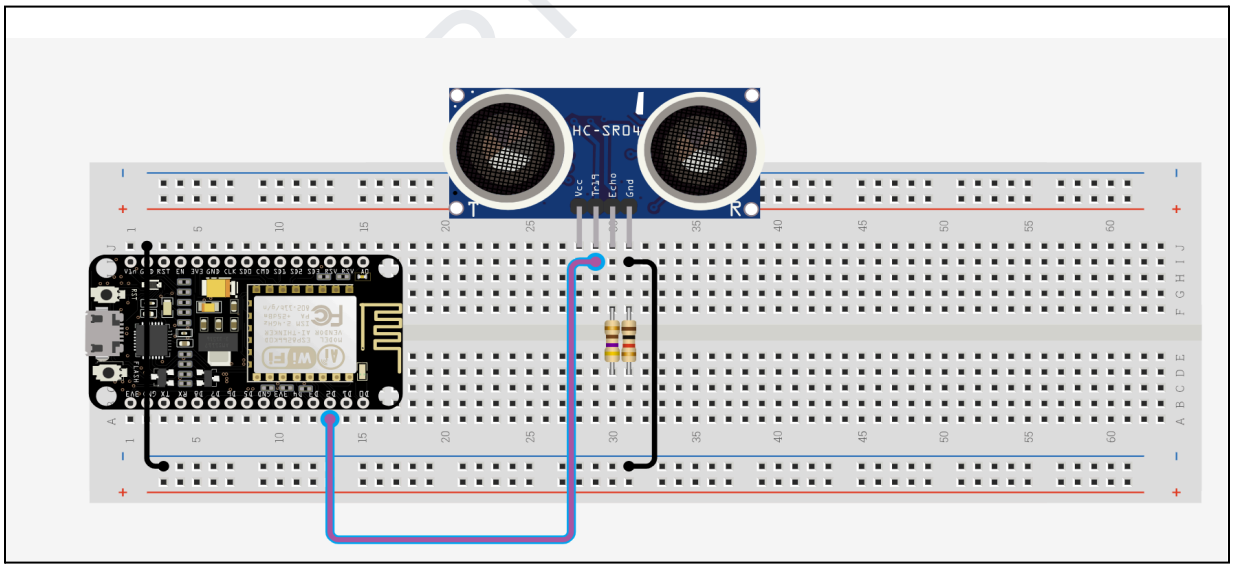
**PROCEDURE:**

1. **Hardware Setup:**

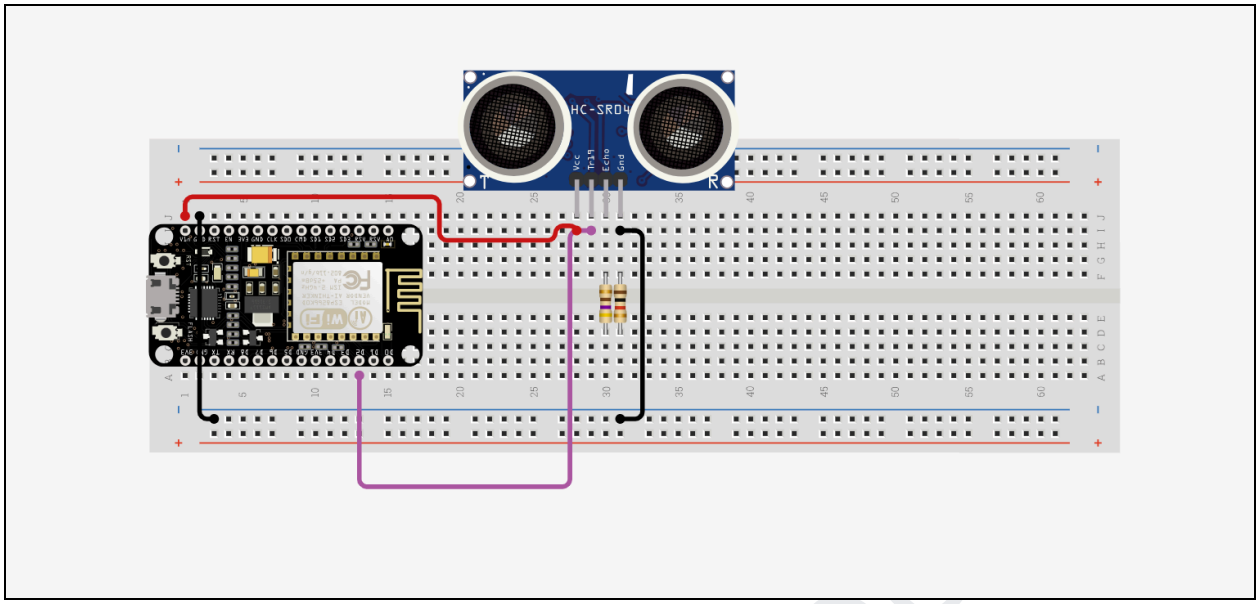
- Connect the GND pin of the esp8266 to GND pin of ultrasonic sensor through Resistor.



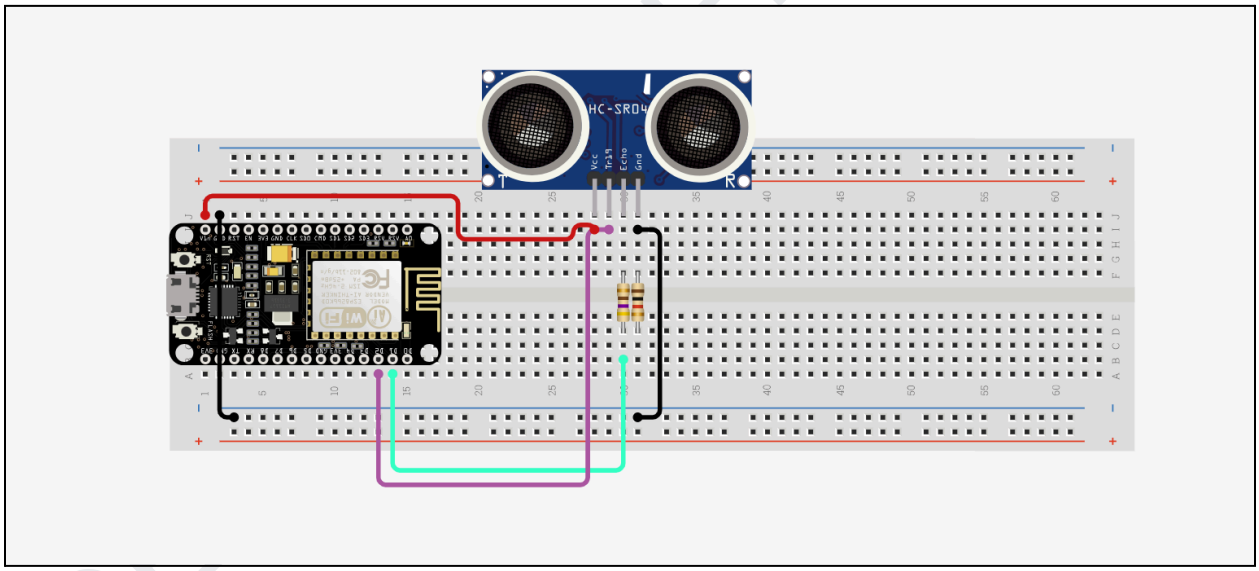

- Connect the one Data pin (GPIO) of the ESP8266 to Trig pin of sensor.




- Connect the $V_{in}$ pin of the ESP8266 to $V_{cc}$ of sensor.




- Connect the another Data pin(GPIO) of the ESP8266 to Echo pin of sensor.




2. **Software Setup:**

- Open the Arduino IDE on your computer.
- Install the Blynk library by navigating to Sketch -> Include Library > Manage Libraries and searching for “Blynk”.

3. **Blynk App Setup:**

- Open the Blynk app on your smartphone/tablet.

- Create a new project and note the Auth Token.
- Add Value Display widgets (Guage) and Lable set them to Virtual pin respectively.

4. **Programming:**

- Connect the ESP8266 to your computer using a USB cable.
- In the Arduino IDE, write the following code:

**Source code:**

```
#define BLYNK_TEMPLATE_ID "//copy your TEPLATE_ID//"
#define BLYNK_TEMPLATE_NAME "//copy your TEPLATE_NAME//"
#define BLYNK_AUTH_TOKEN "//copy yoUR AUTH TOKEN Here//"

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <NewPing.h>

// Your WiFi credentials
char ssid[] = "//your ssid name//";
char pass[] = "// ssid password//";  // Replace with your WiFi password

// Ultrasonic Sensor pins
#define TRIGGER_PIN D1 // D1
#define ECHO_PIN D2     // D2
#define MAX_DISTANCE 200  // Maximum distance we want to measure (in centimeters)

// Initialize NewPing library
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

BlynkTimer timer;
void setup() {
  // Debug console
  Serial.begin(115200);
  // Connect to Wi-Fi and Blynk
```

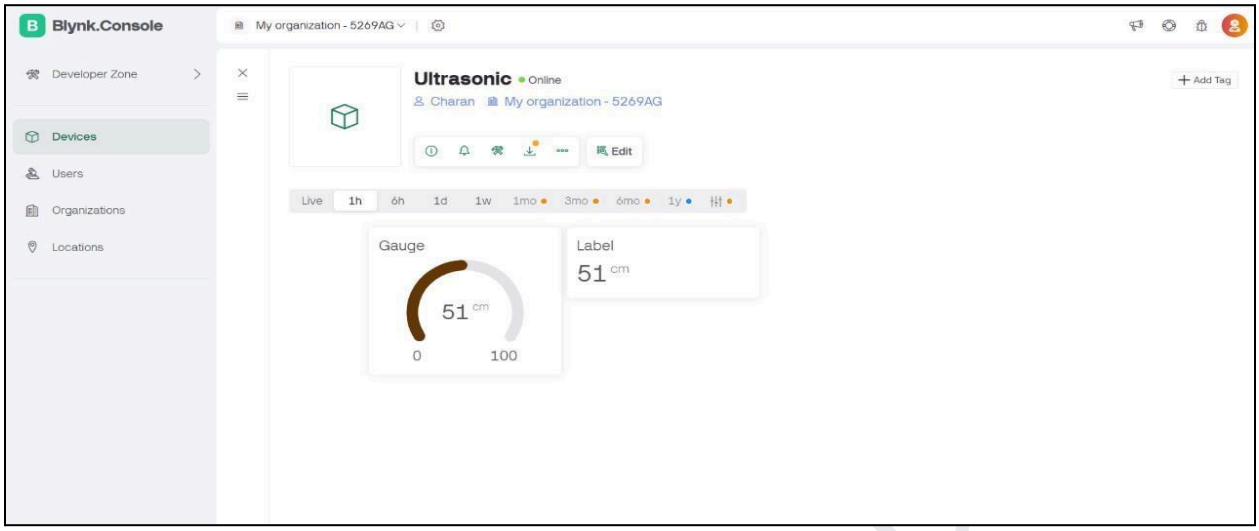
```
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}
void sendSensor() {
  // Measure distance
  unsigned int distance = sonar.ping_cm();

  // Send distance to Blynk app
  Blynk.virtualWrite(V0, distance);  // Gauge widget
  Blynk.virtualWrite(V1, distance);  // Label widget

  // Print distance to Serial Monitor for debugging
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}

void loop() {
  Blynk.run();
  timer.run();  // Initiates BlynkTimer
}
```

- Replace "YourAuthToken", "YourNetworkName", and "YourPassword" with the actual values.
- Upload the code to the ESP8266.

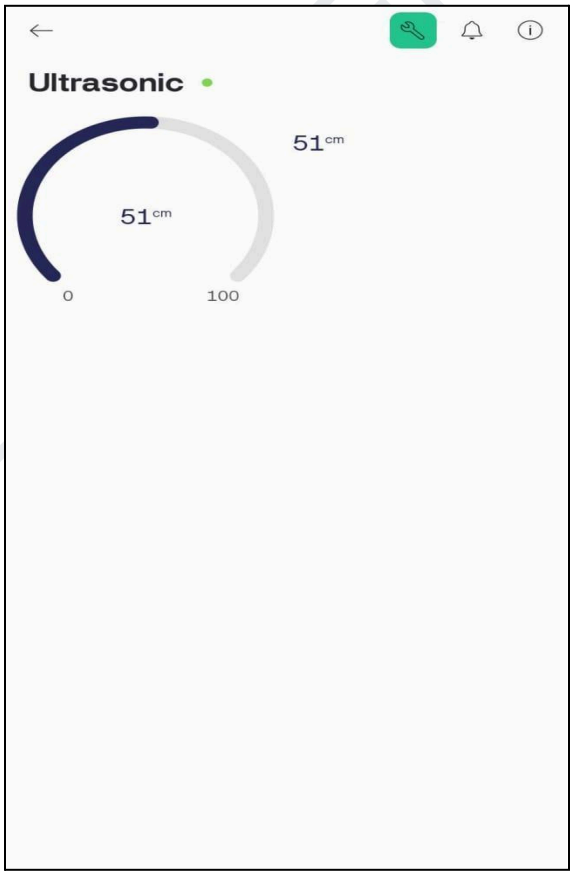
5. **Running the Experiment:**

- Open the Serial Monitor in the Arduino IDE to observe the distance.
- Open the Blynk app to view the live distance data on the Value Display widgets.

**Observation in Blynk website:**




**Observation in Blynk IoT mobile app:**



**Observation:**

| S.NO | Distance in centimeters | Distance in meters |
|---|---|---|
| 1. | 50cm | 0.5m |
| | | |
| | | |
| | | |
| | | |
| | | |

**Result:**

Hence we observed ultrasonic sensor with ESP8266 using with blynk IOT app.

**Conclusion:**

This experiment demonstrates how to interface the ultrasonic sensor with the ESP8266 board and use the Blynk IoT platform to remotely monitor display. The successful implementation confirms the practicality of using ESP8266 and Blynk for IoT applications.

**Appendix:**

**A. Symbols and Abbreviations**

- $V_{CC}$: Voltage Common Collector
- GND: Ground
- GPIO: General Purpose Input/Output

**B. Tools Required**

- ESP8266 NodeMCU board
- Breadboard
- Ultrasonic sensor
- Jumper wires
- USB cable
- Computer with Arduino IDE installed
- Blynk app installed on a smartphone or tablet

**C. Additional Resources**

- ESP8266 Documentation
- Arduino IDE Installation Guide
- Blynk Documentation

**D. Reference link with QR code:**

https://www.youtube.com/watch?v=6BLoUHTrmdo

This format provides a clear and comprehensive guide for conducting the experiment, ensuring students can follow along and achieve the desired outcomes.