Day 5 - Ultrasonic Sensor with NodeMCU

Today we'll learn how to use an HC-SR04 ultrasonic sensor with NodeMCU (ESP8266) to measure distances. We'll then create a mini project that uses this distance data.

Important Note

As you mentioned, the HC-SR04 ultrasonic sensor requires 5V to operate properly, but the NodeMCU GPIO pins can only handle 3.3V. We'll use a voltage divider to safely interface between these voltage levels.

Step 1: Connect the Ultrasonic Sensor

Components Needed:

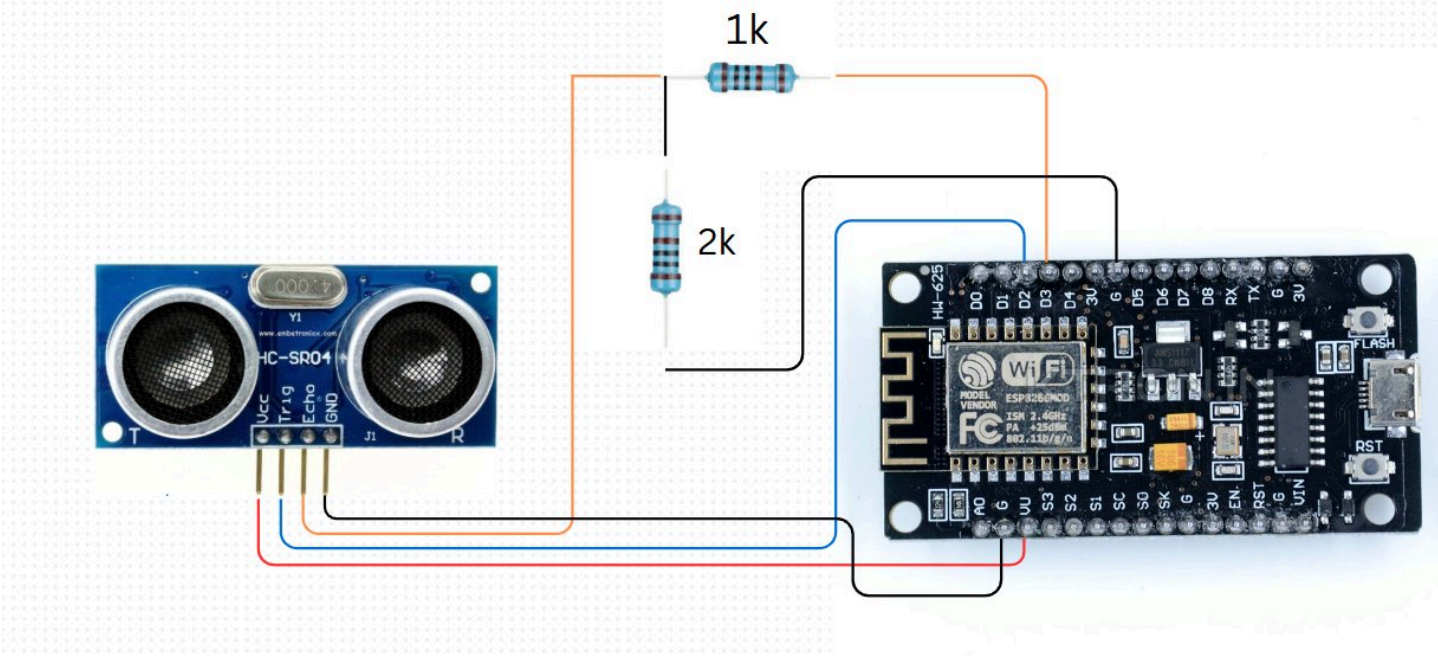
- 1x HC-SR04 Ultrasonic Sensor
- 2x Resistors (1kΩ and 2kΩ) for voltage divider
- Breadboard
- Jumper wires
- NodeMCU (ESP8266)
- LED (for our mini project)
- 220Ω Resistor (for LED)

Connections:

1. VCC pin of HC-SR04 → 5V (Vin) on NodeMCU
2. GND pin of HC-SR04 → GND on NodeMCU
3. Trig pin of HC-SR04 → D1 (GPIO 5) on NodeMCU
4. Echo pin of HC-SR04 → Voltage divider → D2 (GPIO 4) on NodeMCU

Voltage Divider Setup:

- Connect Echo pin to one end of the 1kΩ resistor
- Connect the other end of the 1kΩ resistor to D2 on NodeMCU
- Connect the junction between the resistors to the 2kΩ resistor
- Connect the other end of the 2kΩ resistor to GND



This voltage divider reduces the 5V output from the Echo pin to approximately 3.3V, which is safe for the NodeMCU.

Step 2: Read Distance Values from Ultrasonic Sensor

Basic Code:

```
const int trigPin = D1;  // Trig pin to D1 (GPIO 5)
const int echoPin = D2;  // Echo pin to D2 (GPIO 4)

void setup() {
  Serial.begin(115200);  // Start Serial Monitor
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // Clear the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Send 10µs pulse to trigger
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the echo time (in microseconds)
  long duration = pulseIn(echoPin, HIGH);

  // Calculate the distance in centimeters
  // Sound travels at ~343m/s, which is 0.0343 cm/µs
  // Distance = (Time × Speed) / 2 (divide by 2 because sound travels to object and back)
```

```
  float distance = duration * 0.0343 / 2;

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

  delay(500);  // Delay for better readability
}
```

Upload this code and open the Serial Monitor (115200 baud). You'll see the measured distance changing as you move objects closer or farther from the ultrasonic sensor.

Step 3: Connect the LED (for our mini project)

Additional Connections:

- LED Anode (+) → D5 (GPIO 14) through a 220Ω resistor
- LED Cathode (-) → GND

Step 4: Create a Proximity Alert System Mini Project

Now let's combine the ultrasonic sensor with an LED to create a proximity alert system. The LED will change its blinking rate based on how close an object is to the sensor.

Final Code:

```
const int trigPin = D1;  // Trig pin to D1 (GPIO 5)
const int echoPin = D2;  // Echo pin to D2 (GPIO 4)
const int ledPin = D5;   // LED pin to D5 (GPIO 14)

// Distance thresholds (in cm)
const int closeDistance = 10;
const int mediumDistance = 30;

void setup() {
  Serial.begin(115200);  // Start Serial Monitor
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

float getDistance() {
  // Clear the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Send 10µs pulse to trigger
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the echo time (in microseconds)
  long duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  float distance = duration * 0.0343 / 2;
  return distance;
}

void loop() {
  // Get distance from sensor
  float distance = getDistance();

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

  // LED behavior based on distance
  if (distance < closeDistance) {
    // Very close - rapid blinking
    digitalWrite(ledPin, HIGH);
    delay(100);
    digitalWrite(ledPin, LOW);
    delay(100);
  }
  else if (distance < mediumDistance) {
    // Medium distance - moderate blinking
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    delay(500);
  }
  else {
    // Far away - slow blinking
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
  }
}
```

Key Concepts Explained

HC-SR04 Ultrasonic Sensor:

- The sensor uses sonar to determine distance to an object, like bats or dolphins.

- It sends out an ultrasonic pulse and measures the time it takes for the echo to return.
- Working principle: Time × Speed = Distance

Voltage Divider:

- We use a voltage divider with 1kΩ and 2kΩ resistors to reduce the 5V output from the Echo pin to a safe 3.3V for NodeMCU.
- Formula: Vout = Vin × (R2/(R1+R2)) = 5V × (2kΩ/(1kΩ+2kΩ)) ≈ 3.33V

Timing Functions:

- `pulseIn()` : Measures the time (in microseconds) a pin stays in a specified state (HIGH or LOW).
- `delayMicroseconds()` : Creates precise delays in microseconds for accurate timing.

Distance Calculation:

- Speed of sound in air ≈ 343 meters/second or 0.0343 centimeters/microsecond.
- We divide by 2 because the sound travels to the object and back.

Advanced Project Ideas:

1. Connect this to a buzzer for audible proximity alerts
2. Use multiple LEDs for different distance ranges
3. Implement an OLED display to show the exact distance
4. Create a parking sensor system for vehicles