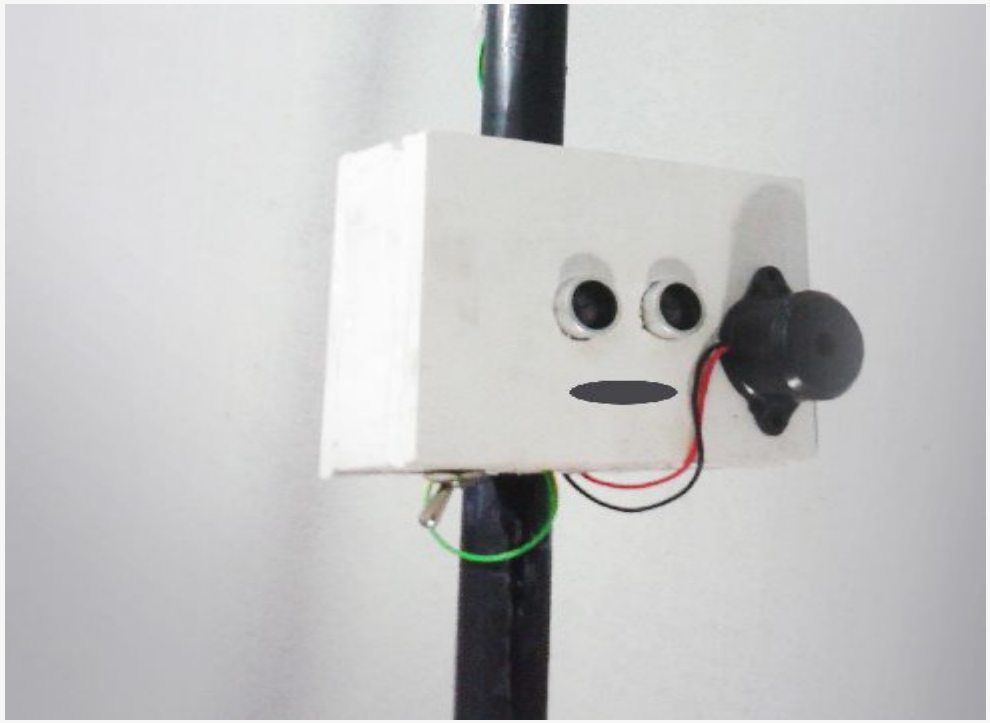
ABLEDUCATION™
ACTIVITY BASED LEARNING

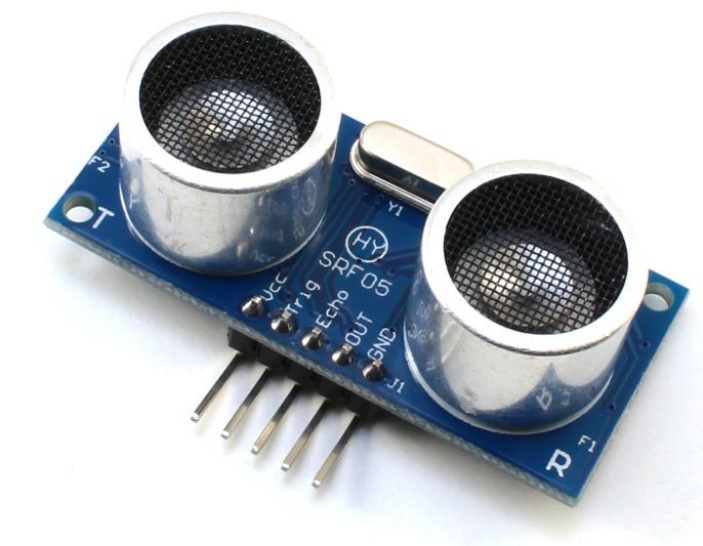
Blind Stick Project



ABLEDUCATION™
ACTIVITY BASED LEARNING

Ultrasonic Sensor HC-SR04

An **ultrasonic sensor** is an instrument that measures the distance to an object using **ultrasonic** sound waves. An **ultrasonic sensor** uses a **transducer** to send and receive **ultrasonic** pulses that relay back information about an object's proximity.



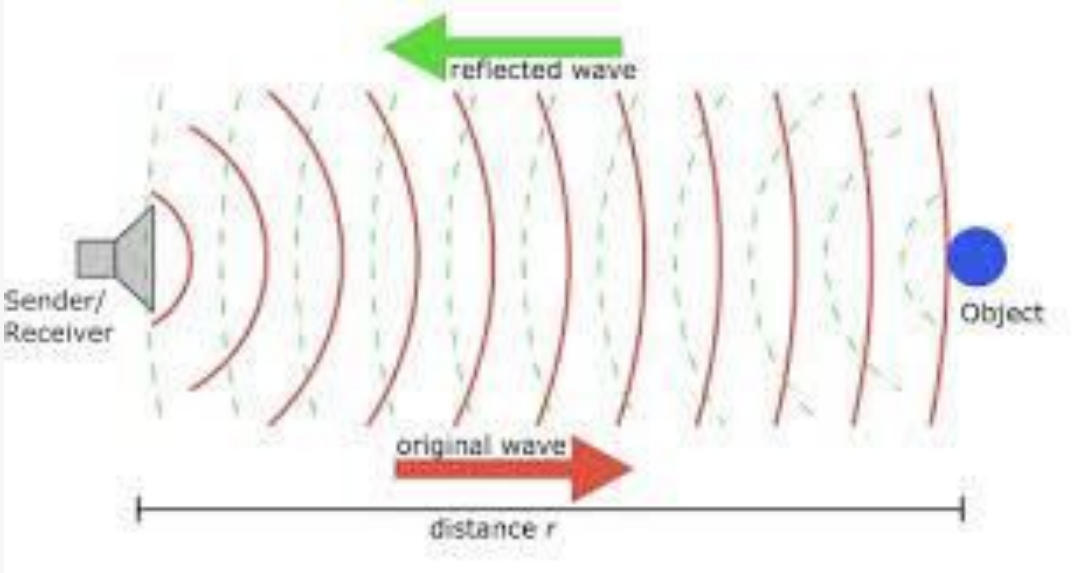
ABLEDUCATION™
ACTIVITY BASED LEARNING

Working principle of Ultrasonic Sensor

- **Ultrasonic sensors work** by emitting sound waves at a frequency too high for humans to hear. Then they wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object.
- The HC-SR04 offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet.

ABLEDUCATION™
ACTIVITY BASED LEARNING

Working Diagram



ABLEDUCATION™
ACTIVITY BASED LEARNING

Working of project

- There are two openings in Ultrasonic sensor first is transmitter (or Trigger) and second is receiver (or Echo).
- Ultrasonic sensor sends high frequency pulses, these pulses reflects from object and takes as Echo, time between echo and Trig is measured by the microcontroller or Arduino which is directly proportional to distance.
- The speed of sound is 341 meter per second in the air, and the distance between sensor and object is equal to time multiplied by speed of sound divided by two.

Distance = (Time * Speed Of Sound) ÷ 2

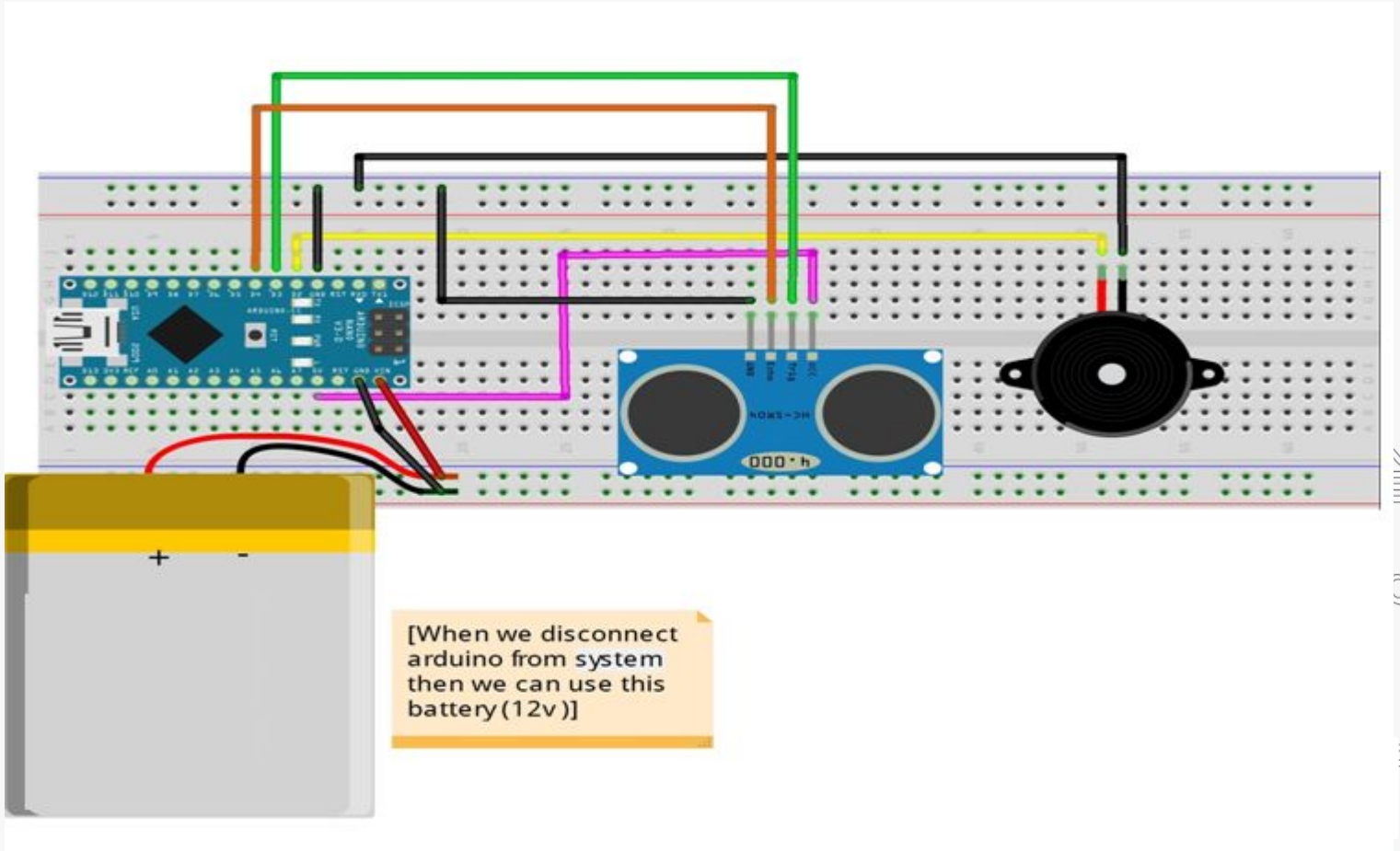
- After the distance measurement, Arduino makes a beep format using buzzer, when distance is high, frequency of beep is decreased and beep frequency is increased when distance is low.

Components Required

- Arduino Nano
- Ultrasonic Sensor HC-SR04
- Buzzer
- Breadboard
- Jumper wires

ABLEDUCATION™
ACTIVITY BASED LEARNING

Connection Diagram



ABLEDUCATION™
ACTIVITY BASED LEARNING

Connections

1. Connect **Trig** pin of Ultrasonic sensor with **D3** pin of Arduino Nano.
2. Connect **Echo** pin of Ultrasonic sensor with **D4** pin of Arduino Nano.
3. Connect **Vcc** pin of Ultrasonic sensor with **5V** of Arduino Nano.
4. Connect **GND** pin of Ultrasonic sensor with **GND** of Arduino Nano.
5. Connect positive pin of buzzer with **D2** pin of Arduino Nano and negative pin with **GND** pin of Arduino Nano.
6. After uploading the code in Arduino, remove Arduino cable and connect **12V** li-ion battery with Arduino Nano.

ABLEDUCATION™
ACTIVITY BASED LEARNING

Code

Blindstick_project | Arduino 1.8.19

File Edit Sketch Tools Help

Blindstick_project §

```
// defines pins numbers
const int trigPin = 3;
const int echoPin = 4;
const int buzzer = 2;
//const int ledPin = 5;

// defines variables
long duration;
int distance;
int safetyDistance;

void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(buzzer, OUTPUT);
//pinMode(ledPin, OUTPUT);
Serial.begin(9600); // Starts the serial communication
}

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

ABLEDUCATION™
ACTIVITY BASED LEARNING

Blindstick_project | Arduino 1.8.19

File Edit Sketch Tools Help

Blindstick_project §

```
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
      distance= duration*0.0066/2;
    safetyDistance = distance;
if (safetyDistance <= 5){
  digitalWrite(buzzer, HIGH);
 // digitalWrite(ledPin, HIGH);
}
else{
  digitalWrite(buzzer, LOW);
  //digitalWrite(ledPin, LOW);
}
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

ABLEDUCATION™
ACTIVITY BASED LEARNING

Project Link: https://youtu.be/bItl44lYJBI