|  |  |
| --- | --- |
| **Maharaja Surajmal Institute of Technology**  *Electronic Devices Project*  Paper Code – ETEC-106 | Abstract  Using the HC-SR04 Ultrasonic Sensor with an Arduino UNO to make a proximity buzzer.  Arjun Handa  ECE-1  01915002820 |

INDEX

|  |  |
| --- | --- |
| S. NO. | TOPIC |
| 1 | INTRODUCTION |
| 2 | ABOUT THE PROJECT |
| 3 | LIST OF COMPONENTS USED |
| 4 | CIRCUIT |
| 5 | SOURCE CODE |

INTRODUCTION

ROBOTICS is a rapidly expanding and fascinating field in today's world. It's the simplest method for keeping up with the latest technological advancements. Because today's communication is based on technological advancements, we chose to work in the robotics field and design something that will make human life easier in today's world. As a result, we are backing this cause.

Robotics is the study of the design, construction, operation, and application of robots and computer systems for control, sensory feedback, and information processing.

ABOUT THE PROJECT

This project is based upon object detection in close proximity. The HC-SR04 Ultrasonic Sensor detects any object in the required range and gives off the signal to Arduino UNO which makes the buzzer alert.

The range can be anywhere from 2 to 400 cm with an accuracy of 3mm.

Range and other parameters can be changed by changing them in the Arduino code and then uploading them to the Arduino UNO via the USB cable.

Power to the circuit is provided with the USB cable only but a battery can also be used with a barrel type jack as there is a voltage regulator of 5V available on the Arduino UNO.

LIST OF COMPONENTS USED

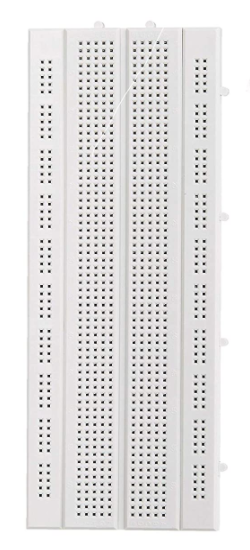
* Arduino UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits. This is the main controller for our project.



* Breadboard

A breadboard, or protoboard, is a construction base for prototyping of electronics.



* Buzzer

Small and light high power-dissipation capability low frequency high efficiency buzzer.



* HC-SR04 Ultrasonic Sensor

This HC-SR04-Ultrasonic Range Finder is a very popular sensor that is found in many applications where it requires measuring distance and detecting the objects.

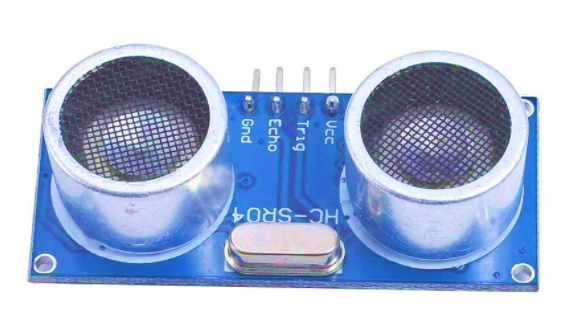
The module has two eyes like projects in the front which form the Ultrasonic transmitter and Receiver. The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object like bats or dolphins do.

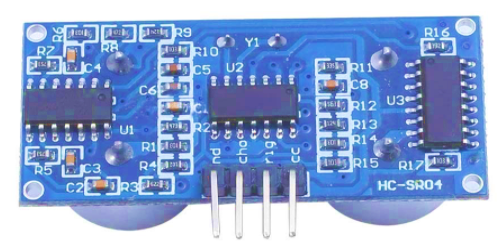
Operating Voltage: 5 V

Sonar Sensing Range: 2-400 cm

Max. Sensing Range: 450 cm

Frequency: 40 kHz



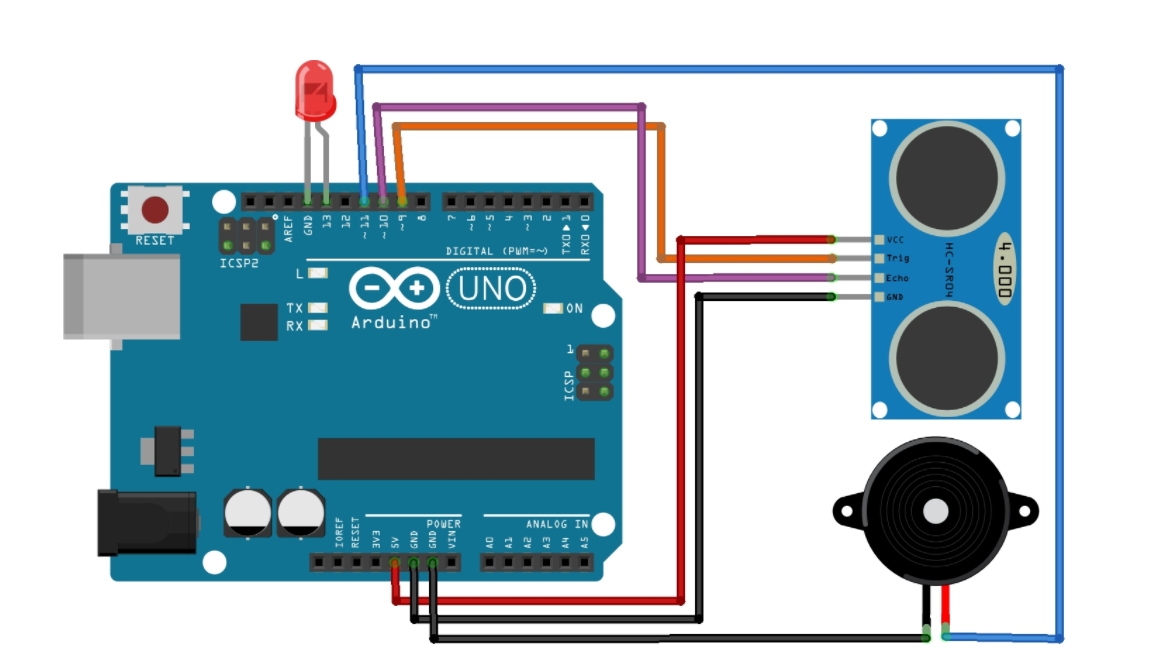


* Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wires.



CIRCUIT



The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.

The HC-SR04 sensor attach to the Breadboard

The Sensor VCC connect to the Arduino Board +5V

The Sensor GND connect to the Arduino Board GND

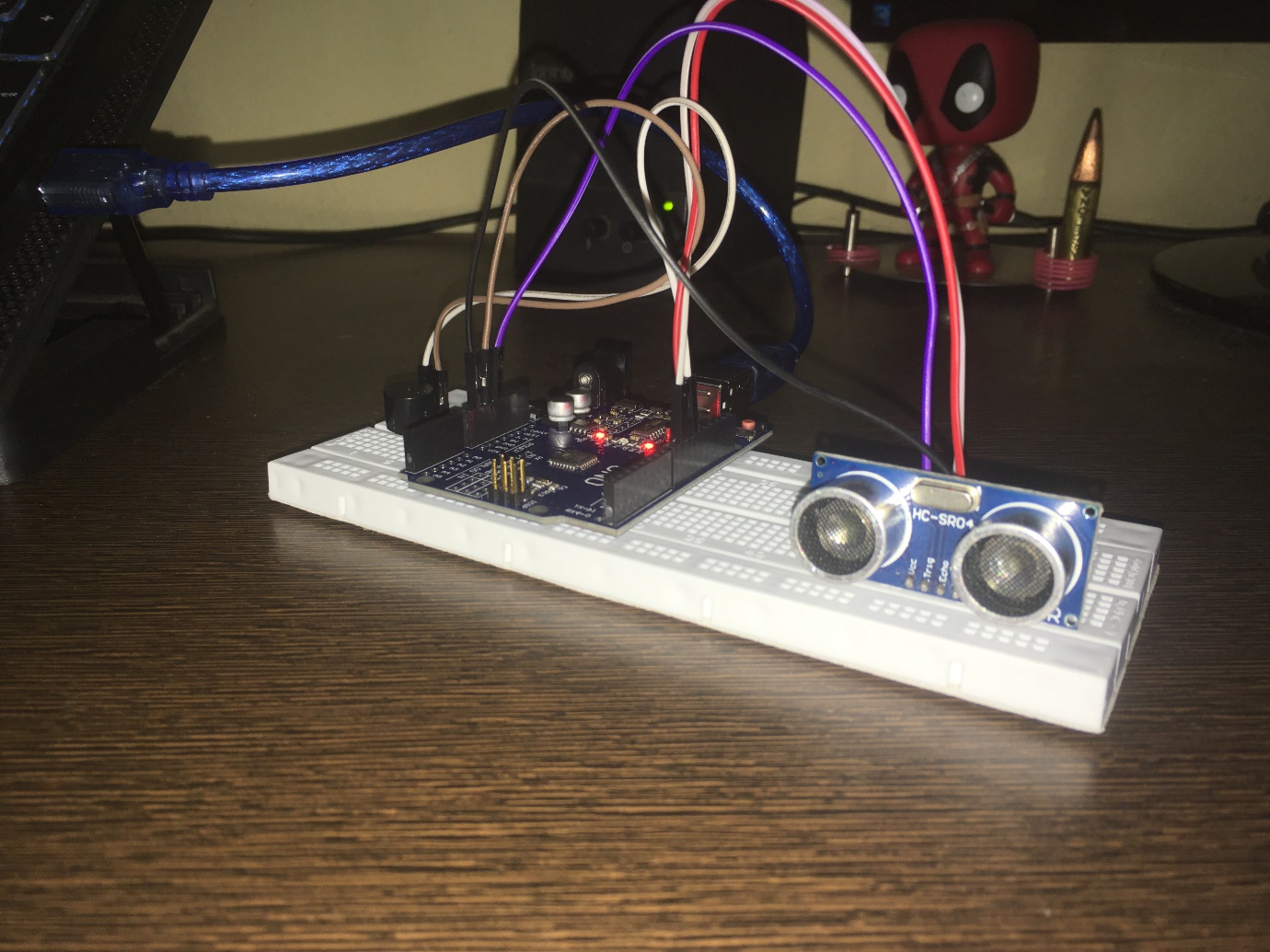
The Sensor Trig connect to the Arduino Board Digital I/O 9

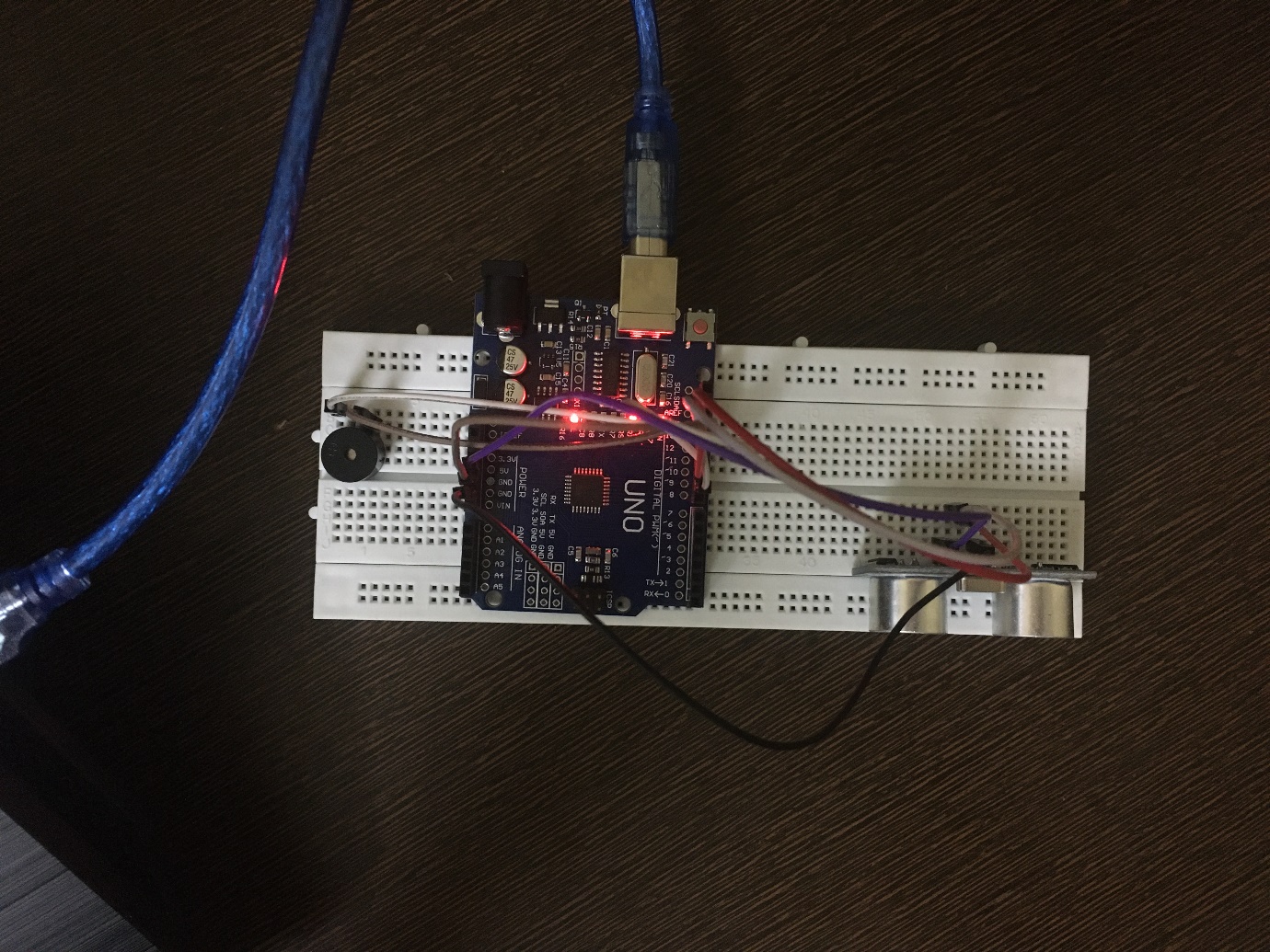
The Sensor Echo connect to the Arduino Board Digital I/O 10

The Buzzer attached to the Breadboard

The Buzzer long leg (+) connect to the Arduino Board Digital 11

The Buzzer short leg (-) connect to the Arduino Board GND





SOURCE CODE

// defines pins numbers

const int trigPin = 9;

const int echoPin = 10;

const int buzzer = 11;

// 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);

Serial.begin(9600); // Starts the serial communication

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// 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\*0.034/2;

safetyDistance = distance;

if (safetyDistance <= 10){

digitalWrite(buzzer, HIGH);

}

else{

digitalWrite(buzzer, LOW);

}

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance);

}

