COMMUNICATION SYSTEMS LABORATORY

MINI PROJECT

SECURITYSYSTEM USING ULTRASONIC SENSOR- HCSR04

- A REPORT

BHARANI V - 20ECEAL59

GOPINATH P - 20ECEA36

SARATH BABU T R - 20ECEA46

SUBRAMANIAN AL - 20ECEA52

SURYA B - 20ECEA53

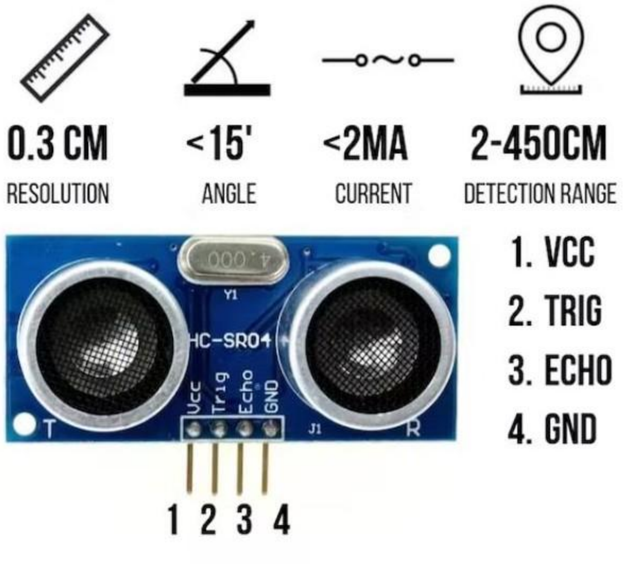
THIRD YEAR,
Department of Electronics and Communication Engineering
VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY,
VIRAGANOOR,MADURAI-625009

OVERVIEW

In this project, a security alarm is a system to detect intrusion-unauthorized entry - into a building or area. Securit alarms are used in residential, commercial, industrial, and military properties for protection against burglary(theft) or property damage, as well as personal personal protection against intruders.

What is an Ultrasonic Sensor?

Ultrasound is acoustic (sound) energy in the form of waves having a frequency above the human hearing range. Ultrasonic Sensor HC-SR04 is a sensor that can measure **distance**. It emits an **ultrasound** at **40 000 Hz (40kHz)** which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

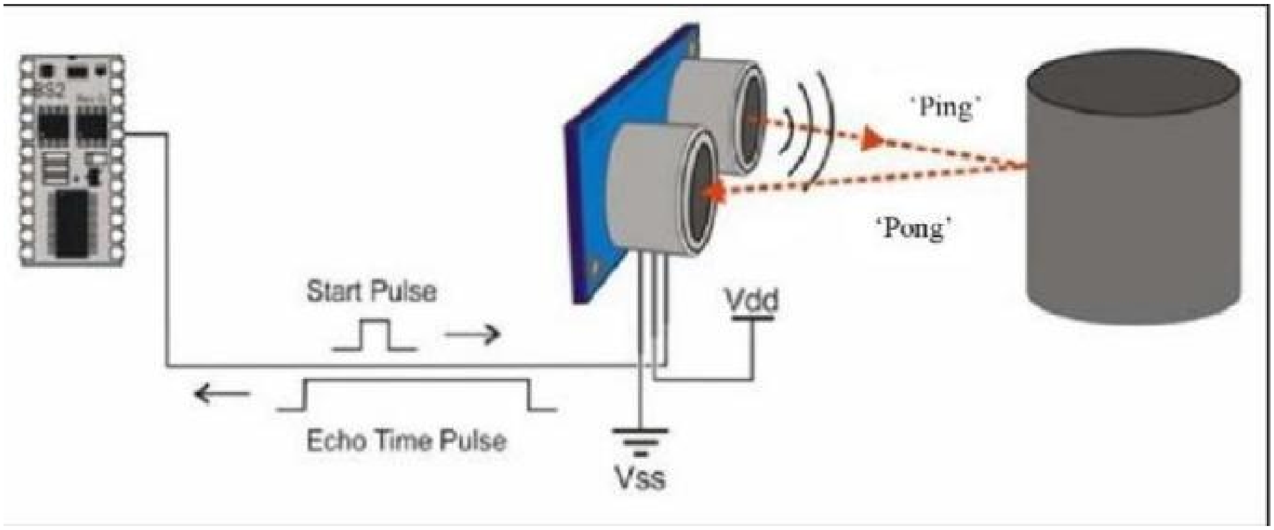


Components Required

The voltage that the sensor output changes accordingly to the smoke or gas level that exists in the atmosphere.The output voltage of the sensor is directly proportional to the concentration of smoke or gas.

In other words,the relationship between voltage and gas concentration is the following

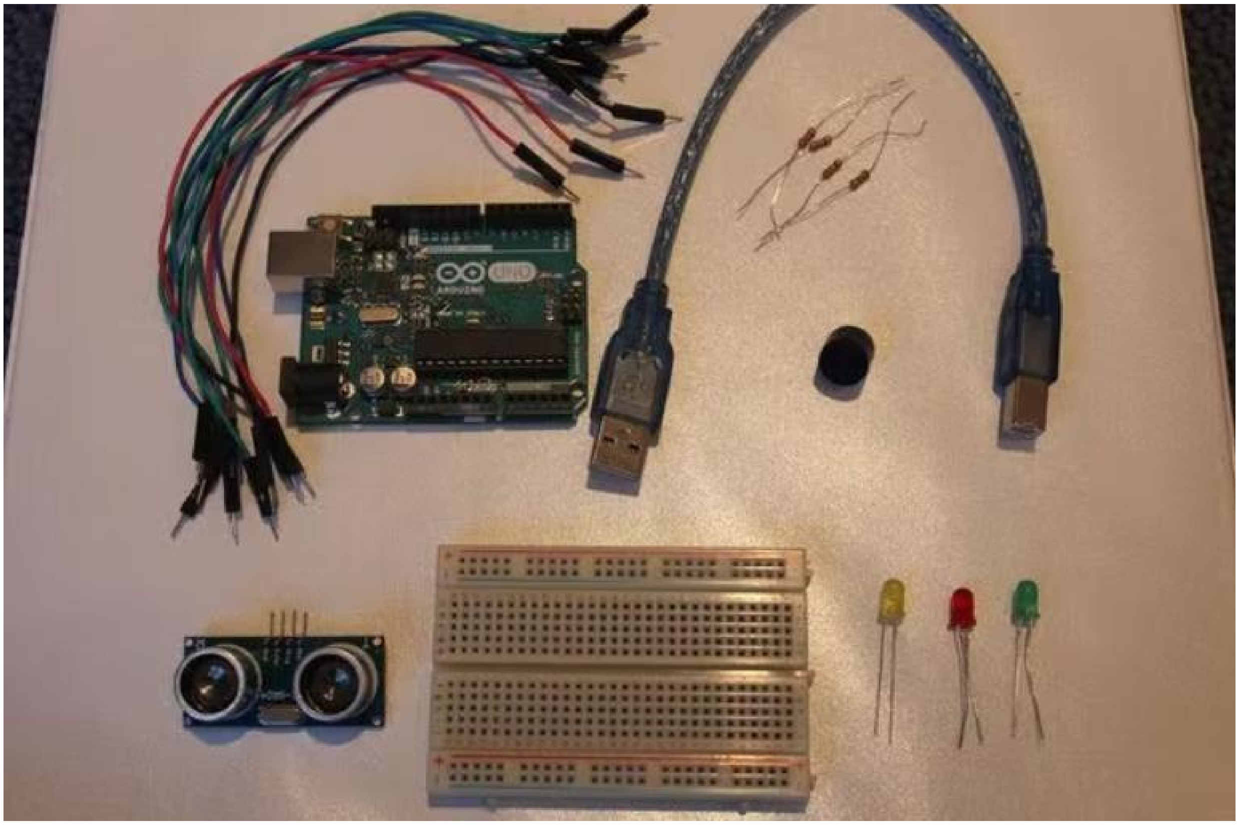
- ❖ **Arduino UNO**
- ❖ **Breadboard (generic)**
- ❖ **Ultrasonic Sensor HC - SR04**
- ❖ **Buzzer**
- ❖ **LED**
- ❖ **Resistors**
- ❖ **Jumper wires**



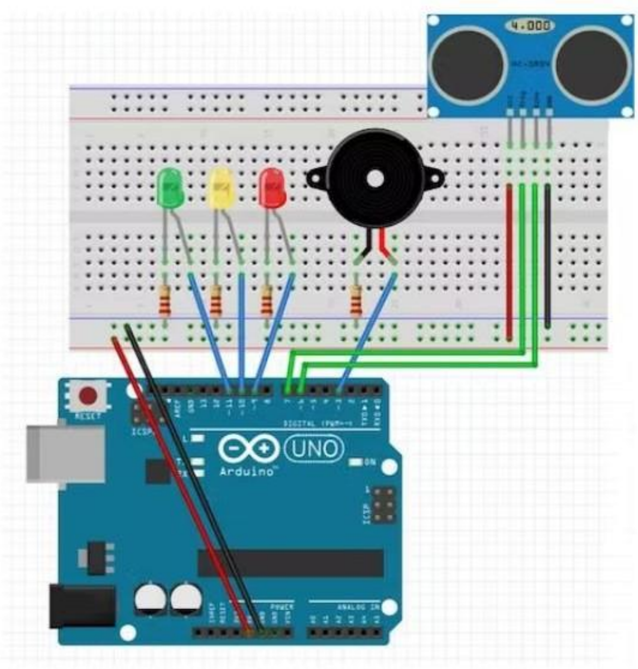
BASIC ULTRASONIC SENSOR WORKING

The illustration in above figure shows how sound waves, transmitted in the shape of a cone, are reflected from a target back to the transducer. An output signal is produced to perform some kind of indicating or control function. A minimum distance from the sensor is required to provide a time delay so that the “echoes” can be interpreted.

ASSEMBLE MATERIALS



SETUP



Connect a red wire from the 5V pin on the Arduino to the positive channel of the breadboard. Connect a black wire from the GND pin on the Arduino to the negative channel of the breadboard:

Buzzer = pin 7

On Ultrasonic sensor:

Echo = pin 3

Trig = pin 2

LEDs:

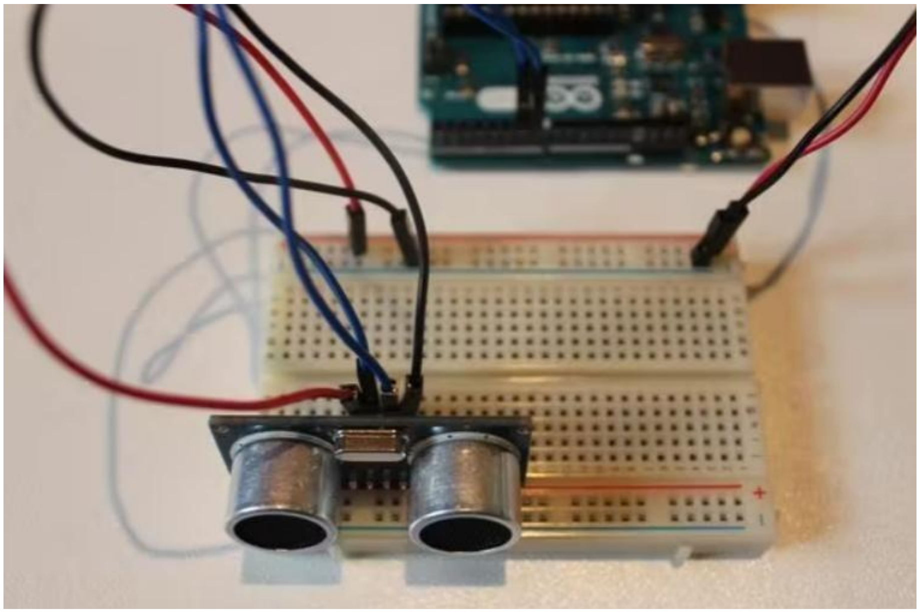
Red LED = pin 4

Yellow LED = pin 5

Green LED = pin 6

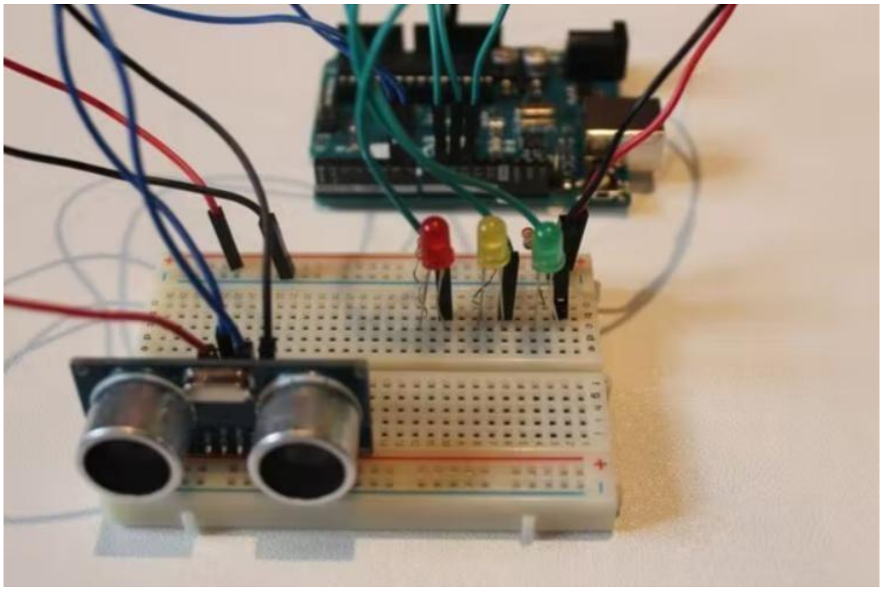
The green wires connected to the LEDs should be connected in line to the positive side of the LED, while the negative side of the LED should be connected to the negative channel of the breadboard using a 220 ohm resistor.

ASSEMBLY - ULTRASONIC SENSOR



Time to connect the HC-SRO4 ultrasonic sensor! A great tip is to place the ultrasonic sensor as far right to the breadboard as possible and make sure that it is facing out. Referring back to the setup picture, you should connect the GND pin on the ultrasonic sensor to the negative channel on the breadboard. Next connect the Trig pin on the sensor to pin 2 on the Arduino and connect the Echo pin on the sensor to pin 3 on the Arduino. Lastly, connect the VCC pin on the ultrasonic sensor to the positive channel on the breadboard. Refer to the picture above if anything gets confusing.

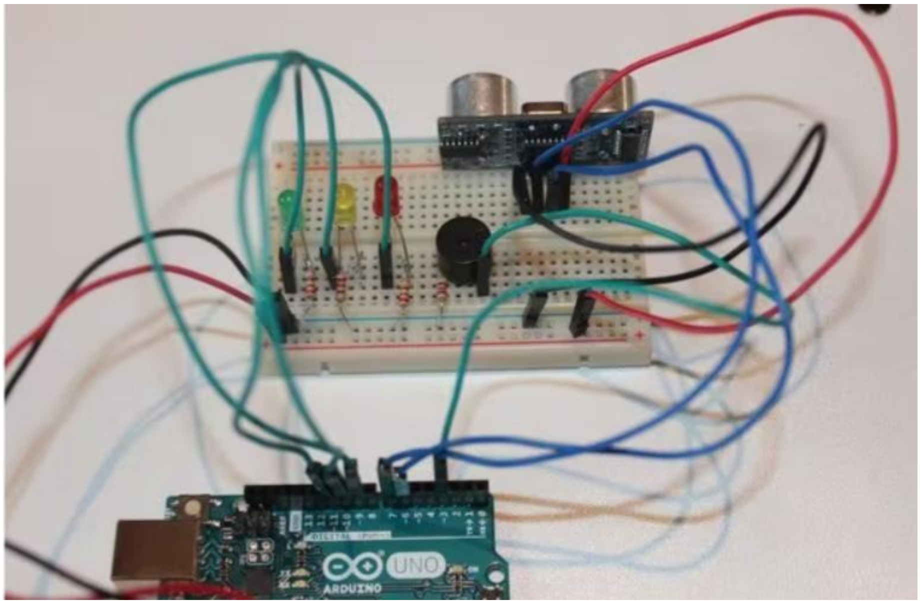
ASSEMBLY - LEDS



The next step is to connect the LED's to the breadboard and Arduino. If you need to, I highly recommend that you refer back to the setup picture (Step 2), attaching the LED's is pretty easy, there's a lot of repetition. Let's first attach the Green LED. So the way to do this, is to connect the anode (the longer leg) to pin 6 on the Arduino with a green wire, and to connect the cathode (the shorter leg) to the negative channel on the breadboard, using a 220 ohm resistor. Then repeat that step for the Yellow and then the Red LED, make sure to connect the anode (the longer leg) of the yellow LED to pin 5 on the Arduino and then connect the anode of the red LED to pin 6. Once you have done that, your setup should look similar to the picture above.

Resistors are not absolutely necessary, however they are highly recommended to be used.

ASSEMBLY - BUZZER



The last part of the setup for this, is connecting the buzzer to the breadboard and the Arduino. This is one of the easiest parts of the whole setup. All that is required to do is to connect the longer leg of the buzzer to pin 7 of the Arduino using a green wire and then connect the shorter leg of the buzzer to the negative channel of the breadboard using a 220 ohm resistor.

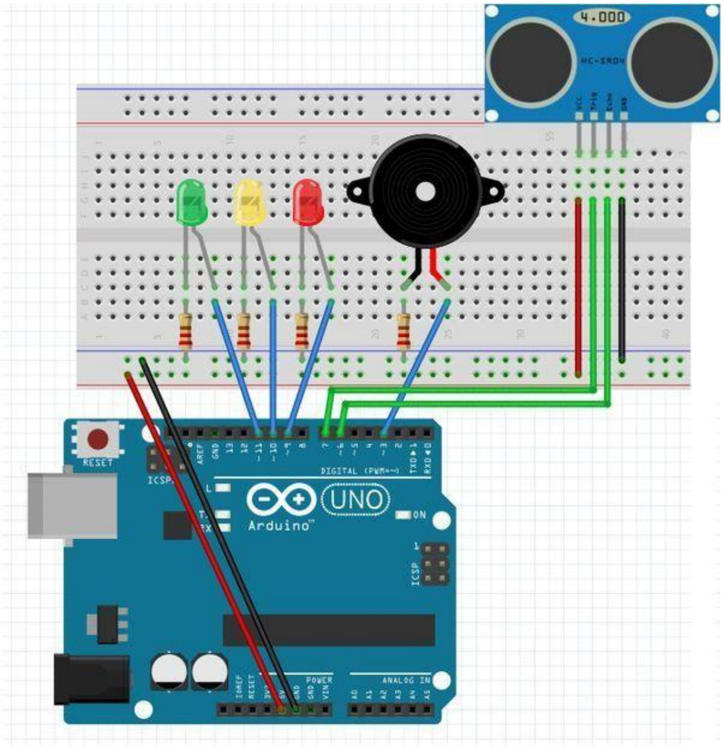
It is HIGHLY recommended to use a resistor in connecting the shorter leg of the buzzer to the negative channel of the breadboard. This greatly reduces the volume of the buzzer and prevent it from dying to quickly.

CODE FOR THE ULTRASONIC SECURITY SYSTEM

```
#define trigPin
#define echoPin
#define  LEDlampRed
#define LEDlampYellow
#define LEDlampGreen
#define soundbuzzer
int sound = 500;
```

```
void setup()
{
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(LEDlampRed, OUTPUT);
pinMode(LEDlampYellow, OUTPUT);
pinMode(LEDlampGreen, OUTPUT);
pinMode(soundbuzzer, OUTPUT);
}
void loop()
{
long durationindigit, distanceincm;
digitalWrite(trigPin, LOW);
delayMicroseconds(2); digitalWrite(trigPin,
HIGH); delayMicroseconds(10);
digitalWrite(trigPin,   LOW);
durationindigit = pulseIn(echoPin, HIGH);
distanceincm =  (durationindigit/5)  /  29.1;
if (distanceincm < 50)
{
digitalWrite(LEDlampGreen,  HIGH);
}
Else
{
digitalWrite(LEDlampGreen,  LOW);
}
if (distance < 20)
{
digitalWrite(LEDlampYellow,  HIGH);
}
Else
{
digitalWrite(LEDlampYellow,LOW);
}
if (distance < 5)
{
digitalWrite(LEDlampRed, HIGH);
sound = 1000;
}
Else
{
digitalWrite(LEDlampRed,LOW);
}
if (distanceincm > 5 || distanceinsm <= 0)
{
Serial.println("Outside the permissible range of distances");
noTone(soundbuzzer);
}
Else
{
Serial.print(distance);
Serial.println("   cm");
tone(buzzer, sound);
}
delay(300);
}
```

FINAL SETUP


4.000
HC-SR04
RESET
ICSP
DIGITAL (PWM=~)
UNO
Arduino
ON
ICSP
POWER
ANALOG IN