21AIE114-PRINCIPLES OF MEASUREMENT AND SENSORS
Project Report

Topic:

Door Buzzer Using Ultrasonic Sensor and Automatic Room temperature controller

GROUP MEMBERS

NAVNEETH SURESH	-	AM.EN.U4AIE21147
AKSHAY KRISHNAN T	-	AM.EN.U4AIE21109
ASWIN US	-	AM.EN.U4AIE21119
MUHAMMED SHAJAHAN	-	AM.EN.U4AIE21144
NIRANJAN PRASANTH	-	AM.EN.U4AIE21148
ISHAAN DADHICH	-	AM.EN.U4AIE21134

INDEX

ABSTRACT

Automatic door systems are very common nowadays. In this project we are making an ultrasonic sensor which is used as a distance sensor, it will tell us the distance at which the object is placed. Using this distance value, we will turn the buzzer on or off. The concept used in developing this technology is ultrasonic reflection similar to that which is used in a SONAR. As the distance reduces to below 10 units the doors will open automatically with a buzzer sound. We also use Arduino UNO R3 in the circuit as a microcontroller. The functionalities of the sensor have been programmed using C language and platform used to develop the circuit is Tinkercad. We will also be a developing a signal conditioning system for the above system using LTSpice.

AC's now have function which automatically cools and turns off according to the temparuture. We have developed such a system in our project, using switching operation of transistors, unidirectional current flow in diodes, the principle of operation of motors, the resistance from resistors combined with the transformation capability of the transducer, which is temperature sensor in this case. The basic idea of the system is it turns on the fan if the temperature sensor detects a temperature which is too high. Similarly it switches on the heater when the temperature is too low. The project is developed using an Arduino as the microcontroller. The functionalities of the Arduino has been programmed in C language. The signal conditioning of the circuit will be simulated using LTSpice software.

UltraSonic Sensor

What is an ultrasonic sensor?

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.

High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

How Ultrasonic Sensors Work.

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

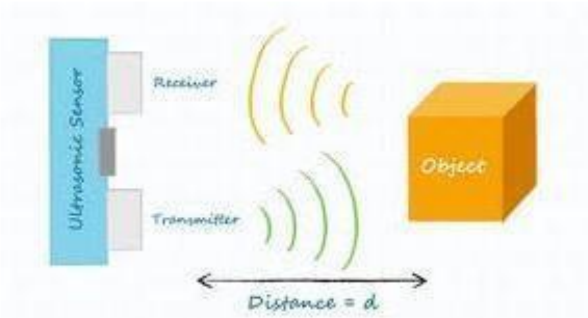
Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required.

If you need to measure the specific distance from your sensor, this can be calculated based on this formula:

Distance = ½ T × c

200C (680F), the speed of sound is 343 meters/second

(1 125 feet/second), but this varies depending on temperature and humidity



Ultrasonic sensors have been used throughout many applications and industries.

Some of its applications include :

Robotic sensing

Stacking height control

Loop control

Liquid level control

Full detection

Counting people/people detection

Presence detection

Detecting breaks in threads or wires

Tank level detection etc

Distance measurement- regardless of an object's shape, colour or surface texture
Properties

The ultrasonic sensor we used in the simulation is a HC-SR04 Ultrasonic sensor. The HC-SR04 Ultrasonic sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.

Vcc

The Vcc pin powers the sensor, usually with 5v

Trigger pin

Trigger pin is an Input pin. This pin has to be kept high for lops to initialize measurement by sending Ultrasonic wave.

Echo

Echo pin is an Output pin. This pin poes high for a period of time which will be equal to the time taken for the Ultrasonic wave to return back to the sensor.

HC-SR04 Characteristics

Supply voltage: 5V (DC).

Supply current: J 5mA.

Modulation frequency: 40Hz.

Output: 0 - 5V (Output high when obstacle detected in range).

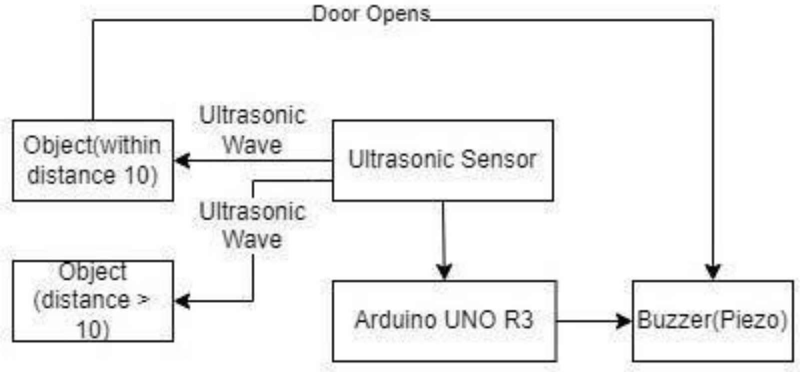
Beam Angle: Max 15 degree.

Theoretical measuring Distance: 2cm - 400cm.

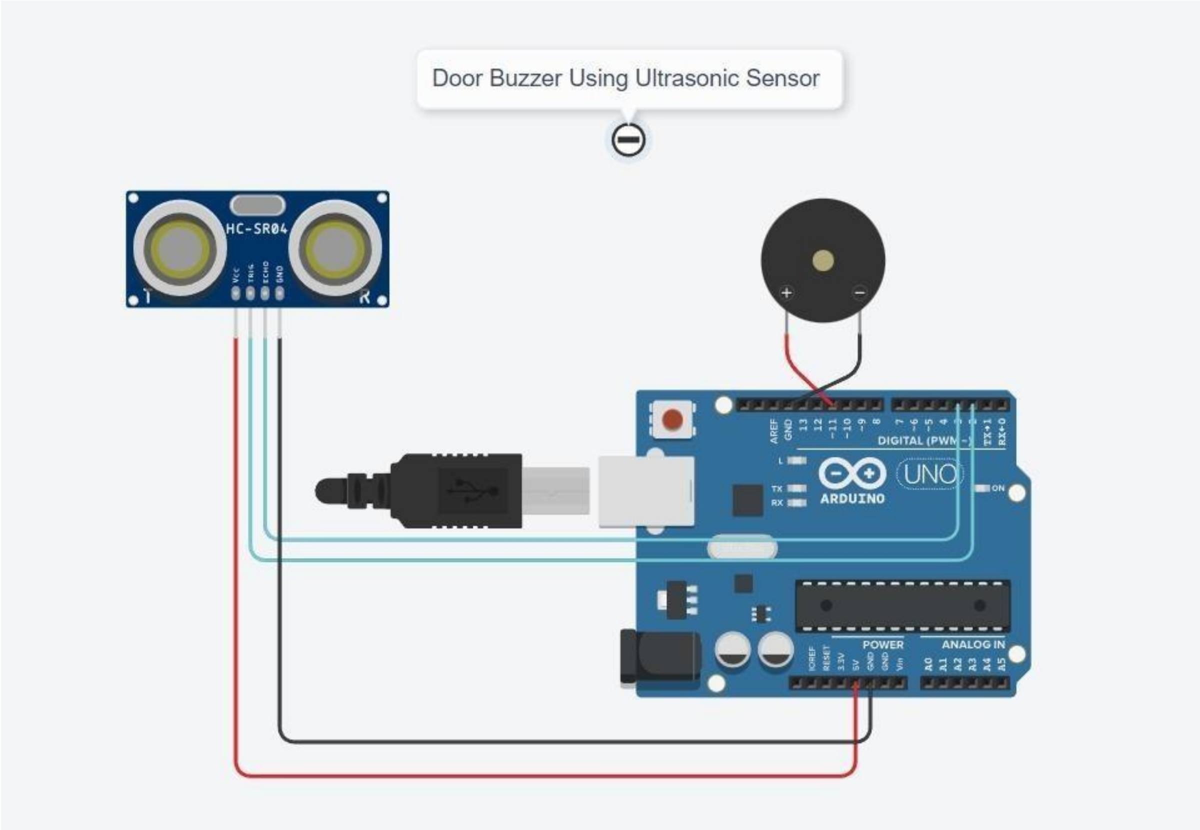
HARDWARE COMPONENTS USED IN CIRCUIT

1. 1 x ARDUINO UNO R3
2. 1 x HC-SR04 ULTRASONIC SENSOR
3. 1 x BUZZER
4. 1 x CONNECTING WIRES
5. 1 x BREADBOARD

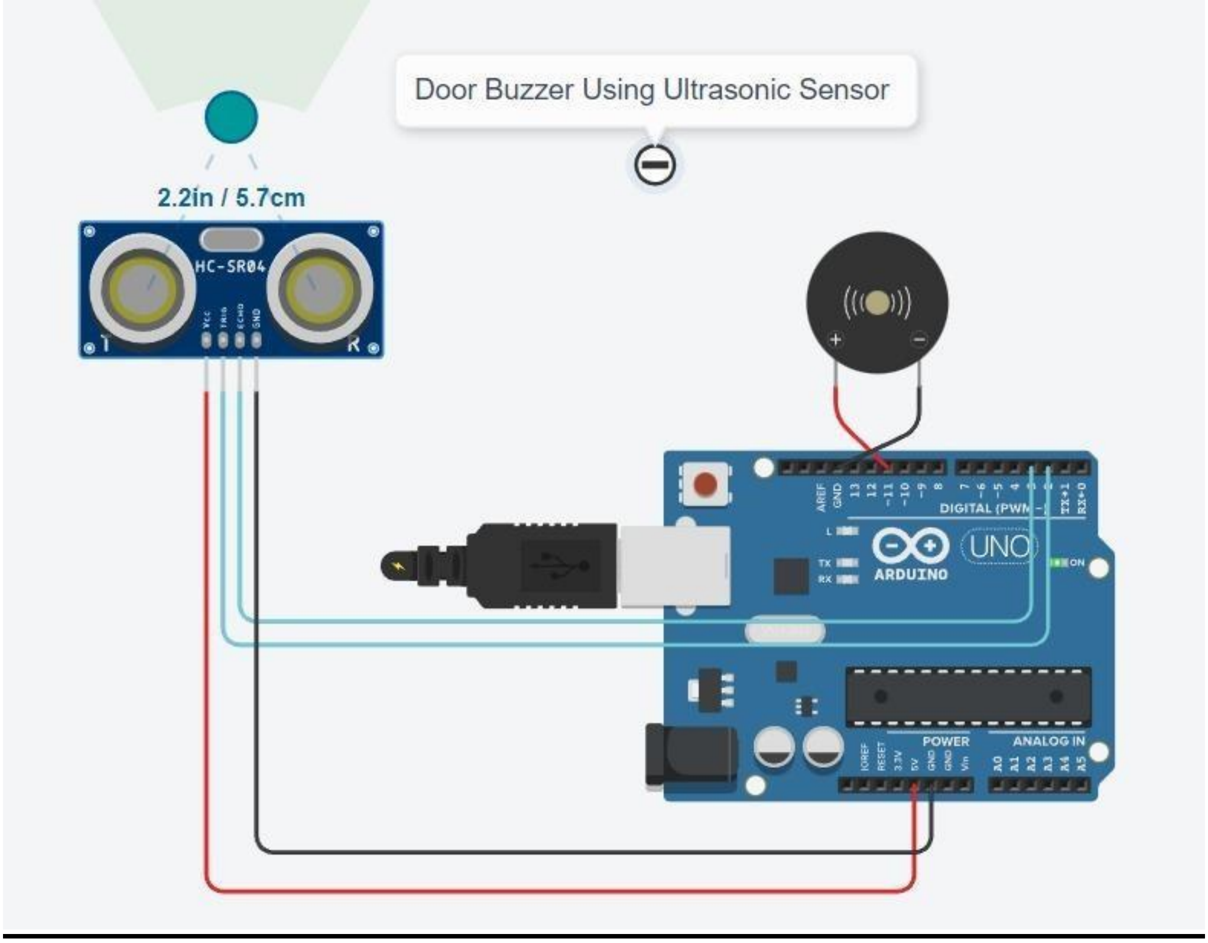
BLOCK DIAGRAM



Tinkercad Simulation Circuit



Working



Code

```
int trigger_pin = 2;
int echo_pin = 3;
int buzzer_pin = 11;
int time; int
distance;  void
setup()  {

        Serial.begin (9600);
pinMode (trigger_pin, OUTPUT);
pinMode (echo_pin, INPUT);
pinMode (buzzer_pin, OUTPUT);
```

```
} void loop()  {     digitalWrite
(trigger_pin, HIGH);
delayMicroseconds (10);
digitalWrite (trigger_pin, LOW);
time = pulseIn (echo_pin, HIGH);
distance = (time * 0.034) / 2;
  if (distance <= 10)

        {

        Serial.println (" Door Open ");

        Serial.print (" Distance= ");
Serial.println (distance);
digitalWrite (buzzer_pin, HIGH);
delay (500);         }   else {

        Serial.println (" Door closed ");

        Serial.print (" Distance= ");
Serial.println (distance);
digitalWrite (buzzer_pin, LOW);
        delay (500);

       }

}
```

SIGNAL CONDITIONING

Most analog signals require some form of preparation before they can be digitized. Signal conditioning is the manipulation of a signal in a way that prepares it for the next stage of processing. Many applications involve environmental or structural measurement, such as temperature and vibration, from sensors. These sensors, in turn, require signal conditioning before a data acquisition device can effectively and accurately measure the signal.

For example, thermocouple signals have very small voltage levels that must be amplified before they can be digitized. Other sensors, such as resistance temperature detectors (RTDs), thermistors, strain gages, and accelerometers, require excitation to operate. All of these preparation technologies are forms of signal conditioning.

What Do Signal Conditioners Do?

Data acquisition systems need to connect to a wide variety of sensors and signals in order to do their job. Signal conditioners take the analog signal from the sensor, manipulate it, and send it to the ADC (analog -to-digital converter) subsystem to be digitized for further processing (usually by computer software).

As the name implies, they are in the business of conditioning signals so that they can be converted into the digital domain by the A/D subsystem, and then displayed, stored, and analysed.

After all, you cannot directly connect 500V to one of the inputs of an A/D card - and thermocouples, RTDs, LVDTs, and other sensors require conditioning to operate and to provide a normalized voltage output that can be input into the A/D card.

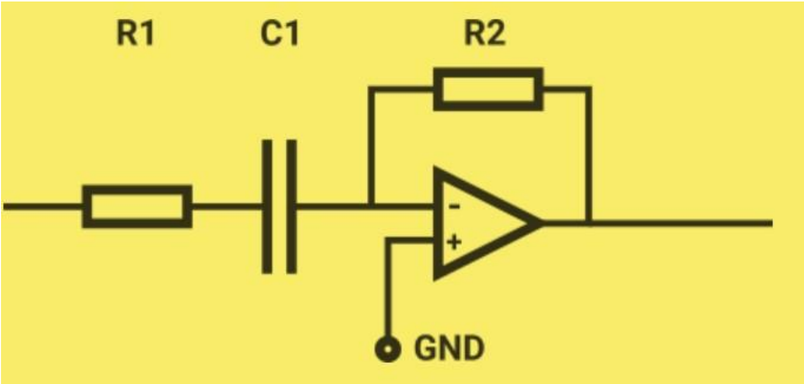
Signal Conditioners Types

In the world of electric sensors, we need different types of signal conditioning circuits in order to properly condition signals coming out from those sensors. Today common types of signal conditioners are:

- Voltage and high-voltage signal conditioners
- Current signal conditioners
- IEPE signal conditioners (or ICP/piezoelectric signal conditioners)
- Charge signal conditioners
- Strain gauge signal conditioners
- Load cell signal conditioners
- Thermocouple signal conditioners
- RTD signal conditioners
- Thermistor signal conditioners
- LVDT signal conditioners
- AC signal conditioning
- DC signal conditioning
- Digital signal conditioners

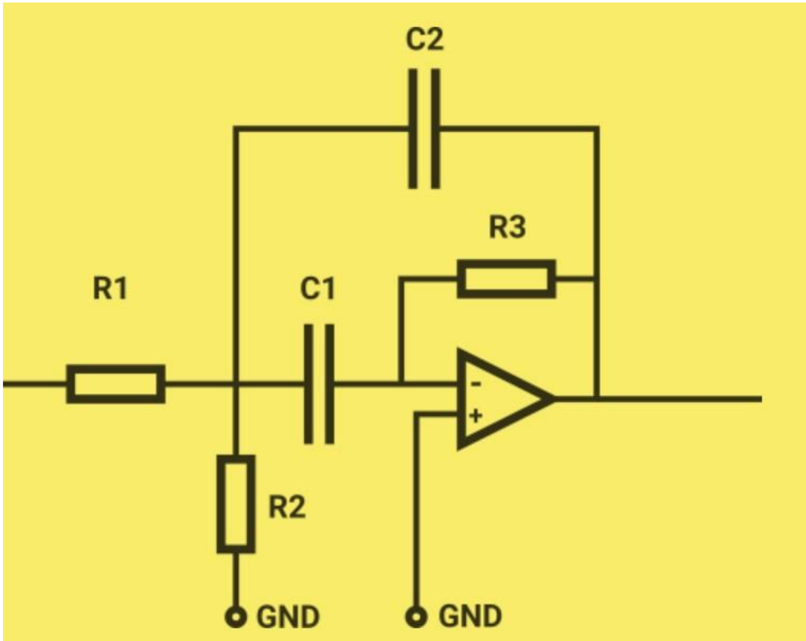
Here the signal conditioning circuit purpose is to filter DC signals and amplify the received ultrasound energy. We achieve this by using an inverting high pass filter and a band pass filter .

Inverting high pass filter

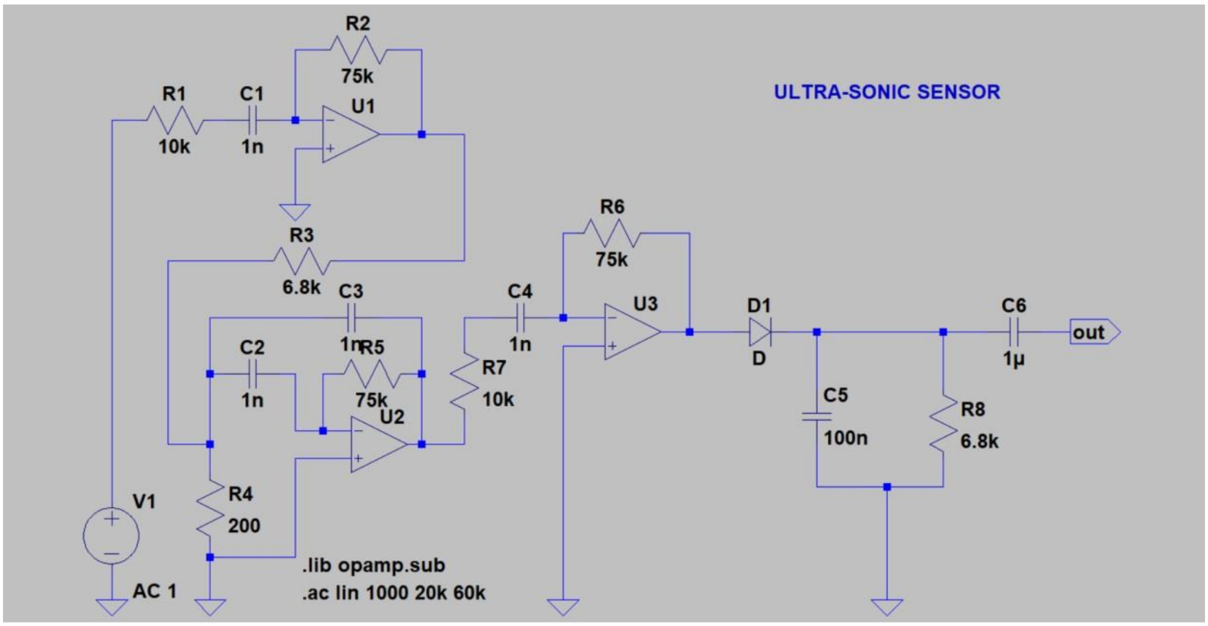


In first order high pass, meaning that it reduces lower frequencies while letting higher frequencies pass. First order comes from the fact that there is only one element in there that influences the frequency response, namely the capacitor.

Bandpass

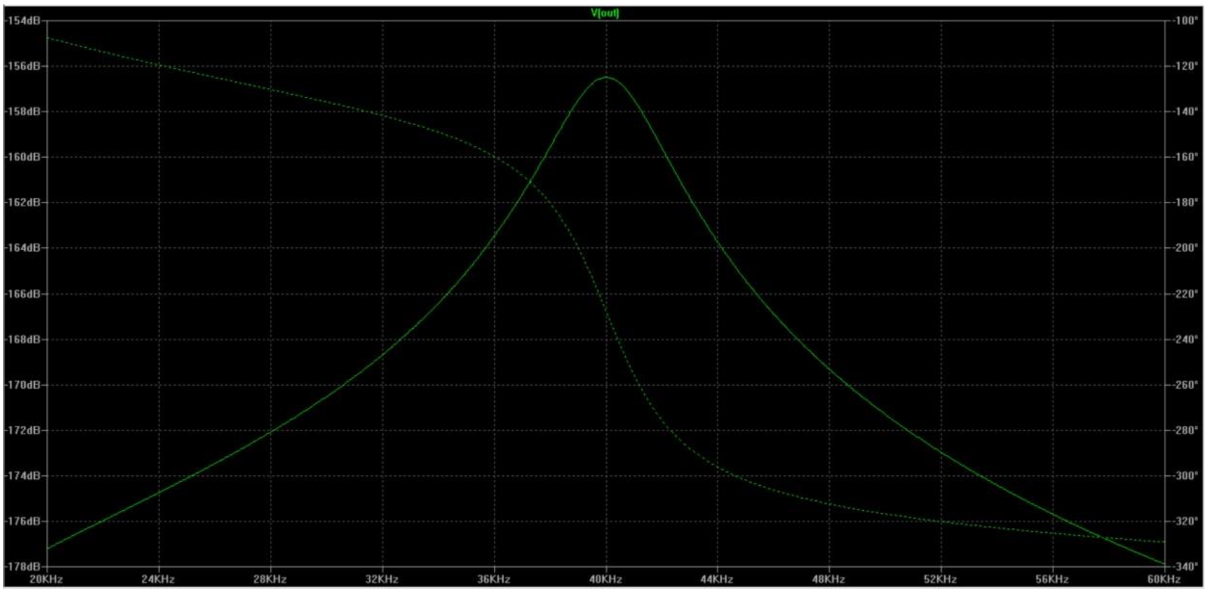


The bandpass amplifier amplifies frequencies around a specific center frequency and reduce everything else that's further away.
LTSPICE



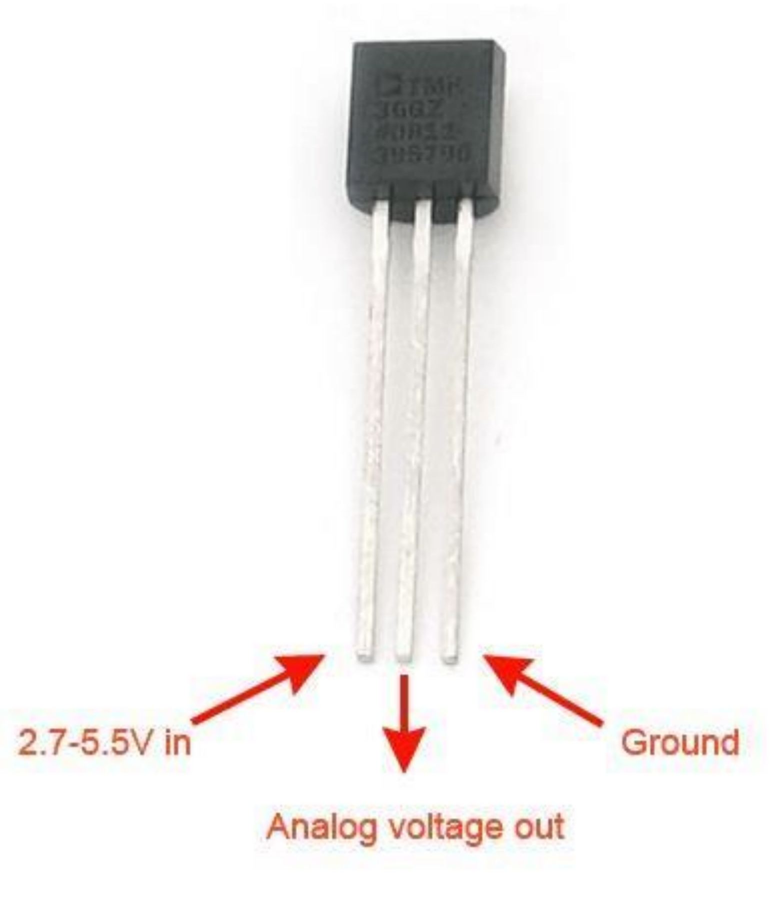
GRAPH OBTAINED

DB VS Vout



AUTOMATIC ROOM TEMPERATURE CONTROLLER

An analog temperature sensor is pretty easy to explain, its a chip that tells you what the ambient temperature is.
These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetallic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors). Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor.) By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.



Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistant between sensors and readings. Moreover they are very inexpensive and quite easy to use.

Its very similar to the LM35/TMP35 (Celsius output) and LM34/TMP34 (Farenheit output). The reason we went with the '36 instead of the '35 or '34 is that this sensor has a very wide range and doesn't require a negative voltage to read sub-zero temperatures. Otherwise, the functionality is basically the same.

- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw

Measuring Temperature

Using the TMP36 is easy, simply connect the left pin to power (2.7-5.5V) and the right pin to ground. Then the middle pin will have an analog voltage that is directly proportional (linear) to the temperature. The analog voltage is independent of the power supply.

To convert the voltage to temperature, simply use the basic formula:

Temp in °C = [(Vout in mV) - 500] / 10

So for example, if the voltage out is 1V that means that the temperature is
((1000 mV - 500) / 10) = 50 °C

If you're using a LM35 or similar, use line 'a' in the image above and the formula:
Temp in °C = (Vout in mV) / 10

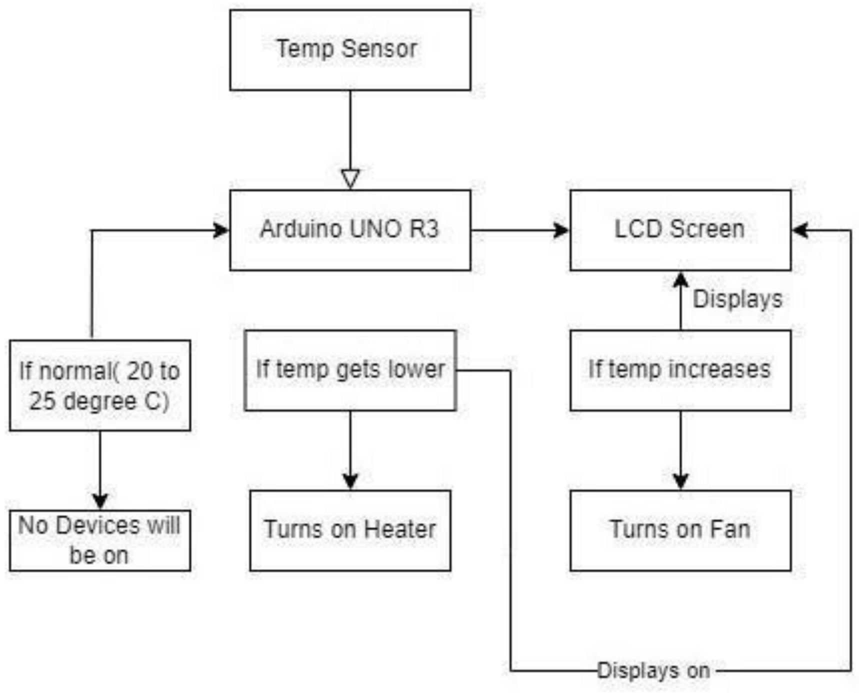
TinkerCad Simulation

HARDWARE COMPONENTS USED IN CIRCUIT

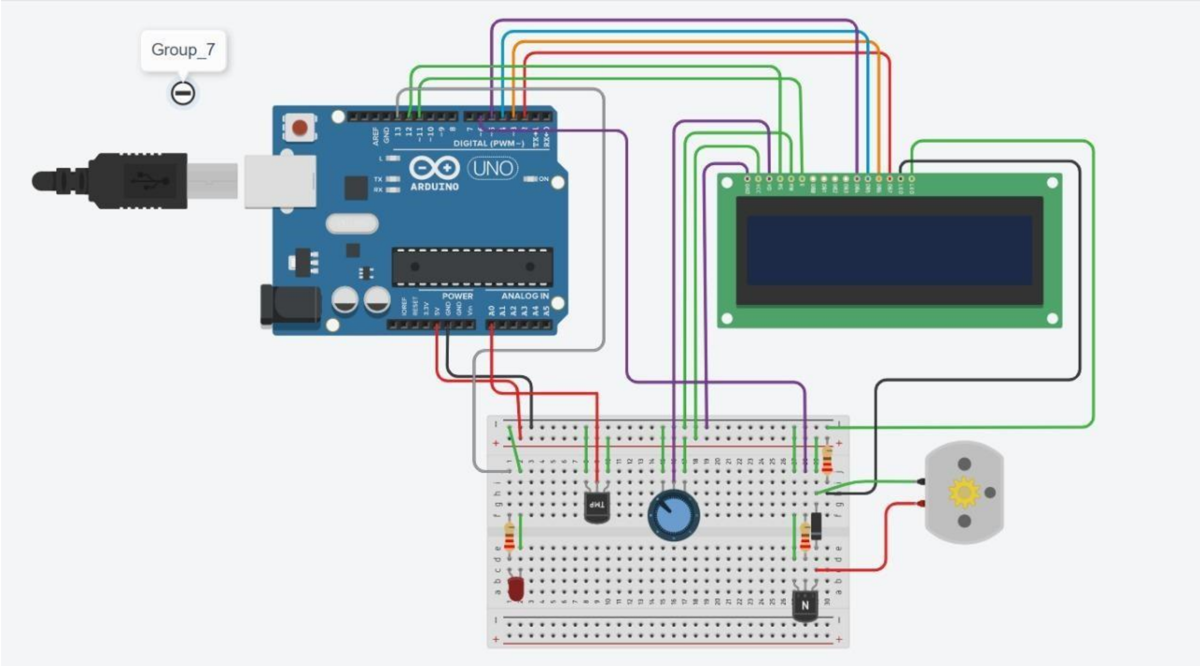
1. 3 x RESISTOR
2. 1 x LED(FOR DEMONSTRATING HEATER)
3. 1 xLCD
4. 1 x NPN TRANSISTOR
5. 1 x HIGH RATED,HIGH VOLTAGE CURRENT DIODE

6. 1 x DC MOTOR(REPRESENTING FAN)
7. 1 x AN ARDUINO UNO R3
8. 1 x TEMPERATURE SENSOR
9. A POTENTIOMETER
10. CONNECTING WIRES

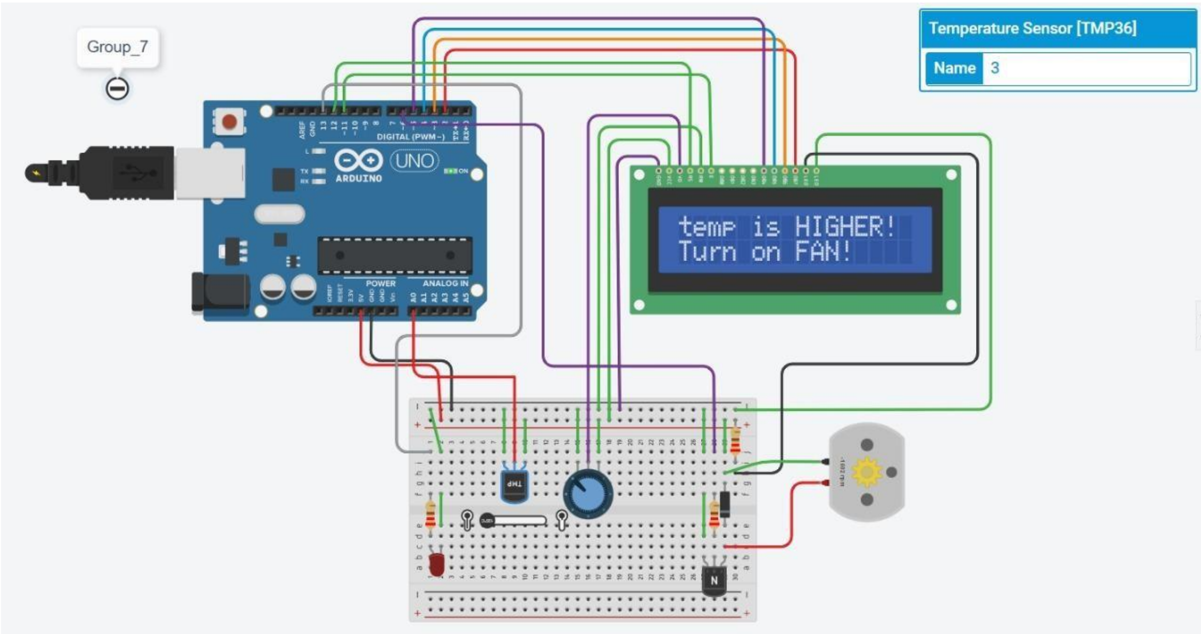
BLOCK DIAGRAM



Circuit



Working



Code:

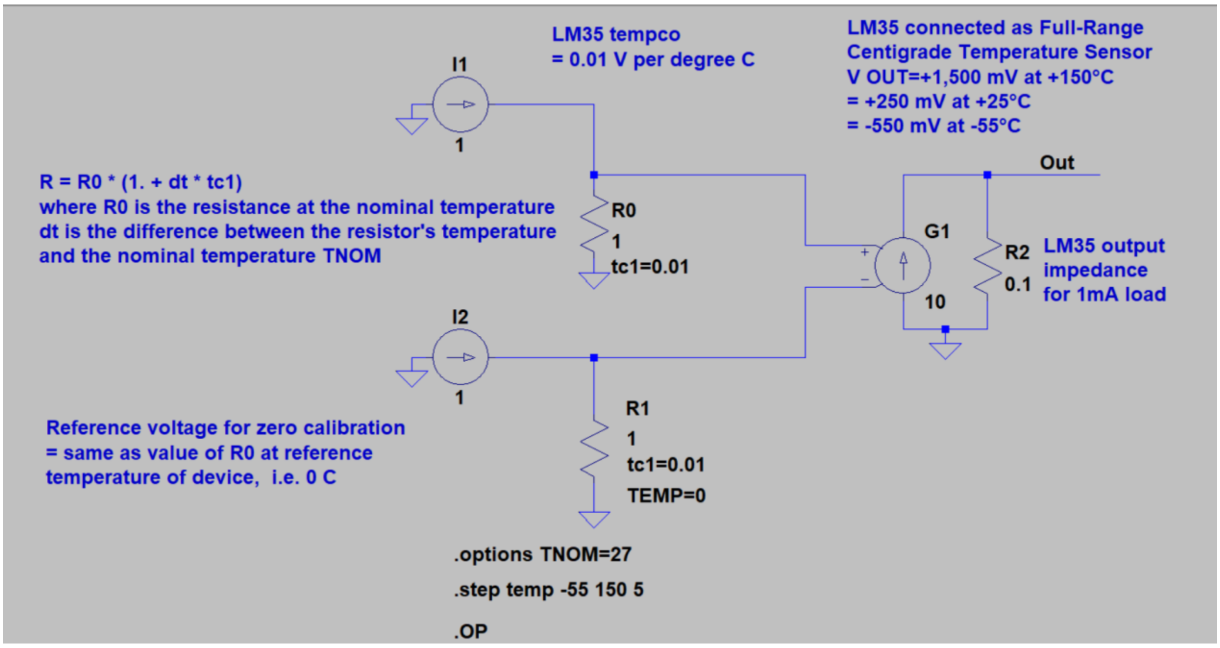
```
// C++ code
//
```

```
// Declare/assign Arduino IO-pins

 const int temp_trans_pin = A0, Heater_pin = 13, FAN_pin = 6;
/*FAN_pin: here I used DC motor in stead of FAN because
 I couldn't find the symbol for it. Similarly,for the
  Heater (Heater_pin), I used LED.*/

// Set the range of the desired temperature

  float MinTemp = 20, MaxTemp = 25;/*Room temperature is [20,25] degree C */

// Include the LCD library code

 #include <LiquidCrystal.h>

 // Initialize the library with the numbers of the interface pins

 LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

 void setup() {

 // System initialization

   LCD.begin(16, 2);
   pinMode(Heater_pin, OUTPUT);//LED in our case
pinMode(FAN_pin, OUTPUT);

 // Display the desired range of temperature

   LCD.print("Room temp(C):");
   LCD.setCursor(2,1);
   LCD.print(MinTemp); LCD.print("-");LCD.print(MaxTemp);

   delay(2000);
 }

 void loop() {

  float Eqv_volt, SensorTemp;

 // Read voltage and convert to temperature (Celsius)

  Eqv_volt = analogRead(temp_trans_pin) * 5.0 / 1023;
```

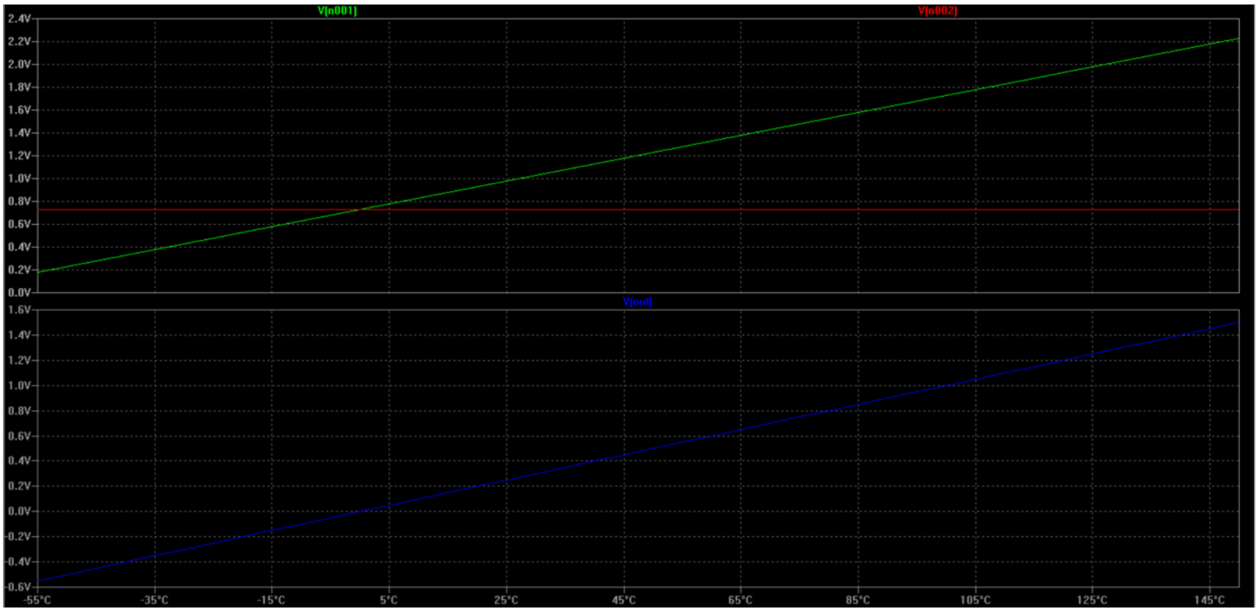
```
  SensorTemp = 100.0 * Eqv_volt-50.0;

 // Display the sensor reading

   LCD.clear();
   LCD.print("Sensor reading:");
   LCD.setCursor(2,1);
   LCD.print(SensorTemp); LCD.print(" C");

   delay(2000);

 /*Compare the sensor reading with the range of
acceptable temperatures*/

  if(SensorTemp > MaxTemp){
    LCD.clear();
    LCD.print("temp is HIGHER!");//higher than the max

    /*Turn on FAN (dc motor)! to regulate the temp.
     Increase FAN speed at a slow rate*/

    LCD.setCursor(0, 1);LCD.print("Turn on FAN!");
for( int i = 0; i <= 255; i++ ) {
      analogWrite(FAN_pin, i);
     }
     delay(2000);

     LCD.clear();
     LCD.print("Now temp is OK!");
     LCD.setCursor(0, 1);
     LCD.print("Turn off FAN!");

 // Turn off FAN slowly
     for( int i = 255; i >= 0; i-- ) {
      analogWrite(FAN_pin, i);
     }
      delay(2000);
     }
 else if(SensorTemp < MinTemp){
    LCD.clear();
    LCD.print("temp is LOWER!");//Less than the mini
    LCD.setCursor(0, 1);
    LCD.print("Turn on HEATER!");
```

```
   //Turn the heater ON, LED in our case

    digitalWrite(Heater_pin, HIGH);

    delay(3000);

    LCD.clear();
    LCD.print("Now temp is OK!");
    LCD.setCursor(0, 1);
    LCD.print("Turn off HEATER!");

    delay(1000);

    digitalWrite(Heater_pin, LOW);
    LCD.clear();
    }
  else if(SensorTemp > MinTemp && SensorTemp < MaxTemp){/*Now temperature is
perfect.
     That is,it is in the desired range. Hence no need of changes!!*/
    LCD.clear();
    LCD.print("Temp is NORMAL!");LCD.setCursor(2,1);
    LCD.print("Turn off all!");

    delay(1000);
    LCD.clear();
  }
  else {
    LCD.clear();
    LCD.print("Something went");
    LCD.setCursor(2,1); LCD.print("WRONG in the ckt");
delay(1000);
    LCD.clear();
   }
   delay(1000);
  }
```

Temperature Sensor circuit

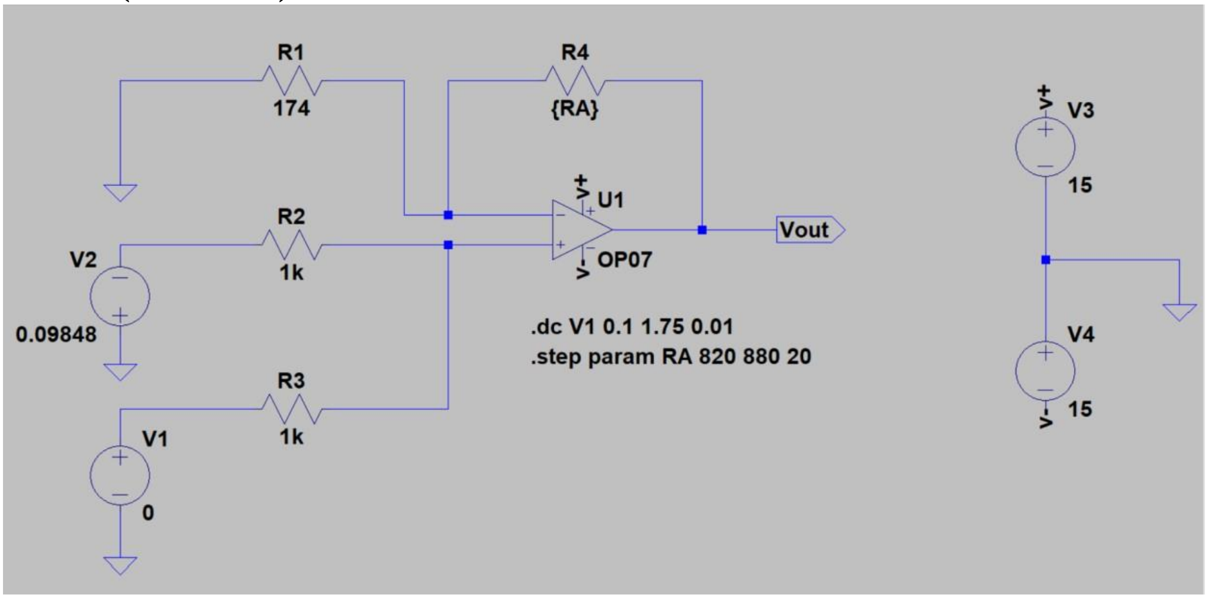


Graph Obtained



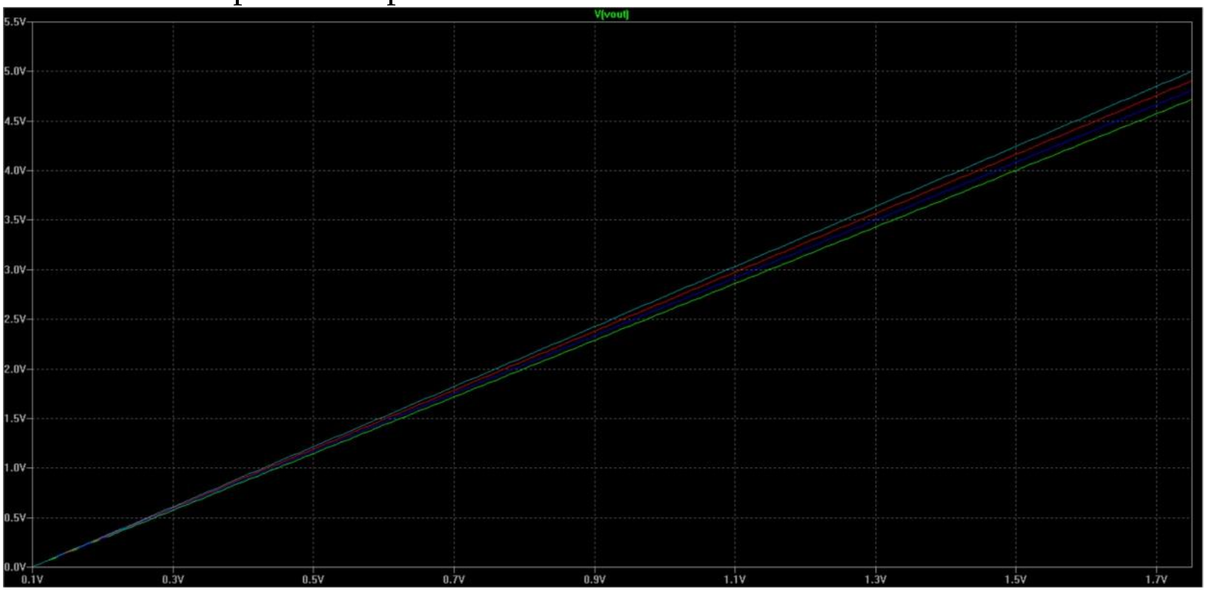
Signal conditioning circuit used

The output of the temp sensor is conditioned to the input parameters of the Arduino(0V to 5V)



Graph obtained

Conditioned output of temp sensor



Conclusion

We have successfully implemented Door Buzzer Using Ultrasonic Sensor and Automatic Room temperature controller using Temperature Sensor.
We have also gotten a better understanding of the use of the Ultrasonic sensor and the Temperature sensor. We saw the working of each sensor and their applications in various fields.

The outcome of this project is considered as the best understanding of basic and advanced mode of connections, codes, the working of sensors and years of innovation lead to the discovery of such extremely beneficial sensors to mankind. If we keep the spirit of innovation burning within us, we can integrate these existing sensors with new technology to create even more beneficial technology.

Bibliography

https://dewesoft.com/daq/what-is-signal-conditioning#introduction

https://colinjs.com/elec/tempsens/tempsens.htm

https://www.fierceelectronics.com/sensors/what-ultrasonicsensor#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20hu mans%20can%20hear).