

BUILDING SECURITY SYSTEM USING ULTRASONIC SESNOR, FLAME SENSOR & PIR SENSOR

Course Title: Sensing and Sensor Technology

Nooruddin : 6218000081

Rahul : 6217000181

Jannatun Ferdous : 6218000014

Course Instructor:

Tianjin University

Department of Software Engineering

**Introduction:**

Security has always been a major concern for all of us and there are many Hi tech and IoT based security and surveillance system are available in the market. We were inspired to build an ultrasonic security system for our project by building security system. Security is very import part of building. The main modules are: detection module, attack early warning module, intelligent light module and fire alarm module.

**Main Modules:**

1. **Detection module:**

Through our tests, we believe our system is capable of detecting intruders of the time, provided that sensor was placed at appropriate position. The few times that intruder get away are when they are capable of crossing past the sensor quicker

1. **Warning module:**

In this module we use passive infrared sensor PIR to detect something when the human body can touch them. We can this application in many real world application. The PIR infrared sensor used to collect signals and then it can alarm when the body can touch it. We can use this sensor for security issues.

1. **Light module:**

The main function of this module is to turn on the lights or the lighting of the system and any type of alert occur from any sensor it may alarm and lightened. This can occur in any crucial time and we can understand by led light and by their alarm

1. **Fire alarm module:**

In this scenario, a flame alarm is used to simulate the fire alarm and, once there is fire, the buzzer and LED alarms are immediately penalized. We can use this sensor in many real world application and also for their security. This can be happened by detecting fire and then alarm by buzzer sound and LED light.

**Components and Languages:**

**Components:**

1. Ultrasonic Sensor
2. Passive Infrared Sensor(PIR)
3. LED Light
4. Buzzer
5. Light Module
6. Arduino nano kit
7. Bread Board

**Languages:**

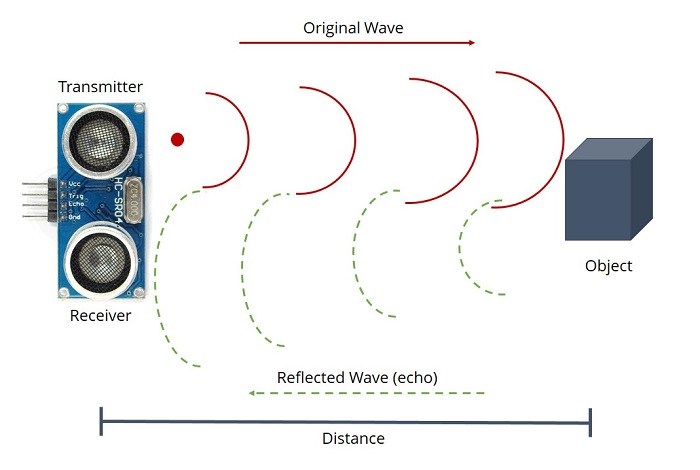
1. C# and VS15
2. Arduino 1.8.7

**Implementation:**

**Distance Measure using Ultrasonic Sensor:**

Ultrasonic Sensor:

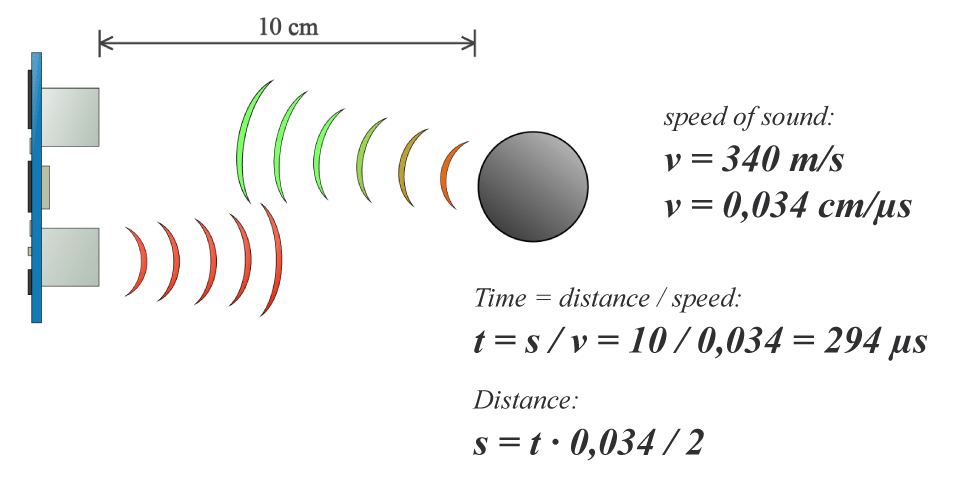
It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.



The 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.

In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.

For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward.  So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.



**FIRE Alarm Using Flame Sensor:**

FLAME SENSOR:

The required kits are :

1. An arduino and its adapter
2. Few jumper Wires
3. A fire sensor
4. A led
5. A breadboard
6. A buzzer
7. A lighter

There are 4 pins on the digital sound sensor. They are :

1. **AO**-Analog pin
2. **GND**-Ground pin
3. **VCC**-positive power pin
4. **DO**-digital output pin

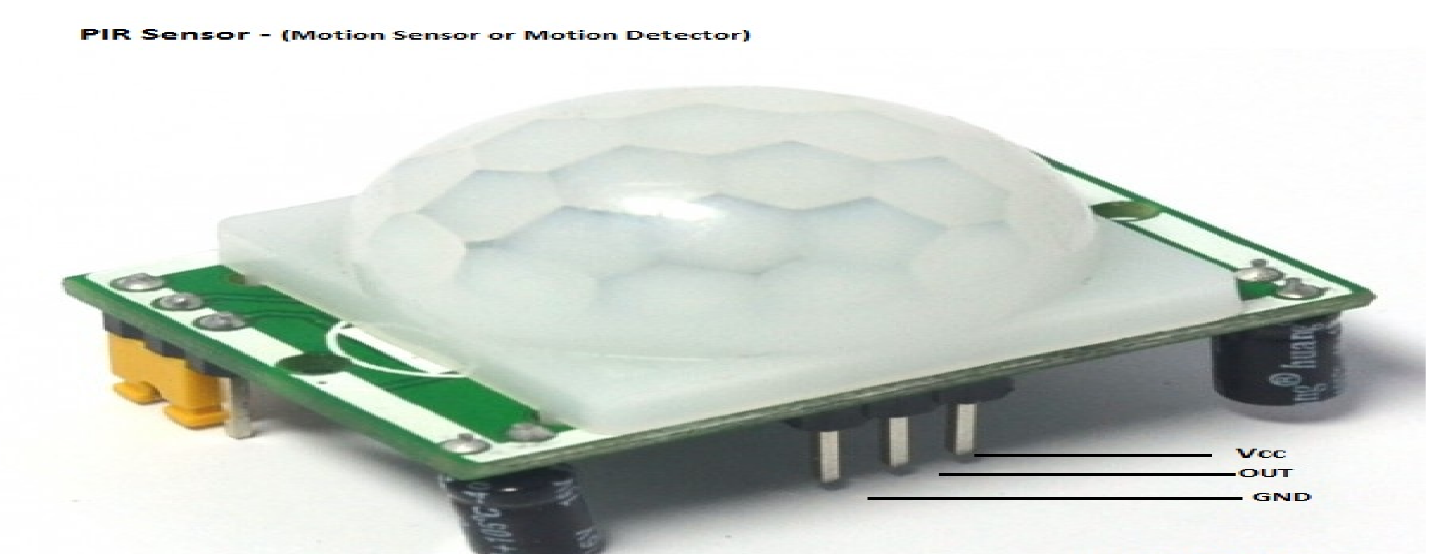
There are two legs on a LED. The longer leg is positive and the shorter leg is negative. A buzzer has similar legs. The steps to setup hardware are listed below:

* Connect **"VCC" pin of the sensor**to 5V pin on the arduino.
* Connect **"GND" ground pin**to GND pin on the arduino.
* Connect **"AO" analog output pin**to any analog or digital pins on the arduino.In this tutorial I connected it with A2
* Connect **longer LED leg**to pin 13 on the arduino. If you want to use anyother pin on the arduino.
* Connect **shorter LED leg**to GND pin on the arduino.
* FIx buzzer on the breadboard.
* Connect **longer buzzer leg**to pin 11(or any other pin) on the arduino.
* Connect **shorter buzzer leg**to GND pin on the arduino.

**Motion Sensing Using PIR:**

PIR:

The module has just three pins, a Ground and a VCC for powering the module and an output pin which gives high logic level if an object is detected. Also it has two potentiometers. One for adjusting the sensitivity of the sensor and the other for adjusting the time the output signal stays high when object is detected. This time can be adjusted from 0.3 seconds up to 5 minutes.



The module has three more pins with a jumper between two of them. These pins are for selecting the trigger modes. The first one is called “non-repeatable trigger” and works like this: when the sensor output is high and the delay time is over, the output will automatically change from high to low level. The other mode called “repeatable trigger” will keep the output high all the time until the detected object is present in sensor’s range.

**Final Complete Project Code:**

// Flame Sensor Code

const int ledpin=2; // ledpin,flamepin and buzpin are not changed throughout the process

const int flamepin=A2;

const int buzpin=13;

const int threshold=300;// sets threshold value for flame sensor

int flamesensvalue=0; // initialize flamesensor reading

// UltraSonic Sensor Code

const int trigPin = 9;

const int echoPin = 10;

// defines variables

long duration;

int distance;

void setup() {

//Flame Sensor Code

Serial.begin(9600);

pinMode(ledpin,OUTPUT);

pinMode(flamepin,INPUT);

pinMode(buzpin,OUTPUT);

//UltraSonic Sensor Code

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

//Flame Sensor Code

flamesensvalue=analogRead(flamepin); // reads analog data from flame sensor

if (flamesensvalue<=threshold) { // compares reading from flame sensor with the threshold value

digitalWrite(ledpin,HIGH); //turns on led and buzzer

Serial.println("High");

delay(1000); //stops program for 1 second

tone(buzpin,100);

}

else{

digitalWrite(ledpin,LOW); //turns led off led and buzzer

Serial.println("Low");

noTone(buzpin);

}

//UltraSonic Sensor Code

// 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;

// Prints the distance on the Serial Monitor

Serial.println(distance);}

**Code for Motion Sensor:**

const int motionpin=A2;

const int ledpin=2;

const int buzzpin=13;

int motionsensvalue=0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(ledpin, OUTPUT);

pinMode(motionpin,INPUT);

pinMode(buzzpin,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

motionsensvalue=analogRead(motionpin); // reads analog data from motion sensor

if (motionsensvalue>=200){

digitalWrite(ledpin,HIGH);

Serial.println("Someone");

tone(buzzpin,100);

}

else {

digitalWrite(ledpin,LOW); //turns led off led and buzzer

Serial.println("No One");

noTone(buzzpin);

}

}

**Final Complete Project Code of C#:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO.Ports;

using System.Threading;

using System.Text.RegularExpressions;

namespace ProjectFinal

{

public partial class Form1 : Form

{

SerialPort port;

public Form1()

{

InitializeComponent();

port = new SerialPort("COM3", 9600);

port.Open();

}

private void button1\_Click(object sender, EventArgs e)

{

/\*for (int i = 1; i <= 100; i++)

{

Thread.Sleep(9000);

textBox1.Text += "Rahul";

}\*/

/\*for (int i = 1; i<= 100; i++) {

string abc = "Hello\nThere\nFriend";

textBox1.Text += abc;

}\*/

/\*while (true) {

string abc = "Hello\nThere\nFriend";

textBox1.Text += abc;

}\*/

string distance = port.ReadExisting();

// Split on one or more non-digit characters.

string[] numbers = Regex.Split(distance, @"\D+");

foreach (string value in numbers)

{

if (!string.IsNullOrEmpty(value))

{

int i = int.Parse(value);

label3.Text = i + "";

}

}

bool result = Regex.IsMatch(distance, "\\bHigh\\b");

if (result)

{

pictureBox1.Image = Properties.Resources.fire1;

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

else {

pictureBox1.Image = Properties.Resources.nofire;

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

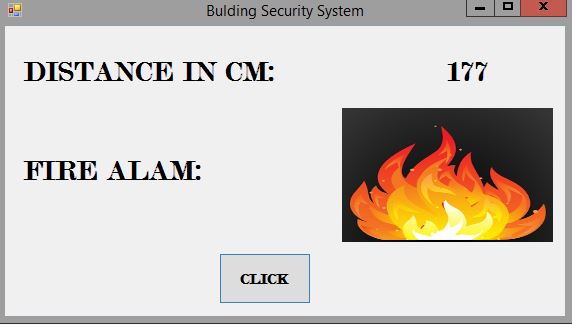
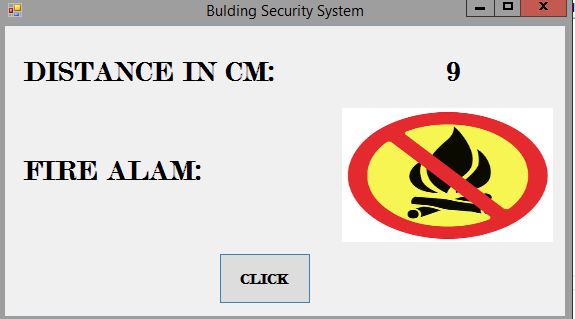
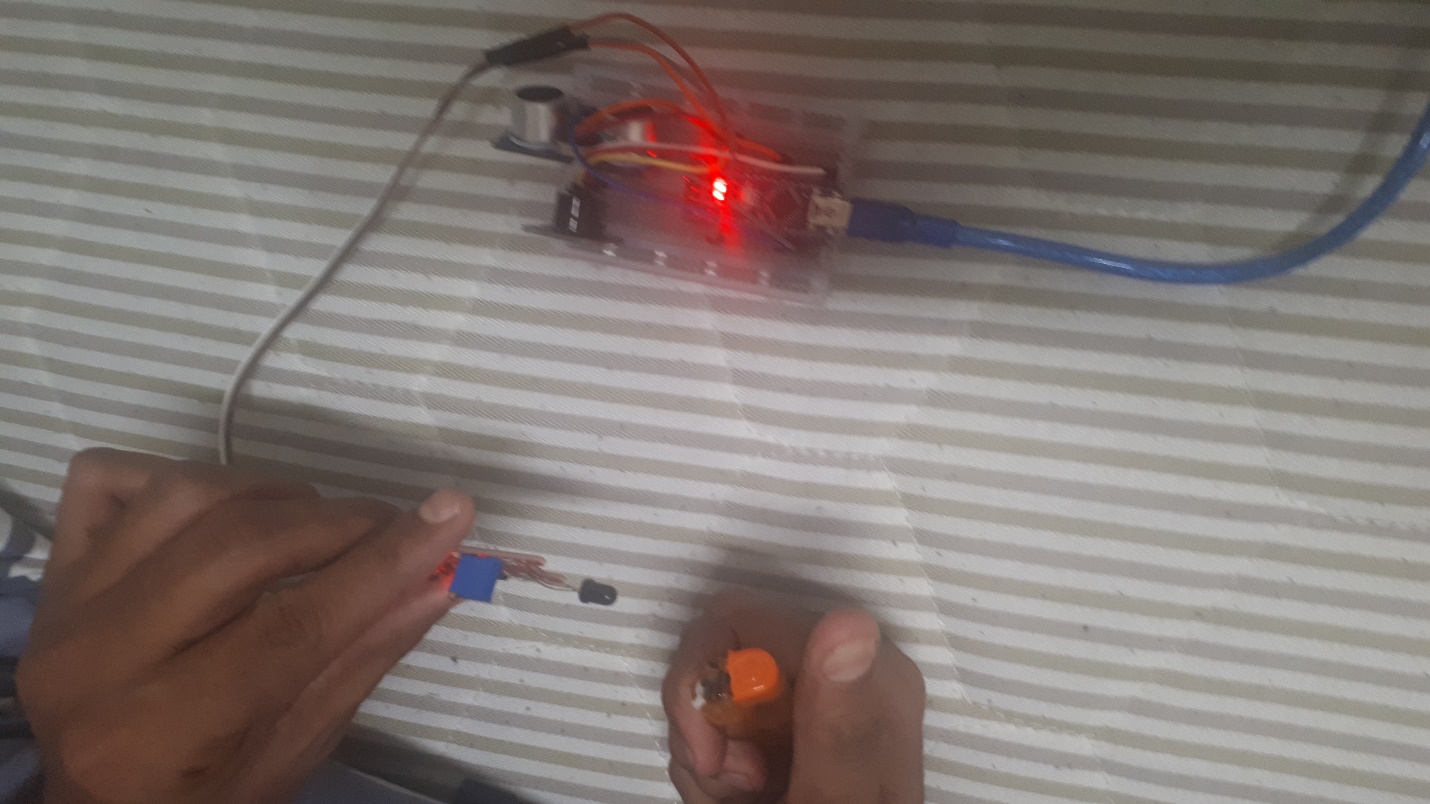
}

}

}

**OUTPUT:**

****

****

**Conclusion:**

We proposed this for security issues especially in any buildings to secure from some harmful activities this system in many real world application like any building security. We have proposed a security alert system we can use them any purpose.