Fundamentals of Mechatronics

Home Automation Project

Submitted To: Sir Tayyab Shahid

Submitted By: Aqib Habib

CMS:266993

Rime Section 2

Table of Contents

Table of Figures:

Chapter 1. Introduction

Home security is a necessity in today's unpredictable world and due to advancement in sensor and automation technology it is now possible integrate automation technology in our homes.

In this project, design of fire alarm system and home security system has been discussed in details and different project aspects has been explained.

Chapter 2. Home Security System

During the design of home security system certain factors were kept in mind such as how to cater for false alarms, and how to improve the overall accuracy of the system.

Two main sensors that are popular for home security are PIR and Ultrasonic proximity sensors. These two sensors are used in fusion to cater for any false alarms.

Figure 1 HC-SR04 Ultrasonic Sensor

Figure 2 PIR Sensor

Besides these two sensors, a buzzer was used for alert when the theft is detected and an LCD was also used to display the distance between the theft and the location of the placement of the sensor.

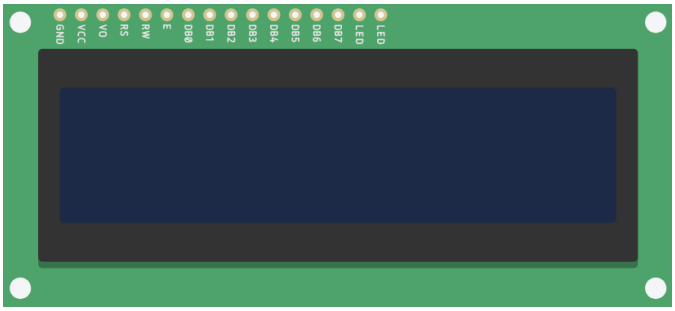

Figure 3 Arduino LCD

Figure 4 Arduino Buzzer

The overall circuit design was done using Autodesk TinkerCAD because all the integrated electronics of Arduino is already present in tinkercad and simulation can also be performed to demonstrated how the circuit will work.

The overall circuit is shown below:

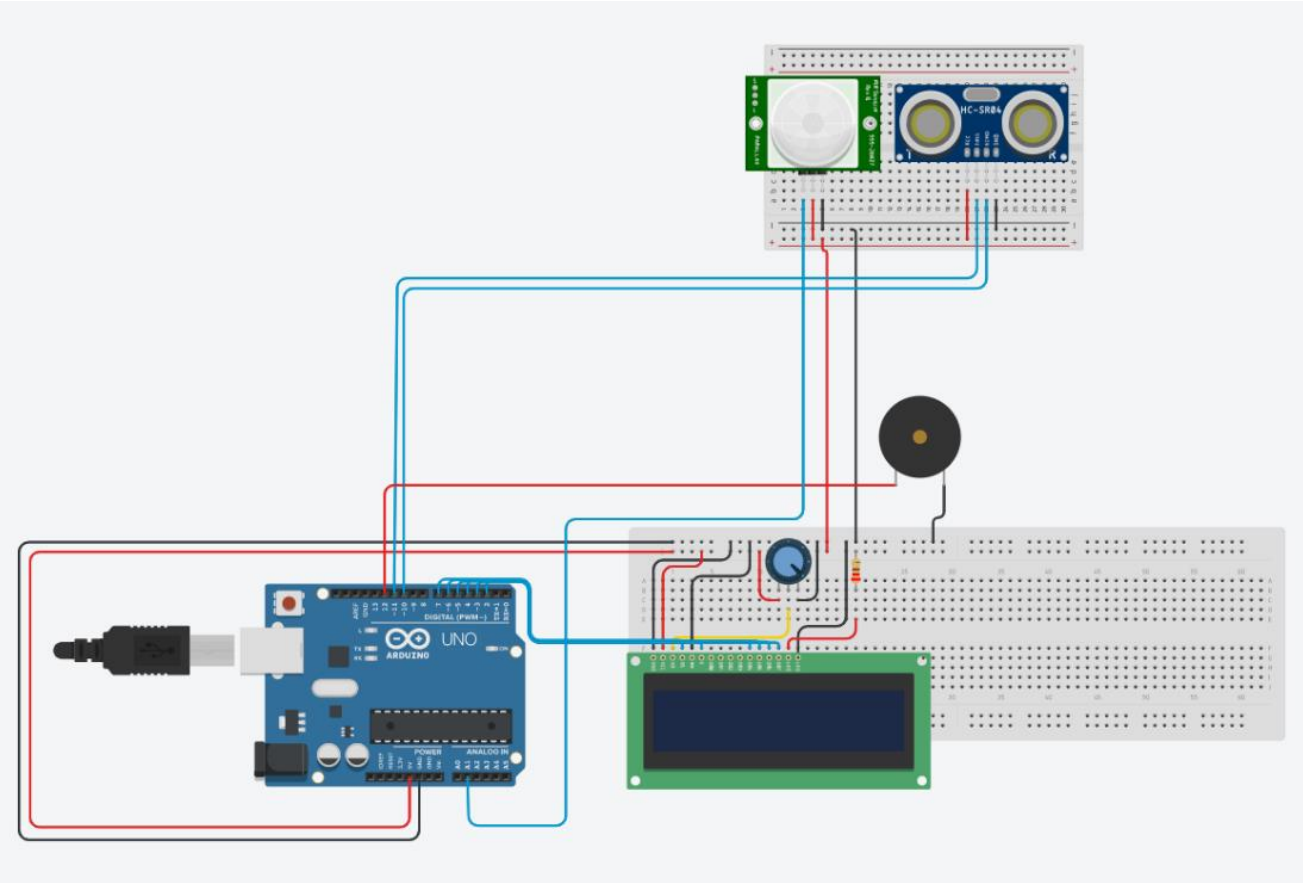

Figure 5 Home Security System

The circuit can be accessed here:

2.1 Working of the Circuit and System:

2.1.1 LCD Screen:

The LCD screen has been used with a potentiometer to control the contrast of the screen. The LCD screen basically performs two functions, one it displays the distance between the sensor and the immediate object placed in front of it and it displays a message “Motion Detected!”.

2.1.2 Ultrasonic Sensor:

The ultrasonic sensors sense the distance between the immediate objects using sound waves transmitter and receiver. The speed of sound was assumed to be 340 m/s but it varies due to temperature so some error can occur while assuming the speed of sound. It can be rectified by using a more accurate system like integrating ultrasonic sensor with LM35 ambient temperature sensor and using the relation $\sqrt{KRT}$ to calculate the velocity of sound.

A threshold distance of 30cm has been taken for which the alarm system will go off if the PIR sensor is activated.

2.1.3 PIR Sensor:

The passive infrared sensor detects the thermal signature of the body and activates in a binary mode i.e., either it is active or it is not.

The PIR sensor is used side by side with the ultrasonic sensor so that when the distance of the immediate object is less than 30 cm and PIR sensor is active, then the alarm will go off indicating some human interference. The sensor was adjusted using potentiometers on the sensor it self and its range was decreased to 3 meters and active time to about 3-4 seconds.

Fused sensor system is used because of avoiding any false alarms due to pets/animals around the house.

The working demonstration video has been uploaded with this report or online simulation can be accessed here.

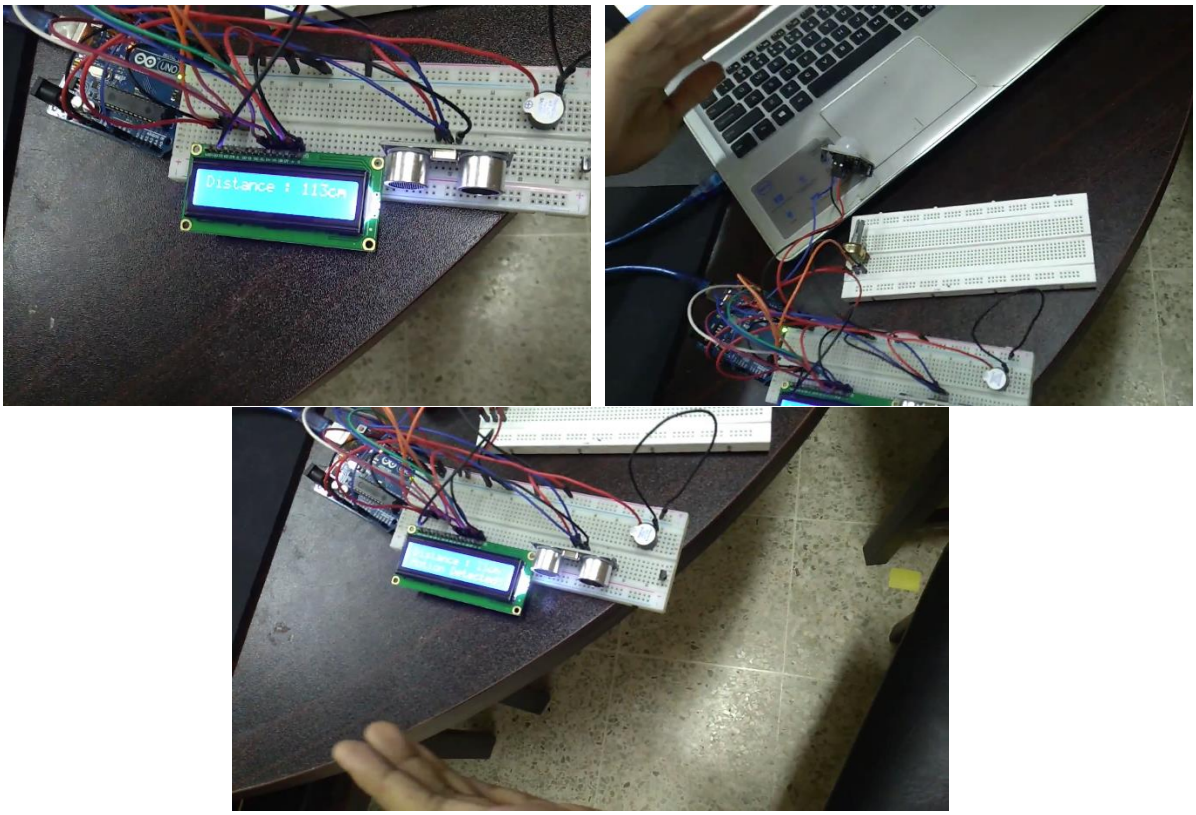

Figure 6 Circuit Testing

2.1.4 Arduino Code:

The full code has been attached in the Appendix I and some parts are explained here:

```
1  #include <LiquidCrystal.h>
2  LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
3  int ultrasonic_trigger_pin=10;
4  int ultrasonic_echo_pin=11;
5  int pin_pir_sensor_input=A1;
6  int pir_state;
7  int pin_buzzer_output=12;
8  int distance;
9  float time;
10 float velocity_sound=0.034;
```

Figure 7 Variables

From Line 1-10 initial variables were defined for sensor input, buzzer output and libraries were included such for lcd.

```
11 void setup()
12 {
13   pinMode(pin_buzzer_output,OUTPUT);
14     pinMode(ultrasonic_echo_pin,INPUT);
15   pinMode(ultrasonic_trigger_pin,OUTPUT);
16   pinMode(pin_pir_sensor_input,INPUT);
17   lcd.begin(16, 2);
18   Serial.begin(9600);
19  lcd.begin(16, 2);
```

Figure 8 Setup

From line 11-19, initial setup codes were written such as declaration of output and input and activation of computer serial and lcd screen.

```
21 void loop()
22 {
23 digitalWrite(ultrasonic_trigger_pin,HIGH);
24 delayMicroseconds(10);
25 digitalWrite(ultrasonic_trigger_pin,LOW);
26 time = pulseIn (ultrasonic_echo_pin, HIGH);
27 distance = (time * velocity_sound) / 2;
28 pir_state=digitalRead(pin_pir_sensor_input);
29 Serial.print("Distance: ");
30 Serial.print(distance);
31 Serial.println(" cm ");
32 lcd.setCursor(0,0);
33 lcd.print("Distance : ");
34 lcd.print(distance);
35 lcd.print("cm ");
36 delay(200);
37 if (pir_state==1) {
38 Serial.println("Motion is Detected!");
39 lcd.setCursor(0,1);
40 lcd.print("Motion Detected!");
41 }
42 else {
43 lcd.clear();
44 }
45 if((distance < 30 )&&(pir_state == 1))
46 digitalWrite(pin_buzzer_output,HIGH);
47 else
48 digitalWrite(pin_buzzer_output,LOW);
49 }
```

Figure 9 Main Security System Logic

From line 21-49 the main logic of the circuit is coded. Firstly, the ultrasonic sensors send a pulse for 10 microseconds and the distance is measured by the time it takes for the pulse to return and by using velocity of sound. Relative motion is not considered since it was not applicable at very low observer speeds. The next bit of code reads the PIR sensor state, i.e., whether it is active or not and based on that logic an if statement with two conditions is used so that when both are satisfied, the alarm goes off

2.1.5 Bill of Materials

The prices are based on actual value at the time of purchase of the sensors and hence all the prices are accurate.

Item	Price (RS)	Quantity
Arduino Uno	600	1
LCD Screen	350	1
Ultrasonic Sensor	280	1
PIR Sensor	250	1
Buzzer	30	1
Potentiometer	20	1
Total	1540	-

Chapter 3. Fire Alarm System

The fire alarm system was used with the integration of IR Flame sensor and MQ-2 smoke sensor. The circuit was tested using a lighter and threshold of the smoke sensor was lowered. Due to safety reasons and hostel rules, a more practical testing was not carried out.

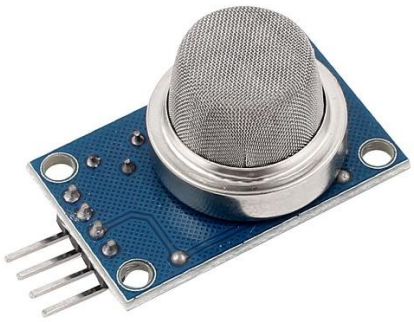

Figure 10 MQ-2 Smoke Sensor

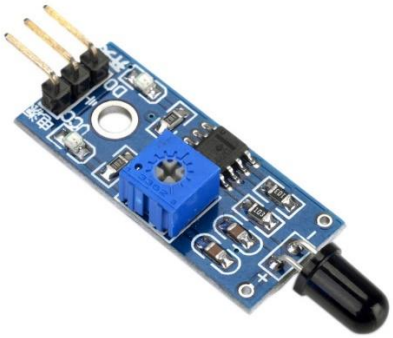

Figure 11 IR Flame Sensor

The gas sensor was integrated with the IR flame sensor to cater for any false alarms such some one smoking or during cooking.

The alarm only goes off, if fire is detected and smoke crossed a certain threshold value. Threshold values can be set by adjusting potentiometers on the sensors itself or during coding.

LCD screen and buzzer was also used to indicate the state of the alarm system. The LCD screen displays a message when fire and smoke is detected.

The overall circuit is shown below.

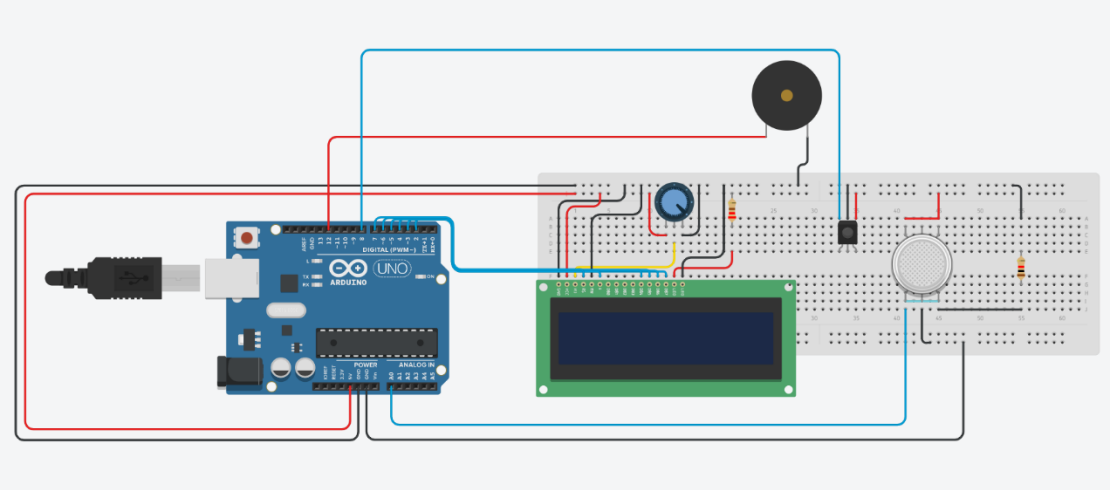

Figure 12 Circuit of Fire Alarm System

The circuit can be accessed here.

3.1 Working of the Circuit

3.1.1 LCD Screen:

Arduino LCD screen has been used with potentiometer to control its contrast.

The LCD screen is used to detect whether any smoke or flame is detected and in case if the flame and smoke is detected, it shows a message that “Fire Detected!”

3.1.2 MQ-2 Smoke Sensor:

The gas sensor outputs a specific voltage whenever the concentration of flammable gas such as methane, propane or butane is detected or if the concentration of carbon dioxide is increased and based on these voltages, we can set a threshold value to detect fires.

3.1.3 IR Flame Sensor:

The IR flame sensor works in a binary mode i.e., whether the flame is detected or not. The flame sensor is used along with the smoke sensor to cater for false alarms.

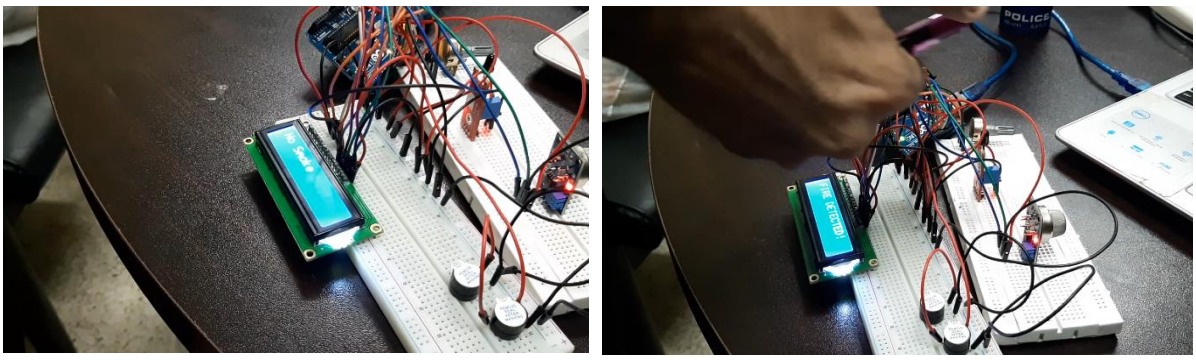

Figure 13 Testing of the Circuit

3.1.4 Arduino Code:

The full code has been attached in the Appendix I and some parts are explained here:

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
3 float gas_sensor_input;
4 int gas_sensor_pin=0;
5 int flame_sensor_pin=8;
6 int flame_sensor_state;
7 int buzzer_pin=12;
8 int threshold=50;
```

Figure 14 Variables of Fire Alarm System

From line 1-8, initial variables of the sensors and libraries were included.

```
10 void setup()
11 {
12 Serial.begin(9600); // sets the serial port to 9600
13 Serial.println("Gas sensor warming up!");
14 delay(100000);
15 pinMode(gas_sensor_pin,INPUT);
16 pinMode(buzzer_pin,OUTPUT);
17 }
```

Figure 15 Initial Setup

From line 10-16 the initial setup of pins was done i.e., which pins are outputs and which pints are inputs and a grace period of 1 minute is used for the sensor to warm up because initially a very high value of concentration of smoke is detected until the sensor reaches a steady state value.

```
19 void loop() {
20 gas_sensor_input=analogRead(gas_sensor_pin);
21 flame_sensor_state=digitalRead(flame_sensor_pin);
22 Serial.print("Gas Level: ");
23 Serial.println(gas_sensor_input);
24 if((gas_sensor_input > threshold)&&(flame_sensor_state==1))
25 {
26 lcd.clear();
27 Serial.println("FIRE DETECTED");
28 lcd.print("FIRE DETECTED!");
29 delay(500);
30 lcd.clear();
31 lcd.print("SMOKE DECTECTED!");
32 delay(500);
33 digitalWrite(buzzer_pin,HIGH);
34 delay(1000);
35 }
36 else if ((gas_sensor_input > threshold)&&(flame_sensor_state==0))
37 {
38 lcd.clear();
39 lcd.print("No Smoke");
40 digitalWrite(buzzer_pin,LOW);
41   }
42 else if ((gas_sensor_input < threshold)&&(flame_sensor_state==1))
43 {
44 lcd.clear();
45 lcd.print("No Smoke");
46 digitalWrite(buzzer_pin,LOW);
47   }
48 else if ((gas_sensor_input < threshold)&&(flame_sensor_state==0))
49 {
50 lcd.clear();
51 lcd.print("No Smoke");
52 digitalWrite(buzzer_pin,LOW);
53   }
54 }
```

Figure 16 Main logic of Fire Alarm System

From line 19-54, main logic of the system is implemented. Firstly, output voltage is read from the gas sensor and aft wards the state of the IR flame sensor is checked.

The threshold voltage of smoke sensor is defined in the first few lines and is implemented in the if statement.

Different scenarios are listed using if and else statements but the alarm only goes off if both smoke and fire is detected. The threshold value is to be calibrated according to working conditions.

The demonstration video is attached with this report and online simulation can be accessed here.

3.1.5 Bill of Materials

The prices are based on actual value at the time of purchase of the sensors and hence all the prices are accurate.

Item	Price (RS)	Quantity
Arduino Uno	600	1
LCD Screen	350	1
MQ-2 Gas Sensor	200	1
IR-Flame Sensor	250	1
Buzzer	30	2
Potentiometer	20	1
Resistor (220 Ω)	10	1
Total	1460	-

Chapter 4. Conclusion

The prototype of the home security and fire alarm system was discussed in this report. We discussed that the use of two sensors side by side can increase the accuracy of the system. The security system was achieved using a PIR sensor and a ultrasonic sensor and was demonstrated using tinkercad and physically and fire alarm was demonstrated in the same way. The overall cost of the system is estimated to be 3000 RS considering only the electronics but can increase more depending upon the requirements of the system that is if we replace buzzer with ventilation system, then additional relays will be required to actuate the system. The sensors and Arduino will also need a casing to mount all the sensors in the required area.

A container can be easily designed using solidworks and can be manufactured physically using acrylic sheet. An example model has been shown in Figure 17

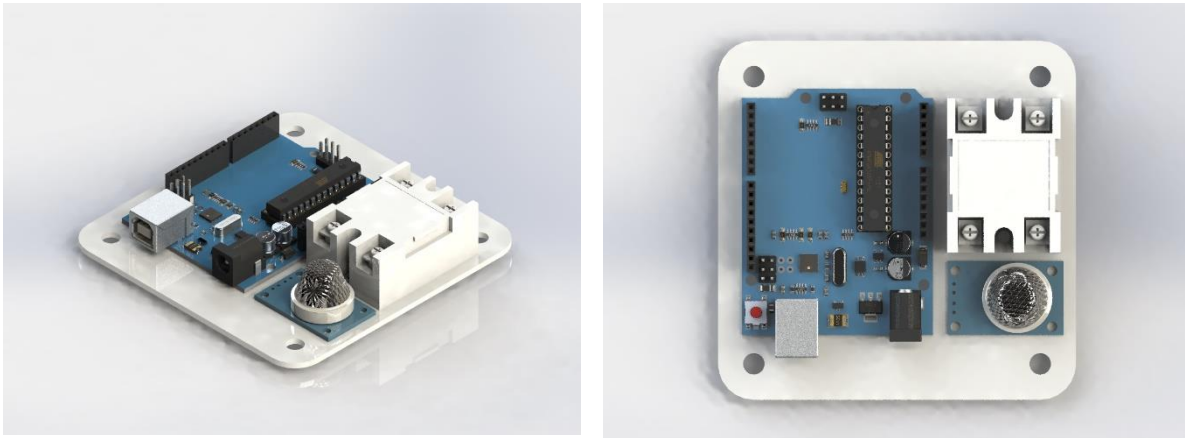

Figure 17 Sensors' Container

Appendix I: Arduino Codes:

Home Security System:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int ultrasonic_trigger_pin=10;
int ultrasonic_echo_pin=11;
int pin_pir_sensor_input=A1;
int pir_state;
int pin_buzzer_output=12;
int distance;
float time;
float velocity_sound=0.034;
void setup()
{
  pinMode(pin_buzzer_output,OUTPUT);
    pinMode(ultrasonic_echo_pin,INPUT);
  pinMode(ultrasonic_trigger_pin,OUTPUT);
  pinMode(pin_pir_sensor_input,INPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);
 lcd.begin(16, 2);
}
void loop()
{
digitalWrite(ultrasonic_trigger_pin,HIGH);
delayMicroseconds(10);
digitalWrite(ultrasonic_trigger_pin,LOW);
time = pulseIn (ultrasonic_echo_pin, HIGH);
distance = (time * velocity_sound) / 2;
pir_state=digitalRead(pin_pir_sensor_input);
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm ");
lcd.setCursor(0,0);
lcd.print("Distance : ");
lcd.print(distance);
lcd.print("cm ");
delay(200);
if (pir_state==1) {
Serial.println("Motion is Detected!");
lcd.setCursor(0,1);
lcd.print("Motion Detected!");
}
else {
lcd.clear();
}
if((distance < 30 )&&(pir_state == 1))
digitalWrite(pin_buzzer_output,HIGH);
else
digitalWrite(pin_buzzer_output,LOW);
}
```

Fire Alarm System:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
float gas_sensor_input;
int gas_sensor_pin=0;
int flame_sensor_pin=8;
int flame_sensor_state;
int buzzer_pin=12;
int threshold=50;
void setup()
{
Serial.begin(9600); // sets the serial port to 9600
Serial.println("Gas sensor warming up!");
delay(100000);
pinMode(gas_sensor_pin,INPUT);
pinMode(buzzer_pin,OUTPUT);
}
void loop() {
gas_sensor_input=analogRead(gas_sensor_pin);
flame_sensor_state=digitalRead(flame_sensor_pin);
Serial.print("Gas Level: ");
Serial.println(gas_sensor_input);
if((gas_sensor_input > threshold)&&(flame_sensor_state==1))
{
lcd.clear();
Serial.println("FIRE DETECTED");
lcd.print("FIRE DETECTED!");
delay(500);
lcd.clear();
lcd.print("SMOKE DECTECTED!");
delay(500);
digitalWrite(buzzer_pin,HIGH);
delay(1000);
}
else if ((gas_sensor_input > threshold)&&(flame_sensor_state==0))
{
lcd.clear();
lcd.print("No Smoke");
digitalWrite(buzzer_pin,LOW);
  }
else if ((gas_sensor_input < threshold)&&(flame_sensor_state==1))
{
lcd.clear();
lcd.print("No Smoke");
digitalWrite(buzzer_pin,LOW);
  }
else if ((gas_sensor_input < threshold)&&(flame_sensor_state==0))
{
lcd.clear();
lcd.print("No Smoke");
digitalWrite(buzzer_pin,LOW);
  }
```