Smart Home Security System Using Arduino

Maisha Haque
School of Data and Sciences
Brac University
Dhaka, Bangladesh
maisha.haque@g.bracu.ac.bd

Department of Computer Science and Engineering
Brac University
August, 2023

I. Introduction

Unwanted home intrusions have been a basic and ancient problem in human society. Even life-threatening incidents take place due to home intrusions, thefts and fires. Accidental fires and arson in homes are deadly especially when owners are not aware of it. Smoke detectors will sense smoke generated from burning objects by the fire and will set an alarm to alert the occupants. Motion detection sensors will detect unwanted motions when a door or window opens or any intruder moves about the home.

II. Application Area

This project is used for home security purposes. Any form of intrusion is detected and alerted. Moreover, the system will detect smoke occurrences in homes. It will particularly detect unwanted door and window motions, unwanted movements in homes while the owner is out.

III. Technology and tools

A. Components

- Arduino Uno
- Breadboard Bundle
- PIR Motion Sensor
- MQ-2 Smoke Sensor
- HC-SR04 Ultrasonic Sensor
- Jumper Wires
- RGB LED
- Piezo Buzzer

B. Programming language:

The programming language used here is C++.

IV. Working mechanism of Sensors

A. PIR motion sensor

A PIR (Passive Infrared) motion sensor senses changes in the infrared radiation given off by objects to detect movement. It has two sensor components that compare the temperatures of the foreground and background. The sensor produces a small electric charge when an object moves in its field of view as a result of temperature variations in its pyroelectric material. This charge is processed, and if there is noticeable movement, an output signal that can be used to turn on lights, alarms, or other devices is triggered. PIR sensors can detect motion without emitting radiation because they are passive and energy-efficient.

B. MQ-2 Flammable Gas Smoke Sensor

The MQ-2 flammable gas smoke sensor operates using chemiresistive technology. Its sensing element interacts with gasses like smoke, alcohol, and methane, causing changes in electrical resistance proportional to gas concentration. A built-in heating element raises the temperature of the sensor to increase gas-material interactions and provide increased sensitivity. The sensing and heating elements are connected to the sensor by two terminals. It converts resistance variations into quantifiable gas concentrations when connected to a microcontroller. For accurate readings, calibration is necessary to establish a link between resistance variations and known gas concentrations. In smoke detection, elevated resistance readings result from increased gas concentrations brought on by smoke. This output may set off alarms or initiate other actions. Although its accuracy can be impacted by environmental factors like temperature and humidity. When calibrated correctly and used in the right circumstances, the MQ-2 smoke sensor offers a cost-effective solution for gas detection applications despite potential variations.

C. HC-SR04 Ultrasonic Sensor

The HC-SR04 Ultrasonic Sensor operates based on the principle of echolocation.. It sends out brief pulses of ultrasonic sound waves and time how long it takes for the waves to bounce off an object and come back. A transmitter that emits ultrasonic pulses and a receiver that picks up reflected waves make up the sensor. A pulse is released and moves through the atmosphere until it comes into contact with an object. The pulse is reflected back towards the sensor when it contacts the surface of the object. The returning pulse is then picked up by the sensor's receiver. The sensor uses the speed of sound in air to determine the distance to the object by timing the interval between the transmission and reception of pulses. This time-of-flight measurement is used by the HC-SR04 sensor to provide precise distance readings for a variety of applications, including obstacle avoidance.

V. Connection with ICs

The following components has been connected to build the security system. The connection description is given below.

1. HC-SR04 Ultrasonic Sensor

- VCC pin to Arduino 5V
- GND pin to Arduino GND
- Trig pin to Arduino digital pin
- Echo pin to Arduino digital pin

2. PIR Motion Sensor

- VCC pin to Arduino 5V
- GND pin to Arduino GND
- Output pin to Arduino digital pin

3. MQ-2 Smoke Detector

- VCC pin to Arduino 5V
- GND pin to Arduino GND
- Analog output pin to Arduino analog input

4. Piezo Buzzer

- Positive to Arduino digital pin
- Negative to Arduino GND

5. RGB LED

- RGB pins to Arduino digital pin
- Cathode to Arduino GND

VI. Estimated Cost Analysis

An approximated cost has been calculated for the security system in I. The price might change depending on the market price.

Total estimated cost would be around 2000 bdt.

TABLE I
ESTIMATED PRICE OF THE COMPONENTS

Component	Price
Arduino UNO	1000 bdt
PIR motion sensor	85 bdt
MQ-2 smoke sensor	150 bdt
HC-SR04 Ultrasonic Sensor	93 bdt
Buzzer	45 bdt
Breadboard bundle	580 bdt
Breadboard bundle	580 bdt
LED	50 bdt
Jumper cables	100 bdt (per bundle)

VII. CONNECTION OF THE COMPONENTS

A. Circuit Diagram

The circuit diagram of the connection of the components can be seen in Fig 1

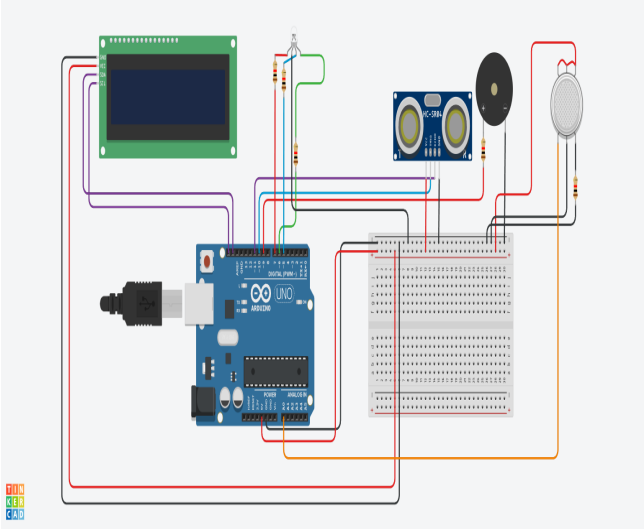

Fig. 1. Circuit diagram of the proposed system

B. Code

The code setup for Arduino is given in Fig 2, Fig 3, Fig 4 & Fig 5.

VIII. CONCLUSION

In conclusion, we can say that ensuring home security is necessary in order to improve the standard of living. Our system will prove to be beneficial in order to tackle any possible threats to one's home and prove to be an updated version to the existing system. The incorporation of PIR motion sensor, MQ-2 smoke sensor and HC-SR04 Ultrasonic

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int PIRSensor = 2; //Define PIR Sensor pin

int Buzzer = 9; //PWM (~) pin BUzzer

int trigPin = 10; //trigger pin
int echoPin = 11; //echopin
int DISTANCE_THRESHOLD = 50; // centimeters


int smoke_detector = A0;
int safety_lim = 100; //Sets  smoke density safe limit

LiquidCrystal_I2C lcd(0x27,16,2);


int redPin= 7;
int greenPin = 6;
int bluePin = 5;

```

Fig. 2. Part 1

```
void setup(){
  Serial.begin(9600); //set baud rate

  //Sensors
  pinMode(PIRSensor, INPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(smoke_detector, INPUT);


  //RGB
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

  //Buzzer
  pinMode(Buzzer, OUTPUT);



  //LCD Setup
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  Serial.println("Alarm is Activated");
  lcd.setCursor(0,0);

  //RGB Setup
  setColor(0,  255, 0); // Green Color


}
```

Fig. 3. Part 2

Sensor will allow the security system to detect any mishaps that might occur due to any unwanted motion, if any intruder tries to barge in or any fire incident or if any doors or windows are tried to open.

```
void loop() {

  //Readings

  //Ultrasonic
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delay(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  int duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  int distance = duration * 0.034 / 2;

  //PIR & Smoke
  int SensorValue = digitalRead(PIRSensor);
  int sensor_read = analogRead(smoke_detector);

  //Conditions for Alarm

  //If triggers PIR Sensor

  if(SensorValue == HIGH){
    setColor(255, 0, 0); // Red Color
    tone(Buzzer, 10000);
    lcd.setCursor(0,0);
    Serial.println("Motion Detected!!!");
    delay(3000);
    lcd.clear();
  }
```

Fig. 4. Part 3

```
  //If triggers Ultrasonic
  else if(distance<DISTANCE_THRESHOLD){
    setColor(255, 0, 0); // Red Color
    tone(Buzzer, 10000);
    //Serial.println("Door Opened!!!");
    Serial.println(sensor_read);
    delay(3000);
    lcd.clear();
  }

  else if (sensor_read > safety_lim){

    setColor(255, 0, 0); // Red Color
    tone(Buzzer, 10000);
    Serial.println("FIRE!!!");
    //Serial.println(sensor_read);
    delay(3000);
  }

  //If nothing is triggered
  else {
    noTone(Buzzer);
    setColor(0,  255, 0); // Green Color
    Serial.println("Alarm Activated");
    delay(1000);
  }

}

void setColor(int redValue, int greenValue,  int blueValue) {
  analogWrite(redPin, redValue);
  analogWrite(greenPin,  greenValue);
  analogWrite(bluePin, blueValue);
}
```

Fig. 5. Part 4