

# **SMART HOME AUTOMATION USING IR AND IOT**

## **SUBMITTED BY**

SURA JAYADEV – 20BCD7264  
VUPPUTURI LAKSHMI MEGHANA – 20BCI7174  
THATAVARTHI BALA SAI VAMSI – 20BCE7407  
RENDLA SAI KRISHNA REDDY – 20BEV7063  
TOTKAR JATIL ADITHYA – 20BCD7104  
RUDRU SAI MANOJ – 20BES7058

## **GUIDED BY**

Prof. Rohit Lorenzo

(Professor, SENSE)



**SCHOOL OF COMPUTER SCIENCE & ENGINEERING, VIT-AP  
UNIVERSITY  
DECEMBER 2022**

## **ABSTRACT:**

Smart home Automation technology, lifetime is getting easier and simpler on all side in life. In the world, automatically systems are being chosen over the manual system. With the fast growth in the number of users of the internet through the past decade, the Internet has become a part and section of human life. In this study, a remotely controlled smart home automation system, for monitoring and controlling the temperature degree, humidity percentage, gas leakage, fires, and rain in the house, was developed.

The developed system consists of two parts: hardware and software. The hardware consists of many units such as smartphones, personal computers, Arduino UNO, Node Mcu, IR sensor and relay board, gas, motion, flame, light and rain sensors, a buzzer and fire alarm. Arduino Mega used as the Micro-controller is the primary hardware unit connected with an Node mcu responsible for sending and receiving directives and status via the internet. The software is the second part of this system; it is Arduino IDE's code used to connect to the telegram. The telegram connected to hardware devices and sensors through Node mcu. Moreover, it works to control and monitor this system from the Node mcu and smartphone.

Finally, this system uses an internet network for communication and alerts due to its low cost and large area coverage and can control every device via the web browser and Android application anywhere and anytime. Moreover, the gas, flame, light, temperature and rain sensors work correctly with excellent results

## **INDEX:**

1. Introduction
2. Background
3. Problem Definition
4. Objectives
5. Methodology/Procedure
6. Results and Discussion
7. Conclusion and Future Scope
8. References
9. Codes in Appendix

## **INTRODUCTION:**

Embedded physical devices, such as household appliances are becoming smarter and smarter, They are equipped with embedded microprocessors and wireless transceivers, offering limited communication capabilities and providing good behavior, Everyday objects fitted with small, cheap mobile processors, sensors, and actuators . Sensors and wireless sensor networks are being deployed in smart home solutions, measuring with precision the environmental conditions in the home environment, their advanced sensing functionalities and their increasing accuracy enable the development of smart home applications that offer advanced automation. Residences are transformed into a smart house, incorporating embedded sensors and actuators, and pervasive technology .

Efficaciously addressing the needs within the realms of the 'Smart Home' environment, requires a multidisciplinary collaboration. This development usually unfolds specialists from Architecture, Electrical and Electronic Engineering, Computer Science and some purpose related to the health sector may also require the participation of professionals from Social Sciences, Medicine, and Occupational Therapy. Although all people can be interested in such undesirable situations, elderly people and people with health problems, require more exclusive monitoring when they are not accompanied by a healthcare professional.

It is possible for someone exhibiting early stages of cognitive impairments too; a "smart home" typically is a domestic environment that has been partially automated. Home automation includes the centralized control for lighting, air conditioning, ventilation and heating also devices controlling and others, Smart Home automation, aims to promote the comfort energy consumption efficiency and safety in domestic scenarios.

Houses are equipped with autonomous control panels to control all of the systems and devices present in the home, Moreover, those control panels are overwhelmingly not regarding each other, The main aim of an intelligent house is to focus the control of all the devices into a private control unit which can be programmed to do special tasks suitable for the possessor and the house in question, The aim of a smart home is not only suitability but also to reduce the consuming of resources such as power, gas, etc.

## BACKGROUND :

With the 1975 release of X10, a communication protocol for home automation, the smart home, once a pipe dream a la *The Jetsons*, came to life. X10 sends 120 kHz radio frequency bursts of digital information onto a home's existing electric wiring to programmable outlets or switches. These signals convey commands to corresponding devices, controlling how and when the devices operate. A transmitter could, for example, send a signal along the house's electric wiring, telling a device to turn on at a specific time.

However, as electrical wiring isn't designed to be free from radio-band "noise," X10 was not always fully reliable. Signals would be lost and, in some cases, signals wouldn't cross circuits that were wired on different polarities, created when 220-volt service is split into a pair of 100-volt feeds, as is common in the U.S. Additionally, X10 was initially a one-way technology, so while smart devices can take commands, they cannot send data back to a central network. Later, however, two-way X10 devices became available, albeit at a higher cost.

When home automation company Insteon came on the scene in 2005, it introduced technology that combined electric wiring with wireless signals. Other protocols, including ZIGBEE and Z-WAVE, have since emerged to counter the problems prone to X10, though X10 remains a widely installed communications protocol to this day.

Nest labs was founded in 2010 and released its first smart product, the Nest Learning Thermostat, in 2011. The company also created smart smoke/carbon monoxide detectors and security cameras. After being acquired by Google in 2015, it became a subsidiary of Alphabet Inc. in the same year.

In 2012, Smart Things Inc. launched a Kick starter campaign, raising \$1.2 million to fund its smart home system. Following additional funding, the company came on the market in August 2013 and was acquired by Samsung in 2014.

More recently, companies including Amazon, Apple and Google have released their own smart home products and domestics platforms, including Amazon Echo, Apple Home Kit and Google Home

## **Problem Definition:**

- 1.General house hold appliances occupies a huge variety of devices which we use daily as a part of our life .A physically handicapped person cannot use these appliance to their full extent, in order to aid them we need a system that can help them in a way such that it requires minimum effort .
- 2.We need a smart system that monitors the entire home such as air quality, fire accidents, unprecedented rains,gas leakage and etc.
- 3.We need a system through which we can control devices in the house from any location in the world with simple operations and an internet facility.
- 4.All these things are achieved by the help of this smart home through IOT

## **Objective of the Smart home:**

- 1.It is to improve the quality of life and convenience in the home, greater security and more efficient use of energy thanks to connected, remote-controllable devices that helps in the process of the achieving the easy and efficient utilization of the household utensils and devices from anywhere around the globe with a simple internet connection between the user and the controlling unit which is placed at the house
- 2.The other objectives are to aid physically challenged people in using the house hold devices without much effort
- 3.Continuous monitoring of the house for emergency cases such as fire accidents and gas leakage
- 4.In order to have a continuous surveillance over the house to prevent any situations of robbery, trespassing and etc.

These are the objectives of this project that are are kept in view.

## **Procedure :**

### **1.1 COMPONENTS REQUIRED:**

1. Arduino uno R3
2. ESP 8266 node mcu
3. 4 channel relay module
4. IR remote and sensor
5. ultrasonic sensor
6. Fire sensor
7. MQ-2 gas and smoke sensor
8. Bread board
9. Jumpers
- 10.Fan
- 11.Arduino IDE

### **1.2 Making process:**

### **Creating a Telegram Bot :**

Open Telegram and follow the steps to create a Telegram Bot. First, search for “botfather” and click the BotFather. Or open this link [t.me/botfather](https://t.me/botfather) in your smartphone.

Type /newbot and follow the instructions to create your bot. Give it a name and username.

If your bot is successfully created, you’ll receive a message with a link to access the bot and the bot token. Save the bot token because you’ll need it so that the ESP32/ESP8266 can interact with the bot

Anyone that knows your bot username can interact with it. To make sure that we ignore messages that are not from our Telegram account (or any authorized users), you can get your Telegram User ID. Then, when your telegram bot receives a message, the ESP can check whether the sender ID corresponds to your User ID and handle the message or ignore it.

In your Telegram account, search for “IDBot” or open this link [t.me/myidbot](https://t.me/myidbot) in your smartphone.

Start a conversation with that bot and type /getid. You will get a reply back with your user ID. Save that user ID.

### **Preparing Arduino IDE:**

We’ll program the ESP8266 boards using Arduino IDE, so make sure you have them installed in your Arduino IDE.

### **Universal Telegram Bot Library**

To interact with the Telegram bot, we’ll use the universal telegram bot library created by Brian Lough that provides an easy interface for the Telegram Bot API.

Follow the next steps to install the latest release of the library.

1. Go to Sketch > Include Library > Add.ZIP Library...
2. Add the library you’ve just downloaded.  
And that’s it. The library is installed.

You also have to install the AduinoJson library. Follow the next steps to install the library.

1. Go to Skech > Include Library > Manage Libraries.
2. Search for “ArduinoJson”.
3. Install the library.

### **Hardware Assembly:**

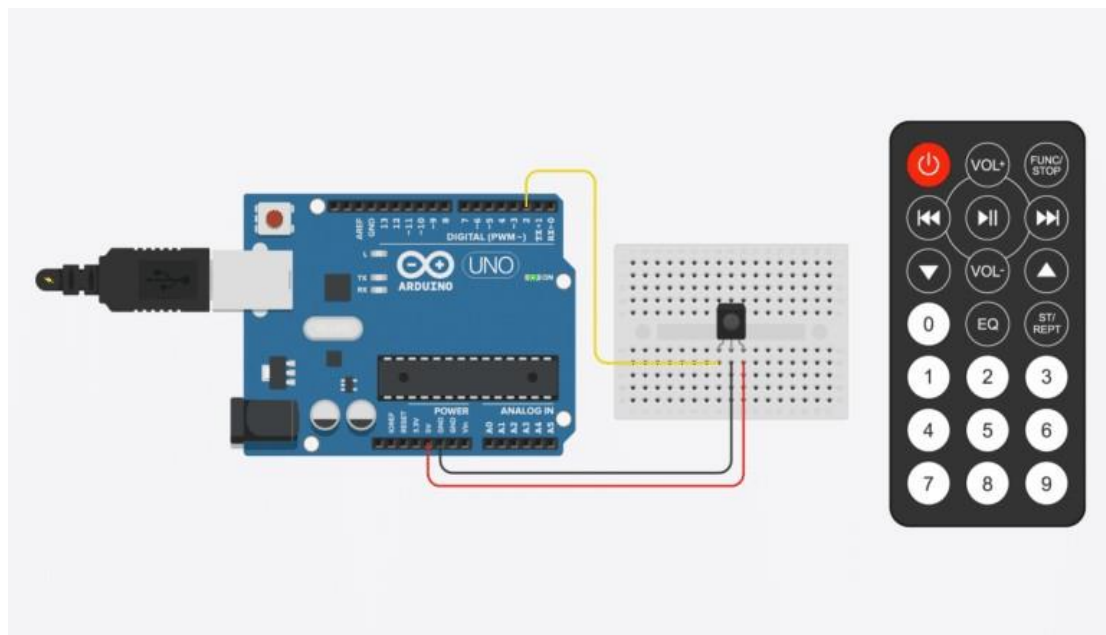
1. Connect the Arduino uno with the Relay Board with a breadboard and make connections accordin to the circuit diagram.
2. Connect the Data pin of Arduino uno with Pin 1,2,3,4 of the Relay.
3. Connect the ground pin of the Relay to the ground of the Arduino uno and vcc.
4. Connect the data pin of the node mcu to the data of the Arduino uno in order to detect the sensor values from the arduino uno.



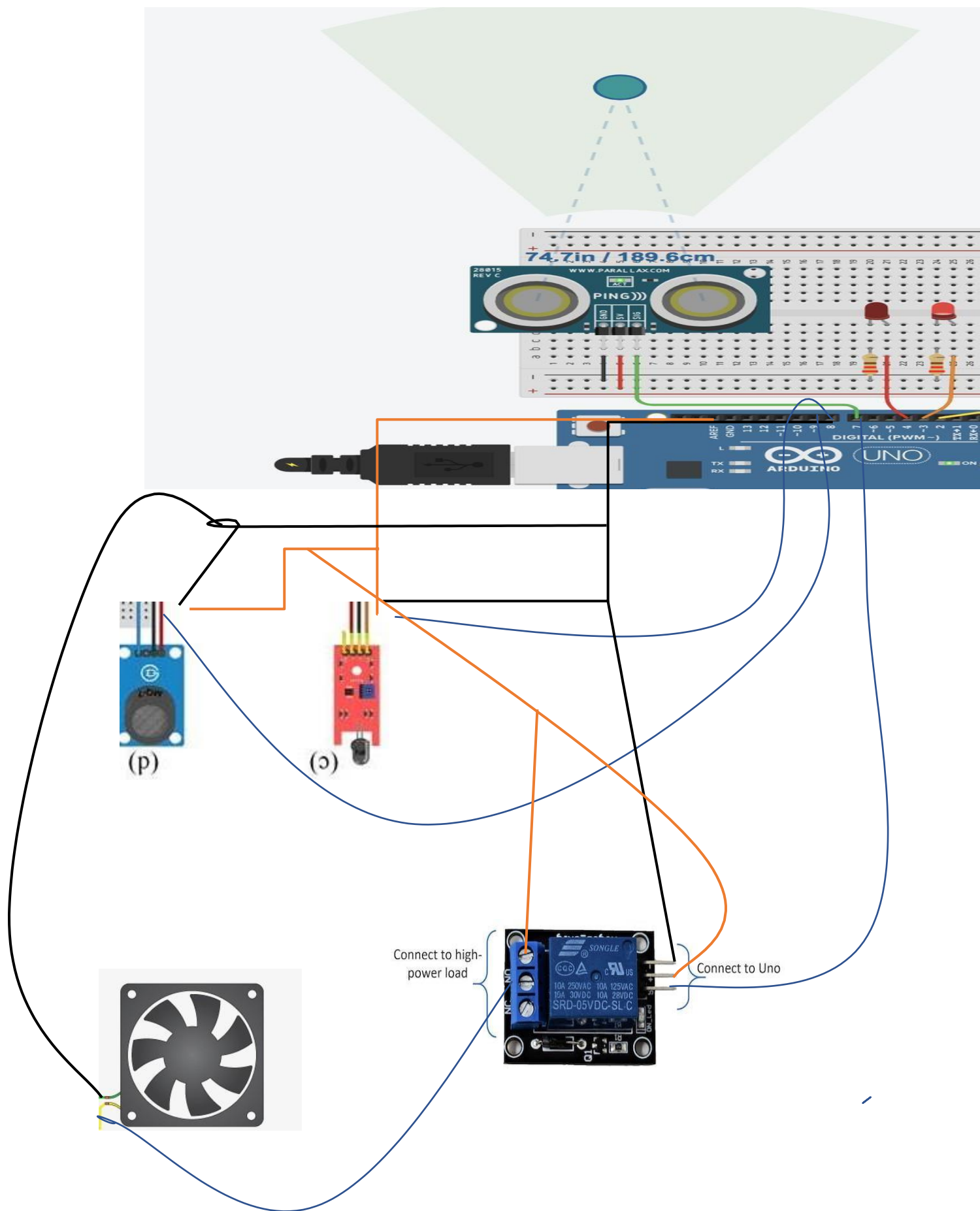
4. Power the Arduino uno an node mcu with a normal low voltage phone charger and power up the Relay Board using Arduino UNO board.
5. Turn on the mobile hotspot whose id was used while configuring the node mcu telegram code.
6. Four electronic appliances can be connected since we're using 4 Relay pins but in our case, we only used one, an FAN .
7. Now this assembly is complete and we can use the Telegram App to turn our electrical appliances on and off with our phone.

## **CIRCUIT DIAGRAM :**

### **1.IR REMOTE CIRCUIT :**



### **2. Peripheral circuit :**



## **RESULTS AND DISCUSSION:**

According to the proposed plan the final outcome of this paper leads to the development of a home automation. Through this project, an automation system has been created so that we can easily control home appliances like as light, fan, tube light, AC, bulb, etc.

One of the objectives of this project is also to get us a smart automation and low cost. This system controlled the light and fans using IR inside the home, we can add more electrical appliances. The hardware consists of many units such as smartphones, rain sensor, fire alarm, ultrasonic sensor. Arduino uno is used as the microcontroller is the primary hardware unit connected with Wi-Fi module for sending and receiving directives and status via telegram. Moreover, it works to control and monitor this system from the telegram and smartphone. Finally, this system uses an internet network for communication and alerts due to its low cost and large area coverage and can control every device via the telegram anywhere and anytime. Moreover, the gas, flame, light, temperature, humidity and rain sensors work correctly with excellent results.

## **CONCLUSION:**

The aim of this project was to design and perform a smart home automation system by remotely controlling and monitoring electronic devices in the house and monitoring of the temperature degree & Humidity percentage, gas leakage, fires, and rain; also Automation light by using motion and light sensors inside and outside the home by using internet of things. This work used an internet network due to its low cost and coverage of a large zone.

It is applied to send information to the system to control devices and receive statuses of the appliances. The target has been carried out successfully. These devices are controlled by the web browser via the internet.

As a result, the system can monitor and controlled from the web browser. Moreover, the gas, flame, light, temperature & Humidity and rain sensors work correctly with excellent results. This system also uses power saving.

An significant advantage of this system is that control every device via the web browser and Android application anywhere and anytime. Finally, the developed system offers a flexible, economical and remotely controllable smart home automation system.

## **REFERENCES:**

- [1] Kamilaris, A., "Enabling smart homes using web technologies," Ph.D. dissertation, Dept. of computer science, University of Cyprus, Cyprus, 2012.
- [2] Bergman E. Information appliances and beyond. Interaction design for consumer products. Morgan Kaufman, pp.2-10, 2000.

- [3] Augusto, J.C., and Nugent, C.D. Smart homes can be smarter. Springer Berlin Heidelberg, pp. 5-12, 2006.
- [4] Staroch P., "A Weather Ontology for Predictive Control in Smart Homes," MSc dissertation, Dept. of Software Engineering and Internet Computing, Vienna, Austria, 2013.
- [5] Algoiare T. O., "Design and Implementation of Intelligent Home Using GSM Network" MSc dissertation, Dept. of Computer Engineering, Ankara, Turkey, 2014.
- [6] Anonymous, Wikipedia, Home automation, [http://en.wikipedia.org/wiki/Home\\_automation](http://en.wikipedia.org/wiki/Home_automation) ,2016.
- [7] Anonymous, Arduino, website. <https://www.arduino.cc/en/Main/Products>, 2016.

## **CODES IN APPENDIX:**

### **Esp code:**

```
//#include <TelegramCertificate.h>
#include <UniversalTelegramBot.h>

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
//#include <UniversalTelegramBot.h>

const char* ssid = "Vamsi";
const char* password = "Vamsi@2003";
#define BOTtoken "5623192294:AAHcLr6VkB2zjczSJQsKV2sfmUZMWuCaWo"
#define CHAT_ID "1489465834"

#define Sensor D10
#define LEDR D3
#define LEDG D9
#define bud D1
#define doore D0
#define in1 D8
#define in2 D7
#define temp D5
#define rain D6
#define gas D4

X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
```

```

int flag=1;//close
bool val;
const int ledPin = 2;
bool ledState = LOW;

// Handle what happens when you receive new messages
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));

  for (int i=0; (i<numNewMessages); i++) {
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;

    if (text == "/start") {

      String welcome = "Welcome, " + from_name + ".\n";
      welcome += "Use the following commands to control your outputs.\n\n";
      welcome += "Enter /open to open the door \n";
      welcome += "Enter /close to close the door \n";
      welcome += "Enter /state to request current door state \n";
      bot.sendMessage(chat_id, welcome, "");
      delay(10000);
      bot.sendMessage(CHAT_ID, "Somebody is at door!!", "");
    }
    val=(digitalRead(Sensor));
    if(val==LOW)
    {
      bot.sendMessage(CHAT_ID, "Somebody is at door!!", "");

      //return;
    }

    if ((text == "/open")&&(flag==1))
    {
      bot.sendMessage(chat_id, "door is set to opened", "");
      ledState = HIGH;
      digitalWrite(ledPin, ledState);
      digitalWrite(doore, HIGH);
      digitalWrite(in1, HIGH);
    }
  }
}

```

```

digitalWrite(in2, LOW);
digitalWrite(LEDG, HIGH);
digitalWrite(LEDOR, LOW);
delay(700);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(LEDOR, HIGH);
digitalWrite(LEDG, LOW);
delay(700);
flag=0;//open
Serial.println("open condition entered");
return;
Serial.println("open condition entered, after return failed");
}

if ((text == "/close")&&(flag==0))
{
  bot.sendMessage(chat_id, "door is set to closed", "");
  ledState = LOW;
  digitalWrite(ledPin, ledState);
  //digitalWrite(doore, LOW);
  digitalWrite(bud,HIGH);
  digitalWrite(LEDOR, HIGH);
  digitalWrite(LEDG, LOW);
  // digitalWrite(doore, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(700);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  delay(700);
  flag=1;
  Serial.println("close condition entered");
  return;
  Serial.println("close condition entered, after return failed");
}

if (text == "/state")
{
  int sensorValue1 = digitalRead(temp);
  int sensorValue2 = digitalRead(rain);
  int sensorValue3 = digitalRead(gas);
  if (sensorValue1== HIGH){
    bot.sendMessage(CHAT_ID, "No fire alert!", "");
    delay(3000);
    //digitalWrite(buzzPin, HIGH);
    //digitalWrite(LED1, HIGH);
  }
  else{
    bot.sendMessage(CHAT_ID, "Fire Alert..!!!!", "");
  }
}

```

```

    delay(3000);
}

if (sensorValue2 == HIGH){
    bot.sendMessage(CHAT_ID, "NO LEAK!", "");
    delay(3000);
    //digitalWrite(buzzPin, HIGH);
    //digitalWrite(LED1, HIGH);
}
else{
    bot.sendMessage(CHAT_ID, "Gas Leakage ....!!!!", "");
    delay(3000);
}

if (sensorValue3 == HIGH){
    bot.sendMessage(CHAT_ID, "It's not raining!!!", "");
    delay(3000);
    //digitalWrite(buzzPin, HIGH);
    //digitalWrite(LED1, HIGH);
}
else{
    bot.sendMessage(CHAT_ID, "It's Raining... !!!!", "");
    delay(3000);
}

if (digitalRead(ledPin)){
    bot.sendMessage(chat_id, "door is open", "");
}
else{
    bot.sendMessage(chat_id, "door is closed", "");
}

}

}

}

void setup() {
    Serial.begin(9600);
    configTime(0, 0, "pool.ntp.org");    // get UTC time via NTP
    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

    pinMode(Sensor, INPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    // pinMode(Buzzer, OUTPUT);
    pinMode(ledPin, OUTPUT);
    pinMode(doore, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);

```

```

    pinMode(temp, INPUT);
pinMode(rain, INPUT);
pinMode(gas, INPUT);
digitalWrite(ledPin, ledState);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

int a = 0;
while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
    a++;
}

Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
delay(500);

bot.sendMessage(CHAT_ID, "/start", "");
}
void loop() {
    /*bool value = digitalRead(Sensor);

    if (value == 1) {
        bot.sendMessage(CHAT_ID, "Somebody is at door!!", "");
//    value = LOW*/

    val=(digitalRead(Sensor));
    if (millis() > lastTimeBotRan + botRequestDelay) {
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while(numNewMessages)
        {
            Serial.println("got response");

            handleNewMessages(numNewMessages);
            Serial.println("void loop entered again");
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        lastTimeBotRan = millis();
    }
}
}

```

## IR CODE:

```

#include <IRremote.h>
#define IR_RECEIVE_PIN 11

```



```

#define IR_BUTTON_2 70
#define IR_BUTTON_3 71
#define IR_BUTTON_4 68
#define IR_BUTTON_5 64
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    IrReceiver.begin(IR_RECEIVE_PIN);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
}

void loop() {
    // put your main code here, to run repeatedly:
    //Serial.begin(9600);
    if (IrReceiver.decode())
    {
        IrReceiver.resume();
        int command = IrReceiver.decodedIRData.command;

        switch (command)
        {
            case IR_BUTTON_2:
            {
                digitalWrite(8,HIGH);

                delay(500);
                break;
            }
            case IR_BUTTON_3:
            {
                digitalWrite(8,LOW);

                break;
            }
            case IR_BUTTON_4:
            {
                digitalWrite(7,HIGH);

                delay(500);
                break;
            }
            case IR_BUTTON_5:
            {
                digitalWrite(7,LOW);

                break;
            }
        }
    }
}

```

```

    }
    default: {
        Serial.println("Button not recognized");
    }
}
}
}
}

```

### Peripheral code:

```

#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH1106.h>

#define OLED_RESET -1
Adafruit_SH1106 display(OLED_RESET);

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3
#define t1 4
#define t2 5
#define t3 6
#define i2c_Address 0x3c //initialize with the I2C addr 0x3C

int smokeA0 = A3;
int temp = A2;
int rain = A1;
long duration;
int distance;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    display.begin(SH1106_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    pinMode(t1,OUTPUT);
    pinMode(t2,OUTPUT);
    pinMode(t3,OUTPUT);
    pinMode(rain,INPUT);
    pinMode(temp,INPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    pinMode(trigPin,OUTPUT);
    pinMode(echoPin,INPUT);
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
    digitalWrite(t1,LOW);
    digitalWrite(t2,LOW);
    digitalWrite(t3,LOW);
}

```

```

display.clearDisplay();
display.setTextSize(2);
display.setTextColor(WHITE);
// display.setCursor(0, 10);
display.setCursor(0, 10);
display.print("System initiated");
display.display();
delay(3000);
}

void loop()
{

int sensorThres = 400;
int analogSensor = analogRead(smokeA0);
if (analogSensor > sensorThres)
{
digitalWrite(t1,HIGH);
// Serial.println(i);
display.clearDisplay();
//display.setTextSize(2);
//display.setTextColor(WHITE);
display.setCursor(0, 10);
display.print("Smoke detected..");
display.setCursor(0, 40);
display.print("GAS Leaking..");
display.display();
delay(3000);
digitalWrite(t1,LOW);

}
int sensorValue1 = digitalRead(temp);
if (sensorValue1 == LOW)
{
digitalWrite(t2,HIGH);
//Serial.println(i);
display.clearDisplay();
//display.setTextSize(2);
//display.setTextColor(WHITE);
display.setCursor(0, 10);

display.print("Temp rise..");
display.display();
delay(3000);
digitalWrite(t2,LOW);

}
}

```

```

int sensorValue2 = digitalRead(A1);
if (sensorValue2 == LOW)
{
    digitalWrite(t3,HIGH);
    display.clearDisplay();
    //display.setTextSize(2);
    //display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.print("Raining...");
    display.display();
    delay(3000);
    digitalWrite(t3,LOW);
}

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin HIGH (ACTIVE) for 10 microseconds
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/2) / 29.1;
int new_delay= (distance *3) +30;
if (distance < 50)
{
    digitalWrite(10,HIGH);
    //delay(new_delay);
    delay(1000);
    digitalWrite(10,LOW);

    display.clearDisplay();
    //display.setTextSize(2);
    //display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.print("Area   breached..");
    display.display();
    delay(3000);
}
else
{
    digitalWrite(10,LOW);
    // Serial.println(i);
    display.clearDisplay();
    //display.setTextSize(1);
    // display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.print("Perimeter secured..");
    display.display();
}

```

```
    delay(3000);  
  }  
}
```