Bangladesh University of Engineering and Technology
Department of Electrical and Electronic Engineering

EEE 416 (JULY 2023)

Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: B1 Group: 02

Home Automation and Safety Using Arduino

Course Instructors:

Sadman Sakib Ahbab, Assistant Professor
Rasin Mohammed Ihtemam, Part-Time Lecturer

Signature of Instructor: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________ **Full Name: Shakir Ahmed** **Student ID: 1906074**	**Signature:** ____________ **Full Name: Md. Sharif Uddin** **Student ID: 1906077**
Signature: ____________ **Full Name: Md. Abu Sayed Chowdhury** **Student ID: 1906076**	**Signature:** ____________ **Full Name: Ahnaf Rashid** **Student ID: 1906086**

Table of Contents

1 Abstract **1**

2 Introduction **1**

3 Design **2**

3.1 Problem Formulation 2

3.1.1 Identification of Scope 2

3.1.2 Literature Review 2

3.1.3 Formulation of Problem 2

3.1.4 Analysis 2

3.2 Design Method 3

- Light and Fan Control: 3

The two IR sensors, temperature sensor and HEX key also sends their data to the Arduino. 3

- Fire Alarm Control: 3
- Security Measures: 3

3.3 Logic Diagram 4

- Human Count 4
- Fan and Light: 4
- HEX Keyboard 5
- Smoke/Fire 5

3.4 Hardware Design 6

3.5 Full Source Code of Firmware 6

4 Implementation **11**

4.1 Description 11

5 Design Analysis and Evaluation **12**

5.1 Novelty 12

5.2 Design Considerations 13

5.2.1 Considerations to public health and safety 13

5.2.2 Considerations to environment 13

5.2.3 Considerations to cultural and societal needs 14

5.3 Investigations 15

5.3.1 Literature Review 15

5.4 Limitations of Tools 16

5.5 Impact Assessment 16

5.5.1 Assessment of Societal and Cultural Issues 16
5.5.2 Assessment of Health and Safety Issues 17
5.5.3 Assessment of Legal Issues 18
5.6 Sustainability and Environmental Impact Evaluation 18
5.7 Ethical Issues 19
6 Reflection on Individual and Team work 21
6.1 Individual Contribution of Each Member 21
1. 1906074 - Shakir Ahmed 21
- Algorithm Formation 21
- Arduino Coding 21
2. 1906076 – Md. Abu Sayed Chowdhury 21
- ESP8266 Coding 21
- Testing Sensor and Building Primary Code 21
3. 1906077 - Md. Sharif Uddin 21
- Hardware Setup 21
- Debugging and Testing 21
4. 1906086 – Ahnaf Rashid 21
- Literature Review 21
- Model Idea 21
- Evaluating Impact and Sustainability 21
6.2 Mode of Teamwork 21
6.3 Diversity Statement of Team 21
6.4 Log Book of Project Implementation 21
7 Communication 22
7.1 Executive Summary 22
7.2 User Manual 22
8 Project Management and Cost Analysis 23
8.1 Bill of Materials 23
9 Future Work 24
10 . References 25

1 Abstract

This project presents the development of a home automation and safety system using Arduino, an open-source electronics platform. The system leverages various sensors to automate tasks and elevate the security and comfort of a living space. For automated lighting, **visitor counting algorithm, using IR senor is used**. Additionally, temperature sensors like DS18B20 will be integrated to regulate activate fans based on real-time temperature readings. Enhancing security and safety is another crucial aspect of the system. **Password protection** will be implemented on doors and windows to identify unauthorized openings and trigger alarms or send alerts. Moreover, smoke and gas sensors like MQ-2 and MQ-7 will be integrated to sound alarms and send notifications in case of emergencies, potentially preventing disasters. To offer user-friendly control and interaction, the system will explore various options. Bluetooth or Wi-Fi modules can be implemented to enable remote control and monitoring capabilities through a mobile application. Alternatively, voice control options using HC**-05**. Additionally, an LCD display can be integrated to provide real-time sensor readings, system status, and user controls. This project aims to contribute to creating a smarter and safer living environment. By automating tasks like lighting and temperature control, the system fosters convenience and energy efficiency. Furthermore, the integrated safety features provide peace of mind and the potential to prevent emergencies. By offering various user interaction options, the system caters to diverse preferences and ensures user-friendliness. Overall, this project demonstrates the versatility and potential of Arduino in creating a more comfortable, secure, and intelligent living space.

2 Introduction

In today's world, advancements in technology are constantly shaping how we interact with our surroundings. Home automation and safety systems are at the forefront of this innovation, offering increased convenience, security, and energy efficiency in our living spaces. This project delves into the development of such a system using Arduino, a widely accessible and versatile open-source electronics platform.

This project aims to create a smart and secure home environment by utilizing various sensors and actuators in conjunction with an Arduino microcontroller. Sensors will play a key role in gathering data about the environment, such as light levels, temperature, and motion. This data will then be processed by the Arduino, enabling it to automate tasks and trigger responses based on pre-defined parameters. Additionally, the system will incorporate safety features to safeguard the home against potential threats.

Throughout this project, various functionalities will be explored and implemented. Users will be able to benefit from automated lighting and temperature control, enhancing both comfort and energy efficiency. The system will also detect motion, allowing for automated responses like activating lights or sending notifications. Furthermore, security features will be integrated to monitor doors and windows for unauthorized access, detect smoke and gas leaks, and identify potential water leaks, providing peace of mind and the potential to prevent emergencies.

This project offers a practical and engaging opportunity to explore the world of electronics, programming, and home automation. By delving into sensor integration, automation techniques, and user interface design, the project fosters a deeper understanding of these concepts and their applications in real-world scenarios. By the project's completion, we aim to have successfully constructed a functional and user-friendly home automation and safety system that demonstrates the potential of Arduino in creating smarter and safer living environments.

3 Design

3.1 Problem Formulation

3.1.1 Identification of Scope

This project, beyond enhancing individual homes, holds potential to benefit society as a whole. By promoting energy efficiency through automation, it can contribute to a smaller collective environmental footprint. Additionally, improved security features like smoke/gas detection and water leak alerts can potentially reduce fire-related accidents and mitigate water damage, leading to safer and healthier communities. Furthermore, the open-source nature of Arduino fosters collaboration and adaptation, allowing for future iterations to address broader societal needs and contribute to a more sustainable and secure future for all.

3.1.2 Literature Review

Research demonstrates the feasibility and effectiveness of Arduino-based home automation systems. Studies by some researchers achieved significant energy savings through automated lighting and temperature control, while others explored security features like door/window monitoring and smoke/gas detection. Mobile app integration for remote control, offers user convenience. However, limitations exist. Existing projects often focus on specific functionalities, lacking comprehensive integration. Additionally, user interface complexity can limit accessibility. Sustainability considerations are also addressed by few. Building upon this foundation, this project aims to develop a home automation and safety system that offers a comprehensive solution, prioritizes user experience through an intuitive interface, and integrates sustainability considerations, contributing to the advancement of Arduino-based systems for a more user-centric, sustainable future.

3.1.3 Formulation of Problem

In the formulation of the problem, specific challenges within each identified aspect are addressed. This includes defining precise CPR performance metrics, resolving intricacies in baseline establishment, overcoming technical constraints for real-time monitoring, determining suitable wireless communication protocols, developing effective algorithmic comparisons, designing an intuitive user interface, addressing integration complexities, optimizing power consumption, creating robust testing protocols, and establishing comprehensive documentation guidelines. The problem formulation sets the stage for developing solutions that cater to these challenges, ensuring the successful design and implementation of the IoT-based device.

3.1.4 Analysis

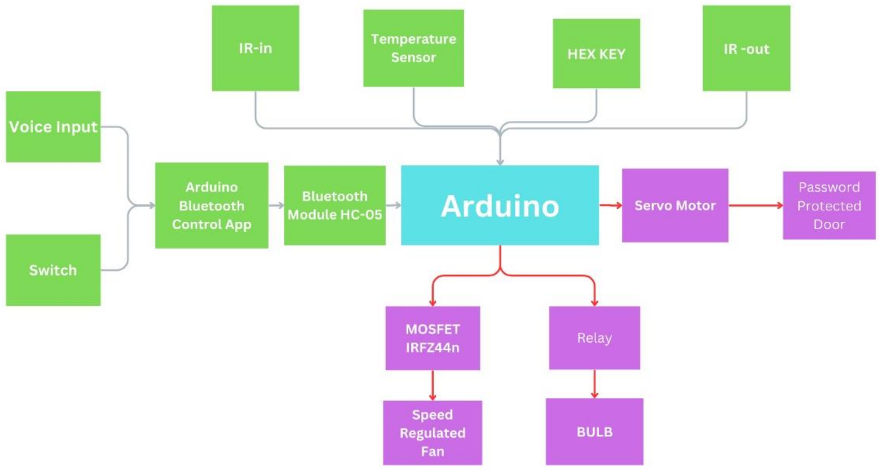
The analysis of the problem formulation reveals the complexity and interconnectedness of the identified challenges. Each element, from defining CPR metrics to establishing documentation requirements, presents unique considerations that impact the overall success of the project. For instance, addressing intricacies in baseline establishment directly influences the accuracy of real-time feedback, while designing an intuitive user interface impacts user understanding during CPR procedures. Successful resolution of these challenges is crucial for creating a device that not only adheres to medical standards but also enhances the user experience and contributes to more effective CPR interventions. The analysis highlights the multidimensional nature of the problem formulation and underscores the need for a holistic approach in designing the IoT-based device.

3.2 Design Method

- **Light and Fan Control:**

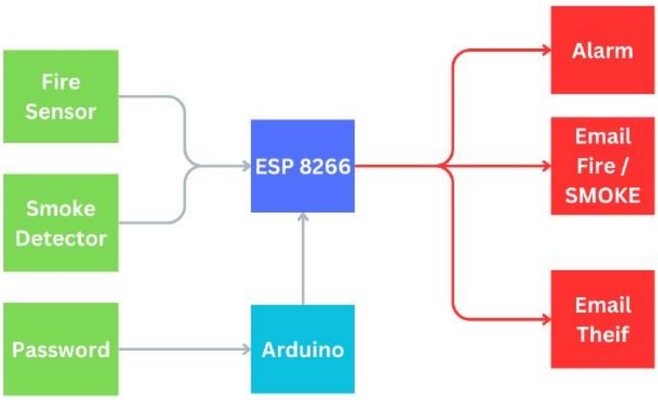
We have voice input and switches as our input methods. We can use either of them. The input signal then goes to the Arduino Bluetooth Control App and then get transferred to the Arduino through the Bluetooth Module HC-05.

The two IR sensors, temperature sensor and HEX key also sends their data to the Arduino. The output lines of the Arduino go to the Servo Motor which control the password protected door. Outputs also reach MOSFET IRFZ44n which controls the speed regulated fan and to a relay which is connected to a bulb.



- **Fire Alarm Control:**

Fire sensor and smoke detector sends their signal to ESP 8266. Which enables the fire alarm and sends an email to warn the user.

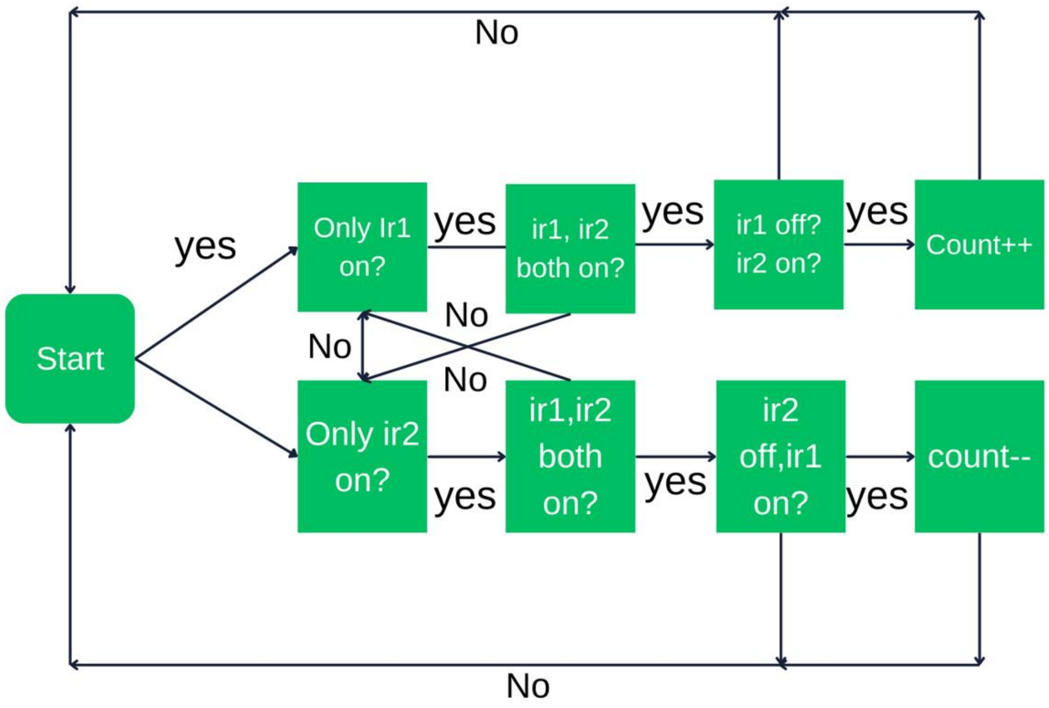


- **Security Measures:**

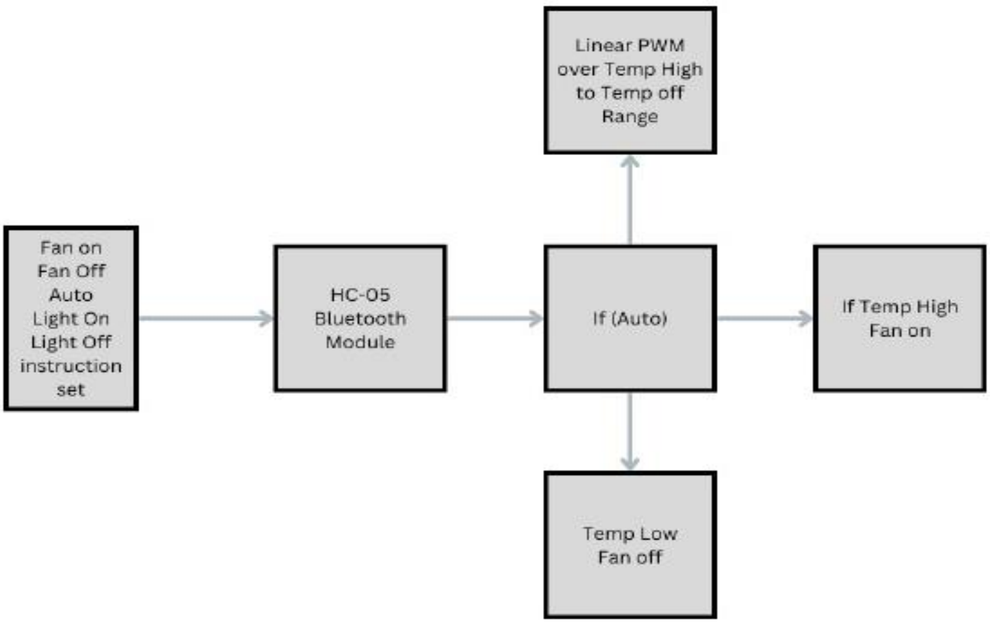
If the typed password is wrong for three times straight, the Arduino detects it and senses that there might be a security breach. Then it sends a signal to the ESP 8266 and an email to the user.

3.3 Logic Diagram

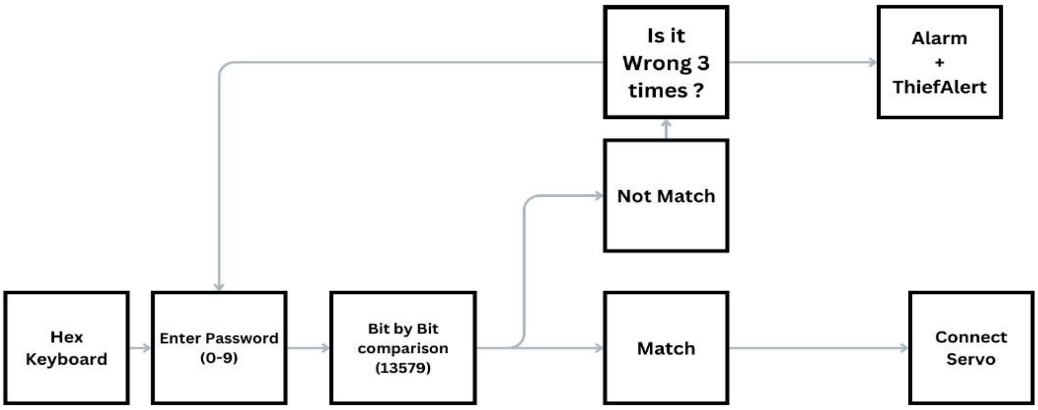
- **Human Count**



- **Fan and Light:**

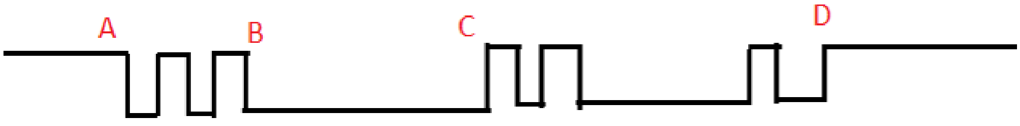


- **HEX Keyboard**

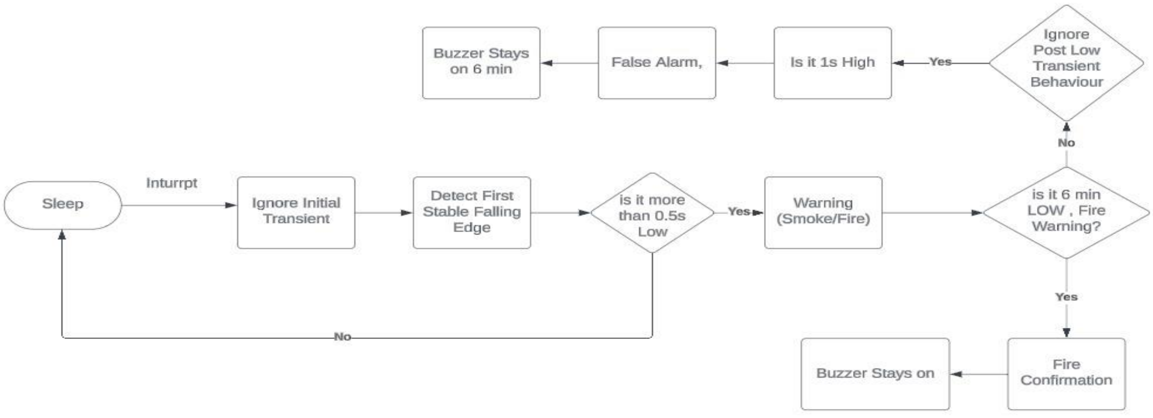


- **Smoke/Fire**

Sensor Data:



Algorithm To counter Transient:



3.4 Hardware Design

3.5 Full Source Code of Firmware

Column 1: Receiver Arduino Code
Column 2: Sender Arduino Code
Column 3: ESP8255 Email Alert Code

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 4
#include <LiquidCrystal.h>
#define tempup   35
#define tempdown 30

const int rs = 8, en = 9, d4 =
10, d5 = 11, d6 = 12, d7 = 13;
LiquidCrystal  lcd(rs,  en,  d4,
d5, d6, d7);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature
sensors(&oneWire);
#define ira 2
#define irb 3
int count=0;
int count0=-1;
int entry=0;
int mode=0;

float Celsius = 0;
int voice = '2';
int fan=6;
int bulb=5;
int duty=0;
int fan_auto=0;
void setup() {
  Serial.begin(9600);
  pinMode(ira,INPUT);
  pinMode(irb,INPUT);

TCCR1B=TCCR1B&0b11111000|0x02;
  lcd.begin(16, 2);
  pinMode(fan,OUTPUT);
  pinMode(bulb,OUTPUT);
}

void loop() {
  int ir1=digitalRead(ira);
  int ir2=digitalRead(irb);

sensors.requestTemperatures();
  Celsius                        =
sensors.getTempCByIndex(0);
  //Serial.print(Celsius);
 // Serial.print(" C  ");
  lcd.setCursor(0,0);
  lcd.print("T= ");
  lcd.print(Celsius);
  lcd.print("C");
  //////entry//////////////////
  if(ir1==1&&ir2==1){
    count=count;
    entry =0;
    mode=0;

  }

 if(ir1==0&&ir2==1){
    if(mode!=1){
      entry=entry+1;
    }
    mode=1;

    }

if(ir1==0&&ir2==0){
  if(mode!=2){
    entry=entry*10;
  }
  mode=2;

    }
if(ir1==1&&ir2==0){
   if(mode!=3){
    entry=entry-1;
   }
   mode=3;

    }

    if(entry==9) {
      count=count+1;
      entry=0;
      }
```

```
#include <SoftwareSerial.h>

 #include <Keypad.h>

 #include <Servo.h>

Servo servoMotor;  // Create a servo
object
SoftwareSerial bt(10,11); /* (Rx,Tx)
*/

int pos = 0;


const byte ROWS = 4;
const byte COLS = 3;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte rowPins[ROWS] = { 9, 8, 7, 6 };
byte colPins[COLS] = { 5, 4, 3 };

Keypad         customKeypad         =
Keypad(makeKeymap(hexaKeys),
rowPins, colPins, ROWS, COLS);
#define passlen 5
#define chorpin 13
#define servoPin 12
int check=0;
int match=0;
int len=0;
int temp=10;
int count=0;
//char
passin[]={'','','','','','','','',''
,'','',''};
char
pass[passlen]={'1','3','5','7','9'};
void setup(){
  Serial.begin(9600);
  bt.begin(9600);
  pinMode(chorpin,OUTPUT);
  digitalWrite(chorpin,HIGH);
  servoMotor.attach(servoPin);
}

void loop(){
   if (bt.available())      /*     If
data is available on serial port */
    {
     Serial.write("B");
     delay(20);
     Serial.write(bt.read());
    }
  char c = customKeypad.getKey();

  if (c){
  Serial.write("p");
  delay(20);
  Serial.write(c);
  if(c=='#'){

check=(int)((match==passlen)&&(len==
passlen));
          Serial.print(check);
         if(check==1)
         {
          servo();
         }
         len=0;
         match=0;
         if(check==0) count++;
         else count=0;
        }
   else{
      if(c=='*'&&len>0){
        len--;
        //match--;
      }
      else{
```

```
#include <ESP8266WiFi.h>

// //#include <Arduino.h>
 #if defined(ESP32)
 #include <WiFi.h>
 #elif defined(ESP8266)

 #endif
 #include <ESP_Mail_Client.h>
#define WIFI_SSID "Black devil"
#define WIFI_PASSWORD "eee19_3005"

//   #define WIFI_SSID "Mahin"
//   #define WIFI_PASSWORD "zxcvbnm,"

 /**    The    smtp    host    name    e.g.
smtp.gmail.com     for     GMail     or
smtp.office365.com   for   Outlook   or
smtp.mail.yahoo.com */
 #define SMTP_HOST "smtp.gmail.com"
 #define SMTP_PORT 465

// /* The sign in credentials */
 #define                    AUTHOR_EMAIL
"ssays2004@gmail.com"
 #define AUTHOR_PASSWORD "dluj bxcl yrif
bcyu"

 /* Recipient's email*/
 #define                 RECIPIENT_EMAIL
"abu.sayed.bueteee@gmail.com"

 /* Declare the global used SMTPSession
object for SMTP transport */
 SMTPSession smtp;

volatile int press=0;
#define fire D1
#define smoke D2
#define buzzar D5
#define button D6
#define thief D7
#define thiefWarn 5
#define confirm 1
int mailsent=0;
volatile unsigned long time_H = 0;
volatile unsigned long time_L= 0;
float gas=0;
volatile unsigned long button_time = 0;
volatile unsigned long time_now = 0;
int trig=-100;
int h=-1;
int edge=1;
void ICACHE_RAM_ATTR isr() {

detachInterrupt(digitalPinToInterrupt(s
moke));
  edge=edge^1;
  button_time = millis();
  if(edge==1)  time_H=button_time;
 // else  time_H=button_time;
  h=0;
  Serial.print(h);
    while(digitalRead(smoke)==0){
      if(millis()-
button_time>500&&trig!=0){
        press=5;
       //         channelNum          =
ISR_PWM.setPWM(buzzar,       PWM_Freq1,
PWM_DutyCycle1);
       analogWrite(buzzar,100);
        trig=0;
        Serial.print("press= ");
        Serial.println(press);
        time_L=millis();
        break;
      }
    }


attachInterrupt(digitalPinToInterrupt(s
moke), isr, CHANGE);
}

void setup() {
    Serial.begin(9600);
```

```
     if(entry==-9){
      count=count-1;
      entry=0;
     }
     if(1){
      Serial.print("count=");
      Serial.print(count);
        Serial.print(" ");
     }
     count0=count;
     delay(2);
 if(count){
   if (Serial.available()  > 0)
    {
      voice = Serial.read();
      Serial.print(voice);
      //Serial.print("\n");
       fan_auto=0;
      switch(voice){
    case                    '0':
digitalWrite(fan,LOW);
            break;
    case                    '1':
digitalWrite(fan,HIGH);
            break;
    case '2': fan_auto=1;
            break;
    case                    '3':
digitalWrite(bulb,HIGH);//
active low
            break;
    case                    '4':
digitalWrite(bulb,LOW);
            break;

    }
    }

    if(fan_auto){
    if(Celsius>tempup){
  digitalWrite(fan,HIGH);
  lcd.setCursor(0,1);
  lcd.print("Motor Fan is ON ");
 }
 else if(Celsius>=tempdown){

duty=map(Celsius,tempdown,tempu
p,1,255);
  analogWrite(fan,duty);
  lcd.setCursor(0,1);
  lcd.print("Motor Fan is ON ");
 }
 else
 { digitalWrite(fan,LOW);
   lcd.print("Motor  Fan  is  OFF
");
}
    }
 }
 else{
    digitalWrite(fan,LOW);
    fan_auto=1;
  }

     delay(2);

 }
```

```
if(c==pass[len]&&len<=passlen)
match++;
        len++;
      }
   }
  //  Serial.println(c);

  }

   if(temp!=check&&0){

    Serial.print(temp);
    Serial.println(check);
    }
    temp=check;
    if(count==3){
        chor();
        count=0;
    }
}


void chor()
{
  digitalWrite(chorpin,LOW);
  delay(5000);
  digitalWrite(chorpin,HIGH);
}


  void servo(){
       for (pos = 0; pos <= 90; pos
+= 3) {
    servoMotor.write(pos);        //
Set the servo position
    delay(20);                    //
Wait for a little while
  }
   delay(3000);
     for (pos = 90; pos >= 0; pos -=
3) {
    servoMotor.write(pos);        //
Set the servo position
    delay(20);                    //
Wait for a little while
  }

  }
```

```
    //timer1_enable(TIM_DIV16,
TIM_EDGE, TIM_SINGLE);
    pinMode(buzzar,OUTPUT);
    analogWrite(buzzar,0);
   // pinMode(pin,OUTPUT);
    pinMode(smoke,INPUT_PULLUP);
    pinMode(button,INPUT_PULLUP);
    pinMode(thief,INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(s
moke), isr, CHANGE);
}

int tempo=trig;
int x=1;
void loop() {
 // Serial.print(1);
  time_now=millis();
 if(edge==0&&millis()-
time_L>15000&&trig==0){
  trig=5;
  Serial.print("trig5= ");
  Serial.println(trig);
  }
 else              if(edge==1&&millis()-
time_H>1000&&trig==0) {
  trig=-100;
 Serial.print("trig1= ");
   analogWrite(buzzar,0);
  Serial.println(trig);
  h=1;
 }
  if(trig==5){
    analogWrite(buzzar,255);

Serial.println("smoke_confirm");
         if(x==1) {

//detachInterrupt(digitalPinToInterrupt
(smoke));
          setup1(confirm);

//attachInterrupt(digitalPinToInterrupt
(smoke), isr, CHANGE);
          x=0;
         }

  }
  if(digitalRead(button)==0){
     trig=-100;
        analogWrite(buzzar,0);
        x=1;
     h=1;
  }
  if(trig!=tempo){
  Serial.print("trig= ");
  Serial.println(trig);
  }
tempo=trig;
 // h=-1;
  if(press==5){
     Serial.print("smoke");
   setup1(2);
    press=0;
  }

if(digitalRead(thief)==0&&mailsent==0){
    setup1(thiefWarn);
    Serial.println("chor");
    mailsent=1;
  }
  else        if(digitalRead(thief)==1)
mailsent=0;

}

void setup1(int alert) {

  Serial.println();
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(300);
```

		`` } Serial.println(); Serial.print("Connected with IP: "); Serial.println(WiFi.localIP()); Serial.println(); /* Set the network reconnection option */ MailClient.networkReconnect(true); /** Enable the debug via Serial port * 0 for no debugging * 1 for basic level debugging * * Debug port can be changed via ESP_MAIL_DEFAULT_DEBUG_PORT in ESP_Mail_FS.h */ smtp.debug(1); /* Set the callback function to get the sending results */ smtp.callback(smtpCallback); /* Declare the Session_Config for user defined session credentials */ Session_Config config; /* Set the session config */ config.server.host_name = SMTP_HOST; config.server.port = SMTP_PORT; config.login.email = AUTHOR_EMAIL; config.login.password = AUTHOR_PASSWORD; config.login.user_domain = ""; /* Set the NTP config time For times east of the Prime Meridian use 0-12 For times west of the Prime Meridian add 12 to the offset. Ex. American/Denver GMT would be -6. 6 + 12 = 18 See https://en.wikipedia.org/wiki/Time_zone for a list of the GMT/UTC timezone offsets */ config.time.ntp_server = F("pool.ntp.org,time.nist.gov"); config.time.gmt_offset = 6; config.time.day_light_offset = 0; /* Declare the message class */ SMTP_Message message; /* Set the message headers */ message.sender.name = F("Saimon"); message.sender.email = AUTHOR_EMAIL; if (alert == confirm) { message.subject = F("Danger: Agun!!! Agun !!! Agun!!!"); } else if (alert == 2) { message.subject = F("Warning: Gas Leak!!! Gas Leak!!! Gas Leak!!!"); } else if (alert == thiefWarn) { message.subject = F("Theif!!! Theif!!! Theif!!!"); } else message.subject = F(""); message.addRecipient(F("CSE"), RECIPIENT_EMAIL); /*Send HTML message*/ /*String htmlMsg = "<div style=\"color:#2f4468;\"><h1>Hello World!</h1><p>- Sent from ESP board</p></div>"; message.html.content = htmlMsg.c_str(); message.html.content = htmlMsg.c_str(); ```

		message.text.charSet = "us-ascii"; message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;*/ //Send raw text message // if (alert==FireAlert){ // String textMsg = "Agun!!! Agun !!! Agun!!!"; // } // else if (alert==SmokeAlert){ // String textMsg = "Gas Leak!!! Gas Leak!!! Gas Leak!!!"; // } // else { // String textMsg = ""; // } String textMsg = " Your House Have Casualties. Come Back !!!"; message.text.content = textMsg.c_str(); message.text.charSet = "us-ascii"; message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit; message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_low; message.response.notify = esp_mail_smtp_notify_success | esp_mail_smtp_notify_failure | esp_mail_smtp_notify_delay; /* Connect to the server */ if (!smtp.connect(&config)) { ESP_MAIL_PRINTF("Connection error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(), smtp.errorCode(), smtp.errorReason().c_str()); return; } if (!smtp.isLoggedIn()) { Serial.println("\nNot yet logged in."); } else { if (smtp.isAuthenticated()) Serial.println("\nSuccessfully logged in."); else Serial.println("\nConnected with no Auth."); } /* Start sending Email and close the session */ if (!MailClient.sendMail(&smtp, &message)) Serial.println("Error sending Email, " + smtp.errorReason()); ESP_MAIL_PRINTF("Error, Status Code: %d, Error Code: %d, Reason: %s", smtp.statusCode(), smtp.errorCode(), smtp.errorReason().c_str()); } //void loop1(){ /* Callback function to get the Email sending status */ void smtpCallback(SMTP_Status status) { /* Print the current status */ Serial.println(status.info()); /* Print the sending result */ if (status.success()) {

		`` // ESP_MAIL_PRINTF used in the examples is for format printing via debug Serial port // that works for all supported Arduino platform SDKs e.g. AVR, SAMD, ESP32 and ESP8266. // In ESP8266 and ESP32, you can use Serial.printf directly. Serial.println("----------------"); ESP_MAIL_PRINTF("Message sent success: %d\n", status.completedCount()); ESP_MAIL_PRINTF("Message sent failed: %d\n", status.failedCount()); Serial.println("----------------\n"); for (size_t i = 0; i < smtp.sendingResult.size(); i++) { /* Get the result item */ SMTP_Result result = smtp.sendingResult.getItem(i); // In case, ESP32, ESP8266 and SAMD device, the timestamp get from result.timestamp should be valid if // your device time was synched with NTP server. // Other devices may show invalid timestamp as the device time was not set i.e. it will show Jan 1, 1970. // You can call smtp.setSystemTime(xxx) to set device time manually. Where xxx is timestamp (seconds since Jan 1, 1970) ESP_MAIL_PRINTF("Message No: %d\n", i + 1); ESP_MAIL_PRINTF("Status: %s\n", result.completed ? "success" : "failed"); ESP_MAIL_PRINTF("Date/Time: %s\n", MailClient.Time.getDateTimeString(result.timestamp, "%B %d, %Y %H:%M:%S").c_str()); ESP_MAIL_PRINTF("Recipient: %s\n", result.recipients.c_str()); ESP_MAIL_PRINTF("Subject: %s\n", result.subject.c_str()); } Serial.println("----------------\n"); // You need to clear sending result as the memory usage will grow up. smtp.sendingResult.clear(); } } ```

Table: Source Code for the main program

4 Implementation

4.1 Description

The Image of the Hardware

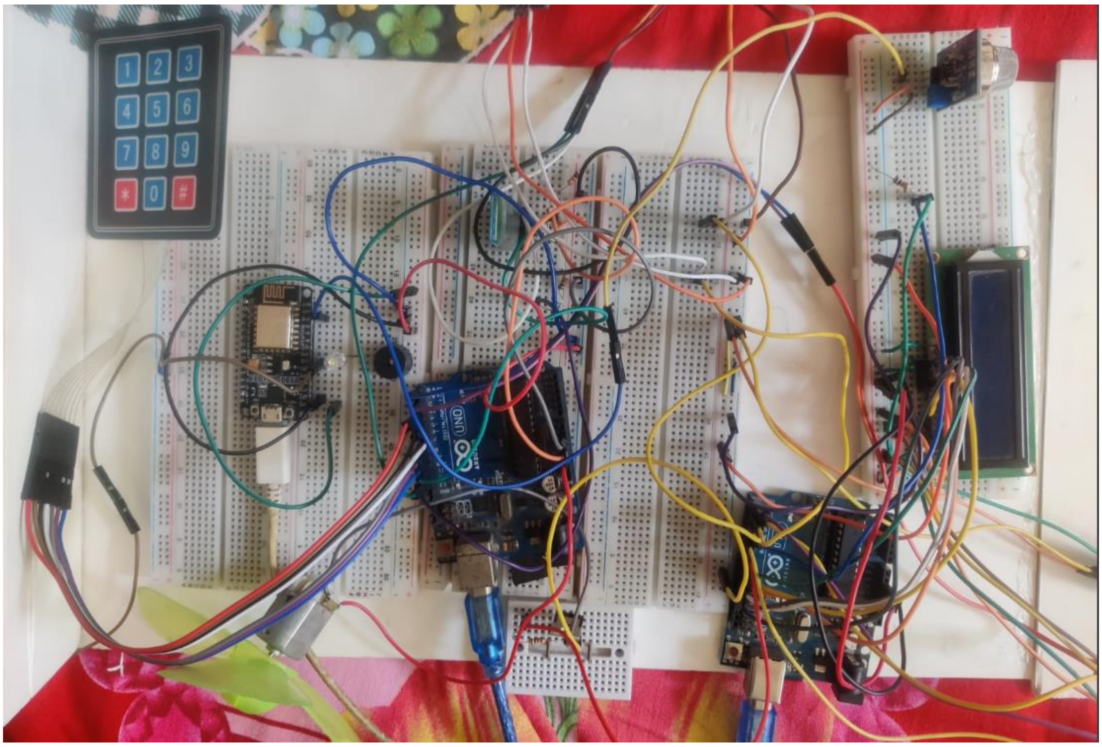


Figure 2: (Left) PCB Layout and (Right) Implementation of Design

5 Design Analysis and Evaluation

5.1 Novelty

While utilizing Arduino for home automation isn't entirely novel, this project aims to incorporate several unique aspects to enhance its functionality and user experience:

- **Combined Focus on Convenience, Security, and Energy Efficiency:** This project seeks to go beyond basic automation by offering a comprehensive system encompassing both comfort-enhancing features and robust security measures. This integrated approach aims to create a well-rounded living environment.
- **Exploration of Multiple User Interaction Options:** While traditional mobile app control exists, this project explores the possibility of incorporating voice control through an ESP32 or Raspberry Pi, offering users a hands-free and potentially more intuitive way to interact with the system.
- **Focus on Preventative Measures for Safety:** The project aims to go beyond simply detecting security threats by incorporating features like water leak detection, offering the potential to prevent emergencies and mitigate potential damage before issues escalate.
- **Customization and Modularity:** The system is designed with modularity in mind, allowing for future expansion and integration of additional functionalities based on evolving needs and preferences. This enables users to personalize and adapt the system to their specific requirements.

By incorporating these novel elements, this project aims to create a unique and user-centric home automation and safety system that surpasses the limitations of basic Arduino-based solutions.

5.2 Design Considerations

5.2.1 Considerations to public health and safety

This home automation and safety system using Arduino prioritizes public health and safety in several ways:

- **Early Leak Detection:** Integrating water leak detection sensors helps identify leaks promptly, potentially preventing significant water damage and the associated health risks like mold growth.
- **Improved Air Quality Monitoring:** Smoke and gas detection can alert residents to potential hazards like carbon monoxide leaks or fires, allowing for timely evacuation and preventing potential injuries or fatalities.
- **Intrusion Detection:** Monitoring doors and windows helps deter unauthorized entry, reducing the risk of theft and personal harm.
- **Fire and Gas Safety:** Prompt alerts about potential fires or gas leaks enable immediate action, minimizing the risk of injuries and property damage.
- **Mobile App and Voice Control (Optional):** These features allow users to remotely monitor and control the system, even when away from home, potentially enhancing security and providing peace of mind.
- **Flexible Design:** The modular design allows for customization to address specific needs, potentially improving accessibility for individuals with disabilities.
- **Adherence to safety guidelines:** During construction and installation, strict adherence to electrical safety guidelines is crucial to prevent electrical hazards and ensure the safe operation of the system.
- **Clear User Manual:** Providing a clear and comprehensive user manual will educate users on the safe operation and maintenance of the system, minimizing potential risks.

While this project prioritizes public health and safety, it is crucial to acknowledge that it is not a substitute for professional security systems or established emergency response protocols. Users should remain vigilant and contact the appropriate authorities in case of emergencies.

5.2.2 Considerations to environment

This home automation and safety system using Arduino incorporates several elements that contribute to environmental awareness and sustainability:

1. Reduced Energy Consumption:

- **Temperature Control Optimization:** Integrating temperature sensors allows the system to optimize thermostat settings or activate fans based on real-time needs, potentially leading to reduced energy use for heating and cooling.

2. Material Selection and Sourcing:

- **Sustainable Materials:** Whenever possible, prioritizing the use of environmentally friendly materials like recycled or recyclable components can minimize the environmental footprint of the system.
- **Sustainable Sourcing:** If feasible, sourcing components from companies with sustainable practices that prioritize ethical sourcing of materials and responsible manufacturing processes can contribute to a broader positive environmental impact.

3. Energy Efficiency Awareness:

- **User Education:** The project can incorporate educational elements within the user manual or mobile app to raise awareness about the energy-saving features of the system and encourage users to adopt additional sustainable practices in their daily lives.

It is important to acknowledge that while the project strives to be environmentally conscious, the overall environmental impact depends on various factors, including the specific components used, their manufacturing processes, and the disposal methods employed. By prioritizing these considerations, the project aims to contribute to creating a more sustainable home environment while maintaining its focus on user safety and convenience.

5.2.3 Considerations to cultural and societal needs

This home automation and safety system using Arduino aims to be culturally sensitive and address societal needs in the following ways:

1. Accessibility and User Interface:

- **Customizable Controls:** Design the system to allow users to personalize control methods and automation schedules to accommodate their cultural preferences and daily routines.

3. Socioeconomic Considerations:

- **Open-Source Platform and Cost-Effectiveness:** By utilizing Arduino, the project leverages an open-source platform, potentially making the system more affordable compared to commercially available solutions. This can make home automation more accessible to a wider range of socioeconomic backgrounds.
- **Scalability and Modularity:** The modular design allows users to start with a basic system and gradually add functionalities as budget allows, catering to diverse financial situations.

4. Cultural Sensitivity:

- **Respectful Design:** Consider potential cultural sensitivities related to noise levels, lighting preferences, and security protocols during the design phase. The system should be adaptable to accommodate different cultural norms.
- **Community Engagement:** For a broader societal impact, consider engaging with local communities to understand their specific needs and preferences. This can inform future iterations of the project to ensure it remains culturally relevant and addresses pressing societal concerns.

By incorporating these considerations, the project aims to create a home automation and safety system that is inclusive, adaptable, and respectful of diverse cultural and societal needs. The focus on affordability and user control empowers individuals to create a safer and more comfortable living environment within their own cultural context.

- .

5.3 Investigations

5.3.1 Literature Review

Voice-controlled home automation systems using Arduino have garnered considerable attention in recent years due to their affordability, versatility, and ease of implementation. This literature review provides an overview of the existing research and developments in this domain, focusing on the technology, design considerations, implementation approaches, and potential applications of voice-controlled home automation systems using Arduino.

The technology behind voice-controlled home automation systems using Arduino typically involves integrating Arduino microcontrollers with voice recognition modules or shields capable of processing voice commands. Researchers have explored various voice recognition modules, such as the EasyVR Shield, the HC-05 Bluetooth module, and the Arduino Voice Recognition Module, to enable voice interaction with Arduino-based devices. Additionally, advances in speech recognition algorithms and libraries, such as the Google Speech Recognition API and the PocketSphinx library, have contributed to enhancing the accuracy and responsiveness of these systems.

Designing a voice-controlled home automation system using Arduino requires careful consideration of hardware components, software libraries, and communication protocols. Researchers have investigated different Arduino board variants, such as the Arduino Uno, Arduino Mega, and Arduino Nano, based on the project's requirements in terms of processing power, memory, and input/output capabilities. Moreover, selecting appropriate voice recognition modules, sensors, actuators, and wireless communication modules is crucial for achieving desired functionality and performance. Additionally, researchers have emphasized the importance of optimizing power consumption, minimizing latency, and ensuring scalability and extensibility in the system design.

Various implementation approaches have been proposed for voice-controlled home automation systems using Arduino, ranging from standalone setups to cloud-connected architectures. Standalone setups typically involve processing voice commands locally on the Arduino board using onboard or external voice recognition modules. In contrast, cloud-connected architectures leverage internet connectivity to offload speech recognition and processing tasks to remote servers, enabling more complex voice interactions and integration with third-party services. Additionally, researchers have explored integrating Arduino-based systems with other smart home platforms and protocols, such as MQTT, Zigbee, and Wi-Fi, to enable seamless interoperability and remote control.

Voice-controlled home automation systems using Arduino have a wide range of potential applications in smart homes, IoT projects, educational environments, and DIY projects. In smart homes, these systems enable users to control lights, appliances, security cameras, and HVAC systems using voice commands, enhancing convenience and energy efficiency. Moreover, voice-controlled Arduino projects have been used in educational settings to teach programming, electronics, and IoT concepts to students of all ages. Additionally, DIY enthusiasts and hobbyists have leveraged Arduino-based voice control for various creative projects, such as voice-activated robots, interactive installations, and personalized assistants.

In conclusion, voice-controlled home automation systems using Arduino offer a cost-effective and customizable solution for implementing voice interaction in smart home environments. While significant progress has been made in technology, design, and implementation approaches, there is still room for improvement in terms of accuracy, reliability, and ease of use. Future research efforts should focus on refining voice recognition algorithms, expanding hardware and software capabilities, and exploring novel applications and integration possibilities. Overall, voice-controlled home automation systems using Arduino hold great potential to democratize access to smart home technology and empower individuals to create personalized and intelligent living spaces

5.4 Limitations of Tools

While the project offers several functionalities, it's important to acknowledge its limitations:

Technical limitations:

- **Complexity for non-technical users:** Setting up and operating the system might require some technical knowledge, especially if using complex features or coding.
- **Limited scalability:** The current design might not be easily scalable to accommodate larger or more complex living environments.
- **Reliance on external power:** The system depends on a constant power supply, and disruptions could render some features inoperable.

Functional limitations:

- **Limited sensor range:** Sensors have a specific range, and their effectiveness might be limited in larger spaces or complex layouts.
- **False alarms:** Sensors, particularly motion detectors, can be susceptible to false alarms triggered by pets, lighting changes, or other environmental factors.
- **Integration with existing systems:** Compatibility with existing home automation systems or smart devices might be limited, requiring additional components or configurations.

Cost and resource limitations:

- **Initial investment:** The cost of components and any required additional hardware (e.g., mobile app development) can be a barrier for some users.
- **Technical support and maintenance:** Users might need to possess technical skills or seek external assistance for troubleshooting and maintenance.
- **E-waste generation:** The use of electronic components contributes to e-waste, and responsible disposal practices are crucial.

5.5 Impact Assessment

5.5.1 Assessment of Societal and Cultural Issues

This assessment analyzes the societal and cultural implications of the home automation and safety system you're developing with Arduino. It considers both positive and potential negative aspects:

- **Improved Safety and Security:** The system's intrusion detection, smoke/gas monitoring, and water leak detection features can significantly enhance home security and potentially reduce crime rates and fire-related accidents, benefiting society as a whole.
- **Increased Accessibility and Independence:** Features like automated lighting and temperature control can improve accessibility for individuals with disabilities, allowing them to maintain greater independence within their homes. The modular design also allows for customization to address specific needs.
- **Environmental Sustainability:** The project promotes energy efficiency through automated lighting and temperature control, potentially leading to reduced household energy

consumption and a smaller collective environmental footprint.

- **Technological Literacy and Innovation:** Exposure to Arduino and home automation principles can spark interest in STEM fields among users, fostering innovation and technological literacy within society.
- **Digital Divide and Affordability:** While Arduino offers a relatively affordable platform, the overall cost of components and technical knowledge required for setup could create a barrier for low-income households, exacerbating the digital divide.
- **Privacy and Security Concerns:** The system collects sensor data, and vulnerabilities could lead to privacy breaches. Transparency about data collection and robust security measures are crucial to address societal concerns.
- **Cultural Sensitivity:** Automation schedules, noise levels, and security protocols might not align with all cultures. The design should be adaptable to accommodate diverse preferences and avoid imposing a single cultural norm.
- **Focus on Inclusivity:** Explore options for multilingual interfaces and culturally sensitive design features.
- **Prioritize Data Security:** Implement robust security measures and clearly communicate data practices to users.
- **Promote Education and Awareness:** Develop educational materials about the project's societal benefits and potential drawbacks.
- **Open-Source Sharing:** Share the project's design openly to encourage collaboration and adaptation for broader societal impact.

By carefully considering these societal and cultural aspects, you can ensure your home automation system is not only technologically sound but also contributes to a more inclusive, secure, and sustainable future.

5.5.2 Assessment of Health and Safety Issues

This assessment analyzes the potential health and safety risks associated with the home automation and safety system developed with Arduino:

- **Adhere to Safety Standards:** Strictly follow electrical safety guidelines during installation and ensure all components comply with relevant safety standards.
- **Robust Hardware and Software Design:** Utilize reliable components and implement robust coding practices to minimize the risk of hardware failure or software bugs.
- **Data Security Measures:** Prioritize data security by encrypting sensitive data, implementing user authentication, and regularly updating software to address potential vulnerabilities.
- **User Education and Training:** Provide clear user manuals and training materials to educate users on safe installation, operation, and maintenance procedures.
- **Emergency Protocols:** Develop clear emergency protocols outlining actions to take in case of system malfunctions or other unforeseen circumstances.
- **Power Outages:** Consider incorporating backup power solutions to ensure critical safety features like smoke and gas detection remain functional during power outages.
- **User Capabilities:** Be mindful of the user's technical abilities and physical limitations during the design phase. The system should be user-friendly and avoid creating new safety hazards for individuals with specific needs.

By implementing these mitigation strategies and carefully considering the potential health and safety risks, we can create a home automation and safety system that enhances the well-being and security of its users.

5.5.3 Assessment of Legal Issues

- **Product liability:** In the event of a system malfunction causing harm or property damage, you might be held liable depending on the specific circumstances and applicable laws.
- **Data privacy:** If the system collects any personal data (e.g., through sensors), you need to comply with data privacy regulations like the General Data Protection Regulation (GDPR) in the European Union or the California Consumer Privacy Act (CCPA) in the United States.
- **Intellectual property:** Ensure you have the necessary rights to use any software or hardware components employed in the system, and avoid infringing on any existing intellectual property rights.
- **Building codes and regulations:** Depending on your location, there might be specific building codes or regulations governing the installation and use of home automation systems.

5.6 Sustainability and Environmental Impact Evaluation

This evaluation explores the potential positive and negative environmental impacts of your Arduino-based home automation and safety system-

- **Reduced Energy Consumption:** Automating lighting and temperature control based on real-time needs can significantly decrease energy usage within the home. This translates to a lower carbon footprint and reduced reliance on non-renewable energy sources.
- **Improved Water Conservation:** Early leak detection through sensors can prevent burst pipes or prolonged leaks, minimizing water waste and its associated environmental impact.
- **Increased Awareness and Behavior Change:** The system can provide real-time feedback on energy and water consumption, potentially motivating users to adopt more sustainable practices in their daily lives.
- **Environmental Impact of Manufacturing:** The manufacturing processes for electronic components can have environmental consequences, including resource extraction, energy use, and pollution.
- **Durable and Long-lasting Components:** Selecting high-quality, durable electronic components can extend the system's lifespan and reduce the frequency of replacements, minimizing e-waste generation.
- **Responsible Disposal Practices:** Develop a plan for responsible disposal or recycling of electronic components at the end-of-life stage, adhering to local regulations and e-waste management best practices.
- **Prioritize Sustainable Materials:** Whenever possible, utilize components with recycled content or those manufactured with sustainable practices to minimize the environmental footprint.
- **Energy-efficient Components:** Select low-power consumption components for the system to maximize energy savings.
- **Promote Sustainable Practices:** Integrate educational elements within the user interface or mobile app to raise awareness about the system's energy-saving features and encourage users to adopt broader sustainable habits.
- **Local Context:** Research local regulations and initiatives related to e-waste management and incorporate responsible disposal practices into your project.
- **Life Cycle Assessment (LCA):** Consider conducting a simplified LCA to assess the

environmental impact of the system throughout its lifecycle, from material extraction to disposal.

The environmental impact of the home automation system depends on various factors, including the specific components used, their manufacturing processes, energy consumption patterns, and disposal methods. By prioritizing the strategies mentioned above, we can significantly reduce the negative environmental impact and enhance the project's overall sustainability.

5.7 Ethical Issues

While the project offers several potential benefits, there are also ethical considerations that require careful thought and mitigation strategies:

- **Privacy and Data Security:**
  - **Data Collection and Usage:** The system collects sensor data, raising concerns about user privacy. Ensure transparent communication about data collection practices, including the type of data collected, its purpose, and how it is stored and secured.
  - **Data Security Vulnerabilities:** Security breaches could expose user data or allow unauthorized access to control features of the system. Implement robust security measures like encryption, user authentication, and regular software updates to minimize these risks.
  - **User Control and Transparency:** Users should have control over what data is collected and how it is used. Allow users to opt-out of data collection or choose what information they share, and provide clear and transparent information about data usage policies.

- **Accessibility and Equity:**
  - **Cost and Affordability:** While Arduino offers a relatively affordable platform, the system might still be out of reach for low-income households, exacerbating the digital divide. Consider exploring cost-reduction strategies or collaboration with community organizations to improve accessibility.
  - **Technical Literacy and User Skills:** The project requires some technical knowledge for setup and operation, potentially excluding individuals with limited technological literacy. Offer clear instructions, user manuals, and consider developing training materials or workshops to improve accessibility.
  - **Cultural Sensitivity:** Automation schedules, security protocols, and comfort levels might not align with all cultures. Ensure the system is adaptable to accommodate diverse preferences and avoid imposing a single cultural norm.

- **Environmental Impact:**
  - **E-waste Generation:** Electronic components contribute to e-waste if not disposed of responsibly. Promote responsible disposal practices and consider the life cycle of components when making selections.
  - **Manufacturing Processes:** The manufacturing of electronic components can have environmental consequences. Explore options for components with recycled content or those manufactured with sustainable practices.
  - **Energy Consumption:** While the system aims to save energy, ensure the energy consumption of the system itself and associated devices (e.g., powered by the system)

doesn't negate the intended energy savings.

- **Ethical Use and Misuse:**
    - **Potential for Malicious Use:** Vulnerabilities could be exploited by individuals with malicious intent, causing harm or manipulating the system for personal gain. Implement robust security measures and educate users about responsible use and potential risks.
    - **Over-reliance on Automation:** Overdependence on automation could lead to a decrease in user vigilance and preparedness in real emergencies. Encourage users to maintain awareness and critical thinking skills alongside the system's features.

6 Reflection on Individual and Team work

6.1 Individual Contribution of Each Member

1. **1906074 - Shakir Ahmed**

- **Algorithm Formation**
- **Arduino Coding**

2. **1906076 – Md. Abu Sayed Chowdhury**

- **ESP8266 Coding**
- **Testing Sensor and Building Primary Code**

3. **1906077 - Md. Sharif Uddin**

- **Hardware Setup**
- **Debugging and Testing**

4. **1906086 – Ahnaf Rashid**

- **Literature Review**
- **Model Idea**
- **Evaluating Impact and Sustainability**

6.2 Mode of Teamwork

6.3 Diversity Statement of Team

6.4 Log Book of Project Implementation

Week	Milestone achieved

1-3	Finding the right topic for the project
4	Project Proposal Presentation
5	Project was selected. List of components was prepared
6	Using IR sensor, counting the number of humans in the room was done
7	Temperature was measured using sensor
8	Speed control of fan according to the temperature
9	First update and presentation
10	Voice command
11	Fire and smoke detector, password protection
12	Uploading in the internet
13	Hardware implementation of the project
14	Final presentation and report submission

7 Communication

7.1 Executive Summary

Introducing a Smarter and Safer Home with Arduino!

5-3-2024
Dhaka, Bangladesh

Calling all tech enthusiasts and safety champions! A new project is underway, developing a home automation and safety system using Arduino, an accessible and user-friendly electronics platform.

Imagine lights that adjust automatically, saving energy and money. Or, receive alerts if smoke or leaks are detected, keeping your home and family safe. This project aims to make these features a reality, offering a glimpse into the future of smart living.

Stay tuned for further updates as we delve into the exciting world of sensors, automation, and creating a smarter, safer home environment!

7.2 User Manual

Introduction:
This user manual provides instructions for operating the home automation and safety system built with Arduino. This system offers various features to enhance comfort, security, and energy efficiency in your living space.

Features:
Automated Lighting: Lights turn on/off based on ambient light levels (using LDRs).

5. Temperature Control: Thermostats or fans adjust based on real-time temperature (using temperature sensors).
6. Motion Detection: Triggers actions like light activation or notifications (using IR sensors).

Mobile App Control:

- Download the dedicated app from play store.
- Connect your phone to the system's Wi-Fi network.
- The app allows you to monitor sensor readings, control features remotely, and receive alerts.

LCD Display:
The LCD displays real-time sensor readings, system status, and user controls.
Navigate the display using the buttons provided.

Important Safety Information:
Electrical Safety: Ensure proper installation and follow electrical safety guidelines.
System Malfunctions: In case of malfunctions, disconnect the power supply and consult the project developer.

Disclaimer:
This system is for educational purposes only and should not be considered a substitute for professional security systems or established emergency response protocols.

8 Project Management and Cost Analysis

8.1 Bill of Materials

Component	Number	Cost
Arduino Uno	1	800
Temperature Sensor(DSB18B20)	1	220
Fire sensor	1	80
IR Sensor	2	120
LCD display	1	200
Bulb	1	30
Relay module	1	180
HC05 Bluetooth Module	1	240
ESP8266 WIFI Module	1	200
DC motor	1	60
Battery(9V)	1	70

Resistor	5	10
Capacitor	2	6
Diode	2	4
Transistor	2	10
Breadboard	2	220
Total Cost	-	2450

9 Future Work

Building upon the current foundation, several exciting avenues exist for future exploration and development of the home automation and safety system:

- **Enhanced Functionality:**

Voice control integration: Integrate voice assistants like Amazon Alexa or Google Assistant for hands-free control of the system.

Advanced security features: Explore options for facial recognition, fingerprint scanners, or smart locks for heightened security measures.

Appliance control: Integrate with smart appliances like thermostats, refrigerators, or washing machines for a more comprehensive smart home experience.

Remote monitoring and control: Allow users to remotely monitor and control the system from anywhere with an internet connection, offering peace of mind while away from home.

- **Improved User Experience:**

Machine learning integration: Implement machine learning algorithms to personalize user experience by adapting automation based on individual preferences and routines.

Enhanced user interface: Develop a user-friendly and intuitive interface for both the mobile app and LCD display, ensuring ease of use for all users.

Multilingual support: Implement multilingual functionality in the mobile app to cater to a wider audience and promote accessibility.

- **Advanced System Features:**

Interconnectivity: Explore smart home ecosystems and enable the system to connect with other smart devices like smart speakers, lighting systems, or security cameras, creating a truly interconnected smart home network.

Energy optimization: Integrate with smart energy meters to monitor energy consumption patterns and implement strategies for optimizing energy usage.

Self-diagnostic capabilities: Develop the system to diagnose potential issues or malfunctions and notify users for preventive maintenance.

- **Sustainability and Ethical Considerations:**

Focus on energy-efficient components: Option for low-power consumption components and explore renewable energy sources like solar panels to power the system.
Sustainable material selection: Prioritize the use of recycled or recyclable materials during development and construction.

Promote responsible e-waste management: Develop a plan for responsible disposal or recycling of electronic components at the end of the system's lifespan.

Continued ethical assessment: Regularly review and address potential ethical concerns related to data privacy, accessibility, and potential misuse of the system.
By pursuing these future works and development areas, the home automation and safety system with Arduino can evolve into a robust, user-centric, and sustainable solution, contributing to a more secure, convenient, and environmentally conscious living environment.

10. References

1.Email Alert using ESP8266**https://randomnerdtutorials.com/esp8266-nodemcu-send-email-smtp-server-arduino/**

2.Interfacing keypad using Arduino https://electropeak.com/learn/interfacing-4x3-membrane-matrix-keypad-with-arduino/

3.HC-05 Bluetooth module interfacing with Arduino https://www.electronicwings.com/arduino/hc-05-bluetooth-module-interfacing-with-arduino-uno

4.ESP8266 interrupt service routine protocol https://arduinokitproject.com/handling-esp8266-gpio-interrupts-tutorial/

5.Serial Communication between two Arduino https://iot-guider.com/arduino/serial-communication-between-two-arduino-boards/