Chapter 1: Introduction

1.1 Necessity

In an era where both urban and rural households are becoming increasingly reliant on technology, home security demands systems that go beyond basic locks and cameras. Conventional alarm systems are limited by their reactive nature—they alert but do not respond. Our project, *“Intelligent Multi-Sensor Based Home Security System with Automated Threat Response”*, bridges this critical gap by combining proactive threat detection with automated, intelligent responses.

The system is tailored for Indian households, where electrical fluctuations, unpredictable weather, and affordability must be considered in the design. By utilizing a fusion of sensors and automated hardware, our solution not only strengthens security but also enhances convenience, energy efficiency, and environmental responsiveness.

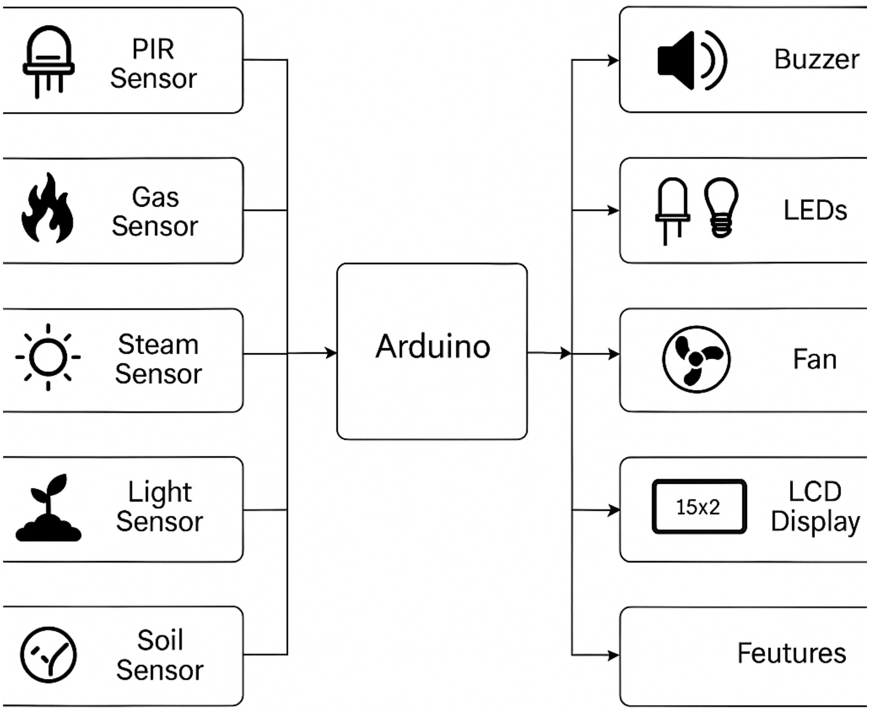


Figure 1 Functional Block diagram of the Home Security System

1.2 Function

Our system operates as a standalone, real-time smart controller for home safety and automation. It intelligently processes data from multiple sensors and triggers a sequence of automated actions based on defined thresholds and environmental conditions.

Core Functions:

- **Intrusion Detection:** Identifies motion and plays a buzzer alert while updating the LCD display.
- **Gas Leakage Control:** Activates exhaust fan and visual warning when gas is detected, preventing buildup.
- **Rain/Steam Management:** Closes windows automatically to prevent water damage using a servo.
- **Lighting Automation:** Turns on lights in low-light conditions, offering better visibility.
- **Soil Monitoring:** Notifies users when plants require watering.
- **Password-Activated System Control:** Prevents unauthorized access using a Morse-code-style button input.

Figure 2 Hardware of the Security System

1.3 Advantages

Our system stands apart due to its versatility, modular design, and deep integration between software logic and hardware response. Unlike single-purpose security systems, ours is multi-functional and proactive.

Key Advantages:

1. Real-time automated response without user intervention.
2. Multi-layered environmental monitoring.
3. Low-cost and open-source hardware (Arduino-based).
4. Morse-based password system for secure access—innovative and interactive.
5. Compact design, requiring minimal space and wiring.
6. Expandable architecture for future integration with IoT or mobile apps

Feature	Conventional Home Security System	Our System
Automation	Limited	Advanced
Multi-Sensor	No	Yes
Real-Time Feedback	No	Yes
Cost	High	Low
Scalability	Limited	High

Table 1 Feature Comparison

1.4 Features

This project incorporates multiple layers of automation and security, making it a comprehensive solution rather than a patchwork of devices. Its major features include:

- Simultaneous monitoring of gas, motion, steam, soil, and light.
- LCD feedback system for real-time status updates.
- Automated window and door control using servo motors.
- Smart lighting control through light sensors.
- Fan control and relay actuation in hazardous conditions.
- Audio feedback via buzzer, including melodies for specific events.
- Compact PCB-ready layout for cleaner integration.
- Morse-code password input for unique system access.

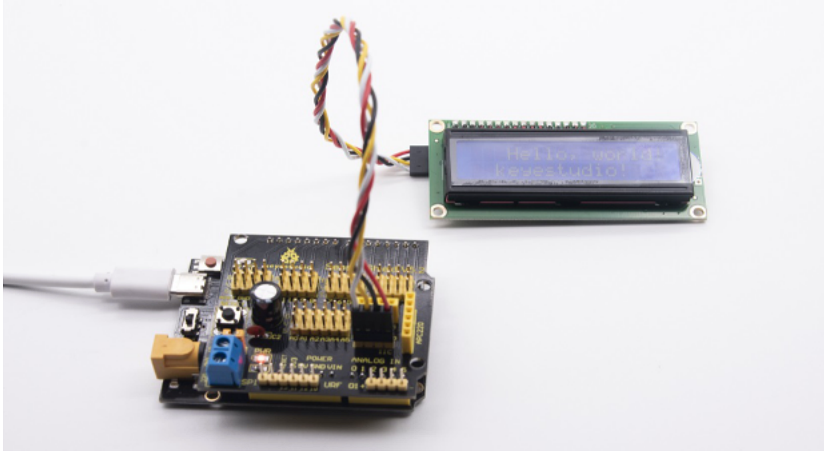


Figure 3 LCD Testing

Chapter 2: Literature Review

2.1 Introduction

The concept of intelligent home security systems has evolved rapidly in the last decade, driven by the convergence of embedded systems, sensor networks, and real-time automation technologies. As urbanization intensifies and smart infrastructure becomes mainstream, the need for systems that not only monitor but also **respond autonomously to environmental threats** has become crucial.

Traditionally, home security focused on **surveillance and deterrence**, often relying on CCTV and alarm systems. However, these are reactive and limited in scope. The current trend emphasizes **proactive systems** capable of detecting and responding to a variety of conditions such as motion, gas leaks, humidity, and environmental changes.

This chapter critically examines the **key literature, technologies, and innovations** that lay the groundwork for our proposed project: **"Design and Development of an Intelligent Multi-Sensor Based Home Security System with Automated Threat Response**."

2.2 Smart Home Security: An Overview

Home security systems have transitioned from lock-and-key mechanisms to comprehensive, **context-aware smart systems**. Early systems provided only basic intrusion alerts, often with high false positives. Contemporary literature reflects a shift towards **sensor-driven, real-time, embedded automation**.

Al-Ali et al. (2010) demonstrated a GSM-based wireless home automation system that remotely monitors temperature, smoke, and motion sensors. Similarly, Kumar et al. (2014) developed a ZigBee-based security solution emphasizing real-time environmental monitoring.

Academic work, unlike commercial products such as **Google Nest** or **Amazon Ring**, focuses on **open-source, cost-effective solutions** tailored to regional requirements. For example, integrating **gas leak detection in kitchens** or **soil moisture sensors for garden automation** is common in academic systems, but lacking in off-the-shelf commercial offerings.

2.3 Embedded Systems in Home Automation

Embedded systems are the foundation of intelligent automation. Microcontrollers like **Arduino Uno** provide a low-cost, highly customizable platform for collecting sensor data, processing it, and actuating outputs.

Most designs follow a **Sense** → **Process** → **Actuate** model:

- **Sense:** Capture environmental parameters via sensors (motion, gas, light, steam, soil).
- **Process:** Use threshold logic or conditions to evaluate risks.
- **Actuate:** Respond using hardware (e.g., servo motor, buzzer, fan, LCD).

Gupta & Kalra (2016) emphasize that Arduino-based architectures reduce design complexity and enable real-time monitoring without needing high-end computational resources.

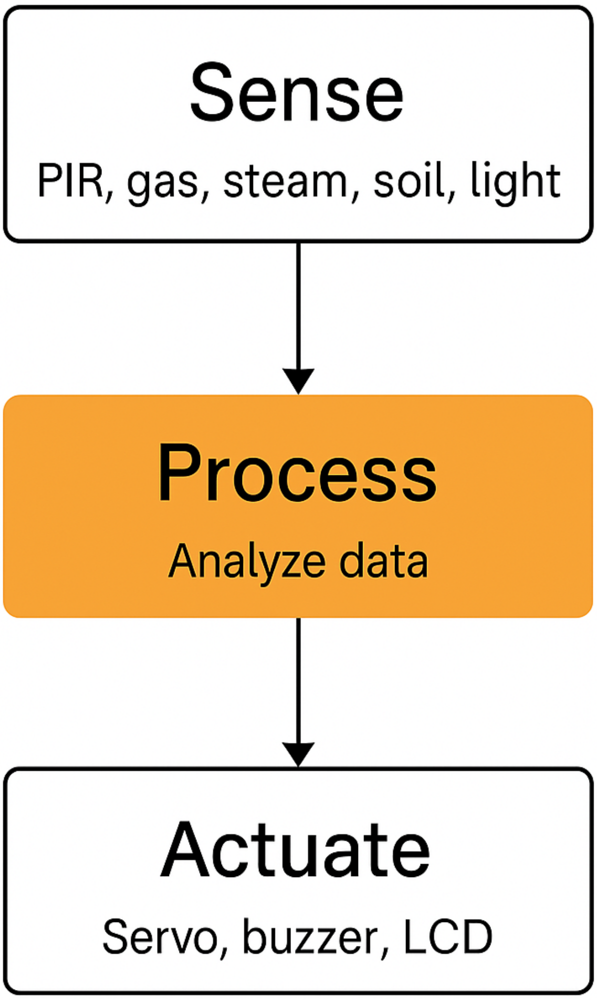


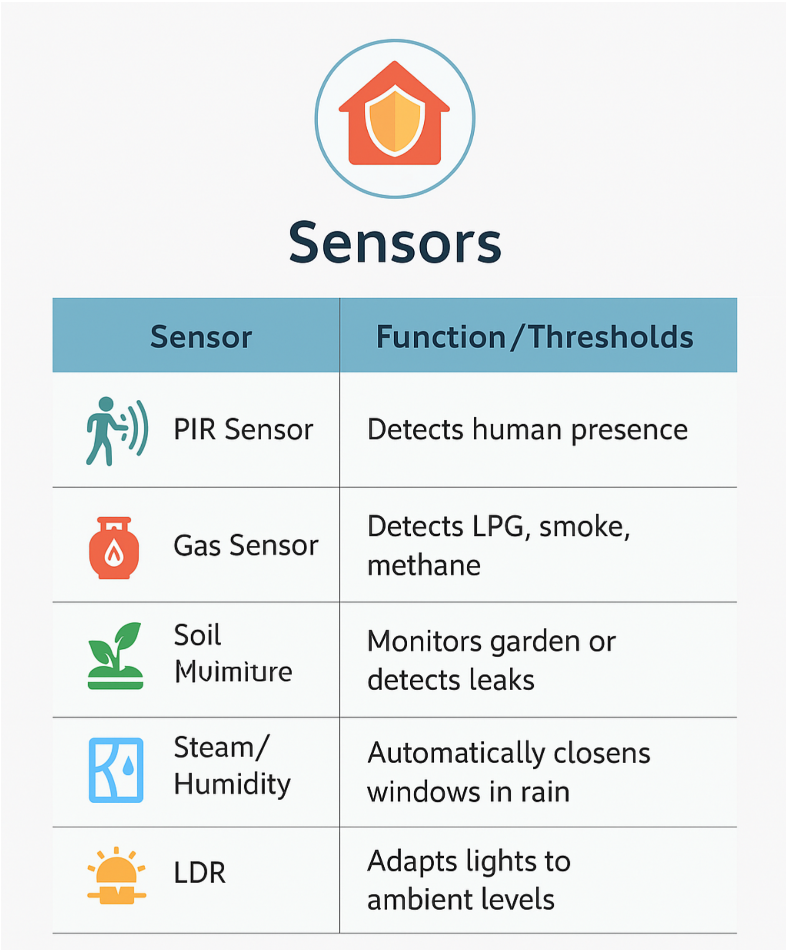
Figure 4 Microcontroller-Centric Automation Flow

2.4 Sensor Technologies for Environmental Monitoring

A **multi-sensor approach** significantly improves system reliability and reduces false alerts by offering cross-verification of environmental states. Key sensors reviewed include:

- **PIR Sensor:** Widely adopted for motion detection due to low power usage and high accuracy (Raza et al., 2015).

- **MQ2 Gas Sensor:** Detects LPG, methane, smoke. Proven effective in indoor safety systems (Hasan et al., 2018).
- **Soil Moisture Sensor:** Enables automated irrigation or water-leak alerts. Used in smart agriculture and water conservation systems.
- **Steam/Humidity Sensor:** Allows automatic closure of windows in response to high moisture; applied in weather-responsive buildings (Lee et al., 2019).
- **LDR (Light Dependent Resistor):** Controls lighting based on ambient light, enhancing energy efficiency.


Sensors

Sensor	Function/Thresholds
PIR Sensor	Detects human presence
Gas Sensor	Detects LPG, smoke, methane
Soil Muimiture	Monitors garden or detects leaks
Steam/ Humidity	Automatically closens windows in rain
LDR	Adapts lights to ambient levels

Figure 5 Sensors and Their Roles in the Proposed System

2.5 Actuators and Alert Mechanisms

Automation is only meaningful when sensing is followed by **physical or visual response**. Literature strongly supports coupling environmental detection with actuation mechanisms:

- **Servo Motors:** Enable precision control over window/door movement. Their low energy use and accuracy make them ideal for home applications (Zhang et al., 2012).

- **Buzzers and LEDs:** Serve as non-intrusive but immediate alert tools. Fernandes & Costa (2017) show enhanced user interaction when melodic tones replace harsh alarms.
- **Relays and Fans:** Smart exhaust systems are activated based on gas or steam levels. These are especially valuable in kitchens and bathrooms.

These actuators **close the automation loop**, turning passive alerts into real-time intervention.

2.6 Access Control Mechanisms

While RFID and PIN-based systems are common in access control, this project uses a **Morse-code-inspired password input**. This approach has rarely been applied in home automation, offering a unique blend of **security and minimal hardware complexity**.

Studies like Patel & Mishra (2020) show Morse-input systems to be resilient to brute-force attacks and user-friendly when limited input interfaces are available. The method also adds a novel interactive element to the system.

2.7 Comparison with Existing Systems

System	Sensing Capabilities	Response	Scalability	Cost	Autonomy
Commercial (Nest, Ring)	Motion, Camera	Manual alert, app notification	Medium	High	Medium
Academic – Single Sensor	PIR or Gas sensor only	LED/Buzzer only	Low	Low	Low
Our Proposed System	Multi-sensor, context-aware logic	Servo, fan, buzzer, LCD	High	Very Low	**High**

Table 2 Comparison with existing systems

2.8 Summary

The literature reveals a **critical need for systems that combine multi-sensor awareness with embedded intelligence**. Most commercial systems are limited in scope and heavily dependent on user intervention. Academic prototypes are often constrained by cost and feature only one or two sensors.

Our proposed system fills this gap by offering:

- Multi-modal sensing,
- Context-aware logic,
- Real-time actuation,
- Cost-effective and scalable design.

By grounding this work in well-established research and addressing clear gaps, our project aspires to offer a **holistic, intelligent, and proactive security solution** for modern homes.

Chapter 3: Problem Formulation

3.1 Objective

The primary objective of this project is to design and implement an **“Intelligent Multi-Sensor Based Home Security System with Automated Threat Response”** that ensures real-time monitoring and autonomous actions in response to environmental and intruder-related events. The system aims to replace conventional passive alarm mechanisms with **smart, context-aware decision-making** to enhance the safety, usability, and affordability of home security systems.

3.2 Problem Statement

Modern home security systems are often either limited in their sensing capabilities or dependent on cloud platforms and mobile applications for response initiation. This presents two critical shortcomings:

1. **Delayed response time** due to human dependence.
2. **Lack of environmental intelligence**, such as failure to detect gas leaks, poor air quality, or water presence.

There exists a **technical and practical gap** in designing a low-cost, real-time, multi-sensor system that can autonomously handle:

- Intrusions
- Fire or gas leaks
- Rain or moisture infiltration
- Light-based automation
- Access control without complex biometric systems

Our proposed solution addresses these gaps through **a robust Arduino-based architecture integrated with sensors and actuators**, capable of both passive monitoring and proactive response.

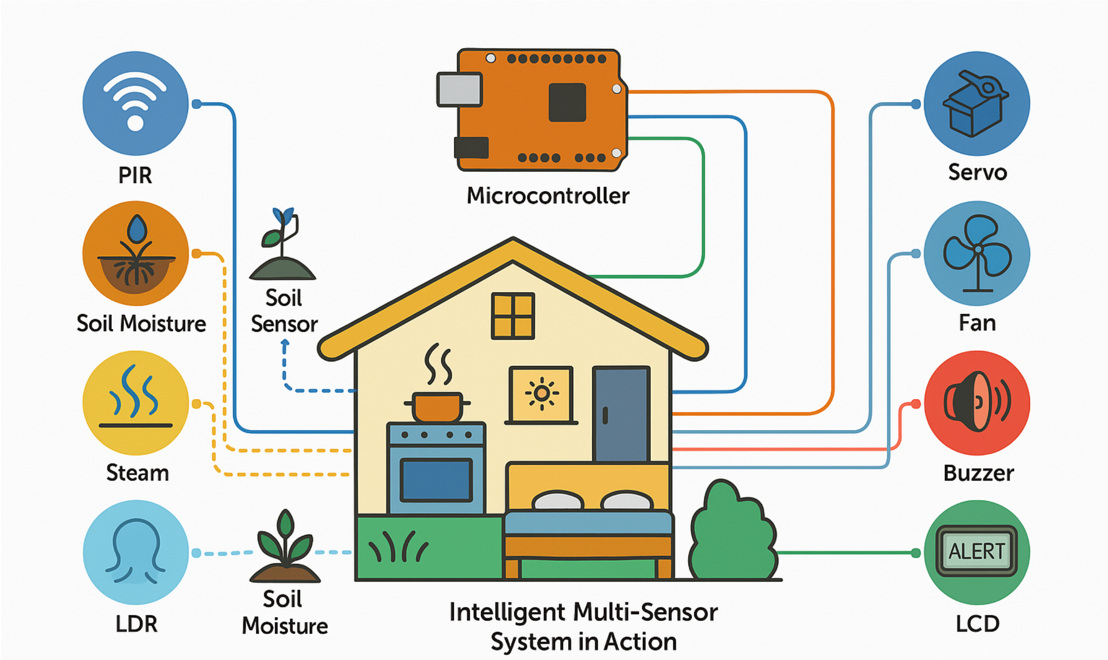


Figure 6 Intelligent Multi-Sensor System in Action

3.3 Challenges

To bring the above system to life, the development process encountered several categories of challenges:

3.3.1 Hardware Challenges

Challenge	Description
Multi-Sensor Integration	Ensuring seamless data collection from sensors with varied analog/digital outputs.
Power Supply Design	Delivering consistent 5V/12V regulated power to sensors and actuators.
Physical Placement and Housing	Positioning sensors (e.g., gas, PIR) and actuators (e.g., servos) for real-world conditions.
Waterproofing & Insulation	Ensuring steam/moisture sensors are safe and reliable under variable conditions.
Servo Load Management	Calibrating motors for smooth, realistic door/window operation without overheating.

Table 3 Hardware Challenges

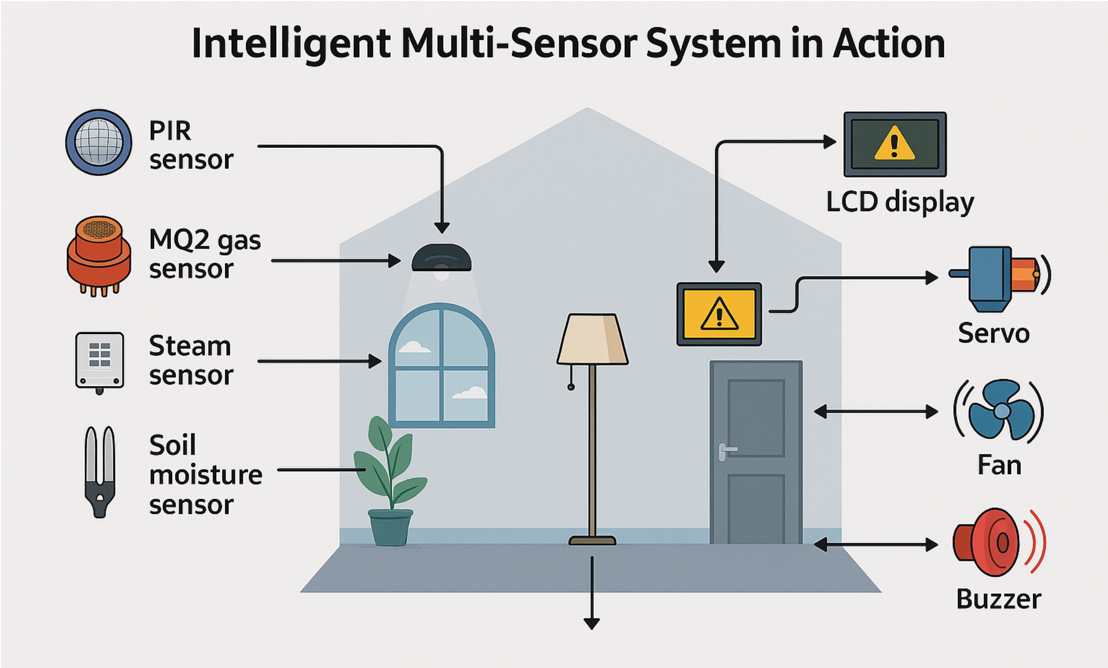


Figure 7 Demonstration of Muti-Sensor system

3.3.2 Software Challenges

Challenge	Description
Real-Time Decision Logic	Managing multiple sensor inputs with conditional branching for accurate actuation.
Morse-Code Password Input	Interpreting short/long button presses in Arduino for a secure and intuitive access system.
LCD Feedback & Interaction	Displaying real-time information while handling interrupts and actions.
Actuator Smoothness	Ensuring servo motor transitions are human-friendly and not abrupt.
Modular Code Design	Structuring Arduino code for flexibility, reuse, and future scalability.

Table 4 Software Challenges

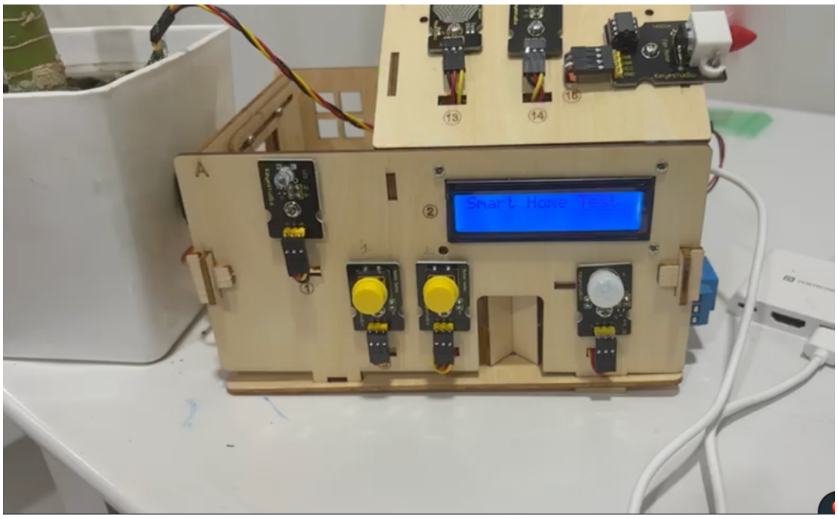


Figure 8 Lcd showing challenges in implementation of the code

3.4 Technical Scope

- **Sensor Modules Used:**
    - PIR, MQ2, LDR, Soil, Steam
- **Actuators:**
    - Servo motors, LEDs, Fan (via Relay), Buzzer
- **Microcontroller:**
    - Arduino Uno
- **User Feedback:**
    - 16x2 I2C LCD Display
- **Access System:**
    - Morse code-based password with button input
- **Power Supply:**
    - USB + External 5V regulated adapter

3.5 Significance

A smart home security system must not only notify, but **act autonomously**. By enabling multiple independent sensors to collectively make decisions and trigger appropriate actions (like opening/closing vents, turning on fans, or sending visual/audio alerts), this project becomes a prototype for the **next generation of fully automated smart homes** — practical, affordable, and customizable for real-world usage.

Chapter 4: Proposed Methodology

4.1 Proposed Work

The **Intelligent Multi-Sensor Based Home Security System with Automated Threat Response** is a microcontroller-based system designed for real-time monitoring of environmental conditions such as gas leaks, light levels, moisture, motion, and rain detection. It also responds automatically through actuators such as fans, servo-controlled doors/windows, LEDs, buzzers, and relays.

The system is built using an **Arduino Uno** platform and consists of the following major components:

- **Input Units:** PIR motion sensor, MQ2 gas sensor, LDR, soil moisture sensor, steam sensor, and two button modules for password input.
- **Output Units:** Servo motors (for door/window), white and yellow LEDs, buzzer, relay module (for ventilation), fan module, and an LCD1602 I2C display.
- **Power Supply:** A regulated 5V and 12V system powered via USB or external adapter (no use of 230V AC in the internal circuit).

The core logic resides on the Arduino, coded in C++ using the Arduino IDE. The system works on the **Sense** → **Process** → **Actuate** model and is completely autonomous post password verification.

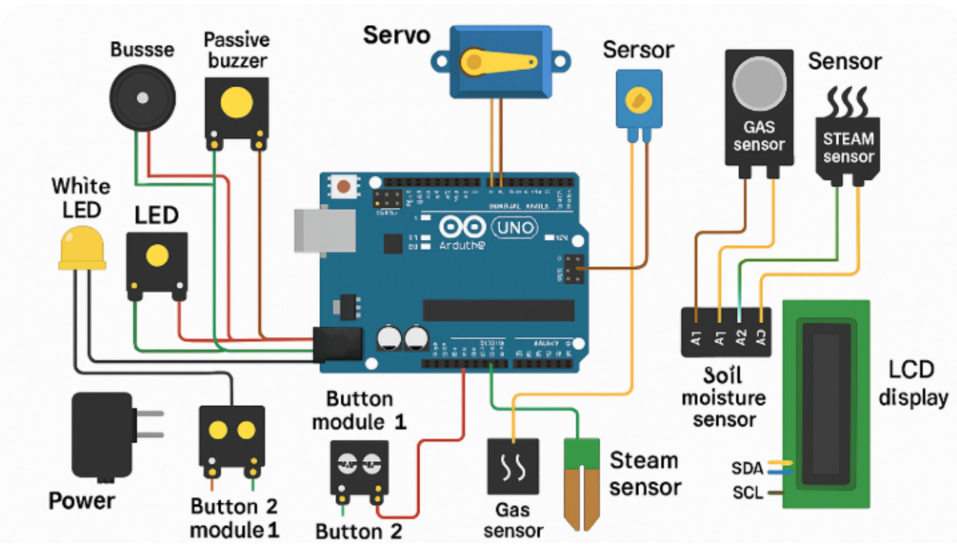


Figure 9 Component Wiring and Interface Map

4.2 System Architecture and Flow

- **Password Authentication (Morse Code Style):**
  The system starts in an idle state. Access is granted only if the correct Morse code sequence ("..") is entered using Button 1 and confirmed with Button 2.
- **Sensor Scanning (Every 500ms):**
  All sensors are polled regularly. Gas, steam, and motion readings trigger actuator responses. Soil and light levels are used for environmental feedback and smart lighting.
- **Actuator Response (Real-Time):**
  - **Door/Window Servos:** Respond to authentication and steam sensor values.
  - **Relay and Fan:** Triggered on high gas values.
  - **Buzzers & LEDs:** Provide alerts and visual cues.
  - **LCD Display:** Cycles through gas, light, motion, soil, door, and steam values every second.

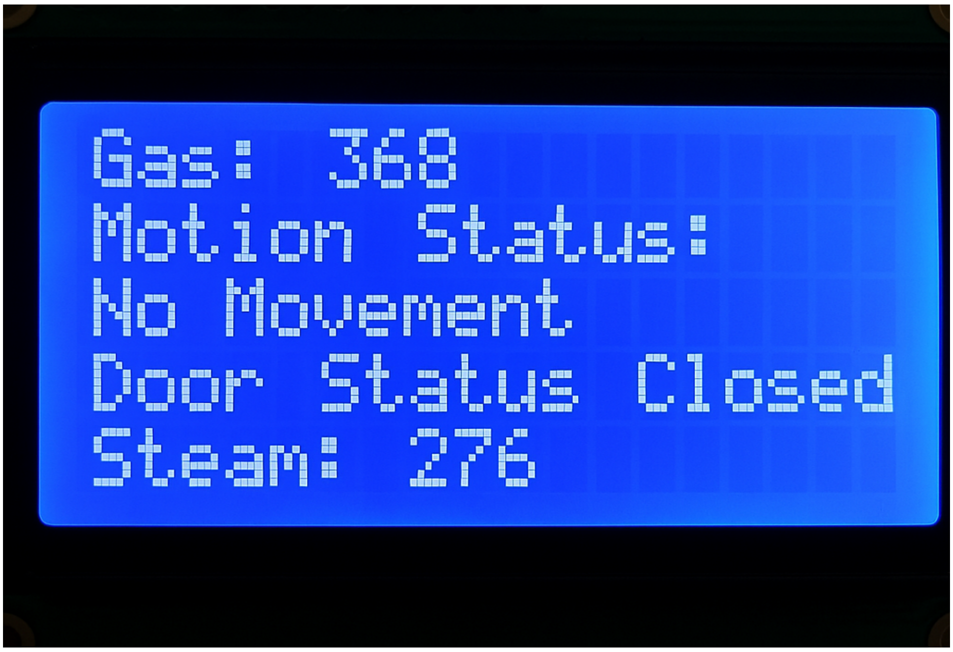


Figure 10 Real Time Sensor Monitoring on the LCD screen

4.3 Sensor-to-Pin Mapping

The table below defines the actual hardware mappings as implemented in your setup:

Component	Arduino Pin Mapping	Purpose
PIR Motion Sensor	Digital 2	Motion detection
Passive Buzzer	Digital 3	Audio alert
Button Module 1 (DOT)	Digital 4	Password input (.)
Relay Module	Digital 5	Ventilation control
Fan Control Pins	Digital 6, 7	Directional fan (IN1, IN2)
Button Module 2 (ENTER)	Digital 8	Password confirmation
Servo – Door	Digital 9	Physical access control
Servo – Window	Digital 10	Smart ventilation
Yellow LED	Digital 12	Gas alert indicator
White LED	Digital 13	Smart lighting
MQ2 Gas Sensor	Analog A0 (D11)	Gas detection
Photocell (LDR)	Analog A1	Light level sensing
Soil Moisture Sensor	Analog A2	Ground wetness
Steam Sensor	Analog A3	Rain/moisture detection
LCD1602 (I2C)	SDA/SCL (A4, A5)	Data display

Table 5 Sensor-to-pin Mapping

4.4 Applications

Area	Use Case
Home Automation	Smart windows, lighting control, and auto-locking doors.
Kitchen Safety	Gas detection with fan + relay triggering and buzzer alarms.
Balcony Protection	Steam-based auto-window closure during rain.

Area	Use Case
Soil Monitoring	Garden watering decisions or moisture leak detection.
Motion-Based Alert	Silent perimeter security with audible or visual alerts.

Table 6 Applications

4.5 Highlights of the Methodology

- Morse-code based secure access
- Non-blocking multitasking via millis()-based logic
- Synchronized motion via gradual servo transitions
- Multi-sensor integration for environmental intelligence
- LCD output for real-time user feedback

Chapter 5: Hardware Implementation

5.1 Introduction

The hardware implementation of our project, **“Design and Development of an Intelligent Multi-Sensor Based Home Security System with Automated Threat Response,”** involves integrating multiple environmental sensors and actuators onto a **custom-built cardboard house prototype**. This setup is powered entirely by **low-voltage regulated 5V and 12V DC supplies**, ensuring complete safety and portability.

Unlike generic smart security kits or camera-based systems, our hardware prototype delivers a **multi-sensor autonomous home defense** system. It physically **interacts with its environment** — opening doors/windows, activating alarms and fans, and adapting in real time based on sensor inputs.

The model includes a **live plant nearby** for soil moisture testing, **laser-cut openings for airflow and motion**, and clearly numbered sensor/actuator slots for demonstration clarity.

5.2 List of Components Used

S.No.	Component	Purpose
1	Arduino Uno R3	Central microcontroller that controls sensing, processing and actuation
2	Sensor Shield V5.2	Easy pin management and breakout for connecting multiple modules
3	LCD1602 (I2C) Display	Displays system states, alerts, and sensor data
4	Passive Buzzer	Audio alarm for alerts and feedback
5	PIR Motion Sensor	Detects human/animal movement
6	MQ-2 Gas Sensor	Detects LPG, methane, and smoke
7	LDR (Photocell)	Measures ambient light for smart lighting
8	Soil Moisture Sensor	Senses soil dryness for plant or water leakage detection
9	Steam Sensor	Detects high humidity or steam (rain detection)

S.No.	Component	Purpose
10	Yellow LED	Visual gas alert
11	White LED	Night light in low-light conditions
12	Relay Module	Controls external high-power devices (fan)
13	Fan Module	Cools or ventilates room when gas detected
14	Servo Motor (Window)	Automatically opens/closes window
15	Servo Motor (Door)	Motorized door access on correct password
16	Button Module 1	DOT button for Morse-code password
17	Button Module 2	ENTER button to submit password
18	Jumper Wires/Breadboard	Wiring and prototyping
19	5V and 12V DC Adapters	Safe low-voltage power supply (no 230V AC used)
20	Wooden-textured Cardboard Box	Represents house layout, sensor/door/window housing

Table 7 List of Component Used

5.3 Component-Level Breakdown

5.3.1 Arduino Uno + Sensor Shield

The heart of the system is an **ATmega328P-based Arduino Uno** running on 5V logic. A **sensor shield V5.2** expands the I/O availability and provides easy VCC/GND/SIGNAL breakouts. It allows safe and firm connections to 15+ components without soldering.

- All analog sensors are mapped to A0–A3.
- Digital pins D2–D13 are mapped to LEDs, buzzers, servos, and buttons.

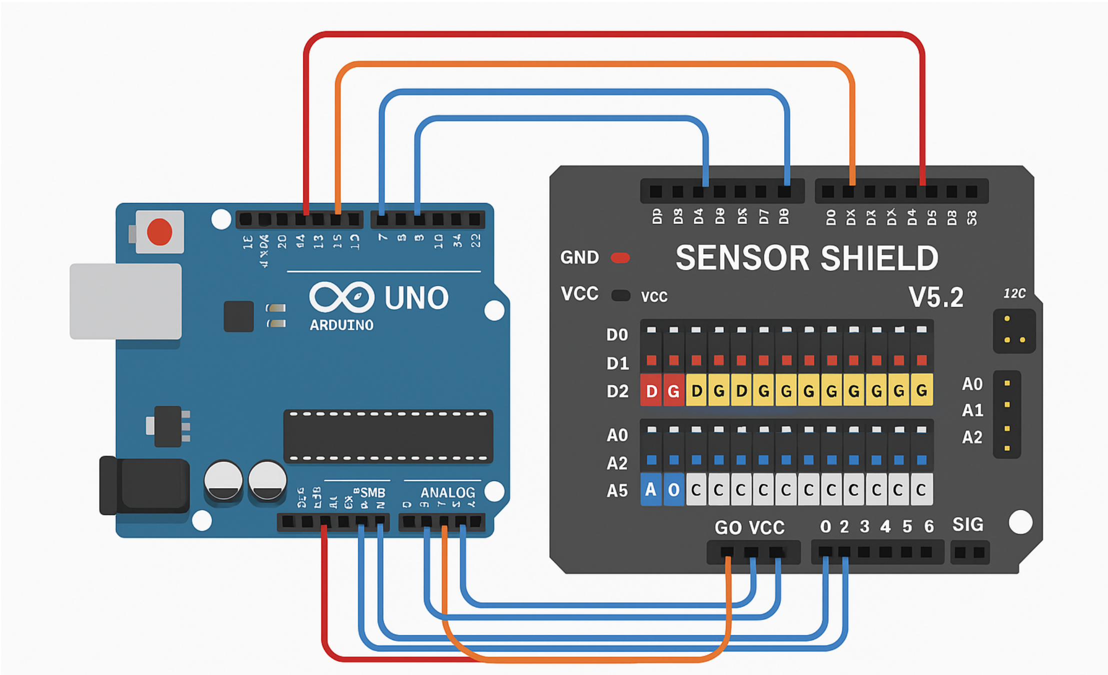


Figure 11 Sensor Shield V5.5

5.3.2 Sensor Network

All environmental sensors are powered using the onboard 5V supply:

- **PIR Sensor** (Pin 2): Detects motion → triggers buzzer and alert screen
- **MQ2 Gas Sensor** (A0): Triggers fan + yellow LED + relay
- **LDR (A1)**: Turns on white LED at night
- **Soil Sensor (A2)**: Reads real-time moisture of nearby plant
- **Steam Sensor (A3)**: Closes window automatically using servo

Thresholds are programmed in code for:

- Gas > 500
- Light < 300
- Soil < 700
- Steam > 800

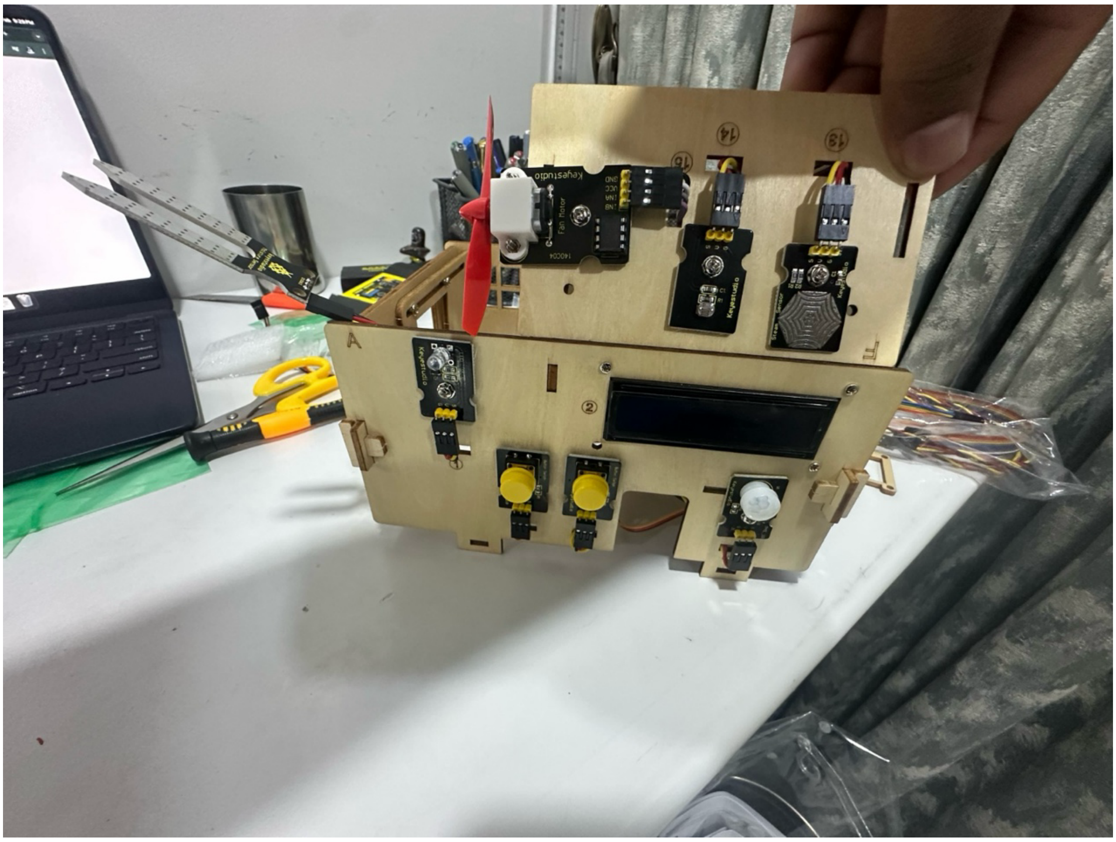


Figure 12 Sensor Positions and Threshold Roles

5.3.3 Actuators and Alerts

- **Servo Motors (D9, D10)**: Smooth motion to open/close windows and doors
- **Relay (D5)**: Activates 12V fan on gas detection
- **Fan (D6, D7)**: Ventilates room automatically
- **Yellow LED (D12)**: Visual alert when gas is present
- **White LED (D13)**: Automatic night light
- **Buzzer (D3)**: Audible alerts for gas and intrusion

All actuations are event-driven based on sensor thresholds in handleAlarms() and handleMotion() functions.

5.3.4 Password System: Morse-Code Authentication

Two push buttons (D4 and D8) are used:

- Button 1 records a DOT "." (if held short)
- Button 2 submits the password

Correct sequence ".." triggers door opening and system activation. Other hidden codes (e.g., "..." for birthday melody) enhance functionality.

This method:

- Minimizes hardware (no keypads needed)
- Enhances security
- Offers fun interaction

5.4 Physical Integration

The project is housed inside a **custom wooden-texture cardboard structure**, carefully laser-cut and assembled:

- Sensors and buttons are inserted into numbered slots.
- A real **potted plant is kept beside** the model to test soil moisture.
- A servo-controlled door and window simulate physical home automation.
- All wiring is internally routed and hidden behind the board.

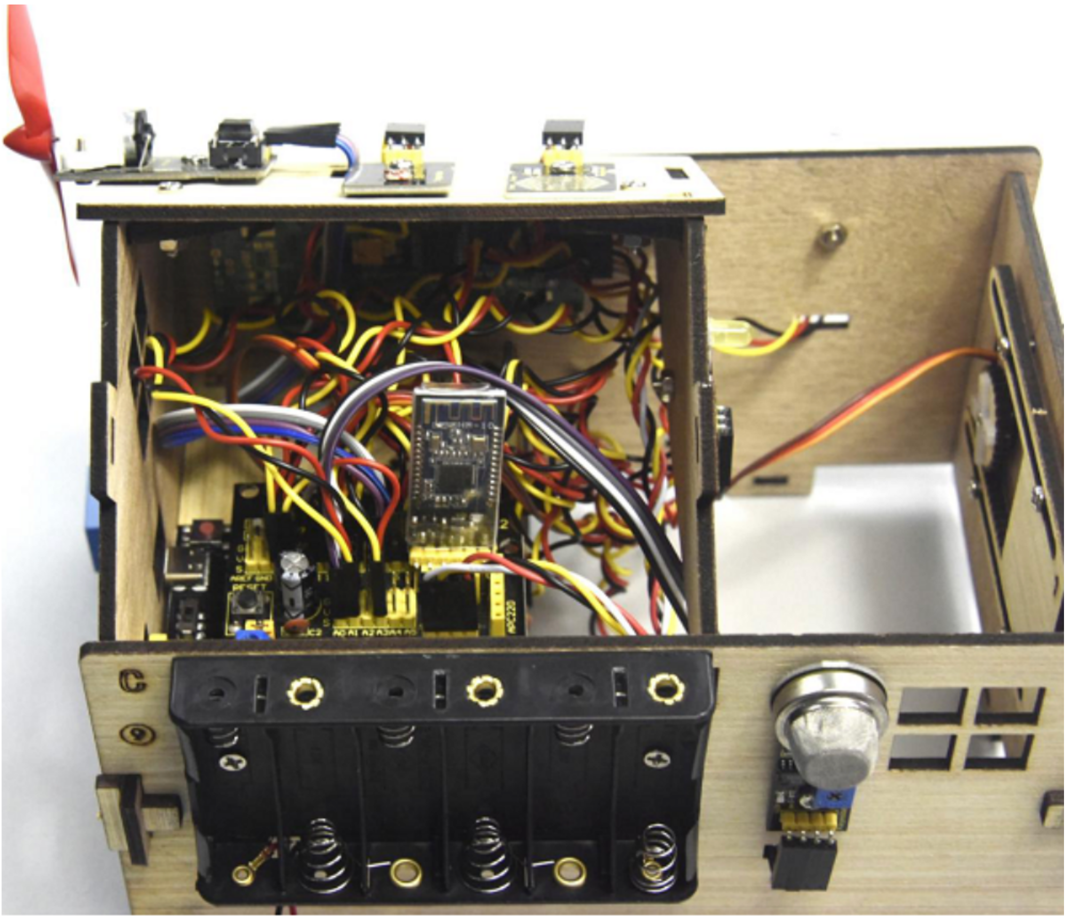


Figure 13 All Components assemble

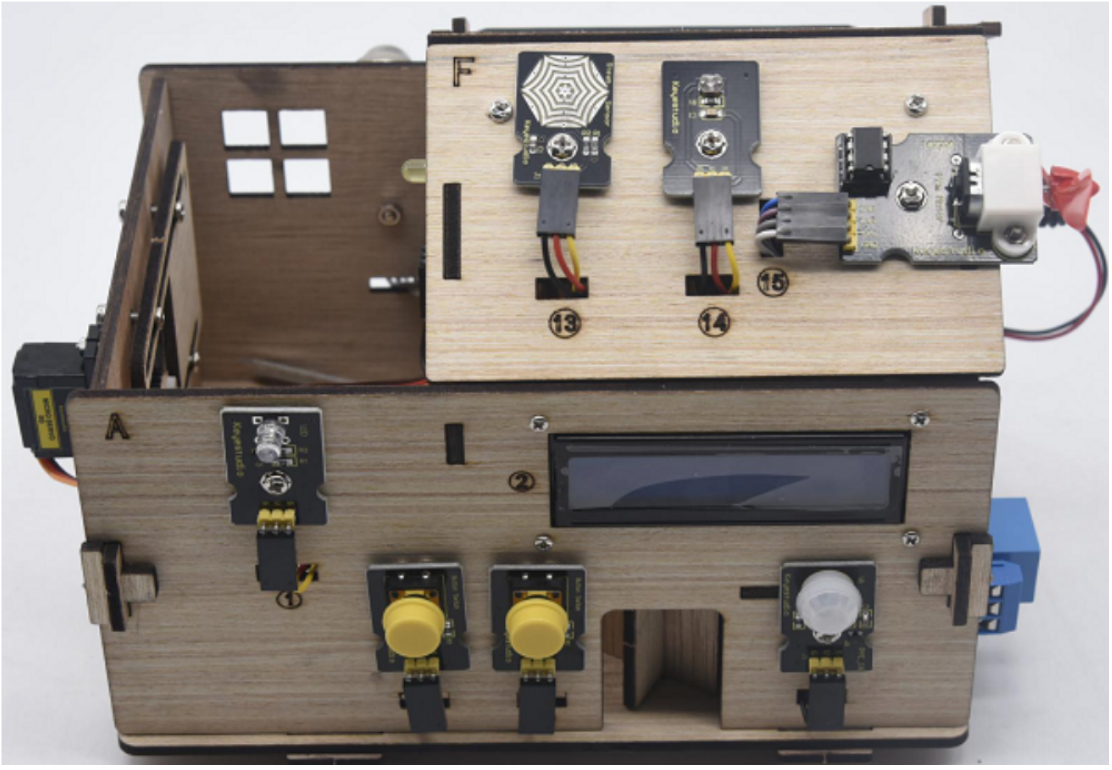


Figure 14 Final Hardware

5.5 Power Architecture

- **5V (Arduino + sensors + logic)**
- **12V (Fan + LED strip if needed)**

Power is sourced through:

- USB cable to Arduino
- External 12V adapter for fans or additional modules (through relay)

This makes the project:

- Safe for students and demo environments
- Efficient and portable

5.6 Connection Diagram

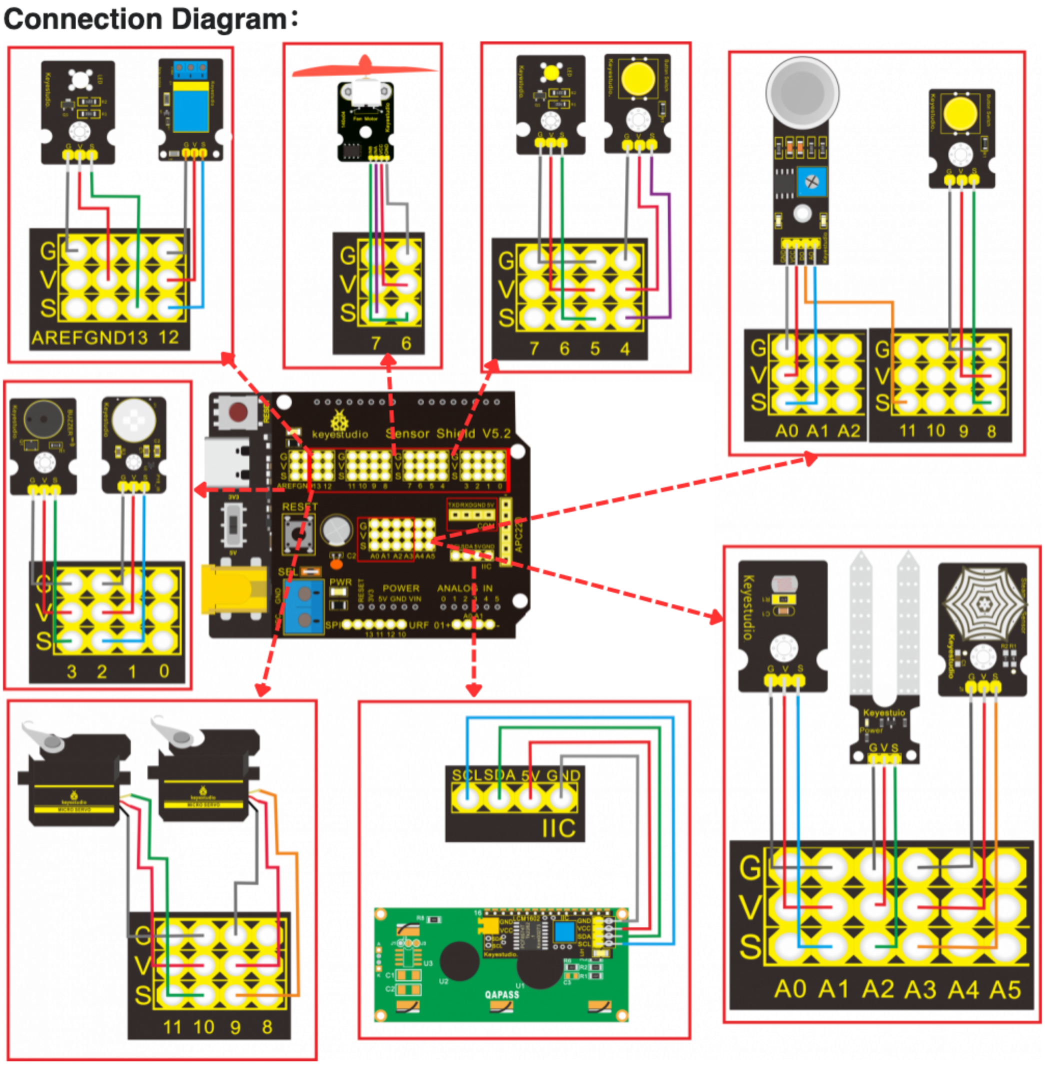


Figure 15 Connection Diagram

5.7 Summary of Implementation Features

- Morse-code authentication
- Context-aware automation
- Multi-sensor security with physical feedback
- Real-time LCD interface
- Fully enclosed smart house demo with live test plant
- Low-voltage operation (no AC hazard)

Chapter 6: Software Used

6.1 Arduino IDE

The development of the *Intelligent Multi-Sensor Based Home Security System with Automated Threat Response* was carried out using the **Arduino Integrated Development Environment (IDE)**. The Arduino IDE is a lightweight and powerful platform used for writing, compiling, and uploading embedded C/C++ code to microcontrollers.

In our project, it was used to:

- Interface with multiple analog and digital sensors.
- Program servo motors for automated window and door control.
- Display real-time system feedback on an LCD.
- Manage event-driven logic such as motion detection and gas leak alerts.
- Enable password-protected system activation using button-based Morse-code input.

The IDE's **Serial Monitor** also allowed observation of live data during testing phases to verify sensor responses and ensure timing precision in automation.

6.2 Libraries Utilized

To streamline interaction with various hardware modules, the following libraries were imported:

Library	**Purpose**
Servo.h	To control door and window servo motors.
Wire.h	Facilitates I2C communication with the LCD1602 display.
LiquidCrystal_I2C.h	Simplifies sending commands and messages to the 16x2 LCD display.

Table 8 Libraries Utilized

These libraries provided high-level abstractions for hardware interaction, making the code modular, readable, and easier to debug.

6.3 System Architecture and Functionality

The Arduino UNO, combined with a sensor shield, acts as the **central controller** of the system. Based on sensor readings and password authentication, it triggers various actuators and provides real-time feedback to the user.

Key Functional Modules:

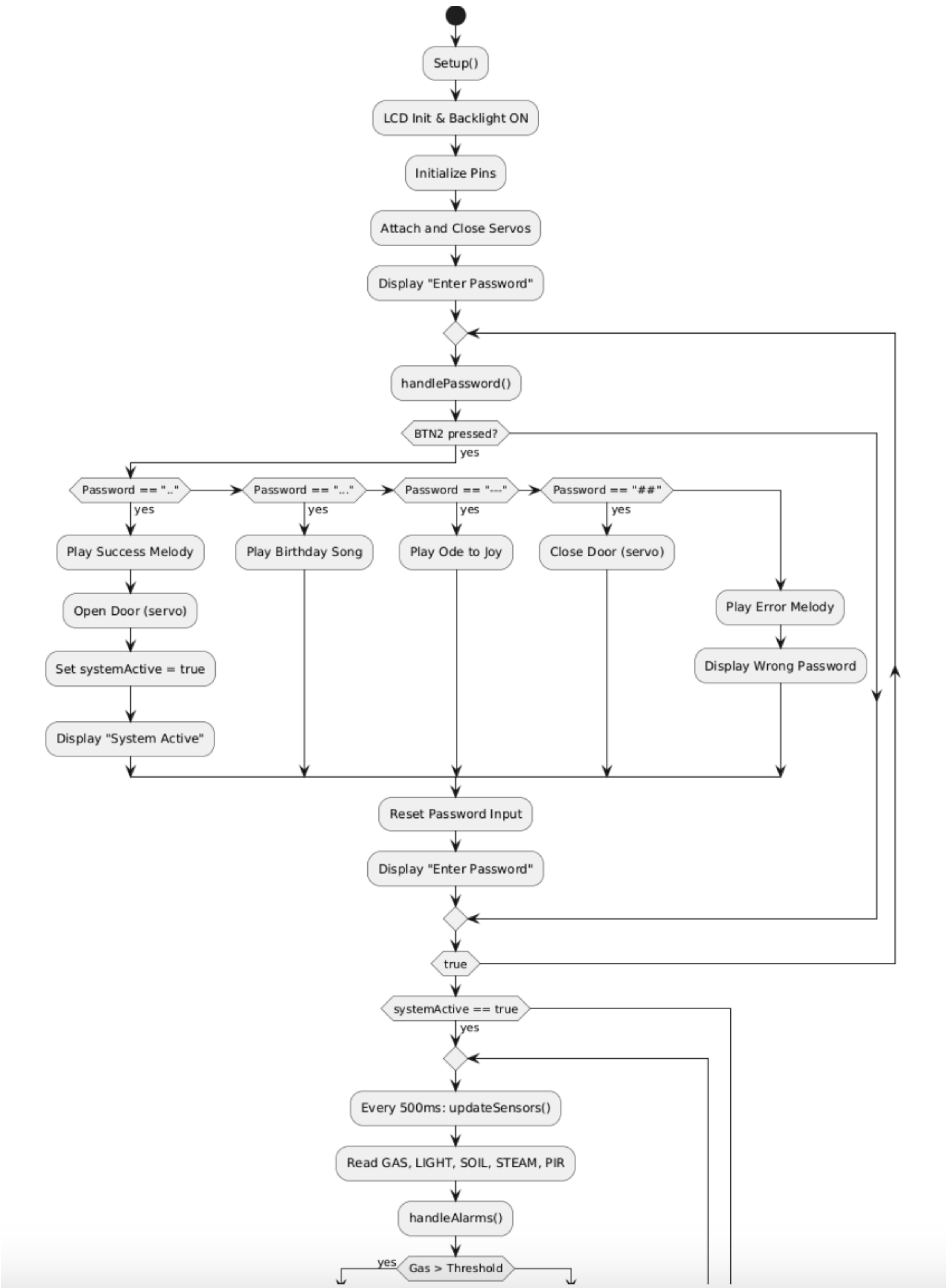
- **Password Verification System:**
  A Morse-style button interface accepts a coded password. Upon correct entry, the system activates and opens the door automatically.
- **Sensor Monitoring:**
  The system continuously monitors:
  - **Gas levels** (MQ2 sensor)
  - **Light intensity** (LDR)
  - **Soil moisture**
  - **Humidity/steam**
  - **Motion detection** (PIR sensor)
- **Actuator Response:**
  Based on real-time data:
  - A **fan and relay** are activated during gas detection.
  - The **window servo** closes on steam/rain detection and reopens when safe.
  - **LEDs and buzzer** provide visual and audible alerts.
  - **LCD display** continuously cycles through sensor values and system messages.
- **User Feedback:**
  Messages like "System Active", "Gas Detected!", "Need Water!", or "Rain Detected!" appear on the 16x2 LCD screen.

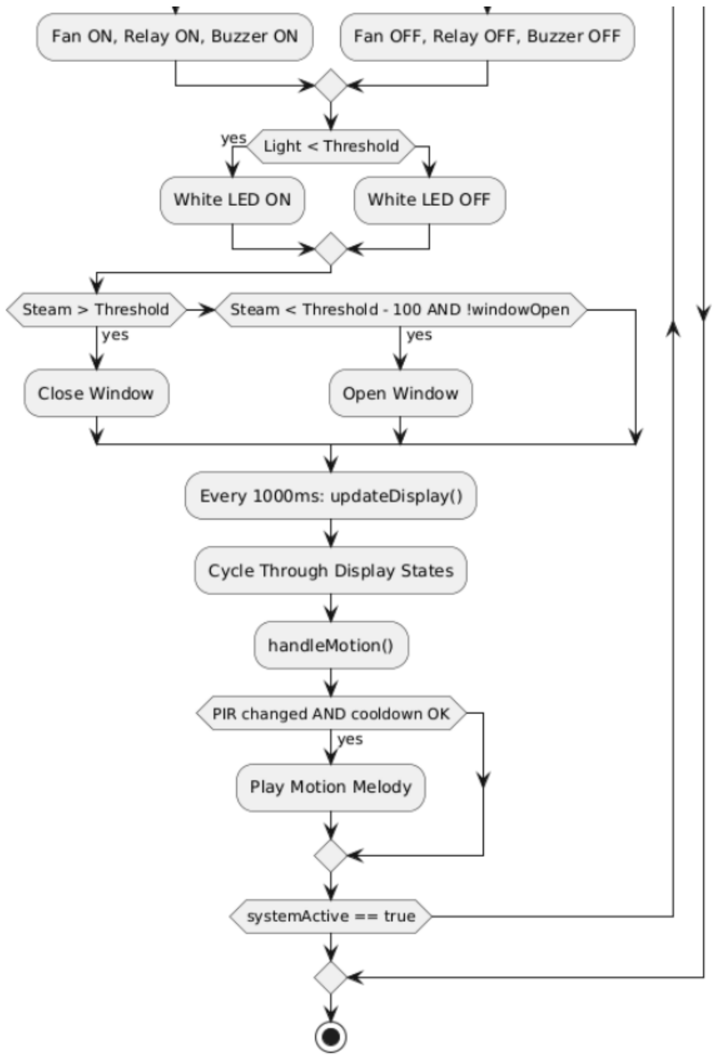
6.4 Circuit Integration

The code is written to align precisely with the hardware layout:

- **Digital pins (D2–D13)** are used for sensors, actuators, and buttons.
- **Analog pins (A0–A3)** handle the gas, soil, light, and steam sensors.
- **I2C lines (A4 - SDA, A5 - SCL)** manage LCD communication.
- Servo motors and fans are powered and controlled via shield-mounted headers for clean and modular wiring.

6.5 Code (Flowchart)




Fan ON, Relay ON, Buzzer ON
Fan OFF, Relay OFF, Buzzer OFF
Light < Threshold
yes
White LED ON
White LED OFF
Steam > Threshold
yes
Steam < Threshold - 100 AND !windowOpen
yes
Close Window
Open Window
Every 1000ms: updateDisplay()
Cycle Through Display States
handleMotion()
PIR changed AND cooldown OK
yes
Play Motion Melody
systemActive == true

6.6 Algorithm (Pseudo-code)

```
// Initialize components
Initialize LCD, Pins, Servos
Display "Enter Password" on LCD

// Main Loop
LOOP:
    handlePassword()
    IF systemActive THEN
      Every 500ms:
          updateSensors()
          handleAlarms()
      Every 1000ms:
         updateDisplay()
      handleMotion()

// Password Handling
FUNCTION handlePassword():
   IF Password button pressed THEN
      Append "." to currentPassword

   IF Enter button pressed THEN
      IF correct password ".." THEN
         Play success melody
         Smoothly open door
         activateSystem()

      ELSE
```

```
        Play error tone
        Display "Wrong Password" on LCD

// Sensor Update
FUNCTION updateSensors():
  Read gas, light, soil, steam sensors
  Read PIR motion sensor

// Alarm Handling
FUNCTION handleAlarms():
  IF gas level > threshold THEN
    Turn ON fan, relay, buzzer
  ELSE
    Turn OFF fan, relay, buzzer

IF light < threshold THEN
    Turn ON white LED
  ELSE
    Turn OFF white LED

  IF steam > threshold THEN
    Close window
  ELSE IF steam < threshold - 100 THEN
    Open window

// Display Update
FUNCTION updateDisplay():
  IF no alarm active THEN
    Show sensor data and system status
```

```
    ELSE
      Show "ALERT: GAS/FIRE!" on LCD

// Motion Detection
FUNCTION handleMotion():
    IF new motion detected and 3 sec passed THEN
      Play motion alert melody

// System Activation and Deactivation
FUNCTION activateSystem():
    Set systemActive = true
    Open door

FUNCTION deactivateSystem():
    Set systemActive = false
    Close door and window
    Turn off outputs
```

6.7 Code Explanation

The Arduino program forms the core intelligence of the *Intelligent Multi-Sensor Based Home Security System with Automated Threat Response*. It is written in C++ using the Arduino framework and integrates multiple sensors, actuators, and an interactive LCD interface to deliver a real-time, reactive home security solution. Below is a breakdown of its key components:

1. Libraries and Initialization

The code begins by importing essential libraries:

- Servo.h – for controlling servo motors.
- Wire.h and LiquidCrystal_I2C.h – to handle I2C communication with the 16x2 LCD display.

Constants and threshold values are defined for sensors (e.g., gas, light, soil moisture, steam), actuators (e.g., fan, LEDs, relays), and system control timings (e.g., dot duration for Morse code password).

2. Global Variables

Flags and timing variables manage the system state (systemActive, doorOpen, windowOpen) and debounce logic. The SensorData structure stores real-time sensor readings, simplifying logic handling.

3. Setup Function

The setup() function performs hardware initialization:

- Configures pin modes.
- Attaches servo motors to pins and sets them to default positions.
- Initializes the I2C LCD and displays a password prompt.

4. Main Loop

The loop() function is the main program cycle:

- Calls handlePassword() to manage secure system access via Morse-code button input.
- If access is granted (systemActive is true), it:
    - Periodically reads sensor values (updateSensors()),
    - Evaluates them (handleAlarms()),
    - Displays outputs (updateDisplay()),
    - Responds to motion detection (handleMotion()).

5. Password System (Morse Code Based)

The password is entered using a single button where:

- Short press = dot (.)
- Long press = ignored (dashes removed for simplicity) An additional button confirms the password. Custom passwords trigger various responses: system activation, birthday music, or door closure.

6. Sensor Monitoring

The updateSensors() function reads values from:

- **MQ2 Gas Sensor** – detects smoke/LPG.
- **LDR (Light Sensor)** – determines ambient light.

- **Soil Moisture Sensor** – detects soil wetness.
- **Steam Sensor** – detects humidity or rain.
- **PIR Motion Sensor** – detects human presence.

7. Intelligent Response and Automation

The handleAlarms() function:

- Activates fans, buzzers, and relays if gas levels are high.
- Automatically adjusts window position based on steam levels.
- Controls indoor lighting based on ambient light.

8. LCD Status Display

The updateDisplay() function displays sensor readings and system alerts cyclically. In case of emergencies, it overrides normal output to display warnings (e.g., "ALERT: GAS/FIRE").

9. Motion Detection Alert

The handleMotion() function continuously checks for motion changes and plays a buzzer melody upon detection, offering real-time feedback and deterring intruders.

10. Music Integration

Special passwords trigger melody playback (e.g., “Birthday Song” or “Ode to Joy”), enhancing user experience and demonstrating custom signal handling.

Chapter 7: Results and Conclusion

7.1 Results

The *Intelligent Multi-Sensor Home Security System with Automated Response* was successfully developed, integrated, and tested. The system is capable of monitoring key environmental and safety parameters in real time, and it responds intelligently based on defined thresholds and user inputs.

Key results from the implementation include:

- **Secure System Activation:**
  The system remains deactivated until a valid Morse-code-based password (..) is entered using button modules. Upon successful authentication, the door opens automatically, and the monitoring system is activated.
- **Gas Detection and Response:**
  When gas levels exceed the predefined safety threshold, the system:
  - Activates the **buzzer** for audio alert,
  - Turns ON the **yellow LED** as a visual warning,
  - Powers ON the **fan** and **relay module** to assist with ventilation.
- **Motion Detection and Alert:**
  The **PIR sensor** continuously monitors for motion. Upon detection, an audio alert melody is triggered, simulating a real-time intrusion alarm.
- **Steam Detection and Automated Window Control:**
  Based on the steam sensor's input:
  - The **window servo motor** automatically closes the window when high moisture is detected (e.g., due to rain).
  - When normal conditions resume, the window reopens automatically.
- **Soil Moisture Monitoring:**
  A real plant is placed near the system to demonstrate the function of the **soil humidity sensor**. The system:
  - Displays "Need Water!" if the soil is dry.
  - Displays "Enough Water!" when moisture is adequate.
- **Ambient Light Control:**
  The **photocell sensor** determines surrounding light levels. If it falls below a certain threshold, the **white LED** is turned ON for illumination.
- **Real-Time LCD Feedback:**
  The **16x2 I2C LCD module** scrolls through sensor readings and system states every second, including gas, light, soil, steam, motion, and door status.

This system was mounted on a **cardboard-based prototype house**, successfully integrating hardware and software to simulate a real-world smart home security model. Every component was individually tested and validated under practical conditions.

7.2 Conclusion

This project demonstrates the practical realization of a **low-cost, efficient, and modular home security system** using Arduino and off-the-shelf sensors and actuators. The implementation provides a proof-of-concept for:

- **Automation and Control:** Intelligent control of fan, lights, window, and door based on environmental input.
- **Security Monitoring:** Real-time alerts for motion and gas detection enhance safety.
- **User Interaction:** A simple and secure button-based password system provides access control.
- **Environmental Feedback:** LCD display and visual/auditory indicators provide live feedback to the user.

The system is scalable and designed in a modular fashion. It can be extended in the future with features such as:

- Wireless communication (e.g., GSM, WiFi modules),
- Mobile application control,
- Cloud-based data logging,
- Solar-powered operation for remote areas.

By integrating real-time sensing, feedback, and automation, this project lays a strong foundation for advanced smart home applications. It validates how embedded systems can bring real-world value through intelligent design and functional integration.