Artificial Intelligence

Home Automation with Voice

January 02/01/2025

Obaidullah | 221109
Moiz | 221073
Ali Afzal | 220827

Submitted to

Maam Fariyal Farooq

Contents

1. ABSTRACT.. 2

2. INTRODUCTION 2

3.OBJECTIVES 2

4.Agent-Based Architecture.. 3

5. Natural Language Processing (NLP) 3

6.Speech Recognition (ASR) 4

7. Large Language Models (LLMs)... 4

8. Function Calling / Tool Use.. 5

9. Intent Recognition & Classification.. 5

10. Reactive Agents 6

11. Perception–Action Loop 6

12. Context-Aware Computing.. 6

13. Multi-Modal AI... 7

14 AI-Centric Summary... 7

14.COMPONENTS USED 8

UI Design :.. 9

15.CIRCUIT DIAGRAM... 9

16.SYSTEM WORKING (MODULE-WISE EXPLANATION) 10

Module 1: LDR and Night LED.. 10

Module 2: Garage Door Automation 11

Module 3: Motion Detection and Security Alert 11

Module 4: Rain Water Detection and Clothes Protection.. 12

Module 5: Soil Moisture and Water Tank Monitoring.. 13

Module 6: Fire and Smoke Detection 15

Module 7: Temperature and Humidity Monitoring 17

17.ARDUINO CODE.. 18

Components:.. 24

Protocols :.. 27

18.CONCLUSION.. 30

1. ABSTRACT

This IoT-based project, Smart Home and Agriculture System, automates and monitors both household and agricultural environments using an Arduino Uno R3 microcontroller. It integrates a wide range of sensors and actuators to handle smart lighting, garage door automation, intruder detection, automatic clothes covering during rain, soil-based irrigation control, fire and smoke alarms, and environmental monitoring (temperature & humidity). The system improves convenience, safety, and environmental sustainability through real-time automation.

2. INTRODUCTION

In recent years, the integration of Internet of Things (IoT) technologies into daily life has transformed traditional systems into smart, automated, and efficient solutions. Smart homes offer convenience, security, and energy efficiency by automating lighting, doors, and monitoring systems. Similarly, modern agriculture benefits from automated irrigation, environmental monitoring, and early warning systems.

This project — Smart Home and Agriculture System — merges both domains using an Arduino Uno and a collection of sensors and actuators. The system responds to environmental data (light, moisture, temperature, rain, motion, etc.) in real-time to trigger appropriate actions. These include turning lights on at night, covering clothes when it rains, opening a garage door, detecting intrusions, watering dry soil, and alerting in case of fire or smoke.

The project is low-cost, modular, and expandable, making it suitable for both home automation and smart farming applications. It demonstrates the power of embedded systems, sensor networks, and real-world automation using Arduino and simple electronics.

3.OBJECTIVES

- Automate daily home and farm tasks
- Improve resource usage (electricity, water)
- Enhance safety using sensors for fire, gas, and intruders
- Provide hands-free environmental monitoring
- Demonstrate practical IoT integration using Arduino

4.Agent-Based Architecture

In Artificial Intelligence, an agent is an autonomous computational entity that continuously perceives its environment through sensors, reasons using internal models, and acts using actuators to achieve predefined or learned goals.

AI Role in This System

The IoT Smart Home operates as a Hybrid Intelligent Agent System, integrating:

Reactive AI (rule-based automation)

Cognitive AI (LLM-based reasoning)

This hybrid design ensures both real-time responsiveness and high-level intelligence.

AI Mapping

AI Stage	**Implementation**
Perception	Sensors (temperature, motion, light, gas)
Reasoning	LLM + rule engine
Action	Smart devices (lights, AC, alarms)

Agent Properties (AI-Aligned)

Property	**AI Interpretation**
Autonomy	Decisions without human intervention
Reactivity	Immediate response to sensor events
Proactiveness	Predictive and preventive actions
Social Ability	Human interaction via natural language

AISignificance:

This architecture aligns with Russell & Norvig’s Intelligent Agent Model.

5. Natural Language Processing (NLP)

NLP enables machines to understand, interpret, and reason over human language using statistical and neural models.

NLP Functions in the System

- Tokenization & semantic parsing
- Intent detection
- Entity recognition (device, location, action)

AI Example

User Input:

“Turn on the living room lights”

AI Output:

```
{
  "intent": "CONTROL_DEVICE",
  "device": "light",
  "location": "living_room",
  "action": "on"
}
```

AI Advantage

Handles synonyms and paraphrasing

Reduces dependency on rigid command structures

Supports conversational interaction

6.Speech Recognition (ASR)

ASR is a deep learning-based supervised learning task that converts acoustic signals into textual representations.

Model Used

OpenAI Whisper (**Transformer-based ASR**)

AI Pipeline:

Audio Signal → Spectrogram → Neural Network → Text Output

AI Importance

- Enables hands-free control
- Improves accessibility
- Bridges human speech with machine cognition

7. Large Language Models (LLMs)

LLMs are transformer-based foundation models trained on large-scale textual data to perform reasoning, planning, and language understanding.

AI Role in the System

- Acts as the cognitive brain
- Resolves ambiguous commands
- Maintains conversational context
- Translates human intent into machine actions

Why LLMs Are AI-Critical

Capability	AI Benefit
Reasoning	Goal-oriented decisions
Context	Memory across interactions
Generalization	Handles unseen instructions
Adaptability	Learns user preferences

8. Function Calling / Tool Use

Function calling enables tool-augmented intelligence, where AI outputs are converted into executable system commands.

AI Flow

Natural Language → LLM Reasoning → Structured JSON → Backend Function → Actuator

Example

```
{
  "function": "switch_device",
  "parameters": {
    "device": "AC",
    "state": "ON"
  }
}
```

AI Benefit

Connects abstract reasoning to physical actions

Ensures safe and deterministic execution

9. Intent Recognition & Classification

Intent recognition is a multi-class classification problem in NLP that maps user input to system goals.

Intent Categories

- Control Intent (device operation)
- Informational Intent (status queries)
- Emergency Intent (fire, gas leakage)

AI Strength

- Supports compound and nested intents
- Improves scalability of interaction design

10. Reactive Agents

Reactive agents operate using if–then rules, responding directly to stimuli without long-term planning.

Role in Hybrid AI

- Fire alarm activation
- Motion-based lighting
- Gas leakage alerts
- AI Justification

Reactive AI is:

- Fast
- Reliable
- Safety-critical
- Used where response time > reasoning complexity.

11. Perception–Action Loop

A fundamental AI loop connecting sensory input to intelligent action.

Loop

Perceive → Interpret → Decide → Act → Feedback

AI Benefit

- Continuous system operation
- Enables future learning integration
- Aligns with reinforcement learning frameworks

12. Context-Aware Computing

Context-aware AI adapts decisions using environmental, temporal, and user-specific data.

Context Dimensions

- Time (day/night)
- Location
- User habits
- Environmental state

AI Example

“Make it comfortable”

AI infers:

Temperature

Humidity

Timeof day

→ Adjusts AC and fan automatically

---

13. Multi-Modal AI

Multi-modal AI combines multiple data types into a unified reasoning framework.

Modalities Used

- Audio (speech)
- Text (commands)
- Sensor data (IoT telemetry)

AI Advantage

- Improved accuracy
- Better situational awareness
- Human-like perception

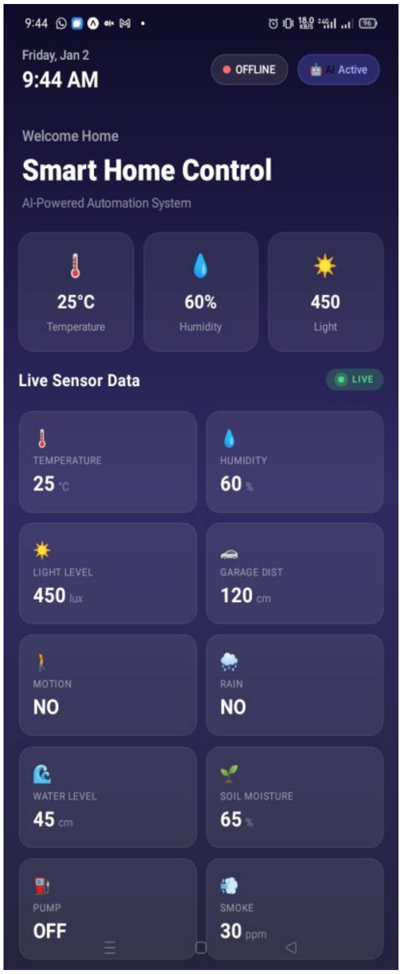
14 AI-Centric Summary

AI Domain	Application
Intelligent Agents	Autonomous control
NLP	Human-friendly commands
ASR	Voice interaction
LLMs	Cognitive intelligence
Tool Use	Action execution
Reactive AI	Safety automation
Context Awareness	Adaptive behavior
Multi-Modal AI	Robust perception

14.COMPONENTS USED

No.	Component	Quantity	Function
1	Arduino Uno R3	1	Controls all components
2	HC-SR04 Ultrasonic Sensor	2	Garage door & water level detection
3	PIR Motion Sensor	1	Detects motion/intruders
4	Rain Sensor Module	1	Detects rainwater
5	LDR Sensor	1	Detects day/night lighting
6	IR Flame Sensor	1	Detects fire/flame
7	MQ2 Smoke Sensor	1	Detects smoke or gas leakage
8	Soil Moisture Sensor	1	Monitors soil dryness
9	SG90 Servo Motors	2	For garage door and clothes cover
10	IRFZ44N MOSFET	1	Controls pump switching
11	Water Pump	1	Waters soil when dry
12	DHT22 Sensor	1	Temperature & humidity monitoring
13	Buzzer	1	Alerts for motion/fire/smoke
14	LEDs	2	Use for Lighting and Indication
15	10kΩ Resistors & Jumper Wires		For voltage division & connections

UI Design :



9:44

Friday, Jan 2

9:44 AM

● OFFLINE

🤖 Active

Welcome Home

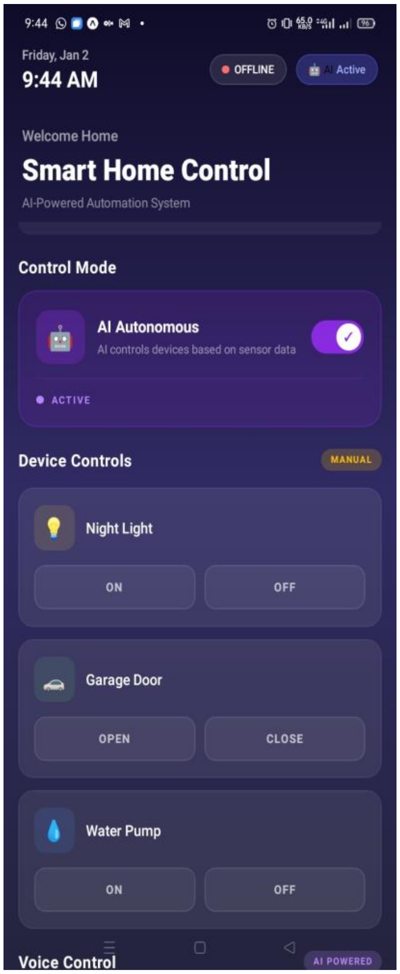
Smart Home Control

AI-Powered Automation System

25°C	60%	450
Temperature	Humidity	Light

Live Sensor Data ● LIVE

TEMPERATURE 25 °C	HUMIDITY 60 %
LIGHT LEVEL 450 lux	GARAGE DIST 120 cm
MOTION NO	RAIN NO
WATER LEVEL 45 cm	SOIL MOISTURE 65 %
PUMP OFF	SMOKE 30 ppm



9:44

Friday, Jan 2

9:44 AM

● OFFLINE

🤖 Active

Welcome Home

Smart Home Control

AI-Powered Automation System

Control Mode

AI Autonomous ✓

AI controls devices based on sensor data

● ACTIVE

Device Controls MANUAL

Night Light

ON OFF

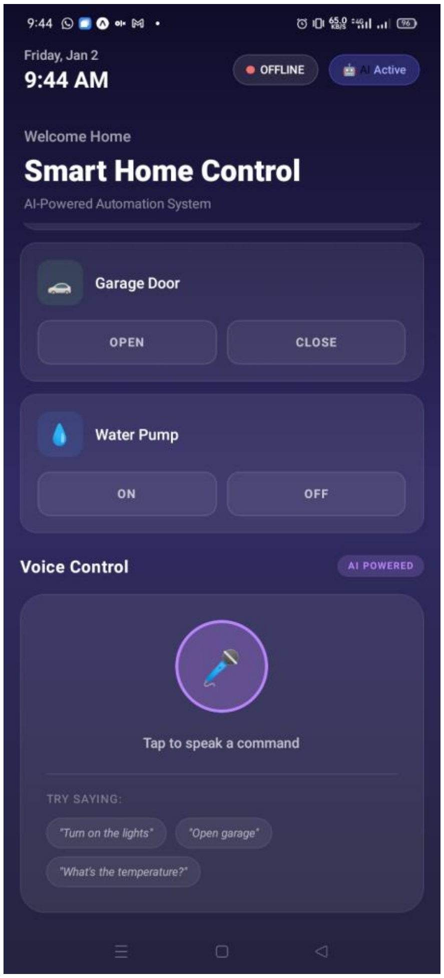
Garage Door

OPEN CLOSE

Water Pump

ON OFF

Voice Control AI POWERED



9:44

Friday, Jan 2

9:44 AM

● OFFLINE

🤖 Active

Welcome Home

Smart Home Control

AI-Powered Automation System

Garage Door

OPEN CLOSE

Water Pump

ON OFF

Voice Control AI POWERED

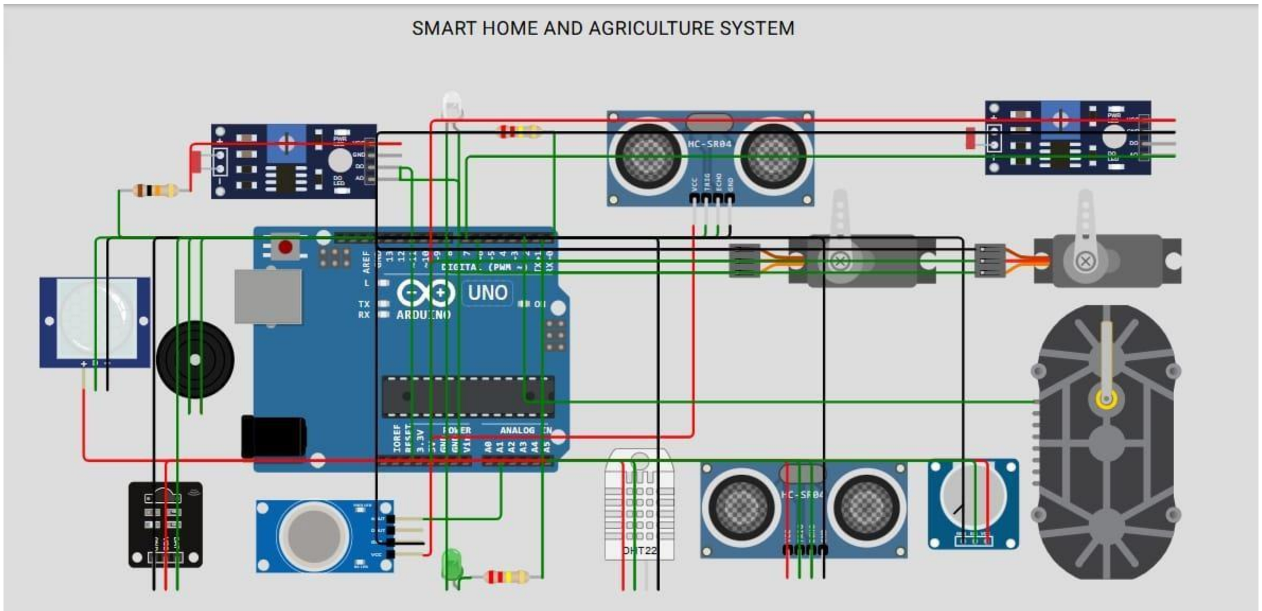
Tap to speak a command

TRY SAYING:

"Turn on the lights" "Open garage"

"What's the temperature?"

15.CIRCUIT DIAGRAM



16.SYSTEM WORKING (MODULE-WISE EXPLANATION)

Module 1: LDR and Night LED

Component: LDR Sensor

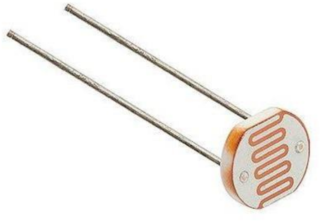


Figure 1 LDR Sensor

Figure 2.White LED

Working: This module uses a Light Dependent Resistor (LDR) to measure ambient light intensity. If the light level falls below a certain threshold (indicating darkness), an LED is automatically turned ON to illuminate the area. This simulates smart lighting that responds to time of day.

Module 2: Garage Door Automation

Components: Ultrasonic Sensor, SG90 Servo Motor

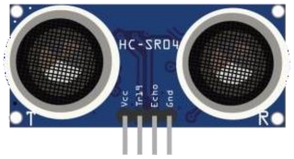


Figure 3. HcSr04 Ultrasonic Sensor

Figure 4. Servo Motor SG-90

Working: An ultrasonic sensor detects if a vehicle or person is within 5–15 cm of the garage. If detected, the Arduino triggers a servo motor to open the garage door. After a short delay, the door closes automatically, providing a contactless entry system.

Module 3: Motion Detection and Security Alert

Components: PIR Motion Sensor, Buzzer

Figure 5. PIR Motion

Figure 6. Buzzer

Working: The PIR sensor monitors movement. When motion is detected (such as an intruder), the Arduino activates a buzzer to alert occupants, enhancing home security.

Module 4: Rain Water Detection and Clothes Protection

Components: Rain Sensor, Servo Motor

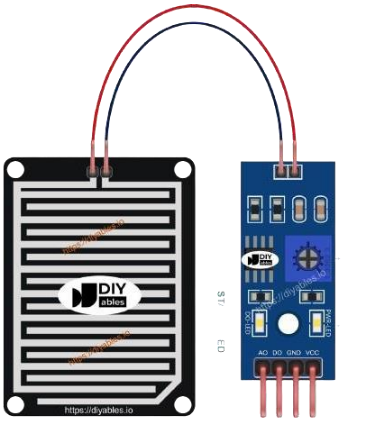


Figure 7. Rain Sensor

Figure 8. Servo Motor SG-90

Working: This module protects clothes drying outdoors. When the rain sensor detects water droplets, a servo motor pulls a cover over the clothes. When no rain is detected, the servo returns to its default position.

Module 5: Soil Moisture and Water Tank Monitoring

Components: Soil Moisture Sensor, Ultrasonic Sensor, Water Pump, IRFZ44N MOSFET

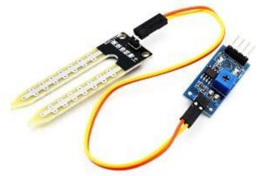


Figure 9. Soil Moisture Sensor

Figure 10. HC-SR04 Ultrasonic Sensor

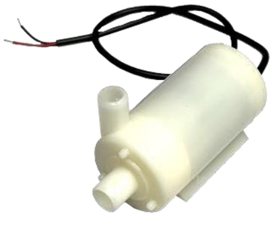


Figure 11. Water Motor

Figure 12. IRFZ44N MOSFET

Working: The soil sensor checks soil wetness while the ultrasonic sensor monitors water tank level. If the soil is dry and water is available, a pump is turned ON using a MOSFET switch to irrigate the field automatically.

Module 6: Fire and Smoke Detection

Components: IR Flame Sensor, MQ2 Gas Sensor, LED and Buzzer

Figure 13. Flame Sensor

Figure 14. MQ2 Gas Sensor

Figure 15. Buzzer

Figure 16.LED

Working: This safety module activates when flame or smoke is detected. If either is sensed, the Arduino turns on an alert LED and sounds a buzzer, warning people in the area.

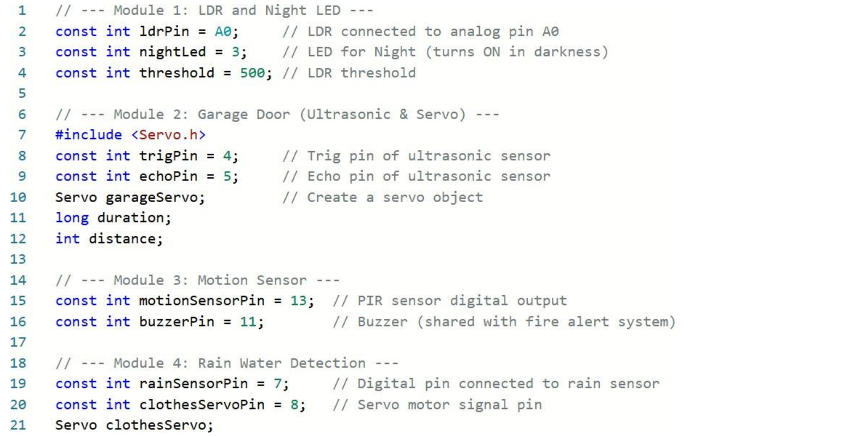
Module 7: Temperature and Humidity Monitoring

Component: DHT22 Sensor

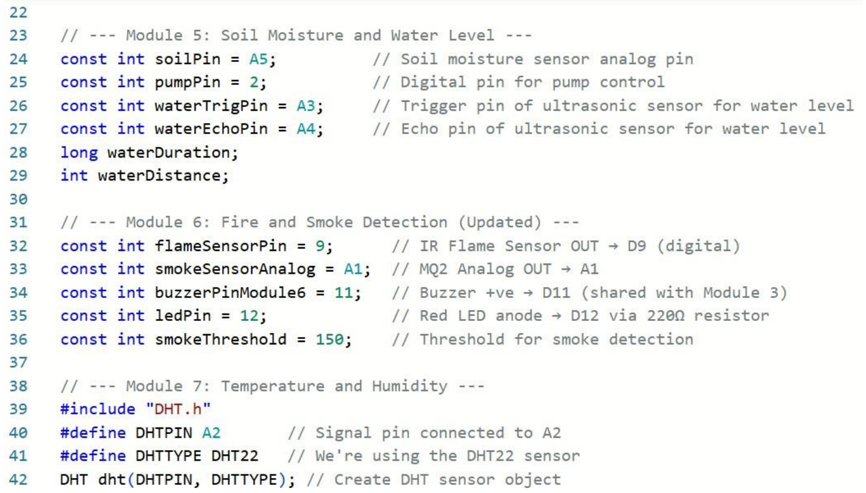
Figure 17. DHT22 Sensor

Working: This sensor continuously reads temperature and humidity levels and sends data to the Serial Monitor. It helps monitor environmental conditions and can be used for future automation like fans or greenhouses.

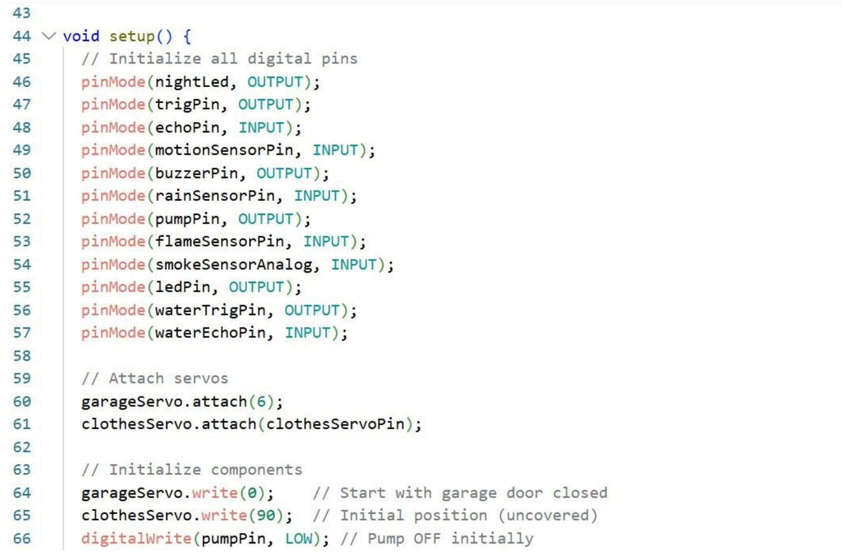
17.ARDUINO CODE



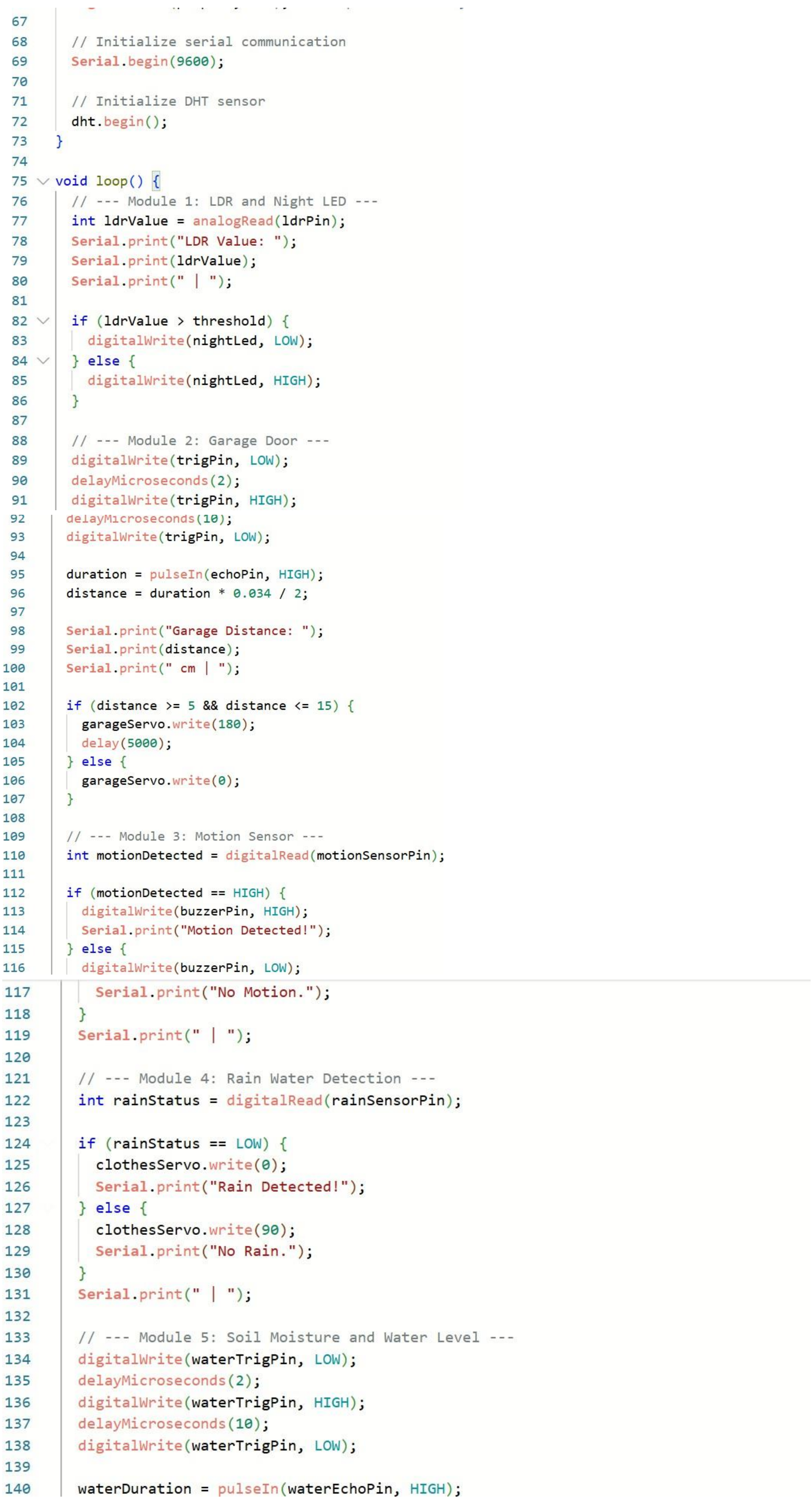
```
// --- Module 1: LDR and Night LED ---
const int ldrPin = A0;     // LDR connected to analog pin A0
const int nightLed = 3;    // LED for Night (turns ON in darkness)
const int threshold = 500; // LDR threshold

// --- Module 2: Garage Door (Ultrasonic & Servo) ---
#include <Servo.h>
const int trigPin = 4;     // Trig pin of ultrasonic sensor
const int echoPin = 5;     // Echo pin of ultrasonic sensor
Servo garageServo;         // Create a servo object
long duration;
int distance;

// --- Module 3: Motion Sensor ---
const int motionSensorPin = 13;  // PIR sensor digital output
const int buzzerPin = 11;        // Buzzer (shared with fire alert system)

// --- Module 4: Rain Water Detection ---
const int rainSensorPin = 7;     // Digital pin connected to rain sensor
const int clothesServoPin = 8;   // Servo motor signal pin
Servo clothesServo;
```



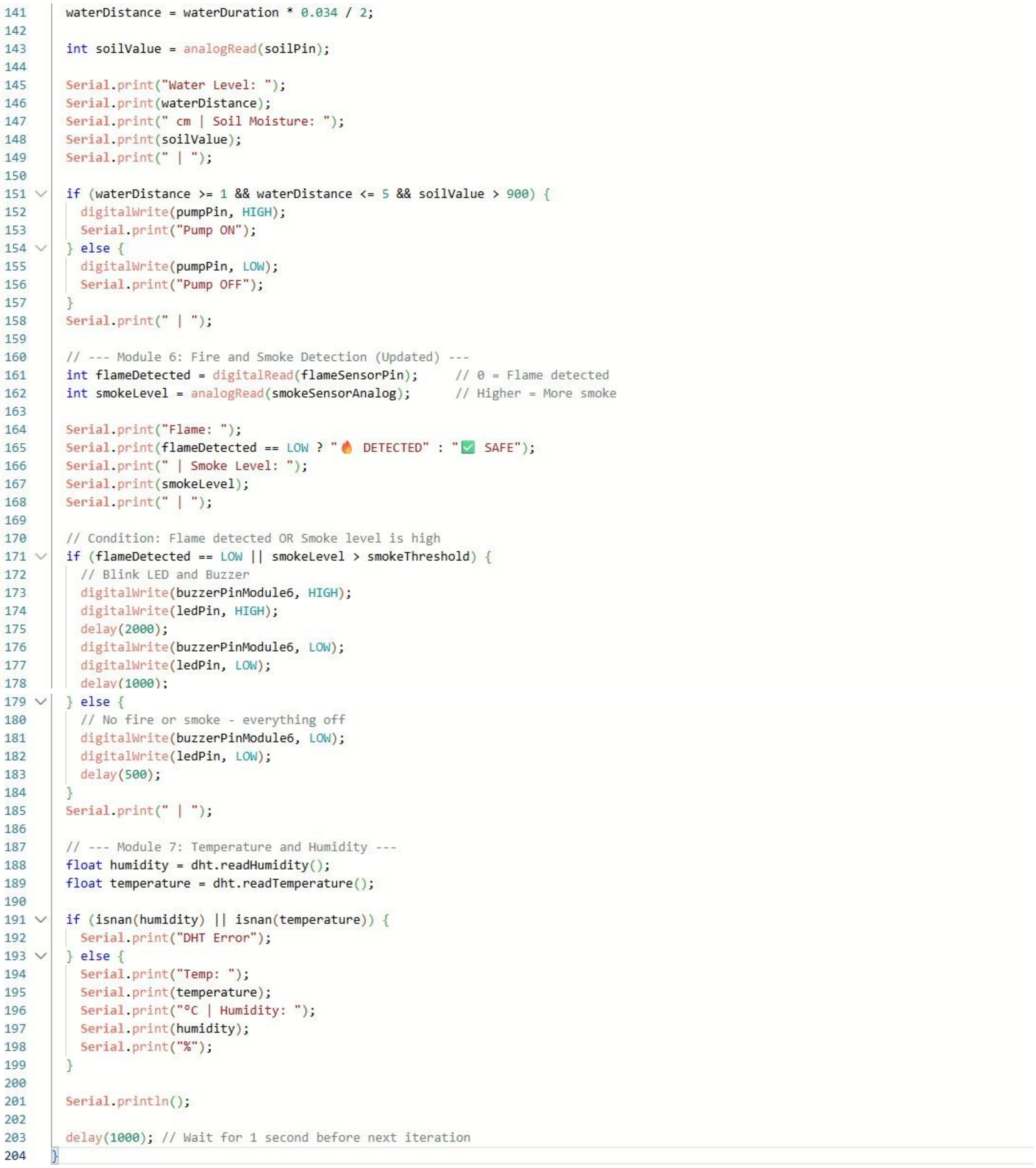
```

// --- Module 5: Soil Moisture and Water Level ---
const int soilPin = A5;          // Soil moisture sensor analog pin
const int pumpPin = 2;           // Digital pin for pump control
const int waterTrigPin = A3;     // Trigger pin of ultrasonic sensor for water level
const int waterEchoPin = A4;     // Echo pin of ultrasonic sensor for water level
long waterDuration;
int waterDistance;

// --- Module 6: Fire and Smoke Detection (Updated) ---
const int flameSensorPin = 9;      // IR Flame Sensor OUT → D9 (digital)
const int smokeSensorAnalog = A1;  // MQ2 Analog OUT → A1
const int buzzerPinModule6 = 11;   // Buzzer +ve → D11 (shared with Module 3)
const int ledPin = 12;             // Red LED anode → D12 via 220Ω resistor
const int smokeThreshold = 150;    // Threshold for smoke detection

// --- Module 7: Temperature and Humidity ---
#include "DHT.h"
#define DHTPIN A2       // Signal pin connected to A2
#define DHTTYPE DHT22   // We're using the DHT22 sensor
DHT dht(DHTPIN, DHTTYPE); // Create DHT sensor object
```



```

void setup() {
  // Initialize all digital pins
  pinMode(nightLed, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motionSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(rainSensorPin, INPUT);
  pinMode(pumpPin, OUTPUT);
  pinMode(flameSensorPin, INPUT);
  pinMode(smokeSensorAnalog, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(waterTrigPin, OUTPUT);
  pinMode(waterEchoPin, INPUT);

  // Attach servos
  garageServo.attach(6);
  clothesServo.attach(clothesServoPin);

  // Initialize components
  garageServo.write(0);    // Start with garage door closed
  clothesServo.write(90);  // Initial position (uncovered)
  digitalWrite(pumpPin, LOW); // Pump OFF initially
```

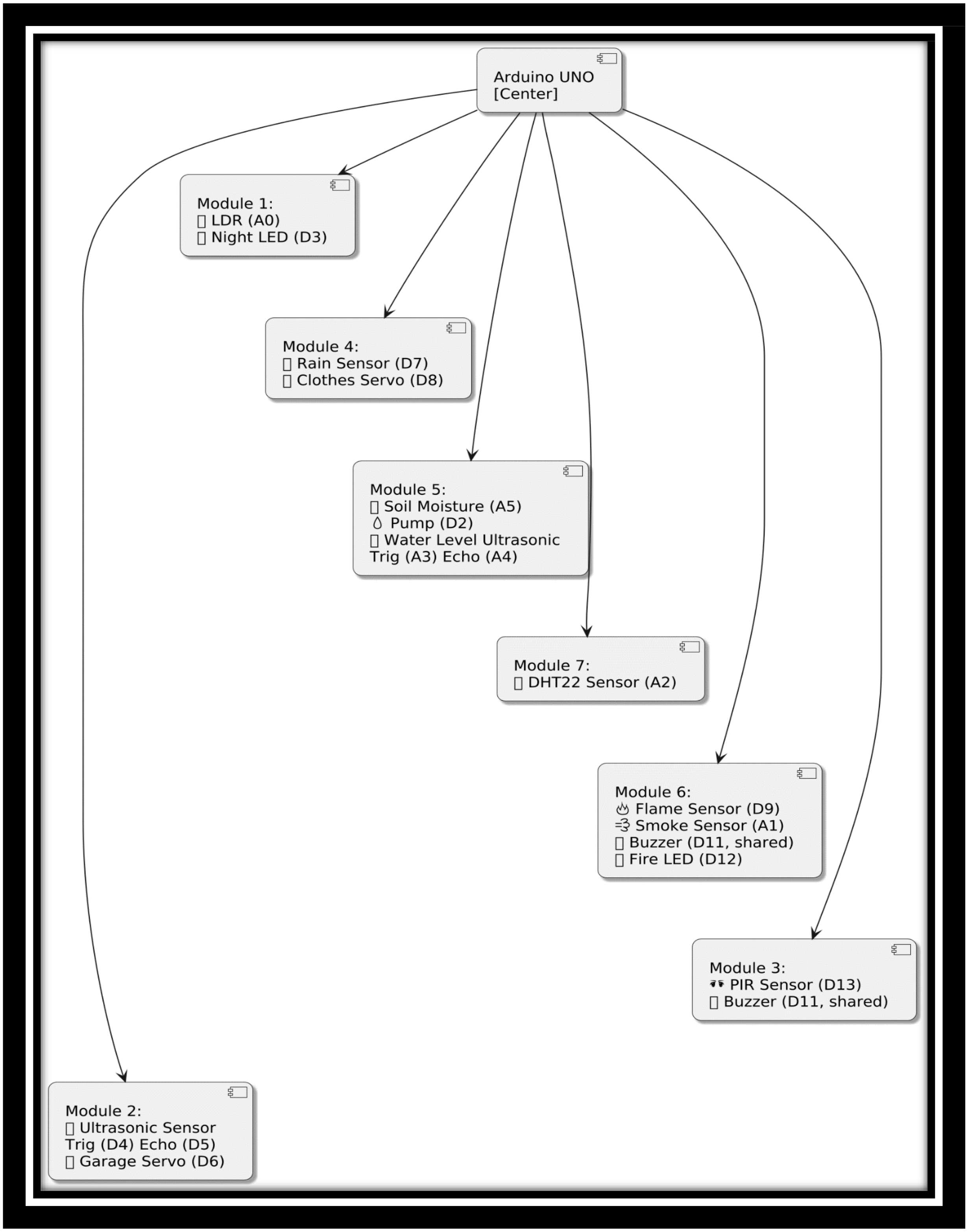


```
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize DHT sensor
  dht.begin();
}

void loop() {
  // --- Module 1: LDR and Night LED ---
  int ldrValue = analogRead(ldrPin);
  Serial.print("LDR Value: ");
  Serial.print(ldrValue);
  Serial.print(" | ");

  if (ldrValue > threshold) {
    digitalWrite(nightLed, LOW);
  } else {
    digitalWrite(nightLed, HIGH);
  }

  // --- Module 2: Garage Door ---
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  Serial.print("Garage Distance: ");
  Serial.print(distance);
  Serial.print(" cm | ");

  if (distance >= 5 && distance <= 15) {
    garageServo.write(180);
    delay(5000);
  } else {
    garageServo.write(0);
  }

  // --- Module 3: Motion Sensor ---
  int motionDetected = digitalRead(motionSensorPin);

  if (motionDetected == HIGH) {
    digitalWrite(buzzerPin, HIGH);
    Serial.print("Motion Detected!");
  } else {
    digitalWrite(buzzerPin, LOW);
    Serial.print("No Motion.");
  }
  Serial.print(" | ");

  // --- Module 4: Rain Water Detection ---
  int rainStatus = digitalRead(rainSensorPin);

  if (rainStatus == LOW) {
    clothesServo.write(0);
    Serial.print("Rain Detected!");
  } else {
    clothesServo.write(90);
    Serial.print("No Rain.");
  }
  Serial.print(" | ");

  // --- Module 5: Soil Moisture and Water Level ---
  digitalWrite(waterTrigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(waterTrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(waterTrigPin, LOW);

  waterDuration = pulseIn(waterEchoPin, HIGH);
```

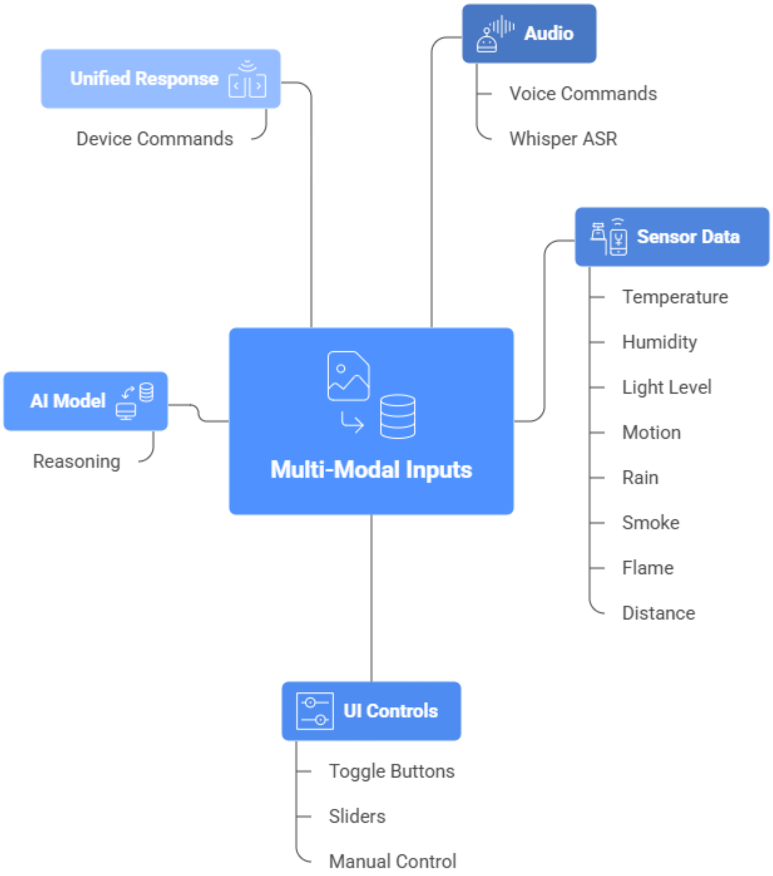


```
  waterDistance = waterDuration * 0.034 / 2;

  int soilValue = analogRead(soilPin);

  Serial.print("Water Level: ");
  Serial.print(waterDistance);
  Serial.print(" cm | Soil Moisture: ");
  Serial.print(soilValue);
  Serial.print(" | ");

  if (waterDistance >= 1 && waterDistance <= 5 && soilValue > 900) {
    digitalWrite(pumpPin, HIGH);
    Serial.print("Pump ON");
  } else {
    digitalWrite(pumpPin, LOW);
    Serial.print("Pump OFF");
  }
  Serial.print(" | ");

  // --- Module 6: Fire and Smoke Detection (Updated) ---
  int flameDetected = digitalRead(flameSensorPin);     // 0 = Flame detected
  int smokeLevel = analogRead(smokeSensorAnalog);      // Higher = More smoke

  Serial.print("Flame: ");
  Serial.print(flameDetected == LOW ? "🔥 DETECTED" : "✅ SAFE");
  Serial.print(" | Smoke Level: ");
  Serial.print(smokeLevel);
  Serial.print(" | ");

  // Condition: Flame detected OR Smoke level is high
  if (flameDetected == LOW || smokeLevel > smokeThreshold) {
    // Blink LED and Buzzer
    digitalWrite(buzzerPinModule6, HIGH);
    digitalWrite(ledPin, HIGH);
    delay(2000);
    digitalWrite(buzzerPinModule6, LOW);
    digitalWrite(ledPin, LOW);
    delay(1000);
  } else {
    // No fire or smoke - everything off
    digitalWrite(buzzerPinModule6, LOW);
    digitalWrite(ledPin, LOW);
    delay(500);
  }
  Serial.print(" | ");

  // --- Module 7: Temperature and Humidity ---
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.print("DHT Error");
  } else {
    Serial.print("Temp: ");
    Serial.print(temperature);
    Serial.print("°C | Humidity: ");
    Serial.print(humidity);
    Serial.print("%");
  }

  Serial.println();

  delay(1000); // Wait for 1 second before next iteration
}
```

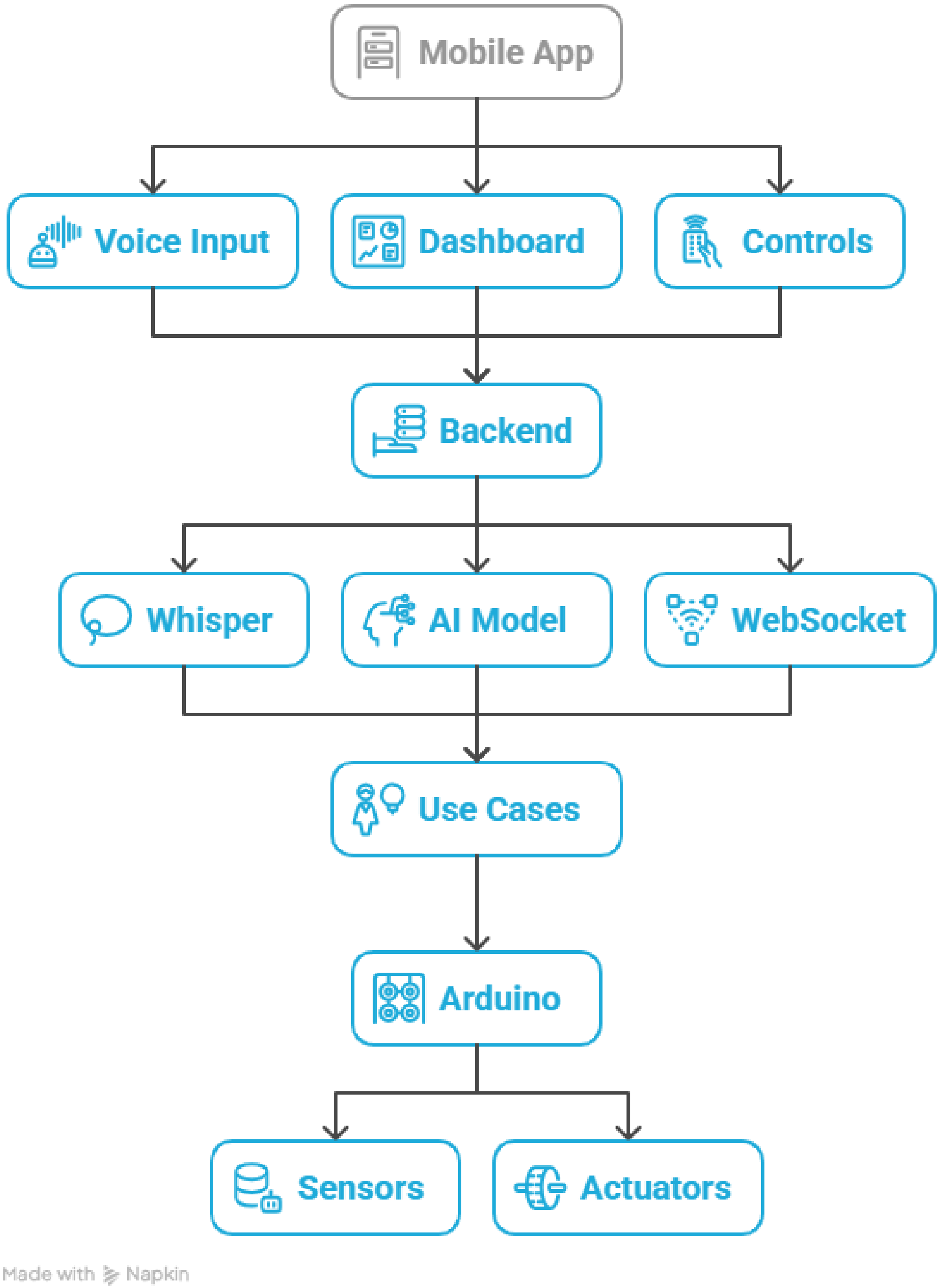
Block Diagram of the Project :



Functionality:



Project Architecture :


Mobile App
Voice Input
Dashboard
Controls
Backend
Whisper
AI Model
WebSocket
Use Cases
Arduino
Sensors
Actuators
Made with Napkin

Components:

Here's a complete list of **components with their module names**, indicating **which port or pin each component is connected to** in IoT project code:

Module 1: LDR and Night LED

Component	Description	Connected to
LDR Sensor	Light Dependent Resistor	A0 (Analog pin)
Night LED	LED turns ON in darkness	D3 (Digital pin)
Threshold	Value to detect darkness/light	Software only

Module 2: Garage Door (Ultrasonic & Servo)

Component	Description	Connected to
Ultrasonic Sensor	Detects distance for garage door	Trig: D4, Echo: D5
Garage Servo Motor	Opens/closes garage door	D6 (Digital PWM pin via garageServo.attach(6))

Module 3: Motion Sensor

Component	Description	Connected to

PIR Motion Sensor	Detects motion	D13 (Digital pin)
Buzzer	Sounds alarm on motion/fire	D11 (Shared with Module 6)

Module 4: Rain Water Detection

Component	Description	Connected to
Rain Sensor	Detects rain	D7 (Digital pin)
Clothes Servo	Moves to cover/uncover clothes	D8 (Digital PWM pin via clothesServo.attach(8))

Module 5: Soil Moisture and Water Level

Component	Description	Connected to
Soil Moisture Sensor	Detects soil wetness	A5 (Analog pin)
Pump	Waters plants	D2 (Digital pin)
Water Level Sensor (Ultrasonic)	Measures water tank level	Trig: A3, Echo: A4 (Analog pins used digitally)

Module 6: Fire and Smoke Detection

Component	Description	Connected to
IR Flame Sensor	Detects fire/flames	D9 (Digital pin)
MQ2 Smoke Sensor	Detects smoke levels	A1 (Analog pin)

Buzzer	Sounds alarm for fire/smoke	D11 (Shared with Module 3)
Red LED	Visual alert for fire/smoke	D12 (Digital pin)

Module 7: Temperature and Humidity

Component	**Description**	**Connected to**
DHT22 Sensor	Measures temperature & humidity	A2 (Digital use)

Summary of Pin Mapping

Pin	**Connected Component**
A0	LDR Sensor
A1	MQ2 Smoke Sensor
A2	DHT22 Sensor
A3	Water Level Sensor (Trigger)
A4	Water Level Sensor (Echo)
A5	Soil Moisture Sensor
D2	Pump
D3	Night LED
D4	Ultrasonic Sensor (Garage - Trigger)
D5	Ultrasonic Sensor (Garage - Echo)
D6	Garage Door Servo Motor
D7	Rain Sensor
D8	Clothes Servo Motor
D9	Flame Sensor
D11	Buzzer (Shared for Motion + Fire Alert)
D12	Fire Alert LED
D13	PIR Motion Sensor

Protocols :

Module 1: LDR and Night LED

- **Protocol:** None (Analog Input / Digital Output)
- **Explanation:**
  The LDR is read as an analog voltage (using `analogRead()` on A0), and the LED is controlled by a digital pin (`digitalWrite()` on pin 3). No communication protocol is involved here, just basic analog and digital signals.

Module 2: Garage Door (Ultrasonic Sensor and Servo)

- **Protocol:** Pulse-based signaling + PWM
- **Explanation:**
  - Ultrasonic sensor uses a **pulse timing** technique: a trigger pulse is sent (`digitalWrite()`), and the echo pulse duration is measured (`pulseIn()`), calculating distance based on the time of flight of the ultrasonic wave.
  - The servo is controlled via **PWM (Pulse Width Modulation)** signals generated by the Arduino servo library (`Servo.attach()` and `Servo.write()`).

Module 3: Motion Sensor (PIR) and Buzzer

- **Protocol:** Digital Input / Output
- **Explanation:**
  The PIR motion sensor outputs a digital HIGH or LOW signal detected on a digital input pin (`digitalRead()`), and the buzzer is turned on or off digitally (`digitalWrite()`). No advanced protocol is used.

Module 4: Rain Sensor and Servo

- **Protocol:** Digital Input / PWM Output
- **Explanation:**

  The rain sensor provides a digital signal indicating wet or dry (`digitalRead()`), and the clothes servo is controlled via PWM signals through the Servo library.

Module 5: Soil Moisture and Water Level (Ultrasonic Sensor and Pump)

- **Protocol:**
  - Soil Moisture: Analog Input
  - Water Level Ultrasonic: Pulse Timing
  - Pump Control: Digital Output
- **Explanation:**
  - Soil moisture sensor provides an analog voltage (`analogRead()`).
  - Water level ultrasonic sensor works the same way as Module 2 ultrasonic sensor using trigger and echo pulses.
  - Pump is turned on/off using digital output (`digitalWrite()`).

Module 6: Fire and Smoke Detection

- **Protocol:**
  - Flame Sensor: Digital Input
  - Smoke Sensor (MQ2): Analog Input
  - Buzzer and LED: Digital Output
- **Explanation:**
  - Flame sensor gives a digital HIGH or LOW signal depending on flame presence.
  - Smoke sensor provides an analog voltage level proportional to smoke concentration (`analogRead()`).
  - Buzzer and LED controlled digitally (`digitalWrite()`).

Module 7: Temperature and Humidity Sensor (DHT22)

- **Protocol: One-Wire / Single-Wire Digital Communication**
- **Explanation:**

  The DHT22 sensor uses a proprietary single-wire digital communication protocol to send temperature and humidity data to the Arduino. The `DHT` library handles the timing and data decoding over one digital pin (A2).

Summary Table of Protocols:

Module	Sensors / Devices	Protocol / Communication Type
1	LDR, LED	Analog Input, Digital Output
2	Ultrasonic sensor, Servo	Pulse timing, PWM
3	PIR sensor, Buzzer	Digital Input, Digital Output
4	Rain sensor, Servo	Digital Input, PWM
5	Soil moisture, Ultrasonic	Analog Input, Pulse timing, Digital Output
6	Flame, Smoke sensors	Digital Input, Analog Input, Digital Output
7	DHT22 sensor	One-Wire digital protocol (single-wire communication)

18.CONCLUSION

This Smart Home and Agriculture System demonstrates how affordable microcontrollers and basic sensors can automate real-life activities. The project effectively integrates environmental monitoring with responsive actions to create a safe, efficient, and smart environment for homes and farms.