An-Najah National University

Faculty of Engineering and Information Technology

Department of Computer Engineering

Graduation Project II

Smart Home Automation System Using ESP and IoT

SmartNest

Supervisor:
Dr. Hanal Abu-Zant

Presented By:
Saba Majdi Shoqo

Submitted in partial fulfillment of the requirements for a bachelor's degree in Computer Engineering

Graduation Project II

Smart Home Automation System Using ESP and IoT

SmartNest

Abstract:

- ESP32-based IoT system to monitor and automate home functions.
- Controls lighting, irrigation, fire alerts, and temperature control when temperatures rise.
- Uses a simple HTML dashboard controller.
- Aims for low cost, energy efficiency, and safety.

Problem Statement:

- Manual climate and lighting control
- Wasted energy and water
- Limited safety alerts
- High cost of smart systems
- Low accessibility for non-technical users

Project Objectives:

- Real-time monitoring and automation
- Reduce resource waste
- Improve safety
- Enable remote control
- Support modular system design

Scope of the Project:

- Monitor temperature, humidity, motion, fire, and soil
- Auto-control fan, pump, and lights
- Wi-Fi communication using ESP32
- User interface via HTML Dashboard
- Next step, Ad Hoc System Using ESP Modules

Constraints

- Limited budget and solo development
- Component availability
- Wi-Fi dependency
- ESP32 processing limits

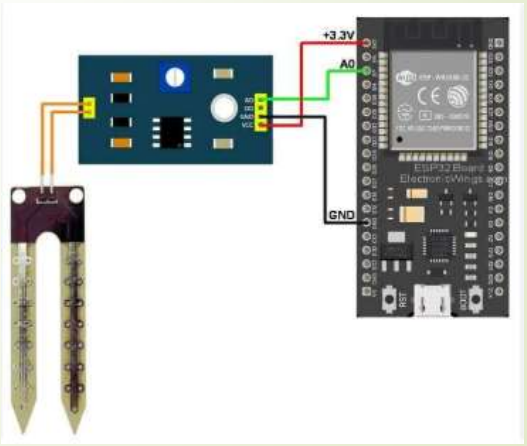
Hardware Components

ESP32 (Main Controller)

Receives commands from the user or a physical button and sends control signals to the second ESP32 over Wi-Fi.

ESP32 (Sensor Node)

Reads data from sensors (temperature, fire, motion, soil) and controls outputs (fan, lights, pump) based on commands or automatic conditions.



Hardware Components

DHT11 Sensor

Measures temperature and humidity. If the temperature is too high, it triggers the fan to maintain comfort.

PIR Motion Sensor

Detects movement inside the house. It automatically turns on the lights and alerts the system of activity.

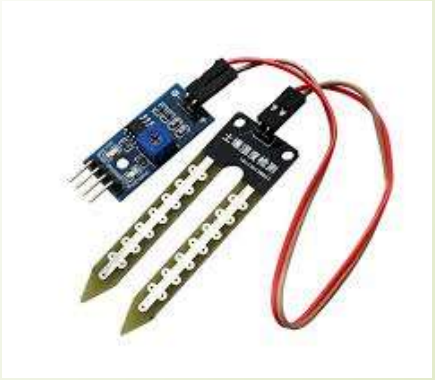
Hardware Components

Flame Sensor

Detects fire or high heat. When triggered, it activates the buzzer and warning lights for safety.

Soil Moisture Sensor

Measures how dry the soil is. When dryness crosses a set threshold, the system turns on the water pump for irrigation.



Hardware Components

Fan (DC Motor)

Used to cool the environment automatically when high temperature is detected.

Mini Water Pump

Irrigates plants based on readings from the soil moisture sensor. Helps save water by running only when needed.

Hardware Components

Relay Modules (x3)

Allow the ESP32 to safely control the fan and pump, which use higher voltages than the microcontroller.

LEDs Light (Interior & Exterior)

Act as lights and indicators. Some LEDs turn on with motion, others signal alerts (like fire).

Hardware Components

Buzzer

Used for audible alerts during fire detection or other critical system states.

System Switch

Connected to the main ESP32 to manually turn the system on or off without using the app.

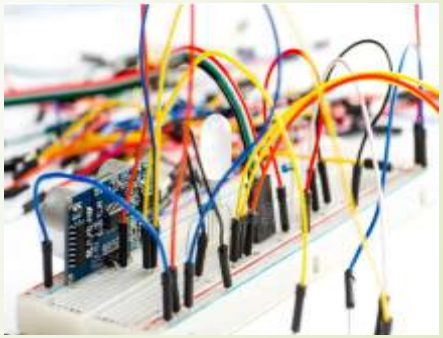
Hardware Components

Power Supply
(Lithium Battery or USB Adapter)

Provides power to both ESP32 units and all connected sensors and actuators.

Breadboards and jumper wires for prototyping

Used in early prototyping to connect components



3D-Printed House Mode

Holds and organizes all components in a realistic, physical layout for demonstration and testing.

This hardware setup was designed to be compact, affordable, and functional for real-time smart home automation, with flexibility for future upgrades.

System Design:

- Two ESP32s (Controller & Sensor Node)
- HTTP + JSON communication
- Device control based on sensor input

Component	ESP GPIO	Mode
DHT11	GPIO 4	Input
PIR Sensor	GPIO 14	Input
Flame Sensor	GPIO 12	Input
Soil Sensor	GPIO 34	Analog
DC Fan	GPIO 5	Output
Water Pump	GPIO 27	Output
LED Light	GPIO 2	Output
Buzzer	GPIO 26	Output
Button	GPIO 15	Input

System Design:

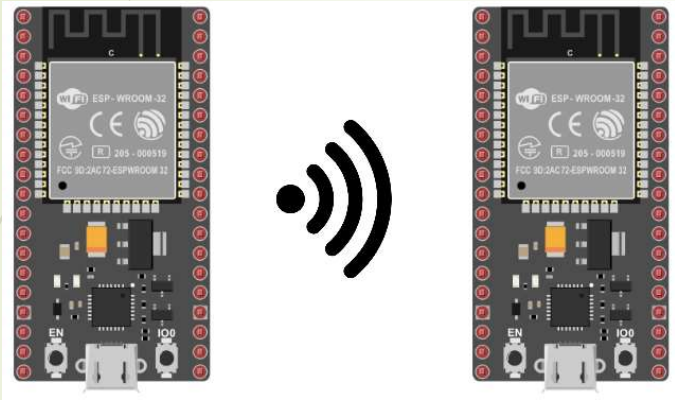
The Sensor Node connects to multiple sensors including:

- DHT11 for temperature and humidity.
- PIR motion sensor for detecting presence.
- Flame sensor for fire alerts.
- Soil moisture sensor for smart irrigation.

It also controls output devices:

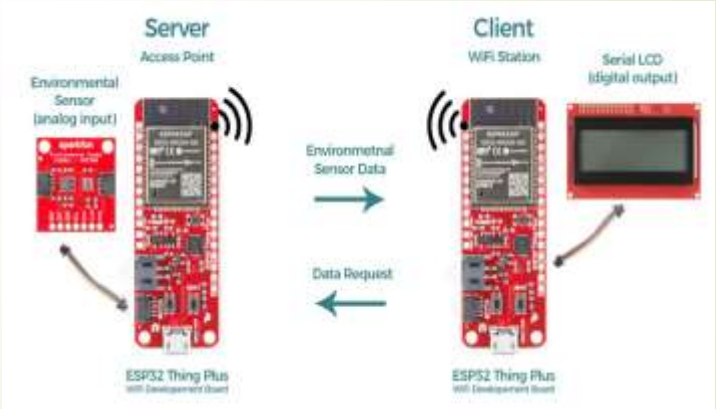
- Fan for cooling.
- Pump for watering.
- LED lights for status or indoor lighting.
- Buzzer for audible alerts

System Design:



The system is made up of two ESP32 boards connected over Wi-Fi. One acts as the Main Controller, which handles input from the user, while the second board is the Sensor Node, responsible for reading data from sensors and activating outputs.

Control commands such as turning the LED or fan on/off are sent using HTTP requests via fetch, for example: http://192.168.1.54/led_on. Sensor readings are automatically fetched through a request to http://192.168.1.54/status, which returns a JSON response containing temperature, humidity, and soil moisture values



Final Product

After validating the system in Proteus, the full circuit was built using breadboards, jumper wires, and relays. Finally, the components were installed inside a 3D-printed house model, with LED lighting for visual feedback. The entire system runs on a 5V USB supply, and can optionally use a rechargeable battery.

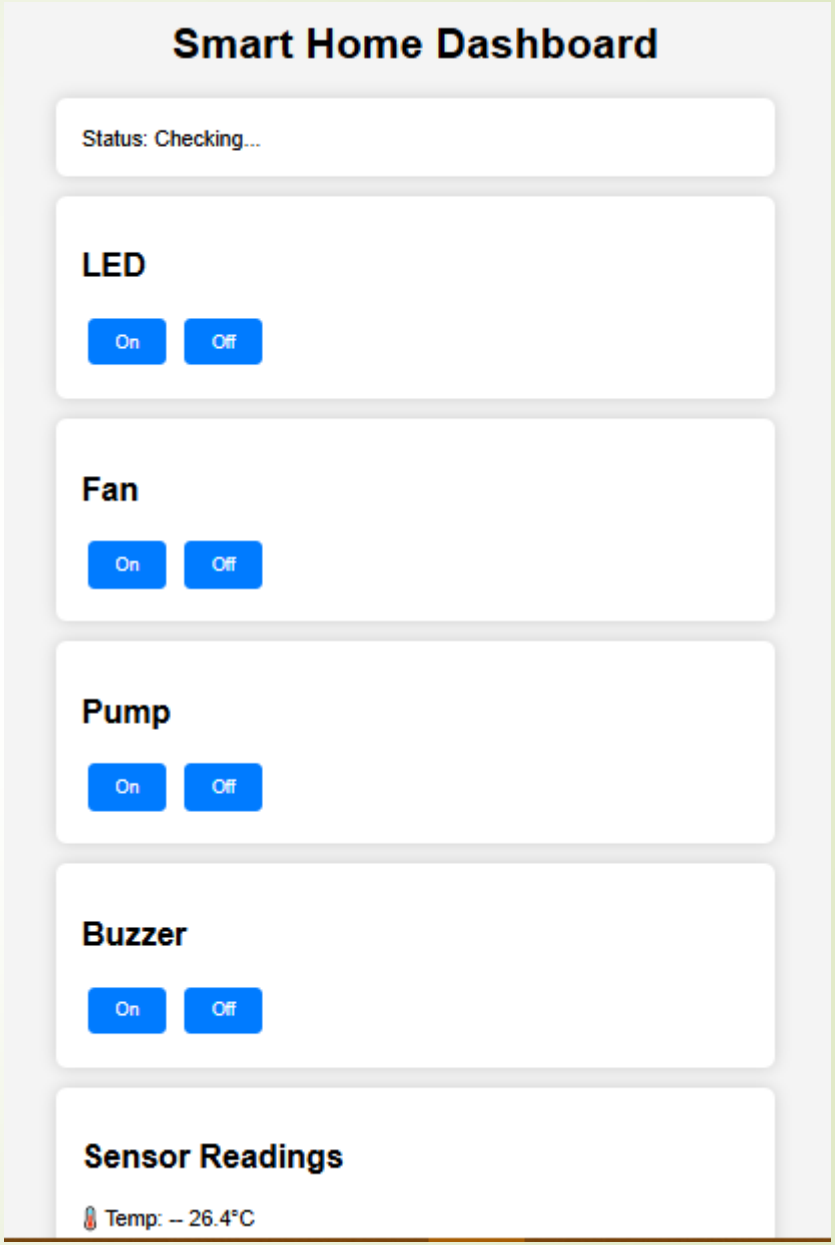
This modular and wireless setup allows the system to grow easily by adding more devices in the future - without needing to rebuild everything.

Workflow:

- Wi-Fi connection and server initialization
- Automatic & manual control via HTML page.
- Real-time sensor monitoring
- JSON API for status and control

Web Interface:

- Built using HTML
- Dashboard for sensor data
- Interactive control switches



Future Work

cloud storage
push notifications
voice assistant support
Ad hoc System

Ad Hoc System Using ESP Modules

- A smart system using ESP32/ESP8266 microcontrollers

- Devices connect wirelessly (Ad Hoc / ESP-NOW)
- no internet needed

- Includes sensors (temperature, motion, etc.)
- and output devices (LEDs, screens)

Future Work: Ad hoc System

How It Works

- Each ESP collects data from sensors
- ESP modules communicate directly with each other
- A main controller processes data and sends outputs
- Used for interactive STEM experiments

Benefits & Applications

- Teaches IoT, coding, and wireless communication
- Hands-on learning and teamwork

Example Use:

- Physics lab: measure motion or temperature
- Data sent wirelessly and displayed instantly

Results:

- Fast and reliable responses
- Accurate sensor readings
- Synchronized control methods
- Clear user feedback and error alerts