

Smart Home Automation PDF – Detailed Version

1. Abstract

What is smart home automation?

- Importance for energy efficiency, security, comfort
- Overview of Arduino, ESP32, Raspberry Pi in automation

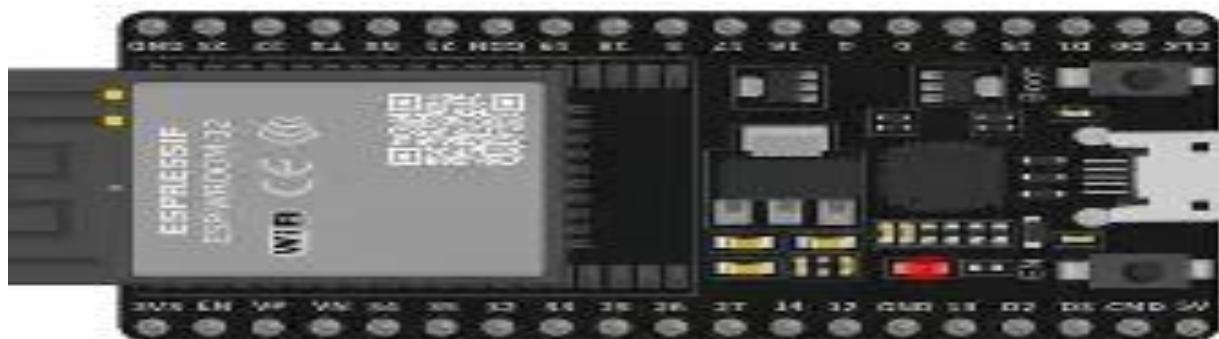
2. Components and Sensors (with Details)

- Arduino



- DHT11/DHT22 – Temperature & Humidity sensor
- PIR Sensor – Motion detection
- LDR – Light intensity detection
- Relay Module – Control AC appliances (lights/fan)

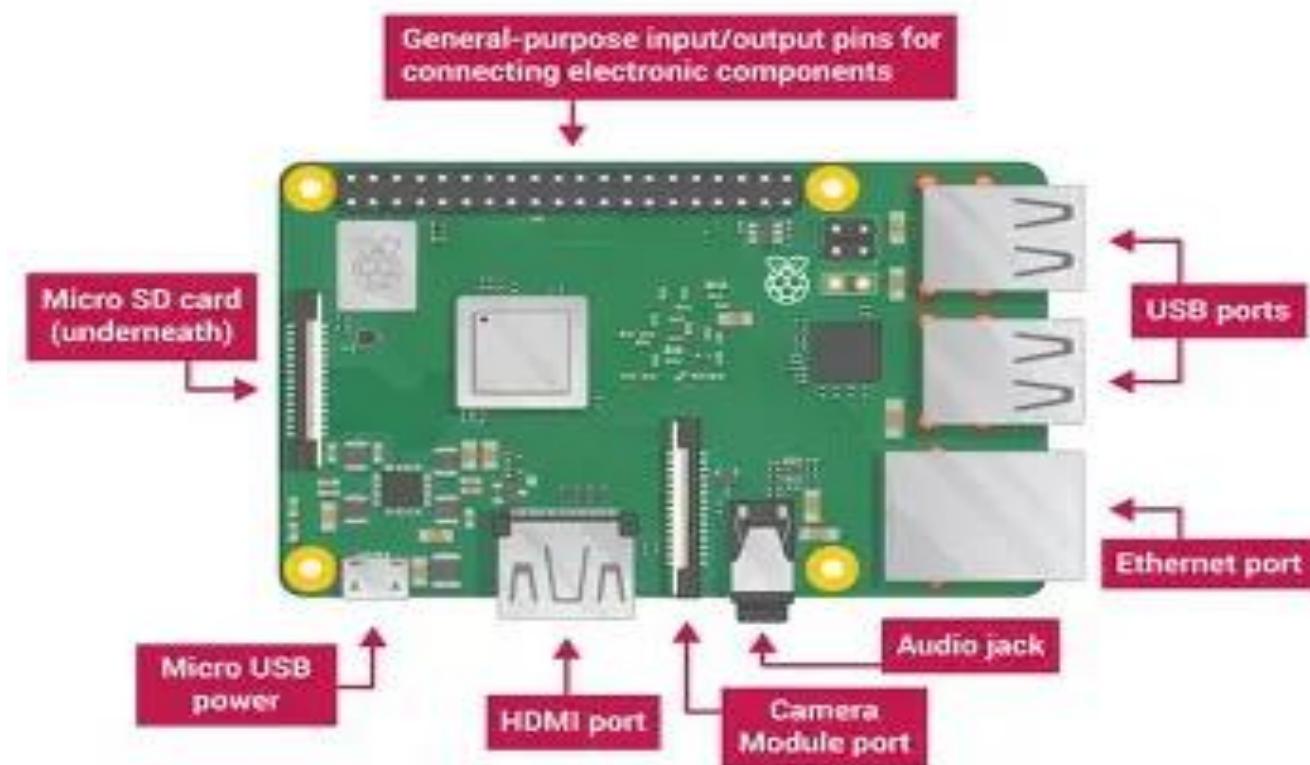
- ESP32



- Same as Arduino

- Built-in Wi-Fi for IoT remote control

- Raspberry Pi



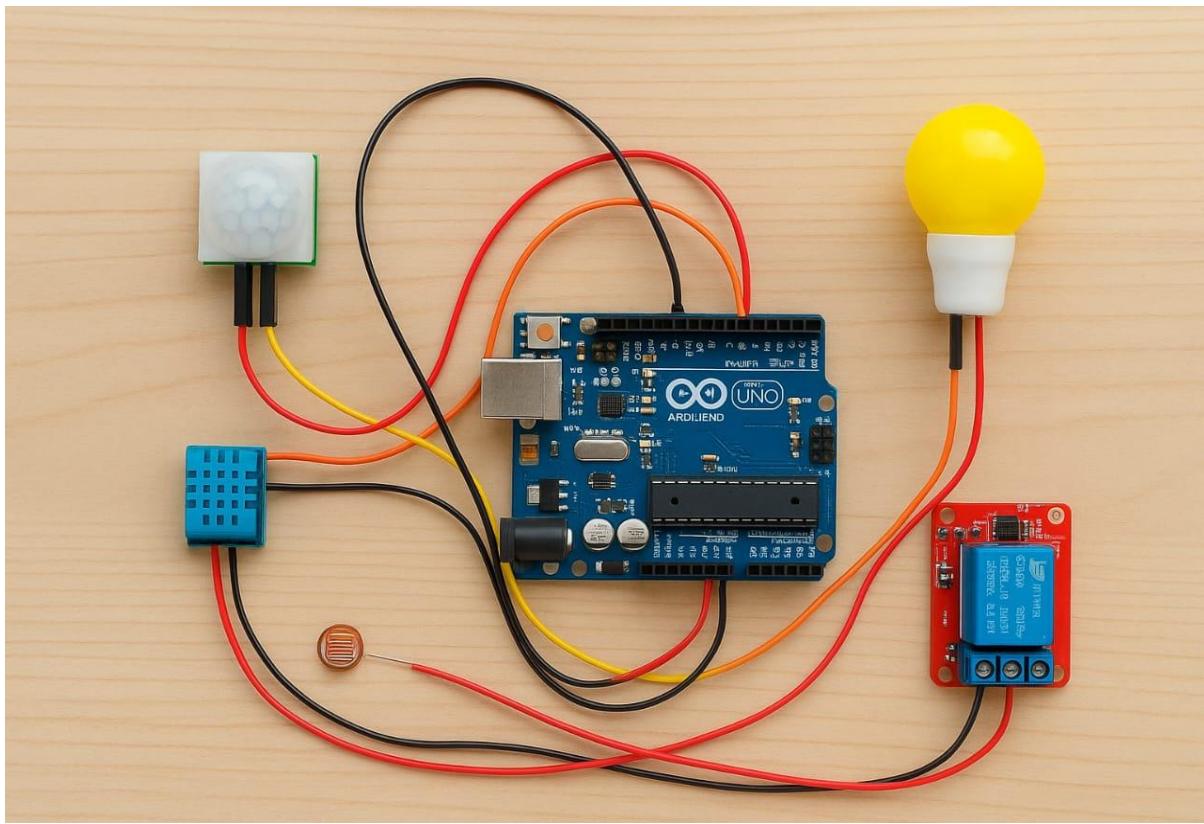
- Camera module – Video monitoring
- Motion sensor – Security
- Relay Module – Appliance control
- Optional: Voice control integration (Google Assistant/Alexa)

3. Software / Programming Tools

- Arduino IDE for Arduino & ESP32
- Python with GPIO library for Raspberry Pi
- IoT Platforms:
 - Blynk – Mobile control
 - ThingSpeak / MQTT – Data monitoring & cloud control

4. Prototypes

This is the prototype for the smart home automation projects.



5. Step-by-Step Implementation

- **Arduino Version**

1. Connect DHT11, PIR, LDR, and Relay to Arduino pins
2. Upload Arduino code to read sensors & control devices
3. Test each sensor separately (temperature, motion, light)

4. Control appliances via relay (ON/OFF)

- **ESP32 IoT Version**

1. Connect sensors & relay modules.

2. Connect ESP32 to Wi-Fi.

3. Use Blynk app or ThingSpeak to monitor sensors & control devices remotely.

4. Test remote control of lights/fan.

- **Raspberry Pi Version**

1. Connect sensors and relay

2. Write Python scripts to read sensors & control devices

3. Optional: Integrate camera module for motion detection alerts

4. Optional: Voice control with Google Assistant / Alexa

6. Sample Codes

Arduino: Light ON/OFF with LDR, motion alert with PIR

Component	Arduino Pin	Description
LDR + 10kΩ	A0	Analog input for light intensity
PIR Sensor OUT	D2	Digital input for motion
LED	D9	Output for light ON/OFF
Buzzer	D8	Output for alert sound
5V & GND	5V & GND	Common power and ground

```
int ldrPin = A0;      // LDR input
int pirPin = 2;       // PIR input
int ledPin = 9;       // LED output
int buzzerPin = 8;    // Buzzer output

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    pinMode(pirPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    int ldrValue = analogRead(ldrPin); // Read LDR value
    int pirValue = digitalRead(pirPin); // Read PIR value

    Serial.print("LDR: ");
    Serial.print(ldrValue);
    Serial.print(" | PIR: ");
    Serial.println(pirValue);

    // Light control based on LDR
    if (ldrValue < 500) { // 500 → threshold (dark)
```

```

    digitalWrite(ledPin, HIGH); // Turn ON light
} else {
    digitalWrite(ledPin, LOW); // Turn OFF light
}

// Motion alert using PIR
if (pirValue == HIGH) {
    digitalWrite(buzzerPin, HIGH); // Buzzer ON
    delay(1000);
    digitalWrite(buzzerPin, LOW); // Buzzer OFF
}

delay(200);
}

```

ESP32: IoT control of devices using Blynk

Component	ESP32 Pin	Description
LED 1	D2 (GPIO 2)	Device 1 control
LED 2	D4 (GPIO 4)	Device 2 control
GND	GND	Common ground

```
#define BLYNK_TEMPLATE_ID "Your_Template_ID"
#define BLYNK_DEVICE_NAME "Device_Control"
#define BLYNK_AUTH_TOKEN "Your_Auth_Token"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Your_WiFi_Name";    // WiFi name
char pass[] = "Your_WiFi_Password"; // WiFi password

int led1 = 2; // Device 1
int led2 = 4; // Device 2

void setup() {
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
}

// Virtual Pin V0 - LED1
BLYNK_WRITE(V0) {
    int pinValue = param.asInt();
```

```

digitalWrite(led1, pinValue);
}

// Virtual Pin V1 - LED2
BLYNK_WRITE(V1) {
    int pinValue = param.asInt();
    digitalWrite(led2, pinValue);
}

void loop() {
    Blynk.run();
}

```

Raspberry Pi: Python script for temperature monitoring & appliance control

Component	Raspberry Pi Pin	Description
DHT11 VCC	5 V	Power
DHT11 GND	GND	Ground
DHT11 DATA	GPIO 4 (Pin 7)	Data Pin
Relay IN	GPIO 17 (Pin 11)	Control Signal
Relay VCC	5 V	Power

Relay GND	GND	Common Ground
Fan/LED	Relay Output	Appliance

```

import Adafruit_DHT
import RPi.GPIO as GPIO
import time

# Sensor setup
sensor = Adafruit_DHT.DHT11
pin = 4 # GPIO4 for DHT11 data

# Relay setup
relay_pin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(relay_pin, GPIO.OUT)
GPIO.output(relay_pin, GPIO.LOW)

try:
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(sensor,
pin)
        if humidity is not None and temperature is not None:
            print(f"Temp={temperature:.1f}°C
Humidity={humidity:.1f}%")

```

```

# Control logic

if temperature > 30: # threshold (30°C)
    GPIO.output(relay_pin, GPIO.HIGH) # Turn ON fan
    print("🔥 High Temp! Fan ON")

else:
    GPIO.output(relay_pin, GPIO.LOW) # Turn OFF fan
    print("✅ Normal Temp. Fan OFF")

else:
    print("Sensor failure! Check wiring.")

time.sleep(2)

except KeyboardInterrupt:
    GPIO.cleanup()
    print("Program stopped")

```

7. Future Scope / Enhancements

- Add smart door lock control
- Integrate AI-based voice assistant

- Energy usage monitoring & automated scheduling

Full Project Details with Code

- 👉 Use ESP32 + Blynk IoT App to control home appliances (Fan, Light, etc.) from mobile via Wi-Fi/Internet.
 - 👉 Add optional Sensors (like DHT11) to monitor temperature & humidity.
 - 👉 You can view readings + control switches remotely .
-

Components Required

1. ESP32 board
 2. 4-channel relay module
 3. Light/Fan/LEDs (4 devices)
 4. DHT11 sensor (*optional*)
 5. Jumper wires + Breadboard
 6. Blynk IoT App (Android/iOS)
 7. Wi-Fi connection
-

Blynk App Setup (New Blynk IoT)

1. Open Blynk IoT app → create new project.
2. Select:
 - Device: ESP32
 - Connection: Wi-Fi
3. Add 4 Button widgets (for 4 devices):

- V0 → Light
- V1 → Fan
- V2 → TV
- V3 → AC

4. Add 2 Display widgets (for Temperature & Humidity):

- V4 → Temperature
- V5 → Humidity

5. Set all Buttons in Switch mode.

6. Note down your Auth Token (from email or dashboard).

⚡ Circuit Connections

Component	ESP32 Pin	Description
Relay 1 (Light)	D23	V0
Relay 2 (Fan)	D22	V1
Relay 3 (TV)	D21	V2
Relay 4 (AC)	D19	V3
DHT11 Data	D4	V4 & V5 readings
VCC & GND	3.3V & GND	Common Power

✳️ Complete Arduino Code

```
// ----- Blynk Smart Home Automation with ESP32 -----
#define BLYNK_TEMPLATE_ID "Your_Template_ID"
#define BLYNK_DEVICE_NAME "SmartHome"
#define BLYNK_AUTH_TOKEN "Your_Auth_Token"
```

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "DHT.h"

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Your_WiFi_Name";    // WiFi Name
char pass[] = "Your_WiFi_Password"; // WiFi Password

#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Relay Pins
int relay1 = 23; // Light
int relay2 = 22; // Fan
int relay3 = 21; // TV
int relay4 = 19; // AC

BlynkTimer timer;

// --- Update sensor data to Blynk ---
void sendSensor()
{
```

```
float h = dht.readHumidity();
float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {
    Serial.println("✖ DHT Sensor Error!");
    return;
}

Blynk.virtualWrite(V4, t);
Blynk.virtualWrite(V5, h);

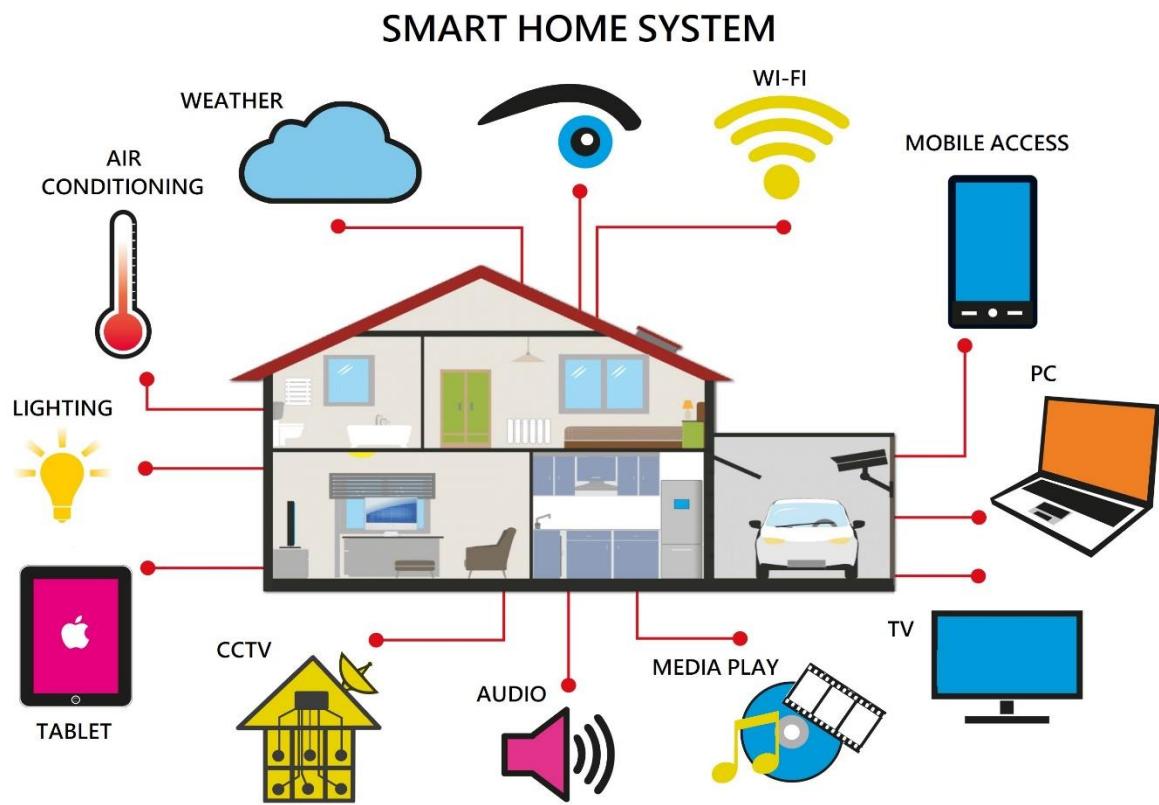
Serial.print("🌡 Temp: ");
Serial.print(t);
Serial.print("°C 💧 Humidity: ");
Serial.print(h);
Serial.println("%");
}

// --- Blynk control functions ---
BLYNK_WRITE(V0) { digitalWrite(relay1, param.asInt()); } // Light
BLYNK_WRITE(V1) { digitalWrite(relay2, param.asInt()); } // Fan
BLYNK_WRITE(V2) { digitalWrite(relay3, param.asInt()); } // TV
BLYNK_WRITE(V3) { digitalWrite(relay4, param.asInt()); } // AC

void setup()
```

```
{  
    Serial.begin(115200);  
    Blynk.begin(auth, ssid, pass);  
    dht.begin();  
  
    pinMode(relay1, OUTPUT);  
    pinMode(relay2, OUTPUT);  
    pinMode(relay3, OUTPUT);  
    pinMode(relay4, OUTPUT);  
  
    digitalWrite(relay1, LOW);  
    digitalWrite(relay2, LOW);  
    digitalWrite(relay3, LOW);  
    digitalWrite(relay4, LOW);  
  
    timer.setInterval(2000L, sendSensor); // every 2 sec update  
    temp/humidity  
}  
  
void loop()  
{  
    Blynk.run();  
    timer.run();  
}
```

Overall project Diagram



Summary

This is the smart home automation using the embedded systems in IoT.