

Home Automation System

Prathyusha Baikani (Robotic Engineer)

1. Project Overview

The Home Automation System allows users to control household devices like lights, fans, air conditioners, and other appliances remotely using a smartphone. The system utilizes sensors (like temperature, humidity, and motion), relay modules, and an Arduino Uno to enable automated control of the devices. It can work over Wi-Fi (using an ESP8266/ESP32) or Bluetooth (using an HC-05/HC-06 Bluetooth module) for communication.

The primary goal of the system is to increase convenience, save energy, and make the home environment smarter by integrating motion detection and environmental conditions.

2. Components Required

- **Arduino Uno** (1x) - The microcontroller to control the system.
- **Relay Modules** (4x) - Used to switch on/off high-voltage devices.
- **Motion Sensor (PIR)** (1x) - Detects motion for automatic lighting.
- **DHT11/DHT22 Sensor** (1x) - Measures temperature and humidity.
- **ESP8266 or ESP32** (1x) - For Wi-Fi connectivity (for remote control via smartphone).
- **HC-05 Bluetooth Module** (1x) - Optional for Bluetooth control.
- **Smartphone** (1x) - To control the system remotely through an app (Blynk or custom app).
- **Wires and Breadboard** - For connections.
- **AC-powered devices** - Like lights, fans, and an AC unit for control.

3. System Design

- **Relay Modules** are used to control high-voltage devices like lights, fans, and ACs.

- **Sensors** are used for detecting environmental conditions. For instance:
 - **PIR Motion Sensor** detects the presence of a person.
 - **DHT Sensor** measures temperature and humidity.
- **Smartphone control** via Wi-Fi (ESP8266/ESP32) or Bluetooth (HC-05) allows remote operation of the system. The system can turn on/off devices like lights or fans based on the data from the sensors.

4. Circuit Diagram

Here is a general **circuit diagram** of how the components are connected:

- **Relay Modules:**
 - Pin **IN1** to Arduino Pin 8 (for controlling light).
 - Pin **IN2** to Arduino Pin 9 (for controlling fan).
 - Pin **IN3** to Arduino Pin 10 (for controlling AC).
 - Pin **IN4** to Arduino Pin 11 (for controlling another device).
- **PIR Motion Sensor:**
 - Pin **OUT** of PIR to Arduino Pin 7.
- **DHT Sensor:**
 - Pin **VCC** to Arduino 5V.
 - Pin **GND** to Arduino GND.
 - Pin **DATA** to Arduino Pin 6.
- **ESP8266/ESP32 (Wi-Fi Module) or HC-05 (Bluetooth Module):**
 - Connect **TX** and **RX** of ESP8266/ESP32 to Arduino **RX** and **TX** pins for communication (for Wi-Fi control).
 - If using Bluetooth, connect the **TX** and **RX** pins of the HC-05 to Arduino's **RX** and **TX** respectively.

5. Software Setup

A. Arduino Code

Below is the Arduino code for controlling devices with motion detection and sensor input. We'll be using the **ESP8266** module for Wi-Fi connectivity and **Blynk** app for smartphone control.

1. Install Libraries:

- Install the following libraries via the Arduino IDE:
 - **DHT sensor library** (for temperature and humidity sensor).
 - **ESP8266WiFi library** (for Wi-Fi communication).
 - **Blynk library** (for smartphone control).

2. Use the **Library Manager** in Arduino IDE (**Sketch > Include Library > Manage Libraries**) to install them.

3. Arduino Code:

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

// Wi-Fi credentials
char auth[] = "YourBlynkAuthToken"; // Blynk Auth Token
char ssid[] = "YourWiFiSSID";
char pass[] = "YourWiFiPassword";

// Pin definitions
#define RELAY_PIN_1 8
#define RELAY_PIN_2 9
#define RELAY_PIN_3 10
#define RELAY_PIN_4 11
#define PIR_SENSOR_PIN 7
#define DHT_PIN 6

DHT dht(DHT_PIN, DHT11); // DHT11 sensor
BlynkTimer timer;

void setup() {
  // Serial monitor setup
  Serial.begin(9600);
  WiFi.begin(ssid, pass);
  Blynk.begin(auth, ssid, pass);

  // Pin setup
  pinMode(RELAY_PIN_1, OUTPUT);
  pinMode(RELAY_PIN_2, OUTPUT);
  pinMode(RELAY_PIN_3, OUTPUT);
  pinMode(RELAY_PIN_4, OUTPUT);
  pinMode(PIR_SENSOR_PIN, INPUT);
}
```

```

// Initialize DHT sensor
dht.begin();

// Set up a timer to send sensor data to the Blynk app
timer.setInterval(1000L, sendSensorData);
}

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

// Function to send temperature and humidity data to the app
void sendSensorData() {
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Send data to the Blynk app
  Blynk.virtualWrite(V1, temp);    // Virtual pin V1 for temperature
  Blynk.virtualWrite(V2, humidity); // Virtual pin V2 for humidity
}

// Function to control devices based on motion
void controlDevicesBasedOnMotion() {
  int motionDetected = digitalRead(PIR_SENSOR_PIN);
  if (motionDetected == HIGH) {
    // Turn on devices
    digitalWrite(RELAY_PIN_1, HIGH); // Turn on light
    digitalWrite(RELAY_PIN_2, HIGH); // Turn on fan
  } else {
    // Turn off devices
    digitalWrite(RELAY_PIN_1, LOW);  // Turn off light
    digitalWrite(RELAY_PIN_2, LOW);  // Turn off fan
  }
}
}

```

B. Blynk App Setup

1. **Install the Blynk App** on your smartphone (available for iOS and Android).
2. **Create a New Project** in the app, select **ESP8266** as the device type, and note the **Auth Token** (you will use it in the code).
3. **Add Widgets:**
 - **Temperature Widget:** Set it to Virtual Pin V1 to display the temperature.
 - **Humidity Widget:** Set it to Virtual Pin V2 to display the humidity.
 - **Button Widgets:** For controlling other devices (lights, fans, etc.), link them to the corresponding relay pins.
4. **Configure Button Widgets:**
 - When the button is pressed, it will control the relay (e.g., turning a device on/off).

6. Working of the System

- **Wi-Fi Communication:** The ESP8266 module connects to your home Wi-Fi network and communicates with the Blynk app on your smartphone.
- **Sensor Feedback:** The system continuously reads data from the **motion sensor** (PIR) and **DHT11** (for temperature and humidity).
 - If motion is detected, the relay modules turn on the devices (e.g., lights, fans).
 - If no motion is detected, the devices are turned off.
- **Smartphone Control:** Through the Blynk app, you can manually control devices and monitor the temperature and humidity of the room.

7. Troubleshooting

1. **No Wi-Fi connection:** Make sure you have the correct Wi-Fi credentials and that the ESP8266 module is within range of your router.
2. **Devices not turning on/off:** Double-check the relay connections and ensure that the relay modules are receiving power.
3. **Sensor not responding:** Ensure the DHT sensor is wired correctly and the **DHT library** is installed.

8. Future Enhancements

- **Add More Sensors:** You can add more sensors like light sensors or gas sensors.
- **Voice Control:** Implement voice control using Amazon Alexa or Google Assistant.
- **Energy Monitoring:** Add energy monitoring to track power consumption of the connected devices.

9. Conclusion

This Home Automation System is a simple yet powerful project that leverages **sensors**, **relay modules**, and **smartphone control** to automate household devices. By integrating **motion sensors** and environmental data, you can automate actions like turning on the lights when someone enters the room or adjusting the AC based on temperature.

-----THANK YOU-----