# Title: Smart Home Automation using Amazon Alexand Echo Dot

# INTRODUCTION:

This project aims to design and implement a smart home automation system using Amazon Alexa and Echo Dot. The system enables voice control of various smart devices, including lights, thermostats, security cameras, and door locks, to create a convenient and secure home environment.

Using Alexa's voice recognition capabilities and Echo Dot's compact design, users can control their smart devices with simple voice commands, eliminating the need for manual switching or remote controls. The system integrates with various smart devices and platforms to provide a seamless and comprehensive automation experience.

# COMPONENTS USED :

**. Alexa dot**

**.ESP 32**

**.8 relay module**

**.Lights**

**.Switch**

**.Wires**

## HARDWARE USED :

**ESP-32**

The ESP32 is a low-cost, low-power system-on-a-chip (SoC) microcontroller with integrated Wi-Fi and Bluetooth capabilities. It features a dual-core 32-bit LX6 microprocessor operating at 160 MHz or 240 MHz, with a maximum clock speed of 240 MHz. The ESP32 has 4MB of flash memory and 520 KB of SRAM, allowing for efficient storage and execution of code. It also features 14 GPIO pins, which can be used for digital input/output operations, I2C, I2S, UART, SPI, and more. Additionally, the ESP32 has built-in power management capabilities, including sleep modes and dynamic voltage scaling, to minimize power consumption. It also features advanced security capabilities, including secure boot, flash encryption, and WPA2 encryption for Wi-Fi. The ESP32 operates at frequencies ranging from 80 MHz to 240 MHz, with a recommended supply voltage of 3.3V and an operating temperature range of -40°C to 125°C. It is widely used in IoT projects, robotics, automation, and wireless communication applications due to its versatility, low cost, and ease of use.

## 8\_RELAY MODULE :

- 8 individual relays, each with a separate input and output

- Typically uses electromechanical relays (EMRs) or solid-state relays (SSRs)

- Input voltage: usually 5V or 12V DC

- Output voltage: depends on the relay type, but often 10A or 15A at 250V AC or 30V DC

- Control inputs: usually digital (0V or 5V) or analog (0-5V)

- Can be controlled using microcontrollers, Arduino, Raspberry Pi, or other digital devices

- Often includes status LEDs for each relay to indicate whether it is on or off

- May include additional features like overcurrent protection, thermal protection, or surge

Protection

**The 8 relay module is useful for controlling multiple devices, such as:**

- Lights

- Fans

- Motors

- Heaters

- Alarms

- Solenoids

- Valves

**It is commonly used in projects like:**

- Home automation systems

- Industrial control systems

- Robotics and mechatronics

- IoT devices

- Automotive systems

**SOFT WARE USED:**

## AURDINO IDE

The Arduino IDE is a software application that allows users to write, compile, and upload code to Arduino boards. It is a free, open-source platform that provides a comprehensive development environment for programming microcontrollers.

**Features:**

Code Editor: A simple and intuitive code editor with syntax highlighting, code completion, and debugging tools.

Project Management: Organize and manage projects with ease, including creating and managing sketches, libraries, and examples

Code Compilation: Compile code with a single click, and upload it to the Arduino board.

Serial Monitor : Monitor and debug serial communication between the Arduino board and the computer.

Library Management: Easily install and manage libraries for various sensors, actuators, and modules.

Examples and Tutorials: Access a vast collection of example sketches and tutorials to help get started with

ardino programming.

Board and Port Management: Select and configure the Arduino board and serial port for uploading and communication.

The Arduino IDE is available for Windows, macOS, and Linux operating systems, and is compatible with a wide range of Arduino boards and devices.

**AMZON ALEXA APP:**

"Control and manage your Alexa-enabled devices, access skills and features, and interact with Alexa on-the-go with the Amazon Alexa app. Discover and enable new skills, customize your Alexa experience, and receive notifications and updates from your devices. Play music, podcasts, and audiobooks, and access news, weather, and more. Available for Android and iOS devices."

**The Amazon Alexa app is a companion app for Alexa-enabled devices, allowing users to:**

Control smart home devices

Discover and enable new skills

Access settings and preferences

Play music and media

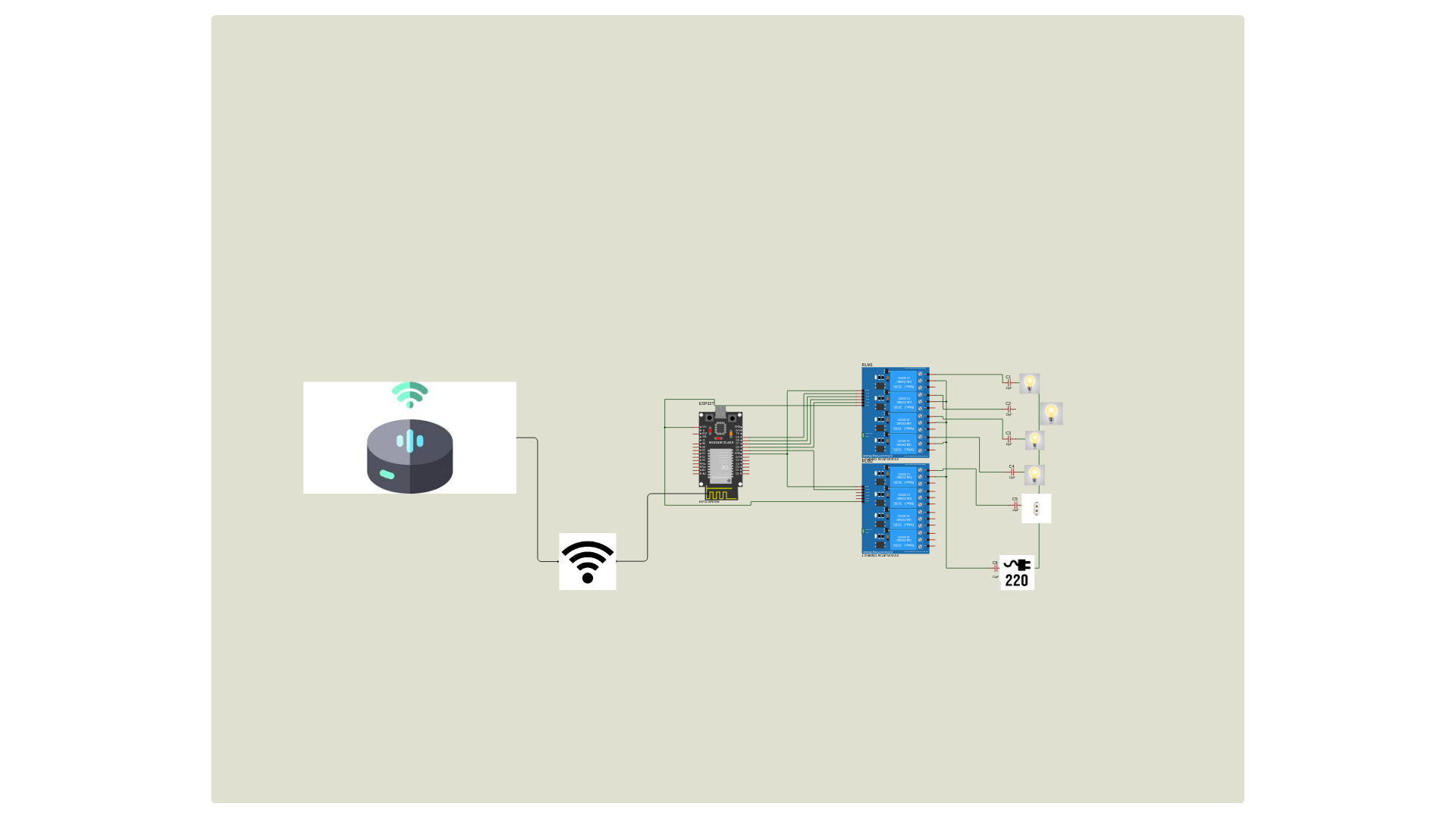
Receive notifications and updates

Set reminders and timers

Access shopping list and order history

Available for Android and iOS devices

**Connections :**

****

1. WiFi connection to the network "gp4" with password "hunza123".
2. Alexa voice control connection using the Espalexa library.

**RELAY CONNECTIONS:**

1. Five devices (relays) connected to GPIO pins

- RelayPin1 (D27) controlling "Room 1" .

- RelayPin2 (D26) controlling "Room 2" .

- RelayPin3 (D25) controlling "Room 3" .

- RelayPin4 (D33) controlling "Room 4" .

- RelayPin5 (D32) controlling "Charging Switch".

4. An LED connected to pin D2 (wifiLed) indicating the WiFi connection status.

* **CODE :**
* #include <WiFi.h>
* #include <Espalexa.h>
* Espalexa espalexa;

#define RelayPin1 27 // D27

#define RelayPin2 26 // D26

#define RelayPin3 25 // D25

#define RelayPin4 33 // D33

#define RelayPin5 32 // D32

* // WiFi Credentials
* const char\* ssid = "gp4";
* const char\* password = "hunza123";
* // Device names
* String Device\_1\_Name = "Room 1";
* String Device\_2\_Name = "Room 2";
* String Device\_3\_Name = "Room 3";
* String Device\_4\_Name = "Room 4";
* String Device\_5\_Name = "Charging Switch";
* // WiFi LED pin
* #define wifiLed 2 // D2
* boolean wifiConnected = false;
* // Callback functions for device control
* void firstLightChanged(uint8\_t brightness) {
* // Control the device connected to RelayPin1
* if (brightness == 255) {
* digitalWrite(RelayPin1, LOW);
* Serial.println("Bedroom Lamp ON");
* } else {
* digitalWrite(RelayPin1, HIGH);
* Serial.println("Bedroom Lamp OFF");
* }
* }
* void secondLightChanged(uint8\_t brightness) {
* // Control the device connected to RelayPin2
* if (brightness == 255) {
* digitalWrite(RelayPin2, LOW);
* Serial.println("Desk Lamp ON");
* } else {
* digitalWrite(RelayPin2, HIGH);
* Serial.println("Desk Lamp OFF");
* }
* }
* void thirdLightChanged(uint8\_t brightness) {
* // Control the device connected to RelayPin3
* if (brightness == 255) {
* digitalWrite(RelayPin3, LOW);
* Serial.println("Ceiling Light ON");
* } else {
* digitalWrite(RelayPin3, HIGH);
* Serial.println("Ceiling Light OFF");
* }
* }
* void fourthLightChanged(uint8\_t brightness) {
* // Control the device connected to RelayPin4
* if (brightness == 255) {
* digitalWrite(RelayPin4, LOW);
* Serial.println("Floor Lamp ON");
* } else {
* digitalWrite(RelayPin4, HIGH);
* Serial.println("Floor Lamp OFF");
* }
* }
* void fifthLightChanged(uint8\_t brightness) {
* // Control the device connected to RelayPin5
* if (brightness == 255) {
* digitalWrite(RelayPin5, LOW);
* Serial.println("Charging Switch ON");
* } else {
* digitalWrite(RelayPin5, HIGH);
* Serial.println("Charging Switch OFF");
* }
* }
* boolean connectWifi() {
* boolean state = true;
* int i = 0;
* WiFi.mode(WIFI\_STA);
* WiFi.begin(ssid, password);
* Serial.println("");
* Serial.println("Connecting to WiFi");
* // Wait for connection
* Serial.print("Connecting...");
* while (WiFi.status() != WL\_CONNECTED) {
* delay(500);
* Serial.print(".");
* if (i > 20) {
* state = false;
* break;
* }
* i++;
* }
* Serial.println("");
* if (state) {
* Serial.print("Connected to ");
* Serial.println(ssid);
* Serial.print("IP address: ");
* Serial.println(WiFi.localIP());
* } else {
* Serial.println("Connection failed.");
* }
* return state;
* }
* void addDevices() {
* espalexa.addDevice(Device\_1\_Name, firstLightChanged);
* espalexa.addDevice(Device\_2\_Name, secondLightChanged);
* espalexa.addDevice(Device\_3\_Name, thirdLightChanged);
* espalexa.addDevice(Device\_4\_Name, fourthLightChanged);
* espalexa.addDevice(Device\_5\_Name, fifthLightChanged); // Adding the fifth device
* espalexa.begin();
* }
* void setup() {
* Serial.begin(115200);
* pinMode(RelayPin1, OUTPUT);
* pinMode(RelayPin2, OUTPUT);
* pinMode(RelayPin3, OUTPUT);
* pinMode(RelayPin4, OUTPUT);
* pinMode(RelayPin5, OUTPUT); // Initialize the new relay pin
* pinMode(wifiLed, OUTPUT);
* digitalWrite(RelayPin1, HIGH);
* digitalWrite(RelayPin2, HIGH);
* digitalWrite(RelayPin3, HIGH);
* digitalWrite(RelayPin4, HIGH);
* digitalWrite(RelayPin5, HIGH); // Ensure all relays are initially off
* wifiConnected = connectWifi();
* if (wifiConnected) {
* addDevices();
* } else {
* Serial.println("Cannot connect to WiFi. So in Manual Mode");
* delay(1000);
* }
* }
* void loop() {
* if (WiFi.status() != WL\_CONNECTED) {
* digitalWrite(wifiLed, LOW);
* } else {
* digitalWrite(wifiLed, HIGH);
* if (wifiConnected) {
* espalexa.loop();
* } else {
* wifiConnected = connectWifi();
* if (wifiConnected) {

addDevices();

* }
* }
* }

# PROJECT HARWARE IMPLEMENTED DIAGRAM

# INSTRUCTIONS:

ALEXA TRUN ON ALL LIGHTS

ALEXA TRUN OFF ALL LIGHTS

ALEXA TURN ON ROOM DEVICE 1

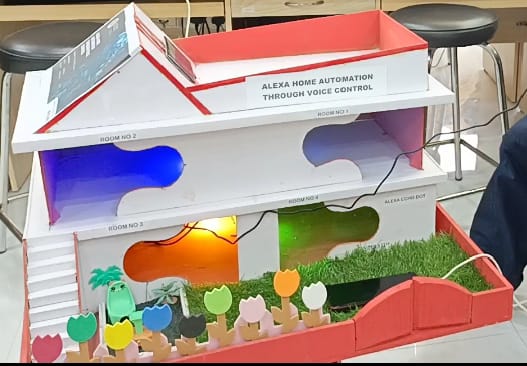
ALEXA TURN ON ROOM DEVICE 2

ALEXA TURN ON ROOM DEVICE 3

ALEXA TURN ON ROOM DEVICE 4

ALEXA TURN ON THE SWITCH

# HARDWARE PROTOTYPE :



# RESULTS

# Only allows for turning lights on and off

All lights turn on and off simultaneously, AND individual control

Only allows for controlling switches, And also individual devices connected to those switches

This means that the current implementation is a basic on/off control, without more advanced features like:

Individual light control (e.g., turning on/off specific lights)

Control of devices connected to switches (e.g., turning on/off a specific appliance)

# CONCLUSION:

Successful implementation of voice-controlled home automation using Amazon Alexa, enabling convenient and hands-free control of various smart devices, including lights, thermostats, security cameras, and more. This project demonstrates the potential of voice assistants in enhancing home automation and improving overall living experiences. With Alexa's intuitive voice commands, users can effortlessly manage and monitor their smart home devices, creating a seamless and connected living space.

Some possible future developments or extensions of the project could include:

- Integrating additional smart devices and sensors

- Implementing advanced automation scenarios and routines

- Exploring AI-powered automation and machine learning capabilities

- Enhancing security and privacy features

- Expanding to other voice assistants or smart home platforms

The project showcases the possibilities of voice-controlled home automation, making it easier for people to control and interact with their living spaces.