VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VOICE BASED HOME AUTOMATION SYSTEM

Under the guidance of Dr Pradnya Ghare

TEAM MEMBERS:

Jude Melath [BT22ECE069]

Barmavathu Narendra Nayak [BT22ECE070]

Chandragiri Balaji [BT22ECE071]

Akhilesh Navalkar [BT22ECE072]

Niharika Desineedi [BT22ECE073]

Roshan Sabkhor [BT22ECE074]

Shreya Ladda [BT22ECE075]

INTRODUCTION

In the evolving landscape of smart home technology, accessibility and convenience remain paramount. Our project introduces a voice-activated home automation system that transforms traditional homes into smart homes by allowing voice control over home appliances. This system is designed to be accessible and user-friendly, catering especially to the needs of differently abled and elderly users, and anyone looking to enhance their living environment with intelligent, automated solutions. This promotes independence and ease of use. To achieve this simply, Arduino Bluetooth is used to communicate with the circuit controlling the appliances. The controller used for the automation in this project is the NodeMCU. When a command is given by the user via mobile, the NodeMCU receives the information and decides the appropriate switching action for the electrical appliances connected to it through relays. The NodeMCU is pre-programmed to connect with the Wi-Fi network and then execute the commands to perform tasks as per the instructions in the code, such as turning on or off the lights or fans.

OVERVIEW

The circuit uses an ESP32, a popular microcontroller with a Bluetooth Module integrated into it, managed via the BluetoothSerial library. For a project like this, a key component is the user interface that allows for remote control. This is typically managed through a mobile app, commonly referred to as an "Arduino Bluetooth Control" app, which acts as the bridge between the user's voice commands or manual input and the hardware controlling home appliances via Bluetooth.

The relays serve as electrically operated switches. Each relay would typically have an input connected to a specific GPIO on the ESP32, and depending on the command received ("on" or "off"), the ESP32 sets its GPIO pin HIGH or LOW, activating or deactivating the relay, and consequently turning the connected appliance on or off.

In short, the ESP32 waits for commands over Bluetooth. When a command is received (SerialBT.readString()), it's checked against predefined strings like "light on", "light off", "fan on", etc. Based on the command, the ESP32 sends a HIGH or LOW signal to the respective pin, controlling the state of the relay and thus the appliance.

MAJOR COMPONENTS USED IN THE CIRCUIT

Relays

It's common to use relays in such setups where microcontrollers control household appliances, due to the electrical requirements of such appliances which typically exceed the output capabilities of microcontroller GPIO pins. The relays serve as electrically operated switches. A relay consists of a coil, an armature and contacts. When you apply voltage to the relay's coil, the electromagnetic field generated pulls a lever (or similar mechanism), causing it to move. If connected to the NO contact, this movement closes the circuit by connecting the common to the NO, allowing current to pass through when activated. If connected to the NC contact, the same movement opens the circuit by disconnecting the common from the NC, stopping the current flow when activated.

In the context of the described home automation project, they play a crucial role in interfacing the low-voltage electronics of the NodeMCU or an Arduino-based system with higher-voltage appliances like lights and fans. Each relay is connected to specific GPIO pins on the microcontroller (ESP32). The appliances are connected to the its Common and NO contacts, assuming the default state for the appliance control is 'off.'

The relays are controlled by software running on the NodeMCU, which interprets commands received via Bluetooth. For instance, when a user sends a "light on" command, the corresponding GPIO pin is set HIGH, energizing the relay's coil, closing the NO contact, and turning on the light.

ESP32

The ESP32 is a highly versatile and powerful microcontroller developed by Espressif Systems. It stands out in the world of IoT and embedded systems due to its comprehensive set of capabilities, including Wi-Fi and Bluetooth connectivity and ample GPIO pins. It features several power modes that are suitable for battery-powered and power-sensitive applications.

In the context of our project, the ESP32 acts as the central computing unit that processes commands received via Bluetooth and controls the operation of external devices (lights, fans) through its GPIO pins. The ESP32's built-in Bluetooth capabilities are utilized to receive commands from a smartphone or other Bluetooth-enabled devices. This feature is vital for allowing remote control without needing physical access to the switches or the appliances. GPIO pins on the ESP32 are configured to interface with relays, which in turn control the power to the household appliances (light and fan in this case).

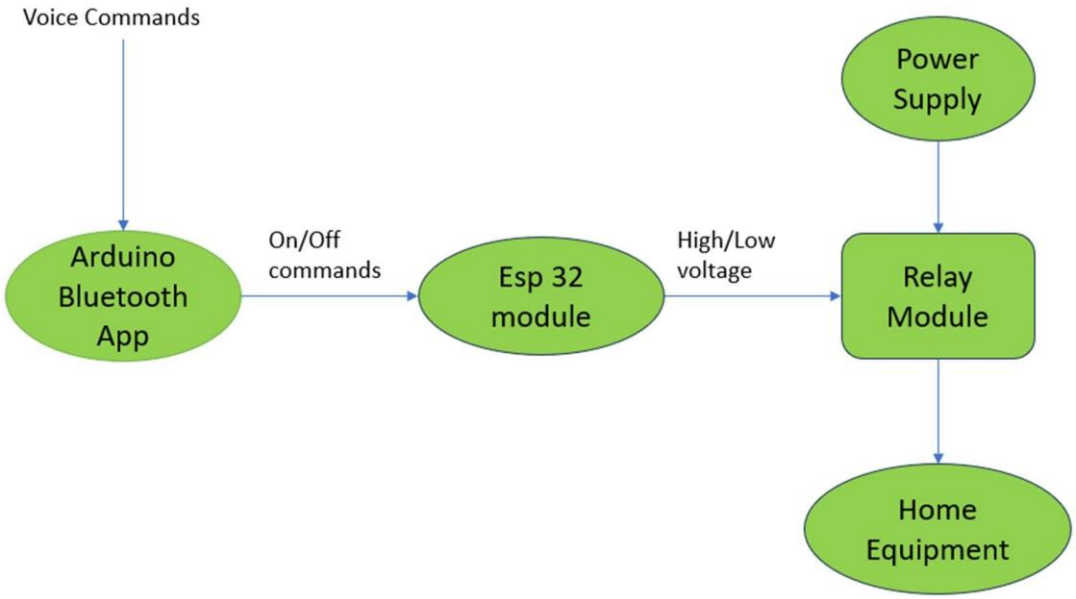
In the Bluetooth-controlled home automation project, the ESP32 enables seamless wireless control over home appliances, enhancing convenience and functionality. Its ability to handle multiple tasks simultaneously and communicate wirelessly makes it an excellent choice for integrating various home automation features under a unified system, thereby increasing the home's intelligence and user interactivity. This capability aligns perfectly with the needs of a modern smart home setup, where convenience, efficiency, and user experience are paramount.

Arduino Bluetooth Control App

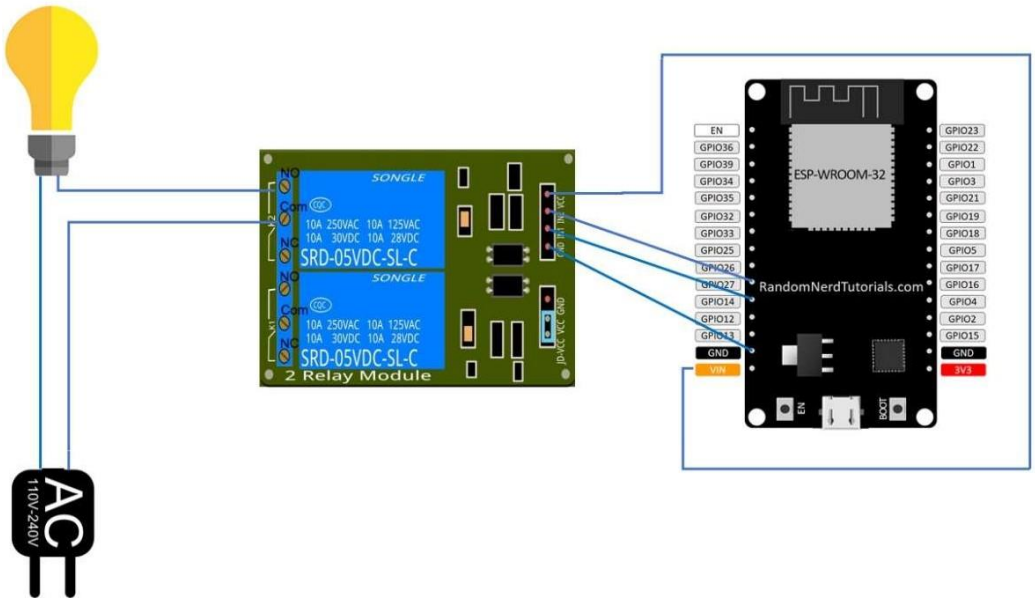
The Arduino Bluetooth Control App serves as the user interface that connects a mobile device to the microcontroller via Bluetooth, allowing users to send commands easily and monitor the status of their projects. It provides a simple, intuitive interface with buttons, sliders, or other input methods by which users can interact to send specific commands. With more advancements, the app has integrated voice recognition technology, allowing users to control devices through voice commands. This is especially useful in accessibility contexts, enhancing usability for users with mobility or vision impairments.

In our home automation project, the Arduino Bluetooth Control App plays a crucial role in interfacing between the user and the system. The app sends direct commands to the ESP32 via Bluetooth. Each command corresponds to an action defined in the ESP32's programming, such as "light on," "fan off," or "on all". The ESP32 receives these commands, interprets them, and controls the relays connected to the appliances accordingly. This eliminates the need for physical switches and is particularly beneficial for the elderly or differently-abled.

BLOCK DIAGRAM



CIRCUIT DIAGRAM:



CONNECTIONS

GPIO pins to relays

The LED light is controlled through a relay connected to GPIO pin 14 on the ESP32. This pin acts as the control pin, outputting a HIGH signal to activate the relay and turn the light on, or a LOW signal to deactivate the relay and turn the light off.

Similarly, the fan is controlled through a relay connected to GPIO pin 27. The ESP32 sends HIGH or LOW signals to this pin to turn the fan on or off, respectively.

Relay Wiring

Each relay will typically have a common (COM) terminal that connects to the power supply line. The NO terminal on each relay connects to the respective appliance (LED light or fan). The NO terminal is used because it ensures that the appliances remain off when the system is powered but inactive (relay not energized).

The high-power source needed for the fan and the LED light connects to the common terminals of their respective relays. The other side of each appliance is then connected back to the neutral power line or ground, completing the circuit.

Bluetooth setup

The ESP32's built-in Bluetooth is initialized with the name "itg219", which allows it to be discoverable and pairable with other Bluetooth-enabled devices like smartphones. The code handles the pairing and communication to receive commands wirelessly.

ARDUINO CODE:

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int lightpin = 27; // define LED pin number
int fanpin=14;
void setup() {
 Serial.begin(115200);
 SerialBT.begin("itg219"); //Bluetooth device name
 Serial.println("The device started, now you can pair it with bluetooth!");
 pinMode(lightpin, OUTPUT);
 pinMode(fanpin, OUTPUT); // set LED pin as an output
}
void loop(){
 if(SerialBT.available()) {
   String c=SerialBT.readString();
   Serial.println(SerialBT.readString());
   if(c=="light on"){
     digitalWrite(lightpin,HIGH);
   }
   if(c=="light off"){
     digitalWrite(lightpin,LOW);
   }
   if(c=="fan on"){
    digitalWrite(fanpin,HIGH);
   }
   if(c=="fan off"){
    digitalWrite(fanpin,LOW);
   }
   if(c=="on all"){
    digitalWrite(lightpin,HIGH);
    digitalWrite(fanpin,HIGH);
   }
   if(c=="kill all"){
    digitalWrite(lightpin,LOW);
    digitalWrite(fanpin,LOW);
   }}
 delay(2000);
}
```

CIRCUIT OPERATION

When a command such as "light on" is received via Bluetooth, the ESP32's GPIO pin 14 goes HIGH, energizing the relay connected to the LED light, which closes the NO contact, allowing current to flow and the light to turn on. Similarly, a command like "fan off" will cause GPIO pin 27 to go LOW, de-energizing the relayconnected to the fan, which opens the NO contact and stops the current, turning the fan off.

APPLICATIONS

The project described, which integrates ESP32 with Bluetooth for controlling home appliances such as LED lights and fans, offers a wide range of practical applications across various settings.

Support for Differently-abled Individuals: By enabling voice or app-based controls, this system can significantly improve the quality of life for people with mobility restrictions or other disabilities, allowing them to control their environment effortlessly.

Elderly Care: Older adults who may struggle with manual controls can benefit from easy-to-use voice commands or simple app interfaces to manage lighting, temperature, and other home functions.

Efficient Use of Resources: By automating the control of lights and fans, users can ensure that these devices are only on when needed, thereby saving energy and reducing utility bills.

Voice Control: Integration with popular voice assistants like Amazon Alexa or Google Assistant can further enhance the user experience, making it even more convenient to manage home environments.

CONCLUSION

The Bluetooth-controlled home automation project using the ESP32 microcontroller represents a significant advancement in home management technology. By integrating Bluetooth for wireless communication, this system allows users to effortlessly control lights and fans from their smartphones or through voice commands. This capability enhances convenience, reduces energy consumption, and improves home security by allowing remote operation of household appliances.

Such features cater especially to the needs of elderly and differently abled individuals by minimizing the physical effort required to manage everyday tasks, thus fostering greater independence and comfort. For the elderly and those with physical limitations, the ability to operate home appliances remotely can significantly reduce the risk of accidents and improve their ability to live independently. The system's straightforward user interface and voice command functionality make it accessible and easy to use, which is crucial for enhancing the quality of life for geriatric users and those with mobility or visual impairments.

Overall, this home automation project not only increases the convenience and efficiency of home environments but also plays a vital role in supporting the autonomy of differently abled and elderly individuals. It demonstrates the impactful benefits of integrating simple technological solutions into everyday living, ensuring safety, energy efficiency, and improved accessibility in modern homes.