Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Project Report

On

"Voice Controlled Home Automation"

[Code No: 306]

(For partial fulfillment of Third Year/ Second Semester in Computer Science & Engineering)

Submitted by:

Yural Pokhrel (Roll No. 37)

Shreela Sapkota (Roll No. 44)

Aawishkar Tiwari (Roll No. 59)

Submitted to

Prof. Sudan Jha, PhD

Department of Computer Science and Engineering

Submission Date: 21/06/2024

Abstract

This project explores the development and implementation of a home automation system utilizing Arduino and Bluetooth technology, enabling control of household appliances via voice commands. The primary objective is to create a convenient and efficient method for remotely managing devices such as lights, fans, and night lamps through a mobile application. By integrating voice recognition with wireless communication, the system offers a user-friendly interface for seamless operation.

The Arduino microcontroller receives voice commands through the Bluetooth module, processes these commands, and actuates corresponding relays to control the connected appliances. The project demonstrates the application of microcontroller programming and wireless communication to achieve practical home automation solutions. This innovative approach enhances user convenience and provides a foundation for future advancements in smart home technology, illustrating the potential of integrating IoT principles into everyday life.

Table of Contents

Chapter 1 Introduction

Home automation is increasingly popular, offering enhanced convenience, energy efficiency, and security. This project aims to design and implement a home automation system using Arduino and Bluetooth technology, enabling users to control household appliances through voice commands.

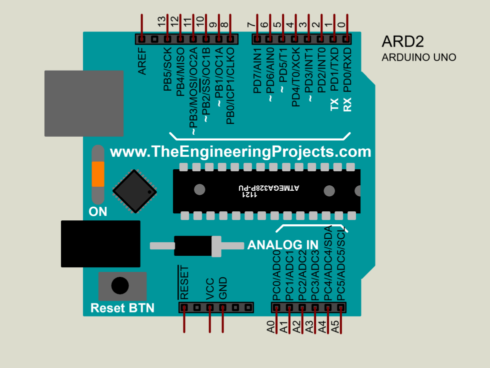
Leveraging an Arduino Uno microcontroller and a Bluetooth HC-05 module, the system wirelessly communicates with a mobile application that captures voice commands. These commands are transmitted to the Arduino, which processes them to control relays managing appliances like lights, fans, and night lamps.Inspired by the Internet of Things (IoT), this project provides a user-friendly, hands-free method for home automation. It involves voice recognition for natural interaction, Bluetooth for wireless communication, microcontroller programming for command processing, and relays for appliance control.

This project demonstrates the integration of simple electronics and modern communication technologies to create an efficient home automation solution, showcasing the potential for future smart home advancements.

Chapter 2 Components Breakdown

Components Used:

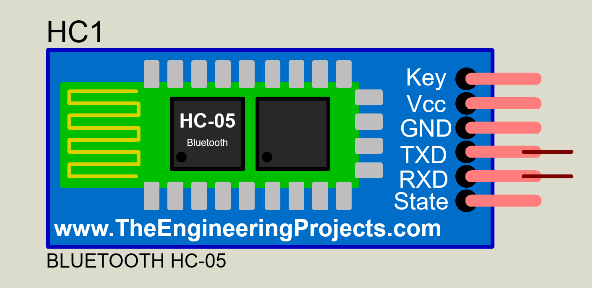
1. Arduino Uno
   Description: The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button.
   Function: Acts as the brain of the system, processing voice commands received from the Bluetooth module and controlling the relays to manage appliances.



2. Bluetooth Module (HC-05)

   Description: The HC-05 is a Bluetooth module designed for wireless communication. It can be used in a Master or Slave configuration, making it a versatile solution for short-range wireless communication.

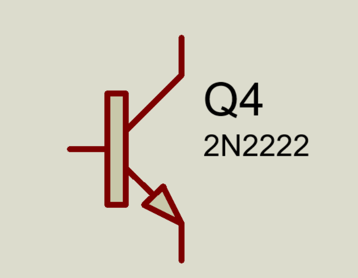
   Function: Receives voice commands from the mobile application and sends them to the Arduino via serial communication.



3. Transistors (2N2222)

   Description: The 2N2222 is a common NPN bipolar junction transistor used for general-purpose low-power amplifying or switching applications.

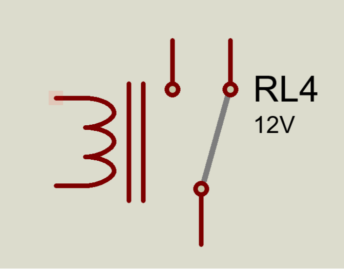
   Function: Acts as a switch to control the relay coils, enabling or disabling the connected appliances.



4. Relays (12V)

   Description: An electromagnetic switch that is used to control high-power devices with a low-power signal. It has a coil that, when energized, pulls a contact to close the circuit.

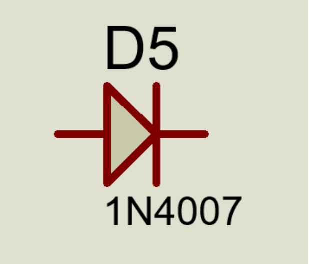
   Function: Controls the high-power appliances (lights, fans, etc.) based on signals from the Arduino.

5. Diodes (1N4007)

   Description: The 1N4007 is a general-purpose silicon rectifier diode. It is used to protect electronic circuits from voltage spikes.

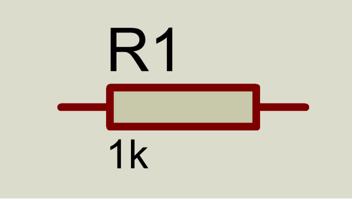
   Function: Acts as a flyback diode to protect the transistor from voltage spikes generated by the relay coils.

6. Resistors (1kΩ)

   Description: Fixed resistors with a resistance of 1kΩ, used to limit the current flowing through a circuit.

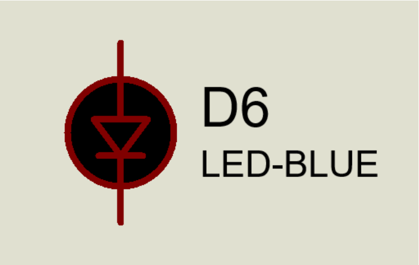
   Function: Controls the base current of the transistors to ensure proper switching.

7. LED (Blue)

   Description: A light-emitting diode that emits blue light when a voltage is applied.

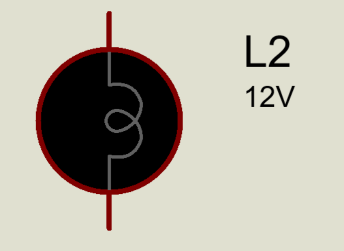
Function: Used as an indicator for the status of the night lamp.



8. Lamp (12V)

Description: A standard 12V lamp used to provide illumination.

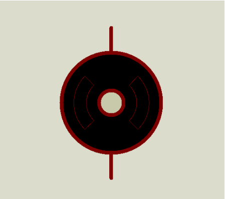
Function: Represents one of the household appliances being controlled.



9. Fan (12V)

Description: A 12V fan used for ventilation.

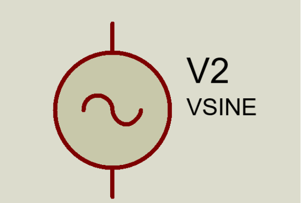
Function: Represents another household appliance being controlled.



10. VSINE (AC Source)

Description: Represents an AC power source for the lamp.

Function: Supplies AC power to the lamp when the relay is activated.



Communication Between Components:

1. Voice Commands to Bluetooth Module:
   Process: The user speaks a command into the mobile application.
   Communication: The mobile application converts the voice command into a text string and sends it via Bluetooth to the HC-05 module.
2. Bluetooth Module to Arduino:
   Process: The HC-05 receives the text string from the mobile application.
   Communication: The HC-05 module sends the text string to the Arduino Uno via serial communication (TXD of HC-05 to RXD of Arduino, and RXD of HC-05 to TXD of Arduino).
3. Arduino Processing:
   Process: The Arduino reads the incoming text string from the serial port.
   Communication: The Arduino processes the string to determine which command was received (e.g., "light on", "fan off").
4. Arduino to Transistors:
   Process: Based on the command, the Arduino sets the corresponding digital output pin HIGH or LOW.
   Communication: The output pin sends a signal to the base of the corresponding 2N2222 transistor through a 1kΩ resistor.
5. Transistors to Relays:
   Process: When the base of the transistor is activated, it allows current to flow from the collector to the emitter.
   Communication: This energizes the relay coil, closing the relay switch and allowing current to flow to the connected appliance.
6. Relays to Appliances:
   Process: The closed relay switch completes the circuit for the appliance.
   Communication: The appliance (lamp, fan, etc.) receives power and turns on or off based on the state of the relay.
7. Diodes Protection:
   Process: When the relay coil is de-energized, it generates a voltage spike.

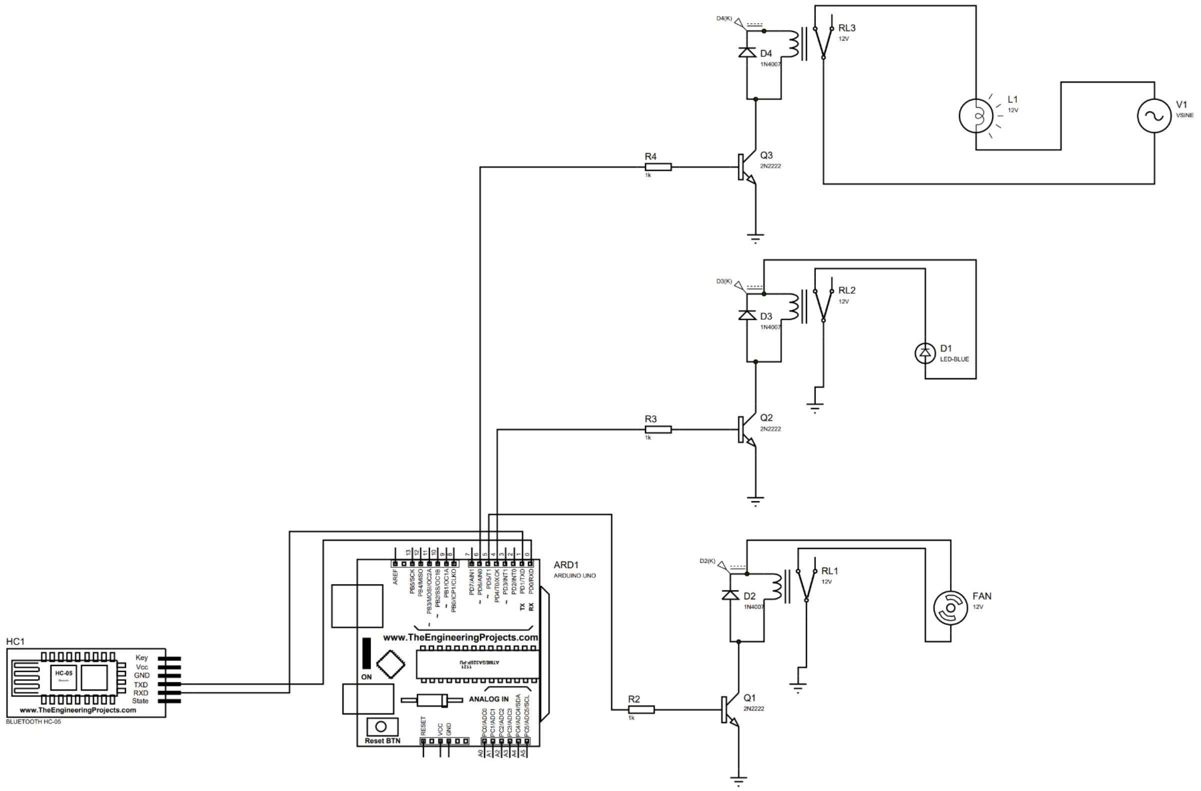
Communication: The 1N4007 diode across the relay coil provides a path for the spike, protecting the transistor from damage.

Chapter 3 Simulation Details

Procedures:

1. Setup Proteus:
    - Add components: Arduino Uno, Bluetooth HC-05, 2N2222 transistors, 1kΩ resistors, 1N4007 diodes, relays, and appliances (lamp, fan, LED).
    - Configure the circuits.
2. Configuring Components:
    - Set properties for each component (e.g., relay voltage, resistor values).
    - Configure the Bluetooth module for serial communication.
3. Writing and Uploading Arduino Code:
    - Write the Arduino code to process voice commands and control the relays.
    - Compile and upload the code to the Arduino with Proteus.
4. Running the Simulation:
    - Start the simulation in Proteus.
    - Use the virtual terminal to send voice command strings to the Arduino.
    - Observe the relay and appliance behavior in response to commands.
5. Observing Results:
    - Verify appliances turn on/off according to commands.

Circuit Diagram:



Arduino Code:

```
String voice;

void setup() {

Serial.begin(9600);

pinMode(6, OUTPUT);

pinMode(5, OUTPUT);

pinMode(4, OUTPUT);
```

```
}

void loop() {

while(Serial.available()){

  delay(3);

  char c = Serial.read();

  voice+=c;}

if(voice.length() >0){

  Serial.println(voice);

  if(voice == "light on")

  {digitalWrite(6, HIGH);}

  else if(voice == "light off")

  {digitalWrite(6, LOW);}

  else if(voice == "fan on")

  {digitalWrite(5, HIGH);}

  else if(voice == "fan off")

  {digitalWrite(5, LOW);}

  else if(voice == "night lamp on")
  {digitalWrite(4, HIGH);}

  else if(voice == "night lamp off")

  {digitalWrite(4, LOW);}

  else if(voice == "all on")

  {digitalWrite(4, HIGH);

    digitalWrite(5, HIGH);

  digitalWrite(6, HIGH);}

  else if(voice == "all off")

  {digitalWrite(4, LOW);

    digitalWrite(5, LOW);

  digitalWrite(6, LOW);}

  voice = "";}

}
```

Chapter 4 Testing

Following are the tested commands:

Commands	Expected Result	Observed Result
“Light on”	Lamp turns on	Lamp turned on
“Light off”	Lamp turns off	Lamp turned off
“Fan on”	Fan turns on	Fan turned on
“Fan off”	Fan turns off	Fan turned off
“Night light on”	LED turns on	LED turned on
“Night light off”	LED turns off	LED turned off
“All on”	All appliances turns on	All appliances turns on
“All off”	All appliances turned off	All appliances turned off
“Invalid”	No Actions	No Actions

Chapter 5 Results and Analysis

The simulation of the home automation system in Proteus showed it worked well. It accurately responded to voice commands sent via Bluetooth, controlling appliances like lamps and fans based on commands like "turn on the light" or "turn off the fan". The system also effectively managed high-power devices using relays controlled by transistors, which were protected by diodes from voltage spikes. It could handle commands that affected all appliances at once, and it ignored commands it didn't recognize to avoid mistakes.

The components like Arduino Uno and HC-05 Bluetooth module proved reliable for real-world use. They managed data well and protected against electrical issues. The system can be expanded easily to control more appliances, and using voice commands made it user-friendly. It responded quickly and accurately to commands, ensuring smooth operation. Future improvements could focus on making commands even more flexible, adding ways for users to see appliance statuses, and enhancing security to prevent unauthorized access.

Chapter 6 Conclusion

The simulation results demonstrate the viability and effectiveness of the home automation system using Arduino and Bluetooth. The system successfully processed and executed voice commands, controlled multiple appliances, and exhibited robustness against invalid inputs. These outcomes validate the system's design and highlight its potential for real-world home automation applications. Further enhancements could improve functionality and user experience, making the system even more robust and user-friendly.

Chapter 7 Contributions

Yural Pokhrel:
Handled the coding of the Arduino microcontroller.
Developed the circuit logic for the home automation system.

Aawishkar Tiwari:
Designed and developed the mobile application used to send voice commands.
Established communication through the Bluetooth Module (HC-05), utilizing the COM 3 port for Bluetooth connectivity.

Shreela Sapkota:
Designed the circuit in Proteus.
Created the final design in Proteus using the Bluetooth module and coded Arduino.