University of Asia Pacific

Department of Computer Science & Engineering

Course Title: Microprocessors & Microcontrollers Lab

Course Code: CSE 316

Lab Report

Experiment Number: 06

Experiment Title: Arduino-Based IR Remote Controlled AC Bulb

Submitted by:

Name: Sajid Shahan Rahman

Student ID: 22201186

Section: D-1

Submitted to:

Zaima Sartaj Taheri

Lecturer

Department of Computer Science and Engineering

Date of Submission: 07/10/2025

1. Experiment Name

Arduino-Based IR Remote Controlled AC Bulb.

2. Objective

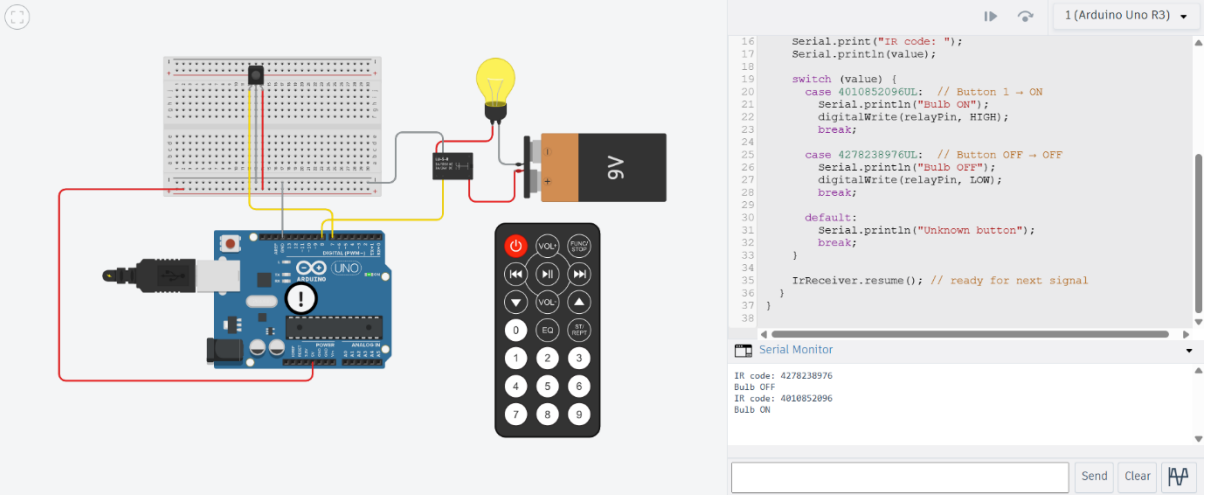
The objective of this project is to design and implement a remote-controlled AC bulb system that integrates convenience and automation. An IR remote is used as the primary input to wirelessly control the switching of an AC bulb via an Arduino, allowing the user to turn the bulb ON or OFF from a distance. A relay module or suitable switching mechanism is employed to safely handle the AC load, ensuring reliable operation. This setup aims to provide a low-cost, user-friendly, and efficient solution for automated home lighting control, combining wireless convenience with safe electrical switching.

3. Apparatus / Hardware & Software Requirements

List of all required tools and components:

- Microcontroller (Arduino UNO)
- Software (Arduino IDE)
- Breadboard
- Jumper Wires
- Power Source (9V Battery)
- AC Bulb (1 piece)
- IR Remote
- IR Sensor (1 piece)
- Relay SPDT (Single Pole Double Throw)

4. Circuit Diagram / Schematic



Tinkercad Simulation

5. Code / Assembly Program

```
#include <IRremote.hpp>

const int relayPin = 8;
const int rcvPin = 7;

void setup() {
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW); // relay off initially
  Serial.begin(9600);
  IrReceiver.begin(rcvPin, ENABLE_LED_FEEDBACK);
}

void loop() {
  if (IrReceiver.decode()) {
    unsigned long value = IrReceiver.decodedIRData.decodedRawData;
    Serial.print("IR code: ");
```

```
    Serial.println(value);

    switch (value) {
      case 4010852096UL:  // Button 1 → ON
        Serial.println("Bulb ON");
        digitalWrite(relayPin, HIGH);
        break;

      case 4278238976UL:  // Button OFF → OFF
        Serial.println("Bulb OFF");
        digitalWrite(relayPin, LOW);
        break;

      default:
        Serial.println("Unknown button");
        break;
    }

    IrReceiver.resume(); // ready for next signal
  }
}
```

6. Output / Observations

In the project, the IR receiver connected to pin 7 successfully detected signals from the remote, which triggered the Arduino to control the relay on pin 8, turning the AC bulb ON or OFF as commanded. In the Tinkercad simulation, all components functioned as expected: the IR receiver correctly recognized the remote signals, and the relay switched the bulb without delay. In the real-life implementation, the relay operated reliably once proper connections were ensured, and the AC bulb responded consistently to the remote commands. Minor adjustments were made to ensure stable power to the relay and correct wiring of the bulb circuit. Overall, the system operated correctly in simulation, while the hardware prototype successfully

achieved the intended objective of remote-controlled AC bulb operation with reliable performance.

7. Result

In this project, the IR receiver connected to pin 7 successfully received signals from the remote, which triggered the Arduino to control the relay on pin 8, switching the AC bulb ON or OFF. During the Tinkercad simulation, all components worked as expected: the IR receiver detected remote signals correctly, and the relay reliably switched the bulb without delay.

During the hardware implementation, a few issues were encountered. Initially, the relay was connected to pin 13, and the bulb only turned ON for a brief moment before switching OFF. This was caused by the onboard LED and resistor on pin 13, which interfered with the relay signal. Switching the control pin to pin 8 resolved this problem, allowing the relay to stay energized while the pin was HIGH. Another minor issue involved ensuring stable connections to the relay coil and proper wiring of the AC bulb; these were fixed by double-checking connections and confirming secure terminals.

After these adjustments, the system operated reliably: the IR remote consistently controlled the AC bulb, the relay switched without flickering, and the hardware prototype performed as intended. Overall, the project successfully demonstrated a low-cost, user-friendly method for wireless AC bulb control with Arduino.

8. Conclusion

This project provided a practical introduction to Arduino-based wireless control of electrical appliances. I learned how to interface an IR receiver, a relay, and an AC bulb with a microcontroller, and gained hands-on experience in controlling devices remotely. The system worked successfully in Tinkercad simulation, demonstrating reliable ON/OFF switching of the bulb through IR remote signals. In the real-life implementation, challenges such as the initial choice of Arduino pin, unstable relay connections, and ensuring proper wiring for the bulb affected performance. These issues were resolved by switching the control pin to a stable digital pin and securing all connections. Overall, the project enhanced my understanding of hardware interfacing, coding logic, and practical applications of automated and remote-controlled home lighting systems.