PROJECT: CONTROLLING LEDS USING AN IR REMOTE AND ARDUINO UNO R3

**Submitted by**

**ANAMARIA CIOCAN – Information Engineering**

**LUCIAN POPESCU – Information Engineering**

**A project report for  
Course: Electronic Devices and Analog Electronics**

Coordinator Teacher

**THUHANG BUI**

**Abstract**

This project showcases the control of multiple LEDs using an Arduino UNO R3 and an IR remote. The system is designed to demonstrate versatility in LED lighting modes and user interaction through remote commands.

The implemented lighting modes include:

1. **Individual LED Control** – Each LED (red, green, white) can be turned on or off individually.
2. **All LEDs On** – All LEDs are illuminated simultaneously at maximum brightness.
3. **Christmas Mode** – LEDs cycle through colors in a festive sequence.
4. **Blink Mode** – LEDs blink on and off with adjustable speed.
5. **Fade Mode** – LEDs fade in and out with a smooth transition and configurable speed.

These modes are activated by specific buttons on the IR remote, identified using their unique HEX codes. The project also utilizes features like PWM (Pulse-Width Modulation) for brightness control and time-based functions for dynamic lighting effects.

The system demonstrates the practical integration of hardware and software, providing an engaging example of Arduino’s capabilities in controlling and enhancing LED functionality.

Table of Contents

# 1. Introduction

1.1. Project Scope and Relevance

1.2. Objective of the Project

# 2. Materials and Tools

2.1. Hardware Components

2.2. Software Environment

# 3. Design and Development

3.1. Circuit Design Overview

3.2. Schematic Representation

3.3. Implementation Workflow

# 4. Results and Analysis

4.1. Hardware Performance

4.2. Software Functionality

4.3. Observations and Analysis

# 5. Lightning Mode And Features

5.1. Individual LED Control Mode

5.2. Simultaneous Illumination Mode

5.3. Christmas Sequence Mode

5.4. Adjustable Blink Mode

5.5. Gradual Fade Mode

# 6. Challenges and Limitations

# 7. Conclusions and Future Improvements

# 8. References

# 1. Introduction

## Project Scope and Relevance

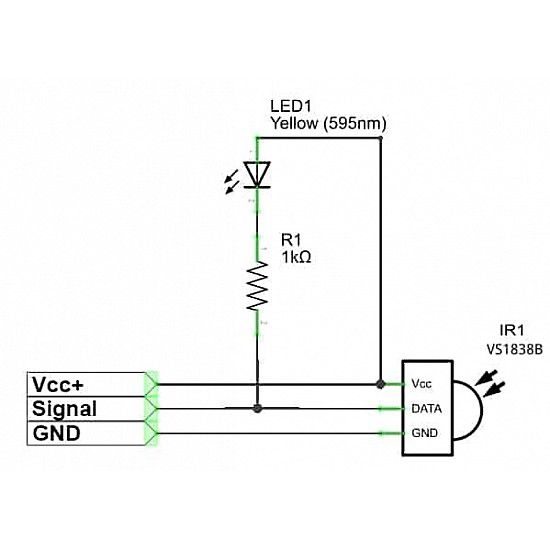
The purpose of this project is to display the application of Arduino UNO R3 in controlling LEDs with multiple dynamic lighting modes. By incorporating various effects, the project not only illustrates basic LED control but also enhances understanding of programmable logic, remote control interfaces, and interactive design. This project is applicable in scenarios where aesthetic lighting or user interaction is required, such as in home automation systems, decorative lighting, and other embedded systems applications.

## 1.2. Objective of the Project

The objective of this project is to control a set of LEDs with an Arduino board and an IR remote control, allowing the user to switch between different lighting modes. These modes include controlling individual LEDs, activating all LEDs simultaneously, and applying lighting effects like blink and fade, with speed control options for the blink and fade modes.

# 2. Materials and Tools

## 2.1. Hardware Components

* **Arduino UNO R3**: The main microcontroller used to control the LEDs(it has 14 digital input/output pins, of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator, an USB connection, a power jack, an ICSP header and reset button)
* **LEDs (Red, White, Green)**: Used to display different lighting effects(with wavelength between 400-630 nm)
* **IR Remote Control**: Used to send commands for controlling the LEDs
* **220 Ω Resistors: Used to limit the current flow throughout the circuit**
* **Jumper wireds**: User for connecting the circuit
* **Breadboard**: For assembling the circuit
* **USB Cable**: To upload code to Arduino and provide power.
* **IR Receiver(VS1838B KY-022)**: Allows the Arduino to receive signals from the IR remote control. It has: voltage supply(vcc): 2.7 - 5.5V, frequency: 38KHz, receiving range: 18m(depending on the strength of the IR signal and environment), current Consumption of ~0.4mA and the following pinout: VCC: Power supply (5V on Arduino); GND:Ground Connection (GND on Arduino); OUT:Digital output pin (choosen pin on Arduino)

## 2.2. Software Environment

* **Arduino IDE**: The programming environment used to write and upload code to the Arduino
* **IRremote Library**: A library used to decode the IR signals sent by the remote control
* **Tinkercad**: Online simulation tool to test the circuit virtually.

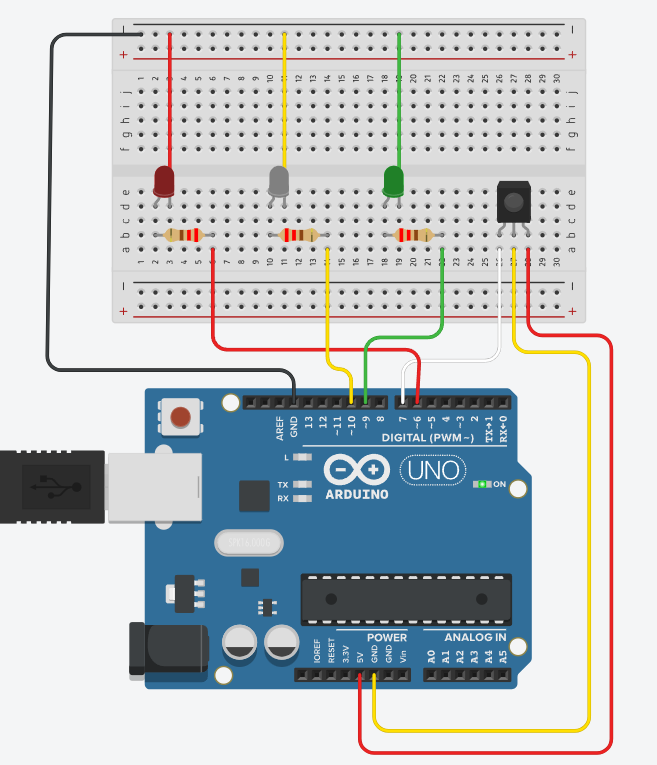
# 3. Design and Development

## 3.1. Circuit Design Overview

## The circuit comprises three LED's connected to an Arduino UNO R3, controlled via digital pins (for obtaining certain results using remote control, we used PWM pins also known as Pulse Width Modulation pins which are a type of digital output pins on Arduino microcontrollers that allow you to simulate analog output by varying the duty cycle of the signal). Resistors are also used to limit the current flow and protect the LED's. The system also includes an IR receiver for remote control functionality.

## 3.2. Schematic Representation

The schematic representation of the circuit is shown below:

1. **Power Supply:**

* The Arduino board provides 5V and GND to the breadboard power rails.

1. **LEDs:**

* Red LED to pin **~6**
* White LED to pin **~10**
* Green LED to pin **~9**

1. **IR Receiver:**

* VCC to Arduino’s +5V
* GND to Arduino’s GND
* Signal pin to pin **7**

**Figure 1**: Circuit diagram showing the connections between the Arduino Uno, LEDs + resistors and IR receiver

## 3.3. Implementation Workflow

1. Assemble the circuit as per the schematic.
2. Install the IRremote library in Arduino IDE.
3. Write the code to implement the lighting modes.
4. Upload the code to the Arduino board.
5. Test the setup using the IR remote

## 3.4. Simulation

The circuit was first tested in Tinkercad for functionality. The potentiometer was adjusted to ensure text visibility.

# 4. RESULTS AND ANALYSIS

## 4.1. Hardware Performance

The hardware performed reliably, with all LEDs responding correctly to the IR remote. The connections were stable, and the resistors effectively limited the current to the LEDs, preventing damage.

## 4.2. Software Functionality

The software accurately decoded signals from the IR remote. Each button press on the remote was logged in the Serial Monitor, displaying:

* The hexadecimal code of the pressed button.
* A corresponding message for each mode (e.g., "Christmas Mode Activated" or "Adjusting Blink Speed").

The use of Serial Monitor provided real-time debugging and confirmed correct mode activation.

## 4.3. Observations and Analysis

1. **Serial Monitor Insights:**

* During testing, the Serial Monitor displayed debug messages and the decoded IR codes, confirming proper communication between the remote and the Arduino.
* Messages like "LED RED ON" or "Blink speed increased to 300ms" allowed us to monitor changes dynamically.

**2. Performance Metrics:**

* The response time from pressing the remote button to the LED action was nearly instantaneous.
* Adjusting parameters like blink speed showed immediate results, confirming software efficiency.

# 5. Lighting Modes and Features

## 5.1 Individual LED Control Mode

* Each LED can be turned on or off independently using specific buttons on the remote as follows: button 1 for the Red LED, button 2 for the White LED, button 3 for the Green LED;

## 5.2 Simultaneous Illumination Mode

* All LEDs light up simultaneously, creating a unified effect pressing button 4 in this case, on the remote;

## 5.3 Christmas Sequence Mode

* The LEDs light up in a predefined sequence resembling Christmas lights(LEDs light up in sequence) using the remote’s button 5;

## 5.4 Adjustable Blink Mode

* LEDs blink at a speed controlled by the user via remote by pressing button 6. The Serial Monitor logs each adjustment with messages like "Blink speed: 500ms”(or the chosen blink speed value);

## 5.5 Gradual Fade Mode

* LEDs gradually increase and decrease in brightness, creating a fade effect, by choosing button 7 on the remote. Adjustments to the fade duration were logged in the Serial Monitor for verification, similarly to the Adjustable Blink Mode;

# Challenges and Limitations

* **Range Limitations:** The IR receiver's effective range was restricted to approximately 5 meters, limiting its usability in larger spaces.
* **Signal Interference:** External IR devices or environmental factors, such as fluorescent lighting, occasionally disrupted signal transmission, reducing reliability.
* **Signal Misreads:** The IR receiver sometimes misinterpreted remote inputs, especially when buttons were pressed rapidly or from extreme angles.
* **Power Constraints:** When multiple LEDs were active simultaneously, their brightness diminished due to the limited power output of the Arduino.
* **Component Durability:** Prolonged high-intensity operation of the LEDs caused heating, potentially impacting their lifespan over time.

# Conclusions and Future Improvements

The project successfully demonstrated multiple lighting modes using Arduino UNO and an IR remote. The Serial Monitor was instrumental in debugging and validating functionality. Future improvements could include:

* Bluetooth-based control for extended range and flexibility.
* Integration of RGB LEDs for more diverse lighting effects.
* A graphical user interface for mode selection.

# References

* 1. Arduino Documentation: <https://www.arduino.cc/>
  2. IRremote Library: <https://github.com/Arduino-IRremote/Arduino-IRremote>
  3. Pulse Width Modulation Documentation: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>
  4. IR Receiver/Remote Tutorial: <https://www.youtube.com/watch?v=CZ_mtpyXEFA>
  5. **Tinkercad:** <https://www.tinkercad.com>