Controlling Three LEDs using an IR Remote with Arduino

1. Overview

This project demonstrates how to control three LEDs using an IR remote and an IR sensor with an Arduino Uno. The IR sensor receives signals from the remote, decodes them, and then controls the LEDs accordingly.

2. Components Used

- Arduino Uno
- Breadboard
- IR Remote
- IR Sensor
- 3 LEDs (Green, Orange, Blue)
- 3 Resistors (220Ω each)
- Jumper Wires

3. Circuit Connections

- **LEDs:**
  - Green LED (Anode) → Digital Pin 2
  - Orange LED (Anode) → Digital Pin 3
  - Blue LED (Anode) → Digital Pin 4
  - Cathodes of all LEDs → 220Ω resistors → Ground
- **IR Sensor:**
  - Middle Pin → Ground (GND)
  - Right Pin → 5V (VCC)
  - Left Pin → Digital Pin 12 (for receiving IR signals)

4. Code Explanation

- The **IRremote** library is used to decode signals from the IR remote.
- Each button on the IR remote corresponds to a unique hexadecimal value.
- The decoded values are used in a switch-case structure to turn LEDs ON or OFF.
- The IRrecv and decode_results classes are used to receive and interpret IR signals.
- The Serial Monitor displays the received IR signal values to identify unique codes for each button.

5. Arduino Code

```
#include <IRremote.h>  // Include the IRremote library (v2.6.0)


#define IR_RECEIVE_PIN 12  // Define the IR receiver pin
```

```
// Define LED pins
#define BLUE_LED 2
#define ORANGE_LED 3
#define GREEN_LED 4

IRrecv irrecv(IR_RECEIVE_PIN);  // Create an IR receiver object
decode_results results;  // Variable to store received IR data

void setup() {
  Serial.begin(9600);
  irrecv.enableIRIn();  // Start the IR receiver

  // Set LED pins as OUTPUT
  pinMode(BLUE_LED, OUTPUT);
  pinMode(ORANGE_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) {  // Check if an IR signal is received
    Serial.println(results.value, HEX);  // Print received value in HEX

    switch (results.value) {
      case 0xFFA25D:  // Button 1
        digitalWrite(BLUE_LED, HIGH);
        break;
      case 0xFF629D:  // Button 2
        digitalWrite(BLUE_LED, LOW);
        break;
      case 0xFFE21D:  // Button 3
        digitalWrite(ORANGE_LED, HIGH);
        break;
      case 0xFF22DD:  // Button 4
        digitalWrite(ORANGE_LED, LOW);
        break;
      case 0xFF02FD:  // Button 5
        digitalWrite(GREEN_LED, HIGH);
        break;
      case 0xFFC23D:  // Button 6
        digitalWrite(GREEN_LED, LOW);
        break;
    }

    irrecv.resume();  // Continue receiving the next IR signal
  }
```

}6. Working Principle

1. The IR sensor detects signals from the IR remote and sends them to Arduino.
2. The Arduino decodes the signal and maps it to specific LED actions.
3. When a button is pressed:
   - Button 1 turns ON the blue LED, and Button 2 turns it OFF.
   - Button 3 turns ON the orange LED, and Button 4 turns it OFF.
   - Button 5 turns ON the green LED, and Button 6 turns it OFF.
4. The serial monitor helps identify unique IR codes for different buttons.

7. Conclusion

This project successfully demonstrates how to interface an IR remote with an Arduino to control multiple LEDs. It can be extended to control other devices like motors or home automation systems.