

# Controlling LEDs With IR Remote

## 1. Introduction

This project demonstrates how to control multiple LEDs using an IR (Infrared) remote and an Arduino board. It showcases wireless control capabilities via remote signals, making it an excellent beginner-level project for learning about IR communication and Arduino programming.

## 2. Key Components

- Arduino Uno board
- IR Receiver (e.g., TSOP1738)
- IR Remote control
- LEDs (3 pieces – Blue, Orange, Green)
- 220Ω Resistors (for LEDs)
- Jumper wires
- Breadboard
- USB Cable (for programming)

## 3. Working Principle

- **Step 1:** An IR remote sends signals encoded as pulses of infrared light when a button is pressed.
- **Step 2:** The IR Receiver connected to the Arduino detects these signals and sends them to the Arduino for decoding.
- **Step 3:** Using the IRremote library, the Arduino decodes the received signal to a unique hexadecimal or decimal code.
- **Step 4:** The Arduino compares the received code against predefined values in a switch-case structure.
- **Step 5:** Depending on which button was pressed, the Arduino turns ON or OFF a specific LED (Blue, Orange, or Green).
- **Step 6:** The system continuously listens for new remote signals and updates LED status accordingly.

#### 4. Circuit Overview

- **IR Receiver:** Connected to Arduino Digital Pin 12 (with VCC to 5V and GND to GND).
- **Blue LED:** Connected to Digital Pin 2 via 220Ω resistor.
- **Orange LED:** Connected to Digital Pin 3 via 220Ω resistor.
- **Green LED:** Connected to Digital Pin 4 via 220Ω resistor.
- **Common Ground:** All LEDs and IR Receiver share Arduino's ground.
- **Power Supply:** Arduino powered through USB or external supply.

#### 5. Code

```
/*Code Written by-
```

```
Fuad Hasan
```

```
BME, KUET
```

```
*/
```

```
//Controlling LEDs With IR remote
```

```
#include<IRremote.h>
```

```
int IR_Recv = 12;    // IR Receiver Pin 3
```

```
int bluePin = 2;
```

```
int orangPin = 3;
```

```
int greenPin = 4;
```

```
IRrecv irrecv(IR_Recv);
```

```
decode_results results;
```

```
void setup(){
```

```
Serial.begin(9600); // starts serial communication

irrecv.enableIRIn(); // starts the receiver

pinMode(2, OUTPUT);    //sets the digital pin as output
pinMode(3, OUTPUT);    //sets the digital pin as output
pinMode(4, OUTPUT);    //sets the digital pin as output
}
```

```
void loop(){
  //decodes the infrared input
  if (irrecv.decode(&results)){
    long int decCode = results.value;
    Serial.println(results.value);

    //switch case to use the selected remote control button
    switch(results.value){
      case 16582903:  //when you press the 1 button
        digitalWrite(2, HIGH);
        break;

      case 16615543:  //when you press the 2 button
        digitalWrite(2, LOW);
        break;

      case 16599223:  //when you press the 3 button
        digitalWrite(3, HIGH);
        break;
```

```

case 16591063: //when you press the 4 button
digitalWrite(3, LOW);
break;

case 16623703: //when you press the 5 button
digitalWrite(4, HIGH);
break;

case 16607383: //when you press the 6 button
digitalWrite(4, LOW);
break;
}

irrecv.resume(); //receives the next value from the button
}

delay(10);
}

```

## 6. Code Explanation

- **Library Inclusion:**

- `#include<IRremote.h>` includes the necessary library to handle IR communication.

- **Pin Definitions:**

- `IR_Recv` is assigned to pin 12 where the IR receiver is connected.
- `bluePin`, `orangPin`, and `greenPin` are assigned to pins 2, 3, and 4 respectively.

- **Object Creation:**

- `IRrecv irrecv(IR_Recv);` creates an IR receiver object.

- `decode_results` results; will store the decoded IR signal data.
- **Setup Function:**
  - `Serial.begin(9600)`; initializes serial communication for debugging.
  - `irrecv.enableIRIn()`; starts the IR receiver.
  - `pinMode(2, 3, 4, OUTPUT)`; sets LED pins as outputs.
- **Loop Function:**
  - **Listening for IR Signal:** Arduino continuously checks for incoming IR signals using `irrecv.decode(&results)`.
  - **Decoding and Action:**
    - If a signal is received, it prints the decoded value to Serial Monitor.
    - It matches the received value using switch statement:
      - Button 1 turns ON blue LED (Pin 2)
      - Button 2 turns OFF blue LED
      - Button 3 turns ON orange LED (Pin 3)
      - Button 4 turns OFF orange LED
      - Button 5 turns ON green LED (Pin 4)
      - Button 6 turns OFF green LED
  - **Resuming Receiver:** After processing, `irrecv.resume()`; prepares for next incoming signals.
- **Delay:**
  - A short delay of 10 milliseconds to stabilize the loop.

## 7. Conclusion

This simple yet effective project enables the wireless control of multiple LEDs using an IR remote. It helps in understanding the basics of IR communication, Arduino programming, and digital outputs, making it a fundamental step toward more complex home automation or IoT applications.

