

**International Center for Free and Open Source
Software**



**Blinking LED Sequence with Arduino
Uno**

**Azwa Harshad
Internship
Open IoT
FEB 2023**

Contents

List of Figures	ii
1 Introduction	1
1.1 General Background	1
1.2 Working Principle	1
1.3 Components Needed	2
2 Circuit Diagram	3
2.1 Circuit Connections	3
3 Implementation	5
3.1 Procedure	5
4 Code	6
4.1 Code for control of servo motor using arduino	6
5 Result	9
5.1 Result	9

List of Figures

2.1 Circuit Diagram 3

Chapter 1

Introduction

1.1 General Background

By integrating eight LEDs with the Arduino Uno microcontroller, we orchestrate a captivating visual display where LEDs alternate their states in a rhythmic pattern. Through this project, we delve into fundamental concepts of digital control, timing, and pin manipulation,

1.2 Working Principle

The working principle of the "Blinking LED Sequence with Arduino Uno" project revolves around the manipulation of digital pins on the Arduino Uno microcontroller. Initially, the program configures the connected LEDs to specific digital pins, establishing them as outputs. Subsequently, the code orchestrates a sequence where odd-numbered LEDs are turned on while even-numbered LEDs are turned off for a predefined duration. This pattern alternates every second, achieved through the implementation of delay functions. By leveraging the ability of the Arduino Uno to control multiple digital outputs simultaneously, we create a visually engaging sequence that showcases the versatility and power of microcontroller-based systems for controlling external hardware components.

1.3 Components Needed

1. Arduino Board (e.g., Arduino Uno)
2. Eight LEDs (any color)
3. Eight current-limiting resistors (typically 220 ohms)
4. LED
5. Breadboard
6. Jumper Wires

Chapter 2

Circuit Diagram

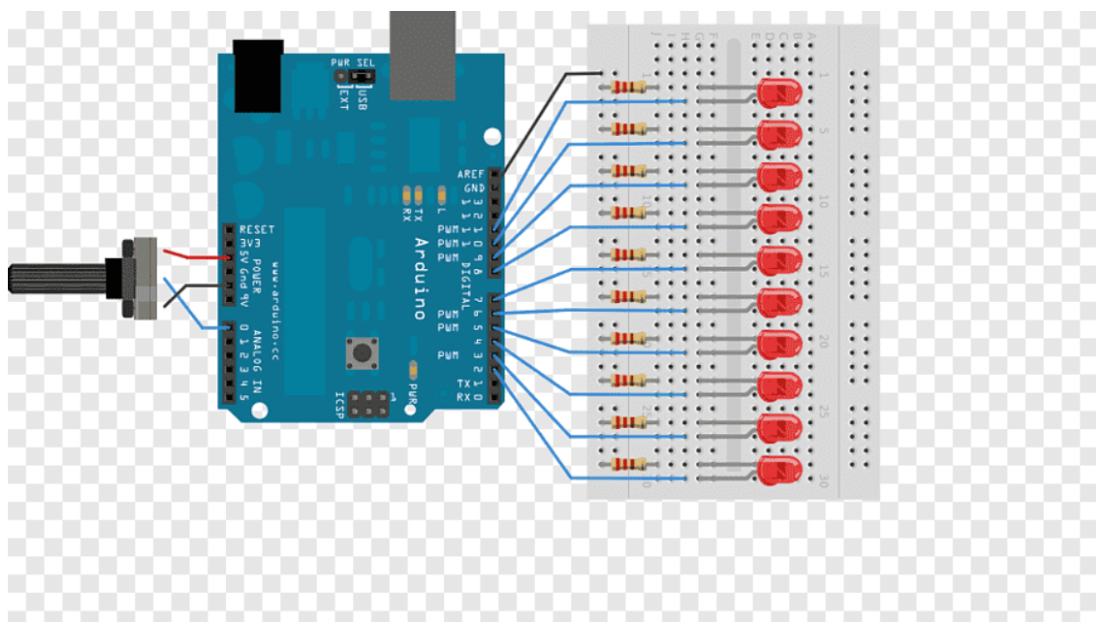


Figure 2.1: Circuit Diagram

2.1 Circuit Connections

1. Connect Arduino Uno to Power: Plug your Arduino Uno into your computer via USB cable for power.
2. Connect the anode (longer leg) of each LED to digital pins 2, 3, 4, 5, 6, 7, 8, and 9 on the Arduino Uno using jumper wires.

3. Connect the cathode (shorter leg) of each LED to the breadboard's ground (-) rail using jumper wires.
4. Connect a 220-ohm current-limiting resistor to the anode (longer leg) of each LED.
5. Connect the other end of each resistor to the corresponding digital pin on the Arduino Uno.
6. Connect a jumper wire from one of the ground (-) pins on the Arduino Uno to the breadboard's ground (-) rail.

Chapter 3

Implementation

3.1 Procedure

1. Setup Hardware: Gather all necessary components. Connect them according to the circuit diagram.
2. Install Arduino IDE: Download and install Arduino IDE from the official website.
3. Open Arduino IDE: Launch the Arduino IDE software.
4. Write Code: Compose your program using Arduino programming language (based on C/C++).
Write setup and loop functions.
5. Verify Code: Click on the Verify button (checkmark icon) to check for any errors in the code.
6. Upload Code: Connect Arduino board to the computer via USB. Select the correct board and port from Tools menu.
Click on the Upload button (right arrow icon) to upload the code to the Arduino board.
7. Test: Make sure the hardware is powered on.

Chapter 4

Code

4.1 Code for control of servo motor using arduino

It is done through ArduinoIDE

Code

```
int led1=8;
int led2=7;
int led3=6;
int led4=5;
int led5=4;
int led6=3;
int led7=2;

void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
    pinMode(led5,OUTPUT);
    pinMode(led6,OUTPUT);
```

```

    pinMode(led7,OUTPUT);

}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(led1,LOW);

    digitalWrite(led3,LOW);

    digitalWrite(led5,LOW);
    digitalWrite(led7,LOW);
    delay(1000);

    digitalWrite(led2,HIGH);

    digitalWrite(led4,HIGH);

    digitalWrite(led6,HIGH);
    delay(1000);

    digitalWrite(led1,HIGH);

    digitalWrite(led3,HIGH);

    digitalWrite(led5,HIGH);
    digitalWrite(led7,HIGH);

```

```
    delay(1000);

    digitalWrite(led2,LOW);

    digitalWrite(led4,LOW);


    digitalWrite(led6,LOW);
    delay(1000);

}
```

Chapter 5

Result

5.1 Result

The blinking of LED sequence using Arduino UNO is obtained. The odd-numbered LEDs illuminate while the even-numbered LEDs dim, creating an eye-catching visual sequence. After a second, the pattern reverses, with the even-numbered LEDs lighting up while the odd-numbered ones turn off. This cyclic pattern continues indefinitely, demonstrating the power of Arduino programming in orchestrating dynamic control of electronic components.