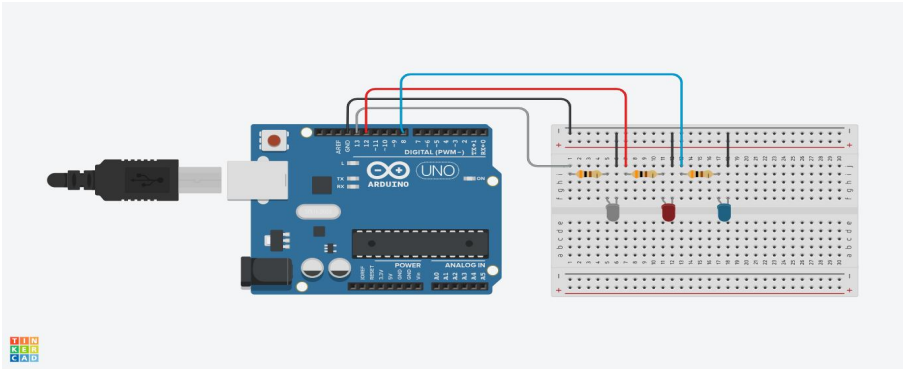
# Morse Code LED Blinking System (SOS Encoder)

## Project Description

This project simulates a Morse code transmitter using an Arduino Uno and three LEDs — a Red LED for dot (.), a Blue LED for dash (-), and a White/Grey LED as a power indicator. When a user inputs a string in the Serial Monitor, the system converts the message into Morse code and blinks the LEDs accordingly, following Morse timing rules.



Circuit Diagram: Circuit View

## Components Used

| Component | Quantity | Purpose |
|---|---|---|
| Arduino UNO | 1 | Main microcontroller |
| Red LED | 1 | Represents Dot (.) |
| Blue LED | 1 | Represents Dash (-) |
| White LED | 1 | Power Status Indicator |
| 330Ω Resistor | 3 | Current limiting for LEDs |
| Breadboard | 1 | Circuit prototyping |
| Jumper Wires | 6+ | Circuit connections |
| USB Cable | 1 | Arduino power & programming |

## Code Workflow

### 1. Setup Phase

Configures digital pins:
- Pin 13 → Power LED (White)
- Pin 12 → Dot LED (Red)
- Pin 8 → Dash LED (Blue)

Blinks all LEDs sequentially as a startup animation.

### 2. Input Phase

Prompts the user via Serial Monitor to input a message and converts the message to lowercase.

### 3. Encoding Phase

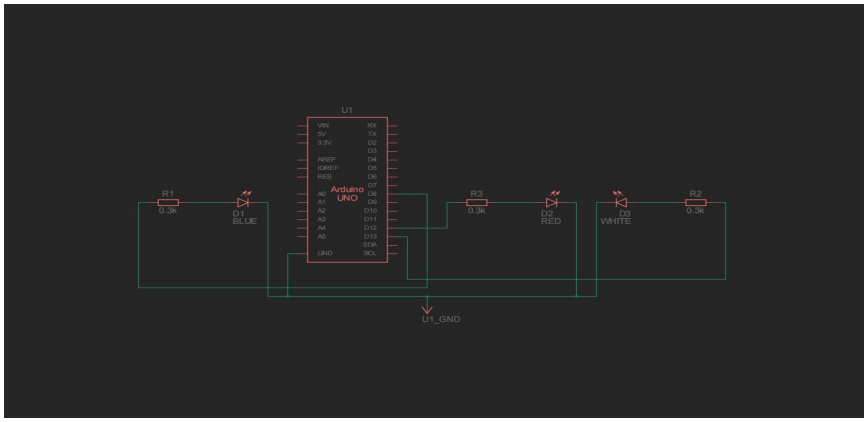
Each character is passed to a function encoder() that returns its Morse equivalent.

### 4. Simulation Phase

Iterates through the Morse string:
- '.' → Red LED blinks for 1 unit time
- '-' → Blue LED blinks for 3 unit time
- Space between letters is 3 units, between words 7 units

## Circuit Explanation

### Schematic Overview




Circuit Diagram: Schematic View

White (Power) → Pin 13
Red (Dot) → Pin 12
Blue (Dash) → Pin 8
All LEDs are connected in series with 330Ω resistors and go to GND.

## Importance of the Project

### Educational Value

Demonstrates:
- Digital output control
- Timing with delay()
- Serial input processing
- String handling in Arduino C++

### Practical Applications

- Emergency signaling (e.g., SOS)
- Visual communication in loud environments
- Teaching aid for learning Morse Code

### Future Extensions

- Add a buzzer for audio output
- Implement a wireless Morse code sender (using RF/Bluetooth)
- Add light sensors or motion detectors for auto-activation