Lab Report

Course: Embedded Systems and IoT Lab

Course Code: CSE234

Experiment No: 02

Experiment Name: Blinking LED through Arduino using Tinkercad

Submitted To

Muha. Humayet Islam

Lecturer,

Department of CSE

Submitted By

Name:

ID:

Section:

Department: CSE

Date: 23/06/2025

Experiment Name: Blinking LED through Arduino using Tinkercad

Introduction:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It allows users to create interactive electronic projects by programming a microcontroller to interact with sensors, actuators, and other electronic components. Essentially, it's a tool that enables interaction between the physical world (hardware) and digital code. We will construct a circuit with three LEDs, each controlled by a separate digital pin on the Arduino. By writing a simple program, we can command the Arduino to send high and low voltage signals to these pins, thus turning the LEDs on and off.

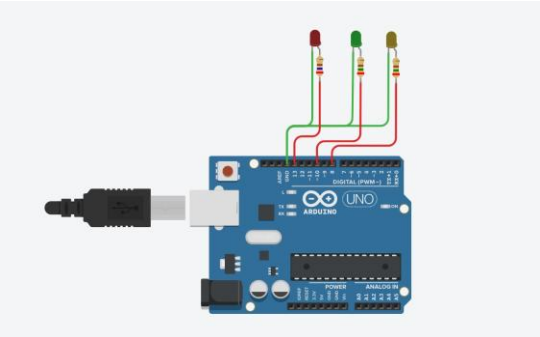
Objective :

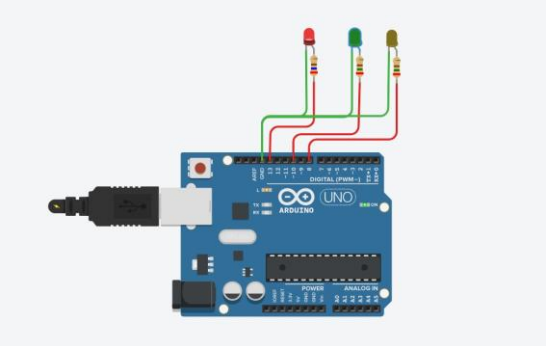
- To learn how to control multiple LEDs using an Arduino.
- To understand how to write and upload Arduino code using Tinkercad.
- To see how LEDs can be turned on and off in a sequence using code.

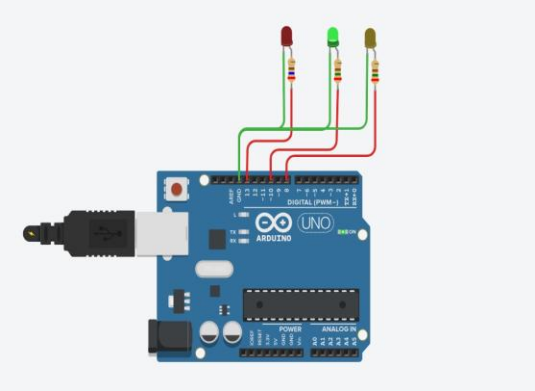
Requirements:

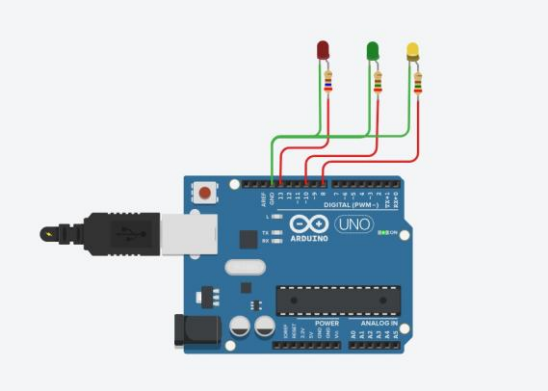
- 1 x Arduino Board
- 3 x LED (Red, Green, Yellow)
- 3 x Resistance (250Ω or suitable values)
- Connecting Wires

Figure:









How the LED Blinking Works:

1. **Pin Organization:** The code first defines an array, `ledPins[]`, to store the digital pin numbers `{13, 10, 8}`. This approach makes the code organized and easy to modify for a different number of LEDs.

2. **System Initialization:** In the `setup()` function, which runs only once at the start, a `for` loop iterates through the `ledPins[]` array. It uses the `pinMode()` command to configure each pin as an `OUTPUT`, preparing it to send voltage signals.

3. **Sequential Iteration:** The main `loop()` function runs continuously and contains another `for` loop. This loop steps through the `ledPins[]` array from the first to the last element, ensuring that each LED is addressed one after another in a defined sequence.

4. **LED Activation (ON State):** For the currently selected pin in the loop, the command `digitalWrite(pin, HIGH)` is executed. This sends a 5V signal to the pin, which completes the circuit and illuminates the LED. The `delay(1000)` command then pauses the program for 1 second, keeping the LED on.

5. **LED Deactivation (OFF State):** Immediately following the ON-state delay, the command `digitalWrite(pin, LOW)` is sent to the same pin. This cuts the voltage to 0V, turning the LED off. A second `delay(1000)` command creates another 1-second pause while the LED is off.

6. **Continuous Repetition:** After one LED completes its full ON/OFF cycle, the `for` loop advances to the next LED in the array. When the

sequence for the last LED is finished, the main `loop()` function restarts the entire process from the beginning, creating a continuous and repeating light pattern.

7. **Component Protection:** In the physical circuit, the 250Ω resistor connected to each LED is crucial. It limits the electrical current flowing from the Arduino's digital pin to a safe level, protecting the LED from burning out and preventing potential damage to the Arduino board itself.

Arduino Code:

```
// C++ code
//
int ledpins[]={13,10,8};
int numled=3;
void setup()
{
  for(int i=0;i<numled;i++)
  {
    pinMode(ledpins[i],OUTPUT);
  }
}
void loop()
{
  for(int i=0;i<numled;i++)
  {
  digitalWrite(ledpins[i], HIGH);
   delay(1000);
  digitalWrite(ledpins[i],LOW);
```

```
  delay(1000);
  }
}
```

Result and Discussion :

Upon running the simulation, the circuit performed as expected. The LEDs began to blink sequentially, validating the integrity of both the virtual wiring and the code. Each LED illuminated for 1 second and then turned off for the same duration, directly corresponding to the `delay(1000)` commands. This outcome successfully demonstrates how arrays and loops can be used to efficiently manage multiple outputs. The experiment also confirmed Tinkercad's utility as a reliable platform for testing circuits virtually.

Reference :

- Tinkercad by Autodesk: https://www.tinkercad.com/m/
- "Practical Electronics for Inventors" by Paul Scherz and Simon Monk
- All About Circuits: https://www.allababoutcircuits.com/m/
- Electronics Tutorials: https://www.electrtronics-tutorials.ws/