Experiment No : 02

Experiment Name: Blinking an LED using Arduino UNO.

Explanation

LED : LEDs (Light Emitting Diodes) are becoming increasingly popular among a wide range of people. When a voltage is given to a PN Junction Diode, electrons, and holes recombine in the PN Junction and release energy in the form of light. An LED's electrical sign is comparable to that of a PN Junction Diode. When free electrons in the conduction band recombine with holes in the valence band in forward bias, energy is released in the form of light.

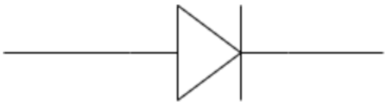


Figure - Diode

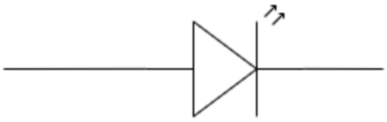


Figure - Light Emitting Diode (LED)

Structure of an LED :

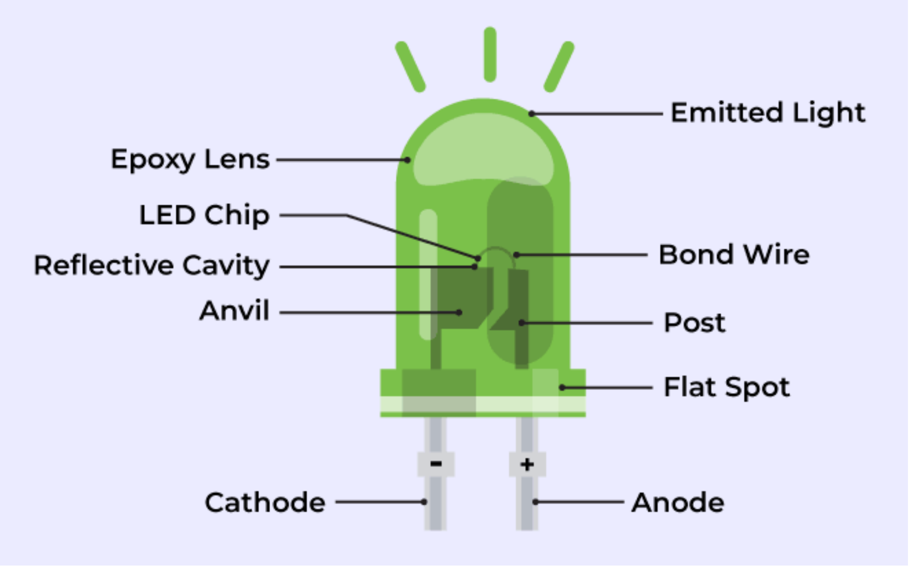


Figure Name: Structure of LED

Blinking an LED

Blinking an LED is an introductory Arduino project in which we control an LED using Arduino. LED blinking refers to the process of continuously turning an LED (Light Emitting Diode) and off in a repetitive pattern. It is a simple and common demonstration in electronics and microcontroller-based projects.

Components Required

1. 2 X LED

2. 1 X Resistor, 1 KOhm

3. Breadboard

4. Arduino UNO .

5. Jumper wires

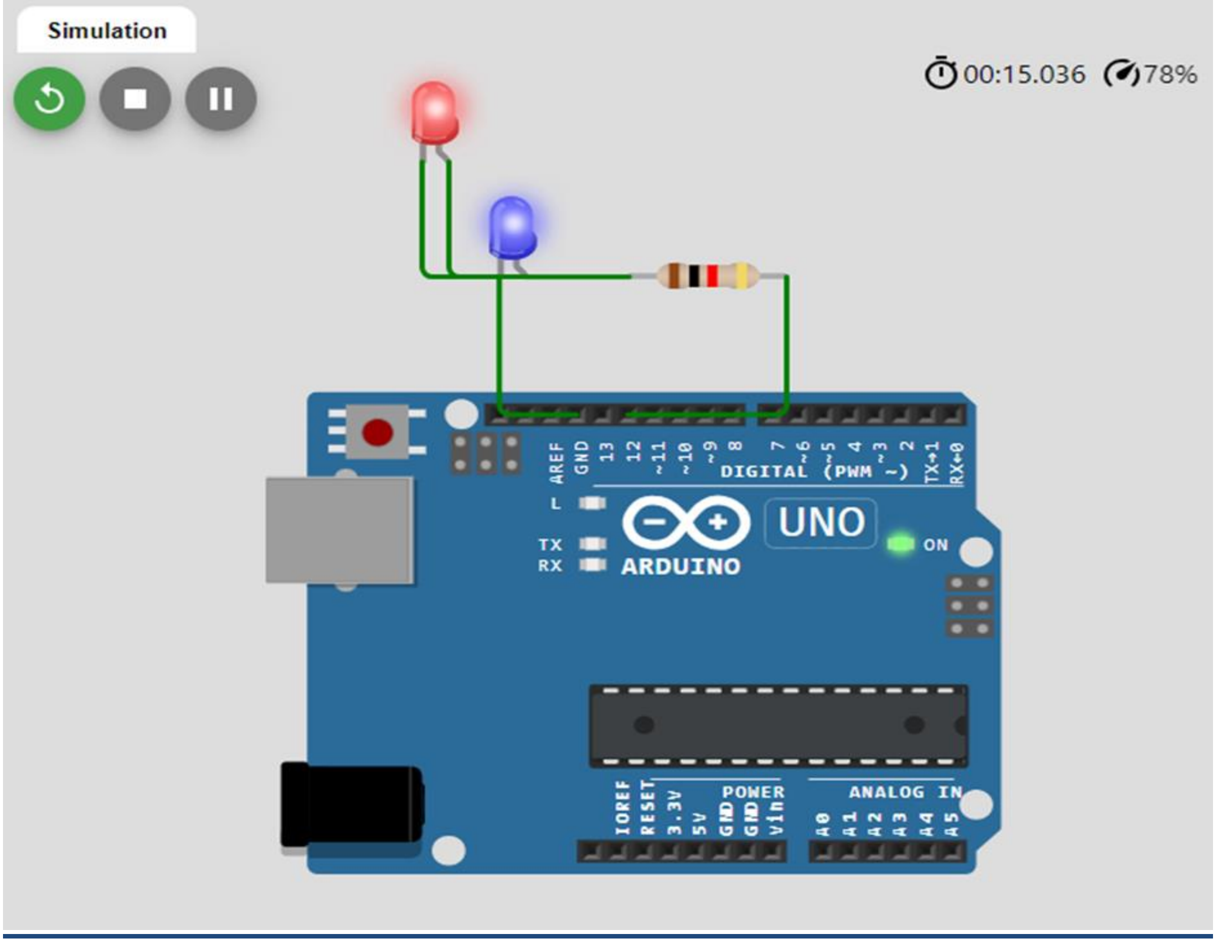
Working Procedure

setup() and loop() are two fundamental Arduino functions for controlling the behavior of your board. The Arduino framework automatically calls these functions, which form the foundation of any Arduino program.

The setup() function is only called once when the Arduino board boots up or is reset. Its goal is to set pin modes, initialize variables, and execute any other necessary setup tasks before the main loop begins. This function can be used to configure settings that should only be changed once over the board’s lifespan.

The loop() function is the heart of an Arduino program. After the setup() function is executed, the loop() function starts running repeatedly until the Arduino is powered off or reset. It contains the main code that performs the desired tasks, controls the board, user input. Whatever is included in the loop() function will be executed in a continuous loop, allowing the Arduino to perform its intended functions continuously.

Diagram



Code Snippet

```
void setup() {
        pinMode(12, OUTPUT);
  }
  void loop() {
       digitalWrite(12, HIGH);
    delay(1000);
      digitalWrite(12, LOW);
   delay(1000);
}
```

Code Explanation

void setup(): In Arduino, the setup function runs once while pressing reset or power the board.

pinMode(12, OUTPUT): Initialize digital pin 12 as an output.

void loop(): The loop function runs over and over again for infinite times.

digitalWrite(12, HIGH): Sets the voltage to High, turning the LED on.

delay(1000): Wait for 1 second, which keeps the LED on for 1 second.

digitalWrite(12, LOW): Sets the voltage to Low, turning the LED off.

delay(1000): Wait for 1 second, which keeps the LED off for 1 second.

Conclusion

The Arduino LED blinking project provides a hands-on introduction to microcontrollers, hardware interfaces, and programming ideas. It serves as a foundation for more sophisticated projects and allows you to experiment with numerous Arduino features and capabilities. Whether you’re new to electronics or an expert maker, this project will help you develop crucial skills and knowledge for future Arduino-based projects.

Experiment No: 03

Experiment Name: Fading   LED using Arduino UNO.

Explanation

Fading LED: A **fading LED** refers to the effect where the brightness of an LED (Light Emitting Diode) gradually increases and decreases in a smooth, continuous manner. This creates a "fade-in" and "fade-out" visual effect, where the LED transitions from being dim to bright and then from bright to dim over time.

PWM (Pulse Width Modulation) technique: Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means.

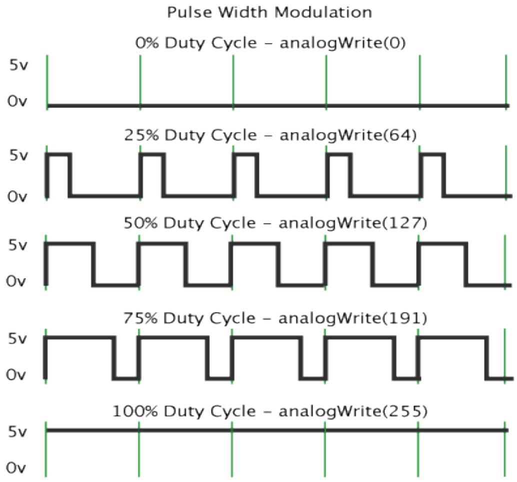

Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
25% Duty Cycle - analogWrite(64)
50% Duty Cycle - analogWrite(127)
75% Duty Cycle - analogWrite(191)
100% Duty Cycle - analogWrite(255)
5v
0v

Figure Name: Pulse Width Modulation.

Fading effect is typically achieved by controlling the voltage or current supplied to the LED. In microcontroller based systems like Arduino, this is done using **PWM**. PWM allows us to change the LED's brightness by varying the amount of time the LED is turned on and off very quickly. By adjusting the ratio of the on-time to off-time (the duty cycle), the LED's brightness appears to fade in and out smoothly to the human eye.

Components Required

1. 2 X LED

2. 1 X Resistor, 1 KOhm

3. Breadboard

4. Arduino UNO.

5. Jumper wires

Working Procedure

I. Plug the longer leg of the LED into pin 9 on your Arduino Uno board.

II. Plug the shorter leg of the LED into the breadboard.

III. Connect one end of the 1KOhm resistor to the shorter leg of the LED.

IV. Connect the other end of the resistor to the ground (GND) pin on the Arduino Uno board.

V. Use a jumper wire to connect pin 9 on the Arduino Uno board to the breadboard.

VI. Plug the other end of the jumper wire into an empty row on the breadboard.

Diagram

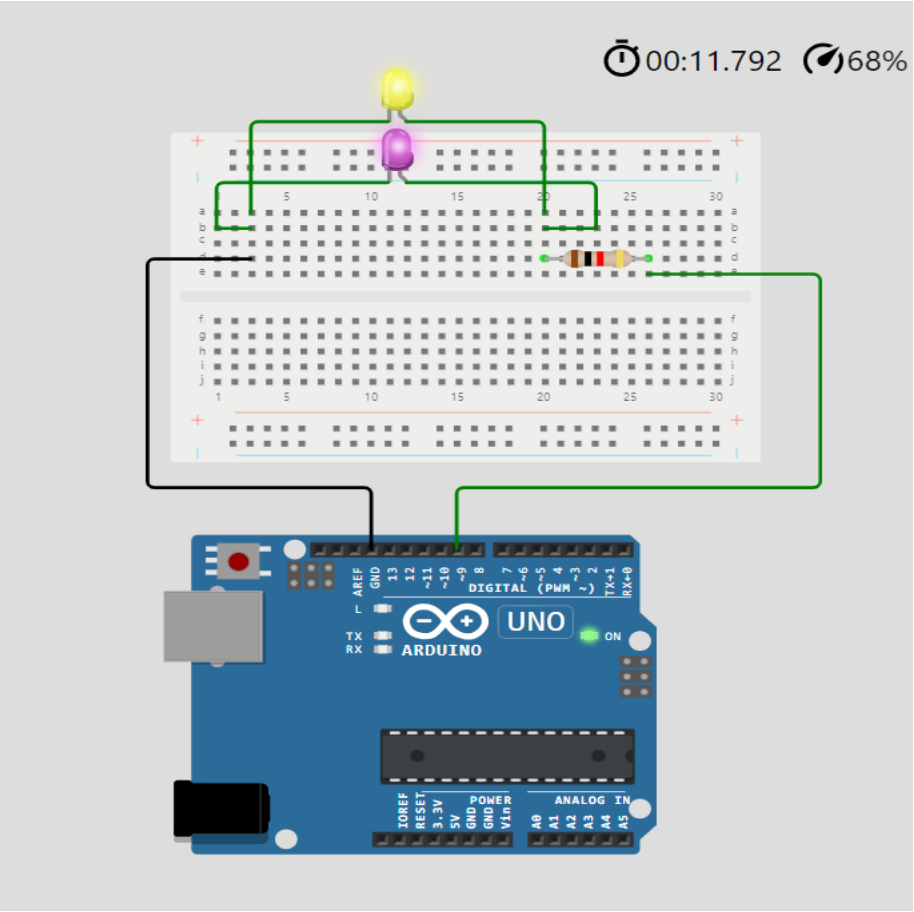


Figure Name: Fading LED.

Code Snippet

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  delay(300);
}
```

Code explanation

led: The variable led is set to pin 9, where the LED is connected.

brightness: It controls the brightness level of the LED, starting at 0.

fadeAmount: It sets how much the brightness will change in each step (increase or decrease by 5).

setup(): Runs once when the board is powered on or reset.

pinMode(led, OUTPUT): Configures pin 9 as an output pin to control the LED.

loop(): Repeats continuously.

analogWrite(led, brightness): Sets the brightness of the LED based on the value of brightness.

brightness = brightness + fadeAmount: Adjusts the brightness by adding or subtracting fadeAmount.

if (brightness == 0 || brightness == 255): Reverses the fading direction when brightness reaches 0 (completely off) or 255 (fully on).

delay(300): Pauses the loop for 300 milliseconds to make the fading effect visible.

Conclusion: In this project we use the Arduino uno r3 to dynamically light a led bulb. The bulb slowly light up then after some time (in this case 300 milliseconds) then slowly fades away. We use a pwm (Pulse with Modulation) pin no 9 to connect with the led. The code here works as an instruction for the Arduino to how to light up the led. From this experiment we learned how to construct a basic circuit with Arduino and apply sketch and logic building.