Lab Report No: 02

Lab Title: Blinking LED in a specified time interval

Introduction:

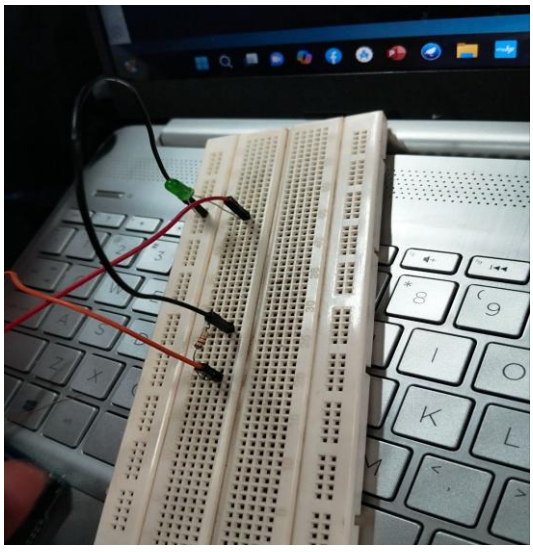
This experiment demonstrates how to control an LED using a microcontroller to blink at specific time intervals. By programming the microcontroller to turn the LED ON and OFF with a delay, the blink rate can be adjusted. This basic concept is essential for understanding embedded systems and serves as the foundation for more advanced applications.

Theory:

Blinking an LED involves toggling its ON and OFF states at regular intervals. A delay function or timer is used to achieve this. The LED is connected to a digital output pin of the microcontroller and controlled using simple high (ON) and low (OFF) signals.

Working Procedure:

- **Circuit Setup:**
    a. Connect the LED and resistor to the breadboard.
    b. Wire the anode of the LED to a digital pin of the microcontroller (e.g., Pin 13).
    c. Wire the cathode to GND.



- **Programming:**
    a. Write a program to blink the LED using the specified time interval.
    b. Upload the program to the microcontroller using Arduino IDE or other software.

Arduino Code:

For Analog Light:

```
int led=13;
int bright;

void setup() {
  // put your setup code here, to run once:
     pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  for(bright=0;bright<=255;bright=bright+5)
  {
    analogWrite(led,bright);
    delay(10);
  }
  for(bright=255;bright>=0;bright=bright-5)
  {
    analogWrite(led,bright);
     delay(10);
  }
}
```

For Digital Light:

```
 int led=13;
void setup() {
  // put your setup code here, to run once:
  pinMode(led,OUTPUT);
}

void loop() {
```

```
  // put your main code here, to run repeatedly:
  digitalWrite(led,HIGH);
  delay(50);
   digitalWrite(led,LOW);
  delay(50);
}
```

- **Testing:**
    a. Observe the LED blinking as per the specified interval.
    b. Adjust the interval in the code to test different durations.

Execution and Observations:

Execution Steps

- **Setup the Circuit:**

    a. Connected the LED anode to digital pin 13 of the microcontroller through a 220Ω resistor.
    b. Connected the LED cathode to the GND pin.
    c. Verified all connections to ensure proper wiring.

- **Upload the Code:**

    a. Uploaded the provided LED blink code to the microcontroller using the Arduino IDE.
    b. Modified the delay() time for different intervals (1000ms, 500ms, 50ms) and uploaded the program for each test.

- **Run the Program:**

    a. Powered the microcontroller and observed the LED behavior for each delay value.

Results:

Delay Time (ms)	Execution	Observed LED Behavior	Remarks
1000	Code executed successfully	LED turned ON for 1 second, OFF for 1 second	Clear and visible blinking pattern
500	Code executed successfully	LED turned ON for 0.5 seconds, OFF for 0.5 seconds	Noticeable but faster blinking
50	Code executed successfully	LED turned ON for 50ms, OFF for 50ms	Blinking too fast to observe distinctly, appeared as dim glow

Additional Notes:

For 50ms delay, the human eye could not detect the ON/OFF transitions clearly due to the high frequency, resulting in a perceived continuous glow.

The experiment demonstrated how reducing the delay time increases the blink frequency. Would you like to include these observations in your lab report or discuss further?

Conclusion:

The experiment successfully demonstrated how to blink an LED using a microcontroller at different time intervals. By adjusting the delay in the program, the ON/OFF duration of the LED was controlled, showcasing the relationship between delay time and blink frequency.

The results highlight the simplicity and effectiveness of using microcontrollers for basic output control, which can be extended to more complex systems like LED patterns, traffic signals, and alarms. This experiment provides a solid foundation for learning embedded systems and understanding the importance of timing in electronic applications.