LAB REPORT

Course Title: **Embedded Systems and IoT Lab**

Course Code: **CSE-234**

Topic Name: Blinking LED circuit with Arduino.

Submitted To:

Avizit Nandi
Lecturer
Department of CSE
Daffodil International University

Submitted By:

Name: Rita Faria Richi

ID: 221-15-5025

Section: 61-T2

Department of CSE

Daffodil International University

Date of Submission: 31-07-2025

Lab Report No: 06

Lab Report Name: Blinking LED circuit with Arduino.

Objective:

- Develop hands-on experience with the Arduino Uno board and digital pin interfacing.
- Understand how microcontrollers interact with external components like LEDs.
- Learn to write and upload basic sketches using the Arduino IDE.
- Explore the use of delay functions to control timing and behavior in embedded systems.
- Build a strong foundation for more advanced electronics and automation projects.

Introduction:

In the world of embedded systems and electronics, the ability to control hardware through software is a foundational skill. Arduino, an open-source microcontroller platform, provides an easy and accessible way to learn this interaction. One of the simplest yet most essential experiments to begin with is the blinking LED circuit — often referred to as the "Hello World" of Arduino programming. This experiment introduces students to the Arduino Uno board and the basics of digital output. By writing a short program that turns an LED on and off at regular intervals, learners gain practical knowledge of microcontroller pin configuration, timing control using delays, and circuit building using breadboards and components. The blinking LED not only demonstrates how code can manipulate hardware in real time but also sets the stage for more complex projects involving sensors, actuators, and real-world automation. Through this lab, students take their first step into the exciting field of embedded electronics and programmable systems.

Theory:

The blinking LED experiment demonstrates the basic concept of digital output using a microcontroller — in this case, the Arduino Uno. An LED (Light Emitting Diode) is a simple electronic component that emits light when current flows through it in the forward direction. It requires minimal voltage and current to operate and is commonly used in electronics to indicate status or activity.

- **Arduino Uno and Digital Pins**
  The Arduino Uno is an open-source microcontroller board based on the ATmega328P chip. It has 14 digital input/output pins, of which certain pins can be programmed to send HIGH (5V) or LOW (0V) signals. By connecting an LED to one of these pins, the Arduino can control its ON and OFF state via software.

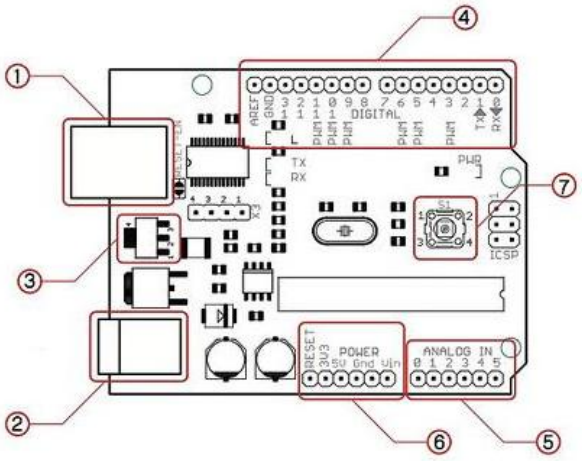

Figure-1: Structure of Arduino

- **Basic Working Principle**
  - ✓ Setting the pin as an OUTPUT using pinMode()
  - ✓ Sending a HIGH signal to turn the LED ON
  - ✓ Introducing a delay using delay () function
  - ✓ Sending a LOW signal to turn the LED OFF

  This cycle causes the LED to blink at a regular interval.

- **Key Functions Used**
  - ✓ pinMode(pin, mode): Configures the specified pin to behave either as an input or an output.
  - ✓ digitalWrite(pin, value): Sets the digital pin to HIGH or LOW.
  - ✓ delay(ms): Pauses the program for a specified number of milliseconds.

Equipment:

- Arduino Uno
- USB Cable
- LED
- Resistor
- Breadboard
- Jumper Wires
- Transistor
- Switch

Circuit Diagram:

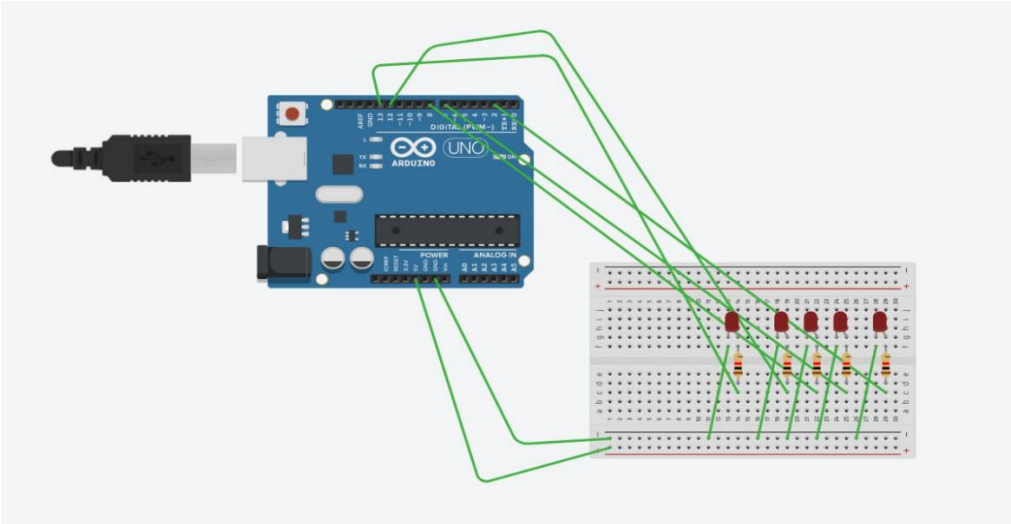

Figure-2: Simulated Output from Tinkercad (using LED,Register & Arduino)

Code:

```
void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  digitalWrite(12, LOW);
  digitalWrite(8, LOW);
  digitalWrite(7, LOW);
  digitalWrite(2, LOW);


  delay(200);
  digitalWrite(13, LOW);
  digitalWrite(12, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(7, LOW);
  digitalWrite(2, LOW);

  delay(200);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(2, LOW);

   delay(200);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(8, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(2, LOW);

    delay(200);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(8, LOW);
  digitalWrite(7, LOW);
  digitalWrite(2, HIGH);

  delay(200);
}
```

Figure-3: Arduino Code for Blinking LED (using LED,Register & Arduino)

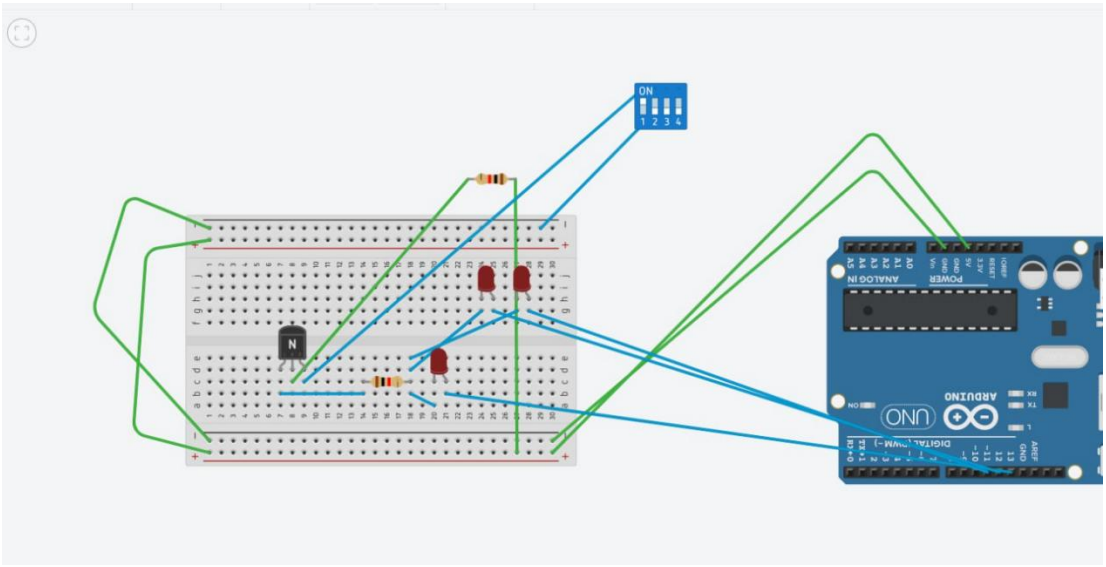

Figure-4: Simulated Output from Tinkercad (using Transistor, LED, Register & Arduino)

Code:

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(300);
  digitalWrite(13, LOW);

  digitalWrite(12, HIGH);
  delay(300);
  digitalWrite(12, LOW);

  digitalWrite(11, HIGH);
  delay(300);
  digitalWrite(11, LOW);

  delay(300);
}
```

Figure-5: Arduino Code for Blinking LED (using Transistor, LED, Register & Arduino)

Discussion:

This lab experiment successfully demonstrated the basic concept of controlling an LED using an Arduino Uno board. The experiment was carried out using two different circuit configurations—one with a direct LED-resistor connection and another involving a transistor to control the LED.

In the first setup, the LED was connected to a digital pin of the Arduino through a current-limiting resistor. The code used pinMode() to set the pin as an output and digitalWrite() along with delay() to alternate between HIGH and LOW signals, creating a blinking effect. This illustrates the fundamental operation of microcontroller GPIO pins and how they interact with external components. In the second configuration, a transistor was used as a switch to control the LED. The Arduino pin sent a signal to the transistor base, allowing current to flow from collector to emitter, thus lighting the LED. This method is more efficient when dealing with components requiring higher current than the Arduino pin can directly supply. It also introduces students to the concept of interfacing external hardware using transistors, which is crucial for advanced embedded applications.

Through simulation (using Tinkercad) and hands-on coding, the lab reinforced essential programming logic, hardware interfacing, and the importance of timing control using delays. The code successfully produced visible blinking effects in both circuits, confirming correct wiring and logic implementation.

Conclusion:

This lab provided valuable insights into how microcontrollers interact with external components like LEDs. The blinking LED circuit, though simple, served as an excellent introduction to digital output, pin configuration, and the use of delay functions in embedded systems.

- ✓ Hands-on experience with Arduino Uno board setup and wiring.
- ✓ Writing, uploading, and running code using the Arduino IDE.
- ✓ Demonstrating digital pin control through simple software logic.
- ✓ Learning the use of delay for timing-based tasks.
- ✓ Understanding how to use a transistor for hardware control.

Overall, the lab effectively fulfilled its learning goals and laid a solid foundation for more complex embedded system applications such as sensor interfacing, motor control, and automation tasks.