Practical Robotics Projects with Arduino

(CSE 4571)

Lab Assignment No – 01

LED Blinking

Submission Date: _______________

Branch: CSE		**Section:2241021**
Name	**Registration No.**	**Signature**
Hridesh Swain	2241019596	

Department of Computer Science and Engineering
Institute of Technical Education and Research (Faculty of Engineering)
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneswar, Odisha-751030.

Aim:

LED Blinking – To write programs using Arduino UNO to blink LEDs in different pattern at regular intervals with digital output pins.

Objectives:

1) **Gain familiarity with the Arduino UNO and robotic system.**
2) **Develop the ability to write an Arduino program to blink the onboard LED of the Arduino Uno with a 25% duty cycle for a blink time of 2 seconds (with and without global variable).**
3) **Build an external circuit using an LED and control it from the Arduino, with a 75% duty cycle and a blink time of 1 second.**
4) **Create a circuit with two LEDs and learn about "for loops" with the serial monitor in the Arduino sketch.** The Red LED (First LED) will blink for 5 times and the Blue LED (Second LED) will blink for 3 times. The anode of the Red LED will be connected toArduino digital pin 9, and the anode of the Blue LED will be connected to Arduino digital pin 10. The duty cycle of the Red LED will be 50%, with an on time of 200ms, and the duty cycle of the Blue LED will be 25%, with an on time of 400ms.

   4.1:Control Two LED Circuit without a loop

   - ✓ Write an Arduino sketch to control the two LEDs without using al oop.
   - ✓ Observe the LED behavior and take note of any challenges orissuesen countered.

   4.2:Control Two LED Circuit with a for loop

   - ✓ Write an Arduino sketch to control the two LEDs using a for loop.
   - ✓ Use the for loop to blink the LEDs the desired number of times.
   - ✓ Observe the LED behavior and compare it to the previous sub-objective.

   4.3 :Control Two LED Circuit with a forl oop using the Serial monitor

   - ✓ Modify the previous sketch to use the Serial monitor to specify the number of times each LED should blink at a baud rate of 9600 bits/sec.

5) **Implement a basic circular LED chaser pattern.:**

   - ✓ WriteanArduinosketchtocontrol8LEDsconnectedfrompins2to9inacircular LED chaser pattern. (The LEDs should be turned on one at a time in a circular pattern starting from LED 0, then turned off and the next LED in the sequence should be turned on until all 8 LEDs have been lit. After the last LED is turned on, LED 0 should be turned off and the pattern should repeat in a continuous circular pattern.)

5.2 Implement a custom LED chaser pattern:

✓ Write an Arduino sketch to control a custom LED chaser pattern that combines five different patterns in a loop, including the "Bouncing Ball," "Zig-Zag," "Random Blink," "Knight Rider" and "Running Lights," patterns:

- The "Bouncing ball" pattern: Turns on the LEDs one by one, then turns them off one by one in the opposite direction.
- The " Zig-zag " pattern: Turns on the LEDs one by one, then turns them off one by one in the opposite direction.
- The"Randomblink"pattern:Generatesarandomnumberbetween0and1 foreachLEDandturnsontheLEDiftherandomnumberis1,thenturnsoff all LEDs.
- The "Knight Rider" pattern: Turns on the LEDs one by one, then turns them off one by one in the opposite direction.
- The “Running Lights” pattern: The LEDs turn on and off in sequence, with each LED turning off before the next one turns on.

Pre-Lab Questionnaire:

A. Experiment-Specific

1. What is the purpose of the pinMode() and digitalWrite() functions in Arduino programming?
2. Why do we use the delay() function in the LED blinking program? What would happen if the delay is removed?
3. Modify the code to make the LED blink twice as fast. What changes are required?
4. How would you connect multiple LEDs and make them blink alternately?
5. If you connect the LED directly to the Arduino pin without a resistor, what problem might occur and why?

B. Basics of Arduino UNO

6. What is the function of the ATmega328P microcontroller on the Arduino UNO board?
7. Identify the role of the digital pins and analog pins on the Arduino UNO.
8. What is the maximum current an I/O pin of Arduino UNO can safely provide?
9. What are the main differences between uploading code to Arduino UNO using USB and powering it using an external power supply?
10. Explain the use of the onboard LED (connected to pin 13).

C. Basics of Robotics

11. Define a robot. What distinguishes a robot from a simple automated machine?
12. List and explain the three main components of a robot (sensing, control, actuation).
13. Why is programmability considered an essential feature of robots?
14. Give two real-world examples of robots that use simple blinking/lighting as part of their operation.
15. How can an LED indicator be useful in debugging robotics systems?

Answers to Pre-Lab Questions

Components/EquipmentRequired:

Sl. No.	Nameofthe Component/Equipment	Specification	Quantity
1)	ArduinoUNOR3	16MHz	1
2)	ArduinoUNOcable	USBTypeAtoMicro-B	1
3)	Resistors(carbontype)	¼watt(330Ω)	8
4)	LED	Anycolourofyourchoice	8
5)	Breadboard	840Tiepoints	1
6)	JumperWire	---------------------------	Asperrequirement

Objective2

Develop the ability to write an Arduino program to blinkthe onboard LED of the Arduino Uno with a 25% duty cycle for a blink time of 2seconds(with and without global variable).

Circuit/Schematic Diagram

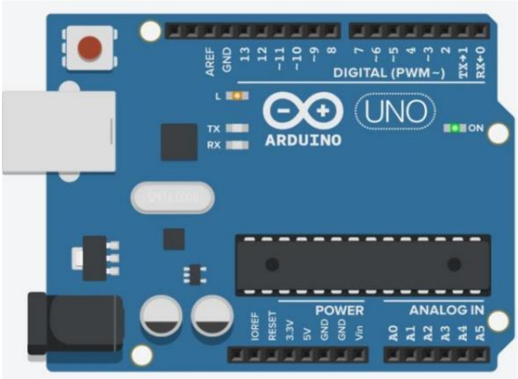


Figure1:Schematic of blinking of the on board LED of the Arduino Uno

Code

2.1 :write an Arduino program to blink the on board LED without using global variable

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Configure onboard LED
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // LED ON
  delay(500);                        // wait 500 ms
  digitalWrite(LED_BUILTIN, LOW);    // LED OFF
  delay(1500);                       // wait 500 ms
}
```

2.2 write an Arduino program to blink the on board LED using global variable

```
  // Global variable
int ledPin = LED_BUILTIN;    // Onboard LED (usually pin 13)

void setup() {
  pinMode(ledPin, OUTPUT);  // Configure onboard LED
}
```

```
    void loop() {
      digitalWrite(ledPin, HIGH);   // LED ON
      delay(500);                   // wait 500 ms
      digitalWrite(ledPin, LOW);    // LED OFF
      delay(1500);                  // wait 500 ms
}
```

Observation :

Figure 2: (Simulation based blinking of the onboard LED of the Arduino Uno with a 25% duty cycle for a blink time of 2 seconds

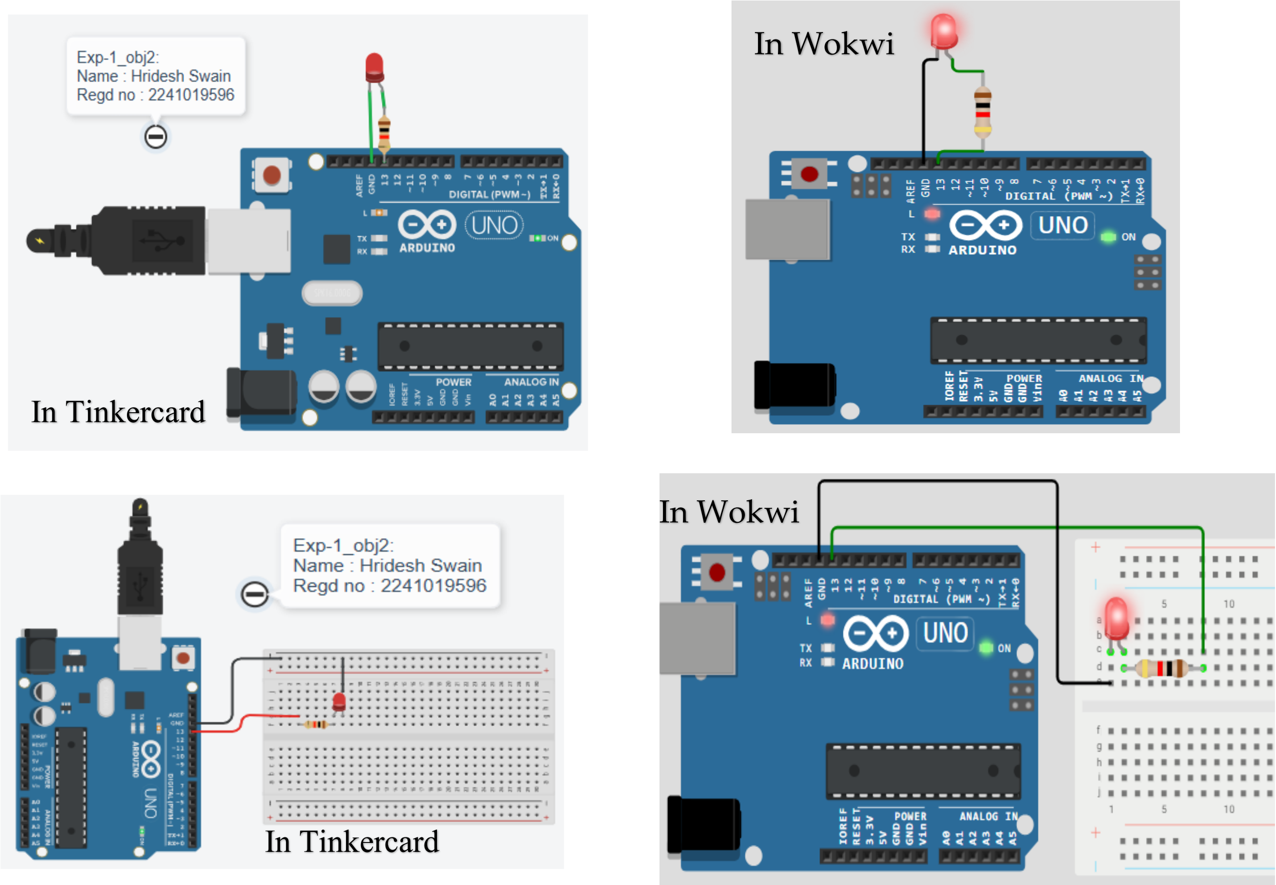
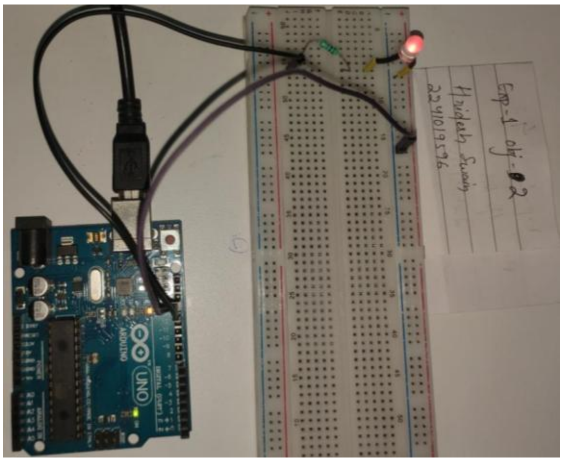
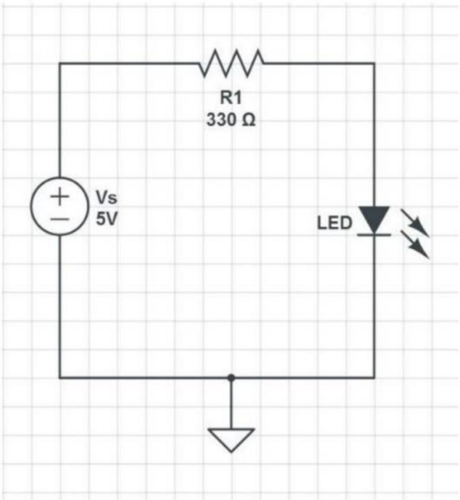


Figure 3: (Hardware Implementation based blinking of the onboard LED of the Arduino Uno with a 25% duty cycle for a blink time of 2 second .



Objective3

Build an external circuit using an LED and control It from the Arduino , with a 75% duty cycle and a blink time of 1 second.



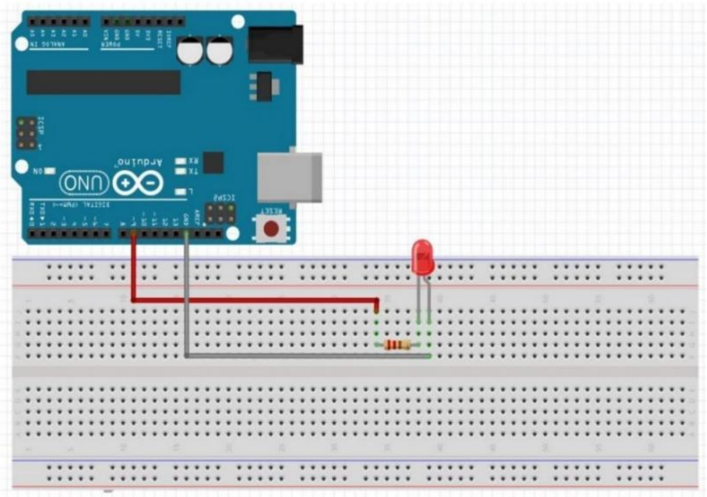


Figure4:Simple LED Circuit for blinking an LED

Figure5:Hardware Implementation to blink an L

Code :

```
void setup() {
          // initialize digital pin LED_BUILTIN as an output.
          pinMode(LED_BUILTIN, OUTPUT);
         }

         // the loop function runs over and over again forever
         void loop() {
          digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
          delay(750);                       // wait for 750 ms
          digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
          delay(250);                       // wait for 250 ms
         }
```

Observation

Figure6:(Simulation based external circuit using an LED and control it from the Arduino, with a 75% duty cycle and a blink time of 1 second.)

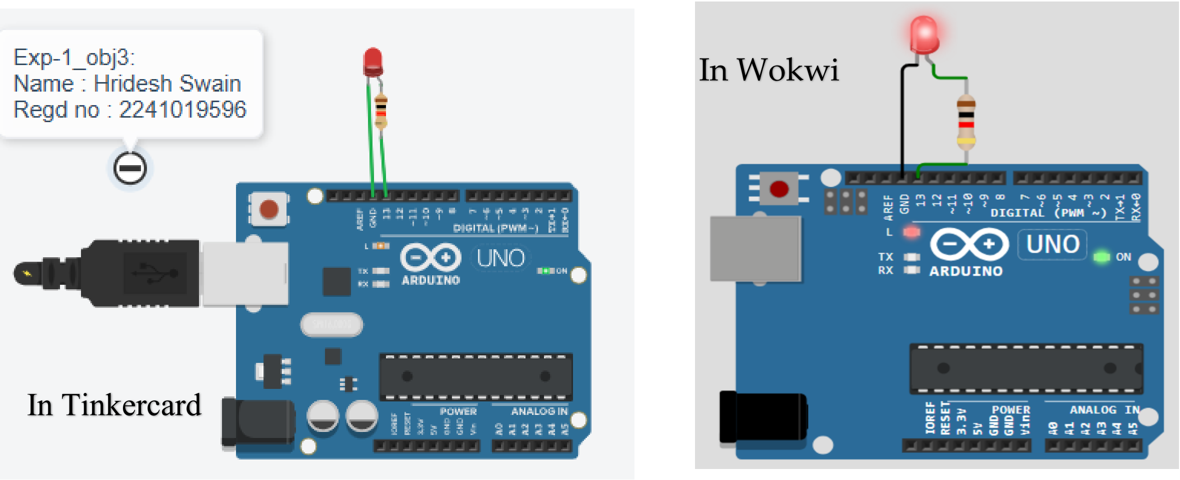
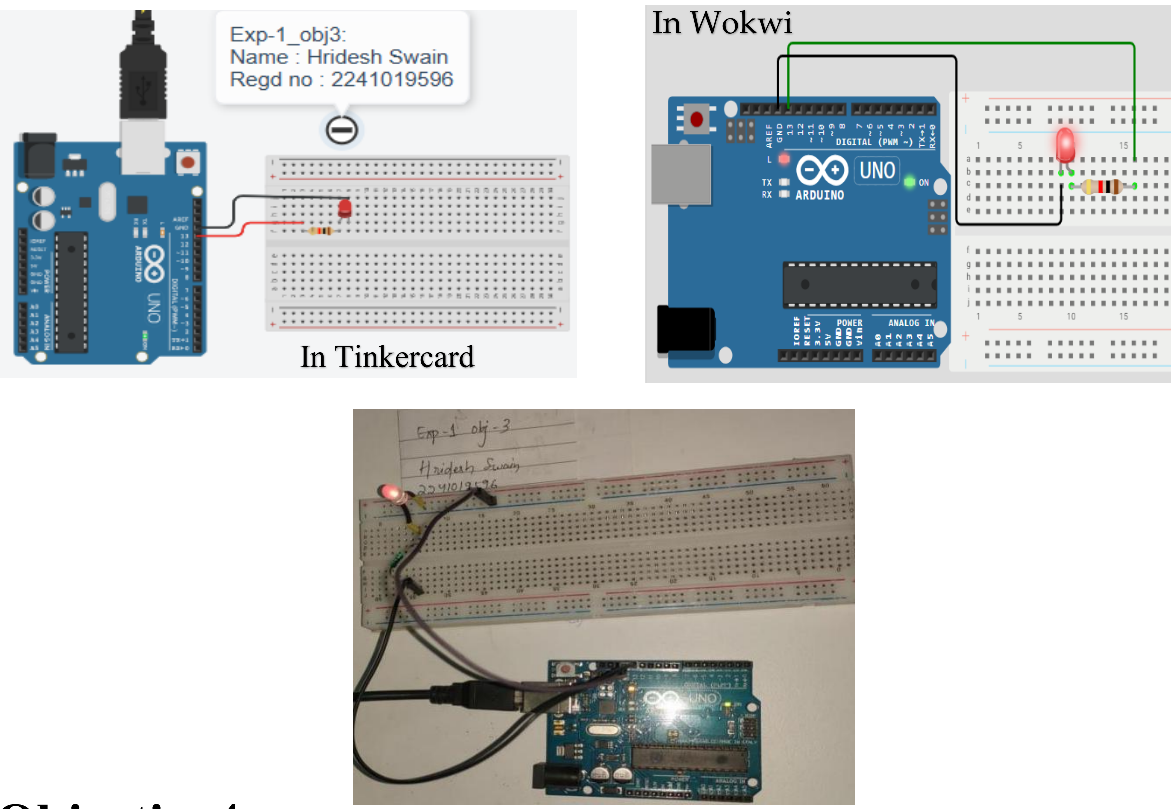


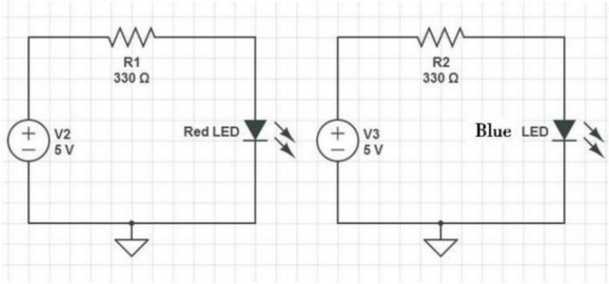
Figure7:(Hardware Implementation based external circuit using an LED and control it from the Arduino, with a 75% duty cycle and a blink time of 1 second.)



Objective4

Create a circuit with two LEDs and learn about "for loops" with the serial monitor in the Arduino sketch. The Red LED (First LED) will blink for 5 times and the Blue LED (Second LED) will blink for 3 times. The anode of the Red LED will be connected to Arduino digital pin 9, and the anode of the Blue LED will be connected to Arduino digital pin 10. The duty cycle of the Red LED will be 50%, with an on time of 200ms, and the duty cycle of the Blue LED will be 25%, with an on time of 400ms.

Circuit/SchematicDiagram



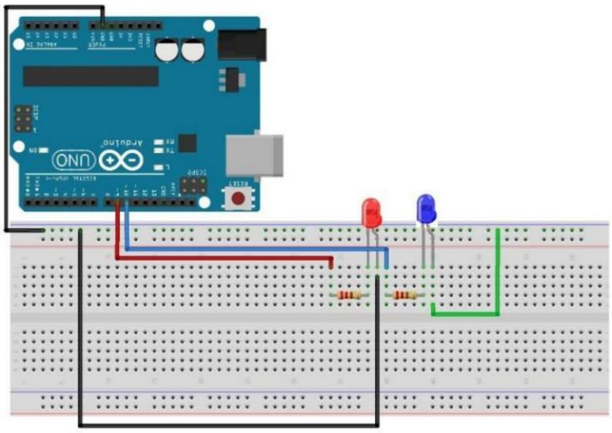


Figure8: TwoLEDCircuit

Figure9:SchematicoftwoLEDCircuit

Code :

4.1:Control Two LED Circuit without a loop

```
int redLED = 9;   // Red LED on pin 9
int blueLED = 10; // Blue LED on pin 10
void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  Serial.begin(9600); // Start serial monitor
}
void loop() {
  // ----- Red LED: Blink 5 times -----
  Serial.println("Red LED blinking 5 times...");
  digitalWrite(redLED, HIGH); delay(200); digitalWrite(redLED, LOW); delay(200); Serial.println("Red blink #1");
  digitalWrite(redLED, HIGH); delay(200); digitalWrite(redLED, LOW); delay(200); Serial.println("Red blink #2");
  digitalWrite(redLED, HIGH); delay(200); digitalWrite(redLED, LOW); delay(200); Serial.println("Red blink #3");
  digitalWrite(redLED, HIGH); delay(200); digitalWrite(redLED, LOW); delay(200); Serial.println("Red blink #4");
  digitalWrite(redLED, HIGH); delay(200); digitalWrite(redLED, LOW); delay(200); Serial.println("Red blink #5");
  delay(1000); // pause before blue LED
  // ----- Blue LED: Blink 3 times -----
  Serial.println("Blue LED blinking 3 times...");
  digitalWrite(blueLED, HIGH); delay(400); digitalWrite(blueLED, LOW); delay(1200); Serial.println("Blue blink
#1");
  digitalWrite(blueLED, HIGH); delay(400); digitalWrite(blueLED, LOW); delay(1200); Serial.println("Blue blink
#2");
  digitalWrite(blueLED, HIGH); delay(400); digitalWrite(blueLED, LOW); delay(1200); Serial.println("Blue blink
#3");
  delay(2000); // Pause before repeating
}
```

4.2:Control Two LED Circuit with a for loop

```
int redLED = 9;   // Red LED on pin 9
int blueLED = 10; // Blue LED on pin 10
void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
 Serial.begin(9600); // Start serial monitor
}
void loop() {
  // Blink Red LED 5 times
  Serial.println("Red LED blinking 5 times...");
   for (int i = 1; i <= 5; i++) {
   digitalWrite(redLED, HIGH);
   delay(200);            // ON for 200 ms
```

```
    digitalWrite(redLED, LOW);

    delay(200);             // OFF for 200 ms

   Serial.print("Red blink #");

  Serial.println(i);

  }
```

4.3 :Control Two LED Circuit with a for loop using the Serial monitor

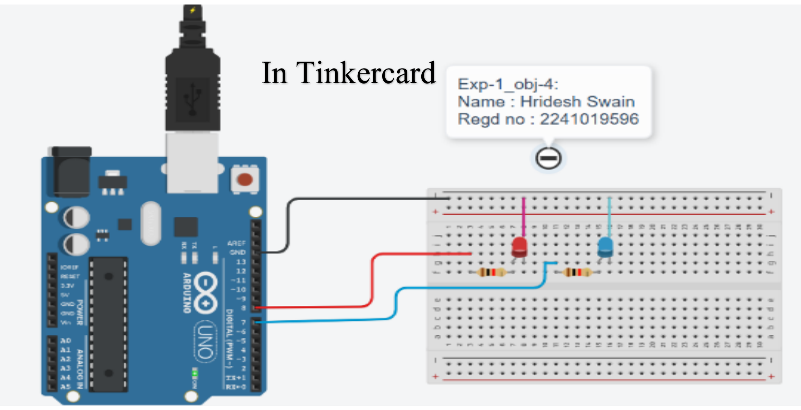
```
int redLED = 13;   // Red LED pin
int blueLED = 7; // Blue LED pin

void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  Serial.begin(9600); // Start serial communication
}
void loop() {
  // ----- Red LED blinks -----
  Serial.println("Red LED blinking 5 times...");
  for (int i = 1; i <= 5; i++) {
    digitalWrite(redLED, HIGH);
    delay(200);             // ON for 200ms
    digitalWrite(redLED, LOW);
    delay(200);             // OFF for 200ms
    Serial.print("Red blink #");
    Serial.println(i);
  }

  delay(1000); // short pause before switching to Blue

  // ----- Blue LED blinks -----
  Serial.println("Blue LED blinking 3 times...");
  for (int j = 1; j <= 3; j++) {
    digitalWrite(blueLED, HIGH);
    delay(400);             // ON for 400ms
    digitalWrite(blueLED, LOW);
    delay(1200);            // OFF for 1200ms
    Serial.print("Blue blink #");
    Serial.println(j);
  }

  delay(2000); // Pause before repeating sequence
}
```

4.4 :Control Two LED Circuit with User Input using Arduino and Serial Monitor

```
int led1 = 8;  // LED 1 on pin 8
int led2 = 9;  // LED 2 on pin 9

void setup() {
  Serial.begin(9600);   // Start Serial Monitor
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  Serial.println("Type a command:");
  Serial.println("'1' = LED1 ON, '2' = LED1 OFF");
  Serial.println("'3' = LED2 ON, '4' = LED2 OFF");
  Serial.println("'5' = Both ON, '6' = Both OFF");
}

void loop() {
  if (Serial.available() > 0) {
    char input = Serial.read();

    switch (input) {
      case '1':
        digitalWrite(led1, HIGH);
        Serial.println("LED1 ON");
        break;
      case '2':
        digitalWrite(led1, LOW);
        Serial.println("LED1 OFF");
        break;
      case '3':
        digitalWrite(led2, HIGH);
        Serial.println("LED2 ON");
        break;
      case '4':
        digitalWrite(led2, LOW);
        Serial.println("LED2 OFF");
        break;
      case '5':
        digitalWrite(led1, HIGH);
        digitalWrite(led2, HIGH);
        Serial.println("Both LEDs ON");
        break;
      case '6':
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        Serial.println("Both LEDs OFF");
        break;
      default:
        Serial.println("Invalid command! Type 1–6.");
        break;
    }
  }
}
```

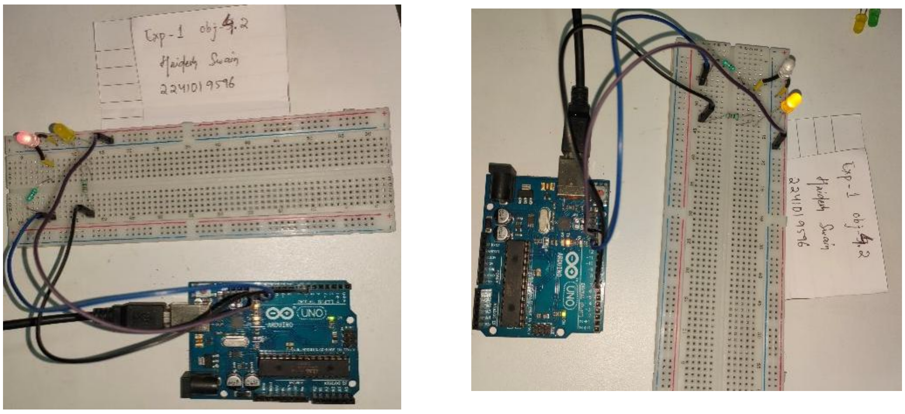
Observation :

Figure10:(Simulation based Two LED Circuit) :



Exp-1_obj-4:
Name : Hridesh Swain
Regd no : 2241019596

Figure11 : (Hardware Implementation based Two LED Circuit)



Objective5

Design and implement an LED chaser using different lighting patterns.

Circuit/Schematic Diagram

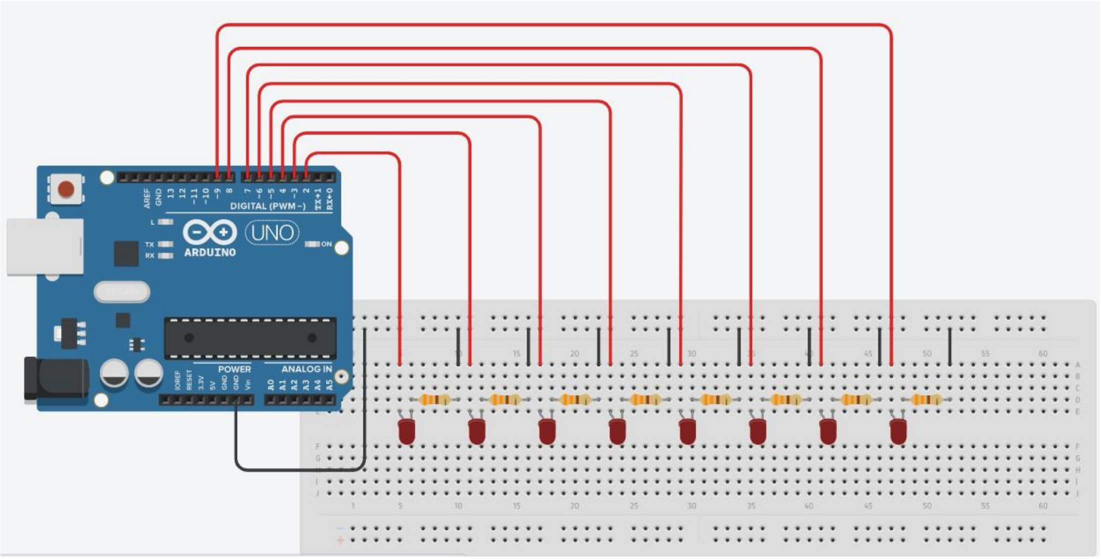


Figure12:Schematicof8-bitLEDchasercircuit

Code

5.1 Build and Check the 8-bit LED chaser circuit : (Verify that all the LEDs light up, indicating that the circuit is connected properly)

```
// 8-bit LED chaser with verification step
const int pinList[8] = {2, 3, 4, 5, 6, 7, 8, 9}; // LED pins D2..D9
const unsigned long verifyMillis = 2000; // how long to show all ON (ms)
const unsigned int chaserDelay = 150;    // delay between steps (ms)

void setup() {
  Serial.begin(9600);
  Serial.println("8-bit LED Chaser - Verify then run");

  // configure pins
  for (int i = 0; i < 8; i++) {
    pinMode(pinList[i], OUTPUT);
    digitalWrite(pinList[i], LOW); // start OFF
  }
  // Verification step: turn all ON briefly
  Serial.println("Verification: turning ALL LEDs ON for 2 seconds...");
  for (int i = 0; i < 8; i++) digitalWrite(pinList[i], HIGH);
  delay(verifyMillis);

  // Turn all OFF, confirm
```

```
 for (int i = 0; i < 8; i++) digitalWrite(pinList[i], LOW);
  delay(300);
  Serial.println("Verification complete. Starting chaser...");
}
void loop() {
  // Simple forward chaser (1 -> 8)
  for (int i = 0; i < 8; i++) {
    digitalWrite(pinList[i], HIGH);
    delay(chaserDelay);
    digitalWrite(pinList[i], LOW);
  }
  // Reverse chaser (8 -> 1)
  for (int i = 7; i >= 0; i--) {
    digitalWrite(pinList[i], HIGH);
    delay(chaserDelay);
    digitalWrite(pinList[i], LOW);
  }
  // small pause between cycles
  delay(200);
}
```

5.2 Implement a basic circular LED chaser pattern.: (8LEDs connected from pins 2 to 9)

```
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
int numLeds = 8;
void setup() {
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
  Serial.begin(9600);
}
void loop() {
  Serial.println("Pattern 1: Bouncing Ball");
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(200);
    digitalWrite(ledPins[i], LOW);
  }
}
```

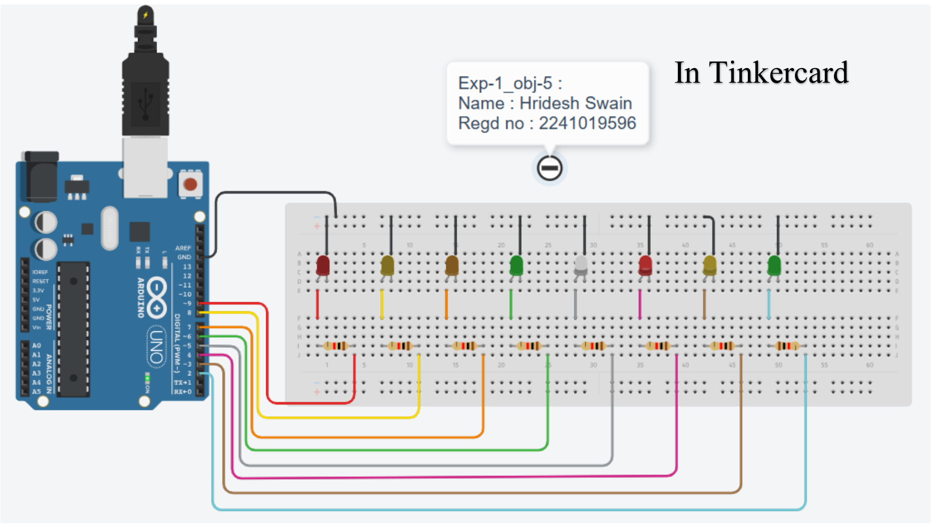
5.3 Implement a custom LED chaser pattern: (that combines five different patterns in a loop, including the "Bouncing Ball," "Zig-Zag," "Random Blink," "Knight Rider" and "Running Lights,")

```
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
int numLeds = 8;
void setup() {
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
  Serial.begin(9600);
}
void loop() {
  bouncingBall();
  zigZag();
  randomBlink();
  knightRider();
  runningLights();
}
```

```
// --- Pattern 1: Bouncing Ball ---
void bouncingBall() {
  Serial.println("Pattern 1: Bouncing Ball");
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(100);
    digitalWrite(ledPins[i], LOW);
  }
  for (int i = numLeds - 2; i > 0; i--) {
    digitalWrite(ledPins[i], HIGH);
    delay(100);
    digitalWrite(ledPins[i], LOW);
  }
}

// --- Pattern 2: Zig-Zag ---
void zigZag() {
  Serial.println("Pattern 2: Zig-Zag");
  for (int i = 0; i < numLeds - 1; i += 2) {
    digitalWrite(ledPins[i], HIGH);
    digitalWrite(ledPins[i + 1], HIGH);
    delay(200);
    digitalWrite(ledPins[i], LOW);
    digitalWrite(ledPins[i + 1], LOW);
  }
  for (int i = numLeds - 1; i > 0; i -= 2) {
    digitalWrite(ledPins[i], HIGH);
    digitalWrite(ledPins[i - 1], HIGH);
    delay(200);
    digitalWrite(ledPins[i], LOW);
    digitalWrite(ledPins[i - 1], LOW);
  }
}
// --- Pattern 3: Random Blink ---
void randomBlink() {
  Serial.println("Pattern 3: Random Blink");
  for (int i = 0; i < 10; i++) { // blink 10 random times
    int randLED = random(0, numLeds);
    digitalWrite(ledPins[randLED], HIGH);
    delay(150);
    digitalWrite(ledPins[randLED], LOW);
    delay(100);
  }
}
// --- Pattern 4: Knight Rider ---
void knightRider() {
  Serial.println("Pattern 4: Knight Rider");
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    if (i > 0) digitalWrite(ledPins[i - 1], LOW);
    delay(120);
  }
  for (int i = numLeds - 2; i >= 0; i--) {
    digitalWrite(ledPins[i], HIGH);
    if (i < numLeds - 1) digitalWrite(ledPins[i + 1], LOW);
    delay(120);
  }
}
// --- Pattern 5: Running Lights ---
void runningLights() {
  Serial.println("Pattern 5: Running Lights");
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(150);
    digitalWrite(ledPins[i], LOW);
  }
}
```

Observation

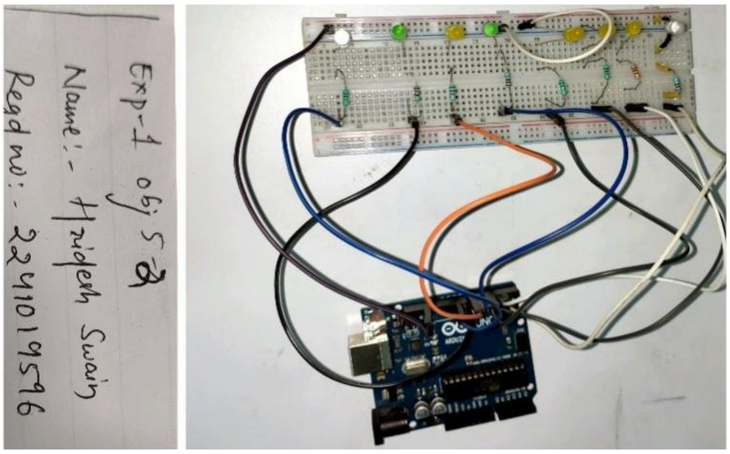
Figure13: (Simulation based 8-bit LED chaser circuits)



Exp-1_obj-5 :
Name : Hridesh Swain
Regd no : 2241019596

In Tinkercard

Figure14:(Hardware Implementation based 8-bit LED chaser circuits)



Exp-1 obj 5 &
Name:- Hridesh Swain
Regd no:- 2241019596

Conclusion:

Through these experiments, we learned to control LEDs using Arduino with different duty cycles, loops, and patterns. The lab enhanced our understanding of digital outputs, timing, and serial communication, which are essential for robotics and embedded systems. Overall, the lab provided hands-on experience with digital output, timing control, and programming structures in Arduino. It improved our understanding of hardware–software interaction, loop structures, serial communication, and circuit design. These concepts form the foundation for more advanced robotics and embedded system applications, where LEDs are often used as indicators for debugging, feedback, and system status

Precautions:

- Always connect a **current-limiting resistor** with LEDs to prevent damage.
- Double-check all **wiring connections** before powering the Arduino.
- Avoid connecting LEDs directly to **5V/GND** without control through Arduino pins.
- Ensure correct use of **delay values** to avoid unexpected LED behavior.
- Handle the Arduino board carefully to prevent **short circuits or overheating**

Post Experiment Questionnaire:

A. Experiment-Specific (LED Blinking & LED Patterns)

1. What differences did you observe between blinking an LED using a global variable versus without a global variable?
2. How did varying the duty cycle (25%, 50%, 75%) affect the LED's brightness and blink timing?
3. Compare the LED behavior when controlled without a loop vs. with a for loop. Which method is more efficient and why?
4. When using the Serial Monitor for user input, what challenges did you face in reading and processing the input values?
5. How did the custom LED chaser patterns (e.g., Knight Rider, Random Blink) differ in implementation compared to the basic circular LED chaser?

B. Arduino & Hardware Concepts

6. Why is it necessary to use resistors with LEDs in Arduino circuits?
7. What role do digital pins play in controlling multiple LEDs simultaneously?
8. Explain how PWM is used in LED brightness control. Could this method be extended to motor control?
9. How does the Arduino distinguish between different LED patterns in the sketch (e.g., sequential vs. random)?

C. Robotics & System Understanding

10. Why are blinking LEDs often used as a first step in learning robotics and embedded systems?
11. In what ways can LED indicators assist in debugging larger robotic systems?
12. If you were to expand this experiment into a real robotic system, how might LED patterns be used for status indication or fault detection?

Answers to Post-Lab Questions

(Signature of theFaculty)

(Signature of the Student)

Name: **Hridesh Swain**

Date:

Registration No.: **2241019596**

Branch: **CSE**

Section : **2241021**