

**Lab Report-02**

**Course Name:** Internet of Things Lab

**Course Code:** CSE 402

**Submitted To**

**S.M. Zafrul Islam**

**Lecturer, Department of CSE, UITS**

**Submitted By**

**Name:** Yeasmin Kabir Keya

**ID:** 0432220005101107

**Batch:** 52

**Section:** B

**Experiment no: 02**

**Experiment Name:** Controlling the LED blink rate with the potentiometer interfacing with

Arduino.

**Objectives**

* Develop an Arduino program to control the blinking speed of an LED using a potentiometer.
* Read analog input values from the potentiometer to dynamically adjust the LED’s blink delay.
* Interface the potentiometer with the Arduino’s analog input pin and connect the LED to a digital output pin with proper current limiting.
* Demonstrate real-time control of the LED’s blink rate by varying the potentiometer position.

**Introduction**

Arduino Uno is a popular open-source microcontroller board based on the ATmega328P microcontroller. It has 14 digital input/output pins (6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and a reset button. The Arduino Uno is widely used in embedded projects due to its simplicity and versatility.

**Pin Diagram**

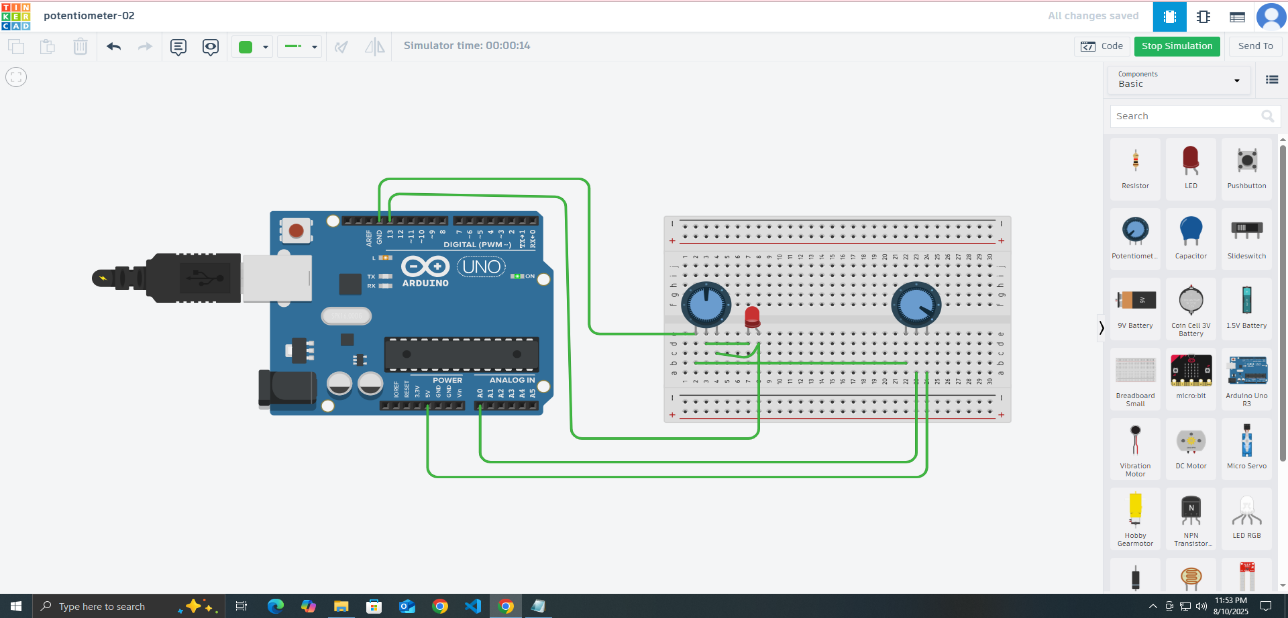
* Digital pins: D0 to D13
* Analog pins: A0 to A5
* Power pins: 5V, 3.3V, GND
* Reset pin

The Arduino Uno can read analog signals from sensors like a potentiometer through its analog pins and control actuators like LEDs connected to its digital pins.

**Apparatus Required**

* Arduino Uno
* LED
* Potentiometer
* Resistor
* Breadboard
* Connecting wires

**Circuit Diagram**

****

**Working Procedure**

* Connect the potentiometer to 5V, GND, and A0 pin of Arduino.
* Connect the LED with a current-limiting resistor to pin 9 and GND.
* Upload the Arduino sketch that reads the analog value from the potentiometer.
* Map the analog input (0-1023) to a delay time suitable for LED blinking.
* Use the mapped delay to blink the LED on and off.
* Turn the potentiometer knob and observe the change in blink rate of the LED.

**Code**

// C++ code //

const int ledPin=13;

const int regPin=A0;

int sensorValue;

void setup()

{ pinMode(regPin, INPUT);

pinMode(ledPin, OUTPUT);

}

void loop()

{

sensorValue=analogRead(regPin);

digitalWrite(ledPin, HIGH);

delay(sensorValue); // Wait for 1000 millisecond(s) digitalWrite(ledPin, LOW);

delay(sensorValue); // Wait for 1000 millisecond(s)

}

**Discussion and Conclusion**

In this experiment, the Arduino successfully reads the analog input from a potentiometer and adjusts the LED's blink rate accordingly. The use of the map() function enables converting the raw analog input (0 to 1023) into a meaningful delay range (100 to 1000 milliseconds).

The LED blink rate varies smoothly as the potentiometer knob is rotated, demonstrating real-time control through hardware input. The current-limiting resistor effectively protects the LED from excessive current.