University of Information Technology & Sciences

Department of
Computer Science and Engineering

Lab Report 02

Course Title: Internet Of Things Lab
Course Code: CSE-402

Submitted To

S. M. Zafrul Islam
Lecturer of CSE Department, UITS

Submitted By

Name : Mobarok Hossain Zobaer
Id : 0432220005101029
Batch : 52
Semester: Autumn 25
Section : 7A

Experiment No. 02
Experiment Title: Controlling the LED blink rate with the potentiometer interfacing with Arduino

Objectives

- Write an Arduino program to interface a potentiometer and LED.
- Use the analog input from a potentiometer to control the blink rate of the LED.
- Understand the use of analogRead() and delay() functions in Arduino.
- Learn how to build and test a basic Arduino circuit on a breadboard.

Introduction

Arduino Uno is an open-source microcontroller board based on the ATmega328P. It has 14 digital I/O pins, 6 analog input pins, a 16 MHz quartz crystal, a USB connection, a power jack, and a reset button.

The pin diagram of Arduino Uno shows:

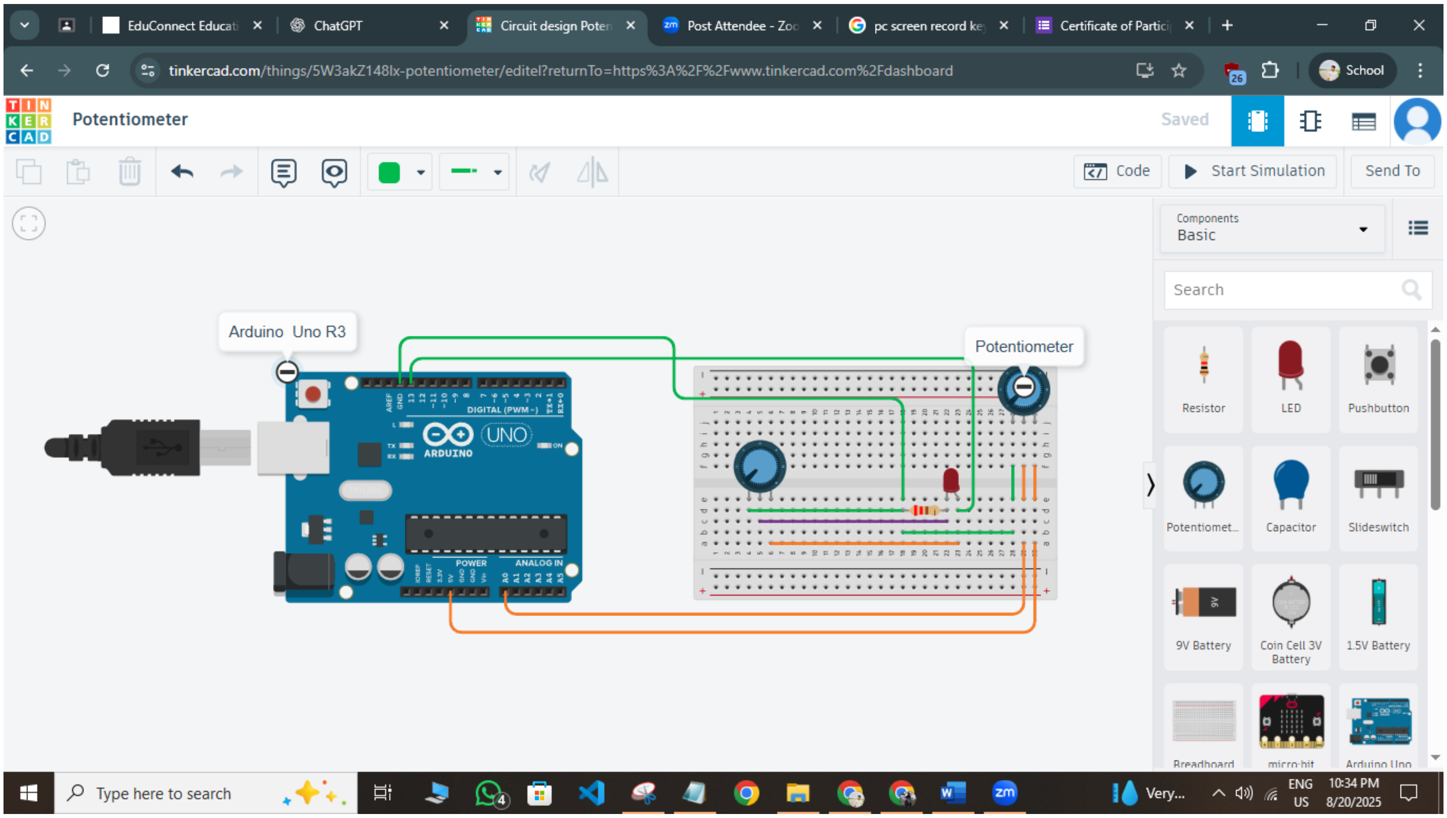
- Digital Pins (0–13): Used for input/output, some with PWM capability.
- Analog Pins (A0–A5): Used to read analog values (0–1023).
- Power Pins: Provide 3.3V, 5V, and GND for external components.

In this experiment, we use a potentiometer connected to pin A0 as an analog input device. The resistance of the potentiometer changes when rotated, and Arduino reads this value to adjust the delay time for the LED connected to pin 13, thus controlling the blink rate.

Apparatus Required

Component	Quantity	Description
Arduino Uno R3	1	Microcontroller board
Breadboard	1	For circuit assembly
LED	1	Output indicator
Resistor (220Ω)	1	To limit current for LED
Potentiometer	1	Variable resistor for input
Jumper Wires	As req.	For connections
USB Cable	1	To connect Arduino to PC

Circuit Diagram



Working Procedure

- Connect the LED to digital pin 13 of Arduino through a 220Ω resistor.
- Connect the potentiometer middle terminal (wiper) to analog input pin A0.
- Connect the other two potentiometer terminals to 5V and GND.
- Upload the given Arduino code to the board.
- When the potentiometer knob is rotated, Arduino reads different analog values (0–1023).
- The read value is used as the delay time in milliseconds, which increases or decreases the LED blink rate.

Code

```
// C++ code
const int ledpin = 13;
const int regpin = A0;
int value;

void setup()
{
  pinMode(ledpin, OUTPUT);
  pinMode(regpin, INPUT);
}

void loop()
{
  value = analogRead(regpin);   // Read potentiometer value (0–1023)

  digitalWrite(ledpin, HIGH);   // Turn LED ON
  delay(value);                 // Delay based on potentiometer

  digitalWrite(ledpin, LOW);    // Turn LED OFF
  delay(value);                 // Delay based on potentiometer
}
```

Discussion and Conclusion

In this experiment, the potentiometer successfully controlled the LED blinking rate. When the potentiometer resistance was low, the delay was short, and the LED blinked faster. When the resistance was high, the delay increased, and the LED blinked slower.

This experiment demonstrates how analog inputs can be used to control digital outputs in Arduino. It also provides hands-on understanding of analog-to-digital conversion (ADC) and timing functions (delay()).

Final Verdict: The experiment achieved its objective of controlling the LED blink rate with a potentiometer using Arduino Uno.