"Heaven's light is our guide"

Rajshahi University of Engineering & Technology

Department of Electrical & Electronic Engineering

Course No : EEE 4210

Course Title : Embedded System Design Sessional

Experiment no : 03

Name of the Experiment: Study of Basic Arduino Programming using Tinkercad

Date of Experiment: November 22, 2023

Date of Submission: December 06, 2023

Submitted By

Md Maruf Hassan

Department : EEE

Roll : 1801105

Section : B

Submitted To

Md Mayenul Islam

Lecturer,

Department of EEE,

RUET

3.1 Experiment No: 03

3.2 Experiment Name: Study of Basic Arduino Programming using Tinkercad

3.3 Objectives:

- Get familiar with Tinkercad and Arduino UNO.
- Make projects with Arduino UNO in Tinkercad platform.

3.4 Theory:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The microcontroller board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by a USB cable or a barrel connector that accepts voltages between 7 and 20 volts, such as a rectangular 9-volt battery.

Microcontroller: At the heart of the Arduino Uno is the Atmel ATmega328P microcontroller. This chip contains the central processing unit (CPU) of the board and is responsible for executing the instructions of the program loaded onto the Arduino.

Clock Crystal: The Arduino Uno is equipped with a crystal oscillator that provides the clock signal to the microcontroller. The clock speed is typically 16 MHz

Voltage Regulator: The board includes a voltage regulator, often an AMS1117 or a similar component, that regulates the incoming voltage to a stable 5 volts. This allows the Arduino Uno to be powered by a range of voltages (typically 7-12V) through the power jack or via the USB port.

Power Supply Section: The Arduino Uno can be powered through either the DC power jack or the USB connection. The power supply section includes components to manage the power input, such as protection diodes.

USB Interface: The USB-to-Serial converter, often an ATmega16U2 or CH340, allows the Arduino Uno to communicate with a computer over USB. This is used for both programming the board and for serial communication with the computer.

Reset Circuitry: There is a reset button on the board that, when pressed, resets the microcontroller. The reset circuitry also includes a pull-up resistor.

Digital and Analog I/O Pins: The board features digital input/output pins, some of which can also function as PWM (Pulse Width Modulation) outputs, and analog input pins. These pins are where external components can be connected to interact with the Arduino Uno.

LEDs: There are a few LEDs on the board for visual indications. The "ON" LED lights up when the board is powered, the "L" LED is connected to digital pin 13 and is often used for basic testing and debugging, and the "TX" and "RX" LEDs indicate data transmission over USB.

Headers and Connectors: The Arduino Uno has female headers for connecting external components, shields, or jumper wires. These headers provide a convenient way to connect and disconnect components without soldering.

3.5 Setup and Observation:

Blinking an LED is a common beginner's project when working with the Arduino Uno. An LED will be blink for 1sec, then it will be of for 1sec and then again blink and it will be continued. The bended terminal of LED is the anode and the straight one is the cathode. Connect the longer leg (anode) to digital pin 2 and shorter leg (cathode) of the LED to the ground (GND) pin on the Arduino.

Module 02 – 01:

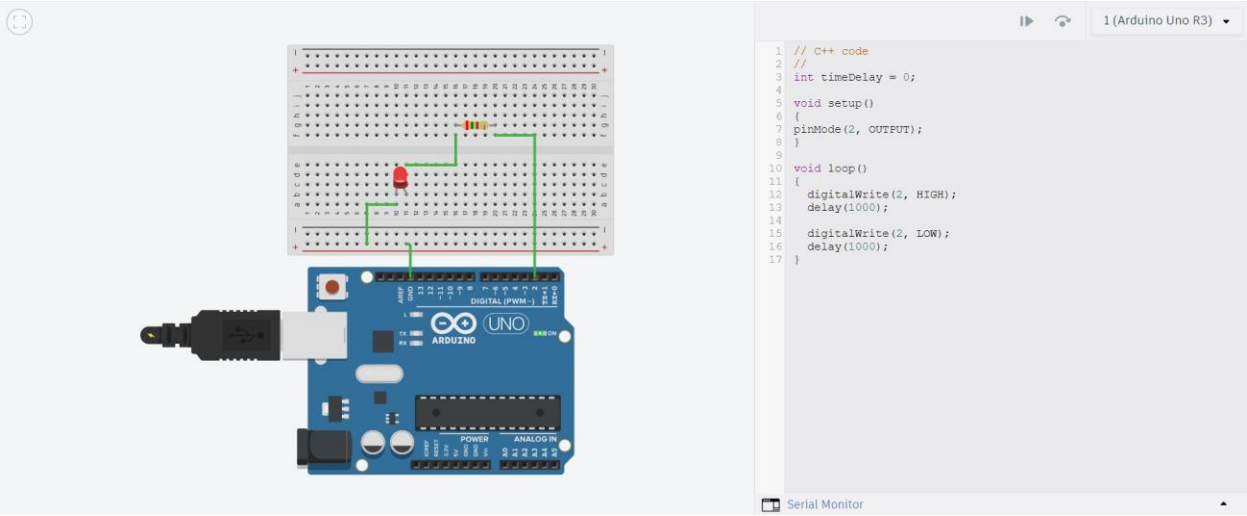

Figure 01: Single LED blinking.

Module 02 – 02:

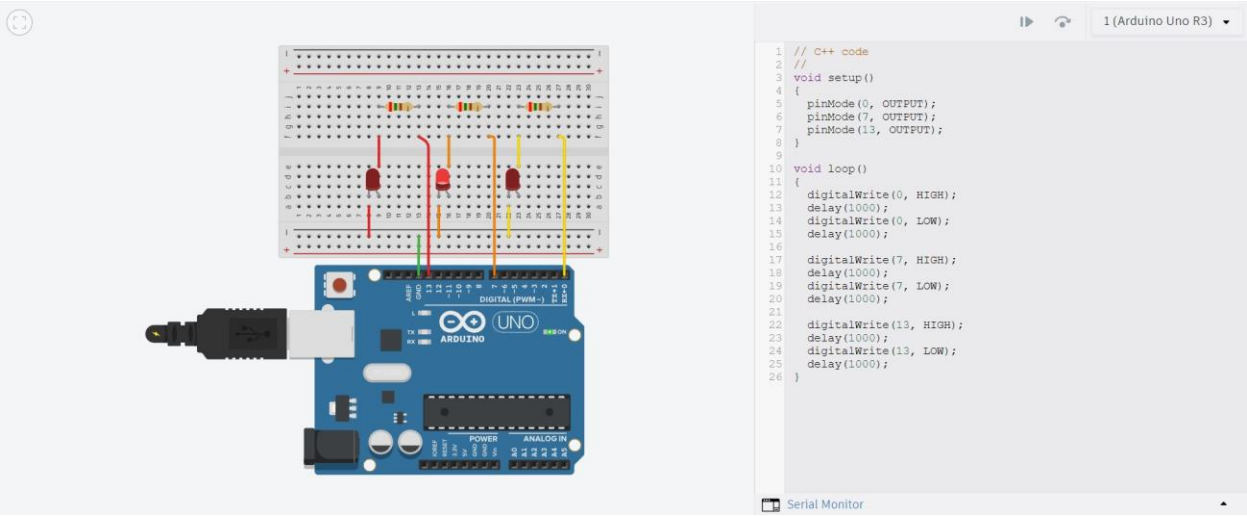

Figure 02: Multiple LED blinking.

A current-limiting resistor is used to protect the LED. A typical value of this resistor is around 220-330 ohms. Serial monitor. Serial. Begin (9600) is used to see serial monitor and Serial.println(value) is used to print values in serial monitor. Potentiometer has 3 terminals, terminals 1 and 2 are connected to 5V and GND pins respectively and middle terminal (wiper) works as output terminal.

Module 03 – 01:

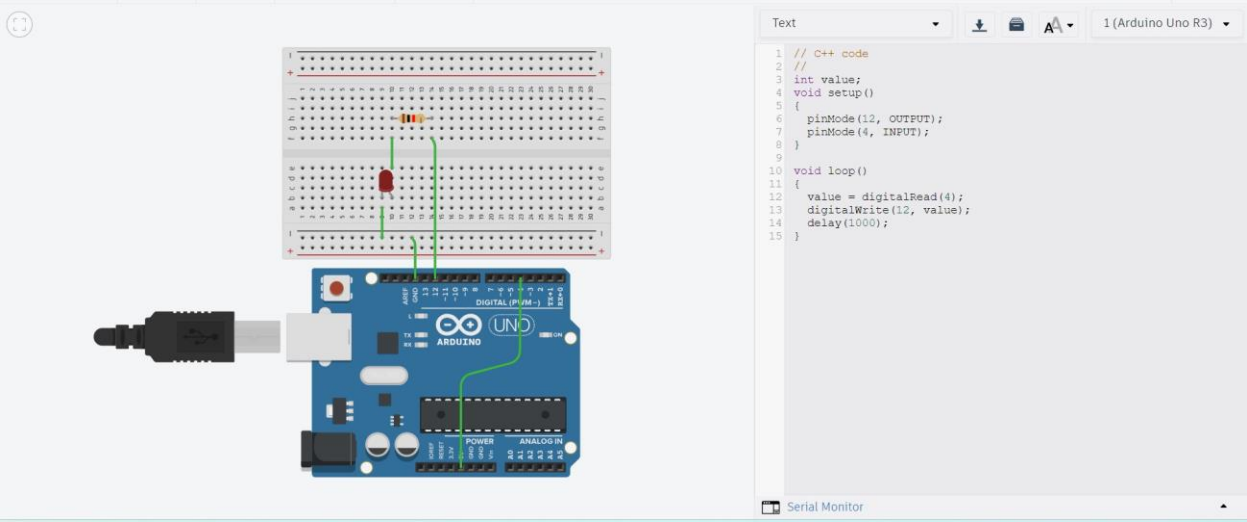

Figure 03: LED blinking with digital input.

The Serial Monitor in the Arduino IDE is a valuable tool for communication between your Arduino board and your computer. It allows you to send and receive data, making it useful for debugging, monitoring sensor readings, and interacting with Arduino projects.

Module 03 – 02:

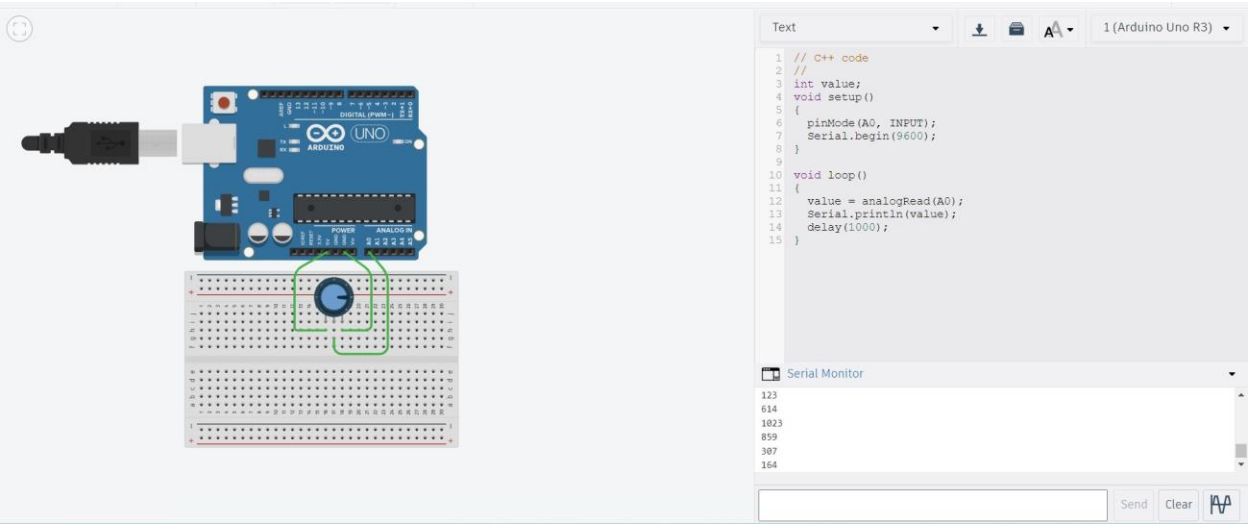

Figure 04: Analog input and serial monitor.

In Arduino programming, use the ‘if-else’ statement to create conditional logic. If the condition is true, it executes the ‘if’ block, if the condition is false then it executes the ‘else’ block. The syntax is similar to the C/C++ programming language.

Module 04:

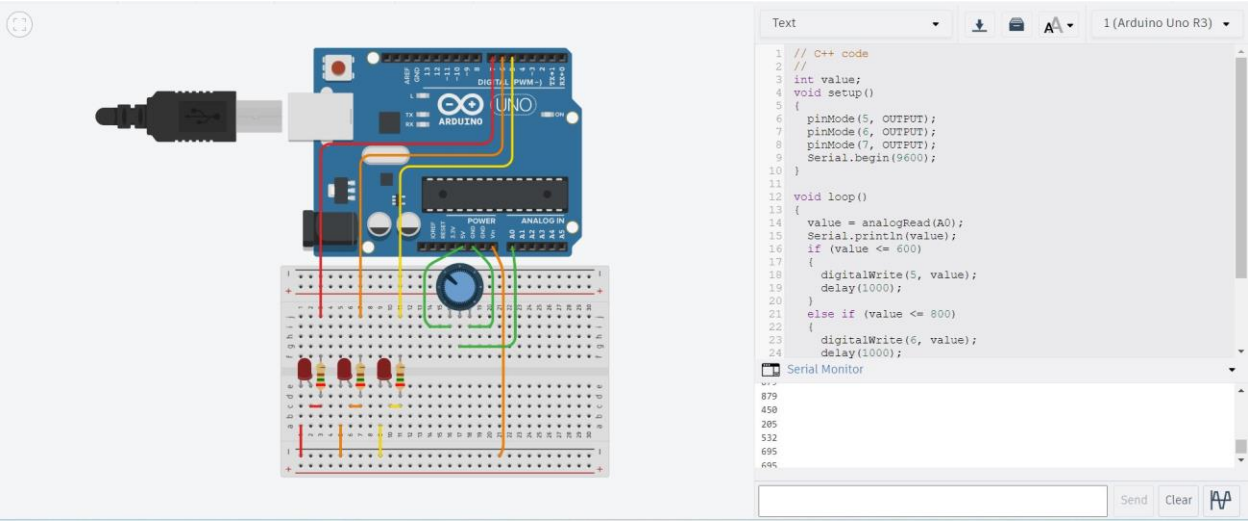

Figure 05: Implementation of ‘if – else’ condition.

The ‘for’ loop is commonly used to repeat a block of code a specific number of times. It executes ‘for’ block for the given iteration number. The syntax of the ‘for’ loop is similar to that of C and C++ programming languages.

Module 05 – 01:

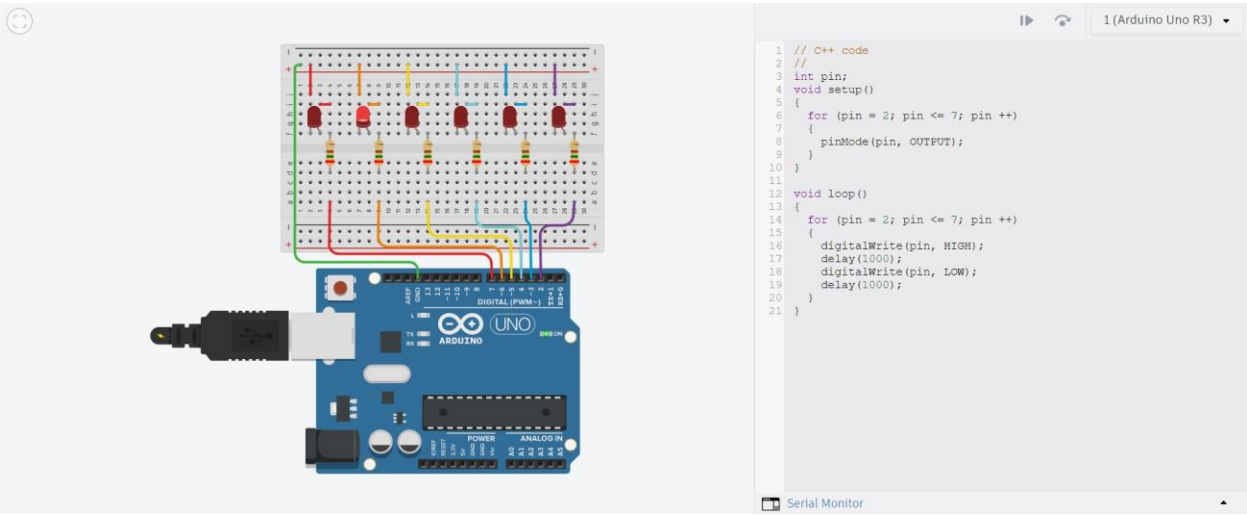

Figure 06: Implementation of ‘for’ loop.

Controlling brightness with Arduino can be done using Pulse Width Modulation (PWM). PWM is a technique that simulates an analog output using digital means, allowing you to control the average voltage applied to a device, such as an LED, and thus control its brightness.

PWM is only effective for controlling devices that respond to variable voltage levels, such as LEDs. If you need to control the speed of a motor, the position of a servo, or other similar applications, PWM can be a useful technique.

Module 05 – 02:

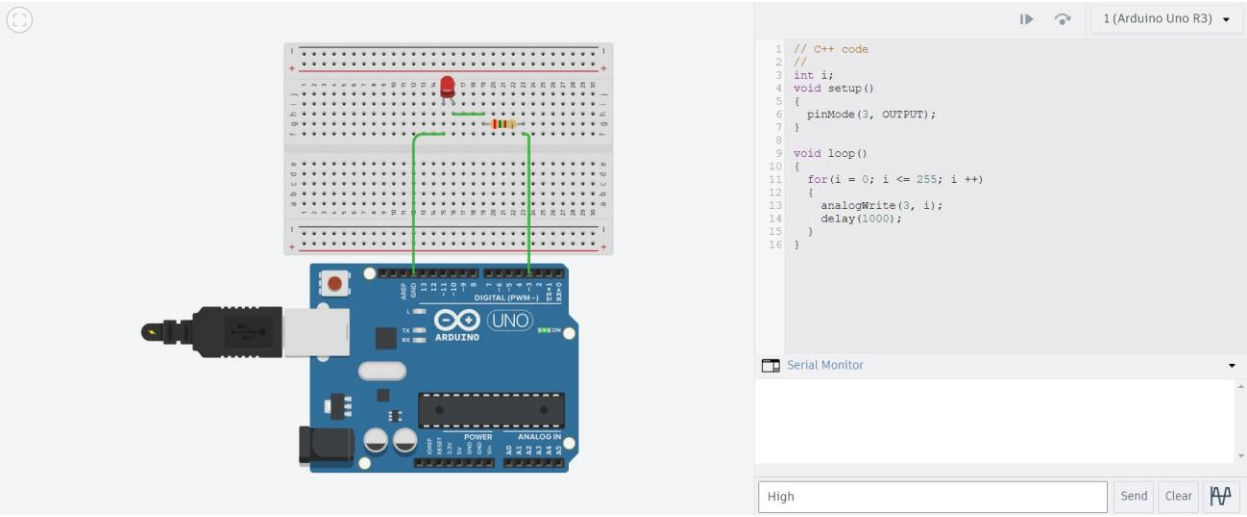

Figure 07: Brightness control with Arduino.

In Arduino programming, arrays used to store and manipulate collections of data of the same type. Arrays provide a convenient way to work with sets of variables. Arrays can be of various types (e.g., int, float, char), and you can use multidimensional arrays for more complex data structures.

Module 06:

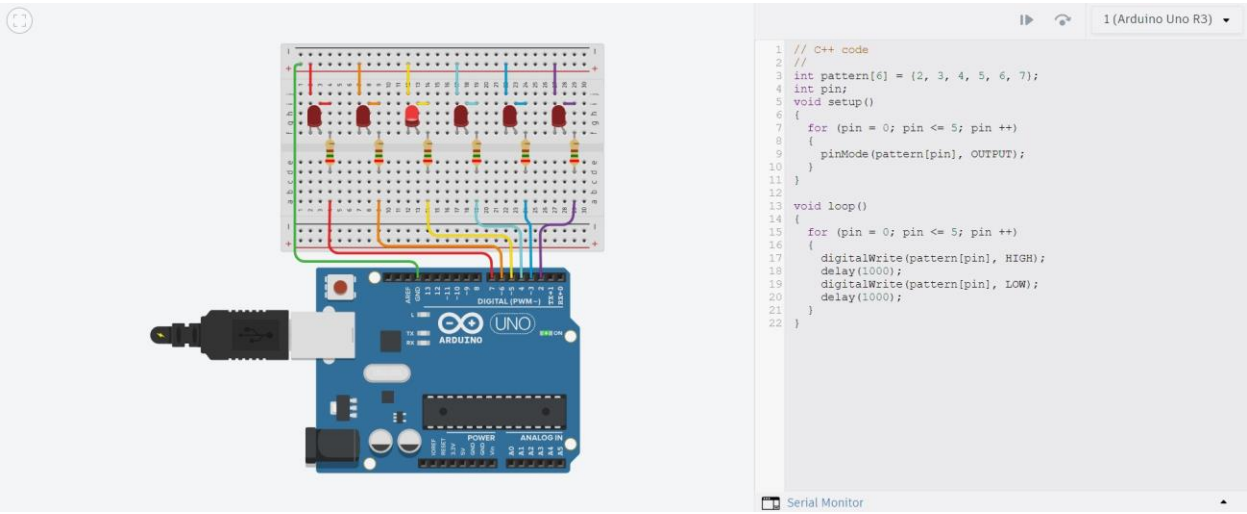

Figure 08: Implementation of ‘Array’ using Arduino.

A Liquid Crystal Display is a flat-panel display technology that uses the light-modulating properties of liquid crystals. LCDs are widely used in a variety of electronic devices. Using an LCD with Arduino is a common practice for displaying information in projects. The most popular type of LCD used with Arduino is the 16x2 character LCD.

Module 07 – 01:

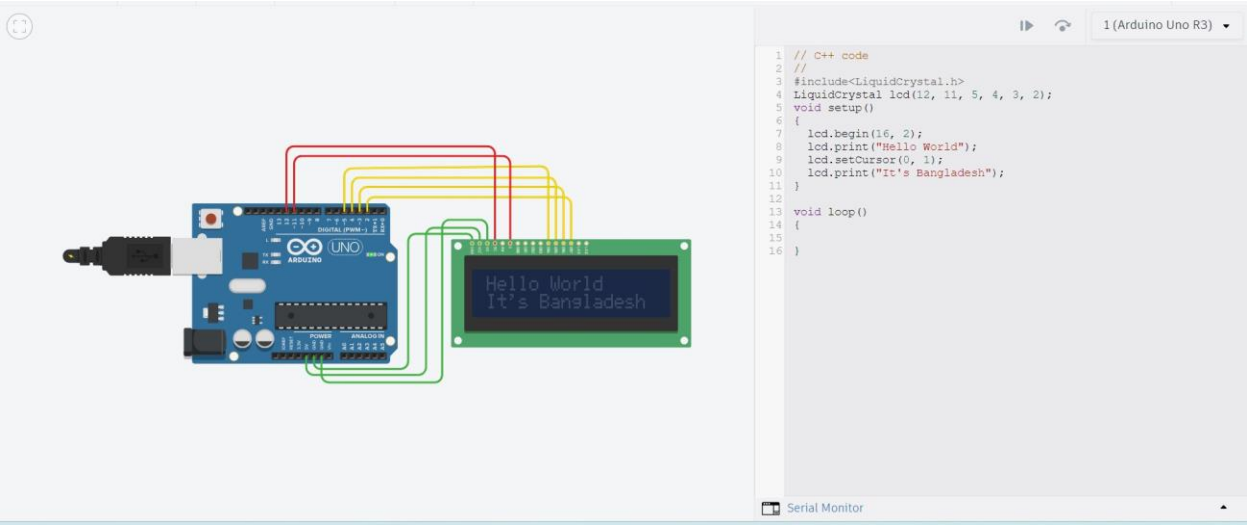

Figure 09: Connection with LCD display.

It customizes the messages, positioning, and timing according to project's needs. The lcd.setCursor() function is used to set the position on the LCD where the text will be displayed.

LCD displays can be used for various purposes, such as displaying sensor data, messages, or real-time information in your Arduino projects.

Module 07 – 02:

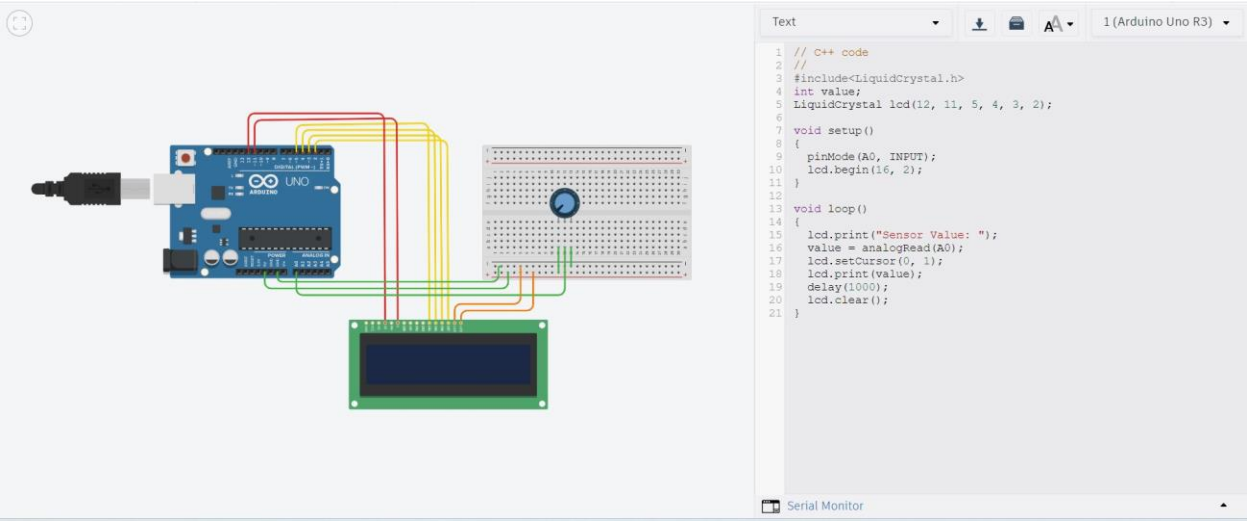

Figure 10: Displaying a sensor value.

A servo motor is a rotary or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It is widely used in various applications where controlled motion is required, such as in robotics, automation, aerospace, and manufacturing. Controlling a servo motor with Arduino is a popular and straightforward project. Servo motors have three wires: power (VCC), ground (GND), and control signal.

Module 08 – 01:

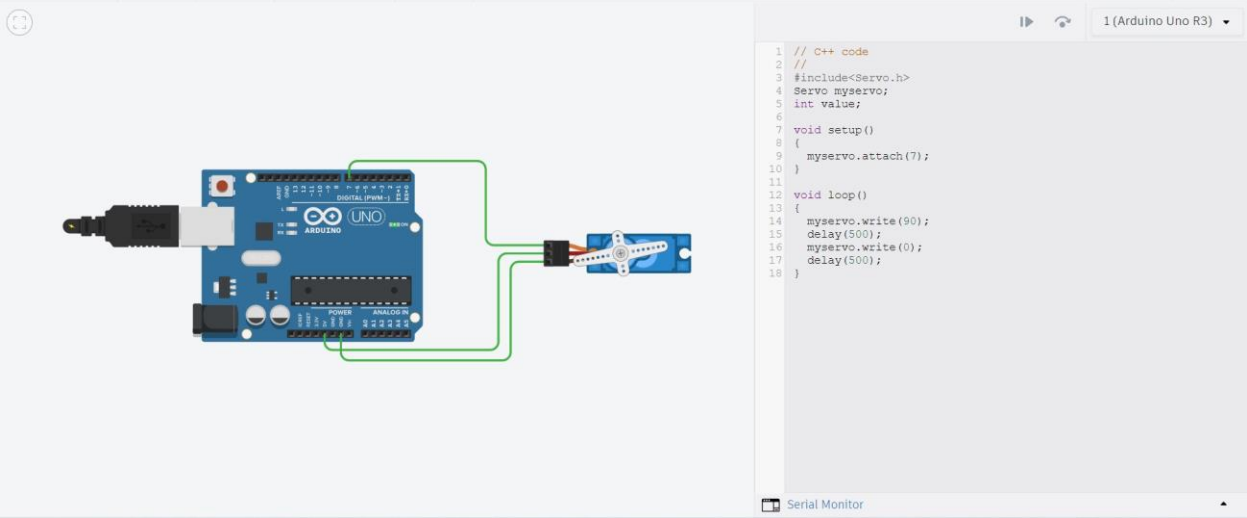

Figure 11: Servo motor control.

A servo motor consists of several key components that work together to enable precise control of position, speed, and acceleration. Here's a basic overview of the construction of a typical servo motor: DC motor, gear train, control circuit, potentiometer, positional sensor, output shaft, housing and mounting, wires and conductors etc.

Module 08 – 02:

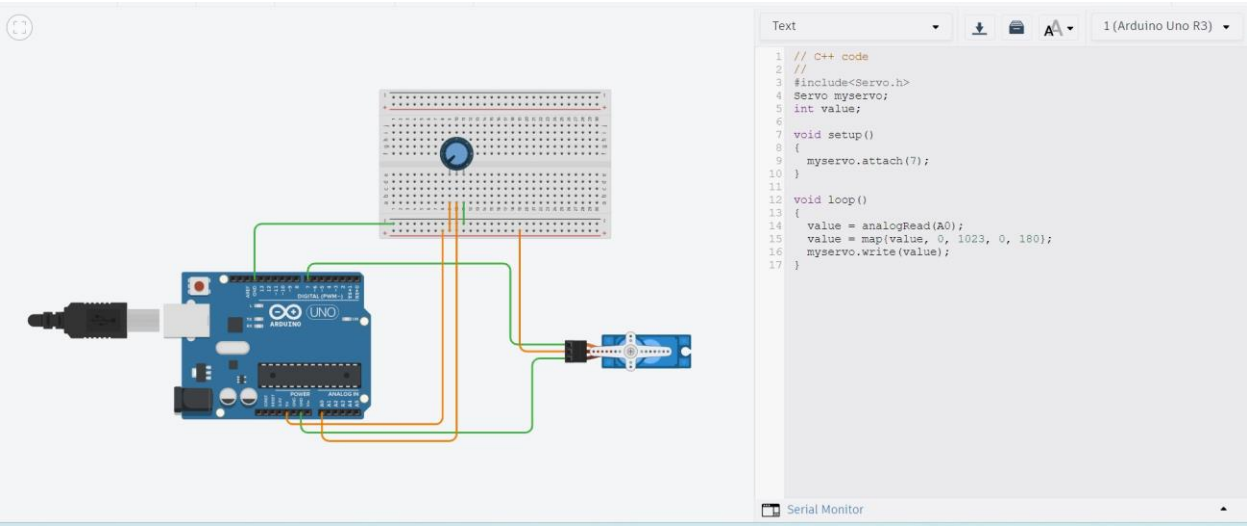

Figure 12: Servo motor control with analog input.

3.6 Discussion and Conclusion:

Arduino is an open-source electronics platform that comprises both hardware and software components, making it accessible to individuals without extensive technical backgrounds. The platform includes Arduino boards equipped with microcontrollers, input/output pins, and a USB interface.

Tinkercad is a web-based platform designed for computer-aided design and 3D modeling, aimed at beginners and educational purposes. Tinkercad provides a user-friendly interface that allows users to create and design 3D models without the need for extensive experience in CAD software. Tinkercad also offers a circuit design platform, providing users with the ability to simulate and experiment with electronic circuits in a virtual environment.

In this experiment, we designed some projects in Tinkercad with Arduino UNO. As a peripheral devise LEDs, potentiometer, LCD display and servo motor is used. Arduino code is provided to control their operations. I have built the design according to the lab manual. Though some different scheme is used for better connection among the equipment. Also modified some code for better control and output.

Code:

```
CLR P3.4
SETB P3.3

CLR P1.0
CLR P1.2
CLR P1.3
CLR P1.5
CLR P1.6

LOOP:
  MOV A, #0FFH
  DEC A
  JZ LOOP1

LOOP1:
  SETB P3.4
  SETB P3.3
  CLR P1.0
CLR P1.1
 CLR P1.4
SETB P1.6
```

Output:

(a)

(b)

Fig. 02: Display output showing last two digit 1801105

Discussion and Conclusion:

In this experiment, a number of activities involving the LEDs and the 7-segment display were carried out once the interface connection was observed. The preceding section displays the associated programs and their results. The outputs were produced in line with the intended outcomes. Therefore, it can be concluded that the observation of the 8051 microcontroller's LED and 7-segment display interface using the edsim51 program was successful.