

EXPERIMENT-01

Aim: Control the LED with Arduino Board and tinkercad software

Objectives: To get the knowledge of Arduino Board and control of output device (LED)

Outcomes: We will be able to write a program using Arduino IDE for Blinking LED.

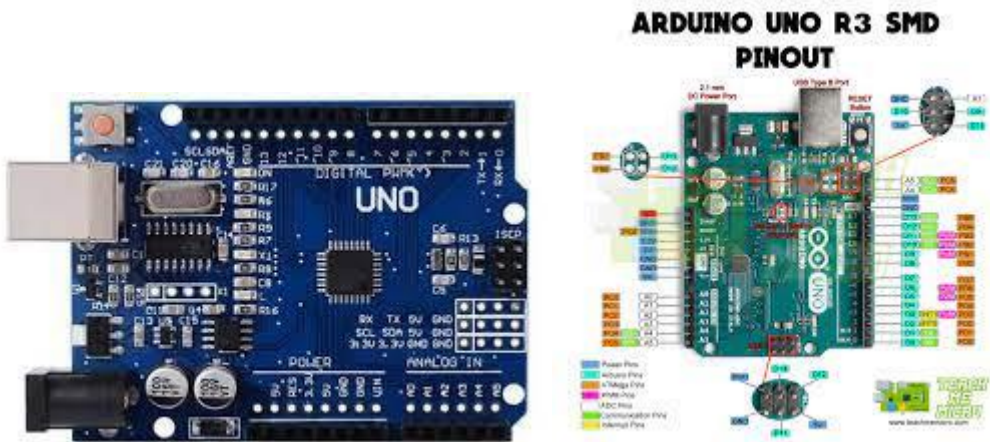
Hardware Requirements:

1. 1x Breadboard
2. 1x Arduino Uno
3. 1x LED
4. 1x 330Ω Resistor
5. 2x JumperWires

Theory:

[Arduino Uno:](#)

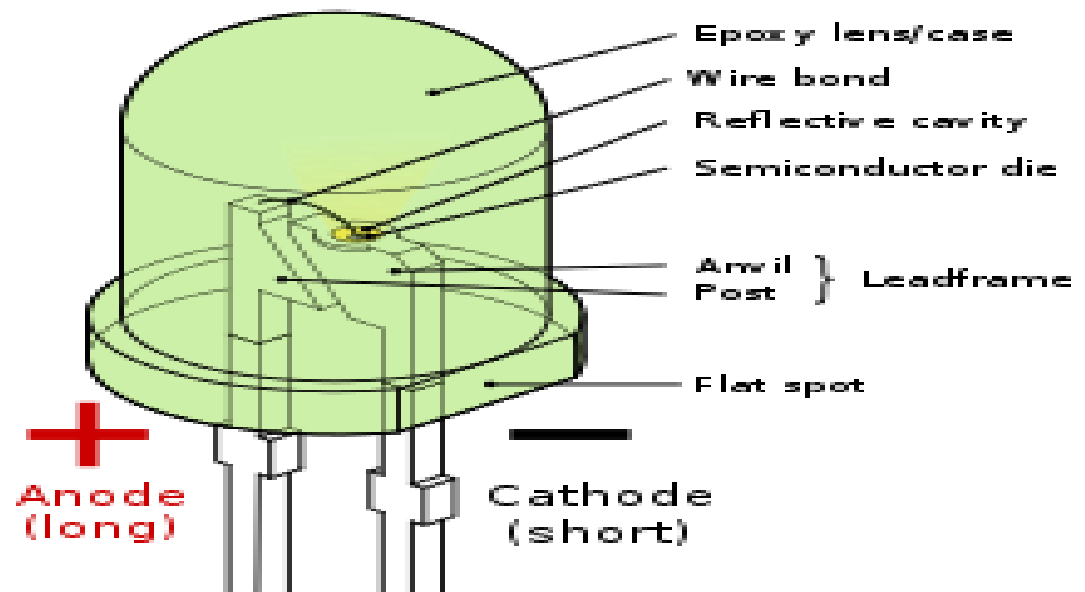
Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.



LED:

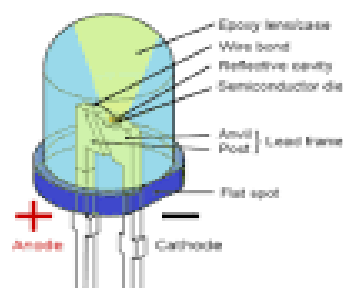
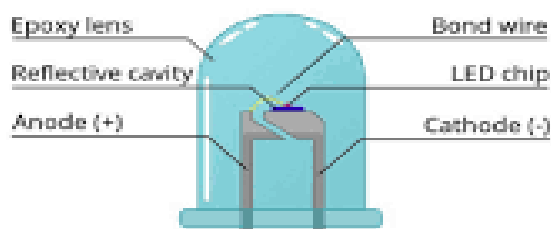
A light-emitting diode is a semiconductor light source that emits light when current flows through it. LEDs (light-emitting diodes) are small, bright, power-efficient lights commonly used in electronic products. An LED light is a polarized part, meaning it has to be connected to a circuit in a certain way to work properly.

LEDs have many advantages over incandescent light sources, including lower power consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. In exchange for these generally favorable attributes, disadvantages of LEDs include electrical limitations to low voltage and generally to DC (not AC) power, inability to provide steady illumination from a pulsing DC or an AC electrical supply source, and lesser maximum operating temperature and storage temperature. In contrast to LEDs, incandescent lamps can be made to intrinsically run at virtually any supply voltage, can utilize either AC or DC current interchangeably, and will provide steady illumination when powered by AC or pulsing DC even at a frequency as low as 50 Hz. LEDs usually need electronic support components to function, while an incandescent bulb can and usually does operate directly from an unregulated DC or AC power source.



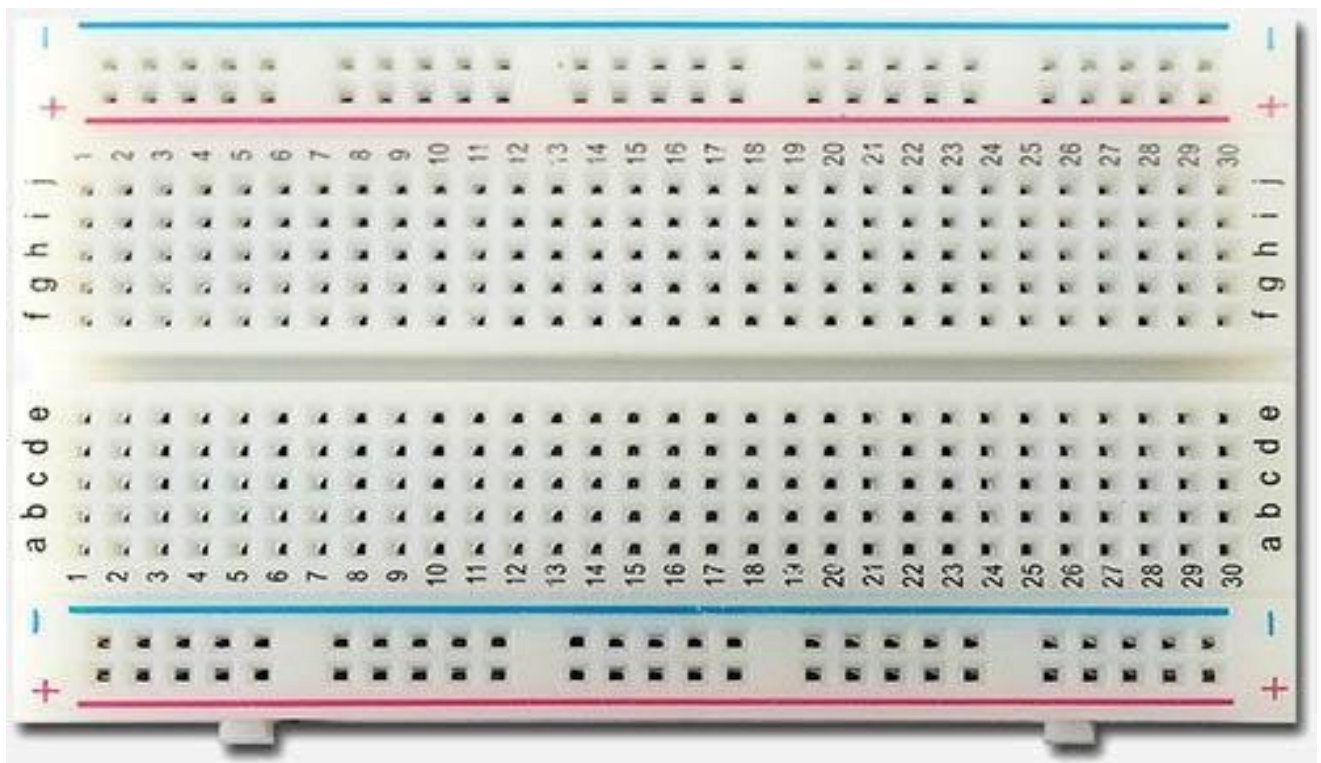
| Light Emitting Diode (LED) | Symbol of LED |
|--|---|
| <p>A physical diagram of an LED with a red anode and a grey cathode. The anode is labeled Anode and Big Leg, and the cathode is labeled Cathode and Small Leg. A red plus sign is next to the anode and a black minus sign is next to the cathode.</p> | <p>The standard circuit symbol for an LED, which is a triangle pointing to the right with two arrows indicating light emission. The anode is on the left and the cathode is on the right. A red plus sign is next to the anode and a black minus sign is next to the cathode.</p> |

How Does an LED Work?



Breadboard:

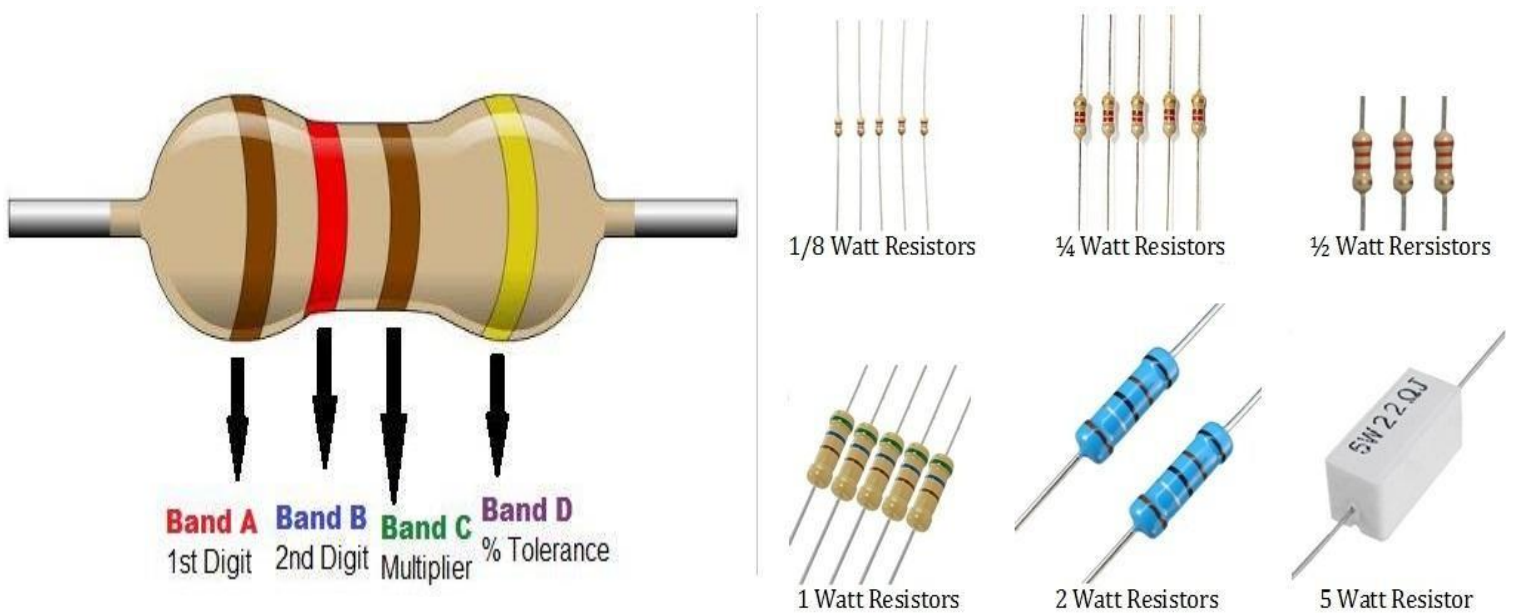
A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode). It is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted.



Resistor:

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to

divide voltages, bias active elements, and terminate transmission lines, among other uses.



Procedure:

- 1 Open and create a new account at www.tinkercad.com or log in with an existing Gmail account.



Welcome back

How will you sign in?

Students, join your class

Email or Username

Sign in with Google

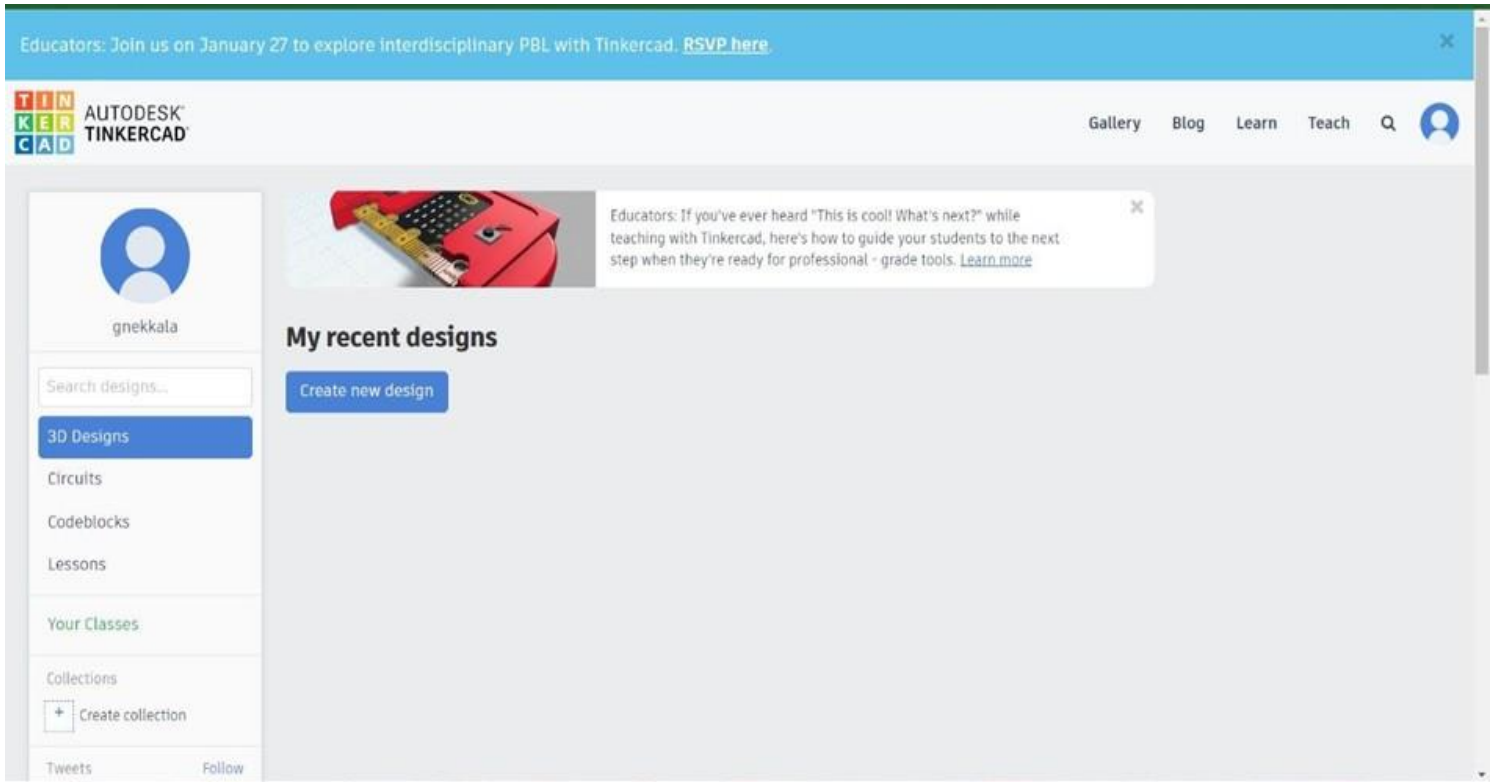
Sign in with Apple

More sign in options...

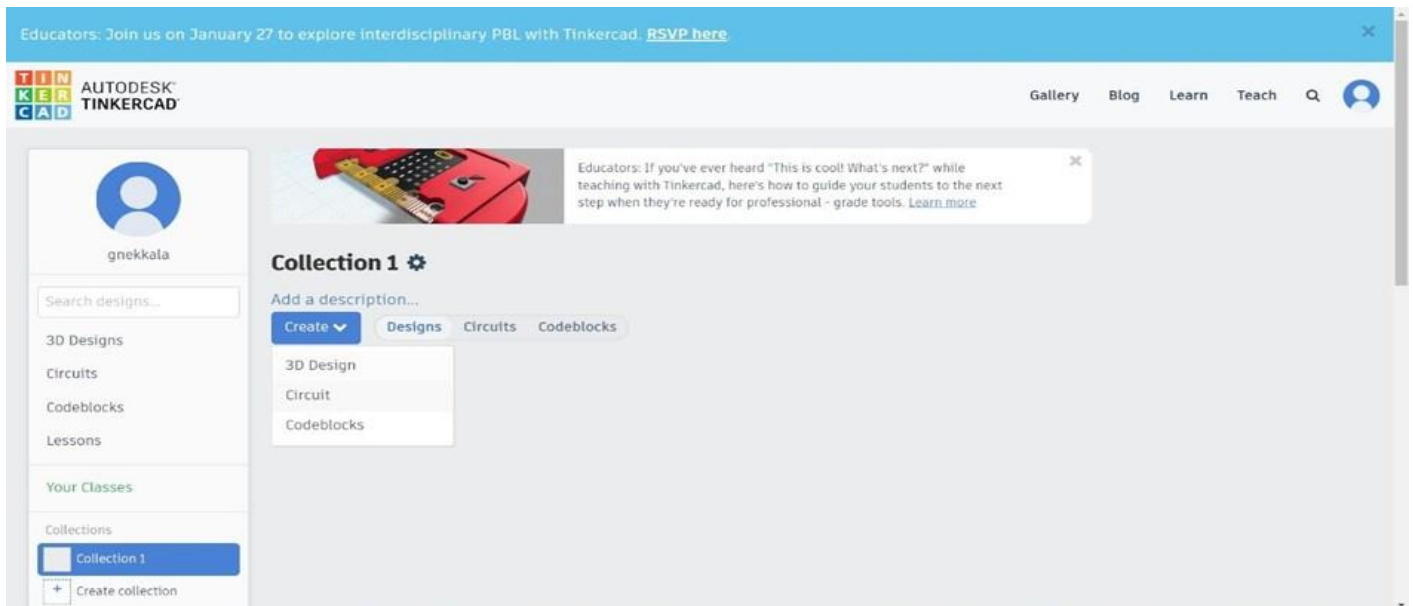
Don't have an account yet?

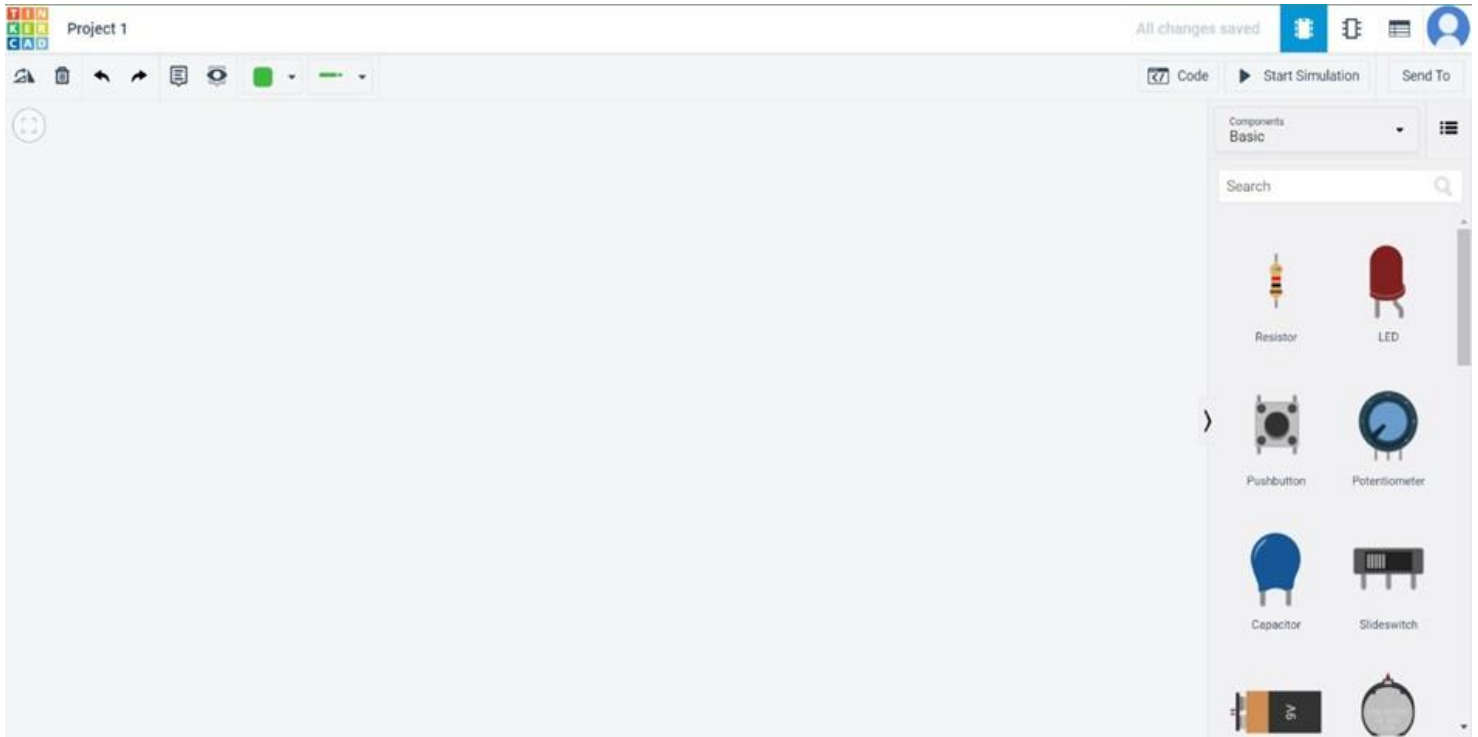
Join Tinkercad

- 2 Click on go to create Collection and create a new collection.



- 3 Go to create menu and select circuit

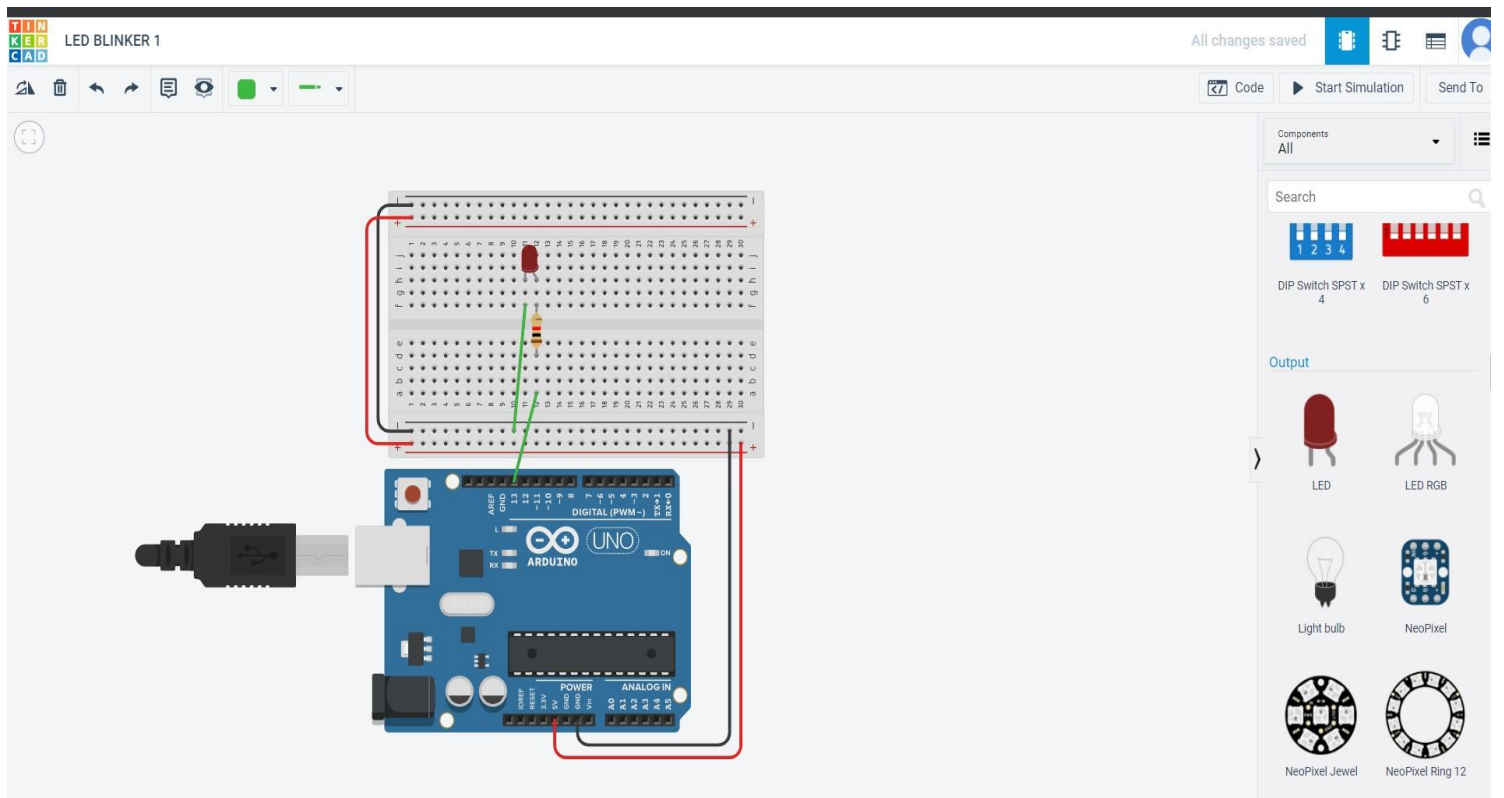
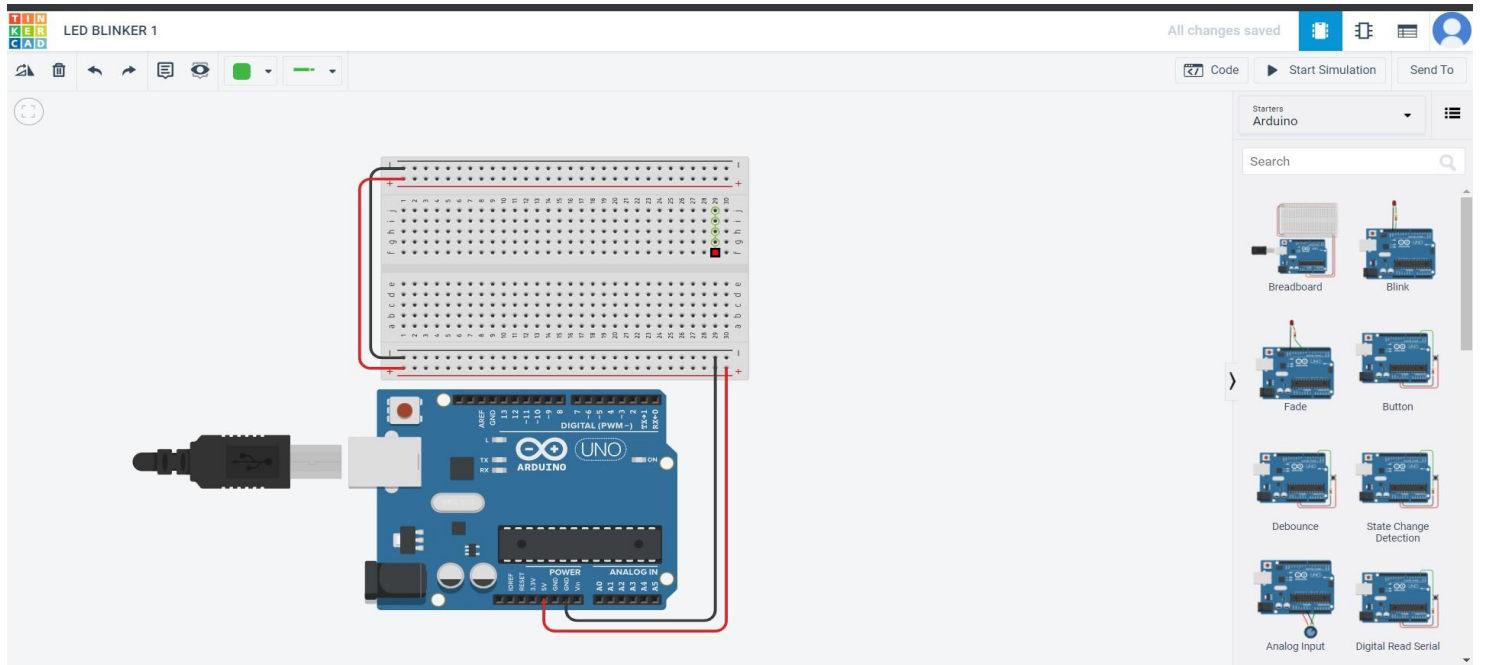




4 Select the Arduino and breadboard and place it in the design area.

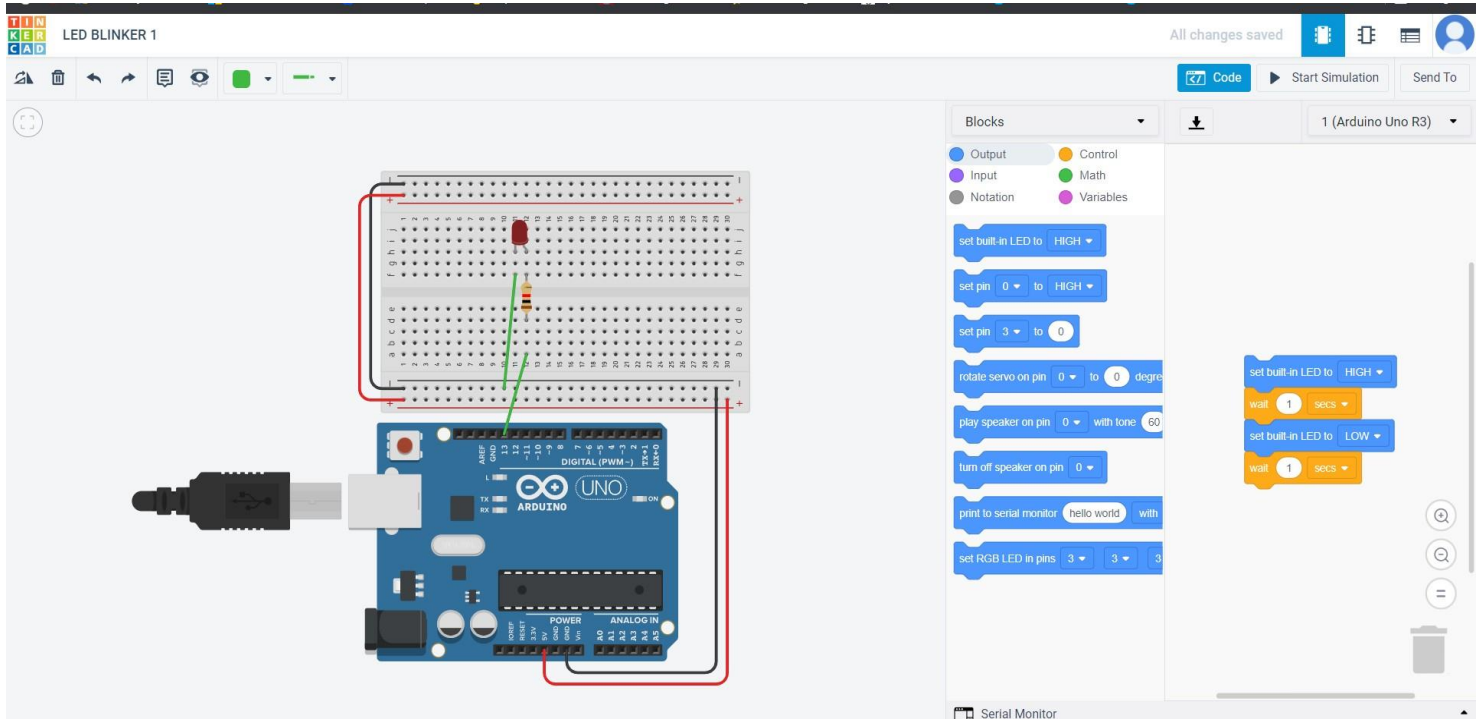
Search the component LED and resistor and make connections.
Configure the resistor value as 330 ohms.

6 Attach the LED to an output pin of the Arduino D13.

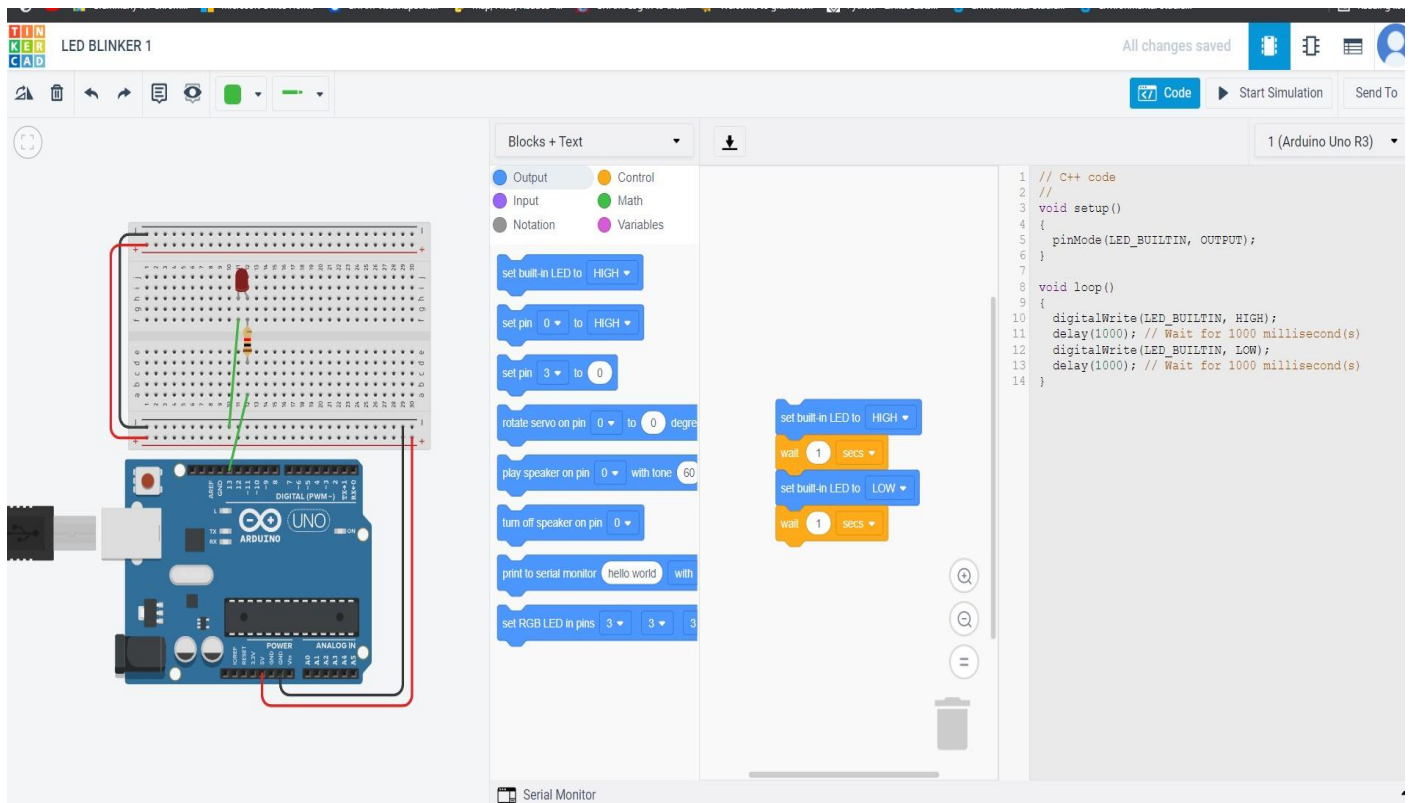


Once the circuit connection are ready, programming the Arduino can be done in three ways.

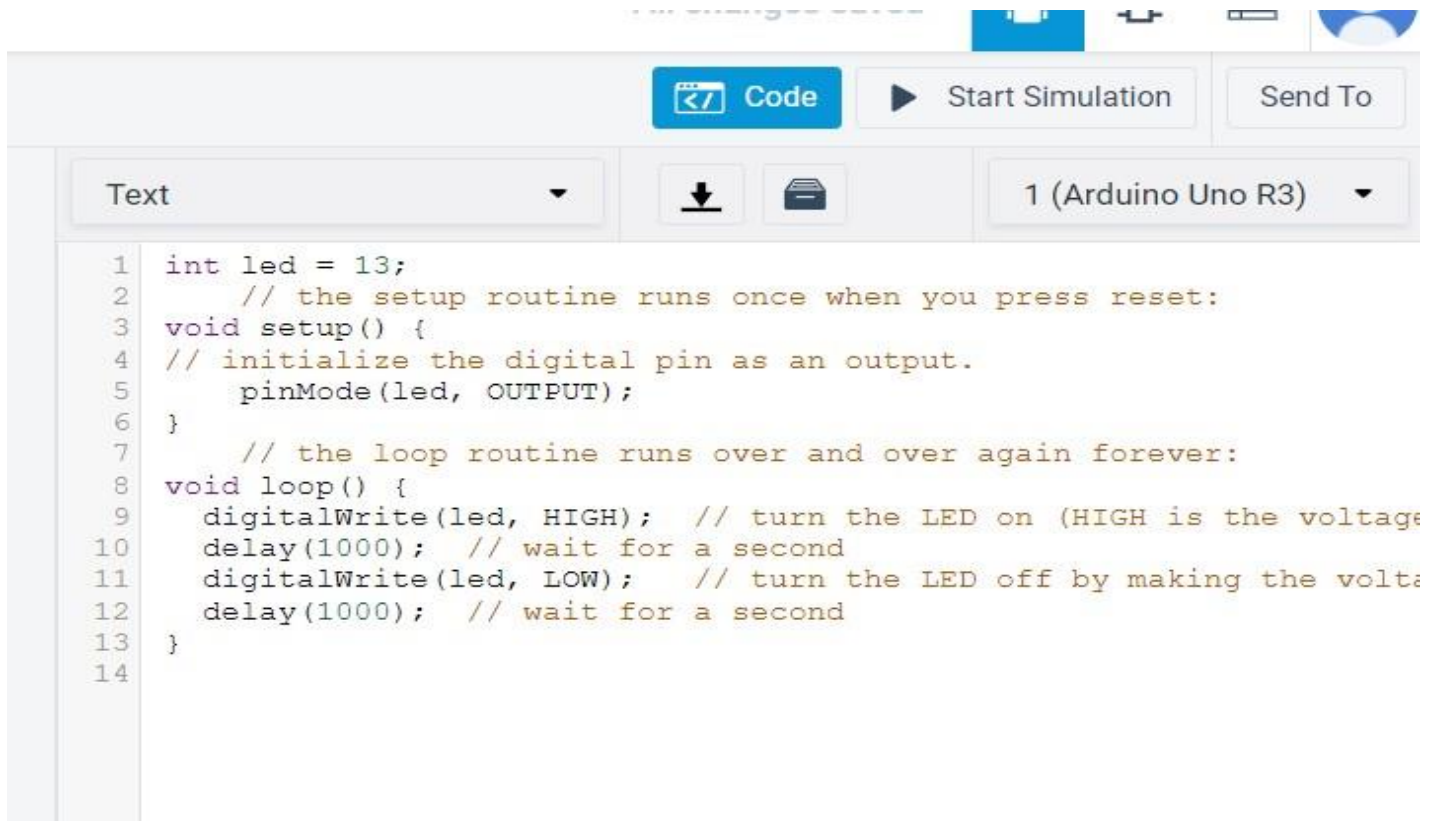
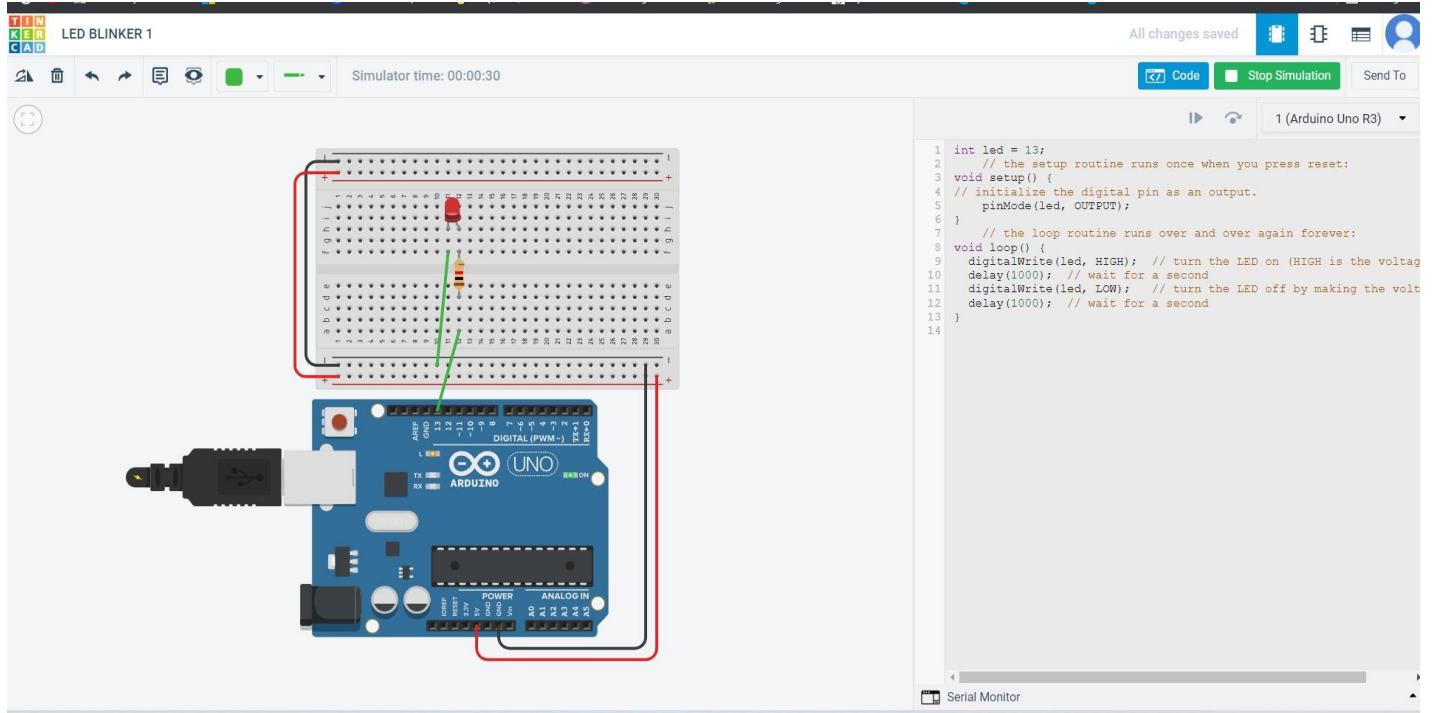
1. Using code blocks



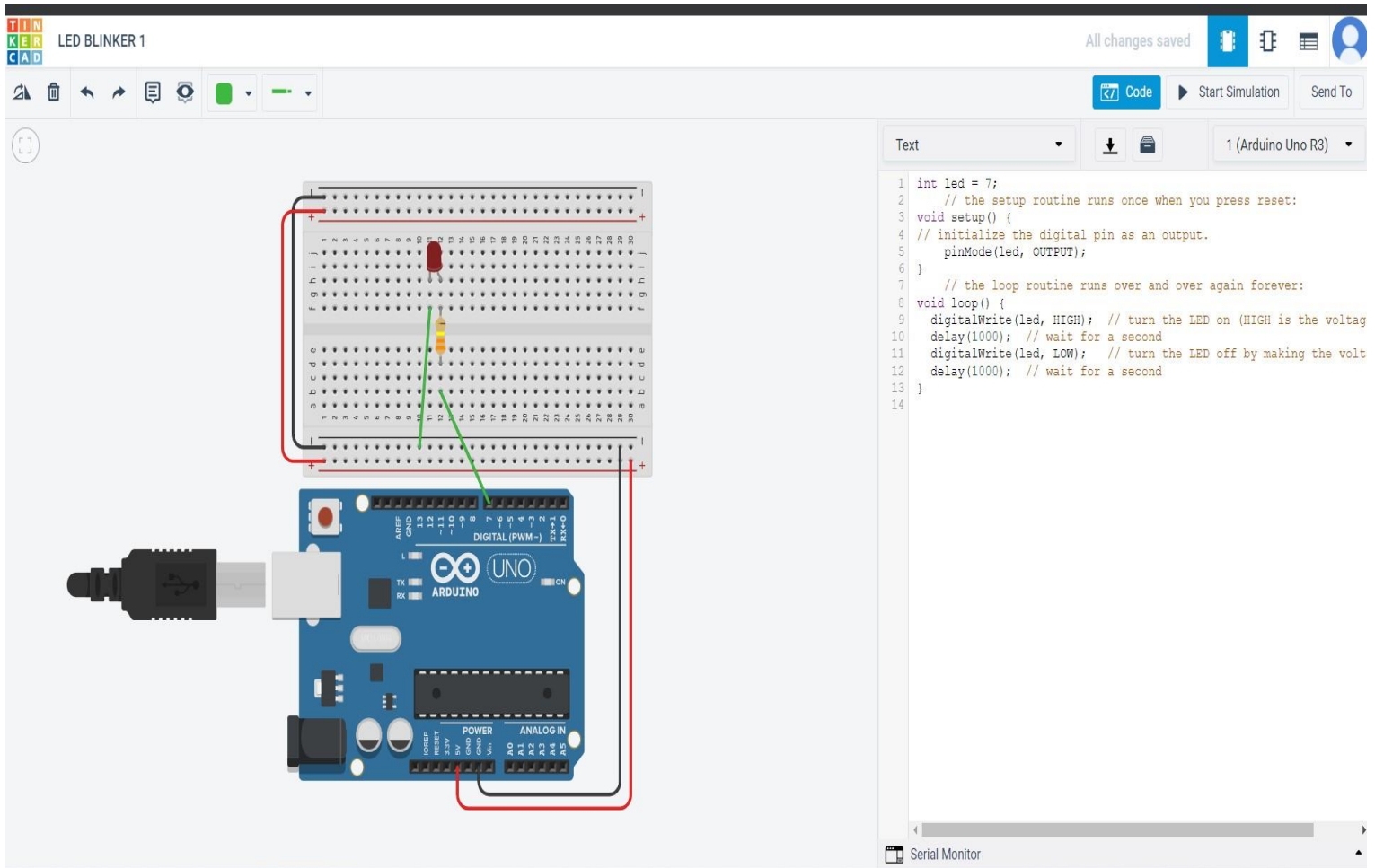
2. Using code blocks + text programming



2. With text program



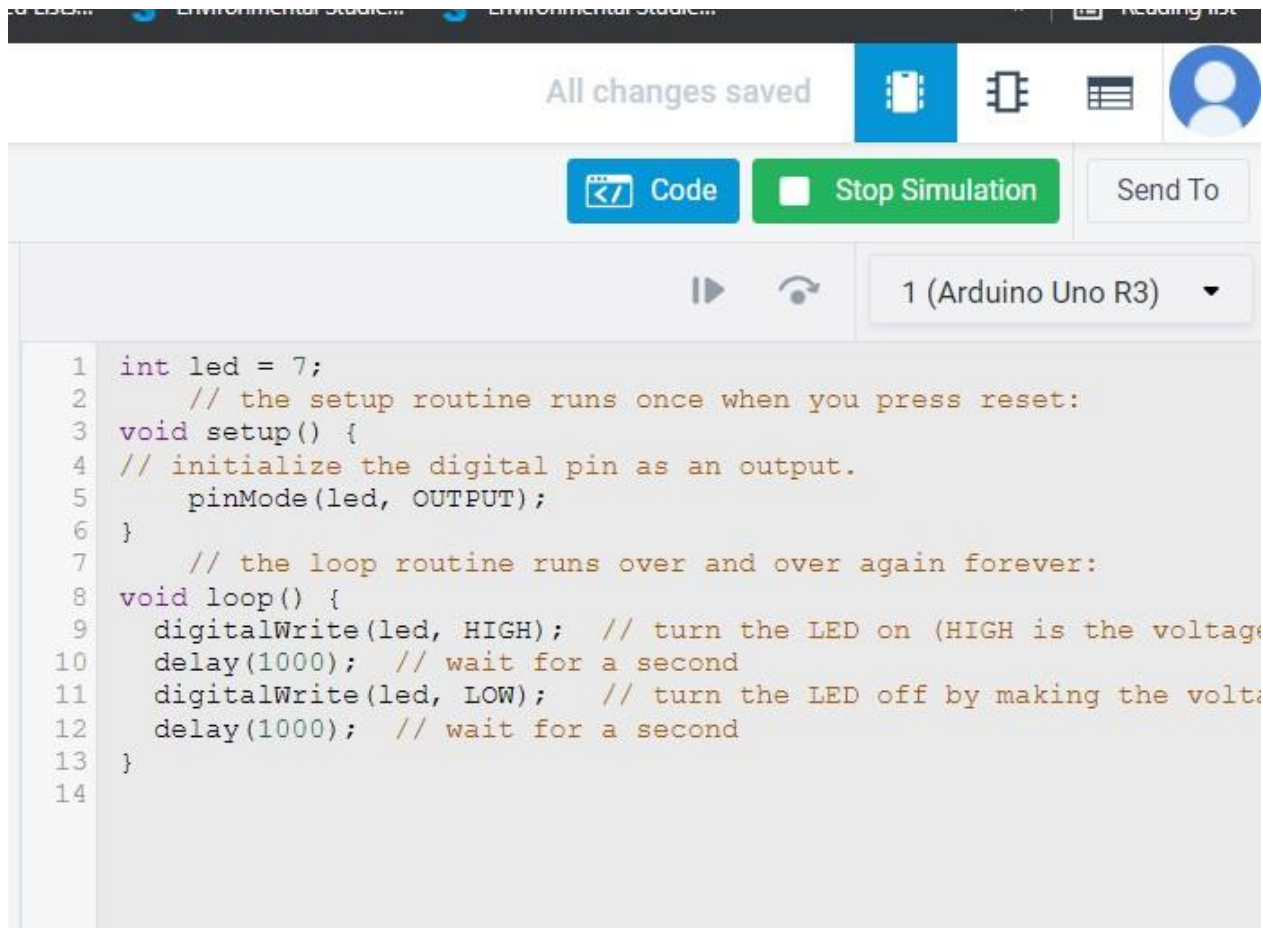
Let us try using a different pin of the Arduino – say D7. Move the red jumper lead from pin D13 to pin D7 and modify the following line near the top of the sketch:



The screenshot shows the TINKER CAD environment. On the left, a breadboard is connected to an Arduino Uno R3. A red LED is connected to pin D7 of the Arduino and ground. The Arduino is connected to a USB cable. On the right, the code editor shows the following sketch:

```
1 int led = 7;
2 // the setup routine runs once when you press reset:
3 void setup() {
4   // initialize the digital pin as an output.
5   pinMode(led, OUTPUT);
6 }
7 // the loop routine runs over and over again forever:
8 void loop() {
9   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage)
10  delay(1000); // wait for a second
11  digitalWrite(led, LOW); // turn the LED off by making the voltage low
12  delay(1000); // wait for a second
13 }
14
```

At the bottom, the Serial Monitor is open, showing the text "1 (Arduino Uno R3)".



The screenshot shows the Tinkercard software interface. At the top, there's a status bar with "All changes saved" and icons for a circuit board, a microcontroller, a list, and a user profile. Below this is a toolbar with "Code" (a blue button with a code icon), "Stop Simulation" (a green button with a stop icon), and "Send To" (a button with a dropdown arrow). The main area displays a code editor for an Arduino Uno R3 board. The code is as follows:

```
1 int led = 7;
2 // the setup routine runs once when you press reset:
3 void setup() {
4 // initialize the digital pin as an output.
5   pinMode(led, OUTPUT);
6 }
7 // the loop routine runs over and over again forever:
8 void loop() {
9   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage
10  delay(1000); // wait for a second
11  digitalWrite(led, LOW); // turn the LED off by making the volta
12  delay(1000); // wait for a second
13 }
14
```

Result:

The controlling of LED with Aurdino Board using Tinkercard software is successfully done.

BY

S.Lohit Krishna

122010307057

B-7

