EXPERIMENT-01

NAME: K.Swethank Rohan
SECTION: B-15
ROLL NUMBER:122010315005

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.

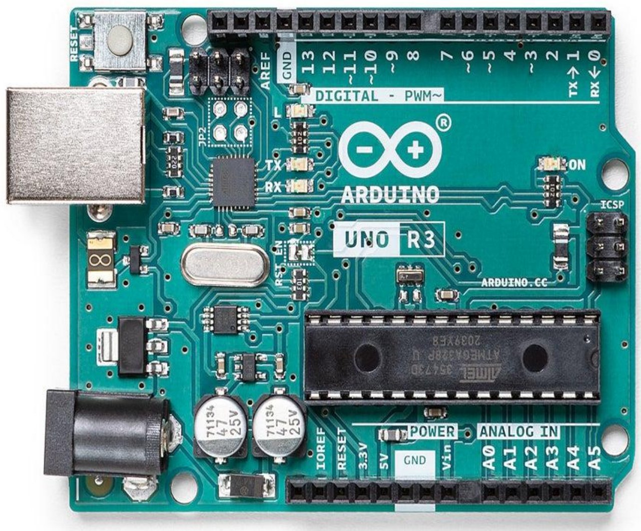
Hardware Requirements:

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

Theory:

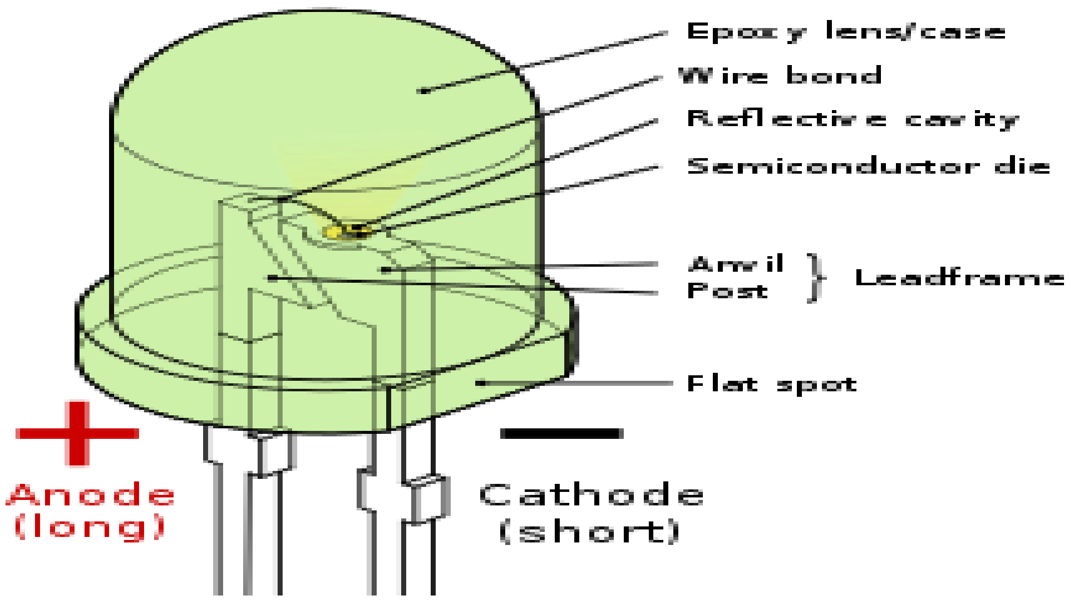
Arduino Uno:

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.



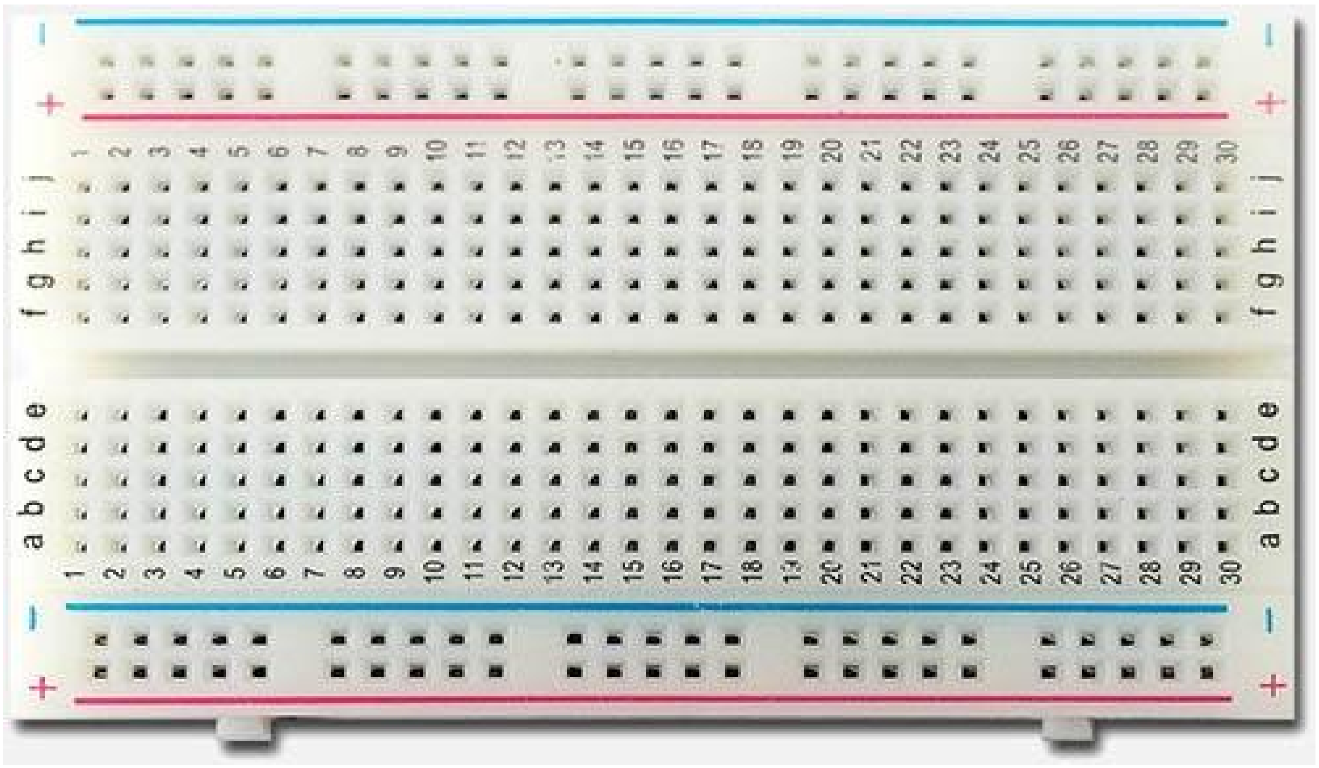
LED:

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.



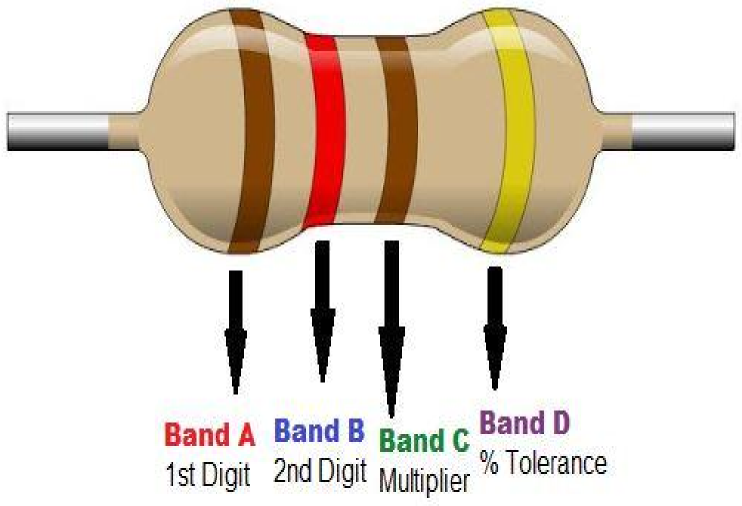
Breadboard:

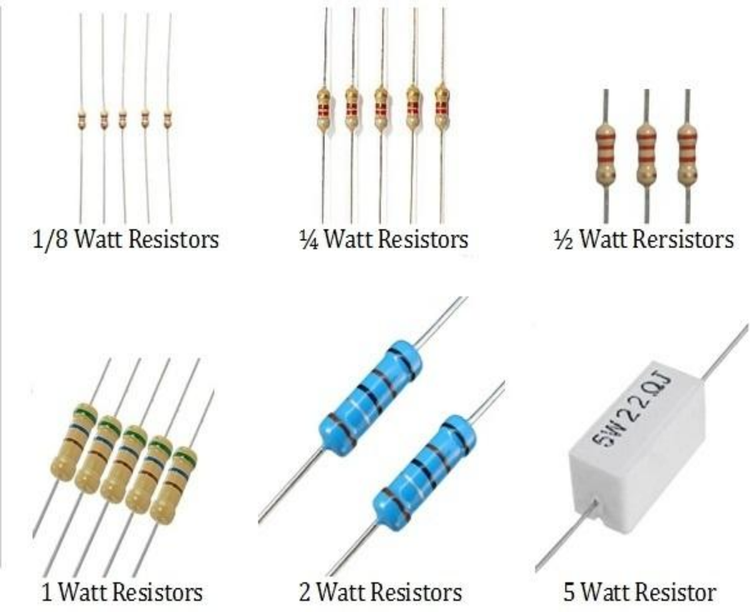
A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode). 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:

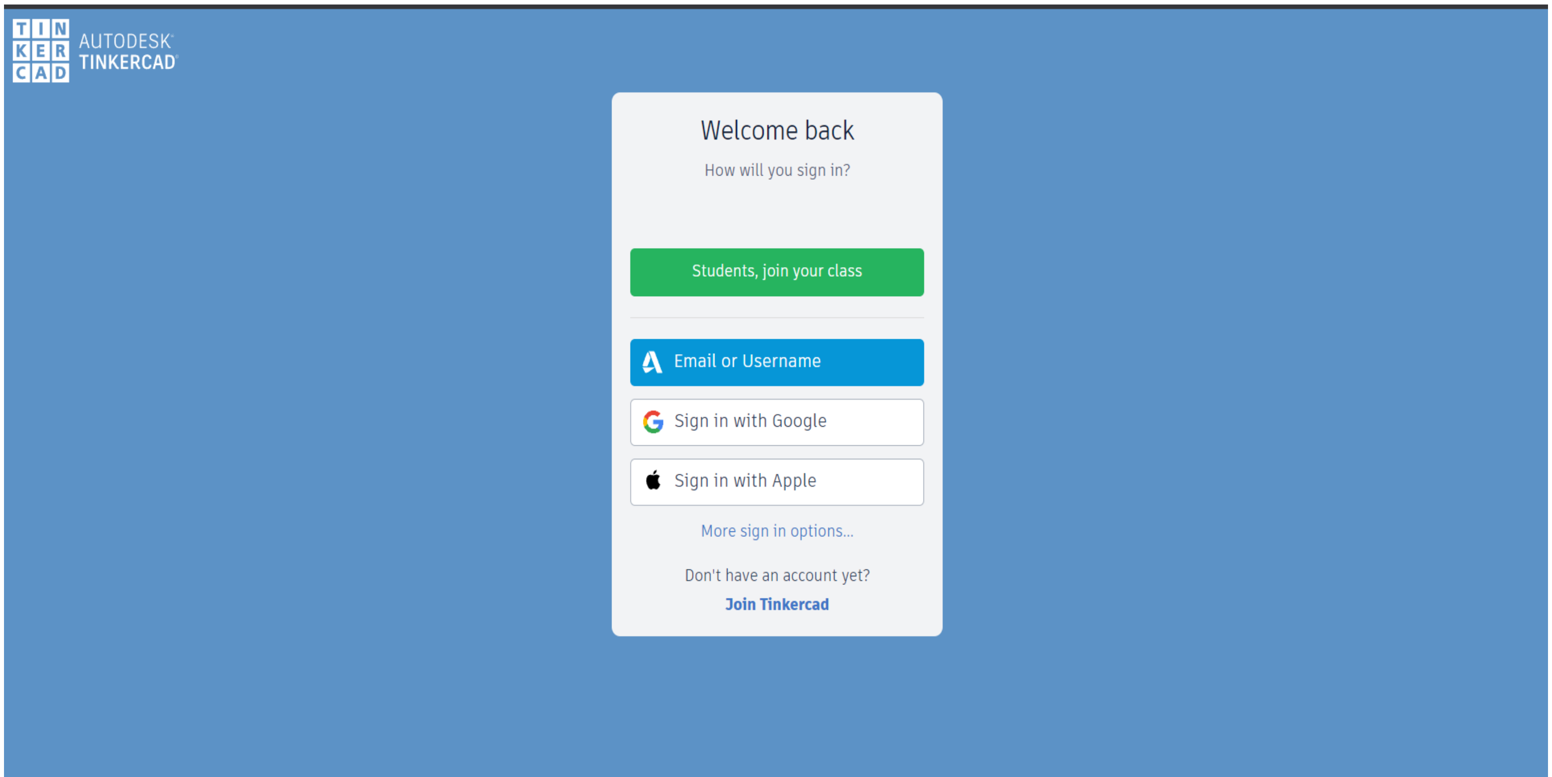
A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.



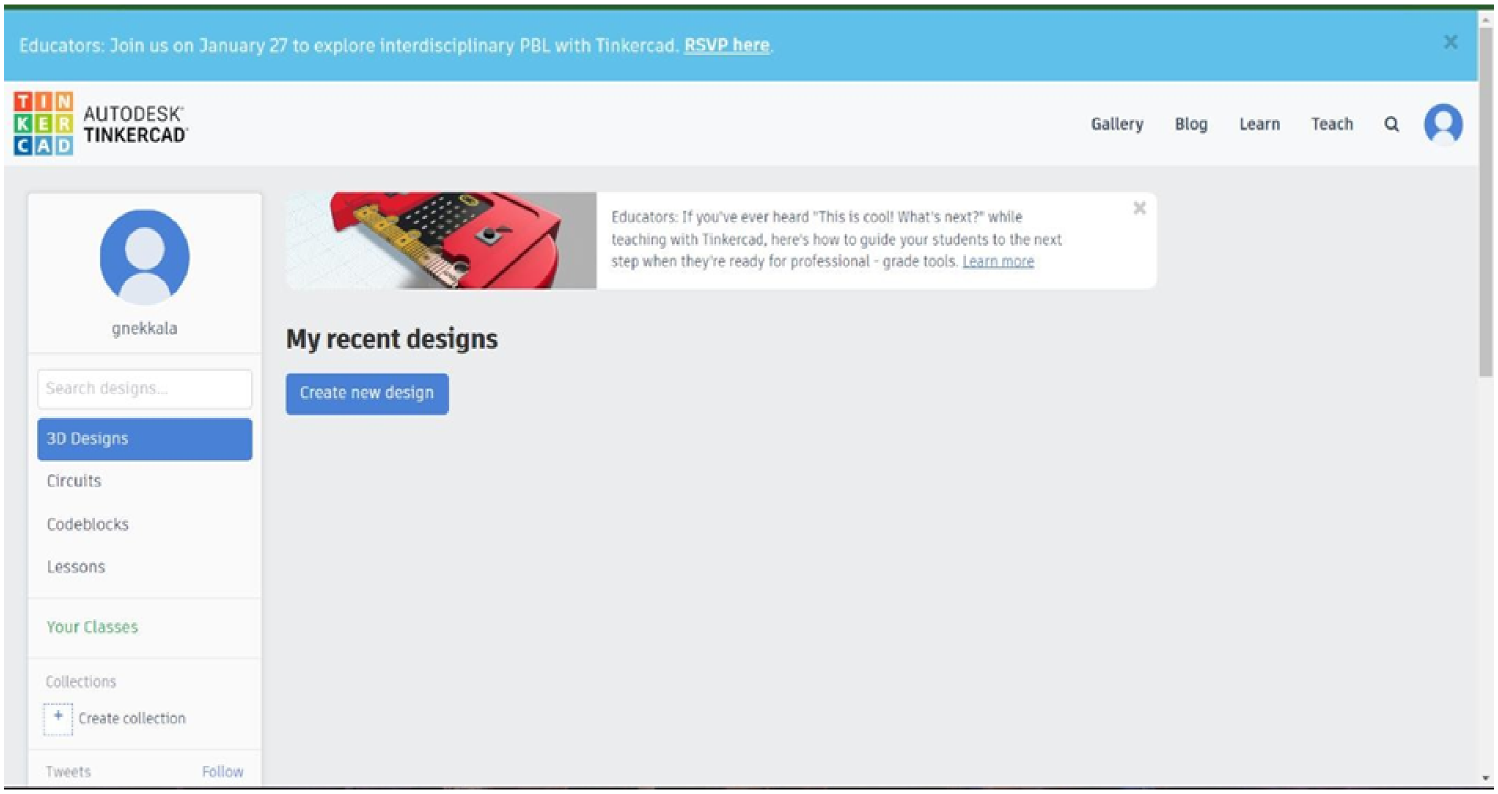


Procedure:

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

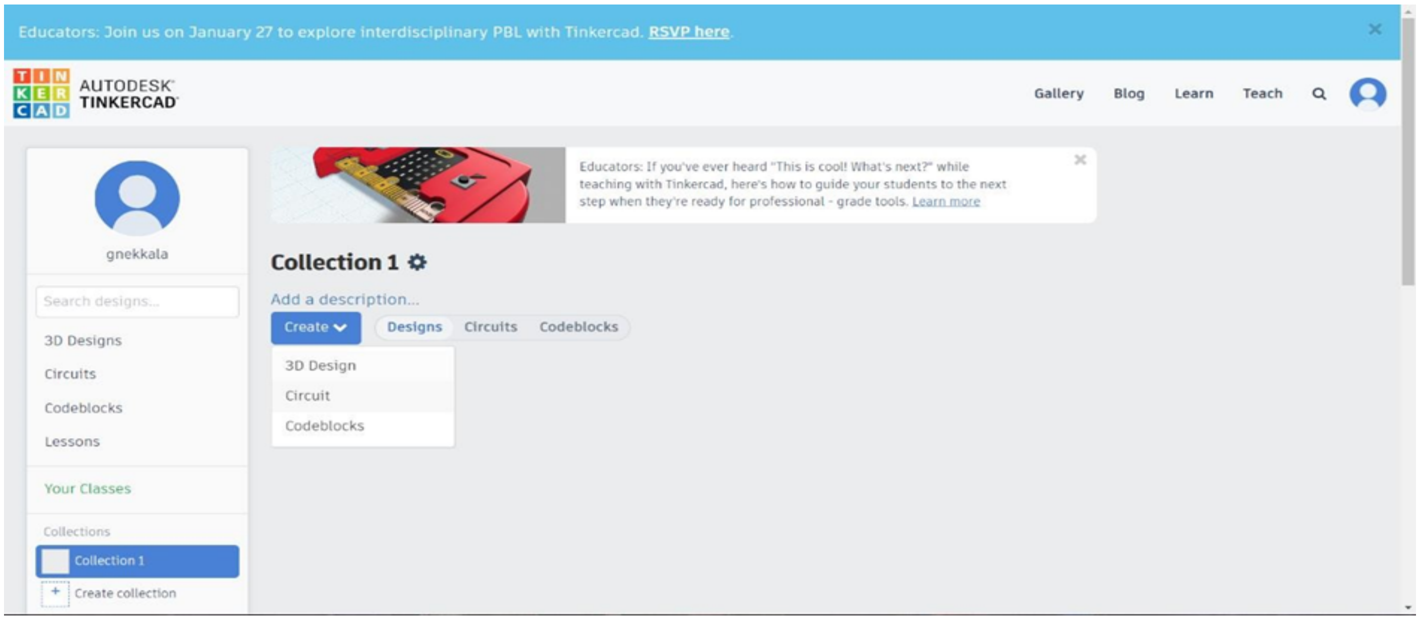


2. Click on go to create Collection and create a new collection.

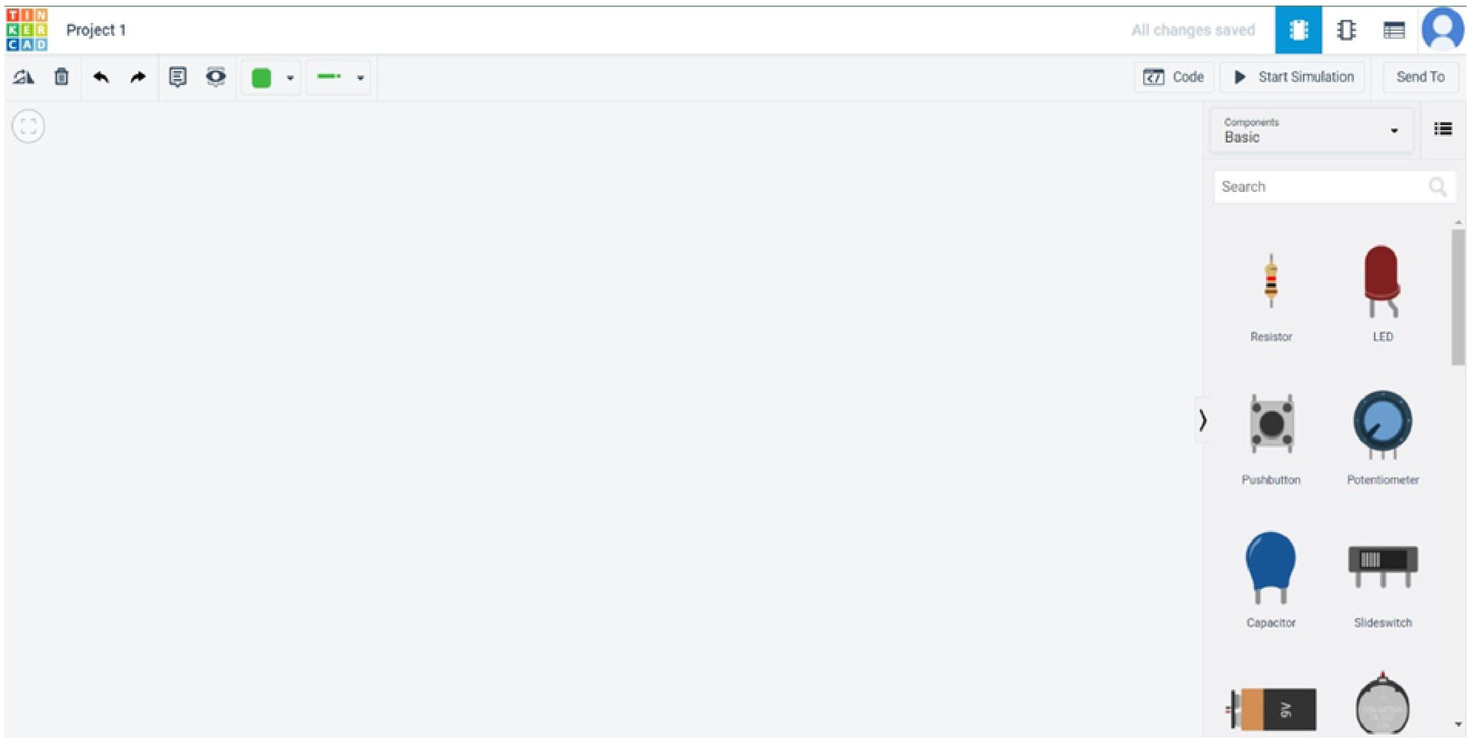

Educators: Join us on January 27 to explore interdisciplinary PBL with Tinkercad. RSVP here.

Educators: If you've ever heard "This is cool! What's next?" while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. Learn more

3. Go to create menu and select circuit


Educators: Join us on January 27 to explore interdisciplinary PBL with Tinkercad. RSVP here.

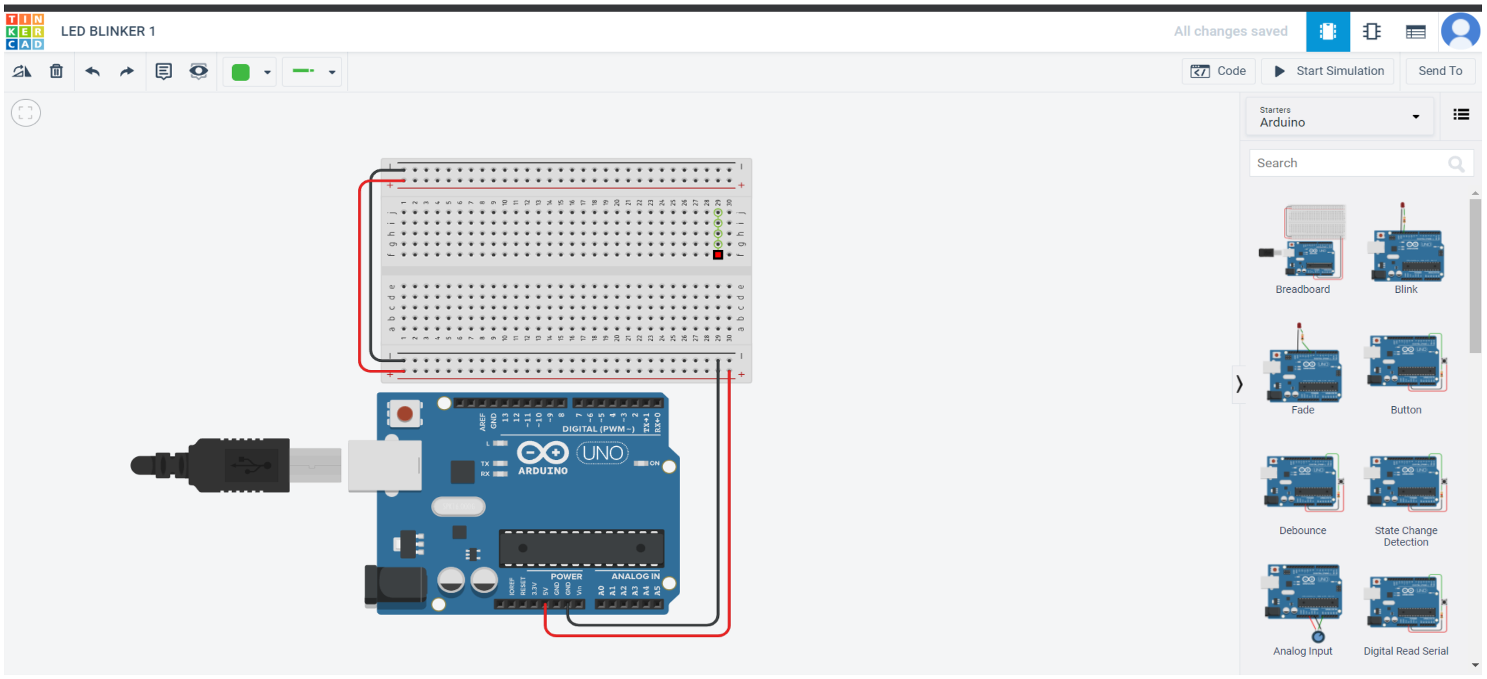
Educators: If you've ever heard "This is cool! What's next?" while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. Learn more

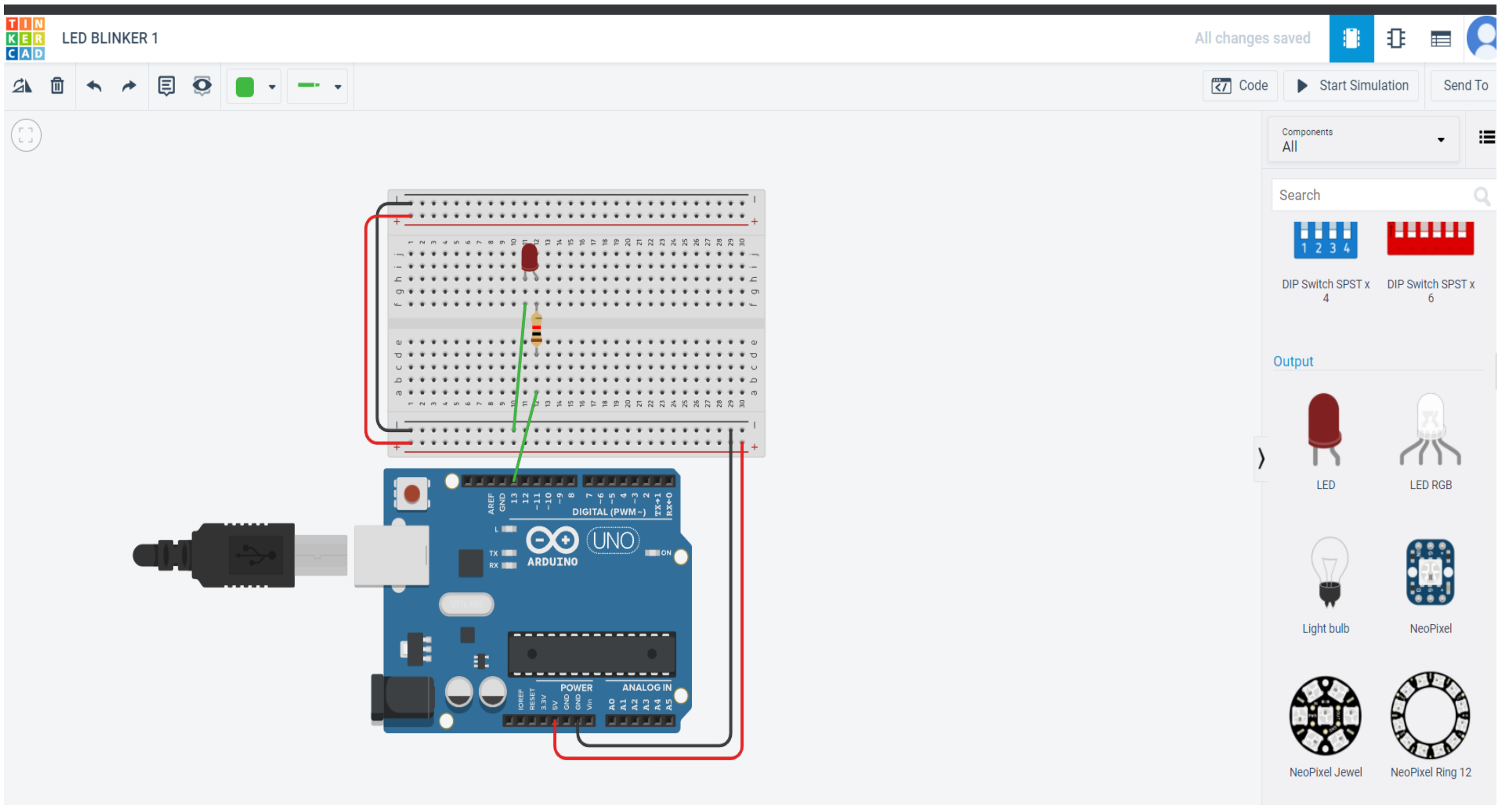


4. Select the Arduino and breadboard and place it in the design area.

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

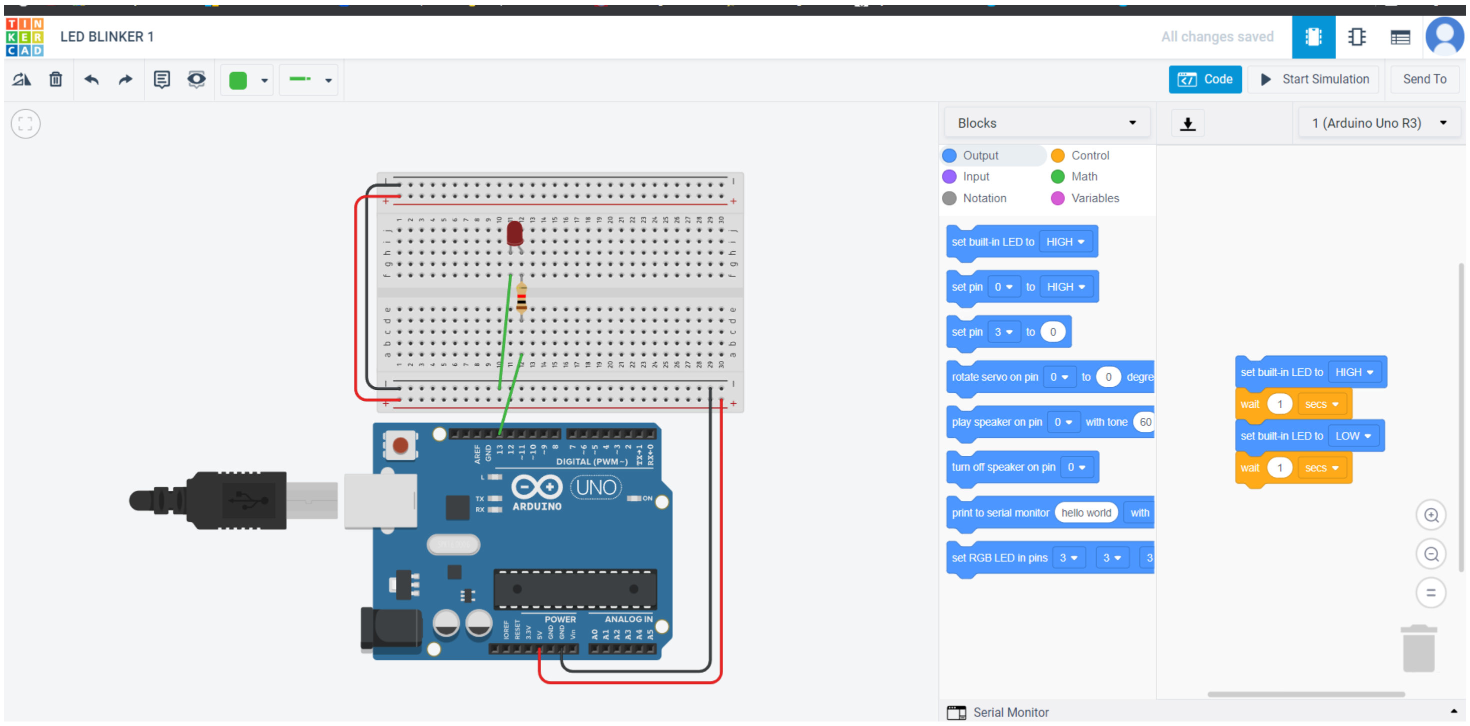
6. Attach the LED to an output pin of the Arduino D13.





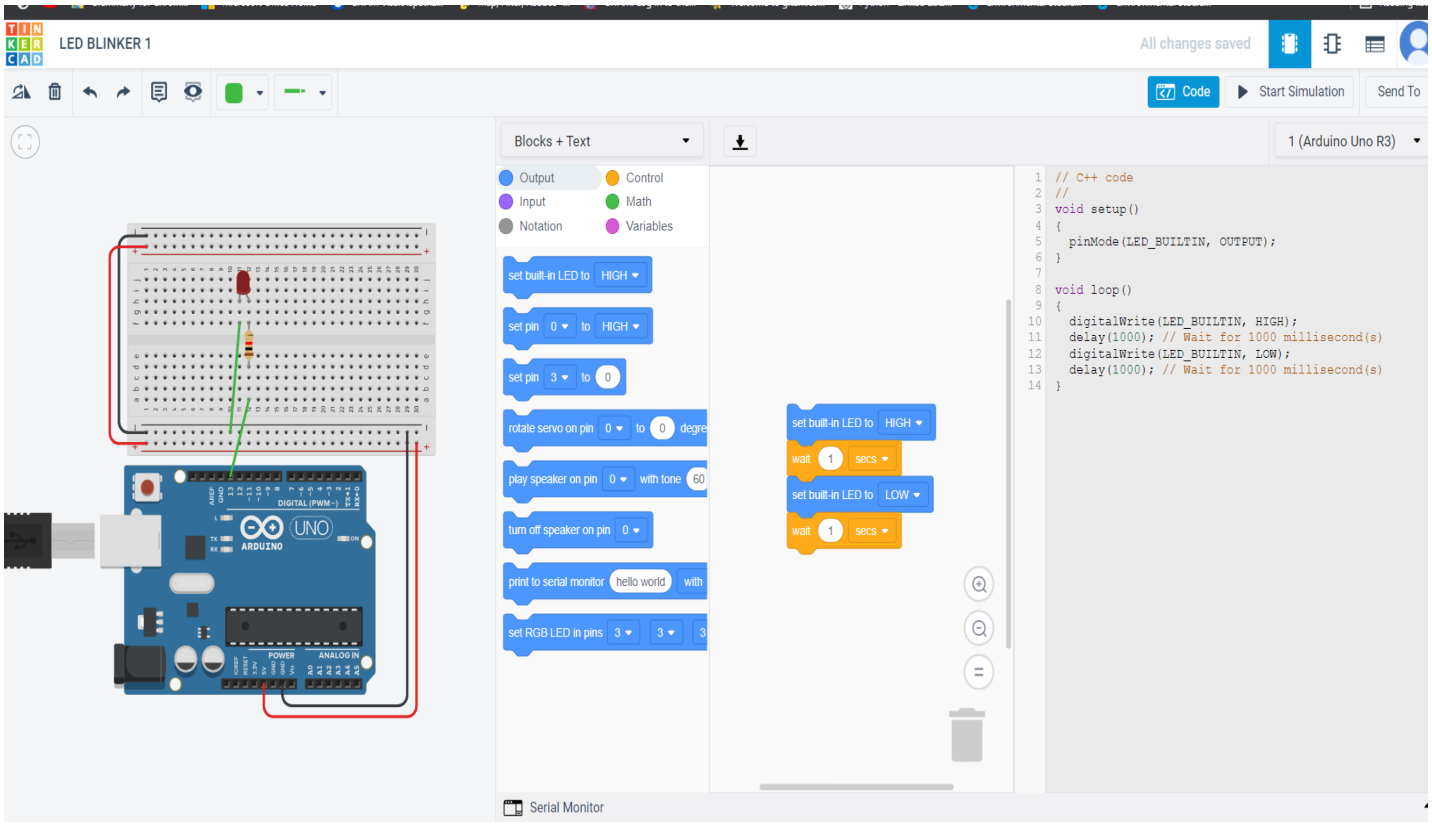
7. Once the circuit connection are ready, programming the Arduino can be done in three ways.

1. Using code blocks



```
set built-in LED to HIGH
wait 1 secs
set built-in LED to LOW
wait 1 secs
```

2. Using code blocks + text programming

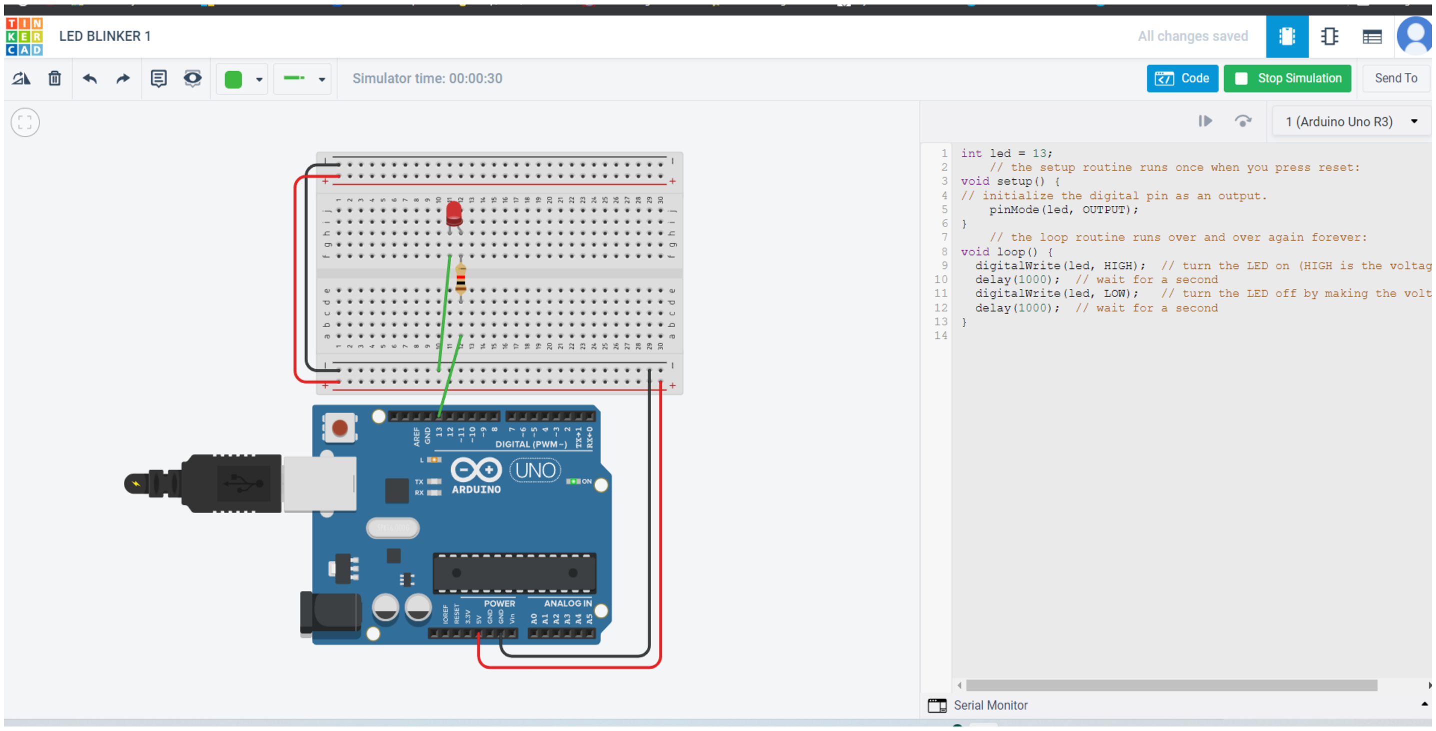


```
set built-in LED to HIGH
wait 1 secs
set built-in LED to LOW
wait 1 secs
```

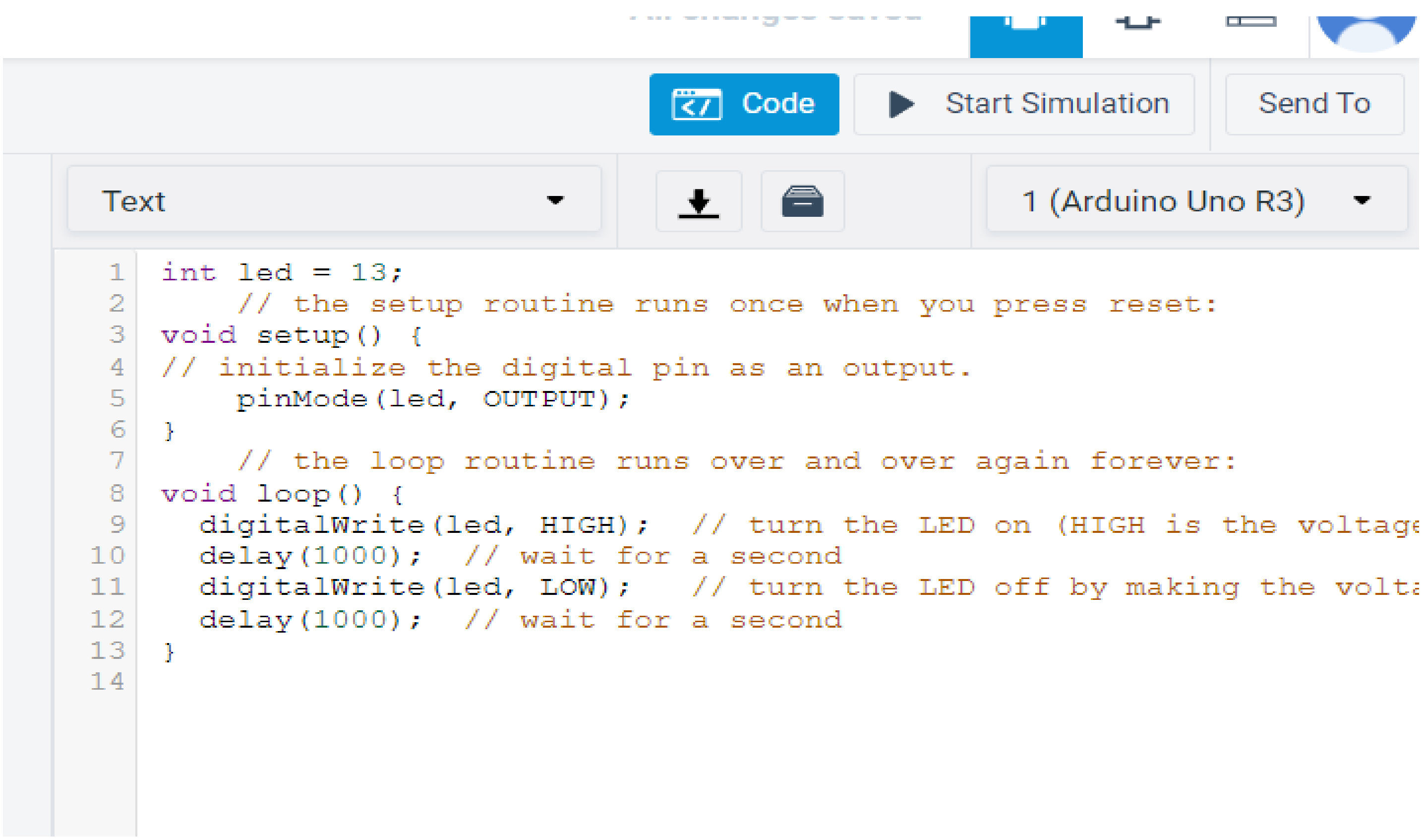
```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

2. With text program

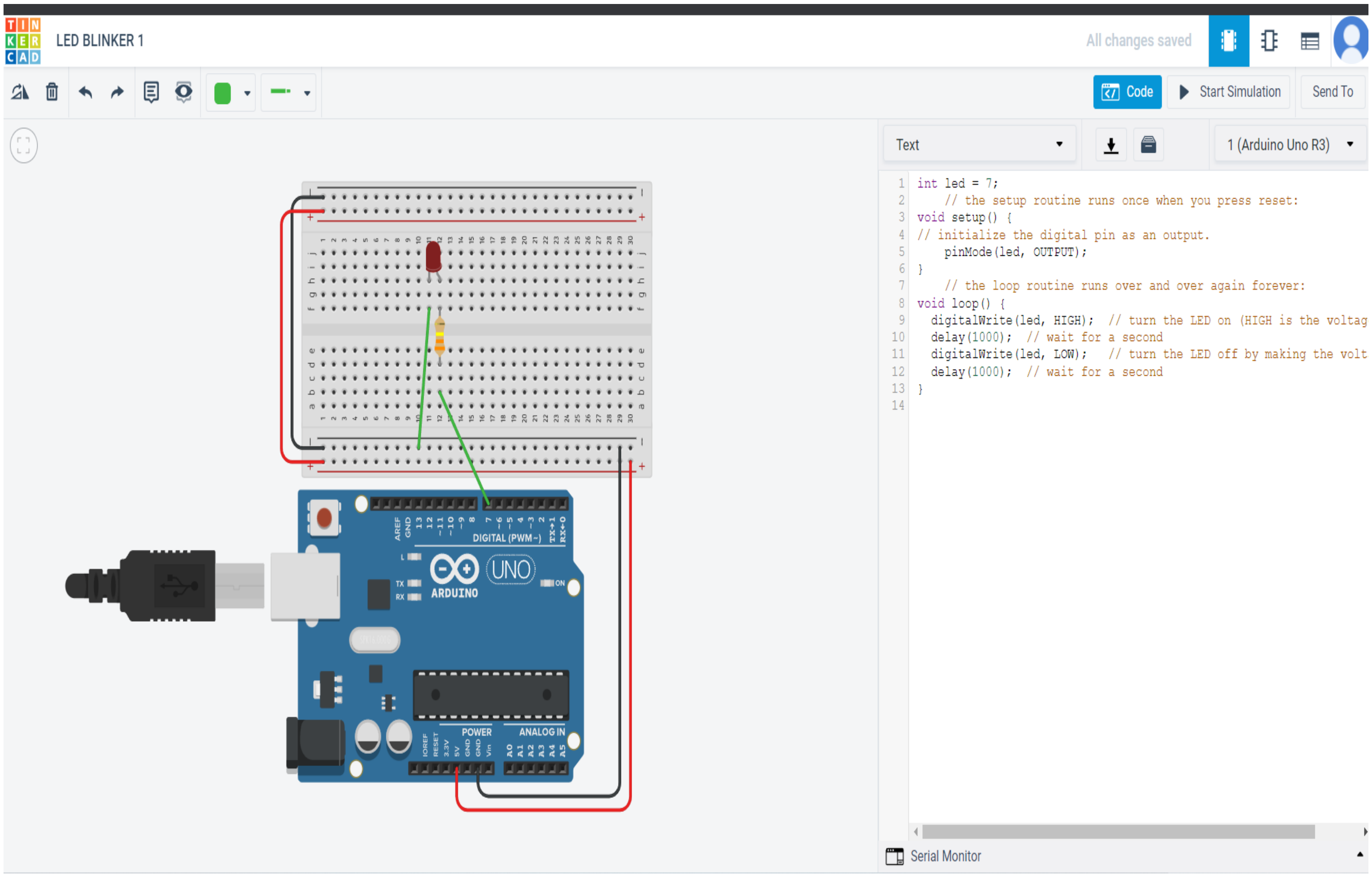


```
int led = 13;
    // the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}
    // the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltag
  delay(1000);  // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the volt
  delay(1000);  // wait for a second
}
```

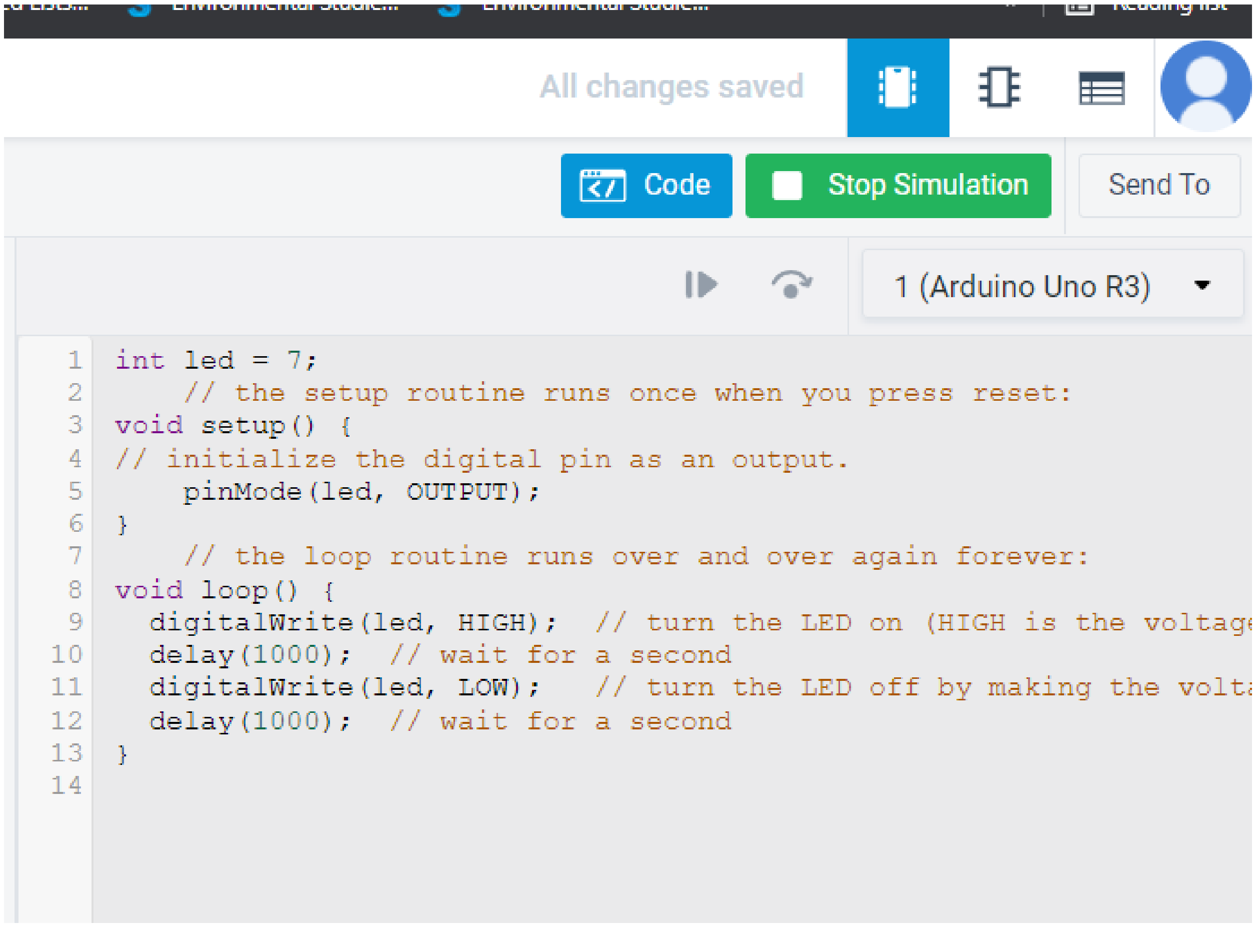


```
int led = 13;
    // the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}
    // the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage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

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:



```
int led = 7;
    // the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}
    // the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltag
  delay(1000);  // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the volt
  delay(1000);  // wait for a second
}
```



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

Result:

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