

EXPERIMENT-1

NAME : P RAJENDRA

ROLL NUMBER:122010315015

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 Jumper Wires

Theory:

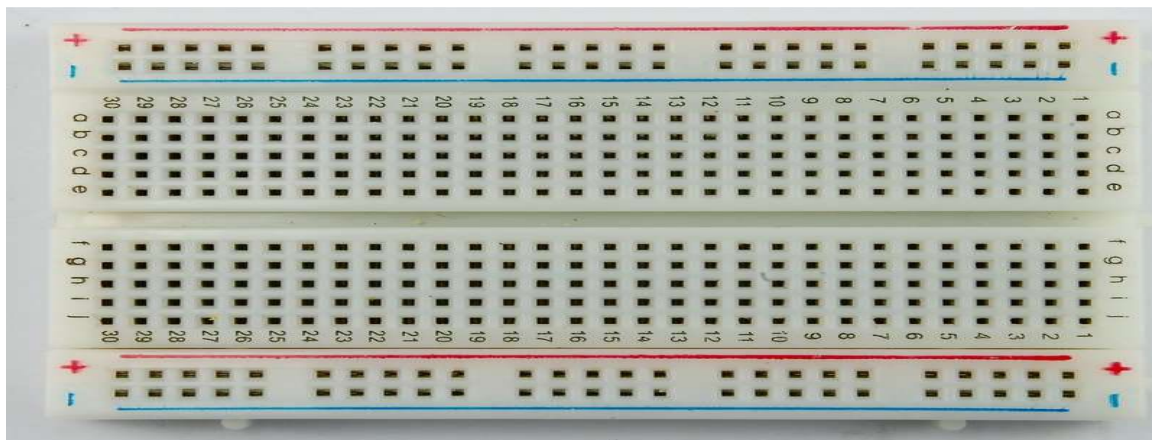
ARDUINO UNO:

Arduino acts as the brain of the system and processes the data from the sensor. Arduino is an open source hardware platform that is readily available for hobbyists & enthusiasts across the globe to build projects. It comes with an ATMEGA microcontroller that processes the data and facilitates the proper working of the IoT system.



BREADBOARD:

A breadboard 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. The bread board has strips of metal which run underneath the board and connect the holes on the top of the board.



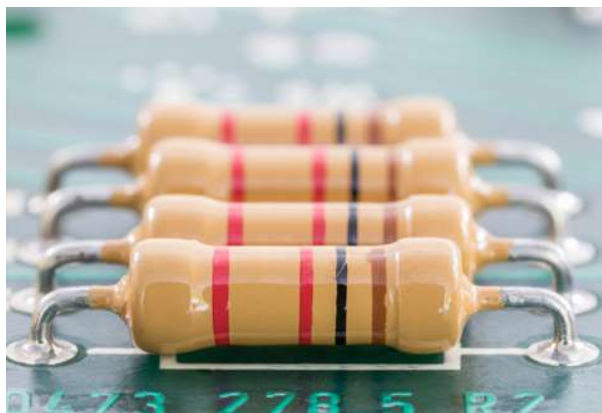
LED:

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.



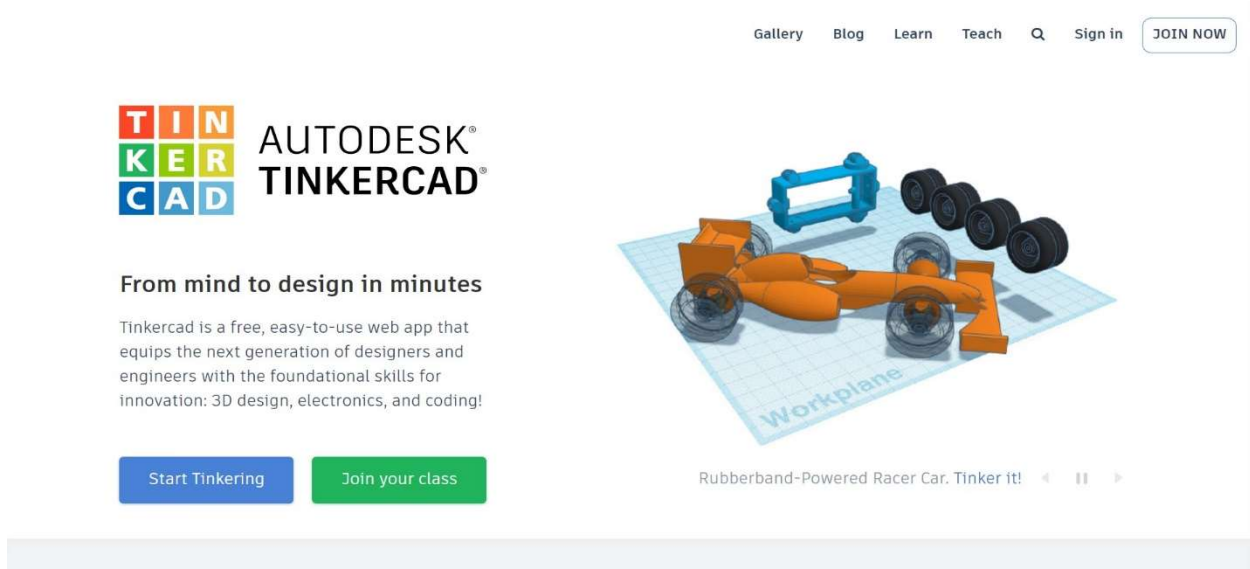
RESISTOR:

Special components called resistors are made for the express purpose of creating a precise quantity of resistance for insertion into a circuit. They are typically constructed of metal wire or carbon and engineered to maintain a stable resistance value over a wide range of environmental conditions.

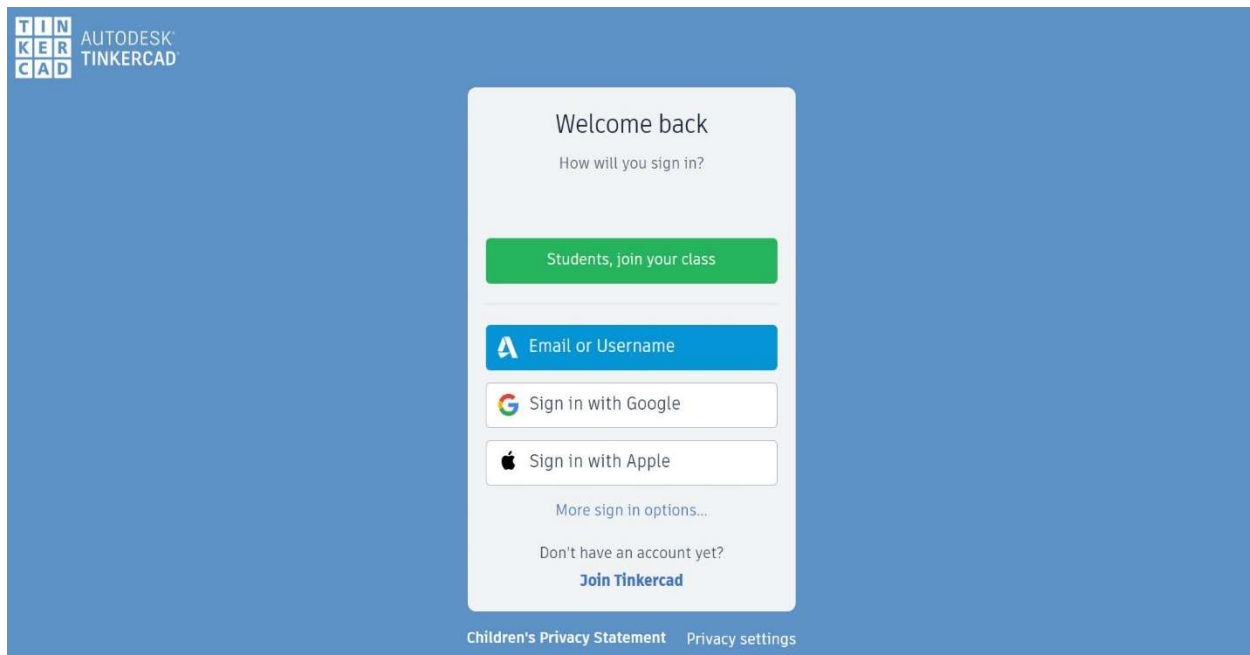


PROCEDURE:

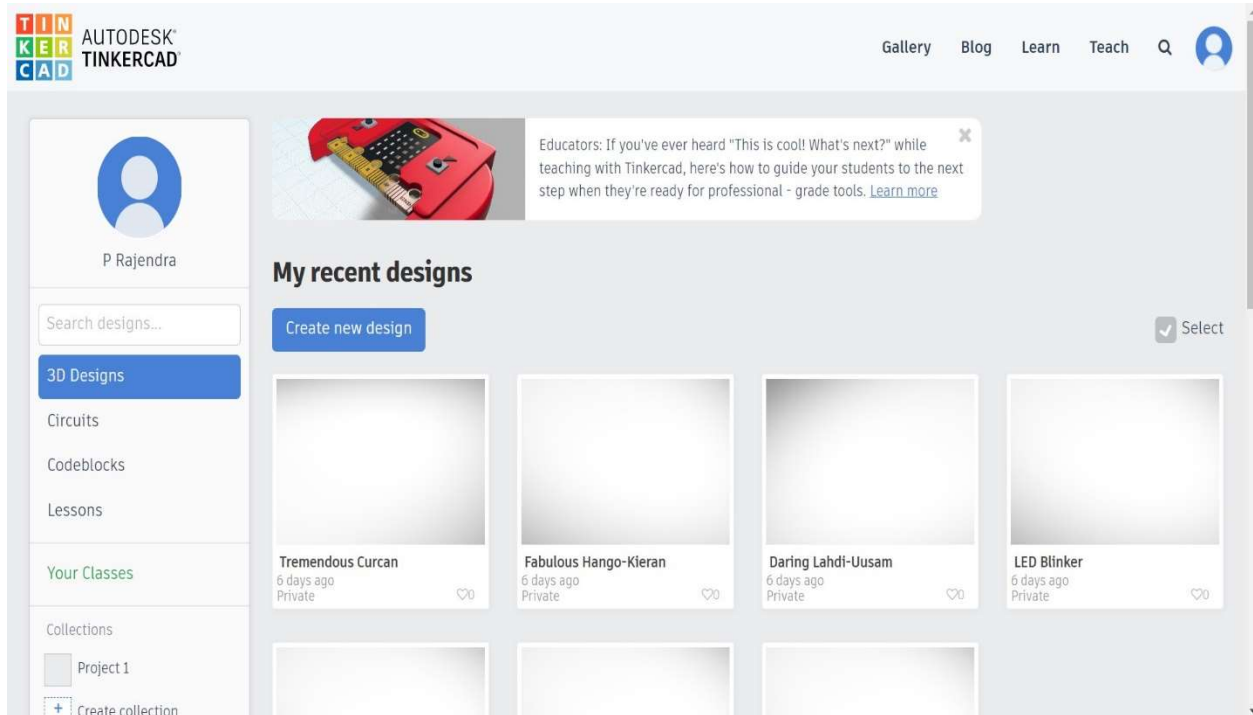
1. Open the tinkercad software in the browser. And after that we should click on **Join Now**.



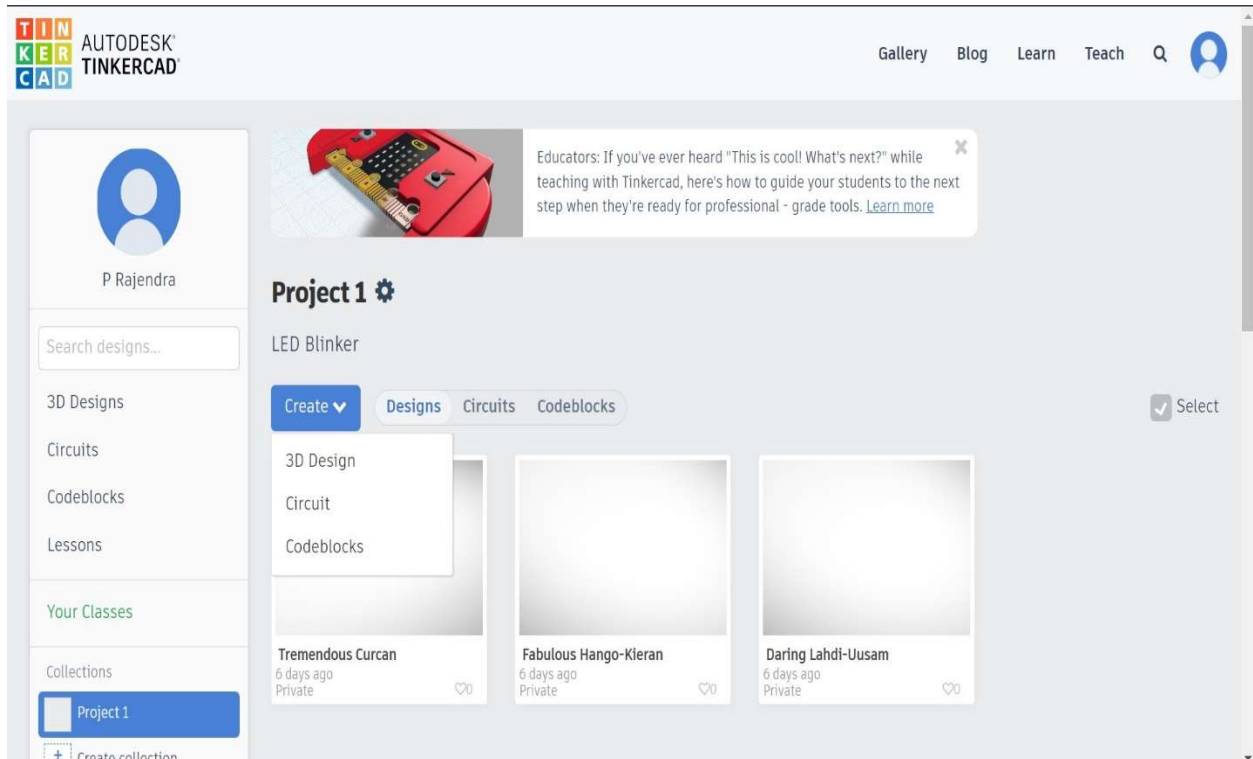
2. Create a new account in www.tinkercad.com or login with existing gmail account.



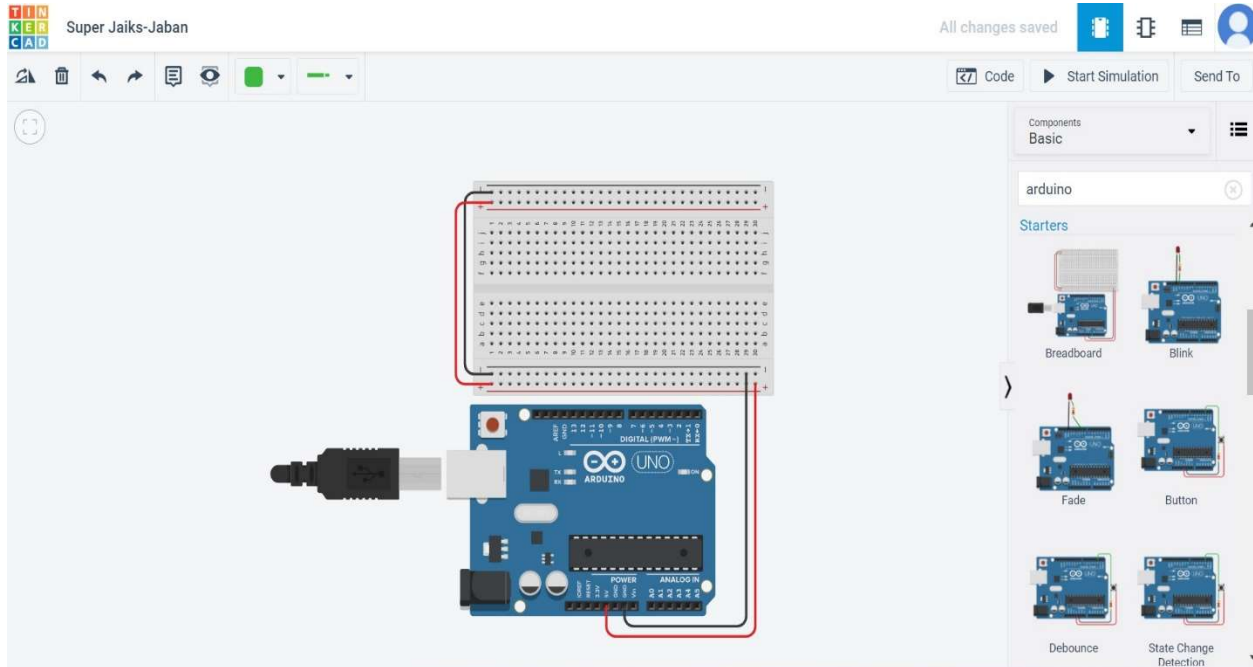
3. Click on go to create Collection and create a new collection.



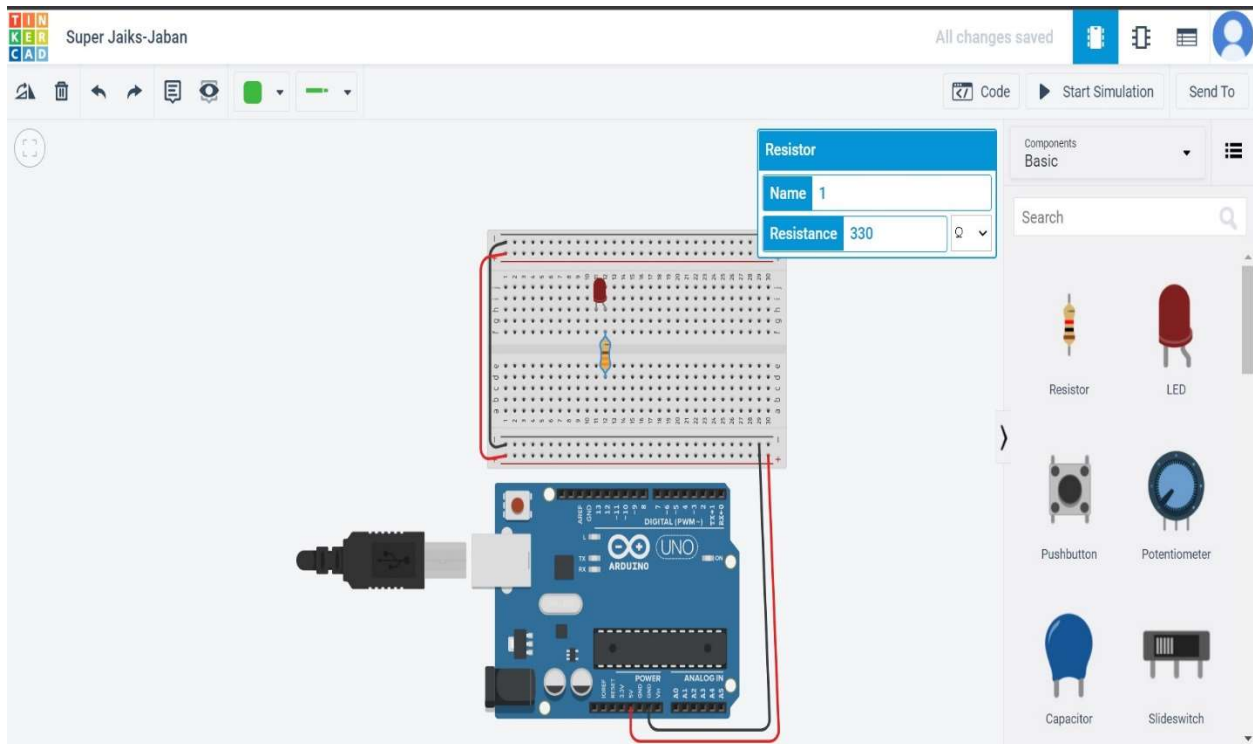
4. Go to create menu and select circuit



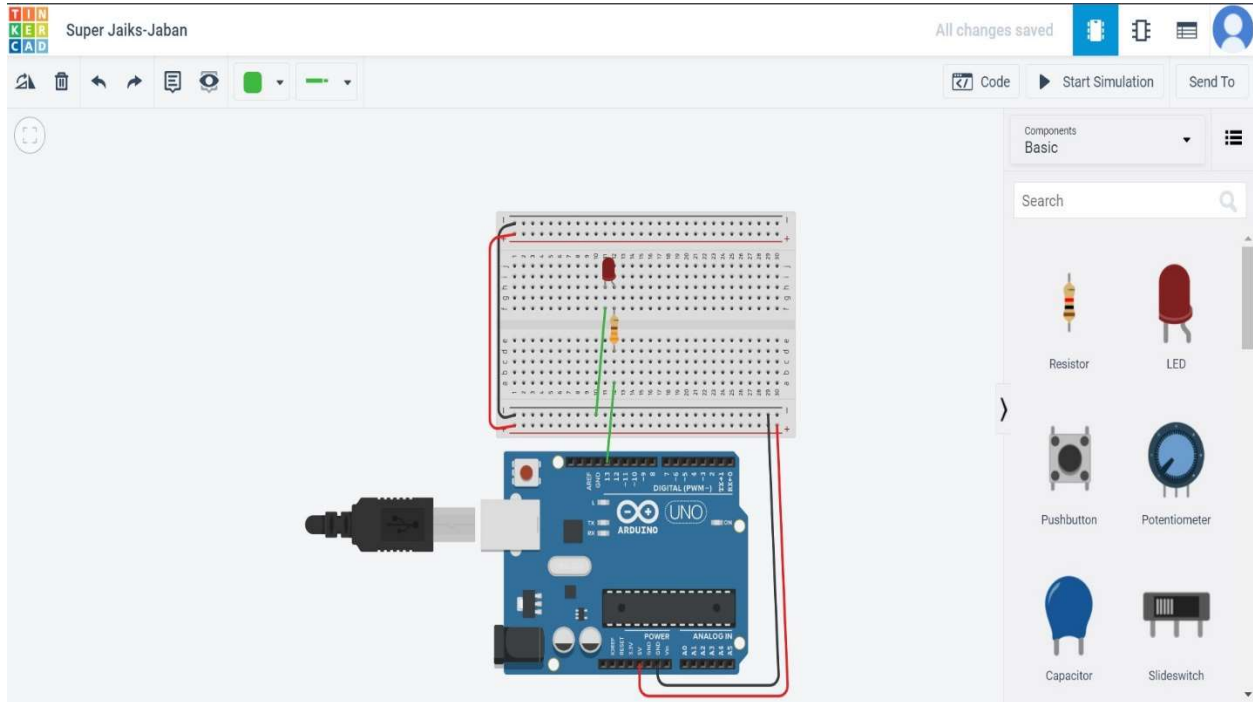
4. Select the Arduino and breadboard and place it in the design area.



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

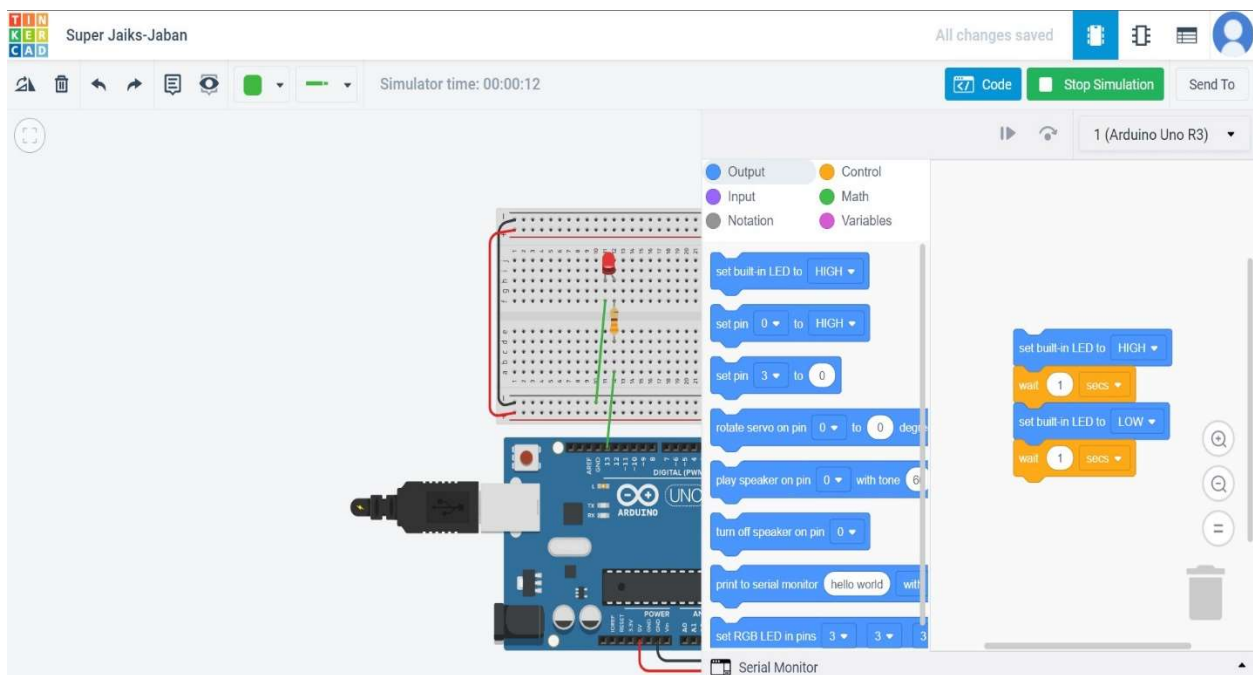


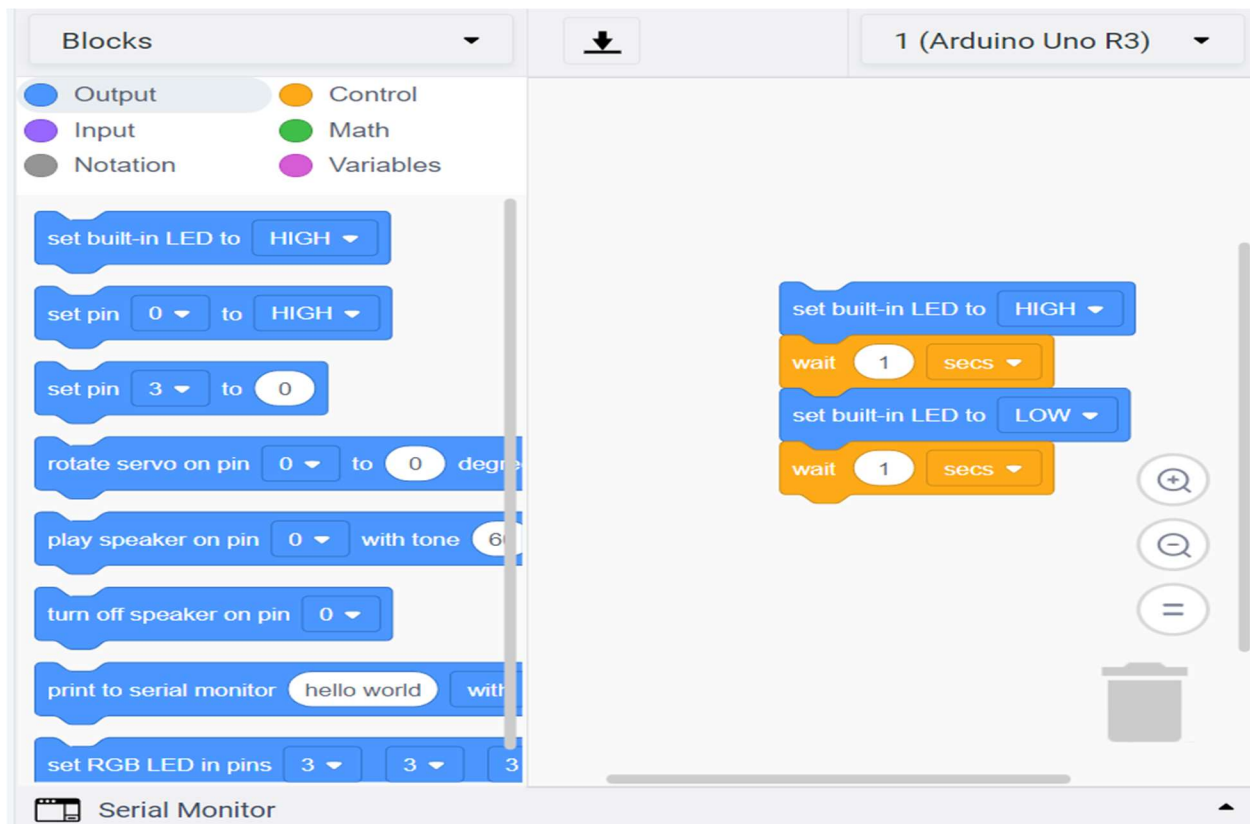
6. Attach the LED to an output pin of the Arduino D13.



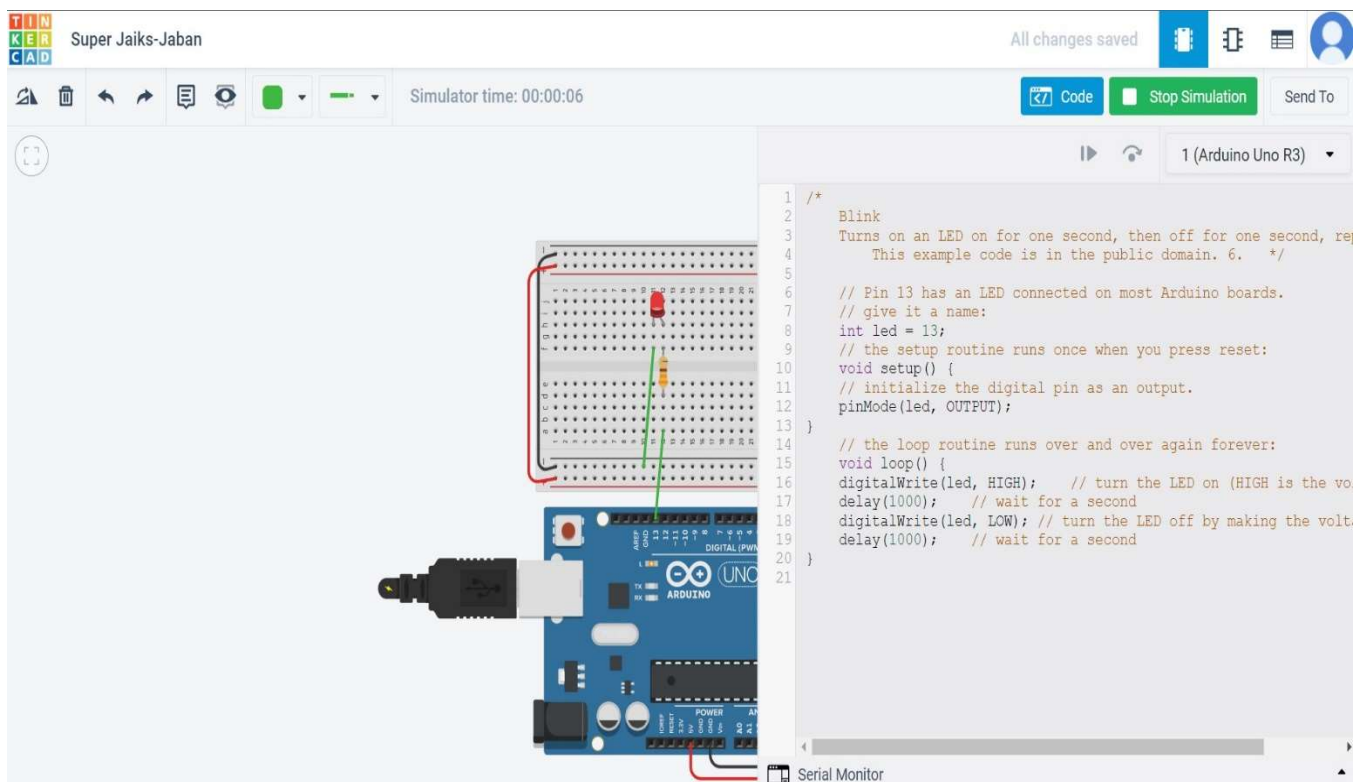
7. Once the circuit connection are ready, the Arduino can be done in three ways.

i. Using code blocks:





ii. Using Text Program:



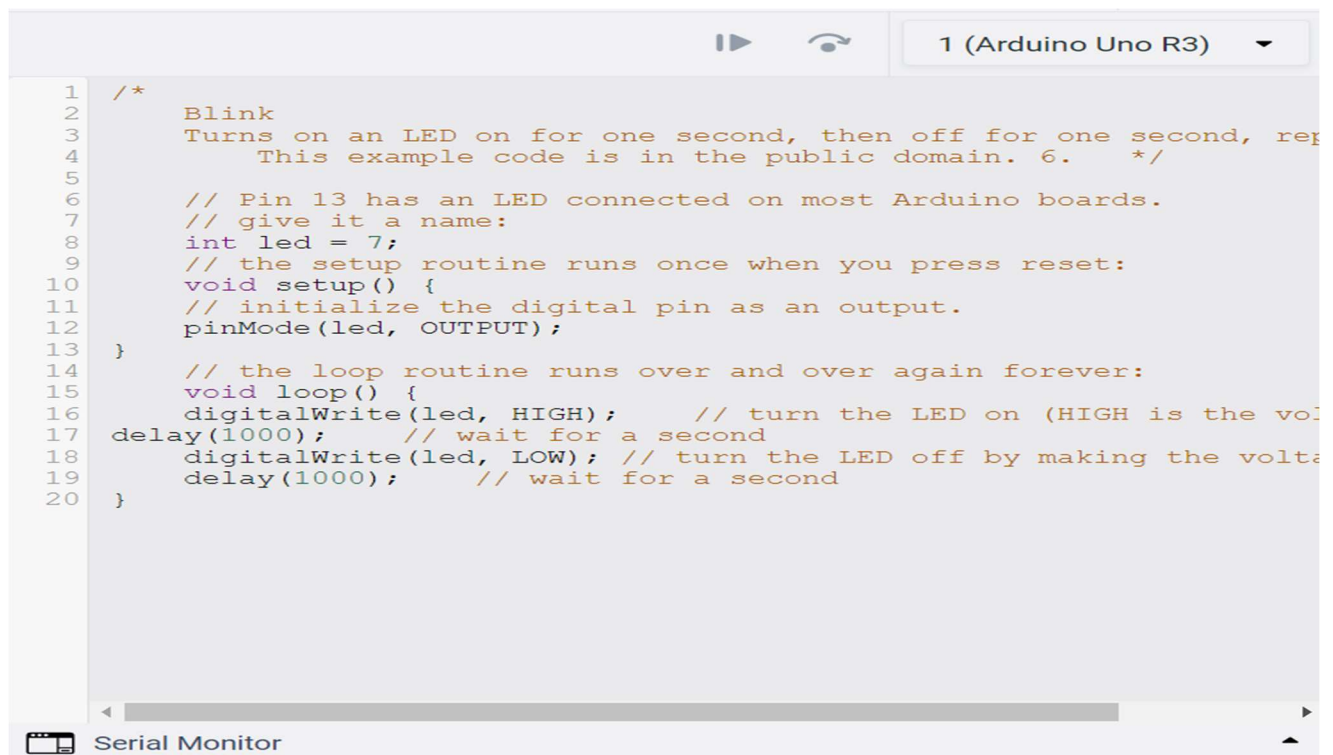
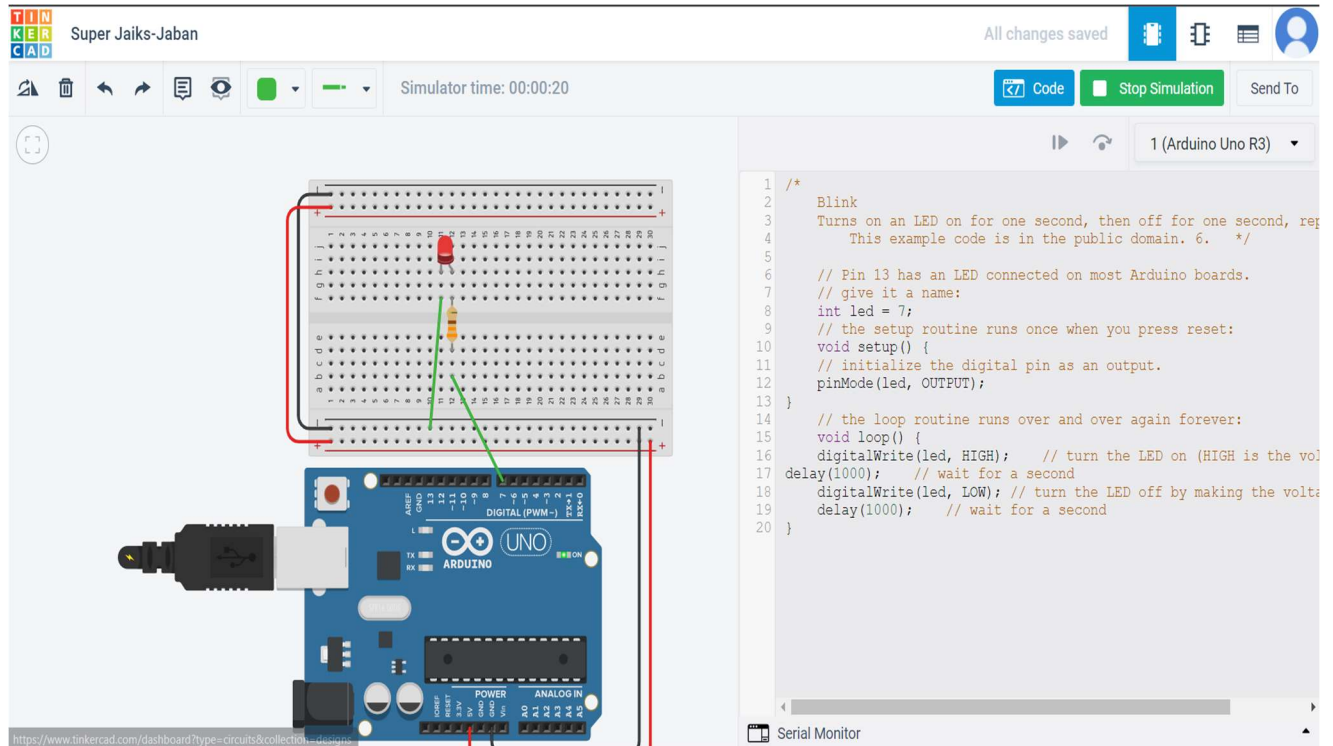

```
1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4  This example code is in the public domain. 6.  */
5
6  // Pin 13 has an LED connected on most Arduino boards.
7  // give it a name:
8  int led = 13;
9  // the setup routine runs once when you press reset:
10 void setup() {
11   // initialize the digital pin as an output.
12   pinMode(led, OUTPUT);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17   digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
18   delay(1000);               // wait for a second
19   digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
20   delay(1000);               // wait for a second
21 }
```

iii. Using code blocks + text programming:

The screenshot displays the Arduino IDE interface with a hybrid programming approach. On the left, a breadboard with an Arduino Uno is shown. The central workspace contains a sequence of code blocks: 'set built-in LED to HIGH', 'set pin 0 to HIGH', 'set pin 3 to 0', 'rotate servo on pin 0 to 0 degrees', 'play speaker on pin 0 with tone 6', 'turn off speaker on pin 0', 'print to serial monitor hello world with', and 'set RGB LED in pins 3 3 3'. To the right, the C++ code editor shows the corresponding text code for the LED and speaker functions. The top bar indicates 'All changes saved' and 'Simulator time: 00:00:33'. The bottom bar shows the 'Serial Monitor' tab.

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

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:



Result: The Controlling of LED with Arduino Board has successfully verified using tinkercad software.