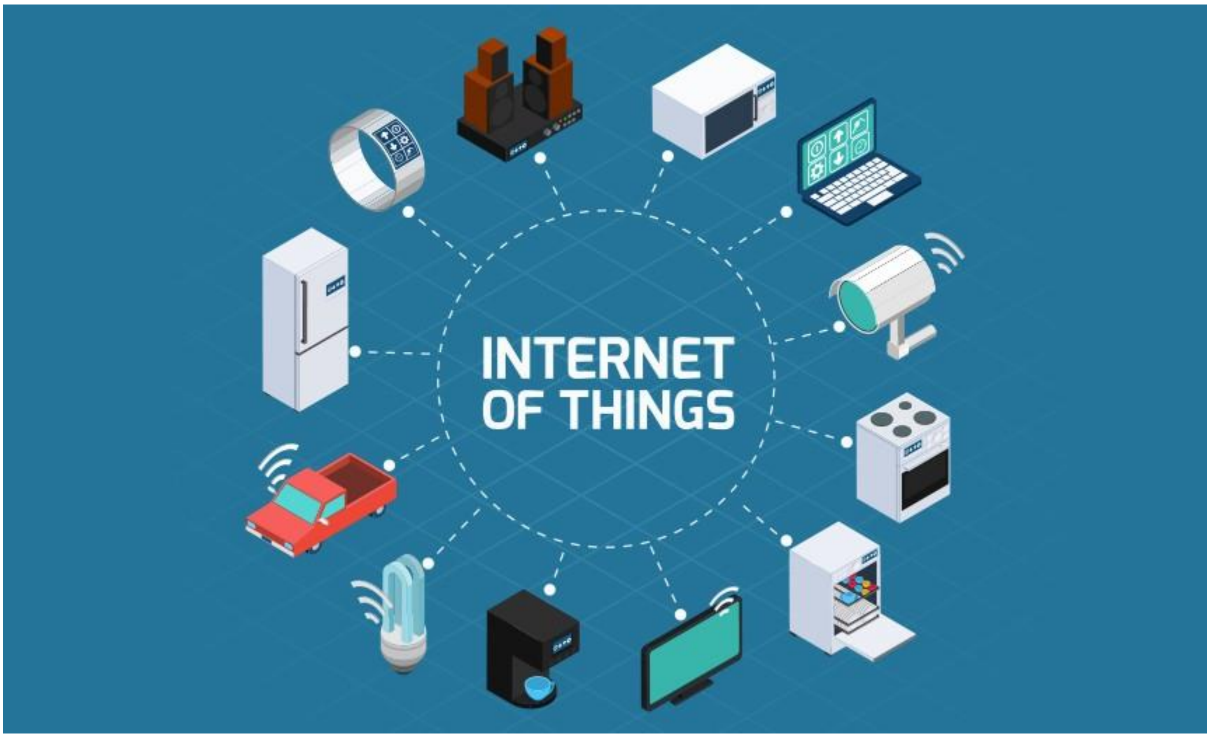
PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

IOT LAB Programs



Name:

Id Number:

Section: 5BCA

Course code and title: CSA 3005 -Internet Of Things

Table Of Contents:

ID Number: 5BCA **NAME**

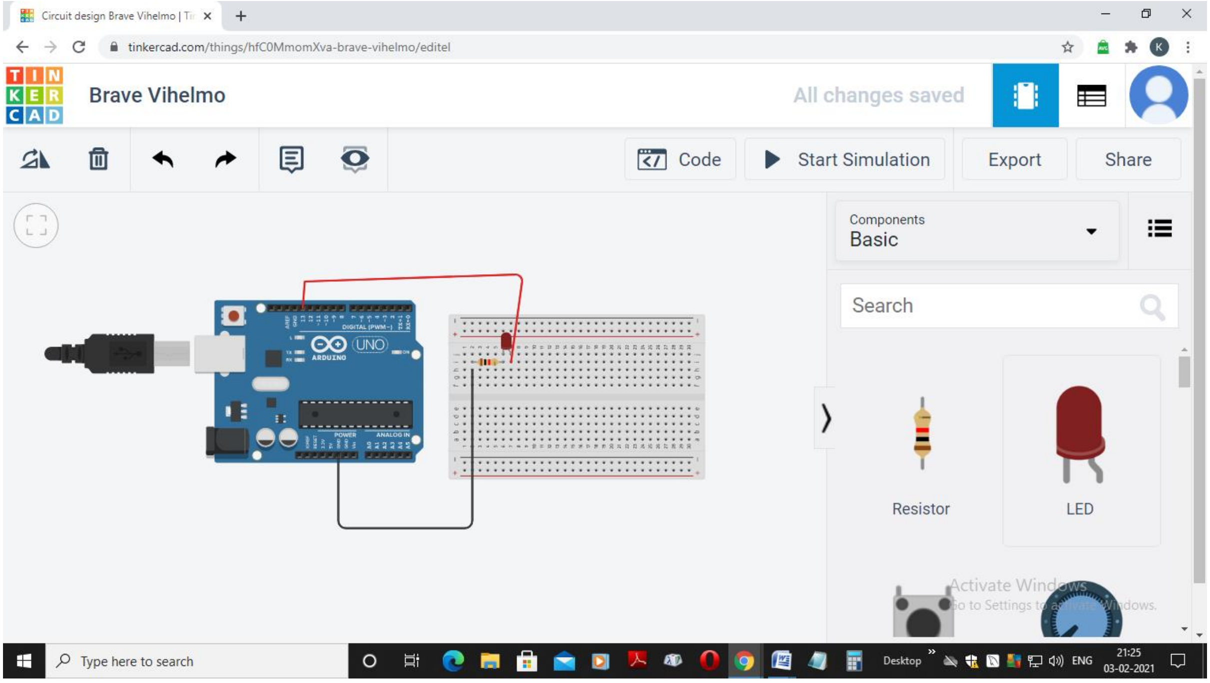
Experiment no:1 **DATE:**

Aim of the experiment: To blink an LED on Arduino Uno and to verify the result on Arduino IDE.

Components required: To blink an LED on Arduino Uno the following components are required.

- Arduino Uno R3
- 1 Led
- 1k Ω Resistor
- Small Breadboard
- Jumper Wires

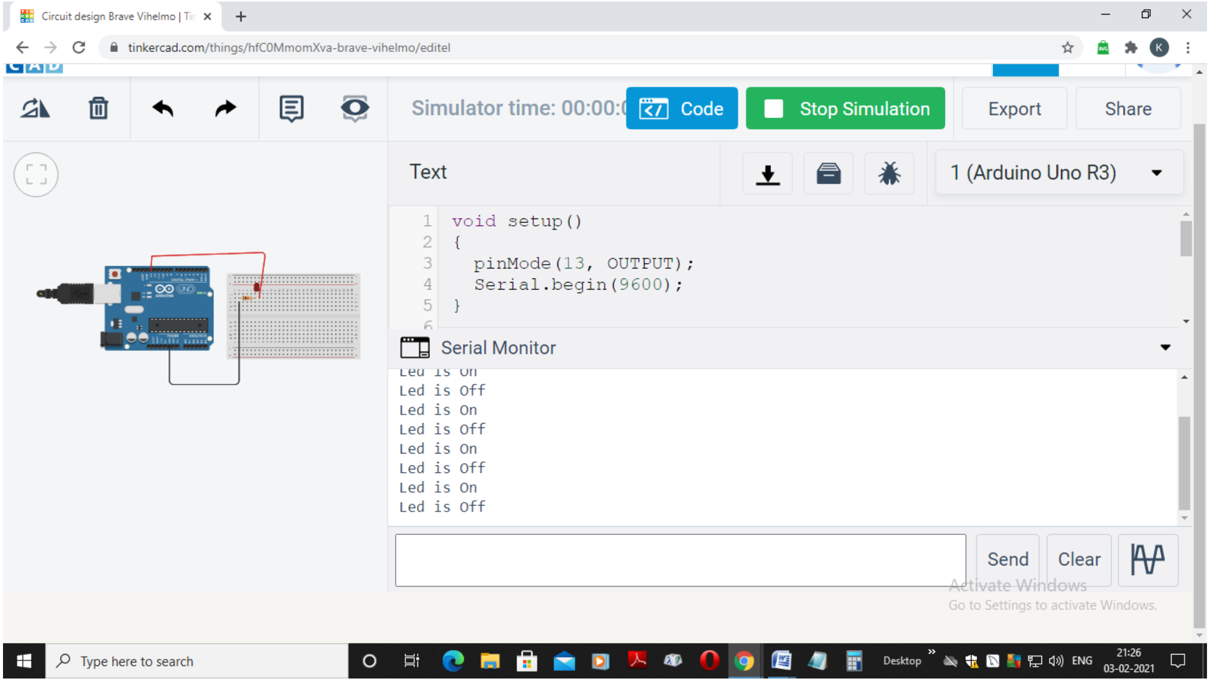
Initial circuit design:



Arduino sketch:

```
void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}


void loop()
{
  digitalWrite(13, HIGH);
  Serial.println("Led is On");
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  Serial.println("Led is Off");
  delay(1000); // Wait for 1000 millisecond(s)
}
```

<u>Output Screenshot:</u>



```
void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
```

```
Led is On
Led is Off
Led is On
Led is Off
Led is On
Led is Off
Led is On
Led is Off
```

<u>Outcome:</u> Led was blinked successfully for every one second

ID Number: 5BCA **NAME**

1(i) **DATE:**

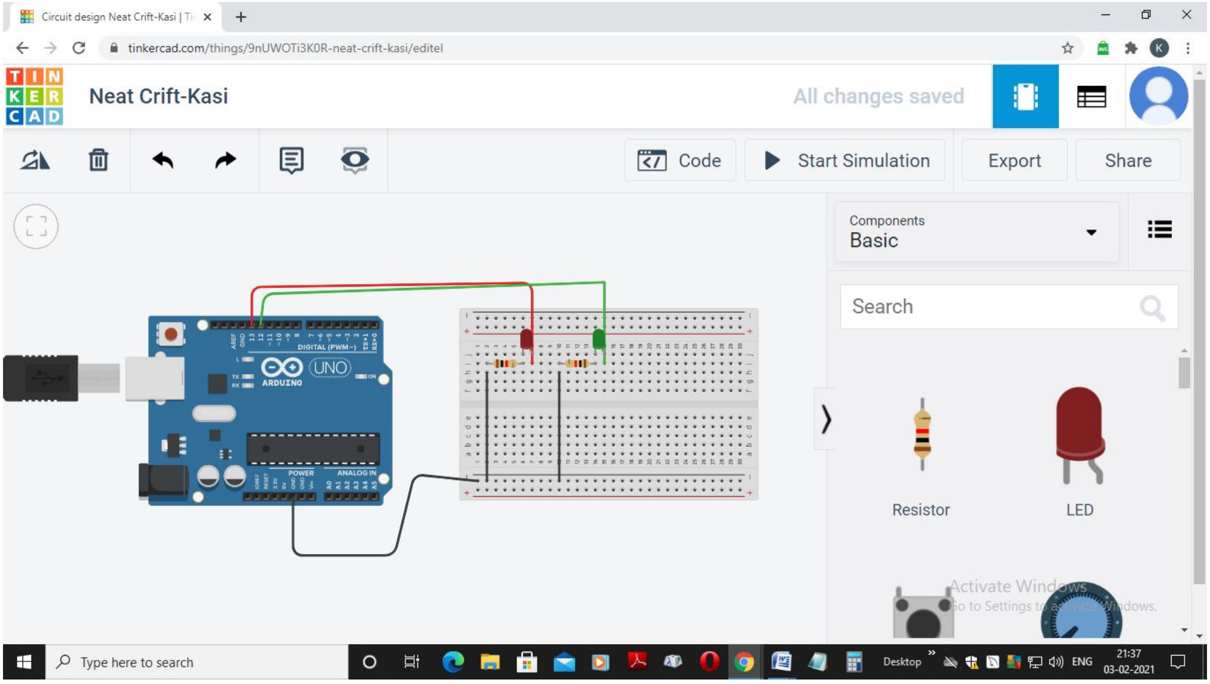
Additional programs:

Aim of the experiment: To blink two LED's on Arduino Uno and to verify the result on Arduino IDE.

Components required: To blink an LED on Arduino Uno the following components are required.

- Arduino Uno R3
- 2 Led
- 2 1k Ω Resistor
- Small Breadboard
- Jumper Wires

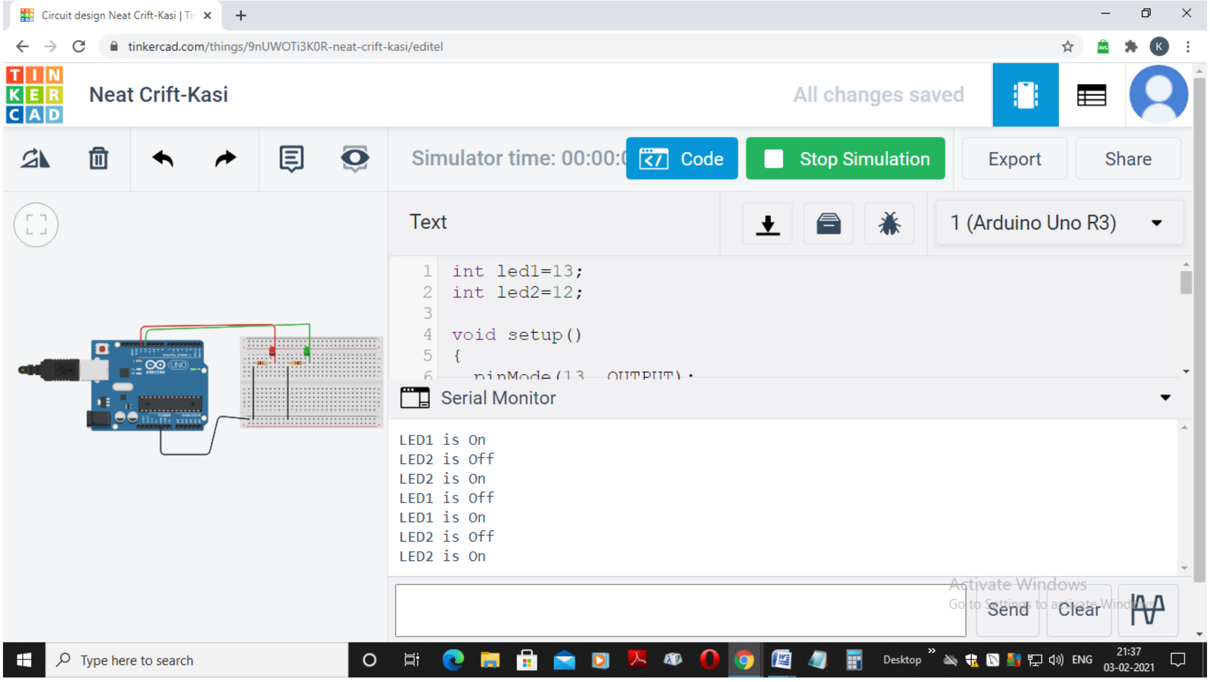
Initial circuit design:



Arduino sketch:

```
int led1=13;

int led2=12;

void setup()

{

  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(led1, HIGH);
  Serial.println("LED1 is On");
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(led2, LOW);
  Serial.println("LED2 is Off");
  digitalWrite(led2, HIGH);
  Serial.println("LED2 is On");
  delay(1000);
  digitalWrite(led1, LOW);
  Serial.println("LED1 is Off");
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Output Screenshot:



```
int led1=13;
int led2=12;

void setup()
{
  pinMode(13, OUTPUT);
```

```
LED1 is On
LED2 is Off
LED2 is On
LED1 is Off
LED1 is On
LED2 is Off
LED2 is On
```

Outcome: Two LED’s were successfully blinked alternatively for every one second.

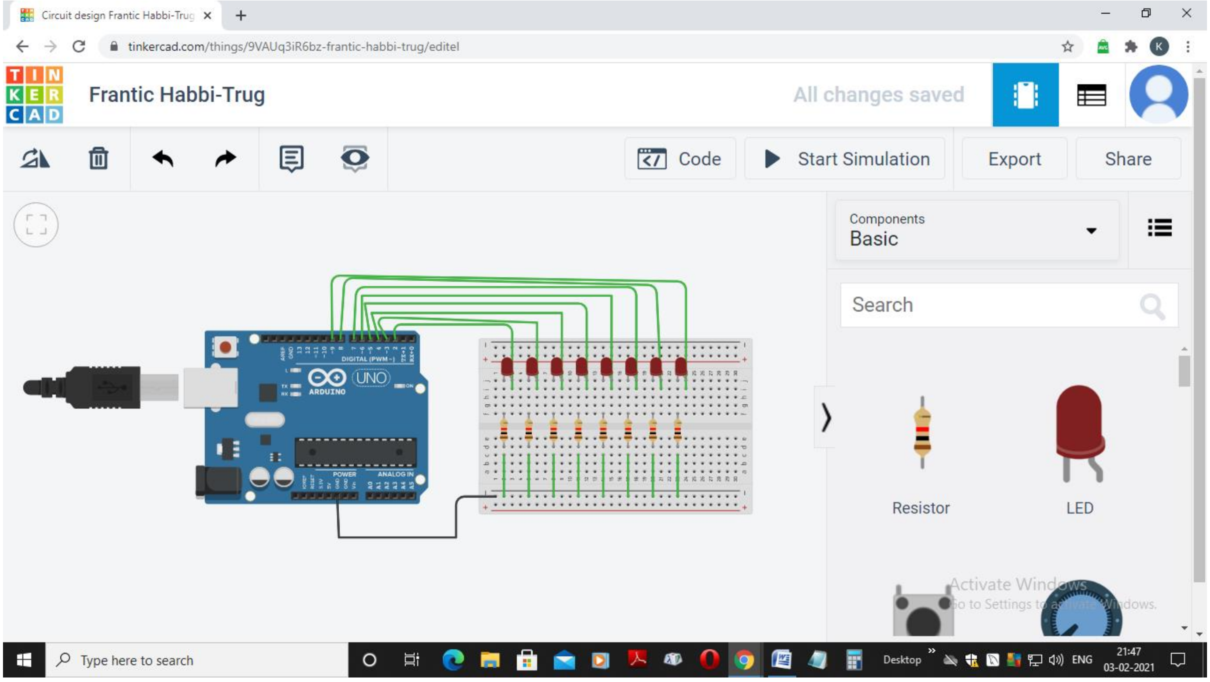
1(ii) **DATE:**

Aim of the experiment: To blink odd and even LED’s on Arduino Uno and to verify the result on Arduino IDE.

Components required: To blink odd and even LED’s on Arduino Uno the following components are required.

- Arduino Uno R3
- 8 Led
- 8 1k Ω Resistor
- Small Breadboard
- Jumper Wires

Initial circuit design:



Arduino sketch:

int del=1000; // variable define the delay

void setup()

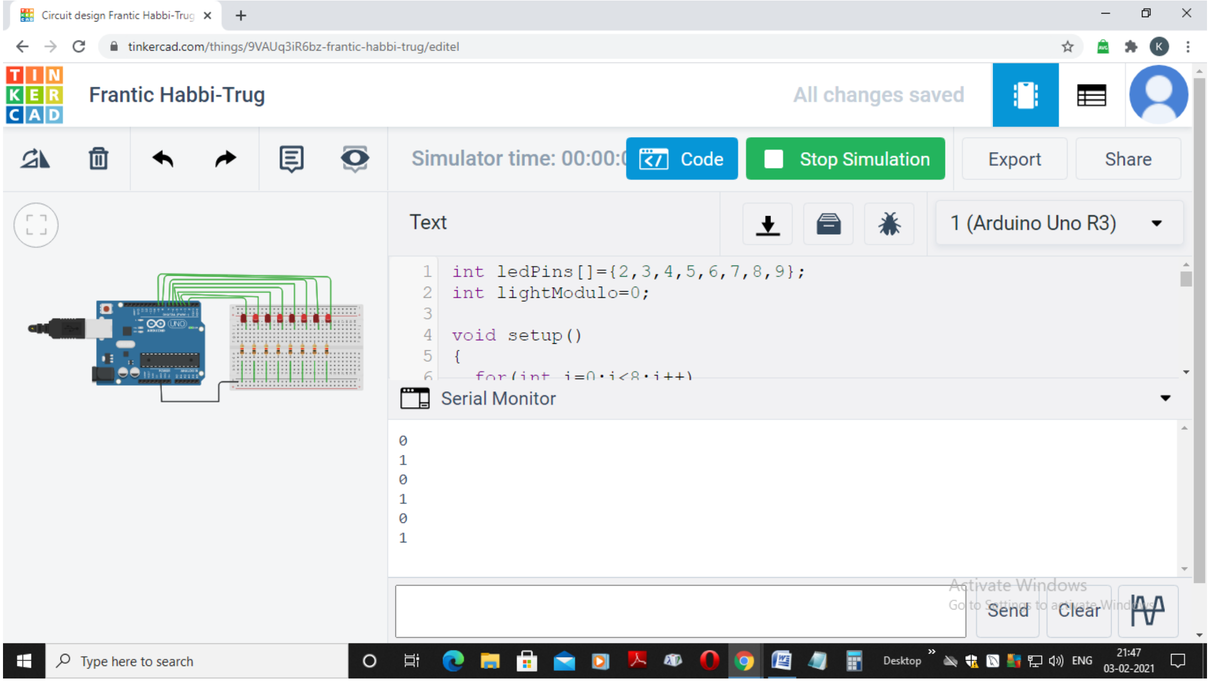
{

// make pins 2, 3, 4 and up to 8 as digital output pins

```
 // for loop to initialize all pins
 for(int i=2; i<=8; i++)
  {
    pinMode(i,OUTPUT); // declare pins as a output
  }

}

void loop()
{
  //  for loop makes an even number of digital output pins digital high
  for(int i=2; i<=8; i++)
  {
            if(i%2==0)
            {
            digitalWrite(i,HIGH);   delay(del);
            digitalWrite(i,LOW);   delay(del);
             }
  }/* end of for loop */

// for loop makes an odd number of digital output pins digital high
for(int i=2; i<=8; i++)
          {
            if(i%2==1)
```

```
        {
          digitalWrite(i,HIGH);  delay(del);
          digitalWrite(i,LOW);  delay(del);
        }
 }/* end of for loop */
}/* end of main loop */
```

<u>Output Screenshot:</u>



```
int ledPins[]={2,3,4,5,6,7,8,9};
int lightModulo=0;

void setup()
{
  for(int i=0;i<8;i++)
```

```
0
1
0
1
0
1
```

<u>Outcome</u>: Odd and even LED's were blinked successfully for every one second

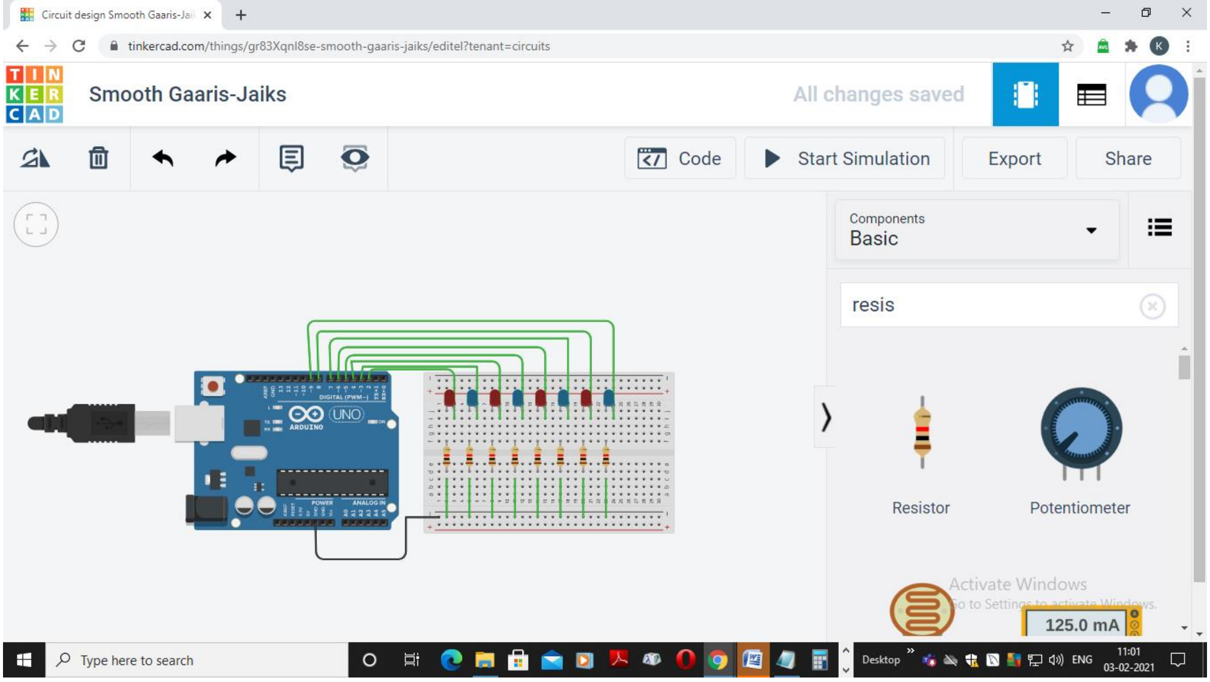
1(iii) **DATE:**

<u>Aim of the experiment:</u> To scroll LED's on Arduino Uno and to verify the result on Arduino IDE.

Components required: To scroll LED's on Arduino Uno the following components are required.

- Arduino Uno R3
- 8 Led
- 8 1k Ω Resistor
- Small Breadboard
- Jumper Wires

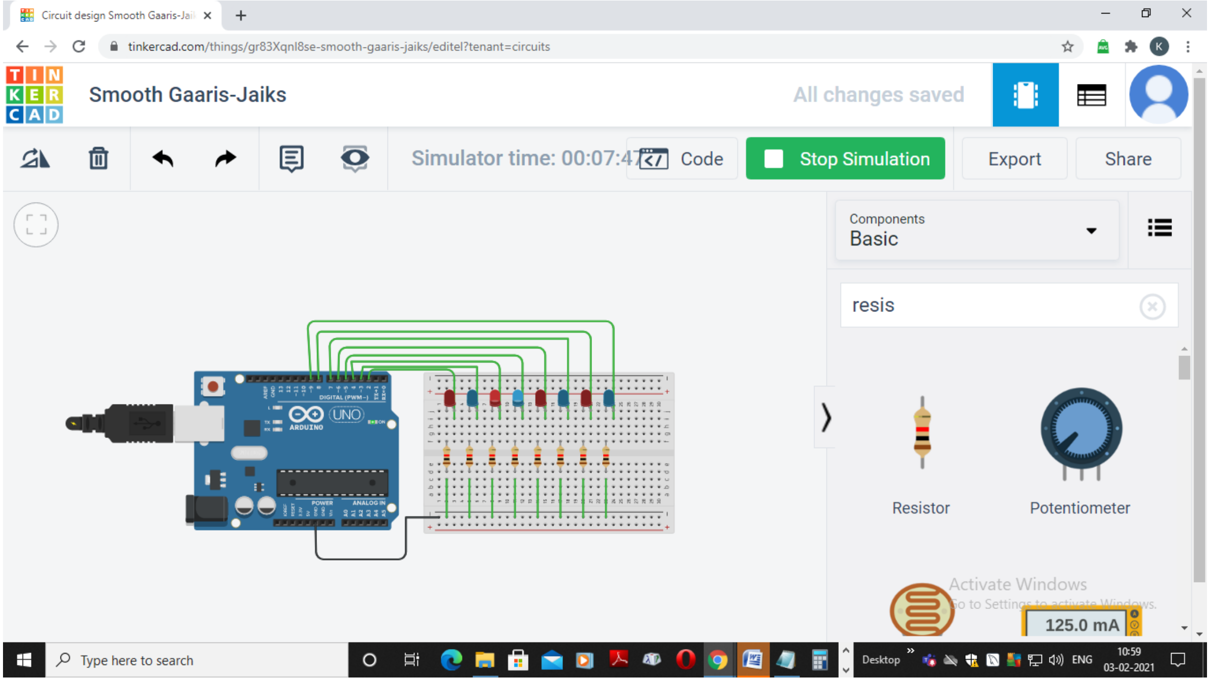
Initial circuit design:



Arduino sketch:

```
int ledPins[]={2,3,4,5,6,7,8,9};

void setup()

{

   for(int i=0;i<8;i++)

  {

   pinMode(ledPins[i], OUTPUT);
  }
  Serial.begin(9600);
}


void loop()
{
   for(int i=0;i<8;i++)
          {
              digitalWrite(ledPins[i],HIGH);     delay(1000);
              digitalWrite(ledPins[i],LOW);
   }
  for(int i=7;i>=0;i--)
          {
            digitalWrite(ledPins[i],HIGH);     delay(1000);
           digitalWrite(ledPins[i],LOW);
   }
}
```

<u>Output Screenshot:</u>



Outcome: Led was scrolled successfully for every one second

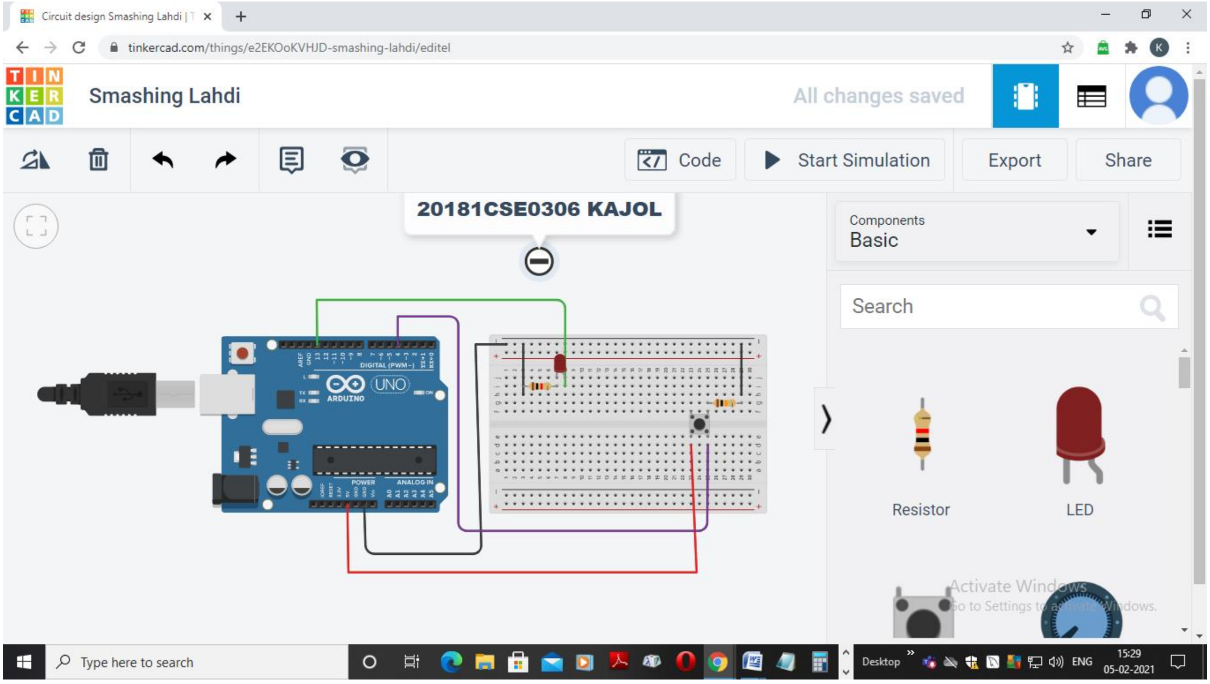
Experiment no:2 DATE:

Aim of the experiment: Interfacing of Arduino Uno with LED and switch and to control LED using switch.

Components required: To interface an Arduino Uno with LED and switch the following components are required.

- Arduino Uno R3
- Led
- 1k **Ω** and 10k **Ω** Resistor
- Small Breadboard
- Jumper Wires
- Push button

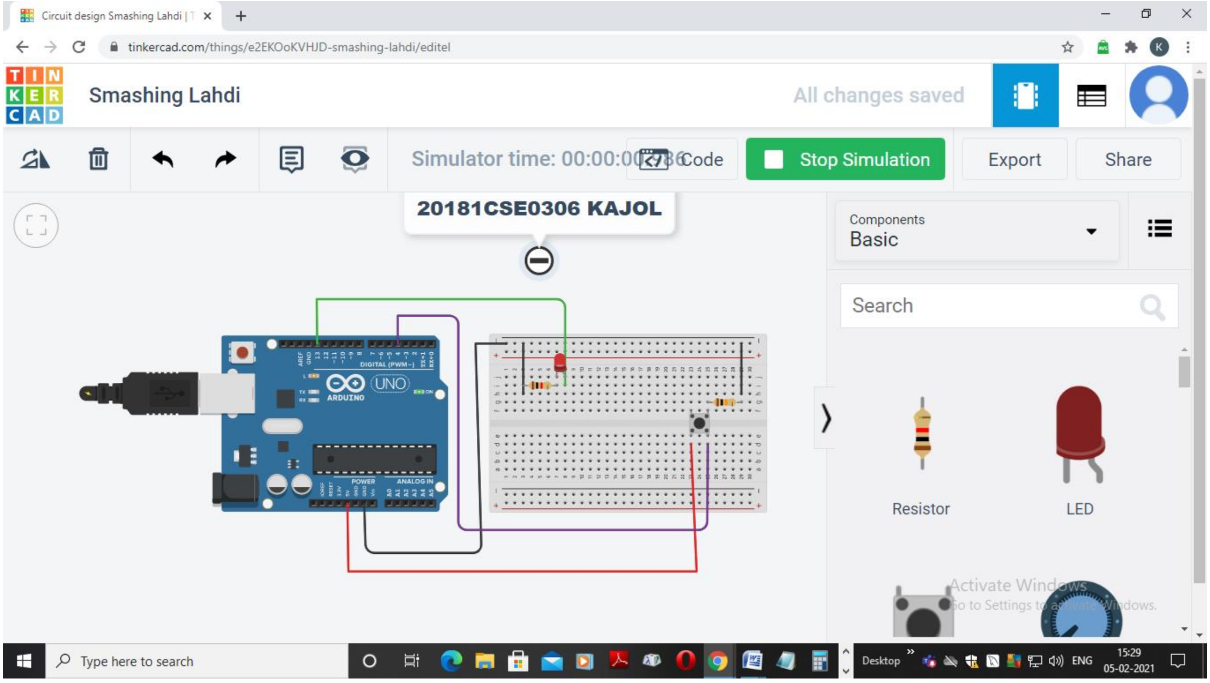
Initial circuit design:



Arduino sketch:

```
void setup()

{
  pinMode(13, OUTPUT);
  pinMode(12, INPUT);
  Serial.begin(9600);
}

void loop()
{
  if(digitalRead(12)==1)
  {
  digitalWrite(13, HIGH);
  Serial.println("LED IS ON.....");
  delay(1000); // Wait for 1000 millisecond(s)
  }
  digitalWrite(13, LOW);
  Serial.println("LED IS OFF.....");
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Output Screenshot:



Outcome: Arduino Uno was successfully interfaced with LED and switch and the LED was controlled using switch.

2(i) **DATE:**

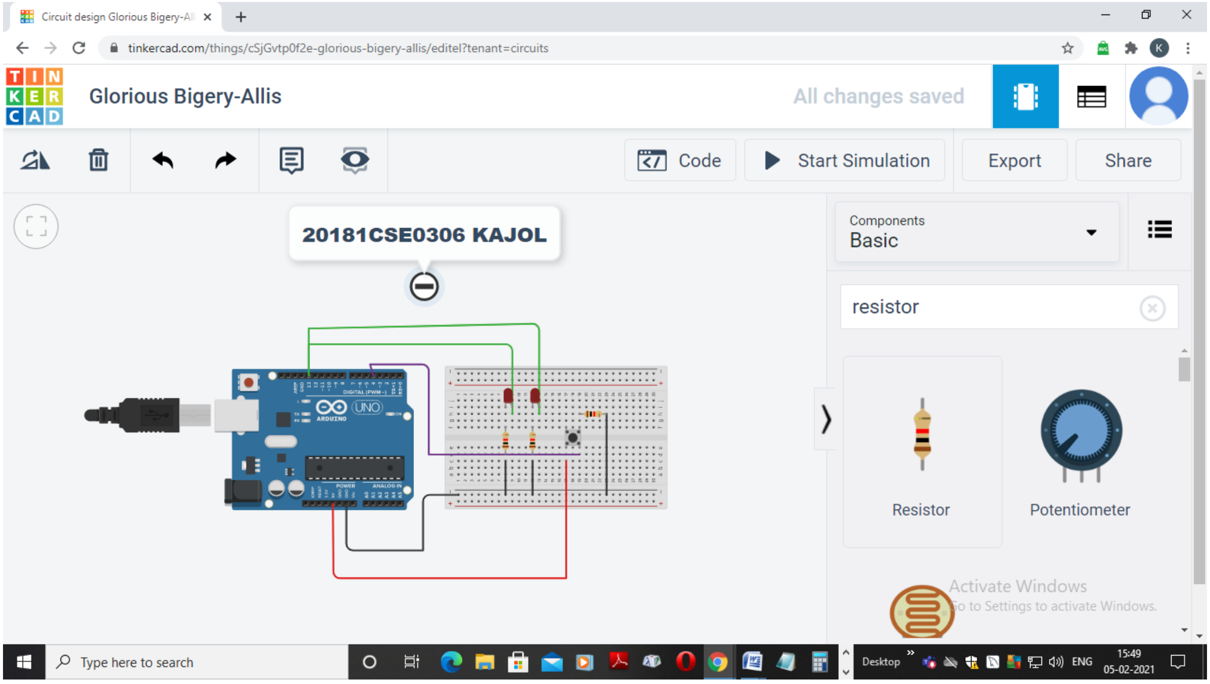
Additional programs:

Aim of the experiment: To control multiple LED's using single switch.

Components required: To control multiple LED's using single switch the following components are required.

- Arduino Uno R3
- Led
- 1k **Ω** and 10k **Ω** Resistor
- Small Breadboard
- Jumper Wires
- Push button

Initial circuit design:



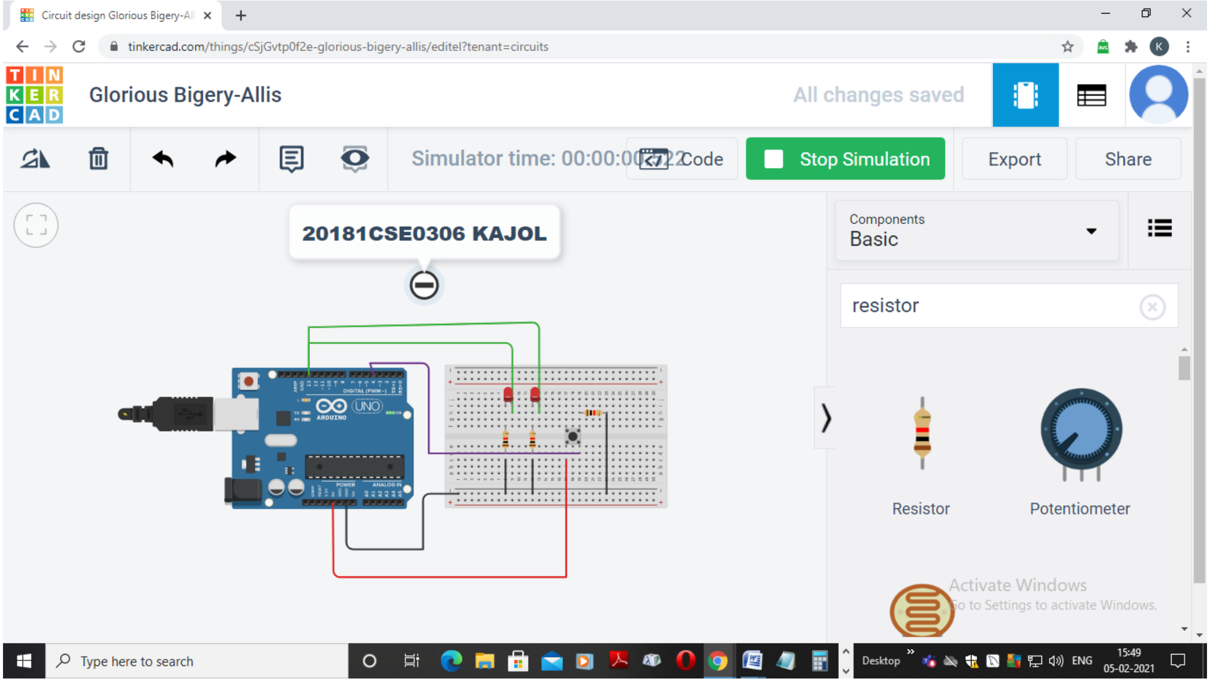
Arduino sketch:

int button_pin=4;

void setup()

```
{
  pinMode(button_pin,INPUT);
  pinMode(13,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Controlling LED through push button");
  int button;
  button=digitalRead(button_pin);

  if(button==HIGH)
  {
    digitalWrite(13,HIGH);
    Serial.println("LED1 is ON");
    digitalWrite(13,HIGH);
    Serial.println("LED2 is ON");

  }
  else
  {
    digitalWrite(13,LOW);
    Serial.println("LED2 is OFF");
    digitalWrite(13,LOW);

    Serial.println("LED2 is OFF");


  }

}
```

Output Screenshot:



Outcome: Multiple LED's were controlled successfully using single switch .

2(ii) **DATE:**

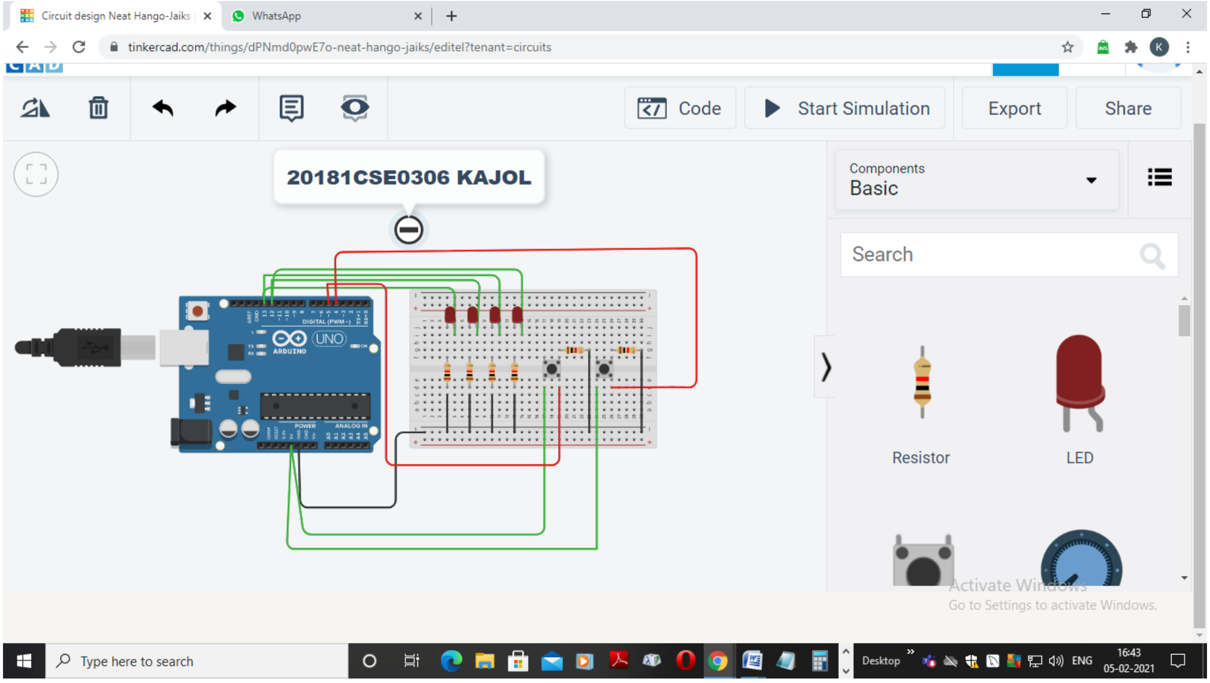
Additional program:

Aim of the experiment: To control multiple LED's using multiple switch.

Components required: To control multiple LED's using single switch the following components are required.

- Arduino Uno R3
- Led
- 1k Ω and 10k Ω Resistor
- Small Breadboard
- Jumper Wires
- Push button

Initial circuit design:

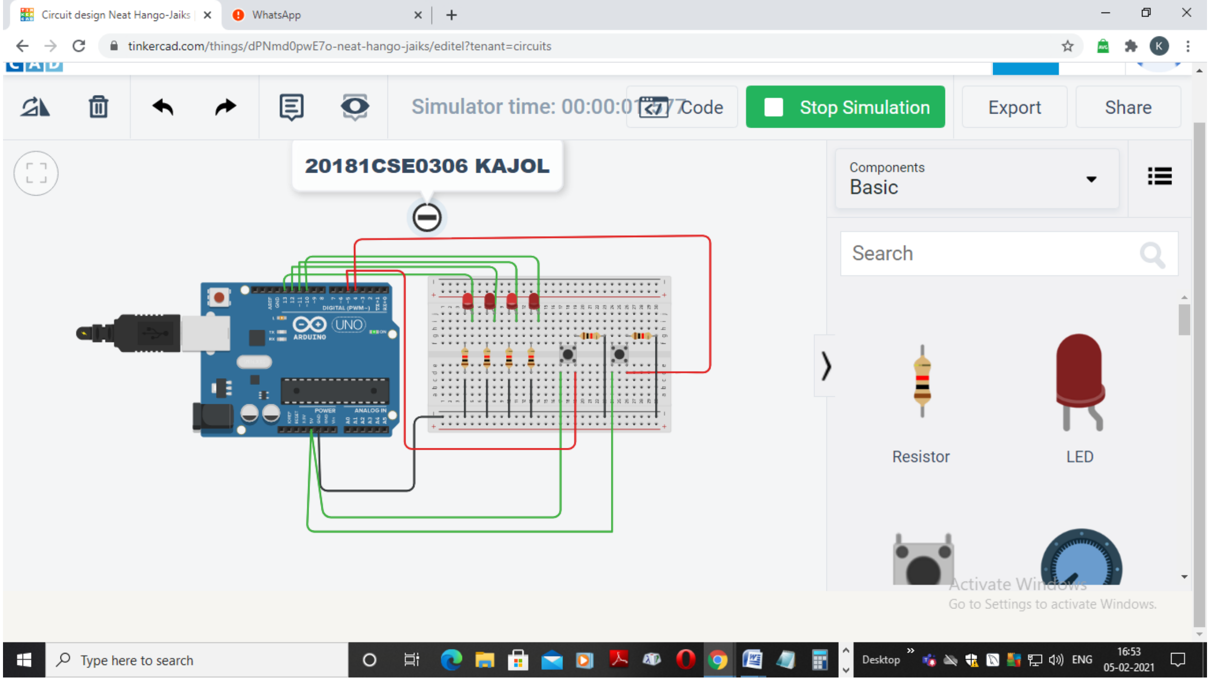


Arduino sketch:

```
int button_pin1=4;
int button_pin2=5;

void setup()

{

  pinMode(13, OUTPUT);

  pinMode(button_pin1,INPUT);

  pinMode(button_pin2,INPUT);

  pinMode(12,OUTPUT);

  Serial.begin(9600);

}


void loop()

{

  int button1,button2;

  button1=digitalRead(button_pin1);

  button2=digitalRead(button_pin2);

  if (button1==HIGH)

  {

    digitalWrite(13, HIGH);

  }

  else

  {

    digitalWrite(13,LOW);

  }

  if (button2==HIGH)
```

```
 {
   digitalWrite(12,HIGH);
 }
 else
 {
   digitalWrite(12,LOW);
 }
}
```

Output Screenshot:



Outcome: Multiple LED's were controlled successfully using multiple switch.

Additional program

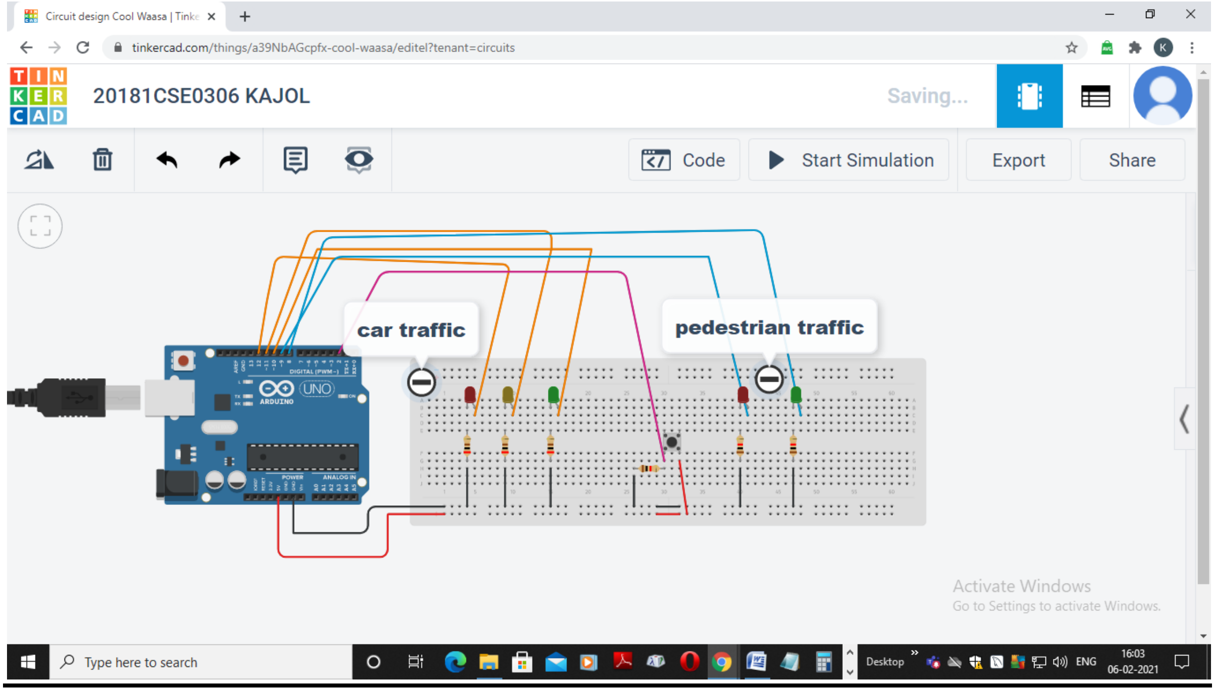
DATE:

Aim of the experiment: Arduino program to implement traffic control system.

Components required: to implement traffic control system the following components are required.

- Arduino Uno R3 board
- Power cable(1)
- Breadboard(1)
- Led (5)
  - ➢ 2 Red LED's
  - ➢ 1 Yellow LED
  - ➢ 2 Green LED's
- 200 Ω Resistor (6)
- Jumper Wires
- Tactile Switch (Push button)

Initial circuit design:

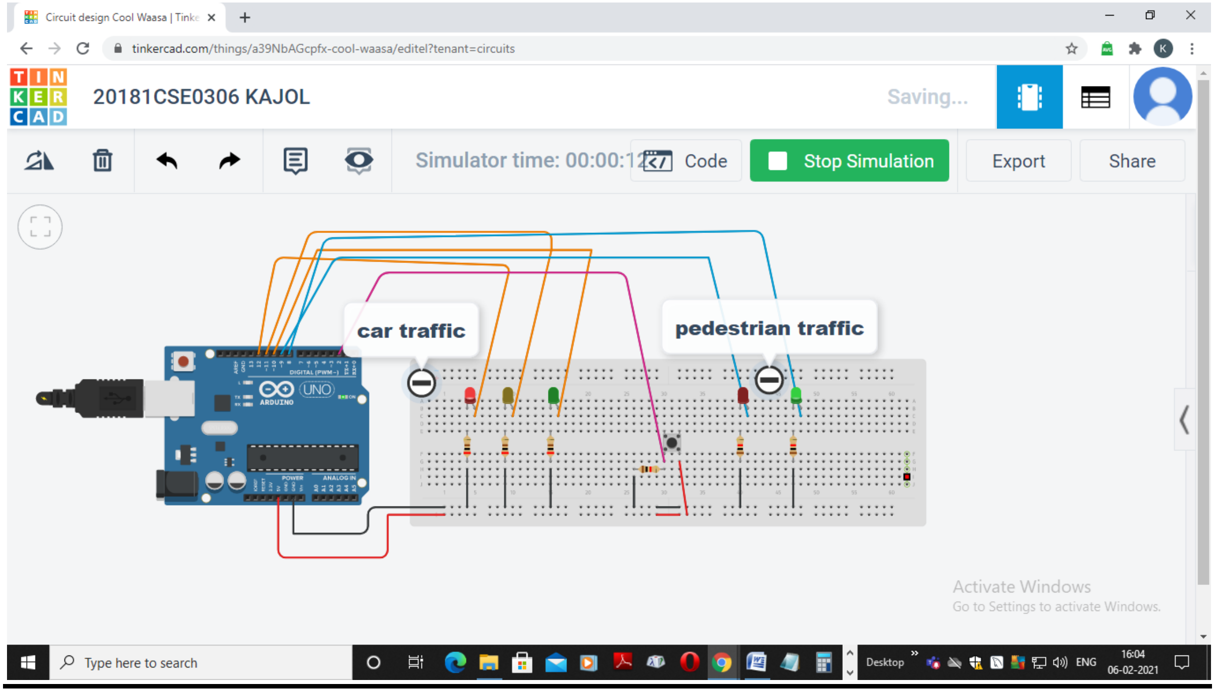


Arduino sketch:

```
int carRed =12;

int carYellow =11;

int carGreen =10;

int pedRed =9;

int pedGreen =8;

int button=2;

int crossTime=5000;


unsigned long changeTime;


void setup()

{

  pinMode(carRed, OUTPUT);

  pinMode(carYellow, OUTPUT);

  pinMode(carGreen, OUTPUT);

  pinMode(pedRed, OUTPUT);

  pinMode(pedGreen, OUTPUT);

  pinMode(button, INPUT);


  digitalWrite(carGreen, HIGH);

  digitalWrite(pedRed, HIGH);

}
```

```
void loop()
{
  int state=digitalRead(button);
  if(state==HIGH &&(millis()-changeTime)>5000)
  {
  changeLights();
  }
}
void changeLights()
{
  digitalWrite(carGreen,LOW);
  digitalWrite(carYellow, HIGH);
  delay(2000);
  digitalWrite(carYellow, LOW);
  digitalWrite(carRed, HIGH);
  delay(1000);
  digitalWrite(pedRed, LOW);
  digitalWrite(pedGreen, HIGH);
  delay(crossTime);

  for(int x=0;x<10;x++)
  {
    digitalWrite(pedGreen, HIGH);
    delay(250);
    digitalWrite(pedGreen, LOW);
    delay(250);
  }
  digitalWrite(pedRed, HIGH);
  delay(500);
  digitalWrite(carYellow, HIGH);
  digitalWrite(carRed, LOW);
  delay(1000);
  digitalWrite(carGreen,HIGH);
  digitalWrite(carYellow, LOW);


  changeTime=millis();
}
```

Output Screenshot:



Outcome: Traffic control system was implemented successfully using Arduino Uno.

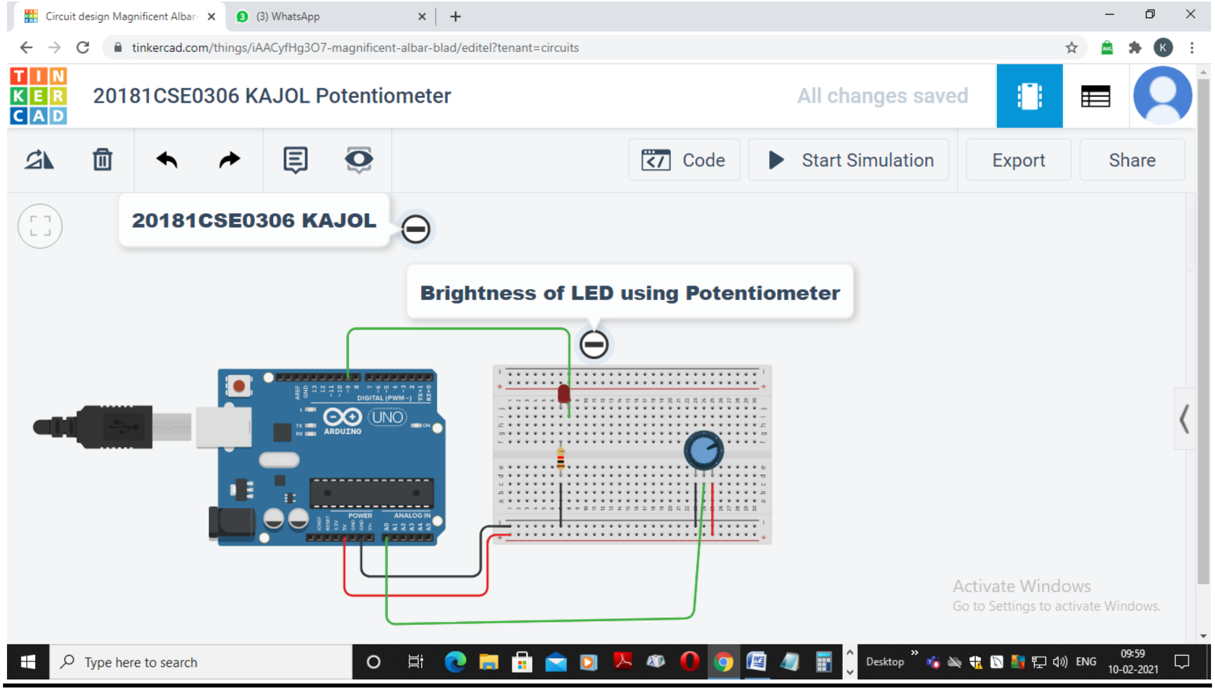
Experiment no:3 **DATE:**

Aim of the experiment: To adjust the brightness **(Fading) of an** LED using Potentiometer.

Components required:

- Arduino Uno
- Led
- Potentiometer(knob)
- Resistor
- Tinkercad simulator

Initial circuit design:


Circuit design Magnificent Albar
(3) WhatsApp
tinkercad.com/things/iAACyfHg3O7-magnificent-albar-blad/editel?tenant=circuits
20181CSE0306 KAJOL Potentiometer
All changes saved
Code
Start Simulation
Export
Share
20181CSE0306 KAJOL
Brightness of LED using Potentiometer
UNO
ARDUINO
Activate Windows
Go to Settings to activate Windows.
Type here to search
Desktop
ENG
09:59
10-02-2021

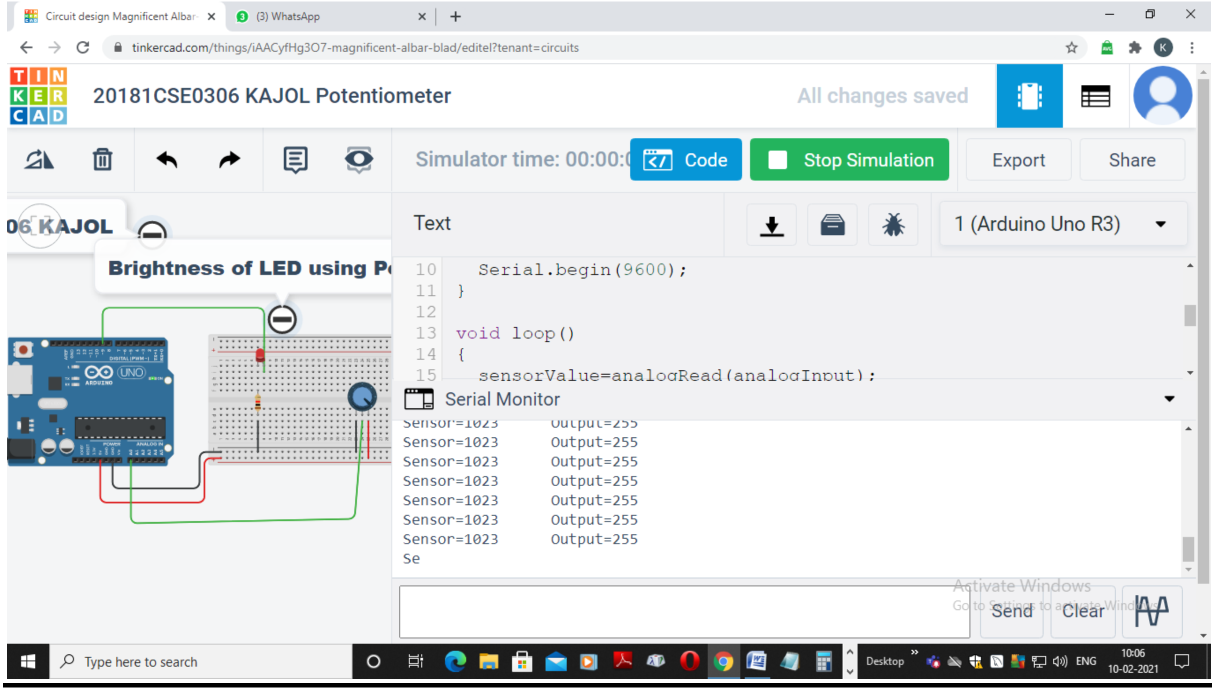
Arduino sketch:

```
const int analogInput=A0;

const int analogOutput=9;
```

```
int sensorValue=0; //wiper-read voltage from potentiometer
int outputValue=0;//value output to the PWM(analog output)



void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sensorValue=analogRead(analogInput);

  outputValue=map(sensorValue,0,1023,0,255);

  analogWrite(analogOutput,outputValue);

  Serial.print("Sensor=");
  Serial.print(sensorValue);
  Serial.print("\t Output=");
  Serial.println(outputValue);
```

}

Output Screenshot:



```
  Serial.begin(9600);
}

void loop()
{
  sensorValue=analogRead(analogInput);
```

```
Sensor=1023      Output=255
Sensor=1023      Output=255
Sensor=1023      Output=255
Sensor=1023      Output=255
Sensor=1023      Output=255
Sensor=1023      Output=255
Sensor=1023      Output=255
Se
```

Outcome: The brightness of an LED was adjusted successfully using Potentiometer.

Additional program:

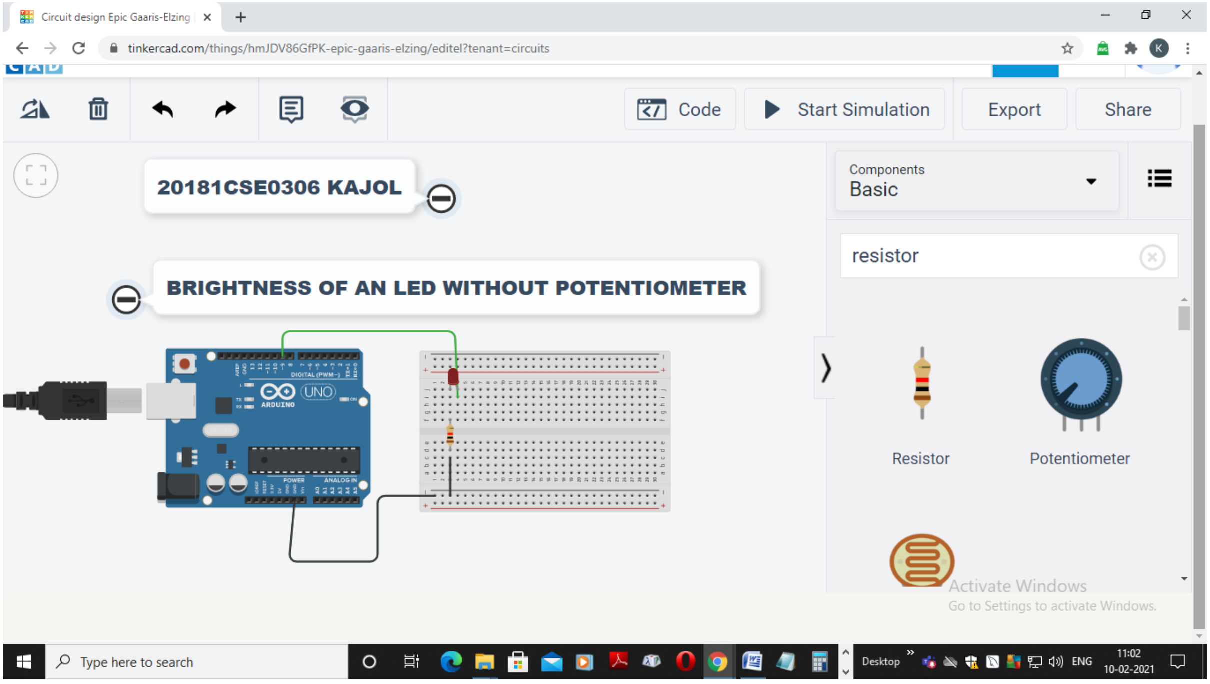
DATE:

Aim of the experiment: To adjust the brightness **(Fading) of an** LED without using Potentiometer.

Components required:

- Arduino Uno
- Led
- Resistor
- Tinkercad simulator

Initial circuit design:



Circuit design Epic Gaaris-Elzing

tinkercad.com/things/hmJDV86GfPK-epic-gaaris-elzing/editel?tenant=circuits

20181CSE0306 KAJOL

BRIGHTNESS OF AN LED WITHOUT POTENTIOMETER

Components Basic

resistor

Activate Windows
Go to Settings to activate Windows.

Arduino sketch:

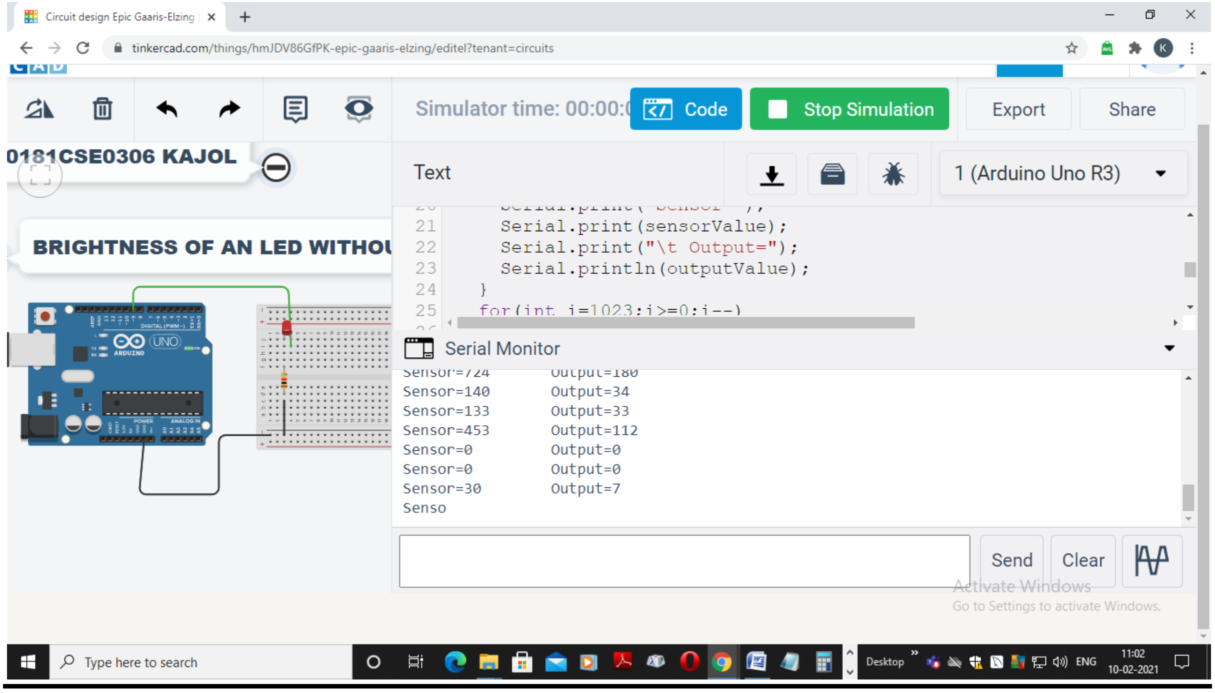
const int analogOutput=9;

int sensorValue=0;

int outputValue=0;

```
void setup()

{

  Serial.begin(9600);

}


void loop()

{

  for(int i=0;i<=1023;i++)

  {

    sensorValue=analogRead(i);

    outputValue=map(sensorValue,0,1023,0,255);


//map is a function which accepts 5 arguments map(input sensor value
,volt_value range and analog  value range;


    analogWrite(analogOutput,outputValue);


    Serial.print("Sensor=");

    Serial.print(sensorValue);

    Serial.print("\t Output=");

    Serial.println(outputValue);

  }

  for(int i=1023;i>=0;i--)

  {

    sensorValue=analogRead(i);
    outputValue=map(sensorValue,0,1023,0,255);
    analogWrite(analogOutput,outputValue);
    Serial.print("Sensor=");
    Serial.print(sensorValue);
    Serial.print("\t Output=");
    Serial.println(outputValue);
  }
}
```

Output Screenshot:



```
    Serial.print("Sensor=");
    Serial.print(sensorValue);
    Serial.print("\t Output=");
    Serial.println(outputValue);
  }
  for(int i=1023;i>=0;i--)
```

```
Sensor=724         Output=180
Sensor=140         Output=34
Sensor=133         Output=33
Sensor=453         Output=112
Sensor=0           Output=0
Sensor=0           Output=0
Sensor=30          Output=7
Senso
```

Outcome: The brightness of an LED was adjusted successfully without using potentiometer

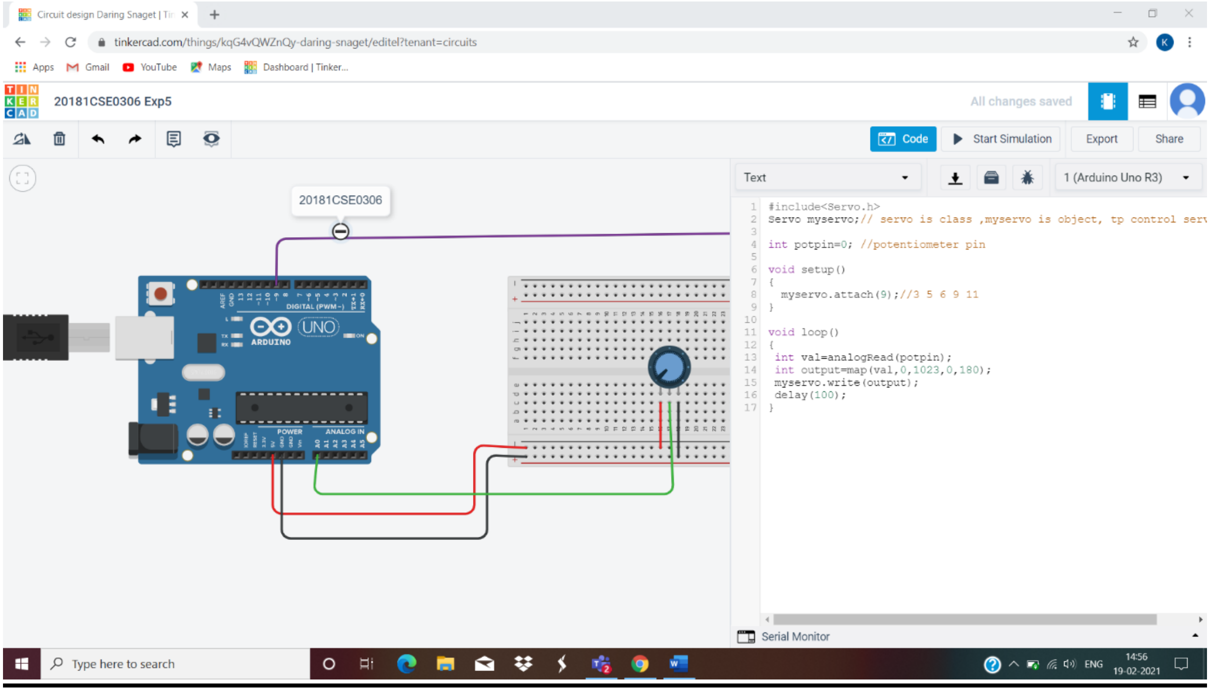
Experiment no:4 DATE:

Aim of the experiment: Arduino program to demonstrate control of servo motor using potentiometer.

Components required:

- Arduino Uno board
- Jumper wires
- Power cable
- Servo Motor
- Potentiometer(knob)

Initial circuit design:

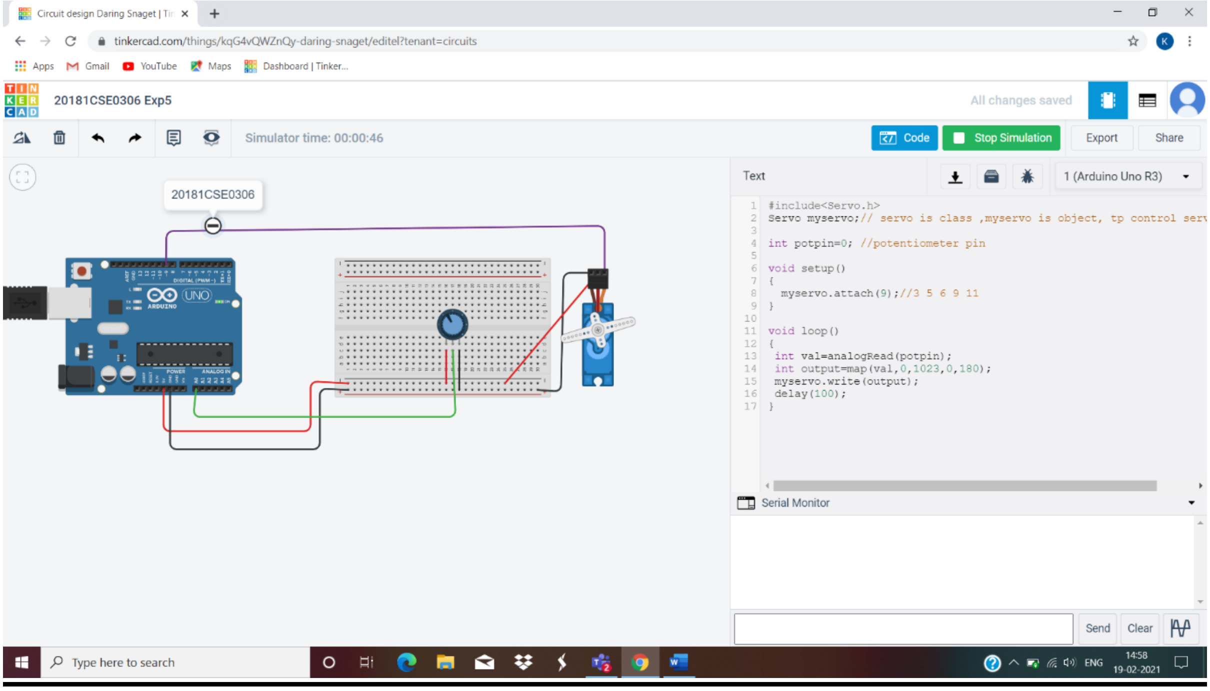


```
#include<Servo.h>
Servo myservo;// servo is class ,myservo is object, tp control serv

int potpin=0; //potentiometer pin

void setup()
{
  myservo.attach(9);//3 5 6 9 11
}

void loop()
{
 int val=analogRead(potpin);
 int output=map(val,0,1023,0,180);
 myservo.write(output);
 delay(100);
}
```

Arduino sketch:

```
#include<Servo.h>

Servo myservo;// servo is class ,myservo is object, tp control servometer


int potpin=0; //potentiometer pin


void setup()

{

  myservo.attach(9);//3 5 6 9 11

}


void loop()

{

 int val=analogRead(potpin);

 int output=map(val,0,1023,0,180);

 myservo.write(output);

 delay(100);

}
```

Output Screenshot:



```
#include<Servo.h>
Servo myservo;// servo is class ,myservo is object, tp control serv

int potpin=0; //potentiometer pin

void setup()
{
  myservo.attach(9);//3 5 6 9 11
}

void loop()
{
 int val=analogRead(potpin);
 int output=map(val,0,1023,0,180);
 myservo.write(output);
 delay(100);
}
```

<u>**Outcome:**</u> Control of servo motor using potentiometer was demonstrated successfully.

Additional Program

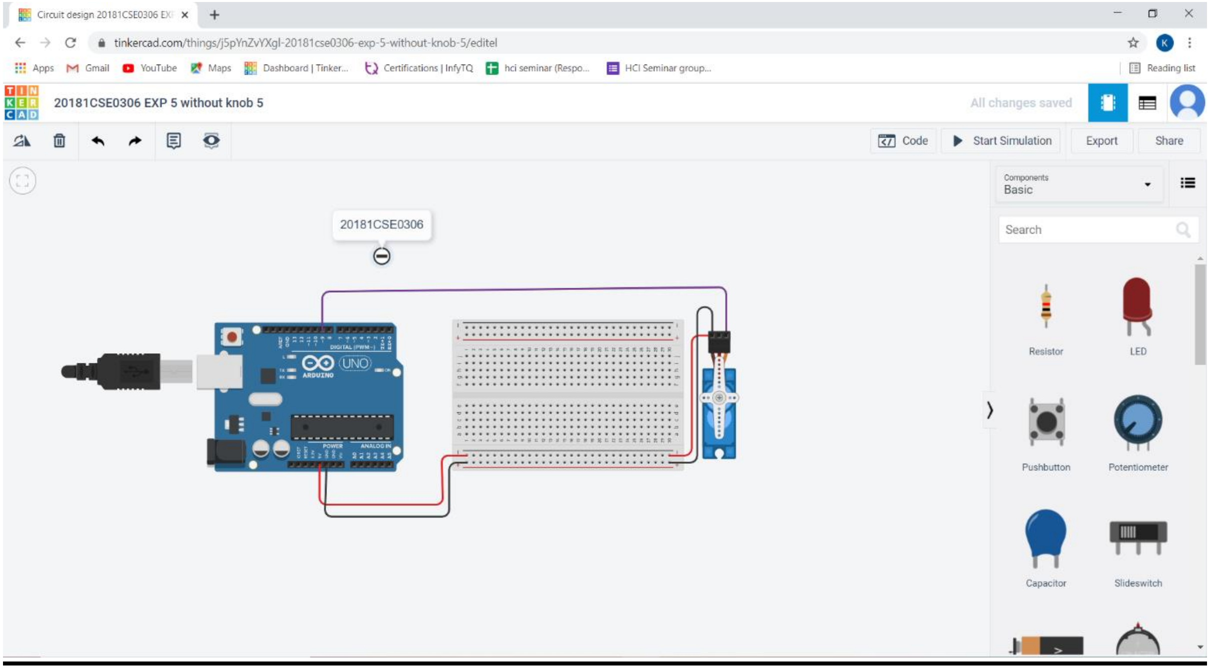
DATE:

Aim of the experiment: Arduino program to demonstrate control of servo motor **without using potentiometer**.

Components required:

- Arduino Uno board
- Jumper wires
- Power cable
- Servo Motor

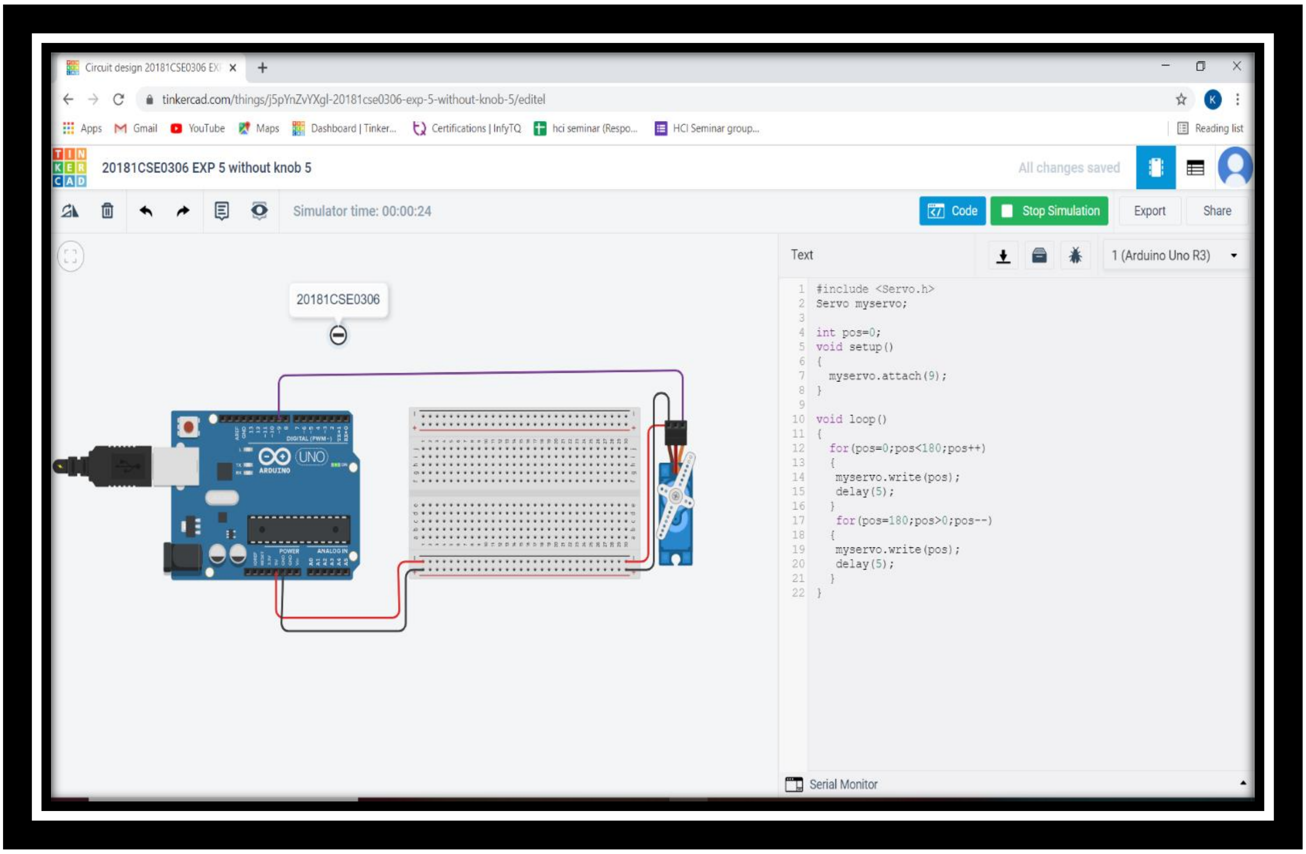
Initial circuit design:



Arduino sketch:

```
#include <Servo.h>
Servo myservo;

int pos=0;
void setup()
{
  myservo.attach(9);
}

void loop()
{
  for(pos=0;pos<180;pos++)
  {
   myservo.write(pos);
   delay(5);
  }
   for(pos=180;pos>0;pos--)
  {
   myservo.write(pos);
   delay(5);
  }
}
```

Output Screenshot:



```
#include <Servo.h>
Servo myservo;

int pos=0;
void setup()
{
  myservo.attach(9);
}

void loop()
{
  for(pos=0;pos<180;pos++)
  {
   myservo.write(pos);
   delay(5);
  }
   for(pos=180;pos>0;pos--)
  {
   myservo.write(pos);
   delay(5);
  }
}
```

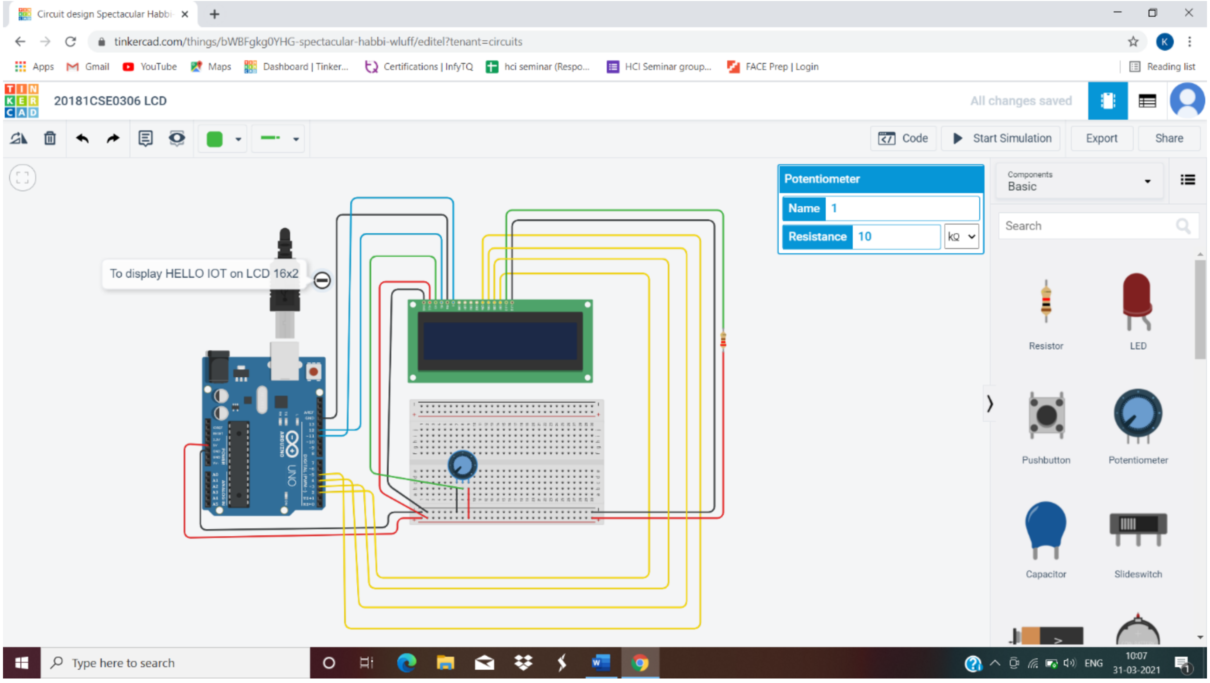
Outcome: Control of servo motor without using potentiometer was demonstrated successfully.

Aim of the experiment: Interfacing of a display device, i.e., LCD 16 x 2 with Arduino Uno. Write a program to display "HELLO IOT" on LCD.

Components required:

- Arduino Uno board
- LCD Screen
- 10k**Ω** potentiometer
- 220**Ω** resistor
- Small breadboard
- Jumper Wires

Initial circuit design:



To display HELLO IOT on LCD 16x2

Potentiometer

Name 1

Resistance 10 kΩ

LCD RS pin to digital pin 12

LCD enablle pin to digital pin 11

LCD D4 pin to digital pin 5

LCD D5 pin to digital pin 4

LCD D6 pin to digital pin 3

LCD D7 pin to digital pin 2

LCD R/W pin to GND

LCD VSS pin to GND

LCD VCC pin to 5V

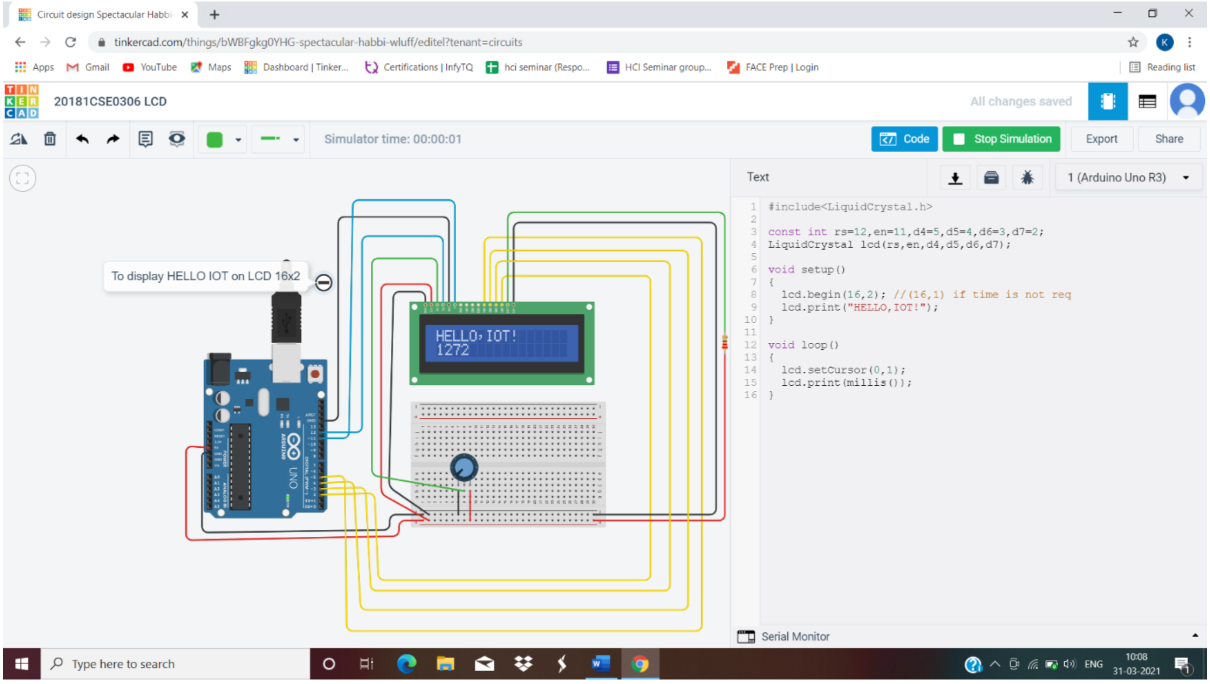
LCD LED+ to 5V through a 220 resistor

LCD LED- to GND

Arduino sketch:

```
#include<LiquidCrystal.h>
const int rs=12,en=11,d4=5,d5=4,d6=3,d7=2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

void setup()
{
  lcd.begin(16,2); //(16,1) if time is not req
  lcd.print("HELLO,IOT!");
}

void loop()
{
  lcd.setCursor(0,1);
  lcd.print(millis());
}
```

Output Screenshot:


Circuit design Spectacular Habbi
tinkercad.com/things/bWBFgkg0YHG-spectacular-habbi-wluff/editel?tenant=circuits
20181CSE0306 LCD
All changes saved
Simulator time: 00:00:01
Code
Stop Simulation
Export
Share
To display HELLO IOT on LCD 16x2
Text
1 (Arduino Uno R3)
Serial Monitor

```
HELLO, IOT!
1272
```

```
#include<LiquidCrystal.h>

const int rs=12,en=11,d4=5,d5=4,d6=3,d7=2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

void setup()
{
  lcd.begin(16,2); //(16,1) if time is not req
  lcd.print("HELLO,IOT!");
}

void loop()
{
  lcd.setCursor(0,1);
  lcd.print(millis());
}
```

Outcome: LCD was interfaced with Arduino Uno and the message “HELLO IOT” was successfully displayed on LCD.