**Name: Mrudula Kulkarni,R.N-31 , Batch:B2**

**PRACTICAL NO. 5 (Group B)**

**Problem Statement:** Write a program using Arduino to control LED (One or more ON/OFF) Or Blinking

**Program:**

void setup()

{

pinMode(13, OUTPUT);

}

void loop()

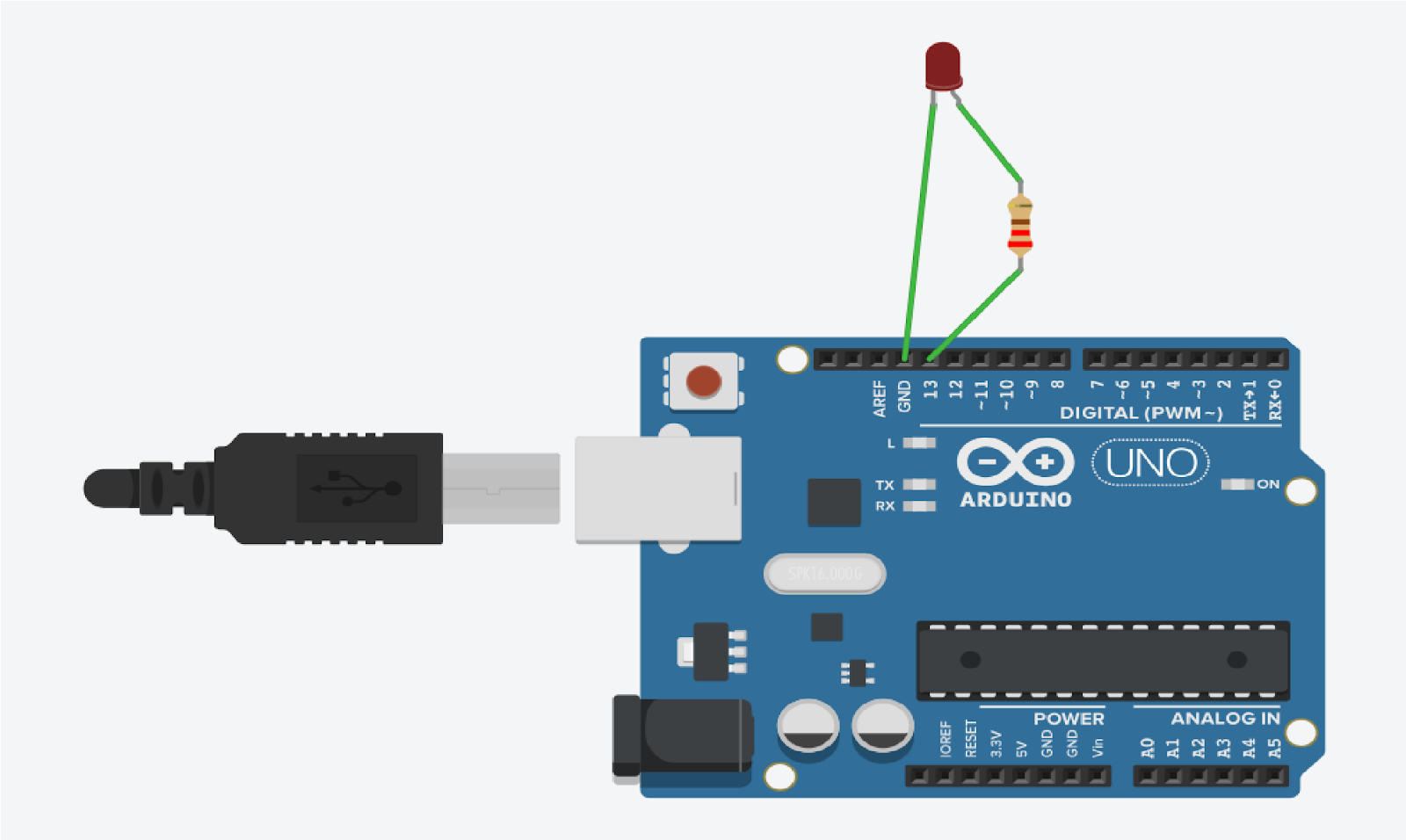
{

digitalWrite(13, HIGH);

delay(1000); digitalWrite(13, LOW); delay(1000);

}

**OUTPUT**



**PRACTICAL NO. 6 (Group B)**

**Problem Statement:** Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200

**PROGRAM :**

int red = 10;

int yellow = 9;

int green = 8;

int counter=0;

void setup() {

pinMode(red, OUTPUT);

pinMode(yellow, OUTPUT);

pinMode(green, OUTPUT);

}

void loop()

{

changeLights();

delay(100);

}

void changeLights(){

if(counter<100)

{

digitalWrite(yellow, HIGH);

digitalWrite(red, LOW);

digitalWrite(green, LOW);

delay(50);

}else if(counter>100 && counter<200)

{

digitalWrite(red, HIGH);

digitalWrite(yellow, LOW);

digitalWrite(green, LOW);

delay(50);

}

else if (counter>200 && counter<500)

{

digitalWrite(green, HIGH);

digitalWrite(red, LOW);

digitalWrite(yellow, LOW);

delay(50);

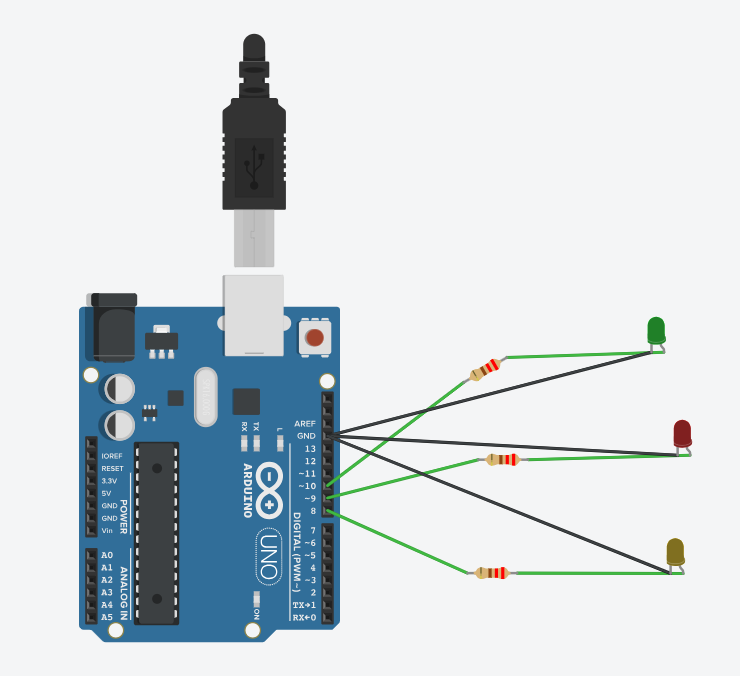
counter=0;

}

counter=counter+50;

}

**OUTPUT**



**PRACTICAL NO. 7 (Group B)**

**Problem Statement:** Create a program so that when the user enters "b" the blue light blinks, "g" the

green light is illuminated, and "r" the red light is illuminated

**PROGRAM :**

int mychar = 0; // for incoming serial data

void setup()

{

Serial.begin(9600); // opens serial port, sets data rate to 9600 bps

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

pinMode(9,OUTPUT);

}

void loop()

{

// send data only when you receive data:

if (Serial.available() > 0)

{

// read the incoming byte:

mychar = Serial.read();

// say what you got:

Serial.print("I received: ");

Serial.println(mychar);

}

if(mychar == 114 ) //r

{

digitalWrite(7,HIGH);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

}

if(mychar == 103 ) //g

{

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

}

if(mychar == 98 ) //b

{

digitalWrite(7,LOW);

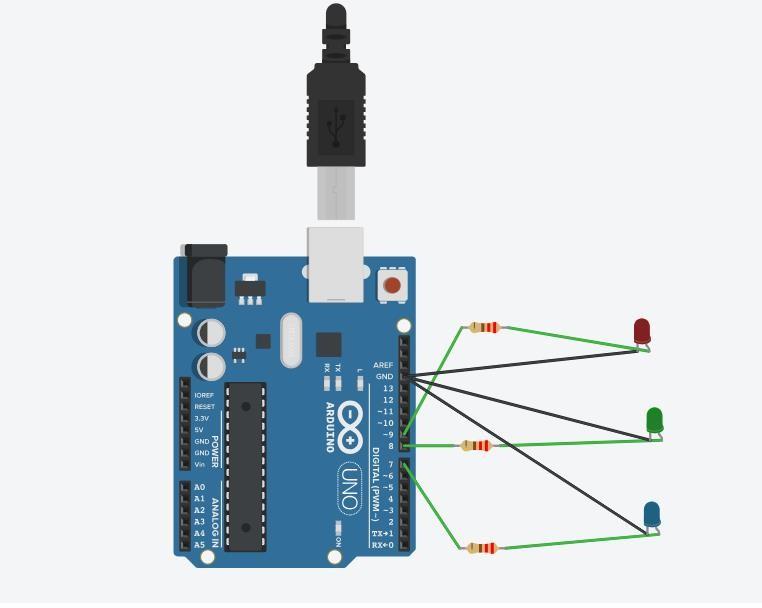
digitalWrite(8,LOW);

digitalWrite(9,HIGH);

}

}

**OUTPUT:**



**PRACTICALNO. 8 (Group B)**

**Problem Statement:** Write a program that asks the user for a number and outputs the number squared that is entered

**Program Code:**

int out;

void setup()

{

Serial.begin(9600); // opens serial port, sets data rate to 9600 bps

}

void loop()

{

// send data only when you receive data:

if (Serial.available() > 0)

{

// read the incoming byte:

int num=Serial.readString().toInt();

// say what you got

Serial.print("I received: ");

Serial.println(num);

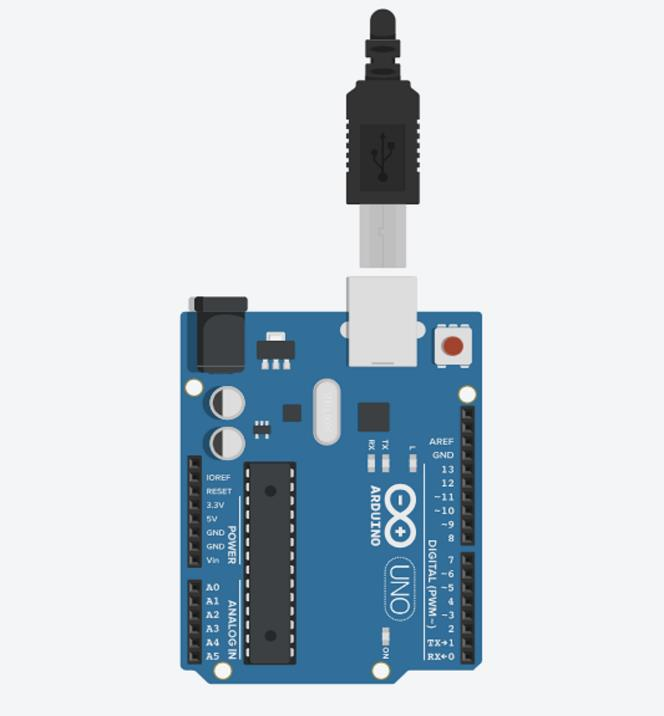
out = num\*num;

Serial.print("Sq of no.: ");

Serial.println(out);

}

**OUTPUT:**



**PRACTICAL NO. 9 (Group B)**

**Problem Statement:** Write a program read the temperature sensor and send the values to the serial monitor on the computer.

**PROGRAM CODE :**

#define DHTPIN 2

void setup()

{

Serial.begin(9600);

}

void loop()

{

int temperature = analogRead(DHTPIN);

Serial.print("Temperature is: ");

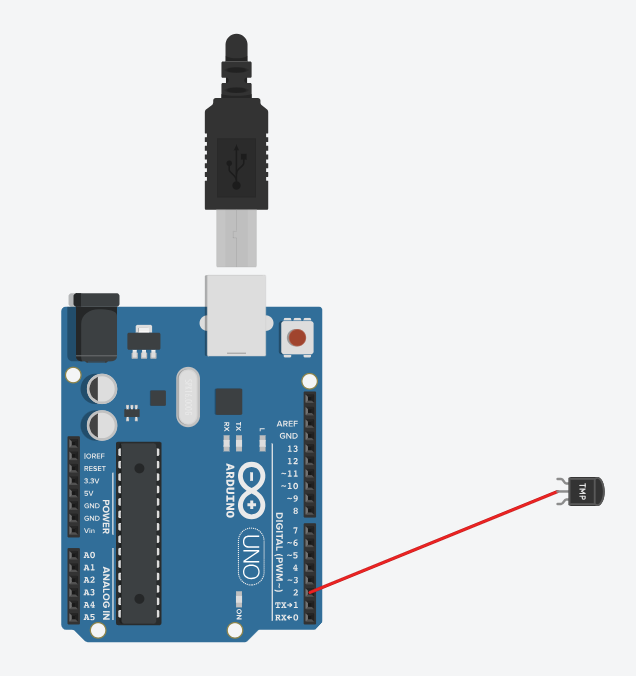
Serial.print(temperature);

Serial.println(" degree Fahrenheit ");

delay(2000);

}

**OUTPUT:**



**PRACTICAL NO. 10 (Group B)**

**Problem Statement:** Write a program so it displays the temperature in Fahrenheit as well as the

maximum and minimum temperatures it has seen.

**PROGRAM CODE :**

int val;

int tempPin = 1;

void setup()

{

Serial.begin(9600);

}

void loop()

{

val = analogRead(tempPin);

float mv = ( val/1024.0)\*5000;

float cel = mv/10;

float farh = (cel\*9)/5 + 32;

Serial.print("TEMPRATURE = ");

Serial.print(cel);

Serial.print("\*C");

Serial.println();

delay(1000);

float tfmax = 100;

float tfmin = 0;

if (farh > tfmax)

{

tfmax = farh;

}

if (farh < tfmin)

{

tfmin = farh;

}

Serial.print("TEMPRATURE = ");

Serial.print(farh);

Serial.print("\*F");

Serial.println();

Serial.print("Max Temp");

Serial.print(tfmax);

Serial.println();

Serial.print("Min Temp");

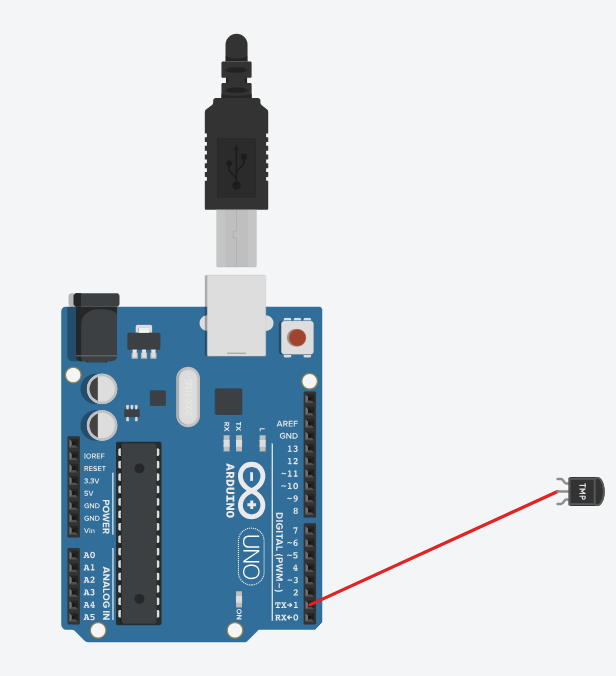
Serial.print(tfmin);

Serial.println();

Serial.println();

}

**OUTPUT:**



**PRACTICALNO. 11 (Group B)**

**Problem Statement:** Write a program to show the temperature and shows a graph of the recent

Measurements

**PROGRAM:**

int inPin = A0;

float tempC = 0;

int n = 0;

void setup()

{

Serial.begin(9600);

}

void loop()

{

int value = analogRead(inPin); // read the value from the sensor

tempC = ((value \* 0.00488) - 0.5) \* 100;

//Serial.print("Temperature in Celsius = ");

Serial.println(tempC);

if(n==30)

{

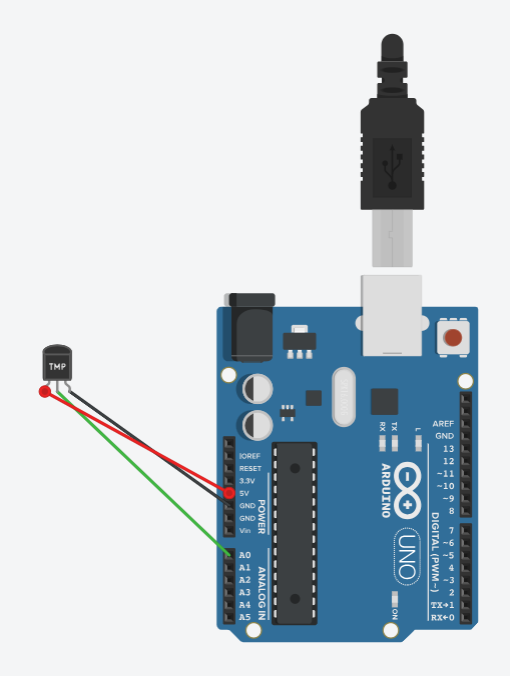
n=0;

delay(5000);

}

}

**OUTPUT:**



**PRACTICAL NO 12. (GROUP B)**

**Problem Statement:** Write a program using piezo element and use it to play a tune after someone knocks

**PROGRAM :**

int pin=8;

void setup()

{

}

void loop()

{

tone(8,200);

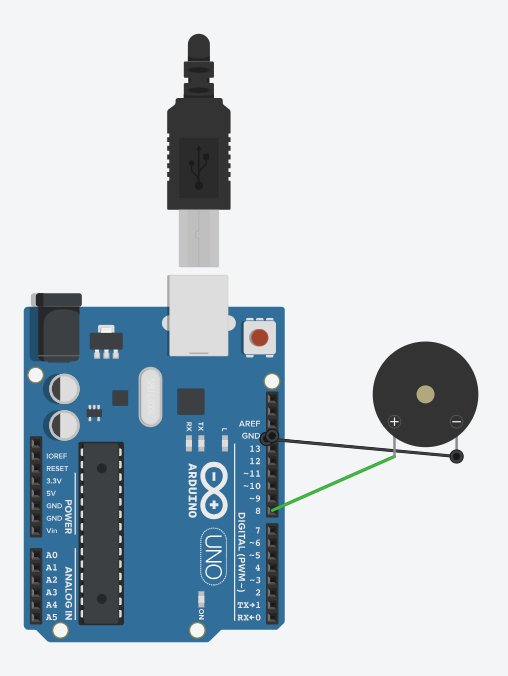
delay(2000);// Wait for 2000 millisecond(s)

noTone(8);

delay(2000);// Wait for 2000 millisecond(s)

}

**OUTPUT:**



**PRACTICALNO. 13 (Group C)**

**Problem Statement:** Write an application to control the operation of hardware simulated traffic signals

**PROGRAM**

byte R1 = 2;

byte Y1 = 3;

byte G1 = 4;

byte R2 = 5;

byte Y2 = 6;

byte G2 = 7;

byte R3 = 8;

byte Y3 = 9;

byte G3 = 10;

byte R4 = 11;

byte Y4 = 12;

byte G4 = 13;

int Lane\_A[] = {R1, Y1, G1};

// Lane 1 Red, Yellow and Green

int Lane\_B[] = {R2, Y2, G2};

// Lane 2 Red, Yellow and Green

int Lane\_C[] = {R3, Y3, G3};

// Lane 3 Red, Yellow and Green

int Lane\_D[] = {R4, Y4, G4};

// Lane 4 Red, Yellow and Green

void setup()

{

for (int i = 0; i < 3; i++)

{

pinMode(Lane\_A[i], OUTPUT);

pinMode(Lane\_B[i], OUTPUT);

pinMode(Lane\_C[i], OUTPUT);

pinMode(Lane\_D[i], OUTPUT);

}

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane\_A[i], LOW);

digitalWrite(Lane\_B[i], LOW);

digitalWrite(Lane\_C[i], LOW);

digitalWrite(Lane\_D[i], LOW);

}

}

void loop()

{

digitalWrite(Lane\_A[2], HIGH); //LaneA Green ON

digitalWrite(Lane\_A[0], LOW); //LaneA Red OFF

digitalWrite(Lane\_C[0], HIGH);

//LaneA Red OFF

digitalWrite(Lane\_D[0], HIGH); //LaneA Red OFF

digitalWrite(Lane\_B[0], HIGH);

//LaneA Red OFF

delay(7000);

digitalWrite(Lane\_A[2], LOW); //LaneA Green OFF

digitalWrite(Lane\_A[1], HIGH); //LaneA Yellow ON

delay(3000);

digitalWrite(Lane\_A[0], HIGH); //LaneA Red ON

digitalWrite(Lane\_A[1], LOW); //LaneA Yellow OFF

digitalWrite(Lane\_B[0], LOW);

//LaneB Red OFF

digitalWrite(Lane\_B[2], HIGH);

//LaneB Green ON

delay(7000);

digitalWrite(Lane\_B[2], LOW); //LaneB Green OFF

digitalWrite(Lane\_B[1], HIGH); //LaneB Yellow ON

delay(3000);

digitalWrite(Lane\_B[0], HIGH); //LaneB Red ON

digitalWrite(Lane\_B[1], LOW); //LaneB Yellow OFF

digitalWrite(Lane\_C[0], LOW); //LaneC Red OFF

digitalWrite(Lane\_C[2], HIGH); //LaneC Green ON

delay(7000);

digitalWrite(Lane\_C[2], LOW); //LaneC Green OFF

digitalWrite(Lane\_C[1], HIGH); //LaneC Yellow ON

delay(3000);

digitalWrite(Lane\_C[0], HIGH); //LaneC Red ON

digitalWrite(Lane\_C[1], LOW);

//LaneC Yellow OFF

digitalWrite(Lane\_D[0], LOW);

//LaneD Red OFF

digitalWrite(Lane\_D[2], HIGH); //LaneD Green ON

delay(7000);

digitalWrite(Lane\_D[2], LOW); //LaneD Green OFF

digitalWrite(Lane\_D[1], HIGH); //LaneD Yellow ON

delay(3000);

digitalWrite(Lane\_D[1], LOW); //LaneD Yellow OFF

}

**OUTPUT:**

