IOT PRACTICAL

Practical 1

Q]Blinking LED-Using Tinkercad

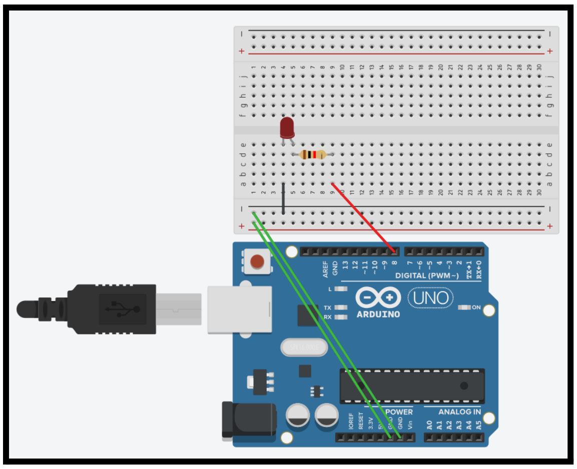
Aim: Displaying blinking LED using Tinkercad

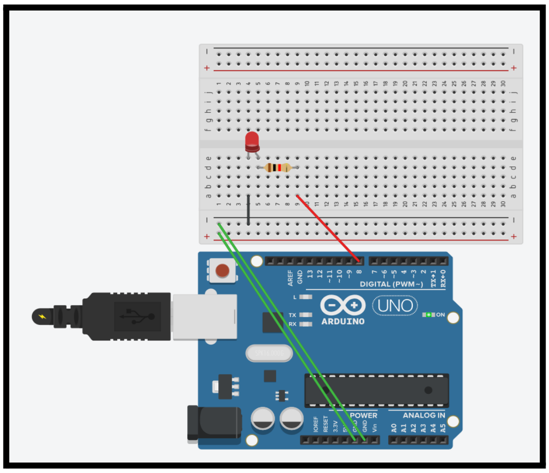
Code:-

```
int ledpin=8;
void setup()
{
  pinMode(ledpin, OUTPUT);
}

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

Output:-





Practical 2]

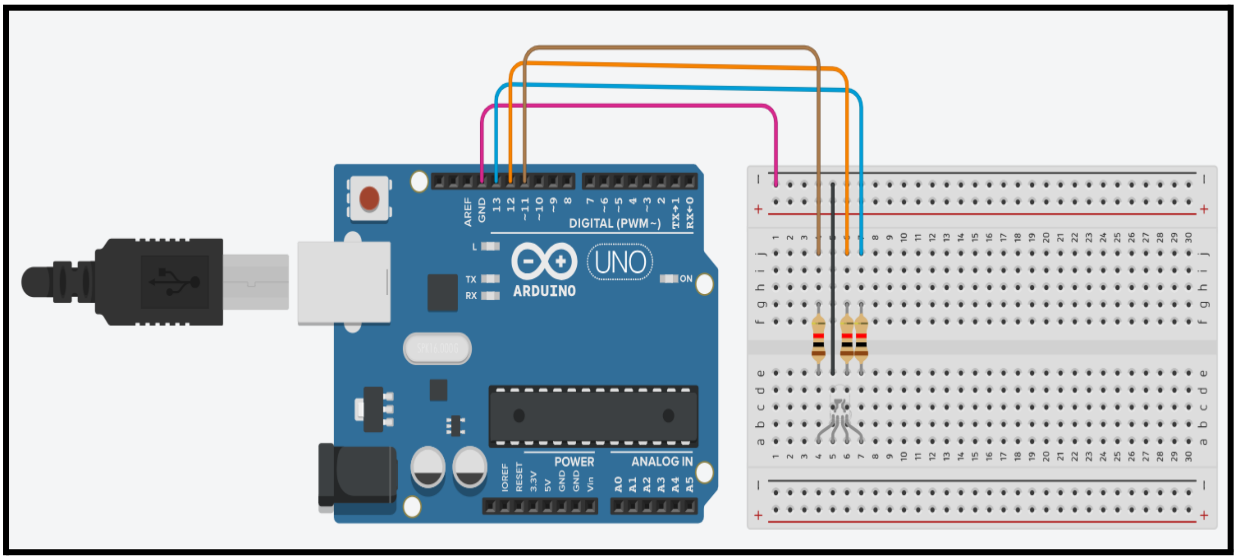
Q]RGB Pattern

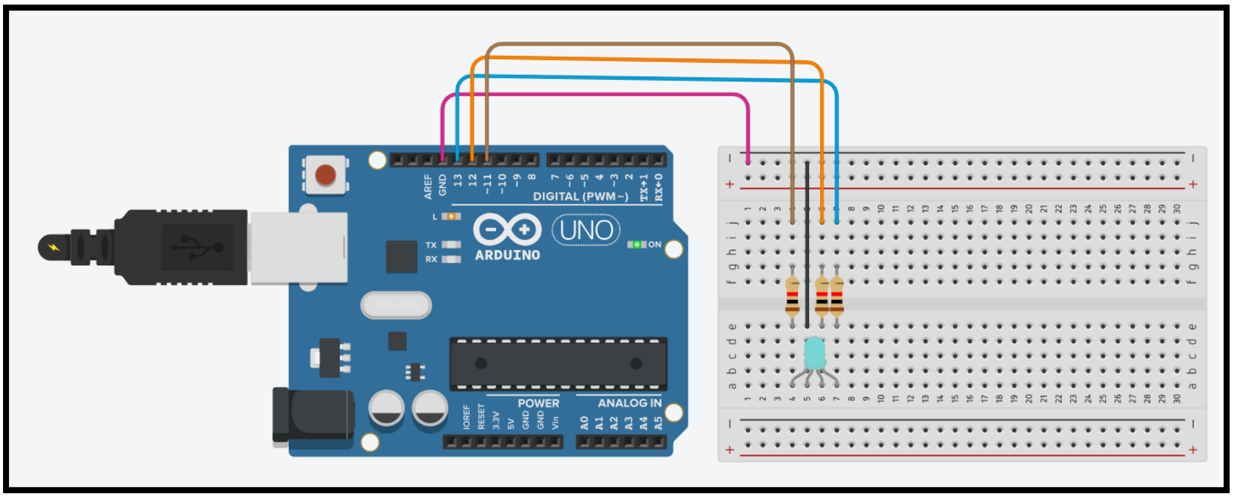
Aim: Displaying RGB Pattern

Code:-

```
#define LEDR 11
#define LEDG 13
#define LEDB 12
void setup()
  {
      pinMode(LEDR,OUTPUT);
      pinMode(LEDG,OUTPUT);
      pinMode(LEDB,OUTPUT);
  }
int r=0;
int g=0;
int b=0;
void loop()
  {
       r=random(0,255);
       g=random(0,255);
       b=random(0,255);
      analogWrite(LEDR,r);
      analogWrite(LEDG,g);
      analogWrite(LEDB,b);
      delay(1000);
 }
```

Output:-





Practical 3

Q]7-Segment LED Display using Tinkercad

Aim: Displaying 7-Segment LED Display using Tinkercad

Code:-

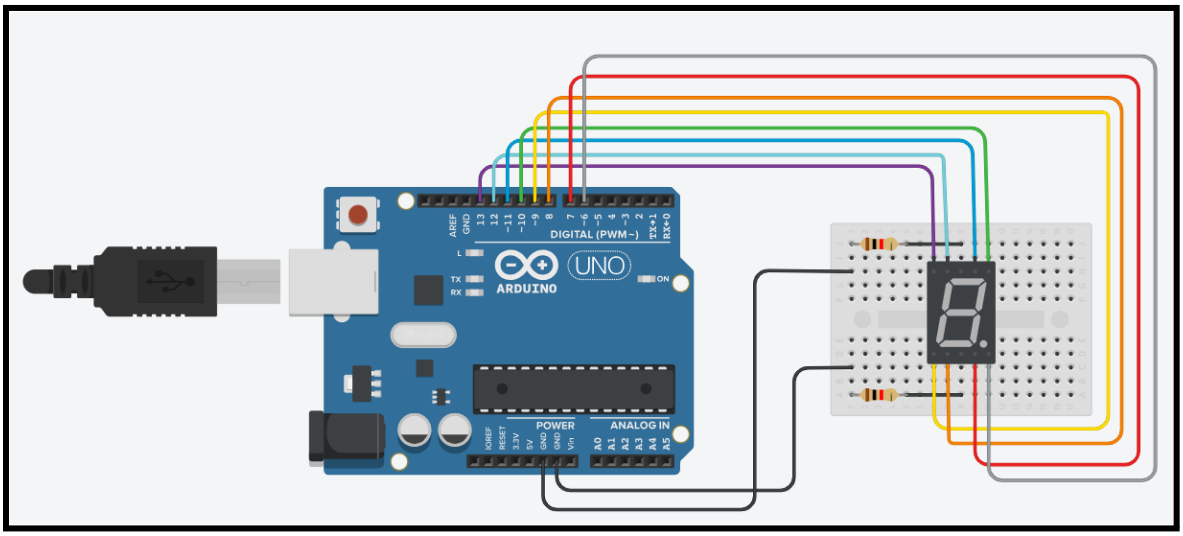
```
unsigned const int A = 13;
unsigned const int B = 12;
unsigned const int C = 11;
unsigned const int D = 10;
unsigned const int E = 9;
unsigned const int F = 8;
unsigned const int G = 7;
unsigned const int H = 6;
void setup(void)
{
        pinMode(A, OUTPUT);
        pinMode(B, OUTPUT);
        pinMode(C, OUTPUT);
        pinMode(D, OUTPUT);
        pinMode(E, OUTPUT);
        pinMode(F, OUTPUT);
        pinMode(G, OUTPUT);
       pinMode(H, OUTPUT);
}
void zero(void)
{
       digitalWrite(A, LOW);
       digitalWrite(B, HIGH);
       digitalWrite(C, HIGH);
```

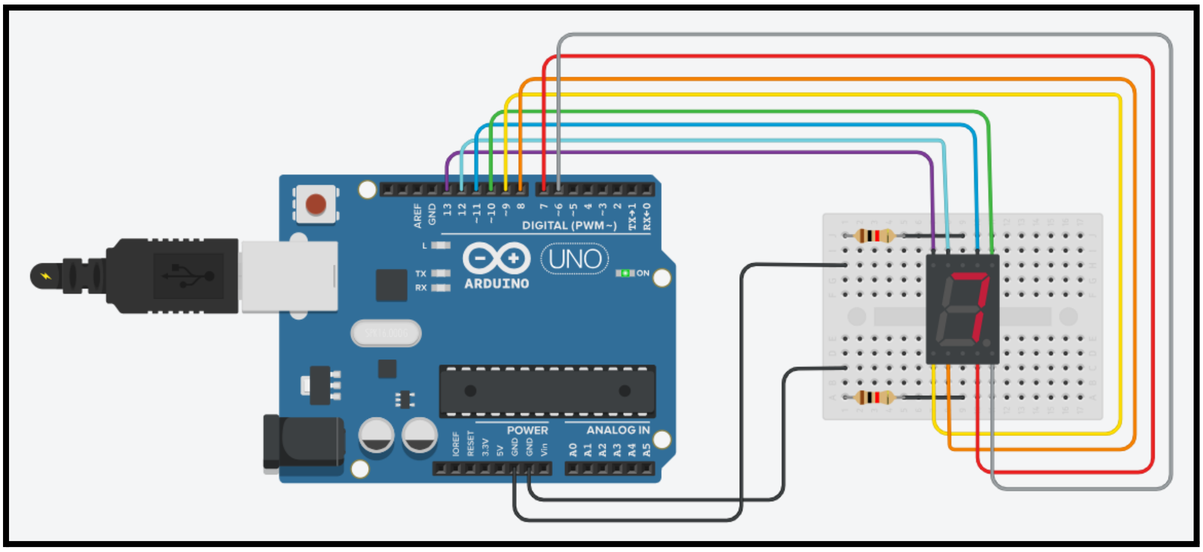
```
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
}
void one(void)
{
        digitalWrite(A, LOW);
        digitalWrite(A, LOW);
        digitalWrite(A, LOW);
        digitalWrite(A, HIGH);
        digitalWrite(A, LOW);
        digitalWrite(A, LOW);
        digitalWrite(A, HIGH);
        digitalWrite(A, LOW);
}
void two(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(B, LOW);
        digitalWrite(C, HIGH);
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, LOW);
        digitalWrite(H, LOW);
}
void three(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(B, LOW);
        digitalWrite(C, HIGH);
        digitalWrite(D, HIGH);
        digitalWrite(E, LOW);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
         digitalWrite(H, LOW);
}
void four(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(A, HIGH);
        digitalWrite(A, LOW);
        digitalWrite(A, HIGH);
        digitalWrite(A, LOW);
        digitalWrite(A, LOW);
        digitalWrite(A, HIGH);
        digitalWrite(A, LOW);
}
void five(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(B, HIGH);
        digitalWrite(C, HIGH);
        digitalWrite(D, LOW);
        digitalWrite(E, LOW);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
}
void six(void)
```

```
{
        digitalWrite(A, HIGH);
        digitalWrite(B, HIGH);
        digitalWrite(C, HIGH);
        digitalWrite(D, LOW);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
}
void seven(void)
{
        digitalWrite(A, LOW);
        digitalWrite(B, LOW);
        digitalWrite(C, HIGH);
        digitalWrite(D, HIGH);
        digitalWrite(E, LOW);
        digitalWrite(F, LOW);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
}
void eight(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(B, HIGH);
        digitalWrite(C, HIGH);
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
```

```
}
void nine(void)
{
        digitalWrite(A, HIGH);
        digitalWrite(B, HIGH);
        digitalWrite(C, HIGH);
        digitalWrite(D, HIGH);
        digitalWrite(E, LOW);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        digitalWrite(H, LOW);
}
void loop(void)
{
  zero();
  delay(1000);
  one();
  delay(1000);
  two();
  delay(1000);
  three();
  delay(1000);
  four();
  delay(1000);
  five();
  delay(1000);
  six();
  delay(1000);
  seven();
  delay(1000);
  eight();
  delay(1000);
  nine();
  delay(1000);
}
```

Output:-





Practical 4

Q]Oscilloscope

Aim: Implementing Oscilloscope

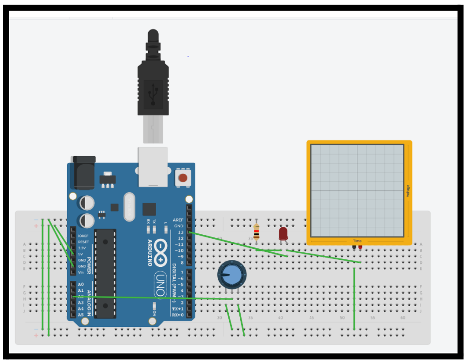
Code:-

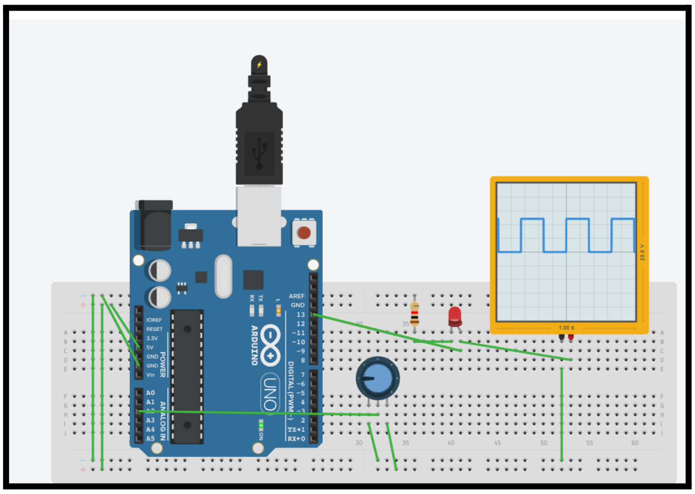
```
int potPin = 2;
int ledPin = 13;
int val = 10;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  val = analogRead(potPin);
  digitalWrite(ledPin, HIGH);
  delay(val);
  digitalWrite(ledPin, LOW);
  delay(val);
}
```

Output:-





Practical 5

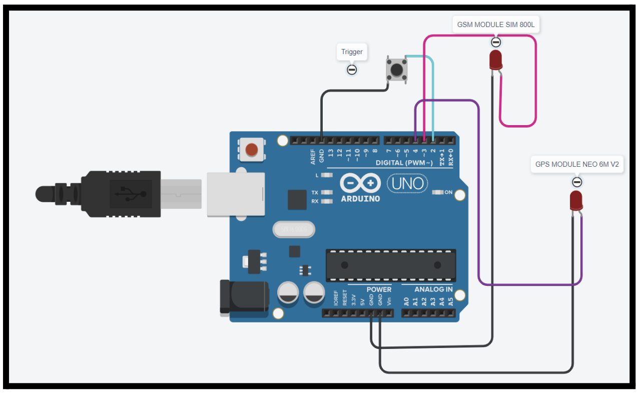
Q]GPS Tracker

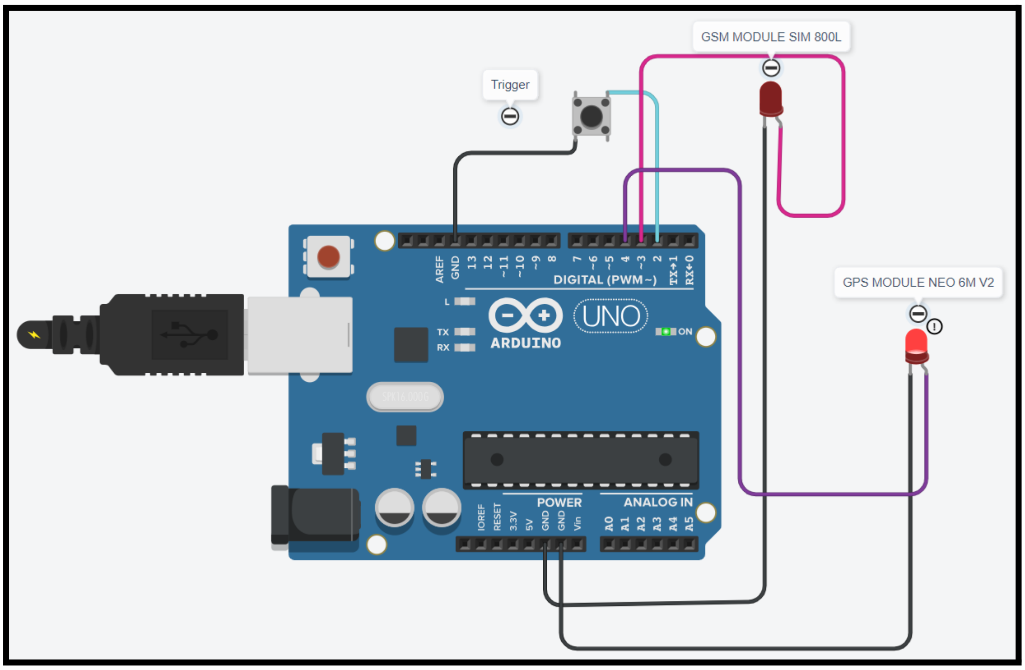
Aim: Displaying GPS track using LED

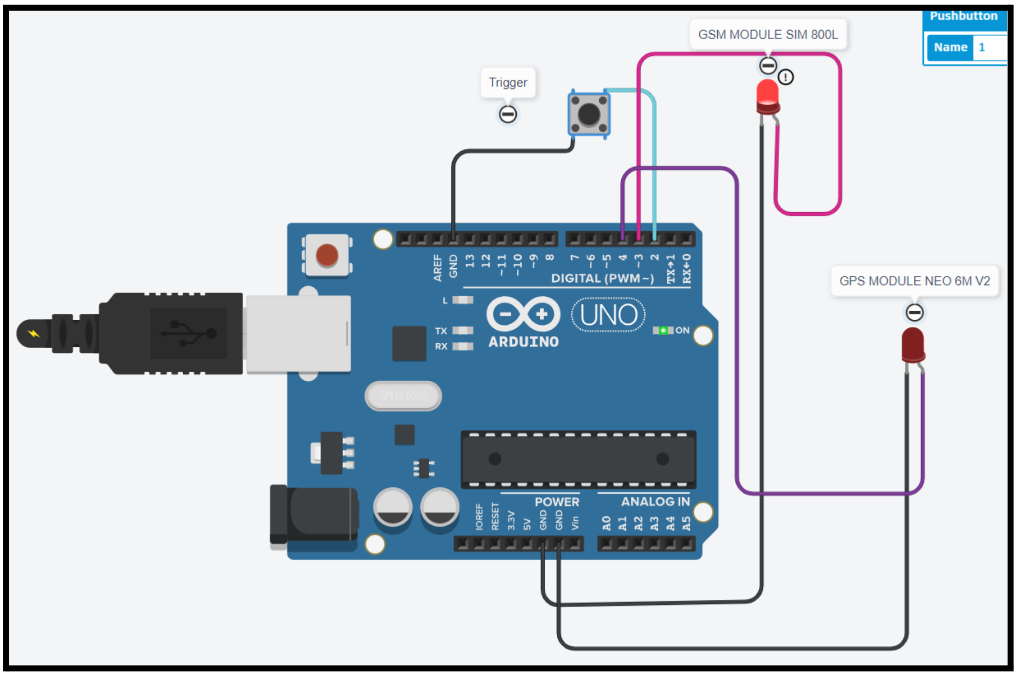
Code:-

```
int BTN=2;
int GSM=3;
int GPS=4;
void setup()
{
        Serial.begin(9600);
        pinMode(GSM,OUTPUT);
        pinMode(GPS,OUTPUT);
        pinMode(BTN,INPUT_PULLUP);
}
void loop()
{
   if (digitalRead(BTN)==LOW)
        {
                digitalWrite(GPS,HIGH);
                Serial.println("getting coordinate");
                delay(5000);
                digitalWrite(GPS,LOW);
                digitalWrite(GSM,HIGH);
                Serial.println("sending location coordinate");
                delay(1000);
                Serial.println("coordinate sent");
                digitalWrite(GSM,LOW);
        }
}
```

Output:-


GSM MODULE SIM 800L
Trigger
GPS MODULE NEO 6M V2
UNO
ARDUINO


GSM MODULE SIM 800L
Trigger
GPS MODULE NEO 6M V2
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN


Pushbutton
Name 1
GSM MODULE SIM 800L
Trigger
GPS MODULE NEO 6M V2
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Practical 6

Q]Home Automation

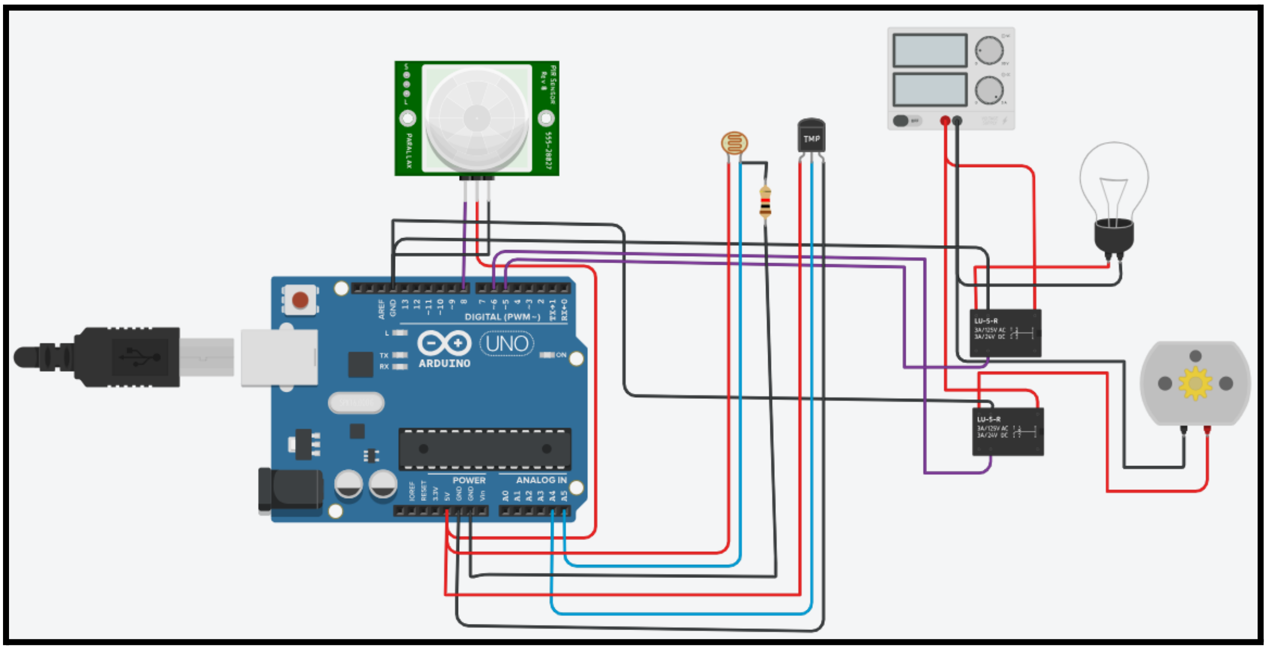
Aim: Implementing a Home Automation System

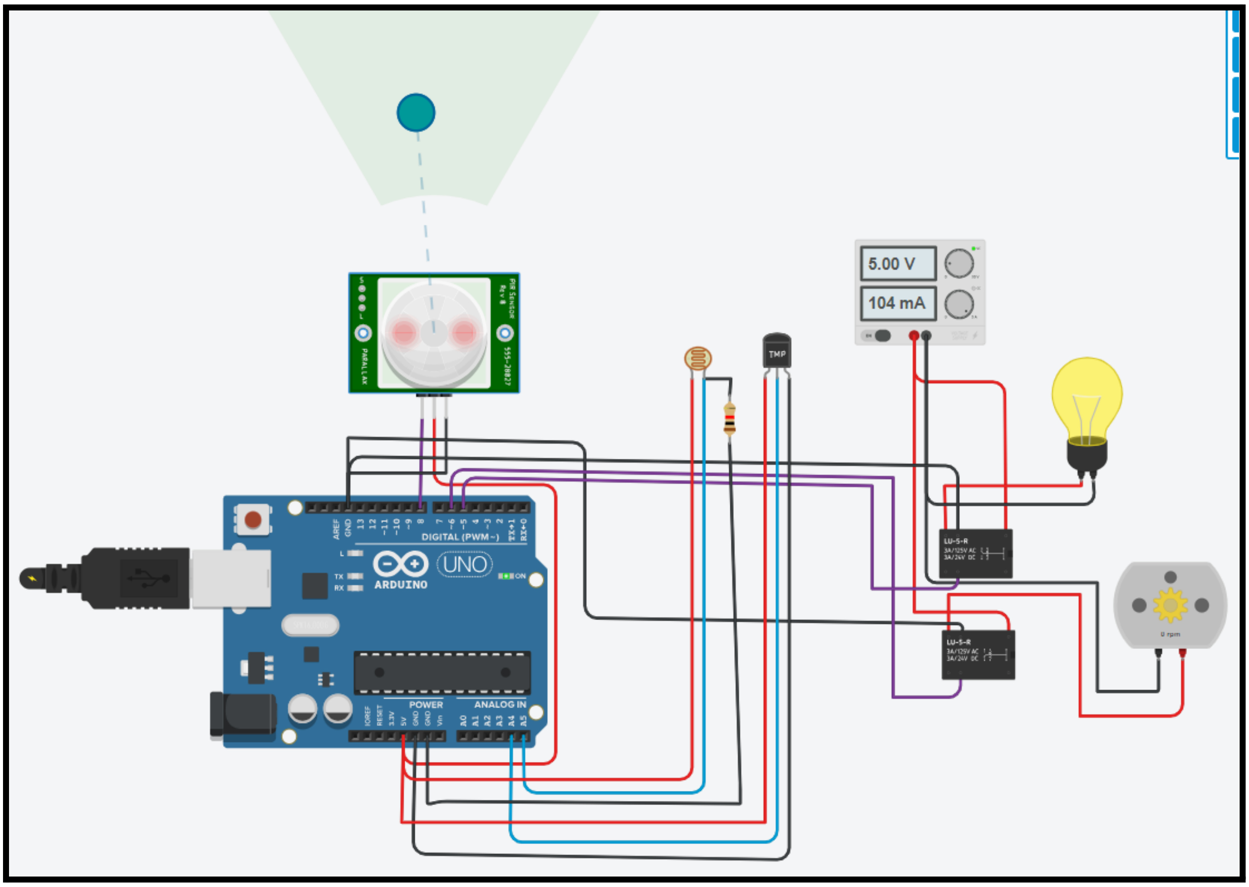
Code:-

```
float x,y,z,temp;
void setup()
{
	pinMode(8, INPUT);
	pinMode(5, OUTPUT);
	pinMode(6, OUTPUT);
	pinMode(A5, INPUT);
	pinMode(A4, INPUT);
	Serial.begin(9600);
}
void loop()
{
	x= digitalRead(8);
	y= analogRead(A5);
	z= analogRead(A4);
	Serial.println(x);
	Serial.println(y);
	Serial.println(z);
	temp = (double)z / 1024;
	temp = temp * 5;
	temp = temp - 0.5;
	temp = temp * 100;
	if ( (x>0) )
	   {
	       if ((y<550)&&(temp>30))
```

```
                    {
                        digitalWrite(5, HIGH);
                        digitalWrite(6, HIGH);
                    }
                else if((y<550)&&(temp<30))
                    {
                        digitalWrite(5, HIGH);
                        digitalWrite(6, LOW);
                    }
                else if((y>550)&&(temp>30))
                    {
                        digitalWrite(5, LOW);
                        digitalWrite(6, HIGH);
                    }
                else if((y>550)&&(temp<30))
                    {
                        digitalWrite(5, LOW);
                        digitalWrite(6, LOW);
                    }
            }
        else
            {
                digitalWrite(5, LOW);
                digitalWrite(6, LOW);
            }
}
```

Output:-


PIR Sensor
PARALLAX
555-28027
TMP
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
LU-5-R
LU-5-R


PIR Sensor
PARALLAX
555-28027
5.00 V
104 mA
TMP
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
LU-5-R
LU-5-R

Practical 7

Q}]Monitoring System

Aim: Monitoring System using LEDs

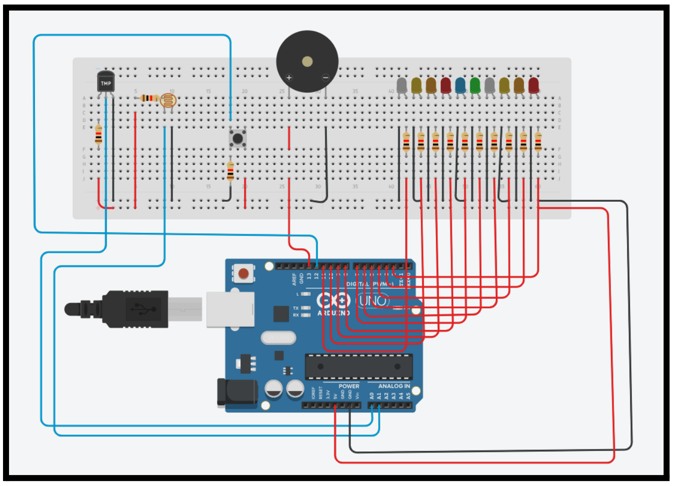
Code:-

```
#define LED1 2
#define LED10 11
#define LED_ON 100
#define TEMP_SENS_PIN A0
#define LIGHT_SENS_PIN A1
#define BUZZER_PIN 13
#define BUTTON_PIN 12
#define THRESHOLD 200
int temperature =0;
int lightness= 0;
int countLeds=0;
boolean previous=true;
boolean trigger=false;
void setup()
{
  for(int i=LED1;i<=LED10 ;i++)
  {pinMode(i,OUTPUT);}
   pinMode(BUZZER_PIN,OUTPUT);
    pinMode(BUTTON_PIN , INPUT);
     Serial.begin (9600);
 Serial.print("t for temperature, l for lightness");
  }
void loop()
{
  temperature = analogRead(TEMP_SENS_PIN );
```

```
lightness = analogRead(LIGHT_SENS_PIN);
 if (Serial.available()>0)
{
  char mode = Serial.read();
  if(mode =='t' || mode== 'T')
  {
    Serial.print("Temperature :");
    Serial.print("\t");
    Serial.println(temperature);
  }
  if(mode =='l' || mode=='L')
  {
     Serial.print("Lightness :");
     Serial.print("\t");
     Serial.println(lightness);
  }
}
  boolean current=digitalRead(BUTTON_PIN);
if(current && previous)
{
  trigger=!trigger;
}
previous=current;
if(trigger)
{
  countLeds=temperature;
}
else
{
  countLeds=lightness;
}
```

```
  int lastLed=map(countLeds ,0 ,500,2,11);
  for(int i=LED1;i<=lastLed;i++)
  {
    digitalWrite(i,HIGH);
    delay(LED_ON);
    digitalWrite(i,LOW);
  }
  if(temperature>THRESHOLD)
  {
    tone(BUZZER_PIN,1000,1000);
  }
}
```

Output:-



Practical 8

Q]RFID

Aim: Interfacing RFID using Tinkercad

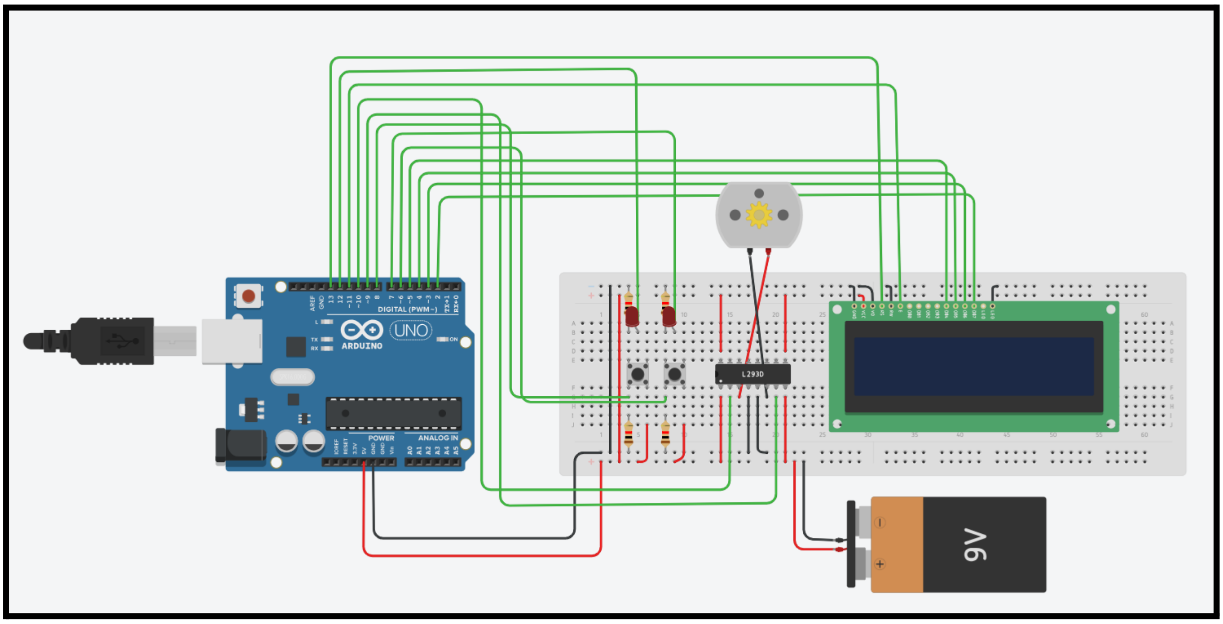
Code:-

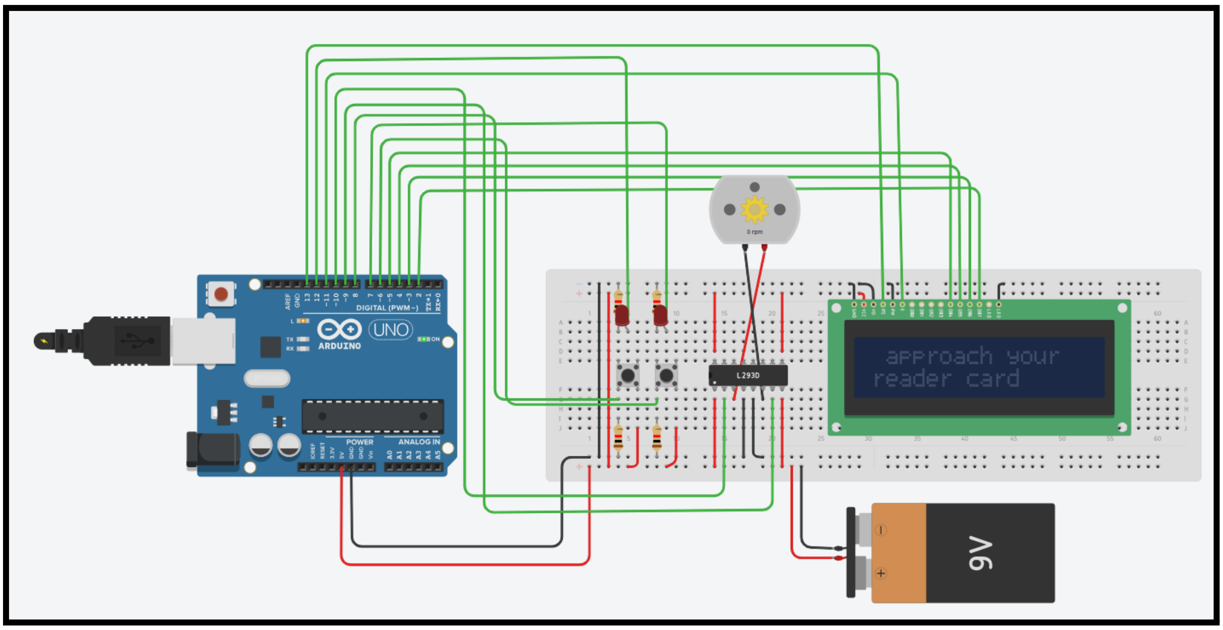
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 11, 5, 4, 3, 2);

int m1 = 9;
int redLed = 12;
int greenLed = 7;
int m2 = 10;
int sucessButton = 8;
int failedButton = 6;

void setup()
{
  lcd.begin(16, 2);
  pinMode(sucessButton,INPUT);
  pinMode(failedButton,INPUT);
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(redLed,OUTPUT);
  pinMode(greenLed,OUTPUT);
  initialMessage();
}
void loop()
{
  if(digitalRead(sucessButton) == 1)
  {
    lcd.clear();
    lcd.print("validated card");
    lcd.setCursor(0,1);
    lcd.print("successfully");
    digitalWrite(m1,HIGH);
    digitalWrite(greenLed,HIGH);
    delay(1000);
    initialMessage();
    digitalWrite(m1,LOW);
    digitalWrite(greenLed,LOW);
  }
  else if(digitalRead(failedButton) == 1)
  {
    lcd.clear();
    digitalWrite(redLed,HIGH);
    lcd.print("validated card");
    delay(1000);
    initialMessage();
    digitalWrite(redLed,LOW);
  }
}
void initialMessage()
{
  lcd.clear();
  lcd.print(" approach your");
  lcd.setCursor(0,1);
  lcd.print("reader card");
}
```

Output:-





```
approach your
reader card
```

Practical 9

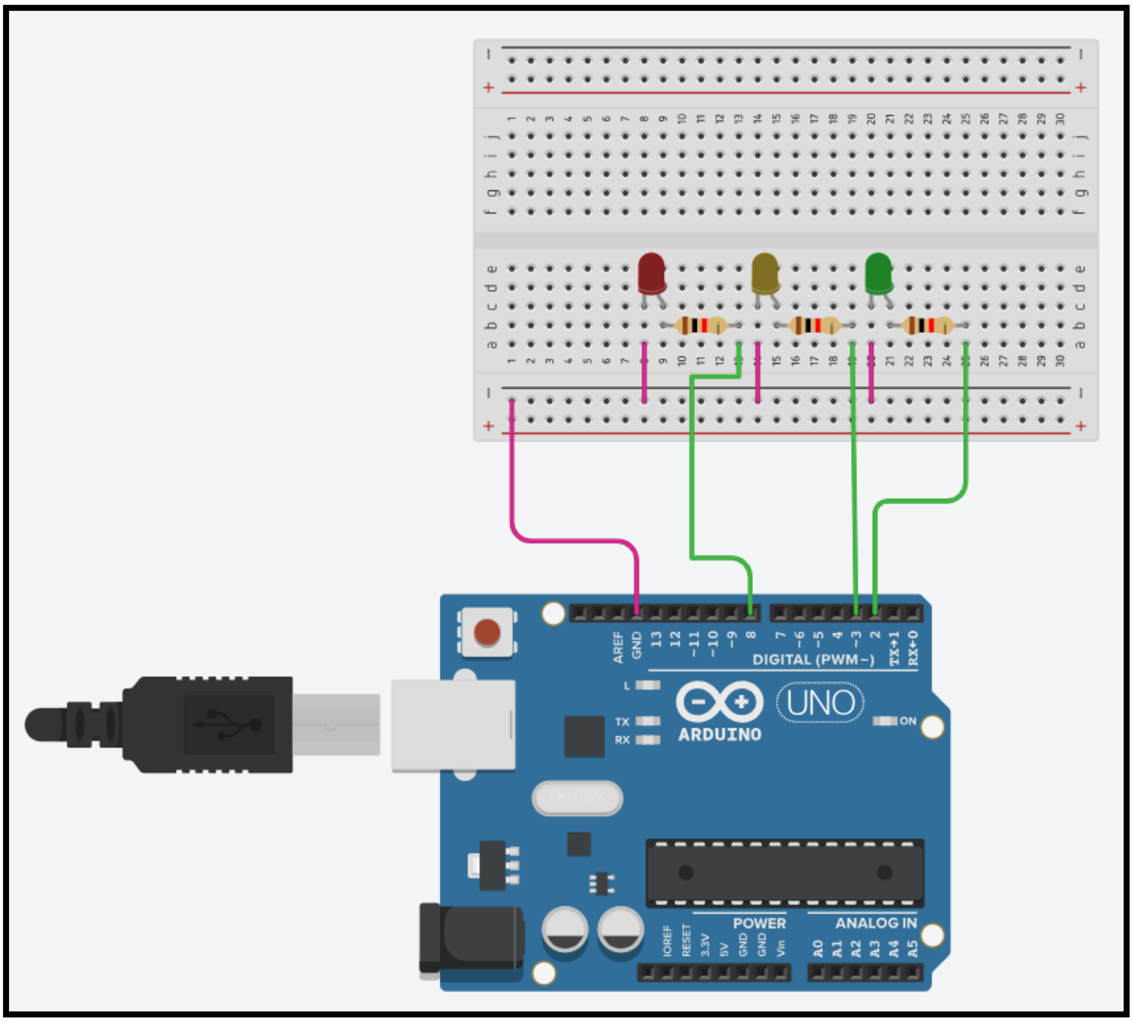
Q]Traffic Signal

Aim: Displaying Traffic Signal using LEDs

Code:-

```
void setup()
{
  pinMode(8, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(2, OUTPUT);
}
void loop()
{
  digitalWrite(8, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(2, LOW);
  delay(5000); // Wait for 5000 millisecond(s)
  digitalWrite(8, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(2, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
  digitalWrite(8, LOW);
  digitalWrite(3, LOW);
  digitalWrite(2, HIGH);
  delay(5000); // Wait for 5000 millisecond(s)
}
```

Output:-



Practical 10

Q]Blinking LED-using RaspberryPI

Aim: Displaying Blinking LED using RaspberryPI

Code:-

Output:-

Practical 11

Q]7-Segment LED Display-using RaspberryPI

Aim: Displaying 7-Segment LED Display-using RaspberryPI

Code:-

Output:-