1. **Program to interface led using bit addressing**

```
#include<reg51.h>                    //include at89c51 microcontoller header file
void delay_ms(unsigned int);
sbit led=P0^0;                       //connect p0.0 to led

void main(void) {
        while(1) {                   //infinite loop
                led=0;               //led off
                delay_ms(500);       //delay 500 milli seconds
                 led=1;              //led on
                delay_ms(500);       //delay 500 milli seconds
        }
}
//generates delay in milli seconds
void delay_ms(unsigned int i) {
        unsigned int j;
        while(i-->0) {
                for(j=0;j<500;j++)
                {
                        ;
                }
        }
}
```

2. **Program to interface leds using byte addressing**

```
#include<reg51.h>                    //include at89c51 microcontoller header file
#define led P0                       //connect lower nibble of p0 to leds
void delay_ms(unsigned int);
void main(void) {
      while(1) {                     //infinite loop
           led=0x0c;                 //led3 & led4 are on
           delay_ms(400);            //delay 400 milli seconds
           led=0x03;                 //led1 & led2 are on
           delay_ms(400);            //delay 400 milli seconds
       }
}
//generates delay in milli seconds
void delay_ms(unsigned int i) {
        unsigned int j;
        while(i-->0) {
                for(j=0;j<500;j++)
                {
                        ;
                }
        }
}
```

3. **Program to interface led & Switch using bit addressing**

```
#include<reg51.h>                          //include at89c51 microcontoller header file
sbit led=P0^0;                             //connect P0.0 to led
sbit sw=P2^0;                              //connect P2.0 to switch
void delay_ms(unsigned int);
void main(void) {
        while(1) {                         //infinite loop
                if(sw==0)  {               //check if switch is pressed
                        led= ~led;         // toggle led
                        delay_ms(200);     //delay 200 milli seconds
        }
        }
}
//generates delay in milli seconds
void delay_ms(unsigned int i)
{
unsigned int j;
        while(i-->0)
        {
                for(j=0;j<500;j++)
                {
                        ;
                }
        }
}
```

4. Program to interface Buzzer using byte addressing

```
#include<reg51.h>                //include at89c51 microcontoller header file
#define buzzer P0                //connect lower nibble of p0 to leds
void delay_ms(unsigned int);
void main(void) {
        while(1) /{
                buzzer=0x2F;       //Buzzer & led are ON
                delay_ms(200); //delay 200 milli seconds
                 buzzer=0x00;       //Buzzer OFF
                delay_ms(400); //delay 400 milli seconds
        }
}
void delay_ms(unsigned int i) {
        unsigned int j;
        while(i-->0){
                for(j=0;j<500;j++)
                {
                        ;
                }
        }
}
```

5. **To interface relay**

```
#include<reg51.h>
#define relay P0
void delay_ms(unsigned int);
void main(void) {
   while(1)   {
      relay=0x16;                  // Relay ON, LED 2 & LED 3 ON
      delay_ms(500);
      relay=0x09;                 // Relay OFF, LED 1 & LED 4 ON
      delay_ms(500);
   }
}
void delay_ms(unsigned int i) {
   unsigned int j;
   while(i-->0){
      for(j=0;j<500;j++){
         ;
      }
   }
}
```

6. **DC Motor**

```
#include<reg51.h>
sbit mtr_1 =P2^0;
sbit mtr_2 =P2^1;
sbit pwm_control =P2^3;
sbit key2_REV   = P2^4;
sbit key1_FRW   = P2^5;
sbit speed_dec  = P2^6;
sbit speed_inc  = P2^7;


void main(void)
{
    unsigned int count,value=500;
    while(1)
    {
        if(key1_FRW == 0)
```

```
        {
            mtr_1 =0;
            mtr_2 =1;
        //  pwm_control = 1;
         }

         if(key2_REV == 0)
        {
            mtr_1 =1;
            mtr_2 =0;
        //  pwm_control = 1;
         }

         if(speed_dec ==0 )
         {
            value -= 1;
            if(value <= 15)
            {
                value = 16;
            }
         }

         if(speed_inc ==0 )
         {
            value += 1;
            if(value >= 1000)
            {
                value = 999;
            }
         }
```

```
        pwm_control=1;
        for(count=0;count<= value ;count++);

        pwm_control=0;
        for(count=0;count<= 500;count++);


    }
    while(1);
}
```

7. **Stepper Motor**

```
#include<reg51.h>
sbit sw=P0^0;
#define motor P
void delay_ms(unsigned int);
bit dir=0;
void main(void) {
    while(1) {
        do {
            if(dir==0) {
                motor=0x06;
                delay_ms(1);
                motor=0x0a;
                delay_ms(1);
                motor=0x09;
                delay_ms(1);
```

```
                motor=0x05;
                delay_ms(1);
            }
            else if(dir==1){
                motor=0x05;
                delay_ms(1);
                motor=0x09;
                delay_ms(1);
                motor=0x0a;
                delay_ms(1);
                motor=0x06;
                delay_ms(1);
            }
        }
        while(sw==1);
        delay_ms(200);
        dir=~dir;
    }
}
void delay_ms(unsigned int i) {
    unsigned int j;
    while(i-->0){
        for(j=0;j<500;j++){
            ;
        }
    }
}
```

8. **Seven Segment Display**

```
#include<reg51.h>
#define sevensegment_data P1
sbit DISP1_se1=P0^3;
sbit DISP2_se1=P0^2;
sbit DISP3_se1=P0^1;
sbit DISP4_se1=P0^0;
sbit lcd_back_light=P0^7;

void delay_ms(unsigned int);
void main(void) {
    unsigned char count0=0, count1=0, count2=0, count3=0, count4=0,
count5=0, count6=0;
    unsigned
bcd_code[]={0x3F,0x06,0x05,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
    do {
        do {
            do {
                do {
                    do {
                        DISP1_se1=0;
                        sevensegment_data=bcd_code[count1];
                        delay_ms(2);
                        DISP1_se1=1;

                        DISP2_se1=0;
                        sevensegment_data=bcd_code[count2];
                        delay_ms(2);
                        DISP2_se1=1;

                        DISP3_se1=0;
                        sevensegment_data=bcd_code[count3];
                        delay_ms(2);
                        DISP3_se1=1;

                        DISP4_se1=0;
```

```
                        sevensegment_data=bcd_code[count4];
                        delay_ms(2);
                        DISP4_se1=1;

                        count0+=1;
                    }
                    while(count0<=25);
                    count0=0;
                    count1+=1;
                }
                while(count1<=9);
                count1=0;
                count2+=1;
            }
            while(count2<=9);
            count2=0;
            count3+=1;
        }
        while(count3<=9);
        count3=0;
        count4+=1;
    }
    while(count4<=9);
    count4=0;
    while(1);
}
void delay_ms(unsigned int itime)   {
    unsigned int i,j;
    for(i=0;i<itime;i++)
        for(j=0;j<100;j++) {
            ;
        }
}
```

9. **Interface LCD**

```
#include<reg51.h>
#define ldata P1
sbit rs=P0^4;
sbit rw=P0^5;
sbit en=P0^6;
sbit back_lite=P0^7;
void ms_delay(unsigned int);
void lcdcmd(unsigned char);
void lcddata(unsigned char);
void lcdready();

void main(void) {
    unsigned char lcd_command[]={0x38,0x0e,0x01,0x06,0x83};
    unsigned char lcd_message[]="Hello";

    unsigned char lcd_command1[]={0x38,0x0e,0x06,0xc0,0xc2};
    unsigned char lcd_message1[]="CBIT";

    unsigned char c,d;
    back_lite=0;
    for(c=0;c<5;c++) {
        lcdcmd(lcd_command[c]);
    }
    for(d=0;d<5;d++) {
        lcddata(lcd_message[d]);
        ms_delay(30);
    }
    back_lite=1;
    for(c=0;c<5;c++) {
        lcdcmd(lcd_command1[c]);
    }
    for(d=0;d<4;d++) {
        lcddata(lcd_message1[d]);
        ms_delay(30);
    }
```

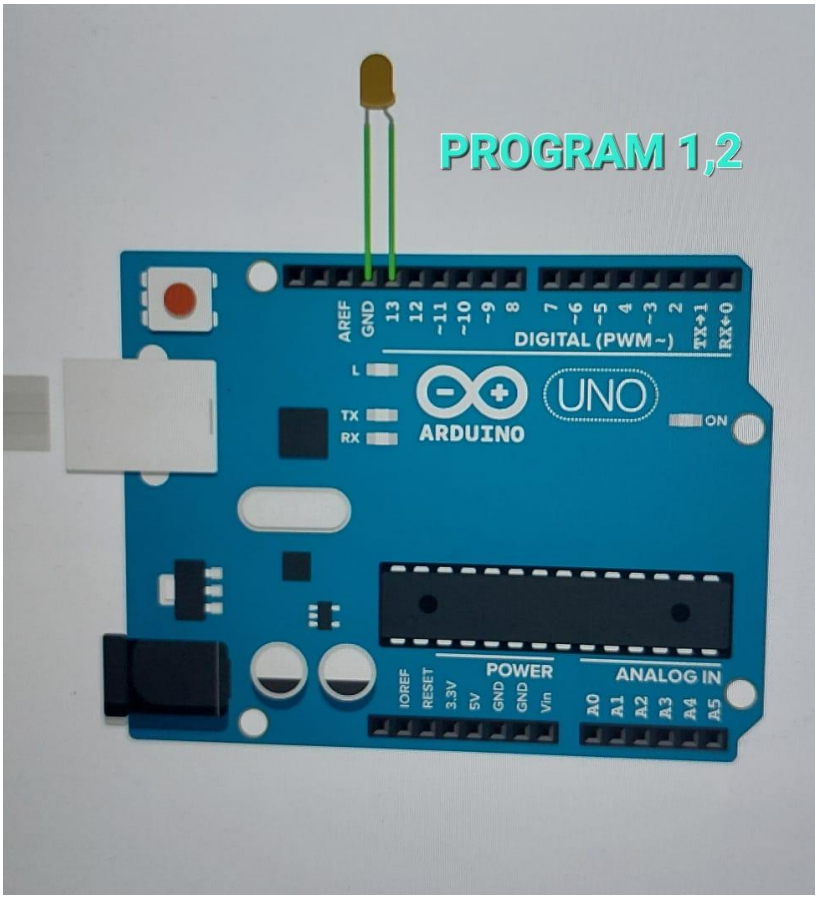
```
    while(1) {
    }
}
void lcdcmd(unsigned char v) {
    ldata=v;   // put the value on the pins
    rs=0;
    rw=0;
    en=1;
    ms_delay(1);
    en=0;
    return;
}
void lcddata(unsigned char v) {
    ldata=v;   // put the value on the pins
    rs=1;
    rw=0;
    en=1;
    ms_delay(1);
    en=0;
    return;
}
void ms_delay(unsigned int time) {
    unsigned int i,j;
    for(i=0;i<time;i++){
        for(j=0;j<1000;j++);
    }
}
```

```c
#include<reg51.h>
#define sevensegment_data P1

// Define BCD values for each letter
unsigned char bcd_code[] = {0x63, 0x79, 0x50, 0x06}; // BCD values for
'C', 'O', 'L', 'D'

sbit DISP1_se1 = P0^3;
sbit DISP2_se1 = P0^2;
sbit DISP3_se1 = P0^1;
sbit DISP4_se1 = P0^0;

void delay_ms(unsigned int);

void main(void){
    unsigned char count = 0;
    while(1){
        // Loop through each letter in "COLD"
            // Display the corresponding letter on each digit
            DISP1_se1 = 0;
            sevensegment_data = bcd_code[0];
            delay_ms(2);
            DISP1_se1 = 1;

            DISP2_se1 = 0;
            sevensegment_data = bcd_code[1]; // Cycle through the
letters
            delay_ms(2);
            DISP2_se1 = 1;

            DISP3_se1 = 0;
            sevensegment_data = bcd_code[2]; // Cycle through the
letters
            delay_ms(2);
            DISP3_se1 = 1;

            DISP4_se1 = 0;
            sevensegment_data = bcd_code[3]; // Cycle through the
letters
            delay_ms(2);
            DISP4_se1 = 1;
    }
```

```
}

void delay_ms(unsigned int itime){
    unsigned int i, j;
    for (i = 0; i < itime; i++){
        for (j = 0; j < 100; j++){
            // Do nothing
        }
    }
}
```

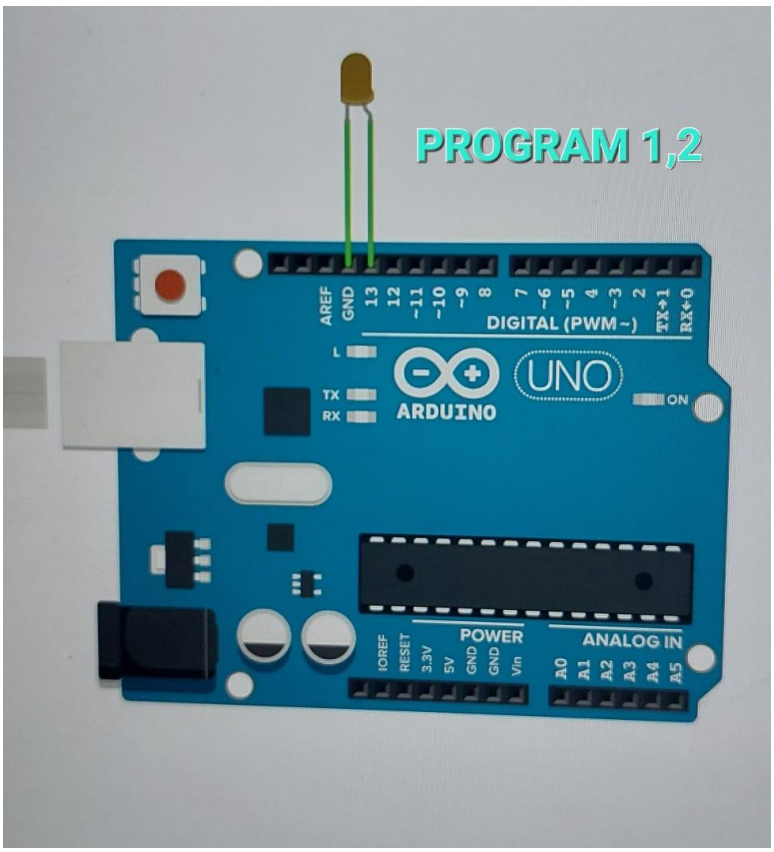
Q. Write a program and demonstrate interfacing the following with Raspberry Pi / Arduino.

a. Blinking an LED



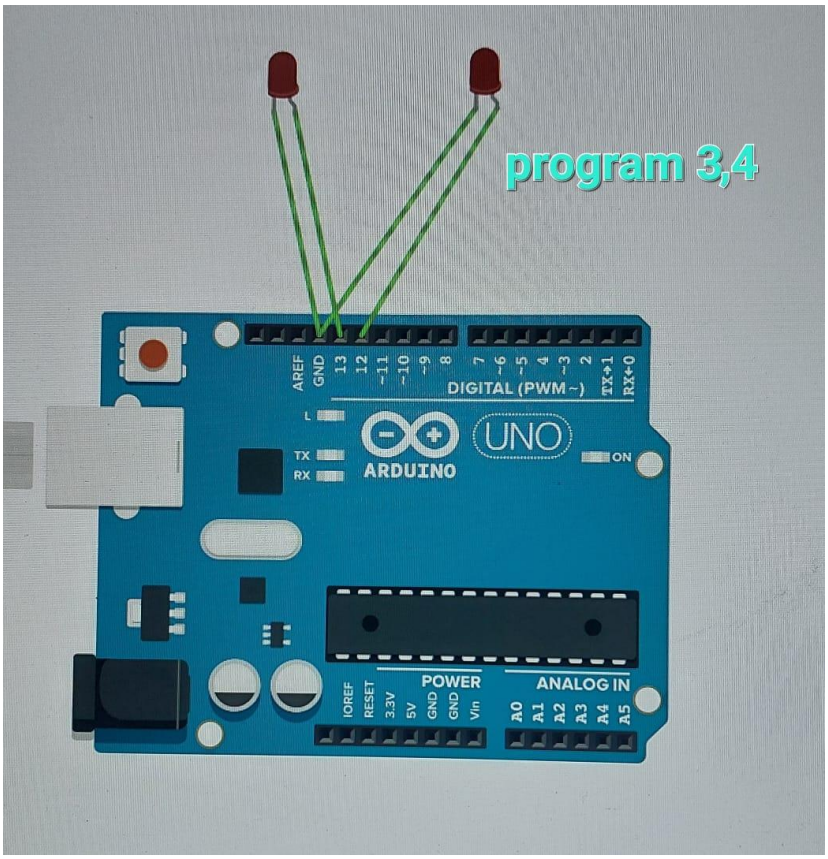

```
int ledPin = 13;

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);

    Serial.println("LED ON");
    digitalWrite(ledPin, HIGH);
    delay(1000);

    Serial.println("LED OFF");
    digitalWrite(ledPin, LOW);
    delay(1000);
}

void loop() {
    // Empty loop
}
```

b. Blinking an LED using loops



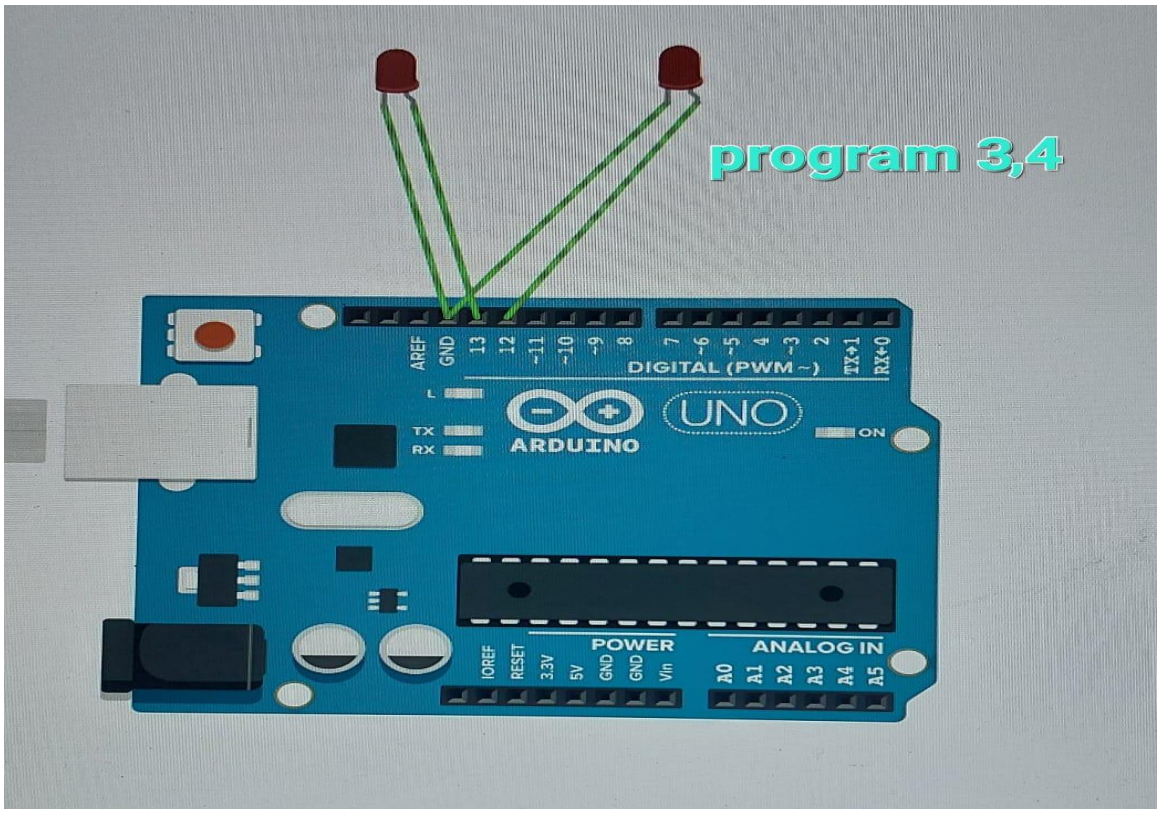

```
int ledPin = 13;
void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("LED ON");
    digitalWrite(ledPin, HIGH);
    delay(1000);

    Serial.println("LED OFF");
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```

c. Blinking 2 LED’s alternatively



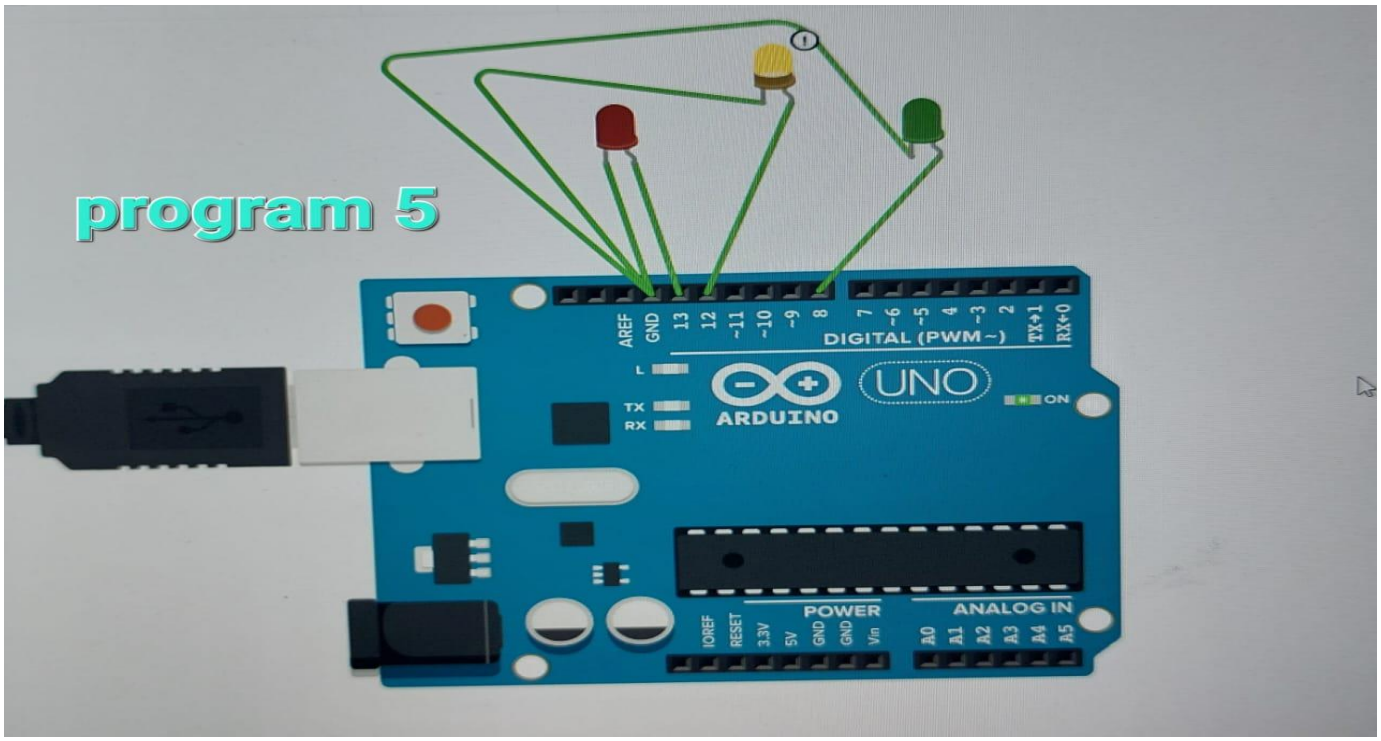

```
int ledPin1 = 13;
int ledPin2 = 12;
void setup() {
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("LED1 ON, LED2 OFF");
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, LOW);
    delay(1000);

    Serial.println("LED1 OFF, LED2 ON");
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
    delay(1000);
}
```

d. Blinking 2 LED’s simultaneously




```
int ledPin1 = 13;
int ledPin2 = 12;
void setup() {
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("LED1 ON, LED2 ON");
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    delay(1000);

    Serial.println("LED1 OFF, LED2 OFF");
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    delay(1000);
}
```

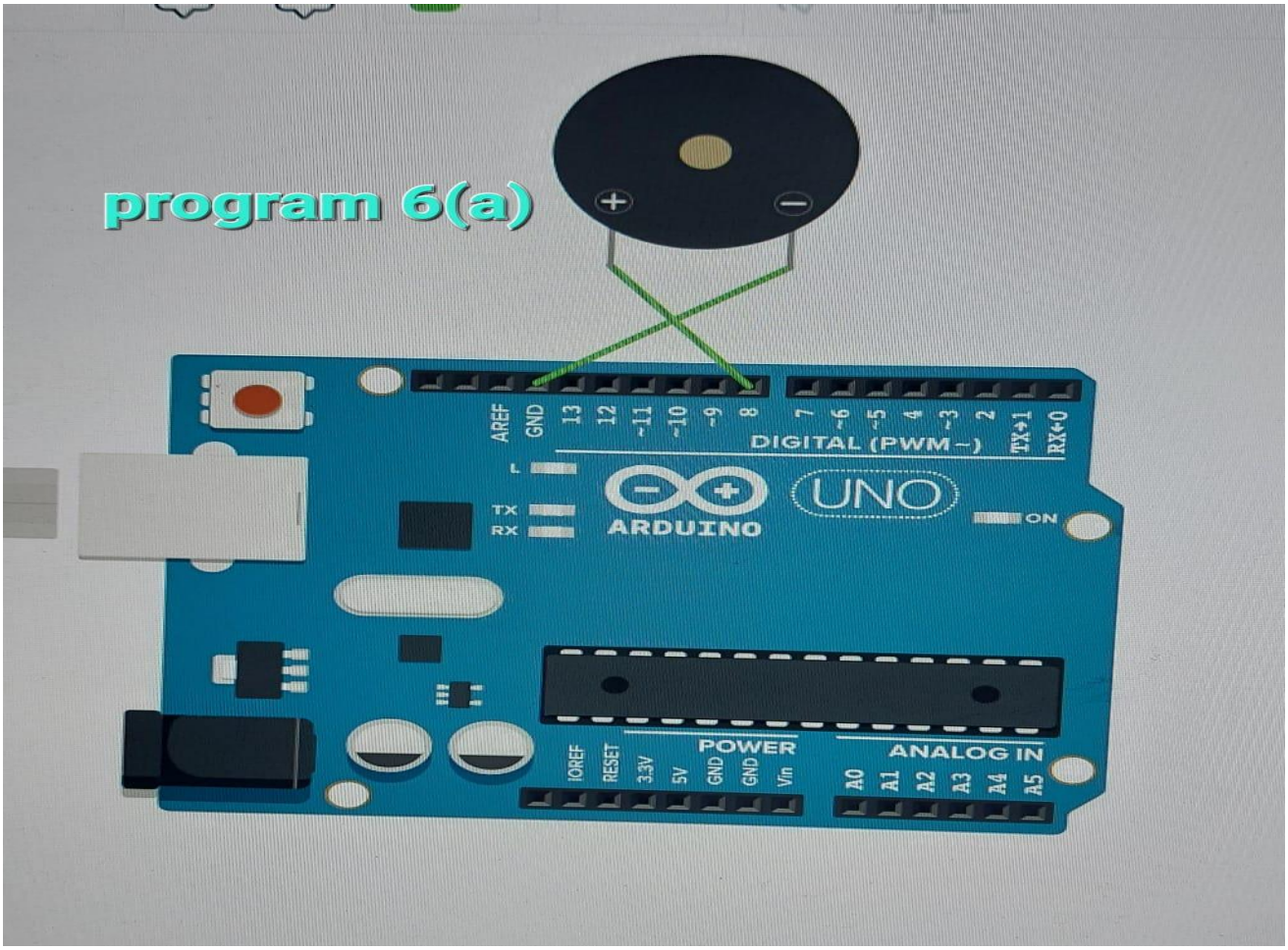
e. Traffic Light Signal system




```
int ledPin1 = 13;
int ledPin2 = 12;
int ledPin3 = 8;
void setup() {
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("Red");
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3,   LOW);
    delay(5000);

    Serial.println("Yellow");
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
    digitalWrite(ledPin3, LOW);
    delay(5000);
```

```
    Serial.println("Green");
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, HIGH);
    delay(5000);
}
```

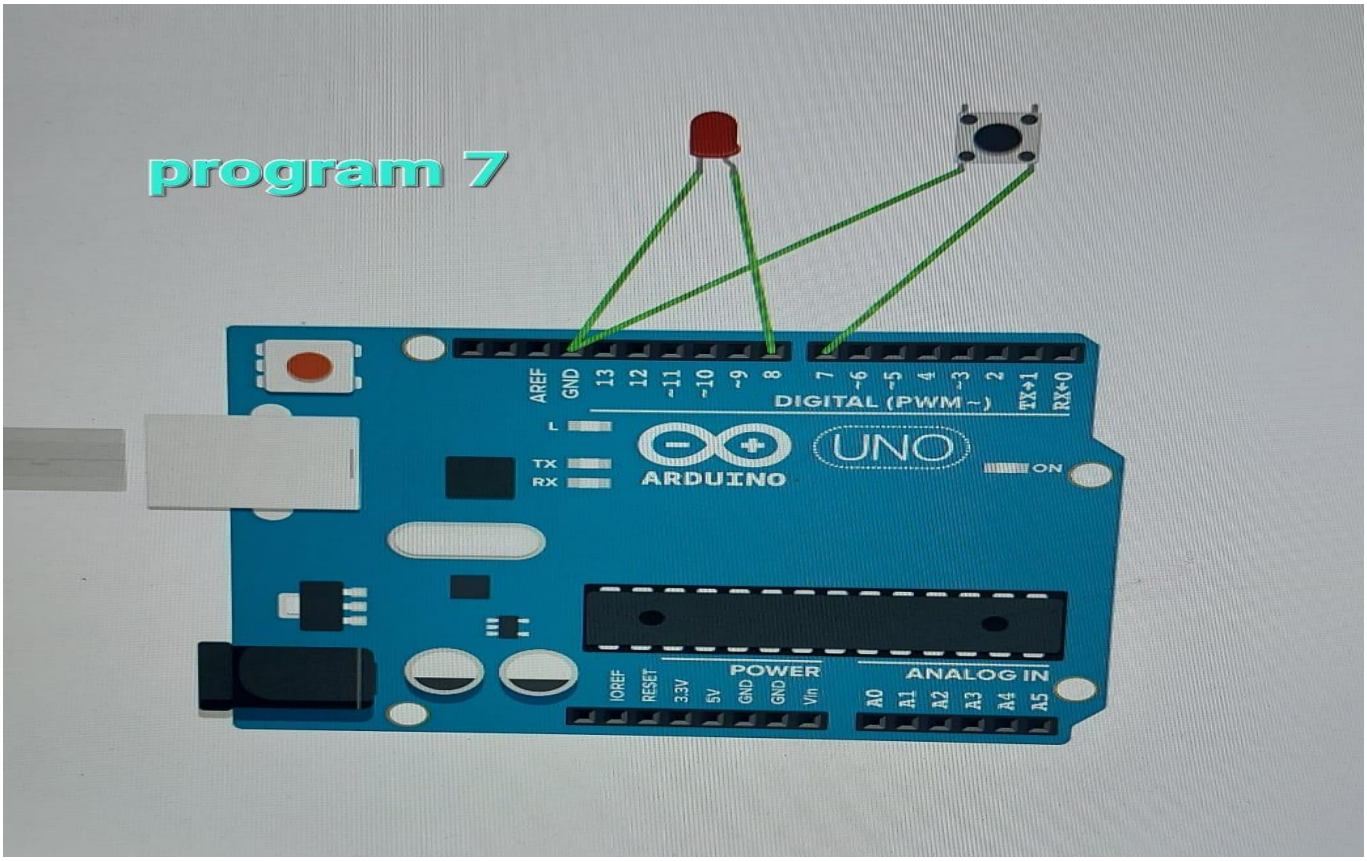
f. Buzzer with and without switch




```
int buzzerPin = 8;

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

void loop() {
```

```
    Serial.println("Buzzer ON");
    digitalWrite(buzzerPin, HIGH);
    delay(1000);

    Serial.println("Buzzer OFF");
    digitalWrite(buzzerPin, LOW);
    delay(1000);
}
```

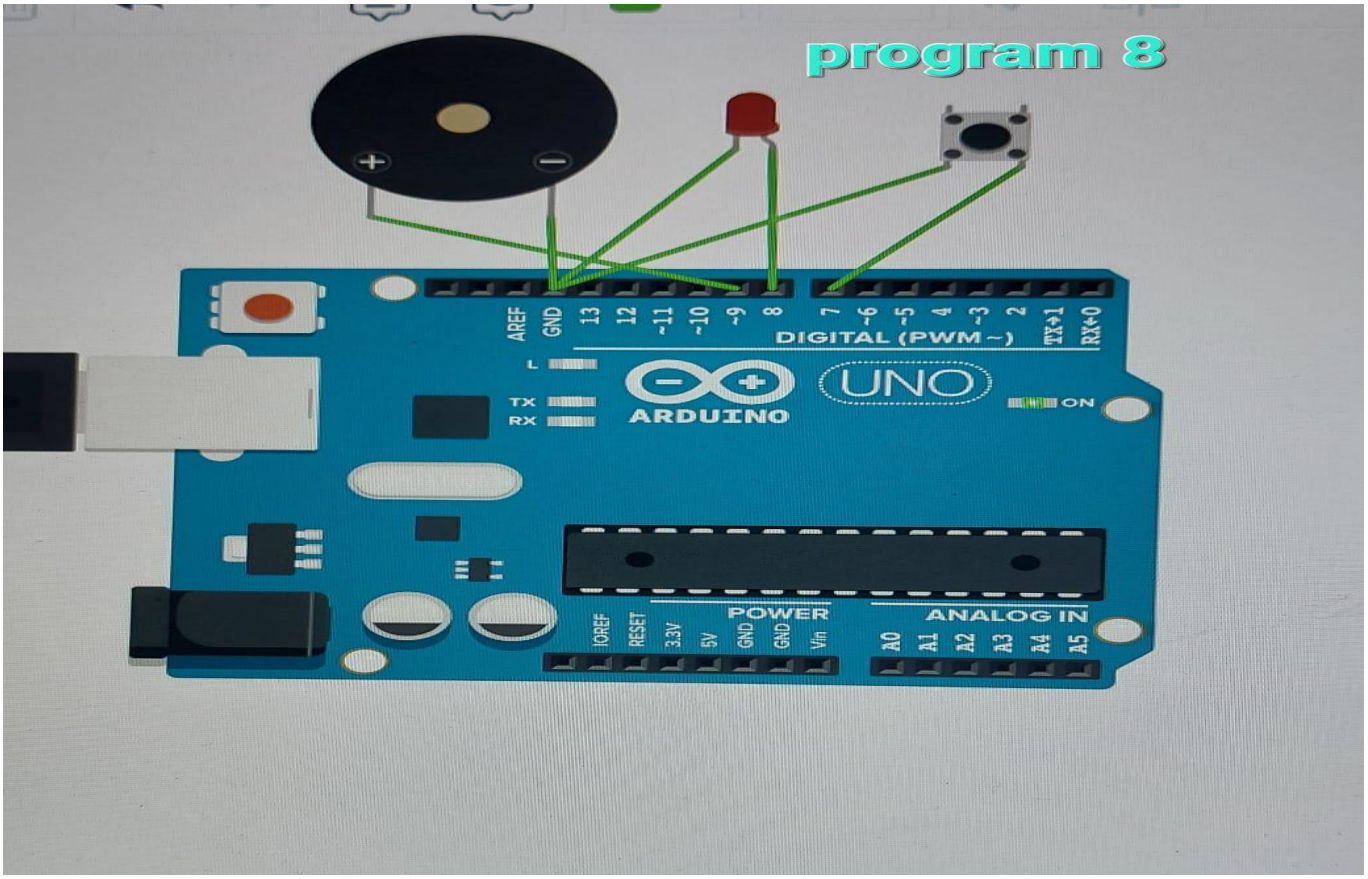
g. LED with Switch




```
int ledPin = 8;
int switchPin = 7;
int switchState = 0;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(switchPin, INPUT_PULLUP);
    Serial.begin(9600);
}
```

```
void loop() {
    switchState = digitalRead(switchPin);
    if (switchState == LOW) {
        Serial.println("LED ON");
        digitalWrite(ledPin, HIGH);
    }
    else {
        Serial.println("LED OFF");
        digitalWrite(ledPin, LOW);
    }
    delay(100);
}
```

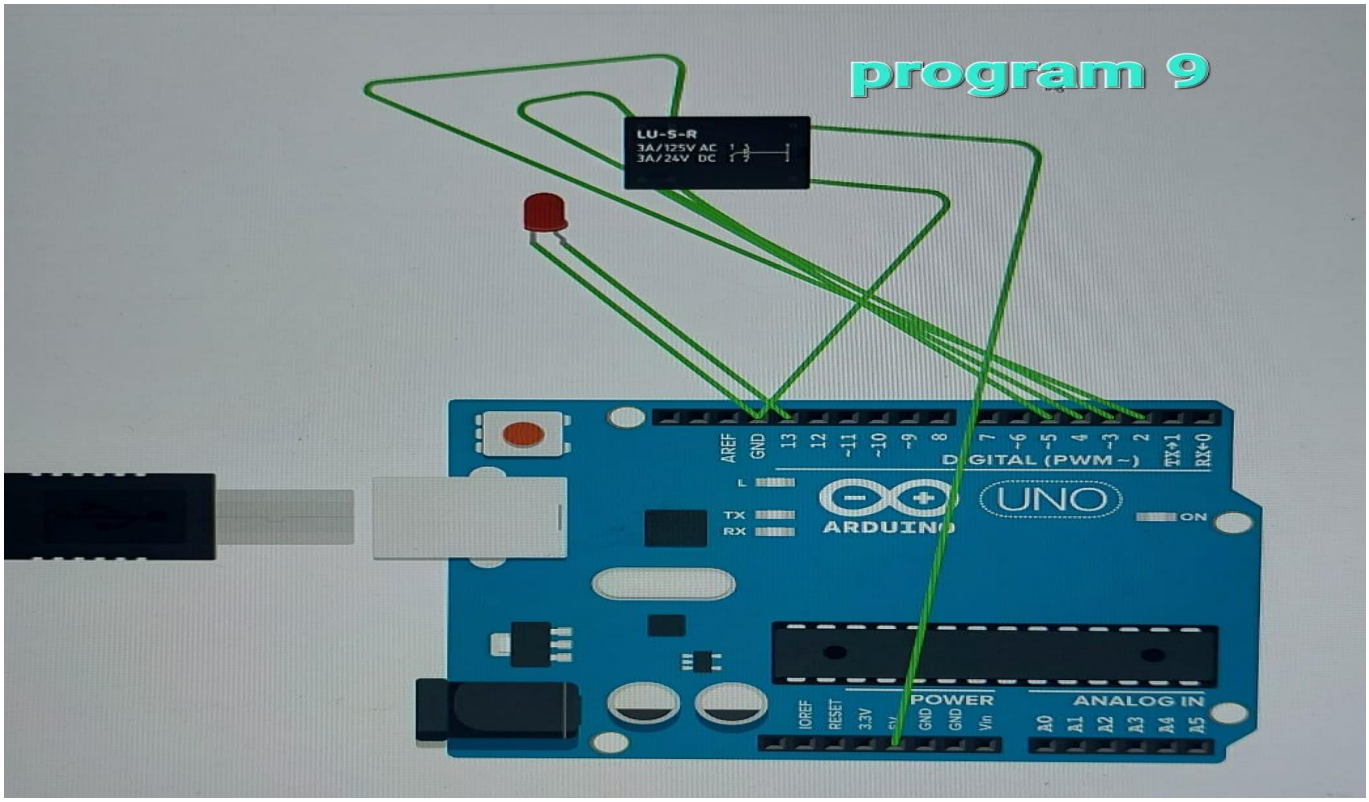
h. LED & Buzzer with Switch (Alert system)




```
int ledPin = 8;
int buzzerPin = 9;
int switchPin = 7;
int switchState = 0;
```

```
void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    pinMode(switchPin, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    switchState = digitalRead(switchPin);
    if (switchState == LOW) {
        Serial.println("LED and Buzzer ON");
        digitalWrite(ledPin, HIGH);
        digitalWrite(buzzerPin, HIGH);
    }
    else {
        Serial.println("LED and Buzzer OFF");
        digitalWrite(ledPin, LOW);
        digitalWrite(buzzerPin, LOW);
    }

    delay(100);
}
```

i. Relays



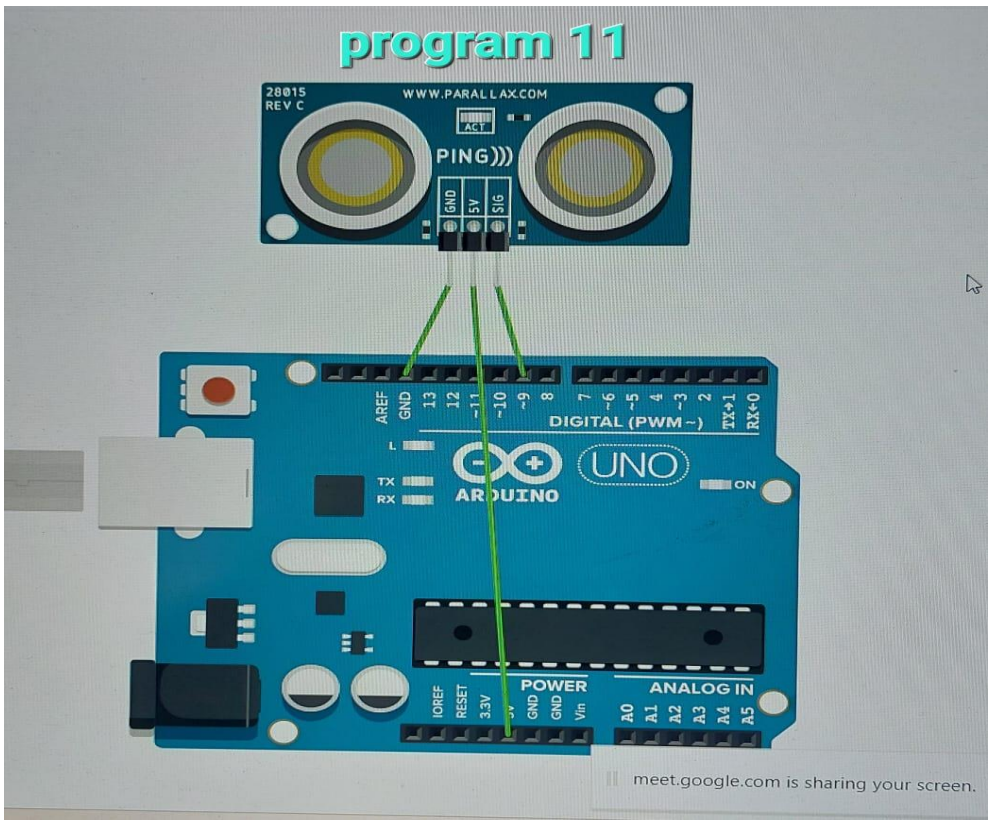

```
int relayPin1 = 2;
int relayPin2 = 3;
int relayPin3 = 4;
int relayPin4 = 5;
int ledpin = 13;
void setup() {
    pinMode(relayPin1, OUTPUT);
    pinMode(relayPin2, OUTPUT);
    pinMode(relayPin3, OUTPUT);
    pinMode(relayPin4, OUTPUT);
    pinMode(ledpin,OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.println("on");
    digitalWrite(ledpin, HIGH);
    digitalWrite(relayPin1, HIGH);
    digitalWrite(relayPin2, HIGH);
    digitalWrite(relayPin3, HIGH);
    digitalWrite(relayPin4, HIGH);
```

```
    delay(2000);

    Serial.println("off");
    digitalWrite(ledpin, LOW);
    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, LOW);
    digitalWrite(relayPin3, LOW);
    digitalWrite(relayPin4, LOW);
    delay(2000);
}
```

j. DHT Sensor (python)

```
# Temperature & Humidity Detection _ DHT Sensor

import sys
import Adafruit_DHT
import time
while True:
    humidity,temperature=Adafruuit_DHT.read_retry(11,4)
print("Temperature:",temperature)
print("Humidity",humidity)
time.sleep(0)
```

k. Distance Sensor




```
const int dataPin = 9;

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

void loop() {
    long duration, distance;
    digitalWrite(dataPin, LOW);
    delayMicroseconds(2);
    digitalWrite(dataPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(dataPin, LOW);
    pinMode(dataPin, INPUT);
    duration = pulseIn(dataPin, HIGH);
    if (duration > 0) {
        distance = duration * 0.034 / 2;
        Serial.print("Distance = ");
        Serial.print(distance);
```

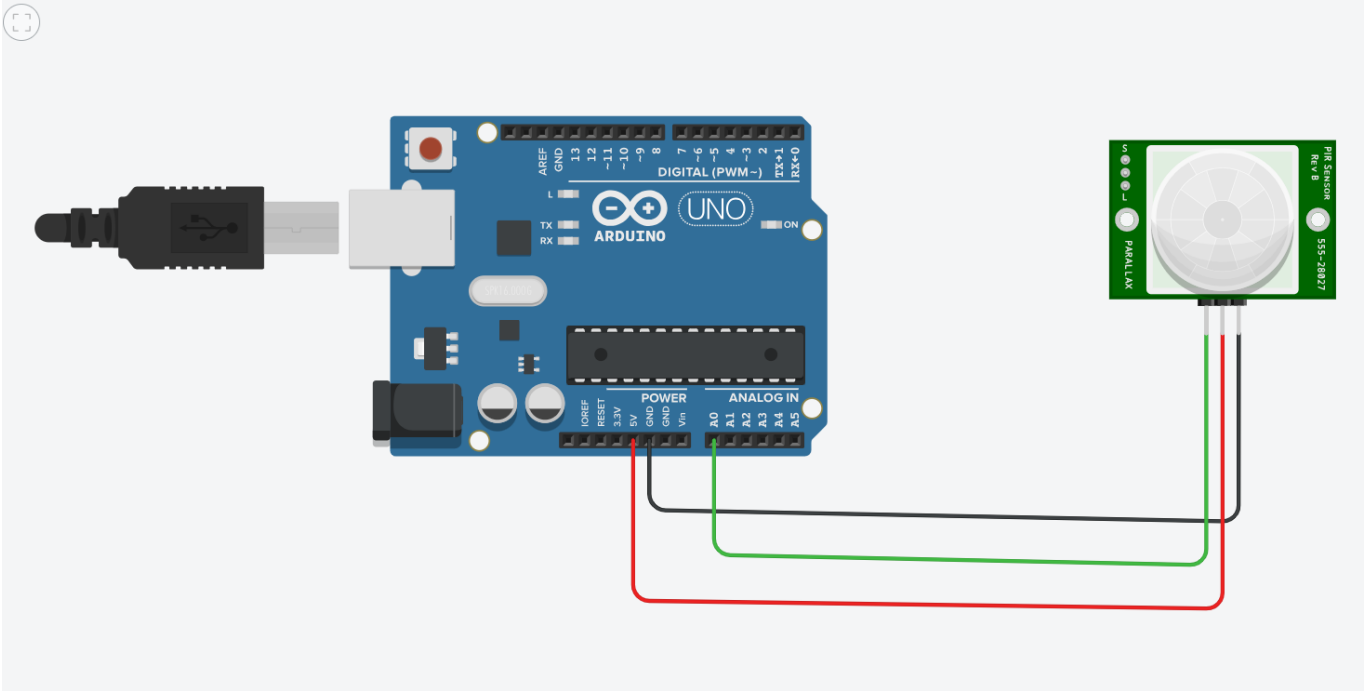
```
        Serial.println(" cm");
    }
    else {
        Serial.println("No echo received");
    }

    pinMode(dataPin, OUTPUT);
    delay(1000);
}
```

l. IR Sensor / Object Detection


```
#include <IRremote.hpp>
const int rcvPin=3;
IRrecv irrecv(rcvPin);
decode_results results;
void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn();
}

void loop() {
    if(IrReceiver.decode()) {
```

```
        auto value=
IrReceiver.decodedIRData.decodedRawData;
        switch(value) {
            case 4010852096:
                Serial.println("1");
                break;

            case 3994140416:
                  Serial.println("2");
                  break;
            default: Serial.println(value);
        }
        IrReceiver.resume();
    }
}
```

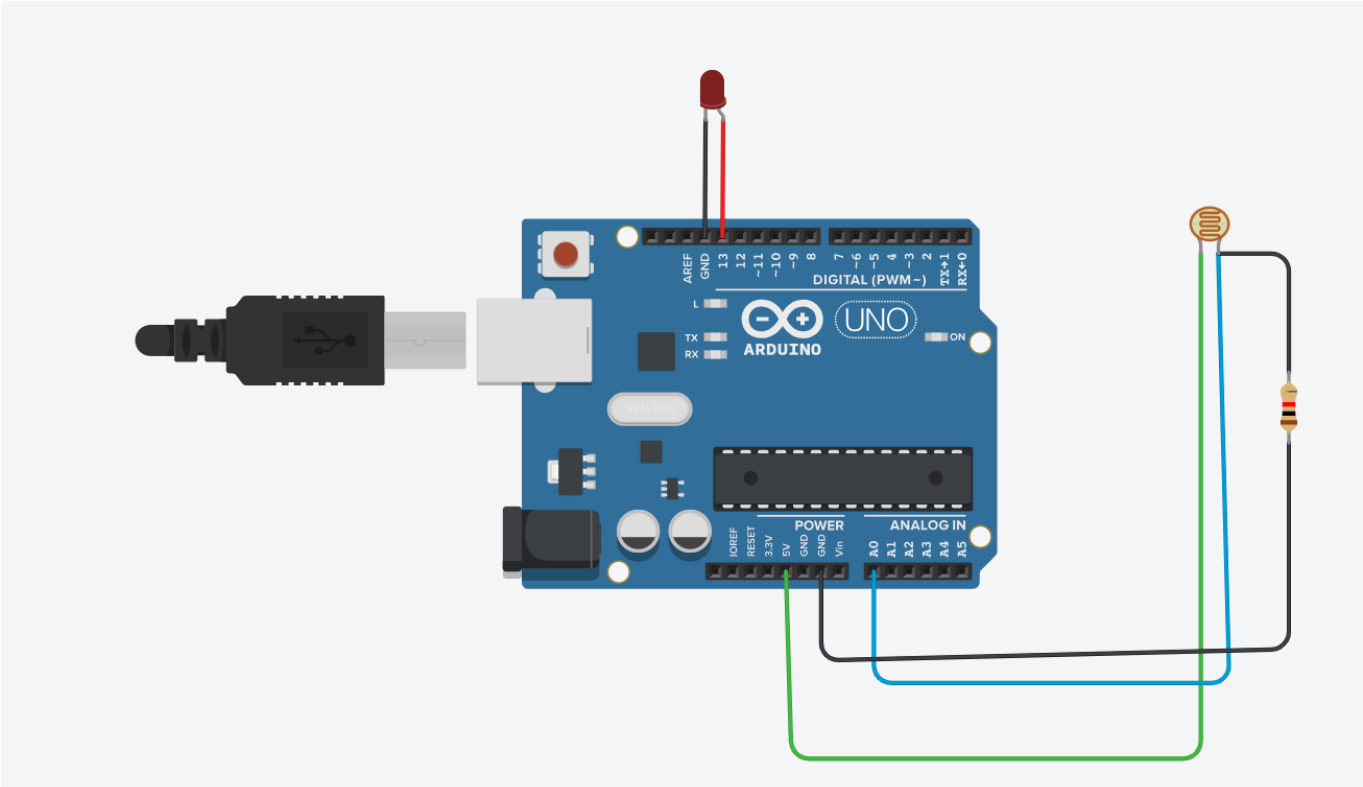
m. PIR Sensor / Motion Detection




```
int a=0;
int b=0;

void setup() {
```

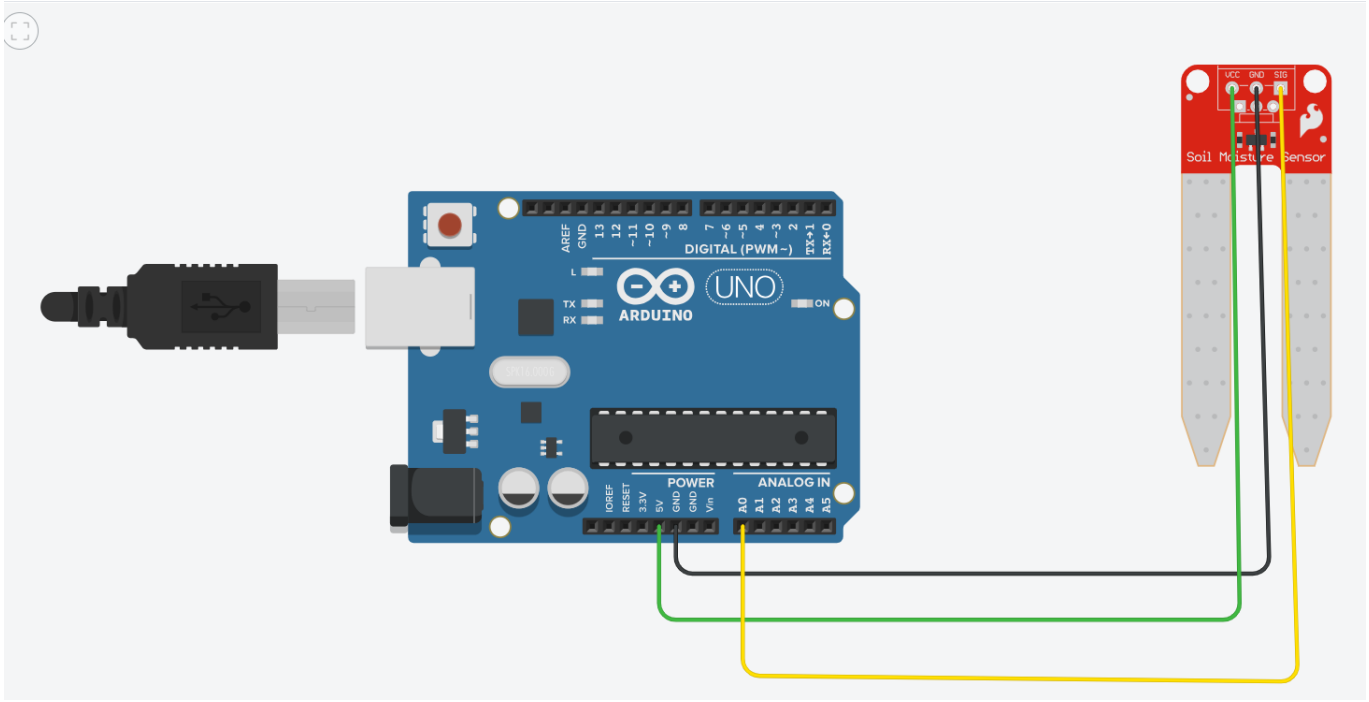
```
    Serial.begin(9600);
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop(){
    a=analogRead(A0);
    b=map(a,0,1023,0,255);
    Serial.println(b);
    if(b>100){
        Serial.println("Motion Detected");
        delay(1000);
    }
    else{
        Serial.println("No Motion Detected");
        delay(1000);
    }
}
```

n. LDR (Light Dependent Resistor)



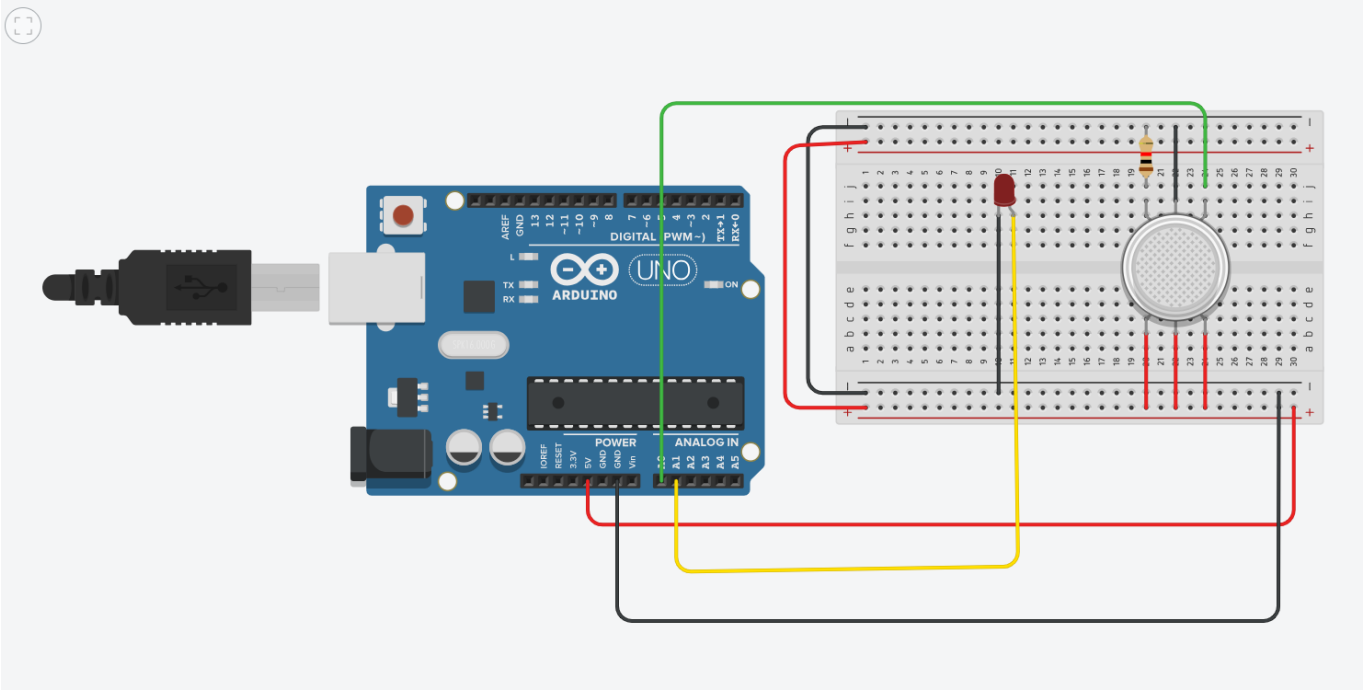
```ino
const int LEDPin = 13;
const int LDRPin = A0;
void setup (){
    Serial.begin (9600);
    pinMode(LEDPin, OUTPUT);
    pinMode(LDRPin, INPUT);
}

void loop(){
    int LDRStatus = analogRead(LDRPin);
    if(LDRStatus<=500){
        digitalWrite (LEDPin, HIGH);
        Serial.print("Current Light Intensity Value is - ");
        Serial.println(LDRStatus);
    }
    else{
        digitalWrite(LEDPin, LOW);
        Serial.print("Current Light Intensity Value is - ");
        Serial.println(LDRStatus);
    }
}
```

o. Soil Moisture sensor



```ino
int moistureValue;
float moisture_percentage;
void setup(){
    Serial.begin(9600);
}

void loop(){
    moistureValue = analogRead(A0);
    moisture_percentage = (moistureValue/539.00 *100);
    Serial.print("\nMoisture Value in percentage:");
    Serial.println(moisture_percentage);

    delay(1000);
}
```

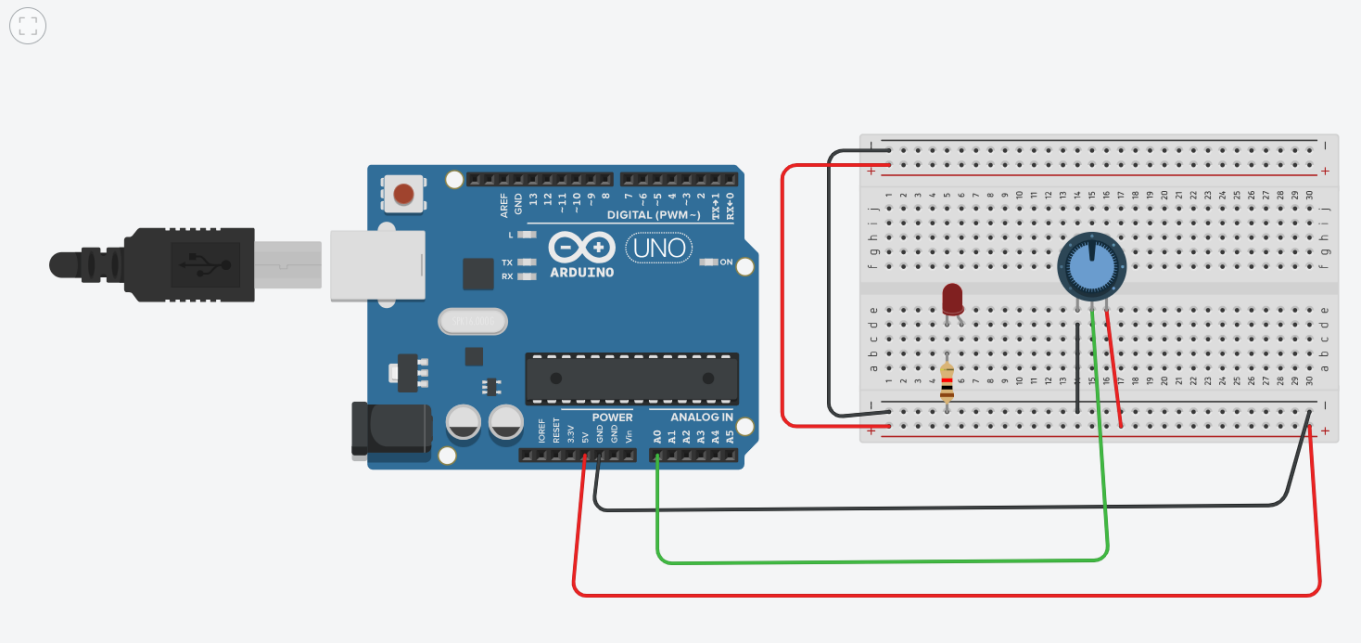
p. Gas sensor / Smoke detection




```ino
int LED = A1;
const int gas = 0;
int MQ2pin = A0;

void setup() {
```

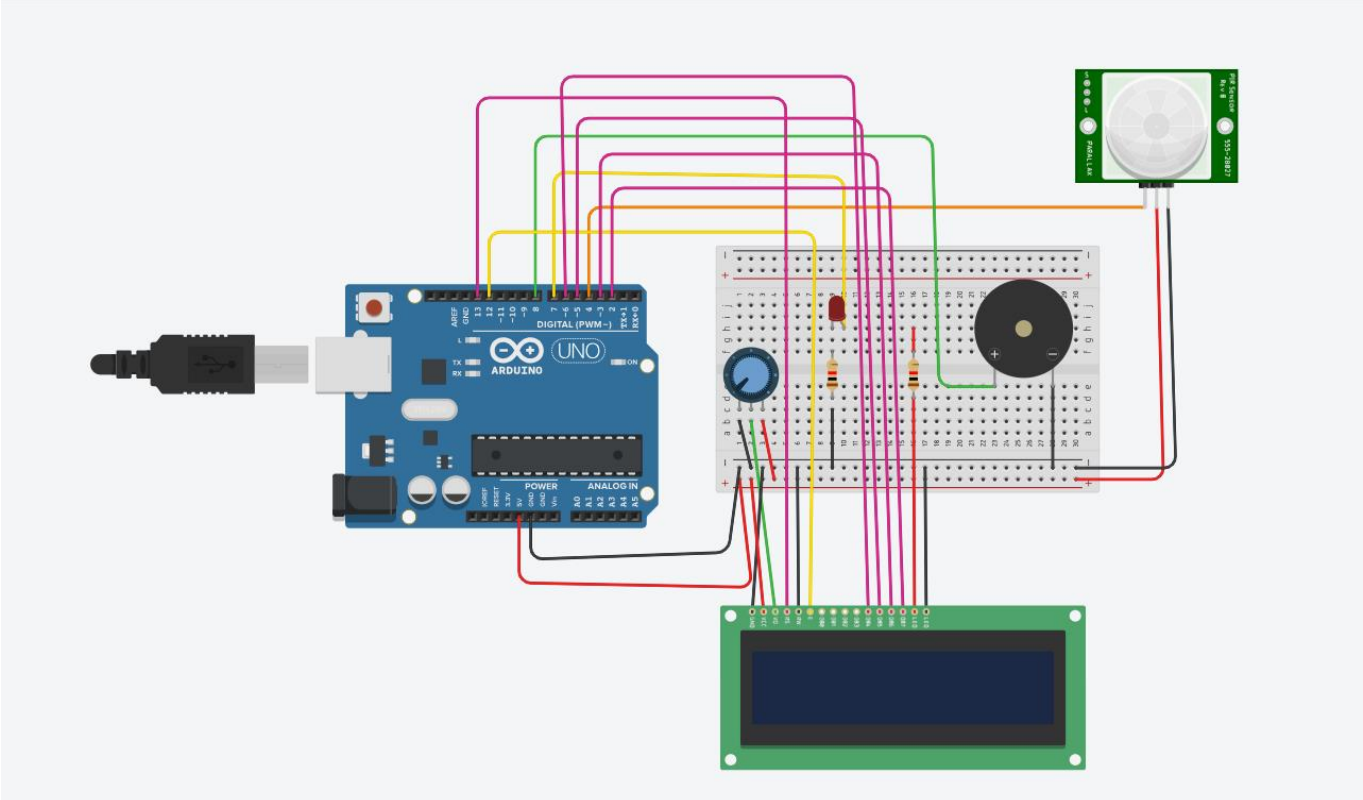
```
    Serial.begin(9600);
}

void loop() {
    float sensorValue,MQ2pin;
    sensorValue = analogRead(MQ2pin);

    if(sensorValue >= 470){
        digitalWrite(LED,HIGH);
        Serial.print(sensorValue);
        Serial.println(" |SMOKE DETECTED");
    }
    else {
        digitalWrite(LED,LOW);
        Serial.println("Sensor Value: ");
        Serial.println(sensorValue);
    }
    delay(1000);
}

float getsensorValue(int pin) {
    return (analogRead(pin));
}
```

q. Potentiometer



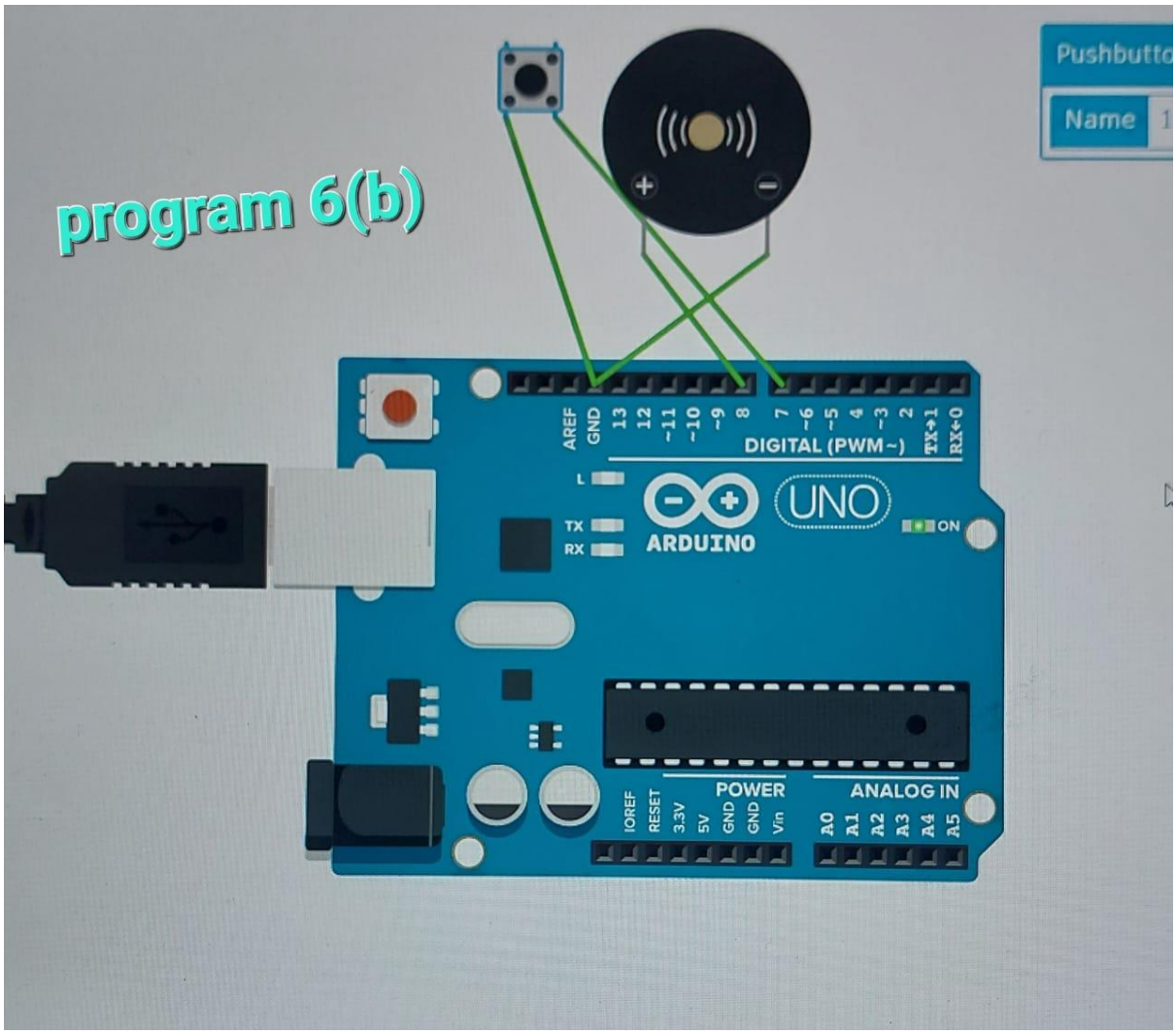
```
int sensorValue = 0;

void setup(){
    pinMode(A0, INPUT);
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop(){
    // read the value of sensor
    sensorValue = analogRead(A0);
    // turn the led on
    digitalWrite(LED_BUILTIN, HIGH);
    // pause the program for <sensorValue> milliseconds
    delay(sensorValue); // Wait for sensorValue millisecond(s)
    // turn the led off
    digitalWrite(LED_BUILTIN, LOW);
    // pause the program for <sensorValue> milliseconds
    delay(sensorValue); // Wait for sensorValue millisecond(s)
}
```

r. Smart Surveillance System



```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,6,5,3,2);
int led=7;
int PIR=4;
int buzzer=8;
int PIRstatus;

void setup(){
    lcd.begin(16,2);
    pinMode(led, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(PIR, INPUT);
    lcd.clear();
}

void loop(){
    PIRstatus=digitalRead(PIR);
    if (PIRstatus==HIGH){
        lcd.clear();
        digitalWrite(led,HIGH);
        digitalWrite(buzzer,HIGH);
        tone(buzzer, 300, 10000);
        lcd.setCursor(0,1);
        lcd.print("ALERT");

        delay(7000);
        lcd.clear();
    }
    else {
        lcd.setCursor(0,0);
        lcd.print("SAFE");
        digitalWrite(led,LOW);
        digitalWrite(buzzer,LOW);
    }
    delay(1000);
}
```

Buzzer with switch




```
int buzzerPin = 8;
int switchPin = 7;
int switchState = 0;

void setup() {
    pinMode(buzzerPin, OUTPUT);
    pinMode(switchPin, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    switchState = digitalRead(switchPin);
```

```
    if (switchState == HIGH) {
        Serial.println("Buzzer ON");
        digitalWrite(buzzerPin, HIGH);
    }
    else {
        Serial.println("Buzzer OFF");
        digitalWrite(buzzerPin, LOW);
    }
  delay(100);
}
```