

INDEX

S.NO	PROGRAM NAME	SIGN
1.	Study and Install IDE of Arduino.	
2.	Write the steps to add libraries in Arduino and setup of Arduino IDE for programming.	
3.	Write a Program using Arduino for Blink LED.	
4.	Write a Program using Arduino for Ultrasonic distance sensor.	
5.	Write a Program using Arduino for water level sensor.	
6.	Write a Program for monitoring Temperature using Arduino and LM35 Temperature Sensors.	
7.	Write a program for Arduino by using Ultrasonic sensors and servo motor (HC-SR04), and make a smart dustbin.	
8.	Write a program to shows how to fade an LED on pin 9 using the analog Write() function.	
9.	Write a program to control LED using loop.	
10.	Write a program to control LED using Serial Number	
11.	Program to measure temperature and humidity	
12.	Program for motion detector or PIR sensor using Arduino.	
13	Program to create traffic light simulator for pedestrians.	
14	Program to create Dimmable LED using potentiometer.	
15	Program to measure speed of sound using ultrasonic sensor	

1. Study and Install IDE of Arduino.

The Arduino Integrated Development Environment (IDE) is a cross-platform application written in Java that is used to write and upload programs to Arduino compatible boards. You can download the latest version of the Arduino IDE from the official Arduino website (<https://www.arduino.cc/en/software>).

Once downloaded, follow the installation instructions for your operating system (Windows, macOS, or Linux). After installation, you can open the IDE and familiarize yourself with its interface, which includes a text editor, a message area, a toolbar with common function buttons, and a series of menus.

2. Write the steps to add libraries in Arduino and setup of Arduino IDE for programming.

To add libraries in the Arduino IDE, follow these steps:

- a. Open the Arduino IDE.
- b. Go to Sketch > Include Library > Manage Libraries.
- c. In the Library Manager, search for the library you want to install.
- d. Select the library and click the "Install" button.

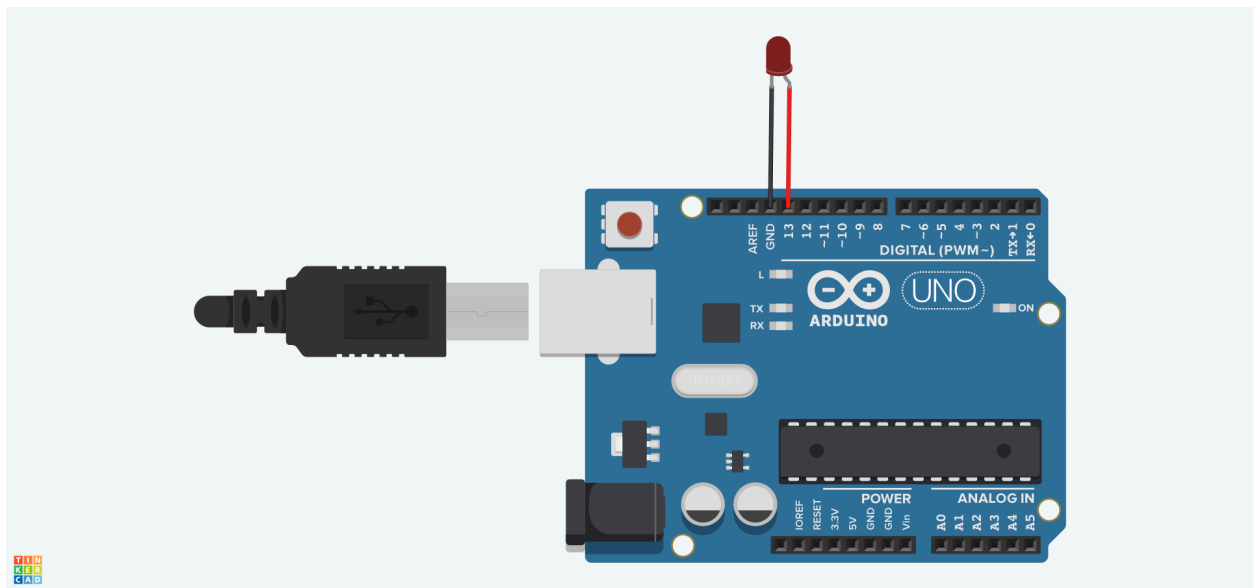
To set up the Arduino IDE for programming, follow these steps:

- a. Connect your Arduino board to your computer using a USB cable.
- b. Open the Arduino IDE.
- c. Go to Tools > Board and select your Arduino board from the list.
- d. Go to Tools > Port and select the serial port to which your board is connected.

After completing these steps, you're ready to start programming your Arduino board using the IDE.

3. Write a Program using Arduino for Blink LED.

```
void setup(){  
  pinMode(13,OUTPUT);  
}  
  
void loop(){  
  digitalWrite(13,HIGH);  
  delay(500);  
  
  digitalWrite(13,LOW);  
  delay(500);  
}
```



4. Write a Program using Arduino for Ultrasonic distance sensor.

```
const int trigPin = 8;
const int echoPin = 7;

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

  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
}

void loop(){
  digitalWrite(trigPin,LOW);
  delay(2);

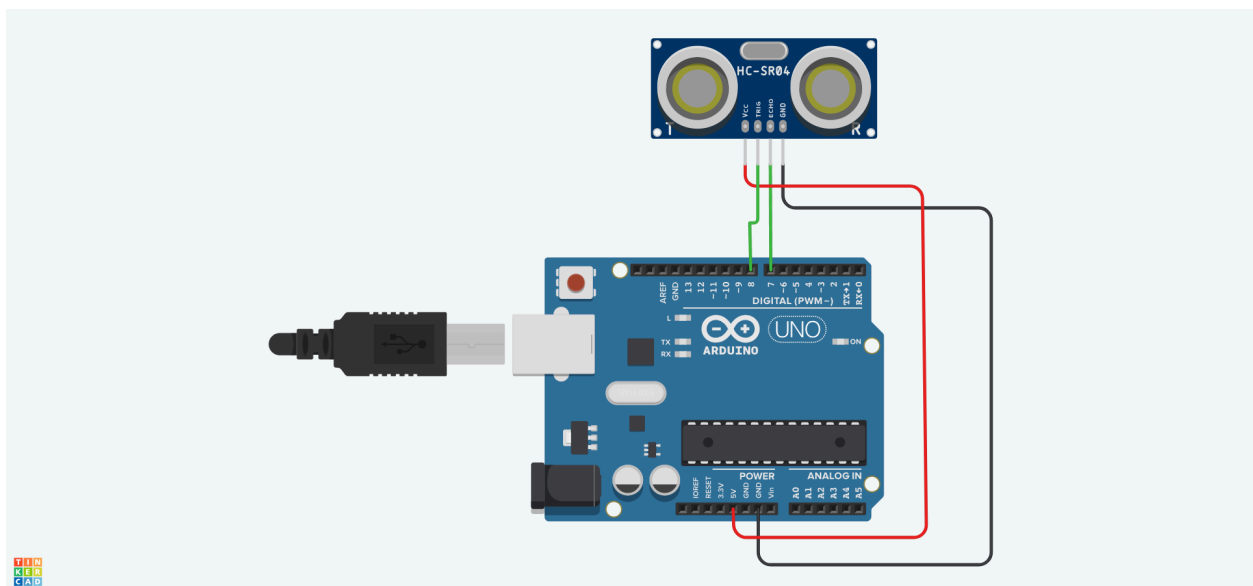
  digitalWrite(trigPin,HIGH);
  delay(10);
  digitalWrite(trigPin,LOW);

  long duration = pulseIn(echoPin, HIGH);

  int distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(100);
}
```



5. Write a Program using Arduino for water level sensor.

```
// Define the pin for the water level sensor
const int sensorPin = A0;

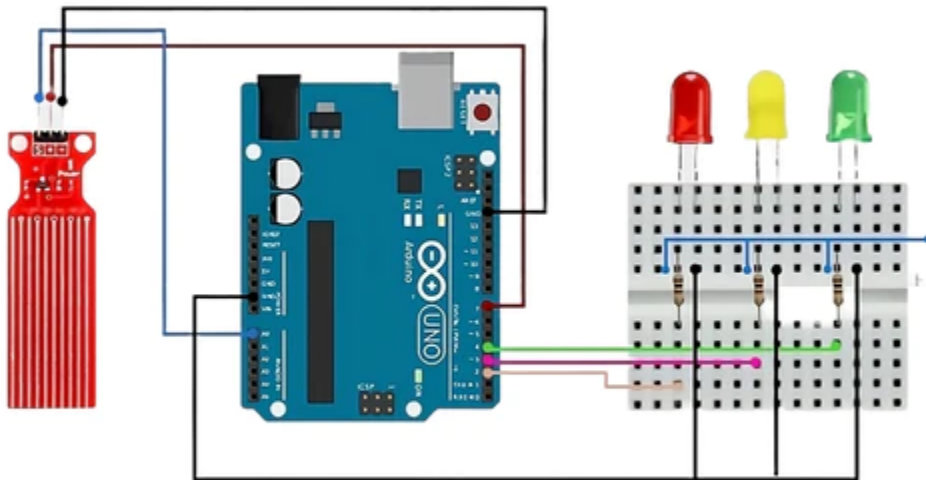
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the sensor value
  int sensorValue = analogRead(sensorPin);

  // Map the sensor value to a percentage
  int waterLevel = map(sensorValue, 0, 1023, 0, 100);

  // Print the water level to the serial monitor
  Serial.print("Water Level: ");
  Serial.print(waterLevel);
  Serial.println("%");

  delay(500); // Wait for 500 milliseconds before repeating the loop
}
```



6. Write a Program for monitoring Temperature using Arduino and LM35 Temperature Sensors.

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

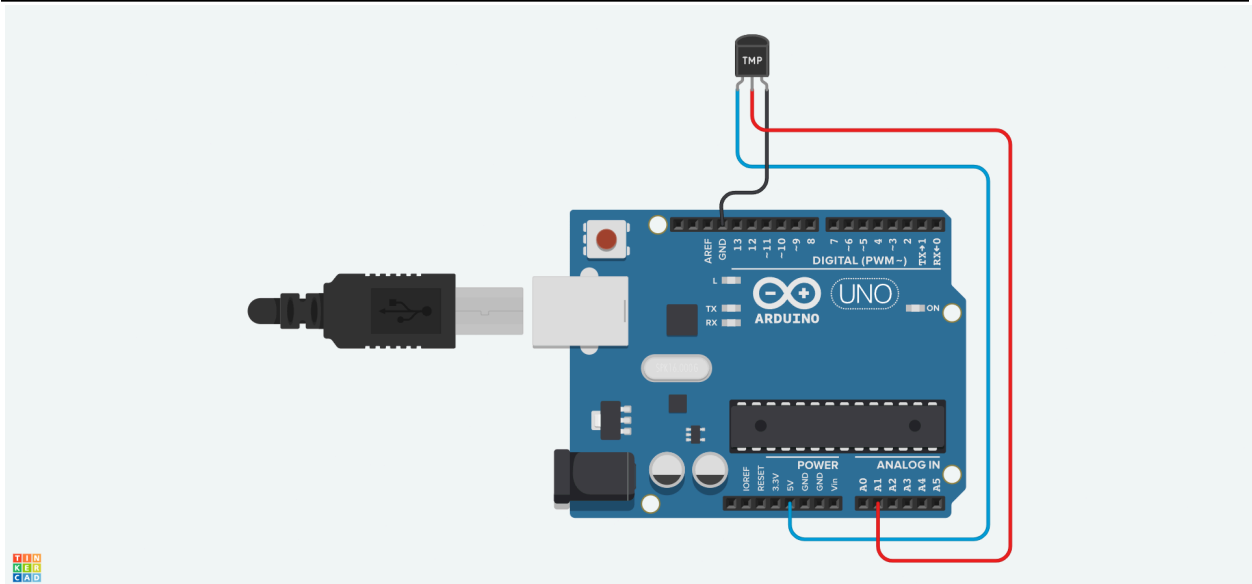
void loop()
{
  int sensorValue = analogRead(A1);

  // Convert the sensor value to voltage
  int voltage = sensorValue * (5.0 / 1024.0);

  // Convert voltage to temperature (LM35 gives 10mV per degree Celsius)
  float temperature = voltage * 100;

  // Print the temperature to the serial monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");

  delay(1000); // Wait for 1 second before repeating the loop
}
```



7. Write a program for Arduino by using Ultrasonic sensors and servo motor (HC-SR04), and make a smart dustbin.

```
#include <Servo.h>

// Define pin numbers for ultrasonic sensor
const int trigPin = 9;
const int echoPin = 10;

// Define pin number for servo motor
const int servoPin = 6;

// Create a Servo object
Servo myServo;

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Set up the ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Attach the servo motor to the designated pin
  myServo.attach(servoPin);
}

void loop() {
  // Generate a trigger pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the echo pulse duration
  long duration = pulseIn(echoPin, HIGH);

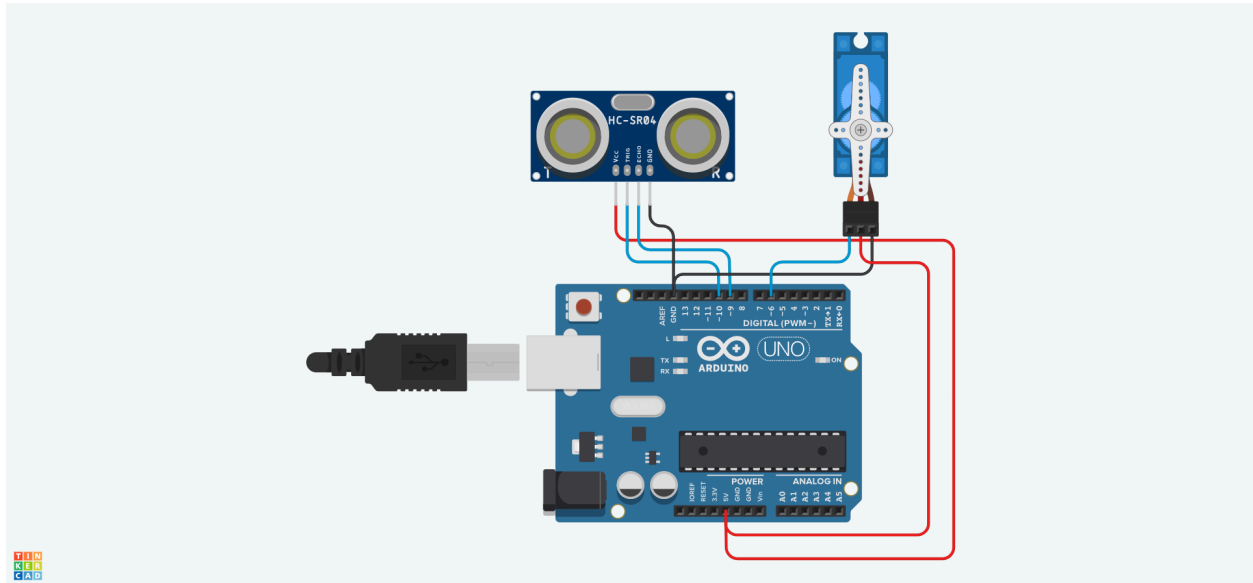
  // Calculate the distance
  int distance = duration * 0.034 / 2;

  // Print the distance to the serial monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Open the dustbin lid if an object is detected within 20 cm
  if (distance < 20) {
```

```
myServo.write(90); // Open the lid
delay(3000); // Keep the lid open for 3 seconds
myServo.write(0); // Close the lid
}

delay(100); // Wait for 100 milliseconds before repeating the loop
}
```



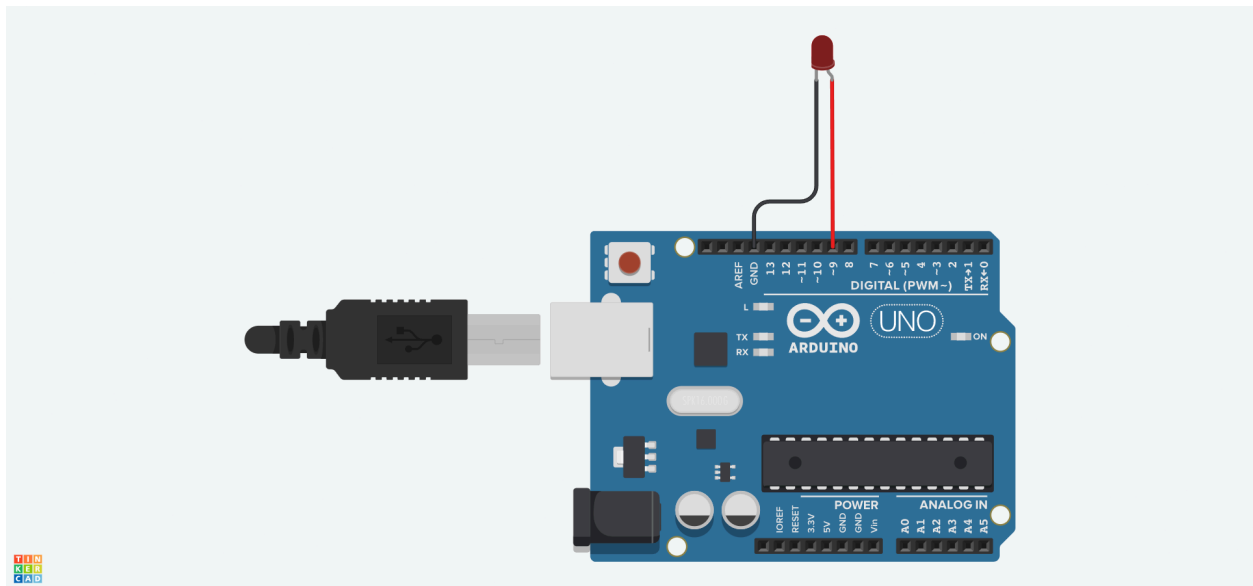
8. Write a program to show how to fade an LED on pin 9 using the analog Write() function.

```
// Define the LED pin
const int ledPin = 9;

void setup() {
  // Set the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Fade the LED from off to full brightness
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness);
    delay(10);
  }

  // Fade the LED from full brightness to off
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
}
```



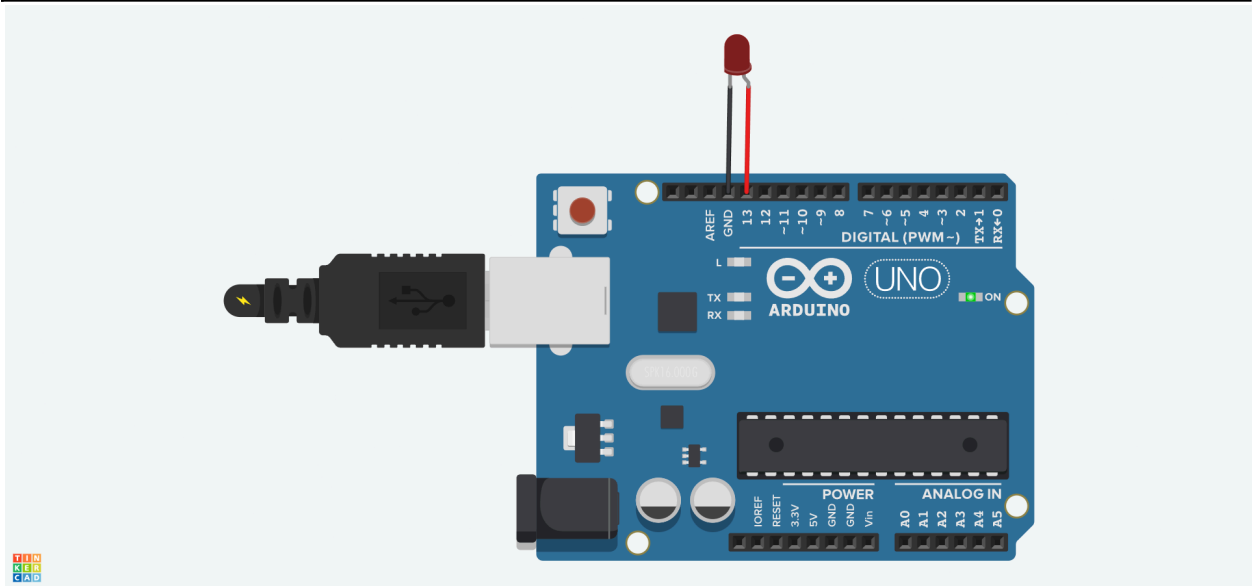
9. Write a program to control LED using loop.

```
// Define the LED pin
const int ledPin = 13;

void setup() {
  // Set the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LED on
  digitalWrite(ledPin, HIGH);
  delay(1000); // Wait for 1 second

  // Turn the LED off
  digitalWrite(ledPin, LOW);
  delay(1000); // Wait for 1 second
}
```



10. Write a program to control LED using Serial Number

```
const int ledPin = 13;
String serialInput;

void setup() {

  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);

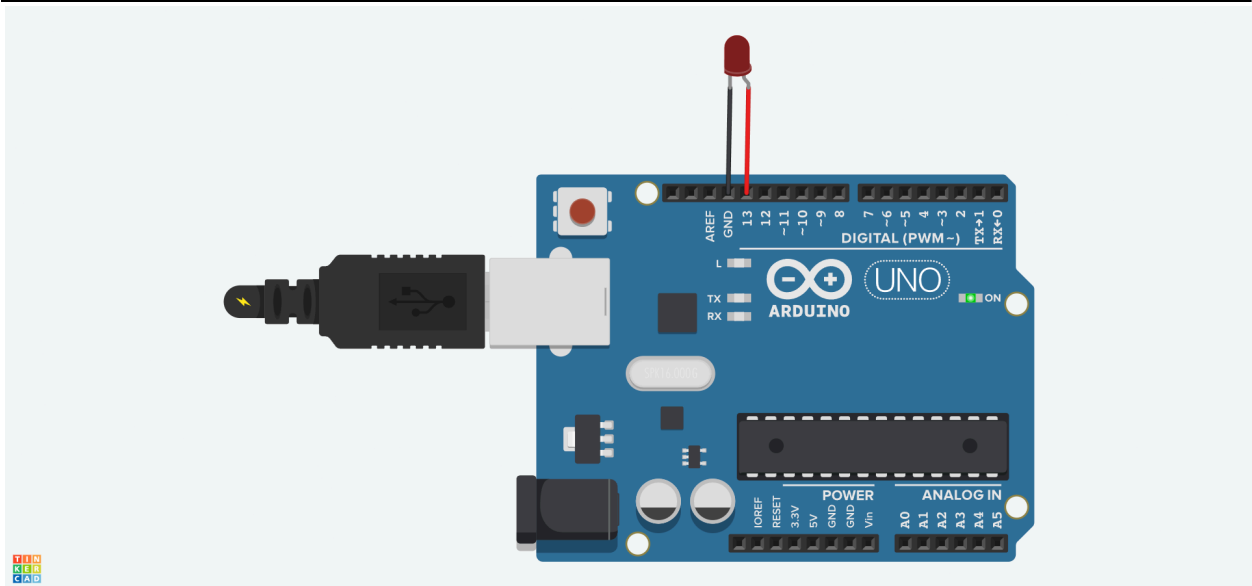
  Serial.println("Enter 'on' to turn the LED on, or 'off' to turn it off.");
}

void loop() {

  if (Serial.available() > 0) {

    serialInput = Serial.readStringUntil('\n');

    if (serialInput == "on") {
      digitalWrite(ledPin, HIGH);
      Serial.println("LED turned on.");
    } else if (serialInput == "off") {
      digitalWrite(ledPin, LOW);
      Serial.println("LED turned off.");
    } else {
      Serial.println("Invalid input. Enter 'on' or 'off'.");
    }
  }
}
```



11. Program to measure temperature and humidity

```
#include <DHT.h> // Include the DHT sensor library
#define DHTPIN 2   // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 11 sensor

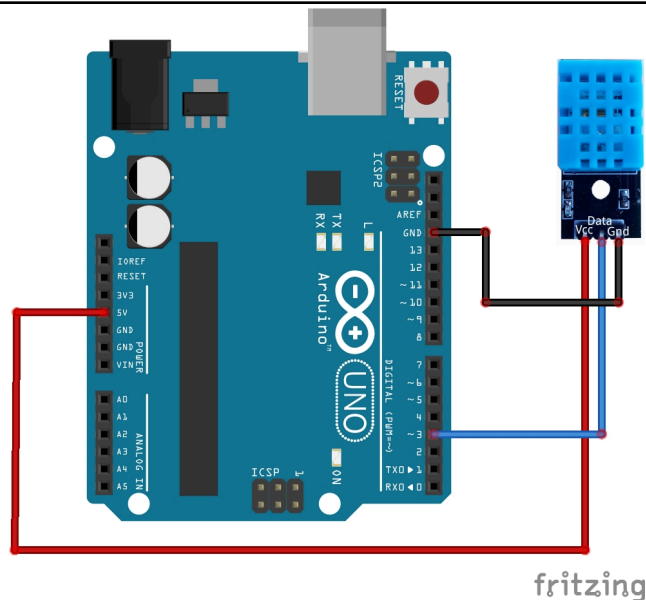
DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  // Read temperature and humidity
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Check if any reads failed
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");
  }

  delay(2000); // Wait 2 seconds before next reading
}
```



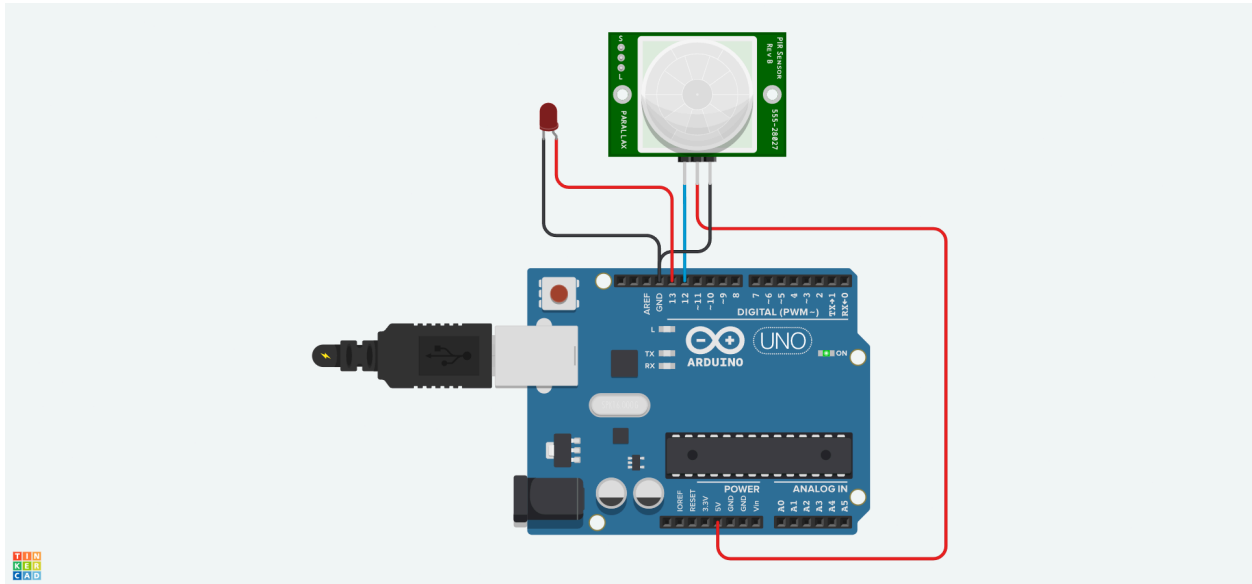
12. Program for motion detector or PIR sensor using Arduino.

```
int pirPin = 2;    // PIR sensor pin
int ledPin = 13;   // LED pin

void setup() {
  pinMode(pirPin, INPUT);    // Set PIR pin as input
  pinMode(ledPin, OUTPUT);    // Set LED pin as output
  Serial.begin(9600);        // Initialize serial communication
}

void loop() {
  int pirValue = digitalRead(pirPin); // Read PIR sensor value

  if (pirValue == HIGH) {          // If motion is detected
    digitalWrite(ledPin, HIGH);    // Turn on the LED
    Serial.println("Motion detected!");
  } else {
    digitalWrite(ledPin, LOW);     // Turn off the LED
  }
}
```



13 Program to create traffic light simulator for pedestrians.

```
int red = 10;
int orange = 9;
int green = 8;
int button = 7;
int readButton = 0;

void setup()
{
  pinMode(red, OUTPUT);
  pinMode(orange, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(button, INPUT);

  digitalWrite(red, LOW);
  digitalWrite(orange, LOW);
  digitalWrite(green, LOW);

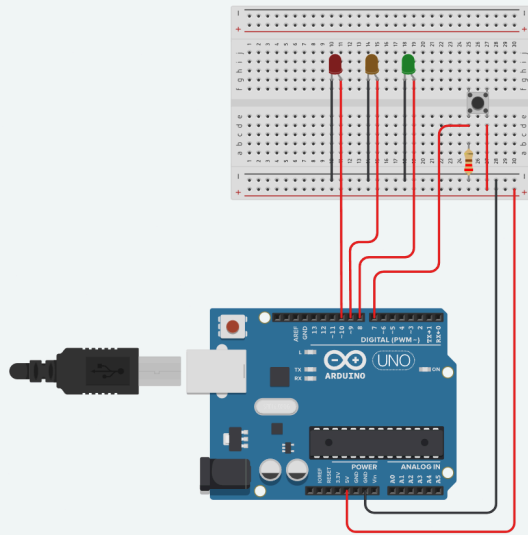
  Serial.begin(9600);
}

void loop()
{
  readButton = digitalRead(button);
  if (readButton == LOW){
    digitalWrite(red, LOW);
    digitalWrite(orange, LOW);
    digitalWrite(green, HIGH);
  }
  else if(readButton == HIGH){
    digitalWrite(red, LOW);
    digitalWrite(orange, HIGH);
    digitalWrite(green, LOW);
    Serial.println("PEDESTRIAN WAITING");

    delay(1000);

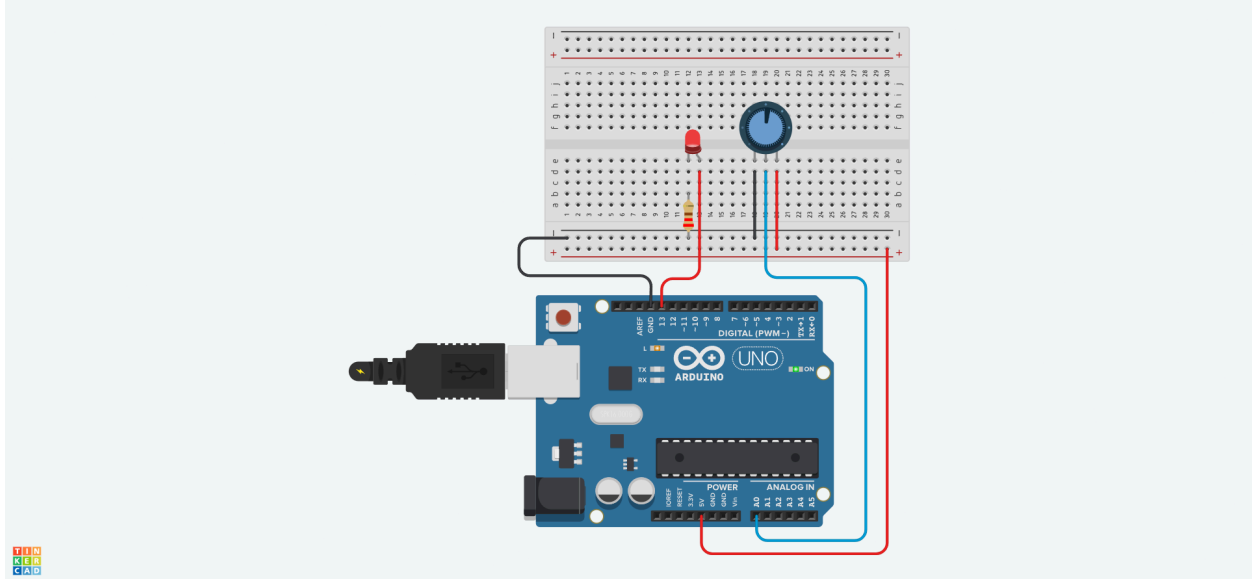
    digitalWrite(red, HIGH);
    digitalWrite(orange, LOW);
    digitalWrite(green, LOW);
    Serial.println("PEDESTRIAN CROSSING");

    delay(4000);
  }
}
```



14 Program to create Dimmable LED using potentiometer.

```
int ledPin = 13;  
int potPin = A0;  
  
void setup() {  
  pinMode(ledPin, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  int potValue = analogRead(potPin);  
  int brightness = map(potValue, 0, 1023, 0, 255);  
  
  analogWrite(ledPin, brightness);  
  Serial.print("Brightness: ");  
  Serial.println(brightness);  
  delay(10);  
}
```



15 Program to measure speed of sound using ultrasonic sensor

```
const int trigPin = 13; // Trigger pin
const int echoPin = 12; // Echo pin
long duration;
float distanceCm;

void setup() {
  Serial.begin(9600); // Initialize serial communication
  pinMode(trigPin, OUTPUT); // Set trigPin as output
  pinMode(echoPin, INPUT); // Set echoPin as input
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance in centimeters
  distanceCm = duration * 0.034 / 2;

  // Calculate the speed of sound
  float speedOfSound = distanceCm * 2 / (duration / 1000000.0);

  // Print the distance and speed of sound
  Serial.print("Distance: ");
  Serial.print(distanceCm);
  Serial.print(" cm, Speed of Sound: ");
  Serial.print(speedOfSound);
  Serial.println(" m/s");

  delay(1000); // Wait for 1 second before repeating the loop
}
```

