SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES,

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
CHITTOOR-517127, A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

III-I SEMESTER

Tinkering Lab Manual

TINKERING LAB

III B. Tech – I semester **L T P**

Subject code: 23ESC352L **2 0 0 1**

The aim of tinkering lab for engineering students is to provide a hands-on learning

environment where students can explore, experiment, and innovate by building and testing prototypes. These labs are designed to demonstrate practical skills that complement theoretical knowledge.

	Course objectives: The objectives of the course are to
1	**Encourage Innovation and Creativity**
2	**Provide Hands-on Learning and Impart Skill Development**
3	**Foster Collaboration and Teamwork**
4	**Enable Interdisciplinary Learning, Prepare for Industry and Entrepreneurship**
5	**Impart Problem-Solving mind-set**

List of experiments:

1. Model on parallel and series circuits using virtual breadboard in wokwi
2. Model on Temperature & Humidity Monitoring System (DHT22 + LCD)
3. Model on Water Level Detection and Alert System (Ultrasonic + Buzzer + LEDs)
4. Model on Automatic Plant Watering System (Potentiometer as Soil Sensor)
5. Model on Fire Detection and Alarm System (LDR/Flame Sensor + Buzzer)
6. Model on Automatic Street Light using LDR
7. Model on Smart Railway Gate (Ultrasonic + Servo)
8. Model on Weather Station Prototype (DHT22 + LCD + LDR)
9. Model on Digital Dice Using Push Button and LEDs
10. Model on Smart Dustbin using Ultrasonic Sensor and Servo Motor

INDEX

TINKERING LAB				
EX.NO.	**DATE**	**MODEL NAME**	**PAGE NO.**	**SIGN**
1.		Introduction to wokwi Arduino uno		
2.		Model on parallel and series circuits using virtual breadboard in Wokwi		
3.		Make a model on Temperature & Humidity Monitoring System (DHT22 + LCD)		
4.		Make a model on Water Level Detection and Alert System (Ultrasonic + Buzzer + LEDs)		
5.		Make a model on Automatic Plant Watering System (Potentiometer as Soil Sensor)		
6.		Make a Model on Fire Detection and Alarm System (LDR/Flame Sensor + Buzzer)		
7.		Make a model onAutomatic Street Light using LDR		
8.		Make a model on Smart Railway Gate (Ultrasonic + Servo)		
9.		Make a model on Weather Station Prototype (DHT22 + LCD + LDR)		
10.		Make a model on Digital Dice Using Push Button and LEDs		
11.		Make a model on Smart Dustbin using Ultrasonic Sensor and Servo Motor		

Signature of the faculty in-charge with date

SITAMS	Tinkering Lab	

INTERNAL MARKS-INDEX SHEET

S.NO	Experiment Name	Knowledge Gained	Analysis, Design /Technique	Ability to do experimentand following of ethical	Result & Conclusion	VIVA VOCE (Communication, Life Longlearning)	TOTAL	SIGNATURE OF THE FACULTY
		3	3	3	2	4	15 M	
1.	Introduction to Wokwi Arduino Uno							
2.	Model on parallel and series circuits using virtual breadboard in Wokwi							
3.	Make a model on Temperature & Humidity Monitoring System (DHT22 + LCD)							
4.	Make a model on Water Level Detection and Alert System (Ultrasonic + Buzzer + LEDs)							
5.	Make a model on Automatic Plant Watering System (Potentiometer as Soil Sensor)							
6.	Make a model on Fire Detection and Alarm System (LDR/Flame Sensor + Buzzer)							
7.	Make a model on Automatic Street Light using LDR							

8.	Make a model on Smart Railway Gate (Ultrasonic + Servo)							
9.	Make a model on Weather Station Prototype (DHT22 + LCD + LDR)							
10.	Make a model on Digital Dice Using Push Button and LEDs							
11.	Make a model on Smart Dustbin using Ultrasonic Sensor and Servo Motor							

Signature of the faculty in-charge with date

CO ATTAINMENT LEVELS

S.NO	Model Name	Knowledge	Analysis	Design	Complex Analysis & Conclusion	Use of Modern Tools	Communication ability	Ethics	Individual / Team Work	Life Long Learning
		CO1	CO2	CO3	CO4	CO5	CO6	CO7	CO8	CO8
1.	Introduction to wokwi									
2.	Model on parallel and series circuits using virtual breadboard for Wokwi application.									
3.	Make a model onTemperature & Humidity Monitoring System (DHT22 + LCD)									
4.	Make a model on Water Level Detection and Alert System (Ultrasonic + Buzzer + LEDs)									
5.	Make a model on Automatic Plant Watering System (Potentiometer as Soil Sensor)									
6.	Make a modelFire Detection and Alarm System (LDR/Flame Sensor + Buzzer)									
7.	Make a model on Automatic Street Light using LDR									

8.	Make a model on Smart Railway Gate (Ultrasonic + Servo)									
9.	Make a model on Weather Station Prototype (DHT11 + LCD + LDR)									
10.	Make a model on Digital Dice Using Push Button and LEDs									
11.	Make a model on Smart Dustbin using Ultrasonic Sensor and Servo Motor									
Average										

Signature of the faculty in-charge with date

Introduction to WOKWI ARDUINO UNO

1. Introduction

The **Wokwi Arduino Uno** is an online simulation platform that allows users to create and test **Arduino-based electronic projects** virtually.
It provides a complete circuit design environment where components such as sensors, LEDs, buzzers, displays, and motors can be connected and programmed.
This platform enables learners to understand both **hardware functioning** and **Arduino coding** before moving to real hardware, promoting efficient and safe experimentation.

2. Uses of Wokwi Arduino Uno

The Wokwi platform serves several important purposes in electronics learning and project development:

a. Circuit Simulation:
Allows users to design and simulate complex circuits in real time without any physical setup.

b. Coding and Debugging:
Enables writing, compiling, and testing of Arduino code, helping users debug and refine their logic.

c. Educational Tool:
Ideal for students, hobbyists, and researchers to learn **embedded systems, microcontrollers, and IoT applications** effectively.

d. Concept Visualization:
Helps visualize the interaction between hardware components and the uploaded program instantly.

3. Advantages of Using Wokwi Arduino Uno

a. Accessibility:
Being a browser-based tool, Wokwi requires no installation — it can be accessed anytime from any device.

b. Cost Efficiency:
Eliminates the need for physical components, reducing the overall cost of practical learning.

c. Safe Testing:
Students can test their connections virtually, preventing damage to hardware due to incorrect wiring or overvoltage.

d. Real-Time Output:
Displays accurate sensor readings and component responses just like real circuits.

e. Automatic Library Installation:
All required libraries are installed automatically once the code is uploaded and approved.

f. Error Correction:
Makes it easy to identify and fix both circuit and programming errors before using real hardware.

g. Collaboration:
Projects can be easily shared with teachers or classmates for review and group learning.

---

4. Reason for Inclusion as the Means of Experimentation

The **Wokwi Arduino Uno** platform is included as the medium for conducting experiments because it:

a. Provides a **safe, controlled, and realistic** simulation of Arduino hardware.
b. Encourages **hands-on learning** and experimentation without requiring physical kits.
c. Supports **multiple electronic components** commonly used in real-world projects.
d. Helps students gain **practical skills** in programming, circuit design, and debugging.
e. Acts as a bridge between **theoretical understanding** and **practical application**, making learning more interactive and engaging.

Conclusion:

The **Wokwi Arduino Uno** is an online simulator for designing and testing Arduino circuits.
It provides a cost-free, realistic environment to learn coding and circuit building.
Thus, it is an ideal platform for experimentation and understanding electronics easily.

Experiment 1: Model on Series and Parallel Circuits using virtual Breadboard in Wokwi

Aim:

To design a model and demonstrate series and parallel LED circuits using Arduino Uno on Wokwi.

Description:

This model demonstrates how current flows differently in series and parallel connections. In series, both LEDs share the same current, so brightness reduces. In parallel, each LED gets full voltage, so both glows brightly. Wokwi allows easy testing by connecting LEDs, resistors, and a power source. No library installation is needed; simple wiring and the LED blink code are enough.

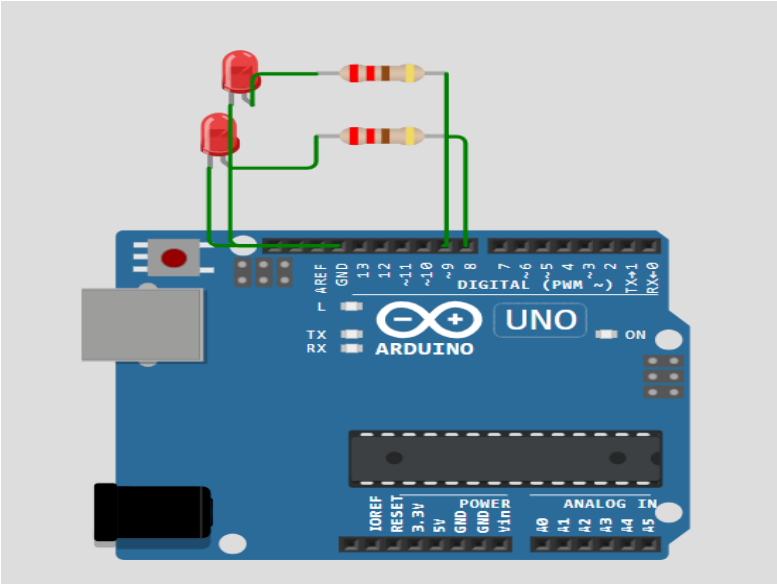
Components:

- Arduino UNO
- 2 LEDs
- 1 × 220Ω resistor
- Breadboard + jumper wires

Connections:

1. Arduino **Pin 8 → resistor → LED1 (anode)**
2. **LED1 cathode → LED2 anode** (join them directly on breadboard)
3. **LED2 cathode → GND**

Diagram for series:



PROGRAM:

Arduino Code (Series LEDs)

```
int led1 = 8;

int led2 = 9;


void setup() {

  pinMode(led1, OUTPUT);

  pinMode(led2, OUTPUT);

}
```

```
void loop() {
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  delay(1000);
}
```

Parallel Circuit (Home Lighting Example)

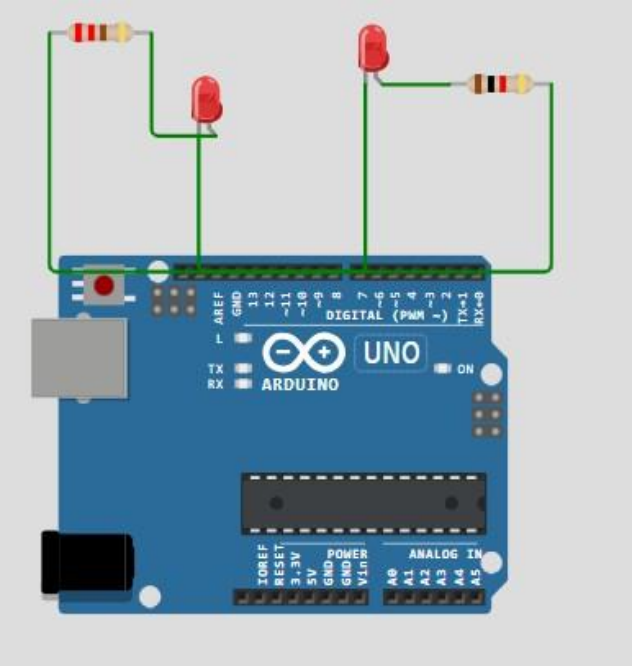
Components:

- Arduino UNO
- 2 LEDs
- 2 × 220Ω resistors
- Breadboard + jumper wires

Connections:

1. Arduino Pin 8 → resistor → LED1 anode
2. LED1 cathode → GND
3. Arduino Pin 9 → resistor → LED2 anode
4. LED2 cathode → GND

Diagram for Parallel:



Arduino Code (Parallel LEDs)

```
int led1 = 8;

int led2 = 9;


void setup() {

  pinMode(led1, OUTPUT);

  pinMode(led2, OUTPUT);

}


void loop() {

  digitalWrite(led1, HIGH);
```

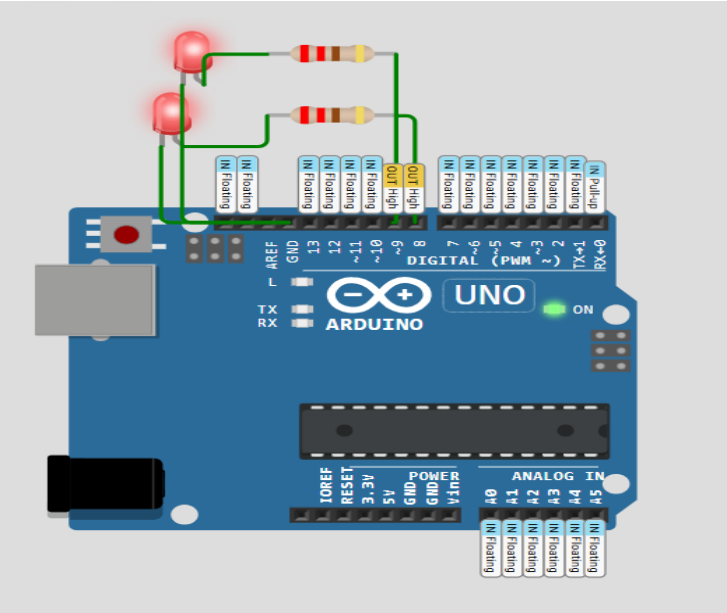
```
  delay(500);

  digitalWrite(led1, LOW);


  digitalWrite(led2, HIGH);

  delay(500);

  digitalWrite(led2, LOW);

}
```

Output:

Series Circuit (Output)

- Both LEDs glow together but dimmer because current is shared.
- If one LED fails → the other also turns OFF.

Output diagram

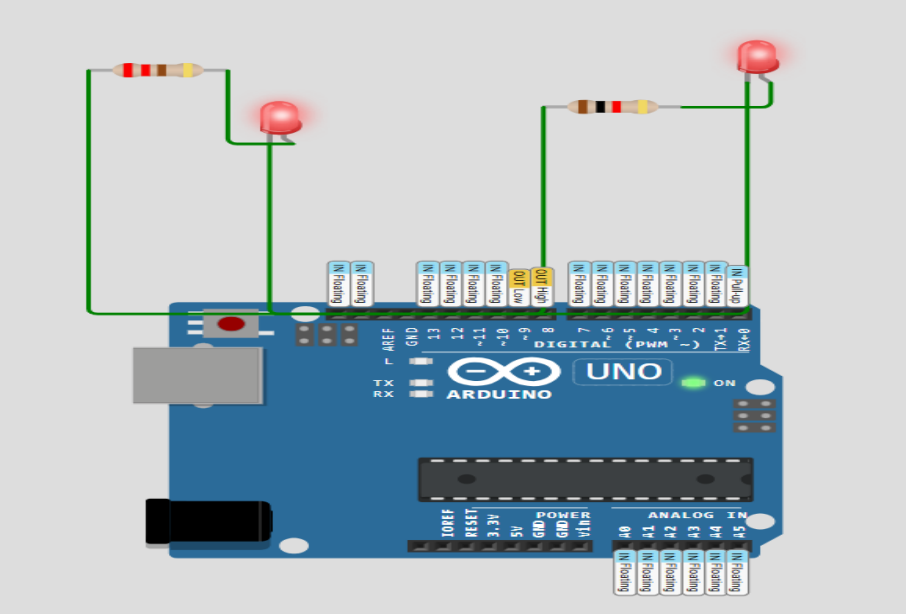


The LED’s glow at the same interval of time and blinks together.

Parallel Circuit (Output)

- Both LEDs glow bright and independent.
- If one LED fails → the other still glows.

Diagram



The LED’s glow alternatively.

Result:

Thus, the model to implement series and parallel LED connections using a virtual breadboard in Wokwi was successfully executed, and the behavior of current and voltage in both circuits was verified.

Experiment 2: Model on Temperature & Humidity Monitoring System (DHT22 + LCD)

Aim:

To design a model and monitor temperature and humidity using DHT22 and display the values on LCD.

Description

The DHT22 sensor measures temperature and humidity, and the LCD displays the readings in real time. Once the code and connections are uploaded, Wokwi shows live sensor readings on the virtual LCD. It requires the DHT and *LiquidCrystal* libraries, which Wokwi installs automatically when the code runs.

Components:

Arduino Uno, DHT22 sensor, 16x2 LCD, Potentiometer, Jumper wires, Breadboard

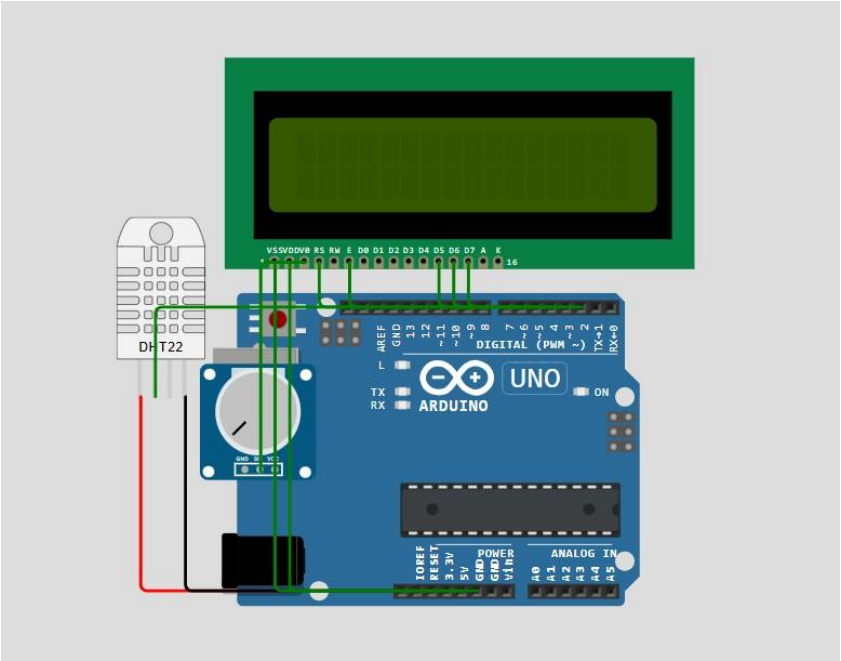
Connections:

• DHT22 VCC → 5V, GND → GND, DATA → pin 2
• LCD RS → 7, EN → 6, D4 → 5, D5 → 4, D6 → 3, D7 → 8
• Potentiometer middle pin → LCD V0 (contrast)
• LCD VCC → 5V, GND → GND

Procedure:

1. Connect all components as per circuit.
2. Upload the code.
3. Observe temperature and humidity on LCD.

Diagram:



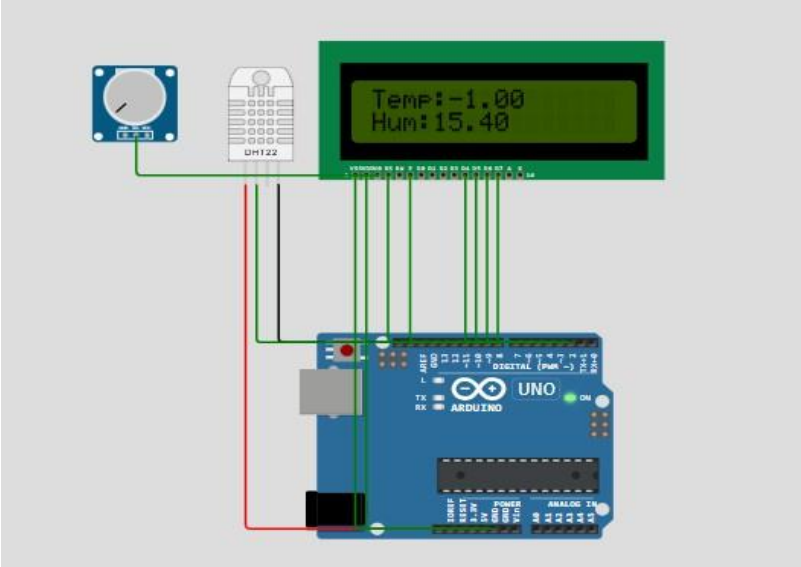
Program:

```
#include <DHT.h>
#include <LiquidCrystal.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(7,6,5,4,3,8);
void setup(){
  lcd.begin(16,2);
  dht.begin();
}
void loop(){
  float h = dht.readHumidity();
  float t = dht.readTemperature();
```

```
  lcd.setCursor(0,0);
  lcd.print("Temp:"); lcd.print(t);
  lcd.setCursor(0,1);
  lcd.print("Hum:"); lcd.print(h);
  delay(2000);
}
```

Output:

The LCD displays real-time readings such as “Temp: -1.00°C” and “Hum: 15.40%”. When temperature or humidity changes, the values update automatically.



Result:

Thus, the model to implement temperature and humidity monitoring using DHT22 and LCD was successfully executed, and the environmental readings were displayed accurately on the LCD.

Experiment 3: Model on Water Level Detection and Alert System

Aim:

To design a model to detect and alert different water levels using Ultrasonic sensor and LEDs.

Description

The ultrasonic sensor measures the distance between the sensor and the water level. Based on the distance, LEDs indicate the water level, and the buzzer sounds when it's too high or low. Wokwi simulates real-time distance changes using the ultrasonic sensor's object slider. No extra library installation is needed.

Components:

Arduino Uno, Ultrasonic Sensor (HC-SR04), 3 LEDs, Buzzer, Jumper wires

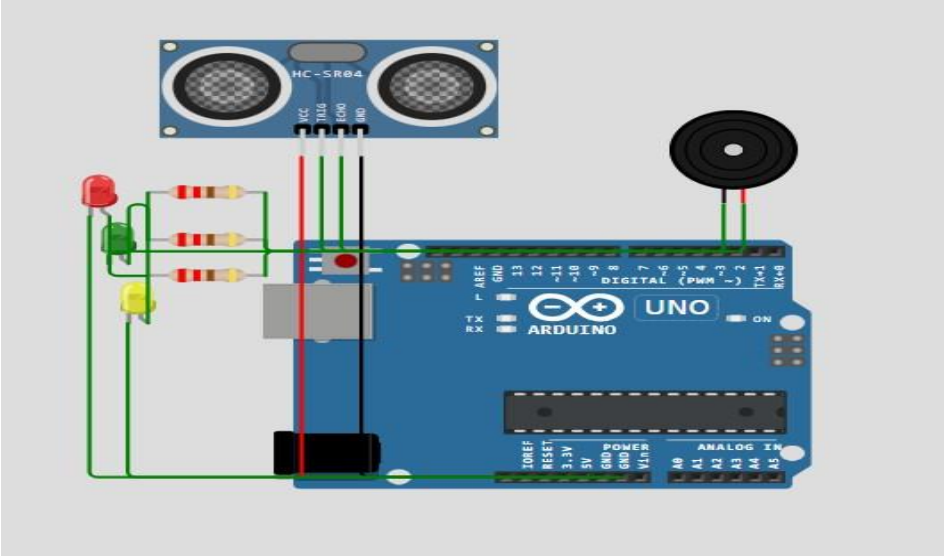
Connections:

• VCC → 5V, GND → GND
• Trig → pin 9, Echo → pin 10
• LEDs connected to pins 3, 4, 5 via resistors
• Buzzer to pin 6

Procedure:

1. Connect ultrasonic and LEDs.
2. Upload code and open serial monitor.
3. Observe LEDs and buzzer indicating different water levels.

Diagram:



Program:

```
const int trig = 9, echo = 10;
long duration; int distance;
void setup(){
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  for(int i=3;i<=6;i++) pinMode(i, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  digitalWrite(trig, LOW); delayMicroseconds(2);
  digitalWrite(trig, HIGH); delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;
  Serial.println(distance);
  if(distance < 10){ digitalWrite(3,HIGH); digitalWrite(6,HIGH);}
```

```
  else if(distance < 20){ digitalWrite(4,HIGH);}
  else digitalWrite(5,HIGH);
  delay(500);
}
```

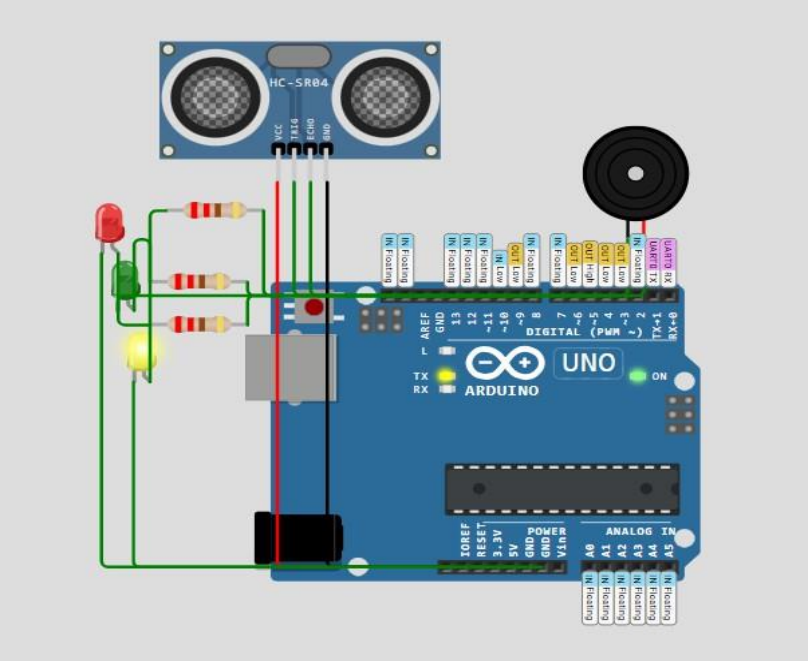
Output:

LEDs light up at various distances, buzzer activates when level is high.

Obtained reading is: 397

When the object (water level) is near, all LEDs glow and the buzzer sounds.
As the object moves farther, LEDs turn OFF sequentially and the buzzer stops.

Output Diagram:



Result:

Thus, to design a model to detect and alert different water levels using Ultrasonic sensor and LEDs was successfully executed.

Experiment 4: Model on Automatic Plant Watering System

Aim:

To design a model to automatically water a plant when the soil becomes dry.

Description:

Here, a potentiometer is used to simulate soil moisture. When the soil becomes “dry” (low analog value), a virtual motor (LED) or pump turns ON automatically. Wokwi lets you adjust the potentiometer to see the system response. This experiment does not need any additional libraries.

Components:

Arduino Uno, Soil Moisture Sensor (use potentiometer), LED, Buzzer, Breadboard, Jumper wires

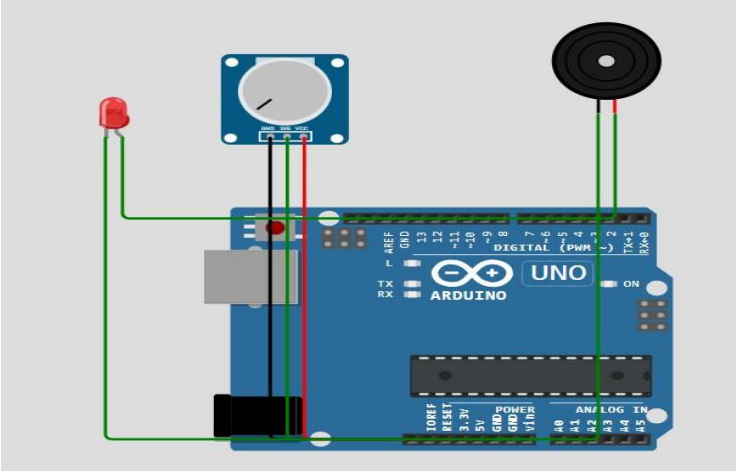
Connections:

• Potentiometer middle pin → A0
• LED → pin 8, Buzzer → pin 9
• VCC → 5V, GND → GND

Procedure:

1. Use potentiometer as soil sensor input.
2. Upload code and vary the potentiometer.
3. Observe LED and buzzer indicate dry soil.

Diagram:



Program:

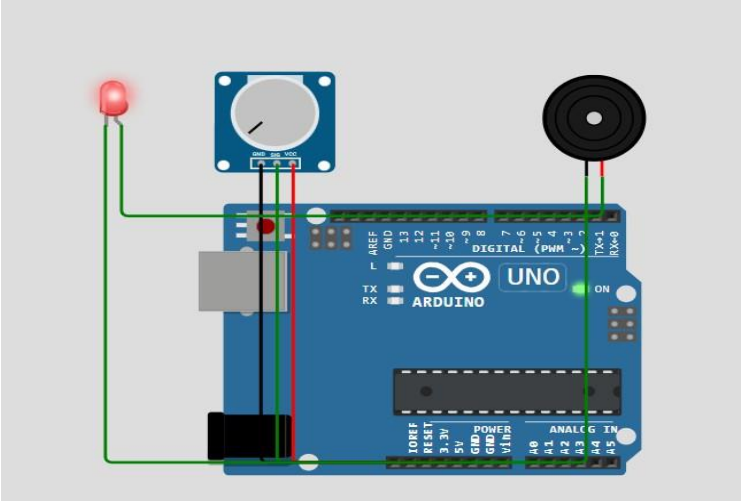
```
int sensor = A0;
int led = 8;
int buzzer = 9;
void setup(){
  pinMode(led, OUTPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  int value = analogRead(sensor);
  Serial.println(value);
  if(value < 400){
    digitalWrite(led, HIGH);
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(led, LOW);
    digitalWrite(buzzer, LOW);
  }
```

```
  delay(500);
}
```

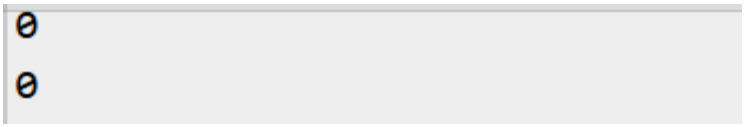
Output:

When the potentiometer is turned to a low value (dry soil), the pump (LED) turns ON. When turned to a high value (wet soil), the pump (LED) turns OFF.

Diagram:



Obtained readings:



Result:

Thus, to design a model to implement an automatic plant watering system using a potentiometer as a soil sensor was successfully executed, and the pump responded correctly to soil moisture levels.

Experiment 5: Model on Fire Detection and Alarm System

Aim:

To design a model to detect fire using LDR sensor and alert through buzzer and LED.

Description:

An LDR is used to sense light intensity, simulating a fire or flame detection. When light is high (flame detected), the buzzer activates. Wokwi allows live adjustment of light levels to simulate this behavior. No extra libraries are needed; analog and digital readings handle the function.

Components:

Arduino Uno, LDR sensor, Buzzer, LED, Resistor (10kΩ), Jumper wires

Connections:

• LDR + resistor divider → A0

(+5V ----[LDR]----(A0)----[Resistor (10kΩ)]---- GND)

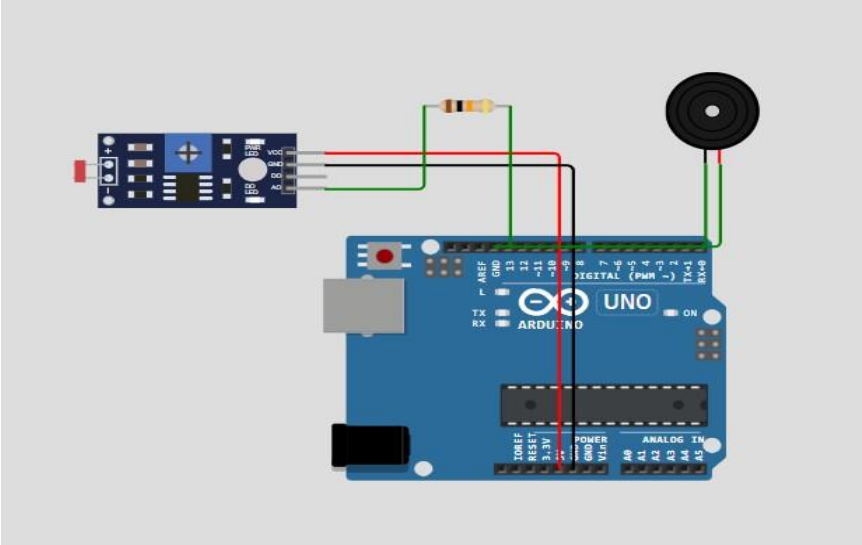
• LED → pin 8, Buzzer → pin 9

• VCC → 5V, GND → GND

Procedure:

1. Connect LDR and LED.
2. Upload code and simulate fire by bright light.
3. Observe buzzer and LED respond to brightness.

Diagram:



Program:

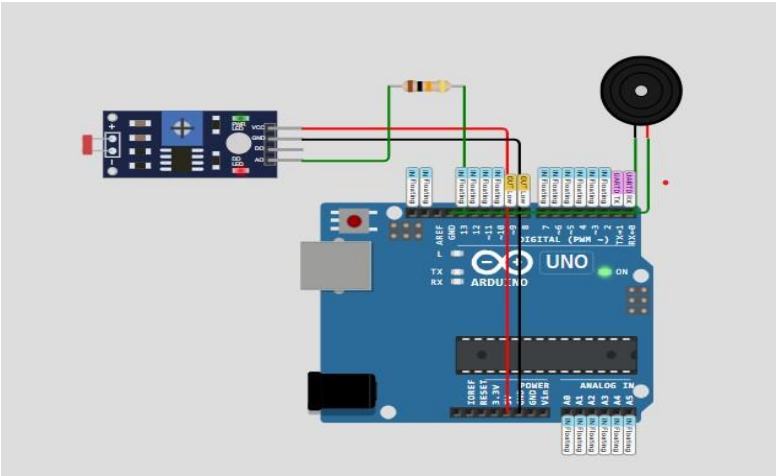
```
int ldr = A0;
int led = 8;
int buzzer = 9;
void setup(){
  pinMode(led, OUTPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  int value = analogRead(ldr);
  Serial.println(value);
  if(value > 700){
  digitalWrite(led, HIGH);
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(led, LOW);
    digitalWrite(buzzer, LOW);
```

```
  }
  delay(500);
}
```

Output:

When the LDR detects bright light (fire), the buzzer activates.

In normal light or darkness, the buzzer remains OFF.



Obtained output readings:

```
372
367
381
364
```

Result:

Thus, to design a model to implement a fire detection and alarm system using an LDR sensor and buzzer was successfully executed, and the alarm triggered correctly under simulated fire conditions.

Experiment 6: Model on Automatic Street Light using LDR

Aim:

To design a model to turn on street light automatically in dark using LDR sensor.

Description:

The LDR measures ambient light. When light levels drop (nighttime), the LED turns ON, simulating a street light. Increasing light turns it OFF again. The LDR can be adjusted in Wokwi to show this behavior. This experiment does not require any libraries.

Components:

Arduino Uno, LDR sensor, LED, Resistor (10kΩ), Jumper wires

Connections:

• LDR + resistor divider → A0

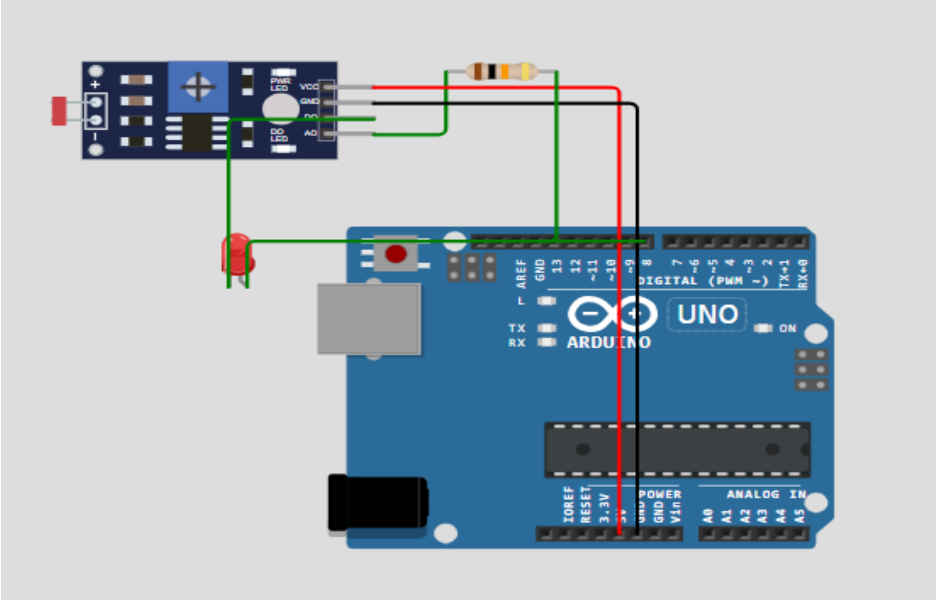
• LED → pin 8

• VCC → 5V, GND → GND

Procedure:

1. Connect LDR and LED.
2. Upload the program.
3. Observe LED turn ON in darkness and OFF in light.

Diagram:



Program:

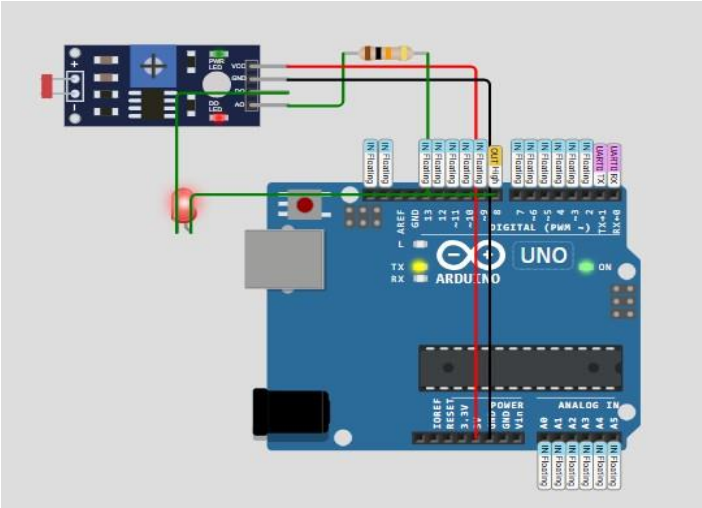
```
int ldr = A0;
int led = 8;
void setup(){
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  int value = analogRead(ldr);
  Serial.println(value);
  if(value < 500) digitalWrite(led, HIGH);
  else digitalWrite(led, LOW);
  delay(500);
}
```

Output:

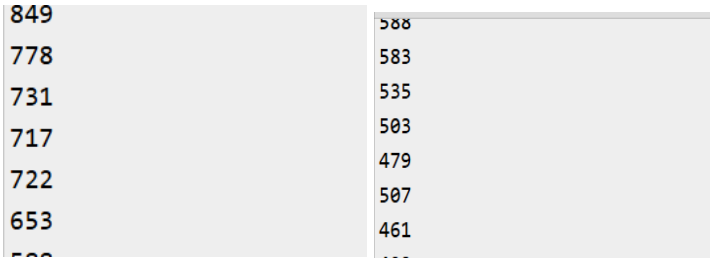
LED turns ON in dark and OFF in light. The LED is in off state at the beginning of the simulation after a few readings it starts to blink

When the LDR value is low (darkness), the LED glows automatically.
When the LDR value increases (daylight), the LED turns OFF.

Diagram:



Output Reading:



Result:

Thus, to design a model to implement an automatic street light system using an LDR sensor was successfully executed, and the LED operated accurately based on light intensity

Experiment 7: Model on Smart Railway Gate (Ultrasonic + Servo)

Aim:

To design a model to automatically open and close a railway gate using ultrasonic sensor and servo motor.

Description:

The ultrasonic sensor detects an approaching object (train). When the train comes close, the servo motor lowers the gate; when it leaves, the gate opens again. Wokwi animates the servo motor's movement as the object distance changes. It requires the **Servo** library, which installs automatically once the code is approved.

Components:

Arduino Uno, Ultrasonic Sensor, Servo Motor, Jumper wires

Connections:

• Ultrasonic VCC → 5V, GND → GND, Trig → 9, Echo → 10

• Servo signal → 6, VCC → 5V, GND → GND

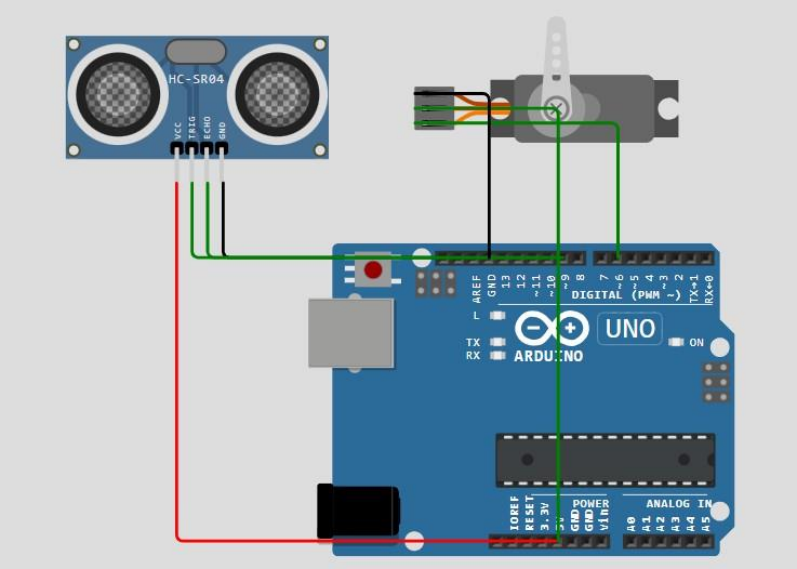
Procedure:

1. Connect components as shown.

2. Upload the code.

3. Observe servo gate movement when object is detected.

Diagram:



Program:

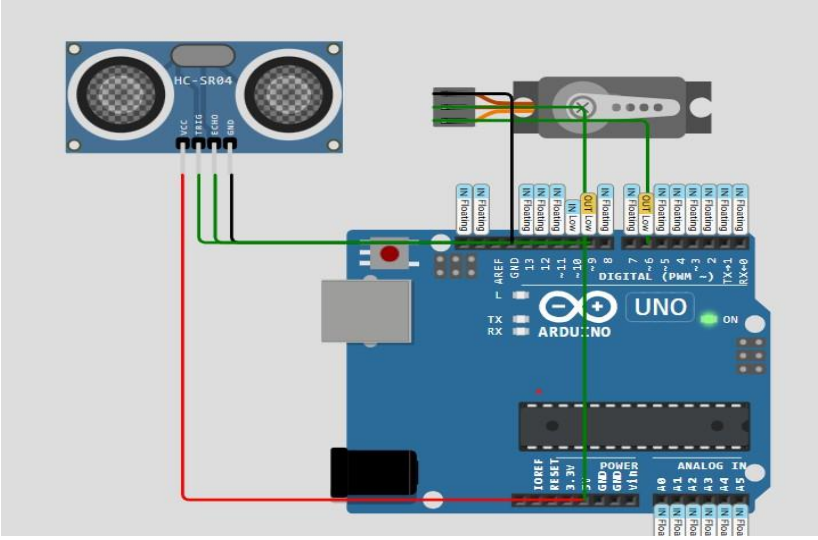
```
#include <Servo.h>
Servo gate;
int trig = 9, echo = 10;
void setup(){
gate.attach(6);
pinMode(trig, OUTPUT);
pinMode(echo, INPUT);
}
void loop(){
  digitalWrite(trig, LOW); delayMicroseconds(2);
```

```
  digitalWrite(trig, HIGH); delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long duration = pulseIn(echo, HIGH);
  int distance = duration * 0.034 / 2;
  if(distance < 15) gate.write(0);
  else gate.write(90);
  delay(500);
}
```

Output:

When the object is close to the ultrasonic sensor, the servo motor rotates to close the gate. When the object moves away, the servo returns to open the gate.

Output Diagram:



Result:

Thus, to design a model to implement an automatic smart railway gate using an ultrasonic sensor and servo motor was successfully executed, and the gate operated automatically based on object detection.

Experiment 8: Model on Weather Station Prototype

(DHT22 + LCD + LDR)

Aim:

To design a model to display environmental data like temperature, humidity, and light using DHT22 and LDR.

Description:

The ultrasonic sensor detects an approaching object (train). When the train comes close, the servo motor lowers the gate; when it leaves, the gate opens again. Wokwi animates the servo motor's movement as the object distance changes. It requires the **Servo** library, which installs automatically once the code is approved.

Components:

Arduino Uno, DHT22 sensor, LDR sensor, resistor(10kΩ),16x2 LCD, Potentiometer, Jumper wires

Connections:

DHT Sensor (DHT22):

• VCC → 5V

• GND → GND

• DATA → Pin 2

LDR with 10kΩ Resistor Divider:

• One side of LDR → 5V

• Other side of LDR → A0 and one end of 10kΩ resistor

• Other end of 10kΩ resistor → GND

16x2 LCD:

• RS → 7

• EN → 6

• D4 → 5

• D5 → 4

• D6 → 3

• D7 → 8

• VSS → GND

• VDD → 5V

• RW → GND

• V0 (contrast pin) → Middle pin of potentiometer

• Potentiometer side pins → 5V and GND

• A (LED+) → 5V

• K (LED–) → GND

Power Lines:

• All VCC pins → 5V

• All GND pins → GND

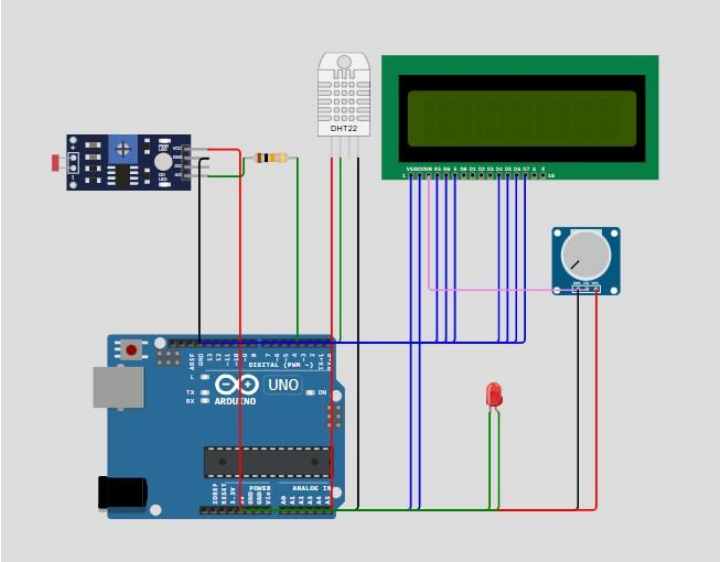
Procedure:

1. Connect sensors and LCD.

2. Upload the code.

3. Observe light, temperature, and humidity readings on LCD.

Diagram:



Program:

```
#include <DHT.h>
#include <LiquidCrystal.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(7,6,5,4,3,8);
int ldr = A0;
void setup(){
  lcd.begin(16,2);
  dht.begin();
}
void loop(){
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  int light = analogRead(ldr);
```

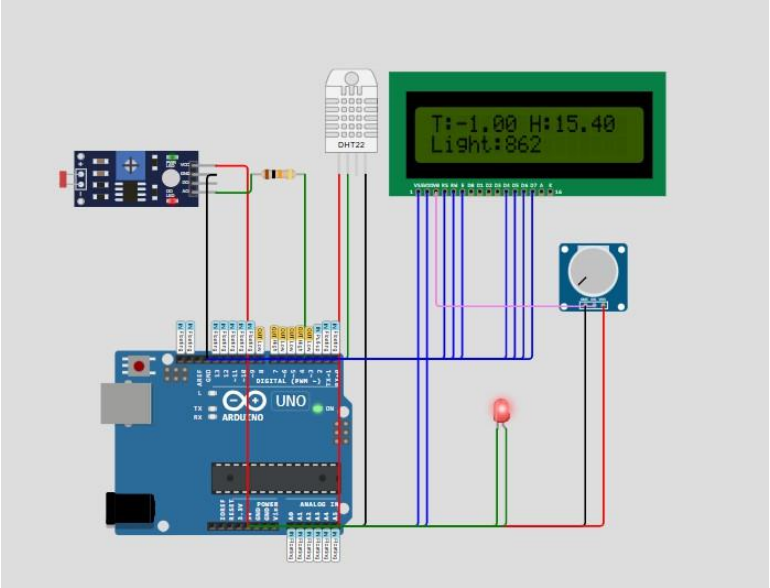
```
  lcd.setCursor(0,0);
  lcd.print("T:"); lcd.print(t);
  lcd.print(" H:"); lcd.print(h);
  lcd.setCursor(0,1);
  lcd.print("Light:"); lcd.print(light);
  delay(2000);
}
```

Output:

LCD shows temperature, humidity, and light values.

The LCD displays “Temp: 28°C Hum: 60%” and “Light: 512”.
As the light intensity or environment changes, values update on the LCD.

Output Diagram:



Result:

Thus, to design a model to implement a weather station prototype using DHT22, LDR, and LCD was successfully executed, and real-time temperature, humidity, and light readings were displayed correctly.

Experiment 9: Model on Digital Dice Using Push Button and LEDs

Aim:

To design a model to simulate a digital dice using Arduino, push button, and LEDs.

Description:

When the button is pressed, LEDs blink rapidly for a few seconds and then stop randomly, showing a number from 1 to 6. Each number corresponds to a unique LED pattern. Wokwi supports buttons and LEDs perfectly, allowing you to observe the random dice effect. No extra library installation is needed.

Components:

- Arduino Uno
- 6 LEDs
- 6 Resistors (220Ω)
- 1 Push Button
- Breadboard
- Jumper Wires

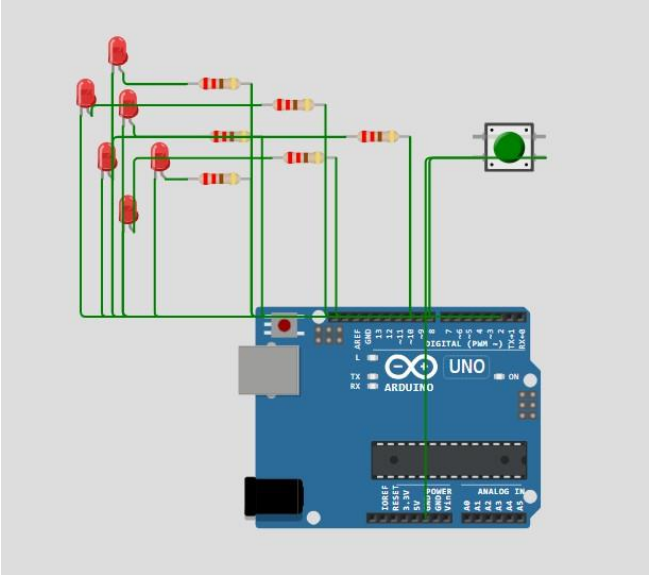
Connections:

- LEDs anodes → Pins 2, 3, 4, 5, 6, 7 (each through a resistor)
- LEDs cathodes → GND
- Push Button → One side to Pin 8, other side to GND
- VCC → 5V
- GND → GND

Procedure:

1. Connect all components as per the above circuit.
2. Upload the program into Wokwi IDE.
3. Press the button to roll the dice and observe random LED patterns.

Diagram:



Program:

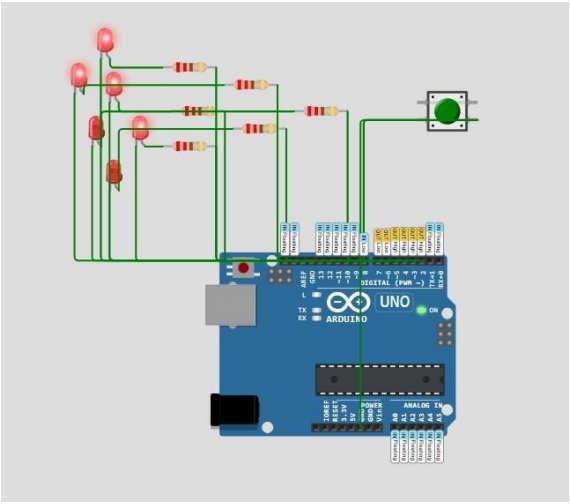
```
int leds[] = {2,3,4,5,6,7};

int button = 8;

void setup() {

  for(int i=0;i<6;i++) pinMode(leds[i], OUTPUT);

  pinMode(button, INPUT_PULLUP);

  randomSeed(analogRead(0));

}

void loop() {

  if(digitalRead(button)==LOW){

  for(int j=0;j<10;j++){
```

```
    int num = random(1,7);
    for(int i=0;i<6;i++) digitalWrite(leds[i], LOW);
    for(int i=0;i<num;i++) digitalWrite(leds[i], HIGH);
    delay(100);
   }
  }
}
```

Output:

When the button is pressed, LEDs blink rapidly and stop at a random pattern representing a dice number. Each run gives a new random output.

Output Diagram:



Result:

Thus, to design a model to implement a digital dice using Arduino Uno, push button, and LEDs was successfully executed, and the random dice simulation was displayed through LED patterns.

Experiment 10: Model on Smart Dustbin using Ultrasonic Sensor and Servo Motor

Aim:

To design a model on Smart Dustbin that automatically opens its lid when an object comes near using an Ultrasonic Sensor and a Servo Motor on Arduino Uno in Wokwi.

Description:

The ultrasonic sensor measures the distance of an approaching object. If the distance is less than 10 centimeters, the servo motor rotates to open the lid. When the object moves away, the servo returns to its original position to close the lid. This demonstrates the working of an automatic dustbin system. The online Wokwi application automatically installs any required libraries once the code is entered and connections are approved.

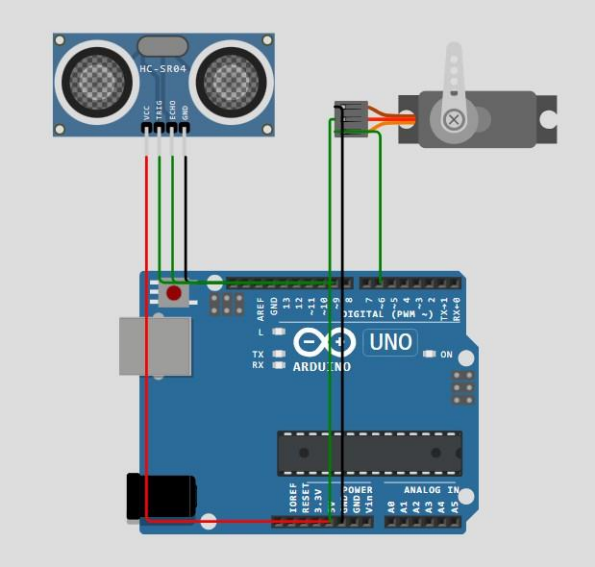
Components Required:

Arduino Uno
Ultrasonic Sensor (HC-SR04)
Servo Motor (SG90)
Jumper Wires
Breadboard (optional)

Connection Procedure:

1. Connect the VCC of the Ultrasonic Sensor to the 5V pin of Arduino Uno.
2. Connect the GND of the Ultrasonic Sensor to the GND of Arduino Uno.
3. Connect the Trig pin of the Ultrasonic Sensor to Digital Pin 9 of Arduino Uno.
4. Connect the Echo pin of the Ultrasonic Sensor to Digital Pin 10 of Arduino Uno.
5. Connect the Signal pin of the Servo Motor to Digital Pin 6 of Arduino Uno.
6. Connect the VCC and GND of the Servo Motor to 5V and GND of Arduino Uno respectively.

Diagram:



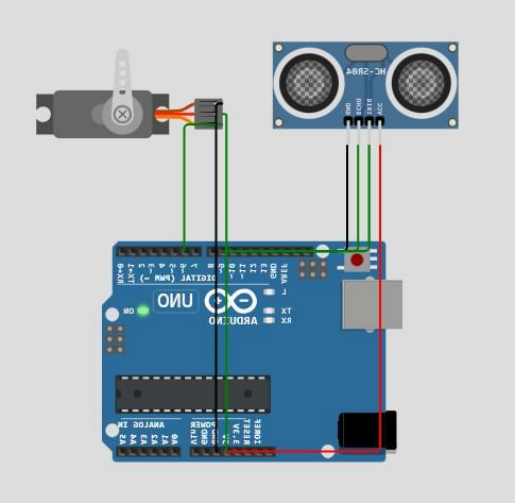
Program:

```
#include <Servo.h>

Servo servo;

int trig = 9;

int echo = 10;

long duration;

int distance;


void setup() {

  servo.attach(6);

  pinMode(trig, OUTPUT);
```

```
  pinMode(echo, INPUT);
  servo.write(0);
}

void loop() {
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;

  if (distance < 10) {
    servo.write(90);
  } else {
    servo.write(0);
  }
  delay(500); }
```

Output:

When an object such as a hand is brought close to the sensor within 10 centimeters, the servo motor rotates and the lid of the dustbin opens automatically. When the object is moved away, the servo motor rotates back to close the lid.



Result:

Thus, to design a model to implement the Smart Dustbin using Ultrasonic Sensor and Servo Motor was executed successfully in the Wokwi Arduino Uno simulator, and the lid opened automatically when an object approached within 10 centimeters.

RUBRICS FOR ARTIFICIAL INTELLIGENCE LAB

	Excellent(3)	Good(2)	Fair(1)
Conduct Experiments (CO1)	Student successfully completes the experiment, records the data, analyzes the experiment's main topics, and explains the experiment concisely and well.	Student successfully completes the experiment, records the data, and analyzes the experiment's main topics	Student successfully completes the experiment, records the data, and unable to analyzes.
Analysis and Synthesis (CO2)	Thorough analysis of circuit designed	Reasonable analysis of circuit designed	Improper analysis of circuit designed
Design (CO3)	Student understands what needs to be tested and designs an appropriate experiment, and explains the experiment concisely and well	Student understands what needs to be tested and designs an appropriate experiment.	Student understands what needs to be tested and does not design an appropriate experiment.
Complex Analysis & Conclusion (CO4)	Thorough comprehension through analysis/ synthesis	Reasonable comprehension through analysis/ synthesis	Improper comprehension through analysis/ synthesis
Lab safety (CO5)	Student will demonstrate good understanding and follow lab safety	Student will demonstrate good understanding of lab safety	Students demonstrate a little knowledge of lab safety.
Ability to work in teams (CO6)	Performance on teams is excellent with clear evidence of equal distribution of tasks and effort	Performance on teams is good with equal distribution of tasks and effort	Performance on teams is acceptable with one or more members carrying a larger amount of the effort
Report Writing (CO7)	Status report with clear and logical sequence of parameter using excellent language	Status report with logical sequence of parameter using understandable language	Status report not properly organized
Continuous learning (CO8)	Highly enthusiastic towards continuous learning	Interested in continuous learning	Inadequate interest in continuous learning