Experiment No.: 08

Experiment Name: Study and Observation of Arduino Uno Based rainSensor

Objective:

Theory:

Required Hardwares with Quantity:

Required Software:

Working Procedure:

Hardware Arrangement Diagram:

Sketch:

Result and Output:

Conclusion:

Basic Knowledge:

A rain sensor is a device used to detect the presence or absence of rain. It is commonly used in systems like automatic windshield wipers, irrigation systems, or weather monitoring. The sensor typically works by detecting the changes in environmental conditions caused by rain, such as the change in resistance or the absorption of light due to water droplets.

How a Resistive Rain Sensor Works:

A resistive rain sensor is commonly used in applications like automatic wipers for vehicles or simple rain detection systems. Here's how it works:

1. Design: The sensor consists of two conductive tracks or electrodes placed on a surface. These tracks are typically made of copper or another conductive material.
2. No Rain Condition: When there is no rain, the resistance between the two tracks remains high because there is no conductive medium (like water) between the tracks.
3. Rain Detection: When rain falls on the sensor, water bridges the gap between the conductive tracks. This reduces the resistance between the tracks, and the change in resistance is detected by the sensor's circuitry.
4. Output: The sensor typically outputs a digital signal (HIGH or LOW) depending on the resistance change, which can then be used by a microcontroller or other devices to take actions, such as turning on a pump, activating a warning light, or triggering an automatic wiper.

Code-1:

```
#define SENSOR_PIN A0  // Analog pin to which the rain sensor is connected

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

void loop() {
  int sensorValue = analogRead(SENSOR_PIN);  // Read the analog sensor value (0-1023)
  Serial.println(sensorValue);  // Print the sensor value to the Serial Monitor
  delay(500);  // Wait for half a second before taking another reading
}
```

□ **Dry Condition:**

- Place the sensor in a dry environment (e.g., indoors, in a place with no moisture).
- You should see a higher sensor value (closer to 1023) because the sensor's resistance is high and it doesn't conduct much electricity.

□ **Wet Condition:**

- Simulate rain by either spraying water on the sensor or exposing it to real rain.
- When the sensor detects rain, water bridges the sensor's conductive paths, reducing the resistance and allowing more current to flow, which will lower the sensor value.
- You should see a drop in the sensor value, possibly closer to 0.

Monitor and record the sensor readings in both conditions:

- In dry conditions, the value will generally be higher, such as 800 or higher.
- In wet conditions, the value will be lower, such as 300 or lower (but this can vary depending on the sensor).

Code-2:

```
#define SENSOR_PIN A0  // Analog pin connected to the rain sensor
#define LED_PIN 13      // Digital pin connected to the LED
#define THRESHOLD ____  // Set the threshold for detecting rain (can be adjusted)

void setup() {
  pinMode(LED_PIN, OUTPUT);      // Set LED pin as output
  Serial.begin(9600);            // Initialize serial communication
}

void loop() {
  int sensorValue = analogRead(SENSOR_PIN);  // Read the sensor value (0-1023)
  Serial.println(sensorValue);   // Print the sensor value to Serial Monitor

  // If the sensor value is below the threshold, it is raining
  if (sensorValue < THRESHOLD) {
    digitalWrite(LED_PIN, HIGH);  // Turn on the LED (indicating rain)
  } else {
    digitalWrite(LED_PIN, LOW);   // Turn off the LED (no rain)
  }

  delay(500);  // Delay for half a second before the next reading
}
```