Rain Detection Alarm

Introduction

The Internet of Things has revolutionized weather monitoring and automation, enabling smart solutions for everyday challenges. One such innovation is the rain detection alarm with automatic shelter closing, designed to provide protection against unexpected rainfall. This system is especially useful in open spaces such as balconies, parking lots, stadiums, and agricultural fields, where sudden rain can cause inconvenience or damage. This IoT-based system functions using a rain sensor, microcontroller such as Arduino, motorized shelter mechanism, and an alert system. When rain is detected, the sensor trans mits a signal to the microcontroller, which then activates the motorized mechanism to close the shelter. At the same time, an alarm or notification is triggered to inform the user of the weather condition. This automation ensures quick and efficient protection without re quiring manual intervention. The system offers multiple benefits, including automation, real-time monitoring, and remote access via mobile applications or cloud-based platforms. Users can monitor and control the system from anywhere, making it highly effective for smart homes, industries, and agricultural applications. By preventing rain-related damage, this technology also helps protect valuable assets and equipment. Future advancements in IoT can further enhance this system by integrating weather prediction data and AI-based decision-making. This would improve efficiency by enabling the system to take preventive actions based on upcoming weather forecasts. As technology continues to evolve, such smart solutions will play a crucial role in creating a safer, more convenient, and automated lifestyle.

Objectives

- Early Rain Detection
- Asset and Property Safety
- Energy-Efficient Operation

Requirements

➢ Hardware Requirements

- Rain sensor
- Arduino Uno
- DCmotors
- Buzzer
- Motor Driver

Functional Requirements

- Rain Detection System
- Automated Roof Closing Mechanism
- User Alerts and Notifications

Non-Functional Requirements

- Performance
- Maintainability
- Reliability
- Cost-Effectiveness

Code

```
 const int rainSensorPin = A0;

 const int b = 8;

 const int motorEnablePin = 9;

const int motorPin1 = 2;

 const int motorPin2 = 3;

const int threshold = 900;

 bool wiperActivated = false;

void beep(){

digitalWrite(b,HIGH);

delay(200) ;

digitalWrite(b,LOW);

delay(200) ;

digitalWrite(b,HIGH);

delay(200) ;

digitalWrite(b,LOW);

delay(200) ;

digitalWrite(b,HIGH);

delay(200) ;

digitalWrite(b,LOW);
```

```
delay(200) ;
}
void setup() {
 pinMode(motorEnablePin, OUTPUT);
pinMode(b, OUTPUT);
pinMode(motorPin1, OUTPUT);
pinMode(motorPin2, OUTPUT);
Serial.begin(9600);  }
 void loop() {
 int sensorValue = analogRead(rainSensorPin);
 Serial.print("Rain Sensor Value: ");
Serial.println(sensorValue);
if (sensorValue < threshold && !wiperActivated) {
 wiperActivated = true;
 beep();
digitalWrite(motorPin1, HIGH);
 digitalWrite(motorPin2, LOW);
analogWrite(motorEnablePin, 255);
 Serial.println("Rain detected! Wiper ON for 2 seconds.");
delay(3000);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
analogWrite(motorEnablePin, 0);
Serial.println("Wiper OFF.");  }
 if (sensorValue > threshold && wiperActivated) {
wiperActivated = false;
 beep();
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin1, LOW);
analogWrite(motorEnablePin, 255);
Serial.println("No Rain detected! Wiper OFF for 2 seconds.");
delay(4000);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
analogWrite(motorEnablePin, 0);
Serial.println("Wiper OFF.");  }
 delay(500);
 }
```