Automated Rain Protected Cloth Drying System

Rohan Rashid
Department of Computer Science and Engineering
Bangladesh University of Business and Technology (BUBT)
Dhaka, Bangladesh
Email: 21222203050@cse.bubt.edu.bd

Abstract: The Automated Rain Protected Cloth Drying System is an IoT-based solution aimed at automating the protection of clothes from rain. This system leverages sensors and actuators to detect environmental changes, such as rain and sunlight, and reacts accordingly to cover or uncover clothes. The system incorporates a NodeMCU microcontroller integrated with the Arduino IDE for processing data from sensors and controlling the motors that move the cover mechanism. Notifications are sent through IoT platforms, ensuring remote monitoring and control. The system enhances efficiency and convenience, reducing manual intervention in managing clothes drying, especially in urban environments.

Keywords—IoT, rain detection, cloth drying automation, NodeMCU, LDR sensor, rain sensor, servo motor, real-time monitoring.

Introduction

The challenges posed by sudden weather changes, particularly rain, necessitate an automated system for protecting drying clothes. The Automated Rain Detection Cloth Drying System utilizes IoT technologies to monitor environmental conditions and react in real-time by covering clothes when it detects rain. This system aims to enhance user convenience and protect clothes effectively without manual supervision.

Literature Review

Ishak et al. (2020) presents an automatic retractable cloth drying system controlled by an Arduino UNO, using rain and light sensors (LDR) to detect weather conditions. The system retracts the clothesline during rain and extends it when sunny, utilizing a power window motor to handle up to 5 kg of wet clothes. This space-saving, automated solution is ideal for busy individuals or those with limited outdoor space, providing convenience in managing laundry. [1]

H Touray et al. highlights the challenges faced by residents in West African tropical regions, where unpredictable weather, particularly the rainy season, complicates clothes drying. Existing solutions like hot dryers and outdoor racks are either expensive, inefficient, or damage clothes. Given the prohibitive cost of electric dryers, there is a clear need for an affordable and efficient drying system. The paper effectively underscores the need for smarter, cost-effective solutions tailored to the local climate, providing a solid foundation for developing automated, retractable clothes drying systems that can improve daily life in such regions.[2]

Felix Larbi Aryeh et al. (2020) introduced an automated clothes drying system designed for the tropical climate of West Africa, addressing challenges like humidity and unpredictable weather. Their system uses automation to retract and extend clothes based on weather conditions, providing an efficient solution for busy individuals. Emmanuel Kweku Eshun et al. (2021) focused on integrating renewable energy and sensors to reduce electricity consumption while ensuring effective drying. Michael Kwasi Antwi et al. (2022) developed a prototype with sensors to detect humidity and temperature, further improving the system's reliability. These studies highlight the potential of smart, energy-efficient drying systems tailored to local environmental conditions.[3]

Methodology

A. Overview

This project integrates multiple components to automate the covering/uncovering of clothes based on weather conditions. It is built around the NodeMCU microcontroller, which processes sensor data and controls the motor for the protective cover.

B. Sensors and Actuators (Working Purpose)

NodeMCU (ESP8266) in Automated Rain Detection Cloth Drying System

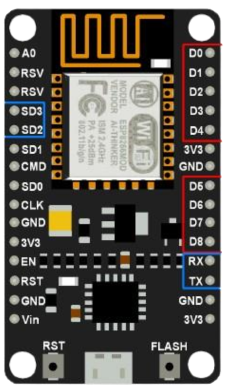


Figure: Node MCU (esp8266)

- **Purpose:** Acts as the central controller for the system.
- **Functions:**
  - Reads Rain Sensor and LDR data.
  - Controls Servo Motor for retracting/extending clothesline.
  - Sends alerts via Blynk and activates Buzzer/LEDs for notifications.
- **Features:**
  - Wi-Fi connectivity for IoT integration.
  - GPIO pins for sensor and actuator interfacing.
- **Performance:** Compact, efficient, and reliable, but dependent on stable internet and power.

Rain Sensor

Figure: Rain Sensor with Module

- **Purpose:** Detects rainfall.
- **Functionality:** When water droplets make contact, it signals the NodeMCU to activate the servo motor for covering the clothes.

Pin Connections:
Analog Output → NodeMCU **A0**
Signal Pin (D0) → NodeMCU **D8**
VCC → NodeMCU **D7**
GND → NodeMCU **GND**

LDR (Light Dependent Resistor)

- **Purpose:** Monitors sunlight intensity to determine whether to uncover the clothes after rain stops.
- **Functionality:** Reads light intensity and sends the data to NodeMCU for decision-making.

Figure: LDR Sensor

Pin Connections:
Analog Output → NodeMCU **A0**
Signal Pin (D0) → NodeMCU **D0**
VCC → NodeMCU **3.3V**
GND → NodeMCU **GND**

Servo Motor

Figure: Servo Motor

- **Purpose:** Moves the protective cover to shield or uncover the clothes. [Servo 1] . continuously moves to wipe the rain sensor. [Servo 2]
-

Pin Connections: (Servo 1)
Yellow Pin → NodeMCU **D6**
RED Pin → NodeMCU **3.3V**
BLACK Pin → NodeMCU **GND**

Pin Connections: (Servo 2)
Yellow Pin → NodeMCU **D5**
RED Pin → NodeMCU **3.3V**
BLACK Pin → NodeMCU **GND**

C. IoT Integration

Blynk IoT Platform: Displays real-time environmental data and allows remote control of the motor.

- Template ID: TMPL6pTCtiIz8 o
- Template Name: Automated IV Bag
- AuthToken: FD3YIIlB2VUPMBP_2Wg4-SidlxhEbDJ

ESP8266 connects to the local WiFi network using the credentials:

- SSID: " We are CSE'ian 🧑‍💻 "
- Password: "@wercsian "

Components:
Virtual Pins for sensor data.
Notifications for user alerts during rain.

D. System Workflow

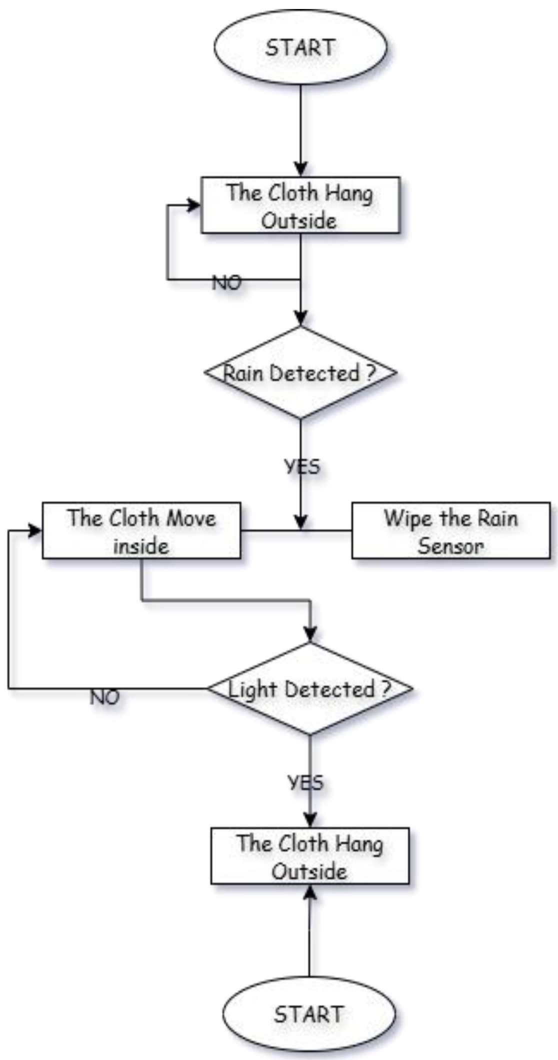
Initialization: All components are initialized, and the system starts monitoring weather conditions.

Rain Detection:
Rain sensor triggers the motor to cover the clothes.
Sends a notification to the user via the Blynk app.

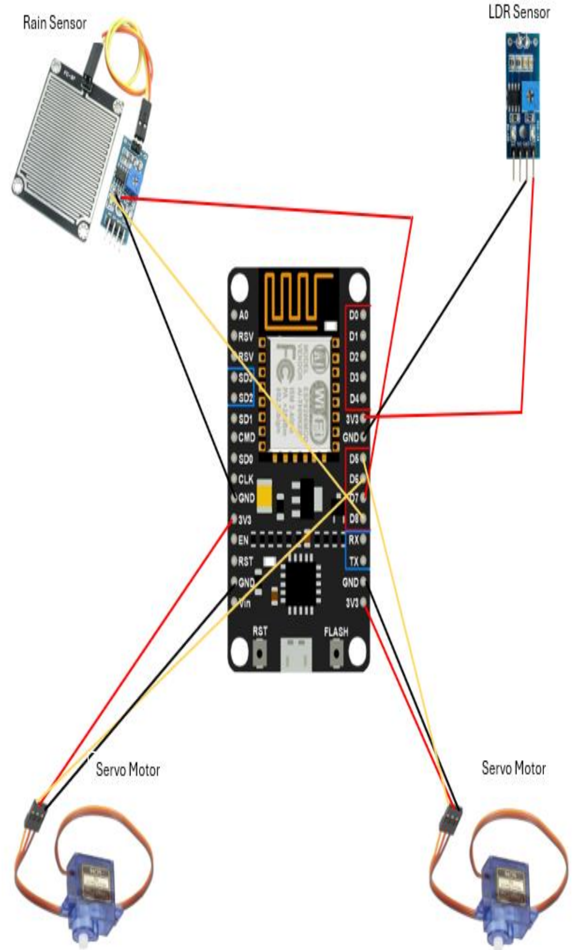
Sunlight Detection:
LDR monitors light levels to uncover clothes when the weather improves.

E. Diagrams:

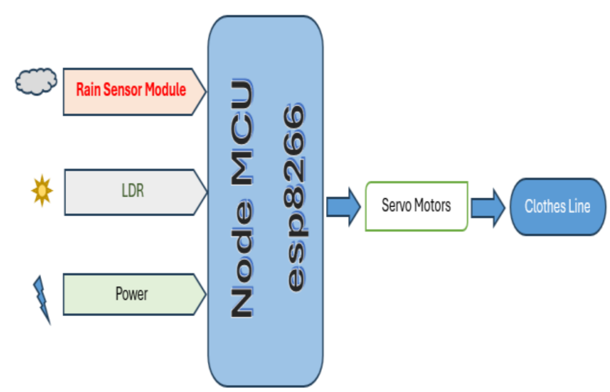
Flow Chart:



Work Flow Diagrams:



Block Diagram:



F. Algorithm

```
#define BLYNK_TEMPLATE_ID
"TMPL6sKQ7KdmN"
#define BLYNK_TEMPLATE_NAME
"Automatic Cloth Drying System1"
#define BLYNK_AUTH_TOKEN
"30bJe6b2uMeWqqtl3QJjpGAhw3FeCdhG"
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_PRINT Serial

char ssid[] = "We are CSE'ian 🧑‍💻"; //
Your WiFi SSID
char pass[] = "@wercseian";

// Servo motor instances
Servo servo1; // Protects clothes
Servo servo2; // Wipes the rain sensor

// Pin definitions
#define SERVO1_PIN D6
#define SERVO2_PIN D5
#define LDR_PIN D0
#define RAIN_SENSOR_PIN D8
#define RAIN_SENSOR_VCC D7

// Threshold for LDR (adjust based on
testing)
const int lightThreshold = 30;

void setup() {
  // Initialize servo motors
  servo1.attach(SERVO1_PIN);
  servo2.attach(SERVO2_PIN);

  // Set initial servo positions
  servo1.write(0); // Start at 0 degrees
  servo2.write(0); // Start at 0 degrees

  // Initialize rain sensor power
  pinMode(RAIN_SENSOR_VCC,
OUTPUT);
  digitalWrite(RAIN_SENSOR_VCC,
HIGH); // Power the rain sensor

  // Initialize serial monitor (optional for
debugging)
  Serial.begin(9600);
}

void loop() {
  Blynk.begin(BLYNK_AUTH_TOKEN,
ssid, pass);
  Serial.println("Connecting to Blynk...");

  // Read LDR value
  int ldrValue = analogRead(LDR_PIN);

  // Read rain sensor value
  int rainValue =
digitalRead(RAIN_SENSOR_PIN);

  // Debugging information (optional)
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  Serial.print("Rain Detected: ");
  Serial.println(rainValue);

  // Handle servo1 based on conditions
  if (rainValue == LOW) {
    Blynk.logEvent("Rain","warning", "Rain
Detected and protect clothes.");
    Serial.print("Rain Detected and protect
clothes.");
    // If rain is detected, protect clothes
    delay(2000);
    servo1.write(0); // Move to 90 degrees to
protect clothes
  } else if (ldrValue >= lightThreshold) {
    Blynk.logEvent("LDR","It's daytime and
no rain, allow clothes to dry.");
Serial.print("it's daytime and no rain, allow
clothes to dry.");
```

```
    // If it's daytime and no rain, allow clothes
to dry

    servo1.write(0); // Move to 0 degrees
  } else {
                    //Blynk.logEvent("warning","it's
nighttime, keep clothes protected.");
      Serial.print("it's nighttime, keep clothes
protected.");
    // If it's nighttime, keep clothes protected
    servo1.write(90); // Keep at 90 degrees
  }

  // Handle servo2 for wiping rain sensor
  if (rainValue == LOW) {
      // If rain is detected, move servo2
continuously
    servo2.write(90); // Move to 90 degrees
    delay(1000);      // Wait for 1 second
    servo2.write(0);  // Move back to 0 degrees
    delay(1000);      // Wait for 1 second
  } else {
    // Stop servo2 movement
    servo2.write(0);
  }
 Blynk.virtualWrite(V0, rainValue);
 Blynk.virtualWrite(V5, ldrValue);
 Blynk.run(); // Run Blynk
   delay(100); // Small delay to stabilize
readings
}

  } else servo2.write(0);

  delay(100);
}
```

G. Blynk IOT Platform

Figure: Blynk Web Dashboard

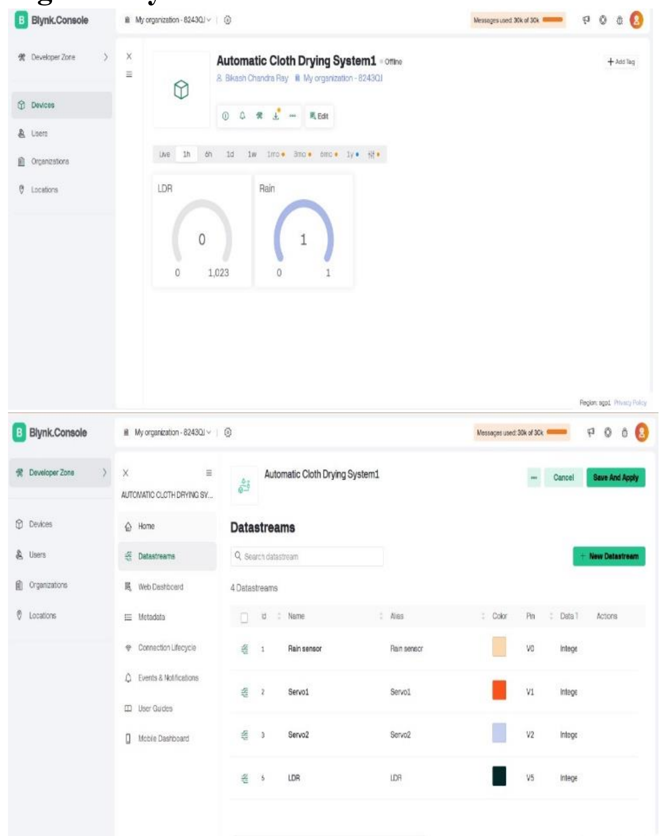
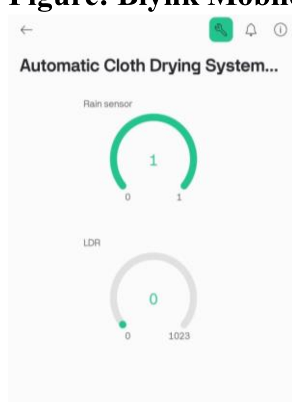


Figure: Blynk Mobile Dashboard



- Purpose:
  Enables real-time monitoring of LDR (light intensity) and rain sensor values remotely.
- Functionality:

- Displays the LDR sensor's analog readings (0 to 1023) and the rain sensor's digital output (0 or 1) on the Blynk dashboard.
- Provides notifications for critical sensor values, such as detecting rainfall or extreme light changes.

- How It Works:
  - The LDR sensor continuously updates its analog value via virtual pins, reflecting changes in light intensity.
  - The rain sensor sends a digital signal (0 for dry, 1 for rain) to the Blynk app for real-time monitoring.
  - Notifications are triggered using Blynk.logEvent() when specific thresholds are crossed.
- Result:
  - Offers real-time visibility of environmental conditions on both web and mobile dashboards.
  - Alerts users to critical changes, such as the onset of rain, enabling timely action.
- Performance Analysis:
  - Strength: Provides user-friendly, remote access to sensor data via the Blynk app.
  - Limitation: Dependent on stable Wi-Fi connectivity and the reliability of the Blynk platform.

H. System Summary

Component	Purpose	Result	Performance
Rain Sensor	Detects rain and triggers cloth retraction	Accurately identifies rainfall conditions	High sensitivity; may require maintenance to avoid false readings due to dust or debris.
LDR Sensor	Detects sunlight for drying purposes	Provides accurate light intensity measurements	Reliable within standard lighting conditions; limited in extreme weather.
Servo Motor	Automates the movement of the clothesline	Ensures smooth extension and retraction of clotheslines	Operates efficiently; performance depends on motor calibration and load weight.
NodeMCU (ESP8266)	Processes sensor data and controls actuators	Handles all operations and connects to the IoT platform	High processing efficiency; dependent on stable power and WiFi connectivity.
Blynk IoT Platform	Enables remote monitoring and control	Sends notifications and displays real-time data	Great for remote access; requires consistent internet connectivity.

Result Analysis

The system successfully integrates the rain sensor, LDR, and DHT11 to automate cloth drying management. The NodeMCU microcontroller ensures seamless data processing and control, while IoT integration enhances remote monitoring capabilities.

Conclusion

The Automated Rain Detection Cloth Drying System demonstrates the potential of IoT in daily life automation. By detecting environmental changes and reacting accordingly, the system reduces manual intervention and ensures the timely protection of clothes.

Limitations

1. **Dependency on Weather Conditions**:
   - The system operates based on sensor readings of rain and sunlight, which may occasionally be influenced by extreme weather conditions or sensor inaccuracies.
2. **Manual Intervention for Errors**:
   - drying setups or environments.
4. **Power Dependency**:
   - The system relies on uninterrupted power supply. A power outage would halt the automated functionality unless a backup power source is integrated.
5. **Material Constraints**:
   - The mechanism for retracting and extending clotheslines may not support heavy or oversized clothing, potentially leading to system strain.
6. **Sensor Calibration**:
   - If the sensor malfunctions or provides incorrect readings, manual adjustments or overrides may be required.
3. **Limited Range of Operation**:
   - The system is designed for small-scale use and may not be effective for larger-scale

   - Requires proper calibration of the rain sensor and LDR (Light Dependent Resistor) for accurate detection, which may degrade over time.
7. **Internet Connectivity Issues**:
   - For remote monitoring or control (if IoT-enabled), the system's functionality is highly dependent on stable internet connectivity.
8. **Environmental Factors**:
   - Dust, dirt, or debris could obstruct the sensors, reducing their efficiency or causing false readings. Regular maintenance is required

Future Plans

Add wind speed sensors for better decision-making during storms.

Develop a mobile application for extended control features.

Integrate machine learning to improve predictive weather responses.

References

[1] H. Nugraha, "Design an automatic clothes dryer in a cabinet with wi-fi transmission." IOP Conference Series: Materials Science and Engineering, vol. 852. no. 1. IOP Publishing, July 2020.

[2] N. G. Kishore Kumar Reddy and K. Rajeshwari, "Interactive clothes based on IoT using NFC and mobile application", 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 1-4, January 2017.

[3] Xiao-song Hu, Li-ling Jiang, Rui Cheng, Tie-jun Wang, and Qing Li, "A probabilistic clothes recommender based on cloth features," 2014 International Conference on Management Science & Engineering 21st Annual Conference Proceedings, Helsinki, pp. 76-81, August 2014.

[4] W. Makni, N. Ben Hadj, H. Samet, R. Neji, "Design simulation and realization of solar battery charge controller using Arduino Uno", 2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, pp. 635-639, December 2016.

[5] Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi and Gaetano Marrocco, "RFID Technology for IoT based personal healthcare in smart spaces", IEEE Internet of Things Journal, vol. 1, N.2, pp. 144-152, April 2014.

[6] Apurva Chaudhari, Bhushan Mapari and Shreenivas Jog, "Effective environmental monitoring & domestic home conditions by implementation of IoT", 2018 Fourth International Conference on Computing Communication Control and Automation

system", Department of Computer Engineering Sinhgad Academy of Engineering, Pune, India, February 2014.

(ICCUBEA), Pune, India, pp. 1-5, August 2018.

[7] Roushan Kumar and Adesh Kumar, "Design and hardware development of power window control mechanism using microcontroller", 2013 International Conference on Signal Processing And Communication (ICSC), Noida, pp. 361-365, December 2013.

[8] Poorva Parkhi, Snehal Thakur, Sonakshi Chauhan, "RFID based parking management

[9] Blynk IoT Platform Documentation. [Available Here]

[10] ESP8266 Configuration .[Available Here]

[10] Servo Configuration. [Available Here]

[11] Rain Sensor Functionality Documentation [Available Here]