Automatic Clothes Drying by Using Arduino UNO

**Project Proposal**

**Supervisor**

**Khandaker Iftakher Ahmed**

Lecturer C.C. of CSE

European University of Bangladesh

**Submitted By**

[**230122017, 230122072, 230122079,**

**230122080, 220122029, 220122030**]

Students of CSE

European University of Bangladesh

[January 2025]

**Department of Computer Science & Engineering**

European University of Bangladesh.

**Automatic Clothes Drying by Using Arduino UNO**

*Faculty of Computer Science & Engineering, European University of Bangladesh.*

***A****bstract*— Our main goal is to develop an ultramodern and reliable automatic clothes drying by using arduino uno. Which is faced in almost every field in our daily life. For example unpredictable weather, time constraints, limited space, heat sensitivity, night drying, etc. To develop this system, we used an Arduino UNO as the main processor, a rain sensor, a servo motor, resistors, jumper wires, a 9V battery, and other components.

# **Introduction**

An automatic clothes drying system using Arduino UNO automates the drying process using sensors to adapt to weather conditions. It retracts clothes during rain and optimally positions them in sunlight. This system is energy-efficient, affordable, and customizable, making it ideal for homes and shared spaces. It provides a reliable, time-saving, and modern solution for efficient laundry management.

# **Problem Description**

Traditional clothes drying methods face challenges, especially in regions with unpredictable weather. Rain, lack of sunlight, and manual effort often delay the process and inconvenience users. Electric dryers, while effective, consume high energy and are costly. An automatic clothes drying system using Arduino UNO addresses these problems by providing a weather-responsive, energy-efficient, and low-cost solution for reliable and convenient clothes drying.

**Methodology**

**Equipments:**

* Arduino UNO
* BreadBoard
* Rain Sensor
* Servo Motor
* Resistors
* Jumper Wires
* 9V Battery

**Use Case Diagram:** A use case diagram, a type of Unified Modeling Language (UML) diagram, visually represents the interactions between users (actors) and a system, outlining the different ways a user can interact with and achieve goals within the system.



**Circuit Diagram:** To develop this system, we used an Arduino UNO as the main processor, a rain sensor, a servo motor, resistors, jumper wires, a 9V battery, and other components.

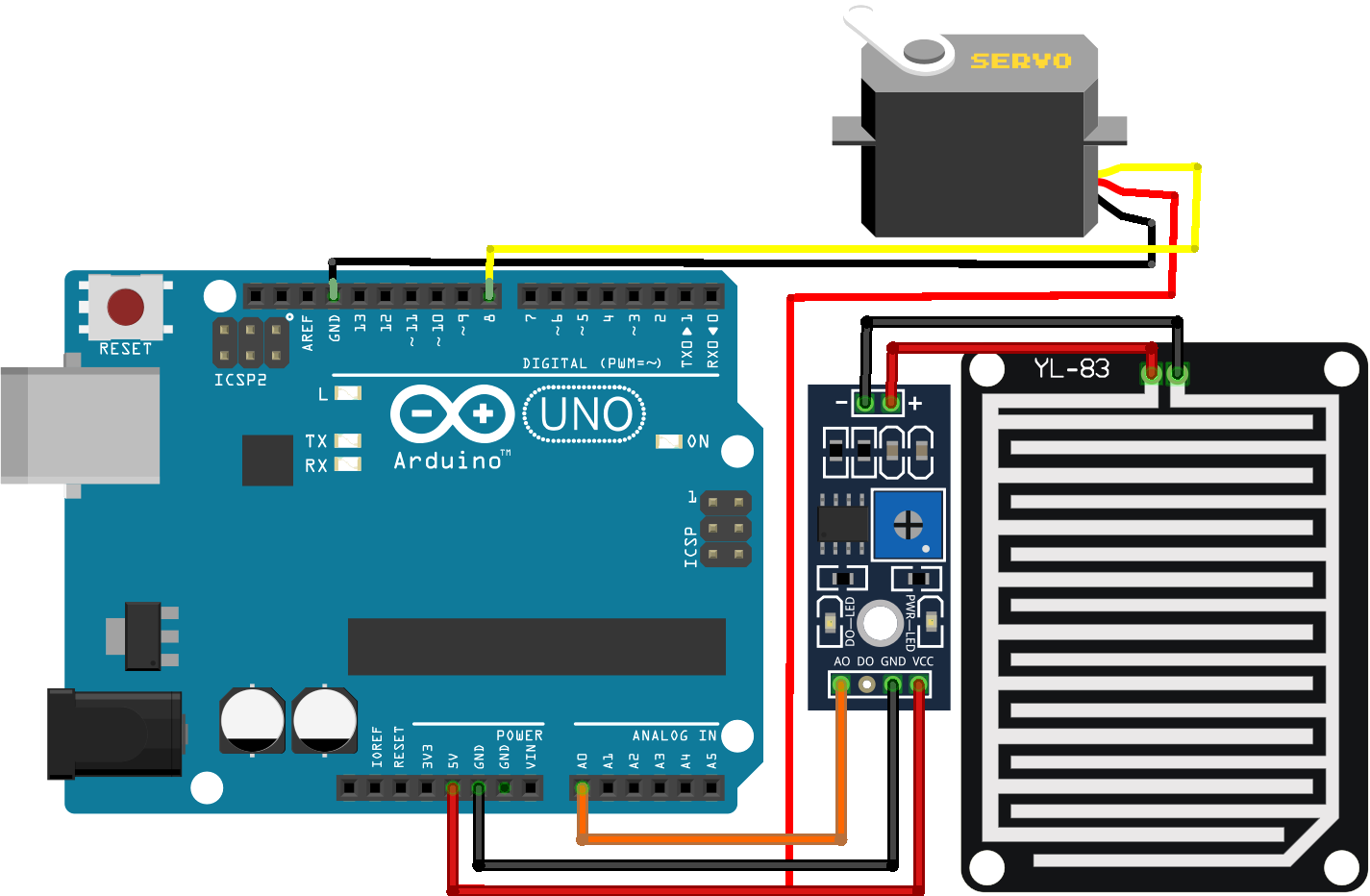


Fig. Circuit Diagram

**Arduino UNO Code:**

| **#include <Servo.h>**  *// Pin configuration* **const** **int** rainSensorPin = A0; *// Analog pin connected to the rain sensor* **const** **int** threshold = 500; *// Threshold value for rain detection (adjust as needed)* Servo clothRackServo; *// Servo object*  **const** **int** servoPin = 9; *// Pin connected to the servo signal* **const** **int** servoOpenAngle = 0; *// Angle when it rains* **const** **int** servoCloseAngle = 90; *// Angle when no rain*  **int** currentAngle = servoCloseAngle; *// Keeps track of the current servo position*  **void** setup() {  Serial.begin(9600);  clothRackServo.attach(servoPin);  clothRackServo.write(currentAngle); *// Default position*  Serial.println("System Initialized"); }  **void** loop() {  **int** rainValue = analogRead(rainSensorPin);  Serial.print("Rain Sensor Value: ");  Serial.println(rainValue);   if (rainValue < threshold && currentAngle != servoOpenAngle) {  *// Rain detected, move to 0 degrees gradually*  moveServoSmoothly(currentAngle, servoOpenAngle);  } else if (rainValue >= threshold && currentAngle != servoCloseAngle) {  *// No rain, move to 90 degrees gradually*  moveServoSmoothly(currentAngle, servoCloseAngle);  }   delay(500); *// Delay for stability* }  *// Function to move servo smoothly to a target angle* **void** moveServoSmoothly(**int** startAngle, **int** targetAngle) {  **int** step = (startAngle < targetAngle) ? 1 : -1; *// Determine direction of movement*  for (**int** angle = startAngle; angle != targetAngle + step; angle += step) {  clothRackServo.write(angle);  delay(10); *// Delay for smoothness (lower value = smoother but slower)*  currentAngle = angle;  } } |
| --- |

# **Project Controlling:** Project controlling, a crucial element of project management, involves monitoring and managing a project's progress to ensure it stays on track, achieves goals, and stays within budget and timeline, by comparing actual performance against planned objectives and taking corrective actions when necessary.



# **Project Scope**

# **Advantages**

# **Energy-saving:** Uses low-power components.

# **Weather-responsive**: Adapts to rain or sunlight.

# **Convenient**: Automates drying without manual effort.

1. **Affordable**: Budget-friendly components like Arduino UNO.
2. **Limitations**
3. **Weather-dependent**: Limited efficiency in extreme weather conditions.
4. **Initial setup**: Requires technical knowledge for assembly and programming.
5. **Maintenance**: Sensors and components may need regular upkeep.
6. **Power source**: Needs a consistent electricity supply or backup.
7. **Enhancements**
8. **Improved sensors**: Use high-precision sensors for better weather detection.
9. **Smart features**: Integrate with IoT for remote control and monitoring.
10. **Backup power**: Add solar panels for uninterrupted operation.
11. **Durable materials**: Use weather-resistant components for longer life.

# **Feasibility Study**

# **Technical Feasibility**

# **Cost-effective**: Arduino UNO and related components are affordable.

# **Implementation**: Suitable for DIY enthusiasts with basic technical skills.

1. **Scalability**: Can be customized or upgraded for advanced features.
2. **Integration**: Compatible with smart home systems for added convenience.
3. **Economic Feasibility**
4. **Low initial cost**: Uses affordable components like Arduino UNO.
5. **Maintenance expenses**: Minimal costs for repairs and upkeep.
6. **Energy efficiency**: Reduces electricity usage, saving on utility bills.
7. **Scalable investment**: Can start small and upgrade over time.
8. **Operational Feasibility**

# **Ease of use**: Simple controls for daily operation.

1. **Maintenance**: Requires minimal upkeep for sensors and components.

# **Reliability**: Functions effectively under normal conditions.

1. **Scalability**: Easily adaptable to varying household needs.

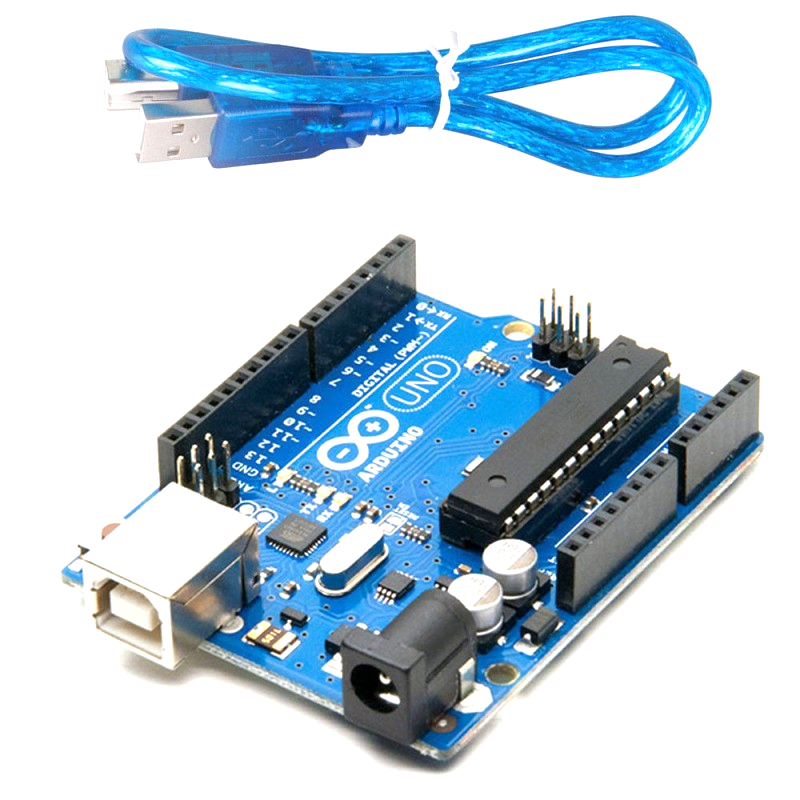
# **Solution Application Areas**

1. **Households**
2. **Convenience**: Automates the drying process for busy families.
3. **Space-saving**: Ideal for small living spaces with limited drying areas.
4. **Weather protection**: Ensures clothes stay dry during unpredictable weather.
5. **Energy efficiency**: Reduces reliance on traditional electric dryers.
6. **Hostels/Dormitories**
7. **Shared convenience**: Simplifies drying for multiple users in shared spaces.
8. **Space optimization**: Maximizes limited drying areas effectively.
9. **Weather adaptability**: Ensures reliable drying during unpredictable weather.
10. **Cost efficiency**: Reduces the need for costly electric dryers.
11. **Hotels/Resorts**
12. **Guest convenience**: Modern drying solution for a superior experience.
13. **Weather reliability**: Consistent drying performance in any condition.
14. **Space utilization**: Optimizes laundry areas for better functionality.
15. **Luxury appeal**: Enhances the reputation of the hotel/resort with modern amenities.

# **Tools/Technology**

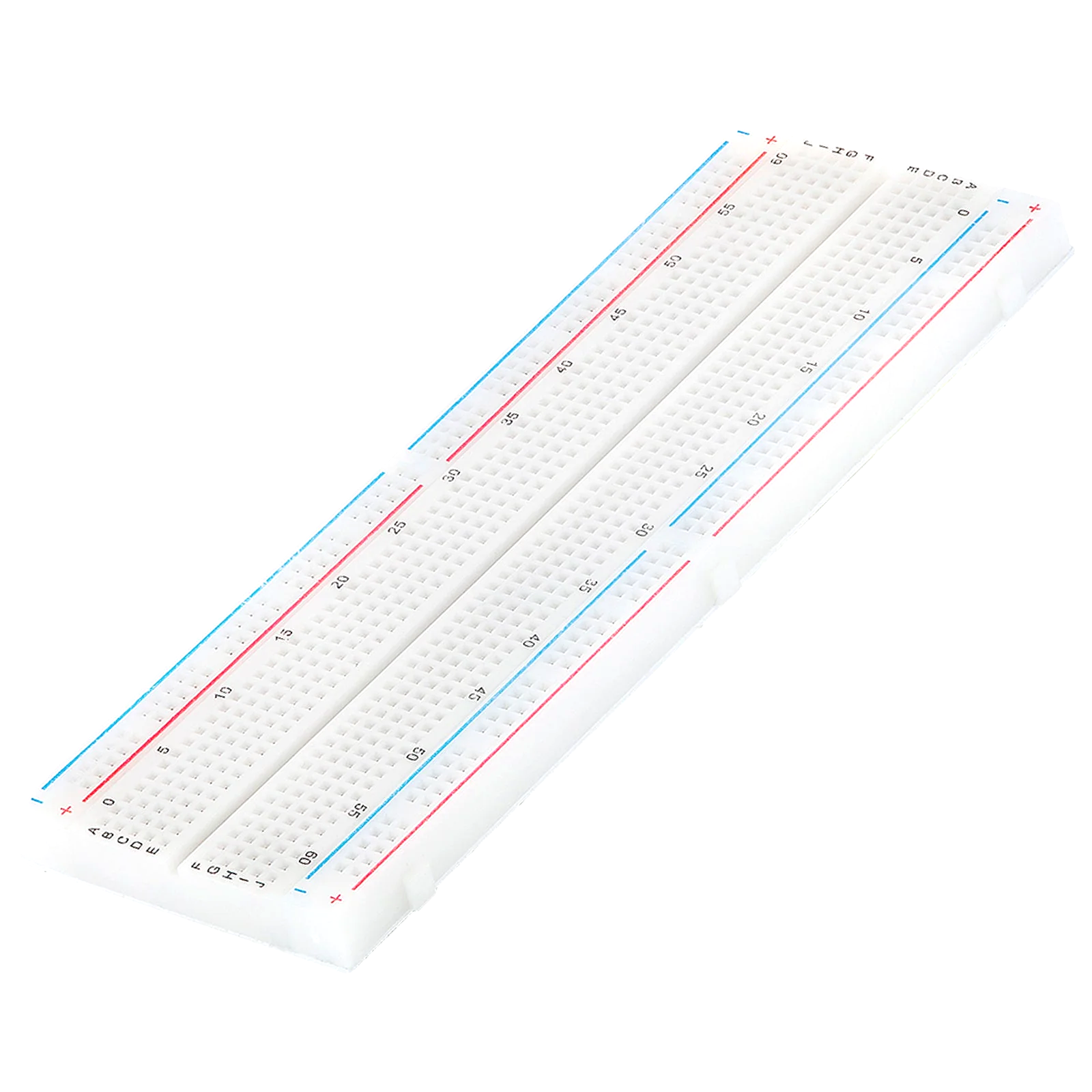
**Hardware:**

* **Arduino UNO**

****

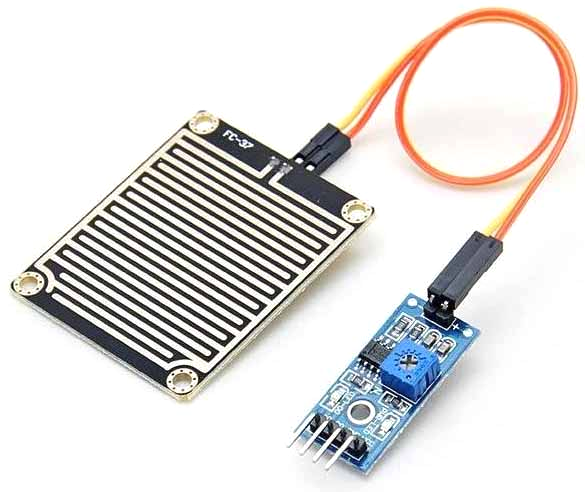
The Arduino Uno is a popular, open-source microcontroller board based on the ATmega328P, designed for creating interactive and electronic projects, and is known for its ease of use and programming, making it a great tool for both beginners and experienced users.

* **BreadBoard**



The breadboard is the basic component of any circuit building process. All components, be it input sensors or output display devices are connected to the power supply, microcontroller using wired connections through a breadboard.

* **Rain Sensor**



A rain sensor, also known as a rain switch or rain shut-off device, is an electronic device that detects rainfall and triggers a programmed action, such as shutting off an irrigation system or activating windshield wipers.

* **Servo Motor**



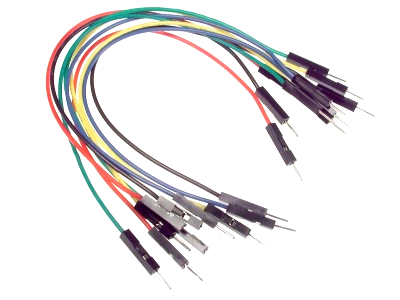
A servomotor is a closed-loop servomechanism that uses position feedback (either linear or rotational position) to control its motion and final position.

* **Resistors**

# 

Resistors are passive devices that restrict the flow of current or divide the voltage through the circuit. The input power passes through these resistors and then to the sensors to avoid damage.

* **Jumper Wires**



These are the main components that are used to establish the connections between different devices of the circuit.

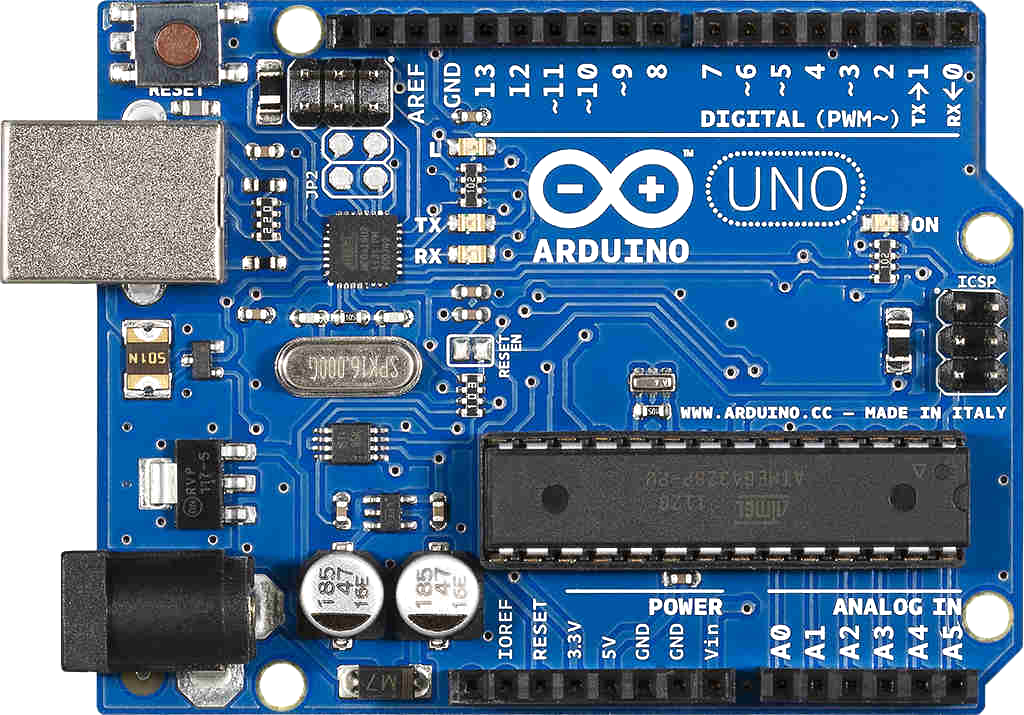
* **9V Battery**



A 9V battery is a compact, rectangular-shaped battery that provides a nominal voltage of 9 volts, commonly used in devices like smoke detectors, clocks, and remote controls, etc.

**Software:**

* Not applicable (if built as a hardware-based project).



The Arduino Uno is a series of open-source microcontroller boards based on a diverse range of microcontrollers. It was initially developed and released by Arduino company in 2010.

* For microcontroller-based systems, software like Arduino UNO IDE may be used.



The Arduino IDE (Integrated Development Environment) is a software application used to write, compile, and upload code to Arduino boards, including the Arduino Uno, facilitating the creation and deployment of projects based on these microcontrollers.

* TinkerCad circuit simulation software.



Tinkercad is a free, web-based 3D design, electronics, and coding platform developed by Autodesk, known for its ease of use and accessibility, making it ideal for beginners and educators to explore 3D modeling, circuits, and coding concepts.

# **Expertise of the Team Members**

We present the significance of our project with the overall cooperation of the team members:

1. **Energy-efficient**: The system uses low-power components such as DC motors and sensors, making it eco-friendly and cost-effective compared to traditional electric dryers.
2. **Weather-adaptive**: Equipped with rain or light sensors, it automatically responds to weather changes, retracting clothes during rain or exposing them to sunlight for faster drying.
3. **Convenient**: By automating the drying process, it saves time and reduces the need for constant manual effort, making life easier for users.
4. **Affordable**: Arduino UNO and its components are budget-friendly and widely available, making the system a cost-effective solution for most households.
5. **Customizable**: The setup can be tailored to include features like Bluetooth/WiFi control or smart home integration, based on user preferences.
6. **Reliable**: When properly designed, it consistently functions under typical conditions, providing dependable drying performance.

# **Milestones**

* **Cost:**

| **SN** | **Hardware** | **Cost** |
| --- | --- | --- |
| 1 | Arduino UNO | 580 |
| 2 | BreadBoard | 70 |
| 3 | Rain Sensor | 20 |
| 4 | Servo Motor | 120 |
| 5 | Resistors | 20 |
| 6 | Jumper Wires | 70 |
| 7 | 9V Battery | 80 |

* **Flowchart:**



**Conclusions**

The “Automatic Clothes Drying Using Arduino UNO” project offers an efficient solution for automating the drying process. By using an Arduino UNO, rain sensor, and servo motor, the system automatically adjusts based on environmental conditions, ensuring optimal drying while conserving energy. This project showcases the potential of smart automation to enhance everyday household tasks.

**References**

We mentioned all the literature or web references, etc here-

* Internet web page & resources: [ResearchGate](https://www.researchgate.net/profile/Md-Abu-Sayed-24), Copilot, etc.

End!