Automatic Grain Protecting Roof Impervious to Rain

Creating an automatic grain protecting roof that is impervious to rain using an Arduino Uno involves a system of sensors, actuators, and control logic to ensure that the roof opens and closes based on weather conditions. Here's a detailed guide on how to design and implement such a system.

Overview

The goal is to develop a roof mechanism that can automatically close when it detects rain and open when conditions are clear. This system can protect stored grains from water damage, utilizing sensors and a motorized mechanism.

Components Needed

1. **Arduino Uno**: The microcontroller that will control the system.
2. **Rain Sensor Module**: Detects rain by measuring moisture.
3. **Servo Motor or DC Motor with Gearbox**: Controls the movement of the roof.
4. **Motor Driver Module**: If using a DC motor (e.g., L298N).
5. **Limit Switches**: To prevent over-rotation of the motor when opening/closing the roof.
6. **Power Supply**: Suitable for the motor (usually 12V for DC motors).
7. **Jumper Wires**: For connections.
8. **Breadboard**: Optional for prototyping.
9. **Resistors**: For voltage dividers or pull-up resistors.

Circuit Diagram

1. **Rain Sensor**:
    - Connect VCC to Arduino 5V.
    - Connect GND to Arduino GND.
    - Connect the output pin to a digital pin on the Arduino (e.g., D2).
2. **Motor Control**:
    - For a **Servo Motor**:
        - Connect the servo signal pin to a PWM-capable pin on the Arduino (e.g., D9).
    - For a **DC Motor**:
        - Connect motor terminals to the motor driver.
        - Connect the motor driver inputs to Arduino digital pins (e.g., D3 and D4).

3. **Limit Switches**:
    - Connect one switch to an Arduino digital pin (e.g., D5 for the roof closed position) and another to a different digital pin (e.g., D6 for the roof open position).

Basic Logic

1. **Sensor Reading**: Continuously read the value from the rain sensor.
2. **Motor Control**: If rain is detected, activate the motor to close the roof. If no rain is detected and the roof is closed, open the roof.
3. **Limit Switch Feedback**: Use limit switches to stop the motor when the roof reaches the fully open or closed position.

Sample Code

Here's an example of how to implement this in Arduino code:

cpp

Copy code

```
#include <Servo.h>


// Define pins
const int rainSensorPin = 2; // Rain sensor digital pin
const int limitSwitchClosedPin = 5; // Limit switch for closed position
const int limitSwitchOpenPin = 6; // Limit switch for open position
const int motorPin = 9; // Servo motor pin

Servo roofServo;

void setup() {
  pinMode(rainSensorPin, INPUT);
  pinMode(limitSwitchClosedPin, INPUT_PULLUP);
  pinMode(limitSwitchOpenPin, INPUT_PULLUP);

  roofServo.attach(motorPin);
  roofServo.write(0); // Start in the closed position
}
```

```
void loop() {
  int rainDetected = digitalRead(rainSensorPin);
  int closedPosition = digitalRead(limitSwitchClosedPin);
  int openPosition = digitalRead(limitSwitchOpenPin);

  if (rainDetected == HIGH) {
    // Rain detected
    if (closedPosition == HIGH) {
      // Close the roof
      roofServo.write(0); // Adjust the angle as necessary
    }
  } else {
    // No rain detected
    if (openPosition == HIGH) {
      // Open the roof
      roofServo.write(180); // Adjust the angle as necessary
    }
  }

  delay(500); // Adjust the delay as necessary
}
```

Adjustments and Improvements

1. **Motor Control**:
    - If using a DC motor, control speed and direction using PWM and the motor driver. Adjust the code to implement speed control if needed.
2. **Sensor Calibration**:
    - Ensure the rain sensor is calibrated correctly to detect rain based on the environmental conditions where it will be used.
3. **Weather Conditions**:
    - Consider additional sensors like temperature or humidity sensors to enhance functionality (e.g., closing the roof if high humidity is detected).
4. **Manual Override**:

- Add a switch to allow manual opening and closing of the roof.

5. **Data Logging**:
    - Optionally, log data on weather conditions for future analysis.

Final Thoughts

This project integrates various components and concepts, including sensor integration, motor control, and Arduino programming. Ensure to test the system thoroughly in a controlled environment before deploying it in real conditions to protect the grain effectively. Happy building!