Click to verify

Arduino rain sensor

A simple Rain Detection System can be easily made using an Arduino connected with a Rain Sensor. The sensor will detect any rainfall and send the signal to the Arduino, which can then perform needed actions. This system has many uses in various fields like agriculture, automobiles, and home automation. In agriculture, it can be used for automatic irrigation control, while in automobiles, windshield wipers can become fully automatic with this rain detection system. Home Automation Systems can also use this system to automatically close windows and adjust room temperature when it rains. In this tutorial, we will create a basic rain sensor using Arduino and a buzzer. This setup can be used as a starting point for other projects. The Rain sensor module is also known as a raindrop sensor or rain gauge sensor depending on its usage, but they all work on the same principle. We have previously built a simple Rain Alarm and automatic car wiper using only a 555 Timer if you prefer not to use an Arduino. Here are the materials needed for this project: * Arduino UNO * Rain Sensor * Buzzer * Breadboard * Connecting wires The rain sensor module consists of two boards: the Rain Board and the Control Board. The Rain Board has two copper tracks that provide high resistance under dry conditions, with an output voltage of 5V. As the board gets wetter, its resistance decreases, and so does the output voltage. The Control Board controls the sensitivity and converts the analog output to digital output using an LM393 OP-Amp Comparator. This comparator compares two different voltage values and outputs a digital signal based on whether the analog value is above or below the threshold value. To connect everything, we need four pins on the Rain Board module and two more on the Control Board module. In this setup, the rain board detects the rainwater, while the control board controls the sensitivity and converts the analog values to digital signals. The working of the rain sensor is simple: under dry conditions, the rain board offers high resistance, resulting in a 5V output voltage. This voltage is read as 1023 by an Arduino. The Rain Drop Sensor works by measuring the change in resistance on its board due to water. When it rains, the board gets wet, causing a decrease in resistance which affects the output voltage. As the board becomes fully wet, the output voltage drops to almost zero (0V), similar to what an Arduino analog pin reads as 0 value. If partially wet, the output is proportional to the offered resistance. The formula ADC = (analog voltage value x 1023)/5 helps convert any analog voltage to Arduino's read value. The sensor circuit diagram connects the rain gauge module to a control board with VCC at 5V and ground at GND. The D0 pin is connected to an Arduino digital pin, which must be declared as output in the program. However, due to limitations in getting exact output voltage values, A0 is used instead for easier monitoring of output changes. The buzzer can be connected to any Arduino digital pin; if it requires more than 5V, a relay circuit or transistor should be used. The code for this project uses the Arduino IDE and defines pin A0 as rainfall and pin 5 as a buzzer. It declares variables 'value' and 'set', setting 'set' to 10 which can be adjusted according to the required level of operation. The setup initializes serial communication, sets the buzzer as an output, and the rainfall pin as both an input and output. In the loop function, it reads the rain sensor value using analogRead, maps this value between 0 and 1023 to 225, and if greater than or equal to 'set', prints "rain detected". If the value from the rain sensor is higher than the set threshold, the program will enter a loop, display a message on the serial monitor, and turn on the buzzer. Otherwise, it will switch off the buzzer. The setup is designed to trigger when there's no rain, indicated by lower values. This Rain Detection system works by using rainwater as a trigger to activate the buzzer. The Arduino Code for this system defines pins 5 and A0 as the buzzer and rainfall sensors respectively, making it easy to modify these settings without affecting the rest of the code. In the loop function, the analogRead command reads the sensor's value, while Serial.println(value) displays it on the serial monitor for debugging purposes. The map function then adjusts this value between 0-225. Based on this adjusted value and the set threshold, the buzzer is either turned on or off. This same principle can be applied to other applications like windshield wipers, home automation, and agriculture sectors. The guide covers using an Arduino UNO R4 with a rain sensor to detect rain or snow, including topics such as connecting the sensor, programming for digital signals, and determining rain intensity through analog signals. Connect the VCC pin of the rain sensor module to either a 3.3V or 5V pin on the Arduino UNO R4, but ensure to use an output pin to minimize electrochemical corrosion. The AO pin outputs a lower value when more water is present on the sensing pad and a higher value when there's less water. You can adjust the sensitivity with the potentiometer for the DO pin. If the rain level exceeds the set limit, the sensor's output becomes LOW, and the DO-LED light turns on; otherwise, it stays HIGH, and the LED stays off. The code defines POWER_PIN as 3 and DO_PIN as 4, setting up the Arduino to read the sensor's output. In the loop function, it checks if rain is detected based on the digitalRead result from the DO_PIN. When running the code with a few drops of water on the sensor, you should see the Serial Monitor display whether rain is or isn't detected. Some water on the rain sensor Check the result on the Serial Monitor. 225 2426 236 563 687 959 975 1009 1017 1053 1078 841 743 440 279 ※ OUR MESSAGES Please feel free to share the link of this tutorial. However, Please do not use our content on any other websites. We invested a lot of effort and time to create the content, please respect our work! The rain sensor can detect and measure rain/snow level. It provides two outputs: digital output (LOW/HIGH) and analog output. In this tutorial, we will learn how to use an Arduino and a rain sensor to detect and measure rain. We will cover connecting the rain sensor to an Arduino, programming the Arduino to detect rain by reading the digital signal, and measuring the rain level by reading the analog signal. Afterward, you can modify the code to activate a motor or warning when it detects rain/snow The rain sensor is useful for detecting presence of rain or measuring water dropped by rain. It provides two options via digital output pin and analog output pin. The rain sensor includes two parts: sensing pad and electronic module. The sensing pad is placed outside that can face with rain/snow, and has exposed copper traces, divided into power traces and sense traces. These power traces and sense traces are not connected unless they are bridged by water or snow. The electronic module converts the signal from the sensing pad to analog or digital value that can be read by Arduino. It includes four pins: VCC pin (connected to 3.3V to 5V), GND pin (connected to 0V), DO pin (digital output pin, HIGH if rain is not detected and LOW if detected), and AO pin (analog output pin, output value decreases with water in sensing pad). # Powering Rain Sensor with Arduino for Reduced Electrochemical Corrosion To minimize the impact of electrochemical corrosion on the rain sensor, connect its VCC pin to an output pin on your Arduino board. This allows you to control the power supply and reduce degradation over time. You can choose one or both outputs depending on your requirements. ## Wiring Diagrams | Connection | Description | | --- | --- | | DO only | Power pins connected as shown in the diagram: [Fritzing Image] | | AO only | Power pins connected as shown in the diagram: [Fritzing Image] | | Both AO and DO | Power pins connected as shown in the diagram: [Fritzing Image] | ## Code Use the following Arduino code to power the rain sensor when reading only: ```c #define POWER_PIN 3 #define DO_PIN 4 void setup() { Serial.begin(9600); pinMode(POWER_PIN, OUTPUT); pinMode(DO_PIN, INPUT); } void loop() { digitalWrite(POWER_PIN, HIGH); delay(10); int rain_state = digitalRead(DO_PIN); digitalWrite(POWER_PIN, LOW); if (rain_state == HIGH) Serial.println("The rain is NOT detected"); else Serial.println("The rain is detected"); delay(1000); } ``` ## How it Works 1. The code defines two constants for the power pin and DO pin. 2. In the setup function, serial communication is initialized at a rate of 9600 bps, and the power pin is set as an output and the DO pin as an input. 3. In the loop function, the power pin is set high, then low after reading the state of the DO pin. 4. If the rain sensor detects rain (HIGH), it prints "The rain is NOT detected" to the serial monitor. Otherwise, it prints "The rain is detected". ## Additional Tips * Adjust the potentiometer to fine-tune the sensitivity of the sensor if you notice the LED status remaining on constantly or off even when the sensor faces rainfall. * If you're unsure about LEDs and rain sensors, check out our Arduino - Rain Sensor tutorial for more information. Looking forward to seeing everyone at the meeting tomorrow and discussing our strategies. Arduino IDE, select the right board and port Copy the above code and open with Arduino IDE Click Upload button on Arduino IDE to upload code to Arduino Move your hand in front of sensor See the change of LED's state Read the line-by-line explanation in comment lines of source code! We are considering to make the video tutorials. If you think the video tutorials are essential, please subscribe to our YouTube channel to give us motivation for making the videos. See the best Arduino kit for beginner Quick and simple start guide for using and exploring the Rain Sensor module sometimes called a "Raindrops Sensor Module" with an Arduino. I recently wrote about flame sensors, so I thought a water sensor might be a good next project.Materials needed: Usage:Rain sensors are used to detect water beyond what a humidity sensor can do. How it works: The rain sensor detects water that completes the circuits on its sensor boards' printed leads. The sensor board acts as a variable resistor that changes from 100k ohms when wet to 2M ohms when dry. Pins: A0..... Analog output D0..... Digital output GND..... Ground VCC..... Positive voltage (input: 5v for analog 3.3v for Digital.)Loop Pins:+ ..... Sensor board hookup A- ..... Sensor board hookup BDimensions:2.17 in x 1.57 in x 0.31 in (5.5 cm x 4.0 cm x 0.8 cm)Weight: 0.28 oz (8 g) Testing:To test the Rain Sensor and ensure it's working correctly, connect the VCC to a 5v power source and GND. Try placing a few droplets of water on the sensor detection board, and the D0-LED should light up. Troubleshooting: If the D0-LED doesn't light up, check if the module is hooked up properly. Sometimes, salinity can be an issue, so using filtered, bottled water might help. Double-check that all solder joints are secure. This Arduino project utilizes a rain sensor to detect rainfall and snowfall, providing real-time feedback. The sensor's resistance value changes according to the amount of water falling on it, allowing for an output voltage that can be converted into a digital value using the LM393 High Precision Comparator. The sensor pad is connected to the LM393 module, which also features a potentiometer for adjusting sensitivity. A serial monitor, such as Putty, can be used to view the rainfall values in real-time. To use this sensor with an Arduino UNO board, various components are required and must be connected according to a provided circuit diagram. The code accompanying the project enables the serial monitor to display the rainfall value, with higher analog values indicating low rainfall. This project showcases the versatility of rain sensors and their potential applications in home automation, such as closing windows or moving fabrics when it rains. { if value < 300 then digitalWrite(4, HIGH); Serial.print("Heavy rain LED on "); else digitalWrite(4, LOW); } The sensor's resistance decreases when it detects rain, sending a signal to the control board. This board translates the signal into a digital and analog output that can be used by an Arduino or microcontroller. Operating specifications include a voltage range of 3.3V to 5V DC, with sensitivity adjustable via potentiometer on the control board. Components required for setup include a rain sensor module, an Arduino board (e.g., Uno), jumper wires, resistors (if needed), and a power supply. Optional components include a relay module, buzzer/LED, servo motor, and additional sensors or actuators. To connect the rain sensor to an Arduino, follow these steps: Connect VCC from the control board to the 5V pin on the Arduino, GND to the GND pin, digital output (D0) to any digital input pin, and analog output (A0) to one of the Arduino's analog input pins. The code for the rain sensor detection involves continuously reading the digital and analog signals from the sensor. If the digital output is HIGH, no rain is detected; if it's LOW, rain is present. The analog output provides a range of values indicating rain intensity. Based on this information, actions like turning on an LED or controlling a motor can be triggered. The code snippet provided is for an Arduino-based weather station that detects rain and controls a LED indicator. The key parts of the code include: - Pin declarations where digital and analog pins are assigned to variables along with the LED pin setup. - The setup function which sets up the rain sensor digital and analog pins as inputs, and the LED pin as output. It also initializes serial communication for debugging purposes. In the loop function, the digital signal from the rain sensor is read using `digitalRead()` and the analog signal is read using `analogRead()`. Depending on whether it detects rain, the code either turns on the LED or keeps it off. Additionally, it prints the analog value to give an idea of the intensity of rain. Smart watering systems for plants use weather forecasts to optimize irrigation schedules, preventing overwatering and saving resources. This tech also enables smart windows or roofs that automatically adjust when it rains, reducing water entry into buildings. Rain sensors can trigger valves in rainwater harvesting systems, efficiently directing water to storage tanks without human intervention. These sensors provide crucial data for weather stations, combining with other sensors like temperature and humidity to create a comprehensive monitoring system. The benefits of using a rain sensor module include low costs, ease of integration with Arduino, reliable moisture detection, energy efficiency, and versatility in various applications from home automation to agriculture. The rain sensor's ability to detect precipitation and trigger actions makes it an ideal tool for DIY projects and professional applications alike. By understanding how this technology works and experimenting with its uses, individuals can enhance their automation systems, contributing to more efficient and responsive environments that conserve water and protect against rain damage.