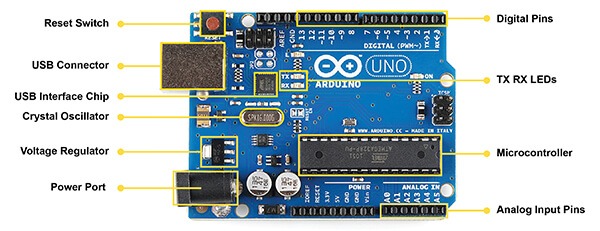
**INTRODUCTION**

**A simple Rain Detection System can be easily built by interfacing an Arduino with Rain Sensor. The sensor will detect any rainfall falling on it and the Arduino board will sense it and can perform required actions. A system like this can be used in many different fields, such as agriculture and automobile fields. Rainfall detection can be used to automatically regulate the Irrigation process. Also, continuous rainfall data can help farmers use this smart system to automatically water the crop only when absolutely required. Similarly, in the automobiles sector windshield wipers can be made fully automatic by using the rain detection system. And the Home Automation Systems can also use rain detection to automatically close windows and adjust room temperature. In this tutorial, we will build a basic rain sensor using Arduino with a buzzer. You can then use this set-up to build anything you wish on top of it. Also, note that the rain sensor module is also referred to as a raindrop sensor or rain gauge sensor or rainwater sensor based on usage, but they all refer to the same sensor used in this project and they all work on the same principle.**

**We have also built a simple Rain Alarm and an automatic car wiper by using 555 Timer only, you might want to check that as well if you do not want to use an Arduino.  That being said, let’s get back to this project and start building our Arduino Rain Gauge.**

**Materials Required**



**(Arduino UNO)**

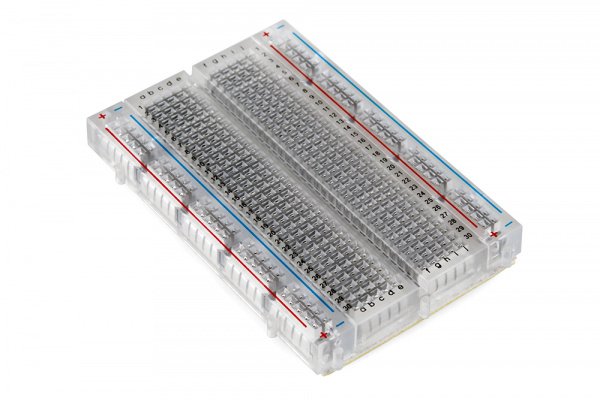
**Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LEDYou can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.**

**A rain sensor or *rain switch* is a switching device activated by rainfall. There are two main applications for rain sensors. The first is a water conservation device connected to an automatic irrigatio system that causes the system to shut down in the event of rainfall. The second is a device used to protect the interior of an automobile from rain and to support the automatic mode of windscreen wipers****.**  
 **(Rain sensor)**

****

**(Buzzer)**

**An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.**



**(Breadboard)**

**Breadboards are one of the most fundamental pieces when learning how to build circuits. In this tutorial, you will learn a little bit about what breadboards are, why they are called breadboards, and how to use one. Once you are done you should have a basic understanding of how breadboards work and be able to build a basic circuit on a breadboard**.



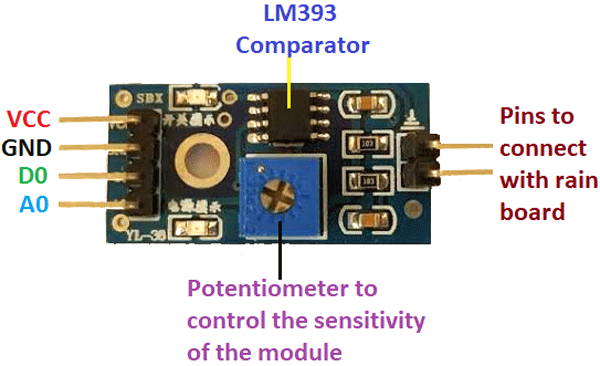
**(Connecting wires)**

**This library allows you to communicate with I2C devices, a feature that is present on all Arduino boards. I2C is a very common protocol, primarly used for reading/sending data to/from external I2C components. To learn more, visit this article for Arduino & I2C.**

**RAIN DETECTING SENSOR**



**The Rain control module which is shown below consists of 4 pins to connect the Arduino namely VCC, GND, D0, A0 and two more pins to connect the rain board module. In summary, the rain board module detects the rainwater, and the control board module is used to control the sensitivity and compare and convert the analog values to digital values.**



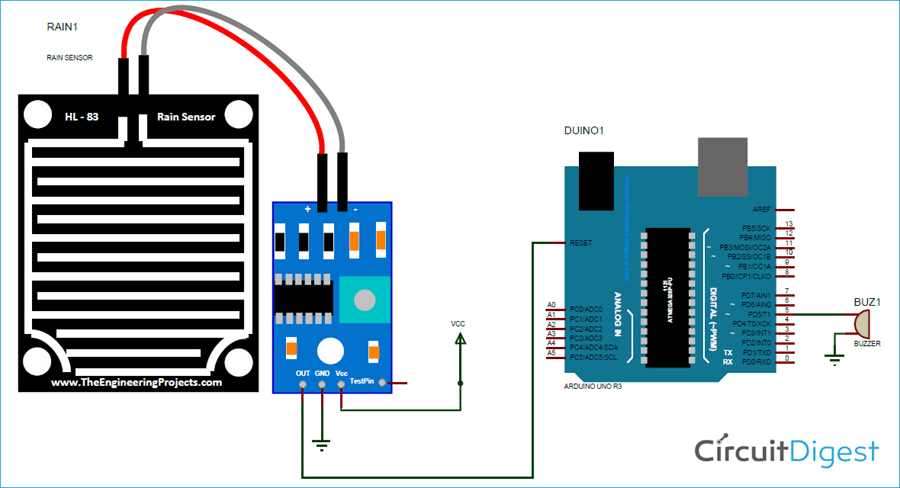
**Working of Rain Sensor**

**Working of the rain sensor module is simple to understand. During a sunny day, due to the dryness on the rain board module, it offers high resistance to the supply voltage. This voltage appears on the output pin of the rain board module as 5V. This 5V is read as 1023 if read by an analog pin of the Arduino.   During rain, the rainwater causes an increase in the wetness on the rain board, which in turn results in the decrease in the resistance offered for the supply. As the resistance decreases gradually, the output voltage starts to decrease.**

**When the rain board is fully wet, and the resistance offered by it is minimum, the output voltage will be as low as possible (approx. 0). This 0V is read as 0 value if read by an analog pin of the Arduino. If the rain board module is partially wet, the output of this rain board module will be with respect to the resistance it offers. If the resistance offered by the rain board module is in such a way that the output is 3V the read analog value will be 613. Formula to find ADC can be given by, ADC = (analog voltage value X 1023)/5. By using this formula you can convert any analog voltage to t Arduino analog read value.**

### ****Circuit Diagram****

**The below circuit diagram shows you the circuit connections for the Rain Drop Sensor with Arduino. The design is done using proteus, the physical modules are similar to the modules which are shown in the circuit diagram.**



**The rain gauge module, which is shown in the circuit diagram is connected to the control board. The control board’s VCC pin is connected to the 5V supply. The ground pin is connected to ground. If needed, the D0 pin is connected to any digital pin of the Arduino, and that pin must be declared as an output pin in the program. The problem we face with the D0 pin is that we can’t get the exact value of the output voltage. If the output crosses the threshold voltage, then the control module can sense the change in the output.  We need to operate the buzzer, even if there is a considerable change in the output voltage in the rain board module. Due to these reasons, the A0 pin is connected to the analog pin of Arduino, which makes monitoring the change in output easy. The buzzer, which is used as a signal to the user, can be connected to any digital pin of the Arduino. If the buzzer needs more than 5V, then try to connect a relay circuit or a transistor and then connecting the load to it.**

### Code Explanation

**The Arduino code for the rain sensor was written using the Arduino IDE. The complete code for this project is given at the end of the page.**

#define rainfall A0

#define buzzer 5

int value;

int set=10;

**Defining pin A0 as rainfall, and pin 5 as a buzzer and declaring variable “value” and “set” as integers and setting its variable set value to 10. This value can be changed according to the required level of operation. If you want the buzzer to activate, even when there is little rain set it to a minimum value**

void setup() {

Serial.begin(9600);

pinMode(buzzer,OUTPUT);

pinMode(rainfall,INPUT);

}

**Initializing the serial communication, and setting the buzzer. Setting the rainfall pin as an output pin and input pin.**

void loop() {

value = analogRead(rainfall);

Serial.println(value);

value = map(value,0,1023,225,0);

**the function analogRead reads the value of the rain sensor. The function map maps the value of the rain sensor from the output pin, and assigns a value to the variable, ranging from 0 to 225.**

if(value>=set){

Serial.println("rain detected");

digitalWrite(buzzer,HIGH);

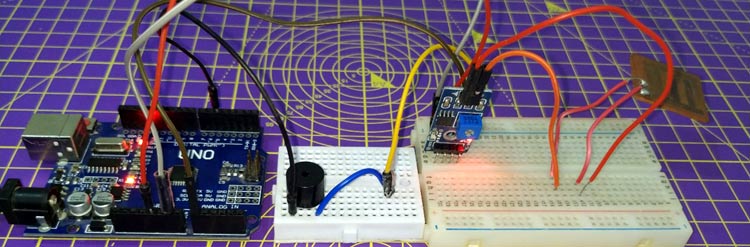
**If the read sensor value is greater than the set value, then the program enters the loop, prints the message on serial monitor and switches on the buzzer**

else{

digitalWrite(buzzer, LOW);

### ****Working of Arduino based Rain Detection system****

**This system works in such a way that, when there is rain, the rainwater acts as a trigger, which switches on the buzzer. In the Rain Drop Sensor Arduino Code**, **we defined that pins 5, and A0 are buzzer and rainfall. By doing this, we can change the pins in the defined part of the function, and the remaining part of the code will be untouched. This will make the programmer in editing the pins easily.**



**In the void loop, the analogRead command reads the value from the sensor. In the next line, the command Serial.println(value), prints the value on the serial monitor. This will be helpful while debugging. The map function maps the incoming value between 0 -225. The function format for the map is a map (value, min value, maximum value, value to be mapped for minimum value, value to be mapped for maximum value). The buzzer will be switched ON or OFF, depending on the set value and the output of the sensor. This value is compared in the if function, with the set value. If the value is greater than the set value, it will switch on the buzzer. If the value is less than the set value, the buzzer will be switched off.**

**The complete working can be found in the video linked below. This is one application among the many, the same principle will be seen in windshield wipers, other home automation, agriculture sectors, etc. Hope you understood the project and enjoyed building something useful. If you have any questions, use the comment section below or use our forums for other technical questions.**

**Code**

#define rainfall A0

#define buzzer 5

int value;

int set=10;

void setup() {

  Serial.begin(9600);

  pinMode(buzzer,OUTPUT);

  pinMode(rainfall,INPUT);

  }

void loop() {

 value = analogRead(rainfall);

 Serial.println("LOL");

 Serial.println(value);

 value = map(value,0,1023,225,0);

 Serial.println(value);

 if(value>=set){

  Serial.println("rain detected");

  digitalWrite(buzzer,HIGH);

 }

 else{

  digitalWrite(buzzer,LOW);

 }

 delay(200);

}