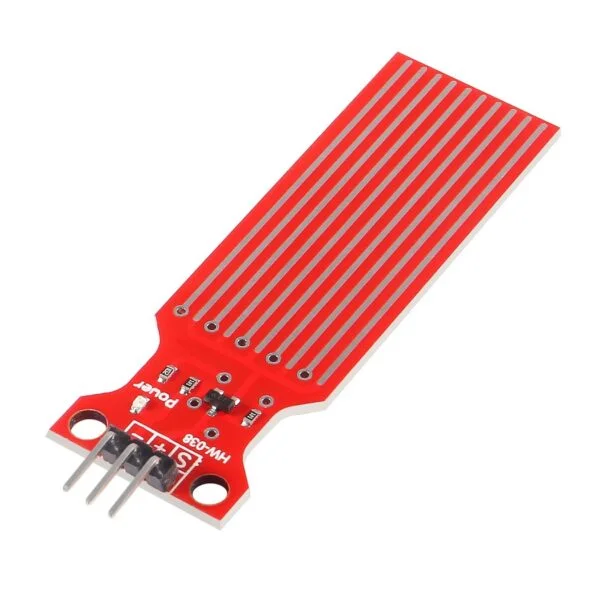
INTRODUCTION

A water level detector module uses metallic probes to measure water conductivity. When water completes the circuit between probes, it triggers an output, such as an LED, buzzer, or relay. Common applications include preventing overflow in tanks and automating pump control. Considerations include corrosion resistance, proper calibration, and power efficiency. Careful adherence to safety standards is crucial during implementation.

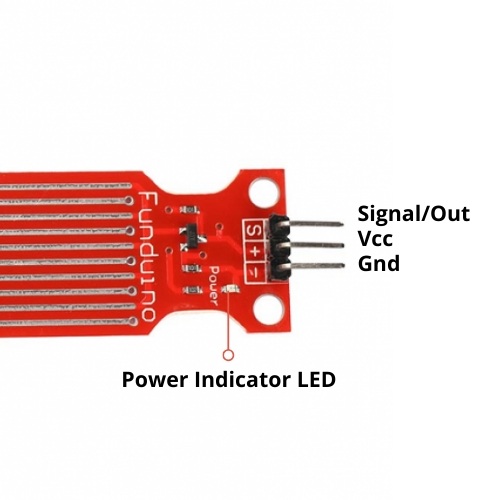




**Specification:**

* Operating Voltage: 3 – 5V
* Operating Current: <20mA
* Sensor Type: Analog
* Detection Area: 40mmX16mm
* Operating Temperature: 10°C - 30°C
* Humidity: 10% - 90% non-condensing
* LED Color: Red

**Water Level Detector Pinout**



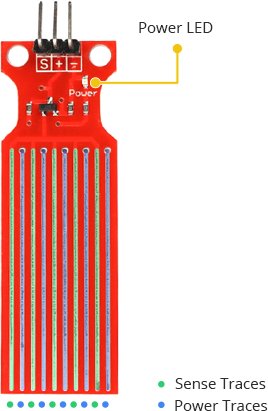
**VCC -** pin provides power to the sensor. It is recommended that the sensor be powered from 3.3V to 5V. Please keep in mind that the analog output will vary depending on the voltage supplied to the sensor.

**GND -** is the ground pin.

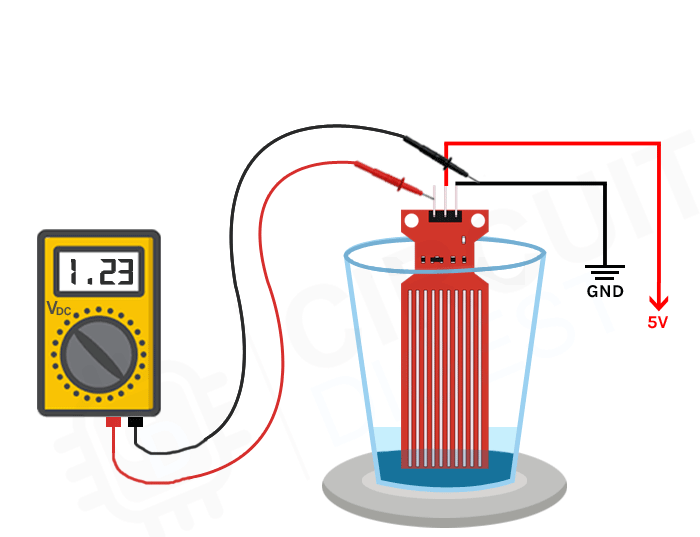
**S (Signal) -** is an analog output pin that will be connected to one of your Arduino’s analog inputs.

**Hardware Overview**

The water level sensor features ten copper traces—five for power and five for sensing—interlaced with one sense trace between each pair of power traces. Normally unconnected, these traces bridge when in water. The board includes a Power LED that illuminates when powered, providing a visual indicator of the sensor's operational status.



**How does a water level sensor work?**

The water level sensor operates by utilizing power and sense traces, creating a variable resistor akin to a potentiometer. The resistance of this circuit varies proportionally with the extent of exposure to water, providing a simple and effective mechanism for water level detection.

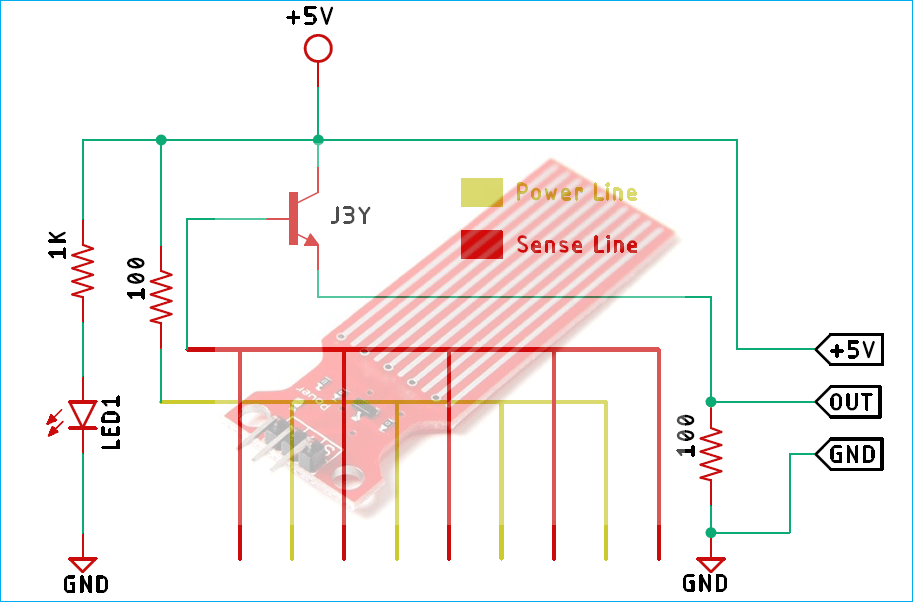
Resistance of the water level sensor varies inversely with immersion depth.

* Greater immersion results in improved conductivity and lower resistance.
* Lesser immersion leads to decreased conductivity and higher resistance.
* The sensor generates an output voltage proportional to the resistance.

Measurement of this voltage allows for accurate determination of water level.

**CIRCUIT DIAGRAM**

The **Circuit Diagram of the Water Level Sensor**is shown below:



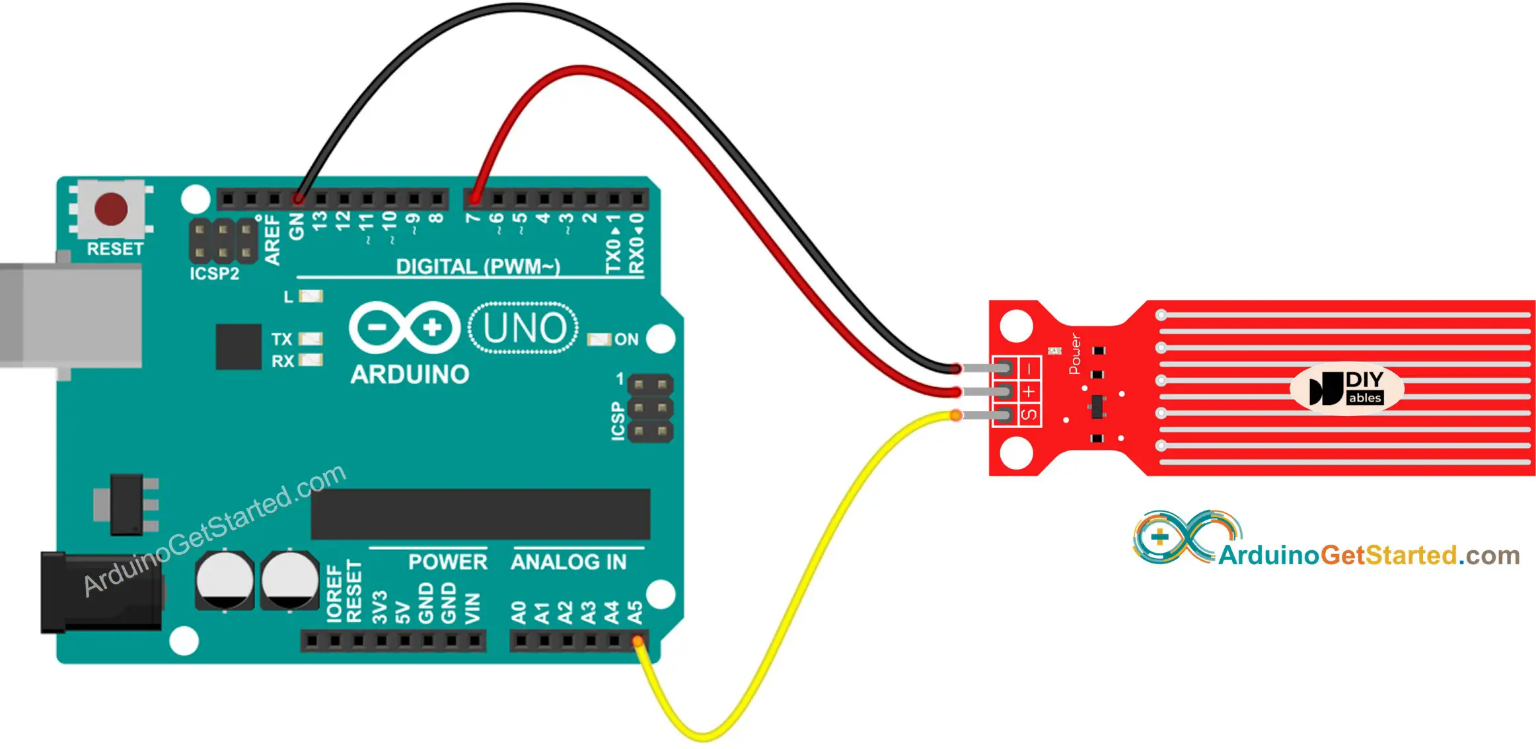
* The collector of the transistor is linked to a 5V supply, while the emitter is grounded through a 100 Ohm resistor.
* In the module, five conducting plates are connected in series with a 100 Ohm resistor to the VCC, and another set of five plates is linked to the base of the NPN transistor.
* When water contacts these plates, current flows from the 5V supply to the transistor's base, causing it to turn on.
* The degree of submersion influences the generated output voltage.
* Increased immersion results in higher output voltage, providing a measure of water level.

**Wiring with Arduino Uno/ Nano**

To initiate the setup, establish a connection by attaching the + (VCC) pin of the sensor module to the 5V port on the Arduino, and link the – (GND) pin to the ground.

A well-recognized concern with these sensors pertains to their limited lifespan due to constant exposure to moisture. Additionally, continuous power application while the sensor is immersed in water significantly hastens the corrosion rate.

To mitigate this issue, it is advised to activate the sensor solely during readings. An effective approach is to connect the sensor's power pin to a digital pin on the Arduino, toggling it between HIGH and LOW states as required. In this case, connect the + (VCC) pin to the Arduino’s digital pin #7.

Lastly, establish a connection by attaching the S (Signal) pin to the Arduino’s A0 ADC pin, enabling the acquisition of water level readings.

**Code:**

// Sensor pins

#define sensorPower 7

#define sensorPin A0

// Value for storing water level

int val = 0;

void setup() {

// Set D7 as an OUTPUT

pinMode(sensorPower, OUTPUT);

// Set to LOW so no power flows through the sensor

digitalWrite(sensorPower, LOW);

Serial.begin(9600);

}

void loop() {

//get the reading from the function below and print it

int level = readSensor();

Serial.print("Water level: ");

Serial.println(level);

delay(1000);

}

//This is a function used to get the reading

int readSensor() {

digitalWrite(sensorPower, HIGH); // Turn the sensor ON

delay(10); // wait 10 milliseconds

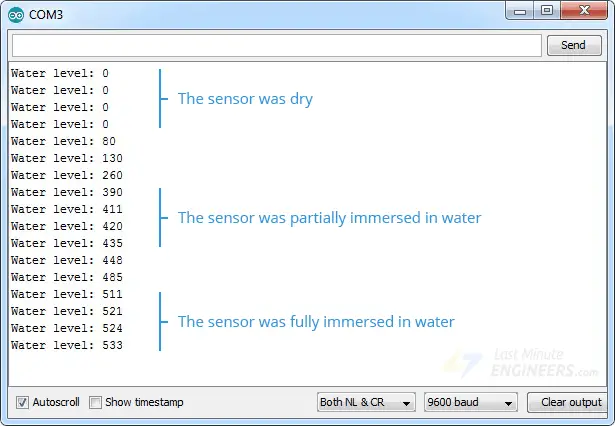
val = analogRead(sensorPin); // Read the analog value form sensor

digitalWrite(sensorPower, LOW); // Turn the sensor OFF

return val; // send current reading

}

After uploading the sketch, the result is given below. If the sensor is not dipped in water and it’s dried the reading is 0. When the sensor is immersed in water, the reading gradually increases.



**Finding the Threshold Values**

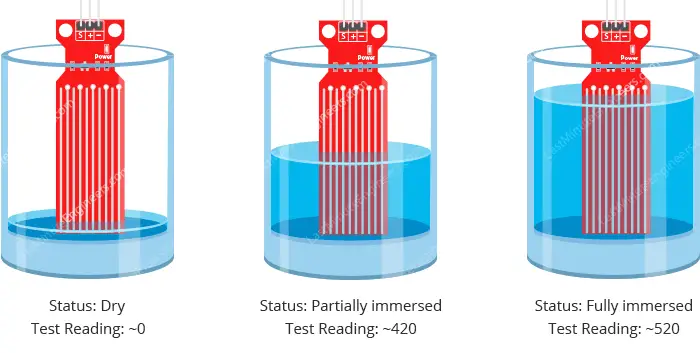
To gauge the water level accurately, note the sensor output values under different conditions: when the sensor is completely dry, partially immersed, and fully immersed in water.

Execute the provided sketch and gather your readings.

Be aware that sensor sensitivity can vary based on the type of water utilized. Pure water, being non-conductive, relies on minerals and impurities for conductivity. Consider this variability when interpreting your sensor readings.

When you run the sketch, you should see readings similar to the ones below:

* when the sensor is dry (0)
* when the sensor is partially immersed in water (~420)
* when the sensor is fully immersed in water (~520)



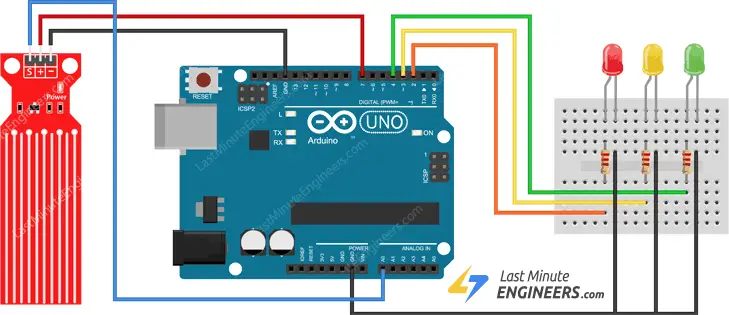
This test may require some trial and error. Once you have the readings, you can use them as a threshold to trigger an action.

**Water level indicator project**

We’ll reuse the previous example’s circuit. We only need to add some LEDs this time.

Connect three LEDs to digital pins #2, #3, and #4 using 220 ohm resistors.

Connect your circuit as shown below:



Code:

/\* Change these values based on your calibration values \*/

int lowerThreshold = 420;

int upperThreshold = 520;

// Sensor pins

#define sensorPower 7

#define sensorPin A0

// Value for storing water level

int val = 0;

// Declare pins to which LEDs are connected

int redLED = 2;

int yellowLED = 3;

int greenLED = 4;

void setup() {

Serial.begin(9600);

pinMode(sensorPower, OUTPUT);

digitalWrite(sensorPower, LOW);

// Set LED pins as an OUTPUT

pinMode(redLED, OUTPUT);

pinMode(yellowLED, OUTPUT);

pinMode(greenLED, OUTPUT);

// Initially turn off all LEDs

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

}

void loop () {

int level = readSensor();

if (level == 0) {

Serial.println("Water Level: Empty");

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

}

else if (level > 0 && level <= lowerThreshold) {

Serial.println("Water Level: Low");

digitalWrite(redLED, HIGH);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

}

else if (level > lowerThreshold && level <= upperThreshold) {

Serial.println("Water Level: Medium");

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, HIGH);

digitalWrite(greenLED, LOW);

}

else if (level > upperThreshold) {

Serial.println("Water Level: High");

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, HIGH);

}

delay(1000);

}//This is a function used to get the reading

int readSensor() {

digitalWrite(sensorPower, HIGH);

delay(10);

val = analogRead(sensorPin);

digitalWrite(sensorPower, LOW);

return val; }