Al-Mustaqbal University

College Of Engineering & Technology

Department of Computer Engineering Techniques

(Stage: 3)

Digital Control

Lecture 12

Arduino programming

Dr.: Fanar Ali Joda

Water Level Indicator: Interfacing water level sensor With Arduino

If you've ever had a water heater explode or attempted to build submersible electronics, you know how important it is to detect the presence of water.

That is exactly what this Water Level Sensor allows you to do! It can measure the water level, monitor a sump pit, detect rainfall, and detect leaks.

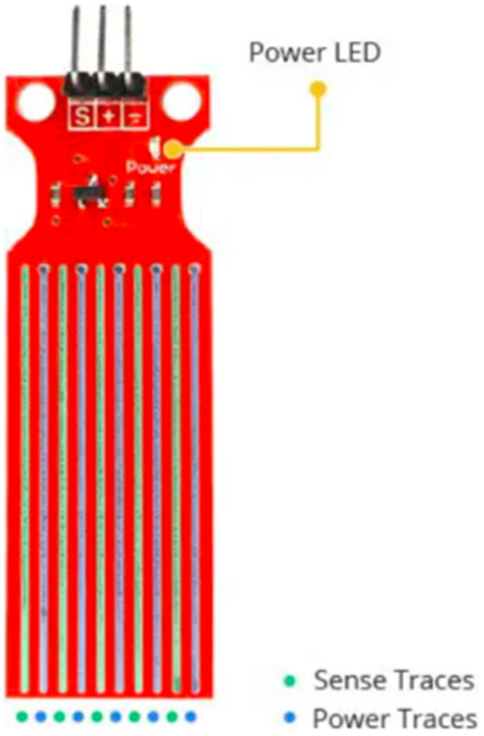
Hardware Overview

The sensor has ten exposed copper traces, five of which are power traces and the remaining five are sense traces. These traces are interlaced so that there is one sense trace between every two power traces.

Normally, power and sense traces are not connected, but when immersed in water, they are bridged.

How Does a Water Level Sensor Work?

The operation of the water level sensor is fairly simple. The power and sense traces form a variable resistor (much like a potentiometer) whose resistance varies based on how much they are exposed to water.



This resistance varies inversely with the depth of immersion of the sensor in water:

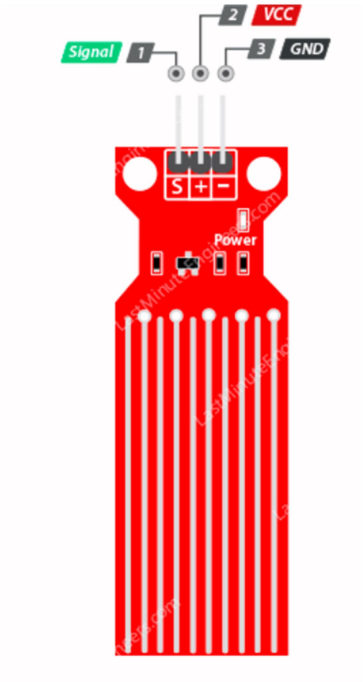
The more water the sensor is immersed in, the better the conductivity and the lower the resistance.

The less water the sensor is immersed in, the poorer the conductivity and the higher the resistance.

The sensor generates an output voltage proportional to the resistance; by measuring this voltage, the water level can be determined.

Water Level Sensor Pinout

The water level sensor is extremely simple to use and only requires three pins to connect.

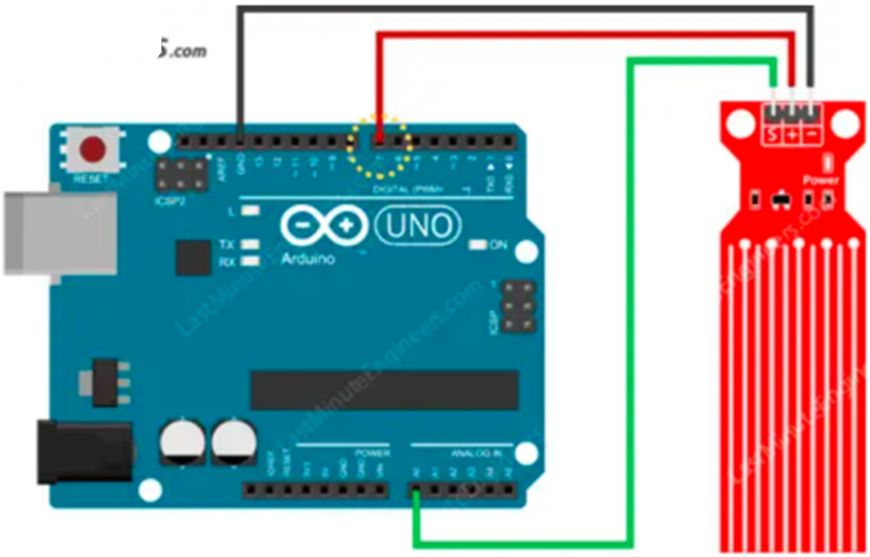


S (Signal) is an analog output pin that will be connected to one of your Arduino's analog inputs.

+ (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.

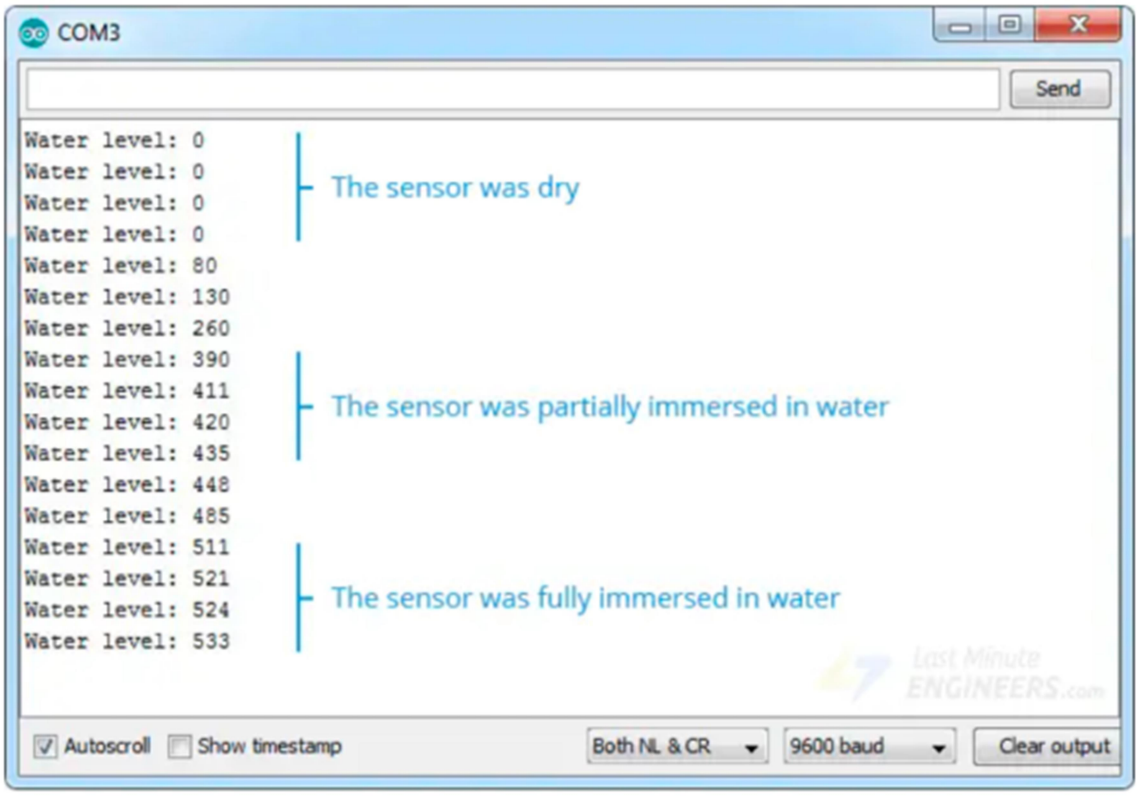
The wiring is shown in the image below.




```
//
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
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
Serial.print("Water level: ");
Serial.println(val);
```

```
delay(1000);
}
```

After uploading the sketch, open the Serial Monitor window to view the output. When the sensor is dry, it will output a value of 0; however, as the sensor is immersed in water, the output will gradually increase.



Finding the threshold values

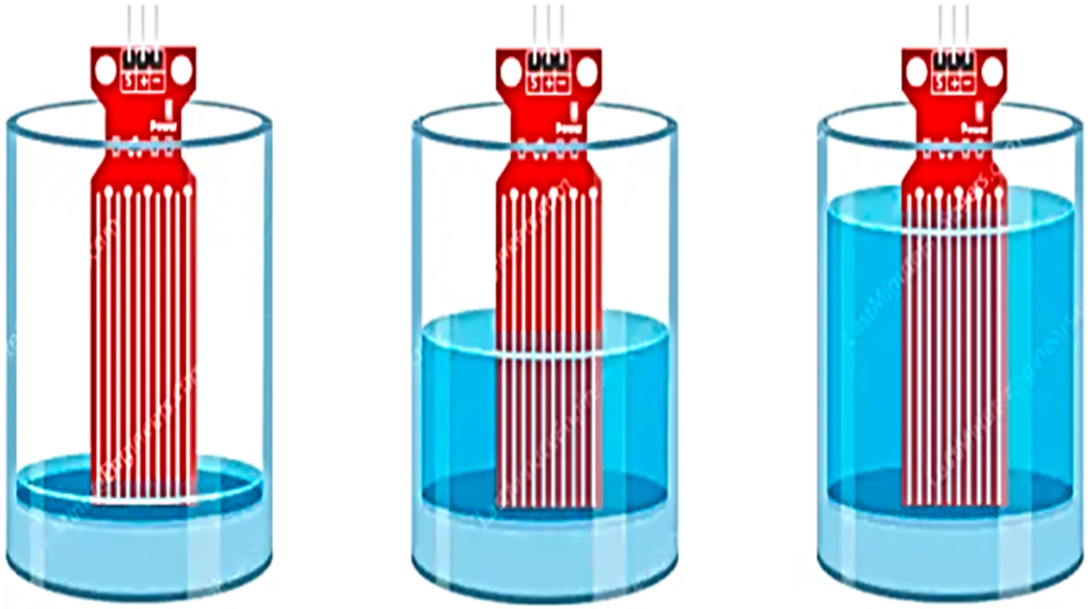
To estimate the water level, record the values of your sensor output when the sensor is completely dry, partially immersed, and fully immersed in water.

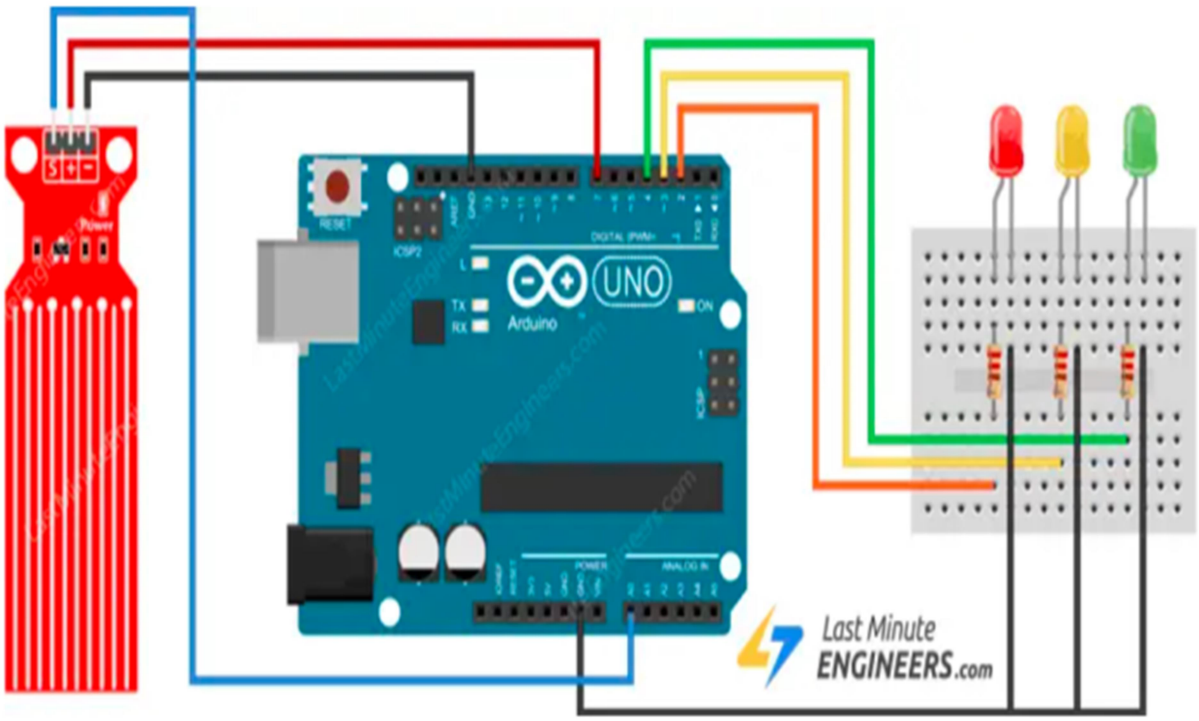
Simply run the above sketch and take your readings.

Keep in mind that your sensor may be more or less sensitive depending on the type of water you use. As you may know, pure water is not conductive; it is the minerals and impurities in water that make it conductive.

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)





Arduino Code

After constructing the circuit, upload the following sketch to your Arduino. This sketch defines two variables: `lowerThreshold` and `upperThreshold`. These variables represent our thresholds.

If the output falls below the lower threshold, the red LED will light up, if it rises above the upper threshold, the green LED will light up, and if it falls somewhere in the middle, the yellow LED will light up.

```
/* 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()
 {
      digitalWrite(sensorPower, HIGH);
      delay(10);
      level = analogRead(sensorPin);
      digitalWrite(sensorPower, LOW);

      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);
}
```