



EAST WEST UNIVERSITY

Water Level Detection Using Arduino Uno

Submitted by

Name: Nushrat Jaben Aurnima

Student Id: 2022-2-60-146

Submitted to

Dr. Raihan Ul Islam

***Associate Professor, Department of Computer Science
& Engineering***

Submitted on

01/07/2025

Table of Contents

Introduction	2
Components	2
Circuit Diagram	2
Working Principle	3
Code Explanation	3
Results.....	4
References	4

Introduction

In today's lab, our group tried to build a basic water level sensor system with an Arduino Uno and a water level sensor. The system will monitor and indicate different water levels by using three LEDs – red, yellow and green. The water level status can be seen in the Serial Monitor for real-time monitoring.

This system is used in many real-world applications like tracking overhead water tanks, flood detection, automated irrigation and so on. The objective was to gain hands-on experience using analog sensors and learn about controlling outputs (LEDs) based on sensor values.

Components

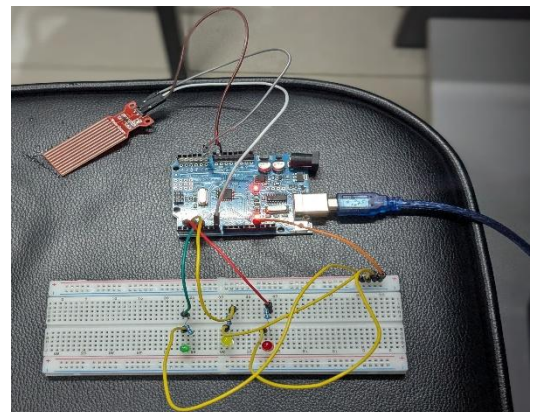
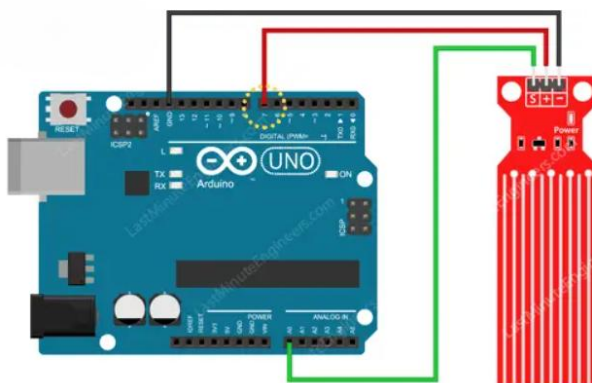
Hardware:

- Arduino Uno
- Water Level Sensor
- Breadboard
- Resistors (220 ohm)
- Jumper Wires
- Red, Yellow, Green LEDs
- USB Cable

Software:

- **Arduino IDE** - <https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE>

Circuit Diagram



From the Diagram we can observe:

- Signal pin is connected to A0.
- LEDs are connected to pins 2, 3, and 4.
- Sensor power is controlled using digital pin 7.

Working Principle

The water level sensor provides analog output based on the amount of it that is in the water. Arduino reads this analog, and based on its value, turns on one of three LEDs:

- **Red LED:** Low water level
- **Yellow LED:** Medium water level
- **Green LED:** High water level

If there's no water, all LEDs remain off.

Code Explanation

Some notable parts of the code –

```
#define sensorPower 7  
#define sensorPin A0
```

- `sensorPower` is used to supply power to the sensor using Arduino. This saves power by turning off the sensor when not in use.
- `sensorPin` is where the sensor is being read (analog).

```
int lowerThreshold = 200;  
int upperThreshold = 400;
```

- These values define the water levels as considered low, medium, or high. Changeable based on sensor's performance and calibration.

```
int level = readSensor();
```

- It calls a custom function `readSensor()` that powers the sensor briefly, reads the analog value, then powers it off.

```
if (level == 0) { ... }  
else if (level > 0 && level <=  
lowerThreshold) { ... }  
else if (level > lowerThreshold &&  
level <= upperThreshold) { ... }
```

- This checks the water level and turns on the LED according to the value.

```
int readSensor() {  
    digitalWrite(sensorPower, HIGH);  
    delay(10);  
    val = analogRead(sensorPin);  
    digitalWrite(sensorPower, LOW);  
}
```

- This function powers the sensor only when needed, reads the data, and turns it off again.

Results

- The system successfully detected four water levels: Low, Medium, and High.
- The Serial Monitor correctly printed the water level.
- The LEDs switched as expected when the water level changed as seen in the video attached.

References

1. [Last Minute Engineers: Water Level Sensor with Arduino](#)
2. [Arduino Project Hub: Water Level Monitor](#)