INTERNATIONAL JOURNAL OF APPLIED ELECTRONICS (IJAE)

Journal ID: 8550-6884

https://iaeme.com/Home/journal/IJAE

High Quality Refereed Peer Reviewed International journal in Industrial Engineering

Published By

IAEME Publication

Plot: 03, Flat S1, Poomalai Santosh Pearls Apartment,
Plot No.10, Vaiko Salai 6Th Street, Jai Shankar Nagar, Palavakkam,
Chennai-600041, Tamilnadu, India
Email: editor@iaeme.com, iaemedu@gmail.com

International Journal of Applied Electronics (IJAE)
Volume 6, Issue 1, January-June 2025, pp. 16-28, Article ID: IJAE_06_01_003
Available online at https://iaeme.com/Home/issue/IJAE?Volume=6&Issue=1
ISSN Online : 2295-5177, Journal ID : 8550-6884
Impact Factor (2025): 8.25 (Based on Google Scholar Citation)
DOI: https://doi.org/10.34218/IJAE_06_01_003

ADJUSTABLE WATER LEVEL SENSOR TO AUTOMATE CONTROL OF FILTRATION SYSTEMS

Ritansh R. Chauhan [1]

[1] Student, 12th grade, The Brigade School @ Whitefield, Bangalore, Karnataka, India.

ABSTRACT

Drinking water is one of the most important resources, making its efficient management a vital necessity in not just workplaces and hospitality venues but also in every single household. In Indian homes, water filtration systems ensure clean and safe drinking water, yet they have one disadvantage -they require frequent manual maintenance for day-to-day functioning. Inefficiencies such as water wastage, overflows, and dry tank scenarios can disrupt filtration performance and impact pump functionality. Despite advancements in technology, accessible and cost-effective automation solutions remain limited. This project aims to solve such challenges by developing an automated water level sensing and control system using Arduino-UNO. The system integrates a control module to monitor and regulate water levels dynamically, optimizing filtration efficiency while minimizing tedious and costly maintenance services. This paper details the system's design, implementation, functionality, and demonstrates its potential to enhance the system.

Keywords: Arduino Uno, Hysteresis, Home automation, Relay switch, Sensor-based control, Smart water management, Water conservation.

Cite this Article: Ritansh R. Chauhan. (2025). Adjustable Water Level Sensor to Automate Control of Filtration Systems. *International Journal of Applied Electronics (IJAE),* 6(1), 16–28.
https://iaeme.com/MasterAdmin/Journal_uploads/IJAE/VOLUME_6_ISSUE_1/IJAE_06_01_003.pdf

1. INTRODUCTION

Overflowing storage tanks, excessive filtration cycles, and dry tank conditions are common problems that affect water filtration efficiency. When water levels are not adequately monitored, filtration systems may operate unnecessarily, leading to higher electricity consumption and a reduction in filter lifespan. Additionally, in regions where water scarcity is a growing concern, these inefficiencies contribute to unnecessary water loss. The solution proposed in this project aims to solve these challenges by automating water level sensing/control using Arduino UNO and sensors to monitor the water filtration process in real time. The following sections provide an overview of the system's design and implementation. The project explores sensor calibration, microcontroller integration, and relay-based automation to achieve a seamless water management system. Graphs and illustrations further highlight the necessity of automation and demonstrate the system's efficiency.

1.1. Limitations of Conventional Water level Monitoring

Water level monitoring in household RO (Reverse Osmosis) water filters have limitations that affect their practicality and efficiency. One major issue is the limited auto turn-off functionality at partial fill levels. Most water filter systems only turn off when the tank is completely full. While this prevents overflows, it does not account for situations that do not require the tank to be filled to its maximum capacity. When fewer people are home, the water demand is lower, and continuously filling the tank to the top may be unnecessary. A system that can stop at customizable levels conserves both waters, making it a more efficient choice. Solutions exist, but they either don't enable the customizable level [2] or are too bulky for a household RO filter [3].

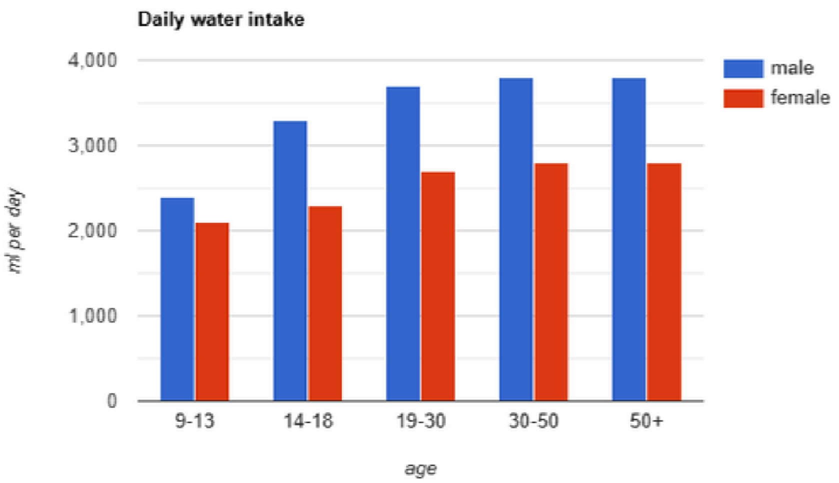


Figure 1: Daily Water Intake

Another drawback is the lack of an auto turn-on feature when the tank is empty. In busy households, water consumption is often high during peak hours, such as mornings. If the water runs out unexpectedly, users have to manually start the pump, which disrupts daily routines and makes it highly inconvenient. An automated system ensures the tank refills promptly, providing a continuous water supply without human intervention. Fixed threshold levels further add to the inefficiency, as they do not allow customization based on varying household needs (Fig 1) [1]. A home with just one or two occupants may not require the tank to be filled completely every day (Fig 2). A flexible system enables users to set different water levels depending on their requirements, reducing water waste and pump wear while prolonging the life of the filtration system.

Figure 2: Daily Water Intake Vs Family Size

2. DESIGN COMPONENTS

2.1. Arduino Uno

The Arduino Uno [4] serves as the central controller of the system. It processes inputs from the water level sensors and controls the relay module based on predefined thresholds. The microcontroller is programmed to automate the water pump's operation, ensuring a seamless and autonomous process.

Figure 3: Arduino Uno

Figure 4: Battery Power Supply

2.2. Power Supply

A stable power source is essential to operate the Arduino, sensors, and relay module. A 9V alkaline battery is used which provides portability and replaceability to the solution module.

Figure 5: Relay Module

Figure 6: Water Level Sensor (XKC-Y25)

2.3. Relay Module

The relay module functions as an electronic switch that controls the water pump's power supply. Based on sensor readings, the Arduino sends signals to the relay to turn the pump ON or OFF.

2.4. Water Level Sensors (x2)

Two contactless water level sensors [5] are used to continuously monitor the tank's water levels:

- Lower Sensor: Detects when the water level drops below the minimum set threshold, signalling the Arduino to activate the water pump.
- Upper Sensor: Detects when the water reaches the maximum set level, prompting the Arduino to deactivate the pump to prevent overfilling.

It is to be noted that sensors can be attached to the outside of a tank or container (opaque or transparent) to measure the inside level of liquids.

Figure 7: DC Power Adaptor Jack

2.5. Power supply connectors

A power connector is used to link the main AC power supply to the RO system through a relay switch. It ensures a safe and stable electrical connection, allowing the relay to control the RO unit based on water level conditions.

3. WORKING OF HARDWARE

The solution consists of two non-contact capacitive water level sensors (XKC-Y25), an Arduino Uno microcontroller, and a relay module. These solutions automatically control the RO water filter based on adjustable water levels set on the storage tank and Material which are used is presented in this section.

3.1. Circuit diagram and operation

Fig 8 shows the circuit implementation of the proposed solution.

Sensor Placement: The lower sensor detects if the tank is nearly empty. Each sensor output is an analog signal depending on whether water is present near it. These values are read by the Arduino's analog input pins (A0 and A1). When the signal is above a certain threshold (e.g., 500 points), the sensor is considered to be detecting water.

Relay Function: The relay module acts as an electronic switch. It is connected to the digital pin(e.g., D7) of the Arduino. When activated, it switches the pump ON or OFF based on the logic above.

Power Supply: The Arduino and sensors are powered through a 5V DC source, while the relay controls the higher voltage circuit powering the water pump.

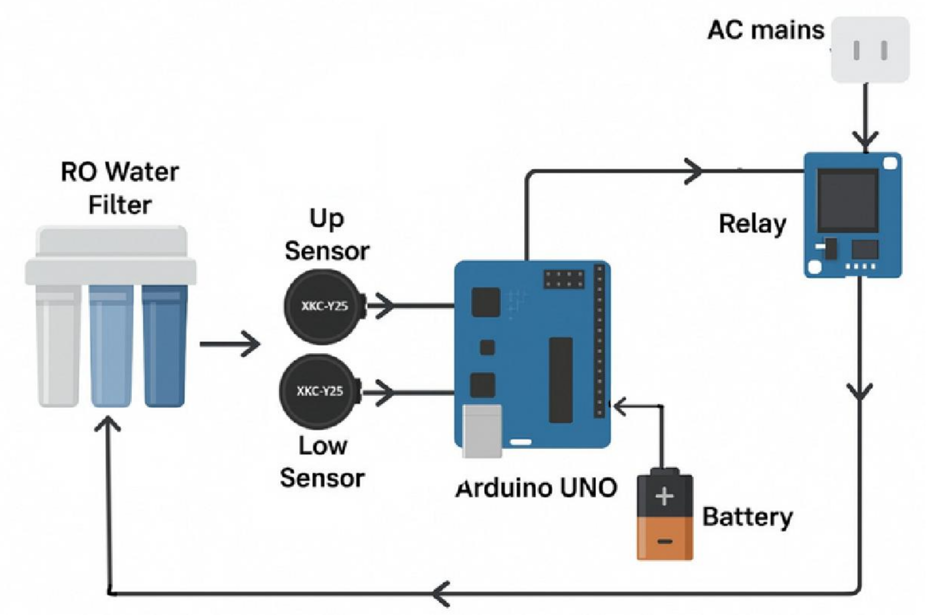


Figure 8: Circuit Diagram.

3.2. Lab setup and bench validation

The proposed solution is implemented and validated in the lab before attaching and utilizing it with a household RO water filter. Glass water jug is used as a prototype for a water tank to validate the operation of upper/lower water level sensors, software code and power relay. (Fig 9). These sensors work equally well with opaque water containers also, where water level is not physically visible.

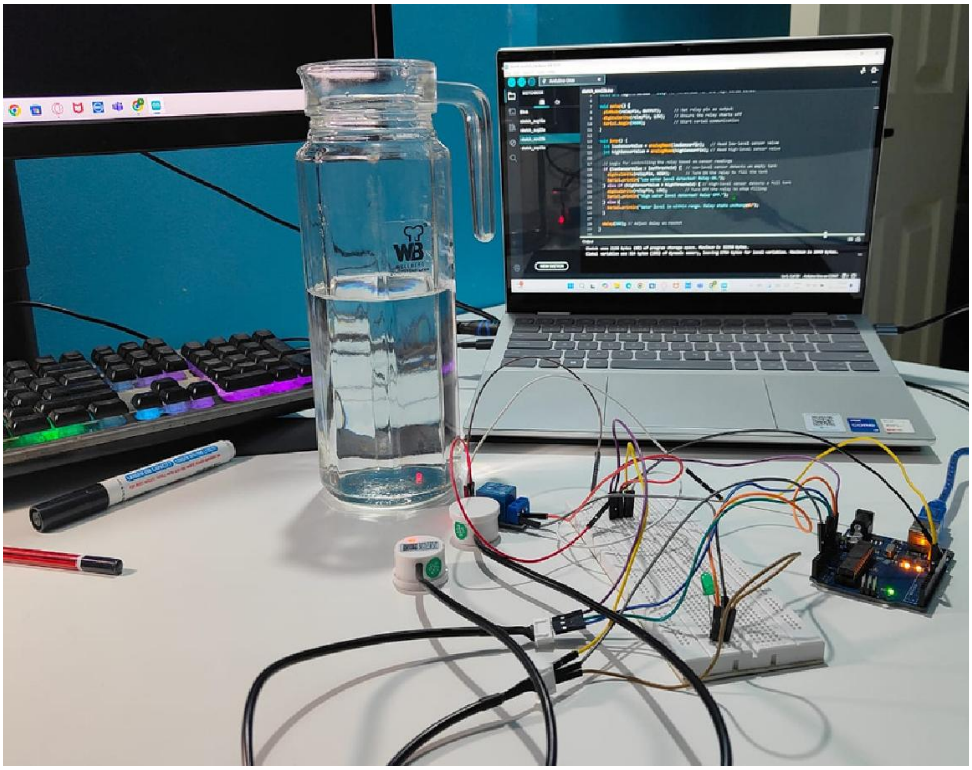
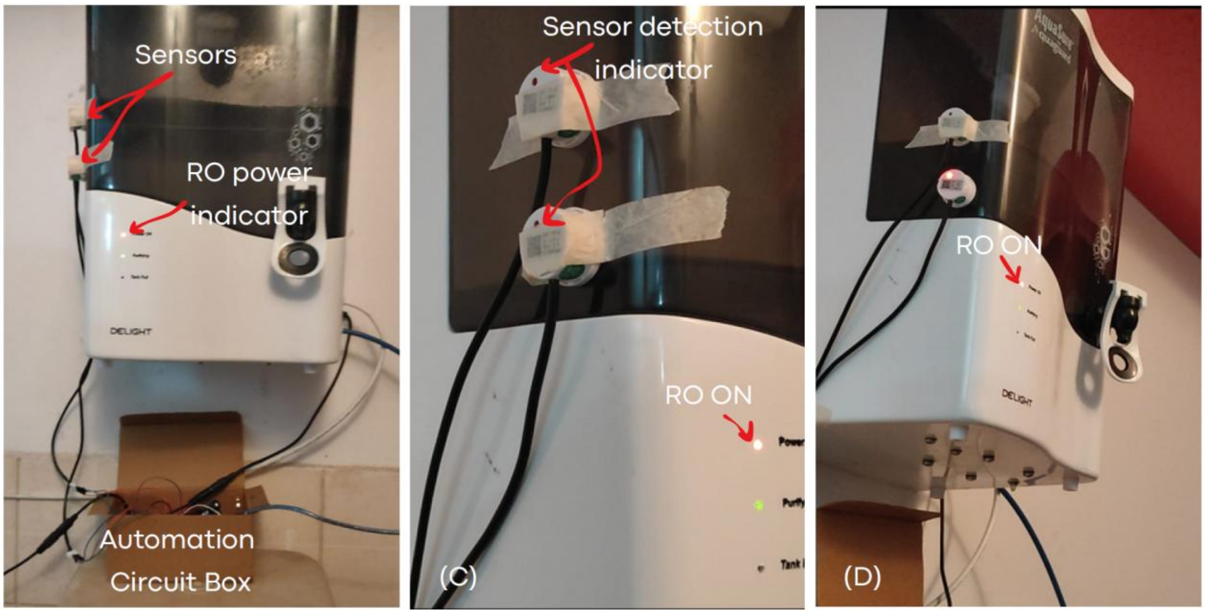


Figure 9: Bench Validation Setup

3.3. Field deployment

Once proven on a bench, this solution is attached with an RO water filter to automatically turn ON/OFF of the water flow at desired threshold. Fig 10: (A to F) shows the setup and state of RO water-filter for different stages of water levels.



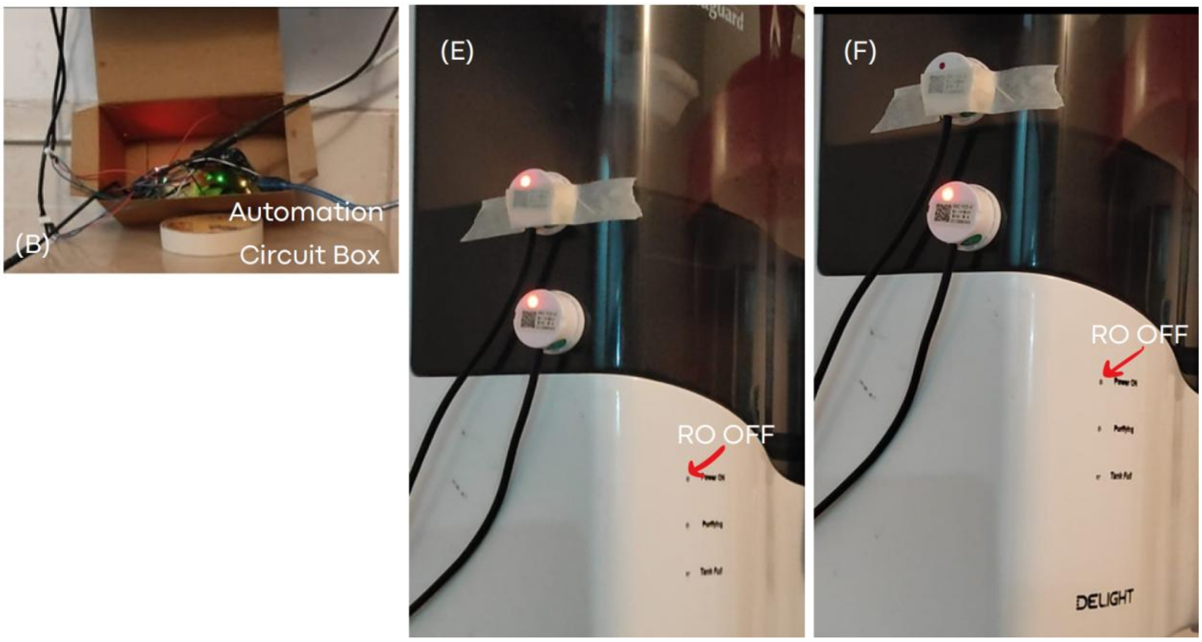


Figure 10: Implementation On RO Water Filter

(A,B) Automation circuit box + sensors attached to RO water filter. Circuit includes relay, Arduino uno, connectors, LED indicators.

(C) Water level below lower sensor, RO is ON.

(D) Low to high transition, water level between lower and upper sensor. RO is ON

(E) Water level below lower sensor, RO is turned OFF.

(F) High to Low transition, water level between lower and upper sensor. RO stays OFF.

4. SOFTWARE CODE FUNCTIONING

The system begins by reading analog voltage values from two sensors placed at different levels on the tank. These values are compared against a fixed threshold to determine if water is present at that level.

```
int lowerSensorValue = analogRead(lowerSensorPin);
int upperSensorValue = analogRead(upperSensorPin);

bool lowerDetected = lowerSensorValue > threshold;
bool upperDetected = upperSensorValue > threshold;
```

4.1. Low-to-High and high-to-low Transitions

If both sensors are ‘not-detecting’, the tank is nearly empty, and the relay is turned ON to start filling the tank. As the water level rises above the lower sensor only, the relay remains ON.

When water level rises further and both sensors start ‘detecting’ water, the tank is full, and the relay is turned OFF. Even after water starts getting used and the upper sensor stops ‘detecting’, the system keeps the relay OFF until the water level is below the lower sensor.

```
if (lowerDetected && upperDetected ) {

    digitalWrite(relayPin, HIGH); // Turn OFF relay
    Serial.println("High to Low: Both sensors detecting. Relay OFF.");
    lastLowerDetected = true;
    lastUpperDetected = true;
  }

  else if (!lowerDetected && !upperDetected) {
    digitalWrite(relayPin, LOW); // Turn ON relay
    Serial.println("Low to High: Both sensors not detecting. Relay ON.");
    lastLowerDetected = false;
    lastUpperDetected = false;
    // Low to High Transition Logic
  }
```

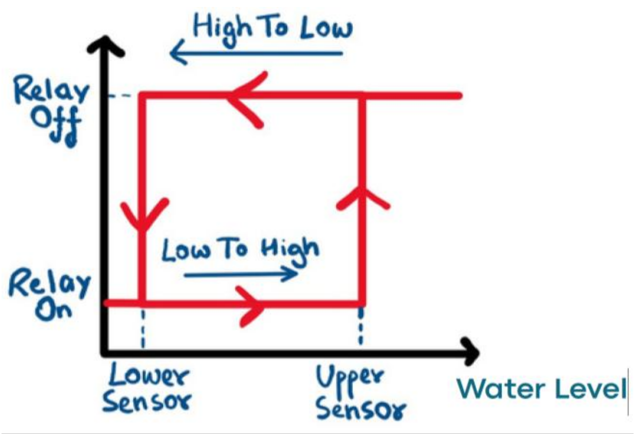
Transition Detection & Hysteresis Logic

To distinguish between rising and falling water levels and define the valid state of the relay, the code uses variables to remember the last known state of each sensor. This introduces hysteresis into the system, allowing it to define a different state (ON or OFF) of relay between rising and falling water level transition, while sensors read the same for level between upper and lower sensor.

```
bool lastLowerDetected = false;
bool lastUpperDetected = false;
```

```
else if (lowerDetected && !upperDetected && lastLowerDetected==false) {
  digitalWrite(relayPin, LOW); // Turn ON relay
  Serial.println("Low to High: Lower sensor detecting, upper not. Relay ON.");
  lastLowerDetected = true;
  lastUpperDetected = false;
}

else if (lowerDetected && !upperDetected && lastUpperDetected==true) {
  digitalWrite(relayPin, HIGH); // Turn OFF relay
  Serial.println("High to Low: Lower sensor detecting, upper not. Relay OFF.");
  lastLowerDetected = true;
  lastUpperDetected = false;
}
```



4.3. Delay to Prevent Frequent Switching

A delay is added to give enough time between readings, ensuring water level changes are stable before making relay decisions.

```
delay(5000);
```

Final Code

```
const int lowerSensorPin = A0; // Analog pin connected to the lower sensor
const int upperSensorPin = A1; // Analog pin connected to the upper sensor
const int relayPin = 7;       // Digital pin connected to the relay
const int threshold = 100;    // Adjust threshold as needed (0-1023 range)

bool lastLowerDetected = false;
bool lastUpperDetected = false;

void setup() {
  pinMode(relayPin, OUTPUT); // Set relay pin as output
  digitalWrite(relayPin, LOW); // Ensure the relay starts off
  Serial.begin(9600);
}

void loop() {
  int lowerSensorValue = analogRead(lowerSensorPin); // Read lower sensor value
  int upperSensorValue = analogRead(upperSensorPin); // Read upper sensor value

  bool lowerDetected = lowerSensorValue > threshold;
  bool upperDetected = upperSensorValue > threshold;

  // Determine if transition is high-to-low or low-to-high
  if (lowerDetected && upperDetected ) {

      digitalWrite(relayPin, HIGH); // Turn ON relay
      Serial.println("High to Low: Both sensors detecting. Relay OFF.");
      lastLowerDetected = true;
      lastUpperDetected = true;
    }

     else if (!lowerDetected && !upperDetected) {
      digitalWrite(relayPin, LOW); // Turn ON relay
      Serial.println("Low to High: Both sensors not detecting. Relay ON.");
      lastLowerDetected = false;
      lastUpperDetected = false;
      // Low to High Transition Logic
    }
    else if (lowerDetected && !upperDetected && lastLowerDetected==false) {
      digitalWrite(relayPin, LOW); // Turn ON relay
      Serial.println("Low to High: Lower sensor detecting, upper not. Relay ON.");
      lastLowerDetected = true;
      lastUpperDetected = false;
    }

     else if (lowerDetected && !upperDetected && lastUpperDetected==true) {
      digitalWrite(relayPin, HIGH); // Turn OFF relay
      Serial.println("High to Low: Lower sensor detecting, upper not. Relay OFF.");
      lastLowerDetected = true;
      lastUpperDetected = false;
     }



   // Update the last state variables
   lastLowerDetected = lowerDetected;
   lastUpperDetected = upperDetected;

   delay(5000); // Adjust delay as needed
 }
```

5. CONCLUSION

This project demonstrates implementation of an automated water level controller. It saves up to 250L/month of drinking water for a single person using a 12L water tank RO. It also ensures a seamless drinking water supply without requiring human involvement, helping to save time in a busy schedule. The system worked as intended, accurately detecting water levels and controlling the relay based on the designed logic. The use of hysteresis added stability to the operation and prevented unnecessary switching.

6. ACKNOWLEDGEMENTS

I would like to sincerely thank my mentors and school principal for their guidance, my parents for their constant support, and my friends for their encouragement throughout the course of this project. Their help and motivation played a key role in its successful completion.

REFERENCES

[1] Daily Water intake. https://www.drlogy.com/calculator/daily-water-intake, https://www.betterhealth.vic.gov.au/health/healthyliving/water-a-vital-nutrient [Accessed: 2-March-2025].

[2] Ms. Pooja K, Automatic Water Level Indicator and Controller Using Arduino, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 05 | May 2020.

[3] Ms. Sarabjeet K., Mrs. Linkan P., Automatic Level Control for Cascaded Water Tanks using PLC, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 04 | Apr-2018.

[4] Arduino, Arduino IDE Software, [Online]. Available: https://www.arduino.cc/en/software [Accessed: 13-March-2025].

[5] XKC-Y25 Sensor Datasheet. Available: https://www.application-datasheet.com/pdf/dfrobot/sen0204.pdf [Accessed: 15-March-2025].

Citation: Ritansh R. Chauhan. (2025). Adjustable Water Level Sensor to Automate Control of Filtration Systems. International Journal of Applied Electronics (IJAE), 6(1), 16–28.

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJAE/VOLUME_6_ISSUE_1/IJAE_06_01_003.pdf

Abstract Link:
https://iaeme.com/Home/article_id/IJAE_06_01_003

✉ **editor@iaeme.com**