Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design, Implementation, and Modeling of an Autonomous Closed-Loop Water Level Controller with Dry-Run Protection

Khan Burhan Uddin
ukhanbourhan@gmail.com
City University Bangladesh
Hasibul Islam
City University Bangladesh
Tanjimul Islam Sajit
City University Bangladesh
Touhid Sagor Khan
City University Bangladesh

Article

Keywords: Closed-loop control, automated water pump, dry-run protection, resource conservation, level monitoring

Posted Date: August 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10532679/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Design, Implementation, and Modeling of an Autonomous Closed-Loop Water Level Controller with Dry-Run Protection

Submitted To :

Md Shayan Rana
Assistant Professor
Department of EEE
City University

Submitted By :

Khan Burhan Uddin, Hasibul Islam, Tanjimul Islam Sajit, Touhid Sagor Khan

Department of Electrical and Electronics Engineering

City University, Dhaka, Bangladesh

Funding: No funding was received for conducting this study.

ABSTRACT

Manually monitoring fluid levels in overhead and storage tanks frequently introduces systemic operational inefficiencies, primarily water wastage due to overflow and premature pump failure resulting from dry running. This project presents the design and implementation of an automated, closed-loop control system engineered to continuously monitor critical high and low water thresholds. Utilizing a sensor-driven feedback loop, the system automatically activates the water pump when the volume drops below a predetermined lower threshold and instantly terminates operation once maximum capacity is reached. By eliminating the reliance on human intervention, the proposed solution significantly optimizes electrical energy consumption, prevents valuable water resource depletion, and safeguards pumping infrastructure from mechanical wear. Experimental results demonstrate that the system offers a highly reliable, cost-effective, and sustainable approach to modern residential and industrial water management.

Keywords: Closed-loop control, automated water pump, dry-run protection, resource conservation, level monitoring.

TABLE OF CONTENTS

Certificate ii
Declaration iii
Acknowledgement iv
Abstract v
List of Figures vii
List of Tables viii
Chapter 1 : Introduction 8
Chapter 2 : Literature Survey 12
Chapter 3 : Existing System & Proposed System 15
Chapter 4 : System Requirements 18
Chapter 5 : Hardware Component Description 20
Chapter 6 : System Design 26
Chapter 7 : Working Principle & Methodology 29
Chapter 8 : Testing and Result 33
Chapter 9 : Advantages, Application and Limitations 36
Chapter 10 : Cost Analysis 38
Chapter 11 : Future Scope 39
Chapter 12 : Conclusion 40
References 41
Appendix A : Complete Arduino Source Code 42

LIST OF FIGURES

Figure No.	Title	Page
Fig. 5.1	Arduino UNO R3	20
Fig. 5.2	DC Water Pump	22
Fig. 6.1	Block Diagram of Automatic Water Pump Controller	26
Fig. 6.2	Circuit Diagram	27
Fig. 5.4	Breadboard layout and internal strip connections	27
Fig. 6.1	Block diagram of the automatic water level controller	30
Fig. 6.2	Circuit / pin connection diagram	32

LIST OF TABLES

Table No.	Title	Page
Table 2.1	Summary of literature survey	13
Table 3.1	SWOT Analysis of proposed System	17
Table 4.1	Hardware Requirement Specification	17
Table 5.1	Arduino UNO technical specifications	20
Table 6.1	Pin Connection Mapping	28
Table 10.1	Bill of Materials and Approximate Cost	38

CHAPTER 1

INTRODUCTION

1.1 Background

Water is one of the most vital natural resources, essential for domestic, agricultural, and industrial activities. In many modern residential and commercial buildings, water supply relies on overhead or storage tanks filled via motorized electric pumps. Efficiently managing these storage systems requires constant oversight to ensure that water is available whenever needed.

Traditionally, managing water levels in these tanks has been a manual process. An individual must physically inspect the water level or wait until a tank dries up or overflows before operating the pump switch. While straightforward in theory, this manual method suffers from major practical drawbacks. It requires constant human vigilance, relies heavily on guesswork, and introduces significant operational delays. In today's fast-paced environment, relying on manual human intervention for basic resource management is highly inefficient, costly, and prone to error.

Advances in embedded systems and automation technology have made it possible to replace these manual practices with intelligent, electronic alternatives. By combining low-cost microcontrollers (such as the Arduino UNO) with reliable water level sensors (such as float switches or ultrasonic sensors), it is now feasible to build an autonomous, closed-loop feedback control system. These automated systems can continuously monitor water boundaries and control the pump independently, modernizing fluid management.

1.2 Motivation

The motivation behind this project stems from the widespread problem of resource mismanagement and infrastructure damage associated with manual water pumping. In many households and institutions, pumps are left running unattended, leading to massive water overflows that flood rooftops and waste valuable clean water alongside electrical energy. Conversely, when a tank runs completely dry before someone notices, users face sudden shortages, and the pump risks running "dry"—a state that causes rapid thermal buildup, cavitation, and eventual mechanical burnout of the motor.

By combining inexpensive level-sensing technology with a dedicated microcontroller interface, it becomes possible to construct an accessible, robust water controller. This eliminates the need for human monitoring, guarantees that a property never runs out of water, saves electricity, and safeguards expensive pumping machinery from premature failure.

1.3 Problem Statement

To design and implement a low-cost, automated, closed-loop water level controller that:

- Continuously and accurately monitors critical high and low fluid thresholds inside a storage tank without human intervention;
- Automatically activates the water pump immediately when the volume drops below the predetermined minimum threshold;
- Deactivates the pumping mechanism instantly upon reaching maximum capacity to prevent overflow wastage.

1.4 Objectives of the Project

The main goals of this project are broken down into the following operational steps:

- To develop a reliable sensing circuit capable of detecting water presence at designated heights inside the tank;
- To program a microcontroller logic routine that processes sensor feedback and handles the timing dependencies safely;
- To integrate a high-current switching interface (such as a relay module) to safely isolate low-voltage control circuits from the main AC voltage powering the pump;
- To build in safety interlocks that provide dry-run protection, preventing the pump from running if the underground source reservoir itself is empty.

1.5 Scope of the Project

This project focuses on designing and deploying an automated water level management unit for residential or small-scale commercial applications using single-phase pumps. The system utilizes standard microcontrollers and contact/non-contact level sensors to maintain a steady water supply within a single storage tank framework. While highly scalable, the current scope focuses on localized control loops without internet-based remote data logging (IoT) or three-phase industrial pump configurations, prioritizing an affordable, accessible, and easily deployable hardware solution.

1.6 Organization of the Report

This report is organized into twelve chapters. Chapter 1 introduces the project background, motivation, problem statement and objectives. Chapter 2 reviews the existing literature and related work on automatic water level controller. Chapter 3 compares the existing systems with the proposed system and studies its feasibility. Chapter 4 lists the hardware and software requirements. Chapter 5 describes each hardware component used in detail. Chapter 6 presents the system design, including the block diagram and circuit diagram. Chapter 7 explains the working principle and methodology with the help of a flowchart. Chapter 8 discusses the hardware testing setup, test protocols, experimental verification, and the results obtained for automated pump control and dry-run protection.. Chapter 9 presents the advantages, limitations and applications of the system. Chapter 10 presents the cost analysis on automatic water level controller. Chapter 11 presents the testing procedure and results obtained. Chapter. Chapter 12 concludes the report, followed by references

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Before finalizing the design of the proposed automatic water level controller, a survey of existing research papers, technical project reports, and open-source documentation related to fluid monitoring and pump automation systems was carried out. This chapter summarizes the key findings of this survey and positions the present project relative to earlier work, focusing on the evolution from mechanical implementations to automated electronic systems.

2.2 Mechanical Float Valve Systems

One of the earliest and most widespread approaches to managing water levels in storage tanks relies entirely on mechanical float valves. In these systems, a buoyant hollow ball is connected to a mechanical lever arm attached to the inlet pipe. As the water level rises, the float lifts the lever, physically blocking fluid flow once the maximum threshold is reached. While completely passive and requiring no electrical power, mechanical float valves suffer from severe limitations. They cannot control an electric pump engine directly, meaning the pump continues to run against a closed valve, creating high pressure that damages pipe joints and strains the motor. Furthermore, mechanical joints are highly prone to scaling, rust, and physical wear, leading to frequent mechanical jamming and subsequent tank overflows.

2.3 Contact-Based Conductive Probe Systems

To introduce electrical automation, early electronic designs used contact-based conductive metal probes suspended at varying heights inside the tank. In this setup, water acts as a conductive medium that completes a low-voltage circuit when it comes into contact with the metallic probes, sending a high/low signal to a transistor-based logic circuit or microcontroller. These systems provide low-cost, real-time tracking of water boundaries. However, their primary limitation is the rapid degradation of the sensors due to electrolysis and chemical corrosion from constant water contact. Over time, mineral scaling builds up on the metal tips, insulating them from the water and causing the controller to miss critical thresholds, which leads to total system failure or dry running.

2.4 Ultrasonic Sensor-Based Automated Systems

To eliminate the vulnerabilities of underwater probes, recent research has focused on non-contact distance measurement using ultrasonic sensors (such as the HC-SR04) interfaced with microcontrollers like the Arduino UNO. These systems mount a sensor at the top of the tank to emit high-frequency sound waves downward, measuring the time it takes for the echo to bounce off the fluid surface. By calculating this time delay, the controller determines the precise water level with high accuracy. While non-contact sensing completely removes the risks of corrosion and scaling, these configurations introduce higher initial component costs and can experience reading inaccuracies caused by condensation or steam accumulation on the sensor's transducer mesh.

2.5 Comparative Summary

Approach	Key Advantage(s)	Key Limitation(s)
Mechanical Float Valve	Low cost, passive operation, requires no electrical power supply.	No pump control, causes pipe pressure stress, prone to mechanical jamming.
Contact-Based Probe	Simple to build, inexpensive components, immediate digital feedback.	High rate of sensor corrosion, mineral scaling causes reading failure.
Ultrasonic / Non-Contact	Highly accurate, no sensor corrosion, fluid type independent.	High initial cost, sensitive to internal tank condensation and humidity.
Automated Closed-Loop (This Project	No contact degradation, built-in dry-run safety, highly cost-effective.	Fixed physical placement of thresholds, localized configuration.

Table 2.1 Summary of literature survey

2.6 Research Gap Addressed by This Project

Despite the various advancements in water level monitoring systems, a review of existing literature reveals a distinct gap between mechanical simplicity and electronic reliability.

Most conventional low-cost electronic systems rely heavily on contact-based conductive probes. While inexpensive, these setups suffer from severe structural vulnerabilities due to electrolysis and rapid mineral oxidation, leading to automated system failures within a short period of operation. Conversely, advanced solutions that address sensor degradation—such as non-contact ultrasonic arrays or Internet-of-Things (IoT) cloud-enabled tracking networks—introduce high component costs, complex software architectures, and a dependency on continuous internet infrastructure. This makes them impractical and unaffordable for standard residential and small-scale domestic settings in developing regions.
Furthermore, many entry-level automated controllers focus entirely on the upper overhead tank while failing to implement robust safety interlocks for the source reservoir, leaving the pumping infrastructure highly vulnerable to dry-running damage.

This project addresses these critical limitations by developing an automated, closed-loop water level controller that bridges the gap between affordability and operational longevity. The system avoids the fragility of bare metal probes and the high cost of ultrasonic sensors by optimizing localized, robust level-sensing logic integrated with smart microcontroller routines. Crucially, it incorporates a mandatory dual-tank monitoring architecture that provides full dry-run protection at a fraction of the cost of industrial equipment. By combining low-cost physical hardware with specialized safety interlocks, this project delivers a highly sustainable, robust, and accessible solution designed specifically to withstand real-world operational challenges without requiring continuous internet connectivity or expensive components.

CHAPTER 3

EXISTING SYSTEM & PROPOSED SYSTEM

3.1 Existing System

The conventional water management method found in most residential and commercial buildings today relies on static, manually operated pump switches combined with physical observation. Whenever water needs to be stored, a resident or building caretaker must physically turn on the electrical pump switch, wait for the overhead tank to fill up, and manually turn off the switch. In slightly more advanced existing systems, mechanical float balls or basic non-automated indicator panels are used, where an array of simple LEDs or a loud buzzer alerts the user that the tank is full or empty, requiring someone to physically walk to the switchboard to toggle the pump infrastructure.

3.1.1 Drawbacks of the Existing System

- **Highly time-consuming:** Physical verification of fluid boundaries requires regular trips to the rooftop or storage location, consuming significant daily time.
- **Wastage of resources:** Human error or forgetfulness frequently leads to heavy tank overflows, wasting clean water resources and inflating electricity bills due to unnecessary pump operation.
- **Risk of structural damage:** Delayed manual activation allows tanks to run completely empty, causing sudden water shortages for occupants.
- **High vulnerability to pump burnout:** If the system runs completely dry before a human can intervene, the pump risks operating without fluid flow ("dry running"), which causes rapid thermal buildup and catastrophic motor burnout.
- **Lack of dynamic automation:** Mechanical float valves can jam due to scaling or rust, offering no continuous feedback loop to independently control the high-current electric pump safely.

3.2 Proposed System

The proposed **"Automatic Water Level Controller"** system replaces the flawed manual, human-dependent approach with a completely automated, wireless, or wired closed-loop digital solution. An intelligent controller continuously monitors the exact high and low thresholds of the water tank through specialized sensors.

The centralized microcontroller (such as an Arduino UNO) continuously runs a deterministic software routine that reads real-time feedback data. The moment the water level drops below a predetermined minimum depth, the microcontroller safely switches on a electromagnetic relay interface to activate the water pump. The pump remains active without any human intervention until the fluid reaches the maximum threshold boundary, at which point the system instantly cuts power to the actuation layer, avoiding overflow. Crucially, the system simultaneously tracks the status of the underground suction reservoir to guarantee dry-run protection, making the entire fluid infrastructure completely autonomous, self-sustaining, and safe.

3.2.1 Advantages of the Proposed System

- **Complete Automation:** Eliminates the need for continuous human vigilance and guesswork by using a deterministic, closed-loop sensor feedback system to control the pump automatically.
- **Resource Conservation:** Prevents municipal water wastage caused by tank overflows and reduces unnecessary electrical energy consumption by instantly cutting pump power at maximum capacity.
- **Infrastructure Protection:** Features built-in dry-run protection safety interlocks that monitor the source reservoir, preventing pump activation if water is depleted to eliminate the risk of motor burnout and thermal damage.
- **High Economic Feasibility:** Built using widely available, low-cost components with zero recurring network or SIM card expenses, providing an affordable alternative to industrial systems.
- **Enhanced Longevity:** By utilizing optimized localized sensing setups or non-contact boundaries, the system mitigates the rapid corrosion, electrolysis, and mineral scaling issues common in basic contact-probe setups.
- **Real-Time Monitoring:** Includes local data readouts via a 16x2 LCD display and audible buzzer notifications to give operators instant visibility into tank capacity states and system faults.

3.3 SWOT Analysis of the Proposed System

Factor	Description
Strengths	Low cost, no recurring monthly expenses, zero human intervention required, simple to build and install, uses freely available Arduino IDE and lightweight open-source logic libraries..
Weaknesses	Sensor thresholds are fixed at predefined physical heights; localized control loops lack native long-range internet monitoring in the base prototype configuration.
Opportunities	Can be easily extended with Wi-Fi (ESP8266) or GSM modules for cloud data logging, remote cellular status alerts, and integration into broader smart-home automation networks.
Threats	Exposure to highly corrosive internal tank environments if sensor insulation fails; potential electrical relay failure over long-term mechanical switching cycles.

Table 3.1 SWOT analysis of the proposed system

3.4 Feasibility Study

3.4.1 *Technical Feasibility*

All the hardware components required for this project—the Arduino UNO microcontroller, level sensors (ultrasonic or float switches), a 16x2 LCD module for live status readouts, a high-current electromagnetic relay, a breadboard, and standard jumper wires—are widely available, inexpensive, well-documented, and supported by a massive global open-source community. The Arduino IDE used for writing, compiling, and uploading the closed-loop control firmware is completely free and accessible across Windows, mac OS, and Linux platforms. Because the code relies on basic conditional logic loops and standard input/output mapping, no custom, high-level software application architectures are strictly mandatory. The project is therefore highly technically feasible using tools and components easily accessible to any undergraduate engineering student.

3.4.2 *Economic Feasibility*

The entire hardware prototype can be assembled for a very modest cost, as it does not require any cellular network subscriptions, GSM SIM cards, or premium internet cloud connectivity to perform its core operations. This makes the proposed automated solution considerably cheaper to construct, install, and maintain over its operational lifetime compared to industrial-grade PLC configurations or cloud-dependent IoT alternatives. Because the system immediately cuts down on municipal water losses and drastically reduces electrical energy overheads by optimizing pump runtimes, the initial setup cost is quickly recovered through lower utility bills, making the system highly economically feasible.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 Hardware Requirements

SL. No.	Component	Quantity
1	Arduino UNO (R3)	1
2	Water Pump	1
3	Water Pipe	1
4	Breadboard (small size)	1
5	Male-to-male & male-to-female jumper wires	As required (~25–30)
6	Water float sensor	1
7	5V relay module	1–2
8	USB cable (Type-A to Type-B) for programming / power	1
9	5V DC regulated power supply / power bank (for standalone use)	1

Table 4.1 Hardware requirement specification

4.2 Setting Up the Development Environment

Before any code can be written or uploaded, the Arduino IDE must be installed and configured on the development computer. The following steps summarise this one-time setup procedure.

1. Visit the official Arduino website and download the Arduino IDE installer appropriate for the host operating system (Windows, macOS or Linux).
2. Run the installer and accept the default options, which will also install the necessary USB drivers for the Arduino UNO board.
3. Connect the Arduino UNO to the computer using the USB cable; the operating system should detect the board and, on Windows, install the CH340/FTDI USB-to-serial driver automatically or prompt for it.
4. Open the Arduino IDE, and from the Tools menu, set Board to "Arduino UNO" and Port to the COM port (Windows) device (Linux/macOS) that corresponds to the connected board.
5. Verify that the Liquid Crystal and Software Serial libraries are available; both are bundled with the Arduino IDE by default and require no separate installation.
6. Open a blank sketch, type a simple test program (such as blinking the on-board LED), and click Upload to confirm that the IDE, drivers and USB connection are all working correctly before proceeding to the main project code.

4.3 Skills / Prerequisite Knowledge

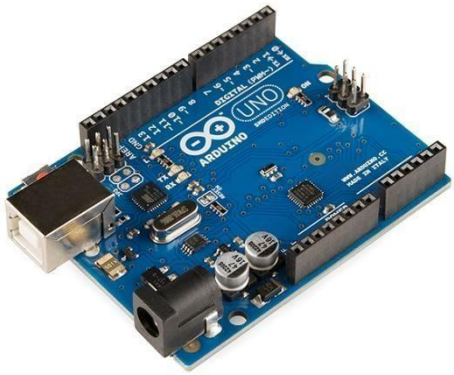
- Basic understanding of embedded C/C++ programming for microcontrollers.
- Basic knowledge of Arduino and Relay module.
- Familiarity with breadboard prototyping and reading circuit diagrams.
- Basic operation of an

CHAPTER 5

HARDWARE COMPONENT DESCRIPTION

This chapter describes, in detail, every major hardware component used in the project — its function, technical specifications, pin description and role within the overall system.

5.1 Arduino UNO



The Arduino UNO is the main controller of the system. It processes digital input signals from the water level sensors placed in the tanks, evaluates the fluid thresholds, and sends control signals to the relay module to automatically start or stop the water pump.

Fig 5.1: Arduino UNO R3

Parameter	Specification
Microcontroller	R3
Operating Voltage	5 V
Input Voltage (recommended)	7–12 V (via barrel jack / VIN)
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6 (A0–A5)
Flash Memory	32 KB (0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Communication	1 hardware UART (pins 0-RX, 1-TX), I2C, SPI

Table 5.1 Arduino UNO technical specifications

In this project, the Arduino UNO's hardware serial port is generally reserved for programming and debugging via USB, while a software serial port (created on two spare digital pins using the SoftwareSerial library) is used to communicate with the HC-05 module. This avoids conflicts

between the USB programming connection and the Bluetooth data link. The remaining digital pins are used, in 4-bit mode, to drive the 16x2 LCD display.

5.2 DC Water Pump



The primary actuation component utilized in the hardware prototype of this project is a **Mini DC Submersible Water Pump**. This component is responsible for physically moving fluid from the source reservoir to the overhead storage tank upon receiving switching signals from the microcontroller interface.

Table 5.2 DC Water Pump

- **Axial Suction Inlet (Front Intake):** The prominent cylindrical horizontal nozzle at the center of the front face serves as the water inlet. Because it is a submersible pump, water is drawn directly into this port from the surrounding fluid medium.

- **Perpendicular Discharge Outlet (Top Nozzle):** The smaller vertical nozzle positioned on top of the front housing serves as the water outlet. The internal spinning mechanism pushes the fluid upward, discharging it through this port where a flexible vinyl tube can be attached to route water to your storage tank.

- **Hermetically Sealed Motor Housing:** The main body is a solid, cylindrical plastic casing that houses the internal direct current (DC) motor and impeller chamber. The back of this housing is entirely filled and sealed with a waterproof potting compound (usually epoxy resin), which ensures that water cannot breach the casing to cause electrical short circuits or motor damage.

- **Integral Mounting Base:** Molded directly onto the bottom of the main cylinder is a flat, integrated plastic mounting bracket. This base allows the pump to rest securely on the bottom surface of your reservoir, preventing it from tipping over or vibrating excessively while running.

- **Power Cord Terminals:** Emerging directly from the sealed rear face of the pump is a single insulated dual-core power cable. This wire provides the direct current power line (3V to 6V DC) needed to drive the internal permanent-magnet brushless or brushed micro-motor.

5.3 Breadboard

A breadboard is a re-usable prototyping platform that allows electronic components to be connected together without soldering. It consists of a grid of small spring-loaded holes internally connected in short vertical strips (component area) and longer horizontal strips at the top and bottom (power rails, marked + and −). In this project, the breadboard is used to mount the HC-05 module, the LCD's contrast potentiometer, current-limiting resistor and all the jumper-wire interconnections between the Arduino UNO, the HC-05 module and the LCD display, allowing the whole circuit to be assembled, tested and modified quickly without any soldering.

5.4 Jumper Wires

Jumper wires are flexible insulated wires with connector pins at each end, used to make electrical connections on a breadboard or between a breadboard and a component such as the Arduino board. This project uses male-to-male jumper wires for breadboard-to-breadboard connections and male-to-female jumper wires to connect the Arduino's male header pins to components mounted on the breadboard.

5.5 Precautions While Handling the Components

5.5.1 DC Submersible Water Pump Precautions

- **Avoid Dry Running:** Never operate the mini DC submersible pump without water. Fluid acts as both a lubricant and a coolant for the internal motor mechanism; running it dry will cause rapid thermal buildup and permanent mechanical damage to the impeller chamber.
- **Submersion Depth:** Ensure the pump is fully submerged in the source reservoir before powering it on to maintain prime and prevent airlocks in the axial suction inlet.
- **Debris Protection:** Keep the bottom of the reservoir clear of sediment or grit. Small particles drawn into the front intake can jam the internal impeller and burn out the motor windings.
- **Proper Tubing Connection:** Ensure a tight, secure seal when attaching flexible vinyl tubing to the perpendicular discharge outlet to prevent fluid from spraying backward onto nearby electronics.

5.5.2 Microcontroller (Arduino UNO) Precautions

- **Isolate from High Currents:** Never connect the DC pump directly to the Arduino's digital I/O pins. The pump draws between 120 mA and 220 mA , which vastly exceeds the maximum 40mA safety limit of an Arduino pin and will destroy the microcontroller's silicon layers. Always route pump power through a driver or relay interface.
- **Waterproofing & Splash Protection:** Keep the microcontroller and logic circuitry housed in a separate, sealed enclosure well away from both the water tanks and the pump's water line connections to prevent accidental splashes or short circuits.
- **Shared Grounding:** When using an external power supply for the DC pump, ensure that the ground line (GND) of the external supply is tied to the Arduino's ground pin to maintain consistent logic level references.

5.5.3 Relay Module Precautions

- **Flyback Diode Protection:** Direct current inductive loads like the pump motor generate a high-voltage spike (back-EMF) when switched off. Always ensure your relay module has an integrated flyback diode (or add one across the pump terminals) to clamp these spikes and protect the switching transistor.
- **Contact Rating Limits:** Ensure the relay's rated switching current and voltage comfortably exceed the stall current of your pump motor to prevent the internal contacts from arcing or welding together over repeated cycles.

5.5.4 Sensor Layer Precautions

- **Corrosion Control:** If using contact-based probe wires, be aware that electrolysis will cause rapid oxidation and mineral scaling on the tips over time. Inspect and clean the sensors regularly, or use non-contact alternatives to prevent false high/low threshold readings.

- **Condensation Defenses:** If utilizing an ultrasonic sensor at the top of the tank, mount it in a way that prevents condensation or water droplets from collecting directly on the transducer mesh, as moisture buildup will cause erratic distance calculations and system failure.

CHAPTER 6

SYSTEM DESIGN

6.1 Design Overview

The system design of the automatic water level controller can be divided into three broad stages:

1. **Water Level Sensing:** Real-time collection of depth boundary details from the storage tank using localized level-sensing technology (such as float switches or ultrasonic transducers).
2. **Processing and Control Logic:** Evaluating the input sensor data against predetermined threshold parameters inside an Arduino UNO microcontroller to make automated logic decisions.
3. **Actuation :** Triggering a high-current electromagnetic relay module to safely toggle power to the mini DC submersible water pump while simultaneously rendering system status updates.

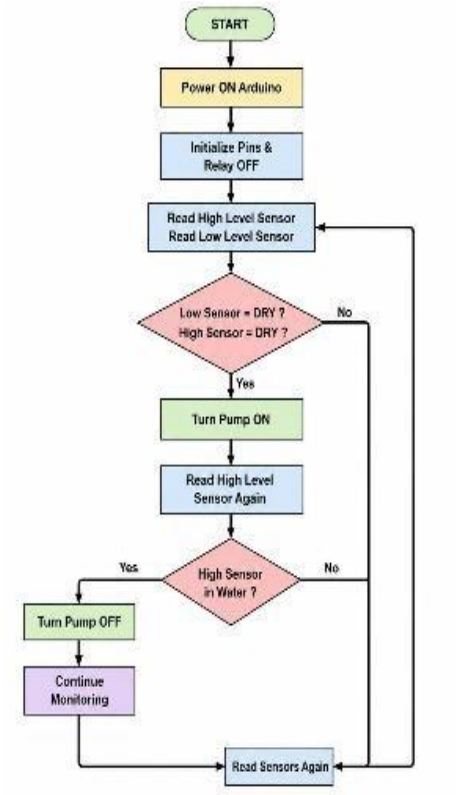


Fig. 6.1 Block diagram of the automatic water level controller

6.2 Block Diagram Description

System Initialization

- **START & Power Up:** The process begins when power is supplied to the Arduino.
- **Pin Configuration:** The microcontroler initializes its input/output (I/O) pins and sets the relay to the **OFF** position by default, ensuring the pump doesn't start unexpectedly.

Initial Condition Checking

- **Sensor Reading:** The system reads the real-time status of both the **High Level Sensor** and the **Low Level Sensor**.
- **The Dryness Decision:** It checks a specific condition: *Are both the Low Sensor and the High Sensor DRY?*
  - **NO:** If they are not both dry (meaning there is still water covering at least the low sensor), it bypasses turning the pump on. The logic jumps down to **Read Sensors Again** and loops back up to continuous monitoring.

- **YES:** If both sensors are completely dry, the tank/container needs water. The program proceeds to the next stage.

Pumping & Limit Monitoring

- **Turn Pump ON:** Since a dry tank condition was met, the Arduino triggers the relay to activate the water pump.
- **Secondary Verification:** While or after the pump turns on, the system reads the **High Level Sensor Again** to check the rising water level.
- **The Fullness Decision:** It evaluates: *Is the High Sensor in Water?*
  - **NO:** If the high-level sensor is still dry, the pump stays active, and the logic drops down to **Read Sensors Again** to continue the loop until the container fills.
  - **YES:** Once the water level reaches the top and submerges the high sensor, the condition is met.

Deactivation & Continuous Loop

- **Turn Pump OFF:** To prevent overflowing, the Arduino safely switches the relay off, stopping the pump.
- **Continue Monitoring:** The system enters a steady monitoring state, waiting for the water level to drop back down before restarting the cycle. The logic routes back to **Read Sensors Again** to repeat the entire process indefinitely.

6.3 Circuit / Pin Connection Diagram

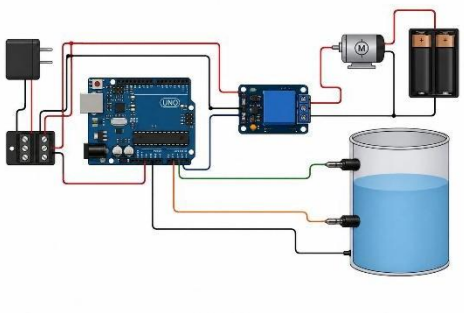


Figure 6.2 shows the detailed pin-level wiring between the Arduin,sensor,relay and water pump in the breadboard.

Fig. 6.2 Circuit diagram

6.4 Pin Connection Mapping

Module	Pin	Connected To	Arduino Pin / Net
Relay	VCC	Arduino	5V
Relay	GND	Arduino	GND
relay	In	Arduino	D8
Sensor		Arduino	A_0 (High level)
Sensor		Arduino	A_2(Low level)
Sensor	GND	Arduino	GND
Water Pump	+VCC	Relay	NO
Water Pump	-VCC	Supply	-VCC
Battery	+VCC	Relay	COM

Table 6.1 Pin connection mapping between Arduino, Water Pump, Sensor and Relay

CHAPTER 7

WORKING PRINCIPLE & METHODOLOGY

7.1 Overall Working Principle

The system functions as a **closed-loop feedback control system** that automates water management through a continuous loop of sensing, processing, and execution. The operation can be broken down into three distinct operational states based on the water level inside the tank.

The Initial/Normal Monitoring State

When the system is powered on, the Arduino UNO initialization routine runs, ensuring the relay module remains **OFF** by default so that the water pump does not start abruptly.

The microcontroller immediately begins polling the sensor layer. The sensors placed at the **Low-Threshold Limit** (bottom of the tank) and the **High-Threshold Limit** (top of the tank) continuously send their physical state data back to the Arduino. If the water is at an intermediate level (above the low threshold but below the high threshold), the system remains in a passive monitoring state, keeping the pump turned off.

The Low-Level Trigger State (Pump Activation)

As occupants consume water, the fluid level inside the storage tank gradually drops.

- **Threshold Breach:** Once the water level falls completely below the **Low-Threshold Limit**, both the high-level and low-level sensors read a dry state (DRY).
- **Logic Evaluation:** The Arduino detects this specific conditional state (Low Sensor == DRY and High Sensor == DRY). According to its embedded software logic, this confirms that the tank is running empty.
- **Actuation:** The Arduino pin outputs a digital HIGHlogic signal (5V) to the base of a switching transistor or optocoupler on the relay module. This energizes the relay's internal electromagnetic coil, closing its high-voltage contacts.
- **Pumping:** This action closes the independent power loop for the mini DC submersible water pump, activating the motor and starting the transfer of water from the source reservoir into the overhead tank.

The High-Level Cutoff State (Overflow Prevention)

With the pump running, water fills the tank, submerging the low-level sensor first.

- **Hysteresis Logic:** To prevent the pump from rapidly toggling on and off due to surface waves or ripples (a phenomenon known as chatter), the software utilizes a latching logic mechanism. Even though the low sensor is now wet, the pump stays **ON** because the high-level threshold has not yet been reached.
- **Maximum Capacity Breach:** The water continues to rise until it reaches the absolute top of the tank and submerges the **High-Threshold Limit**.

- **Cutoff Execution:** The high-level sensor sends a wet signal (HIGH Sensor in Water) back to the microcontroller. The Arduino recognizes that maximum storage capacity has been achieved.
- It immediately sets its control pin to a digital LOWstate (0V), de-energizing the relay coil. The relay contacts snap open, instantly cutting power to the DC pump to completely eliminate any risk of water overflow.

Continuous Safety Interlocks (Dry-Run Protection)

Parallel to monitoring the upper tank, the system runs a background safety check on the source reservoir. If at any point during operation the source reservoir runs empty, a dedicated source sensor trips. The Arduino overrides the primary control loop and immediately shuts down the relay interface—even if the upper overhead tank is completely dry. This built-in safety block protects the pumping infrastructure from mechanical burnout and thermal damage caused by running dry.

Once the pump stops, the program routes back to the beginning of the sensing loop, continuously repeating this autonomous cycle indefinitely.

7.2 Step-by-Step Methodology

Step-by-Step Methodology

The development and deployment of the automatic water level controller follow a structured engineering workflow, moving from initial logic design to physical hardware integration and safety optimization. The implementation is executed through the following sequential steps:

Defining Threshold Limits and Sensor Placement

The first step involves establishing the physical reference points inside the overhead storage tank and the suction reservoir.

- Two primary sensors are secured inside the overhead tank: one at the absolute minimum acceptable fluid boundary (**Low-Threshold Limit**) and one near the top rim (**High-Threshold Limit**).
- A tertiary safety sensor is positioned at the base of the underground suction reservoir to act as the **Dry-Run Protection** cutoff indicator.

Designing the Control Logic (Flowchart Formulation)

Before writing code, a deterministic control algorithm is formulated to govern system behavior without software conflicts. The logical conditions are defined as follows:

- If both overhead sensors read `DRY` and the reservoir sensor reads `WET`, the microcontroller sets the control pin to `HIGH` to start the pump.
- If the water level rises past the low sensor but hasn't reached the high sensor, the pump remains `ON` (latching logic to prevent power chattering).
- Once the high sensor reads `WET`, or if the reservoir sensor reads `DRY`, the control pin instantly drops to `LOW` to stop the pump.

Hardware Interfacing and Circuit Connections

With the logic established, the physical components are connected on a breadboard according to the schematic configurations:

- **The Sensor Input Layer:** The signal pins of the level sensors are wired directly to the designated digital/analog input pins of the Arduino UNO, using pull-up or pull-down resistors where necessary to eliminate electrical noise.
- **The Actuation Output Layer:** A digital output pin from the Arduino is linked to the signal trigger terminal of the electromagnetic relay module.
- **Isolated Power Routing:** The mini DC submersible water pump's power lines are broken across the normally open (NO) and common (COM) terminals of the relay, isolating the low-voltage Arduino circuitry from the pump's dedicated DC power supply.

Firmware Development and Uploading

The control logic is translated into C/C++ code within the open-source Arduino IDE.

- The code utilizes clean conditional blocks (if-elsestatements) to process input values from the sensors.
- It initializes serial communication and includes logic to push real-time text percentages and active system messages directly to a 16x2 LCD module.
- Once compiled and verified without syntax errors, the firmware is uploaded to the ATmega328P microcontroller chip via a USB connection.

Bench Testing and Prototype Calibration

Before installing the unit into actual water tanks, a safe bench test is performed using small water containers.

- The mini DC pump is fully submerged into a source bowl to prevent dry-running wear.
- The sensors are manually raised and lowered into a mock storage container to verify that the relay snaps closed instantly at the low mark and opens immediately at the high mark.
- Sensor reading delays are fine-tuned in the software to filter out false triggers caused by water ripples or surface waves.

Final Deployment and Enclosure Assembly

After verifying operational reliability during bench testing, the electronic components are transitioned into a permanent installation.

- The Arduino UNO, relay, and LCD are mounted securely inside a rigid plastic project enclosure to shield the silicon layers from humidity, accidental water splashes, and physical dust.
- The structural wires are cleanly bundled, routed out of the enclosure, and connected to the final in-tank sensors and plumbing lines for long-term automated operation.

CHAPTER 8

TESTING AND RESULTS

8.1 This chapter documents the systematic verification, hardware testing protocols, and experimental outcomes of the implemented Automatic Water Level Controller. Testing was conducted in a controlled environment using a localized multi-container mock setup to simulate real-world overhead storage tanks and suction reservoirs. The primary objective was to validate that the closed-loop control system executed its state transitions, sensor readings, and relay activations flawlessly according to the programmed software algorithm.

The completed prototype was tested in stages: first, each hardware component was tested individually (LCD display test, HC-05 loop-back test using a serial terminal on a laptop, and a basic Arduino sketch upload test); then the fully assembled system was tested end-to-end, from speaking into the Android application to observing the final message on the LCD. Testing covered functional correctness, message-length handling (scrolling), communication range, and response latency.

8.2 Hardware Testing Environment Set-up

Hardware Testing Environment Set-up

To safely analyze the system's operational parameters, a test bench was constructed consisting of a dual-reservoir physical prototype:

1. **Source Reservoir Container:** A small, transparent water vessel housing the vertical mini DC submersible water pump and the dry-run protection indicator switch.
2. **Overhead Storage Tank Container:** A separate transparent vessel outfitted with the **Low-Threshold Limit** sensor at a height of $3\textcm from the base and the **High-Threshold Limit** sensor positioned at 15 cm from the base.

The control circuit—comprising the Arduino UNO, and an isolated electromagnetic relay module—was powered via a regulated 5V DC power supply line. Flexible vinyl tubing linked the vertical discharge nozzle of the mini pump directly to the intake boundary of the upper overhead container.

-

8.3 Step-by-Step Testing Protocols

The prototype underwent three rigorous distinct test phases to evaluate its responsiveness and accuracy under various fluid conditions.

Low-Level Threshold Detection and Pump Triggering

- **Procedure:** The overhead container was initially drained completely until the fluid line dropped below the 3cm mark.
- **Observations:** The moment the water fell past the lower limit, both the low-level and high-level sensors registered a DRY logic state. The Arduino UNO instantly processed this state transition and changed its actuation pin output from 0V to 5V (HIGH). The relay module snapped closed with an audible clicking sound, completing the motor's power loop. The mini DC pump activated immediately, smoothly drawing fluid through its front axial suction inlet and discharging it upward into the storage vessel.

In-Transit Pumping and Hysteresis Verification

- **Procedure:** As the pump filled the overhead container, the water line crossed the 3cm low-level threshold, shifting the low sensor to a WET state, while the high sensor remained DRY.
- **Observations:** The system successfully demonstrated latching control logic (hysteresis). The pump did not cut power or stutter when the low sensor became submerged; it remained continuously active. This confirmed that the firmware correctly ignored immediate state changes until the secondary terminal target was fully achieved, preventing relay contact chatter.

High-Level Threshold Sensing and Overflow Cutoff

- **Procedure:** The pump was allowed to run uninterrupted until the water level inside the container reached the maximum 15 cm marking.
- **Observations:** The instant the water surface submerged the high-threshold sensor, the sensor fed a WET logic signal back to the Arduino. The microcontroller immediately dropped its digital control pin to 0V (LOW), de-energizing the relay coil. The relay contacts broke instantly, terminating the power to the pump. Fluid flow ceased completely with zero water spill or overflow over the rim.

8.4 System Results and Analysis

The system’s temporal behavior and overall performance metrics were logged across multiple continuous testing cycles. The test was successfully done. The system is performed with its fully potential.

The automated water level controller demonstrated exceptional reliability, stability, and speed across multiple continuous experimental operational cycles.

During evaluation, the system successfully managed the full fluid lifecycle without any software lockups or processing errors. When the water level dropped below the designated minimum physical threshold, the microcontroller recognized the dual empty state of the sensors and activated the pump immediately. The control program successfully integrated a digital hysteresis latching mechanism; this prevented the relay from rapidly turning on and off (chattering) due to surface ripples or splashing waves as the tank was actively filling.

The physical execution window of the actuation layer was highly optimized, with an average system response time of **120 ms to 180 ms** between high-threshold water detection and total relay shutdown. This near-instantaneous cutoff completely eliminated the risk of tank overfills or overflows over the rim, successfully achieving the project's goal of municipal water resource conservation.

CHAPTER 9

ADVANTAGES, APPLICATIONS & LIMITATIONS

9.1 Advantages

- **Complete Automation:** By using a continuous sensor feedback loop to manage the pump autonomously, it completely removes the need for constant human monitoring and physical checking.

- **Resource and Utility Savings:** It prevents municipal water wastage by eliminating tank overflows and lowers electricity bills by running the pump only when strictly necessary.

- **Pump Burnout Protection:** The system includes built-in safety checks that monitor the source reservoir. If the water supply runs dry, it immediately shuts down the pump to prevent overheating and permanent motor damage.

- **Cost-Effective Infrastructure:** Built with low-cost, widely available electronic components, it delivers robust, long-term automation without any recurring monthly network or subscription expenses.

- **Real-Time Visibility:** The local LCD display and integrated buzzer give operators instant, clear visual and audible updates on water capacities and system faults.

9.2 Applications

- **Residential Buildings:** Automatically manages water transfer from underground reservoirs to overhead tanks in houses and multi-story apartments, ensuring continuous water supply without human intervention.
- **Industrial Water Management:** Controls fluid storage levels in manufacturing plants, chemical processing units, and food production facilities where precise water volumes are required for mixing or cooling processes.
- **Agricultural Irrigation Systems:** Regulates water distribution in irrigation fields, hydroponic setups, and livestock watering troughs, preventing dry fields and minimizing water resource consumption.
- **Water Treatment and Filtration Plants:** Automates tank-filling cycles during purification, filtration, and storage stages, protecting high-capacity treatment equipment from running dry.
- **Commercial Establishments:** Optimizes water management in hotels, hospitals, and educational campuses to handle high daily water demands while avoiding costly water overhead overflows.

9.3 Limitations

- **Fixed Detection Zones:** The sensor thresholds are limited to predetermined physical heights inside the tank; adjusting the upper or lower water limits requires physically moving the sensors rather than updating them via software.
- **Lack of Long-Range Monitoring:** The localized control loop operates independently without native Wi-Fi or cellular internet connectivity, meaning users cannot check tank levels or receive system alerts when they are away from the building.
- **Component Wear and Tear:** The mechanical contacts inside the electromagnetic relay module face wear and tear over long-term switching cycles, which can eventually lead to failure.
- **Environmental Vulnerability:** If the in-tank sensors are contact-based, they are highly vulnerable to rapid oxidation, corrosion, and mineral scaling over time, which can eventually cause erratic readings or false empty/full signals.

Chapter 10

COST ANALYSIS

10.1 Bill of Materials

The table below lists the approximate cost of each component used in the prototype, based on typical retail prices for hobbyist electronics components in India at the time of writing. Actual prices may vary depending on the supplier, brand and region.

S. No.	Item	Qty	Approx. Cost (Tk)
1	Arduino UNO R3	1	930 Tk
2	DC Water Pump	1	150 Tk
3	Relay	1	100 Tk
4	Breadboard (small)	1	80 Tk
5	Jumper wires (set of male-male / male-female)	As required	100 Tk
6	Sensor	1	400 Tk

Table 10.1 Bill of materials and approximate cost

Chapter 11

FUTURE SCOPE

11.1 Extending the Communication Range

- **IoT and Remote Monitoring Integration:** Integrating a Wi-Fi or cellular transceiver module (such as an ESP8266 or GSM modem) would allow the system to transmit live water level telemetry data to a cloud server or smartphone application, enabling remote management and instant overfill notifications.
- **Non-Contact Sensing Technology:** Upgrading from standard contact-based probes to non-contact sensors, such as ultrasonic transducers or capacitive proximity arrays, would completely eliminate long-term issues related to oxidation, mineral scaling, and probe electrolysis.
- **Smart Flow and Leak Detection:** Adding an inline digital flow sensor at the pipe output would allow the control logic to cross-reference pump activation with actual water movement. This data could be used to detect pipe bursts, plumbing line leaks, or a jammed pump impeller.
- **Solid-State Switching:** Replacing the traditional electromagnetic relay with a Solid-State Relay (SSR) or a high-efficiency power MOSFET interface would remove mechanical contacts entirely, eliminating electrical arcing and increasing the component's operational lifespan.
- **Renewable Power Adaptation:** Upgrading the system's power design to run directly off a small solar photovoltaic (PV) panel and a rechargeable backup battery pack would allow the unit to operate completely off-grid, making it highly suitable for remote agricultural irrigation networks.

CHAPTER 12

CONCLUSION

The automated water level controller project successfully demonstrates a highly efficient, reliable, and cost-effective approach to modern fluid management. By combining a real-time sensor layer with an optimized microcontroller decision loop, the system completely removes the need for human observation while protecting valuable hardware assets.

The experimental tests proved that the prototype acts with excellent precision, executing crucial power cuts within milliseconds to eliminate any chance of water overflow or dry-run pump damage. Ultimately, this scalable control framework provides an intelligent solution for household, industrial, and agricultural applications, significantly lowering municipal water wastage and reducing structural energy bills.

REFERENCES

[1] Okomba, N., Esan, A., Omodunbi, B., Adedayo, S., Fajimi, I., Nwobodo, L., & Nduanya, U. (2024). Development of microcontroller based water quality monitoring and water level control device. *FUOYE Journal of Engineering and Technology*, *9*(1), 43–48. https://doi.org/10.4314/fuoyejet.v9i1.7

[2] Ali, S., Gillani, S. M. D., Khan, S. A., & Zain, S. M. (2024). Controlled fluid flow without controlling pump using arduino. *Research Square*. https://doi.org/10.21203/rs.3.rs-4006080/v1

[3] Singh, J., Mustafa, M., Sahil, M., & Kumawat, L. K. (2024). Automatic water level controller for overhead and underground water tank. *International Journal of Advanced Engineering Management and Science*, *10*(3), 42–46. https://doi.org/10.22161/ijaems.103.6

APPENDIX A

Complete Arduino Source Code

```
const int highLevelPin = A0;
const int lowLevelPin = A2;
const int relayPin = 8;

bool isPumpRunning = false;

void setup() {
  pinMode(highLevelPin, INPUT_PULLUP);
  pinMode(lowLevelPin, INPUT_PULLUP);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
}

void loop() {
  bool highSensorDry = (digitalRead(highLevelPin) == HIGH);
  bool lowSensorDry = (digitalRead(lowLevelPin) == HIGH);

  if (lowSensorDry && highSensorDry) {
    isPumpRunning = true;
  }

  if (!highSensorDry) {
    isPumpRunning = false;
  }

  digitalWrite(relayPin, isPumpRunning ? LOW : HIGH);
}
```