International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Automatic Hot Water Generation and Level Control System Using Microcontroller

Udaysinh Jadhav[1], Dr. A. P. Pandhare[2]

[1]Student, Department of Mechanical Engineering, Sinhgad College of Engineering, Vadgaon Budruk, Pune, Maharashtra 411041, India

[2]Professor and Head (Guide), Department of Mechanical Engineering, Sinhgad College of Engineering, Vadgaon Budruk, Pune, Maharashtra 411041, India

---

Abstract - This paper presents the design of an Arduino-based automatic hot-water machine that coordinates liquid-level monitoring, water filling, heating, and user indication. The proposed controller combines an ultrasonic level sensor, a digital temperature sensor, relay-isolated actuators, an inlet pump or motorized valve, an electric heater, and a liquid-crystal display. A state-based control method gives level safety priority: heating is permitted only when a valid minimum water level is available, filling stops at the upper threshold, and the heater cycles around the selected temperature using hysteresis. The architecture addresses common problems in manually operated tanks, including overflow, dry heating, delayed response, water wastage, and dependence on continuous supervision. Functional analysis shows how sensor validation, interlocks, timeout logic, and fault shutdown can provide predictable operation, while a structured test plan is defined for later prototype verification. The design is intended as a low-cost platform for domestic and small-scale process applications where automatic level and temperature management are required.

Key Words: Arduino Uno, hot-water machine, liquid-level control, temperature control, ultrasonic sensor, safety interlock.

1. INTRODUCTION

Water-heating systems are often operated as separate filling and heating processes. When the operator must watch the tank, stop the pump, start the heater, and judge completion manually, the process is vulnerable to overflow, insufficient filling, dry heating, and unnecessary energy use. An integrated controller can make these dependent actions explicit and repeatable.

The project information defines an Arduino-based automatic liquid-level adjustment and monitoring system using a level sensor, pump or motorized valve, electric heater, temperature sensor, relay, display, and boiler tank. The design developed here adapts that concept to a hot-water machine: the controller first establishes a safe water inventory, then heats to a selected set point, maintains the temperature within a narrow band, and reports operating state to the user.

The Arduino Uno is suitable for the prototype because its ATmega328P platform provides digital input/output, analog input, timers, serial interfaces, and sufficient program memory for sensor acquisition and sequential control [1]. The controller is not a substitute for certified over-temperature, pressure, leakage, and electrical protection. Those protections remain independent hardware requirements in any deployable appliance.

2. PROBLEM STATEMENT AND OBJECTIVES

The engineering problem is to regulate both water level and temperature without continuous human intervention while preventing unsafe actuator combinations. Level and temperature are coupled: the heater must never operate when the sensed level is below the safe immersion point, and filling should not continue beyond the tank's upper limit.

- Measure tank level and water temperature continuously and reject implausible sensor readings.
- Fill the tank automatically between defined lower and upper thresholds without rapid relay switching.
- Enable heating only after the safe-level condition is confirmed and stop heating at the selected temperature.
- Display the measured values, operating state, and fault condition in a user-readable form.
- Provide fail-safe shutdown for sensor loss, fill timeout, over-temperature, and dry-run risk.

3. SYSTEM ARCHITECTURE

Figure 1 shows the information and power-control paths. Sensors connect to low-voltage controller inputs. Relay or driver stages isolate the Arduino from the pump, valve, and heater circuits. The controller treats the level interlock as the primary permissive for heat generation.

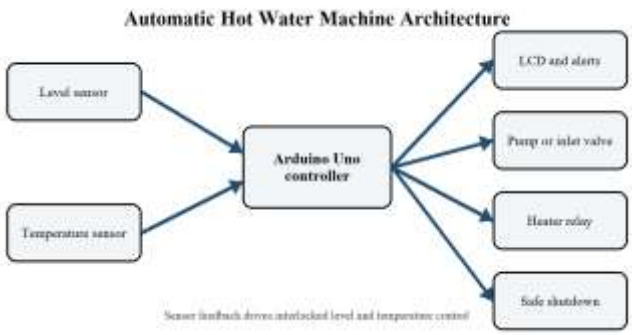


Fig -1: Proposed control architecture

3.1 Hardware Selection

Element	Proposed device	Role in the system
Controller	Arduino Uno	Acquires sensors and executes the control states.
Level	HC-SR04 or equivalent	Measures air gap above the water surface.
Temperature	Waterproof DS18B20	Provides digital temperature feedback.
Actuation	Relay/contactor and valve or pump	Switches mains-rated loads through isolation.
Interface	16 x 2 LCD and switches	Shows level, temperature, state, and alarms.
Process	Insulated tank and heater	Stores and heats the controlled volume.

Table -1: Principal hardware elements

The HC-SR04 class of sensor uses a trigger pulse and measures echo time; its commonly published nominal range is 2 cm to 400 cm [2]. A waterproof temperature probe based on the DS18B20 is suitable for low-voltage measurement because it provides a one-wire digital interface, programmable 9- to 12-bit resolution, and a specified operating range from -55 °C to 125 °C [3]. The selected probe, enclosure, insulation, relay, wiring, and heater must be rated for the actual water temperature, supply voltage, current, and environment.

3.2 Measurement Model

For an ultrasonic sensor mounted a known distance H above the tank bottom, the measured air gap d is estimated from the echo duration t as $d = c(T)t/2$, where $c(T)$ is the speed of sound at ambient temperature. The water level is then $L = H - d$. Calibration should determine the empty and full references in the installed tank because surface turbulence, foam, condensation, sensor angle, and tank geometry can bias the echo.

The temperature channel should be checked for communication errors and rate-of-change plausibility before it is used for control. A moving median or short average can suppress isolated samples, but filtering must not delay fault detection. Sensor validity is therefore evaluated separately from the displayed, smoothed reading.

4. CONTROL METHODOLOGY

The firmware is organized as a non-blocking state machine so that sensor acquisition, display refresh, actuator control, and safety checks continue concurrently. Figure 2 summarizes the main sequence. A watchdog and bounded timers prevent the program from remaining indefinitely in a fill or heat state.

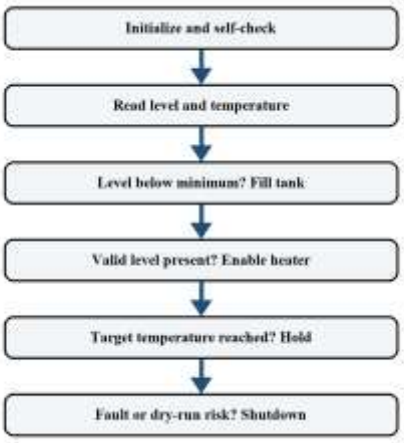


Fig -2: Control and safety sequence

4.1 Level Control

Two level thresholds provide hysteresis. When L is below Llow, the controller opens the inlet valve or runs the pump. Filling continues until L reaches Lhigh. The separation between Llow and Lhigh prevents rapid switching caused by surface ripple or measurement noise. A maximum fill time detects an empty source, blocked line, failed pump, or unreliable level signal.

4.2 Temperature Control

Heating is permitted only when the level-valid flag is true and L exceeds the heater-safe threshold. With a target temperature Tset and hysteresis band DeltaT, the heater turns on when T is at or below Tset - DeltaT and turns off at or above Tset. A separate hardware high-limit cut-out must interrupt heater power even if the microcontroller or its relay fails.

4.3 Operating States

State	Controller action
STARTUP	Outputs off; initialize display and validate sensors.
FILL	Admit water until the upper threshold or timeout.
HEAT	Energize heater only with confirmed safe level.
READY	Hold temperature, display availability, and monitor demand.
FAULT	De-energize heater and pump/valve; latch a diagnostic code.

Table -2: State-machine behavior

5. FUNCTIONAL ANALYSIS AND VALIDATION PLAN

The design logic resolves the most important failure modes before prototype performance is measured. Overflow is limited by the upper threshold and fill timeout. Dry heating is limited by a level permissive that is checked continuously, not only when heating starts. Relay chatter is reduced by hysteresis and minimum switching intervals. A

latched fault state makes failures visible and requires an intentional reset after the cause is removed.

Because the supplied project record does not include calibrated readings, energy data, or repeated trials, quantitative accuracy and efficiency are not claimed. Prototype verification should record raw sensor values and actuator states against traceable reference instruments. At minimum, the following tests are required:

- Calibrate level at empty, intermediate, and full points and repeat measurements with calm and disturbed water surfaces.
- Compare temperature readings with a calibrated thermometer through the intended operating range.
- Verify fill stop, heater cut-off, hysteresis, and recovery after a brief power interruption.
- Disconnect or obstruct each sensor and confirm transition to the safe fault state.
- Simulate pump failure, empty source, stuck relay, and low-water conditions without energizing unsafe mains loads.
- Measure heat-up time and electrical energy for a known water mass only after protective hardware is installed.

6. IMPLEMENTATION AND SAFETY CONSIDERATIONS

The low-voltage controller and sensor wiring should be physically segregated from heater and pump conductors. The Arduino output must drive a correctly rated relay module, contactor, or solid-state switching stage; it must not carry load current directly. Protective earthing, fusing, residual-current protection, strain relief, enclosure ingress protection, thermal insulation, and over-pressure provisions depend on the final appliance and installation.

The LCD can be driven through the standard Arduino LiquidCrystal library when a compatible HD44780-type module is used [4]. The interface should show current level, temperature, set point, active state, and a concise fault code. On restart after a power interruption, outputs should remain off until sensors pass plausibility checks; the controller may then resume the process from the measured physical state rather than assuming that the interrupted state is still safe.

7. CONCLUSION

An Arduino-based controller can integrate water filling, level monitoring, temperature regulation, indication, and fault handling in a compact hot-water machine. The proposed design gives level safety priority, separates sensing from mains-power switching, and uses hysteresis and state-based logic for stable operation. Its main contribution is a clear control architecture and verification plan derived from the project requirements. The next engineering step is to build a protected prototype, calibrate both sensors, execute the defined fault tests, and report measured accuracy, heat-up time, energy consumption, and repeatability before considering practical deployment.

ACKNOWLEDGEMENT

The project team acknowledges the guidance and facilities provided by Algorithmic Electronics during development of the system concept.

REFERENCES

[1] Arduino, 'Arduino UNO R3 Datasheet,' product A000066, revised 28 October 2024. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[2] Elec Freaks, 'Ultrasonic Ranging Module HC-SR04 Product User's Manual.' Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[3] Analog Devices, 'DS18B20 Programmable Resolution 1-Wire Digital Thermometer,' Rev. 6, 2019. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20.pdf

[4] Arduino, 'LiquidCrystal Library for Arduino.' Available: https://github.com/arduino-libraries/LiquidCrystal

[5] Microchip Technology, 'ATmega328P 8-bit AVR Microcontroller Datasheet.' Available: https://www.microchip.com/en-us/product/atmega328p