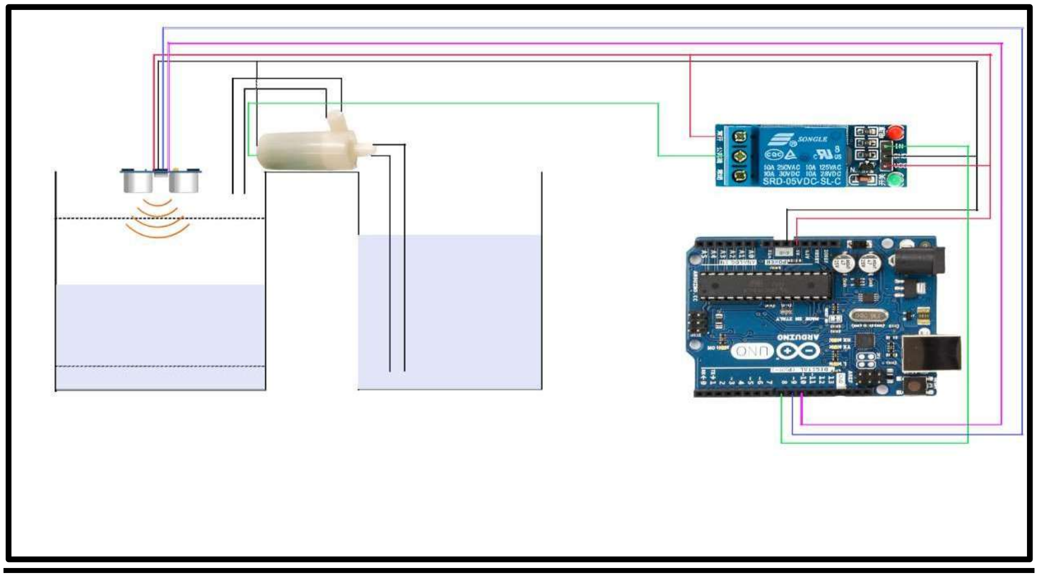
AUTOMATIC WATER PUMP ON/OFF



By, Purbanka Ghosh

A Student of
National Institute of Technology Agartala

Check it out on Github- purbanka0708

Acknowledgement

I gratefully acknowledge the constant and invaluable academic and personal support received from my sister throughout my project work. She helped me make my project report and also gave valuable suggestions.

………………………

Purbanka Ghosh

INDEX

ABSTRACT

This project presents the design and implementation of an **Automatic Water Pump Control System** using an **Arduino Uno** and an **ultrasonic sensor (HC-SR04)**. The primary objective is to automate the operation of a water pump, ensuring the pump is activated only when needed, by turning the pump **ON** when the water level falls below a set minimum and turning it **OFF** when the tank is full. This helps to prevent water overflow and dry running by continuously monitoring the water level, also conserves energy and reduces manual intervention.

The working principle of the system is based on an ultrasonic sensor, which is mounted at the top of the water tank, continuously measures the distance from itself to the water surface. Based on predefined threshold values, the Arduino decides when to activate or deactivate the pump using a **relay module**. If the water level falls below a certain point (indicating low water), the pump is turned ON automatically. Once the tank fills up and the water reaches the high-level threshold, the pump is turned OFF. The relay acts as a switch that controls the power flow to a low-voltage DC water pump.

This setup provides a cost-effective, energy-efficient, and reliable solution to everyday water management problems, especially in households, agriculture, and industrial settings. The system is easy to build and requires minimal maintenance. The simplicity of the project also allows for further upgrades, such as mobile app control, display integration, and water usage analytics. The design is also scalable and can be enhanced further by integrating wireless control, display modules, or IoT-based monitoring.

Overall, this project demonstrates a simple yet effective approach to automating water pump systems using embedded technology, improving reliability, reducing water wastage, and minimizing human effort..

INTRODUCTION

In many households, agricultural fields, and industrial environments, water tanks are manually monitored and filled, which often leads to overflow, water wastage, or dry tanks. To address these inefficiencies, an automatic water pump control system offers a reliable and cost-effective solution for managing water levels without human intervention.

This project presents a smart water-level-based pump controller using an **Arduino Uno**, an **ultrasonic distance sensor (HC-SR04)**, and a **relay-controlled DC water pump**. The system continuously monitors the water level in the tank using the ultrasonic sensor and intelligently switches the pump **ON** when the water level drops below a preset threshold and **OFF** when the tank is full.

This project eliminates manual supervision, conserves energy and water, and can be easily adapted for home or small-scale agricultural use. It is also beginner-friendly and suitable for electronics and embedded systems learners.

WORKING PRINCIPLE

The system operates on the principle of **ultrasonic distance measurement** to detect the water level inside a tank. An **ultrasonic sensor (HC-SR04)** is mounted at the top of the tank and continuously sends out high-frequency sound waves. When these waves hit the water surface, they bounce back and are detected by the sensor's receiver. The time taken for the echo to return is used to calculate the **distance between the sensor and the water surface**.

This measured distance correlates to the water level in the tank:

- If the water level is **low** (i.e., the distance from sensor to water is **greater than a set threshold**, e.g., 15 cm), the **Arduino activates the pump** by sending a signal to the **relay module**, turning the pump ON to fill the tank.
- As the tank fills, the water level **rises** and the distance **decreases**.
- Once the water level is **high enough** (i.e., the distance is **less than a set threshold**, e.g., 5 cm), the Arduino sends another signal to the relay to **turn OFF the pump**, preventing overflow.

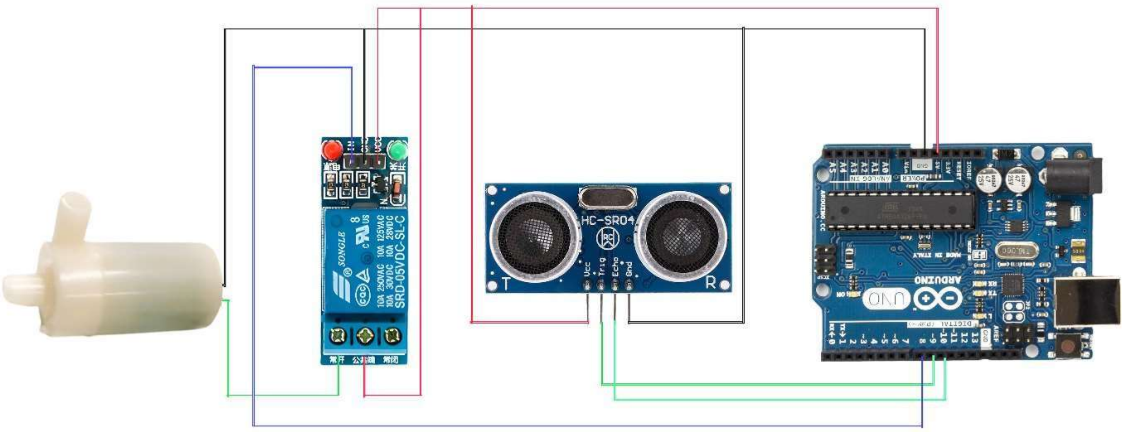


Fig 1: Diagram Showing Connections

Water Level (Distance)	**Arduino Action**	**Pump Status**
Low level (> 15 cm)	Turn relay **ON**	Pump ON
High level (< 5 cm)	Turn relay **OFF**	Pump OFF
In between	Maintain current state	Depends on previous action

This process is **continuous and automated**, ensuring efficient tank refilling without human intervention.

APPARATUS

Component	Specification	Quantity
Ultrasonic Sensor	HC-SR04	1
Arduino Uno	ATmega328P	1
Relay Module	5V 1-Channel	1
Water Pump	5V mini DC pump	1
Jumper Wires	M-M and M-F	10–12
Breadboard	800 approx. pins	1
Power Source	USB	1
Pipe	1 metre	2
Water containers	-	2

1. Arduino Uno (V173):

- **Type:** Microcontroller Board (ATmega328P-based).
- **Purpose:** Acts as the brain of the system. It reads sensor data, processes it, and controls the relay to turn the water pump ON or OFF.
- **Use:** Easy to program, widely available, and has sufficient I/O pins for this project.

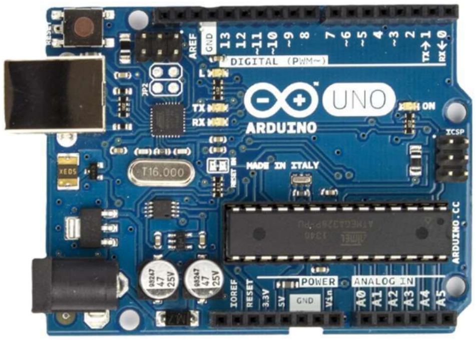


Fig 2: Arduino Uno ATmega328P

2. HC-SR04 Ultrasonic Sensor:

- **Purpose:** Measures the distance from the sensor to the water surface using ultrasonic waves.
- **Working Range:** 2 cm to 400 cm.
- **How It Works:** Sends an ultrasonic pulse and measures the time it takes to reflect back. The distance is calculated based on this time.
- **Use:** Affordable, accurate, and easy to interface with Arduino.

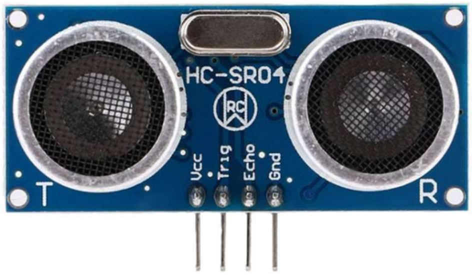


Fig. 3: Ultrasonic Sensor HC-SR04

3. Relay Module (JQC3F-05VDC-C):

- **Type:** 5V Single-Channel Relay Module.
- **Specifications:** Supports up to 10A at 250V AC or 30V DC.
- **Purpose:** Acts as a switch to control the water pump, isolating high-voltage pump circuits from the Arduino.
- **Use:** Safe way to control high-power devices using a low-power microcontroller.

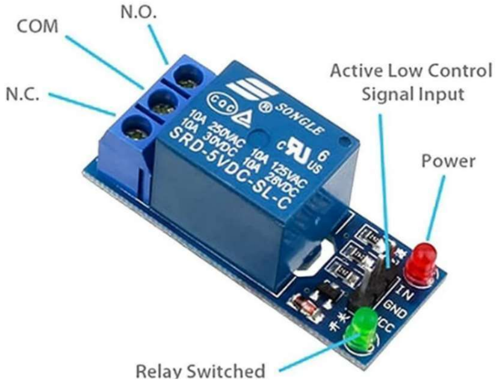


Fig. 4: Relay Module 5V 1-Channel

4. Water Pump (Small DC Pump):

- **Type:** Low-voltage DC Submersible Pump.
- **Purpose:** Moves water into the tank when the level is low.
- **Use:** Small, safe and easily powered by a DC power supply.

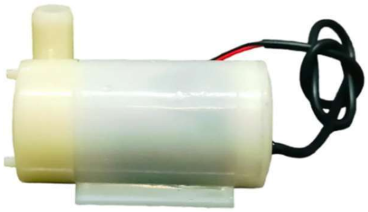


Fig. 5: Water Pump 5V

5. Jumper Wires:

- **Type:** Male-to-Male and Male-to-Female.
- **Purpose:** Used to make electrical connections between components on the breadboard or directly to the Arduino.
- **Use:** Flexible, reusable, and required for prototyping circuits.

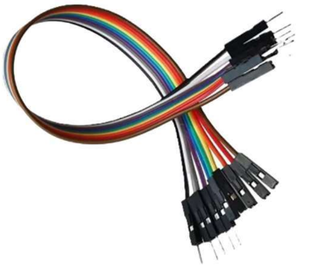


Fig. 6: Jumper Wires

6. Breadboard:

- **Purpose:** A prototyping board to connect components without soldering.
- **Use:** Makes it easy to build and test the circuit before finalizing.

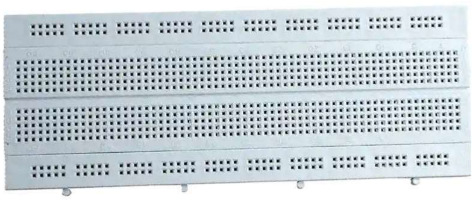


Fig. 7: Breadboard

METHODOLOGY

The development of the **Automatic Water Pump Control System** involves both hardware assembly and software programming. The step-by-step process is described below:

Component Setup:

- **Arduino Uno (v173)**: Acts as the brain of the system.
- **Ultrasonic Sensor (HC-SR04)**: Mounted at the top of the tank to measure water level via distance calculation.
- **Relay Module (JQC3F-05VDC-C)**: Controls the switching of the DC water pump based on Arduino signals.
- **DC Water Pump**: Pumps water into the tank.
- **Breadboard & Jumper Wires**: Used for non-permanent connections between components.

Wiring & Circuit Connections:

- **HC-SR04:**
  - VCC → 5V on Arduino
  - GND → GND on Arduino
  - TRIG → Digital Pin 9
  - ECHO → Digital Pin 10
- **Relay Module:**
  - VCC → 5V on Arduino
  - GND → GND on Arduino
  - IN → Digital Pin 8
- **Pump Control:**
  - The relay is connected in series with the pump and power supply.
  - When the relay is triggered, it closes the circuit and powers the pump.

Programming the Arduino:

- The Arduino continuously triggers the ultrasonic sensor and listens for the echo.
- It calculates the distance using the time delay between sending and receiving the ultrasonic pulse.
- Based on the distance:
  - If distance > low level → turn pump ON
  - If distance < high level → turn pump OFF

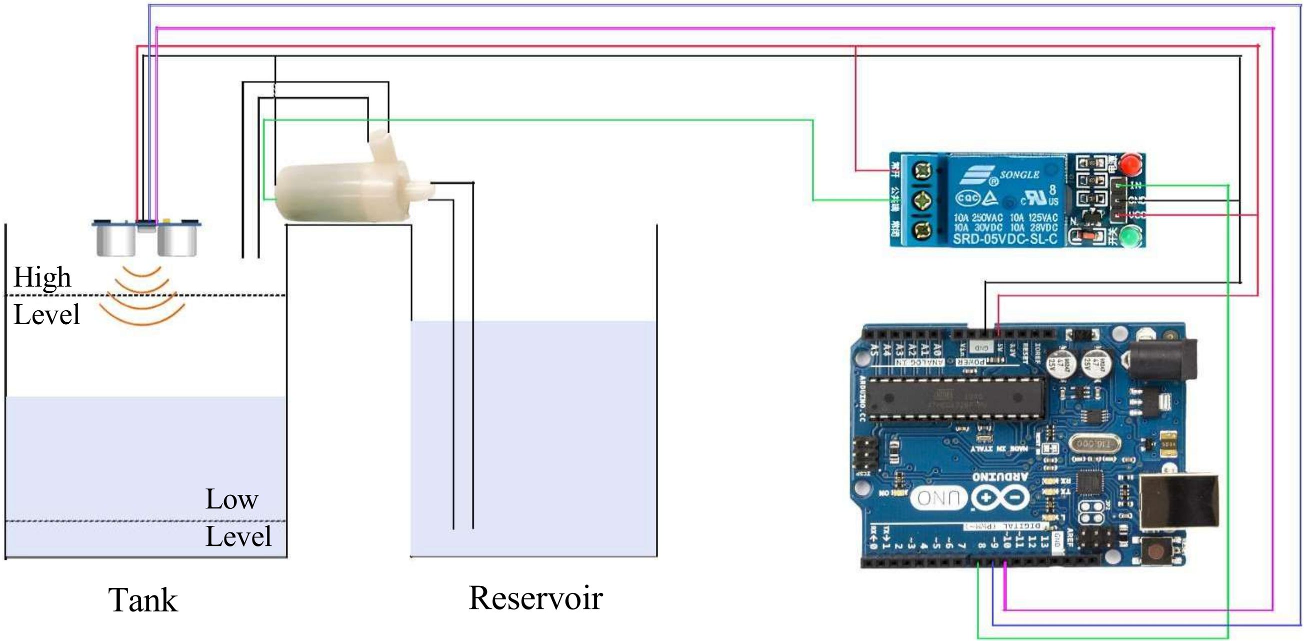
Testing:

- After uploading the code, the Serial Monitor is opened to verify that the distances are correctly measured.
- The pump behaviour is tested by adjusting water level.
- Relay clicks ON/OFF, according to distance thresholds, is ensured.

Final Setup:

- The sensor is firmly mounted at the top of the water tank using a holder or plastic bracket.
- The wires are secured to avoid water contact.
- Optionally, Arduino and circuitry are enclosed in a waterproof or splash-resistant box for safety.

Circuit Diagram:



Circuit Diagram Explanation:

The circuit consists of three main parts: the **ultrasonic sensor**, the **relay module**, and the **DC water pump**, all connected to the **Arduino Uno (V173)** for control and logic.

1. Ultrasonic Sensor (HC-SR04) Wiring:

Sensor Pin	Connected to Arduino	Purpose
VCC	5V	Power supply to the sensor
GND	GND	Common ground
TRIG	D9	Sends ultrasonic pulse
ECHO	D10	Receives the echo to measure distance

- The sensor sends out an ultrasonic wave via the **TRIG pin**.
- The **ECHO pin** listens for the reflected signal.
- The Arduino calculates the distance based on the time it takes for the signal to return.

2. Relay Module Wiring:

Relay Pin	Connected to Arduino	Purpose
VCC	5V	Powers the relay module's internal logic
GND	GND	Common ground
IN	D8	Signal pin to control the relay (active LOW)

- The relay switches the water pump ON or OFF.
- When Arduino sends a **LOW signal to D8**, the relay **activates**, turning the pump ON.
- When the signal is **HIGH**, the relay is OFF and the pump is disconnected.

---

3. Water Pump Wiring (via Relay):

DC water pump is connected through the **normally open (NO)** and **common (COM)** terminals of the relay. Here's how:

- **Pump Positive (V+)** → Connected to **NO (Normally Open)** terminal of relay.
- **COM terminal of relay** → Connected to **positive terminal of external power supply**.
- **Pump Ground (V−)** → Connected to **ground of power supply** (and optionally GND of Arduino if using same source).

This ensures the Arduino controls the pump through the relay **without directly powering it** from the board (which can damage it).

Summary of Arduino Pin Assignments:

Component	Arduino Pin
Ultrasonic TRIG	D9
Ultrasonic ECHO	D10
Relay Control	D8

CODE FOR ARDUINO UNO

```
const int TRIG_PIN = 9;
const int ECHO_PIN = 10;
const int RELAY_PIN = 8;

const int LOW_LEVEL_CM = 15;
const int HIGH_LEVEL_CM = 5;

bool pumpOn = false;

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);

  digitalWrite(RELAY_PIN, HIGH);
  Serial.begin(9600);
}

void loop() {
  long duration;
  float distance;

  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  duration = pulseIn(ECHO_PIN, HIGH, 30000);
  distance = duration * 0.034 / 2;
  if (distance <= 0 || distance > 200 || duration == 0) {
    Serial.println("Invalid distance reading. Skipping...");
    delay(1000);
    return;
  }

  Serial.print("Distance to water: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > LOW_LEVEL_CM && !pumpOn) {
    digitalWrite(RELAY_PIN, LOW);
    pumpOn = true;
    Serial.println("Water LOW → Pump turned ON");
  } else if (distance < HIGH_LEVEL_CM && pumpOn) {
    digitalWrite(RELAY_PIN, HIGH);
    pumpOn = false;
    Serial.println("Tank FULL → Pump turned OFF");
  }

  Serial.print("Pump Status: ");
  Serial.println(pumpOn ? "ON" : "OFF");

  delay(1000);
}
```

Explanation of the code:

1. Pin Configuration

```
const int TRIG_PIN = 9;

const int ECHO_PIN = 10;

const int RELAY_PIN = 8;
```

- **TRIG_PIN & ECHO_PIN**: Connected to the ultrasonic sensor (HC-SR04) to send and receive ultrasonic pulses.
- **RELAY_PIN**: Controls the relay module that switches the water pump ON or OFF.

---

2. Threshold Setup

```
const int LOW_LEVEL_CM = 15;

const int HIGH_LEVEL_CM = 5;
```

- These values define when the pump should turn **ON** (if water is too low) and **OFF** (when the tank is full).
- This should be based on the height of the tank.

---

3. Setup Function

```
void setup()

{

  pinMode(...);

  digitalWrite(RELAY_PIN, HIGH);

  Serial.begin(9600);

}
```

- Initializes the sensor and relay pins.
- Turns **pump OFF** initially (HIGH if relay is active LOW).
- Starts serial communication to monitor readings.

4. Ultrasonic Distance Measurement

```
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH, 30000);
distance = duration * 0.034 / 2;
```

- Sends an ultrasonic pulse.
- Measures the **time it takes to reflect back**.
- Converts duration to **distance in cm**.

---

5. Invalid Distance Handling

```
if (distance <= 0 || distance > 200 || duration == 0)
{
  Serial.println("Invalid distance reading. Skipping...");
  delay(1000);
  return;
}
```

- Filters out readings that are **out of range** or failed.
- Prevents the system from acting on bad data.

6. Pump Control Logic

```
if (distance > LOW_LEVEL_CM && !pumpOn) {
  digitalWrite(RELAY_PIN, LOW); // Pump ON
  pumpOn = true;
}
else if (distance < HIGH_LEVEL_CM && pumpOn) {
  digitalWrite(RELAY_PIN, HIGH); // Pump OFF
  pumpOn = false;
}
```

- **Turns pump ON** when the tank is low (distance is greater than threshold).
- **Turns pump OFF** when the tank is full (distance is smaller than threshold).

---

7. Status Printing

```
Serial.print("Distance to water: ...");
Serial.println(pumpOn ? "ON" : "OFF");
```

- Displays real-time water level and pump status on the Serial Monitor.

---

8. Loop Delay

```
delay(1000);
```

- Adds a 1-second delay between each cycle to reduce flickering and unnecessary relay toggling.

EXPENSES (COST ESTIMATION)

The following table lists the estimated cost of each component used in the project. Prices may vary slightly depending on availability and quality, but this provides a reasonable cost breakdown for the setup:

Component	Quantity	Estimated Cost (INR)
Arduino Uno (V173)	1	₹800
Ultrasonic Sensor (HC-SR04)	1	₹180
Relay Module (JQC3F-05VDC-C)	1	₹200
Small DC Water Pump (5V)	1	₹100
Breadboard	1	₹80
Jumper Wires (Male-Male, Male-Female)	20+	₹100
USB Cable (for Arduino)	1	₹90
Water Pipe	1 meter	₹50

Total Estimated Cost: ₹1500 – ₹2000

This project is cost-effective and the components are readily available both online and in local electronics markets.

DATA AND RESULTS

This section shows the observed behaviour of the system during testing and simulation based on water level changes.

Sensor Behaviour and Distance Measurement:

The **HC-SR04 Ultrasonic Sensor** was used to measure the distance from the top of the tank to the water surface. The Arduino converts the time taken by the echo to return into distance using the formula:

Distance (cm) = (Echo Time in microseconds × 0.0343) / 2

Pump Control Logic and Results:

Measured Distance (cm)	Pump Status	Expected Behavior	Observed Behavior
Low level (> 15 cm)	ON (Relay = HIGH)	Pump should start filling	Pump turns ON
Between (5 – 15 cm)	Remains in last state	Monitoring only	Stable monitoring
High level (< 5 cm)	OFF (Relay = LOW)	Pump should stop	Pump turns OFF

Loop Execution:

- The system continuously monitored the water level every few milliseconds.
- Serial Monitor output displayed real-time distances.
- No major lag or misreadings were observed after stabilizing sensor alignment and connections.

Test Results Summary:

- **Response Time:** Less than 1 second delay in switching.
- **Sensor Accuracy:** ~±1 cm under still air and dry conditions.
- **System Stability:** Stable for continuous 10–15 minute tests.

ADVANTAGES

The project offers several practical and technical benefits that make it useful for home automation and water conservation:

1. Water Conservation:

- Prevents water overflow by automatically turning off the pump when the tank is full.
- Helps save water, especially in areas facing water shortages.

2. Energy Efficiency:

- Avoids unnecessary pump operation, reducing electricity usage.
- Extends the life of the pump by preventing overuse.

3. Low Maintenance:

- Uses solid-state electronics with minimal moving parts.
- Once installed, requires little to no maintenance.

4. Automation & Convenience:

- Fully automatic – no manual switching required.
- Ideal for busy households or remote operation (with future upgrades).

5. Affordable:

- Uses inexpensive, easily available components.

DISADVANTAGES

While the system is highly useful and functional, there are a few limitations to consider:

1. Sensor Limitations:

- Ultrasonic sensors can give false readings in:
    - Splashing water
    - Sudden temperature changes
- This may affect accuracy unless properly shielded and mounted.

2. Dependency on Power Supply:

- System fails if there's no power backup.
- Requires a continuous power supply to function.

3. Risk of Short Circuit:

- If wiring is not properly insulated or enclosed, it may cause a short circuit or component damage.

4. Limited Pump Control:

- Advanced features like scheduling or remote control require extra modules (e.g., WiFi/Bluetooth).

5. Calibration Needed:

- The system might need distance calibration based on tank height.

APPLICATIONS

The Automatic Water Pump Control System has a wide range of real-world applications, especially in areas where efficient water use and convenience are critical.

1. Domestic Water Tanks:

- Automatically fills overhead tanks at homes without human intervention.
- Prevents overflow and dry running, protecting the motor and saving resources.

2. Apartment Buildings & Hostels:

- Can be extended to control larger tanks shared by multiple units.
- Helps in centralized water management and reduces the need for a caretaker.

3. Agricultural Irrigation Systems:

- Monitors water levels in small reservoirs.
- Can be paired with timers to automate farm irrigation.

4. Industrial Water Management:

- Used in water storage systems in factories to regulate usage.
- Helps in efficient and waste-free resource management.

FUTURE SCOPE

The current model of the automatic water pump controller is functional and ideal for small-scale use, but there are several ways the system can be improved and scaled in the future:

1. Mobile App Integration:

- A smartphone app could be developed to monitor the water level remotely and receive alerts or notifications when the tank is full or empty.
- It can also allow manual control of the pump from anywhere.

2. IoT and Cloud Integration:

- Incorporating Wi-Fi modules (like ESP8266/ESP32) or Bluetooth for wireless data transfer.
- Useful for smart homes and agricultural automation.

3. Multiple Tank Support:

- Monitor and control multiple tanks using a multiplexer or additional sensors and relays.
- Beneficial for farms, apartment buildings, and industrial facilities.

4. Automatic Overflow Protection:

- Adding sensors to detect leaks and trigger alarms or shut off the system to prevent water wastage or damage.

5. Solar Power Support:

- Integrate solar panels and battery management systems to make the system energy-efficient and off-grid.
- Particularly useful in remote areas with limited electricity access.

6. Dry Run Protection:

- A flow sensor or water presence detector can be added to prevent the pump from running when the water source is dry, thereby protecting the motor from damage.

7. LCD/Touchscreen Interface:

- A display module (like 16x2 LCD or OLED) can be added to show real-time tank level, pump status, and error messages on the hardware itself.

CONCLUSION

The **Automatic Water Pump Control System** using an **Arduino Uno** and **ultrasonic sensor (HC-SR04)** proves to be a simple yet effective solution for automating water tank refilling. It successfully addresses common issues like tank overflows, dry running of pumps, and manual monitoring.

By accurately measuring water levels and controlling the pump based on set distance thresholds, the system conserves water, saves electricity, and reduces human effort. The project is low-cost, easy to implement, and customizable for various domestic, agricultural, and industrial applications.

Though it has some limitations, such as sensor sensitivity and power dependency, these can be overcome with minor improvements like waterproof enclosures, power backups, or sensor filtering. Overall, it is an excellent automation project that introduces key concepts in embedded systems and real-world interfacing.

REFERENCES

[1] Arduino, “Arduino UNO R3 (SKU A000066) User Manual,” *Arduino Official Documentation*, 22 May 2025.

[2] TeachEngineering (Univ. of Colorado Boulder), “How Does an Ultrasonic Sensor Work?,” 2014.

[3] ElectricalTechnology.org, “What Is an Electromechanical Relay? (EMR) – Types, Working and Applications,” Apr. 2024.

[4] VOVYO Pump (www.vovyopump.com), “DC Water Pump: Types and Applications,” 2023.

[5] Arduino Project Hub, “Water Pump Automation with Ultrasonic Sensor,” Apr. 10, 2021.

[6] K. S. Varun, K. A. Kumar, V. R. Chowdary, and C. S. K. Raju, “Water Level Management Using Ultrasonic Sensor(Automation),” Int. J. Comput. Sci. Eng., vol. 6, no. 6, Jun. 2018. [Online]. Available: https://www.researchgate.net/publication/326555781_Water_Level_Management_Using_Ultrasonic_SensorAutomation

[7] A. Nagpure, S. Naidu, and G. Reddy, “Water Level Controller Using Ultrasonic Sensor,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 12, pp. 252–257, Dec. 2022, doi: 10.22214/ijraset.2022.47856.

[8] M. A. Baballe et al., “A Review of an Automatic Water Level Indicator,” Global J. Res. Eng. Comput. Sci., vol. 02, no. 03, pp. 13–17, May–Jun. 2022, doi: 10.5281/zenodo.6551413.

[9] Arduino, "Arduino Uno Rev3," Arduino Official Documentation. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3. Accessed: May 10, 2025.