



University of Asia Pacific

Department of Computer Science and Engineering

CSE 316: Microprocessors and Microcontrollers Lab

LAB REPORT

Experiment Number: 04

Experiment Title: Water Level based Automatic Pump Control

Submitted by:

Name : H. M. Tahsin Sheikh

Student ID : 22201243

Section : E1

Submitted to:

Zaima Sartaj Taheri

Lecturer,

Department of Computer Science and Engineering

Date of Submission: 17/09/25

1. Experiment Name

Mini Project 4: Water Level based Automatic Pump Control.

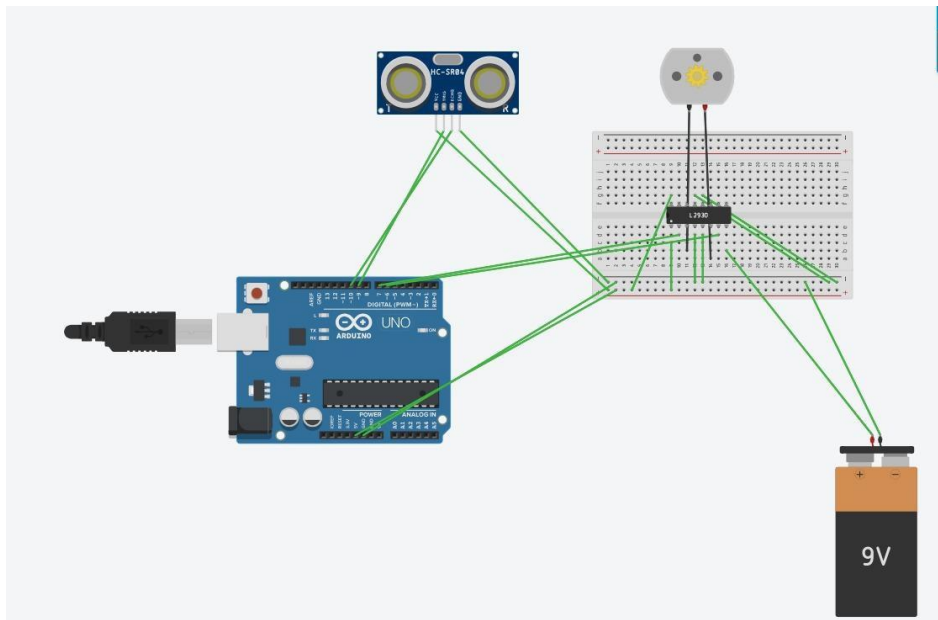
2. Objective

The goal of this work is to design a simple water-level-based pump control system using Arduino UNO, an ultrasonic sensor to detect tank levels, and a DC motor to pump water. The pump should start automatically when the water level is low and stop when the tank is full. This mimics modern automatic pumping systems in households and industries.

3. Apparatus / Hardware & Software Requirements

- Arduino Uno (or other compatible microcontroller)
- USB cable and computer
- Arduino IDE software
- Ultrasonic Sensor (HC-SR04)
- DC Motor/Pump
- L293D Motor Driver IC
- TinkerCAD
- Breadboard, jumper wires, and power supply

4. Circuit Diagram / Schematic



5. Code / Assembly Program

```
const int TRIG = 9;

const int ECHO = 10;

const int IN1 = 6;

const int IN2 = 7;

long duration; int distance;

const int minDistance = 30;

const int maxDistance = 200;

void setup() {

  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  Serial.begin(9600);

}

void loop() {

  // Send ultrasonic pulse

  digitalWrite(TRIG, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG, LOW);

  // Measure echo

  duration = pulseIn(ECHO, HIGH);
```

```

distance = duration * 0.034 / 2;

// Print distance

Serial.print("Water Level Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Pump control

if (distance > minDistance && distance
<= maxDistance) { // Tank low → pump ON

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

Serial.println("Pump ON");

}

else if (distance <= minDistance) {

// Tank full → pump OFF

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

Serial.println("Pump OFF");

}

else {

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

Serial.println("Pump OFF (Safe Range)");

}

delay(1000); }

```

6. Output / Observations



When the tank is low (distance greater than the minimum limit but within range), the ultrasonic sensor detects it and the Arduino activates the pump. As water rises, the distance decreases and the pump keeps running. Once the tank reaches the upper threshold (distance less than or equal to minimum), the pump is turned off to prevent overflow. The system continues to repeat this automatically.

7. Result

The system functioned correctly, measuring water levels through the ultrasonic sensor and controlling the pump automatically. It switched the pump ON at low levels and OFF when full, matching the expected behavior of an automatic water pump controller.

8. Conclusion

Through this experiment, I learned how to connect sensors and actuators with Arduino and write logic to control them. It gave me hands-on practice in using an ultrasonic sensor (HC-SR04) to detect water levels and the L293D motor driver to manage a pump. The task improved my understanding of automating processes using real-time sensor feedback.